Learning 6D Object Pose Estimation using 3D Object Coordinates - Supplementary Material -

Eric Brachmann[1], Alexander Krull[1], Frank Michel[1], Stefan Gumhold[1], Jamie Shotton[2], and Carsten Rother[1]

[1] TU Dresden, Dresden, Germany
[2] Microsoft Research, Cambridge, UK

This supplementary material is not necessary to understand the main paper. As mentioned in the main paper the supplementary material discusses the following points in more detail:

- Details on our dataset
- Details on the decision forest
- Deduction of Eq.1 of the main paper (calculating object probabilities)
- List of parameter settings
- Exact definition of the pose tolerance
- Detailed qualitative and quantitative results for the data of Hinterstoisser et al. [2] and our data.
- Additional experimental results regarding detection, occlusion and contribution of our different energy terms.
- Description of the supplementary video

1 Data Acquisition

In this section we describe our data acquisition process. Fig. 1 displays the scan and preprocessing pipeline. The objects are scanned with a commercially available Kinect camera and are segmented in each RGB-D image afterwards. This procedure is done three times with varying lighting conditions. For each object we obtain three sequences which are shown exemplarily for five of our objects in Fig. 2. Fig. 3 shows every object of our dataset with its name.

2 Details on the Decision Forest

Forest Features. In the feature design for the tree nodes we follow [7] in their use of various simple RGB and depth features. The features are extremely fast to evaluate because they are based on simple pixel comparisons [3, 6]. The feature responses can be computed as follows:

$$f^{\text{da-d}}(\boldsymbol{\theta}, \mathbf{p}_i) = d\left(\mathbf{p}_i + \frac{\boldsymbol{\omega}_1}{d_i}\right) - d\left(\mathbf{p}_i + \frac{\boldsymbol{\omega}_2}{d_i}\right) \tag{1}$$

$$f^{\text{da-rgb}}(\boldsymbol{\theta}, \mathbf{p}_i) = I\left(\mathbf{p}_i + \frac{\boldsymbol{\omega}_1}{d_i}, \gamma_1\right) - I\left(\mathbf{p}_i + \frac{\boldsymbol{\omega}_2}{d_i}, \gamma_2\right) \ , \tag{2}$$

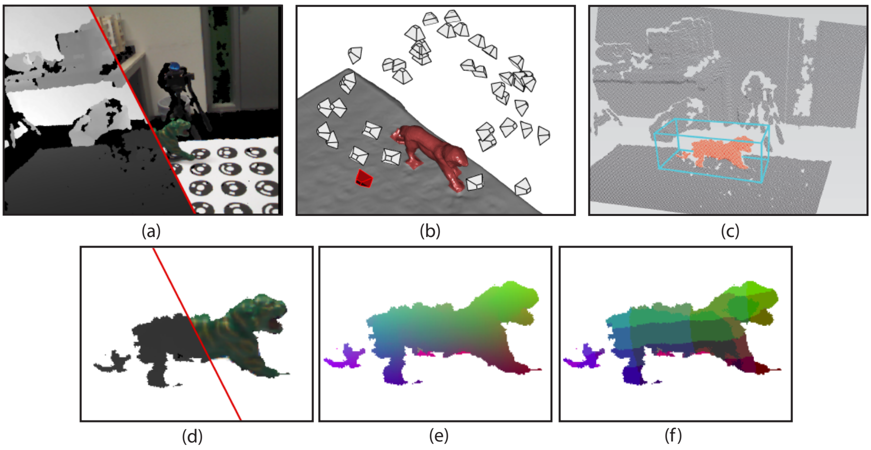

Fig. 1. Training data acquisition pipeline. For each object we run the following acquisition procedure. We use the commercially available Kinect camera and KinectFusion [4] system to obtain a 3D scan of the scene together with the camera poses (b) and the RGB-D data for each captured frame (a). The black holes in the RGB-D images correspond to pixels were no depth information was available. (b) A top-side view with a subset of 50 out of 1000 camera-frusta, and the object (red) in the center. The object has been manually segmented in 3D. (c) A 3D bounding box (light blue) is positioned around the object (shown as the depth map from the camera with red frustum in (b)). The object mask for each RGB-D frame is then defined by all object pixels where the corresponding depth values fall inside the bounding box. (d) For training we use the segmented RGB-D images. The segmentation has holes and is imperfect at boundaries, due to the inaccurate and missing depth values from Kinect. However, the test data presents similar noise characteristics, and so it is beneficial to have such noise in the training images (rather than e.g. rendering from the 3D reconstruction in (b)). In (e) we show the continuous 3D object coordinates, and in (f) the quantized 3D object coordinates on a $5 \times 5 \times 5$ grid. This quantization is used to compute the objective function for learning the tree structure. Both, (e) and (f) are derived directly from the 3D object bounding box only.

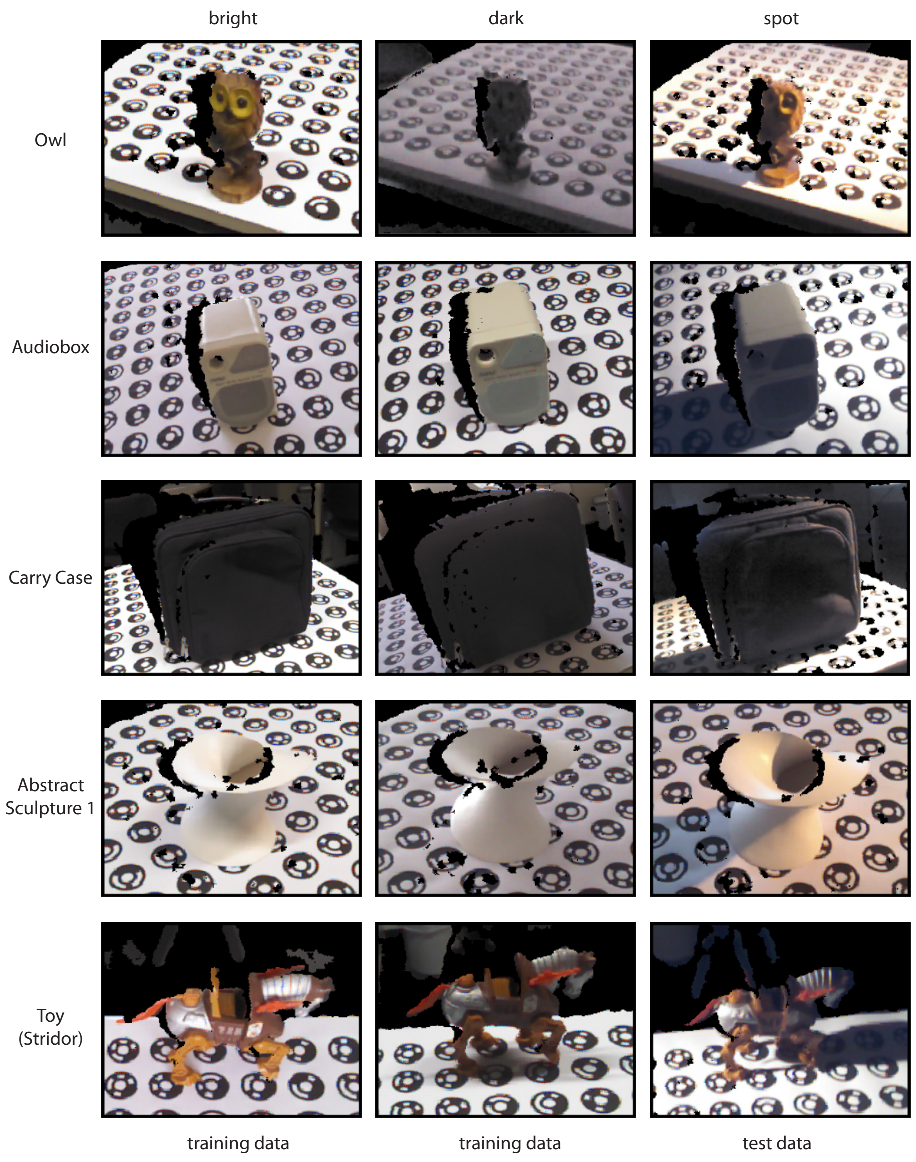

Fig. 2. Three different lighting conditions. For each object we gathered training and test images under three different lighting conditions: bright artificial light (left), darker natural light (middle), directional spot light (right). Note that the markers on the table are used only to register scans of different lighting conditions.

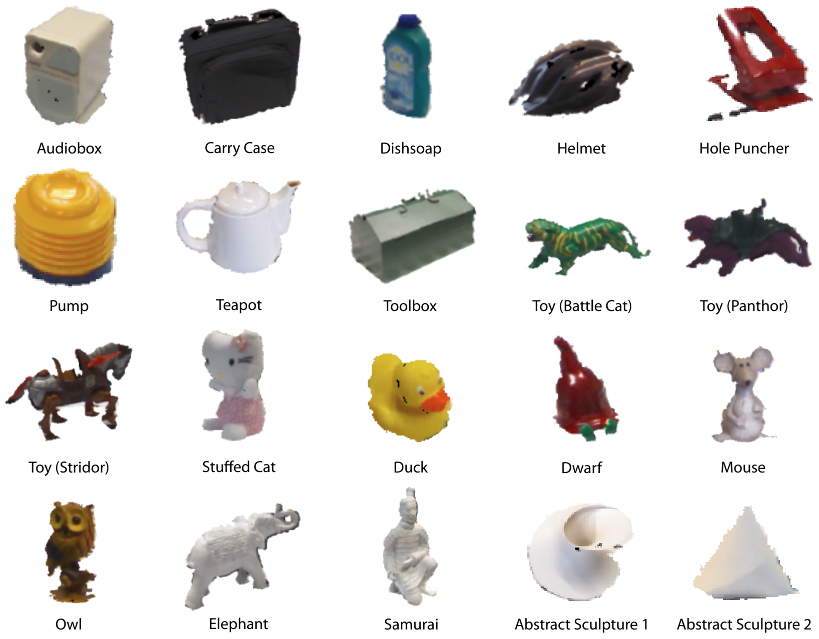

Fig. 3. Each object of our dataset segmented and labeled with its name. This is the same segmentation we use for training our decision forests.

where $I(\mathbf{p}_i, \gamma) = \mathbf{x}_i^{\mathbf{rgb}}[\gamma]$ returns the R, G, or B channel of a pixel according to γ, and $d(\mathbf{p}_i) = d_i$ returns the depth at position $\mathbf{p}_i$. Abbreviations 'da-rgb' and 'da-d' stand for depth adaptive RGB differences and depth adaptive depth differences. $\boldsymbol{\omega}$ indicates a 2D offset. The division by d_i makes the features largely depth invariant, and is similar in spirit to [8]. Each split node in the forest thus stores a unique set of parameters $\boldsymbol{\theta}_n \subseteq \{\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, \gamma_1, \gamma_2, z, \tau_f\}$, with $z \in \{\text{da-d}, \text{da-rgb}\}$ indicating the type of feature to use. τ_f denotes a threshold on the feature response that decides whether a pixel goes to the left or the right child of a node. This threshold is also stored per node.

We also tested absolute LAB features, and features built on the angles between estimated normals but both without performance gain.

Forest Training. In the following, we describe the training of a randomized decision tree. The procedure is the same for all trees of the forest. We will treat the training images as a combined set of training pixels. Each training pixel i is characterized by its position $\mathbf{p}_i$, its color $\mathbf{x}_i^{\mathbf{rgb}}$, its depth d_i, its object instance label c_i and its object coordinate $\mathbf{y}_i$.

Training starts at the top node where a feature with parameters $\boldsymbol{\theta}$ is selected with the goal to reduce the uncertainty in $p(c)$ and $p(\mathbf{y}|c)$ the most (based on the training data). The selected feature divides the data to go to the left resp. the right child node, where the feature selection process repeats. This process iterates until a stopping criterion is met. This is the case if a maximum depth has been reached, or not enough training pixels arrived at the node. Due to runtime complexity only a random sub-sampling of both, training pixels and features, is used. This also introduces variability between trees in the forest, and hence the ability to generalize to unseen data.

The selection of features at each node is based on a split score which should be able to handle well the discrete distribution $p(c)$ and the continuous location distribution $p(\mathbf{y}|c)$. We quantize the continuous object coordinate labels $\mathbf{y}$ based on a grid to obtain discrete object coordinate labels $\hat{\mathbf{y}}$. We denote by $p_\eta(\hat{\mathbf{y}}, c)$ the joint distribution at node η, and by $p_\eta^{\boldsymbol{\theta},\leftarrow}(\hat{\mathbf{y}}, c)$ resp. $p_\eta^{\boldsymbol{\theta},\rightarrow}(\hat{\mathbf{y}}, c)$ the distributions in the left resp. the right child of node η, split by parameters $\boldsymbol{\theta}$. We select parameters $\boldsymbol{\theta}$ such that the information gain IG in objects and classes is maximized:

$$\begin{aligned} IG(\eta, \boldsymbol{\theta}) &= \mathcal{H}(p_\eta(\hat{\mathbf{y}}, c)) \\ &- \sum_{k \in \{\leftarrow, \rightarrow\}} \Big[\frac{|\eta^k|}{|\eta|} \mathcal{H}(p_\eta^{\boldsymbol{\theta},k}(\hat{\mathbf{y}}, c)) \Big] \end{aligned} \tag{3}$$

where $\mathcal{H}(p_\eta(\hat{\mathbf{y}}, c)) = -\sum_c \sum_{\hat{\mathbf{y}}} p_\eta(\hat{\mathbf{y}}, c) \log p_\eta(\hat{\mathbf{y}}, c)$, $|\eta|$ is the number of training pixels which arrived at that node, and $|\eta^k|, k \in \{\leftarrow, \rightarrow\}$ is the number of pixels split to the left resp. the right child node.

3 Combining Object Probabilities from Multiple Trees

In this section we will give a deduction of Eq. 1 in the main paper, where we combine the probability outputs of the individual trees to calculate the object probabilities $p_{c,i}$. Our goal is to calculate the approximate probability $p_{c,i} \approx p(c|\mathbf{l}_i)$, that a pixel i belongs to object c given it ended up in the leafs $\mathbf{l}_i = (l_i^1, \dots, l_i^{|\mathcal{T}|})$ of the trees $T^1 \dots T^{|\mathcal{T}|}$. Based on Bayes' theorem we can calculate this probability as

$$\begin{aligned} p(c|\mathbf{l}_i) &= \frac{p(c, \mathbf{l}_i)}{p(\mathbf{l}_i)} && (4)\\ &= \frac{p(c, \mathbf{l}_i)}{\sum_{\hat{c} \in C} p(\hat{c}, \mathbf{l}_i) + p(bg, \mathbf{l}_i)}, && (5)\end{aligned}$$

where $p(\mathbf{l}_i)$ is the joint probability that a pixel ends up in the leafs $\mathbf{l}_i$ regardless the object it belongs to. The expression $p(bg, \mathbf{l}_i)$ denotes the joint probability, that the pixel is part of the background and ends up in the leafs $\mathbf{l}_i$. We will first focus on calculating the joint probability $p(c, \mathbf{l}_i)$ that the pixel belongs to object c and ends up in the leafs $\mathbf{l}_i$. It can be calculated as

$$\begin{aligned} p(c, \mathbf{l}_i) &= p(c) p(\mathbf{l}_i|c) && (6)\\ &= p(c) \prod_{j=1}^{|\mathcal{T}|} p(l_i^j|c), && (7)\end{aligned}$$

where equation 7 is based on the assumption of conditional independence of the leaf outcomes $\mathbf{l}_i$ given the pixel's object affiliation c. This assumption can be seen as problematic, since the trees were trained to separate pixels not only according to object affiliation but also according to their position in object space. Our calculations should thus be viewed as an approximation. We can calculate the conditional probability $p(l_i^j|c)$ for a leaf outcome l_i^j given the pixel is part of object c as

$$p(l_i^j|c) = \frac{p(c|l_i^j) p(l_i^j)}{p(c)}, \quad (8)$$

where $p(l_i^j)$ is the a priori probability of the leaf outcome l_i^j. We can thus calculate the joint probability $p(c, \mathbf{l}_i)$ for the object affiliation c and leaf outcome $\mathbf{l}_i$ as

$$p(c, \mathbf{l}_i) = p(c) \prod_{j=1}^{|\mathcal{T}|} \frac{p(c|l_i^j) p(l_i^j)}{p(c)}, \quad (9)$$

In a similar fashion we can calculate the joint probability $p(c, \mathbf{l}_i)$ for the pixel is part of the background and leaf outcome $\mathbf{l}_i$ as

$$p(bg, \mathbf{l}_i) = p(bg) \prod_{j=1}^{|\mathcal{T}|} \frac{p(bg|l_i^j)p(l_i^j)}{p(bg)}, \tag{10}$$

where $p(bg)$ is the a priori probability that a pixel belongs to background. By combining equations 5, 9 and 10 we can calculate the desired probability

$$\begin{aligned}
p(c|\mathbf{l}_i) &= \frac{p(c, \mathbf{l}_i)}{p(\mathbf{l}_i)}, & (11)\\
&= \frac{p(c, \mathbf{l}_i)}{\sum_{\hat{c} \in C} p(\hat{c}, \mathbf{l}_i) + p(bg, \mathbf{l}_i)}, & (12)\\
&= \frac{p(c) \prod_{j=1}^{|\mathcal{T}|} \frac{p(c|l_i^j)p(l_i^j)}{p(c)}}{\left(\sum_{\hat{c} \in C} p(\hat{c}) \prod_{j=1}^{|\mathcal{T}|} \frac{p(\hat{c}|l_i^j)p(l_i^j)}{p(\hat{c})}\right) + p(bg) \prod_{j=1}^{|\mathcal{T}|} \frac{p(bg|l_i^j)p(l_i^j)}{p(bg)}}, & (13)\\
&= \frac{p(c)^{-(|\mathcal{T}|-1)} \prod_{j=1}^{|\mathcal{T}|} p(c|l_i^j)p(l_i^j)}{\left(\sum_{\hat{c} \in C} p(\hat{c})^{-(|\mathcal{T}|-1)} \prod_{j=1}^{|\mathcal{T}|} p(\hat{c}|l_i^j)p(l_i^j)\right) + p(bg)^{-(|\mathcal{T}|-1)} \prod_{j=1}^{|\mathcal{T}|} p(bg|l_i^j)p(l_i^j)}. & (14)
\end{aligned}$$

If we assume that the a priori probability $p(c)$ that a pixel is part of an object c is the same for each object and identical to the a priori probability $p(bg)$ that a pixel is part of the background, we can simplify:

$$\begin{aligned}
p(c|\mathbf{l}_i) &= \frac{\prod_{j=1}^{|\mathcal{T}|} p(c|l_i^j)p(l_i^j)}{\left(\sum_{\hat{c} \in C} \prod_{j=1}^{|\mathcal{T}|} p(\hat{c}|l_i^j)p(l_i^j)\right) + \prod_{j=1}^{|\mathcal{T}|} p(bg|l_i^j)p(l_i^j)} & (15)\\
&= \frac{\left(\prod_{j=1}^{|\mathcal{T}|} p(l_i^j)\right) \prod_{j=1}^{|\mathcal{T}|} p(c|l_i^j)}{\left(\prod_{j=1}^{|\mathcal{T}|} p(l_i^j)\right)\left(\left(\sum_{\hat{c} \in C} \prod_{j=1}^{|\mathcal{T}|} p(\hat{c}|l_i^j)\right) + \prod_{j=1}^{|\mathcal{T}|} p(bg|l_i^j)\right)} & (16)
\end{aligned}$$

The factor $\prod_{j=1}^{|\mathcal{T}|} p(l_i^j)$ is present in the numerator and denominator. We can thus finally write:

$$\boxed{p_{c,i} = p(c|\mathbf{l}_i) = \frac{\prod_{j=1}^{|\mathcal{T}|} p(c|l_i^j)}{\left(\sum_{\hat{c} \in C} \prod_{j=1}^{|\mathcal{T}|} p(\hat{c}|l_i^j)\right) + \prod_{j=1}^{|\mathcal{T}|} p(bg|l_i^j)}}.$$

The probabilities $p(c|l_i^j)$ and $p(bg|l_i^j)$ are stored at the leaf l_i^j. In our implementation we add a small constant ($const = 10^{-8}$) in the denominator for numerical stability.

4 Complete List of Parameter Settings

The following settings were used in all pose estimation experiments:

Training Parameters	
maximum feature offset:	20 pixel meters
number of features generated at each node:	1000
ratio of 'da-d' to 'da-rgb' features	0.5
number of trees $\|\mathcal{T}\|$:	3
random pixels per image to learn tree structure:	1000
random pixels per image to learn leaf distributions:	5000
stopping criterion: minimum number of pixels per node:	50

Testing Parameters	
depth comp. weight λ^{depth}:	1.5
coordinate comp. weight λ^{coord}:	1
object comp. weight λ^{obj}:	1
threshold τ_d used in E_c^{depth}:	50 mm
threshold τ_y used in E_c^{coord}:	$(0.2 \cdot \delta_c)^2$
threshold τ_{pc} used in E_c^{coord}:	10^{-8}
number of Hypothesis to be sampled:	210
threshold used during sampling of poses:	$0.05 \cdot \delta_c$
inlier threshold used in refinement:	20 mm
number of Hypothesis to be refined:	25

5 Exact Definition of the Pose Tolerance

We follow Hinterstoisser[2] by measuring accuracy as the fraction of test images where the pose of the object in question was estimated correctly. Poses are correct, if the following inequality holds:

$$\tau_p < \frac{1}{|\mathcal{M}|} \sum_{\mathbf{x} \in \mathcal{M}} ||H\mathbf{x} - \tilde{H}\mathbf{x}|| \tag{17}$$

where $\mathcal{M}$ is the set of all object vertices, H is the ground truth pose and $\tilde{H}$ is the estimated pose. For rotationally symmetric objects (e.g. a bowl) a slightly different metric is used:

$$\tau_p < \frac{1}{|\mathcal{M}|} \sum_{\mathbf{x}_1 \in \mathcal{M}} \min_{\mathbf{x}_2 \in \mathcal{M}} ||H\mathbf{x}_1 - \tilde{H}\mathbf{x}_2||. \tag{18}$$

The threshold τ_p is fixed to be 10% of the object diameter. The following objects from our dataset were considered rotationally symmetric: Dish Soap, Abstract Sculpture 2 and Toolbox. The following objects from the Hinterstoisser dataset were considered rotationally symmetric: Glue and Box.

6 Detailed Qualitative and Quantitative Results

Complete quantitative results for all of our own objects are included in Table 1, and Fig. 4 shows more qualitative examples of the output of our system on our dataset. Table 2 lists all quantitative results on the Hinterstoisser dataset, and Fig. 5 shows more qualitative examples.

Table 1. Accuracy on our dataset when testing with different lighting (spot light) and same lighting (bright light) compared to the training data.

Object	*bright* test condition	*spot* test condition
Audio Box	**90.5%**	75.4%
Carry Case	**97.9%**	95.9%
Dish Soap	**100.0%**	**100.0%**
Helmet	**91.0%**	77.6%
Hole Puncher	**99.9%**	98.1%
Pump	**81.5%**	69.3%
Teapot	**99.5%**	91.9%
Toolbox	99.0%	**99.5%**
Toy (Battle Cat)	**96.9%**	91.8%
Toy (Panthor)	**99.7%**	96.9%
Toy (Stridor)	**97.8%**	94.0%
Stuffed Cat	**100.0%**	98.3%
Duck	**89.9%**	81.6%
Dwarf	**87.7%**	67.6%
Mouse	**94.6%**	89.1%
Owl	**97.3%**	60.5%
Elephant	**98.6%**	94.7%
Samurai	97.7%	**98.5%**
Sculpture 1	**92.5%**	82.7%
Sculpture 2	99.9%	**100.0%**
Average	**95.6%**	88.2%
Median	**97.7%**	93.0%
Best	**100.0%**	**100.0%**
Worst	**81.5%**	60.5%

7 Additional experimental results

In this section we report results of additional experiments that proof the detection capability of our approach and its performance on occluded objects. Furthermore, we analyze the contribution each part of our energy formulation.

7.1 Detection Task

Our main experimental setup deals with the task of pose estimation of a known object in a RGB-D image. In many applications, the presence of an object is

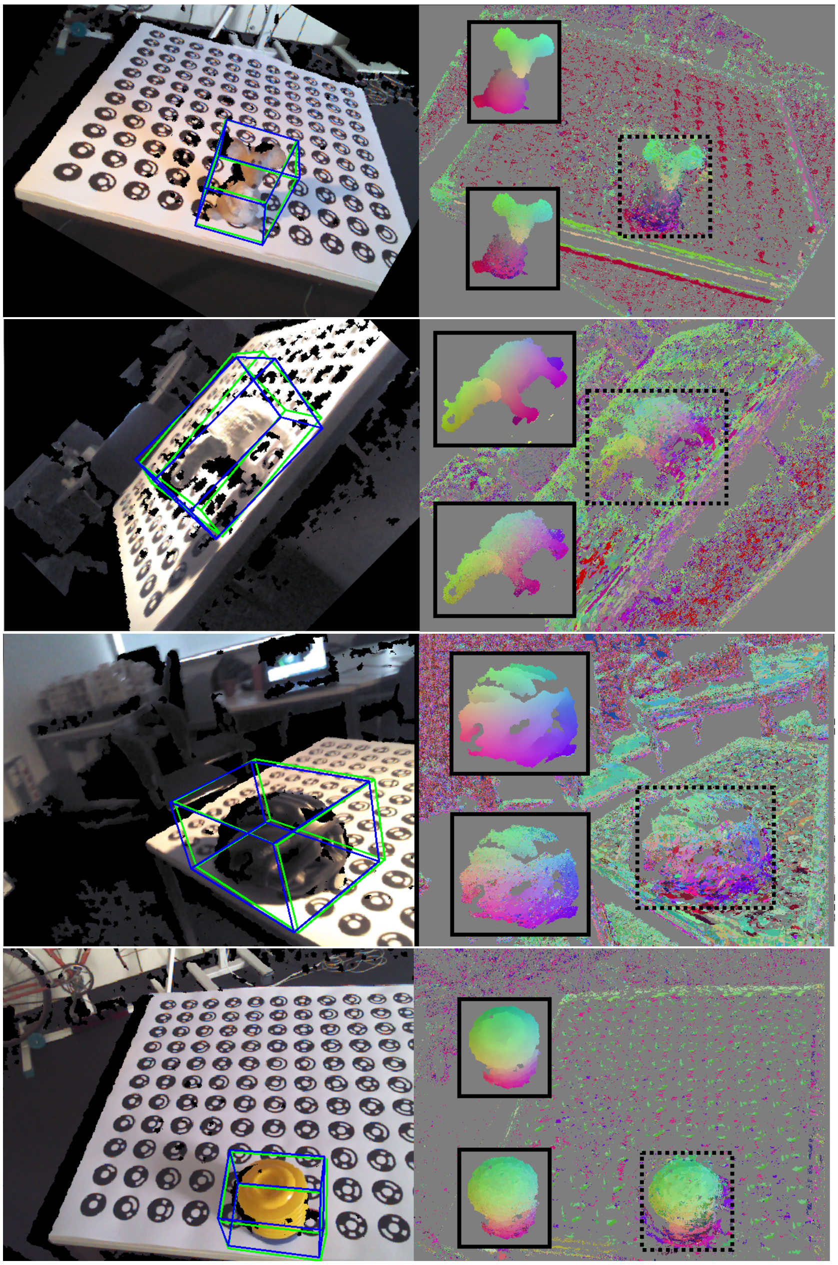

Fig. 4. Qualitative results on our dataset. Poses estimated with our system (blue bounding box) versus the ground truth pose (green bounding box). Next to each test image are the predicted object coordinates **y** from one tree of the forest. The inlay figures show the ground truth object coordinates (top) and the best object coordinates (bottom), where “best” is the best prediction of all trees with respect to ground truth (for illustration only).

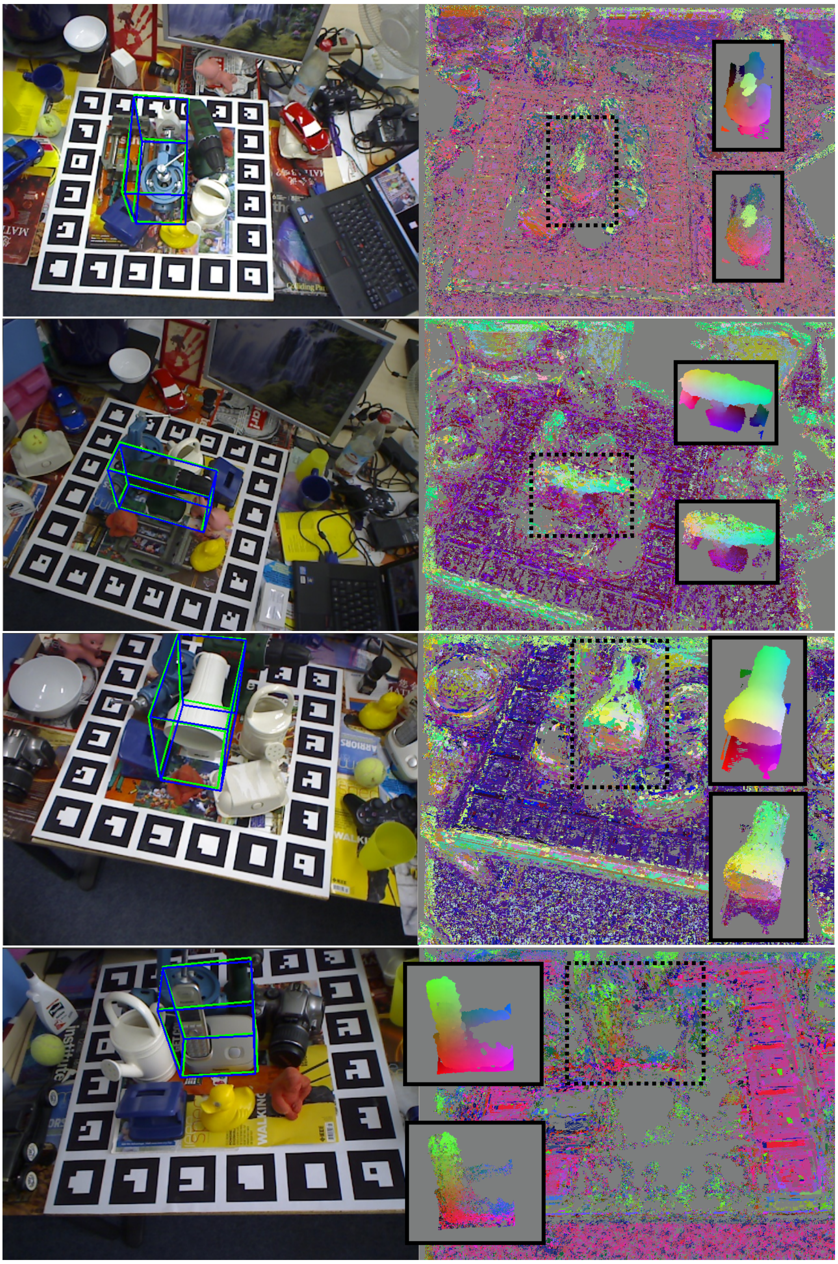

Fig. 5. Qualitative results on the Hinterstoisser dataset. Poses estimated with our system (blue bounding box) versus the ground truth pose (green bounding box). Next to each test image are the predicted object coordinates $\mathbf{y}$ from one tree of the forest. The inlay figures show the ground truth object coordinates (top) and the best object coordinates (bottom), where "best" is the best prediction of all trees with respect to ground truth (for illustration only).

Table 2. Results on the Hinterstoisser et al. dataset with synthetic training data, real training data and different background models (plane, noise). We see that our approach is consistently superior to [2, 5].

	Synth. Training				Real Training	
	LINEMOD[2]	DTT-3D[5]	Our(plane)	Our(noise)	Our(plane)	Our(noise)
Ape	**95.8%**	95.0%	**95.8%**	85.4%	**91.1%**	89.2%
Bench V.	98.7%	98.9%	**100.0%**	98.9%	**100.0%**	99.7%
Cam	97.5%	98.2%	**99.6%**	92.1%	**98.7%**	95.5%
Can	95.4%	**96.3%**	95.9%	84.4%	**99.6%**	98.9%
Cat	99.3%	99.1%	**100.0%**	90.6%	**99.7%**	98.8%
Driller	93.6%	94.3%	99.5%	**99.7%**	99.9%	**100.0%**
Duck	**95.9%**	94.2%	**95.9%**	92.7%	**96.8%**	94.4%
Box	**99.8%**	**99.8%**	98.0%	91.1%	**100.0%**	99.2%
Glue	91.8%	96.3%	**98.9%**	87.9%	91.7%	**96.7%**
Hole P.	95.9%	97.5%	**99.4%**	97.9%	**99.6%**	99.0%
Iron	97.5%	98.4%	97.6%	**98.8%**	99.9%	**100.0%**
Lamp	97.7%	97.9%	**99.8%**	97.6%	**99.1%**	98.7%
Phone	93.3%	95.3%	**97.6%**	86.1%	**98.8%**	95.8%
Bowl	99.9%	99.7%	-	-	-	-
Cup	97.1%	97.5%	-	-	-	-
Avg.	96.6%	97.2%	**98.3%**	92.6%	**98.1%**	97.4%
Med.	97.1%	97.5%	**98.9%**	92.1%	**99.6%**	98.8%
Max.	99.9%	99.8%	**100.0%**	99.7%	**100.0%**	**100%**
Min.	91.8%	94.2%	**95.8%**	84.4%	**91.1%**	89.2%

unknown and has to be established first. This can be done by defining a threshold on the energy of the final pose estimate. The system would report a detection only if this energy is below the threshold. In the following experiment we evaluate the detection performance of this approach.

We perform this experiment on the Hinterstoisser [2] images since they contain dense clutter. In each of the 13 object image sets we only search for the corresponding object, although other objects might be present. This is because the Hinterstoisser dataset only provides ground truth for one object per set. We run our full pipeline to extract one hypothesis per image. We extract a 2D bounding box based on this hypothesis[3], following the detection evaluation setup in [5]. The bounding boxes of all images of a sequence are ranked according to their hypothesis' energy. The ground truth bounding box is extracted using the ground truth pose. As in [5], we consider a detection correct if the intersection over union of detected bounding box and ground truth bounding box is at least 70%. We generated precision-recall curves for each of the 13 Hinterstoisser objects and calculated the average precision[4], AP.In Fig. 6 we show precision-recall

[3] We render the object segmentation mask based on the pose hypothesis and use its bounding box.

[4] We calculate AP according to VOC2012[1].

curves for 4 representative objects, with large and small AP. The mean AP is 0.88, which proves sensible detection performance on the Hinterstoisser dataset.

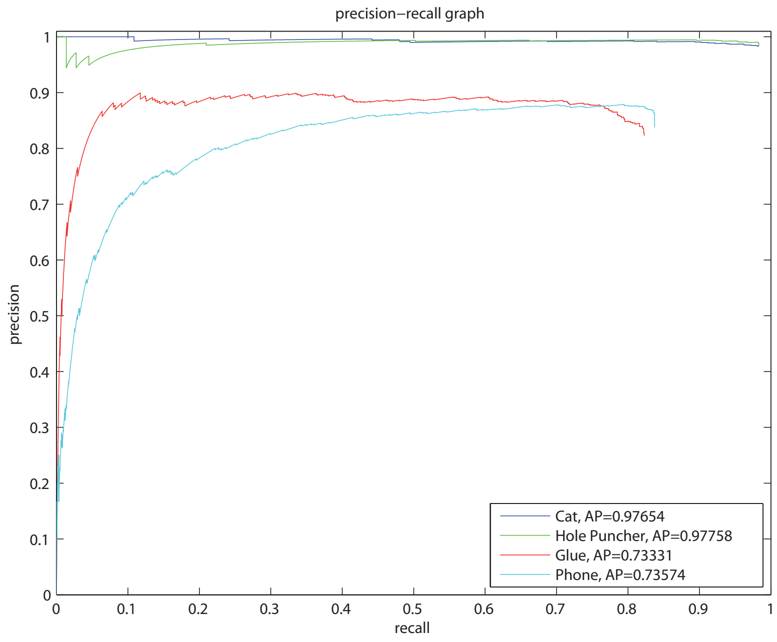

Fig. 6. Detection performance. Precision-recall curves of 4 different objects. The mean AP of all 13 objects is 0.88

7.2 Occlusion Dataset

Our dataset and the dataset of [2] are free of occlusions. While the objects annotated in the dataset of [2] are embedded in dense clutter, they are still fully visible in each frame. Hence, to demonstrate robustness against occlusion we created a new dataset. We annotated one sequence ("Bench Vise", ca. 1200 frames) of the dataset of [2] with 6DOF poses of 8 additional objects present in the scene. Depending on the viewing direction, these objects occlude each other to a large extent making this dataset very challenging. Fig. 7 shows one frame with all annotations marked, and a closeup of a heavily occluded object.

We annotated the sequence by initializing the pose of each object by hand, and propagating the object pose via the groundtruth transformation of each frame. If the propagation produced errors or when the object was moved within the scene we reinitialized by hand. For each frame, all poses were refined by ICP. We term this dataset *occlusion dataset* and make the annotation data publicly available.

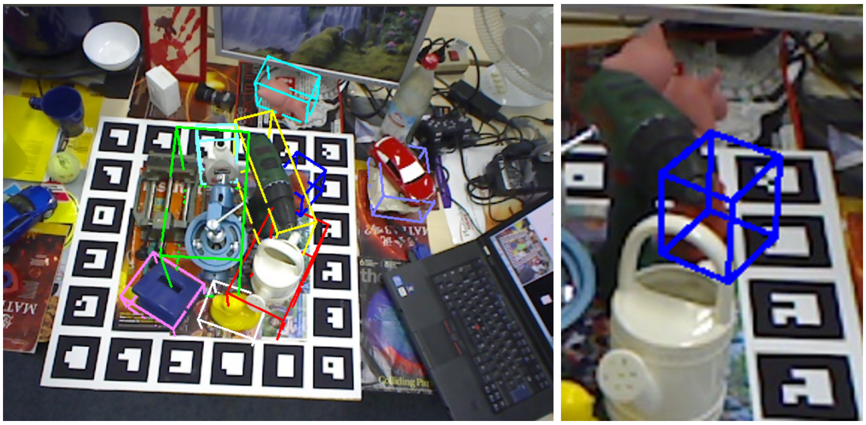

Fig. 7. Left: One frame of the Bench Vise sequence of the dataset of [2] annotated with poses of 8 additional objects (each shown as bounding box). The original dataset contains only the pose of one object (green bounding box). Note that some objects are occluded. Right: Our annotations also include heavily occluded objects (here Ape, blue bounding box).

We applied our full pose estimation approach to this challenging dataset, resulting in 67.3% average accuracy. Our re-implementation of [2] achieves only 54.4% average accuracy. Consequently, we demonstrate superior robustness to occlusion. Table 3 shows detailed results.

Table 3. Results on the occlusion dataset. We compare our full energy to an energy which uses depth only, and to the approach of [2].

	Full Energy	Depth C. Only	LINEMOD[2]
Ape	**62.6%**	51.9%	49.8%
Bench V.	**100.0%**	98.8%	98.7%
Can	80.2%	**98.8%**	51.2%
Cat	**50.0%**	27.7%	34.9%
Driller	**84.3%**	71.8%	59.6%
Duck	**67.6%**	57.8%	65.1%
Box	8.5%	2.4%	**39.6%**
Glue	**62.8%**	33.3%	23.3%
Hole P.	**89.9%**	71.5%	67.2%
Avg.	**67.3%**	57.1%	54.4%
Med.	**67.6%**	57.8%	51.2%
Max.	**100.0%**	98.8%	98.7%
Min.	8.5%	2.4%	**23.3%**

7.3 Contribution of Energy Terms

We conducted further experiments to reveal the contribution of our individual energy terms. We repeated pose estimation experiments on the dataset of [2], but using each energy component alone. Results are included in Table 4. The coordinate component and the object component alone give clearly inferior results. The depth component alone gives results comparable to the full energy. However, we repeated tests on the occlusion dataset introduced above, and observe that, here, the depth component alone achieves only 57.1% average accuracy instead of 67.3% with our full energy (see third column of Table 3).

Furthermore, we include an additional baseline: Similar to [7] we used the percentage of inlier pixels in $M_c^L(H_c)$ (see Sec. 3.2 of the main paper) instead of our energy to rate hypotheses. Inliers are defined as in Sec. 3.3 of the main paper. We observe 40.3% average accuracy on the dataset of [2], which clearly demonstrates that our energy formulation is superior.

Table 4. Results on the dataset of [2] with different variants of our energy.

	Full Energy	Depth C. Only	Obj. C. Only	Coord. C. Only	Inlier Energy[7]
Ape	**95.8%**	88.8%	75.5%	79.8%	61.5%
Bench V.	**100.0%**	98.8%	50.6%	89.5%	37.7%
Cam	**99.6%**	96.9%	32.8%	85.3%	21.3%
Can	95.9%	**97.4%**	59.7%	70.0%	23.3%
Cat	**100.0%**	97.8%	77.6%	96.7%	47.1%
Driller	**99.5%**	99.1%	55.3%	78.4%	50.0%
Duck	**95.9%**	93.4%	49.2%	76.2%	42.9%
Box	**98.0%**	97.5%	52.6%	49.8%	27.3%
Glue	**98.9%**	95.3%	82.5%	58.7%	63.8%
Hole P.	**99.4%**	98.1%	25.6%	91.6%	34.5%
Iron	97.6%	**98.6%**	23.1%	80.5%	44.7%
Lamp	**99.8%**	**99.8%**	35.7%	91.6%	40.9%
Phone	**97.6%**	91.8%	15.1%	60.3%	29.1%
Avg.	**98.3%**	96.4%	48.9%	77.6%	40.3%
Med.	**98.9%**	97.5%	50.6%	79.8%	40.9%
Max.	**100.0%**	99.8%	82.5%	96.7%	63.8%
Min.	**95.8%**	88.8%	15.1%	49.8%	21.3%

8 Supplementary Video

In our supplementary video we show pose estimation results for single and multiple objects under different lighting conditions and occlusion. We use the same forest as for the experiments on our dataset in the main paper. It was trained with *bright* and *dark* training sets of all of our 20 objects.

In order to obtain more precise pose estimates we take the hypothesis with the best energy after refinement and use it as initialization for local optimization of our energy function using a general purpose optimization algorithm. This would not give substantially different results in our experiments due to the pose tolerance thresholds, but the visual result is more pleasing.

In addition we run this local optimization for the pose estimated in the last frame, and add the refined pose to the hypotheses pool of the current frame. We use the pose with lowest energy as the pose estimate displayed.

References

1. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html
2. Hinterstoisser, S., Lepetit, V., Ilic, S., Holzer, S., Bradski, G., Konolige, K., Navab, N.: Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes. In: ACCV. (2012)
3. Lepetit, V., Fua, P.: Keypoint recognition using randomized trees. IEEE Trans. on PAMI. 28(9) (2006)
4. Newcombe, R., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A., Kohli, P., Shotton, J., Hodges, S., Fitzgibbon, A.: KinectFusion: Real-time dense surface mapping and tracking. In: ISMAR. (2011)
5. Rios-Cabrera, R., Tuytelaars, T.: Discriminatively trained templates for 3D object detection: A real time scalable approach. In: ICCV. (2013)
6. Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., Blake, A.: Real-time human pose recognition in parts from a single depth image. In: CVPR. (2011)
7. Shotton, J., Glocker, B., Zach, C., Izadi, S., Criminisi, A., Fitzgibbon, A.: Scene coordinate regression forests for camera relocalization in rgb-d images. In: CVPR. (2013)
8. Wu, C., Clipp, B., Li, X., Frahm, J., Pollefeys, M.: 3D model matching with viewpoint-invariant patches (VIP). In: CVPR. (2008)