

# Benchmarking pose estimation for robot manipulation<sup>☆</sup>

Antti Hietanen<sup>a,b,\*</sup>, Jyrki Latokartano<sup>b</sup>, Alessandro Foi<sup>a</sup>, Roel Pieters<sup>b</sup>, Ville Kyrki<sup>c</sup>,  
Minna Lanz<sup>b</sup>, Joni-Kristian Kämäräinen<sup>a</sup>

<sup>a</sup> Computing Sciences, Tampere University, Finland

<sup>b</sup> Automation Technology and Mechanical Engineering, Tampere University, Finland

<sup>c</sup> Department of Electrical Engineering and Automation, Aalto University, Finland

## ARTICLE INFO

### Article history:

Received 28 November 2019

Received in revised form 30 March 2021

Accepted 10 May 2021

Available online 18 May 2021

MSC:

00-01

99-00

### Keywords:

Object pose estimation

Robot manipulation

Evaluation

## ABSTRACT

Robot grasping and manipulation require estimation of 3D object poses. Recently, a number of methods and datasets for vision-based pose estimation have been proposed. However, it is unclear how well the performance measures developed for visual pose estimation predict success in robot manipulation. In this work, we introduce an approach that connects error in pose and success in robot manipulation, and propose a probabilistic performance measure of the task success rate. A physical setup is needed to estimate the probability densities from real world samples, but evaluation of pose estimation methods is offline using captured test images, ground truth poses and the estimated densities. We validate the approach with four industrial manipulation tasks and evaluate a number of publicly available pose estimation methods. The popular pose estimation performance measure, Average Distance of Corresponding model points (ADC), does not offer any quantitatively meaningful indication of the frequency of success in robot manipulation. Our measure is instead quantitatively informative: e.g., a score of 0.24 corresponds to average success probability of 24%.

© 2021 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

A common task in robotics is object manipulation. Successful object manipulation requires accurate object 3D pose estimates so that the robot end effector can be successfully positioned and closed. There is a number of works that focus on *object grasping* [1–5], but these works do not address *precision manipulation* where precise grasping is needed for task completion. In this work, we focus on industrial assembly where precision manipulation is needed. The manipulated objects are grasped and manipulation tasks executed using estimates from vision-based pose estimation methods.

Many pose estimation methods use point clouds as input [6–8]. They estimate the object 6D pose parameters, three translation and three orientation variables, by matching stored point cloud models to observed point clouds. A number of offline datasets have been proposed to evaluate pose estimation methods [9–13]. It is noteworthy that most of the datasets have been proposed for vision research community and only a few have connection to robot manipulation [13]. The two most popular performance measures used in the datasets are (1) *Absolute Translation and*

*Orientation error* and (2) *Average Distance of Corresponding model points* (ADC). While the absolute errors in translation and orientation are intuitive choices, ADC has recently become more popular as it provides a single number for method comparison. A variant of ADC is used in the annual 6D pose estimation challenge that contains multiple datasets<sup>1</sup> [9]. However, it remains unclear how well the two performance measures predict success in robot precision manipulation. Precision manipulation depends on many factors beyond vision, for example, the manipulated object dimensions and material, the selected gripper, the selected grasping point and the task itself.

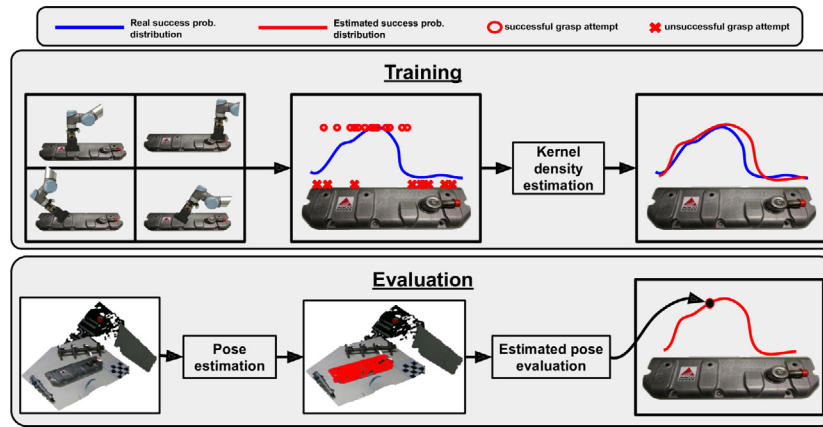
We propose a novel approach to benchmark object poses so that the performance indicates success in real manipulation tasks. The main contributions are:

- An approach that connects the object pose error and success in a manipulation task. Success is modeled as a probability density that is estimated by sampling with a real physical setup (“Training” stage in Fig. 1). The probability model allows to collect an offline test dataset that does not require the physical setup (“Evaluation” in Fig. 1).
- A statistical formulation of the success in a robot manipulation task (Section 3.1). The formulation provides intuitive performance numbers for robot manipulation. For example,

<sup>☆</sup> This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 825196.

\* Corresponding author at: Computing Sciences, Tampere University, Finland.  
E-mail address: [antti.hietanen@tuni.fi](mailto:antti.hietanen@tuni.fi) (A. Hietanen).

<sup>1</sup> <https://bop.felk.cvut.cz/challenges/>



**Fig. 1.** Illustration of our approach. “Training” stage requires a physical setup to sample pose errors. The samples are used to estimate the probabilistic model of success probability in the given task. “Evaluation” stage does not require the physical setup and is used in offline evaluation of pose estimation methods (with test images, ground truth poses of objects in test images, and the estimated probability densities).

0.9 means that on average ninety out of one hundred attempts succeed if the given pose estimate is used with a real physical setup.

- Kernel regression based method and algorithm (Section 3.2 and Alg. 1) to sample and estimate the real success probabilities (“Training”).
- A public test dataset using our approach. The dataset contains four different industrial manipulations tasks with four different objects. Probability densities are estimated using more than 3,000 samples with the physical setups.

In the experimental part, we validate our approach by evaluating a number of public 3D pose estimation methods with the new test dataset and performance measure.

## 2. Related work

*Pose estimation datasets.* A seminal benchmark for 3D pose estimation is *LineMod* by Hinterstoisser et al. [12]. They introduced a data collection protocol and performance metric used in more recent benchmarks. *LineMod* training data contains 3D reconstructions (object models) and the original turn-table captured RGB and depth images used for dense reconstruction. *LineMod* test data consists of various cluttered scenes that were captured from multiple view points using RGB and depth cameras. Hinterstoisser et al. propose *Average Distance of Corresponding points* (ADC) performance metric that computes the average distance of 3D model points between the ground truth and estimated pose. ADC provides a single performance number that measure how well the two surfaces fit. However, since certain objects, such as “bottle” are symmetric, Hinterstoisser et al. later proposed a symmetry invariant ADC where the distances are measured between the closest and not the corresponding points. There are also other popular performance metrics, such as the *absolute translation* and *rotation errors* [14], but ADC is more widely used as it provides a single value for method comparison.

A recent effort to unify datasets and evaluation protocols for 3D pose estimation is the *BOP Benchmark for 6D Object Pose Estimation* by Hodan et al. [9]. BOP includes *LineMod* and 7 other datasets that are all acquired similarly to *LineMod*. BOP includes more variety in object types varying from household objects [15] to industrial parts [16]. The evaluation protocol of BOP is a further adaption of the symmetry-invariant ADC called *Visible Surface Discrepancy* (VSD) [11]. VSD takes into account view point dependent pose uncertainty and measures the ADC error only for visible points of the object.

The main difference between our work and BOP is that their evaluation is not connected to any *task* that requires object pose estimation. BOP provides only vision-based evaluation of pose parameters. As a more suitable approach for robotics, our work connects the error in pose estimate to a specific robot manipulation task and therefore answers to the question: “Does the robot succeed in the task given the measured error in pose estimate?”

*Pose estimation methods.* In the experimental part of this work, a number of publicly available 3D pose estimation methods are compared using the introduced dataset and the proposed performance measure. The methods are selected among the best performing ones in the recent evaluation of Yang et al. [10] and are described in more details in Section 4.3.

## 3. Object pose evaluation for robot manipulation

A standard procedure in industrial robotics is to manually set up and program robot manipulation tasks. An experienced engineer manually selects a stable grasp location and selects a suitable gripper and fingers. A physical part feeder guarantees precision feeding of the parts and therefore pre-programmed grasping and manipulation perform with a high success rate. It is difficult to make a generic feeder and therefore feeder design makes it difficult to reconfigure the robot setup for new tasks. The feeder is not needed if there is a method that can estimate the object pose with sufficient accuracy.

### 3.1. Statistical model of task success $P(X = 1)$

The success of a robot to complete its task is a binary random variable  $X \in \{0, 1\}$ , where a successful attempt  $X = 1$  occurs with probability  $p$  and unsuccessful attempt (failure)  $X = 0$  occurs with  $1 - p$ . Therefore,  $X$  follows the Bernoulli distribution,  $P(X|p) = p^X(1 - p)^{1-X}$ , with complementary probability of success and failure:  $E(X) = P(X = 1) = 1 - P(X = 0)$ , where  $E$  denotes the mathematical expectation. The pose is defined by 6D pose coordinates  $\theta = (t_x, t_y, t_z, r_x, r_y, r_z)^T$ . The translation vector  $(t_x, t_y, t_z)^T \in \mathbb{R}^3$  and 3D rotation  $(r_x, r_y, r_z)^T \in SO(3)$  both have three degrees of freedom. The rotation is in axis-angle representation, where the length of the 3D rotation vector is the amount of rotations in radians, and the vector itself gives the axis about which to rotate. Adding pose to the formulation makes the success probability a conditional distribution and expectation a conditional expectation. The conditional probability of a successful attempt is

$$p(\theta) = E(X|\theta) = P(X = 1|\theta) = 1 - P(X = 0|\theta). \quad (1)$$

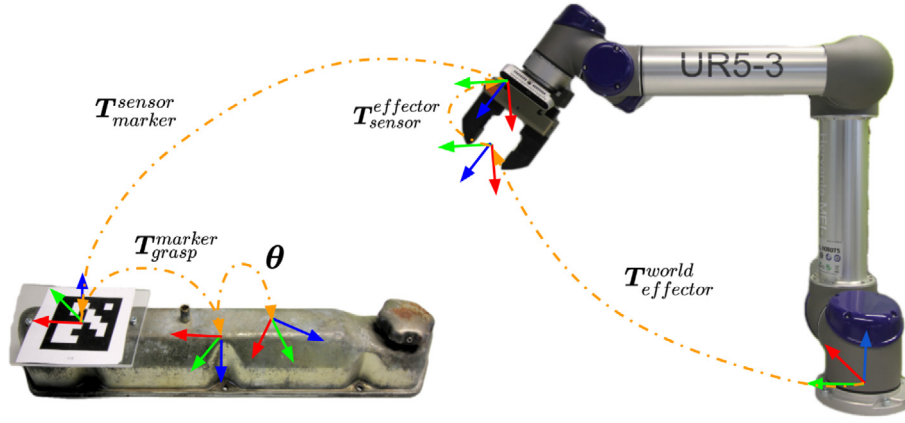


Fig. 2. Coordinate frames used in the pose sampling algorithm.

The maximum likelihood estimate of the Bernoulli parameter  $p \in [0, 1]$  from  $N$  homogeneous samples  $y_i$ ,  $i = 1, \dots, N$ , is the sample average

$$\hat{p}_{\text{ML}} = \frac{1}{N} \sum_{i=1}^N y_i, \quad (2)$$

where homogeneity means that all samples are realization of a common Bernoulli random variable with unique underlying parameter  $p$ . However, guaranteeing homogeneity would require that the samples  $\{y_i, i = 1, \dots, N\}$  were either all collected at the same pose  $\theta_1 = \dots = \theta_N$ , or for different poses that nonetheless yield same probability  $p(\theta_1) = \dots = p(\theta_N)$ , i.e. it would require us either to collect multiple samples for each  $\theta \in SE(3)$  or to know beforehand  $p$  over  $SE(3)$  (which is what we are trying to estimate). This means that in practice  $p$  must be estimated from non-homogeneous samples, i.e. from  $\{y_i, i = 1, \dots, N\}$  sampled at pose  $\{\theta_i, i = 1, \dots, N\}$  which can be different and having different underlying  $\{p(\theta_i), i = 1, \dots, N\}$ .

The actual form of  $p$  over  $SE(3)$  is unknown and depends on many factors, e.g., the shape of an object, properties of a gripper and a task to be completed. 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 attempt* as

$$\hat{p}_{\mathbf{h}}(\theta) = \frac{\sum_{i=1}^N y_i K_{\mathbf{h}}(\theta_i - \theta)}{\sum_{i=1}^N K_{\mathbf{h}}(\theta_i - \theta)}, \quad (3)$$

where  $\theta_i$  denotes the poses at which  $y_i$  has been sampled and  $K_{\mathbf{h}} : \mathcal{E} \rightarrow \mathbb{R}^+$  is a non-negative multivariate kernel with vector scale  $\mathbf{h} = (h_{t_x}, h_{t_y}, h_{t_z}, h_{r_x}, h_{r_y}, h_{r_z})^T > 0$ .

In this work,  $K_{\mathbf{h}}$  is the multivariate Gaussian kernel

$$K_{\mathbf{h}}(\theta) = G\left(\frac{t_x}{h_{t_x}}\right) G\left(\frac{t_y}{h_{t_y}}\right) G\left(\frac{t_z}{h_{t_z}}\right) \sum_{j \in \mathbb{Z}} G\left(\frac{r_x + 2j\pi}{h_{r_x}}\right) \sum_{j \in \mathbb{Z}} G\left(\frac{r_y + 2j\pi}{h_{r_y}}\right) \sum_{j \in \mathbb{Z}} G\left(\frac{r_z + 2j\pi}{h_{r_z}}\right), \quad (4)$$

where  $G$  is the standard Gaussian bell,  $G(\theta) = (2\pi)^{-\frac{1}{2}} e^{-\frac{1}{2}\theta^2}$ . The three sum terms in (4) realize the modulo- $2\pi$  periodicity of  $SO(3)$ .

The performance of the estimator (3) is heavily affected by the choice of  $\mathbf{h}$ , which determines the influence of samples  $y_i$  in computing  $\hat{p}_{\mathbf{h}}(\theta)$  based on the difference between the estimated and sampled poses  $\theta$  and  $\theta_i$ . Indeed, the parameter  $\mathbf{h}$  can be interpreted as reciprocal to the bandwidth of the estimator: too large  $\mathbf{h}$  results in excessive smoothing whereas too small results in localized spikes.

To find an optimal  $\mathbf{h}$ , 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:

$$\hat{p}_{\mathbf{h}}^{\text{LOO}}(\theta, i) = \frac{\sum_{j \neq i} y_j K_{\mathbf{h}}(\theta_j - \theta)}{\sum_{j \neq i} K_{\mathbf{h}}(\theta_j - \theta)}.$$

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

$$\hat{\mathbf{h}} = \arg \max_{\mathbf{h}} \sum_{i|y_i=1} \log(\hat{p}_{\mathbf{h}}^{\text{LOO}}(\theta, i)) + \sum_{i|y_i=0} \log(1 - \hat{p}_{\mathbf{h}}^{\text{LOO}}(\theta, i)).$$

Our choices of the kernel and LOO optimization of the kernel parameters result to probability estimates that are verifiable by controlled experiments (as illustrated in Fig. 4).

### 3.2. Sampling the pose space

Section 3.1 provides us a formulation of the probability of successful robotic manipulation given the object relative grasp pose  $P(X = 1|\theta)$ . The practical realization of the probability values is based on Nadaraya–Watson non-parametric kernel estimator that requires a number of samples in various poses  $\theta_i$  and information of success  $y_i = 1$  or failure  $y_i = 0$  for each attempt. In this stage, a physical setup is needed for sampling, but the users of the benchmark do not need to replicate the setup. For practical reasons we make the following assumptions:

- We assume a user defined *canonical grasp pose* with respect to the manipulated object. The user can freely select the canonical pose based on the object intrinsic parameters (e.g. the distribution of mass and object dimensions) and task requirements (i.e. on which way the object is being installed). During the training stage (pose samples) the canonical pose is obtained automatically using 2D markers and cameras.
- The pose space is randomly sampled around the canonical grasp pose. Each sample  $\theta = (t_x, t_y, t_z, r_x, r_y, r_z)^T$  defines  $SE(3)$  “displacement” from the canonical grasp pose. Sampling was initialized by first finding the success limits of each dimension. The limits were found by step wise guiding the end effector away from the canonical pose until the task execution starts to fail on every attempt. The limits found for the objects in our experiments are listed in Table 1.
- The estimated probability models were validated by sampling each dimension separately on grid points and executing the task ten times on each point with real robot. The averaged task success rate on real robot was then compared

against the proposed models and the estimated probabilities matched well as can be seen in Fig. 4.

With the help of the above assumptions we define a sampling procedure that records samples and their success/failure automatically.

### 3.2.1. Coordinate transformations

In our notation  $\mathbf{T}_B^A$  denotes a  $4 \times 4$  homogeneous transformation matrix that describes the position of the frame B origin and the orientation of its axes, relative to the reference frame A. The setup used in our experiments is illustrated in Fig. 2. The transformations are:

- $\mathbf{T}_{grasp}^{marker}$  – a constant transformation from the canonical grasp pose to the marker frame;
- $\mathbf{T}_{marker}^{sensor}$  – camera and marker obtained transformation from the marker frame to the sensor frame;
- $\mathbf{T}_{sensor}^{effector}$  – a constant transformation from the sensor frame to the robot end effector frame (camera is attached to the end effector);
- $\mathbf{T}_{effector}^{world}$  – robot kinematics obtained transformation from the end effector frame to the world frame.

The world frame is fixed to the robot frame (i.e. center of the robot base). The robot program is based on moving the tool point that is the end effector frame. The coordinate transformation  $\mathbf{T}_{effector}^{world}$  can be automatically calculated using the joint angles and known kinematic equations.  $\mathbf{T}_{sensor}^{effector}$  is computed using the standard procedure for hand-eye calibration with a printed chessboard pattern [17]. Automatic and accurate estimation of the object pose during the sampling is realized by attaching 2D markers to the manipulated objects (see Fig. 3). For a calibrated camera the ArUco library [18] provides real-time poses of the marker with respect to the sensor frame  $\mathbf{T}_{marker}^{sensor}$ . The constant offset  $\mathbf{T}_{grasp}^{marker}$  from the marker to the actual grasp pose is object-marker specific. The offset is estimated manually by hand-guiding the end effector to the desired grasp location on the object (canonical grasp pose) and measuring the difference between this pose and the marker pose:

$$\mathbf{T}_{grasp}^{marker} = (\mathbf{T}_{marker}^{world})^{-1} \mathbf{T}_{grasp}^{world}$$

During pose sampling, the canonical grasp pose is calculated with respect to the world frame from:

$$\mathbf{T}_{grasp}^{world} = \mathbf{T}_{effector}^{world} \cdot \mathbf{T}_{sensor}^{effector} \cdot \mathbf{T}_{marker}^{sensor} \cdot \mathbf{T}_{grasp}^{marker} \quad (5)$$

Finally, samples around the canonical grasp pose are generated from

$$\hat{\mathbf{T}}_{grasp}^{world} = \mathbf{T}_{grasp}^{world} \cdot \Phi(\theta) \quad (6)$$

where  $\Phi(\cdot)$  maps the 6D pose vector to a  $4 \times 4$  matrix representation

$$\Phi(\theta) = \begin{bmatrix} \mathbf{R}_{3 \times 3} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}. \quad (7)$$

The generated pose sample is defined in the vicinity of the canonical pose by the translation shift  $\mathbf{t} = (t_x, t_y, t_z)^T$  and rotation matrix  $\mathbf{R} \in \mathbb{R}^{3 \times 3}$  constructed from the axis-angle vector  $(r_x, r_y, r_z)^T$ .

### 3.2.2. Automatic failure detection

The objects in our experiments have a predefined position and orientation how they should be installed. For example, the motor parts have to be placed on the motor block precisely in order to fasten the screws. The task is to place the part to the installation pose and release it by opening the gripper fingers. In addition, excessive force during the task can cause damage to the manipulated objects. Therefore success is detected based on two factors:

- After release the part must be within pre-set limits in the correct installation pose.
- The online measured wrench torque at the end effector must remain below a pre-set limit during the task.

The limits are task specific and are set during the system setup. For pose limits, two thresholds were used:  $\tau_t$  for the maximum translation error and  $\tau_r$  for the maximum orientation error. These are computed between the camera obtained installation pose  $\hat{\mathbf{T}} = [\hat{\mathbf{R}} | \hat{\mathbf{t}}]$  and the ground truth installation pose  $\mathbf{T} = [\mathbf{R} | \mathbf{t}]$ . The task was marked successful if

$$\begin{aligned} \|\mathbf{t} - \hat{\mathbf{t}}\| &\leq \tau_t \\ \arccos\left(\frac{\text{trace}(\hat{\mathbf{R}}\mathbf{R}^{-1}) - 1}{2}\right) &\leq \tau_r \end{aligned} \quad (8)$$

The torque is used to detect if the robot collides with its environment. 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. The robot's internal sensors provide the torque measurements  $\mathbf{F} = (f_x, f_y, f_z)$ , where  $f_x, f_y$  and  $f_z$  are the forces in the axes of the robot frame coordinates and measured in Newtons. For each task the limit  $f_{max}$  was manually set for each operation stage during the system setup. If  $\|\mathbf{F}\| > f_{max}$  at any stage of the task, then the attempt was labeled as failure. Automatic pose sampling procedure is defined in Algorithm 1.

#### Algorithm 1: Automatic pose sampling

**Input:** Waypoints  $\mathcal{W} := \{w_{start}, w_{grasp}, w_{install}\}$ ; Num. of samples  $S$

**Output:** Set of samples  $\{(\theta_i, y_i)\}_{i=1, \dots, S}$

```

1 for  $i = 1$  to  $S$  do
2   Init:  $y_i \leftarrow \text{success}$ ;  $\theta_i \leftarrow$ 
   SampleRandomDisplacement();
3    $\mathbf{T}_{marker}^{sensor} \leftarrow \text{DetectMarker}()$ ;
4    $\mathbf{T}_{sensor}^{world} \leftarrow \text{ComputeForwardKinematics}()$ ;
5    $\hat{\mathbf{T}}_{grasp}^{marker} \leftarrow \text{AddDisplacement}(\theta_i, \mathbf{T}_{grasp}^{marker})$ ;
6    $\hat{\mathbf{T}}_{grasp}^{world} \leftarrow \mathbf{T}_{sensor}^{world} \cdot \mathbf{T}_{marker}^{sensor} \cdot \hat{\mathbf{T}}_{grasp}^{marker}$ ;
7   GraspObject( $\hat{\mathbf{T}}_{grasp}^{world}, w_{grasp}$ );
8   if NOT SuccessfulGrasp() then
9      $y_i \leftarrow \text{failure}$ 
10  else
11    InstallObject( $w_{install}$ );
12    if NOT SuccessfulInstall() then
13       $y_i \leftarrow \text{failure}$ 
14    Store( $\theta_i, y_i$ );
15    MoveObjectToStart( $w_{start}$ );
```

## 4. Experiments

To experimentally validate our approach we collected data from four industrial assembly tasks and evaluated a number of publicly available pose estimation methods.

### 4.1. Physical setup

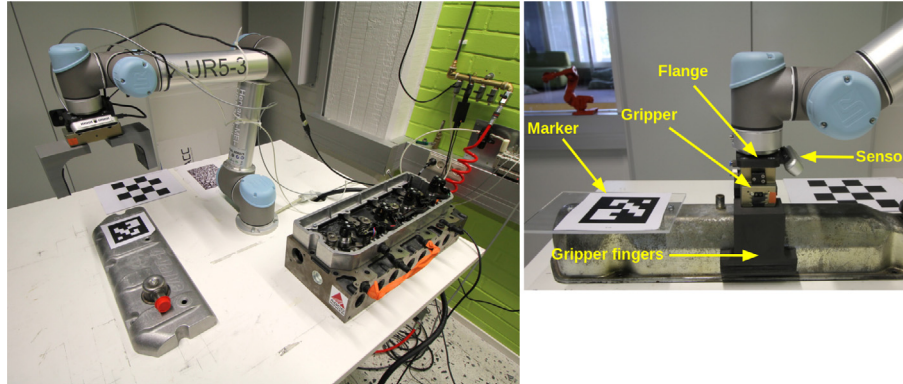
Fig. 3 illustrates the physical setup used in the experiments. The setup consists of a model 5 Universal Robot Arm (UR5) and a Schunk PGN-100 gripper. The gripper operates pneumatically and was configured to have a high gripping force (approximately 600N) to prevent object slippage. In addition, the gripper has custom 3D printed fingers plated with rubber. For visual perception, Intel RealSense D415 RGB-D sensor was secured on a



**Table 1**

Sampling limits for translation ( $t_x, t_y, t_z$ ) and rotation ( $r_x, r_y, r_z$ ) in meters and degrees, respectively. Beyond these limits the task always fails.

Variable	Task name			
	Task 1	Task 2	Task 3	Task 4
$t_x$	$[-9.0, 9.0] \cdot 10^{-3}$	$[-6.0, 6.0] \cdot 10^{-3}$	$[-9.0, 9.0] \cdot 10^{-3}$	$[-6.5, 8.5] \cdot 10^{-3}$
$t_y$	$[-1.0, 1.0] \cdot 10^{-3}$	$[-3.0, 2.5] \cdot 10^{-3}$	$[-5.0, 6.0] \cdot 10^{-3}$	$[-2.1, 2.1] \cdot 10^{-2}$
$t_z$	$[-1.0, 5.0] \cdot 10^{-3}$	$[-2.0, 4.0] \cdot 10^{-3}$	$[-2.0, 5.0] \cdot 10^{-3}$	$[-1.2, 1.7] \cdot 10^{-2}$
$r_x$	$[-6.3, 6.3] \cdot 10^0$	$[-6.3, 6.3] \cdot 10^0$	$[-2.0, 1.0] \cdot 10^0$	$[-1.5, 1.5] \cdot 10^1$
$r_y$	$[-5.0, 5.0] \cdot 10^{-1}$	$[-2.5, 1.0] \cdot 10^0$	$[-2.0, 2.0] \cdot 10^0$	$[-1.5, 1.5] \cdot 10^1$
$r_z$	$[-5.0, 5.0] \cdot 10^{-1}$	$[-1.5, 1.5] \cdot 10^0$	$[-4.0, 4.0] \cdot 10^0$	$[-1.5, 1.5] \cdot 10^1$



**Fig. 3.** The setup used to sample pose errors of the engine cap 1. The task is to pick and move the cap to the engine block. Failures in task execution were automatically detected during sampling (see Section 3.2 for details).

**Table 2**

Summary of the data used in our experiments. Training samples were used to estimate the probability densities. Training samples required execution of the task and took approximately 45–55 s each. Test images were used for method evaluation.

Task	Training samples	Test images	Description
1	3416	152	Pick and install (motor cap 1)
2	3805	152	Pick and install (motor frame)
3	3722	161	Pick and install (motor cap 2)
4	3550	158	Pick and drop (faceplate)

3D printed flange and mounted between the gripper and the robot end effector. 3D prints were made in-house using nylon reinforced with carbon fiber to tolerate external forces during the experiments. The computation was performed on a single laptop with Ubuntu 18.04. All tasks and the preset grasp poses were validated by executing the task 100 times with poses obtained from the camera system (Section 3.2). No failures occurred during the setup validation. On average, successful executions took 45–55 s and in 24 h the robot was able to execute approximately 1,100 attempts. The setup recovered automatically from most of the failure cases (dropping the object, object collision, etc.). Only if the printed marker was occluded or the manipulated object was jammed against the parts of the engine, the system was restarted manually.

#### 4.2. Benchmark dataset

The benchmark dataset for testing requires only the images from the test scenes, part ground truth poses, and the estimated success probability densities. The training samples and the physical setup are needed in the “training stage” where the probability densities are estimated. The dataset used in the experiments is summarized in Table 2.

The three first tasks were selected from the production line of a local engine manufacturing company: motor cap 1 assembly (Task 1), motor frame assembly (Task 2) and motor cap 2

assembly (Task 3). The fourth task is a validation task that does not require precise manipulation and is used to sanity check the evaluation methods. The faceplate part used in Task 4 bin picking is from the Cranfield assembly benchmark. The tasks were programmed by a team of experienced lab engineers who carefully selected the grippers and fingers.

*Success probability estimation.* Tested methods were evaluated using their test image pose estimates. Given the estimates and the stored ground truth poses the average success probabilities over all test images is computed. The success probability is calculated using the probability model  $P(X = 1|\theta)$  in Section 3.1. The probability densities were estimated using the kernel density model and the sampling procedure described in Section 3.2. For the Tasks 1–4 the densities were estimated using 3,416–3,805 samples (Table 2). The true and estimated probabilities for Task 3 are illustrated in Fig. 4 where they match well.

*Test models.* Test models are full 3D models of the four parts used in Tasks 1–4. They are stored as 3D point clouds  $\{\mathbf{x}_i | i = 1, \dots, N\}$  and RGB color vectors  $\mathbf{c}_i$ . The part point clouds were obtained by moving the robot arm with the attached RGB-D sensor around each part (Fig. 5). The captured point clouds were manually verified and all artifacts and redundant parts of the reconstructed point cloud were removed using the MeshLab software [19].

*Test scenes.* Test scenes were captured by moving the RGB-D camera to arbitrary locations and capturing a point cloud. No manual cleaning of the data was performed. 152–161 test images were collected under three different settings: (1) a single target object present (ideal case), (2) multiple objects present (background clutter) and (3) multiple objects and partial occlusion. Ground truth poses were obtained through robot kinematics and stored using the part model coordinate system as the world frame.

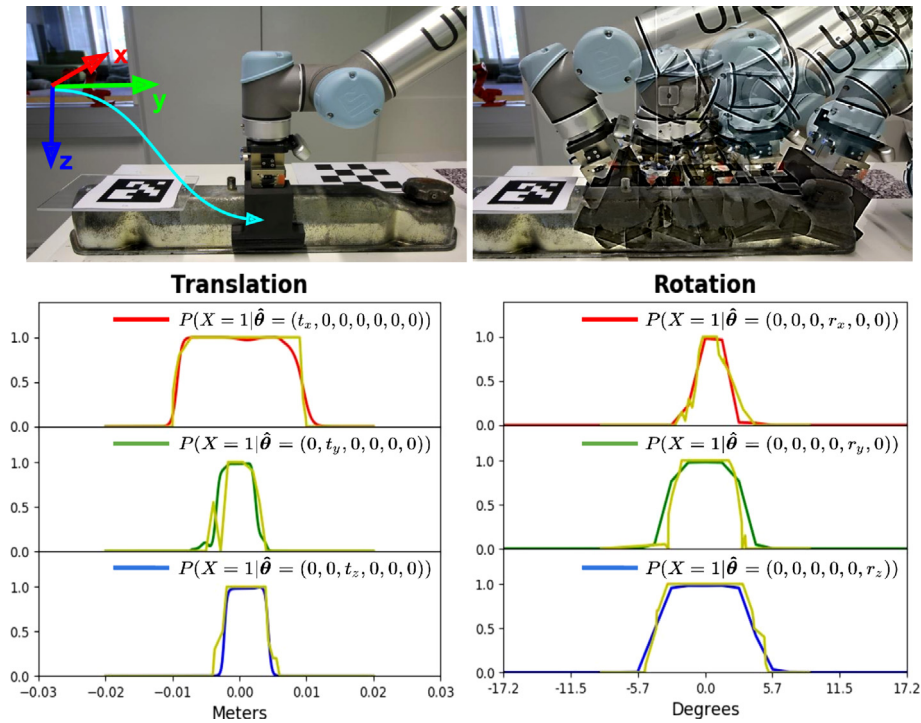


Fig. 4. Grasping of the motor cap 2 used in Task 3. The used coordinate system is object centric (top left) and pose samples are taken around the pre-defined “canonical grasp pose” (top right). The six estimated success probabilities (red, green and blue lines) match well with the true success probabilities (yellow line). The six graphs correspond to the three translation axes and the three rotation angles in 3D.

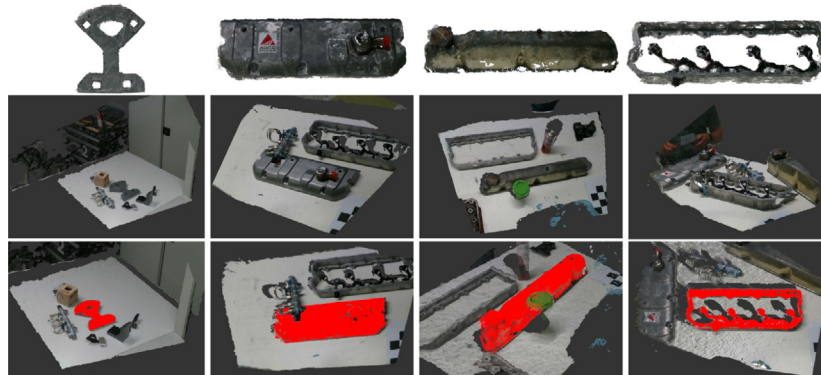


Fig. 5. Top: Point cloud models of the parts: faceplate, motor cap 1, motor cap 2 and a motor frame (points clouds acquired by multiview capturing). Middle: example test scenes. Bottom: models rendered to the scene using ground truth pose.

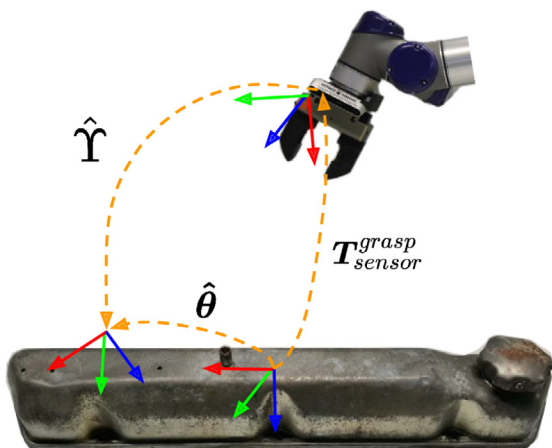


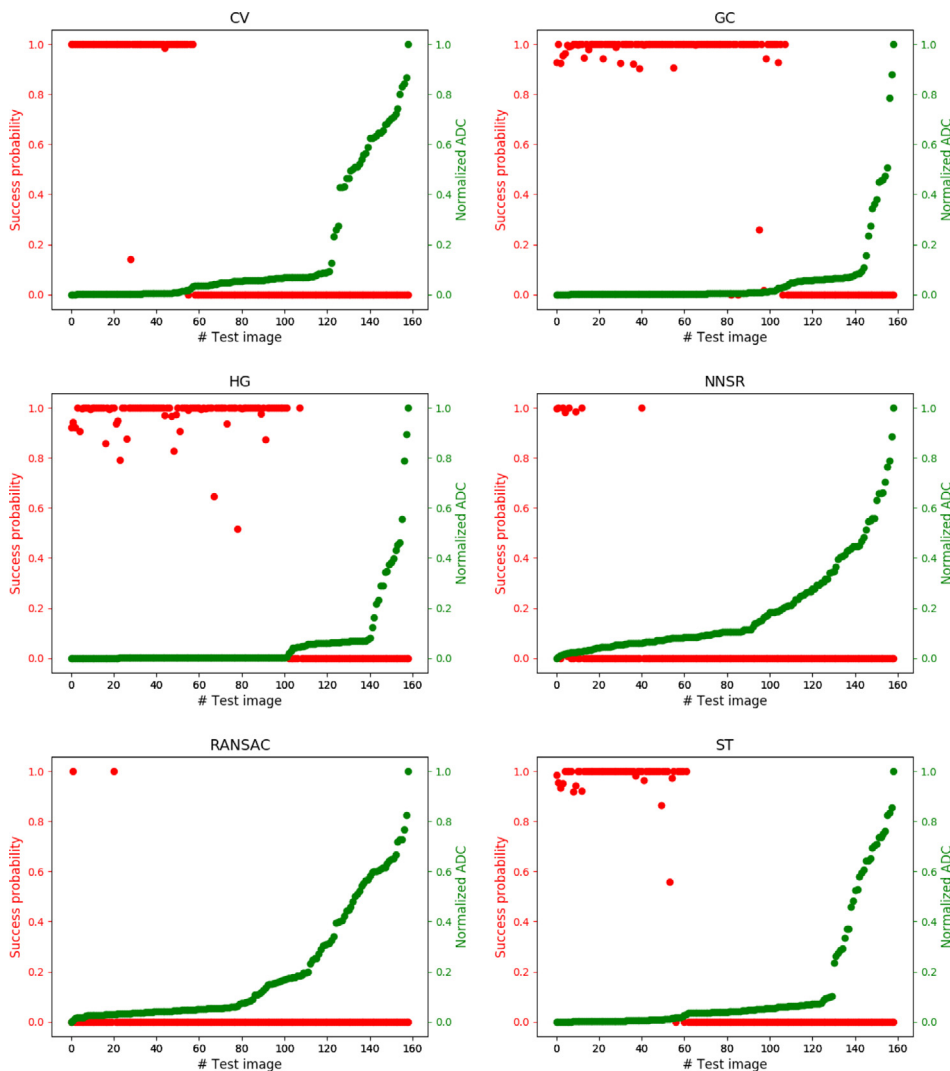
Fig. 6. Coordinate frames in the evaluation procedure.

### 4.3. Pose estimation baselines

The baseline methods were selected among the best performing methods in the recent evaluation by Yang et al. [10].

*Random sample consensus (RANSAC).* RANSAC is a widely used technique for 6D pose estimation [20–22]. It is an iterative process that uses random sampling technique to generate candidate transformations that align the two surfaces. The design parameter of the method is  $N_{RANSAC}$  which is the maximum count of transformation. The transformations are evaluated by transforming all points and calculating the Euclidean distance between the corresponding points. Matches with distance less than  $d_{RANSAC}$  are counted as inliers. The final pose is estimated using all inlier points for the transformation with the largest number of inliers.

*Hough transform (HG).* Hough transform [23] goes through all point correspondences which cast votes and pose with the largest number of votes is selected. There are variants available [24,25]



**Fig. 7.** ADC pose error (green) and success probability (red) on the Task 4 test set for the six tested methods. Images (x-axis) are sorted according to their ADC. Note the rapid drops from successful (1.0) to unsuccessful (0.0) estimates while the ADC error produces smooth curves without clear indication of success.

from which the Hough Grouping (HG) by Tombari et al. [25] was selected. For fast computation, the method uses a unique model reference point (mass centroid) and bins represent pose around the reference point. To make correspondence points invariant to rotation and translation between the model and scene, every point is associated with a local reference frame [26]. The main parameter of the method is the pose bin size – coarse size provides faster computation but increases pose uncertainty.

*Spectral technique (ST).* Leordeanu and Hebert [27] proposed a spectral grouping technique to find coherent clusters from the initial set of feature matches. The method takes into account the relationship between points and correspondences and finally uses an eigen-decomposition to estimate the confidence of a correspondence to be an inlier.

*Geometric consistency (GC).* While the RANSAC and Hough transform based methods operate directly on the 3D points there are methods that exploit the local neighborhood of points to establish more reliable matches between model and scene point clouds [6, 28]. Geometric Consistency Grouping (GC) [28] is a strong baseline and it has been implemented in several point cloud libraries.

GC works independently from the feature space and utilizes only the spatial relationship of the corresponding points. The algorithm evaluates the consistency of two correspondences  $c_i$  and  $c_j$  using a compatibility score





$$d(c_i, c_j) = \left| \|\mathbf{x}_i - \mathbf{x}_j\| - \|\mathbf{x}'_i - \mathbf{x}'_j\| \right| < \tau_{GC}. \tag{9}$$

GC simply measures distances near the points and assigns correspondences to the same cluster if their geometric inconsistency is smaller than the threshold value  $\tau_{GC}$ .





*Search of inliers (SI).* A recent method by Buch et al. [6] achieves state-of-the-art 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 between a model and scene, and the score is computed using their pair-wise similarity score  $s_l(c)$ . 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 combining the local and global scores, and finally  $s(c)$  are thresholded to inliers and outliers based on Otsu’s bimodal distribution thresholding.

**Table 3**

Comparison of pose estimation methods with our dataset (single: single object in the scene; multi: multiple objects (clutter); occ: multiple objects and occlusion; all: average over all test samples).

Task: Task 1								Task: Task 2							
Method	Part: Motor cap 1; Gripper: Shunker Fingers: Custom made							Part: Motor frame; Gripper: Shunker Fingers: Custom made							
															
Average success probability				%{p ≥ 0.9}		Avg. ADC		Average success probability				%{p ≥ 0.9}		Avg. ADC	
	single	multi	occ	all	all	all	best-25%	single	multi	occ	all	all	all	best-25%	
GC [28]	0.24	0.18	0.12	0.19	12%	0.08	3.83·10 <sup>-3</sup>	0.21	0.22	0.19	0.21	9%	0.02	5.36·10 <sup>-3</sup>	
HG [25]	0.31	0.29	0.20	0.26	14%	0.06	3.87·10 <sup>-3</sup>	0.28	0.27	0.27	0.28	15%	0.03	5.19·10 <sup>-3</sup>	
SI [6]	0.00	0.00	0.00	0.00	0%	0.46	1.78·10 <sup>-1</sup>	0.14	0.04	0.03	0.07	5%	0.42	1.81·10 <sup>-2</sup>	
ST [27]	0.01	0.03	0.00	0.01	0%	0.35	9.12·10 <sup>-2</sup>	0.23	0.16	0.07	0.17	7%	0.34	4.38·10 <sup>-3</sup>	
NNSR [23]	0.00	0.00	0.00	0.00	0%	0.26	1.18·10 <sup>-1</sup>	0.00	0.00	0.00	0.00	0%	0.36	1.50·10 <sup>-1</sup>	
RANSAC [20]	0.00	0.00	0.00	0.00	0%	0.75	1.71·10 <sup>-1</sup>	0.00	0.00	0.00	0.00	0%	0.65	2.03·10 <sup>-1</sup>	

Task: Task 3								Task: Task 4							
Method	Part: Motor cap 2; Gripper: Shunker Fingers: Custom made							Part: Cranfield faceplate; Gripper: Shunker Fingers: Custom made							
															
Average success probability				%{p ≥ 0.9}		Avg. ADC		Average success probability				%{p ≥ 0.9}		Avg. ADC	
	single	multi	occ	all	all	all	best-25%	single	multi	occ	all	all	all	best-25%	
GC [28]	0.24	0.25	0.20	0.24	13%	0.09	6.28·10 <sup>-3</sup>	0.66	0.67	0.59	0.64	65%	0.15	4.57·10 <sup>-3</sup>	
HG [25]	0.13	0.21	0.10	0.15	9%	0.11	7.81·10 <sup>-3</sup>	0.64	0.68	0.56	0.63	60%	0.16	3.43·10 <sup>-3</sup>	
SI [6]	0.11	0.19	0.11	0.13	8%	0.09	1.11·10 <sup>-2</sup>	0.37	0.43	0.20	0.35	35%	0.39	9.94·10 <sup>-3</sup>	
ST [27]	0.17	0.18	0.08	0.15	8%	0.11	5.46·10 <sup>-3</sup>	0.40	0.39	0.30	0.37	36%	0.30	6.47·10 <sup>-3</sup>	
NNSR [23]	0.02	0.00	0.00	0.01	1%	0.19	6.10·10 <sup>-2</sup>	0.05	0.04	0.07	0.05	5%	0.28	7.16·10 <sup>-2</sup>	
RANSAC [20]	0.00	0.00	0.00	0.00	0%	0.28	1.24·10 <sup>-1</sup>	0.00	0.04	0.00	0.01	1%	0.51	1.05·10 <sup>-1</sup>	

#### 4.4. Performance indicators

The main performance metric in our work is the estimated success probability defined in Section 3.1. The probabilities were computed around the canonical grasp pose to find how the performance depends on misalignment to different dimensions. The corresponding object-relative grasp pose of the pose estimate  $\hat{\mathbf{Y}}$  is calculated as:

$$\hat{\theta} = \Phi^{-1}(\mathbf{T}_{\text{sensor}}^{\text{grasp}} \hat{\mathbf{Y}}), \quad (10)$$

where the transformation matrix  $\mathbf{T}_{\text{sensor}}^{\text{grasp}}$  defines the canonical grasp pose respect to the sensor coordinate system (see Fig. 6). The  $\Phi^{-1}(\cdot)$  operator converts the  $4 \times 4$  pose matrix to 6D vector representation. Finally, the task success is evaluated using the proposed metric as  $P(X = 1|\hat{\theta})$ . As the final performance indicators the average probability over test scenes is computed and also the proportion of images for which the probability is greater or equal to 0.90.

In addition to the proposed indicator we also report the ADC error calculated over the points transformed by the ground truth and estimated object pose as suggested in [12]. The ADC error is computed from

$$\epsilon_{\text{ADC}} = \frac{1}{|\mathcal{M}|} \sum_{\mathbf{x} \in \mathcal{M}} \|\hat{\mathbf{r}}_{\mathbf{x}} - \mathbf{r}_{\mathbf{x}}\| \quad (11)$$

where  $\mathcal{M}$  is the set of model 3D points. We also report the top-25% ADC error, which is less affected by outliers.

#### 4.5. Results

The results for all baseline methods and tasks are summarized in Table 3. The two best methods are Hough Transform (HG) by

Tombari et al. [25] and GC by Chen and Bhanu [28]. HG and GC perform considerably better than the two more recent methods SI and ST. The two simplest methods, Hough voting (NNSR) and RANSAC, are the worst. It should be noted that the ADC provides the same rank order as the proposed success probability, but ADC does not indicate success or failure rates in the tasks.

It is noteworthy that even for the simplest bin-picking task (Task 4) the best average success probability is only 0.64 and the number of accurate estimates ( $p \geq 0.9$ ) is only 65% which are clearly below what is expected for the typical assembly lines (> 99%).

*Success probability vs. ADC.* While the ADC and the proposed success probability indicators both provide similar ranking of the best methods in Table 3, it remains unclear what is the effect of pose estimation error to the task success. It turns out that in our tasks the success probability rapidly changes from “successful” to “unsuccessful” which is not indicated by the ADC metric. This is evident in Fig. 7. While the change points are partly visible in the ADC measurements the ADC indicator behaves smoothly in regions that are irrelevant for the studied task and for which the probability is nearly 0.0.

## 5. Conclusions

This work addresses the question of how to evaluate object pose estimation methods for robot manipulation. In the reported experiments, the widely used performance measure, *Average Distance of Corresponding model points (ADC)*, provides method ranking, but cannot indicate whether the task could be completed using the given estimates.

The proposed novel performance measure, *success probability*, connects the success rate and pose error, and clearly indicates



that none of the tested methods would perform well in any of the four manipulation tasks. The best success rates (GC and HG) varied from 0.24 to 0.64 meaning that only 24% to 64% of attempts would succeed on average. The proposed approach allows offline method evaluation similar to the existing datasets which is an important factor for fair comparisons and method development. The physical setup is needed to collect training data for success probability estimation and that is the laborious part of our approach.

### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

### References

- [1] R. Detry, D. Kraft, O. Kroemer, L. Bodenhausen, J. Peters, N. Krüger, J. Piater, Learning grasp affordance densities, *Paladyn J. Behav. Robot. 2* (1) (2011) 1–17.
- [2] J. Redmon, A. Angelova, Real-time grasp detection using convolutional neural networks, in: 2015 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2015, pp. 1316–1322.
- [3] M. Gualtieri, A. Ten Pas, K. Saenko, R. Platt, High precision grasp pose detection in dense clutter, in: IROS, IEEE, 2016, pp. 598–605.
- [4] L. Pinto, A. Gupta, Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours, in: ICRA, IEEE, 2016, pp. 3406–3413.
- [5] T.-T. Do, A. Nguyen, I. Reid, Affordancenet: An end-to-end deep learning approach for object affordance detection, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2018, pp. 1–5.
- [6] A. Glent Buch, Y. Yang, N. Kruger, H. Gordon Petersen, In search of inliers: 3d correspondence by local and global voting, in: CVPR, 2014, pp. 2067–2074.
- [7] F. Manhardt, W. Kehl, N. Navab, F. Tombari, Deep model-based 6D pose refinement in RGB, in: ECCV, 2018.
- [8] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, N. Navab, SSD-6D: Making RGB-based 3D detection and 6D pose estimation great again, in: ICCV, 2017.
- [9] 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, 2018, pp. 19–34.
- [10] J. Yang, K. Xian, Y. Xiao, Z. Cao, Performance evaluation of 3D correspondence grouping algorithms, in: 3DV, IEEE, 2017, pp. 467–476.
- [11] T. Hodaň, J. Matas, Š. Obdržálek, On evaluation of 6D object pose estimation, in: ECCV, Springer, 2016, pp. 606–619.
- [12] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, N. Navab, Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes, in: ACCV, Springer, 2012, pp. 548–562.
- [13] A. Hietanen, J. Halme, A.G. Buch, J. Latokartano, J.-K. Kämäräinen, Robustifying Correspondence Based 6D Object Pose Estimation, in: IEEE Int. Conf. on Robotics and Automation (ICRA), Singapore, 2017.
- [14] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, A. Fitzgibbon, Scene coordinate regression forests for camera relocalization in RGB-D images, in: CVPR, 2013, pp. 2930–2937.
- [15] A. Doumanoglou, R. Kouskouridas, S. Malassiotis, T.-K. Kim, Recovering 6D object pose and predicting next-best-view in the crowd, in: CVPR, 2016, pp. 3583–3592.
- [16] T. Hodan, P. Haluza, Š. Obdržálek, J. Matas, M. Lourakis, X. Zabulis, T-LESS: An RGB-D dataset for 6D pose estimation of texture-less objects, in: WACV, IEEE, 2017, pp. 880–888.
- [17] F.C. Park, B.J. Martin, Robot sensor calibration: solving  $ax = xb$  on the Euclidean group, *IEEE Trans. Robot. Autom.* 10 (5) (1994) 717–721.
- [18] S. Garrido-Jurado, R. Muñoz Salinas, F.J. Madrid-Cuevas, M.J. Marín-Jiménez, Automatic generation and detection of highly reliable fiducial markers under occlusion, *Pattern Recognit.* 47 (6) (2014) 2280–2292.
- [19] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, G. Ranzuglia, Meshlab: an open-source mesh processing tool, in: Eurographics Italian Chapter Conference, Vol. 2008, 2008, pp. 129–136.
- [20] M.A. Fischler, R.C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, *Commun. ACM* 24 (6) (1981) 381–395.
- [21] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, C. Rother, Learning 6d object pose estimation using 3d object coordinates, in: ECCV, Springer, 2014, pp. 536–551.
- [22] E. Brachmann, F. Michel, A. Krull, M. Yang, S. Gumhold, C. Rother, Uncertainty-driven 6D pose estimation of objects and scenes from a single RGB, in: CVPR, 2016.
- [23] P.V. Hough, Method and Means for Recognizing Complex Patterns, Google Patents, 1962, US Patent 3, 069, 654.
- [24] J. Knopp, M. Prasad, G. Willems, R. Timofte, L. van Gool, Hough transform and 3D SURF for robust three dimensional classification, in: ECCV, 2010.
- [25] F. Tombari, L. Di Stefano, Object recognition in 3d scenes with occlusions and clutter by hough voting, in: PSIVT, IEEE, 2010, pp. 349–355.
- [26] F. Tombari, S. Salti, L. Di Stefano, Unique signatures of histograms for local surface description, in: ECCV, Springer, 2010, pp. 356–369.
- [27] M. Leordeanu, M. Hebert, A spectral technique for correspondence problems using pairwise constraints, in: Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1, Vol. 2, IEEE, 2005, pp. 1482–1489.
- [28] H. Chen, B. Bhanu, 3D Free-form object recognition in range images using local surface patches, *Pattern Recognit. Lett.* 28 (10) (2007) 1252–1262.



**Antti Hietanen** received his M.Sc degree in Computer Science from the Tampere University of Technology, Finland, in 2016.

He is a member of the Computer Vision group where he focuses on computer vision and robotic applications.

He is currently pursuing the Ph.D. degree at the Tampere University and his research interests are 6D pose estimation, object recognition and cognitive robotic systems.



**Mr. Jyrki Latokartano** has over 20 years of experience in industrial robotics, robot simulation and off line programming both from research and education done at Tampere University. Recently he has been focusing on industrial human-robot collaboration.

Currently he is working at TAU as a project manager in TRHINITY DIH.



**Alessandro Foi** received the M.Sc. degree in Mathematics from the Università degli Studi di Milano, Italy, in 2001, the Ph.D. degree in Mathematics from the Politecnico di Milano in 2005, and the D.Sc.Tech. degree in Signal Processing from Tampere University of Technology, Finland, in 2007. He is Professor of Signal Processing at Tampere University, Finland.

His research interests include mathematical and statistical methods for signal processing, functional and harmonic analysis, and computational modeling of the human visual system. His work focuses on spatially adaptive (anisotropic, nonlocal) algorithms for the restoration and enhancement of digital images, on noise modeling for imaging devices, and on the optimal design of statistical transformations for the stabilization, normalization, and analysis of random data.

He is a Senior Member of the IEEE, Member of the Image, Video, and Multidimensional Signal Processing Technical Committee of the IEEE Signal Processing Society, an Associate Editor for the SIAM Journal on Imaging Sciences, and a Senior Area Editor for the IEEE Transactions on Computational Imaging.



**Roel Pieters** received his Ph.D. degree in The Netherlands at Eindhoven University of Technology in 2013. From 2013 till 2017 he was a postdoctoral researcher in ETH Zurich, Switzerland (3 years) and Aalto University, Finland (1.5 years). Currently, he works at the unit of Automation Technology and Mechanical Engineering, Tampere University, Finland, as assistant professor in the field of cognitive and collaborative robotics. His research interests are perception, cognition and autonomy for human–robot interaction, applied in industrial and domestic settings. His research has led to two spin-offs and won several design and best paper awards and he is (co-)responsible for Tampere University's recently started robotics major (fall 2017) and Tampere Robolab.



**Ville Kyrki** joined School of Electrical Engineering at Aalto University as an Associate Professor in 2012. He serves as the head of the Intelligent Robotics research group.

His research interests lie mainly in intelligent robotic systems and robotic vision with a particular emphasis on developing methods and systems that cope with imperfect knowledge and uncertain senses. His published research covers feature extraction and tracking in computer vision, visual servoing, tactile sensing, robotic grasping and manipulation, sensor fusion (especially fusion of vision and other senses), planning under uncertainty, and machine learning related to the previous. His research has been published in numerous forums in the area, including IEEE Transactions on Robotics, International Journal of Robotics Research, IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Transactions on Image Processing, and Artificial Intelligence.

and machine learning related to the previous. His research has been published in numerous forums in the area, including IEEE Transactions on Robotics, International Journal of Robotics Research, IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Transactions on Image Processing, and Artificial Intelligence.



**Mrs. Minna Lanz** (D.Sc) works as a full professor Tampere University focusing on Production Systems and Technologies.

She obtained her doctorate in 2010. She coordinates research and development projects focusing on following topics on Manufacturing ICT, Industrial robotics, human–robot collaboration, sustainable manufacturing, and education research in technology rich environments.

Aside of the research her interests have been in public private partnerships, financial instruments for SMEs and collaboration in ECfunded projects and schemes.

She is a member of Finnish Robotics Association, EFFRA, Manufacture, Vanguard Initiative and euRobotics, and actively contributes to Visions and Strategic Research and Innovation roadmaps.

She has published over 100 research papers, articles and/or book chapters.



**Joni-Kristian Kämäräinen** is an associate professor at Computing Sciences department at Tampere University where he is part of the Vision Group. He did his postdoc in Center of Vision, Speech and Signal Processing, University of Surrey, with Josef Kittler. After his postdoc he spent five years in the Faculty of the LUT School of Engineering Science, LUT University, after which he was selected to the current tenure-track position in 2012. His research interests are computer vision, image processing, machine learning and pattern recognition. He is also involved in many applied research projects

where he collaborates with the local industry.