The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Automatic Observation for 3D Reconstruction of Unknown Objects using Visual Servoing

Guillaume Walck and Michel Drouin
UPMC Univ Paris 06, F-75005, Paris, France
{guillaume.walck,michel.drouin}@upmc.fr

Abstract— **This paper presents a method for the automatic observation of unknown objects. We aim at finding the position of the targeted object and at capturing multiple views of its shape using a single eye-in-hand camera. The main goal is modeling the unknown object via 3D reconstruction before grasping and manipulating it with a robot hand. The proposed method built over a visual servo loop uses simple features to constantly gaze at the object and handles a three-step exploration with the same control structure. Images captured with the experimental setup are processed online to recover a model with millimetric accuracy.**

I. INTRODUCTION

Grasping everyday objects is one of the complex tasks that robots should perform autonomously when operating in human environments. For this purpose, anthropomorphic hands are used to provide in-hand manipulation possibilities. Such hands are not object-specific tools as grippers are in factory, and so require the object model to plan grasping and manipulations. Some knowledge of the scene is then needed. Analyzing the environment in the specific context of assistance to people is more challenging since several constraints on the workspace are added. Indeed, installing cameras in a large domestic environment is not as easy as it would be in a limited industrial workplace and it may reduce people's intimacy. Therefore, the robot system should only use a few embedded sensors when operating in human places. We choose to equip our robot with an eye-in-hand camera. This placement satisfies the previous constraint and allows enhanced observation capabilities.

In the field of object manipulation, grasping is sometimes considered without a 3D model. In such a case, objects and their grasping points are recognized through demonstration or previous learning in image space directly [1], [2] but no fine in-hand manipulation can be performed without a precise 3D model. A model of the shape can be constructed through several methods. Some works consider model fitting with 2D or 3D data provided by camera or laser scanner [3], [4], while others recover a model from exploration and image data using contour fitting [5], [6] or back-projection probability minimization [7]. A third method consists in reconstructing a more precise model with techniques issued from the computer vision community as reported in [8], [9].

The research leading to these results has been partially supported by the HANDLE project, which has received funding from the European Community's Seventh Framework Program (FP7/2007-2013) under grant agreement. ICT 231640

We have also previously proposed a 3D multi-view reconstruction technique [10] to recover a precise model of unknown objects. It can be decomposed in two major steps. A first step for the volumetric reconstruction is based on a progressive estimation of the visual hull made online during an observation movement. A second step taking advantage of the graph-cuts [11] refines the coarse reconstructed model.

To get the data required for model reconstruction, information on the object and its environment should be retrieved through observation or scene exploration. With the chosen robot system using a monocular image as the only input data, a multi-task visual servoing system has to be developed to fulfill additional objectives while permanently gazing at the object. These objectives are defined as follows:

1) Determining the coordinates $\boldsymbol{O}(x_o, y_o, z_o)$ of the object center w.r.t. the robot base frame $\mathcal{R}_b$ and estimating its bounding-box size: both will serve to restrict the reconstruction processing zone to a small volume around the object and thus improve the shape recovering speed.
2) Reaching multiple observation stations around the object: the main source of data for the 3D modeling part will come from these captures of the object surface.

In order to feed our reconstruction process, we propose in this paper a well suited approach that estimates the initial size and relative position of the bounding-box enclosing the object and that generates a movement around it. The experimental setup consists of a six degrees-of-freedom (DoF) serial manipulator with an eye-in-hand camera mounted on its end-effector (see Fig. 1). In this work, a standard visual servoing method centering the object in the camera field-of-view (FoV) is adapted to enable direct change of the sensor orientation. The eye-in-hand camera reaches various viewpoints without specific feature constraints or complex trajectory generation, while still focusing on the object.

The remaining of the paper is organized as follows. Section II analyzes previous related works on scene exploration, section III describes the visual control loop, the image-based features and the control inputs. Section IV explains how the observation is performed, and the reconstruction process in combination with the exploration is proposed in section V. Experiments and their results for viewpoint generation and for object modeling are presented in section VI. Section VII concludes and proposes future work research.

Fig. 1. Six degree-of-freedom manipulator with an eye-in-hand camera

II. Related works

Using scene exploration with embedded sensors to build a 3D model of unknown objects is one solution proposed in several previous works. As a precursor in 1995, Marchand et al. [5] developed a structure-from-motion method for object modeling, improved by a visual servoing scheme to strategically observe the unknown scene. Within this scheme, the selected features are segments that should remain vertical while moving in the horizontal direction to offer better reconstruction. Unfortunately, with this solution, the exploration movement only works for primitives shapes. In [7], a rough approximation of the object location is first computed by triangulation with the two first views and is then refined during the movement. The authors propose an ellipsoidal approximation of the object shape based on probability distribution in multiple views. Their robot performs a visual servoed exploration motion around the object by computing the next best view to improve the model estimation. The same kind of primitives are recovered in [6] as they fit 2D object contours detected in several views in a 3D quadric model. Active vision helps them optimize the quadric parameters by moving the embedded sensor around the object accordingly with a Simplex method. However, even if their exploration strategies work well, these solutions can only recover primitive shapes. Such simplified models can be sufficient for grasping with a gripper but not for in-hand manipulation with an anthropomorphic hand. The latter requires a precise object envelope to position the fingers while maintaining force-closure [12].

In [8], the authors use a shape-from-silhouette technique to recover a full object model by means of embedded vision improved by a laser stripe. In order to find the object location, no baseline stereo is used but the single camera is initially oriented downward and the first picture helps find the (x_o, y_o) coordinates of the object in some restricted cases. Instead of optimizing the camera motion, they rather execute simple observation movements along a top circular trajectory. An omnidirectional camera is installed in the gripper in [9] and a similar small circular movement is executed with the gripper itself on top of the object whose position is known. A coarse 3D model is built from a visual volume reconstruction technique adapted for omnidirectional sensors. For both approaches, the circular top trajectories do not fully cover the object surface to retrieve all details of its shape.

Even if Berry et al. [13] do not build a model in the end, their observation technique is interesting. The concept of task functions is used to smoothly combine two tasks to turn around an unmodeled object while gazing at it. A primary task handles the visual servoing part to keep the object centered in the view at a constant distance from the camera using three DoF (R_x, R_y, V_z) while a second task controls the remaining translational motion of the vision system (V_x, V_y) to follow a defined trajectory. Such a choice makes it more complicated to generate any motion around the object since the 3D trajectory must be given in the image plane. To comply with this multi-task approach, a specific feature must defined according to the motion direction, which is not straight forward.

In the proposed automatic observation method, we choose to reach selected viewing angles around the object rather than to generate a precise trajectory. For this purpose, a permanent visual control structure is defined that handles two modes: a multi-task mode used to find the object position and to observe it from several viewpoints, and a single-task mode to estimate the object bounding box by only acting on the set-points. Our visual servoing loop will not only control the camera pose to permanently keep the object centered in its FoV and but also ensure that the object silhouette occupies a correct ratio in the picture. This will provide a good image capture, totally enclosing the unknown target in the views.

III. Visual Servoing Structure

This section describes the servoing structure (Fig. 2), which is an adaptation of a standard one avoiding invariant feature selection and still providing multi-task possibilities.

A. General Principle

The model of the object is the missing element we want to reconstruct, which implies that the visual servo control must necessarily be image-based. The general structure of an image-based visual servoing loop is presented in [14]. It includes a block that extracts m features $\boldsymbol{s}$ to give visual feedback and a control law block that provides n control inputs $\boldsymbol{X}$ in task-space. In visual control systems, a redundancy occurs if $m < n$. It is possible to take advantage of such a redundancy to perform other tasks.
In literature, there exist two major approaches to build a multi-task control law :

- The hybrid task function approach [15] used in [13]
- The configuration control approach by Seraji [16]

Both approaches are complex for the considered application. We rather propose to use a simpler control structure (Fig. 2) based on the selection of the control inputs, so that the camera viewpoints can be modified while controlling the sensor position to keep the object in the FoV with three visual features ($\boldsymbol{s}^* = (0, 0, 0.25)$). Among six components of the camera control input vector $\bar{\boldsymbol{X}}_c = [\boldsymbol{V}, \boldsymbol{\omega}]^\top$, three are chosen to achieve the primary visual control ($\bar{\boldsymbol{X}}_{vs}$) whereas three

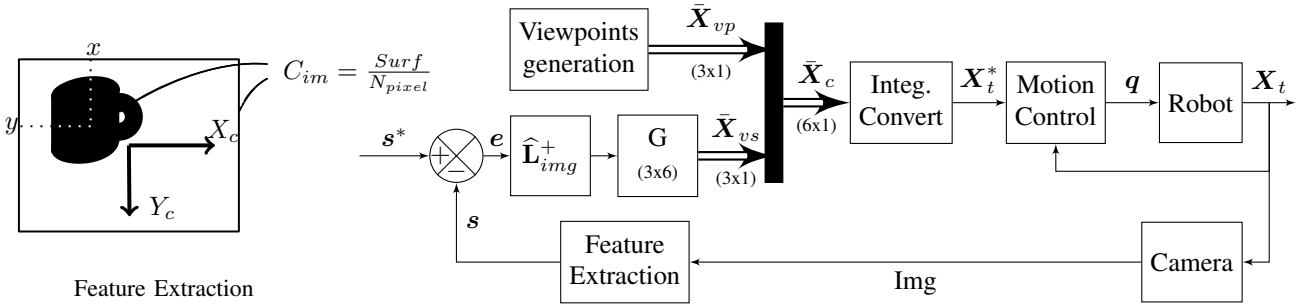

Fig. 2. The visual servoing control loop structure: A primary loop handles camera translations whereas the viewpoints generation changes its orientation. Our robot is internally position-controlled, in task-space directly. The gain matrix G also serves to the selection of the control inputs

other inputs ($\bar{\boldsymbol{X}}_{vp}$) are used to generate arm motions. These remaining control inputs can be viewed as a disturbance input vector that will be rejected if the gains of the servo loop are correctly tuned. The desired tool configuration $\boldsymbol{X}_t^*$ is obtained after a camera to tool transformation and a specific integration block. The system output is internally regulated in task-space with regard to $\boldsymbol{X}_t^*$.

B. Visual Features

For the primary visual control loop, three features must be defined and extracted from the image data. They only rely on the object silhouette in our method. Keeping the object in the FoV requires a minimum of two features to center the object silhouette in the view. The normalized coordinates (x, y) of the center point of the object silhouette will serve this purpose. They correspond to the barycenter coordinates of the silhouette pixels extracted from a binary mask. The mask itself is obtained by some robust and fast processing not detailed here but based on a color region segmentation by graph-cut [11], completed by standard geometry regularization. Initially, the object is supposed to be at least partially visible in the first view and selected by the user by clicking on it. The quality of the features depends on the segmentation but does not really influence the objective of this task since no precise centering is needed.

Additionally, we need another feature to regulate the object occupancy ratio C_{im} in the image. We found that the silhouette surface of standard graspable objects does not vary much when seen from various points of view at a certain distance. Therefore, this third feature will also indirectly play the role of controlling the relative distance between the camera and the object up to a maximum limit depending on the robot workspace. We then obtain a feature vector $\boldsymbol{s}$ as:

$$\boldsymbol{s} = [x; y; C_{im}]^\top$$

The interaction matrix (1) generated by those features and linking $\dot{\boldsymbol{s}}$ (3×1) to $\bar{\boldsymbol{X}}_c$ (6×1), consists of two parts. Its two first lines are computed from the variation $(\dot{x}, \dot{y})$ of the silhouette center coordinates relatively to the camera velocity. The last line depends on the surface that theoretically varies with the square of the object distance Z on the camera depth axis $\vec{Z}_c$. The distance Z is initially estimated to the height of the camera $z_c \,|_{\mathcal{R}_b}$ and is set as the distance from camera to object once its position is known. We approximate the parameter to make it depend on a coefficient K_{surf} and on Z^{-2}. We also limit its influence to the camera depth velocity (V_z). This way, we ignore the effect of the shape variation seen from different view angles.

$$\mathbf{L}_{img} = \begin{pmatrix} -\frac{1}{Z} & 0 & \frac{x}{Z} & xy & -1-x^2 & y \\ 0 & -\frac{1}{Z} & \frac{y}{Z} & 1+y^2 & -xy & -x \\ 0 & 0 & \frac{K_{surf}}{Z^2} & 0 & 0 & 0 \end{pmatrix} \tag{1}$$

In the end, we use $\widehat{\mathbf{L}}_{img}^+$, the pseudo-inverse of the estimated interaction matrix.

C. Control inputs

In this part, the control inputs for each task are described. The three features defined previously can be controlled efficiently with three control inputs (three DoF). Within our structure, we have several selection options, depending on the control inputs used for the visual control task.

We choose the camera translations (V_x, V_y, V_z) as control inputs for the visual servoing (2): two translations for centering operations and one for the silhouette occupancy ratio C_{im}, which induces motion along the optical axis. This choice is not usual since centering is generally done using camera orientations (pan, tilt), but we found that translations are able to fulfill the task perfectly, if the control law is tuned well.

$$\bar{\boldsymbol{X}}_{vs} = [V_x; V_y; V_z]^\top \tag{2}$$

The remaining control inputs $(\omega_x, \omega_y, \omega_z)$ are then directly devoted to viewpoint reaching (3). Indeed, two rotation variables α and β defined as the viewing angles w.r.t. the robot arm base reference are equivalent to spherical coordinates (elevation θ and azimuth φ). These angles are sufficient to reach any 3D position on a sphere as long as the radius is defined (see Fig. 3). Since the camera depth is visually controlled, the radius is implicit. So, using camera rotation velocities $\boldsymbol{\omega}$ is an efficient way to obtain desired camera viewpoints α, β. The camera roll angle γ around the optical axis remains constant and is equal to zero.

$$\bar{\boldsymbol{X}}_{vp} = [\omega_x; \omega_y; \omega_z]^\top \tag{3}$$

The combined visual servoing $\bar{\boldsymbol{X}}_{vs}$ and viewpoint $\bar{\boldsymbol{X}}_{vp}$ input vectors finally control the six DoF with $\bar{\boldsymbol{X}}_c$.

IV. Automatic observation

As several information need to be retrieved from the environment, we propose three steps to fulfill the automatic observation. Two steps A and C use the multi-task mode whereas step B only uses the single task mode.

A. Pose estimation

The first information to be determined from the environment is the object location $\boldsymbol{O}(x_o, y_o, z_o)$ w.r.t. the robot base frame $\mathcal{R}_b$. Orientation is not relevant as the object shape is unknown. This relative position is sufficient to correctly grasp the object from a robot point of view, without any assumption on the absolute location as long as it is in a reachable range. We choose a similar technique as proposed in [8], with no initial orientation however, as it works from any viewpoint at any distance, with the object at least partially in the FoV. We act on viewing angles to reach the downward orientation, which indirectly move the arm above the object.

With our proposed servoing loop structure, both camera viewing angles (α, β) can evolve easily while maintaining the object centered (see Fig. 3). As a response to this task fixing $\alpha = \pi$ and $\beta = 0$, the arm automatically reaches the top viewing station where $\vec{Z}_c$ and $\vec{Z}_o$ are coincident. Then, (x_c, y_c) coordinates of the camera w.r.t. $\mathcal{R}_b$ are the same as the object center coordinates $(x_o, y_o)\,|_{\mathcal{R}_b}$.

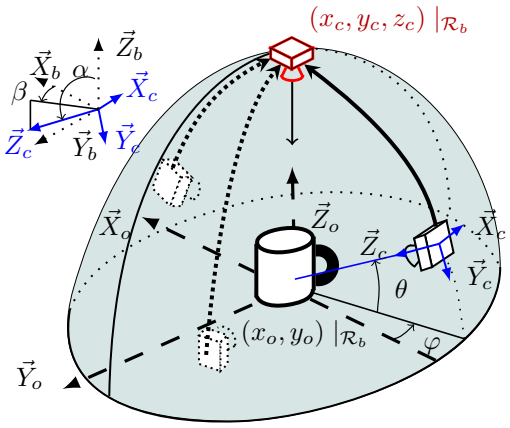

Fig. 3. Pose estimation: For $\alpha = \pi$ and $\beta = 0$, the camera reaches a top view of the object while servoed to keep it centered. (x_c, y_c) are always the same as the object center (x_o, y_o) at the end of the movement.

B. Bounding box estimation

Once the camera centered over the object, looking downward, coordinate $z_o\,|_{\mathcal{R}_b}$ of the object bounding-box (b.box) and its dimensions (D_x, D_y, D_z) can be estimated using the main control loop by only changing the set-points. In this mode, the viewpoint generation block provides null orientation velocities. The b.box dimension D_z is arbitrarily fixed but its position $z_o\,|_{\mathcal{R}_b}$ will be set to the estimated distance z_{sil} between the silhouette plane and the robot base plane (Fig. 4). z_{sil} and dimensions D_x, D_y will be obtained in a similar fashion, directly from the following technique.

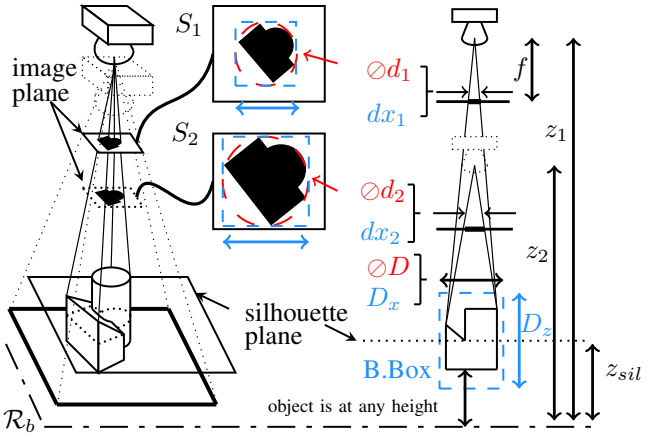

Fig. 4. Bounding-box estimation: The camera is servoed along the optical axis for two different surface ratios C_1, C_2. With two consecutive measurements of z_c, an approximate object position $z_o = z_{sil}$ w.r.t. $\mathcal{R}_b$ can be determined as well as D_x and D_y.

Consider a particular case when the object is a cylinder with diameter D and height D_z. Diameter D is measured in the silhouette plane and is projected as d_k in the image plane. d_k is related to z_{sil} and to the height z_c of the camera. Two different set-points C_1 and C_2 for the occupancy feature are successively set. Since the resulting motion occurs on the optical axis oriented downward, we get two camera distance measurements z_1 and z_2 w.r.t. the robot base. d_1, d_2 are the diameters of the object silhouettes S_1 and S_2 in the image plane. Eq. 4 is used to extract z_{sil}:

$$D = \frac{d_1}{f}(z_1 - z_{sil}) = \frac{d_2}{f}(z_2 - z_{sil}) \quad (4)$$
$$\Leftrightarrow z_{sil} = \frac{d_2 z_2 - d_1 z_1}{d_2 - d_1}$$

In the case of the cylinder, d can be written as an exact function of the silhouette surface S; $d = \mu\sqrt{S}$. This relation is also true for a box with dimensions (D_x, D_y, D_z) and is a rough approximation for any other forms. Eq. 5 can first be derived according to the surface ratio $C = \frac{S}{S_{im}}$ with S_{im} the total image surface and then simplified using the known variation between the desired surface ratios $n = \frac{C_2}{C_1}$,

$$z_o = z_{sil} = \frac{z_2\sqrt{C_2} - z_1\sqrt{C_1}}{\sqrt{C_2} - \sqrt{C_1}} = \frac{z_2\sqrt{n} - z_1}{\sqrt{n} - 1} \quad (5)$$

Although objects may have various forms, if the D_z is large enough around the estimated position z_o, the b.box encloses the object and still reduces the working zone without any knowledge on its absolute z location. The same way, with the previous measurements, completed by the known camera focal and the size dx_1, dx_2 (resp. dy_1, dy_2) of the bounding square including the object silhouette in the image, we get an approximate D_x (resp. D_y) size of the 3D b.box as:

$$D_x = \frac{z_2 - z_1}{f(\frac{1}{dx_2} - \frac{1}{dx_1})} \quad (6)$$

C. Turning around the object

For this step, the viewpoint generation block (Fig. 2) provides camera orientation velocities. Still starting from

the top view, it generates slow variations on $\boldsymbol{\omega}$ to reach different viewing stations defined by (α, β) viewing angles. No 3D position relative to the object center is needed as the correct camera position and orientation come from the combined action of the visual servoing loop and this viewing angles generation. To cover a majority of the object surface, and have a large amplitude of movement, angles have been determined experimentally. A sufficient coverage can be done with 10 viewpoints only since adding more views does not improve the model significantly [10]. As the distance to the object varies in a short range, the camera appears to move on a pseudo-half-hemisphere centered on the object (Fig. 5). The path passing through the viewing stations goes along a descending arc and a surrounding arc. The arc radius is not fixed but is induced by the third visual feature regulation.

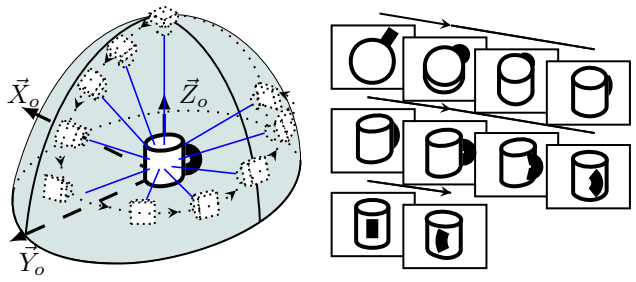

Fig. 5. Turning around: Starting from the top position, the apparent motion goes along a descending arc and a surrounding arc whose radius are induced by the surface ratio regulation. 10 images are captured on the fly.

V. 3D Reconstruction

To finally build an object model online, the reconstruction process described in [10] is combined with the proposed exploration technique in three steps.

A. Pre-processing

During the observation movement, at each observation station, a picture is taken on the fly (Fig. 5) and processed to extract a more precise silhouette than the one used for visual features. We adapted a growcut technique [17] for precisely separating the object from the background thanks to known seeds. Indeed, since the object is known to be centered by visual servoing for this part of the exploration, seeds defined in an area at the center of the image belong to the object whereas the ones at the border belong to the background.

The picture position and orientation are retrieved from the robot end-effector position according to the camera to tool frame transformation. This transformation is obtained off-line through eye-hand calibration [18] and is done once. Precise tool measurements and camera versus robot synchronization are the key points for correct extrinsic parameters computing. The first two parts of the observation movement finally provide a bounding-volume and the calibrated silhouettes needed for the reconstruction process.

B. Visual Volume reconstruction

In [10], we proposed a simple volumetric reconstruction of the object visual hull similar to [19]. It consists in progressively carving the initial bounding volume divided in sufficiently small elements called voxels. Each voxel is tested for consistency to the visual hull by checking if its projection lies in the silhouettes. We have added a progressive test to reconstruct this envelop during the observation movement. Indeed, as soon as the first data are introduced in the pipeline, the consistency test is started for the first silhouette. The resulting voxel set is stored in a linked-list that will be used as the new volume to be checked for the next image, progressively removing inconsistent voxels from the list during the robot movement. This solution results in a faster reconstruction taking advantage of the time running between captures at the viewing stations.

C. Refining the model

At the end of the previous step, the observation movement combined with the visual volume extraction process gives a first coarse model of the object, more precise than previous solutions based on primitives, which can already serve for approach strategy planning. In the same time, the model is refined through an optimization process of colorimetric data. We use the following principle: Voxels that belong to the true object surface are photo-consistent in each view where they are visible. A photo-consistency is computed for each voxel in the coarse envelope and is embedded in a graph. A graph-cuts process [11] is applied to reject inconsistent voxels, most of which are in the object concavities. The refined model can be used for fine manipulation.

VI. Experiments & Results

A. Experiments

We tested our visual control loop and reconstruction pipeline first in a virtual environment using Marilou Robotic Studio that offers simulated physics and embedded vision, and then in a real world setup on our industrial Adept robot equipped with a VGA Sony camera. The camera intrinsic parameters and tool relative transformation are determined by the standard calibration process provided by the OpenCV library. The camera is synchronized with the robot measurements by software triggering. A frame can be captured every 32 ms at the same instant the robot end-effector transformation is measured. Note that the height of the object support seen in Fig. 1 is not known and is not required anyway.

B. Results

First, some observation motions were performed for different kinds of objects and positions. As long as the object is in a reachable range, the automatic exploration worked well for any starting position where the object is seen at least partially. The controller gain matrix for the visual loop has been manually tuned once for the first experiments so as to reject the disturbance introduced by the viewpoint change. Measurements on the real robot for a centering movement during the pose estimation task is presented in Fig. 6. Object locating task is executed smoothly and the x_o, y_o position is retrieved at the end of the movement. The motion around the

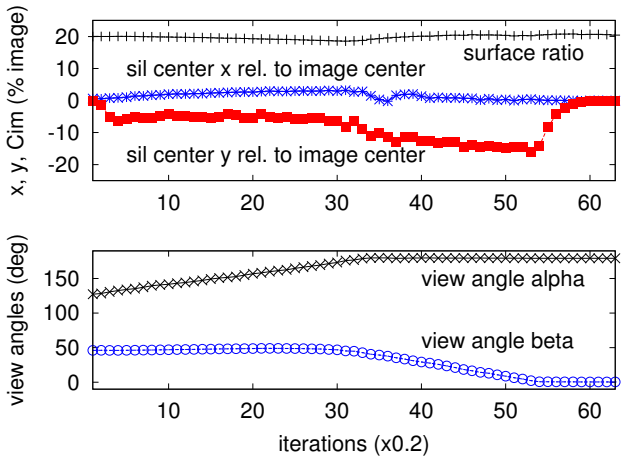

Fig. 6. Feature x, y, C_{im}, and view angles α, β during a pose estimation motion. C_{im} shows an insignificant variation during the movement to the top view, while (x, y) have small drag errors

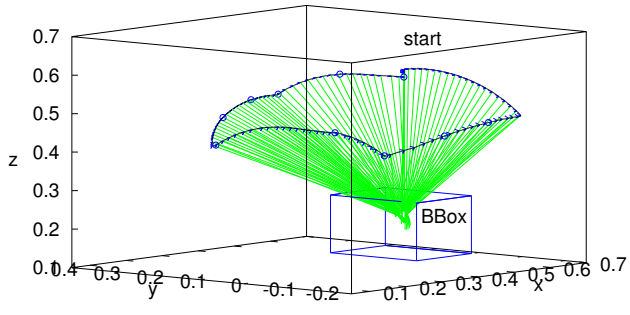

Fig. 7. Trajectory of the camera around the object for 10 viewpoints.

object is shown in 3D on Fig. 7. The 10 stations are reached progressively by changing the viewpoints.

Real objects have been reconstructed with the calibrated pictures, captured at these viewpoints (Fig. 8). The visual volume is computed in a maximum of 5 seconds, so the robot movement is executed over a slightly larger time range. Presently, additional 10 seconds are required for the refinement step. In the end, the model accuracy is millimetric.

Fig. 8. 8 of the 10 silhouettes, one camera view and a rendering of the reconstructed model for real objects.

VII. Conclusion & Future Work

In this paper, we have proposed a method for automatic observation of unknown objects to feed a modeling process in the case of grasping and manipulation applications. Input images are captured from various viewpoints around the object thanks to a well suited visual control structure that keeps the object in the field of view of an eye-in-hand camera while moving the arm. The relative position and size of a bounding-box enclosing the object are automatically found. An online multi-view reconstruction method combined with the presented scene exploration provides a model of the object with millimetric accuracy.

As a future work, we intend to convert our algorithms to parallel computing on GPU and to execute pre-processing of colorimetric or visibility data during the movement to improve the speed of the modeling steps. A complementary exploration depending on reconstruction quality measurements is our next goal to recover more details. The best viewpoints will then be computed according to less visible zones in the model.

References

[1] M. Hüser, T. Baier-Löwenstein, and J. Zhang, "Learning of demonstrated grasping skills by stereoscopic tracking of human head configuration," in *ICRA'06*, 2006, pp. 2795–2800.

[2] A. Saxena, J. Driemeyer, and A. Y. Ng, "Robotic grasping of novel objects using vision," in *IJRR*, 2008.

[3] G. Taylor and L. Kleeman, "Grasping unknown objects with a humanoid robot," in *Proc of 2002 Australasian conference on Robotics and Automation*, 2003, pp. 191–196.

[4] J. Kuehnle, Z. Xue, M. Stotz, J. Zoellner, A. Verl, and R. Dillmann, "Grasping in depth maps of time-of-flight cameras," in *International Workshop on Robotic and Sensors Environments*, 2008, pp. 132–137.

[5] E. Marchand, F. Chaumette, and E. Rutten, "Real time actual visual reconstruction using the synchronous paradigm," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'95*, 1995, pp. 96–102.

[6] C. Dune, A. Remazeilles, E. Marchand, and C. Leroux, "Vision-based grasping of unknown objects to improve disabled people autonomy," in *Robotics: Science and Systems Manipulation Workshop: Intelligence in Human Environments*, Zurich, Switzerland, June 2008.

[7] G. Flandin and F. Chaumette, "Autonomous visual exploration of complex objects," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'01*, vol. 3, 2001, pp. 1533–1539.

[8] G. M. Bone, A. Lambert, and M. Edwards, "Automated modeling and robotic grasping of unknown three-dimensional objects," in *International Conference on Robotics and Automation (ICRA)*, 2008.

[9] T. Yoshikawa, M. Koeda, and H. Fujimoto, "Shape recognition and grasping by robotic hands with soft fingers and omnidirectional camera," in *ICRA'08*, 2008, pp. 299–304.

[10] G. Walck and M. Drouin, "Progressive 3d reconstruction of unknown objects using one eye-in-hand camera," in *IEEE Intl. Conf. on Robotics and Biomimetics, ROBIO'09*, 2009, pp. 971–976.

[11] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.

[12] J.-P. Saut, A. Sahbani, S. E. Khoury, and V. Perdereau, "Dexterous manipulation planning using probabilistic roadmaps in continuous grasp subspaces," in *IROS'07*, 2007, pp. 2907–2912.

[13] F. Berry, P. Martinet, J. Gallice, U. Blaise, and P. C. Ferrand, "Real time visual servoing around a complex object," in *IEICE Transactions on Information and Systems*, 2000, pp. 1358–1368.

[14] F. Chaumette and S. Hutchinson, "Visual servo control, part i: Basic approaches," *IEEE Robotics and Automation Magazine*, vol. 13, pp. 82–90, 2006.

[15] C. Samson, M. Le Borgne, and B. Espiau, *Robot Control. The Task Function Approach*, ser. 22. Oxford: Oxford engineering science series, 22, Clarendon Press, 1991.

[16] H. Seraji and R. Colbaugh, "Singularity-robustness and task-prioritization in configuration control of redundant robots," vol. 6, Dec 1990, pp. 3089–3095.

[17] V. Vezhnevets and V. Konouchine, ""grow-cut" - interactive multi-label n-d image segmentation," in *GraphiCon*, 2005, pp. 150–156.

[18] K. H. Strobl and G. Hirzinger, "Optimal hand-eye calibration," in *IROS'06, Beijing, China*, 2006, pp. 4647–4653.

[19] A. Laurentini, "The visual hull concept for silhouette-based image understanding," *PAMI*, vol. 16, no. 2, pp. 150–162, February 1994.