

# Global Hypothesis Generation for 6D Object Pose Estimation

Frank Michel, Alexander Kirillov, Eric Brachmann, Alexander Krull  
Stefan Gumhold, Bogdan Savchynskyy, Carsten Rother  
TU Dresden

frank.michel@tu-dresden.de

## Abstract

This paper addresses the task of estimating the 6D pose of a known 3D object from a single RGB-D image. Most modern approaches solve this task in three steps: i) Compute local features; ii) Generate a pool of pose-hypotheses; iii) Select and refine a pose from the pool. This work focuses on the second step. While all existing approaches generate the hypotheses pool via local reasoning, e.g. RANSAC or Hough-voting, we are the first to show that global reasoning is beneficial at this stage. In particular, we formulate a novel fully-connected Conditional Random Field (CRF) that outputs a very small number of pose-hypotheses. Despite the potential functions of the CRF being non-Gaussian, we give a new and efficient two-step optimization procedure, with some guarantees for optimality. We utilize our global hypotheses generation procedure to produce results that exceed state-of-the-art for the challenging “Occluded Object Dataset”.

## 1. Introduction

The task of estimating the 6D pose of texture-less objects has gained a lot of attention in recent years. From an application perspective this is probably due to the growing interest in industrial robotics, and in various forms of augmented reality scenarios. From an academic perspective the dataset of Hinterstoisser *et al.* [10] marked a milestone, since researchers started to benchmark their efforts and progress in research started to be more measurable. In this work we focus on the following task. Given an RGB-D image of a 3D scene, in which a known 3D object is present, *i.e.* its 3D shape and appearance is known, we would like to identify the 6D pose (3D translation and 3D rotation) of that object.

Let us consider an exhaustive-search approach to this problem. We generate all possible 6D pose hypotheses, and for each hypothesis we run a robust ICP algorithm [2] to estimate a robust geometric fit of the 3D model to the underlying data. The final ICP score can then be used as the objective function to select the final pose. This approach

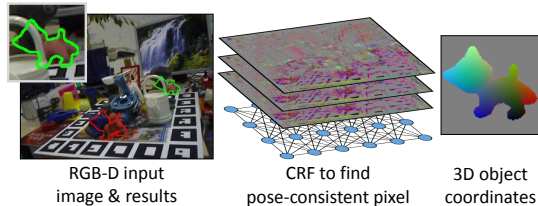


Figure 1. **Motivation.** Given an RGB-D input image (left) we aim at finding the 6D pose of a given object, despite it being strongly occluded (see zoom). Here our result (green) is correct, while Krull *et al.* [19] outputs an incorrect pose (red). The key concept of this work is to have a *global*, and hence powerful, geometric check, in the beginning of the pose estimation pipeline. This is in stark contrast to *local* geometric checks performed by all other methods. In a first step, a random forest predicts for each pixel a set of three possible object coordinates, *i.e.* dense continuous part labeling of the object (middle). Given this, a fully-connected pairwise Conditional Random Field (CRF) infers globally those pixels which are consistent with the 6D object pose. We refer to those pixels as *pose-consistent*. The final pose is derived from these pose-consistent pixels via an ICP-variant.

has two great advantages: (i) It considers all hypotheses; (ii) It uses a geometric error to prune all incorrect hypotheses. Obviously, this approach is infeasible from a computational perspective, hence most approaches generate first a pool of hypotheses and use a geometrically motivated scoring function to select the right pose, which can be refined with robust ICP if necessary. Table 1 lists five recent works with different strategies for “hypotheses generation” and “geometric selection”. The first work by Drost *et al.* [5], and recently extended by Hinterstoisser *et al.* [11], has no geometric selection process, and generates a very large number of hypotheses. The pool of hypotheses is put into a Hough-space and the peak of the distribution is found as the final pose. Despite its simplicity, the method achieves very good results, especially on the “Occluded Object Dataset”<sup>1</sup>, *i.e.* where objects are subject to strong occlusions. We conjecture that the main reason for its success is that it generates

<sup>1</sup><http://cvlab-dresden.de/iccv2015-occlusion-challenge/>

Method	Intermediate Representation	Hypotheses Generation	Average Number of Hypotheses	Hypotheses Selection	Hypotheses Refinement	Run Time
Drost <i>et al.</i> [5] Hinterstoisser <i>et al.</i> [11]	Dense Point Pair Features	All local pairs (large neighbourhood)	~ 20.000	Sub-optimal search	ICP	0.4s
Zach <i>et al.</i> [33]	multiple object coordinates	All local triplets with geometric check	2.000	Optimal w.r.t. PDA	PDA	0.5s
Brachmann <i>et al.</i> [3]	multiple object coordinates	Sampling triplets with geometric check	210	Optimal w.r.t. Energy	ICP variant	2s
Krull <i>et al.</i> [19]	multiple object coordinates	Sampling triplets with geometric check	210	Optimal w.r.t. CNN	ICP variant	10s
Our	multiple object coordinates	Fully-connected CRF with geometric check	0-10	Optimal w.r.t. ICP variant	ICP variant	1-3s

Table 1. A broad categorization of six different 6D object pose estimation methods with respect to four different computational steps: (a) Intermediate representation, (b) Hypotheses generation, (c) Hypotheses selection, (d) Hypotheses refinement, (e) Runtime. The key difference between the methods is marked in red: the number of generated hypotheses. We clearly generate least amount of hypotheses. For this we run an CRF-based hypotheses generation method which is more time-consuming and complex than in other approaches. Please note that our overall runtime is competitive. On the other hand, since we have fewer hypotheses, we can afford a more expensive ICP-like procedure to optimally select the best hypothesis. We show that we achieve results which are superior to all other methods on the challenging “Occluded Object Dataset”. (Note PDA stands for “projective data association”.)

hypotheses from all local neighborhoods in the image. Especially for objects that are subject to strong occlusions, it is important to predict poses from as local information as possible. The other three approaches [3, 19, 33] use triplets, and are all similar in spirit. In a first step they compute for every pixel one, or more, so-called object coordinates, a 3D continuous part-label on the given object (see Fig. 1 right). Then they collect locally triplets of points, in [33] these are all local triplets and in [3, 19] they are randomly sampled with RANSAC. For each triplet of object coordinates they first perform a geometry consistency check (see [3, 19, 33] for details<sup>2</sup>), and if successful, they compute the 6D object pose, using the Kabsch algorithm. Due to the geometric check it is notable that the amount of generated hypotheses is substantially less for these three approaches [3, 19, 33] than for the previously discussed [5, 11]. Due to this reason, the methods [3, 19, 33] can run more elaborate hypotheses selection procedures to find the optimal hypothesis. In [33] this is done via a so-called robust “projective data association” procedure, in [3] via a hand-crafted, robust energy, and in [19] via a CNN that scores every hypothesis. Our work is along the same direction as [3, 19, 33], but goes one step forward. We present a novel, and more powerful, geometric check, which results in even fewer hypotheses (between 0-10). For this reason we can also afford to run a complex ICP-like scoring function for selecting the best hypothesis. Since we achieve results that are better than state-of-the-art on the challenging occlusion dataset, our pool of hypotheses has at least the same quality as the larger hypotheses pool of all other methods. Our geometric check works roughly as follows. For each pair of object

<sup>2</sup>For instance, the geometric check of [3, 19] determines whether there exists a rigid body transformation of the triplets of 3D points, given by the depth image, for the triplet of 3D points from the object coordinates.

coordinates a geometry-consistency measure is computed. We combine a large number of pairs into a fully-connected Conditional Random Field (CRF) model. Hence, in contrast to existing work we perform a *global* geometry check and not a *local* one. It is important to note that despite having a complex CRF, we are able to have a runtime which is competitive with other methods, even considerably faster than [19]. As a side note, we also achieve these state-of-the-art results with little amount of learning, in contrast to e.g. [19].

**Our contributions** are in short:

- We are the first to propose a novel, *global* geometry check for the task of 6D object pose estimation. For this we utilize a fully-connected Conditional Random Field (CRF) model, which we solve efficiently, although its pairwise costs are non-Gaussian and hence efficient approximation techniques like [18] cannot be utilized.
- We give a new theoretical result which is used to compute our solutions. We show that for binary energy minimization problems, a (partial) optimal solution on a subgraph of the graphical model can be used to find a (partial) optimal solution on the whole graphical model. Proper construction of such subgraphs allows to drastically reduce the computational complexity of our method.
- Our approach achieves state-of-the-art results on the challenging “Occluded Object Dataset”, in reasonable runtime (1-3s).

## 2. Related Work

The topic of object detection and pose estimation has been widely researched in the past decade. In the brief review below, we focus only on recent works and split them into three categories. We will omit the methods [3, 19, 5, 33, 11] since they were already discussed in the

previous section.

**Sampling-Based Methods.** Sparse feature based methods ([8, 20]) have shown good results for accurate pose estimation. They extract points of interest and match them based on a RANSAC sampling scheme. With the shift of the application scenario into robotics their popularity declined since they rely on texture. Shotton *et al.* [25] addressed the task of camera re-localization by introducing the concept of scene coordinates. They learn a mapping from camera coordinates to world coordinates and generate camera pose hypotheses by random sampling. Most recently Phillips *et al.* [21] presented a method for pose estimation and shape recovery of transparent objects where a random forest is trained to detect transparent object contours. Those edge responses are clustered and random sampling is employed to find the axis of revolution of the object. Instead of randomly selecting individual pixels we will use the entirety of the image to find pose hypotheses.

**Non-Sampling-Based Methods.** An alternative to random sampling of pose hypotheses are Hough-voting based methods where all pixels cast a vote into a quantized prediction space (*e.g.* 2D object center and scale). The cell with the majority of votes is taken as the winner. [7, 26] used a Hough-voting-scheme for 2D object detection and coarse pose estimation. Tejani *et al.* [30] proposed an iterative latent-class Hough-voting-scheme for object classification and 3D pose estimation with RGB-D data as input. Template based methods [10, 9, 13] have also been applied to the task of pose estimation. To find the best match the template is scanned across the image and a distance metric is computed at each position. Those methods are harmed by clutter and occlusion which disqualifies them to be applied to our scenario. In our approach each pixel is processed, but instead of them voting individually we find pose-consistent pixel-sets by global reasoning.

**Pose Estimation using Graphical Models.** In an older piece of work the pose of object categories was found in images either in 2D [32] or in 3D [12]. They also use the key concept of discretized object coordinates for object detection and pose estimation. The MRF-inference stage for finding pose-consistent pixels is closely related to ours. Foreground pixels are accepted when the layout consistency constraint (where layout consistency means that neighboring pixels should belong to the same part) is satisfied. However since the shape of the object is unknown, the pairwise terms are not as strong as in our case. The closest related work to ours is Bergholdt *et al.* [1]. They use the same strategy of discriminatively modeling the local appearance of object parts and globally inferring the geometric connections between them. To detect and find the pose of articulated objects (faces, human spines, human poses) they extract feature points locally and combine them in a probabilistic, fully-connected, graphical model. However they

rely on an exact solution to the problem while a partial optimal solution is sufficient in our case. We therefore employ a different approach to solve the task.

### 3. Method - Overview

Before we describe our work in detail, we will introduce the task of 6D pose estimation formally and provide a high-level overview of our method. The objective is to find the 6D pose  $\mathbf{H}_c = [R_c | t_c]$  of object  $c$ , with  $R_c$  ( $3 \times 3$  matrix) describing a rotation around the object center and  $t_c$  ( $3 \times 1$  vector) representing the position of the object in camera space. The pose  $\mathbf{H}_c$  transforms each point in object coordinate space  $\mathbf{y} \in \mathcal{Y} \subseteq \mathbb{R}^3$  into a point in camera space  $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^3$ .

Our algorithm consists of three stages (see Fig. 2). In the first stage (Sec. 3.1) we densely predict object probabilities and object coordinates using a random forest. Instead of randomly sampling pose hypotheses as *e.g.* in [3] we use a graphical model to globally reason about hypotheses inliers. This second stage is described in Section 3.2 roughly and in Section 4 in detail. In the final stage (Sec. 3.3) we refine and rank our pose hypotheses to determine the best estimate.

#### 3.1. Random Forest

We use the random forests from Brachmann *et al.* [3]. Each tree  $T$  of the forest  $\mathcal{T}$  predicts for each pixel an object probability and an object coordinate. As mentioned above, an object coordinate corresponds to a 3D point on the surface of the object. In our case we have  $T = 3$ . As in [3] the object probabilities from multiple trees that are combined to one value using Bayes rule. This means that for a pixel  $i$  and object  $c$  we have the object probability  $p_c(i)$ . The object probabilities can be seen as a soft segmentation mask.

#### 3.2. Global Reasoning

In general, to estimate the pose of a rigid object, a minimal set of three correspondences between 3D points on the object and in the 3D scene is required [14]. The 3D points on the object, *i.e.* in the object coordinate system, are predicted by the random forest. One possible strategy is to generate such triplets randomly by RANSAC [6], as proposed in [3]. However, this approach has a serious drawback: the number of triples which must be generated by RANSAC in order to have at least a correct triple with the probability of 95%, is very high. Assuming that  $n$  out of  $N$  pixels contain correct correspondences, the total number of samples is  $\frac{\log(1-0.95)}{\log(1-(1-n/N)^3)}$ . For  $n/N = 0.005$ , which corresponds to a state-of-the-art local classifier, this constitutes  $\sim 24.000.000$  RANSAC iterations. Therefore, we address this problem with a different approach. Our goal is to assign to each pixel either one of the possible correspondence candidates, or an “outlier” label. We achieve this by formalizing a graphical model where each pixel is connected

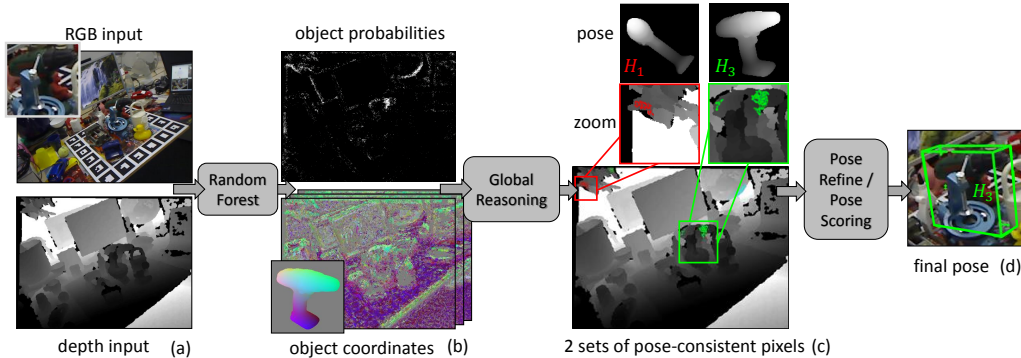


Figure 2. **Our pipeline:** Given an RGB-D image (a) a random forest provides two predictions: object probabilities and object coordinates (b). In a second stage our novel, fully-connected CRF infers pose-consistent pixel-sets (see zoom) (c). In the last stage, pose hypotheses given by pose-consistent pixels of the CRF are refined and scored by an ICP-variant. The pose with the lowest score is given as output (d).

to every other pixel with a pairwise term. The pairwise term encodes a geometric check which is defined later. The optimization problem of this graphical model is discussed in Sec. 4.2.

### 3.3. Refinement and Hypothesis Scoring

The output of the optimization of the graphical model is a collection of pose-consistent pixels where each of those pixels has a unique object coordinate. The collection is clustered into sets. In the example in Fig. 2(c) there are two sets (red, green). Each set provides one pose hypothesis. These pose hypotheses are refined and scored using our ICP-variant. In order to be robust to occlusion we only take the pose-consistent pixels within the ICP [2, 22] for fitting the 3D model.

## 4. Method - Graphical Model

After a brief introduction to graphical models (Sec. 4.1), we define our graphical model used for object pose estimation (Sec. 4.2). This is a fully-connected graph where each node has multiple labels, here 13. The globally optimal solution of this problem gives a pose-consistent (inlier) label to only those pixels that are part of the object, ideally. Since our potential functions are non-Gaussian the optimization problem is very challenging. We solve it approximately, but very efficiently, in a two stage procedure. The first stage conservatively prunes those pixels that are *likely* not inliers. This is done with a sparsely connected graph and TRW-S [16] as inference procedure (Sec. 4.3). The second stage (Sec. 4.4 - 4.6) describes an efficient procedure for solving the problem with only the inlier candidates remaining. We prove that by splitting this problem further into subproblems, in a proper way, a (partial) solution to one of these subproblems is guaranteed to be the (partial) optimal solution of the whole second stage problem. We use the found solutions to the subproblems to generate pose hypotheses.

### 4.1. Energy Minimization

Let  $G = (V, E)$  be an undirected graph with a finite set of nodes  $V$  and a set of edges  $E \in \binom{V}{2}$ . With each node  $u \in V$  we associate a finite set of labels  $L_u$ . Let  $\prod$  stand for the Cartesian product. The set  $\mathbb{L} = \prod_{u \in V} L_u$  is called *the set of labelings*. Its elements  $l \in \mathbb{L}$ , called *labelings*, are vectors  $l = (l_u \in L_u : u \in V)$  with  $|V|$  coordinates, where each one specifies a label assigned to the corresponding graph node. For each node a *unary cost function*  $\theta_u : L_u \rightarrow \mathbb{R}$  is defined. Its value  $\theta_u(l_u)$ ,  $l_u \in L_u$  specifies the cost to be paid for assigning label  $l_u$  to node  $u$ . For each two neighboring nodes  $\{u, v\} \in E$  a *pairwise cost function*  $\theta_{uv} : L_u \times L_v \rightarrow \mathbb{R}$  is defined. Its value  $\theta_{uv}(l_u, l_v)$  specifies compatibility of labels  $l_u$  and  $l_v$  in the nodes  $u$  and  $v$ , respectively. The triple  $(G, \mathbb{L}, \theta)$  defines a *graphical model*.

The *energy*  $E_V(l)$  of a labeling  $l \in \mathbb{L}$  is a total sum of the corresponding unary and pairwise costs

$$E_V(l) := \sum_{u \in V} \theta_u(l_u) + \beta \sum_{uv \in E} \theta_{uv}(l_u, l_v). \quad (1)$$

Finding a labeling with the lowest energy value constitutes an *energy minimization problem*. Although this problem is NP-hard, in general, a number of efficient approximative solvers exist, see [15] for a recent review.

### 4.2. Pose Estimation as Energy Minimization

Consider the following energy minimization problem:

- The set of nodes is the set of pixels of the input image, i.e., each graph node corresponds to a pixel. To be precise, we scale down our image by a factor of two for faster processing, i.e. each graph node corresponds to  $2 \times 2$  pixels.
- Number of labels in every node is the same. The label set  $L_u := \hat{L}_u \cup \{o\}$  consists of two parts, a subset  $\hat{L}_u$  of correspondence proposals and a special label  $o$ . In total, each node is assigned 13 labels: The forest  $\mathcal{T}$  provides 3 candidates for object coordinates in each pixel,  $2 \times 2$  pixels result in 12 labels, and the last label is the “outlier”.

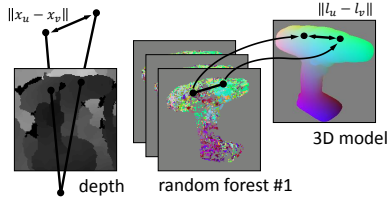


Figure 3. Visualization of our binary potential as defined in Eq. 2.

Each label from the subset  $\hat{L}_u$  corresponds to a 3D coordinate on the object. Therefore, we will associate such labels  $l_u$  with 3D vectors and assume vector operations to be well-defined for them. Unary costs  $\theta_u(l_u)$  for these labels are set to  $(1 - p_c(u))\alpha$ , where  $p_c(u)$  is defined in Section 3.1 and  $\alpha$  is a hyper-parameter of our method. We will call the labels from  $\hat{L}_u$  *inlier labels* or simply *inlier*.

The special label  $o$  denotes a situation in which the corresponding node does not belong to the object, or none of the labels in  $\hat{L}_u$  predicts a correct object coordinate. We call  $o$  the “outlier label”. Unary costs for the outlier labels are:  $\theta_u(o) = \frac{\sum p_c(u)\alpha}{12}$ ,  $u \in V$ .

Let us define pose-consistent pixels. If a node, comprising of  $2 \times 2$  pixels, is an inlier then the pixel with the respective label is defined as pose-consistent. The remaining three pixels are not pose-consistent and are ignored in the hypotheses selection stage. Also all pixels for which the node has an outlier label are not pose-consistent.

- Let  $x_u$  and  $x_v$  be 3D points in the camera coordinate system, corresponding to the nodes  $u$  and  $v$  in the scene. For any two inlier labels  $l_u \in \hat{L}_u$  and  $l_v \in \hat{L}_v$  we assign the pairwise costs as follows

$$\theta_{uv}(l_u, l_v) = \begin{cases} \left| \|l_u - l_v\| - \|x_u - x_v\| \right|, & \|x_u - x_v\| \leq D \\ \infty, & \text{otherwise.} \end{cases} \quad (2)$$

That is,  $\theta_{uv}(l_u, l_v)$  is equal to the absolute difference of distances between points  $l_u, l_v$  on the object and  $x_u, x_v$  in the scene (see Fig. 3) if the latter difference does not exceed the object size  $D$ .

Additionally, we define  $\theta_{uv}(l_u, o) = \theta_{uv}(o, l_v) = \gamma$  for  $l_u \in L_u, l_v \in L_v$ . Here  $\gamma$  is another hyper-parameter of our method. A sensible setting is  $\gamma = 0$ , however, we will choose  $\gamma > 0$  in parts of the optimization (see details below). We also assign  $\theta_{uv}(o, o) = 0$ , for all  $\{u, v\} \in E$ .

- The graph  $G$  is fully-connected, i.e., any two nodes  $u, v \in V$  are connected by an edge  $\{u, v\} \in E$ .

Given a labeling  $l \in \mathbb{L}$  we will speak about *inlier* and *outlier* nodes as those labeled with inlier or outlier labels, respectively.

The energy of any labeling is a sum of (i) the total unary costs for inlier labels, (ii) total geometrical penalty of the inlier labels, and (iii) total cost for the outlier labels. A labeling with the minimal energy corresponds to a geometrically consistent subset of coordinate correspondences with

a certain confidence for the local classifiers. We believe, there are such hyper-parameter settings that these coordinates would provide approximately correct object poses.

**Why a fully-connected graph?** At the first glance, one could reasonably simplify the energy minimization problem described above by considering a sparse, e.g. grid-structured graph. In this case the pairwise costs would control not all pairs of inlier labels, but only a subset of them, which may seem to be enough for a selection of inliers defining a good quality correspondence. Unfortunately, such a simplification has a serious drawback, nicely described in [1]: As soon as the graph is not fully connected, it tends to select an optimal labeling, which contains separated “islands” of inlier nodes, connecting to other “inlier-islands” only via outlier nodes. Such a labeling may contain geometrically independent subsets of inlier labels, which may “hallucinate” the object in different places of the image. Moreover, from our experience many of such “islands” contain less than three nodes, which increases the probability for pairwise geometrical costs to be low just by chance.

Concerning energy minimization. Our graph contains  $320 \times 240$  nodes which corresponds to the size of our discretized input image. Solving an energy minimization problem on such a fully-connected graph, even approximately, is in general infeasible if Gaussian potentials (like e.g. [18]) cannot be applied. Therefore, we suggest a *problem-specific*, but *very efficient* two-stage procedure for generating approximative solutions of the considered problem. In a first stage (Sec. 4.3) we reduce the size of the optimization problem, in the second (Sec. 4.4) we generate solution candidates.

**Graph Matching** problem (see e.g. [4, 27]) is another formalism used to find true correspondences from a large number of hypothetic correspondences using geometric constraints. However, one key aspect of graph matching is that one discrete feature (e.g. discrete object coordinate of a 3D model) can only match to one other discrete feature (e.g. discrete object coordinate candidate in the image (output from a decision tree)). Our problem formulation, in contrast, has continuous object coordinates.

### 4.3. Stage One: Problem Size Reduction

Despite what is discussed above about having a fully-connected graph, we used a sparse graphical model to reduce the number of possible correspondence candidates. An optimal labeling of this sparse model provides us with a set of inlier nodes, which hopefully contain the true inliers. On the second stage of our optimization procedure, described below, we build several fully-connected graphs from these nodes. For the sparse graph we use the following neighborhood structure: we connect each node to 48 closest nodes excluding the closest 8. We believe that the distance mea-

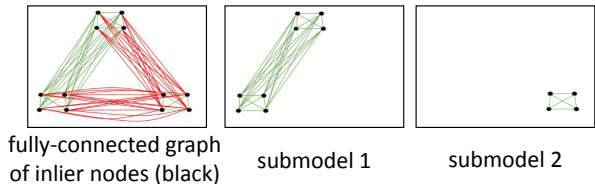


Figure 4. **Illustrating Optimization Stage Two.** (Left) the black pixels are all those pixels which were labeled as inliers, (potentially pose-consistent) in the first stage of the optimization. The first stage is opportunistic in the sense that wrong inliers may still be present. The goal of the second stage is to determine exactly the true inliers, from which we will determine the final pose. For this we have to solve the fully-connected graph shown, where each pixel has two labels, being an inlier (1) or outlier (0). Here the red links mark pairwise terms which contain  $\infty$  values. Unfortunately, state of the art solvers struggle with this problem, due to the presence of red links. We solve this by solving two (in practice many more) submodels (middle, right) that contain no red links. Each sub-problem produces a partial optimal solution  $\{0, 1, ?\}$ , where nodes that do not belong to the submodel are labeled 0. We can now guarantee that one of the partial optimal solution is the partial optimal solution of the full graph shown on the left.

sure between the closest nodes is very noisy.

We assign a positive value to the parameter  $\gamma$  penalizing transitions between inlier and outlier labels. This decreases the number of “inlier islands” by increasing the cost of the transition. We approximately solved this sparse problem with the TRW-S algorithm [16], which we run for 10 iterations. We found the recent implementation [24] of this algorithm to be up to 8 times faster than the original one [16] for our setting.

#### 4.4. Stage Two: Generation of Solution Candidates

**Fully-Connected Graphical Model.** As mentioned above, in the second stage we consider a *fully-connected* graphical model, where the node set contains only inlier nodes from the solution of the sparse problem. Moreover, to further reduce the problem size, we reduce the label set in each node to only two labels  $L_u := \{0, 1\}$ , where the label 0 corresponds to an outlier and the label 1 corresponds to the label associated with the node in the solution of the sparse problem. The unary and pairwise costs are assigned as before, but the hyper-parameters  $\alpha$ ,  $\beta$  and  $\gamma$  are different. In particular  $\gamma = 0$  since there is no reason to penalize transitions between inlier and outlier on this stage. Further, we will refer to  $(G, \mathbb{L}, \theta)$  defined above, as to master (**fully-connected**) model  $F$ .

Although such problems usually have a much smaller size (the solution of the sparse problem typically contains 20 to 500 inliers) our requirements to a potential solver are much higher at this stage. Whereas in the first stage we require only that the set of inlier nodes contains enough of

correct correspondences, the inliers obtained on the second stage must be *all* correct (have small geometrical error). Incorrect correspondences may deteriorate the final pose estimation accuracy. Therefore the quality of the solution becomes critical on this stage. Although problems of this size are often feasible for exact solvers, obtaining an exact solution may take multiple minutes or even hours. Therefore, we stick to the methods delivering only *a part of an optimal solution (partial optimal labeling)*, but being able to do this in a fraction of seconds, or seconds, depending on the problem size. Indeed, it is sufficient to have only three inlier to estimate the object pose.

**Partial Labeling.** A partial labeling can be understood as a vector  $l \in \{0, 1, ?\}^{|V|}$  with only a subset  $V' \subset V$  of coordinates assigned a value 0 or 1. The rest of coordinates take a special value  $?$  = “unlabeled”. The partial labeling is called *partial optimal labeling*, if there exists an *optimal* labeling  $l^* \in \mathbb{L}$  such that  $l_u^* = l_u$  for all  $u \in V'$ .

There are a number of efficient approaches addressing partial optimality (obtaining partial optimal labelings) for discrete graphical models for both multiple [28, 24] and two-label cases [17, 31]. We refer to [23] for an extensive overview. For problems with two labels the standard partial optimality method is QPBO [17], which we used in our experiments.

All partial optimality methods are based on sufficient optimality conditions, which have to be fulfilled for a partially optimal labeling. However, as it directly follows from [29, Prop.1], these conditions can hardly be fulfilled for label  $l_u$  in a node  $u$ , if for some neighboring node  $v: \{u, v\} \in E$  the difference between the smallest pairwise potential “attached” to the label  $l_u$ ,  $\min_{l_v \in L_v} \theta_{uv}(l_u, l_v)$  and the largest one  $\max_{l_v \in L_v} \theta_{uv}(l_u, l_v)$  is very large. In our setting this is the case, e.g., if for two nodes  $u$  and  $v$  (connected by an edge as any pair in a fully-connected graph) it holds  $\|x_u - x_v\| > D$ , see (2). Existence of such infinite costs leads to deterioration of the QPBO result: in many cases the returned partial labeling contains less than three labeled nodes, which is not sufficient for pose estimation.

To deal with this issue, we propose a novel method to find *multiple* partial labelings: We consider a set of induced submodels (see Definition 1 below) and find a partial optimal solution for each of them. We guarantee, however, that *at least* one of these partial labelings is a partial optimal one *for the whole graphical model* and not only for its submodel. Considering submodels allows to significantly reduce the number of node pairs  $\{u, v\}$  with  $\theta_{uv}(1, 1) = \infty$ . In its turn, it leads to many more nodes being marked as partially optimal by QPBO and therefore, provides a basis for a high quality pose reconstruction (see Fig. 4).

The theoretical background for the method is provided in the following subsection.

#### 4.5. On Optimality of Subproblem Solutions for Binary Energy Minimization

Let  $G = (V, E)$  be a graph and  $V' \subset V$  be a subset of its nodes. A subgraph  $G' = (V', E')$  is called *induced* w.r.t.  $V'$ , if  $E' = \{\{u, v\} \in E : u, v \in V'\}$  contains all edges of  $E$  connecting nodes within  $V'$ .

**Definition 1.** Let  $M = (G, \mathbb{L}, \theta)$  be a graphical model with  $G = (V, E)$  and  $\mathbb{L} = \prod_{u \in V} L_u$ . A graphical model  $M' = (G', \mathbb{L}', \theta')$  is called *induced* w.r.t.  $V' \subseteq V$  if

- $G'$  is an induced subgraph of  $G$  w.r.t.  $V'$ .
- $\mathbb{L}' = \prod_{u \in V'} L_u$ .
- $\theta'_u = \theta_u$  for  $u \in V'$  and  $\theta'_{uv} = \theta_{uv}$  for  $\{u, v\} \in E'$ .

**Proposition 1.** Let  $M = (G, \mathbb{L}, \theta)$  be a graphical model, with  $G = (V, E)$ ,  $\mathbb{L} = \{0, 1\}^{|V|}$  and  $\theta$  such that

$$\theta_{uv}(0, 1) = \theta_{uv}(1, 0) = \theta_{uv}(0, 0) = 0 \quad \forall \{u, v\} \in E. \quad (3)$$

Let  $\hat{l} \in \mathbb{L}$  be an energy minimizer of  $M$  and  $\hat{V} := \{u \in V : \hat{l}_u = 1\}$ .

Let  $M' = (G', \mathbb{L}', \theta')$  be an induced model w.r.t. some  $V' \supseteq \hat{V}$  and  $l'$  be an energy minimizer of  $M'$ . Then there exists a minimizer  $l^*$  of energy of  $M$ , such that  $l'_u = l^*_u$  for all  $u \in V'$ .

*Proof.*  $E_V(\hat{l}) = E_{V'}(\hat{x}_{V'}) + E_{V \setminus V'}(\hat{x}_{V \setminus V'}) \geq E(l') + E_{V \setminus V'}(\bar{0})$ . Since  $x_{V \setminus V'} = \bar{0}$  due to (3), the equality holds. The inequality holds by definition of  $l'$ . Let us consider the labeling  $l^* := (l', \bar{0})$  constructed by concatenation of  $l'$  on  $V'$  and  $\bar{0}$  on  $V \setminus V'$ . Its energy is equal to the right-hand-side of the expression, due to (3). Since  $\hat{l}$  is an optimal labeling, the inequality holds as equality and the labeling  $l^*$  is optimal as well. It finalizes the proof.  $\square$

**Corollary 1.** Let under condition of Proposition 1  $l'$  be a partial optimal labeling for  $M'$ . Then it is partial optimal for  $M$ .

Note, since pairwise costs of any two-label (pairwise) graphical model can be easily transformed to the form (2), see e.g. [17], Proposition 1 is generally applicable to all such models.

#### 4.6. Obtaining Candidates for Partial Optimal Labeling

To be able to use Proposition 1 we need a way to characterize possible optimal labelings for the master model  $F$  (defined in Section 4.4) to be able to generate possible sets  $V'$  containing all inlier nodes of an optimal labeling. Indeed, this characterization is provided by the following proposition:

**Proposition 2.** Let  $l^*$  be an optimal solution to the fully-connected problem described above. Then for any two inlier nodes  $u$  and  $v$ ,  $l^*_u = l^*_v = 1$ , it holds  $\|x_u - x_v\| \leq D$  or, in other words,  $\theta_{uv}(l^*_u, l^*_v) < \infty$ .

This proposition has a trivial proof: as soon as there is a labeling with a finite energy (e.g.  $l_u = 0$  for all  $u \in V$ ), an optimal labeling can not have an infinite one.

An implication of the proposition is quite clear from the applied point of view: all inlier nodes must be placed within a circle with a diameter equal to the maximal linear size of the object. Combining this observation with Proposition 1, we will generate a set of submodels, which contain all possible subsets of nodes satisfying the above condition.

A simple, yet inefficient way to generate all such submodels, is to go over all nodes  $u$  of the graph  $G$  and construct a subproblem  $M_u$  induced by nodes, which are placed at most at the distance  $D$  of  $u$ . A disadvantage of this method is that one gets as many as  $|V|$  subproblems, which leads to the increased runtime and too many almost equal submodels. Instead, we consider all connected inlier components obtained on the first stage as a result of the problem reduction. We remove all components with the size less than three, because, as we found experimentally, they mostly represent only noise. We enumerate all components, i.e., assign a serial number to each. For each component  $f$  we build a fully-connected submodel, which includes itself and all components with bigger serial number within the distance  $D$  from all nodes of  $f$ . Such an approach usually leads to at most 20 submodels and most of them get more than three partial optimal labels by QPBO.

Ignoring the heuristic removal of the components with the size less than three, such a procedure is guaranteed to provide a partial optimal solution of the whole problem, independent of the selected ordering of the components. Indeed, let an optimal labeling include inliers from  $m > 1$  components. Then select the component with the smallest index out of these  $m$  ones. By construction, the corresponding submodel will contain all the  $m$  components (since they all lie within the distance  $D$  and have larger indices) and therefore all the inliers of an optimal solution.

## 5. Experiments

We evaluated our method on a publicly available dataset. We will first introduce the dataset and then the evaluation protocol (Sec. 5.1). After that, we quantitatively compare our work with three competitors, and also present qualitative results (Sec. 5.2).

### 5.1. Dataset

To evaluate our method, we use the publicly available dataset of Brachmann *et al.* [3], known as ‘‘Occluded Object Dataset’’<sup>3</sup>. This dataset was presented in [3] and is an extension of [10]. They annotated the ground truth pose for 8 objects in 1214 images with various degrees of object occlusions.

<sup>3</sup><http://cvlab-dresden.de/iccv2015-occlusion-challenge/>



Figure 5. Qualitative results of our method on the ‘‘Occluded Object Dataset’’ [3]. Results of our method are depicted as green silhouettes, the ground truth pose is shown as a blue silhouette and results of the method by Krull *et al.* [19] are shown as red silhouettes. Note, since these results shows correct poses of our method the green silhouette is on top of the blue one.

Method	Our method	Hinterstoisser et al.[11]	Krull et al.[19]	Brachmann et al.[3]
Object	Scores			
Ape	80.7%	<b>81.4%</b>	68.0%	53.1%
Can	88.5%	<b>94.7%</b>	87.9%	79.9%
Cat	<b>57.8%</b>	55.2%	50.6%	28.2%
Driller	<b>94.7%</b>	86.0%	91.2%	82.%
Duck	74.4%	<b>79.7%</b>	64.7%	64.3%
Eggbox	47.6%	<b>65.5%*</b>	41.5%	9.0%
Glue	<b>73.8%</b>	52.1%	65.3%	44.5%
Hole Puncher	<b>96.3%</b>	95.5%	92.9%	91.6%
Average	<b>76.7%</b>	76.3%	70.3%	56.6%

Table 2. Quantitative comparison of [3], [19], [11] and our approach for all objects in the challenging ‘‘Occluded Object Dataset’’. \*The number for the Eggbox differs from [11] since they did not consider all images of the sequence (private e-mail exchange with the authors).

To evaluate our method we use the criteria from [10]. This means we measure the percentage of correctly estimated poses for each object. To determine the quality of an estimated pose we calculate the average distance of each point with respect to the estimated pose and the ground truth pose. The pose is accepted if the average distance is below 10% of the object diameter.

To find good parameters for our graphical model we created a validation set, which we will make publicly available. For this we annotated an additional image sequence (1235 images) of [10] containing 6 objects. The final set of parameters for stage one is  $\alpha = 0.21, \beta = 23.1, \gamma = 0.0048$  and stage two is  $\alpha = 0.2, \beta = 2.0, \gamma = 0.0$ .

## 5.2. Results

In the following we compare to the methods of Brachmann *et al.* [3], Krull *et al.* [19] and to the recently published state-of-the-art method of Hinterstoisser *et al.* [11]. Results are shown in Table 2. We achieve an average accuracy of 76.7% over all objects, which is 0.4% better than the current state-of-the-art method of Hinterstoisser *et al.* [11]. With respect to individual objects our method performs best on four objects and [11] on the other four. In comparison with [3] and [19] we achieve an improvement of 20.1% and 6.4% respectively. Since these two methods

use the same random forest, as we do, the benefits of using global reasoning can be seen. See Fig. 5 for qualitative results.

## 6. Conclusion and Future Work

In this work we have focused on the pose-hypothesis generation step, which is part of many pipelines for 6D object pose estimation. For this, we introduced a novel, global geometry check in form of a fully connected CRF. Since this direct optimization on the CRF is hardly feasible, we present an efficient two-step optimization procedure, with some guarantees on optimality. There are many avenues for future work. An obvious next step is to improve on the regression procedure for object coordinates, *e.g.* by replacing the random forests with a convolutional neural network.

**Acknowledgements:** This project has received funding from the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (grant agreement No 647769). The computations were performed on an HPC Cluster at the Center for Information Services and High Performance Computing (ZIH) at TU Dresden.



## References

- [1] M. Bergtholdt, J. Kappes, S. Schmidt, and C. Schnörr. A study of parts-based object class detection using complete graphs. *International Journal of Computer Vision*, 87(1):93, 2009. 3, 5
- [2] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 14(2):239–256, 1992. 1, 4
- [3] E. Brachmann, A. Krull, F. Michel, J. Shotton, S. Gumhold, and C. Rother. Learning 6d object pose estimation using 3d object coordinates. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014. 2, 3, 7, 8
- [4] D. Conte, P. Foggia, C. Sansone, and M. Vento. Thirty years of graph matching in pattern recognition. *International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI)*, 18(3):265–298, 2004. 5
- [5] B. Drost, M. Ulrich, N. Navab, and S. Ilic. Model globally, match locally: Efficient and robust 3d object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. 1, 2
- [6] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 3
- [7] J. Gall, A. Yao, N. Razavi, L. J. V. Gool, and V. S. Lempit-sky. Hough forests for object detection, tracking, and action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 33(11):2188–2202, 2011. 3
- [8] I. Gordon and D. G. Lowe. *What and Where: 3D Object Recognition with Accurate Pose*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. 3
- [9] S. Hinterstoisser, C. Cagniart, S. Ilic, P. F. Sturm, N. Navab, P. Fua, and V. Lepetit. Gradient response maps for real-time detection of textureless objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 34(5):876–888, 2012. 3
- [10] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. R. Bradski, K. Konolige, and N. Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2012. 1, 3, 7, 8
- [11] S. Hinterstoisser, V. Lepetit, N. Rajkumar, and K. Konolige. Going further with point pair features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. 1, 2, 8
- [12] D. Hoiem, C. Rother, and J. M. Winn. 3d layouterf for multi-view object class recognition and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007. 3
- [13] D. P. Huttenlocher, G. A. Klanderman, and W. Rucklidge. Comparing images using the hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 15(9):850–863, 1993. 3
- [14] W. Kabsch. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A*, 32(5):922–923, Sep 1976. 3
- [15] J. H. Kappes, B. Andres, F. A. Hamprecht, C. Schnörr, S. Nowozin, D. Batra, S. Kim, B. X. Kausler, T. Kröger, J. Lellmann, N. Komodakis, B. Savchynskyy, and C. Rother. A comparative study of modern inference techniques for structured discrete energy minimization problems. *International Journal of Computer Vision*, pages 1–30, 2015. 4
- [16] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2006. 4, 6
- [17] V. Kolmogorov and C. Rother. Minimizing non-submodular functions with graph cuts—a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 29(7):1274–1279, 2007. 6, 7
- [18] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *Proceeding of the Conference Neural Information Processing Systems (NIPS)*, 2011. 2, 5
- [19] A. Krull, E. Brachmann, F. Michel, M. Y. Yang, S. Gumhold, and C. Rother. Learning analysis-by-synthesis for 6d pose estimation in rgb-d images. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2015. 1, 2, 8
- [20] M. Martinez, A. Collet, and S. S. Srinivasa. Moped: A scalable and low latency object recognition and pose estimation system. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2010. 3
- [21] C. J. Phillips, M. Lecce, and K. Daniilidis. Seeing glassware: from edge detection to pose estimation and shape recovery. In *Robotics: Science and Systems*, 2016. 3
- [22] R. B. Rusu and S. Cousins. 3D is here: Point Cloud Library (PCL). In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2011. 4
- [23] A. Shekhovtsov. Maximum persistency in energy minimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 6
- [24] A. Shekhovtsov, P. Swoboda, and B. Savchynskyy. Maximum persistency via iterative relaxed inference with graphical models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 6
- [25] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013. 3
- [26] M. Sun, G. R. Bradski, B.-X. Xu, and S. Savarese. Depth-encoded hough voting for joint object detection and shape recovery. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010. 3
- [27] P. Swoboda, C. Rother, H. Alhajja, D. Kainmüller, and B. Savchynskyy. A study of Lagrangean decompositions and dual ascent solvers for graph matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 5
- [28] P. Swoboda, B. Savchynskyy, J. H. Kappes, and C. Schnörr. Partial optimality by pruning for map-inference with general graphical models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1170–1177, 2014. 6

- [29] P. Swoboda, A. Shekhovtsov, J. Kappes, C. Schnörr, and B. Savchynskyy. Partial Optimality by Pruning for MAP-Inference with General Graphical Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 38(7):1370–1382, 7 2016. 6
- [30] A. Tejani, D. Tang, R. Kouskouridas, and T.-K. Kim. Latent-class hough forests for 3d object detection and pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014. 3
- [31] C. Wang and R. Zabih. Relaxation-based preprocessing techniques for markov random field inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 6
- [32] J. Winn and J. Shotton. The layout consistent random field for recognizing and segmenting partially occluded objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006. 3
- [33] C. Zach, A. Penate-Sanchez, and M.-T. Pham. A dynamic programming approach for fast and robust object pose recognition from range images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2