Seeing the Unseen: Simple Reconstruction of Transparent Objects from Point Cloud Data

Sven Albrecht
Institute of Computer Science
Osnabrück University
Osnabrück, Germany
Email: sven.albrecht@uni-osnabrueck.de

Stephen Marsland
School of Engineering and Advanced Technology
Massey University
Palmerston North, New Zealand
Email: s.r.marsland@massey.ac.nz

***Abstract*—Robot mapping, both indoor and outdoor, is typically based on sets of 3D measurements of the environment (point clouds) coming from either laser range finders or RGB-D cameras. While both of these sensors provide accurate data about objects within a relatively wide range, they fail to provide directly informative readings about transparent or highly reflective objects, which are commonly found in cluttered indoor environments such as homes and offices.**

This paper describes a method of recognising that there are transparent objects within a scene and reconstructing them from the limited information that is available. Our method is based on reconstructing geometric properties of the missing objects using inference from the shadows that are left. This provides an estimation of the volume of missing objects.

We demonstrate the methods first on regular measurable object to compare our estimation with measured data and present the reconstruction of two exemplary transparent objects.

I. Introduction

The aim of interacting with mobile robots in human environments (whether the aim is household assistance or service robotics within an office environment) necessitates the robot being able to reliably sense and represent its environment. Research over the past few years has resulted in the reliable generation of consistent 3D maps of human environments based on data from 3D laser scanners and RGB-D cameras, which produce point cloud data. Depending on the sensor, additional information such as pixel colour or remission values may be attached to each measurement point.

Many of the problems of dealing with such data, such as simultaneous localisation and mapping (SLAM) [2, 5] and 'closing the loop' [7, 12], are generally well-researched. However, there are limitations on the environmental materials that the sensors can detect. Laser range finders and RGB-D cameras have problems measuring distances to both transparent and reflective surfaces: laser beams get refracted, resulting in faulty measurements at some locations and the same holds true for the infrared pattern projected by RGB-D cameras.

An example of this 'blindness' is shown in Figure 1. On the left is a 2D image of the scene, while on the right is a 2D projection of the corresponding 3D point cloud. There is a cafetiere (or French Press) for making coffee on the table (circled in the point cloud data), which is made of pyrex and thus hard to see in both images. In particular, in the point cloud data, only the handle can be seen. The 2D image also shows that transparent objects are hard to see using normal camera images, except for reflections at certain angles.

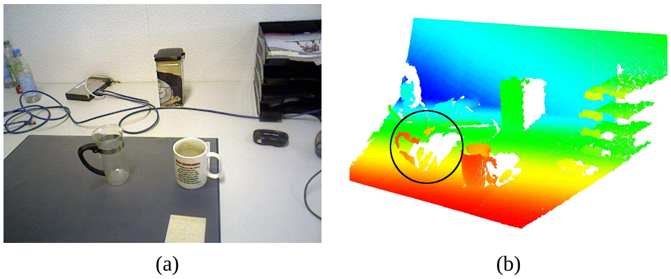

(a) (b)

Fig. 1: Example of a scene with a transparent object. *Left:* a photograph of the scene, and *right:* a 2D projection of the corresponding 3D point cloud with points coloured by depth. Note that the transparent French press (circled in black) only provides point cloud information for the handle.

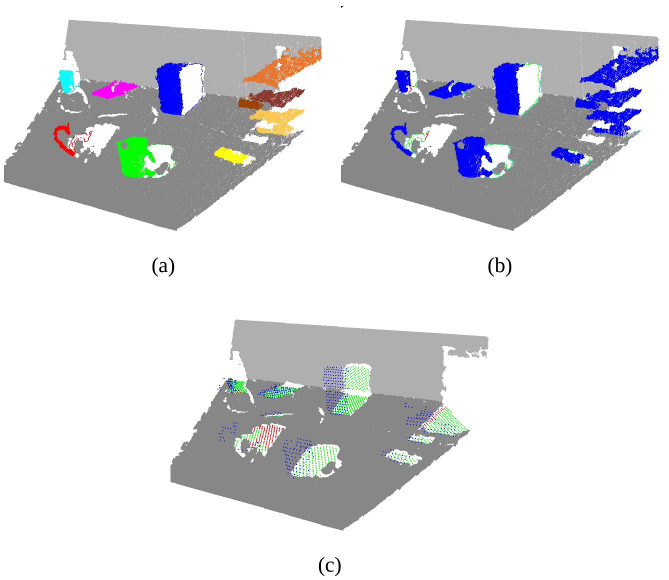
However, note that there is some indication of the presence of the cafetiere in the point cloud in that the 'shadow' of the object is present. It is this shadow that provides the information that we can use to reconstruct the object, as we will demonstrate in this paper.

A. Related Work

There have been three principal approaches to the detection and recognition of transparent objects in the literature; for a review of methods, see [4]. In the first approach reflections that appear from certain angles are used to infer information about the pose of transparent objects [8], while in the second physical properties of the materials are used [4, 13].

However, the techniques that bear most similarity to our own are in the third class. These are based on either time of flight (ToF) cameras (e.g., [6]) or the Microsoft Kinect, a common RGB-D camera, such as [10] and very recently [9], and the sensor that is used for the experiments in this paper.

In [6] the fact that in ToF intensity images any transparent objects appear darker than their background is used to detect potential transparent objects. The same scene is then viewed from a different viewpoint, and the assumption of planarity is used to reconstruct them via triangulation.


(a)

(b)

(c)

Fig. 2: Result of scene analysis for the data shown in Figure 1: (a) shows the detected clusters and their concave hulls projected onto the two detected planes, clusters are distinguished by colour; (b) shows the detected clusters (blue) and the projection of the corresponding concave hulls, red indicates that the concave hull point does not have any points from the planes in its neighbourhood, while for the green points one of the points in cyan lies in their neighbourhood; (c) shows the results of analysis of the detected holes: green indicates that this part of the hole can be explained by a measurement in front of it (blue points), while red points are holes that are unexplained by the data, and therefore potentially from reflective or transparent objects.

By way of contrast, machine learning methods can be used, as in Lysenkov et al. [9, 10], where single frames taken by a Kinect sensor are used to first detect positions for transparent objects and then apply edge fitting to identify the object and its pose from a set of trained objects.

Both [6] and [9, 10] attempt to grasp identified transparent objects, providing feedback on correctness of their approaches.

Our approach is also based on data from a Kinect sensor, but we do not require a learning process. Instead, we use a first frame to detect transparent objects, and then, similar to [6], acquire additional frames from different viewpoints. However, we only assume that there is a planar surface underneath or behind the transparent object (i.e., the floor or wall, or a table).

II. Detecting Transparent Objects

Our detection of transparent objects requires the assumption that at least one planar surface is present in the observed scene and that the transparent object is placed either on top or in front of that planar surface. This means that the 'shadow' of the object lies on one or more planes. This is less restrictive than comparable approaches (i.e. [6, 9, 10]) which assume that the transparent object be placed on top of a planar surface.

In accordance with our assumption, the first step in the detection is a segmentation of planar surfaces within the scene, which proceeds in an iterative RANSAC fashion and terminates if either the number of points that lie within the plane is below a given threshold or the number of remaining scene points to be analysed is sufficiently small.

Planes are represented as a point in the plane (d) and the normal vector $\mathbf{n}$ so that every point $\mathbf{p}_\mathrm{p}$ in the plane satisfies:

$$\mathbf{p}_\mathrm{p} \cdot \mathbf{n} - d = 0 \qquad (1)$$

All remaining points are clustered based on the Euclidean distance between each point and its neighbours, reasoning that each of these clusters belongs to one object or to nearby multiple objects if they are positioned close together.

While transparent objects do not usually provide any points on their surface, reflective objects often feature correct measurements for portions of their surface. To determine if a cluster reflects the actual size of the surface of the object, we compare the cluster with the corresponding shadow on the planar surface(s). This is done by computing the concave hull of the object clusters and projecting these points onto the planar surface of the detected planes. To project a point $\mathbf{h}$ belonging to the concave hull, we write the equation for each point $\mathbf{p}_\mathrm{l}$ on the line between the sensor and the hull point as:

$$\mathbf{p}_\mathrm{l} = a\,(\mathbf{h} - \mathbf{s}_0) + \mathbf{h} \qquad (2)$$

where $\mathbf{s}_0$ is the sensor location and a is a scalar value.

Obviously, the intersection of the plane defined in (1) and the line in (2) corresponds to the projection of the point $\mathbf{h}$ on the plane observed from position $\mathbf{s}_0$. The intersection can be computed by solving:

$$a = \frac{-\mathbf{h} \cdot \mathbf{n} + d}{(\mathbf{h} - \mathbf{s}_0) \cdot \mathbf{n}} \qquad (3)$$

For each of the projected points a radius search is performed on the inliers of the planar surface that they are projected onto. In the case of regular objects, where the shadow corresponds to the measured cluster points, the distance between a projected point of the cluster's concave hull and the enclosing planar inliers is quite small, thus the radius search will return one (or more) points in the neighbourhood of the projected point. Clusters that contain only partial measurements of the surface of the object, observable from the current viewpoint, will feature a substantial portion of projected points where the radius search will not return any planar inlier; an illustration is provided in Figure 3. The result of this is that the fraction of projected points that do not have a corresponding point on the plane to the total number of projected points gives a good estimate whether a cluster resembles the object it belongs to.

With the methods discussed so far we are able to determine if there is an object present in the scene that was only partially measured (which is therefore potentially reflective), but not whether any transparent objects are present, since they typically do not yield any cluster of measurement points.

To detect those we now consider the 'holes' in our segmented planar surfaces, where by 'hole' we mean regions

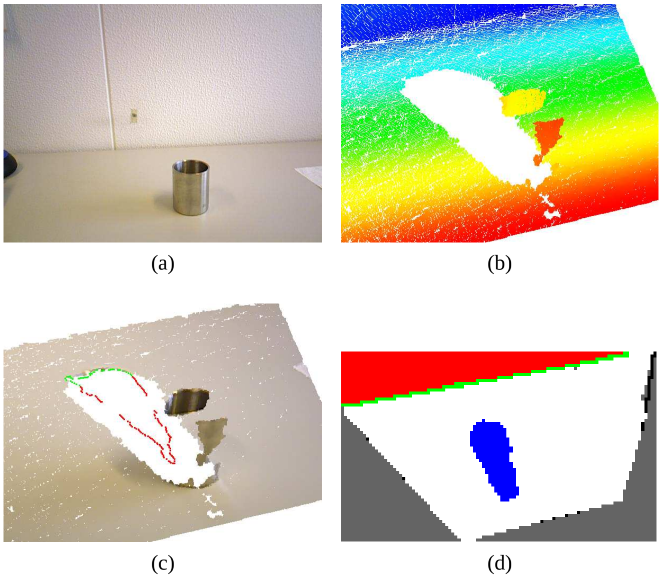


Fig. 3: Projection of cluster hulls on the detected plane: (a) photograph of the scene; (b) cutout of the corresponding point cloud, coloured by depth from sensor; (c) planar inliers and projected cluster hull points: red indicates that no point was returned from the radius search; (d) 2D grid representation of the planar surface, which is used to identify the 'holes' that signify transparent objects.

within the boundaries of the convex hull of the planar surface that have an absence of measurement points and are either surrounded by measurement points or share a (partial) border with another plane. These holes can easily be determined if we transform the planar surface into a 2D grid and start a simple region growing process from grid cells that do not contain any point data. If multiple empty grid cells are connected to each other, this region becomes a hole in the previously described sense. The advantage of the grid representation is not only that it becomes fast and easy to detect the hole regions, but by choosing an appropriate grid cell size we gain some robustness towards sensor noise, which is needed if we want to apply our approach to RGB-D data from Kinect-like cameras.

An example for the 2D grid is shown in Figure 3d: Regularly filled grid cells are coloured white and empty cells either in blue (detected hole) or gray (outside of the planar surface). Black points indicate the convex plane hull, if it diverges from the border of the filled cells, while green marks the intersection with other planes and red the grid points that lie behind another planar surface.

Once the holes in the planar surfaces have been determined in the 2D grid we can transform the corresponding grid cell centres back into their 3D position in the frame. For each of the resulting 3D points we define the ray from them towards the origin (i.e. the position of the sensor). For each of these rays we check if it intersects any of the acquired measurement points, which can be done efficiently by creating an octree from all points of the current frame that do not belong to any of the planar surfaces. If the ray intersects a leaf of the octree, we know that something was measured between the camera and this particular part of the hole, thus its existence is explained by the data as a regular occlusion. Alternatively, if the ray does not intersect any leaf, then this part of the hole region is not explained by the available data.

A visualization is given by Figure 2c, where the centres of empty voxel cells, transformed back into the scene, are coloured in green, if they can be explained by some measurement – the intersected octree leaves are shown in blue. Red points indicate that no octree leaf was positioned between this grid cell and the sensor. Not surprisingly the hole caused by the cafetiere in Figure 2c features a lot of red points. However regular objects, like the paper trays on the right of the scene in Figure 2c, can exhibit some of these points on their edges, due to sensor noise. Thus we defined a threshold for the fraction of unexplained empty grid cells in relation to the total of the cells defining a hole, to determine that a transparent (or at least, only partially measured object) is present in the current frame. In our experiments a threshold of 0.5 worked well, i.e. we assume a transparent object if 50% or more of the grid cells defining a hole could not be explained by measured data. Via the location of the hole in the 3D point cloud of the current scene we also have a good estimate about the position of the transparent object.

III. Reconstruction

In section II we presented a simple approach to determining whether or not a transparent or only partially measured specular object is present in a single RGB-D frame / laser scan. In such a case we can use the information gathered by several frames from different viewpoints in order to give an estimate of the size and position of the object. Once this has been done the individual frames have to be transformed into the same reference frame. This problem is well studied and several robust solutions exist like ICP [1], so we do not consider this problem here, although a good registration is crucial for the subsequent reconstruction process.

The basic idea for the reconstruction itself is pretty simple: While a single frame / scan does not provide sufficient information to estimate the size and shape of an object without prior knowledge, it does provide information via the occlusion that the object caused. From a single frame we can only deduce that somewhere between the camera and the hole on the planar surface there must have been some object that refracted or otherwise obstructed the measurements of the sensor. This basically leaves a volume that resembles a cone, except that its base is composed of the shape of the detected hole on the planar surface. In the remainder of this paper we will refer to such a volume as an (occlusion) *frustum*. Such a frustum can be constructed for each individual frame. If all frames are registered consistently than we can easily conclude that the object that caused the occlusion in each individual frame has to be part of the intersection of all frusta.

Thus, the more observations from different viewpoints we gather from the object in question, the more precise our estimation of the object becomes, since the intersection can only shrink with additional information. If viewed as an iterative process the intersection operation first takes the frustum of some arbitrary initial frame and then fuses it with the new information provided by the next frame: all parts of the initial frustum which do not fit the newly gathered data from the second frame are removed. The caveat in this is that if the registration of the frames is skewed, then the resulting intersection will be skewed as well – there might be parts where we 'chiseled' away too much or not enough. However, if the registration error is not too large then we will still get a reasonable estimate of the objects volume.

In order to compute the intersections we chose a sample-point-based approach. In a first step we create a point sample representing the planar occlusion caused by the transparent object. If only a very coarse approximation is desired then it can be sufficient to simply reuse the centres of the empty voxel grid cells, otherwise the hole region can be resampled with a desired density. The results presented in section IV were obtained using random point samples with a density of 10 points per cm^2, but we believe that a less dense representation will also provide good results, although we have not yet performed any experiments in that direction.

In a similar way to the method by which transparent objects are detected, a ray is project from each sample point to the origin (i.e., the camera position) using equation (2). If parts of the occlusion could be explained by regular measurement points, for example if an object is a combination of transparent and non-transparent materials, then the rays are only appropriately sampled from the planar surface to the measurement points on the rays, thus expressing the knowledge that the space between the sensor and the measured point is free and we don't have any information about the volume between the measurement point and the planar surface.

Using the same poses that we obtained from registering the individual frames, we can transform the points of all sampled occlusion frusta into a common reference frame. As a next step we simply create an octree containing all sample points and afterwards iterate over all leaves: If a leaf contains enough sample points and we detect all labels of the involved frames, this leaf is considered to be part of the intersection. Otherwise we can discard the leaf, since it only contained a few points, thus meaning that it was on or near the borders of the frusta or was not part of the intersection (since in at least one frame this particular volume was not occluded, i.e. in that frame the leaf's volume is between the camera and some regular measurements).

IV. Results

In our current experiments we used data provided by Microsoft Kinect and Asus Xtion Live sensors. However since we only rely on the 3D information of the point clouds we believe that we can achieve results of at least the same quality if we were to use a 3D laser scanner instead, since the point clouds provided by a laser scanner usually feature much less noise. The implementation was done in C++ and makes use of the *Point Cloud Library* (see [11]) and its many data structures.

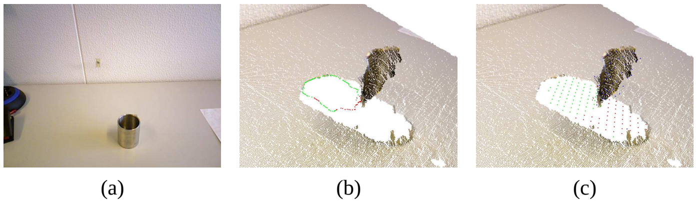


Fig. 4: Example of misclassification: (a) photo of the captured frame; (b) projected concave hull, for the red points no neighbour was returned by the radius search; (c) analysis of the hole: red points are unexplained, green points are explained by measurements (in blue).

A. *Detection Results*

As a first evaluation for our detector we took a small set of frames (30 in total), each containing at least one planar surface and at least one object on top of or in front of the planar surface. The objects that we used were either (1) regular (i.e. they can be measured without any problems), (2) reflective (so that they provided only partial measurements) or (3) transparent (i.e., providing very few measurements). There were 32 regular objects, 7 reflective ones, and 9 transparent ones. 16 of the 30 frames contained at least one non-regular object. The detection rates of this small experiment are shown in Table I and suggest that our approach is worthy of further investigation. Of the 16 frames that contained at least one non-regular object, 14 of them were correctly identified as containing the objects.

The false positives for the regular objects are caused by our choice of parameters when to split two clusters of points. We used a maximal Euclidean distance of 2 cm for points to belong to the same cluster. This can cause some objects to be split into two distinct clusters, while larger thresholds will cause distinct objects to be fused into one cluster. Depending on the data a misclassification can cause three negative entries in the table: if the projected concave hull of a specular object for a large part approximated the shape of the occlusion, it will be labelled as a regular object (thus resulting in one false positive and one false negative). However, if the analysis of corresponding hole features a large portion of unexplained empty grid cells, the same object can additionally cause a false positive transparent object to be detected. An example is shown in Figure 4. To avoid such types of misclassification the information of the analysis of the concave hulls needs to be combined with the analysis of the holes, which we have not done, yet.

A better choice of the required thresholds (currently set by hand) should lead to further improvements and we intend to test this on a larger dataset in the future.

B. *Reconstruction Results*

As a first experiment for the reconstruction we applied our method to the occlusion caused by a non-transparent object.

TABLE I: Detection Results

	true positives	false positives	false negatives	precision	recall	F_1 score
Regular	32	4	0	88.9 %	100.0 %	94.1 %
Reflective	5	0	2	100.0 %	71.4 %	83.3 %
Transparent	8	4	1	66.7 %	88.9 %	76.2 %

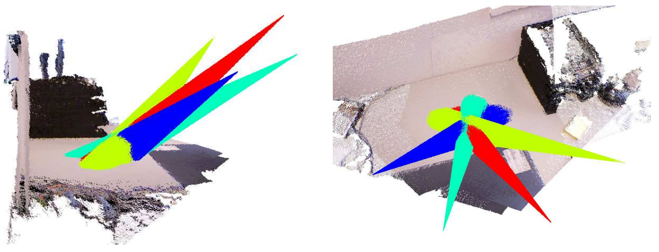


Fig. 5: Partial view of the point clouds from the mug reconstruction: Point clouds and variously colour occlusion frustra from two different viewpoints of the mug.

This way we are able to compare the results of our reconstruction with some actual measurements to provide evidence of how well the approach works. Apart from the fact that we fed the detected holes, caused by the occlusion of the mug, directly to our reconstruction, this experiment does not differ from the subsequent experiments. The object chosen was a normal coffee mug. We observed it from 4 different points of view The frusta computed from each of the 4 frames (transformed into the common reference frame) are presented in Figure 5a and 5b respectively. Comparison between the real observed data (which we interpret as some kind of ground truth) and the volume estimated by our reconstruction shows that while the reconstruction is not perfect, it should be sufficient for manipulation tasks.

After the promising results from the mug reconstruction, we applied our method to several transparent objects. In the following we show the results for the reconstruction of a glass and a French press, sporting a non-transparent handle, each composed of images of the scene from six different points of view (see Figures 6 and 8). To provide an impression of the quality of the reconstruction we present the registered point cloud from two perspectives, first together with the registered occlusion frusta and secondly with the intersection resulting from the frusta. The results are in Figure 7 for the glass and Figure 9 for the cafetiere.

V. Conclusion and Future Work

In this paper we have presented a novel and simple approach to estimate the size, shape and position of transparent objects in point cloud data. Our results indicate that the resulting estimation will be suitable for collision avoidance or interactions like grasping of the transparent objects.

Currently, we are working on an alternative representation for describing the transparent objects. While the filled leaves of the octree give a good impression for the size and shape of the object its volume is constructed from cuboid elements (i.e. the octree leaves). This can, depending on the chosen size of the leaves, lead to a 'blocky' representation of the surface, which might provide subsequent interactions like grasping with slightly wrong information. The alternative representation that we intend to employ makes use of polyhedra to constrain the frusta caused by the occlusions.

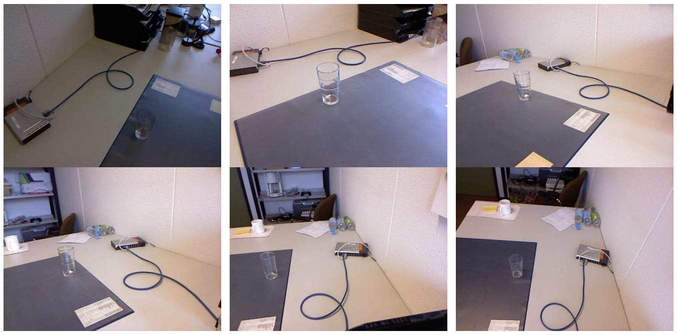


Fig. 6: The six different point of views used to reconstruct the glass on the tabletop.

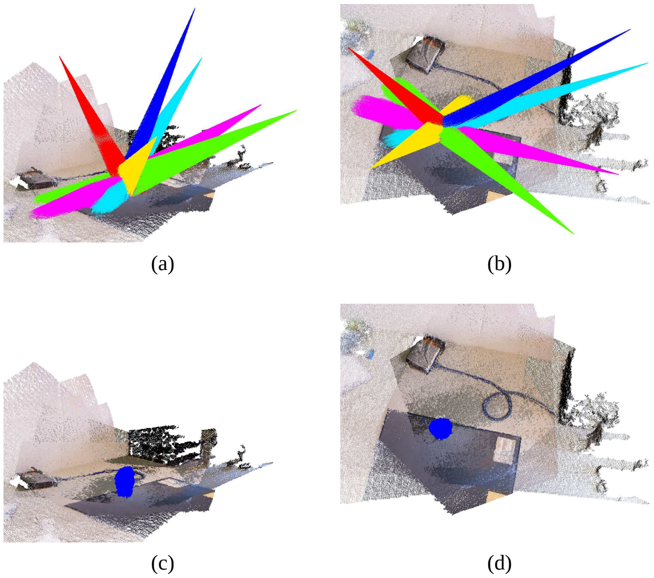


Fig. 7: Reconstruction result for dataset shown in Figure 6: (a) and (b) show the occlusion frusta (colours used to distinguish between each frame); (c) and (d) display the resulting intersection (for viewing purposes completely coloured in blue).

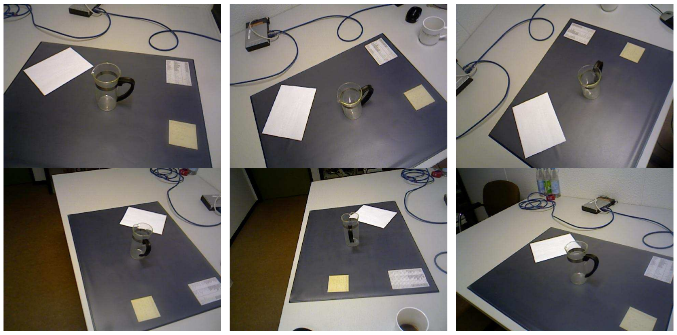
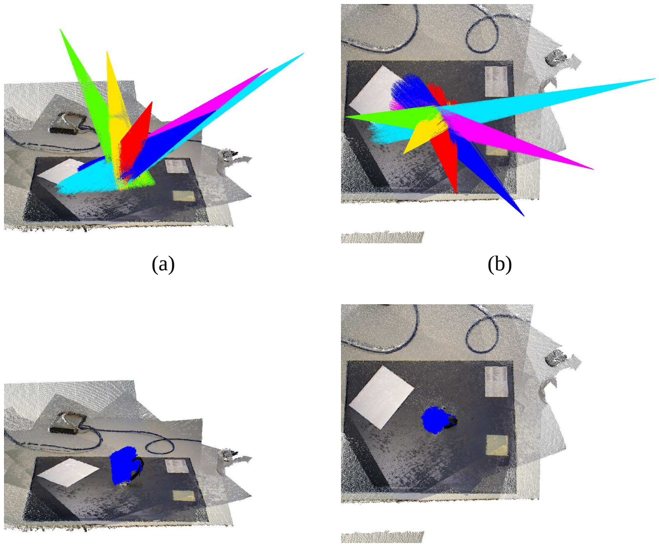


Fig. 8: Six different viewpoints for dataset featuring a French press



(a)

(b)

(c)

(d)

Fig. 9: Results for French press reconstruction from the data shown in Figure 8: (a) shows a side view and (a) a top view of the scene with the occlusion frusta; (c) and (c) display the corresponding intersection from the same view points. Note that the plastic handle nicely fits onto the reconstructed shape.

Since we use only the 3D information and not any colour information available by RGB-D cameras, it should be possible to further improve our reconstruction method – often some measurements of the tabletop are obtained at the base of the transparent object. An example can be seen in Figure 1b, where partial measurements of the tabletop were obtained at the base of the French press (in the lower half of the black circle). However, in such cases we believe it is possible to use an edge detection algorithm in a restricted region of the image, corresponding to the frame, to determine if these measurements should be part of the occlusion frustum or if they are part of the directly observed tabletop. An approach of Fritz et al. [3] would also be suitable to solve this problem.

Additionally, we plan to acquire the frames with one of our robotic platforms and use the estimated surface to actually manipulate transparent objects with a robotic arm. Also, in order to obtain a better evaluation for the reconstruction we are planning to compare the reconstructions with models of the objects in question – either created by hand or by obtaining measurements from non-transparent equivalents as in [9, 10].

Acknowledgments

The work on this approach started during the stay of Sven Albrecht at Massey University, New Zealand, as part of the Mermaid project. The work started in Mermaid reported here was partially funded by BMBF (grant NZL 10/003) and the RSNZ (grant FRG10-11) under a bilateral research collaboration agreement. This is gratefully acknowledged.

References

[1] P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):239–256, 1992.

[2] D. Borrmann, J. Elseberg, K. Lingemann, A. Nüchter, and J. Hertzberg. Globally consistent 3D mapping with scan matching. *Robotics and Autonomous Systems*, 56 (2):130–142, 2008.

[3] M. Fritz, M. J. Black, G. R. Bradski, S. Karayev, and T. Darrell. An additive latent feature model for transparent object recognition. In *NIPS*, pages 558–566, 2009.

[4] I. Ihrke, K. N. Kutulakos, H. P. A. Lensch, M. Magnor, and W. Heidrich. State of the art in transparent and specular object reconstruction. In *STAR Proc. Eurographics*, pages 87–108, 2008.

[5] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon. Kinectfusion: real-time 3D reconstruction and interaction using a moving depth camera. In *Proc. UIST*, pages 559–568, 2011.

[6] U. Klank, D. Carton, and M. Beetz. Transparent object detection and reconstruction on a mobile platform. In *ICRA*, pages 5971–5978, 2011.

[7] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g^2o: A general framework for graph optimization. In *ICRA*, pages 3607–3613, 2011.

[8] P. Lagger, M. Salzmann, V. Lepetit, and P. Fua. 3D pose refinement from reflections. In *CVPR*, 2008.

[9] I. Lysenkov and V. Rabaud. Pose estimation of rigid transparent objects in transparent clutter. In *ICRA*, pages 162–169, 2013.

[10] I. Lysenkov, V. Eruhimov, and G. R. Bradski. Recognition and pose estimation of rigid transparent objects with a Kinect sensor. In *Robotics: Science and Systems*, 2012.

[11] R. B. Rusu and S. Cousins. 3D is here: Point cloud library (PCL). In *ICRA*, 2011.

[12] J. Sprickerhof, A. Nüchter, K. Lingemann, and J. Hertzberg. A Heuristic Loop Closing Technique for Large-Scale 6D SLAM. *Automatika*, 52(3):199–222, December 2011.

[13] A. Wallace, P. Csakany, G. Buller, and A. Walker. 3D imaging of transparent objects. In *Proc. British Machine Vision Conference*, pages 466–475, 2000.