Remaining edges linking method of motion segmentation based on edge detection

Long Xiang, Wu Xiaoqing
College of Information Science and Technology
Hainan University
Haikou, China.

YangXiong
College of Information Science and Technology
Hainan University
Haikou, China.

Abstract— **In our previous work, we proposed a new motion segmentation method based on edge detection. In order to solve the problem of the disconnection of edges of the object in the segmentation result, we proposed a new edge linking method in this paper. In order to reduce the influence of noise, a double threshold segmentation method is used here. From the experimental result, we found that our edge linking method is satisfaction.**

Keywords- motion segmentation; motion parameter vector; edge linking; Gravitation; double threshold

I. INTRODUCTION

Model-based motion segmentation is an important method in computer vision. In this kind of method, statistical theory is usually employed to estimate model parameters and motion regions. Fablet and Bouthemy[1] used the Markov random field and the multiscale Gibbs models to recognize motion in image sequences. Rasmussen and Haiger[2] used a Karman filter and the Probabilistic Data Association Filter to predict the most likely location of a known target to initialize the segmentation process. Cremmers and Soatto[3] used the level set method to implement the segmenting process. Shen et al [4] proposed a MAP method. According to it, an iterative algorithm was used to update the motion field and the segmentation fields along with the high-resolution images.

There are two problems in these model-based methods: the first one is occlusion. When objects are moving, some parts of background can be occluded, while other parts can be appeared. Because of the effect of occlusion, the result of segmentation always can't be very accurate. The other problem is there always are model parameters be estimated in most model-based methods. These parameter estimation processes are always complex and time-consuming.

In order to solve these problems, we proposed a new model-based motion segmentation method—motion segmentation based on edge detection. In this method, the objects we process are not motion regions, but edges. At first, we get all edges of the image by Canny operator in one frame. And then, we use model-based method to remove the edges belonging to the background and remain the edges belonging to the moving object. This method can get the edges of the moving object accurately, and quickly.

There is a problem in our method. After motion segmentation, we can get the edges of the moving object. However, because the move of the object is always not rigid, and there always is effect of complex background and camera noise, some parts of the object's edge are labeled as background, and some background's edges are labeled as the object after segmentation. In [5], a simple technique is proposed to process small pixel groups (Salt and Pepper noise) that differ from their neighbors' labels. Some common morphological methods to combine connected region and remove those isolated region can be found in [5-7]. But all these method are used to process "regions" in the image, but not "edges", so we have to propose our own method to process this problem.

In this paper, we proposed a new edge linking method to connect the disconnected edge pixels in the image after segmentation. This method can complete 2 tasks: A. Linking disconnected edges belonging to the object. B. Eliminating those isolated edges belonging to the background.

This paper is organized as follows: In section 2, our previous work of motion segmentation based on edge detection is introduced simply. In section 3, our edge linking method is discussed in detail. At last, experimental result will be shown in section 4.

II. MOTION SEGMENTATION BASED ON EDGE DETECTION

In model-based method, motion segmentation is actually a process of pattern recognition, where every pixel is processed and classified into two classes—one is background, the other is the moving object. In traditional motion segmentation methods, pixels are the processed one by one, and classified one by one. This process needs large number of calculation work, and occlusion could make the result of this process inaccurate. In our method, only those pixels on the edge of an image are used. Of cause, these edges include the edges of the moving object and edges of the background. The pattern recognition work is to classify these edge pixels into two classes—object or background. In this process, occlusion will not be occurred.

According to the discussion above, the data source of the pattern recognition process is not all of pixels of images of the image sequence, but only the edge pixels of one image of the sequence. In this paper, we use the Canny edge detection method to find edges of one image. Canny edge detection algorithm was proposed in 1986 by J. F. Canny [8]. Today, Canny operator has already been a standard edge detection algorithm in the world.

978-1-4673-0024-7/10/$26.00 ©2012 IEEE

After Canny operator be used, we can get an edge mask image. This mask image will be used as the data source of the following pattern recognition process, which is to extract edges of the moving object from the mask. Here we make an assumption as following: The background is moving or not. If the background is moving, it should be a global dynamic. That is to say, the moving of the background should be rigid.

According to above assumptions, we should distinguish edges in the mask into two kinds—background or object. If the edge is belong to the background, pixels at the two side of the edge should move the same way; if the edge is belong to the moving object, pixels at the two side of the edge should move the different way. Then we can classify the edges in the mask image according to following criteria: If the pixels at the two side of the edge are moving the same way, the edge should belong to the background; if the pixels at the two side of the edge are moving the different way, the edge should belong to the object. In this paper, we use the pattern recognition method to distinguish the different moving of pixels. After all edges in the mask image are distinguished, the edge belonging to the object should be remained, while those edges belonging to background should be removed.

We called this pattern recognition process as background edge removal. This process is implemented by a six-parameter affine motion model, which come from a 2-D motion field on the image:

$$v_{(A,b)}(x,y)=\begin{pmatrix} v_x(x,y) \\ v_y(x,y) \end{pmatrix}=A\begin{pmatrix} x \\ y \end{pmatrix}+b=\begin{pmatrix} a_1 & a_2 \\ a_4 & a_5 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix}+\begin{pmatrix} a_3 \\ a_6 \end{pmatrix} \quad (1)$$

According to the model, we defined the affine motion parameter vector as following:

$$A_i=\begin{pmatrix} a_{i,1} & a_{i,2} & a_{i,3} & a_{i,4} & a_{i,5} & a_{i,6} \end{pmatrix}^T \qquad i=1,2 \quad (2)$$

If all pixels in a small area are moving the same way, their affine motion parameter vector should fulfill following function:

$$\overline{A}_i=\arg\min_{A}\sum_{(x,y)\in C}\left\|v_{A,b}(x,y)-\overline{v}_{\overline{A},\overline{b}}(x,y)\right\|^2 \quad (3)$$

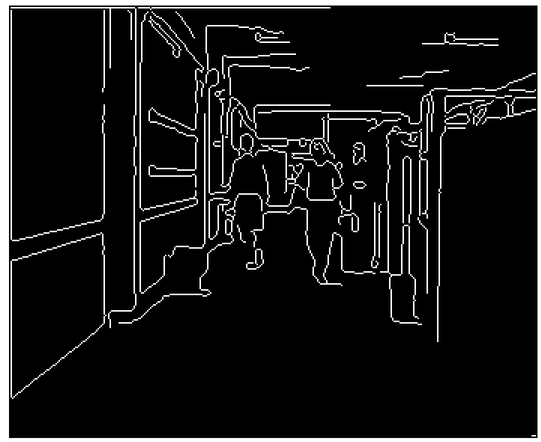
So for a pixel on an edge, we denoted the affine motion parameter vector on its two sides as A1 and A2 separately, there is a threshold DK and a distance measure D(x, y), if $D_{(x,y)} \ge D_K$, then the edge of the pixel should belong to the moving object; if $D_{(x,y)} < D_K$, then the edge of the pixel should belong to the background. Here we define the DK as following:

$$D_{(x,y)}(A_1,A_2)=(A_1-A_2)^T M(A_1-A_2) \quad (4)$$

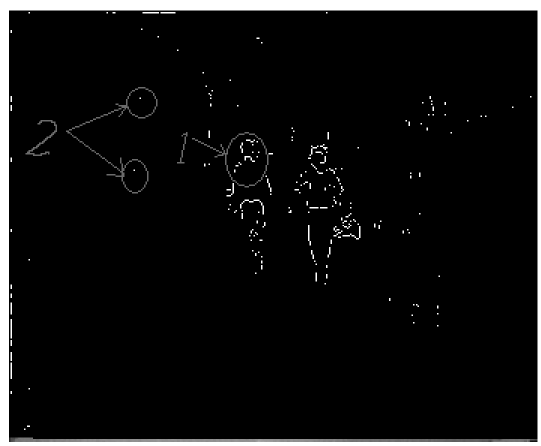
III. REMAINING EDGES LINKING

A. The problem after motion segmentation

According to the method discussed above, we can get the edge of the moving object in an image sequence. But unfortunately, there are some problems in the result. Because the moving of the object is always not rigid, .and the effects of complex background and camera noise are always existing, there are some pixels on an object's edge are labeled as background, and some pixels on an background's edge are labeled as object. This problem is shown in Figure 1 as following:



(a)



(b)

Figure 1. Canny edge image and segmentation result based on edge detection

Figure 1 shows the problem of our previous work. Figure 1 (a) is the Canny edge image, which come from a Canny edge detector, and Figure 1 (b) is the result of motion segmentation based on edge detection of our work. From Figure 1 (b), we can find that there are 2 kinds of false classifications. One is that some edge pixels of the moving men were classified as the background, such as the head of the left man, which was circled in circle 1. The other false classification is that some pixels of the background were classified as the moving men, such as the two pixels on the up-left of the image, which were circled in circle 2. Of cause, we can find other miss classified pixels in the image, which had the same problems as pixels in circle 1 and circle 2 too.

To solve this problem, we proposed the remaining edges linking method. Here we define POB as the pixel which is belonging to the object, but is classified as background; and define PBO as the pixel which is belonging to the background, but is classified as the object. So we called the edge consisting of PBO as PBO-edge, the edge consisting of POB as POB-edge, and edges segmented by our motion segmentation method as P-edges. According to analysis of number of images, we found that edge pixels had following characteristics:

- The PBO-edge is always shorter, and isolated.
- The distance between two PBO-edges is always longer.
- The length of POB- edge is always shorter.
- A “real” object’s edge is always longer.
- The distance between two “real” edges is always shorter.

B. *Edge linking*

In this paper, our work is to link those disconnected “real” edges together, and to remove false edges consisting of PBO.

In order to complete our linking work, we should define the term “distance” at first. Because our method is established on Canny edge detection, all edge pixels, what ever belonging to “real” edges or belonging to “false” edges, are locating at the Canny’s edges. If we want to link two pixels together, we should link them along with the Canny’s edge. So the “distance” between two pixels should be the number of pixels located at the Canny’s edge between these two pixels. We called this kind of distance as “Canny-Distance” or “C-Distance”.

In order to link the disconnected edges of the object, here we define another term—Gravitation. If there are two edges of the object, we denote the number of pixels of one of the edges as m1, and the number of pixels of the other edge as m2, then m1 and m2 should fulfill the following function:

$$G = \frac{g \cdot m_1 \cdot m_2}{d^2} \tag{5}$$

where G is called as the Gravitation of these two edges, d is the C-Distance of the terminals of them, and g is the constant of gravitation .

According to characteristics discussed above, we can make up the following linking criteria:

- There is a threshold L. If The C-Distance between terminals of two P-edge is shorter that L, they should be linked together.
- If The C-Distance between terminals of two P-edges is longer that L, there should be another threshold GT. If the gravitation of these two edges is larger than GT, they should be attracted each other, and should be linked together along the Canny edge between them. Otherwise, they can’t be connected each other.
- After all edges were linked together, there should be a threshold LT. If the total length of an edge is less than LT, this edge should be a PBO-edge, and should be removed. Otherwise, it should be the real edge, and should be remained.

C. *Double thresholds*

According to criteria discussed above, we can begin to do our edge linking work. However, there is still a problem. In some image sequence, the camera noise is severe. The severe noise will cause large number of PBO-edges. This can make our algorithm inaccurate.

In this paper, we made the following assumption: During motion segmenting, we will calculate motion parameter vector of pixels at the two sides of edges. We used the distance D(x, y) to measure the difference of the motion between pixels at the two sides of edges. Here we assumed that if the distance D(x, y) of a edge pixel is larger than most distances of pixels in segmentation result image, this pixel must belong to the object. We called this assumption as L-D assumption.

According to above assumption, we set two thresholds of D(x, y), one of them was denoted as DKL, the other is DKH, where DKL less than DKH. Then we could get two segmentation results, one of them used the threshold DKL, the other used DKH. So in our linking algorithm, DKH image would be used as the seed image of the linking process, because according to the L-D assumption, all P-edges in DKH image should belong to the object.

The problem of the DKH image is that the threshold of segmentation process is so large that a number of POB-edges would be created. In order to solve this problem, the DKH image only can be used as the seed of the linking algorithm. During the linking process, the DKL image should be used.

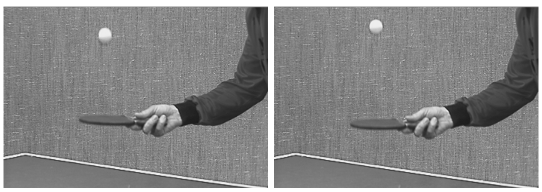
IV. EXPERIMENTAL RESULTS.

In this paper, we used MPEG4 test sequence “Hall monitor”, “Big.Buck.Bunny”, and “Ping Pang” to test our algorithm. Figure 2 shows images selected from these image sequences. Figure 2 (a) and 2(b) shows the 200th and 201st frames of the “Hall monitor” sequence; Figure 2 (c) and 2(d) shows the 5th and 6th frames of the “Hall monitor” sequence; Figure 2 (e) and 2(f) shows the 2013th and 2014th frames of the “Big.Buck.Bunny” sequence; Figure 2 (g) and 2 (h) shows the 16th and 17th frames of the “Ping Pang” sequence.

(a) (b)

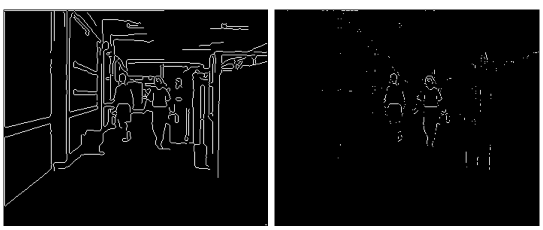
(c) (d)

(e) (f)

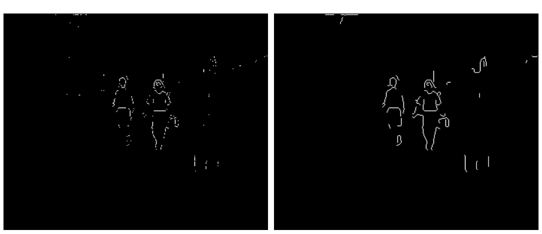


(g) (h)

Figure 2. Images selected from test sequences “Hall monitor”, “Silence” and “Ping Pang”

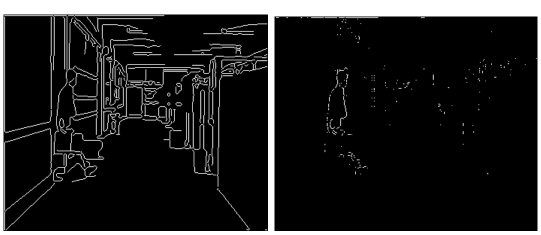


(a) (b)



(c) (d)

Figure 3. Segmentation results of the Sequence “Hall monitor”



(a) (b)

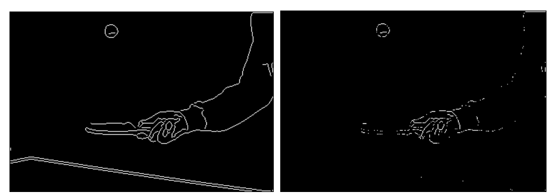
(c) (d)

Figure 4. Segmentation results of the Sequence “Hall monitor”

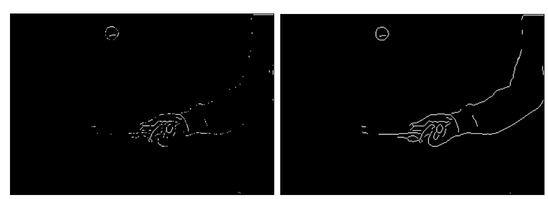
(a) (b)

(c) (d)

Figure 5. Segmentation results of the Sequence “Big.Buck.Bunny”



(a) (b)



(c) (d)

Figure 6. Segmentation results of the Sequence “Ping Pang”

TABLE I. CALCULATION TIME OF EDGE LINKING ALGORITHM

Image sequence	Calculation Time (ms)
Hall Monitor of Fig 3	16
Hall Monitor of Fig 4	16
Big.Buck.Bunny	15
Ping Pang	15

Figure 3 shows the edge linking result of sequence of “Hall Monitor”. Figure 3 (a) is the Canny edge image of Figure 2 (b). Figure 3 (b) is the motion segmentation result by the threshold DKL. Fig 3 (c) is the motion segmentation result by the threshold DKH. Figure 3 (d) is the result of edge linking. Fig 4 shows the edge linking result of sequence of “Hall Monitor”. Figure 4 (a) is the Canny edge image of Figure 2 (d). Figure 4 (b) is the motion segmentation result by the threshold DKL. Figure 4 (c) is the motion segmentation result by the threshold DKH. Figure 4 (d) is the result of edge linking. Figure 5 shows the edge linking result of sequence of “Big.Buck.Bunny”. Figure 5 (a) is the Canny edge image of Figure 2 (d). Figure 5 (b) is the motion segmentation result by the threshold DKL. Figure 5 (c) is the motion segmentation result by the threshold DKH. Figure 5 (d) is the result of edge linking. Figure 6 shows the edge linking result of sequence of “Ping Pang”. Figure 6 (a) is the Canny edge image of Figure 2 (f). Figure 6 (b) is the motion segmentation result by the threshold DKL. Figure 6 (c) is the motion segmentation result by the threshold DKH. Figure 6 (d) is the result of edge linking. In these experiments, the constants of gravitation g are all 10.

Table 1 shows the calculation time of the edge linking algorithm of this paper. The calculation time of the motion segmentation algorithm isn’t listed here, but is listed in another paper of us. Our program is written by Microsoft Visual C++ 6.0, and our CPU is IntelCore2 DUO T7100 with 1 GB internal memory.

From the result of experiments above, we can find that disconnected edges gotten from our motion segmentation method were linked together. However, we can find some edges were not connected. This is caused by the disconnected of these edges in the Canny’s image itself. In some sequences, our method didn’t remove all PBO-edges, because the noise of these sequences is very severe. But this drawback can be overcome in the special applications.

V. CONCLUSIONS

In order to solve the problem of disconnection of object’s edges after the motion segmentation based on edge detection of our precious work, we proposed an edge linking method in this paper. The edge linking algorithm consists of several edge linking criteria. In these criteria, we proposed a Gravitation function to link the edges of the moving object. In order to reduce the influence of noise during linking process, a double threshold segmentation method was used in this paper. According to the result of some experiments, the result of our edges linking method is satisfaction.

REFERENCES

[1] R. Fablet, P. Bouthemy. “Motion recognition using non parametric image motion models estimated from temporal and multiscale cooccurrence statistics.” IEEE Trans. on Pattern Analysis and Machine Intelligence, 25(12):1619-1624, December 2003.

[2] C. Rasmussen and G. D. Hager, “Probabilistic data association methods for tracking complex visual objects,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 6, pp. 560–576, 2001.

[3] D. Cremers and S. Soatto, “Motion competition: A variational approach to piecewise parametric motion segmentation,” International Journal of Computer Vision, vol. 62, no. 3, pp. 249–265, May 2005.

[4] H. Shen, L. Zhang, B. Huang, and P. Li, “A map approach for joint motion estimation, segmentation, and super resolution,” IEEE Transactions on Image Processing, vol. 16, no. 2, pp. 479–490, 2007.

[5] Tuzel O, Porikli F, Meer P. A Bayesian Approach to Background Modeling. IEEE Workshop on Machine Vision for Intelligent Vehicles (MVIV) 2005; 3: 58.

[6] Stauffer C, Grimson W. Learning Patts of activity using realtime Tracking. In IEEE Trans. on Patt Anal Mach Intel Aug 2000; 22:747-57.

[7] Kentaro T, John K, Barry B, Brian M. Wallflower: Principles and Practice of Background Maintenance, ICCV, Seventh Int. Conf. on Com Vision (ICCV'99) - 1999; 1: 255-261.

[8] John Canny. “A computational approach to edge detection.” Pattern Analysis and Machine Intelligence, IEEE Transactions on, PAMI-8(6):679–698, Nov. 1986.