A Modified Mean Shift Algorithm for Visual Object Tracking

Shu-Wei Chou[1], Chaur-Heh Hsieh[2], Bor-Jiunn Hwang[3], Hown-Wen Chen[4]
Department of Computer and Communication Engineering, Ming-Chuan University, Taoyuan, Taiwan
[1]Email: nnnn0730@hotmail.com Tel: +886-911-270-979
[2]Email: hsiehch@mail.mcu.edu.tw Tel: +886-3-3507001-3441
[3]Email: bjhwang@mail.mcu.edu.tw Tel: +886-3-3507001-3430
[4]Email: hwenchen@ mail.mcu.edu.tw Tel: +886-3-3507001-3743

Abstract—The CamShift is an adaptive version of Mean Shift algorithm. It has received wide attention as an efficient and robust method for object tracking. However, it is often distracted or interfered by the other larger objects with similar colors. This paper presents a novel tracking algorithm based on the mean shift framework. Unlike the CamShift, which uses the probability density image determined by the color feature, the proposed algorithm employs the probability density image derived from both color and shape features. Experimental results indicate the proposed algorithm improves robustness without sacrificing computational cost, as compared to the conventional CamShift algorithm.

I. INTRODUCTION

Object tracking is an important task in video processing and computer vision. It has wide applications such as video surveillance, human computer interaction, vehicle navigation, and intelligent transport system [1]. Several challengeable issues on the robustness of object tracking include illumination changes, occlusions, and distractions, etc. Real-time processing is a fundamental requirement in object tracking. However, robustness and processing efficiency are often conflict, and thus a compromise must be taken. Many methods have been presented to improve robustness and/or computational efficiency [2]-[6].

Among various tracking methods presented, the Mean Shift algorithm is a popular one due to its simplicity and efficiency. It is an iterative kernel-based deterministic procedure which converges to a local maximum of the measurement function [6]. Bradski [6] modified the Mean Shift and developed the Continuously Adaptive Mean Shift (CamShift) algorithm for face tracking. CamShift is an adaptive version of mean shift based on a probability density image obtained with a back projection method from the color histogram of the target. It has received wide attention recently as an efficient and robust method for object tracking. However, the conventional CamShift method still suffers from drawbacks such as distraction by similar color objects, interference of background, and occlusion.

This paper presents a novel tracking algorithm based on the mean shift framework. Unlike the conventional CamShift algorithm, which uses the probability density image determined by color feature, our method calculates the probability density image from color and shape features of the target to be tracked. Using this density image, we can compute the moment features and then estimate the object center based on the zeroth order moment. Experiments indicate the proposed algorithm perform better than the conventional CamShift in distraction and interference.

II. THE PROPOSED TRACKING ALGORITHM

The main difference between the conventional CamShift and our algorithm is the derivation of the probability density image (probability map). In the conventional CamShift, the probability density image is obtained from the color histogram of the target model. However, our method designs a dominant color filter function based on the histogram of the target image to generate the object template, which implicitly includes the shape feature of the target. The dominant color filter function is also applied to the search window to generate a dominant color map. By matching the object template with each target candidate in the dominant color map, we obtain probability map. Using the probability map, we compute the moment features and then estimate the object center and object size based on the moments. The proposed tracking algorithm is summarized in Figure 1, and the detail procedures are described step by step in the following.

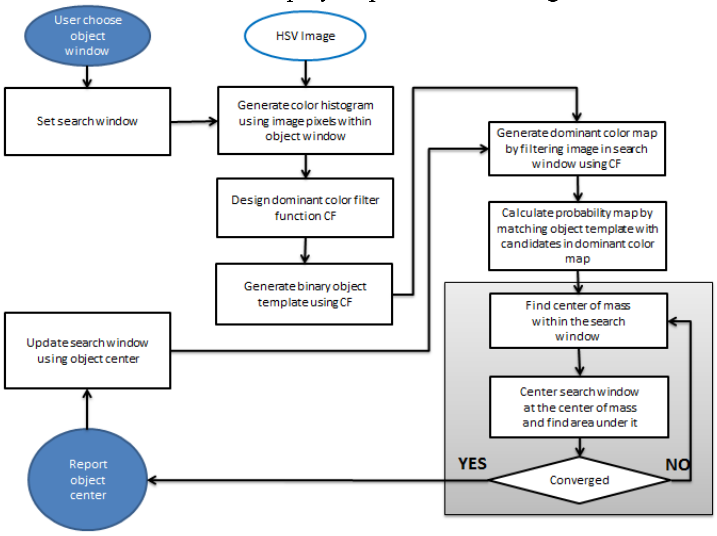


Figure 1. The block diagram of the proposed tracking algorithm

Step 1

Initially, a user manually chooses a target with size $w \times h$ to be tracked, which is referred to as object window. We then define a search window centered at the object window with size $(s\times w)\times(s\times h)$, where the scaling factor s is set to 1.3.

Step 2

Generate color histogram with N bins using the image pixels within the object window, as illustrated in Figure 2. The histogram denotes the color distribution of the object to be tracked.

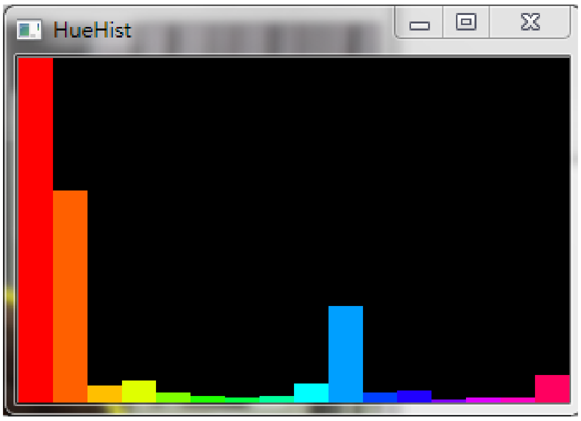


Figure 2. Color histogram

Step 3

Define M (M<N) dominant colors and then design color filter function as

$$\mathrm{CF}(k) = \begin{cases} 1 & \text{if k belongs to defined dominant} - \text{color bins} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Figure 3 demonstrates the color filter function with M=2.

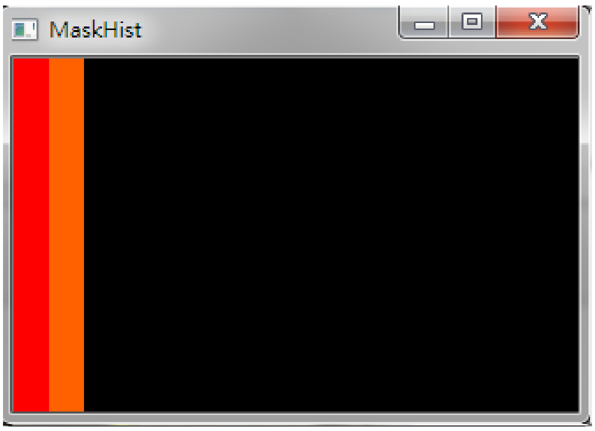


Figure 3. Binary color filter function

Step 4

Generate a binary object template by applying CF to the images pixels within the object window as

$$\mathrm{O}(i,j) = \begin{cases} 1, & \text{if } CF(k) = 1 \text{ for pixel } x(i,j) \\ 0, & \text{if } CF(k) = 0 \text{ for pixel } x(i,j) \end{cases} \quad (2)$$

Note that O(i,j)=1 represents that the color of the pixel at (i,j) belongs to the defined dominant colors. The object template includes the implicit shape information of the target to be tracked, as demonstrated in Figure 4.

Figure 4. Tracking object template

Step 5

Generate a dominant color map by filtering the image in the search window using CF. The operation of this step is similar to Step 4. It is noted that the output dominant color map is also binary, as shown in Figure 5.

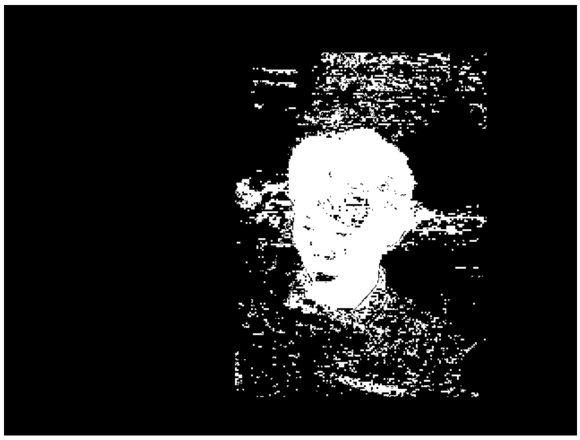


Figure 5. Dominant color map of the search window

Step 6

Calculate a probability map by matching the object template with each candidate in the dominant color map. A candidate is a block with the same size as the object template, i.e., $w \times h$, taken from the dominant color map. The distance of neighboring candidates can be defined in advance. In our work, it is set to 10 for the consideration of computational efficiency.

The matching between the object template O and a candidate block C is done with a simple binary comparison operation

$$S(O, C) = \sum O(i,j) \oplus C(i,j) \quad (3)$$
$\oplus$: inversion of Exclusive OR

The match probability map is obtained as

$$P(m, n) = \frac{S(O,C)}{w \times h} \quad (4)$$

(m, n) : location of a candidate

Figure 6 shows an example of the probability map generated, in which different colors in the target circle represent different probability values. The center of the target has the largest probability value, and the regions which are far from the center have smaller probability. Since the distance of the neighboring candidates are 10, we copy the probability of a candidate to its neighboring pixels in the range of (-5 ~ +5) in x and y directions, respectively, and construct the probability map.

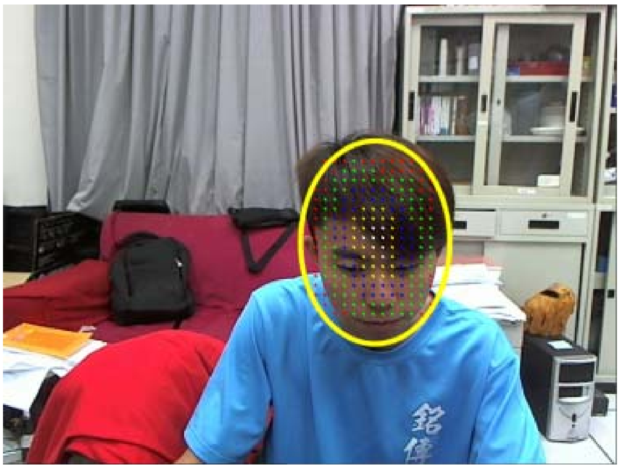


Figure 6. An example of probability map

Step 7

Perform Mean Shift algorithm using the match probability map until convergence. Then we have the object center of the video frame.

Step 8

Update search window with the object center and then repeat Step 5 to Step 8 for the next frame.

III. EXPERIMENTAL RESULTS

The proposed tracking was applied to two video sequences. In our experiments, all videos are converted into HSV color space. Two types of features are conducted separately: H histogram and (H,S) histogram. They are quantized into 16 bins and 16×16 bins, respectively. The number of dominant colors is set to 2.

The first video sequence has 328 frames of 640×480. The tracking target is the left hand of a human, which moves randomly across the human face. As shown in Figure 7 (a)-(b), the tracking of hand with the conventional CamShift is distracted by the human face, and consequently loses track. However, Figure 7(c) indicates our algorithm can attack this drawback.

The second video has 199 frames of 640×480, and the human head is the tracking target. In this scenario, a man who carries a box with color similar to tracking human face is moving from the one side to the other side. As shown in Figure 8(a)-(b), with CamShift, the search window size is affected by the box, and thus reducing tracking accuracy. Nevertheless, our algorithm keeps the same window size in the tracking process under the interference, as shown in Figure 8(c).

From the experiments of the above two videos, we found that our algorithm can achieve satisfactory results with only H component. Thus we use H component only in this work for simplicity.

IV. CONCLUSIONS

This paper has presented a robust adaptive version of the Mean shift tracking algorithm. It derives the probability density image of a particular frame based on color and shape features. The implicit shape information of the region to be tracked is extracted with a novel dominant color filter function, which is derived from color histogram of the target to be tracked. Experimental results indicate our proposed algorithm offers better robustness than the conventional CamShift algorithm especially in distraction and interference. The robustness to occlusion and illumination change will be further investigated in the future.

ACKNOWLEDGEMENT

This work was supported by the National Science Counsel, Taiwan, Republic of China, under Grants NSC 99-2632-E-130-001-MY3 and NSC 102-2221-E-130 -019.

REFERENCES

[1] A. Yilmaz, O. Javed, and M. Shah, “Object tracking: A survey, “ ACM Comput. Survey, vol. 38, no. 4, pp. 1–45, Dec. 2006.

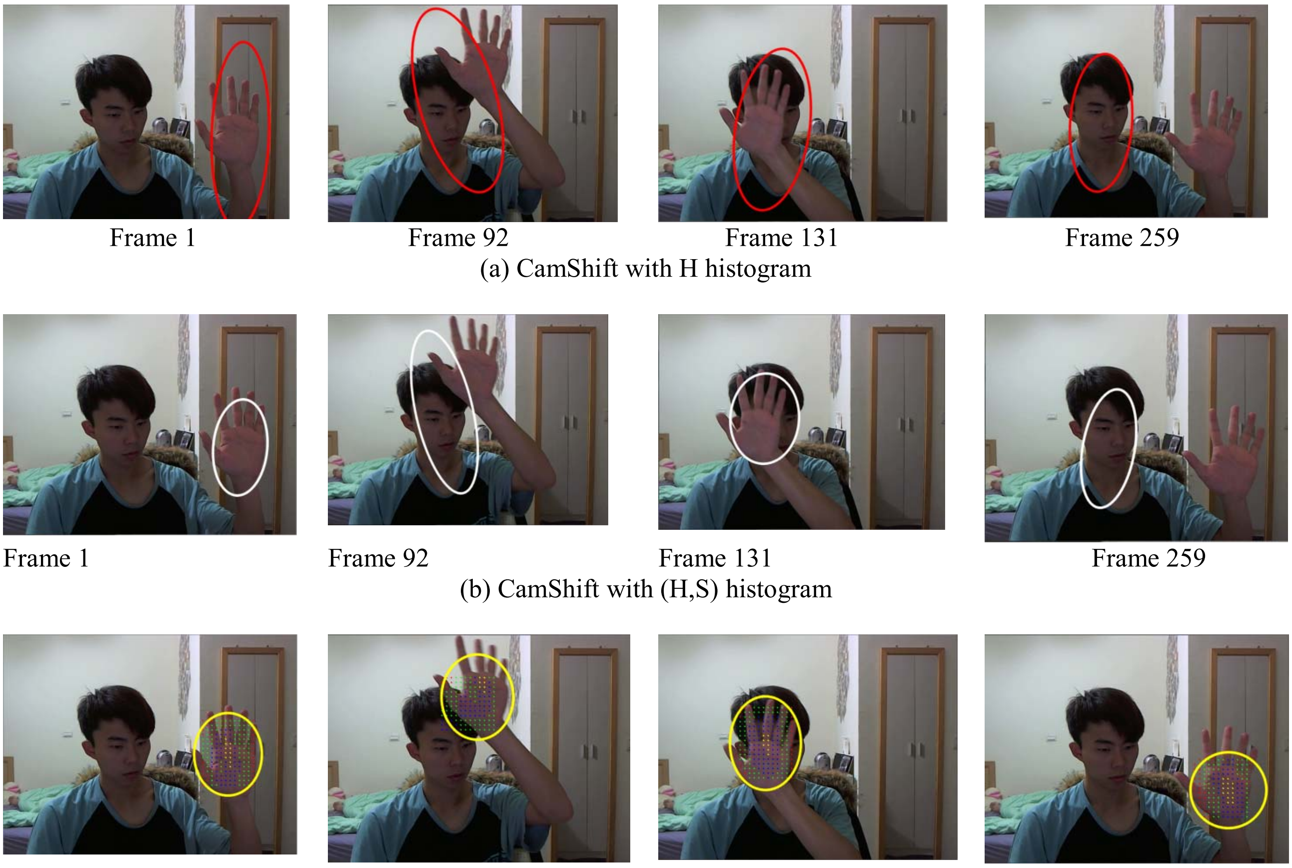
[2] D. Comaniciu, V. Ramesh, and P. Meer, “Real-time tracking of non-rigid objects using mean shift,” In IEEE Proc. on Computer Vision and Pattern Recognition, pages 673–678, 2000.

[3] D. Comaniciu, V. Ramesh, and P. Meer , “Kernel-based object tracking,” IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 25, No. 5, pp. 564-577, May 2003.

[4] L. Ido, and L. Michael, and R. Ehud, “Mean Shift tracking with multiple reference color histograms”. Computer Vision and Image Understanding, no. 9, pp. 400–408. 2010

[5] J. G. Allen, R. Y. D. Xu, and J. S. Jin, “Object tracking using CamShift algorithm and multiple quantized feature spaces,” In Proc. of Pan-Sydney Area Workshop on Visual Information Processing VIP2003, 2003.

[6] G. R. Bradski. “Computer vision face tracking for use in a perceptual user interface,” Intel Technology Journal, 1998.



Frame 1 Frame 92 Frame 131 Frame 259

(a) CamShift with H histogram

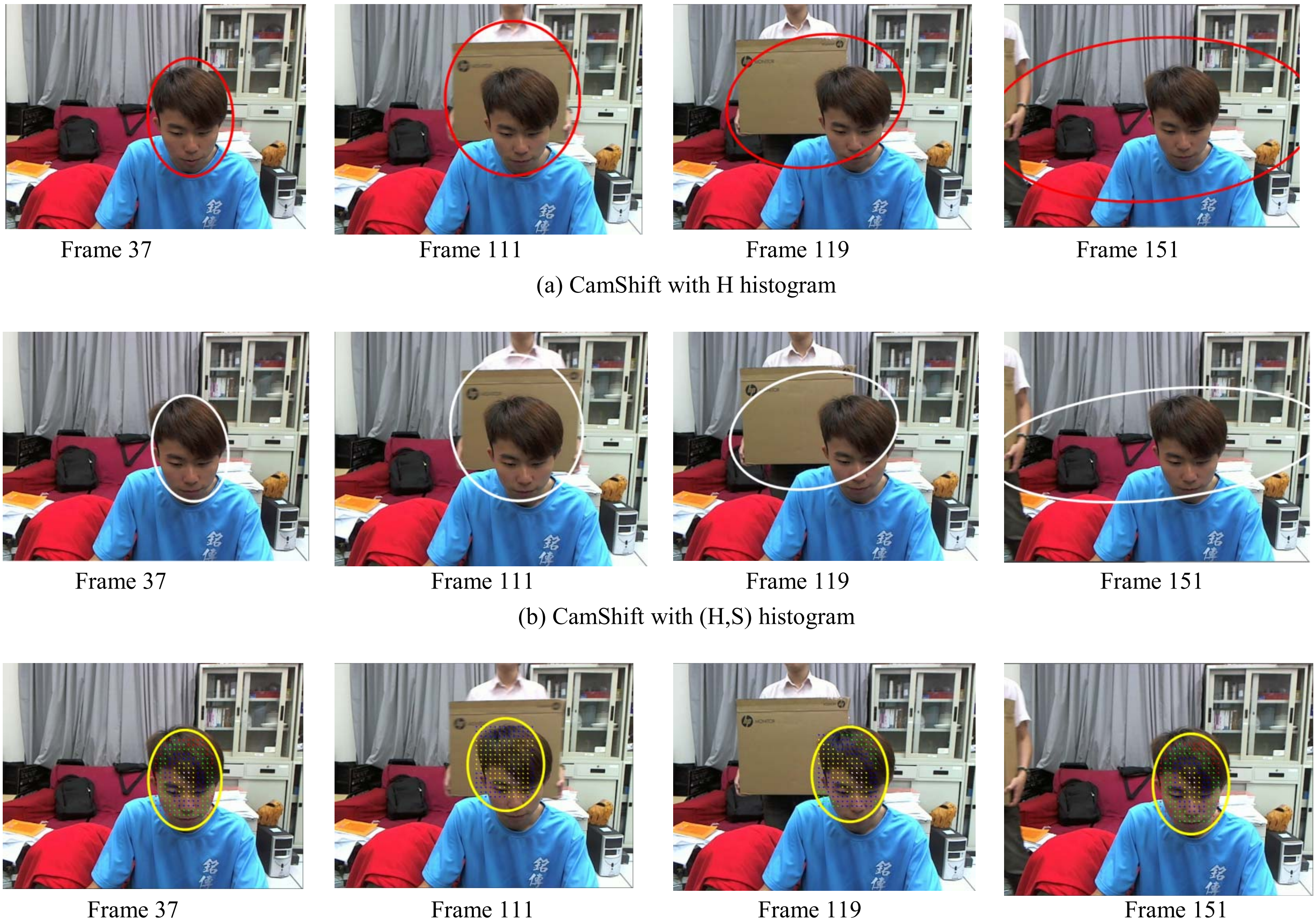
Frame 1 Frame 92 Frame 131 Frame 259

(b) CamShift with (H,S) histogram

Frame 1 Frame 92 Frame 131 Frame 259

(c) Proposed algorithm with H histogram

Figure 7. Comparison of proposed algorithm and conventional CamShift for hand tracking



Frame 37 Frame 111 Frame 119 Frame 151

(a) CamShift with H histogram

Frame 37 Frame 111 Frame 119 Frame 151

(b) CamShift with (H,S) histogram

Frame 37 Frame 111 Frame 119 Frame 151

(c) Proposed algorithm with H histogram

Figure 8. Comparison of proposed algorithm and conventional CamShift for head tracking