PEDESTRIAN CONTOUR DETECTION BASED ON IMAGE SEGMENTATION

[1] *Tai-Yuan Hsu* (*許泰源*), [2] *Chiou-Shann Fuh* (*傅楸善*)

[1] Dept. of Mechanical Engineering,
National Taiwan University, Taiwan

[2] Dept. of Computer Science and Information Engineering,
National Taiwan University, Taiwan

E-mail: r04522602@ntu.edu.tw fuh@csie.ntu.edu.tw

ABSTRACT

In our previous research, we proposed a method to tracking pedestrian with RGB image based on clothes' colors and control an AR. Drone 2.0 to follow the target with PID control. Though it is robust, a trivial solution needs preset RGB thresholds and ROI. Thus we want to detect ROI automatically and applied it on the previous works.

Keywords Image segmentation; Seeded Region Growing; OpenCV; Pedestrian Contour Detection

1. INTRODUCTION

Image segmentation and pedestrian recognition is one of the most popular research of computer vision. Actually, it is hard to extract the object of interest correctly and even know which one is the contour of pedestrian. In our system, we most bases on the color images and separates different part of object by CIE76 color-difference formula. Thought the formula can handle some the color-difference problem, we still have to process the images first to make the images be less noisy.

The contribution of our research is to apply several simple technique to process the input images. Therefore we do not need to adjust the threshold of HSV color space every time when we need to set a new target person.

Our device is Intel® Core(TM) i3-2330M CPU @ 2.20 GHz / 6.00GB RAM / Windows 7 64 bits.

2. RELATED WORKS

2.1. Image Pre-processing

To make the image easier to classify, we want to group the similar colors together and separate the different colors by large ΔE^*_{ab} or edge gap. We use OpenCV library to do the most of the basic image processing.

First, we increase the contrast of input image. Then, process the input image with edge enhancement kernel [1]. These operations make the edges more strong and the Euclidean distance of similar colors shorter.

2.2. CIE76 Color-Difference Formula

RGB color space is the most popular representation of color image, however it is not the best color space to calculate the difference between colors. Thus, we convert the RGB color space to the CIELAB color space and apply CIE76 formula [2]

$$\Delta E^*_{ab} = \sqrt{(L^*_2 - L^*_1)^2 + (a^*_2 - a^*_1)^2 + (b^*_2 - b^*_1)^2}$$

to evaluate the difference. In this equation, (L^*_1, a^*_1, b^*_1) and (L^*_2, a^*_2, b^*_2) are used to represent two colors in Lab color space, and ΔE^*_{ab} is the Euclidean distance between the two colors. The recommended threshold

$$\Delta E^*_{ab} \approx 2.3$$

corresponds to a JND (Just Noticeable Difference).[3]

2.3. Image Segmentation with SRG

Seeded Region Growing (SRG) [4] is the algorithm to segment image. In order to separate input image in to several part, we apply the basic algorithm on it. To make sure all the picked seeds are not in the same region, we pick a seed to grow first. After the seed region is fully-grown, we pick the next one with checking the seed is not within any grown region and so on.

In the SRG algorithm, CIE76 Color-Difference Formula is used to distinguish different colors. We segmented the image into 30 fragments at least and labeled them with random colors and the minimum size of fragments is also limited (Figure 1).

Figure 1 The segmented image.

After image segmentation, these fragments will pass a size filter to remove the large parts like the background of the scene. The other remaining fragments will be combined and fit the standing pose templates (Figure 2).

Figure 2 Standing pose templates.

2.4. Fragment grouping and fitting

Since the color based SRG separates the image into small pieces, we need to pick the related part and grouping them into one object.

Here, the standing pose templates are used to evaluate which combination is mostly like a human body.

3. EXPERIMENTS

3.1. Description

In order to evaluate the accuracy of our algorithm, we choose several samples for our experiments.

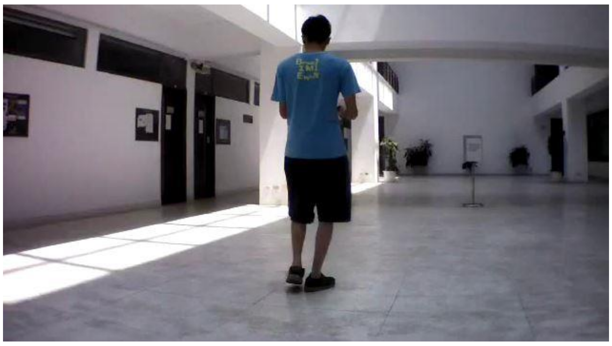

Figure 3 (a) Sample A. (b) Sample B.

Figure 4 (a) Segmentation A. (b) Segmentation B.

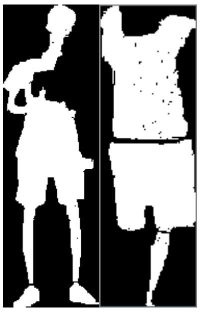

Figure 5 (a) Result A. (b) Result B.

In our experiments, 10 independent images and 10 sequential images are chosen. The independent images are used to evaluate the accuracy of the system. The sequential images are used to locate the target in each frame and evaluate the tracking robustness.

3.2. Independent images testing

First, we picked 10 independent images from our sample data. Some of them are captured from video and the others are photos.

The following table is the experiment data about the execute time and the number of potential parts. In our algorithm, the number of potential parts are limited under 30 pieces in order to reduce the depth of searching.

Sample	Execution Time (sec)	Number of segments
1	18.0	22
2	27.0	23
3	19.0	12
4	210.0	24
5	24.0	17
6	20.0	24
7	19.0	22
8	23.0	12
9	8.0	5
10	59.0	20

Table 1 Execution time and Number of potential segments

Sample	Number of picked segments	Accuracy
1	4	100.00 %
2	6	74.50 %
3	2	80.23 %
4	12	58.73 %
5	6	76.94 %
6	6	85.55 %
7	6	42.44 %
8	8	87.61 %
9	2	100.00 %
10	4	88.70 %

Table 2 Number of picked parts (Definition of Accuracy is the correct recognized area divided by total picked area.)

The following pictures are the result of experiment that the left side is the combination of origin image with recognized area and the right side is the recognized area composed of picked segments.

Figure 6 (a) Sample 1 (b) Output 1

Figure 7 (a) Sample 2 (b) Output 2

Figure 8 (a) Sample 3 (b) Output 3

Figure 9 (a) Sample 4 (b) Output 4

Figure 10 (a) Sample 5 (b) Output 5

Figure 11 (a) Sample 6 (b) Output 6

Figure 12 (a) Sample 7 (b) Output 7

Figure 13 (a) Sample 8 (b) Output 8

Figure 14 (a) Sample 9 (b) Output 9

Figure 15 (a) Sample 10 (b) Output 10

From our result, we can find out that most of the target segmentations are correct, though some of them grouping the incorrect parts.

However, it does not affect the application a lot since the center of mass (COM) of each target are not biased by the error part too much. The COM are still inside the convex hull of human body contour.

In Table 3, we show the coordinate of correct position of COM and the detected position of COM.

In Table 4, we show the comparison result of correct position of COM and the detected position of COM.

From the result shown in Table 4, the percentage error are all under 5%. It is accurate enough for tracking the target with AR. Drone by image information.

Sample	Position of COM(Real)	Position of COM (Detected)
1	(389,107)	(384,136)
2	(285,119)	(264,156)
3	(274,119)	(272,132)
4	(245,121)	(228,116)
5	(325,106)	(324,152)
6	(373,133)	(380,180)
7	(346,139)	(376,132)
8	(303,143)	(312, 180)
9	(422,189)	(404, 224)
10	(274,139)	(272, 168)

Table 3 Real position and Detected position

Sample	Error Distance	Percentage Error
1	7.29	1.14%
2	13.78	2.15%
3	4.40	0.69%
4	6.48	1.01%
5	13.46	2.10%
6	12.00	1.87%
7	8.26	1.29%
8	11.37	1.78%
9	8.51	1.33%
10	9.46	1.48%

Table 4 Distance of Error (Unit: pixel) and Error Percentage (Definition of Error Percentage is the Error Distance divided by the width of the frame where the frame size is 640 x 360

3.3. Sequential image testing

In order to evaluate the tracking algorithm, we apply the algorithm on sequential images from the navigation video. We pick 24 continuous pictures to evaluate the performance of our system.

In this experiment, the system mistake the tree for human in 3 frames. They are no. 6, 10 and 11 frame. The error result image are shown below.

Figure 16 (a) Frame 6 (b) Error Output 6

Figure 17 (a) Frame 10 (b) Error Output 10

Figure 18 (a) Frame 11 (b) Error Output 11

In figure 19, the dots plot shows that there are 3 mistake target detected by the system, but can be removed easily by Kalman filter or setting a threshold to reject sudden movement of targets which often means mistake.

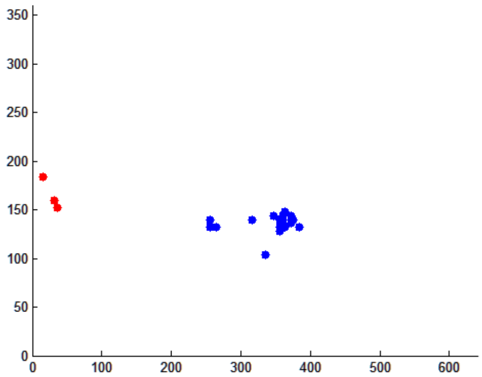

Figure 19 Position of detected targets (**Blue** dots represents the correct target and **Red** dots represents the error target). Here, axis x and y represents the horizontal and vertical position of targets separately.

The execute time and number of segments are shown in Table 5. We can find out that each frame takes less execution time because of less potential segments. Since the background of the 24 frames are simpler than the independent images.

Frame	Execution Time (sec)	Number of segments
1	5.0	9
2	3.0	11
3	3.0	8
4	4.0	7
5	6.0	9
6	8.0	10
7	12.0	11
8	9.0	9
9	8.0	11
10	5.0	8
11	3.0	7
12	3.0	9
13	4.0	10
14	6.0	11
15	8.0	11
16	12.0	10
17	9.0	10
18	8.0	12
19	6.0	9
20	8.0	12
21	7.0	12
22	6.0	7
23	11.0	10
24	3.0	10

4. CONCLUSION AND FUTURE WORK

The result of experiments shows that our algorithm can do segmentation well and can group up the related part into a human body contour. The execution time is still too long, it can be solved by high end hardware device. Now, the system can extract a rough contour, we will still make efforts to improve the robustness and extract contours as complete as possible.

As we mentioned in the introduction, our purpose is to find the human body contour automatically and set the COM point as target point for tracking application. We will try to combine this algorithm with our previous system, improve the performance and make it to be a real time system.

REFERENCES

[1] Mr. Salem Saleh Al-amri1 , Dr. N.V. Kalyankar2 and Dr. Khamitkar S.D 3 1 Research Student Computer Science Dept., Yeshwant College, Nanded 2 Principal, Yeshwant College, Nanded 3 Schoolof Computational Science Swami Ramanand Teerth Marathwada University, Nanded 431606 (M.S., India), *IMAGE SEGMENTATION BY USING EDGE DETECTION*, Salem Saleh Al-amri et. al. / (IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 03, 2010, 804-807

[2] Gaurav Sharma, Digital Color Imaging Handbook, CRC Press, Inc. Boca Raton, FL, USA ©2002, p.31

[3] M. Mahy, L. Van Eyckden, and A. Oosterlinck, “Evaluation of uniform color spaces developed after the adoption of CIELAB and CIELUV,” Color Res. Appl., vol. 19, no. 2, pp. 105–121, Apr. 1994.

[4] Yu-Hsiang Wang, *Tutorial: Image Segmentation*, Graduate Institute of Communication Engineering National Taiwan University, Taipei, Taiwan. Available: http://disp.ee.ntu.edu.tw/meeting/%E6%98%B1%E7%BF%94/Segmentation%20tutorial.pdf