Bottom-Up Segmentation Based Robust Shape Matching in the Presence of Clutter and Occlusion

Hanbyul Joo, Yekeun Jeong, In-So Kweon
Dept. of Electrical Engineering and Computer Science
KAIST, Deajeon, Korea
hbjoo@rcv.kaist.ac.kr, ykjeong@rcv.kaist.ac.kr, iskweon@kaist.ac.kr

Abstract

In this paper, we present a robust shape matching approach based on bottom-up segmentation. We show how over-segmentation results can be used to overcome both ambiguity of contour matching and occlusion. To measure the shape difference between a template and the object in the input, we use oriented chamfer matching. However, in contrast to previous work, we eliminate the affection of the background clutters before calculating the shape differences using over-segmentation results. By this method, we can increase the matching cost interval between true matching and false matching, which gives reliable results. Finally, our experiments also demonstrate that our method is robust despite the presence of occlusion.

1. Introduction

Shape is an useful information to recognize the objects. Human visual system can often recognize object using shape alone. This ability makes it possible to recognize objects which have similar shapes in the same categories even if they have totally different textures or colors. Shape information have been used in computer vision area for object recognition and figure-ground segmentation [2, 4, 5, 6]. And, chamfer distance is often used to measure shape differences between the templates and the objects of input images.

Chamfer distance provides reasonable measurement to compare two shapes, tolerant to considerable misalignment of position and scale. However, when we consider the contour information alone in cluttered images, there could be mismatches because of some combinations of small unrelated contour fragments which is similar to the template shapes if we piece it together. To overcome this problem, we use bottom-up segmentation to eliminate the affection of background clutter. Before calculating a shape difference, our algorithm selects segment candidates which compose true object. And then, our algorithm calculates chamfer distance between a template and the composed shapes by selected segment candidates. By this approach, our algorithm can solve the intrinsic problem of previous chamfer matching.

This paper is structured as follows. In Section 2, we briefly review about chamfer matching and introduce some intrinsic problems of chamfer matching. And in section 3, we describe in detail how we use bottom-up segmentation for robust shape matching. The experimental result are explained in section 4, and a conclusions are given in section 5.

2. Contour Features

In this section, we briefly review Chamfer Matching algorithm to measure the similarity of two shapes. And we will describe intrinsic problems of chamfer matching in the cluttered scenes.

2.1. Chamfer Matching

Chamfer matching [1] is a measurement to compare two shapes in the images. This measurement is based on the assumption that when we put a template exactly right position in the image, the correspondent pixel pair between template and edge map is the closest one. When we think two sets of edge pixels, the edge pixels in the input edge map , $E = \{e\}_{i=1}^{n}$, and the pixels of template , $T = \{t\}_{j=1}^{m}$, Chamfer distance is defined as a function of some transformation A:

$$d_{chamfer}^{(T,E)}(A) = \frac{1}{N_t} \sum_{t_j \in T} \min_{e_i \in E} \parallel A(t) - e \parallel \quad (1)$$

where A is some transformations such as rotation, translation and scaling. N_t is the number of edge pixels in

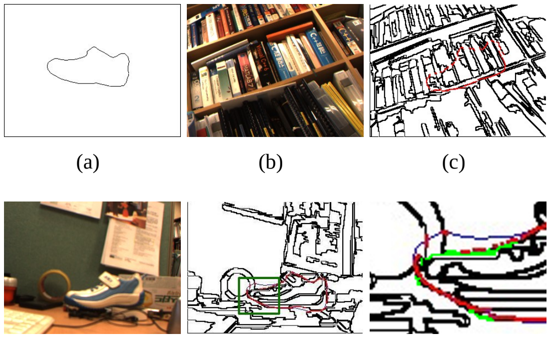
T. This measurement means the average distance between edge pixels of template and their correspondent edge pixels in the input image. So if there is a similar shape to template, chamfer distance may have the lowest cost with the transformation which put the template in that position. The distance between a template and an edge map can be efficiently computed by Distance Transform.

2.2. Problems of Chamfer Matching

Chamfer measurement works correctly and tolerates considerable scale changes and deformations if there are few background clutters. However, with complex background clutters, as our experiments, it might fail to find the true shape because some densely cluttered regions has low chamfer distance regardless of template shape.

Oriented chamfer matchings overcomes this problem by giving an explicit gradient difference penalty in the cost function [5]. In our experiments, we use similar measurement with [5] using gradient information.

However, this improved chamfer matching still have problems if the a correct shape of object in the image is slightly different from template or there are partial occlusions. Fig. 1 shows these problems. After extracting edges from the original image Fig. 1 (b), we tried to find the shape which is most similar to template Fig. 1 (a) even if there is no actual object in the image. The red colored pixels in Fig. 1 (c) are pixels which are closest to each template pixel with the lowest cost. Notice that the combination of small edge fragments is very similar to template shape although there is no actual object there. So, the chamfer distance in this position could be small, and this problem might cause mismatching. Fig. 1(d-f) shows another problem. Fig. 1(d) is an original image, and Fig. 1(e) is a matching result with slightly different alignment. And Fig. 1(f) shows a magnified image of green rectangular region in Fig. 1(e). The blue line is template edge and red line are the closest pixels. We colored actual contour of a shoe as green for visibility. And we can see that there is some red pixels(closest to template) in background clutter edge, and this edges gave large cost because their orientation is totally different. Because chamfer distance only considers the edges in shortest distance, background clutters is often included for shape difference calculation. This could be the reason increasing the cost of true matching, and also often causes mismatching.



(a) (b) (c)

(d) (e) (f)

Figure 1. (a) template (b) original image without any object (c) similar shape with template (d) ordinal image (e)matching result (f) magnified image

3. Segmentation based Contour matching

When we think just edge information in the image, there could be many ambiguous situation like Fig. 1 (c). However, this ambiguity can be eliminated if we use the result of over-segmentation. The red lines in Fig. 1 (c) are the combination of small edge fragments extracted from unrelated part such as a monitor, a chair, and a desk. If we can know that this edge fragments can not compose one object, then we can eliminate this ambiguity. Because over-segmentation algorithm merges homogeneous region into a segment, we can think that the edges included in a segment as a group. And considering the position and size of the segment, we can determine whether the group of edges included in a segment could be a part of template object or not. Using this idea, we can eliminate the edges originated from background clutters, and we can calculate reliable shape measurement using only the edges composing true object.

In this section, we describe in detail how we use over-segmentation for robust shape matching and occluded object detection.

3.1 Edge Detection using Over-segmentation

At first, we divide an image into homogeneous regions using over-segmentation algorithm. Any over-segmentation algorithm can be used, but we use the segmentation method of Felzenszwalb and Huttenlocher [3] because it generates reasonable results in the extremely cluttered images.

And, we extract edge maps from over-segmented images. Because the pixels included in a segment have similar intensity, the boundaries of each segment can be considered to have large gradient change. So, we can treat segment boundaries as edges. This edge extraction is necessary to use over-segmentation result for shape matching. So, every edge maps in this paper are extracted by this method.

3.2 Selecting Inner Segments

Before computing a shape difference, we have to eliminate the edges originated from background clutters. For this purpose, our algorithm selects segments included in the shape of template to eliminate the segments which is impossible to be a part of object. If we assume that over-segmentation does not merge object region with background, and if we assume that the template is located exactly right position in the image, the shape made by combination of selected segments would be most similar to template.

The most accurate way to find the segment included inside the template is checking every pixel of segment whether it is included in the template or not. However, to reduce time, we just check the center of mass of segment, and randomly selected K pixels. Additionally, we give some margin along with the template boundary to allow some scale changes.

Fig. 2 shows the result. Yellow regions are the combination of segment included in the template in the position. The blue line is actual template contour, and the green line is the extended template edge to tolerate slightly bigger scale.

(a) (b)

Figure 2. (a) original image (b) selected inner segments

3.3 Segmentation Based Shape Matching

The main idea of our approach is comparing template contour with only the true object boundaries without any affection of background clutter. Because the edges originated from background clutter is excluded in previous step, we can achieve our purpose by comparing template contour with the outline of selected segment region (yellow region in 2 (b)) To compare the template contour with the outline of selected segment region, our algorithm extracts outline from the region. This is carried out by follow steps.

1. After extracting edges from an over-segmentation result, visit every edge pixel and save its adjacent segment indexes.
2. For each pose of template, find inner segment using the the algorithm described in Section 3.2.
3. Visit every edge pixel of inner segments, and choose the edge pixel which has non-inner segment as adjacent segment. These pixels are outline of inner segment region.

After finding the outline of inner segments, we make distance transform image using only that outline. And then, calculate the shape difference using that distance transform image. To reduce time, we calculate distance value for the pixels around template, not whole image. Our cost function is almost similar with equation (4) of [5] except that our algorithm uses the edge pixels,

$$E_{inner} = \{e|e \in boundaries\, of\, inner\, region\}$$

instead of whole edge pixels, E.

4. Experimental Results

For the experiments, we made a template by extracting edges from object. And then, we carry out our algorithm using some test images which have complex background clutters and occlusions. Fig. 3 shows templates and their matching results. In column (a), the template are shown, and column (b) shows input image. Column (c), (d) compare oriented chamfer matching of [5] and proposed algorithm. And (e) is segmentation results using the result of proposed algorithm.

The blue lines in column (c) (d) is template edges and the green lines are extended template region to allow slight scale change when we selects inner segments. The red lines mean the closest edge pixels of template edges. And, the light blue lines in column (d) represent occluded edges. Note that the red line in column (d) are boundaries of inner segments. It means that the our algorithm is comparing the template and only the edges of true object parts. Finally, our results demonstrate that our algorithm handle occlusion correctly.

5. Conclusion

We proposed a robust shape matching based on over-segmentation. We introduced some problems of

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

Figure 3. (a) template (b) input image (c) matching result of oriented chamfer matching (d) proposed algorithm (e) segmentation result using proposed algorithm

chamfer matching, and described how we use over-segmentation to overcome that problems. Because our algorithm eliminates background clutter edges before calculating chamfer distance, the cost function is much more reliable than previous chamfer matchings. The results demonstrate that our algorithm improves shape matching result dramatically in the presence of clutter and occlusion.

ACKNOWLEDGEMENT

This work was supported by the IT R&D program of MCST [2008-F030-01, Development of Full 3D Reconstruction Technology for Broadcasting Communication Fusion], and by the Defence Acquisition Program Administration and Agency for Defense Development, Korea, through the Image Information Research Center at KAIST under the contract UD070007AD.

References

[1] G. Borgefors. Hierarchical chamfer matching: a parametric edge matching algorithm. *IEEE trns. Pattern Analysis Machine Intelligence*, 10(6):849–865, 1998.

[2] T. Cour and J. Shi. Recognizing objects by piecing together the segmentation puzzle. *CVPR*, 2008.

[3] P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[4] A. Opelt, A. Pinz, and A. Zisserman. A boundary fragment model for object detection. *ECCV*, 2:575–588, 2006.

[5] J. Shotton, A. Blake, and R. Cipolla. Contour-based learning for object detection. *ICCV*, 2005.

[6] A. Thayananthan, B. Stenger, P. Torr, and R. Cipolla. Shape context and chamfer maching in cluttered scenes. *CVPR*, pages 127–134, 2003.