Improvement of Object Recognition for Grasping Task using SURF and Background Subtraction

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 robotic applications grasping the daily objects is an essential requirement. In that context object and obstacle detection are used to find the object and to plan an obstacle-free path for the robot in order to manipulate the object. In this paper, an efficient object recognition and obstacle detection methods based on the background subtraction and speeded-up robust features (SURF) algorithms are proposed. Instead of performing tracking the object in full image, we search and match features in the window of attention which contains only the object. Therefore, the tracked interest points are more repeatable and robust to noise. In addition, the background subtraction helps to monitor the workspace and detect the appearance of obstacles. Various experiments show that objects can be grasped safely and stably in the dynamic environment using the proposed method.

Keywords − grasp, SURF, object recognition, background subtraction.

1. Introduction

Grasping an object is a simple task for humans; but not for robots. First, the robot must know the model of an object to distinguish it from a cluttered environment. Second, the robot needs to estimate the path along which it approaches the object without colliding with other obstacles. Third, the robot needs to find the area on the object on which it places its figures to keep the object stable in all grasping and subsequent steps.

Several grasp algorithms using vision sensors have been proposed so far. In [1], using local visual features (based on the 2D contour) and other properties such as form- and force-closure, the method determined the 2D locations for grasp. Another approach is the learning method [2] that used visual features to predict good grasping points for a wide range of objects. Using a synthetic data set for training, the 2D grasp location in each image was predicted. Then, given two or more images of an object taken from different camera views, the 3D position of a grasping point was predicted. Different from previous methods, our work in robot manipulation using a vision sensor focuses on finding and grasping an object in a cluttered environment. Object recognition and obstacle detection are used to find the object and to plan an obstacle-free path for the robot in order to grasp the object.

In this paper, we propose to use the window approach for the SURF algorithm to recognize the object part in the image and construct a background model efficiently. In addition, the background subtraction helps to monitor the workspace and detect the appearance of dynamic obstacles. The information on the desired object and obstacles is used to plan an obstacle-free path from the current position of the robot to the position of the object. Based on the path a robot can move without colliding to static or dynamic obstacles in the given workspace. The effectiveness and robustness of the object grasp task is experimentally validated using a robot arm with a stereo camera attached on its base.

The paper is organized as follows. Section 2 provides a brief description on the SURF and the effect of image resolution in the speed of the algorithm. Section 3 introduces the construction of background model based on the window of attention of SURF. It also explains how the integration of background subtraction and SURF can make a high-speed detection of an object as well as obstacles. In Section 4, the experiment of grasping and putting an object to a desired place using a light weight robot (LWR) manipulator and a stereo camera based on the proposed method is described. The paper concludes with a summary in Section 5.

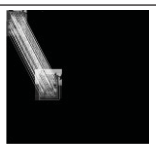
2. Speed-Up Robust Features Algorithm

The speeded-up robust features (SURF) algorithm [3] is an efficient object recognition algorithm with the fast scale-invariant and rotation-invariant detector and descriptor. The speed of the SURF detector improves due to the use of integral images which significantly reduce the number of operations for simple box convolutions, independent of the chosen scale. The descriptor describes the distribution of the intensity content of the neighborhood of interest point in only 64-dimensions vectors. The distribution of first-order Haar wavelet responses, again exploits integral images for speed, reduced the time for feature computation and matching. Furthermore, an indexing step based on the sign of the Laplacian improves not only the robustness of the descriptor, but also the matching speed.

However, with a high image resolution, the time required to process the image by SURF algorithm increase, as shown in Table 1. In case of (a), the object is searched for in a 640x480 image by the SURF detector. The algorithm found 679 interest points and the time required for extracting these points was 110ms. By using a background subtraction algorithm (case (b)) it found 238 interest points in 77ms. If we can extract the region of

interest (ROI) which only contained our object (case (c)) the resolution of image decreased significantly and it took only 17ms to extract 194 interest points. Furthermore, with a smaller set of interest points, the time for the matching process also decreased and it made the SURF algorithm run much farther.

Table 1 Effects of image resolution in SURF detector.

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

3. Background Subtraction

In case of the fixed camera position, background subtraction is the fundamental image process step for visual surveillance applications. In this research, a background subtraction algorithm is used to decrease the processing time of the SURF algorithm and to monitor the dynamic obstacle.

3.1 Background model

The typical subtraction algorithm proceeds by warping each pixel of the background image *I'(x)* to the current image *I(x)* positions and comparing the intensity values. Therefore, we need to define a binary masking function $f(x)$ for construction of the background subtracted image. The masking function $f(x)$ determines the complexity and accuracy of the algorithm. In general, $f(x)$ is a Boolean function which takes a value of 1 for all locations x, which belong to the set of foreground pixels:

$$f(x) = \begin{cases} 1, \text{if} \left| I(x) - I'(x) \right| > \varepsilon \\ 0, \text{otherwise} \end{cases} \quad (1)$$

We efficiently invest our resources in cleaning up the false positive pixels that result of unpredictable environments. The cleanup takes the form of image processing operations such as 'erode' and 'dilate' that eliminate the false pixels. The subtracted image is formed by applying the mask to the current camera view:

$$S(x) = f(x)I(x) \quad (2)$$

Figure 1 illustrates the background subtraction scheme and a practical example for the indoor environment. To perform background subtraction, the model of the background should be found. Then, the model is compared to the current image to subtract the background part. The objects left after subtraction are probably new foreground objects. When we use the subtraction method in a dynamic environment, the dynamic obstacles can be detected by the difference image between two successive background images.

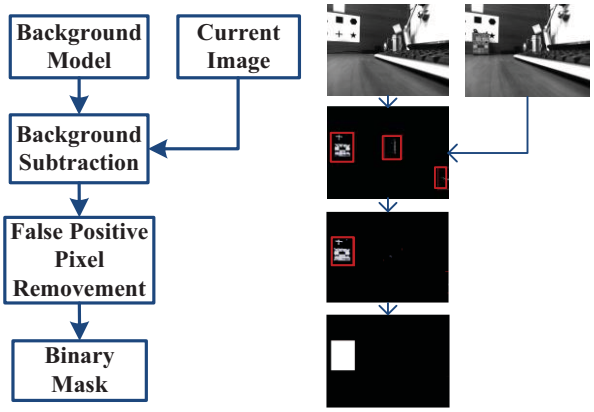


Fig. 1. Scheme of the background subtraction.

3.2 Background subtraction using SURF detection

In our application, together with speedy object recognition we need to monitor the background to find out the obstacles. Figure 2 summarizes the scheme of the background model construction using the SURF detector. 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 images, only the interest points in this ROI are extracted and matched to find the object. If no object is detected, the initial process is repeated, so that the full image is searched to find the object. Then, the background is obtained by subtracting the object part from the current image.

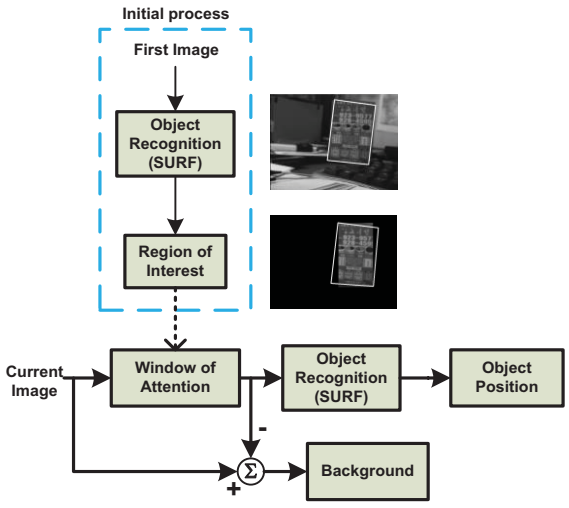


Fig. 2. Scheme of background model construction.

In detail, from the first image of the sequence image captured by the camera, the object is recognized using the SURF algorithm by matching the interest points between the current and the reference view of the object. The fundamental matrix can compute from the set of perfectly matched interest points. However, in practice, not all of the interest points are distinctive, which may lead to false matches. Therefore, once the closest match of interest point in an image to the point of another image is found, it should be checked to see whether it is an outlier or inlier. We used the nearest Euclidean distance of the 64-vector of the interest point description and applied a threshold to decide if the interest point is an inlier. Another approach [4] to find the interest point matching is to look at the second-closest distance to the candidate points. If the closest distance was significantly smaller than the second-closest distance, we have a strong candidate and the match is most probably correct. Finally, the random sample consensus (RANSAC) algorithm is used to eliminate the rest of mismatches and estimate the fundamental matrix. The window which contains the object is obtained as illustrated in Fig. 3. The four 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

$$\begin{cases} X = c[\min x], \\ Y = c[\min y], \\ w = c[\max x] - c[\min x], \\ h = c[\max y] - c[\min y]. \end{cases} \tag{3}$$

where

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

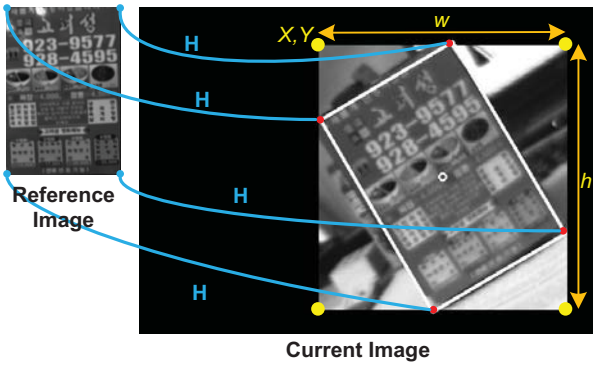


Fig. 3. Construction of window of attention.

Since the contours of the object are not used in the method, if the object is partially occluded by other objects or scene, the window which contains the object can still be constructed. In the next sequence image, this window is used as the ROI and the interest points are only extracted from this window. Because the ROI is only the small part of the image, the speed of image processing improves significantly. Then, the background is obtained by subtracting the window containing object from the current image.

In a static environment, the poses of static obstacles in the workspace are recognized by extracting their color and edge information from the background image. In a dynamic environment, the difference image between two successive background images is used to detect dynamic obstacles. Then, the distances from the robot to the object and to the obstacles are computed from the stereo camera geometry. The image processing system transmits the state information of a static or dynamic environment to the path planning agent [5].

In object grasping application, when the dynamic obstacles around the object are found by the difference background image, the object itself also can be partially occluded by an obstacle. In case the dynamic obstacles appear in object position, i.e. object is partially occluded, the grasping position has to change to avoid collision with the obstacle. A method to select the grasping position in the occluded object scenario is described in Fig. 4. 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 is decreased. 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 is the position in the object which coincides with the center of the end-effector.

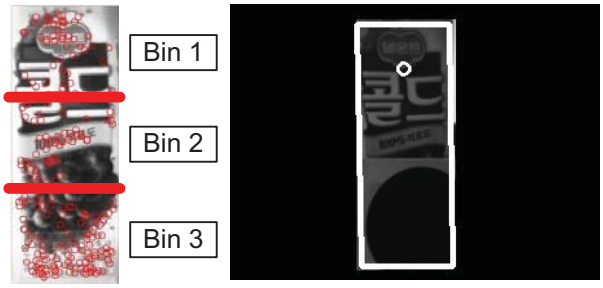


Fig. 4. Selection of grasping position.

4. Experiments

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 a camera to find, recognize and pick up daily objects. In this experiment section, the task of grasping and putting an object to a desired place using a LWR robot and a stereo camera attached on its base is described.

When the mobile base approaches the object position, it stops and its manipulator starts to operate. From the image taken by the camera, the object is detected and the

background image is extracted. The change of environment was closely monitored by the background subtracted images. Since we need to recognize small objects (e.g., bottle, can …) to grasp in a cluttered environment, the number of features of the object is important. Therefore, we used a 640x480 pixel image instead of the 320x240 image. However, the higher image resolution usually requires an increase in processing time as described in Section 2. By using the window attention for the SURF algorithm, the robot can recognize the object in the presence of partial occlusion and the interest points detection and matching processes run at a high-speed rate of approximately 20Hz (for a camera resolution of 640x480 pixels on a Core 2 Duo running at 2.53 GHz). In addition, when a dynamic obstacle appears in the workspace, it will be detected by the difference image between two successive background images. Then, the suitable grasping points are selected.

Figure 5 describes the detected object inside the red rectangle and a new obstacle on the left after the background subtraction and window approach of SURF processes.

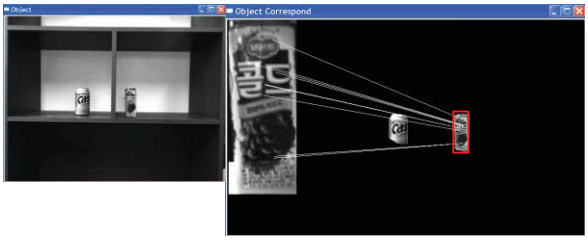


Fig. 5. Result of background subtraction and object recognition.

When the information on the desired object and obstacles are given in the workspace, path planning for the robot manipulator is used to generate and update an optimized path from the current position to the position of the object. Based on the path, the robot can avoid colliding with static or dynamic obstacles in the workspace.

Fig. 6. Object grasping task carried out by a LWR manipulator.

Figure 6 (left to right, top to bottom) shows an object grasping task carried out by the LWR manipulator and Bumblebee stereo camera. The camera recognized the object and detected all obstacles in the workspace together with their distances to the camera. Specifically, the image from the right camera was used to carry out the background subtraction and the SURF algorithms, and then the distance from the object plane to the camera was estimated using the knowledge of the stereo camera geometry. The positions of all objects in the camera frame were converted to the positions in the robot base frame by the transformation matrix. Based on these positions, an obstacle-free path was generated and the end-effector of the robot approached the object without colliding with any obstacles. Finally, the robot grasped and moved the object to a chosen target position.

5. Conclusions

This paper proposed an efficient object recognition scheme based on the window approach of the SURF and background subtraction. The experiments showed that the object grasping process using the proposed scheme can work safely and stably in dynamic environments. From various experiments, the following conclusions are drawn:

1. The method is generic as it does not depend on the object model but automatically extracts SURF interest points from object images. The window approach not only increases the tracking speed significantly but also constructs the background image for monitoring all obstacles in the workspace.
2. The poses of static and dynamic obstacles in the workspace are detected effectively by extracting the information from the background image and the difference image between two successive background images.
3. The choice of stable object grasp position enables the manipulator to grasp even when the object is partially occluded.

Acknowledgement

This research was supported by Korea Foundation grant (R11-2007-028-01002-0) and by the Personal Robot Development Project funded by the Ministry of Knowledge Economy of Korea.

References

[1] P.J. Sanz, A. Requena, J.M. Inesta, and A.P. Del Pobil, “Grasping the not-so-obvious: vision-based object handling for industrial applications,” *IEEE Rob. & Auto. Mag.*, vol.12, no.4, pp.44-52, 2005.

[2] A. Saxena, J. Driemeyer, and A. Y. Ng, “Robotic grasping of novel objects using vision,” *IJRR*, vol. 27, no. 2, pp. 157-173, 2008.

[3] H. Bay, A. Ess, T.Tyutelaars, and L.V. Gool, “Speed-Up Robust Features (SURF),” *Computer Vision and Image Understanding (CVIU)*, vol. 110, no. 3, pp. 346-359, 2008.

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

[5] J.J. Park, J.H. Kim, and J.B. Song, “Path Planning for a Robot Manipulator based on Probabilistic Roadmap and Reinforcement Learning,” *International Journal of Control, Automation, and System*, vol. 5, no. 6, pp. 674-680, 2007.