Journal of Information Hiding and Multimedia Signal Processing ©2018 ISSN 2073-4212
Ubiquitous International Volume **9**, Number **1**, January **2018**

Visual Object Tracking Based on Color and Implicit Shape Features

Chaur-Heh Hsieh
Department of Computer and Communication Engineering
Ming Chuan University
No. 5, De Ming Rd., Gui Shan District, Taoyuan City, 333, Taiwan
hsiehch@mail.mcu.edu.tw

Mao-Hsiung Hung*
College of Information Science and Engineering
Fujian University of Technology
No.3, Xueyuan Road, University Town, Minhou, Fuzhou, 350118, China
*Corresponding author: mhhung@fjut.edu.cn,

Shiuh-Ku Weng
Department of Information Engineering
Chung Cheng Institute of Technology, National Defense University
No.190, Sanyuan 1st St., Dashi Jen, Taoyuan City, 335, Taiwan
skweng.tw@yahoo.com.tw

Received March, 2017; revised October, 2017

Abstract. *MeanShift and its adaptive version CamShift have received wide attention as an efficient and robust method for object tracking. Both methods try to reach the peak of the probability distribution using gradient ascent approach, and thus they are easily prone to local maxima. This paper presents a novel tracking algorithm based on the CamShift framework. It first removes small local peaks with a dominant color filter to reduce the possibility of the trap to local maxima. Next, our method embeds the color and implicit shape features of the target object into the probability distribution image. Finally, it computes the moments of the probability distribution image frame by frame and then estimates the center and scale of the tracking object accordingly. Experiments on five challenging video databases indicate the proposed algorithm performs significantly better than the conventional CamShift and MeanShift based algorithms.*
Keywords: Object tracking, Meanshift, Camshift, Gradient ascent.

1. 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, intelligent transport system, and advanced intelligent systems for human event analysis, etc. Several challengeable issues on the robustness of object tracking include distraction, illumination changes, occlusions, shadow and reflection, shape and appearance change, etc. [1]-[2]. 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 MeanShift 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]. The basic MeanShift algorithm uses color histogram to model object probability density and moves the object region in the largest gradient direction. Bradski [7] modified the MeanShift and developed the Continuously Adaptive Mean Shift (CamShift) algorithm for face tracking. CamShift is an adaptive version of the MeanShift based on a probability distribution 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 when the object and background (or other objects) have significantly different colors. Two major components in object tracking are target representation and localization [3]. In object tracking, a target is usually defined as a rectangle or an ellipsoidal region in the image. Currently, a widely used target representation is the color histogram because of its RST (rotation, scaling and translation) invariance and its robustness to partial occlusions. The basic MeanShift/CamShift algorithm [3] also employs color histogram for target representation. However, color features often yield drawbacks such as (a) distraction by similar color objects, (b) interference of background with similar colors, (c) illumination change, and (d) occlusion during tracking process.

To attack the drawbacks above, several methods which exploited multiple features have been proposed to improve the basic MeanShift (CamShift) tracker [8]-[15]. Haritaoglu and Flickner [8] integrated color and edge density in the MeanShift tracking framework. Wang and Yagi [9] extended the basic mean-shift tracker to an adaptive tracker by selecting reliable features from color and shape-texture according to their descriptive ability. The shape-texture feature is the histogram of gradient orientation. The authors in [10] presented a modified MeanShift algorithm by replacing color histogram of the basic MeanShift with joint color-texture histogram. The idea is that every pixel in the search window is described by the combination of color feature and the texture feature using local binary pattern (LBP). Ganoun et. al. [11] proposed a modified CamShift by adding feature points descriptors to the object model. Hidayatullah and Konik [12] enhanced CamShift for multi-object tracking. They selected each dominant color object part using a combination of MeanShift segmentation and region growing to achieve more precise object localization. Although the shape of object is also useful to improve the tracking performance, the construction of an explicit shape (contour) model is not easy [16].

The key issue in the MeanShift and CamShift tracking methods is the calculation of probability distribution image of the target candidates. Both methods try to reach the peak of the probability distribution using gradient ascent approach. The common drawback of both algorithms is that they are easily prone to local maxima (peaks) when some of the target features present in the background. This paper presents a novel tracking algorithm based on the CamShift framework. It first removes small local peaks with a dominant color filter derived from the target object to reduce the possibility of the trap to local maxima. In addition, to further increase the robustness, our method embeds the color and implicit shape features of the target object into the probability distribution image. Using this probability distribution, we can compute the moment features and then estimate the object center and scale based on the moments. Quantitative and qualitative measures on five challenging video databases indicate the proposed method performs significantly better than the conventional CamShift and MeanShift based methods.

The organization of this paper is as follows. In Section 2, we first introduced the basic concept and idea of the proposed method and then describe the details of the algorithm. The experiments and evaluations for various algorithms are explained in Section 3. Finally, the conclusions are drawn in Section 4.

2. **Proposed Object Tracking Method.**

2.1. **Concept of Proposed Method.** Our investigation indicates the major problems of MeanShift/CamShift are: (a) trapping in the local peak of probability distribution, and (b) distracted by other larger objects or background. The local peaks may occur where several small-size clutters locate around the target and they have similar color with the target. The problem will yield the localization drift. Even the small peaks are removed, there may still exist larger peaks which have the same probability value with the target. This occurs when other objects which are with color similar to the target but with size larger than the target. The problem will result in the so-called distraction, which means the system tracks to a wrong object for a long time.

The major ideas behind our tracking method include (a) removing local peaks with smaller probability value to avoid trapping in the local maximum; and (b) reducing the probability value of large interference object to solve the distraction problem. Figs.1(a) and 1(b) illustrate the idea. In Fig.1 (a), the probability distribution f(x) has two smaller local peaks at different positions (x=7 and x=38) of the image. First, we design a dominant color filter to remove the local peaks, and the result is shown in Fig.1 (b). The dominant color filter is derived from the color histogram of the target object. Next, we extract implicit shapes of the tracking object and target candidates, which are called object template and candidate maps respectively. Through the matching of the object template and the candidate maps, the probability of the tracking object (at x=26) is enhanced but the probability of the interference object (at x=49) is suppressed, as illustrated in Fig.1 (c).

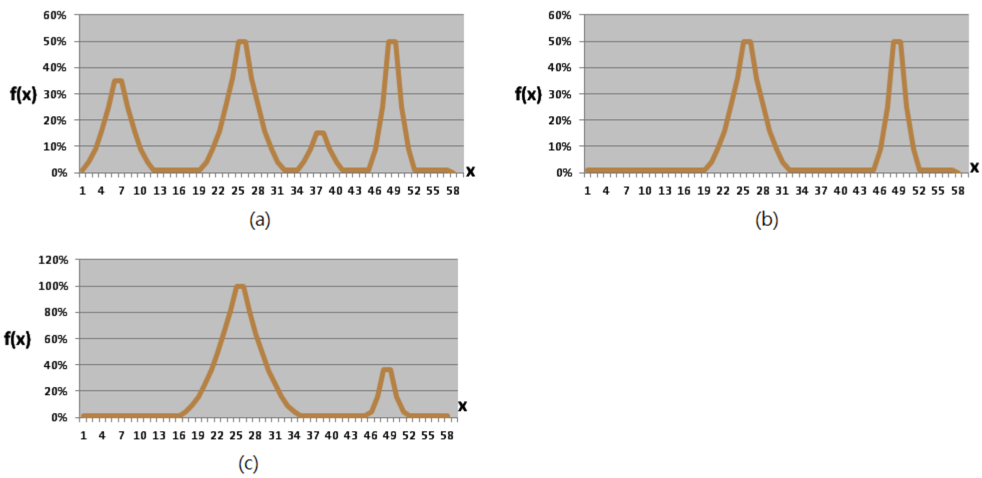
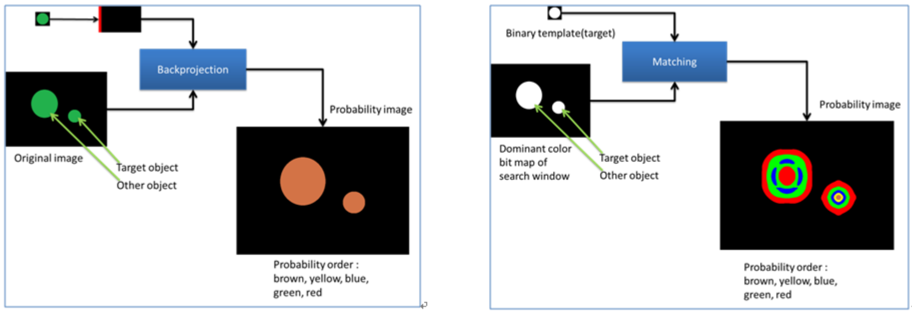


Figure 1. An illustration of how to avoid trapping in local maximum

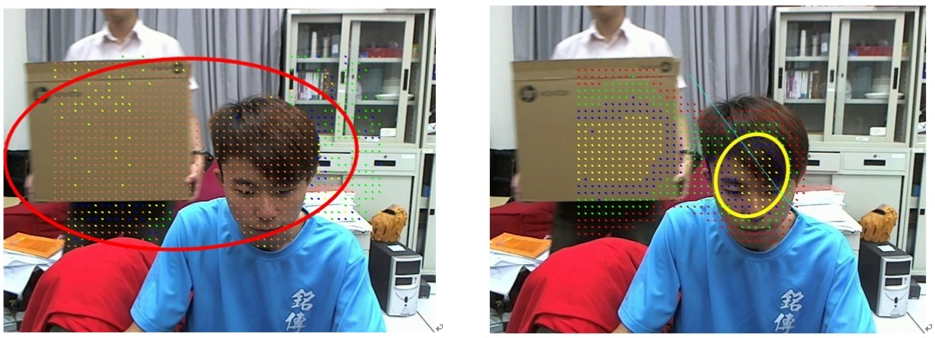
To further to explain the function of the implicit shape, we assume a circle object with green color is selected as tracking target, and an interference object with similar color exists in the image. If only color histogram information is considered, then the probability distribution image obtained with the back projection of the basic CamShift algorithm will be the same as Fig.2(a). It is seen that the target object and interference object have the same probability. However, combining the color and implicit shape, our matching function will yield different probability distributions for target object and interference object, as shown in Fig.2(b). More importantly, the probability value of the target object center (mark with brown color) is greater than that of the interference object (marked with red color), which means the interference of non-tracking object is suppressed. Fig.3(a) and 3(b) compares the probability distribution images in the real tracking cases. The left figure is obtained by the basic CamShift with only color information, and the right one is that by our proposed tracking algorithm combing color and implicit shape information. It is seen that with only color information, the probability distribution is corrupted by the background object, and thus yielding distraction as shown in Fig.3(a). The reason is that the background object, the box, has similar color to the skin color of the tracking face. On the contrary, our proposed algorithm which fuses color and implicit shape can attack the problem, as demonstrated in Fig.3(b).



(a) Color information only (with CamShift)

(b) Fusion of color and implicit shape
(with proposed method)

FIGURE 2. Probability distribution images generated for synthetic case



(a) Color information only (with CamShift)

(b) Fusion of color and implicit shape
(with proposed method)

FIGURE 3. Probability distribution images generated in real case of tracking

Although the probability of the target center is larger than those of other locations, we still cannot assure the centroid of the search window obtained by moments is exactly equal to the center of the target. The reason is stated in the following. Assume the probability distribution image of the search window is $I(x, y)$, the zero-order moment and the first-order moments of the object can be calculated by [7]

$$M_{00} = \sum \sum I(x, y), (x, y) \in R \tag{1}$$

$$M_{10} = \sum \sum xI(x, y), (x, y) \in R \tag{2}$$

$$M_{01} = \sum \sum yI(x, y), (x, y) \in R \tag{3}$$

The centroid of the object is calculated from the above moments as

$$x_c = \frac{M_{10}}{M_{00}}, y_c = \frac{M_{01}}{M_{00}} \tag{4}$$

It is obvious that the centroid is linearly proportional to the first-order moments. When an interference object exists, even its probability value is smaller than the target object, if the non-zero calculation region R is greater than the target area. The first-order moment and the resulting centroid could shift towards the interference object, which results in the localization error. In this work, we design a mapping function which enhances the portion with higher probability but suppresses the one with lower probability to avoid the centroid shift. The design mapping function is shown in Eq.(5) and illustrated in Fig.4.

$$w(k,l) = \begin{cases} 0, & \text{if } p(k,l) \le t_1 \\ q_1, & \text{if } t_1 < p(k,l) \le t_2 \\ q_2, & \text{if } t_2 < p(k,l) \le t_3 \\ \vdots & \\ q_n, & \text{if } t_{n-1} < p(k,l) \le t_n \end{cases} \tag{5}$$

Referring to Fig.4, the t_i, $i = 1, 2, ..., n$, are the decision thresholds, and the mapping output are discrete values with $q_i < q_{i+1}$. In Fig.4, A_i is the areas of the respective decision regions. If the areas satisfy the following condition, the distraction of the interference objects can be avoided.

$$A_i = q_i(t_{i+1} - t_i) \text{ and } \sum_{i=1}^{k-1} A_i < A_k \tag{6}$$

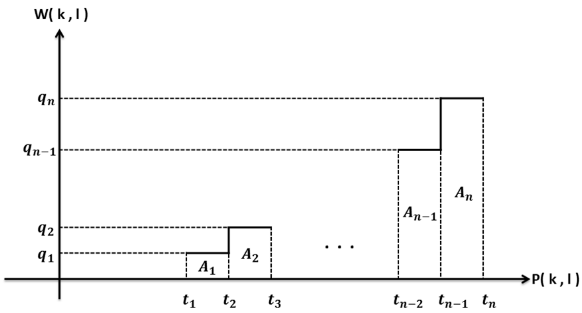


FIGURE 4. An illustration of mapping function

Through the above process, the implicit shape feature is fused with color feature and embedded into the probability distribution image $w(k,l)$. Based on $w(k,l)$, we compute the moment features and then estimate the centroid of the object using gradient ascent iterative algorithm. When the iterative procedure converges, the object location of the current frame and its zero-th moment are obtained. Using the zero-th moment, we update the object window size for the next frame.

2.2. **Proposed Tracking Algorithm.** The block diagram of the proposed tracking algorithm is shown in Fig.5. It consists of two phases: Initial phase and running phase. The initial phase is to generate a binary template of the target object, and a dominant color filter. The binary template preserves the rough shape of the target object, so we refer to it as *implicit shape*. The phase is done only in the first frame of the video sequence. After the first frame, the system goes to the running phase, which estimates the location of the object over time until video frames run out. This phase consists of two main processing units. One is to calculate the probability distribution image of the search window of every frame by employing color and implicit shape feature. The other unit is to find the centroid of the object using gradient ascent algorithm, and then estimate the size of the object window for the next frame. The detailed tracking procedure is described step by step in the following.

2.2.1. *Initial Phase.* **Step 1** Initially, a user manually chooses a target to be tracked with size $w \times h$, 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 greater than one and determined in experiments. In our work, $s = 1.3$.
Step 2 Generate color histogram with N bins using the image pixels within the object window, as illustrated in Fig.6(a). The histogram is further normalized to the range of $0, 1$, which represents the color distribution of the object to be tracked.
Step 3 Design a dominant color filter. Assume the color distribution of the object window is $\{C(c_i, p_i), i = 1, 2, ..., N\}$, where c_i is color bin, p_i is the percentage of the color bin i, and $\sum p_i = 1$. Sort the color bins in an descent order, and then select the M largest bins to form a dominant color set, $\{D(k), k = 1, 2, ..., M\}$, where $M < N$.
Step 4 Generate a binary object template by filtering the images pixels within the object window as

$$O(i,j) = \begin{cases} 1, & \text{if } C(i,j) \in \{D(k), k = 1, 2, ..., M\} \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

where $C(i, j)$ is the color bin of the pixel at (i, j). Note that $O(i, j) = 1$ indicates 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 Fig.6(b).

2.2.2. *Running Phase.* Calculate Probability distribution image of Search Window:

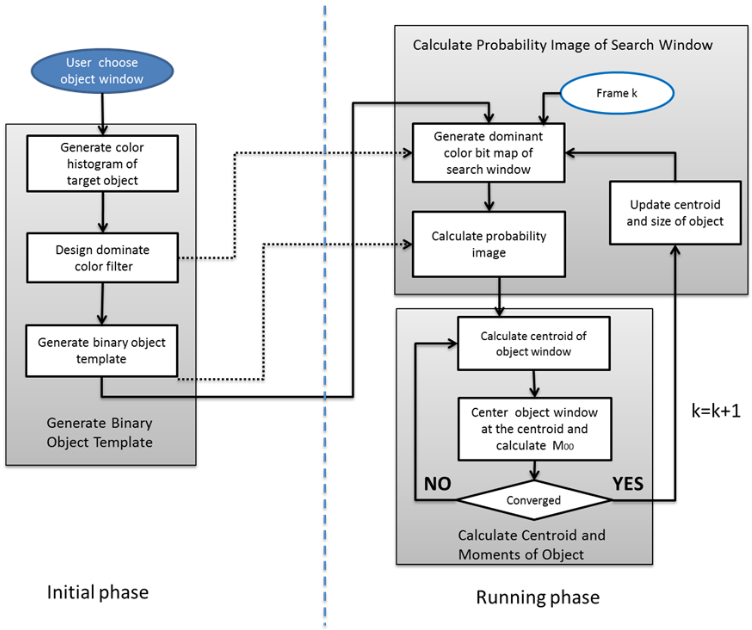


FIGURE 5. The block diagram of the proposed tracking algorithm

FIGURE 6. Color histogram of the target object and tracking object template

This part is to calculate the probability distribution image of the search window for each input frame. The search window is obtained by expanding the object window similar to the Step 1 of the initial phase.

Step 5 Input an image frame, and then generate a dominant color bit map by filtering the image in the search window using the dominant color filter. 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 Fig.7.

FIGURE 7. Dominant color bit map of the search window

Step 6 Calculate a matching score map by matching the object template with each candidate in the dominant color bit map. A candidate is a block with the same size as the object template, i.e., $w \times h$, taken from the dominant color bit map. The adjacent candidates are overlapped in general and the distance of neighboring candidates can be defined in advance. The distance affects the localization accuracy and computational load. It is determined in the experiments which compromises the two performance factors. The matching score between the object template O and a candidate block C is done with a simple binary comparison operation

$$S(O,C) = \sum O(i,j) \oplus C(i,j), \oplus\text{: bit comparison} \tag{8}$$

The matching score of the candidate at (k,l) is obtained by

$$P(k,l) = \frac{S(O,C)}{w \times h} \tag{9}$$

Step 7 For each candidate, we transform its matching score with the mapping function in Eq.(5). Fig.8 shows an example of the probability distribution image 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.

Calculate the Centroid and Size of Object Window:

Step 8 Perform gradient ascent iterative algorithm using the probability distribution image until convergence. After the convergence, we have the centroid and the zeroth moment M_{00} of the object of the current frame.

Step 9 Update the size $(w \times h)$ of the object window with M_{00} by Eq.(10), and then repeat Step 5 to Step 9 for the succeeding frame.

$$w = 2 \times \sqrt{\frac{M_{00}}{256}}, h = h_0 \times \frac{w}{w_0} \tag{10}$$

where h_0 and w_0 are the height and width of the object window initially selected by the user.

3. **Experimental Results.** To evaluate the tracking performance of the proposed method, we applied it on a wide variety of challenging videos in different environments and applications. And the results are compared with the mean-shift related algorithms including the basic MeanShift [3],[4] and CamShift [7], which utilize color feature, and the modified MeanShift [10] which combines color and LBP texture feature. The color features are extracted either from RGB or HSV color space.

3.1. **Video Databases.** We collected five videos, each of which has its particular challenge in object tracking. Video 1 to Video 4 are downloaded from the websites, and the remaining video (Video 5) is captured by the authors. The picture resolution of video 1 to 4 is 720x576, and the video lengths are respectively 680, 910, 179 and 487 frames. The picture resolution of Video 5 is 640x480, and its length is 199.

Video 1 is captured in dark room, where the light is off, but the light is turned on at a particular time. The selected tracking object is the face of a subject. In the background, there are objects with color similar to the human face. Fig.9(a) shows the several samples of this video.

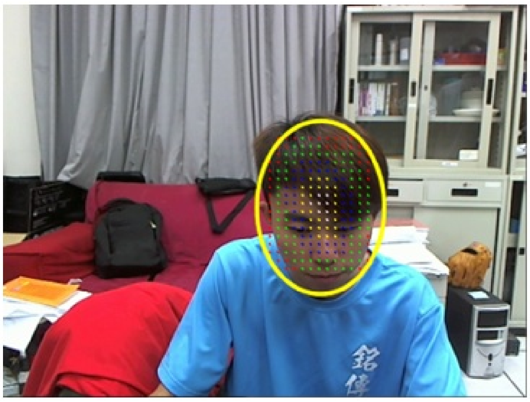


FIGURE 8. An example of probability distribution image

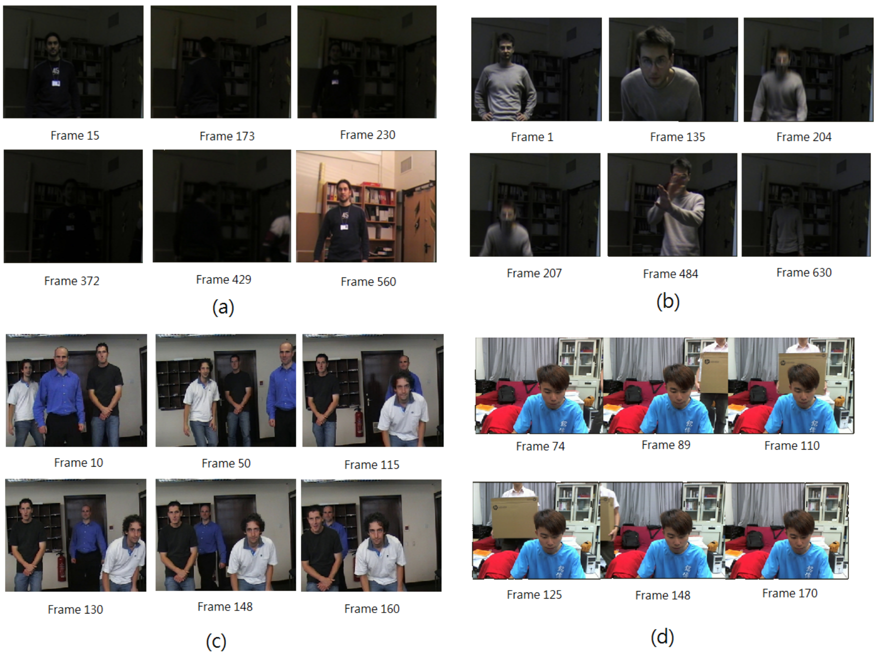


FIGURE 9. Sample frames of Video 1, 2, 3/4 and 5

The Video 2, as shown in Fig.9(b), is also captured in the dark room, but some frames have significant scale changes due to zoom in/out of camera. Around the frame 200, tracking object has a rapid movement, resulting in blur of image frames. Moreover, hands occlude face sometimes.

Videos 3 and 4 are the same video but with different tracking subjects. As demonstrated in Fig.9(c), in the video, there are many subjects moving around and crossing over others some times. In Video 3, the tracking object is the people wearing blue shirt. Several occlusions occur, especially from frame 128 to frame 150. In Video 4, the tracking target is the man wearing white shirt. Few occlusions and many background clutters occur in the test video.

For Video 5, the human head is the tracking target. As shown in Fig.9(d), 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 of a room.

3.2. **Tracking success rate.** Assume a video with length M frames. The actual target region (ground truth) for a frame is represented as a set A, which is obtained manually, and the detected target region is represented as a set B. The tracking overlap rate for a frame is calculated by [17]

$$P_i = \frac{area(A \cap B)}{area(A \cup B)} \tag{11}$$

The value of overlap rate lies between 0 to 1. When P_i is greater than 0.5, the frame is regarded as being successfully tracked.

$$S_i = \begin{cases} 1 \text{ (success)}, & \text{if } P_i \geq 0.5 \\ 0 \text{ (fail)}, & \text{if } P_i < 0.5 \end{cases} \tag{12}$$

We define the average tracking success rate for the video with length M as

$$R_s = \frac{1}{M} \sum_{i=1}^{M} S_i \tag{13}$$

3.3. **Parameters setting.** 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 and computational saving.

3.3.1. *Number of dominant colors.* The dominant color filter is used to segment the object (region) selected by the user from the input color frame. The segmented binary object image should be noiseless. More specifically, the segmented object should have no breakage and no noise is generated in background area. The number of dominant colors M affects the extent of breakage and the amount of background noise. To determine an appropriate value of M, we collect many different color objects, as shown in Fig.10. And then we perform color filtering with different values of M for the color objects, and the results are shown Fig.11. It is seen when $M = 1$, the resulting binary object is not complete (has many breakages). On the contrary, when $M = 3$, many noises are generated in the non-object (background) area. Thus we choose $M = 2$ in our work for the balance of object completeness and background noise.

FIGURE 10. Color object samples for the determination of the number of dominant colors

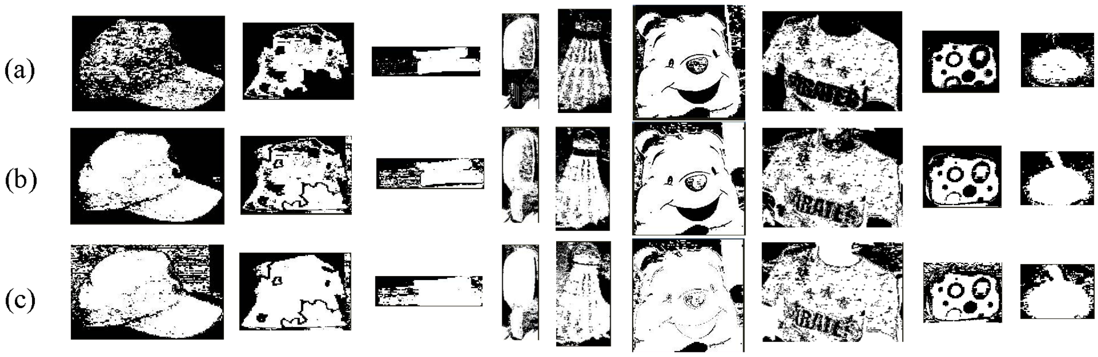
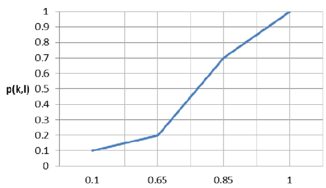
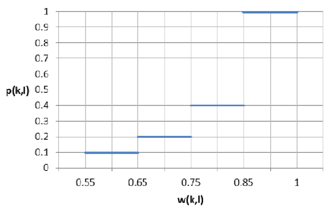


FIGURE 11. Filtered results with different values of M, (a) $M = 1$, (b) $M = 2$, (c) $M = 3$

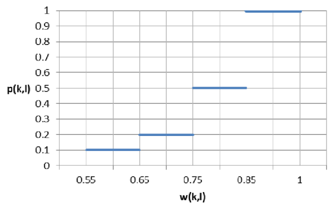
3.3.2. *Probability mapping functions.* We design six discrete probability mapping functions (M2 to M6) according to the constraint in Eq.(6). For comparison, we also design a piecewise linear mapping function (M1). All mapping functions are shown in Fig.12(a) to Fig.12(f), and the corresponding tracking success rates are demonstrated in Table 1. It is seen that the mapping function M5 performs the best, and M2 the worst. For M5, the mapping outputs are $q1 = 0.1$, $q2 = 0.2$, $q3 = 0.7$, $q4 = 1.0$.



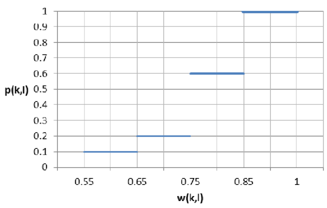
(a) M1(piecewiselinear)



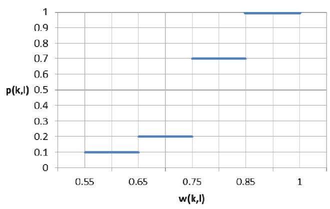
(b) M2 (1241)



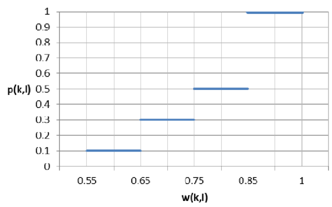
(c) M3 (1251)



(d) M4 (1261)



(e) M5 (1271)



(f) M6 (1351)

FIGURE 12. Mapping functions

TABLE 1. The tracking success rates with various mapping functions

Mapping function	M1	M2	M3	M4	M5	M6
Video 1	0.400	0.214	0.234	0.401	0.419	0.235
Video 2	0.389	0.034	0.132	0.335	0.401	0.510
Video 3	0.733	0.554	0.598	0.640	0.778	0.562
Video 4	0.866	0.985	0.975	0.965	0.953	0.989
Video 5	0.920	0.893	0.884	0.882	0.960	0.879

3.4. **Evaluation of Tracking Performance.** The evaluation of tracking performance in this work includes quantitative and qualitative measures. Qualitative evaluation is performed on visual interpretation, by looking at tracked rectangular masks yielded by the algorithms. On the other hand, quantitative evaluation requires a numeric comparison of computed results with ground truth data. In this work, we employ tracking success rate defined in Session 3.2 for quantitative evaluation, and the results are demonstrated in Table 2 and Fig.13. It is seen that on the average the proposed algorithm performs significantly better than other algorithms; the average tracking success rate of our algorithm is greater than that of the second best with approximate 20%. Since the lighting in either Video 1 or Video 2 is too low most of time and changes dramatically, all algorithms perform poor in these two videos. However, our method is much better than the existing methods except MeanShift(RGB+LBP) in Video 1. It is noted that the existing algorithms perform unstably for various videos. For example, the tracking success rate of MeanShift(HSV) for Video 5 is 100%, but it is very poor for Video 1, 2 and 4. On the contrary, our algorithm is relatively more robust for various videos.

TABLE 2. The tracking success rates with various mapping functions

Methods	Video 1	Video 2	Video 3	Video 4	Video 5	Average
Proposed	0.419	0.401	0.778	0.953	0.960	0.702
CamShift(H) [7]	0.225	0.058	0.517	0.064	0.578	0.288
CamShift(HS) [7]	0.090	0.116	0.561	0.831	0.804	0.480
CamShift(HSV) [7]	0.124	0.272	0.639	0.900	0.874	0.562
MeanShift(RGB) [4]	0.203	0.234	0.877	0.081	1.000	0.479
MeanShift(HSV) [4]	0.097	0.188	0.816	0.074	1.000	0.435
MeanShift(RGB+LBP) [10]	0.469	0.368	0.134	0.231	0.197	0.28
MeanShift(HSV+LBP) [10]	0.054	0.053	0.011	0.137	0.518	0.155

Fig.14 shows the qualitative evaluation of eight tracking schemes for the five videos. For Video 1, as shown in Fig.14(a), the CamShift algorithm based on H, HS and HSV colors generate poor results. The other algorithms also lost track in some frames. However, the proposed algorithm performs well. For Video 2, it is seen that only our algorithm and MeanShift based on HSV color and LBP texture features [10] perform relatively well in tracking. However, our algorithm can automatically adjust the tracking object window to adapt the scale change of the object, as shown in Fig.14(b).

Fig.14(c) shows the tracking results for some frames of Video 3. The predefined tracking object is the man wearing blue shirt. The crossing over of the tracking object with the man wearing white shirt makes the tracking challengeable. It is seen that CamShift-based algorithms and MeanShift(RGB+LBP), MeanShift(HSV+LBP) tracked the wrong object in some frames. Only our proposed algorithm, MeanShift(RGB) and MeanShift(HSV)

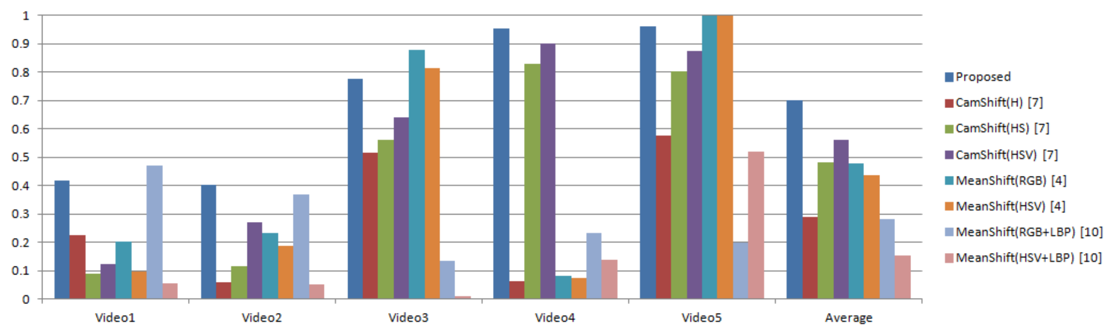


FIGURE 13. Tracking success rates of various tracking algorithms

achieve correct tracking. Fig.14(d) demonstrates the tracking results for some frames of Video 4. The tracking object is the man wearing white shirt. Only our algorithm and CamShift(HSV) achieve correct tracking. Fig.14(e) shows the scenario that a man who carries a box with color similar to tracking human head is moving from the one side to the other side. As shown in Row(2) to (4) of Fig.14(e), with CamShift, the tracking procedure does not converge for some frames, yielding very large tracking window and thus reducing tracking accuracy significantly. Nevertheless, our algorithm and MeanShift-based algorithms performs well except the MeanShift(RGB+LBP).

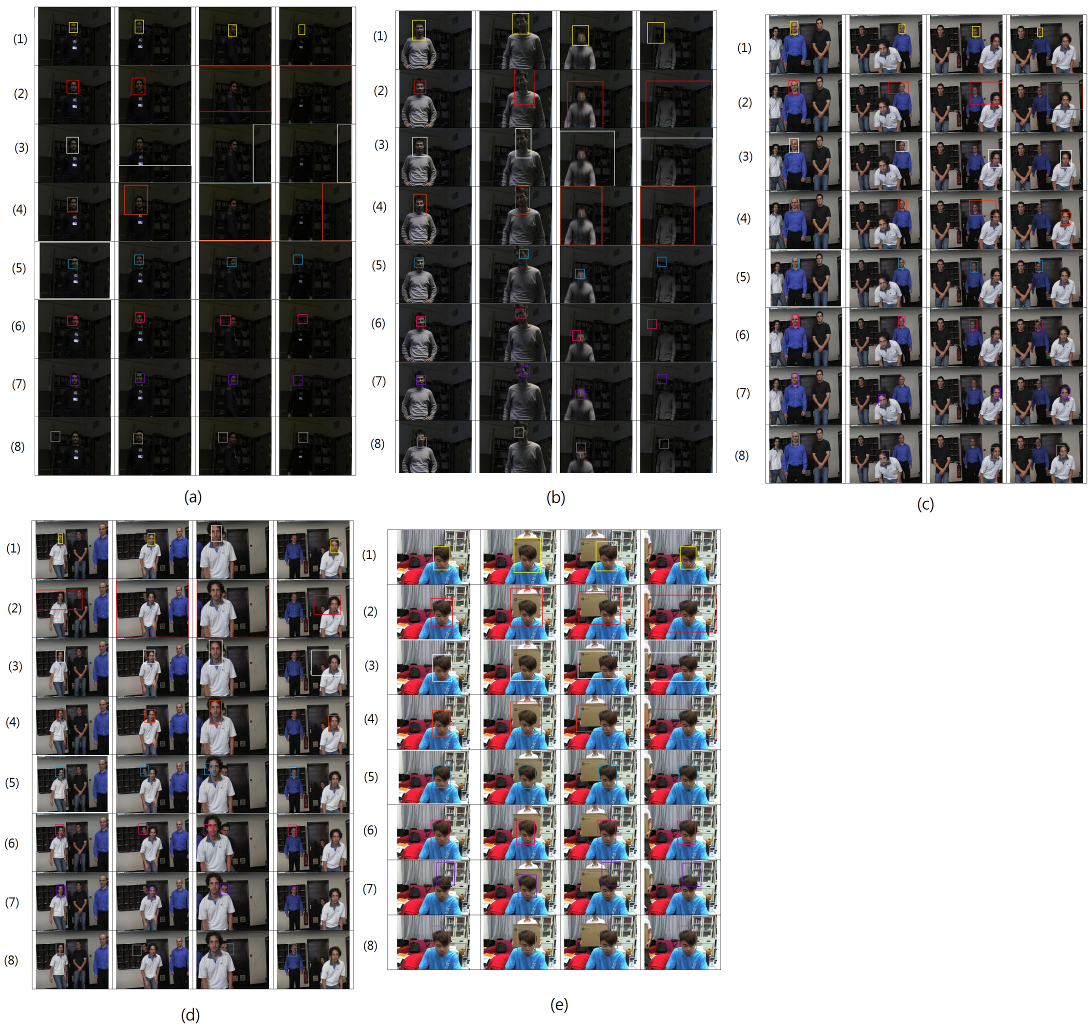


FIGURE 14. Tracking results of Video 1, 2, 3, 4 and 5 by various methods. Top to bottom rows: (1)Proposed(H), (2)CamShift(H), (3)CamShift(HS), (4)CamShift(HSV), (5)MeanShift(RGB), (6)MeanShift(HSV), (7)MeanShift(RGB+LBP), (8)MeanShift(HSV+LBP)

4. **Conclusions.** This paper has presented a robust real-time tracking algorithm based on CamShift framework. It derives the probability distribution image of a particular frame based on color and implicit shape features. The implicit shape information of the object is extracted with a novel dominant color filter function, which is derived from color histogram of the target to be tracked. The dominant color filter also reduces the possibility of the trap to a local maximum of probability distribution. Experimental results on various challenging video sequences indicate our proposed algorithm offers better robustness than

the conventional CamShift and MeanShift-based related algorithms. The robustness to significant rotation and illumination change will be further investigated in the future.

Acknowledgment. This work was supported in part by National Science Counsel, Taiwan Granted MOST 102-2221-E-130-019 and MOST 103-2221-E-130-006, and in part by Fujian Provincial Department of Science and Technology, China Granted No.2017J01729.

REFERENCES

[1] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Survey,* vol. 38, no. 4, pp.1-5, Dec. 2006.

[2] A. S. Jalal, and V. Singh, "The state-of-the-art in visual object tracking," *Informatica,* vol. 36, pp.227–248, 2012.

[3] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," *IEEE Proc. on Computer Vision and Pattern Recognition*, 2000, pp.673–678.

[4] D. Comaniciu, V. Ramesh, and P. Meer , "Kernel-based object tracking," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, May 2003, pp.564–577.

[5] L. Ido, L. Michael, and R. Ehud, "Mean Shift tracking with multiple reference color histograms," *Computer Vision and Image Understanding*, 2010, pp.400–408.

[6] C. Shen, J. Kim, and H. Wang, "Generalized Kernel-Based Visual Tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 1, Jan 2010, pp.119–130.

[7] G. R. Bradski, "Computer vision face tracking for use in a perceptual user interface," *Intel Technology Journal*, 1998.

[8] I. Haritaoglu and M. Flickner, "Detection and tracking of shopping groups in stores," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, pp.431–438, 2000.

[9] J. Wang, and Y. Yagi, "Integrating color and shape-texture features for adaptive real-time object tracking," *IEEE Transactions on Image Processing,* vol. 17, no. 2, pp.235–240, Feb 2008.

[10] J. Ning, L. Zhang, D. Zhang, C. Wu, "Robust object tracking using joint color-texture histogram," *International Journal of Pattern Recognition and Artificial Intelligence*, 23(7), pp.1245–1263, 2009.

[11] A. Ganoun, N. Ould-Dris, and R. Canals, "Tracking system using CamShift and feature points," *14th European Signal Processing Conference (EUSIPCO 2006)*, Florence, Italy, 2006.

[12] P. Hidayatullah, H. Konik, "CAMSHIFT improvement on multi-Hue and multi-object tracking," *2011 International Conference on Electrical Engineering and Informatics*, July 2011.

[13] N. Cai, N. Zhu, W. Guo, W. K. Ling, H. Wang, and Q. Dai, "Object tracking using mean shift for adaptive weighted-sum histograms," *Circuits, Systems and Signal Processing*, vol. 33, no. 2, pp.483–499, 2014.

[14] S. W. Chou, C. H. Hsieh, B. J. Hwang, H. W. Chen, "A modified Mean Shift algorithm for visual object tracking," *IEEE Asia Pacific Signal and Information Processing Association Annual Summit and Conference*, Oct.2013.

[15] N. Cai, et al., "Object tracking using mean shift for adaptive weighted-sum histograms," *Circuits System Signal Process*, vol. 33, no. 2, pp 483-499, Feb. 2014.

[16] M. Isard and A. Blake, "Condensation-conditional density propagation for visual tracking," *Int? J. Computer Vision*, vol. 29, no. 1, 1998.

[17] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "Pascal Visual Object Classes Challenge Results," 2010.

[18] L. Li, W. Huang, I. Gu, Q. Tian, "Statistical modeling of complex background for foreground object detection," *IEEE Transactions on Image Processing*, vol. 13, no. 11, pp.1459–1472, 2004.