

System for Conveyor Belt Part Picking using Structured Light and 3D Pose Estimation

J. Thielemann^a, Ø. Skotheim^{a†}, J. O. Nygaard^a, T. Vollset^b

^aSINTEF ICT, PB 124 Blindern, N-0314 Oslo, Norway

^bTordivel AS, Storgata 20, N-0184 Oslo, Norway

ABSTRACT

Automatic picking of parts is an important challenge to solve within factory automation, because it can remove tedious manual work and save labor costs. One such application involves parts that arrive with random position and orientation on a conveyor belt. The parts should be picked off the conveyor belt and placed systematically into bins. We describe a system that consists of a structured light instrument for capturing 3D data and robust methods for aligning an input 3D template with a 3D image of the scene. The method uses general and robust pre-processing steps based on geometric primitives that allow the well-known Iterative Closest Point algorithm to converge quickly and robustly to the correct solution. The method has been demonstrated for localization of car parts with random position and orientation. We believe that the method is applicable for a wide range of industrial automation problems where precise localization of 3D objects in a scene is needed.

Keywords: 3D Pose Estimation, 3D Robot Vision, Pick and Place, Structured Light

1. BACKGROUND

Many production processes involve intermediate conveying of produced parts. These parts need to be picked off the conveyor and inserted either into containers for later shipping or into further production processes. Manual picking of conveyed parts is a dull and tiresome job that also exposes people to the risk of physical strain. Naturally, automation of the task is highly desirable.

In many production lines, the parts that are produced arrive with random position and orientation on a conveyor belt. To be able to pick the parts off the conveyor belt and place them with known orientation in bins, it is necessary to recognize the position and pose of all of the objects. This information can in turn be used by the robot to systematically pick and place the objects. This paper describes our system for object pose recovery of conveyed parts. The system includes a structured light instrument combined with robust methods for aligning components of a 3D image of the scene with a 3D template. The work extends on previously described methods for quick recognition of geometrical primitives in a 3D image[1], and the algorithms presented here have been implemented into Scorpion Vision Software®, a commercial software package developed by Tordivel AS for typical 2D and 3D machine vision tasks.

Systems used in the industry for conveyor picking have traditionally been based on grayscale or silhouette images of the objects on the conveyor captured with a standard machine vision camera. Most of these systems utilize 2D image analysis tools, such as blob analysis, pattern matching or analysis of the object contour to find *e.g.* the position in *x* and *y* of the center of the objects. These systems can be sensitive to changes in lighting conditions, changes in color or texture on the object surface or other artifacts in the 2D image.

In order to increase the robustness of the system, minimize influences from lighting or color changes and to be able to localize the objects with all six degrees of freedom, we have chosen to use a structured light camera to capture 3D images of the scene and to use 3D image analysis algorithms to determine the position and pose of the objects on the conveyor belt. As input to the algorithm for pose determination, we use a template in the form of a point cloud. This template can be obtained by performing a 3D scan of the object, or it might be obtained by sampling points on the surface of a CAD model of the object.

[†] Send correspondence to Ø. Skotheim (E-mail: oystein.skotheim@sintef.no, Telephone: +47 73597047)

Current methods for 3D part picking in the literature can be grouped into those attempting to recover the complete object geometry, and those that only focus on sufficient recognition for robot picking. In the latter group, the focus is on detecting surface areas suitable for picking, like planar surfaces [3] or parallel opposing box sides [4]. Obviously this limits the system to work only with objects that exhibit this kind of features.

Another strategy is to detect the object pose prior to picking the part. Such object pose recognition can be done either by first detecting and matching feature-points (using methods like harmonic shape contexts [5], tensors [6] or spin images [7, 8]), or by doing partial surface registration by *e.g.*, matching histograms of surface normals [9]. Some of these methods require that the object first has been segmented from the background, enabling the object to be analyzed without influence from other parts of the scene.

Typically these initial alignment algorithms are used prior to a final registration step which is done using Iterative Closest Points (ICP). Different variants of ICP are well surveyed in [10] and also [11]. The reason for not directly using ICP is that its convergence basin is rather small, so that a good initial guess is required.

Our approach is based on a similar multi-step refinement of object pose, and uses a mixture of the strategies outlined in the references above. We first initialize the detection by performing range image segmentation using depth discontinuities in the scene combined with a measure of range data quality. We initialize the pose detection by detecting geometric primitives in the object and thus removing many degrees of freedom, somewhat similar to the strategies that search for areas that are pickable [3, 4]. We further refine the object pose estimate by exhaustively searching through the remaining degrees of freedom using a scaled-down model, and by measuring distance between segmented range data and the model. This provides us with starting points for final fitting by ICP.

Due to the fact that only one part is to be picked at a time, it is sufficient for our algorithm to be able to localize one part within the image. This means that parts occluded by other more visible parts can safely be ignored, as these will no longer be occluded when the robot has picked the occluding part. We do not consider cases of severe (circular) tangling.

To evaluate our method experimentally, we have tested it for position and pose recognition of a car part made of cast steel. We have captured data of a series of different distributions of three such car parts, and we have measured the effectiveness of the method by evaluating the RMS error after aligning an a priori known 3D model and segmented objects from the 3D image of the car part.

The paper is structured as follows. Section 2 contains a description of the measurement setup used for acquiring 3D images of the components. In section 3 an overview of the different steps of the proposed pose recovery algorithm is given. Section 4 describes a set of experiments that were done to evaluate the performance of the system, and the results of these experiments are presented in section 5. Finally, a brief discussion and conclusion is given in section 6.

2. SYSTEM DESCRIPTION

Our structured light system based on a standard multimedia projector and a digital camera was used to acquire the 3D shapes of the components on the conveyor belt. Structured light was chosen due to relatively low component cost, its ability to capture full-field 3D images and its high accuracy.

Figure 1 shows a sketch of the measurement system. The projector and the camera are mounted on a frame that is attached to the ceiling above the conveyor. The projector illuminates an area of the conveyor belt via a mirror, while a camera mounted at an angle of approximately 30 degrees to the projector captures images of the same area. The conveyor is brought to rest while acquiring the shape of the components. The conveyor does not advance until all the components have been identified and picked by the robot. If the system fails to identify one or more components, these components will be collected in a tray located at the end of the conveyor belt and they may later be sent back again into the system.

The measurement system is based on a combination of Gray code and phase stepping fringe projection. The phase stepping patterns are used to obtain a high-resolution wrapped phase map of the object, while the Gray code patterns are used to associate a binary code with each individual stripe in the pattern and hence eliminate the need for phase unwrapping. Our structured light instrument is further described in [2]. A new, quick calibration procedure was developed based on the acquisition of a planar white glass plate with a printed chessboard pattern in a set of arbitrary positions in the volume. The calibration procedure is inspired by [12]. The result of a measurement is an ordered point

cloud with 1.2 million 3D points, each with a resolution (RMS noise value) of approximately 0.05 mm for our field of view of approximately 400x300 mm.

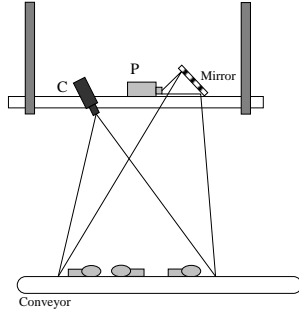
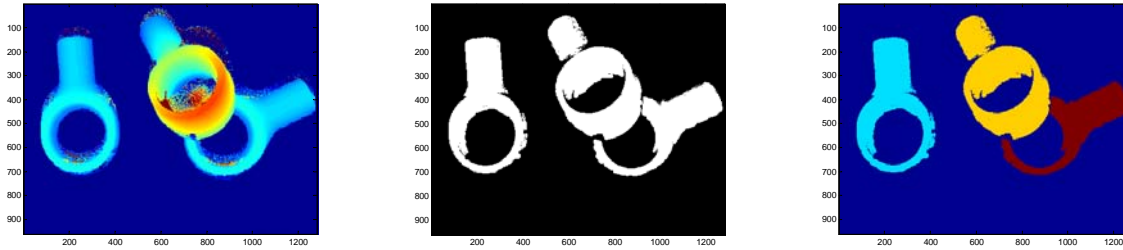


Figure 1: Measurement setup. The projector P projects a series of patterns which is captured by the camera C. The parts arrive on a conveyor belt, which is temporarily stopped during picking.



Figure 2: 3D model of the car part obtained by stitching together six 3D images taken from different angles of the component.



(a) Original range image

(b) Binary image indicating connectedness ($D_T(r,c)$)

(c) Resulting clustering of measurements into separate objects.

Figure 3: Segmentation of the scene into objects using distances to neighboring pixels in the range image.

3. OBJECT POSE RECOVERY

We have chosen to break the object pose recovery method up into multiple steps in order to make the recovery process less expensive. The method proceeds from an initial segmentation of the range image through gradual refinement of object pose. The goal is to estimate all six degrees of freedom of at least one of the parts present in the scene.

3.1 Range image segmentation

A single range image may contain multiple objects. It is beneficial for later methods to only analyze one single object at a time. To achieve this, we use height discontinuities to separate the parts from each other and from the conveyor belt.

This is performed by first calculating the function $D_T(r,c)$ for each range measurement $P(r,c)$ in the image:

$$D(r,c) = \min \left\{ \|P(r+1,c) - P(r,c)\|, \|P(r,c+1) - P(r,c)\| \right\} \quad (1)$$

$$D_T(r,c) = \begin{cases} 1 & D(r,c) < d \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where r, c indicates row and column in the range image (Figure 3a), and $\| \cdot \|$ indicates Euclidean distance. This provides a binary image, where non-zero pixels are connected to its neighbors to the left and below (Figure 3b). As separate objects may be too close to be properly separated using this method, we perform binary morphology to remove any spurious

links between objects. Separation of this binary image into separate objects is done by a 2D connected components algorithm [13], which groups neighboring non-zero pixels into objects (Figure 3c). The separate objects are subsequently analyzed separately.

3.2 Initial recognition of object pose

The pose of the objects consists of an estimate for position, (x,y,z) , and rotation, (α,β,γ) , in total six degrees of freedom. To speed up the pose estimation, we reduce the search space by performing an initial rough recognition of the object's pose. This search is done on the previously segmented object. The search is performed by searching for known geometric primitives within the object, like cylinders, spheres or planes. The obtained result can subsequently be used for eliminating some of the degrees of freedom from the further search of object pose. For instance, localization of a cylinder within the object will provide information about all degrees of freedom except from the rotation around the axis or the shift along the axis.

For detecting the geometric primitives we use our previously described RANSAC-based methods[1]. The main benefit of such RANSAC-based methods is that they provide resilience against outliers, *i.e.* data points that are not part of the geometric primitive.

Briefly described, they work by first selecting a small subset of random points within the dataset (assumed to be inliers), and use these initial points to estimate the parameters of the selected primitive. For the primitives supported, this can be done analytically based on either three independent points (if no surface normals are used) or fewer points (if surface normals are used). For instance, cylinder parameters can be estimated from two points p_1, p_2 with surface normals n_1, n_2 , both assumed to be on the cylinder and a previously known cylinder radius r . By knowing the radius r , only the cylinder axis parameters need to be estimated. Axis parameters can be estimated from finding two points a_1, a_2 on the axis, estimated through

$$a_i = p_i - rn_i \quad (3)$$

The remaining data points are then checked for consistency with the estimated parameters. Consistency is checked by first calculating the distance e_i^2 from each point p_i in the object to the closest point on the estimated geometric primitive, and then estimating the deviation d through the use of the MSAC criteria [14] to achieve robustness:

$$\rho(e_i^2) = \begin{cases} e_i^2 & e_i^2 < T \\ T & e_i^2 > T \end{cases} \quad (4)$$

$$d = \sum_i \rho(e_i^2) \quad (5)$$

An appropriate value for the constant T is determined experimentally.

The process of subset selection, parameter estimation and data consistency measurement is repeated, and the parameters that minimize d are selected for further use.

3.3 Rough pose estimation

A rough estimate of the remaining degrees of freedom is found through an exhaustive search. The goal is to find a pose that is close enough to the correct object pose to ensure that a subsequent run of the ICP algorithm will converge to the correct, global minimum. To do this in a timely manner, we discretize the remaining degrees of freedom, by choosing a step size that we have experimentally determined to match the convergence basin of ICP. Combined with the object pose from the previous geometrical primitive detection step (section 3.2), this provides a list of possible object poses which can be evaluated independently.

This pose evaluation is done by subsampling the data points contained in the measured object, and applying an affine transform to these points which represent the pose to be evaluated. The quality of the pose is then estimated by calculating

$$Q(P, M) = \sum_i \arg \min_j \|p_i - m_j\| \quad (6)$$

where P represent the data points p_i of the transformed object and M represent the data points m_i of the model. This is accelerated through the use of a KD-tree.

The pose that provides the minimum value for $Q(P,M)$ is chosen for further processing.

3.4 Final matching

As a final step, we execute the ICP algorithm on the pose candidate selected from the previous round. We use a free implementation of ICP found in the Visualization Toolkit (VTK) library [15]. As for the previous coarse pose estimation step, we subsample the data points on the measured object, while the target 3D model is left with full resolution.

4. EXPERIMENTS

The following experiment was carried out to characterize the performance of our system for position and pose estimation of one particular car part (see Figure 2 for a 3D model of the car part) The part is made of massive steel and measures approximately 170x100 mm. The set of 36.600 3D points making up the model shown in Figure 2 was used as a template in our alignment algorithms, as will be discussed further below.

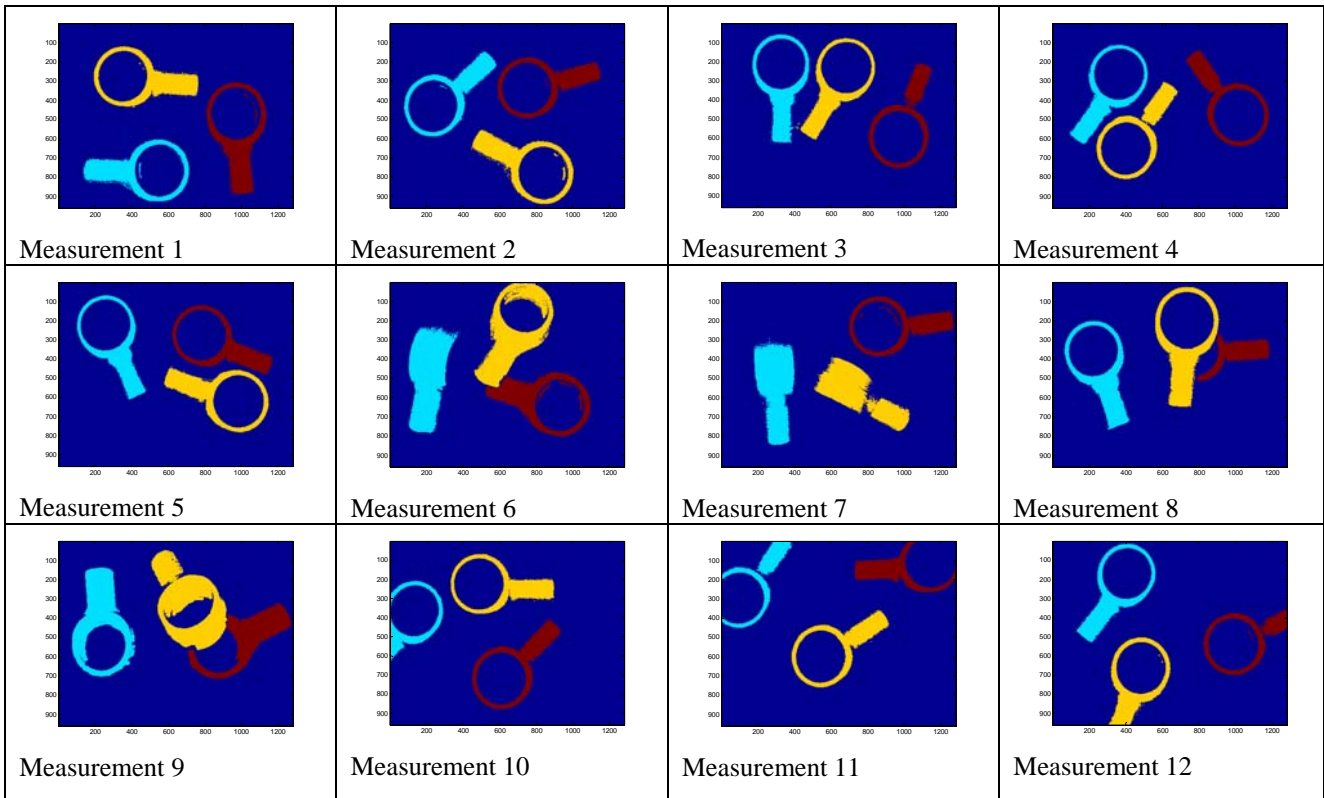


Figure 4: 12 measurements of parts on conveyor, shown as segmented range images

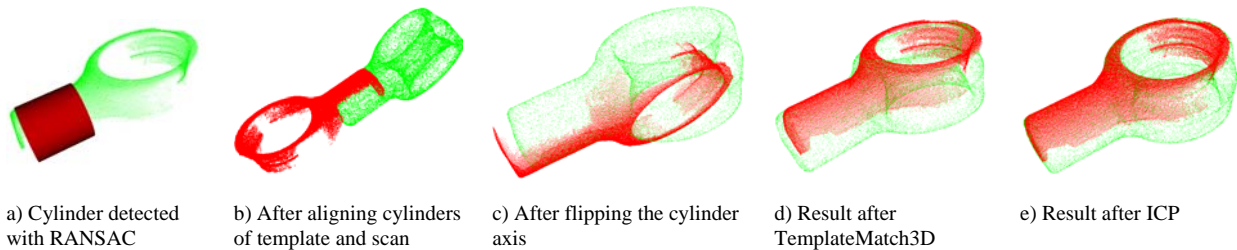
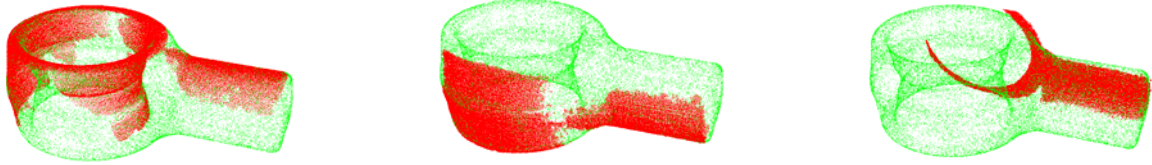


Figure 5: Results on various stages of our multi-step algorithm for 3D pose estimation for a particular car part.

As the conveyor belt was not operative at the time of writing, we decided to perform the experiment on the floor of our laboratory. 12 measurements were performed in which three car parts were distributed more or less randomly on the laboratory floor. The different distributions of parts for the 12 measurements are shown in Figure 4 as segmented range images. Most of the parts are lying flat on the floor, but the data set also includes parts resting on their short edges and even partly occluded parts supported on top of neighboring parts. Most of the parts are contained entirely within the field of view of the camera, but a few parts are also located partly outside the image.



a) Measurement 9, CC2 (success) b) Measurement 7, CC2 (success) c) Measurement 11, CC3 (failure)

Figure 6: Selected results of pose estimation after final ICP step.

Our structured light measurement system consists of a U5-632h DLP video projector and a Sony XCD-SX910 FireWire camera. The system was located approximately 70 cm above the parts and an area of about 400x300 mm was illuminated by the projector and captured by the camera.

After applying the range segmentation algorithm to identify three different connected components in each of the range images (section 3.1), we initialize the pose recognition by using our RANSAC methods to detect the cylinder in each part (Figure 5a). This lets us perform an initial alignment of the cylinder axes of the connected component and the template.

After alignment of the cylinder axes, three degrees of freedom remain to be estimated for this particular part: Translation along cylinder axis, rotation around cylinder axis, and flipping of the entire part (Figure 5b and 5c). We discretize the search space into 10 different translations along and 10 different rotations around the cylindrical axis, plus flipping the entire part.

When evaluating $Q(P,M)$ (see (6)) we sample 100 random points from the input point clouds. However, we choose to use all of the 36.600 points in our 3D template to ensure that there exist points in the template in close proximity of all of the more coarsely sampled points in our input point cloud. The pose that corresponds to the minimum value of $Q(P,M)$ provides a rough estimate of the correct object pose (Figure 5d).

The final matching is performed by applying the ICP algorithm after the first two steps for rough alignment have been carried out. We choose to sample 1000 points from our input point cloud and once again the full point cloud of the template is used. The maximum number of iterations for the ICP algorithm was set to 200. Figure 5e shows an example of a successful pose estimation obtained after ICP.

Performance evaluation is done by measuring the RMS error between the input point cloud and the model after the entire procedure, as a low RMS value indicates good pose estimation.

5. RESULTS

5.1 Object pose recovery

The algorithm was applied to the test data set presented in Figure 4, where the three connected components have been recognized and labeled as connected component number 1 (cyan), number 2 (yellow) and number 3 (red). The three columns labeled CC1, CC2 and CC3 in Table 1 correspond to each of these three identified connected components.

We obtained an average RMS error after executing the described algorithm (TemplateMatch3D) of 6.12 mm (Table 1a) and 0.67 mm after ICP (Table 1b), excluding results where the MSAC cylinder detection step failed. For this dataset, object pose was recovered for 95% of the parts imaged.

The algorithm succeeded for all the three connected components of the first 9 measurements, where the quality measure is in the range of 0.5 to 0.8 mm after alignment. An example of two successful alignments are shown in Figure 6 a and b. For measurement 10 our RANSAC method for cylinder detection failed because the cylindrical part is almost entirely

#	CC1	CC2	CC3
1	7.01	8.02	9.06
2	3.27	8.87	8.19
3	1.57	7.33	0.82
4	10.84	9.89	7.51
5	8.76	7.13	6.91
6	2.15	8.11	2.01
7	4.28	4.51	8.50
8	7.68	4.74	2.62
9	8.93	11.73	3.47
10	N/A	9.16	4.29
11	4.34	0.91	4.96
12	7.87	3.91	5.08

#	CC1	CC2	CC3
1	0.61	0.59	0.54
2	0.52	0.60	0.58
3	0.62	0.50	0.53
4	0.52	0.49	0.59
5	0.64	0.56	0.58
6	0.75	0.67	0.62
7	0.78	0.79	0.59
8	0.57	0.53	0.61
9	0.55	0.69	0.52
10	N/A	0.65	0.57
11	0.47	0.54	3.56
12	0.52	0.52	0.51

a) Results after TemplateMatch3D

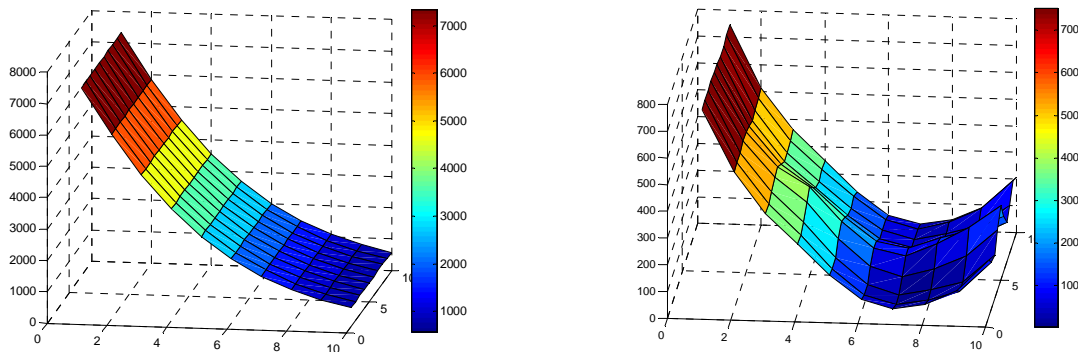
b) Results after final ICP alignment

Table 1: 3D pose estimation results for parts on conveyor. N/A indicates parts where the cylinder detection part of the algorithm failed, rendering further pose recovery impossible and RMS calculation meaningless.

outside the camera image. However, this is not a big problem since we have detected at least two components that are suitable for picking by the robot.

The only erroneous result seems to be that of measurement 11, which is shown in Figure 6c, where the ICP step has converged to a local minimum that does not coincide with the correct pose.

Figure 7 shows a graph of the quality measure, $Q(P,M)$, for the different poses evaluated for connected component number 2 of measurement 6. The exhaustive search, made computationally possible through the use of the MSAC geometric primitive detection step, makes the search for the global minimum possible.



a) $Q(P,M)$ for 100 different poses before flipping the cylinder axis

b) $Q(P,M)$ for 100 different poses after flipping the cylinder axis

Figure 7: Evaluation of $Q(P,M)$ for 200 poses of connected component 2 of measurement 6 after alignment of cylinders, before (a) and after (b) flipping of the cylinder axis. X-axis: Translation along cylinder axis. Y-axis: Rotation around cylinder axis. Z-axis: Evaluation of $Q(P,M)$. Note the difference in magnitude for $Q(P,M)$ in figure a and b.

5.2 Time usage

Detection of the cylinder part is done in approximately one second in Matlab on a 2.66 GHz Pentium Xeon computer. Further evaluation of the 200 poses is performed in approximately 0.5 seconds. This includes the time necessary to build up the necessary data structures, such as a KD-tree for the template and for the conversion of point cloud representations from MATLAB arrays to C++ arrays. The final ICP algorithm adds another second to the recognition. Total time required for position and pose estimation is therefore approximately three seconds for each part.

6. DISCUSSION AND CONCLUSION

Our results indicate that a multi-step approach to object pose recovery based on initial recognition of geometric primitives can be successful. Object pose is recovered for most of our test objects, even in the presence of partial occlusion and for parts resting on their short edges.

We believe that one of the keys to our success is our robust recognition of geometric primitives. Many industrial parts, especially those that are designed and/or manufactured by the use of CAD/CAM systems, contain primitives such as planes, cylinders or spheres. A robust detection of such primitives allows us to determine several degrees of freedom for the position and pose of the component.

The remaining degrees of freedom are detected through a manually configured exhaustive search. The reason for this search is that the measurement of pose similarity using point-to-point distances is prone to many local minima, something that prevents correct convergence for other popular pose estimation algorithms such as ICP. Due to the fact that only a few degrees of freedom remain after the geometric primitive detection step, our exhaustive search method becomes computationally feasible. Even using a mixture of Matlab and C++ algorithms, we were able to achieve full pose recovery in less than 3 seconds.

Further work will focus on increasing the robustness and speed of the system. This will be done by focusing on increasing the convergence basin of the ICP algorithm, so that fewer initial poses have to be evaluated. We will furthermore increase the speed of convergence by subsampling the data at the initial stages of the algorithm and gradually increase the number of points in order to recover the object pose precisely. The method will be tested for more challenging industrial automation tasks, such as random bin picking, where the objects are distributed randomly in a bin in multiple layers and possibly with more significant occlusion.

Today the setup of the exhaustive search is a manual process, which requires detailed knowledge of both the part and the convergence properties of ICP. We believe this process can be automated, partly by using CAD models to identify suitable geometric primitives contained in the part and partly by automatically determining the necessary bin sizes for the exhaustive search.

It would also be interesting to combine the proposed approach with other, local feature based approaches like harmonic shape contexts or spin images.

In summary, we believe that our presented method provides a robust and rapid way of recovering the 3D position and pose of many industrial parts, and that our system for part picking is well suited for industrial use.

REFERENCES

1. Skotheim, Ø., et al. *A flexible 3D vision system based on structured light for in-line product inspection* in *Three-Dimensional Image Capture and Applications 2008*, San Jose, CA, USA (2008):Proc. SPIE Vol. 6805. SPIE.
2. Skotheim, Ø. and F. Couwelleers. *Structured light projection for accurate 3D shape determination* in *12th International Conference on Experimental Mechanics*, Bari, Italy (2004). McGraw-Hill.
3. Martin, B., B. Gernot, and S. Stefan, *Vision guided bin picking and mounting in a flexible assembly cell*, in *Proceedings of the 13th international conference on Industrial and engineering applications of artificial intelligence and expert systems: Intelligent problem solving: methodologies and approaches*. 2000, Springer-Verlag New York, Inc.: New Orleans, Louisiana, United States.
4. Kristensen, S., et al., *Bin-picking with a solid state range camera*. Robotics and Autonomous Systems, 2001. **35**: p. 143-151.
5. Moeslund, T.B. and J. Kirkegaard, *Pose Estimation using Structured Light and Harmonic Shape Contexts*, in *Advances in Computer Graphics and Computer Vision*, J. Braz, et al., Editors. 2008, Springer. p. 281-292.
6. Mian, A.S., M. Bennamoun, and R. Owens, *Three-Dimensional Model-Based Object Recognition and Segmentation in Cluttered Scenes*. IEEE Transactions on Pattern Recognition and Machine Intelligence, 2006. **28**(16): p. 1584-1601.
7. Xiao, G., S.H. Ong, and K.W.C. Foong, *Efficient partial-surface registration for 3D objects*. Computer Vision and Image Understanding, 2005. **98**(2): p. 271-293.
8. Johnson, A.E. and M. Hebert, *Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes*. IEEE Trans. Pattern Anal. Mach. Intell., 1999. **21**(5): p. 433-449.
9. Makadia, A., A.I. Patterson, and K. Daniilidis. *Fully Automatic Registration of 3D Point Clouds* in *Computer Vision and Pattern Recognition*, (2006) Vol. 1. IEEE Computer Society.
10. Rusinkiewicz, S. and M. Levoy. *Efficient variants of the ICP algorithm* in *3D Digital Imaging and Modeling*, Quebec City, Canada (2001). IEEE Computer Society.
11. Gruen, A. and D. Akca, *Least squares 3D surface and curve matching*. ISPRS Journal of Photogrammetry and Remote Sensing, 2005. **59**: p. 151-174.
12. Vargas, J., J.A. Quiroga, and M.J. Terron-Lopez, *Flexible calibration procedure for fringe projection profilometry*. Optical Engineering, 2007. **46**(2): p. 023601-6.
13. Haralick, R.M. and L.G. Shapiro, *Computer and Robot Vision*. Vol. 1. 1992: Addison-Wesley. 28-48.
14. Torr, P.H.S. and A. Zisserman, *MLESAC: a new robust estimator with application to estimating image geometry*. Comput. Vis. Image Underst., 2000. **78**(1): p. 138-156.
15. Schroeder, W., K. Martin, and B. Lorensen, *The Visualization Toolkit, Third Edition*: Kitware Inc.