INTERNATIONAL JOURNAL OF PRECISION ENGINEERING AND MANUFACTURING Vol. 13, No. 3, pp. 387-393

DOI: 10.1007/s12541-012-0049-8

Robotic Grasping Based on Efficient Tracking and Visual Servoing using Local Feature Descriptors

La Tuan Anh[1] and **Jae-Bok Song**[1,#]

1 School of Mechanical Engineering, Korea University, Anam-dong, Seongbuk-gu, Seoul, South Korea, 136-713
Corresponding Author / E-mail: jbsong@korea.ac.kr, TEL: +82-2-3290-3363, FAX: +82-2-3290-3757

KEYWORDS: Grasping, Object tracking, Visual servoing, SURF

In service robotic applications, grasping daily objects is an essential requirement. In this context, object and obstacle detection are used to find the desired object and to plan an obstacle-free path for a robot to successfully manipulate the object. In this paper, we propose a high-speed object tracking method based on a window approach and a local feature descriptor called speeded-up robust features (SURF). Instead of tracking the object in full image, we search and match features in the window of attention that contains only the object. Therefore, the tracked interest points are more repeatable and robust against noise. The visual servo controller uses geometrical features that are computed directly from the set of interest points, which makes the method robust against the loss of features caused by occlusion or changes in the viewpoint. Furthermore, these features decouple the translations and rotations from the image Jacobian, and also keep the object inside the camera's field of view. Various experiments with a robotic arm equipped with a monocular eye-in-hand camera demonstrate that objects can be grasped safely and in a stable manner in a cluttered environment using the proposed method.

Manuscript received: May 12, 2011 / Accepted: October 12, 2011

1. Introduction

Grasping an object is a simple task for humans, but not for robots. First, the robot needs to know the model of an object to distinguish it in a cluttered environment. Second, the robot must estimate a path to approach the object without colliding with other obstacles. Third, the optimum grasping locations – the points on the object where the robot should place its fingers – are required so that the robot can hold the object in a stable manner throughout subsequent steps.

In the literature, studies of object grasping show two main trends. The first is to design the grasping fingers to prevent slipping from a grasp, and to perform some simple reorientations and re-grasping. This requires fingers with tactile sensors as well as the use of position, force, and stiffness controllers. The second trend is to design an autonomous grasping system that mainly uses vision sensors to grasp known or unknown objects.

Several grasp algorithms using vision sensors have been proposed. In,[1] a method to determine two dimensional (2-D) grasp points using local visual features based on the 2-D contour and other properties (e.g., form-closure and force-closure) was proposed. Another approach is a learning method[2] that uses visual features to predict good grasp points for a wide range of objects. Using a synthetic data set for training, the 2-D grasp location in each image was predicted. Then, given two or more images of an object taken from different camera views, the three-dimensional (3-D) position of a grasp point was estimated.

In contrast to previous studies, our framework uses a vision sensor that focuses on a scheme to find, recognize, and grasp everyday objects (for example, the objects are showed in Fig. 1, right) in a cluttered environment. A robot with an eye-in-hand configuration (i.e., a camera mounted on the endpoint of a manipulator) was employed to find objects in the workspace (Fig. 1 left). When a robot recognizes an object, a position controller is used to align the manipulator with the object. After a grasp point on the object is chosen, the robot approaches, grasps the object, and moves the object to the desired target position.

The visual servo-controller is used in this context to align the manipulator with the object. Most studies of visual servoing have been limited to the final alignment in which point-to-point[9,10] or color[11] visual information is employed. The approach used in our study is visual servoing for a gripper mounted on an autonomous robot that is capable of using its arm and eye-in-hand camera to recognize and move a large set of objects in real time. We use an

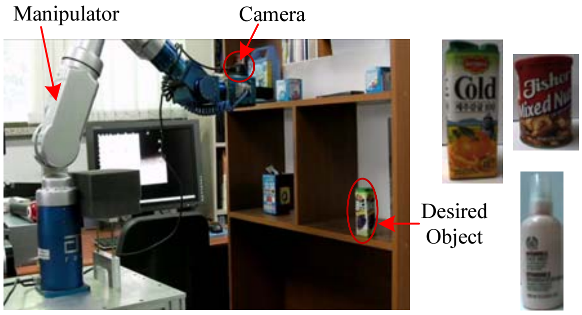


Fig. 1 Object-grasping workspace and some everyday objects

image-based visual servo (IBVS) controller instead of position-based visual servoing (PBVS) because the IBVS does not need precise calibration or modeling since a closed-loop scheme is used directly in the image frame. Furthermore, the IBVS is robust to measurement and modeling errors if the initial and desired configurations are closed. Therefore, the stability is ensured in the neighborhood of the desired position.

The goal of the IBVS is to control a robot to a specific pose in an environment by regulating the error term estimated by matching the image features of reference and current images to zero. Therefore, the visual features based on local feature descriptors such as scale-invariant feature transformation (SIFT)[3] and speeded-up robust features (SURF)[4] offer particular advantages to the IBVS method when applied to everyday objects. Most importantly, these methods are independent of changes in scale, orientation, illumination, and affine transformations. Furthermore, the features are uniquely identifiable across multiple views of an object,[7] which allows the development of a robust model-free IBVS.

Research on the use of invariant interest points to estimate features for visual servo-control inspired our approach.[5,6] Visual servoing with the SIFT algorithm was first introduced in.[5] Their approach focused on feature extraction and viewpoint reconstruction based on epipolar geometry. However, in this method, the image sequences under examination are not significantly different, so the SIFT feature points must be in the neighborhood of the matched feature point in the previous frame. In addition, over the entire trajectory, a conformity constraint would remain to choose strong candidates. This set of feature points was only robust to small deviations from a specific trajectory, and hence only useful to track the object when the camera moved along a similar trajectory that was learned in the training phase. In comparison, our method does not require the training phase and can work for arbitrary trajectories.

Another approach[6] emphasized the design of an image-based controller that augmented point features by the additional attributes of scale and keypoint orientation of SIFT features, and made them suitable to control the distance to the object and the rotation around the optical axis. However, this method required a set of n matched keypoints to be detected in all trajectories, and a single incorrect reference point might affect proper convergence to the goal pose. This assumption is difficult to validate in a real environment in which partial object occlusion and a change in viewpoint and illumination usually occur. Furthermore, it is difficult to keep the entire object in the camera's field of view throughout the trajectories.

To enable tracking of everyday objects in interactive frame rates, we propose using the window approach for local feature descriptors (e.g., the SURF algorithm) to search for interest points at locations in each frame where they are most likely to appear. From the set of interest points, robust and intuitive features of the object are selected for visual servoing. These features decouple the translations and rotations from the image Jacobian and keep the object inside the field of view of the camera. This approach also allows the controller to deal with partially occluded objects which often occur in dynamic environments. The effectiveness and robustness of the visual servo-controller was experimentally validated using a real robot arm with an eye-in-hand configuration and a standard two-finger gripper.

This paper is organized as follows. Section 2 introduces the proposed object tracking strategy based on the window of attention from SURF. We explain how the integration of the window approach and SURF allows high speed detection of an object. Section 3 describes the derivations of the set of visual features and the corresponding image Jacobian used to design a visual servo-controller. In Section 4, we describe an experiment involving grasping and placing an object in the desired location using a manipulator and a monocular camera-in-hand, based on the proposed visual servoing controller. Our conclusions are given in Section 5.

2. Object Tracking Strategy

Object tracking is an important task in computer vision. Tracking objects in real environments can be complicated due to the change in scene illumination, object-to-object and object-to-scene occlusions, camera motion, and real-time processing requirements. The proposed object tracking algorithm contains three key steps: detecting the object of interest, tracking the object from frame to frame, and an analysis of tracked object to obtain the features for visual servoing.

2.1 Selection of Interest Points

Every tracking method requires an object detection mechanism, either in every frame or when the object first appears. In this study, we extended the interest point detection algorithm so that the interest points are tracked across the video frames efficiently. Although the proposed approach is suitable for many feature-based algorithms, we focused on the SURF algorithm.

The SURF algorithm is an efficient object recognition algorithm with a fast scale-invariant and rotation-invariant detector and descriptor. Therefore, the SURF algorithm is particularly attractive for model-free image-based visual servoing because the same interest points are visible and robustly related across the different views. However, it has been noted that SURF feature points are not always uniquely matched between consecutive images when grabbed from a camera moving on an arbitrary trajectory with the

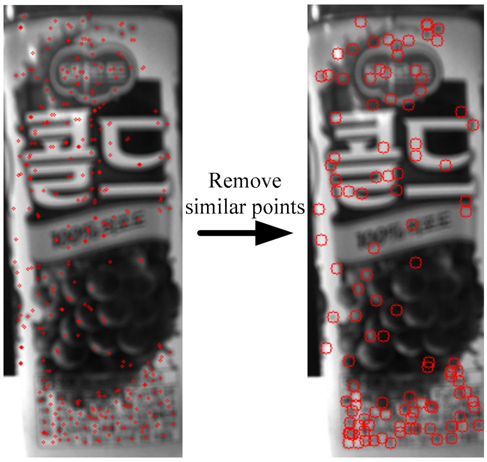


Fig. 2 Extracted and verified SURF interest points

change of posture of manipulator and direction of camera. Thus, selection of the optimum interest points in terms of the discrimination, stability, and detection ability across the workspace is crucial. To detect an object, SURF interest points in the current image are matched with their corresponding interest points from the reference image in terms of their distinctive descriptors. Naturally, the interest point descriptors of the same feature in different views are, although similar, not exactly identical. This variation might lead to incorrect matches among points if two different interest points share similar descriptors. Hence, those pairs of ambiguous interest points in the reference image that are too similar to each other are rejected to avoid subsequent confusion between them. Depending on the texture of the object and the parameter settings of the algorithm, about 50-150 verified interest points passed the similarity test and were included in the database of reference interest points. Figure 2 shows the set of extracted and verified points in the reference view. The dots represent the initial set of 352 candidate interest points. From this initial set, 112 points marked by circles exhibited sufficiently distinct interest point descriptors to pass the similarity test.

Furthermore, to increase the recognition rate in the significant change of direction of camera, our method to make additional images seen from various viewpoints for an identical object is used.[7] If the object can be assumed to be locally planar, an additional database image with a different viewpoint can be created by warping the original object image using an affine transformation which approximates the actual homography.

2.2 Window Approach for SURF

After the reference image is processed to select the confident interest points, a new query image is grabbed from the camera to be matched to the reference image. Since we must recognize and track small everyday objects (e.g., bottles, cans, etc.) and grasp them in a cluttered environment, the number of detected interest points of the object is important. Therefore, we used a high resolution image (640 x 480) in our experiments.

With high image resolution, the time required to process the image using the SURF algorithm increases. If we can extract the region of interest (ROI) that contains only our object, the time

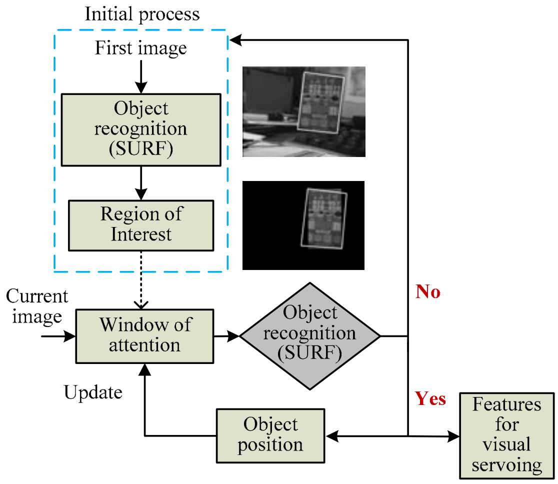


Fig. 3 Scheme of the window approach for SURF

needed for the recognition process will decrease. Furthermore, with a smaller set of interest points, the time for the matching process also decreases, and the algorithm will run much faster and remove most outliers. Based on this idea, our object tracker significantly increases the speed by providing only the region in the image that is occupied by the object at every time instant.

Figure 3 summarizes a scheme of the window approach using SURF detection. The object region is jointly estimated by iteratively updating the object's location and the region information obtained from the previous frame. After the process of object recognition based on the SURF algorithm, the ROI containing the object is obtained using the object's position. In the successive image, only the interest points in this updated ROI are extracted and matched to find the object. With the smaller set of interest points, the time for the SURF process also decreases and the algorithm runs much faster. If no object is detected, the initial process is repeated. Then, the features used for visual servo-control are computed from the set of interest points.

In detail, the object is recognized using the SURF algorithm from the first image of the image sequences captured by the camera by matching the interest points between the current and the reference views. The fundamental matrix can be computed from the set of perfectly matched interest points. However, in practice, not all of the interest points are distinct. Therefore, once the closest match of the interest point in an image to the point in another image is found, it should be checked to determine whether it is an outlier or inlier. We used the nearest Euclidean distance of the 64-dimensional vector of the interest point description and applied a threshold to determine if the interest point is an inlier. Another approach used to find the interest point matching is to examine the second closest distance to the candidate points. Because of the interest point pre-selection (as described in Section 2.1), all similar interest points in the reference image are removed. Then, if the closest distance is significantly smaller than the second closest distance, it is highly likely that the match is correct. Finally, the random sample consensus (RANSAC) algorithm was used to eliminate the remaining mismatches and to estimate the fundamental matrix. The window that contains the object is

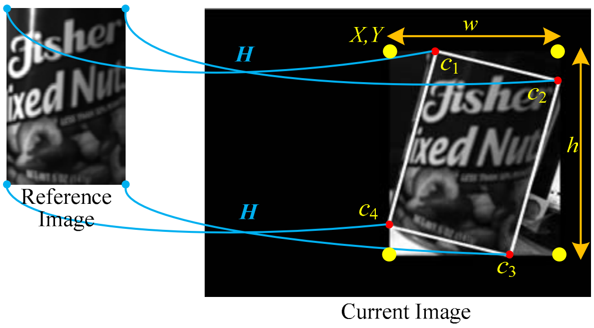


Fig. 4 Construction of window of attention

obtained as illustrated in Fig. 4. The four corners c_1, c_2, c_3, c_4 (shown in red) of the object in the current image are estimated using the fundamental matrix H and the corresponding corners in the reference image. The window (shown with yellow corners) that covers the entire object in the current image can be defined by the corner (X, Y), the width w, and the height h where

$$\begin{cases} X = c[\min x], \\ Y = c[\min y], \\ w = c[\max x] - c[\min x], \\ h = c[\max y] - c[\min y]. \end{cases} \tag{1}$$

where $\begin{cases} c[\min x] = \min\limits_{x}(c_1, c_2, c_3, c_4), \\ c[\max x] = \max\limits_{x}(c_1, c_2, c_3, c_4), \\ c[\min y] = \min\limits_{y}(c_1, c_2, c_3, c_4), \\ c[\max y] = \max\limits_{y}(c_1, c_2, c_3, c_4). \end{cases}$

Since this method does not use the contours of the object, if the object is partially occluded by other objects or the scene, the window that contains the object can still be constructed. In the next image sequence, this window is used as the ROI, and the interest points are only extracted from this window. If no object is detected, the posture of the robot is changed and the initial process is repeated; i.e., the full image is searched to find the object. Therefore, the robust interest points can be found effectively by using the window approach. Here we assume that the images in a sequence are not significantly different from each other when the camera moves over an arbitrary trajectory. This turns out to be a valid assumption because the algorithm can track the object in the presence of partial occlusion. The interest point extraction and matching based on our proposed method runs at high speed, which makes the difference between two successive images minor. The features used for visual servo-control, which is described in Section 3, are computed from the robust set of interest points.

3. Visual Servo-Controller Design

We could choose four corners of the object obtained from the object tracking stage as the features for the visual servo-controller. However, the conventional image-based visual servoing scheme with point features suffers from the coupling between translational and rotational components, which might result in singularities or

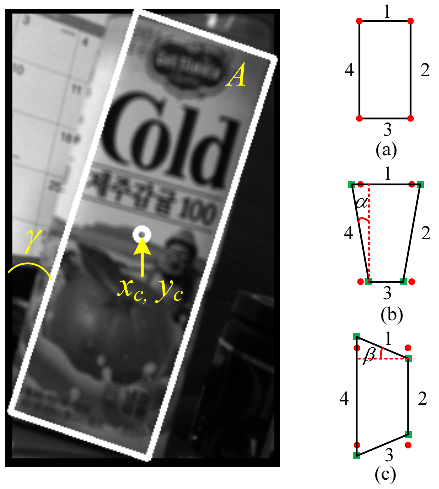


Fig. 5 Selection of visual features

infeasible camera trajectories.[8] Therefore, we selected s, which is a set of six features (s_x, s_y, s_z, s_α, s_β, s_γ), to control the six degrees of freedom (DOFs) of a robot as described in Section 3.1.

3.1 Selection of Visual Features

A set of interest points is automatically extracted from the image of the desired object in the visual tracking step. Our approach uses the intuitive geometrical features directly generated from this set. As described in Fig. 5 (left), the visual features s_x and s_y, which capture the translations along the x- and y-axes, are expressed by the virtual center (x_c, y_c) of the object in the current image. This center is computed from the center of the object in the reference image and the fundamental matrix H, which is estimated from the set of matched interest points. The feature s_z is selected to be the area A of the object in the image that is obtained from the four corners of the object image. The visual feature s_γ, defined by the rotation angle γ around the camera axis, is computed from the object contours and the window covering the object. In addition, the actual distance z is computed from the area A under the assumption that the desired area A^* and distance z^* at the reference image are known:

$$z = z^* \sqrt{\frac{A^*}{A}} \tag{2}$$

The advantages of using the area as a feature are that it is rotation-invariant, thereby decoupling camera rotation from the z-axis translation, and it can be easily computed from four corners of the object image.

Six-DOF visual control was completed by the visual features s_α and s_β that captured the rotations around the x- and y-axes. Both features represent the effects of perspective distortions on the lines caused by the yaw and the pitch motion of the camera. Visual features s_α and s_β that captured the rotations are coupled with the translation along the x- and y-axes, however we can remove the coupling by using our previous study[7] in which manipulator adjusts its position by rotating around the x- and y-axes to make s_α and s_β converged to small value. Figure 5 (right) illustrates the effect for a rectangular configuration of four corner points that form four lines.

Figure 5(a) depicts the image of the rectangle for parallel features and the image plane. Figure 5(b) shows the image with the camera tilted around the x-axis and a compensation of the shift along the y-axis. The distortion increased the length of line 1 and simultaneously decreased the length of line 3. The dilation and compression of the lines are captured by angle α. Figure 5(c) represents the equivalent effects of dilation and compression of the lines caused by rotations about the y-axis. Similarly, the dilation and compression of the lines are described by angle β.

3.2 Controller Design

The visual control scheme refers to the image Jacobian for the set of visual features defined in Section 3.1. The visual features (s_x, s_y, s_z, s_α, s_β, s_γ) were specifically designed to be sensitive to one particular degree of motion and relatively invariant with respect to the remaining motions. This property suggests a simplified controller design in which the off-diagonal elements of the image Jacobian matrix are neglected and the control obtains a one-to-one scalar relationship between the features and the degrees of motion. Furthermore, the control of the object center purely by T_x and T_y contributes to robustness because the image features are less likely to disappear from the field of view. The camera motions T_x, T_y, T_z, w_x, w_y, and w_z were calculated according to the feature error $e = [s_x, s_y, s_z, s_\alpha, s_\beta, s_\gamma]^T - [s_x^*, s_y^*, s_z^*, s_\alpha^*, s_\beta^*, s_\gamma^*]^T$ and the gain matrix λ:

$$\begin{aligned} &T_x = -\lambda_x \cdot e_x, \quad T_y = -\lambda_y \cdot e_y, \quad T_z = -\lambda_z \cdot e_z, \\ &w_\alpha = -\lambda_\alpha \cdot e_\alpha, \quad w_\beta = -\lambda_\beta \cdot e_\beta, \quad w_\gamma = -\lambda_\gamma \cdot e_\gamma. \end{aligned} \tag{3}$$

The decoupled controller controlled each degree of motion by a single feature and ignored the remaining cross-couplings between the feature and the camera motion. This approach is robust to occlusion, illumination changes, and perspective distortions because the performance of the controller are not affected as long as some interest points in the current image still match the reference points.

3.3 Selection of the Grasping Point

Using a stored image taken from the reference position, the manipulator can move in such a way that the current camera view is gradually changed to match the stored reference view. After the visual servoing step in which the arm is aligned with the object, grasping can be performed.

However, in the grasping step in dynamic environments, the object itself can be partially occluded by an obstacle. If the obstacles partially occlude an object, the grasping position must change to avoid a collision with the obstacle. A method to select the grasping position in the cluttered scenarios with occluded object is described in Fig. 6 in which the obstacle is in the red rectangle.

In detail, the object image is divided into three bins of the same size. If any bin is occluded by the obstacle, the number of matched interest points in the bin will decrease. By comparing the ratio of the interest points in the reference bins to those in the current bins, the occluded part of the object can be detected. The bin with the highest ratio is chosen as the suitable position for grasping. The white circle in Fig. 6 is the position on the object that coincides with the center of the end-effector in grasping step.

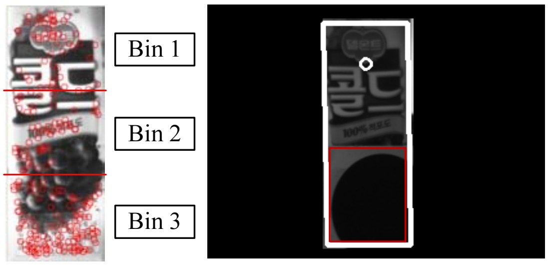
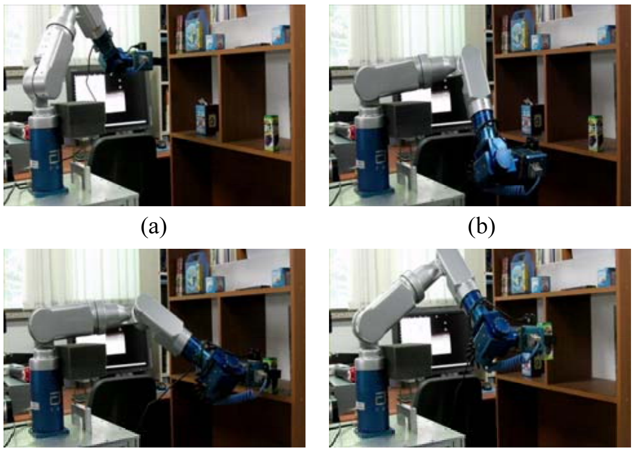


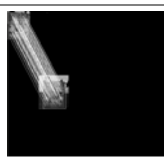
Fig. 6 Selection of the grasping position of an occluded object



(a) (b)
(c) (d)

Fig. 7 Grasping experiment

Table 1 Required time to extract interesting points: (a) SURF detector, (b) background subtraction method, (c) our tracking method

Case	Recognition case	Specifications
a		Object interest points: 128 Image interest points: 679 Extraction time: 110 ms
b		Object interest points: 128 Image interest points: 238 Extraction time: 77 ms
c		Object interest points: 128 Image interest points: 194 Extraction time: 17 ms

4. Experimental Results

The goal of this study is to use vision for real-world grasping tasks such that a robot should be capable of using its arm and camera to find, recognize, and pick up everyday objects. In this experiment, we conducted the task of grasping and positioning an object to a desired place using a LWR arm (50μm repeatability) and an eye-in-hand “firefly” camera (60frames/s) as described in Fig. 1.

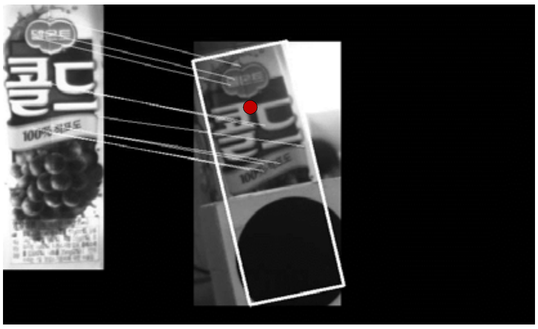


Fig. 8 Object recognition based on our method

In step 1, the robot searches the workspace to find the object using the window approach of the SURF method. In step 2, the robot uses the features obtained from step 1 in the visual servo-controller to track and approach the object. In step 3, the robot selects the grasping position and grasps the object. After grasping the object, the robot moves to the desired location to place the object. The experimental procedure for vision-assisted object manipulation is depicted in Fig. 7.

Using the algorithms described in Section 3, the time required by the original SURF algorithm to process an image significantly decreased, as shown in Table 1. In case (a), the object was searched for in a 640 x 480 image by the SURF detector. The algorithm found 679 interest points, and the time required to extract these points was 110 ms. By using a background subtraction algorithm (case (b)), the algorithm found 238 interest points in 77 ms. If we extracted the region of interest (ROI) that only contained the object using the method described in Section 2, then it took only 17 ms to extract 194 interest points (case (c)). Overall, for tracking purposes, we achieved a more than ten-fold speedup compared to the original SURF scheme.

The proposed visual servoing method was used to control the end-effector to a previously demonstrated pre-grasping pose. The manipulator was moved so that the current camera view was gradually changed to match the stored reference view. The object with large viewpoint change (over 60°) also can be detected if the additional database is used.[7] Once the desired pose with respect to the object was attained, the end-effector moved to grasp the object. The suitable grasping points were selected, and the grasping closure was finally executed. In the case the object is partially occluded by other objects or the scene, the window that contains the object can still be constructed as described in Fig. 8 and a grasping point (red circle) can be re-estimated by the method described in Section 3.1.

Figures 9 and 10 show the evolution of image space error and camera velocity during the visual servoing pre-grasping phase. Both errors quickly converged to zero. The residual error for the translational motion is less than 1 mm, and less than 1° for the rotational motion compare to the position where we took the reference image. This level of accuracy was sufficient for our object manipulation.

The computational demand for extracting and matching SURF interest points based on our window approach enabled real time visual control at a frame rate of 15 Hz. Our experimental results demonstrate that by using the proposed approach, it is possible to control a manipulator with a monocular camera to grasp everyday objects based on a visual servo-controller.

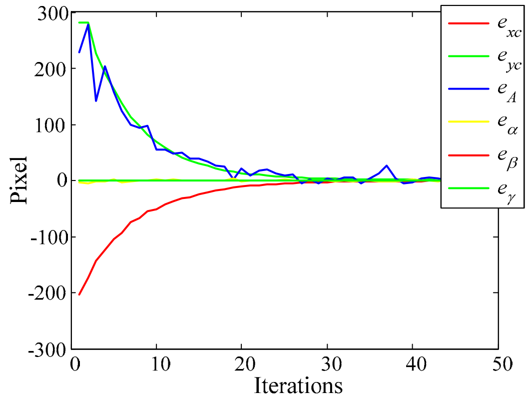


Fig. 9 Image space error

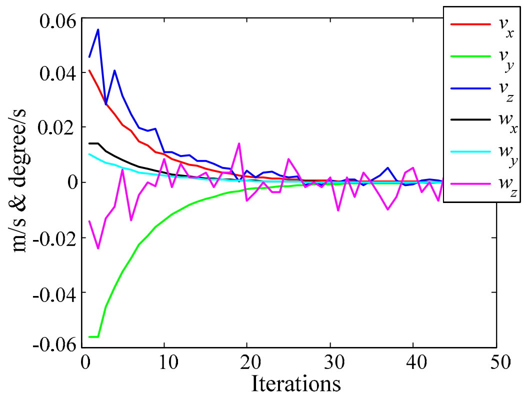


Fig. 10 Velocity of camera

5. Conclusion

We proposed a method that integrates visual tracking and visual servoing based on a window approach and local feature descriptor, called SURF, to enable a manipulator to grasp everyday objects in a cluttered environment. Based on our experiments, the following conclusions are drawn:

1. The tracking strategy is generic because it does not depend on the object model; rather, it automatically extracts the interest points from object images.
2. The window approach not only increases the tracking speed significantly, but also constructs intuitive geometrical features for the visual servoing process.
3. The visual servoing on intuitive features exhibits the decoupling of features and degrees of freedom, and this method is robust against occlusion and changes in viewpoint.

ACKNOWLEDGEMENT

This work was supported by Human Resources Development Program for Convergence Robot Specialists (Ministry of Knowledge Economy, NIPA-2011-C7000-1001-0003), and by Basic

Science Research Program through the National Research Foundation funded by the Ministry of Education, Science and Technology (No. 2011-0001150).

REFERENCES

1. Sanz, P. J., Requena, A., Inesta, J. M. and Del Pobil, A. P., "Grasping the not-so-obvious: Vision-based Object Handling for Industrial Applications," IEEE Robotics and Automation Magazine, Vol. 12, No. 3, pp. 44-52, 2005.
2. Saxena, A., Driemeyer, J. and Ng, A. Y., "Robotic Grasping of Novel Objects using Vision," Int. J. Robotics Res., Vol. 27, No. 2, pp. 157-173, 2008.
3. Bay, H., Tuytelaars, T. and Gool, L. V., "SURF: Speeded Up Robust Features," Computer Vision: Lecture Notes in Com. Sci., Vol. 3951, pp. 404-417, 2006.
4. Lowe, D. F., "Distinctive Image Features from Scale-Invariant Keypoints," Int. J. Com. Vision, Vol. 60, No. 2, pp. 91-110, 2004.
5. Shademan, A. and Janabi-Sharifi, F., "Using Scale-Invariant Feature Points in Visual Servoing," Proc. of SPIE, Vol. 5603, pp. 63-70, 2004.
6. Hoffmann, F., Nierobisch, T., Seyffarth, T. and Rudolph, G., "Visual Servoing with Moments of SIFT Features," Proc. of IEEE Int. Conf. on Systems, Man, and Cybernetics, pp. 4262-4267, 2006.
7. Anh, L. T. and Song, J. B., "Improvement of Feature-based Object Recognition using Affine Transformation for Mobile Robot Navigation," Proc. of Int. Conf. on Ubiquitous Robots and Ambient Intelligence, pp. 825-829, 2008.
8. Chaumette, F., "Potential Problems of Stability and Convergence in Image-based and Position-based Visual Servoing," The Confluence of Vision and Control: Lecture Notes in Control and Info. Sci., Vol. 237, pp. 66-78, 1998.
9. Kim, C. S., Mo, E. J., Han, S. M., Jie, M. S. and Lee, K. W., "Robust Visual Servo Control of Robot Manipulators with Uncertain Dynamics and Camera Parameters," Int. J. Control, Automation and Systems, Vol. 8, No. 2, pp. 308-313, 2010.
10. Corke, P. I. and Hutchinson, S., "A New Partitioned Approach to Image-based Visual Servo Control," IEEE Trans. on Robotics and Automation, Vol. 17, No. 4, pp. 507-515, 2001.
11. Kar, I. and Behera, L., "Visual Motor Control of a 7 DOF Robot Manipulator using a fuzzy SOM Network," Intelligent Service Robotics, Vol. 3, No. 1, pp. 49-60, 2010.