Journal of King Saud University – Computer and Information Sciences 30 (2018) 416–429

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

CLBP for scale and orientation adaptive mean shift tracking

Oumaima Sliti [a,*], Habib Hamam [b], Hamid Amiri [a]

[a] National Engineering School of Tunisia, Tunis Al Manar 1002, Tunisia
[b] Department of Electrical engineering at University of Moncton, NB, Canada

ARTICLE INFO

Article history:
Received 14 February 2017
Revised 7 May 2017
Accepted 7 May 2017
Available online 19 May 2017

Keywords:
Tracking
Motion
Texture

ABSTRACT

In this paper, we address the problem of tracking an object with pose and appearance changes, under possible occlusions by presenting an effective way to embed the texture information provided by the Local Binary Pattern (LBP), Local Ternary Pattern (LTP) and the Complete Local Binary Pattern (CLBP) in the mean shift framework. We combine the information of color distribution with variants of Local Binary Pattern texture for the purpose of robust tracking. Four adaptive scale and orientation mean shift trackers are proposed; the LBP_MS, LTP_MS, CLBP_MS1 and the CLBP_MS2. The last tracker can handle both textured and non-textured objects, and deals with the specific weaknesses of motion trackers, such as failures under specific conditions. As it can exploit more of the image information, we use seven public videos that contain a variety of challenges to illustrate the accuracy of the proposed approaches. The trackers successfully cope with fast moving objects, target scale and orientation changes, and prove to be more stable and less prone to drift away from the target than purely colored or feature-based ones.

1. Introduction

As tracking applications become pervasive in our daily lives, considerable research has focused on video based motion tracking to minimize human intervention, Sridhar et al. (2015), Sliti et al. (2014a), Du et al. (2013) and Wu et al., 2015. This topic attracts the interest of many fields in the industry, and is necessary in the development of driving assistance systems and many household technologies, Zhang et al. (2014) and Henriques et al. (2015). All tracking applications require high accuracy, especially in the presence of a dynamic background, and it should be performed in a way, which deals with occlusion, cluttering, object merging or splitting and other complicated events related to the moving targets. Besides, the algorithms should also overcome the limitations of scaling problems when the object expands or shrinks its scale.

The tracking methods must have a low computational complexity, because, the tracking systems are often devoted for high-level tasks, such as semantic interpretation and target recognition. In order to deal with all these problems, the intuitive idea is to choose a tradeoff between the target model in the previous frame and the target candidate corresponding to the current object appearance.

Essentially, the mean shift algorithm meets our requirements since it optimally estimates the target information from current measurements and previous states. It is employed in the tracking algorithm by measuring the similarity based on the Bhattacharyya coefficient of the target model and the target candidate in consecutive frames, Singh and Mishra (2011). The mean shift process is based on finding local maxima of the density function from a given discrete data sample by sweeping a search window over the frame. In fact, there are plenty of approaches that employ the mean shift algorithm in object tracking systems. The representation of the target model and candidate are mainly based on color and are described in each frame by its color histogram, Ju et al. (2010) and Ning et al. (2012).

Indeed, in order to cope with different circumstances, we should take advantage of multiple image properties, as none of them alone provides invariance from different imaging conditions. We suggest joining a variety of Local Binary Pattern (LBP and LTP) texture to the color feature in order to enhance the target representation within the adaptive scale and orientation mean shift framework. Thereafter, we propose to use the Complete Local Binary Pattern (CLBP) texture proposed in Guo et al. (2010), within two schemes; the first proposed method called CLBP_MS1 describes the target by using only the CLBP operator, (the color

* Corresponding author.
E-mail addresses: ou.sliti@gmail.com (O. Sliti), habib.hamam@umoncton.ca (H. Amiri).
Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.jksuci.2017.05.003
1319-1578/

information in the mean shift framework is replaced), while in the CLBP_MS2 representation we propose to combine the Complete Local Binary Pattern (CLBP) texture and color features. In this paper CLBP_MS2 is presented under the mean shift framework and proves to be a robust scale and orientation adaptive mean shift tracking algorithm. In fact, by using these two properties of color and texture the performance of our trackers is enhanced and becomes more robust against the large variation of data common in sequences acquired under unconstrained conditions.

This paper is organized as follows: in Section 2, we review some of the most influential tracking algorithms based on the mean shift framework. Section 3 reviews the adaptive scale and orientation-based mean shift tracker, while Section 4, describes the proposed methods: LBP_MS, LTP_MS, CLBP_MS1 and CLBP_MS2, using the new feature extraction embedded into the mean shift based tracking system. Experimental results and discussion are presented in Section 5 followed by the conclusion in Section 6.

2. Related work

There are a large number of object tracking methods relying on the mean shift algorithm. Comaniciu et al. (2000) were the first to handle the tracking of non-rigid objects from a moving camera by employing mean shift algorithms. The performance of such algorithms depends on the separability of the target against the background. To overcome this problem, the Corrected Background-Weighted Histogram (CBWH) proposed by Ning et al. (2012), succeeded in correcting the BWH approach of Comaniciu et al. (2003), by reducing the prominent influence of background features in the target candidate calculation. But unfortunately the CBWH cannot handle changes in color and illumination.

In order to reduce the interference caused by the lighting changes in the mean shift tracking algorithm, Ju et al. (2010) describes a fuzzy color histogram given by a self-constructing fuzzy cluster where the number of color bins generated by the proposed method depends on the target image. Besides, the mean shift algorithm was modified to deal with the changing color probability distributions derived from the frame. This improved version of mean shift was called CAMShift (Continously Adaptive Mean Shift), Martin et al. (2010). Unlike the mean shift that deals only with statistic distributions when searching for a window with a fixed size, it uses a dynamical search window that continuously adapts its size (i.e. after every video frame, depending on the size of the target object). Nevertheless, it supplies robust results only in cases where there are accurate boundaries that clearly distinguish the target from the background, and when its colors differ significantly.

Thereafter, Stolkin et al. (2008) proposed a new color based tracking algorithm named ABCshift (the Adaptive Background CAMSHIFT) tracker which models the background using Bayesian probability method. Afterward, Zivkovic et al. (2004) developed the EM_shift and succeeded in simultaneously estimating the covariance matrix that describes the approximate shape and the position of the local mode. Among this wide variety of approaches, our work is inspired by the proposed scale and orientation adaptive mean shift tracking (SOAMST) algorithm, developed by Ning et al. (2012) regarding the estimation of the scale and orientation of the target under the mean shift tracking framework. This tracker has proved to be efficient in solving the problem of how to estimate the scale and orientation changes of the object to track. In fact, EM_shift and SOAMST use only color information to describe the model, and in general, the performance of such algorithms relies heavily on the target representation.

Yet, the lack of spatial information can cause a loss of the target position under certain conditions, such as similar occlusion and variation of lightness. Consequently, a more discriminating representation of the target is the first step toward an accurate tracking; thus, our proposed trackers describe the object using two weighted feature spaces of color and texture, and they benefit of their complementarities to ensure a perfect estimation of its position, scale and orientation changes of the target.

3. Conventional mean shift algorithm

The scale-orientation adaptive mean shift algorithm is a robust technique in tracking applications. It is summarized in two components:

3.1. Target representation

In this section, we review the target representation using the color histogram in the classical mean shift tracker, Comaniciu et al. (2003). The target model $\hat{q}_u$of the object being tracked is defined as:

$$\begin{cases} \hat{q} = \{\hat{q}_u\}_{u=1\ldots m} \\ \hat{q}_u = C\sum_{i=1}^{n} k(||x_i^*||^2)\delta[b(x_i^*) - u] \end{cases} \quad (1)$$

Likewise the target candidate model $\hat{p}_y$ of the candidate region is defined bellow:

$$\begin{cases} \hat{p} = \{\hat{p}_u(y)\}_{u=1\ldots m} \\ \hat{p}_u(y) = C_h\sum_{i=1} n_h k\left(||\frac{y-x_i}{h}||^2\right)\delta[b(x_i) - u] \end{cases} \quad (2)$$

With $\hat{q}_u$and $\hat{p}_u(y)$being the probabilities of feature u in $\hat{q}$and target candidate $\hat{p}(y)$respectively, δrepresents the Kronecker delta function, m is the number of feature spaces, $\{x_i^*\}_{i=1\ldots n}$ presents in (1) the normalized pixel positions in the target region centered at the original position, and describes in (2) the pixel location in the target candidate region centered at y in the current frame, and $b(x_i^*)$ associates the pixel x_i^* to the histogram bin. Finally, the constant C and C_h are a normalization function defined respectively as:

$$C = \frac{1}{\sum_{i=1}^{n} k(||x_i^*||^2)}, \quad C_h = \frac{1}{\sum_{i=1}^{n_h} k(||\frac{y-x_i}{h}||^2)} \quad (3)$$

Note that $k(x)$ is an isotropic kernel and it attributes smaller weights to pixels distant from the center. The kernel $k(x)$ has a monotonic and convex decreasing profile, and it is defined as a function: $k : [0, \infty) \rightarrow R$with $k(x) = k||x||^2$, Comaniciu et al. (2000). The similarity between the target model and the target candidate is computed using the Bhattacharyya coefficient which defines the correspondence between the two normalized histograms $\hat{q}_u$and $\hat{p}_u(y)$:

$$\rho[\hat{p}_u(y), \hat{q}_u] = \sum_{u=1}^{m}\sqrt{\hat{p}_u(y)\hat{q}_u} \quad (4)$$

The scale-orientation adaptive mean-shift tracking algorithm employs the Bhattacharyya coefficient to estimate the target position, scale and orientation. This similarity function takes high values (close to 1) when the color distribution in the target region (in the previous frame) is close to the one of the candidate region (in the current frame). So, if the object and its background have significantly different color distributions, as the Bhattacharrya coefficient increases, the target candidate is more and more likely to capture the actual object in the current frame, Comaniciu et al. (2003). Consequently, the distance between the target model and the candidate model is:

$$d[\hat{p}(y),\hat{q}] = \sqrt{1-\rho[\hat{p}(y),\hat{q}]} \tag{5}$$

3.2. Target localization

The minimization of the distance (5) between $\hat{p}(y)$and $\hat{q}$ requires the maximization of the Bhattacharyya coefficient (4); thus, Taylor expansion is used to linearly approximate (4) as follows, Comaniciu et al. (2000):

$$\rho[\hat{p}(y),\hat{q}] \approx \frac{1}{2}\sum_{u=1}^{m}\sqrt{\hat{p}_u(y_0)\hat{q}_u} + \frac{1}{2}C_h\sum_{i=1}n_h\omega_i k(||\frac{y-x_i}{h}||^2) \tag{6}$$

where y_0 is the position in the previous frame, and ω_i is calculated by:

$$\omega_i = \sum_{u=1}^{m}\sqrt{\frac{\hat{q}_u}{\hat{p}_u(y_0)}}\delta[b(x_i^*)-u] \tag{7}$$

The weight value given by (7) of a pixel in the target candidate region describes the probability that it belongs to the target. Thus, the sum of the weights of all pixels (i.e. the zeroth-order moment) could be considered as the weighted area of the object to track in the target candidate region, Ning et al. (2012) and Singh et al. (2012). M_{00} is an estimate of the target region and is formally given by:

$$M_{00} = \sum_{i=1}^{n}\omega(x_i) \tag{8}$$

Whenever the weights from the target $\omega(x_i)$ become bigger, the estimation error will be significant and vice versa. According to Ning et al. (2012), Bhattacharyya coefficient could be used to adjust M_{00} in estimating the target region, by proposing the equation below:

$$A = c(\rho)M_{00} \tag{9}$$

With $c(\rho)$ as a monotonically increasing function. It is used to shrink M_{00} back to the real target scale and defined as:

$$c(\rho) = exp\left(\frac{\rho-1}{\sigma}\right) \tag{10}$$

With $(0 \leqslant \rho \leqslant 1)$ and the optimal value of σ set between 1 and 2, Ning et al. (2012). If ρ takes approximately the upper bound 1, it means that the target candidate model approaches the target model. If it is closer to 0, the candidate model is not identical to the target one, as $c(\rho)$ becomes very small and M_{00} enlarges until it exceeds the target region. Suppose the coordinate of each pixel in the candidate region is $(x_{i,1}, x_{i,2})$, the moments of the candidate region become, Ning et al. (2012):

$$M_{10} = \sum_{i}^{n_h}\omega_i x_{i,1}, M_{01} = um_i^{n_h}\omega_i x_{i,2} \tag{11}$$

$$M_{20} = \sum_{i}^{n_h}\omega_i x_{i,1}^2, \quad M_{02} = \sum_{i}^{n_h}\omega_i x_{i,2}^2 \tag{12}$$

By considering (8) and supposing $(\bar{x}_1, \bar{x}_2)$ is the centroid of the target candidate region, the new location of the target y_1 would be defined by:

$$y_1 = (\bar{x}_1, \bar{x}_2) = \left(\frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}}\right) \tag{13}$$

The ratio of the first-order moment to the zeroth-order moment y_1 describes the position, while the second-order center moment describes the shape and orientation of the target. Eq. (7) can be rewritten into the second-order center moment using (8), (11) and (12):

$$\mu_{20} = \frac{M_{20}}{M_{00}} - \bar{x}_1^2, \quad \mu_{11} = \frac{M_{11}}{M_{00}} - \bar{x}_1\bar{x}_2, \quad \mu_{02} = \frac{M_{02}}{M_{00}} - \bar{x}_1^2 \tag{14}$$

To estimate the scale and orientation of the object to track, (14) is converted to the covariance matrix:

$$Cov = \begin{pmatrix} \mu_{20} & \mu_{11} \\ \mu_{11} & \mu_{02} \end{pmatrix} \tag{15}$$

Using singular value decomposition (SVD), (15) becomes:

$$Cov = U \times S \times T^T = \begin{pmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{pmatrix} \times \begin{pmatrix} \lambda_1^2 & 0 \\ 0 & \lambda_2^2 \end{pmatrix} \times \begin{pmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{pmatrix}^T \tag{16}$$

With $U = \begin{pmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{pmatrix}, S = \begin{pmatrix} \lambda_1^2 & 0 \\ 0 & \lambda_2^2 \end{pmatrix}$, λ_1^2 and λ_2^2 as the eigenvalues of Cov, the two vector $(u11, u21)^T$ and $(u12, u22)^T$ are the two main axes of the target in the target candidate area. The target is represented by an ellipse and the lengths of its semi-minor axis and semi-major axis are denoted by a and b respectively:

$$a = \sqrt{\frac{\lambda_1 A}{\pi\lambda_2}}, \quad b = \sqrt{\frac{\lambda_2 A}{\pi\lambda_1}} \tag{17}$$

After estimating the location, scale and orientation of the target in the current frame, the following covariance matrix is represented to define the size of the target candidate in the next frame:

$$Cov_2 = U \times \begin{pmatrix} (a+\Delta d)^2 & 0 \\ 0 & (a+\Delta d)^2 \end{pmatrix} \times U^T \tag{18}$$

With Δd is employed to increment the target candidate region in the next frame. The initial position of the target candidate region is given by the following ellipse region:

$$(x-y_1) \times Cov_2^{-1} \times (x-y_1)^T \leqslant 1 \tag{19}$$

4. Proposed method

Several mean-shift tracking algorithms take the color information into account for the representation of the target. Although time-effective, they usually fail when non-rigid objects are considered, Ning et al. (2012). Mean-shift based on the color discrimination of the target can be affected by partial occlusions, especially when the object intended to be tracked and its respective background have the same color. Besides, nowadays most cameras use color filters to adjust their white balance and perform some color corrections based on the global image intensities, which can dramatically affect accuracy.

To solve these problems, various types of extensions have been proposed and a large number of authors have improved the procedure either by combining mean-shift with local approaches, Sliti et al. (2014b) or by introducing an object/background classification, Aliabadian et al. (2012), in order to deal with severe occlusions.

Using variants of the LBP texture, we introduce in this section four scale-orientation adaptive Mean-Shift tracking algorithms: LBP_MS, LTP_MS, CLBP_MS1 and CLBP_MS2. We present a careful combination of color and texture features and show that this enhances the accuracy of tracking non-rigid objects in highly dynamic and noisy scenes. The features used for texture analysis are the traditional LBP, Ojala et al. (1996), the local ternary pattern LTP, Tan and Triggs (2010), and the Complete Local Binary Pattern (CLBP), Singh et al. (2012). Our hypothesis is that by integrating those texture features in the mean shift framework, and later combining it with color features in LBP_MS, LTP_MS, CLBP_MS1 and

CLBP_MS2, the estimation of the position, scale and orientation changes of the target will be notably enhanced.

4.1. LBP_MS and LTP_MS

Local Binary Pattern: The LBP operator is well known within the research area of texture analysis and pattern recognition. Proposed by Ojala et al. (1996), the texture of a region in the frame could be characterized by the distribution of the LBP. It is an efficient yet simple operator, which describes the spatial structure of the local texture of an image. This method thresholds P neighbors of each pixel with the center pixel, multiplies the thresholded values by the binomial weights and concatenates the results to get the LBP code. Finally, the resulting binary code is assigned to the center pixel (Fig. 1). Given a pixel in an image, LBP is generated by comparing it with its neighbors:

$$LBP_{N,R}(x_c,y_c)=\sum_{n=0}^{n-1}S_n(p_n-p_c)2^n,\quad S_n=\begin{cases}1,(p_n-p_c)\geqslant 0\\0,(p_n-p_c)<0\end{cases} \tag{20}$$

With p_c representing the grey value of the central pixel, p_n as the gray value of the neighbors; N and R are the number and the radius of the neighbors respectively. Assuming that the size of the image is $I\times J$, the coordinate of p_c are $(0,0)$ and those of p_n are $(Rcos(2\pi n/N),Rsin(2\pi n/N))$. The LBP texture is generated by building a histogram:

$$H(T)=\sum_{i=1}^{I}\sum_{j=1}^{J}f(LBP_{N,R}(i,j),T),T\in[0,T],\quad f(x,y)=\begin{cases}1,x=y\\0,\text{otherwise}\end{cases} \tag{21}$$

where T is the maximal LBP code value. Essentially, in areas with a frame of a nearly uniform appearance, the gray values of the central pixel and its neighbors are very close. Thus, the LBP operator leads to a paltry description of the target. By replacing the term in the LBP operator $s_n(p_n-p_c)$ in (19) with $s_n(p_n-p_c+a)$, Heikkilä and Pietikäinen (2006) succeed in modifying the thresholding strategy, which makes the LBP more discriminant. In order to hold the robustness of the LBP operator, the value of (a) should be relatively small.

Local Ternary Pattern LTP: The success of LBP operators in diverse computer vision applications has inspired researchers to develop different variants. Due to its flexibility, a generalization of the Local Ternary Patterns (LTP) descriptor has been invented. In order to build a new texture operator, Tan and Triggs (2010) were the first to propose this feature; which is a simple extension of the binary pattern to 3-valued pattern codes. As in LBP, a 3×3neighborhood around the center pixel in the frame is considered, the mathematical expression of the LTP is described as follows:

$$LTP_{N,R}=\sum_{n=0}^{N-1}s(p_n-p_c)2^n,\quad s(x)=\begin{cases}1,x\geqslant(p_n-p_c+t).\\0,(p_n-p_c-t)<x<(p_n-p_c+t).\\-1,x<(p_n-p_c-t).\end{cases} \tag{22}$$

With t representing a user-defined threshold. In LTP method, pattern strings are generated with three values $(1,0,-1)$ according to the threshold t set in advance, unlike the LBP which thresholds at the value of the central pixel. The variations in pixel value are more important as we increase the value of t, which enhance the thresholding results, and in our implementation we chose $t=5$. In order to remove the negative values, the ternary pattern is converted into two LBP units; the positive one is the upper LTP (LTPU) and the negative one is the lower LTP (LTPL), as shown in Fig. 2. The LTPU is generated by replacing the negative values in the original

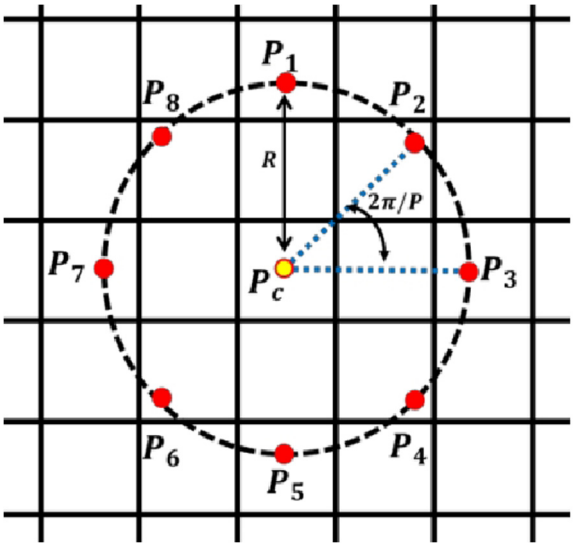

Fig. 1. Central pixel p and its circularly neighbors N = 8 with radius R = 1.

LTP by zeros, while the LTPL is generated in two steps: first, all values of 1 in the original LTP are replaced by zeros, afterward, the negative values are replaced by 1, Akhloufi et al. (2010). Two separate channels of LBP descriptors for both positive and negative components are calculated and finally the results are concatenated.

In general, color could provide many cues, and the wellknown color descriptor is the RGB color histogram which is often used for tracking in different occasions. In order to extract relevant information of the target, the last one was selected together with the LBP patterns extracted by (20) or LTP patterns extracted by (22). In the target model $\hat{q}_u$ of the Eq. (1), the color information of the object can be described adequately by fixing the three components of RGB color space to $16\times16\times16$ and by adding a new bin to the color histogram for the texture value, resulting thus $u=16\times16\times16\times LBP$ in the LBP_MS tracker and $u=16\times16\times16\times LTP$ in the LTP_MS tracker. Similarly, the candidate model $\hat{p}_u(y)$ is calculated with (2).

4.2. Complete local binary pattern texture

The CLBP is an extended version of LBP with some differences, Guo et al. (2010); Ahmed et al., 2011; Singh et al., 2012. In the conventional LBP feature only the signs of pixel differences are considered. In contrast, the CLBP feature further takes into account the magnitude (M) of local differences and the original center grey level (C). Given a central pixel p_c and its neighbors p_n, we suppose that d_n is the local difference vector, which characterizes the image local structure at p_c with $d_n=p_n-p_c$. We suggest embedding the LBP process proposed in Heikkilä and Pietikäinen (2006) into the new CLBP operator, thus, the local difference will be $d_n=p_n-p_c+a$. Formally, CLBP decomposes d_n into two components called the local difference sign-magnitude transform (LDSMT), Singh et al. (2012):

$$d_n=s_n\times m_n,\text{with}\begin{cases}s_n=sign(d_n)\\m_n=\|d_n\|\end{cases} \tag{23}$$

With m_n equaling the value of the difference between the central pixel and its neighbors. In our experiments, we fixed $R=1,N=8$ and $a=5$ to compute the CLBP texture. Consequently, we get three features which are very important for texture synthesizes;

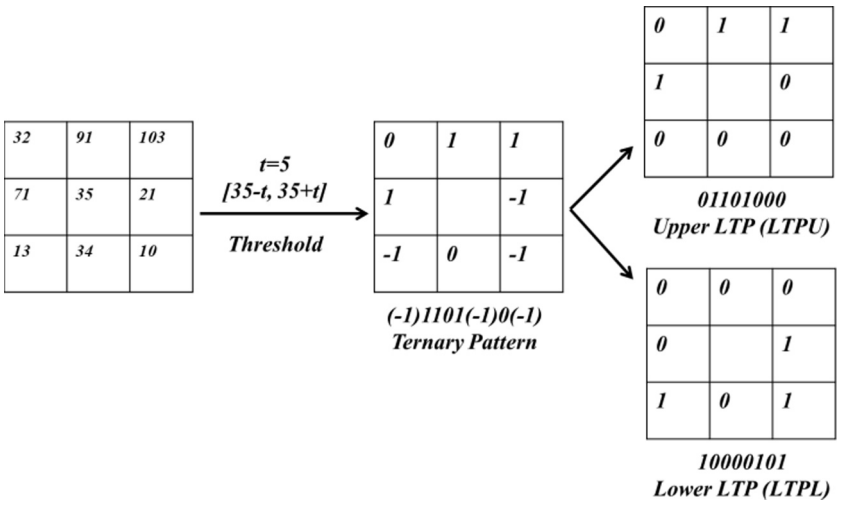

Fig. 2. Illustration of the basic LTP operator.

– CLBP_S, which is the conventional LBP,
– CLBP_M exploits the magnitude information, with c is a threshold.

$$CLBP\_M_{N,R} = \sum_{n=0}^{n-1} t(m_n, c)2^n, \quad t(x,c) = \begin{cases} 1, x \geqslant c \\ 0, x < c \end{cases} \tag{24}$$

– CLBP_C is used to code the grey level value of the original center:

$$CLBP\_C_{N,R} = t(g_c, c_I), \quad t(x,c) = \begin{cases} 1, x \geqslant c \\ 0, x < c \end{cases} \tag{25}$$

With c_I representing the threshold in the average grey level of the input frame. This subsection has two main parts; firstly, we summarized the feature extraction using only CLBP texture (CLBP_MS1 method), and then joined it with the color feature (CLBP_MS2 method), which collectively yield an extremely high performance. Secondly, we describe the methodology to embed it in the "scale and orientation adaptive mean shift tracking framework".

4.3. CLBP_MS1

The CLBP_MS1 is based only on texture, which is generated with the CLBP process. It is considered as an important feature, and the key items in this operator are CLBP(S/M/C). In CLBP_MS1 the target is presented by a distinctive histogram. The object to track is identified by an image region, and the information contained within is used to describe the moving object; but instead of focusing only on the value of an individual pixel (CLBP_C), the distribution of features defined at each pixel is used (CLBP_S and CLBP_M). The information may consist of uniformity, contrast, density and coarseness. Indeed, tracking is performed by sweeping the actual frame in order to find a similar region, with an histogram that best matches the target model histogram from the previous one. The process of feature extraction plays a major role in tracking an algorithm. CLBP texture histogram is commonly based on the properties of its three operators: CLBP_S, CLBP_M, and CLBP_C (Fig. 3). Thus, we propose to replace the R/G/B channels with CLBP(S/M/C). Consequently, the appearances of the target will be modeled with this texture distribution. The target model and the target candidate will be presented as follows and u will be the texture feature:

$$\begin{cases} \hat{q} = \{\hat{q}_u\}_{u=CLBP\_S,CLBP\_M,CLBP\_C}. \\ \hat{p} = \{\hat{p}_u(y)\}_{u=CLBP\_S,CLBP\_M,CLBP\_C}. \\ \omega_i = \sum_{u=1}^{m} \sqrt{\frac{\hat{q}_u}{\hat{p}_u(y_0)}} \delta[b(x_i^*) - u]. \end{cases} \tag{26}$$

4.4. CLBP_MS2

In this proposed method, the three features CLBP_S, CLBP_M, and CLBP_C will be fused to create a numeric matrix by using an alpha blending method. This technique is employed in computer graphics to blend each pixel from the first image with the corresponding pixel in the second one, Cao et al. (2010). The blending factor is called -alph-. We associate to the target model in the frame the -opacity value = alpha value-, which represents the probability of luminance energy. In this technique, the alpha value describes for each pixel the transparency of the target. It takes 1 if the surface is 100% opaque, and 0 if it represents a transparent object, Singh and Mishra (2011). Firstly, each pixel of the CLBP_S component would be combined with the values in the CLBP_M using the blending equation:

$$CLBP(S/M) = CLBP\_S \times (\partial) + CLBP\_M(1 - \partial) \tag{27}$$

We use CLBP_S and CLBP_M in first step then we add the CLBP_C. The result would be the new image, which overlays the original three CLBP component matrixes S/M/C.

$$CLBP(S/M/C) = CLBP(S/M) \times (\partial) + CLBP\_C(1 - \partial) \tag{28}$$

The shrinking or enlarging of the object in consecutive frames is usually a gradual process. In fact, the abrupt changes of the scale in adjacent frames make the tracking task very challenging. Consequently, we suppose that the size alteration of the target is sleek and minor, and this assumption is sustained correctly in most videos, including, those with high dynamic scenes. The valuation of the scale and orientation of the moving object are assured by computing the weight image derived from the target model $\hat{q}$ and the target candidate $\hat{p}$. In the conventional scale adaptive mean shift tracking algorithm the weight value of each pixel represents the probability that it belongs to the target candidate, and it is defined by (7) as the square root of the ratio of its color

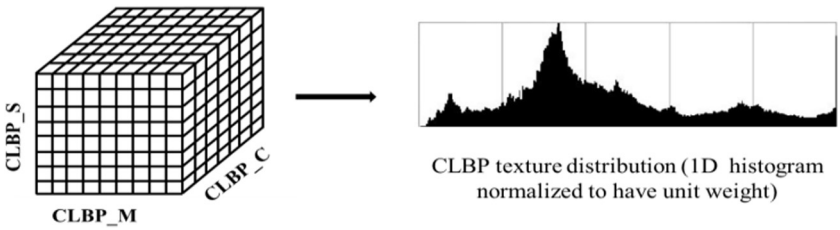

Fig. 3. Appearance via CLBP Histograms.

probability in $\hat{q}$ to its color probability in $\hat{p}$, Ju et al. (2010). In our method, such weight image value can be generated by the combination of the CLBP texture with the density distribution function of the target model in the target candidate region. Using (1), (2) and (28), CLBP texture and the RGB channels describe the target model $\hat{q}_u$ and the target candidate $\hat{p}_u(y)$. In order to acquire the texture and color distribution of the target region, we use (7) while adding to u another bin CLBP for the texture value; consequently, it becomes: $u = R \times G \times B \times CLBP$. The first three dimensions illustrate the quantized bins of the color channels, and the fourth dimention (i.e. CLBP) is the bin of the $CLBP(S \setminus M \setminus C)$ texture patterns (Fig. 4).

5. Experimental results and discussion

In this section, we perform experiments on a range of real-world sequences that represent a variety of challenges, such as illumination changes, quick moving objects, multiple object intersections, and moving cameras. The sequences contain various types of targets (rigid, articulated) and different scenarios of occlusions. We applied three state-of-the-art tracking algorithm to a series of experiments aiming to prove the reliability of our proposed tracking methods using variants of LBP texture and color quantification. Our proposed algorithms LBP_MS, LTP_MS, CLBP_MS1 and CLBP_MS2 are compared with the corrected background-weighted histogram method (CBWH), Ning et al. (2012), the EM_shift tracking algorithm (EM_shift), Zivkovic et al. (2004), and the scale and orientation adaptive mean shift tracking (SOAMST), Ning et al. (2012). In all tracking methods, we choose RGB color as the feature space and we quantize it to $16 \times 16 \times 16$ bins. The first video is an auto racing sequence of 100 frames; some samples of the CBWH tracker, the EM_shift tracker, the SOAMST tracker and our proposed tracker's results are demonstrated in Fig. 5, with 4 random frames of the video sequences shown. The frame indexes are respectively 51, 71, 79 and 100. This sequence is registered with an atypical digital camera; thus, the contrast between the target and its background is very poor due to the poor illumination and the blurred quality of the sequence. Moreover, the noise level is relatively high, as it is caused by the continuous speedy movement of the auto racing and crashing. The tracking effectiveness of the seven algorithms for the first part of this sequence is plotted in. Since the device is equipped with a powerful zoom, we see an obvious scale change; consequently the fixed-scale mean shift algorithm CBWH is unable to achieve good tracking results.

Starting form frame 71, the EM_shift, SOAMST and CLBP_MS1 can successfully localize the auto racing, but occasionally lose the orientation and the scale. Despite the uses of color and texture information, the LBP_MS and LTP_MS tracking results are unsatisfying, while the CLBP_MS2 method succeeds in localizing the target, adjusting the ellipse to its exact size and dealing with the orientation of the auto racing while spinning out of the pathway during the accident. A single hypothesis is carried; it is when the target representation is based on a single characteristic such as color in CBWH, EM_shift and SOAMST, or texture in CLBP_MS1, the tracker cannot deal with the challenging video and does not extract the best characteristics of the target. Actually, in practice, the reliability of LBP decreases notably under large illumination variations. And in this experiment, the LTP operator proves its sensitivity to such noise. The combinations of the color with CLBP texture succeed in describing the object and present its accurate feature, and by embedding this feature extraction in the scale and orientation adaptive mean shift tracking algorithm, our CLBP_MS2 method makes the convergence to the target candidate easier.

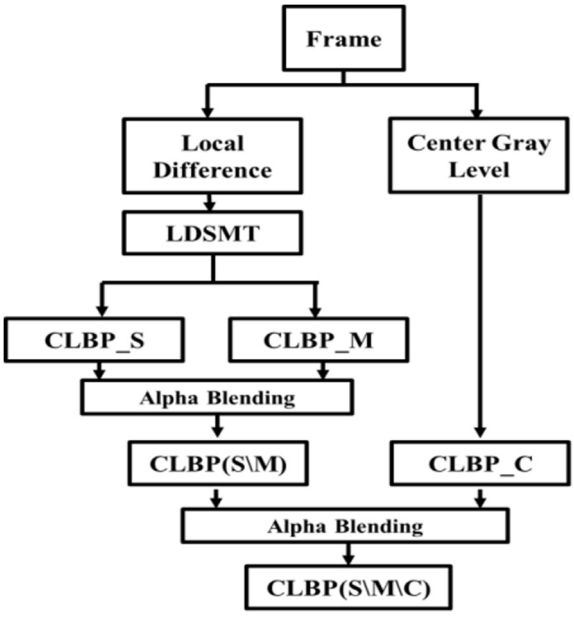

Fig. 4. Framework of CLBP_MS2 texture feature.

The second sequence is filmed with a small camera taped to the bottom of a skateboard. It shows a skateboarding performance. This video is considered challenging because of the speed of the target, as well as its turns and twists. Fig. 6 presents the tracking results of the seven methods in frame 25, 46, and 56. CBWH and CLBP_MS1 could not track the skateboarder and eventually drift away from the skateboarder, while all other trackers successfully localize the target. The skateboarders legs have the same color as the road which leads to an overlap in target boundaries. This disables the ellipse of EM_shift, SOAMST, LBP_MS and LTP_MS to fit the size and the orientation of the target. Thanks to the CLPB texture, spurious edges and boundary gaps could not prevent our proposed method, CLBP_MS2, to localize and adapt the scale and orientation of the skateboarder. The combination of the color features and CLBP texture are sufficient to overcome such difficulties, which remarkably enhance the feature extraction process. Consequently, CLBP_MS2 robustly tracks the target through clutter.

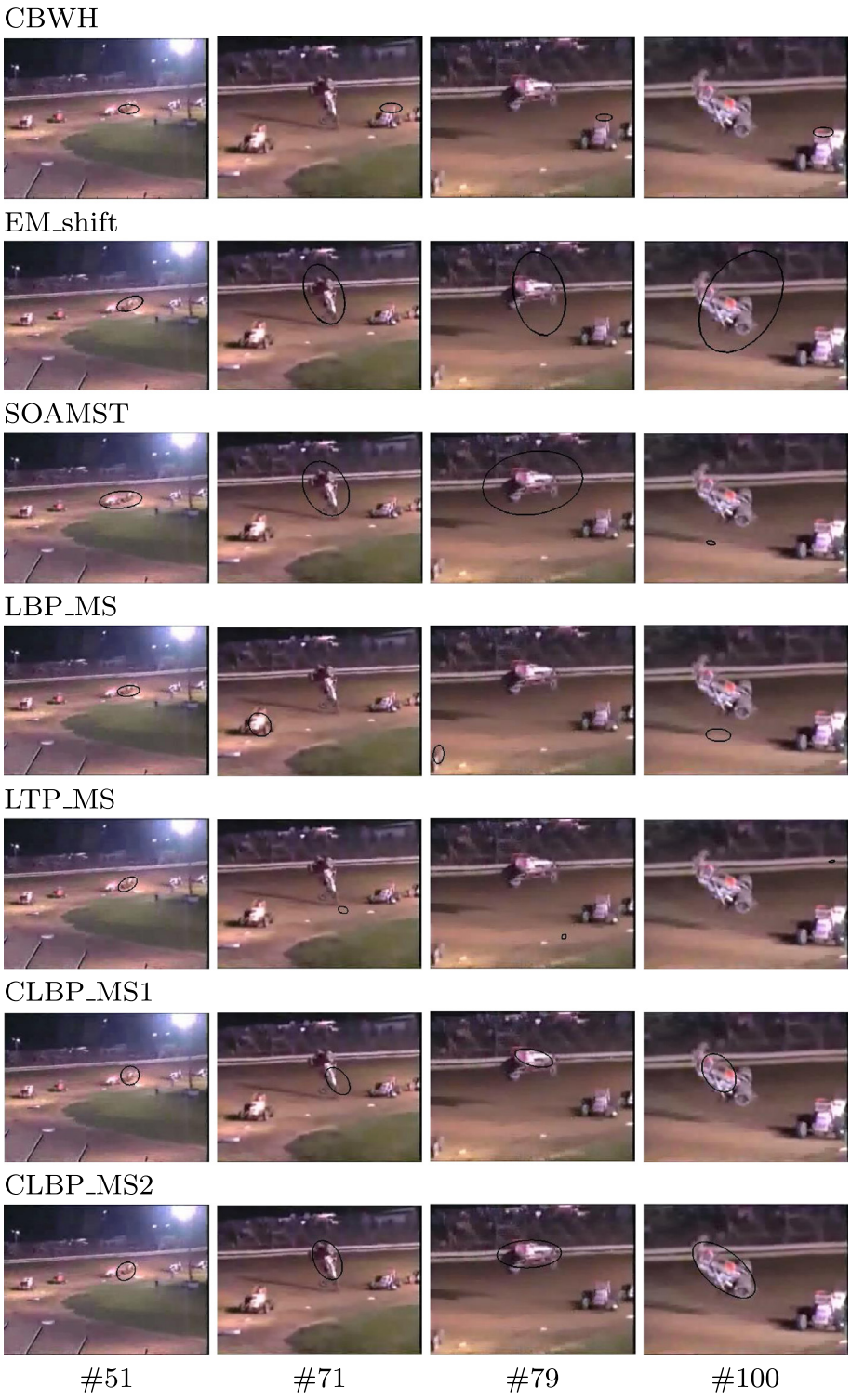

Fig. 5. Tracking results of an auto racing sequence by the target representation models CBWH, EM_shift, SOAMST, LBP_MS,LTP_MS, CLBP_MS1, and CLBP_MS2. Frames 51, 71, 79 and 100 are displayed.

Fig. 7 shows tracking results of frames 9, 47 and 60 of a sequence with 79 frames where the target is a white truck. CBWH, EM_shift and CLBP_MS1 perform well to track the target, but could not settle its right size and orientation, as those tracking methods cannot deal with pose changes. In contrast, SOAMST, LBP_MS, LTP_MS and CLBP_MS2 could achieve the hole tracking process successfully. In fact, the use of the Bhattacharyya coeffcient, the zeroth-order moment and the corrected second-order center moments between the target model and the candidate model make the estimation of the height, width and orientation changes very accurate. Since SOAMST and CLBP_MS2 achieve the best results in this video, we propose to compare the number of iterations of each method while converging to the right position and orientation.

Fig. 8, presents the number of iterations performed by the new method CLBP_MS2 and the SOAMST on the third experimental

Fig. 6. Tracking results of a skateboard sequence by the target representation models CBWH, EM_shift, SOAMST, LBP_MS,LTP_MS, CLBP_MS1, and CLBP_MS2. Frames 25, 46 and 56 are displayed.

sequence. In the following figure, the total number of iterations is depicted for computing the whole process to track the truck based on the two tracking methods. Based on the above observation, we can see that the iteration number of our method is less than that of SOAMST (presented in red). The red curve is often above the blue one which indicates that this method requires more iterations than our method (frame 11, 17, 33, 43). The inferior iteration number needed for CLBP_MS2 to find the best scale position and orientation under the clutter and the poor condition of this sequence proves that the combination of color and CLBP texture generates an effective feature that strongly matches the model and the candidate target.

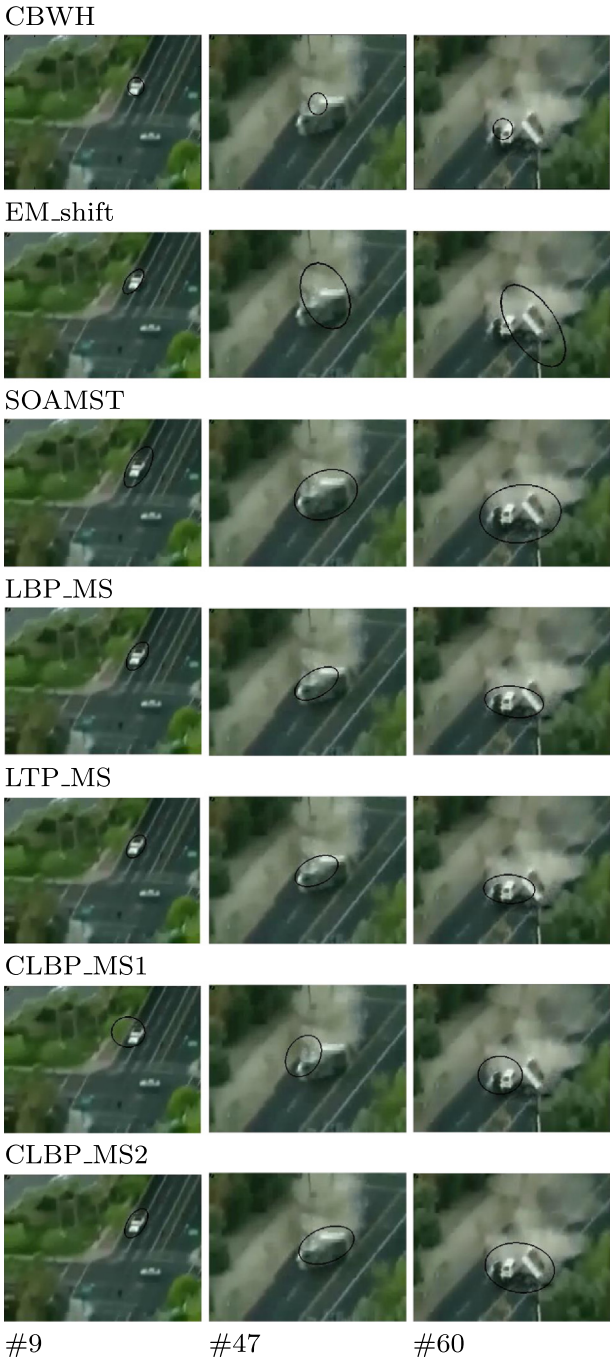

Fig. 7. Tracking results of a white truck sequence by the target representation models CBWH, EM_shift, SOAMST, LBP_MS, LTP_MS, CLBP_MS1, and CLBP_MS2. Frames 9, 47 and 60 are displayed.

This experiment (Fig. 9) is a sequence of 235 frames of parachuting sports. Our target is the skydiver spinning out of control. The CBWH method is designed to reduce the background's intrusion in target localization, so this characteristic enables the method to give good tracking results. But since it is a fixed scale mean shift tracking method it makes the settlement of the scale and orientation impossible. The EM_shift robustly tracks the skydiver through clutter in frame 165, 186 and 294, but it drifts away from the target and cannot recover from the lost track. Thereby, this method does not handle inconsistent motion and pose change. The SOAMST tracking results are the worst in this experiment and

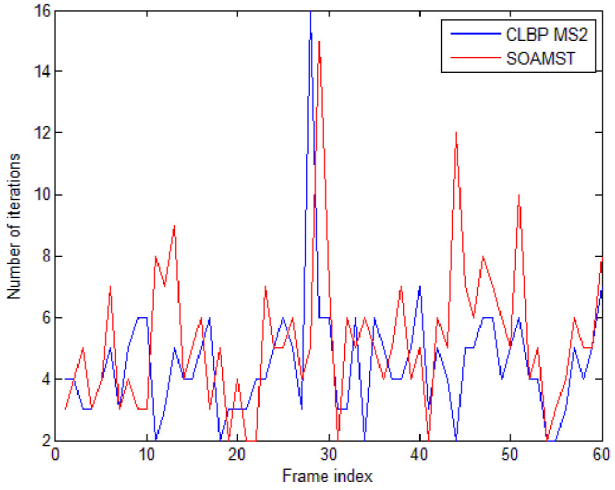

Fig. 8. Iteration (SOAMST, CLBP_MS2).

the method is unable to even localize the target. It does not work in cluttered backgrounds, as it picks discriminative color features in each frame which make it less effective in dealing with low-contrast images. Consequently, it fails to accomplish the tracking process when there is a similarity between the target and its background.

Observing the target's location, CLBP_MS1 method tracks the skydiver correctly but fails when it comes to identifying the size and orientation of the target. This is due to the fact that this tracker is mainly based on the three features of the CLBP (M/S/C) (the description of the target requires only the texture feature). The LBP_MS, LTP_MS and the CLBP_MS2 methods use the color and variants of LBP texture jointly to describe the skydiver, which turns these features less sensitive to image blur. These trackers work very well in this video when the target undergoes large pose and appearance change in a cluttered background, explaining the lesser number of drifting errors when compared to the other methods.

Figs. 10 and 12 present the tracking results on another two different sequences. The experiment in Fig. 10 is on a vehicle in a rugby game, where the vehicle moves quickly. The results attained through SOAMST and CLBP_MS2 are identical, achieving very good results not only in localizing the object, but also in effectively defining its scale and orientation. The similarity in the results given by LTP_MS and LBP_MS is explained by the fact that the LTP operator inherits most of the other key advantages of LBP. The value of the Bhattacharyya distance calculated by (5) for each frame is shown in Fig. 11. This similarity coefficient indicates a perfect match between the target model and the chosen candidate when it takes the zero value. In the first frames of this sequence, the car is well detected, thus the value of the Bhattacharyya distance is between 0.1 and 0.2. Afterward, signifcant deviations from this value, as shown in frame 36 by peaks that reach $0.61, 0.65, 0.68$ and 0.69 for CLBP_MS2, SOAMST, LTP_MS and LBP_MS respectively. The complete occlusion caused by the crowd, elevates the residual distance value. The blue curve of the CLBP_MS2 tracker is inferior to the other curves in this sequence. It can be understood that under the difficult experimental setup and among the four trackers, the Bhattacharyya distance of CLBP_MS2 is the lowest. Consequently, the combined color and the CLBP texture are an adequate representation of the target. These results demonstrate the superiority of the proposed CLBP against the newly developed texture operators, LBP and LTP. This is because the CLBP texture is insensitive to noise and possesses a high discriminating property that achieves impressive tracking results.

Fig. 9. Tracking results of a parachuting sequence by the target representation models CBWH, EM_shift, SOAMST, LBP_MS, LTP_MS, CLBP_MS1, and CLBP_MS2. Frame 165, 186, 195, and 294 are displayed.

Fig. 12 is on a more complex Gym_ball sequence filmed with a fixed camera. The object tracked here is the bleu ball, and it exhibits obvious shape changes and spurious edges of the ball due to its rapidity (frames 41 and 75). The experimental results show that the LBP_MS, LTP_MS and CLBP_MS2 achieve much better performances in localizing the ball than the rest of trackers, which fail to track it quickly. SOAMST uses the RGB color histogram to represent the target, which is adequate in this experiment. Table 1 states the average numbers of iterations by the seven trackers on the experimental sequences. The iteration number of CLBP_MS2 is the lowest because it perfectly models the target so it does not need significant iteration to converge. The average number of iterations of the proposed CLBP_MS2 and the SOAMST are approximately equal. The main factor that affects the convergence speed of the EM-shift and those two trackers is the computation of the covariance matrix. For each iteration, EM_shift estimates it and runs the mean shift algorithm three times, whereas SOAMST, LBP_MS, LTP_MS, CLBP_MS1 and CLBP_MS2 only estimate iteration it once for each frame.

All algorithms are implemented in MATLAB R2013a interface and ran on a PC with Intel® *Core*™ 2 Duo 2.1 GHz CPU and 2 GB RAM. We compute the time requirement analysis for each method. Table 2 shows that the CBWH, EM_shift, SOAMST and CLBP_MS1 have the lowest computational time because it uses only color or texture to model the target and the candidate. For the LBP_MS, LTP_MS and CLBP_MS2, the trackers employ jointly the color and texture feature in the tracking algorithm, thus the computational time is important comparing to other methods. In the future, we will work on finding a balance between reducing computational time and increasing the tracking accuracy.

Finally, the CLBP_MS2 is evaluated in a noisy sequence of a racing car displayed on Fig. 13. The video suffers from various types of artifacts. Therefore, only the second method succeeds in accurately tracking the racing car under these inconvenient clutters and occlusions. In fact, to achieve robust tracking results, the appear-

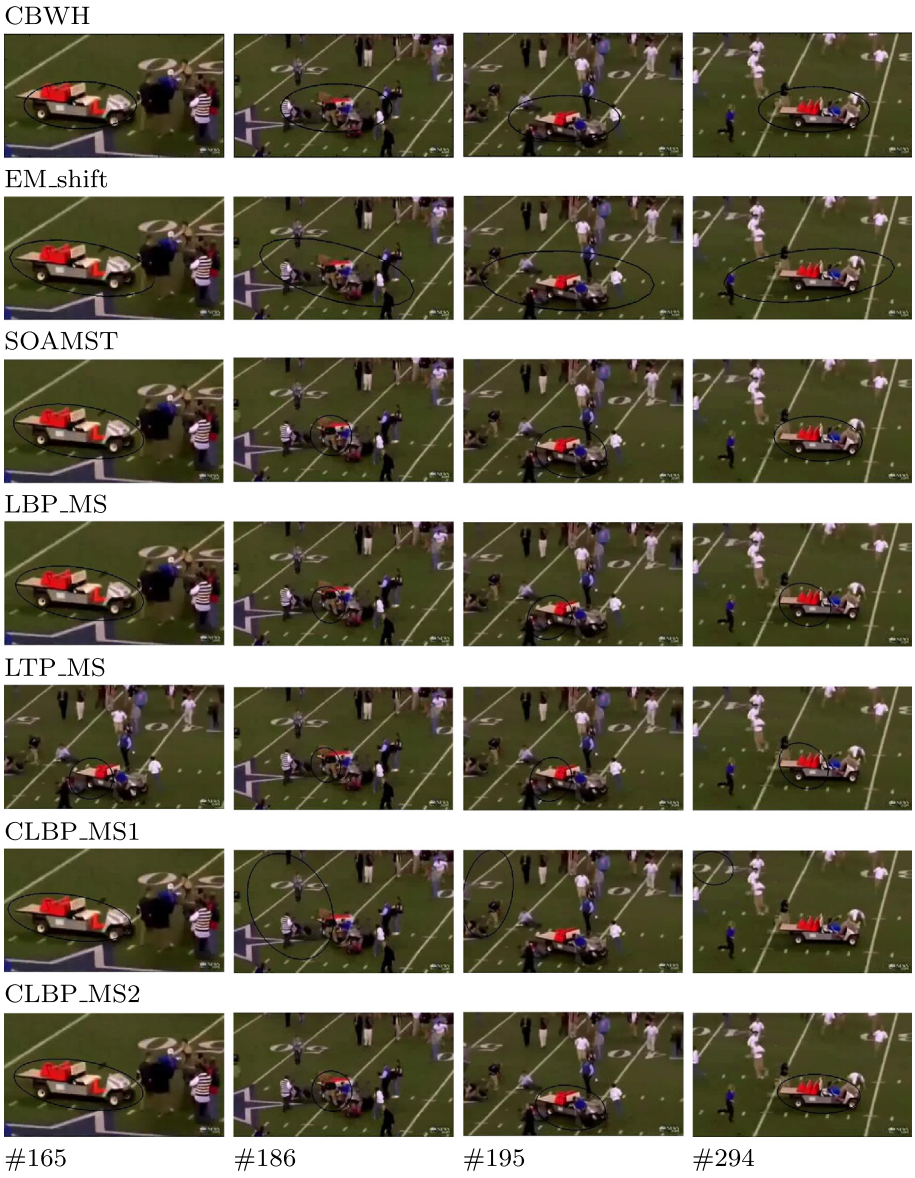

Fig. 10. Tracking results of a vehicle sequence by the target representation models CBWH, EM_shift, SOAMST, LBP_MS, LTP_MS, CLBP_MS1 and CLBP_MS2. Frame 165, 186, 195, and 294 are displayed.

ance and pose updates must be taken into careful consideration. Those transformations of the targets form during the entire tracking period, lead to poor localization and make the tracker super-sensitive to noise and occlusion. Only five frames are displayed for this experiment where the quality of the image is blurry and fuzzy, especially in frame 141.

Despite the partial occlusion caused by the post in frame 99, our method proves without any increase in complexity to be a robust tracker. The accurate presentation and description of the target model and candidate managed to overcome those limitations, which make the CLBP_MS2 method effectively handle different conditions. Table 3 lists the estimated semi-major length (width), semi-minor (length) height and orientation of the ellipse in the racing car sequence by using the CLBP_MS2 scheme. In each frame (1, 19, 85, 99, 107 and 141) we calculate the real dimension of the red car and the estimated one for the ellipse. The orientation is computed as the angle between the major axis and the horizontal axis. The sequences (auto racing, skateboard, white truck,

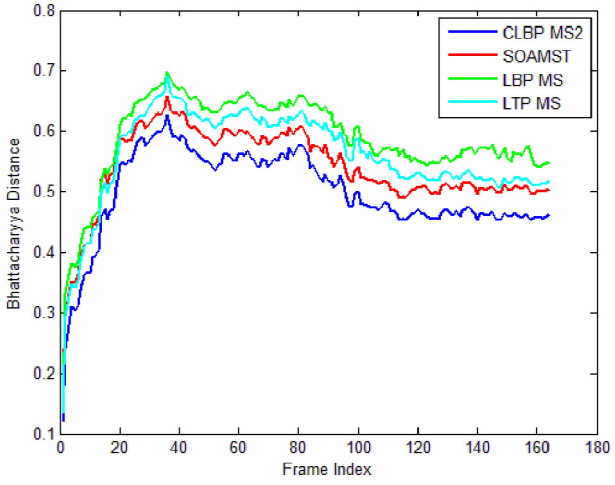

Fig. 11. The minimum value of the Bhattachayya distance function of the frame index for the vehicle sequence. The mean distance of Bhattacharyya is 0.2894 per frame.

Fig. 12. Tracking results of a Gym_ball sequence by the target representation models CBWH, EM_shift, SOAMST, LBP_MS, LTP_MS, CLBP_MS1, and CLBP_MS2. Frames 1, 41, 52 and 75 are displayed.

Table 1
Mean shift iterations of the experiments.

Video sequence	Frames	Target Representations	Mean shift iteration Average number
Auto-racing	100	CBWH	5,85
		EM_shift	6
		SOAMST	5.15
		LBP_MS	5.09
		LTP_MS	6.21
		CLBP_MS1	4.52
		CLBP_MS2	4.46
Skateboard	35	CBWH	6.5714
		EM_shift	6.02
		SOAMST	5.2286
		LBP_MS	5.1142
		LTP_MS	7
		CLBP_MS1	5.4857
		CLBP_MS2	5.3429
Truck	79	CBWH	3.9747
		EM_shift	5.9873
		SOAMST	4.9114
		LBP_MS	4.7468
		LTP_MS	5.9113
		CLBP_MS1	4.3165
		CLBP_MS2	4.5696
		CBWH	3.9830
		EM_shift	6.0128
		SOAMST	5.0511

(*continued on next page*)

Table 1 (*continued*)

Video sequence	Frames	Target Representations	Mean shift iteration Average number
Parachuting	235	LBP_MS	4.2893
		LTP_MS	6.0978
		CLBP_MS1	5.8085
		CLBP_MS2	4.7447
		CBWH	6.7622
		EM_shift	7.5793
		SOAMST	7.4878
Vehicle	164	LBP_MS	6.5610
		LTP_MS	7.3049
		CLBP_MS1	7.6585
		CLBP_MS2	7.3293
		CBWH	7.0533
		EM_shift	6.0133
		SOAMST	9.2933
Gym-ball	75	LBP_MS	9.1067
		LTP_MS	9.4800
		CLBP_MS1	8.4267
		CLBP_MS2	8.3600

Table 2
Time requirement analysis of each of the methods (second).

Trackers	Auto-racing	Skateboard	Truck	Parachuting	Vehicle	Gym-ball
CBWH	20.2845	9.1553	12.9281	168.3900	245.1453	147.0714
EM_shift	25.6560	14.4544	23.8235	188.7431	260.1622	153.8044
SOAMST	21.7122	10.0975	14.7655	175.6555	253.4505	150.7876
LBP_MS	27.0098	17.1576	30.9597	190.7060	276.0838	160.0211
LTP_MS	27.2322	17.1419	31.6991	192.0318	277.9619	161.2321
CLBP_MS1	20.0443	8.1626	11.2543	170.9502	241.0046	146.5545
CLBP_MS2	25.5456	13.1386	17.5472	174.4898	251.2400	161.7803

Fig. 13. Tracking results of a racing car sequence by CLBP_MS2. Frames 19, 85, 99, 107 and 141 are displayed.

Table 3
Dimension of the ellipse tracking the target object.

Frame	Semi-major length a			Semi-minor length b			Orientation		
No.	Real	Estimated	Error (%)	Real	Estimated	Error (%)	Real	Estimated	Error (%)
1	80.59	84.90	5.08	40.1028	53.6062	2.51	7.4463	4.7255	5.75
19	73.38	81.27	9.71	36.2651	49.2403	2.63	7.6319	10.8085	2.93
85	49.33	69.07	2.85	27.4545	44.8459	3.87	18.4129	10.4273	7.65
99	69.12	81.36	1.5	21.2542	38.1142	4.42	11.1217	7.4258	4.97
107	59.37	73.65	1.93	32.6693	48.5529	3.27	3.2127	4.1812	2.31
141	52.72	52.62	0.19	31.3939	27.6344	1.36	1.7497	2.6551	3.41

parachuting, vehicle, Gym_ball, car) can be downloaded from the URL.[1] Further results of the CLBP_MS2 are displayed in Fig. 14. The four sequences (Skiing, Deer, David3, Coke), can be downloaded from the URL.[2] This novel approach, which jointly uses CLBP feature generation method and color feature, for describing the target in the scale-orientation adaptive mean shift tracking algorithm proved to

[1] https://www.youtube.com/channel/UCPFX0LadjF_R1cro2vxGjcA.

[2] https://sites.google.com/site/trackerbenchmark/benchmarks/v10.

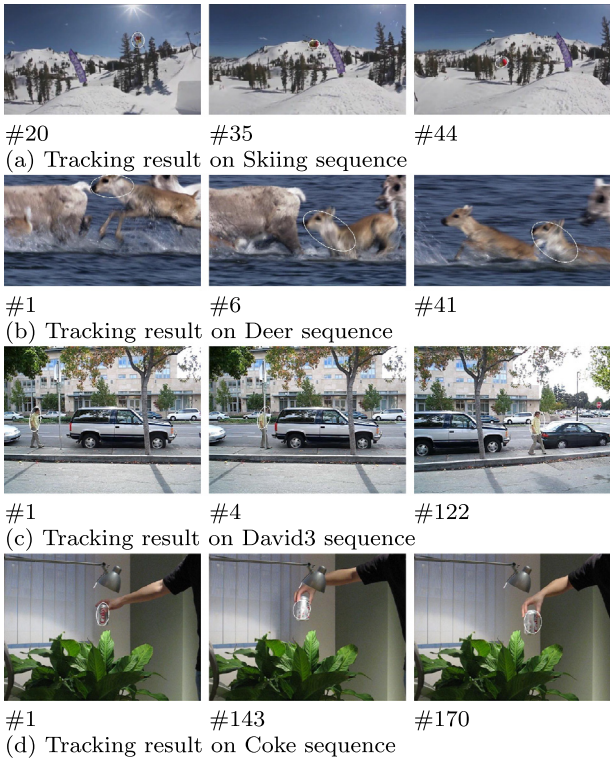

Fig. 14. The tracking results of selected frames using the proposed method CLBP_MS2.

be efficient in localizing the racing car. The estimation accuracy of the scale and orientation of the target provides acceptable results which could be improved in future works to get the exact size and orientation.

6. Conclusion

In this paper, we present an efficient and robust tracker by embedding a variety of LBP texture feature in the mean shift framework. Image fusion is carried out using the three features CLBP(S/M/C) based on an alpha blending technique to obtain the CLBP texture. The expected benefit is an improvement in the accuracy of the tracking; thus, we test it in terms of handling the challenging factors in the tracking of objects. We evaluate and compare our methods with state-of-the-art tracking algorithms. The experimental results demonstrate the strength of the CLBP MS2, as well as the weaknesses of other tracking algorithms. Our proposed feature extraction is designed to account not only for dynamic motion, but also for appearance, pose, and orientation changes. However, the methods presented in this article require manual initialization of the object to track. Consequently, an automatic initialization scheme should be considered in future works according to a specific application. Finally, since our proposed CLBP_MS2 method demonstrates acceptable accuracy against blurry public sequences and defocused sequence, we are now considering the possibility of tracking multiple items through occlusion. Moreover, future research work on tracking random object will include incorporation of the dictionary learning features to improve the efficiency of our proposed trackers.

References

Ahmed, Faisal, Hossain, Emam, Shihavuddin, A.S.M., et al., 2011. Compound local binary pattern (clbp) for robust facial expression recognition. In: IEEE 12th International Symposium on Computational Intelligence and Informatics (CINTI), 2011. IEEE, pp. 391–395.

Akhloufi, Moulay, Bendada, Abdelhakim, et al., 2010. Locally adaptive texture features for multispectral face recognition. In: IEEE International Conference on Systems Man and Cybernetics (SMC), 2010. IEEE, pp. 3308–3314.

Aliabadian, Amir, Akbarpour, Esmaeil, Yosefi, Mohammad, 2012. Kernel based approach toward automatic object detection and tracking in surveillance systems. Int. J. Soft Comput. Eng. 2, 82–87.

Cao, Wei, Duan, Fuzhou, Gong, Huili, Zhao, Wenji, 2010. A fast and automatic method to update the scene of disaster area by normal photos in three-dimensional environment. In: 18th International Conference on Geoinformatics, 2010. IEEE, pp. 1–4.

Comaniciu, Dorin, Ramesh, Visvanathan, Meer, Peter, 2000. Real-time tracking of non-rigid objects using mean shift. In: IEEE Conference on Computer Vision and Pattern Recognition, 2000. Proceedings, vol. 2. IEEE, pp. 142–149.

Comaniciu, Dorin, Ramesh, Visvanathan, Meer, Peter, 2003. Kernel-based object tracking. IEEE Trans. Pattern Anal. Mach. Intell. 25 (5), 564–577.

Du, Dawei, Qi, Honggang, Huang, Qingming, Zeng, Wei, Zhang, Changhua, 2013. Abnormal event detection in crowded scenes based on structural multi-scale motion interrelated patterns. In: Multimedia and Expo (ICME), 2013 IEEE International Conference on. IEEE, pp. 1–6.

Guo, Zhenhua, Zhang, Lei, Zhang, David, 2010. A completed modeling of local binary pattern operator for texture classification. IEEE Trans. Image Process. 19 (6), 1657–1663.

Heikkilä, Marko, Pietikäinen, Matti, 2006. A texture-based method for modeling the background and detecting moving objects. IEEE Trans. Pattern Anal. Mach. Intell. 28 (4), 657–662.

Henriques, JoãoF, Caseiro, Rui, Martins, Pedro, Batista, Jorge, 2015. High-speed tracking with kernelized correlation filters. IEEE Trans. Pattern Anal. Mach. Intell. 37 (3), 583–596.

Ju, Ming-Yi, Ouyang, Chen-Sen, Chang, Hao-Shiu, 2010. Mean shift tracking using fuzzy color histogram. In: Machine Learning and Cybernetics (ICMLC), 2010 International Conference on, vol. 6. IEEE, pp. 2904–2908.

Martin, Rhys, Arandjelović, Ognjen, 2010. Multiple-object tracking in cluttered and crowded public spaces. In: Advances in Visual Computing. Springer, pp. 89–98.

Ning, Jicai, Zhang, Leiqi, Zhang, Dejing, Chunlin, Wu., 2012. Scale and orientation adaptive mean shift tracking. Comput. Vision, IET 6 (1), 52–61.

Ning, Jifeng, Zhang, Lei, Zhang, David, Chengke, Wu., 2012. Robust mean-shift tracking with corrected background-weighted histogram. IET Comput. Vision 6 (1), 62–69.

Ojala, Timo, Pietikäinen, Matti, Harwood, David, 1996. A comparative study of texture measures with classification based on featured distributions. Pattern Recogn. 29 (1), 51–59.

Singh, AkhilPratap, Mishra, Agya, 2011. Wavelet based watermarking on digital image. Indian J. Comput. Sci. Eng. 1 (2), 86–91.

Singh, Sushil, Maurya, Ritesh, Mittal, Anish, 2012. Application of complete local binary pattern method for facial expression recognition. In: 4th International Conference on Intelligent Human Computer Interaction (IHCI), 2012. IEEE, pp. 1–4.

Sliti, Oumaima, Gmati, Chekib, Benzarti, Fouzi, Amiri, Hamid, 2014a. Robust object tracking using log-gabor filters and color histogram. In: Proceedings of the 3rd International Conference on Pattern Recognition Applications and Methods. SCITEPRESS-Science and Technology Publications, Lda.

Sliti, Oumaima, Hamam, Habib, Benzarti, Faouzi, Amiri, Hamid, 2014b. A more robust mean shift tracker using joint monogenic signal analysis and color histogram. In: 22nd International Conference on Pattern Recognition (ICPR), 2014. IEEE, pp. 2453–2458.

Sridhar, Srinath, Mueller, Franziska, Oulasvirta, Antti, Theobalt, Christian, 2015. Fast and robust hand tracking using detection-guided optimization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3213–3221.

Stolkin, Rustam, Florescu, Ionut, Baron, Morgan, Harrier, Colin, Kocherov, Boris, 2008. Efficient visual servoing with the abcshift tracking algorithm. In: IEEE International Conference on Robotics and Automation, 2008. ICRA 2008. IEEE, pp. 3219–3224.

Tan, Xiaoyang, Triggs, Bill, 2010. Enhanced local texture feature sets for face recognition under difficult lighting conditions. IEEE Transac. Image Process. 19 (6), 1635–1650.

Wu, Yi, Lim, Jongwoo, Yang, Ming-Hsuan, 2015. Object tracking benchmark. In: 1848. IEEE Transactions on Pattern Analysis and Machine Intelligence 37 (9), 1834.

Zhang, Kaihua, Zhang, Lei, Liu, Qingshan, Zhang, David, Yang, Ming-Hsuan, 2014. Fast visual tracking via dense spatio-temporal context learning. In: European Conference on Computer Vision. Springer, pp. 127–141.

Zivkovic, Zoran, Krose, Ben, 2004. An em-like algorithm for color-histogram-based object tracking. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004, vol. 1. IEEE, pp. I–798.