2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Realtime Tracking and Grasping of a Moving Object from Range Video

Farzad Husain, Adrià Colomé, Babette Dellen, Guillem Alenyà and Carme Torras

Abstract— **In this paper we present an automated system that is able to track and grasp a moving object within the workspace of a manipulator using range images acquired with a Microsoft Kinect sensor. Realtime tracking is achieved by a geometric particle filter on the affine group. Based on the tracked output, the pose of a 7-DoF WAM robotic arm is continuously updated using dynamic motor primitives until a distance measure between the tracked object and the gripper mounted on the arm is below a threshold. Then, it closes its three fingers and grasps the object. The tracker works in real-time and is robust to noise and partial occlusions. Using only the depth data makes our tracker independent of texture which is one of the key design goals in our approach. An experimental evaluation is provided along with a comparison of the proposed tracker with state-of-the-art approaches, including the OpenNI-tracker. The developed system is integrated with ROS and made available as part of IRI's ROS stack.**

I. INTRODUCTION

In the field of robotics, many applications have been tailored towards servoing using visual information. The goal is to use information obtained from vision inside a servo loop to control a mobile manipulator. Visual servoing is broadly classified into two categories, i.e., image-based and position-based [1]. Most of the servoing tasks require the target to be stationary. Camera configurations such as eye-in-hand [2] and eye-to-hand [3] have been used. Tracking is often performed on the basis of color/grayscale images [4], [5], [6]. However, if the objects to be tracked are only weakly textured and do not contain distinctive color features, tracking may fail.

Recent advancements in range sensing technology for indoor scenes have lead to the development of a multitude of practical vision applications, but only little work has been done with respect to visual-servoing solutions based on range images. In this paper, we present a novel approach for eye-to-hand, moving target, position-based servoing using depth as the only visual cue. This has the immediate advantage that the performance of the tracker is independent of the appearance of the objects in terms of color, and may thus generalize better to different scenarios. Tracking is achieved by a geometric particle filter on the affine group [7]. The respective, estimated affine transformation is applied to a bounding box placed in the range image, and a rigid transform is used to compute the measurement function. In this sense, tracking is entirely data driven, and no 3D object model needs to be used, neither for tracking nor for grasping. This also reduces the computational cost of the method. For the experiments, we use a Microsoft Kinect sensor callibrated to a Barret WAM arm with 7-DoF as shown in Fig. 1. The output from the tracker is efficiently coupled with the pose of the end-effector. A smooth trajectory is created online from the pose of the object which controls the joint angles of the WAM arm, by using a robust inverse kinematics algorithm, as in [8].

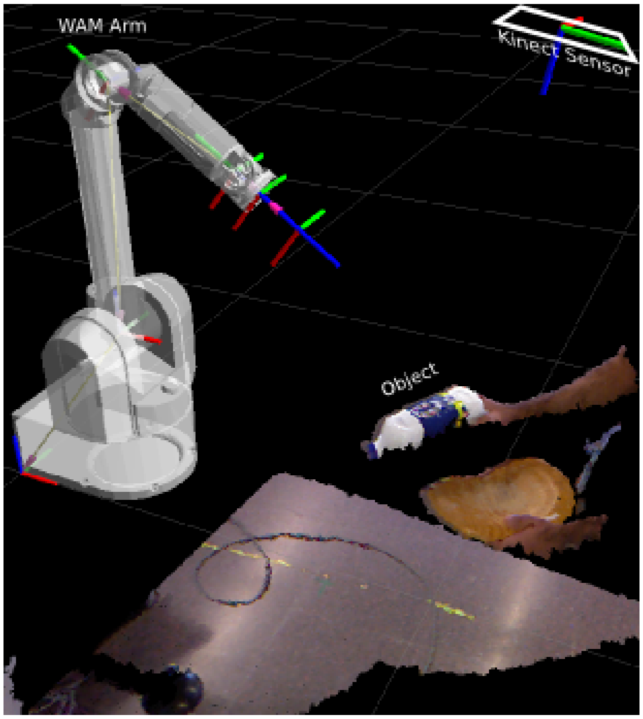

Fig. 1. Illustration of the experimental setup containing the Barret WAM arm, the Kinect sensor mounted above the scene, and the 3D reconstruction of the scene, including the target object.

Potential applications include automatically picking up objects from a moving conveyor belt, incremental learning from demonstration, and human-robot interaction [9].

The paper is organized as follows. In Section II, we provide a brief description of the existing work on tracking and grasping of moving objects. The problem formulation is provided in Section III. In Section IV, we present a novel method for tracking with range images. Section V describes how the tracked information is used for dynamic repositioning of the end-effector. Results of our approach are shown in Section VI-C, and discussed in Section VII.

II. RELATED WORK

In the past, many different approaches have been developed for tracking and grasping of moving objects [10],

The authors are with the Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Llorens i Artigas 4-6, 08028, Barcelona, Spain, (e-mail: {shusain, acolome, bdellen, galenya, torras}@iri.upc.edu).

978-1-4799-3685-4/14/$31.00 ©2014 IEEE

[11], [12], [13], [14], [15], [16], [17], [18], [19], [20]. Most of these methods use stereo images to compute the 3D pose for grasping. Matching scores required for tracking are usually computed from the color/grayscale features of the object. For example, in [10], the 3D pose of a moving object is computed using stereoscopic optic flow and used to control the motion of the robotic arm until interception and grasping is performed. The motion model employed to control the positioning of the robotic arm is restricted to planar trajectories, hence limiting the approach to grasping objects moving on a planar surface only. Similar constraints for motion in 2D have been employed in [11]. In [12], objects move on a conveyor belt in a straight line, and the precise trajectory of the objects could thus be provided.

Grasping a moving object using an eye-in-hand configuration has been performed in [14]. However, the method presented therein is restricted to a single class of objects. Also the motion of the object is restricted to translations. A simulation of tracking and grasping a moving object based on a CAD model along with the dynamic selection of feature points has been presented in [15]. Based on the tracked results a real-time motion planning method is proposed. Simulation results have also been provided in [16] for an integrated sensing and actuation system for grasping a moving object.

A position-based and an image-based scheme for tracking and grasping problem have been proposed and compared using simulations in [17]. However, here it is assumed that the pose of the object is already known.

Our setup closely resembles the one used in [19], but the tracking algorithm used in that work is rather simple. A blob detection algorithm based on the target image color is employed which can easily fail under varying appearance or lighting conditions.

Another possible approach to tracking using depth is to directly determine the 3D rigid transform that the tracked surface undergoes. This approach has been adopted in the OpenNI tracker for the Kinect sensor in the Point Cloud Library [1], where the 6 parameters of the rigid transform are predicted using a particle filter. Different to our approach, no motion model is employed, resulting in less efficient prediction of future states.

III. PROBLEM STATEMENT

Given an object moving in 3D space, we would like to reposition the end-effector of the manipulator continuously, until the distance between the moving object and the manipulator is smaller than a threshold. We assume that the object remains inside the workspace of the manipulator and it moves slow enough to be tracked and approached by the manipulator.

IV. OBJECT SURFACE TRACKING

We track the object surface in the 2D image plane using a bounding box supplied by the user in the initial image, enclosing the pixel coordinates of the tracked surface template in the image plane. To determine the pose of the surface at each time instant t, we apply a geometric particle filter on the affine group, yielding an estimate of the 2D affine transform of the tracked surface in the image plane with respect to the initial coordinates, as proposed in [7]. The estimator uses a constant velocity model for the state dynamics i.e., the state update equation is expressed as

$$X_t = X_{t-1} e^{[a \ \log(X_{t-2}^{-1} X_{t-1}) + W_t]}, \tag{1}$$

where X is the 3×3, 2D affine transformation matrix, a is the autoregressive process parameter, and W is the Wiener process noise with covariance $Q \in \Re^{6\times 6}$, owing to the 6 free parameters of the 2D affine group. The measurement equation is expressed as

$$y_t = h[X_t, I_{t=0}(P)] + v_t, \tag{2}$$

where h is the measurement function, P is the set of points representing the pixel coordinates of the tracked template in the image plane. I is the range image, i.e., for each $p \in P$, $I(p)$ gives the actual range value $(x, y, z)_p$ in the Euclidean space and v is the the Gaussian noise with covariance $R \in \Re^1$. Unlike [7], we have defined the measurement function as

$$h(X_t, I_{t=0}(P)) = \left\| I_{t=0}(P) - I'_t(P'_t) \right\|_1, \tag{3}$$

where P'_t is the set of pixel coordinates obtained after transforming every $p \in P$ with X_t. Since we are using the range data, we cannot directly compute the difference between $I_{t=0}(P)$ and $I_t(P'_t)$ as it is the case of color/grayscale images [7]. This is because the range data provides the Euclidean distances relative to the camera pose, hence we first determine the rigid transform such that the distance between $I_{t=0}(P)$ and $I_t(P'_t)$ is minimized in a least square sense [21]. $I'_t(P'_t)$ represents this set of points obtained after applying the rigid transformation.

The surface template $I_{t=0}(P)$ is periodically updated every five frames by computing the mean of the 3D shape that has been tracked during this interval. Detailed description for calculating the measurement likelihood $p(y_t|X_t)$ for the importance sampling step as well as the particle resampling step can be found in [7], [22].

The affine transform encodes the deformation of a planar shape moving in 3D space and acquired under an orthogonal projection. In the case of non-planar surfaces with out-of-plane rotations, the affine transformation cannot determine exact point correspondences. However, extensive experimental results with non-planar surfaces have revealed that under weak-perspective assumption such a transformation can be approximated with a two-dimensional affine transform in the image plane. Previously, this assumption has been made in [22], [23], [24], [25] for color/grayscale images.

[1] Available at: `http://pointclouds.org/`

V. POSITIONING WAM END-EFFECTOR

We continuously update the goal position of the WAM end-effector until the generalized Cartesian space error (including position and orientation)

$$e = \left\| \begin{bmatrix} e_p \\ e_0 \end{bmatrix} \right\|_2 + \delta \tag{4}$$

is below a threshold. Here e_p, e_o are the position and orientation errors with respect to the desired pose of the target object, as defined in Section 3.7 of [26], and δ is a small offset defined as half the length of the bounding box.

The position of the target is defined as the centroid of the points in 3D space, enclosed within the bounding box. In order to determine the orientation, we fit a plane to the 3D points, corresponding to three corners of the bounding box and compute its angular displacement relative to the camera axes. Once the error e is less than a threshold, the gripper closes its fingers and grasps the object. Using a robust Inverse Kinematics (IK) algorithm [8], we can convert the generalized position of the object into a joint vector goal that places the robot in the target.

Then, the robot's goal can be updated online by using Dynamic Motor Primitives (DMP) [27] at a joint level, where a desired trajectory of the robot is computed with a second order dynamic system:

$$\dot{\mathbf{z}}/\tau = \alpha_z (\beta_z (\mathbf{G} - \mathbf{q}) - \mathbf{z}) + \mathbf{f}(t), \tag{5}$$

where $\mathbf{q}$ is the joint position, $\mathbf{G}$ the goal position, α_z, β_z are proportional-derivative constants, τ a time constant, $z = \dot{\mathbf{q}}/\tau$ a rescaled velocity, and $\mathbf{f}(t)$ a shaping function of the trajectory.

This characterization gives us a desired acceleration, velocity and position at each time instant. For $\beta_z = \alpha_z/4$, the system is critically damped and these derived signals can be sent to any controller to track the desired trajectory. In the case of object tracking, the shaping function $\mathbf{f}(t)$ can be set to zero to have a pure critically damped attractor to the goal $\mathbf{q} = \mathbf{G}$, or used as an obstacle-avoidance term as in [27].

The DMP representation allows us to change the trajectory goal online, while maintaining the continuity on the position and velocity commands, and without needing to recompute the whole trajectory (as we would have to if using splines). Thus we can update the goal according to the movement of the tracked object using an adequate inverse kinematics algorithm and the default PID controller provided with the arm. A general scheme of the whole tracking system implemented can be seen in Fig 2.

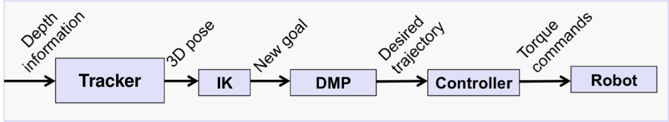

Fig. 2. General scheme of the experiment. The depth information from the Kinect camera is used to obtain a 3D pose and is sent to the IK algorithm, which computes a new goal. This goal is sent to the DMP, which immediately updates the trajectory. Finally, the desired trajectory to the current goal is sent to the controller.

VI. RESULTS

A. Pros and cons of using depth instead of color images

Depth data lacks texture information which could be essential to constrain the affine transform. However, we found that texture can also be a limiting factor during tracking, as can be seen when looking at the example shown in Fig. 3. During manipulation, the texture on the cup changes rapidly in the image because of its changing orientation and self-occlusions, while the surface shape remains approximately the same, allowing our tracker to succeed in cases where others fail. This behavior has been observed in several cases. We also tested texture-based trackers for gray-scale images directly on range images, not surprisingly they failed as well.

Figure 4 shows a quantitative comparison using the root mean square (RMS) error of the centroid of the bounding box with respect to the ground truth at each time instant for the video corresponding to Fig. 3. To generate the closest possible ground truth, we first over-segmented the video using the method proposed in [28], and then manually relabeled the segments that belong to the tracked object.

It can be seen in Fig. 4 that the tracker from [7] and the one from [25] eventually loose track ($\sim$ frame 169 and $\sim$ frame 853, respectively), whereas ours successfully tracks the object until the end.

B. Comparison with the OpenNI tracker

We compare our method with the OpenNI tracker for the Kinect sensor in the Point Cloud Library. The OpenNI tracker directly determines the 3D rigid transform from the points on the tracked surface model and predicts the six rigid transformation parameters using a particle filter.

In order to make a fair comparison, we omitted the color information in the measurement function, yielding

$$h(p,q) = \sum_j \left(1 + |p_j - q_j|^2\right), \tag{6}$$

where j ranges over to all the points in the reference model, p_j is the 3D position of the predicted point, and q_j is the 3D position of the nearest point in the input point cloud.

We conducted several experiments and observed that the OpenNI tracker is more prone to failure due to occlusions than ours, presumably because it does not employ a motion model. Figure 5 shows two of the cases where the OpenNI tracker failed to track the object while ours kept on tracking.

C. Tracking and Grasping of moving objects

Several experiments were conducted with objects of different shapes and appearances, e.g., a milk bottle, a carton box, and a ball. In all experiments, the WAM arm was able to successfully chase the object using the tracking information and eventually grasp it. Selected frames are shown in Fig. 6 and Fig. 7.

Figure 6 shows a bottle that is being manipulated by a human. This scene was recorded with a Kinect camera

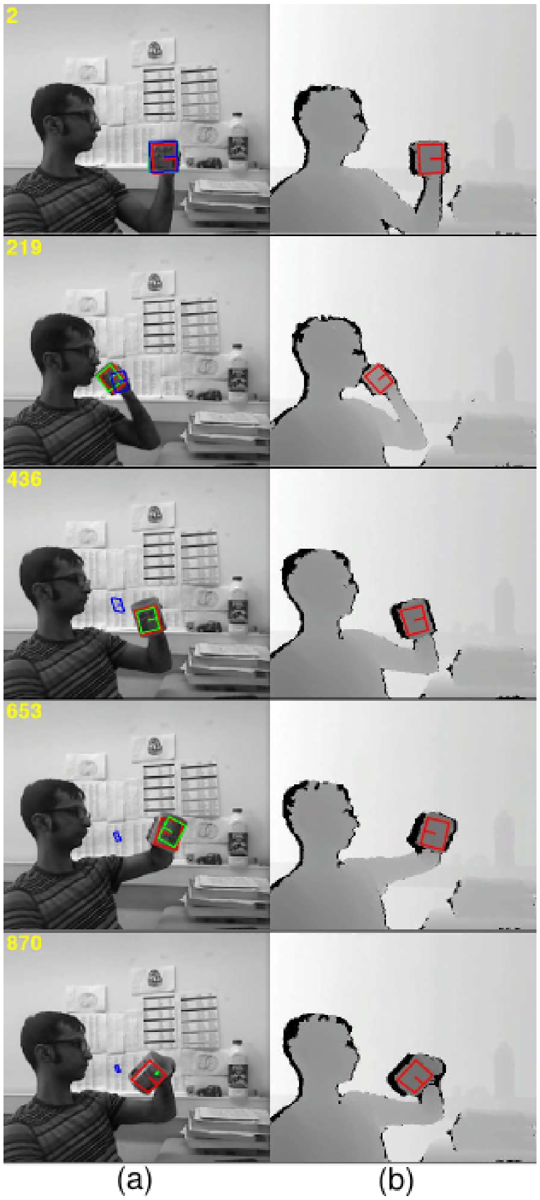

Fig. 3. Comparison of different tracking algorithms for grayscale (a) and range images (b). Our tracker (red rectangle) uses range data only, whereas the tracker from [7] (blue rectangle) and the one from [25] (green rectangle) use grayscale images. More results are provided in the accompanying video.

mounted above (the same as illustrated in Fig. 1). During the experiment, the robot arm follows the motion of the bottle and tries to minimize the distance between its gripper and the bottle (see frames 401-1201). Once the distance is below a threshold, the gripper closes and grasps the object (see frame no. 1401). Similar results were obtained for other objects, see Fig. 7. Figure 8 shows the trajectory of a box and the end-effector of the manipulator during one of the experiments.

D. *Tracking-speed analysis*

The tracker is implemented in C++ and runs at ~ 20 fps on an Intel Xeon quad core processor. We expect further speedup by porting the code to a GPU. The maximum object speed with which our tracker can cope with depends on factors such as the size of the object in the image plane, its shape, and also the way it is being manipulated. In order to get an idea of the speed that our tracker can handle during the experiments, we tracked a cylindrical surface which moved freely in 3D space at a speed that allowed correct tracking of the object (see Fig. 9(a)). Figure 9(b) shows the distance that the object covered within each second. The tracker was able to track objects moving up to 21 cms/sec.

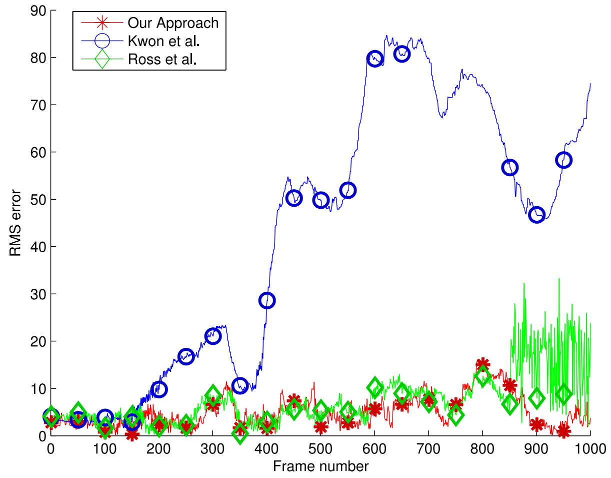

Fig. 4. A comparison of RMS error of the centroid with respect to the ground truth for our tacker (red color), the tracker of [7] (blue color) and the tracker of [25] (green color). More results are provided in the accompanying video.

VII. CONCLUSIONS

This paper presents a complete system which is able to robustly track and grasp moving objects in 3D space. Our code and complete videos are available for download at `http://www.iri.upc.edu/groups/perception/#trackGrasp`.

We demonstrated that reliable realtime tracking can be achieved and used for robotic manipulation of moving objects using depth data alone. For this purpose, we developed a novel method for tracking in depth images based on a geometric particle filter on the affine group. This type of tracking paradigm has been used before in color images [7], [22]. An advantage of our method compared to color-based tracking is that its performance is independent of the appearance of the object in terms of color and texture (see Fig. 3). Compared to the OpenNI tracker, our method showed equal performance in most cases and even outperformed it in some cases (see Fig. 5).

The system has some limitations which we plan to address in the future. If the manipulator moves to a location where it partially occludes the tracked object or if the object undergoes self-occlusions, then the centroid may deviate from the correct position, and the gripper may be moved away from the desired grasping position. This limits the

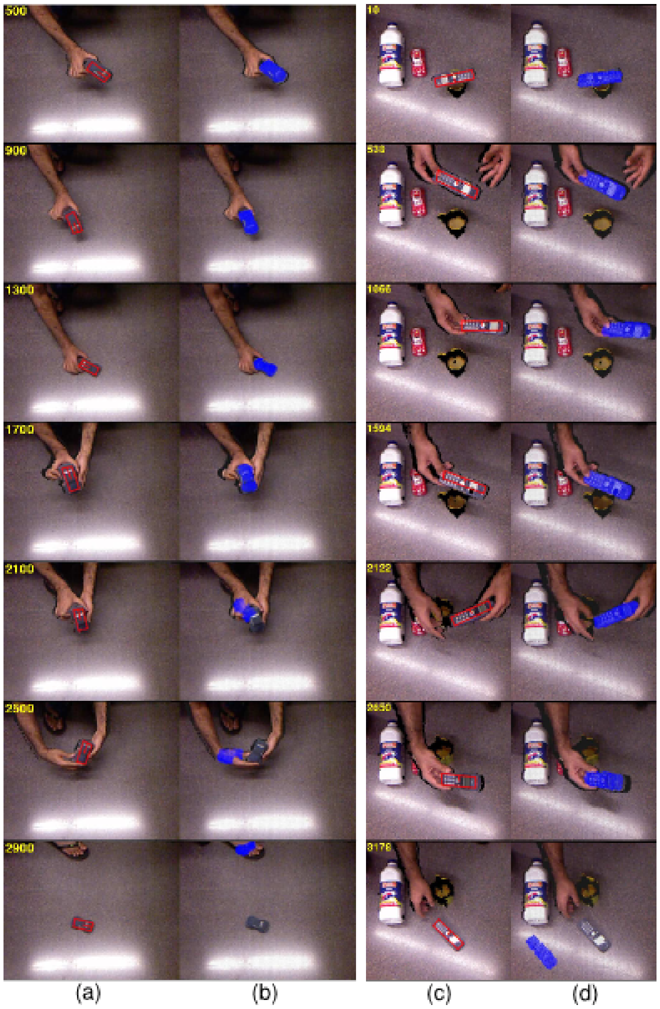

Fig. 5. Comparison of ((a) and (c)) our tracker (red rectangle) with ((b) and (d)) the tracker available in the Point Cloud Library (blue pixels). The color images are shown here for illustration only and not used in the tracking procedure. More results are provided in the accompanying video.

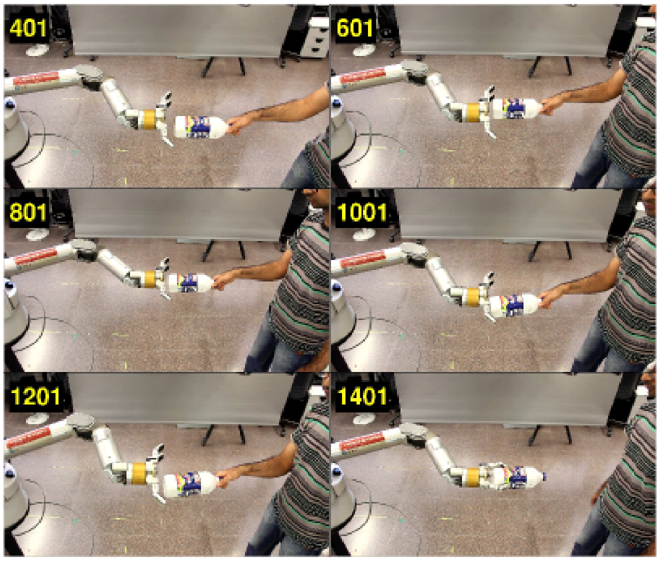

Fig. 6. Tracking and grasping of a moving bottle.

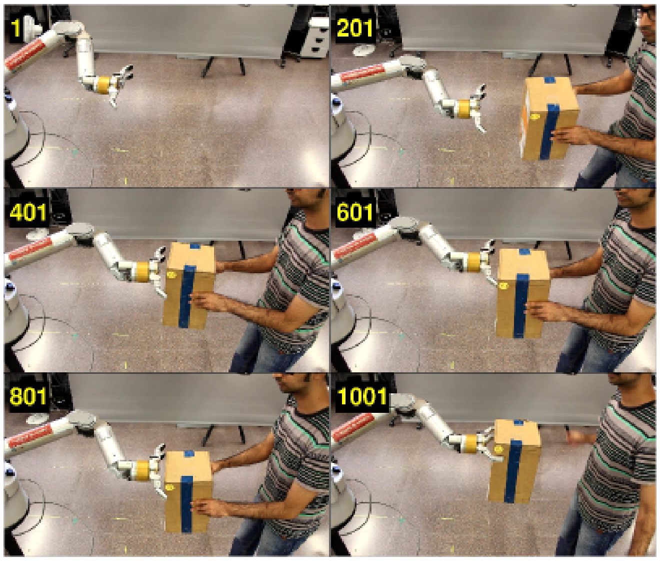

Fig. 7. Tracking and grasping of a moving carton box.

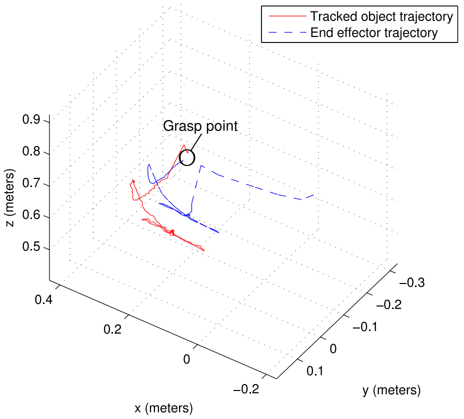

Fig. 8. Object and end-effector trajectories during tracking and grasping of a moving box.

workspace of the manipulator. In the future we plan to use a setup with multiple range sensors to cope with this situation.

In situations where a user supplies a bounding box which does not contain the entire surface to be tracked, our tracker may fail. For instance, we cannot track a small planar patch inside a plane. For the aforementioned reason, our tracker requires that the four corners of the bounding box supplied initially by the user lie on the edges of the surface that is to be tracked.

ACKNOWLEDGEMENTS

This work was supported by the EU project IntellAct FP7-269959, the project PAU+ DPI2011-27510 and the project CINNOVA 201150E088. B. Dellen was supported by the Spanish Ministry for Science and Innovation via a Ramon y Cajal fellowship. The authors would like to thank Sergi Foix for technical support in the experiments.

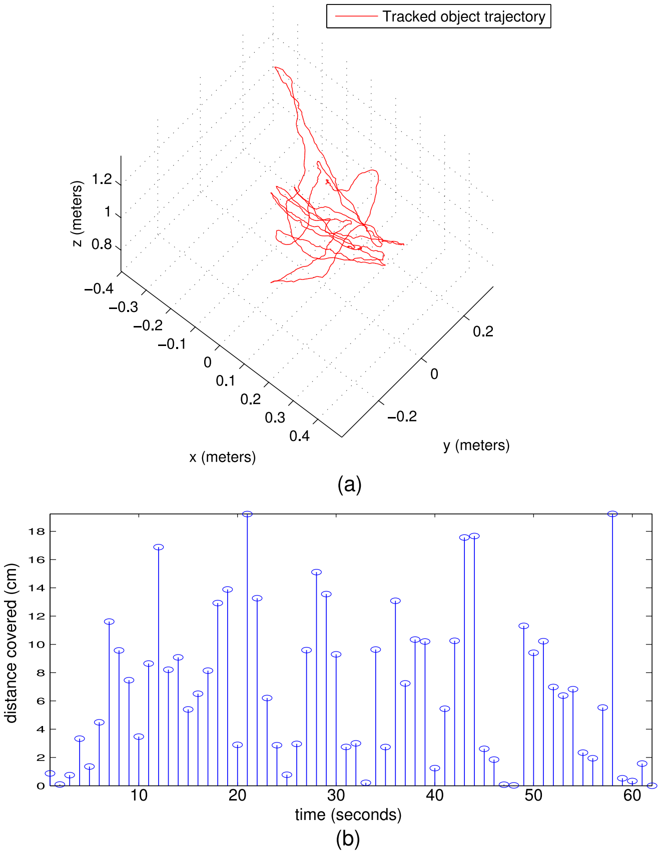

Fig. 9. Trajectory of tracking a cylindrical surface manipulated freely in 3D space (a) and the distance it covered within each second (b).

References

[1] F. Chaumette and S. Hutchinson, "Visual servo control. i. basic approaches," *IEEE Robotics Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.

[2] E. Malis, F. Chaumette, and S. Boudet, "2-1/2 D visual servoing," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 2, pp. 238–250, 1999.

[3] C. Kulpate, M. Mehrandezh, and R. Paranjape, "An eye-to-hand visual servoing structure for 3d positioning of a robotic arm using one camera and a flat mirror," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, pp. 1464–1470.

[4] K. Maeda, M. Minami, A. Yanou, H. Matsumoto, F. Yu, and S. Hou, "Frequency response experiments of 3-d pose full-tracking visual servoing with eye-vergence hand-eye robot system," in *Proceedings of SICE Annual Conference*, 2012, pp. 101–107.

[5] W. Song, M. Minami, F. Yu, Y. Zhang, and A. Yanou, "3-d hand and eye-vergence approaching visual servoing with lyapunouv-stable pose tracking," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 5210–5217.

[6] P. Li, F. Chaumette, and O. Tahri, "A shape tracking algorithm for visual servoing," in *IEEE International Conference on Robotics and Automation*, 2005, pp. 2847–2852.

[7] J. Kwon, K. M. Lee, and F. Park, "Visual tracking via geometric particle filtering on the affine group with optimal importance functions," in *CVPR*, 2009, pp. 991–998.

[8] A. Colome and C. Torras, "Redundant inverse kinematics: Experimental comparative review and two enhancements," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 5333–5340.

[9] M. A. Goodrich and A. C. Schultz, "Human-robot interaction: a survey," *Found. Trends Hum.-Comput. Interact.*, vol. 1, no. 3, pp. 203–275, Jan. 2007.

[10] P. Allen, A. Timcenko, B. Yoshimi, and P. Michelman, "Automated tracking and grasping of a moving object with a robotic hand-eye system," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 2, pp. 152–165, 1993.

[11] C. Archibald and M. van de Panne, "Tracking and grasping moving objects using reflex behaviour," in *Fifth Int. Conf. on Advanced Robotics*, vol. 1, 1991, pp. 643–648.

[12] K. Kondak, S. Binner, G. Hommel, and M. Neumann, "Time optimal manipulator control for sensor guided grasping of moving objects," in *International Conference on Intelligent Robots and Systems*, vol. 4, 2001, pp. 1912–1917.

[13] L. Ge and Z. Jie, "A real-time stereo visual servoing for moving object grasping based parallel algorithms," in *2nd IEEE Conf. on Industrial Electronics and Applicat.*, 2007, pp. 2886–2891.

[14] C. Smith and N. Papanikolopoulos, "Grasping of static and moving objects using a vision-based control approach," in *Int. Conf. on Intelligent Robots and Systems*, vol. 1, 1995, pp. 329–334.

[15] W. Bing and L. Xiang, "A simulation research on 3d visual servoing robot tracking and grasping a moving object," in *15th Int. Conf. on Mechatronics and Machine Vision in Practice*, 2008, pp. 362–367.

[16] K. Benameur and P. Belanger, "Grasping of a moving object with a robotic hand-eye system," in *Int. Conf. on Intelligent Robots and Systems*, vol. 1, 1998, pp. 304–310.

[17] M. Lei and B. Ghosh, "Visually guided robotic tracking and grasping of a moving object," in *Proceedings of the 32nd IEEE Conference on Decision and Control*, vol. 2, 1993, pp. 1604–1609.

[18] K. Hirota and H. Watanabe, "Grasping 2d irregularly moving object using fuzzy controlled arm robot," in *First International Symposium on Uncertainty Modeling and Analysis*, 1990, pp. 91–95.

[19] I. Siradjuddin, L. Behera, T. McGinnity, and S. Coleman, "A position based visual tracking system for a 7 dof robot manipulator using a kinect camera," in *International Joint Conference on Neural Networks*, 2012, pp. 1–7.

[20] B. Dellen, F. Husain, and C. Torras, "Joint segmentation and tracking of object surfaces along human/robot manipulations," in *International Conference on Computer Vision Theory and Applications*, vol. 1, 2013, pp. 244–251.

[21] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Journal of the Optical Society of America A*, vol. 4, pp. 629–642, 1987.

[22] J. Kwon and F. C. Park, "Visual tracking via particle filtering on the affine group." *Int. J. Rob. Res.*, vol. 29, no. 2-3, pp. 198–217, 2010.

[23] S. Zhou, R. Chellappa, and B. Moghaddam, "Visual tracking and recognition using appearance-adaptive models in particle filters," *IEEE Transactions on Image Processing*, vol. 13, no. 11, pp. 1491–1506, 2004.

[24] X. Li, W. Hu, Z. Zhang, X. Zhang, and G. Luo, "Robust visual tracking based on incremental tensor subspace learning," in *ICCV*, 2007, pp. 1–8.

[25] D. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *Int. Journal of Comput. Vision*, vol. 77, no. 1, pp. 125–141, May 2008.

[26] B. Siciliano, L. Sciavicco, G. Oriolo, and L. Villani, *Robotics: Modelling, Planning and Control.* Advanced Textbooks in Control and Signal Processing, Springer, 2009.

[27] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: Learning attractor models for motor behaviors," *Neural Comput.*, vol. 25, no. 2, pp. 328–373, feb 2013.

[28] M. Grundmann, V. Kwatra, M. Han, and I. Essa, "Efficient hierarchical graph-based video segmentation," in *CVPR*, June 2010, pp. 2141–2148.