Object Recognition and Segmentation Using SIFT and Graph Cuts

Akira Suga† Keita Fukuda†† Tetsuya Takiguchi††† Yasuo Ariki†††
† *Department of Computer Science and Systems Engineering, Kobe University*
†† *Graduate School of Engineering, Kobe University*
††† *Organization of Advanced Science and Technology, Kobe University*
{*akira1234,fukuda*}*@me.cs.scitec.kobe-u.ac.jp* {*takigu,ariki*}*@kobe-u.ac.jp*

Abstract

In this paper, we propose a method of object recognition and segmentation using Scale-Invariant Feature Transform (SIFT) and Graph Cuts. SIFT feature is invariant for rotations, scale changes, and illumination changes and it is often used for object recognition. However, in previous object recognition work using SIFT, the object region is simply presumed by the affine-transformation and the accurate object region was not segmented. On the other hand, Graph Cuts is proposed as a segmentation method of a detail object region. But it was necessary to give seeds manually. By combing SIFT and Graph Cuts, in our method, the existence of objects is recognized first by vote processing of SIFT keypoints. After that, the object region is cut out by Graph Cuts using SIFT keypoints as seeds. Thanks to this combination, both recognition and segmentation are performed automatically under cluttered backgrounds including occlusion.

1. Introduction

Object recognition is one of the important research fields to realize cognitive ability of the computers, and is expected to be applied to Robot eyes or head mounted display. Recently, manual retrieval and classification of the image become difficult as volume of data becomes huge. Therefore, the importance of the object recognition by the computer has increased.

The problem in the object recognition is to deal with the rotations of the object, scale changes, and illumination changes. Moreover, there is the problem of occlusion, and these make the object recognition difficult. SIFT was proposed by Lowe as a robust feature for these problems [1], and the object recognition method which uses SIFT is also proposed [2]. But it could not segment the object region.

About segmentation, Graph Cuts that solves the segmentation problem as an energy minimization problem was proposed. In Snakes [3] or Level Set Method [4], it calculates the local minimum solution of the energy function at a boundary. On the other hand, since Graph Cuts defines the energy function in both boundaries and regions, it can calculate global minimum solution and can perform better segmentation. Boykov proposed Interactive Graph Cuts which performs energy minimization using Min cut/Max flow algorithm [5]. However, it was necessary to give seeds manually before starting the segmentation.

In order to solve these problems and to carry out both recognition and segmentation automatically, we proposed to use SIFT keypoints as seeds of Graph Cuts. Accordingly, object recognition and segmentation can be performed without giving seeds manually.

2. SIFT and Graph Cuts

2.1. SIFT

2.1.1. Keypoint detection. Keypoints are detected by DoG image $D(x, y, \sigma)$ which is the difference of smoothed images $L(x, y, \sigma)$. $L(x, y, \sigma)$ is obtained from the convolution of variable scale Gaussian with the input image $I(x, y)$.

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \quad (1)$$

$$L(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \exp\left(- \frac{x^2 + y^2}{2\sigma^2} \right) * I(x, y) \quad (2)$$

It is performed between different scales σ, and number of DoG images are obtained. Local extremes are detected from these DoG images by comparing 26 neighborhood of a pixel within a set of three DoG images, and if it is an extremum, the pixel is detected as the keypoint.

978-1-4244-2175-6/08/$25.00 ©2008 IEEE

2.1.2. Keypoint descriptor. The gradient magnitude $m(x, y)$ and orientation $\theta(x, y)$ at each pixel of the smoothed image that the keypoints are detected are calculated using the following expressions:

$$m(x,y) = \sqrt{f_x(x,y)^2 + f_y(x,y)^2} \tag{3}$$

$$\theta(x,y) = \tan^{-1} \frac{f_y(x,y)}{f_x(x,y)} \tag{4}$$

$$\begin{cases} f_x(x,y) = L(x+1,y) - L(x-1,y) \\ f_y(x,y) = L(x,y+1) - L(x,y-1) \end{cases} \tag{5}$$

The weighted histogram of 36 directions is made by gradient magnitude and orientation in the region around the keypoint, and the peak that is 80% or more of the maximum value of the histogram is assumed to be the orientation of the keypoint. After rotating the region around the keypoint to the orientation, the descriptor is created. Then, the region is divided into the blocks of 4×4, and the histogram of eight directions is made at each block. Thus, we can obtain 4×4×8=128 element feature vector for each keypoint.

2.2. Graph Cuts

2.2.1. Image segmentation by energy minimization. First of all, the pixel of image P is defined as $p \in P$ and the neighborhood of p is $q \in N$. And the label is defined as $A = (A_1, A_2, ..., A_p, ..., A_{|P|})$, where $A_p \in \{"obj", "bkg"\}$. The energy function used in Graph Cuts is shown as follows:

$$E(A) = \lambda \cdot R(A) + B(A) \tag{6}$$

The coefficient $\lambda(\geq 0)$ specifies the relative importance of the region properties $R(A)$ to the boundary properties $B(A)$.

$$\begin{cases} R(A) = \sum_{p \in P} R_p(A_p) \\ B(A) = \sum_{\{p,q\} \in N} B_{\{p,q\}} \cdot \delta(A_p, A_q) \end{cases} \tag{7}$$

Here, $\delta(A_p, A_q)$ is 1 if $A_p \neq A_q$, otherwise 0. The term $R(\cdot)$ may reflect how the intensity of pixel p fits into a known intensity model of object and background. $B_{\{p,q\}}$ shows relations with the pixel of the neighborhood, and it becomes large value if the brightness value of p resembles q. Segmentation is performed by obtaining the label A that makes an energy function $E(A)$ the minimum.

2.2.2. Outline of Graph Cuts. The general approach to construct a graph from an image is shown in Fig. 1. Each pixel in the image is viewed as a node in graph. There are two additional nodes: an "object" terminal (a source S) and a "background" terminal (a sink T). Edges that connects each pixel are called n-link, and another edges that connects pixel and terminal are called t-link.

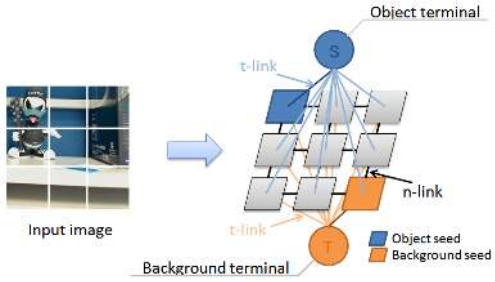


Figure 1. Creating graph

Then, costs are given to all edges. The costs of each edge are shown in Table 1. O and B represent "object" and "background" seeds respectively. $R_p("obj")$, $R_p("bkg")$, $B_{\{p,q\}}$, and K are calculated by following expressions:

$$\begin{cases} R_p("obj") = -lnPr(I_p \mid O) \\ R_p("bkg") = -lnPr(I_p \mid B) \end{cases} \tag{8}$$

$$B_{\{p,q\}} \propto exp\left(-\frac{(I_p - I_q)^2}{2\sigma^2}\right) \cdot \frac{1}{dist(p,q)} \tag{9}$$

$$K = 1 + \max_{p \in P} \sum_{q:\{p,q\} \in N} B_{\{p,q\}} \tag{10}$$

where, I_p is brightness value at pixel p. The boundary between object and background is founded by applying Min cut/ Max flow algorithm to the graph [6].

Table 1. Edge cost

edge		cost	for
n-link	$\{p,q\}$	$B_{\{p,q\}}$	$\{p,q\} \in N$
t-link	$\{p,S\}$	$\lambda \cdot R_p("bkg")$	$p \in P, p \notin O \cup B$
		K	$p \in O$
		0	$p \in B$
	$\{p,T\}$	$\lambda \cdot R_p("obj")$	$p \in P, p \notin O \cup B$
		0	$p \in O$
		K	$p \in B$

3. Recognition and segmentation

3.1. Recognition

This section shows how to recognize objects. It consists of three phases (matching process, voting process,

clustering process), and we describe about each process in turn.

3.1.1. Matching process. First of all, the SIFT database is created by extracting SIFT keypoints from model image which is taken from different angles for each object. Second, SIFT keypoints are extracted from the input image. Then, matching SIFT keypoints between the input image and model images is performed. Let s^m and w^n define mth keypoint in a certain model image and nth keypoint in the input image respectively. The keypoint n' which makes Euclid distance minimum is obtained by the following expressions.

$$n' = \arg\min_n \sqrt{\sum_i^{128} (s_i^m - w_i^n)^2} \tag{11}$$

The corresponding point of the object is obtained by performing this calculation for all keypoints of all models.

3.1.2. Voting process. Beforehand, we set the object's center point in model image. For each keypoint, a position vector (from the center point) $(\Delta x, \Delta y)$ is calculated as shown in Fig. 2.

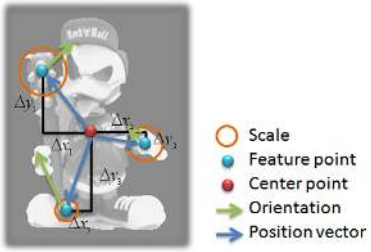


Figure 2. Calculation of position vector

Then, using the position vector, the object's center candidate point (X, Y) in input image is calculated and voted for all keypoints by following expression.

$$\begin{cases} X = x_{in} + \frac{\sigma_{in}}{\sigma_{mdl}} \times \sqrt{\Delta x^2 + \Delta y^2} \times \cos(\theta + \theta_{mdl} - \theta_{in}) \\ Y = y_{in} + \frac{\sigma_{in}}{\sigma_{mdl}} \times \sqrt{\Delta x^2 + \Delta y^2} \times \sin(\theta + \theta_{mdl} - \theta_{in}) \end{cases} \tag{12}$$

where, $\theta = \arctan(\frac{\Delta y}{\Delta x})$, and $\sigma_{\{in,mdl\}}$ is the scale of the keypoint in the input image and the model image, $\theta_{\{in,mdl\}}$ is the orientation. When the model object exists in the input image, center candidate points gather at the same location [7]. Then, the voted points are grouped into some clusters.

3.1.3. Clustering process. It turns out that the object exists in the location in which votes have gathered densely. We use Ward method for clustering. In our method, we define a threshold experimentally and stop processing when distance between clusters reaches the threshold. And we obtain the cluster at that time as a clustering result. Then, we count the number of votes in each cluster, and when it obtains more than a voting threshold, the system recognizes the existence of the object.

3.2. Segmentation

Seeds are created automatically. The matched SIFT keypoints are used as object seeds. But there are a lot of mismatch when the matched keypoints are simply used. Then, we make use of an advantage of earlier recognition, The keypoints which vote the recognized cluster (that have more than a voting threshold) are used as a object seeds. Accordingly, objects seeds are obtained with high accuracy (The red points in Fig. 3). Background seeds are the outside region of the object obtained by affine-transformation of the model image by using object seeds. (The blue region in Fig. 3).

(a) Input image (b) Seeds

Figure 3. Creating seeds automatically

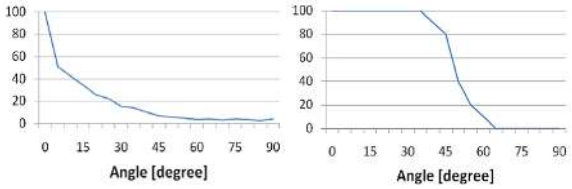
4. Experiment

4.1. Experimental condition

Recognition and segmentation experiments were carried out with 20 object models which are seen well in daily life and 100 test images. Model images were taken from the angles of every 45° per object i.e. 8 images per object. This angle was determined based on the experimental result using Columbia University Image Library (COIL-100) [8]. Fig. 4 shows the average matching rate and recognition rate as a function of viewing angles. As the viewing angle changes, although the matching rate falls rapidly, the recognition rate is stably high to 45°. Based on this experiment, the viewing angle was determined. Test images were taken from various angles with various scales. The number of the test images is five per model object. This test set includes some images that no model object appears in.

The recognition accuracy was evaluated by recall and precision. The segmentation accuracy was evaluated by error rate to correct mask data as shown in Eq. (13).

$$Err[\%] = \left(\frac{miss\ detected\ pixels\ in\ the\ object}{all\ pixels} + \frac{miss\ detected\ pixels\ in\ the\ background}{all\ pixels} \right) \times 100 \quad (13)$$



(a) Matching rate (b) Recognition rate

Figure 4. Matching and recognition rate to angles

4.2. Experimental result

The results of the recognition accuracy and the segmentation accuracy are shown in Table 2 and Table 3. Success examples of the segmentation are shown in Fig. 5.

Table 2. Recognition accuracy

Recall	Precision
0.81	1.00

Table 3. Segmentation accuracy

Object region error[%]	Background region error[%]	Total error[%]
3.73	6.21	9.94

5. Conclusion

This paper proposes the method of both object recognition and segmentation. In our approach, the object is previously recognized by voting process, and segmentation is carried out successively using SIFT keypoints as seeds. Accordingly, we succeeded in cutting out detailed object region even if it is the technique that used local features. Moreover, since the correspondence points are taken by SIFT, it is not necessary to give some seeds manually in Graph Cuts. However, if there are few keypoints, the accuracy of recognition and segmentation will fall down. And the computing time increases when the number of models increases. We will consider about these problems in the future.

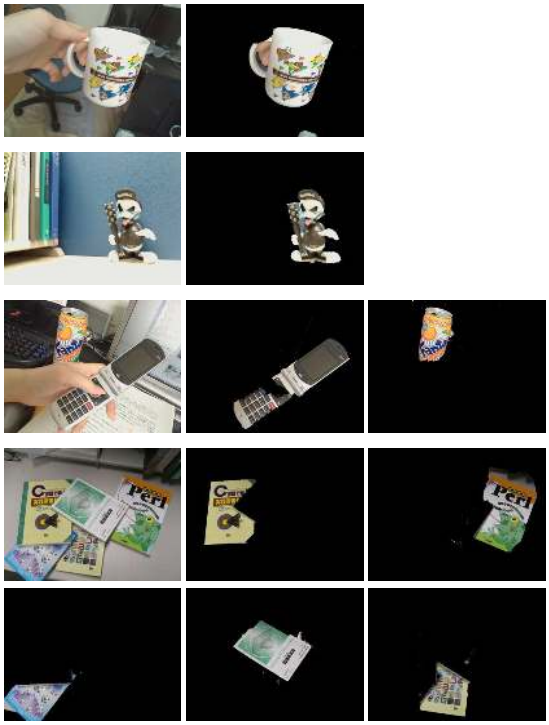


Figure 5. Segmentation result

References

[1] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Journal of Computer Vision*, 60, 2, pp.91-110, 2004.

[2] D. G. Lowe. Object recognition from local scale-invariant features. *International Conference on Computer Vision*, Corfu, Greece , pp. 1150-1157, 1999.

[3] M.Kass, A. Witkin and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, Vol. 1, No. 4, pp. 321-331, 1988.

[4] James A. Sethian. Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry Fluid Mechanics. *Computer Vision, and Materials Science. Cambridge University Press*, 1999.

[5] Y. Boykov, M.P.Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. *IEEE International Conference on Computer Vision and Pattern Recognition*, volume 2, pp. 731-738, 2004.

[6] Y. Boykov and V. Kolmogorov. An Experimenal Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimizatio in Computer Vision. *In IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2004.

[7] M.Takagi and H. Fujiyoshi. Road Sign Recognition using SIFT feature. *Symposium on Sensing via Image Information*, LD2-06, 2007.

[8] Columbia Object Image Library (COIL-100). http://www1.cs.columbia.edu/CAVE//software/softlib/coil-100.php