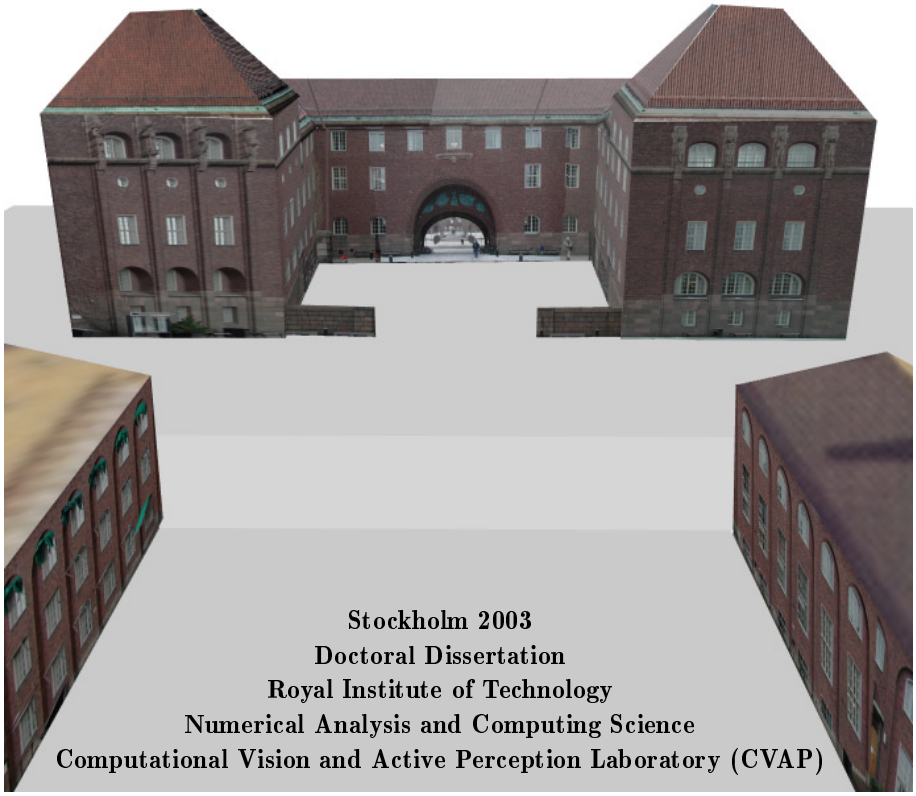
KUNGL
TEKNISKA
HÖGSKOLAN

Multi-View Reconstruction and Camera Recovery using a Real or Virtual Reference Plane

Carsten Rother



Stockholm 2003
Doctoral Dissertation
Royal Institute of Technology
Numerical Analysis and Computing Science
Computational Vision and Active Perception Laboratory (CVAP)

KUNGL
TEKNISKA
HÖGSKOLAN

Multi-View Reconstruction and Camera Recovery using a Real or Virtual Reference Plane

Carsten Rother

Stockholm 2003
Doctoral Dissertation
Royal Institute of Technology
Numerical Analysis and Computing Science
Computational Vision and Active Perception Laboratory (CVAP)

Akademisk avhandling som med tillstånd av Kungl Tekniska Högskolan framlägges till offentlig granskning för avläggande av teknisk doktorsexamen måndagen den 27 januari 2003 kl 10.00 i Kollegisalen, Administrationsbyggnaden, Kungl Tekniska Högskolan, Valhallavägen 79, Stockholm.

TRITA–NA–0226
ISSN 0348–2952
ISBN 91–7283–422–6
CVAP 271

Abstract

Reconstructing a 3-dimensional scene from a set of 2-dimensional images is a fundamental problem in computer vision. A system capable of performing this task can be used in many applications in robotics, architecture, archaeology, biometrics, human computer interaction and the movie and entertainment industry.

Most existing reconstruction approaches exploit one source of information to tackle the problem. This is the motion of the camera, the 2D images are taken from different viewpoints. We exploit an additional information source, the *reference plane*, which makes it possible to reconstruct difficult scenes where other methods fail. A *real* scene plane may serve as the reference plane. Furthermore, there are many alternative techniques to obtain *virtual* reference planes. For instance, orthogonal directions in the scene provide a virtual reference plane, the plane at infinity, or images taken with a parallel projection camera. A collection of known and novel reference plane scenarios is presented in this thesis.

The main contribution of the thesis is a novel multi-view reconstruction approach using a reference plane. The technique is applicable to three different feature types, points, lines and planes. The novelty of our approach is that *all* cameras and *all* features (off the reference plane) are reconstructed *simultaneously* from a *single linear system* of image measurements. It is based on the novel observation that cameras and features have a *linear* relationship if a reference plane is known. In the absence of a reference plane, this relationship is *non-linear*. Thus many previous methods must reconstruct features and cameras sequentially. Another class of methods, popular in the literature, is factorization, but, in contrast to our approach, this has the serious practical drawback that all features are required to be visible in all views. Extensive experiments show that our approach is superior to *all* previously suggested reference plane and non-reference plane methods for difficult reference plane scenarios.

Furthermore, the thesis studies scenes which do not have a unique reconstruction, so-called critical configurations. It is proven that in the presence of a reference plane the set of critical configurations is small.

Finally, the thesis introduces a complete, automatic multi-view reconstruction system based on the reference plane approach. The input data is a set of images and the output a 3D point reconstruction together with the corresponding cameras.

Acknowledgements

Since I anyway did not achieve the goal of writing a short thesis (more than 200 pages – who is going to read it?) there is no need of having a short acknowledgment section.

First of all I would like to thank my supervisors **Stefan Carlsson** and **Jan-Olof Eklundh**. I owe Stefan the reference plane idea, which obviously affected my work in the last years (please see the following 215 pages). Moreover, I would like to thank him for the support and great enthusiasm about new ideas. Jan-Olof for being always a source of new ideas and being concerned about many aspects of life, not always related to research.

Thanks to **all** people at CVAP (no one excluded) for the nice "working" atmosphere and for sharing ideas. **Ivan** for the pleasant company at past and future hiking trips with a cool bottle of vodka. **Dennis** for joint work and good barbecues. **Tony** for many good discussions and tips about the Sarek. **Philipp** and **Frank** for keeping the German board game culture alive. **Fredrik** and **Ola** for a nice lunch while missing all blue wales. **Johan** for always joining me for pannkakor med ärtsoppa at GIH. **Jeanna** and **Emma** for their help with administration matters. A special thank goes to **Eric** and **Josephine** for spreading a nice atmosphere and for proof-reading the thesis, i.e. spotting all i.e.'s and e.g.'s. **Martin, Peter, Lars B., Daniel, Ronnie, Dani, Hedvig** and **Mårten** for many computer vision and non-computer vision related discussions. Many thanks go as well to "computer vision people" outside CVAP, for fruitful discussions and sharing software. Some of them are Andrew Zisserman, Bill Triggs, Richard Hartley, Marc Pollefeys, Fredrik Kahl, David Liebowitz, David Nistér, Thomás Pajdla, Daniel Martinec, Peter Sturm and Adrien Bartoli.

Ein besonderer Dank geht an meine Eltern und meine Schwester die, trotz der Distanz, mich stets unterstützen und sich um mich sorgen. Last but not least, I would like to thank **Gerit** for her love, support and the great idea of moving to Sweden.

This work was supported by the Swedish Foundation for Strategic Research in the VISIT program. The funding is gratefully acknowledged.

Contents

Chapter 1

Introduction

This chapter gives a gentle introduction to the topic of 3-dimensional reconstruction. The main and novel contributions of the thesis are highlighted. Furthermore, an outline of the following chapters is given. The first section motivates our approach to 3-dimensional reconstruction from a human perspective. The reconstruction problem is then formulated more mathematically. The difference between assuming a known or unknown reference plane is described. This will reveal the novelty of our method. Finally, the main contributions, an overview of the thesis and a list of publications are presented.

1.1 Reconstruction from a Human Perspective

Various information sources for 3-dimensional reconstruction
Motion (of the observer or the scene)
Parallel and orthogonal lines
Shape and size of familiar objects
Stereo (binocular) vision
Shading of a surface
Position relative to the horizon
Shadows induced by a light source

Table 1.1. Different cues to derive 3-dimensional information from an image.

Vision is probably the most important sense for humans. It allows many potentially complicated tasks to be performed with ease, such as walking along a corridor, picking up an object or orientating oneself in a city. To perform these complex tasks successfully, 3-dimensional knowledge about our environment is helpful or even necessary[1]. How do we

[1]For which tasks humans exploit 3-dimensional knowledge is an interesting and still open research question.

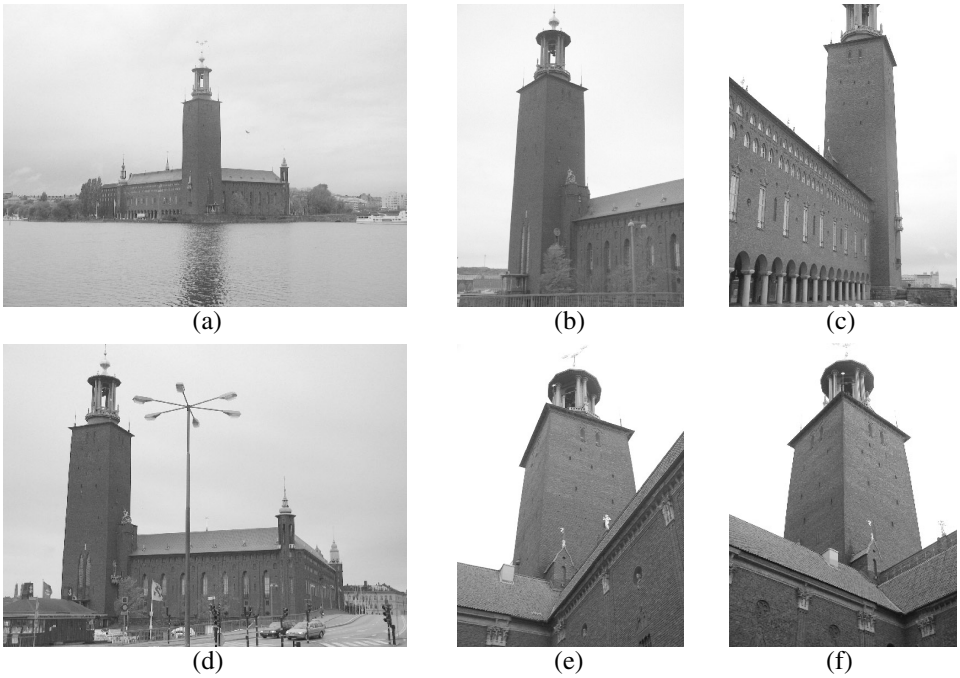


Figure 1.1. Six images of the city hall in Stockholm. The positions from where the images were taken are depicted on the 2D map in fig. 1.2.

get this knowledge? An image of our environment provides many cues about the scene's depth. Table 1.1 lists some of these information sources (see Palmer (1999) for a complete list). One of the goals of psychophysics is to understand how humans use these information sources (Gibson, 1950; Palmer, 1999). One ultimate goal of computer vision is to imitate the human, to build a computer that "sees". Marr (1982) and his colleagues, e.g. Ullman (1979), introduced this idea. To achieve this, the computer system has to automatically derive information from various sources. The goal of this thesis is to create a virtual 3D reconstruction of a scene from images of it. Most previous reconstruction systems exploit merely *one* source of information. In this work we use *two* sources and show that this gives a system which is superior to previous systems.

Let us explain this simple idea with an experiment. Fig 1.1 illustrates 6 pictures of the tower of the city hall in Stockholm taken from different viewpoints. Your task is to draw a top view (map) of this scene including the tower and the positions from where the pictures were taken. Although this task is not simple, we are able to solve it approximately[2]. Fig. 1.2 shows the accurate result[3]. The bold 6 arrows represent the viewpoints with respect

[2]Probably this task is easier for me after working with these images for $3\ 1/2$ years.

[3]The last chapter 9 discusses the fact that humans are better in performing this task approximately, qualitatively, than very accurately.

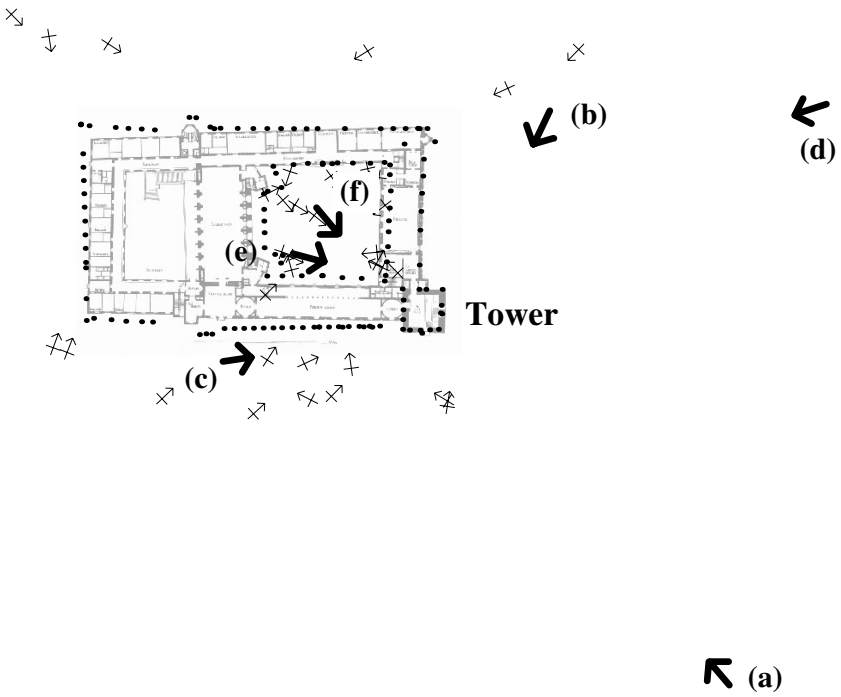


Figure 1.2. Top view of the reconstruction of the city hall with 134 model points (dots) and 37 cameras (arrows). A map of the city hall is superimposed. The 6 bold arrows correspond to the locations where the images in fig. 1.1 were taken.

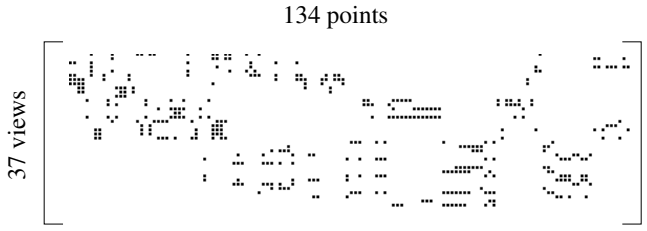


Figure 1.3. The "visibility" matrix of the city hall scenario. A dot indicates that a certain 3D point is visible in a certain view.

to a 2D map of the complete city hall. Which information sources in the images help us, as humans, to solve this specific reconstruction task? The first information cue in table 1.1 is motion. By walking through an environment, we constantly change the viewpoint with respect to a static object. This continuous image sequence can be used to infer the

object's structure. The 6 images in fig 1.1 represent a very discrete motion. In general, 3D information of a static object can be derived from two or more views taken from different viewpoints. The second source of information, parallel and orthogonal lines, is a very important cue in man-made environments. Parallel lines define a direction in the scene and orthogonal lines give rise to a right angle in the 3D scene. Using this information, we are able to infer a rough shape of an object, like the tower is a rectangular box. Furthermore, orthogonal scene directions may be used to orientate yourself in the environment. For example, the orientation of viewpoint 1.1 (a) in the top view (fig. 1.2) is defined by the orthogonal directions of the tower. However, the distance (position) between the viewpoint and the tower is undefined given the directions only. The third source of information, shape and size of familiar objects, is a very useful cue as well. From image 1.1(f) we suspect that the two sides of the tower have equal length. However, it is not possible to prove this geometrically on the basis of one image. Image 1.1(d) gives another example. It depicts a lamp post and the tower which have approximately the same size in the image. We know that a lamp post is between 4 and 8 meters tall. A tower is usually taller than 8 meters[4]. Therefore, the city hall has to be several meters farther away than the lamp post. This experiment could almost continue indefinitely with the reader in the role of the detective searching for clues about the 3D structure, however, let us now summarize the discussion. A wide variety of information sources exist to infer 3D information (depth) from one or more images. We may conjecture that humans exploit combinations of information sources to solve the reconstruction task. The choice of sources might also depend on the observed environment. Furthermore, humans may exploit the different cues cumulatively, i.e. not in isolation. For example, the orientation with respect to a 3D scene may be derived from parallel and orthogonal lines. Given the orientation, the position in the scene may be determined from the motion cue. However, on the basis of the camera's motion only, the task of deriving both the orientation and the position might be significantly more difficult. As we will see, this is the basic idea of our computer system for 3D reconstruction.

Before continuing the experiment, it should be noted that the information sources may also be used to fool humans resulting in the hallucination of incorrect 3D scenes. To derive information about the scene humans make implicit assumptions. If these assumptions are violated, we infer wrong information. This is a well known trick in architecture. Ragnar Östberg, the architect of the city hall in Stockholm, used this trick in the early 1900's when designing the tower of the city hall. The cross-section of the tower is larger at the bottom than at the top. Since humans suspect that the tower is a rectangular box, this makes them hallucinate a taller tower[5]. Artists like MC Escher perfected this skill (see fig. 1.4).

Consider how a computer system may solve this specific reconstruction task. The 6 images in fig. 1.1 and a further 31 images of the city hall are fed into our reconstruction system. Moreover, the position of 134 projected 3D points are identified in the 37 images (matching task). Fig. 1.3 shows the "visibility" matrix, where a dot indicates that a certain 3D point is visible in a certain view. As expected, most of the 3D points are only visible in a limited subset of views. In this case 90% of the image data is missing. The output

[4]Probably people in the computer vision community remember the size from the banquet at ECCV 94.

[5]We would like to thank Jan-Olof Eklundh for pointing this out to us.

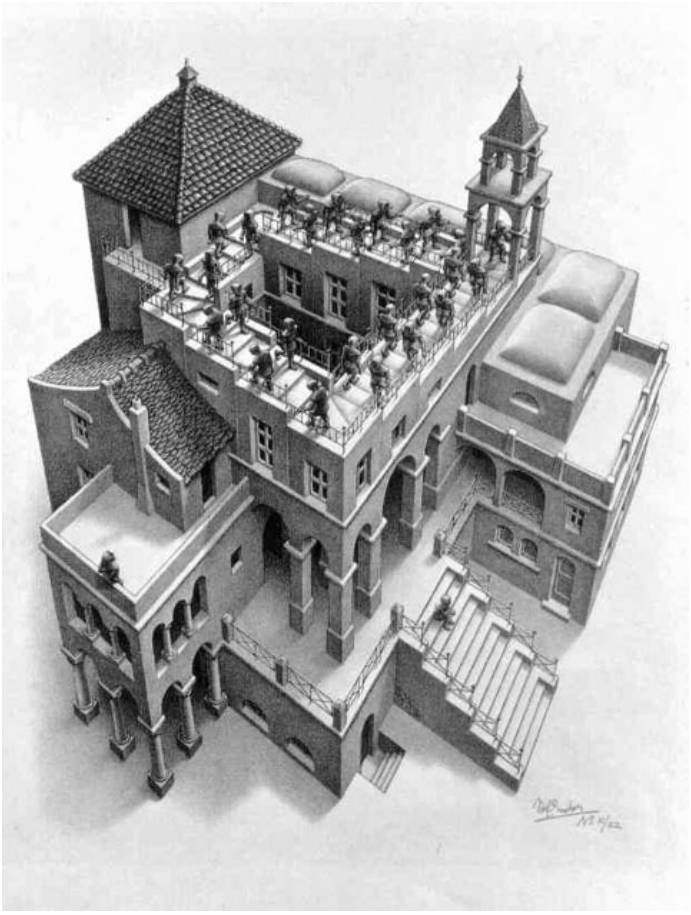


Figure 1.4. *Ascending and descending*, a famous picture by MC Escher.

Figure 1.5. Three novel views of the virtual model of the city hall.

of the system is depicted in fig. 1.2. The 3D points are plotted as black dots and the cameras corresponding to an image as an arrow. How did our system achieve this? It exploits cumulatively *two* information sources, (a) parallel and orthogonal lines and (b) the camera's motion. Reconsider the information from parallel and orthogonal lines. Image 1.1(f) shows explicitly that man-made environments are often characterized by 3 mutually orthogonal directions[6]. Parallel and orthogonal lines in the scene define these 3 directions. If they are identified in an image, two properties about the corresponding camera may be derived, its *orientation* and its *internal parameters*[7], like the focal length. Abstractly, common scene knowledge "available" in all views can be used to derive information about all cameras. The remaining unknown information is the 3D positions of the 134 points and the 37 cameras (viewpoints). To solve this problem, the second information source (motion) is used. In summary, our reconstruction system uses two sources of information cumulatively:

1. Compute the cameras' *orientation* and *internal parameters* from parallel and orthogonal lines.

2. Use this information to derive the *3D positions* of the cameras and points from the camera's motion.

From the reconstruction of the 3D points, a textured virtual model may be extracted. Fig. 1.5 shows novel views of the virtual model of the city hall.

The next section explains mathematically why this two-step method is significantly superior to methods which use motion as the only information source. The main advantage is that with some knowledge about all cameras (first step), the 3D positions of all cameras and all points can be computed *directly* and *simultaneously* (second step). If motion is the only information source, such a simple solution does not exist. The cameras and points must be estimated *sequentially*. On the basis of a few, noisy image measurements these approaches sequentially accumulate the reconstruction error and eventually fail. Experiments will confirm that motion based systems cannot reconstruct the city hall from the given image data with 90% of missing measurements (fig. 1.3).

An obvious question is, why does our system not rely on more information sources? This is an interesting and open question for future research. However, one has to keep in mind that inferred 3D information is not always correct, as demonstrated by Escher's drawing (fig. 1.4). The attentive reader might wonder how the second part of title of the thesis "using a real or virtual reference plane" fits into the framework just discussed. The above presentation is a simplified (Euclidean) version of our system. The next section explains that the information source of *parallel and orthogonal lines* can be expressed more generally (projective) as a *real or virtual reference plane*.

[6] The reader might think of a Cartesian coordinate system.
[7] This is defined mathematically in the next chapter.

1.2 The Reconstruction Problem

This thesis addresses one of the most fundamental problems in computer vision, determining 3-dimensional information about a scene from 2-dimensional images of it. The previous section motivated the importance of 3D information for a general computer vision system that "sees", i.e. imitates the human vision. Potentially, a reconstruction system can be part of any application where 3D-information is useful or necessary, some specific applications being:

- An architect takes pictures of a city block. For the planning of a new building it is important to have both an accurate reconstruction as well as a nice, virtual model of the city block.
- A reconstruction system is useful for robots to navigate in an unknown environment and to build iteratively a map.
- Insertion of synthetic objects into an existing video sequence is an important task in movie making.
- A biometrics system can identify people from a 3D profile of the face.
- An estate agent provides a virtual tour of a building.
- In medicine, 3D information can be used to guide the doctor during an operation.

The reconstruction problem may be defined as follows: Given a set of images, determine the 3D scene and the viewpoints of the images. This problem can be formulated in a simple way by the introduction of one formula. With this formula we can also compare the general reconstruction problem with our specific (reference plane) reconstruction problem. For simplicity, the scene consists of a set of 3D points called the *structure*. The projection of a Cartesian 3D point $\bar{\mathbf{X}}$ to the image point $\mathbf{x}$ using a camera with Cartesian centre $\bar{\mathbf{Q}}$ and matrix H may be formulated mathematically as[8]

$$\mathbf{x} = H\,(\,\bar{\mathbf{X}} - \bar{\mathbf{Q}}\,)\ .$$

The matrix H is known as the *infinite homography* and will be considered later. The unknown parameters in this equation are the point $\bar{\mathbf{X}}$, and the camera, with centre $\bar{\mathbf{Q}}$ and infinite homography H. As we saw in the previous section, in practice many 3D points are observed by many cameras. Each point visible in a certain view gives one projection equation. The reconstruction problem is to determine the unknown structure, 3D points, and cameras solely from image measurements $\mathbf{x}$. Therefore, this problem is called the *structure and camera recovery problem*. In the literature, it is frequently referred to as the structure and motion problem or the structure from motion problem. However, this formulation is often interpreted to mean that the images are sorted, like from a continuously

[8]For simplicity, the unknown scale factor (depth) of a homogeneous image point is omitted. However, this simplification does not affect the following conclusions.

moving video camera. Furthermore, it implies that the distance between successive camera positions is small. This thesis considers a more general scenario, an unorganized collection of images like a photo album.

In the computer vision community, the first algorithm to solve the reconstruction problem was presented more than twenty years ago by Longuet-Higgins (1981). Since that time a vast number of approaches have been suggested. The most recent and excellent books about this problem are by Hartley and Zisserman (2000) and Faugeras and Luong (2001). So one might wonder what can be gained by yet another publication on the topic. Intuitively, such a well defined problem should be solved by now. Consider, however, the formula on the previous page. The unknown parameters, $\bar{\mathbf{X}}$, H and $\bar{\mathbf{Q}}$, are multiplied together. This means that the problem is *non-linear*. Unfortunately, there is no simple, direct method which can compute the solution of a non-linear problem. However, if the infinite homography H is known, the relationship between the unknown 3D point $\bar{\mathbf{X}}$ and camera centre $\bar{\mathbf{Q}}$ is a subtraction. This gives an extremely simplified *linear* problem. The solution can be computed with well known standard methods, like singular value decomposition. Consequently, all 3D points and all camera centres can be reconstructed simultaneously from a single linear system formed from image measurements only. This simple method is the most important contribution of the thesis. The thesis also considers two other types of features, 3D lines and 3D planes.

The remaining question is, how do we obtain the infinite homography H? The most well known approach is to derive it from a real scene plane visible in the image. This scene plane is called the *real reference plane*. An alternative technique was described in the previous section. Since the infinite homography encodes the camera's orientation and internal parameters, it can also be derived from parallel and orthogonal lines in the scene. In this case, the infinite homography represents a *virtual reference plane*. This thesis presents many different methods to compute real or virtual reference planes.
To summarize, the *reference plane approach* divides the reconstruction task in two steps:

1. Determine a real or virtual reference plane.

2. Use the reference plane to compute simultaneously the *3D position* of the cameras and points from a single linear system.

The difference to the two-step approach in the previous section is that the specific information source of *parallel and orthogonal lines* is replaced by the more general information source of *real or virtual reference planes*. The idea of using a real or virtual reference plane to divide the reconstruction task into two (or more) steps is not novel and has been suggested in many previous works. In some systems a real scene plane is used (e.g. Irani and Anandan, 1996), other methods derive a virtual reference plane by explicitly computing the camera's internal parameters and orientation (e.g. Shum et al., 1998). However, many systems apply the reference plane approach without mentioning it explicitly (e.g. Van Gool et al., 1994). This thesis presents many known and novel techniques for deriving a real or virtual reference plane. In contrast to all previous publications, we show that a known reference plane transforms the difficult, non-linear reconstruction problem into a simple, linear problem.

1.3 Contributions

The main contribution of the thesis is a novel multi-view reconstruction approach for points, lines and planes using a real or virtual reference plane. The approach is linear and reconstructs all cameras and all 3D features (off the reference plane) simultaneously from a single linear system of image measurements. It requires that 3D features are visible only in a minimum number of views. For reference plane scenarios, this makes it potentially superior to all previously presented reconstruction methods. We call it the *D*irect *R*eference *P*lane (DRP) approach. It is based on the novel observation that the general *non-linear* relationship of cameras and 3D features is *linear* if a reference plane is known. This simple result was first presented for point features in (Rother and Carlsson, 2001). Indeed, most, but not all, parts of the thesis and of the following contributions are limited to point features.

The experiments performed in this thesis demonstrate that our method can reconstruct difficult reference plane scenes where general (non reference plane) reconstruction methods fail. Furthermore, for some difficult scenarios our method is significantly superior to all previously suggested reference plane methods. However, we also show that reference plane methods are inferior to general methods if the reference plane is detected very inaccurately. The main drawback of our method is that 3D points on and off a finite reference plane have to be reconstructed separately. It is demonstrated experimentally that our method is very stable for scenarios where the 3D points are *not close* to the reference plane, for instance an infinite reference plane. An extended iterative version of our method can also handle scenes with 3D points on or close to the reference plane.

A further contribution is the presentation of a collection of real and virtual reference planes configurations. This includes previously known configurations like a real scene plane and novel configurations like cameras with parallel projection or cameras with known epipolar geometry. Consequently, the reference plane approach is applicable to general scenarios where no real reference plane is visible.

Furthermore, we investigate critical reference plane configurations. These are configurations of multiple cameras, multiple 3D points and a known reference plane which do not have a unique projective reconstruction. The main and novel result is that all non-trivial configurations where points and camera centres are non-coplanar are non-critical. This is an important observation since the scenario of one dominant scene plane visible in multiple views appears frequently in practice and is critical in the general case.

A final contribution is a completely automatic multi-view reconstruction system using the reference plane approach. This includes novel methods for vanishing point detection and robust multi-view point matching using a reference plane.

1.4 Thesis Outline

The reference plane approach divides the reconstruction task into two steps. First, determine a real or virtual reference plane. Second, use the reference plane to reconstruct the 3D features and cameras. The organization of the thesis reflects this partitioning. Chapter 5 investigates the first step. Alternative techniques to determine a real or virtual reference

plane are explored. The second step is analyzed theoretically in chapter 3. The result of this analysis is our direct reference plane method which is outlined chapter 6. This chapter also examines the performance of our method. A brief description of the subsequent chapters is as follows:

2. Basic Concepts of Geometry. This chapter reviews basic concepts of geometry which are necessary for the understanding of the thesis. The important idea of the stratification of 2D and 3D geometry is explained. Moreover, the camera as a Euclidean and projective device is presented. The last section introduces the key concepts of plane+parallax, stabilizing a reference plane, infinite homographies and real or virtual reference planes.

3. Projective Multi-View Geometry: General versus Reference Plane. This chapter is from a theoretical point of view the most important. It compares general and reference plane configurations of points, lines and planes visible in multiple views. The main observation is that the relationship between the 3D features and the cameras is *linear* if a reference plane is known. For general configurations, the relationship is *bi-linear*. This is the key observation for our novel reconstruction method which simultaneously reconstructs *all* cameras and *all* features (off the reference plane) in a *single* linear system. We call this approach the *D*irect *R*eference *P*lane (DRP) approach. Moreover, 3 alternative categories of solving the reconstruction problem are reviewed, (a) camera constraints, (b) structure constraints and (c) factorization. These categories are analyzed for all 3 feature types and both general and reference plane configurations.

4. Structure and Camera Recovery – A Review and Comparison. Whereas the previous chapter presented *theoretical* ways of solving the reconstruction problem, this chapter reviews and compares existing multi-view reconstruction systems which tackle all "real world" problems. For reference plane configurations our method is compared to the camera constraint method of Hartley et al. (2001) and the factorization method of Triggs (2000). It is seen that all three methods have their strengths and weaknesses.

5. Determining a Real or Virtual Reference Plane. The investigation of alternative techniques to determine a real or virtual reference plane is carried out in this chapter. This analysis is important since it shows that the reference plane approach is applicable in many scenarios where no real reference plane is visible. The collection of known and novel reference plane configurations is presented in table 5.1.

6. Structure and Camera Recovery using a Reference Plane. This chapter is from a practical point of view the most important. First, practical algorithms for our novel direct reference plane methods for points, lines and planes are outlined. Secondly, our methods are compared with other approaches under various conditions using real and synthetic data. The experimental study focuses on point features. The main observation is that for difficult, reference plane scenarios our method performs successfully where general reconstruction methods fail. Furthermore, we will demonstrate that our method performs very stably if the 3D points are not close to the finite reference plane. For "flat scenes", where many

3D points are on or close to the finite reference plane, several reference plane methods are compared. However, for this scenario we cannot recommend the best reconstruction method.

7. Critical Reference Plane Configurations. In this chapter we search for configurations of multiple cameras, multiple 3D points and a known reference plane which do not have a unique projective reconstruction. This study is novel since to our knowledge critical configurations have only been examined for the general case. The main result is that all non-trivial configurations where points and camera centres are non-coplanar are non-critical. This is an important observation since scenes with one dominant plane appear frequently in practice and are critical in the general case.

8. An Automatic Multi-View Reconstruction System. A complete automatic multi-view reconstruction system shows the capability of the reference plane approach. Two novel methods are part of this system, automatic vanishing point detection and robust multi-view point matching using a reference plane.

9. Conclusions. Finally, the thesis is summarized and possible avenues of future work are discussed.

1.5 Publications

Most of the thesis is based on the following publications:

- Carsten Rother and Stefan Carlsson, Linear Multi View Reconstruction and Camera Recovery Using a Reference Plane, *International Journal of Computer Vision (IJCV)* 49(2/3):117-141, 2002.
- Carsten Rother, A New Approach to Vanishing Point Detection in Architectural Environments, *Image and Vision Computing (IVC)* 20(9-10):647-656, 2002.
- Carsten Rother, Stefan Carlsson and Dennis Tell, Projective Factorization of Planes and Cameras in Multiple Views, *International Conference* on *Pattern Recognition (ICPR)*, Quebec, Canada, pp. 737-740, 2002.
- Carsten Rother and Stefan Carlsson, Linear Multi View Reconstruction with Missing Data, *European Conference on Computer Vision*, Copenhagen, Denmark, pp. II 309-324, 2002.
- Carsten Rother and Stefan Carlsson, Linear Multi View Reconstruction and Camera Recovery, *International Conference on Computer Vision*, Vancouver, Canada, pp. 42-51, 2001.
- Carsten Rother, A New Approach for Vanishing Point Detection in Architectural Environments, *British Machine Vision Conference (BMVC)*, Bristol, UK, pp. 382-391, 2000.

Chapter 2

Basic Concepts of Geometry

This chapter reviews basic concepts of geometry which are essential for understanding the rest of the thesis. We begin the discussion with n-dimensional projective and affine spaces (sec. 2.1). Since the world is 3-dimensional and its projection onto an image plane is of dimensionality 2, spaces of these dimensionality are considered in more detail. In particular, points, lines and planes in 2D and 3D are analyzed. Mappings between spaces play an important role in geometry, for instance a camera maps the world (3D space) to the image (2D space). These mappings represent a useful tool for classifying geometric spaces into projective, affine, metric and Euclidean space. This leads to the important concept of stratification of 2D and 3D geometry (sec. 2.2). Furthermore, the camera, a "mapping device" between 3D and 2D, is analyzed in terms of its projective and Euclidean properties (sec. 2.3). Finally, the geometry of a reference plane visible in 2 images is considered in more detail (sec. 2.4).

Recently, two books focusing on geometry for computer vision have been published by Hartley and Zisserman (2000) and Faugeras and Luong (2001). The style and notation of this chapter is closely related to (Hartley and Zisserman, 2000). A further good reference is the book of Faugeras (1993). General mathematical textbooks about projective, affine, and Euclidean geometry are (Semple and Kneebone, 1952; Springer, 1964).

This chapter does not contain any novel aspects about geometry for computer vision. Readers which are familiar with this topic might skip this chapter. However, we recommend to read sec. 2.4 which introduces the basic concepts of using a reference plane. Furthermore, the thesis does not present the subject of geometry for computer vision in the "traditional way" as for instance Hartley and Zisserman (2000). We present the abstract n-dimensional projective space before the "simpler" 2D and 3D Euclidean space. The reason is to show that Euclidean geometry is a specialization, subgroup, of the more general concept of projective geometry. Moreover, a thesis is not an introduction for readers which are unfamiliar with this topic.

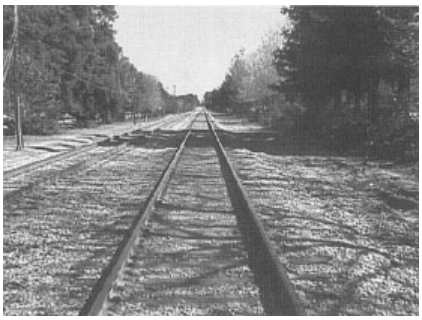


Figure 2.1. The image of a railway line. Courtesy of (Faugeras and Luong, 2001).

2.1 Projective & Affine Spaces

Most people are familiar with the 2D and 3D Euclidean space. A point in 2D can be expressed as a 2-vector (x, y) and in 3D as a 3-vector (x, y, z) with $x, y, z \in \mathbf{R}$. A line is defined by two non-identical points and a plane in 3D by three non-identical points.

Fig. 2.1 shows an image of a railway line. It appears that the two rails intersect in 3D "at infinity". Such a 3D point at infinity cannot be described in the 3D Euclidean space, since the triplet (x, y, z) can only represent points which are not at infinity. If we consider the 2D image as the projection of the 3D Euclidean world, a 3D world point, e.g. the tip of a tree, is projected onto a 2D point in the image. Surprisingly, the projection of this 3D point "at infinity" is a "normal" 2D point in the image as well, i.e. not at infinity on the image plane. Such a 2D point is referred to as the vanishing point of the railway line. This shows that in order to describe mathematically the 3D world and its projection onto the 2D image plane, it would be very useful to have a concept which unifies points at infinity and points not at infinity. However, how can points at infinity be expressed in the 3D Euclidean space? We would have to introduce another Euclidean space which comprised of all points at infinity. How many points at infinity are there? Each pair of parallel lines (railway line) in 3D induces a point at infinity, which encodes the direction of the lines. The space of all possible directions is 2-dimensional. This leads us to the idea of the 3-dimensional projective space: The 3-dimensional projective space is the 3D Euclidean space plus a 2-dimensional Euclidean subspace. More abstractly, the n-dimensional projective space is an extension of the n-dimensional Euclidean space (more precisely affine space) by a (n-1)-dimensional Euclidean (affine) subspace. In the following section this concept is introduced more formally. The basic concepts of affine and projective space are essential for the rest of the thesis. We will begin the discussion with the n-dimensional projective and affine space. However, for the purpose of structure and camera recovery in the 3D world, the 2- and 3-dimensional spaces are of main interest. Furthermore, the three different geometric elements: points, lines and planes are introduced in the projective space $\mathcal{P}^2$ and $\mathcal{P}^3$.

2.1.1 Spaces of n Dimension

A point in the n-dimensional projective space $\mathcal{P}^n$ is a $(n+1)$-dimensional vector $\mathbf{x} = (x_1, \ldots, x_{n+1})^T$ with at least one $x_i \neq 0$. Two points $\mathbf{x}_1$ and $\mathbf{x}_2$ in $\mathcal{P}^n$ are equal if a non-zero scalar λ exists such that $\mathbf{x}_1 = \lambda \mathbf{x}_2$. We will write "projective equality" in a more compact way as: $\mathbf{x}_1 \sim \mathbf{x}_2$. As an example, in the projective plane $\mathcal{P}^2$ the points $(1,0,0)^T$ and $(2,0,0)^T$ are equal, that is $(1,0,0)^T \sim (2,0,0)^T$.

A point is a one-dimensional subspace of the projective space $\mathcal{P}^n$. A $(n-1)$-dimensional subspace of $\mathcal{P}^n$ is called a **hyperplane**. A hyperplane π can be defined as the incidence relation with a point $\mathbf{x}$:

$$\pi^T \mathbf{x} = 0 \ . \tag{2.1}$$

This equation says that the point $\mathbf{x}$ lies on the hyperplane π and vice versa the hyperplane π intersects the point $\mathbf{x}$. Eqn. 2.1 can be written in two different ways: $\pi^T \mathbf{x} = 0$ and $\mathbf{x}^T \pi = 0$. This implies that points and hyperplanes are symmetric in projective space. This symmetry leads to a the basic **Duality Principle**.

Proposition 1 (Duality Principle) *To any theorem in projective space $\mathcal{P}^n$ which includes points and hyperplanes there exists a dual theorem, which may be derived by interchanging the role of points and hyperplanes in the original theorem.*

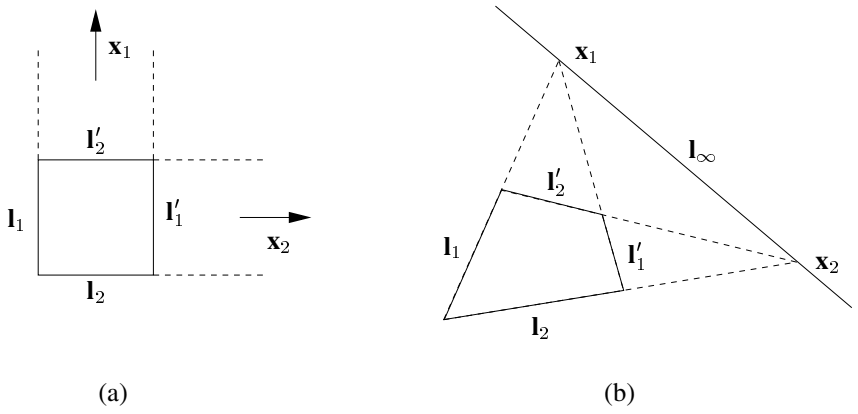
Examples of this principle in the projective space $\mathcal{P}^2$ are given later.

Let us introduce the affine space and its relation to the projective space. A point in the n-dimensional affine space $\mathcal{A}^n$ is an n-dimensional vector $\mathbf{x} = (x_1, \ldots, x_n)^T$. These two different representations of a point in projective and affine space are defined as follows. Writing a point (or hyperplane) with an $(n+1)$-dimensional vector means that a point (or hyperplane) is expressed in **homogeneous coordinates**. The representation of a point as an n-dimensional vector means that a point is expressed in **non-homogeneous coordinates**. Throughout the thesis we use the following notation: Points in non-homogeneous coordinates are denoted with a bar, $\bar{\mathbf{x}}$, and points in homogeneous coordinates without a bar, $\mathbf{x}$. The affine space may be embedded in the projective space by the following mapping:

$$\mathcal{A}^n \rightarrow \mathcal{P}^n : (x_1, \ldots, x_n)^T \rightarrow (x_1, \ldots, x_n, 1)^T \ . \tag{2.2}$$

This mapping is a one-to-one mapping, i.e. injective. However, a certain subspace of $\mathcal{P}^n$ is not part of $\mathcal{A}^n$, therefore the mapping is not surjective. This subspace contains points at "infinity". All points of the form $(x_1, \ldots, x_n, 0)$ are in this subspace. Therefore, a point $(x_1, \ldots, x_n, 0)$ is denoted a **point at infinity**. Furthermore, points at infinity are classified as **infinite points** and points not at infinity as **finite points**. The non-homogeneous vector $(x_1, \ldots, x_n)^T$ of a point at infinity can be regarded as the direction of this point in an affine space. From eqn. 2.1 we see that the subspace of all points at infinity is a hyperplane of the form $(0, \ldots, 0, 1)^T$. This hyperplane is denoted the **plane at infinity** π_∞. Therefore, the affine space $\mathcal{A}^n$ is the projective space $\mathcal{P}^n$ without the plane at infinity π_∞. Note, the embedding of the affine in the projective space was chosen in a "canonical" way. Other ways of embedding are possible, for instance choosing the plane at infinity as $(1, 0, \ldots, 0)^T$.

We are now ready to define the key concept of **parallelism**.



(a) (b)

Figure 2.2. Two squares in the projective space $\mathcal{P}^2$ with parallel lines $\mathbf{l}_1, \mathbf{l}'_1$ and $\mathbf{l}_2, \mathbf{l}'_2$

Definition 1 *Two hyperplanes are parallel if and only if they intersect on the plane at infinity.*

The reader might wonder how the concept of parallelism can be introduced without using (metric) properties such as the distance between points or lines. Fig. 2.2 (a) displays a square on the projective plane $\mathcal{P}^2$. The two pairs of parallel lines $\mathbf{l}_1, \mathbf{l}'_1$ and $\mathbf{l}_2, \mathbf{l}'_2$, which are the hyperplanes in $\mathcal{P}^2$, intersect at infinity. The two corresponding points at infinity $\mathbf{x}_1$ and $\mathbf{x}_2$ represent the directions of the line pairs. Furthermore, these two points define the line at infinity. This looks perfectly all right – from a Euclidean viewpoint. However, fig. 2.2 (b) shows another square on the projective plane $\mathcal{P}^2$. The two pairs of lines $\mathbf{l}_1, \mathbf{l}'_1$ and $\mathbf{l}_2, \mathbf{l}'_2$ intersect at two points $\mathbf{x}_1, \mathbf{x}_2$. These two points define the line at infinity $\mathbf{l}_\infty$. In both cases the projective and affine space is uniquely defined. We can state, parallel lines are "really" parallel, i.e. the distance between the lines is constant (which is a metric concept), if the line at infinity is at its "correct" position. The affine space defined in fig. 2.2 (b) is not the correct affine space, although it is an affine space. Later we will see that specifying a certain hyperplane as the plane at infinity can be very powerful, even if it is not the correct plane at infinity. The task of detecting the correct plane at infinity is revisited in the next section in the context of transformation groups.

Let us now introduce a basis for a projective and affine space. A set of $(n+2)$ points is called a basis of $\mathcal{P}^n$ if and only if any subset of $(n+1)$ points is linearly independent. The **canonical** or **standard basis** of the projective space $\mathcal{P}^n$ is defined as:

$$(\mathbf{e}_1 \cdots \mathbf{e}_{n+2}) = \begin{pmatrix} 1 & & & 1 \\ & \ddots & & \vdots \\ & & 1 & 1 \end{pmatrix} . \tag{2.3}$$

A basis of the affine space $\mathcal{A}^n$ is a set of $(n+1)$ linear independent points. The standard basis of $\mathcal{A}^n$ is defined as

$$(\mathbf{e}_1 \cdots \mathbf{e}_{n+1}) = \begin{pmatrix} 0 & 1 & & \\ \vdots & & \ddots & \\ 0 & & & 1 \end{pmatrix} \quad . \tag{2.4}$$

A hyperplane in $\mathcal{P}^n$ is defined as the join of n points. Dual to this, a point in $\mathcal{P}^n$ is defined as the intersection of n hyperplanes. In both cases we assume that points and hyperplanes are in general pose, e.g. n points form a basis of the hyperplane. These join and intersection relationships can be derived compactly from the homogeneous coordinates of points and hyperplanes respectively. Before doing this the following two lemmas are needed.

Lemma 1 *The $(n+1)$ points $\boldsymbol{x}_1, \ldots \boldsymbol{x}_{n+1}$ in projective space $\mathcal{P}^n$ are on a hyperplane if and only if the determinant $|\boldsymbol{x}_1 \cdots \boldsymbol{x}_{n+1}|$ is zero.*

Proof The condition that a point $\mathbf{x}_i$ is on a certain hyperplane π is $\mathbf{x}_i^T \ \pi = 0$. Stack all these linear condition into a system

$$\begin{pmatrix} \mathbf{x}_1^T \\ \vdots \\ \mathbf{x}_{n+1}^T \end{pmatrix} \pi = \mathbf{0} \ .$$

To obtain a unique solution for the hyperplane π, the nullspace of the linear system has to be 1-dimensional. This means that the determinant $|\mathbf{x}_1 \cdots \mathbf{x}_{n+1}|$ has to be zero. This concludes both directions of the lemma.

□

Using the duality principle 1, the dual lemma to lemma 1 is the following.

Lemma 2 *The $(n+1)$ hyperplanes $\pi_1, \ldots \pi_{n+1}$ in projective space $\mathcal{P}^n$ intersect at one point if and only if the determinant $|\pi_1 \cdots \pi_{n+1}|$ is zero.*

Proof This proof is dual to the proof of lemma 1 by interchanging the role of points and hyperplanes.

□

Let us define the generalized cross(vector-) product $\times$ for the projective space $\mathcal{P}^n$ which maps n homogeneous vectors $\mathbf{x}_1, \ldots \mathbf{x}_n$ onto a single vector $\mathbf{x}$. This crossproduct can be defined informally as

$$\times(\mathbf{x}_1, \ldots, \mathbf{x}_n) \quad = \quad \begin{vmatrix} \mathbf{e}_1 & \cdots & \mathbf{e}_n \\ & \mathbf{x}_1^T & \\ & \vdots & \\ & \mathbf{x}_n^T & \end{vmatrix}$$

and more formally

$$\times(\mathbf{x}_1,\ldots,\mathbf{x}_n) = \mathbf{e}_1 \begin{vmatrix} x_{1,2} & \cdots & x_{1,n+1} \\ & \vdots & \\ x_{n,2} & \cdots & x_{n,n+1} \end{vmatrix} - \mathbf{e}_2 \begin{vmatrix} x_{1,1} & x_{1,3} & \cdots & x_{1,n+1} \\ & & \vdots & \\ x_{n,1} & x_{n,3} & \cdots & x_{n,n+1} \end{vmatrix} + \cdots (-1)^{n+1} \mathbf{e}_n \begin{vmatrix} x_{1,1} & \cdots & x_{1,n} \\ & \vdots & \\ x_{n,1} & \cdots & x_{n,n} \end{vmatrix}, \tag{2.5}$$

where $\mathbf{e}_1,\ldots\mathbf{e}_n$ represent the first n vectors of the standard projective basis. In the projective plane P^2 it is $\times(\mathbf{x}_1,\mathbf{x}_2) = \mathbf{x}_1 \times \mathbf{x}_2$.

Using this crossproduct, the join of n points, which define a hyperplane, can be expressed as follows.

Theorem 1 *The hyperplane π defined by the n points $\boldsymbol{x}_1,\ldots\boldsymbol{x}_n$ is $\pi = \times(\boldsymbol{x}_1,\ldots\boldsymbol{x}_n)$.*

Proof Choose $n+1$ points $\mathbf{x}_1,\ldots\mathbf{x}_{n+1}$ which lie on a hyperplane π. According to lemma 1 $|\mathbf{x}_1\cdots\mathbf{x}_{n+1}| = 0$. Using eqn. 2.5 this is equivalent to $\mathbf{x}_{n+1}^T \times (\mathbf{x}_1,\ldots\mathbf{x}_n) = 0$. Since the point $\mathbf{x}_{n+1}$ can be any point on π, the hyperplane π is defined by the the points $\mathbf{x}_1,\ldots\mathbf{x}_n$ as $\pi = \times(\mathbf{x}_1,\ldots\mathbf{x}_n)$.

□

The dual theorem to theorem 1 is the following.

Theorem 2 *The point $\boldsymbol{x}$ defined by the intersection of n hyperplanes $\pi_1,\ldots\pi_n$ is $\boldsymbol{x} = \times(\pi_1,\ldots\pi_n)$.*

Proof This proof is dual to the proof of theorem 1 by interchanging the role of points and hyperplanes.

□

Let us consider the transformation between projective spaces of the same dimension in more detail.

Definition 2 *An invertible mapping from $\mathcal{P}^n \rightarrow \mathcal{P}^n$ is called a **projectivity**, **projective transformation** or **collineation** if and only if collinear points are mapped onto collinear points.*

Such a mapping can be algebraically expressed as a *non-singular* matrix $H \in \mathbf{R}^{n+1\times n+1}$ as: $\mathbf{x} \rightarrow \mathbf{x}' \sim H\,\mathbf{x}$, where $\mathbf{x},\mathbf{x}'$ are points in $\mathcal{P}^n$. Therefore, H defines a *linear* mapping of points in homogeneous coordinates. All possible projective transformations form a group called the **general linear group** of projective transformations. In the next section this group and important subgroups are discussed in more detailed. Since H is *non-singular*, it is bijective and we obtain $\mathbf{x}$ from $\mathbf{x}'$ as: $\mathbf{x} \sim H^{-1}\,\mathbf{x}'$. Note, since points are only defined up to scale, projectivities are also only unique up to scale. This can be seen from: $\mathbf{x}' \sim \lambda H\,\mathbf{x} \sim H\lambda\,\mathbf{x} \sim H\,\mathbf{x}$.

According to the duality principle 1 there is a theorem for the transformation of hyperplanes.

Proposition 2 *If points in $\mathcal{P}^n$ transform with $\boldsymbol{x}' \sim H\boldsymbol{x}$ then hyperplanes transform as $\pi' \sim H^{-T}\pi$.*

Proof All points $\mathbf{x}$ which lie on a specific hyperplane π are defined by $\pi^T\ \mathbf{x} = 0$. This is equivalent to $\pi^T\ H^{-1}\ H\ \mathbf{x} = 0$ and $(H^{-T}\pi)^T\ \mathbf{x}' = 0$. Therefore, the hyperplane π is mapped to π' by $\pi' \sim H^{-T}\pi$.

□

The following theorem is a classical theorem of projective geometry and proved in (Semple and Kneebone, 1952).

Theorem 3 *Any projective basis in $\mathcal{P}^n$ can be transformed by a unique projective transformation into the standard basis.*

This transformation T can be written explicitly for a projective basis $\mathbf{x}_1, \ldots \mathbf{x}_{n+2}$ as

$$T = (\lambda_1\mathbf{x}_1 \cdots \lambda_{n+1}\mathbf{x}_{n+1}) \text{ where } (\lambda_1, \ldots \lambda_{n+1}) = (\mathbf{x}_1 \cdots \mathbf{x}_{n+1})^{-1}\ \mathbf{x}_{n+2}\ . \tag{2.6}$$

Furthermore, any two bases in $\mathcal{P}^n$ define a unique projective transformation.

We will see later, that singular transformation matrices may occur.

Theorem 4 *If the mapping H: $\boldsymbol{x} \in \mathcal{P}^n \rightarrow \boldsymbol{x}' \in \mathcal{P}^n$ is singular than all points $\boldsymbol{x}'$ lie on a hyperplane in $\mathcal{P}^n$.*

Proof W.l.o.g the standard projective basis is mapped with H onto the points $\mathbf{x}_1, \ldots \mathbf{x}_{n+2}$. According to eqn. 2.6, the mapping H is of the form: $H = (\lambda_1\mathbf{x}_1 \cdots \lambda_{n+1}\mathbf{x}_{n+1})$. Since H is singular the determinant $|\lambda_1\mathbf{x}_1 \cdots \lambda_{n+1}\mathbf{x}_{n+1}|$ is zero. Using lemma 1 we can conclude that the points $\mathbf{x}_1 \ldots \mathbf{x}_{n+1}$ lie on a hyperplane in $\mathcal{P}^n$. The point $\mathbf{x}_{n+2}$ can be expressed as: $\mathbf{x}_{n+2} = \lambda_1\mathbf{x}_1 + \cdots + \lambda_{n+1}\mathbf{x}_{n+1}$. Therefore, the determinant $|\lambda_1\mathbf{x}_1 \cdots \lambda_{n+1}\mathbf{x}_{n+1}|$ can be written as well as $|\lambda_2\mathbf{x}_2 \cdots \lambda_{n+1}\mathbf{x}_{n+1}\ \mathbf{x}_{n+2}|$. This means, according to lemma 1, that the point $\mathbf{x}_{n+2}$ lies in the same hyperplane as $\mathbf{x}_1 \ldots \mathbf{x}_{n+1}$.

□

A singular matrix H is not injective, i.e. two different points $\mathbf{x}_1 \not\sim \mathbf{x}_2$ can be mapped to the same point $H\mathbf{x}_1 \sim H\mathbf{x}_2$. This means that such a mapping is no longer a collineation. In the projective plane $\mathcal{P}^2$, the standard basis (non-collinear points) are mapped onto a line (collinear points).

A mapping from $\mathcal{P}^n \rightarrow \mathcal{P}^m$ where $n > m$ is called a **projection**. As in the case of projectivities this mapping can be expressed algebraically by a matrix H of size $(m+1) \times (n+1)$ as: $\mathbf{x} \rightarrow \mathbf{x}' \sim H\ \mathbf{x}$ where $\mathbf{x} \in \mathcal{P}^n$ and $\mathbf{x}' \in \mathcal{P}^m$. An example of a projection is a "projective" camera which maps $\mathcal{P}^3 \rightarrow \mathcal{P}^2$. It will be the subject of discussion in a later chapter. Similar to theorem 4 for projectivities, we can state the following theorem.

Theorem 5 *If the mapping H: $\boldsymbol{x} \in \mathcal{P}^n \rightarrow \boldsymbol{x}' \in \mathcal{P}^m$ $(m > n)$ is of rank less than $m+1$, then all points $\boldsymbol{x}'$ lie on a hyperplane in $\mathcal{P}^m$.*

Proof W.l.o.g the standard projective basis is mapped with H onto the points $\mathbf{x}_1, \ldots \mathbf{x}_{n+2} \in \mathcal{P}^m$. According to eqn. 2.6, the mapping H is of the form:

$H = (\lambda_1 \mathbf{x}_1 \cdots \lambda_{n+1}\mathbf{x}_{n+1})$. Since the rank of H is less than $m+1$ all $\binom{m+1}{n+1}$ subdeterminants have to be zero. According to lemma 1 this means that the points $\mathbf{x}_1 \ldots \mathbf{x}_{n+1}$ lie on a hyperplane in $\mathcal{P}^m$. As in the proof of theorem 4 we may conclude that the point $\mathbf{x}_{n+2}$ lies on this hyperplane as well.

□

2.1.2 The Projective Spaces $\mathcal{P}^2$ and $\mathcal{P}^3$

Fist consider the projective plane $\mathcal{P}^2$. A point $\mathbf{x}$ and a line $\mathbf{l}$ are define as homogeneous 3-vectors: $\mathbf{x} = (x, y, w)^T$, $\mathbf{l} = (l_1, l_2, l_3)^T$. A line is a hyperplane of $\mathcal{P}^2$ and therefore dual to a point. A point $\mathbf{x}$ lies on the line $\mathbf{l}$ if $\mathbf{l}^T\ \mathbf{x} = 0$ or $\mathbf{x}^T\ \mathbf{l} = 0$. Obviously, all theorems developed in the previous section for n-dimensional projective spaces hold for the 2-dimensional projective plane. As an example, two points at infinity $\mathbf{x}_1 = (1, 0, 0)^T$ and $\mathbf{x}_2 = (0, 1, 0)^T$ define the line at infinity $\mathbf{l}_\infty$. According to theorem 1 it is:

$$\mathbf{l}_\infty = \mathbf{x_1} \times \mathbf{x_2} = \left(\begin{vmatrix} 0 & 0 \\ 1 & 0 \end{vmatrix}, - \begin{vmatrix} 1 & 0 \\ 0 & 0 \end{vmatrix}, \begin{vmatrix} 1 & 0 \\ 0 & 1 \end{vmatrix} \right)^T = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad . \tag{2.7}$$

In the projective space $\mathcal{P}^3$ points and planes are dual. They can be written in homogeneous coordinates as $\mathbf{X} = (X, Y, Z, W)^T$ and $\mathbf{\Pi} = (\Pi_1, \Pi_2, \Pi_3, \Pi_4)^T$. Throughout the thesis, geometric elements of the projective space $\mathcal{P}^2$ are denoted by lower case letters, e.g. a point $\mathbf{x}$, and elements of the projective space $\mathcal{P}^3$ by upper case letters, e.g. a point $\mathbf{X}$.

The representation of a line in $\mathcal{P}^3$ is more complex. A line in $\mathcal{P}^3$ has 4 degrees of freedom. The reader might think of 2 arbitrary 2D points located onto 2 distinct planes, which are not identical. A minimal representation of a 3D line is, however, not compact enough for later use. The simplest representation of a line $\mathbf{L}$ requires two distinct non-homogeneous points $\bar{\mathbf{X}}_1, \bar{\mathbf{X}}_2$:

$$\mathbf{L} : \ \bar{\mathbf{X}}_1 + \lambda(\bar{\mathbf{X}}_2 - \bar{\mathbf{X}}_1) \ , \ \ \lambda \in \mathbf{R} \ . \tag{2.8}$$

Each value of λ gives a point $\bar{\mathbf{X}}$ on the line. However, this representation is only useful in an affine space $\mathcal{A}^3$. Therefore, several different line representations have been suggested in $\mathcal{P}^3$ (Hartley and Zisserman, 2000). Let us define a 3D line as a 4×4 matrix L of two homogeneous points $\mathbf{X}_1, \mathbf{X}_2$:

$$L = \mathbf{X}_1\mathbf{X}_2^T \ - \ \mathbf{X}_2\mathbf{X}_1^T \ . \tag{2.9}$$

This is called the **Plücker matrix** of a line. Eqn. 2.9 shows that a scaling of $\mathbf{X}_1$ and $\mathbf{X}_2$ results in a scaling of L, i.e. $\lambda\mathbf{X}_1$ and $\mu\mathbf{X}_2$ have a Plücker matrix $\lambda\mu L$. Therefore, L is a homogeneous representation of a line. The Plücker matrix L is skew symmetric, i.e. $L_{ij} = -L_{ji}$, with zeros on its diagonal, i.e. $L_{ii} = 0$. Therefore, L is described by 6 numbers. Since a line has only 4 degrees of freedom, the Plücker matrix has to satisfy one extra constraint, which is that its determinant has to be zero. These 6 elements form

a homogeneous vector $\mathbf{L}$ called the **Plücker line coordinates**. Explicitly, the Plücker line coordinates are

$$\mathbf{L} = (L_{12}, L_{13}, L_{14}, L_{23}, L_{42}, L_{34})^T \ . \tag{2.10}$$

A **dual Plücker matrix** representation L^* is based on 2 distinct planes $\mathbf{\Pi}_1, \mathbf{\Pi}_2$ and defined in a similar way to 2.9:

$$L^* = \mathbf{\Pi}_1 \mathbf{\Pi}_2^T \ - \ \mathbf{\Pi}_2 \mathbf{\Pi}_1^T \ . \tag{2.11}$$

The two different Plücker matrix representations L, L^* are related by the substitution:

$$(L_{12}, L_{13}, L_{14}, L_{23}, L_{42}, L_{34}) \ \leftrightarrow \ (L^*_{34}, L^*_{42}, L^*_{23}, L^*_{14}, L^*_{13}, L^*_{12}) \ . \tag{2.12}$$

On the basis of these representations, joint and incident relationships of a point $\mathbf{X}$, a line $\mathbf{L}$ and a plane $\mathbf{\Pi}$ can be written compactly as

$$\mathbf{\Pi} \sim L^* X \quad \text{and} \quad \mathbf{X} \sim L \, \mathbf{\Pi} \ . \tag{2.13}$$

Consider the transformation of a 3D point $\mathbf{X}$: $\mathbf{X}' \sim H\mathbf{X}$. In this case the Plücker matrix L transforms to $L' \sim HLH^T$ and the dual Plücker matrix $L^{*'}$ to $L^{*'} \sim H^{-T} L H^{-1}$.

2.2 Stratification of 2D and 3D Geometry

The previous section introduced two geometric spaces, the affine and projective space. A different way to define a geometric space is by relating it to a group of transformations of this space. Such a transformation group can be characterized by certain invariants. Historically, Klein (1893) declared in the "Erlanger Programme" that geometry should be considered as the study of invariance of transformation groups. An **invariant** of a transformation group is a property of geometric elements (or a geometric element itself) which remains unchanged under any transformation of this group. The most general transformation group is the general linear group, which is composed of all projectivities of $\mathcal{P}^n$ (see definition 2). It is also denoted the **projective group**. We have already seen one invariant property of this group: collineation of points. In general there are three other transformation groups: **Affine group, similarity group** and **Euclidean group**. Each group defines a space: **Affine space** (by the affine group), **metric space** (by the similarity group) and **Euclidean space** (by the Euclidean group). An informal description of these groups is: The Euclidean group allows rigid transformation, the similarity group allows scaling additionally, and the affine group preserves parallelism. These three groups, together with the projective group, form a hierarchy of subgroups:

$$Euclidean \subset similarity \subset affine \subset projective \ .$$

This means that invariants of a certain group are necessarily invariants of a subgroup. This hierarchy of transformation groups reflects a hierarchy of the corresponding spaces which is called the **stratification of geometry**.

Why is this concept of stratified geometry important for the problem of 3D reconstruction? The reconstruction task is to create a Euclidean (or at least metric) reconstruction

of the 3D Euclidean world from 2D (Euclidean) images of the world. In general, nothing prevents us from using a more general concept than Euclidean geometry, e.g. projective geometry, to achieve this goal. The central idea is to treat the the camera as a projective device, a projection from $\mathcal{P}^2$ to $\mathcal{P}^3$, not as an Euclidean device, i.e. consider explicitly camera properties like the focal length. This allows 3D reconstruction from multiple images in the projective space $\mathcal{P}^3$ without any knowledge about the camera's calibration[1]. This reconstruction task is known as **uncalibrated structure and camera recovery**. Since humans perceive the 3D world as being Euclidean, the reconstruction has to be upgraded from the projective space to at least the metric space. This process is equivalent to the task of calibrating the camera. The traditionally approach to reconstruction is the *reverse* of this order. order: First calibrate the camera and then recover structure and cameras. The traditional way is called **calibrated structure and camera recovery**. However, there is a significant advantage of uncalibrated reconstruction in contrast to calibrated reconstruction. The traditional way of calibration is to use a *single* view of a calibration object with known properties, usually a calibration grid. However, in the uncalibrated case information derived from *multiple* views can be used. It has been demonstrated (Faugeras et al., 1992) that a projective reconstruction from multiple views of a camera with constant internal parameters is sufficient to calibrate the camera and upgrade the reconstruction to the metric space. This approach to camera calibration is called **auto-** or **self-calibration**. It has been an active field of research and the interested reader is referred to (Pollefeys et al., 1998; Triggs, 1997a; Faugeras et al., 1992) and for an overview (Faugeras and Luong, 2001; Hartley and Zisserman, 2000; Pollefeys, 1999).

The task of camera calibration does not play an essential role in this thesis. Therefore, only the basic concepts of stratified geometry are presented here. Note, for some specific tasks, e.g. new view synthesis (generating new views from a reconstruction), it is not necessary to upgrade the projective reconstruction to the metric space. The concept of stratified geometry was introduced into the field of computer vision by Faugeras (1995).

In the following, we consider the transformation groups for 2- and 3-dimensional spaces. A summary of the four transformation groups together with their transformation matrices and main invariant properties are given in table 2.1. Let us begin with the most general group, the projective group.

2.2.1 Projective Group

The projective transformation in $\mathcal{P}^2$ is also known as a **homography**. It can be represented by a matrix H of size 3×3 and maps a point $\mathbf{x}$ onto $\mathbf{x}'$ by:

$$\mathbf{x}' \sim H\mathbf{x} \quad \text{or} \quad \begin{pmatrix} x' \\ y' \\ w' \end{pmatrix} \sim \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ w \end{pmatrix}. \tag{2.14}$$

It plays an essential role in this thesis. In the 3D world three different types of planes can be identified: a "real" plane in the scene, the plane at infinity and the image plane. In

[1] The formal definition of calibration will be given later.

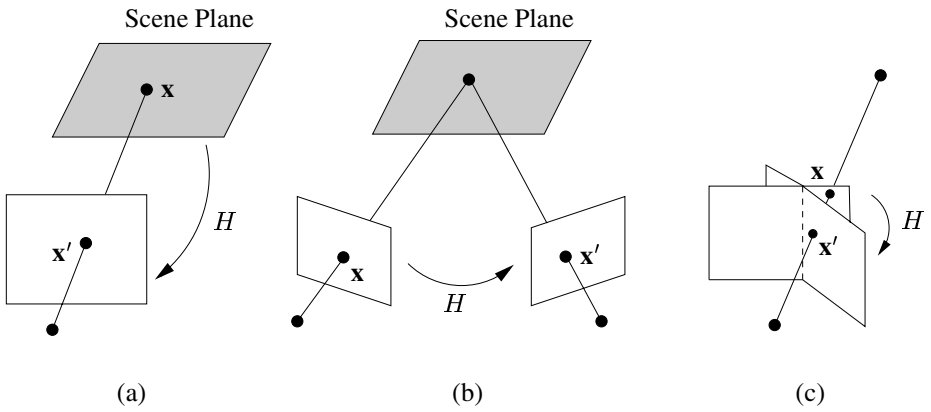


Figure 2.3. Different types of homographies between a scene plane and an image (a), two images via a scene plane (b), and two images of a rotating camera (c).

fig. 2.3 three different types of homographies are shown. The homography in (a) relates a scene plane to an image plane, the homography in (b) relates two images via a scene plane and the homography in (c) relates two image planes of a rotating camera, i.e. via the plane at infinity. The algebraic derivation of these homographies is discussed in a later section. A homography has 8 degrees of freedom (dof), since it has 9 elements and the overall scale is undetermined.

The projective transformation in $\mathcal{P}^3$ is represented by a matrix of size 4×4. It has 15 degrees of freedom. Probably the most interesting invariant of the projective group is the **cross-ratio**. Let $dis(\mathbf{x}_1, \mathbf{x}_2)$ define the distance between two points $\mathbf{x}_1$ and $\mathbf{x}_2$. The cross ratio is defined as the ratio of ratios of four collinear points $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ and $\mathbf{x}_4$:

$$\frac{dis(\mathbf{x}_1, \mathbf{x}_2)}{dis(\mathbf{x}_1, \mathbf{x}_4)} : \frac{dis(\mathbf{x}_3, \mathbf{x}_2)}{dis(\mathbf{x}_3, \mathbf{x}_4)} \quad . \tag{2.15}$$

More invariants are summarized in table 2.1.

2.2.2 Affine Group

The affine space differs from the projective space by not including the plane at infinity. The affine group is the set of all projective transformations which leaves the plane at infinity in its canonical position, i.e. $(0, 0, 0, 1)^T$ in $\mathcal{P}^3$. The affine transformation $T : \mathbf{x} \rightarrow \mathbf{x}'$ for the affine space $\mathcal{A}^n$ can be written in homogeneous coordinates $\mathbf{x}, \mathbf{x}'$ and non-homogeneous coordinates $\bar{\mathbf{x}}, \bar{\mathbf{x}}'$ as:

$$\mathbf{x}' \sim \begin{pmatrix} A & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix} \mathbf{x} \quad \text{and} \quad \bar{\mathbf{x}}' = A\,\bar{\mathbf{x}} + \mathbf{t} \ , \tag{2.16}$$

where A is a general, non-singular matrix of size $n \times n$, $\mathbf{t}$ a general vector of size n and $\mathbf{0}$ the null vector of size n. It is easy to verify that this T does not move the plane at infinity

π_∞, since $T^{-T}\pi_\infty \sim \pi_\infty$. Note, T can change the position of points on π_∞but does not move them out of this plane. Therefore, parallelism is an invariant property of affine transformations. Further invariants are summarized in table 2.1. The number of degrees of freedom of T is 12 for $\mathcal{A}^3$ and 6 for $\mathcal{A}^2$.

The result of an uncalibrated reconstruction process is a projective reconstruction in $\mathcal{P}^3$. In order to upgrade it to an affine reconstruction, the **correct plane at infinity** has to be detected. This can be done by exploiting the information of affine (or metric, Euclidean) invariants in the scene. For example, if we know that a pair of 3D reconstructed lines in $\mathcal{P}^3$ is parallel in reality, their intersection point has to lie on the *correct* plane at infinity. Three such points would uniquely identify the correct plane at infinity. If these 3 points define a finite plane π, the transformation

$$T: \quad \mathbf{x}' \sim \begin{pmatrix} I & \mathbf{0} \\ \pi^T & \end{pmatrix} \mathbf{x} \ , \tag{2.17}$$

where I is the identity matrix, moves π to its canonical position, π_∞, since $T^{-T}\pi \sim \pi_\infty$.

2.2.3 Similarity Group

The group of similarity transformations allows the rotation, transformation and scaling of geometric elements in the metric space $\mathcal{M}$. A transformation of this group can be written in homogeneous coordinates $\mathbf{x}, \mathbf{x}'$ and non-homogeneous coordinates $\bar{\mathbf{x}}, \bar{\mathbf{x}}'$ as:

$$\mathbf{x}' \sim \begin{pmatrix} \lambda R & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix} \mathbf{x} \quad \text{and} \quad \bar{\mathbf{x}}' = \lambda R\, \bar{\mathbf{x}} + \mathbf{t} \ , \tag{2.18}$$

where R is a rotation matrix of size 2×2 in $\mathcal{M}^2$ or 3×3 in $\mathcal{M}^3$. For a rotation matrix $| R | = 1$ and $R^T = R^{-1}$. Therefore, $R\, R^T = R^T R = I$. The metric space $\mathcal{M}^2$ has 4 degrees of freedom and $\mathcal{M}^3$ 7 degrees of freedom.

For the purpose of visualizing a 3D reconstruction, the metric space is sufficient. Humans are naturally familiar with the similarity group, since the 3D world is Euclidean and the additional scaling of an object is equivalent to seeing an object from different distances. For instance, the picture of a real building from a far distance and a toyhouse from a close distance have the same size on the human retina.

In order to upgrade an affine reconstruction to a metric reconstruction, metric (or Euclidean) invariants have to be known or determined. Invariant property which may be identified for some scenes are angles and ratio of distances. As invariant elements, the circular points in $\mathcal{M}^2$ and the absolute conic in $\mathcal{M}^3$ can be identified. These are important entities for the task of auto- or self-calibration, which are, however, not introduced here (see Pollefeys, 1999).

Group	Dof 2D/3D	Matrix 2D(3D)	Invariants 2D/3D
Projective	8 / 15	$H_{3\times3(4\times4)}$	Cross-ratio, collinearity, concurrency, intersection
Affine	6 / 12	$\begin{pmatrix} A & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix}$	Parallelism, centre of mass, line/plane at infinity, ratio of areas
Metric	4 / 7	$\begin{pmatrix} \lambda R & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix}$	Angles, ratio of lengths, circular points/absolute conic
Euclidean	3 / 6	$\begin{pmatrix} R & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix}$	Length, area

Table 2.1. The hierarchy of transformation groups for 2D and 3D, where A is a general, non-singular matrix of size 2×2 or 3×3, $\mathbf{t}$ is a general vector of size 2 or 3, $\mathbf{0}$ is the null-vector of size 2 or 3 and R is a 2D or 3D rotation matrix with $\mid R \mid = 1$.

2.2.4 Euclidean Group

The Euclidean group differs from the similarity group by fixing the scale factor λ. A transformation of this group is

$$\mathbf{x}' \sim \begin{pmatrix} R & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix} \mathbf{x} \quad \text{or} \quad \bar{\mathbf{x}}' = R\,\bar{\mathbf{x}} + \mathbf{t} \ . \tag{2.19}$$

In order to determine the unknown scale factor, i.e. to upgrade the metric reconstruction to Euclidean, an absolute distance in the scene has to be known.

Table 2.1 summarizes the four transformation groups together with their transformation matrices, degrees of freedom and main invariants.

2.3 Camera Geometry

The following treatment of the geometry of cameras in the Euclidean and projective space is closely related to the book of Hartley and Zisserman (2000). However, only some important properties are discussed. The interesting reader is referred to (Hartley and Zisserman, 2000; Faugeras and Luong, 2001; Faugeras, 1993) for a more exhaustive study of this subject.

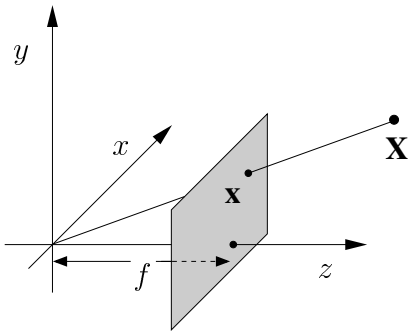


Figure 2.4. The pinhole camera is an idealization of a real world camera.

2.3.1 Cameras in Euclidean Space

A common simplification of a real world camera is to consider it as a central projection device of the Euclidean space $\mathcal{E}^3$ onto the image plane $\mathcal{E}^2$. Advantageously such a projection is a *linear* mapping of homogeneous coordinates as seen in the previous chapter. Such a simplified camera is called a **pinhole camera**. If this simplification is not sufficient, as in the case of wide-angle cameras, *non-linear* mappings, e.g. radial distortion, have to be incorporated in the camera model (e.g. Devernay and Faugeras, 1995).

Consider a special pinhole camera with its centre of projection at the origin of the Euclidean space (see fig. 2.4). The mapping of a scene point $\mathbf{X} = (X, Y, Z, 1)^T$ onto a point $\mathbf{x} = (x, y, 1)^T$ on the image plane can be written as

$$x = f\,\frac{X}{Z} \quad \text{and} \quad y = f\,\frac{Y}{Z} \tag{2.20}$$

or in homogeneous coordinates as a linear mapping

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \sim \begin{pmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{pmatrix} (I_{3\times 3} \mid \mathbf{0}_{3\times 1}) \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} . \tag{2.21}$$

The centre of projection is denoted the **camera centre** $\bar{\mathbf{Q}}$ and the scalar f, which is the distance between the camera centre and the image plane, is called the **focal length** of the camera. The line perpendicular to the image plane and passing through the camera centre represents the **optical axis** of the camera. The intersection of the optical axis with the image plane is the **principle point** $\mathbf{x}_0$ of the camera. Furthermore, the plane parallel to the image plane through the camera centre is called the **principle plane**. In fig. 2.4, the optical axis coincide with the z-axis, the principle point is $\bar{\mathbf{x}}_0 = (0, 0)^T$ and the principle plane is the plane $z = 0$.

A unit of the Euclidean image plane can be considered as a pixel. This is especially useful for CCD cameras. A more detailed inspection of a CCD camera reveals that a pixel

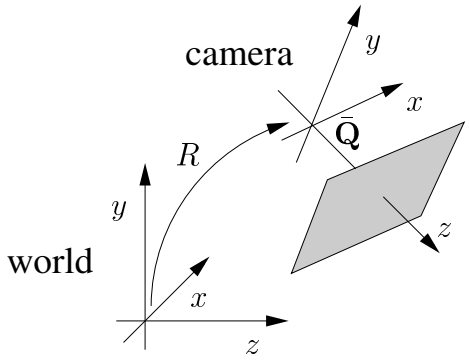


Figure 2.5. A pinhole camera which is rotated and translated with respect to a world coordinate system.

is not necessarily square. We introduce the **aspect ratio** r and the **skew** s of a pixel as additional parameters of the camera model.

Such a camera with unknown focal length, principle point, aspect ratio and skew can be formulated as a linear mapping of Euclidean spaces:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \sim \begin{pmatrix} f & s & x_0 \\ 0 & r\,f & y_0 \\ 0 & 0 & 1 \end{pmatrix} (I_{3\times 3} \mid \mathbf{0}_{3\times 1}) \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{2.22}$$

or compactly

$$\mathbf{x} \sim K\,(I_{3\times 3} \mid \mathbf{0}_{3\times 1})\,\mathbf{X} \ . \tag{2.23}$$

The matrix K is called the **calibration matrix** and its five, non-zero parameters the **intrinsic camera parameters**. The process of determining the calibration matrix is called **camera calibration** and its result is a **calibrated camera**. If the matrix K is unknown the camera is denoted an **uncalibrated camera**.

For most real world cameras, four out of five intrinsic camera parameters are predictable. The aspect ratio is close to one, the skew close to zero and the principle point close to the centre of the image. In contrast to this, the focal length might vary largely depending on the aperture angle of the lens. Additionally, if the focal length is defined in terms of number of pixels, its size depends on the resolution of the CCD camera. Furthermore, for zoom-cameras the focal length might change considerably during the process of capturing a scene.

As previously mentioned, camera calibration is not an essential topic in the thesis. The only calibration procedure explained in more detail in sec 5.1.2 is based on three vanishing points of orthogonal directions (Caprile and Torre, 1990). In this case a special "square pixel" camera is used, which has aspect ratio one and skew zero.

So far, we have assumed that the coordinate system of the camera (ccs) is aligned with the world coordinate system (wcs) in a special way (see fig. 2.4). In general, the camera coordinate system is rotated and translated with respect to the world coordinate system (see fig. 2.5). A point in the world coordinate system $\mathbf{X}_{wcs}$ is transformed to a point in the camera coordinate $\mathbf{X}_{ccs}$ as

$$\bar{\mathbf{X}}_{ccs} = R\,(\,\bar{\mathbf{X}}_{wcs} \;-\; \bar{\mathbf{Q}}\,) \quad \text{or} \quad \mathbf{X}_{ccs} \;\sim\; \begin{pmatrix} R & -R\bar{\mathbf{Q}} \\ \mathbf{0} & 1 \end{pmatrix} \mathbf{X}_{wcs} \;, \tag{2.24}$$

where $\bar{\mathbf{Q}}$ is the camera's centre and R the camera's rotation. The rotation matrix R can be written as $R = (x_w \mid y_w \mid z_w)$ where x_w, y_w, z_w is the $x-, y-, z-$axis of the world coordinate system. The 3 parameters of R and the 3 parameters of $\bar{\mathbf{Q}}$ are called the **extrinsic camera parameters**. Combining eqn. 2.23 and 2.24, the linear mapping of a camera is defined as

$$\mathbf{x} \;\sim\; K\,R\,(I \mid -\bar{\mathbf{Q}}\,)\,\mathbf{X} \;\sim\; P\,\mathbf{X}\,. \tag{2.25}$$

The 3×4 matrix P is called the **camera projection matrix**. Since P is only unique up to scale, it has 11 degrees of freedom. This matches with the complete number of unknown camera parameters which is 11 (5 intrinsic and 6 extrinsic).

It has been shown (e.g. Hartley and Zisserman, 2000) that all intrinsic and extrinsic camera parameters can be identified uniquely from a given camera matrix P. First, the camera centre $\bar{\mathbf{Q}}$ is identified. Secondly, a QR-decomposition of the matrix KR with the assumption that $f > 0$ provides the remaining camera parameters.

2.3.2 Cameras in Projective Space

The stratified approach to structure and camera recovery interprets the camera as a "projective device" rather than a Euclidean device. A pinhole camera can then be expressed most generally as a projection from the projective space $\mathcal{P}^3$ to $\mathcal{P}^2$. Such a camera is called a **projective camera**. The 3×4 projection matrix P of a projective camera may be written as

$$P \;\sim\; H_{3\times 3}\,(I \mid \mathbf{0}\,)\,H_{4\times 4} \;, \tag{2.26}$$

where $H_{3\times 3}$ and $H_{4\times 4}$ represent arbitrary projective transformations. The free choice of the projective basis on the 2D image plane and in the 3D world is described by the projective transformations $H_{3\times 3}$ and $H_{4\times 4}$ respectively.

Eqn. 2.26 forces P to have rank 3. If the rank of P is less than 3, all points in $\mathcal{P}^3$ are mapped onto a line in the image. This is a conclusion of theorem 5. Let us write the camera matrix as $P = (H \mid \mathbf{t})$. Since P must have rank 3, the matrix H has either rank 2 or 3. We will see that these two types of cameras, where H is either non-singular or singular, have different properties.

Let us investigate the camera centre of a general projective camera P. It has been shown (e.g. Hartley and Zisserman, 2000), that the camera centre $\mathbf{Q}$ is defined by $P\,\mathbf{Q} = \mathbf{0}$. If H is non-singular we may write $P \;\sim\; H\,(I \mid H^{-1}\,\mathbf{t})$. This means that the camera centre is $\mathbf{Q} = (-H^{-1}\,\mathbf{t}, 1)^T$ since $P\,\mathbf{Q} = H\,(\bar{\mathbf{Q}} \;-\; \bar{\mathbf{Q}}) = \mathbf{0}$. Therefore, we may write

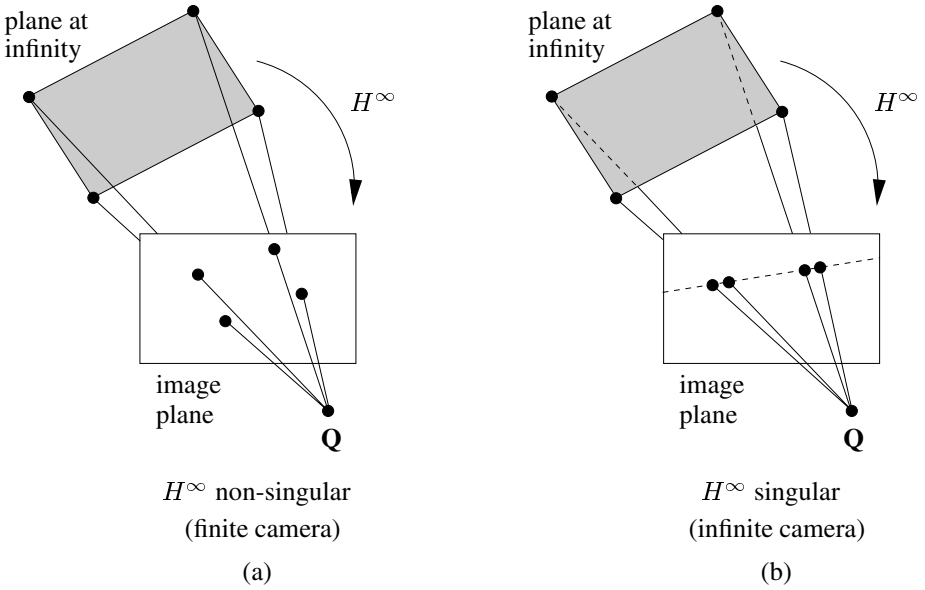


Figure 2.6. Projection of the plane at infinity onto a finite (a) and infinite (b) camera. An infinite camera means that the centre of projection **Q** lies on the plane at infinity.

$P \sim H\ (I \mid -\bar{\mathbf{Q}})$ and the mapping of a scene point $\mathbf{X} \in \mathcal{P}^3$ onto an image point $\mathbf{x} \in \mathcal{P}^2$ is

$$\mathbf{x} \sim H\ (\ I \mid -\bar{\mathbf{Q}}\ )\ \mathbf{X}\ . \tag{2.27}$$

Eqn. 2.27 corresponds to eqn. 2.25 in the Euclidean case where $H = KR$. This is correct since the matrix KR is always non-singular. Furthermore, the camera centre was defined as a finite point in the Euclidean space. If H is singular, it can be shown (e.g. Hartley and Zisserman, 2000) that $\mathbf{Q}^T = (\mathbf{d}, 0)^T$ where $H\ \mathbf{d} = 0$. Note, the right nullspace of H is one-dimensional since H is singular. This means that the camera centre is at infinity. Therefore, a camera with a non-singular matrix H is called a **finite camera** and a camera with a singular matrix H an **infinite camera**.

Consider the matrix H of eqn. 2.27 in more detail. A point $\mathbf{X} = (X, Y, Z, 0)^T$, lying on the plane at infinity π_∞, is mapped by eqn. 2.27 onto the image plane as

$$\mathbf{x} \sim H \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}\ . \tag{2.28}$$

Therefore, H can be considered as the homography between the plane at infinity and the image plane. In the following, H is denoted the **infinite homography**[2] H^∞. Therefore, a

[2]Note, this definition of the infinite homography is slightly different to (Hartley and Zisserman, 2000; Faugeras and Luong, 2001).

finite camera P_i with a non-singular H_i^∞ may be written as

$$P_i \sim H_i^\infty (I \mid -\bar{\mathbf{Q}}_i) . \tag{2.29}$$

If H^∞ is singular the plane at infinity is mapped, according to theorem 5, onto a line in the image. This corresponds to the fact that in this case the camera centre is at infinity, i.e. lies on the plane at infinity. These two situation are illustrated in fig. 2.6.

Finally, we will consider a special infinite camera, which has attracted considerable interest for the task of structure and camera recovery. A camera of the form

$$P = \begin{pmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.30}$$

is called an **affine camera**. The main property of an affine camera is that parallel lines in the world are projected to parallel lines in the image. This can be verified by projecting a point at infinity $(X, Y, Z, 0)^T$ to $(x, y, 0)^T$. Furthermore, the centre of projection is obviously at infinity since the last row of the infinite homography is $\mathbf{0}^T$. It can be shown that the last row of a general camera matrix represents the principle plane (Hartley and Zisserman, 2000). This means that the principle plane of an affine camera is the plane at infinity, i.e. $(0, 0, 0, 1)^T$. For a non-homogeneous 3D point $\bar{\mathbf{X}}$ and an image point $\bar{\mathbf{x}}$, the projection equation 2.25 of an affine camera may be written in *non-homogeneous* coordinates as

$$\bar{\mathbf{x}} = M\bar{\mathbf{X}} + \mathbf{t} , \tag{2.31}$$

where M is the top left 2×3 submatrix of P and $\mathbf{t} = (p_{14}, p_{24})^T$.

2.4 Reference Planes & Plane + Parallax

We come now to a simple concept which is essential for the understanding of the thesis. Assume that a real scene plane is "known". This scene plane is called the reference plane. How can this known reference plane be used for 3D reconstruction? The previous section defined the relationship between the plane at infinity and the camera matrix. Furthermore, in the projective space $\mathcal{P}^3$ any (reference) plane may represent the plane at infinity (sec. 2.1). Consequently, a known finite (or infinite) reference plane may give information about the cameras. This idea and further consequences are now explained in more detail.

Consider a real, finite plane (the reference plane) visible in two views (see fig. 2.7). A point $\mathbf{x}$ on the reference plane (with an arbitrary basis) is projected onto the image plane i as $\mathbf{x}_i$. This projection can be described by the projective transformation $\mathbf{x}_i \sim H_i\,\mathbf{x}$. Similarly, the point $\mathbf{x}$ is mapped onto the image plane j as $\mathbf{x}_j \sim H_j\,\mathbf{x}$. Since the set of all projective transformations form a group, the mapping between image i and j is a homography H_{ij}, which may be derived from H_i and H_j as

$$\mathbf{x}_j \sim H_j\, H_i^{-1}\, \mathbf{x}_i \sim H_{ij}\, \mathbf{x}_i . \tag{2.32}$$

Since the basis of the reference plane may be chosen freely, we choose it so that H_i is the identity matrix. Consequently, the homography H_{ij} is H_j.

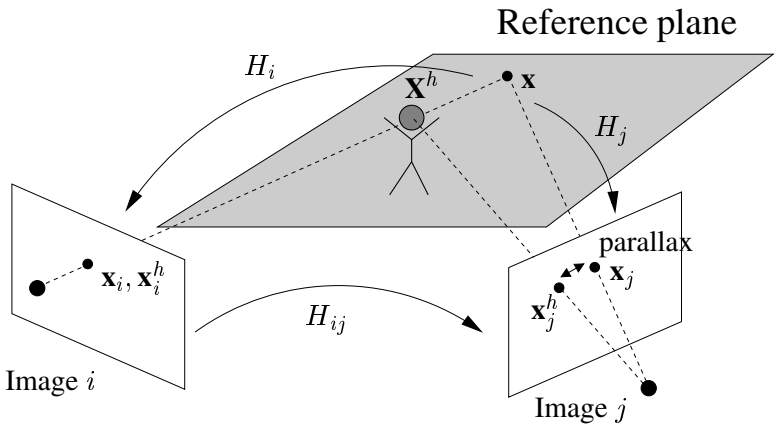


Figure 2.7. Explanation for the concept of plane + parallax

Consider the head of the man as a 3D point $\mathbf{X}^h$ which does not lie on the reference plane. The head is projected into image i and j as $\mathbf{x}_i^h$ and $\mathbf{x}_j^h$ respectively. We chose $\mathbf{X}^h$ such that its projection in the image i is $\mathbf{x}_i$, i.e. $\mathbf{x}_i \sim \mathbf{x}_i^h$. However, the two projections in image j, $\mathbf{x}_j$ and $\mathbf{x}_j^h$, do not coincide. The vector between $\mathbf{x}_j$ and $\mathbf{x}_j^h$ is called the **parallax vector** of the point $\mathbf{X}^h$ with respect to the reference plane. Obviously this vector vanishes if $\mathbf{X}^h$ is on the plane. A set of 3D points together with a plane is in the literature referred to **plane + parallax** (Carlsson and Eklundh, 1990; Kumar et al., 1994; Sawhney, 1994).

Reconsider this scenario from an "image-based" point of view. It would be convenient to superimpose the two images so that all points on the reference plane are identical and all points off the reference plane move and hence induce a parallax vector. This idea is known as **stabilizing a reference plane**. It can be achieved by specifying the reference plane as the plane at infinity and transforming the cameras into calibrated translating cameras. Although this sounds complicated, it is mathematically fairly simple. Specifying the reference plane as the plane at infinity means that the homographies H_i and H_j are represented by the infinite homographies H_i^∞ and H_j^∞. With the assumption of finite cameras P_i and P_j, a 3D point $\mathbf{X}$ is projected to the image points $\mathbf{x}_i$ and $\mathbf{x}_j$ as (see eqn. 2.29)

$$\mathbf{x}_i \sim H_i^\infty (I \mid -\bar{\mathbf{Q}}_i)\,\mathbf{X} \quad \text{and} \quad \mathbf{x}_j \sim H_j^\infty (I \mid -\bar{\mathbf{Q}}_j)\,\mathbf{X} \quad . \tag{2.33}$$

Let us warp the images i and j with the inverse homographies $H_i^{\infty -1}$ and $H_j^{\infty -1}$ respectively

$$\mathbf{x}_i' \sim H_i^{\infty -1}\,\mathbf{x}_i \sim (I \mid -\bar{\mathbf{Q}}_i)\,\mathbf{X} \quad \text{and} \quad \mathbf{x}_j' \sim H_j^{\infty -1}\,\mathbf{x}_j \sim (I \mid -\bar{\mathbf{Q}}_j)\,\mathbf{X} \quad . \tag{2.34}$$

The two new cameras matrices P_i' and P_j', of the warped images $\mathbf{x}_i'$ and $\mathbf{x}_j'$, are therefore $P_i' = (I \mid -\bar{\mathbf{Q}}_i)$ and $P_j' = (I \mid -\bar{\mathbf{Q}}_j)$. From a Euclidean point of view, these two cameras have the identity matrix as calibration matrix and no relative rotation, i.e. **calibrated**

translating cameras. Fig. 2.8 depicts the two superimposed images after stabilizing the reference plane. The feet of the man are in both images at the same position (stabilized). The head of the man is at different positions in the stabilized images. However, why do points at infinity, points on the reference plane, not move, are stabilized? Mathematically, a point at infinity, e.g. a foot of the man, is $\mathbf{X} = (X, Y, Z, 0)^T$. The projection of $\mathbf{X}$ is independent of the camera centre $\bar{\mathbf{Q}}$ and identical in both images (see eqn. 2.34)

$$\mathbf{x}'_i \sim \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \quad \text{and} \quad \mathbf{x}'_j \sim \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \quad . \tag{2.35}$$

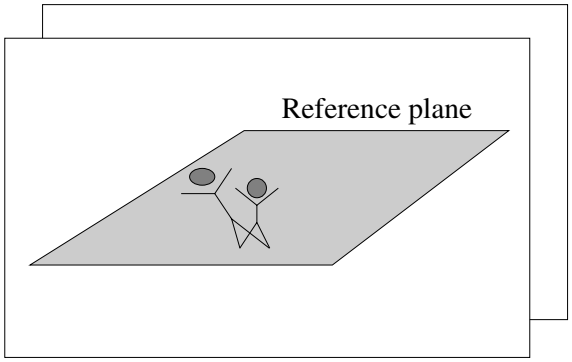


Figure 2.8. Superimposed images after stabilizing the reference plane.

In the above example we assumed that the reference plane is a *real*, finite plane in the scene. In order to stabilize the reference plane, this finite plane has been specified as the plane at infinity. In the following we will consider the case where the reference plane is already the *correct* plane at infinity. Since such a reference plane does not represent a real, finite scene plane, it is called a **virtual** reference plane. In chapter 5, different techniques of deriving a virtual reference plane are reviewed. Consider the projection of a point $\mathbf{X}$ into a *Euclidean camera* P_i with calibration matrix K_i, rotation R_i and camera centre $\bar{\mathbf{Q}}_i$ (see eqn. 2.25)

$$\mathbf{x}_i \;\sim\; K_i \, R_i \, (\, I \mid \, -\bar{\mathbf{Q}}_i) \, \mathbf{X} \quad . \tag{2.36}$$

The difference to projection equation 2.33 is that the infinite homography H_i^∞ represents now the camera's calibration and rotation matrix, $H_i^\infty = K_i \, R_i$. Since the infinite homography maps the plane at infinity to the image plane, *the correct plane at infinity may be considered as a (virtual) reference plane.* As in the real reference plane case, the image may be stabilized by warping, i.e. $\mathbf{x}'_i \;\sim\; H_i^{\infty -1} \, \mathbf{x}_i$. The result is a calibrated translating camera with $K_i = I$ and $R_i = I$. Such a scenario is shown in fig. 2.9 (a). The car represents a calibrated translating camera. Two pictures at a different time are superimposed

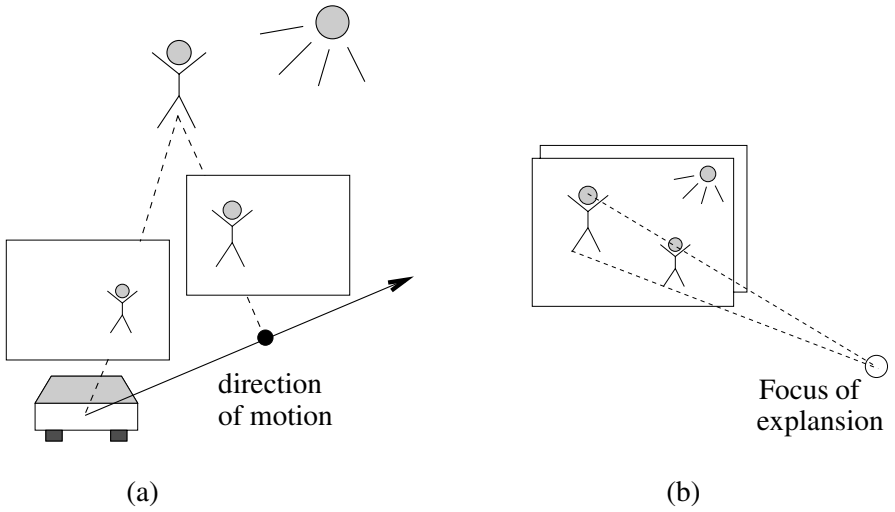


Figure 2.9. A translating car in a Euclidean space (a). Two superimposed images at a different time (b) show that the sun, a point at infinity, does not move. In this scenario the stabilized (virtual) reference plane is the correct plane at infinity.

in fig. 2.9 (b). The man which is closed to the car moves and increases in size. However, the sun, which is a point close to infinity, does not move. Therefore, the stabilized (virtual) reference plane is the correct plane at infinity. Another point at infinity is interesting in this scenario. It is specified by the direction of the moving car. The projection of this point at infinity onto the superimposed images is denoted the **focus of expansion**. This point also represents the projection of one camera centre into the other camera. The fact that the focus of expansion is defined by the projection of two distinctive 3D points, e.g. the head and one foot of the man (fig. 2.9 (b)), will be considered in the next chapter.

So far, we have not addressed the question if the formula $H_i^\infty = K_i\, R_i$ is valid for a real, finite reference planes as well? Let us write eqn. 2.33 for camera P_i as

$$\mathbf{x}_i \;\sim\; H_i^\infty \,(\, I \mid -\bar{\mathbf{Q}}_i)\, T\, T^{-1}\, \mathbf{X} \;. \tag{2.37}$$

The 4×4 transformation matrix T represents the choice of the projective coordinate system (sec. 2.3.2). Consider the case where T is an affine transformation T_A (sec. 2.2.2). Projection equation 2.37 may then be written as

$$\mathbf{x}_i \;\sim\; K_i\, R_i\, A\, (\, I \mid -\bar{\mathbf{Q}}_i)\, T_A^{-1}\, \mathbf{X} \qquad \text{for} \qquad T_A \;=\; \begin{pmatrix} A & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix} \;. \tag{2.38}$$

If the matrix T_A is the identity matrix, $T_A = I$, the space is Euclidean and $H_i^\infty = K_i\, R_i$. However, in an affine space, A may represent *any* matrix. In this case, a known infinite homography $H_i^\infty = K_i\, R_i\, A$ can *not* be decomposed into a calibration and rotation matrix. More precisely, H_i^∞ is only the product of a calibration and rotation matrix if $A = \lambda R$,

i.e. the space is metric. As discussed in sec. 2.2.3, λ and R represent the scaling and rotation of a metric space. In the example of a real, finite reference plane, the plane at infinity is not in its correct position. Therefore T represents a projective transformation and H_i^∞ is *not* the product of a calibration and rotation matrix. To conclude, the infinite homography is *only* in a metric space the product of the camera's calibration and rotation matrix, $H_i^\infty = K_i\, R_i$.

Let us summarize the above discussion. Any real or virtual plane may be chosen as a reference plane. In a projective setting, the reference plane may represent the plane at infinity. This specification has the advantage that the homography between the reference plane and an image is the *infinite homography* of the respective camera. The reference plane homography "encodes" the calibration and rotation information of the respective camera. However, the infinite homography is only in a metric space the product of the camera's calibration and rotation matrix, $H^\infty = KR$. These basic observations have been noted in many publications about plane+parallax (e.g. Irani et al., 1998; Triggs, 2000). By stabilizing a reference plane in an image, the calibration and rotation of the respective camera is canceled out. This gives a set of *calibrated translating cameras with the camera centre as the only remaining unknown*. The idea of stabilizing the images to obtain calibrated translating cameras has as well been suggested by Heyden and Åström (1995a), Triggs (2000) and Shashua and Navab (1994). Irani et al. (1998) and Criminisi et al. (1998) investigate the plane+parallax scenario by mapping the 3D points onto the reference plane itself. This transformation is basically identical to stabilizing the images. The condition for image stabilization is that the cameras are finite, i.e. their centres do not lie on the reference plane. This is a valid assumption for almost all real and virtual reference planes (see chapter 5). However, the reference plane concept, i.e. choosing the reference plane as the plane at infinity, may be applied as well to infinite cameras. In most parts of the thesis, finite cameras and therefore stabilized images are assumed for simplicity. However, for the sake of generality, the main results and algorithms using reference planes are derived for both infinite and finite cameras.

2.5 Conclusion

This chapter reviewed basic concepts of geometry for computer vision. Section 2.1 presented general n-dimensional affine and projective spaces. Furthermore, points, lines and planes in $\mathcal{P}^2$ and $\mathcal{P}^3$ were analyzed. The important concept of stratification of 2D and 3D geometry was the subject of sec. 2.2. Moreover, the approach of uncalibrated structure and camera recovery was introduced. The advantages of this approach in contrast to the traditional way of calibrated reconstruction were explained. Section 2.3 described the differences between a Euclidean and a projective camera. The term infinite homography was introduced slightly different to (Hartley and Zisserman, 2000). Finally, we considered the scenario of a plane visible in two views (sec. 2.4). The key concepts of plane+parallax, stabilizing a reference plane, real and virtual reference planes and calibrated translating cameras were reviewed.

Chapter 3

Projective Multi-View Geometry: General versus Reference Plane

This chapter comprises of most theoretical concepts used in the thesis. It investigates theoretically the geometry of multiple images for two different configurations: *scenes with a reference plane* and *general scenes*. The analysis is carried out for three different feature types: *points, lines* and *planes*. The result are different theoretical approaches to reconstruct multiple features observed in multiple views. The approaches may be categorized into: *camera constraint*, *structure constraint* and *factorization* methods. These categories are well known and will be reviewed here for both scene types, general and reference plane. The main contribution of this chapter is the introduction of a *new category* for reference plane configurations. This category is based on the *novel* observation that the relationship between 3D features and cameras is *linear* if a reference plane is given. In contrast, this relationship is *non-linear* for general configurations. The linear relationship makes it possible to simultaneously reconstruct *all* cameras and *all* features in a *single* linear system. We call this new category the *direct reference plane (DRP) approach*.

The emphasis of this chapter is *not* on a literature review of practical reconstruction methods. The purpose is to review theoretical approaches of solving the reconstruction problem. In order to present the basic ideas in a simple way, some assumptions will be made, such as image features being matched correctly. These assumptions are abolished in the next chapter. The next chapter presents a literature review and comparison of practical reconstruction methods.

The first sec. 3.1 introduces formally the reconstruction task and related problems. The remaining sections examine separately the three different feature types: *points* (sec. 3.2), *lines* (sec. 3.3) and *planes* (sec. 3.4). The main focus is on point features. The novel discussion of points for reference plane configurations is in sec. 3.2.1 (single view case) and sec. 3.2.2 (multi-view case). This is based on our publications (Rother and Carlsson, 2001; Rother and Carlsson, 2002b; Rother and Carlsson, 2002a). The readers who are familiar with the subject of multi-view geometry might skip the other subsections

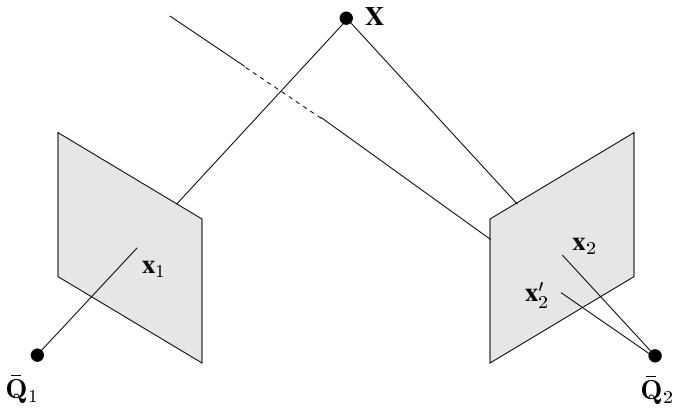


Figure 3.1. Explanation of the concept of geometric constraints. Corresponding image points $\mathbf{x}_1$ and $\mathbf{x}_2$ constrain both the position of the 3D point $\mathbf{X}$ and the two camera centres $\bar{\mathbf{Q}}_1, \bar{\mathbf{Q}}_2$.

for point features. However, we point out that the review of reference plane configurations in sec. 3.2.3 (camera and structure constraints) and 3.2.4 (factorization) is less well known. For lines, the novel aspects for reference plane configurations are in sec. 3.3.1 (single view case) and sec. 3.3.2 (multi-view case). This discussion is not part of any of our previous publications. For planes, the sections 3.4.1, 3.4.2 and 3.4.3 comprise of novel aspects which are partly based on (Rother et al., 2002). The idea of factorizing planes and cameras (sec. 3.4.4) has been presented in (Triggs, 2000; Rother et al., 2002).

This chapter describes theoretically our novel direct reference plane methods for the three feature types, points, lines and planes. The corresponding practical algorithms are outlined in sec. 6.1.1 for points, sec. 6.2.1 for lines and sec. 6.3.1 for planes.

After considering separately the three feature types, sec. 3.5.1 discusses methods of combining them. Furthermore, we examine how to incorporate scene constraints, like a point lies on a plane, in the reconstruction process (sec. 3.5.2). The main observation is that our direct reference plane reconstruction method extends straightforward to all three types of features and may include several interesting scene constraints, such as incidence relationships. Section 3.5 does not involve any novel concepts, and is not important for the understanding of the following chapters.

3.1 Introduction

Consider a 3D point $\mathbf{X}$ seen by two cameras with centres $\bar{\mathbf{Q}}_1$ and $\bar{\mathbf{Q}}_2$ (fig. 3.1). The projection of $\mathbf{X}$ into cameras 1 and 2 gives the image points $\mathbf{x}_1$ and $\mathbf{x}_2$. This means that the two rays of sight $(\bar{\mathbf{Q}}_1, \mathbf{x}_1)$ and $(\bar{\mathbf{Q}}_1, \mathbf{x}_2)$ intersect in the 3D point $\mathbf{X}$. What about the two image points $\mathbf{x}_1$ and $\mathbf{x}'_2$, do they represent a unique point in space? Obviously not, since in this case the two rays of sight $(\bar{\mathbf{Q}}_1, \mathbf{x}_1)$ and $(\bar{\mathbf{Q}}_1, \mathbf{x}'_2)$ do not intersect in space. However, by moving the camera centre $\bar{\mathbf{Q}}_2$, these two rays would intersect in the unique

3D point $\mathbf{X}$. Equivalently, the 3D point $\mathbf{X}$ can be moved, so that the two rays $(\bar{\mathbf{Q}}_1, \mathbf{x}_1)$ and $(\bar{\mathbf{Q}}_1, \mathbf{x}'_2)$ intersect uniquely in this point. This means, that the image points constrain both the position of the cameras and of the 3D point. More generally, any feature seen in two or more views constrains the position of cameras and the feature. These **geometric constraints** make it possible to pose the following question: *Given a sufficient number of image features, what are the positions of the features and cameras in space?* It is important to note that the two tasks of reconstructing the structure and the cameras are interlinked and should not be considered as two decoupled problems.

In the discussion above we implicitly made the assumption that it is known which image features in two or more views represent the same 3D feature in space. In general, the only source of information for the reconstruction problem is images. Therefore, the first step of any "feature-based" reconstruction algorithm is to detect image features which are in correspondence. In general, image features of multiple views are said to be in **correspondence** if they represent the same feature in 3D space. The problem of detecting corresponding image features is known as the **matching problem**. With a continuous image sequence, e.g. from a video camera, this problem can be solved by tracking image features (e.g. Isard and Blake, 1998). If the cameras are far apart, i.e. have a wide-baseline, the problem is substantially more difficult (e.g. Tell, 2002). In this chapter it is assumed that the matching problem is solved. However, the matching problem and the reconstruction problem are linked together via the geometric constraints introduced above. The image points $\mathbf{x}_1$ and $\mathbf{x}'_2$ in fig. 3.1 do not correspond for these two cameras since they do not define a unique point in space. In general, a number of image features only correspond if they define a unique reconstruction, i.e. a unique set of cameras and features in space. Chapter 8 of the thesis presents a system which solves both the matching and reconstruction problem automatically.

A further implicit assumption made so far is that the scene is rigid and the camera is moving, i.e. it is at different positions in space. From a geometric point of view this is equivalent to a fixed camera and a moving scene. The more general task is to have a moving camera which observes independent moving objects in the scene. This task is known as the **reconstruction of dynamic scenes** and has recently raised a lot interest (*Vision and Modelling of Dynamic Scenes*, 2002). However, it is still less understood then the more simple rigid scene case. A further useful application of the geometric constraints is to detect moving objects in a static scene. Assume that the 3D point $\mathbf{X}$ in fig. 3.1 moves, while the camera moves from position 1 to 2. This means that $\mathbf{X}$ is projected in image 1 as $\mathbf{x}_1$ and image 2 as $\mathbf{x}'_2$. With the assumption of known cameras, the geometric constraint that $\mathbf{x}_1$ and $\mathbf{x}'_2$ do correspond would be violated. Therefore, the geometric constraints can be used to detect moving features of a rigid scene. In summary, the geometric constraints may be applied to the four fundamental tasks:

- Reconstruction of 3D features and cameras
- Matching of image features
- Detecting moving features in a rigid scene
- New view synthesis

The idea of new view synthesis is to create new views of a scene from images only, i.e. without computing explicitly the 3D structure and cameras. This thesis concentrates on the reconstruction task. However, a completely automatic reconstruction system involves the matching task as well. Such a system is presented in chapter 8. The third task of moving object detection will be briefly addressed in this chapter.

Let us now specify the problems involved in determine structure and cameras more formally. This will be done on the basis of point features, however the extension to other feature types is straightforward. In the previous chapter we have mathematically formulated the projection of one 3D point feature onto the image plane via a pinhole camera. Let us generalize this setting for multiple point features and multiple cameras in projective space. As was seen in the first chapter, points inevitably become occluded as the camera's view changes. This problem is known as the **missing data problem** and every "real world" reconstruction algorithm has to deal with it.

Definition 3 (Structure and camera recovery) *Given a sufficient number of corresponding image points $\boldsymbol{x}_{ij}$. The task of structure and camera recovery is to determine uniquely the unknown 3D points $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$, unknown cameras $P_1, \ldots, P_m$ and unknown scalars λ_{ij}, so that the projection relation:*

$$\lambda_{ij}\, \boldsymbol{x}_{ij} = P_j\, \boldsymbol{X}_i \tag{3.1}$$

is satisfied.

The unknown scalars λ_{ij} are sometimes denoted **projective depths** (e.g. Sturm and Triggs, 1996). This definition of the reconstruction problem immediately raises two questions: When is the reconstruction unique and how many image features are needed?

The first question is known as the **critical configuration problem**. Consider two configurations $(\mathbf{X}_i, P_j)$ and $(\mathbf{X}'_i, P'_j)$ of 3D points $\mathbf{X}_i, \mathbf{X}'_i$ and cameras P_j, P'_j which are related by a projective transformation H so that

$$\mathbf{X}_i \sim H\mathbf{X}_i \quad \text{and} \quad P'_j \sim P_j H^{-1} \quad . \tag{3.2}$$

These two configurations have the same projective image coordinates, since $\mathbf{x}_{ij} \sim P_j X_i \sim P_j H^{-1} H X_i \sim P'_j X'_i$. Therefore we denote these configurations as *equivalent*, any configuration $(\mathbf{X}_i, P_j)$ has an equivalent class of solutions $(H\,\mathbf{X}_i,\ P_j\ H^{-1})$, which have the same projective image coordinates. The remaining question is if there are any *inequivalent* configurations which have the same image coordinates. This leads to the definition of critical configurations.

Definition 4 (Critical Configurations) *A configuration $(\boldsymbol{X}_i, P_j)$ is called a critical configuration, if an inequivalent configuration $(\boldsymbol{X}'_i, P'_j)$ exists such that both configurations have the same projective image coordinates, i.e. $P_j\, \boldsymbol{X}_i \ \sim\ P'_j\, \boldsymbol{X}'_i$ for all i, j.*

The study of critical configurations has a long history in the field of photogrammetry and computer vision. The first publication was probably by Krames (1942). This problem is the subject of discussion in chapter 7.

The second question of **sufficient image data** is "fairly" easy to answer if all 3D features are visible in all views (e.g. Hartley and Zisserman, 2000). The projections of the 3D points into a camera give a certain number of constraints which has to be equal to or larger than the number of unknowns contained in the 3D points and cameras. This chapter gives a complete answer to this question for the three feature types: points, lines and planes and two different configurations: general and reference plane. With missing data this problem is more difficult to solve (e.g. Quan et al., 1999). It will be addressed in chapter 7.

The reconstruction problem becomes fundamentally more simple if either the cameras or the structure is known a-priori. The problem of determine the structure from *known cameras* is called the **intersection** or **triangulation problem** and the the problem of camera recovery from *known structure* the **resection problem**. Eqn. 3.1 indicates that both problems are linear in the unknown structure and scale factors or the unknown cameras and scale factors. This means, that both problems can be solved with "standard" linear methods as we will see later. Furthermore, the resection problem is identical to the "classical" calibration problem, where a camera is calibrated from a calibration object with known 3D coordinates.

The thesis limits the investigation of multi-view geometry to three feature types: **points, lines and planes**. For the task of 3D reconstruction, points have been the most popular feature type. This can be seen as well from the larger number of publications devoted to the topic of point matching in contrast to line matching (see Tell (2002) for an overview). A 3D points has 3 degrees of freedom in $\mathcal{P}^3$, whereas a 3D line has 4 degrees of freedom. As a consequence, more 3D lines than 3D points are needed to obtain a 3D reconstruction. In particular, uncalibrated reconstruction of two views is impossible with 3D lines only. An interesting aspect of 3D points is that camera centres are 3D points as well. This leads to the fundamental duality theorem, which is discussed in this chapter. An advantage of line features is that they can be detected more accurately in the image, since a line or line segment extends over a larger image area. Furthermore, man-made environments are characterized by a lot of linear structures. The third feature type, i.e. planes, is present in many man-made environments. Planes can be represented by homographies between pairs of views. One way of estimating homographies is to use point and/or line features which are already matched. Therefore, planes are seldom used as the only feature type for 3D reconstruction. It is more popular to considered them in a post-processing step for improving a point or line reconstruction. Alternatively, homographies can be obtained directly from greylevels in a sequence of images (e.g. Bergen et al., 1992; Irani and Anandan, 1999a). However, even in this case the task of plane reconstruction from homographies may be circumvented by "hallucinating" 3D points. From the homographies, 3 corresponding image points (or lines) in multiple views may be hallucinated, to represent the 3D plane uniquely.

After considering the three feature types separately, methods of combining them are investigated in the last section. Such methods have the advantage that all three feature types constrain the camera's position simultaneously. Furthermore, external constraints, e.g. the incidence relationship of a point lying on a plane, may be integrated in the framework.

3.2 Points

We start the discussion by comparing a *general* point configuration with a *plane+parallax* point configuration, i.e. 3D points and a reference plane, in the single view case. The conclusion of this discussion will be that the structure and camera recovery problem is *bi-linear* in the general case and *linear* in the reference plane case. On the basis of this result we will present four different categories of formulating and investigating the relationship between multiple points in multiple views:

- Use the reference plane to reconstruct cameras and points simultaneously from a linear system (sec. 3.2.2).
- Derive from the bi-linear projection relation the so-called *camera constraints* which involve only cameras and image points. From the camera constraints the cameras may be derived (sec. 3.2.3).
- The dual counterpart to the camera constraints are the *structure constraints*, which involve image points and 3D points. From the structure constraints the structure may be derived (sec. 3.2.3).
- Combine all image coordinates into a large "measurement matrix", which can be used to compute points and cameras simultaneously by factorization (sec. 3.2.4).

The last three approaches are not specialized to reference plane configurations. Therefore, they are studied for general and reference plane configurations. It will turn out that the reference plane case simplifies these approaches.

3.2.1 Single View: General versus Reference Plane

In order to simplify the analysis of single view geometry, a specific projective basis using reference points is chosen in the image and 3D world (e.g. Faugeras, 1992; Quan et al., 1994; Heyden and Åström, 1995a; Carlsson, 1995). How to choose the projective basis independently of specific reference points is discussed later. Figure 3.2 shows two point configurations where the 3D points $\mathbf{X}_{1-4}$ are either in general pose (a) or on on a reference plane (b).

General configurations

We have seen in sec. 2.3.2 that in a projective setting a projective basis in the image and the 3D world may be chosen freely. Let us use for both spaces the standard basis as defined in eqn. 2.3. The scene points $\mathbf{X}_{1-4}$ are mapped to the image points $\mathbf{x}_{1-4}$ as

$$\begin{array}{ccccc} \mathbf{X}_1 & \mathbf{X}_2 & \mathbf{X}_3 & \mathbf{X}_4 & \mathbf{Q} \\ - & - & - & - & - \\ 1 & 0 & 0 & 0 & A \\ 0 & 1 & 0 & 0 & B \\ 0 & 0 & 1 & 0 & C \\ 0 & 0 & 0 & 1 & D \end{array} \quad \longrightarrow \quad \begin{array}{ccccc} \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & \mathbf{x}_4 & \mathbf{0} \\ - & - & - & - & - \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{array} \tag{3.3}$$

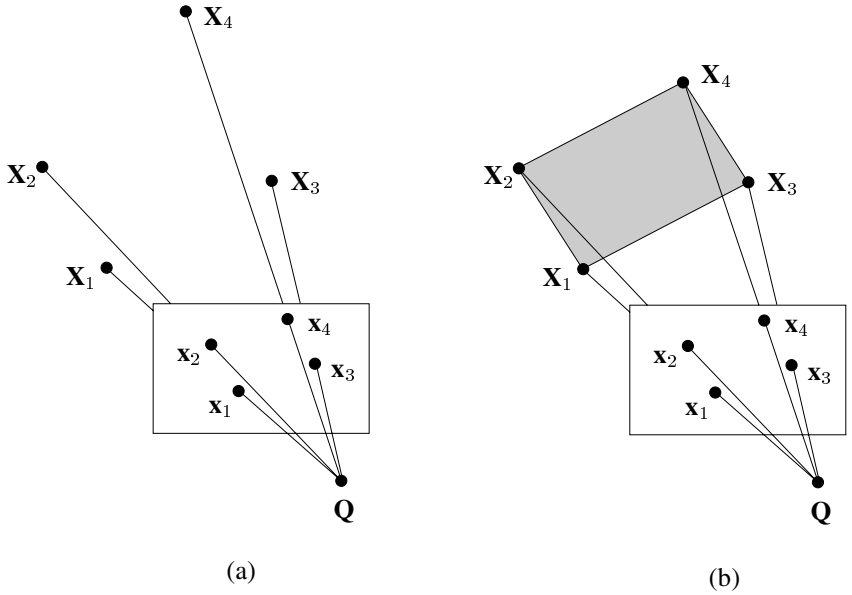


Figure 3.2. A general point configuration (a) and reference plane configuration (b). The 3D points $\mathbf{X}_{1-4}$ and image points $\mathbf{x}_{1-4}$ are used as a projective basis in space and in the image.

where the camera centre is mapped onto the centre of projection, i.e. $(0,0,0)^T$. On the basis of this choice, the projection of any point $\mathbf{X}$ to the image point $\mathbf{x}$ is defined as (Carlsson and Weinshall, 1998)

$$\begin{pmatrix} x \\ y \\ w \end{pmatrix} \sim \begin{pmatrix} A^{-1} & 0 & 0 & -D^{-1} \\ 0 & B^{-1} & 0 & -D^{-1} \\ 0 & 0 & C^{-1} & -D^{-1} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ W \end{pmatrix} . \tag{3.4}$$

This shows, that in a projective setting a camera is uniquely defined by its projection centre. Furthermore, the position and calibration of the image plane is not relevant in this context. Eqn. 3.4 can be transformed into the following projection relation:

$$\begin{pmatrix} x \\ y \\ w \end{pmatrix} \sim \begin{pmatrix} \frac{X}{A} - \frac{W}{D} \\ \frac{Y}{B} - \frac{W}{D} \\ \frac{Z}{C} - \frac{W}{D} \end{pmatrix} . \tag{3.5}$$

The unknown scale factor can be eliminated, which gives the two equations

$$\begin{aligned} w\,\frac{X}{A} \;-\; x\,\frac{Z}{C} \;+\; (x-w)\,\frac{W}{D} \quad &= \quad 0 \\ w\,\frac{Y}{B} \;-\; y\,\frac{Z}{C} \;+\; (y-w)\,\frac{W}{D} \quad &= \quad 0\ . \end{aligned} \tag{3.6}$$

These equations describe explicitly the duality of space points and camera centres in the sense that the *homogeneous* projective coordinates of a space point $(X, Y, Z, W)^T$ and the inverse coordinates of a camera centre $(A^{-1}, B^{-1}, C^{-1}, D^{-1})^T$ are *bi-linearly* related in a symmetric way. The choice of the fifth basis point P_5 has further consequences (e.g. Carlsson, 1995) which are not, however, relevant in this context.

Reference plane configurations

Let us analyze the reference plane case, where the four points $\mathbf{X}_{1-4}$ define a reference plane in the scene (see fig. 3.2 (b)). The mapping of these points can be used as a basis in the projective image plane if the centre of projection is not on the reference plane. If the camera centre lies on the reference plane (see fig. 2.6 (b)), the four basis points are collinear in the image. Note that in the reference plane case, the point $\mathbf{X}_4$ cannot be used as a basis point for the projective space $\mathcal{P}^3$, which has to consist of five non-coplanar points. Let us specify these four points so that they define the plane at infinity, i.e. $\mathbf{X}_4 = (1, 1, 1, 0)^T$, (e.g. Shashua and Navab, 1994; Heyden and Åström, 1995a; Triggs, 2000). The mapping between scene points and image points is now

$$\begin{array}{cccccc} \mathbf{X}_1 & \mathbf{X}_2 & \mathbf{X}_3 & \mathbf{X}_4 & \mathbf{Q} \\ - & - & - & - & - \\ 1 & 0 & 0 & 1 & A \\ 0 & 1 & 0 & 1 & B \\ 0 & 0 & 1 & 1 & C \\ 0 & 0 & 0 & 0 & D \end{array} \quad \longrightarrow \quad \begin{array}{ccccc} \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & \mathbf{x}_4 & \mathbf{0} \\ - & - & - & - & - \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{array}\ . \tag{3.7}$$

This means that the mapping of a general 3D point $\mathbf{X}$ onto the image point $\mathbf{x}$ is

$$\begin{pmatrix} x \\ y \\ w \end{pmatrix} \sim \begin{pmatrix} 1 & 0 & 0 & -A/D \\ 0 & 1 & 0 & -B/D \\ 0 & 0 & 1 & -C/D \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ W \end{pmatrix} \tag{3.8}$$

or more compactly

$$\mathbf{x} \;\sim\; (\, I \mid -\bar{\mathbf{Q}})\, \mathbf{X}\ . \tag{3.9}$$

This equation is identical to eqn. 2.33, which represented the projection equation for a stabilized image, i.e. $\mathbf{x}' \sim H^{\infty-1}\,\mathbf{x}$. Therefore, such a stabilization is equivalent to

changing the image basis as described above. Furthermore, the assumption of having four coplanar 3D points is equivalent to the assumption of having a known reference plane, i.e. the infinite homography H^∞. The infinite homography can be defined explicitly on the basis of the image points as:

$$H^\infty : \quad \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \longrightarrow (\mathbf{x}_1\ \mathbf{x}_2\ \mathbf{x}_3\ \mathbf{x}_4) \quad . \tag{3.10}$$

Obviously, it is implicitly assumed that the camera does not lie on the reference plane, i.e. that the infinite homography is non-singular.

Let us now consider only those scene points which do not lie on the plane at infinity, i.e. $W \neq 0$. Those points form an affine space, which is the projective space without the plane at infinity. From eqn. 3.8 the following projection relation may be derived

$$\begin{pmatrix} x \\ y \\ w \end{pmatrix} \sim \begin{pmatrix} \frac{X}{W} - \frac{A}{D} \\ \frac{Y}{W} - \frac{B}{D} \\ \frac{Z}{W} - \frac{C}{D} \end{pmatrix} \tag{3.11}$$

or more compactly

$$\mathbf{x} \sim \bar{\mathbf{X}} - \bar{\mathbf{Q}} \quad . \tag{3.12}$$

If we compare this to the general projection relation in eqn. 3.5, we see that the relationship between points and cameras is different, though, still symmetric. The symmetry now relates to the substitutions $(X, Y, Z, W) \leftrightarrow (A, B, C, D)$. More importantly the relationship between a non-homogeneous point and a non-homogeneous camera centre is **linear** in contrast to bi-linear in the general case. We will see that this linearity leads to a simple relationship between points and cameras in the multiple view case.

The projection relation can be transformed into linear constraints by eliminating the unknown scale:

$$\begin{array}{rcrcl} x\,(\bar{Z} - \bar{C}) & - & w\,(\bar{X} - \bar{A}) & = & 0 \\ y\,(\bar{Z} - \bar{C}) & - & w\,(\bar{Y} - \bar{B}) & = & 0 \\ x\,(\bar{Y} - \bar{B}) & - & y\,(\bar{X} - \bar{A}) & = & 0\ , \end{array} \tag{3.13}$$

where $\bar{X}, \bar{Y}, \bar{Z}$ and $\bar{A}, \bar{B}, \bar{C}$ represent the coordinates of the non-homogeneous point $\bar{\mathbf{X}}$ and camera centre $\bar{\mathbf{Q}}$, e.g. $\bar{X} = X/W$. Obviously, only two of the three equations are linearly independent. However, two relations are insufficient for special cases where e.g. $w = 0$ and $\bar{Z} - \bar{C} = 0$.

What happens to infinite points, i.e. points which lie on the plane at infinity? Using eqn. 3.9, a point at infinity $\mathbf{X} = (X, Y, Z, 0)^T$ is related directly with its image point $\mathbf{x} = (x, y, w)^T$ as:

$$\begin{pmatrix} x \\ y \\ w \end{pmatrix} \sim \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \quad . \tag{3.14}$$

Note that the above discussion assumed that the image is stabilized and the infinite homography is non-singular, i.e. the camera is finite. The same analysis can be carried out without these restrictions. This will be done in sec. 6.1.1 and it will turn out that the relationship between point and camera parameters is even in this case linear.

Conclusion

The general *bi-linear* projection relation between a point $\mathbf{X}$ and a camera P is

$$\mathbf{x} \sim P\,\mathbf{X} \sim H^{\infty}\,(I \mid -\bar{\mathbf{Q}})\,\mathbf{X} \ , \tag{3.15}$$

where $\bar{\mathbf{Q}}$ is the camera centre and H^{∞} the infinite, non-singular homography. If four 3D points are coplanar, the projection relation is *linear* for *non-homogeneous* points $\bar{\mathbf{X}}$:

$$\mathbf{x}' \sim H^{\infty -1}\,\mathbf{x} \sim \bar{\mathbf{X}} - \bar{\mathbf{Q}} \ . \tag{3.16}$$

Note that the assumptions of having four coplanar points or a known reference plane, i.e. H^{∞}, are equivalent. This simple relationship between points and cameras is achieved by stabilizing the image, i.e. $\mathbf{x}' \sim H^{\infty -1}\,\mathbf{x}$, or equivalently choosing a specific image basis. Therefore, the difference between general point configurations and point configurations with a reference plane may be summarized as follows:

> *In general, points and cameras have a bilinear relationship in projective space. If four points are on a plane, points and cameras have a linear relationship in an affine space where this plane represents the plane at infinity.*

Shashua and Navab (1994), Heyden and Åström (1995a) and Triggs (2000) use the same projective basis to formulate the relationship between points and cameras. However, they did not discover and use the linear relationship for reconstruction, as we did (Rother and Carlsson, 2001). Shashua and Navab (1994) called the affine space with the reference plane as the plane at infinity the "relative affine structure".

3.2.2 Multiple Views & Reference Plane: Linear System of Cameras and Points

We introduces now our direct reference plane (DRP) method for multiple points observed in multiple views. This novel reconstruction approach is based on our publications (Rother and Carlsson, 2001; Rother and Carlsson, 2002b; Rother and Carlsson, 2002a). Note that the practical algorithm of this method is outlined in sec. 6.1.1.

Consider the reference plane case for multiple 3D points and multiple cameras. All points which are not on the reference plane give 3 linear equations of the form 3.13. Therefore, for an arbitrary numbers of points and views, we can build a *single* linear system consisting of *all* projection relations. For n points in m views the linear system takes the form:

$$\begin{pmatrix} S_{11} & 0 & 0 & \dots & 0 & 0 & -S_{11} & 0 & \dots & 0 \\ S_{12} & 0 & 0 & \dots & 0 & 0 & 0 & -S_{12} & \dots & 0 \\ \vdots & & & & & \vdots & & & & \\ S_{1m} & 0 & 0 & \dots & 0 & 0 & 0 & 0 & \dots & -S_{1m} \\ 0 & S_{21} & 0 & \dots & 0 & 0 & -S_{21} & 0 & \dots & 0 \\ 0 & S_{22} & 0 & \dots & 0 & 0 & 0 & -S_{22} & \dots & 0 \\ & \vdots & & & & \vdots & & & & \\ 0 & S_{2m} & 0 & \dots & 0 & 0 & 0 & 0 & \dots & -S_{2m} \\ & & & & \vdots & & & & & \\ 0 & 0 & 0 & \dots & 0 & S_{n1} & -S_{n1} & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 & S_{n2} & 0 & -S_{n2} & \dots & 0 \\ & & & & & \vdots & & & & \\ 0 & 0 & 0 & \dots & 0 & S_{nm} & 0 & 0 & \dots & -S_{nm} \end{pmatrix} \begin{pmatrix} \bar{X}_1 \\ \bar{Y}_1 \\ \bar{Z}_1 \\ \vdots \\ \bar{X}_n \\ \bar{Y}_n \\ \bar{Z}_n \\ \\ \bar{A}_1 \\ \bar{B}_1 \\ \bar{C}_1 \\ \vdots \\ \bar{A}_m \\ \bar{B}_m \\ \bar{C}_m \end{pmatrix} = 0 \tag{3.17}$$

for non-homogeneous projective point coordinates $\bar{\mathbf{X}}_i$ and camera centres $\bar{\mathbf{Q}}_j$. The 3×3 matrices $S_{i,j}$ are defined as

$$S_{i,j} = \begin{pmatrix} 0 & w_{ij} & -y_{ij} \\ -w_{ij} & 0 & x_{ij} \\ y_{ij} & -x_{ij} & 0 \end{pmatrix} \tag{3.18}$$

and are built up solely from image coordinates of point i visible in view j. In the following we denote the matrix which forms the linear system in eqn. 3.17 as the system matrix or S-matrix.

The linear system can be used to compute the unknown 3D points and cameras directly from the known image measurements[1]. Throughout the thesis this method is denoted the **DRP method**, i.e. *D*irect *R*eference *P*lane method. Note that the linear system deals naturally with the problem of missing data, since the projection relations of points which are *not* visible in a certain view are not part of the system. However, how are 3D points on the reference plane reconstructed? A point at infinity $\mathbf{X} = (X, Y, Z, 0)^T$ can be reconstructed directly from image coordinates using eqn. 3.14.

[1]The experiments in sec. 6.1.2 will demonstrate that this method is capable of reconstructing difficult scenes where general reconstruction methods fail.

The solution of the linear system in 3.17 can be obtained by a Singular Value Decomposition (SVD) of the S-matrix: $S = U\ D\ V^T$ (Golub and Van Loan, 1996). In practice, only the matrix V is computed which reduces the computation time considerably as discussed in sec. 6.1. The singular vectors of V, which correspond to the singular values in D that are zero, represent the right nullspace of the S-matrix. Furthermore, the nullspace represents the set of all solutions of the homogeneous linear system. Let us consider the size of this nullspace. Apart from the true solution for all points and cameras, the following three trivial solution exist: $\bar{\mathbf{X}}_i = \bar{\mathbf{Q}}_j = (1,0,0)^T$, $\bar{\mathbf{X}}_i = \bar{\mathbf{Q}}_j = (0,1,0)^T$ and $\bar{\mathbf{X}}_i = \bar{\mathbf{Q}}_j = (0,0,1)^T$. This means that the nullspace is (at least) of dimension four. Let us reconsider this more formally. We have seen in sec. 2.2.1, that the 3D projective space has 15 degrees of freedom. This can be expressed as a 4×4 homography which transforms a point $\mathbf{X}$ to $\mathbf{X}'$ as:

$$\mathbf{X}' \sim \begin{pmatrix} A & \mathbf{t} \\ \mathbf{b}^T & \lambda \end{pmatrix} \mathbf{X} \ , \tag{3.19}$$

where A is a 3×3 matrix, $\mathbf{b}^T, \mathbf{t}$ are 3-dimensional vectors and λ a scalar (e.g. Hartley and Zisserman, 2000; Faugeras and Luong, 2001). The special choice of the 3D points $\mathbf{X}_{1-4}$ as in eqn. 3.7 implies that $A = \mu I$ and $\mathbf{b}^T = (0,0,0)$. This means that 11 of the 15 degrees of freedom of the projective space are fixed. The remaining 4 degrees of freedom correspond to the arbitrary choice of $\mathbf{t}$, μ and λ (minus an overall scale). Therefore, the nullspace of the S-matrix has to be (at least) of dimension four. However, we have assumed implicitly in this argumentation that only one non-trivial solution exists. A point configuration with more than one non-trivial solution has been denoted a *critical configuration*. Such critical configurations in the reference plane case are discussed in chapter 7. In practice, the non-trivial solution of the linear system can be obtained by either fixing a point or a camera as the origin of the space, e.g. $\bar{\mathbf{Q}}_1 = (0,0,0)^T$, or by summation of the four singular vectors of the nullspace.

What happens if we put a point which lies on the reference plane, e.g. $\mathbf{X}_1 = (X_1, Y_1, Z_1, 0)$, into the linear system (3.17)? The projection relations (3.14) can be written as linear constraints:

$$\begin{array}{rcl} xZ_1 - wX_1 & = & 0 \\ yZ_1 - wY_1 & = & 0 \\ xY_1 - yX_1 & = & 0 \end{array} \ . \tag{3.20}$$

In this case the submatrices $S_{1,j}$ contain these equations instead of the equations in 3.13. This means that the vector $(X_1, Y_1, Z_1, 0, \ldots 0)$ represents now the non-trivial solution. Furthermore, a second point on the reference plane, e.g. $\mathbf{X}_2 = (X_2, Y_2, Z_2, 0)$, would give an *additional* solution $(0,0,0,X_2,Y_2,Z_2,0,\ldots 0)$. This means that n points on the reference plane give a nullspace of dimensionality $n+3$. As a consequence, the reconstruction of all points and cameras, which are not on the reference plane, cannot be obtained from the linear system if one or more points lie on the reference plane. This means that points on and off the reference plane have to be separated and reconstructed independently. Note, this "separation" is not necessary if the reference plane is the "correct" plane at infinity, since

all finite points in the 3D world do not lie on the reference plane. How this "separation" can be done automatically and how the linear system can be formulated in a numerically optimal way will be discussed in chapter 6. Furthermore, the linear system may include infinite cameras as well, where the cameras' centre lies on the reference plane. This is discussed in sec. 6.1.1, where different versions of the DRP method are presented.

Let us summarize the three main advantages of this approach for multiple view reconstruction: *All* points not on the reference plane and *all* camera centres are reconstructed simultaneously, the process is linear and missing data is handled naturally. A linear process has the great advantage that only one non-trivial solution exists (for non-critical configurations). In contrast to this, the space of all solutions of a non-linear process, e.g. bi-linear, might have many local minima. The task of finding the global minimum is in general complex and known as non-linear optimization (Press et al., 1988).

One remaining question is: What is the minimum number of points and cameras needed for reconstruction? As we have seen, the number of unknowns is $3(m+n)-4$ for n points and m views. If all points are visible in all views, the number of constraints is $2mn$. A projective reconstruction is possible if the number of unknowns is equal to or smaller than the number of constraints, i. e.

$$2mn \geq 3(m+n)-4 \quad \text{or} \quad n \geq 2 - \frac{m-2}{2m-3} \ . \tag{3.21}$$

Since the method uses all projection constraints, two points outside the reference plane are sufficient for any ($m \geq 2$) number of views. In chapter 7 we will give a formal proof. This result is consistent with the investigation in sec. 2.4: The focus of expansion, which is the camera centre with unknown scale, can be determined from 2 image points. If more than two 3D points or views are used, the linear system is over-constrained.

A further application of the linear system is to **verify the consistency of the parallax geometry**. As a preprocessing step, 3D points on the reference plane, i.e. points which do not move on the stabilized images, have to be detected and removed from linear system. A necessary condition for a consistent parallax geometry is then that all minors in eqn. 3.17 of size $3(m+n)-4 \times 3(m+n)-4$ vanish. An equivalent more compact condition is that the 4th singular value is zero. In the presence of image noise this is never exactly true and therefore the ratio between the 5th and 4th singular value should be considered. We may state: *If the linear system is over-constrained, e.g. there are* 3 *points in* 2 *views, and the ratio between the* 5*th and* 4*th singular value is large, then the configuration is either consistent or critical*. Note, the 5th singular value is zero for critical configurations. However, as we will see in chapter 7, the number of critical configuration is very limited in practice. Such a consistency check could be used to detect moving 3D points in a rigid environment.

3.2.3 Multiple Views: Camera and Structure Constraints

The previous sec. 3.2.2 introduced the first approach for reconstructing multiple points visible in multiple views. This section reviews two other approaches based on camera constraints and the dual structure constraints. In contrast to the previous section, these

approaches apply to general and reference plane configurations. Since most practical reconstruction methods are based on camera constraints (see chapter 4), we will review them in detail. In the experiments (chapter 6), several camera constraints methods are compared with our direct reference plane method (sec. 3.2.2).

The readers who are familiar with these two approaches might skip this section. However, we point out that the review of reference plane configurations for both camera and structure constraints is less well known.

General configurations

The projection constraints in the general case (see eqn. 3.6) involve image measurements $\mathbf{x}$, 3D points $\mathbf{X}$ and camera centres $\mathbf{Q}$. Since 3D points and cameras have a bi-linear relationship, it is not possible to determine them directly from image measurements as in the multi-view reference plane case. In this section, we will derive constraints which involve only cameras and image measurements. On the basis of these constraints, the cameras can be derived linearly. Furthermore, we introduce dual constraints which involve only 3D points and image measurements. In this case, 3D points may be obtained linearly from the structure constraints. If either the cameras or the structure is known, the remaining unknown structure or cameras can be determined linearly. For simplicity, we investigate these constraints by using the special projective basis introduced in sec. 3.2.1, which is formed by reference points (Carlsson, 1995; Carlsson and Weinshall, 1998). An analysis independent of reference points has been presented by Triggs (1995) and Heyden (1998). In the following, only the main results about camera and structure constraints are summarized (see Hartley and Zisserman (2000) and Faugeras and Luong (2001) for an overview).

Consider a 3D point $\mathbf{X}$, projected into multiple cameras with centres $\bar{\mathbf{Q}}_1, \ldots, \bar{\mathbf{Q}}_m$ as 2D points $\mathbf{x}_1, \ldots, \mathbf{x}_m$. All $2m$ projection relations (3.6) may be written in the form:

$$\begin{pmatrix} w_1 A_1^{-1} & & -x_1 C_1^{-1} & (x_1 - w_1) D_1^{-1} \\ & w_1 B_1^{-1} & -y_1 C_1^{-1} & (y_1 - w_1) D_1^{-1} \\ & & \vdots & \\ w_m A_m^{-1} & & -x_m C_m^{-1} & (x_m - w_m) D_m^{-1} \\ & w_m B_m^{-1} & -y_m C_m^{-1} & (y_m - w_m) D_m^{-1} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ W \end{pmatrix} = 0 . \qquad (3.22)$$

The linear system in 3.22 can be used to reconstruct a 3D point from known cameras and image measurements. This represents the linear solution to the *intersection problem*. Since the 3D point is only unique up to scale, the rank of the left matrix in eqn. 3.22 has to be less then 3. This means that all 4×4 minors have to vanish, i.e. have to be zero. These minors give constraints on cameras and image measurements, which are the so-called **camera constraints** or **matching constraints**. The camera constraints have been the subject of many publications in the last decade (e.g. Faugeras, 1992; Hartley, 1992; Hartley, 1994; Shashua, 1994; Triggs, 1995; Hartley, 1995; Carlsson, 1995; Faugeras and Mourrain, 1995; Heyden, 1998). The first observation about camera constraints is that

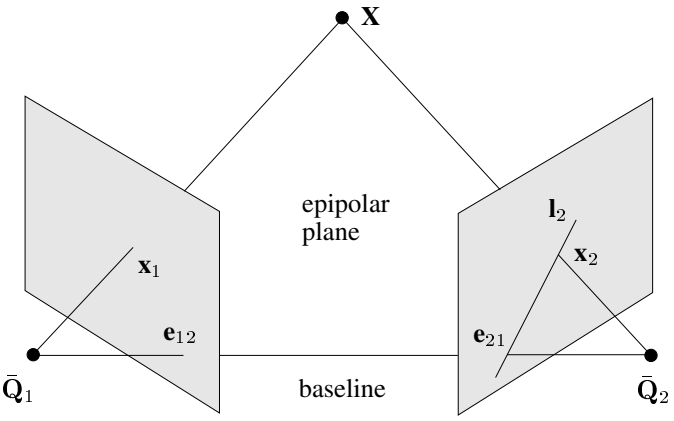


Figure 3.3. The geometry of two views and a 3D point **X**.

any constraint cannot involve more than 4 views, since any minor in eqn. 3.22 contains a maximum of 4 different rows. Furthermore, any constraint is bi-, tri-, or quadri-linear in the homogeneous image coordinates x_i, y_i, w_i. As a consequence, we will see that the constraints for $2, 3$ and 4 views can be written compactly in matrix or tensor form.

In the 2-view case, the left matrix in eqn. 3.22 is of size 4×4 and we obtain one constraint, which is called the **epipolar constraint**. This can be written in matrix form for the two image points $\mathbf{x}_1$ and $\mathbf{x}_2$ in camera 1 and 2 respectively as

$$\mathbf{x}_2^T \, F \, \mathbf{x}_1 \;=\; 0 \;. \tag{3.23}$$

The matrix F is denoted the **fundamental matrix**. The vector $\mathbf{l}_2 = F\,\mathbf{x}_1$ can be interpreted as a line in the second image (see fig. 3.3). It is denoted the **epipolar line** to point $\mathbf{x}_1$. The epipolar constraint says that the point $\mathbf{x}_2$ has to lie on $\mathbf{l}_2$, i.e. $\mathbf{x}_2^T \, \mathbf{l}_2 \;=\; 0$. Vice versa, $\mathbf{l}_1^T = \mathbf{x}_2^T F$ defines the epipolar line in image 1 to the point $\mathbf{x}_2$. The line connecting the two camera centres is called the **baseline**. Furthermore, the plane containing the two camera centres and an arbitrary 3D point **X** is called the **epipolar plane** of **X**. The projection of the second camera centre into the first image is denoted the **epipole** $\mathbf{e}_{12}$ and the projection of the first camera centre into the second image as the epipole $\mathbf{e}_{21}$. This means that they are defined as (see eqn. 2.29)

$$\begin{aligned} \mathbf{e}_{12} &\sim H_1^\infty \, (I \mid -\bar{\mathbf{Q}}_1)\, \mathbf{Q}_2 \sim H_1^\infty \, (\bar{\mathbf{Q}}_2 - \bar{\mathbf{Q}}_1) \quad \text{and} \\ \mathbf{e}_{21} &\sim H_2^\infty \, (I \mid -\bar{\mathbf{Q}}_2)\, \mathbf{Q}_1 \sim H_2^\infty \, (\bar{\mathbf{Q}}_2 - \bar{\mathbf{Q}}_1) \quad . \end{aligned} \tag{3.24}$$

Consequently, if $\bar{\mathbf{Q}}_1$ is chosen as $\mathbf{0}$, any camera i can be written as $P_i = [H_i^\infty \mid \lambda \mathbf{e}_{i1}]$. The epipoles can be obtained from the right nullspace of F and F^T:

$$F^T \, \mathbf{e}_{12} \;=\; 0 \quad \text{and} \quad F \, \mathbf{e}_{21} \;=\; 0 \;. \tag{3.25}$$

Furthermore, the epipolar constraint may be written in terms of the infinite homographies as

$$\mathbf{x}_2^T \, H_2^{\infty -T} \, [\bar{\mathbf{Q}}_2 - \bar{\mathbf{Q}}_1]_\times H_1^{\infty -1} \mathbf{x}_1 \; = \; 0 \; . \tag{3.26}$$

Note, the matrix $[\mathbf{a}]_\times$ defines the cross product: $[\mathbf{a}]_\times \mathbf{b} = \mathbf{a} \times \mathbf{b}$. It can be written as the skew-symmetric matrix

$$[\mathbf{a}]_\times = \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix} \; . \tag{3.27}$$

Eqn. 3.26 can be interpreted as follows. The vectors $\mathbf{x}_1' = H_1^{\infty -1} \mathbf{x}_1$ and $\bar{\mathbf{Q}}_2 - \bar{\mathbf{Q}}_1$ define the normal of the epipolar plane, i.e. $\mathbf{n} = \mathbf{x}_1' \times (\bar{\mathbf{Q}}_2 - \bar{\mathbf{Q}}_1)$. Eqn. 3.26 says that the vector $\mathbf{x}_2' = H_2^{\infty -1} \mathbf{x}_2$ has to be orthogonal to this plane, i.e. $\mathbf{x}_2'^T \, \mathbf{n} = 0$. This can be verified with fig. 3.3 and the fact that $\mathbf{x}_2'$ represents the vector between $\bar{\mathbf{Q}}_2$ and $\mathbf{X}$. From eqn. 3.26 we may derive a further property of the fundamental matrix. Since the skew-symmetric matrix $[\bar{\mathbf{Q}}_2 - \bar{\mathbf{Q}}_1]_\times$ is of rank 2, F has to be of rank 2 as well.

The fundamental matrix has 9 elements and 8 independent ratios of elements, since it is homogeneous. Therefore, 8 point matches are sufficient to determine the fundamental matrix linearly (e.g. Hartley, 1997). We have seen from eqn. 3.22 that the fundamental matrix comprises solely of camera parameters, i.e. the unknown camera centres. It can be written in tensor notation[2] and in terms of unknown camera matrices P_1 and P_2 as

$$F_{ij} = (-1)^{i+j} \begin{vmatrix} \tilde{P}_1^i \\ \tilde{P}_2^j \end{vmatrix} \; , \tag{3.28}$$

where $\tilde{P}_j^i$ is the camera matrix P_j without the ith row. If the fundamental matrix F is known and therefore the epipole $\mathbf{e}_{21}$ as well, the two cameras may be retrieved as

$$P_1 = (I \mid \mathbf{0}) \;\; \text{and} \;\; P_2 = ([\mathbf{e}_{21}]_\times F \mid \mathbf{e}_{21}) \; . \tag{3.29}$$

This definition was suggested by Luong and Viéville (1996) and is independent of the choice of reference points. Carlsson (1995) discusses how the cameras, i.e. camera centres, are derived from F for the case of a projective basis formed by reference points. If the cameras are known, the structure, i.e. 3D points, can be determined linearly by intersection (see eqn. 3.22). This was probably the first fundamental discovery (Faugeras, 1992; Hartley, 1992) in uncalibrated structure and camera recovery:

> *The structure (3D points) and two uncalibrated cameras can be determined (linearly) in projective space on the basis of image measurements only.*

Let us consider the minimum number of points needed to reconstruct the scene. The number of unknowns for n points and m views is $11m+3n-15$, since a camera has 11 degrees

[2] The understanding of tensor notation (Hartley and Zisserman, 2000) is not necessary for the understanding of the thesis.

of freedom and the projective space 15 degrees of freedom. The number of constraints derived from n 3D points visible in all m views is $2mn$. Since the number of constraints has to be equal or higher the number of unknowns, the following condition has to be satisfied:

$$2mn \geq 11m + 3n - 15 \quad \text{or} \quad n \geq 5 + \frac{m}{2m-3} \quad . \tag{3.30}$$

With 2-views a minimum of 7 points is sufficient. Furthermore, since F has 8 independent ratios of elements and consists of camera parameters only, it has to satisfy $8 - (2 \cdot 11 - 15) = 1$ extra constraint. As we have seen, this extra constraint is that F has rank 2, i.e. $det(F) = 0$. However, this is a non-linear constraint on the elements of F and therefore either one or three real solutions for F exist in the 7 point case.

Let us continue with the 3-view case. For three cameras and one 3D point, the left matrix in eqn. 3.22 is of size 6×4. This means that there are $\binom{6}{4} = 15$ possible minors of size 4×4, which have to be zero. However, it turns out, that there are only four linearly independent constraints which involve three views. These tri-linear constraints can be written, using a $3 \times 3 \times 3$ tensor $\mathcal{T}_i^{jk}$, as

$$\mathbf{x}_1^i \, \mathbf{x}_2^j \, \mathbf{x}_3^k \, \epsilon_{jqu} \, \epsilon_{krv} \, \mathcal{T}_i^{qr} \;=\; 0_{uv} \quad , \tag{3.31}$$

where $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ are the projections of a 3D point in camera $1, 2$ and 3 respectively. Note, only four of the nine tri-linear point relations are linearly independent. The tensor is called the **trifocal tensor** and encodes completely the geometry of 3 views. This means that $\mathcal{T}$ consists solely of camera parameters and may be written as

$$\mathcal{T}_i^{qr} = (-1)^{i+1} \begin{vmatrix} \tilde{P}_1^i \\ P_2^q \\ P_3^r \end{vmatrix} \quad , \tag{3.32}$$

where where P_j^i represents the ith row of camera matrix j. Note that with the trifocal tensor there is one distinguished view, in this case view 1. Equivalent to the fundamental matrix, the 3 cameras may be derived from a known trifocal tensor (e.g. Hartley and Zisserman, 2000). Since $\mathcal{T}$ has 26 independent ratios (27 elements minus an overall scale), it can be obtained linearly with eqn. 3.31 from the images of 7 3D points. Eqn. 3.30 says that $\mathcal{T}$ can be determined with a non-linear process from a minimum of 6 3D points. In the 6 points case, either one or three real solutions for $\mathcal{T}$ exist. Furthermore, $\mathcal{T}$ has to satisfy $26 - (3 \cdot 11 - 15) = 8$ extra constraints. On the basis of the trifocal tensor, the 3 pairs of fundamental matrices of the 3 cameras can be determined. One remaining question is: Can the trifocal tensor be replaced by the 3 pairs of fundamental matrices? It can be shown that the 3 F-matrices uniquely determine $\mathcal{T}$ (Hartley and Zisserman, 2000). However, the trifocal tensor is needed for the task of matching the images of a 3D point in 3 views. If a 3D point lies on a certain plane, the 3 epipolar constraints of each pair of views is satisfied but not the trifocal constraints of eqn. 3.31. This special plane is the **trifocal plane** defined by the 3 camera centres. According to eqn. 3.31, this result can be expected since 4 of the 9 constraints involve 3 image points and not only 2.

We have seen that the maximum number of views involved in a camera constraint is four. In the 4-view case the left matrix in eqn. 3.22 is of size 8×4. This gives $\binom{8}{4} = 70$ possible minors of size 4×4, which have to vanish. It has been shown, that there are *no* algebraically new constraints which involve 4 views. Therefore, the 4-view case is less important for matching and has raised less interest. As in the 2 and 3-view cases, a $3 \times 3 \times 3 \times 3$ tensor, the so-called **quadrifocal tensor**, can be used to express the quadrilinear relationship in the image coordinates:

$$\mathbf{x}_1^i \, \mathbf{x}_2^j \, \mathbf{x}_3^k \, \mathbf{x}_4^l \, \epsilon_{ipw} \, \epsilon_{jqx} \, \epsilon_{kry} \, \epsilon_{lsz} \, \mathcal{Q}^{pqrs} \;=\; 0_{wxyz} \;\; . \tag{3.33}$$

Eqn. 3.33 has 16 linearly independent equations. The minimum number of points to compute $\mathcal{Q}$ is 6 (see eqn. 3.30). In this case, the 81 elements of $\mathcal{Q}$ can be obtained linearly from eqn. 3.33. Furthermore, $\mathcal{Q}$ has to satisfy $80 - (4 \cdot 11 - 15) = 51$ extra constraints. The quadrifocal tensor encodes the geometry of 4 views and the camera matrices may be derived from it (e.g. Hartley and Zisserman, 2000). Furthermore, it can be specified in terms of camera matrices as

$$\mathcal{Q}^{pqrs} = \left| \begin{array}{c} P_1^p \\ P_2^q \\ P_3^r \\ P_4^s \end{array} \right| \;\; . \tag{3.34}$$

We have seen that the camera geometry may be determined linearly from image measurements in 2, 3 or 4 views. However, how do we reconstruct a scene from $n > 4$ views? One strategy is to divide the set of all images into subsets of a maximum of 4 views. After obtaining a projective reconstruction of each subset, they have to be merged to obtain *one* complete reconstruction. Different techniques for doing this are discussed in the next chapter. However, this strategy is obviously sub-optimal since not all cameras are considered simultaneously. One way to overcome this problem is to use the so-called **joint image closure constraints** introduced by Triggs (1997b). These constraints represent a bi-linear relationship between matching tensors and cameras, i.e. their projection matrices. This means that *all* cameras can be obtained directly and linearly from a set of known bi-, tri- or quadri-focal tensors. However, in order to use the tensors, they have to be scaled correctly. This is a non-trivial task especially for the case of missing data (Triggs, 1997b). A further well known technique is bundle-adjustment (Slama, 1980), which will be explained in the next chapter 4. Since it is based on non-linear optimization, a good initial reconstruction is necessary.

All geometric constraints we have discussed so far involved cameras and image measurements and were derived from the linear system in 3.22. If we interchange the role of 3D points and 3D camera centres we obtain a set of equations which is similar to 3.22. The projection of $n - 4$ 3D points in one camera gives $2(n - 4)$ projection relations (3.6) of

the form

$$\begin{pmatrix} w_5X_5 & & -x_5Z_5 & (x_5-w_5)W_5 \\ & w_5Y_5 & -y_5Z_5 & (y_5-w_5)W_5 \\ & & \vdots & \\ w_nX_n & & -x_nZ_n & (x_n-w_n)W_n \\ & w_nY_n & -y_nZ_n & (y_n-w_n)W_n \end{pmatrix} \begin{pmatrix} A^{-1} \\ B^{-1} \\ C^{-1} \\ D^{-1} \end{pmatrix} = 0 \ , \qquad (3.35)$$

where $\mathbf{x}_i$ represents the projection of the 3D point $\mathbf{X}_i$. Note, the first 4 points $\mathbf{X}_{1-4}$ were used for the projective basis. This linear system can be used to determine a camera from known 3D points, which was denoted the *resection problem*. Since the camera centre is only unique up to scale, all the 4×4 minors of the left matrix in 3.35 have to vanish. These minors consist of 3D point coordinates and image measurements only. Therefore, the constraints derived from the minors are denoted as **structure constraints**. More importantly, they are identical to the minors derived from eqn. 3.22 by the substitution: $(X, Y, Z, W) \leftrightarrow (A^{-1}, B^{-1}, C^{-1}, D^{-1})$. This leads to the following fundamental duality theorem (Carlsson, 1995):

Theorem 6 (Structure and camera duality) *The constraints for camera reconstruction from n points in m views are mathematically identical to the constraints for structure reconstruction from $m+4$ points in $n-4$ views.*

A consequence of the duality theorem is, that to all camera constraints and multi-view tensors there are dual structure constraints and structure tensors with the same properties. Therefore, any reconstruction algorithm for n points and m cameras can be used as well to reconstruct $m+4$ points and $n-4$ cameras. Furthermore, the 3D structure can be determined from known structure tensors. If the 3D structure is known, the cameras may be obtained be resection. For example, the *same* linear algorithm to compute the fundamental matrix from 2 views and n points (e.g. Hartley, 1997) can be used to compute the dual fundamental matrix, the so-called **G-matrix**, from 6 points and $n \geq 4$ views (Carlsson and Weinshall, 1998). On the basis of the G-matrix, the 6 3D points and consequently the n cameras can be reconstructed. The discussion on dualizing reconstruction algorithms (e.g. Hartley and Debunne, 1998) is continued in the next chapter. Obviously, the joint image closure constraints could be formulated on the basis of structure constraints as well. This would give the "joint structure closure constraints" which, however, have not been studied so far.

In the above discussion, camera, structure and closure constraints have been investigated for the for the general case of projective cameras. The same study can be carried out for more specific camera models or scenarios. The camera constraints and the corresponding tensors for *affine views* have been studied in (e.g. Bretzner and Lindeberg, 1998; Kahl and Heyden, 1999; Thorhallsson and Murray, 1999). The closure constraints for affine cameras were investigated by Kahl and Heyden (1999). Let us now consider the camera constraints for the special reference plane scenario.

Reference plane configurations

In the following we will briefly review the camera and structure constraints for the reference plane case. Probably the most complete study of the camera constraints in this case has been done by Triggs (2000). Furthermore, a thorough study of the dual structure constraints (and camera constraints) has been carried out by Irani et al. (1998) and Criminisi et al. (1998).

For simplicity, we assume that the infinite homography H_i^∞ of a camera i is non-singular and known so that it can be used to obtain calibrated translating cameras. The projection relations of this special camera type were derived in eqn. 3.13. Let us write these equations as in the general case (eqn. 3.22) for one 3D point $\mathbf{X}$ and m camera centres $\bar{\mathbf{Q}}_1, \ldots, \bar{\mathbf{Q}}_m$. For simplicity, we use only the first two constraints in 3.13 and obtain the following linear system, which consists of $2m$ equations:

$$\begin{pmatrix} -w_1 & & x_1 & w_1\bar{A}_1 - x_1\bar{C}_1 \\ & -w_1 & y_1 & w_1\bar{B}_1 - y_1\bar{C}_1 \\ & & \vdots & \\ -w_m & & x_m & w_m\bar{A}_m - x_m\bar{C}_m \\ & -w_m & y_m & w_m\bar{B}_m - y_m\bar{C}_m \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ W \end{pmatrix} = 0 \,. \tag{3.36}$$

Note, in this case the 3D point $\mathbf{X}$ can be written in homogeneous form. As in the general case, all the 4×4 minors for m-views have to vanish. However, the constraints derived from the minors are in this case significantly simpler. Eqn. 3.36 shows, that *any camera constraint is linear in the unknown coordinates of a camera centres* $\bar{\mathbf{Q}}_i$, i.e. $\bar{A}_i, \bar{B}_i$ or $\bar{C}_i$. As a consequence, these constraints can be used to solve for the unknown cameras from a linear system. Therefore, the complete projection matrix of a camera i is given as $P_i = H_i^\infty \, [I \mid -\bar{\mathbf{Q}}_i]$. If necessary, the multi-view tensors can be computed directly from the known cameras with eqns. 3.28, 3.32 and 3.34 respectively. Let us reconsider these ideas for the 2, 3 and 4 view cases in more detail.

In the 2-view case, eqn. 3.36 gives one constraint of the form

$$\begin{aligned} \bar{A}_1(w_1y_2 - y_1w_2) \quad &+ \bar{A}_2(y_1w_2 - w_1y_2) + \quad \bar{B}_1(x_1w_2 - w_1x_2) \; + \\ \bar{B}_2(w_1x_2 - x_1w_2) \quad &+ \bar{C}_1(y_1x_2 - x_1y_2) + \quad \bar{C}_2(x_1y_2 - y_1x_2) \; = 0 \; . \end{aligned} \tag{3.37}$$

Eqn. 3.37 shows an important property of camera constraints in the reference plane case. For any 3D point $\mathbf{X} = (X, Y, Z, 0)^T$ which lies on the reference plane, any camera constraint is satisfied independently of the camera centre. In the 2-view case, this can be verified from eqn. 3.37 by substituting $\mathbf{x}_1 \sim (X, Y, Z)^T \sim \mathbf{x}_2$. For 2 3D points, we obtain a linear system with two equations of the form (3.37):

$$M \; (\bar{A}_1, \bar{B}_1, \bar{C}_1, \bar{A}_2, \bar{B}_2, \bar{C}_2)^T = 0 \;\; \text{or compactly} \;\; M \, \mathbf{t} = 0 \; . \tag{3.38}$$

Since the reference plane fixes 11 of the 15 degrees of freedom of the projective space, the unknown cameras have $2 \cdot 3 - 4 = 2$ degrees of freedom. Therefore, 2 3D points in

general position are sufficient to determine 2 uncalibrated cameras in the reference plane case. This is the minimum number of 3D points, as can be seen from eqn. 3.21. The nullspace of M gives four linearly independent solutions for $\mathbf{t}$. These are the three trivial and the one non-trivial solution discussed in sec. 3.2.2. Eqn. 3.37 can be written more compactly as

$$\mathbf{x}_2^T \, [\bar{\mathbf{Q}}_{12}]_\times \, \mathbf{x}_1 \;=\; 0 \;\; , \tag{3.39}$$

where $\bar{\mathbf{Q}}_{ij} = \bar{\mathbf{Q}}_i - \bar{\mathbf{Q}}_j$. The derivation of eqn. 3.39 may be verified from eqn. 3.26 and the substitution $H_1^\infty = H_2^\infty = I$, i.e. calibrated translating cameras. Furthermore, from eqn. 3.24 we see that the translation vector $\bar{\mathbf{Q}}_{12}$ represents the epipoles, i.e. $\mathbf{e}_{12} \sim \mathbf{e}_{21} \sim \bar{\mathbf{Q}}_{12}$. The epipoles are as well denoted the focus of expansion $\mathbf{e}$ (see sec. 2.4). Therefore, the fundamental matrix for calibrated translating cameras may be written as

$$F \sim [\bar{\mathbf{Q}}_{12}]_\times \sim [\mathbf{e}_{12}]_\times \sim [\mathbf{e}_{21}]_\times \sim [\mathbf{e}]_\times \;\; . \tag{3.40}$$

As we have seen in fig. 2.9, the focus of expansion is defined by the projections of two distinct 3D points. If $\mathbf{x}_{ij}$ is the projection of a 3D point $\mathbf{X}_i$ into view j, we may write

$$\mathbf{e} = \mathbf{l}_1 \times \mathbf{l}_2 \;\text{ where }\; \mathbf{l}_1 = \mathbf{x}_{11} \times \mathbf{x}_{12} \;\text{ and }\; \mathbf{l}_2 = \mathbf{x}_{21} \times \mathbf{x}_{22} \;\; . \tag{3.41}$$

This is true if the image points $\mathbf{x}_{11}, \mathbf{x}_{12}$ and $\mathbf{x}_{21}, \mathbf{x}_{22}$ do not coincide and the epipolar lines $\mathbf{l}_1, \mathbf{l}_2$ are not collinear. These conditions mean that the two 3D points must not lie on the reference plane and that the two camera centres and the two 3D points are not coplanar. We will prove in chapter 7 that these configuration are actually the only critical configurations in the 2-view case.

In the 3-view case we obtain $\binom{6}{4} = 15$ minors of size 4×4 from eqn. 3.36, which have to vanish. According to Triggs (2000) these tri-linear constraints may be written as

$$(\mathbf{x}_1 \times \mathbf{x}_2) \, (\bar{\mathbf{Q}}_{13} \times \mathbf{x}_3)^T - (\bar{\mathbf{Q}}_{12} \times \mathbf{x}_2) \, (\mathbf{x}_1 \times \mathbf{x}_3)^T = \; 0_{3\times 3} \;\; , \tag{3.42}$$

where the first image is the distinguished view. It is straightforward to show that only 3 of the 9 constraints are linearly independent. Therefore, 2 3D points outside the reference plane and in "general position" are sufficient to determine linearly the $3 \cdot 3 - 4 = 5$ degrees of freedom of the cameras, and consequently the trifocal tensor. Furthermore, 2 3D points give even one additional, independent constraint, which is not necessary to determine the geometry. It can be used to verify the consistency of the geometry. We will return to this property later.

For 4 views, eqn. 3.36 gives $\binom{8}{4} = 70$ minors of size 4×4 which have to vanish. All the quadri-linear constraints may be written in tensor notation as

$$\mathbf{x}_1^i \, \mathbf{x}_2^j \, \mathbf{x}_3^k \, \mathbf{x}_4^l \, \epsilon_{ipw} \, \epsilon_{jqx} \, \epsilon_{kry} \, \epsilon_{lsz} \, (\, \epsilon_{qrs} \, \bar{\mathbf{Q}}_{14}^p \; - \; \epsilon_{prs} \, \bar{\mathbf{Q}}_{24}^q \; + \; \epsilon_{pqs} \, \bar{\mathbf{Q}}_{34}^r \,) \;=\; 0_{wxyz} \;\; . \tag{3.43}$$

In this case only 5 of the 81 constraints are linearly independent. As in the 2- and 3-view case, 2 3D points outside the reference plane and in "general position" are sufficient to solve for the remaining $4 \cdot 3 - 4 = 8$ degrees of freedom of the cameras using a linear system.

In summary, configurations with a known reference plane need 2 3D points outside the reference plane and in "general position" to obtain the $2, 3$ or 4 cameras linearly from the camera constraints. If necessary, the cameras can be used to derive the bi-, tri-, or quadri-focal tensor. However, the linearity condition of the camera constraints can be further exploited. Hartley et al. (2001) suggested to use the multi-view camera constraints, to compute *all* camera centres simultaneously from one linear system of equations. The constraints between each possible pair (3.39), triplet (3.42) or quadruplet (3.43) of views can be stacked into a linear system of the form:

$$M\ (\bar{A}_1, \bar{B}_1, \bar{C}_1, \ldots, \bar{A}_m, \bar{B}_m, \bar{C}_m)^T = 0 \ \text{ or compactly } \ M\,\mathbf{t} = 0\ . \tag{3.44}$$

The four dimensional nullspace of M gives *all* unknown cameras directly. However, in contrast to the reconstruction method in sec. 3.2.2, only the cameras and not the cameras and structure are reconstructed simultaneously. The original formulation of Hartley et al. (2001) uses tensor notation and does not stabilize the images, i.e. may be applied to finite and infinite cameras. The camera constraints in the general case for 2, 3 and 4 views, i.e. 3.23, 3.31 and 3.33, are linear in the elements of the respective tensor. Let us write a camera j as $P_j = (H_j^\infty \mid \mathbf{t}_j)$, where the infinite homography H_j^∞ is known. If we substitute P_j into the definition of a multi-view tensor, i.e. 3.28, 3.32 and 3.34, it is straightforward to show that any tensor element is a linear combination of the unknown camera parameters contained in $\mathbf{t}$. Consequently, multiple points give a linear system of the form 3.44, using either F-matrices, trifocal tensors or quadrifocal tensors. According to Hartley et al. (2001), the complexity of the linear system in 3.44 may be reduced by reducing the number of equations derived from each subset of views. For example, n points visible in a subset of 2 views give an over-constrained linear system of the form $n \times 6$. This linear system has a maximum of 6 linearly independent equations, which may be obtained by row reduction (see Hartley et al., 2001). Instead of the original n equations, these 6 equations may now be stacked into the linear system in 3.44.

The dual linear system to 3.36, which consists of one camera and multiple 3D points $\mathbf{X}_i$ projected as $\mathbf{x}_i$, may be written as

$$\begin{pmatrix} -w_1 & & x_1 & w_1\bar{X}_1 - x_1\bar{Z}_1 \\ & -w_1 & y_1 & w_1\bar{Y}_1 - y_1\bar{Z}_1 \\ & & \vdots & \\ -w_m & & x_m & w_m\bar{A}_m - x_m\bar{Z}_m \\ & -w_m & y_m & w_m\bar{B}_m - y_m\bar{Z}_m \end{pmatrix} \begin{pmatrix} A \\ B \\ C \\ D \end{pmatrix} \;=\; 0\,. \tag{3.45}$$

The two systems 3.36 and 3.45 are dual according to the substitution $(X, Y, Z, W) \leftrightarrow (A, B, C, D)$. Consequently, the duality theorem in the reference plane case is simpler than in the general case.

Theorem 7 (Structure and camera duality – reference plane) *The constraints for camera reconstruction from n points in m views are mathematically identical to the constraints for structure reconstruction from m points in n views by substituting $\boldsymbol{X}$ with $\boldsymbol{Q}$.*

Using the duality theorem, the dual structure constraints may be derived directly from the camera constraints by substitution. For example, in the 2-view case the structure constraint dual to 3.39 is

$$\mathbf{x}_2^T \left[\bar{\mathbf{X}}_1 - \bar{\mathbf{X}}_2\right]_\times \mathbf{x}_1 \;=\; 0 \;. \tag{3.46}$$

Furthermore, the **dual epipole** $\mathbf{e}^d$ of the two distinct 3D points $\bar{\mathbf{X}}_1$ and $\bar{\mathbf{X}}_2$ can be computed with formula 3.41. On the basis of the duality theorem, we may conclude that all structure constraints for 2, 3 and 4 points are linear in the 3D point coordinates. This means that the 3D points can be reconstructed from a minimum of 2 views. If necessary, the dual tensors can be derived from the known 3D points. As for camera constraints, the structure constraints may as well be used to compute the complete 3D point structure linearly. However, a significant disadvantage of the dual system 3.45 is that the structure constraints are only valid for points outside the reference plane. The same applies to the dual counterpart 3.36, which is only valid for camera centres outside the reference plane. This is not surprising, a requirement for reconstructing 2 points in 2 views, with e.g. the dual epipolar constraint 3.46, is that the two points lie outside the reference plane. Therefore, the structure constraints have to be used carefully for the task of structure recovery, especially for scenarios where the 3D points are "close to" or on the reference plane. Furthermore, in general the number of 3D points is larger than the number of unknown cameras. This means that a complete linear system to obtain the unknown structure is larger in the dual case.

Irani et al. (1998) and Weinshall et al. (1998) computed the dual fundamental matrix from two 3D reference points *outside* the reference plane. On the basis of this, they derived a formula to compute directly the relative height of a third 3D point from the reference plane. In their representation the reference plane is *not* moved to infinity, which is good for "nearly flat" scenes. This idea is based on earlier work by Sawhney (1994), Kumar et al. (1994) and Kumar et al. (1995), which study the task of shape recovery from projective and affine cameras using plane+parallax. A similar direct formula of height computation from 3 3D points has been suggested by Criminisi et al. (1998). It is based on properties of the planar homology, which will be introduced later. Such height measurements can be obtained as well from a 3D reconstruction where the reference plane is at infinity, e.g. using the linear system in 3.17. In this case, the 3D reconstruction has to be transformed so that the reference plane is a finite plane, e.g. $Z = 0$. Furthermore, the correct Euclidean height of the 3D points can be derived if the Euclidean heights of the two reference points are known and additionally some affine measurements on the world plane, e.g. two parallel lines on the plane (Weinshall et al., 1998; Criminisi et al., 1998).

The camera and structure constraints for 2 and 3 views in the reference plane case have been studied as well in (Irani and Anandan, 1996; Irani et al., 1998; Weinshall et al., 1998; Criminisi et al., 1998). Irani et al. (1998) and Criminisi et al. (1998) gave them a concrete physical meaning by projecting the scene onto the reference plane, which is equivalent to stabilizing the images. Figure 3.4 shows the geometry of 3 points $\mathbf{P}, \mathbf{Q}, \mathbf{R}$ in 2 views, where the stabilized images are superimposed. The three parallax vectors $\mathbf{p}_1 - \mathbf{p}_2, \mathbf{q}_1 - \mathbf{q}_2$

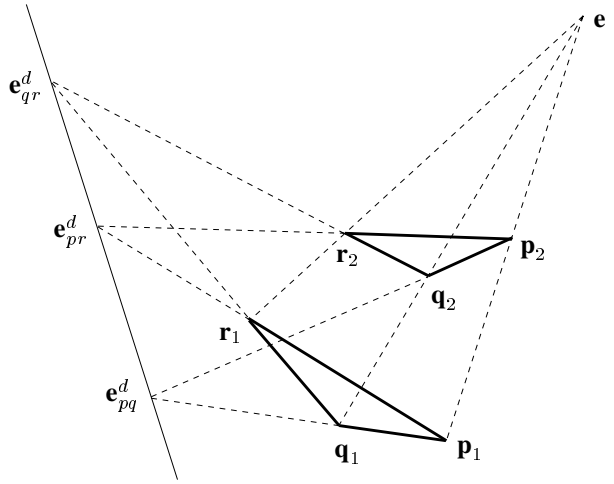


Figure 3.4. The images of three 3D points **P**, **Q** and **R** in two views define a Desargues configuration.

and $\mathbf{r}_1 - \mathbf{r}_2$ intersect in the epipole $\mathbf{e}$. The intersection of two pairs of points $\mathbf{p}_1 - \mathbf{q}_1$ and $\mathbf{p}_2 - \mathbf{q}_2$ define the dual epipole $\mathbf{e}^d_{pq}$. The three dual epipoles $\mathbf{e}^d_{pq}$, $\mathbf{e}^d_{pr}$ and $\mathbf{e}^d_{qr}$ lie on a line. This is true since the 6 image points define a **Desargues configuration** (Semple and Kneebone, 1952). Criminisi et al. (1998) showed that the mapping: $\mathbf{p}_1 \rightarrow \mathbf{p}_2$, $\mathbf{q}_1 \rightarrow \mathbf{q}_2$ and $\mathbf{r}_1 \rightarrow \mathbf{r}_2$ defines a planar homology (Semple and Kneebone, 1952). Since 3 3D points over-constrain the homology, which has 5 degrees of freedom, this mapping has to satisfy certain "homology constraints", which verify the parallax geometry. This means, that the image measurements of 3 points in 2 views or dually 2 points in 3 views can be used directly to check the **consistency of the parallax geometry**. This is not surprising since 2 parallax vectors, i.e. $\mathbf{p}_1 - \mathbf{p}_2, \mathbf{q}_1 - \mathbf{q}_2$, define explicitly the epipole $\mathbf{e}$, which has to lie on the third parallax vector, i.e. $\mathbf{r}_1 - \mathbf{r}_2$, (see fig. 3.4). However, the advantage of a direct check on the homology is that it is not necessary to compute the epipole whose location might be ill-conditioned. Irani and Anandan (1996) introduced an alternative formula to check directly the consistency of the parallax geometry. Such a consistency check can for instance be used to detect moving objects in a rigid scene. We have seen in sec. 3.2.2 that the consistency of the parallax geometry can as well be checked by a singular value analysis of the linear system in 3.17. A further useful application of the simplified camera and structure constraints in the reference plane case is the direct generation of novel views, i.e. **new view synthesis**, as shown in (Irani et al., 1998; Irani et al., 2002). However, this topic is beyond the scope of this thesis.

A discussion of the closure constraints, as in the general case, is not necessary when a reference plane is available. The cameras of $m \geq 4$ views (or $n \geq 4$ 3D points) can be determined simultaneously and linearly from the camera constraints (or the structure constraints).

3.2.4 Multiple Views: Factorization of Cameras and Points

This section reviews a fourth approach for reconstructing multiple points visible in multiple views. Since factorization methods are important for practical applications (see chapter 4), the following discussion is detailed. It contains well known methods for both general and reference plane configurations and may be skipped by readers familiar with the topic.

Consider all image points being collected into a large measurement matrix. A 3D point is projected onto an image point by multiplying it with the respective camera (see eqn. 3.1). Therefore, the measurement matrix is the product of a matrix consisting of all cameras and a matrix consisting of all 3D points. The basic idea of the factorization approach is to factorize, decompose, the known measurement matrix into the unknown camera and point matrices. This already shows the main drawback of all factorization methods, to factorize the measurement matrix *all* image data has to be available. The assumption of all 3D points being visible in all views is a major restriction in practice. Several methods to overcome this limitation have been suggested in the literature and will be discussed in chapter 4.

As in the previous sections we consider projective cameras first. Although, factorization for affine cameras has been introduced before factorization for projective cameras.

General configurations

Let us assume that all 3D points are visible in all views, i.e. there is no missing data. The projection relations (eqn. 3.1) of n 3D points $\mathbf{X}_i$ and m cameras P_j can be written compactly in the form

$$\begin{pmatrix} \lambda_{11}\mathbf{x}_{11} & \dots & \lambda_{1n}\mathbf{x}_{1n} \\ \vdots & \ddots & \vdots \\ \lambda_{m1}\mathbf{x}_{m1} & \dots & \lambda_{mn}\mathbf{x}_{mn} \end{pmatrix} = \begin{pmatrix} P_1 \\ \vdots \\ P_m \end{pmatrix} (\mathbf{X}_1, \dots, \mathbf{X}_n) \quad \text{or} \quad W = P\ X \ . \tag{3.47}$$

The matrix W depends only on the unknown scales λ_{ij} (projective depths) and known image measurements and is therefore called the **unscaled measurement matrix**. Let us assume that by some means the unknown scales are determined. The unscaled measurement matrix is then denoted the **measurement matrix**. The measurement matrix has the fundamental property that its rank is 4, since it is the product of two rank 4 matrices. As a consequence, the unknown structure and cameras may be obtained directly from W by factorization. Explicitly, the singular value decomposition (SVD) of $W = UDV^T$ gives

$$P = [\sigma_1\mathbf{u}_1 \ \ \sigma_2\mathbf{u}_2 \ \ \sigma_3\mathbf{u}_3 \ \ \sigma_4\mathbf{u}_4] \quad \text{and} \quad X = [\mathbf{v}_1\ \mathbf{v}_2\ \mathbf{v}_3\ \mathbf{v}_4]^T \ , \tag{3.48}$$

where $\mathbf{u}_i$ and $\mathbf{v}_i$ represents the ith column in U and V respectively and σ_i is the ith largest singular value in D. Note, since W is of rank 4 only the first 4 singular values in D are different from zero. The space of all projectively equivalent solutions can be obtained by a transformation $H_{4\times4}$, since $W = P\ X = PH^{-1}HX$. This idea of projective structure and camera recovery is due to Sturm and Triggs (1996) under the name of **projective factorization**. Various ways to compute the projective depths have been suggested and will be discussed in chapter 4.

The original idea of "factorizing" the structure and camera recovery problem was presented by Tomasi and Kanade (1992) for the case of affine cameras. Let us reconsider the non-homogeneous mapping 2.31 of a 3D point $\bar{\mathbf{X}}_i$ by an affine camera P_j onto the image point $\bar{\mathbf{x}}_{ij}$:

$$\bar{\mathbf{x}}_{ij} = M_j \bar{\mathbf{X}}_i + \mathbf{t}_j \ . \tag{3.49}$$

The first observation is that the camera vectors $\mathbf{t}_j$ may be derived directly from those image points which are represented in all views. A property of affine cameras is, that they map the centroid of a set of 3D points onto the centroid of their projections. Let us define $\mathbf{t}_j = \frac{1}{n}\sum_{i=1}^{n} \bar{\mathbf{x}}_{ij}$ as the centroid of all image points in view j. By moving all the image points in each view, the vector $\mathbf{t}_j$ can be eliminated, i.e. $\mathbf{t}_j = (0,0)^T$. The transformed image points are then $\bar{\mathbf{x}}'_{ij} = \bar{\mathbf{x}}_{ij} - \mathbf{t}_j$. This means that the origin of the 3D space is projected onto $\mathbf{t}_j$, i.e. the origin of the transformed images. As in the projective case, the transformed image points may now be stacked into a measurement matrix of the form

$$\begin{pmatrix} \bar{\mathbf{x}}'_{11} & \dots & \bar{\mathbf{x}}'_{1n} \\ \vdots & \ddots & \vdots \\ \bar{\mathbf{x}}'_{m1} & \dots & \bar{\mathbf{x}}'_{mn} \end{pmatrix} = \begin{pmatrix} M_1 \\ \vdots \\ M_m \end{pmatrix} \left(\bar{\mathbf{X}}_1, \dots, \bar{\mathbf{X}}_n\right) \ \text{ or } \ W = M\ X \ . \tag{3.50}$$

However, in contrast to the general projective case, the unknown projective depths disappear. The structure and cameras can now be determined as a rank 3 factorization. Explicitly, if $W = UDV^T$ is the Singular Value Decomposition of W, the affine cameras and 3D points are given as

$$M = [\sigma_1 \mathbf{u}_1 \ \ \sigma_2 \mathbf{u}_2 \ \ \sigma_3 \mathbf{u}_3] \quad \text{and} \quad X = [\mathbf{v}_1 \ \mathbf{v}_2 \ \mathbf{v}_3]^T \ . \tag{3.51}$$

This reconstruction method is known as **affine factorization**. As in the projective case, the space of all affine equivalent solutions is obtained by a matrix $A_{3\times 3}$, since $W = M\ X = MA^{-1}AX$. Note, the origin of the 3D affine space was already fixed.

Reference plane configurations

For reference plane configurations, Triggs (2000) suggested an alternative factorization method, which is more efficient than the projective version. Let us reconsider the projection relation 3.9 of a 3D point $\mathbf{X}_i$ onto the image point $\mathbf{x}_{ij}$ for a calibrated translating camera with centre $\bar{\mathbf{Q}}_j$:

$$\lambda_{ij}\, \mathbf{x}_{ij} \ = \ (\, I \mid -\bar{\mathbf{Q}}_j \,)\, \mathbf{X}_i \ = \ \mathbf{X}'_i - \bar{\mathbf{Q}}_j W_i \ , \tag{3.52}$$

where $\mathbf{X}_i = (\mathbf{X}'_i, W_i)^T$. Let us assume that the projective depths λ_{ij} have been determined in a pre-processing step as in the projective case. We may choose the origin of the projective space as the centroid of all camera centres[3]:

$$\mathbf{0} = \frac{1}{m}\sum_{j=1}^{m} \bar{\mathbf{Q}}_j \ . \tag{3.53}$$

[3]The weighting of the camera centres may be chosen differently.

A first observation is that $\mathbf{X}'_i$ can be determined directly from image measurements. If the point $\mathbf{X}_i$ is visible in all views, we may compute $\mathbf{X}'_i = \frac{1}{m}\sum_{j=1}^{m}\lambda_{ij}\mathbf{x}_{ij}$, since

$$\frac{1}{m}\sum_{j=1}^{m}\lambda_{ij}\mathbf{x}_{ij} = \frac{1}{m}\sum_{j=1}^{m}(\mathbf{X}'_i - \bar{\mathbf{Q}}_j W_i) = \frac{1}{m}\sum_{j=1}^{m}\mathbf{X}'_i - \frac{1}{m}\sum_{j=1}^{m}\bar{\mathbf{Q}}_j W_i = \frac{1}{m}\sum_{j=1}^{m}\mathbf{X}'_i = \mathbf{X}'_i \ . \tag{3.54}$$

With known $\mathbf{X}'_i$, new image points $\mathbf{x}'_{ij}$ may be derived from eqn. 3.52 as

$$\mathbf{x}'_{ij} = \mathbf{X}'_i - \lambda_{ij}\mathbf{x}_{ij} = \bar{\mathbf{Q}}_j W_i \ . \tag{3.55}$$

These new image points may now be used to obtain the measurement matrix

$$\begin{pmatrix} \mathbf{x}'_{11} & \cdots & \mathbf{x}'_{1n} \\ \vdots & \ddots & \vdots \\ \mathbf{x}'_{m1} & \cdots & \mathbf{x}'_{mn} \end{pmatrix} = \begin{pmatrix} \bar{\mathbf{Q}}_1 \\ \vdots \\ \bar{\mathbf{Q}}_m \end{pmatrix} (W_1, \ldots, W_n) \ . \tag{3.56}$$

Consequently, all camera centres and unknown 3D point "depths" W_i may be derived from a rank 1 factorization of the measurement matrix. Since the centre of projection is already fixed, the remaining ambiguity in the projective space is a simple scaling: $\bar{\mathbf{Q}}_j \rightarrow \mu\bar{\mathbf{Q}}_j$ and $W_i \rightarrow W_i/\mu$. One advantage of this **plane+parallax factorization** method, in contrast to the direct reference plane method (see sec. 3.2.2), is that points on and off the reference plane are reconstructed simultaneously. As already discussed, this is advantageous for "nearly flat" scenes.

3.3 Lines

The relationship between multiple cameras and 3D lines will be discussed in the same way as for point features. Since lines are less frequently used for 3D reconstruction than points, this discussion is shorter than in the previous sec. 3.2. We begin the investigation with a comparison of general configurations and reference plane configurations in the single view case (sec. 3.3.1). As for 3D points, the novel observation is that the relationship between cameras and 3D lines is *bi-linear* in the general case and *linear* in the reference plane case. Furthermore, we will show that 2 parameters of a 3D line (its orientation) may be derived from a given reference plane. This leads to three different approaches of writing the relationship between multiple 3D lines observed in multiple cameras. First, as a single linear system consisting of image lines only (sec. 3.3.2), secondly, as camera constraints including cameras and image lines (sec. 3.3.3) and thirdly, as a large image line measurement matrix (sec. 3.3.4). The first approach is a novel contribution and not part of any of our previous publications. We call it the *direct reference plane approach* for lines (Line-DRP). The last two approaches are known and reviewed here for both general and reference plane configurations. Note that cameras and 3D lines do not have a dual relationship which means that there are no structure constraints including 3D lines and image lines. Furthermore, practical algorithms of the Line-DRP approach are outlined in sec. 6.2.1.

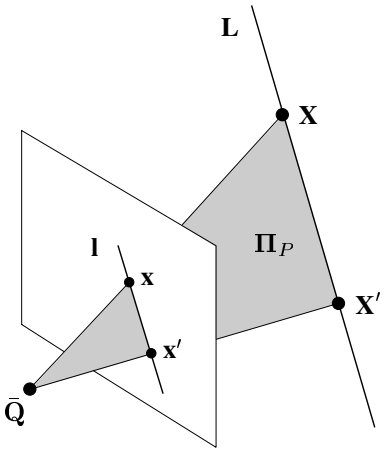


Figure 3.5. The projection of a 3D line **L**, which is represented by the two points **X** and $\mathbf{X}'$, on the image line **l** by a camera with centre $\bar{\mathbf{Q}}$. The plane $\mathbf{\Pi}_P$ is defined as $\mathbf{\Pi}_P = P^T\,\mathbf{l}$.

3.3.1 Single View: General versus Reference Plane

General configurations

Unlike our approach for points, when using line features we will not choose a specific projective basis in the world $\mathcal{P}^3$ and in the image plane $\mathcal{P}^2$. A camera will be represented by a general 3×4 matrix P. Section 2.1.2 introduced several representations of a 3D line. Represent a 3D line by two distinct 3D points **X** and $\mathbf{X}'$ (see fig. 3.5). The two points are projected into camera P as $\mathbf{x} \sim P\,\mathbf{X}$ and $\mathbf{x}' \sim P\,\mathbf{X}'$. The condition that the 3D points lie on the 3D line **L** can be expressed as

$$\begin{aligned} \mathbf{x}^T\,\mathbf{l} = 0 \quad &\text{or} \quad (P\,\mathbf{X})^T\,\mathbf{l} = 0 \quad \text{and} \\ \mathbf{x}'^T\,\mathbf{l} = 0 \quad &\text{or} \quad (P\,\mathbf{X}')^T\,\mathbf{l} = 0\ . \end{aligned} \tag{3.57}$$

This shows, that the relation between a 3D line (represented by two 3D points) and a camera is *bi-linear*. The projection relation 3.57 for e.g. point **X** can be rewritten as well as

$$\mathbf{X}^T\,P^T\,\mathbf{l} = 0 \quad \text{or} \quad \mathbf{X}^T\,\mathbf{\Pi}_P = 0\ , \tag{3.58}$$

which means that $P^T\mathbf{l}$ represents the plane $\mathbf{\Pi}_P$. We will see that in the multiple view case extra constraints are needed to specify the two distinct 3D points of a line uniquely in space.

If a 3D line is represented by a Plücker matrix L, it can be shown (Hartley and Zisserman, 2000) that it projects onto the image line **l** by the camera P as

$$\lambda\,[\mathbf{l}]_\times = P\,L\,P^T = M\ . \tag{3.59}$$

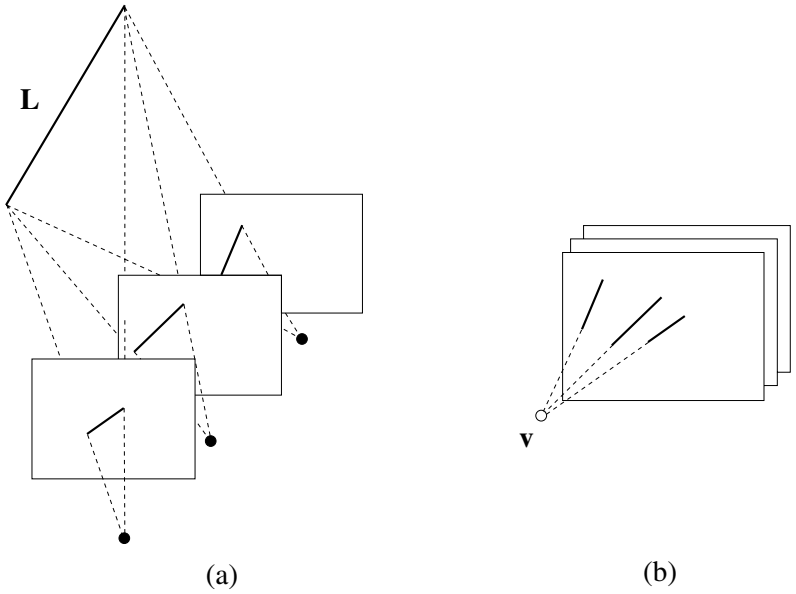


Figure 3.6. The projections of a 3D line **L** onto three cameras (a) intersect in the vanishing point **v** (b), which is the projection of the direction of the line.

If we eliminate the unknown scale λ in eqn. 3.59, we obtain 3 projection relations of the form

$$\begin{aligned} l_1 M_{13} + l_2 M_{23} &= 0 \\ l_1 M_{12} - l_3 M_{23} &= 0 \\ l_2 M_{12} - l_3 M_{13} &= 0 \ . \end{aligned} \tag{3.60}$$

Obviously, only two of the three projection relations are linearly independent since an image line has 2 degrees of freedom. As in the previous case, the coordinates of a 3D line and the camera are bi-linear related. In this case it is even quadratic in the elements of the camera matrix. For instance the element M_{12} is

$$\begin{aligned} M_{12} &= L_{12}(P_{11}P_{22} - P_{21}P_{12}) + L_{13}(P_{11}P_{32} - P_{31}P_{12}) + L_{14}(P_{11}P_{42} - P_{41}P_{12}) \\ &+ L_{23}(P_{21}P_{32} - P_{31}P_{22}) + L_{42}(P_{41}P_{22} - P_{21}P_{42}) + L_{34}(P_{31}P_{42} - P_{41}P_{32}) \quad . \end{aligned} \tag{3.61}$$

Reference plane configurations

With a reference plane the infinite homography H^∞ between the reference plane and a certain view is known. For simplicity, we assume that H^∞ is a non-singular matrix which can be used to stabilize the image points, as $\mathbf{x}' \sim H^{\infty -1}\mathbf{x}$. Note that this restriction is relaxed in sec. 6.2.1. The stabilizing process gives a calibrated camera of the form

$P = [I \mid -\bar{\mathbf{Q}}]$ (see sec. 2.4). According to proposition 2 (sec. 2.1.1), an image line $\mathbf{l}$ is stabilized as $\mathbf{l}' \sim H^T \mathbf{l}$. In the following, the stabilized images are used as input images and therefore are the dashes dropped, i.e. $\mathbf{l}$ instead of $\mathbf{l}'$.

As in the general case, we begin by representing a 3D line $\mathbf{L}$ by two distinct 3D points $\mathbf{X}$ and $\mathbf{X}'$ (see fig. 3.5). The constraint that a 3D point lies on $\mathbf{L}$ may in this case be written as

$$\begin{array}{lll} \mathbf{x}^T \mathbf{l} = 0 & \text{or} & \bar{\mathbf{X}}^T \mathbf{l} - \bar{\mathbf{Q}}^T \mathbf{l} = 0 \quad \text{and} \\ \mathbf{x}'^T \mathbf{l} = 0 & \text{or} & \bar{\mathbf{X}}'^T \mathbf{l} - \bar{\mathbf{Q}}^T \mathbf{l} = 0 \ , \end{array} \tag{3.62}$$

where $\mathbf{x} \sim P\,\mathbf{X}, \mathbf{x}' \sim P\,\mathbf{X}'$ and $\mathbf{l}$ is the projection of $\mathbf{L}$. The main difference to the general case is, that the relationship between a 3D line (represented by two 3D points) and a camera is now *linear*. Eqn. 3.62 holds if both the camera centre and the two 3D points do not lie on the plane at infinity. In particular, the 3D line, i.e. both 3D points, should not lie on the plane at infinity. This special case will be considered later.

The requirement that neither 3D point $\mathbf{X}, \mathbf{X}'$ lies on the reference plane limits the applicability of the linear projection constraints in eqn. 3.62. In order to increase the scope, we have to consider in more detail a 3D line seen in several views in more detail. Figure 3.6(a) shows three translating cameras which observe one 3D line $\mathbf{L}$. The 3 images are superimposed in fig. 3.6(b). We saw in sec. 2.4 that points at infinity are stationary in the superimposed images. Therefore, the 3 image lines have to intersect in one image point $\mathbf{v}$ which is the vanishing point of the line, i.e. the projection of the point at infinity $\mathbf{V}$ of the line. The infinite point $\mathbf{V}$ represents as well the direction of the line. In the multi-view case, $\mathbf{v}$ may be determined linearly from the image lines $\mathbf{l}_1, \ldots \mathbf{l}_m$ by the linear system

$$\begin{pmatrix} \mathbf{l}_1^T \\ \vdots \\ \mathbf{l}_m^T \end{pmatrix} \mathbf{v} = 0 \quad \text{or} \quad M\,\mathbf{v} = 0 \ . \tag{3.63}$$

The 1-dimensional nullspace of M gives the correct solution, provided the image point is unique. Note that the nullspace is 2-dimensional if all image lines are collinear, i.e. the 3D line lies on the plane at infinity. From the image lines of several views it is possible to determine the point at infinity of the line $\mathbf{L}$ as $\mathbf{V} = (\mathbf{v}, 0)^T$. Note that for calibrated translating cameras the vanishing point $\mathbf{v}$ in the image is identical to the direction of the 3D line in space. Consequently, $\mathbf{V}$ may be chosen as one 3D point $\mathbf{X}'$ of the line. Therefore, it is sufficient to reconstruct only the point $\mathbf{X}$, in order to determine the line $\mathbf{L}$ completely. Given $\mathbf{V}$ and assuming that $\mathbf{L}$ does not lie on the plane at infinity yields a further advantage, namely that we know that any other point $\mathbf{X}$ on $\mathbf{L}$ cannot be infinite, i.e. eqn. 3.62 is valid.

However, this is an over-parameterization since $\mathbf{L}$ has 2 degrees of freedom once $\mathbf{V}$ is known, and $\mathbf{X}$ has 3 degrees of freedom. Let us derive a minimal representation of $\mathbf{L}$. Since the direction $\mathbf{v}$ of the 3D line is known, we may derive the normals $\mathbf{n}$ and $\mathbf{n}'$ of two different planes $\mathbf{\Pi}, \mathbf{\Pi}'$ (see fig. 3.7). The linear system

$$\mathbf{v}^T \mathbf{n} = 0 \quad \text{or} \quad M\,\mathbf{n} = 0 \tag{3.64}$$

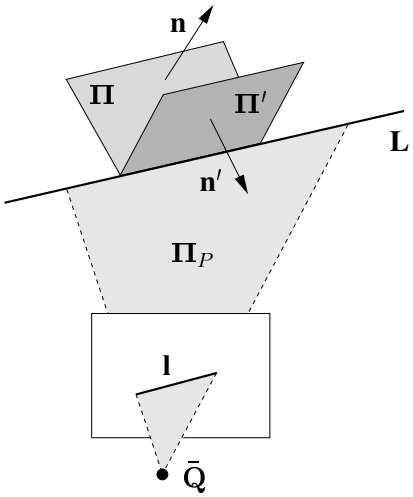


Figure 3.7. The 3 planes $\mathbf{\Pi}, \mathbf{\Pi}', \mathbf{\Pi}_P$ must intersect in the 3D line $\mathbf{L}$.

has a 2-dimensional space of solutions $\mathbf{n}, \mathbf{n}'$, which can be obtained from the nullspace of M. Therefore, any 3D line $\mathbf{L}$ may be uniquely described be the two planes $\mathbf{\Pi} = (\mathbf{n}, d)^T$ and $\mathbf{\Pi}' = (\mathbf{n}', d')^T$. This representation of the line is minimal since d and d' are the only unknown parameters. The condition that $\mathbf{L}$ projects onto $\mathbf{l}$ is equivalent to the constraint that the 3 planes $\mathbf{\Pi}, \mathbf{\Pi}', \mathbf{\Pi}_P$ intersect uniquely in the line $\mathbf{L}$ (see fig. 3.7). The plane $\mathbf{\Pi}_P$ is, according to eqn. 3.58, $\mathbf{\Pi}_P = P^T \mathbf{l} = (\mathbf{l}, -\bar{\mathbf{Q}}^T \mathbf{l})^T$. Let us stack the planes into a 4×3 matrix

$$M = [\mathbf{\Pi}_P \; \mathbf{\Pi} \; \mathbf{\Pi}'] = \begin{pmatrix} \mathbf{l}_x & \mathbf{n}_x & \mathbf{n}'_x \\ \mathbf{l}_y & \mathbf{n}_y & \mathbf{n}'_y \\ \mathbf{l}_z & \mathbf{n}_z & \mathbf{n}'_z \\ -\bar{\mathbf{Q}}^T\mathbf{l} & d & d' \end{pmatrix} . \tag{3.65}$$

Algebraically, the constraint that the three planes intersect in $\mathbf{L}$ means that the rank of M is 2. This can be proved by considering all points $\mathbf{X}$ which lie on the 3D line $\mathbf{L}$. A 3D point $\mathbf{X}$ lies on all three planes if $M^T \mathbf{X} = 0$. Since a 3D line forms a one-dimensional subspace of $\mathcal{P}^3$, M^T has a 2 dimensional nullspace. This means that M is of rank 2. Therefore, the 4 subdeterminants of size 3×3 have to be zero. Those 3 determinants which involve unknown parameters give three constraints of the form:

$$\begin{vmatrix} \mathbf{l}_x & \mathbf{n}_x & \mathbf{n}'_x \\ \mathbf{l}_y & \mathbf{n}_y & \mathbf{n}'_y \\ -\bar{\mathbf{Q}}^T\mathbf{l} & d & d' \end{vmatrix} = 0 , \quad \begin{vmatrix} \mathbf{l}_x & \mathbf{n}_x & \mathbf{n}'_x \\ \mathbf{l}_z & \mathbf{n}_z & \mathbf{n}'_z \\ -\bar{\mathbf{Q}}^T\mathbf{l} & d & d' \end{vmatrix} = 0 , \quad \begin{vmatrix} \mathbf{l}_y & \mathbf{n}_y & \mathbf{n}'_y \\ \mathbf{l}_z & \mathbf{n}_z & \mathbf{n}'_z \\ -\bar{\mathbf{Q}}^T\mathbf{l} & d & d' \end{vmatrix} = 0 . \tag{3.66}$$

The main observation is that the constraints are *linear* in the unknown camera centre $\bar{\mathbf{Q}}$ and the unknown 3D line parameters d and d'.

An alternative derivation of these constraints is based on the dual Plücker matrix representation L^* of a 3D line. Let us represent the 3D line with the two planes $\mathbf{\Pi}, \mathbf{\Pi}'$ as $L^* = \mathbf{\Pi}\,\mathbf{\Pi}'^T - \mathbf{\Pi}'\,\mathbf{\Pi}^T$. It can be verified that only 6 elements, $L^*_{1-3,4}$ and $L^*_{4,1-3}$, contain the unknown parameters d or d'. We may transform L^* into L according to eqn. 2.12. As in the general case (see eqn. 3.59), the projection of a line $\mathbf{L}$ in a camera $P = [I \mid -\bar{\mathbf{Q}}]$ is given as

$$\lambda\,[\mathbf{l}]_\times = [I \mid -\bar{\mathbf{Q}}]\,L\,[I \mid -\bar{\mathbf{Q}}]^T \quad . \tag{3.67}$$

Explicitly, this gives the three equations

$$\begin{aligned} dn'_x - d'n_x + \bar{B}\,(n_x n'_y - n'_x n_y) + \bar{C}\,(n_x n'_z - n'_x n_z) &= \lambda\,\mathbf{l}_x \\ dn'_y - d'n_y + \bar{A}\,(n'_x n_y - n_x n'_y) + \bar{C}\,(n_y n'_z - n'_y n_z) &= \lambda\,\mathbf{l}_y \\ dn'_z - d'n_z + \bar{A}\,(n'_x n_z - n_x n'_z) + \bar{B}\,(n'_y n_z - n_y n'_z) &= \lambda\,\mathbf{l}_z \quad , \end{aligned} \tag{3.68}$$

where $\bar{\mathbf{Q}} = (\bar{A}, \bar{B}, \bar{C})^T$. λ may be eliminated by taking ratios of these equations, yielding three constraints of which two are independent. Similar to eqn. 3.66, these constraints are *linear* in the unknown camera centre $\bar{\mathbf{Q}}$ and the unknown line parameters d, d'.

In summary, 3D lines and general cameras have a linear relationship if a reference plane is known. The linear relationship holds even for a minimal representation of a 3D line with 2 parameters. This is true for all 3D lines which do not lie on the reference plane. 3D lines on the reference plane can be detected with a singular value analysis of the matrix M in eqn. 3.63. As with points, such 3D lines may be determined directly by 2 3D points $\mathbf{X} = (\mathbf{x}, 0)^T$ and $\mathbf{X}' = (\mathbf{x}', 0)^T$, where $\mathbf{x}$ and $\mathbf{x}'$ are two arbitrary points of the image line $\mathbf{l}$ in an arbitrary view.

3.3.2 Multiple Views & Reference Plane: Linear System of Cameras and Lines

We introduces now our direct reference plane approach (Line-DRP) for multiple lines observed in multiple views. Note that the practical algorithms of this approach are outlined in sec. 6.2.1.

Let us consider the reference plane case for multiple 3D lines and multiple cameras. A 3D line $\mathbf{L}_i$ is projected by a camera with centre $\bar{\mathbf{Q}}_j$ onto the image line segment $\mathbf{l}_{ij}$, which has endpoints $\mathbf{x}_{ij}$ and $\mathbf{x}'_{ij}$. Since the relationship between lines and cameras is linear, we can build a *single* linear system consisting of *all* projection relations in terms of image coordinates. In the previous chapter we derived different projection relations depending on the representation of the 3D lines. These different relations lead to 3 different approaches to reconstruct the lines and cameras simultaneously in a linear system.

The first approach represents a 3D line $\mathbf{L}_i$ with two distinct points $\mathbf{X}_i$ and $\mathbf{X}'_i$. In this case the two projection constraints in 3.62 have to be fulfilled. However, these projection relations do not specify the position of the two points on the line, i.e. the two points have 2 degrees of freedom. The remaining degrees of freedom can be fixed by using additionally

the projection constraints for points (eqn. 3.13) for one reference view. For n lines in m views the linear system takes the form:

$$\begin{pmatrix} S_{11} & 0 & \dots & -S_{11} & 0 & \dots \\ 0 & S'_{11} & \dots & -S'_{11} & 0 & \dots \\ & & \vdots & & & \\ \mathbf{l}_{11}^T & 0 & \dots & -\mathbf{l}_{11}^T & 0 & \dots \\ 0 & \mathbf{l}_{11}^T & \dots & -\mathbf{l}_{11}^T & 0 & \dots \\ \mathbf{l}_{12}^T & 0 & \dots & 0 & -\mathbf{l}_{11}^T & \dots \\ 0 & \mathbf{l}_{12}^T & \dots & 0 & -\mathbf{l}_{11}^T & \dots \\ & & \vdots & & & \end{pmatrix} \begin{pmatrix} \bar{\mathbf{X}}_1 \\ \bar{\mathbf{X}}'_1 \\ \vdots \\ \bar{\mathbf{X}}_n \\ \bar{\mathbf{X}}'_n \\ \bar{\mathbf{Q}}_1 \\ \vdots \\ \bar{\mathbf{Q}}_m \end{pmatrix} = 0 \quad , \tag{3.69}$$

where S_{ij} and S'_{ij} (for $\mathbf{x}'_{ij}$) are defined as in eqn. 3.18. In this case the first view is used as a reference view. The system matrix S in eqn. 3.69 is of size $2n(m+2) \times 3(2n+m)$. Throughout the thesis, this method is denoted the **Line-DRP method**, i.e. *D*irect *R*eference *P*lane method for lines.

It has been shown that the direction of a 3D line $\mathbf{L}_i$ can be determined directly from multiple image lines $\mathbf{l}_{i1}, \dots, \mathbf{l}_{im}$. This means that one point $\mathbf{X}_i$ is sufficient to determine $\mathbf{L}_i$. This gives the second method of reconstructing lines and cameras from the reduced linear system:

$$\begin{pmatrix} S_{11} & 0 & \dots & -S_{11} & 0 & \dots \\ & & \vdots & & & \\ \mathbf{l}_{11}^T & 0 & \dots & -\mathbf{l}_{11}^T & 0 & \dots \\ \mathbf{l}_{12}^T & 0 & \dots & 0 & -\mathbf{l}_{11}^T & \dots \\ & & \vdots & & & \end{pmatrix} \begin{pmatrix} \bar{\mathbf{X}}_1 \\ \vdots \\ \bar{\mathbf{X}}_n \\ \bar{\mathbf{Q}}_1 \\ \vdots \\ \bar{\mathbf{Q}}_m \end{pmatrix} = 0 \quad . \tag{3.70}$$

The system matrix S is now of size $n(m+2) \times 3(m+n)$. The extension to general cameras is as in the previous approach. We will call this variation of the Line-DRP method the **Line-DRP(1p) approach**, since only one point per 3D line is reconstructed by the linear system.

Finally, a 3D line $\mathbf{L}_i$ may be represented by the minimum number of unknown parameters: d_i, d'_i. With this representation, the relationship between a camera and a line is still

linear, which can be used to derive the linear system:

$$\begin{pmatrix} \vdots \\ \text{formed from eqns. 3.66} \\ \vdots \end{pmatrix} \begin{pmatrix} d_1 \\ d'_1 \\ \vdots \\ d_n \\ d'_n \\ \bar{\mathbf{Q}}_1 \\ \vdots \\ \bar{\mathbf{Q}}_m \end{pmatrix} = 0 \quad . \tag{3.71}$$

For this approach, the system matrix S is of size $3mn \times 2n + 3m$. This variation will be denoted the **Line-DRP(min) method**, since it is based on a minimal representation of a line.

All three linear systems can be used to determine the unknown lines and cameras directly from image measurements. This can be done by applying the SVD to S, i.e. $S = U\, D\, V^T$. S has a 4-dimensional nullspace since the reference plane fixes 11 of the 15 degrees of freedom of the projective space $\mathcal{P}^3$. The non-trivial solution may be obtained by summation of the 4 last singular vectors of V. In practice only the matrix V is computed which reduces the computation time considerably, as will be discussed in sec. 6.1. As already mentioned, the three systems are only valid for 3D lines outside the reference plane. Those lines on the reference plane can be detected and reconstructed separately (see sec. 3.3.1). The advantages of this linear reconstruction approach are the same as in the point case: *All* lines not on the reference plane and *all* camera centres are reconstructed simultaneously, the process is linear and missing data is handled naturally.

However, the three different linear approaches have their advantages and disadvantages. Let us assume that all lines are visible in all views and e.g. $n = 100$ and $m = 10$. The first approach is obviously the simplest one since the image data is used directly. Unfortunately, it gives a very large system of equations, i.e. S is of size 2400×630. In addition, it cannot be used if any 3D line might be on or close to the reference plane. The second approach is not sensitive to this issue. Furthermore, the number of unknowns is smaller, i.e. S is of size 1200×330. The disadvantage of this approach is that the directions of the lines are derived directly from the reference plane. Therefore, uncertainty in the reference plane leads to inaccurately estimated line directions and consequently affects the solution obtained from the linear system negatively. The third approach has even fewer unknowns, i.e. S is of size 3000×230. The drawback of the third approach in comparison with the second is that it uses more information which is derived directly from the reference plane. However, it has the advantage, that the "artificial" extra constraints for a reference view are dispensable. An experimental comparison of the three approaches will be given in sec. 6.2.

The three linear systems use *all* available projection relations of the lines. Consequently, a minimum number of 3D lines is sufficient. The remaining question is, how many 3D lines are needed to determine a projective reconstruction? As we have seen, 3D

lines and 2D image lines have fewer degrees of freedom in the reference plane case in comparison with the general case. Since a 3D line can be specified with 2 unknown parameters, the total number of unknown camera and line parameters is $3m + 2n - 4$. A projected line gives only one constraint, since it has to pass through the vanishing point $\mathbf{v}$ of the line. In order to obtain a projective line reconstruction, the following condition has to be satisfied:

$$mn \geq 3m + 2n - 4 \quad \text{or} \quad n \geq 3 + \frac{2}{m-2} \ . \tag{3.72}$$

This means that for 3 views 5 lines are need and for more than 3 views 4 lines are sufficient.

3.3.3 Multiple Views: Camera Constraints

The previous sec. 3.3.2 introduced the first approach for reconstructing multiple 3D lines observed in multiple views. As with points, we review a second approach based on constraints which solely involve camera parameters. This approach is applicable to both general and reference plane configurations.

General configurations

Let us first count the minimum number of 3D lines needed to determine the geometry. We saw in sec. 3.3.1, that n lines and m cameras give $2mn$ constraints. The number of unknown camera and line parameters is $11m + 4n - 15$, since a 3D line has 4 degrees of freedom. In order to obtain a projective reconstruction the following relation has to hold:

$$2mn \geq 11m + 4n - 15 \quad \text{or} \quad n \geq \frac{11m - 15}{2m - 4} \ . \tag{3.73}$$

Therefore, as in the reference plane case, 2 views are insufficient to determine the geometry on the basis of line correspondences only. Furthermore, with 3 views at least 9 lines are needed and with 4 views a minimum of 8 lines is required. For these minimal cases no practical non-linear reconstruction method is known so far. In the following we will review linear methods.

With 3 views, a 3D line $\mathbf{L}$ is projected by a camera P_i on the image line $\mathbf{l}_i$. Let us consider the three planes $\mathbf{\Pi}_1 = P_1^T\, \mathbf{l}_1, \mathbf{\Pi}_2 = P_2^T\, \mathbf{l}_2$ and $\mathbf{\Pi}_3 = P_3^T\, \mathbf{l}_3$ (see fig. 3.5). Since they must intersect uniquely in the 3D line $\mathbf{L}$, the 4×3 matrix

$$M = [\mathbf{\Pi}_1\ \mathbf{\Pi}_2\ \mathbf{\Pi}_3] = [P_1^T\mathbf{l}_1\ \ P_2^T\mathbf{l}_2\ \ P_3^T\mathbf{l}_3] \tag{3.74}$$

is of rank 2. Therefore, all 4 subdeterminants of M of size 3×3 must vanish. Moreover, eqn. 3.74 shows that each subdeterminant is tri-linear in the image line coordinates. As a consequence, these constraints may be written using a trifocal tenor $\mathcal{T}$ as

$$\mathbf{l}_{1p}\, \mathbf{l}_{2q}\, \mathbf{l}_{3r}\, \epsilon^{piw}\, \mathcal{T}_i^{qr} \ = \ 0^w \ . \tag{3.75}$$

Only 2 of the 3 constraints are linearly independent. This means that 13 lines are needed to obtain the trifocal tensor linearly, which has 26 independent ratios of elements. Furthermore, the trifocal tensor $\mathcal{T}$ is identical to the trifocal tensor in 3.32 derived from constraints

on 3D points. This is an important result since the remaining 3D line reconstruction task is now identical to the point case. The three cameras, and consequently the 3D lines, may be derived from $\mathcal{T}$. Other approaches which are based on the trifocal tensor, like the joint-image closer constraint method (Triggs, 1997b), may be applied as well for 3D lines.

With 4 views, the 4 planes $\mathbf{\Pi}_{1-4} = P_{1-4}^T \mathbf{l}_{1-4}$ have to intersect uniquely in the 3D line $\mathbf{L}$. Combining the planes into the 4×4 matrix M gives

$$M = [\mathbf{\Pi}_1 \ \mathbf{\Pi}_2 \ \mathbf{\Pi}_3 \ \mathbf{\Pi}_4] = [P_1^T \mathbf{l}_1 \ \ P_2^T \mathbf{l}_2 \ \ P_3^T \mathbf{l}_3 \ \ P_4^T \mathbf{l}_4] \ . \tag{3.76}$$

As with 3 views, M is of rank 2, since the linear system $M^T \mathbf{X} = \mathbf{0}$ has a 2-dimensional space of solutions. Therefore, a first condition is that the 4×4 determinant of M has to vanish. This gives a quadri-linear constraint on the image line coordinates. However, this condition is insufficient, since all 3×3 subdeterminant must vanish as well. Therefore, the quadri-focal tensor $\mathcal{Q}$ has to satisfy the following tri-linear constraints for each combination of three lines:

$$\begin{aligned} \mathbf{l}_{1p} \, \mathbf{l}_{2q} \, \mathbf{l}_{3r} \ \mathcal{Q}^{pqrs} = 0^s \quad &, \quad \mathbf{l}_{1p} \, \mathbf{l}_{2q} \, \mathbf{l}_{4s} \ \mathcal{Q}^{pqrs} = 0^r \ , \\ \mathbf{l}_{1p} \, \mathbf{l}_{3r} \, \mathbf{l}_{4s} \ \mathcal{Q}^{pqrs} = 0^q \quad &, \quad \mathbf{l}_{2q} \, \mathbf{l}_{3r} \, \mathbf{l}_{4s} \ \mathcal{Q}^{pqrs} = 0^p \ . \end{aligned} \tag{3.77}$$

Only 9 of the 12 constraints are linearly independent. Therefore, 9 lines determine linearly the quadri-focal tensor, which has 80 independent ratios of elements. Note that there are no algebraically new constraints which involve 4 views.

As already seen for points, the representation of the camera geometry by multi-view tensors is limited to 4 views. The matrix M, which combines all planes $\mathbf{\Pi}_i = P_i^T \mathbf{l}_i$, is in the m-view case $M = [\mathbf{\Pi}_1 \ \ldots \ \mathbf{\Pi}_m]$. The constraint that an arbitrary 4×4 (and 3×3) subdeterminant has to vanish involves a maximum of 4 views. The camera constraints for 3D lines in 3 and 4 views have been studied together with the point case in (Triggs, 1995; Hartley, 1995; Faugeras and Mourrain, 1995; Heyden, 1998; Schmid and Zisserman, 2000). Furthermore, mixtures of 3D points and 3D lines have been investigated (see sec. 3.5).

Reference plane configurations

Let us consider the camera constraints for the reference plane case. A camera i can be written as $P_i = [I \mid -\bar{\mathbf{Q}}_i]$ if the invertible, infinite homography H_i^∞ is known. The plane $\mathbf{\Pi}_i$, which includes the image line $\mathbf{l}_i$ of a 3D line $\mathbf{L}$ in camera i (see fig. 3.5), is $\mathbf{\Pi}_i = P_i^T \mathbf{l}_i = (\mathbf{l}_i, -\bar{\mathbf{Q}}_i^T \mathbf{l}_i)^T$. We may combine the planes $\mathbf{\Pi}_i$ for m views into the matrix

$$M = [\mathbf{\Pi}_1 \ \ldots \mathbf{\Pi}_m] = \begin{pmatrix} \mathbf{l}_1 & \cdots & \mathbf{l}_m \\ -Q_1^T \mathbf{l}_1 & \cdots & -Q_1^T \mathbf{l}_1 \end{pmatrix} \ . \tag{3.78}$$

As in the general case, M is of rank 2. The condition that all 3×3 subdeterminants have to vanish gives the different camera constraints. The main observation is here that *all camera constraints are linear in the unknown camera centres.* As with points, these constraints can be used to reconstruct 3 or 4 cameras from a linear system of camera constraints. If

the cameras are known, the tensors can be computed directly from the known cameras with eqns. 3.32 and 3.34 respectively. Furthermore, the constraints between any triplet or quadruplet of views can be used to compute *all* cameras simultaneously. This leads to a similar approach as described by Hartley et al. (2001) for points.

According to Triggs (2000), the trifocal constraints can be written as

$$(\mathbf{l}_1 \times \mathbf{l}_2)\ (\mathbf{l}_3^T\ \bar{\mathbf{Q}}_{13})\ -\ (\mathbf{l}_2^T\ \bar{\mathbf{Q}}_{12})\ (\mathbf{l}_1 \times \mathbf{l}_3) = \mathbf{0}\ , \tag{3.79}$$

where $\bar{\mathbf{Q}}_{ij} = \bar{\mathbf{Q}}_i - \bar{\mathbf{Q}}_j$. It can be shown that only one of the three constraints is linearly independent. Therefore, 5 lines are needed to compute linearly the camera geometry, which has $3 \cdot 3 - 4 = 5$ degrees of freedom. In the 4-view case the the matrix M is of size 4×4. As in the general case, all 3×3 subdeterminants have to vanish, which gives 4 tri-linear constraints for each combination of 3 lines:

$$\begin{aligned}
(\mathbf{l}_2 \times \mathbf{l}_3)\ (\mathbf{l}_1^T\ \bar{\mathbf{Q}}_{14})\ +\ (\mathbf{l}_3 \times \mathbf{l}_1)\ (\mathbf{l}_2^T\ \bar{\mathbf{Q}}_{24}) + (\mathbf{l}_1 \times \mathbf{l}_2)\ (\mathbf{l}_3^T\ \bar{\mathbf{Q}}_{34}) &= \mathbf{0}\ , \\
(\mathbf{l}_2 \times \mathbf{l}_4)\ (\mathbf{l}_1^T\ \bar{\mathbf{Q}}_{13})\ +\ (\mathbf{l}_4 \times \mathbf{l}_1)\ (\mathbf{l}_2^T\ \bar{\mathbf{Q}}_{23}) + (\mathbf{l}_2 \times \mathbf{l}_1)\ (\mathbf{l}_4^T\ \bar{\mathbf{Q}}_{34}) &= \mathbf{0}\ , \\
(\mathbf{l}_4 \times \mathbf{l}_3)\ (\mathbf{l}_1^T\ \bar{\mathbf{Q}}_{12})\ +\ (\mathbf{l}_4 \times \mathbf{l}_1)\ (\mathbf{l}_3^T\ \bar{\mathbf{Q}}_{23}) + (\mathbf{l}_1 \times \mathbf{l}_3)\ (\mathbf{l}_4^T\ \bar{\mathbf{Q}}_{24}) &= \mathbf{0}\ , \\
(\mathbf{l}_4 \times \mathbf{l}_3)\ (\mathbf{l}_2^T\ \bar{\mathbf{Q}}_{12})\ +\ (\mathbf{l}_2 \times \mathbf{l}_4)\ (\mathbf{l}_3^T\ \bar{\mathbf{Q}}_{13}) + (\mathbf{l}_2 \times \mathbf{l}_3)\ (\mathbf{l}_4^T\ \bar{\mathbf{Q}}_{14}) &= \mathbf{0}\ .
\end{aligned} \tag{3.80}$$

However, only 2 of the 12 constraints are linearly independent. As a consequence, 4 3D lines are needed to resolve the $4 \cdot 3 - 4 = 8$ degrees of freedom of the camera geometry. As we have seen in sec. 3.3.2, this represents the minimum number of lines needed to determine the geometry in the 3- and 4-view case.

3.3.4 Multiple Views: Factorization of Cameras and Lines

As with points, a third approach for reconstructing multiple 3D lines and cameras is factorization. We will review this technique for both general and reference plane configurations.

General configurations

Triggs (1996) suggested extending the projective point factorization algorithm (sec. 3.2.4) for lines by hallucinating 2 3D points. Let us choose two well spaced image points on the image line in a reference view. If the multi-view tensors are known, their position on the image line can be established in any view. This means that a set of 3D lines can be reconstructed by a set of two distinct 3D points using the projective point factorization algorithm. However, this approach requires that the multi-view camera tensors are determined in a pre-processing step. Note, in the point case such a pre-processing step can be avoided (see sec. 4.2.3). Another point is that the points can be ill-conditioned.

With affine views, Quan and Kanade (1997), Kahl and Heyden (1999) and Bretzner and Lindeberg (1998) suggest a factorization method which uses the image lines directly. However, this is not a one-step method as in the point case. Let us represent a 3D Line $\mathbf{L}$ by two non-homogeneous points $\bar{\mathbf{X}}_1, \bar{\mathbf{X}}_2$ as in eqn. 2.8, i.e. $\mathbf{L}: \ \bar{\mathbf{X}} = \bar{\mathbf{X}}_1 + \mu(\bar{\mathbf{X}}_2 - \bar{\mathbf{X}}_1)$. The vector $\bar{\mathbf{D}} = \bar{\mathbf{X}}_1 - \bar{\mathbf{X}}_2$ is the direction of the line. The affine projection (2.31) of a 3D line

L, represented by all points $\bar{\mathbf{X}}$, onto the image line **l**, represented by all non-homogeneous image points $\bar{\mathbf{x}}$, is

$$\bar{\mathbf{x}} = M\ \bar{\mathbf{X}} + \mathbf{t} = M\ (\bar{\mathbf{X}}_1 + \mu\bar{\mathbf{D}}) + \mathbf{t} = M\bar{\mathbf{X}}_1 + \mu M\bar{\mathbf{D}} + \mathbf{t} = \bar{\mathbf{x}}_1 + \mu M\bar{\mathbf{D}}\ . \quad (3.81)$$

Therefore, the image line **l** can be represented as $\bar{\mathbf{x}} = \bar{\mathbf{x}}_1 + \mu M\bar{\mathbf{D}}$. This means that the 2D direction vector $\bar{\mathbf{d}}$ of the image line is

$$\lambda\bar{\mathbf{d}} = M\ \bar{\mathbf{D}}\ . \quad (3.82)$$

In contrast to the affine projection equation for points (3.49), lines have an additional unknown scale factor λ. As in the point case we may formulate the measurement matrix for n line and m views:

$$\begin{pmatrix} \lambda_{11}\bar{\mathbf{d}}_{11} & \dots & \lambda_{1n}\bar{\mathbf{d}}_{1n} \\ \vdots & \ddots & \vdots \\ \lambda_{m1}\bar{\mathbf{d}}_{m1} & \dots & \lambda_{mn}\bar{\mathbf{d}}_{mn} \end{pmatrix} = \begin{pmatrix} M_1 \\ \vdots \\ M_m \end{pmatrix} \begin{pmatrix} \bar{\mathbf{D}}_1, \dots, \bar{\mathbf{D}}_n \end{pmatrix}\ . \quad (3.83)$$

In a first step, all unknown scales have to be determined. Kahl and Heyden (1999) presented a method which uses subsets, e.g. triplets, of views. On the basis of this, the directions $\bar{\mathbf{D}}_i$ of the 3D lines and the affine cameras M_i are determined by affine factorization. The remaining two degrees of freedom of each 3D line, i.e. the point $\bar{\mathbf{X}}_1$, are obtained in the last step.

Reference plane configurations

For a known reference plane, Triggs (2000) suggested to hallucinate 2 3D points as in the general projective case. Since the direction of the lines, i.e. the intersection with the plane at infinity, can be determined in advance, only one 3D point per line has to be reconstructed. However, as in the general case, the major drawback of hallucinating points is that the multi-view tensors have to be known in advance.

3.4 Planes

For the task of reconstructing multiple 3D planes and cameras, there is *no considerable difference between general configurations and reference plane configurations*. Any 3D plane may serve as the reference plane. Note, as in the previous sections the (virtual) reference plane might represent the correct plane at infinity. The only difference between the two configurations is that the reference plane has to be visible in all views. In the following, we assume that this condition is satisfied – at least for a subset of views.

Plane features are substantially different to line or point features. First, the projection of a 3D plane "onto" an image does not give an image feature like e.g. an image point. Therefore, a 3D plane is represented by a homography between *two* views (see sec. 2.4). Secondly, the detection of a 3D plane in multiple images, i.e. its homographies, is different to the point or line case. The homographies can be derived from point and/or line

features which are already matched. However, this needs an additional process of grouping coplanar point and/or line features. Alternatively, homographies can be determined directly from greylevels in a sequence of images (e.g. Bergen et al., 1992; Irani and Anandan, 1999b). Thirdly, the task of reconstructing multiple 3D planes and cameras can be solved by *any* reconstruction algorithm for points and/or lines presented in sec. 3.2 and 3.3 respectively. On the basis of the detected homographies, point and/or line correspondences can be hallucinated. These hallucinated 3D points and/or lines describe uniquely the 3D plane in space. Consequently, the task of plane reconstruction from homographies may be circumvented by using the hallucinated 3D points and/or lines instead (e.g. Szeliski and Torr, 1998). The advantages and drawbacks of this technique will be discussed in more detail in sec. 3.4.3. Finally, the task of reconstructing 3D planes and cameras from homographies only has not been addressed frequently in the past. The reason is that in most practical applications point and line features are first matched and then grouped according to coplanarity properties (e.g. Baillard and Zisserman, 1999; Bartoli et al., 2001b). This gives the task of performing a *constrained 3D reconstruction: reconstructing 3D points, 3D lines and cameras with additional coplanarity constraints* (see sec. 3.5).

In this section we will investigate different plane reconstruction methods which are based directly on homographies induced by scene planes, as opposed to hallucinating scene points. We will begin the discussion by considering a single plane observed in two views (sec. 3.4.1). On the basis of the known homography, we will derive different novel constraints which are *linear* in the unknown plane and camera parameters. Furthermore, we will show that 3 parameters of a 3D plane (its normal/orientation) may be derived from a given reference plane. As with points and lines, this makes it possible to *reconstruct multiple planes and cameras simultaneously from a linear system of equations* (sec. 3.4.2). This direct reference plane method (Plane-DRP) is the main and novel contribution for plane features and not published yet. Furthermore, we will review two other multiview reconstruction approaches. The first is reconstruction based on camera constraints (sec. 3.4.3) and the second on factorization of cameras and planes (sec. 3.4.4). Both approaches use the known homographies directly. The camera constraints method is part of our publication (Rother et al., 2002) and the factorization technique has been presented in (Triggs, 2000; Rother et al., 2002). As for points and lines, these methods are discussed theoretically here. Practical algorithms of the Plane-DRP, camera constraints and factorization method are in sec. 6.3.1.

3.4.1 Two Views: Single Plane

As already mentioned, we will assume throughout this section that a real or virtual reference plane is visible in all (subset of) views. Therefore, the corresponding infinite homographies H_i^∞ are known. We have seen, that the transformation $x'_i \sim H_i^{\infty -1} x_i$ stabilizes the reference plane in the images. Let us assume that a second plane $\mathbf{\Pi}_k$ is given by its homography H_{ij}^k between view i and j, i.e. $x_j \sim H_{ij} x_i$ (see sec. 2.4). For stabilized images points x', the corresponding homography $H_{ij}^{\prime k}$, i.e. $x'_j \sim H'_{ij} x'_i$, is given as

$$H_{ij}^{\prime k} = H_j^{\infty -1} \, H_{ij}^k \, H_i^\infty \quad . \tag{3.84}$$

As in the previous sections, we assume that the reference plane is already stabilized and therefore H'^k_{ij} will be denoted H^k_{ij}. Furthermore, any camera may be written as $P_i = [I \mid -\bar{\mathbf{Q}}_i]$.

Following (Szeliski and Torr, 1998), we will derive a relationship between a given homography, its plane parameters and the camera.

Proposition 3 *The homography H^k_{ij} of a plane $\mathbf{\Pi}_k = (\boldsymbol{n}_k, d_k)^T$ between the cameras $P_i = [I \mid -\bar{\mathbf{Q}}_i]$ and $P_j = [I \mid -\bar{\mathbf{Q}}_j]$ may be expressed as*

$$H^k_{ij} = \lambda \left(I + (\boldsymbol{n}_k^T \bar{\mathbf{Q}}_i + d_k)^{-1} \, (\bar{\mathbf{Q}}_j - \bar{\mathbf{Q}}_i) \, \boldsymbol{n}_k^T \right) \ , \tag{3.85}$$

where I is the 3×3 identity matrix and λ an unknown scalar. The vector $\boldsymbol{n}_k$ represents the plane's normal and d_k the distance to the origin.

Proof Consider a non-homogeneous point $\bar{\mathbf{X}}$ which is projected into camera P_i and P_j as $\mathbf{x}_i$ and $\mathbf{x}_j$:

$$\lambda_i \mathbf{x}_i = \bar{\mathbf{X}} - \bar{\mathbf{Q}}_i \ , \quad \text{i.e.} \quad \bar{\mathbf{X}} = \lambda_i \mathbf{x}_i + \bar{\mathbf{Q}}_i, \quad \text{and} \tag{3.86}$$

$$\lambda_j \mathbf{x}_j = \bar{\mathbf{X}} - \bar{\mathbf{Q}}_j \ . \tag{3.87}$$

The point $\bar{\mathbf{X}}$ may be eliminated using eqns. 3.86 and 3.87, which gives

$$\lambda_j \mathbf{x}_j = \lambda_i \mathbf{x}_i + \bar{\mathbf{Q}}_i - \bar{\mathbf{Q}}_j \ , \quad \text{i.e.} \quad \lambda_j \lambda_i^{-1} \mathbf{x}_j = \mathbf{x}_i - \lambda_i^{-1} (\bar{\mathbf{Q}}_j - \bar{\mathbf{Q}}_i) \ . \tag{3.88}$$

The constraint that $\bar{\mathbf{X}}$ lies on $\mathbf{\Pi}_k$ is $\mathbf{n}_k^T \bar{\mathbf{X}} + d_k = 0$. Using eqn. 3.86, this can be written as

$$\mathbf{n}_k^T \bar{\mathbf{X}} + d_k = \mathbf{n}_k^T (\lambda_i \mathbf{x}_i + \bar{\mathbf{Q}}_i) + d_k = 0 \ . \tag{3.89}$$

To obtain the scalar factor λ_i^{-1}, eqn. 3.89 may be rewritten as

$$\lambda_i^{-1} = -(\mathbf{n}_k^T \bar{\mathbf{Q}}_i + d_k)^{-1} \, \mathbf{n}_k^T \mathbf{x}_i \ . \tag{3.90}$$

Combining eqn. 3.88 and 3.90 gives

$$\lambda_j \lambda_i^{-1} \mathbf{x}_j = \mathbf{x}_i + (\mathbf{n}_k^T \bar{\mathbf{Q}}_i + d_k)^{-1} \, \mathbf{n}_k^T \mathbf{x}_i \, (\bar{\mathbf{Q}}_j - \bar{\mathbf{Q}}_i) \ . \tag{3.91}$$

Finally, eqn. 3.91 may be reformulated as

$$\lambda_j \lambda_i^{-1} \mathbf{x}_j = \left(I + (\mathbf{n}_k^T \bar{\mathbf{Q}}_i + d_k)^{-1} (\bar{\mathbf{Q}}_j - \bar{\mathbf{Q}}_i) \, \mathbf{n}_k^T \right) \mathbf{x}_i \ . \tag{3.92}$$

The 3×3 matrix in eqn. 3.92 represents the homography H^k_{ij} as in eqn. 3.85.

□

Proposition 3 is valid for all planes $\mathbf{\Pi}_k$, e.g. the plane at infinity $\mathbf{\Pi} = (0, 0, 0, 1)^T$ gives $H^k_{ij} = I$. It has been shown (e.g. Johansson, 1999; Triggs, 2000), that the unknown scalar λ may be determined directly from H^k_{ij}. Eqn. 3.85 may be rewritten as

$$H^k_{ij} - \lambda I = \lambda \, (\mathbf{n}_k^T \bar{\mathbf{Q}}_i + d_k)^{-1} \, (\bar{\mathbf{Q}}_j - \bar{\mathbf{Q}}_i) \, \mathbf{n}_k^T \ . \tag{3.93}$$

Since this matrix is the product of two rank 1 matrices, H^k_{ij} has the double eigenvalue λ. This is related to the fact that H^k_{ij} is a planar homology (Criminisi et al., 1998).

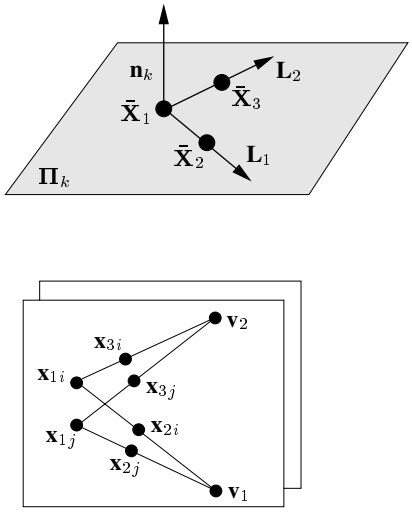


Figure 3.8. The 3 finite points $\bar{\mathbf{X}}_1, \bar{\mathbf{X}}_2, \bar{\mathbf{X}}_3$ define the plane $\mathbf{\Pi}_k$ uniquely. Furthermore, they define the lines $\mathbf{L}_1 = (\bar{\mathbf{X}}_1, \bar{\mathbf{X}}_2)$ and $\mathbf{L}_2 = (\bar{\mathbf{X}}_1, \bar{\mathbf{X}}_3)$. The directions of the lines may be determined from the vanishing points $\mathbf{v}_1$ and $\mathbf{v}_2$.

Eqn. 3.85 shows that the relationship between unknown plane parameters and camera parameters is *bi-linear*. In the following we will investigate how to linearize this relationship. As with lines, we will derive information about a plane directly from its homography and the given infinite homographies. Consider 3 finite 3D points $\bar{\mathbf{X}}_1, \bar{\mathbf{X}}_2$ and $\bar{\mathbf{X}}_3$, which uniquely define the 3D plane $\mathbf{\Pi}_k$ (see fig. 3.8). They are projected into camera P_i as $\mathbf{x}_{1-3\ i}$ and P_j as $\mathbf{x}_{1-3\ j}$. Furthermore, the 3 points define the two 3D lines $\mathbf{L}_1 = (\bar{\mathbf{X}}_1, \bar{\mathbf{X}}_2)$ and $\mathbf{L}_2 = (\bar{\mathbf{X}}_1, \bar{\mathbf{X}}_3)$. As we saw in sec. 3.3.1, the vanishing points $\mathbf{v}_1$ and $\mathbf{v}_2$ of the lines are

$$\begin{aligned} \mathbf{v}_1 &= (\mathbf{x}_{1i} \times \mathbf{x}_{2i}) \times (\mathbf{x}_{1j} \times \mathbf{x}_{2j}) \\ \mathbf{v}_2 &= (\mathbf{x}_{1i} \times \mathbf{x}_{3i}) \times (\mathbf{x}_{1j} \times \mathbf{x}_{3j}) \ . \end{aligned} \tag{3.94}$$

For calibrated translating cameras, a vanishing point in the image represents the direction of the corresponding 3D lines in space. Therefore, the normal $\mathbf{n}_k$ of the plane $\mathbf{\Pi}_k$ is defined as

$$\mathbf{n}_k = \mathbf{v}_1 \times \mathbf{v}_2 \ . \tag{3.95}$$

This means that the normal of a plane $\mathbf{\Pi}_k$ may be derived directly from its homography and the known infinite homographies. This is true for all planes except the plane at infinity, where $\mathbf{n}_k = \mathbf{0}$. The only remaining unknown parameter of a plane is d_k, which is the distance of the plane from the origin. This is the same observation as with line. *If the plane at infinity is known, the orientation of 3D lines and planes is given.* In practice the question arises, which 3 points should be "hallucinated" in order to obtain a stable estimate

the plane's normal? Szeliski and Torr (1998) showed that it is important to hallucinate points inside the image area from which the homography has been derived. This is an obvious conclusion since a homography is less accurate for the case of point extrapolation, i.e. points outside this image area. Additionally, it is important to check that the 3 points do not lie on the plane at infinity.

We are now able to linearize the relationship between planes and cameras. Let us rewrite eqn. 3.85 and define $\hat{H}^k_{ij}$ as

$$\hat{H}^k_{ij} = \lambda^{-1} H^k_{ij} - I = (\mathbf{n}_k^T \bar{\mathbf{Q}}_i + d_k)^{-1} \, (\bar{\mathbf{Q}}_j - \bar{\mathbf{Q}}_i) \, \mathbf{n}_k^T \quad . \tag{3.96}$$

Since $\hat{H}^k_{ij}$ and $\mathbf{n}_k$ are now known, the following 9 constraints have to be fulfilled:

$$\begin{aligned}
\bar{A}_i \, (\hat{h}_{1l} \, n_1 - n_l) + \bar{B}_i \, (\hat{h}_{1l} \, n_2) + \bar{C}_i \, (\hat{h}_{1l} \, n_3) + \bar{A}_j \, n_l + d_k \, \hat{h}_{1l} &= 0 \\
\bar{A}_i \, (\hat{h}_{2l} \, n_1) + \bar{B}_i \, (\hat{h}_{2l} \, n_2 - n_l) + \bar{C}_i \, (\hat{h}_{2l} \, n_3) + \bar{B}_j \, n_l + d_k \, \hat{h}_{2l} &= 0 \\
\bar{A}_i \, (\hat{h}_{3l} \, n_1) + \bar{B}_i \, (\hat{h}_{3l} \, n_2) + \bar{C}_i \, (\hat{h}_{3l} \, n_3 - n_l) + \bar{C}_j \, n_l + d_k \, \hat{h}_{3l} &= 0
\end{aligned} \tag{3.97}$$

for $l = 1, 2, 3$, where $\hat{H}^k_{ij} = (\hat{h})_{ml}$, $\mathbf{n}_k = (n_1, n_2, n_3)^T$ and $\bar{\mathbf{Q}}_i = (\bar{A}_i, \bar{B}_i, \bar{C}_i)^T$. These constraints are *linear in the unknown plane and camera parameters*. It is straightforward to show that only 3 of the 9 equations are linearly independent. As a consequence, one plane, which is observed in two views, is sufficient for a projective reconstruction, i.e. to determine the $2 \cdot 3 + 1 - 4 = 3$ degrees of freedom of the geometry. Furthermore, a scene consisting of one plane and any number of views (at least 2) can be reconstructed since the number of constraints is $3(m-1)$ and the number of unknowns $3m + 1 - 4 = 3(m-1)$. This result is not surprising. In all views a plane can be used to hallucinate 2 or more points, which are not at infinity. As we have seen, this is sufficient for a projective reconstruction of 3D points and cameras.

Finally, let us consider the special case where one camera centre is the origin of the projective space, e.g. $\mathbf{Q}_1 = (0, 0, 0, 1)^T$. Consequently, the fourth coordinate of any plane $\mathbf{\Pi}_k$, i.e. d_k, may be chosen as 1 since the camera centre $\mathbf{Q}_1 = (0, 0, 0, 1)^T$ cannot lie on $\mathbf{\Pi}_k$, i.e. $\mathbf{\Pi}_k^T \mathbf{Q}_1 \neq 0$. If we apply this to proposition 3, we obtain the following well known relation (e.g. Luong and Viéville, 1996)

Corollary 1 *The homography H^k_{1j} of plane $\mathbf{\Pi}_k = (\boldsymbol{n}_k, 1)^T$ between the cameras $P_1 = [I \mid \boldsymbol{0}]$ and $P_j = [I \mid -\bar{\mathbf{Q}}_j]$ may be expressed as*

$$H^k_{1j} = \lambda \left(I + \bar{\mathbf{Q}}_j \, \boldsymbol{n}_k^T \right) \quad . \tag{3.98}$$

As in the general case, λ is the double eigenvalue of H^k_{1j}. However, the normal $\mathbf{n}_k$, which can be computed from eqn. 3.95, must *not* be inserted into eqn. 3.98, since the condition that $d_k = 1$ would no longer be valid. We will see that the bi-linear relationship of planes and camera parameters in eqn. 3.98 can be used for 3D reconstruction by factorization.

To summarize, if a reference plane is known, the plane's normal and the unknown scale of the plane's homography may be derived directly. On the basis of this information, the general bi-linear relationship between 3D planes and cameras may be linearized.

3.4.2 Multiple Views: Linear System of Cameras and Planes

We introduces now our direct reference plane (Plane-DRP) method for multiple planes observed in multiple views. The practical algorithm of the method is outlined in sec. 6.3.1.

Consider the case of n planes which are partly visible in m views. Note that the reference plane is an additional plane which has been used to transform the cameras into calibrated translating cameras. The input data is a set of homographies H_{ij}^k, which represent a plane $\mathbf{\Pi}_k$ between the views i and j. We have seen that the normal $\mathbf{n}_k$ of any plane and the unknown scale λ in eqn. 3.85 may be derived directly from H_{ij}^k and the known reference plane. With this information we established a linear relationship between plane and camera parameters (eqn. 3.97). This makes it possible to reconstruct all distances d_k and all cameras simultaneously from a *single* linear system of equations. The system has the form

$$\begin{pmatrix} \vdots \\ \text{formed from eqns. 3.97} \\ \vdots \end{pmatrix} \begin{pmatrix} d_1 \\ \vdots \\ d_n \\ \bar{\mathbf{Q}}_1 \\ \vdots \\ \bar{\mathbf{Q}}_m \end{pmatrix} = 0 \quad . \tag{3.99}$$

Since the projective space with a fixed plane at infinity has 4 degrees of freedom, all cameras and planes can be obtained from the 4 dimensional nullspace of S as with points and lines. Note that the only plane which cannot be reconstructed in this way is the plane at infinity $\mathbf{\Pi}_\infty$, since $\mathbf{n}_k^\infty = \mathbf{0}$ and therefore d_∞ is not unique. Let us summarize the main advantages of this approach: *All* planes and *all* camera centres are reconstructed simultaneously, the process is linear and missing data is handled naturally. The main drawback of this approach is that some information about the planes, i.e. their normals and λ, has to be derived in advance, in order to linearize the problem. With noisy input data, errors in the recovery of these parameters could reduce the accuracy of the solutions for d_k and $\bar{\mathbf{Q}}_j$ considerably. If we assume, that all planes are visible in all views, i.e. all possible H_{ij}^k are known, the system matrix S is of size $\frac{9}{2}nm(m-1) \times 3m+n$. For instance, with $n = 20$ and $m = 10$, the size of S is 8100×50. As with points and lines only the matrix V of a SVD of $S = UDV^T$ has to be computed. As we will see in sec. 6.1, the complexity of this computation depends mainly on the number of unknowns, i.e. 50, and less on the number of equations, i.e. 8100. In the following, this method is denoted the *D*irect *R*eference *P*lane method for planes, the **Plane-DRP method**.

3.4.3 Multiple Views: Camera Constraints

Here we present two different approaches for deriving the unknown cameras. given the homographies. First, we will present a linear method which computes *all* the camera centres simultaneously based on camera constraints involving homographies only. This is the counterpart to Hartley et al.'s (2001) camera constraint reconstruction method for point features. In contrast to the linear method in 3.4.2 it does not need the normals of the planes

to be known. However, the drawback of this approach is that cameras and planes are reconstructed separately, first all cameras are found before all planes are recovered in a second step. Secondly, we will address the question of how the multi-view tensors may be derived from the homographies. i.e. $2, 3$ and 4, is encoded by the multi-view tensors. As we have seen in sections 3.2.3 and 3.3.3, the multi-view tensors encode the geometry of a limited number (2, 3 or 4) of cameras.

Let us begin with the linear reconstruction method based on camera constraints presented in (Rother et al., 2002). From the SVD of the matrix $\hat{H}^k_{ij} = UDV^T$ (see eqn. 3.96) we obtain

$$\bar{\mathbf{Q}}_j - \bar{\mathbf{Q}}_i = \lambda^k_{ij}\, \sigma_1\, \mathbf{u}_1 \tag{3.100}$$

$$\mathbf{n}_k = \frac{1}{\lambda^k_{ij}} \left(\mathbf{n}_k^T \bar{\mathbf{Q}}_j + d_k\right)^{-1} \mathbf{v}_1, \tag{3.101}$$

where σ_1 is the first singular value of D and $\mathbf{u}_1$ and $\mathbf{v}_1$ are the first columns of U and V respectively. The scalar factor λ^k_{ij} is undetermined in this case. From eqn. 3.100 we may derive the following three relations which are linear in the camera parameters:

$$\begin{aligned}
\sigma_1 u_y(\bar{A}_j - \bar{A}_i) - \sigma_1 u_x(\bar{B}_j - \bar{B}_i) &= 0 \\
\sigma_1 u_z(\bar{A}_j - \bar{A}_i) - \sigma_1 u_x(\bar{C}_j - \bar{C}_i) &= 0 \\
\sigma_1 u_z(\bar{B}_j - \bar{B}_i) - \sigma_1 u_y(\bar{C}_j - \bar{C}_i) &= 0 \ ,
\end{aligned} \tag{3.102}$$

where $\bar{\mathbf{Q}}_i = (\bar{A}_i, \bar{B}_i, \bar{C}_i)^T$ and $\mathbf{u}_1 = (u_x, u_y, u_z)^T$ and σ_1 as the first singular value. Therefore, each homography $\hat{H}^k_{ij}$ provides 2 linearly independent equations of the form (3.102). The reason for keeping σ_1 in the equations is that they are satisfied automatically for the plane at infinity where $\hat{H}^k_{ij} = 0$ and therefore $\sigma_1 = 0$. The linear equations give the following system:

$$\begin{pmatrix} \vdots \\ \text{formed from eqns. 3.102} \\ \vdots \end{pmatrix} \begin{pmatrix} \bar{\mathbf{Q}}_1 \\ \vdots \\ \bar{\mathbf{Q}}_m \end{pmatrix} = 0 \quad . \tag{3.103}$$

As with the linear reconstruction method for points (sec. 3.2.3), all camera centres are determined by the 4 dimensional nullspace of the system matrix S using SVD. For efficiency, only the matrix V of $S = UDV^T$ needs to be computed. Furthermore, constraints derived from points which lie on the plane at infinity do not influence the solution. The scalar factors λ^k_{ij} can now be derived from the camera centres using eqn. 3.100. If the scalar factors λ^k_{ij} and the camera centre $\bar{\mathbf{Q}}_i$ are known, each plane $\mathbf{\Pi}_k = (\mathbf{n}_k, d_k)^T$ may be determined directly from eqn. 3.101. Optionally, these two steps can be performed iteratively. This means that the scalar factors λ^k_{ij} are recalculated from the plane $\mathbf{\Pi}_k$ and used to recompute all the camera centres simultaneously.

Let us now address the second question of computing the multi-view tensors from homographies only. There are two options, either use the homographies directly or hallucinate point and/or line correspondences. If the multi-view tensors are known, the cameras and eventually the planes may be derived by hallucinating 3 3D points.

We begin with the first approach, and initially consider 2 views (Luong and Faugeras, 1993). A plane is visible in two views and its homography H_{12} is given. Two image points which are related by H_{12}, i.e. $\mathbf{x}_2 \sim H_{12}\,\mathbf{x}_1$, define a 3D point which lies on the plane. Furthermore, the two image points have to fulfill the epipolar constraint: $\mathbf{x}_2^T\,F_{12}\,\mathbf{x}_1 = 0$ (eqn. 3.23). Combining both constraints gives the following equation:

$$\mathbf{x}_1\,H_{12}^T\,F_{12}\,\mathbf{x}_1 = 0 \quad . \tag{3.104}$$

Since this condition has to be fulfilled for all $\mathbf{x}_1$, the matrix $H_{12}^T\,F_{12}$ has to be skew-symmetric and therefore

$$S = H_{12}^T\,F_{12} + F_{12}^T\,H_{12} = 0 \quad . \tag{3.105}$$

This gives 6 constraints on the 9 unknown elements of F_{12}:

$$s_{ij} = \sum_k h_{ki} f_{kj} + f_{ki} h_{kj} = 0 \quad \text{for} \quad i,j = 1,2,3 \quad , \tag{3.106}$$

where $H_{12} = (h)_{ki}$ and $F_{12} = (f)_{ki}$. These constraints can be used to determine F_{12} linearly from one or more planes as shown in (Luong and Faugeras, 1993). Note that eqn. 3.106 can also be used to verify if a given homography represents a "real" plane in the scene. Since we assume stabilized images, the F-matrix is of the form $F_{12} = [e_{12}]_\times$. Obviously, eqn. 3.106 holds for the original images as well. Luong and Faugeras (1993) reported that this method of determining the F-matrix is not very stable. Later, Szeliski and Torr (1998) showed that two constraints in 3.106 correspond to constraints which can be derived from the two hallucinated image points $(1,0,0)^T$ and $(0,1,0)^T$. Obviously, this is a bad choice since points at infinity do not lie inside the image area from which the homography has been derived. As previously mentioned, a homography is less accurate for point extrapolation than for point interpolation. Szeliski and Torr (1998) demonstrated that F can be determined stably if the hallucinated image points lie inside the image area of the homography, Consequently, the second alternative of computing the multi-view tensors from hallucinated points is probably preferable to the first option. Therefore, we will *not* investigate methods to determine the tri- or quadrifocal tensor from homographies only.

An alternative way of computing F_{12}, i.e. the epipole e_{12}, from the planar homography H_{12} was introduced by Johansson (1999). We saw (eqn. 3.98) that the homography H_{12} has a double eigenvalue λ. Johansson showed that the eigenvector corresponding to the remaining eigenvalue is the epipole $\mathbf{e}_{12}$ (up to scale). Furthermore, the fundamental matrix derived in this manner and the planar homographies can be used directly for 3D reconstruction of two views and multiple planes. Additionally, Johansson presented a simple method for view synthesis of a piecewise planar scene from one view. Xu et al. (2000) exploited the same idea to obtain a Euclidean reconstruction of two planes visible in two views.

Let us briefly review the second method of determining the camera geometry, i.e. the multi-view tensors, from hallucinated image points. The approach of hallucinating image points is simple and in some cases (Szeliski and Torr, 1998) more stable than using H directly. On the basis of 2 (or more) hallucinated image points and a known reference plane,

any relevant technique described in sec. 3.2 can be used to reconstruct the scene and/or the multi-view tensors. However, at least 2 of the image points must not lie on the plane at infinity. Alternatively, 3 (or more) hallucinated image points can be used directly, i.e. without a reference plane, to obtain the multi-view tensors and/or the projective reconstruction (see relevant methods in sec. 3.2). The comparison of direct methods and hallucinating image points methods will be continued in the experiments in sections 6.3.2 and 6.3.3.

3.4.4 Multiple Views: Factorization of Cameras and Planes

Finally, we present a factorization method based on homographies. As with all factorization approaches, no missing data is allowed, all planes have to be visible in all views. Let us choose the first view P_1 as a reference view: $P_1 = [I \mid \mathbf{0}]$. With the assumption that the reference plane represents the plane at infinity, the projective space has only $15 - 11 - 3 = 1$ degree of freedom. By choosing $\bar{\mathbf{Q}}_1$ as $\mathbf{0}$, any plane which does not pass through $\bar{\mathbf{Q}}_1$ may be defined as $\mathbf{\Pi}_k = (\mathbf{n}_k, 1)^T$. In the following, only homographies of the type H^k_{1j} will be used. A homography H^k_{1j} relates the reference view with view j via the plane $\mathbf{\Pi}_k$. For simplicity, H^k_{1j} will be denoted H^k_j. For this special scenario, corollary 1 (sec. 3.4.1) defines an equation which relates H^k_j, the plane's normal $\mathbf{n}_k$ and a camera's centre $\bar{\mathbf{Q}}_j$. Since λ is the double eigenvalue of H^k_j, we may compute $\hat{H}^k_j$ from eqn. 3.98 as

$$\hat{H}^k_j = \lambda^{-1} H^k_j - I = \bar{\mathbf{Q}}_j \mathbf{n}_k^T \quad . \tag{3.107}$$

We are now able to construct a measurement matrix of size $3m \times 3n$, which is the product of two vectors: one vector containing all the camera centres and another vector containing all the planes:

$$W = \begin{pmatrix} \hat{H}^1_2 & \cdots & \hat{H}^n_2 \\ \vdots & & \vdots \\ \hat{H}^1_m & \cdots & \hat{H}^n_m \end{pmatrix} = \begin{pmatrix} \bar{\mathbf{Q}}_2 \\ \vdots \\ \bar{\mathbf{Q}}_m \end{pmatrix} \begin{pmatrix} \mathbf{n}_1^T, \ldots, \mathbf{n}_n^T \end{pmatrix} \quad . \tag{3.108}$$

The matrix W has rank at most 1, which corresponds to the single degree of freedom of our specific projective space. As with all factorization methods, the final reconstruction of the cameras and planes can be obtain from the SVD of W. The idea of factorizing planes and cameras from given homographies was introduced by Triggs (2000). However, the approach is explained very briefly and without experiments. Rother et al. (2002) provided a more extensive description of the theory and additionally presented experimental results.

A less compact factorization method was introduced by Shashua and Avidan (1996) for the case of 2 views. From eqn. 3.98, the homography between view 1 and 2 via a plane k is $H^k = \lambda_k I + \lambda_k \bar{\mathbf{Q}} \mathbf{n}_k^T$, where $H^k = H^k_{12}$ and $\bar{\mathbf{Q}} = \bar{\mathbf{Q}}_1$. Writing the homography H^k as a vector h_k of size 9×1, the following relation holds:

$$\begin{pmatrix} h_1 & \cdots & h_n \end{pmatrix}_{9 \times n} = \begin{pmatrix} \bar{\mathbf{Q}} & \mathbf{0} & \mathbf{0} \\ I & \mathbf{0} & \bar{\mathbf{Q}} & \mathbf{0} \\ & \mathbf{0} & \mathbf{0} & \bar{\mathbf{Q}} \end{pmatrix}_{9 \times 4} \begin{pmatrix} \lambda_1 & \cdots & \lambda_n \\ \lambda_1 \mathbf{n}_1 & \cdots & \lambda_n \mathbf{n}_n \end{pmatrix}_{4 \times n} \tag{3.109}$$

where I is written as a 9-vector as well. This means, that all plane normals $\mathbf{n}_k$ and all unknown scalar factors λ_k may be obtained from a rank-4 factorization. Furthermore, the space of all the homographies is a 4 dimensional subspace of $\mathcal{P}^8$. The same observation was used by Zelnik-Manor and Irani (1999) for the alignment of multiple planes in multiple views.

With the assumption of calibrated cameras, Sturm (2000) presented several reconstruction algorithms for multiple planes visible in multiple views. In a first step, the planes' orientations and the cameras' rotations are computed simultaneously by factorization. In a second step, the positions of the cameras and planes are determined.

3.5 Combining Feature and Scene Constraints

This section will address two issues, combining point, line and plane features for reconstruction, and applying scene constraints to the reconstruction process. The main observation will be that *our direct reference plane (DRP) reconstruction method extends straightforward to all three types of features and may include several interesting scene constraints, such as incidence relationships.* Since our method is based on a linear system, it obviously extends to all feature or scene constraints which are linear in the unknown feature and camera parameters. Note that other linear methods like (Hartley et al., 2001) can *not* integrate scene constraints since 3D features are reconstructed separately. We would like to point out that neither issue is part of the main contribution of the thesis nor are they necessary for the understanding of the following chapters. Both issues are not part of any of our previous publications. Furthermore, the techniques discussed below are not novel and are not evaluated experimentally in this thesis.

So far, we have discussed several methods of reconstructing multiple points, lines or planes *separately* in multiple views. A natural and practical question is whether it is possible to combine these methods? Combined methods have the advantage that, two (or three) feature types constrain the camera's position simultaneously. This issue is addressed in sec. 3.5.1.

The second issue, discussed in sec. 3.5.2, concerns the task of applying constraints on the scene. Consider reconstructing a man-made environment. Such an environment has salient properties like parallelism, orthogonality, symmetry or coplanarity of features. These properties are constraints on the scene, the so-called **scene or geometric constraints**, e.g. parallel planes or points lying on a plane. To create realistic looking reconstructions, it is important that these properties are preserved. This may be denoted **constrained reconstruction**. To this end, we will show that all incidence relationships of points, lines and planes can be incorporated in our DRP reconstruction method. Furthermore, constraints concerning known 3D features can be applied, for instance the coordinates of a 3D line being known.

3.5.1 Combination of Features

As in the previous chapters, the combination of feature types are discussed separately for the three different reconstruction approaches: the direct reference plane method, applying camera constraints and factorization. Note that the structure constraints are irrelevant here since they only exist for point features. Furthermore, the "camera constraint" and "factorization" approaches are discussed for both general configurations and reference plane configurations. We will begin with our direct reference plane approach.

Linear system of points, lines and planes

With a known reference plane, the relationship between 3D points and cameras (sec. 3.2.2), 3D lines and cameras (sec. 3.3.2) and 3D planes and cameras (sec. 3.4.2) are all linear. Thus it is simple to combine the three different feature types into one linear system. For n points, k lines, p planes and m cameras the linear system has the form

$$\begin{pmatrix} \vdots \\ \text{formed from eqns. 3.13} \\ \vdots \\ \text{formed from eqns. 3.66} \\ \vdots \\ \text{formed from eqns. 3.97} \\ \vdots \end{pmatrix} \begin{pmatrix} \bar{\mathbf{X}}_1 \\ \vdots \\ \bar{\mathbf{X}}_n \\ l_1 \\ l'_1 \\ \vdots \\ l_k \\ l'_k \\ d_1 \\ \vdots \\ d_p \\ \bar{\mathbf{Q}}_1 \\ \vdots \\ \bar{\mathbf{Q}}_m \end{pmatrix} = 0 \quad , \tag{3.110}$$

where l_i and l'_i denote the two unknown line parameters, which were denoted d_i and d'_i in sec. 3.3.1. A solution for all three feature types and the cameras can be obtained simultaneously using SVD. The four-dimensional nullspace of the system matrix gives the unique reconstruction, provided the configuration is not degenerate.

Alternatively, a 3D line may be represented by two points $\mathbf{X}_i$ and $\mathbf{X}'_i$, instead of l_i and l'_i, which means that eqn. 3.66 is replaced by eqn. 3.69. Furthermore, a 3D plane may be represented by three finite points, which means that d_i is replaced by $\bar{\mathbf{X}}_i, \bar{\mathbf{X}}'_i, \bar{\mathbf{X}}''_i$ and eqn. 3.97 is used instead of eqn. 3.13.

Camera constraints

As was seen in sections 3.2.3 and 3.3.3, the camera constraints may be represented by multi-view tensors which encode the camera geometry. Point and line features give constraints on the tensors which are linear in the tensor elements, e.g. eqn. 3.23. If planes are represented by 3 hallucinate points (or lines), they give linear constraints on the tensors as well. Furthermore, it is possible to use incidence equations relating point and line correspondences to each other in 3 or 4 views. Note that chapter 4 includes a literature survey of camera constraints. For example, consider three cameras which observe a 3D line and a 3D point, where the point lies on the line. The 3D point is projected into one view as $\mathbf{x}_1$ and the 3D line into the two other views as $\mathbf{l}_2, \mathbf{l}_3$. This gives one **trifocal constraint** of the form

$$\mathbf{x}_1^i \; \mathbf{l}_{2q} \; \mathbf{l}_{3r} \; \mathcal{T}_i^{qr} \;=\; 0 \;\; . \tag{3.111}$$

Further examples are listed in (Hartley and Zisserman, 2000). As a consequence, points, lines and planes can be used together to determine linearly the geometry of $2, 3$ or 4 views. Furthermore, the joint-image closer constraint method (Triggs, 1997b) may be used the derived the geometry of all views from points, lines and planes together.

The same observations are valid for the reference plane case. Furthermore, *the camera constraints in this case have the main advantage that they are linear in the unknown camera parameters*. This has been shown for points (eqn. 3.39, 3.42 and 3.43), lines (eqn. 3.79 and 3.80) and planes (eqn. 3.102). Optionally, 3 points may be hallucinated for planes. Even the trifocal constraint (Triggs, 2000) for a 3D point lying on a 3D line is linear for calibrated translating cameras $P_j = [I| - Q_j]$:

$$(\mathbf{l}_2^T \mathbf{x}_1) \; (\mathbf{l}_3^T \; \bar{\mathbf{Q}}_{13}) \;-\; (\mathbf{l}_2^T \; \bar{\mathbf{Q}}_{12}) \; (\mathbf{l}_3^T \mathbf{x}_1) = \; 0 \;\; , \tag{3.112}$$

where $\bar{\mathbf{Q}}_{ij} \;=\; \bar{\mathbf{Q}}_i - \bar{\mathbf{Q}}_j$. It is expected that all mixtures of point and line features give constraints which are linear in the unknown camera parameters, although we do not provide an analysis here. Therefore, all camera constraints derived from points, lines and planes and all possible all mixtures may be used to reconstructed all cameras simultaneously. This idea is a straightforward extension of Hartley et al.'s (2001) point-based reconstruction approach described in sec. 3.2.3. However, it has not to our knowledge been presented or evaluate experimentally in a previous publication.

Factorization

The simplest way of extending a point based factorization method for lines and/or planes is by representing a line with 2 points and a plane with 3 points. For planes, the 3 corresponding image points can be determined from the homographies. The correspondence problem is, however, more tricky for lines. To obtain corresponding image lines either the endpoints of the image line segments are in correspondence or some multi-view tensors are known in advance. The idea of combining features for factorization has been investigated in the affine case by Morris and Kanade (1998). They represented image features with a directional uncertainty model. Furthermore, they included the coplanarity constraint of multiple

features in the factorization framework. In the projective case, Triggs (1996) suggested the combination of point and line features on the basis of known multi-view tensors.

Combining points and lines for affine factorization was investigated in (Quan and Kanade, 1997; Bretzner and Lindeberg, 1998; Kahl and Heyden, 1999). Kahl and Heyden (1999) included also conics in this framework. As seen in sec. 3.3.4, they did *not* represent a 3D line by 2 3D points. The combined affine factorization method for points and lines is then a simple combination of the two measurement matrices in eqns. 3.50 and 3.83. However, as was seen in sec. 3.3.4, the unknown scales λ_{ij} of the lines have to be determined first.

For a known reference plane, Triggs (2000) suggested combining the three individual factorization methods for points (sec. 3.2.4) , lines (sec. 3.3.4) and planes (sec. 3.3.4). However, a 3D line has to be presented by 2 3D points, which means that some multi-view tensors have to be known. Using the measurement matrices in eqn. 3.56 and 3.108, this gives for n points, k lines, p planes and m cameras:

$$\begin{pmatrix} \mathbf{x}'_{11} & \dots & \mathbf{x}'_{1\ (n+k)} & \hat{H}_2^2 & \dots & \hat{H}_2^p \\ \vdots & \ddots & \vdots & \vdots & & \vdots \\ \mathbf{x}'_{m1} & \dots & \mathbf{x}'_{m\ (n+k)} & \hat{H}_m^2 & \dots & \hat{H}_m^p \end{pmatrix} = \begin{pmatrix} \bar{\mathbf{Q}}_1 \\ \vdots \\ \bar{\mathbf{Q}}_m \end{pmatrix} \left(W_i, L_i, L'_i, \mathbf{n}_i^T\right) \quad (3.113)$$

where W_i are the 3D point "depths", L_i, L'_i are the two 3D point "depths" of the lines and $\mathbf{n}_i$ are the planes' normals.

3.5.2 Scene Constraints

The goal of this section is to investigate how scene constraints may be incorporated into the reconstruction process. Several well known techniques and reconstruction systems are presented and discussed. Furthermore, we classify the most important scene constraints. In particular, we are interested in constraints which are linear in the unknown feature parameters. We will show that the incidence relationships, e.g. a point lies on a plane, and the constraints concerning known 3D features, for instance the coordinates of a 3D line being known, are linear. Therefore, they can be integrated into our DRP method for multiple features presented in sec. 3.5.1.

Scene constraints may be classified into three categories: projective invariant, affine invariant and metric (or Euclidean) invariant constraints. The main invariant properties or constraints of the three different spaces are listed in table 2.1. Note that a constrained, projective reconstruction can only incorporate projective invariant constraints. Furthermore, a constrained reconstruction using affine properties needs a known plane at infinity. Similarly, for a constrained reconstruction including metric or Euclidean constraints, the cameras have to be calibrated. This shows that the tasks of self-calibration and constrained reconstruction are tightly coupled. In general, scene constraints can be enforced or used in the reconstruction process in three different ways:

- Use scene constraints in the calibrated or uncalibrated reconstruction process (constrained reconstruction).

- Use scene constraints in self-calibration, which upgrades a projective to an affine or Euclidean one.
- Enforce scene constraints on a reconstructed 3D model in a post-processing step.

The task of constrained reconstruction is beyond the scope of this thesis. Since it has a long history in computer vision and photogrammetry, we only review some methods and systems.

First consider the task of constrained reconstruction. With pre-calibrated cameras, Shum et al. (1998) and Robertson and Cipolla (2002) presented reconstruction methods which use many different metric scene constraints, like lines with known direction. The task of constrained, uncalibrated reconstruction using 3D points and 3D planes was investigated by Bartoli et al. (2001b) for the 2-view case. The basic idea of this approach is to represent planes, points and cameras in an optimal way, so that the condition that certain points lie on certain planes is fulfilled and that the reprojection error of 3D points is optimized. This work was extended to multiple views in (Bartoli et al., 2001a). Sturm and Maybank (1999) investigated the case of multiple points lying on different, multiple planes seen in one view of a calibrated camera. They demonstrated, that points and planes can be obtained simultaneously by factorization from the condition that certain points lie on certain planes

The second idea is to use the scene constraints for (self-)calibration. This was discussed in (e.g. Caprile and Torre, 1990; Liebowitz and Zisserman, 1999; Svedberg and Carlsson, 1999; Bondyfalat et al., 2001). Caprile and Torre's (1990) method is based on three orthogonal vanishing points and will be explained in more detail in sec. 5.1.2. The calibration information may be used to upgrade a projective reconstruction to a metric reconstruction. However, the metric reconstruction might not fulfill all given scene constraints perfectly. Usually a post-processing step is needed to enforce all scene constraints.

The third approach of enforcing the constraints on the reconstructed 3D model in a post-processing step can be achieved in many different ways. Robertson and Cipolla (2000) expressed the scene constraints of a 3D point reconstruction as linear equations. These equations may be stacked into a linear system which gives a new 3D point reconstruction that satisfies the constraints. This idea will be explained later in more detail. An alternative way of incorporating scene constraints is to consider the reconstruction process as a model based recognition task. Models are defined using a set of primitives, such as windows for buildings. Obviously, these models already satisfy the scene constraints, like orthogonality. Model based reconstruction approaches are often based on a coarse unconstrained 3D reconstruction. In the field of architecture, two popular system are Façade (Debevec et al., 1996) and the commercial product Canoma (Canoma, n.d.). Werner and Zisserman (2002) automated the Façade system. Recently, Dick and coauthors (2001, 2002) used a Bayesian framework to tackle the model based reconstruction problem.

A further aspect of applying scene constraints is the autonomy of the reconstruction system. The reconstruction techniques we have described in sections 3.1 to 3.4 are solely based on matched image features. Many techniques, see (Tell, 2002) for an overview, have been presented to solve the matching process automatically and robustly. Chapter 8 will introduce such a system. In contrast, most scene constraints have to be specified explicitly

by a user in an interactive system. Such interactive systems might be of particular interest for difficult reconstruction tasks like large scale environments, e.g. cities. The task is then to produce a high quality reconstruction with a minimum of user interaction.

Finally, let us analyze the main scene constraints for points, lines and planes. In particular we will investigate which of these constraints are linear in the unknown point, line or plane parameters. The main observation will be that all incidence relationships of points, lines and planes are linear. These constraints can be used for our linear system in eqn. 3.110. This gives a simple, yet, powerful constrained reconstruction method for the reference plane case. Note that the linear reference plane method based on camera constraints (sec. 3.5.1) can *not* integrate scene constraints since 3D features are reconstructed separately. However, one has to keep in mind, that the reference plane has to be the correct plane at infinity if affine invariant constraints are used. For the use of metric (or Euclidean) invariant properties, the calibration matrix K_i and the rotation R_i of a camera P_i have to be known, i.e. the infinite homography has to be $H_i^\infty = K_i\, R_i$.

Projective invariant constraints

Let us represent a finite 3D point by $\bar{\mathbf{X}}$, a 3D line by two planes $\mathbf{\Pi}_l = (\mathbf{n}, l)$ and $\mathbf{\Pi}'_l = (\mathbf{n}'_l, l')$ (sec. 3.3.1) and a 3D plane as $\mathbf{\Pi} = (\mathbf{n}, d)$. We have seen that the orientation of a lines ($\mathbf{n}_l$ and $\mathbf{n}'_l$) and planes ($\mathbf{n}$) may be derived directly from the known reference plane. Therefore only the parameters l, l' and d are unknown.

The most interesting projective invariant constraints are the incidence relationships of points, lines and planes. We will see that these constraints are linear in the unknown feature parameters. The scene constraint of a 3D points $\bar{\mathbf{X}}$ lying on a 3D plane $\mathbf{\Pi}$ is:

$$\mathbf{n}^T\, \bar{\mathbf{X}} \;+\; d = 0 \;. \tag{3.114}$$

A 3D point $\bar{\mathbf{X}}$ lying on a 3D line $\mathbf{L}$ gives two scene constraints,

$$\mathbf{n}^T\, \bar{\mathbf{X}} \;+\; l = 0 \quad \text{and} \quad \mathbf{n}'^T\, \bar{\mathbf{X}} \;+\; l' = 0 \;. \tag{3.115}$$

Finally, the case of a 3D line $\mathbf{L}$ lying on a 3D plane $\mathbf{\Pi}$ can be investigated using the 4×3 matrix $M = [\mathbf{\Pi}\; \mathbf{\Pi}_l\; \mathbf{\Pi}'_l]$, which comprises of three planes (see eqn. 3.65). The condition that $\mathbf{L}$ lies on $\mathbf{\Pi}$ means that M has rank 2. This implies that the following 3 determinants have to vanish:

$$\begin{vmatrix} \mathbf{n}_x & \mathbf{n}_{lx} & \mathbf{n}'_{lx} \\ \mathbf{n}_y & \mathbf{n}_{ly} & \mathbf{n}'_{ly} \\ d & l & l' \end{vmatrix} = 0 \;, \quad \begin{vmatrix} \mathbf{n}_x & \mathbf{n}_{lx} & \mathbf{n}'_{lx} \\ \mathbf{n}_z & \mathbf{n}_{lz} & \mathbf{n}'_{lz} \\ d & l & l' \end{vmatrix} = 0 \;, \quad \begin{vmatrix} \mathbf{n}_y & \mathbf{n}_{ly} & \mathbf{n}'_{ly} \\ \mathbf{n}_z & \mathbf{n}_{lz} & \mathbf{n}'_{ly} \\ d & l & l' \end{vmatrix} = 0 \;. \tag{3.116}$$

The determinants are linear in the unknown parameters l, l' and d.

Consequently, the linear constraints in eqns. 3.114, 3.115 and 3.116 may be incorporated into the linear system in eqn. 3.110.

Affine invariant constraints

An important affine invariant property is parallelism. Constraining 3D lines or planes to be parallel means that their orientations must be adjusted. However, we have seen that the plane at infinity, i.e. the reference plane, provides the orientation of planes and lines (see sections 3.3.1 and 3.4.1). For the linear system in eqn. 3.110 we have chosen a minimal representation of the features, which includes that the orientation of lines and planes is fixed. The only freedom is their position in space. Therefore, the constraint of 2 (or more) lines or planes being parallel can *not* be included in the linear system in eqn. 3.110.

Furthermore, an over-parameterized representation of a line or plane gives a non-linear constraint on the unknown parameters. Let two lines $\mathbf{L}_1$ and $\mathbf{L}_2$ be represented by two finite points, i.e. $\mathbf{L}_1 = (\bar{\mathbf{X}}_1, \bar{\mathbf{X}}_1')$ and $\mathbf{L}_2 = (\bar{\mathbf{X}}_2, \bar{\mathbf{X}}_2')$. The condition that both lines are parallel may be expressed as

$$\bar{\mathbf{X}}_1 - \bar{\mathbf{X}}_1' \;=\; \lambda\,(\,\bar{\mathbf{X}}_2 - \bar{\mathbf{X}}_2') \quad . \tag{3.117}$$

After eliminating the unknown scalar λ, this constraint is non-linear in the unknown point coordinates. Therefore, such a scene constraint can only be handled by non-linear process.

Metric & Euclidean invariant constraints

There are many important metric and Euclidean properties like lengths, angles and ratios of lengths. However, all of them are non-linear in the unknown feature parameters except of constraints concerning known 3D feature. For example, the constraint that the distance between two points $\bar{\mathbf{X}}_1$ and $\bar{\mathbf{X}}_2$ is α may be expressed as

$$\bar{\mathbf{X}}_1 - \bar{\mathbf{X}}_2 \;=\; \alpha\,\frac{\bar{\mathbf{X}}_1 - \bar{\mathbf{X}}_2}{||\bar{\mathbf{X}}_1 - \bar{\mathbf{X}}_2||_2} \quad , \tag{3.118}$$

where $||\cdot||_2$ represents the Euclidean norm. Eqn. 3.118 is obviously non-linear in the unknown point parameters. Let us assume that some 3D features are given: A point $\bar{\mathbf{X}}_g$, a line $\mathbf{L}_g$ as $\mathbf{\Pi}_l = (\mathbf{n}_l, l_g)$, $\mathbf{\Pi}_l' = (\mathbf{n}_l', l_g')$ and a plane as $\mathbf{\Pi}_g = (\mathbf{n}, d_g)$. The condition that a certain 3D feature is identical with a given 3D feature can be formulated as

$$\bar{\mathbf{X}} = \bar{\mathbf{X}}_g \;\;,\quad l = l_g \;,\; l' = l_g' \;\text{ and }\quad d = d_g \quad . \tag{3.119}$$

These constraints are, in contrast to all previous constraints, non-homogeneous. Consequently, incorporating these constraints means that our linear system is non-homogeneous as well, which, however, can still be solved using SVD (Hartley and Zisserman, 2000). Note that the orientation of lines and planes is fixed.

In the following we will briefly discuss how to formulate constraints on lines and planes so that their orientation may be also adapted to a given orientation. These constraints were used for instance by Robertson and Cipolla (2000) to determine linearly a constrained 3D point reconstruction from an unconstrained reconstruction. If a 3D line is represented by

views	general – points		general – lines		ref plane – points	ref plane – lines	planes
	linear	non-lin.	linear	non-lin.	linear	linear	linear
2	8	7	–	–	2	5	1
3	7	6	13	9	2	4	1
4	6	6	9	8	2	4	1
> 4	–	–	–	–	2	4	1

Table 3.1. Summary of the minimum number of 3D points, lines or planes needed to obtain directly a projective reconstruction of the cameras and optionally of the 3D features in closed-form. General configurations are compared with reference plane configurations.

two points $\bar{\mathbf{X}}$ and $\bar{\mathbf{X}}'$, its orientation may be adapted to a given orientation $\bar{\mathbf{X}}_g$ by the constraint

$$(\bar{\mathbf{X}} - \bar{\mathbf{X}}') \times \bar{\mathbf{X}}_g = \mathbf{0} \ . \tag{3.120}$$

This gives 2 linear independent equations in the unknown point coordinates. The same applies to a 3D plane, which is represented by three points $\bar{\mathbf{X}}, \bar{\mathbf{X}}'$ and $\bar{\mathbf{X}}''$. A plane has a given orientation, i.e. the normal $\mathbf{n}_g$, if the following two conditions hold

$$(\bar{\mathbf{X}} - \bar{\mathbf{X}}') \mathbf{n}_g = 0 \quad \text{and} \quad (\bar{\mathbf{X}} - \bar{\mathbf{X}}'') \mathbf{n}_g = 0 \ . \tag{3.121}$$

In man-made environments, 3D lines and 3D planes are often known to be vertical or horizontal to a ground plane. This gives constraints of the form 3.120 and 3.121 with $\bar{\mathbf{X}}_g$ and $\mathbf{n}_g$ as $(0, 0, 1)^T$ and $(x, y, 0)^T$ respectively.

Let us summarize the projective and Euclidean constraints which are linear in the unknown feature parameters. Probably the most important constraints are the (projective) incidence relationships of features (eqn. 3.114, 3.115 and 3.116). Furthermore, the Euclidean (non-homogeneous) constraints concerning known 3D features are also linear (eqn. 3.119). Finally, if lines and planes are represented by points, their orientation may be constrained to a given orientation (eqn. 3.120 and 3.121). All these constraints may be incorporated in our DRP method for different feature types (sec. 3.5.1). This gives a very powerful constrained reconstruction method for the reference plane case.

3.6 Summary

We have compared general configurations with reference plane configurations of multiple views and multiple features like points, lines and planes. The relationship between cameras and features is *bi-linear* in the general case. If a reference plane is known, this relationship becomes *linear* in an affine space where the reference plane represents the plane at infinity. This makes it possible to simultaneously reconstruct *all* cameras and *all* features in a *single* linear system. *This system embodies the core of our direct reference plane (DRP) reconstruction approach and represents the main and novel contribution of the thesis* (Rother and Carlsson, 2002a). Furthermore, the linear system handles missing data naturally. Features on the reference plane can be reconstructed directly, since their positions are independent

of the cameras' positions. Additionally, the technique permits the simple incorporation of incidence relationships, e.g. a point lies on a plane, and constraints concerning known 3D features, for instance the coordinates of a 3D line being known. Moreover, we have seen that with a known reference plane, i.e. plane at infinity, the orientation of 3D features may be determined directly. Consequently, a 3D line can be represented by only 2 parameters (4 in general) and a 3D plane by one parameter (3 in general).

For general configurations we have reviewed three well known approaches of solving the reconstruction problem: using (a) camera constraints, (b) structure constraints, and (c) factorizing cameras and features from a measurement matrix. We have seen that the assumption of having a known reference plane simplifies considerably these approaches (Triggs, 2000). In particular, the camera constraints can be used to reconstruct *all* cameras simultaneously from a linear system (Hartley et al., 2001). A comparison of these approaches, and our direct reference plane approach, for both general and reference plane configurations is carried out in the next chapter 4.

With the assumption of no missing data, table 3.1 depicts the minimum number of 3D features needed to reconstruct directly the cameras and optionally the 3D features in closed-form. The first observation is that in the general case more 3D features are needed. Secondly, the reference plane case offers simple, linear methods which give a unique solution for non-critical configurations. Thirdly, in the general case there is no direct, closed-form camera reconstruction method for more than 4 views.

All these different aspects of the above discussion suggest that reference plane reconstruction methods are superior to general reconstruction methods. The experiments in chapter 6 will reveal that for difficult reference plane scenarios this is indeed the case. However, one has to keep in mind that reference plane methods heavily rely on the quality of the determined reference plane, i.e. the infinite homographies. How this restriction influences the reconstruction results is a further important aspect investigated in the experiments.

Chapter 4

Structure and Camera Recovery – A Review and Comparison

In the previous chapter we discussed different approaches of solving the reconstruction problem for point, line and plane features. The emphasis was *not* to review practical methods, but, to review and categorize theoretical approaches of solving the reconstruction problem. These different categories were analyzed for general configurations and reference plane configurations. In order to simplify the presentation, some assumptions were made, like image features being match correctly or specific 3D points forming a projective basis in $\mathcal{P}^3$. In practice, these assumptions are either violated or might lead to numerically unstable methods in the presence of noisy input data. In this chapter, we review practical reconstruction methods which tackle these "real world" problems. The different methods are classified according to the categories presented in the previous chapter. In particular, for the general configurations (sec. 4.2) we will review methods based on camera constraints (sec. 4.2.1), structure constraints (sec. 4.2.2) and factorization (sec. 4.2.3). For reference plane configurations (sec. 4.3), we discuss methods which use our direct reference plane approach (sec. 4.3.1), camera constraints (sec. 4.3.2) and factorization (sec. 4.3.3). This chapter is structured differently to the previous chapter. The emphasis of the previous chapter was to compare general configurations with reference plane configurations. In this chapter, methods for these two types of configurations are divided into separate sections. The emphasis is here to review and compare methods which assume the same scenario, i.e. either general or reference plane. Furthermore, the *main focus* of this chapter is a not only to review, but also to *compare* practical reconstruction methods. In order to achieve this, several different criteria will be introduced (sec. 4.1). As was seen in the previous chapter, the different categories have different properties which are inherit by any reconstruction algorithm of a certain category. For example, factorization methods do not allow missing data which is a problem that has to be addressed by any method of this category. Therefore, in order to simplify the comparative study on the essential aspects, solely the different categories will be compared.

Before reviewing different reconstruction methods, we would like to point out that reference plane methods have to address the task of determining a reference plane, i.e. the infinite homography of each camera, in a pre-processing step. We will describe in the next chapter that there are many different ways of solving this problem. For example, for general scenes the assumption of affine cameras is sufficient to obtain the infinite homography of each camera. Consequently, reference plane methods can be applied to general configurations as well. Note that the converse is also true. General reconstruction methods can obviously be used as well for reference plane configurations. However, in order to keep a coherent structure throughout the thesis, the general case and the reference plane case are discussed separately in the following.

4.1 Criteria for Reconstruction Methods

In order to develop an automatic reconstruction system for practical applications, certain design issues have to be considered. These issues are summarized in the following nine criteria. The first eight criteria apply to both the general and the reference plane cases. The last criterion applies only to methods which assume a known reference plane.

Missing data
We saw in chapter 1, that in most "real world scenarios" some 3D points are not visible in all views. Therefore, it is inevitable that a reconstruction algorithm has to deal with the problem of missing data. In practice, the amount of missing data may vary: all features are visible in all views (no missing data), some features are visible in all views, a reference plane is visible in all views or no feature is visible in all views. Obviously a minimum number of features have to be visible in some views (e.g. Quan et al., 1999).

Number of used features and cameras
Ideally a multi-view reconstruction algorithm should compute all features and all cameras simultaneously. The category of factorization algorithms comprises of such closed-form methods. Sequential algorithms, which exploit only a subset of cameras or features at a time are potentially inferior for the case of noisy and missing input data.

Robustness (outlier rejection)
If the image features in multiple views are matched automatically (see overview Tell, 2002), it may *not* be assumed that all matches are correct. Mismatched features are denoted as outliers and might have a bad influence on the result of any reconstruction method. For tracked image features, the proportion of outliers is in practice very low. In general, any reconstruction algorithm which uses automatically matched image features needs an outlier rejection process. A popular strategy for outlier rejection is to apply a method called "random sample consensus" (RANSAC) by Fischler and Bolles (1981) as a possible preprocessing step (e.g. Torr, 1995). The general idea of RANSAC is to randomly select a minimal dataset to compute a candidate solution. The model can be used to classify the remaining data into in- and outliers. The ratio of in- and outliers represents the quality of the estimated model. After a certain number of iterations, the best candidate is selected.

Objective function
In practice, the image measurements are corrupted by noise, which means that the projection **x** of a 3D point **X** by a camera P does not satisfy exactly the projection relation $x \sim P\,X$. The task of any reconstruction algorithm is to find a solution for the projection relation for multiple, unknown features and cameras. However, this process can be achieved by applying different objective (cost or error) functions. A meaningful geometric error is the distance between the image point **x** and its reprojection $P\mathbf{X}$:

$$|| \bar{\mathbf{x}} - \overline{P\mathbf{X}} ||_2 \quad , \tag{4.1}$$

where $|| \cdot ||_2$ is the Euclidean distance. The objective function is then the summation of this geometric error for all points visible in all possible views. If the image noise is assumed to be isotropic zero mean Gaussian, this gives the Maximum Likelihood solution. For our direct reference plane method in sec. 3.2.2, the transformed projection relations in 3.13 have been used. Therefore, this method minimizes a suboptimal algebraic error. A possible post-processing step for a suboptimal method is bundle adjustment (e.g. Triggs et al., 1999), which is a standard method in photogrammetry (Slama, 1980). It is a non-linear optimization with the geometric error in eqn. 4.1 as its objective function. Since most reconstruction methods do not minimize the geometric error in eqn. 4.1, it is a common post-processing step. The reason why bundle adjustment is not used directly for reconstructing unknown features and cameras is that non-linear methods need a good initial guess.

Combining features and scene constraints
Most of the existing reconstruction methods are formulated for point features. However, we have motivated in the previous chapter that a natural and practical issue is the extension to lines, planes or other features. A further extension is the use of scene constraints as discussed in sec. 3.5.2.

Distinguished cameras or features
Ideally a multi-view reconstruction method should treat all image features and all views uniformly. This is not always the case. In sec. 3.2.1 the projective basis in $\mathcal{P}^3$ was represented by specific 3D points, which are therefore treated differently to the remaining points. The reconstruction is therefore more sensitive to noisy projections of these *distinguished* features than other points. Another example is an algorithm which processes multiple views sequentially. Such a process might perform differently depending on the order of the images.

Specifying the projective basis
In sec 3.2.1 the projective basis was specified by 3D points. Alternatively, choosing two cameras matrices as in eqn. 3.29 uniquely defines the projective space. Ideally, a reconstruction method should not choose a specific projective basis, since it might be an unfortunate choice. For example, the measurement matrix W for factorization methods represents all possible projective bases by the transformation H, since $W = P\,X = PH^{-1}HX$.

Computational effort and implementational complexity
Obviously, a reconstruction method should be as fast as possible. A faster, though inferior, method might produce the same result after bundle adjustment as a slower, but superior, method. A further aspect, probably worth mentioning, is how difficult is is to implement.

Specific reference plane criteria
Reconstruction methods which exploit the knowledge about a given reference plane might treat features or cameras which lie on this distinguished plane differently to all other features. We saw in sec. 3.2.1 that features on the reference plane do not give any information about the cameras' position and may therefore be reconstructed separately and directly. Consequently, only the features which do not lie on the reference plane are needed to determine the multi-view geometry. However, especially for "nearly flat" scenes, such a separation of features might affect the quality of the result. Therefore, ideally *all* features are reconstructed simultaneously. Furthermore, it is worth specifying if a certain reference plane method applies as well to cameras where the projection centre lies on the reference plane.

4.2 General Configurations

The task of uncalibrated camera and structure recovery for general scenes has been one of the main subjects in the field of computer vision in the last decade. The result is a long list of publications. In the following, only some important publications are reviewed. These are grouped into the 3 categories: *camera constraints* (sec. 4.2.1), *structure constraints* (sec. 4.2.2) and *factorization* (sec. 4.2.3). The basic, theoretical concepts of these methods were presented for point features in sec. 3.2.3 (camera and structure constraints) and sec. 3.2.4 (factorization). In the following, we concentrate on techniques which are capable of reconstructing "large-scale" environments, e.g. buildings, from a set of images with a wide baseline. Therefore, methods which are based on small baseline images using optical flow or recursive filters, e.g. Kalman filer, to update a 3D reconstruction sequentially (for an overview (Zucchelli, 2002)) are not reviewed here.

4.2.1 Camera Constraints

Constraints which involve only cameras parameters and image measurements are called camera constraints. Most existing uncalibrated reconstruction algorithms are based on camera constraints. The most likely reason is that camera constraints provide a linear and closed-form solution for *multiple* features visible in a limited number of views via the multifocal tenors. However, the main drawback is that multifocal tensors do not exist for more than four views. Therefore, it then becomes necessary to stitch multifocal tensors of subsets of views together.

The camera constraints of points, lines and planes visible in $2, 3$ and 4 projective views were discussed in detail in sec. 3.2.3, 3.3.3 and 3.4.3. Faugeras (1992) and Hartley (1992)

simultaneously introduced the fundamental matrix for 2 views. This was the starting point for the study of the tri- and quadri-focal tensor for point and line features (e.g. Hartley, 1994; Shashua, 1994; Triggs, 1995; Hartley, 1995; Carlsson, 1995; Faugeras and Mourrain, 1995; Heyden, 1998). Furthermore, other feature types have been investigated like planes (e.g. Luong and Faugeras, 1993; Szeliski and Torr, 1998) and conics (e.g. Kahl and Heyden, 1999; Schmid and Zisserman, 2000). A summary of the camera constraints can be found in the books of Hartley and Zisserman (2000) and Faugeras and Luong (2001). On the basis of the camera constraints, the geometry of multiple views may be formulated (e.g. Heyden and Åström, 1995a; Faugeras and Mourrain, 1995; Luong and Viéville, 1996; Heyden and Åström, 1995b). If the input data consists of outliers, robust estimator techniques like RANSAC (Fischler and Bolles, 1981) can be applied to robustly determine the camera geometry (e.g. Torr, 1995; Torr and Zisserman, 1997; Torr and Murray, 1997).

Robust and automatic projective multi-view reconstruction systems have been presented by Beardsley et al. (1996) and Fitzgibbon and Zisserman (1998). Beardsley et al.'s (1996) linear reconstruction approach is sequential and based on an "intersection-resection" scheme. A new view is integrated into an existing 3D reconstruction by *resection*. In the next step, new 3D structure may be derived from the new view and some other views by *intersection*. Finally, a bundle adjustment step might be applied. The initial reconstruction may be obtained from the fundamental matrix of 2 views. Fitzgibbon and Zisserman (1998) improved this method by introducing a hierarchical framework which optimally distributes the error over the set of images. The set of all views is divided into manageable subsets of 2 or 3 images, where a closed-form solution is computed. On the basis of overlapping structure and cameras, two subsets may be merged into a larger subset. This merging process is done in a hierarchical fashion to spread the error equally over the set of views. Additionally, the bundle adjustment algorithm may be applied at various stages. These two methods have been successfully applied to sequences of images. However, in order to handle longer image sequences, it is recommended to skip intermediate frames with a small baseline (e.g. Nistér, 2000a). Variations of these ideas have been discussed and used, together with self-calibration methods, to build complete 3D metric reconstructions from image sequences (e.g. Heyden and Åström, 1995b; Pollefeys et al., 1998; Pollefeys, 1999; Nistér, 2000b; Nistér, 2001; Georgescu and Meer, 2002; Pollefeys et al., 2002). An alternative method of merging subsets of $2, 3$ or 4 views into a complete multi-view reconstruction has been suggested by Triggs (1997b) for projective cameras and (Kahl and Heyden, 1999) in the affine case. These methods are based on the so-called joint image closure constraints, which represent a bi-linear relationship between matching tensors and cameras. These constraints make it possible to reconstruct *all* cameras directly and linearly from a set of known bi-, tri- or quadri-focal tensors. The drawback of the projective closure constraints method is that the multi-view tensors have to be scaled correctly, which is a non-trivial task.

The application of these multi-view reconstruction methods is obviously not restricted to a sequence of images. They might be applied as well to a set of *unorganized* images with a considerable amount of missing data for e.g. large scale reconstructions like city blocks. However, due to a lack of available features, such applications demand a lot of the "sequential" multi-view reconstruction methods. We will see in sec. 6.1.3 that a "sequential"

reconstruction method based on camera constraints can fail for difficult scenarios where a large scale environment is reconstructed from a few manually matched image features of very wide baseline images.

The main characteristics of projective multi-view reconstruction methods based on camera constraints may be summarized as following:

Advantages

- Missing data is treated naturally, since a subset of $2, 3$ or 4 views contains in general very little missing image data.
- It is possible to design a robust reconstruction method.
- Since most of the camera constraint methods are sequential, the geometric error in 4.1 may be minimized using bundle adjustment after each "step".
- Other features like lines and planes may be integrated.
- The computational effort of a *linear* reconstruction method (e.g. Beardsley et al., 1996) is small. However, intermediate *non-linear* processes like bundle adjustment increase the computation time considerably. A simple version of a robust multi-view reconstruction method is straightforward to implement.

Drawbacks

- Only a limited number $(2 - 4)$ of cameras may be computed in closed-form. On the basis of the $2, 3$ and 4 view tensors, the joint image closure constraints may be used to derive all cameras in closed-form.
- Since not all cameras are considered simultaneously, the result depends on the ordering of the images.
- The projective space is in general defined by the first subset of $2-4$ views, e.g. using eqn. 3.29 in the 2-view case.

4.2.2 Structure Constraints

The dual counterpart to the camera constraints are the structure constraints. They involve only 3D features and image measurements and were introduced in sec. 3.2.3. The structure constraints only exist for point features. These constraints make it possible to switch the role of points and cameras. Consequently, all cameras may be reconstructed on the basis of $6, 7$ or 8 3D points. In order to achieve this, the image points of four reference points must represent the canonical image basis defined in eqn. 3.3.

The dualization of reconstruction algorithms was investigated by Hartley and Debunne (1998) from both a theoretical and practical point of view. They found that the main drawback of dual reconstruction algorithms is the fact that a specific, canonical image basis has to be chosen. It was shown in (Hartley, 1997) that it is essential to choose a "normalized"

basis in each image. Later, Schaffalitzky et al. (2000) represented a robust reconstruction method for multiple views based on structure constraints. They demonstrated a simple algorithm for reconstructing 6 points in $m \geq 3$ views which minimizes the geometric error in 4.1. This algorithm for the minimal case may be used as a "search engine" for RANSAC to robustly compute all cameras and finally all 3D points. The main drawback of this approach is that a sufficient number (≥ 6) of points must be visible in all views.

Let us summarize the main characteristics of this category of reconstruction method:

Advantages

- All cameras are computed simultaneously.
- Dual reconstruction methods may be designed robustly if a sufficient number ($\gg 6$) of points are available for RANSAC.
- It is possible to minimize a meaningful geometric error.
- The computational effort is small. All cameras may be obtained in a first step and all points in a second step by triangulation. Note that no pre-processing step like outlier rejection is needed. Furthermore, the robust 6 point algorithm by Schaffalitzky et al. (2000) is "fairly" simple to implement.

Drawbacks

- Only a limited number of 6 to 8 points may be computed simultaneously.
- Missing data is allowed only partly: A sufficient number (≥ 6) of points have to be visible in all views.
- An extension to other feature types is not possible.
- Dual reconstruction methods distinguish the $6 - 8$ reference points.
- The projective basis is specified by the reference points.

4.2.3 Factorization

The idea of factorizing cameras and structure from a measurement matrix was introduced by Tomasi and Kanade (1992) (sec. 3.2.4). Reid and Murray (1996) showed that under the assumption of isotropic zero mean Gaussian noise independent and equal for each image point, factorization achieves a Maximum Likelihood affine reconstruction, which means that the reprojection error (eqn. 4.1) is minimized. Irani and Anandan (2000) and Morris and Kanade (1998) extended the factorization method by representing point features by a correlated, directional uncertainty model. Hence, the objective function minimizes a covariance-weighted squared-error, i.e. the Mahalanobis distance. Affine factorization was extended to para-perspective cameras by Poelman and Kanada (1994). Sparr (1996) presented a method, similar to factorization, based on the idea of "kinetic depths" introduced in (Sparr, 1994).

A perspective version of the factorization method was first presented by Sturm and Triggs (1996). In contrast to the case of parallel projection, the measurement matrix contains unknown projective depths, i.e. the "unscaled" measurement matrix (sec. 3.2.4). However, these scale factors are not arbitrary, they must satisfy certain internal constraints. Sturm and Triggs (1996) suggested determining the projective depths sequentially from the fundamental matrices of pairs of views. However, this step is not trivial since computing the fundamental matrices is equivalent to solving the complete reconstruction problem. An alternative and more simple way is to determine the projective depths iteratively (Triggs, 1996; Qian and Medioni, 1999; Heyden et al., 1999; Hartley and Zisserman, 2000). All projective depths are initially set to 1, which corresponds to affine cameras. Given an estimate of the projective depths, factorization is applied to obtain the cameras and 3D structure. The projective depths can then be re-estimated by reprojection of the reconstructed 3D points. This approach produces good results (e.g. Heyden et al., 1999; Hartley et al., 2001). However, if the number of points or views is large, this method is very time consuming.

The main drawback of all factorization methods is that it is awkward to handle missing data. Tomasi and Kanade (1992) suggested choosing the largest full submatrix of the measurement matrix for an initial reconstruction. The missing elements may be derived from the initial reconstruction by intersection. However, searching for this submatrix is an NP-hard problem. Jacobs (1997) improved this method of filling in missing data, by fitting a rank-3 matrix to the measurement matrix. This fitting is achieved by considering randomly small submatrices of the full measurement matrix. Consequently, no image data is distinguished as in (Tomasi and Kanade, 1992). Recently, Martinec and Pajdla (2002) suggested combining Jacobs (1997) method of handling missing data and Sturm and Triggs (1996) method of deriving the projective depths from known epipolar geometry. This gives a general projective factorization method which handles occlusions.

Let us summarize the main characteristics of factorization methods:

Advantages

- All cameras and all point features are reconstructed simultaneously.
- For affine cameras, the geometrically meaningful reprojection error is minimized. Even more advanced error models might be incorporated in the affine factorization framework. In the projective case, an algebraic error is minimized.
- All cameras and features are treated uniformly when there is no missing data.
- No specific projective basis is chosen.
- With of no missing data, the affine factorization method is very simple to implement. For projective cameras and missing data, factorization methods become more complex.

Drawbacks

- Missing data must be "hallucinated" by a non-trivial pre-processing step.

- Outlier rejection requires an additional pre-processing step. However, this issue has not been addressed to our knowledge.
- Using lines or other features together with point features or separately is possible in the affine case (e.g. Quan and Kanade, 1997; Kahl and Heyden, 1999), but, not in a one-step factorization as for point features only (sec 3.3.4). Alternatively, lines or planes may be represented by 2 or 3 3D point features (e.g. Triggs, 1996; Morris and Kanade, 1998). However, this means that for lines some multi-view tensors have to be known. Projective factorization of planes and cameras from planar homographies only has been suggested by (Shashua and Avidan, 1996; Triggs, 2000; Rother et al., 2002).
- In the projective case, the "projective depths" have to be derived in a non-trivial pre-processing step.
- Factorization methods are in general slow, since the size of the measurement matrix grows linearly with the number of features and cameras, i.e. $W = 3m \times n$, for n points and m cameras.

4.3 Reference Plane Configurations

The idea of investigating multi-view structure and camera recovery from a plane+parallax point of view has a long history (e.g. Carlsson and Eklundh, 1990; Luong and Faugeras, 1993; Shashua and Navab, 1994; Sawhney, 1994; Kumar et al., 1994; Kumar et al., 1995; Oliensis, 1995; Boufama and Mohr, 1995; Irani and Anandan, 1996; Shashua and Avidan, 1996; Avidan and Shashua, 1998; Irani et al., 1998; Szeliski and Torr, 1998; Weinshall et al., 1998; Criminisi et al., 1998; Cross et al., 1999; Johansson, 1999; Triggs, 2000; Sturm, 2000; Xu et al., 2000; Rother and Carlsson, 2001; Hartley et al., 2001; Rother and Carlsson, 2002b; Irani et al., 2002; Robertson and Cipolla, 2002; Rother et al., 2002; Rother and Carlsson, 2002a). In the following we will review those works which exploit a *known* reference plane for the reconstruction task. Publications which discuss the task of *determining* the reference plane are reviewed in chapter 5. Most of the reference plane methods which address the task of 3D reconstruction have already been discussed in sec. 3.2.3. They use either camera or structure constraints to solve the problem. In the following this discussion is briefly summarized. Moreover, we review reference plane methods belonging to the 3 categories: *Direct reference plane approach* (sec. 4.3.1), *camera constraints* (sec. 4.3.2) and *factorization* (sec. 4.3.3). The basic, theoretical concepts of these methods were presented for point features in sec. 3.2.2 (direct reference plane method), sec. 3.2.3 (camera constraints) and sec. 3.2.4 (factorization). In particular we compare our direct reference plane method (Rother and Carlsson, 2002a), with the factorization method of Triggs (2000) and the camera constraints method of Hartley et al. (2001). In our opinion, these are the best techniques from each category.

The papers of (Irani and Anandan, 1996; Irani et al., 1998; Weinshall et al., 1998; Criminisi et al., 1998) analyzed the structure and camera constraints for 2 and 3 views.

The developed techniques can be used for different tasks: verifying the consistency of the parallax geometry, computing the Euclidean height from the reference plane or view synthesis (Irani et al., 2002). These papers are based on early work (Sawhney, 1994; Kumar et al., 1994; Kumar et al., 1995), which studied shape recovery from projective and affine cameras using plane+parallax. Note that these methods and its applications were discussed in sec. 3.2.3 in more detail. Another interesting early paper by Shashua and Navab (1994) investigates the 3D scene including a reference plane. They introduce the term "relative affine structure", which represents the affine 3D space with the reference plane as the plane at infinity. Furthermore, they propose a multi-view, point-based reconstruction method which computes first the epipolar geometry and secondly the 3D structure depending on a reference view. Cross et al. (1999) investigated the multi-view geometry of smooth objects together with a reference plane. The task of reconstructing multiple planes and cameras (see sec. 3.4) has been studied in (Luong and Faugeras, 1993; Shashua and Avidan, 1996; Szeliski and Torr, 1998; Johansson, 1999; Sturm, 2000; Xu et al., 2000).

Oliensis (1995) (also Oliensis, 1999; Oliensis and Genc, 1999) applied the idea of linearizing the reconstruction problem from known infinite homographies to continuous image sequences. We illustrated in fig. 2.3(c) that two images of a purely rotating camera are related by an infinite homography. This will be proved formally in sec. 5.2.4. The basic assumption in Oliensis's (1995) work is a small movement of the camera between successive frames, i.e. small baseline. This means that the infinite homography may be determined approximately. As was seen in sec. 3.2.1, this information is enough to determine a linear relationship between points and cameras. This may be used to initialize an iterative reconstruction algorithm (Oliensis, 1999). Furthermore, if the camera calibration is known, i.e. K, the rotation R can be determined since $H^\infty = KR$, and a Euclidean reconstruction may be obtained (Oliensis and Genc, 1999).

We now compare three different reference plane reconstruction methods in detail: Rother and Carlsson (2002a)(sec. 4.3.1), Triggs (2000)(sec. 4.3.3) and Hartley et al. (2001)(sec. 4.3.3). Since all approaches compute the solution from a large measurement matrix using SVD, their computational effort may be compared. According to Golub and Van Loan (1996) a full SVD of a matrix of size $a \times b$ requires $4a^2b + 8ab^2 + 9b^3$ flops. For some methods it is sufficient to compute solely V and D of the SVD of a matrix $S = UDV^T$. This requires only $4ab^2 + 8b^3$ flops. Specifically, we consider a realistic scenario of $n = 400$ points visible in all $m = 10$ views, a scenario which corresponds roughly to the "house" example in sec. 6.1.3.

4.3.1 Direct Reference Plane Approach

The idea of reconstructing points and cameras simultaneously from one linear system was originated with Rother and Carlsson (2001). Rother and Carlsson (2002b) improved numerical issues of the method (see also (Rother and Carlsson, 2002a)). We call our method the *D*irect *R*eference *P*lane (DRP) method. It was introduced in sec. 3.2.2 for points features and extended to line (sec. 3.3.2) and plane features (sec. 3.4.2) in sec. 3.5.1. For the special case of calibrated cameras with known orientation, the fact that points and cameras have a linear relationship was previously mentioned in (Debevec et al., 1996; Shum

et al., 1998; Robertson and Cipolla, 2000). However, they did not exploit this idea to reconstruct structure and cameras simultaneously. Recently, (Robertson and Cipolla, 2002) used our DRP reconstruction method together with a 2D map for the reconstruction of large-scale architectural environments (sec. 5.1.3).

The main difference between the DRP method and other reference plane approaches presented in this section is that features and cameras are reconstructed simultaneously. However, as explained in sec. 3.2.1, this is only true for features which do *not* lie on the reference plane. Therefore, features on and off the reference plane have to be separated in a pre-processing step. How this task is solved in practice is discussed in chapter 6. Note that this issue is irrelevant if the reference plane is the correct plane at infinity.

Let us discuss the main characteristics of the DRP method:

Advantages

- Missing data is handled naturally, since the projection equations are used directly.
- All features (not on the reference plane) and all cameras are determined simultaneously.
- Since points, lines, planes and cameras give linear projection relations, all three feature types may be reconstructed together with all cameras in a single linear system (sec. 3.5.1). Furthermore, the methods allows the simple incorporation of incidence relationships, e.g. a point lies on a plane, and constraints concerning known 3D features, for instance the coordinates of a 3D line being known (sec. 3.5.2).
- All cameras and features are treated uniformly if the reference plane is known.
- The method may use infinite cameras on the reference plane as well (sec. 6.1).
- The method is simple to implement, especially if no point lies on or close to the reference plane, for instance the reference plane is the correct plane at infinity.

Drawbacks

- The outlier rejection process is a required pre-processing step. RANSAC may be applied here as well. The advantage is here that the DRP method simplifies the non-linear step of determining a minimal multi-view reconstruction. This will be discussed in more detail in chapter 8.
- An algebraic error is minimized. Therefore, a final bundle adjustment step is recommended.
- Choosing the reference plane as the plane at infinity fixes 11 of the 15 degrees of freedom of the projective space. The remaining 4-dimensional space of solutions is represented by the 4-dimensional nullspace of the system matrix. However, this specific projective space biases the linear system, especially for "nearly flat" scenes. 3D points which are very close to the reference plane have very large coordinates. Such a negative effect can be eliminated by weighting the linear equations as discussed in chapter 6.

- Features on and off the reference plane are reconstructed separately.
- The linear system is of size $3mn \times 3(m+n) = 12000 \times 1230$. Since it is sufficient to compute only V of the SVD, this requires approximately 87.5 Gflops. This is slowest of all 3 methods considered here, which is due to the fact that both features and cameras are reconstructed simultaneously.

4.3.2 Camera Constraints

Using the camera constraints for computing all cameras simultaneously was suggested by Hartley et al. (2001), which is based on a brief description in (Hartley and Zisserman, 2000). In contrast to all other reconstruction methods which use either camera or structure constraints (e.g. Shashua and Navab, 1994; Irani and Anandan, 1996; Irani et al., 1998), Hartley et al.'s (2001) method derives all cameras simultaneously. For the case of point features, this approach was explained in sec. 3.2.3. The extension to line (sec. 3.3.3) and plane (sec. 3.4.3) features is straightforward (sec. 3.5.1), since they give linear equations for the camera centres as well.

Most of the characteristic of this method have already been discussed for the general case (sec. 4.2.1). The different and reference plane specific characteristics are:

Advantages

- All cameras are determined simultaneously from camera constraints of points, lines and planes.
- No difference between features on and off the reference plane.
- Hartley et al. (2001) suggested reducing the computation time by reducing the linear system of each subset of views. With the worst case assumption of no missing data, the final linear system after reduction is of size $12\binom{m}{4} \times 3m = 2520 \times 30$, which requires 9.3 Mflops. As with the previous method, only V of the SVD has to be computed. The time required for reducing a 4-view system of size $81 \cdot 400 \times 12$ is 18.6 Mflops. Therefore, the total computation time is $(18.6\binom{m}{4} + 9.3)$ Mflops = 3.9 Gflops. Consequently, this approach is faster than our DRP method. The computation of the point structure in the second stage may be neglected. This is not surprising since only the cameras and not the features are reconstructed. Note that with missing data the computation time is considerably smaller.
- The method may use infinite cameras on the reference plane as well.

Drawbacks

- The features are *not* determined simultaneously with the cameras.
- Since the structure is computed in a post-processing step, no scene constraints may be included in the approach.
- The method is more complicated to implement than the DRP method, since combinations of views have to be considered.

4.3.3 Factorization

Triggs (2000) suggested a point-based factorization method for the reference plane case (see sec. 3.2.4). Furthermore, he introduced line (sec. 3.3.4) and plane (sec. 3.4.4) factorization methods, which can be combined into one system (sec. 3.5.1). Later, Rother et al. (2002) explained the plane-based factorization approach in more detail and presented experimental results. These factorization methods share most of the characteristics of the general factorization approaches listed above. Especially, the problems of missing data and "projective depths" computation have to be addressed. An important aspect of the point-based factorization method is that the heights of the points from the reference plane are reconstructed. Consequently, this method is not applicable for infinite reference planes. For the reference plane case, the factorization methods have the following different or additional characteristics:

Advantages

- All features and all cameras are determined simultaneously. Note that for the integration of lines, the multifocal tensors must be known.
- Features on and off the reference plane are reconstructed simultaneously. This is possible since all information about the features, except their distance to the plane, is derived in advance. The only remaining unknown, i.e. the distance, is a finite value for points on and "close to" the plane. Note, the DRP method uses the inverse depth, i.e. points on the reference plane are at infinity.
- The measurement matrix is, in contrast to the other 2 reference plane methods, small, i.e. $3m \times n = 30 \times 400$. In this case a full SVD has to be performed which requires 0.6 Gflops. Consequently, this is the fastest of all 3 reference plane methods considered here. Factorization might only be slower than Hartley et al.'s (2001) method if a very large number of points is used.

Drawbacks

- Only applicable for finite reference planes.
- Triggs (2000) suggested fixing 14 of the 15 degrees of freedom of the projective space in advance.
- The methods have not been formulated for cameras on the reference plane[1].

[1] It might be possible to reformulate the approach for infinite cameras as well.

4.4 Conclusion

Several practical, multi-view reconstruction methods for general and reference plane configurations were reviewed and compared. The basic, theoretical concepts of these methods were introduced in the previous chapter. The discussion was based on several criteria which "real world" multi-view reconstruction systems have to fulfill. For general configurations, existing methods have been categorized into three categories, methods using camera constraints, structure constraints and factorization methods. The main characteristic of camera constraint methods is that they are sequential. Structure constraint methods have the limitation that a minimum number of 6 points has to be visible in all views. The main drawback of factorization methods is that missing data has to be hallucinated. If a reference plane is known, three different methods representing three different categories were reviewed. These were our direct reference plane (DRP) method (Rother and Carlsson, 2002a), the camera constraint method (Hartley et al., 2001) and the factorization method (Triggs, 2000). All three methods reconstruct the scene directly from a singular value decomposition of a measurement matrix. The DRP method and factorization method compute both features and cameras simultaneously. In contrast to this, the camera constraint method determines *only* the cameras simultaneously. The main drawback of our DRP method is that features on the reference plane have to be reconstructed separately. Note that this is not a problem if the reference plane is the actual plane at infinity. The main disadvantage of the factorization method is that it is *not* applicable for infinite reference planes, and missing data is not treated naturally.

The main conclusion of the comparative study is that each category has its advantages and drawbacks. This is valid for general and reference plane configurations. Therefore, the decision of the best method is application dependent. In section 6.1, the performance of several existing methods of different categories are evaluated in experiments. They are applied to different 3D scenarios ranging from simple small-scale environments to difficult large-scale environments. The experimental study will support the conclusion that the choice of method depends heavily on the 3D scenario.

Chapter 5

Determining a Real or Virtual Reference Plane

The basic idea of the reference plane approach is to divide the reconstruction task into two steps. First, determine a real or virtual reference plane and second, use the reference plane to reconstruct the structure and cameras linearly and simultaneously. The second step was discussed for different feature types in chapter 3 and we presented our direct reference plane (DRP) approach. Chapter 6 will analyze experimentally this approach. This chapter investigates the first step of determining a real or virtual reference plane. As was seen in sec. 3.2.1, four or more coplanar 3D points define a real reference plane. Their projection into an image is sufficient to determine uniquely the infinite homography of this view. With known infinite homographies, 3D features and cameras may be reconstructed simultaneously from a linear system. This leads to the important observation: The key to simplifying the problem of structure and camera recovery is the *infinite homography*. Naturally the question arises: *What source of information is needed to determine the infinite homography.* Obviously, 3D features lying on a *real* reference plane are sufficient, however, are there alternative ways? For all alternatives, the infinite homography represents a *virtual* reference plane. Virtual reference planes allow us to apply the reference plane reconstruction approach to many practical scenarios where no real reference plane is visible. The source of information needed to determine the infinite homography can be broadly classified into: Information about the scene (sec. 5.1) and information about the cameras (sec. 5.2). The first class consists of *scenes* which have: a real reference plane (sec. 5.1.1), three mutually orthogonal directions (sec. 5.1.2) or an additional orthographic "over"view (sec. 5.1.3). The second class consists of *cameras* which have: constant or known rotation and calibration (sec. 5.2.1), parallel projection (affine camera) (sec. 5.2.2), known epipolar(multi-view) geometry (sec. 5.2.3) or a small baseline (sec. 5.2.4). Table 5.1 summarizes this collection of possible techniques together with the type of reconstruction, metric, affine or projective. Note that we do *not* claim that this collection is complete.

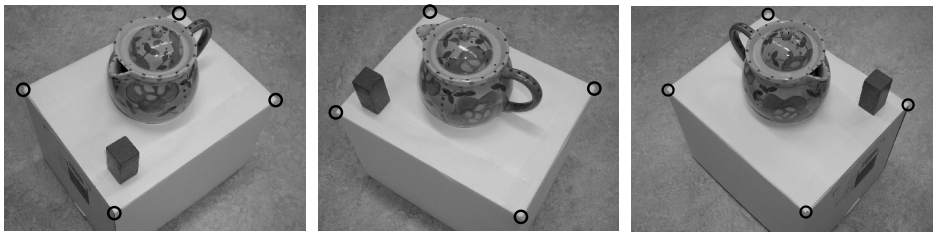
These general techniques to determine the infinite homography are part of earlier publications and (mostly) well known. Our main contribution is twofold. First, we unify these different techniques to determine the infinite homography with the term *reference plane*. Secondly, we point out that both *real* and *virtual* reference plane configurations can be reconstructed with e.g. our direct reference plane approach. This contribution is part of (Rother and Carlsson, 2002b). This publication also introduces a novel algorithm for the simultaneous computation of the infinite homographies from known epipolar geometry (sec. 5.2.3).

For readers who are unfamiliar with the concept of real and virtual reference planes, we would like to stress the following. *All configurations for which the infinite homography of each camera can be derived are reference plane configurations.* The infinite homography may be induced by a *real* reference plane. In all other cases, the infinite homography represents a *virtual* reference plane in the scene. The virtual reference plane can be a finite plane or the correct plane at infinity. Consequently, the two expressions "known infinite homographies" and "known real or virtual reference plane" are equivalent. For succinctness the former is used more frequently in the thesis. Moreover, for 3D reconstruction using a reference plane, it is *not* important if the plane is real or virtual. Therefore, the prefix "real or virtual" is commonly omitted, thus "real or virtual reference plane" becomes the "reference plane". However, it is important to remember that a reference plane does *not only* refer to a real reference plane.

5.1 Information About the Scene

The first source of information to determine the infinite homographies, i.e. a real or virtual reference plane, is the scene or objects in the scene. We will analyze certain scene properties and constraints which allow the calculation of the infinite homographies.

5.1.1 Real Reference Plane



(a) (b) (c)

Figure 5.1. An example of a real scene plane where 4 coplanar 3D points (circles) are visible in all views.

Fig. 5.1 shows a scenario with a real finite reference plane. The four corners of the box define the plane and are visible in all views. As discussed in sec. 3.2.1, these 4 points may be used to determine the infinite homographies H_i^∞ for each view i from eqn. 3.10. If the real reference plane is already stabilized, according to eqn. 2.33, the infinite homographies are $H_i^\infty = I$. This means that we obtain calibrated, translating camera with an identity calibration matrix. This real reference plane scenario has been considered as well in (Shashua and Navab, 1994; Heyden and Åström, 1995a; Triggs, 2000). Obviously, more 3D points give an over-constrained linear system which can be solved using SVD (e.g. Hartley and Zisserman, 2000). Lines may be used instead of points. If a camera centre lies on the reference plane, the corresponding infinite homography is singular (see fig. 2.6). This is, however, an unrealistic scenario for a real reference plane.

Furthermore, it is *not* necessary to have the *same* 3D features visible in all views. A certain number of known inter-image homographies $H_{ij}^\infty = H_j^\infty H_i^{\infty -1}$ (sec. 2.4), between view i and j, is sufficient as well. However, the set of known homographies has to be consistent, i.e. $H_{ij}^\infty = H_{ik}^\infty H_{kj}^\infty$. One way of deriving the individual infinite homographies H_i^∞ from the inter-image homographies H_{ij}^∞ is to assume that $H_1^\infty = I$, which gives $H_i^\infty = H_{1i}^\infty$. However, depending on the image coordinates, some inter-image homographies might be inaccurate. This could introduce a substantial numerical instability in the reconstruction process. How this issue is solved in an optimal way is not discussed in the thesis. However, we will see in the experiments (chapter 6) that for practical applications it is important to obtain accurate homographies. In order to avoid this source of error, we will concentrate in the experiments on the case of 4 or more coplanar scene points visible in all views.

The process of automatically determining the inter-image homographies of a real scene plane can be solved in different ways. The dominant scene plane might be found on the basis of point matches (e.g. Tell, 2002; Hartley et al., 2001). In a first step all potential point matches are determined. The second step finds robustly the dominant homography which has the highest number of inliers, i.e. point matches. Alternatively, direct methods to determine the inter-image homographies have been suggest (Bergen et al., 1992; Irani and Anandan, 1999a; Irani and Anandan, 1999b). Note that both methods might return a homography which does not correspond to a real plane in the scene. This can be seen from the fact that 4 corresponding image points, which are in "general" pose, do *always* specify a homography. However, there is no constraint that the 4 corresponding 3D points are coplanar. A homography which is induced by a real scene plane has to satisfy eqn. 3.105.

5.1.2 Orthogonal Scene Directions

In the previous sec. 5.1.1 we assumed that a real scene plane is visible in all views. Obviously, this is a considerable restriction. The motivation of this section, and of subsequent ones in this chapter, is to derive a *virtual* reference plane by some mean. This makes it possible to apply the reference plane approach for 3D reconstruction to scenarios where no real reference plane is visible. One way is to use points at infinity as reference points. Such points are given by directions in the scene. Figure 5.2 shows such an example of a

(a) (b) (c)

Figure 5.2. The royal castle in Stockholm. It is a building which has three dominant directions (superimposed straight lines).

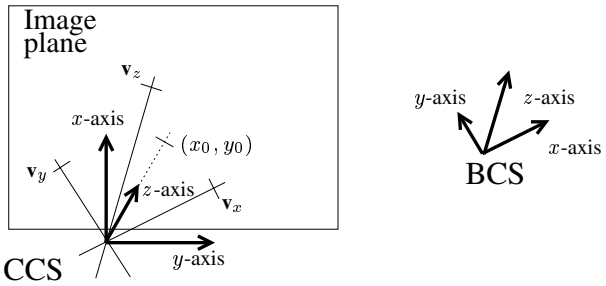


Figure 5.3. The relationship between the camera coordinate system (CCS) and the Cartesian basis coordinate system (BCS). The vanishing points $\mathbf{v}_x\,, \mathbf{v}_y$ and $\mathbf{v}_z$ correspond to mutually orthogonal scene directions.

building which has 3 dominant directions. The projection of a point at infinity is a vanishing point (sec. 2.1). The advantage of using infinite instead of finite reference points is that a scene direction might be observed from very different camera positions. In general, 4 infinite reference points can be used in the same way as in the finite reference point case (sec. 5.1.1). Unfortunately, not many scenes have 4 different dominant directions which can be detected in an image. However, man-made environments are often characterized by 3 mutually orthogonal directions (fig. 5.2). In the following we will show that 3 vanishing points of orthogonal scene directions can be used to compute the camera's rotation R and calibration matrix K and therefore the infinite homography $H^\infty = KR$. In sec. 6.1.3, we use this approach to reconstruct buildings, like the city hall in Stockholm, from a collection of images of very different viewpoints.

It is well known (Caprile and Torre, 1990) that a special camera with zero skew and aspect ratio one,

$$K = \begin{pmatrix} f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{pmatrix} \quad , \tag{5.1}$$

can be calibrated from 3 vanishing points of orthogonal directions. We denote such a camera as a "square pixel" camera. The first step of the calibration process is to determine the 3 vanishing points $\mathbf{v}_x, \mathbf{v}_y$ and $\mathbf{v}_z$ of mutually orthogonal scene directions. Section 8.2 describes an algorithm, based on (Rother, 2000; Rother, 2002), to perform this task automatically. Assume that the 3 vanishing points have been detected successfully. The corresponding scene directions define a Cartesian basis coordinate system (BCS) in the scene. Fig. 5.3 shows the geometrical relation between the BCS and the camera coordinate system (CCS). The orthogonality constraint of the 3 directions can be algebraically expressed as

$$\langle K^{-1}\mathbf{v}_x, K^{-1}\mathbf{v}_y\rangle = 0 \ , \ \ \langle K^{-1}\mathbf{v}_x, K^{-1}\mathbf{v}_z\rangle = 0 \ \text{ and } \ \langle K^{-1}\mathbf{v}_y, K^{-1}\mathbf{v}_z\rangle = 0 \ , \qquad (5.2)$$

where $\langle \cdot, \cdot \rangle$ denotes the scalar product. From these equations the focal length f and the principal point (x_0, y_0) of the specific camera model in eqn. 5.1 can be computed (Caprile and Torre, 1990). The focal length is given as

$$f \ = \ \sqrt{\mathbf{v}_{xx}\mathbf{v}_{yx} + \mathbf{v}_{xy}\mathbf{v}_{yy}} \ . \qquad (5.3)$$

Alternatively, two other pairs, $(\mathbf{v}_x, \mathbf{v}_z)$ or $(\mathbf{v}_y, \mathbf{v}_z)$, can be used to compute f. The principal point is the orthocentre of the triangle formed by $\mathbf{v}_x, \mathbf{v}_y, \mathbf{v}_z$. Liebowitz and Zisserman (1999) use the absolute conic for this calibration process. However, if one or two of the vanishing points are at infinity in the image (so-called degenerate cases) not all internal camera parameters can be derived (Liebowitz and Zisserman, 1999; Rother, 2000). This occurs when the image plane is parallel to the corresponding axis of the BCS. In practice, we avoid these degenerated cases by assuming fixed internal camera parameters for the process of acquiring images. This allows us to improve the camera calibration significantly by averaging all those internal camera parameters which were derived from non–degenerated cases.

With the knowledge of K, the rotation matrix R can be determined. Therefore, the correspondence between the 3 vanishing points and the x-, y- and z-axis of the BCS has to be known. Additionally, the "direction" given by each vanishing point has to be uniquely determined, i.e. the sign of $K^{-1}\mathbf{v}_{x,y,z}$. If this has been solved, we may define

$$R = \left(\pm K^{-1}\mathbf{v}_x \middle| \pm K^{-1}\mathbf{v}_y \middle| \pm K^{-1}\mathbf{v}_z\right) \ \text{ with } \ \det(R) = 1 \ .$$

Since the condition $\det(R) = 1$ has to be fulfilled, there are 24 possible rotation matrices in case of unknown correspondence. If the correspondence between the vanishing points and the axis of the BCS are known, however, their direction is unknown, there are still 4 possible rotation matrices. To determine R, 2 of the 3 vanishing points are sufficient since $K^{-1}\mathbf{v}_z = (K^{-1}\mathbf{v}_x) \times (K^{-1}\mathbf{v}_y)$. How to solve the task of computing R automatically, given the 2 or 3 vanishing points, is explained in sec. 8.3. Furthermore, sec. 8.3 describes the complete process of computing the rotation and calibration of multiple cameras which observe an object with 3 dominant orthogonal directions.

5.1.3 Using an Additional Orthographic "Over"view

The previous method to determine the infinite homographies, i.e. a virtual reference plane, has the disadvantage that all objects in the scene must have the *same* 3 dominant orthogonal directions. Consider a city environment with many buildings. All buildings might have a common vertical direction. However, a map of this environment might reveal that they have a different orientation on the ground plane, i.e. different dominant directions. Recently, Robertson and Cipolla (2002) considered this scenario. They derive the infinite homographies from an additional orthographic "over"view like a 2D map. This is possible if five or more corresponding points between the map and each image is identified. Additionally, the vanishing point of the vertical direction in the scene has to be determined. Moreover, they derive explicitly the camera's rotation and focal length. This has been achieved by assuming a "square pixel" camera with a known principle point. The camera's rotation is uniquely defined with the additional constraint that the picture was taken the "right way up". On the basis of the known infinite homographies, they applied our DRP reconstruction method together with scene constraints on 3D lines with a known direction (see eqn.3.120).

This idea can be further exploited for the reconstruction of any object where an additional orthographic "over"view is available. One could think of technical drawings, i.e. CAD drawings, which are orthographic views. Navab et al. (2000) considered the scenario of lines in one orthographic and two perspective views. They applied their technique for the reconstruction of pipelines in a power plant. In this case two views and a CAD drawing have been accessible.

5.2 Information About the Camera

The second source of information to determine a virtual reference plane, i.e. the infinite homographies, is the camera. Certain properties of the camera or its motion are sufficient to retrieve the infinite homographies.

5.2.1 Constant or Known Rotation and Calibration

Consider the general projection of a point $\mathbf{X}_i$ by a *Euclidean camera* P_j with calibration matrix K_j, rotation R_j and camera centre $\bar{\mathbf{Q}}_j$ (see eqn. 2.25)

$$\mathbf{x}_{ij} \;\sim\; K_j\, R_j\, (\, I \mid -\bar{\mathbf{Q}}_j)\, T\, T^{-1}\, \mathbf{X}_i \;\; . \tag{5.4}$$

As discussed in sec. 2.4, the infinite homography H_j^∞ depends on the camera's calibration K_j and rotation R_j, $H_j^\infty = K_j\, R_j$. The idea of this section is to compute H_j^∞ from some knowledge about K_j and R_j. Obviously, if both K_j and R_j are known, the infinite homography $H_j^\infty = K_j\, R_j$ is known as well. However, what happens if one or both of them are constant? In eqn. 5.4 the 4×4 transformation matrix T is introduced, which represents, as

discussed in sec. 2.3.2, the choice of the projective coordinate system. Consider the case where T is an affine transformation

$$T_A = \begin{pmatrix} A & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix} \text{ with } \mathbf{0} = (0,0,0)^T \ . \tag{5.5}$$

This transformation leaves the plane at infinity unchanged since $T_A^T \pi_\infty \sim \pi_\infty$. Note, in eqn. 5.4 the transformation of a 3D point is $T_A^{-1}\mathbf{X}$. If T_A is the identity matrix, $T_A = I$, the final reconstruction of the points and cameras is metric, since we assumed a Euclidean camera. As explained in sec. 2.2.3, the reconstruction is still metric if T_A is a similarity transformation, i.e. $A = \lambda R$ where R is a rotation matrix. This reflects the freedom to rotate and scale (λ) the complete reconstruction. However, for an unconstrained matrix A the final reconstruction is affine, as discussed in sec. 2.2.2. Eqn. 5.4 may be written for the transformation matrix T_A as

$$\mathbf{x}_{ij} \sim K_j \, R_j \, A \, (\, I \mid \mathbf{t} - \bar{\mathbf{Q}}_j) \, T_A^{-1} \, \mathbf{X}_i \ . \tag{5.6}$$

The vector $\mathbf{t}$ defines the origin of the coordinate system and is not relevant in this context. For calibrated cameras with a known rotation, i.e. K_j and R_j are known, we may choose $A = I$ and H_j is then given as $H_j = K_j \, R_j$. In this case $T_A = I$ and the reconstruction is metric as discussed above. In case of calibrated, translating cameras, i.e. K_j is known and R_j is constant ($R_j = R$), we may choose $A = R^{-1}$ and H_j is given as $H_j = K_j$. The matrix T_A represents a similarity transformation and the final reconstruction is therefore metric. Finally, for translating cameras with fixed internal camera parameters, i.e. K_j is constant ($K_j = K$) and R_j is constant ($R_j = R$), we may choose $A = (K \, R)^{-1}$ and $H_j = I$. The reconstruction is affine since T_A represents an affine transformation. An experimental analysis showed, however not depicted here, that if the camera calibration is approximately known, it is preferable to set H_j as $H_j = K$. Table 5.1 summarizes these 3 cases. For all three approaches the virtual reference plane is the correct plane at infinity since the transformation T in eqn. 5.4 is either affine or metric. Consequently, no finite 3D point can lie on the virtual reference plane. This has the important advantage that our DRP reconstruction method can be applied directly without excluding any finite 3D point (or line) from the linear system.

In practice, the camera matrix K_j can be obtained, i.e. calibrated, in many different ways from metric scene properties. The traditional way is to use a calibration object with known properties, like a calibration grid. Section 5.1.2 explained how to calibrate a "square pixel" camera from 3 vanishing points of mutually orthogonal scene directions. Note that the task of auto- (or self-) calibration is not relevant in this context since the cameras have to be calibrated before performing the reconstruction task.

The fact that purely translating cameras produce affine structure was first noted by Van Gool et al. (1994). It was also discovered by Shashua and Navab (1994) under the name of "relative affine structure". Shashua and Navab (1994) and Beardsley et al. (1994) suggested "sequential" reconstruction methods for this case. However, as we have seen in sec. 3.2.2, this approach is sub-optimal since both cameras and 3D features may be determined linearly and simultaneously. The reconstruction task for calibrated cameras

with a known orientation has been studied in (Debevec et al., 1996; Shum et al., 1998; Robertson and Cipolla, 2000; Antone and Teller, 2002). These publications once again did not exploit the fact that known infinite homographies can be used to reconstruct both cameras and 3D features simultaneously.

5.2.2 Affine Cameras

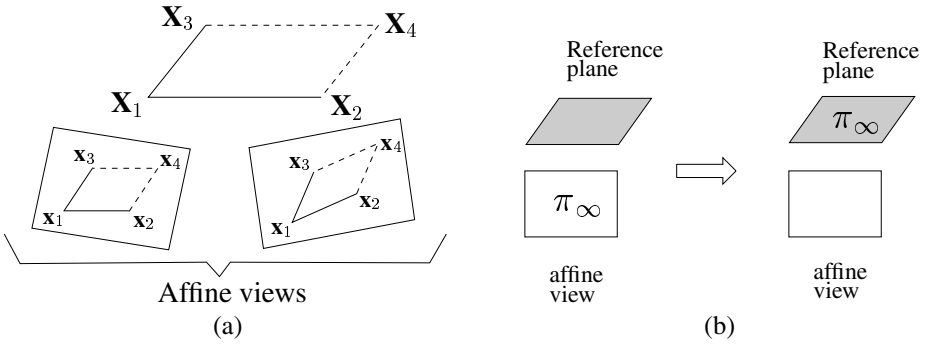


Figure 5.4. (a) Determining the image of a fourth coplanar point $\mathbf{X}_4$ for affine cameras. (b) The concept of moving the plane at infinity π_∞ from its "true" location to the reference plane.

In this section we will show that the simple assumption of having affine cameras can be used to reconstruct general, unconstrained scenes where at least 3 points are visible in all views. This approach to determine a reference plane together with our DRP method has been introduced in (Rother and Carlsson, 2002b), and can be considered as an alternative approach to affine factorization (Tomasi and Kanade, 1992) (sec. 3.2.4). Both approaches determine the cameras and 3D points simultaneously from a measurement matrix. The main drawback of factorization methods is that missing data is not handled naturally. The main disadvantage of our method is that certain reference points are distinguished, i.e. have a greater influence on the performance (see sec. 6.1.2).

The basic idea of this approach is to "hallucinate" a finite reference plane in the scene. The reference plane is defined uniquely by 3 reference points visible in all views. As shown in sec. 5.1.1, 4 coplanar 3D points are sufficient to determine uniquely the infinite homographies. A fourth "virtual", coplanar reference point can be determined with the assumption of parallel projection. Assume that 3 reference points $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3$ in "general" position are visible in all views. A fourth coplanar point $\mathbf{X}_4$ may be determined as $\mathbf{X}_4 = \mathbf{X}_3 + \mathbf{X}_2 - \mathbf{X}_1$ (see fig. 5.4 (a)). Since affine cameras perform a parallel projection on scene points, the affine image of $\mathbf{X}_4$ in view j is $\mathbf{x}_{4j} = \mathbf{x}_{3j} + \mathbf{x}_{2j} - \mathbf{x}_{1j}$. Alternatively, the fourth point could be chosen as the centroid of the 3 reference points.

However, how are affine cameras embedded in our "reference plane concept", where the reference plane represents the plane at infinity? Consider the mapping of a general

projective camera as defined by eqn. 2.27

$$\mathbf{x}_{ij} \sim \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ v_1 & v_2 & v_3 & v_4 \end{pmatrix} \mathbf{X}_i \ . \tag{5.7}$$

As described in sec. 2.3.2, the last row of the camera matrix is the principle plane $\pi_{prin} = (v_1, v_2, v_3, v_4)^T$ of the camera. It contains the camera centre and is parallel to the image plane. In a projective space where the plane at infinity is at its true location, the principle plane of an affine camera is the plane at infinity, $\pi_{prin} = \pi_\infty = (0, 0, 0, 1)^T$ (fig. 5.4(b) left). However, we have seen in sec. 3.2.1 that the reference plane has to be the plane at infinity in order to linearize the relationship between 3D features and cameras (fig. 5.4(b) right). This means that in this particular projective space all camera centres lie on a plane π_{prin} which is different to π_∞. Therefore, affine cameras may be treated as projective cameras. This leads to a unified treatment of parallel and perspective projection in a single framework, which will be explained in more detail later. From the 4 coplanar reference points, the infinite homographies H_j can be derived as discussed in sec. 5.1.1. The reconstructed cameras provide the principle plane π_{prin}, which contains all the camera centres. Finally, by mapping π_{prin} to π_∞ using eqn. 2.17 the projective reconstruction transforms into an affine reconstruction.

How does this approach compare to other affine reconstruction methods? In our approach 6 parameters of each affine camera are determined directly by the infinite homography. The remaining 2 unknown parameters, which represent the direction of an affine camera, are reconstructed simultaneously with the scene points. In contrast, affine factorization (Tomasi and Kanade, 1992) determines 2 parameters ($m_{1-2,4}$) of each affine camera directly. The remaining 6 parameters of each camera are determined simultaneously with the scene points. However, this method does not allow missing data. It has been shown (Koenderink and Doorn, 1991; Heyden and Kahl, 2000) that all 8 unknown parameters ($m_{1-2,1-4}$) of an affine camera can be determined directly by choosing a special affine basis of 4 3D points visible in all views. Each 3D point $\mathbf{X}_i$ gives two linearly independent projection equations, derived from eqn. 5.7, which are sufficient to determine the unknown camera parameters $m_{1-2,1-4}$. However, from an numerical point of view this is less favorable. An improvement of Heyden and Kahl's (2000) method is to use only the 6 parameters $m_{1-2,1-3}$, which define the infinite homography. The remaining camera parameters together with all 3D features can then be reconstructed with our DRP method. Note that the homography is in this case singular since the last row consists of zeros. In sec. 6.1 we extend our DRP method to general homographies, i.e. singular and non-singular.

To apply this method of hallucinating a reference plane, we must answer the question: Which are the best reference points to define the plane? In practice there may be many possible reference points which are visible in all views. Consider the criteria for good reference points. First, a camera centre must not lie on the reference plane. This means that the 3 reference points must not be collinear in any view, since the homography is not unique for 4 collinear image points. Second, in the presence of noise the infinite homography is determined more accurately if the projected reference points are far apart in the image. Since the two criteria are not contradictory, we choose as reference points those 3 points

which are "least collinear". This is done by considering the distance between one reference point to the line defined by the other two reference points.

Finally, we address the question of how to extend this approach for affine cameras to general projective cameras. Assume that a 3D reconstruction has been computed with e.g. our DRP method and the assumption of affine cameras. One idea is to iterate the reconstruction process in order to compensate for the perspective effects. A similar idea has been suggested by (Triggs, 1996; Qian and Medioni, 1999; Heyden et al., 1999; Hartley and Zisserman, 2000) to circumvent the pre-estimation of the projective depths, i.e. the perspective effects, for projective factorization. In our case, this iterative process involves the infinite homographies. This means updating the infinite homographies H_j on the basis of known 3D points $\mathbf{X}_i$ and camera centres $\bar{\mathbf{Q}}_j$. The projection relation (eqn. 2.27) may be written as

$$\mathbf{x}_{ij} \;\sim\; H_j \,(\, I \mid -\bar{\mathbf{Q}}_j)\, \mathbf{X}_i \;\sim\; H_j \,\mathbf{x}'_{ij} \;\;, \tag{5.8}$$

where $\mathbf{x}'_{ij}$ is the projection of point $\mathbf{X}_i$ by camera $(\, I \mid -\bar{\mathbf{Q}}_j)$. Since $\mathbf{x}_{ij}$ and $\mathbf{x}'_{ij}$ are known, the infinite homography H_j can be determined for each image j individually (e.g. Hartley, 1997).

5.2.3 Known Epipolar (Multi-View) Geometry

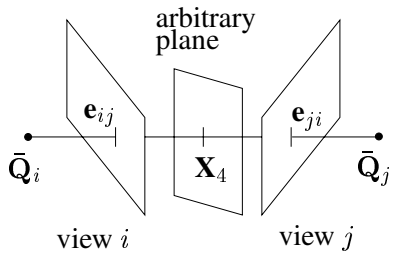


Figure 5.5. A fourth coplanar point $\mathbf{X}_4$ can be determined on any plane, e.g. the reference plane, from the known epipoles $\mathbf{e}_{ij}$ and $\mathbf{e}_{ji}$ of projective cameras.

Hallucinating a finite reference plane can also be applied to general projective cameras with known epipolar geometry. Consequently, our DRP reconstruction method can be applied to general scenes, with at least 3 points visible in all views, and general projective cameras (Rother and Carlsson, 2002b). This gives an alternative reconstruction method to projective factorization (Sturm and Triggs, 1996) (sec. 3.2.4). Both approaches require that the epipolar (or multi-view) geometry is known. However, projective factorization has to pre-compute the projective depths and "hallucinate" missing data. As already mentioned, the main disadvantage of our method is that certain reference points are distinguished, which has an influence on the performance (see sec. 6.1.2).

Assume that the 3 reference points $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3$ are in "general position" and have canonical coordinates in the projective space and in the image as defined in eqn. 3.3. The infinite homography H_j for each view j is then described as

$$H_j = \begin{pmatrix} a_j & 0 & 0 \\ 0 & b_j & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad . \tag{5.9}$$

The assumption that no camera centre lies on the reference plane allows the arbitrary scale of the matrix to be set, $H_j(3,3) = 1$. The variables a_j and b_j are unknown in each view j and can be considered as the mapping of point $(1,1,1,0)^T$ onto view j: $H_j\ (1,1,1)^T = (a_j, b_j, 1)^T$. This derivation of the infinite homography has been suggested earlier by Boufama and Mohr (1995). Assume that the epipolar geometry is known, for instance the fundamental matrices between each pair of views. As introduced in sec. 3.2.3, the epipole $e_{ij} = (e_{ijx}, e_{ijy}, e_{ijw})^T$ is denoted as the projection of camera centre j into view i (see fig. 5.5). Moreover, the inter-image homography from view i to view j via the reference plane is defined as $H_{ij}^{\infty} = H_j^{\infty} H_i^{\infty -1}$. It is well known (e.g. Hartley and Zisserman, 2000) that the epipoles of two views are in correspondence via *any* plane in the scene (see fig. 5.5). Consequently, we may write

$$e_{ji} \ \sim \ H_{ij}^{\infty}\, e_{ij} \quad \Leftrightarrow \quad H_j^{\infty -1}\, e_{ji} \ \sim \ H_i^{\infty -1}\, e_{ij} \quad . \tag{5.10}$$

Using eqn. (5.9) and (5.10) we obtain two constraints between views i and j:

$$\begin{aligned} a_i\, e_{jix}\, e_{ijw} \ - \ a_j\, e_{ijx}\, e_{jiw} \quad &= \quad 0 \\ b_i\, e_{jiy}\, e_{ijw} \ - \ b_j\, e_{ijy}\, e_{jiw} \quad &= \quad 0 \ . \end{aligned} \tag{5.11}$$

All the a_i's and b_i's may now be determined separately but simultaneously. Each pair of images i and j, which are linked by a fundamental matrix, gives one linear equation in a_i, a_j and b_i, b_j respectively. With m images we obtain two sets of linear equations:

$$\begin{aligned} L_a\, h_a \quad &= \quad 0 \quad \text{with} \quad h_a = (a_1, \ldots, a_m)^T \quad \text{and} \\ L_b\, h_b \quad &= \quad 0 \quad \text{with} \quad h_b = (b_1, \ldots, b_m)^T \quad . \end{aligned} \tag{5.12}$$

The last singular vector of the SVD of L_a and L_b gives the solution for h_a and h_b respectively. The vectors h_a and h_b have an arbitrary scale which corresponds to the fact that the fourth unknown reference point on the reference plane has two degrees of freedom. As for affine cameras, the question arises: Which 3 reference points should be used to hallucinate the reference plane? This leads to the same algorithm as discussed for affine cameras (sec. 5.2.2).

This novel algorithm for determining the infinite homographies of multiple views from known epipolar geometry has been introduced in (Rother and Carlsson, 2002b). It has the advantage that all homographies are determined in one step which implies that the complete information given by the geometry is used simultaneously. A similar algorithm has been suggested by Avidan and Shashua (1998). They derive a consistent set of projective cameras from an image sequence by "threading" fundamental matrices. This is

achieved *sequentially* on the basis of fundamental matrices only. As in our method, the infinite homography of each camera matrix is determined by "hallucinating" a reference plane. However, the main advantage of our method is that all infinite homographies are computed simultaneously and not sequentially. Furthermore, we derive only the infinite homographies of the projective cameras. The remaining camera centres are computed simultaneously with the 3D structure. The idea of hallucinating a reference plane by 3 scene points has also been exploited by Boufama and Mohr (1995) to compute the fundamental matrix.

5.2.4 Small Baseline

Two images of a rotating camera are related by an infinite homography (fig. 2.3(c) in sec. 2.2.1). Let us prove this now. Two views $P_1 = H_1^\infty(I|-\bar{\mathbf{Q}})$ and $P_2 = H_2^\infty(I|-\bar{\mathbf{Q}})$ of a rotating camera have a common centre of projection $\bar{\mathbf{Q}}$. The projection of a homogeneous point $\mathbf{X}$ into these cameras can be written as (eqn. 2.27)

$$\mathbf{x}_1 \sim H_1^\infty(\, I \mid -\bar{\mathbf{Q}})\,\mathbf{X} \quad \text{and} \quad \mathbf{x}_2 \sim H_2^\infty(\, I \mid -\bar{\mathbf{Q}})\,\mathbf{X} \quad . \tag{5.13}$$

Substituting $(\, I \mid -\bar{\mathbf{Q}})\mathbf{X}$ in the second of these equations gives the relation

$$\mathbf{x}_2 \sim H_2^\infty\, H_1^{\infty -1}\mathbf{x}_1 \quad . \tag{5.14}$$

Therefore $H_{12} = H_2^\infty\, H_1^{\infty -1}$ is a homography between the two views via the correct plane at infinity. The homography H_{12} may be determined from at least 4 corresponding image points like $\mathbf{x}_1, \mathbf{x}_2$. Finally, the infinite homography H_2^∞ may be derived from H_{12} with the assumption that $H_1^\infty = I$.

Oliensis (1995) exploited this idea for 3D reconstruction from continuous image sequences (see as well Oliensis, 1995; Oliensis, 1999; Oliensis and Genc, 1999). The basic assumption in this work is a small movement of the camera between successive frames, a *small baseline*. This is an approximation of a rotating camera and means that the infinite homographies may be determined approximately.

5.3 Summary

This chapter investigated alternative techniques to determine the infinite homographies, i.e. a real or virtual reference plane. This increases the applicability of the reference plane reconstruction approach to many different scenarios where no real reference plane is visible. The idea of the reference plane reconstruction approach is to determine first a real or virtual reference plane and then the cameras and structure with e.g. our direct reference plane (DRP) approach. Table 5.1 summarizes possible techniques of deriving the infinite homographies given information about either the *scene* or the *cameras*. Note that we do *not* claim that this collection is complete. Most of these techniques are well known and part of earlier publications. The main contribution of this chapter is twofold. First, we unify these different techniques to determine the infinite homography with the term *reference*

Real or Virtual Reference Plane Configurations	Reconstruction
Real reference plane	projective
3 mutually orthogonal scene directions	metric
Additional orthographic "over"view	metric
Calibrated cameras with known or constant rotation	metric
Translating camera with constant calibration	affine
Affine cameras	affine
Known epipolar(multi-view) geometry	projective
Small baseline	projective

Table 5.1. Some possible techniques to determine a real or virtual reference plane.

plane. Secondly, we point out that both *real* and *virtual* reference plane configurations can be reconstructed with our DRP method. A further contribution is a method to compute simultaneously the infinite homographies from known epipolar geometry (sec. 5.2.3).

For a reference plane reconstruction method, it is *not* important whether the plane is real or virtual.[1] However, the position of the real or virtual reference plane is *important*. Table 5.1 lists in addition to the reference plane configuration the type of reconstruction. There is a considerable difference between those cases which give a metric or an affine reconstruction and those that give a projective reconstruction. In the latter case, the plane at infinity, i.e. the reference plane, is *not* at its correct position.[2] This has important consequences for our DRP reconstruction method, i.e. the second step of the reference plane reconstruction approach. If the plane at infinity is at its correct position, no finite 3D point can lie on the reference plane. Therefore, our reconstruction method can be applied directly without excluding any finite 3D point from the linear system. Furthermore, since no finite 3D point lies close to the reference plane, all reconstructed 3D points have the same order of magnitude. As will be demonstrated in the experimental chapter 6, this simplifies our method and improves the performance.

A further aspect about the different techniques of deriving a reference plane is that they may be combined. For example, 3 points are used to hallucinate a reference plane as described in sec. 5.2.2 and 5.2.3. The position of some of the cameras might be further away from the scene than others. Consequently, the infinite homography of those views which are far away may be determined directly by assuming parallel projection (sec. 5.2.2). The infinite homographies of the other views, which are closer to the scene, may be derived from known epipolar geometry (sec. 5.2.2).

Finally, there may be further techniques to determine a virtual reference plane. For instance, is it possible to exploit symmetry properties or the contours of an object? More generally, might it be enough to know that an object belongs to a certain class with some "geometric" properties? These are interesting open questions for future research.

[1] This is the reason why in most parts of the thesis we simple use the term "reference plane" instead of "real or virtual reference plane".

[2] Note that affine cameras are an exception, where the plane at infinity is a real scene plane (sec. 5.2.2).

Chapter 6

Structure and Camera Recovery using a Reference Plane

The previous chapters introduced many theoretical concepts for the task of 3D reconstruction. However, are these concepts really applicable to real world scenarios? This chapter will demonstrate that our direct reference plane reconstruction method can be used to reconstruct the city hall in Stockholm from 37 real images, as motivated in chapter 1.

The idea of the reference plane approach is to divide the reconstruction task into two steps. First, determine a real or virtual reference plane and second, use the reference plane to reconstruct the structure and cameras simultaneously and linearly. The first step was discussed in detail in the previous chapter. The theoretical background of the second step, our direct reference plane (DRP) approach, was presented in chapter 3. This chapter has two main goals. First, practical algorithms of our DRP methods are formulated for points (sec. 6.1), lines (sec. 6.2) and planes (sec. 6.3). Secondly, the performance of our methods are analyzed under real world conditions and compared with a variety of other methods.

In chapter 3, several practical issues were neglected, to simplify the presentation of our methods. This included the task of separating features on and off a finite reference plane. Furthermore, it was assumed that the cameras are finite, i.e. the cameras' centre must not lie on the reference plane, and that each image has a specific projective basis. These issues are addressed here which extends our methods to use normalized image data and general cameras. The novel presentation for point features is based on (Rother and Carlsson, 2002b; Rother and Carlsson, 2002a). Readers who are familiar with this material should note that the aspects of normalized image points and general cameras have not been published in our journal and conference papers. For lines, we introduce novel methods which are not part of any of our previous publication. For planes, three methods are presented, (a) our novel direct reference plane method, (b) our linear, camera-constraint method (Rother et al., 2002), and (c) a factorization method (Triggs, 2000; Rother et al., 2002).

The main focus of the experimental study is on point features (sec. 6.1.2 and 6.1.3). We are particularly interested in reconstructing large scale environments like the city hall in Stockholm. Such scenarios are difficult since the amount of missing data is high, up to 90%. *The conclusion will be that for such difficult scenarios our method outperform all non-reference plane and reference plane reconstruction methods.* A further conclusion will be that reference plane methods are inferior to general methods if the reference plane is detected very inaccurately. The synthetic experiments compare various versions of our method with two other reference plane methods (Triggs, 2000; Hartley et al., 2001). We will show that our method performs very stably if the 3D points are *not close* to the reference plane, such as the reference plane is the correct plane at infinity. If 3D points are on or close to the reference plane, it is on the basis of this examination not possible to select the best reference plane method. Most of the experiments on real and synthetic data for point features were presented in (Rother and Carlsson, 2001; Rother and Carlsson, 2002b; Rother and Carlsson, 2002a). However, these publications do *not* contain the extensive comparative study with other point based reconstruction methods (sec. 6.1.3). The detailed study in (Rother and Carlsson, 2002b) about "hallucinating" a reference plane by assuming either known epipolar geometry or affine cameras is not presented in full length here. This has the reason that these general reconstruction methods do not perform very stably with respect to noisy image data, due to the distinguishing of reference points. The experiments on lines (sec. 6.2.2 and 6.2.3) have not been published previously. For planes, a simple comparative study between methods using the planar-homographies directly and methods which hallucinate image features is conducted (sec. 6.3.2 and 6.3.3). This discussion goes beyond our publication (Rother et al., 2002) and is similar to (Szeliski and Torr, 1998).

6.1 Points

We begin this section by outlining a practical algorithm of our direct reference plane method (sec. 6.1.1). Furthermore, several issues of optimizing this method are discussed. After that, experiments are presented based on synthetic and real data (sections 6.1.2 and 6.1.3). For synthetic data, various aspects of our DRP method were analyzed and compared with two other reference plane methods (Triggs, 2000; Hartley et al., 2001). The main focus of this chapter is, however, on real world experiments. In this section, difficult real world scenes, with a high percentage of missing data, are reconstructed with our DRP method and several other point-based reconstruction methods. As already mentioned, for some difficult scenarios our DRP method is significantly superior to all reference plane and non-reference plane reconstruction methods.

6.1.1 Outline of the DRP Method & Optimization

Section 3.2.2 presented the direct reference plane (DRP) approach for the simultaneous reconstruction of points and cameras from a single linear system. This systems forms the core of the DRP method. However, we made a number of simplifications in order to obtain

this linear system. First, the images were stabilized, i.e. a special image basis was chosen. Consequently, the linear reconstruction method is not applicable for general cameras. Furthermore, it is not possible to choose a different image basis, such as normalized image points. Secondly, in case of a finite reference plane, points on and off the reference plane must be separated. Note that the linear system does not provide the correct solution if one of the 3D points lies on the reference plane. Furthermore, it can be expected that the linear system becomes unstable if a 3D point is close to the reference plane. This can be prevented by reducing the influence of such a point on the solution of the linear system. All these practical issues are addressed in the following. After this discussion we will outline our DRP method and variations of it.

Most of these ideas were published in (Rother and Carlsson, 2002b; Rother and Carlsson, 2002a). However, the reader which is familiar with these publications should notice that the normalization of point features was presented previously in a sub-optimal way. This is corrected here.

Different types of projection equations

In the following we will derive four different projection relations for 3D points and cameras in the reference plane case. The differences are related to general (or finite) cameras, normalized (or non-normalized) images and with (or without) the explicit treatment of the projective depths.

The projection relation of a 3D points $\bar{\mathbf{X}}_i$ and a finite camera $P_j = H_j^\infty (I| - \bar{\mathbf{Q}}_j)$ was discussed in sec. 3.2.1. A finite camera is characterized by a non-singular homography H^∞ and a centre of projection $\bar{\mathbf{Q}}$ which lies not on the reference plane. This relation may be written in terms of the stabilized image points $\mathbf{x}'_{ij}$ as (see eqn. 3.16)

$$\mathbf{x}'_{ij} \;\sim\; H_j^{\infty\,-1}\,\mathbf{x} \;\sim\; \bar{\mathbf{X}}_i - \bar{\mathbf{Q}}_j \;\; . \tag{6.1}$$

The unknown scale factor in eqn. 6.1 may be eliminate by taking ratios, which gives the 3 projection equations in eqn. 3.13. These equations are linear in the unknown parameters $\bar{\mathbf{Q}}_j$ and $\bar{\mathbf{X}}_i$.

In general, a camera can be written as $P_j = (H_j^\infty|\mathbf{t}_j)$ (see sec. 2.3.2). This is true for finite cameras, where H_j^∞ is non-singular, and infinite cameras, where H_j^∞ is singular. For general cameras the projection relation (eqn. 6.1) may be written as

$$\mathbf{x}_{ij} \;\sim\; H_j^\infty\,\bar{\mathbf{X}}_i \;+\; \mathbf{t}_j \;\; . \tag{6.2}$$

As before, the unknown scale factor can be eliminated, which gives the 3 equations

$$\begin{aligned}
x\,(h_{21}\bar{X} + h_{22}\bar{Y} + h_{23}\bar{Z} + \mathbf{t}_y) \;-\; y\,(h_{11}\bar{X} + h_{12}\bar{Y} + h_{13}\bar{Z} + \mathbf{t}_x) \;&=\; 0 \\
x\,(h_{31}\bar{X} + h_{32}\bar{Y} + h_{33}\bar{Z} + \mathbf{t}_z) \;-\; z\,(h_{11}\bar{X} + h_{12}\bar{Y} + h_{13}\bar{Z} + \mathbf{t}_x) \;&=\; 0 \\
y\,(h_{31}\bar{X} + h_{32}\bar{Y} + h_{33}\bar{Z} + \mathbf{t}_z) \;-\; z\,(h_{21}\bar{X} + h_{22}\bar{Y} + h_{23}\bar{Z} + \mathbf{t}_y) \;&=\; 0 \;\; ,
\end{aligned} \tag{6.3}$$

where the indices i and j are dropped for simplicity. These equations are, as for finite cameras, linear in the unknown point parameters $\bar{\mathbf{X}}_i$ and camera parameters $\mathbf{t}_j$.

Let us consider the issue of normalizing the image points. It was shown in Hartley (1997) that the normalization of image coordinates can dramatically influence the result of a computation, e.g. F-matrix, based on image coordinates. Normalization means that the centroid of all image coordinates is the origin and the average distance of an image point to the origin is equal to $\sqrt{2}$. This can be achieved by changing the basis in each image by a matrix B_j, $\mathbf{x}'_{ij} = B_j \mathbf{x}_{ij}$. The projection relation 6.2 of normalized image coordinates can be written as

$$\mathbf{x}'_{ij} \;=\; B_j\, \mathbf{x}_{ij} \;\sim\; B_j\, H_j^{\infty}\, \bar{\mathbf{X}}_i \;+\; B_j\, \mathbf{t}_j \;\; . \tag{6.4}$$

Note that $\mathbf{x}'_{ij}$ is now a normalized and not stabilized image point. After elimination of the unknown scale, these projection equations are, as above, linear in the unknown parameters $\bar{\mathbf{X}}_i$ and $\mathbf{t}_j$. Note that in case of stabilized image points (eqn. 6.1), the normalization is canceled out. The matrix B_j was computed differently in (Rother and Carlsson, 2002b; Rother and Carlsson, 2002a). It was suggested to choose the *same* matrix B for all images, as the average of all B_j's. However, this choice is obviously suboptimal.

Finally, consider the case of including the projective depths into the projection relation. Let us rewrite the projection relation in eqn. 6.1 as

$$\lambda_{ij}\, \mathbf{x}'_{ij} \;=\; \bar{\mathbf{X}}_i - \bar{\mathbf{Q}}_j \;\; , \tag{6.5}$$

where λ_{ij} is an unknown scale factor called projective depth. Obviously, the three equations in eqn. 6.5 are linear in the unknown parameters λ_{ij}, $\bar{\mathbf{Q}}_j$ and $\bar{\mathbf{X}}_i$. We denote these projection equations *extended equations*, since they include additionally the unknown parameter λ_{ij}. Eqn. 6.4 may be written in extended form as

$$\lambda_{ij}\, \mathbf{x}'_{ij} \;=\; B_j\, H_j^{\infty}\, \bar{\mathbf{X}}_i \;+\; B_j\, \mathbf{t}_j \;\; . \tag{6.6}$$

To summarize, we derived 4 different types of linear equations: 6.1 (stabilized, not normalized), 6.4 (normalized), 6.5 (extended, stabilized), 6.6 (extended, normalized). All relations may be used to formulate a single linear system to reconstruct multiple scene points and cameras simultaneously.

Distance between scene points and a finite reference plane

We discussed in sec. 3.2.2 that points on and off a finite reference plane must be separated in order to obtain a correct solution for *all* points and *all* cameras from a linear system. In practice, points which are "close to" the reference plane potentially decrease the numerical stability of the reconstruction as well. This separation process is in practice a non-trivial issue. Assume that all scene points are sorted according to their distance to the reference plane. One strategy is to choose a fix threshold to separate points on and off the plane. A different, however, more time consuming method is to exclude successively points from the linear system based on the ranking. The second method is completely automatic, i.e. independent of a threshold parameter.

The remaining question is, how to estimate the distance $dis(\bar{\mathbf{X}})$ of a point $\bar{\mathbf{X}}$ to the reference plane. Consider a configuration with two cameras $\bar{\mathbf{Q}}_1, \bar{\mathbf{Q}}_2$, a 3D point $\bar{\mathbf{X}}$ and a

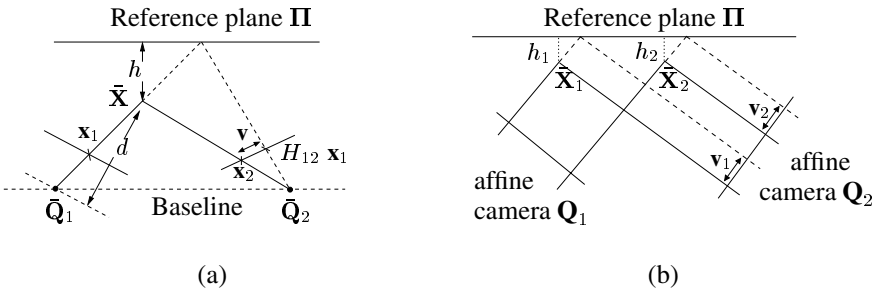


Figure 6.1. Parallax geometry for (a) projective and (b) affine cameras.

reference plane $\mathbf{\Pi}$, where $\bar{\mathbf{X}}$ does not lie on $\mathbf{\Pi}$ (fig. 6.1(a) depicts a top view). The inter-image homography from the first to the second view via the reference plane is defined as H_{12}. The residual parallax vector in the second view is given as $\mathbf{v} = \mathbf{x}_2 - H_{12}\ \mathbf{x}_1$. Obviously, $\mathbf{v}$ is null if $\bar{\mathbf{X}}$ lies on $\mathbf{\Pi}$. However, $\mathbf{v}$ vanishes as well if $\bar{\mathbf{X}}$ lies on the baseline of the two views. Therefore, the distance of a point to the reference plane cannot be determined directly from its parallax vector. Let us define $\gamma_i = \frac{h_i}{d_i}$, where h_i is the perpendicular distance of $\bar{\mathbf{X}}_i$ to the reference plane, and d_i is the depth of $\bar{\mathbf{X}}_i$ with respect to the first view (see fig. 6.1(a)). It is known (Irani and Anandan, 1996) that the relative depth $\frac{\gamma_1}{\gamma_2}$ of two points $\bar{\mathbf{X}}_1$ and $\bar{\mathbf{X}}_2$ can be derived directly from their parallax vectors $\mathbf{v}_1, \mathbf{v}_2$. This means that the relative distance $\frac{h_1}{h_2}$ of two points depends on both their parallax vectors and their depths. However, if we assume parallel projection, d_i is constant and we obtain the relative distance of two points as

$$\frac{\gamma_1}{\gamma_2} = \frac{h_1}{h_2} = \frac{||\mathbf{v}_1||_2}{||\mathbf{v}_2||_2} \ . \tag{6.7}$$

Figure 6.1(b) depicts a configuration with affine cameras where $h_1 = h_2$ and therefore $\mathbf{v}_1 = \mathbf{v}_2$. We will use eqn. 6.7 as an approximation for projective cameras.

The original task was to determine a unique function $dis(\bar{\mathbf{X}})$ which represents the distance between a point $\bar{\mathbf{X}}$ and the reference plane. Eqn. 6.7 supplies a distance function $dis_{j_1 j_2}(\cdot)$ between each pair of views j_1, j_2, which is unique up to scale. A unique function $dis(\cdot)$ can be obtained by recursively merging the set of functions $dis_{j_1 j_2}(\cdot)$. Finally, $dis(\cdot)$ is scaled so that the maximal distance of a point to the reference plane is equal to one, i.e. $dis(\cdot) \in [0, 1]$.

Weighting the projection equations

Let us consider a point $\bar{\mathbf{X}}_1$ which is closer to a finite reference plane than another point $\bar{\mathbf{X}}_2$. Since the reference plane is the plane at infinity in the chosen projective space, the coordinates of the reconstructed point $\bar{\mathbf{X}}_1$ are larger than the ones of $\bar{\mathbf{X}}_2$. This means that in the presence of noise, the point with larger coordinates is reconstructed more accurately.

In order to eliminate this favoring of certain points, an image point $\mathbf{x}_{ij}$ may be weighted with a factor s_{ij}, i.e. $\mathbf{x}_{ij} = s_{ij}\mathbf{x}_{ij}$. Consequently, the projection equations involving $\mathbf{x}_{ij}$ are also weighted. As weighting factors we suggest to choose[1]

$$s_{ij} = dis(\bar{\mathbf{X}}_i) \ , \tag{6.8}$$

where $dis(\cdot) \in [0, 1]$. This means that image coordinates of scene points which are closer to the reference plane are inhibited. The same applies to the projection equations of such a point, which form the linear system.

Outline of the algorithm

On the basis of the previous considerations, the practical algorithms for finite and infinite reference planes can be formulated. In case of a **finite reference plane**, the DRP algorithm is composed of the following steps.

1. Determine H_j of a finite reference plane.
2. Compute the distance $dis(\bar{\mathbf{X}}_i)$ between points $\bar{\mathbf{X}}_i$ and the reference plane.
3. Exclude iteratively points from the S-matrix (or choose a fix threshold).
    4. Determine scales $s_{ij} = dis(\bar{\mathbf{X}}_i)$ and image points $s_{ij}\left\|\mathbf{x}'_{ij}\right\|_2$ (eqns. 6.1, 6.4).
    5. Obtain $\bar{\mathbf{X}}_i, \bar{\mathbf{Q}}_j(\mathbf{t}_j)$ (and λ_{ij}) by SVD using projection relations 6.1, 6.4, 6.5 or 6.6.
    6. Compute points $\bar{\mathbf{X}}_i$ on (or close to) the reference plane with eqn. 3.14.
7. Take the best result (RMS-error between image points and reprojected scene points).

The Euclidean norm is denoted $\left\| \cdot \right\|_2$. The quality of the reconstruction is evaluated in terms of the Root-Means-Square (RMS) error between image points and reprojected scene points. However, other criteria could be used. Note that it is sufficient to compute the matrix V, i.e. its last four singular vectors, from the singular value decomposition UDV^T of the system matrix S. This is more efficient since, according to Golub and Van Loan (1996), a full SVD of a matrix of size $m \times n$ requires $4m^2n + 8mn^2 + 9n^3$ flops. However, to compute solely V and D needs only $4mn^2 + 8n^3$ flops.

In sec. 5.2.2, a further extension of this algorithm was suggested. The infinite homographies H_j may be computed from 3 reference points and the assumption of affine cameras. Perspective effects may be compensated iteratively. This means that the steps $1 - 7$ are executed for each iteration of a new H_j (see eqn. 5.8).

To summarize, the different versions of our DRP algorithm depend on (a) 4 different projection equations, (b) weighting the projection equations and (c) separating points on and off the reference plane either by iteration or a fixed threshold. The versions which compute additionally the unknown projective depths, i.e. use the extended eqns. 6.5 or 6.6, are obviously more time consuming than the other versions, since the number of unknowns and consequently the linear system is larger. Based on the results of the experimental section, we suggest to choose for general scenes with a known finite reference plane the

[1] This particular choice of the weighting factors s_{ij} is motivated by the mapping $(0, 1)^T \rightarrow (1, 0)^T$ and $(1, 1)^T \rightarrow (1, 1)^T$ in the projective space P^1.

iterative algorithm which uses the weighted projection eqn. 6.1. The drawback of this version is that the iteration is time consuming and depends on the number of scenes points. For scenes where 3D points are *not* close to the reference plane the simple, non-iterative DRP algorithm is recommended.

In case of an **infinite reference plane** our DRP algorithm is significantly more simple, since finite scene points cannot lie on the reference plane. The outline of the algorithm is as follows.

1. Determine H_j of an infinite reference plane.
2. Compute the image points $\left\|\mathbf{x}'_{ij}\right\|_2$ (eqns. 6.1, 6.4).
3. Obtain $\bar{\mathbf{X}}_i$, $\bar{\mathbf{Q}}_j(\mathbf{t}_j)$ (and λ_{ij}) by SVD using projection relations 6.1, 6.4, 6.5 or 6.6.

As in the finite reference plane case, we suggest to choose the projection eqn. 6.1.

6.1.2 Experiments: Synthetic Data

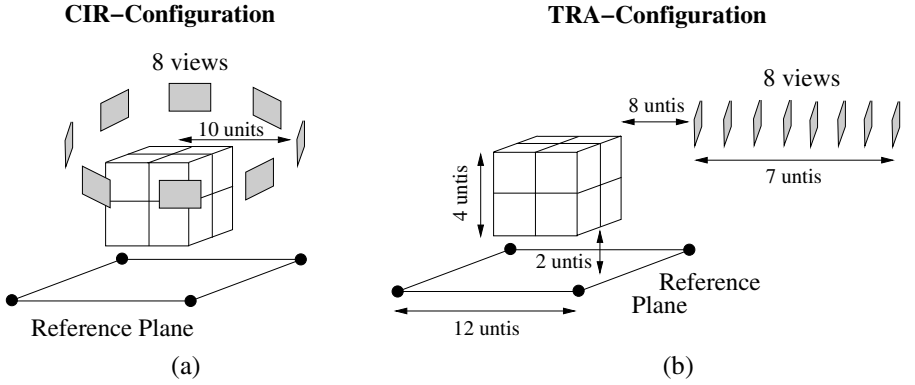


Figure 6.2. Two synthetic configurations with circular motion of the camera (a) and translational movement of the camera towards the scene (b).

The goal of this section is to analyze in detail our DRP method under various conditions and compare it, if appropriate, with the camera constraint method of Hartley et al. (2001) and the factorization method of Triggs (2000). The three main aspects are the *position* and *quality* of the reference plane and the case of *missing data*. The main result will be that for scenes where 3D points are *not close to* the reference plane, our simple, non-iterative DRP version performs very stably. The performance is virtually optimal when the reference plane is the correct plane at infinity. Furthermore, our algorithm performs very good even for a substantial amount of missing data, i.e. up to 63%. For "flat scenes" with many 3D points *on* or *close to* the reference plane, our iterative DRP method performed stably.

However, for such scenes the factorization method and the camera constraint method are more efficient since non-iterative. For flat scenes, the factorization method performed best. This method has, however, the drawback that it does not handle the case of missing data naturally. Moreover, the factorization method is *not* applicable for infinite reference plane. The camera constraint method performs for flat scenes not as stably as our or Triggs's (2000) method. The investigation about the quality of the reference plane will give the expected conclusion that noise on the reference points affect considerably the performance of our DRP algorithm.

In the previous chapter 5, many different techniques of deriving a real or virtual reference plane, i.e. the infinite homographies, were discussed, like assuming cameras with parallel projection. For 3D reconstruction using a reference plane it is *not important* whether the plane is real or virtual. The only important aspect is the *position* of the reference plane. A *real* reference plane is always a finite plane. A *virtual* reference plane can be either a finite plane (e.g. affine cameras) or the plane at infinite (e.g. 3 orthogonal scene directions). For practical applications both cases of a finite and infinite reference plane are important. This experimental study is mainly focused on *finite* reference planes. The reason is that in this case our DRP algorithm is more complex since points on and off the reference plane must be reconstructed separately (sec. 6.1.1). However, most of the conclusions drawn from the experiments apply also to infinite reference planes.

To investigate the performance of our DRP algorithm, it was applied to two different synthetic configurations (see fig. 6.2). The synthetic scene consists of a cube with 26 points floating above a real reference plane. The reference plane is a square where the four corners depict the reference points. In some synthetic experiments, the cube is replaced by 50 points distributed randomly in a sphere of radius 2. In the first configuration (fig. 6.2 (a)) a camera circled around the cube, with a radius of 10 units, and shot 8 images (Cir-configuration). In the second configuration (fig. 6.2 (b)) a camera moved translationally towards the scene (Tra-configuration). The dimensions of the configurations are as in fig. 6.2. The internal calibration matrix of the camera was set to diag(1000,1000,1). Affine cameras were derived from the projective cameras by moving the centre of projection to infinity, where the image size remained fixed (Hartley and Zisserman, 2000).

Most of the synthetic experiments were carried out with respect to different levels of Gaussian noise: $\sigma = 0, 0.2, \ldots, 3.0$ (standard deviation). In order to obtain average performance, the following two steps were conducted 10 times for each noise level: (a) determine the scene points (randomly for the point cloud) and the cameras, (b) add Gaussian noise on the reprojected 3D points. The computed reconstruction was evaluated in terms of the Root-Mean-Square (RMS) error between reprojected 3D points and 2D image data (potentially corrupted by noise). If not mentioned differently, no missing data is assumed. This has the advantage that factorization algorithms can be applied and compared more easily. Furthermore, in all experiments the theoretical minimum, i.e. Cramer-Rao lower bound, is shown, which depends solely on the noise level, the number of unknown parameters and the number of measurements.

In the following our **DRP** method is compared with 9 other reconstruction methods:

Fmat The Fmat algorithm merges subsets of views in an optimal hierarchical fashion as suggested by Fitzgibbon and Zisserman (1998). Since the real and synthetic dataset consists of images with a considerable wide baseline, the algorithm is initialized with all possible subsets of 2 views (F-matrices) (sec. 4.2.1).

FmatBa A variation of the Fmat method. After *each* merging process of the Fmat method an additional bundle adjustment step is applied.

ResInt The ResInt method of Beardsley et al. (1996) is based on the "intersection-resection" scheme. We choose to optimize neither the structure nor the cameras by a non-linear optimization process (sec. 4.2.1).

ResIntBa A variation of the ResInt method with an additional bundle adjustment process after *each* resection-step.

ProjFac The projective factorization method (ProjFac) of Sturm and Triggs (1996) is applied in the case of no missing data (sec. 3.2.4). The projective depths are determined sequentially from the F-matrices. For missing data, the extended version of projective factorization by Martinec and Pajdla (2002) is used.

AffFac The affine factorization method (AffFac) of Tomasi and Kanade (1992) is extended for the case of missing data by the "fitting" algorithm of Jacobs (1997) (sec. 3.2.4).

AffClos For affine views, the AffClos method uses the image closure constraints, based on multi-view affine tensors, to determine all cameras simultaneously (sec. 3.2.3). As suggested in (Kahl and Heyden, 1999) only the centered trifocal tensors are applied.

RefCam The RefCam method represents Hartley et al.'s (2001) reference plane method which reconstructs all cameras simultaneously based on camera constraints (sec. 3.2.3). In order not to "miss" corresponding image points, the camera constraints derived from the 2, 3 and 4 views are used simultaneously. Furthermore, the run-time improvement, discussed in sec. 3.2.3, was applied. The 3D points were computed linearly from all available views.

RefFac In the reference plane case, Triggs (2000) suggested a factorization method (RefFac) (sec. 3.2.4). As in the general case, the projective depths are determined sequentially from the F-matrices. We did not extend this method for the case of missing data, since the choice of projective depths is in this case a non-trivial issue.

A further interesting method, which we did not implement, is the joint image closure constraints method of Triggs (1997b). The problem of this method is that in case of an unorganized set of images with only a few correspondences, it is a non-trivial task to scale all multi-view tensors correctly.

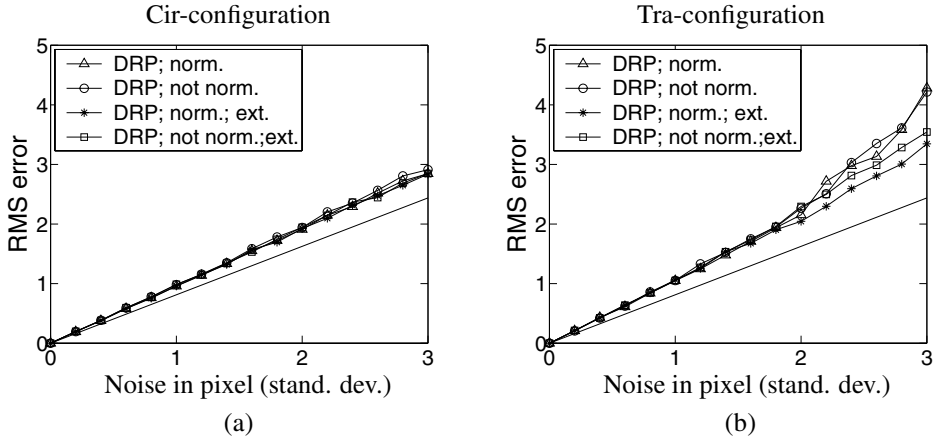


Figure 6.3. Graphs in respect to 4 different types of projection equations, eqn. 6.1 (not norm.), eqn. 6.4 (norm.), eqn. 6.5 (not norm.; ext.) and eqn. 6.6 (norm.; ext.).

Normalization & Extended linear system

In a first experiment, different versions of the DRP algorithm were analyzed, depending on the 4 different types of projection equations introduced in sec. 6.1.1. In this case no points on the reference plane were excluded, apart from the reference points, and no weighting was applied. Fig. 6.3 shows the performance for the Cir-configuration (a) and the Tra-configuration (b). As in all other experiments, the straight line indicates the Cramer-Rao lower bound. A first observation is that the performance of the DRP algorithm (all versions) is better for the Cir-configuration than for the Tra-configuration. This result can be expected since the Tra-configuration has a shorter baseline relative to the scene.

The performance of the 4 different types of projection equations is fairly similar for both configurations. This conclusion has been supported by further experiments, not reported here, for different scenarios and both finite and infinite reference planes. It is worth mentioning that the normalization of the image coordinates did not improve considerably the results. Note that it is, however, important that all homogeneous image points are normalized to 1. In general, all 4 projection equations minimize a geometrically not meaningful algebraic error. Consequently, the simplest version using the projection equations in 6.1, which stabilizes the images, are used for the rest of the experiments, if not stated differently.

Thresholding & Weighting

In this experiment, the practical important scenario is investigated of many 3D points on or close to a finite reference plane. Fig. 6.4 shows an experiment were the distance between the cube and the reference plane varied between 0 and 2 units. In this case the

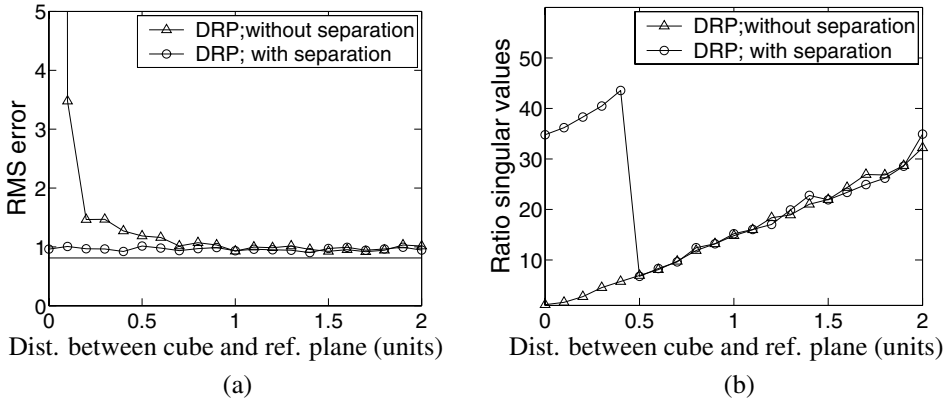


Figure 6.4. The performance of two versions of the DRP algorithm: without separating points on and off the reference plane and separating points by iteration. The performance is analyzed in terms of the RMS-error (a) and the ratio between the fifth and fourth last singular value (b).

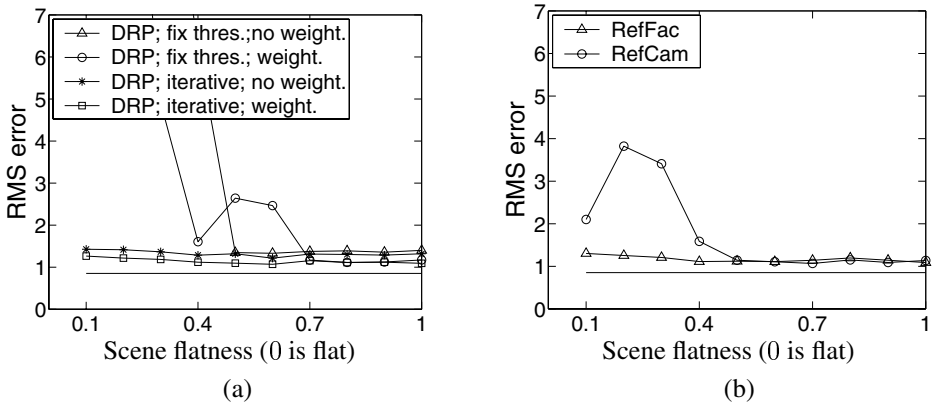


Figure 6.5. Performance of the DRP algorithm (a) and RefCam and RefFac algorithm (b) in respect to the flatness of the scene.

Cir-configuration 6.2(a) was used and the noise level was set to 1. If the distance is 0, 9 of the 26 points of the cube lie on the reference plane. Two different versions of the DRP algorithm were investigated: always *all* points are used for the S-matrix (*without separation*) and points are iteratively excluded form the S-matrix (*with separation*). Fig. 6.4 (a) shows that the performance of both versions is very similar above a certain distance, i.e. about 0.5 units. However, if the cube is closer to the reference plane, the performance of the version "without separation" is worse and eventually fails. The version "with separation" has a constant performance for all distances. The ratio between the fifth and fourth last singular value is depicted in fig. 6.4 (b). The curves are as to expected. The solution

is less stable if the cube moves closer to the reference plane. If the cube is closer than 0.5 units to the reference plane, the version "with separation" was considerable more stable than the one "without separation". This is due to the fact that in this case 9 of the 26 points of the cube were reconstructed separately. We may draw the conclusion that the problem of separating points on (or close to) and off the reference plane can be handled by the iterative version of the DRP algorithm. However, a version which does not take care of this problem eventually fails if points are on or close to the reference plane.

The next experiment analyses the performance of the 3 different reference plane algorithm, DRP, RefCam and RefFac, for "flat scenes". The cube was replaced by a sphere of radius 2 with 50 randomly distributed points. The centre of the sphere is on the reference plane. As above, the Cir-configuration with a noise level of 1 was chosen. The height of each point is multiplied with a "flatness-factor" between 0 and 1. For a factor of 0, all points are on the reference plane. Such a scenario cannot be reconstructed (see chapter 7). Fig. 6.5(a) shows the performance of the different versions of the DRP algorithm. The versions with a fixed threshold of 0.1 performed good for not flat scenes (above 0.7). However, these algorithms fail for very flat scenes. A possible explanation is that in this case a "rough" classification of scene points by a fixed threshold is not suitable. However, the iterative versions performed constantly good for all flatness parameters. This shows, that a carefully chosen threshold gives good results. Furthermore, weighting the projection equations improved the performance slightly. In contrast to the DRP algorithm, the RefCam and RefFac algorithm, displayed in fig. 6.5(b), do not reconstruct separately points on and off the plane. The performance of the RefFac algorithm is constantly good. This can be expected, since in contrast to the DRP and RefCam algorithm the heights of the points from the finite plane are computed. The performance of the RefCam algorithm is good for "non-flat scenes". However, the performance is unexpectedly unstable for very "flat scenes" (less than 0.4).

We may conclude that the RefFac and DRP method performed stably, independently of the scene's structure. However, the RefFac method has the drawback of not handling missing data naturally. The main drawback of the DRP method is that the iteration process is complex and time consuming, since it depends linearly on the number of 3D points. Note, for scenes where 3D points are not close to the reference plane the more efficient non-iterative DRP version performed good and is strongly recommended. The RefCam algorithm is simpler than the DRP method, since points on and off the reference plane are not separated. However, this method was unexpectedly unstable for very "flat" scenes. Therefore, in our opinion the best reference plane method for "flat scenes" has not yet been found.

Position of the reference plane

The previous experiment discussed the case of a finite reference plane, where points are on or close to the reference plane. As was seen in chapter 5, the reference plane might as well represent the correct plane at infinity. These two cases of a finite and infinite reference plane are analyzed in following for the 3 different algorithms DRP, RefCam and RefFac (fig. 6.6). To compare both cases, we assume that the infinite homographies

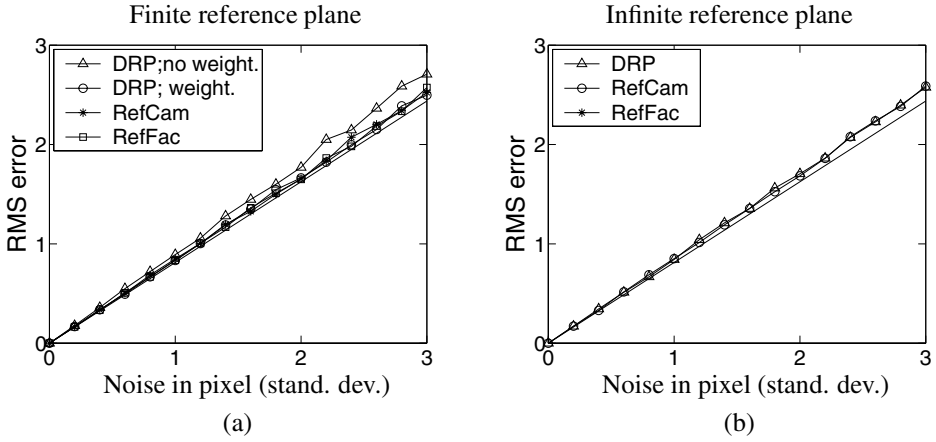


Figure 6.6. Performance of various algorithms for a finite (a) and infinite (b) reference plane.

are not corrupted by noise. For the experiments, the Cir-Configuration (fig 6.2(a)) was chosen, where no 3D point lies close to the finite reference plane. The main observation is that the RefFac method is not applicable for infinite reference planes (fig. 6.6 (b)), RMS error of 107 for no noise. This can be expected since it reconstructs the height of the 3D points from the reference plane. As seen in the previous experiment, the DRP method without weighting performs slightly worse than the version with weighting. For an infinite reference plane, the simple DRP version (without weighting) performs equally good as the weighted DRP version and a finite reference plane. Consequently, weighting eliminates the effect of choosing a finite reference plane. More important, this shows that *for an infinite reference plane, the DRP method is most simple (non-iterative, non-weighting) and performs virtually optimal.* This is due to the fact that all reconstructed 3D points have the same order of magnitude. The RefCam method performs for both finite and infinite reference planes constantly good.

3-Point algorithms

We saw in sec. 5.2.2 and 5.2.3 that general scenes without a reference plane can be reconstructed using the "reference plane approach". The infinite homographies may be derived from 3 arbitrary points (visible in all views), which define the reference plane, and the assumption of parallel projection or known epipolar geometry. This section illustrates only a few of the experiments presented in (Rother and Carlsson, 2002b). The synthetic scene (cube and reference points) in fig. 6.2 (a, b) is replaced by a sphere of radius 2 with 50 randomly distributed points.

Fig. 6.7 shows the performance for affine cameras (a) and projective cameras (b). The performance of the DRP algorithm is compared with the AffFac and ProjFac algorithm. In the affine case, the DRP algorithm performed reasonable good. However, the result of

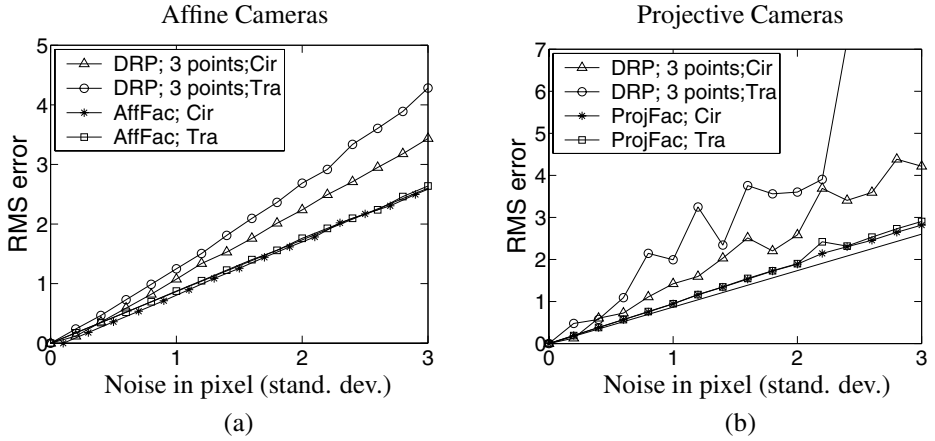


Figure 6.7. Performance of various algorithms for a general scene (without a reference plane) using parallel projection (a) and perspective projection (b).

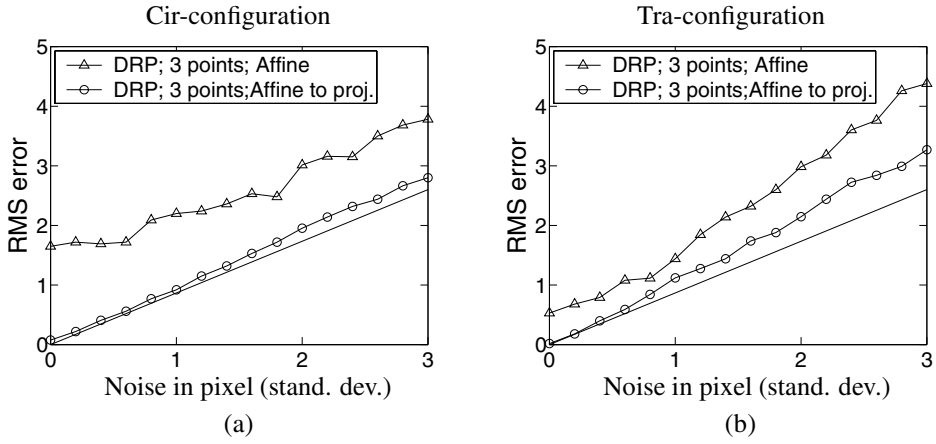


Figure 6.8. Performance of the iterative DRP algorithm, introduced in sec. 5.2.2, for a perspective projection of a general scene (without a reference plane). It is initialized by the DRP algorithm assuming parallel projection (affine).

the AffFac algorithm is significantly better. The main difference between these algorithms is that the DRP algorithm depends heavily on the quality of the 3 reference points. This conjecture is confirmed in the next section. In the projective case, the different performance of the DRP and ProjFac algorithm is even more obvious. Furthermore, the DRP algorithm was less stable for projective cameras (b) than for affine cameras (a). The only difference

between these two cases is that the epipoles are used for the homography computation of the projective cameras. A more detailed analyses confirmed that this computation is fairly sensitive to noise in the epipoles.

In sec. 5.2.2 an iterative DRP algorithm was introduced which assumes parallel projection and compensates iteratively for the perspective effects. Fig. 6.8 shows the performance of this algorithm for projective cameras. It stands out that for both configurations the initial reconstruction (affine) is significantly improved by iteration (affine to projective). In the case of no noise it converged to the theoretical minimum.

The conclusion of these experiments is that the factorization algorithms, which assume no missing data, are superior. This can be expected since they do not distinguish certain points. However, the DRP algorithm might be useful for specific scenarios with a high percentage of missing data and only a few (at least 3) points visible in all views. Further experiments about "hallucinating" a reference plane are in our publication (Rother and Carlsson, 2002b).

Quality of the reference plane

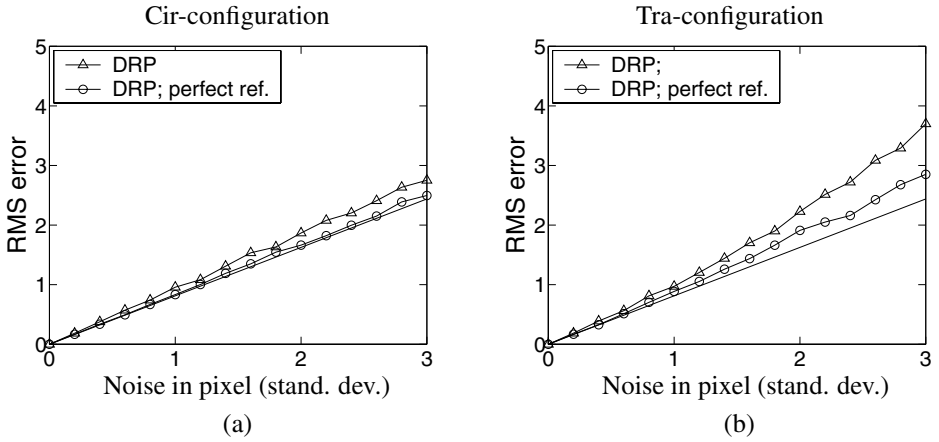


Figure 6.9. Results of the DRP algorithm where the 4 reference points are either perfect or corrupted by noise.

In the following, we will repeat some of the experiments of the previous sections. However, Gaussian noise will be added to all image points *except for* the reference points. Fig. 6.9 shows the performance of the DRP algorithm for the Cir- (a) and Tra-configuration (b). In this case the iterative, weighted version of the DRP algorithm is applied. Obviously, if the reference points are not corrupted by noise (perfect reference), i.e. the infinite homographies are correct, the performance of the DRP algorithm improves. For the Cir-configuration the performance is very close to the theoretical minimum. Fig. 6.10 shows

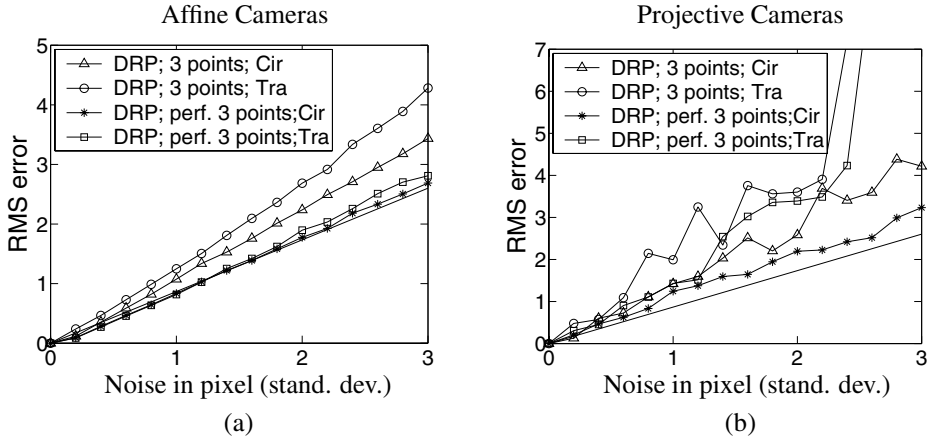


Figure 6.10. Replication of the experiment for general scenes (fig. 6.7) with perfect and noisy reference points.

the experiments for general scenes (fig. 6.7) with and without noise on the reference points. In the affine case (a), the improvement of the performance is significant. However, for perspective projection (b) the improvement is not that evident. As already mentioned above, non-optimal epipoles influence the estimation of the infinite homographies. These experiments lead to the expected conclusion that the quality of the reference points is crucial for the performance of the different DRP algorithms.

Missing data

In all previous experiments it was assumed that all points are visible in all views. In the following we will analyze how various algorithms perform in the case of missing data. Therefore, the Cir-configuration was used consisting of 8 cameras with parallel projection. Apart from the 4 reference points, each scene point is only visible in a fraction of $3-8$ views. This fraction is taken randomly. More realistic scenarios of missing data are analyzed in sec. 6.1.3. Fig. 6.11 depicts the performance for a standard deviation of 1 (a) and 3 (b). The AffFac algorithm handles missing data by the "Rank-3 approximation" algorithm of Jacobs (1997). Jacobs uses all *visible* scene points for the computation of the centroid. However, in case of missing data this is incorrect and leads to significant errors as fig. 6.11 indicates. We corrected this method by computing a Rank-4 approximation of the "projective" measurement matrix (eqn. 3.47)[2]. This version performed stably for all cases of missing data. The DRP version was stable as well with respect to the theoretical minimum. Due to the distinguishing of reference points it performed slightly worse than

[2]The Rank-3 and Rank-4 approximation algorithms of Jacobs (1997) are available at: http://www.neci.nj.nec.com/homepages/dwj/.

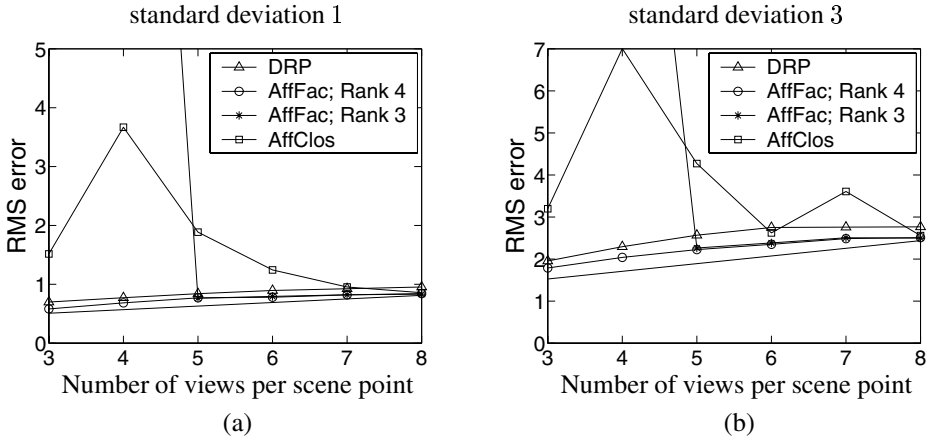


Figure 6.11. Various algorithms for the case of missing data. The Cir-configuration and parallel projection was used together with Gaussian noise of standard deviation 1 (a) and 3 (b).

the AffFac algorithm. In contrast to the DRP and AffFac algorithm is the performance of the AffClos algorithm fairly unstable. A possible explanation is that the image data was not sufficient to obtain reliable trifocal tensors.

6.1.3 Experiments: Real Data

This section represents the main and novel contribution of this chapter. It compares our direct reference plane (DRP) method with several other point-based reconstruction methods for difficult real world scenes with a high percentage of missing data (up to 90%). The important conclusion will be that for such difficult reference plane scenarios our DRP method outperform all other methods. This is due to the fact that common scene knowledge, i.e. the reference plane, is exploited to reconstruct the cameras and the structure simultaneously. In particular, we will analyze the general reconstruction methods **Fmat, FmatBa** (Fitzgibbon and Zisserman, 1998), **ResInt, ResIntBa** (Beardsley et al., 1996) and **ProjFac** (Sturm and Triggs, 1996; Martinec and Pajdla, 2002), the "affine" methods **AffFac** (Tomasi and Kanade, 1992; Jacobs, 1997) and **AffClos** (Kahl and Heyden, 1999) and the plane-based methods **DRP** (Rother and Carlsson, 2002b; Rother and Carlsson, 2002a) and **RefCam** (Hartley et al., 2001). These methods and their abbreviation were introduced in the previous sec. 6.1.2. Additionally, the performance of all methods is analyzed after a final bundle adjustment process. The **RefFac** method (Triggs, 2000) was not analyzed for real data, since its extension for missing data is not straightforward. Furthermore, the joint image closure constraints method of Triggs (1997b) has not been compared.

The advantage of synthetic experiments is that the results can be compared quantitatively with the "ground truth". For real image data the ground truth is most of the time not available. In order to carry out realistic "real world" experiments, the image data was "syn-

thesized". This means that a qualitatively correct metric 3D reconstruction of the scene and the cameras served as a synthetic configuration, i.e. as ground truth.

In the following, 2 experiments using a real, finite reference plane and 3 experiments with an virtual, infinite reference plane are discussed. The virtual reference plane is derived from 3 orthogonal vanishing points and the assumption of a "square pixel" camera. In the first 4 examples, the image points and their correspondences were established manually. For the last example (house), this process was performed completely automatical (see chapter 8).

Real, finite reference plane – Small scale environment

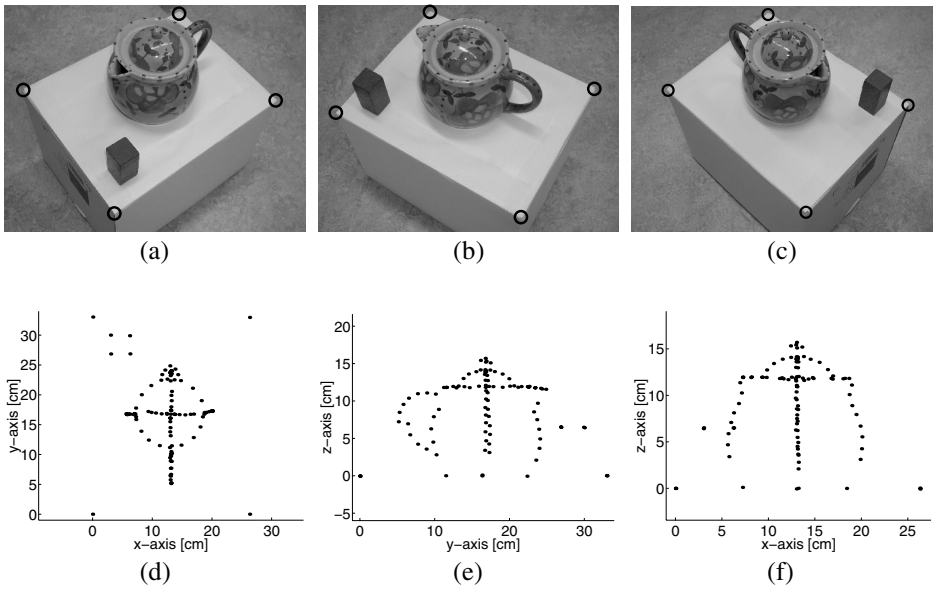


Figure 6.12. Three of the eight original views of the teapot (a-c). The top (d), side (e) and front (f) views of the reconstruction using the iterative DRP algorithm and the four marked reference points.

Figure 6.13. The visibility matrix of the teapot (a) and the tape holder (b). If the jth point is visible in the ith view, the corresponding element $V(i, j)$ is set (a black square).

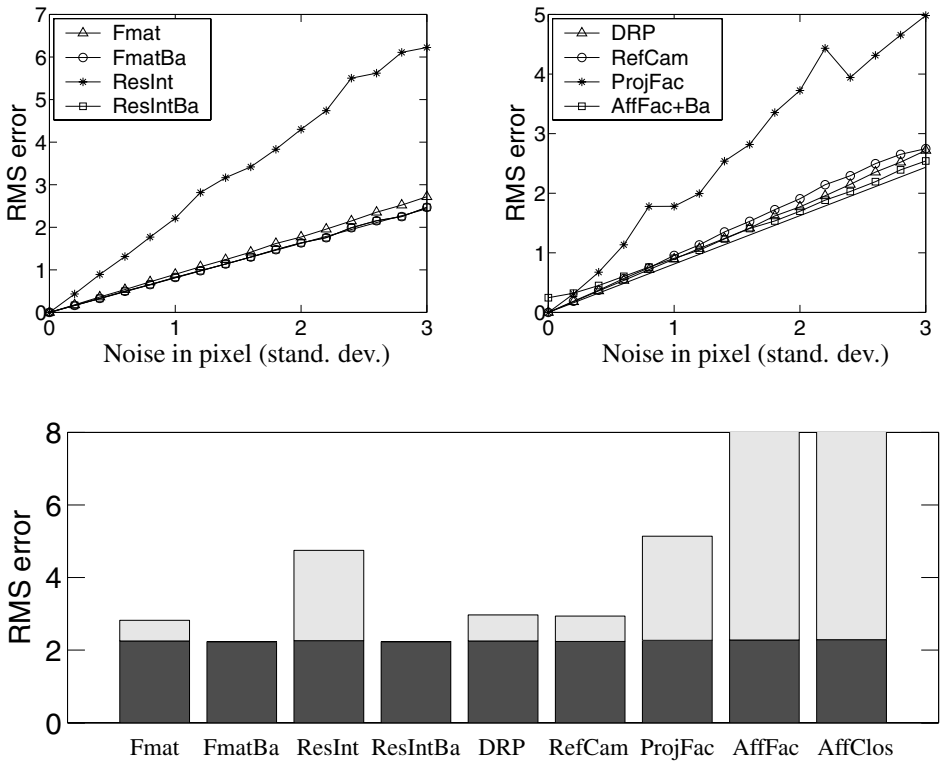


Figure 6.14. Results of various algorithms on the *synthetic* teapot sequence (top) and the *real* teapot sequence (bottom), before bundle adjustment (grey) and after bundle adjustment (black).

In a first experiment a teapot, which was placed on a box, was reconstructed from 8 views (see fig. 6.12(a-c)). The four corner points of the box, which are marked with circles, specify the real reference plane. For a better visualization, only those model points were reconstructed which lie on the contour in the top, side or front view of the model. Fig. 6.13(a) shows the visibility matrix of the teapot sequence. An element $V(i,j)$ of the matrix is set if the jth point is visible in the ith view. In this case 67% of the elements are set. Fig. 6.12(d-f) shows the reconstruction of 104 model points using the iterative DRP method with additional weighting. Model points on the reference plane were detected automatically by iteration. The ratio between the fifth last singular value (360.36) and the fourth last singular value (4.65) is 77.5, i.e. considerably large. The Euclidean coordinates of the cuboid and the reference points were used to rectify the projective reconstruction.

Let us consider the performance of various algorithms for the *synthetic* teapot sequence (fig. 6.14(top)). In this case all points, including the reference points, were corrupted by noise. Most of the algorithms performed very good, i.e. close to the theoretical minimum. This shows that this scenario is, in contrast to the following scenarios, "fairly" simple to

reconstruct. The ResInt method without bundle adjustment performed poor due to the problem of error accumulation. However, if the error is corrected after each resection step by bundle adjustment, i.e. the ResIntBa method, the final RMS error is close to the theoretical minimum. Both reference plane algorithms, i.e. DRP and RefCam, performed similarly good. As already mentioned, the RefFac method is not shown here, since it's extension for missing data, i.e. real image data, is not straightforward. The AffFac+Ba method computes a reconstruction with the assumption of parallel projection and corrects for perspective effects by bundle adjustment. The ProjFac method does perform unexpectedly poor. The experiment with real image data (fig. 6.14(bottom)) confirms the conclusion drawn from the synthetic experiment. Even if the RMS error is fairly different before bundle adjustment, it is identical good for *all* algorithms after bundle adjustment.

In a second experiment a tape holder was reconstructed from 4 views with considerably wide mutual baseline (see fig. 6.15(a-c)). In contrast to the teapot scenario, the tape holder itself contains a real, finite reference plane which is visible in all images. It is defined by the four coplanar points which are marked by circles. Fig. 6.13(b) depicts the visibility matrix which has 20% missing data. The 3D reconstruction of the 32 model points using the iterative DRP algorithm with additional weighting is displayed in fig. 6.15(d-f). In order to visualize the result we assumed knowledge of five Euclidean coordinates to rectify the projective structure. The ratio between the fifth last singular value (0.766) and the fourth last singular value (0.031) is substantially high, i.e. 24.7. By manually selecting points which lie on same model planes we created a VRML model, which consists solely of planes. Fig. 6.15(g-i) depicts 3 novel views of the VRML model.

Consider the *synthetic* tape holder experiment (fig. 6.16(top and middle)). It stands out that nearly all methods performed worse compared to the teapot sequence. Only the three methods FmatBa, ResIntBa and AffFac+Ba, which all use bundle adjustment, performed good. As above, the ProjFac method performed unexpectedly poor. Furthermore, the two reference plane methods, i.e. DRP and RefCam, were very unstable. The difference between the tape holder configuration and the teapot configuration is that scene points and camera centres lie closer to the reference plane (reference points are close to a line in an image) and the reference points lie closer together in an image. This indicates that noisy reference points affect the quality of the infinite homographies more than in the teapot case. Therefore, we repeated the experiment using perfect reference points and perfect infinite homographies (fig. 6.16(middle)). Using perfect infinite homographies means that the correct plane at infinity is used as the reference plane. In case of perfect reference points, the DRP method performs better then the RefCam method. This is the same observation as with synthetic data and a "flat" scene (fig. 6.5(b)). If the correct plane at infinity is used, both algorithms performed good. To summarize, the *quality* and the *position* of the reference plane are important aspects for reference plane methods.

In case of real image data, the results of most algorithms correspond to the results of the synthetic experiments. The general Fmat and ResInt method performed better than the DRP and RefCam method. The ProjFac algorithm produced a better result than in the synthetic case. In general, *all* methods computed a 3D reconstruction which was sufficiently good to perform successfully a bundle adjustment process, which gives a result with a low RMS error of 0.7.

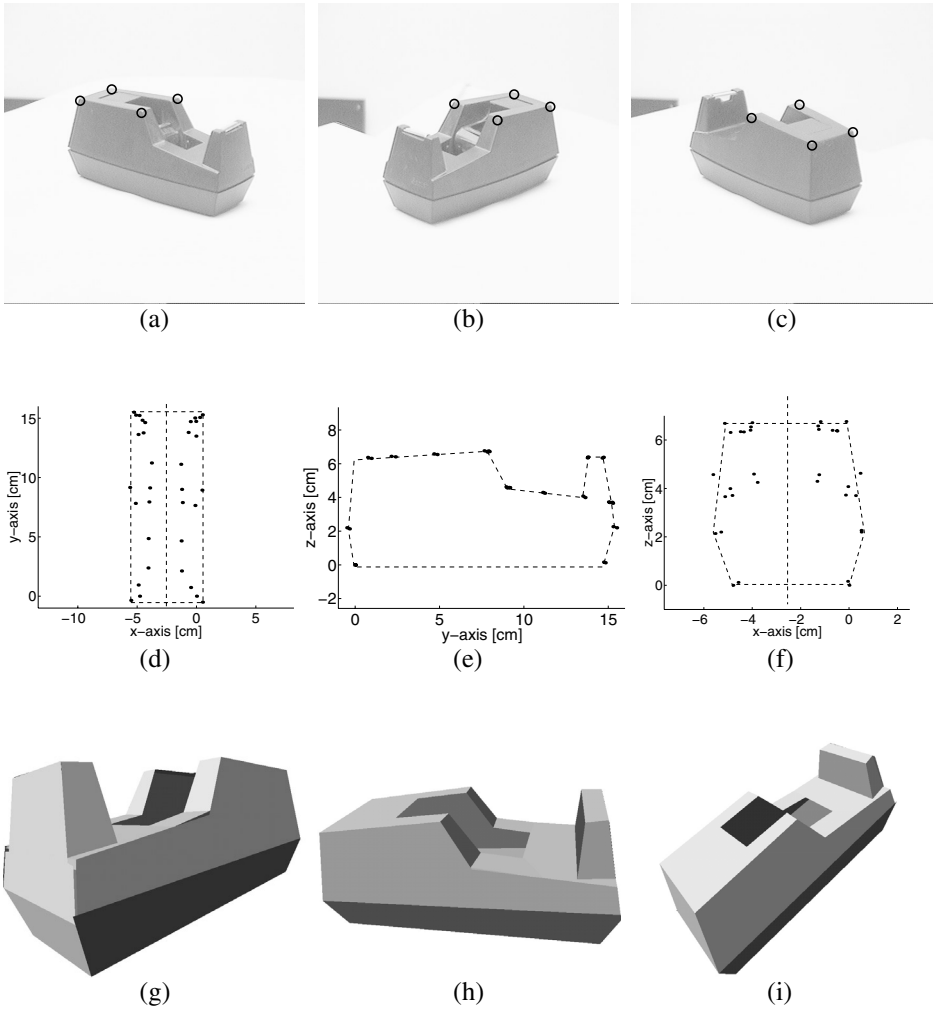


Figure 6.15. Three of the four original views of the tape holder (a-c). The top (d), side (e) and front (f) view of the reconstruction using the iterative DRP algorithm and the four marked reference points. The dashed lines display the contour and the symmetry axis of the model. Three novel views of a VRML model of the tape holder (g-i).

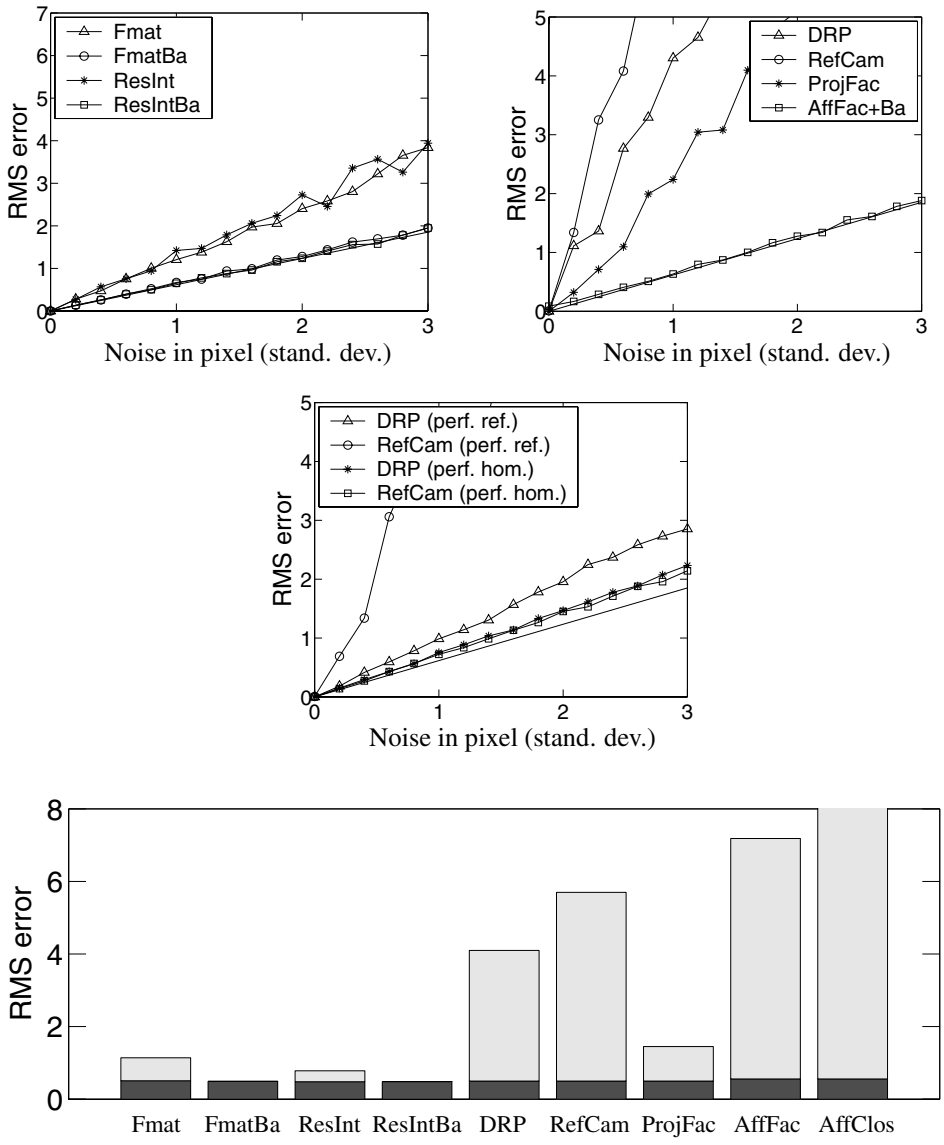


Figure 6.16. Performance of various algorithms on the *synthetic* tape holder sequence (top and middle) and the *real* tape holder sequence (bottom), before bundle adjustment (grey) and after bundle adjustment (black).

Virtual, infinite reference plane – Large scale environment

We will now analyze more difficult, large-scale scenarios with a substantial amount of missing data. Man made environments are often characterized by dominant directions. Figure 6.17(c) shows on the right hand side three buildings with the same orientation. As was seen in sec. 5.1.2, three mutual orthogonal directions can be used to compute the infinite homographies of a "square pixel" camera. In this case the infinite homographies represent a virtual reference plane, the correct plane at infinity. Exploiting dominant directions, in contrast to finite reference points, increases the flexibility and applicability of our DRP method, like reconstructing several buildings. In the following three examples are discussed, the campus, city hall and house example. The vanishing points, which correspond to the dominant directions, were detected manually for the first two examples (campus and city hall) and automatically for the last example (house). Furthermore, the ambiguity in the cameras' rotation was resolved manually for the campus and the city hall example and automatically for the house example. The automatic vanishing point detection and rotation matrix computation methods are described in chapter 8.

In a first experiment, of a large scale environment, we reconstructed three buildings of the campus of the Royal Institute of Technology in Stockholm. The reconstructed area is approximate of size 130×90 meters. 27 images of size 1600×1200pixels were taken with a hand-held of the shelf camera (Olympus 3030) (see fig. 6.17(a, b)). The internal camera parameters remained fix while the pictures were taken. In order to establish a correspondence between the three buildings, we used additionally a postcard of the campus (see fig. 6.17(c)). Naturally, we had no calibration information, e.g. the focal length, of the postcard available. The calibration K and the rotation R, of the cameras were computed from manually selected image lines, which correspond to mutual orthogonal directions in the scene. The camera's calibration was improved by assuming fixed internal camera parameters (sec. 8.3). In case of the postcard, one of the vanishing points is close to infinity (horizontal lines). However, the focal length can still be determined for this degenerate configuration with the additional assumption that the principal point is close to the middle of the image. Furthermore, the correspondences of 191 manually selected model points were manually achieved. The visibility matrix in fig. 6.19 shows that only a few correspondences, i.e. 10.4%, are given. On the basis of this, the campus was reconstructed with our DRP method using the infinite reference plane and the projection equation 6.1. The fourth and fifth last singular value of the SVD were 12.55 and 143.5 respectively, which corresponds to a ratio of 11.44. Fig. 6.18 shows the top view of the reconstruction, where the dots represent reconstructed points, arrows depict cameras and the grey structure represents the superimposed map of the campus. The labeled cameras correspond to images in the respective figures. The accurate match between the top view of the reconstruction and the true map of the campus demonstrates the high quality of the reconstruction. We stress that no further constraints, e.g. orthogonality, were imposed which would presumably improve the reconstruction. As in the previous experiment, a VRML model is determined by manually selecting model points which lie on same model planes. By projecting image texture onto the planes, the final VRML model of the campus is acquired. Fig. 6.20 shows 6 novel views of the VRML model.

(a) (b) (c)

Figure 6.17. Two original views (a, b) and a postcard (c) of the campus. The corresponding camera positions are labeled in the top view (fig. 6.18).

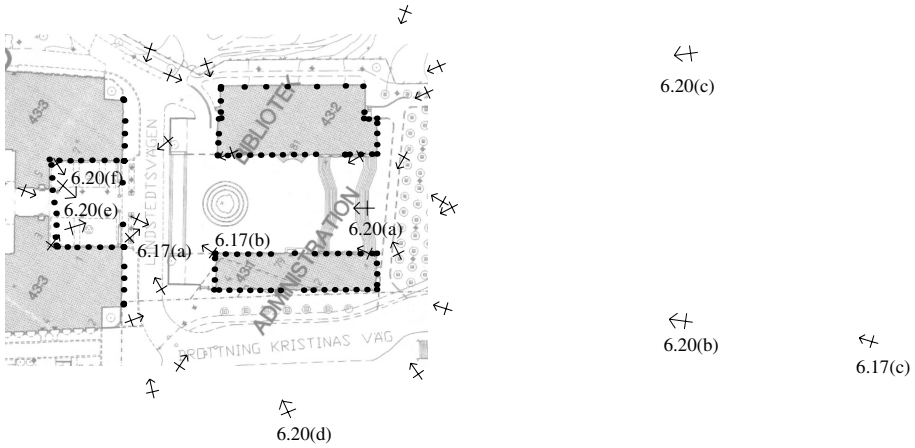


Figure 6.18. Top view of the reconstruction of the campus with 191 model points (dots) and 27 cameras (arrows). A map of the campus is superimposed. The labeled cameras correspond to images in the respective figures.

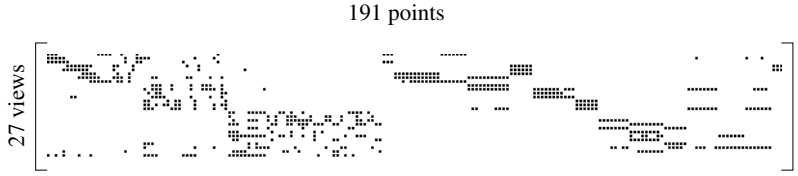


Figure 6.19. The visibility matrix of the campus.

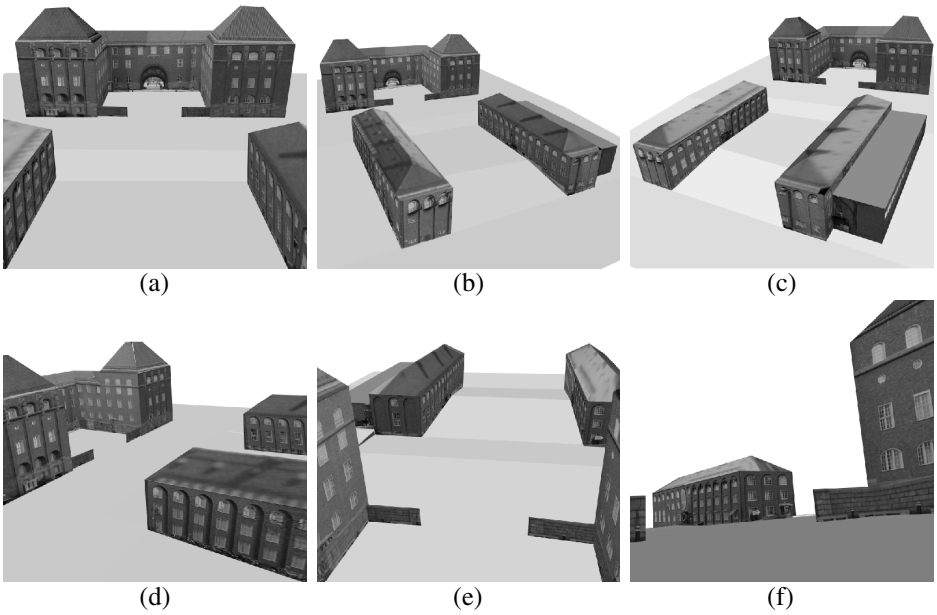


Figure 6.20. Six novel views of the campus. The corresponding camera positions are labeled in the top view (fig. 6.18).

On the basis of the 3D reconstruction in 6.18, synthetic experiments were conducted (fig. 6.21(top)). The first observation is that the image data was not sufficient to obtain a reconstruction with the ResInt, ResIntBa, ProjFac, AffFac and AffClos method. In case of the ResInt and ResIntBa method at least 6 reconstructed model points must be visible in a new camera for resectioning. The ProjFac and AffFac method build on the "Rank approximation" method of Jacobs (1997). However, this method did not converge, i.e. could not fill in all missing elements. The AffClos method requires a minimum of 4 points in 3 successive views to compute the affine trifocal tensor. Only the Fmat and FmatBa method and the two plane-based methods, i.e. DRP and RefCam, were applicable in this case. However, the Fmat and FmatBa method did only produce an acceptable result in case of no noise. Both plane-based methods could reconstruct the scene for different noise levels. The DRP method was, however, significantly superior. Note, since the infinite homographies do not rely on finite scene points, they were not corrupted by noise. With real image data (fig. 6.21(bottom)) the Fmat and FmatBa method did not converge either. The DRP method gave a better result than the RefCam method before bundle adjustment. Both methods converged to the same local minimum after bundle adjustment.

In a second experiment we reconstructed the outside and inside (courtyard) of the city hall in Stockholm. The city hall has an approximated top view size of 130×80 meters. Therefore, 37 images of size 1600×1200pixels were taken, where the internal camera parameters remained fix (see fig. 6.22 and 1.1). As in the previous experiment, the manu-

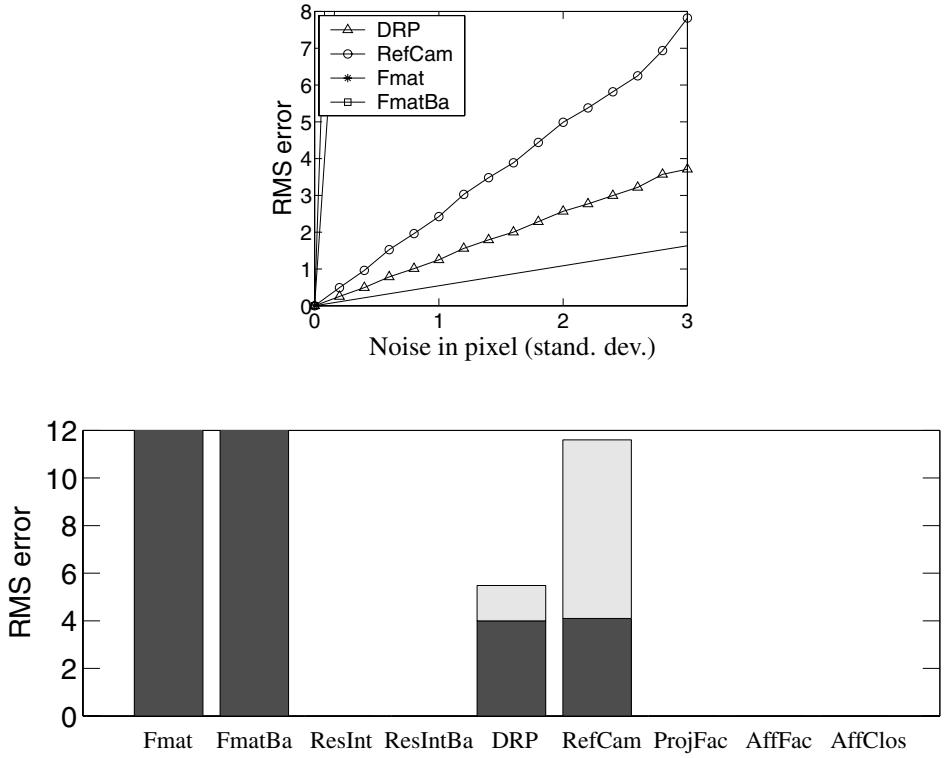
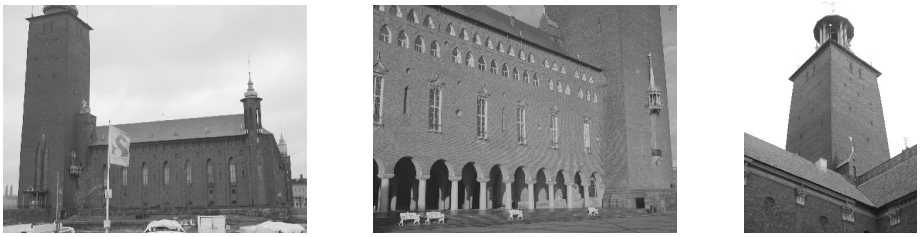


Figure 6.21. Results of various algorithms on the *synthetic* campus sequence (top) and the *real* campus sequence (bottom), before bundle adjustment (grey) and after bundle adjustment (black).

ally selected image lines, which correspond to mutual orthogonal directions in the scene, were used to determine the infinite homography of a "square pixel" camera. Since some parts of the building can be seen from both the outside and inside, e.g. the tower (see fig. 6.22 (a-c)), a correspondence between the outside and inside can be established. Fig. 6.24 depicts the extremely sparse visibility matrix with 9.7% of set elements. With the knowledge of the correspondences of 134 model points, the building was reconstructed with our DRP method. The ratio between the fifth last singular value (57.24) and the fourth last singular value (12.75) was 4.49, i.e. considerable larger than 1. The top view of the reconstruction with a superimposed map of the city hall is shown in fig. 6.23. Fig. 6.25 displays 6 novel views of the textured VRML model of the building. The roof was not reconstructed since it cannot be seen from a ground plane position. As in the previous example, no further constraints were imposed, in order to improve the reconstruction. Let us consider the quality of the reconstruction (see fig. 6.23). It stands out that the building was not designed as a perfect rectangular building. However, this fact did not considerably affect the good reconstruction. The fact that the detected vanishing points are not perfectly



(a) (b) (c)

Figure 6.22. Three original views of the city hall. The corresponding camera positions are labeled in the top view (fig. 6.23).

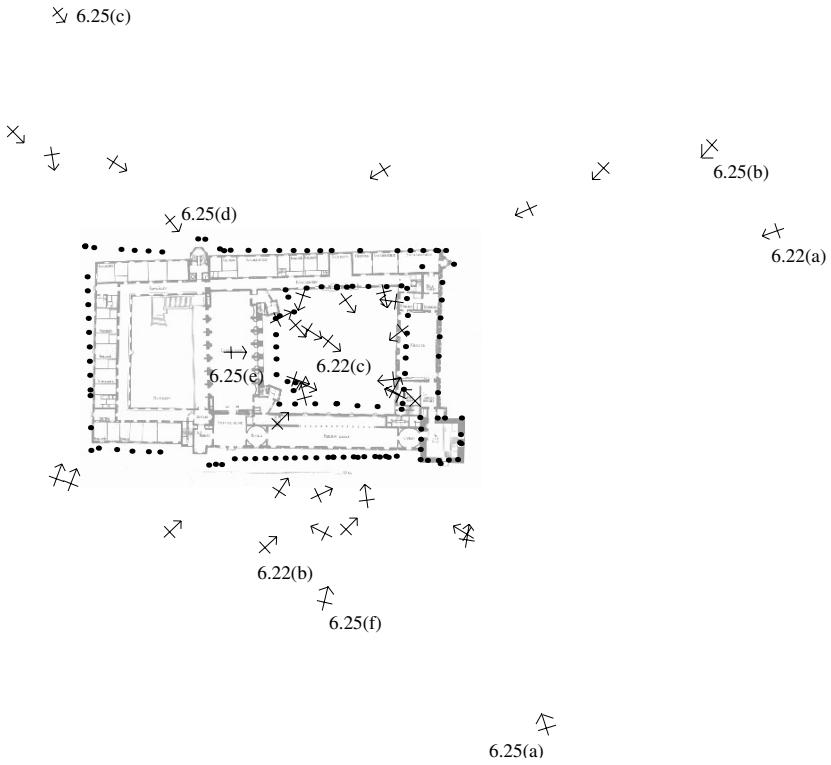


Figure 6.23. Top view of the reconstruction of the city hall with 134 model points (dots) and 37 cameras (arrows). A map of the city hall is superimposed. The labeled cameras correspond to images in the respective figures.

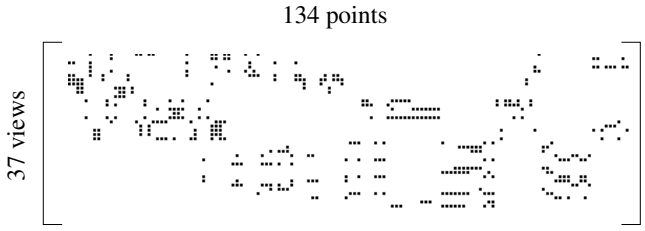


Figure 6.24. The visibility matrix of the city hall

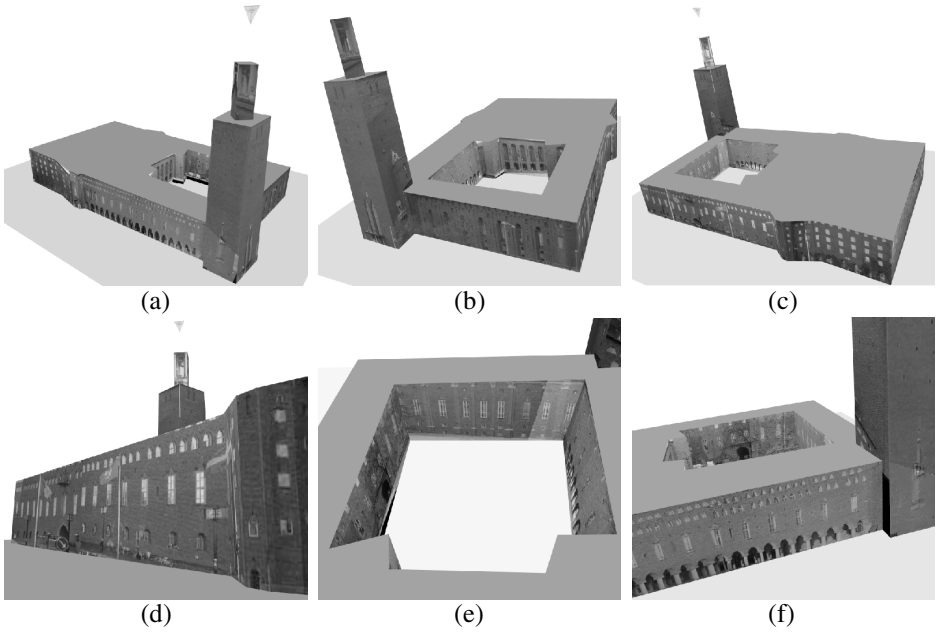


Figure 6.25. Six novel views of the city hall. The corresponding camera positions are labeled in the top view (fig. 6.23).

mutually orthogonal influences the camera calibration as well as the estimation of the rotation matrix R. Since the accuracy of R directly affects the camera's position, we would expect a higher "positioning error" for cameras with less accurate R. This reasoning would explain the deviation between the reconstruction and the true map at the top, left corner of the building.

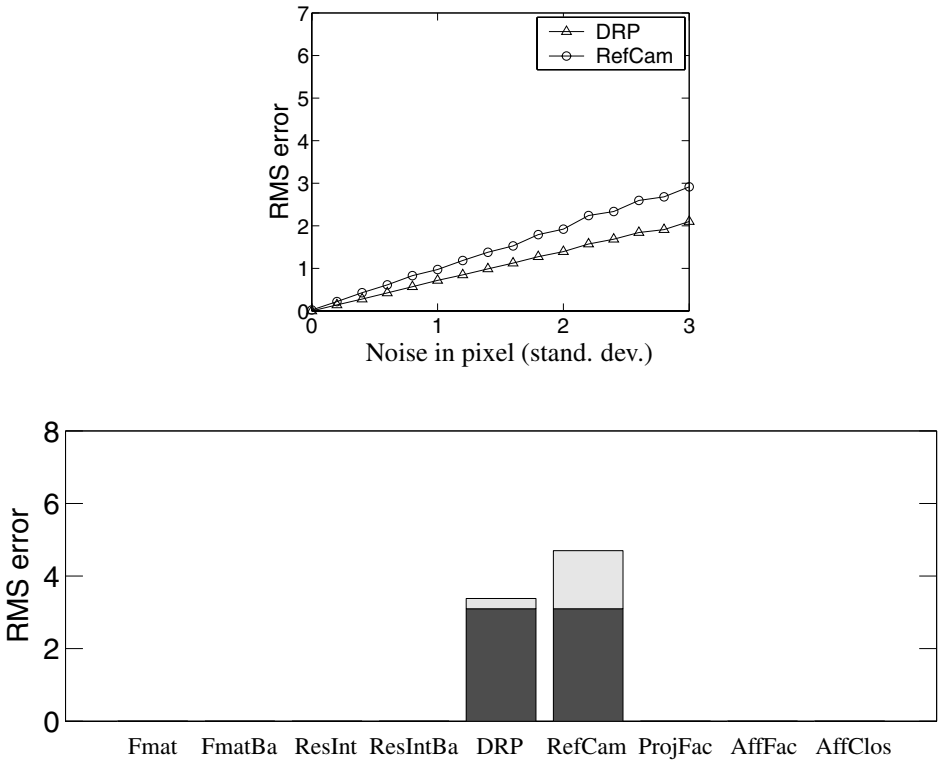


Figure 6.26. Results of various algorithms on the *synthetic* city hall sequence (top) and the *real* city hall sequence (bottom), before bundle adjustment (grey) and after bundle adjustment (black).

Let us consider the *synthetic* city hall sequence (fig. 6.26(top)). In this case only the plane-based methods DRP and RefCam were applicable. The image data was not sufficient for all other general reconstruction methods. The Cramer-Rao lower bound is 0 since the number of unknowns is larger than the number of constraints. As in the previous experiment, the DRP method was significantly superior to the RefCam method. The same applies to the experiment with real image data (fig. 6.26(bottom)). However, they converged to the same local minimum after bundle adjustment.

Figure 6.27. Three of the nine original views of the house sequence. The corresponding camera positions are labeled in the top view (fig. 6.28).

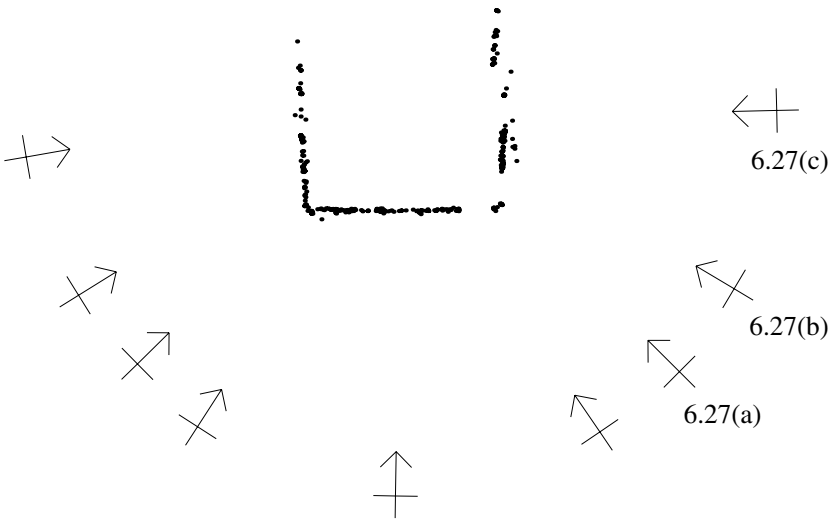


Figure 6.28. Top view of the reconstruction of the house sequence with 451 model points (dots) and 9 cameras (arrows). The labeled cameras correspond to images in fig. 6.27.

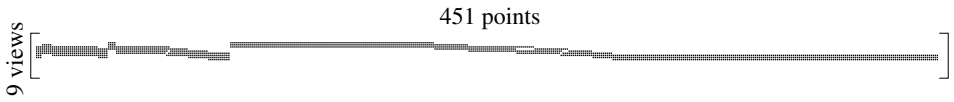


Figure 6.29. The visibility matrix of the house sequence

In a last experiment a house, which has an approximated top view size of 10×10 meters, was reconstructed from 9 images (fig. 6.27). As in the previous cases, the internal camera parameters remained fix during the capturing process, which can be exploited in the calibration process. In contrast to the previous experiments, the complete reconstruction process was automated as far as to the 3D point reconstruction. This includes Harris corner detection, matching, vanishing point detection, camera calibration and rotation matrix estimation (see chapter 8 for details). Fig. 6.29 depicts the visibility matrix for 451 detected model points, where only 36.7% of the image data is available. Most of the scene points are visible in 3 successive views, which gives the visibility matrix a diagonal form. A top view of the reconstructed scene and cameras is shown in fig. 6.28. In this case, the ratio between the fifth last ($2.7 * 10^{-5}$) and fourth last ($4.8 * 10^{-3}$) singular value is 175.5.

Let us consider the *synthetic* house sequence (fig. 6.30(top)). Since more image data is available than in the campus or city hall experiment, all methods were applicable. The results are analyzed in terms of the 3D error between the reconstruction and the ground truth after aligning both in an optimal way. One unit of the 3D error is approximately one meter. The reason for choosing a different measurement is that in this case some incorrect 3D reconstruction, i.e. a large 3D error, had a small RMS error, e.g. the ResIntBa method in fig. 6.30 middle and bottom. The general ResInt, ResIntBa, Fmat and FmatBa method could reconstruct the scene only up to a noise level of 0.4. A possible explanation for the failure of the Fmat and FmatBa method is that the scene is close to a critical configuration. Consider the view in the middle, i.e. number 5, in fig. 6.28. This view and any other view observe only a real scene plane, which is a critical configuration (chapter 7). This problem appears frequently in man-made environments and was addressed in (Pollefeys et al., 2002). In contrast to the "sequential" methods, *both plane-based methods, i.e. DRP and RefCam, performed excellent for noisy input data.* Due to the fact that these methods exploit common scene knowledge, the critical configuration of one real scene plane is circumvented. Fig. 6.30 depicts the results of different algorithms for real image data in terms of RMS error (middle) and 3D error (bottom). For the 3D error, the reconstruction of the DRP algorithm was taken as "ground truth". Only the ResInt, DRP and RefCam method produced an acceptable result. The ResIntBa and ProjFac method had a "fairly" low RMS error, however, the 3D error was high, i.e. about 20 meters. The difference between the result of the ResInt method and the plane-based methods is approximately $1meter$, i.e. fairly large. Both plane-based methods, i.e. DRP and RefCam, give a low RMS error of 0.7. This is less than in the campus and city hall experiment, since automatically detected image points are in general more accurate.

Let us summarize the result from the real world experiments. *The main conclusion of the real world experiments is that for difficult scenes with a high percentage of missing data, up to 90%, our DRP methods and Hartley et al.'s (2001) RefCam method outperform all other general reconstruction methods.* In particular, we analyzed the general reconstruction methods **Fmat, FmatBa, ResInt, ResIntBa, ProjFac**, the "affine" methods **AffFac, AffClos** and the plane-based methods **DRP, RefCam**. The "failure" of the general and "affine" methods had the following reasons: (a) too few image measurements are available, (b) error accumulation due to noisy image measurements or (c) critical configurations (one real scene plane). The plane based methods circumvent all these problems

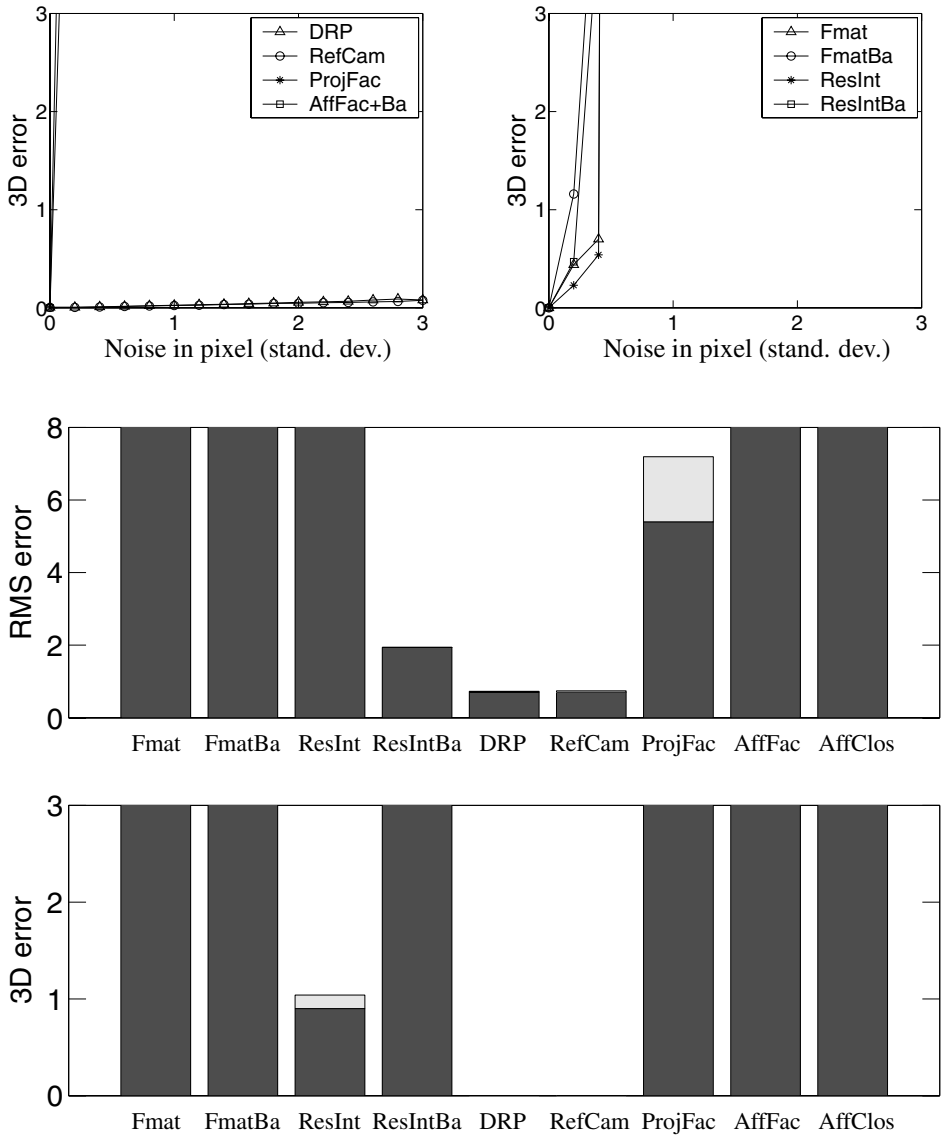


Figure 6.30. Performance of various algorithms on the *synthetic* house sequence (top). The RMS-error (middle) and the 3D error (bottom) of the *real* house sequence before bundle adjustment (grey) and after bundle adjustment (black).

since they exploit scene knowledge which is available in all images, e.g. vanishing points, in order to reconstruct the cameras (and the structure) simultaneously. Our DRP method performed for some scenarios significantly better than the RefCam method. However, both methods converged to the same local minimum after bundle adjustment in all experiments. The "tape holder" sequence showed that *reference plane methods are inferior to general methods if the reference plane is detected very inaccurately.*.

6.2 Lines

We begin this section with an outline of our novel direct reference plane method for lines, Line-DRP method. Since a 3D line can be represented in different ways, three versions of the Line-DRP algorithm are suggested (sec. 6.2.1). The performance of the different Line-DRP methods are then analyzed in experiments based on real and synthetic data (sections 6.2.2 and 6.2.3). Since lines features are not as frequently used as point features, for the task of reconstruction, fewer experiments are conducted here. The goal is to show that the two reference plane methods, Line-DRP and Line-Cam, perform successfully under various conditions using real and synthetic data. The Line-Cam method is an extension of (Hartley et al., 2001) for line features (sec. 3.3.3). It will turn out that the Line-DRP method is superior to the Line-Cam method for scenes were 3D lines are not "close to" the reference plane, e.g. an infinite reference plane. For a real world scenario, with a high percentage of missing data, both methods are, however, significantly inferior to our DRP method for points. A comparative study with other line-based reconstruction methods was not carried out. The outline of our Line-DRP method and the experimental study was not published earlier.

6.2.1 Outline of the DRP Method & Optimization

The presentation of our Line-DRP algorithm in sec. 3.3.2 omitted several practical issues, such as normalizing the image lines and separating 3D lines on and off a finite reference plane. These issues are discussed here.

Different types of projection equations

On the basis of two different line representations, sec. 3.3.1 introduced constraints (eqn. 3.62 and 3.66) which are linear in the unknown line and camera parameters. As in the point case, we will formulate these linear constraints for general cameras and normalized image lines.

Consider the representation of a 3D line $\mathbf{L}_i$ by two points $\bar{\mathbf{X}}_i$ and $\bar{\mathbf{X}}'_i$. The constraints in eqn. 3.62, for finite cameras, may be written for a general camera $P_j = [H_j^\infty \mid \mathbf{t}_j]$, where the camera centre may be on the reference plane, as:

$$\begin{aligned} \bar{\mathbf{X}}_i^T \, H_j^{\infty T} \, \mathbf{l}_{ij} \; + \; \mathbf{t}_j^T \, \mathbf{l}_{ij} \;\; &= \;\; 0 \quad \text{and} \\ \bar{\mathbf{X}}_i^{\prime T} \, H_j^{\infty T} \, \mathbf{l}_{ij} \; + \; \mathbf{t}_j^T \, \mathbf{l}_{ij} \;\; &= \;\; 0 \; . \end{aligned} \tag{6.9}$$

These two equations are still linear in the unknown parameters $\bar{\mathbf{X}}_i, \bar{\mathbf{X}}'_i, \mathbf{t}_j$. As in the point case, it is preferable that the image lines are normalized in each image. This can be achieved by computing the normalization matrix B_j, in each image j, from the set of all endpoints of the line segments (sec. 6.1.1). According to proposition 2 (sec. 2.1.1), the line segments are normalized as $\mathbf{l}'_{ij} = B_j^{-T} \mathbf{l}_{ij}$. For normalized image lines $\mathbf{l}'_{ij}$ the constraints in eqn. 6.9 may be formulated as

$$\begin{aligned} \bar{\mathbf{X}}_i^T \, (B_j \, H_j^\infty)^T \, \mathbf{l}'_{ij} \; + \; \mathbf{t}_j^T \, B_j^T \, \mathbf{l}'_{ij} \;\; &= \;\; 0 \quad \text{and} \\ \bar{\mathbf{X}}_i^{\prime T} \, (B_j \, H_j^\infty)^T \, \mathbf{l}'_{ij} \; + \; \mathbf{t}_j^T \, B_j^T \, \mathbf{l}'_{ij} \;\; &= \;\; 0 \; . \end{aligned} \tag{6.10}$$

Since the matrices B_j and H_j are known, these constraints are, as in eqn. 6.9, linear in the unknown parameters $\bar{\mathbf{X}}_i, \bar{\mathbf{X}}'_i$ and $\mathbf{t}$. Note that in order to specify the 2 degrees of the 3D points $\bar{\mathbf{X}}_i$ and $\bar{\mathbf{X}}'_i$, eqn. 6.4, for normalized image points, must be used.

Consider the case where a 3D line $\mathbf{L}_i$ is represented by the minimum number of unknown parameters, d_i and d'_i. The constraints for this line representation were introduced in eqn. 3.66. For general cameras $P_j = [H_j^\infty \mid \mathbf{t}_j]$ and normalized image lines $\mathbf{l}'_{ij}$, the 3 constraints in eqn. 3.66 may be formulated as

$$\begin{vmatrix} (B_j H_j^\infty)^{T1} \mathbf{l}'_{ij} & \mathbf{n}_{i,x} & \mathbf{n}'_{i,x} \\ (B_j H_j^\infty)^{T2} \mathbf{l}'_{ij} & \mathbf{n}_{i,y} & \mathbf{n}'_{i,y} \\ (B_j \mathbf{t}_j)^T \mathbf{l}'_{ij} & d_i & d'_i \end{vmatrix} = 0 \; , \quad \begin{vmatrix} (B_j H_j^\infty)^{T1} \mathbf{l}'_{ij} & \mathbf{n}_{i,x} & \mathbf{n}'_{i,x} \\ (B_j H_j^\infty)^{T3} \mathbf{l}'_{ij} & \mathbf{n}_{i,z} & \mathbf{n}'_{i,z} \\ (B_j \mathbf{t}_j)^T \mathbf{l}'_{ij} & d_i & d'_i \end{vmatrix} = 0 \; ,$$

$$\begin{vmatrix} (B_j H_j^\infty)^{T2} \mathbf{l}'_{ij} & \mathbf{n}_{i,y} & \mathbf{n}'_{i,y} \\ (B_j H_j^\infty)^{T3} \mathbf{l}'_{ij} & \mathbf{n}_{i,z} & \mathbf{n}'_{i,z} \\ (B_j \mathbf{t}_j)^T \mathbf{l}'_{ij} & d_i & d'_i \end{vmatrix} = 0 \; , \tag{6.11}$$

where A^i represents the ith row of A. Since the unknown parameters, $\mathbf{t}_j, d_i$ and d'_i, do only appear in the last row of each determinant, this gives three equations which are linear in the unknown parameters.

To summarize, the constraints in eqns. 6.10 and 6.11 may be used instead of the constraints in eqns. 3.62 and 3.66 for the reconstruction of multiple lines in multiple views.

Outline of the algorithms

Using the different constrains for multiple lines, we formulate now three linear algorithms for the reconstruction of multiple lines and cameras. The different linear systems, using eqn. 3.62 or 3.66, were introduced in sec. 3.3.2. This section also discussed the main advantages and drawbacks of this approach.

We begin with the simplest algorithm **Line-DRP**. It is based on the 2 point representation of a 3D line and uses the linear system in eqn. 3.69. The algorithm is composed of the following steps

1. Determine H_j of a reference plane.
2. Compute the image lines $\left\|\mathbf{l}'_{ij}\right\|_2$ (with $\mathbf{l}'_{ij} = B_j^T\mathbf{l}$), or $\left\|\mathbf{l}''_{ij}\right\|_2$ (with $\mathbf{l}''_{ij} = H_j^T\mathbf{l}$).
3. Obtain $\bar{\mathbf{X}}_i, \bar{\mathbf{X}}'_i, \bar{\mathbf{Q}}_j(\mathbf{t}_j)$ by SVD using eqns. 6.10, 6.4 (for $\mathbf{l}'$) or 3.62, 3.13 (for $\mathbf{l}''$).

The image lines $\mathbf{l}'$ are for general cameras and normalized image lines. In case of stabilized images, the image lines $\mathbf{l}''$ are applied. Note that it is, as in the point case, more efficient to compute only the matrix V, i.e. its last four singular vectors, from the SVD of the system matrix $S = UDV^T$.

The Line-DRP method uses the image measurements directly. However, if any of the 3D points $\bar{\mathbf{X}}_i$ or $\bar{\mathbf{X}}'_i$ lie on the reference plane, the Line-DRP method does not return the correct solution. In sec. 3.3.1 it was shown that the direction V_i of a 3D line $\mathbf{L}_i$ can be determined directly from the multiple stabilized image lines $\mathbf{l}_{i1}, \ldots, \mathbf{l}_{im}$. This is achieved by computing the intersection point $\mathbf{v}_i$ of the image lines, using the linear system in eqn. 3.63. The direction of a line $\mathbf{L}_i$ is then given as $\mathbf{V}_i = (\mathbf{v}_i, 0)^T$. The linear system in eqn. 3.63 may also be used to determine whether a 3D line lies on or off the reference plane. The second last singular value of the system is 0 for a 3D line on the reference plane. Since $\mathbf{V}_i$ is a point on the line $\mathbf{L}_i$, i.e. $\mathbf{X}'_i = \mathbf{V}_i$, it is sufficient to reconstruct only one point $\bar{\mathbf{X}}_i$, in order to determine $\mathbf{L}_i$ completely. In practice, the point $\bar{\mathbf{X}}_i$ may be chosen as the endpoint, of a line segment $\mathbf{l}_{i1}$, which is further away from the vanishing point $\mathbf{v}_i$. Since $\mathbf{V}_i$ is at infinity, $\bar{\mathbf{X}}_i$ must be a finite point. The outline of this algorithm, based on the linear system in 3.70, is

1. Determine H_j of a reference plane.
2. Compute the image lines $\left\|\mathbf{l}'_{ij}\right\|_2$ (with $\mathbf{l}'_{ij} = B_j^T\mathbf{l}$), or $\left\|\mathbf{l}''_{ij}\right\|_2$ (with $\mathbf{l}''_{ij} = H_j^T\mathbf{l}$).
3. Compute the direction V_i of each 3D line $\mathbf{L}_i$ using $\mathbf{l}''$ and eqn. 3.63.
4. Determine and reconstruct 3D lines on the reference plane by checking if the second last singular value in eqn. 3.63 is smaller than a certain threshold.
5. Obtain $\bar{\mathbf{X}}_i, \bar{\mathbf{Q}}_j(\mathbf{t}_j)$ by SVD using eqns. 6.10, 6.4 (for $\mathbf{l}'$) or 3.62, 3.13 (for $\mathbf{l}''$).

We denote this method **Line-DRP(1p)**. It is, in contrast to the Line-DRP method, more efficient since the linear system comprises of fewer unknowns.

Finally, a 3D line $\mathbf{L}_i$ may be represented by a minimum of unknown parameters d_i and d'_i. These two parameters represent the distance of the planes $\mathbf{\Pi}_i = (\mathbf{n}_i, d_i)^T$ and $\mathbf{\Pi}'_i = (\mathbf{n}'_i, d'_i)^T$ from the origin. The planes' normals $\mathbf{n}_i$ and $\mathbf{n}'_i$ may be derived directly from the direction $\mathbf{V}_i$ of the line. On the basis of this line representation, we formulate the 3 equations of the form eqn. 3.66, for stabilized images, and eqn. 6.10. These equations may be stacked into a linear system shown in eqn. 3.71. In contrast to the previous methods, the "artificial" extra constraints for the 3D points $\bar{\mathbf{X}}_i$ and $\bar{\mathbf{X}}'_i$ are dispensable. The algorithm, which we called the **Line-DRP(min)** method, is composed of the following steps

1. Determine H_j of a reference plane.
2. Compute the image lines $\left\|\mathbf{l}'_{ij}\right\|_2$ (with $\mathbf{l}'_{ij} = B_j^T\mathbf{l}$), or $\left\|\mathbf{l}''_{ij}\right\|_2$ (with $\mathbf{l}''_{ij} = H_j^T\mathbf{l}$).
3. Compute the direction V_i of each 3D line $\mathbf{L}_i$ using $\mathbf{l}''$ and eqn. 3.63.
4. Determine and reconstruct 3D lines on the reference plane by checking if the second last singular value in eqn. 3.63 is smaller than a certain threshold.
5. Obtain $d_i, d'_i, \bar{\mathbf{Q}}_j(\mathbf{t}_j)$ by SVD using eqns. 6.11 (for $\mathbf{l}'$) or 3.66 (for $\mathbf{l}''$).

This methods has, in contrast to the Line-DRP(1p) method, even less number of unknowns. On the other hand, it derives more information, i.e. $\mathbf{n}_i, \mathbf{n}'_i$, directly from the image data.

In contrast to the point based DRP method (sec. 6.1.1), the distance between the 3D lines and the reference plane was not computed for any of the Line-DRP methods. Consequently, no weighting of the linear equations was suggested and the separation of lines on and off the plane was solved by a simple thresholding. Is it possible to compute this distance? If the line is represented by one or two 3D points, i.e. $\bar{\mathbf{X}}$ and $\bar{\mathbf{X}}'$, the distance of these points to the reference plane could be established. However, their correspondence in multiple views is unknown. Therefore, determine this distance is difficult or probably impossible. The effect of a non-weighted linear system on the solution will be demonstrated in the experimental section. Note, if the reference plane represents the correct plane at infinity this issue can be neglected. Alternatively, the equations of the linear system for 3D lines could be weighted according to the length of a line segment. This would reflect the fact that longer line segments are detected more accurately than shorter ones.

6.2.2 Experiments: Synthetic Data

This section investigates the performance of different versions of our **Line-DRP** method under specific aspects, e.g. position of the reference plane. We will see that our novel method performs successfully under various conditions. For comparison, we also analyze a linear reference plane method which is based on camera constraints obtained from line segments (sec.3.3.3). This method is an extension of (Hartley et al., 2001) for line features. We denote it **Line-Cam**. The comparison between the Line-DRP and the Line-Cam method will reveal that the Line-DRP is superior for scenes where 3D lines lie not close to the reference plane. Otherwise, the camera constraint method was slightly better, since our method does not weight 3D lines according to their distance to the plane. The main advantage of our Line-DRP method is that lines *and cameras* are estimated simultaneously. A comparison with other line reconstruction methods, e.g. hierarchical merging of trifocal tensors, has not been carried out.

The synthetic experiments for line features were conducted in the same way as for point features. However, since lines features are not as frequently used as point features, less extensive experiments were conducted here. For the experiments the synthetic Circonfiguration in fig. 6.2(a) was used. The synthetic scene consists of a cube, with 24 lines segments, floating above a reference plane. As in the point case, the infinite homographies are derived from the 4 reference points, which lie on the reference plane. In order to simulate noisy line segments, Gaussian noise was added to the endpoints of the line

segments. In contrast to the point case, it is difficult to compare a 3D line reconstruction with the "ground truth". Therefore, the 3D endpoints of the line segments were determined by intersection using the reconstructed cameras. The Root-Mean-Square (RMS) error between reprojected 3D points and 2D endpoints of the line segments (potentially corrupted by noise) served as the quantitive measurement of the performance. As in the point case, the Cramer-Rao lower bound, which is depicted as a straight line, indicates the theoretical minimum.

Different Line-DRP versions

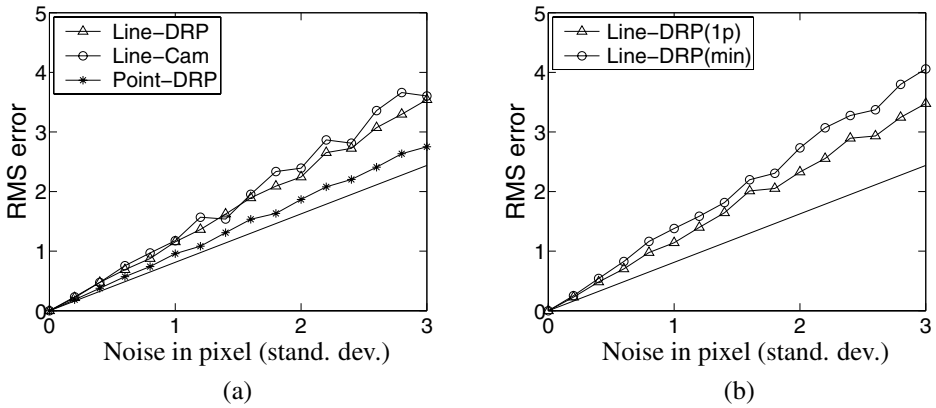


Figure 6.31. Performance of the Line-Cam and various Line-DRP algorithms on the synthetic Cir-Configuration (see fig. 6.2(a)).

In a first experiment, different versions of the Line-DRP algorithm were analyzed, depending on the the different constraints discussed in the previous section (see fig. 6.31). The performance of the the Line-DRP, Line-DRP(1p) and Line-Cam method is very similar. The Line-DRP(min) algorithm performed in this case slightly worse. The difference between this version and the other Line-DRP algorithms is that more information is used, which is derived directly from the reference plane and the image lines. This potentially decreases the numerical stability of the Line-DRP(min) method. The performance of the different Line-DRP versions using the *normalized* image lines, i.e. eqn. 6.10 or 6.11, were close to identical to the algorithms which apply *non-normalized* image lines. Therefore, this is not discussed here. The Line-DRP algorithm is used for the following experiments, in case no line segment lies on or close to the reference plane.

Let us investigate the difference between point and line features. The 26 endpoints of the 24 line segments were reconstructed with the DRP method based on point features. Fig. 6.31(a) shows, that the performance is more stable in the point case. This can be expected since 3D lines provide fewer geometric constraints than 3D points.

Thresholding

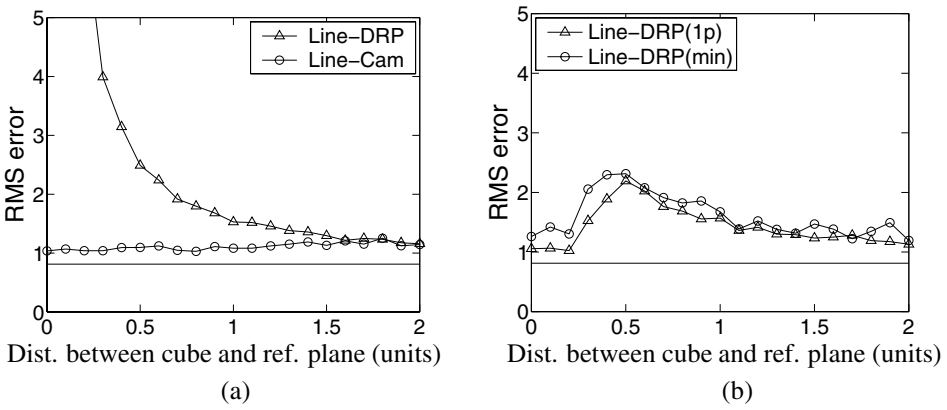


Figure 6.32. Performance of the Line-Cam and various Line-DRP algorithms in respect to the distance between the cube and the reference plane.

We saw that 3D lines which lie on or close to the reference plane have to be reconstructed separately. Otherwise, the linear system in eqn. 3.69, 3.70 or 3.71 does not provide the correct solution. How the different algorithms handle this issue is investigated experimentally here. Fig. 6.32 depicts an experiment were the distance between the cube and the reference plane varied between 0 and 2 units (see fig. 6.2(a)). If the distance is 0, 6 out of 24 3D lines of the cube lie on the reference plane. The performance of the different algorithms is as expected. The Line-DRP algorithm, which does not consider the issue of separating lines on and off the plane, performs gradually worse corresponding to the height of the cube. In the point case this effect was eliminated by weighting the linear system according to the height of a 3D point (see fig. 6.4(a)). As explained above, this weighting is not applied for line reconstruction. Eventually, the algorithm fails if some 3D lines lie on the plane, i.e. the height of the cube is 0. The Line-Cam algorithm which reconstructs only the cameras and not the 3D lines performed constantly good, i.e. independent of the height of the cube. The RMS error of the Line-DRP(1p) and Line-DRP(min) algorithm increases, like the Line-DRP algorithm, corresponding to height of the cube. However, these algorithms separate lines on and off the plane by a simple thresholding, which was set in this case to 0.2. Therefore, the performance improves considerably for a certain height of the cube, i.e. 0.4. However, to find this optimal threshold is difficult. Consequently, in practice the Line-Cam algorithm is preferable for scenarios where the lines might lie on or close to the reference plane. Otherwise, we recommend the Line-DRP, Line-DRP(1p) or Line-DRP(min) method, since lines and cameras are reconstructed simultaneously.

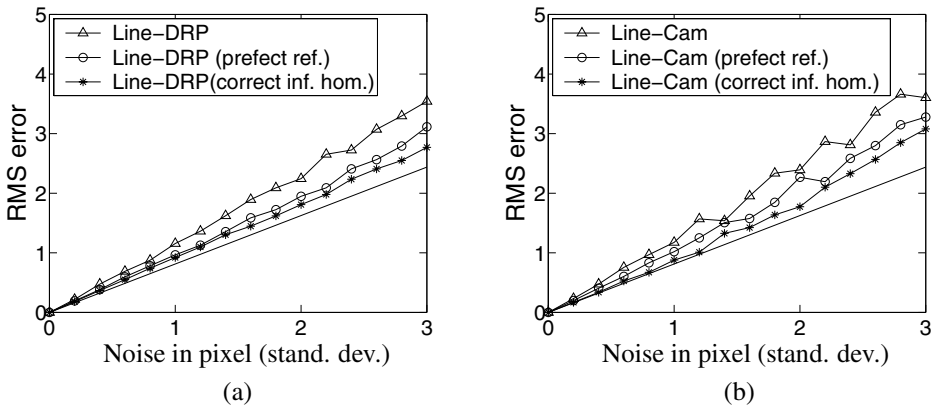


Figure 6.33. Results for the Line-DRP (a) and Line-Cam (b) algorithm with respect to different infinite homographies

Position and quality of the reference plane

As in the point case, it can be expected that the performance improves with the quality of the infinite homographies. Furthermore, it can be expected that the result of the Line-DRP and Line-Cam method is better if the plane at infinity, i.e. the reference plane, is at its correct position. The quantity of these improvements is analyzed here. Let us repeat the first experiment (see fig. 6.31(a)) for the Line-DRP and Line-Cam algorithm (fig. 6.33). The Line-DRP/Cam (perfect ref.) algorithm shows the case where Gaussian noise was added to all image points *except for* the reference points. The improvement is obvious for both algorithms. Additionally, fig. 6.33 depicts the case where both algorithms use the correct plane at infinity as the reference plane (Line-DRP/Cam (correct inf. hom.)). The performance improves slightly, compared to a perfect finite reference plane. This can be expected since for the correct plane at infinity the endpoints of the 3D lines have the same order of magnitude.

Missing data

In a last, synthetic experiment, the case of missing data is analyzed. Each 3D line is visible in a fraction of $3 - 8$ views. This fraction is taken randomly. A more realistic scenario of missing data is considered in the next section. The reference plane, i.e the 4 reference points, are visible in all views. Fig. 6.11 depicts the performance for a standard deviation of 1 (a) and 3 (b). The theoretical minimum is in this case constant since all 3D points, i.e. endpoints of the line segments, are visible in all views.

Fig. 6.11 shows that both methods, i.e. Line-DRP and Line-Cam, handle a substantial amount of missing data, i.e. up to 50%. However, they decrease in performance corresponding to the amount of missing data. This was not the case for our point based DRP method (fig. 6.11). As already mentioned, 3D lines provide fewer geometric constraints

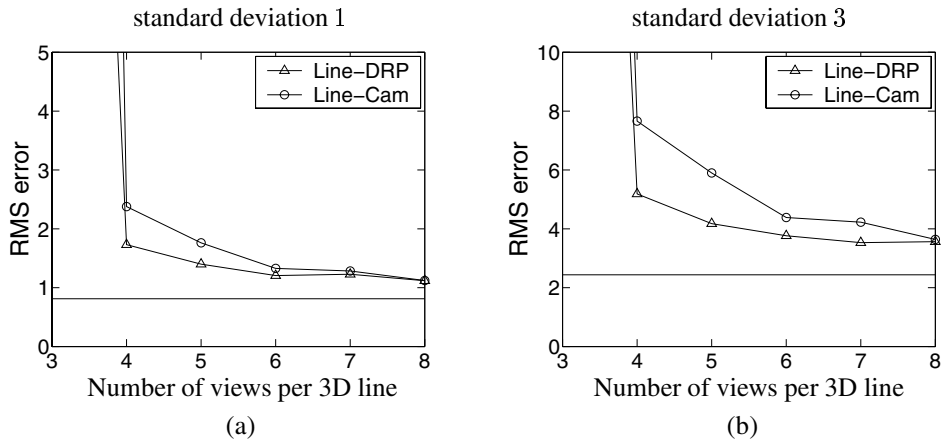


Figure 6.34. Performance of the Line-DRP and Line-Cam method for the case of missing data. The standard deviation of the Gaussian noise is 1 (a) and 3 (b).

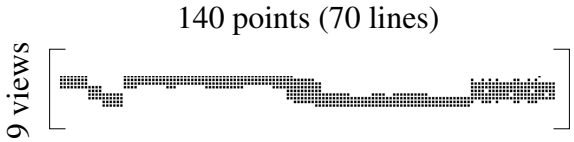


Figure 6.35. The visibility matrix of the house sequence for 70 manually selected lines.

than 3D points. For both noise levels the Line-Cam algorithm performed slightly worse than the Line-DRP method. However, it is to expect, which was, however, not shown, that both reference plane methods are superior to general line-based methods for difficult scenes with a high percentage of missing data.

6.2.3 Experiments: Real Data

In contrast to point features, only one real world experiment was conducted for line features. The goal is to demonstrate that the Line-DRP method, as well as the Line-Cam method, can be successfully applied to real world scenes. Additionally, we will compare their performance with the point based DRP and RefCam method using the endpoints of the line segments. The main conclusion will be that the point based methods are significantly superior to the line based methods. This can be expected since 3D points provide more geometric constraints than 3D lines.

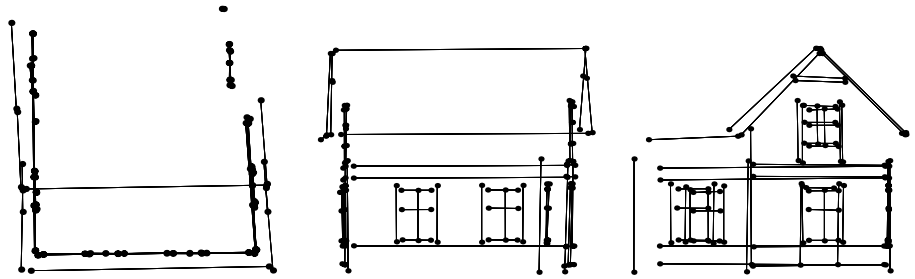


Figure 6.36. The top (a), side (b) and front (c) views of the reconstruction using the DRP algorithm and the endpoints of the line segments. The result is better than in fig. 6.37 using the Line-DRP method.

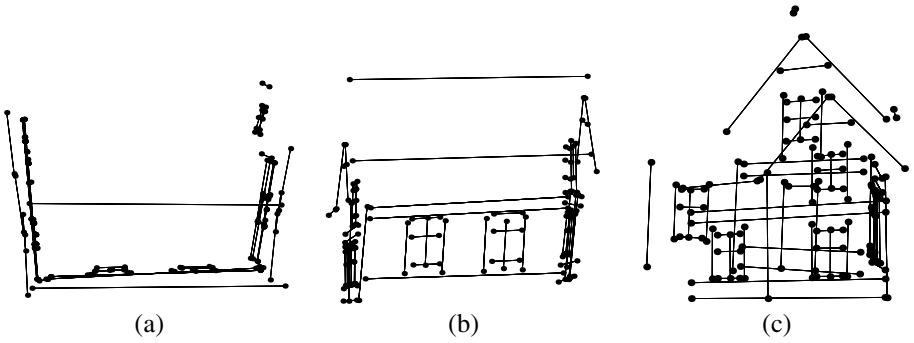


Figure 6.37. The top (a), side (b) and front (c) views of the reconstruction using the Line-DRP algorithm and no bundle adjustment. The result is significantly worse than in fig. 6.36, where the DRP method is used. However, the approximate shape of the house is preserved.

We selected manually 70 line segments in the 9 images of the house sequence (see fig. 6.27). The visibility matrix of the 140 endpoints of the line segments is depicted in fig. 6.35. The amount of missing data is 54%. The endpoints were reconstructed with our DRP algorithm (fig. 6.36). This result is qualitatively correct. Our Line-DRP method, using the corresponding line segments, gave the result in fig. 6.37 before bundle adjustment. Obviously, this result is significantly worse, however, the approximate shape of the house is preserved.

The reconstruction in fig. 6.36 was used as a *synthetic* house sequence. The performance of the different Line-DRP methods and the point based DRP and RefCam methods are depicted in fig. 6.38. The performance is analyzed in terms of the mean 3D error of the reconstructed endpoints of the 3D line segments. The point based methods are very stable.

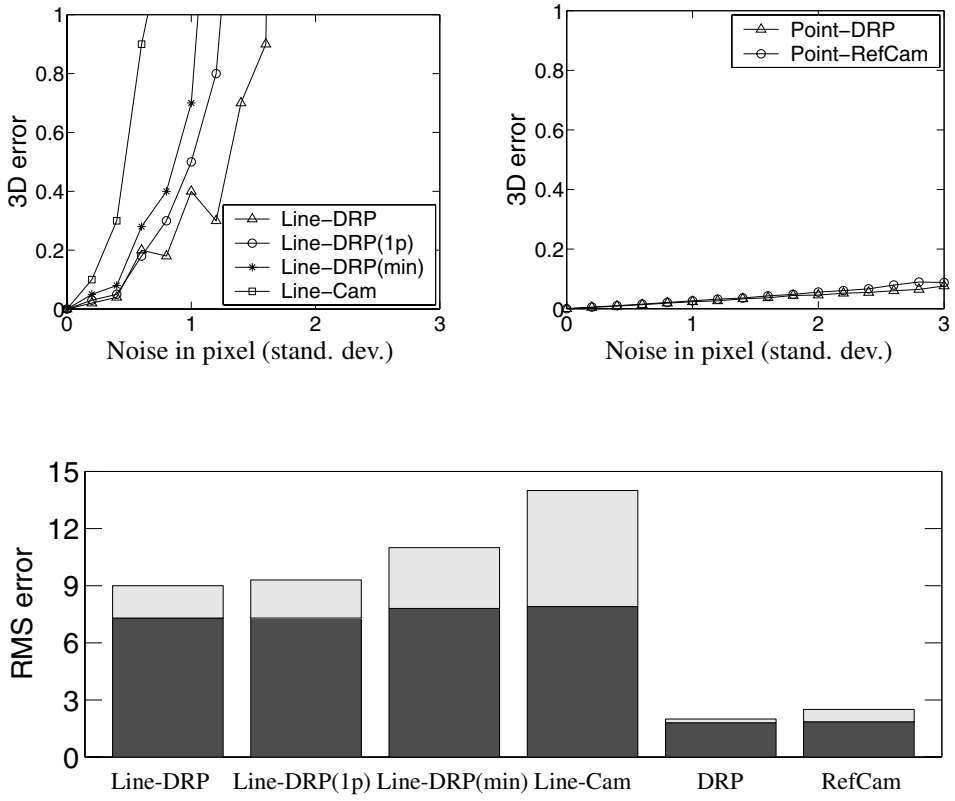


Figure 6.38. Results of various algorithms on the *synthetic* house sequence (top) and the *real* house sequence (bottom), before bundle adjustment (grey) and after bundle adjustment (black).

The line based methods can only reconstruct the house for a small noise level, $\sigma < 0.5$. As in the synthetic experiments (sec. 6.2.2), the Line-DRP method performed better than the other Line-DRP versions and, especially, the Line-Cam algorithm. The experiment on the *real* house sequence confirms the conclusion drawn from the synthetic house experiment. All line based methods were inferior to the point based methods, before and after bundle adjustment.

To summarize, for this real world scenario, with a high percentage of missing data, the Line-DRP and the Line-Cam methods are significantly inferior to our point based DRP method, however, could approximately reconstruct the scene.

6.3 Planes

The reconstruction of multiple planes observed in multiple views was discussed in sec. 3.4. We saw that there are in general two options, either use directly the homographies or "hallucinate" point and/or line correspondences. Any reconstruction algorithm (with and without a reference plane) may be applied to the hallucinate image features. Section 3.4 presented three different reference plane algorithms which use directly the given homographies. Practical versions of these three methods are formulated in sec. 6.3.1. The reason that methods based on camera constraints and factorization are also presented here is that they were published in (Rother et al., 2002).

After this presentation, synthetic and real world experiments are discussed in sections 6.3.2 and 6.3.3. The goal of the experiments is to show that the three homography-based methods perform successfully for one synthetic and one real world scenario. Furthermore, we will demonstrate that advanced methods which use hallucinated image points are slightly superior to methods which use the homographies directly. Parts of these experiments were published in (Rother et al., 2002). As for line features, more experiments have to be conducted to confirm these conclusions.

6.3.1 Outline of the DRP Method & Other Linear Methods

This section will outline our three plane-based reconstruction methods (see sections 3.4.2, 3.4.3 and 3.4.4) which are based directly on the given homographies. This means that the input data consists of a set of homographies H_{ij}^k, which represent a plane $\mathbf{\Pi}_k$ visible in the views i and j. In the following, the reference plane is considered as an additional plane which is visible in all views. The reference plane homographies, i.e. the infinite homographies, are denoted as H_i^∞.

For points and lines, the linear constraints on features and cameras were extended to use normalized image features and general cameras. These aspects are *not* discussed here, since the projection of a plane "into" an image is not an "image feature".

Plane-DRP algorithm

We saw in sec. 3.4.1 that the unknown scale λ of a homography H_{ij}^k in eqn. 3.85 may be derived directly as the double eigenvalue of H_{ij}^k. Furthermore, the normal $\mathbf{n}_k$ of a plane $\mathbf{\Pi}_k = (\mathbf{n}_k, d_k)^T$ may be computed directly from H_{ij}^k. This information is sufficient to derive eqn. 3.97, which is linear in the remaining unknown parameters, $\bar{\mathbf{Q}}_j$ and d_k. Therefore, multiple planes visible in multiple views may be reconstructed by our novel linear method **Plane-DRP** introduced in sec. 3.4.2, which consist of the following steps

1. Move the reference plane to infinity, i.e. recompute all H_{ij}^k as $H_{ij}^k = H_j^{\infty -1}\, H_{ij}^k\, H_i^\infty$.
2. Compute the normal $\mathbf{n}_k$ of plane $\mathbf{\Pi}_k = (\mathbf{n}_k, d_k)^T$ from H_{ij}^k (see below).
3. Derive λ as the double eigenvalue of H_{ij}^k and determine $\hat{H}_{ij}^k = \lambda^{-1} H_{ij}^k - I$.
4. Obtain $\bar{\mathbf{Q}}_j$ and d_k (of plane $\mathbf{\Pi}_k = (\mathbf{n}_k, d_k)^T$) by SVD using eqn. 3.97 and $\hat{H}_{ij}^k$.

As in the point (and line) case it is more efficient to compute only the matrix V, i.e. its last four singular vectors, from the SVD of the system matrix $S = UDV^T$. In order to compute reliably the normal $\mathbf{n}_k$ of a plane $\mathbf{\Pi}_k$, we suggest the following method

1. For each pair of views i and j, where H_{ij}^k exist, repeat N times.
   2. Hallucinate 3 points $\mathbf{x}_{1i}, \mathbf{x}_{2i}$ and $\mathbf{x}_{3i}$ inside the "homography area" of image i.
   3. Derive the 3 corresponding points in image j as $\mathbf{x}_{lj} = H_{ij}^k \mathbf{x}_{li}$.
   4. Stabilize the image points in image i: $\mathbf{x}_{li} = H_i^{\infty -1} \mathbf{x}_{li}$ and j: $\mathbf{x}_{lj} = H_j^{\infty -1} \mathbf{x}_{lj}$.
   5. Compute a normal $\mathbf{n}$ using eqn. 3.94 and 3.95.
6. Take $\mathbf{n}_k$ as the average normal from the set of all normals $\mathbf{n}$.

Plane-Cam algorithm

In sec. 3.4.3, a novel linear algorithm (Rother et al., 2002) was presented, which reconstructs *all* camera centres simultaneously, from camera constraints involving homographies only. On the basis of the known cameras, the planes may be derived. The main advantage of this method, in contrast to the Plan-DRP method, is that the normals of the planes are not needed. This method is composed of the following steps:

1. Move the reference plane to infinity, i.e. recompute all H_{ij}^k as $H_{ij}^k = H_j^{\infty -1} \; H_{ij}^k \; H_i^{\infty}$.
2. Derive λ as the double eigenvalue of H_{ij}^k and determine $\hat{H}_{ij}^k = \lambda^{-1} H_{ij}^k - I$.
3. Obtain camera centres $\bar{\mathbf{Q}}_j$ by SVD using eqn. 3.102 and $\hat{H}_{ij}^k$.
4. Compute $\mathbf{\Pi}_k = (\mathbf{n}_k, d_k)^T$ from eqn. 3.101 and λ_{ij}^k (eqn. 3.100).
5. Optionally, iterate steps 1 – 4 (using λ_{ij}^k) until cameras and/or planes are unchanged.

In the following, this method without iteration, i.e. without step 5, is denoted the **Plane-Cam** method. The iterative version is called the **Plane-CamIt** method.

Plane-Fac algorithm

Finally, a simple factorization method for multiple planes and cameras, denoted the **Plane-Fac** method, was introduced in sec. 3.4.4. This method was presented in (Triggs, 2000; Rother et al., 2002). The main difference of this method, in contrast to the Plane-DRP and Plane-Cam method, is that one view is considered as the reference view. Consequently, only those homographies which include the reference view are used. This is a drawback since not all given information is exploited. It was shown, that with this assumption a plane may be expressed as $\mathbf{\Pi}_k = (\mathbf{n}_k, 1)^T$. This leads to the following factorization method:

1. Move the reference plane to infinity, i.e. recompute all H_{ij}^k as $H_{ij}^k = H_j^{\infty -1} \; H_{ij}^k \; H_i^{\infty}$.
2. Derive λ as the double eigenvalue of H_{ij}^k and determine $\hat{H}_{ij}^k = \lambda^{-1} H_{ij}^k - I$.
3. Obtain $\bar{\mathbf{Q}}_j$ and $\mathbf{n}_k$ ($\mathbf{\Pi}_k = (\mathbf{n}_k, 1)^T$) by SVD of the measurement matrix in eqn. 3.108.

In contrast to the Plane-DRP and Plane-Cam method, a full SVD of the measurement matrix has to be carried out since both U and V of the SVD of $W = UDV^T$ are needed.

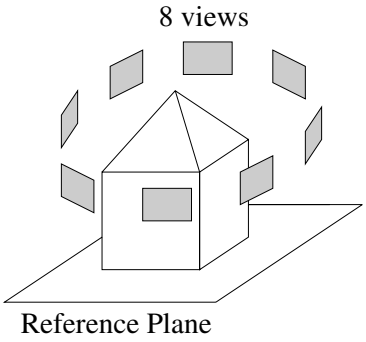


Figure 6.39. The synthetic configuration comprises of 8 images of a house which consists of 9 planes. The ground plane, which is visible in all views, served as the reference plane.

6.3.2 Experiments: Synthetic Data

This experimental section comprises *not* of extensive experiments as in the point case. We will compare the performance of homography-based methods, introduced in sec. 6.3.1, with methods using hallucinated image points for one synthetic scenario. The synthetic experiments for planes were conducted in the same fashion as for point and line features. A 8 frame synthetic sequence was generated based on the scene depicted in fig. 6.39, which consists of 9 planes forming a house. The homographies between the views were computed from point matches, where each plane has on average 20 points. The size of the ground plane (reference plane) is 12×12 units and the height of the house is 6 units. As in the line case, it is difficult to compare a 3D plane reconstruction with the "ground truth". Therefore, the 3D points which lie on the planes were determined by intersection using the reconstructed cameras. Afterwards, the reconstruction and the ground truth were aligned in an optimal way. The mean 3D error between the point reconstruction and the ground truth served as the quantitive measurement of the performance.

In the following, three reconstruction methods which use directly the homographies are compared: **Plane-DRP, Plane-Cam(Plane-CamIt)** and **Plane-Fac**. As mentioned above, a planar scene may as well be reconstructed by any point-based reconstruction algorithm using hallucinated point features. Therefore, 20 points per plane were hallucinated in all views using the estimated homographies. This set of hallucinated points is then reconstructed with the general methods **Fmat, FmatBa** and the reference plane methods **DRP, RefCam**.

In a first experiment, the assumption was made that all planes are visible in all views, i.e. all planes are transparent. Fig. 6.40 shows the performance of various algorithms based on the homographies directly (a) or hallucinated image points (b). The performance of the Plane-DRP, Plane-Cam and Plane-CamIt method are virtually identical. This shows that iterating the solution with the Plane-CamIt method did not improve the result considerably. The Plane-Fac algorithm performed slightly worse for smaller noise levels, i.e. $\sigma < 2$, and

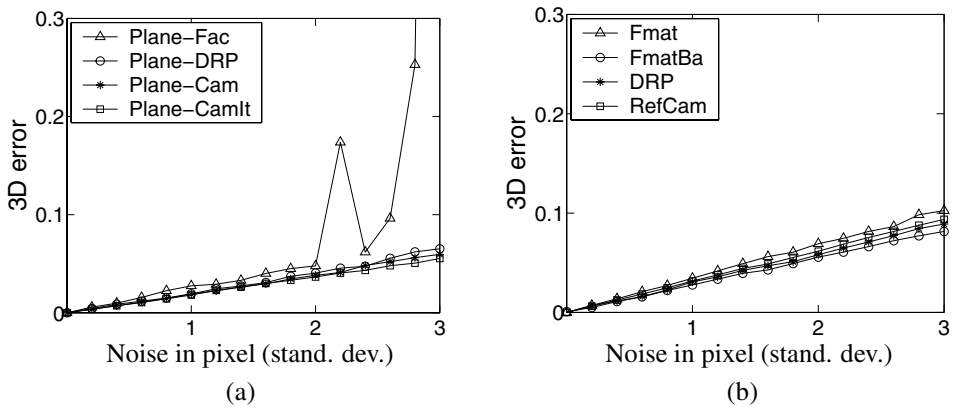


Figure 6.40. Results for the case of *no missing data*. The algorithms in (a) are based on homographies directly and in (b) on hallucinated image points.

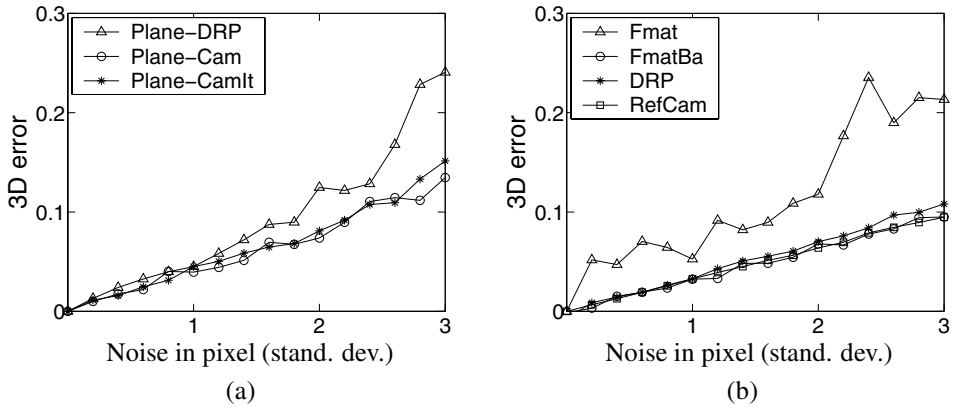


Figure 6.41. Results for the case of *missing data*. The algorithms in (a) are based on homographies directly and in (b) on hallucinated image points.

significantly worse otherwise. A plausible explanation is that this method uses only a part of the available information, i.e. only the reference view homographies. The methods based on hallucinated point features, e.g. DRP, RefCam, performed, in contrast to the direct homography methods, slightly worse (for $\sigma = 3$ is the 3D error 1 with the DRP method and 0.6 with the Plane-DRP method).

Let us repeat the first experiment with the more realistic assumption of non-transparent planes of the synthetic house (fig. 6.39). This means that each plane, except of the ground plane (reference plane), is visible in only 3 successive views. Fig. 6.41 depicts the performance of the different algorithms based on the homographies directly (a) or hallucinated

image points (b). In this case, the Plane-Cam (and Plane-CamIt) method performed slightly better than the Plane-DRP methods. The difference between these two methods is that the Plane-DRP method derives and uses the normals of the planes. This is a potential source of error. The Plane-Cam method does not rely on the planes' normal. If we compare the "homographies methods", i.e. Plane-DRP and Plane-Cam, with the "hallucinating point methods", i.e. FmatBa, DRP and RefCam, it turns out that the "hallucinating point methods" performed more stably. The bundle adjustment process of the FmatBa method was necessary to improve the performance of the Fmat method.

6.3.3 Experiments: Real Data

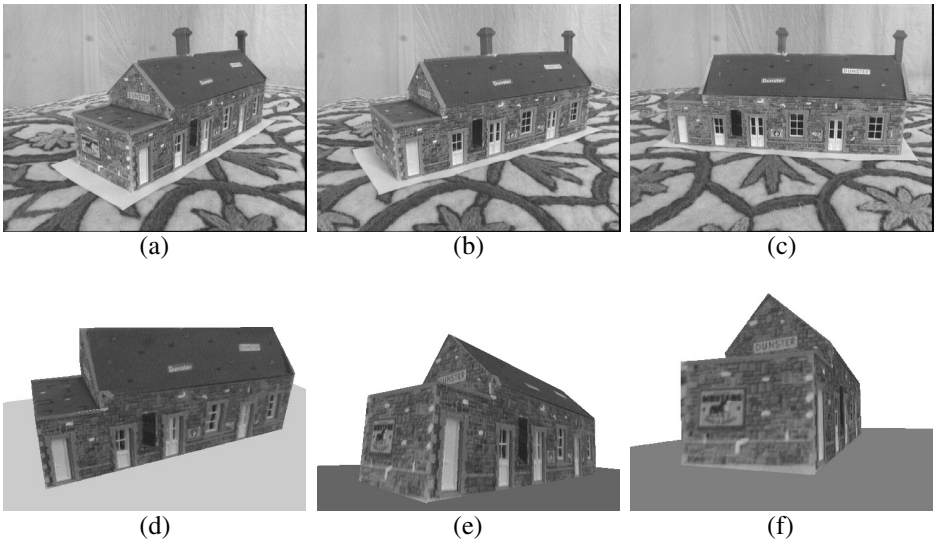


Figure 6.42. Three of the eight original views of the toyhouse sequence. The corresponding camera positions are labeled in fig. 6.43. Three novel views of a VRML model of the toyhouse (d-f).

As in the line case, only one real world experiments using planes was conducted. The goal is to demonstrate that both approaches, i.e. "direct homography methods" and "hallucinating points methods", perform successfully under real world conditions. As real image data, a toyhouse sequence[3] shown in fig. 6.42(a-c) was used. The frontal part of the toyhouse consists of 6 planes. The ground plane, which served as the reference plane, the roof and the front side of the house are visible in all views. The other 3 planes are only visible in the first four views. The homographies were determined on the basis of manually selected point matches. Fig. 6.43 shows the top view of the reconstruction of 39 model points and

[3]The toyhouse sequence is available at http://www.robots.ox.ac.uk/vgg/data/

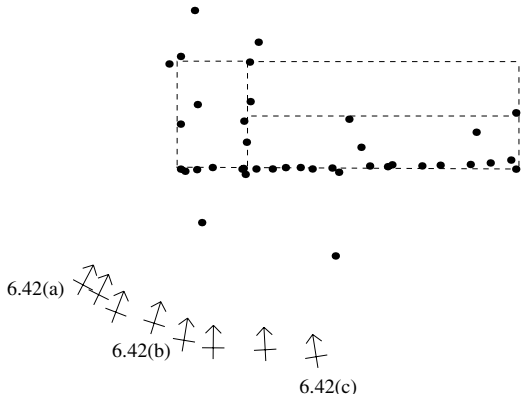


Figure 6.43. Top view of the reconstruction of the toyhouse sequence with 39 model points (dots) and 8 cameras (arrows). The dashed lines indicate the model planes. The reconstructed model points were not constrained to lie on the planes. The labeled cameras correspond to images in fig. 6.42.

8 cameras using the Plane-DRP method. The planes of the model are indicated by dashed lines. In this case the reconstructed model points were not constrained to lie on the planes. Fig. 6.43 shows that the baseline of the cameras is fairly small for this scenario, in contrast to all previous real world experiments, e.g. fig. 6.23. This potentially decreases the accuracy in the reconstruction for noisy image measurements. As in previous real world experiments, the toyhouse sequence was synthesized. In order to meet the requirement of a planar scene, all reconstructed model points were projected onto the corresponding model planes. Furthermore, the ground truth reconstruction was scaled so that the front side of the toyhouse had the same size as the synthetic house in fig. 6.39

Fig. 6.44(top) shows the performance of various algorithms for the synthetic toyhouse sequence. A first observation is that all algorithms performed worse (in terms of mean 3D error) compared to the synthetic house sequence in fig. 6.41. This is most likely due to the small baseline of the cameras for the toyhouse sequence. In this experiment the "hallucinating point methods", i.e. FmatBa, DRP and RefCam, performed considerably better than the "homography methods", i.e. Plane-DRP, Plane-Cam and Plane-CamIt. Especially for small noise levels, i.e. $\sigma < 1$. As in the previous experiment, the bundle adjustment process of the FmatBa method was necessary to improve considerably the performance of the Fmat method. Fig. 6.44(bottom) depicts the results of the different algorithms for the real toyhouse sequence in terms of the RMS error. The results confirm most of the conclusions drawn from the synthetic toyhouse experiment. The "hallucinating point methods" are superior to the "homographies methods". The performance before bundle adjustment was 13.1 for the Plane-DRP method and 20.8 for the Plane-Cam (and Plane-CamIt) method. However, *all* methods computed a 3D reconstruction which was sufficiently good in order to perform successfully a bundle adjustment process, which gave a result with a low

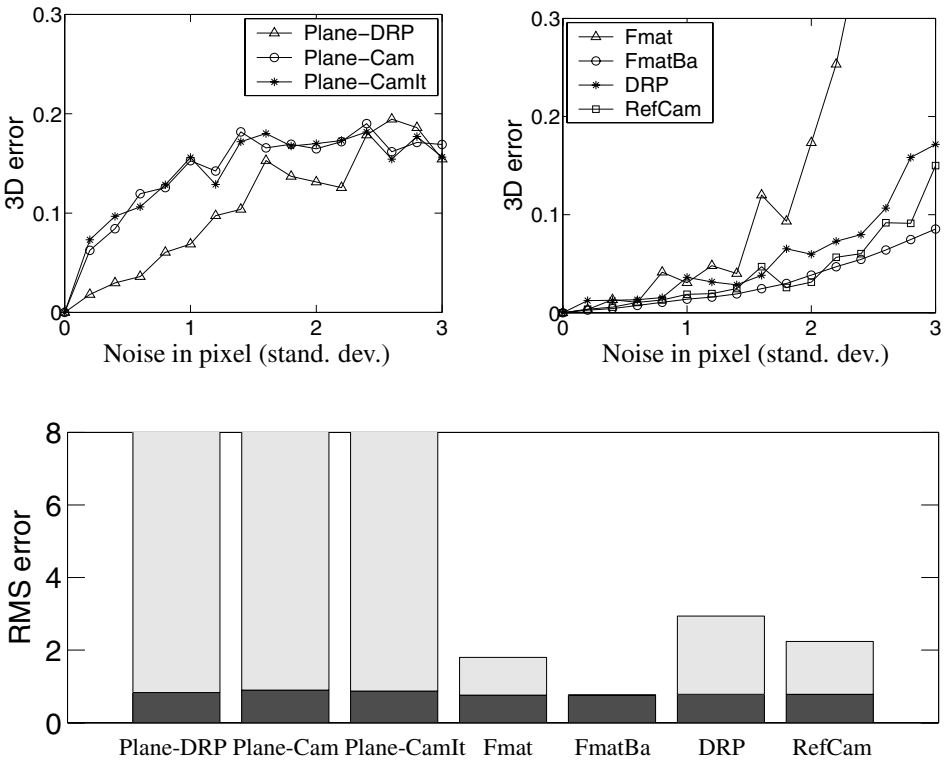


Figure 6.44. Results of various algorithms on the *synthetic* toyhouse sequence (top) and the *real* teapot sequence (bottom), before bundle adjustment (grey) and after bundle adjustment (black).

RMS error of 0.8. In this case, the Fmat method without bundle adjustment performed unexpectedly good.

Let us summarize the experiments using real (and synthetic) data. What is the best reconstruction method for planar scenes? To answer this question profoundly, more experiments must be conducted. On the basis of the conducted experiments it seems that methods which use hallucinated points, e.g. DRP, RefCam or FmatBa, are slightly superior to methods which use the homographies directly, i.e. Plane-DRP or Plane-Cam. The results of the "homographies methods" were only very stable for the "simple" scenario of wide baseline images and no missing data. However, two aspects about "hallucinating point methods" have to be kept in mind. First, as pointed out in (Szeliski and Torr, 1998), it is important that points are hallucinated inside the image area from which the respective homography was derived. Secondly, only the advanced hallucinating point methods, i.e. not Fmat, performed good in all experiments. The Plane-Fac method is less recommendable for practical usage since it specializes one reference view and the problem of missing data must be solved.

6.4 Conclusions and Future Work

This chapter presented practical algorithms for our direct reference plane (DRP) methods for points, lines and planes based on chapter 3. The main contribution is the experimental comparison of these methods with other approaches under various conditions using real and synthetic data. Note that reference plane methods may apply any real or virtual reference plane (chapter 5). A virtual reference plane may represent a finite plane or the correct plane at infinity.

For *point features*, our DRP method was formulated as in (Rother and Carlsson, 2002b; Rother and Carlsson, 2002a). Furthermore, it has been extended to use normalized image data and general cameras. The most important part of this chapter is the extensive comparative study on real image data. Therefore, real image data was "synthesized", i.e. a quantitatively correct reconstruction was taken as "ground truth". *The main conclusion is that for difficult, reference plane scenarios with a high percentage of missing data, up to* 90%*, our DRP method and Hartley et al.'s (2001) reference plane method performed successfully, where general reconstruction methods fail.* In particular, we analyzed the general reconstruction methods suggested by Fitzgibbon and Zisserman (1998), Beardsley et al. (1996), Sturm and Triggs (1996), and Martinec and Pajdla (2002) and the "affine" reconstruction methods of Tomasi and Kanade (1992), Jacobs (1997) and Kahl and Heyden (1999). The "failure" of these methods had the following reasons: (a) too few image measurements are available, (b) error accumulation due to noisy image measurements or (c) critical configurations (dominant scene plane). The reference plane methods circumvent all these problems since they exploit a real or virtual reference plane visible in all views, in order to reconstruct the cameras (and the structure) simultaneously. Our DRP method was considerably superior to Hartley et al.'s (2001) method for three out of five real world experiments. However, both methods converged to the same local minimum after bundle adjustment. The "tape holder" sequence demonstrated that *reference plane methods are inferior to general methods if the reference plane is detected very inaccurately*.

The synthetic experiments compared various versions of our method with two other reference plane methods, the camera constraint method of Hartley et al. (2001) and the factorization method of Triggs (2000). We may conclude that for scenes where 3D points are *not close* to the reference plane, our simple, non-iterative DRP method and the camera constraint method were very stable. The results are virtually optimal when the reference plane is the correct plane at infinity. For "flat scenes" with many 3D points on or close to the reference plane, our and the factorization method performed best. However, in this case our method is complex and iterative, depending on the number of 3D points. The factorization method has the drawback of "hallucinating" missing data and is *not* applicable for infinite reference planes. The camera constraint method performed unexpectedly unstably for "flat scenes". Consequently, in our opinion there is still no satisfactory method for "flat scenes". Additionally, we applied our DRP method to general scenes by assuming affine cameras (sec. 5.2.2) or known epipolar geometry (sec. 5.2.3). It turned out that our method is very sensitive to the estimated epipoles and noise in the reference points. Consequently, this approach is potentially inferior to general reconstruction methods which do not distinguish reference points.

The second part of this chapter outlined three versions of our DRP method for *line features*. These algorithms are novel and were not presented in our previous publications. The main goal of the experimental study was to demonstrate that these methods perform "successfully" under various conditions using real and synthetic data. In contrast to point features, a smaller number of experiments were conducted. For comparison, an extension of Hartley et al.'s (2001) point based method for line features was analyzed, which is based on camera constraints (sec. 3.3.3). It turned out that our method is slightly superior to the camera constraint method for scenes were 3D lines are *not close* to the reference plane. Otherwise, the camera constraint method was slightly better, since our method does not weight 3D lines according to their distance to the plane. Both methods can handle a substantial amount of missing data, i.e. up to 50%. However, for a real world scenario, with a high percentage of missing data, both methods are significantly inferior to our point based DRP method. This can be expected since 3D lines provide fewer geometric constraints than 3D points.

Finally, for *scene planes* we outlined our novel DRP algorithm, our linear, camera-constraint method (Rother et al., 2002), and a factorization method (Triggs, 2000; Rother et al., 2002). These three methods *directly* apply the homographies induced by scene planes. As discussed in sec. 3.4, image points (or lines) may be hallucinated using the homographies. Consequently, any point-based reconstruction method can reconstruct scene planes based on hallucinated image points. The primary goal of the experimental study was to demonstrate that the three "direct homographies methods" perform successfully under various conditions using real and synthetic data. Apart from the factorization method, the performance was very stable for simple scenarios of wide baseline images and no missing data. The factorization method distinguishes one reference view which affected the results negatively. A comparative study with "hallucinating point methods" showed that they were slightly superior for difficult scenes with missing data. However, this was only true for more advanced point-based methods like our DRP method. As for line features, more experiments have to be conducted to confirm these conclusions.

In future work we plan to study experimentally our DRP method for combinations of points, lines and planes (sec. 3.5.1). Additionally, those scene constraints which provide linear equations can be included in our DRP method (sec. 3.5.2).

Chapter 7

Critical Reference Plane Configurations

The previous chapter presented many examples where our direct reference plane approach was applied to reconstruct reference plane scenarios. In the absence of noise, our method returned for all examples the correct solution. This chapter investigates configurations of multiple cameras, multiple 3D points and a known reference plane which do *not* have a unique projective reconstruction. This set of configurations is called *critical reference plane configurations*. A necessary condition is that enough 3D points are visible in multiple views. This leads to the questions of *sufficient visibility* which is also addressed here. We restrict this investigation to point features. All results are novel, based on our publication (Rother and Carlsson, 2002a), since to our knowledge critical configurations have only been studied for the general, non-reference plane, case (e.g. Kahl et al., 2001). Since for reference plane configurations the relationship between 3D points and cameras is linear, this analysis is simpler compared to the general case.

We begin the discussion with the case of no missing data, i.e. all 3D points are visible in all views (sec. 7.2). The main observation is that for multiple views all non-trivial configurations where points and camera centres are non-coplanar are non-critical. Therefore, the typical scenario of one dominant scene plane visible in multiple views is not critical if the reference plane is different to the scene plane. This is an important practical result since this scenario is critical in the absence of a known reference plane.

For the case of missing data (sec. 7.3) we will introduced a method to construct non-critical configurations.

7.1 Introduction

In the following we investigate the constraints that 3D points and cameras have to satisfy in order to obtain a unique projective reconstruction for the case of having a known reference plane visible in all views. Note that this is equivalent to the assumption of having 4 points

in the scene which are known to be coplanar. Throughout this chapter we assume that no camera centre lies on the reference plane, i.e. all cameras are finite. This means that all infinite homographies are non-singular.

We saw in the previous chapter that in all real world examples many 3D points were only visible in a limited number of views. In order to specify a certain overlap between points and views we introduced the **visibility matrix** V. An element $V(i,j)$ of the visibility matrix is set if the jth point is visible in the ith view. The following example shows a specific visibility matrix of $n = 5$ points partly visible in $m = 3$ views

$$V = \begin{array}{c} \\ \text{views} \end{array} \begin{array}{c} \begin{array}{ccccc} \multicolumn{5}{c}{\text{points}} \\ 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{|ccccc|} \hline \bullet & \bullet & & & \\ \bullet & \bullet & \bullet & \bullet & \bullet \\ & & \bullet & \bullet & \bullet \\ \hline \end{array} \end{array} \begin{array}{c} 1 \\ 2 \\ 3 \end{array} . \tag{7.1}$$

Note that the visibility matrix does only specify which point is visible in which view. However, it does not specify the actual placement of points and camera centres in the scene. Therefore, we denote a specific placement of points and cameras in the scene as a **configuration**. We saw in sec. 3.1 that a configuration is only unique up to a projective transformation (see eqn. 3.2). The configurations $(\mathbf{X}_i, P_j)$ and $(H\mathbf{X}_i, P_j H^{-1})$ which are related by a projective transformation H are denoted as equal. Let us define the notion of sufficient visibility given a certain visibility matrix.

Definition 5 (Sufficient Visibility) *A visibility matrix is called sufficient if there is at least one configuration which has a unique reconstruction.*

Sufficient visibility does not necessarily imply a unique reconstruction. Therefore, a configuration is called a **critical configuration** if the visibility matrix is sufficient but the projected image points are insufficient to determine a unique reconstruction (see definition 4 in sec. 3.1).

Let us consider the questions of sufficient visibility and critical configurations for the case of n 3D points and m views with the additional assumption of having a known reference plane visible in all views. Consider the case that non of the points lie on the reference plane. We saw in sec. 3.2.2 that the total number of degrees of freedom ($\#dofs$) of the the linear system in eqn. 3.17 is: $\#dofs = 3(m+n) - 4$. Consider the rank of the S-matrix (eqn. 3.17). This is at the most $\#dofs$. If the rank of the S-matrix is smaller than $\#dofs$, the dimensionality of the nullspace is larger than four which means that the reconstruction is not unique. We can state: *A given visibility matrix is sufficient if the rank of the S-matrix is equal to the number of dofs, for a generic set of points and camera centres, i.e. points and camera centres in "general pose".* Furthermore, we can state: *A given configuration is critical if the rank of the S-matrix is smaller than the number of dofs for this configuration.* The question of critical configurations is not only of theoretical interest, however, from a numerical point of view we should expect instabilities whenever the S-matrix comes close to rank deficiency, i.e. whenever a configuration is close to a critical one.

In the past the two questions of sufficient visibility and critical configurations were investigated for two different cases: *no missing data and missing data.* This corresponds to two different types of visibility matrices: full and not full visibility matrix. We will discuss these two types of visibility matrices separately as well.

Before addressing these questions, let us recapitulate the constraints on points and cameras discussed in sec. 3.2. We saw that 3D points on the reference plane can be reconstructed directly and independently of the position of all cameras with eqn. 3.14. However, in order to reconstruct all cameras outside the reference plane, a minimum number of points outside the reference plane are necessary.

Proposition 4 *A configuration with n points and m views is critical if $n-1$ or n points lie on the reference plane.*

Proof W.l.o.g. we choose the reference plane as the plane at infinity. This means that the projection relations of any point on reference plane do not constrain the position of the respective camera centre. However, one point is not sufficient to reconstruct all cameras and this point, since $\#equations = 2m < 3m - 1 = \#dofs$ for $m > 1$.

□

7.2 No Missing Data – Full Visibility Matrix

The problem of critical configurations for general, non-reference plane, configurations has received considerable interest in computer vision and photogrammetry in the past (Maybank, 1992; Hartley and Debunne, 1998; Hartley, 2000; Kahl et al., 2001; Kahl and Hartley, 2002). The classical case of 2-view critical configurations implies that the two camera centres and all 3D points are located on a ruled quadric (Krames, 1942). From the duality of camera centres and space points (Carlsson, 1995) follows that this applies also for 6 points and any number of cameras (Hartley and Debunne, 1998). The case of three cameras and an arbitrary number of points was investigated in (Hartley, 2000). It was shown that the intersection of two distinct quadrics, which is a fourth-degree curve, is critical. Recently, Kahl et al. (2001) investigated the multi-view case. It turned out that a curve which is critical for three views remains critical for any number of views. These investigations were carried out for the general projective case, i.e. uncalibrated cameras. Critical configurations for calibrated cameras were investigated in (Maybank, 1992; Kahl and Hartley, 2002).

The non-linearity of the general projective case means that critical configurations generally imply a finite number of multiple solutions given projected image data. Having a known reference plane on the other hand, gives us a linear reconstruction problem and therefore either a unique solution or an infinite number of solutions. The case of an infinite number of solutions will occur when the S-matrix (eqn. 3.17) becomes rank deficient so that the dimensionality of the nullspace increases.

We will prove that the only critical configurations for 2 points not on the reference plane visible in 2 views are if the camera centres and the points are coplanar. This is not a contradiction to the general case, without a known reference plane, since less information

is given in this case. Note that two planes (the reference plane and the plane containing all the camera centres and points) describe a ruled quadric and is also critical in the general case. For the multi-view case we will prove that a configuration is non-critical if (a) the points and the camera centres are non-coplanar and (b) all camera centres and one of the points are non-collinear and (c) all points and one of the camera centres are non-collinear and (d) at least 2 points do not lie on the reference plane. We saw already in sec. 3.2.2 that for the case of no missing data a minimum requirement is to have 2 points outside the reference plane. Furthermore, we will prove formally that a full visibility matrix with 2 or more points and 2 or more views is sufficient.

7.2.1 Two-View Configurations

Let us consider the case of 2 points not on the reference plane visible in 2 views. From the projection relations in eqn. 3.13, which are valid for cameras outside the reference plane, we obtain at the most 8 linearly independent constraints for the S-matrix. Note that only 2 of the 3 projection relations are linear independent. If all 8 equations are linearly independent we would get a unique reconstruction, since the number of dofs is 8. We will now prove that only a limited set of configurations is critical.

Theorem 8 *A configuration of* 2 *points not on the reference plane visible in* 2 *views is critical if and only if the points and the camera centres are coplanar.*

Proof First, all points on the reference plane are detected and reconstructed independently of the cameras' position. Since the S-matrix has a four dimensional nullspace, we are free to choose either a space point or a camera centre as the origin, i.e. $(0,0,0,1)$, of the projective space. The S-matrix (eqn. 3.17) then takes on either of the forms:

$$\begin{pmatrix} S_{21} & -S_{21} & 0 \\ S_{22} & 0 & -S_{22} \\ 0 & S_{11} & 0 \\ 0 & 0 & S_{12} \end{pmatrix} \begin{pmatrix} \bar{X}_2 \\ \bar{Y}_2 \\ \bar{Z}_2 \\ \\ \bar{A}_1 \\ \bar{B}_1 \\ \bar{C}_1 \\ \\ \bar{A}_2 \\ \bar{B}_2 \\ \bar{C}_2 \end{pmatrix} = 0\,, \quad \begin{pmatrix} S_{12} & 0 & -S_{12} \\ 0 & S_{22} & -S_{22} \\ S_{11} & 0 & 0 \\ 0 & S_{21} & 0 \end{pmatrix} \begin{pmatrix} \bar{X}_1 \\ \bar{Y}_1 \\ \bar{Z}_1 \\ \\ \bar{X}_2 \\ \bar{Y}_2 \\ \bar{Z}_2 \\ \\ \bar{A}_2 \\ \bar{B}_2 \\ \bar{C}_2 \end{pmatrix} = 0 \tag{7.2}$$

where

$$S_{ij} = \begin{pmatrix} 0 & w_{ij} & -y_{ij} \\ -w_{ij} & 0 & x_{ij} \\ y_{ij} & -x_{ij} & 0 \end{pmatrix} \tag{7.3}$$

are 3×3 matrices built up from image coordinates of point i visible in view j.

In case of a non-critical configuration these matrices are of rank 8 which means that the null vector is unique up to scale. If the matrices were of rank 7 or less, the dimension of the nullspaces would be larger than one and the null vector no longer unique up to scale. Rank deficiency of a matrix is generally checked by computing the singular values. In our case, however, we are interested in the algebraic conditions on the elements of the matrix for it to be rank deficient. Rank deficiency, i.e. a rank less than 7, of the S-matrix implies that the determinants of all 8×8 sub-matrices of the S-matrix are zero.

These subdeterminants were computed using MAPLE and it was found that all subdeterminants, which are not generically zero, have a simple common structure. By reordering rows and columns it can be shown that the two cases in eqn. 7.2 are completely equivalent by the choice of the origin. Therefore, all computations were made for the case of choosing the first camera as the origin, i.e. $\bar{A}_1 = \bar{B}_1 = \bar{C}_1 = 0$. The elements in the S_{ij} matrix can be expressed in terms of coordinates of space points $\bar{\mathbf{X}}_1, \bar{\mathbf{X}}_2$ and coordinates of the second camera centre $\bar{\mathbf{Q}}_2$. Explicitly, it is $x_{ij} = \bar{X}_i - \bar{A}_j$, $y_{ij} = \bar{Y}_i - \bar{B}_j$ and $z_{ij} = \bar{Z}_i - \bar{C}_j$. It was found that all 8×8 subdeterminants could be factored into:
A) The determinant:

$$det(\bar{\mathbf{X}}_1 \; \bar{\mathbf{X}}_2 \; \bar{\mathbf{Q}}_2) \tag{7.4}$$

B) A factor computed by selecting one coordinate element from five vectors in three different ways:

$$\begin{array}{lccccc} 1. & (\bar{\mathbf{X}}_2 - \bar{\mathbf{Q}}_2) & (\bar{\mathbf{X}}_1 - \bar{\mathbf{Q}}_2) & \bar{\mathbf{X}}_1 & \bar{\mathbf{X}}_2 & \bar{\mathbf{Q}}_2 \\ 2. & (\bar{\mathbf{X}}_2 - \bar{\mathbf{Q}}_2) & (\bar{\mathbf{X}}_1 - \bar{\mathbf{Q}}_2) & \bar{\mathbf{X}}_1 & \bar{\mathbf{X}}_2 & \bar{\mathbf{X}}_1 \\ 3. & (\bar{\mathbf{X}}_2 - \bar{\mathbf{Q}}_2) & (\bar{\mathbf{X}}_1 - \bar{\mathbf{Q}}_2) & \bar{\mathbf{X}}_1 & \bar{\mathbf{X}}_2 & \bar{\mathbf{X}}_2 \end{array} \; . \tag{7.5}$$

This factor is then computed by multiplying these five elements together, e.g.

$$(\bar{X}_2 - \bar{A}_2)\,(\bar{Y}_1 - \bar{B}_2)\,\bar{X}_1\,\bar{Z}_2\,\bar{A}_1 \quad . \tag{7.6}$$

Rank deficiency of the S-matrix, implying that all subdeterminants are zero, will occur if either the A factor or the B factor is zero for all combinatorial choices. Obviously rank deficiency will occur if:

$$det(\bar{\mathbf{X}}_1 \; \bar{\mathbf{X}}_2 \; \bar{\mathbf{Q}}_2) \; = \; 0 \tag{7.7}$$

which means that points P_1, P_2 and Q_2 are coplanar with the origin, i.e. point Q_1.

We will now show that all rank deficient configurations are described by this coplanarity condition. Suppose this condition is not fulfilled, i.e.

$$det(\bar{\mathbf{X}}_1 \; \bar{\mathbf{X}}_2 \; \bar{\mathbf{Q}}_2) \; \neq \; 0 \quad . \tag{7.8}$$

This means that the B factor has to be zero for every determinant. This in turn implies that at least one of the conditions:

$$\bar{\mathbf{X}}_2 - \bar{\mathbf{Q}}_2 = 0, \quad \bar{\mathbf{X}}_1 - \bar{\mathbf{Q}}_2 = 0, \quad \bar{\mathbf{X}}_1 = 0 \text{ or } \bar{\mathbf{X}}_2 = 0 \tag{7.9}$$

has to be fulfilled. Let us assume that this is *not* the case. Consider the determinants which were constructed as in the second and third way (eqn. 7.5). For such a determinant

there is at least one element of each vector which is non-zero. If we select these very elements for the computation of the B factor we obtain a non-zero B factor after multiplying all those elements. Since the A factor was assumed to be non-zero we would obtain a subdeterminant which is non-zero and therefore an S-matrix which is not rank deficient. Therefore, at least one of the four conditions in eqn. 7.9 has to be fulfilled. Since these conditions imply coincidence of points and cameras they all imply coplanarity of the four points $\bar{\mathbf{X}}_1, \bar{\mathbf{X}}_2, \bar{\mathbf{Q}}_2, \bar{\mathbf{Q}}_1 = 0$, i.e. $det(\bar{\mathbf{X}}_1\ \bar{\mathbf{X}}_2\ \bar{\mathbf{Q}}_2) = 0$. This concludes the proof that all rank deficient configurations are given by the coplanarity of the two points $\bar{\mathbf{X}}_1, \bar{\mathbf{X}}_2$ and camera centres $\bar{\mathbf{Q}}_1, \bar{\mathbf{Q}}_2$.

□

We are now able to answer the question of sufficient visibility.

Corollary 2 *A visibility matrix containing* 2 *points not on the reference plane visible in* 2 *views is sufficient and minimal.*

Proof 2 points visible in 2 views is obviously sufficient. All configurations where $\bar{\mathbf{X}}_1, \bar{\mathbf{X}}_2, \bar{\mathbf{Q}}_1$ and $\bar{\mathbf{Q}}_2$ are not coplanar give a unique reconstruction.

Furthermore, we have to prove that this visibility matrix is minimal. Let us assume that not all points are visible in all views. This means that we obtain: $\#equations < 8 = \#dofs$. If we assume that only one view is available, we obtain $\#equations = 2n < 3n - 1 = \#dofs$ for $n > 1$. However, one point visible in one camera cannot be reconstructed. The case of one point is dual to the case of one view. This concludes the proof.

□

7.2.2 Multi-View Configurations

For the case of n points visible in all m views the S-matrix has (at the most) $2mn$ linear independent equations and $3(n+m)-4$ dofs. This means that S is over-constrained, if it is not rank deficient. Let us investigate the critical configurations for such a case.

Theorem 9 *A configuration of* n *points visible in* m *views is non-critical if (a) the points and the camera centres are non-coplanar and (b) all camera centres and an arbitrary point are non-collinear and (c) all points and an arbitrary camera centre are non-collinear and (d) at least* 2 *points do not lie on the reference plane.*

Proof We will show that a configuration which does fulfill the conditions (a), (b), (c) and (d) is a non-critical configuration. This will be done by actually constructing such a unique reconstruction.

As in the previous proof all points on the reference plane are detected and reconstructed separately. Furthermore, we state, that a point and a camera centre can never coincide, since such a point would not have a unique projection in such a camera. With the assumption that the conditions (a) and (d) are fulfilled we have at least two camera centres and two points which are not coplanar and not on the reference plane. W.l.o.g we denote the views as $\bar{\mathbf{Q}}_1$ and $\bar{\mathbf{Q}}_2$ and the points as $\bar{\mathbf{X}}_1$ and $\bar{\mathbf{X}}_2$. In the previous section we proved

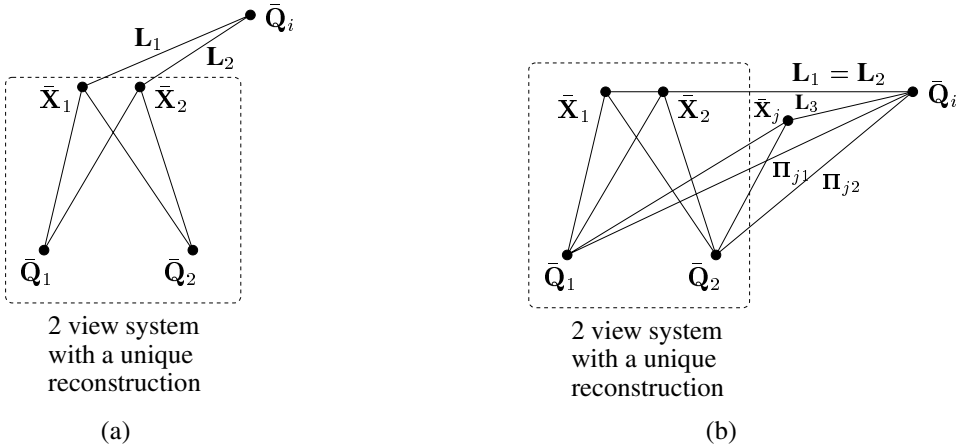


Figure 7.1. Geometric interpretations for the proof of theorem 9.

that we obtain a unique reconstruction for such a configuration. We will now show that we can add an arbitrary view $\bar{\mathbf{Q}}_i$ to the 2-view system and obtain a 3-view system with a unique reconstruction. Let us assume that the points $\bar{\mathbf{X}}_1, \bar{\mathbf{X}}_2$ and the camera centre $\bar{\mathbf{Q}}_i$ are not collinear. Fig. 7.1 (a) shows the geometric interpretation of such a configuration. Obviously the lines $\mathbf{L}_1 = \bar{\mathbf{X}}_1 - \bar{\mathbf{Q}}_i$ and $\mathbf{L}_2 = \bar{\mathbf{X}}_2 - \bar{\mathbf{Q}}_i$ uniquely define the camera centre $\bar{\mathbf{Q}}_i$.

In the other case, if $\bar{\mathbf{X}}_1, \bar{\mathbf{X}}_2$ and $\bar{\mathbf{Q}}_i$ are collinear, the lines $\mathbf{L}_1$ and $\mathbf{L}_2$ coincide (see fig. 7.1 (b)). This means that the camera centre $\bar{\mathbf{Q}}_i$ has one degree of freedom, i.e. has to lie on the line $\mathbf{L}_1$. Since we assume that the condition (c) is fulfilled there is a point $\bar{\mathbf{X}}_j$ which does not lie on the line $\mathbf{L}_1$. Let us consider the epipolar plane $\mathbf{\Pi}_{j1}$, which is defined by $\bar{\mathbf{X}}_j, \bar{\mathbf{Q}}_i$ and $\bar{\mathbf{Q}}_1$, and the epipolar plane $\mathbf{\Pi}_{j2}$, which is defined by $\bar{\mathbf{X}}_j, \bar{\mathbf{Q}}_i$ and $\bar{\mathbf{Q}}_2$. The intersection of the epipolar plane $\mathbf{\Pi}_{j1}$ and the line $\mathbf{L}_1$ defines the camera centre $\bar{\mathbf{Q}}_i$ uniquely if $\mathbf{L}_1$ and $\mathbf{\Pi}_{j1}$ do not coincide. The same applies to the epipolar plane $\mathbf{\Pi}_{j2}$. We will now show that either of these two cases is true. Let us assume that the two planes $\mathbf{\Pi}_{j1}$ and $\mathbf{\Pi}_{j2}$ are different. This implies that the two planes intersect uniquely in the line $\mathbf{L}_3 = \bar{\mathbf{X}}_j - \bar{\mathbf{Q}}_i$. Since $\bar{\mathbf{X}}_j$ does not lie on $\mathbf{L}_1$, the two lines $\mathbf{L}_1$ and $\mathbf{L}_3$ are different. Therefore, either the plane $\mathbf{\Pi}_{j1}$ or the plane $\mathbf{\Pi}_{j2}$ does specify the camera centre $\bar{\mathbf{Q}}_i$ uniquely. We are left with the case that $\mathbf{\Pi}_{j1}$ and $\mathbf{\Pi}_{j2}$ are identical. This implies that the plane $\mathbf{\Pi}_{j1}$ contains the camera centres $\bar{\mathbf{Q}}_1$ and $\bar{\mathbf{Q}}_2$. However, if $\mathbf{L}_1$ coincided with the plane $\mathbf{\Pi}_{j1}$, the condition (a) would be violated, i.e. $\bar{\mathbf{X}}_1, \bar{\mathbf{X}}_2, \bar{\mathbf{Q}}_1$ and $\bar{\mathbf{Q}}_2$ would be coplanar. Therefore, $\mathbf{L}_1$ cannot coincide with $\mathbf{\Pi}_{j1}$ and the camera centre $\bar{\mathbf{Q}}_i$ is uniquely defined by $\mathbf{L}_1$ and $\mathbf{\Pi}_{j1}$.

Furthermore, if $\bar{\mathbf{X}}_j$ lies on the baseline between $\bar{\mathbf{Q}}_1$ and $\bar{\mathbf{Q}}_i$ or on the baseline between $\bar{\mathbf{Q}}_2$ and $\bar{\mathbf{Q}}_i$, the point $\bar{\mathbf{X}}_j$ would specify this very baseline which means that the camera centre $\bar{\mathbf{Q}}_i$ is uniquely defined as well.

In this way all views can be added to the 2-view system. Therefore, we obtain a unique reconstruction with m views and the two points $\bar{\mathbf{X}}_1$ and $\bar{\mathbf{X}}_2$.

With the assumption that the condition (b) is fulfilled we can finally reconstruct all points. This means that for every configuration which does satisfy the conditions (a), (b), (c) and (d) we obtain a unique reconstruction. This concludes the proof.

□

Let us consider, which of the configurations that do not fulfill the requirements (a-d) are actually critical. Configurations that do not fulfill assumption (d) are critical as shown in proposition 4. A configuration that does not fulfill requirement (b), where all camera centres and one of the points are collinear, is obviously critical. Such a point lies on the baselines of all pairs of cameras and cannot be reconstructed. Therefore, configurations that do not fulfill assumption (c) are critical as well, since they are dual to configurations that do not fulfill assumption (b). However, a configuration that does not fulfill requirement (a), where all camera centres and points are coplanar, is not necessarily critical. Note that the fact that all possible pairs of 2 views are critical (as proved in theorem 1) does not imply that the complete configuration is critical. However, the investigation of configurations that do not fulfill assumption (a) for n points and m views would only be of theoretical interest.

Let us consider the question of sufficient visibility for n points and m views.

Corollary 3 *Every visibility matrix which contains* 2 *or more points (with at least* 2 *points outside the reference plane) and* 2 *or more views is sufficient if all points are visible in all views.*

Proof We choose a configuration which does fulfill the conditions (a), (b), (c) and (d). Obviously, this can be done for an arbitrary (more than 2) number of views and points. Such a configuration has a unique reconstruction as proved in theorem 9.

□

With the corollaries 2 and 3 we can conclude that the basic condition that $\#equations \geq \#dofs$ is a sufficient check for sufficient visibility in the case of no missing data. With a full visibility matrix we obtain: $\#equations = 2mn$ and $\#dofs = 3(m+n) - 4$.

7.3 Missing Data – Not Full Visibility Matrix

Compared to the case of *no missing data*, critical configurations for multiple projective views with *missing data* have received less attention in the past. In (Quan et al., 1999), all reconstructions with a sufficient visibility matrices of 3 and 4 images were cataloged. This work was extended in (Oskarsson et al., 2001) for one-dimensional cameras.

We will now address the questions of sufficient visibility and critical configurations for the case of missing data and with the assumption of having a known reference plane visible in all views. We will introduce a constructive method of choosing points and cameras which provide sufficient visibility and non-critical configurations.

7.3.1 Critical Configurations and Sufficient Visibility

Consider first the question of sufficient visibility for missing data. The basic condition that $\#equations \geq \#dofs$ is *insufficient* to answer the question of sufficient visibility. For a non-full visibility matrix, the maximum number of linearly independent equations is $\#equations = 2\#(V(i,j) = set)$ and the number of degrees of freedom is $\#dofs = 3(m+n)-4$ (points on the reference points excluded). However, if these equations include linear dependences, the number of linearly independent equations reduces. In order to give a complete answer for a given visibility matrix, the rank (or the subdeterminants) of the corresponding S-matrix has to be investigated for a generic set of points and cameras. Such an investigation can be carried out with MAPLE.

Let us consider the specific visibility matrix in eqn. 7.1. Although the number of equations is equal the number of degrees of freedom, i.e. $\#equations = 20 = \#dofs$, the corresponding S-matrix has rank 19, i.e. is rank deficient, for a generic set of points. In this case the linear dependence of equations can be seen if we consider the views 1 and 2 and the views 2 and 3 as separated 2-view systems. The second 2-view system includes a linear dependence since $\#equations = 12 > 11 = \#dofs$. Excluding e.g. point 5 results in linear independent equations for the second 2-view system, since $\#equations = 8 = \#dofs$. However, in this case the resulting S-matrix for the 3-view system is under-constrained since $\#equations = 16 < 17 = \#dofs$.

The general problem of critical configurations in the case of missing data is very complex. Basically ever specific visibility matrix might give a different set of critical configurations. Therefore, the rank (or the subdeterminants) of the S-matrix for a specific configuration has to be investigated in the same manner as we did in the 2-view case with no missing data.

7.3.2 A Constructive Method

So far we considered the questions of sufficient visibility and critical configurations for a given visibility matrix. However, in practice the placement of cameras and the number of visible points can be chosen freely to a certain extent. Therefore, it is of particular interest of having a method of choosing points and cameras which provide sufficient visibility and non-critical configurations.

We will now introduce and prove such a method for the multi-view case. This will be done in an iterative way in terms of the number of cameras. Assume the task of adding a new view $\bar{\mathbf{Q}}_{m+1}$ to a m-view system, where the reconstruction of n points and m views is unique. In order to obtain a unique reconstruction with the additional view, we have to specify the 3 dofs of the new camera centre $\bar{\mathbf{Q}}_{m+1}$. There are various ways. Assume that a point $\bar{\mathbf{X}}_i$, which is already reconstructed and does not lie on the reference plane, is visible in the view $\bar{\mathbf{Q}}_{m+1}$. Furthermore, a new point $\bar{\mathbf{X}}_{n+1}$, which does not lie on the reference plane, is visible in $\bar{\mathbf{Q}}_{m+1}$ and $\bar{\mathbf{Q}}_j$, which belongs to the m-view system. Fig. 7.2 shows the geometric interpretation. The point $\bar{\mathbf{X}}_i$ gives at least 2 more constraints. The point $\bar{\mathbf{X}}_{n+1}$ adds 3 dofs to the new $(m+1)$-view system, however, it supplies 4 more constraints on the system as well. This is sufficient for specifying the 3 dofs of the new camera centre

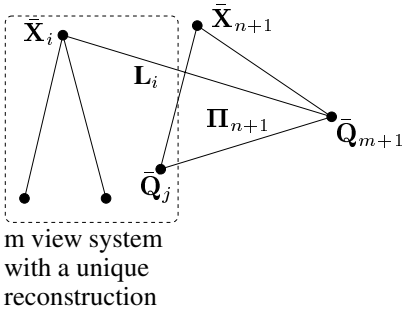


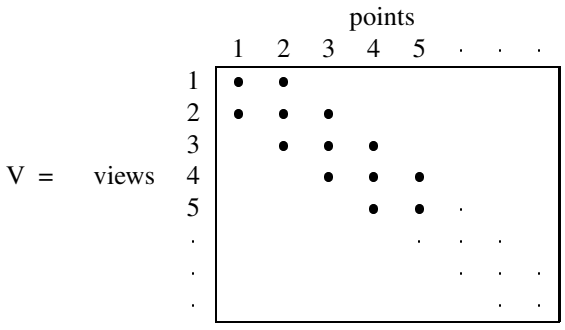
Figure 7.2. Adding a new view $\bar{\mathbf{Q}}_{m+1}$ to a system with n points and m views which has a unique reconstruction.

$\bar{\mathbf{Q}}_{m+1}$. Therefore, such a visibility is sufficient for obtaining a unique reconstruction for $m+1$ views.

The remaining question is: which are the critical configurations of such a multi-view system? Fig. 7.2 shows the geometric relationship between points and cameras. The point $\bar{\mathbf{X}}_i$ introduces a line $\mathbf{L}_i = \bar{\mathbf{X}}_i - \bar{\mathbf{Q}}_{m+1}$, where the camera centre $\bar{\mathbf{Q}}_{m+1}$ has to lie on. Furthermore, the point $\bar{\mathbf{X}}_{n+1}$ introduces the epipolar plane $\mathbf{\Pi}_{n+1}$, which contains the camera centres $\bar{\mathbf{Q}}_j$ and $\bar{\mathbf{Q}}_{m+1}$. Let us assume that the point $\bar{\mathbf{X}}_{n+1}$ does not lie on the baseline between $\bar{\mathbf{Q}}_j$ and $\bar{\mathbf{Q}}_{m+1}$. In this case, the camera centre is uniquely defined if $\mathbf{L}_i$ does not coincide with the plane $\mathbf{\Pi}_{n+1}$. This is true if $\bar{\mathbf{X}}_i, \bar{\mathbf{X}}_{n+1}, \bar{\mathbf{Q}}_j$ and $\bar{\mathbf{Q}}_{m+1}$ are not coplanar. We are left with the case that $\bar{\mathbf{X}}_{n+1}, \bar{\mathbf{Q}}_j$ and $\bar{Q}_{m+1}$ are collinear. The point $\bar{\mathbf{X}}_{n+1}$, which is assumed to be visible only in $\bar{\mathbf{Q}}_j$ and $\bar{\mathbf{Q}}_{m+1}$, cannot be reconstructed, since it lies on the baseline between $\bar{\mathbf{Q}}_j$ and $\bar{\mathbf{Q}}_{m+1}$. Let us summarize: *a configuration of such a multi-view system is critical if and only if (a) $\bar{\mathbf{X}}_i, \bar{\mathbf{X}}_{n+1}, \bar{\mathbf{Q}}_j$ and $\bar{\mathbf{Q}}_{m+1}$ are coplanar or (b) $\bar{\mathbf{Q}}_j, \bar{\mathbf{X}}_{n+1}$ and $\bar{\mathbf{Q}}_{m+1}$ are collinear.*

Obviously, adding more points to this system does not affect a non-critical configuration as long as the following condition is satisfied, such a point is not collinear with those camera centres from which the point is visible.

A possible visibility matrix for such a multi-view system is:



Such a band-structured matrix typically appears for reconstructing large scale scenes, e.g. architectural environments, as we saw in the previous chapter. It reflects the fact that model points appear and disappear in the sight of view while the camera moves around an object, e.g. a building.

7.4 Conclusions

We investigated configurations of multiple cameras, multiple 3D points and a known reference plane which do not have a unique projective reconstruction, so-called critical reference plane configurations. Furthermore, we addressed the question of sufficient visibility, the number of features necessary for a unique projective reconstruction. The assumption of having a known reference plane is equivalent to having 4 points in the scene which are visible in all views and known to be coplanar. All presented results are novel (Rother and Carlsson, 2002a) since to our knowledge critical configurations have only been studied for the general case (e.g. Kahl et al., 2001). We proved that a configuration of 2 views and 2 points (outside the reference plane) is critical if and only if the points and the camera centres are coplanar. For multiple views, we showed that if all points are visible in all views, i.e. no missing data, all configuration (apart from trivial ones) where points and camera centres are non-coplanar are non-critical. This is an important result since the scenario of one scene plane visible in multiple views appears frequently in practice (e.g. Pollefeys et al., 2002) and is critical in the general case. In the reference plane case this scenario is not critical if the reference plane is different to the scene plane, e.g. the reference plane is the correct plane at infinity. Furthermore, we introduced a method to construct non-critical configurations for the case of missing data.

Since lines, planes and cameras have also a linear relationship, the investigation of those features might be carried out in a similar way as for point features. This is an interesting topic for future research.

Chapter 8

An Automatic Multi-View Reconstruction System

The previous chapters presented and analyzed the reference plane approach to reconstruction. Chapter 6 demonstrated that it can be used to reconstruct difficult scenarios where general reconstruction methods fail. This chapter briefly outlines a complete automatic multi-view reconstruction system using the reference plane approach. The system reconstructs only 3D points, though it can be extend to use line and plane features. As the reference plane, the correct plane at infinity is used, derived from the vanishing points of mutually orthogonal directions (sec. 5.1.2). The input data to the system is a set of images. The output is a reconstruction of the 3D points and the corresponding cameras. The only user interaction is to specify which pair of images observe the same part of the scene.

The main and novel contributions of the system are a vanishing point detection method and a robust multi-view point matching algorithm. The key idea of the vanishing point detection method is to reject falsely detected vanishing points which do not give a reasonable calibration or rotation of the camera. It is based on our publications (Rother, 2000; Rother, 2002), but in the following presentation, certain aspects have been improved, like the use of RANSAC (Fischler and Bolles, 1981). The multi-view matching method uses the 2-view matching algorithm of (Tell and Carlsson, 2002) and the $m \geq 3$-view algorithm in Hartley and Zisserman (2000) (sec. 15.7.1 in their book). The novel idea is here to exploit a known reference plane. Consequently, our direct reference plane (DRP) method can be integrated in the robust matching process. An important advantage of our matching method is that it is not critical for the typical scenario of one dominant scene plane. In absence of a reference plane, this scenario leads to a difficult model selection problem (Pollefeys et al., 2002).

We will first give an overview of the complete system (sec. 8.1). Then each step is discussed in more detail. The result of each stage is documented for the house sequence. This sequence consists of 9 images. Fig. 8.6 shows the first 3 and fig. 6.27 the last 3 images of the sequence. A final 3D point reconstruction, using the system, is depicted in figures 8.10 and 8.11.

8.1 System Overview

The complete system consists of the following 6 steps:

1. For each view determine vanishing points of 2 or 3 orthogonal scene directions.
2. Calibrate each camera and determine its rotation (up to a 24-fold ambiguity).
3. Perform point matching between each pair of views.
4. Resolve the 24-fold ambiguity in the rotation matrices.
5. Perform multi-view point matching.
6. Reconstruct 3D points and cameras using our DRP method.

In the first step, 2 or 3 vanishing points of dominant, mutually orthogonal scene directions are detected in each image. This algorithm is described in sec. 8.2. Using the vanishing points, a "square pixel" camera with zero skew and aspect ratio one may be calibrated (sec. 5.1.2). Furthermore, the rotation matrix may be determined up to a 24-fold ambiguity. Consequently, the infinite homography $H = KR$ of each camera is known (up to a 24-fold ambiguity). This procedure is described in sec. 8.3. Section 8.4 presents first a point matching method for pairs of views (Tell and Carlsson, 2002). We extend this algorithm by incorporating the known infinite homographies (up to the 24-fold ambiguity). Using the 2-view point matches, the 24-fold ambiguity in the rotation matrix may be resolved (sec. 8.3). Furthermore, sec. 8.4 describes a robust, m-view ($m \geq 3$) point matching algorithm based on the known homographies and our DRP method. From the set of matched points, the scene and the cameras are reconstructed simultaneously (sec. 8.5).

8.2 Orthogonal Vanishing Point Detection

Man-made environments are often characterized by many parallel lines and orthogonal edges. Examples are depicted in fig. 8.1 and 5.2. Section 5.1.2 explained the importance of vanishing points of mutually orthogonal scene directions. They can be used to derive the camera's calibration and rotation matrix. This gives the correct plane at infinity which can be used as a reference plane for reconstruction. This section addresses the task of determining vanishing points of mutually orthogonal directions. Consider two images of the house sequence in fig. 8.1. About 1000 line segments were detected automatically in each image using a standard image processing method (Rosin and West, 1989). In fig. 8.1 (a), many line segments belong to one (or two)[1] of the three dominant scene directions of the house. In contrast to this, most of the line segments in fig. 8.1 (b) only belong to two dominant directions (horizontal and vertical). The third direction is parallel to the optical axis of the camera and only about 5 short line segments on the roof lie in this direction.

[1] Line segments which lie on or close to the vanishing line of two vanishing points belong to both vanishing points. The vanishing line in fig. 8.1 (a) is the horizon.

(a) (b)

Figure 8.1. Two images of the house sequence. In (a) 846 and in (b) 1174 line segments were detected automatically.

These are typical scenarios for man-made environments. We may specify the vanishing point detection task as follow. The input data is a set of line segments. The output is 0, 2 or 3 vanishing points of mutually orthogonal scene directions. With no vanishing points, the scene did not contain any dominant orthogonal directions.

The vanishing point detection task has raised considerable interest in the past (Barnard, 1983; Quan and Mohr, 1989; Brillault-O'Mahony, 1991; van den Heuvel, 1998; Tuytelaars et al., 1998; Coughlan and Yuille, 1999; Rother, 2000; Deutscher et al., 2002; Koseká and Zhang, 2002; Rother, 2002). Since in the 1980's computational power was very limited, most of the early works concentrated on efficiency. The problem may be simplified by mapping the line segments from the image onto a Gaussian sphere (Barnard, 1983). However, as we pointed out in (Rother, 2000), this may introduce a substantial error. Mapping the image onto another surface might change considerably the distances between line segments and vanishing points. We will see later that any vanishing point detection process has to formulate this distance function in some way. Most of the recent works use the line segments directly, i.e. do not perform a mapping. (Coughlan and Yuille, 1999; Deutscher et al., 2002; Koseká and Zhang, 2002) formulated the problem in a probabilistic framework. More references and a detailed discussion can be found in our journal paper (Rother, 2002). The main and novel contribution of our work is the identification of *all* conditions given by 2 or 3 vanishing points of mutually orthogonal directions. The conditions are that vanishing points have to define a "reasonably" square pixel camera with a correct rotation matrix. This gives a robust vanishing point detection algorithm as will be explained now. In the thesis we improved the efficiency of our previous method by using RANSAC. Before we can formulate our vanishing point detection approach, two issues have to be addressed. First, what are the criteria for orthogonal vanishing points? Secondly, what is the distance between a line segment and a vanishing point?

Camera and orthogonality criterion

Consider the criteria which 2 or 3 vanishing points of mutually orthogonal directions have to satisfy. The vanishing points can be used to calibrate the principal point and the focal length of a square pixel camera (sec. 5.1.2). The first criterion, the *camera criterion*, is fulfilled if the principal point and the focal length are inside a certain range, in case they are calculable. Using the calibrated camera, the second criterion, the *orthogonality criterion*, is fulfilled if the 2 or 3 directions, given by the vanishing points, are orthogonal.

In the following we will describe how to calibrate a square pixel camera from a *single* view. The *multi*-view case is considered in sec. 8.3. Assume that the 2 or 3 vanishing points were identified in an image as $\mathbf{v}_1, \mathbf{v}_2$ and $\mathbf{v}_3$. To calibrate a square pixel camera, Caprile and Torre (1990) assumed that the 3 vanishing points are *finite*. However, this is not always the case as shown in fig. 8.1(a), where two of the vanishing points are close to infinity in the image. Liebowitz and Zisserman (1999) investigated the different cases where some of the vanishing points are infinite. This discussion is summarized here, together with the case of 2 vanishing points.

1. *Three finite vanishing points $\boldsymbol{v}_1, \boldsymbol{v}_2$ and $\boldsymbol{v}_3$:*
   The focal length and principal point are uniquely defined (sec. 5.1.2). The orthogonality criterion is given as the condition that each angle of the triangle formed by $\mathbf{v}_1, \mathbf{v}_2$ and $\mathbf{v}_3$ is smaller than 90^o.

2. *Two finite vanishing points $\boldsymbol{v}_1, \boldsymbol{v}_2$ and one infinite vanishing point $\boldsymbol{v}_3$:*
   The principal point lies on the line segment which is defined by the two endpoints $\mathbf{v}_1$ and $\mathbf{v}_2$. For real world cameras the principal point is more likely to be positioned in the centre of the CCD array. Therefore, we choose the principal point as the point which lies on the line segment and is closest to the image centre. By determining the principal point, the focal length is uniquely defined. In this case the orthogonality criterion is defined by the condition that the direction of the infinite vanishing point $\mathbf{v}_3$ is orthogonal to the line defined by $\mathbf{v}_1$ and $\mathbf{v}_2$.

3. *One finite vanishing point $\boldsymbol{v}_1$ and two infinite vanishing points $\boldsymbol{v}_2, \boldsymbol{v}_3$:*
   In this case the principal point is identical to the vanishing point $\mathbf{v}_1$. The focal length cannot be determined. The orthogonality criterion is defined by the condition that the directions of v_2 and v_3 are orthogonal.

4. *Two finite vanishing points $\boldsymbol{v}_1, \boldsymbol{v}_2$:*
   The principal point is chosen as the image centre. The focal length is then uniquely defined. The orthogonality criterion is, as above, the condition that each angle of the triangle formed by $\mathbf{v}_1, \mathbf{v}_2$ and $\mathbf{v}_3$ is smaller than 90^o.

5. *One finite vanishing point $\boldsymbol{v}_1$ and one infinite vanishing point $\boldsymbol{v}_2$:*
   The principal point lies on the line which passes through $\mathbf{v}_1$ and is perpendicular to $\mathbf{v}_2$. As the principal point we choose the point on the line which is closest to the image centre. The focal length is then uniquely defined. There is no orthogonality criterion in this case.

6. *Two infinite vanishing points* $\boldsymbol{v}_1, \boldsymbol{v}_2$*:*
   Neither the principal point nor the focal length can be determined. The orthogonality criterion is defined by the condition that the directions of $\mathbf{v}_1$ and $\mathbf{v}_2$ are orthogonal.

In practice, a detected vanishing point might be close to infinity in the image, but seldomly exactly at infinity. The simplest way to deal with this issue is to introduce a threshold for points being infinite or finite. In a probabilistic framework, this binary decision could be formulated with the likelihood of a point being at infinity. In our current implementation we use a threshold.

The distance measurement between a vanishing point and a line segment

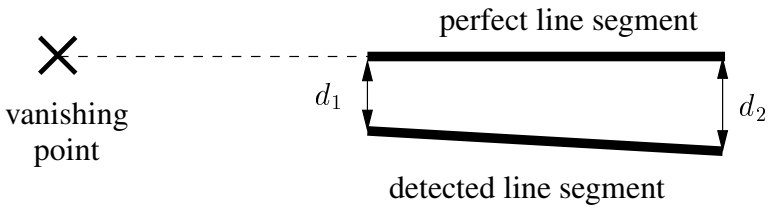


Figure 8.2. The perfect line segment, which supports a vanishing point, is not identical to the detected line segment due to noise.

Consider a perfect line segment which supports a vanishing point, i.e. its extension passes through the vanishing point (fig. 8.2). Due to various reasons, e.g. noise and lens imperfections, the perspective projection of a line segment from the 3D scene onto the 2D image is not congruent with the line segment detected in the image. All vanishing point detection methods have to formulate either implicitly or explicitly a distance function between the detected and the perfect line segment. However, in practice the perfect line segment is unknown and therefore all line segments in the image could be the perfect one. Which is the "closest" perfect line segment? To simplify the problem, we consider only the endpoints of a line segment. The distance between the detected and perfect line segment may be approximated by $\sqrt{d_1^2 + d_2^2}$ (see fig. 8.2). Given a detected line segment $\mathbf{l}$ and vanishing point $\mathbf{v}$, Liebowitz (2001) presented a closed-form solution for the closest perfect line segment based on this distance function. If the perfect line segment is known, we may define the distance between $\mathbf{l}$ and $\mathbf{v}$ as $dis(\mathbf{l}, \mathbf{v}) = (d_1 + d_2)/2$. Alternatively, a line segment could be represented by a collection of points. This would lead towards a more "correct" distance function. However, it might be very difficult or impossible to formulate a closed-form solution for this problem. Since this distance function has to be computed many times in any vanishing point detection method, a closed-form solution is preferable.

Vanishing point detection

Our vanishing point detection method is based on RANSAC (Fischler and Bolles, 1981). The basic idea of RANSAC is to estimate a model by randomly selecting the minimal number of data needed to predict the model (see sec. 4.1). In our case, the minimum number of 3 pairs of line segments is chosen, to compute all the 3 potential vanishing points simultaneously. The outline of our method is as follows.

1. Repeat N times: take randomly 6 line segments and compute $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$.
   2. Compute the total length of all inliers for the 4 combinations: $(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)$, $(\mathbf{v}_1, \mathbf{v}_2)$, $(\mathbf{v}_1, \mathbf{v}_3)$ and $(\mathbf{v}_2, \mathbf{v}_3)$. Each line segment with $dis(\mathbf{l}, \mathbf{v}) < T$ is an inlier.
   3. Check if one of the combinations is better than the best solution so far.
      4. Store this combination as the best solution if the camera criterion and the orthogonality criterion is fulfilled.
5. Take the best solution if the ratio (length of all inliers) / (length of all line segments) is larger than a threshold. Otherwise, no orthogonal directions could be detected.

The reason why we explicitly consider the length of a line segment is that longer line segments are detected more reliably than shorter ones. The number of iterations N can be adapted during the search (Hartley and Zisserman, 2000).

Fig. 8.3 shows the result of our method for the image in fig. 8.1 (a). The threshold T was chosen as 5pixels. All three vanishing points were determined successfully. The result of the image in fig. 8.1 (b) is depicted in fig. 8.4. In this case only 2 dominant vanishing points were detected. In 4 out of 9 images of the house sequence 3 vanishing points were detected. In the remaining 5 images, 2 vanishing points were found.

We would like to mention that the algorithm presented above is fairly simple. Further practical issues are discussed in (Rother, 2002). In this publication we also documented that the method has been successfully applied to different man-made scenes. To improve the simple method a probabilistic framework, like in (Coughlan and Yuille, 1999; Deutscher et al., 2002; Koseká and Zhang, 2002), could to be introduced.

(a) (b) (c)

Figure 8.3. The 846 line segments in fig. 8.1(a) were classified into three dominant directions. The first direction (a) is supported by 237 line segments, direction (b) by 355 line segments and direction (c) by 301 line segments. In this case 155 line segments were assigned to both the direction in (b) and (c). The remaining 148 line segments were classified as outliers.

(a) (b)

Figure 8.4. The 1174 line segments in fig. 8.1(b) were classified into two dominant directions. The first direction (a) is supported by 308 line segments. The second direction (b) by 643 line segments. The remaining 223 line segments were classified as outliers.

8.3 Multi-View Camera Calibration and Rotation

For each image the previous step gives 2 or 3 vanishing points. Section 8.2 described how to use them for the estimation of the principal point and the focal length of a *single* square pixel camera. This section considers *multiple* cameras. The idea is to improve the camera's calibration by assuming constant internal parameters of all multiple cameras. Furthermore, sec. 5.1.2 showed that the 2 or 3 vanishing points determine the camera's rotation up to a 24-fold ambiguity. For multiple cameras this ambiguity can be resolved, as shown here.

Camera calibration

Fig. 8.5 shows the estimated focal lengths (a) and principal points (b) of 7 cameras of the house sequence. For 2 cameras (5 and 9), 2 infinite vanishing points were detected, which means that no camera parameters can be estimated.

All of the focal lengths are between 600 and 800pixels, apart from one exception. In this case we chose the focal length as the average of all estimated focal lengths, $f = 734$pixels.

The principal point lies for most real world cameras close to the centre of the CCD array, shown as a dashed cross in fig. 8.5(b). The estimated, average principal point is close to the image centre. However, the variation in the estimated principal points is fairly large. Therefore, on the basis of this data it is doubtful that the average principal point is a better estimation of the real principal point than the image centre. In this experiment, we chose the image centre as the principal point.

We also implemented a MAP estimator including an uncertainty model for the line segments, principal points and focal lengths. The probability functions of the principal point and focal length were approximated by a Monte-Carlo simulation. However, this non-linear optimization did not have a large effect on the final 3D reconstruction and is

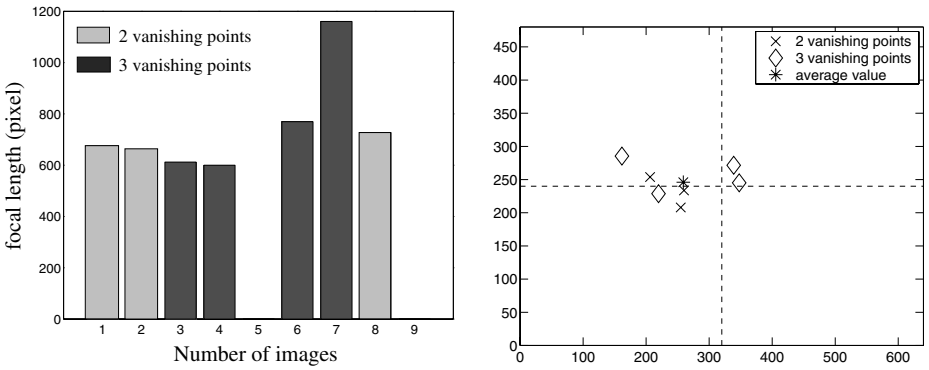


Figure 8.5. Estimation of the focal lengths (a) and the principal points (b) of the 9 images of the house sequence. The image in (b) is of size 640×480pixels.

therefore not documented here. For such an approach more images are probably necessary, like a continuous image sequence.

Determining the camera's rotation

Assume that 2 or 3 vanishing points are given as $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$. If the camera calibration K is known, the vector $\mathbf{d}_i = K^{-1}\mathbf{v}_i$ represents a direction in the scene. For 2 vanishing points, the third direction $\mathbf{d}_3$ can be obtained by $\mathbf{d}_3 = \mathbf{d}_1 \times \mathbf{d}_2$. The three orthogonal directions define the rotation matrix of a camera as

$$R = (\pm\mathbf{d}_{1,2,3}| \pm \mathbf{d}_{1,2,3}| \pm \mathbf{d}_{1,2,3}) \quad , \tag{8.1}$$

where $\mathbf{d}_i$ are normalized to unit length. With the condition that the determinant of R is one, R has 24 possible solutions (see sec. 5.1.2). In the following we will discuss how to resolve this ambiguity for multiple cameras. We assume that each pair of views, which observes the same part of the scene, has been matched successfully. This gives a number of matched image points and the fundamental matrix between each pair.

Assume 2 views each with 24 different rotation matrices, R_1^{1-24} and R_2^{1-24}. For the first view we may choose one of the 24 rotation matrices as R_1. This fixes the general rotation of the metric space. What are the conditions for the correct rotation matrix R_2? Assume that image points in these two views have been matched correctly, and that the fundamental matrix is known (see system overview in sec. 8.1). The fundamental matrix between two views may be written as (eqn. 3.26)

$$F \;\sim\; (K_2 R_2^{1-24})^{-T}\; [\mathbf{e}]_\times (K_1 R_1)^{-1} \quad , \tag{8.2}$$

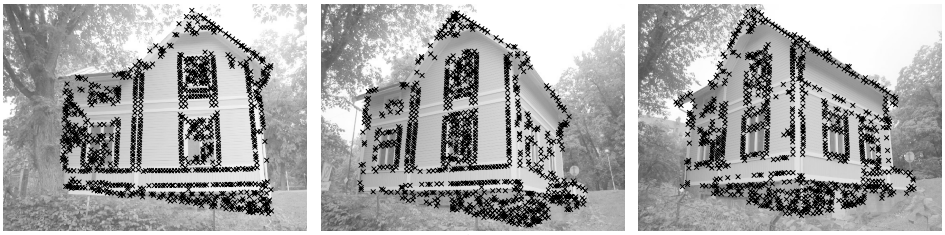
where $\mathbf{e}$ contains the two camera centres, i.e. $\mathbf{e} = \bar{\mathbf{Q}}_2 - \bar{\mathbf{Q}}_1$, and $[\cdot]_\times$ denotes a skew-symmetric matrix (eqn. 3.27). Eqn. 8.2 can be transformed into

$$(K_2 R_2^{1-24})^T\; F\; (K_1 R_1) \;\sim\; [\mathbf{e}]_\times \quad . \tag{8.3}$$

If F, K_1, K_2, R_1 and R_1^{1-24} are known we may compute the skew-symmetric matrix in eqn. 8.3. Therefore, the condition for the correct rotation matrix R_2 is, that the matrix in eqn. 8.3 is skew-symmetric. An algebraic "skew-symmetric check" is to compute the perfect skew-symmetric matrix from the given matrix and then determine the Frobenius norm between the two matrices. Unfortunately, this is a necessary but not sufficient condition. It can be shown that for certain camera motions, like $\mathbf{e} \sim (1,0,0)^T$, two or more of the rotation matrices satisfy this condition. A second, sufficient condition is that the 3D reconstruction of all matched points must lie in front of both cameras. For each rotation matrix, this can be evaluated using our DRP method. However, computing a 3D reconstruction is time consuming. Therefore, we suggest the following 2-step method. First, sort all 24 rotation matrices according to the first "skew-matrix condition". Secondly, evaluate the second, "reconstruction condition" using successive candidate rotation matrices. The first rotation matrix which satisfies the second condition is taken as the correct rotation.

For multiple views we may perform the described 2-view method for each possible pair of views. The correct orientation is then achieved by chaining the 2-view solutions together. For the house sequence this method gave the correct set of rotation matrices.

8.4 Multi-View Matching



(a) (b) (c)

Figure 8.6. The first three image of the house sequence with superimposed Harris corners. Image (a) has 941 corners, (b) 902 corners and (c) 871 corners.

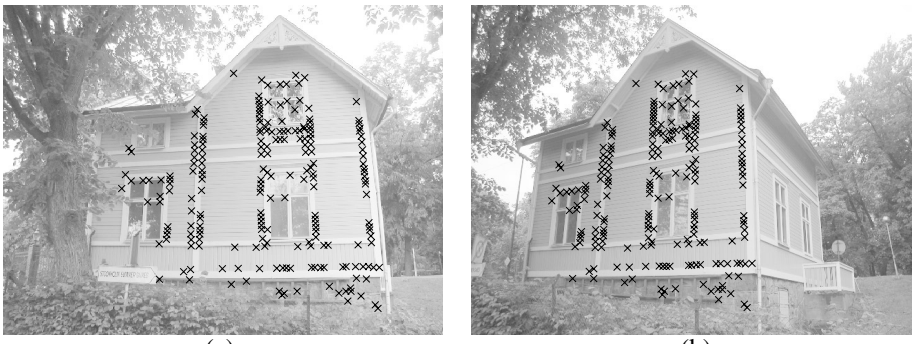
Consider the first 3 images of the house sequence (fig 8.6). In each image approximately 900 Harris corners were detected automatically (Harris and Stephens, 1988). In this case the part of the image, containing the house, was manually segmented. This was done to simplify the matching task[2]. For a 3D point reconstruction, the image points have to be matched. This section explains how to detect those image points (in 2 or more views) which represent the same 3D point in space. Such image points have to satisfy two conditions. First, the intensity neighborhood of the image points have to be similar, so-called photometric constraints. Secondly, the image points have to satisfy geometric constraints. The geometric constraints were discussed in detail in chapter 3, for general and reference

[2]We did not apply our matching algorithm to all Harris corners of the complete image, including the background.

plane configurations. Intuitively, the corresponding image points have to define a unique set of cameras and 3D points. The geometric constraints for point features in multiple views are now well understood, due to a considerable research effort in the last decade (Hartley and Zisserman, 2000).

In the following we will present a sequential method which first matches all pairs of views, then triplets of views and so on[3]. The 2-view problem is the most difficult. Many point matching methods have been suggested (see Tell (2002) for an overview). Most methods solve the problem in 2 steps. First, find a set of candidate matches using the photometric constraints. Then, the unique set of point matches is determined robustly using the geometric constraints. For wide baseline images, like the house sequence, the first step is the bottleneck. The 3-view matching problem is considerably simpler, since the check of the photometric constraints may be omitted. The set of 2-view matches can be used to create candidate 3-view matches. For more than 3 views, neither the photometric nor the geometric constraints have to be evaluated, since there are no geometric constraints which involve more than 3 views (sec. 3.2.3). For a small baseline image sequence, it might be advantageous to evaluate the geometric constraints for more than 3 views as well. We introduce now a robust point matching method which uses a known reference plane and can handle any number of views.

2-view matching



(a) (b)

Figure 8.7. Between the images in fig. 8.6 (a) and (b), 198 image points where matched correctly. Two image points match correctly if they lie on the corresponding epipolar lines.

Our 2-view point matching approach is based on Tell and Carlsson's (2002) algorithm. The key idea of Tell and Carlsson's (2002) matching method is to consider the intensity profile of pairs of image points. This gives a set of candidate matches. The geometric constraints, in terms of the fundamental matrix, are evaluated using the robust RANSAC

[3]In practice, we investigate only those sets of images which observe the same part of the scene.

(a) (b)

Figure 8.8. Between the images in fig. 8.6 (b) and (c), 406 image points where matched correctly. Two image points match correctly if they lie on the corresponding epipolar lines.

	1	2	3	4	5	6	7	8	9
1		198	65						
2	198		406	42					
3	65	406		379	53	21			
4		42	379		255	37	24		
5			53	255		328	28		
6			21	37	328		309	56	
7				24	28	309		367	94
8						56	367		323
9							94	323	

Table 8.1. The table lists the number of matches between each pair of views.

method. This means that one or three F matrices are computed from 7 randomly selected point matches. In this work a general scene without a known reference plane is assumed.

In our system, the infinite homography (up to a 24-fold ambiguity) of each camera is derived before the matching, $H^{1-24} = KR^{1-24}$ (see system overview in sec. 8.1). This information may be included in the geometric check. As discussed in sec. 3.2.3, for reference plane configurations, 2 point matches are sufficient to determine the cameras and also the fundamental matrix. In this case, 2 points matches give 24 possible fundamental matrices. They are used instead of the 3 matrices in the general case.

This reference plane matching approach has two main advantages. First, choosing 2 instead of 7 points improves the computation time. Eventually, more trials might also improve the quality of the result. Secondly, it performs well for the difficult scenarios containing one dominant scene plane. Scenes with one dominant plane are typical in man-made environments. Fig. 8.7 shows an example. Without a known reference plane, this scenario is critical (chapter 7). The matching task can, however, still be performed using a homography instead of the fundamental matrix. Since in general, information about

the scene is not available, this leads to the difficult model selection problem discussed in (Pollefeys et al., 2002).

The performance of our 2-view matching method is depicted in table 8.1. It lists the number of matches between each pair of views. We matched only those views which observe the same part of the scene. Furthermore, only those image pairs with more than 15 matches were accepted. Since the images are ordered, the matrix has a diagonal form. Furthermore, adjacent images, i.e. close to the diagonal, have a higher number of matches. Fig. 8.7 shows the matched points of the first two images of the house sequence (fig. 8.6(a) and (b)). All 3D points lie approximately on a plane. A general, non reference plane, matching method would have to use a homography instead of the fundamental matrix for this matching task. Fig. 8.8 shows the correctly matched image points for the images in fig. 8.6(b) and (c). In this case the 3D points do not lie on a dominant plane.

M-view matching

We will now introduce an m-view ($m \geq 3$) reference plane matching algorithm which uses the $(m-1)$-view matches. The algorithm inspects the geometric constraints in a robust manner using RANSAC. It is very similar to the approach in (Hartley and Zisserman, 2000) (sec. 15.7.1). In contrast to their method, we apply our DRP method.

The previous steps of the system computed uniquely the infinite homography for each camera and all 2-view matches (see system overview in sec. 8.1). The following m-view algorithm is first applied to all triplets of views, $m = 3$. After that 4 and more views are considered. As already mentioned, the candidate matches for $m \geq 4$ may be accepted without an additional geometric check.

The outline of the m-view matching algorithm is as follows. The reprojection error of a 3D point $\mathbf{X}_i$ in a camera P_j is denoted $d_{ij} = || \bar{\mathbf{x}}_{ij} - \overline{P_j \mathbf{X}_i} ||_2$.

1. Create m-view candidate matches from a set of $(m-1)$-view matches.
2. Repeat N times: take randomly 2 m-view matches.
   3. Reconstruct the cameras P_j and 2 points with our DRP method using the infinite homographies H_j.
   4. Reconstruct linearly all points $\mathbf{X}_i$ by intersection.
   5. Perform a non-linear optimization of $\sum_j d_{ij}$ for each point $\mathbf{X}_i$.
   6. Count the number of inliers $\mathbf{X}_i$, which have $\max_i d_{ij} < T$.
7. Take the solution with the largest number of inliers.

The main difference to Hartley and Zisserman's (2000) approach is the use of our DRP method to reconstruct 2 3D points visible in multiple views (step 3.). This simplifies their approach. Note that it is also feasible to use our DRP method for finite reference planes. The scenario of one (or two) 3D points on a finite reference plane cannot be reconstructed by any method, since it is critical (chapter 7). As mentioned above, this m-view reference plane matching approach is not critical for the typical scenario of one dominant scene plane.

Fig. 8.9 shows the result of the 3-view matching algorithm, using the two 2-view matches in fig. 8.7 and 8.8. From the 198 matches in fig. 8.7 and the 406 matches in fig.

8.8, 125 3-view matches were found. In this case the threshold on the reprojection error was set to $T = 4$pixels. Table 8.2 lists the number of m-view matches for the complete house sequence with 9 images. All candidate matches for more than 3 views were accepted directly. The total number of matches is 451 for $T = 4$pixels and 631 for $T = 7$pixels. In both cases no match which is in more than 7 views was found. All 2-view matches were deleted, since they might not represent a correct 3D point due to the weaker geometric constraints of 2 views. The visibility matrix for these two cases is shown in fig. 8.12 ($T = 4$pixels) and 8.15 ($T = 7$pixels).

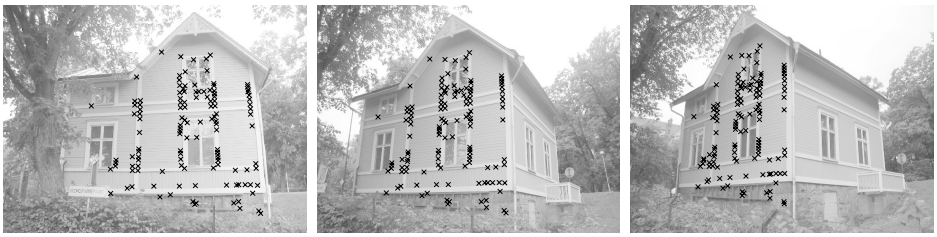


Figure 8.9. Between the first three images of the house sequence (fig. 8.6), 125 image points where matched correctly.

number of views	number of matches ($T = 4$pixels)	number of matches ($T = 7$pixels)
2	—	—
3	354	420
4	61	159
5	33	45
6	3	6
7	0	1
total	451	631

Table 8.2. Number of m-view matches for the house sequence with 9 images.

8.5 3D Reconstruction and Camera Recovery

All m-view matches together with the infinite homographies of all cameras may now be used to reconstruct all 3D points and cameras simultaneously with our DRP method. For the case of 451 point matches, a top and side view of the reconstruction is depicted in figures 8.10 and fig. 8.11. Nearly all reconstructed 3D points belong to one of the 3 dominant planes of the house. Figures 8.13 and 8.14 show the reconstruction of 631 point matches. In this case, some 3D points of the roof and the balcony were also detected.

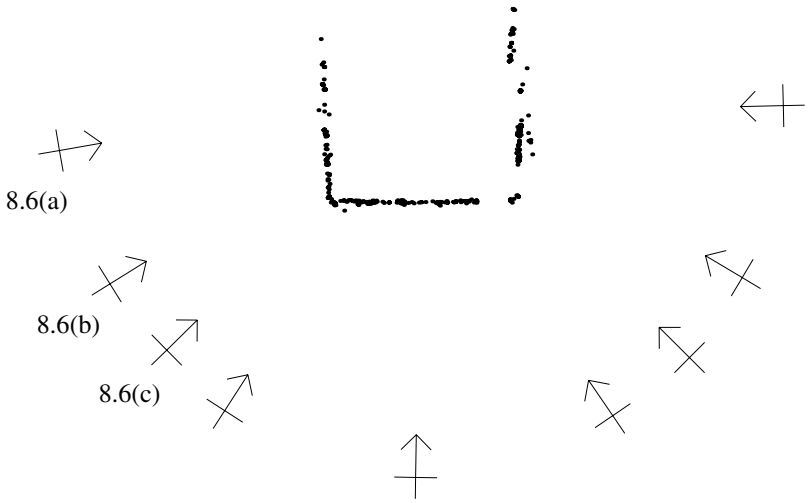


Figure 8.10. Top view of the reconstruction of 451 3D points ($T = 4$pixels). The cameras are depicted as arrows. The labeled cameras correspond to images in fig. 8.6.

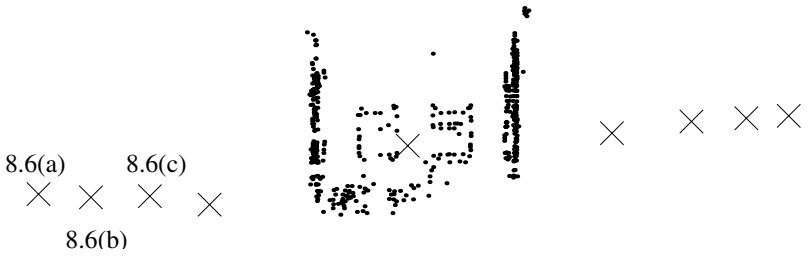


Figure 8.11. Side view of the reconstruction of 451 3D points ($T = 4$pixels). The cameras are depicted as crosses. The labeled cameras correspond to images in fig. 8.6.

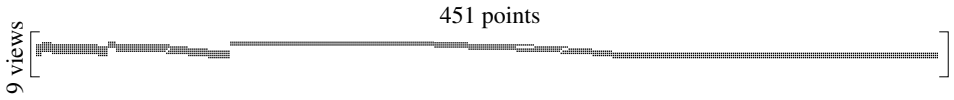


Figure 8.12. The visibility matrix of the house sequence and $T = 4$pixels.

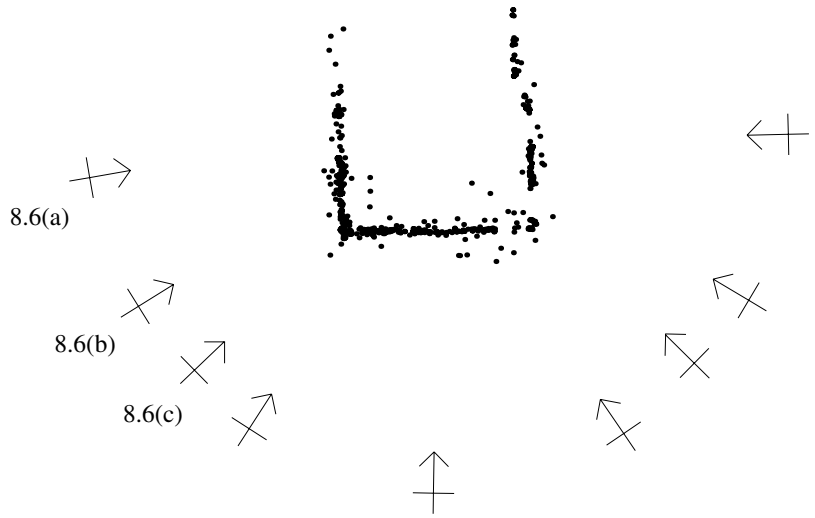


Figure 8.13. Top view of the reconstruction of 631 3D points ($T = 7$pixels). The cameras are depicted as arrows. The labeled cameras correspond to images in fig. 8.6.

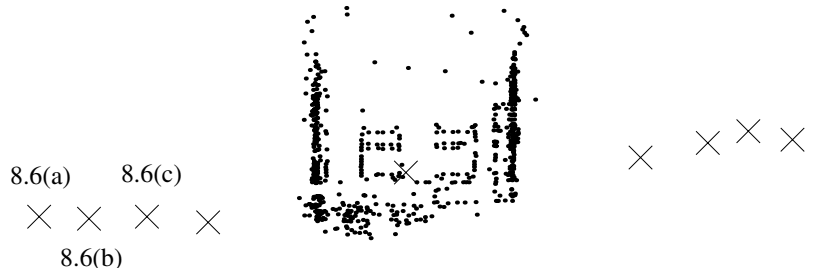


Figure 8.14. Side view of the reconstruction of 631 3D points ($T = 7$pixels). The cameras are depicted as crosses. The labeled cameras correspond to images in fig. 8.6.

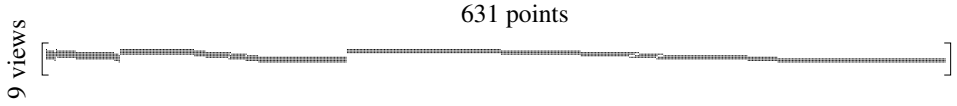


Figure 8.15. The visibility matrix of the house sequence and $T = 7$pixels.

8.6 Summary

This chapter introduced an automatic multi-view reconstruction system for point features. The main contribution is to demonstrate that the reference plane approach has the capability of performing automatically the complete reconstruction task. Two novel methods were part of this system, a vanishing point detection method and a robust multi-view point matching algorithm. The detection of vanishing points of 2 or 3 mutually orthogonal scene directions is based on our publications (Rother, 2000; Rother, 2002). As discussed in sec. 5.1.2, the vanishing points of mutually orthogonal directions can be used to determine the camera's calibration and rotation. The novel idea of this method is to reject falsely detected vanishing points which do not give a reasonable calibration or rotation of the camera. We improved the computation time of the method considerably by using RANSAC. The multi-view point matching algorithm is based on the 2-view matching technique of Tell and Carlsson (2002) and the m-view ($m \geq 3$) matching algorithm described in Hartley and Zisserman (2000) (sec. $15.7.1$ in their book). The novel idea is to exploit a known reference plane. In this case 2 point matches are sufficient to determine the camera matrices of multiple views. In contrast to (Hartley and Zisserman, 2000), our direct reference plane method is used for this minimal reconstruction task. It is important to note that our approach is also feasible for *finite* reference planes. The scenario of one (or two) 3D points on a finite reference plane cannot be reconstructed by any method, since it is critical (chapter 7). A further, important advantage of this reference plane matching approach is that it is not critical for the typical scenario of one dominant scene plane. In the general case, without a reference plane, this scenario leads to the difficult model selection problem (Pollefeys et al., 2002).

A main limitation of this system is that it is only applicable for man-made environments which contain at least two dominant orthogonal directions. The house example fulfilled this condition. In our publications (Rother, 2000; Rother, 2002) more examples are depicted, where our method detected successfully mutually orthogonal directions. However, more experiments using the complete system are necessary. Furthermore, line and plane features, as well as scene constraints, can be integrated into the system. Furthermore, it would be interesting to extract completely automatically a textured, virtual model from the 3D point reconstruction. Finally, an automatic reconstruction system for other reference plane approaches, such as a real scene plane, could be developed. The finite reference plane can then be used for self-calibration.

Chapter 9

Conclusions

The task of 3D reconstruction from a set of 2D images is a complex and difficult problem as can be seen from decades of research devoted to this topic. One immediate conclusion is that there does not exist a *single* technique which can be applied in *any* situation. This thesis presents a feature based reconstruction approach which is applicable to a wide range of situations. These situations are characterized as "reference plane configurations". The main contribution of the thesis is a novel reconstruction approach for reference plane configurations and a demonstration that it can be used to reconstruct difficult scenes where general (non reference plane) reconstruction methods fail. For reference plane configurations the reconstruction task may be divided into two steps:

1. Determine a real or virtual reference plane
2. Use the reference plane for 3D reconstruction

This rather vague formulation poses two questions:

- What are real or virtual reference plane configurations?
- How can the reference plane be used for 3D reconstruction?

These are the two fundamental questions addressed in the thesis. A simple answer to both questions can be given on the basis of a *single* formula.[1] The projection of a Cartesian 3D point $\bar{\mathbf{X}}$ to the image point $\mathbf{x}$ using a camera with Cartesian centre $\bar{\mathbf{Q}}$ and matrix H may be formulated algebraically as[2]

$$\mathbf{x} = H (\bar{\mathbf{X}} - \bar{\mathbf{Q}}) .$$

The camera matrix H is called the infinite homography and can be derived from a *real* scene plane visible in the image. However, this is not the only approach for identifying the

[1] This shows as well that great ideas can sometimes be very simple. On the other hand, it is amazing that one formula can keep one busy for $3\ 1/2$ years.

[2] For simplicity, the unknown scale factor (depth) of a homogeneous image point is omitted. However, this simplification does not affect the following conclusions.

infinite homography. We presented various approaches, such as using orthogonal scene directions or cameras with parallel projection. The reference plane is denoted as *virtual* if the infinite homography does not represent a real reference plane in the scene. In summary, the answer to the first question is, all possible configurations where the infinite homography can be derived by some means are (real or virtual) reference plane configurations. However, what is the advantage of knowing the infinite homography for 3D reconstruction? For general (non reference plane) configurations, the task of 3D reconstruction is to compute the unknown points $\bar{\mathbf{X}}$, the unknown cameras centres $\bar{\mathbf{Q}}$ and the unknown infinite homographies H. The only given information is the image points $\mathbf{x}$. This is a "difficult" problem since the three unknown parameters have a *non-linear* relationship. For reference plane configurations, the reference plane supplies the infinite homography H. This transforms the difficult, non-linear problem into a simple, *linear* problem with $\bar{\mathbf{X}}$ and $\bar{\mathbf{Q}}$ as the only unknowns. Therefore, the answer to the second question is, 3D points and camera centres can be reconstructed simultaneously from a single, linear system which consists of image measurements only. This "surprisingly" simple result was first published in (Rother and Carlsson, 2001) and is the most important contribution of the thesis.

9.1 Summary and Future Work

The thesis introduced novel reconstruction algorithms for points, lines and planes using a real or virtual reference plane. We call them the *D*irect *R*eference *P*lane (DRP) methods. They were presented theoretically in chapter 3 and are outlined in chapter 6. The main characteristic of the novel algorithms is that they are linear and reconstruct all cameras and all 3D features (off the reference plane) simultaneously from a single linear system of image measurements. This makes them potentially superior to all previously presented reference plane reconstruction methods as discussed in chapter 4. Chapter 5 investigated alternative techniques to determine a real or virtual reference plane (see table 5.1). Therefore, the reference plane approach is applicable to scenarios where no real reference plane is visible. We demonstrated experimentally in chapter 6 that our novel algorithms can be used to reconstruct difficult reference plane configurations where general (non reference plane) reconstruction methods fail. The question of critical reference plane configurations was addressed in chapter 7. It turned out that there are fewer critical configurations than in the general case, which has an important practical impact for scenes with one dominant plane. Finally, chapter 8 introduced a completely automatic multi-view reconstruction system using the reference plane approach. A summary of each individual chapter, including the main contributions and possible extensions, is given below.

In chapter 3, general configurations were compared with reference plane configurations of multiple views and features like points, lines and planes. The relationship between cameras and 3D features is *bi-linear* in the general case. If a reference plane is known, this relationship becomes *linear* in an affine space where the reference plane represents the plane at infinity. This makes it possible to reconstruct *all* cameras and *all* features (points, lines, and planes) in a *single* linear system simultaneously. *This novel approach represents the main contribution of the thesis*. The discussion for points and planes is based

on (Rother and Carlsson, 2001; Rother and Carlsson, 2002b; Rother et al., 2002; Rother and Carlsson, 2002a). Additionally, the linear system permits the simple incorporation of incidence relationships, such as a point lies on a plane, and constraints concerning known 3D features, for instance the coordinates of a 3D line being known. Furthermore, we have seen that with a known reference plane, i.e. the plane at infinity, the orientation of 3D features may be determined directly. Consequently, a 3D line is represented by 2 parameters (4 in general) and a 3D plane by one parameter (3 in general). Moreover, this chapter reviewed 4 categories of solving the reconstruction problem (a) our direct reference plane approach, (b) camera constraints, (c) structure constraints and (d) factorization.

Chapter 4 reviewed and compared multi-view reconstruction methods for general and reference plane configurations. The discussion was based on several criteria which "real world" multi-view reconstruction systems have to fulfill. The methods were compared in terms of categories introduced in the previous chapter. The main conclusion of the comparative study is that each category has its advantages and drawbacks. Therefore, the decision of the best method is application dependent. For reference plane configurations, three point based methods were compared in detail (a) our direct reference plane method, (b) a camera constraint method by Hartley et al. (2001) and (c) a factorization method by Triggs (2000). All three methods reconstruct the scene in closed-form from a singular value decomposition of a measurement matrix. Our method and the factorization method reconstruct both 3D features and cameras simultaneously. In contrast to this, the camera constraint method determines *only* the cameras simultaneously. The main drawback of our method is that features on the reference plane have to be reconstructed separately. Note that this is not a problem if the reference plane is the actual plane at infinity. The main disadvantage of the factorization method is that it is *not* applicable for infinite reference planes, and missing data is not treated naturally.

Chapter 5 investigated alternative techniques for determining the infinite homographies, induced by a real or virtual reference plane. These making use of a real scene plane, orthogonal scene directions, cameras with parallel projection or cameras with known epipolar geometry (see table 5.1). The main contribution (Rother and Carlsson, 2002b) of this chapter is twofold. First, we unify these different approaches of determining the infinite homography with the term *reference plane*. Secondly, we point out that both *real* and *virtual* reference plane configurations can be reconstructed with our direct reference plane approach. Consequently, the reference plane approach is applicable in many scenarios where no real reference plane is visible. A further contribution is a method to compute simultaneously the infinite homographies from known epipolar geometry. We do not claim that the list of alternative techniques in table 5.1 is complete. Probably, there are further alternatives which might have an important practical impact. For instance, is it possible to exploit symmetry properties or the contours of an objects? More generally, might it be enough to know that an object belongs to a certain class with some "geometric" properties? These are interesting open questions for future research.

Chapter 6 outlined practical algorithms of our novel direct reference plane approach for points, lines and planes. The methods for points and lines were extended to use normalized image points and general cameras, which did not appear in any of our previous publications. The main contribution of this chapter is the experimental comparison of

these methods with other approaches under various conditions using real and synthetic data. The experimental study focused on point features. The main observation is that for difficult, reference plane scenarios with a high percentage of missing data, up to 90%, our direct reference plane method performed successfully, where general reconstruction methods fail. The "failure" of these methods had the following reasons (a) too few image measurements are available, (b) error accumulation due to noisy image measurements or (c) critical configurations (dominant scene plane). Reference plane methods circumvent all these problems since they exploit a real or virtual reference plane visible in all views. Our method was significantly superior to Hartley et al.'s (2001) reference plane approach for some difficult scenarios. A further important though unsurprisingly result is that reference plane methods are inferior to general methods if the reference plane is detected very inaccurately. Synthetic experiments showed that our method and Hartley et al.'s (2001) method were very stable if the 3D scene points are *not close* to the reference plane, such as an infinite reference plane. For "flat scenes" where many 3D points are on or close to the reference plane, our method and the factorization approach of Triggs (2000) performed best. However, in this case our method is complex and iterative, depending on the number of 3D points. The factorization method has the drawback of "hallucinating" missing data. Consequently, in our opinion the best reference plane method for "flat scenes" has not yet been found. The main goal of the experiments for lines and planes was to demonstrate that they perform successfully under real world conditions. For planes, it turned out that our direct reference plane method is slightly inferior to methods which hallucinate image points given the planar homographies. In future work we plan to study experimentally our method for combinations of points, lines and planes, including additional scene constraints.

In chapter 7 we investigated critical reference plane configurations which do not have a unique projective reconstruction. All presented results are novel (Rother and Carlsson, 2002a) since to our knowledge critical configurations have only been studied for the general case (e.g. Kahl et al., 2001). The main contribution is that for multiple views and no missing data, i.e. all points are visible in all views, all non-trivial configurations where points and camera centres are non-coplanar are non-critical. This is an important result since the scenario of one dominant scene plane visible in multiple views appears frequently in practice and is critical in the general case. In the reference plane case this scenario is not critical if the scene plane and the reference plane are different, for instance the reference plane is the actual plane at infinity. Furthermore, we introduced a novel method to construct non-critical configurations for the case of missing data.

Finally, chapter 8 introduced an automatic multi-view reconstruction system for point features. The main contribution is to demonstrate that the reference plane approach has the capability of performing automatically the complete reconstruction task. This includes two novel methods, vanishing point detection (Rother, 2000; Rother, 2002) and robust multi-view point matching using a reference plane, which is based on (Tell and Carlsson, 2002). Possible future work will include further experiments, the extension to lines and planes and the automatic extraction of textured, virtual models from the 3D point reconstruction. Furthermore, an automatic reconstruction system using a real scene plane could exploit this finite reference plane for self-calibration.

9.2 Discussion

Probably the previous section gave the impression that 3D reconstruction is the "most important" task in computer vision and that this task is basically solved. Well, let us clarify this aspect by moving to a more "neutral" perspective.

The thesis discussed the specific task of feature based 3D reconstruction using a reference plane. The input data is a set of 2D images and the output a (metric) reconstruction of features and cameras in the 3D space. This process may be performed automatically for certain scenarios as demonstrated in chapter 8. Certainly, this is an important result and probably sufficient for applications like visualizing of man-made environments, robotics and augmented and virtual reality. However, from a broader perspective, how does our reconstruction system compares to a complete system that "sees". For comparison we choose the human visual system[3].

The *input data*, a sequence/set of 2D images, is similar in both systems. This may be assumed if only one eye is considered (monocular vision) and effects like the spatial arrangements of rods and cones are neglected. Chapter 1 reviewed many different cues to derive 3D information from the images, like motion, shading, parallel lines or familiar objects (see table 1.1). Our system exploits two sources of information to solve the reconstruction task: the camera's motion and a reference plane. As discussed in chapter 1, humans are not that limited. Psychophysicists analyze human behavior in simulated naturalistic virtual environments which include a combination of information sources (Gibson, 1950; Palmer, 1999). We may conjecture that humans use different combinations of information sources depending on the observed environment. In the field of computer vision the wide range of information sources is well known. This is due to the pioneer work of Marr (1982) and his colleagues, e.g. Ullman (1979), who were inspired by Gibson (1950). The 3D reconstruction task is also called *structure-from-X*, where *X* is a variable which represents any source of information like motion or shading. However, in our opinion the combination of information cues, i.e. structure-from-X *and* Y, is less frequently studied. The thesis demonstrated that a reference plane reconstruction system, which combines two sources (motion and reference planes), is significantly superior to general reconstruction systems, which exploits motion as the only source. Naturally the question arises if other information sources might be combined to give an improved reconstruction system.

Consider the *output* of both reconstruction systems? Our method returns the *exact* 3D position of the features and the observer in some (metric) space. What is the output data of the human system? Consider the experiment in chapter 1 of drawing a map of the city hall in Stockholm from a set of views (fig. 1.1). The result would quite certainly be less accurate compared to our method (fig. 1.2). However, is this task really relevant for us? How often do we have to answer the question: What is the height of the tower of the city hall? Humans have to solve a wide variety of *high level tasks* which involve 3D reconstruction (depth estimation), like picking up an object or walking through a corridor. In order to complete a high level task, some *low level tasks* have to be solved. The low level task of 3D

[3] A comparison with other biological "seeing systems" would be interesting as well.

reconstruction is one of many, like segmentation of an object in the image, object recognition and categorization. If we consider 3D reconstruction in more detail, two different tasks may be identified: *qualitative* and *quantitative* reconstruction. A quantitative reconstruction comprises of metric measurements, like the height of the tower of the city hall. A qualitative reconstruction only reflects the depth ordering of objects, for instance the top of the tower is farther away than the bottom for an observer on the ground. Some of the information sources in table 1.1 provide quantitative information, like motion, and others more qualitative information, like shading. In our opinion, most of the research in computer vision is focused on the quantitative reconstruction task, like the system presented in this thesis. Obviously, a qualitative reconstruction may be derived from a quantitative reconstruction. However, is a quantitative reconstruction always needed and useful? Consider the human system. What high level tasks must a human solve, and which low levels tasks are necessary to complete a certain high level task? Moreover, how do the low level tasks interact to complete a high level task? For example, in a scene with two objects, the observer has to pick the closer of the two. This high level task involves two low levels tasks, object segmentation (recognition) and qualitative reconstruction (depth ordering). If the objects are already segmented in the image, the qualitative reconstruction task is significantly simpler. On the other hand, additional knowledge about the depth ordering of image parts (pixels) is very useful information for the segmentation task. To summarize, the understanding and imitation of low level tasks is undoubted an important problem in computer vision. However, the key for imitating the human visual system is not merely the tasks themselves but just as importantly their interaction and correct combination.

Bibliography

Antone, M. and Teller, S. (2002). Scalable extrinsic calibration of omni-directional image networks, *International Journal of Computer Vision* **4**9(2/3): 143–174.

Avidan, S. and Shashua, A. (1998). Threading fundamental matrices, *European Conference on Computer Vision*, Freiburg, Germany, pp. I: 124–140.

Baillard, C. and Zisserman, A. (1999). Automatic reconstruction of piecewise planar models from multiple views, *IEEE Computer Vision and Pattern Recognition*, Fort Collins, Colorado, pp. 559–565.

Barnard, S. T. (1983). Interpreting perspective images, *Artificial Intelligence* **2**1: 435–462.

Bartoli, A., Sturm, P. and Horaud, R. (2001a). Constrained structure and motion from n views of a piecewise planar scene, *International Symposium on Virtual and Augmented Architecture*, Dublin, Ireland, pp. 195–206.

Bartoli, A., Sturm, P. and Horaud, R. (2001b). Projective structure and motion from two views of a piecewise planar scene, *International Conference on Computer Vision*, Vancouver, Canada, pp. 593–598.

Beardsley, P., Torr, P. H. S. and Zisserman, A. (1996). 3D model acquisition from extended image sequences, *European Conference on Computer Vision*, Cambridge, UK, pp. 683–695.

Beardsley, P., Zisserman, A. and Murray, D. W. (1994). Navigation using affine structure and motion, *European Conference on Computer Vision*, Stockholm,Sweden, pp. 85–96.

Bergen, J. R., Anandan, P., Hanna, K. J. and Hingorani, R. (1992). Hierarchical model-based motion estimation, *European Conference on Computer Vision*, Santa Margherita Ligure, Italy, pp. 237–252.

Bondyfalat, D., Papadopoulo, T. and Mourrain, B. (2001). Using scene constraints during the calibration procedure, *International Conference on Computer Vision*, Vancouver, Canada, pp. 124–130.

Boufama, B. and Mohr, R. (1995). Epipole and fundamental matrix estimation using the virtual parallax property, *International Conference on Computer Vision*, Cambridge, MA, pp. 1030–1036.

Bretzner, L. and Lindeberg, T. (1998). Use your hand as a 3-d mouse or relative orientation from extended sequences of sparse point and line correspondences using the affine trifocal tensor, *European Conference on Computer Vision*, Freiburg, Germany, pp. I: 141–157.

Brillault-O'Mahony, B. (1991). New method for vanishing point detection, *Computer Vision, Graphics, and Image Processing* **5**4(2): 289–300.

Canoma (n.d.). http://www.canoma.org.

Caprile, B. and Torre, V. (1990). Using vanishing points for camera calibration, *International Journal of Computer Vision* **4**: 127–139.

Carlsson, S. (1995). Duality of reconstruction and positioning from projective views, *in* P. Anandan (ed.), *IEEE Workshop on Representation of Visual Scenes*, Boston, USA.

Carlsson, S. and Eklundh, J. (1990). Object detection using model based prediction and motion parallax, *European Conference on Computer Vision*, Antibes, France, pp. 297–306.

Carlsson, S. and Weinshall, D. (1998). Dual computation of projective shape and camera positions from multiple images, *International Journal of Computer Vision* **2**7(3): 227–241.

Coughlan, J. M. and Yuille, A. L. (1999). Manhattan world: Compass direction from a single image by bayesian inference, *International Conference on Computer Vision*, Kerkyra, Greece, pp. 941–947.

Criminisi, A., Reid, I. and Zisserman, A. (1998). Duality, rigidity and planar parallax, *European Conference on Computer Vision*, Freiburg, Germany, pp. 846–861.

Cross, G., Fitzgibbon, A. W. and Zisserman, A. (1999). Parallax geometry of smooth surfaces in multiple views, *International Conference on Computer Vision*, Kerkyra, Greece, pp. 323–329.

Debevec, P. E., Taylor, C. J. and Malik, J. (1996). Modelling and rendering architecture from photographs, *ACM Computer Graphics (Proceedings SIGGRAPH)* pp. 11–20.

Deutscher, J., Isard, M. and MacCormick, J. (2002). Automatic camera calibration from a single manhattan image, *European Conference on Computer Vision*, Copenhagen, Denmark, pp. 175–188.

Devernay, F. and Faugeras, O. D. (1995). Automatic calibration and removal of distortion from scenes of structured environments, *SPIE*, San Diego, CA.

Faugeras, O. D. (1992). What can be seen in three dimensions with an uncalibrated stereo rig?, *European Conference on Computer Vision*, Santa Margherita Ligure, Italy, pp. 563–578.

Faugeras, O. D. (1993). *Three-Dimensional Computer Vision: a Geometric Viewpoint*, The MIT Press.

Faugeras, O. D. (1995). Stratification of three-dimensional vision: projective, affine, and metric representation, *Journal of the Optical Society of America* **A**12: 465–484.

Faugeras, O. D., Luong, Q. and Maybank, S. (1992). Camera self-calibration: Theory and experiments, *European Conference on Computer Vision*, Santa Margherita Ligure, Italy, pp. 321–334.

Faugeras, O. D. and Luong, Q.-T. (2001). *The Geometry of Multiple Images*, The MIT Press.

Faugeras, O. D. and Mourrain, B. (1995). On the geometry and algebra of point and line correspondences between n images, *International Conference on Computer Vision*, Cambridge, MA, pp. 951–962.

Fischler, M. and Bolles, R. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, *Communications of the Association for Computing Machinery (ACM)* **2**4(6): 381–385.

Fitzgibbon, A. W. and Zisserman, A. (1998). Automatic camera recovery for closed or open image sequences, *European Conference on Computer Vision*, Freiburg, Germany, pp. 311–326.

Georgescu, B. and Meer, P. (2002). Balanced recovery of 3D structure and camera motion from uncalibrated image sequences, *European Conference on Computer Vision*, Copenhagen, Denmark, pp. 294–308.

Gibson, J. J. (1950). *The perception of the visual world*, Boston: Houghton Mifflin.

Golub, G. H. and Van Loan, C. F. (1996). *Matrix Computations*, third edn, The John Hopkins University Press, Baltimore, MD.

Harris, C. and Stephens, M. (1988). A combined corner and edge detector, *Alvey vision conference*, pp. 147–151.

Hartley, R. I. (1992). Estimation of relative camera positions for uncalibrated cameras, *European Conference on Computer Vision*, Santa Margherita Ligure, Italy, pp. 579–587.

Hartley, R. I. (1994). Projective reconstruction from line correspondence, *IEEE Computer Vision and Pattern Recognition*, Seattle,WA.

Hartley, R. I. (1995). A linear method for reconstruction from lines and points, *International Conference on Computer Vision*, Cambridge, MA, pp. 882–887.

Hartley, R. I. (1997). In defence of the 8-point algorithm, *IEEE Trans. Pattern Analysis and Machine Intelligence* **1**9(6): 580–593.

Hartley, R. I. (2000). Ambiguous configurations for 3-view projective reconstruction, *European Conference on Computer Vision*, Dublin, Ireland, pp. 922–935.

Hartley, R. I., Dano, N. and Kaucic, R. (2001). Plane-based projective reconstruction, *International Conference on Computer Vision*, Vancouver, Canada, pp. 420–427.

Hartley, R. I. and Debunne, G. (1998). Dualizing scene reconstruction algorithms, *in* R. Koch and L. Van Gool (eds), *Workshop on 3D Structure from Multiple Images of Large-scale Environments (SMILE)*, LNCS 1506, Springer-Verlag, Freiburg, Germany, pp. 14–31.

Hartley, R. I. and Zisserman, A. (2000). *Multiple View Geometry in Computer Vision*, Cambridge University Press.

Heyden, A. (1998). Algebraic varieties in multiple view geometry, *European Conference on Computer Vision*, Freiburg, Germany, pp. 3–19.

Heyden, A. and Åström, K. (1995a). A canonical framework for sequences of images, *in* P. Anandan (ed.), *IEEE Workshop on Representation of Visual Scenes*, Boston, USA.

Heyden, A. and Åström, K. (1995b). Simplifications of multilinear forms for sequences of images, *Image and Vision Computing* **1**5(10): 749–757.

Heyden, A., Berthilsson, R. and Sparr, G. (1999). An iterative factorization method for projective structure and motion, *Image and Vision Computing* **1**7(13): 981–991.

Heyden, A. and Kahl, F. (2000). Direct affine reconstruction, *International Conference on Pattern Recognition*, Barcelona, Spain, pp. 885–888.

Irani, M. and Anandan, P. (1996). Parallax geometry of pairs of points for 3D scene analysis, *European Conference on Computer Vision*, Cambridge, UK, pp. 17–30.

Irani, M. and Anandan, P. (1999a). About direct methods, *Workshop on Vision Algorithms*, Kerkyra, Greece, pp. 267–278.

Irani, M. and Anandan, P. (1999b). Direct recovery of planar-parallax from multiple frames, *Workshop on Vision Algorithms*, Kerkyra, Greece, pp. 85–99.

Irani, M. and Anandan, P. (2000). Factorization with uncertainty, *European Conference on Computer Vision*, Dublin, Ireland, pp. 539–553.

Irani, M., Anandan, P. and Weinshall, D. (1998). From reference frames to reference planes: Multi-view parallax geometry and applications, *European Conference on Computer Vision*, Freiburg, Germany, pp. 829–845.

Irani, M., Hassner, T. and Anandan, P. (2002). What does the scene look like from a scene point?, *European Conference on Computer Vision*, Copenhagen, Denmark, pp. 883–898.

Isard, M. and Blake, A. (1998). Condensation – conditional density propagation for visual tracking, *International Journal of Computer Vision* **2**9(1): 5–28.

Jacobs, D. (1997). Linear fitting with missing data for structure-from-motion, *IEEE Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, pp. 206–212.

Johansson, B. (1999). View synthesis and 3D reconstruction of piecewise planar scenes using intersection lines between the planes, *International Conference on Computer Vision*, Kerkyra, Greece, pp. 54–59.

Kahl, F., Hartley, R. and Åström, K. (2001). Critical configurations for n-view projective reconstruction, *IEEE Computer Vision and Pattern Recognition*, Hawaii, USA, pp. II:158–163.

Kahl, F. and Hartley, R. I. (2002). Critical curves and surfaces for euclidean reconstruction, *European Conference on Computer Vision*, Copenhagen, Denmark, pp. II: 447–462.

Kahl, F. and Heyden, A. (1999). Affine structure and motion from points, lines and conics, *International Journal of Computer Vision* **3**3(3): 163–180.

Klein, F. (1893). Vergleichende Betrachtung über neuere geometrische Forschungen (Erlangen), *Mathematische Annalen (reprint)* **4**3.

Koenderink, J. J. and Doorn, A. J. (1991). Affine structure from motion, *Journal of the Optical Society of America* **8**(2): 377–385.

Koseká, J. and Zhang, W. (2002). Video compass, *European Conference on Computer Vision*, Copenhagen, Denmark, pp. 476–490.

Krames, J. (1942). Über die bei der Hauptaufgabe der Luftphotogrammetrie auftretenden "gefährlichen" Flächen, *Bildmessung und Luftbildwesen* **1**7(Heft 1/2): 1–18.

Kumar, R., Anandan, P. and Hanna, K. (1994). Direct recovery of shape from multiple views: a parallax based approach, *International Conference on Pattern Recognition*, Jerusalem, Israel, pp. 685–688.

Kumar, R., Anandan, P., Irani, M., Bergen, J. and Hanna, K. (1995). Representation of scenes from collections of images, *in* P. Anandan (ed.), *IEEE Workshop on Representation of Visual Scenes*, Boston, USA.

Liebowitz, D. (2001). *Camera Calibration and Reconstruction of Geometry*, PhD thesis, University of Oxford, UK.

Liebowitz, D. and Zisserman, A. (1999). Combining scene and auto-calibration constraints, *International Conference on Computer Vision*, Kerkyra, Greece, pp. 293–300.

Longuet-Higgins, H. (1981). A computer algorithm for reconstructing a scene from two projections, *Nature* **2**93: 133–135.

Luong, Q.-T. and Faugeras, O. D. (1993). Determing the fundamental matrix with planes, *IEEE Computer Vision and Pattern Recognition*, New York, NY, pp. I:489–494.

Luong, Q.-T. and Viéville, T. (1996). Canonical representations for the geometries of multiple projective views, *Computer Vision and Image Understanding* **4**18: 1–15.

Marr, D. (1982). *Vision: a computational investigation into the human representation and processing of visual information*, W. H. Freeman and Comapny.

Martinec, D. and Pajdla, T. (2002). Structure from many perspective images with occlusions, *European Conference on Computer Vision*, Copenhagen, Denmark, pp. 355–369.

Maybank, S. J. (1992). *Theory of Reconstruction from Image Motion*, Springer-Verlag.

Van Gool, L., Moons, T., Proesmans, M. and Van Diest, M. (1994). Affine reconstruction from perspective image pairs obtained by a translating camera, *International Conference on Pattern Recognition*, Jerusalem, Israel, pp. 290–294.

Morris, D. D. and Kanade, T. (1998). A unified factorization algorithm for points, line segments and planes with uncertainty models, *International Conference on Computer Vision*, Bombay, India.

Navab, N., Genec, Y. and Appel, M. (2000). Lines in one orthograpophic and two perspective views, *IEEE Computer Vision and Pattern Recognition*, Hilton Head Island, South Carolina, pp. 607–616.

Nistér, D. (2000a). Frame decimation for structure and motion, *Workshop on 3D Structure from Multiple Images of Large-scale Environments (SMILE)*, Springer-Verlag, Dublin, Ireland, pp. 2–9.

Nistér, D. (2000b). Reconstruction from uncalibrated sequences with a hierachy of trifocal tesors, *European Conference on Computer Vision*, Dublin, Ireland, pp. 649–662.

Nistér, D. (2001). *Automatic Dense Reconstruction from Uncalibrated Video Sequences*, PhD thesis, KTH, Stockholm, Sweden. ISSN 1101-2250.

Oliensis, J. (1995). Multiframe structure from motion in perspective, *in* P. Anandan (ed.), *IEEE Workshop on Representation of Visual Scenes*, Boston, USA.

Oliensis, J. (1999). A multi-frame structure-from-motion algorithm under perspective projection, *International Journal of Computer Vision* **3**4(2/3): 163–192.

Oliensis, J. and Genc, Y. (1999). Fast algorithms for projective multi-frame structure from motion, *International Conference on Computer Vision*, Kerkyra, Greece, pp. 536–542.

Oskarsson, M., Åström, K. and Overgaard, N. C. (2001). Classifying and solving minimal structure and motion problems with missing data, *International Conference on Computer Vision*, Vancouver, Canada, pp. 628–634.

Palmer, S. E. (1999). *Vision Science Photons to Phenomenology*, The MIT Press.

Poelman, C. and Kanada, T. (1994). A paraperspective factorization method for shape and motion recovery, *European Conference on Computer Vision*, Stockholm,Sweden, pp. II:97–108.

Pollefeys, M. (1999). *Theory of Reconstruction from Image Motion*, PhD thesis, K. U. Leuven. ISBN 90-5682-193-8.

Pollefeys, M., Koch, R. and Van Gool, L. (1998). Self calibration and metric reconstruction in spite of varying and unknown internal camera parameters, *International Conference on Computer Vision*, Bombay, India, pp. 90–96.

Pollefeys, M., Verbiest, F. and Van Gool, L. (2002). Surviving dominant planes in uncalibrated structure and motion recovery, *European Conference on Computer Vision*, Copenhagen, Denmark, pp. 837–851.

Press, W., Flannery, B., Teukolsky, S. and Vetterling, W. (1988). *Numerical Recipes in C*, Cambridge University Press.

Qian, C. and Medioni, G. (1999). Efficient iterative solution to m-view projective reconstruction problem, *IEEE Computer Vision and Pattern Recognition*, Fort Collins, Colorado, pp. 55–61.

Quan, L., Heyden, A. and Kahl, F. (1994). Invariants of 6 points from 3 uncalibrated images, *European Conference on Computer Vision*, Stockholm, Sweden, pp. 450–470.

Quan, L., Heyden, A. and Kahl, F. (1999). Minimal projective reconstruction with missing data, *IEEE Computer Vision and Pattern Recognition*, Fort Collins, Colorado, pp. 210–216.

Quan, L. and Kanade, T. (1997). Affine structure from line correspondences with uncalibrated affine cameras, *IEEE Trans. Pattern Analysis and Machine Intelligence* **1**9(8): 834–845.

Quan, L. and Mohr, R. (1989). Determining perspective structures using hierarchical Hough transform, *Pattern Recognition Letters* **9**: 279–286.

Reid, I. D. and Murray, D. W. (1996). Active tracking of foveated feature clusters using affine structure, *International Journal of Computer Vision* **1**8(1): 41–60.

Robertson, D. P. and Cipolla, R. (2000). An interactive system for constraint-based modelling, *British Machine Vision Conference*, Bristol, UK, pp. 536–545.

Robertson, D. P. and Cipolla, R. (2002). Building architectural models from many views using map constraints, *European Conference on Computer Vision*, Copenhagen, Denmark, pp. 155–169.

Rosin, P. and West, G. (1989). Segmentation of edges into lines and arcs, *Image and Vision Computing* **7**: 109–114.

Rother, C. (2000). A new approach for vanishing point detection in architectural environments, *British Machine Vision Conference*, Bristol, UK, pp. 382–391.

Rother, C. (2002). A new approach to vanishing point detection in architectural environments, *Image and Vision Computing* **2**0(9-10): 647–656.

Rother, C. and Carlsson, S. (2001). Linear multi view reconstruction and camera recovery, *International Conference on Computer Vision*, Vancouver, Canada, pp. 42–51.

Rother, C. and Carlsson, S. (2002a). Linear multi view reconstruction and camera recovery using a reference plane, *International Journal of Computer Vision* **4**9(2/3): 117–141.

Rother, C. and Carlsson, S. (2002b). Linear multi view reconstruction with missing data, *European Conference on Computer Vision*, Copenhagen, Denmark, pp. II: 309–324.

Rother, C., Carlsson, S. and Tell, D. (2002). Projective factorization of planes and cameras in multiple views, *International Conference on Pattern Recognition*, Quebec, Canada, pp. 737–740.

Sawhney, H. S. (1994). 3d geometry from planar parallax, *IEEE Computer Vision and Pattern Recognition*, Seattle,WA, pp. 929–934.

Schaffalitzky, F., Zisserman, A., Hartley, R. I. and Torr, P. H. S. (2000). A six point solution for structure and motion, *European Conference on Computer Vision*, Dublin, Ireland, pp. 632–648.

Schmid, C. and Zisserman, A. (2000). The geometry and matching of lines and curves over multiple views, *International Journal of Computer Vision* **4**0(3): 199–233.

Semple, J. and Kneebone, G. (1952). *Algebraic projective geometry*, Oxford University Press.

Shashua, A. (1994). Trilenarity in visual recognition by alignment, *European Conference on Computer Vision*, Stockholm,Sweden, pp. I:479–484.

Shashua, A. and Avidan, S. (1996). The rank 4 constraint in multiple ($\geq$ 3) view geometry, *European Conference on Computer Vision*, Cambridge, UK, pp. 196–206.

Shashua, A. and Navab, N. (1994). Relative affine structure: Theory and application to 3D reconstruction from perspective views, *IEEE Computer Vision and Pattern Recognition*, Seattle,WA, pp. 483–489.

Shum, H.-Y., Han, M. and Szeliski, R. (1998). Interactive construction of 3D models from panoramic mosaics, *IEEE Computer Vision and Pattern Recognition*, Santa Barbara, pp. 427–433.

Slama, C. (1980). *Manual of Photogrammetry*, fourth edn, American Society of Photogrammetry, Falls Church, VA, USA.

Sparr, G. (1994). A common framework for kinetic depth, motion and reconstruction, *European Conference on Computer Vision*, Stockholm,Sweden, pp. 471–482.

Sparr, G. (1996). Simultaneous reconstruction of scene structure and camera locations from uncalibrated image sequences, *International Conference on Pattern Recognition*, Vienna, Austria, pp. 328–333.

Springer, C. (1964). *Geometry and analysis of projective spaces*, Freeman.

Sturm, P. (2000). Algorithms for plane-based pose estimation, *IEEE Computer Vision and Pattern Recognition*, Hilton Head Island, South Carolina, pp. 706–711.

Sturm, P. and Maybank, S. J. (1999). A method for interactive 3D reconstruction of piecewise planar objects from single images, *British Machine Vision Conference*, Nottingham, UK, pp. 265–274.

Sturm, P. and Triggs, B. (1996). A factorization based algorithm for multi-image projective structure and motion, *European Conference on Computer Vision*, Cambridge, UK, pp. 709–719.

Svedberg, D. and Carlsson, S. (1999). Calibration, pose and novel views from single images of constrained scenes, *Scandanavian Conference on Image Analysis*, Kangerlussuaq, Greenland, pp. 111–118.

Szeliski, R. and Torr, P. H. S. (1998). Geometrically constrained structure from motion: Points and planes, *in* R. Koch and L. Van Gool (eds), *Workshop on 3D Structure from Multiple Images of Large-scale Environments (SMILE)*, LNCS 1506, Springer-Verlag, Freiburg, Germany, pp. 171–186.

Tell, D. (2002). *Wide Baseline Matching with Applications to Visual Servoing*, PhD thesis, KTH, Stockholm, Sweden. ISBN 91-7283-254-1.

Tell, D. and Carlsson, S. (2002). Combining topology and appearance for wide baseline matching, *European Conference on Computer Vision*, Copenhagen, Denmark, pp. I: 68–81.

Thorhallsson, T. and Murray, D. W. (1999). The tensors of three affine views, *IEEE Computer Vision and Pattern Recognition*, Fort Collins, Colorado, pp. 450–456.

Tomasi, C. and Kanade, T. (1992). Shape and motion from image streams under orthography: a factorization method, *International Journal of Computer Vision* **9**(2): 137–154.

Torr, P. H. S. (1995). *Motion segmentation and outlier detection*, PhD thesis, Dept. of Engineering Science, Univ. of Oxford.

Torr, P. H. S. and Murray, D. W. (1997). The development and comparison of robust methods for estimating the fundamental matrix, *International Journal of Computer Vision* **2**4(3): 271–300.

Torr, P. H. S. and Zisserman, A. (1997). Robust parametrization and computation of the trifocal tensor, *Image and Vision Computing* **1**5: 591–605.

Triggs, B. (1995). Matching constraints and the joint image, *International Conference on Computer Vision*, Cambridge, MA, pp. 338–343.

Triggs, B. (1996). Factorization methods for projective structure and motion, *IEEE Computer Vision and Pattern Recognition*, San Francisco, CA, pp. 845–851.

Triggs, B. (1997a). Auto-calibration and the absolute quadric, *IEEE Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, pp. 609–614.

Triggs, B. (1997b). Linear projective reconstruction from matching tensors, *Image and Vision Computing* **1**5: 617–625.

Triggs, B. (2000). Plane + parallax, tensors and factorization, *European Conference on Computer Vision*, Dublin, Ireland, pp. 522–538.

Triggs, B., McLauchlan, P., Hartley, R. and Fitzgibbon, A. (1999). Bundle adjustment – a modern synthesis, *Workshop on Vision Algorithms*, Kerkyra, Greece, pp. 298–376.

Tuytelaars, T., Van Gool, L., Proesmans, M. and Moons, T. (1998). The cascaded Hough transform as an aid in aerial image interpretation, *International Conference on Computer Vision*, Bombay, India, pp. 67–72.

Ullman, S. (1979). *The Interpretation of Visual Motion*, The MIT Press.

van den Heuvel, F. A. (1998). Vanishing point detection for architectural photogrammetry, *International Archives of Photogrammetry and Remote Sensing* **X**XXII(5): 652–659.

Vision and Modelling of Dynamic Scenes (2002). Copenhagen, Denmark.

Weinshall, D., Anandan, P. and Irani, M. (1998). From ordinal to euclidean reconstruction with partial scene calibration, *in* R. Koch and L.Van Gool (eds), *Workshop on 3D Structure from Multiple Images of Large-scale Environments (SMILE)*, LNCS 1506, Springer-Verlag, Freiburg, Germany, pp. 208–223.

Werner, T. and Zisserman, A. (2002). New techniques for automated architectural reconstruction from photographs, *European Conference on Computer Vision*, Copenhagen, Denmark, pp. 541–555.

Xu, G., Terai, J. and Shum, H. (2000). A linear algorithm for camera self-calibration, motion and structure recovery for multi-planar scenes from two perspective images, *IEEE Computer Vision and Pattern Recognition*, Hilton Head Island, South Carolina, pp. 474–479.

Zelnik-Manor, L. and Irani, M. (1999). Multi-view subspace constraints on homographies, *International Conference on Computer Vision*, Kerkyra, Greece, pp. 710–715.

Zucchelli, M. (2002). *Optical Flow Based Structure from Motion*, PhD thesis, KTH, Stockholm, Sweden. ISBN 91-7283-308-4.