Robust Visual Object Tracking with Extended CAMShift in Complex Environments

Lae-Kyoung Lee, Su-Yong An, and Se-Young Oh
Department of Electrical and Electronic Engineering
Pohang University of Science and Technology (POSTECH)
Pohang, Gyeongbuk, Korea
{laeklee, grasshop, syoh}@postech.ac.kr

Abstract*—This paper presents a new approach to solve the problem of real-time robust object tracking in complex environments. Generally, traditional CAMShift (Continuous Adaptive Mean Shift) provides speed and robustness for visual tracking, but it will become unstable when similar objects are presented in background or occlusion happens. In this paper, we proposed an advanced two level approach towards these problems with improved back-projection and kalman filter based occlusion handling. The lower level of the approach implements the multidimensional color histogram and the combination of color and motion information based improved histogram back-projection process. The higher level of the approach implements kalman filter based occlusion handling process with CAMShift. With this method, our proposed method robustly tracks the target object under rapid illumination changes, highly similar-colored background, and occlusion handling condition in real-time with high accuracy. The experimental results show that the proposed algorithm is robust and efficient to track the various object in indoor/outdoor complex environments

I. Introduction

Visual object tracking is one of the most important research subjects in the fields of pattern recognition and computer vision research, which has an important applied value in the fields of intelligent video surveillance, advance human-computer interaction and autonomous mobile robot application. Although many approaches exist in tracking an object, it still remains difficult to track an object under complex environments such as highly similar-colored background, illumination variation, shape deformation and object occlusion.

To deal with these problems based on high accuracy and real-time processing, many approaches such as Mean Shift [1], CAMShift [2], and Sequence Karhunen Loeve particle filter [3] are proposed. Moreover, the combinations of different approaches [4],[5],[6], such as Mean Shift and kalman filter [7], Mean Shift embedded particle filter [8],[9], have provided reliable solutions for objects tracking. Mean Shift algorithm widely used for real-time object tracking, which is a robust non-parametric method of finding local maximum in the direction of gradient on fixed, static density distribution and use Bhattacharyya coefficient as a measure of comparability. CAMShift was first proposed by G.Bradski to perform efficient head and face tracking in a perceptual user interface. It is based on an adaptation of Mean Shift that, given a probability density image, finds the mean of the distribution by iterating in the direction of maximum increase in probability density. The primary difference between CAMShift and Mean Shift algorithm is that CAMShift uses continuously adaptive probability distributions (that is, distributions that may be recomputed for each frame) while MeanShift is based on static distributions, which are not updated unless the target experiences significant changes in shape, size or color. Although CAMShift provides speed and robustness for visual tracking, it will become unstable when similar objects are presented in background or occlusion happens. In addition, the width of tracking window increases dramatically, sometimes even leading to loss of target object.

In this paper, we propose on robust tracking method (Extended CAMShift) that jointly employs multi-cue based CAMShift and kalman filter. Comparing with the existing works, the main contribution of this research is two-fold. First, most of the tracking approaches suffer from the problem of highly similar-colored background/objects and appearance changes within various illuminations. This paper uses multi-dimensional histogram and improved back-projection method using weighted combination of color and optical flow [10] based motion histogram to achieve robust tracking with a reduced computation cost. Second, we adapt the optimal estimation theory based on the linear stochastic kalman filter algorithm for making it more robust object identification with occlusion, as has been shown in the experiments.

This paper is organized as follows. In section 2 and 3, we deal with the original CAMShift algorithm and our proposed extended CAMShift algorithm. Section 4 describes the validity of our proposed method with various experimental results, performance evaluations and comparisons. Finally, the main conclusions are drawn in Section 5.

II. CAMShift Algorithm

CAMShift algorithm is a computer vision color tracking algorithm which belongs to moving object region based tracking techniques. It is able to track in real time yet not absorb a major share of computational resources. The basic idea of CAMShift is based on robust nonparametric for climbing density gradients to find the mode (peak) of probability distributions called the mean shift algorithm to compute all frames in image sequence, iteratively taking the results of the previous frame as the initial values of Mean Shift

978-1-61284-972-0/11/$26.00 ©2011 IEEE

TABLE I
THE PROCESSES OF CAMSHIFT ALGORITHM

Original CAMShift Algorithm
1 : Input RGB image from image sequence.
2 : Initialize the size and location of the search window.
3 : Convert color space RGB to HSV.
4 : **while** (*# of iteration*) **do**
5 : Calculate the color probability distribution of hue component within the search window.
6 : Find the location of the centroid of the search window.
7 : Run Mean-Shift algorithm, and get access to the new size and location of the search window.
8 : In the next frame of sequence images, initialize the size and location Of the search window with the values obtained in step 7.
9 : **endwhile**
10 : Go to step 3.

algorithm search window of current frame. The overall process of the original CAMShift is depicts in Table I.

On the basis of the dynamic probability distribution of image colors, this algorithm converts input images into color probability distribution maps after establishing the color probability model for tracking targets. Then, after initializing a rectangular searching window in the first frame image, the size and location of the window can be adaptively adjusted according to the relative velocity and acceleration of the moving targets.

III. PROPOSED EXTENDED CAMSHIFT ALGORITHM

In this section, we describe real-time extensions to the original CAMShift algorithm for making it more robust against similar object colors and rapid illumination changes, and for enabling object identification with occlusion handling (Fig. 1). Comparing against original CAMShift (which does not incorporate any of these extensions) will make the full benefit of our approach clear.

A. Extended CAMShift with Multi-dimensional histogram and Improved Back-Projection Algorithm

Generally the original CAMShift tracking algorithm has many advantages, such as real-time, robust, self-adaptive window etc, the original method only used H-component in HSV color space is obviously not enough to express object's color. For example, if the color of object is similar to that background color, it's easily to lose the tracking object. Based on to achieve the real-time tracking robustness, we proposed the multi-dimensional histogram with Hue and Saturation

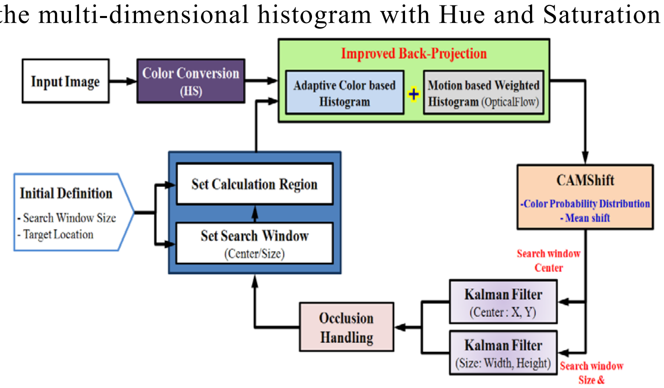

Fig. 1 Proposed extended CAMShift flow chart.

TABLE II
THE IMPROVED BACK-PROJECTION WITH COLOR-MOTION INFORMATION

Improved Back-projection Algorithm with Color-Motion Info.
1 : Input RGB image from image sequence.
2 : Convert color space RGB to HSV.
3 : Calculate the HS color probabilistic distribution map $\mathbf{P}_c(x_i, t)$ through back-projection.
4 : Calculate the global motion flow based local motion probabilistic distribution map $\mathbf{P}_m(x_i, t)$ through the proposed motion detection.
4.1. Extract a set of feature points inside previous tracking window with offset using Lucas-Kanade optical flow.
4.2. Extract the valid feature points within current global flow angle range.
4.3. Generate the motion boundary rectangle region for extracted feature points.
4.4. Create the gaussian distribution kernel for current motion region.
4.5. Perform the convolution of current $\mathbf{P}_c(x_i, t)$ by created gaussian distribution kernel to make the motion based PDM, $\mathbf{P}_m(x_i, t)$.
5 : Integrate the two probabilistic distribution map using linear weighted summation using (3).
6 : Generate the final histogram back-projection for current image.

component in HSV model. According to the character of human vision system, our proposed solution is simple, but effective. We utilize an arbitrary fixed number of histograms to model different appearances of target objects. For every appearance condition that strongly varies in color, we pre-compute and store one re-scaled ([0; 1]) reference histogram. In addition, we sum all reference histograms that belong to the same object, and re-scale the result back to [0; 1]. This accumulated histogram is then used for computing the probability back-projection of the corresponding object during run-time, while the individual reference histograms are only applied for object identification. Furthermore, in most cases, the histogram of target object is influenced by illumination changes, cluttered backgrounds and large geometric deformations, therefore it is very important to use statistic algorithm with additional capabilities for robust tracking.

In this paper we deal with these problems by proposing the improved back-projection method which contains multiple feature distributions including the color distribution from current single image and the motion distribution from two or more consecutive image frames using Lucas-Kanade optical flow and global motion flow based motion detection method [11],[12] (Table II, Fig. 2). To generate the motion based probabilistic distribution map (PDM), we firstly determine the global motion flow from current local motion vectors (Fig. 2c) and then create the 2-dimensional gaussian distribution kernel (Fig. 2g) with detected motion boundary region, as in (1) and (2).

$$\boldsymbol{X} = [x_1, x_2]^T, \quad \mu = [\mu_1, \mu_2]^T, \quad \Sigma = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}, \tag{1}$$

$$\mathbf{G}(\mathbf{x}; \mu, \Sigma) = \frac{1}{2\pi\sigma_1\sigma_2} \mathbf{e}^{\frac{-\{(x_1-\mu_1)^2+(x_2-\mu_2)^2\}}{2\sigma_1^2\sigma_2^2}} \tag{2}$$

Thus according to the above procedure, the motion based PDM (Fig. 2d) is generated by result of the convolution between multi-dimensional color histogram based PDM and created gaussian motion kernel. Finally, in order to obtain the integrated PDM with color and motion histogram back-projection, we apply the weighted linear combination method

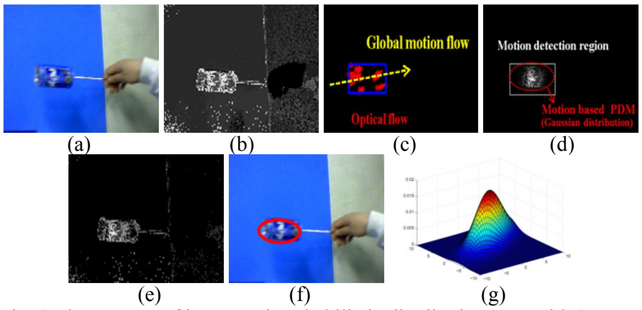

Fig. 2 The process of integrated probabilistic distribution map with 2-dimensional gaussian distribution over two variable x_1 and x_2.
(a) Input RGB image, (b) HS color model based back-projection, (c) Local motion vectors with LK Optical flow (red line) and global motion flow (yellow dot line) calculation within tracked window, (d) Motion based PDM within motion boundary region (white rectangle) using 2d gaussian distribution kernel, (e) Finally integrated back-projection result, (f) Target object tracking result (red ellipse), (g) 2-dimensional gaussian distribution kernel graph

using (3). Let $\tau \in (0, 1)$ is the integration coefficient in (3), and then the integrated new histogram back-projection $\hat{\mathbf{P}}_t$ for current image should be

$$\hat{\mathbf{P}}_t = (1-\tau) * \mathbf{P}_c + \tau \mathbf{P}_m \tag{3}$$

The integration coefficient τ serves as a parameter that controls the rate of the adaptation of the target model to the candidate model. Using this process, it robustly maintains an accurate tracking performance over changes in illumination. The above Fig. 2 shows motion based PDM generation with 2-dimensional gaussian distribution kernel and final integrated back-projection result.

B. Combining Kalman Filtering with Exteneded CAMShift Algorithm

CAMShift is a tracking algorithm based on external feature and track non-rigid objects in real-time. But when tracking multiple objects in complex background, if the color distribution of the target model is similar between objects, kalman filter [13] is used as an optimal estimator that predicts and corrects the states of linear processes. After each CAMShift optimization that gives the measured location and size of the target object, the uncertainty of the estimate can also be computed and then followed by the kalman iteration, which drives the predicated position and size of the target object. The uncertainty is determined by image noise, the similarity between target object colors, clutter colors and the percentage of occlusion. As such, the equations for the kalman filter fall into two groups: time update equations and measurement update equations. The update equations are responsible for projecting forward the current state and error covariance estimates to obtain the priori estimates for the next time step. The correct equations are responsible for the feedback, in other words, for incorporating a new measurement into the prior estimate to obtain an improved posterior estimate. In our case, we use two kalman filter models simultaneously which define the state vector $\mathbf{x}_t$ as 4-dimension vector, and the measurement vector $\mathbf{z}_k$ consists of the location of centroid point of object region and the size of tracking window in (4);

TABLE III
KALMAN FILTER WITH EXTENDED CAMSHIFT

Kalman filter with Extended CAMShift
Given State vector $\mathbf{x}_t = (x(t), y(t), v_x(t), v_y(t))^T$
$\begin{cases} \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k) + \mathbf{w}_k, \ \mathbf{w}_k \sim N(\mathbf{w}_k; \mathbf{0}, \mathbf{Q}_k) \\ \mathbf{z}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k, \ \mathbf{v}_k \sim N(\mathbf{v}_k; \mathbf{0}, \mathbf{R}_k) \end{cases},$
1 : Initialization
$\hat{\mathbf{x}}_0 = E[\mathbf{x}_0]$
$\mathbf{P}_0 = E[(\mathbf{x}_0 - \hat{\mathbf{x}}_0)(\mathbf{x}_0 - \hat{\mathbf{x}}_0)^T]$
2 : Time Update (Prediction)
$\hat{\mathbf{x}}_k^- = \mathbf{F}_{k\vert k-1} \hat{\mathbf{x}}_{k-1}$
$\mathbf{P}_k^- = \mathbf{F}_{k\vert k-1} \mathbf{P}_{k-1} \mathbf{F}_{k\vert k-1}{}^T + \mathbf{Q}_k$
$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1}$
3 : Measurement Update (Estimation)
$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (z_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-)$
$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_k^-$
$\mathbf{F}_k = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{H}_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$
4 : Repeat step 2 – step 3

$$\mathbf{x}_t = (x(t), y(t), v_x(t), v_y(t))^T, \ \mathbf{z}_t = (x(t), y(t))^T, \tag{4}$$

where $(x(t), y(t))$ refers to x, y coordinate component of the object central position or window size and $(v_x(t), v_y(t))$ is the velocity component of the object in the t^{th} image frame.

Given state vectors, over all process of kalman filter with extended CAMShift is described in Table III. F_k is the transition matrix with associated noise Bu and Q that is the Gaussian process noise with zero mean. And H_k represents a measurement matrix with associated noise R that is the error between real and detected location of the object, K_k is Kalman gain. Process noise represents the accuracy of the model and is determined empirically and the measurement noise is derived directly from the off-line calibration test where an estimate of $\hat{\mathbf{x}}_{k-1}$ and $\mathbf{P}_{k-1}$ is initialized. We obtain object trajectory by taking the correspondences of detected object between successive image frames.

C. The processing of Occlusion Problem Handling

When the target is occluded for some time, it may be lost due to the deficiency of the CAMShift algorithm, so an occlusion handling processing is necessary to realize continuous tracking. In order to accurately detect whether the tracking objects are obscured by large areas, it is a good approach to analyze the difference between the estimated value of the approximate location of the tracked objects by kalman filtering and the observed value of the tracking objects by our proposed algorithm. Such a difference is called filter's residual, and is calculated by the following equation.

$$r(k) = \sqrt{(x(k) - \hat{x}(k))^2 + (y(k) - \hat{y}(k))^2}, \tag{5}$$

where $\hat{x}(k)$ and $\hat{y}(k)$ represent the currently estimated value of the tracked object, whereas $x(k)$ and $y(k)$ stand for the observed value of the tracked object in the current image. In order to judge the occlusion using (5), we choose a threshold value α. If $\boldsymbol{r(k)} > \alpha$, kalman filter stop working. At this time,

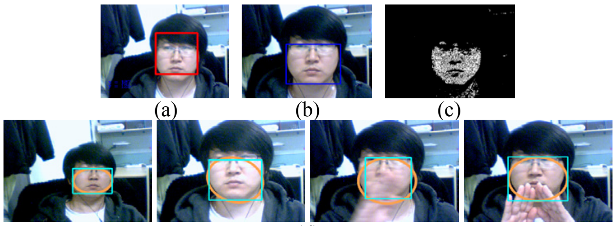

Fig. 3 Proposed extended CAMSHIFT tracking under occlusion.
(a) Detected face regions, (b) Face tracking with detected color histogram (blue rectangle : face tracking window), (c) Back-projection image for tracked face, (d) Proposed method based face tracking result

the starting point of the next frame's target is predicting by using the center position of the previous frame target. Then, target's real position of the current frame is searched by using CAMShift, at the same time, still calculating the value of $\boldsymbol{r(k)}$, compared with α, if $\boldsymbol{r(k)} < \alpha$, kalman filter restarts in the next frame. If the occlusion is caused by a moving object with a different color, the width of the tracking window will shrink even to zero, so we define a threshold W_{min}, if the width of the tracking window bellows W_{min}; it is believed that an occlusion is happening. If the occlusion is caused by a moving object with the same or similar color of the target, the two objects will be considered as one by the algorithm when they begin to overlap. In this case, the width of the tracking window will step up to a larger value. So we define a threshold ΔW, an occlusion is believed happen when $W_{t+1}/W_t > \Delta W$, if an occlusion is detected, a number N_f is set by the algorithm, in the next N_f frames the camshaft algorithm is not called for searching the real location of the target, but use the prediction of kalman filter as the real position and it is used to update the state of the kalman filter. When the N_{f+1} frame comes, the CAMShift algorithm is called to search the target and the result is considered to be the real position and used to update the state of the kalman filter again. Fig.3 represents the extended CAMShift based face tracking result with occlusion handling.

IV. Experimental Results

The proposed tracking method was tested on dynamic sequential images with various illumination and complex background environment. The proposed method took approximately max. 30~35 fps on an Intel Core2 Quad-core 2.3GHz processor without optimizing the code and was applied with low-end camera operating at a standard resolution of 320×240 pixels under Microsoft Visual C++ 8.0 Compiler.

The tracker was initialized manually by placing a rectangle region in the first image. The tracking performance is represented by Squared Root Euclidean distance (SRED) and Normalized Euclidean distance (NED) between the ground truth center ($\mathbf{c}$) of the object and the estimated location of the object center ($\hat{\mathbf{c}}$). The definition of SRED and NED distance are described by

$$SRED(c,\hat{c}) = \sqrt{(c_x - \hat{c}_x)^2 + (c_y - \hat{c}_y)^2}, \tag{6}$$

$$NED(c,\hat{c}) = \sqrt{\frac{(c_x - \hat{c}_x)^2}{h_x} + \frac{(c_y - \hat{c}_y)^2}{h_y}}, \tag{7}$$

where h_x and h_y are the tracked object ellipse dimension. This implies that if minimum NED($\mathbf{c}$, $\hat{\mathbf{c}}$), then the estimated ellipse center $\hat{\mathbf{c}}$ is inside the ground truth region. In the following sections, we illustrate several representative experiments with real-world video sequences.

A. Static Object Tracking under Rapid Illumination Changes

To evaluate the proposed algorithm, we recorded image sequences with sudden global illumination change with similar colored background. Fig. 4a shows example images from the image sequences with average intensity value. The rapidly varying illumination conditions are created by randomly turning on/off different light sources and opening/closing blinds (4 times changes, max. intensity difference = 171). Fig. 4b shows the results of tracking a static object under rapidly illumination changes. Based on the results, the propose method could track the target object robustly over all sequences. On the other hand, original CAMShift methods (with 8bins, 30bins) continually fail to track the object during illumination changes and show the inaccurate target location and size estimation. Fig. 4c shows the SRED and NED result between the ground truth and the estimated center and Table IV lists the average SRED and NED values between two algorithms on the same data set. The position and size errors of original method were the highest during illumination change, but our proposed method consistently low error value during all over frames. From this result, our proposed method is robust to global illumination changes without additional learning method.

B. Moving Object Tracking with various deformable and illumination condition

1) Moving Vehicle Tracking in outdoor environment

For various object tracking tasks, we examined moving vehicle tracking experiment on the real-time video sequences of 601 frames with changing illumination and background condition in outdoor environment. The following Fig. 5a shows proposed method based moving vehicle tracking results. By combination of the motion and color based weighted back-projection, we could obtain more precise and stable vehicle tracking result which reliably deals with lighting changes and long-term scene changes in outdoor environment (Fig. 5b). In contrast, original CAMShift methods (with 8bins and 30bins) performed the inaccurate tracking results for the position and size estimation of target vehicle and even failed to track properly for the rest of the video sequences. For example, it is clearly shown in Fig.5b where the tracking result for original method is only plotted to 113th frame, for after which the tracker diverges.

2) Moving Object Tracking with various deformable and similar-colored background condition

In this case, we carried out deformable object tracking performance comparisons under the local illumination and the object's appearance changes within highly similar-colored background condition. For the fixed parameters condition (original method: 30bins, proposed method: 8bins for each

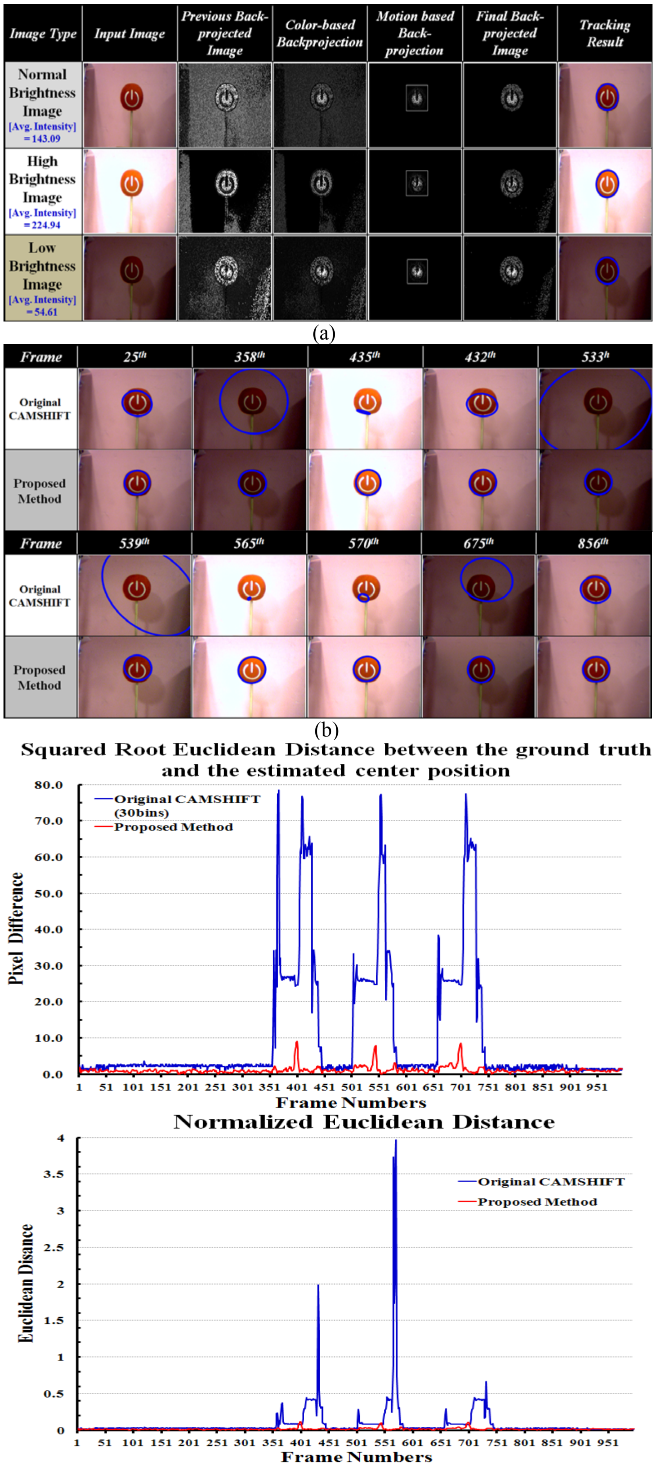

Fig. 4 The tracking performance comparison with sudden global illumination changes for static object image sequences.
(a) Tracking results with representative images with average intensity value, (b) Object tracking results over a sequence frames under sudden illumination changes, (c) The SRED and NED performance graph.

color channel, same init target model and back-projection integration rate (τ = 0.5)), The stability and performance of each method were analyzed by using SRED and NED. As shown in Fig.6, our proposed method could reliably track the object with higher tracking accuracy and robust performance

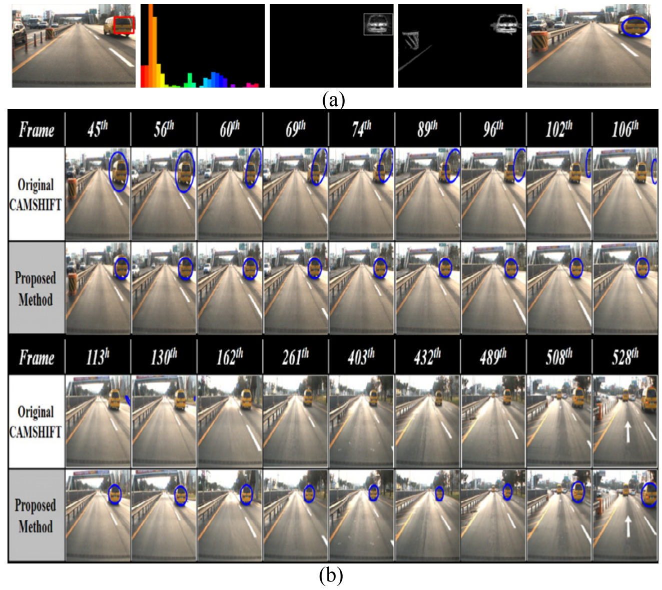

Fig. 5 Moving vehicle tracking result in outdoor environment.
(a) Main results of the proposed tracking method (1st column: Initial object tracking region, 2nd column: object's color histogram image, 3rd column: motion based weighted back-projection result, 4th column: totally integrated histogram back-projection result, 5th column: final tracking result), (b) Vehicle tracking process results over a sequence frames.

performance, while original CAMShift which use only single color histogram model obtained incorrect estimation results or lost the target object's position and size throughout the tracking sequence due to the similar colored background area (Fig. 6b, Table IV). Under similar condition, we executed the additional test to check the tracking performance under abrupt changes of the lightning conditions and surface reflection noise. In this time we chose a shiny metallic object which has a similar color distribution to that of the background. During object movement, the surface of object such as a metallic component had regions that are very smooth or flat, the reflection of light from the surface tends to be specular (Fig. 7). Even in such a situation, our proposed method still shows robust tracking performance with precise position and size estimation.

C. *Object tracking with Occlusion Handling*

This experiment shows the occlusion handling result which caused by a static and dynamic object with a highly similar color. The 1st experiment was designed to compare the temporal partial/total occlusion handling result (Fig. 8a). The initial target face is totally occluded and moving hand with a similar hue to the face disturbs the tracking. Like the tracking result in Fig. 8, when the moving hands passed through the target face, the original CAMShift lose the face and tracked the hand instead. But our proposed method couldn't be disturbed and continuously maintained correct identities during occlusion. The 2nd experimental result shows the successful hand tracking during total occlusion (Fig. 8b).

D. *Performance Comparison and Discussion*

In this section, we evaluate the performance of proposed method in terms of tracking accuracy and computational time

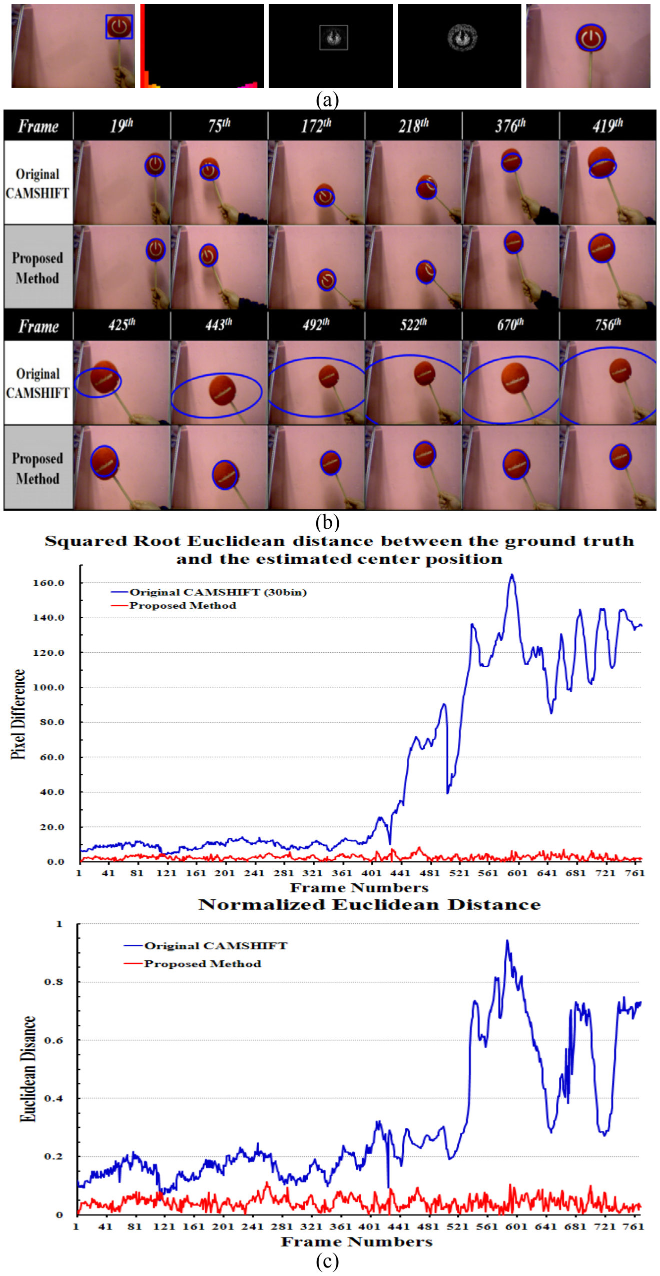

Fig. 6 The tracking performance comparison with local illumination and appearance changes for Moving object #1 image sequences.
(a) Main results of the proposed tracking method, (b) Object tracking process results over a sequence frames, (c) SRED and NED performance graph.

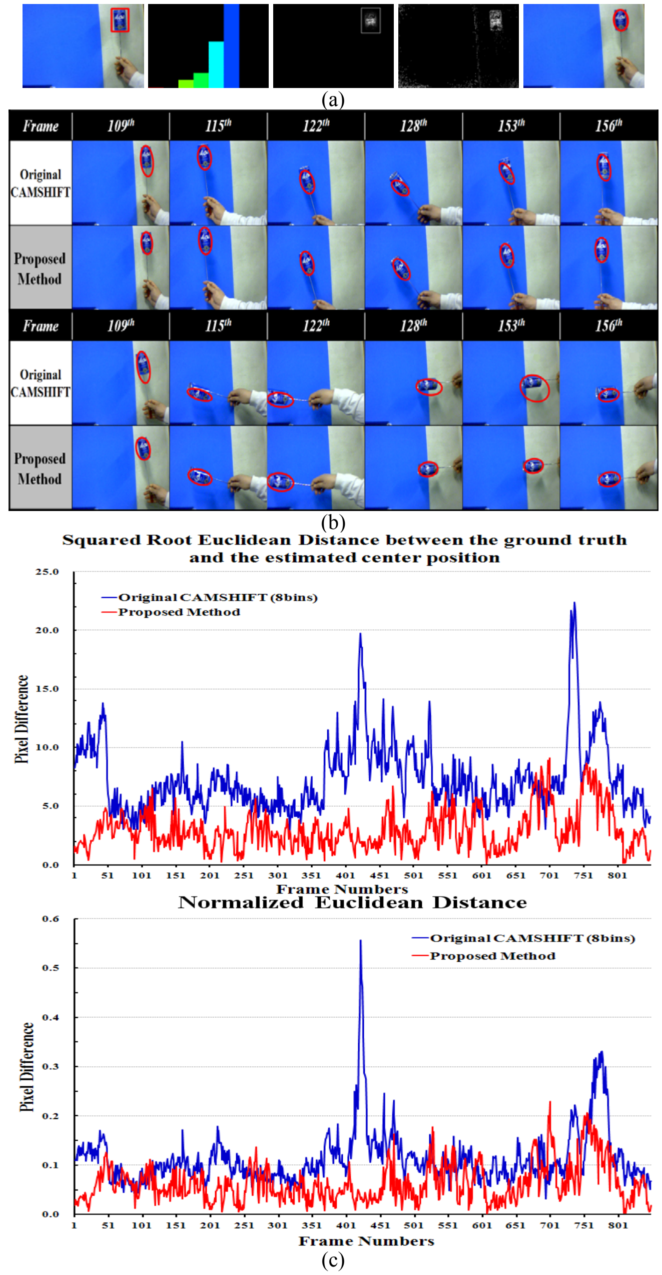

Fig. 7 The tracking performance comparison with local illumination and light reflection noise for Moving object #2 image sequences.
(a) Main results of the proposed tracking method (b) Object tracking process results over a sequence frames, (c) SRED and NED performance graph.

with the ground truth of each experimental image sequences. Firstly, we compared the performance of tracking accuracy in various conditions (static vs. moving object, global vs. local illumination change). The following Table IV shows the performance comparison of average SRED and NED value for each image sequences. Generally, the number of bins in color histograms is a crucial parameter. Too many bins in a histogram do not cope with changes in illumination or in the model appearance and most of the time the algorithm drifts away from the target. On the opposite, too few bins do not allow a good discrimination of the target, and the tracking fails.

For more precise tracking result in original method, we assigned to 30 bins in histogram, but for our proposed method, we kept fixed number of bins (8bins for 1D color histograms) in most of the experiments. Despite using a smaller number of bins for our proposed method, we observed significant performance improvements in terms of tracking accuracy than original method in all conditions. Additionally, to see the effectiveness of our proposed method, we conducted the performance comparison between our proposed methods under different parameter condition (Table V). Given static object with global illumination change sequences, the data shows that

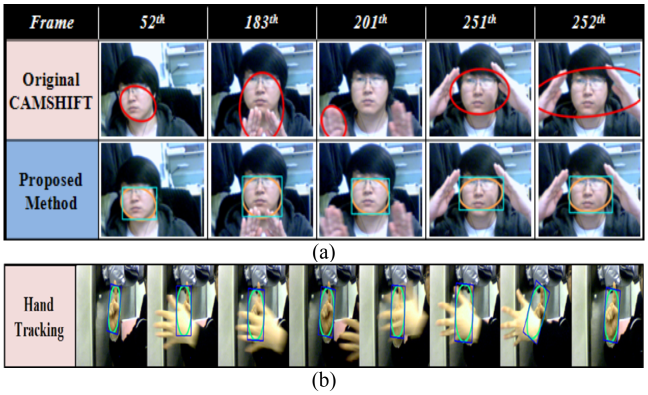

Fig. 8 The tracking performance comparison with occlusion handling. (a), (b) Face and Hand tracking result from sequential images.

there was a significant improvement effects by increasing the number of color histogram bins for adaptive color only model (except for motion based weighted histogram). However, our proposed method with combination of color and motion histogram achieves the most apparent accuracy without increasing the number of bins. Consequentially, in order to further improve the tracking quality, the usage of motion based histogram has been proved to be more efficient than simply increasing the number of histogram bins.

Next, we compared the performance of average computation time between original and proposed method. During tracking, the average processing time of the original CAMShift method (the order of complexity: $\mathbf{O}(\alpha N^2)$, α = constant, $N \times N$ = image size) has a mean value of 3.4*ms* and a standard deviation of 0.6*ms,* while our proposed method has a mean value of 28.6*ms* and a standard deviation of 1.5*ms*. Especially in our method, Optical flow [14] computation time may constitute 40~60% of the total processing time (the order of complexity: $\mathbf{O}(n^2N + n^3)$, n = # warp parameters, N = integration window size). Although proposed method has relatively more computation time, but it also guarantees the real-time processing over 30 fps and its performance are more precise tracking result than others. So we sufficiently conclude that our proposed method is an efficient and robust tracking algorithm.

TABLE IV. OBJECT TRACKING EXPERIMENTAL RESULTS

Experimental Image Data	Original Method		Proposed Method		Remark (No. of Bins)	
	SRED	NED	SRED	NED	Original Method	Proposed Method
Static Object / Global Illumination Changes	10.307	0.088	**1.093**	**0.014**	30	8
Moving Object / Local Illum. Changes # 1	52.854	0.794	**2.752**	**0.044**	30	8
Moving Object / Local Illum. Changes # 2	7.367 (6.80)	0.118 (0.11)	**2.949**	**0.063**	8 (30)	8

TABLE V. PROPOSED METHOD BASED OBJECT TRACKING PERFORMANCE COMPARISON

Experimental Image Data	8 bins, Color + Motion		8 bins, Color Only		30 bins, Color Only	
	SRED	NED	SRED	NED	SRED	NED
Static Object /Global Illumination Changes	**1.093**	**0.015**	28.49	0.093	5.583	0.071

V. CONCLUSION

In this paper, we have proposed the robust visual object tracking algorithm based on extended CAMShift for solving complex background and occlusion problems. By the combination of multi-dimensional color model and optical flow based global motion model, our proposed algorithm improves the histogram back-projection and tracking performance with reduced computation cost under complex background condition (rapidly illumination change, similar-colored background and etc.). Additionally, the proposed extended CAMShift tracking has advantages in rejecting static background objects of similar color without background training and subtraction, and in solving the problem of occlusion among target objects at same time. The experimental results have demonstrated the robustness and efficiency of our algorithm in real-time object tracking under various environment conditions.

ACKNOWLEDGMENT

This research was supported by the MKE (The Ministry of Knowledge Economy), Korea, under the Core Technology Development for Breakthrough of Robot Vision Research support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2011-C7000-1101-0006).

REFERENCES

[1] D. Comaniciu, V. Ramesh, and P. Meer, “Kernel-based object tracking,” IEEE Trans. on Pattern Analysis and Machine Intelligence, 25(5):564-577, 2003.

[2] G. Bradski, “Computer vision face tracking as a component of a perceptual user interface,” in Workshop on App. of Comp. Vision, Princeton, NJ, 1998.

[3] Levy, A., Lindenbaum, M. “Sequential Karhunen-Loeve basis extraction and its application to image,” IEEE Tran. Image Processing 9, 1371-1374, 2000.

[4] A. Yilmaz, “Object tracking by asymmetric kernel mean shift with automatic scale and orientation selection,” in Proc. IEEE Int'l. Conferrence Computer Vision PatternRecognition, Jun. 2007, pp. 1-6.

[5] C. Shen, M. Brooks and A. Hengel, “Fast Global Kernel Density Mode Seeking with Application to Localisation and Tracking,” IEEE Int'l. Conference on Computer Vision, 2005.

[6] Jwu-Sheng Hu, Tzung-Min Su, Chung-Wei Juan, and George Wang, “3D Object Tracking Using Mean-Shift and Similarity-Based Aspect-Graph Modeling,” The 33rd Annual Conference of the IEEE Industrial Electronics Society (IECON 2008), pp. 2383-2388, 2007.

[7] D. Comaniciu, V. Ramesh, “Mean Shift and optimal prediction for efficient object tracking,” Proc. of the IEEE Int'l Conferrence on Image Processing (ICIP), pp.70-73, 2000.

[8] E. Maggio and A. Cavallaro, “Hybrid Particle Filter and Mean Shift tracker with adaptive transition model,” in Proc. of Acoustics, Speech, and Signal Processing (ICASSP), pp. 221-224, 2005.

[9] Shan, C., Tan, T., Wei, Y. “Real time hand tracking using a mean shift embedded particle filter,” Pattern Recognition, 1958-1970, 2007.

[10] B.K.P. Horn and B.G. Schunck, “Determining Optical Flow,” Artificial Intelligence, 17, 1981, 185-204.

[11] B. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” In Proc. of the Int’l. Joint Conference on Artifcial Intelligence. pp. 674-679, 1981.

[12] J. Shi, C. Tomasi, “Good features to track,” in: IEEE Int'l Conference on Computer Vision and Pattern Recognition, pp. 593-600, 1994.

[13] Applied Optimal Estimation, A. Gelb, ed. MIT Press, 1992.

[14] S. Baker and I. Matthews, “Lucas-kande 20 years on: A unifying framework: Part 1,” Technical Report CMU-RI-TR-02-16, Carnegie Mellon University Robotics Institute, 2002.