Detect Globally, Label Locally: Learning Accurate 6-DOF Object Pose Estimation by Joint Segmentation and Coordinate Regression

Apurv Nigam, Adrian Penate-Sanchez, and Lourdes Agapito

***Abstract*—Coordinate regression has established itself as one of the most successful current trends in model-based 6 degree of freedom (6-DOF) object pose estimation from a single image. The underlying idea is to train a system that can regress the three-dimensional coordinates of an object, given an input RGB or RGB-D image and known object geometry, followed by a robust procedure such as RANSAC to optimize the object pose. These coordinate regression based approaches exhibit state-of-the-art performance by using pixel-level cues to model the probability distribution of object parts within the image. However, they fail to capture global information at the object level to learn accurate foreground/background segmentation. In this letter, we show that combining global features for object segmentation and local features for coordinate regression results in pixel-accurate object boundary detections and consequently a substantial reduction in outliers and an increase in overall performance. We propose a deep architecture with an instance-level object segmentation network that exploits global image information for object/background segmentation and a pixel-level classification network for coordinate regression based on local features. We evaluate our approach on the standard ground-truth 6-DOF pose estimation benchmarks and show that our joint approach to accurate object segmentation and coordinate regression results in the state-of-the-art performance on both RGB and RGB-D 6-DOF pose estimation.**

***Index Terms*—Object detection, segmentation and categorization, deep learning in robotics and automation.**

I. Introduction

ESTIMATING the six degree of freedom (6-DOF) pose of the instance of an object of known geometry is a fundamental task for many robotics applications. In many cases, the objects that robots interact with will not exhibit rich visual textures and for this reason the pose estimation task cannot depend on the recognition of point-based features such as hand-crafted features or even features learnt from data. Although, in the case of RGB-D input images, the availability of depth data allows to obtain more accurate poses, in essence, the problem to be solved is the same as in the RGB-only case: first the algorithm should identify which pixels are part of the object and which are background or clutter; next their coordinates should be regressed and finally a robust estimator, such as RANSAC [1] used to solve either a 2D–3D or 3D–3D registration problem.

Current top performing state-of-the-art approaches for 6-DOF object instance pose estimation perform coordinate regression to recognize the different parts of the object in the image [2], [3]. The key idea is not to predict the object pose directly but to first regress an intermediate representation in object coordinates. This leads to a labelling problem where each pixel is associated a label that indicates which object part it belongs to. Once object parts are identified, correspondences can be established with the 3D representation, usually a 3D model, and the 6-DOF pose can be solved for using geometric validation. The main challenge with coordinate regression approaches is to minimize the effect that incorrect object part detections have in the final pose estimate. In [2] the authors introduced a formulation that exploited local features for dense part labelling which modeled the probability distribution over the parts of the object as well as the background. This enabled the system to learn how to discern if a pixel was part of the background or the object, as well as which object part it belonged to (or in other words, what might its coordinates/position be within the object). The main drawback with this approach is that modeling this probability distribution at the local, pixel-level, rather than at a global object-level, can lead to ambiguities and to numerous incorrect object-part labellings over the input image.

In this letter we argue the use of global learned features to discover a pixel-accurate segmentation of the image into foreground/background pixels and, in contrast, local features to regress object coordinates. Our new approach combines a dense fully convolutional segmentation neural network, that is trained to classify pixels into background and foreground object, and a second network that learns to regress object coordinates given the output segmentation mask from the first network. While the fully convolutional segmentation network naturally makes use of global image information, which results in accurate segmentation boundaries (and consequently fewer misclassified pixels), the coordinate regression focuses on

Manuscript received February 24, 2018; accepted July 9, 2018. Date of publication July 23, 2018; date of current version August 8, 2018. This letter was recommended for publication by Associate Editor A. Argyros and Editor T. Asfour upon evaluation of the reviewers' comments. This work was supported by the Second Hands project, funded by the EU Horizon 2020 Research and Innovation programme under Grant 643950 and by the EPSRC under Grants EP/R026084/1 and EP/R026173/1. *(Corresponding author: Adrian Penate-Sanchez.)*

A. Nigam is with the Department of Computer Science, University College London, London NW1 9HZ, U.K., and also with the ANI Technologies Pvt. Ltd., Dehradun 248006, India (e-mail: apurv.nigam.16@ucl.ac.uk).

A. Penate-Sanchez is with the Department of Computer Science, University College London, London NW1 9HZ, U.K., and also with the Oxford Robotics Institute, University of Oxford, Oxford OX2 6NN, U.K. (e-mail: andarinneo@gmail.com).

L. Agapito is with the Department of Computer Science, University College London, London NW1 9HZ, U.K. (e-mail: l.agapito@cs.ucl.ac.uk).

This letter has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors. The Supplementary Materials contain a video of the experiments in the paper. This material is 25.6 MB in size.

Digital Object Identifier 10.1109/LRA.2018.2858446

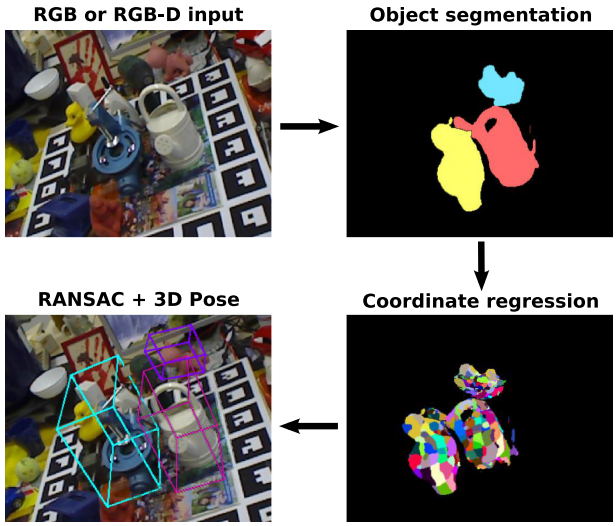

Fig. 1. Given an input RGB or RGB-D image our approach combines the use of global image cues for foreground/background object segmentation and local cues for coordinate regression (formulated as a discrete labeling problem). The use of a fully convolutional architecture for the segmentation task results in pixel-accurate object boundaries that lead to fewer outliers in the coordinate regression task and ultimately more accurate 6-DOF poses.

pixel-level evidence to associate pixels with object parts. Since background pixels will no longer be present, far fewer outlier matches will be effectively handled by the pose estimation algorithm. An overview of our approach can be seen in Fig. 1.

Our quantitative evaluation on benchmark datasets shows that our approach outperforms state of the art methods in RGB-D pose estimation of texture-less objects and manages to obtain very reliable results when extending this approach to RGB only inputs. Crucially, our method leads to fewer foreground/background segmentation errors which in turn results in improvements in the average percentage of accurately estimated poses of 12% over previous baselines. Further, our approach is able to halve the median rotation estimation error of its closest competitor [4].

Our work as another clear example of the profound impact of deep learning on robotics applications in multiple areas like recognizing change detection in SLAM [5], performing part affordance detection [4], [6] or camera re-localization [7], [8].

II. Related Work

Traditionally, the most popular approaches to 6-DOF instance object pose estimation were based on detecting keypoint features coupled with a robust estimation scheme such as RANSAC [1]. However, an important limitation of this widely tested, efficient and robust solution is that it requires the objects to be highly textured. The sudden availability of inexpensive low cost depth sensors enabled a surge of new methods suitable for the more challenging problem of texture-less object pose estimation taking advantage of the depth channel. Our literature review will therefore focus on methods that can cope with texture-less objects using RGB-D or even RGB-only images.

In a series of papers [9], [10] Hinterstoisser *et al.* introduced solutions to both the RGB and RGB-D cases that relied on the calculation of templates based on RGB boundaries and depth normals to then perform a dictionary matching scheme to find the pose of the object. In [11] the template features from [9] were used in combination with a Hough forest modified to add robustness to clutter and to perform simultaneous 3D object detection and pose estimation. Later, Brachmann *et al.'s* work [2], [3] proposed solutions that could be used on both RGB and RGB-D scenarios with minor alterations. They were inspired by [12], using similar feature representations and random forests in a new formulation to obtain state of the art results on object pose estimation. Their approach proposes to split the object into parts, and then formulates the recognition of object parts in an input image as a labelling problem. Similarly to our approach, they rely on the use of robust estimators [1] and classical pose estimation approaches [13] to estimate the final pose of the object.

RGB-D sensors are specially important in robotics as the presence of 3D data alongside the color input has managed to achieve great improvements in many vital tasks. This increase in performance is in several cases the difference between being capable of relying on vision to control the robot or not. Examples of this increase in performance are palpable in visual odometry [14], environment mapping [15] or SLAM [16], [17]. RGB-D cameras offer that extra bit of performance to make even tasks as complex as understanding how to interact physically with the environment [18].

The use of CNN-based approaches has recently become the established trend in many robotics applications – a good example of this is Maturana *et al.'s* [19], [20] approach to object recognition in RGB-D data – and has inevitably become the dominant paradigm in 6-DOF object pose estimation. We also draw inspiration from deep learning architectures to solve semantic segmentation [21] and image classification [22] problems. In [21] the authors proposed an approach that can robustly estimate pixel labelling by learning from semantic annotations and the use of de-convolutional layers to upscale and merge progressively coarse to fine results.

Recent work by Porzi *et al.* [23] proposes a CNN-based approach to coordinate regression to obtain the 6-DOF pose of an object in RGB-D data. Wolhart *et al.* [24] instead focus on learning to map input images to descriptors that can be used in nearest neighbor search using Euclidean distance. The aim is to enforce feature descriptors from the same object and with similar poses to be similar while those between features arising from different poses are forced far apart in feature space. Kehl *et al.* [25] used regressed descriptors of locally sampled RGB-D patches to perform a voting scheme in pose space. They create a dictionary of feature-pose pairs by training a convolutional auto-encoder [26] on depth invariant RGB-D patches. Object-class, as opposed to instance specific, pose estimation is tackled in [27] by learning semantic descriptors for each part of an object category and then solving the pose with a deformable shape model. Recently [28] proved that there is much to gain from doing using segmentation together with coordinate regression, in their case to improve the estimation of 3D flow between consecutive images.

To sum up, our main contribution is a novel dual-network deep architecture with an instance-level object segmentation network that exploits global image information for object/background segmentation and a pixel-level classification network for coordinate regression based on local features. Our approach can be

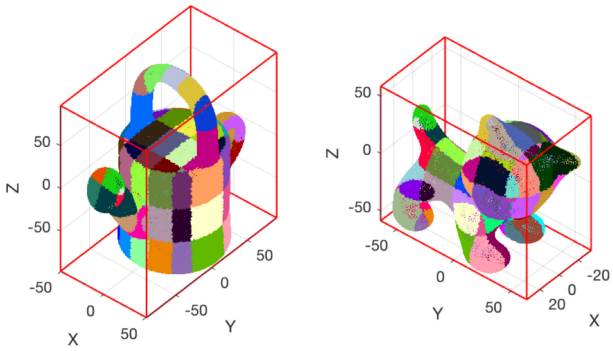

Fig. 2. Binning of object coordinates in the reference 3D models. We show two examples of object coordinate clusterings (object-part labellings). The coordinate regression problem is re-cast as a labelling problem where, at test time, each pixel is associated with an object-part label.

used for RGB-D or RGB-only images and outperforms all other approaches on the standard LINEMOD benchmark [9].

III. Method

Our work focuses on 6-DOF pose estimation in the case of texture-less objects when knowledge of their 3D shape is provided. The assumption is that a 3D model or a 3D point cloud is known for the objects that we seek to find.

A. Discretization of Object Coordinates

Given a test image containing one of the query objects, our solution is based on predicting the 3D coordinates on the corresponding 3D model for every 2D image pixel depicting the object. Once the matches between 2D coordinates in the test scene and 3D coordinates in the reference model have been established, the rigid transformation that explains their relative pose can be easily estimated. We formulate the problem of finding the corresponding 3D object coordinates for each pixel in the input image as a multi-class classification problem. We denote the input RGB image $I \in \mathbb{R}^{w \times h \times 3}$, its associated depth image $D \in \mathbb{R}^{w \times h}$ (in the case of RGB-D input) and the set of 3D model point coordinates $X_i^m = (x_i, y_i, z_i)$.

Given the ground truth transformation $[R|t]$ between points in the object X_i^m and the camera X_i^c coordinate systems, and known camera calibration parameters K, the 3D points X_i^m can be projected onto the image plane to establish 2D-3D correspondences.

The span of 3D object coordinates for each model $(x_{min} : x_{max}, y_{min} : y_{max}, z_{min} : z_{max})$ is discretized along each direction into N bins such that the 3D coordinate space is divided into bins $B_j \in N^3$. We used 5 bins per axis which gives $5 \times 5 \times 5 = 125$ discrete bins (as seen in Fig. 2), with an additional bin used to label background pixels. We train a CNN to predict, for every pixel on the object, the id of the bin of its corresponding object coordinate. Which implies that, $\forall (u, v) \in I \Rightarrow \exists B_j$; it is important to underline the fact that there exists a unique labeling per image and that all pixels in the image have a label. Fig. 2 illustrates the binning of two objects from [9]. The number of bins used has been chosen empirically from our observations, in any case small changes to the binning size should not affect the performance dramatically.

B. Object Segmentation

Fig. 3 shows the architecture of our multi-stage CNN pipeline. The upper part of the figure shows the Fully Convolutional Network (FCN) used to predict the probabilities of pixels in the input image being background or one of the object instances. Our FCN maps an RGB image $I \in \mathbb{R}^{w \times h \times 3}$ to a probability distribution $H \in \mathbb{R}^{w \times h \times (n+1)}$ where n is the number of objects in the scene, $n + 1$ to account for the background.

Our FCN was built using VGG16 as an encoder whose last fully connected layers were followed by a deconvolution layer to up-scale the convolutional responses to the original size of the input. To preserve the finer details in the upscaled output of the FCN, we combined the output of the last fully connected layer $fc7$ with finer features from $pool4$ and $pool3$ layers as suggested by [21] as FCN-8. More deeper architectures could have been used but we selected VGG16 as a compromise between computation cost and performance.

The training data for the FCN was generated synthetically. Ground truth segmentation masks were generated using ground truth poses. The training error was chosen to be the average soft-max cross entropy loss between the ground truth and network prediction.

$$L(w) = -\frac{1}{N} \sum_{k=1}^{n} \left[y_n \log(\hat{y_n}) - (1 - y_n) \log(1 - \hat{y_n}) \right]$$

The final image segmentation I_{seg} is obtained by thresholding the soft-max output of the up-scaled features. This stage contributed greatly to avoid potential object outliers which should not be part of the object coordinate regression and the final robust pose estimation. We can see an example of an actual segmentation in the top right image in Fig. 4

C. Object Coordinate Regression

The lower half of Fig. 3 shows the second stage of our CNN architecture used to predict the object coordinate bin labels $\forall$(u,v) $\in I_{seg}$ where $I_{seg}(u, v) \neq Background$. Our choice of architecture is based on the standard AlexNet [22]. We train on patches randomly sampled from I_{seg} where $I_{seg} =$ Object Class and $I_{seg} = Background$. For every patch that we sample from the object, the ground truth label for that patch is chosen to be the object coordinate bin id of the center pixel. The network is trained to minimize the softmax cross entropy loss between the ground truth label and the predicted probabilities for each class. To obtain good classification results the number of patches sampled from the background must be similar to those sampled from other bins. This makes sense in our approach as most of the background pixels have been removed in the object segmentation. It is due to this reason that we select very few training patches from regions outside the segmentation mask of the object of interest.

To achieve depth invariance, patch sizes are scaled during training and testing based on the distance to the object. We first set an original patch size of $w \times h$, with a distance to the object of d; when creating a new patch we select the center point, that has a distance to the object d_i, and we scale the patch relative

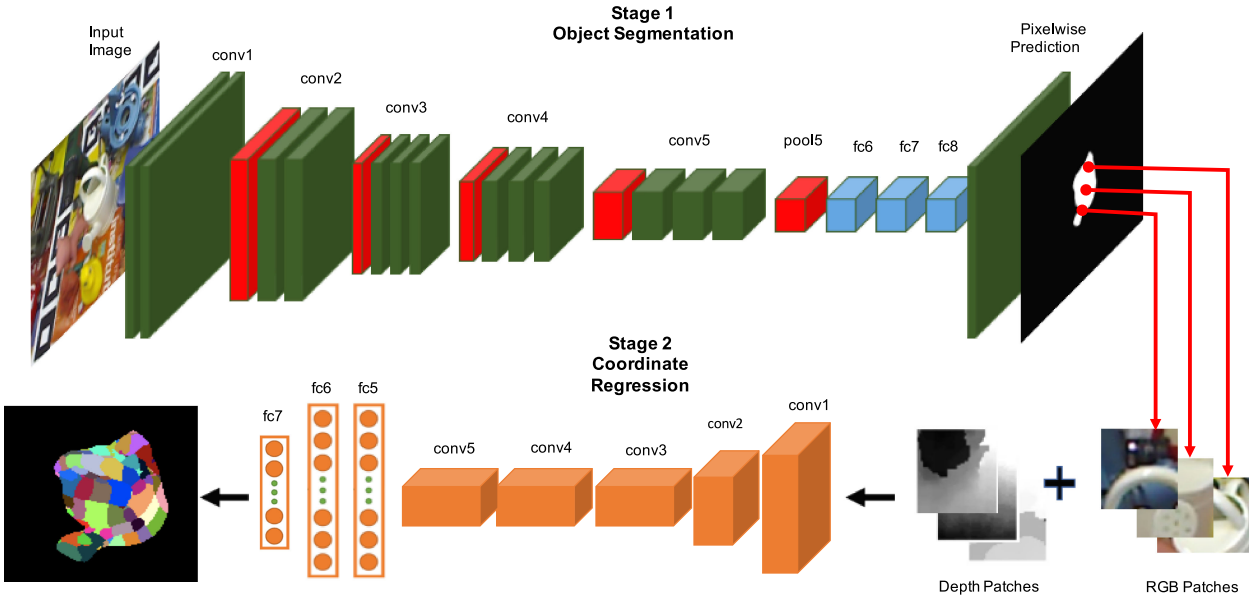

Fig. 3. Multistage CNN architecture: A fully convolutional network (FCN) predicts the segmentation mask of the object. In the second stage, object-part labels are predicted for pixels inside the segmentation mask.

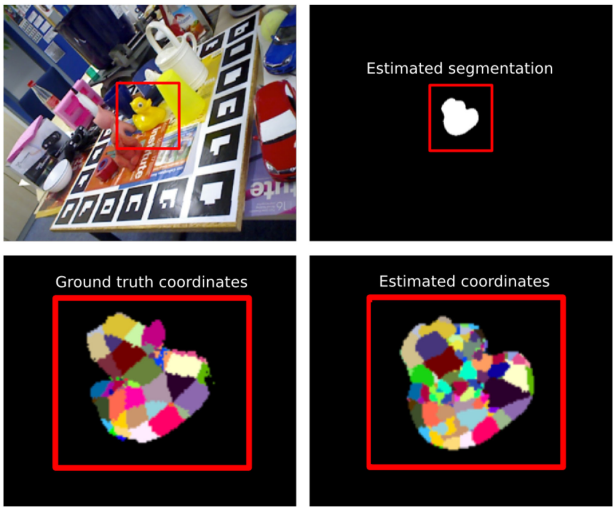

Fig. 4. Example outputs from segmentation and coordinate regression steps. (Top left) Input image. (Top right) Estimated segmentation mask. (Bottom left) Ground truth coordinates. (Bottom left) Estimated coordinates.

to the distance as follows

$$w_i = \frac{w * d}{d_i},\ h_i = \frac{h * d}{d_i}$$

An example of the estimated segmentation for a test input image is shown in Fig. 4 (top right). The bottom left image shows the ground truth coordinate bin labelling for that test scene while the bottom right shows the estimated coordinates.

D. RANSAC Pose Estimation

The final step in our approach is the estimation of the 6-DOF pose of the object instance using the object class label and the regressed object coordinates. When testing on RGB-D images, the availability of depth information allows us to formulate this problem as the estimation of the rigid transformation between corresponding pairs of 3D points. We use the Kabsch Algorithm [13] to calculate the object's rotation and translation of the object. In the case of RGB images, we estimate the rigid transformation between the set of corresponding 2D-3D points using the Direct Linear Transformation method. When selecting image points for potential matches we only consider those inside the estimated object segmentation mask.

In both cases, to improve the robustness of our approach we opted to embed the estimation within a preemptive RANSAC [32] framework. Our sampling strategy is as follows: points in the test image are chosen such that no two pixels have the same object coordinate labels, we ensure that point sets are not collinear and that the distance between any two points is always greater than a threshold of 1 cm. Finally, points are reprojected onto the image plane using the estimated pose and rejected as a hypothesis if their consensus is lower than an inlier threshold of 100. We also calculate the 2D bounding box and discard hypothesis that occupies less than 400 pixels. Meeting the previous criteria we generate 100 pose hypothesis using a 1000 points that we will score and rank while discarding the worst half and increasing iteratively the number of points used to calculate the reprojection error by a 100 points in each iteration. We keep discarding the worst half in each step and continue until we are left with the best candidate.

IV. Experiments

The evaluation has been carried out on the standard benchmark for 6-DOF pose estimation, Hinterstoisser *et al.*'s dataset [10] — a public dataset of RGB-D images of textureless objects in a cluttered scene — and compared against several state of the art algorithms. The dataset contains approximately 1200 images of each object instance (with the object depicted in the center of the image) and ground truth 6D pose labels are available. The dataset also provides 3D point clouds for the query objects. Our work has focused on the RGB-D scenario but results for RGB inputs are also shown to demonstrate that the ideas of this letter are not specific to a single problem. Two different metrics have been used to evaluate the quality of the

TABLE I
NETWORK ARCHITECTURES: DETAILED DESCRIPTION OF THE LAYERS USED IN OUR OBJECT SEGMENTATION AND COORDINATE REGRESSION NETWORKS

Coordinate regression	c[55,96],p[3,2],c[27,256],p[3,2],c[13,384],c[13,384],c[13,256],p[3,2],f[4096,1],f[4096,1],f[125,1]
Object Segmentation	c[224,64],c[224,64],p[2,2],c[112,128],c[112,128],p[2,2],c[56,256],c[256,256],c[56,256],p[2,2],c[28,512], c[28,512],c[28,512],p[2,2],c[14,512],c[14,512],c[14,512],p[2,2],f[4096,1],f[4096,1] , d[224,2]

Convolution: c[size, filters]; Fully Connected: f[size, filters]; Pooling: p[size, stride]; Deconvolution: d[size, numClasses];

TABLE II
6-DOF POSE ESTIMATION RESULTS ON RGB-D INPUT IMAGES ON THE STANDARD LINEMOD BENCHMARK DATASET [9]: COMPARISON WITH A LARGE NUMBER OF COMPETITORS USING THE METRIC DEFINED IN [10]. OUR APPROACH OUTPERFORMS ALL BASELINES IN ALL BUT ONE SEQUENCE

RGB-D 3D Detection								
	6-DOF Pose: Mean point error less than 10% object diameter [10]							
Sequence (# pics)	Drost CVPR 10 [29]	Linemod ACCV 12 [10]	DTT ICCV 13 [30]	Brachmann ECCV 14 [2]	Hinterstoisser ECCV 16 [31]	Brachmann CVPR 16 [3]	Porzi IROS 17 [23]	**Ours**
Ape (1235)	86.5%	95.8%	95.0%	85.4%	98.5%	98.1%	98.8%	**99.2%**
Benchvise (1214)	70.7%	98.7%	98.9%	98.9%	99.8%	99.0%	99.6%	**100%**
Camera (1200)	78.6%	97.5%	98.2%	92.1%	99.3%	99.7%	94.2%	**100%**
Can (1195)	80.2%	95.4%	96.3%	84.4%	98.7%	99.7%	92.1%	**100%**
Cat (1178)	99.1%	99.3%	99.1%	90.6%	99.9%	99.1%	89.9%	**100%**
Duck (1253)	46.0%	95.9%	94.2%	92.7%	98.2%	96.2%	96.8%	**99.6%**
Iron (1151)	84.9%	97.5%	98.8%	98.8%	98.3%	99.9%	97.0%	**100%**
Lamp (1226)	93.3%	97.7%	97.9%	97.6%	96.0%	**99.5%**	95.1%	97.4%
Average	79.9%	97.2%	97.3%	92.6%	98.6%	98.9%	95.4%	**99.53%**

TABLE III
ACCURACY OF ESTIMATED 6-DOF POSES USING RGB-D INPUTS: EVALUATION ON THE LINEMOD DATASET [9] USING SHOTTON *et al.*'S METRIC [12] (PERCENTAGE OF IMAGES WITH LESS THAN 5 CM AND 5° ERROR IN THE POSE CALCULATION) COMPARING WITH OUR CLOSEST COMPETITOR [3] FROM TABLE II

RGB-D - Percentage of accurate poses		
	6D Pose 5cm5° [12]	
Sequence (# pics)	Brachmann CVPR 16 [3]	**Ours**
Ape (1235)	59.0%	**80.9%**
Benchvise (1214)	92.9%	**96.6%**
Camera (1200)	92.8%	**94.5%**
Can (1195)	89.6%	**98.7%**
Cat (1178)	80.1%	**97.5%**
Duck (1253)	52.1%	**86.4%**
Iron (1151)	96.9%	**100%**
Lamp (1226)	91.7%	**96.6%**
Average	81.89%	**93.9%**

TABLE IV
MEDIAN ROTATION AND TRANSLATION ERRORS ON THE LINEMOD DATASET [9] FOR OUR APPROACH AND OUR CLOSEST COMPETITORS [2], [3], [23]. OUR APPROACH HALVES THE ROTATION ERROR AND ACHIEVES AN IMPROVEMENT OF 2 MM IN TRANSLATION ERROR OVER THE SECOND BEST PERFORMING METHOD [23]

RGB-D	**Median angle**	**Median distance**
Brachmann14 [2]	7.94°	0.94 mm
Brachmann16 [3]	4.90°	n.a.
Porzi17 [23]	4.66°	0.58 cm
Ours	**2.10°**	**0.38 cm**

TABLE V
ACCURACY OF ESTIMATED 6-DOF POSES USING RGB-ONLY INPUTS: EVALUATION ON THE LINEMOD DATASET [9] USING SHOTTON *et al.*'S METRIC [12] (PERCENTAGE OF IMAGES WITH LESS THAN 5 CM AND 5° ERROR IN THE POSE CALCULATION) COMPARING WITH [3] AND AN ALGORITHM BUILT ONLY FOR RGB INPUTS [33]

RGB - Percentage of accurate poses			
	6D Pose 5cm5° [12]		
Sequence	Brachmann 2016 [3]	**Ours**	BB8 [33]
Ape	34.4%	47.7%	80.2%
Bench	40.6%	37.9%	81.5%
Cam	30.5%	31.5%	60.0%
Can	48.4%	48.5%	76.8%
Cat	34.6%	37.4%	79.9%
Duck	22.0%	52.8%	53.2%
Iron	58.7%	41.6%	61.1%
Lamp	49.3%	51.9%	67.5%
Average	39.8%	43.7%	70.0%

Fig. 5. Example of a Augmented reality application of our approach using RGB-D images. We can see that the estimation of the pose is accurate enough to perform such tasks.

results. To further asses the results of our method and the accuracy of the retrieved poses we show qualitative results of our results in Fig. 5 and Fig. 6.

A. Training the Object Segmentation Network

To train the Object Segmentation Network we use around 70% of all images. Since this is a small number for the task of training a FCN, we use data augmentation techniques such as random flipping and rotation. For further robustness we added synthetic images to the training data by rendering the objects from different viewpoints using the given 3D models. We used pre-trained weights to initialize our network owing to the importance of good initialization of weights and biases. We observed that choosing a high learning rate was forcing the network to learn the segmentation of the object only when the object was

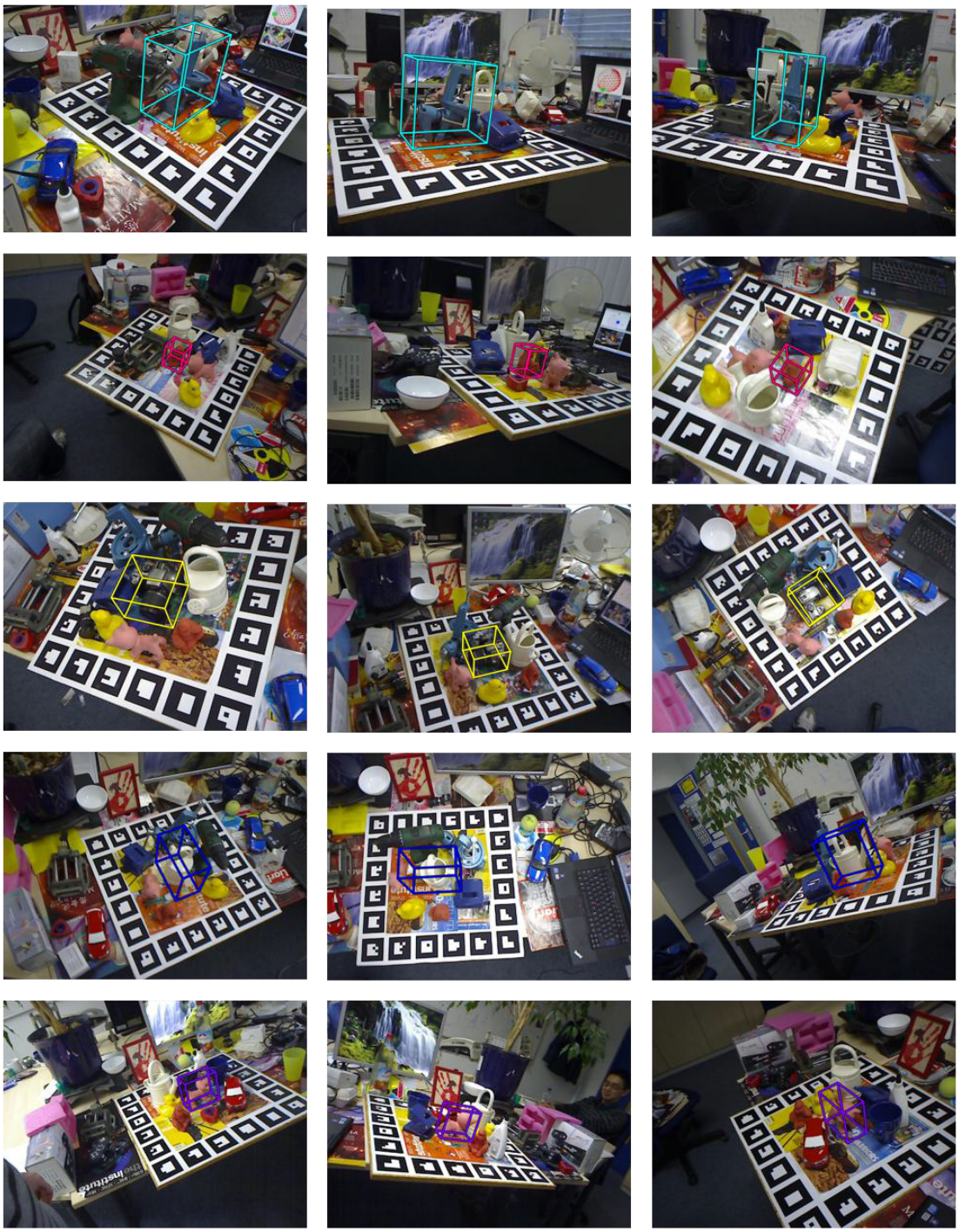

Fig. 6. Qualitative results of our 6D Pose Estimation Algorithm, estimated bounding boxes drawn around different object instances in different scenes using RGB-D images. The reprojected 3D bounding box of the object is shown to better asses the performance that can be expected from the proposed approach. The images show that our pose estimation approach is robust to heavily cluttered environments and large differences in viewpoint.

in the center of the image as in all training images. The learning rate and the batch size were adjusted to ensure that the network was able to detect objects placed in different image locations (i.e., not just at the centre of the image). Fig. 7 shows that the segmentation network works well even in highly occluded scenes. We used the Adam optimization algorithm, a batch size of 2 and learning rate of 10^{-4}. Table I shows the exact details of all the layers in the architecture.

B. Training the Object Coordinate Regression Network

We initialized the network using pre-trained weights from AlexNet, trained on 15 million images across 22000 categories. In the case of RGB-D images the 4^{th} channel was initialised using the mean of the R,G and B channels. Table I shows details of the exact architecture of the network. Deciding the number of patches to be sampled from each training image

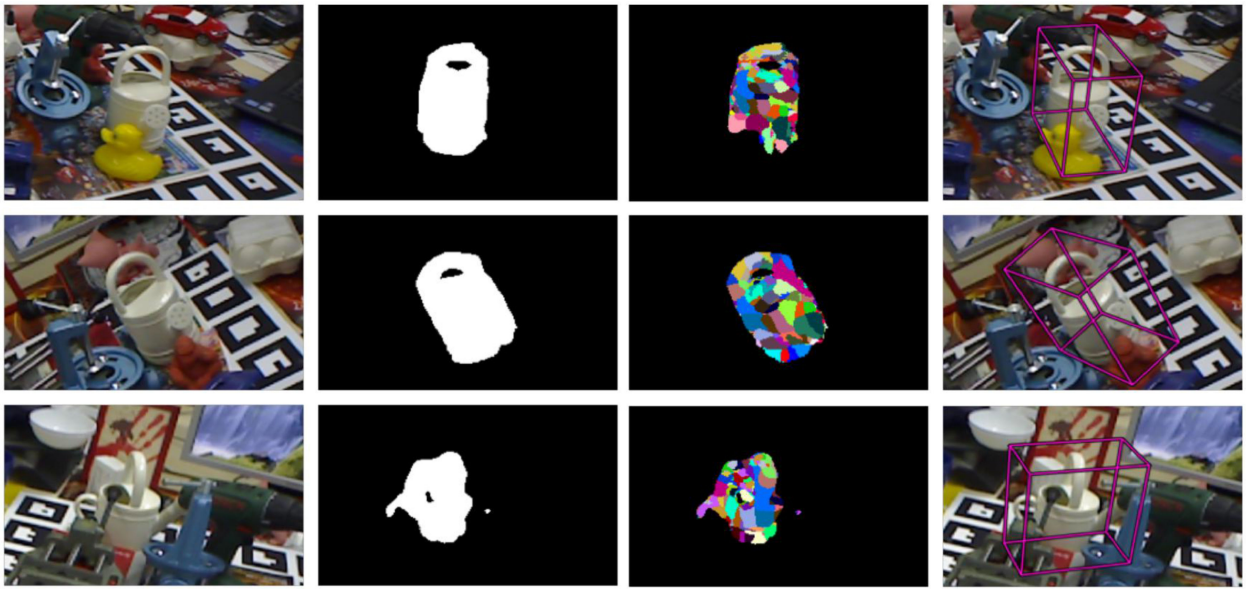

Fig. 7. This figure showcases the different situations that we encounter when using our approach. The first row shows the case in which our segmentation partially fails, we can see that the duck has been detected as part of the can, but the coordinate regression correctly estimates those pixels as background. The second row shows the case in which both the object segmentation and the coordinate regression fail to detect the incorrect pixels, but because we are using a robust estimator like RANSAC we can still filter out the error. The third row depict the case in which we correctly segment the object, we correctly regress the coordinates and we correctly estimate the pose.

needed some careful thought. Because not all labels are present in the images depicting each pose but the background is present in every scene, we chose to sample very few background patches from each scene so that the total number of background patches would be balanced with respect to the other 125 classes (the discretized object coordinate labels). We sampled 800 foreground and 6 background patches from each image. Other hyper-parameters of our network are the patch size, learning rate and batch size which were chosen after experimenting with several choices. Although we determine patch size according to depth of center pixel of the object to achieve depth invariance in detection, the patch size of one of the scenes was determined as reference and had to be chosen manually, we found 20×20 pixels to work best for us. We chose the learning rate to be 10^{-4} and a batch size of 10 in our experiments. For this network we used Stochastic Gradient Descent as the optimization algorithm.

C. Evaluation Metrics

Our evaluation uses both of the standard metrics defined in the literature. Hinterstoisser *et al.*'s metric [10] focuses mostly on the success of the 3D detection task. A detection is considered correct if the average distance between the 3D model points aligned with the ground truth pose or the estimated pose is less than 10% of the total size of the object. While this metric is quite relaxed in terms of actual accuracy, it is useful to determine whether or not the object has been correctly detected in 3D space. The results for RGB-D input data are shown in Table II.

The second metric, defined in [12], considers a pose to be correct if the error between the ground truth pose and the estimated pose is less than 5° and 5 cm. This metric is much more stringent in what it considers to be a correct pose and gives better insight into the actual accuracy of detections. The results for RGB-D input images are shown in Table III while Table V shows results on RGB inputs.

To further asses the quality of the estimated poses we provide the median rotation and translation errors. The median clearly shows the typical pose error that we might expect from our approach. These results are shown in Table IV.

D. Evaluation Results for RGB-D images

Table II shows a comparison with a large number of competing approaches [2], [3], [10], [23], [29]–[31] on the task of 3D detection using Hinterstoisser *et al.*'s metric [10]. Our approach achieves the best average 3D detection rate, outperforming all baselines in all but one object sequence. To further evaluate actual accuracy, we selected our closest competitor [3] from Table II and compared using Shotton *et al.*'s stricter metric [12]. Table III shows an average improvement of 12% over [3]. Finally, we assess the typical errors one can expect from our algorithm by calculating the median rotation and translation errors over all images (see Table IV). Our approach halves the rotation error of its closest competitor [4] (2.1° vs 4.6°) and achieves an improvement of 2 mm in translation error. Since all approaches use Kabsch's algorithm [13] to estimate the final pose, we conclude that our improvements in performance must be due to the pixel-accurate object boundary detections, leading to a substantial reduction in outliers, achieved thanks to our strategy of combining the object segmentation and coordinate regression tasks.

E. Evaluation Results for RGB Images

The extension of our algorithm to RGB input images is fairly straightforward with slight modifications to the architecture (to adapt from four to three input channels) and to the geometric validation step. In essence, a 2D–3D perspective pose estimation problem must be solved instead of 3D–3D. Once more we

show a comparison of our approach against the top performing baseline solves both for RGB and RGB-D [3] where we show an improvement of 5% in the average percentage of accurately estimated poses. We also show results against the best performing RGB only algorithm [33], in this work the focus is in refining the pose rather than creating a general approach that can be applied to both RGB and RGB-D cases. Another of the drawbacks of the approach in [33] is that the employ a holistic approach rather than a part based approach. Part based approaches are robust to partial occlusions by construction while [33] needs to introduce occlusion during the training process to be able to cope with them.

F. Robustness to Occlusion and Clutter

Fig. 7 showcases our algorithm's ability to cope with occlusions. Since we largely depend on correct prediction of object coordinate labels, the successful pose estimation in these tough scenarios can be attributed to the robust segmentation achieved with our segmentation network leading to higher accuracy in the coordinate regression step.

V. Conclusion and Future Work

We have shown that the use of global object segmentation and local labeling of coordinates leads to accurate estimations that can be exploited by classic geometric pose estimation. The numerous safeguards put in place to avoid outliers alleviates greatly the task that the final classic 6-DOF pose estimation algorithm needs to solve. The versatility of our approach that can provide accurate poses for both RGB and RGB-D input images is a clear strength. An example of its application of our approach to augmented reality can be seen in Fig. 5. Interesting future work would be to extend the successful features of our algorithm to the case of dealing with object classes/categories instead of object instances.

References

[1] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

[2] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother, "Learning 6D object pose estimation using 3D object coordinates," in *Proc. Eur. Conf. Comput. Vision*, 2014, pp. 536–551.

[3] E. Brachmann *et al.*, "Uncertainty-driven 6D pose estimation of objects and scenes from a single rgb image," in *Proc. Comput. Vision Pattern Recognit.*, 2016, pp. 3364–3372.

[4] L. Porzi, S. Rota-Bulo, A. Penate-Sanchez, E. Ricci, and F. Moreno-Noguer, "Learning depth-aware deep representations for robotic perception" *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 468–475, Apr. 2017.

[5] P. Alcantarilla, S. Stent, G. Ros, R. Arroyo, and R. Gherardi, "Street-view change detection with deconvolutional networks," in *Proc. Robot. Sci. Syst.*, 2016, pp. 1–22.

[6] A. Myers, C. L. Teo, C. Fermüller, and Y. Aloimonos, "Affordance detection of tool parts from geometric features," in *Proc. Int. Conf. Robot. Automat.*, 2015, pp. 1374–1381.

[7] N. Suenderhauf *et al.*, "Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free," in *Proc. Robot. Sci. Syst.*, 2015.

[8] A. Kendall and R. Cipolla, "Modelling uncertainty in deep learning for camera relocalization," *Proc. Int. Conf. Robot. Automat.*, 2016, pp. 4762–4769.

[9] S. Hinterstoisser *et al.*, "Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes," in *Proc. IEEE Int. Conf. Comput. Vision*, 2011, pp. 858–865.

[10] S. Hinterstoisser *et al.*, "Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes," in *Proc. Asian Conf. Comput. Vision*, 2013, pp. 548–562.

[11] A. Tejani, D. Tang, R. Kouskouridas, and T.-K. Kim, "Latent-class hough forests for 3D object detection and pose estimation," in *Proc. Eur. Conf. Comput. Vision*, 2014, pp. 462–477.

[12] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. W. Fitzgibbon, "Scene coordinate regression forests for camera relocalization in RGB-D images," in *Proc. Comput. Vision Pattern Recognit.*, 2013, pp. 2930–2937.

[13] W. Kabsch, "A solution for the best rotation to relate two sets of vectors," *Acta Crystallographica Section A: Cryst. Phys. Diffraction Theor. General Crystallogr.*, vol. 32, no. 5, pp. 922–923, 1976.

[14] M. Jaimez, C. Kerl, J. Gonzalez-Jimenez, and D. Cremers, "Fast odometry and scene flow from RGB-D cameras based on geometric clustering," in *Proc. Int. Conf. Robot. Automat.*, 2017, pp. 3992–3999.

[15] F. Steinbruecker, J. Sturm, and D. Cremers, "Volumetric 3D mapping in real-time on a CPU," in *Proc. Int. Conf. Robot. Automat.*, 2014, pp. 2021–2028.

[16] K. Yousif, Y. Taguchi, and S. Ramalingam, "MonoRGBD-SLAM: Simultaneous localization and mapping using both monocular and RGBD cameras," in *Proc. Int. Conf. Robot. Automat.*, 2017, pp. 4495–4502.

[17] J. McCormac, A. Handa, A. Davison, and S. Leutenegger, "Semanticfusion: Dense 3D semantic mapping with convolutional neural networks," in *Proc. Int. Conf. Robot. Automat.*, 2017, pp. 4628–4635.

[18] P. Kaiser, E. E. Aksoy, M. Grotz, and T. Asfour, "Towards a hierarchy of loco-manipulation affordances," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 2839–2846.

[19] D. Maturana and S. Scherer, "Voxnet: A 3D convolutional neural network for real-time object recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 922–928.

[20] A. Eitel, J. T. Springenberg, L. Spinello, M. Riedmiller, and W. Burgard, "Multimodal deep learning for robust RGB-D object recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 681–687.

[21] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. Comput. Vision Pattern Recognit.*, 2015, pp. 3431–3440.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[23] L. Porzi, A. Penate-Sanchez, E. Ricci, and F. Moreno-Noguer, "Depth-aware convolutional neural networks for accurate 3D pose estimation in RGB-D images," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 5777–5783.

[24] P. Wohlhart and V. Lepetit, "Learning descriptors for object recognition and 3D pose estimation," in *Proc. Comput. Vision Pattern Recognit.*, 2015, pp. 3109–3118.

[25] W. Kehl, F. Milletari, F. Tombari, S. Ilic, and N. Navab, "Deep learning of local RGB-D patches for 3D object detection and 6D pose estimation," in *Proc. Eur. Conf. Comput. Vision*, 2016, pp. 205–220.

[26] J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber, "Stacked convolutional auto-encoders for hierarchical feature extraction," in *Proc. Int. Conf. Artif. Neural Netw.*, 2011, pp. 52–59.

[27] G. Pavlakos, X. Zhou, A. Chan, K. G. Derpanis, and K. Daniilidis, "6-DOF object pose from semantic keypoints," in *Proc. Int. Conf. Robot. Automat.*, 2017, pp. 2011–2018.

[28] A. Behl, O. H. Jafari, S. K. Mustikovela, H. A. Alhaija, C. Rother, and A. Geiger, "Bounding boxes, segmentations and object coordinates: How important is recognition for 3D scene flow estimation in autonomous driving scenarios?" in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2593–2602.

[29] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: Efficient and robust 3D object recognition," in *Proc. Comput. Vision Pattern Recognit.*, 2010, pp. 998–1005.

[30] R. Rios-Cabrera and T. Tuytelaars, "Discriminatively trained templates for 3D object detection: A real time scalable approach," in *Proc. IEEE Int. Conf. Comput. Vision*, 2013, pp. 2048–2055.

[31] S. Hinterstoisser, V. Lepetit, N. Rajkumar, and K. Konolige, "Going further with point pair features," in *Proc. Eur. Conf. Comput. Vision*, 2016, pp. 834–848.

[32] D. Nister, "Preemptive ransac for live structure and motion estimation," in *Proc. IEEE Int. Conf. Comput. Vision*, 2003, pp. 321–329.

[33] M. Rad and V. Lepetit, "BB8: A scalable, accurate, robust to partial occlusion method for predicting the 3D poses of challenging objects without using depth," in *Proc. IEEE Int. Conf. Comput. Vision*, 2017, pp. 3848–3856.