International Conference on Control, Automation and Systems 2010
Oct. 27-30, 2010 in KINTEX, Gyeonggi-do, Korea

Object Tracking and Visual Servoing using Features Computed from Local Feature Descriptor

La Tuan Anh and Jae-Bok Song
School of Mechanical Engineering, Korea University, Seoul, Korea
(E-mail: latuananh, jbsong@korea.ac.kr)

Abstract: In service robot applications, tracking and visual servoing are essential to find objects and position the end-effector of a robot to manipulate an object. In this paper, we propose a high-speed object tracking method based on a window approach and a local feature descriptor, SURF (Speeded-Up Robust Features). The visual servo controller uses geometrical features that are computed directly from the set of SURF interest points, which makes a method robust to the loss of features caused by occlusion or changes in the view point. Furthermore, these features decouple the translations and rotations from the image Jacobian and also keep the object inside the field of view of the camera. Various experiments with a robotic arm equipped with a monocular eye-in-hand camera demonstrate that objects can be grasped safely and stably in the cluttered environment using the proposed method.

Keywords: Visual servoing, speeded-up robust features (SURF), grasp.

1. INTRODUCTION

Visual servoing has received significant attention during the past few years. Most studies are still limited to final alignment in which point-to-point visual servoing is employed. However, manipulation of everyday objects is required in service robot applications.

The goal of image-based visual servoing (IBVS) is to control a robot to a specific pose in the environment by regulating to zero an error term estimated by matching the image features between the reference and the current images. Therefore, visual features based on local feature descriptors such as scale invariant feature transformation (SIFT) [1] and speeded-up robust features (SURF) [2] offer particular advantages for the purpose of the IBVS with everyday-life 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, which allows the development of robust model-free IBVS.

Some works that inspired our approach were the application of the invariant interest points to the estimation of features for visual servo control [3, 4]. Visual servoing with SIFT was first introduced by [3]. Their approach focused on the feature extraction and view point reconstruction based on the epipolar geometry. However, in this method image sequences under examination could not differ so significantly that 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 was useful to track the object when the camera moved along a similar trajectory which learned from the training phase. In comparison, our method does not require the training phase and can work in arbitrary trajectories.

In another approach, [4] 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 the proper convergence to the goal pose. This assumption was difficult to validate in the real environment in which the partial object occlusion and the change of viewpoint and illumination usually occurred. Furthermore, it was difficult to keep the entire object in the camera's field of view in all trajectories.

In this paper, we propose to use the window approach for one of the local feature descriptors, SURF algorithm, to select robust and intuitive features of the object for visual servoing. The task is to position the robot's end-effector in the desired pose relative to the object. 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 the object occluded partially, which always occurs in dynamic environments. The effectiveness and robustness of the visual servo controller is experimentally validated using a real robot arm with an eye-in-hand configuration and a standard two-finger gripper in the grasping task.

The paper is organized as follows. Section 2 introduces the object tracking strategy based on the window of attention of SURF. It explains how the integration of the window approach and SURF can make a high-speed detection of an object. Section 3 describes derivations of the set of visual features and the corresponding image Jacobian to design a visual servo controller. In Section 4 the experiment of grasping and putting an object to a desired place using a light-weight robot (LWR) manipulator and a monocular camera-in-hand based on the proposed visual servoing controller is described. The paper concludes with a summary in Section 5.

2. OBJECT TRACKING STRATEGY

Every tracking method requires an object detection mechanism either in every frame or when the object first appears. The speeded-up robust features (SURF) algorithm [3] has been demonstrated as an efficient object recognition method with a fast scale- and rotation-invariant detector and descriptor. Our object tracker increases the speed by providing the complete region in the image that is occupied by the object at every time instant. The object region is jointly estimated by iteratively updating the object location and region information obtained from the previous frame. Fig. 1 summarizes the scheme of the window approach using SURF detection. After the process of object recognition based on the SURF algorithm, the ROI which contains the object is obtained using the object 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 it makes the algorithm run much farther.

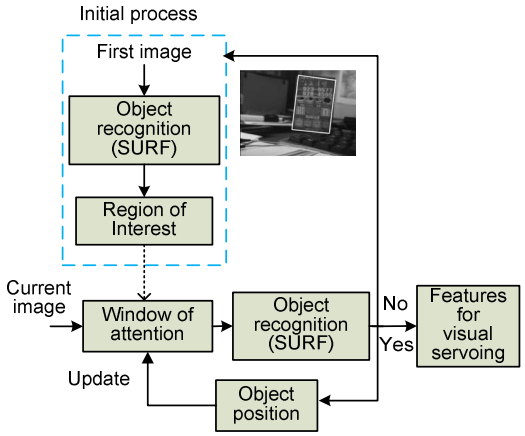

Fig. 1 Scheme of the window approach for SURF.

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, but, in practice, not all of the interest points are distinctive. 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 see whether it is an outlier or inlier. We have used the nearest Euclidean distance of the 64-vector of the interest point description and applied a threshold to decide whether or not the interest point is an inlier. Another approach used to find the interest point matching is to look at the second-closest distance to the candidate points. Because of the interest point pre-selection, 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 is used to eliminate the rest of mismatches and to estimate the fundamental matrix. The window which contains the object is obtained as illustrated in Fig. 2. The 4 corners c_1, c_2, c_3, c_4 (in red color) of the object in the current image are estimated using the fundamental matrix **H** and the corresponding corners in the reference image. The window (with the yellow corners) which 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

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

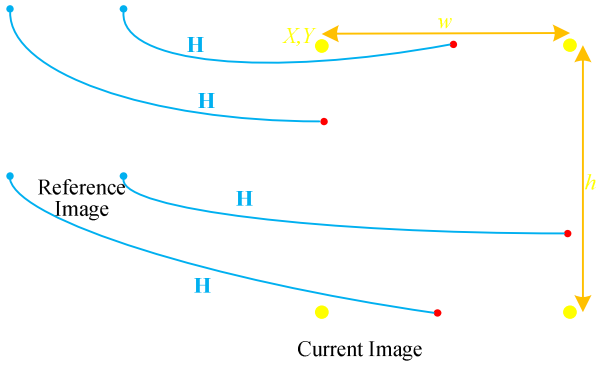

Fig. 2 Construction of window of attention.

Since this method does not use the contours of the object, if the object is partially occluded by other objects or scene, the window that contains the object can still be constructed as described in Fig. 3. 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 initial process is repeated, i.e. the full image is searched to find the object. The robust SURF interest point can be found effectively 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 SURF algorithm can track the object in the presence of partial occlusion and the interest point extraction and matching based on our method run at a high-speed and it made the difference between two successive images minor. The features used for visual servo control are computed from the robust set of interest points.

Fig. 3 Object recognition based on the window approach of SURF.

3. VISUAL SERVO CONTROLLER

We can choose four corners of the object obtained from the object tracking stage as features. 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 infeasible camera trajectories [6]. Therefore, we select s, which is a set of six features (s_x, s_y, s_z, s_α, s_β, s_γ), to control the six DOFs of a robot.

The aim of the vision-based control scheme is to minimize an error $e(t)$ between the current values s of the features and the desired values s^* of the features, which is defined by:

$$e(t) = s - s^* \tag{2}$$

Visual servoing schemes mainly differ in the way that s is designed. In IBVS s consists of a set of features that are available in the image data. The most straightforward approach is to design a velocity controller. To do this, we require the relationship between the time variation of s and the camera velocity. Let the spatial velocity of the camera be denoted by $v_c = (T_x, T_y, T_z, w_\alpha, w_\beta, w_\gamma)$, where T is the instantaneous linear velocity of the origin of the camera frame and w is the instantaneous angular velocity of the camera frame. The relationship between $\dot{s}$ and v_c is given by:

$$\dot{s} = L_s v_c \tag{3}$$

where $L_s \in \Re^{k\times 6}$ is the image Jacobian related to s. If the robot controller allows v_c to be taken as inputs, a control law can be obtained:

$$v_c = -\lambda L_s^+ (s - s*) \tag{4}$$

in which L_s^+ denotes the pseudo-inverse of the image Jacobian and λ is a gain matrix.

3.1 Selection of visual features

Our approach uses the intuitive geometrical features directly generated from the visual tracking stage. As described in Fig. 4(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 which is obtained from the four corners of the object. The visual feature s_γ, defined by the rotation angle γ around the camera axis, is computed from the object contours and the window covered 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

$$z = z^* \sqrt{\frac{A^*}{A}} \tag{5}$$

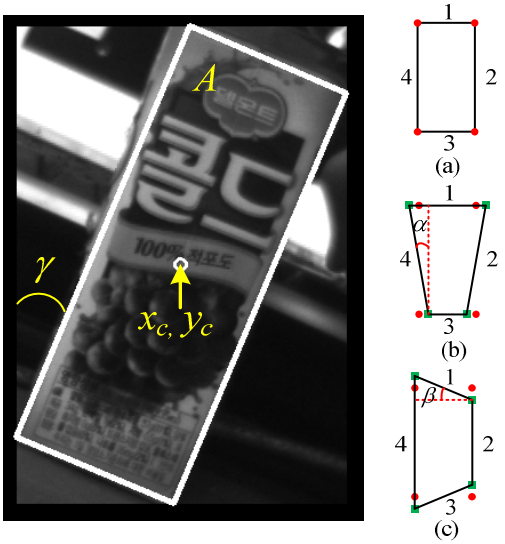

Fig. 4 Selection of visual features.

The 6-DOF visual control is completed by the visual features s_α and s_β that capture the rotations along the x and y-axes. Both features represent the effects of perspective distortions on the lines caused by the yaw and pitch motion of the camera. Fig. 4(right) illustrates the effect for a rectangular configuration of four corner points that form four lines. Fig. 4(a) depicts the image of the rectangle for parallel feature and image plane, Fig. 4(b) shows the image with the camera is tilted around the x-axis and a compensation of the shift along the y-axis. The distortion increases the length of line 1 and simultaneously decreases the length of line 3. The dilation and compression of the lines are captured by the angle α. Fig. 4(c) represents the equivalent effects of dilations and compression of the lines caused by rotations along the y-axis. Similarly, the dilation and compression of the lines are captured by the angle β.

3.2 Controller design

The visual features (s_x, s_y, s_z, s_α, s_β, s_γ) are designed

such that they are sensitive to a certain degree of motion and relatively invariant to the remaining motions. This property suggests a simplified controller design in which the off-diagonal elements of the image Jacobian are neglected and the control assumes a one-to-one scalar relationship between features and degrees of motion. Furthermore, the control of the object center purely by v_x and v_y contributes to the robustness as image features are less likely to disappear from the field of view. The camera motion v_x, v_y, v_z, w_x, w_y, and w_z are calculated according to the feature error between the current and the design images

$$e=[s_x\ s_y\ s_z\ s_\alpha\ s_\beta\ s_\gamma]^T-[s_x^*\ s_y^*\ s_z^*\ s_\alpha^*\ s_\beta^*\ s_\gamma^*]^T \quad (6)$$

and the gain matrix λ

$$T_x=-\lambda_x\cdot e_x,\ T_y=-\lambda_y\cdot e_y,\ T_z=-\lambda_z\cdot e_z,$$
$$w_\alpha=-\lambda_\alpha\cdot e_\alpha, w_\beta=-\lambda_\beta\cdot e_\beta, w_\gamma=-\lambda_\gamma\cdot e_\gamma. \quad (7)$$

4. EXPERIMENTAL RESULTS

The aim of this research is to use vision for real-world grasping tasks, in which the robot should be capable of using its arm and eye-in-hand camera to find, recognize and pick up everyday objects. In this experiment, the task of grasping and putting an object to a desired place using a LWR robot and an eye-in-hand "Firefly" camera is based on the following scheme. In step 1, the robot wandered around the workspace to find the object using the window approach of the SURF method. In step 2, the robot used the features obtained from step 1 in the visual servo controller to track and approach the object. In step 3, the robot chose the grasping position and grasped the object. After grasping the object, the robot moved to the desired target to place the object. The experimental procedure for the two-finger grasping for vision-assisted object manipulation is depicted in Fig. 5.

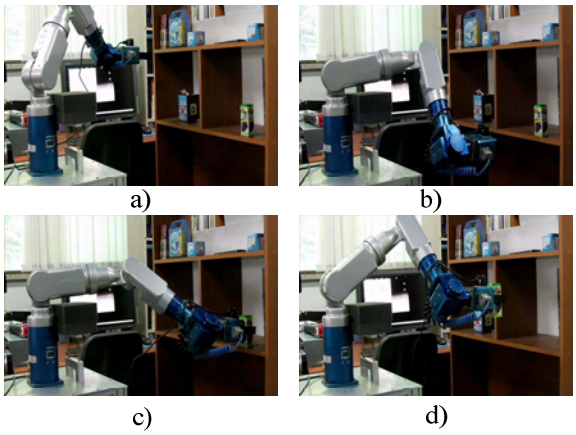

Fig. 5 Grasping experiment.

The proposed visual servoing method is used to control the end-effector to a previously demonstrated pre-grasping pose. Using a stored image taken from the reference position, the manipulator can be moved so that the current camera view is gradually changed to match the stored reference view. Once the desired pose with respect to the object is attained, the end-effector is moved forward by a fixed amount equal to the distance between the camera and the link to which it is attached in order to position the gripper above the object. The suitable grasping points are selected to avoid a collision and the grasping closure is finally executed as described in [5].

Figures 6 and 7 show the evolution of image space error and camera velocity during the visual servoing pre-grasping phase. Both errors converge quickly to zero. The residual error for the translational motion is less than 1mm and less than 1° for the rotation. It should be noted that this accuracy is sufficient for our object manipulation.

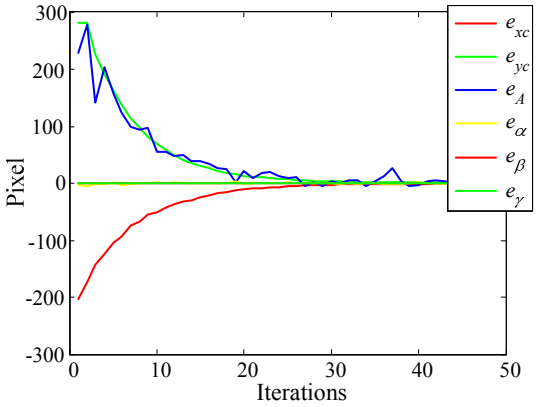

Fig. 6 Image space error.

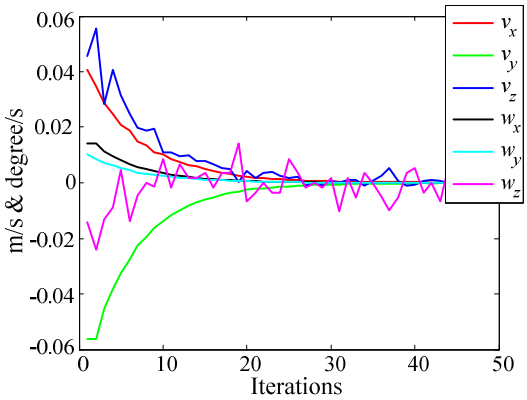

Fig. 7 Velocity of camera.

5. CONCLUSIONS

In this paper, we have proposed a method which integrates visual tracking and visual servoing based on a window approach and a local features descriptor, SURF, to enable the manipulator to grasp everyday object in cluttered environments. From various experiments, the following conclusions are drawn:

1. The tracking strategy is generic as it does not depend

on the object model but automatically extracts the interest points from object images.

2. The window approach not only increases the tracking speed significantly but also constructs the intuitive geometrical features for the visual servoing process.
3. The foundation of visual servoing on the intuitive features exhibits the decoupling of features and degrees of freedoms and renders a method robust to occlusion or changes in view point.

Acknowledgement

This work was supported by the Center for Autonomous Intelligent Manipulation under Human Resources Development Program for Robot Specialists (Ministry of Knowledge Economy).

REFERENCES

[1] D.F Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision,* vol.60, no. 2, pp. 91-110, 2004.

[2] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," *Computer Vision - ECCV 2006*, (A. Leonardis, H. Bischof, and A. Pinz, eds.), vol. 3951, pp. 404-417, of *Lecture Notes in Computer Science*, Springer, 2006.

[3] A. Shademan and F. Janabi-Sharifi, "Using scale-invariant feature points in visual servoing," *Optomechatronic Sensors, Actuators, and Control, Proceedings of the SPIE*, vol. 5603, pp.63-70, 2004.

[4] F. Hoffmann, T. Nierobisch, T.Seyffarth and G.Rudolph "Visual servoing with moments of SIFT features," *IEEE Int. Conf. on Systems, Man, and Cybernetics*, pp. 4262-4267.

[5] La Tuan Anh, Jae-Bok Song, "Improvement of Object Recognition for Grasping Task using SURF and Background Subtraction," *Proc. of Int. Conf. on Ubiquitous Robots and Ambient Intelligence*, pp. 545-548, 2009.10.

[6] F. Chaumette, "Potential problems of stability and convergence in image-based and position-based visual servoing," *The Confluence of Vision and Control*, D. Kriegman, G. Hager, A. Morse (eds), LNCIS Series, Springer Verlag, vol.237, pp.66-78, 1998.