Extracting 3D Scene-consistent Object Proposals and Depth from Stereo Images

Michael Bleyer[1⋆], Christoph Rhemann[1⋆,2], and Carsten Rother[2]

[1] Vienna University of Technology, Vienna, Austria
[2] Microsoft Research Cambridge, Cambridge, UK

Abstract. This work combines two active areas of research in computer vision: unsupervised object extraction from a single image, and depth estimation from a stereo image pair. A recent, successful trend in unsupervised object extraction is to exploit so-called "3D scene-consistency", that is enforcing that objects obey underlying physical constraints of the 3D scene, such as occupancy of 3D space and gravity of objects. Our main contribution is to introduce the concept of 3D scene-consistency into stereo matching. We show that this concept is beneficial for both tasks, object extraction and depth estimation. In particular, we demonstrate that our approach is able to create a large set of 3D scene-consistent object proposals, by varying e.g. the prior on the number of objects. After automatically ranking the proposals we show experimentally that our results are considerably closer to ground truth than state-of-the-art techniques which either use stereo or monocular images. We envision that our method will build the front-end of a future object recognition system for stereo images.

1 Introduction

The use of cameras which can capture 3D information has increased tremendously in the last years and enabled impressive systems in computer vision, robotics, human computer interaction, e.g. [2], and other domains. In this work we assume that we have as input a single shot from a passive-stereo camera, such as the commercial FujiFilm FinePix 3D or the new LG Optimus mobile phone. The goal of this work is to automatically extract jointly the scene depth as well as all objects present in the scene. Such an output can then be fed into other systems, e.g. for object recognition or augmented reality, as discussed later. Furthermore, we do not assume that the images were captured in a certain environment, such as in- or outdoor[1]. Our only assumption is that the scene is assembled of objects.

There is a large body of work which has tackled similar problems. If the object class is known, e.g. pedestrians or cars, impressive detection systems have been built. The performance of such systems improves further if prior knowledge of the (approximate) 2D or 3D shape of the object, or object class, is known. For the

⋆ This work was supported in part by the Vienna Science and Technology Fund (WWTF) under project ICT08-019.

[1] Passive stereo cameras, in contrast to active stereo or time-of-flight cameras, can capture high quality RGB images and operate in- and outdoor.

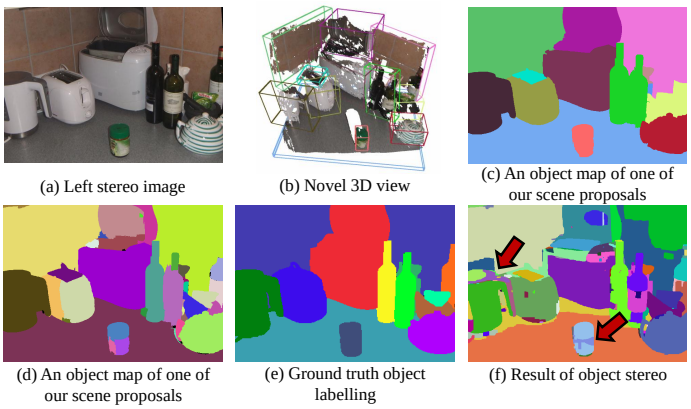

Fig. 1. Our approach. Given a stereo pair (a), our algorithm jointly estimates a 3D reconstruction (b) and object maps (c,d) using physics-based reasoning. The result is considerably closer to ground truth (e) than the one of object stereo [1] (f).

task of class independent object detection from stereo images, we are only aware of a few works. The approach which is closest to ours is object stereo [1] and we discuss it in detail later. Object stereo estimates jointly depth and objects, and is to our knowledge the only work which has shown a synergy effect between the two tasks. In this context, Lubor et al. [3] have shown a synergy between depth estimation and object-class extraction, which however relies on a-priori defined object classes. A different research direction is to solve the two tasks separately, that is objects are extracted based on a pre-computed depth image. For instance in [4], a system is developed for interactive foreground extraction in stereo images. In robotics, Björkman and Kragic [5, 6] have recently presented a system for automatic foreground extraction by combining color and stereo cues. As in our work, they perform a certain amount of 3D reasoning, e.g. by exploiting the knowledge of a flat 3D supporting surface and that objects are of an approximate 3D size. While these systems are a good step towards "3D reasoning", we believe that they have not yet exploited the full potential of it.

The main inspiration of our work stems from recent progress for the task of object extraction from a single image, e.g. [7–9][2]. It has been shown that results considerably improve when reasoning is not in 2D but in 3D, that is the objects live in 3D space and have to obey the implicit physical constraints and forces. In this work we call this concept "3D scene-consistency" and we formally define it later. While there has been a considerable amount of work on object extraction from a single image which exploits the concept of "3D scene-consistency", there have been rather few works, such as [5, 6, 1], with respect to stereo images. This is surprising, since stereo images provide an approximate depth and hence are an ideal input.

In the following, we review in detail the relationship to object stereo [1], which is closest to our work. Note that the focus in [1] was to show that depth estimation can be improved by introducing the notion of objects. In contrast, the

[2] See also ICCV'09/'11 workshop on 3D representation and recognition.

main focus of this work is on the extraction of objects. Given this focus, one has to ask the question: what will the retrieved objects be used for? We are inspired by the work of [10] which is the front-end of a system which won the PASCAL recognition and segmentation challenge in the last three years, see also [11, 8]. This work, and the companion work [12], propose the following 3-step pipeline: 1) generate a large pool of object proposals; 2) rank the proposals according to a learned objectness score; 3) perform object recognition on the top ranked object proposals. Our idea is to follow this line of research, but to build a new step 1) of their pipeline which takes stereo images as input, instead of a single image. Consider fig. 1. Given a stereo image (a) the goal is to estimate a pool of so-called **"scene proposals"**. A scene proposal consists of (1) a disparity map and (2) an object map that assigns each pixel to an object. Figs. 1(c) and (d) visualize two example object maps. A corresponding depth map for the object map in (c) is shown in (b). (To show the depth map we render the scene from a novel viewpoint, and the recovered objects are marked by 3D bounding boxes.) As in [10] we do not make a hard decision on what is the best solution, but we return a pool of scene proposals, which can then can be used for other tasks, such as object recognition. In fact we applied the automatic ranking technique (step 2) of [10] to all objects extracted from our scene proposals, and observed that we considerably outperform all state-of-the art techniques which use either monocular or stereo images in terms of segmentation accuracy.

Let us return to the comparison to object stereo [1]. A result of [1] is shown in fig. 1(f). Note that [1] did not introduce the concept of computing a pool of object maps. The *key difference* to our work is that objects in [1] were approximated by flat 2D planes, without any 3D extent. In contrast, we give objects a 3rd dimension by using an enclosing 3D bounding box. By doing so we can introduce and exploit physical constraints which are not possible in the flat "billboard" world of [1]. Note that this is analog to the case of 2D images, where [7] argues that it is important to change the typical surface-based representation to a 3D representation. Figure 1(b) illustrates our recovered 3D bounding boxes for the object map in (c). Let us discuss particular objects in more detail. The water kettle (as well as the toaster behind it) in fig. 1(f; left arrow) cannot be expressed by a flat 2D plane. Hence [1] cannot detect it as one object, in contrast to our result (c,d). An important physical constraint we add is to reason about occupancy in 3D space, i.e. bounding boxes should not (considerably) overlap in 3D space. Consider fig. 1(f, right arrow), where object stereo assigns the top and bottom part of the can to the same object, while the middle part is assigned to a different object. This volume intersection is physically very unlikely. Our results (c,d) are physically plausible. Another physical constraint we add is gravity which cannot be realized with a surface based representation [1].

2 Model

We now describe our model for jointly computing a disparity map and an object map. Let $\mathcal{I}$ denote all pixel coordinates of the left image. To estimate disparity we assign each pixel $p \in \mathcal{I}$ to a 3D plane. We compute a mapping $F : \mathcal{I} \rightarrow \mathcal{F}$

where $\mathcal{F}$ denotes the set of all possible 3D planes. Once F is known, we can compute a pixel p's disparity d_p using its plane f_p as $d_p := a_{f_p} p_x + b_{f_p} p_y + c_{f_p}$, where p_x and p_y are p's x- and y-coordinates and a_{f_p}, b_{f_p} and c_{f_p} are three parameters defining plane f_p. Hence, we also refer to F as the disparity map. Note that as an alternative to planes one could directly assign each pixel to a disparity. We opted for planes since this enables the application of a powerful local stereo algorithm, i.e., PatchMatch Stereo [13] that gives excellent results for highly slanted surfaces and sub-pixel disparities via the use of slanted planar support windows. As described below, our planes are used to model such slanted support windows. Note that our algorithm operates in metric space. To convert from disparity to metric measures the stereo system needs to be calibrated.[3]

Let us now discuss the object level that is used to compute an object segmentation and enables modeling physics-based constraints. We compute a second mapping $O : \mathcal{I} \rightarrow \mathcal{O}$ where $\mathcal{O}$ denotes the set of all objects. We refer to a mapping O as an object map. Here, an object is defined by two parameters: (1) an oriented 3D bounding box that is used as a proxy for the object's spatial extent in 3D space and (2) a color model for the object.[4]

A pair $\langle F, O \rangle$ forms a so-called scene proposal. To measure the quality of a scene proposal, we define an energy, that is subject to minimization, as

$$\begin{aligned} E(F,O) = E_{pc}(F) + E_{col}(O) + E_{ol}(O,F) + \\ E_{tight}(O) + E_{is}(O) + E_{gravity}(O) + E_{mdl}(O). \end{aligned} \tag{1}$$

The individual terms are explained next.

Photo Consistency Term E_{pc} The photo consistency term measures the quality of a disparity map by computing pixel dissimilarities of corresponding points in left and right images and is defined as in [13]:

$$E_{pc}(F) = \sum_{p \in \mathcal{I}} m(p, f_p). \tag{2}$$

Here, the function $m()$ computes aggregated matching costs, i.e., it performs the aggregation step that is the core of local stereo algorithms:

$$m(p, f_p) = \sum_{q \in W_p} w(p,q) \cdot \rho(q, q - (a_{f_p} q_x + b_{f_p} q_y + c_{f_p})), \tag{3}$$

where W_p is a squared window centered at pixel p. The function $w()$ assigns a support weight to each pixel within a support window, which implements the adaptive support weight idea [14] and considerably improves disparity results at depth discontinuities. It is defined as $w(p,q) = exp\{-\frac{||I_p - I_q||}{\gamma}\}$. Here, γ represents a parameter, $||I_p - I_q||$ denotes the L_1-distance of p and q's colors in RGB

[3] We use images for which the calibration parameters are unknown and approximate their values such that the 3D reconstruction looks reasonable.

[4] Note, pixels are assigned to the same object identity if they map to the same object bounding box and to the same object color model.

space. The function $\rho(q, q')$ computes the pixel dissimilarity between a pixel q of the left and a pixel q' of the right image. Our match measure computes a weighted sum of gradient and color differences, which is described in the supplementary material. All parameters of this term are set as in [13], i.e., the size of W_p is 35×35 pixels and $\gamma = 10$.

Color Term E_{col} As in [1], we prefer objects that are compact in color over those that are not. We model the color of an object using a Gaussian Mixture Model (GMM). Function $\pi(c, o)$ returns the probability that color c belongs to object o. We define the color term as

$$E_{col}(O) = \sum_{p \in \mathcal{I}} \sum_{q \in W_p} w(p, q) \cdot -\log(\pi(c_q, o_p)) \cdot \lambda_{color} \tag{4}$$

where λ_{color} is a penalty for color inconsistency.

Note that we aggregate the color costs over a small local window W_p centered at p. This means that the costs are averaged locally which acts as a local smoothness constraint. Similar to the photo-consistency term in eq. (3) this averaging is weighted according to function $w()$ such that we do not smooth over object boundaries. As shown in [15] such a local edge-preserving smoothing operation on the data costs gives results comparable to global edge-preserving smoothness terms, which are more difficult to optimize.

Bounding Box Outlier Term E_{ol} An object contains a 3D bounding box that is defined in metric 3D space. Our bounding boxes are oriented in order to compactly capture the spatial extent of arbitrarily oriented objects. A bounding box represents an approximation of an object's spatial extent and forms the basis for modeling physical constraints. The outlier term enforces 3D compactness of an object, i.e., the reconstructed 3D coordinates of all pixels assigned to an object have to lie within the object's bounding box. The object map O and disparity map F influence each other via this term. Formally, the term is defined as

$$E_{ol}(O, F) = \sum_{p \in \mathcal{I}} outsideBB(2D3D(p, f_p), o_p) \cdot \lambda_{outlier} \tag{5}$$

where $2D3D()$ is a function that maps pixel p to metric 3D space given its disparity plane f_p. The function $outsideBB(P, o_p)$ returns 1 if the 3D point P lies outside of object o_p's bounding box and 0 otherwise. ($\lambda_{outlier}$ should ideally be set to infinity, however, this is difficult to enforce in our optimization procedure. We therefore use a high constant value instead.)

Bounding Box Tightness Term E_{tight} This term ensures tightness of bounding boxes. It prevents bounding boxes from unnecessarily filling free space between objects. We impose a penalty on the volume of the bounding box:

$$E_{tight}(O) = \sum_{o \in O} volume(o) \cdot \lambda_{tight} \tag{6}$$

where the function $volume(o)$ returns the volume of object o's bounding box in m^3 and λ_{tight} is a penalty.

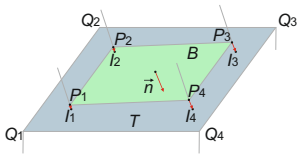

Fig. 2. Checking the relationship "on top of each other".

Bounding Box Intersection Term E_{is} This term is based on the observation that objects do usually not intersect each other. As stated above, we use a 3D bounding box as a proxy for the 3D spatial extent of an object. This term imposes a penalty if two bounding boxes intersect:

$$E_{is}(O) = \sum_{o_1 \in O} \sum_{o_2 \in O - \{o_1\}} intersection(o_1, o_2) \cdot \lambda_{intersect}. \tag{7}$$

Here, the function *intersection*() returns the bounding box intersection volume in m^3 and $\lambda_{intersect}$ is a penalty.

Gravity Term $E_{gravity}$ This term encodes the observation that objects usually do not float in the air. Due to gravity, objects typically stand on top of each other or on top of a ground plane. We implement this observation by encouraging bounding boxes to stand on top of each other:

$$E_{gravity}(O) = \sum_{o_1 \in O} \sum_{o_2 \in O - \{o_1\}} ontop(o_1, o_2) \cdot \lambda_{gravity}, \tag{8}$$

where $\lambda_{gravity}$ is a negative constant. The function $ontop(o_1, o_2)$ returns 1 if object o_1 stands on top of object o_2 and 0 otherwise and is explained as follows (also see fig. 2). We first extract the bottom surface B of o_1's bounding box and the top surface T of o_2's bounding box. We now project the bottom surface B onto the top surface T in direction of the normal vector $\vec{n}$ of B. The corner points of this projection are denoted as $\{I_1, \ldots, I_4\}$. We then check if the distances between the corner points $\{P_1, \ldots, P_4\}$ of surface B and their corresponding projections $\{I_1, \ldots, I_4\}$ are below a small threshold. If at least one of these checks fails *ontop*() returns 0. Otherwise we additionally check if at least two of the projected corner points $\{I_1, \ldots, I_4\}$ lie within the bounds of surface T. If this is the case, *ontop*() returns 1 and 0 otherwise. Note that ideally all four projected points should lie within the bounds of surface T. However, we use a less conservative check because the back part of an object is usually occluded and hence it is difficult to estimate the real spatial extent of o_2's bounding box.

Object-MDL Term E_{mdl} The object minimum description length (MDL) [16] term encodes the assumption that a simple explanation of the scene, i.e., by a small number of objects, is better than an unnecessarily complex one, consisting of a large number of objects. Hence, it puts a penalty on the number of objects present in the object map O:

$$E_{mdl}(O) = \sum_{o \in \mathcal{O}} T[\exists p \in \mathcal{I} : o_p = o] \cdot \lambda_{mdl}, \tag{9}$$

where $T[]$ is the indicator function that returns 1 if its argument is true and 0 otherwise. λ_{mdl} is a constant penalty.

3 Optimization

The goal is to find a scene proposal $\langle F, O \rangle$ that minimizes energy (1). Fig. 3 shows the steps of our optimization procedure that is described next.

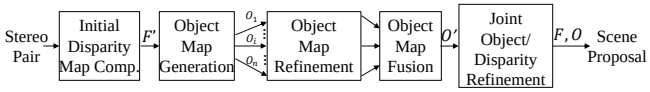

Fig. 3. Steps of our optimization algorithm.

Fig. 4. Object maps. We show 12 object maps generated for the Teapot scene. Pixel colors represent object identities.

Initial Disparity Map Computation We compute an initial disparity map using our re-implementation of PatchMatch Stereo [13]. PatchMatch Stereo optimizes the photo consistency term (eq. (2)) of our energy and returns an initial mapping F' of pixels to 3D planes. The disparity map F' is now kept fixed and refined in the last step of our pipeline, see fig. 3.

Object Map Generation This step generates n different object maps $O_1, \cdots, O_n$ (see fig. 3). To obtain one object map O_i, we first apply depth segmentation on the left input view. Our depth segmentation algorithm first divides the image into color segments using mean shift segmentation [17]. A disparity plane is fitted to each color segment using the disparity map F'. We then group segments of similar disparity planes.[5] Each group now forms a single depth segment. For each depth segment, we generate one 3D object. The 2D spatial extent of this object is defined by the pixels of the depth segment. The 3D extent of the object is approximated by fitting the tightest possible 3D bounding box with arbitrary orientation to the metric 3D coordinates[6] of the object's pixels. The parameters of the object's GMM (color model) are inferred from its pixels' color values.

The goal of the object map generation step is to generate n different object maps. This is accomplished by varying the parameters of the mean shift color segmentation algorithm (using two different settings) and the parameter of the disparity plane grouping method (six different settings). In total, this leads to $n = 2 \cdot 6 = 12$ different object maps. Object maps for the "Teapot" scene are shown in fig. 4.

Object Map Refinement We now process each object map O_i separately (see fig. 3). Our goal is to optimize the bounding boxes of objects in O_i so that energy (1) is minimized. Our optimization is based on hypotheses testing.

We go through all objects present in O_i starting with those that have the lowest bounding box volume. We align the current object's bounding box so that it stands on top of another bounding box that is spatially close in 3D space. If this aligned bounding box leads to a lower energy for O_i (which is likely due to the

[5] Details of the grouping algorithm are given in the sup. material.
[6] The 3D coordinates are reconstructed using disparity map F'.

gravity constraint of eq. (8)), it is immediately accepted as the current object's new bounding box. We then loop through the set of objects a second time. We check whether changing the spatial extent of the current object's bounding box leads to an object map of lower energy. We therefore shift the border planes of the bounding box by a random offset and accept this modified bounding box if energy (1) is reduced. This random test is repeated 100 times per object.

Object Map Fusion The goal is to select objects from object maps $O_1, \cdots, O_n$. The selected objects then form a new (fused) object map O' that minimizes our energy. Using the example of fig. 4, we can obtain O' by copying the teapot from O_3, the statue from O_2 and so on. We use simulated annealing to find an optimal selection of objects, i.e. starting from our current object map we apply a move to obtain a new one. If the new object map leads to a lower energy, we always accept it as our new solution. If not, we occasionally accept it depending on the amount of energy increase and the temperature in the annealing process.

Let us now discuss our method for making a move in the simulated annealing algorithm. We delete a random number of objects from our current object map. All objects that are not present in the resulting object map are inserted into a candidate set. We now iterate the following procedure until the candidate set is empty. A random object is selected and removed from the candidate set. We check whether inclusion of the selected object decreases the energy of our object map. If this is the case, we accept it as a new object in our object map.

To avoid a trivial optimum of energy (1) where no objects are selected, we impose a relatively high penalty $\lambda_{incomplete}$ for pixels that are not assigned to an object. Nevertheless, due to the different spatial extents of objects (see fig. 4) unassigned pixels may still be present in the final fused object map O' and there may also be pixels that are covered by two or more objects.[7] The joint object / disparity map refinement step, discussed next, ensures that each pixel is assigned to exactly one object.

Joint Object / Disparity Map Refinement We now extract all objects present in the fused object map O'. Our goal is to find a refined mapping O of pixels to the extracted objects and a refined disparity map F that is consistent with the objects' bounding boxes. We jointly estimate object labels and disparities that optimize our energy (1).

For accomplishing this joint optimization, we extend PatchMatch Stereo [13] so that it does not only assign each pixel to a plane, but also to an object label. To our knowledge, using the PatchMatch framework to perform such a joint optimization task is new and works as follows. In the initialization step of our extended PatchMatch algorithm we assign each pixel to a random plane and a random object label.[8] For each pixel, we can now compute the costs of its labeling

[7] We do not need an explicit term that penalizes cases where the spatial extent of two objects overlap, since for overlapping pixels the color penalty in eq. (4) is imposed multiple times, which leads to high energy solutions.

[8] This is the general perspective for solving a joint optimization task. For faster convergence, we do not perform random initialization, but assign the pixel to the values of F' and O'.

by evaluating the photo consistency, the color and the bounding box outlier terms of our energy.[9] After random initialization, the plane and object label of a pixel are propagated to its neighbors. A neighbor accepts the propagated label pair if it leads to lower costs than its current label pair. We also modify the refinement step of PatchMatch Stereo so that in addition to modifying the current plane, modification of the current object label is also allowed.

4 Results

Scene Proposal Generation As described above, our algorithm generates a scene proposal $\langle F, O \rangle$ as output. To create a large pool of scene proposals we run our method multiple times and vary the prior on color compactness (color term, eq. (4)) as well as the prior on the number of objects (object-MDL term, eq. (9)). By doing so we create a rich set of scene proposals that accounts for scenes of varying texture and different complexity. In detail, we run our method with parameters λ_{color} and λ_{mdl} in the range $\{0.001, \ldots, 0.01\}$ and $\{25, \ldots, 500\}$, respectively, to generate 34 scene proposals per stereo pair. We rank each scene proposal according to the similarity of its parameter settings to a default parameter set. The default parameters $\{\lambda_{color}, \lambda_{mdl}\} = \{0.005, 30\}$ were chosen such that they give a visually pleasing scene proposal[10] (i.e. object map and disparity map) for the "Parade" test image (fig. 6 left). The remaining parameters are set to the following fixed values $\{\lambda_{tight}, \lambda_{intersect}, \lambda_{gravity}, \lambda_{outlier}\} = \{0.005, 0.3, 20, 0.1\}$.

Quality of Object Map Pool We assessed the quality of our scene proposals on 10 stereo images shown in figs. 1(a), 7(top row) and 8(a). The dataset contains 4 Middlebury images and 6 self-recorded ones that show in- and outdoor scenes with a variety of different objects (e.g. cars and office equipment). To obtain ground truth segmentations for these images (fig. 7 middle row), we manually assigned each pixel in the image to an object, with in total 124 labeled objects. Note that we labeled both, things and stuff.

As quality measure for the proposal pool we use the accuracy score of [18] that is close to a Pascal VOC challenge score.[11] Given an object map O and a ground truth object map O^*, the accuracy score is defined as $C(O^*, O) = \frac{1}{N} \sum_{o^* \in O^*} |o^*| \cdot \max_{o \in O} sim(o^*, o)$, where N is the number of labeled pixels of the ground truth, $|o^*|$ is the number of pixels comprising object o^* and sim is a similarity function for the overlap of objects o and o': $sim(o, o') = |o \cap o'| / |o \cup o'|$.

[9] The values of the other terms of our energy are not affected by the joint object / disparity refinement step. There are also no pairwise smoothness terms in our energy. Hence the overall energy can be computed by summing up the costs of photo consistency, color and outlier terms over all pixels and adding the values of the other terms.

[10] Ideally the default parameters should be learned from a set of training images, which we leave for future work.

[11] Note that [18] uses the same score, but calls it "covering score". We use the term "accuracy score" since it penalizes both under- and over-segmentations. Hence, a proposal covering the whole image would be heavily penalized by the score.

Fig. 7 (bottom row) shows the object maps of our pool with the highest accuracy score. They correspond well to the ground truth.[12]

We follow the protocol of [10] to further process the scene proposal pool. First, we remove very small objects and near duplicate ones. We then rank the remaining objects according to the learned objectness measure of [10] which returns a probability for each 2D region of being a real-world object.

The curves in fig. 5 show the accuracy score averaged over all 10 test images. For each method, the extracted objects are sorted along the x-axis according to their rank. The solid pink curve and the red dashed curve show the accuracy score of the objects from our scene proposal pool. These two curves are clearly superior to the curves of the competitors, they achieve a high accuracy score faster. The difference between the pink and the red dashed curve is as follows. For plotting the red curve, we sort along the x-axis only according to the rank of the individual objects (defined by the objectness of [10]). For generating the pink curve, we first sort the objects according to the rank of the scene proposal (defined by the similarity to a default parameter set - see above) from which the objects originate.[13] This means that objects originating from a highly ranked scene proposal are plotted more towards the left of the x-axis. The two curves show that our sorting method (pink curve) performs better than solely sorting by objectness (red dashed curve), because fewer object hypothesis are necessary to reach the same accuracy.

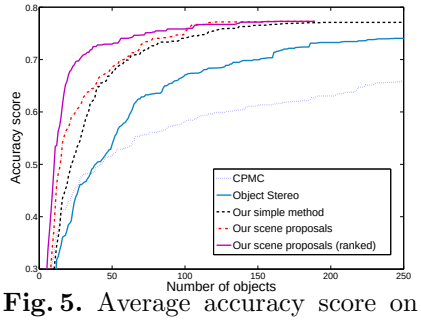

Fig. 5. Average accuracy score on our dataset for different methods.

As a simple competitor we use the object maps generated in the object map generation step of sec. 3 (black dashed curve, denoted as "our simple method"). These objects are not necessarily 3D scene-consistent and this is presumably the reason why this curve rises less steeply than those generated by our scene proposals that are scene-consistent. Note that the black curve is longer and reaches about the same average score as the pink one.[14] The solid cyan line was generated using scene proposals obtained from object stereo [1]. Although object stereo was designed to return only a single scene proposal we generated a set of 34 scene proposals per stereo pair by running object stereo with multiple different parameter settings. The curve of object stereo rises less steep than

[12] Note that some objects are split into multiple parts if they are separated by an occluding object, because the correct solution has not been generated by the object map generation step of sec. 3. The final refinement step can overcome this problem, see puppets in fig. 8(b), however, it is not guaranteed. We may leave an explicit "merging" step as future work.

[13] Objects of the same scene proposal are sorted according to objectness.

[14] This is due to the fact that our object pool is a (scene-consistent) subset of the initial object maps. This suggests that our energy selects those object hypotheses which correspond well to real-world objects.

those obtained by our method and reaches a lower average accuracy score. As a baseline (blue dotted curve), we plot the results of the CPMC approach [10] that uses no depth information and is clearly inferior.

To understand why our approach outperforms competitors, consider fig. 6. It shows those objects, in the pool of scene proposals, that give the highest accuracy score for three selected objects marked in fig. 6. Our approach gives results which are close to the ground truth, while CPMC [10] cannot obtain good results in the presence of color ambiguity. A typical failure case of object stereo can be seen in fig. 6(blue box). By grouping pixels of similar color and depth, object stereo assigns the elephant (lower left corner) to the same object as the frog (lower right corner). Such a configuration is unlikely with our model because the bounding box of this configuration is likely to intersect with the bounding boxes of other objects. Another competitor is "Blocks World" [7] that gives a mapping of image pixels to one out of seven classes. Though this result is not directly comparable to our method the class labeling can be regarded as object segmentation result. The publicly available code of [7] gives very coarse segmentations on our test images, which are inferior to our result (see sup. material).

Physical Reasoning for Stereo Matching We now show that our physical model also helps to improve stereo matching. Fig. 8 shows our results on the "Parade image" that is difficult due to its low-textured background. Fig. 8(b) shows our best object map. Fig. 8(e) shows our 3D reconstruction and bounding boxes that are aligned with the slanted ground plane due to the gravity term. The overlap of the bounding boxes is small due to the bounding box intersection term.

To see why the bounding boxes improve the disparity result, let us first look at the disparity map of our re-implementation of PatchMatch Stereo [13] fig. 8(c).[15]. Due to the low textured background, PatchMatch Stereo generates wrong matches that lead to 3D points floating in the air (left arrow) and erroneously assign the small background region between the two puppets to the foreground disparity (right arrows). Our physical model can overcome these problems, fig. 8(d). In the first case (left arrow), there is no bounding box that would support the wrongly reconstructed floating 3D points (see fig. 8(e)). (Floating bounding boxes are discouraged by the gravity term of our energy.) The correct reconstruction is accomplished due to the bounding box outlier term, which forces the reconstructed 3D points to lie inside a bounding box, here that of the background object. The second case (right arrow) is slightly different in that a reconstruction of the untextured background at the foreground disparity would lie inside a bounding box, i.e. that of the puppets. Hence, the outlier term would not impose a penalty. The color term of our energy resolves the matching ambiguity by looking at the pixels' colors. Since their colors better fit the color model of the background, the reconstruction is biased to lie inside the bounding box of the background. Figs. 8(f,g) show a similar example in 3D. The reconstruction of [13] produces floating pixels and attaches 3D points of the background to the foreground object "Dog" (fig. 8(f)). Our method overcomes these problems, which leads to a visually improved 3D view (fig. 8(g)).

[15] Pixels that fail the left-right consistency check are shown in black.

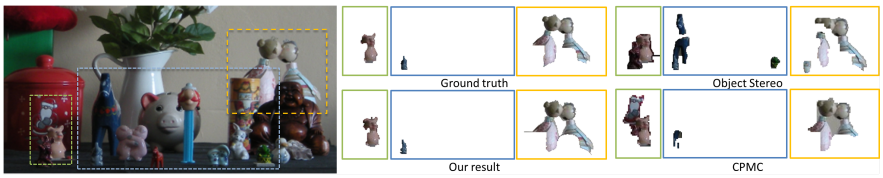

Fig. 6. Qualitative comparison of the object pool. The object hypotheses with the largest accuracy score with respect to three selected ground truth objects (i.e. "pig", "elephant", "puppets") are shown for different methods (CPMC [10] and object stereo [1]).

Fig. 7. Our best object maps for all test images. From top to bottom: Left image of stereo input pair; Ground truth; Our object map that gives highest accuracy score with respect to the ground truth.

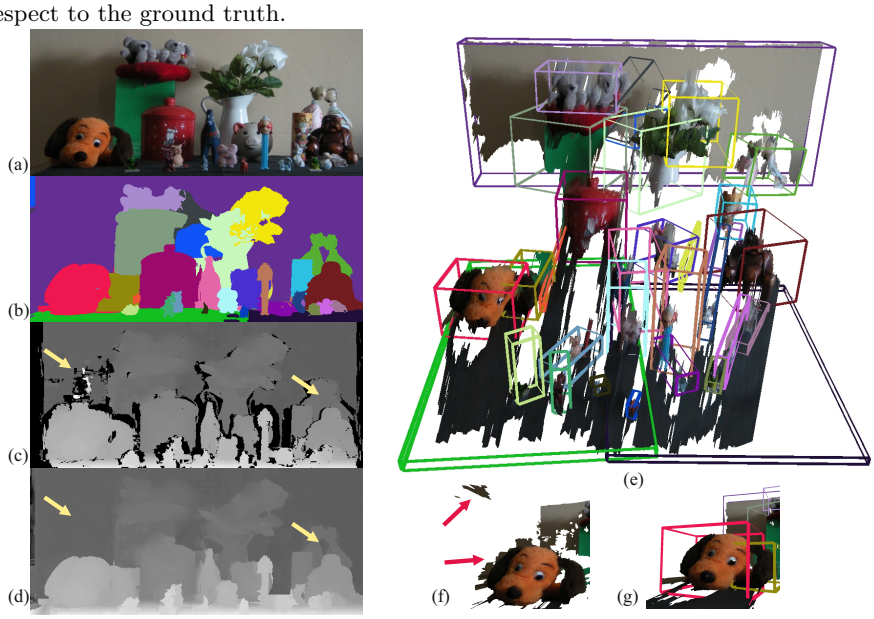

Fig. 8. The Parade stereo pair. (a) Left input image. (b) Our best object map. (c) Disparity Map produced by our re-implementation of PatchMatch Stereo [13]. Note, disparity errors marked by the arrows. (d) Our final disparity map. In contrast to [13] we can correctly reconstruct the disparity for regions marked by the arrows using our physical model (see text for an explanation). (e) 3D view generated using our disparity map in (d). We also show the computed bounding boxes. (f) 3D reconstruction results using the disparity map of [13] in (c). (g) Our reconstruction using the disparity map of (d). Red arrows in (f) mark artifacts which were corrected in (g).

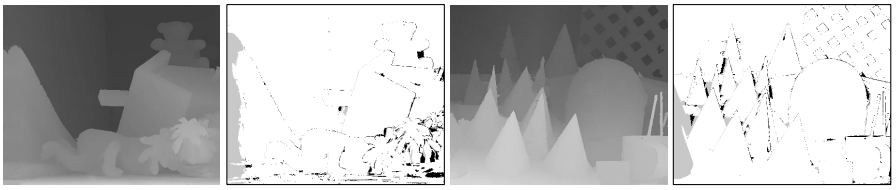

Fig. 9. Results of our method on the Middlebury Cones and Teddy sets. Our method takes ranks 1 (Teddy) and 2 (Cones) in the Middlebury table.

Middlebury Benchmark Results We have tested our method using the Middlebury benchmark [19] where it currently takes rank 13 out of 117 algorithms[16], which is in the ballpark of good methods.[17] Our method performs particularly well on the challenging Teddy and Cones images (see fig. 9) where it ranks 1st and 2nd (according to the error percentage in non-occluded regions). It also performs better than our reimplementation of [13], which ranks 17th (also see table in the sup. material). Please note that in contrast to almost all other methods in the table (except [1]), our method also provides a segmentation of the input images into objects.

Generality To test the generality of our approach, we did a proper train/test experiment, i.e., we only looked at the test data once. We train [1], our reimplementation of [13] and our algorithm on the 4 Middlebury evaluation images, i.e., the parameter setting that has led to the highest Middlebury ranking for each method is selected. We use these parameter settings to compute disparity maps for the Middlebury 2005 data set (Art, Books, Dolls, Laundry, Moebius, Reindeer). This set is more challenging than the evaluation set, i.e., error rates are typically higher. The average error percentage (in unoccluded regions) computed over all 6 sets is 7.90 for our method, while it is 8.40 for [13] and 10.90 for [1]. We achieve the lowest error percentage for 3 pairs. [13] is the winner for 2 pairs and [1] for 1 pair. Overall, our method outperforms [1] and [13] on this more difficult data. Exact numbers are found in tab. 1 of the sup. material.

Individual terms of the energy function We now use the same parameter setting for our method as in the previous experiment. To demonstrate that our "physics-based constraints" contribute to the quality of disparity maps, we disable the gravity, intersection and tightness terms, one after the other. For example, when switching off the gravity constraint we set $\lambda_{gravity} := 0$, while the other parameters are set to the values used in the previous experiment. Disabling the gravity constraint increases the average error on the 2005 data set from 7.90% to 8.30%, while disabling the intersection and tightness terms leads to errors of 7.91% and 8.21%, respectively. Note that apart from the Art set, disabling any of the 3 terms always leads to increased error percentages on the individual images. (An exception is Dolls where the intersection term seems to have a negative effect.) More information about this experiment is found in tab. 2 of the sup. material.

[16] Note that three methods take the same rank 13.

[17] The corresponding table and results on all four Middlebury images are found in the supplementary material.

5 Conclusions

We have presented an algorithm that jointly infers an object labeling as well as a disparity map. Our key contribution is to introduce physical constraints into this process. We have demonstrated that our approach can be used to generate a variety of physically plausible object hypotheses and outperforms state-of-the-art methods in this domain. The object hypotheses may serve as a valuable input for object recognition systems. For stereo matching, we have shown that our method is state-of-the art for some complex scenes in the Middlebury benchmark. In future work we plan to leverage further ideas, presented in [7], such as stability or merging of bounding boxes to better handle disconnected objects.

References

1. Bleyer, M., Rother, C., Kohli, P., Scharstein, D., Sinha, S.: Object stereo - joint stereo matching and object segmentation. (In: CVPR '11)
2. Izadi, S., Agarwal, A., Criminisi, A., Winn, J., Blake, A., Fitzgibbon, A.: C-slate: Exploring rem. collaboration on horiz. multi-touch surfaces. In: Tabletop. (2007)
3. Ladicky, L., Sturgess, P., Russell, C., Sengupta, S., Bastanlar, Y., Clocksin, W., Torr, P.: Joint optimisation for object class segmentation and dense stereo reconstruction. In: BMVC. (2010)
4. Price, B., Cohen, S.: Stereocut: Consistent interactive object selection in stereo image pairs. In: ICCV. (2011)
5. Björkman, M., Kragic, D.: Active 3d scene segmentation and detection of unknown objects. In: Conference on Robotics and Automation. (2010)
6. Björkman, M., Kragic, D.: Active 3d segmentation through fixation of previously unseen objects. In: BMVC. (2010)
7. Gupta, A., Efros, A., Hebert, M.: Blocks world revisited: Image understanding using qualitative geometry and mechanics. In: ECCV. (2010)
8. Ion, A., Carreira, J., Sminchisescu, C.: Image segmentation by discounted cumulative ranking on maximal cliques. In: ICCV. (2011)
9. Hoiem, D., Efros, A., Hebert, M.: Putting objects in perspective. IJCV (2008)
10. Carreira, J., Sminchisescu, C.: Constrained parametric min-cuts for automatic object segmentation. PAMI (2012)
11. Endres, I., Hoiem, D.: Category independent object proposals. In: ECCV. (2010)
12. J. Carreira, F.L., Sminchisescu, C.: Object recognition by sequential figure-ground ranking. IJCV (2012)
13. Bleyer, M., Rhemann, C., Rother, C.: Patchmatch stereo - stereo matching with slanted support windows. In: BMVC. (2011)
14. Yoon, K., Kweon, I.: Locally adaptive support-weight approach for visual correspondence search. In: CVPR. (2005)
15. Rhemann, C., Hosni, A., Bleyer, M., Rother, C., Gelautz, M.: Fast cost-volume filtering for visual correspondence and beyond. In: CVPR. (2011)
16. Delong, A., Osokin, A., Isack, H., Boykov., Y.: Fast approximate energy minimization with label costs. In: CVPR. (2010)
17. Christoudias, C., Georgescu, B., Meer, P.: Synergism in low-level vision. In: ICPR. Volume 4. (2002) 150–155
18. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: From contours to regions: An empirical evaluation. In: CVPR. (2009)
19. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. IJCV **47** (2002.) 7–42