ROSE 2008 - IEEE International Workshop on
Robotic and Sensors Environments
Ottawa, Canada, 17-18 October 2008

Grasping in Depth Maps of Time-Of-Flight Cameras

J. U. Kuehnle[1], Z. Xue[2], M. Stotz[1], J. M. Zoellner[2], A. Verl[1], R. Dillmann[2]

[1]Department Information Processing,
Fraunhofer Institute Manufacturing Engineering and Automation (IPA),
Nobelstrasse 12 – 70569 Stuttgart, Germany
Phone: +49 711 970 1861, Fax: +49 711 970 1004, E–mail: kuehnle@ipa.fhg.de.

[2]Department Interactive Diagnosis- and Servicesystems,
Research Center for Information Technology (FZI),
Haid-und-Neu-Strasse 10-14 – 76131 Karlsruhe, Germany
Phone: +49 721 9654 204, Fax: +49 721 9654 205, E–mail: xue@fzi.de.

Abstract – *The recognition and localization of objects in space is of fundamental interest in many robot vision applications, especially in those that are supposed to provide services to human beings. The trivial example of any such task is manipulation, i.e., providing a robot the means of handling objects. In this work, we discuss the chances and the problems experienced when using a time-of-flight camera as the only measurement device of an object recognizer. The localization is based on the best-fit of the geometric primitives within the objects considered (such as planes, cylinders, cones, or spheres). Such shapes are of special interest not only in recognition but also in grasp planning. We use a time-of-flight camera SwissRanger SR-3000 and localize a selected set of objects in order to grasp them with a dexterous robotic hand.*

Keywords – *Object Recognition and Localization, Time-Of-Flight Camera, Calibration, Robotic Manipulation*

Fig. 1. Experimental platform.

I. INTRODUCTION

Robot vision applications can choose from a variety of sensing systems. For object recognition and localization, there are several range imaging systems available that combine image processing with distance measurement aspects. One of the most widely used ranging principles is stereo vision. A stereo rig measures the disparity between corresponding points in two images taken from different viewpoints. The main issue in the stereo analysis of images is the correspondence problem. That is, the search for the correct match of feature points in the two images. Especially in scenes with few texturing or a lot of repeating patterns, correspondence often can not be achieved and thus, range information can not be obtained. Furthermore, trying to solve the correspondence problem is computationally expensive. Another widely used ranging device is the laser range finder. It utilizes a laser beam to determine the distance to an object. Since only one measurement can be taken at a time, the laser beam must scan over the scene in order to measure a robot's workspace for recognition purposes. However, scanning takes time.

In recent years, a novel type of solid state sensors emerged on the market allowing the capture of range information at video frame rates. Instead of providing the illuminance (i.e., gray-scale or color) as conventional CCD/CMOS chips do, such sensors measure the viewed scene with respect to distance. Surface texturing is not required, and ambient light is not crucial due to the illumination unit on the sensors. These features make time-of-flight cameras quite interesting to robot vision. However, since time-of-flight cameras are still somewhat prototypical, they usually do not reach the levels of accuracy required in some robot vision applications. A thorough calibration of the sensors is unavoidable in order to recover stable range information. Time-of-flight cameras are often utilized in obstacle avoidance or navigation tasks. However, there are also several attempts described to use the sensors in object recognition and localization [1].

In this paper we discuss an object recognizer that localizes objects by looking at the geometric primitives within the objects considered. The basic idea is that the model features (such as planes, cylinders, cones or spheres) usually carry enough information to make possible proper object recognition and localization needed for grasping with a robotic hand. After the

978-1-4244-2595-2/08/$25.00 ©2008 IEEE

object with geometric primitive shape is recognized, approach directions for the geometric primitive, along which the object can be better grasped by the robotic hand are automatically generated. In a grasping simulator "GraspIt!" [2], the grasps are simulated and its grasp qualities are evaluated. At the end, a grasp with high grasp quality which is also feasible for the robotic arm will be executed.

The paper is organized as follows. In Sec. II, we discuss characteristics of time-of-flight cameras and propose calibration and filtering procedures to enhance the quality of distance measurements. In Sec. III, we describe the best-fitting and segmentation of geometric primitives in point clouds. In Sec. IV, we present the use of geometric primitives when computing grasps. And in Sec. V, we evaluate our object recognizer that localizes through best-fit of the geometric primitives contained within the object under consideration.

II. TIME-OF-FLIGHT CAMERAS

A time-of-flight camera measures the surrounding scene with respect to distance. Following the time-of-flight measurement principle, the sensor emits modulated light, which is reflected by objects in the scene and projected onto the chip. In each pixel, the reference signal is correlated with the modulated light that it receives. The distance d of an object along the line of sight of any pixel is half the distance the light beam travels in the corresponding direction. Since the distance d is recovered in each pixel, the sensor yields the surface of the objects in the scene observable in the sensor's viewpoint. That is why precisely speaking it only captures a 2.5D depth map, not 3D data. It is worth emphasizing that a time-of-flight camera provides range information at video frame rates without the need to scan the viewed scene.

A. Characteristics of Time-Of-Flight Cameras

Most manufacturers of time-of-flight cameras—such as PMD Technologies GmbH, MESA Imaging AG or Canesta Inc.—use amplitude modulation by a sinusoidal on eye safe near-infrared light emitting diodes (LED). Complementary to the range information, the sensors also provide intensity and amplitude information as near-infrared images of ambient illuminance and reflectance. Ambiguity in range measurement due to aliasing occurs when the phase-shift observed exceeds 2π. The non-ambiguity or effective range d_{max} satisfies $d_{max} = \frac{c}{2f_{mod}}$, where c denotes the speed of light and f_{mod} denotes the modulation frequency.

The intensity and the amplitude have a significant influence on the accuracy of the distance measurement d. The standard deviation σ_d of the distance measurement is Poisson distributed and given by [3]

$$\sigma_d = \frac{d_{max}}{\sqrt{8}} \frac{\sqrt{I}}{2A}, \tag{1}$$

where the intensity I is the average amount of ambient light and light of the sensor's illumination and the amplitude A is the signal's amplitude influenced by the objects distance and reflective properties. σ_d suggests that I should be low and A should be high in order to obtain proper distance measurements. Varying the integration time is the only means to influence the overall intensity and amplitude of the measurements. Low integration times usually result in noise due to bad signal-to-noise ratio whereas high integration times usually blur or even cause saturation on the chip. Several authors propose ways to adjust the integration time or to combine depth maps taken at various integration times [4] [5].

B. Calibration of Time-Of-Flight Cameras

Since time-of-flight cameras are still somewhat prototypical, there are numerous factors that affect the accuracy of distance measurements. Lots of research is being done to understand the effects and to compensate the errors introduced [6], [7], [8], [4]. Distance errors may be caused by internal effects such as noise (thermal, electronic, photon shot), propagation delay in the chip's circuits, the exact form of the diode's signal or lens distortion as well as by external effects such as ambient light, the reflective properties and the complexity of the scene or viewed scattering due to objects near the sensor. Although some of these effects are well described in the literature, a complete model to correct all of them is still missing.

In order to obtain stable range measurements we propose the following three stage calibration procedure. The first and second stages are well described in the literature. In the third stage we may assume parts of the sensor's environment to be known. We use such parts as reference and best-fit the observed measurements to it.

Stage 1: In order to get rid of the distortion of the lens and the misalignment of the chip, we calibrate the time-of-flight camera with the MATLAB Camera Calibration Toolbox [9] using 50 intensity images. By means of the estimated intrinsic and extrinsic camera parameters an orthogonal point cloud in Cartesian coordinates corresponding to the radial distance measurements of the depth map can be computed.

Stage 2: A time-of-flight camera usually possesses considerable variability across its depth sensing range due to internal and external influences. As a consequence, a wall may appear vaulted. To correct the errors, we perform a per-pixel correction. Taking several depth maps of a white wall with the sensor mounted towards it in various defined distances, a nominal/actual value comparison is performed. Within 0.5m to 1.25m an affine relation between nominal and actual values may suffice to approximate the errors properly (beyond this range, a more complex relation is required [6]). In each pixel, a linear regression on the nominal and actual values provides a lookup value with both an offset and a multiplier. The lookup values constitute a lookup table which is used to correct the radial distance measurements:

$$c_{i,j} = m_{i,j} d_{i,j} + o_{i,j}, \tag{2}$$

where the indices i, j specify the pixel in the depth map, $d_{i,j}$ the measured and $c_{i,j}$ the corrected distance, $o_{i,j}$ the offset and

$m_{i,j}$ the multiplier. Furthermore, we clamp $c_{i,j}$ to the ambiguity range $[0, d_{max}]$. We call this procedure an offline correction of the depth map since it has to be calibrated only once.

Stage 3: In the strict sense though, the depth correcting lookup table is only valid under the internal and external conditions present during its calibration. Influencing conditions are for example temperature, ambient light, surface reflectance and angle of intrusion. As a consequence, if any of these conditions changes, then reliable correction of the depth map by means of the lookup table can not be guaranteed and even may cause further errors. We address this problem by yet another per-pixel correction complementing the lookup table. Assuming parts of the sensor's environment to be known (for instance, a table board), we transform the observed distance measurements in such a way that it best-fits the known parts of the environment. Of course, we must account for possible occlusion of known parts by unknown objects within the scene. We call this procedure an on-line correction of the depth map since it has to be calibrated for each individual depth map.

C. Median of Least Span Filter and Scalable Span Filter

The depth measurements of time-of-flight cameras experience considerable noise. Therefore, smoothing of the data is essential. Measurement errors occur also on jump edges, i.e., gaps in distance between neighboring pixels due to occlusion. The measured distances at such jump edges constitute pseudo measurements through mixture of reflected light of foreground and background objects.

Classical edge-finding filters not only filter the pseudo measurements but also valuable surface information of the objects. Their thresholds can be determined experimentally such that the jump edges are found more or less reliably. However, the threshold is fixed and does not vary with the distance range considered. Furthermore, the meaning of the threshold with respect to a depth map is not obvious. More sophisticated edge-finding filters in depth maps are described in the literature but usually are computationally quite expensive [10]. The authors of [11] propose a simplified jump edge filter that thresholds with respect to the opposing angles of the triangle spanned by the focal point and two neighboring data points.

A similar result is achieved by the following two stage filter procedure which combines an edge-enhancing smoothing filter, called median of least span filter, and a jump edge-finding filter, called scalable span filter.

Stage 1: The median of least span filter is an edge-enhancing smoothing filter. It follows the value-and-criterion filter structure [12]. Such a filter has a 'value' function, V, and a 'criterion' function, C, each operating on the raw data using a common structuring element, $\tilde{N}$ (for instance, the same window). To obtain the filtered data, a 'selection' function, S, evaluates the criterion C within a structuring element N and assigns the value V that corresponds to the selected pixel. Let $r(\cdot)$ denote the raw measurement and let $f(\cdot)$ denote the filtered measurement. Let $v(\cdot)$ and $c(\cdot)$ denote the value V and the criterion C within $\tilde{N}$, respectively. The value-and-criterion

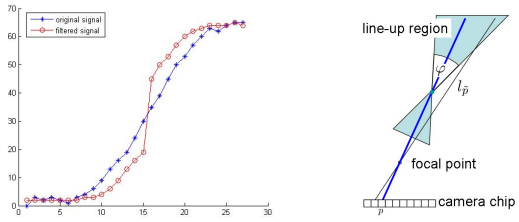

Fig. 2. Left: Edge-enhancing filter response of a one dimensional signal using a median of least span filter with a window of size 3. Since time-of-flight cameras smooth jump edges considerably, the proposed edge-sharpening makes jump edge-finding more robust. Right: Principle of the jump edge-finding filter centered at pixel p.

filter structure is defined by

$$v(\tilde{p}) = V\big(\{r(\tilde{N}_{\tilde{p}})\}\big), \tag{3}$$

$$c(\tilde{p}) = C\big(\{r(\tilde{N}_{\tilde{p}})\}\big), \tag{4}$$

$$f(p) = v\Big(\big\{\tilde{p} \in N_p : c(\tilde{p}) = S\big(\{c(N_p)\}\big)\big\}\Big), \tag{5}$$

where $\tilde{N}_{\tilde{p}}$ denotes the structuring element $\tilde{N}$ centered at $\tilde{p}$ and N_p that of N centered at p. However, the evaluation of $c(\tilde{p}) = S\big(\{c(N_p)\}\big)$ may select more than one pixel $\tilde{p}$ of some pixel p with raw measurement $r(p)$ and thus, the filter may yield several filtered measurements $f(p)$. In this case, some method that chooses among these measurements is required. One method could be to average the values $v(\tilde{p})$ of all the selected pixels $\tilde{p}$. Another method could be to take the selected pixel $\tilde{p}$ such that the filtered measurement $f(p) = v(\tilde{p})$ and the raw measurement $r(p)$ differ the least, that is $|f(p) - r(p)| \rightarrow \min$. The value-and-criterion filter structure allows the use of different linear and nonlinear operators and retains much of the structuring control of the standard morphological operators (such as opening or closing) [12]. In case the value operator V is chosen to be the median, the criterion operator C is chosen to be the span (i.e., $\max - \min$) and the selection operator S is chosen to be the minimum, we obtain the median of least span filter. Since the span of an inhomogeneous region containing a jump edge is typically wider than that of a homogeneous region, the median of least span filter sharpness inhomogeneous jump edges by means of averaging over homogeneous neighborhoods (see Fig. 2).

Stage 2: The scalable span filter is a jump edge-finding filter on depth maps. It is a span filter with a threshold that scales with distance. A data point is marked as jump edge in case its neighborhood contains measurements that line up with the line of sight of the corresponding pixel. The degree of line-up is controlled by an apex angle, φ. We describe the principle of the jump edge-finding filter using the illustration in Fig. 2 in two dimensions. We note, the line-up regions become line-up cones in three dimensions. Instead of computing opposing angles of the triangle spanned by the focal point and data points of pixel p and any of its neighbors, $\tilde{p}$, we consider the lengths of the line segments $d_{\tilde{p}}(\varphi)$ and $D_{\tilde{p}}(\varphi)$ between the focal point and the

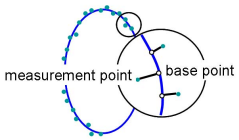

Fig. 3. Best-Fit Principle.

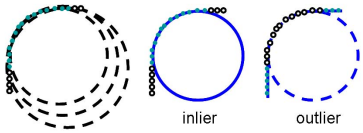

Fig. 4. Segmentation in inlier and outlier.

intersection points of the line of sight of $\tilde{p}$, $l_{\tilde{p}}$, with the line-up region. Let $d_p(\varphi) = \min_{\tilde{p}\in N_p}\{d_{\tilde{p}}(\varphi)\}$ denote the minimum and $D_p(\varphi) = \max_{\tilde{p}\in N_p}\{D_{\tilde{p}}(\varphi)\}$ the maximum length of the line segments in the neighborhood of pixel p. Regarding the interception theorem, $d_p(\varphi)$ and $D_p(\varphi)$ scale with respect to the distance measurement $r(p)$ at p. Therefore, it suffices to compute lookup tables with values $d_p(\varphi)$ and $D_p(\varphi)$ assuming $r(p) = 1$. A pixel p at distance $r(p) \neq 1$ is regarded a jump edge in case any of its neighboring pixels, $\tilde{p}$, lie within the line-up region, i.e., $r(\tilde{p}) < r(p)d_p(\varphi)$ or $r(\tilde{p}) > r(p)D_p(\varphi)$.

III. BEST-FIT METHOD AND SEGMENTATION OF GEOMETRIC PRIMITIVES

Best-fitting of geometric primitives (such as planes, cylinders, cones or spheres) in measurement point clouds is achieved via non-linear least-squares [13]. The optimization problem is depicted in Fig. 3. Given a hypothesis on the feature to be fitted, the model parameters, a, consisting of shape parameters (for instance, the radius of a cylinder) as well as of position parameters (translation and rotation) are estimated. Let $X = (X_1, \ldots, X_n)$ denote n data points and let $X^a = (X_1^a, \ldots, X_n^a)$ denote the n corresponding base points on the model feature with parameters a. Let $d^a = (d_1^a, \ldots, d_n^a)$ denote the Euclidean distances $d_m^a = \|X_m - X_m^a\|$, where $m = 1, \ldots, n$. The objective function is given by $\sigma_0^2 = d^{a\prime}P'Pd^a$ with some weighting matrix $P'P$. The estimation of the parameters a of the model feature F becomes:

$$\min_a \min_{\{X_m^a\}_{m=1}^n \in F} \sigma_0^2\big(\{X_m^a\}\big). \tag{6}$$

While best-fitting is in progress, the measurements can also be segmented into inlier (i.e., data points belonging to the model feature F to be fitted) and outlier (i.e., data points not belonging to F). As segmentation criterion, we use the Euclidean distance. More precisely, the error of fit is assumed to be equal to an order of magnitude of the noise level of the measurement device (see Fig. 4). The algorithms for best-fitting and segmentation of geometric primitives can operate in manual mode. Given a measurement point cloud, the user specifies the model feature to be fitted and selects a measurement point with which the best-fit is initiated.

An analysis of the principal curvatures within a surrounding region of a selected measurement point is used in automatic identification of geometric primitives within point clouds [14]. Hypotheses on model features to be fitted to the point cloud are established, and the model feature that best-fits the measurements is determined. That way, the algorithms for best-fitting and segmentation of geometric primitives can also operate in automatic mode. Using randomly chosen measurement points to initiate the best-fit on hypothesized model features, a point cloud is segmented into all its identifiable geometric primitives.

Object recognition and localization based on geometric primitives can be achieved by operating the best-fit and segmentation algorithms either in automatic or in manual mode. In automatic mode, the whole scene viewed by the sensor is segmented into all its identifiable geometric primitives. Separate geometric primitives may be combined to form composite model features by evaluating the interrelations of the identified geometric primitives (for instance, discs may be combined with a cylinder to represent its top and bottom, since only its hull is best-fitted). In manual mode, the scene is separated into patches. For any patch, the user specifies the composite model feature (with fixed shape parameters) and either selects a measurement with which the best-fit is initiated or provides an initial guess of the position. The best-fit and segmentation algorithms estimate the shape and position parameters of geometric primitives within measurement point clouds that are used in classification.

IV. GRASP PLANNING

The shape (sphere, cylinder, cone or box), the position and the orientation of a recognized and localized object are used to plan feasible grasps. The grasping simulator "GraspIt!" [2] is optimized to plan grasps of simple geometric primitives within a few seconds. The method is based on the observation that humans unconsciously simplify the grasping task to selecting one of only a few different preshapes appropriate for the object and for the task to be performed. From the object shape, starting positions and approach directions are generated, from which the object can be grasped well by the robotic hand. In the simulation, the hand in predefined preshape is placed at starting position, moves along one of the approach directions towards the object and the fingers close around the object. The contact points between the hand and the object are collected, with which the grasp quality is evaluated. "Largest sphere in grasp wrench space" is used as grasp quality, so that the grasp with higher quality can resist with independence of the pertur-

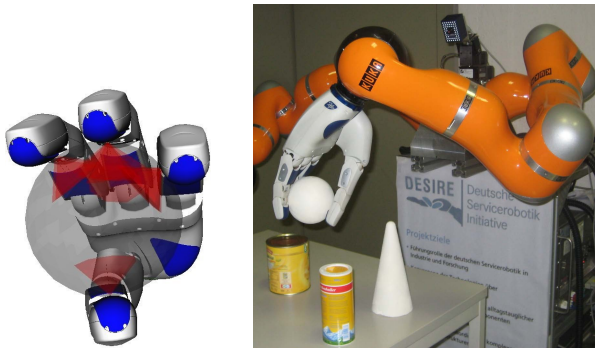

Fig. 5. Computed grasp and its real execution.

bation direction [15]. To execute a planned grasp, the grasp pose relative to the object must be reachable by the robotic arm. Besides the reachability of the arm, collisions between the robot and the environment should also be avoided. These two constraints are used to filter the approach directions in the simulation in two ways. The starting position relative to object is transformed into arm coordinate system, and checked by the arm inverse kinematic algorithm, whether the position is reachable for the arm. If not, the grasps from this starting position will not be simulated. Secondly, ray intersection from the starting position along the approach direction toward the object is checked. If the ray intersects firstly with other obstacles near the object to be grasped, the hand would collide with these obstacles. This approach direction does not need to be tested. In this work, the localized geometric primitives that should not be grasped are treated as obstacles to be avoided. After a grasp is planned in the simulation, a motion planning system is used to bring the arm collision-free to the grasp position. With a grasp force optimization step, the object can be grasped firmly and safe [16].

V. EXPERIMENTAL RESULTS

A. Configuration and Task

In the work presented, a SwissRanger SR-3000 manufactured by MESA Imaging AG is used. It provides 176x144 pixels on a custom designed CCD/CMOS chip with up to 25 frames per second. Using a sinusoidal with a frequency of $f_{mod} = 20$MHz, it measures a distance range of $d_{max} = 7.5$m. The effective resolution in an operation distance of 1.5m is about 6mm lateral and 8mm in depth. For the sake of the work presented, any other time-of-flight camera with about 20.000 pixels could have been used. All of them exhibit similar performances.

The time-of-flight camera is mounted on a pan/tilt unit. We assume the robot to be situated at a table such that the object on it are within the viewing frustum of the sensor as well as within the working space of the manipulator. We assume the position of the table board to be known. Given the configuration of our experimental platform, the sensor observes the objects at a distance of about 0.75m tilted about 35° towards the table. The robot is equipped with two seven degrees of freedom (DOF) arms (KUKA LBR3) with each an Schunk Anthropomorphic Hand (SAH) attached. We establish a data base of 8 objects whose shapes consist of geometric primitives (2 spheres, 3 cylinders, 1 cone and 2 boxes). We consider the task of grasping objects off the table.

B. Evaluation of Calibration

It is reported in the literature that the distance accuracy of the camera chips available is limited to 5-10mm due to chip design issues (in ideal conditions) [17]. Since our robot's environment is anything but ideal (varying ambient light, varying surface reflectance, varying angle of intrusion, ...), the measurement errors we experience are typically well beyond that limit.

By applying the first and second stage of our calibration procedure planes in the sensor's environment are corrected to be represented by plane patches in the measurement point cloud. Under the internal and external conditions present during calibration, the measurements are corrected with an error less than 5mm. However, when viewing different table boards with differing reflective properties at different angles of intrusion we find that though represented by plane patches the corresponding measurements are typically shifted and tilted in depth (sometimes even 7.5 cm). Taking this observation into account, we execute the third stage of our calibration procedure with reference to the table board that is assumed to be known. In order to do so, the inlier/outlier segmentation of our best-fit algorithms for planes makes the calibration possible even though objects may be placed on the table. Performing the plane best-fit on approximately 20.000 measurement points takes about 100msec.

As a result of our calibration method, any table board with objects we tested was accurately measured with an error less than 5mm.

C. Evaluation of Jump Edge-Filtering

The proposed jump edge-finding filter performed as well as other jump-edge finding filters described in the literature. The key feature of the filter is that applying it just requires simple operators (edge-sharpening and thresholding with a lookup table that scales in depth).

D. Evaluation of Object Recognition and Localization

Due to the multipath problem of time-of-flight cameras [5] (i.e., direct and indirect reflections in the viewed scene can not be separated and may cause errors in depth measurement at edged surfaces), objects placed upright on the table board usually lean towards the viewing direction of the sensor. For instance, the measurements corresponding to a cylinder standing on the table board constitute a point cloud that may not be upright. We observe that best-fitting a cylinder to the measurement points may even yield a cylinder axis that comprises an angle of about 10° with the table board's surface normal. The

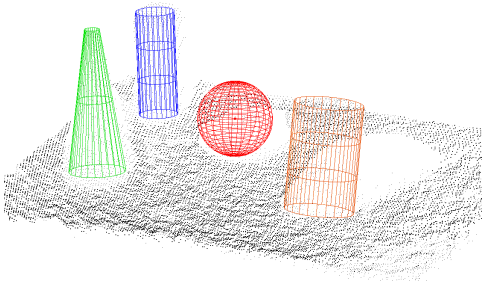

Fig. 6. Best-fit and segmentation of geometric primitives in measurement point cloud (manual mode).

extent of the error varies with respect to surface reflectance and angle of intrusion. In other words, the estimation of the rotation is expected to be erroneous due to the multipath problem and therefore, we correct the estimate such that the object stands steady on the known table board. Concerning the estimation of the translation, we experience another measurement error, especially with curved surfaces, that squeezes the measurement points representing objects on the table board. For instance, a cylinder of radius 5cm may be displayed as cylindrical point cloud with a radius of only 4.25cm, a sphere of radius 4.5cm may be displayed as spherical point cloud with a radius of only 4cm. The extent of the error again varies with respect to surface reflectance. The estimate of the translation may be erroneous in automatic mode because the estimated shape parameters may be erroneous and in manual mode because with the fixing of the shape parameters more local minima occur.

The automatic mode requires a reasonable amount of surface points to establish a model feature hypothesis. Given the density of measurements, their noise level and other errors of time-of-flight cameras, we notice that the model features processable by the algorithms need to be 2-3 times the size of the objects manageable by our robotic hand. Running several tests on such objects in automatic mode we observe translation errors of about 1.5cm. Avoiding the classification step in manual mode, we observe translation errors of about 2.5cm when running tests on objects manageable by the robotic hand.

E. Evaluation of Manipulation Task

Running several tests we are able to grasp the objects found during recognition. Fig. 6 shows the localization results corresponding to the scene in Fig. 1.

VI. CONCLUSION

Time-of-flight cameras are interesting to robot vision. However at the present stage of sensor design, reliable calibration and filtering are key issues in any application. This paper proposes a calibration procedure that performs very well in applications like grasping objects off a table (even in varying conditions). Also, a simple jump edge-finding filter is presented.

Objects can be localized by means of their shape. In grasp planning, geometric primitives are of special interest. This paper discusses the chances and the problems experienced when recognizing geometric primitives within measurement point clouds of time-of-flight cameras. Although feasible, we believe that further sensor developments are in need in order to reliably best-fit and segment geometric primitives in data of time-of-flight cameras.

ACKNOWLEDGMENT

This work was funded as part of the research project DESIRE by the German Federal Ministry of Education and Research (BMBF) under grant no. 01IME01A and 01IME01E.

REFERENCES

[1] S. Hussmann and T. Liepert, "Robot vision system based on a 3D-TOF camera," *Instrumentation and Measurement Technology Conference Proceedings, 2007. IMTC 2007. IEEE*, pp. 1–5, 2007.

[2] A.T. Miller and P.K. Allen, "Graspit! a versatile simulator for robotic grasping," *IEEE Robotics & Automation Magazine*, vol. 11, no. 4, pp. 110–122, 2004.

[3] R. Lange, *3D Time-of-Flight Distance Measurement with Custom Solid-State Image Sensors in CMOS/CCD-Technology*, Ph.D. thesis, Universität Siegen, 2000.

[4] A. Sabov and J. Krüger, "Improving the data quality of PMD-based 3D cameras," in *12th International Fall Workshop on Vision, Modeling, and Visualization (VMV 2007)*, 2007.

[5] S. May, B. Werner, H. Surmann, and K. Pervölz, "3D time-of-flight cameras for mobile robotics," *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pp. 790–795, 2006.

[6] T. Kahlmann, F. Remondino, and H. Ingensand, "Calibration for increased accuracy of the range imaging camera swissranger," 2006.

[7] M. Plaue, "Analysis of the PMD imaging system," Tech. Rep., University of Heidelberg, Germany, 2006.

[8] S.A. Guomundsson, H. Aanaes, and R. Larsen, "Environmental effects on measurement uncertainties of time-of-flight cameras," *International Symposium on Signals, Circuits and Systems*, vol. 1, pp. 1–4, 2007.

[9] J.Y. Bouguet, "Matlab calibration toolbox," http://www.vision.caltech.edu/bouguetj/calib_doc/.

[10] X. Jiang and H. Bunke, "Edge detection in range images based on scan line approximation," *Comput. Vis. Image Underst*, vol. 73, no. 2, pp. 183–199, 1999.

[11] S. May, D. Droeschel, D. Holz, C. Wiesen, and S. Fuchs, "3D pose estimation and mapping with time-of-flight cameras," in *IEEE/RS International Conference on Intelligent Robots and Systems(IROS), Workshop on 3D-Mapping, Nice, France*, 2008.

[12] M.A. Schulze and J.A. Pearce, "A morphology-based filter structure for edge-enhancing smoothing," in *ICIP (2)*, 1994, pp. 530–534.

[13] S.J. Ahn, *Least Squares Orthogonal Distance Fitting of Curves and Surfaces in Space*, Ph.D. thesis, University Stuttgart, 2004.

[14] S.J. Ahn, I. Effenberger, and L. Bolboaca, "Automatic feature identification in 3-d measuring data," in *8th Int. Symp. on Measurement and Quality Control in Production (ISMQC)*, 2004.

[15] C. Ferrari and J. Canny, "Planning optimal grasps," in *IEEE International Conference on Robotics and Automation*, 12-14 May 1992, pp. 2290–2295.

[16] Zhixing Xue, J. Marius Zoellner, and Ruediger Dillmann, "Automatic optimal grasp planning based on found contact points," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, 02-05 Jul. 2008, pp. 1053–1058.

[17] P. Seitz, "Unified analysis of the performance and physical limitations of optical range-imaging techniques," *In: Proceedings of the 1st Range Imaging Research Day at ETH Zurich*, pp. 9–19, 2003.