Recognition and Pose Estimation of Rigid Transparent Objects with a Kinect Sensor

Ilya Lysenkov
Itseez
ilya.lysenkov@itseez.com

Victor Eruhimov
Itseez
victor.eruhimov@itseez.com

Gary Bradski
Willow Garage
gary@industrial-perception.com

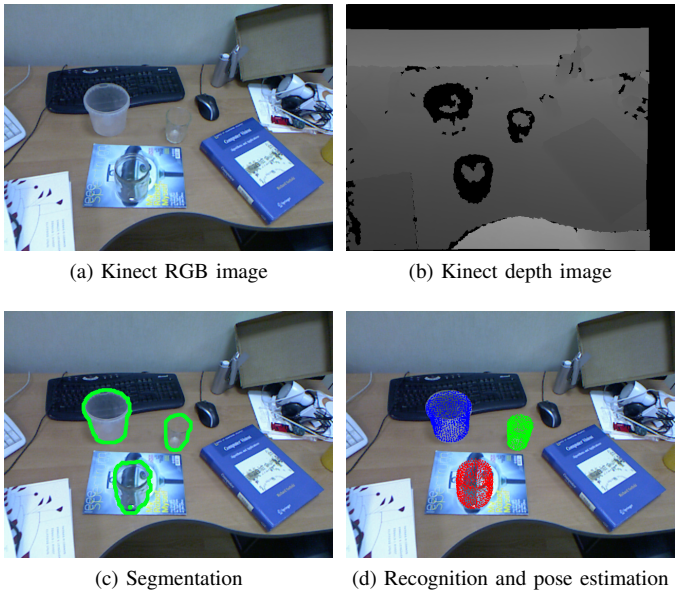
***Abstract*—Recognizing and determining the 6DOF pose of transparent objects is necessary in order for robots to manipulate such objects. However, it is a challenging problem for computer vision. We propose new algorithms for segmentation, pose estimation and recognition of transparent objects from a single RGB-D image from a Kinect sensor. Kinect's weakness in the perception of transparent objects is exploited in their segmentation. Following segmentation, edge fitting is used for recognition and pose estimation. A 3D model of the object is created automatically during training and it is required for pose estimation and recognition.**

The algorithm is evaluated in different conditions of a domestic environment within the framework of a robotic grasping pipeline where it demonstrates high grasping success rates compared to the state-of-the-art results. The method doesn't deal with occlusions and overlapping transparent objects currently but it is robust against non-transparent clutter.

I. Introduction

Transparent objects are a common part of domestic and industrial environments. Recognition and 6 degree of freedom (6DOF) pose estimation of objects is required for robotic grasping and manipulation. However, transparent objects are very challenging for robot perception with RGB images as well as with modern 3D sensors.

- 2D Computer Vision. The appearance of a transparent object strongly depends on its background and lighting. Transparent objects usually don't have their own texture features, their edges are typically weak and intensity gradient features are heavily influenced by see through background clutter as well as specularity edges induced by lighting. So classical computer vision algorithms for recognition and pose estimation are difficult to apply to transparent objects.
- 3D Computer Vision. 3D point clouds are successfully used for object recognition and pose estimation (Hinterstoisser et al. [10]). However, modern sensors (Kinect, ToF cameras, stereo cameras, laser scanners) can't estimate depth reliably and produce point clouds for transparent and specular objects so these algorithms can not be applied. Cross-modal stereo can be used to get depth estimation on transparent objects with Kinect (Chiu et al. [4]) but its quality is far from estimation on Lambertian objects. Acquisition of 3D data and reconstruction of transparent objects is a challenging and unsolved problem (Ihrke et al. [11], Mériaudeau et al. [18]).



(a) Kinect RGB image (b) Kinect depth image

(c) Segmentation (d) Recognition and pose estimation

Fig. 1. Example of test data with 3 transparent objects with segmentation, recognition and pose estimation results using the proposed algorithms.

We address these challenges and propose an algorithm for segmentation, pose estimation and recognition of transparent objects. Unknown transparent objects are segmented from a single image of a Kinect sensor by exploiting its failures on specular surfaces. 3D models of objects created at the training stage are fitted to extracted edges. A cost function value is used to make a decision about an instance of the object and determine its 6DOF pose relative to the robot.

The proposed algorithm was integrated into the object recognition stack (Rublee et al. [25]) and connected with the robotic grasping pipeline (Ciocarlie [5]) in the Robot Operating System (ROS, Quigley and WillowGarage [23]) and evaluated on a Willow Garage's PR2 robot. The grasping is robust to non-transparent clutter and success rate is over 80%.

II. Related work

Transparent objects and the challenges they bring in perception were rarely addressed in research papers until recently. McHenry et al. [17] derive features from inherent properties of transparent objects (transparency, reflection and refraction). Usually these properties are obstacles for computer vision

algorithms and features like specular highlights are considered as noise. The situation is different for transparent objects. These features are characteristic of transparency and so they are used to segment unknown transparent objects from a single RGB image by McHenry et al. [17], McHenry and Ponce [16]. This problem is also addressed by Kompella and Sturm [13] in a robotic context of transparent obstacles detection. However, these approaches require that background behind a transparent object is available as foreground on the same image. In this case correspondence between a foreground region and its distorted image behind the transparent object can be established and these correspondences are used to segment the transparent object. These counterpart regions should be large enough to estimate such features as blurring, texture distortion and other transparent overlay effects. The algorithm proposed in this paper allow segmentation even if it is not the case and background is different from foreground.

Lei et al. [14] use light detection and ranging (LIDAR) data to segment unknown transparent objects: highlight spots from a RGB image are used to find candidate areas of transparent objects and then the GrabCut segmentation algorithm (Rother et al. [24]) is applied to a depth image and a laser reflectance intensity image to compute the final segmentation. The experimental setup consist of a 2D LIDAR device, a registered RGB camera and a pan-tilt unit so it is more complex and expensive than Kinect used in the current paper.

Osadchy et al. [21] utilize specular highlights of transparent objects in the recognition problem. Successful recognition of 9 transparent objects was demonstrated with this single feature. Unfortunately, a test scene must have a dominant light source with known position so applicability of the approach is limited.

Fritz et al. [8] use an additive model of latent factors to learn appearance of transparent objects and remove influence of a background behind them. It allows recognition of transparent objects in varying backgrounds. The algorithm was evaluated on the problem of transparent object detection in challenging conditions of domestic environment with complex backgrounds using a dataset of 4 transparent objects.

Klank et al. [12] detect unknown transparent objects and reconstruct them using two views of a test scene from a ToF camera. The problem is challenging especially because no training is involved and a 3D model of an object is reconstructed directly from test data. The algorithm is tolerant to difficult lighting conditions thanks to data from a ToF camera. Transparent objects are assumed to stay on a flat surface but the algorithm is tolerant to violations of this assumption. The algorithm is not misled by opaque objects and distinguishes them from transparent objects correctly. The algorithm was evaluated on a dataset of 5 transparent objects with uniform backgrounds behind them. The objects were placed on a dark table separately from each other. Transparent objects were reconstructed successfully in 55% of 105 attempts. Also transparent object grasping with a robot was evaluated. It was successful in 41% of the cases when reconstruction was successful so the robot grasped 23% of all transparent objects overall.

Phillips et al. [22] propose to use inverse perspective mapping for detection and pose estimation of transparent objects. This cue can be computed if two views of a test scene are available and objects stay on a support plane. The algorithm was evaluated on a dataset of 5 transparent objects that were isolated from each other in a test scene. Poses of the objects are estimated with the accuracy of 2.7-11.1 millimeters with the mean error of 7.8 millimeters. However, pose is estimated with exhaustive coarse search in 2D space of poses on a table (with refinement of the best candidate) so this is not scalable to 6-DOF pose estimation. The approach was demonstrated to be applicable to the detection problem.

III. Proposed approach

We consider problems of segmentation, pose estimation, recognition and grasping of transparent objects in this paper. Our main focus is accurate pose estimation of transparent objects which is used to enable robotic grasping. Previous works by Klank et al. [12], Phillips et al. [22] are the most relevant here because other papers don't deal with pose estimation of transparent objects. These works require two views of a test scene, in contrary we use a single image of Kinect at the test stage to recognize and estimate pose of transparent objects. Kinect uses structured light and projects a IR-pattern on objects to estimate their depth. However, such a technique doesn't work with specular, refractive, translucent objects and objects with very low surface albedo (Ihrke et al. [11]). So Kinect can't produce point clouds of transparent objects but we take advantage of this fact. The regions where Kinect fails to estimate depth are likely to be produced by specular or transparent surfaces and we use invalid areas in a Kinect depth map as an interest area operator. Using this cue and a corresponding RGB image, we segment transparent objects from background and extract their silhouettes. 3D models of transparent objects are created during a training stage by painting objects with color and scanning them. These learned models are fitted to the extracted RGB silhouettes on the test image by varying poses of the object. We treated this as an optimization problem with regard to 6 parameters that define a 3D pose of a rigid object relative to the robot's camera. This multi-dimensional optimization problem is decomposed into several parts. Some of them have closed-form solutions and others can be solved with standard iterative algorithms like Levenberg-Marquardt. The inherent ambiguity in pose estimation of transparent objects is discussed. This ambiguity is resolved with a support plane assumption and the correct pose is returned which can be used for robotic grasping.

Our main contributions are:

- A model of transparent objects that takes into account both silhouette and surface edges.
- An algorithm for 6DOF pose estimation of transparent objects. It is based on existing CAD-based approaches to pose estimation (Ulrich et al. [27], Liu et al. [15]) but our algorithm doesn't need a CAD model and utilizes the specifics of transparent objects to meet practical requirements in performance and accuracy.

- A complete system for grasping transparent objects with Kinect as input, going from the model capture up to segmentation, pose estimation and grasping.

The proposed algorithm was evaluated on 13 transparent objects in different conditions. As stated above, our algorithm was integrated into a robotic grasping pipeline and evaluated on a PR2 robot. Grasping was successful in 80% of cases and this result is robust even to challenging complex backgrounds behind the transparent objects.

IV. Training stage

Our method creates a 3D model of an object that allows us to generate edgels (pixels belonging to edges) in a 2D image given the object's 6DOF pose. Transparent as well as any untextured objects create two types of edges: surface and silhouette edges. A surface edge is produced by a discontinuity in surface orientation. Image edgels corresponding to a surface edge are made visible by lighting discontinuities. A silhouette edge is produced by the object border and the corresponding image edgels are made visible by both lighting and contrast with the background. Examples of surface edges are edges where the stem joins the rest of the glass and the back edge of the cup lip (Fig. 5), while silhouette edges are side edges of the glass. These two types of edges have a different dependence on the object pose. However, they are equally important for the pose estimation, so we have to take into account both.

We define the object model that contains a silhouette model and a surface edge model. The silhouette model is a 3D point cloud of the whole object surface and is used to generate silhouette edges. The surface edge model is a set of 3D points corresponding to surface curvature discontinuities.

The creation of a silhouette model requires 3D scanning of a transparent object. However, there are no robust algorithms to do this (Ihrke et al. [11]). One possible solution is to create the model manually but that is a time-consuming process. Another approach is to find and download a similar 3D model from the Internet as done by Phillips et al. [22] for 5 transparent objects. Unfortunately, existing databases of 3D models are limited and problems arise if there are no similar models available. Finally, one can cover transparent objects with powder or paint and use standard scanning algorithms on these now Lambertian objects as done by Osadchy et al. [21]. We use the last approach because it allows us to create high-quality models. We paint a transparent object white and use KinectFusion (Newcombe et al. [19]) to compute a clean point cloud of the object. The KinectFusion algorithm creates a single point cloud of a test scene by registering point clouds from different viewpoints. A modeled object was put on a transparent stand on a table to enable easy 3D segmentation of the model as a cluster of points above the table plane. KinectFusion does not use 2D photometric data and so relies on tracking/registering to 3D structures only. To ensure good 3D registration, we placed a 3D calibration rig near the object to be modeled. This allows robust and precise camera tracking that results in accurate 3D models. One of the created models is shown in the Fig. 2.

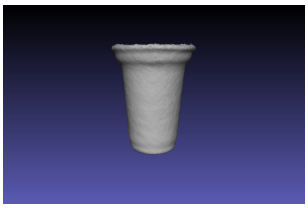


Fig. 2. A 3D model of a transparent object created with KinectFusion at the training stage. The transparent object was sprayed white so that a Kinect sensor could be used to estimate surface depth.

The surface edge model is created from the silhouette model. All points of the silhouette model are projected on the training images. Points that are often projected on Canny edges are used to form the surface edge model.

The models are used to compute edges of the object for given poses. Points of the silhouette model are projected onto an image for a given pose and then morphological operations are applied to get a single blob. The contour of this blob is the silhouette of the object. Surface edges are computed by projecting the surface edge model. Objects are transparent so there are no self-occlusions and we get surface edges directly.

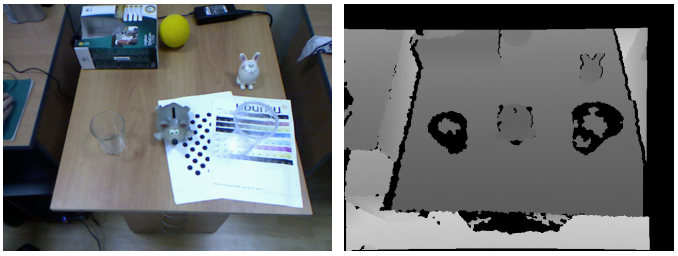
Template-based approaches are popular in computer vision, a recent example is Hinterstoisser et al. [10]. Templates are created for several poses by sampling views from a viewing sphere of the object. These templates are matched to a test image and a full pose is estimated. Standard template matching algorithms are not suitable for transparent objects because these objects have weak gradients and they can be severely disturbed by gradients of background behind the object. These algorithms degrade significantly in such cases and should be improved by using a depth map (Hinterstoisser et al. [10]) but it is not available for transparent objects.

We adopt a template-based approach for pose estimation of transparent objects. Pose estimation of a rigid object requires estimation of 6 parameters: 3 for rotation and 3 for translation. A rotation matrix can be decomposed into a superposition of three rotations around coordinate axes: $R = R_z R_x R_y$, where we consider fixed coordinate axes. An optical axis is z and the upright direction of the object is aligned with y-axis for the identity transformation. We will find translation parameters and R_z at the testing stage. So, we create templates of the object for different R_y and R_x during the training stage. We sample possible rotations R_y and R_x with some discrete steps and for each step, we project our model to get a silhouette of the object for this pose. The y-axis corresponds to the upright direction, many transparent objects common in the household environment (plates, glasses) are rotationally symmetric around this direction. So the algorithm checks to see if an object has rotation symmetry around the y-axis and templates are sampled for R_x only if it is the case. 100 templates were sampled for non-symmetric objects and 10 for symmetric objects when evaluating the algorithm.

As a result, we get a 3D model of the object and silhouettes for possible rotations R_y and R_x after the training stage.

V. Testing stage

A Kinect RGB image of two transparent objects is shown in the Fig. 3a. Kinect has a low quality RGB camera and it is difficult to detect and estimate pose from such images robustly, especially when background is not uniform. However, in the Fig. 3b a corresponding Kinect depth map is shown where invalid depth, that is regions where Kinect fails to estimate depth, is colored by black. Kinect is not able to estimate depth on transparent objects but we can take advantage of this fact: regions where Kinect doesn't work are likely to belong to transparent objects. So this cue can be used to segment transparent objects on an image.



(a) Kinect RGB image (b) Kinect depth image

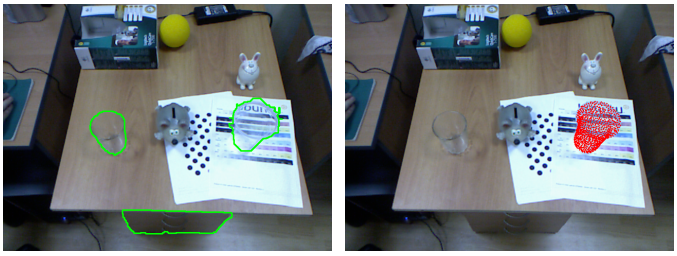
Fig. 3. Kinect images of two transparent objects. It is hard to detect transparent objects from the RGB image but the depth map provides an informative cue for segmentation: Kinect can't estimate depth in regions where transparent objects are located.

A. Transparent objects segmentation

Kinect can fail to estimate depth not only on transparent objects, it often fails on contours of objects. Sometimes it can also return valid depth of the background behind transparent objects. To clean this up, morphological operations (closing and opening) are applied to find and eliminate small regions. The result is a proposed mask representing transparent objects.

However, invalid depth doesn't always correspond to transparent objects, and noise distorts masks so these operations can only produce approximate regions of transparent objects. To refine these masks, we use them as an initialization and constraints for the GrabCut segmentation algorithm (Rother et al. [24]) on the corresponding Kinect RGB image. Regions with transparent objects differ from surrounding background only slightly so many segmentation algorithms fail to segment them without additional information. However, this mask derived from a Kinect depth map provides good initialization and enough constraints to allow GrabCut to segment transparent objects accurately.

Results of segmentation are shown in the Fig. 4a. Segmentation of the left object is accurate but the right object is segmented incorrectly due to non-uniform background. However, subsequent steps of our algorithm are robust to such errors of the segmentation. Also there is a third false segmentation mask because Kinect failed on the non-transparent object. This is a rare case but the pose estimation algorithm is robust to such situations too since it is unlikely to match any of the models well.



(a) Results of glass segmentation (b) Projection of the estimated pose

Fig. 4. (a) The algorithm uses simple morphological operations to get approximate masks from a Kinect depth map and then refines them by applying the GrabCut algorithm to a Kinect RGB image. Pose estimation is robust to errors of the segmentation. (b) The algorithm recognized the object correctly and its pose was estimated accurately.

B. Initial pose estimation

We find the initial pose estimation using a test image silhouette. We already have silhouettes of the object for various R_x and R_y from the training stage. So we need to estimate the translation parameters and R_z for each silhouette. In order to do that, we first find a two-dimensional similarity transformation between train and test silhouettes. The transformation consists of a 2D translation, uniform scaling and rotation in the image plane. This is done using Procrustes Analysis (2D shape matching, Dryden and Mardia [6]).

2D translation is estimated by aligning centroids:

$$(\bar{x}, \bar{y}) = \frac{1}{n}\sum_{i=1}^{n}(x_i, y_i), \tag{1}$$

where n is the number of points in a silhouette with points $(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$.

2D scale is estimated by aligning scatters of the points:

$$s = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 + (y_i - \bar{y})^2}. \tag{2}$$

Correspondences between points in the training and test silhouette are not known. So a 2D-2D ICP (Besl and McKay [1]) is used to estimate rotation between these silhouettes and refine other parameters of the similarity transformation.

Now we need to compute a corresponding 3D transformation that maps points of a 3D model to the same locations as this 2D similarity transformation. Unfortunately, there is no such transformation under the perspective camera model in general case. So we use a weak perspective projection model (Hartley and Zisserman [9]) that assumes all object points have the same depth. This assumption is valid if object size is small comparable to the distance to the camera. This model produces the following equation which must be solved for all x, y simultaneously with regard to translation (t_x, t_y, t_z) and rotation R_z:

$$\frac{1}{\bar{z}} SK \begin{pmatrix} x \\ y \\ \bar{z} \end{pmatrix} = \frac{1}{\bar{z} + t_z} K \begin{pmatrix} r_{11} & r_{12} & 0 & t_x \\ r_{21} & r_{22} & 0 & t_y \\ 0 & 0 & 1 & t_z \end{pmatrix} \begin{pmatrix} x \\ y \\ \bar{z} \\ 1 \end{pmatrix}, \tag{3}$$

where K is the camera matrix of intrinsics parameters, S is the similarity transformation written as the 3×3 matrix and

$$R_z = \begin{pmatrix} r_{11} & r_{12} & 0 \\ r_{21} & r_{22} & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{4}$$

There are two solutions to this problem but one of them places an object behind the camera. Denoting a matrix $A = K^{-1}SK$ with elements $A = (a_{ij})$, the unique physically realizable solution is given by equations:

$$\begin{aligned} t_z &= \Big(\frac{1}{\sqrt{\det A}} - 1\Big)\bar{z}, \\ t_x &= a_{13}(\bar{z} + t_z), \\ t_y &= a_{23}(\bar{z} + t_z), \\ \begin{pmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{pmatrix} &= \Big(1 + \frac{t_z}{\bar{z}}\Big) \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}. \end{aligned} \tag{5}$$

Translation parameters and R_z are computed by these equations for each R_y and R_x from the training stage. If R_y and R_x are close to the correct values of the test object pose then the training silhouette and the test silhouette will be matched well but incorrect R_x and R_y will produce bad matches. So we need a function to measure the quality of silhouettes matchings to find a correct pose. We use the Chamfer Distance (Borgefors [2]) for this task. Its main weakness is low accuracy in cluttered scenes and various modifications were proposed (Olson and Huttenlocher [20], Shotton et al. [26]) but we have segmented silhouettes so it is not a problem in our pipeline. Poses with large Chamfer Distance are discarded and non-maximum suppression is applied to remove similar transformations. Several plausible poses remained after these procedures (usually 1-3 for a test silhouette). Computed poses are not very accurate due to discrete sampling of R_y and R_x and weak perspective assumption so they need to be refined.

C. Pose refinement

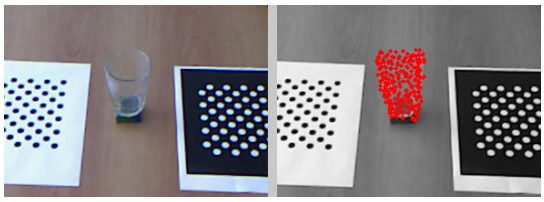
Surface and silhouette edges are represented by 3D point clouds that should be aligned with a segmented silhouette and Canny edges of a test image. However we do not know correspondences between 3D points and edge points. So this is a problem of 3D-2D registration and we use a robust variant of Levenberg-Marquardt Iterative Closest Points (LM-ICP, Fitzgibbon [7]) to solve it. The result of this algorithm is the refined pose.

D. Ambiguity of poses

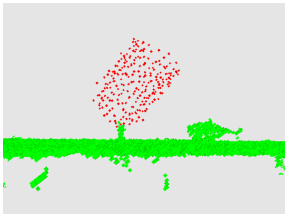
The problem of pose estimation from one monocular image has inherent ambiguity for transparent objects. There exist significantly different poses that have very similar projections to a test image. For example, see the Fig. 5. After a while you can see two different poses in this image: either the wineglass is upside-down or it is lying on its side on the table.

Also see the projected object model for a pose in the Fig. 6a. It appears to be a good pose because model points are projected on the object and model edges are aligned with test edges too. But a point cloud of this scene presented in Fig. 6b shows that in fact this pose is not correct.

Fig. 5. Ambiguous pose. There are two possible interpretations: either the wine glass is turned upside-down or it is laid on its side. This ambiguity wouldn't arise if the object was opaque.



(a) Ambiguous projection



(b) Point cloud of this test scene

Fig. 6. (a) The projected model onto the image is shown. The projection looks good because it is aligned well with edges of the object. (b) A view from the side shows that in fact the model is far away from the true pose (green is the table and red is the object model).

Additional information is required to resolve this ambiguity. For example, one solution is to use a second view of the scene from different viewpoint. However, this solution may require a robot moving and it complicates the pose estimation pipeline. So instead, we use the assumption that the transparent object stays on a support plane. It allows us to resolve pose ambiguity and also be used to refine computed poses. Note that this assumption is not inherent to the approach and can be easily relaxed.

This assumption implies that we need to transform a computed pose in such a way that the object will be put on a support plane and the projection of a model with the new pose should be similar to the projection with the old pose. Denoting the vector of residual errors for a pose T as $err(T)$ and the starting pose as T_0, we need to solve the following

problem:

$$\min_{T} ||err(T) - err(T_0)||, \tag{6}$$

with such constraints on the pose T that the object with this pose must stay on the support plane.

We expand $err(T)$ in Taylor series in the point T_0 and discard the higher order terms because T_0 should be close to the correct pose. So $||err(T) - err(T_0)|| \approx ||J\Delta T||$, where J is Jacobian and $\Delta T = T - T_0$. Minimization of $||J\Delta T||$ is equivalent to minimization of the dot product $(J\Delta T, J\Delta T) = \Delta T^T J^T J \Delta T = \Delta T Q \Delta T$, where $Q = J^T J$.

Constraints on the pose T consist of constraints on rotation of the object (the upright direction must be aligned with the plane normal) and its location (the bottom of the object must belong to the support plane). These are linear constraints on ΔT which we denote as $E\Delta T = d$. So the final problem is:

$$\begin{aligned} &\min_{\Delta T} \Delta T^T Q \Delta T \\ &E\Delta T = d. \end{aligned} \tag{7}$$

This is a quadratic programming problem with linear equality constraints. The solution of the problem is derived with Lagrange multipliers as the solution to the linear system (Boyd and Vandenberghe [3]):

$$\begin{pmatrix} Q & E^T \\ E & 0 \end{pmatrix} \begin{pmatrix} \Delta T \\ \lambda \end{pmatrix} = \begin{pmatrix} 0 \\ d \end{pmatrix}. \tag{8}$$

Estimated poses are updated with computed ΔT and further refined by LM-ICP (Fitzgibbon [7]) with the constraint that the object stays on the support plane. The best pose that achieves the minimum value of the cost function in LM-ICP, is returned as the pose of the object. An example of estimated pose is shown in Fig. 4b.

E. Recognition technique

Each silhouette can belong to only one object because we assume transparent objects don't intersect each other in the test image. So we recognize objects by final values of the cost function in LM-ICP. Its minimum value indicates the correct object which is the most similar to this silhouette.

VI. Experiments and results

We scanned 13 transparent objects and created 3 datasets to evaluate the algorithm. The first dataset had 5 transparent objects and it was used to measure accuracy of pose estimation and recognition. The second and the third dataset had 8 other objects and they were used to evaluate robotic grasping with uniform and textured background. The open source implementation of the system can be obtained from https://github.com/wg-perception/transparent_objects.

A. Pose estimation accuracy

Each of 5 transparent objects from the first dataset was placed on the table with uniform background, one object at time. The table had fiducial markers attached to it. They were used to estimate the ground truth pose. An example of a test image is in the Fig. 7. About 500 640×480 test images were captured (70-120 images per object).

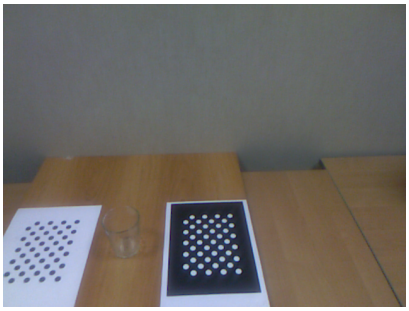


Fig. 7. Example of a test image used to evaluate pose estimation accuracy.

The results are shown in the Fig. 8. Poses are estimated with the accuracy of several millimeters. Ground truth data can have a systematic error of about 4 millimeters due to approximate measurement of the distance to a test object. So we also report relative translation error to exclude this bias in the ground truth data. Rotation error is not reported because all objects in this dataset have rotation symmetry and constraints of the support plane define rotation of an object completely in this case.

Object	Success rate	Mean translation error (cm)	
		absolute	relative
bank	0.99	0.3	0.2
bottle	0.99	1.2	0.5
bucket	1.00	0.5	0.5
glass	0.94	0.4	0.3
wine glass	0.97	1.1	0.5
mean	0.98	0.7	0.4

Fig. 8. Accuracy of pose estimation. Success rate is the proportion of cases when a final pose was closer than 2cm to the ground truth pose. Translation error is the distance between an estimated location of the object and (a) the ground truth location (absolute error) or (b) the mean estimated location computed from all test images (relative error).

B. Recognition

Recognition was evaluated on the same dataset as pose estimation. However, the algorithm had to estimate poses of all objects in this case and return the one with the lowest cost function. Results are in the Fig. 9. Bank, bucket and glass are correctly recognized in 93% of cases on average. However, the wine glass is not recognized at all. This is due to failures at the glass segmentation step. The wine glass has a very thin leg and so was discarded by morphological operations. So, segmentation of the wine glass without the leg looks like a regular glass. It is interesting to note that pose estimation is still robust to such segmentation errors and it has good accuracy for the wine glass as reported in the Fig. 8.

Recognition results show that our algorithm can be applied to the problem of transparent object recognition. However, a more robust segmentation algorithm is required to use it in practice for this task. For example, the segmentation should

be improved with using both the invalid depth cue proposed in this paper and complementary algorithms proposed by McHenry et al. [17], McHenry and Ponce [16].

	bank	bottle	bucket	glass	wine glass
bank	0.97	0	0	0.03	0
bottle	0.01	0.57	0	0.42	0
bucket	0	0	0.84	0.16	0
glass	0.01	0	0.01	0.98	0
wine glass	0	0	0	1	0

Fig. 9. Normalized confusion matrix. Actual classes are in the rows and predicted classes are in the columns. The algorithm recognizes bank, bucket and glass in 93% of cases on average but fails to recognize the wine glass due to problems with segmentation of its thin leg.

C. Grasping

Fig. 10. Dataset of 8 transparent objects with different shapes, materials and characteristics (one of them is translucent). Test objects for evaluating robotic grasping are at left. Identical copies of these objects, painted with color, are on the right. They were used to create 3D models with KinectFusion at the training stage.

The dataset of 8 transparent objects was used to evaluate grasping (Fig. 10). A transparent object was put before the robot on a table in different places. Then our pose estimation algorithm was executed to get the pose of this object. The computed pose was used to grasp the object. The robot had to lift the object for 10 cm. Grasping was considered successful if the robot could hold the object for at least 30 seconds.

We evaluated the algorithm in two different conditions. The first experiment had simple uniform background (Fig. 11a). The second experiment had challenging background with non-transparent clutter (Fig. 11b). We put objects on one of the paintings that lie on the table.

We had 10 grasp attempts for each object in each experimental setup. Results are presented in the Fig. 12. The robot can grasp a transparent object in 80% of cases and this means the algorithm is stable to cluttered background. However, the results in two setups are different for the wine glass. There are two possible ways to grasp the objects from the database: from the side or from the top. Grasps from the top are easy

(a) Uniform background (b) Cluttered background

Fig. 11. Typical images of a transparent object (it is in the middle) which were used to evaluate robotic grasping.

Object	Uniform background	Non-transparent clutter
tall glass	8	8
small cup	10	8
middle cup	10	10
wine glass	9	4
yellow cup	9	10
perfume bottle	9	10
perfume ball	3	7
perfume box	8	7
mean	8.25	8.00

Fig. 12. Number of successful grasps for each object from 10 attempts. Grasps are successful in 80% of cases and this result is robust to complex non-transparent clutter.

because they will be successful even if the gripper is shifted slightly from the correct position. Grasps from the side are very difficult for some objects. For example, the wine glass is almost as big as the maximum width of the PR2 grip. This means that even small errors at any stage of the grasping pipeline (model capture, pose estimation, PR2 calibration) will result in a failed grasp. The wine glass was grasped mostly from the top when background was uniform. But the robot tried mostly to accomplish grasps from the side when the background was cluttered because additional objects changed the collision map and other grasp points were selected in the PR2 grasping pipeline (Ciocarlie [5]). So this difference in results are explained not by the difference in background but by these two types of grasps. Also, results are different for the perfume ball in two setups. The perfume ball is a challenging object to grasp because it is heavy and can easily slip out of the gripper, resulting in less stable grasps than other objects used. The difference in results is explained by higher variance of successful grasps for this object.

The results of our algorithm don't change with different backgrounds. This property is very important when dealing with transparent objects because various cluttered backgrounds behind transparent objects is one of the main challenges for computer vision algorithms. Our algorithm is robust to clutter because a Kinect depth map is used for segmentation and it is not affected by background behind transparent objects.

VII. Conclusion

The paper presents new algorithms for segmentation, pose estimation and recognition of transparent objects. A robot is able to grasp 80% of known transparent objects with the proposed algorithm and this result is robust across non-specular backgrounds behind the objects. Our approach and other existing algorithms for pose estimation (Klank et al. [12], Phillips et al. [22]) can not handle overlapping transparent objects so this is the main direction for future work.

Acknowledgments

We would like to thank Vincent Rabaud (Willow Garage, research engineer) and Ethan Rublee (Willow Garage, software engineer) for help with integration of the algorithm into the object recognition stack and connection with the PR2 grasping pipeline. Also we would like to thank the anonymous reviewers for their useful comments and suggestions that helped to improve the paper.

References

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

[2] G. Borgefors. Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1988.

[3] S.P. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge Univ Pr, 2004.

[4] Walon Chiu, Ulf Blanke, and Mario Fritz. Improving the Kinect by Cross-Modal Stereo. In *22nd British Machine Vision Conference (BMVC)*, 2011.

[5] M. Ciocarlie. Object manipulator package (ros). *http://www.ros.org/wiki/object_manipulator*, 2012.

[6] I.L. Dryden and K.V. Mardia. *Statistical shape analysis*, volume 4. John Wiley & Sons New York, 1998.

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

[8] M. Fritz, M. Black, G. Bradski, and T. Darrell. An additive latent feature model for transparent object recognition. *Advances in Neural Information Processing Systems (NIPS)*, 2009.

[9] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge Univ Press, 2000.

[10] S. Hinterstoisser, S. Holzer, C. Cagniart, S. Ilic, K. Konolige, N. Navab, and V. Lepetit. Multimodal Templates for Real-Time Detection of Texture-Less Objects in Heavily Cluttered Scenes. In *IEEE International Conference on Computer Vision*, 2011.

[11] 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*, pages 87–108, 2008.

[12] 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.

[13] V.R. Kompella and P. Sturm. Detection and avoidance of semi-transparent obstacles using a collective-reward based approach. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011.

[14] Zhong Lei, K. Ohno, M. Tsubota, E. Takeuchi, and S. Tadokoro. Transparent object detection using color image and laser reflectance image for mobile manipulator. In *Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on*, pages 1 –7, dec. 2011.

[15] 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.

[16] K. McHenry and J. Ponce. A geodesic active contour framework for finding glass. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006)*. IEEE, 2006.

[17] K. McHenry, J. Ponce, and D. Forsyth. Finding glass. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005)*. IEEE, 2005.

[18] F. Mériaudeau, R. Rantoson, D. Fofi, and C. Stolz. Review and comparison of non-conventional imaging systems for three-dimensional digitization of transparent objects. *Journal of Electronic Imaging*, 21:021105, 2012.

[19] R.A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A.J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. In *10th IEEE International Symposium on Mixed and Augmented Reality*, 2011.

[20] C.F. Olson and D.P. Huttenlocher. Automatic target recognition by matching oriented edge pixels. *Image Processing, IEEE Transactions on*, 6(1):103–113, 1997.

[21] M. Osadchy, D. Jacobs, and R. Ramamoorthi. Using specularities for recognition. In *9th IEEE International Conference on Computer Vision*. IEEE, 2003.

[22] 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.

[23] M. Quigley and WillowGarage. Robot operating system, (ros). *http://www.ros.org/wiki/*, 2012.

[24] C. Rother, V. Kolmogorov, and A. Blake. GrabCut: interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (TOG)*, 2004.

[25] E. Rublee, T. Straszheim, and V. Rabaud. Object recognition. *http://www.ros.org/wiki/object_recognition*, 2012.

[26] J. Shotton, A. Blake, and R. Cipolla. Multiscale categorical object recognition using contour fragments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1270–1281, 2007. ISSN 0162-8828.

[27] M. Ulrich, C. Wiedemann, and C. Steger. CAD-based recognition of 3d objects in monocular images. In *International Conference on Robotics and Automation*, volume 1191, page 1198, 2009.