Reconstruction of transparent objects in unstructured scenes with a depth camera

Nicolas Alt, Patrick Rives, Eckehard Steinbach

RECONSTRUCTION OF TRANSPARENT OBJECTS IN UNSTRUCTURED SCENES WITH A DEPTH CAMERA

Nicolas Alt[⋆] *Patrick Rives*[†] *Eckehard Steinbach*[⋆]

[⋆] Institute for Media Technology, Technische Universität München, Munich, Germany
[†] INRIA Sophia Antipolis Méditerranée, Sophia Antipolis, France

ABSTRACT

The visual 3D reconstruction of transparent objects in unstructured scenes is challenging due to the complex image formation principles underlying their visual appearance. Most state-of-the-art reconstruction methods ignore this problem and assume Lambertian reflection. Yet, transparent objects are relevant scene information for applications in intelligent robotics (such as grasping) or virtual reality. In this work, we present an approach to detect non-planar transparent objects, like bottles or glasses, by specifically searching for geometry inconsistencies caused by refraction or reflection. Depth information is acquired using a Kinect sensor, which is moved within the scene in order to acquire multiple views. The individual measurements are combined into a 3D volume, yielding the objects' location and a rough shape estimate. Results are presented using various household objects made of glass or plastic.

***Index Terms*—** Computer Vision, 3D Reconstruction, Transparency Detection

1. INTRODUCTION

Visual geometry reconstruction of unstructured domestic or industrial scenes is an important problem for applications in virtual reality, 3D video or robotics. With the advent of the Kinect sensor, which provides scene depth information using a projected pseudo-random IR pattern, more reliable, accurate and fast methods for 3D reconstruction have been proposed, such as KinectFusion [1]. Typically, these methods build high-quality 3D models by fusing multiple measurements and thus reducing sensor noise.

Transparent objects, however, cannot be reconstructed with methods which assume a consistent appearance of the observed 3D structure for different viewpoints. Regardless of whether an active visual 3D sensing technique or a passive stereo camera is used – the appearance of transparent objects depends on multiple factors such as the background behind the object, viewing angle, object geometry and local reflectance. The image formation model is more complex than for diffusely reflecting (Lambertian) surfaces, which are assumed by many existing reconstruction methods.

We propose an algorithm that searches the depth map acquired by a depth camera for inconsistency effects caused by transparent objects. Consistent scene parts are filtered out. The result of our method hence complements existing approaches for 3D reconstruction of Lambertian objects. The scene must be acquired from multiple views, for instance by moving the camera along a trajectory. Inconsistent regions are accumulated along the trajectory, which allows us to reconstruct the 3D position and a rough shape estimate of the transparent region. The proposed method detects objects with a smooth, curved surface exhibiting dominant refractive effects and, with limitations, surfaces with specular reflection.

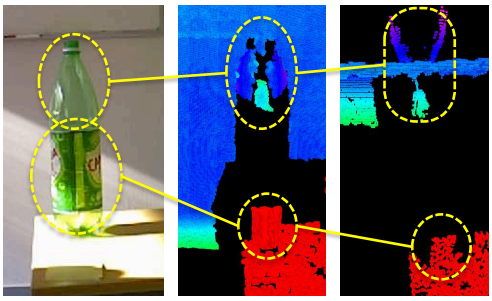


Fig. 1: *Left*: A partly transparent bottle in front of a wall is observed by the Kinect. *Center*: The corresponding pointcloud (rendered above the viewpoint of the real camera) is correct around the label, whereas the transparent bottleneck is missing. Instead, this part of the object distorts the background by refraction, clearly visible in the top view (*right*).

2. RELATED WORK

Lysenkov et al. propose in [2] a Kinect-based detection method for transparent objects which takes advantage of the fact that many transparent objects appear as holes in the depth map (no/invalid data) with this sensor. These holes are used as candidates for transparent objects and serve as an initialization for a segmentation algorithm which is used to extract contours in the RGB image. Objects and their poses are detected in the image by comparing the region with pre-learned models. Our method, in contrast, processes "valid" measurements from the sensor and does not rely on holes, which are also caused by many other effects (see below). Furthermore, we do not rely on pre-learned object models.

Other recent approaches for the detection of transparent objects work with transparency features that model the appearance as an additive combination of patches [3] or rely on the partial absorption in transparent material, measured from two viewpoints with an active sensor [4]. In [5], an overview of methods for the reconstruction of specular and transparent objects using various sensors and setups is given. Several groups of image formation principles (such as diffuse reflection, refraction, scattering) are identified and used to classify the different approaches. It can be noted that the presented approaches all address very specific problems (concerning object type, sensor, setup) and that a general framework for the perception of objects with non-Lambertian reflection – to the best of our knowledge – is not available.

Additionally, the proposed method relies on a pose estimator in order to allow for projections between different views. Many systems exist for this problem, but ideally the pose estimate is based

on the same (visual) sensor in order to obtain synchronous results. Here, a visual tracker is used which estimates the camera pose from the entire grayscale image, ensuring a high accuracy [6]. The tracker relies on a model which consists of several 2.5D reference views and can be learned online or offline. For initialization, a search in the reference views is performed. Special markers are not needed. The approach selects salient and robust pixels in the image, which allows for real-time operation and ensures stability to partly changing or dynamic scenes.

3. EFFECTS OF TRANSPARENCY

Image formation for transparent objects is based on a multitude of effects, as discussed in [5]. In this work we mainly exploit the "lens effect" caused by refraction of light passing through a curved transparent object. This is the dominant effect for clear, smooth materials hit by light rays with a non-acute incident angle. Looking at a transparent object, an image of the background is observed which is distorted by refraction of the light ray as it passes through the air-surface boundaries of the transparent object. In this distorted image, depth estimation methods detect a "virtual object" whose depth is shifted from the real depth of the background, see Fig. 1. The depth distortion depends on a multitude of variables, such as the real depth of the background, angle and position of the refracting surfaces (i.e. pose and geometry of the transparent object), camera position, refraction index and thickness of the material. As the main purpose of our approach is to find unknown transparent objects, only very weak assumptions about their geometry can be made. Therefore, it is impossible to predict or model the expected background distortion. For instance, even a bottle refracts a light ray four times on its inner and outer surfaces. Due to these considerations, we do not try to model the expected distortion, but rather focus on detecting *any* background distortion. This approach is also more robust when incorrect or no depth data are measured, and it can deal with additional depth distortion effects like specular reflection.

For active sensors, such as the Kinect, refraction on the object does not only occur for incoming light rays (traveling towards the camera), but also for outgoing light rays (from the projector). Like that, an already distorted pattern may be projected onto the background. Depending on camera-projector baseline and scene geometry, refraction at the transparent object may affect the measurement in three ways: Distortion of incoming rays, of outgoing rays or in both directions. In addition, due to the strength of the projected rays, a significant portion is reflected on the transparent surface, leading to specularities in the IR image.

The Kinect sensor requires the projected pattern to be intact in a certain local neighborhood for successful matching. This is the case if refraction along a smooth surface is the dominant effect. However, if the pattern is distorted by too many effects – such as diffuse reflectance, attenuations or sub-surface scattering – pattern matching will fail, yielding an invalid depth (or hole in the depth map). Objects exhibit different dominant effects in different regions of the surface and depending on the camera pose. In a single view, the depth image of the transparent object is often "sparse", i.e. (distorted) depth data are only available in some areas. Thus, measurements from multiple viewpoints must be combined.

4. METHOD FOR TRANSPARENCY DETECTION

The detection method is based on a search for geometry distortions caused by transparent objects. These distortions are geometrically inconsistent from different viewpoints, as they depend on multiple variables discussed above. The sensor is moved along a trajectory which should ensure a broad coverage of viewpoints onto the relevant scene parts (space of interest, Fig. 2). On a robot, this step could be performed during navigation or while positioning the arm. During the movement, the expected, stable scene geometry is predicted from past measurements (background model) and compared to the current observation.

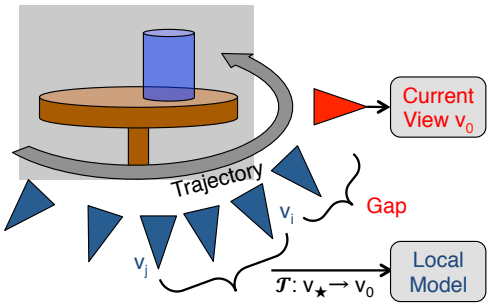


Fig. 2: Proposed setup: While the sensor is moving around a transparent object (blue) in the scene, a background model is generated and compared to the current observation.

4.1. Local Background Model

A local scene model, yielding a geometry estimate along with a reliability measure, is built based on depth maps taken from multiple viewpoints. These viewpoints are located on the sensor trajectory, close to the current observation both in time and space. We work with 8 views roughly 0.02m apart along a distance of 0.2m. Compared to a global model, this approach offers several advantages: The model is generated online, there is no need for a separate, time-consuming exploration step, changes in the scene cause only local distortions and there is no need for global geometric consistency.

There are a number of error sources, which must be dealt with by the model. First of all, the Kinect sensor exhibits a certain noise level over time and space. Under ideal conditions (e.g. a bright matt planar surface), as shown in [7], the noise can be modeled by a Gaussian with standard deviation σ_k dependent on depth z (in meters). Furthermore, there are small tracking and calibration errors which are dominant for small depths and modeled by observation as a Gaussian with σ_t. Together, these components determine the standard deviation of the expected minimum sensor noise $\mathcal{N}_{\sigma_s}$:

$$\sigma_s(z) = \sqrt{\sigma_k^2 + \sigma_t^2} = \sqrt{\left(\frac{1}{2} 2.9 \cdot 10^{-3} z^2\right)^2 + 0.01^2} \quad (1)$$

On the other hand, scene parts such as edges, fine structures or certain material types exhibit a much higher noise level. For instance, depth edges are depicted frazzled and flicker between background and foreground over time due to the matching neighborhood used by the sensor. In other cases, the sensor yields mostly invalid data. This is the case for surfaces hit by the projected rays with a very acute angle, some active light sources, surfaces with a very low albedo, scene parts which are beyond the measurement range as well as for transparent materials. Even though the (few) valid observations of these scene regions might be consistent, their reliability is low. Finally, errors may be caused by jumps in the estimated pose or losses-of-track which temporarily invalidate all measurements.

A model for each current viewpoint $c = v_0$ is built from depth maps taken nearby at $v_i, \dots v_j$ with $j > i, i > 0$. Depth maps are projected into the view c, see Fig. 2, using poses obtained by the tracker (Sec. 2). Hence, $(j - i + 1)$ measurements are available for each depth pixel whereof some provide invalid data, leaving a lower number of valid measurements denoted n. Assuming a Gaussian distribution, a mean depth $\overline{D_c}$ and a standard deviation $\hat{\sigma}_c$ can be estimated with a maximum likelihood estimator. The estimation bias of $\hat{\sigma}_c$ is corrected by $\sigma_c = \sqrt{\frac{n}{n-1}}\hat{\sigma}_c$. As n is quite low, it is infeasible to estimate more complex models – such as a Gaussian mixture model – even though they might be more realistic.

The observed Gaussian distribution $\mathcal{N}_{\sigma_c}$ is compared to the expected error model $\mathcal{N}_{\sigma_s}$ given by Eqn. (1). Reliable regions in the scene exhibit a standard deviation σ_c which is lower or equal to the expected error. The reliability is quantified with a score determined by comparison of the two distributions using the squared Hellinger distance. This distance is commonly used to quantify the similarity between two probability distributions, see [8]. For two Gaussians with equal mean, it is calculated according to $H^2 = 1 - \sqrt{\frac{2\sigma_1\sigma_2}{\sigma_1^2+\sigma_2^2}}$. The reliability score is evaluated per pixel as follows:

$$R_c = \ell\left(1 - wH^2\right) = \ell\left[1 - w\left(1 - \sqrt{\frac{2\sigma_s\sigma_c'}{\sigma_s^2 + \sigma_c^{2\prime}}}\right)\right] \text{ if } n \geq 3 \tag{2}$$

$$\text{with } \sigma_c' = \max\left(\sigma_c, \sigma_s\right)$$

The value of the score is 1 for reliable regions where $\sigma_c \leq \sigma_s$ and 0 for $n < 3$ or very unreliable regions. The weight w is chosen such that R_c drops to 0 for $\sigma_c \approx 4\sigma_s$. Function ℓ limits the lower bound of the score value to 0 in case of large errors. An erosion operation is applied to the reliability image R_c in order to add a margin to the unstable regions.

The views $v_i, \dots v_j$ are taken in a dense fashion from the recent sensor trajectory. A gap of about 16cm is maintained between the closest view v_i and the current view v_0 in order to avoid model distortions in the same image area as in the current observation, caused by the transparent object itself. For instance, assume the camera turns around the space of interest, as depicted in Fig. 2. The most recent views are distorted by the transparent object in almost the same image area as where the object is currently seen. Older views, on the other hand, provide a valid estimate of the current background, as the distortion from a transparent object is projected to other image regions, see the "shadow" in Fig. 3 (center).

The model for the current view consists of $\left(\overline{D_c}, \sigma_c', R_c\right)$, which is a straight-forward and sufficient representation of the expected scene geometry at the current viewpoint, merging information from past views $v_i, \dots v_j$. It can be directly compared to the depth map at $c = v_0$. Processing of depth maps is fast, especially compared to operations on volumetric or point-based data. While many scene reconstruction methods try to find the best guess in noisy data, we need to suppress any unstable regions, in order to allow for a reliable rating of the current observation.

4.2. Detection of Transparency

Transparency is detected by comparing the current depth observation z to the scene model while moving the sensor around the space of interest. The comparison is performed at regular intervals (such as 2cm) and should be carried out over a preferably large range of different viewpoints. Given the model $\left(\overline{D_c}, \sigma_c', R_c\right)$ and the observation $z' = \left|z - \overline{D_c}\right|$, an error signal is derived from the probability $p\left(\left|Z \sim \mathcal{N}_{\sigma_c'}\right| > z'\right) = 1 - \text{erf}\left(\frac{z'}{\sqrt{2}\sigma_c'}\right)$, which identifies observations that contradict with the model. Noise is suppressed and stability is increased by only considering high probabilities, using a mapping of $p \subset [\theta, 1.0] \rightarrow [0, 1]$ and $p < \theta \rightarrow 0$ with $\theta = 0.9$. Together with the model reliability R_c this yields the error signal:

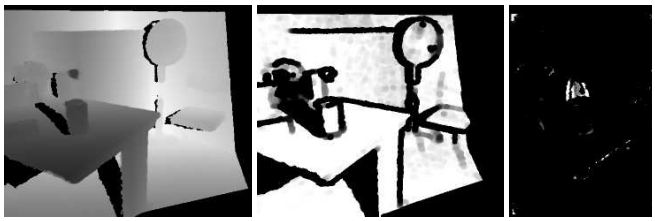


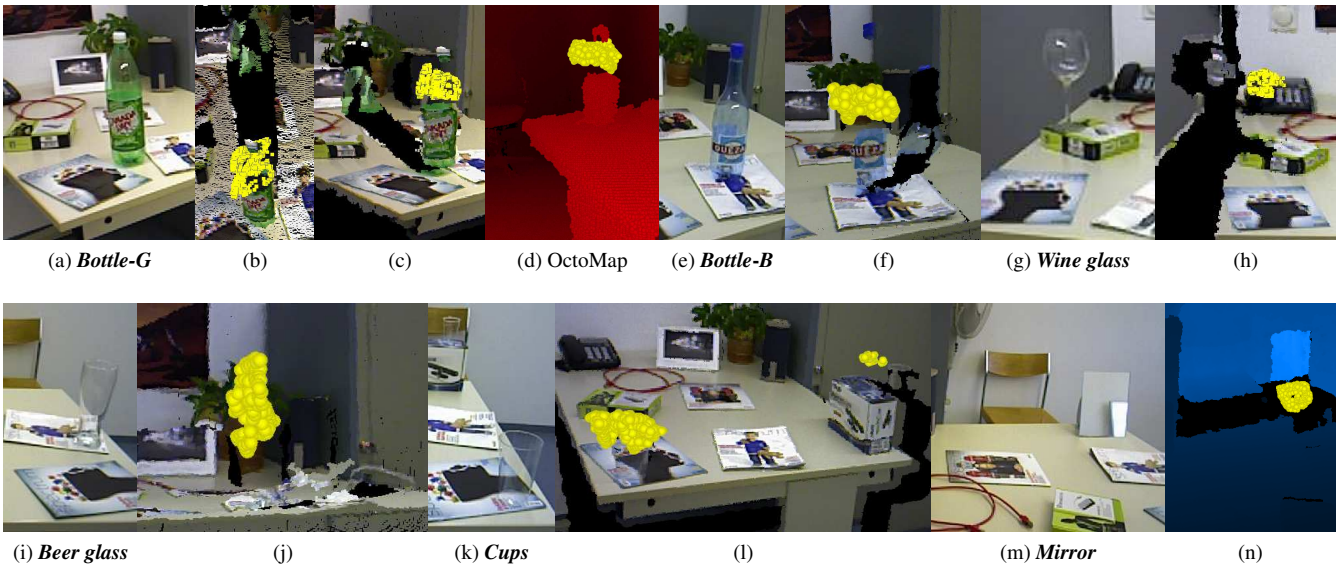
Fig. 3: Model depth $\overline{D_c}$ (*left*), reliability R_c (*center*) and cropped error image E_c (*right*), all shown in the current view v_0. The model is generated from past views taken right of v_0, leaving a region of low reliability that appears as a shadow left of the object. This scene is also shown in Fig. 4a.

$$E_c = \mathcal{O} \cdot R_c \cdot \ell\left[\frac{1}{1-\theta}\left(1 - \text{erf}\left(\frac{z'}{\sqrt{2}\sigma_c'}\right) - \theta\right)\right] \tag{3}$$

The boolean term $\mathcal{O}$ determines whether the observed 3D point was already in the field of view during model generation. Like that, an object which suddenly comes into view of the camera does not trigger an error. A large error signal identifies geometric inconsistency caused by transparent objects, see Fig. 3: The model predicts the depth $\overline{D_c}$ of the bottleneck with a high certainty R_c at the background, which is *not* confirmed by the current observation. Thus, this region triggers a large error signal E_c. The error signal does not trigger on inconsistent regions caused by static effect other than transparency or specular reflection, as those regions are assigned a low value of R_c during model generation. Moving objects, however, would result in a large value of E_c. Yet, there is some tolerance to this case, as the error will not concentrate in a certain region of space. In general, however, we assume static scenes.

If a large error signal is found, no assumptions can be made about the real depth at the affected pixel. Each pixel corresponds to a ray in space, and all we know is that there is something unreliable along that ray. Therefore, we reverse-project the 2D error signal into a 3D "transparency volume" which covers the space of interest, using the known intrinsic and extrinsic camera parameters. While the camera moves along its trajectory, error rays with $E_c > 0$ are cast into the volume, accumulating at voxels in or near the unreliable object. If enough viewpoints are integrated, ideally, the convex hull of the object is regenerated within the transparency volume. For a good location and rough shape estimate of the object, the viewing angles of the space of interest should cover 90° or more. Measurements with large erroneous regions are ignored, as they are typically caused by scene movements or tracking errors.

Clusters of high values in the transparency volume correspond to a single object or a connected transparent part within that object. The edge voxels of a cluster are used to generate a pointcloud which shows the rough shape of the object.



(a) ***Bottle-G*** (b) (c) (d) OctoMap (e) ***Bottle-B*** (f) (g) ***Wine glass*** (h)

(i) ***Beer glass*** (j) (k) ***Cups*** (l) (m) ***Mirror*** (n)

Fig. 4: The proposed approach is tested for a typical office scene with 6 different objects depicted in (a), (e), (g), (i), (k) and (m). Detected transparent parts are visualized as yellow pointclouds, together with colored raw data from the sensor. Non-transparent parts of the bottles in (b), (c) and (f), such as the labels, are visible in the raw pointcloud. The upper transparent parts, however, are measured incorrectly and distort the background (see area near the plant in (c)). (d): A 3D model of the scene built with OctoMap [9] is shown in red, see text. (n): The mirror shows the opposite wall (bright blue area), which is also detected as a geometric inconsistency.

5. EXPERIMENTS AND RESULTS

The method is evaluated in a cluttered office scene. Different transparent objects made of glass or plastic are placed onto the table, and depth maps are acquired along a trajectory. The space of interest covers the space on and above the table and is set to $1m^3$. The camera pose is estimated continuously using the tracker from [6]. As the tracker offers an online learning mode, it is not required to build any model of the scene beforehand.

Reconstruction results are shown in Fig. 4. The detected transparent regions are depicted as yellow "transparency" pointclouds, together with a registered colored pointcloud obtained directly from the Kinect. As the viewpoint is not equal to the camera position, there are unknown areas which appear in black. The two bottles in 4a) and 4e) exhibit surface areas with Lambertian reflection at the label which are measured correctly by the Kinect. Our method accurately finds the transparent parts above the label. The objects in 4e) and 4g) exhibit relatively fine transparent structures which are too small for detection – only the larger parts are reconstructed. Furthermore, an example of specular reflection is shown in Fig. 4n). Here, the sensor measures the depth of the scene shown in the mirror, which is also geometrically inconsistent.

Finally, in Fig. 4d) we show the result of an existing approach for the reconstruction of 3D models called OctoMap [9] as a red pointcloud. The approach combines multiple views to reconstruct Lambertian scenes in a compact volumetric representation. Here, we work with a voxel resolution of 5mm and the camera pose is taken from the above-mentioned tracker. The scene is accurately reconstructed, including the opaque label of the bottle. However, the transparent part of the bottle is missing. The detected transparency (yellow pointcloud) complements the 3D model of the scene.

Our approach is implemented in C++ using the ROS framework [10] and tested on a modern Intel i7 machine with 4 real cores. Equations (2) and (3) are approximated by a quadratic polynomial for faster processing. The measured runtime for model generation and detection is 180ms for 8 model views and 250ms on average for processing of the transparency volume running in a parallel thread. Using the suggested frame distance of 2cm, realtime processing is possible for a motion speed around $8\frac{cm}{s}$. The tracker [6] runs in parallel at camera framerate, partly using the GPU.

6. CONCLUSION

In this work, a method to reconstruct transparent objects such as bottles or glasses in unstructured environments is presented. The approach is based on identifying inconsistent depth measurements while moving a visual depth sensor around the scene. These inconsistencies are caused by refractive (or reflective) effects on the surfaces of the objects. Experiments are presented, showing that a location and rough shape estimate can be acquired for various objects made of glass or plastics. Detection is limited to transparent objects with smooth and curved surfaces where refractive effects dominate.

In future work, the results of this approach will be fused with existing 3D reconstruction methods. Like that, transparent objects can be more accurately represented as a connected entity in the scene model. Additionally, it can be avoided that reconstructed 3D models are distorted by transparency. Furthermore, it will be evaluated whether refraction effects can be detected on a lower level, using the raw IR images from the Kinect.

7. ACKNOWLEDGEMENT

This work was supported, in part, by Campus France, by the TUM Graduate School and by the DFG excellence initiative research cluster *Cognition for Technical Systems (CoTeSys)*, see `www.cotesys.org`.

8. REFERENCES

[1] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, and Andrew Fitzgibbon, "KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera," in *Proc. of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST)*, Santa Barbara, CA, USA, Oct. 2011, pp. 559–568.

[2] Ilya Lysenkov, Victor Eruhimov, and Gary Bradski, "Recognition and pose estimation of rigid transparent objects with a kinect sensor," in *Proc. of Robotics: Science and Systems*, Sydney, Australia, July 2012.

[3] Mario Fritz, Michael Black, Gary Bradski, Sergey Karayev, and Trevor Darrell, "An additive latent feature model for transparent object recognition," in *Advances in Neural Information Processing Systems (NIPS) 22*, Vancouver, BC, Canada, Dec. 2009, pp. 558–566.

[4] Ulrich Klank, Daniel Carton, and Michael Beetz, "Transparent object detection and reconstruction on a mobile platform," in *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 2011, pp. 5971–5978.

[5] Ivo Ihrke, Kiriakos N. Kutulakos, Hendrik P. A. Lensch, Marcus Magnor, and Wolfgang Heidrich, "Transparent and specular object reconstruction," *Computer Graphics Forum*, vol. 29, no. 8, pp. 2400–2426, 2010.

[6] Cedric Audras, Andrew I. Comport, Maxime Meilland, and Patrick Rives, "Real-time dense appearance-based SLAM for RGB-D sensors," in *Proc. of Australian Conference on Robotics and Automation*, Melbourne, Australia, Dec. 2011.

[7] Kourosh Khoshelham and Sander O. Elberink, "Accuracy and resolution of kinect depth data for indoor mapping applications," *Sensors*, vol. 12, no. 2, pp. 1437–1454, 2012.

[8] David Pollard, *A User's Guide to Measure Theoretic Probability*, Cambridge University Press, 2001.

[9] Kai M. Wurm, Armin Hornung, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard, "OctoMap: a probabilistic, flexible, and compact 3D map representation for robotic systems," in *In Proc. of the ICRA 2010 workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation*, Anchorage, AK, USA, May 2010.

[10] Morgan Quigley, Ken Conley, Brian P. Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng, "ROS: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, Kobe, Japan, May 2009.