Contents lists available at ScienceDirect

Computer Vision and Image Understanding

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

# A motion observable representation using color correlogram and its applications to tracking

Qi Zhao *, Hai Tao

*Department of Computer Engineering, University of California at Santa Cruz, 1156 High Street, Santa Cruz, CA 95064, USA*

ARTICLE INFO

*Article history:*
Received 21 March 2006
Accepted 18 October 2008
Available online 14 November 2008

*Keywords:*
Simplified color correlogram (SCC)
Kernel based tracking
Optimal feature selection

ABSTRACT

This paper presents a special form of color correlogram as representation for object tracking and carries out a motion observability analysis to obtain the optimal correlogram in a kernel based tracking framework. Compared with the color histogram, where the position information of each pixel is ignored, a simplified color correlogram (SCC) representation encodes the spatial information explicitly and enables an estimation algorithm to recover the object orientation. In this paper, based on the SCC representation, the mean shift algorithm is developed in a translation–rotation joint domain to track the positions and orientations of objects. The ability of the SCC in detecting and estimating object motion is analyzed and a principled way to obtain the optimal SCC as object representation is proposed to ensure reliable tracking. Extensive experimental results demonstrate SCC as a viable object representation for tracking.

## 1. Introduction

As a challenging task, real-time tracking in complex environment requires both a discriminative object representation and an effective inference framework for object localization [20,26,33], with the former component being the foundation for the latter one. This paper focuses on the object representation and the related object localization problems.

Object representations model the characteristics of the object being tracked, which is the combined outcome of the object's shape, pose, motion, reflectance properties and illumination conditions. Among existing representations, some embody the details of the object appearance [14,16,17,22,31,28], which explicitly describe the spatial arrangement of the pixels in the object window. These representations prove effective when the object is rigid or the appearance and shape changes are insignificant or modelled in the representation. On the other hand, a simple yet robust object representation is the color histogram [4,10,13,29,32]. Swain and Ballard [29] employed color histogram as a global visual feature for object representation, demonstrating that color can be exploited as a useful feature for rapid detection. Funt [13] used ratios of colors from neighboring locations, so as to extend Swain's method to be insensitive to illumination changes. More recent methods such as the mean shift based tracking algorithm [9] and the CAMShift algorithm [4] were proposed using this representation for object tracking. Although color histograms discard all spatial information and algorithms solely depending on them often fail to estimate changes in orientation and shape of the objects, the use of color histograms for tracking is considered a reasonable choice for many applications due to their simplicity, efficiency and robustness.

To improve the traditional histogram based tracking, the deformation and scale compensations have been addressed. In the last decade, Shi and Tomasi [28] were aware of the inadequacy of pure translation as motion model and they suggested to estimate linear warping as well as translation for tracking. This early work applied template based representation where details are encoded. In [7], the scale selection problem was approached by adapting Lindeberg's theory of feature scale selection based on local maxima of differential scale-space filters. However, there are only a few published papers discussing the problem of estimating object orientation using color distributions. Recently, Birchfield and Sriram [3] proposed spatiograms to incorporate spatial information into histogram. Spatiogram is a single histogram in which each bin is spatially weighted by the mean and covariance of the locations of the pixels that contribute to that bin. Due to the encoding of spatial information, this representation is rotationally variant therefore able to acquire orientation information. However, the focus in [3] is to achieve more robust performance, instead of estimating orientation information. Fan and Wu [11] developed the method of multiple collaborative kernel tracking to cope with the singularity problem of kernel based tracking. With multiple kernels, object orientation can be computed by estimating the relative positions of the kernels.

To our knowledge, the work discussed in this paper presents the first effort to estimate orientation using a single kernel. The goal is to extend the color histogram approach to provide spatial information while preserving its efficiency and robustness. The new

* Corresponding author.
*E-mail addresses:* zhaoqi@soe.ucsc.edu (Q. Zhao), tao@soe.ucsc.edu (H. Tao).

1077-3142/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2008.10.007

representation enables robust estimation of both object orientation and position, as illustrated in Fig. 1.

### 1.1. Introduction to color histograms with spatial information

Adding spatial information into a color histogram improves its representation power, which is essential for estimating object motion and discriminating objects from the background and from each other. One way of obtaining a richer representation is to generalize the color histogram to a geometric histogram [27,24,25], and a geometric histogram is the histogram of color subsets arranged according to predefined geometric configurations. A geometric configuration is defined as an arrangement of finite pixels on the plane of an image. It may be as simple as a single pixel, which reduces to the conventional color histogram; or as complex as all pixels of the image. Among various geometric histograms, the color correlogram is commonly used, where the geometric configuration is a line segment. Huang [19] proposed to use the color correlogram as a new color feature for image indexing/retrieval and it proved to be an effective tool for describing image content in image indexing and retrieval [18,19]. Later, autocorrelogram [23], which considers the correlation between identical colors, has been suggested in the same field for efficiency concerns. The ideas of geometric histogram, correlogram and autocorrelogram are made more precise in Section 2.

### 1.2. Introduction to kernel based tracking methods

Mean shift procedure was first proposed in 1975 by Fukunaga and Hostetler [12] and revived by Cheng [6] in 1995. More recently, Comaniciu et al. [9,10] successfully applied this technique to object tracking. The mean shift algorithm is an example of computer vision techniques which are based on *in situ* optimization, where each data point is associated with a voting kernel to produce a more dense structure from where the feature can be reliably extracted [8,30]. The advantages of the mean shift based tracking include (1) efficiency, which is critical for real-time systems; (2) modularity, as the localization scheme can be integrated with various motion filters and data association techniques; and (3) easy implementation. Moreover, the adaptive magnitude of the mean shift vector eliminates the need for additional procedures to choose the adequate step size. To make this paper more self contained, we provide a rather detailed description of this specific kernel based tracking framework. However, the proposed representation can be combined with other computational framework to perform tracking.

The main idea behind mean shift tracking, or kernel based tracking methods is combining the statistical features with a stochastic gradient descend method for optimization [10]. By convolving the features with an isotropic kernel, a spatially smooth similarity function can be defined and the object localization problem is then reduced to the optimization of this function. The smoothness of the similarity function allows the application of gradient descent methods, such as the mean shift algorithm, to find the location of an object very efficiently.

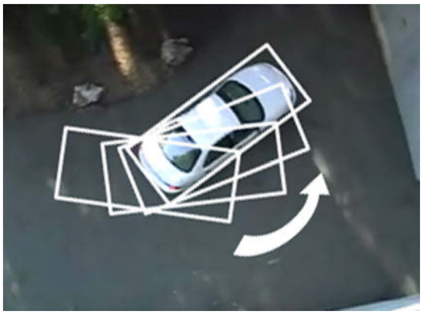

**Fig. 1.** SCC based tracker. Orientation of an object is estimated simultaneously with its position.

Most existing kernel based tracking methods are concerned only with the tracking of object locations [9,10], or object locations and scales [7]. The wide use of isotropic kernels is largely due to its efficiency and the effective incorporation with existing gradient descent techniques. However, isotropic kernels result rotationally-invariant statistical features [9–11,15] therefore rotational motion cannot be estimated using these methods. In the method proposed in this paper, although the kernel used is also rotationally-symmetric, the underlying SCC representation is sensitive to orientation changes. This property makes the representation sensitive to orientation changes, therefore capable of tracking rotational motion as well as translational motion. As in most kernel based algorithms, the assumption is that the statistics of the SCC feature should be sufficient to determine the motion of the object [15]. However, this assumption needs to be validated. This paper shows that under certain degenerated cases, translational/rotational motion may not cause changes in the SCC, then the motion is not observable.

### 1.3. Our approach

The main contributions of the research presented in this paper are:

1. We propose a simplified form of color correlogram (SCC) for tracking, which is demonstrated to be both motion observable and computationally efficient.
2. We explore the spatial information in the proposed SCC representation to estimate object orientation. As a result, restriction on object shapes in the traditional color histogram based methods is removed.
3. We extend the mean shift method to the joint translation–rotation space to integrate position and orientation estimation in a single framework. The SCC can be easily generalized and incorporated into tracking framework other than the mean shift based method.

The paper is organized as follows. In Section 2, the color geometric histogram, color correlogram and color autocorrelogram are introduced, and a special form of color correlogram is proposed for tracking purpose. In Section 3, the new representation is incorporated into the extended translation–rotation concatenated mean shift framework. In Section 4, properties of the SCC and kernel based tracking method are further investigated, and the criterion to obtain the maximal observability is proposed. In Section 5, details of implementation are discussed. In Section 6, the proposed method is applied to several typical tracking scenarios and promising experimental results are demonstrated. Section 7 concludes the paper with the discussion of the strengths and the weaknesses of the proposed method.

## 2. Object representation

### 2.1. Introduction to color geometric histograms

The main idea of color geometric histogram is to exploit the spatial relationship between colors, which enhances the discrimination ability of color histogram. Specifically, denote the quantization levels of the color space as $m$; represent pixel positions as $p$ and the intensity of the pixel at position $p$ as $I(p)$. For an ordered set of pixels $P$, its geometric configuration is represented as $\mathsf{F}(P)$. Given a geometric configuration $f_G$, the ordered set of pixels in the image that associates with $f_G$ is denoted as $P_G$ and the corresponding set of pixel colors as $I_G$. Note that the dimension of the geometric histogram $G(f_G)$ is $\mid P_G \mid$. For a particular color set $\{u_1, u_2, \ldots, u_{|P_G|}\}$, the probability of concurrence is

$$G(f_G)_{u_1,u_2,\ldots,u_{|P_G|}} = Pr(I(p_1) = u_1 \wedge \cdots \wedge I(p_{|P_G|}) = u_{|P_G|} | \mathsf{F}(P) = f_G). \quad (1)$$

The color geometric histogram corresponding to the geometric configuration $f_G$ is a table indexed by color sets $\{u_1, u_2, \ldots, u_{|P_G|}\}$.

#### 2.1.1. Color correlogram

Color correlogram [19] is a special case of color geometric histogram, where the geometric configuration is line segments. Formally, quantize the distance space into $Z$ levels, let a distance $z \in \{1, 2, \ldots, Z\}$ be fixed as a priori, the probability of concurrence of the gray levels $u$ and $v$ for two pixels with distance $d \in \{1, 2, \ldots, z\}$ is

$$C^d_{u,v} = Pr(I(p_1) = u \wedge I(p_2) = v | |p_1 - p_2| = d). \quad (2)$$

The color correlogram of an image is a table indexed by color pairs and distance, where the $d$-th entry for $(u, v)$ specifies the probability of finding a pixel of color $u$ at a distance $d$ from a pixel of color $v$. Note that the size of the correlogram is $O(m^2 z)$.

#### 2.1.2. Color autocorrelogram

For efficiency, Ojala et al. [23] proposed the autocorrelogram that only considers the correlation between identical colors. Formally,

$$A^d_u = C^d_{u,u}. \quad (3)$$

Therefore, instead of $O(m^2 z)$ in the color correlogram, the size of color autocorrelogram is $O(mz)$.

### 2.2. The simplified color correlogram (SCC)

As mentioned in Section 1.1, color correlogram is a commonly used feature for image indexing/retrival. Compared with image indexing/retrieval that usually deals with various images in a large data set where content ambiguity frequently occurs, in the tracking context, normally a small number of objects are dealt with, which implies that the correlogram can be simplified to save computational resources. In addition, the goal to track orientation imposes requirements on the representation that it should be sensitive to orientation changes of objects. Therefore, instead of considering pixel pairs along all directions and with all distances, we propose a simplified representation counting pixel pairs along only one or several selected directions with predefined distances.

Let us denote the directions along which pixel pairs are counted as *axes*, where the axes are defined in the object coordinate system. For two pixels $p_1$ and $p_2$ in an image, represent $f$ as a function to obtain the direction and the distance of the pixel pair, the probability of concurrence of the intensity levels $u$ and $v$ for two pixels with distance $d$ and direction $\theta$ can be expressed as

$$S(d, \theta)_{u,v} = Pr(I(p_1) = u \wedge I(p_2) = v | f(p_1 - p_2) = (\theta, d)). \quad (4)$$

Recall that the color space is quantized into $m$ levels, the number of color pairs is $M_0 = m \times m$. The SCC with a single axis of pair distance $d$ and direction $\theta$ is represented as a vector of size $M_0$ as

$$\mathbf{S}(d, \theta) = [S(d, \theta)_{1,1} \cdots S(d, \theta)_{1,m} \cdots S(d, \theta)_{m,1} \cdots S(d, \theta)_{m,m}]^T \in \mathbb{R}^{M_0}. \quad (5)$$

In Eq. (4), the $L_2$ norm is applied to measure the distance between pixels, i.e., for pixels $p_1 = (x_1, y_1), p_2 = (x_2, y_2)$, we define $\mid p_1 - p_2 \mid = ((x_1 - x_2)^2 + (y_1 - y_2)^2)^{1/2}$. For multiple axes SCCs, multiple $\mathbf{S}(d, \theta)$ can be defined, where for each one $\theta \in \{\theta^l, l = 1, \ldots, L\}$ and $d \in \{d^l, l = 1, \ldots, L\}$ are predefined. Here, $L$ is the number of axes, $\theta^l$ is the direction of axis $l$ and $d^l$ the pair distance along axis $l$.

Fig. 2 illustrates the simplification from the conventional correlograms to our proposed SCCs: firstly, we fix the distance between pixel pairs (Fig. 2a-3 and b-3). In other words, select one correlogram from the correlogram set, where a number of distances are considered. Secondly, we fix the directions of the pixel pairs (Fig. 2a-4 and b-4). The two moons in the figure, one viewed at the first half of a month while the other at the second half, have not only similar histograms, but similar correlograms, as both the histogram and correlogram are rotationally-invariant. However, the structural difference is exhibited in the single-axis SCCs. It is also interesting to see that by reversing the direction of the pixel pair for the second moon, its SCC (Fig. 2b-5) is similar to the one of the first moon (Fig. 2a-4), as the second moon can be seen approximately as the first one rotating 180°. The observations provide evidence that the same object with different orientations are distinguishable by the SCC. In the context of tracking, this means that the representation provides the capability of capturing orientation variance.

One possible problem is that the SCCs with arbitrarily chosen axes and pair distances may fail in motion recovery due to singularity, as will be discussed in detail in the following sections. One way to ensure more reliable tracking is to select more than one axis to form a multi-axis SCC, therefore motion ignored by pairs along one axis can be recovered along other axes. This strategy suffices in most cases. However, efficiency consideration suggests a more sophisticated alternative, which is to directly obtain one optimal axis and its corresponding pair distance, so that the resulting SCC is the most sensitive to all different motions. For example, the most optimal axis for the moon (Fig. 2a-1) is along 120° or 60°. In Section 4, we would introduce in detail the criteria and algorithm to obtain the optimal axis.

In Fig. 3, a typical example of the SCC is shown, where two orthogonal directions are selected as axes. It should be noted that axes are defined in the object coordinate system, and rotate when the object rotates. Pixel pairs along both axes with predefined distances are counted for the SCC computation.

The advantages of the simplified color correlogram (SCC) can be summarized as follows:

1. The SCC makes use of the spatial correlation between color pairs to achieve a natural integration of both color and spatial information, as the original color correlogram. This richer representation has larger discriminative power compared with color histogram, therefore it is not susceptible to false positive matches.
2. It is computationally inexpensive to obtain this representation. Since the pair distance is predetermined for each direction, the size is reduced from $O(m^2 z)$ for the conventional color correlogram to $O(m^2)$ for the SCC. Moreover, the number of the representation elements is significantly reduced due to the fixed directions.
3. The SCC is effective in manifesting rotational variations. Unlike the conventional color correlogram Eq. (2), which treats each direction equally, the SCC emphasized certain directions therefore is sensitive to orientation changes.

### 2.3. Representing objects using color correlogram

In this work, we propose to represent the pixel pairs in a joint position-orientation domain with distance between two pixels fixed. Each pair of pixels can be parameterized using a 3-dimen-

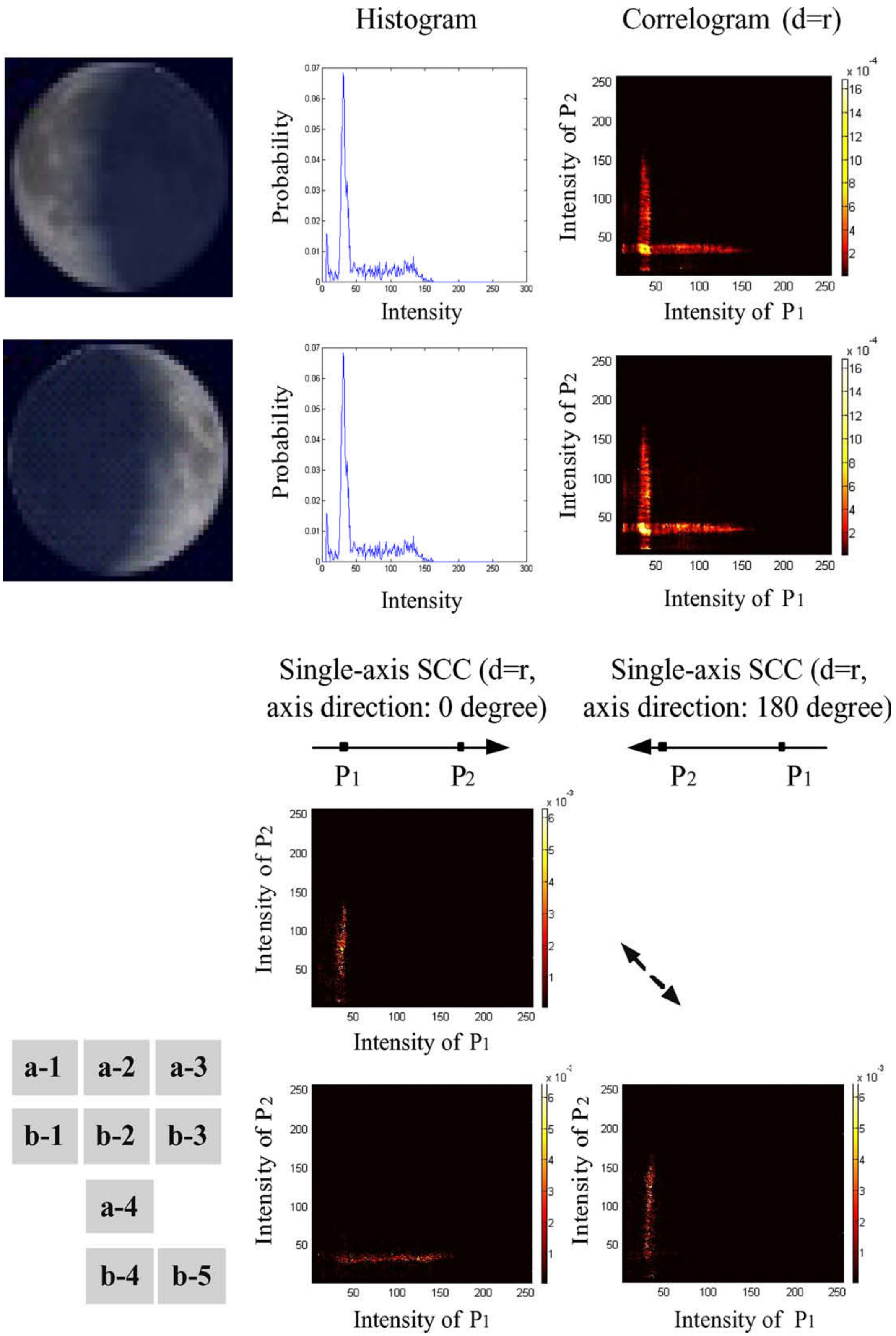

**Fig. 2.** Simplified correlogram: (a-1, b-1) objects: the *moons*, (a-2, b-2) histograms of the gray-scale images of the moons shown in (a-1) and (b-1), (a-3, b-3) correlograms of the gray-scale images of the moons, with pair distance being the same as the radius of the moons ($P_1$ and $P_2$ are two pixels in a pixel pair), (a-4, b-4) single-axis SCCs of the gray-scale images of the moons, with pair distance being the same as the radius of the moons and pair direction of 0°, (b-5) single-axis SCC of the gray-scale image of the moon shown in (b-1), with pair distance being the same as the radius of the moon and pair direction of 180°.

sional vector $\mathbf{\Phi} = [x^c, y^c, \theta]^T$, where $(x^c, y^c)$ are the image coordinates of the midpoint of each pair, as illustrated in Fig. 3, and $\theta$ is the angle between the major axis of the object and the object coordinate system.

### 2.3.1. Target model

Consider for a moment a target model for the SCC with one axis $l$, the index $l$ is omitted to keep the notation simple. For each chosen axis $l$ of direction $\theta^l$, the corresponding pair distance $d^l$ is pre-

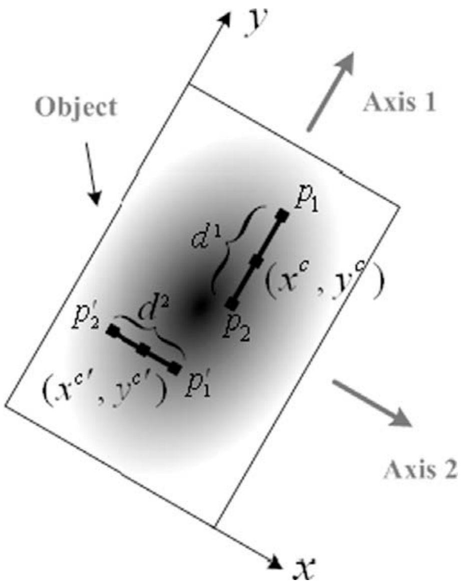

**Fig. 3.** Illustration of the two-orthogonal-axis SCC.

determined. The orientation space is quantized with equal intervals within a certain range centered at $\theta^l$.

Using the SCC as object representation for tracking, pixel pairs of a predefined distance and along certain directions are used to represent the target model. Formally, we denote $\mathbf{\Phi}_{ijr} = [x_i^c, y_j^c, \theta_r]^T$ to represent a pixel pair $(\mathbf{P}^0_{ijr}, \mathbf{P}^1_{ijr})$, where $i, j$ are indexes for the pair locations and $k$ is the index for the discrete directions after quantization in the orientation space. Specifically,

$$\begin{cases} \mathbf{P}^0_{ijr} = (x_i^c, y_j^c) - 1/2(d\cos\theta_r, d\sin\theta_r) \\ \mathbf{P}^1_{ijr} = (x_i^c, y_j^c) + 1/2(d\cos\theta_r, d\sin\theta_r) \end{cases}, \tag{6}$$

Following the definition of color correlogram Eq. (2), for each color pair $(u, v)$ in the target model, its probability of concurrence is

$$M_{u,v} = \alpha \sum_{i,j,r} \delta[I(\mathbf{P}^0_{ijr}) - u]\delta[I(\mathbf{P}^1_{ijr}) - v] K\left(\frac{\mathbf{\Phi}_{ijr}}{\mathbf{h}}\right), \tag{7}$$

where $K()$ is a kernel function that assigns a smaller weight to the locations and orientations that are farther from the center of the object. For example, the widely used Epanechnikov kernel is defined as $K_E(\mathbf{x}) = \begin{cases} \lambda(1 - \|\mathbf{x}\|^2), & \text{if} \|\mathbf{x}\| < 1 \\ 0, & \text{otherwise} \end{cases}$. In Eq. (7), $\mathbf{\Phi}_{ijr}$ is in the 3-dimensional location–orientation concatenated domain, and $\mathbf{h}$ is a bandwidth in the same domain. The division of $\mathbf{\Phi}_{ijr}$ by $\mathbf{h}$ normalizes each element of $\mathbf{\Phi}_{ijr}$ to the range of $[0, 1]$. The summations are performed over a local window of pixel pairs $\mathbf{\Phi}_{ijr}$ centered at $\mathbf{0}$ (assuming the location and orientation of the target model are 0s). $\delta$ is the Kronecker delta function and the normalization factor is given by $\alpha = 1/\sum_{i,j,r} K(\frac{\mathbf{\Phi}_{ijr}}{\mathbf{h}})$.

Similar as Eq. (5), the target model is then defined as a vector of size $M_0 = m \times m$, where each term in $\mathbf{M}$ is calculated using Eq. (7), as

$$\mathbf{M} = [M_{1,1} \cdots M_{1,m} \cdots M_{m,1} \cdots M_{m,m}]^T \in \mathbb{R}^{M_0} \tag{8}$$

In the example shown in Fig. 4, three directions including the axis itself ($\theta_1 = \theta^l - 15^\circ, \theta_2 = \theta^l, \theta_3 = \theta^l + 15^\circ$) are used to obtain voting pairs.

To better understand the idea of the extended translation–rotation joint space based mean shift algorithm, it should be emphasized that for each selected axis, to apply mean shift procedure for orientation, we need to fit an isotropic kernel over the orientation dimension and obtain new orientation as a search in the basin of attraction, i.e., a region covering the target model pixel pairs, of a defined similarity function. Therefore the SCC representation in this algorithm needs to include pixel pairs along several directions around the axis, where each pair votes for a new orientation.

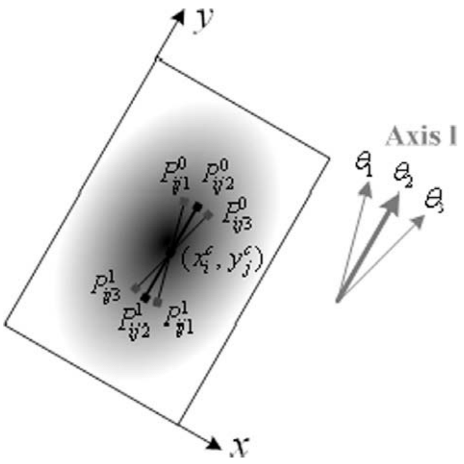

**Fig. 4.** Illustration of SCC representation for extended mean shift iterations.

By definition, $L$ different axes yield $L$ different target models $\mathbf{M}_1, \ldots, \mathbf{M}_L$.

To formulate the target model in a more compact form [15], represent $I(\mathbf{P}^0_{ijr})$ and $I(\mathbf{P}^1_{ijr})$ as the colors of the pixel pair $\mathbf{\Phi}_{ijr}$ in the image $I$, let $\delta[I(\mathbf{\Phi}_{ijr}) - C_{uv}] = \delta[I(\mathbf{P}^0_{ijr}) - u]\delta[I(\mathbf{P}^1_{ijr}) - v]$, denote $W$, $H$ and $\Theta$ as the numbers of $x^c$s, $y^c$s and $\theta$s considered in the SCC, which results the total number of pixel pairs in the SCC to be $N_0 = W \times H \times \Theta$, we define

$$\mathbf{U}_M = \begin{pmatrix} \delta[I(\mathbf{\Phi}_{111}) - C_{11}] & \cdots & \delta[I(\mathbf{\Phi}_{111}) - C_{1m}] & \cdots & \delta[I(\mathbf{\Phi}_{111}) - C_{mm}] \\ \delta[I(\mathbf{\Phi}_{112}) - C_{11}] & \cdots & \delta[I(\mathbf{\Phi}_{112}) - C_{1m}] & \cdots & \delta[I(\mathbf{\Phi}_{112}) - C_{mm}] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \delta[I(\mathbf{\Phi}_{WH\Theta}) - C_{11}] & \cdots & \delta[I(\mathbf{\Phi}_{WH\Theta}) - C_{1m}] & \cdots & \delta[I(\mathbf{\Phi}_{WH\Theta}) - C_{mm}] \end{pmatrix} \in \mathbb{R}^{N_0 \times M_0}. \tag{9}$$

In Eq. (9), each row associates a pixel pair, and the element in the row corresponding to its pair colors is assigned 1 while others in the same row are assigned 0.

Further, we stack the kernel functions for each pixel pair into a column vector as

$$\mathbf{K}(\mathbf{0}) = \left[K\left(\frac{\mathbf{\Phi}_{111}}{\mathbf{h}}\right)K\left(\frac{\mathbf{\Phi}_{112}}{\mathbf{h}}\right) \cdots K\left(\frac{\mathbf{\Phi}_{WH\Theta}}{\mathbf{h}}\right)\right]^T \in \mathbb{R}^{N_0}. \tag{10}$$

Then the compact form of the target model Eq. (8) can be written as

$$\mathbf{M} = \alpha \mathbf{U}_M^T \mathbf{K}(\mathbf{0}) \in \mathbb{R}^{M_0}. \tag{11}$$

### 2.3.2. Target candidate

As in Section 2.3.1, we consider for a moment a target candidate for the SCC with one axis $l$ and omit the index $l$. Define $\mathbf{\Phi}_0$ as the initialized location and orientation in the current frame. Similar to the target model, the target candidate is defined as

$$\mathbf{C}(\mathbf{\Phi}_0) = [C_{1,1}(\mathbf{\Phi}_0) \cdots C_{1,m}(\mathbf{\Phi}_0) \cdots C_{m,1}(\mathbf{\Phi}_0) \cdots C_{m,m}(\mathbf{\Phi}_0)]^T \in \mathbb{R}^{M_0}, \tag{12}$$

where each term in $\mathbf{C}(\mathbf{\Phi}_0)$ is

$$C_{u,v}(\mathbf{\Phi}_0) = \beta \sum_{i,j,r} \delta\left[I(\mathbf{P}^0_{ijr}) - u\right]\delta\left[I\left(\mathbf{P}^1_{ijr}\right) - v\right] K\left(\frac{\mathbf{\Phi}_{ijr} - \mathbf{\Phi}_0}{\mathbf{h}}\right), \tag{13}$$

with $\begin{cases} \mathbf{P}^0_{ijr} = (x^c_i, y^c_j) - 1/2(d\cos\theta_r, d\sin\theta_r) \\ \mathbf{P}^1_{ijr} = (x^c_i, y^c_j) + 1/2(d\cos\theta_r, d\sin\theta_r) \end{cases}.$

In Eq. (13), the normalization factor is given as $\beta = 1/\sum_{i,j,r} K(\frac{\mathbf{\Phi}_{ijr}-\mathbf{\Phi}_0}{\mathbf{h}})$. Note that $\beta$ is independent of $\mathbf{\Phi}_0$. Both $\alpha$ and $\beta$ can be precalculated for a given kernel and different values of $\mathbf{h}$, $\theta$ and $d$.

Similarly, its matrix form is

$$\mathbf{C}(\mathbf{\Phi}_0) = \beta \mathbf{U}_C^T \mathbf{K}(\mathbf{\Phi}_0) \in \mathbb{R}^{M_0}, \tag{14}$$

where $\mathbf{U}_C$ is defined the same way as $\mathbf{U}_M$ Eq. (9) and

$$\mathbf{K}(\mathbf{\Phi}_0) = \left[ K\left(\frac{\mathbf{\Phi}_{111}-\mathbf{\Phi}_0}{\mathbf{h}}\right) K\left(\frac{\mathbf{\Phi}_{112}-\mathbf{\Phi}_0}{\mathbf{h}}\right) \cdots K\left(\frac{\mathbf{\Phi}_{WHK}-\mathbf{\Phi}_0}{\mathbf{h}}\right) \right]^T \in \mathbb{R}^{N_0}. \tag{15}$$

$L$ target candidates $\mathbf{C}_1, \ldots, \mathbf{C}_L$ can be defined for $L$ axes.

## 3. Simplified color correlogram (SCC) based tracking

### 3.1. Overview

One important aspect of the work is to capture both the translational and rotational movements of the objects. The most straightforward way of accomplishing this is the exhaustive search. However, it is computationally expensive to search in a joint position-orientation domain. To make the histogram matching problem more efficient, various gradient descent algorithms can be applied. For example, one possible approach is to first adopt the traditional mean shift based tracking algorithm to locate the object position and then apply a search step in the orientation space to estimate the orientation information. In this paper, we present a novel method that simultaneously estimates orientation and position in a mean shift framework.

### 3.2. Introduction to the mean shift based tracking algorithm and its limitations

The mean shift algorithm is a nonparametric statistical method for seeking the nearest mode of a point sample distribution. The algorithm has recently been adopted as an efficient technique for blob tracking [4,7,9] and feature space analysis [6,8]. The key component of the mean shift algorithm is the computation of an offset value from a location $\mathbf{y}$ to a new location according to the mean shift vector [9]

$$\Delta\mathbf{y} = \frac{\sum_{i=1}^n \mathbf{x}_i w_i g(\frac{\mathbf{y}-\mathbf{x}_i}{\mathbf{h}})}{\sum_{i=1}^n w_i g(\frac{\mathbf{y}-\mathbf{x}_i}{\mathbf{h}})} - \mathbf{y}, \tag{16}$$

where $g(\mathbf{x}) = -k'(\mathbf{x})$, and $k$ is the profile of a kernel $K$, which is a function of $[0,\infty) \to R$ such that $K(\mathbf{x}) = k(\|\mathbf{x}\|^2)$. The summations are performed over a local window around location $\mathbf{y}$, with $\mathbf{x}_i$ representing the pixels and $\mathbf{h}$ the window radius. $n$ describes sample size and $w_i$ is the sample weight defined as

$$w_i = \sum_{u=1}^m \delta[I(\mathbf{x}_i) - u] \sqrt{H_u^M / H_u^C}. \tag{17}$$

In Eq. (17), $H_u^M$ and $H_u^C$ represent the probability of color $u$ in the model representation and that in the representation of the window of the current frame, respectively. The sample weight at a pixel with color $u$ being proportional to $\sqrt{H_u^M/H_u^C}$ implies that in the current window, if a pixel is of the color which has larger probability compared with that in the target model, the pixel is de-emphasized by being assigned a smaller weight ($w < 1$). The intuitive understanding is that the new window center is pushed away from this pixel. Conversely, a pixel with the color which accounts for a smaller proportion of colors than the target representation, is emphasized by given a larger weight ($w > 1$) and therefore the new center is pulled toward it. The traditional mean shift based tracking algorithm is essentially a voting algorithm to decide a new center location with each pixel within the window giving its own vote.

The above intuition is extended in this work. Besides allowing all the elements in the window to vote for the location of the new center, we also allow each element of the representation to vote for the orientation of the object in a new frame. This is made possible by using the SCC representation proposed in Section 2, where each voting pixel pair contains orientation information. The details of the extended mean shift based algorithm using SCC are given in the next subsection.

### 3.3. Single-axis SCC based tracking algorithm using extended mean shift iterations

In this subsection, we continue addressing our notations and derivations considering the SCC with one axis $l$ and omit the index $l$ for simplicity.

#### 3.3.1. Metric based on Bhattacharyya coefficient

Similar to [9], we employ a distance with a metric structure to accommodate comparisons among various targets. Using the notations given in Section 2.2, the distance between two discrete distributions is denoted as

$$D(\mathbf{\Phi}_0) = \sqrt{1 - \rho[\mathbf{C}(\mathbf{\Phi}_0), \mathbf{M}]}. \tag{18}$$

Here,

$$\rho[\mathbf{C}(\mathbf{\Phi}_0), \mathbf{M}] = \sum_{u,v} \sqrt{C_{u,v}(\mathbf{\Phi}_0) M_{u,v}}, \tag{19}$$

which is the sample estimate of the Bhattacharyya coefficient between the $\mathbf{C}(\mathbf{\Phi}_0)$ and $\mathbf{M}$ [21].

#### 3.3.2. Target localization

The target localization problem aims to minimize the distance defined in Eq. (18), which can be solved by running the mean shift iterations. Using the notations given in Section 2.3, the kernel is recursively moved from the current location and orientation $\mathbf{\Phi}_0$ according to the moving vector,

$$\Delta\mathbf{\Phi} = \frac{\sum_{i,j,r} \mathbf{\Phi}_{ijr} w_{ijr} g(\| \frac{\mathbf{\Phi}_0 - \mathbf{\Phi}_{ijr}}{\mathbf{h}} \|^2)}{\sum_{i,j,r} w_{ijr} g(\| \frac{\mathbf{\Phi}_0 - \mathbf{\Phi}_{ijr}}{\mathbf{h}} \|^2)} - \mathbf{\Phi}_0, \tag{20}$$

where

$$w_{ijr} = \sum_{u,v} \sqrt{M_{u,v}/C_{u,v}(\mathbf{\Phi}_0)} \cdot \delta[I(\mathbf{P}^0_{ijr}) - u]\delta[I(\mathbf{P}^1_{ijr}) - v], \tag{21}$$

and $g(\mathbf{x}) = -k'(\mathbf{x})$, recall that $k$ is the profile of a kernel $K$, as described for Eq. (16). Derivations to obtain Eq. (20) are provided in Appendix A.

It is interesting to see that the $\sqrt{M_{u,v}/C_{u,v}(\mathbf{\Phi}_0)}$ term here is quite consistent with the $\sqrt{H_u^M/H_u^C}$ term discussed in Section 3.2.

### 3.4. Multi-axis SCC for object tracking

In Section 3.3, discussions focus on single-axis SCCs. The notions are generalized in this section to provide definitions for the general multi-axis SCCs. Specifically, we vertically stack the correlograms defined by Eqs. (11) and (14), which yields

$$\mathbf{M}(L) = \begin{pmatrix} \mathbf{M}_1 \\ \vdots \\ \mathbf{M}_L \end{pmatrix} = \alpha \begin{pmatrix} (\mathbf{U}_M^1)^T K(\mathbf{0}) \\ \vdots \\ (\mathbf{U}_M^L)^T K(\mathbf{0}) \end{pmatrix}, \tag{22}$$

and

$$\mathbf{C}(L,\mathbf{\Phi}_0)=\begin{pmatrix}\mathbf{C}_1(\mathbf{\Phi}_0)\\ \vdots\\ \mathbf{C}_L(\mathbf{\Phi}_0)\end{pmatrix}=\beta\begin{pmatrix}(\mathbf{U}_C^1)^TK(\mathbf{\Phi}_0)\\ \vdots\\ (\mathbf{U}_C^L)^TK(\mathbf{\Phi}_0)\end{pmatrix}. \tag{23}$$

A multi-axis correlogram provides a more stable solution than using one corresponding single-axis SCC in that when a certain single-axis SCC produces unstable solution, the other ones can reduce its influence on the final solution. Theoretical justifications are given in Section 4.

### 3.5. Multi-axis SCC based tracking algorithm

The general SCC based tracking algorithm in an extended mean shift tracking framework is presented as follows:

**Algorithm 1.** SCC based gradient descent algorithm in the location–orientation joint domain

```
Input: Object location and orientation of the previous frame
    $\mathbf{\Phi}_0=\mathbf{\Phi}^{t-1}$
Output: Object location and orientation for the current frame $\mathbf{\Phi}^t$
• Initialize the location and orientation of the target in the current frame with $\mathbf{\Phi}_0$. Evaluate
    $\rho(\mathbf{\Phi}_0)=\sum_l\sum_{u,v}\sqrt{C^l_{u,v}(\mathbf{\Phi}_0)M^l_{u,v}}$,
  $\mathbf{M}$ here are fixed while $\mathbf{C}(\mathbf{\Phi}_0)$ need to be updated for each iteration.
• Derive the weights $\{w_{ijr}\}$ for pairs $\{\mathbf{\Phi}_{ijr}\}$ along the directions around the chosen axis/axes according to Eq. (21),
    $w_{ijr}=\sum_{u,v}\sqrt{M_{u,v}/C_{u,v}(\mathbf{\Phi}_0)}\cdot\delta[I(\mathbf{P}^0_{ijr})-u]\delta[I(\mathbf{P}^1_{ijr})-v]$.
  For the $l$th axis, $\mathbf{M}$ and $\mathbf{C}$ are substituted by $\mathbf{M}_l$ and $\mathbf{C}_l$.
• Based on the mean shift vector, derive the new location and orientation of the target according to Eq. (20) and obtain
    $\mathbf{\Phi}_1=\frac{\sum_{i,j,r}\mathbf{\Phi}_{ijr}w_{ijr}g(\|\frac{\mathbf{\Phi}_0-\mathbf{\Phi}_{ijr}}{\mathbf{h}}\|^2)}{\sum_{i,j,r}w_{ijr}g(\|\frac{\mathbf{\Phi}_0-\mathbf{\Phi}_{ijr}}{\mathbf{h}}\|^2)}$
  with all pairs along the directions around the chosen axis/axes and with corresponding distances.
• Evaluate
    $\rho(\mathbf{\Phi}_1)=\sum_l\sum_{u,v}\sqrt{C^l_{u,v}(\mathbf{\Phi}_1)M^l_{u,v}}$.
  if $\rho(\mathbf{\Phi}_1)<\rho(\mathbf{\Phi}_0)$, let $\mathbf{\Phi}_1\leftarrow\frac{1}{2}(\mathbf{\Phi}_0+\mathbf{\Phi}_1)$.
• If $\|\mathbf{\Phi}_1-\mathbf{\Phi}_0\|<\varepsilon$, stop; otherwise, set $\mathbf{\Phi}_0\leftarrow\mathbf{\Phi}_1$ and go to the 1st step.
```

### 3.6. Scale adaptation

The scale adaptation scheme exploits the property of the SCC distance to be invariant to object scale changes. We apply the simple method of modifying the radius of the kernel with a certain fraction, i.e., +/− 10%, and choosing the radius yielding the best match to the target model. Different from the color histogram based methods, the pair distance of the SCC needs to be changed proportional to the kernel radius. To make scale adaptation more stable, an Infinite Impulse Response (IIR) filter is used to derive the new radius by calculating $\mathbf{h}_{new}=\gamma\mathbf{h}_{cur}+(1-\gamma)\mathbf{h}_{prev}$, where $\mathbf{h}_{new}$, $\mathbf{h}_{cur}$ and $\mathbf{h}_{prev}$ denote the kernel radius of the new frame, the current measurement and the previous kernel radius, respectively; $\gamma$ is a coefficient for the weight whose default value is 0.15 in our implementation. This calculation takes both the current measurement and the history information into consideration. Further, to avoid unnecessary shrinkage [7] which is especially likely to happen when partial occlusion exists, decisions of decreasing the radius are made with more caution, i.e., we allow to increase the radius when the correlation (calculated by Eq. (19) is $\geqslant$1% larger than that for the previous frame while size reduction is allowed when the correlation is at least 1.5% larger. In addition, a larger weight of history information ($\gamma=0.1$) is used for radius decrease. The adaptation of the radius provides superior results when compared with the fixed radius procedure.

## 4. Motion observability analysis of the tracking algorithm

Most kernel based algorithms risk motion unobservability. The proposed SCC based tracking method enables the detection of both translational and rotational motion, therefore reliable tracking in this context requires that both types of motion be distinctly observed and reliably recovered. Important insight can be gained when we reformulate the SCC based tracking, and observe the solution of the motion vector in a more direct way.

### 4.1. The objective function and solution for reliable tracking

In this subsection, we again assume the single-axis case for simplicity. Multi-axis cases would be analyzed in the next subsection. For the mean shift based tracking algorithms, the objective of the tracking procedure is to seek the maximum of the Bhattacharyya coefficient, as mentioned in Section 3.3.2. Its well known connection with the Matusita metric [5,14] opens the possibility that we analyze the Matusita metric other than the Bhattacharyya coefficient to better illustrate the inherent problem of kernel based tracking.

Using the notations given in Section 2.2, the Matusita metric is defined as

$$D_M(\Phi)=\|\sqrt{\mathbf{M}}-\sqrt{\mathbf{C}(\mathbf{\Phi})}\|^2=\sum_{u,v}(\sqrt{M_{u,v}}-\sqrt{C_{u,v}(\mathbf{\Phi})})^2, \tag{24}$$

where $\mathbf{\Phi}$ is the object position and orientation in the current frame. The square root operators over the vectors $\mathbf{M}$ and $\mathbf{C}(\mathbf{\Phi})$ are taken to apply componentwise operation to the vector arguments. Denoting $\rho$ as the Bhattacharyya coefficient yields the following expression as its relation with the Matusita metric $D_M(\mathbf{\Phi})=2(1-\rho(\mathbf{\Phi}))$.

According to Eq. (24), the objective of tracking using the Matusita metric can be described as

$$argmin\|\sqrt{\mathbf{M}}-\sqrt{\mathbf{C}(\mathbf{\Phi})}\|^2. \tag{25}$$

Recall that $g(\mathbf{x})=-k'(\mathbf{x}),k(\|\mathbf{x}\|^2)=K(\mathbf{x})$, we have

$$\nabla_{\mathbf{\Phi}}K\left(\frac{\mathbf{\Phi}_{ijr}-\mathbf{\Phi}_0}{\mathbf{h}}\right)=\frac{2}{\|\mathbf{h}\|^2}(\mathbf{\Phi}_{ijr}-\mathbf{\Phi}_0)g\left(\left\|\frac{\mathbf{\Phi}_{ijr}-\mathbf{\Phi}_0}{\mathbf{h}}\right\|^2\right), \tag{26}$$

stacking the derivative of kernel functions yields

$$\mathbf{J}_K(\mathbf{\Phi}_0)=\begin{bmatrix}\frac{2}{\|\mathbf{h}\|^2}(\mathbf{\Phi}_{111}-\mathbf{\Phi}_0)g(\|\frac{\mathbf{\Phi}_{111}-\mathbf{\Phi}_0}{\mathbf{h}}\|^2)\\ \frac{2}{\|\mathbf{h}\|^2}(\mathbf{\Phi}_{112}-\mathbf{\Phi}_0)g(\|\frac{\mathbf{\Phi}_{112}-\mathbf{\Phi}_0}{\mathbf{h}}\|^2)\\ \vdots\\ \frac{2}{\|\mathbf{h}\|^2}(\mathbf{\Phi}_{WHK}-\mathbf{\Phi}_0)g(\|\frac{\mathbf{\Phi}_{WHK}-\mathbf{\Phi}_0}{\mathbf{h}}\|^2)\end{bmatrix}\in\mathbb{R}^{N_0}. \tag{27}$$

Denote $\Delta\mathbf{\Phi}$ as the motion vector to be estimated, $\mathbf{\Phi}_0$ as the initialized object position and orientation for the current frame and $diag(\mathbf{C}(\mathbf{\Phi}_0))$ the matrix with $\mathbf{C}(\mathbf{\Phi}_0)$ on its diagonal. Applying a Newton-style procedure, the optimization problem of Eq. (25) can be converted to a more explicit form (derivations provided in Appendix B)

$$diag(\mathbf{C}(\mathbf{\Phi}_0))^{-\frac{1}{2}}\mathbf{U}_C^T\mathbf{J}_K(\mathbf{\Phi}_0)\Delta\mathbf{\Phi} = 2(\sqrt{\mathbf{M}} - \sqrt{\mathbf{C}(\mathbf{\Phi}_0)}). \tag{28}$$

Denoting $\mathbf{A} = diag(\mathbf{C}(\mathbf{\Phi}_0))^{-\frac{1}{2}}(\mathbf{U})_C^T\mathbf{J}_K(\mathbf{\Phi}_0) \in \mathbb{R}^{M_0\times 3}$ and converting the matrix before $\Delta\mathbf{\Phi}$ to a square matrix for further analysis, we obtain

$$\mathbf{A}^T\mathbf{A}\Delta\mathbf{\Phi} = 2\mathbf{A}^T(\sqrt{\mathbf{M}} - \sqrt{\mathbf{C}(\mathbf{\Phi}_0)}). \tag{29}$$

Different from the previous work that discusses the limitations of the kernel based methods [15], the motion vector $\Delta\mathbf{\Phi}$ in our tracking framework is in a joint domain of two translational dimensions and one rotational dimension. Therefore reliable estimation requires the motion vector to be unique and stable for both types of motion. In the following, we provide theoretical criteria for the judgement of a solution based on the above equation:

- The solution to the optimization problem is unique if and only if the $3 \times 3$ matrix $\mathbf{A}^T\mathbf{A}$ is of full rank.
- The stability of the solution depends on the magnitude of its condition number.

As a consequence, the condition number is naturally employed as the criterion to judge the numerical stability of the solution. If the condition number is large, slight perturbation of the SCC tends to make the motion vector change significantly. In the single-axis case, the SCC with the parameters $(\theta, d)$ corresponding to the smallest condition number of $\mathbf{A}^T\mathbf{A}$ is the optimal SCC.

### 4.2. SCC with multiple axes

Formulating the objective for this 3-dimensional case Eq. (28) provides a principled way to evaluate multi-axis SCCs. Specifically, for the $l$-th axis in a $L$-axis SCC, we denote

$$\mathbf{A}_l = diag(\mathbf{C}^l(\mathbf{\Phi}_0))^{-\frac{1}{2}}(\mathbf{U}_C^l)^T\mathbf{J}_K^l(\mathbf{\Phi}_0), l = 1, .., L, \tag{30}$$

and

$$\mathbf{A}(L) = \left[A_1^T A_2^T \cdots A_L^T\right]^T \in \mathbb{R}^{(L\times M_0)\times 3}, \tag{31}$$

then we obtain

$$\mathbf{A}^T(L)\mathbf{A}(L) = \sum\nolimits_{l=1}^{L}\mathbf{A}_l^T\mathbf{A}_l. \tag{32}$$

In this paper, we explore further into this multi-axis problem considering the simple yet effective two-axis cases. A useful property of the semi-positive definite matrices $A_1^T A_1$ and $\mathbf{A}_2^T\mathbf{A}_2$ states as,

$$\begin{aligned} min(cond(\mathbf{A}_1^T\mathbf{A}_1), cond(\mathbf{A}_2^T\mathbf{A}_2)) &\leqslant cond(\mathbf{A}^T(2)\mathbf{A}(2)) \\ &\leqslant max(cond(\mathbf{A}_1^T\mathbf{A}_1), cond(\mathbf{A}_2^T\mathbf{A}_2)), \end{aligned} \tag{33}$$

where $\mathbf{A}^T(2)\mathbf{A}(2) = \mathbf{A}_1^T\mathbf{A}_1 + \mathbf{A}_2^T\mathbf{A}_2$.

The inequalities indicate that the condition number of a two-axis correlogram is between the two condition numbers of the corresponding single-axis correlograms. A consequence is that when more axes are used, unfavorable condition numbers are less possible to be generated, since it requires all corresponding single-axis correlograms to have sufficiently large condition numbers.

### 4.3. Visual interpretation of SCC patterns

The pattern of SCC determines the motion estimation outcome. SCCs of certain patterns are destined to cause tracking failure. To provide a visual interpretation on the SCC patterns, we make further analysis into the matrix $A$, which is

$$diag(\mathbf{C}(\mathbf{\Phi}_0))^{-\frac{1}{2}}\mathbf{U}_C^T\mathbf{J}_K(\mathbf{\Phi}_0) = \begin{bmatrix} \frac{2}{\|\mathbf{h}\|^2}C_{11}^{-\frac{1}{2}}\sum\limits_{I(\mathbf{\Phi}_{ijr})=C_{11}}(\mathbf{\Phi}_{ijr} - \mathbf{\Phi}_0)g(\|\frac{\mathbf{\Phi}_{ijr}-\mathbf{\Phi}_0}{\mathbf{h}}\|^2) \\ \vdots \\ \frac{2}{\|\mathbf{h}\|^2}C_{1m}^{-\frac{1}{2}}\sum\limits_{I(\mathbf{\Phi}_{ijr})=C_{1m}}(\mathbf{\Phi}_{ijr} - \mathbf{\Phi}_0)g(\|\frac{\mathbf{\Phi}_{ijr}-\mathbf{\Phi}_0}{\mathbf{h}}\|^2) \\ \vdots \\ \frac{2}{\|\mathbf{h}\|^2}C_{m1}^{-\frac{1}{2}}\sum\limits_{I(\mathbf{\Phi}_{ijr})=C_{m1}}(\mathbf{\Phi}_{ijr} - \mathbf{\Phi}_0)g(\|\frac{\mathbf{\Phi}_{ijr}-\mathbf{\Phi}_0}{\mathbf{h}}\|^2) \\ \vdots \\ \frac{2}{\|\mathbf{h}\|^2}C_{mm}^{-\frac{1}{2}}\sum\limits_{I(\mathbf{\Phi}_{ijr})=C_{mm}}(\mathbf{\Phi}_{ijr} - \mathbf{\Phi}_0)g(\|\frac{\mathbf{\Phi}_{ijr}-\mathbf{\Phi}_0}{\mathbf{h}}\|^2) \end{bmatrix}. \tag{34}$$

To obtain a unique solution for $\Delta\mathbf{\Phi}$ in Eq. (28), at least three of the row vectors of the above matrix need to be linearly independent. Due to the fact that the SCC sums to 1, at least four different features are necessary to track the two degrees freedom of translation and one degree freedom of orientation. In this paper, we analyze two typical image patterns that are inherently unobservable to certain motions.

#### 4.3.1. Concentric circles (Fig. 5a)

In color histogram based kernel methods, *concentric circles* are regarded as a degenerated case [15], where translation cannot be detected. In this SCC based kernel methods, due to the spatial information encoded in the pixel pairs, translation along the SCC axis can now be observed.

Without loss of generality, we set the SCC axis to be along the $y$ direction, as shown in Fig. 5a, and validate the translation observability through Eq. (34) by examining the weighted distance vectors $(\mathbf{\Phi}_{ijr} - \mathbf{\Phi}_0)g(\|\frac{\mathbf{\Phi}_{ijr}-\mathbf{\Phi}_0}{\mathbf{h}}\|^2)$ for pixel pairs of two certain distinct colors. The $x^c$ component (recall that $\mathbf{\Phi} = [x^c, y^c, \theta]^T$) of this term is cancelled out by every two corresponding pixel pairs, i.e., two symmetric pairs w.r.t. the SCC axis, like pairs $a$ and $b$ in Fig. 5a, since for each such pixel pairs of the same colors, the $x^c$ component of the weighted distances sum up to 0. However, the $y^c$ component

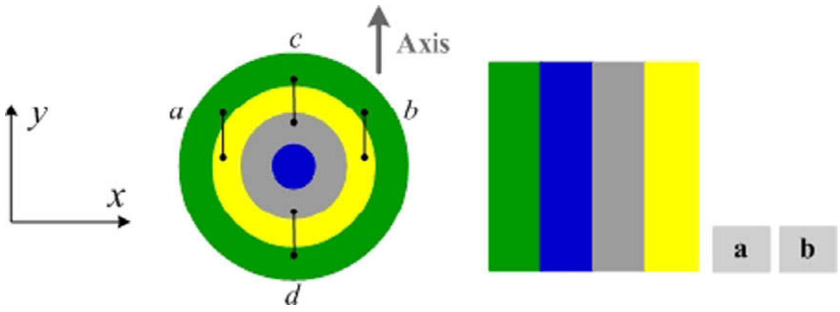

**Fig. 5.** Illustration for visual interpretation of SCC choice.

of the term cannot be always cancelled out by corresponding pairs, i.e., symmetric ones w.r.t. the $x$ direction. For example, pair $c$ in Fig. 5a is of color $(i, j)$, while its corresponding pair (pair $d$) is of color $(j, i)$, therefore they can neither cancel out each other for $C_{ij}$, nor for $C_{ji}$. Since the representation is rotationally-symmetric, the $\theta$ components are all cancelled out. As a result, the row vectors in Eq. (34) is $\kappa[0, 1, 0]^T$, where $\kappa$ is simply a coefficient. The form of $\kappa[0, 1, 0]^T$ means that among the three degrees of motion, only the translation along the SCC axis can cause sufficient changes to the SCC. Although one may use two axes to detect translation in both dimensions, the blindness to rotation is the inherent limitation of *concentric circles*.

### *4.3.2. Parallel stripes (Fig. 5b)*

Independent of the axis choice in the SCC, the *parallel stripes* pattern is sensitive to motion along the $x$ direction, while blind to motion along the $y$ direction. However, its observability to rotation depends on the direction of the SCC axis. If the axis is defined to be along the $x$ direction, then the elements for the orientation dimension in $(\mathbf{\Phi}_{ijr} - \mathbf{\Phi}_0)g(\|\frac{\mathbf{\Phi}_{ijr}-\mathbf{\Phi}_0}{\mathbf{h}}\|^2)$ cancel out. Intuitively, this means that slight rotation does not cause enough change to the SCC. On the other hand, if the axis is some degrees away from the $x$ direction, then rotation makes a difference in the SCC by causing some pixels in the boundary of two stripes to be the other color.

We evaluated the quantitative relationship between condition numbers and parameters $(\theta, d)$ for the image patch of the *parallel stripes* pattern, shown in Fig. 6a. The axis direction $\theta$ is defined in the object coordinate system, i.e., with along the $x$ direction being 0 degree, and increases counterclockwise. We observe the condition numbers of single-axis SCCs with different axis directions and those of two-orthogonal-axis SCCs, where the directions of

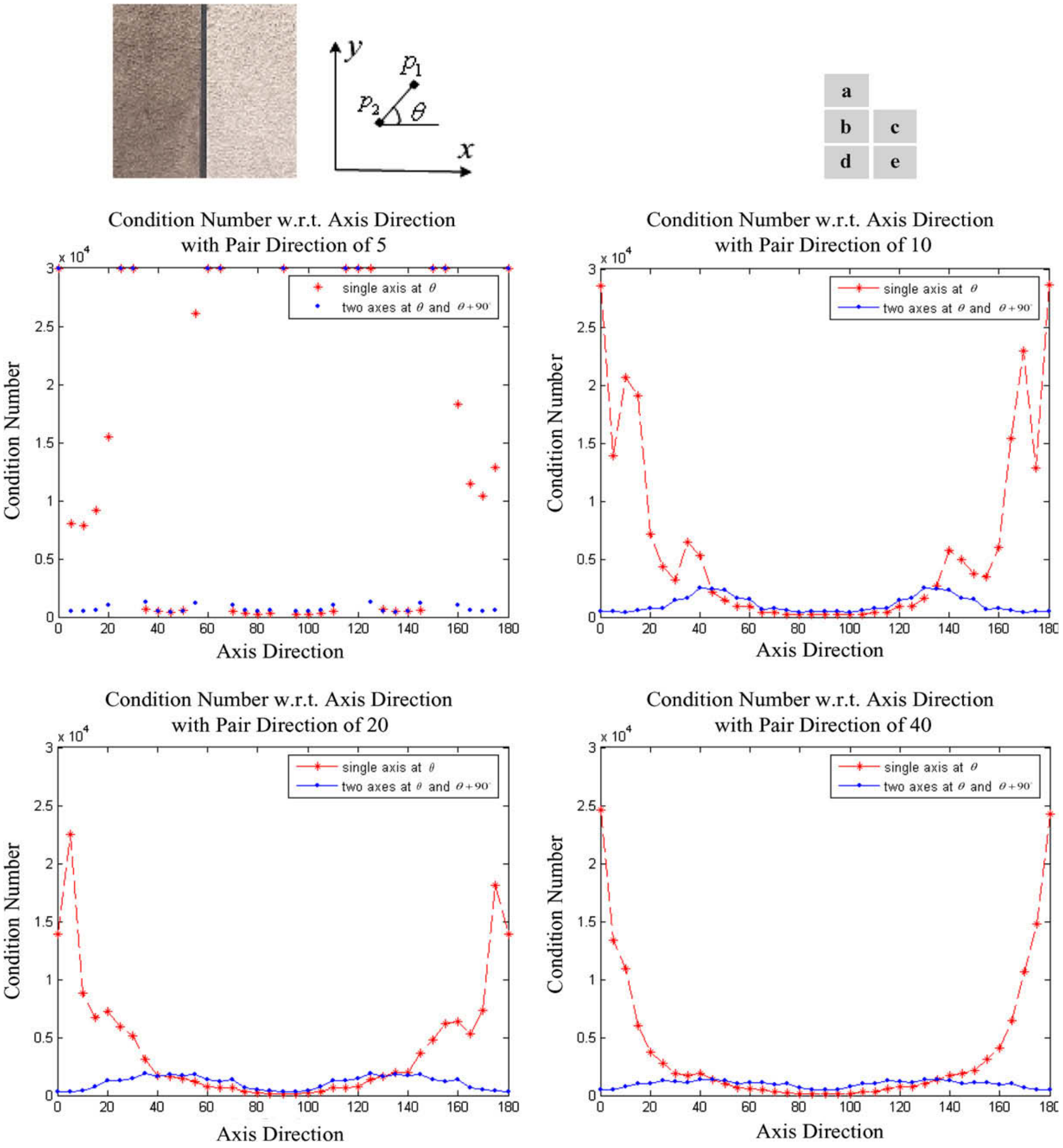

**Fig. 6.** Quantitative relationship between condition numbers and parameters $(\theta, d)$: (a) image patch ($96 \times 96$), (b)–(e) condition numbers w.r.t. SCC axis directions with different pair distances: (b) $d = 5$ (symbols at the top line represent overflow values), (c) $d = 10$, (d) $d = 20$, (e) $d = 40$.

the axes are $\theta$ and $\theta + 90$. Fig. 6b–e show the relationship of condition numbers w.r.t. different axis directions with pair distances of 5, 10, 20, and 40, respectively. From the illustrated outputs, following conclusions can be made:

- The pair distance cannot be too small for stable tracking. This is due to the discrete nature of the image and the fact that colors of each pixel pair in the SCC representation are assigned in a nearest-neighbor manner. For close-by pixel pairs, small image rotation may not cause any changes to the SCC; in other words, SCCs with small pair distances are not sensitive to rotation. As shown in Fig. 6, the results with pair distance of 5 pixels (Fig. 6b) are much poorer than those with longer distances (Fig. 6c–e).
- Some axis directions are more favorable than others in terms of stability. For the patch shown in Fig. 6a, the most favorable axis direction for single-axis SCCs is along the $y$ direction ($\theta = 90$ in Fig. 6b–e).
- When two axes are used, due to the inequality given in Eq. (33), no matter what the directions of the axes are, the condition number is between the two condition numbers generated independently by the two single-axis SCCs. In Fig. 6b–e, outputs show that the average condition numbers provided by the two-orthogonal-axis SCCs are significantly smaller than those generated by the single-axis ones.

These typical patterns occur, though not frequently, in real world scenarios (e.g., the wheels of vehicles). However, many factors in real images such as noises, partial occlusions make the patterns in real applications more observable than the illustrations as shown in Fig. 5.

### 4.4. SCC based tracking algorithm with observability analysis

The SCC based tracking algorithm with observability analysis is summarized as follows:

**Algorithm 2.** SCC based tracking algorithm with observability analysis

Input: Object location and orientation of the previous frame $\mathbf{\Phi}_0 = \mathbf{\Phi}^{t-1}$
Output: Object location and orientation for the current frame $\mathbf{\Phi}^t$

- If rigorous speed requirement is imposed, obtain the optimal axis based on the criteria discussed in Section 4.1; otherwise, set two arbitrary orthogonal directions as axes. (See Section 5.2.1 for details).
- Apply Algorithm 1 (Section 3.5).

## 5. Implementation details

### 5.1. Computational complexity analysis

As the pair direction and distance are pre-determined for a SCC, representing an object using SCC requires $O(N)$ time ($N$ is the number of pixels in the image region), which is similar to that of the color histogram. When calculating the goodness-of-fit, however, the SCC requires $O(m^2)$ computations ($m$ is the number of bins after quantization for each color channel), while the color histogram needs only $O(m)$ time.

In our implementation, where the extended mean shift based algorithm is employed as the central computational module, the additional computational cost, compared with the original mean shift based tracking algorithm, arises from the following two aspects: firstly, the proposed algorithm runs the mean shift procedure in a space with one more dimension than the original one. To obtain the new orientation at the basin of attraction of the similarity function in the orientation dimension, pixel pairs along directions around the chosen axis need to be counted for the computation. $L$ axes with $\Theta$ directions around each axis increase the computational burden of the relevant part by a factor of $L \times \Theta$, compared with the conventional histogram based methods. Secondly, the number of iterations necessary for each frame influences the computational complexity of the method. Therefore an efficient computation of the algorithm requires both the numbers of axes and directions counted around each axis and the number of iterations for each frame to be as small as possible.

#### 5.1.1. Quantization scheme in the orientation space

Two parameters determine the number of directions around the chosen axis: the quantization range and the quantization unit. The basic principle used to specify the two parameters is that the quotient of them is small while sufficient information is contained form the basin of attraction. On one hand, a wide range of orientation is both inefficient and more likely to induce confusions due to locality principle. On the other hand, the range cannot be too small, since rotation can never be recovered if it is out of the quantization range. Therefore a reasonable compromise is to choose the smallest range that is supposed to cover all inter-frame rotations. In our implementation, the default values of the quantization range and unit are 30° and 5°, therefore we have $\Theta = 5$ as non-zero weighted directions.

#### 5.1.2. Iterations of the mean shift procedure

For discrete data, the number of iterations depends on the employed kernel. In our implementation, Epanechnikov kernel is employed, therefore the $g$ in the motion vector computation (Eq. (20)) is a uniform function. However, the $w$ term imposes weights on the pixel pairs, which makes the mean shift procedure to be possibly infinitely convergent [8]. More detailed description of the convergence property can be found at [2,6]. A practical way to stop the iterations is to set a suitable lower bound of the magnitude of the motion vector.

### 5.2. Axis and pair distance selection

The direction and distance of the pixel pairs determine a unique SCC.

#### 5.2.1. Discussions on the axis selection

As discussed in Section 4, for some image patterns, the SCCs with certain axes are sensitive to motion, while others are not. For most real objects, textures are irregular enough, i.e., condition numbers small enough, to avoid those extreme cases as discussed in Section 4.3, therefore for most tasks, the two-orthogonal-axis SCC suffices for reliable tracking. However, in applications where speed is an important factor, single-axis representation is greatly favored for efficiency considerations. In this case, we search for the optimal axis direction in the orientation space to obtain the corresponding SCC with the smallest condition number to ensure reliable tracking.

#### 5.2.2. Discussions on the pair distance selection

The pair distance is an important parameter in that it influences not only the SCC's sensitivity to rotation, but also the stability of the solution. On one hand, the larger the pair distance, the more observable the orientation changes, as explained in Section 4.3; on the other hand, SCC with a large pair distance tends to end up with having too few pairs counted (both pixels should be in the tracking window), which decreases the stability of the tracking. By trial and error, we set the default distance to be

$max((l+w)/8, 10)$, where $l$ and $w$ are the length and width of the kernel size. The lower bound of 10 pixels ensures the stability of the tracker when the object is small.

### 5.3. Discussions on the color space

Generally speaking, although the gray-scale color space contains less information than the RGB or HSV color spaces, it has its own benefits. Firstly, the size of the SCC is $O(m^2)$ instead of $O(m^6)$ for the three channel color space ($m$ is the number of bins after quantization for each color channel), therefore the storage requirements are considerably smaller. Secondly, the average number of entries per SCC bin increases from $O(S_I/m^6)$ to $O(S_I/m^2)$, where $S_I$ denotes the size of the image. This increase improves the statistical reliability of the SCC. However, in some cases, the background and foreground color are too similar when mapping to the gray-scale color space, then the SCC in RGB or HSV space needs to be taken as the feature.

## 6. Applications

The proposed object representation, with its ability to model spatial information and orientation variations, can be used to develop computer vision algorithms for a wide variety of tasks. The simplicity and versatility of the representation enables the design of algorithms in which the user controls performance through a small set of parameters. Several related applications in object tracking are discussed in the following sections: vehicle and pedestrian tracking, and structured and articulated object tracking. Comparisons between the SCC based tracker and the standard mean shift tracker are carried out over six sequences, the *car-chasing* one, the *arm* one, three sequences from *PETS 2001* dataset, and one sequence from *PETS 2004* dataset. These sequences contain typical scenarios where the standard mean shift (MS) tracking algorithm does not work well, while the SCC based one can effectively solve the problems.

### 6.1. Vehicle and pedestrian tracking

Real-time tracking imposes rigorous requirements on the algorithm speed, and it is important to use fewer axes for the model definitions and the motion vector calculations, as mentioned in Section 5.2.1. To more safely use single-axis SCC in this category of application, the optimal single-axis SCC should be obtained for object representation. The proposed tracking algorithm has been tested on vehicle and pedestrian sequences under various environmental conditions.

#### 6.1.1. Car-chasing

The *car-chasing* is a live video sequence of 2250 frames with 350 × 480 pixels each. The sequence has been used to test the original MS based tracking algorithm, the proposed SCC based tracking algorithm with and without optimal SCC selection.

In Fig. 7a-1, b-1 and c-1, we show sample frames of the tracking procedure using the SCC based tracker with optimal SCC selection. The tracker keeps track of the car through the entire 2250 frames. The kernel radius is initialized as $h_{SCC_{opt}} = (19, 27)$. The optimal pair direction (in the object coordinate system, which is determined by the bounding box of the object) and distance are $\theta_{SCC_{opt}} = 75^\circ$, $d_{SCC_{opt}} = 12$ (one eighth of the sum of the object length and width). In the implementation, the effective range of angles around $\theta$, with non-zero kernel weights, is chosen to be $20^\circ$ with $5^\circ$ as quantization unit (5 directions at $\theta$, $\theta \pm 5^\circ$, $\theta \pm 10^\circ$ are counted). The Epanechnikov profile was used for SCC computation, therefore, the mean shift iterations were computed with the uniform profile. The results prove that the new tracker deals with object rotations, heavy occlusions, background clutters and scale changes elegantly. In addition, the motion blurs present in some frames (last three frames in Fig. 7a did not influence the tracking performance. The same effect, however, can severely distract contour based trackers.

In comparison, Fig. 7b-2 and c-2 demonstrate possible problems in vehicle tracking using the conventional MS tracking methods: (1) Loss of tracking tends to occur when the car makes turns (Fig. 7b-2); (2) fixed orientation of the tracking window causes scale adaptation difficult and the mismatch of the window to the object makes the tracking too sensitive to background clutter (Fig. 7c-2). To compare the performance of the MS tracker with the optimal SCC based one, the initialized radii for the MS tracker (Fig. 7b-2 and c-2) are specified the same as the kernel radii of the corresponding frames in the tracking procedure using the optimal SCC based tracker, i.e., $h^b_{MS} = (19, 27), h^c_{MS} = (18, 26)$. Besides, Fig. 7a-2 illustrates the instability of the proposed tracker without optimal SCC selection, where the initialized radius is $h_{SCC_{nonopt}} = (19, 27)$, the pair distance is $d_{SCC_{nonopt}} = 12$, but the pair direction is not selected to be the optimal one, instead, we set it to the default direction of $\theta_{SCC_{nonopt}} = 90^\circ$. We observe that although the window direction adapts to the object, it risks failure when partial occlusions happen or background scene changes much.

The average number of iterations for the optimal SCC based tracker on the *car-chasing* sequence is 3.58 iterations per frame. In comparison, the MS tracker on the successfully tracked frames has a average of 2.77 iterations per frame. Both algorithms are implemented using the same stopping criteria for location (the MS tracker does not have such a parameter for orientation) for a fair comparison. Specifically, denote $\epsilon_{xy}$ as the stopping criteria for the translation domain, the mean shift iterations stop when the magnitude of the translation vector is less than or equal to $\epsilon_{xy} = 2$. For the SCC based method, the stopping criteria for rotation is $\epsilon_\theta = 0.5$. The results indicate that the iteration numbers of SCC is comparable with the conventional MS tracker. The small number of iterations to find the location as well as the orientation of the object is critical for a real-time tracking system. In our implementation, the MS tracker can reach processing speed of hundreds of frames per second, and the SCC based tracker runs comfortably at 30 frames per second on a 3.2 GHz PC, C++ implementation. The speed difference is largely due to the multiple directions estimated to obtain the object orientation; however, the increase of computation does not sacrifice performance as real-time processing can be easily obtained.

#### 6.1.2. Person-cart

The SCC based tracker was also applied on tracking non-rigid objects, like a person pushing a cart. The *person-cart* sequence, which contains a sharp turn, is captured with a hand-held camera. It has a total length of 90 frames and 320 × 240 pixels each frame. The initialized parameters of radius, pair directions and pair length are $(h = (32, 12), \theta_{SCC_{nonopt}} = 39^\circ, d = 11)$. The tracking results are presented in Fig. 8. The proposed tracker proved to be robust to orientation changes, partial occlusions (Fig. 8d and e) and camera motion.

#### 6.1.3. PETS 2001 data

The algorithm is further evaluated on the *PETS 2001 dataset* [1]. The main difficulty for the conventional kernel based tracking algorithms in this sequence is that due to the specific camera viewpoint, the movement of the objects (vehicles and pedestrians) exhibits the needs of not only location tracking, but also orientation adaptation. As shown in Fig. 9a-1, b-1 and c-1, the standard MS tracker cannot keep track of the entire objects due to its incapability of detecting rotational motion. The SCC based tracker, however, successfully removes such restrictions brought up by cer-

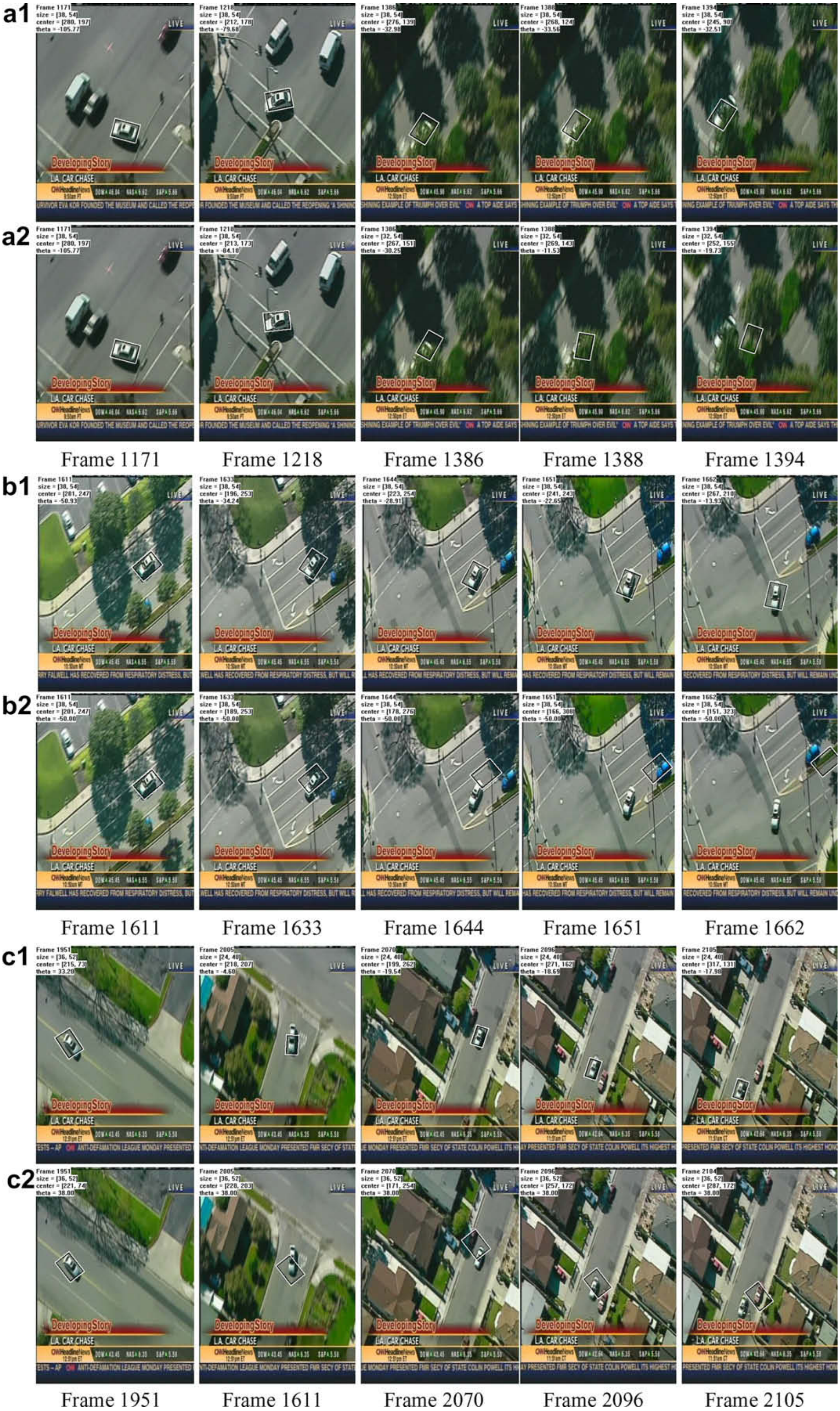

**Fig. 7.** *Car-chasing* sequence: (a-1, b-1, c-1) the optimal SCC based tracker tracks the car throughout the 2250 frames. (a-2) Non-optimal SCC based tracker fails during partial occlusion. (b-2, c-2) The MS tracker loses track of the object when (b-2) the car makes turns, (c-2) background scene is cluttered.

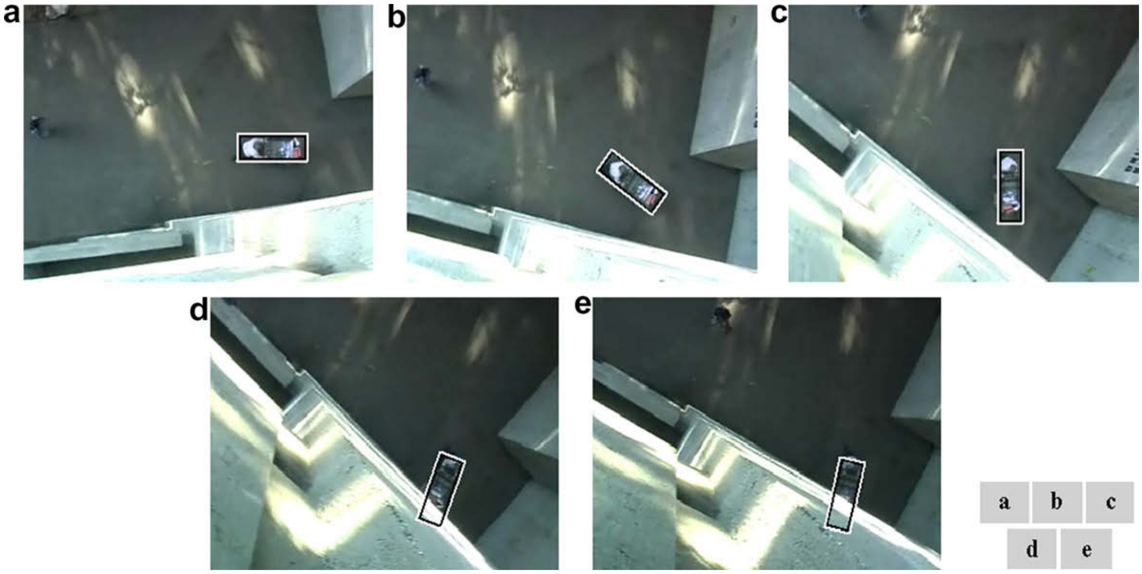

**Fig. 8.** *Person-cart* sequence: the optimal SCC based tracker is robust against object rotation, partial occlusions and camera motion.

tain object shapes and/or camera viewpoints. Sample frames of the experimental outputs are shown in Fig. 9. The initialized parameters of radii, pair directions and pair lengths for Fig. 9a-2, b-2 and c-2 are $(h^a=(14,7),\theta^a_{SCC_{opt}}=35^\circ,d^a=10),(h^b=(18,7),\theta^b_{SCC_{opt}}=81^\circ,d^b=10),(h^c=(4,9),\theta^c_{SCC_{opt}}=98^\circ,d^c=10)$.

Fig. 10 shows the number of iterations for the optimal SCC based tracker on the *White Vehicle* sequence (Fig. 9a). The average number of iterations is 2.48. Using the same stopping criteria as the SCC based tracker, the average iteration numbers of the MS tracker is 1.49 for this sequence. The extra iterations are mostly for the orientation estimation, as shown in Fig. 9a, where the object constantly changes its orientation.

We have also conducted a quantitative comparisons of the optimal SCC based tracking algorithm with the MS tracker using the 3 sequences in the *PETS 2001 Dataset*. Since the MS tracker is incapable in estimating the object orientation, we only list the position and size errors of the two methods for a fair comparison. In both algorithms, the same scale adaptation scheme, i.e., varying the object size by $+/-10\%$ and choosing the one with smallest distance [10], is implemented. We manually label ground truth for the sequences and quantitative results are shown in Table 1. All the objects are initialized using ground truth data. Tracking is deemed to fail if the tracker-identified bounding box has no overlap with the ground truth bounding box. The object centroid position error is calculated as the Euclidian distance between the centroids of the bounding boxes of the ground truth and the tracking results on frames of successful tracking. To prevent errors in frames with larger object scales from dominating the averaged error, the centroid error is normalized with respect to the ground truth length of the object's diagonal. Similarly, the size error is defined as the Euclidian distance between the two (height, width) vectors, normalized by the ground truth length of the object's diagonal.

### 6.1.4. PETS 2004 data

The proposed SCC based tracker has also been tested on multiple people tracking, where transient occlusions are handled. As shown in Fig. 11, the two interacting objects have similar color distributions, but quite different spatial arrangements, i.e., the guy wears red pants while the girl wears red jacket. The optimal SCC based tracker distinguishes the two objects well in this case.

## 6.2. Structured and articulated object tracking

There is currently little work on applying kernel based methods for structured or articulated object tracking. Fan and Wu [11] used collaborative kernels to solve the problem. To the best of our knowledge, we present the first work to handle this problem using a single kernel.

### 6.2.1. Arm

The arms, due to their elongated shape, cannot be satisfactorily tracked by the original mean shift based algorithm, as shown in Fig. 12a. The tracking window drifts along the lower-arm all the time, since the colors of the lower arm region are very similar. On the other hand, if an elongated kernel is imposed on the whole lower-arm region, it would easily get lost when the part begin rotating. This problem is solved by using the proposed SCC based tracking algorithm, which adapts the window direction throughout the sequence, as demonstrated in Fig. 12b.

### 6.2.2. Handset

We also tested the tracker using the *Handset* sequence from [11]. Although the object rotates fast (more than four revolutions in 135 frames), the tracker keeps track of the object orientation accurately, as illustrated in Fig. 13. On the other hand, both Fan and Wu's work [11] and our implementation show that the traditionally symmetric kernels, if applied independently, would drift along the object, similar to the Fig. 12a.

### 6.2.3. Multiple human parts

Other experiments of tracking multiple human parts are shown in Figs. 14 and 15. Although only color information is extracted, elegant results indicate the algorithm's potential in being a useful module in any human tracking or behavior analysis tasks.

### 6.2.4. Face

Another application of the work is to track human face when the head tilts. Fig. 16 shows sample frames of a sequence of 230 frames with 320 × 240 each, captured by a Logitech web camera. In experiments, we observed that the tracker succeeds in tracking a wide range of head tilt. The angles of head tilt obtained by the algorithm may serve as useful information for human computer interaction.

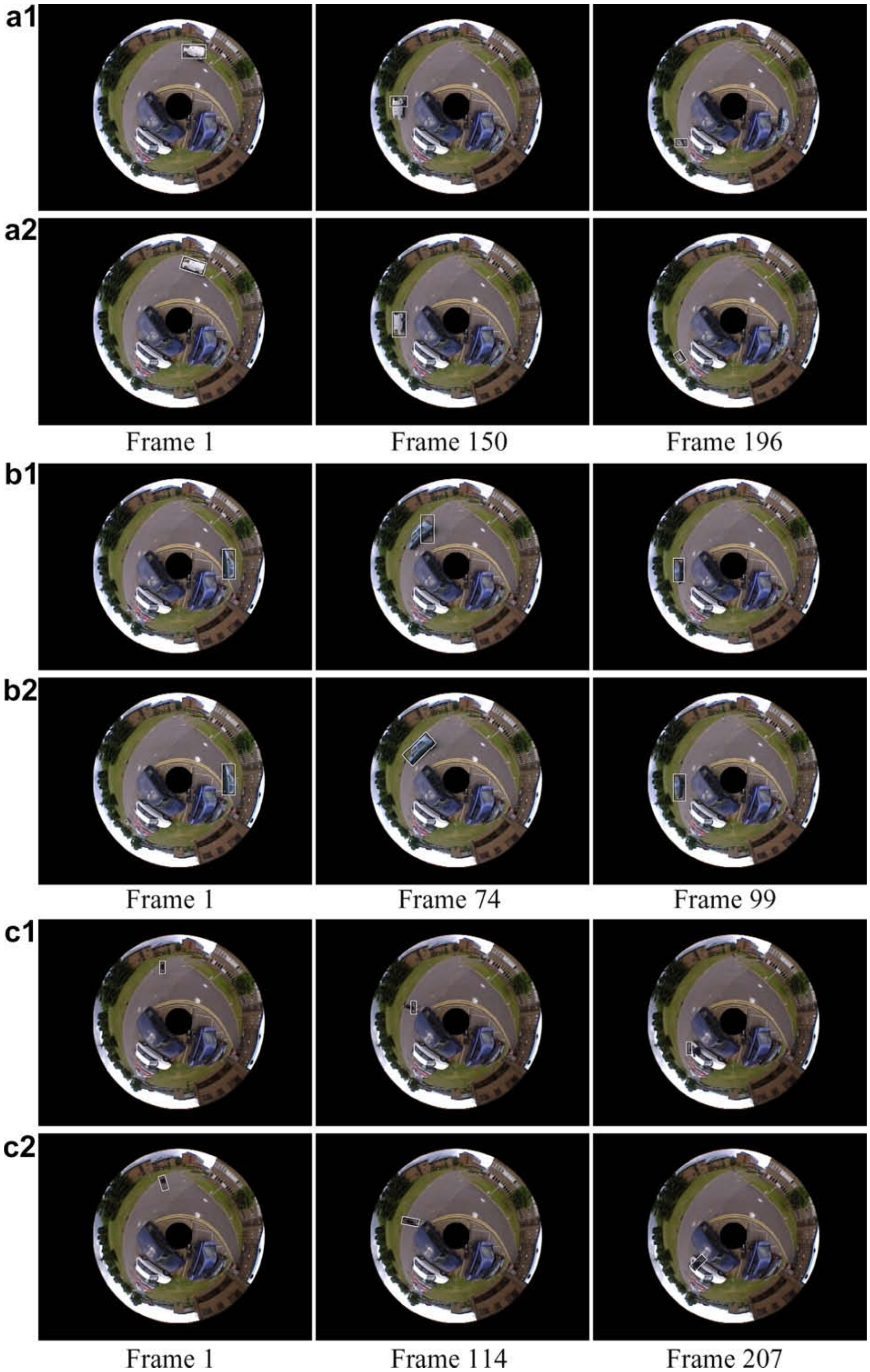

**Fig. 9.** *PETS 2001* sequences: (a-1, b-1, c-1) the MS tracker fails to keep track of the entire objects due to its incapability of detecting rotational motion. (a-2, b-2, c-2) The optimal SCC based tracker maintains a secure focus on the object throughout the sequence.

## 7. Discussions

Spatial information is an indispensable component for object representation. For the purpose of tracking, such knowledge is needed to obtain orientation information. The novelty in this paper includes a representation method based on simplified color correlogram, an extended mean shift based tracking algorithm, and a thorough analysis of the motion observability problem in the translation–rotation joint case.

Color correlogram was previously used in the image retrieval literature, yet to our knowledge, few people have been using it for object tracking. We consider the proposed approach a new effort in exploring the middle ground between the template based approaches, where lots of details are embodied, and histogram based approaches, where spatial information is totally lost. Moreover, instead of the original form, simplified versions of correlogram have been created which are sufficient to character the objects in the tracking context while greatly reduce computational complexity.

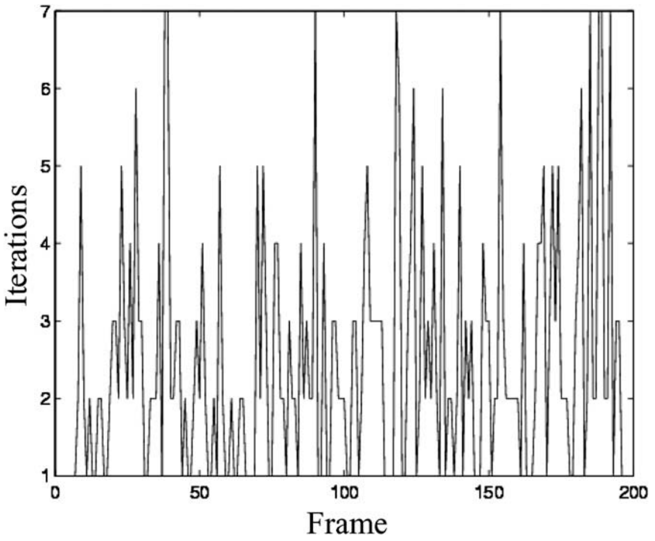

**Fig. 10.** The number of iterations vs. frame index for the *white vehicle* sequence.

**Table 1**
Quantitative results for PETS 2001 public data of the proposed SCC based tracker and its comparison with the standard mean shift (MS) tracker. In datasets with *, ground truth data is manually labelled every three frames and we count only those frames with ground truth for comparison. Other sequences have ground truth data labelled for each frame.

| Source dataset | Description/ file name | Frames tracked | | Position error | | Size error | |
|---|---|---|---|---|---|---|---|
| | | MS | SCC | MS | SCC | MS | SCC |
| PETS’01 | White vehicle | 197/197 | 197/197 | 0.311 | **0.123** | 0.290 | **0.121** |
| PETS’01 | Blue car | 102/102 | 102/102 | 0.270 | **0.099** | 0.259 | **0.132** |
| PETS’01* | Person | 210/282 | 282/282 | 0.332 | **0.103** | 0.174 | **0.092** |

Bold values correspond to smaller errors therefore better accuracy.

The gradient descent method mean shift algorithm is adopted as the central computational module and further extended to a translation–rotation joint domain to locate the object position and orientation simultaneously. The algorithm essentially belongs to computer vision tasks which are based on *in situ* optimization, which is a very powerful method, as mentioned in [8,30]. For the proposed tracker, each input data unit, a pixel pair in this case, is associated with a voting kernel to produce a more dense structure where the feature can be reliably extracted. In addition, detailed analysis into the kernel based tracking algorithm and the principled way to ensure motion observability are provided. The proposed representation in an extended mean shift tracking framework is not computationally expensive, which allows the tracker to run at a real-time performance.

Further, the representation of the simplified color correlogram can be conveniently generalized and incorporated into many tracking frameworks. In this sense, we propose a new object representation for object tracking rather than extending the mean shift algorithm.

The advantages of the proposed tracker can be summarized as so: firstly, by adding spatial information into object representation, the tracker successfully obtains its orientation as well as location, which is useful for applications like vehicle systems and human computer interaction. Secondly, the tracker performs significantly better in tracking rotational elongated objects by adapting the window direction according to object rotation, which many currently used algorithms fail to do. Such capability not only removes the restrictions on the object shape and camera viewpoint, but shows potential applications in articulated objects tracking and human behavior analysis. As demonstrated in the sequences, the tracker also preserves pleasing characteristics like being robust to partial occlusions and background clutters.

To conclude, the simplified color correlogram based tracking algorithm which captures orientation information is a valuable computational module whose versatility can make it a useful component in many vision tasks.

## Appendix A. Derivation of Eq. (20)

*The target localization problem of minimizing the distance defined in Eq. (18) can be solved by running the mean shift procedure and, in each iteration, moving the vector indicated by Eqs. (20) and (21).*

The minimization of the distance is equivalent to maximizing the Bhattacharyya coefficient, defined in Eq. (19).

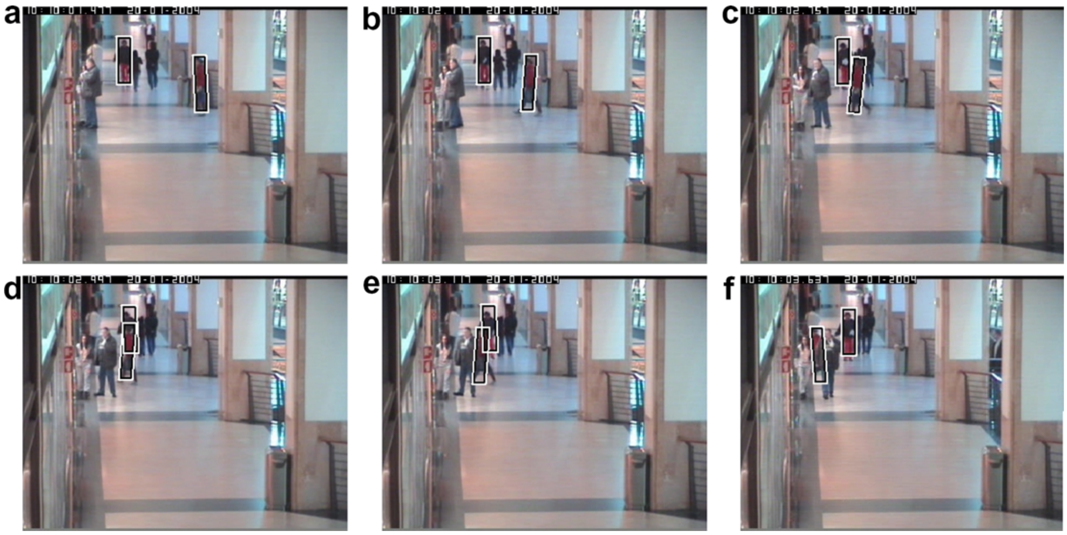

**Fig. 11.** *Multiple people* sequence: the optimal SCC based tracker is robust against interactions between objects with similar color distributions, but different color spacial arrangements. (a-f) Frames 1, 13, 26, 30, 33 and 43 are shown.

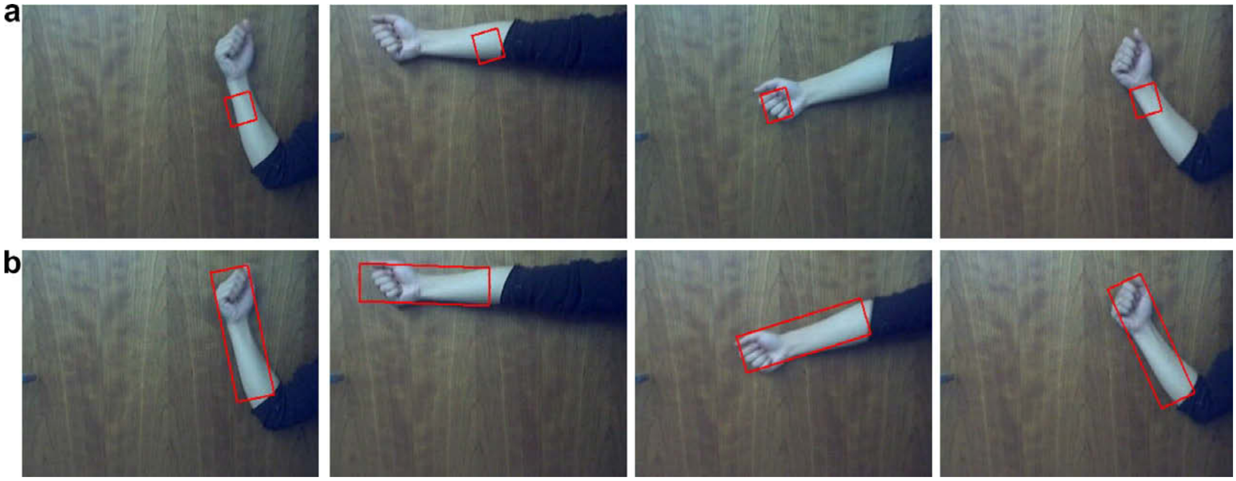

**Fig. 12.** *Arm* sequence: (a) the MS tracker drifts along the lower arm due to color similarity in the region. (b) The SCC based tracker adapts its direction and keeps track of the arm.

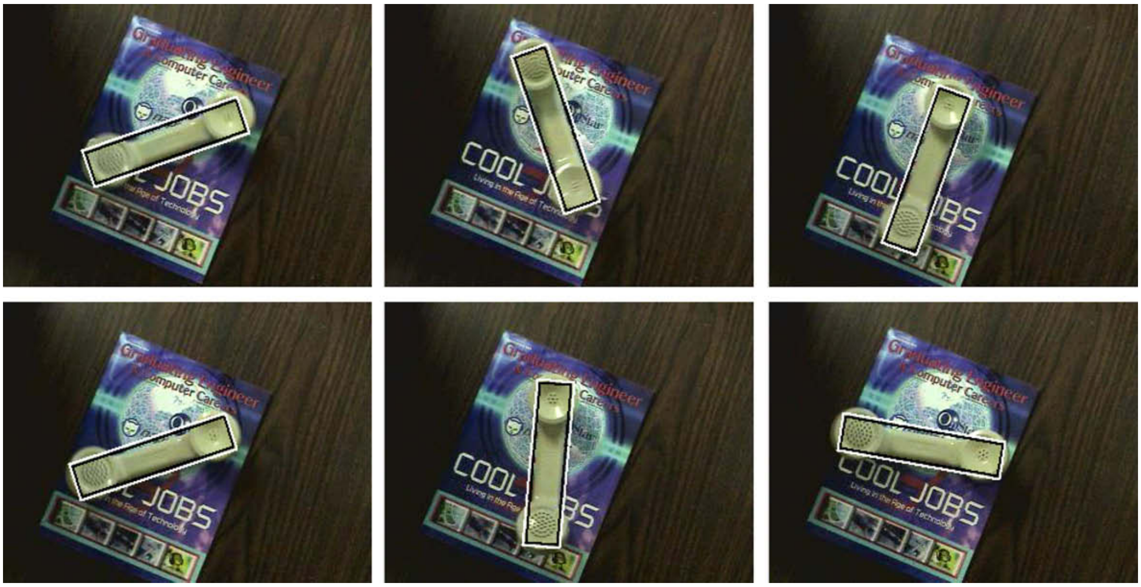

**Fig. 13.** *Handset* sequence: the SCC based tracker successfully tracks the object despite of wide rotation of the object.

Using Taylor expansion around $\mathbf{C}(\mathbf{\Phi}_0)$ and removing high order terms yields

$$\rho[\mathbf{C}(\mathbf{\Phi}),\mathbf{M}] \approx \frac{1}{2}\sum_{u,v}\sqrt{C_{u,v}(\mathbf{\Phi}_0)M_{u,v}} + \frac{1}{2}\sum_{u,v}C_{u,v}(\mathbf{\Phi}_0)\sqrt{M_{u,v}/C_{u,v}(\mathbf{\Phi}_0)}. \tag{A.1}$$

This approximation is satisfactory when we assume that the inter-frame movement is mild, which means that $\mathbf{C}(\mathbf{\Phi})$ does not change too much from the initial $\mathbf{C}(\mathbf{\Phi}_0)$.

Recalling Eq. (13) results

$$\rho[\mathbf{C}(\mathbf{\Phi}),\mathbf{M}] \approx \frac{1}{2}\sum_{u,v}\sqrt{C_{u,v}(\mathbf{\Phi}_0)M_{u,v}} + \frac{\beta}{2}\sum_{i,j,r}w_{ijr}K\left(\frac{\mathbf{\Phi}_{ijr}-\mathbf{\Phi}_0}{\mathbf{h}}\right), \tag{A.2}$$

where

$$w_{ijr} = \sum\nolimits_{u,v}\sqrt{M_{u,v}/C_{u,v}(\Phi_0)}\cdot\delta[I(\mathbf{P}^0_{ijk})-u]\delta[I(\mathbf{P}^1_{ijk})-v]. \tag{A.3}$$

The first term in Eq. (A.2) is independent of $\mathbf{\Phi}$, therefore the objective here is to maximize its second term, which represents the density estimation computed with kernel profile $k$ at $\mathbf{\Phi}$, with the data being weighted by $w$, defined in Eq. (A.3). This is realizable by applying mean shift iterations [9].

## Appendix B. Derivation of Eq. (28)

*Optimization on the objective function of argmin*$\|\sqrt{\mathbf{M}}-\sqrt{\mathbf{C}(\mathbf{\Phi})}\|^2$ *results the following equation: diag*$(\mathbf{C}(\mathbf{\Phi}_0))^{-\frac{1}{2}}\mathbf{U}_C^T\mathbf{J}_K(\mathbf{\Phi}_0)\Delta\mathbf{\Phi} = 2(\sqrt{\mathbf{M}}-\sqrt{\mathbf{C}(\mathbf{\Phi}_0)})$, *where the square root operator is taken to apply componentwise operation to the vector argument.* $\Delta\mathbf{\Phi}$ *is the motion vector including translation and rotation, and* $\mathbf{\Phi}_0$ *is the initialized object center in the current frame.*

Applying the Taylor expansion on $\sqrt{\mathbf{C}(\mathbf{\Phi})}$ yields

$$\begin{aligned}\sqrt{\mathbf{C}(\mathbf{\Phi})} &= \sqrt{\mathbf{C}(\mathbf{\Phi}_0+\Delta\mathbf{\Phi})}\\ &= \sqrt{\mathbf{C}(\mathbf{\Phi}_0)} + \frac{d\sqrt{\mathbf{C}(\mathbf{\Phi})}}{d\mathbf{\Phi}}\Big|_{\mathbf{\Phi}=\mathbf{\Phi}_0}\Delta\mathbf{\Phi} + O((\Delta\mathbf{\Phi})^2). \end{aligned} \tag{B.1}$$

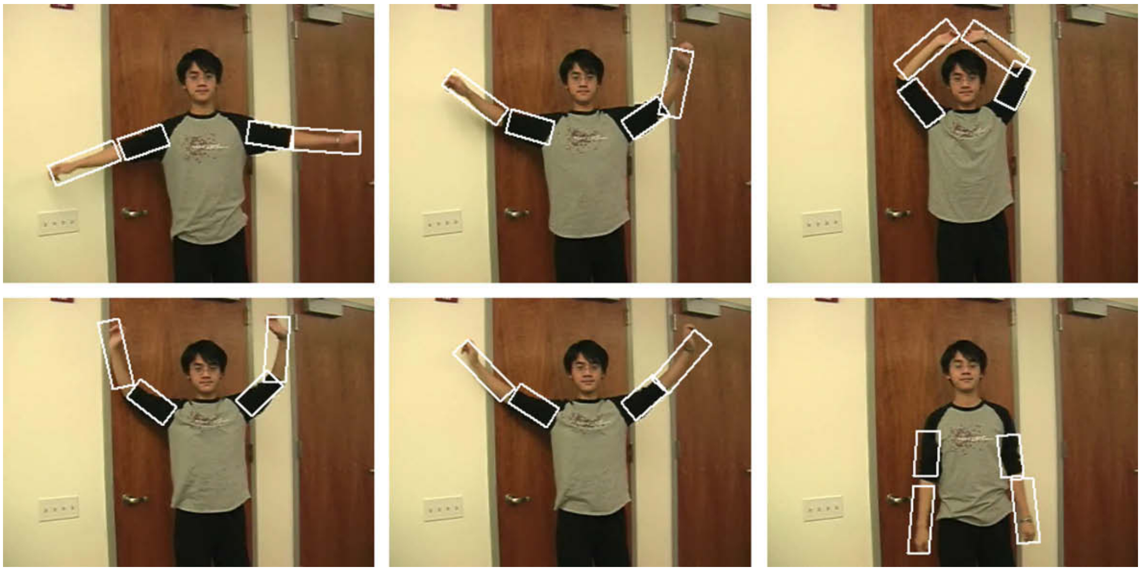

**Fig. 14.** *Stretching* sequence: the SCC based tracker tracks multiple human parts reliably.

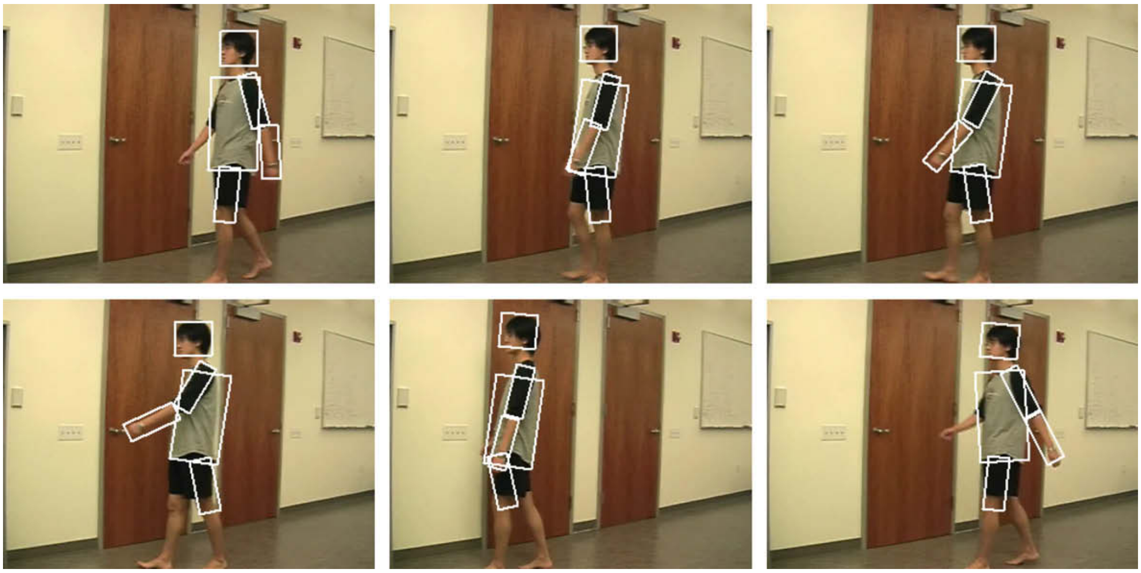

**Fig. 15.** *Walking* sequence.

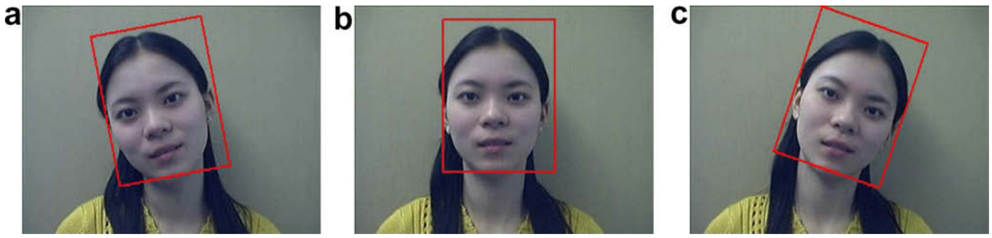

**Fig. 16.** *Face* sequence: tracking results of the SCC based tracker are shown for heads (a) left tilted, (b) upright, and (c) right tilted.

Drop higher order terms, Eq. (B.1) becomes

$$\sqrt{\mathbf{C}(\mathbf{\Phi})} \approx \sqrt{\mathbf{C}(\mathbf{\Phi}_0)} + \frac{d\sqrt{\mathbf{C}(\mathbf{\Phi})}}{d\mathbf{\Phi}}|_{\mathbf{\Phi}=\mathbf{\Phi}_0}\Delta\mathbf{\Phi}. \tag{B.2}$$

Introduce $\mathbf{C}(\mathbf{\Phi}_0) = \mathbf{U}_C^T\mathbf{K}(\mathbf{\Phi}_0)$ and $\frac{d\mathbf{C}(\mathbf{\Phi})}{d\mathbf{\Phi}}|_{\mathbf{\Phi}=\mathbf{\Phi}_0} = \mathbf{U}_C^T\nabla\mathbf{K}(\mathbf{\Phi}_0)$ into the second term of the right hand side of Eq. (B.2), we have

$$\sqrt{\mathbf{C}(\mathbf{\Phi})} \approx \sqrt{\mathbf{C}(\mathbf{\Phi}_0)} + \frac{1}{2}diag(\mathbf{C}(\mathbf{\Phi}_0))^{-\frac{1}{2}}\mathbf{U}_C^T\nabla\mathbf{K}(\mathbf{\Phi}_0)\Delta\mathbf{\Phi}, \tag{B.3}$$

where $diag(\mathbf{C}(\mathbf{\Phi}_0))$ is the matrix with $\mathbf{C}(\mathbf{\Phi}_0)$ on its diagonal.

Now rewrite the objective function in terms of the motion vector $\Delta\mathbf{\Phi}$, we obtain

$$argmin_{\Delta\mathbf{\Phi}}\|\sqrt{\mathbf{M}}-\sqrt{\mathbf{C}(\mathbf{\Phi}_0+\Delta\mathbf{\Phi})}\| \tag{B.4}$$

Substitute Eq. (B.3) into Eq. (B.4), the resulting objective function is

$$argmin_{\Delta\mathbf{\Phi}}\|\sqrt{\mathbf{M}}-\sqrt{\mathbf{C}(\mathbf{\Phi}_0)}-\frac{1}{2}diag(\mathbf{C}(\mathbf{\Phi}_0))^{-\frac{1}{2}}\mathbf{U}_C^T\nabla\mathbf{K}(\mathbf{\Phi}_0)\Delta\mathbf{\Phi}\|, \tag{B.5}$$

the solution of which equates to the solution of the linear system

$$\frac{1}{2}diag(\mathbf{C}(\mathbf{\Phi}_0))^{-\frac{1}{2}}\mathbf{U}_C^T\nabla\mathbf{K}(\mathbf{\Phi}_0)\Delta\mathbf{\Phi}=\sqrt{\mathbf{M}}-\sqrt{\mathbf{C}(\mathbf{\Phi}_0)}. \tag{B.6}$$

Denoting $\nabla\mathbf{K}(\mathbf{\Phi}_0)$ as $\mathbf{J}_K(\mathbf{\Phi}_0)$ and scaling both sides up by a factor of 2 results

$$diag(\mathbf{C}(\mathbf{\Phi}_0))^{-\frac{1}{2}}\mathbf{U}_C^T\mathbf{J}_K(\mathbf{\Phi}_0)\Delta\mathbf{\Phi}=2(\sqrt{\mathbf{M}}-\sqrt{\mathbf{C}(\mathbf{\Phi}_0)}). \tag{B.7}$$

## References

[1] http://visualsurveillance.org/PETS2001/.

[2] D.P. Bertsekas, Nonlinear Programming, Athena Scientific, 1995.

[3] S. Birchfield, R. Sriram, Spatiograms versus histograms for region-based tracking, in: IEEE Conference on Computer Vision and Pattern Recognition, vol. II, 2005, pp. 1158–1163.

[4] G. Bradski, Computer vision face tracking for use in a perceptual user interface, in: IEEE Workshop on Applications of Computer Vision, 1998, pp. 214–219.

[5] J. Canny, A computational approach to edge detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 8 (6) (1986) 679–698. November.

[6] Y. Cheng, Mean shift, mode seeking, and clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence 17 (8) (1995) 790–799.

[7] R. Collins, Mean-shift blob tracking through scale space, in: IEEE Conference on Computer Vision and Pattern Recognition, 2003, pp. 234–240.

[8] D. Comaniciu, P. Meer, Mean shift: a robust approach toward feature space analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (5) (2002) 603–619. May.

[9] D. Comaniciu, V. Ramesh, P. Meer, Real-time tracking of non-rigid objects using mean shift, in: IEEE Conference on Computer Vision and Pattern Recognition, vol. II, 2000, pp. 142–149.

[10] D. Comaniciu, V. Ramesh, P. Meer, Kernel-based object tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (5) (2003) 564–577.

[11] Z. Fan, Y. Wu, Multiple collaborative kernel trackingm, in: IEEE Conference on Computer Vision and Pattern Recognition, vol. II, 2005, pp. 502–509.

[12] K. Fukunaga, L.D. Hostetler, The estimation of the gradient of a density function, with applications in pattern recognition, IEEE Transactions on Information Theory 21 (1975) 32–40.

[13] B. Funt, G. Finlayson, Color constant color indexing, IEEE Transactions on Pattern Analysis and Machine Intelligence 17 (5) (1995) 522–529. May.

[14] G. Hager, P. Belhumeur, Efficient region tracking with parametric models of geometry and illumination, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (10) (1998) 1025–1039.

[15] G. Hager, M. Dewan, C. Stewart, Multiple kernel tracking with SSD, in: IEEE Conference on Computer Vision and Pattern Recognition, vol. I, 2004, pp. 790–797.

[16] B. Han, L. Davis, On-line density-based appearance modeling for object tracking, in: IEEE International Conference on Computer Vision, vol. II, 2005, pp. 1492–1499.

[17] J. Ho, K. Lee, M. Yang, D. Kriegman, Visual tracking using learned subspaces, in: IEEE Conference on Computer Vision and Pattern Recognition, vol. I, 2004, pp. 782–789.

[18] W. Hsu, T. Chua, H. Pung, An integrated color-spatial approach to content-based image retrieval, in: ACM Multimedia Conference, 1995, pp. 305–313.

[19] J. Huang, Color-spatial image indexing and applications, Ph.D. thesis, Cornell University, 1998.

[20] M. Isard, A. Blake, Condensation—conditional density propagation for visual tracking, International Journal of Computer Vision 29 (1) (1998) 5–28.

[21] T. Kailath, The divergence and Bhattacharyya distance measures in signal selection, IEEE Transactions on Communications Technology 15 (1) (1967) 52–60.

[22] I. Matthews, T. Ishikawa, S. Baker, The template update problem, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (6) (2004) 810–815.

[23] T. Ojala, M. Rautiainen, E. Matinmikko, M. Aittola, Semantic image retrieval with hsv correlograms, in: Scandinavian Conference on Image Analysis, 2001, pp. 621–627.

[24] A. Rao, R. Srihari, Z. Zhang, Spatial color histograms for content-based image retrieval, in: IEEE International Conference on Tools with Artificial Intelligence, 1999, pp. 183–186.

[25] A. Rao, R. Srihari, Z. Zhang, Geometric histogram: a distribution of geometric configurations of color subsets, in: SPIE: Internet Imaging, vol. 3964, 2000, pp. 91–101.

[26] D. Reid, An algorithm for tracking multiple targets, IEEE Transaction on Automatic Control 24 (6) (1979) 843–854.

[27] R. Rickman, J. Stonham, Content-based image retrieval using color tuple histograms, in: SPIE proceedings: Symposium on Electronic Imaging: Science and Technology-Storage and Retrieval for Image and Video Database, 1996, pp. 2–7.

[28] J. Shi, C. Tomasi, Good features to track, in: IEEE Conference on Computer Vision and Pattern Recognition, 1994, pp. 593–600.

[29] M. Swain, D. Ballard, Color indexing, International Journal of Computer Vision 7 (1) (1991) 11–32.

[30] C. Tang, G. Medioni, M. Lee, Epipolar geometry estimation by tensor voting in 8d, in: IEEE International Conference on Computer Vision, vol. I, 1999, pp. 502–509.

[31] H. Tao, H.S. Sawhney, R. Kumar, Object tracking with bayesian estimation of dynamic layer representations, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (1) (2002) 75–89.

[32] T. Yang, S. Li, Q. Pan, J. Li, Real-time multiple objects tracking with occlusion handling in dynamic scenes, in: IEEE Conference on Computer Vision and Pattern Recognition, vol. I, 2005, pp. 970–975.

[33] Q. Zhao, S. Brennan, H. Tao, Differential EMD tracking, in: IEEE Conference on Computer Vision.