3rd HAND
FP7-ICT-2013-10-610878
1 October 2013 (48months)

# D2.1:
# Scientific Report on Perception for Scenario 1: Objects

Ö. Erkent, D. Shukla, S. Stabinger, J. Piater

*UIBK*
⟨Justus.Piater@uibk.ac.at⟩

| | |
|---|---|
| *Due date of deliverable:* | Y12 |
| *Actual submission date:* | M12 |
| *Lead Partner:* | UIBK |
| *Partners:* | UIBK, TUDa |
| *Revision:* | draft |
| *Dissemination level:* | PU |

This work deals with the perception of objects for the Scenario 1. We have proposed a new probabilistic, appearance-based method which integrates diverse feature types, including edge orientations, depth and color. We have used the approach in a real scenario for detecting objects and estimating their poses. The estimation errors have been compared with other state-of-the-art methods and it has been observed that the probabilistic, appearance-based method has a better detection rate for textureless objects with flat surfaces like wooden boards. We have also acquired the object models required for the first year scenario. We have used the robot arms to obtain view samples of the object from controlled directions. We also work on making a pose estimation about the small parts, including screws and screw-driver bits for robot manipulation.

## Executive Summary

This report presents the first year progress related to developing the perceptual capabilities for detection and pose estimation of objects, including tools and parts. We have acquired the object models by using the robot arms to move the objects of interest with the robotic system based in UIBK which has two seven-DOF lightweight arms. After the motion segmentation, these objects have been learned without explicit user cooperation. The robot rotated the object to obtain view samples from controlled directions. We have introduced the probabilistic, appearance-based pose estimation (PAPE) method [ESS+15] which integrates diverse feature types, including edge orientations, depth and color. The probabilistic, appearance-based models can be used to recognize large flat textureless similar parts such as wooden boards. The method has been evaluated by using the objects which will be used for the 3rdHand Project, including the wooden parts of a toolbox and the plastic textureless parts of a chair. It has been observed that even in cluttered environments, the parts can be recognized. Furthermore, we have investigated the accuracy of the pose estimation by using the ground truths obtained from the opto-tracker system which is available at TU-Darmstadt. The pose estimation errors reveal that the estimations can be used to grasp, manipulate and monitor the status of the object. We are working on the integration between appearance based pose estimation and the interaction primitives. We are also working on pose estimation and detection of small parts [SEP15]. Since today's sensing technology places lower limits on the size of objects to be detected, we are going to use a camera dedicated to detecting the small parts manipulated by the robot.

## Role of Perception for Scenario 1: Objects in 3rdHand

Our role is to provide the pose of the objects that will be grasped/manipulated by the robot. We model the objects by using a probabilistic, appearance-based method since they cannot be characterized by their shape or texture. We learn the object models by observing them from controlled viewpoints. Depth information provides new solutions to the pose estimation of textureless objects since it can remove some of the ambiguities; therefore we have used RGB-D sensors which integrate both conventional and depth cameras. Conventional cameras are also installed in the workspace to provide a higher-resolution image of the workspace. We estimate the poses of the objects on demand and inform the robot by using different feature types, including edge orientations, depth and color. For details, please refer to [ESS+15]. We are currently working learning an object when it is presented by a human worker to the robot and estimating the poses of small parts

including screws [SEP15].

## Contribution to the 3rdHand scenario

We have developed a probabilistic, appearance-based pose estimation (PAPE) method [ESS+15] which integrates diverse feature types, including edge orientations, depth and color. We have also recorded a dataset of parts that will be used in the 3rdHand Scenario 1. We evaluated the pose estimation method with the learned models in the test scene by using the pose estimation error. The results reveal that the pose estimation method can be used for grasping / manipulation tasks. We are also working on detecting the poses of the screws and screw-driver-bits in the end effector of the robot, since they are small parts with surfaces that are not possible to detect with cost-effective sensors (e.g. Kinect), we are developing an appearance-based method which uses Hu-moments to precisely estimate the pose of the small parts.

# 1 Tasks, objectives, results

### 1.1 Planned work

In the Work Package 2 (WP2), (Perception for Cooperative Manipulation), the main concern was to detect the objects and estimate their poses. To be able to estimate the poses of the objects, their models should be obtained on the fly. In a typical scenario, a human worker can pick up and move the object of interest in order to show it to the robot from diverse viewpoints. Additionally, the robot can learn the object by turning it on its manipulator to obtain views from different viewpoints. Since large, flat parts such as wooden boards do not have a distinctive texture or surface, appearance-based models were mentioned for recognizing. For distinguishing smaller parts, (e.g., screws, screw-driver bits) since today's sensing technology has lower limits for distinguishing such small parts, we found an alternative solution to using stationary cameras.

### 1.2 Actual work performed

We have been able to detect the objects for the Scenario 1 and estimate their poses. We have used an appearance-based model to distinguish large flat textureless parts such as wooden boards. We have learned the object models by using the robot arms to obtain view samples from controlled viewpoints. We tested the proposed approach with the object parts to be used in the first year of the project.

We have evaluated the accuracy of the pose estimation method by using the pose estimation of an opto-tracker as the ground-truth. The results re-

vealed that the pose estimation results may be used for grasping/manipulation tasks.

We are using a dedicated camera to detect the small parts and estimate their poses when they are hold by the robot manipulator. Since it is difficult to detect the screws or screw-driver bits with a camera at a distance, we are using cameras that are placed at around $30cm$ to the robot hand holding the screw and making a pose estimation based on an appearance-based method and increasing the precision of the estimation by using a newly proposed method which uses Hu-moments.

We are going to include the capability of learning the object's appearance model shown by the human worker. In principal, this model will be similar to the one obtained by the robot arms. Since it has been shown that the appearance-based model can be used for pose estimation, the usage of the appearance model shown by the human worker can also be used for estimating the pose.

### 1.3 Relation to the state-of-the-art

We have compared our approach against other state-of-the-art pose estimation methods which use a CAD model of the object for training, also we used an appearance-based model which does not use depth or color features. The results showed that the correct detection rates of the flat textureless objects are higher for our method even in cluttered environments. The details of the comparison between our proposed approach and the other approaches can be found in [ESS+15].

## References

[ESS+15] Ozgur Erkent, Dadhichi Shukla, Sebastian Stabinger, Rudolf Lioutikov, and Justus Piater. Probabilistic, appearance-based object detection and pose estimation of textureless objects for robot manipulation. In *Submitted to: Proceedings of 2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[SEP15] Dadhichi Shukla, Ozgur Erkent, and Justus Piater. Detection and pose estimation of screws and screw-driver bits for robot manipulation. In *in preparation*, 2015.

## 2 Annexes

- Probabilistic, Appearance-Based Object Detection and Pose Estimation of Textureless Objects for Robot Manipulation

Erkent, Ö.; Shukla, D.; Stabinger, S.; Lioutikov, R.; Piater J.; Submitted to: 2015 IEEE International Conference on Robotics and Automation (ICRA)

This paper proposes a probabilistic, appearance-based pose estimation (PAPE) method to detect and estimate the poses of textureless objects in cluttered environments.

Abstract:

We propose a probabilistic, appearance-based pose estimation (PAPE) method to detect and estimate the poses of textureless objects in cluttered environments. Our probabilistic, appearance-based model can integrate diverse feature types, including edge orientations, depth and color. We evaluate our approach in a real environment and compare it against other state-of-the-art methods on a dataset of textureless objects. The results show that our method performs better at finding textureless objects. Finally, we demonstrate the capabilities of our system in grasping/manipulation tasks.

- Detection and Pose Estimation of Screws and Screw-Driver Bits for Robot Manipulation

  Shukla, D.; Erkent, Ö.; Piater, J.; In Preparation

  Abstract:

  Object recognition and pose estimation of small parts, including screws and screw driver bits, is a challenging problem in computer vision. However, a precision estimation of the poses is necessary for robot manipulation. In this paper, we propose detection and pose estimation of the minuscule objects for handing over the objects to humans, putting a screw in the hole and fitting the screw with an automatic screw driver. We propose an approach which finds the tips of the small parts by using image moment invariants after applying a probabilistic, appearance-based object detection algorithm. In the evaluation we intend to show the approximate precision and detection rate of the proposed algorithm in a real scenario.

# Probabilistic, Appearance-Based Object Detection and Pose Estimation of Textureless Objects for Robot Manipulation

Özgür Erkent[1], Dadhichi Shukla[1], Sebastian Stabinger[1], Rudolf Lioutikov[2], Justus Piater[1]

***Abstract*— We propose a probabilistic, appearance-based pose estimation (PAPE) method to detect and estimate the poses of textureless objects in cluttered environments. Our probabilistic, appearance-based model can integrate diverse feature types, including edge orientations, depth and color. We evaluate our approach in a real environment and compare it against other state-of-the-art methods on a dataset of textureless objects. The results show that our method performs better at finding textureless objects. Finally, we demonstrate the capabilities of our system in grasping/manipulation tasks.**

## I. Introduction

Recognition and pose estimation of a target object is a necessary task in robotic perception. 6-DOF pose estimation of textureless objects in a cluttered scene is still a challenging problem in robot vision, due to occlusions and variations in object appearance as a result of different viewpoints. When the object has a distinct texture, stable keypoint descriptors can be used [1], [2], but these are not suitable for textureless objects. Integration of 3D information provide new solutions to the pose estimation of textureless objects since it can remove some of the ambiguities by using depth information about the object. Emergence of RGB-D sensors with low costs provide efficient opportunities.

In [3], a method based on a probabilistic model of appearance is suggested to estimate the poses of the objects. A technique to recognize objects in 2D images is introduced which is applicable to low-level, dense and/or non-descriptive image features. In this paper, we propose a novel joint object recognition and pose estimation method based on a probabilistic model of appearance [3] which can also be used together with RGB-D images. Our probabilistic appearance-based pose estimation (PAPE) method can be used with both textured and textureless objects in cluttered scenes. The contribution of the paper is twofold: first, we introduce depth and color information into the probabilistic appearance model of objects. Secondly, a confidence rate is introduced on the set of hypotheses. This confidence rate behaves as a hypothesis verification step. We evaluate our method by estimating the poses of the textureless objects in cluttered and uncluttered scenes with known ground truth. We show that our method works with a better accuracy than existing approaches even in cluttered scenes for textureless objects.

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 (Specific Programme Cooperation, Theme 3, Information and Communication Technologies) under grant agreement no. 610878, 3rd HAND.

[1] Özgür Erkent, Dadhichi Shukla, Sebastian Stabinger and Justus Piater are with the Intelligent and Interactive Systems Lab, Institute of Computer Science, University of Innsbruck, Austria. (ozgur.erkent@uibk.ac.at)

[2] Rudolf Lioutikov is with the Intelligent Autonomous System Lab, Technische Universitaet Darmstadt, Germany.

The paper is organized as follows: In Section I-A, related work is reviewed. In Section II, our probabilistic model of appearance is explained in detail, and in Section III, the confidence rate is described. The proposed algorithm is evaluated in Section IV, and Section V concludes the paper with a brief summary.

### A. Related Work

Robotic interaction, manipulation and grasping require a precise estimation of the pose. Some approaches use 2D local keypoint descriptors which are obtained from different viewpoints of the object or the 3D model of the object. For example, in the MOPED framework [4], SIFT [1] descriptors are used. Although this framework is reported to provide a fast and accurate object match, it requires textured objects. Some of the recent studies including [5], [6] can be inserted into this category. Studies in this category cannot be used for pose estimation of textureless objects.

There are also some studies which detect and track the textureless objects [7], however since they make a coarser estimation of the object pose, here we only consider studies which claim that they have a sufficient precision to be used with robot tasks including grasping and manipulation. Precise pose estimation is generally handled by recognition methods that use a 3D model of the object. In some of these approaches, explicit 3D models of the object are used [8], however the modeling of differences in the appearance is limited by the shape [9]. [10] uses an efficient RANSAC-like sampling strategy to establish a correspondence between the scene and the model; however this work requires a robust local descriptor like SHOT descriptors [11]. For objects without distinctive depth features from the features in the scene, it can be difficult to find a correct match. The OUR-CVFH descriptor is based on a global pipeline that uses the histograms in which the distributions of surface normals are important [12]. Therefore, for surfaces in which the distribution of normals is not distinctive, it can be difficult for the approach to find a match.

Recently, there are also studies which consider the integration of different modalities for recognition of textureless objects. For example in [13], for each point, a corresponding color information is obtained and integrated into recognition. Point pair features are used as features, and color is used for pruning hypotheses. Although the method is capable of recognizing daily objects in a cluttered environment, its computation is expensive since all of the point pairs are

considered. In another study, the integration of multiple features is given in a more generalized framework [3]. In this multiview probabilistic model of appearance, edge orientations and coarse-scale gradients are learned from several training samples. Since edges are selected as one of the features in this framework, the number of features to be compared decreases significantly with respect to the methods where the dense surface normals are compared. Also, since this approach uses a continuous pose estimation strategy, it is capable of making a precision pose estimate with a reduced search space.

Our approach adopts the idea of a probabilistic appearance model [3] and integrates the depth and color information without increasing the computation time. We also introduce a novel confidence rate on the possible set of results, and make a final decision by using this confidence rate. We also modify the continuous pose estimation approach by using a densely-sampled viewpoint database. A general flow of our pipeline can be seen in Fig. 1.

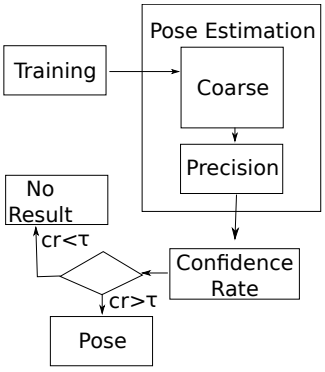



Fig. 1: General pipeline of our algorithm.

## II. Probabilistic Model of Appearance

In this Section we briefly explain how we turn a set of image features into a distribution of features. These distributions, from multiple images from different viewpoints, build our multiview object model. By using these distributions, we describe how to recognize the learned object in the test image. Then we explain the coarse-to-precision search strategy and finally we describe the voting table which gives a score on our hypothesis about the pose estimation.

### A. Probability Distributions

Let each type of feature index be denoted by: $f = 1, \ldots, F$. These features can be edge points, depth values obtained from the RGB-D sensor, or the hue value of the HSV color space. Then, each feature can be defined by its position in the image plane $p_x \in \mathbb{R}^2$ and its appearance property $a_x$; where $a_x = [a_x^o, a_x^d, a_x^c]'$ is consisting of depth, orientation and color with corresponding positions $p_x = [p_x^d, p_x^o, p_x^c]'$.

For the edges, the local orientation is used ($a_x^o \in S_1^+ = [0, \pi]$); for depth, the depth value is used ($a_x^d \in \mathbb{R}^+$ in meters) and for color, hue value from HSV color space is used ($a_x^c \in [0, 1]$). Then, a test image contains multiple feature types, i.e., $\mathcal{I}_{test} = \bigcup_f^F \mathcal{I}_{test}^f$ where each feature type contains multiple features, that is, $\mathcal{I}_{test}^f = \bigcup_i^{\|f\|} x_i^f$, and each feature consists of an appearance and a position $x_i^f =< a_{x_i}^f, p_{x_i}^f >$; $\mathcal{A}^f = x_i^f, \forall i = 1...\|f\|$.

After the features for the test image have been extracted, the distribution of feature $f$ can be defined as defined in [3]:

$$\phi_{test}^f(x^f) = \sum_{x_i^f \in \mathcal{I}_{test}^f} \mathrm{w}(x_i^f)\mathcal{N}(p_{x_i^f}; \mathrm{p}_x^f, \sigma_p)\mathrm{K}^f(a_{x_i}; \mathrm{a}_x), \quad (1)$$

where $\mathcal{N}$ is Gaussian kernel for feature positions and $\mathrm{K}^f$ is an appearance kernel. $\mathrm{w}(x_i^f)$ is the weight of feature $x_i$ and set, i.e., $\mathrm{w}(x_i) = 1/\|\mathcal{I}_{test}^f\|^c \ \forall x_i \in \mathcal{I}_{test}^f$, and $c \in \mathbb{R}$ is a priori constant to control the effect of the size of the number of features on the weight. If $c = 1$, then an image with a small number of features will have larger weight values w.r.t. an image with large number of features.

In a similar manner, a distribution $\phi_{\mathcal{I}_{train}}^f$ can be obtained for the features of a set of training images, containing one image for each viewpoint $v \in S^2$.

### B. Recognition with Probability Distributions

Let in-plane transformations $w^*$ (a translation, rotation and scale) and viewpoint $v^* \in S^2$ denote the optimal set of the learned viewpoint in the test image. Then $(v^*, w^*)$ represents the 6 DoF of the object.

The similarity between the test and the training images can be obtained by the cross-correlations of the images as mentioned in [3]:

$$\left(\phi_{test}^f \star \phi_{train_v}^f\right)(w) = \int_{\mathcal{A}^f} \phi_{test}^f \phi_{train_v}^f(\mathrm{transform}_w(x))\mathrm{d}x \quad (2)$$

To increase the efficiency of the system, samples are drawn from the test and training samples by using Monte Carlo integration [14]. $L$ particles from test distribution and $L'$ particles are drawn from the training distribution. Then the cross-correlation for one feature type $f$ becomes

$$\left(\phi_{test}^f \star \phi_{train_w}^f\right)(w) \approx \frac{1}{LL'} \sum_i^L \sum_j^{L'} \mathrm{w}(x_j)\mathcal{N} (p_{x_i}; \mathrm{transform}_w(p_{x_j}), \sigma_{pos})\mathrm{K}^f(a_{x_i}; a_{x_j}). \quad (3)$$

This can be further defined over all the feature types:

$$\mathrm{similarity}_{\mathrm{test,train}_v}(w) = \prod_f (\phi_{\mathrm{test}}^f \star \phi_{\mathrm{train}_v}^f)(w) \quad (4)$$

From Eqs. 3 and 4, the likelihood of observing the trained object from the viewpoint $v$ with the transformation $w$ can be obtained. Then, the pose estimation problem becomes that of finding the maxima of Eq. 4:

$$(v^*, w^*) = \arg\max_{v,w} (\mathrm{similarity}_{\mathrm{test,train}_v}(w)) \quad (5)$$

### C. Coarse-to-Fine Search

In this study, we apply a coarse-to-fine search strategy to decrease the size of the search space. First, a coarse pose estimation is performed on the uniformly selected viewpoints. Let the distance between the uniformly selected viewpoints be denoted by $\theta_v$. A set of hypotheses $\mathcal{H}_c = (v_c, w_c)$ is obtained which give the viewpoints $v_c$ and transformations $w_c$ in the test image. In the close neighborhood $v_c + \delta v_c, w_c + \delta w_c$ of each hypothesis, a new search is made which we call a *precision search*. It should be noted that although the neighborhood can be selected to be any arbitrary value, $\delta v_c < \theta_v$ is a well-motivated choice. Then, for each coarse estimation $\mathcal{H}_c$, a precise estimate $\mathcal{H}_p$ is obtained. The selection procedure of the best estimate is explained in Section III.

### D. Feature Types: Edge Orientations, Depth and Color

*a) Edge Orientations:* We use an intensity-based Canny edge detector [15]. Each edge point feature has an appearance attribute of the local orientation of the edge at a given position, $\mathcal{A}^o = \mathbb{R}^2 \times S_1^+$. The kernel uses a von Mises distribution on the half circle, which is defined as: $K^o(a^o_{x_1}, a^o_{x_2}) = C_1 \mathrm{e}^{\kappa \cos(a^o_{x_1} - a^o_{x_2})}$. Our distance measure can be said to be a general form of the directed chamfer distance [16].

*b) Depth Values:* The depth value obtained from the RGB-D sensor is taken as the depth feature. Each depth feature has only one depth value as an appearance attribute, $\mathcal{A}^d = \mathbb{R}^+$. The kernel can be defined as $K^d(a^d_{x_1}, a^d_{x_2}) = C_2 \mathrm{e}^{(a^d_{x_1} - a^o_{x_2})^2}$, $\forall a^d_{x_1}, a^d_{x_1} \in \mathbb{R}^+$. For the locations where it is not possible to find a depth value, due to transparency or shiny surfaces, the kernel is set to a predefined constant $K^d = C_3$. Then, the method tries to use other features (e.g. edges and color) if the depth is not available. Since the values of features are combined with a product, $C_3$ sets the importance of other features in the absence of depth.

*c) Color Values:* The color feature is selected from the hue component of the HSV color space. Then the kernel can be defined as $K^c(a^c_{x_1}, a^c_{x_2}) = C_4 \mathrm{e}^{\kappa \cos(a^c_{x_1} - a^c_{x_2})}$, $\forall a^c_{x_1}, a^c_{x_2} \in [0, 1]$, $K^c = C_5$.

### E. Voting Table

Finding the maximum similarity between the training features and the test features will result in a pose estimate. For efficiency, since all of the feature values are integrated via a product in the similarity measure, first the sparse edge points are found. Then the depth and color features around these sparse edge points are computed. It would also be possible to directly use depth and color features if the edge features are not detectable.

If you consider a voting table $\mathcal{H}$ with discrete votes at image locations $p_{v_j}$ with weights $w_{v_j}$, and if you convolve this table with an isotropic Gaussian kernel with $\sigma_{pos}$, then you can obtain a score at each location $l$

$$\mathcal{H}(l) = \sum_j w_{v_j} \mathcal{N}(l; p_{v_j}, \sigma_{pos}). \tag{6}$$

## III. Confidence Rate

The score obtained from the voting table $\mathcal{H}(l)$ computes the vote for the viewpoint with the downsampled edge orientation features based on Monte Carlo integration [14]. As it was explained in Section II, this is necessary to increase the efficiency of the algorithm. However, this downsampling can affect the accuracy of the result. To select the best possible match, a measure is necessary to decide on the best hypothesis by using various available cues including edge, intensity, depth and surface normals on the whole object. After this measure is computed, the one with the highest score is selected. The confidence is selected as the following logistic function for any hypothesis of pose estimate $j$:

$$p^m(j) = \prod_{i \in \mathcal{C}} \frac{1}{1 + \exp\left(\gamma_i^m (c_i^m - y_i^m(j))\right)} \tag{7}$$

Here, $c_i^m$ and $\gamma_i^m$ are a priori constants for each object model $m$ and cue type $i$. Although, they are determined by using the object model images manually, they could also be computed automatically from these images. $\mathcal{C}$ is the set of cues used to test the hypothesis, and $y_i^m(j)$ is the corresponding distance measure for hypothesis $j$ for cue $i$. The hypothesis with the highest confidence is selected as the candidate pose:

$$p^{m*} = \arg\max_j p^m(j) \tag{8}$$

If the highest confidence is lower than a threshold, $p^{m*} \leq \tau_c$, then the method decides that the object is not in the scene. The distance measure $y_i^m(j)$ is computed by finding the difference between the features of the corresponding viewpoint of the training image and the features in the vicinity area of the transformed test image for the corresponding hypothesis:

$$y_f^m(j) = \frac{(\mathcal{I}_{train^f_{w_j}} - \mathcal{I}_{train^f_{v_j}})^2}{\|\mathcal{I}_{train^f_{v_j}}\|} \tag{9}$$

where $f$ is the feature type of hypothesis cue. The difference is normalized with the size of the corresponding viewpoint of the training image. The following cue features are used:

- Edge: Intensity-based Canny edge detector [15] is used to extract the edge features.
- Intensity: The intensity values are taken as the features. It should be noted that the camera in the test scene is color corrected at the beginning of the experiments by using a gray card.
- Depth: The depth information from the RGB-D sensor is taken as the feature after the corresponding transformations are applied on the corresponding area. (including in-plane rotations and scale)
- Surface Normals: The surface normal at each pixel $\nabla\mathcal{I}$ is taken as the feature.

It should be noted that these features could also be used in Sec. II-D, however due to efficiency, they are used only to validate the final hypothesis.

## IV. EXPERIMENTS

In this section we compare our results against [3], [10], and [12]. The implementation of [3] is available online[1]; [10] and [12] are included in the Point Cloud Library (PCL) [17]. This allows for a fair comparison of the algorithms. The test scenarios consist of various object parts which have to be assembled by a human with the help of a robot to demonstrate human-robot interaction. We compare the pose estimation methods in two ways, (1) the number of objects detected per test scene and (2) the object pose error with respect to the ground truth data. The ground truth data was acquired using a Natural Point OptiTrack system with eleven "Flex13" cameras to track 4 reflective markers, shown in Fig. 2 (left). OptiTrackers are placed manually at approximately the center of the object. The test scene can be seen in Fig. 2 (right). The two Kinects that we use to capture the images can be seen (1) positioned above the robot arms and (2) suspended from the ceiling which is orthogonal to the first Kinect. We use SHOT [11] descriptors for [10] and [12] with a descriptor radius of $0.04$, and $5$ iterations for the ICP (Iterative Closest Point) method [18]. For [3] and the proposed method, in-plane rotations are in the range of $\alpha \in [0, \pi/2]$ and scale is in the range of $sc \in [0.5, 1.1]$. Additionally, we use a confidence rate of $0.5$ for the proposed method in the first set of experiments.

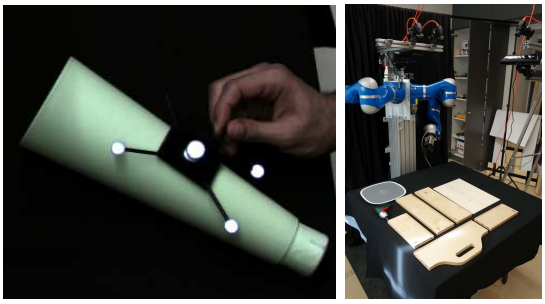


Fig. 2: Left: 4 reflective markers of the OptiTracker placed on an object. Right: The camera setup.

The point cloud methods [10] and [12] initially removes planar surfaces (e.g. floor, wall, etc.). Therefore these methods tend to have problems in detecting planar objects placed on a surface (e.g. a table) as shown in Fig. 3. We therefore limit our comparison to test scenes like the ones shown in Figs. 5a and 6a.

The training data is captured using a Kinect and 2 KUKA Light weight robot arms. The used setup is shown in Fig. 4. The left arm moves the camera along an arc around the object and the right arm changes the azimuth (i.e. rotation) of the object. The training images are captured for azimuths in the range of $\theta \in [0, 2\pi]$ and elevations in the range of $\psi \in [0, \pi/2]$. After segmentation the objects are used, along with ground truth poses, as training data.

[1]http://www.montefiore.ulg.ac.be/ dteney/code.htm

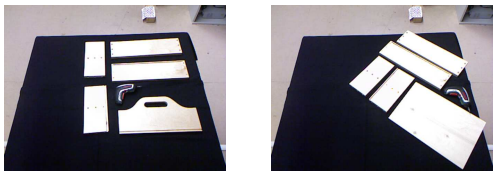


Fig. 3: Object placed flat on the surface of table.

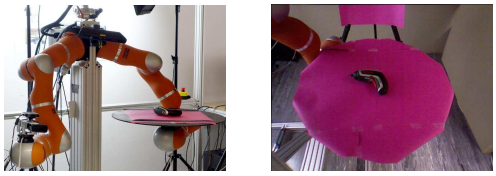


Fig. 4: Robot training.

### A. Object recognition

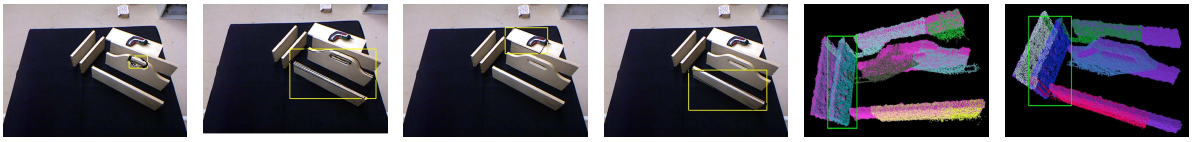
As with conventional object recognition algorithms, an object is said to be detected if the estimated bounding box overlaps the ground truth by more than $50\%$ [19]. This criterion is not sufficient for robot manipulation. For our purposes we consider an object to be detected if there is an overlap of at least $90\%$. We compare our results in a practical setup as shown in Figs. 5a and 6a. The setup consists of three visually similar tool box parts (Box part1, Box part2, Box part3) and an automatic drill. A comparison of the investigated methods can be seen in Figs. 5 and 6. Due to space limitation we can only present some of the detections.

As can be seen, the proposed method is robust when detecting highly similar objects, while [3] tends to get them confused. The point cloud methods [10] and [12] can detect Box part2 (*both instances*; Figs. 5j, 5k, 6j, 6k) but also generates false positives for other Box parts, despite being fully observable. As previously discussed, [10] and [12] tend to have problems with detecting objects placed flat on the table plane due to planar segmentation.

We perform a quantitative comparison for different test scenes where the objects are placed flat on the table, occluded by other objects, or seen only partially by the camera. The different tested configurations can be seen in Table I. Results for those configurations are presented in Table II. We measure the performance of the algorithms by comparing the number of correctly detected objects (true positives, TP) and the number of incorrectly detected objects (false positive, FP) to the total number of objects detected (DO). Our experiments suggest that the proposed method (PAPE) performs better than state-of-the-art methods for our test cases. The test scenes are combinations of objects placed flat and askew on the table plane, which can be encountered in a real robot-human interaction scenario.

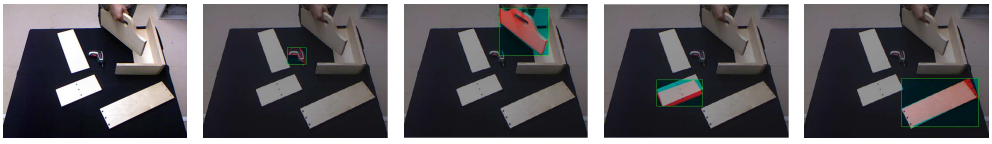
The superiority in the performance of PAPE for planar objects can be attributed to its appearance-based method which can integrate different features like edge orientation, depth and color. Additionally, the low number of false positives in object detection can be associated with the confidence rate.

(a) Test scene (b) Automatic Drill (c) Box part1 (d) Box part2 (e) Box part3

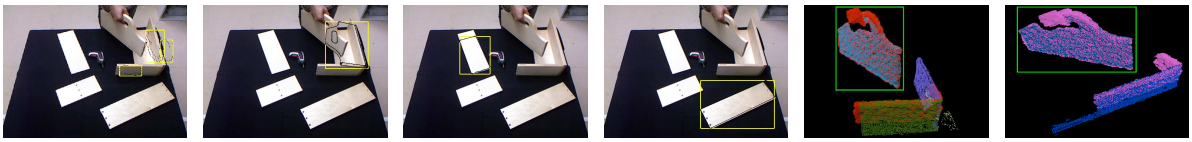


(f) Automatic Drill (g) Box part1 (h) Box part2 (i) Box part3 (j) All objects (k) All objects

Fig. 5: Object detection test scene 1: (a) Test scene, (b-e) PAPE, (f-i) Objects detected by [3], (j) True positive by [10] and (k) True positive by [12].



(a) Test scene (b) Automatic Drill (c) Box part1 (d) Box part2 (e) Box part3



(f) Automatic Drill (g) Box part1 (h) Box part2 (i) Box part3 (j) All objects (k) All objects

Fig. 6: Object detection test scene 2: (a) Test scene, (b-e) PAPE, (f-i) Objects detected by by [3], (j) True positive by [10] and (k) True positive by [12].

| Test Scenes | 1 | 2 | 3 |
|---|---|---|---|
| Nr. of Objects | 7 | 6 | 6 |
| Nr. of Planar Objects | 1 | 3 | 3 |
| Nr. of Occluded Objects | 1 | 1 | 2 |

TABLE I: Test scene configurations.

| | | Test scenes | | |
|---|---|---|---|---|
| | | 1 | 2 | 3 |
| Method [12] | DO | 8 | 4 | 4 |
| | TP | 2 | 2 | 0 |
| | FP | 6 | 2 | 4 |
| Method [10] | DO | 5 | 2 | 2 |
| | TP | 2 | 1 | 0 |
| | FP | 3 | 1 | 2 |
| Method [3] | DO | 7 | 6 | 6 |
| | TP | 1 | 1 | 2 |
| | FP | 6 | 5 | 4 |
| PAPE | DO | 7 | 6 | 4 |
| | TP | 7 | 5 | 4 |
| | FP | 0 | 1 | 0 |

TABLE II: Test scenes results: DO - Detected objects, TP - True positives and FP - False positives.

### *B. Evaluation on a Robot*

In the second part of the experiments, we investigate the precision of our method in terms of pose estimation error. As it was mentioned in Sec. IV-A, the RANSAC-like sampling strategy [10] and OUR-CVFH descriptor [12] tend to eliminate flat objects on the table while performing plane segmentation. Since the objects in this Section are lying flat on the table, these methods have a poor performance in detection. Therefore we investigate the pose estimation output of only our method. The confidence threshold is selected to be $\tau_c = 0.6$, the scale range is $sc \in [0.6, 0.9]$, and $c = 0.125$, which controls the the effect of the size of the number of features on the weight $w$.

In the experiments, the same scene is observed by two RGB-D cameras as shown in Fig. 7 except the scene on the extreme bottom right. The average error of the estimated pose with respect to the ground truth is shown in Table III. The left column in Table III refers to different scenes captured by two Kinects. For example, S1K1 means that the image is taken from Scene 1 with Kinect 1. There are a total of three object classes: Legs, Bottom and Back; and Legs can be observed with more than one instance in the images. We compute the mean error of estimated poses for multiple instances of Leg. It should be noted that in all of the images, all of the objects are recognized, therefore an evaluation of the number of detected objects is not repeated in this Section.

As can be observed in Table III, the precision does not change significantly with changes in viewpoint. Part of the

(a) S1K1 (b) S2K1 (c) S3K1

(d) S1K2 (e) S2K2 (f) S3K2 (g) S4K2

Fig. 7: Scenes with chair parts from two different cameras. S: Scene, K: Kinect, Top: Scenes seen by Kinect 1, Bottom: Scenes seen by Kinect 2.

remaining error is likely a result of placing the reflective markers manually on the objects as shown in Fig. 2 (left). This also indicates presence of the precision error. Therefore, if the manipulation/grasping task requires a more precise estimate of the objects, a method to register the 3-D Shape of the obtained training viewpoint $v^*$ to the transformation $w^*$ in the test scene should be applied, e.g. ICP (Iterative Closest Point) [18]. The implementation of such a method at the final stage of our approach is not in the scope of this paper.

## V. Conclusion

We proposed a novel joint object recognition and pose estimation method that can be used with RGB-D images. We showed that our method can be used with textureless objects in cluttered scenes. Its capability of detecting objects and estimating their poses is superior to the other methods we compared. This can be attributed to its probabilistic framework that integrates diverse types of features such as edge orientation, depth and color. A particular contribution is the definition of a confidence rate to select the best hypothesis of pose estimation of the object. By using the confidence rate, the number of false positives is reduced significantly, as shown in the experiments. We also showed the capacity of our system to be used for grasping/manipulation tasks by giving the mean pose estimation error for textureless objects.

In the future, we plan to develop our method for use with multiple cameras to increase its accuracy. Moreover, our focus will be on active learning of object appearances in an autonomous manner.

| | Leg | Bottom | Back |
|---|---|---|---|
| S1K1 | 0.0315 | 0.0420 | 0.0479 |
| S1K2 | 0.0075 | 0.0426 | 0.0500 |
| S2K1 | 0.0605 | 0.0253 | - |
| S2K2 | 0.0665 | 0.0262 | - |
| S3K1 | 0.0639 | 0.0650 | - |
| S3K2 | 0.0593 | 0.0434 | - |
| S4K2 | 0.0392 | 0.0320 | - |

TABLE III: Average Pose Estimation Errors (m)

## References


[1] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int. J. of Computer Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[2] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[3] D. Teney and J. Piater, "Multiview feature distributions for object detection and continuous pose estimation," *Computer Vision and Image Understanding*, vol. 125, pp. 265–282, 8 2014. [Online]. Available: https://iis.uibk.ac.at/public/papers/Teney-2014-CVIU.pdf

[4] A. Collet, M. Martinez, and S. S. Srinivasa, "The MOPED framework: Object recognition and pose estimation for manipulation," *The International Journal of Robotics Research*, vol. 30, pp. 1284–1306, 2011.

[5] Z. Teng and J. Xiao, "Surface-based General 3D Object Detection and Pose Estimation," in *2014 IEEE International Conference on Robotics and Automation*, 2014, pp. 5473–5479.

[6] K. Li and M. Meng, "Robotic Object Manipulation with Multilevel Part-based Model in RGB-D Data," in *2014 IEEE International Conference on Robotics and Automation*, 2014, pp. 3151–3156.

[7] S. Hinterstoisser, C. Cagniart, S. Ilic, P. Sturm, N. Navab, P. Fua, and V. Lepetit, "Gradient response maps for real-time detection of textureless objects." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 5, pp. 876–88, May 2012.

[8] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce, "3d object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints," *International Journal of Computer Vision*, vol. 66, no. 3, pp. 231–259, 2006.

[9] D. Hoiem, C. Rother, and J. Winn, "3d layoutcrf for multi-view object class recognition and segmentation," in *Computer Vision and Pattern Recognition, IEEE Conference on*. IEEE, 2007, pp. 1–8.

[10] C. Papazov and D. Burschka, "An efficient ransac for 3d object recognition in noisy and occluded scenes," in *Computer Vision–ACCV 2010*. Springer, 2011, pp. 135–148.

[11] F. Tombari, S. Salti, and L. Di Stefano, "Unique signatures of histograms for local surface description," in *11th European Conference on Computer Vision (ECCV)*, vol. 6313 LNCS, Hersonissos, Greece, 2010, pp. 356–369.

[12] A. Aldoma, F. Tombari, R. B. Rusu, and M. Vincze, "OUR-CVFH - Oriented, unique and repeatable clustered viewpoint feature histogram for object recognition and 6DOF pose estimation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7476 LNCS, 2012, pp. 113–122.

[13] C. Choi and H. I. Christensen, "3D pose estimation of daily objects using an RGB-D camera," in *IEEE International Conference on Intelligent Robots and Systems*, 2012, pp. 3342–3349.

[14] R. E. Caflisch, "Monte Carlo and quasi-Monte Carlo methods," *Acta Numerica*, vol. 7, pp. 1–49, 1998.

[15] J. Canny, "A computational approach to edge detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 6, pp. 679–698, 1986.

[16] M.-Y. Liu, O. Tuzel, A. Veeraraghavan, and R. Chellappa, "Fast directional chamfer matching," in *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*. IEEE, 2010, pp. 1696–1703.

[17] A. Aldoma, Z.-C. Marton, F. Tombari, W. Wohlkinger, C. Potthast, B. Zeisl, R. B. Rusu, S. Gedikli, and M. Vincze, "Point cloud library," *IEEE Robotics & Automation Magazine*, vol. 1070, no. 9932/12, 2012.

[18] P. Besl and N. McKay, "A Method for Registration of 3-D Shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, pp. 239–256, 1992.

[19] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge–a retrospective," *Int J Computer Vision*, 2014.

# Detection and Pose Estimation of Screws and Screw-Driver Bits for Robot Manipulation

Dadhichi Shukla, Özgür Erkent, and Justus Piater

Intelligent and Interactive Systems, University Of Innsbruck, Institute of Computer Science
Technikerstr. 21a, 6020 Innsbruck, Austria,
dadhichi.shukla@uibk.ac.at,

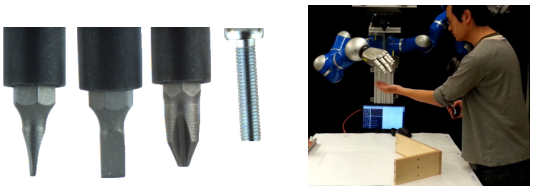
**Abstract.** Object recognition and pose estimation of small parts, including screws and screw driver bits, is a challenging problem in computer vision. However, a precision estimation of the poses is necessary for robot manipulation. In this paper, we propose detection and pose estimation of the minuscule objects for handing over the objects to humans, putting a screw in the hole and fitting the screw with an automatic screw driver. We propose an approach which finds the tips of the small parts by using image moment invariants after applying a probabilistic, appearance-based object detection algorithm. In the evaluation we intend to show the approximate precision and detection rate of the proposed algorithm in a real scenario.

**Keywords:** Pose estimation, Tool tip detection

## 1 Introduction and Related Work

Object recognition and pose estimation has been a challenging problem in computer vision over the years. Tasks such as pose estimation, grasping, navigating, and learning the structure of new objects, are well enhanced in human. Performing the same by a robot with vision system further enhances the challenge. For a robot to work in a given environment, the vision system should be capable of estimating the pose of the object to be manipulated. SIFT [1] and SURF [2] still seem to be the most appealing descriptors used for object detection. They have been widely employed in robot vision due to their robustness and relatively fast nature, which crucial for on-line applications. More recently, the emergence of low-cost RGB-D sensors has added flexibility to robot vision applications.

In this paper, we address detection and pose estimation of minuscule objects like screws and various screw driver bits, as shown in Fig. 1a for robot manipulation. The test scenarios consist of various parts of an object which have to be assembled by a human with the help of a robot to demonstrate human-robot interaction as shown in Fig. 1b. Some of the tasks performed by robot will be to hand over the objects to humans, put a screw in the hole and fit the screw with an automatic screw driver. Such tasks require highly accurate pose estimates of object parts as well as tiny objects. A common convention in object detection is that an object is said to be detected accurately if the estimated bounding box overlaps the ground truth by at least 50% [3]. There have been ample methodologies demonstrating impressive results based on such a criterion.



(a) Minuscule objects (b) Robot handing over screw

Fig. 1: Human robot interaction.

However, is not sufficient for robot manipulation task. A pixel-accurate object detection, though difficult, would be an ideal fit to perform the aforementioned robot tasks. The popular SIFT [1] and SURF [2] methods can be used to perform pixel-accurate object detection, but they work only with textured objects. Some of the recent methodologies [4, 5] can be included in this category. Such methods cannot be used for pose estimation of textureless objects.

The RGB-D sensors like Kinect work in the range greater than 600*mm*. And to detect minuscule objects at distances above 300*mm* will be an extremely difficult challenge. At such high distance they appear to be blurry in RGB images and have no depth data, thus constituting RGB-D sensors unworkable. Moreover, reflective objects like screws and relatively dark objects like screw driver tips will not provide any depth data. The vision systems in industrial applications to detect minuscule objects employ *telecentric lenses*, which are comparatively very expensive. Visual servoing methods [6] along with known 3D models of the objects and pose of the robot manipulator might achieve the task of inserting a screw into its hole. Still, this requires the detection of the screw and the screw driver tip, which are hard tasks in a real robot scenario.

Pose estimation of the object in a single 2D image has been a challenging problem for vision systems over the years. The object detection and pose estimation framework proposed by [7], used in this work, is purely probabilistic, appearance-based and naturally accommodates variability in scale, shape as well as appearance of objects. The method can accommodate different types of image features, but in this work, very basic edge features along with their tangent orientation are chosen. The proposed method comprises of two steps: (1) Object detection and approximate pose estimation with [7], and (2) accurately estimate pose and tip of the minuscule objects.

## 2 Probabilistic Appearance-based Model

The probabilistic appearance-based model proposed in [7] performs object recognition and pose estimation in 2D images, and is applicable to various types of image features. In this framework, it is not easy to match the edge points in training and test views. In this section we briefly explain the probabilistic approach to detect objects.

Fig. 2: Center point feature.

### 2.1 Learning object models: Pose-Appearance space

We create training data from real images (see Fig. 3a) to create a pose-appearance space for each object as detailed in [7]. For each image in training data, we extract edge points with their tangent orientation, defined on $\mathbb{R}^2 \times S_1^+$, accounting for the position in the image and orientation. This space is defined as the appearance space $A$. Additionally, we associate a center point feature $c_p$ with each training image which aids in separation between tool tip and tip holder as shown in Fig. 2. This is not necessary for screw. Edge features $x$ of each image are then associated with the respective pose $w$ to obtain a set of *pose/appearance* pairs. Considering the whole training set $T$, these pairs of all the images are used to define the continuous probability distribution $\psi$ as:

$$\psi(w,x) = \frac{1}{M} \sum_{(w_i,x_i)\in T} K_1(w,w_i) K_2(x,x_i), \tag{1}$$

where $w \in SE(3)$ and $x \in A$. The use of kernels $K_1$ and $K_2$ on the training data can be seen here as a smoothing over the available training edge points, effectively yielding a continuous distribution and allowing to interpolate, to some extent, the value $\psi$ over regions not covered by the training data [8].

### 2.2 Pose Inference

The test scene is a new 2D image of the scene, it undergoes a similar procedure as that of the training images. The same type of features (edges) are extracted and are stored as the *observations* $O = \{x\}_{i=1}^{N}$, where $x_i \in A$ and $N$ is number of test scenes. For a given set of test scenes (or a single test scene), the continuous probability density $\phi$ on the appearance space $A$ is defined as:

$$\phi(x) = \frac{1}{N} \sum_{x_i \in O} K_2(x,x_i). \tag{2}$$

The estimated pose of the object in a test scene is modeled as random variable $W \in SE(3)$, the distribution of which is the likelihood function given by:

$$p(w) = \int_A \psi(w,x)\, \phi(x) \mathrm{d}x. \tag{3}$$

As mentioned in [7], the above expression measures the compatibility of the training data at a pose $w$, with the distribution of features observed in the test image. It can be interpreted as the cross-correlation of the distribution $\phi$ of observation in the test image with the distribution $\psi(w,.)$ of the training points. The method uses a probabilistic voting scheme on the 6D pose space to identify the modes and peaks of the distribution of $W$. The algorithm locally fits such a distribution on the peaks of $p(w)$, using non-linear least squares. The mean of the fitted distribution is then retained as the peak of that particular mode of the distribution i.e *score* of the pose. The estimated pose of the object is described by: azimuth & elevation angle, in-plane rotation, location of the pose with respect to the image center, change in the scale of the pose, score and the training image index. For further details on the approach, we refer readers to [7], [9]. With this section, we achieve pixel-accurate detection of the minuscule objects and we begin search for the tip of screw driver bit and its orientation in next section.

## 3 Tip Detection and Pose Estimation with Image Moments

The probabilistic object detection framework by [7] results in pixel-accurate object detection. Despite of achieving pixel-accurate detection of the minuscule objects, the non-parametric nature of the method raises difficulties to detect exact location of the tip and its orientation. To overcome this challenge, we adopt to image moment invariants as detailed in [10].

### 3.1 Image moments

The concept of image moments was introduced in [10]. Here, we briefly detail the concept of image moments and how to extract orientation of an image.

Image moments can be used to derive simple image properties like area (for binary images), centroid, or sum of grey level (for greyscale image). For an image the raw image moments $M_{ij}$ can be calculated as:

$$M_{ij} = \sum_x \sum_y x^i y^j I(x,y), \tag{4}$$

where $i, j$ are moment indexes, $x, y$ are pixel location, and $I(x,y)$ is pixel intensity. For practical use, the image is summarized with functions of a few lower order moments. The central moments for an image can be defined as:

$$\mu_{ij} = \sum_x \sum_y (x-\bar{x})^i (y-\bar{y})^j I(x,y), \tag{5}$$

where, $\bar{x} = \frac{M_{10}}{M_{00}}$ and $\bar{y} = \frac{M_{01}}{M_{00}}$ are the components of the centroid. We can derive image orientation using the second order central moments to construct a covariance matrix. The covariance matrix of the image $I(x,y)$ is given by:

$$cov[I(x,y)] = \begin{bmatrix} \mu'_{20} & \mu'_{11} \\ \mu'_{11} & \mu'_{02} \end{bmatrix}. \tag{6}$$

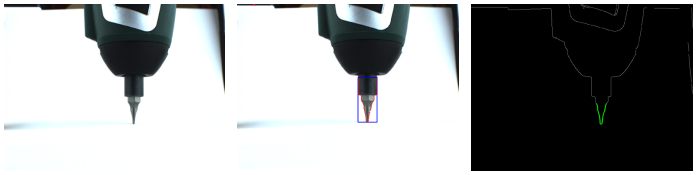
We can then compute eigenvectors of the covariance matrix to obtain major and minor axes of the image, so the orientation can thus be extracted from the angle of eigenvector with largest eigenvalue. The orientation of the eigenvector can be given as

$$\tan 2\theta = \frac{2\mu'_{11}}{\mu'_{20} - \mu'_{02}}, \tag{7}$$

as long as: $\mu'_{20} - \mu'_{02} \neq 0$. We refer readers to [10] for mathematical details.

### 3.2 Tip detection

To detect tool tip, first, we extract the edge points within the bounding box $I_{bb}$ obtained from [7]. Edge features $x_{bb}$ within the bounding box $I_{bb}$ are retained, while the others are discarded, $x_{bb} \in I_{bb}$. Further, we use center point feature $c_p$ along with in-plane rotation $\alpha$ obtained from estimated pose, to crop the tip from tip holder. We create a binary image by filling the region within those edge points. A morphological erosion on the binary image eliminates the unwanted edge points in the background, improving accuracy. We then compute orientation and major axis of the binary image using image moments. This can be seen as ellipse fitting on the binary mask. A step-by-step visualization of the procedure is shown in Fig. 3.



(a) Test scene (b) Detected minuscule object (c) Cropped edge features

(d) Binary mask (e) Ellipse fitting

Fig. 3: Step-wise visualization for tip detection.

After knowing the location of centroid, major axis, and its orientation $\theta$, the tip can be detected by using a nearest neighbour algorithm. One of the end points $ep_1, ep_2$ of the major axis is located closest to the tool tip. Since, we know the pose of the robot hand, we also know whether the tip is facing vertically *up or down* in the image plane. We use this information to select the appropriate end point $ep$ of the major axis, either

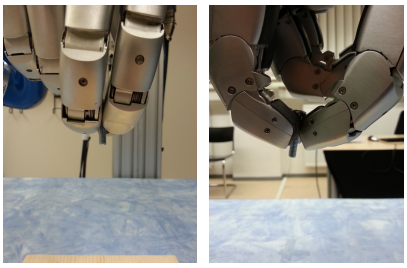
$ep_1$ or $ep_2$. The estimated tip of the object in image plane is computed by applying the nearest neighbour algorithm:

$$x_n = \underset{i}{\operatorname{argmin}} ||x_i - ep||^2, x_i \in x_{bb}, \tag{8}$$

where, $x_n$ is the nearest edge point to end point $ep$, and $x_i$ are all the edge points within the bounding box $I_{bb}$. The location of the estimated tip is given by $x_n$ and its orientation is given by $\theta$.

## 4 Experiments and Results

The experimental setup consist of two orthogonally placed cameras. Once an object is grasped by the robot, the robot moves to a known location in world frame such that plane of the robot hand is parallel to the image plane of one of the cameras. An example test scenario is shown in Fig. 4. We test the proposed approach on a diverse set of



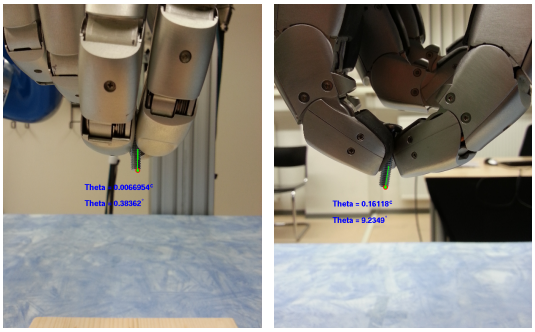
(a) View 1 (b) Corresponding orthogonal view

Fig. 4: Orthogonal view of the grasped object.

images, where a screw is held at different orientations. The results of screw tip detection are shown in Fig. 5. We test the algorithm with different screw driver bit placed at orientations. The results for tip detection and estimation of the pose of bit are shown in Fig. 6. It can be seen from the results that the proposed method is able to detect the tip accurately in the image plane.

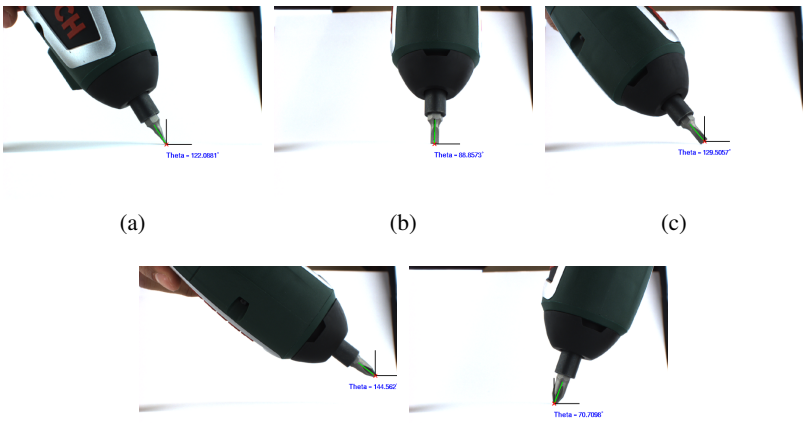
## 5 Conclusion

We propose a novel approach to detect minuscule objects like screw and screw driver bits. The proposed method finds the tip of small objects in two steps: (1) Pixel-based object detection with a probabilistic, appearance-based framework and (2) Estimate tip location and orientation by adopting to the concept of image moments. The method demonstrates accurate detection of the tip, which can be used in human-robot interactions.



(a) Screw tip in view 1 (b) Screw tip in view 2

Fig. 5: Screw tip detection in two orthogonal views: $\theta_a = 0.38362°$, $\theta_b = 9.2349°$



(a) (b) (c)

(d) (e)

Fig. 6: Screw driver bit tip detection: $\theta_a = 122.088°$, $\theta_b = 88.8573°$, $\theta_c = 129.5057°$, $\theta_d = 144.562°$, $\theta_e = 70.7098°$

The research leading to these results has received funding from the European Communitys Seventh Framework Programme FP7/2007-2013 (Specific Programme Cooperation, Theme 3, Information and Communi- cation Technologies) under grant agreement no. 610878, 3rd HAND.

## References

1. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. Int. J. of Computer Vis. **60**(2) (2004) 91–110
2. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (surf). Computer Vision and Image Understanding **110**(3) (2008) 346–359
3. Everingham, M., Eslami, S., Van Gool, L., Williams, C., Winn, J., Zisserman, A.: The pascal visual object classes challenge: A retrospective. International Journal of Computer Vision (2014) 1–39
4. Teng, Z., Xiao, J.: Surface-based General 3D Object Detection and Pose Estimation. In: 2014 IEEE International Conference on Robotics and Automation. (2014) 5473–5479
5. Li, K., Meng, M.: Robotic Object Manipulation with Multilevel Part-based Model in RGB-D Data. In: 2014 IEEE International Conference on Robotics and Automation. (2014) 3151–3156
6. Prats, M., Martinet, P., del Pobil, A.P., Lee, S.: Robotic execution of everyday tasks by means of external vision/force control. Intelligent Service Robotics **1**(3) (2008) 253–266
7. Teney, D., Piater, J.: Modeling Pose/Appearance Relations for Improved Object Localization and Pose Estimation in 2D images. In: 6th Iberian Conference on Pattern Recognition and Image Analysis. Volume 7887 of LNCS., Berlin, Heidelberg, New York, Springer (6 2013) 59–68
8. Teney, D., Piater, J.: Continuous Pose Estimation in 2D Images at Instance and Category Levels. In: Tenth Conference on Computer and Robot Vision, IEEE (5 2013) 121–127
9. Teney, D., Piater, J.: Multiview feature distributions for object detection and continuous pose estimation. Computer Vision and Image Understanding **125** (8 2014) 265–282
10. Hu, M.K.: Visual pattern recognition by moment invariants. Information Theory, IRE Transactions on **8**(2) (1962) 179–187