OPTO-ELECTRONICS REVIEW **13**(4), 325–330

# An efficient object tracking method based on adaptive nonparametric approach

L. LI* and Z. FENG

Institute of System Engineering, Xi'an Jiaotong University, Xi'an, ShanXi, 710049 P.R. China

*In this paper, an efficient method for object tracking based on nonparametric approach is presented. The density we estimated is based on an adaptive kernel model, which is driven by the intensity difference between the target and the background. The background-weighted histogram for statistics of feature takes into account the relevance between the target and background. What is more, this approach extends the range that is needed for searching object. The target model is updated according to the change of the object and environment. Experimental results on real image sequences demonstrate its robust performance in visual tracking and require less iteration computations when compared to other method.*

**Keywords:** object tracking, adaptive, nonparametric, update, kernel.

## 1. Introduction

Object tracking is a central theme in computer vision with applications ranging from surveillance to human-computer interfaces, whose goal is finding and following moving objects between consecutive frames. A variety of algorithms have been developed for visual tracking, but they can be categorized into two main kinds of methodologies, i.e., top-down and bottom-up approaches [1]. Top-down approaches generate and evaluate a set of state hypotheses based on target model, while the performance of tracking is largely determined by the methods of evaluating and verifying these hypotheses on image observations. To achieve robust tracking, a large number of hypotheses may be maintained so that more computation would be involved for evaluating them. The condensation algorithm, also known as particle filters [2], is a typical top-down method for its robust performance in visual tracking, but it has two insoluble problems. Firstly, it is not real-time for the computation of large numbers of particles. Secondly, the observed likelihood function and motion model must be learned by some sample image sequences before tracking is performed, so it can only be used to video analysis, cannot be used to practical application such as spot surveillance, and obviously, it may fail when the environment is greatly changed. Bottom-up approaches generally tend to reconstruct the target states by analysing the image contents. It might be computationally efficient, yet the robustness largely depends on the ability of image analysis. To discriminate the target from other objects and describe the correlation between the appearance and the state of the object, the target representation is a fundamental problem. Tracking based on a rough target model would not be robust.

Many parametric statistical techniques have been applied to object representation, which describe the appearance of the object by statistics. Parametric approaches are based on the assumption of specific forms of the features in the images. Usually, image data are assumed to be normally distributed, and given known distributional forms. Thus, parametric methods restrict the form of the statistic to those for which distributional results are available, and rely on many assumptions and approximations. Furthermore, parametric method such as learning Gaussian distribution using EM [3] algorithm is very time-consuming, which is inappropriate to real-time tracking.

On the other hand, the nonparametric statistical methods do not necessarily depend on a presumed distribution model of the object, thus they are more widely applicable. This approach tends to let the data guide a search for the function which fits them best without the restrictions imposed by a parametric model [4]. Colour distribution histograms is a nonparametric approach which has many advantages for tracking non-rigid objects, as they are robust to partial occlusion, are rotation and scale invariant and are calculated efficiently [5,6]. Kernel density estimation is another kind of nonparametric technique, which has also been used as an important data analysis tool [7]. Taking the advantages of colour histograms and spatial kernel, Comaniciu [8,9] proposes a nonparametric tracking approach based on mean shift analysis (MST). It is deterministic and data driven for climbing density gradient to find the peak of probability distributions, and track a distribution by maximizing the Bhattacharyya measure between a model distribution and an empirical distribution. However, the classically used kernel cannot adapt to the changing of colour distribution histograms [10]. What is more, it is unable to track targets with large motion between two consecutive frames.

*e-mail: Longford@xjtu.edu.cn

We are interested in tracking non-rigid, complex objects with large motion under cluttered environments, for those cases when traditional techniques are not applicable. When tracking any kind of features in these cases, several specific problems appear. In particular, there are always difficult and ambiguous situations in real world generated by cluttered backgrounds, occlusions, large geometric deformations, illumination changes or noisy data [11].

In this paper, we present an efficient visual tracking algorithm (EVT), which integrates the advantages of the background-weighted histogram with adaptive kernel density estimation. An adaptive kernel is modelled according to the local histogram. In most cases, some of the target features may be blend with a part of background, thus the background information is also very important for the feature statistics. What is more, this approach extends the range that is needed for searching object. Comparative results with other method are also provided.

This paper is organized as follows. In Sec. 2, an adaptive kernel model is proposed for the kernel-based tracking. The background-weighted histogram and the target model update process are described in Sec. 3. In Sec. 4, the tracking algorithm on video streams is applied and its good performance is demonstrated.

## 2. Adaptive kernel model

The kernel or Parzen density estimation is one of the most well-known techniques for nonparametric density estimation. Given d-dimensional samples $X_1$, $X_1$,..., $X_n$ drawn from a population with density function $f(x)$, the Parzen density estimation at $x$ is given by

$$\hat{f}(x) = \frac{1}{nh^d}\sum_{i=1}^{n} k\left(\frac{x - X_i}{h}\right), \tag{1}$$

where $k(\cdot)$ is the kernel function and $h$ is the kernel bandwidth. Traditionally, it is assumed that $\int k(x)dx = 1$, $k(\cdot)$ is symmetric, i.e., $k(x) = k(-x)$.

Let $V$ be the set of pixels to be processed in the image, $x$ be the element in $V$ that is currently being processed, and $I(x)$ be the colour value of a pixel. Generally, a unit flat kernel is defined as

$$k(x) = \begin{cases} 1 & \text{if } \|x\| \le \eta \\ 0 & \text{otherwise} \end{cases}, \tag{2}$$

where $\eta$ is the radius of the neighbourhood in image space. In the same way, the unit Gaussian kernel can be defined as

$$k(x) = e^{-\|x\|^2}. \tag{3}$$

For practical applications, the Gaussian kernel is not used, since each pixel has the whole image as a neighbourhood. A mixture of a flat kernel and Gaussian kernel can be used, called a truncated Gaussian kernel

$$k(x) = \begin{cases} e^{-\phi\|x\|^2} & \text{if } \|x\| \le \eta \\ 0 & \text{otherwise} \end{cases}. \tag{4}$$

A problem with this kernel is that it treats everything the same, in that it will preserve all objects within the image if there are enough pixels in its neighbourhood with similar intensity. By modifying the kernel to adapt to different local histograms, we can achieve at

$$k(x) = \begin{cases} e^{-\phi(S)\|x\|^2} & \text{if } \|x\| \le \eta \\ 0 & \text{otherwise} \end{cases}, \tag{5}$$

where $S$ is the set of all the pixels in the neighbourhood of $x$, satisfying $S \subseteq V$, and the function $\phi(S)$ is defined as

$$\phi(S) = \Gamma \frac{\sum_{u \in U} \Psi(|I(u) - I(x)|)}{M}, \tag{6}$$

where $M$ is the number of elements in the set $S$, and $\Gamma$ is the constant. The function $\Psi(\cdot)$ satisfies

$$\psi(x) = \begin{cases} 1 & \text{if } \|x\| \le \zeta \\ 0 & \text{otherwise} \end{cases}, \tag{7}$$

where $\zeta$ is the given threshold. As $\psi(x)$ decreases, it likes a flat kernel. Flat kernel tends to smooth out small objects, a few pixels that are outliers but are still within $\eta$ will push the pixels toward the intensity of the majority of the pixels in the neighbourhood. A few pixels of noises in an image will have a low $\phi(S)$ value, because the intensity difference between them and their surrounding region is large. The target for tracking usually is composed of many pixels in a small area, so $\phi(S)$ will be higher since there are many pixels with similar intensity close together. The high $\phi(S)$ will result in a high standard deviation Gaussian kernel which doesn't allow the background pixels to influences the intensity of the target pixels too mach, since pixels farther away from the pixel intensity will be weighted much lower. It is obvious that a flat kernel tends to merge objects into the background, while a Gaussian kernel tends to preserve the targets better. Therefore, driven by the intensity difference between the target and the background, the truncated Gaussian kernel can be adaptively modelled.

## 3. Tracking based on background weighted histogram

In most applications, it is difficult to exactly describe the target, and its model might contain background as well. However, improper use of the background information may affect the tracking efficiency, make impossible to measure the similarity between the target and the new candidate. Hence, our approach is to use the background information for selecting only the striking parts from the representations of the target and candidate model.

 

### 3.1. Target model representation

Here, we consider a target chosen for tracking as a defined region of pixel locations $\{x_i\}_{i=1,\ldots,n}$ in an image. Let $\mu = 1,\ldots,m$ represent the colour feature bins in the target model, and the function $b(x_i,t)\colon R^2 \rightarrow \mu$ denotes the feature bins corresponding to the pixel at location $x_i$ with time index $t$. Let $\{g_u\}_{u=1,\ldots,m}$ represent the colour feature distribution of the background with the normalization constraint $\sum_{u=1}^{m} g_u = 1$. Assume the smallest nonzero value of $g_u$ is $g_s$ and the area of the background equals to two times the target area, thus the background factor can be obtained by

$$\left\{ d_u = \min\left( \frac{g_s}{g_u}, 1 \right) \right\}_{u=1,\ldots,m}. \tag{8}$$

These background factors decrease the importance of those features that have low $g_u$, especially in the background area. With these definitions, the probability distribution model of a kernel-weighted histogram in the target region can be computed as

$$q_u = C d_u \sum_{i=1}^{n} k(x_i - a^*)\delta[b(x_i,t) - u], \tag{9}$$

where $a^*$ is the centre location of the kernel, and $\delta$ is the Kronecker delta function. To impose the normalization constraint $\sum_{u-1}^{m} q_u = 1$, the constant $C$ is expressed as

$$C = \frac{1}{\sum_{i=1}^{n} k(x_i - a^*) \sum_{i=1}^{m} d_u \delta(b(x_i,t) - u)}. \tag{10}$$

### 3.2. Candidate model representation

A more compact form can be expressed for the probability distribution of the target model. For each feature vector $u$, we can combine these vectors into a matrix $U = [u_1, u_2, \ldots u_m]$. In the same way, we can define the kernel function $k(x_i - a^*)$ by $K(a^*)$, and define the background factor $\{d_u\}_{u=1,\ldots,m}$ by $D_u$. Depending upon these simplifications, we can rewrite the target model in a more concise form

$$q = K(a^*)U(t)D. \tag{11}$$

Assume we are now considering a candidate region centred at $a$ with subsequent time index $t'$. Thus the probability distribution model of a kernel-weighted histogram in the candidate region would be

$$p(a) = K(a)U(t')D. \tag{12}$$

### 3.3. Similarity measurement

The target tracking procedure is to find the most similar candidate in terms of the features we are interested. This process can be stated as follows, given the target model distribution $q$ and the candidate model distribution $p(a)$, finding out a location that maximizes the similarity between the target model distribution and the candidate model distribution. To measure the similarity between the probability density functions of two distributions, the Bhattacharyya coefficient is an efficient and divergence-type tool for statistical measurement. It can be solved by minimizing the distance between two discrete distributions

$$d(a) = \sqrt{1 - \rho(p(a),q)}, \tag{13}$$

where $\rho(p(a),q)$ is the similarity measurement. The equation above is equivalent to maximizing the sample estimation of the Bhattacharya coefficient

$$\rho(a) \equiv \rho(p(a),q) = \sum_{u=1}^{m} \sqrt{p_u(a) q_u}. \tag{14}$$

Applying the Taylor expansion around the values $p(a)$, the linear approximation of $\rho(a)$ can be rewritten as

$$\rho(a) \approx \frac{1}{2} \sum_{u=1}^{m} \sqrt{p_u(a) q_u} + \frac{1}{2} \sum_{u=1}^{m} p_u(a) \sqrt{\frac{q_u}{p_u(a)}}. \tag{15}$$

It is assumed that the target candidate $\{p_u(a)\}_{u=1,\ldots,m}$ does not change drastically between any two consecutive frames, so the equation above will always be feasible. Since $p(a)$ is independent of $a$, we can substitute Eq. (12) for $p_u(a)$, and $\rho(a)$ can be rewritten in a new form

$$\rho(a) \approx \frac{1}{2} \sum_{u=1}^{m} \sqrt{p_u(a) q_u} + \frac{C}{2} \sum_{i=1}^{n} w_i k(x_i - a^*), \tag{16}$$

where

$$w_i = \sum_{u=1}^{m} d_u \sqrt{\frac{q_u}{p_u(a)}} \delta[b(x_i,t) - u]. \tag{17}$$

By seeking the maximum mode of the density in the local region, the kernel can be recursively moved from the current location $a_0$ to the next location $a_1$ until achieving at the density mode according to the relation of

$$a_1 = \frac{\sum_{i=1}^{n} x_i w_i g(x_i - a_0)}{\sum_{i=1}^{n} w_i g(x_i - a_0)}, \tag{18}$$

where $g(x) = -k'(x)$. It is assumed that $k(x)$ is derivable for all $x \in [0,\infty]$ except a finite set of points. From the equation above, we can find that the new location is the weighted centroid of the sample points $\{x_i\}_{i=1,\ldots,n}$. However, the weights for these sample points are dependent on two parts. One part is the weight $w_i$ on background-factored colour features, and the other part is the weight from the kernel function $k(x)$ which assigns smaller weights to points farther away from the centre point $a_0$ of the target.

### 3.4. Update of target model

In most cases, the image of target feature is influenced by illumination changes, cluttered backgrounds and large geometric deformation, therefore it is very important to update the target model is for robust tracking. Let $\tau \in (0,1)$ be the update coefficient, then the new feature model $q^*$ for the target should be

$$q^* = (1 - \tau)q + \tau p. \tag{19}$$

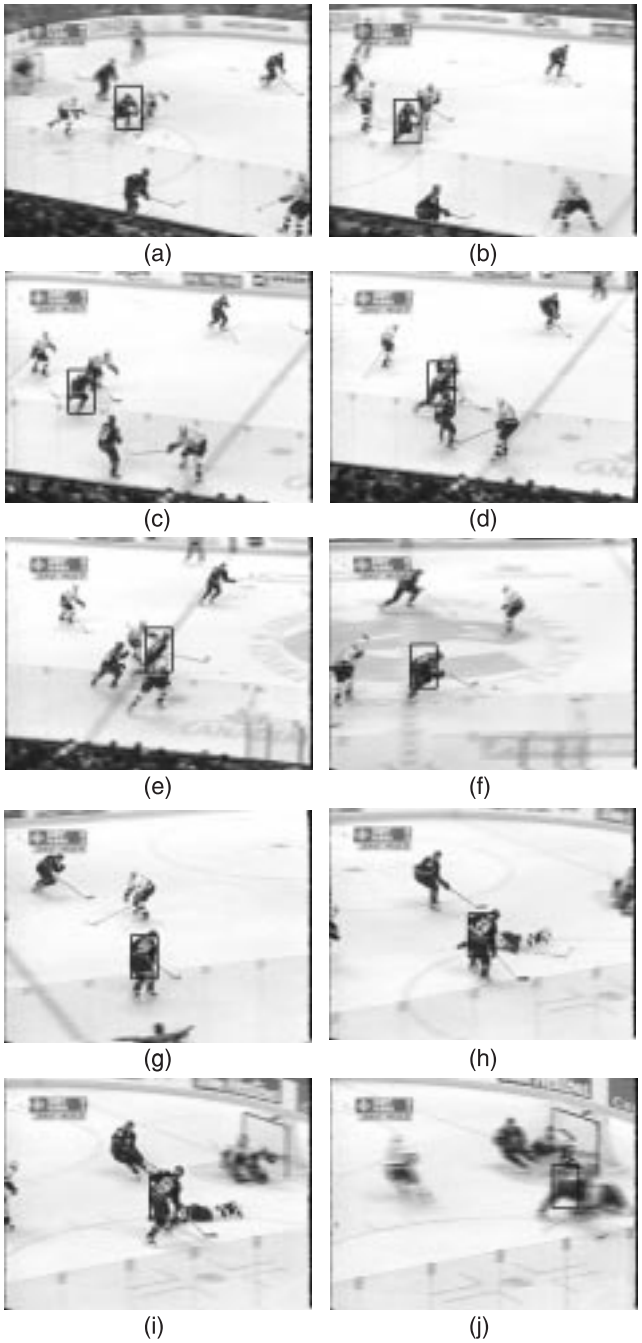


Fig. 1. Tracking results of the hockey sequence; frame 5 (a), frame 20 (b), frame 35 (c), frame 45 (d), frame 59 (e), frame 90 (f), frame 128 (g), frame 158 (h), frame 168 (i), and frame 180 (j).

The update coefficient $\tau$ serves as a parameter that controls the rate of the feature adaptation of the target model to the candidate model. With the higher the illumination change between the frames, the larger update coefficient should be selected, thus the new feature model is more influenced by the candidate model. However, such high values may result in a stopped target to track. Of course, smaller update coefficient denotes fewer changes of environment.

## 4. Applying a tracking algorithm on video streams

In order to analyse the performance of the proposed method, two video sequences are tested in the experiments. Our programs are performed on a Pentium IV 2.4 GHz computer using Matlab language V7.0. The RGB colour space is used as the feature space, and the target that we want to track is chosen by hand in the first frame.

The first experiment is performed on the hockey video sequence which has 188 frames of size 320×240. The target region that was selected initially with size 26×46. Figure 1 shows the hockey player is well tracked in partial occlusion and clutter environment. The tracked target is traced out by rectangular window. The number of iterations required in our EVT algorithm for each corresponding frame is compared to the MST method in Fig. 2. The average iteration required in MST method is 6.4168 per frame, while the average iteration required in our proposed EVT is 4.0162 per frame. Thus, our approach needs less iteration to find out the optimum mode.

Another experiment is performed on the tennis video sequence which has 260 frames of size 176×144. The target region was initially defined with size 14×18. Figure 3 shows the head of the tennis player is well tracked in clutter environment. The tracking target is traced out by rectangular window. The number of iterations required in our

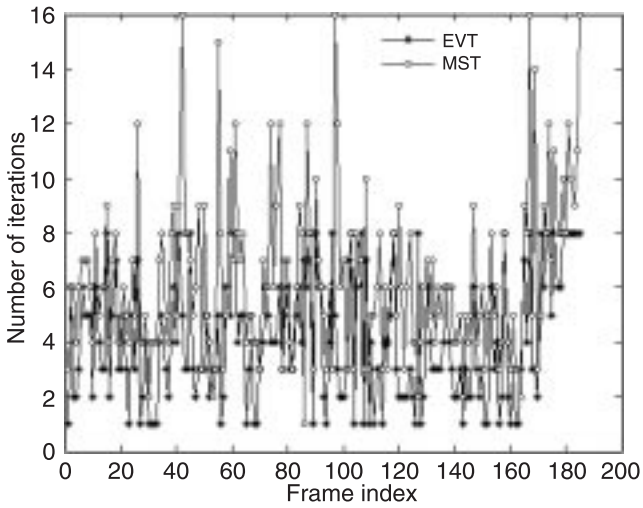


Fig. 2. The number of iteration required in the MST and our proposed EVT.

 

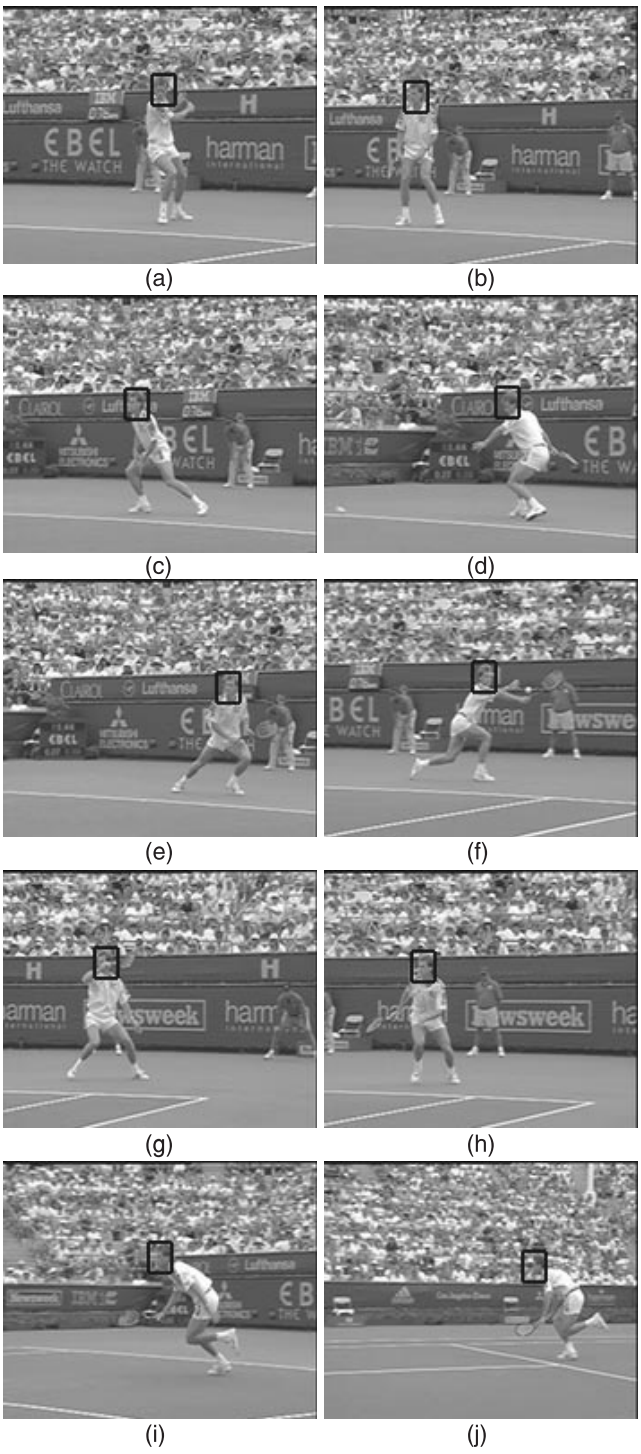


Fig. 3. Tracking results of the tennis sequence; frame 10 (a), frame 50 (b), frame 80 (c), frame 100 (d), frame 170 (e), frame 185 (f), frame 205 (g), frame 220 (h), frame 245 (i), and frame 260 (j).

EVT algorithm for each corresponding frame is compared to the MST method in Fig. 4. The average iteration required in MST method is 4.4692 per frame, while the average iteration required in our proposed EVT is 2.6346 per frame. Thus it is shown that our approach can arrive at optimum mode with less iteration.

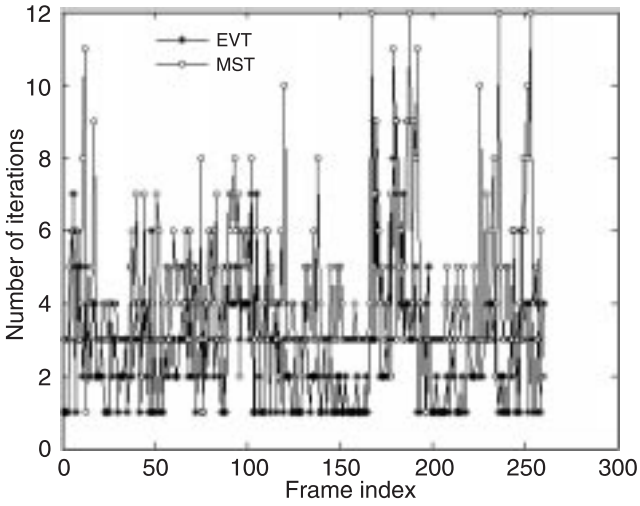


Fig. 4. The number of iteration required in the MST and our proposed EVT.

## 5. Conclusions

In this paper, we described an efficient visual tracking technique based on adaptive kernel model, which is date-driven by local histograms. The background factor we considered is very relevant to the target feature, since it is difficult to distinguish them on the boundary region. Our approach is robust to environment changes and noisy data for the target model is dynamically updated according to the corresponding frames. Experimental results demonstrate that the proposed algorithm requires less iteration, thus it is faster than other method in the literature.

## References

1. W. Ying, "Robust visual tracking by integrating multiple cues based on co-inference learning", *International Journal on Computer Vision* **58**, 55–71 (2004).
2. M. Isard and A. Blake, "Condensation – conditional density propagation for visual tracking", *International Journal on Computer Vision* **29**, 5–28 (1998).
3. A. Logothetis, V. Krishnamurthy, and J. Holst, "A Bayesian EM algorithm for optimal tracking of a maneuvering target in clutter", *Signal Processing* **82**, 473–490 (2002).
4. J.R. Jimenez, V. Medina, and O. Yanez, "Nonparametric MRI segmentation using mean shift and edge confidence maps", *Proc. SPIE* **5032**, 1433–1441 (2003).
5. D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift", *Computer Vision and Pattern Recognition* **2**, 142–149 (2000).
6. K. Nummiaro, E. Koller-Meier, and L. Van Gool, "An adaptive colour-based particle filter", *Image and Vision Computing* **21**, 99–110 (2003).
7. G.D. Hager, M. Dewan, and C.V. Stewart, "Multiple kernel tracking with SSD", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, I-790–I-797 (2004).

8. D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis", *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**, 603–619 (2002).
9. D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-based object tracking", *IEEE Transactions on Pattern Analysis and Machine Intelligence* **25**, 564–577 (2003).
10. M.A. King, T.K. Lee, M.S. Atkins, and D.I. McLean, "Automatic nevi segmentation using adaptive mean shift filters and feature analysis", *Proc. SPIE* **5370**, 1730–1737 (2004).
11. E. Arnaud and E. Memin, "Optimal importance sampling for tracking in image sequences: application to point tracking", *ECCV* 2004, *LNCS* 3023, 302–314 (2004).

 