Model based Detection and 3D Localization of Planar Objects for Industrial Setups

Basak Sakcak, Luca Bascetta and Gianni Ferretti
Dipartimento di Elettronica Informazione e Bioingegneria, Politecnico di Milano,
Piazza L. da Vinci, 32-20133 Milano, Italy

Keywords: Model based Object Recognition, Pose Estimation, Chamfer Matching.

Abstract: In this work we present a method to detect and estimate the three-dimensional pose of planar and textureless objects placed randomly on a conveyor belt or inside a bin. The method is based on analysis of single 2D images acquired by a standard camera. The algorithm exploits a template matching method to recognize the objects. A set of pose hypotheses are then refined and, based on a gradient orientation scoring, the best object to be manipulated is selected. The method is flexible and can be used with different objects without changing parameters since it exploits a CAD model as input for template generation. We validated the method using synthetic images. An experimental setup has been also designed using a fixed standard camera to localize planar metal objects in various scenarios.

1 INTRODUCTION

Recent improvements in robotic manipulation and vision systems triggered the interest in using visual systems for manipulation of objects in industrial environments, instead of ad-hoc mechanical solutions. Such problem requires identifying and locating the searched object in a scene of randomly placed parts. Despite being studied extensively, recognition and localization still remains challenging for industrial setups. One of the main challenge is related to the presence of dimly or unevenly lit environments, in which the acquired images have high contrast variations making it difficult to process them. Most of the time the parts are randomly scattered in a bin or improperly stacked, hence it is necessary for the vision system to cope with clutter and occlusions. Another challenge is caused by the properties of the objects to be recognized and located. They generally have simple shapes with no texture information, such that most of the recognition algorithms that account for salient features regarding complex shapes, surface normals or texture information would fail, leaving the contour of the object being the most reliable feature. With the increased availability of 3D vision systems, there is a trend in using 3D data instead of traditional 2D images. 3D vision systems give the possibility to directly register the part to the received point cloud and they do not suffer from viewpoint changes or projection errors, which are crucial problems for conventional cameras. However, 3D sensors still have important limitations; they are not robust with respect to surface reflectance and most of the time they require expensive equipment to detect thin objects, which also limits the field of view. Therefore considering also their ease of availability and cost effectiveness 2D cameras are still popular.

In this work we present a framework for model based detecting and localizing multiple objects using a single image acquired by a monocamera, and for finding the best match representing the best object to be manipulated. The general pipeline of the algorithm is provided in Figure 1, where a sequence of steps are proposed to recognize and localize the best object to be manipulated. First step is the recognition of the objects in the scene. This part is based on the template matching approach proposed by Liu et. al. (2012), rooted from the traditional chamfer matching. We have picked a template matching method that exploits the contour considering the shape property of the objects to be recognized. The gist of the algorithm is to find the best parameters that align the template within the image with respect to a dissimilarity cost based on chamfer distance augmented by a term accounting for the orientation mismatch. Line based representation of the detected edges is used as well for computing the integral of the distance transform along quantized directions to improve the speed and

Sakcak, B., Bascetta, L. and Ferretti, G.
Model based Detection and 3D Localization of Planar Objects for Industrial Setups.
DOI: 10.5220/0005982503600367
In *Proceedings of the 13th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2016) - Volume 2*, pages 360-367
ISBN: 978-989-758-198-4

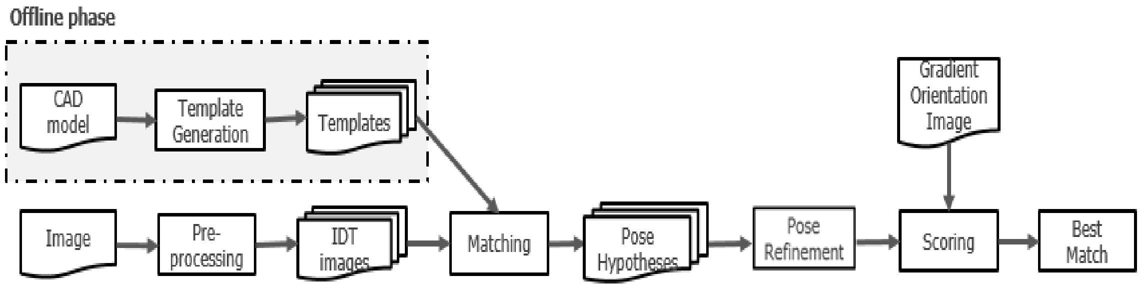


Figure 1: Pipeline of the recognition and localization.

reduce the complexity of the search. The recognition procedure provides a set of coarse pose hypotheses. A pose refinement method, that takes into consideration also the missing edges and occlusions, allows then to precisely localize multiple objects of the same kind. We also augmented the framework with a scoring part based on the gradient orientation computed on the query image that allows us to pick the best object to be manipulated i.e. the topmost one and discard false matches.

1.1 Related Works

Standard cameras and 2D image analysis have been studied for various applications, however, matching the detected object in an image with its 3D model is still challenging. For that purpose one popular approach is to use invariant feature descriptors such as SIFT (Lowe, 2004), SURF (Bay et al., 2006), SceneTensor (Söderberg et al., 2005) to find the correspondences between the acquired image and the 3D model. Another approach is to match the retrieved image with a database of images trained from the object. However, this approach is computationally inefficient and also prone to errors due to appearance changes. To overcome these issues, a hierarchical view-based approach is proposed in (Ulrich et al., 2009), adopting the descriptor that uses the difference between the gradient orientations computed in the image and the orientation attributed to the edge (Steger, 2002), which is robust to appearance changes and occlusions. Another method improves the usage of this descriptor for small translations and rotations by spreading the gradient orientations and computing offline response maps for a highly optimized matching procedure (Hinterstoisser et al., 2012). Despite being one of the earliest shape matching algorithms Chamfer Matching (Barrow et al., 1977) still remains popular. This approach relies on the minimization of the distance between two sets of edge points, which can be speeded up using Distance Transform. There are several variants proposed in the literature that account also for the edge orientation in the computation of the cost together with the traditional chamfer distance (Shotton et al., 2008), (Liu et al., 2012). Other shape matching methods, such as voting based approaches, utilize Hough Transform or a Hough like voting scheme. Cozar's group exploits General Hough Transform to locate 3D arbitrary planar shapes, where the parameter detection is uncoupled by the usage of invariants (Cózar et al., 2001). Pretto et al. (2003) presented a novel cost function based on dynamically adapted gradient magnitude to be implemented in a Hough-like voting approach. In the recent years, with 3D sensors becoming more cost effective, there has been an increased interest in research on using directly the 3D data for localization. Nieuwenhuisen et al. (2013) proposed a method to detect an object using its primitives. They used the data obtained by a Microsoft Kinect RGB-D camera attached to the pan-tilt head of an anthropomorphic robot. To overcome the limited field-of-view, multiple scans are overlapped using the ICP algorithm (Zhang, 1994). Based on the algorithm initially proposed in (Schnabel et al., 2008) an annotated graph is formed where the nodes correspond to simple shapes (spheres, cylinders, planes) both for the model and the scene. These two graphs are then queried for matching. Papazov's group presented a method based on a robust geometric descriptor, hashing technique and a RANSAC-like sampling strategy (Papazov et al., 2012). In that approach the object model is prepared as oriented point pairs and its geometric descriptors are stored in a hash table. The retrieved point pairs are used to compute the same descriptor, which is then used as a key to access a hash table. Finally, similar points give out the transformation that maps the object to the scene. This solution is accepted or rejected based on a RANSAC-based acceptance function. Voting approach is employed also for methods based on 3D data. A novel technique is introduced to create a global model description and match it locally, allowing to use sparser point clouds (Drost et al., 2010). The global model description is carried out offline and it represents a mapping from the point pair feature space to the model. A Hough-like voting scheme is used on a geometric feature descriptor of pairs of oriented point pairs. A subset of hypothesis are then chosen using a nearest neighbour clustering algorithm. The final pose refinement is sustained by using the ICP algorithm. Another approach is to enhance this method by various pre and post processings, e.g. (Skotheim et al., 2012) and (Choi et al., 2012) introduced new point pair descriptors.

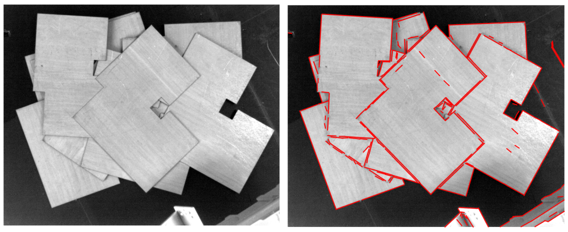
2 OBJECT RECOGNITION

For the object recognition step, we adopt a template matching approach that recognizes the object in the scene by searching for the parameters that align the template while minimizing a cost including the chamfer distance and a direction mismatch called *Directional Chamfer Matching* (Liu et al., 2012). At the end of the recognition part we obtain a set of coarse pose hypotheses with minimum costs corresponding to the 3D poses of the searched objects in the scene. In this section we will explain template generation and line segment based representation of the edges. A brief description of the shape matching algorithm will be given as well.

2.1 Template Generation

Assume that a virtual camera is located at the origin of a world coordinate frame while aligning the camera optical axis with the z-axis, the rotation about this axis and the translations in the plane orthogonal to the optical axis are defined as in-plane parameters. During matching, the algorithm takes into account in-plane rotation (θ_z) and translations (t_x, t_y). However, rotations about the remaining axes (θ_x and θ_y) result in different object contours in two-dimensional image plane. In this work, we particularly focused on detection of planar objects. For this case, the shape deformation caused by the out-of-plane rotations is relatively small to be handled by the chamfer matching algorithm. Thus, we used a single reference template, to be searched in the acquired image. Nonetheless the algorithm is flexible to function also with multiple templates.

Template is generated automatically during the off-line phase. For that purpose we first obtain a set of n 3D points defined as $\tilde{U} = \{\tilde{u}_i\}_{i=1}^n$ and their directions in the object reference frame. The points are generated by rasterization of a 3D CAD model with a step selected considering the size of the object. The direction of each point is obtained as the local direction of the edge line at which the point belongs. Considering a transformation matrix $\mathbf{T_m}$ that expresses the pose of the object model in the camera coordinate frame exploiting the pose $\mathbf{p} = [t_x\ t_y\ t_z\ \theta_x\ \theta_y\ \theta_z]^T$



(a) (b)

Figure 2: (a) Query image. (b) Line based representation of the edges.

$$\mathbf{T_m}(\mathbf{p}) = \begin{bmatrix} \mathbf{R_m} & \mathbf{t_m} \\ 0 & 1 \end{bmatrix} \tag{1}$$

a template is generated by projecting the points defined in the object frame to the image frame using the following equation

$$u_i = \mathbf{P}\mathbf{T_m}(\mathbf{p})\tilde{u}_i \tag{2}$$

where $\mathbf{P}$ is the 3x4 projection matrix obtained by the camera calibration, and $\mathbf{T_m}$ is the pose of the 3D model with respect to the camera frame. Points are represented in homogeneous coordinates, and since the objects have a planar shape all 3D points of the template in the object frame have $z = 0$. Initial pose $\mathbf{p}_0$ consists of only an assumption for the distance of the objects from the camera, hence $\mathbf{p}_0 = [0\ 0\ t_z\ 0\ 0\ 0]^T$.

2.2 Representation of Edge Points

The edge map of a query image is represented as composed of line segments instead of a binary image. This provides a minimal representation since all the edge pixels along a line segment can be represented by using only two points. Furthermore, direction information of each edge point can be easily computed using a line based representation. Considering industrial setups and the related objects, traditional edge detection methods tend to provide poor results due to low gradient values. In order to increase the detection rate we used a state of the art edge detection algorithm (Figure 2) called *Line Segment Detector* (von Gioi et al., 2008). As a result we obtain a set of line segments and their directions. For the computation of directional chamfer matching cost and the related cost maps, directions are quantized into a number q of orientation channels that equally spans $[0, \pi)$. We used $q = 60$ orientation channels.

2.3 Directional Chamfer Matching

Chamfer Matching (Barrow et al., 1977) is a contour based technique to detect a template in an image and to find the best alignment parameters. Two point sets can be defined as $U = \{u_i\}_{i=1}^{n}$ being the edge pixels of the template edge map and $V = \{v_j\}_{j=1}^{|V|}$ being the edge pixels of the query image edge map, where n and $|V|$ represent the total number of the edge points in each set. Then, the *chamfer distance* which expresses the dissimilarity between these two point sets can be defined as the average of the distances between the edge pixels $u_i \in U$ and their nearest pixel in V

$$d_{CM}(U,V) = \frac{1}{n} \sum_{u_i \in U} \min_{v_j \in V} ||u_i - v_j|| \tag{3}$$

For efficiency, the chamfer distance between two point sets can be computed using the *distance transform* (DT). DT inputs a binary edge map and assigns to each pixel $\mathbf{x}$ the minimum distance to the nearest edge point

$$DT_V(\mathbf{x}) = \min_{v_j \in V} ||\mathbf{x} - v_j|| \tag{4}$$

Using the Generalized Distance Transform (Felzenszwalb and Huttenlocher, 2004), the DT of an image can be computed in linear time by dynamic programming. As a result, the problem of computing the chamfer distance between the template points and the image edge points is transformed into a look-up table and is defined as;

$$d_{CM}(U,V) = \frac{1}{n} \sum_{u_i \in U} DT_V(u_i) \tag{5}$$

Directional chamfer matching (DCM) makes use of the direction term attributed to each pixel thanks to a line based representation. The DCM cost is given as

$$d_{DCM}(U,V) = \frac{1}{n} \sum_{u_i \in U} \min_{v_j \in V} \left(\left\| u_i - v_j \right\| + \lambda \left\| \phi(u_i) - \phi(v_j) \right\| \right) \tag{6}$$

where λ is the weight for the direction term and $\phi(u_i)$, $\phi(v_j)$ represent the orientations of the template and edge points, respectively. In order to reduce the complexity of the problem of finding the in-plane parameters that minimize this cost, a three-dimensional distance transform ($DT3_V$) is used to compute the matching cost in linear time. $DT3_V$ jointly computes the minimum distance of each pixel to an edge point in terms of location and orientation. For each pixel and direction channel, $DT3_V$ cost is expressed as

$$DT3_V(\mathbf{x}, \phi(\mathbf{x})) = \min_{\hat{\phi}_i \in \Phi} \left(DT_{V(\hat{\phi}_i)} + \lambda \left| \left| \hat{\phi}(x) - \hat{\phi}_i \right| \right|_{\pi} \right) \tag{7}$$

where $\hat{\phi}(x)$ and $\hat{\phi}_i$ represent the quantized orientation assigned to a pixel and the orientation channel of the cost map the pixel belongs to. In order to compute the $DT3_V$ edges that belong to the same orientation channel are grouped in the same binary edge image and a distance transform is computed for each of them. A total number of q distance transform images ($DT_{V(\hat{\phi}_i)}$) are obtained and using dynamic programming the difference between the orientation of the edge point that corresponds to a pixel and the orientation channel is added to the cost. As a result the directional chamfer matching cost for a template U is computed as

$$d_{DCM}(U,V) = \frac{1}{n} \sum_{u_i \in U} DT3_V(u_i, \hat{\phi}(u_i)) \tag{8}$$

Considering all pixels along a line segment belonging to the same orientation channel the cost is easily computed using the integral distance transform representation. Integral distance transform represented as

$$IDT3_V(\mathbf{x}, \hat{\phi}_i) = \sum_{x_j \in [x_0, x]} DT3_V(\mathbf{x}_j, \hat{\phi}_i) \tag{9}$$

is achieved by summing the $DT3_V$ cost of a specific orientation channel over the points along that direction. Taking $L_U = l_{[s_i,e_i]}, i = 1, \ldots, m$ as the representation of the line segments defining a template, where s_i and e_i are the start and the end points of the i^{th} segment, the chamfer matching score of each segment is computed as

$$d_{DCM}(U,V) = \frac{1}{n} \sum_{l_i \in L_U} \left(IDT3_V(e_i, \hat{\phi}(l_i)) - IDT3_V(s_i, \hat{\phi}(l_i)) \right) \tag{10}$$

2.4 Matching

Matching is the process of finding the best in-plane parameters $\hat{s} = (\theta, t_x, t_y)$ with the lowest DCM cost that align the template within the query image. Searching these parameters individually with a brute-force approach is computationally inefficient. Therefore, as proposed by Liu et. al. (2012) the search is guided by using the longest line segments from the corresponding query and template sets. To this intent, the template is rotated and translated to be aligned with the query line segment such that the end point of the template line segment coincides with the query line segment.

3 POSE REFINEMENT AND BEST MATCH SELECTION

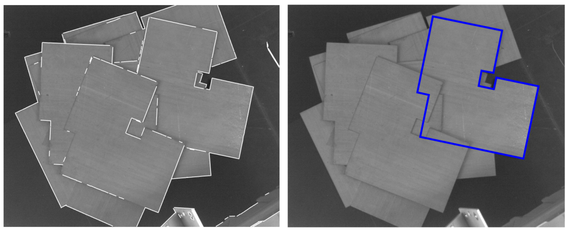
The coarse pose hypotheses obtained from the recognition part are limited to the out-of plane parameters, that are used to render the template, and the search step taken during matching phase. Thus, in order to precisely estimate the exact three dimensional pose of the object, a fine refinement step is necessary. Furthermore, we intend to identify the best object to be manipulated, which we define as the topmost in the batch. In an ideal case of all possible edges detected this object would be the one with the lowest DCM cost, however low contrast regions due to overlapping textureless objects give rise to image zones with less detected edge lines, hence resulting in higher scored regions. One such example is reported in Figure 3, where the minimum cost pose hypothesis does not correspond to the topmost object. In order to avoid this, we exploit the local gradient orientations computed in the query image to perform a scoring step.

3.1 Pose Refinement

First the translation parameters t_x and t_y, that are expressed in image coordinates, are back-projected to the camera coordinate frame using the camera projection matrix $\mathbf{P}$. Back-projected translation parameters and the estimated rotation about the camera axis (θ_z) combined with the out-of-plane parameters (θ_x, θ_y) and the initial assumption of distance from the camera t_z are used to define the matrix $\mathbf{T_m}(\mathbf{p})$ as in (1), that represents the coarse pose of the object in the camera coordinate system. We minimize the least squares projection error to refine the pose estimate. In order to compute the least squares error a set of correspondences is necessary. To this extent, template points are projected on the image plane using (2). For each u_i projected on the image plane a nearest edge point v_i in V is found which minimizes the DCM cost, such that

$$\arg\min_{v_i \in V} \left\| u_i - v_j \right\| + \lambda \left\| \phi(u_i) - \phi(v_j) \right\| \tag{11}$$

Considering the missing edges due to low contrast or objects with partial occlusion, it is not logical to use



(a) (b)

Figure 3: (a) Missing edge lines caused by low contrast. (b) Hypothesis with the minimum DCM cost which is not the best match.

all the correspondences as they might cause the optimization algorithm to converge to a wrong result. Hence, we use a thresholding such that only the pairs that have a DCM cost smaller than a threshold are used in the refinement process. We use a thresholding based on a factor δ related to the median of the costs computed for every template point, unless it is below a certain value μ_{base}, such that the cost threshold of accepting a point pair for refinement can be expressed as

$$\mu = \begin{cases} \delta \mathrm{median}(\mathrm{d_{DCM}}), & \text{if } \mu_{base} < \delta \mathrm{median}(\mathrm{d_{DCM}}) \\ \mu_{base}, & \text{otherwise} \end{cases} \tag{12}$$

As a result 3D-2D point correspondences are established as $(\tilde{u_k}, v_k)$, where $\tilde{u_k}$ is a subset of rasterized template points $\tilde{u_i}$ that have correspondences v_k within the cost bound defined by the thresholding. Using these point pairs the least squares projection error is defined as follows[1]

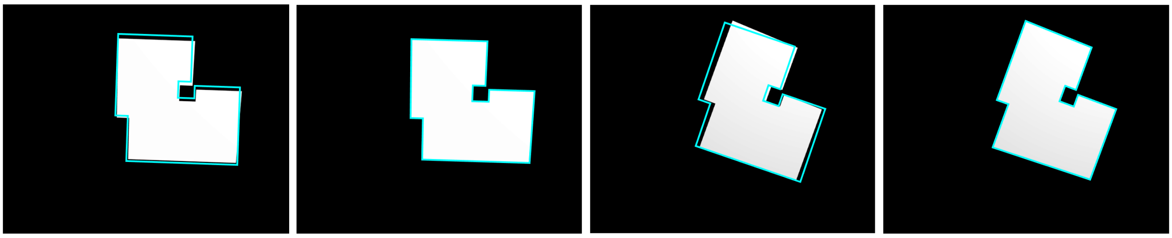
$$\varepsilon(\mathbf{p}) = \sum_{\tilde{u_k}} \left\| \mathbf{P}\mathbf{T}_m(\mathbf{p})\tilde{u_k} - v_k \right\|^2 \tag{13}$$

Error function is then minimized for each hypothesis using the Levenberg-Marquardt algorithm by finding at each step a set of point pairs after outliers have been removed.

3.2 Best Match Selection

Assuming the best object to be manipulated is topmost object in the batch, it should have all of its edges visible. Hence, when projected on the image plane the normals of the direction terms assigned to the template edge points should coincide with the corresponding local gradient orientations in the query image. A similar measure is also used in (Pretto et al.,

[1]For the ease of notation we assume that the projection of 3D points are already converted into image coordinates to compute the error.



(a) (b) (c) (d)

Figure 4: Examples of correct pose estimation in various out-of-plane rotations using single template (a)(c) Coarse pose estimations (b)(d) Estimated poses after fine registration.

Table 1: Results from the experiments using synthetic images.

Average error	t_x	t_y	t_z	θ_x	θ_y	θ_z
	<0.5 mm	<0.5 mm	1.2 mm	1deg	1deg	0.3deg

2013). As a result the scoring function is defined as follows

$$\mathcal{S}(U, I_{go}) = \sum_{u_i \in U} |\cos(I_{go}(\mathbf{x}_i)) - n_o(u_i))| \quad (14)$$

where $I_{go}(\mathbf{x}_i)$ is the gradient orientation of the pixel corresponding to the template point u_i projected on the image frame using (2). The best matching pose $\mathbf{p}^*$ is then selected being the one that gives out a transformation matrix $\mathbf{T}^*_{\mathbf{m}}$ that projects 3D template points $\tilde{u}_i$ on the image plane with the highest score $\mathcal{S}(U, I_{go})$. This scoring also allows us to discard falsely recognized and located objects as well. Considering the best match would have a score equal to 1 we eliminate all matches that receive a score less than a threshold.

4 EXPERIMENTS AND RESULTS

We conduct experiments both on synthetic and real images used as two different mediums to verify the functionality of the recognition and localization framework.

4.1 Synthetic Data

Synthetic images are used to test the accuracy of the pose estimation. We used Blender software to render a set of images of 3D objects. The object is rendered changing out-of-plane and in-plane parameters. Using the synthetic images we found that the fine refinement algorithm is capable of recovering an out-of plane rotation bounded in $\pm 20^\circ$ when a single template is used. Figure 4 shows two examples of the projected pose of the template after coarse pose estimation and fine registration, respectively. It is possible to observe that the shape change due to out-of-plane rotation of the object is rather small, and a single template is sufficient to achieve a precise localization of the object. As a result, 50 synthetic images are generated with known poses and the results as the averages of the errors for translation and rotation are reported in Table 1.

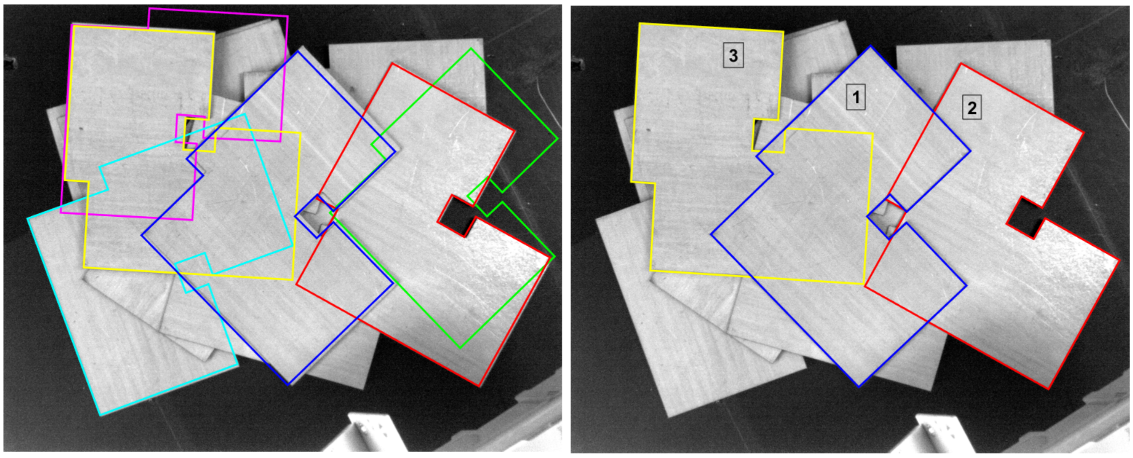
4.2 Real Images

In order to test the algorithm using real images a simple experimental set-up was designed (Figure 7). We used a 1.3 megapixels grey-level CMOS camera mounted on a fixed frame facing the table where the work pieces are located, and an ABB IRB 140 robotic manipulator with a calibration tool attached to the end effector. Camera intrinsic parameters are computed using the single camera calibrator application of MATLAB. Camera is then calibrated with respect to the robot world frame to obtain the transformation matrix T^{cam}_{base} which maps the points represented in the camera frame to the robot world frame.

Table 2: Results from real images.

Average error	t_x	t_y	t_z
	<0.5 mm	<0.5 mm	2 mm

The procedure to verify the localization accuracy of the algorithm in a real world setup starts with obtaining the values of the corner points of the query object in the camera frame using the best match as shown in Figure 5. These points are then mapped to the robot world frame using the transformation matrix. The actual values of the corner points are obtained directly in the robot world frame by manually touching the corners of the object in the jogging mode using the calibration tool, a probe with a known



(a) (b)

Figure 5: Final results of the localization on real images a)selected hypotheses before pose refinement b)refined poses and the selected matches that have sufficient score value, the best match is denoted as '1'.

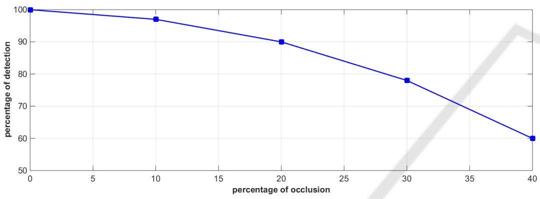


Figure 6: Percentage of recognition with respect to percentage of occlusion.

length mounted on the robot wrist. These results are then compared to test the performance of the algorithm. Results in the form of the averages of the errors are reported in Table 2. One can also see that the hypotheses that have higher scores but falsely recognized or highly occluded are removed after the refinement/scoring step. We have evaluated the localization of the other hypotheses qualitatively, by projecting the 3D template points on the image plane.

We have also tested the performance of the algorithm when the objects to be detected are occluded. The occlusion is defined as the amount of area of an object covered by other objects, the results of the detection rate are reported in Figure 6. The algorithm performs well in detecting objects with less than %10 of occlusion.

5 CONCLUSIONS

We presented a framework for model based recognition and pose estimation of planar, textureless objects. The method can be used to avoid rigid mechanical solutions for manipulation and inspection purposes in industrial environments. As the original algorithm exploits a special vision system that allows to detect edges accurately, the single output provides the best object to be manipulated. However when a conventional camera is used the resulting best match might not be the best object to be manipulated, due to the low contrast regions with less amount of detected edges. For that reason we have modified the pose refinement step, that now allows to recognize and localize multiple objects of the same kind, and augmented the algorithm with a scoring step based on the gradient orientation that gives out the best match. With the proposed approach, we obtained a good best match recognition rate and localization accuracy that is suitable for industrial environments.

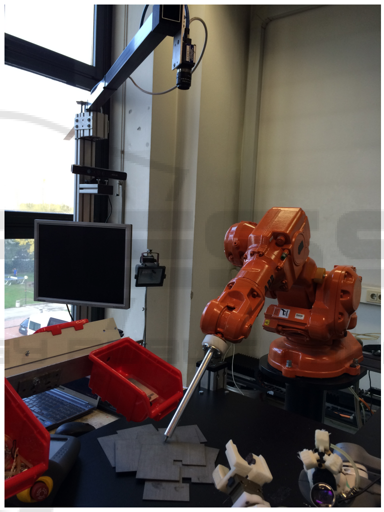


Figure 7: Experimental setup.

REFERENCES

Barrow, H., Tenenbaum, J., Bolles, R., and Wolf, H. (1977). Parametric correspondence and chamfer matching: two new techniques for image matching. In *Proceedings of the 5th international joint conference on Artificial intelligence-Volume 2*, pages 659–663. Morgan Kaufmann Publishers Inc.

Bay, H., Tuytelaars, T., and Van Gool, L. (2006). Surf: Speeded up robust features. In *Computer vision–ECCV 2006*, pages 404–417. Springer.

Choi, C., Taguchi, Y., Tuzel, O., Liu, M.-Y., and Ramalingam, S. (2012). Voting-based pose estimation for robotic assembly using a 3d sensor. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1724–1731. IEEE.

Cózar, J. R., Guil, N., and Zapata, E. L. (2001). Detection of arbitrary planar shapes with 3d pose. *Image and Vision Computing*, 19(14):1057–1070.

Drost, B., Ulrich, M., Navab, N., and Ilic, S. (2010). Model globally, match locally: Efficient and robust 3d object recognition. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Felzenszwalb, P. and Huttenlocher, D. (2004). Distance transforms of sampled functions. Technical report, Cornell University.

Hinterstoisser, S., Cagniart, C., Ilic, S., Sturm, P., Navab, N., Fua, P., and Lepetit, V. (2012). Gradient response maps for real-time detection of textureless objects. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(5):876–888.

Liu, M.-Y., Tuzel, O., Veeraraghavan, A., Taguchi, Y., Marks, T. K., and Chellappa, R. (2012). Fast object localization and pose estimation in heavy clutter for robotic bin picking. *The International Journal of Robotics Research*, 31(8):951–973.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110.

Papazov, C., Haddadin, S., Parusel, S., Krieger, K., and Burschka, D. (2012). Rigid 3d geometry matching for grasping of known objects in cluttered scenes. *The International Journal of Robotics Research*, page 0278364911436019.

Pretto, A., Tonello, S., and Menegatti, E. (2013). Flexible 3d localization of planar objects for industrial bin-picking with monocamera vision system. In *Automation Science and Engineering (CASE), 2013 IEEE International Conference on*, pages 168–175. IEEE.

Schnabel, R., Wessel, R., Wahl, R., and Klein, R. (2008). Shape recognition in 3d point-clouds. In *The 16-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, volume 8. Citeseer.

Shotton, J., Blake, A., and Cipolla, R. (2008). Multiscale categorical object recognition using contour fragments. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(7):1270–1281.

Skotheim, Ø., Lind, M., Ystgaard, P., and Fjerdingen, S. A. (2012). A flexible 3d object localization system for industrial part handling. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 3326–3333. IEEE.

Söderberg, R., Nordberg, K., and Granlund, G. (2005). An invariant and compact representation for unrestricted pose estimation. In *Pattern Recognition and Image Analysis*, pages 3–10. Springer.

Steger, C. (2002). Occlusion, clutter, and illumination invariant object recognition. *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, 34(3/A):345–350.

Ulrich, M., Wiedemann, C., and Steger, C. (2009). Cad-based recognition of 3d objects in monocular images. In *ICRA*, volume 9, pages 1191–1198.

von Gioi, R. G., Jakubowicz, J., Morel, J.-M., and Randall, G. (2008). Lsd: A fast line segment detector with a false detection control. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (4):722–732.

Zhang, Z. (1994). Iterative point matching for registration of free-form curves and surfaces. *International journal of computer vision*, 13(2):119–152.