Object Tracking Algorithm based on Combination of Edge and Color Information

[1] *Hsiao-Chi Ho* (*賀孝淇*), [2] *Chiou-Shann Fuh* (*傅楸善*) , [3] *Feng-Li Lian* (*連豊力*)

[1] Dept. of Electronic Engineering
National Taiwan University, Taiwan
Email: r02921061@ntu.edu.tw
[2] Dept. of Computer Science and Information Engineering,
National Taiwan University, Taiwan
E-mail: fuh@csie.ntu.edu.tw
[3] Dept. of Electronic Engineering
National Taiwan University, Taiwan
Email: fengli@ntu.edu.tw

ABSTRACT

To the problem of moving object tracking, CAMSHIFT (Continuously Adaptive Mean SHIFT) is widely used because of its fast computation. But, there are some situations, for example complex color information which will lead to CAMSHIFT misdetect. In this paper, we propose a modified CAMSHIFT based on edge orientation histogram and background color suppression to improve those cases. The edge information is used as the matching information to complement the indeterminacy of color information and background color suppression is used to reduce the noise and outlier of the tracking object. The experiments show that the modified algorithm can deal with tracking complex-color object well and realizing tracking efficiently.

Keywords *object tracking***,** *CAMSHIFT, edge orientation histogram, background suppression;*

1. INTRODUCTION

Tracking mobile object is a popular issue in computer vision theory, and there are many researchers have proposed such algorithms and its applications. The algorithm of MeanShift is widely used because of its rapid matching and low computation cost. MeanShift was first proposed by Funkunage [1] to the problem of recognition and Cheng [2] applied to the field of computer vision. But, MeanShift can not deal with the continuous object models. When the tracking object has severe deformation, it will lost the target. To solve this problem, Bradski [3] proposed an algorithm, CAMSHIFT, which apply MeanShift to continuous image sequence and automatically adjust the size of tracking window. However, it is based on color information building the histogram to realize the tracking, which is easily disturbed by similar color object or background.

In this paper, we propose a modified CAMSHIFT which combine the information of edge orientation histogram and color histogram; then, construct the background color histogram to distinguish the background and foreground used to suppress the background information. The proposed method could provide more robust and specific tracking results and higher precision and accuracy while complicated background or object.

2. RELATED WORKS

As mentioned above, several works have tried to improve CAMSHIFT's performance and accuracy. Traditionally, CAMSHIFT only works in one dimension color histogram [5] to represent a target. Therefore, combining other information with color information is a popular way to improve the accuracy of algorithms. In [6], Yin et al. propose an improved CAMSHIFT which combines texture information with color to model the tracking target.

Tian et al. in [7], on the other hand, use improved HSV combined with color histogram model for CAMSHIFT to realize object tracking which settle specific color bins with designed ranges. Moreover, the approach in [8] by Exner et al. combines multiple color histograms with accumulated histogram to compute the probability distribution and represent the appearances of a target.

To reduce the color interference between target and background, there is an improvement made by background modeling. In [9], the approach modifies CAMSHIFT using adaptive background (ACBShift) based on Baysian probability model.

3. OBJECT TRACKING

A. Edge orientation histogram

1) Edge Detection

The edge of image is composed of some specific pixel of its orientation and magnitude. We use Sobel operator [4] to detect the edge of gray image G by those two models:

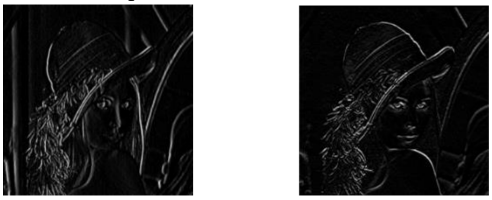
$$S1 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & 1 \end{bmatrix} \qquad S2 = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$$

Those models do convolution with G; then, gain the horizontal and vertical edge images G1 and G2 as shown in Fig. 1. The value of magnitude and orientation in a pixel g_{ij} of G could be shown as below:

$$M(g_{ij}) = \sqrt{|g_{ij}{}^{(1)}|^2 + |g_{ij}{}^{(2)}|^2} \quad (1)$$

$$O(g_{ij}) = \tan^{-1}\frac{g_{ij}{}^{(2)}}{g_{ij}{}^{(1)}} \times \frac{180°}{\pi} \quad (2)$$

Here, $g_{ij}{}^{(1)} \in G1$, $g_{ij}{}^{(2)} \in G2$, $g_{ij} \in G$ and i, j respectively correspond to the row and column of a pixel. The range of $O(g_{ij})$ is $0° \leq O(g_{ij}) \leq 360°$, which is based on polar coordinate.



(a) (b)

Fig. 1 Edge detection of Lena.
(a) Horizontal edge image. (b) Vertical edge image.

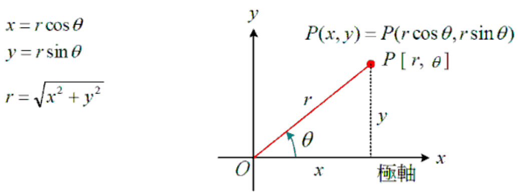
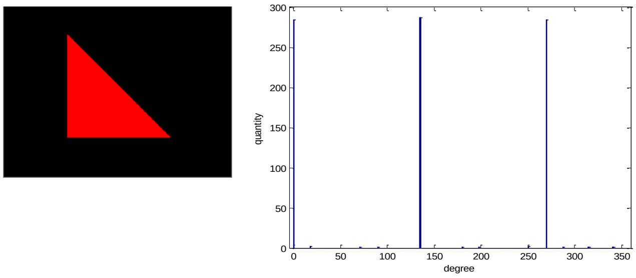


Fig. 2 Polar coordinate.

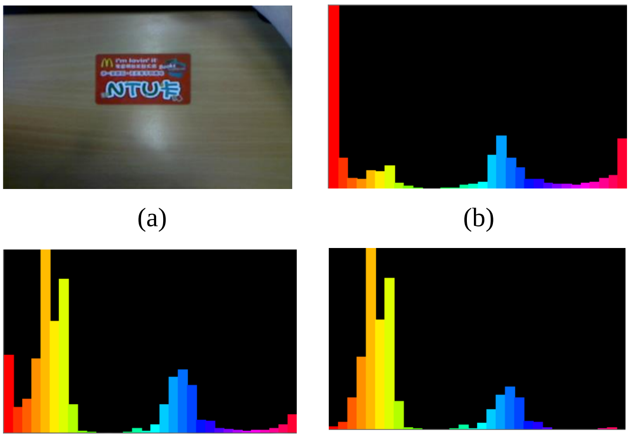
2) Edge Orientation Histogram

To easily build histogram [4], we divide $O(g_{ij})$ to n pieces and n could be chosen. The larger n you choose, which the larger accuracy you could get, however, the larger computation cost you would get. Here, we choose n = 16, which means the interval is between $\left[-\frac{2\pi}{n} + \frac{2\pi}{n}k, \frac{2\pi}{n}k\right]$, for k = 1, 2,⋯, n.



(a) (b)

Fig. 3 Example of edge orientation histogram. (a) Right triangle for computing the edge orientation example. (b) Outcome of the right triangle.



(a) (b)

(c) (d)

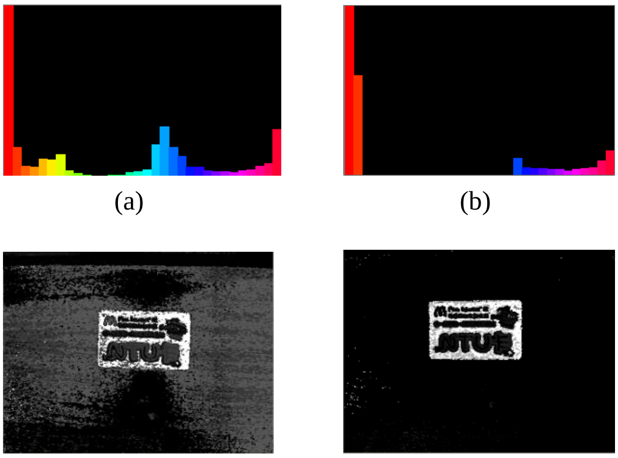
Fig. 4 (a) Original image. (b) Foreground (card in the middle) color histogram. (c) Full-scene color histogram. (d) Background color histogram which is computed by subtraction between full-scene and foreground d. The BG_i , FS_i , FG_i histograms are normalized, so they are equal-height.

To analyze the quantity of histogram, we quantize the information and normalize them to prevent from the influence of image scaling. The computation of edge orientation histogram is shown below:

$$H_k = n_k = \sum \text{num. of pixel}, \text{ if } \delta[M(g_{ij})] \neq 0 \quad (3)$$

$$\delta[M(g_{ij})] = \begin{cases} 1, & \text{if } M(g_{ij}) > 90 \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

H_k is the value of edge orientation histogram , and $n_k$is the numbers of pixels satisfied $\delta[M(g_{ij})] \neq 0$. The method of normalization is that compress the largest value of H_k. Then, according to the scale, compress the others.



(a) (b)

(c) (d)

Fig. 5 (a) Original foreground color histogram. (b) The foreground color histogram after background suppression. (c) Back projection of histogram (a). (d) Back projection of histogram (b).

B. Background Color Suppression

In CAMSHIFT algorithm, it is mainly influenced by color information, but there is a good problem to deal with color similarity between foreground and background, especially in the case of complicated scene [9].

1) *Background Color Histogram Extraction*

Because CAMSHIFT is semi-automatic tracking algorithm, we have to select the tracking object first. According to the object we select, we can gain the color histogram of it; then compute the background color histogram as below:

$$BG_i = FS_i - FG_i \quad , i = 1,2, \ldots ,32 \qquad (5)$$

BG_i , FS_i , FG_i respectively represent background, full-scene, foreground and *i* means the color bin of histogram. Here, we choose the range of *i* is between [1, 32]. In Fig. 4, we could see the background color extraction by this method.

2) *Weight Function*

In Fig4, it can be seen that the most color bins of background are higher than foreground, so if we do back projection in traditional way, most of the color of background will also be projected. To avoid this problem, we hope to suppress the color of background and highlight the importance of foreground color.

To reconstruct the foreground color histogram, we propose a weight function W_i.

$$W_i = \mathrm{Max}\{\left(1 - \frac{BG_i}{FG_i}\right), 0\} \qquad (6)$$

$$Newbin_i = W_i \times FG_i \quad , i = 1,2, \ldots ,32 \qquad (7)$$

Where W_i is determined by the rate of foreground and background color, if the quantity of the background color bin is higher than foreground, the value of W_i is equal to 0 in order to suppress the background color. In fig. 5, we can see the differece of backprojection after the background color is suppressed.

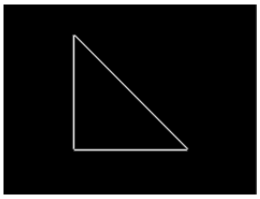


Fig. 6 Back projection of edge orientation histogram of fig 3 (a).

C. Modified CAMSHIFT Tracking

The principle of original CAMSHIFT is based on the H component of HSV (Hue, Saturation, Value) which build color histogram and produce the back projection to match the maximum density distribution similarity.

1) *Back projection*

After we get the color and edge orientation histogram, we can use them to reproduce an image of interested information. There is the principle of back projection [8].

I. In each pixel of image, collect the data and find the correspondent bin for that pixel.
II. Lookup the model histogram in the correspondent bin and read bin value.
III. Store this bin value in a new image, back projection. Also, we normalize the model histogram for easy analysis.

The back projection of edge orientation histogram is shown in fig. 6, which is based on the histogram and image in fig. 3.

Therefore, as we select the target, we can gain the projection of interested orientation. According to it, we can not only gain the similar contour set of target but also eliminate the outlier.

2) *CAMSHIFT algorithm*

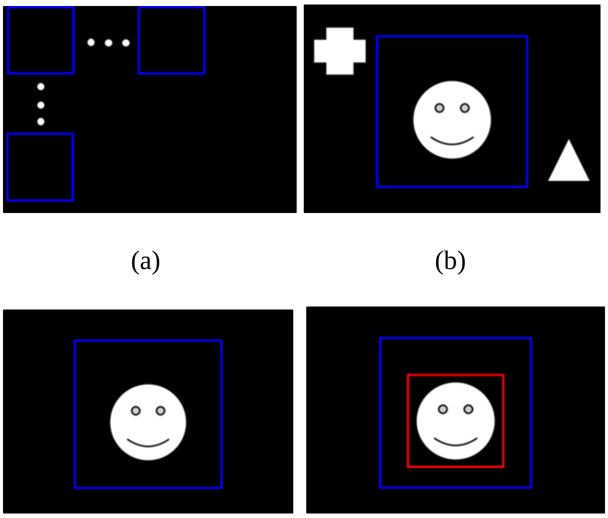
The CAMSHIFT algorithm can be summarized shown as below [11]:

I. Set the initial region of interest to obtain the initial position of the object we want to track.
II. Calculate the color histogram of the region.
III. Produce a probability distribution image using the histogram.
IV. According to MeanShift algorithm, find the center mass of probability distribution image.
V. Iteratively adjust the search window to the point from step V until converge.
VI. For the following frame, process again with the search window from step VI.

In this paper, we use not only color histogram but also edge orientation histogram to the algorithm. Then, we use back-projection approach to construct the probability distribution image.

(a) (b)

Fig. 7 (a) Original edge back projection, (b) the edge back projection after dilation.



(a) (b)

(c) (d)

Fig. 8 The example of information combination. (a) Initial tracking window searching. (b) Initial tracking window selection. (c) Outcome after ROI. (d) Finial tracking output. The blue tracking window is the initial tracking window of step II, and the red one is the final output tracking window.

3) *Information Combination*

Because the density of edge information is not obvious, I dilate the back projection of edge orientation in 5*5 block to emphasize the information. In fig. 7, we can see the difference of the information dilated.

There are two main information (edge, color) we could analyze. In order to use them in efficiency, we follow the principles shown as below to combine them:

I. Do CAMSHIFT with edge back projection based on initial selection for initial tracking window.

II. According to the initial tracking window, set region of interest (ROI) for color back projection.

III. Use region growing to reconstruct the required information eliminated by ROI.

IV. Do CAMSHIFT again with color back projection based on the initial tracking window produced by step I and output the finial tracking window.

The concept of ROI is to eliminate the outlier; then, we can get the more specific information we want. However, if the initial tracking window doesn't track well, some parts of color back projection will be eliminated. To solve the problem, we use region growing to reconstruct the missing information. First, select a threshold λ_1 and growing seed. Second, according to the seed, find the 8-connected points. Third, if the pixel value of 8-connected points is larger than λ_1, replace the value with μ, chosen 255 here. With this approach, if we choose the exact growing seed, we can reconstruct the information. We choose them in the center of initial tracking box, and give another threshold λ_2 to prevent selecting the undesired points.

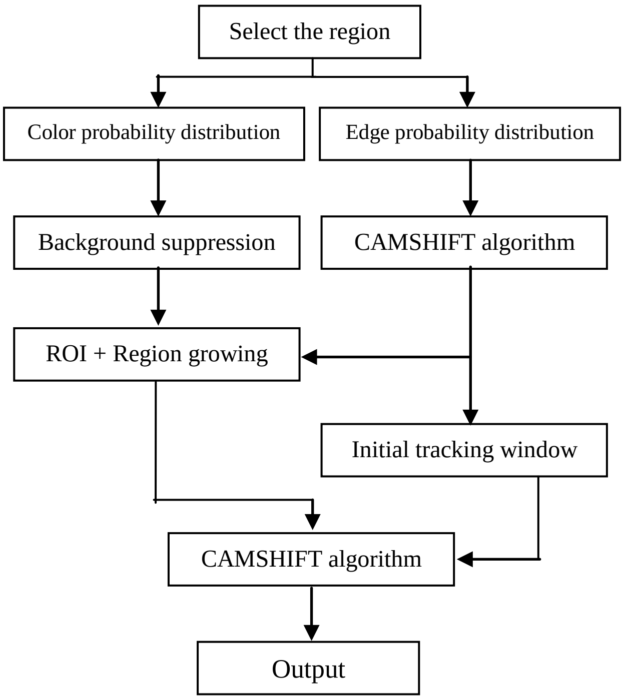


Fig. 9 The proposed method schema.

The tracking outcome of edge orientation is not always precise. In real world, although we have selected an object gaining the edge orientation histogram, it might also project the information of background. So, the tracking box is usually larger than the object we want track; therefore. As we do step I, doing CAMSHIFT with edge back projection, it can roughly gain the initial position of the object. Then, use step IV to obtain more precise position with background color suppressed information. Besides, in step I, we track the object based on initial selection. However, rather than initial selection in step IV, we use the initial tracking window produced by step I to CAMSHIFT. It can dynamically update the iteration position and prevent losing the target, which increase the tracking accuracy and make it more robust.

4. EXPERIMENTS

The experiments were performed in real time on a PC with Intel Core i7 3.4GHz CPU, 4GB RAM and we have implemented all algorithms with C++ code on Visual Studio 2010, using OpenCV 2.4.0 library.

The tracked region is marked by an ellipse surrounding the object. In fig. 10 and fig. 11, we respectively selected the magnet in the middle and the card on the left side as tracking target. Besides, in (a), it was traditional way of CAMSHIFT only using color information.

In (b), it was the result of the method we proposed using edge, color information combination and background color suppression. Although the object was motionless, we moved the webcam to make the environment dynamic. In fig. 10 and fig. 11 (a), the tracking window were going to be out of our expected as we only used color information, but in (b), the object were successfully detected and tracked.

As shown in the results, the algorithm in this paper can deal with the problem of tracking complex color target and improve the robustness of tracking moving target.

Frame 14 Frame 80 Frame 112 Frame 182

(a)

Frame 23 Frame 72 Frame 154 Frame 223

(b)

Fig. 10 Automatically track the object, magnet in the middle. (a) Tracking of CAMSHIFT based on the color information only. (b) Tracking of CAMSHIFT based on edge orientation histogram and background color suppression.

Frame 16 Frame 34 Frame 92 Frame 171

(a)

Frame 37 Frame 76 Frame 134 Frame 201

(b)

Fig. 11 Automatically track the object, card on the left. (a) Tracking of CAMSHIFT based on the color information only. (b) Tracking of CAMSHIFT based on edge orientation histogram and background color suppression.

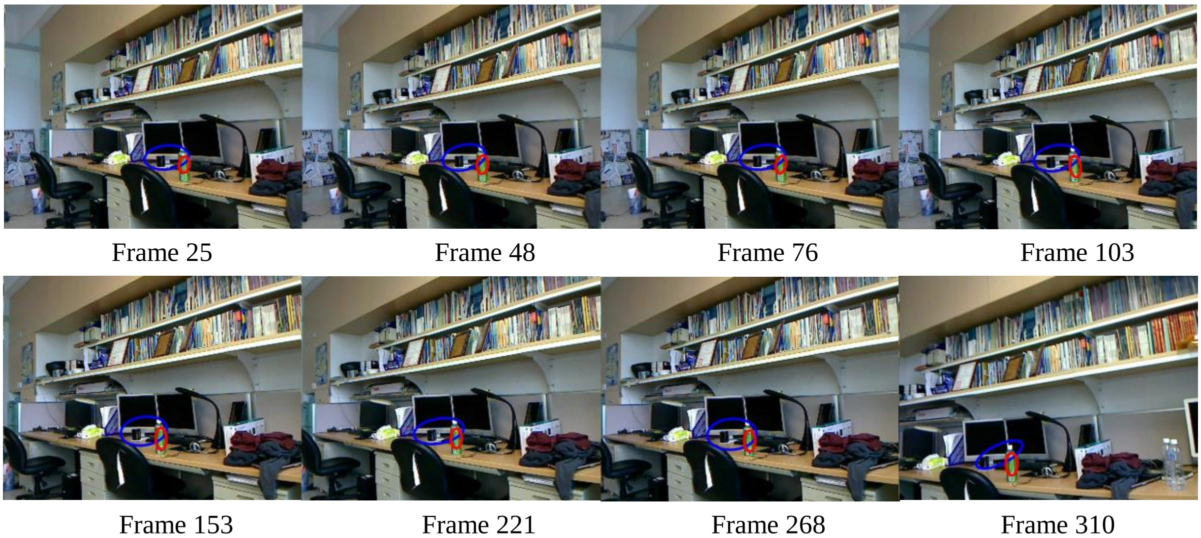


Fig. 12 Automatically track the object, bottle in the middle. The blue ellipse is initial tracking window and the red ellipse is output tracking window. Although the background is complex, the proposed method continues tracking correctly.

5. CONCLUSIONS

In this paper, the proposed algorithm is a novel automatic object tracking using edge orientation histogram and background color suppression. Use the edge orientation histogram on CAMSHIFT to obtain the initial tracking window, then, do CAMSHIFT with color histogram based on the initial tracking window. Only using color features are not reliable for many tracking task such as tracking complex color object, similar color background, and complex scene. However, the proposed algorithm combines edge and color information, so we can prevent the problem of only using an information. The experimental results showed that the approach was able to achieve good tracking result and make tracking system more robust. However the limitation of the proposed method is that the object should have relatively simple contour which will increase the accuracy of edge orientation histogram. And the object could not have severe movement to avoid the edge orientation histogram mismatch.

REFERENCES

[1] K. Fukunaga, and L. Hostetler, “The estimation of gradient of a density function, with applications in pattern recognition,” IEEE Transactions on Information Theory, Vol. 21, No. 1, pp. 32-40, 1975.

[2] Y. Cheng, “Mean Shift, Mode Seeking, and Clustering,” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 17, No. 8, pp. 790-799, 1995.

[3] G. R. Bradski, “Computer Vision Face Tracking for Use in a Perceptual User Interface,” IEEE Workshop Applications of Computer Vision, 1998.

[4] N. Dalai, and B Triggs, “Histograms of oriented gradients for human detection,” in Proceedings of IEEE international Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, Vol. 1, pp. 886-893, 2005.

[5] G.J. Allen, Y.D. Richard Xu, and S. Jin Jesse, “Object Tracking Using CAMSHIFT Algorithm and Multiple Quantized Feature Spaces,” Inc. Australian Computer Society, vol.36, 2004.

[6] J. Yin, Y. Han, J. Li, and A. Cao, “Research on Real-Time Object Tracking by Improved CAMSHIFT,” International Symposium on Computer Network and Multimedia Technology, Wuhan, China, pp. 1-4, Jan. 2009.

[7] G. Tian, R. Hu, Z. Wang, and Y. Fu, “Improved Object Tracking Algorithm Based on New HSV Color Probability Model,” International Symposium on Neural Networks: Advances in Neural Networks – Part II, Wuhan, China, 2009.

[8] D. Exner, E. Bruns, D. Kurz, A. Grundhofer, and O. Bimber, “Fast and Robust CAMSHIFT Tracking,” in Proceeding of IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, San Francisco, CA, pp. 9-16, Jun. 2010.

[9] R. Stolkin, I. Florescu, M. Baron, C. Harrier, and B. Kocherov, “Efficient Visual Servoing with the ABCShift Tracking Algorithm,” in Proceeding of IEEE International Conference on Robotics and Automation, Pasadena, CA, pp. 3219-3224, May 2008.

[10] E. Emami, and M. Fathy, ”Object Tracking Using Improved CAMSHIFT Algorithm Combined with Motion Segmentation,” in Proceeding of IEEE Iranian Conference on Machine Vision and Image Processing, Tehran, Iran, pp. 1-4, Nov. 2011.

[11] G. R. Bradski, “Computer Vision Face Tracking for Use in a Perceptial User Interface,” Intel Technology Journal, Vol. 2, pp. 13-27, 1998.