Hodan, Tomas, Frank Michel, Eric Brachmann, Wadim Kehl, Anders GlentBuch, Dirk Kraft, Bertram Drost et al.
"BOP: Benchmark for 6D object pose estimation." In Proceedings of the European Conference on Computer Vision (ECCV),

BOP: Benchmark for 6D Object Pose Estimation

Tomáš Hodaň[1*], Frank Michel[2*], Eric Brachmann[3], Wadim Kehl[4]
Anders Glent Buch[5], Dirk Kraft[5], Bertram Drost[6], Joel Vidal[7], Stephan Ihrke[2]
Xenophon Zabulis[8], Caner Sahin[9], Fabian Manhardt[10], Federico Tombari[10]
Tae-Kyun Kim[9], Jiří Matas[1], Carsten Rother[3]

[1]CTU in Prague, [2]TU Dresden, [3]Heidelberg University, [4]Toyota Research Institute
[5]University of Southern Denmark, [6]MVTec Software, [7]Taiwan Tech
[8]FORTH Heraklion, [9]Imperial College London, [10]TU Munich

Abstract. We propose a benchmark for 6D pose estimation of a rigid object from a single RGB-D input image. The training data consists of a texture-mapped 3D object model or images of the object in known 6D poses. The benchmark comprises of: i) eight datasets in a unified format that cover different practical scenarios, including two new datasets focusing on varying lighting conditions, ii) an evaluation methodology with a pose-error function that deals with pose ambiguities, iii) a comprehensive evaluation of 15 diverse recent methods that captures the status quo of the field, and iv) an online evaluation system that is open for continuous submission of new results. The evaluation shows that methods based on point-pair features currently perform best, outperforming template matching methods, learning-based methods and methods based on 3D local features. The project website is available at `bop.felk.cvut.cz`.

1 Introduction

Estimating the 6D pose, i.e. 3D translation and 3D rotation, of a rigid object has become an accessible task with the introduction of consumer-grade RGB-D sensors. An accurate, fast and robust method that solves this task will have a big impact in application fields such as robotics or augmented reality.

Many methods for 6D object pose estimation have been published recently, e.g. [34,24,18,2,36,21,27,25], but it is unclear which methods perform well and in which scenarios. The most commonly used dataset for evaluation was created by Hinterstoisser et al. [14], which was not intended as a general benchmark and has several limitations: the lighting conditions are constant and the objects are easy to distinguish, unoccluded and located around the image center. Since then, some of the limitations have been addressed. Brachmann et al. [1] added ground-truth annotation for occluded objects in the dataset of [14]. Hodaň et al. [16] created a dataset that features industry-relevant objects with symmetries and similarities, and Drost et al. [8] introduced a dataset containing objects with reflective surfaces. However, the datasets have different formats and no standard evaluation methodology has emerged. New methods are usually compared with only a few competitors on a small subset of datasets.

[*] Authors have been leading the project jointly.

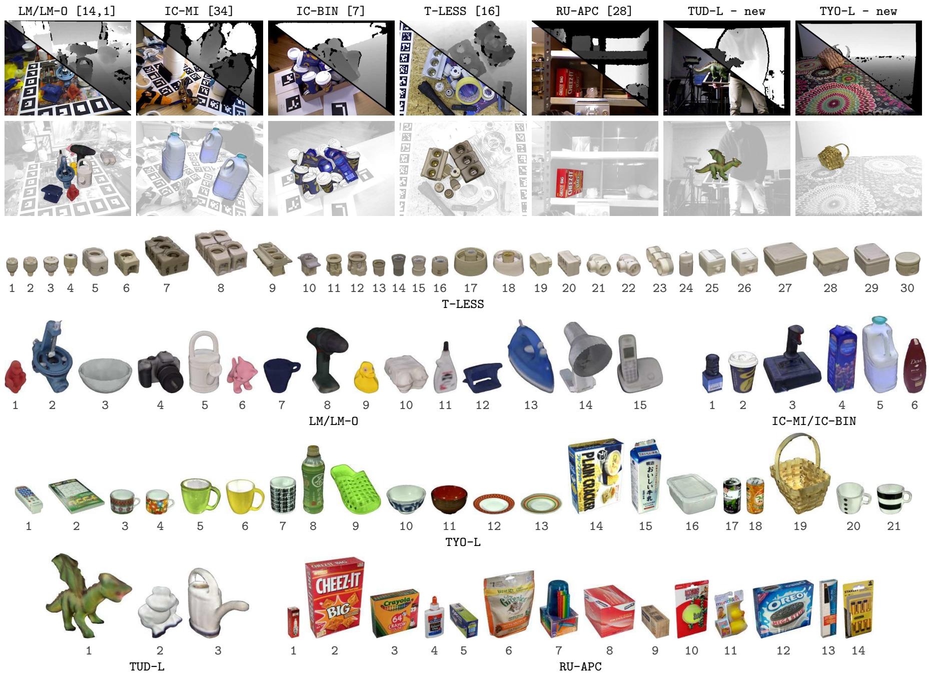

Fig. 1. A collection of benchmark datasets. Top: Example test RGB-D images where the second row shows the images overlaid with 3D object models in the ground-truth 6D poses. Bottom: Texture-mapped 3D object models. At training time, a method is given an object model or a set of training images with ground-truth object poses. At test time, the method is provided with one test image and an identifier of the target object. The task is to estimate the 6D pose of an instance of this object.

This work makes the following contributions:

1. **Eight datasets in a unified format**, including two new datasets focusing on varying lighting conditions, are made available (Fig. 1). The datasets contain: i) texture-mapped 3D models of 89 objects with a wide range of sizes, shapes and reflectance properties, ii) 277K training RGB-D images showing isolated objects from different viewpoints, and iii) 62K test RGB-D images of scenes with graded complexity. High-quality ground-truth 6D poses of the modeled objects are provided for all images.
2. **An evaluation methodology** based on [17] that includes the formulation of an industry-relevant task, and a pose-error function which deals well with pose ambiguity of symmetric or partially occluded objects, in contrast to the commonly used function by Hinterstoisser et al. [14].
3. **A comprehensive evaluation** of 15 methods on the benchmark datasets using the proposed evaluation methodology. We provide an analysis of the results, report the state of the art, and identify open problems.
4. **An online evaluation system** at `bop.felk.cvut.cz` that allows for continuous submission of new results and provides up-to-date leaderboards.

1.1 Related Work

The progress of research in computer vision has been strongly influenced by challenges and benchmarks, which enable to evaluate and compare methods and better understand their limitations. The Middlebury benchmark [31,32] for depth from stereo and optical flow estimation was one of the first that gained large attention. The PASCAL VOC challenge [10], based on a photo collection from the internet, was the first to standardize the evaluation of object detection and image classification. It was followed by the ImageNet challenge [29], which has been running for eight years, starting in 2010, and has pushed image classification methods to new levels of accuracy. The key was a large-scale dataset that enabled training of deep neural networks, which then quickly became a game-changer for many other tasks [23]. With increasing maturity of computer vision methods, recent benchmarks moved to real-world scenarios. A great example is the KITTI benchmark [11] focusing on problems related to autonomous driving. It showed that methods ranking high on established benchmarks, such as the Middlebury, perform below average when moved outside the laboratory conditions.

Unlike the PASCAL VOC and ImageNet challenges, the task considered in this work requires a specific set of calibrated modalities that cannot be easily acquired from the internet. In contrast to KITTY, it was not necessary to record large amounts of new data. By combining existing datasets, we have covered many practical scenarios. Additionally, we created two datasets with varying lighting conditions, which is an aspect not covered by the existing datasets.

2 Evaluation Methodology

The proposed evaluation methodology formulates the 6D object pose estimation task and defines a pose-error function which is compared with the commonly used function by Hinterstoisser et al. [13].

2.1 Formulation of the Task

Methods for 6D object pose estimation report their predictions on the basis of two sources of information. Firstly, at training time, a method is given a training set $T = \{T_o\}_{o=1}^{n}$, where o is an object identifier. Training data T_o may have different forms, e.g. a 3D mesh model of the object or a set of RGB-D images showing object instances in known 6D poses. Secondly, at test time, the method is provided with a test target defined by a pair (I, o), where I is an image showing at least one instance of object o. The goal is to estimate the 6D pose of one of the instances of object o visible in image I.

If multiple instances of the same object model are present, then the pose of an arbitrary instance may be reported. If multiple object models are shown in a test image, and annotated with their ground truth poses, then each object model may define a different test target. For example, if a test image shows three object models, each in two instances, then we define three test targets. For each test target, the pose of one of the two object instances has to be estimated.

This task reflects the industry-relevant bin-picking scenario where a robot needs to grasp a single arbitrary instance of the required object, e.g. a component such as a bolt or nut, and perform some operation with it. It is the simplest variant of the 6D localization task [17] and a common denominator of its other variants, which deal with a single instance of multiple objects, multiple instances of a single object, or multiple instances of multiple objects. It is also the core of the 6D detection task, where no prior information about the object presence in the test image is provided [17].

2.2 Measuring Error

A 3D object model is defined as a set of vertices in $\mathbb{R}^3$ and a set of polygons that describe the object surface. The object pose is represented by a 4×4 matrix $\mathbf{P} = [\mathbf{R}, \mathbf{t}; \mathbf{0}, 1]$, where $\mathbf{R}$ is a 3×3 rotation matrix and $\mathbf{t}$ is a 3×1 translation vector. The matrix $\mathbf{P}$ transforms a 3D homogeneous point $\mathbf{x}_m$ in the model coordinate system to a 3D point $\mathbf{x}_c$ in the camera coordinate system: $\mathbf{x}_c = \mathbf{P}\mathbf{x}_m$.

Visible Surface Discrepancy. To calculate the error of an estimated pose $\hat{\mathbf{P}}$ w.r.t. the ground-truth pose $\bar{\mathbf{P}}$ in a test image I, an object model $\mathcal{M}$ is first rendered in the two poses. The result of the rendering is two distance maps[1] $\hat{S}$ and $\bar{S}$. As in [17], the distance maps are compared with the distance map S_I of the test image I to obtain the visibility masks $\hat{V}$ and $\bar{V}$, i.e. the sets of pixels where the model $\mathcal{M}$ is visible in the image I (Fig. 2). Given a misalignment tolerance τ, the error is calculated as:

$$e_{\mathrm{VSD}}(\hat{S}, \bar{S}, S_I, \hat{V}, \bar{V}, \tau) = \underset{p \in \hat{V} \cup \bar{V}}{\mathrm{avg}} \begin{cases} 0 & \text{if } p \in \hat{V} \cap \bar{V} \, \wedge \, |\hat{S}(p) - \bar{S}(p)| < \tau \\ 1 & \text{otherwise.} \end{cases} \tag{1}$$

Properties of $e_{\mathbf{VSD}}$. The object pose can be ambiguous, i.e. there can be multiple poses that are indistinguishable. This is caused by the existence of multiple fits of the visible part of the object surface to the entire object surface. The visible part is determined by self-occlusion and occlusion by other objects and the multiple surface fits are induced by global or partial object symmetries.

Pose error e_{VSD} is calculated only over the visible part of the model surface and thus the indistinguishable poses are treated as equivalent. This is a desirable property which is not provided by pose-error functions commonly used in the literature [17], including e_{ADD} and e_{ADI} discussed below. As the commonly used pose-error functions, e_{VSD} does not consider color information.

Definition (1) is different from the original definition in [17] where the pixel-wise cost linearly increases to 1 as $|\hat{S}(p) - \bar{S}(p)|$ increases to τ. The new definition is easier to interpret and does not penalize small distance differences that may be caused by imprecisions of the depth sensor or of the ground-truth pose.

[1] A distance map stores at a pixel p the distance from the camera center to a 3D point $\mathbf{x}_p$ that projects to p. It can be readily computed from the depth map which stores at p the Z coordinate of $\mathbf{x}_p$ and which can be obtained by a Kinect-like sensor.

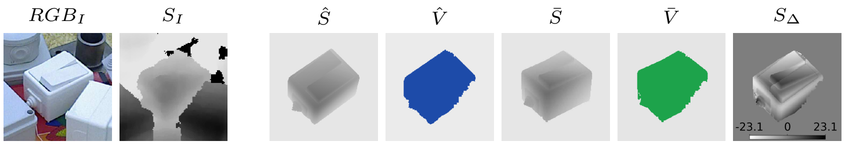

Fig. 2. Quantities used in the calculation of e_{VSD}. Left: Color channels RGB_I (only for illustration) and distance map S_I of a test image I. Right: Distance maps $\hat{S}$ and $\bar{S}$ are obtained by rendering the object model $\mathcal{M}$ at the estimated pose $\hat{\mathbf{P}}$ and the ground-truth pose $\bar{\mathbf{P}}$ respectively. $\hat{V}$ and $\bar{V}$ are masks of the model surface that is visible in I, obtained by comparing $\hat{S}$ and $\bar{S}$ with S_I. Distance differences $S_\Delta(p) = \hat{S}(p) - \bar{S}(p)$, $\forall p \in \hat{V} \cap \bar{V}$, are used for the pixel-wise evaluation of the surface alignment.

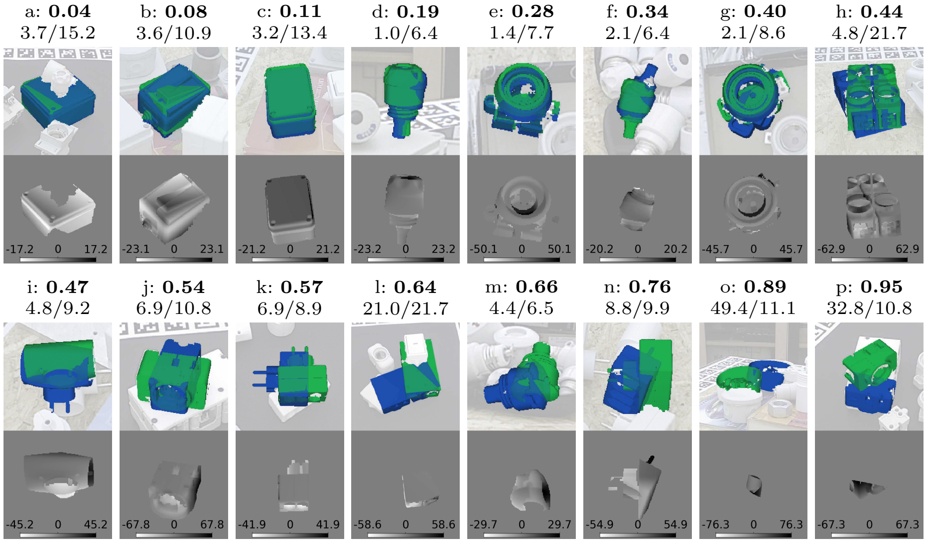

Fig. 3. Comparison of e_{VSD} (bold, $\tau = 20\,\mathrm{mm}$) with $e_{\mathrm{ADI}}/\theta_{\mathrm{AD}}$ (mm) on example pose estimates sorted by increasing e_{VSD}. Top: Cropped and brightened test images overlaid with renderings of the model at i) the estimated pose $\hat{\mathbf{P}}$ in blue, and ii) the ground-truth pose $\bar{\mathbf{P}}$ in green. Only the part of the model surface that falls into the respective visibility mask is shown. Bottom: Difference maps S_Δ. Case (b) is analyzed in Fig. 2.

Criterion of Correctness. An estimated pose $\hat{\mathbf{P}}$ is considered correct w.r.t. the ground-truth pose $\bar{\mathbf{P}}$ if the error $e_{\mathrm{VSD}} < \theta$. If multiple instances of the target object are visible in the test image, the estimated pose is compared to the ground-truth instance that minimizes the error. The choice of the misalignment tolerance τ and the correctness threshold θ depends on the target application. For robotic manipulation, where a robotic arm operates in 3D space, both τ and θ need to be low, e.g. $\tau = 20\,\mathrm{mm}$, $\theta = 0.3$, which is the default setting in the evaluation presented in Sec. 5. The requirement is different for augmented reality applications. Here the surface alignment in the Z dimension, i.e. the optical axis of the camera, is less important than the alignment in the X and Y dimension. The tolerance τ can be therefore relaxed, but θ needs to stay low.

Comparison to Hinterstoisser et al. In [14], the error is calculated as the average distance from vertices of the model $\mathcal{M}$ in the ground-truth pose $\bar{\mathbf{P}}$ to vertices of $\mathcal{M}$ in the estimated pose $\hat{\mathbf{P}}$. The distance is measured to the position of the same vertex if the object has no indistinguishable views (e_{ADD}), otherwise to the position of the closest vertex (e_{ADI}). The estimated pose $\hat{\mathbf{P}}$ is considered correct if $e \leq \theta_{\mathrm{AD}} = 0.1d$, where e is e_{ADD} or e_{ADI}, and d is the object diameter, i.e. the largest distance between any pair of model vertices.

Error e_{ADI} can be un-intuitively low because of many-to-one vertex matching established by the search for the closest vertex. This is shown in Fig. 3, which compares e_{VSD} and e_{ADI} on example pose estimates of objects that have indistinguishable views. Overall, (f)-(n) yield low e_{ADI} scores and satisfy the correctness criterion of Hinterstoisser et al. These estimates are not considered correct by our criterion. Estimates (a)-(e) are considered correct and (o)-(p) are considered wrong by both criteria.

3 Datasets

We collected six publicly available datasets, some of which we reduced to remove redundancies[2] and re-annotated to ensure a high quality of the ground truth. Additionally, we created two new datasets focusing on varying lighting conditions, since this variation is not present in the existing datasets. An overview of the datasets is in Fig. 1 and a detailed description follows.

3.1 Training and Test Data

The datasets consist of texture-mapped 3D object models and training and test RGB-D images annotated with ground-truth 6D object poses. The 3D object models were created using KinectFusion-like systems for 3D surface reconstruction [26,33]. All images are of approximately VGA resolution.

For training, a method may use the 3D object models and/or the training images. While 3D models are often available or can be generated at a low cost, capturing and annotating real training images requires a significant effort. The benchmark is therefore focused primarily on the more practical scenario where only the object models, which can be used to render synthetic training images, are available at training time. All datasets contain already synthesized training images. Methods are allowed to synthesize additional training images, but this option was not utilized for the evaluation in this paper. Only T-LESS and TUD-L include real training images of isolated, i.e. non-occluded, objects.

To generate the synthetic training images, objects from the same dataset were rendered from the same range of azimuth/elevation covering the distribution of object poses in the test scenes. The viewpoints were sampled from a sphere, as in [14], with the sphere radius set to the distance of the closest object instance in the test scenes. The objects were rendered with fixed lighting conditions and a black background.

[2] Identifiers of the selected images are available on the project website.

Dataset	Objects	Training images/obj.		Test images		Test targets	
		Real	Synt.	Used	All	Used	All
LM [14]	15	–	1313	3000	18273	3000	18273
LM-O [1]	8	–	1313	200	1214	1445	8916
IC-MI [34]	6	–	1313	300	2067	300	2067
IC-BIN [7]	2	–	2377	150	177	200	238
T-LESS [16]	30	1296	2562	2000	10080	9819	49805
RU-APC [28]	14	–	2562	1380	5964	1380	5911
TUD-L - new	3	>11000	1827	600	23914	600	23914
TYO-L - new	21	–	2562	–	1680	–	1669
Total	89			7450	62155	16951	110793

Table 1. Parameters of the datasets. Note that if a test image shows multiple object models, each model defines a different test target – see Sec. 2.1.

The test images are real images from a structured-light sensor – Microsoft Kinect v1 or Primesense Carmine 1.09. The test images originate from indoor scenes with varying complexity, ranging from simple scenes with a single isolated object instance to very challenging scenes with multiple instances of several objects and a high amount of clutter and occlusion. Poses of the modeled objects were annotated manually. While LM, IC-MI and RU-APC provide annotation for instances of only one object per image, the other datasets provide ground-truth for all modeled objects. Details of the datasets are in Tab. 1.

3.2 The Dataset Collection

LM/LM-O [14,1]. LM (a.k.a. Linemod) has been the most commonly used dataset for 6D object pose estimation. It contains 15 texture-less household objects with discriminative color, shape and size. Each object is associated with a test image set showing one annotated object instance with significant clutter but only mild occlusion. LM-O (a.k.a. Linemod-Occluded) provides ground-truth annotation for all other instances of the modeled objects in one of the test sets. This introduces challenging test cases with various levels of occlusion.

IC-MI/IC-BIN [34,7]. IC-MI (a.k.a. Tejani et al.) contains models of two texture-less and four textured household objects. The test images show multiple object instances with clutter and slight occlusion. IC-BIN (a.k.a. Doumanoglou et al., scenario 2) includes test images of two objects from IC-MI, which appear in multiple locations with heavy occlusion in a bin-picking scenario. We have removed test images with low-quality ground-truth annotations from both datasets, and refined the annotations for the remaining images in IC-BIN.

T-LESS [16]. It features 30 industry-relevant objects with no significant texture or discriminative color. The objects exhibit symmetries and mutual similarities in shape and/or size, and a few objects are a composition of other objects. T-LESS includes images from three different sensors and two types of 3D object models. For our evaluation, we only used RGB-D images from the Primesense sensor and the automatically reconstructed 3D object models.

RU-APC [28]. This dataset (a.k.a. Rutgers APC) includes 14 textured products from the Amazon Picking Challenge 2015 [6], each associated with test images of a cluttered warehouse shelf. The camera was equipped with LED strips to ensure constant lighting. From the original dataset, we omitted ten objects which are non-rigid or poorly captured by the depth sensor, and included only one from the four images captured from the same viewpoint.

TUD-L/TYO-L. Two new datasets with household objects captured under different settings of ambient and directional light. TUD-L (TU Dresden Light) contains training and test image sequences that show three moving objects under eight lighting conditions. The object poses were annotated by manually aligning the 3D object model with the first frame of the sequence and propagating the initial pose through the sequence using ICP. TYO-L (Toyota Light) contains 21 objects, each captured in multiple poses on a table-top setup, with four different table cloths and five different lighting conditions. To obtain the ground truth poses, manually chosen correspondences were utilized to estimate rough poses which were then refined by ICP. The images in both datasets are labeled by categorized lighting conditions.

4 Evaluated Methods

The evaluated methods cover the major research directions of the 6D object pose estimation field. This section provides a review of the methods, together with a description of the setting of their key parameters. If not stated otherwise, the image-based methods used the synthetic training images.

4.1 Learning-Based Methods

Brachmann-14 [1]. For each pixel of an input image, a regression forest predicts the object identity and the location in the coordinate frame of the object model, a so called “object coordinate”. Simple RGB and depth difference features are used for the prediction. Each object coordinate prediction defines a 3D-3D correspondence between the image and the 3D object model. A RANSAC-based optimization schema samples sets of three correspondences to create a pool of pose hypotheses. The final hypothesis is chosen, and iteratively refined, to maximize the alignment of predicted correspondences, as well as the alignment of observed depth with the object model. The main parameters of the method were set as follows: maximum feature offset: 20 px, features per tree node: 1000, training patches per object: 1.5M, number of trees: 3, size of the hypothesis pool: 210, refined hypotheses: 25. Real training images were used for TUD-L and T-LESS.

Brachmann-16 [2]. The method of [1] is extended in several ways. Firstly, the random forest is improved using an auto-context algorithm to support pose estimation from RGB-only images. Secondly, the RANSAC-based optimization hypothesizes not only with regard to the object pose but also with regard to the object identity in cases where it is unknown which objects are visible in the input image. Both improvements were disabled for the evaluation since we deal with RGB-D input, and it is known which objects are visible in the image. Thirdly, the random forest predicts for each pixel a full, three-dimensional distribution over object coordinates capturing uncertainty information. The distributions are estimated using mean-shift in each forest leaf, and can therefore be heavily multimodal. The final hypothesis is chosen, and iteratively refined, to maximize the likelihood under the predicted distributions. The 3D object model is not used for fitting the pose. The parameters were set as: maximum feature offset: 10 px, features per tree node: 100, number of trees: 3, number of sampled hypotheses: 256, pixels drawn in each RANSAC iteration: 10K, inlier threshold: 1 cm.

Tejani-14 [34]. Linemod [14] is adapted into a scale-invariant patch descriptor and integrated into a regression forest with a new template-based split function. This split function is more discriminative than simple pixel tests and accelerated via binary bit-operations. The method is trained on positive samples only, i.e. rendered images of the 3D object model. During the inference, the class distributions at the leaf nodes are iteratively updated, providing occlusion-aware segmentation masks. The object pose is estimated by accumulating pose regression votes from the estimated foreground patches. The baseline evaluated in this paper implements [34] but omits the iterative segmentation/refinement step and does not perform ICP. The features and forest parameters were set as in [34]: number of trees: 10, maximum depth of each tree: 25, number of features in both the color gradient and the surface normal channel: 20, patch size: 1/2 the image, rendered images used to train each forest: 360.

Kehl-16 [22]. Scale-invariant RGB-D patches are extracted from a regular grid attached to the input image, and described by features calculated using a convolutional auto-encoder. At training time, a codebook is constructed from descriptors of patches from the training images, with each codebook entry holding information about the 6D pose. For each patch descriptor from the test image, k-nearest neighbors from the codebook are found, and a 6D vote is cast using neighbors whose distance is below a threshold t. After the voting stage, the 6D hypothesis space is filtered to remove spurious votes. Modes are identified by mean-shift and refined by ICP. The final hypothesis is verified in color, depth and surface normals to suppress false positives. The main parameters of the method with the used values: patch size: 32×32 px, patch sampling step: 6 px, k-nearest neighbors: 3, threshold t: 2, number of extracted modes from the pose space: 8. Real training images were used for T-LESS.

4.2 Template Matching Methods

Hodaň-15 [18]. A template matching method that applies an efficient cascade-style evaluation to each sliding window location. A simple objectness filter is applied first, rapidly rejecting most locations. For each remaining location, a set of candidate templates is identified by a voting procedure based on hashing, which makes the computational complexity largely unaffected by the total number of stored templates. The candidate templates are then verified as in Linemod [14] by matching feature points in different modalities (surface normals, image gradients, depth, color). Finally, object poses associated with the detected templates are refined by particle swarm optimization (PSO). The templates were generated by applying the full circle of in-plane rotations with 10° step to a portion of the synthetic training images, resulting in 11–23K templates per object. Other parameters were set as described in [18]. We present also results without the last refinement step (Hodaň-15-nr).

4.3 Methods Based on Point-Pair Features

Drost-10 [9]. A method based on matching oriented point pairs between the point cloud of the test scene and the object model, and grouping the matches using a local voting scheme. At training time, point pairs from the model are sampled and stored in a hash table. At test time, reference points are fixed in the scene, and a low-dimensional parameter space for the voting scheme is created by restricting to those poses that align the reference point with the model. Point pairs between the reference point and other scene points are created, similar model point pairs searched for using the hash table, and a vote is cast for each matching point pair. Peaks in the accumulator space are extracted and used as pose candidates, which are refined by coarse-to-fine ICP and re-scored by the relative amount of visible model surface. Note that color information is not used. It was evaluated using function `find_surface_model` from HALCON 13.0.2 [12]. The sampling distances for model and scene were set to 3% of the object diameter, 10% of points were used as the reference points, and the normals were computed using the `mls` method. Points further than 2 m were discarded.

Drost-10-edge. An extension of [9] which additionally detects 3D edges from the scene and favors poses in which the model contours are aligned with the edges. A multi-modal refinement minimizes the surface distances and the distances of reprojected model contours to the detected edges. The evaluation was performed using the same software and parameters as Drost-10, but with activated parameter `train_3d_edges` during the model creation.

Vidal-18 [35]. The point cloud is first sub-sampled by clustering points based on the surface normal orientation. Inspired by improvements of [15], the matching strategy of [9] was improved by mitigating the effect of the feature discretization step. Additionally, an improved non-maximum suppression of the pose candidates from different reference points removes spurious matches. The most voted 500 pose candidates are sorted by a surface fitting score and the 200 best candidates are refined by projective ICP. For the final 10 candidates, the consistency of the object surface and silhouette with the scene is evaluated. The sampling distance for model, scene and features was set to 5% of the object diameter, and 20% of the scene points were used as the reference points.

4.4 Methods Based on 3D Local Features

Buch-16 [3]. A RANSAC-based method that iteratively samples three feature correspondences between the object model and the scene. The correspondences are obtained by matching 3D local shape descriptors and are used to generate a 6D pose candidate, whose quality is measured by the consensus set size. The final pose is refined by ICP. The method achieved the state-of-the-art results on earlier object recognition datasets captured by LIDAR, but suffers from a cubic complexity in the number of correspondences. The number of RANSAC iterations was set to 10000, allowing only for a limited search in cluttered scenes. The method was evaluated with several descriptors: 153d SI [19], 352d SHOT [30], 30d ECSAD [20], and 1536d PPFH [5]. None of the descriptors utilize color.

Buch-17 [4]. This method is based on the observation that a correspondence between two oriented points on the object surface is constrained to cast votes in a 1-DoF rotational subgroup of the full group of poses, SE(3). The time complexity of the method is thus linear in the number of correspondences. Kernel density estimation is used to efficiently combine the votes and generate a 6D pose estimate. As Buch-16, the method relies on 3D local shape descriptors and refines the final pose estimate by ICP. The parameters were set as in the paper: 60 angle tessellations were used for casting rotational votes, and the translation/rotation bandwidths were set to 10 mm/22.5°.

5 Evaluation

The methods reviewed in Sec. 4 were evaluated by their original authors on the datasets described in Sec. 3, using the evaluation methodology from Sec. 2.

5.1 Experimental Setup

Fixed Parameters. The parameters of each method were fixed for all objects and datasets. The distribution of object poses in the test scenes was the only dataset-specific information used by the methods. The distribution determined the range of viewpoints from which the object models were rendered to obtain synthetic training images.

Pose Error. The error of a 6D object pose estimate is measured with the pose-error function e_{VSD} defined in Sec. 2.2. The visibility masks were calculated as in [17], with the occlusion tolerance δ set to 15 mm. Only the ground truth poses in which the object is visible from at least 10% were considered in the evaluation.

Performance Score. The performance is measured by the recall score, i.e. the fraction of test targets for which a correct object pose was estimated. Recall scores per dataset and per object are reported. The overall performance is given by the average of per-dataset recall scores. We thus treat each dataset as a separate challenge and avoid the overall score being dominated by larger datasets.

Subsets Used for the Evaluation. We reduced the number of test images to remove redundancies and to encourage participation of new, in particular slow, methods. From the total of 62K test images, we sub-sampled 7K, reducing the number of test targets from 110K to 17K (Tab. 1). Full datasets with identifiers of the selected test images are on the project website. TYO-L was not used for the evaluation presented in this paper, but it is a part of the online evaluation.

5.2 Results

Accuracy. Tab. 2 and 3 show the recall scores of the evaluated methods per dataset and per object respectively, for the misalignment tolerance $\tau = 20\,\text{mm}$ and the correctness threshold $\theta = 0.3$. Ranking of the methods according to the recall score is mostly stable across the datasets. Methods based on point-pair features perform best. Vidal-18 is the top-performing method with the average recall of 74.6%, followed by Drost-10-edge, Drost-10, and the template matching method Hodaň-15, all with the average recall above 67%. Brachmann-16 is the best learning-based method, with 55.4%, and Buch-17-ppfh is the best method based on 3D local features, with 54.0%. Scores of Buch-16-si and Buch-16-shot are inferior to the other variants of this method and not presented.

Fig. 4 shows the average of the per-dataset recall scores for different values of τ and θ. If the misalignment tolerance τ is increased from 20 mm to 80 mm, the scores increase only slightly for most methods. Similarly, the scores increase only slowly for $\theta > 0.3$. This suggests that poses estimated by most methods are either of a high quality or totally off, i.e. it is a hit or miss.

Speed. The average running times per test target are reported in Tab. 2. However, the methods were evaluated on different computers[3] and thus the presented running times are not directly comparable. Moreover, the methods were optimized primarily for the recall score, not for speed. For example, we evaluated Drost-10 with several parameter settings and observed that the running time can be lowered by a factor of ~5 to 0.5 s with only a relatively small drop of the average recall score from 68.1% to 65.8%. However, in Tab. 2 we present the result with the highest score. Brachmann-14 could be sped up by sub-sampling the 3D object models and Hodaň-15 by using less object templates. A study of such speed/accuracy trade-offs is left for future work.

Open Problems. Occlusion is a big challenge for current methods, as shown by scores dropping swiftly already at low levels of occlusion (Fig. 4, right). The big gap between LM and LM-O scores provide further evidence. All methods perform on LM by at least 30% better than on LM-O, which includes the same but partially occluded objects. Inspection of estimated poses on T-LESS test images confirms the weak performance for occluded objects. Scores on TUD-L show that varying lighting conditions present a serious challenge for methods that rely on

[3] Specifications of computers used for the evaluation are on the project website.

#	Method	LM	LM-O	IC-MI	IC-BIN	T-LESS	RU-APC	TUD-L	Average	Time (s)
1.	Vidal-18	87.83	59.31	95.33	96.50	66.51	36.52	80.17	74.60	4.7
2.	Drost-10-edge	79.13	54.95	94.00	92.00	67.50	27.17	87.33	71.73	21.5
3.	Drost-10	82.00	55.36	94.33	87.00	56.81	22.25	78.67	68.06	2.3
4.	Hodan-15	87.10	51.42	95.33	90.50	63.18	37.61	45.50	67.23	13.5
5.	Brachmann-16	75.33	52.04	73.33	56.50	17.84	24.35	88.67	55.44	4.4
6.	Hodan-15-nopso	69.83	34.39	84.67	76.00	62.70	32.39	27.83	55.40	12.3
7.	Buch-17-ppfh	56.60	36.96	95.00	75.00	25.10	20.80	68.67	54.02	14.2
8.	Kehl-16	58.20	33.91	65.00	44.00	24.60	25.58	7.50	36.97	1.8
9.	Buch-17-si	33.33	20.35	67.33	59.00	13.34	23.12	41.17	36.81	15.9
10.	Brachmann-14	67.60	41.52	78.67	24.00	0.25	30.22	0.00	34.61	1.4
11.	Buch-17-ecsad	13.27	9.62	40.67	59.00	7.16	6.59	24.00	22.90	5.9
12.	Buch-17-shot	5.97	1.45	43.00	38.50	3.83	0.07	16.67	15.64	6.7
13.	Tejani-14	12.10	4.50	36.33	10.00	0.13	1.52	0.00	9.23	1.4
14.	Buch-16-ppfh	8.13	2.28	20.00	2.50	7.81	8.99	0.67	7.20	47.1
15.	Buch-16-ecsad	3.70	0.97	3.67	4.00	1.24	2.90	0.17	2.38	39.1

Table 2. Recall scores (%) for $\tau = 20\,\text{mm}$ and $\theta = 0.3$. The recall score is the percentage of test targets for which a correct object pose was estimated. The methods are sorted by their average recall score calculated as the average of the per-dataset recall scores. The right-most column shows the average running time per test target.

#	Method	LM															LM-O								TUD-L		
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	1	5	6	8	9	10	11	12	1	2	3
1.	Vidal-18	89	96	91	94	92	96	89	89	87	97	59	69	93	92	90	66	81	46	65	73	43	26	64	79	88	74
2.	Drost-10-edge	77	97	94	40	98	94	83	96	45	94	68	66	72	88	79	47	82	46	75	42	44	36	57	85	88	90
3.	Drost-10	86	83	89	84	93	87	86	92	66	96	53	67	79	91	80	62	75	39	70	57	46	26	57	73	90	74
4.	Hodan-15	91	97	79	97	91	97	73	69	90	97	81	79	99	74	95	54	66	40	26	73	37	44	68	27	63	48
5.	Brachmann-16	92	93	76	84	86	90	44	72	85	79	46	67	94	60	66	64	65	44	68	71	3	32	61	81	95	91
6.	Hodan-15-nr	91	57	40	89	66	87	59	49	92	90	65	63	71	54	79	47	35	24	12	63	9	32	53	12	52	20
7.	Buch-17-ppfh	77	65	0	94	84	60	24	59	75	67	24	39	75	47	62	59	63	18	35	60	17	5	30	55	89	63
8.	Kehl-16	60	52	81	25	79	68	17	68	42	91	45	42	78	83	46	39	47	24	30	48	14	13	49	0	23	0
9.	Buch-17-si	40	43	1	63	81	47	12	8	36	43	18	3	46	19	43	54	63	11	2	16	9	1	3	2	74	48
10.	Brachmann-14	74	70	77	75	88	66	11	81	69	66	50	75	92	75	49	50	48	27	44	60	6	30	62	0	0	0
11.	Buch-17-ecsad	31	2	2	19	66	3	3	0	9	49	1	0	3	7	6	29	29	0	0	7	8	1	0	1	62	10
12.	Buch-17-shot	3	4	11	9	9	4	1	3	2	10	1	0	10	12	14	2	7	0	0	1	1	1	0	1	33	17
13.	Tejani-14	36	0	36	0	1	0	1	11	1	70	27	0	0	0	0	26	2	0	1	0	0	10	0	0	0	0
14.	Buch-16-ppfh	11	0	1	22	3	7	2	7	18	12	4	3	9	12	14	4	0	0	2	11	1	1	1	2	0	0
15.	Buch-16-ecsad	2	0	0	9	5	0	0	4	5	8	0	0	17	3	5	1	3	0	2	2	0	0	0	0	1	0

		IC-MI						-BIN		T-LESS																	
		1	2	3	4	5	6	2	4	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1.	Vidal-18	80	100	100	98	100	94	100	93	43	46	68	65	69	71	76	76	92	69	68	84	55	47	54	85	82	79
2.	Drost-10-edge	78	100	100	100	90	96	100	84	53	44	61	67	71	73	75	89	92	72	64	81	53	46	55	85	88	78
3.	Drost-10	76	100	98	100	96	96	100	74	34	46	63	63	68	64	54	48	59	54	51	69	43	45	53	80	79	68
4.	Hodan-15	100	100	100	74	98	100	100	81	66	67	72	72	61	60	52	61	86	72	56	55	54	21	59	81	81	79
5.	Brachmann-16	42	98	70	88	64	78	84	29	8	10	21	4	46	19	52	22	12	7	3	3	0	0	0	5	3	54
6.	Hodan-15-nr	100	100	92	62	60	94	93	59	64	67	71	73	62	57	49	56	85	70	57	55	60	23	60	82	81	77
7.	Buch-17-ppfh	88	100	94	100	100	88	100	50	1	7	0	5	25	16	4	35	37	48	4	10	4	0	0	12	34	49
8.	Kehl-16	22	100	70	72	96	30	71	17	7	10	18	24	23	10	0	2	11	17	5	1	0	9	12	56	52	22
9.	Buch-17-si	62	100	94	62	52	34	97	21	0	1	17	17	9	3	1	4	0	8	2	0	0	0	0	20	26	12
10.	Brachmann-14	96	100	66	72	46	92	28	20	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	2
11.	Buch-17-ecsad	66	88	0	56	34	0	95	23	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	8
12.	Buch-17-shot	52	88	38	36	40	4	66	11	0	0	1	0	1	5	0	2	1	0	0	1	0	1	1	2	1	3
13.	Tejani-14	42	36	0	40	26	74	4	16	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14.	Buch-16-ppfh	28	34	20	6	24	8	4	1	1	6	3	1	24	4	10	13	10	13	3	8	1	0	0	5	32	13
15.	Buch-16-ecsad	4	4	8	4	2	0	5	3	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2

		T-LESS												RU-APC													
		19	20	21	22	23	24	25	26	27	28	29	30	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1.	Vidal-18	57	43	62	69	85	66	43	58	62	69	69	85	39	38	42	54	53	43	4	82	32	0	48	47	20	8
2.	Drost-10-edge	55	47	55	56	84	59	47	69	61	80	84	89	0	20	35	47	35	39	0	89	28	0	48	21	15	3
3.	Drost-10	53	35	60	61	81	57	28	51	32	60	81	71	0	11	29	45	33	29	26	71	10	0	47	9	0	0
4.	Hodan-15	59	27	57	50	74	59	47	72	45	73	74	85	4	36	59	24	47	46	52	97	28	28	34	52	17	0
5.	Brachmann-16	38	1	39	19	61	1	16	27	17	13	6	5	6	64	25	21	32	41	47	37	1	0	18	40	0	5
6.	Hodan-15-nr	58	27	55	50	73	60	49	72	40	72	76	85	4	39	50	24	41	15	43	91	25	33	31	39	16	1
7.	Buch-17-ppfh	31	25	36	35	71	46	64	51	4	44	49	58	16	5	17	51	27	6	57	24	8	10	55	5	11	0
8.	Kehl-16	35	5	26	27	71	36	28	51	34	54	86	69	19	14	46	38	54	40	4	80	3	5	3	37	7	5
9.	Buch-17-si	11	21	18	11	37	4	52	53	3	35	32	53	24	49	16	39	3	4	32	54	14	9	43	15	17	5
10.	Brachmann-14	0	0	0	0	1	0	1	1	0	0	0	0	6	80	42	19	31	33	52	89	19	1	0	40	7	0
11.	Buch-17-ecsad	16	11	16	8	27	20	51	31	0	32	22	3	1	2	0	1	3	8	23	34	5	8	2	0	3	1
12.	Buch-17-shot	6	6	8	2	28	3	17	13	0	11	7	6	0	0	0	0	0	0	0	1	0	0	0	0	0	0
13.	Tejani-14	0	0	0	0	0	0	0	0	0	0	0	2	1	0	0	3	9	0	0	5	0	0	0	3	0	0
14.	Buch-16-ppfh	3	3	8	8	16	2	24	4	5	11	6	1	0	0	6	19	2	12	34	8	0	0	38	2	5	0
15.	Buch-16-ecsad	2	1	3	0	10	0	12	1	2	4	1	1	0	3	5	0	1	1	11	13	0	0	3	2	0	1

Table 3. Recall scores (%) per object for $\tau = 20\,\text{mm}$ and $\theta = 0.3$.

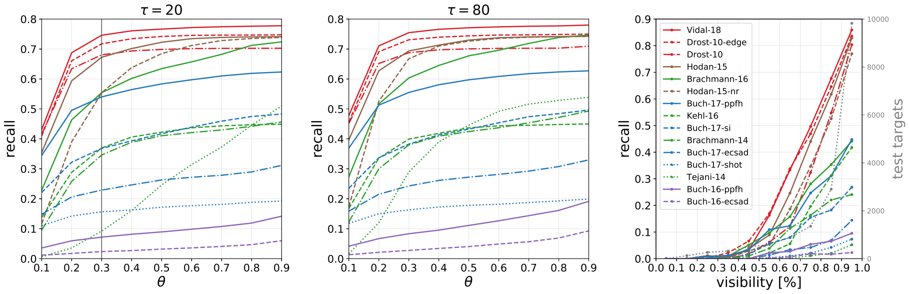

Fig. 4. Left, middle: Average of the per-dataset recall scores for the misalignment tolerance τ fixed to 20 mm and 80 mm, and varying value of the correctness threshold θ. The curves do not change much for $\tau > 80$ mm. Right: The recall scores w.r.t. the visible fraction of the target object. If more instances of the target object were present in the test image, the largest visible fraction was considered.

synthetic training RGB images, which were generated with fixed lighting. Methods relying only on depth information (e.g. Vidal-18, Drost-10) are noticeably more robust under such conditions. Note that Brachmann-16 achieved a high score on TUD-L despite relying on RGB images because it used real training images, which were captured under the same range of lighting conditions as the test images. Methods based on 3D local features and learning-based methods have very low scores on T-LESS, which is likely caused by the object symmetries and similarities. All methods perform poorly on RU-APC, which is likely because of a higher level of noise in the depth images.

6 Conclusion

We have proposed a benchmark for 6D object pose estimation that includes eight datasets in a unified format, an evaluation methodology, a comprehensive evaluation of 15 recent methods, and an online evaluation system open for continuous submission of new results. With this benchmark, we have captured the status quo in the field and will be able to systematically measure its progress in the future. The evaluation showed that methods based on point-pair features perform best, outperforming template matching methods, learning-based methods and methods based on 3D local features. As open problems, our analysis identified occlusion, varying lighting conditions, and object symmetries and similarities.

Acknowledgements

We gratefully acknowledge Manolis Lourakis, Joachim Staib, Christoph Kick, Juil Sock and Pavel Haluza for their help. This work was supported by CTU student grant SGS17/185/OHK3/3T/13, Technology Agency of the Czech Republic research program TE01020415 (V3C – Visual Computing Competence Center), and the project for GAČR, No. 16-072105: Complex network methods applied to ancient Egyptian data in the Old Kingdom (2700–2180 BC).

References

1. Brachmann, E., Krull, A., Michel, F., Gumhold, S., Shotton, J., Rother, C.: Learning 6D object pose estimation using 3D object coordinates. In: ECCV (2014) 1, 2, 7, 8
2. Brachmann, E., Michel, F., Krull, A., Yang, M.Y., Gumhold, S., Rother, C.: Uncertainty-driven 6D pose estimation of objects and scenes from a single RGB image. In: CVPR (2016) 1, 8
3. Buch, A.G., Petersen, H.G., Krüger, N.: Local shape feature fusion for improved matching, pose estimation and 3D object recognition. SpringerPlus (2016) 11
4. Buch, A.G., Kiforenko, L., Kraft, D.: Rotational subgroup voting and pose clustering for robust 3D object recognition. In: ICCV (2017) 11
5. Buch, A.G., Kraft, D.: Local point pair feature histogram for accurate 3D matching. In: BMVC (2018) 11
6. Correll, N., Bekris, K.E., Berenson, D., Brock, O., Causo, A., Hauser, K., Okada, K., Rodriguez, A., Romano, J.M., Wurman, P.R.: Lessons from the Amazon picking challenge. ArXiv e-prints (2016) 8
7. Doumanoglou, A., Kouskouridas, R., Malassiotis, S., Kim, T.K.: Recovering 6D object pose and predicting next-best-view in the crowd. In: CVPR (2016) 2, 7
8. Drost, B., Ulrich, M., Bergmann, P., Härtinger, P., Steger, C.: Introducing MVTec ITODD – a dataset for 3D object recognition in industry. In: ICCVW (2017) 1
9. Drost, B., Ulrich, M., Navab, N., Ilic, S.: Model globally, match locally: Efficient and robust 3D object recognition. In: CVPR (2010) 10
10. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The PASCAL visual object classes (VOC) challenge. IJCV (2010) 3
11. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? The KITTI vision benchmark suite. In: CVPR (2012) 3
12. MVTec HALCON: `https://www.mvtec.com/halcon/` 10
13. Hinterstoisser, S., Cagniart, C., Ilic, S., Sturm, P., Navab, N., Fua, P., Lepetit, V.: Gradient response maps for real-time detection of texture-less objects. TPAMI (2012) 3
14. 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) 1, 2, 6, 7, 9, 10
15. Hinterstoisser, S., Lepetit, V., Rajkumar, N., Konolige, K.: Going further with point pair features. In: ECCV (2016) 10
16. Hodaň, T., Haluza, P., Obdržálek, Š., Matas, J., Lourakis, M., Zabulis, X.: T-LESS: An RGB-D dataset for 6D pose estimation of texture-less objects. In: WACV (2017) 1, 2, 7
17. Hodaň, T., Matas, J., Obdržálek, Š.: On evaluation of 6D object pose estimation. In: ECCVW (2016) 2, 4, 11
18. Hodaň, T., Zabulis, X., Lourakis, M., Obdržálek, Š., Matas, J.: Detection and fine 3D pose estimation of texture-less objects in RGB-D images. In: IROS (2015) 1, 9, 10
19. Johnson, A.E., Hebert, M.: Using spin images for efficient object recognition in cluttered 3D scenes. TPAMI **21**(5) (1999) 11
20. Jørgensen, T.B., Buch, A.G., Kraft, D.: Geometric edge description and classification in point cloud data with application to 3D object recognition. In: VISAPP (2015) 11

21. Kehl, W., Manhardt, F., Tombari, F., Ilic, S., Navab, N.: SSD-6D: Making RGB-based 3D detection and 6D pose estimation great again. In: ICCV (2017) 1
22. Kehl, W., Milletari, F., Tombari, F., Ilic, S., Navab, N.: Deep learning of local RGB-D patches for 3D object detection and 6D pose estimation. In: ECCV (2016) 9
23. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: NIPS (2012) 3
24. Krull, A., Brachmann, E., Michel, F., Ying Yang, M., Gumhold, S., Rother, C.: Learning analysis-by-synthesis for 6D pose estimation in RGB-D images. In: ICCV (2015) 1
25. Michel, F., Alexander Kirillov, Brachmann, E., Krull, A., Gumhold, S., Savchynskyy, B., Rother, C.: Global hypothesis generation for 6D object pose estimation. In: CVPR (2017) 1
26. Newcombe, R.A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A.J., Kohi, P., Shotton, J., Hodges, S., Fitzgibbon, A.: KinectFusion: Real-time dense surface mapping and tracking. In: ISMAR (2011) 6
27. Rad, M., Lepetit, V.: BB8: A scalable, accurate, robust to partial occlusion method for predicting the 3D poses of challenging objects without using depth. In: ICCV (2017) 1
28. Rennie, C., Shome, R., Bekris, K.E., De Souza, A.F.: A dataset for improved RGBD-based object detection and pose estimation for warehouse pick-and-place. Robotics and Automation Letters (2016) 2, 7, 8
29. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: ImageNet large scale visual recognition challenge. IJCV (2015) 3
30. Salti, S., Tombari, F., Di Stefano, L.: SHOT: Unique signatures of histograms for surface and texture description. Computer Vision and Image Understanding (2014) 11
31. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. IJCV (2002) 3
32. Scharstein, D., Pal, C.: Learning conditional random fields for stereo. In: CVPR (2007) 3
33. Steinbrücker, F., Sturm, J., Cremers, D.: Volumetric 3D mapping in real-time on a CPU. In: ICRA (2014) 6
34. Tejani, A., Tang, D., Kouskouridas, R., Kim, T.K.: Latent-class hough forests for 3D object detection and pose estimation. In: ECCV (2014) 1, 2, 7, 9
35. Vidal, J., Lin, C.Y., Martí, R.: 6D pose estimation using an improved method based on point pair features. In: ICCAR (2018) 10
36. Wohlhart, P., Lepetit, V.: Learning descriptors for object recognition and 3D pose estimation. In: CVPR (2015) 1