Pose Refinement of Transparent Rigid Objects With a Stereo Camera

Ilya Lysenkov
Itseez
ilya.lysenkov@itseez.com

Victor Eruhimov
Itseez
victor.eruhimov@itseez.com

Abstract

We propose a new method for refining 6-DOF pose of rigid transparent objects. The algorithm is based on minimizing the distance between edges in a test image and a set of edges produced by the training model with a specific pose. The model is scanned with a monocular camera and a 3D sensor such as a Kinect device. The pose is estimated from a monocular image or a stereo pair. The method does not require a CAD model of the object. We demonstrate experimental results on a set of kitchen items essential for any home and office environment.

Keywords: *pose estimation, localization, transparent objects.*

1. INTRODUCTION

Perception for personal robotics is a wide and important application of computer vision. A personal robot is expected to efficiently interact with the environment. In particular, it has to be able to detect a specific object in a scene and find its pose for grasping and manipulation. Recent advances in object recognition and pose estimation [1] demonstrate good results with a monocular camera for textured objects. SIFT features are used to find similarities between training and test textured image patches and then geometric validation is used to filter out false matches. Since the training set contains 3D coordinates of all features, pose estimation in this approach is done by solving a PnP problem on SIFT matches. However if an object has few textured features, local descriptors will produce few matches and detection will fail. Moreover, if only a small part of the object is textured, it will be detected but there may be a substantial error in the pose estimation. Also, this method does not work with transparent objects.

Both textureless and transparent objects such as cups, dishes, staplers etc. are an essential part of home and office environment. The problem of estimating the pose of such objects is important for personal robotics. While recent developments in structured light sensors such as Kinect shows promising results in finding the pose of textureless objects, this type of technology does not work with specular and transparent surfaces. Our work in this paper is largely influenced by the methods for textureless objects coming from industrial robotics [2] that use a CAD model of an object to estimate its pose from a monocular camera by projecting the model to a test image and comparing object features with image edges. While CAD models of manipulated objects in industrial settings are usually available anyway, CAD models of all objects in the personal space are hard to capture.

We present an algorithm for refining 6-DOF pose of a transparent object using edge features. The method does not require a CAD-model, it needs a 3D scan of an object including a point cloud and images registered to each other. We show that the method can be used for accurate pose estimation of transparent rigid objects.

2. RELATED WORK

Transparent objects are very challenging objects in computer vision because their appearance in an image largely depends on a background. Also it is hard to capture a 3D model or a point cloud for transparent objects due to limitations in technologies of existing 3D sensors and because reconstruction of transparent objects is still a very hard problem [3].

The algorithm for detection and reconstruction of unknown transparent objects was proposed in [4]. The algorithm uses two views of a test scene captured by a ToF camera. The algorithm is insensitive to changes in illumination and it was applied for grasping of isolated transparent objects by a robot. Grasping was successful in 41% of reconstructed objects and failed attempts are explained by errors in objects reconstruction and pose estimation.

The algorithm for pose estimation of transparent objects from two views of a test scene was proposed in [5]. Accurate pose estimation was achieved but the objects are required to stay on a table plane and they should be separated from each other. So the algorithm is not able to estimate 6-DOF pose.

Kinect sensor is used for pose estimation and recognition of transparent objects in [6]. However, results are reported only in case when objects are assumed to stay on a table plane. So accuracy in case of 6-DOF pose estimation is unclear.

Specularities are important features when working with transparent objects and there are very promising approaches to pose estimation [7, 8, 9] using this cue. However, these algorithms of pose estimation require a triangulated mesh or a CAD model of an object and they were evaluated with textureless objects only.

Texture features like SIFT are not suitable when working with textureless and transparent objects because such objects don't have their own texture. Computer vision research [10] and psychological studies [11] show that edges and contours of objects are important features and they can be used successfully for the object recognition problem. For example, humans can recognize objects from rough pencil sketches although texture is missing. This cue is available both for transparent and textureless objects and it makes the problem of pose estimation of transparent objects related to pose estimation of textureless objects.

The problem of untextured pose estimation has a long history in computer vision. See [12] for a detailed overview of the 2D-3D pose estimation problem. [13] shows that it is possible to estimate a pose of a textureless object by using single-view object detection algorithms. However the 2D object representation used in this method is viewpoint-dependent, so a set of detectors has to be trained for different viewpoints. Running all detectors is infeasible in the general case so pose clustering is used [14, 13, 15], first to make a rough estimation of the pose and then refine it by running a smaller set of detectors. The pose corresponding to the most confident detector is returned as an estimation of the object pose. But the accuracy of this estimation is bounded by the number of detectors that also defines the computational cost.

General multi-view approaches and a 3D model of an object are required to balance between the computational cost and the pose estimation accuracy. The idea to use a 3D representation of an object for recognition is going back to early computer vision of 70's and 80's, see, for example, [16]. Approaches [17, 2, 18] utilize this idea and they can estimate a pose of a textureless object quite accurately. Algorithms [17, 18] find the closest training pose and run a local optimization of it using a CAD model of an object. High-quality CAD-models are hard to obtain and although there are some CAD-models of typical household objects (like a cup or a bottle), models are not available for all specific objects that robots need to grasp in a household environment.

Our approach to pose refinement step is similar to [17, 18] and also based on edges cue. However, it does not require a CAD model and it is able to estimate 6-DOF pose of transparent objects.

3. PROPOSED APPROACH

To solve the considered problem we divide it to following tasks:

1. Create a 3D model which allows to generate object edgels (points on edges) for different poses. Our model contains a 3D object model and a 3D edge model. The 3D object model is a point cloud of the whole object and it is used to generate silhouette edges. The 3D edge model is a point cloud with points on surface edges, that is edges created by depth discontinuities or texture.

2. Determine a cost function which estimates dissimilarity between generated edgels and the observed test data and then minimize the cost function by varying parameters that determine pose of the object.

We will address all of these steps in the following subsections.

3.1 Creation of the 3D model

There are no stable ways to estimate depth or produce point clouds for transparent objects [3]. So we take a copy of the object, paint it with a color and use the painted object in the model creation pipeline.

The 3D object model is created automatically from the train data. We scan each object on a planar surface with a Kinect device. Two fiducial markers consisting of grids of circles are placed in the field of view to provide accurate registration of frames. Depth map from Kinect allows us to segment the plane and calculate the object mask in each image.

We illustrate the algorithm of the surface edge model creation using a textureless object that has many surface edges (Fig. 1). First, we extract 3D points that correspond to surface edges in each frame, then we register point clouds from different frames, and, finally, we build a surface edge model.

Detecting edges in each frame

1. Find edges on each image of the object using Canny edge detector . Then find edges of the object by intersecting the detected edges with the object mask.

2. Select the points from the 3D cloud that correspond to image edges. Our point cloud is interpolated to the size of the train image and so there is a bijection between 3D points and image pixels. As a result we get a 3D edge model for each training image.

Registering point clouds

1. Transform all models to the same coordinate system associated with the first frame, using the poses from the fiducial markers. The corresponding points from different frames would coincide with each other in the ideal case but there are always some deviations in practice due to noise (see the Fig. 2A).

2. Register transformed point clouds. There is a classic and widely used algorithm Iterative Closest Point (ICP) for registration of two point clouds [19, 20]. Global approaches like [21] are used for registration of multiple point clouds because they can distribute registration error between all point clouds evenly. We have a good initial alignment of point clouds using the poses from the fiducial markers so we have used more simple global algorithm [22] with LM-ICP [23] to register pairs of point clouds.

Creating a surface edge model

1. Partition all transformed points into groups where each group corresponds to the same point of the object. This allows to get more accurate coordinates of the object point by its noised observations. Partitioning is done by solving the problem of k-partite matching which is a generalization of the bipartite matching for the case of k-partite graphs. It is known to be an NP-hard problem [24] so we used a heuristic algorithm based on [25].

2. For each group compute accurate coordinates of the model point using robust estimation of location [26] e.g. the minimum covariance determinant estimator (MCD) [27]. The constructed model is given in the Fig. 2B and it represents edges of the object much better than transformed point clouds in Fig. 2A.

3. Downsample the constructed 3D edge model. The 3D edge model of the whole object is given in the Fig. 2C. It contains many close points that don't give additional information. So we keep 10% of points to lower computational costs of further processing. It is done by a trivial adaptation of the Douglas-Peucker algorithm [28] for this task. The downsampled model is given in the Fig. 2D.

It is important to note that as a result of the k-partite matching the silhouette edges, which presence depends on a point of view, will be automatically filtered out as they will not have correspondences in different frames.

We group all surface edge points into contours by proximity. 3D orientation of a contour at a point can be estimated as direction of the tangent vector to the 3D contour at this point. We do this by generalizing [29] to the 3D case by means of multi-dimensional robust statistics [26]. When the model is transformed in 3D space, orientations of points are transformed as usual 3D points. We use the contours to calculate the orientation in each projected edgel that can be used in the cost function.

The algorithm for constructing a silhouette model is similar. We register dense point clouds that we obtain from a Kinect by using the same algorithm (ICP registration with the initial pose from the fiducial markers). The coordinate system origin is placed into the mass center of the joint point cloud.

In order to guarantee that poses that are close to each other are produced by close rotation and translation vectors, we place the coordinate system origin into the center of mass for each of the objects.

3.2 The cost function

The cost function is defined by comparing detected test image edges with projections of 3D surface and silhouette edges that depend on the object pose. Given a rotation and translation of the object, we transform the point cloud into the test camera reference frame. Surface edges are projected into the image and they give us 2D surface edges because transparent objects don't have self-occlusions. In order to get silhouette edges, we project a dense point cloud into a test image, apply several closing operations to the resulting set of pixels and find the borders of the connected components. These borders constitute silhouette edges.

Now we want to construct a cost function that compares two sets of edges in an image. Let $E = \{e_j\}$ be a set of pixels that belong to edges of a test image, $T = \{t_i\}$ is a set of the model points projected into the image plane. One of the most popular cost functions is Chamfer Matching (CM):

$$d_{CM}(E,T) = \frac{1}{|T|}\sum_{t_i \in T} \min_{e_j \in E} ||t_i - e_j||, \qquad (1)$$

where $||\cdot||$ is Eucledian norm. However, mean is not a robust statistic because a single outlier can affect the final value severely. Edge detection is an unstable operation that produces a lot of variation, especially in edge endpoints. In order to overcome this

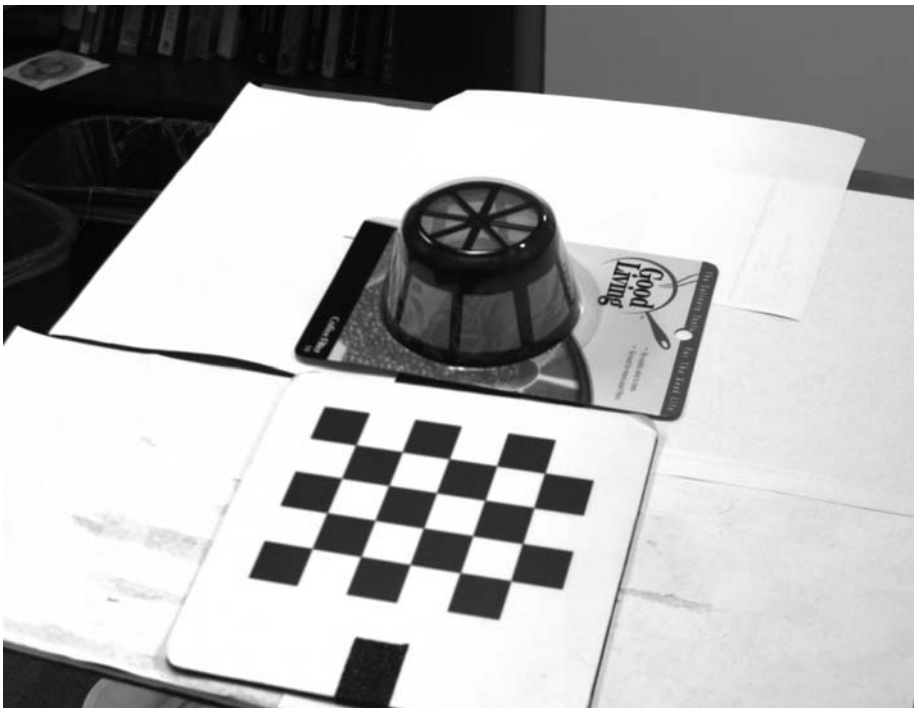

Figure 1: Example of an object to be modeled.

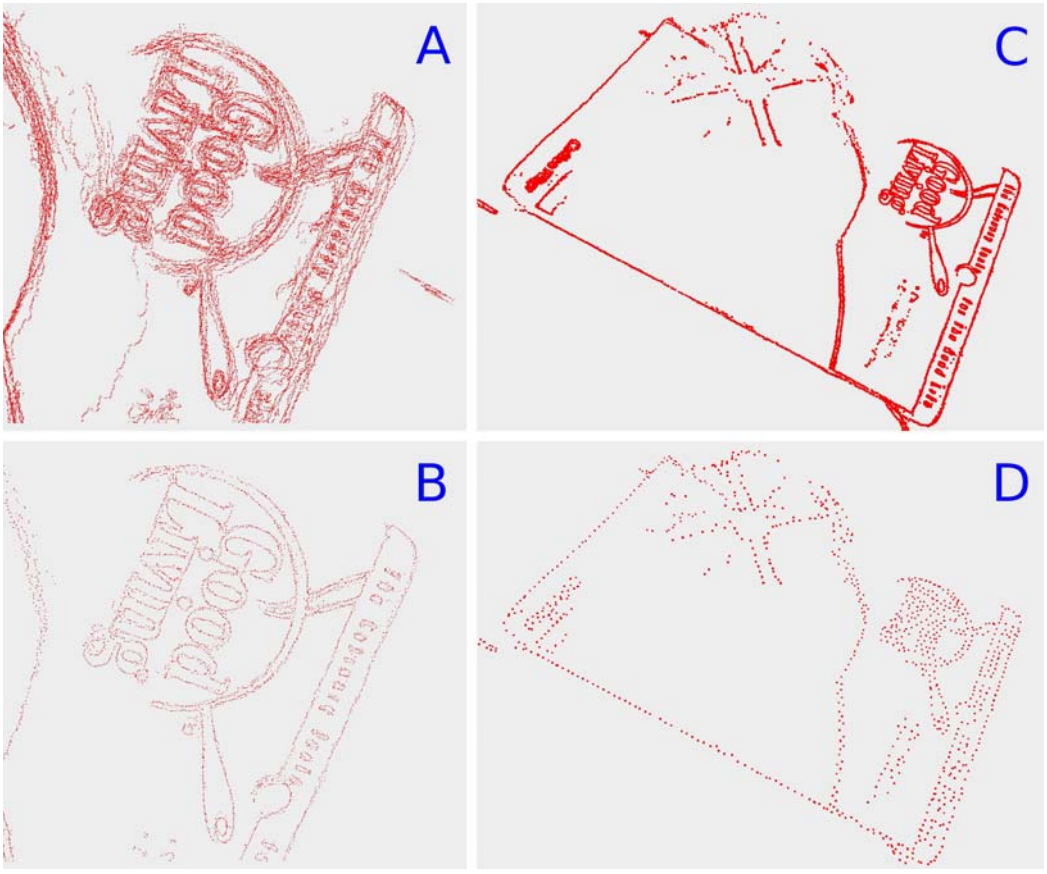

Figure 2: Creation of the surface edge model. (A) All train point clouds transformed to the same coordinate system. (B, C) Refined and denoised point cloud with k-partite matching and robust statistics. (D) Downsampled point cloud which approximates the full model well.

issue we use the Partial Directed Hausdorff (PDH) distance [30]:

$$d_H(E,T) = K^{th}_{t_i \in T} \min_{e_j \in E} ||t_i - e_j||. \quad (2)$$

Here $K^{th}_{t_i \in T}(\mathbf{X})$ is the Kth ranked value in the sorted set $\mathbf{X}$. Throughout the paper we use $K = 0.8|T|$, where $|T|$ is the number of elements in T. However, this distance can be set to zero by placing the object infinitely far away from the camera. It means the global minimum will be achieved in the incorrect pose for this distance. So we introduce a Normalized PDH (NPDH) distance:

$$d_H(E,T) = \frac{1}{\sqrt{\det C}} K^{th}_{t_i \in T} \min_{e_j \in E} ||t_i - e_j||, \quad (3)$$

where C is the covariance matrix of the projections of the point cloud into the image.

Both CM and PDH distances are known to behave incorrectly in clutter. Oriented Chamfer Matching (OCM) [10] is known to handle clutter better. However, it is more computationally expensive, so we use NPDH throughout the paper.

The PDH cost function is computed separately for surface and silhouette edgels. The resulting distances are added with different weights: $2/3$ for surface and $1/3$ for silhouette edges. The weight for surface edges is higher because surface edges are more stable: they are constructed by fusing edgels from many training frames, so we know that each surface edge is found robustly by the edge detector.

The cost function 3 is minimized by the global optimization algorithm DIRECT [31] from the NLopt library [32] by varying the 6 parameters defining pose of the object: a translation and rotation vectors.

4. EXPERIMENTAL RESULTS

The algorithm was tested on the base of 5 transparent objects. We take 5 pairs of kitchen items and paint one object in each pair in white color to make it opaque because there are no reliable way to scan a transparent object [3]. We use the painted object to scan it with Kinect to create object models. Each training sequence contains 12 frames with different poses of the table relative to Kinect. The asymmetric circles pattern from the OpenCV library was used as the fiducial marker to estimate poses between frames. To create test data we used the corresponding transparent objects captured by a calibrated stereo pair of Canon EOS 40D cameras from a distance about 1 meter. Images were resized to resolution of about one megapixel (1166x778). Each test object is placed exactly as the corresponding training object relatively to the fiducial marker, so we know the ground truth.

The objective of these experiments it to investigate how accurate the initial guess about the object pose should be for the algorithm to produce a stable correct result. We ran the algorithm with many different initial guesses generated randomly. In particular, the correct pose was translated in random direction on the specified distance d and rotated in random direction on the specified angle α. Each experiment with specific values of d and α was repeated 50 times and we take 27 different combinations of these values. All objects in the test base have rotation symmetry and this was taken into account when evaluating pose returned by the algorithm but this knowledge was not used by the algorithm itself.

The example of the results is given in Fig. 3. Points of objects' models are colored. They are projected into the image plane using initial hypothesis of objects' poses and poses refined by the algorithm. Initial poses are quite far away from correct poses. However, final poses are accurate enough for grasping.

We run the algorithm on all 5 objects to see how often the algorithm returns a correct pose. We consider the pose estimation successful if the difference between the returned and correct poses is less than 2 cm in translation and 10 degrees in rotation.

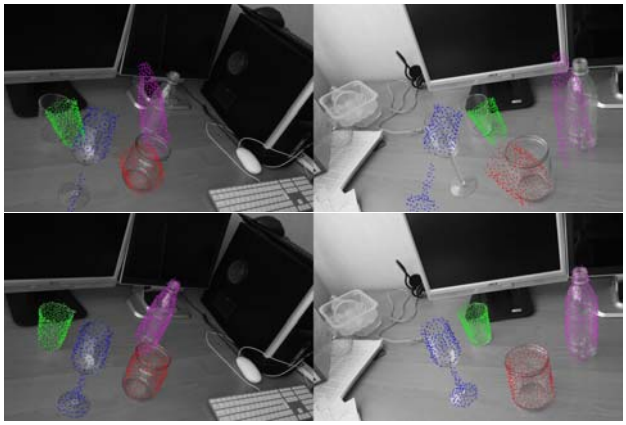

Figure 3: Images from a stereo pair with the projected poses found with the algorithm, initial (upper row) and refined (bottom row).

Fig. 4 shows the statistics for all 5 objects. The percent of runs when the algorithm succeeded is plotted on the y-axis. The chart shows that if the initial translation error is less than 2 cm, we can successfully reconstruct the pose in more than 80% of the cases. Black area in Kinect depth map that corresponds to specular and transparent surfaces can give us a hypothesis about the object location. This information can be used to generate a good initial guess about the translation vector. If the initial error of the translation vector is 2cm, the rate of successful reconstructions (averaged over all angles) is 88%, if the initial translation error is 5cm, then the rate of successful reconstructions is 77%. Note that part of the error comes from poses that are upside-down to the ground truth: since many objects are close to cylindrical shape, the final result can put the top of the glass to the bottom.

See more examples at Fig. 5. Also see Fig. 6 for example of the algorithm failure. The algorithm returned the pose which is upside down of correct one because the object has nearly cylindrical shape.

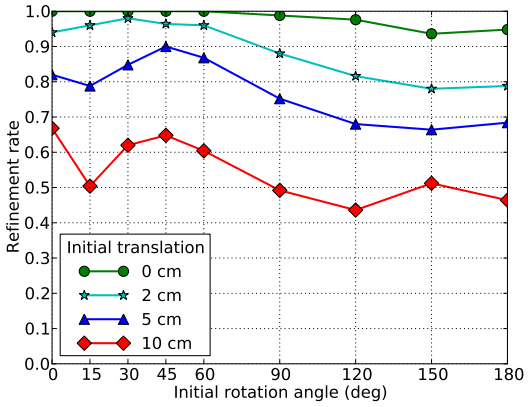

Figure 4: Statistics of the algorithm working on rigid transparent objects. Pose can be refined successfully if an initial pose is not very far from the correct pose. The algorithm is robust to incorrect initial rotation but it is more sensitive to initial translation.

The proposed algorithm can refine poses of transparent objects in some cases, but it has several limitations. The approach demands good initial hypothesis of the object pose, otherwise the search for the global minimum takes too much time. The algorithm is unstable in clutter e.g. if the object is surrounded by other objects. But in the case of low clutter the algorithm works with

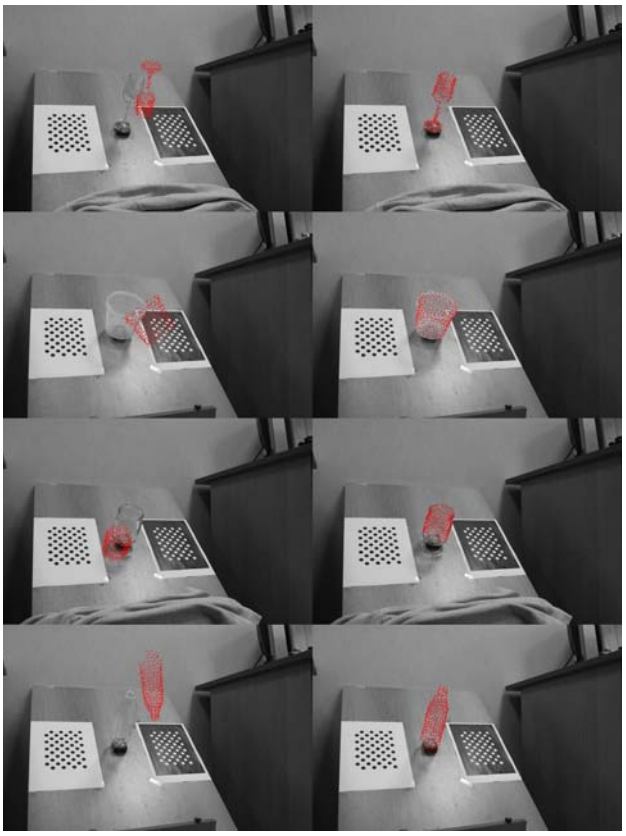

Figure 5: Examples of successful pose refinement for different objects. Left images are initial poses and right images are refined poses. Only one image from the stereo pair is shown.

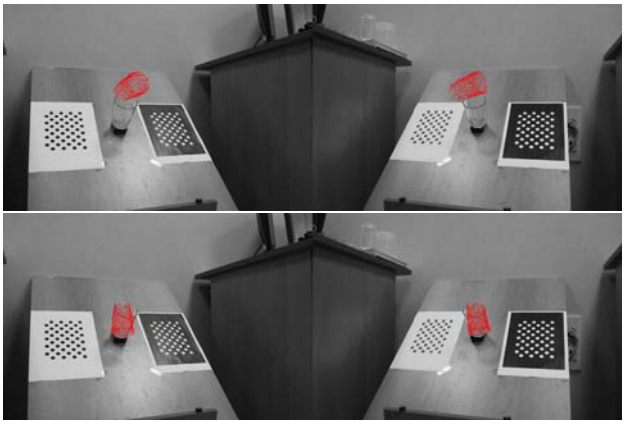

Figure 6: Example of the algorithm failure due to cylindrical shape of the object. Two images from a stereo pair are shown: initial pose (upper row) and refined pose (lower row).

sufficient speed and quality to be applied for pose refinement of rigid transparent objects.

Another limitation of the proposed method is using a calibrated stereo pair for generating test images instead of a single monocular camera. The main obstacle for a monocular camera is ambiguity that cannot be resolved from a single image without additional assumptions or priors.

There exist significantly different poses that have very good projections to a test image and it is specificity of transparent objects. For example, two different poses of an opaque object are shown in the Fig. 7 and there is no ambiguity between them. However, if the same object is transparent then there are two different plausible interpretations of the same projection (Fig. 8) because transparent objects don't have self-occlusions and all edges are visible.

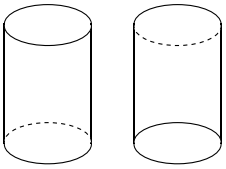

Figure 7: Two different poses of an opaque object. There is no ambiguity between them because different edges are visible in different poses.

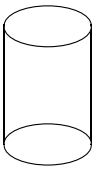

Figure 8: Ambiguous projection of a transparent object. Two plausible poses of the object are possible because all edges are visible on the same image.

We evaluated the algorithm with a monocular camera on the same dataset using only left images of our stereo test set. The statistics of pose estimation is shown in the Fig. 9. One can see that there is a significant degradation of accuracy compared to the stereo case.

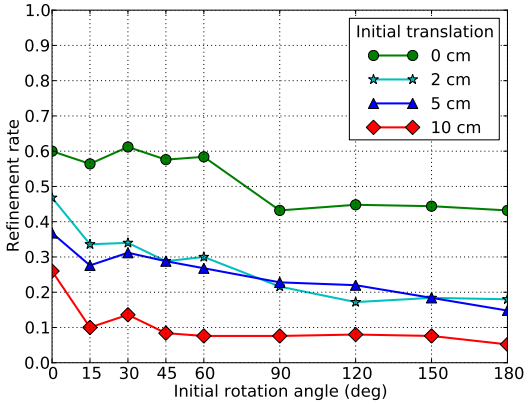

Figure 9: Statistics of the algorithm working when using a monocular camera only. The results degrade significantly comparing to the stereo camera due to inherent ambiguity of pose estimation of transparent objects from a single view.

5. CONCLUSION

The paper presents the algorithm for refining the 6-DOF pose of transparent objects. Our method only requires a calibrated stereo pair during the online stage. Given an initial estimate that has an error in translation less than 5cm, the rate of accurate pose estimations is higher than 75%. The method allows to grasp transparent objects without using expensive sensors such as TOF cameras.

6. REFERENCES

[1] A. Collet, D. Berenson, S.S. Srinivasa, and D. Ferguson, "Object recognition and full pose registration from a single image for robotic manipulation," in *IEEE International Conference on Robotics and Automation*, 2009.

[2] B. Rosenhahn, T. Brox, and J. Weickert, "Three-dimensional shape knowledge for joint image segmentation and pose tracking," *International Journal of Computer Vision*, vol. 73, no. 3, pp. 243–262, 2007.

[3] Ivo Ihrke, Kiriakos N. Kutulakos, Hendrik P. A. Lensch, Marcus Magnor, and Wolfgang Heidrich, "State of the Art in Transparent and Specular Object Reconstruction," in *STAR Proceedings of Eurographics*, 2008, pp. 87–108.

[4] U. Klank, D. Carton, and M. Beetz, "Transparent Object Detection and Reconstruction on a Mobile Platform," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011.

[5] C.J. Phillips, K.G. Derpanis, and K. Daniilidis, "A Novel Stereoscopic Cue for Figure-Ground Segregation of Semi-Transparent Objects," in *1st IEEE Workshop on Challenges and Opportunities in Robot Perception*, 2011.

[6] I. Lysenkov, V. Eruhimov, and G. Bradski, "Recognition and pose estimation of rigid transparent objects with a kinect sensor," in *Robotics: Science and Systems Conference*, 2012.

[7] P. Lagger, M. Salzmann, V. Lepetit, and P. Fua, "3d pose refinement from reflections," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008.

[8] J.Y. Chang, R. Raskar, and A. Agrawal, "3d pose estimation and segmentation using specular cues," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009.

[9] A. Netz and M. Osadchy, "Using specular highlights as pose invariant features for 2d-3d pose estimation," in *Computer Vision and Pattern Recognition, 2011. CVPR 2011. IEEE Conference on*. IEEE, 2011.

[10] J. Shotton, A. Blake, and R. Cipolla, "Multiscale categorical object recognition using contour fragments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1270–1281, 2007.

[11] I. Biederman and G. Ju, "Surface versus edge-based determinants of visual recognition," *Cognitive Psychology*, vol. 20, no. 1, pp. 38–64, 1988.

[12] B. Rosenhahn, *Pose estimation revisited*, Ph.D. thesis, Universität Kiel, Sept. 2003.

[13] S. Hinterstoisser, V. Lepetit, S. Ilic, P. Fua, and N. Navab, "Dominant orientation templates for real-time detection of texture-less objects," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 2257–2264.

[14] D.M. Gavrila, "A bayesian, exemplar-based approach to hierarchical shape matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1408–1421, 2007.

[15] C. Reinbacher, M. Ruther, and H. Bischof, "Pose estimation of known objects by efficient silhouette matching," in *2010 International Conference on Pattern Recognition*. IEEE, 2010, pp. 1080–1083.

[16] D.G. Lowe, "Three-dimensional object recognition from single two-dimensional images," *Artificial intelligence*, vol. 31, no. 3, pp. 355–395, 1987.

[17] M.Y. Liu, O. Tuzel, A. Veeraraghavan, R. Chellappa, A. Agrawal, and H. Okuda, "Pose estimation in heavy clutter using a multi-flash camera," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2010.

[18] M. Ulrich, C. Wiedemann, and C. Steger, "CAD-based recognition of 3d objects in monocular images," in *International Conference on Robotics and Automation*, 2009, vol. 1191, p. 1198.

[19] P.J. Besl and N.D. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1992.

[20] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *International Journal of Computer Vision*, vol. 13, no. 2, pp. 119–152, 1994.

[21] K. Pulli, "Multiview registration for large data sets," in *3-D Digital Imaging and Modeling, 1999. Proceedings. Second International Conference on*. IEEE, 1999, pp. 160–168.

[22] R. Bergevin, M. Soucy, H. Gagnon, and D. Laurendeau, "Towards a general multi-view registration technique," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 18, no. 5, pp. 540–547, 1996.

[23] A.W. Fitzgibbon, "Robust registration of 2D and 3D point sets," *Image and Vision Computing*, 2003.

[24] E. Hazan, S. Safra, and O. Schwartz, "On the hardness of approximating k-dimensional matching," in *Electronic Colloquium on Computational Complexity, TR03-020*, 2003.

[25] R. Singh, J. Xu, and B. Berger, "Global alignment of multiple protein interaction networks," in *Proc. Pacific Symp. Biocomputing*. Citeseer, 2008, vol. 13, pp. 303–314.

[26] M. Hubert, P.J. Rousseeuw, and S. Van Aelst, "High-breakdown robust multivariate methods," *Statistical Science*, vol. 23, no. 1, pp. 92–119, 2008.

[27] P.J. Rousseeuw, "Multivariate estimation with high breakdown point," *Mathematical statistics and applications*, vol. 8, pp. 283–297, 1985.

[28] D.H. Douglas and T.K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica: The International Journal for Geographic Information and Geovisualization*, vol. 10, no. 2, pp. 112–122, 1973.

[29] J. Matas, Z. Shao, and J. Kittler, "Estimation of curvature and tangent direction by median filtered differencing," in *Image Analysis and Processing*. Springer, 1995, pp. 83–88.

[30] D.P. Huttenlocher, G.A. Klanderman, and WA Rucklidge, "Comparing images using the hausdorff distance," *IEEE Transactions on pattern analysis and machine intelligence*, pp. 850–863, 1993.

[31] D.R. Jones, C.D. Perttunen, and B.E. Stuckman, "Lipschitzian optimization without the Lipschitz constant," *Journal of Optimization Theory and Applications*, vol. 79, no. 1, pp. 157–181, 1993.

[32] Steven G. Johnson, "The nlopt nonlinear-optimization package," http://ab-initio.mit.edu/nlopt.