Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

An advanced integrated framework for moving object tracking*

Gwang-Min CHOE[†1,2], Tian-jiang WANG[†‡1], Fang LIU[†1], Chun-Hwa CHOE[2], Hyo-Son SO[2], Chol-Ung PAK[3]

([1]*School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China*)

([2]*School of Computer Science and Technology, Kim Il Sung University, Pyongyang, DPR of Korea*)

([3]*School of Wireless Engineering, Huichon Institute of Technology, Huichon, DPR of Korea*)

[†]E-mail: cca2005@foxmail.com; tjwang@hust.edu.cn; fang.liu@hust.edu.cn

Received Jan. 5, 2014; Revision accepted Mar. 28, 2014; Crosschecked Aug. 11, 2014

Abstract: This paper first introduces the concept of a geogram that captures richer features to represent the objects. The spatiogram contains some moments upon the coordinates of the pixels corresponding to each bin, while the geogram contains information about the perimeter of grouped regions in addition to features in the spatiogram. Then we consider that a convergence process of mean shift is divided into obvious dynamic and steady states, and introduce a hybrid technique of feature description, to control the convergence process. Also, we propose a spline resampling to control the balance between computational cost and accuracy of particle filtering. Finally, we propose a boosting-refining approach, which is boosting the particles positioned in the ill-posed condition instead of eliminating the ill-posed particles, to refine the particles. It enables the estimation of the object state to obtain high accuracy. Experimental results show that our approach has promising discriminative capability in comparison with the state-of-the-art approaches.

Key words: Geogram, Mean shift, Hybrid gradient descent algorithm, Particle filter, Spline resampling, Matrix condition number

doi:10.1631/jzus.C1400006 **Document code:** A **CLC number:** TP391

1 Introduction

Visual tracking is a fundamental research area for video surveillance, video compression, 3D reconstruction, and other applications such as intelligent traffic navigation and human-computer interaction. Among the available visual tracking approaches, the association of particle filter (PF) (Isard and Blake, 1998) and kernel-based object tracking (KBOT) (Comaniciu *et al.*, 2003; Yilmaz *et al.*, 2006), has achieved considerable success over the last decade. Work on the association of PF and KBOT is summarized into four categories: switching, parallel, serial, and integrated.

First, a kind of tracking algorithm was proposed which combines KBOT and PF using the essentiality function (Han *et al.*, 2004; Wang and Liang, 2011). Under the condition without occlusion, KBOT is used to track the object and, when the object is occluded, PF is applied to accomplish the later object tracking. KBOT and PF are alternated by a given threshold.

Second, the algorithm was proposed where PF and KBOT trackers are run in a parallel way (Jia *et al.*, 2006). The two approaches discussed so far are convenient for implementation, but little attention has been paid to the combination of their advantages; thus, they do not perform well when both trackers are unreliable.

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (No. 61073094)

Third, approaches were proposed which belong to the serial category (Maggio and Cavallaro, 2005; Wang *et al.*, 2007; Bai and Liu, 2007; Liu *et al.*, 2008; Khan *et al.*, 2011), consisting of three stages: (1) The mean shift algorithm is employed to search for an object candidate near the target state; (2) If this candidate is good enough, it is used to adapt the particle filter parameters; (3) The particle filter estimates the target state based on these new parameters. Using these approaches, it is possible to find a very accurate solution under single mode visual tracking, but it is not under multi-mode tracking.

Fourth, algorithms were proposed which integrate the mean shift algorithm into the particle filter (Wang *et al.*, 2009; Yang *et al.*, 2009; Gao and Chen, 2011). The mean shift algorithm serves as an efficient gradient estimation and mode seeking procedure in the particle filter. Particles move towards the modes of posterior kernel density estimation. This approach belongs to the 'integrated' category.

The four types of frameworks discussed above are referred to as the switching, parallel, serial, and integrated frameworks, respectively. Among these frameworks, the integrated framework is the most widely used because of its fine association property. Recently, a compact association was proposed as an integrated framework (Yao *et al.*, 2012). This approach, however, is not very effective for several reasons: (1) The spatiogram based on the polar coordinates is used for feature description. In fact, the spatiogram itself is not very discriminative, especially when it is used to track an object with complex textures, and a spatiogram based on the polar coordinates is computationally too expensive. The spatiogram cannot distinguish two objects having the same color and the different distributions of coordinates. (2) A spatiogram based KBOT may easily undergo divergence, because the variation of the derivative function on the similarity surface is very serious. (3) PF needs a great number of particles for accurate estimation and has the possibility of tracking failure, because it is based on general resampling. To enhance the efficiency of particle filtering for a small number of particles, some methods have been established, such as systematic resampling (Arulampalam *et al.*, 2002; Wang and Lin, 2009). In practice, however, they are not ideally suited to real-time, highly accurate tracking. (4) A compact association is based on eliminating the ill-positioned particles. In fact, it decreases the number of particles that must take part in the convergence, and thus reduces the accuracy of tracking.

The main contributions of this paper can be summarized as follows:

1. Information about a perimeter of grouped regions is embedded into the spatiogram for better description of features in the manner of a union formula. Such a geogram has the capability to contain more accurate information about the distribution upon coordinates of pixels.

2. A mean shift procedure is derived based on the proposed geogram, and relationship between the convergence behavior of the mean shift procedure and the feature description is analyzed from the view of automatic control theory. Then a hybrid technique is used to control the convergence behavior of the mean shift procedure.

3. A spline resampling in the particle filter is proposed to deal with two issues of PF: computational cost and accuracy of particle filtering. We propose a resampling algorithm based on the spline transformation of weights to control the balance between these two issues.

4. The boosting-refining approach, in which the particles are refined after boosting the particles positioned in an ill-posed condition using optimal kernel placement (Fan *et al.*, 2006), is proposed to make each particle move towards a more accurate state.

2 Preliminaries

2.1 Particle filtering

PF is a state space approach for implementing the recursive Bayesian filter via sequential Monte Carlo (SMC) simulation. Let x_t denote the object state at time t, $Z_t = \{z_1, z_2, \cdots, z_t\}$ the observation sequence up to time t, $p(z_t|x_t)$ the observation likelihood function, and $p(x_t|x_{t-1})$ the state transition model. The visual tracking problem in the Bayesian filter is defined to model a dynamic system by recursively estimating the posterior probability distribution function (PDF):

$$p(x_t|Z_t) \propto p(z_t|x_t) \int p(x_t|x_{t-1}) p(x_{t-1}|Z_{t-1}) \mathrm{d}x_{t-1}. \tag{1}$$

In contrast to other approaches such as the Kalman filter and extended Kalman filter which provide the

solutions to problem (1) under their respective conditions, PF is designed to address problem (1) under more general situations where PDFs $p(z_t|x_t)$ and $p(x_t|x_{t-1})$ are usually nonlinear and non-Gaussian. The basic idea of PF is to offer a discrete approximation of PDF $p(x_{t-1}|Z_{t-1})$ by randomly sampling a set of N particles with states $\{x_{t-1}^i\}_{i=1}^N$ and importance weights $\{w_{t-1}^i\}_{i=1}^N$. By substituting PDF $p(x_{t-1}|Z_{t-1})$ with $\{x_{t-1}^i, w_{t-1}^i\}_{i=1}^N$, Eq. (1) can be expressed as

$$p(x_t|Z_t) \propto p(z_t|x_t)\sum_{i=1}^{N} w_{t-1}^i p(x_t^i|x_{t-1}^i). \quad (2)$$

According to Eq. (2), the state estimation problem can be iteratively solved via prediction and update steps. In practice, to estimate the object position at time t, a regular PF algorithm generally has four steps:

1. Propagating: According to the state transition model $p(x_t|x_{t-1} = \bar{x}_{t-1}^i)$, propagate each resampled particle state $\bar{x}_{t-1}^i$ to obtain a new state x_t^i for time t.

2. Weighting: Based on the state x_t^i and corresponding observation z_t^i, compute the weight w_t^i of each propagated particle at time t as $p(z_t^i|x_t^i)$ first, and then normalize it by

$$w_t^i = \frac{p(z_t^i|x_t^i)}{\sum_{j=1}^{N} p(z_t^j|x_t^j)}. \quad (3)$$

3. Re-sampling: From the particle set $\{x_{t-1}^i, w_{t-1}^i\}_{i=1}^N$ at time $t-1$, generate a new particle set $\{\bar{x}_{t-1}^i, \bar{w}_{t-1}^i = 1/N\}_{i=1}^N$ by repositioning particles.

4. Estimating: Calculate the object state at time t as

$$E(x_t) = \sum_{i=1}^{N} w_t^i x_t^i. \quad (4)$$

In the resampling step, the number of low-weighted particles is decreased and the number of particles with high weights is increased. In general, the resampling map is based on a linear function. It usually leads to incorrect tracking, because the estimation result may not trend towards the desired position, when weights of particles are similar to each other around the target place. This needs to make the estimate trend towards the target position using the similarity surface transformed nonlinearly. A spline transformation can be used to transform nonlinearly the similarity surface.

2.2 Bézier spline and properties

In general, the Bézier curve can be fitted by any number of control points. The number of control points determines the degree of the Bézier curve polynomial. The Bézier curve can be specified with blending functions. This Bézier polynomial function is represented by

$$B(u) = \sum_{i=0}^{n} P_i B_{i,n}(u), \quad 0 \le u \le 1, \quad (5)$$

where P_i are the control points of the Bézier spline and $B_{i,n}(u)$ are the Bernstein basis polynomials of degree n, expressed as

$$B_{i,n}(u) = \binom{n}{i} u^i (1-u)^{n-i}, \quad i = 0, 1, \cdots, n. \quad (6)$$

Note that $u^0 = 1$, $(1-u)^0 = 1$, and the binomial coefficient, also expressed as C_i^n, is

$$\binom{n}{i} = \frac{n!}{i!(n-i)!}. \quad (7)$$

The polygon formed by connecting the Bézier points with lines, starting with P_0 and finishing with P_n, is called the Bézier polygon (or control polygon). The convex hull of the Bézier polygon contains the Bézier curve.

The curve begins at P_0 and ends at P_n. This is the so-called endpoint interpolation property. The curve is a straight line if and only if all the control points are collinear. The start (end) of the curve is tangent to the first (last) section of the Bézier polygon. A curve can be split at any point into two subcurves, or into arbitrarily many subcurves, each of which is also a Bézier curve.

Every quadratic Bézier curve is also a cubic Bézier curve, and more generally, every degree n Bézier curve is also a degree m curve for any $m > n$. Detailedly, a degree n curve with control points $\{P_0, P_1, \cdots, P_n\}$ is equivalent (including the parametrization) to the degree $n+1$ curve with control points $\{P_0', P_1', \cdots, P_{n+1}'\}$, where $P_k' = \frac{k}{n+1}P_{k-1} + (1 - \frac{k}{n+1})P_k$.

3 The proposed method

3.1 Feature description based on the geogram

In histogram based representations, several patches with the same color feature are often treated

as a union connected region. This implies that histogram based representations are able to represent the global property for the distribution of the given feature, but are not appropriate to represent the local property of the distribution. To deal with this problem, the spatiogram has been proposed, but it is also not very discriminative because it is constructed only by the mean and covariance for coordinates. In this section, we give a definition of a geogram that is able to represent more complete spatial information about the domain of an image function and more detailed interpretation about its geometrical meaning.

3.1.1 Definition of the geogram

Given a discrete function $f\colon \boldsymbol{x} \to v$, e.g., an image function, where $\boldsymbol{x} \in \chi$ and $v \in \nu$, and $(i-1)$th-order functions $G_f^{(i)}(v) = \sum_{\boldsymbol{x}\in\chi} x^{i-1} S_f(x,v) P_f^{[\frac{1}{i+1}]}(x)$, we use the term geometric histogram, or geogram, distinguishable from the original spatiogram, to refer to a tuple of these $(i-1)$th-order functions. We define the kth-order geogram $G^k(v)$ to be a tuple of all the component functions up to order $k-1$ multiplied by the differential of a selected function f. The kth-order geogram is

$$G^k(v) = \langle G_f^{(0)}(v), G_f^{(1)}(v), \cdots, G_f^{(k)}(v)\rangle, \quad (8)$$

where

$$G_f^{(i)}(v) = \sum_{x\in\chi} x^{i-1} S_f(x,v) P_f^{[\frac{1}{i+1}]}(x), \quad (9)$$

$$S_f(x,v) = \begin{cases} 1, & f(x) = v, \\ 0, & \text{otherwise}, \end{cases} \quad (10)$$

$$P_f(x) = \begin{cases} x, & f'(x) \neq 0, \\ 0, & \text{otherwise}, \end{cases} \quad (11)$$

where $[\cdot]$ is the function that takes the integer part, given any real number. Also, to calculate $P_f^{[\frac{1}{i+1}]}(x)$, we introduce $0^0 \triangleq 1$. Then, the zeroth-order geogram is constructed by just the perimeters of a region where the given feature is distributed. Afterwards, the region where the given feature is distributed will be referred to as the homogeneous region. The geogram has the capability to represent all information from lower to higher levels of features as a tuple of all the elements up to order k. Its structure enables it to represent spatial information more completely than a spatiogram. The geograms are not computationally expensive and retain important information at lower levels, and can contain other information because of their structural property. To our knowledge, the geograms that embed information for the perimeter of the homogeneous region have not been explored.

3.1.2 Interpretation of the geogram

In practice, the third-order geogram is fairly popular and practical when considering even the compactness and the variation of coordinates. Given an image that is a 2D mapping $I\colon \boldsymbol{X} \to v$ from pixels $\boldsymbol{X} = [x,y]^{\mathrm{T}}$ to values v, the third-order geogram of an image is represented as

$$G^3(b) = \langle p_b, n_b, \boldsymbol{\mu}_b, \boldsymbol{\Sigma}_b\rangle, \quad b = 1,2,\cdots,B, \quad (12)$$

where p_b is the length of the perimeter captured from the homogeneous region, n_b is the number of pixels whose value is that of the bth bin (i.e., n_b is equivalent to a square of the homogeneous region), and $\boldsymbol{\mu}_b$ and $\boldsymbol{\Sigma}_b$ are the mean vector and covariance matrices calculated by the coordinates of pixels, respectively. In our formulation, $\boldsymbol{\Sigma}_b$ is supposed to be a diagonal matrix, considering that the covariance matrices are computationally expensive. $B = |\nu|$ is the number of bins in the geogram. Note that

$$G^0(b) = p_b, \quad b = 1,2,\cdots,B \quad (13)$$

is just the perimeter of a homogeneous region. The perimeter of the homogeneous region is able to play an important role in the description of features. In particular, for a feature descriptor that belongs to the signature category, this information is even more important. From the definition of the third-order geogram, it is clear that the third-order geogram is divided into two parts, one representing a geometrical property and the other representing a distribution property, as in Eq. (14), where the first two items represent the geometrical property of the homogeneous region, and the remaining two items represent the distribution of coordinates of all pixels in the homogeneous region.

$$G^3(b) = \langle \underbrace{p_b,\ n_b,}_{\text{Geometrical feature}} \underbrace{\boldsymbol{\mu}_b,\ \boldsymbol{\Sigma}_b}_{\text{Distribution feature}} \rangle. \quad (14)$$

Consider the distribution follows a Gaussian distribution of coordinates:

$$f_b(\boldsymbol{x}) = \frac{1}{2\pi||\boldsymbol{\Sigma}_b||^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_b)\boldsymbol{\Sigma}_b^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_b)^{\mathrm{T}}\right]. \tag{15}$$

Then, from Eq. (14), it is clear that the third-order geogram contains richer information for the geometrical property than the second-order spatiogram. In fact, the geometrical property of the homogeneous region can be more accurately represented by the square together with the perimeter, than by the single square. Moreover, the third-order geogram implicitly contains the topological property. That is, in addition to the geometrical property, it is possible to analyze the topological property from the first two items in the third-order geogram. Given the perimeter and the square of a region, a parameter, or compactness, which can summarize and represent two items in a synthetic manner, can be calculated directly and naturally:

$$R_b = \frac{4\pi n_b}{p_b^2}, \tag{16}$$

where p_b is the perimeter and n_b is the area of the given region. In fact, compactness is an example of point-set topology that establishes the foundational aspects of topology and investigates concepts inherent to topological spaces. From the above discussion it can be seen that geograms contain richer information of the geometrical property due to its definition. In practice, compactness is also able to represent the distribution property of the given feature at a competitive level of covariance, and sometimes demonstrates more improved performance. This means that it is possible to use lower-order geograms instead of the third-order ones.

$$G^2(b) = \langle p_b, n_b, \boldsymbol{\mu}_b \rangle, \quad b = 1, 2, \cdots, B. \tag{17}$$

From the definition of the second-order geogram, it is clear that using the second-order geogram decreases the amount of computation.

3.1.3 Similarity measurement for the geogram

Given geograms g and g', the similarity between two geograms can be calculated as

$$\rho(g, g') = \sum_{b=1}^{B} \psi_b \rho_n(n_b, n_b'), \tag{18}$$

where ψ_b is the weight, or the similarity belief between the matched bins n_b and n_b', and $\rho_n(n_b, n_b')$ is the similarity between the histogram bins.

$$\psi_b = \psi_b^{(1)} + \psi_b^{(2)}, \tag{19}$$

where $\psi_b^{(1)}$ and $\psi_b^{(2)}$ are the similarity beliefs corresponding to the geometrical feature and distribution feature, respectively:

$$\begin{cases} \psi_b^{(1)} = \dfrac{1}{1+\eta_b} \exp[-\beta(p_b - p_b')^2], \\ \psi_b^{(2)} = \dfrac{\eta_b}{1+\eta_b} \exp[-\dfrac{1}{2}(\boldsymbol{\mu}_b - \boldsymbol{\mu}_b')^{\mathrm{T}} \hat{\boldsymbol{\Sigma}}_b^{-1} (\boldsymbol{\mu}_b - \boldsymbol{\mu}_b')]. \end{cases} \tag{20}$$

Thus,

$$\begin{aligned} \psi_b = \frac{1}{1+\eta_b} \Big\{ & \exp[-\beta(p_b - p_b')^2] \\ & + \eta_b \exp[-\frac{1}{2}(\boldsymbol{\mu}_b - \boldsymbol{\mu}_b')^{\mathrm{T}} \hat{\boldsymbol{\Sigma}}_b^{-1} (\boldsymbol{\mu}_b - \boldsymbol{\mu}_b')] \Big\}, \end{aligned} \tag{21}$$

where β is the weight for the geometrical feature and η_b is the Gaussian normalization constant:

$$\eta_b = \frac{1}{2\pi||\hat{\boldsymbol{\Sigma}}_b||^{\frac{1}{2}}}, \quad \hat{\boldsymbol{\Sigma}}_b^{-1} = \boldsymbol{\Sigma}_b^{-1} + (\boldsymbol{\Sigma}_b')^{-1}. \tag{22}$$

In Eq. (19), $\psi_b^{(2)}$ can be calculated using several approaches proposed by, e.g., Jia *et al.* (2006) and Le *et al.* (2009), to increase the accuracy of the similarity measure. In this study, to emphasize the framework based on the geogram, the traditional approach proposed by Han *et al.* (2004) is used. The similarity between the histogram bins can be calculated using the Bhattacharyya coefficient:

$$\rho_n(n_b, n_b') = \frac{\sqrt{n_b n_b'}}{\sqrt{(\sum_{j=1}^{B} n_j)(\sum_{j=1}^{B} n_j')}}. \tag{23}$$

In Eq. (21) the spatiogram based distribution property and the perimeter based geometrical property are both considered. For the second-order geogram, $\boldsymbol{\Sigma}_b$ is supposed to be an identity matrix. For the first-order geogram, $\boldsymbol{\mu}_b$ is supposed to be a zero vector.

Fig. 1 shows some example results to compare the geogram and the spatiogram. Fig. 1a shows an image which contains two objects neighbored mutually. One is in the green box, and the other is the chess board outside the box. In Fig. 1a, the green box corresponds to the target region, and the red box indicates the initialization position to scan

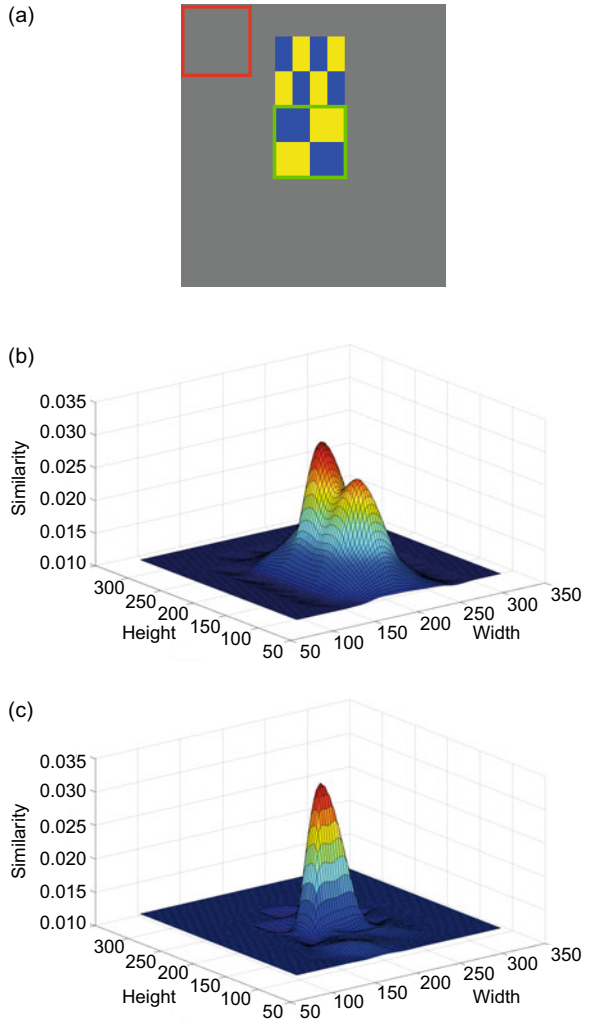

Fig. 1 Comparison of similarities for the geogram and the spatiogram: (a) the object in the green box and the other object outside the green box but within the chess board; (b) spatiogram; (c) geogram. References to color refer to the online version of this figure

the entire image from left to right and from top to bottom. The results correspond to the second-order spatiogram and the second-order geogram with the identity covariance matrix. It can be seen that the peaks of similarity surfaces for the two objects are similar to each other in the spatiogram, but completely distinguishable in the geogram. In fact, for the two objects in Fig. 1a, the spatiograms are equal to each other, because the means and covariances upon the coordinates of pixels are equal, respectively. However, the geograms are not equal, because the perimeters of the homogeneous regions contained in the two objects are different. This means geograms have better discriminative capability in comparison with spatiograms.

3.2 Hybrid control based mean shift optimization

3.2.1 Geogram based mean shift optimization

The mean shift algorithm is essentially a gradient descent algorithm. This algorithm has the capability of real-time and robust calculating and is easy to realize (Liu *et al.*, 2008). As a kernel-based technique, it requires that the geogram be smoothed with the profile $k:[0,\infty)\longrightarrow\mathbb{R}$ of a suitable kernel; i.e., n_b and n'_b are redefined as follows:

$$\begin{cases} n'_b = C\sum\limits_{i=1}^{N} k(\|\boldsymbol{x}_i\|^2)\delta_{ib}, \\ n_b(\boldsymbol{y}) = C_h\sum\limits_{i=1}^{N_h} k(\|(\boldsymbol{y}-\boldsymbol{x}_i)/h\|^2)\delta_{ib}, \end{cases} \tag{24}$$

where N is the number of pixels in the model region, $\boldsymbol{h}$ is the bandwidth vector of $k(x)$, $N_{\boldsymbol{h}}$ is the number of pixels in the region of size $\boldsymbol{h}$, and δ_{ib} is 1 if the value of $\boldsymbol{x}_i$ is that of the bth bin and 0 otherwise. The kernel function is an Epanechnikov function:

$$k(x)=\begin{cases} \dfrac{1}{2}c_d^{-1}(d+2)(1-x), & |x|\le 1, \\ 0, & |x|>1, \end{cases} \tag{25}$$

where $x\in[0,\infty)$, c_d is the volume of the unit d-dimensional sphere, e.g., $d=2$ and $c_d=\pi$ for a 2D image space.

$$\begin{cases} C=\dfrac{1}{\sum_{i=1}^{N} k(\|\boldsymbol{x}_i\|^2)}, \\ C_h=\dfrac{1}{\sum_{i=1}^{N_h} k(\|(\boldsymbol{x}_i-\boldsymbol{y})/\boldsymbol{h}\|^2)}. \end{cases} \tag{26}$$

The geogram based mean shift is completed using three stages: (1) a Taylor expansion of the histogram $n(\boldsymbol{y}_0)$, mean vector $\boldsymbol{\mu}(\boldsymbol{y}_0)$, and perimeter vector $\boldsymbol{p}(\boldsymbol{y}_0)$ at the current location, (2) taking the derivative with respect to the position variable, and (3) solving the derivative equation for the variable. The mean and covariance of coordinates are calculated using the following equations:

$$\boldsymbol{\mu}_b(\boldsymbol{y})=\frac{1}{\sum_{j=1}^{N_h}\delta_{jb}}\sum_{i=1}^{N_h}(\boldsymbol{x}_i-\boldsymbol{y})\delta_{ib}, \tag{27}$$

$$\boldsymbol{\Sigma}_b(\boldsymbol{y})=\frac{1}{\sum_{j=1}^{N_h}\delta_{jb}}\sum_{i=1}^{N_h}(\boldsymbol{x}_i-\boldsymbol{\mu}_b(\boldsymbol{y}))^{\mathrm{T}}(\boldsymbol{x}_i-\boldsymbol{\mu}_b(\boldsymbol{y}))\delta_{ib}. \tag{28}$$

The similarity measure in Eq. (18) can be considered as a function of a position variable:

$$\begin{aligned}\rho(\boldsymbol{y}) &= \sum_{b=1}^{B}\psi_b(\boldsymbol{y})\rho_n(n_b, n_b') \\ &= \sum_{b=1}^{B}(\psi_b^{(1)}(\boldsymbol{y}) + \psi_b^{(2)}(\boldsymbol{y}))\rho_n(n_b, n_b').\end{aligned} \tag{29}$$

A linear approximation from a Taylor expansion of the current location is

$$\rho(\boldsymbol{y}) \approx \rho(\boldsymbol{y}_0) + \Gamma_n(\boldsymbol{y};\boldsymbol{y}_0) + \Gamma_{\boldsymbol{\mu}}(\boldsymbol{y};\boldsymbol{y}_0) + \Gamma_{\boldsymbol{p}}(\boldsymbol{y};\boldsymbol{y}_0), \tag{30}$$

where

$$\begin{aligned}\Gamma_n(\boldsymbol{y};\boldsymbol{y}_0) &= \big(\boldsymbol{n}(\boldsymbol{y}) - \boldsymbol{n}(\boldsymbol{y}_0)\big)^{\mathrm{T}}\frac{\partial\rho(\boldsymbol{y}_0)}{\partial\boldsymbol{n}} \\ &= \frac{1}{2}\sum_{b=1}^{B}\psi_b(\boldsymbol{y}_0)\sqrt{\frac{n_b'}{n_b(\boldsymbol{y}_0)}}n_b(\boldsymbol{y}) - \frac{1}{2}\rho(\boldsymbol{y}_0),\end{aligned} \tag{31}$$

$$\begin{aligned}\Gamma_{\boldsymbol{\mu}}(\boldsymbol{y};\boldsymbol{y}_0) &= \big(\boldsymbol{\mu}(\boldsymbol{y}) - \boldsymbol{\mu}(\boldsymbol{y}_0)\big)^{\mathrm{T}}\frac{\partial\rho(\boldsymbol{y}_0)}{\partial\boldsymbol{\mu}} \\ &= \sum_{b=1}^{B}\psi_b^{(2)}(\boldsymbol{y}_0)\sqrt{n_b'n_b(\boldsymbol{y}_0)} \\ &\cdot(\boldsymbol{\mu}_b' - \boldsymbol{\mu}_b(\boldsymbol{y}_0))^{\mathrm{T}}\hat{\boldsymbol{\Sigma}}_b^{-1}(\boldsymbol{y}_0)(\boldsymbol{\mu}_b(\boldsymbol{y}) - \boldsymbol{\mu}_b(\boldsymbol{y}_0)),\end{aligned} \tag{32}$$

$$\begin{aligned}\Gamma_{\boldsymbol{p}}(\boldsymbol{y};\boldsymbol{y}_0) &= \big(\boldsymbol{p}(\boldsymbol{y}) - \boldsymbol{p}(\boldsymbol{y}_0)\big)^{\mathrm{T}}\frac{\partial\rho(\boldsymbol{y}_0)}{\partial\boldsymbol{p}} \\ &= -2\beta\sum_{b=1}^{B}\psi_b^{(1)}(\boldsymbol{y}_0)\sqrt{n_b'n_b(\boldsymbol{y}_0)} \\ &\cdot(\boldsymbol{p}_b(\boldsymbol{y}_0) - \boldsymbol{p}_b')(\boldsymbol{p}_b(\boldsymbol{y}) - \boldsymbol{p}_b(\boldsymbol{y}_0)).\end{aligned} \tag{33}$$

The derivative with respect to position variable $\boldsymbol{y}$ is

$$\begin{aligned}\frac{\partial\Gamma_n}{\partial\boldsymbol{y}} &= -\frac{C_h}{h^2}\sum_{b=1}^{B}\psi_b(\boldsymbol{y}_0)\sqrt{\frac{n_b'}{n_b(\boldsymbol{y}_0)}}\sum_{i=1}^{N_h}k(\cdot)\delta_{ib}(\boldsymbol{y} - \boldsymbol{x}_i) \\ &= -\sum_{i=1}^{N_h}\alpha_i k(\|\frac{\boldsymbol{y}_0 - \boldsymbol{x}_i}{h}\|^2)(\boldsymbol{y} - \boldsymbol{x}_i),\end{aligned} \tag{34}$$

$$\begin{aligned}\frac{\partial\Gamma_{\boldsymbol{\mu}}}{\partial\boldsymbol{y}} = -\sum_{b=1}^{B}\psi_b^{(2)}(\boldsymbol{y}_0)\sqrt{n_b'n_b(\boldsymbol{y}_0)}\hat{\boldsymbol{\Sigma}}_b^{-1}(\boldsymbol{y}_0) \\ \cdot(\boldsymbol{\mu}_b' - \boldsymbol{\mu}_b(\boldsymbol{y}_0)),\end{aligned} \tag{35}$$

$$\begin{aligned}\frac{\partial\Gamma_{\boldsymbol{p}}}{\partial\boldsymbol{y}} = -2\beta\sum_{b=1}^{B}\Big[\psi_b^{(1)}(\boldsymbol{y}_0)\sqrt{n_b'n_b(\boldsymbol{y}_0)} \\ \cdot(\boldsymbol{p}_b(\boldsymbol{y}_0) - \boldsymbol{p}_b')\frac{\partial\boldsymbol{p}_b(\boldsymbol{y})}{\partial\boldsymbol{y}}\Big],\end{aligned} \tag{36}$$

where

$$\begin{aligned}\alpha_i &= \frac{C_h}{h^2}\sum_{b=1}^{B}\psi_b(\boldsymbol{y}_0)\sqrt{\frac{n_b'}{n_b(\boldsymbol{y}_0)}}\delta_{ib} \\ &= \sum_{b=1}^{B}(\psi_b^{(1)}(\boldsymbol{y}_0) + \psi_b^{(2)}(\boldsymbol{y}_0))\sqrt{\frac{n_b'}{n_b(\boldsymbol{y}_0)}}\delta_{ib}.\end{aligned} \tag{37}$$

The solution of derivative equation $\frac{\partial\rho}{\partial\boldsymbol{y}} = 0$ of the similarity function ρ for the position variable is

$$\boldsymbol{y}_1 = \frac{\sum_{i=1}^{N_h}\boldsymbol{x}_i\alpha_i g(\|\frac{\boldsymbol{y}_0 - \boldsymbol{x}_i}{h}\|^2) - \sum_{b=1}^{B}\boldsymbol{\nu}_b - \sum_{b=1}^{B}\boldsymbol{s}_b}{\sum_{i=1}^{N_h}\alpha_i g(\|\frac{\boldsymbol{y}_0 - \boldsymbol{x}_i}{h}\|^2)}, \tag{38}$$

where

$$\begin{aligned}\boldsymbol{\nu}_b &= \psi_b^{(2)}(\boldsymbol{y}_0)\sqrt{n_b'n_b(\boldsymbol{y}_0)}\hat{\boldsymbol{\Sigma}}_b^{-1}(\boldsymbol{\mu}_b' - \boldsymbol{\mu}_b(\boldsymbol{y}_0)) \\ &= \frac{\eta_b}{1+\eta_b}\exp[-\frac{1}{2}(\boldsymbol{\mu}_b(\boldsymbol{y}_0) - \boldsymbol{\mu}_b')^{\mathrm{T}}\hat{\boldsymbol{\Sigma}}_b^{-1}(\boldsymbol{\mu}_b(\boldsymbol{y}_0) - \boldsymbol{\mu}_b')] \\ &\cdot\hat{\boldsymbol{\Sigma}}_b^{-1}(\boldsymbol{\mu}_b' - \boldsymbol{\mu}_b(\boldsymbol{y}_0)),\end{aligned} \tag{39}$$

$$\begin{aligned}\boldsymbol{s}_b &= \psi_b^{(1)}(\boldsymbol{y}_0)\sqrt{n_b'n_b(\boldsymbol{y}_0)}(\boldsymbol{p}_b(\boldsymbol{y}_0) - \boldsymbol{p}_b')\frac{\partial\boldsymbol{p}_b(\boldsymbol{y}_0)}{\partial\boldsymbol{y}} \\ &= \frac{2\beta}{1+\eta_b}\exp[-\beta(\boldsymbol{p}_b(\boldsymbol{y}_0) - \boldsymbol{p}_b')^2] \\ &\cdot(\boldsymbol{p}_b(\boldsymbol{y}_0) - \boldsymbol{p}_b')\frac{\partial\boldsymbol{p}_b(\boldsymbol{y}_0)}{\partial\boldsymbol{y}},\end{aligned} \tag{40}$$

and

$$\frac{\partial\boldsymbol{p}_b(\boldsymbol{y}_i)}{\partial\boldsymbol{y}} = \frac{\boldsymbol{p}_b(\boldsymbol{y}_i) - \boldsymbol{p}_b(\boldsymbol{y}_{i-1})}{||\boldsymbol{y}_i - \boldsymbol{y}_{i-1}||}, \quad \frac{\partial\boldsymbol{p}_b(\boldsymbol{y}_0)}{\partial\boldsymbol{y}} = 1. \tag{41}$$

If the Epanechnikov kernel profile is used, the derivative of the kernel is constant and disappears:

$$\boldsymbol{y}_1 = \frac{\sum_{i=1}^{N_h}\alpha_i\boldsymbol{x}_i - \sum_{b=1}^{B}\boldsymbol{\nu}_b - \sum_{b=1}^{B}\boldsymbol{s}_b}{\sum_{i=1}^{N_h}\alpha_i}. \tag{42}$$

3.2.2 Hybrid geogram-histogram based gradient descent

From Fig. 1, it can be observed that the spatiograms obtain very similar scores for many parts of

the region, while similarity for our geogram can better discriminate the points that are close to the target center from those relatively far from it. The gradient descent algorithm for the geogram, spatiogram, and histogram is just implemented on such similarity surfaces. The gradient descent algorithm for the histogram is able to steadily converge to the nearby target center; however, as it has very similar scores to the nearby target center, it cannot easily seek the accurate target center. For the geogram, if the initial position is very far from a target center, it is difficult to converge to a nearby target center; in contrast, if the initial position is very near to a target center, it is possible to find the accurate target center. From the view of automatic control, a convergence process on similarity surfaces can be considered as a control process based on gradient. The temporal property of the control process has two parts, dynamic and steady states, through which all state variables converge to the given values. Then, the histogram has good behavior for a steady state, and bad behavior for a dynamic state. The geogram has good behavior for a dynamic state, and bad behavior for a steady state. To obtain a good control result, it is possible to introduce a hybrid technique; i.e., a dynamic state of convergence is controlled by a traditional histogram while a steady state is controlled by our geogram. The hybrid approach based on the geogram and the histogram follows Eq. (43) with a threshold:

$$\boldsymbol{y}_1=\begin{cases}\dfrac{\sum_{i=1}^{N_h}\boldsymbol{x}_i\alpha_i g(\|\frac{\boldsymbol{y}_0-\boldsymbol{x}_i}{h}\|^2)-\sum_{b=1}^{B}\boldsymbol{\nu}_b-\sum_{b=1}^{B}s_b}{\sum_{i=1}^{N_h}\alpha_i g(\|\frac{\boldsymbol{y}_0-\boldsymbol{x}_i}{h}\|^2)}, & ||\boldsymbol{y}_1-\boldsymbol{y}_0||\le T,\\ \dfrac{\sum_{i=1}^{N_h}\boldsymbol{x}_i w_i g(\|\frac{\boldsymbol{y}_0-\boldsymbol{x}_i}{h}\|^2)}{\sum_{i=1}^{N_h}w_i g(\|\frac{\boldsymbol{y}_0-\boldsymbol{x}_i}{h}\|^2)}, & ||\boldsymbol{y}_1-\boldsymbol{y}_0||>T,\end{cases} \tag{43}$$

where

$$w_i=\sum_{b=1}^{B}\sqrt{\frac{n_b'}{n_b(\boldsymbol{y}_0)}}\delta(f(\boldsymbol{x}_i),b),$$

and $\delta()$ is the Kronecker delta function. Fig. 2 shows a control technique for the entire convergence process. The blue arc indicates the histogram based similarity surface, and the red curve indicates the geogram based similarity surface. The convergence process starts at state $\boldsymbol{x}_0$. First, the gradient decrease for the histogram is implemented on the histogram based similarity surface from state $\boldsymbol{x}_0$ to state $\boldsymbol{x}_4$. At this time, if it is measured that the convergence based on the histogram comes into a no-sense region, the system is switched to the gradient descent algorithm based on the geogram, and implemented on the geogram based similarity surface, where the no-sense region means there is almost no variation in the convergence process. This gives the convergence process speed and makes the state move from $\boldsymbol{x}_4$ to $\boldsymbol{x}_5$, which is closer to the position of the target (Fig. 2b). It is clear that the hybrid approach based on geogram and histogram is able to make gradient based convergence more stable and accurate. For convenience, we use the term 'hybrid gradient descent' or 'hybrid GD' to refer to our proposed hybrid approach, and 'single gradient descent' or 'single GD' to refer to the gradient descent algorithm without the hybrid approach. Otherwise, because our proposed gradient descent is directly used in the mean shift algorithm, we also use 'hybrid mean shift' or 'hybrid MS' to refer to mean shift with hybrid GD, and 'single mean shift' or 'single MS' to refer to mean shift with single GD.

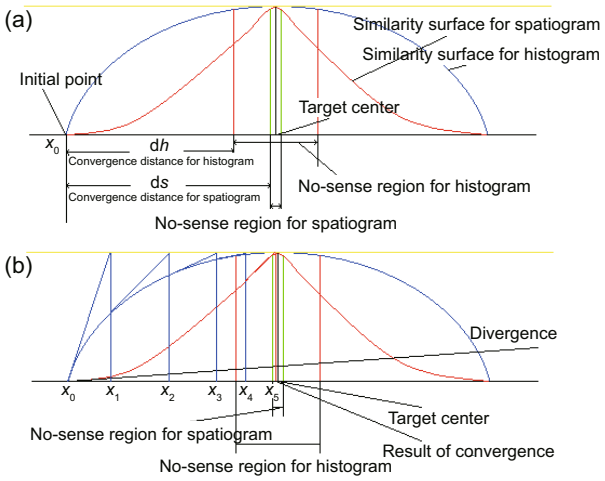

Fig. 2 Control of convergence using the hybrid technique for the geogram and the histogram (diagrammatic sketch): (a) similarity surfaces; (b) convergence process on the similarity surfaces. References to color refer to the online version of this figure

3.3 Particle filter based on spline resampling

In this section, we give a definition of spline resampling to deal with two issues of the particle filter, computational cost and accuracy. We incorporate a spline transformation function into the resampling algorithm to choose a few of the best particles with high weights by reducing the search area and also to increase the accuracy of particle filtering.

A particle filter can track multiple hypotheses simultaneously; each hypothesis is represented by a particle that has a weight corresponding to belief in the hypothesis. At time t, this set consists of N object states $x_t^1, x_t^2, \cdots, x_t^N$ and their associated weights $w_t^1, w_t^2, \cdots, w_t^N$. By the particle set, the posterior distribution of the real object state is approximated to a discrete set, given the observations up to time t, $p(x_t|z_t)$. The particles are resampled according to their weights to generate a new particle set. As a result of resampling, particles with large weights are replicated, and those with negligible weights are removed. Resampling maps the weighted random measure $\{x_t^i, w_t^i\}$ onto the equally weighted random measure $\{x_t^i, 1/N\}$ by sampling uniformly with replacement from the sample space according to the probabilities. In general, the resampling map is based on a linear function. It usually leads to incorrect tracking, because the estimation result may not trend towards the desired position, when weights of particles are similar to each other around the target place. In other words, when weights of particles are similar to each other around the target place, the steepness of the similarity surface is not very high, so the estimation results may trend towards an undesired position. To solve these problems, the estimate should trend towards the target position with the similarity surface being transformed nonlinearly. That is, we need to reduce the possibility of a tracking failure and enhance performance significantly, concentrating high-weight particles on the tracked object.

We introduce the resampling based on a nonlinear function instead of a linear function. Our goal is to obtain the best tracking output by concentrating high-weight particles on the tracked object and reducing the number of particles. Our proposed method uses a spline transformation of weights to obtain the best tracking output, even with fewer particles (Fig. 3). The spline transformation can also control the number of best-weighted particles, parameters of which are mainly application-dependent, as desired. At time t, the linear mapping function of previous resampling that maps the weights to the particles with those weights can be expressed as

$$R_t^i = w_t^i N, \quad i = 1, 2, \cdots, N, \tag{44}$$

where R_t is the size of the set of newly sorted particles after resampling by linear transformation, w is the relevant weight, and N is the number of particles. We can copy the more effective particles by discarding those associated with insignificant weights using a nonlinear transformation expressed as

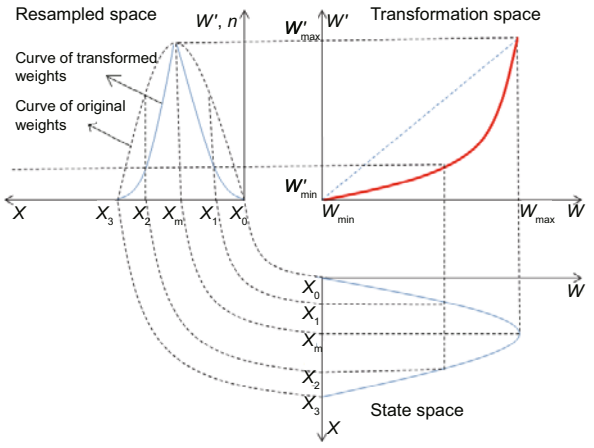

Fig. 3 Effect of the spline transformation of weights

$$S_t^i = N f(w_t^i), \tag{45}$$

where S_t is the size of the set of newly sorted particles after resampling by nonlinear transformation, and $f(\cdot)$ is a nonlinear transformation function of the following form:

$$f(w_t^i) = \frac{P_t(w_t^i)}{\sum_{i=1}^{N} P_t(w_t^i)}, \tag{46}$$

where $P_t(\cdot)$ is a parameter function for the nonlinear transformation of weights. In this study, the Bézier spline function is used to obtain a complete effect of nonlinear control of weights before resampling. The nonlinear transformation function $P_t(\cdot)$ is defined by the linear transformation of a given Bézier spline (the default spline):

$$\begin{cases} u = B_x^{-1}((w_t^i - m_x)/k), & 0 \le u \le 1, \\ P_t(u) = kB_y(u) + m_y, \end{cases} \tag{47}$$

where B is the default Bézier spline, and k and m are the scale and translation parameters, respectively. The default Bézier spline is defined as follows:

$$B(u) = \begin{pmatrix} B_x(u) \\ B_y(u) \end{pmatrix} = \sum_{i=0}^{n} \begin{pmatrix} P_{i,x} \\ P_{i,y} \end{pmatrix} B_{i,n}(u), \quad 0 \le u \le 1, \tag{48}$$

where $(P_{i,x}, P_{i,y})$ are the coordinates of control points $\boldsymbol{P}_i$ of the Bézier spline and $B_{i,n}(u)$ are the

Bernstein basis polynomials of degree n. In fact, from the property of the Bézier spline, this corresponds to an affine transformation of control points of the default spline:

$$P_t(u) = kB_y(u) + m_y = k\sum_{i=0}^{n} P_{i,y}B_{i,n}(u) + m_y, \tag{49}$$

$$\sum_{i=0}^{n} B_{i,n}(u) = 1, \tag{50}$$

$$\begin{aligned} P_t(u) &= k\sum_{i=0}^{n} P_{i,y}B_{i,n}(u) + m_y\sum_{i=0}^{n} B_{i,n}(u) \\ &= \sum_{i=0}^{n}(kP_{i,y} + m_y)B_{i,n}(u). \end{aligned} \tag{51}$$

Thus, with a different distribution of weights, the property of the transformation based on the default spline can be fairly maintained.

This nonlinear mapping enables concentration of high-weight particles on the tracked object and discarding of low-weight particles, and is better than linear mapping used in conventional resampling. The number of particles copied for resampling can be controlled by assignment of the control points. The control points of the default spline are determined by the demand of the real problems, and, in Fig. 4, they are expressed as

$$\begin{cases} \boldsymbol{P}_0 = (W_{\min}, W'_{\min})^{\mathrm{T}}, \\ \boldsymbol{P}_i = (W_x, W'_{\min})^{\mathrm{T}}, \\ \qquad i = 1, 2, \cdots, n_1 \ (n_1 = \lceil n/2 + 0.5 \rceil - 1), \\ \boldsymbol{P}_i = (W_{\max}, W'_y)^{\mathrm{T}}, \\ \qquad i = n_1 + 1, n_1 + 2, \ldots, n - 1, \\ \boldsymbol{P}_n = (W_{\max}, W'_{\max})^{\mathrm{T}}, \end{cases} \tag{52}$$

where $W_{\min}$, $W_{\max}$, $W'_{\min}$, $W'_{\max}$ represent the domain and the range of mapping, respectively, and W_x, W'_y are parameters that control the nonlinearity of the transformation. In this study, these parameters are set as follows: $W_{\min} = 0$, $W'_{\min} = 0$, $W_{\max} = 1$, $W'_{\max} = 1$, $W_x = 0.5$, $W'_y = 0.8$. Control point $\boldsymbol{P}_1$ is assigned on the w_t axis without transformation, and $\boldsymbol{P}_{n-1}$ on the w'_t axis with transformation. Also, we can find a more suitable assignment of the control points considering the overall property of a given tracking problem. In fact, assignment of control points is related to the probable distribution of

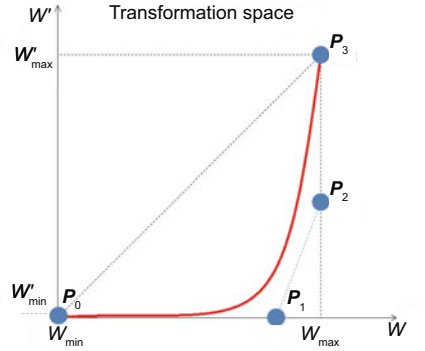

Fig. 4 Control points of the Bézier spline for the spline transformation of weights

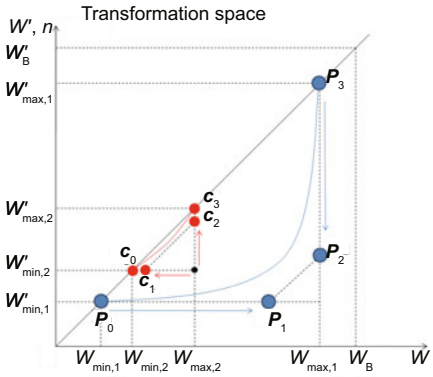

Fig. 5 Relationship between assignment of control points and variance of weights

weights obtained during tracking (Fig. 5). If the distribution of weights on the w_t axis has a large variance, then we need to discard low-weight particles as much as possible, i.e., to make the peak (the bending part of the curve) of the transformation curve move towards a region where high-weight particles are concentrated (Fig. 5). At this time, $\boldsymbol{P}_1$ must increase and $\boldsymbol{P}_{n-1}$ must decrease according to the convex property of the Bézier spline. Also, if the distribution of weights has a small covariance, we need to maintain the original state as much as possible, because it means that the peak of the transformation curve has already moved to a region where significant-weight particles are concentrated. At this time, $\boldsymbol{P}_1$ must decrease and $\boldsymbol{P}_{n-1}$ must increase. Given a set of weights w_t^i, scale k and translation m are expressed as

$$\begin{cases} k = \dfrac{w_{t,\max} - w_{t,\min}}{W_{\max} - W_{\min}}, \\ \boldsymbol{m} = (w_{t,\min},\ W'_{\min})^{\mathrm{T}}, \end{cases} \tag{53}$$

where $w_{t,\min}$ is the minimum weight at time t, $w_{t,\max}$ is the maximum weight, $w'_{t,\min}$ is the minimum transformed weight corresponding to $w_{t,\min}$ at time t, and

$w'_{t,\max}$ is the maximum transformed weight corresponding to $w_{t,\max}$. In practice, $w_{t,\min} = w'_{t,\min}$, $w_{t,\max} = w'_{t,\max}$.

It is easy to solve Eq. (47) for the above discussed spline transformation of weights. Before we start tracking, the default spline $B(u)$ must first be prepared. During the tracking we need only to implement the affine transformation for measured weights. This means that this approach can be directly used for real-time tracking.

Finally, transformed weights are normalized using Eq. (46). However, the problem of impoverishment of particle filtering still exists in this straightforward algorithm.

3.4 Association based on boosting-refining of particles

In the integrated framework of KBOT and PF, although the propagated particles are supposed to move towards modes with high probability in the state space through an iterative mode seeking procedure, it is not reliable enough to directly run KBOT on arbitrary particles. In this section we give a boosting-refining approach where all particles are refined after particles positioned in the ill-posed condition are boosted using the optimal kernel placement. From the view of KBOT, in the integrated framework the result of the propagating or re-sampling stage in the PF framework can be considered as initializations of KBOT trackers for refining the states of particles. In KBOT, it is often observed that different initializations of the tracker (i.e., the initializations that delineate the region to track and accordingly place the kernel) may largely influence the performance. In fact, for particles resulting in failure, either the initializations are not good, or they are positioned in an ill-posed condition. Whether particles are positioned in an ill-posed condition is evaluated by condition numbers of state matrices at their initialization positions. The state matrix of particles at the initialization positions is given by

$$\boldsymbol{M} = \begin{pmatrix} d_x^1 & d_y^1 \\ \vdots & \vdots \\ d_x^m & d_y^m \end{pmatrix}, \tag{54}$$

where $\boldsymbol{M}$ is the state matrix of the state equation $\boldsymbol{M}\Delta c = \sqrt{q} - \sqrt{p(c)}$ derived from $\rho(p(c), q) = \sum_{j=1}^{m} \sqrt{p_j(c)q_j}$ for KBOT, and

$$(d_x^j \;\; d_y^j) = \left(\frac{1}{2\sqrt{p_j}} \sum_{i, b(\boldsymbol{x}_i)=j} (\boldsymbol{x}_i^j - \boldsymbol{c}) g\Big(\| \frac{\boldsymbol{x}_i^j - \boldsymbol{c}}{h} \|^2 \Big) \right), \tag{55}$$

where $\{\boldsymbol{x}_i\}_{i=1,2,\cdots,n}$ are the pixel locations in the image, $b(\boldsymbol{x}_i)$ is a binning function that maps the color of $\boldsymbol{x}_i$ into a histogram bin j with $j \in \{1, 2, \cdots, m\}$, $\boldsymbol{c}$ is the position where the kernel function is spatially centered, and $g(\cdot) = -k(\cdot)$ with $k(\cdot)$ being the profile of the kernel function. The closed-form expression of the S-norm condition number is

$$\begin{aligned} k_s(\boldsymbol{M}^{\mathrm{T}}\boldsymbol{M}) &= \|(\boldsymbol{M}^{\mathrm{T}}\boldsymbol{M})\|_S \|(\boldsymbol{M}^{\mathrm{T}}\boldsymbol{M})^{-1}\|_S \\ &= \frac{\left(\sum (d_x^j)^2 + \sum (d_y^j)^2\right)^2}{\sum (d_x^j)^2 \sum (d_y^j)^2 - \left(\sum d_x^j d_y^j\right)^2}. \end{aligned} \tag{56}$$

That is, the condition number is the function of the center of the kernel function. As for KBOT, the particle with a smaller matrix condition number can more easily converge with a numerically stable solution to the state equations for KBOT. That is, the larger the condition number, the more probable the particle will be trapped in a problematic solution. In the integrated framework, if particles with large condition numbers are omitted in the refining stage, the number of particles used to estimate the accurate position is decreased. In an opposite manner, from the view of PF, in the integrated framework the greater the number of particles that take part in the convergence, the more accurate the result of tracking is likely to be. To increase the number of particles that can take part in the convergence, we boost the particles positioned in an ill-posed condition, i.e., move the particles positioned in an ill-posed condition towards positions with small matrix condition numbers using a gradient-based optimal kernel placement. In the results all particles in the integrated framework should not suffer from initializations positioned in an ill-posed condition. The gradient-based optimal kernel placement is implemented by the gradient descent algorithm, expressed as

$$\begin{cases} k_s(c) = 0, \\ c = c_0 - \eta k'_s(c_0), \end{cases} \tag{57}$$

where η is a factor for controlling the convergence, and calculating of $k'_s(c)$ follows the method used by Fan *et al.* (2006). Fig. 6 shows the boosting of the

particle positioned in an ill-posed condition and the process of convergence in a synthesized video. Fig. 7 shows the boosting of all particles positioned in an ill-posed condition and the process of convergence.

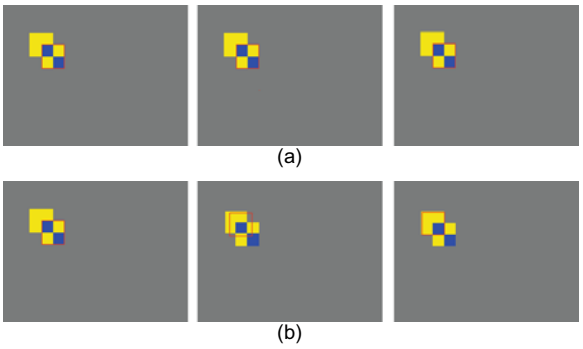

Fig. 6 The process of convergence without (a) or with (b) the boosting of a particle positioned in an ill-posed condition

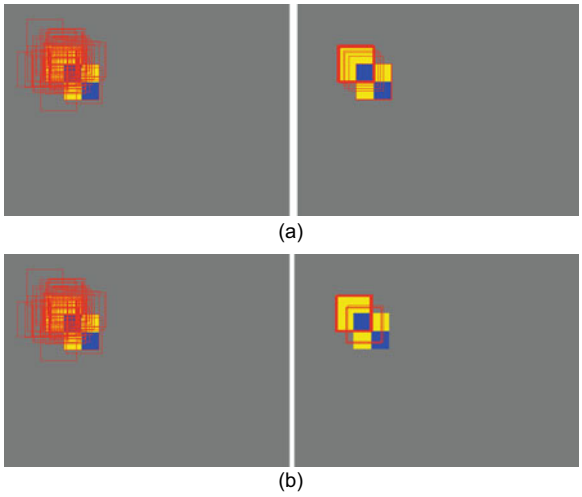

Fig. 7 The process of convergence without (a) or with (b) the boosting of all particles positioned in an ill-posed condition

4 Experimental results

We first evaluate the performance of our proposed feature descriptor and the hybrid control of convergence in a tracking scenario based on a traditional mean shift algorithm. The exhaustive search methods, of course, may be used for this tracking scenario to obtain high accuracy of tracking as in Wang and Liang (2011). However, our experiment should be aimed at the relative evaluation of the performances of our proposed approaches in relation with previous approaches. Therefore, in these experiments only the geogram, spatiogram, or histogram is used as the feature descriptor. Also, only the traditional mean shift algorithm is used as the tracking algorithm. We test our proposed approach and the previous approach in synthesized image sequences and real image sequences. The performance evaluation includes three parts. The first part contains the evaluation of the performance of our proposed feature descriptor and hybrid approach. The convergence in hybrid geogram-histogram based gradient descent is compared with that of spatiogram based gradient descent in synthesized image sequences. The second part contains experiment results for the above discussed hybrid approach and the spatiogram based approach in real image sequences. The third part compares the calculation time for the geogram based on Cartesian coordinates with that for the spatiogram based on polar coordinates. All experiments are implemented using the second-order spatiogram and the third-order geogram. Also, experiments are conducted on an Intel Core 2 GHz PC with 2 GB memory (1.32 GHz). The real image sequences are available at http://www.ces.clemson.edu/~stb/research/headtracker.

The first set of experiments is conducted on a video sequence containing 480×360-pixel synthesized color images. In this video, the two objects in Fig. 1a with the same distribution, or the same mean and covariance, are moving in the diagonal direction of the video region, maintained with a small gap. The object in the chess board of the two objects corresponds to the target to be tracked. The purpose of this experiment is to evaluate the performance of the tracker based on the geogram and the tracker based on the spatiogram. At this time, our tracker is associated with the hybrid approach proposed in this paper. The performance is evaluated in terms of tracking accuracy and the tracking error histogram. The errors are computed as the Euclidean distance in pixels from the true ground positions to the center of the candidate object in each frame. The true ground positions are functionally generated for synthesized image sequences. We use a tracker with $\beta = 1.0$, $T = 0.5h$ for our proposed tracker, where h is the size of the search region. Fig. 8 shows the comparison of the tracking result in a synthesized color video, the absolute error for every frame, and the error histogram for two approaches. An ellipti-

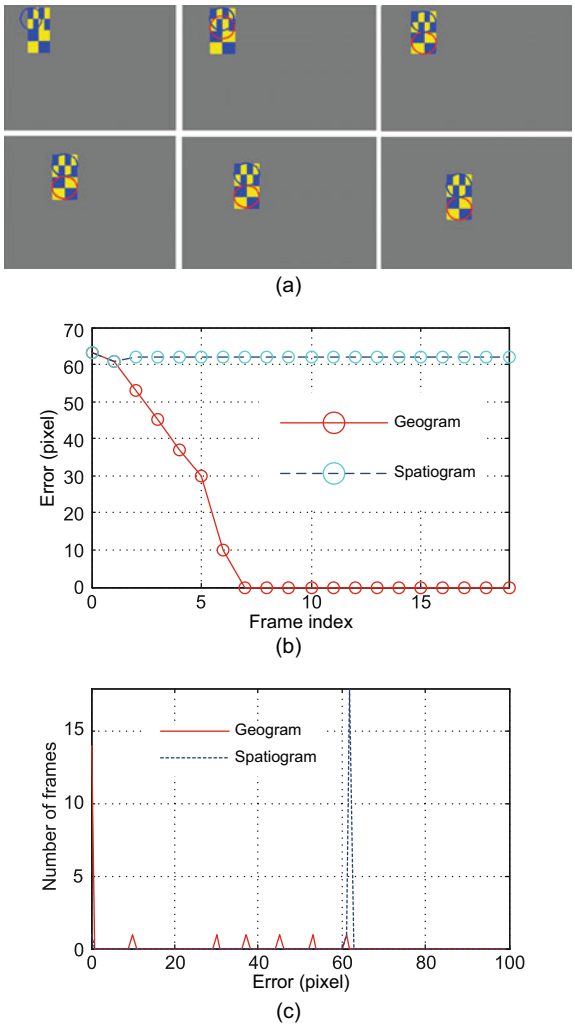

Fig. 8 Comparison of tracking results in a synthesized color video (a), errors (b), and error histograms (c). The red circle indicates the result of our approach, and blue for the spatiogram. The frames shown are frames 0, 2, 7 (top) and 9, 15, 17 (bottom). References to color refer to the online version of this figure

cal boundary box at the first frame is supposed to be the result of tracking in the previous frame. This initialization simulates time t and $t-1$ in tracking to evaluate the convergence of the hybrid geogram-histogram based gradient descent. It is clear that the best tracking performance in this synthesized video sequence is obtained using our proposed approach.

The second set of experiments is implemented in a popular real video sequence containing 500 and 30 frames, respectively, with a resolution of 128×96 pixels. These videos contain such variations as model distortion, occlusion, and appearance of multiple objects and noise. The purpose of this experiment is to evaluate the robustness of our proposed approach through comparison with an approach based on the spatiogram. Our tracker is implemented by the hybrid MS for the geogram containing the two approaches that we have proposed in this paper. The performance is evaluated in terms of tracking accuracy and the tracking error histogram. The errors are computed as the Euclidean distance in pixels from the true ground positions to the center of the candidate object in each frame. The true ground positions are manually labeled for real image sequences. We represent a target using an image region specified by its elliptical boundary box. The geogram and the spatiogram for this region in the first frame are used as appearance models of the tracked object, respectively. The tracker is set to $\beta = 1.0$, $T = 0.5h$. Figs. 9 and 10 show the comparison of tracking results in a real color video, the absolute error for every frame, and the error histograms for two approaches. The experiment results show that our proposed approach has robust performance for real video sequences. Specifically, in the fourth subfigure of Fig. 9a where the face is totally hidden by the hair, the geogram method still successfully tracks the target. In fact, in such a case, the spatiogram cannot capture any spatial information for the given target, while the geogram can still preserve the information for the perimeter of the hair boundary. Moreover, this result is determined by accumulative tracking results in previous frames.

In the third set of experiments, to estimate the performance of our proposed feature descriptor, the Cartesian coordinates based geogram and the polar coordinates based spatiogram (Yao *et al.*, 2010) are applied to the same video sequence as in the second set of experiments. Of course, the spatiogram can be used with Cartesian coordinates as well. However, as discussed above, this approach has a problem with the discriminative capability when attenuating the expensive calculation of the covariance matrix or using the lower-order spatiogram. This problem may be solved using the polar coordinates based spatiogram or the Cartesian coordinates based geogram; i.e., the discriminative capability of these two approaches is competitive with each other. The remaining problem is to compare their computational cost. In the polar coordinates based spatiogram, the mean vector and the covariance matrix of the spatiogram are calculated using the polar coordinates. The Cartesian coordinates system specifies

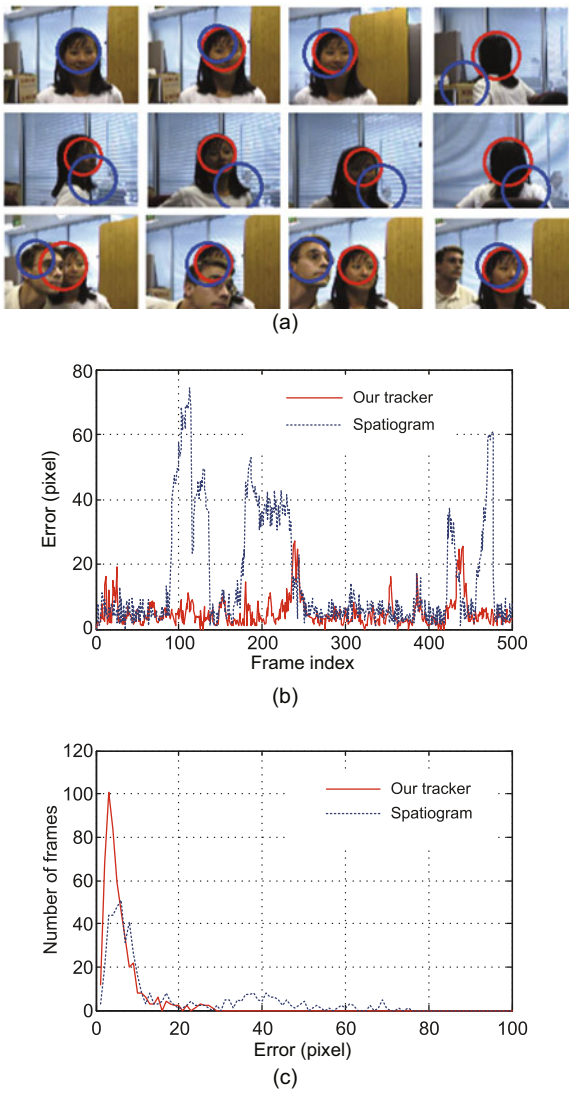

Fig. 9 Comparison of tracking results on a real color video (a), errors (b), and error histograms (c). The red circle indicates the result of our approach, and blue for the spatiogram. The frames shown are frames 0, 3, 22, 98 (top), 117, 126, 135, 188 (middle), and 427, 457, 471, 500 (bottom). References to color refer to the online version of this figure

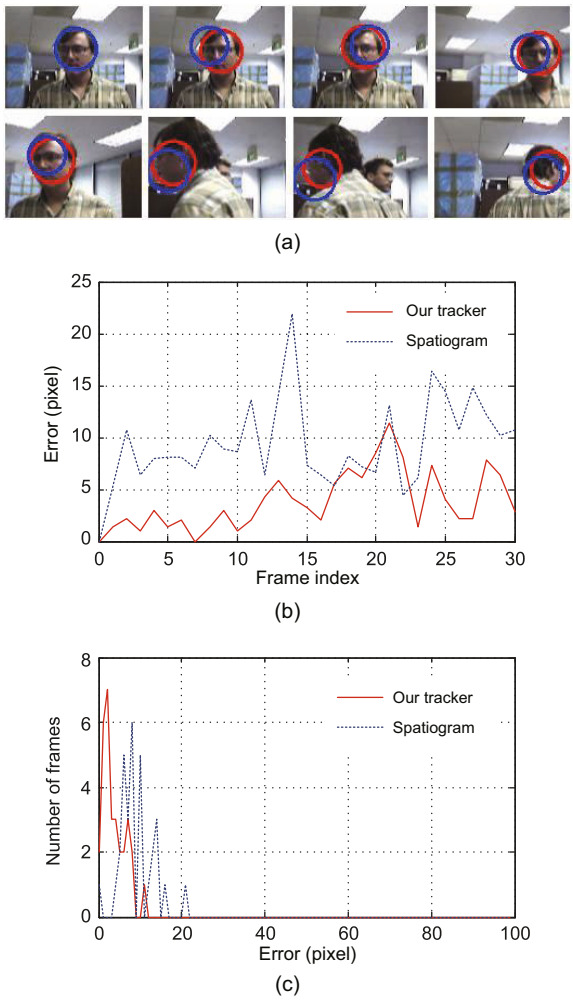

Fig. 10 Comparison of tracking results in a real color video (a), errors (b), and error histograms (c). The red circle indicates the result of our approach, and blue for the spatiogram. The frames shown are frames 0, 2, 5, 15 (top) and 18, 24, 25, 30 (bottom). References to color refer to the online version of this figure

each point uniquely in a plane by a pair of numerical coordinates, which are the signed distances from the point to two fixed perpendicular lines, measured in the same unit of length. The polar coordinates system is a 2D system in which each point on a plane is determined by a distance from a fixed point and an angle from a fixed direction. Therefore, calculating the polar coordinates based spatiogram needs several operations for the square, the square root, and the triangular function from Cartesian coordinates, while calculating the Cartesian coordinates based geogram can directly use Cartesian coordinates. There is no big difference in the tracking accuracy of the two approaches according to the experiment results. Thus, we consider only computational expense. The performance is estimated by the time needed for processing a frame. Table 1 shows that the Cartesian coordinates based geogram has lower calculation cost than the polar coordinates based spatiogram.

Then, we evaluate the effect of particle filtering based on the spline transformation of weights in our proposed spline resampling by comparison with SR. Approaches are tested by image sequences in a well-known database, AVATAR @ USC - Videos, which is captured by hand camera and contains a variation of the background itself. We represent an object using

Table 1 Comparison of the time needed for processing a frame for two approaches

Coordinates	Time (s/frame)	
	Spatiogram	Geogram
Cartesian	–	0.125
Polar	0.375	–

an image region specified by its elliptical boundary box. The traditional color histogram of the region in the first frame is used as an appearance model of the tracked object in our proposed and previous approaches. Also, to evaluate the result of visual tracking, we use the root mean square error (RMSE) of the state vector as the performance metric, which yields a combined measure of the bias and the variance of a filter estimate, to compare the performances of various algorithms. In our experiments, the state vector consists of coordinate components x, y and weight component w. We define the average RMSE over all sampling times as

$$\text{RMSE} = \frac{1}{K}\sum_{i=1}^{K}\sqrt{\frac{1}{T}\sum_{t=1}^{T}||\boldsymbol{x}_t^i - \boldsymbol{g}_t^i||^2}, \qquad (58)$$

where $\boldsymbol{g}_t^i$ and $\boldsymbol{x}_t^i$ are the true state vector and the estimate of the state vector at time step t in the ith Monte Carlo run, respectively. Also, T is the number of steps, and K is the number of iterations for the experiment.

Fig. 11 shows the results of tracking based on our approach and the traditional algorithm using 100 particles. This experiment is implemented 100 times under the same conditions to consider the probability of a particle filter. The video sequence corresponds to the 2nd landing (400 frames) AVATAR @ USC - Videos. In this sequence, a helicopter is flying to behind trees. Then, the helicopter is occluded temporarily by the head of a passing person. That is, our subject is completely occluded more than twice. Fig. 12 compares the performance of tracking based on our approach and that based on SR. The tracking algorithm based on our approach shows better performance. In our approach, high-weight particles are concentrated on the tracked object by spline transformation of weights in every frame. In contrast, in the previous approach the estimation result may not trend towards the desired position, because the resampling step is still based on a linear function.

Fig. 13 shows the results of tracking based on our approach and the traditional algorithm using 10 particles for the same image sequence. Frames are shown in order of top to bottom and left to right. Also, the performance of the proposed approach is compared with that of the SR-based particle filter (Fig. 14). Our approach works well with few particles, whereas the SR-based particle filter totally fails to track the object.

Finally, we evaluate the performance of our proposed approach, or the boosting-refining approach, by comparing the tracking accuracy and the tracking error histogram. For the previous approach, the experiment is implemented using AIBS, the boosting-refining algorithm, and the tracking algorithm discussed in Yao *et al.* (2012) for the integrated framework. For our proposed approach, our tracking algorithm is used instead of the tracking algorithm proposed by Yao *et al.* (2012) in the above experi-

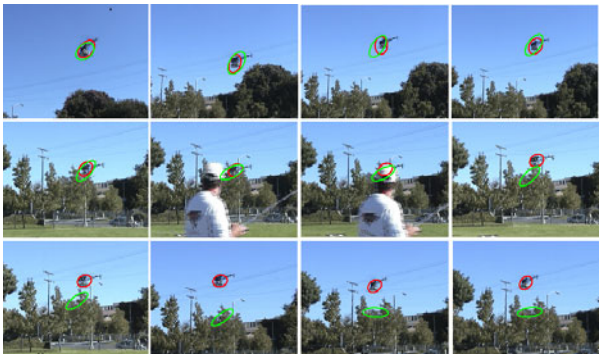

Fig. 11 Tracking in a real video with traditional and proposed resampling using 100 particles. The red ellipse indicates the result of our approach, and green for the previous approach. The frames shown are frames 1, 30, 90, 120 (top), 200, 230, 260, 281 (middle), and 283, 295, 300, 399 (bottom). References to color refer to the online version of this figure

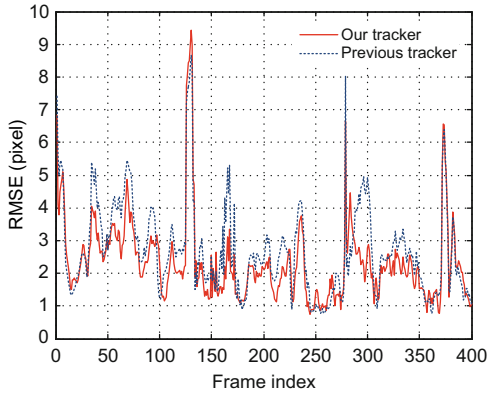

Fig. 12 Performance of the proposed method and the conventional SR filter

Fig. 13 Tracking in a real video with traditional and proposed resampling using 10 particles. The red ellipse indicates the result of our approach, and green for the previous approach. The frames shown are frames 1, 104, 165, 184 (top), 252, 280, 282, 298 (middle), and 305, 343, 386, 399 (bottom). References to color refer to the online version of this figure

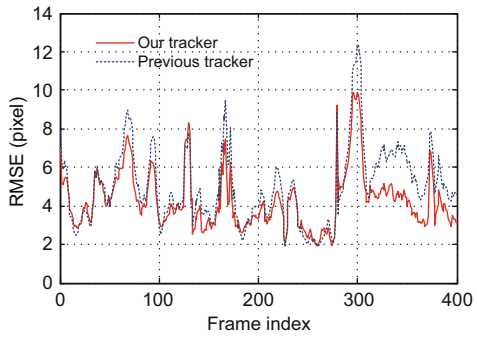

Fig. 14 Performance of the proposed method and the conventional SR filter

mental system. The experiment is done in a video sequence containing 128×96-pixel real color images. The tracker is set to $N = 50$, $\Delta x = 1.0$, $\Delta y = 1.0$, and $\Delta s = 0.01$. Fig. 15 shows the tracking results in a real color video, the absolute error for every frame, and the error histograms for the two approaches. Frames are shown in order of top to bottom and left to right. The previous approach has greater errors after the target undergoes some distortion and occlusion from moving. The experiment results show that our proposed approach, or the boosting-refining approach, has robust performance for real video sequences.

5 Conclusions

We have first presented a novel concept of geogram, which contains the geometric and distribution features and makes representation of spatial information for objects more attainable and accurate.

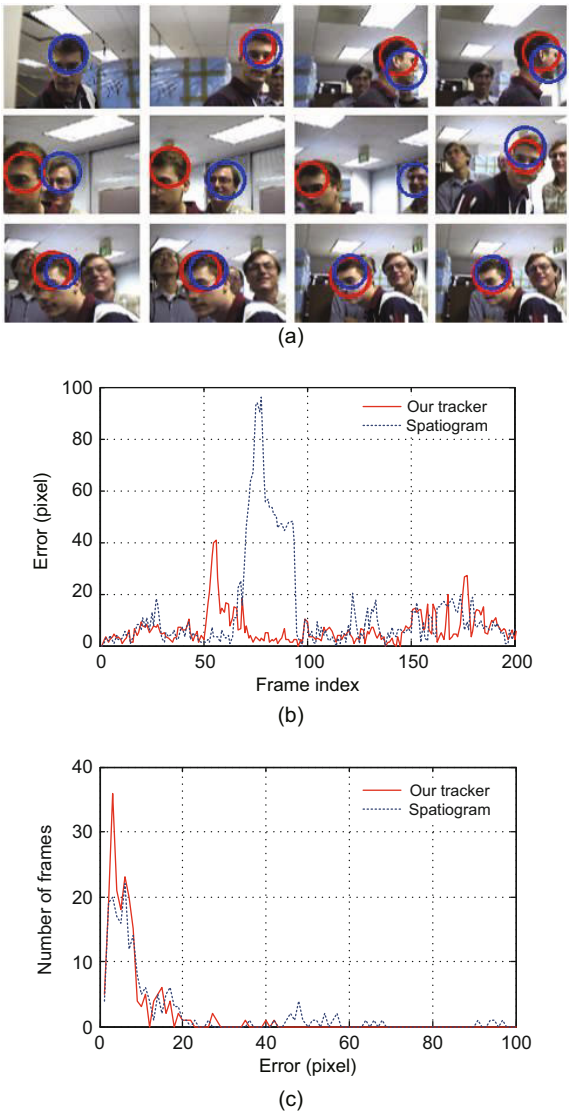

Fig. 15 Comparison of tracking results in a real color video (a), errors (b), and error histograms (c). The red circle indicates the result of the boosting-refining approach, and blue for eliminating ill-positioned particles. The frames shown in (a) are frames 0, 3, 22, 98 (top), 117, 126, 135, 188 (middle), and 427, 457, 471, 500 (bottom). References to color refer to the online version of this figure

Next, we analyze the convergence in the mean shift procedure and present the hybrid approach to control the convergence in the mean shift procedure for the geogram, motivated by the hybrid control technique of dynamic and steady states in the region for automatic control. Also, we derive the formula for the mean shift in the context of using the geogram. Then we introduce the concept of 'spline resampling' in a particle filter to obtain high accuracy in particle filtering and to reduce computational cost. Finally,

the boosting-refining approach makes the particles positioned in an ill-posed condition move towards positions with small matrix condition numbers using a gradient-based optimal kernel placement, so a more accurate estimation of the tracking result requires a greater number of particles. Future work should be aimed at extending the bin-by-bin geogram to the cross-bin geogram to enhance the performance of discriminating the object from the background and using more sophisticated particle filter algorithms.

References

Arulampalam, M.S., Maskell, S., Gordon, N., *et al.*, 2002. A tutorial on particle filter for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. Signal Process.*, **50**(2):174-188. [doi:10.1109/78.978374]

Bai, K., Liu, W., 2007. Improved object tracking with particle filter and mean shift. Proc. IEEE Int. Conf. on Automation and Logistics, p.431-435.

Comaniciu, D., Ramesh, V., Meer, P., 2003. Kernel based object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, **25**(5):564-577. [doi:10.1109/TPAMI.2003.1195991]

Fan, Z., Yang, M., Wu, Y., *et al.*, 2006. Efficient optimal kernel placement for reliable visual tracking. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, p.658-665. [doi:10.1109/CVPR.2006.109]

Gao, C., Chen, W., 2011. Ground moving target tracking with VS-IMM using mean shift unscented particle filter. *Chin. J. Aeronaut.*, **24**(5):622-630. [doi:10.1016/S1000-9361(11)60073-3]

Han, B., Comaniciu, D., Zhu, Y., *et al.*, 2004. Incremental density approximation and kernel-based Bayesian filtering for object tracking. Proc. IEEE Computer Vision and Pattern Recognition, p.638-644.

Isard, M., Blake, A., 1998. Condensation-conditional density propagation for visual tracking. *Int. J. Comput. Vis.*, **29**(1):5-28. [doi:10.1023/A:1008078328650]

Jia, J., Wang, Q., Chai, Y., *et al.*, 2006. Object tracking by multi-degrees of freedom mean shift procedure combined with the Kalman particle filter algorithm. Proc. IEEE Int. Conf. on Machine Learning and Cybernetics, p.3793-3797.

Khan, Z.H., Gu, I.Y.H., Backhouse, A.G., 2011. Robust visual object tracking using multi-mode anisotropic mean shift and particle filters. *IEEE Trans. Circ. Syst. Video Technol.*, **21**(1):74-87. [doi:10.1109/TCSVT.2011.2106253]

Le, P., Duong, A.D., Vu, H.Q., *et al.*, 2009. An adaptive mean shift particle filter for moving objects tracking. Int. Conf. on Adaptive Hybrid Mean Shift and Particle Filter, Computing and Communication Technologies, p.1-4.

Liu, H., Li, J., Qian, Y., *et al.*, 2008. Robust multi-target tracking using mean shift and particle filter with target model update. Proc. 3rd Int. Conf. on Computer Vision Theory and Applications, p.605-610.

Maggio, E., Cavallaro, A., 2005. Hybrid particle filter and mean shift tracker with adaptive transition model. Proc. IEEE Signal Processing Society Int. Conf. on Acoustics, Speech, and Signal Processing, p.221-224.

Wang, F., Lin, Y., 2009. Improving particle filter with a new sampling strategy. Proc. 4th Int. Conf. on Computer Science and Education, p.408-412.

Wang, H., Yang, B., Tian, G., *et al.*, 2009. Object tracking by applying mean-shift algorithm into particle filtering. 2nd IEEE Int. Conf. on Broadband Network & Multimedia Technology, p.550-554.

Wang, J., Liang, W., 2011. Robust tracking algorithm using mean-shift and particle filter. 4th Int. Conf. on Machine Vision: Computer Vision and Image Analysis; Pattern Recognition and Basic Technologies, p.1-5.

Wang, X., Zha, Y., Bi, D., 2007. An adaptive mean shift particle filter for moving objects tracking. *SPIE*, **6279**:1-7. [doi:10.1117/12.725431]

Yang, W., Hu, S., Li, J., *et al.*, 2009. Robust tracking in FLIR imagery by mean shift combined with particle filter algorithm. IEEE Int. Symp. on Knowledge Acquisition and Modeling Workshop, p.761-764.

Yao, A., Wang, G., Lin, X., *et al.*, 2010. An incremental Bhattacharyya dissimilarity measure for particle filtering. *Pattern Recogn.*, **43**(4):1244-1256. [doi:10.1016/j.patcog.2009.09.024]

Yao, A., Lin, X., Wang, G., *et al.*, 2012. A compact association of particle filtering and kernel based object tracking. *Pattern Recogn.*, **45**(7):2584-2597. [doi:10.1016/j.patcog.2012.01.016]

Yilmaz, A., Javed, O., Shah, M., 2006. Object tracking: a survey. *ACM Comput. Surv.*, **38**(4):13.1-13.45. [doi:10.1145/1177352.1177355]