Benchmarking 6D Object Pose Estimation for Robotics

Antti Hietanen¹,², Jyrki Latokartano², Alessandro Foi¹, Roel Pieters², Ville Kyrki³, Minna Lanz² and Joni-Kristian Kämäräinen¹

Abstract—Benchmarking 6D object pose estimation for robotics is not straightforward as sufficient accuracy depends on many factors, e.g., the selected gripper, dimensions, weight and material of an object, grasping point, and the robot task itself. We formulate the problem as a “successful grasp”, i.e. for a fixed set of factors affecting the task, will the given pose estimate provide sufficiently good grasp to complete the task. The successful grasp is modelled in a probabilistic framework by sampling in the pose error space and executing the task and automatically detecting success or failure. Hours of sampling and thousands of samples are used to construct a non-parametric probability of a successful grasp given the pose residual. The framework is experimentally validated with real objects and assembly tasks and comparison of several state-of-the-art point cloud based 3D pose estimation methods.

I. INTRODUCTION

Many robot tasks, for example, industrial assembly and bin picking, require manipulation of objects. Object manipulation, on the other hand, benefits from vision based grasping where the 3D geometric shape information of known objects are used to generate stable 6D grasp poses with the help of pose estimation methods. A typical performance measure in pose estimation benchmarks [1]–[4] is the alignment error between an estimated and ground truth pose, but this is not sufficient for robot manipulation where the success depends on many other factors: i) the object and its properties (material, weight, dimensions), ii) the selected gripper, iii) a selected grasp point, and iv) a task itself to be completed (bin picking vs. precision wrenching). Several prior works [5]–[8] also report success rates for their specific setups and tasks, but for comparison the whole setup should be re-built.

In this work, we formulate the problem in probabilistic manner; what is the conditional probability of a successful grasp ($X = 1$) given the pose residual ($\epsilon$): $P(X = 1 | \epsilon)$. To compute the probability we need to sample the residual $\epsilon = (\epsilon_x, \epsilon_y, \epsilon_z, \epsilon_{\theta_1}, \epsilon_{\theta_2}, \epsilon_{\theta_3})$ in 6D space using a real setup. Finally, using the probabilistic model we can generate a large number of grasping samples (see Fig. 1) and benchmark 6D pose estimation methods using the realistic and practical performance measures, successful grasp probability and proportion of successful grasps on a pre-defined probability level (e.g., 0.90).

The main contributions of this work are:

• A formulation of the conditional probability of a successful grasp, $P(X = 1 | \epsilon)$, given a 6D pose residual

This work was supported by the projects TRINITY, UNITY and ROSE.

¹Computing Sciences and ²Automation Technology and Mechanical Engineering, Tampere University, Finland, First.Second@tuni.fi
³Aalto University, Finland, First.Second@aalto.fi

Fig. 1. Top: our experimental setup with a Intel D415 RGB-D sensor, UR5 robot manipulator with a gripper and the engine part. Middle: the reference grasp point (blue) is estimated using an artificial marker and the altered grasp poses (red) are generated by adding 6D residuals to the reference. Bottom: XY-plane of the estimated grasp success probability over 6D grasp pose residuals.
point cloud based pose estimation methods using the proposed measure. Essentially, to adopt the proposed benchmark one needs the 3D point clouds models, test images and the grasp probability function which will all be made publicly available to facilitate fair comparisons.

II. RELATED WORK

A. Benchmarking 6D pose estimation

The first complete 6D pose estimation benchmark with a dataset and an evaluation procedure was created by Hinterstoisser et al. [4] and later improved in [9]. Despite of a relatively easy test images the dataset is still frequently used as the main benchmark. Hodan et. al [1] recently proposed a 6D pose estimation benchmark comparing 15 state-of-the-art methods covering all the main research branches with local-feature-based, template matching and learning-based approaches. The benchmark uses 8 different datasets containing different types of images ranging from typical household object [10] to industry-relevant objects with no significant texture [11]. For evaluations the authors propose a variant of their previously introduced Visible Surface Discrepancy (VSD) [3] error metric. In addition the authors introduce an online evaluation system that provides updated leaderboard system. However, it remains unclear how well these benchmarks measure pose estimation for practical applications such as robotic assembly and disassembly.

B. Performance measures

There are several commonly used performance metrics for pose estimation. The most popular is the alignment error proposed by Hinterstoisser et al. [4] which calculates the average distance transformed by the ground truth pose and the estimated pose. Another popular metric is to report separately 3D translation and 3D rotation errors [12].

Recently Hodan et al. [3] proposed the VSD metric where the error is only calculated over the visible part of the surface model. The main contribution of the metric is invariance to pose ambiguity, i.e. due to the object symmetry there can be multiple poses that are indistinguishable. However the method requires additional ground truth visibility mask to be estimated.

All above metrics measure the pose error in the terms of misalignment between the ground truth and estimated object surface points. This requirement is important, for example, in augmented reality applications where the perceived virtual object must align well with the real environment. However, in robotic manipulation such as industrial assembly and disassembly a suitable performance metric must reflect the fact whether the programmed task can be completed or not. Finding such a metric is the main objective of this work.

C. 6D pose estimation methods

Many recent state-of-the-art methods divide 6D pose estimation to two distinct steps [13], [14]: i) RGB-based object detection and ii) point cloud based 6D pose estimation. Since the object detection is out of the scope in our work we only focus on the recent 6D pose estimation methods. In the following we briefly review several baselines and state-of-the-arts which are included to the experimental part. All methods operate directly on point clouds captured by depth sensors.

Nearest Neighbor Similarity Ratio (NNSR) [15] is a popular yet very efficient correspondence grouping method utilizing only feature matches. It flags a correspondence as an outlier if the closest and the second closest match are too similar i.e. descriptor $f$ can not be distinctively described:

$$\quad 1 - \frac{\|f - f_1\|}{\|f - f_2\|} \geq \tau_{\text{NNSR}}, \quad (1)$$

where $\tau_{\text{NNSR}} \in [0, 1]$.

Random Sample Consensus (RANSAC) [16] is another widely used method in 2D and 3D domains. It is an iterative process that uses random sampling technique to generate candidate solutions for a model (transformation) that aligns two surfaces with a minimum point-wise error. Free parameter of the method is $N_{\text{RANSAC}}$ which is the number of iterations the algorithm samples matches from the correspondence set. The samples are used to generate candidate transformations which fitness are evaluated by transforming all query keypoints target surface and calculating Euclidean distance between the corresponding points. All the keypoints having smaller distance than $d_{\text{RANSAC}}$ are counted as inliers for the specific transformation candidate and finally the candidate having the most inliers is selected as the estimated pose transformation.

Search of Inliers (SI) [17] is a recent method that provides state-of-the-art accuracy on several benchmarks. It uses two consecutive processing stages, local voting and global voting. The first voting step performs local voting, where locally selected correspondence pairs are selected from the query and a target, and the score is computed using their pairwise similarity score $s_{ij}(c)$. At the global voting stage, the algorithm samples point correspondences, estimates a transformation and gives a global score to the points correctly aligned outside the estimation point set: $s_G(c)$. The final score $s(c)$ is computed by integrating both local and global scores and finally $s(c)$ are thresholded to inliers and outliers based on Otsu’s bimodal distribution thresholding.

Geometric Consistency Grouping (GC) [18] is popular baseline which is implemented in several point cloud libraries. In contrast to NNSR the GC grouping algorithm operates only on the spatial space and evaluates the consistency of two correspondences $c_i$ and $c_j$ by:

$$\quad d(c_i, c_j) = \left\| p_i - p_j \right\|^2 - \left\| p_i - p'_j \right\|^2 < \tau_{\text{GC}}. \quad (2)$$

The algorithm is initialized with a fixed number of clusters and in principle the GC can return more than one correspondence cluster. In our work we select the cluster having the largest number of correspondence.
Hough Grouping (HG) [19] extends the original Hough Transformation algorithm [20] to 3D correspondence grouping where the key idea is to iteratively cast votes for object location and pose bins in the Hough parameter space. At the end of the process the highest accumulated bins represent the most likely pose candidates and correspondence contributed to the bins are accepted. The method requires an unique reference point in the model, typically the model centroid, and each bin represents a single pose instance candidate. Therefore correct correspondence vote a same bin which gets quickly accumulated. To make correspondence points invariant to rotation an translation between the model and scene, every point is associated with local Reference Frame (RF) [21]. In the voting stage each correspondence between a capture scene and a model cast a single vote to a single or multiple bins in the 3D translation Hough accumulator space and pose is stored in the local reference frame. Finally, correspondence contributing bins having votes more than a set threshold which is adaptively set as it depends on the number of available points and the most important parameter is the Hough accumulator bin size.

III. OBJECT POSE FOR ROBOT MANIPULATION

In this section we propose a novel approach to evaluate and benchmark 6D object pose estimation algorithms for robotics. The approach is based on a probabilistic formulation of a successful grasp (Section III-A) and sampling in the residual space to estimate the probability model (Section III-B).

A. Measuring the probability of a successful grasp

The success of a grasp is a binary random variable \( X \in \{0, 1\} \) where \( X = 1 \) denotes a successful grasp and \( X = 0 \) denotes an unsuccessful grasp attempt. Therefore, \( X \) follows a Bernoulli distribution with complementary probability of success and failure: \( E(X) = P(X = 1) = 1 - P(X = 0) \), where \( E \) denotes the mathematical expectation. The fundamental problem in our case is the fact that the probability of success depends on the pose error \( \epsilon = (\epsilon_x, \epsilon_y, \epsilon_z, \epsilon_\theta_1, \epsilon_\theta_2, \epsilon_\theta_3)^T \) where the vector elements denote the error which respect to each pose parameter (the translation error 3D-vector and the rotation error 3D-vector). Thus, we should focus on the conditional distributions, conditional probabilities, and conditional expectations, as functions of \( \epsilon \). For any given error \( \epsilon \in \mathbb{E} \), the conditional probability of a successful grasp attempt is

\[
p(\epsilon) = E(X|\epsilon) = P(X = 1|\epsilon) = 1 - P(X = 0|\epsilon) .
\]

The maximum likelihood estimate of the Bernoulli parameter \( p \in [0, 1] \) from \( N \) homogeneous samples is the sample average

\[
\hat{p}_{\text{ML}} = \frac{1}{N} \sum_{i=1}^{N} y_i , \tag{4}
\]

where homogeneity means that all samples \( y_i, i = 1, \ldots, N \), are realization of a common Bernoulli random variable with unique underlying parameter \( p \). However, guaranteeing homogeneity would require that the samples \( \{y_i, i = 1, \ldots, N\} \) were either all collected at the same residual \( \epsilon_1 = \cdots = \epsilon_N \), or for different residuals that nonetheless yield same probability \( p(\epsilon_1) = \cdots = p(\epsilon_N) \), i.e. it would require us either to collect multiple samples for each \( \epsilon \in \mathbb{E} \) or to know beforehand \( p \) over \( \mathbb{E} \) (which is what we are trying to estimate). This means that in practice \( p \) must be estimated from non-homogeneous samples, i.e. from \( \{y_i, i = 1, \ldots, N\} \) sampled at residuals \( \{\epsilon_i, i = 1, \ldots, N\} \) which can be different and having different underlying \( \{p(\epsilon_i), i = 1, \ldots, N\} \).

The actual form of \( p \) over \( \mathbb{E} \) is unknown and depends on many factors, e.g., the shape of an object and the properties of a gripper. Therefore it is not meaningful to assume any parametric shape such as the Gaussian or uniform distribution. Instead, we adopt the Nadaraya-Watson non-parametric estimator which gives the probability of a successful grasp as

\[
\hat{p}_h(\epsilon) = \frac{\sum_{i=1}^{N} y_i K_h(\epsilon_i - \epsilon)}{\sum_{i=1}^{N} K_h(\epsilon_i - \epsilon)} , \tag{5}
\]

where \( \epsilon_i \) denotes the pose error at which \( y_i \) has been sampled and \( K_h : \mathbb{E} \rightarrow \mathbb{R}^+ \) is a nonnegative multivariate kernel with vector scale \( h = (h_x, h_y, h_z, h_{\theta_1}, h_{\theta_2}, h_{\theta_3})^T > 0 \).

In this work, \( K_h \) is the multivariate Gaussian kernel

\[
K_h(\epsilon) = G \left( \frac{\epsilon_x}{h_x} \right) G \left( \frac{\epsilon_y}{h_y} \right) G \left( \frac{\epsilon_z}{h_z} \right) \sum_{j \in \mathbb{E}} G \left( \frac{\epsilon_{\theta_1}}{h_{\theta_1}} \right) + 2 \pi \frac{\epsilon_{\theta_2}}{h_{\theta_2}} \sum_{j \in \mathbb{E}} G \left( \frac{\epsilon_{\theta_3}}{h_{\theta_3}} \right) + 2 \pi \frac{\epsilon_{\theta_3}}{h_{\theta_3}} , \tag{6}
\]

where \( G \) is the standard Gaussian bell, \( G(\epsilon) = (2\pi)^{-\frac{3}{2}} e^{-\frac{\epsilon^2}{2}} \).

The process of the estimator (5) is heavily affected by the choice of \( h \), which determines the influence of samples \( y_i \) in computing \( \hat{p}_h(\epsilon) \) based on the difference between the pose errors \( \epsilon \) and \( \epsilon_i \). Indeed, the parameter \( h \) can be interpreted as reciprocal to the bandwidth of the estimator: too large \( h \) results in excessive smoothing whereas too small results in localization spikes.

To find an optimal \( h \), we use the leave-one-out (LOO) cross-validation method. Specifically, we construct the estimator on the basis of \( N - 1 \) training examples leaving out the \( i \)-th sample:

\[
\hat{p}_h^{\text{LOO}}(\epsilon, i) = \frac{\sum_{j \neq i} y_j K_h(\epsilon_j - \epsilon)}{\sum_{j \neq i} K_h(\epsilon_j - \epsilon)} .
\]

The likelihood of \( y_i \) given \( \hat{p}_h^{\text{LOO}}(\epsilon, i) \) is either \( \hat{p}_h^{\text{LOO}}(\epsilon_i, i) \) if \( y_i = 1 \), or \( 1 - \hat{p}_h^{\text{LOO}}(\epsilon_i, i) \) if \( y_i = 0 \). We then select \( h \) that maximizes the total LOO log-likelihood over the whole set \( S_y \):

\[
h = \arg \max_h \sum_{i \in S_y} \log \left( \hat{p}_h^{\text{LOO}}(\epsilon, i) \right) + \sum_{i \in S_y} \log \left( 1 - \hat{p}_h^{\text{LOO}}(\epsilon, i) \right) .
\]

B. Sampling the residual space \( \epsilon \)

Evaluation of the success of the grasp is based on 2D markers attached to the manipulated objects and that can be accurately detected and localized with their 6D pose
by the ArUco library [22]. In the work marker is detected using the RGB channels of the RGB-D sensor. A grasp in manually programmed and the grasp point is defined with respect to the markers by a 6D similarity transformation (3D translation and 3D rotation) that is a 4 × 4 matrix $T^{\text{marker}}_{\text{grasp}}$ for homogeneous coordinates $(x, y, z, 1)^T$. The gripper final pose defines the grasp. However, the object pose estimate is given in the world frame that corresponds to the robot frame in our case. The coordinate transformation from the grasp to the world (robot) frame is defined as the transformation chain:

$$T^{\text{world}}_{\text{grasp}} = T^{\text{effector}}_{\text{marker}} T^{\text{sensor}}_{\text{marker}} T^{\text{world}}_{\text{marker}}$$

(7)

where the transformations are:
- $T^{\text{marker}}_{\text{grasp}}$ – a constant transformation that is measured for a closed gripper with respect to the 2D marker;
- $T^{\text{sensor}}_{\text{marker}}$ – the transformation from the detected marker from the sensor values (RGB-D) to the sensor frame;
- $T^{\text{effector}}_{\text{sensor}}$ – a constant transformation from the sensor frame to the robot end-effector frame (note that the RGB-D sensor is mounted to the end-effector);
- $T^{\text{world}}_{\text{effector}}$ – the robot end-effector position and orientation with respect to the base (world) frame calculated using the joint values and known kinematic equations.

Using the robot forward kinematics $T^{\text{world}}_{\text{effector}}$ and a printed chessboard pattern we compute $T^{\text{sensor}}_{\text{marker}}$ using the standard procedure for hand-camera calibration [23]. For a calibrated camera the ArUco library provides a real-time pose of the marker with respect to the world frame $T^{\text{world}}_{\text{marker}}$. The constant offset from the marker to the actual grasp pose is estimated from

$$T^{\text{marker}}_{\text{grasp}} = \left( T^{\text{world}}_{\text{marker}} \right)^{-1} T^{\text{world}}_{\text{grasp}} \cdot T^{\text{grasp}}_{\text{world}}$$

where $T^{\text{world}}_{\text{grasp}}$ is estimated manually by hand-guiding the end-effector to the predefined grasp pose using the forward kinematics engine.

a) 6D residual: An residual is generated by sampling $\epsilon_x, \epsilon_y, \epsilon_z \in [\min x_{\text{ga}}, \max x_{\text{ga}}]$ and $\epsilon_{\theta_1}, \epsilon_{\theta_2}, \epsilon_{\theta_3} \in [\min \theta_{\text{ga}}, \max \theta_{\text{ga}}]$ and forming a 3D isometry transformation $T^\epsilon$ which is added to the final pose

$$T^{\epsilon}_{\text{world}}_{\text{grasp}} = T^{\text{world}}_{\text{grasp}} T^\epsilon \cdot$$

(8)

Our sampling procedure starts by one by one beam searching the min. and max. values for each in the residual parameter when the grasp fails. The limits are then used to generate uniform samples in the residual space (see Table 1).

b) Grasp success validation: Our tasks of interest contain two important stages, grasp and placement, which both need to succeed. We define success automatically by two different triggers: error in the final pose of the manipulated object and wrench torque at the end-effector during execution. The final pose error measures success in task completion while torque measures collisions during the task.

For the pose we manually define the thresholds for the translation error $\tau_T$ and orientation error $\tau_\theta$. These can be computed using the markers attached to the manipulated objects and using the target pose rotation and translation $P = [R | t]$ and the measured pose $\hat{P} = [\hat{R} | \hat{t}]$ as

$$\| t - \hat{t} \| \leq \tau_t, \quad \arccos \left( \frac{\hat{R} R^{-1} - 1}{2} \right) \leq \tau_\theta$$

The torque is used to detect if the robot collides with its environment while grasping and moving the object. In addition, if the robot places the object to the correct position with too high wrench the whole operation is considered as an unsuccessful attempt. The external wrench is computed based on the error between the joint torques required to stay on the programmed trajectory and the expected joint torques. Using the robot internal sensors we get the torque measurements $F = (f_x, f_y, f_z)$, where $f_x$, $f_y$, and $f_z$ are the forces in the axes of the robot frame coordinates and measured in Newtons. We manually set the limits $\min_f$ and $\max_f$ during each operation stage and trajectory and measurements violating the limits are considered as a failure.

### IV. EXPERIMENTS

In the following, we explain the experimental setup, data and evaluation procedure to benchmark pose estimation methods. We report the results for five point cloud based pose estimation methods.

A. Setup

In Figure 2 is illustrated the setup used in our experiments. The experiments were conducted using the model 5 Universal Robot Arm (UR5) with a Schunk PGN-100 gripper. The gripper operates pneumatically and was configured to have a high gripping force (approximately 600N) to prevent object slippage. In addition, the gripper had custom 3D printed fingers plated with rubber. For perception data, an Intel RealSense D415 RGB-D sensor was secured on a 3D printed flange statically mounted between the gripper and the robot end-effector. All the in-house made 3D prints were made using nylon reinforced with carbon fiber to tolerate external forces during the experiments. All the computation was performed on a single laptop with Ubuntu 18.04.

B. Data

a) Work object 6D residual: Every work object used in the experiments was first tested 100 times on the pipeline described in Table B with zero residual to confirm that the robot can perform the task automatically without problems. On average, a successful work operation took 45-55 seconds to execute and in 24 hours the robot could automatically perform the task approximately 1,100 times depending on the work operation. The setup can automatically recover from most of the failure cases (dropping the object, object collision, etc.), however, if the marker was occluded by the environment or if the manipulated object got jammed against the internal parts of the motor, the pipeline was restarted by the human operator. For both tasks we generated approximately 3,300 valid samples.
the resulting object surfaces we estimate the local surface of the cloud the size of the voxel was the amount of data for processing. Depending on the density resolution using a regular voxel grid based method to limit and scene point clouds were first downsampled to a fixed use point clouds as input (Section II). The input models of the work pieces were semi-automatically created and assemble the parts to the engine body. The point cloud a motor top cap and a motor frame. The task is to grasp and redundant parts of the reconstructed point cloud were removed manually.

The test dataset was generated in a similar manner by moving the arm around the work pieces. For each of the objects we collected 150 test images in three different settings: 1) a single target object present, 2) multiple objects present and 3) a single target object present with partial occlusion. The dataset contains also the ground truth information to align the model to the test images.

C. Data preparation

All pose estimation algorithms used in the experiments use point clouds as input (Section II). The input models and scene point clouds were first downsampled to a fixed resolution using a regular voxel grid based method to limit the amount of data for processing. Depending on the density of the cloud the size of the voxel was 0.5 – 1.0mm. For the resulting object surfaces we estimate the local surface normals by the least squares plane fitting on points in a small neighborhood. The object surfaces are down-sampled to a resolution that results in approximately 1000 – 3000 points depending on the object. On these points, the local descriptors for point matching were computed using the local point neighborhoods. The SHOT [21] feature descriptor was selected since it performed the best in our preliminary experiments. The descriptor support radius was computed as 0.125×object model bounding box diagonal. During the experiments the most similar descriptors in a $L_2$ sense between the model and scene were found using a randomized $kd$-tree similarity search. The found matches formed then the initial set of correspondences.

D. Error measurement

In this work we use 2 different metrics to compare the performance of the different pose estimation methods. The first method is the proposed probabilistic approach described in Section III-A. The 6D pose residual from the estimated pose $\hat{P}$ is computed from

$$\epsilon = \hat{P}^{-1} P,$$

where $P$ is the ground truth pose transform. The actual grasp success rate is then calculated using Eq. 5. In addition to the average of the computed grasp probabilities we report the proportion of the estimates for which the success probability is above a threshold value 0.90.

To put the proposed error function into perspective with the current literature we use the surface MSE [4] where the pose error is calculated as the average distance between transformed model points:

$$\epsilon_{RMS} = \frac{1}{\mathcal{M}} \sum_{x \in \mathcal{M}} (\hat{P}x - Px)^2.$$

Since complete failure of a method for certain test images might influence the result too much, we also report the top–25% MSE error, which is less affected by estimation failures providing large errors.

E. Results

The results for all the methods are shown in Table II. Rather strikingly, the best 3D pose estimate in all cases and using all metrics is provided by the Geometrical Consistency (GC) method which is clearly superior to HG and SI that represent more state-of-the-art. GC has the lowest MSE error for both objects and in the Motor Cap pose estimation the error is over three orders of magnitude lower as compared to other methods. SI performs slightly better than HG and, surprisingly, RANSAC is behind the NNSR which is considered the baseline.

Observing the grasp rates, again the GC method outperforms all the other methods with a clear margin although the successful rate is relatively low. This is due to the fact that for the both test objects the estimated grasp pose has to be very close to the reference (Table I).

![Fig. 2. The experimental robotic setup to sample the residual space. The task is adopted from a local diesel engine manufacturing company.](image)
V. CONCLUSIONS
The main outcome of this work is a novel evaluation metric to benchmark 3D object pose estimation methods, the average grasping success probability. The metric measures the true probability to succeed in the given task using the given setup and has therefore clear practical relevance. Other research groups can benchmark their pose estimation methods without the same physical setup using the provided test images and pre-computed probability model. Instead of the popular MSE metric our metric provides direct interpretation of the performance, i.e. can the task be solved using the estimated poses. We provide all benchmark images and tools through a public Web page to facilitate fair comparison and to promote more practical research on grasping and 3D object pose estimation.

REFERENCES
[1] T. Hodan, F. Michel, E. Brachmann, W. Kehl, A. GlentBuch, D. Kraft, B. Drost, J. Vidal, S. Ihrke, X. Zabulis, et al., “Bop: benchmark for 6d object pose estimation,” in ECCV, pp. 606–619, Springer, 2016.
[2] J. Yang, K. Xian, Y. Xiao, and Z. Cao, “Performance evaluation of 3d correspondence grouping algorithms,” in 3DV, pp. 467–476, IEEE, 2017.
[3] T. Hodan, J. Matas, and Š. Obdržálek, “On evaluation of 6d object pose estimation,” in ECCV, pp. 606–619, Springer, 2016.
[4] S. Hinterstoisser, V. Lepeitl, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab, “Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes,” in ACCV, pp. 548–562, Springer, 2012.
[5] U. Vireck, A. t. Pas, K. Saenko, and R. Platt, “Learning a visuomotor controller for real world robotic grasping using simulated depth images,” arXiv preprint arXiv:1708.04652, 2017.
[6] M. Gualtieri, A. Ten Pas, K. Saenko, and R. Platt, “High precision grasp pose detection in dense clutter,” in IROS, pp. 598–605, IEEE, 2016.
[7] L. Pinto and A. Gupta, “Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours,” in ICRA, pp. 3406–3413, IEEE, 2016.
[8] A. Saxena, L. Wong, M. Quigley, and A. Y. Ng, “A vision-based system for grasping novel objects in cluttered environments,” in Robotics research, pp. 337–348, Springer, 2010.
[9] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother, “Learning 6d object pose estimation using 3d object coordinates,” in ECCV, pp. 536–551, Springer, 2014.
[10] A. Doumanoglou, R. Kouskouridas, S. Malassiotis, and T.-K. Kim, “Recovering 6d object pose and predicting next-best-view in the EGOCN framework,” in CVPR, pp. 3583–3592, 2016.
[11] T. Hodan, P. Haluza, Š. Obdržálek, J. Matas, M. Lourakis, and X. Zabulis, “T-less: An rgb-d dataset for 6d pose estimation of texture-less objects,” in WACV, pp. 880–888, IEEE, 2017.
[12] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon, “Scene coordinate regression forests for camera relocalization in rgb-d images,” in CVPR, pp. 2930–2937, 2013.
[13] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab, “Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again,” in ICCV, Oct 2017.
[14] F. Manhardt, W. Kehl, N. Navab, and F. Tombari, “Deep model-based 6d pose refinement in rgb,” in ECCV, September 2018.
[15] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” International Journal of Computer Vision, vol. 60, no. 2, pp. 91–110, 2004.
[16] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” Communications of the ACM, vol. 24, no. 6, pp. 381–395, 1981.
[17] A. Glent Buch, Y. Yang, N. Kruger, and H. Gordon Petersen, “In search of inliers: 6d correspondence by local and global voting,” in CVPR, pp. 2067–2074, 2014.
[18] X. Chen and B. Bhanu, “3d free-form object recognition in range images using local surface patches,” Pattern Recognition Letters, vol. 28, no. 10, pp. 1252–1262, 2007.
[19] F. Tombari and L. Di Stefano, “Object recognition in 3d scenes with occlusions and clutter by hough voting,” in PSIVT, pp. 349–355, IEEE, 2010.
[20] P. V. Hough, “Method and means for recognizing complex patterns,” Dec. 18 1962. US Patent 3,069,654.
[21] F. Tombari, S. Salti, and L. Di Stefano, “Object recognition in 3d scenes with occlusions and clutter by hough voting,” in CVPR, pp. 349–355, IEEE, 2010.
[22] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” Communications of the ACM, vol. 24, no. 6, pp. 381–395, 1981.
[23] F. Glent Buch, Y. Yang, N. Kruger, and H. Gordon Petersen, “In search of inliers: 6d correspondence by local and global voting,” in CVPR, pp. 2067–2074, 2014.
[24] K. Chen and B. Bhanu, “3d free-form object recognition in range images using local surface patches,” Pattern Recognition Letters, vol. 28, no. 10, pp. 1252–1262, 2007.
[25] F. Tombari and L. Di Stefano, “Object recognition in 3d scenes with occlusions and clutter by hough voting,” in PSIVT, pp. 349–355, IEEE, 2010.
[26] P. V. Hough, “Method and means for recognizing complex patterns,” Dec. 18 1962. US Patent 3,069,654.
[27] F. Tombari, S. Salti, and L. Di Stefano, “Object recognition in 3d scenes with occlusions and clutter by hough voting,” in CVPR, pp. 349–355, IEEE, 2010.
[28] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” Communications of the ACM, vol. 24, no. 6, pp. 381–395, 1981.