H-044

Object segmentation based on multi-resolution texture analysis

Lingkun LUO and Hiroshi SAKO

Graduate School of Computer and Information Sciences
Hosei University

1. Introduction

Image recognition is one of important technologies in recent applications including image retrieval systems, surveillance systems, medical systems, etc. In the recognition procedure, the object is segmented from the input image and suitable features of the object are detected to classify the object. In the object segmentation, there are two approaches; the detection of similar grey/color area and of similar texture area. The former is based on the thresholding of the image [1] or the detection of object contours [2], while the latter is on the direct analysis of object textures [3]. These methods are sometimes followed by the processing such as religion-growing.

The method proposed in this paper belongs to the latter approach which can be widely applied to objects having their own textures.

Known difficulties in the texture analysis is how to realize robustness against a variety of 1) texture kinds, 2) texture sizes, and 3) backgrounds to be discriminated.

In this paper, we firstly proposed a pre-/post- processing to cope with the issue 2) and examined two simple texture analysis methods to cope with 1) and 3), and then experiments show the effectiveness of the proposed method.

2. Method

Figure 1 shows the outline of the proposed method which is composed of three parts; a pre-processing, a texture analysis, and a post processing.

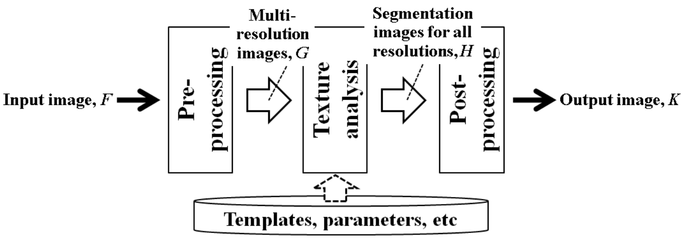


Figure 1. Outline of the proposed method

2.1 Pre-processing

The purpose of the pre-processing is to obtain multi-resolution images from an input image. This can be realized by a simple averaging operation formulated by;

$$G(x, y; I, J, s) = (1/s^2) \sum_{-[s/2] <= i, j <= [s/2]} F(x+i+s*I, y+j+s*J),$$

where G is the output multi-resolution image, F is the input image, x and y are coordinates of the images, s expresses the degree of resolution (the size of area to average), (I, J) takes (-1 or 0 or 1, -1 or 0 or 1) which covers nine cases, and [] denotes Gauss' symbol. The value s depends on the application, but in the following experiments, it is set by 1, 3, and 7. This pre-processing has a favorable effect in order to adjust the size of texture for the next texture analysis part. Note that the output G has nine values corresponding to all combinations of (I, J). They can be calculated at each (x, y).

2.2 Texture analysis

The purpose of the texture analysis is to detect candidate area of the object having similar textures. There are several methods to realize this purpose. The major difference between the methods exists in how to express the texture itself or how to express the difference between the texture and the background.

(1) Expression of textures: Two simple methods exist; *k*-Nearest Neighbor (*k*-NN) and Liner Regression (LR). In *k*-NN, template images with tagged information of the typical texture area (tagged by "1") and the background (tagged by "0") are prepared. In LR, the texture (sized by 3x3, for example) is expressed by a liner equation like $a0*p0+a1*p1+....+a8*p1=1$ (in the texture area) and $a0*p0+a1*p1+....+a8*p1=0$ (in the background area), where $p0$ … $p8$ are neighboring-pixel values, and $a0$ … $a8$ are constant values determined in advance so as to minimize the discrimination error in the template images.

(2) Expression of the difference between the texture and the background: Normally, the discriminant function is determined by training typical images as referring their ground truths (GTs) composed of the texture area (tagged by "1") and the background (tagged by "0"). One of the methods to determine the function is Neural Networks (NN). The function can be expressed in the form of networks using sigmoid functions and weights W_{ij}. Back propagation can tune the weights so as to minimize the discrimination error in the training images.

The above methods are pixel-based approach but there is another approach based on features. The features include Tamura's features [4] and the one based on co-occurrence matrix. In this paper, we examine two simple pixel-based approaches; *1*-NN and NN. In these approaches, the selection of templates or training images is very important to cope with the issue 1) and 3), that will be discussed in Chapter 3. For each resolution s, the nine values of G at (x, y) are input to the texture analysis part.

In the *1*-NN, these values are compared with the corresponding neighboring-pixels' values in the template images, and select the tag (1/0) information at the position with best matching (of a minimum distance). In the NN, the values at (x, y) are input to the trained NN and the output determines if the pixel at (x, y) belongs to the texture (1) or the background (0). The structure of the NN, here, is based on 9 * 15 * 1 neurons.

Hereinafter, let the output binary image for each resolution *s* from the texture analysis part be *H(x, y; s)*.

2.3 Post-processing

The purpose of the post-processing is to produce a final segmentation image of the object. Since we have no information about the texture size then we can't select a suitable *H(x, y; s)* logically. Therefore, a simple solution is to combine all of *H(x, y; s)* by an OR-operation regarding resolution *s*. This produces the output image *K(x, y)*.

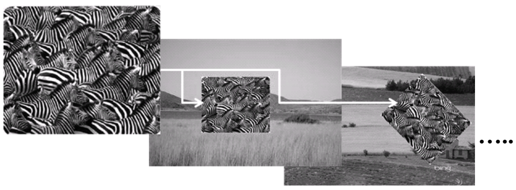
3. Experiments

In this chapter, we show some segmentation results in order to confirm the effectiveness and the robustness of the proposed method.

In the experiments, we use animal images such as zebras and cheetahs with various backgrounds. These images are suitable for evaluating the method at an early R&D stage because the objects have their own texture. The data is retrieved from public websites [5] by a search engine. The experimental environment is; 1) CPU: Intel/Core/i7-970, 3.23GHz, 12MB-Cashe, 2) OS: Windows7 Pro, SP1, 64 bits, 3) Software: VC++2008, OpenCV 2.1, etc.

3.1 Templates and images for training

An image which contains various zebra textures, shown in Figure 2(a), is selected. Nine templates (training images) are created by shifting, rotating, and scaling the texture area in the image (Figure 2(b)), and corresponding GTs are created as well.



(a) Original zebra image (b) Created images for training

Figure 2. Templates (training images)

3.2 Segmentation accuracy

The segmentation accuracy of texture area, R_t, and that of background area, R_b, are equally evaluated by comparing the result with the GT. The total accuracy *R* is calculated by $0.5 * R_t + 0.5 * R_b$, where R_t is the ratio of the matched number of texture pixels (tagged by "1") against the total number of texture pixels of the GT, and R_b is the ratio of the matched number of background (tagged by "0") pixels against the total number of background pixels of the GT.

3.3 Results by the *1*-NN and the NN

Table 1 shows the *1*-NN and the NN segmentation results and the accuracies of seven test images. The average accuracy by the *1*-NN and the NN are 77.2% and 76.6%, respectively.

3.4 Comparison

The *1*-NN and NN give us the similar accuracy in the experiments. The computational complexity of the *1*-NN and the NN is proportional to (template image size * test image size) and (constant * test image size), respectively. Therefore, NN usually has some advantage in computational time.

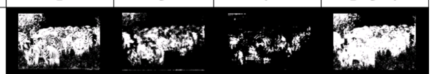
Table 1. Segmentation by *1*-NN and NN

Test no.	1	2	3	4	5	6	7
Input, *F*							
1-NN Output, *K*							
NN Output, *K*							
1-NN Accuracy *R* [%]	86.9	71.3	<u>78.7</u>	82.9	<u>71.7</u>	<u>70.4</u>	78.5
NN Accuracy *R* [%]	<u>88.6</u>	<u>72.7</u>	76.5	<u>86.2</u>	68.2	62.7	<u>81.2</u>

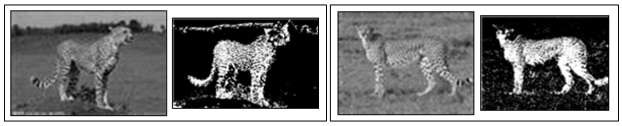
3.5 Effect of multi-resolution

Table 2 shows the accuracy in single and multi-resolution. This proves the significant effect in the multi-resolution case.

Table 2. Accuracy in single and multi-resolution

Resolution, *s*	Single			Multi
	1	3	7	1+3+7
NN Output, *K* (Test no. 4)				
NN Accuracy *R* [%]	81.5	73.7	58.9	<u>86.2</u>

3.6 Application to other objects



(a) Cheetah 1 (*R*: 79.5%) (b) Cheetah 2 (*R*: 81.8%)

Figure 3. Segmentation of other objects

4. Conclusions

This paper proposes the object segmentation method which consists of the pre-processing to produce multi-resolution images and the texture analysis using *1*-Nearet Neighbor and Neural Networks. This structure could cope with the variety of the texture in some extent. The experiments show that the method achieved about 77% segmentation accuracy on average. The future study includes an improvement of the method by using some feature based approach.

References

[1] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics (SMC)*, 9(1), 62–66, 1979.

[2] J. Malik, S. Belongie, T. Leung, and J. Shi, "Contour and Texture Analysis for Image Segmentation," *International Journal of Computer Vision*, 43(1), 7–27, 2001.

[3] G. McLean, "Vector quantization for texture classification," *IEEE Transactions on SMC*, 23(3), 637–649. 1993.

[4] H. Tamura, S. Mori, and T. Yamawaki, "Textural features corresponding to visual perception," *IEEE Transaction on SMC*, 8(6), 460–472, 1978.

[5] http://kikulacho.com/2010/12/zebra-maasai-mara/, etc