Pattern Recognition Letters 90 (2017) 50–57

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Real time visual tracking using a spatially weighted von Mises mixture model

Vasileios Karavasilis*, Christophoros Nikou, Aristidis Likas

Department of Computer Science and Engineering, University of Ioannina, Ioannina 45110, Greece

ARTICLE INFO

Article history:
Received 19 August 2016
Available online 18 March 2017

Keywords:
Visual tracking
Von Mises mixture model (VMMM)
maximum likelihood
Expectation-Maximization (EM)
Model-free tracking
Kernel-based tracking

ABSTRACT

A real time, kernel based, model-free tracking algorithm is proposed, which employs a weighted von Mises mixture as the target's appearance model. The mixture weights, which are provided by a spatial kernel, along with the hue values are used in order to estimate the parameters of the weighted von Mises mixture model. The von Mises distribution is suitable for circular data and it is employed in order to eliminate drawbacks in kernel-based tracking caused by eventual shifts of the target's histogram bins. The weights allow a mean shift-like gradient based optimization by maximizing the weighted likelihood, which would not be feasible in the context of a standard von Mises mixture. Moreover, as only the hue component of the target is involved, many quantities of the algorithm may be pre-calculated for given parameters and therefore the algorithm can perform in real time, which is experimentally confirmed. Finally, it is shown that the proposed method has comparative performance in terms of accuracy and robustness with other state-of-the-art tracking algorithms.

1. Introduction

Visual object tracking is among the challenging problems in computer vision, with many applications, such as human computer interaction, robotics and surveillance. The challenges in visual object tracking include the presence of occlusion, change of scene illumination, variations in object's scale and shape, camera movement and varying viewpoints. In order to tackle these challenges, many trackers have been proposed with different approaches to the problem [34]. Some of these techniques include particle filters [18,29,30], template matching [31,39,42], target's subregion tracking [10,43,45] and object - background distinction [2,9,11,37,40,41]. Usually, the representation of the target and the function that matches the target between the frames is part of the proposed methodology. However, in a different approach the matching function is learned by the proposed Siamese deep neural network [36].

A category of trackers addresses the problem of model-free shape by employing spatial kernels and modeling the color distribution of the target [6,23–25,28,38]. Although the algorithms in this category have some differences, they share some common properties. In general, the region of the object is approximated by an ellipse and it is spatially masked by a kernel which assigns greater weights to pixels near the center of the ellipse. The kernel makes feasible a gradient-based optimization instead of a brute force search for target localization and real-time performance. Moreover, its color distribution is usually modeled by a histogram. Here, we must note that the term kernel which is used in the current work should not be confused with the kernel trick that is used in some approaches and can efficiently perform a non-linear classification. These approaches include a kernelized SVM classifier which learns a prediction function that directly estimates the object transformation between frames [12] and a kernelized correlation filter which has the exact same complexity as its linear counterpart and can process hundreds of frames per second [14].

A representative algorithm of this category is the mean shift algorithm [7]. In the first frame, it estimates a target model which is represented by a histogram. In consecutive frames, the location in which the corresponding histogram is similar to the target model is estimated. This approach compares only the corresponding bins between histograms, so if the bins values are shifted due to illumination change, then the object may be lost. Some other approaches have tried to solve this issue. The earth mover's distance (EMD) was used in order to estimate the distance between the target model and target candidate histogram signatures [44]. However, the solution is not in closed form and each iteration is limited to one pixel movement. Similar in spirit, an algorithm was proposed which minimizes the EMD between Gaussian mixture models [16]. In an other method, the EMD is minimized for 1D histograms but its computation is avoided for multidimensional histograms using a cross-bin metric [22]. Finally, weighted Gaussian mixture models

* Corresponding author.
E-mail addresses: vkaravas@cs.uoi.gr (V. Karavasilis), cnikou@cs.uoi.gr (C. Nikou), arly@cs.uoi.gr (A. Likas).

http://dx.doi.org/10.1016/j.patrec.2017.03.013
0167-8655/

have been employed in order to model the target in the first frame of the image sequence and to estimate the spatial location of the target in subsequent frames using a mean shift update [17].

Although the features that represent the object are usually the luminance or the RGB color value of each pixel, the HSV provides an attractive representation that has been used for visual tracking [1]. Moreover, the HSV color space was employed in the context of mean shift [4]. The hue component is a flexible representation, due to the fact that is closely related to what humans perceive as color. Moreover, hue is unrelated to illumination changes, as these changes are encoded in the saturation and value components. These properties are highlighted by the author in order to support his decision to use only the hue component. Another advantage of using the hue component of the HSV color space instead of the full RGB color space is that the dimensions of the problem are reduced to one instead of three.

The hue component does not depend on illumination changes, but this does not prevent its histogram bins to be shifted (Fig. 3). In these cases, we can not directly apply the approaches that were proposed for the mean shift algorithm, due to the fact that the hue is periodic with period 2π and these methodologies have been proposed for linear color spaces.

In this work, we use the hue component of the HSV color space for visual object tracking and we employ a weighted von Mises mixture model in order to overcome drawbacks caused by shifting histogram values. The von Mises distribution is the circular analog of the normal distribution on a line, and it can be used in order to model circular data. The values of the hue component that is periodic with period 2π, can be represented as points in the two dimensional unit circle. Thus, the terms periodic random variable and circular data will be considered interchanged in this paper. Moreover, we propose the weighted von Mises mixture to model the distribution of the hue value when a single von Mises distribution is not flexible enough to describe the target. To the best of our knowledge, this is the first time a von Mises mixture is used in visual object tracking in order to model the object appearance. Moreover, the proposed weighted von Mises mixture employs the spatial weights that are provided by the kernel. The von Mises distribution has been employed in order to model sensor noise [32], the direction of the movement [5,19,35], and the pose of an object [15]. Moreover, a single wrapped Gaussian distribution, which was also designed for circular data, was used in order to model the background hue component [33].

In the remaining of the paper, Section 2 reviews the von Mises distribution and presents the proposed weighted von Mises mixture model, Section 3 integrates the proposed weighted von Mises mixture model in the visual tracking framework. Experimental results are presented in Section 4 and conclusions are drawn is Section 5.

2. Weighted von Mises mixture model

2.1. Introduction to the von Mises distribution

There are cases in image processing and analysis where the measured quantity is periodic and modeling it by a periodic variable may be an advantage (e.g. the hue component of an image in the HSV color space). In what follows, we assume that the period is 2π and the periodic variable is defined in the interval $[0, 2\pi)$. If the variable is defined in another interval, we may map this interval to $[0, 2\pi)$. We will also refer to the observations (e.g. hue values) as angles accordingly.

The main drawback of circular data is that we can not directly apply a conventional distribution (e.g. Gaussian) as there is a dependence on the choice of the origin. For example, if we have two angles one at 0 and one at π, then if we select 0 as the origin then the mean of these angles is $\pi/2$. However, if we select $\pi/2$ as the origin, that is the interval is $[\pi/2, 5\pi/2)$, the mean is $3\pi/2$ due to the fact that the angle 0 is mapped to the angle 2π. In order to overcome these drawbacks, the von Mises distribution has been proposed. For a complete reference to its properties, the reader is referred to [3]. Here we summarize the key points.

The von Mises probability density function for an angle a is given by:

$$\mathcal{M}(a;\theta,m)=\frac{1}{2\pi I_0(m)}e^{m\cos(a-\theta)}, \tag{1}$$

where θ is the mean, m is the concentration (analogous to the inverse variance), $I_0(m)$ is the zeroth-order Bessel function of the first kind, which is defined as $I_0(m)=\int_0^{2\pi}e^{m\cos(t)}dt$. For large values of m the distribution becomes Gaussian and for $m=0$ it becomes uniform.

In order to estimate the parameters θ and m having some observed angles $\boldsymbol{A}=\{a_n\}_{n=1,\dots,N}$, we can use the maximum likelihood estimation. The log-likelihood of the model is given by:

$$\begin{aligned}\ln p(\boldsymbol{A};\theta,m) &= \ln\prod_{n=1}^{N}\mathcal{M}(a_n;\theta,m)\\ &= -N\ln(2\pi)-N\ln(I_0(m))+m\sum_{n=1}^{N}\cos(a_n-\theta).\end{aligned} \tag{2}$$

By maximizing (2) with respect to θ we obtain:

$$\theta=\tan^{-1}\left(\frac{\sum_{n=1}^{N}sin(a_n)}{\sum_{n=1}^{N}cos(a_n)}\right). \tag{3}$$

By maximizing (2) with respect to m we obtain the equation:

$$\frac{I_1(m)}{I_0(m)}=\frac{1}{N}\sum_{n=1}^{N}\cos(a_n-\theta), \tag{4}$$

which can be numerically solved, where $I_1(m)=I_0'(m)=\int_0^{2\pi}e^{m\cos(t)}\cos(t)dt$.

2.2. Von Mises mixture model

If the von Mises distribution is not flexible enough in order to model the observations, then we can use the von Mises mixture model as a linear superposition of von Mises components. The probability density function of an angle a for a von Mises mixture model can be defined as:

$$\mathcal{L}(a;\boldsymbol{\theta},\boldsymbol{m},\boldsymbol{\pi})=\sum_{k=1}^{K}\pi_k\mathcal{M}(a;\theta_k,m_k), \tag{5}$$

where K is the number of components, $\boldsymbol{\theta}=\{\theta_k\}_{k=1,\dots,K}$ are the means of the components, $\boldsymbol{m}=\{m_k\}_{k=1,\dots,K}$ are the concentrations of the components and $\boldsymbol{\pi}=\{\pi_k\}_{k=1,\dots,K}$ are the importances (weights) of the components.

In order to estimate the parameters we have to maximize the log-likelihood function with respect to these parameters, which can be achieved using the Expectation-Maximization algorithm [3,5]. We assume that we have observed N angles $\boldsymbol{A}=\{a_n\}_{n=1,\dots,N}$ and we want to estimate the parameters $\boldsymbol{\theta}$, $\boldsymbol{m}$ and $\boldsymbol{\pi}$ of the von Mises mixture model. The log-likelihood function is defined as:

$$\ln L(\boldsymbol{A};\boldsymbol{\theta},\boldsymbol{m},\boldsymbol{\pi})=\sum_{n=1}^{N}\ln\left(\mathcal{L}(a_n;\boldsymbol{\theta},\boldsymbol{m},\boldsymbol{\pi})\right). \tag{6}$$

We can define a set of random variables $\boldsymbol{Z}=\{\boldsymbol{z}_n\}_{n=1,\dots,N}$, where $\boldsymbol{z}_n$ is a K-dimensional binary random variable which has $z_{nk}=1$ if the nth angle is produced from the kth component, and $z_{nj}=0$ for $j\neq k$. Thus $\boldsymbol{z}_n$ can reveal from which component the observation

a_n has been generated. In practice, the values of the variables $\boldsymbol{Z}$ are not known, so they are called latent variables. If the value of the corresponding latent variable $\boldsymbol{z}_n$ is known for each observation a_n, then the set $\{\boldsymbol{A}, \boldsymbol{Z}\}$ is called the complete data set. The complete data log-likelihood function is given by:

$$\begin{aligned}&\ln L(\boldsymbol{A}, \boldsymbol{Z}; \boldsymbol{\theta}, \boldsymbol{m}, \boldsymbol{\pi})\\ &= \ln\left(\prod_{n=1}^{N}\prod_{k=1}^{K}(\pi_k \mathcal{M}(a_n; \theta_k, m_k))^{z_{nk}}\right)\\ &= \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk}(\ln \pi_k + \ln \mathcal{M}(a_n; \theta_k, m_k)).\end{aligned} \tag{7}$$

Due to the fact that the latent variables $\boldsymbol{Z}$ are not known, we can only use their posterior distribution:

$$\begin{aligned}p(\boldsymbol{Z}; \boldsymbol{A}, \boldsymbol{\theta}, \boldsymbol{m}, \boldsymbol{\pi}) &= \frac{p(\boldsymbol{A}; \boldsymbol{Z}, \boldsymbol{\theta}, \boldsymbol{m}, \boldsymbol{\pi})\, p(\boldsymbol{Z}; \boldsymbol{\theta}, \boldsymbol{m}, \boldsymbol{\pi})}{p(\boldsymbol{A}; \boldsymbol{\theta}, \boldsymbol{m}, \boldsymbol{\pi})}\\ &\propto \prod_{n=1}^{N}\prod_{k=1}^{K}(\pi_k \mathcal{M}(a_n; \theta_k, m_k))^{z_{nk}}.\end{aligned} \tag{8}$$

The expectation of the complete data log-likelihood is given by:

$$\begin{aligned}Q &= \sum_{\boldsymbol{Z}} p(\boldsymbol{Z}; \boldsymbol{A}, \boldsymbol{\theta}, \boldsymbol{m}, \boldsymbol{\pi}) \ln L(\boldsymbol{A}, \boldsymbol{Z}; \boldsymbol{\theta}, \boldsymbol{m}, \boldsymbol{\pi})\\ &= \sum_{n=1}^{N}\sum_{k=1}^{K} r(z_{nk})(\ln \pi_k + \ln \mathcal{M}(a_n; \theta_k, m_k)),\end{aligned} \tag{9}$$

where the $r(z_{nk})$ is the expectation of the latent variable z_{nk}:

$$\begin{aligned}r(z_{nk}) = E[z_{nk}] &= \frac{\sum_{z_{nk}} z_{nk}\, p(z_{nk}; a_n, \theta_k, m_k, \pi_k)}{\sum_{z_{nj}} p(z_{nj}; a_n, \theta_j, m_j, \pi_j)}\\ &= \frac{\pi_k \mathcal{M}(a_n; \theta_k, m_k)}{\sum_{j=1}^{K} \pi_j \mathcal{M}(a_n; \theta_j, m_j)}.\end{aligned} \tag{10}$$

Now, the maximization of (9) with respect to $\boldsymbol{\theta}, \boldsymbol{m}, \boldsymbol{\pi}$ can be easily achieved.

Thus, in order to evaluate the parameters $\boldsymbol{\theta}, \boldsymbol{m}, \boldsymbol{\pi}$ of the von Mises mixture model, we initialize these parameters to some values and repeatedly apply the E-step and M-step.

E-step:

$$r(z_{nk}) = \frac{\pi_k \mathcal{M}(a_n; \theta_k, m_k)}{\sum_{j=1}^{K} \pi_j \mathcal{M}(a_n; \theta_j, m_j)}. \tag{11}$$

M-step:

$$N_k = \sum_{n=1}^{N} r(z_{nk}), \tag{12}$$

$$\pi_k = \frac{N_k}{N}, \tag{13}$$

$$\theta_k = \tan^{-1}\frac{\sum_{n=1}^{N} r(z_{nk}) \sin(a_n)}{\sum_{n=1}^{N} r(z_{nk}) \cos(a_n)}, \tag{14}$$

$$\frac{I_1(m)}{I_0(m)} = \frac{1}{N_k}\sum_{n=1}^{N} r(z_{nk}) \cos(a_n - \theta_k). \tag{15}$$

Note that (15) is not in closed form but can be numerically solved with respect to the parameter m.

2.3. Weighted von Mises mixture model

In our approach, we want the pixels that are more likely to belong to the object to affect the estimation of the log-likelihood more than the pixel that are less likely to belong to the object. In the general case, we seek to adapt (6) in order to model the fact that some observations are considered more important than others:

$$\ln L(\boldsymbol{A}, \boldsymbol{w}; \boldsymbol{\theta}, \boldsymbol{m}, \boldsymbol{\pi}) = \sum_{n=1}^{N} w_n \ln\big(\mathcal{L}(a_n; \boldsymbol{\theta}, \boldsymbol{m}, \boldsymbol{\pi})\big), \tag{16}$$

where $\boldsymbol{w} = \{w_n\}_{n=1,\ldots,N}$ are the weights of the observations. The rational behind this approach is that the values a_n are used multiple times in (6) instead of just one, virtually changing the number of observations N. However, by aggregating their appearances into w_n we can keep the number of observations N constant, which yields the definition of the weighted log-likelihood (16).

By using the same approach the E-step Eq. (11) remains the same. On the other hand, the M-step Eqs. (13)–(15) change accordingly:

$$N_k = \sum_{n=1}^{N} w_n r(z_{nk}), \tag{17}$$

$$\pi_k = \frac{N_k}{\sum_{n=1}^{N} w_n}, \tag{18}$$

$$\theta_k = \tan^{-1}\frac{\sum_{n=1}^{N} w_n r(z_{nk}) \sin(a_n)}{\sum_{n=1}^{N} w_n r(z_{nk}) \cos(a_n)}, \tag{19}$$

$$\frac{I_1(m)}{I_0(m)} = \frac{1}{N_k}\sum_{n=1}^{N} w_n r(z_{nk}) \cos(a_n - \theta_k). \tag{20}$$

3. Tracking using the weighted von Mises mixture model

In this work we assume that the images employ the HSV color model and we use only the hue component, that is, each pixel is represented by a single value in the interval $[0, 2\pi)$. We use only the hue component as it provides a good representation of the target while being less computational intensive compared to other approaches like salient region detection [27]. Moreover, we assume that the object to be tracked can be represented by an ellipse. The ellipse has a center denoted by $\boldsymbol{y} = [y^{(1)}, y^{(2)}]^T$, where $y^{(1)}$ is the horizontal coordinate and $y^{(2)}$ is the vertical coordinate of the center in the image coordinates system, and a vector $\boldsymbol{h} = [h^{(1)}, h^{(2)}]^T$, where $h^{(1)}$ is the length of the horizontal semi-axis and $h^{(2)}$ is the length of the vertical semi-axis of the ellipse.

Having set the parameters $\boldsymbol{y}$ and $\boldsymbol{h}$, we can assign a weight to every pixel of the image by using a spatial kernel $k(t)$ which will assign greater weights to pixels near the center of the ellipse. More specifically, we use a kernel with exponential profile:

$$k(t) = \begin{cases} e^{(-t/\sigma)} & \text{if } \ t \le 1\\ 0 & \text{otherwise}\end{cases}. \tag{21}$$

Using this kernel, the weight $w_n(\boldsymbol{y})$ of the nth pixel with spatial coordinates $\boldsymbol{x}_n = [x_n^{(1)}, x_n^{(2)}]^T$ is given by:

$$w_n(\boldsymbol{y}) = k(M(\boldsymbol{x}_n; \boldsymbol{y}, \boldsymbol{h})), \tag{22}$$

where

$$\begin{aligned}M(\boldsymbol{x}_n; \boldsymbol{y}, \boldsymbol{h}) &= \left(\frac{x_n^{(1)} - y^{(1)}}{h^{(1)}}\right)^2 + \left(\frac{x_n^{(2)} - y^{(2)}}{h^{(2)}}\right)^2\\ &= (\boldsymbol{x}_n - \boldsymbol{y})^T \boldsymbol{H}^{-1} (\boldsymbol{x}_n - \boldsymbol{y}),\end{aligned} \tag{23}$$

is the squared Mahalanobis distance between $\boldsymbol{x}_n$ and $\boldsymbol{y}$ with diagonal covariance matrix $\boldsymbol{H} = diag(h^{(1)}, h^{(2)})$. By using the function M in (22) the drawback of the difference in axis lengths is overcome because the normalized pixel coordinates, for pixels inside the ellipse, are now in the interval [0, 1]. Thus, the weights $w_n(\boldsymbol{y})$ for pixels inside the ellipse are greater than zero, while pixels outside the ellipse have weights equal to zero.

3.1. First frame

We assume that the position of the ellipse is known in the first frame of the sequence. Thus, the objective here is to estimate the von Mises mixture model using the hue component of the pixels. The image consists of N pixels (with some given order, e.g. row-by-row), each pixel's weight $w_n(\boldsymbol{y})$ is given by (22) and each pixel's hue component is denoted by a_n. We can now estimate the von Mises mixture model parameters $\boldsymbol{\theta}$, $\boldsymbol{m}$, $\boldsymbol{\pi}$ using Eqs. (11), (18)–(20) of the EM algorithm.

3.2. Tracking in consecutive frames

In every frame of the video (except for the first), we know: (*i*) the center $\boldsymbol{y}$ and the size $\boldsymbol{h}$ of the ellipse which represents the target in the immediately previous frame and (*ii*) the parameters $\boldsymbol{\theta}$, $\boldsymbol{m}$, $\boldsymbol{\pi}$ of the von Mises mixture model which models the distribution of the hue component of the object's pixels. In order to estimate the center of the ellipse in the current frame, a gradient based technique will be used.

We seek to estimate the position $\boldsymbol{y}$ which maximizes the log-likelihood:

$$\ln L(\boldsymbol{A}, \boldsymbol{w}(\boldsymbol{y}); \boldsymbol{\theta}, \boldsymbol{m}, \boldsymbol{\pi}) = \sum_{n=1}^{N} w_n(\boldsymbol{y}) \ln\left(\mathcal{L}(a_n; \boldsymbol{\theta}, \boldsymbol{m}, \boldsymbol{\pi})\right). \tag{24}$$

This can be achieved by taking the derivative of (24) and setting it to zero. The derivative of (24) is defined as:

$$\frac{dL}{d\boldsymbol{y}} = \left[\frac{dL}{dy^{(1)}}, \frac{dL}{dy^{(2)}}\right]^T, \tag{25}$$

where:

$$\frac{dL}{dy^{(j)}} = \sum_{n=1}^{N} \frac{dw_n(\boldsymbol{y})}{dy^{(j)}} \mathcal{L}(a_n; \boldsymbol{\theta}, \boldsymbol{m}, \boldsymbol{\pi}). \tag{26}$$

The only term that depends on $\boldsymbol{y}$ is $w_n(\boldsymbol{y})$. By defining the negative derivative of the kernel function as $g(t) = -\frac{dk(t)}{dt}$, we have:

$$\frac{dk(M(\boldsymbol{x}_n; \boldsymbol{y}, \boldsymbol{h}))}{dy^{(j)}} \propto g(M(\boldsymbol{x}_n; \boldsymbol{y}, \boldsymbol{h})) \frac{x_n^{(j)} - y^{(j)}}{h^{(j)^2}}. \tag{27}$$

By substituting (27) into (26) we have:

$$\frac{dL}{dy^{(j)}} \propto \sum_{n=1}^{N} g(M(\boldsymbol{x}_n; \boldsymbol{y}, \boldsymbol{h})) \frac{x_n^{(j)} - y^{(j)}}{h^{(j)^2}} \mathcal{L}(a_n; \boldsymbol{\theta}, \boldsymbol{m}, \boldsymbol{\pi}). \tag{28}$$

By setting (28) equal to zero, we get the update formula (in vector form):

$$\boldsymbol{y} = \frac{\sum_{n=1}^{N} \boldsymbol{x}_n g(M(\boldsymbol{x}_n; \boldsymbol{y}, \boldsymbol{h})) \mathcal{L}(a_n; \boldsymbol{\theta}, \boldsymbol{m}, \boldsymbol{\pi})}{\sum_{n=1}^{N} g(M(\boldsymbol{x}_n; \boldsymbol{y}, \boldsymbol{h})) \mathcal{L}(a_n; \boldsymbol{\theta}, \boldsymbol{m}, \boldsymbol{\pi})}. \tag{29}$$

Thus, in every frame, starting from $\boldsymbol{y}$ estimated at the previous frame, we iteratively apply Eq. (29) in order to move the center $\boldsymbol{y}$ to a new position, until (24) decreases. Using this approach, the local maximum of the weighted likelihood (24) is computed, which is equivalent of finding the position that best matches the appearance model of the object [7]. In (29), the value of $\boldsymbol{y}$ in the left side of the equation is the new center while the value of $\boldsymbol{y}$ in the right side of the equation is the old center. Scale estimation can be performed by increasing and decreasing the ellipse size by a percentage (for example 10%) and choose the ellipse with the bigger average likelihood.

3.3. Implementation details

The execution time of the proposed algorithm can be improved as the values of the hue component are integers in the interval [0, 359].

First, in (16), the term $\mathcal{L}(a_n; \boldsymbol{\theta}, \boldsymbol{m}, \boldsymbol{\pi})$ depends only on the hue value of the pixel. Thus, we can aggregate the weight w_n of the pixels that have the same hue value to a new weight W_n. This is equivalent to creating a new image with 360 pixels having values from 0 to 359 and assign to each pixel the corresponding weight $W_i = \sum_{n=1}^{N} w_n \delta(i - a_n)$. The delta function is zero everywhere except the $\delta(0) = 1$. Using this approach, the number of the pixels used by the EM algorithm is constant and this makes also the time needed for the initialization on the first frame relatively constant. Here, we must highlight that this is not an approximation and the result is the same as it would be if we used every pixel of the original image.

Second, in (29), the term $\mathcal{L}(a_n; \boldsymbol{\theta}, \boldsymbol{m}, \boldsymbol{\pi})$ can be pre-calculated for all the values of a_n. The parameters $\boldsymbol{\theta}$, $\boldsymbol{m}$ and $\boldsymbol{\pi}$ are determined for the first frame and are keep constant in the subsequent frames. Thus, we can have an array of 360 values which can be computed after the estimation of the parameters $\boldsymbol{\theta}$, $\boldsymbol{m}$ and $\boldsymbol{\pi}$. During the tracking procedure, we can use this array instead of the Eqs. (16) and (1).

4. Experimental results

In this section we evaluate the performance of the optimized versions of the proposed algorithm and compare its performance with other state-of-the art algorithms. The single parameter of the algorithm is the number of components K, which is set a priori but its estimation is not a guess. The components of the von Mises mixture model, roughly represent the number of colors of the object. In our experiments, we used $K = 10$ but we noticed that if the actual number of colors of the object is smaller than K, some components will have $\pi_k = 0$. This effect may be easily clarified using an example. If the object to be tracked is a ball with only red and green patches, then the hue distribution of the object will have the majority of its values around the green and red points in the histogram needing only two components in the mixture. If the model contains more than two components, some of them could have mixing proportions $\pi_k = 0$ as a result of the EM algorithm. Then, we could delete these components from the mixture model to further speed up the computation.

First, we examine the performance benefits of the proposed optimized implementations that have been presented in Section 3.3. In Table 1, the performance of the different initialization strategies and likelihood estimations is presented. The first column indicates the size of the target in pixels. K indicates the number of components used by mixture based algorithms and B is the number of bins used by histogram based algorithms. The optimized implementations are described in Section 3.3. For the initialization, our algorithm uses only 360 pixels in order to estimate the parameters of the model through the EM algorithm. For the likelihood estimation, the algorithm employs a precomputed array with elements the values of the likelihood for a given set of parameters. The standard implementations do not use the ideas of Section 3.3. The column indicated by GMM is an implementation which uses a Gaussian mixture model. The column indicated by Hist refers to an approach that uses histograms in order to estimate the likelihood for each pixel and was originally proposed in the framework of

Table 1
Comparison of different initialization and likelihood estimation approaches presented in Section 3.3. GMM indicates a Gaussian mixture model and Hist a histogram approach employed in the mean shift algorithm. All times are in microseconds (10^{-6} s).

Size	Initialization				Likelihood estimation					
	Optimized		Standard		Optimized		Standard		GMM	Hist
	$K = 5$	$K = 10$	$K = 5$	$K = 10$	$K = 5$	$K = 10$	$K = 5$	$K = 10$	$K = 10$	$B = 16$
100 × 100	866 ± 20	1754 ± 60	114325 ± 2130	209932 ± 7705	10 ± 4	10 ± 3	2735 ± 148	5039 ± 144	4066 ± 154	202 ± 21
100 × 200	728 ± 19	1428 ± 60	280364 ± 3404	671230 ± 5922	20 ± 5	21 ± 5	5363 ± 173	10091 ± 209	8112 ± 175	393 ± 29
100 × 300	902 ± 21	1777 ± 72	412384 ± 4080	1056531 ± 36563	31 ± 6	31 ± 6	8057 ± 191	15061 ± 285	12152 ± 211	586 ± 40
200 × 200	764 ± 23	1468 ± 58	686327 ± 6888	1669781 ± 14176	42 ± 7	42 ± 9	10633 ± 224	19970 ± 249	19742 ± 257	789 ± 52
200 × 300	954 ± 27	1508 ± 62	851812 ± 8411	2652596 ± 372650	64 ± 9	64 ± 10	15920 ± 283	29980 ± 1079	29439 ± 310	1155 ± 18
300 × 300	1018 ± 52	1574 ± 84	1370657 ± 13097	4111642 ± 219722	94 ± 4	94 ± 3	24359 ± 225	45011 ± 278	40829 ± 295	1749 ± 19

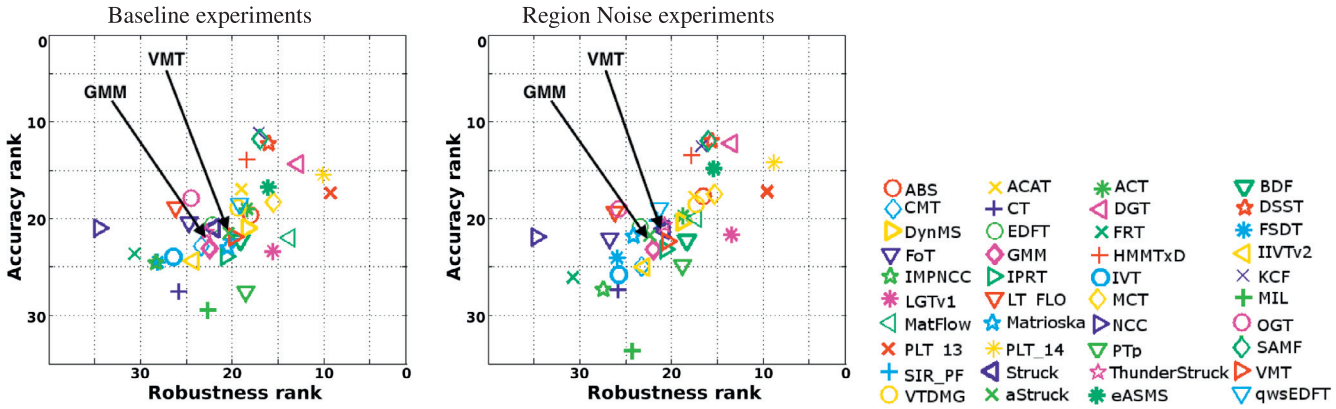

Fig. 1. Comparative evaluation of the proposed VMT with respect to state-of-the-art algorithms over all the video sequences of the VOT2014 data set. The plot is generated by the VOT 2014 toolset.

the mean shift tracker [7]. The results for the initialization step show that the time needed by the optimized implementation is relatively constant and around 1 ms. This time also includes the time needed in order to create the image with the 360 pixels and its aggregated weights. On the other hand, the time needed by the standard implementation increases as the number of pixels increases. The number of components K has the same impact on both approaches. If K is doubled, then the execution time is also doubled. This is expected, as the number of factors in the Eqs. (11), (13)–(15) is doubled. The results for the likelihood estimation step show that the proposed optimized method is around 200 times faster than the straightforward approach that evaluates the exponential and cosine functions in every pixel and 20 times faster than the approach that uses histograms. Moreover, the time needed by the optimized approach does not depend on the number of components K, as the likelihood of the mixture is evaluated beforehand for every possible input value. In the experiments above, all mixtures have the same number of components and produce the same results. The evaluation has been performed 10000 times and we present here the mean values. The machine that was used is a laptop with a dual core CPU at 2.26 GHz.

In order to evaluate the tracking performance of the proposed method we used the Visual Object Tracking (VOT) 2014 dataset (URL: http://votchallenge.net). The authors of the VOT dataset [20] provided a detailed description of the dataset and the evaluation methodology. Here we will provide a quick overview of the dataset and the toolkit. The VOT dataset consist of 25 color image sequences with one moving object in each sequence. In every image, the ground truth of the target has been manually annotated by bounding boxes. The information provided for the initialization of the tracker is the bounding box in the first frame. A target is considered to have lost the object when there is no overlap between the estimated target and the ground truth. If the tracker loses the object, then it is reinitialized in a subsequent frame. The evaluation of the tracker is performed N_{rep} times for each image sequence. The accuracy is associated with the average overlap per repetition per frame between the target's ground truth bounding box and the bounding box which was estimated by the tracker. The robustness index is associated with the average number of times the tracker failed per repetition. The performance of the tracker is evaluated in two sets of experiments. In the first set, which is called Baseline, the initialization of the target is done using the exact ground truth bounding box. In the second set, which is called Region Noise, a noisy initialization is done. The authors of the VOT dataset [20] has already tested the performance of 38 trackers and the tools needed to compare a new tracker with these state-of-the-art algorithms are included in the toolkit.

In Fig. 1, the plots for the Baseline and Region Noise experiments are presented. The proposed method is called VMT, which stands for von Mises Tracker. The horizontal and vertical axis denote the robustness rank and accuracy rank respectively. The proposed tracker, which is highlighted, is placed near the center of the plot, thus it has average performance in both measures with respect to the other algorithms. It is worth noting that the 38 other trackers constitute the state of the art in the framework of the VOT2014 dataset [20]. Moreover, the performance of a similar tracker, (denoted by GMM) that used the Gaussian distribution instead of the von Mises distribution is presented. Due to the fact that the Gaussian distribution can not model circular data in the beginning of the axis accurately, it exhibits an inferior performance than the von Mises distribution.

Table 2 presents the comparison of VMT with the top ranked methods of VOT2014. There are cases where VMT's performance is similar or even better than the top ranked methods. The cases that

Table 2
Performance of VMT with respect to the top ranked methods presented in [21]; DSST [8], SAMF [26] and KCF [13]. The attributes column contains the number of frames having: camera motion, illumination change, object motion, occlusion and object size change.

Seq.	Attributes	Baseline experiments								Region noise experiments							
		Accuracy rank				Robustness rank				Accuracy rank				Robustness rank			
		VMT	DSST	SAMF	KCF	VMT	DSST	SAMF	KCF	VMT	DSST	SAMF	KCF	VMT	DSST	SAMF	KCF
Ball	602, 0, 375, 0, 192	9.5	25.0	5.0	5.0	10.5	28.5	28.5	28.5	9.5	26.5	4.0	10.0	13.0	27.5	23.1	29.5
Basketball	725, 0, 238, 53, 40	14.0	14.5	4.0	14.5	23.5	15.0	6.5	6.5	12.5	17.5	9.5	29.0	26.0	13.6	2.5	14.6
Bicycle	271, 0, 178, 4, 100	30.9	13.0	11.6	10.0	36.0	11.0	11.0	11.0	35.0	14.0	4.0	4.0	37.0	7.5	8.5	7.1
Bolt	350, 0, 119, 0, 0	17.5	17.5	17.5	21.0	11.0	11.5	18.0	22.5	18.5	16.0	15.0	18.5	17.0	6.6	17.0	17.5
Car	252, 0, 162, 24, 157	32.0	3.0	20.0	4.5	14.5	15.0	15.0	15.0	32.0	2.0	17.0	5.0	15.5	15.0	15.0	15.0
David	770, 520, 706, 0, 135	21.0	7.5	5.5	4.5	12.0	12.5	12.5	12.5	20.0	9.0	6.5	6.5	11.0	11.5	11.5	11.5
Diving	219, 0, 54, 0, 74	11.0	5.1	30.1	28.7	4.0	11.2	34.0	34.0	8.5	9.0	29.5	29.5	3.5	19.1	37.5	36.5
Drunk	0, 0, 79, 0, 69	23.7	11.5	10.1	15.5	34.0	15.5	15.5	15.5	22.0	7.5	8.5	7.5	32.0	13.0	13.0	13.0
Fernando	274, 55, 176, 67, 142	24.0	24.0	19.6	16.1	20.5	11.5	11.5	11.5	26.0	18.5	15.5	13.6	17.4	14.2	8.8	11.0
Fish1	436, 0, 304, 0, 293	4.0	28.2	4.2	15.5	30.0	9.0	22.5	22.5	8.0	13.0	5.14	12.5	23.0	15.4	14.9	20.3
Fish2	278, 0, 106, 71, 132	12.5	9.5	16.3	20.2	4.0	16.0	20.5	28.5	11.0	11.5	15.5	26.0	4.0	16.5	17.7	25.5
Gymnastics	138, 0, 82, 0, 46	21.9	18.1	19.5	19.5	4.5	37.5	24.0	14.5	20.0	20.5	20.0	19.5	4.0	33.0	27.0	16.5
Hand1	0, 0, 239, 0, 171	14.0	38.5	13.5	13.0	12.7	19.6	24.1	24.1	13.0	24.0	25.0	18.5	14.0	23.0	34.0	26.7
Hand2	0, 0, 252, 0, 146	24.6	9.67	12.5	11.7	29.5	18.5	13.5	18.5	25.5	15.5	17.3	16.2	30.1	17.0	19.5	22.0
Jogging	307, 0, 305, 22, 124	28.0	10.0	7.7	9.5	15.5	16.0	16.0	16.0	26.5	12.0	14.0	12.5	26.7	16.0	15.5	16.0
Motocross	164, 0, 155, 0, 99	25.8	14.1	22.4	20.8	35.0	29.5	29.5	15.0	32.5	12.8	20.5	23.0	33.4	29.5	24.9	27.1
Polarbear	371, 0, 133, 0, 46	16.5	17.0	6.5	3.0	20.5	20.5	20.5	20.5	16.5	19.5	9.0	5.5	20.0	20.0	20.0	20.0
Skating	347, 400, 57, 42, 64	32.5	9.5	20.5	3.5	3.5	3.5	3.5	14.5	34.0	9.5	13.5	9.0	12.0	2.75	3.0	10.5
Sphere	201, 32, 189, 0, 34	16.0	1.5	3.8	3.0	16.5	16.5	16.5	16.5	13.0	6.0	4.5	4.5	17.5	17.5	17.5	17.5
Sunshade	172, 75, 170, 0, 22	28.6	9.5	10.0	10.0	13.0	13.5	13.5	13.5	29.5	9.5	12.0	10.0	9.0	9.5	9.5	9.5
Surfing	282, 0, 30, 30, 0	24.0	5.0	14.0	13.5	19.0	19.5	19.5	19.5	24.0	12.5	13.0	13.0	18.5	18.0	18.0	18.0
Torus	0, 0, 236, 0, 68	36.3	5.5	2.5	2.5	37.5	10.0	10.0	10.0	38.5	4.5	5.5	4.5	38.0	7.0	8.5	7.0
Trellis	569, 403, 391, 0, 93	22.0	2.0	2.0	2.0	8.0	8.5	8.5	8.5	21.0	2.0	2.0	2.0	7.0	7.5	7.5	7.5
Tunnel	731, 335, 356, 0, 114	23.0	1.0	7.5	3.0	34.0	11.0	11.0	11.0	27.0	1.5	8.0	1.5	36.0	9.7	9.7	9.7
Woman	597, 0, 236, 343, 43	22.5	6.6	8.0	9.0	36.0	17.0	17.0	17.0	24.5	5.5	5.0	10.5	33.5	19.3	15.5	9.0
Average		21.4	12.3	11.8	11.2	19.4	15.9	16.9	17.1	21.9	12.0	12.0	12.5	19.9	15.6	16.0	16.7

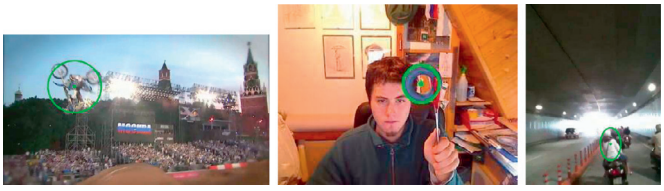

Fig. 2. Videos where the proposed method fails.

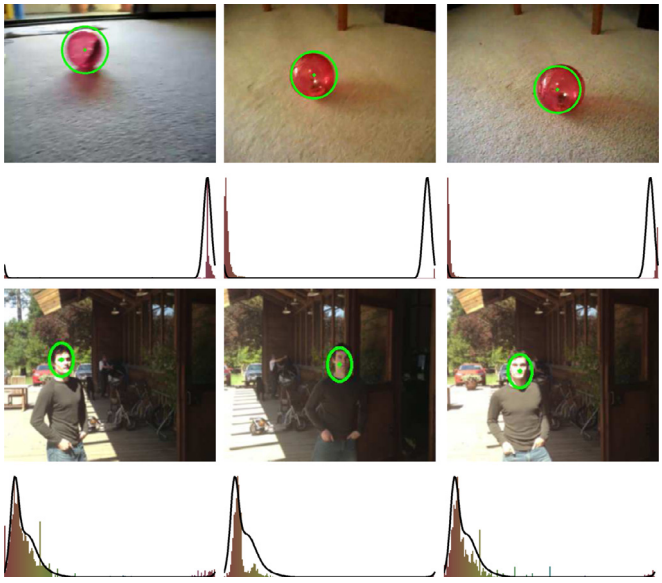

Fig. 3. Representative frames and the corresponding histograms with the estimated von Mises mixture superimposed on it.

most influence the performance of VMT are the shape of the target and the color distribution. Fig. 2 contains the first frame of some sequences where VMT does not perform well. In the motocross video, a large part of the ellipse which represents the target contains background elements. This is even worse in the torus video, where the center of the ellipse is entirely covered by the background. The situation is different in the tunnel video, where the target is white and its values have saturation near zero, making the hue component meaningless. This results from the fact that in order to model the black and white colors in the HSV color space we have to use both the value and the saturation components. On the other hand, VMT performs well if the shape of the target can be modeled by an ellipse and its color distribution by a von Mises mixture. Moreover, VMT is not strongly affected by camera motion, illumination change, object motion, occlusion or size change.

In Table 2, various cases concerning the performance of VMT are presented. For example hand1 and hand2 videos have nearly the same attributes but VMT performs better that the average in hand1. In hand2 sequence, the hand moves in front of the face many times, so VMT may follow the face instead of the hand. This is not the case in hand1, where the hand moves above the head in the majority of the frames. The person also opens and closes the hand in hand1, but this does not affect VMT. Finally, the proposed optimized VMT can process hundreds of frames per second by keeping the same accuracy and robustness as the standard VMT.

Some representative frames from the VOT's sphere and sunshade image sequence along with their corresponding histograms are presented in Fig. 3. In these figures, the first row shows some frames while the second row shows the corresponding histogram bins (computed from pixels inside the target) and the weighted von Mises mixture (estimated in the first frame and not changing along time) which is indicated by a continuous black line. For demonstration purposes, the histogram bins are normalized to [0 –

1]. In the sphere video, the target has a dominant red color, which in the beginning of the sequence is located mainly at the right side of the histogram while at the end the bins are shifted to the right and circularly appear at the left side of the histogram (due to the fact that the Hue component is periodic). Even in these cases, the proposed algorithm successfully tracks the object due to the fact that the von Mises distribution is periodic. More specifically, it assigns a likelihood to the pixels whose colors belong to the right side of the histogram which is sufficient to distinguish the object from the background. In the sunshade video, the target has two color components. The target oscillates back and forth from the sunshade to the sun, thus it illumination changes. As the hue component is immutable to changes in the brightness, the proposed method successfully follows the object between these transitions.

From these experiments, we can underpin some properties of the algorithm: a) The performance of VMT if not affected significantly when the initialization in the first frame of the image sequence does not contain exactly the target. This can be confirmed by the results of the Baseline and the Region Noise experiments, where the performance, both in terms of accuracy and robustness, remain nearly the same (Fig. 1). b) The tracker continues to perform well when the histogram of the color is shifted, like for example in the sphere sequence (Fig. 3).

5. Conclusion

The proposed algorithm eliminates drawbacks in kernel-based tracking which usually appear in standard applications and are due to periodic shift of the histogram bins of the target. Although some approaches have been proposed to handle this issue for linear spaces [16,17,22,44], these methods can not be directly applied for circular data as the determination of the origin of the axis affect the distance between two points. The VMT method proposed herein, employs the weighted von Mises mixture in order to estimate the target position within a maximum likelihood framework using a gradient based approach. As the von Mises is a continuous distribution, the likelihood is not affected by shifts in the histogram bins of the hue. Moreover, as the hue values are integers in [0, 359], the pre-calculation of key quantities of the likelihood of the mixture model, both in terms of computational time and memory. Although VMT uses the hue values, other circular data could be used, like the angle of the image gradient. Furthermore, a perspective of this work is to intergrade periodic and non-periodic spaces, as the full HSV space, in the same distribution.

Acknowledgment

Vasileios Karavasilis is funded by the Greek State Scholarships Foundation (IKY).

References

[1] H. Bao, M. Lin, Z. Chen, Robust visual tracking based on hierarchical appearance model, Neurocomputing 221 (2017) 108–122.
[2] V. Belagiannis, F. Schubert, N. Navab, S. Ilic, Segmentation based particle filtering for real-time 2d object tracking, in: Proceedings of the European Conference on Computer Vision (ECCV), 2012, pp. 842–855.
[3] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.
[4] G.R. Bradski, Computer vision face tracking for use in a perceptual user interface, Intel Technol. J. Q2 (1998).
[5] S. Calderara, A. Prati, R. Cucchiara, Mixtures of von Mises distributions for people trajectory shape analysis, IEEE Trans. Circuits Syst. Video Technol. 21 (4) (2011) 457–471.
[6] H.S. Choi, I.S. Kim, J.Y. Choi, Combining histogram-wise and pixel-wise matchings for kernel tracking through constrained optimization, Comput. Vision Image Understanding 118 (0) (2014) 61–70.
[7] D. Comaniciu, V. Ramesh, P. Meer, Kernel-based object tracking, IEEE Trans. Pattern Anal. Mach. Intell. 25 (5) (2003) 564–577.
[8] M. Danelljan, G. Hager, F.S. Khan, M. Felsberg, Accurate scale estimation for robust visual tracking, in: Proceedings of the British Machine Vision Conference, 2014.
[9] S. Duffner, C. Garcia, Pixeltrack: A fast adaptive algorithm for tracking non–rigid objects, in: IEEE International Conference on Computer Vision (ICCV'13), 2013, pp. 2480–2487.
[10] L. Ellis, N. Dowson, J. Matas, R. Bowden, Linear regression and adaptive appearance models for simultaneous modelling and tracking, Int. J. Comput. Vis. 95 (2) (2011) 154–179.
[11] M. Godec, P.M. Roth, H. Bischof, Hough-based tracking of non-rigid objects, in: IEEE International Conference on Computer Vision (ICCV'11), 2011, pp. 81–88.
[12] S. Hare, A. Saffari, P.H.S. Torr, Struck: Structured output tracking with kernels, in: IEEE International Conference on Computer Vision (ICCV'11), 2011, pp. 263–270.
[13] J. Henriques, R. Caseiro, P. Martins, J. Batista, High-speed tracking with kernelized correlation filters, IEEE Trans. Pattern Anal. Mach. Intell. 1 (2) (2014) 125–141.
[14] J.F. Henriques, R. Caseiro, P. Martins, J. Batista, High-speed tracking with kernelized correlation filters, IEEE Trans. Pattern Anal. Mach. Intell. 37 (3) (2015) 583–596.
[15] O. Javed, M. Shah, D. Comaniciu, A probabilistic framework for object recognition in video, in: International Conference on Image Processing, 2004 (ICIP '04), 4, 2004, pp. 2713–2716.
[16] V. Karavasilis, C. Nikou, A. Likas, Visual tracking using the earth mover's distance between Gaussian mixtures and Kalman filtering, Image Vis. Comput. 29 (5) (2011) 295–305.
[17] V. Karavasilis, C. Nikou, A. Likas, Visual tracking using spatially weighted likelihood of gaussian mixtures, Comput. Vision Image Understanding 140 (2015) 43–57.
[18] J. Kim, Z. Lin, I.S. Kweon, Rao-Blackwellized particle filtering with Gaussian mixture models for robust visual tracking, Comput. Vision Image Understanding 125 (1) (2014) 128–137.
[19] L. Kratz, K. Nishino, Going with the flow: Pedestrian efficiency in crowded scenes, in: Proceedings of the European Conference on Computer Vision (ECCV), 2012, pp. 558–572.
[20] M. Kristan, J. Matas, A. Leonardis, T. Vojir, R. Pflugfelder, G. Fernandez, G. Nebehay, F. Porikli, L. Cehovin, A novel performance evaluation methodology for single-target trackers, IEEE Trans. Pattern Anal. Mach. Intell. 38 (11) (2016) 2137–2155.
[21] M. Kristan, R. Pflugfelder, A. Leonardis, et al., The visual object tracking vot2014 challenge results, in: Proceedings of the European Conference on Computer Vision (ECCV) Visual Object Tracking Challenge Workshop, Zurich, Switzerland, 2014, pp. 98–111.
[22] I. Leichter, Mean shift trackers with cross-bin metrics, IEEE Trans. Pattern Anal. Mach. Intell. 34 (4) (2012) 695–706.
[23] I. Leichter, M. Lindenbaum, E. Rivlin, Mean shift tracking with multiple reference color histograms, Comput. Vision Image Understanding 114 (3) (2010) 400–408.
[24] S. Li, O. Wu, C. Zhu, H. Chang, Visual object tracking using spatial context information and global tracking skills, Comput. Vision Image Understanding 125 (2014) 1–15.
[25] S.-X. Li, H.-X. Chang, C.-F. Zhu, Adaptive pyramid mean shift for global real–time visual tracking, Image Vis. Comput. 28 (3) (2010) 424–437.
[26] Y. Li, J. Zhu, A scale adaptive kernel correlation filter tracker with feature integration, in: Proceedings of the European Conference on Computer Vision (ECCV), 2015, pp. 254–265.
[27] M. Lin, C. Zhang, Z. Chen, Global feature integration based salient region detection, Neurocomputing 159 (2015) 1–8.
[28] B. Liu, J. Huang, C. Kulikowski, L. Yang, Robust visual tracking using local sparse appearance model and k-selection, IEEE Trans. Pattern Anal. Mach. Intell. 35 (12) (2013) 2968–2981.
[29] S. Liwicki, G. Tzimiropoulos, S. Zafeiriou, M. Pantic, Efficient online subspace learning with an indefinite kernel for visual tracking and recognition, IEEE Trans. Neural Netw. Learn. Syst. 23 (2012) 1624–1636.
[30] I. Marras, G. Tzimiropoulos, S. Zafeiriou, M. Pantic, Online learning and fusion of orientation appearance models for robust rigid object tracking, Image Vis. Comput. 32 (10) (2014) 707–727.
[31] X. Mei, H. Ling, Robust visual tracking and vehicle classification via sparse representation, IEEE Trans. Pattern Anal. Mach. Intell. 33 (11) (2011) 2259–2272.
[32] G. Pons-Moll, A. Baak, J. Gall, L. Leal-Taixe, M. Muller, H. Seidel, B. Rosenhahn, Outdoor human motion capture using inverse kinematics and von Mises-Fisher sampling, in: Proceedings of International Conference on Computer Vision (ICCV'11), 2011, pp. 1243–1250.
[33] F. Seitner, A. Hanbury, Fast pedestrian tracking based on spatial features and color, in: Proceedings of the Computer Vision Winter Workshop, 2006, pp. 105–110.
[34] A. Smeulders, D. Chu, R. Cucchiara, S. Calderara, A. Dehghan, M. Shah, Visual tracking: an experimental survey, IEEE Trans. Pattern Anal. Mach. Intell. 36 (7) (2014) 1442–1468.
[35] B. Song, T.-Y. Jeng, E. Staudt, A.K. Roy-Chowdhury, A stochastic graph evolution framework for robust multi-target tracking, in: Proceedings of the European Conference on Computer Vision (ECCV), 2010, pp. 605–619.
[36] R. Tao, E. Gavves, A.W. Smeulders, Siamese instance search for tracking, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR'16), 2016, pp. 1–10.
[37] G. Tzimiropoulos, S. Zafeiriou, M. Pantic, Sparse representations of image gradient orientations for visual recognition and tracking, in: Proceedings of Workshop on CVPR for Human Behaviour Analysis (CVPR-W11), 2011, pp. 26–33.

[38] T. Vojir, J. Noskova, J. Matas, Robust scale-adaptive mean-shift for tracking, in: Proc. of the 18th Scandinavian Conf. on Image Analysis (SCIA), 2013, pp. 652–663.
[39] D. Wang, H. Lu, M.-H. Yang, Online object tracking with sparse prototypes, IEEE Trans. Image Process. 22 (1) (2013) 314–325.
[40] F. Yang, H. Lu, M.-H. Yang, Robust superpixel tracking, IEEE Trans. Image Process. 23 (4) (2014) 1639–1651.
[41] K. Zhang, L. Zhang, M.-H. Yang, Real-time object tracking via online discriminative feature selection, IEEE Trans. Image Process. 22 (12) (2013) 4664–4677.
[42] K. Zhang, L. Zhang, M.-H. Yang, Fast compressive tracking, IEEE Trans. Pattern Anal. Mach. Intell. 36 (10) (2014) 2002–2015.
[43] L. Zhang, L. van der Maaten, Preserving structure in model-free tracking, IEEE Trans. Pattern Anal. Mach. Intell. 36 (4) (2014) 756–769.
[44] Q. Zhao, H. Tao, Differential earth mover's distance with its application to visual tracking, IEEE Trans. Pattern Anal. Mach. Intell. 32 (5) (2010) 274–287.
[45] K. Zimmermann, J. Matas, T. Svoboda, Tracking by an optimal sequence of linear predictors, IEEE Trans. Pattern Anal. Mach. Intell. 31 (4) (2009) 677–692.