Pattern Recognition Letters 29 (2008) 2165–2173

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

A spatial-color mean-shift object tracking algorithm with scale and orientation estimation

Jwu-Sheng Hu *, Chung-Wei Juan, Jyun-Ji Wang

Department of Electrical and Control Engineering, National Chiao Tung University, Hsinchu 300, Taiwan, ROC

ARTICLE INFO

Article history:
Received 30 October 2007
Received in revised form 26 May 2008
Available online 22 August 2008

Communicated by Y. Ma

Keywords:
Mean-shift
Object tracking
Principle component analysis
Object deformation

ABSTRACT

In this paper, an enhanced mean-shift tracking algorithm using joint spatial-color feature and a novel similarity measure function is proposed. The target image is modeled with the kernel density estimation and new similarity measure functions are developed using the expectation of the estimated kernel density. With these new similarity measure functions, two similarity-based mean-shift tracking algorithms are derived. To enhance the robustness, the weighted-background information is added into the proposed tracking algorithm. Further, to cope with the object deformation problem, the principal components of the variance matrix are computed to update the orientation of the tracking object, and corresponding eigenvalues are used to monitor the scale of the object. The experimental results show that the new similarity-based tracking algorithms can be implemented in real-time and are able to track the moving object with an automatic update of the orientation and scale changes.

1. Introduction

In visual tracking, object representation is an important issue, because it can describe the correlation between the appearance and the state of the object. An appropriate object representation makes the target model more distinguishable from the background, and achieves a better tracking result. Comaniciu et al. (2003) used the spatial kernels weighted by a radially symmetric normalized distance from the object center to represent blob objects. This representation makes mean-shift tracking more efficient. Radially symmetric kernel preserves representation of the distance of a pixel from the center even the object has a large set of transformations, but this approach only contains the color information of the target and the spatial information is discarded. Parameswaran et al. (2006) proposed the tunable kernels for tracking, which simultaneously encodes appearance and geometry that enable the use of mean-shift iterations. A method was presented to modulate the feature histogram of the target that uses a set of spatial kernels with different bandwidths to encode the spatial information. Under certain conditions, this approach can solve the problem of similar color distribution blocks with different spatial configurations.

Another problem in the visual tracking is how to track the scale of object. In the work by Comaniciu et al. (2003), the mean-shift algorithm is run several times, and for each different window size, the similarity measure Bhattacharyya coefficient is computed for comparison. The window size yielding the largest Bhattacharyya coefficient, i.e. the most similar distribution, is chosen as the updated scale. Parameswaran et al. (2006), Birchfield and Rangarajan (2005) and Porikli and Tuzel (2005) use the similar variation method to solve the scale problem. But this method is not always stable, and easily makes the tracker lose the target. Collins (2003) extended the mean-shift tracker by adapting Lindeberg's theory (Lindeberg, 1998) of feature scale selection based on local maxima of differential scale-space filters. It uses blob tracking and a scale kernel to accurately capture the target's variation in scale. But the detailed iteration method was not described in the paper. An EM-like algorithm (Zivkovic and Krose, 2004) was proposed to estimate simultaneously the position of the local mode and used the covariance matrix to describe the approximate shape of the object. However, implementation details such as deciding the scale size from the covariance matrix were not given. Other attempts were made to study different representation methods. Zhang et al. (2004) represented the object by a kernel-based model, which offers more accurate spatial–spectral description than general blob models. Later, they further extend the work to cope with the scaling and rotation problem under the assumption of affine transformation (Zhang et al., 2005). Zhao and Tao (2005) proposed the color correlogram to use the correlation of colors to solve the related problem. But these methods did not consider the influence of complex background.

This work extends the traditional mean-shift tracking algorithm to improve the performance of arbitrary object tracking. At the same time, the proposed method tries to estimate the scale and orientation of the target. This idea is similar to the CAMSHIFT

* Corresponding author. Tel.: +886 3 5712121x54318; fax: +886 3 5715998.
E-mail address: jshu@cn.nctu.edu.tw (J.-S. Hu).

0167-8655/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2008.08.007

algorithm (Bradski, 1998) except spatial probability information as well as background influence are considered. The subject of this paper is divided into two parts. The first part is to develop the new spatial-color mean-shift trackers for the purpose of capturing the target more accurately than the traditional mean-shift tracker. The second part is to develop a method for solving the scale and orientation problem mentioned above. The solution, though seems straightforward, has never been proposed in literature. The effectiveness proved by experiments shows a further enhancement on the mean-shift algorithm.

2. Model definition

Birchfield and Rangarajan (2005) proposed the concept of spatial histogram, or spatiogram, in which each histogram bin contains the mean and covariance information of the locations of the pixels belonging to that bin. This idea involves the spatially weighted mean and covariance of the locations of the pixels. The spatiogram captures the spatial information of the general histogram bins. However, as shown in Fig. 1, if cyan and blue belong to the same bin, these two blocks have the same spatiogram, even though they have different color patterns.

Let the image of interest have M pixels and the associated color space can be classified into B bins. For example, in RGB color space, if each color is divided into 8 intervals, the total number of bins is 512. The image can be described as $I_x = \{\boldsymbol{x}_i, \boldsymbol{c}_{\boldsymbol{x}_i}, b_{\boldsymbol{x}_i}\}_{i=1,\ldots,M}$ where $\boldsymbol{x}_i$ is the location of pixel i with color feature vector $\boldsymbol{c}_{\boldsymbol{x}_i}$ which belongs to the $b_{\boldsymbol{x}_i}$th bin. The color feature vector has the dimension d which is the color channels for the pixel (for example, in RGB color space, $d = 3$ and $\boldsymbol{c}_{\boldsymbol{x}_i} = (R_{\boldsymbol{x}_i}, G_{\boldsymbol{x}_i}, B_{\boldsymbol{x}_i})$). To keep the robustness of color description of the spatiogram, we extend the spatiogram and define a new joint spatial-color model of the image I_x as

$$h_{I_x}(b) = \left\langle n_b, \boldsymbol{\mu}_{P,b}, \sum_{P,b}, \boldsymbol{\mu}_{C,b}, \sum_{C,b} \right\rangle, \quad b = 1, \ldots, B \tag{1}$$

where n_b, $\mu_{P,b}$, and $\sum_{P,b}$ are the same as the spatiogram proposed by Birchfield and Rangarajan (2005). Namely, n_b is the number of pixels, $\mu_{P,b}$ the mean vector of pixel locations, and $\sum_{P,b}$ the covariance matrix of pixel locations belonging to the bth bin. In (1), we add two additional elements. $\mu_{C,b}$ is the mean vector of the color feature vectors and $\sum_{C,b}$ is the associated covariance matrix.

3. Spacial-colour mean-shift object tracking algorithm

Using the spatial-color feature and the concept of expectation (Yang et al., 2005), two different tracking algorithms are proposed as the following.

3.1. Spatial-color mean-shift tracking algorithm (tracker 1)

The p.d.f. of the selected pixel $\boldsymbol{x}, \boldsymbol{c}_{\boldsymbol{x}}, b_{\boldsymbol{x}}$ in the image model $h_{I_x}(b)$ (see (1)) can be estimated using kernel density function.

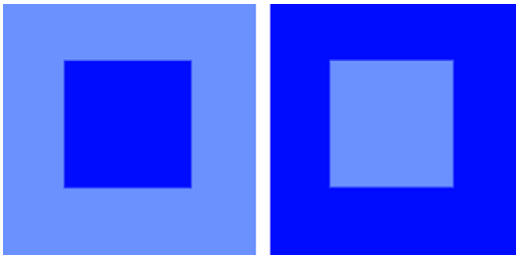

Fig. 1. Illustration of the same spatial information with different color distribution for one bin.

$$p(\boldsymbol{x}, \boldsymbol{c}_{\boldsymbol{x}}, b_{\boldsymbol{x}}) = \frac{1}{B}\sum_{b=1}^{B} K_P\left(\boldsymbol{x} - \boldsymbol{\mu}_{P,b}, \sum_{P,b}\right) K_C\left(\boldsymbol{c}_{\boldsymbol{x}} - \boldsymbol{\mu}_{C,b}, \sum_{C,b}\right)\delta(b_{\boldsymbol{x}} - b) \tag{2}$$

where K_P and K_C are multivariate Gaussian kernel functions and can be regarded as the spatially weighted and color-feature weighted function respectively. It is also possible to use a smooth kernel such as Gaussian (Yang et al., 2005). Using the concept of the expectation of the estimated kernel density, we can define a new similarity measure function between the model $h_{I_x}(b)$ and candidate $I_y = \{\boldsymbol{y}_j, \boldsymbol{c}_{\boldsymbol{y}_j}, b_{\boldsymbol{y}_j}\}_{j=1,\ldots,N}$ as

$$\begin{aligned} J(I_x, I_y) = J(\boldsymbol{y}) &= \frac{1}{N}\sum_{j=1}^{N} p(\boldsymbol{y}_j, \boldsymbol{c}_{\boldsymbol{y}_j}, b_{\boldsymbol{y}_j}) \\ &= \frac{1}{NB}\sum_{j=1}^{N}\sum_{b=1}^{B} K_P\left(\boldsymbol{y}_j - \boldsymbol{\mu}_{P,b}, \sum_{P,b}\right) K_C\left(\boldsymbol{c}_{\boldsymbol{y}_j} - \boldsymbol{\mu}_{C,b}, \sum_{C,b}\right)\delta(b_{\boldsymbol{y}_j} - b) \end{aligned} \tag{3}$$

The spatial-color model in (2) under measures like (3) might be sensitive to small spatial changes. This problem was discussed by O'Conaire et al. (2007) and Birchfield and Rangarajan (2007). However, this model also gives advantages of orientation estimation. As shown in Fig. 2, if there is no deformation between candidate and target, and the distance of motion is not excessively large between two adjacent frames, we can consider the motion of object of two frames as a pure translation. Under these assumptions, the center of target model $\boldsymbol{x}_0$ is proportional to the center of candidate $\boldsymbol{y}$ in the candidate image. As a result, we can normalize the pixels location and then obtain the new similarity measure function as the following:

$$\begin{aligned} J(\boldsymbol{y}) = \frac{1}{NB}\sum_{j=1}^{N}\sum_{b=1}^{B} & K_P\left(\boldsymbol{y}_j - \boldsymbol{y} - (\boldsymbol{\mu}_{P,b} - \boldsymbol{x}_0), \sum_{P,b}\right) \\ & \times K_C\left(\boldsymbol{c}_{\boldsymbol{y}_j} - \boldsymbol{\mu}_{C,b}, \sum_{C,b}\right)\delta(b_{\boldsymbol{y}_j} - b) \end{aligned} \tag{4}$$

The best candidate for matching can be found by computing the maximum value of the similarity measure. Let the gradient of the similarity function with respect to the vector $\boldsymbol{y}$ equal to $\mathbf{0}$, i.e., $\nabla J(\boldsymbol{y}) = \mathbf{0}$, we can obtain the new position $\boldsymbol{y}_{\text{new}}$ of the target to be tracked,

$$\begin{aligned} & \nabla J(\boldsymbol{y}) = \mathbf{0} \\ & \Rightarrow \frac{1}{NB}\sum_{j=1}^{N}\sum_{b=1}^{B}\left(\sum_{P,b}\right)^{-1}(\boldsymbol{y}_j - \boldsymbol{y} - \boldsymbol{\mu}_{P,b} + \boldsymbol{x}_0)K_P K_C \delta(b_{\boldsymbol{y}_j} - b) = \mathbf{0} \\ & \Rightarrow \left\{\sum_{j=1}^{N}\sum_{b=1}^{B}\left(\sum_{P,b}\right)^{-1} K_P K_C \delta(b_{\boldsymbol{y}_j} - b)\right\}(\boldsymbol{y} - \boldsymbol{x}_0) \\ & = \sum_{j=1}^{N}\sum_{b=1}^{B}\left(\sum_{P,b}\right)^{-1}(\boldsymbol{y}_j - \boldsymbol{\mu}_{P,b)}K_P K_C \delta(b_{\boldsymbol{y}_j} - b) \end{aligned}$$

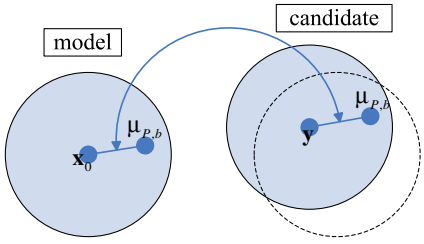

Fig. 2. Illustration of pure translation.

$$\boldsymbol{y}-\boldsymbol{x}_0=\left\{\sum_{j=1}^{N}\sum_{b=1}^{B}\left(\sum_{P,b}\right)^{-1}K_PK_C\delta(b_{\boldsymbol{y}_j}-b)\right\}^{-1}\times\left\{\sum_{j=1}^{N}\sum_{b=1}^{B}\left(\sum_{P,b}\right)^{-1}(\boldsymbol{y}_j-\boldsymbol{\mu}_{P,b})K_PK_C\delta(b_{\boldsymbol{y}_j}-b)\right\} \tag{5}$$

As a result, the new position $\boldsymbol{y}_{\text{new}}$ is described as (6).

$$\boldsymbol{y}_{\text{new}}=\left\{\sum_{j=1}^{N}\sum_{b=1}^{B}\left(\sum_{P,b}\right)^{-1}K_PK_C\delta(b_{\boldsymbol{y}_j}-b)\right\}^{-1}\times\left\{\sum_{j=1}^{N}\sum_{b=1}^{B}\left(\sum_{P,b}\right)^{-1}(\boldsymbol{y}_j-\boldsymbol{\mu}_{P,b})K_PK_C\delta(b_{\boldsymbol{y}_j}-b)\right\}+\boldsymbol{x}_0 \tag{6}$$

where

$$K_P=K_P\left(\boldsymbol{y}_j-\boldsymbol{y}_{\text{old}}-\boldsymbol{\mu}_{P,b}+\boldsymbol{x}_0,\sum_{P,b}\right)=\frac{1}{2\pi|\sum_{P,b}|^{1/2}}\mathrm{e}^{\left(-\frac{1}{2}\left(\boldsymbol{y}_j-\boldsymbol{y}_{\text{old}}-\boldsymbol{\mu}_{P,b}+\boldsymbol{x}_0\right)^{\mathrm{T}}\left(\sum_{P,b}\right)^{-1}\left(\boldsymbol{y}_j-\boldsymbol{y}_{\text{old}}-\boldsymbol{\mu}_{P,b}+\boldsymbol{x}_0\right)\right)} \tag{7}$$

and

$$K_C=K_C\left(\boldsymbol{c}_{\boldsymbol{y}_j}-\boldsymbol{\mu}_{C,b},\sum_{C,b}\right)=\frac{1}{(2\pi)^{3/2}|\sum_{C,b}|^{1/2}}\mathrm{e}^{\left(-\frac{1}{2}\left(\boldsymbol{c}_{\boldsymbol{y}_j}-\boldsymbol{\mu}_{C,b}\right)^{\mathrm{T}}\left(\sum_{C,b}\right)^{-1}\left(\boldsymbol{c}_{\boldsymbol{y}_j}-\boldsymbol{\mu}_{C,b}\right)\right)} \tag{8}$$

Eq. (6) is the mean shift vector as well as an iterative function with respect to $\boldsymbol{y}$. In the sequel, we define $\boldsymbol{y}_{\text{old}}$ as the current position.

3.2. Tracking algorithm with reduced complexity (tracker 2)

Based on the definition of the p.d.f. in (2), the kernel density functions of (7) and (8) have to be computed in each iterative step during tracking. Therefore, it is possible to reduce the computational complexity when the variation of the target image is small. Rewrite (2) as

$$p(b)=\frac{1}{M}\sum_{i=1}^{M}K_P\left(\boldsymbol{x}_i-\boldsymbol{\mu}_{P,b(i)},\sum_{P,b(i)}\right)K_C\left(\boldsymbol{c}_{\boldsymbol{x}_i}-\boldsymbol{\mu}_{C,b(i)},\sum_{C,b(i)}\right)\delta(b-b(i)) \tag{9}$$

where $b(i)$ is the color bin to which pixel i belongs. We can derive new kernel density estimation functions as

$$K_P\left(\boldsymbol{x}_i-\boldsymbol{\mu}_{P,b(i)},\sum_{P,b(i)}\right)=\frac{1}{2\pi|\sum_{P,b(i)}|^{1/2}}\mathrm{e}^{\left(-\frac{1}{2}\left(\boldsymbol{x}_i-\boldsymbol{\mu}_{P,b(i)}\right)^{\mathrm{T}}\left(\sum_{P,b(i)}\right)^{-1}\left(\boldsymbol{x}_i-\boldsymbol{\mu}_{P,b(i)}\right)\right)} \tag{10}$$

$$K_C\left(\boldsymbol{c}_{\boldsymbol{x}_i}-\boldsymbol{\mu}_{C,b(i)},\sum_{C,b(i)}\right)=\frac{1}{(2\pi)^{3/2}|\sum_{C,b(i)}|^{1/2}}\mathrm{e}^{\left(-\frac{1}{2}\left(\boldsymbol{c}_{\boldsymbol{x}_i}-\boldsymbol{\mu}_{C,b(i)}\right)^{\mathrm{T}}\left(\sum_{C,b(i)}\right)^{-1}\left(\boldsymbol{c}_{\boldsymbol{x}_i}-\boldsymbol{\mu}_{C,b(i)}\right)\right)} \tag{11}$$

K_P and K_C are also the spatially weighted and color-feature weighted functions which depend on the image model. Using the similar concept of the expectation of the estimated kernel density, another new similarity measure function between the model and candidate $I_y=\{\boldsymbol{y}_j,\boldsymbol{c}_{\boldsymbol{y}_j},b_{\boldsymbol{y}_j}\}_{j=1,\ldots,N}$ can be defined as

$$J(I_x,I_y)=J(\boldsymbol{y})=\frac{1}{N}\sum_{j=1}^{N}G(\widetilde{\boldsymbol{y}}_j)p(b_{\boldsymbol{y}_j}) \tag{12}$$

where $\widetilde{\boldsymbol{y}}_j=\frac{1}{\alpha}(\boldsymbol{y}-\boldsymbol{y}_j)$ and $\boldsymbol{y}$ is the center of the candidate image. $G(\widetilde{\boldsymbol{y}}_j)$ is a spatially weighted function depending on the candidate image and $\alpha=\underset{j=1\sim N}{\text{Max}}\|\boldsymbol{y}-\boldsymbol{y}_j\|$.

The best candidate is obtained by finding the maximum value of the similarity measure, i.e.,

$$\begin{aligned}&\nabla J(\boldsymbol{y})=\boldsymbol{0}\\&\Rightarrow\frac{1}{\alpha N}\sum_{j=1}^{N}(\boldsymbol{y}-\boldsymbol{y}_j)G'(\widetilde{\boldsymbol{y}}_j)p(b_{\boldsymbol{y}_j})=\boldsymbol{0}\\&\Rightarrow\boldsymbol{y}\sum_{j=1}^{N}G'(\widetilde{\boldsymbol{y}}_j)p(b_{\boldsymbol{y}_j})=\sum_{j=1}^{N}\boldsymbol{y}_jG'(\widetilde{\boldsymbol{y}}_j)p(b_{\boldsymbol{y}_j})\\&\Rightarrow\boldsymbol{y}=\frac{\sum_{j=1}^{N}\boldsymbol{y}_jG'(\widetilde{\boldsymbol{y}}_j)p(b_{\boldsymbol{y}_j})}{\sum_{j=1}^{N}G'(\widetilde{\boldsymbol{y}}_j)p(b_{\boldsymbol{y}_j})}\end{aligned} \tag{13}$$

The spatially weighted term $G'(\widetilde{\boldsymbol{y}}_j)$ can be derived by choosing function G as the Epanechnikov kernel function:

$$K(\boldsymbol{x})=\begin{cases}\frac{1}{2C_d}(d+2)(1-\|\boldsymbol{x}\|^2), & \text{if } \|\boldsymbol{x}\|<1\\0, & \text{otherwise}\end{cases} \tag{14}$$

where d is the dimension of space, C_d is the volume of the unit d-Dimensional sphere. Let $K(\boldsymbol{x})=k(\|\boldsymbol{x}\|^2)$, we obtain

$$k(x)=\begin{cases}\frac{1}{2C_d}(d+2)(1-x), & \text{if } x<1\\0, & \text{otherwise}\end{cases}$$

For two-dimensional image processing applications ($d=2$ and $C_d=\pi$), the kernel function is reduced to

$$k(x)=\begin{cases}\frac{1}{2\pi}(2+2)(1-x)=\frac{2}{\pi}(1-x), & \text{if } x<1\\0, & \text{otherwise}\end{cases} \tag{15}$$

Assigning $G(x)=k(x)$, the derivative of G becomes a constant as

$$G\prime(x)=k\prime(x)=-\frac{2}{\pi} \tag{16}$$

The result is simple and easy to compute. Finally, by substituting (16) into (13), we can obtain the second similarity-based mean-shift algorithm as follows:

$$\boldsymbol{y}_{\text{new}}=\frac{\sum_{j=1}^{N}\boldsymbol{y}_jp(b_{\boldsymbol{y}_j})}{\sum_{j=1}^{N}p(b_{\boldsymbol{y}_j})} \tag{17}$$

3.3. Weighted-background information

Like most of the tracking algorithms, the proposed method may arrive at an incorrect result if the background contains similar information in foreground. The problem becomes more serious if the scale and orientation of the target have to be followed. One way to reduce the influence of background is to apply a weighting function to the image surrounding the target. The combination of the weighting function with the spatial-color mean-shift tracking algorithms proposed before is explained below.

Let $N_{F,b}$ be the normalized histogram of the foreground of the bth bin ($\sum_b N_{F,b}=1$), and $N_{O,b}$ the normalized histogram of the background of the bth bin ($\sum_b N_{O,b}=1$). The histogram of

background is computed in the region around the foreground (target). The size is equal to two times the target size and the area is equal to three times the target area. Define the *background influence factor* of the bth bin as $\frac{N_{F,b}}{N_{O,b}}$ for $N_{O,b} \neq 0$. The maximum value of the factor for all bins are defined as $\beta = \max_{\substack{b=1\sim B,\\ N_{O,b}\neq 0}} \left(\frac{N_{F,b}}{N_{O,b}}\right)$. When $\beta \ll 1$, certain bins in background contain more related features than the corresponding foreground bins. This should results in a small background weighting factor for those bins. Note that we exclude the cases when $N_{O,b} = 0$ in computing β. Therefore, we should also make the background weighting factor small for the cases when $\beta \approx 1$. Based on the analysis, the background weighting factor can be defined as

$$W_b = \begin{cases} \left(1 - e^{-\frac{\beta}{\beta_0}}\right) \frac{1}{\beta} \frac{N_{F,b}}{N_{O,b}}, & \text{if } N_{O,b} \neq 0 \\ 1, & \text{if } N_{F,b} \neq 0 \text{ and } N_{O,b} = 0 \\ 0, & \text{otherwise} \end{cases} \tag{18}$$

where β_0 is a constant. Note that when $N_{F,b} \neq 0$ and $N_{O,b} = 0$, the background has no influence to the foreground at the bth bin. Therefore, it is given the largest weighting in (18). The weighted-background information is added into the mean-shift tracking algorithms developed in (6) and (17), and the algorithms is derived again as

$$\boldsymbol{y}_{\text{new}} = \left\{ \sum_{j=1}^{N} \sum_{b=1}^{B} \left(\sum_{P,b} \right)^{-1} W_{b_{\boldsymbol{y}_j}} K_P K_C \delta[b_{\boldsymbol{y}_j} - b] \right\}^{-1} \times \left\{ \sum_{j=1}^{N} \sum_{b=1}^{B} \left(\sum_{P,b} \right)^{-1} (\boldsymbol{y}_j - \boldsymbol{\mu}_{P,b}) W_{b_{\boldsymbol{y}_j}} K_P K_C \delta[b_{\boldsymbol{y}_j} - b] \right\} + \boldsymbol{x}_0 \tag{19}$$

and

$$\boldsymbol{y}_{\text{new}} = \frac{\sum_{j=1}^{N} \boldsymbol{y}_j W_{b_{\boldsymbol{y}_j}} p(b_{\boldsymbol{y}_j})}{\sum_{j=1}^{N} W_{b_{\boldsymbol{y}_j}} p(b_{\boldsymbol{y}_j})} \tag{20}$$

3.4. Update of scale and orientation

The characteristic values of the covariance matrix of the spatial-color distribution can be utilized to represent the scale and orientation of the target. A similar idea was proposed in the algorithm called CAMSHIFT (Bradski, 1998). From the experimental results shown later, this simple calculation provides a fairly robust scale and orientation tracking performance which greatly enhances the capability of mean-shift algorithm.

We define several new elements as follows. $\boldsymbol{\mu}_{\text{T}}$ is the total mean vector of the locations of all pixels in the target, $\sum'_W$ is the within-class covariance matrix of the B bins by adding the background weighting. $\sum'_B$ is the between-class covariance matrix of the B bins by adding the background weighting. $\sum'_{\text{T}}$ is the total covariance matrix of locations of all data by adding the background weighting.

$$\boldsymbol{\mu}_{\text{T}} = \frac{1}{M} \sum_{b=1}^{B} n_b \boldsymbol{\mu}_{P,b} \tag{21}$$

$$\sum_{W}^{'} = \sum_{b=1}^{B} \sum_{i=1}^{M} W_b (\boldsymbol{x}_i - \boldsymbol{\mu}_{P,b})(\boldsymbol{x}_i - \boldsymbol{\mu}_{P,b})^{\text{T}} \delta[b_{\boldsymbol{x}_i} - b] \tag{22}$$

$$\sum_{B}^{'} = \sum_{b=1}^{B} W_b n_b (\boldsymbol{\mu}_{P,b} - \boldsymbol{\mu}_{\text{T}})(\boldsymbol{\mu}_{P,b} - \boldsymbol{\mu}_{\text{T}})^{\text{T}} \tag{23}$$

$$\sum_{\text{T}}^{'} = \sum_{i=1}^{M} W_b (\boldsymbol{x}_i - \boldsymbol{\mu}_{\text{T}})(\boldsymbol{x}_i - \boldsymbol{\mu}_{\text{T}})^{\text{T}} \delta[b_{\boldsymbol{x}_i} - b] \tag{24}$$

It can be shown first that $\sum'_{\text{T}}$ can be computed from the between-class covariance matrix and within-class covariance matrix as

$$\begin{aligned} \sum_{\text{T}}^{'} &= \sum_{i=1}^{M} W_b (\boldsymbol{x}_i - \boldsymbol{\mu}_{\text{T}})(\boldsymbol{x}_i - \boldsymbol{\mu}_{\text{T}})^{\text{T}} \delta[b_{\boldsymbol{x}_i} - b] \\ &= \sum_{b=1}^{B} \sum_{i=1}^{M} W_b (\boldsymbol{x}_i - \boldsymbol{\mu}_{P,b} + \boldsymbol{\mu}_{P,b} - \boldsymbol{\mu}_{\text{T}})(\boldsymbol{x}_i - \boldsymbol{\mu}_{P,b} + \boldsymbol{\mu}_{P,b} - \boldsymbol{\mu}_{\text{T}})^{\text{T}} \delta[b_{\boldsymbol{x}_i} - b] \\ &= \sum_{b=1}^{B} \sum_{i=1}^{M} W_b (\boldsymbol{x}_i - \boldsymbol{\mu}_{P,b})(\boldsymbol{x}_i - \boldsymbol{\mu}_{P,b})^{\text{T}} \delta[b_{\boldsymbol{x}_i} - b] \\ &\quad + \sum_{b=1}^{B} \sum_{i=1}^{M} W_b (\boldsymbol{x}_i - \boldsymbol{\mu}_{P,b})(\boldsymbol{x}_i - \boldsymbol{\mu}_{I})^{T} \delta[b_{\boldsymbol{x}_i} - b] \\ &\quad + \sum_{b=1}^{B} \sum_{i=1}^{M} W_b (\boldsymbol{\mu}_{P,b} - \boldsymbol{\mu}_{\text{T}})(\boldsymbol{x}_i - \boldsymbol{\mu}_{P,b})^{\text{T}} \delta[b_{\boldsymbol{x}_i} - b] \\ &\quad + \sum_{b=1}^{B} \sum_{i=1}^{M} W_b (\boldsymbol{\mu}_{P,b} - \boldsymbol{\mu}_{\text{T}})(\boldsymbol{\mu}_{P,b} - \boldsymbol{\mu}_{\text{T}})^{\text{T}} \delta[b_{\boldsymbol{x}_i} - b] \end{aligned} \tag{25}$$

Because

$$\begin{aligned} &\sum_{b=1}^{B} \sum_{i=1}^{M} W_b (\boldsymbol{\mu}_{P,b} - \boldsymbol{\mu}_{\text{T}})(\boldsymbol{x}_i - \boldsymbol{\mu}_{P,b})^{\text{T}} \delta[b_{\boldsymbol{x}_i} - b] \\ &= \sum_{b=1}^{B} W_b (\boldsymbol{\mu}_{P,b} - \boldsymbol{\mu}_{\text{T}}) \sum_{i=1}^{M} (\boldsymbol{x}_i - \boldsymbol{\mu}_{P,b})^{\text{T}} \delta[b_{\boldsymbol{x}_i} - b] \\ &= \sum_{b=1}^{B} W_b (\boldsymbol{\mu}_{P,b} - \boldsymbol{\mu}_{\text{T}})(n_b \boldsymbol{\mu}_{P,b} - n_b \boldsymbol{\mu}_{P,b})^{\text{T}} = 0 \end{aligned}$$

and

$$\begin{aligned} &\sum_{b=1}^{B} \sum_{i=1}^{M} W_b (\boldsymbol{\mu}_{P,b} - \boldsymbol{\mu}_{\text{T}})(\boldsymbol{\mu}_{P,b} - \boldsymbol{\mu}_{\text{T}})^{\text{T}} \delta[b_{\boldsymbol{x}_i} - b] \\ &= \sum_{b=1}^{B} W_b (\boldsymbol{\mu}_{P,b} - \boldsymbol{\mu}_{\text{T}})(\boldsymbol{\mu}_{P,b} - \boldsymbol{\mu}_{\text{T}})^{\text{T}} \sum_{i=1}^{M} \delta[b_{\boldsymbol{x}_i} - b] \\ &= \sum_{b=1}^{B} W_b n_b (\boldsymbol{\mu}_{P,b} - \boldsymbol{\mu}_{\text{T}})(\boldsymbol{\mu}_{P,b} - \boldsymbol{\mu}_{\text{T}})^{\text{T}} \end{aligned}$$

we can obtain

$$\begin{aligned} \sum_{\text{T}}^{'} &= \sum_{b=1}^{B} \sum_{i=1}^{M} W_b (\boldsymbol{x}_i - \boldsymbol{\mu}_{P,b})(\boldsymbol{x}_i - \boldsymbol{\mu}_{P,b})^{\text{T}} \delta[b_{\boldsymbol{x}_i} - b] \\ &\quad + \sum_{b=1}^{B} W_b n_b (\boldsymbol{\mu}_{P,b} - \boldsymbol{\mu}_{\text{T}})(\boldsymbol{\mu}_{P,b} - \boldsymbol{\mu}_{\text{T}})^{\text{T}} \\ &= \sum_{W}^{'} + \sum_{B}^{'} \end{aligned} \tag{26}$$

Based on (26), we can compute $\sum'_{\text{T}}$ by the information of bins. It means that even the target does not contain the model completely. We can also obtain the approximated $\sum'_{\text{T}}$ from the bins in the target. Suppose the data dimension is limited to 2. Using principle component analysis method to solve the eigen-equation (Hastie et al., 2001), we have,

$$\sum_{\text{T}}^{'} \boldsymbol{v} = \lambda \boldsymbol{v} \tag{27}$$

The corresponding eigen-vectors, $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$, are the direction of long axis and short axis of the data distribution. λ_1 and λ_2 are the largest and smallest eigen-values. Further, suppose the data is uniformly distributed in an ellipse. The principle direction of the ellipse is $\boldsymbol{v}_1$ and the length of long axis and short axis is equal to two times of λ_1 and λ_2, respectively. So $\boldsymbol{v}_1$, $2\lambda_1$ and $2\lambda_2$ are the orientation and scales of the target.

4. Experiment results

The proposed spatial-color mean-shift tracking algorithms were implemented in C and tested on a 2.8GHz Pentium 4 PC with 1GB memory. We use normalized color variables r and g as the feature space as

$$r = \frac{\text{Red}}{(\text{Red} + \text{Green} + \text{Blue})}, \quad g = \frac{\text{Green}}{(\text{Red} + \text{Green} + \text{Blue})}$$

The color histograms are divided into 512 bins, i.e. the value B of (1) is equal to 512. Three video clips are used in the first experiment for fixed scale and orientation cases: the face sequence for face tracking; the cup sequence with complex appearance in complex background; and the walking girl sequence which is obtained from Adam et al. (2006) with partial occlusions. To demonstrate the scale and orientation tracking ability, two video clips are tested in the second experiment. The first sequence is the person walking away from and toward the camera with a large variation of scale. The second sequence which is obtained from the CAVIAR database (2004) illustrates the problem of large deformation. The image size of face sequence, cup sequence, walking girl sequence, and walking person sequence are 320×240, and the image size of the CAVIAR database is 352×288. The tracking window sizes of face sequence, cup sequence, walking girl sequence are 59×82, 50×65, and 27×98, respectively. In the following figures, tracker 1 means the algorithm of (19) and its extension while tracker 2 the algorithm of (20) and its extension. To compute the tracking error, the ground-truth of the object location is marked visually in every 10 frames and the error is determined by Euclidean distance.

Fig. 3. Face sequence of frames 33, 93, 117, 126, 183, 256, 271, and 455. (Red: tracker 1, blue: tracker 2, green: traditional mean-shift tracker). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

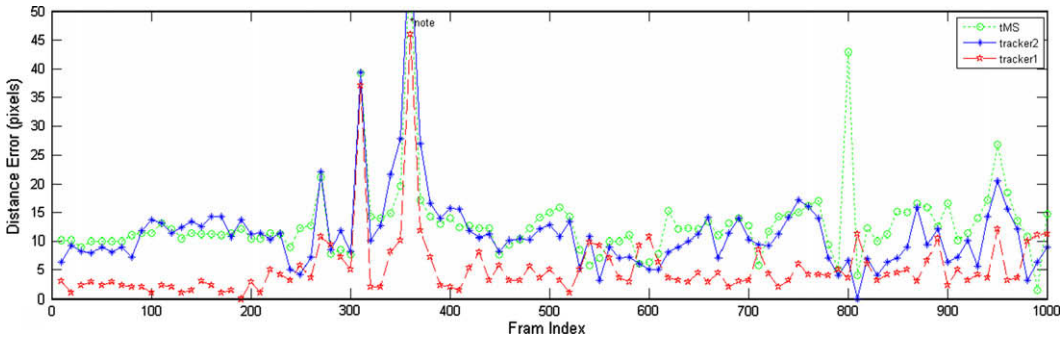

Fig. 4. Distance error of face sequence. (*Note*: we only consider the distance error which is smaller than 50 pixels).

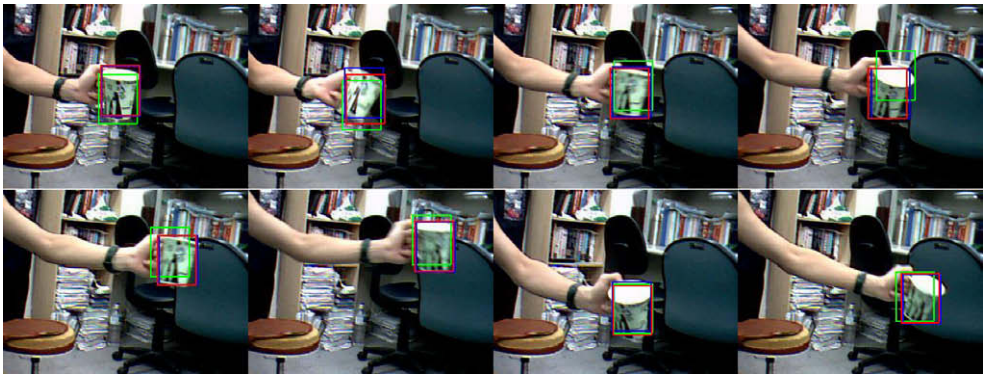

Fig. 5. Cup sequence of frames 4, 45, 63, 69, 81, 105, 166, and 243. (Red: tracker 1, blue: tracker 2, green: traditional mean-shift tracker). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.1. Spatial-color mean-shift trackers with single scale tracking

Part of the frames in the face tracking experiment is shown in Fig. 3. Fig. 4 shows the tracking error. For comparison purpose, traditional mean-shift algorithm is also implemented (marked as tMS in the figures). This example shows under simple background, all these methods have similar performance but on the average, the proposed methods outperform the traditional one.

Similarly, tracking of a cup with complex feature in a complex background are shown in Figs. 5 and 6. As shown in Fig. 6, traditional mean-shift algorithm has a larger tracking error and sometimes loses the target.

In the case of partial occlusion (Fig. 7), tracker 1 always captures the target under the circumstances of the variation of illumination and partial occlusion, but tracker 2 and traditional mean-shift fail in the tracking process.

4.2. Spatial-color mean-shift trackers with scale and orientation

Figs. 8 and 9 shows the results of the spatial-color mean-shift trackers with the PCA method. In the video clip, the target person

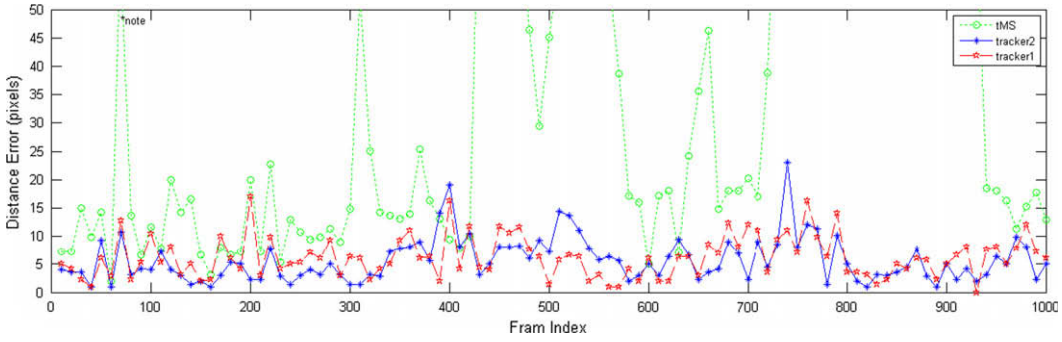

Fig. 6. Distance error of cup sequence. (*Note*: we only consider the distance error which is smaller than 50 pixels).

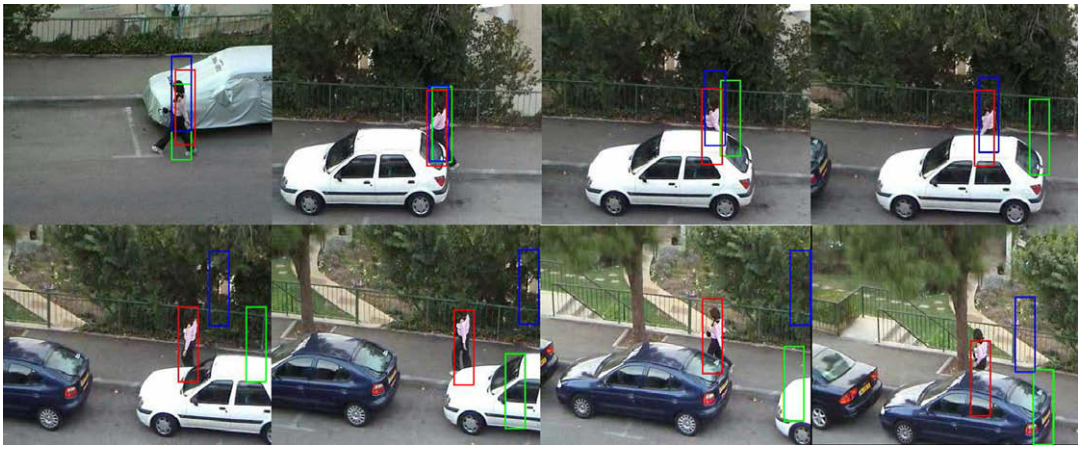

Fig. 7. Walking girl sequence of frames 28, 111, 124, 130, 153, 166, 196, and 220. (Red: tracker 1, blue: tracker 2, green: traditional mean-shift tracker). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 8. Walking person tracking results of spatial-color mean-shift tracker1 with PCA scale method (frames 83, 358, 494, 598, 689, 733, 914, and 1000).

Fig. 9. Walking person tracking results of spatial-color mean-shift tracker2 with PCA scale method (frames 83, 358, 494, 598, 689, 733, 914, and 1000).

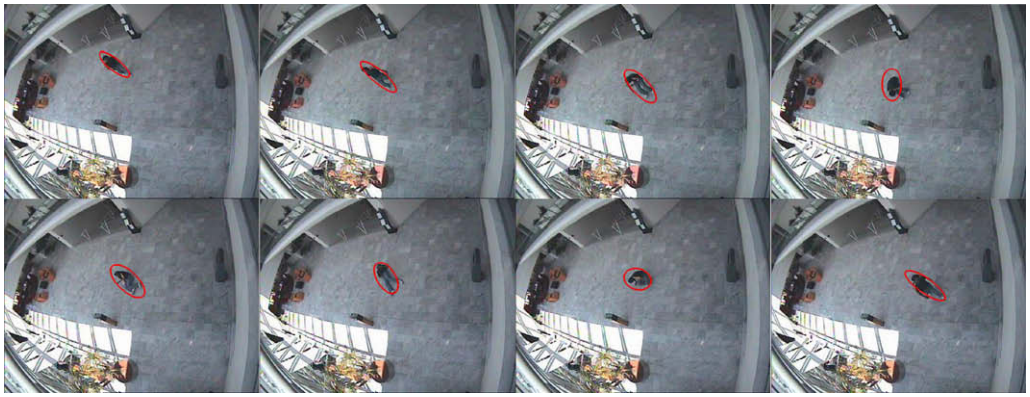

Fig. 10. Surveillance tracking results of spatial-color mean-shift tracker1 with PCA scale method (frames 3, 24, 32, 51, 59, 286, 318, and 353).

Fig. 11. Surveillance tracking results of spatial-color mean-shift tracker2 with PCA scale method (frames 3, 24, 32, 51, 59, 286, 318, and 353).

walks away from and toward the camera, the two trackers capture the target at all times. These show that both trackers are capable of tracking the size of the target. In the surveillance sequence obtained from the CAVIAR database (2004), a person walks, lies down, and finally stands up and resumes walking. These different actions give significant deformation of the target. Figs. 10 and 11 show that the trackers proposed in this paper always track the target with the corresponding scale, orientation, and shape.

4.3. Performance analysis

The performance of the proposed algorithms is analyzed in two different aspects: the preprocessing time of building the model and the computational time for each iterative step. The face sequence and cup sequence are used to test the performance of the proposed trackers. The models are built from the first image of these two sequences, and the preprocessing procedure is executed five times to

Table 1
The preprocessing time of both trackers (in second)

	Tracker 1	Tracker 2
Face sequence	0.027858	0.031299
Cup sequence	0.017306	0.022448

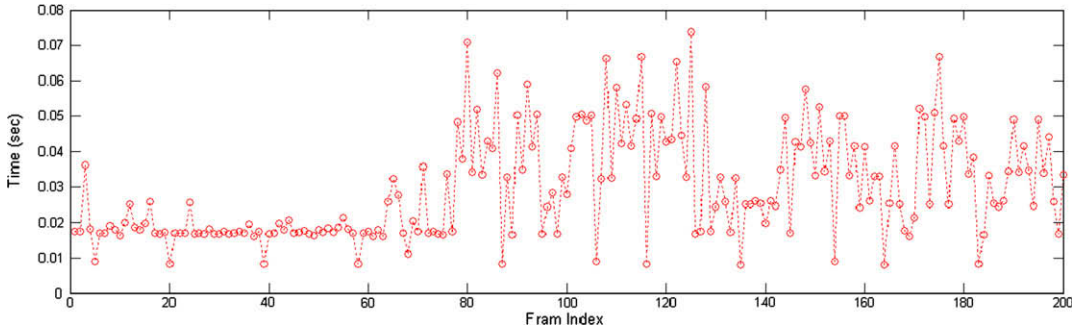

Fig. 12. The computing time of tracker 1 for the first 200 frames of face sequence.

obtain the average computing time. Table 1 shows the preprocessing time of both trackers.

Figs. 12 and 13 show the iteration time of the first 200 frames of face sequence and cup sequence for tracker 1. The average time of total frames (about 2300 frames) is 0.035855 s (about 28 frames/s). The average time of an iteration of total frames (about 1900 frames) of cup sequence is 0.017854 (about 56 frames/s). Figs. 14 and 15 show the results for tracker 2. The average time of total frames is 0.020670 s (about 48 frames/s). The average time of an iteration of total frames of cup sequence is 0.006608 (about

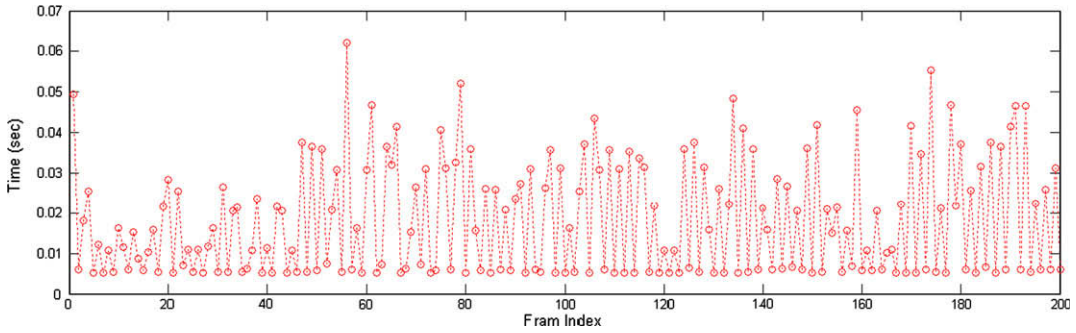

Fig. 13. The computing time of tracker 1 for the first 200 frames of cup sequence.

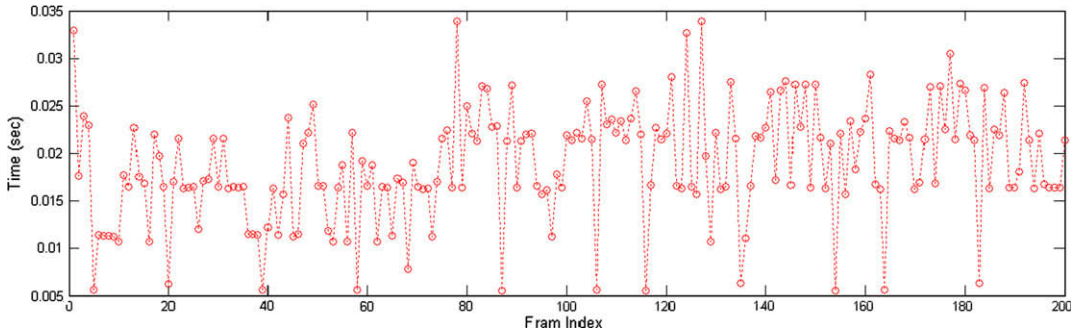

Fig. 14. The computing time of tracker 2 for the first 200 frames of face sequence.

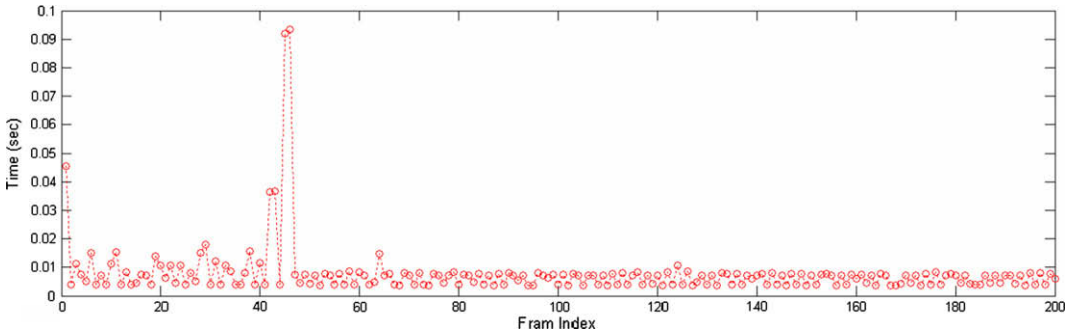

Fig. 15. The computing time of tracker 2 for the first 200 frames of face sequence.

151 frames/s). The tracking time of tracker 2 is smaller than that of tracker 1 because tracker 2 computes K_P and K_C at the preprocessing stage instead of at each iteration. Nevertheless, both trackers satisfy the real-time requirement of current image sampling rate for most cameras (30 frames/s).

5. Conclusion

A spatial-color mean-shift object tracking algorithm is proposed in this paper. Combining the spatial information with color feature makes the model more robust in tracking applications. New tracking algorithms are proposed based on the proposed similarity measure using the concept of the expectation of the estimated kernel density. Moreover, the principal component analysis is applied to the covariance matrix of the spatial-color distribution to access the scale and orientation of the target. The experiment results show the effectiveness and real-time capability of the proposed algorithms. The update of scale and orientation in this paper are based on the image tracked by the proposed algorithm. This information is not considered for the tracking in the next step, which is an interesting research topic for further study. To do so, a modified model like the affine transformation in Zhang et al. (2005), might have to be considered. However, this means certain scale and orientation restrictions are imposed on the target image. These aspects will be investigated in our future research.

Acknowledgements

This work was supported in part by the National Science Council of Taiwan, ROC under Grant No. NSC 95-2218-E-009-064 and the Ministry of Economic Affairs under Grant No. 95-EC-17-A-04-S1-054.

References

Adam, Amit, Rivlin, Ehud and Shimshoni, Ilan, 2006. Robust fragments-based tracking using the integral histogram. In: Proc. 2006 IEEE Comput. Soc. Conf. on Computer Vision and Pattern.

Birchfield, S., Rangarajan, S., 2005. Spatiograms versus histograms for region-based tracking. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'05) 2, pp. 1158–1163.

Birchfield, S., Rangarajan, S., 2007. Spatial histograms for region-based tracking. ETRI J. 29 (5), 697–699.

Bradski, G.R., 1998. Computer vision face tracking for use in a perceptual user interface. Intel. Technol. J. 2 (2), 1–15.

CAVIAR database, 2004. <http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/>.

Collins, R., 2003. Mean-shift blob tracking through scale space. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'03) 2, 2003, p. 234.

Comaniciu, D., Ramesh, V., Meer, P., 2003. Kernel-based object tracking. IEEE Trans. Pattern Anal. Machine Intell. 25 (5), 564–577.

Hastie, T., Tibshirani, R., Friedman, J., 2001. The Elements of Statistical Learning. first ed. Springer.

Lindeberg, T., 1998. Feature detection with automatic scale selection. Internat. J. Comput. Vision 30 (2), 79–116.

O'Conaire, C., O'Connor, N.E., Smeaton, A.F., 2007. An improved spatiogram similarity measure for robust object localization. In: Proc. ICASSP, 2007.

Parameswaran, V., Ramesh, V., Zoghlami, I., 2006. Tunable kernels for tracking. In: Proc. 2006 IEEE Comput. Soc. Conf. on Computer Vision and Pattern Recognition 2, pp. 2179–2186.

Porikli, F., Tuzel, O., 2005. Object tracking in low-frame-rate video. In: Proc. SPIE, vol. SPIE-5685, 2005, pp. 72–79.

Yang, C., Duraiswami, R., Davis, L., 2005. Efficient mean-shift tracking via a new similarity measure. IEEE Conf. Comput. Vision and Pattern Recognition 1, 176–183.

Zhang, H., Huang, Z., Huang, W., Li, L., 2004. Kernel-based method for tracking objects with rotation and translation. In: Proc. 17th Internat. Conf. on Pattern Recognition, (ICPR'04), vol. 2, pp. 728–731.

Zhang, H., Huang, W., Huang, Z., Li, L., 2005. Affine object tracking with Kernel-based spatial-color representation. In: IEEE Comput. Soc. Conf. on Comput. Vision and Pattern Recognition (CVPR'05) 1, 20–25 June, pp. 293–300.

Zhao, Q., Tao, H., 2005. Object tracking using color correlogram. In: Second Joint IEEE Internat. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 15–16 October, pp. 263–270.

Zivkovic, Z., Krose, B., 2004. An EM-like algorithm for color-histogram-based object tracking. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'04) 1, pp. 798–803.