# Kernel-based Tracking from a Probabilistic Viewpoint

Quang Anh Nguyen[§] Antonio Robles-Kelly[§†] Chunhua Shen[§†]
[§]RSISE, Bldg. 115, Australian National University, Canberra ACT 0200, Australia
[†]National ICT Australia (NICTA),* Locked Bag 8001, Canberra ACT 2601, Australia

## Abstract

*In this paper, we present a probabilistic formulation of kernel-based tracking methods based upon maximum likelihood estimation. To this end, we view the coordinates for the pixels in both, the target model and its candidate as random variables and make use of a generative model so as to cast the tracking task into a maximum likelihood framework. This, in turn, permits the use of the EM-algorithm to estimate a set of latent variables that can be used to update the target-center position. Once the latent variables have been estimated, we use the Kullback-Leibler divergence so as to minimise the mutual information between the target model and candidate distributions in order to develop a target-center update rule and a kernel bandwidth adjustment scheme. The method is very general in nature. We illustrate the utility of our approach for purposes of tracking on real-world video sequences using two alternative kernel functions.*

## 1. Introduction

Object tracking and recognition is a classical problem in the areas of pattern recognition, computer vision and robotics. In visual tracking, one of the main challenges is to achieve robustness to variation in the target model so as to correlate it with an observation in the scene.

Along these lines, kernel-based methods for computer vision [3, 5] have recently gained a great deal of attention and have shown to be a successful approach in the pursuit of robust tracking [6, 4, 11]. Moreover, kernel-based tracking has been a popular alternative to particle filter-based techniques due to its efficiency and robustness. Kernel-based trackers build upon mean-shift optimisation methods which aim at finding the location of the target in the scene. The basic idea here is to represent the target by a convolution of the features with a spatially weighted kernel.

*National ICT Australia is funded by the Australian Government's Backing Australia's Ability initiative, in part through the Australian Research Council.

The mean shift tracker, as introduced in [6], aims at estimating the translation shift of a sub-image area. Elgammal *et al.* [8] have cast the tracking framework in a more general form so as to model joint feature-spatial distributions. In this manner, the spatial structural information of the object can be utilised to improve the performance of the tracker. Yang, Duraiswami and Davis [14] use a fast Gaussian transform to improve on the computation cost of the mean shift optimisation procedure.

In a related development, Collins [4] has employed a scale-governed kernel in addition to the spatial one so as to recover the scale of the target. In [15] a quadratic density distance is adopted for tracking objects under affine transformations. Fan, Wu and Yang [9] build upon [15] and use multiple kernels to enhance the kernel observability by imposing additional constraints upon the tracking process. They also present a strategy so as to avoid singularities in the optimisation involved in the kernel tracking task. Hager *et al.* [11] employ multiple kernels so as to address invariance to rotation and scaling. Here, in order to simplify the optimisation procedure, the tracking equation is linearised and solved by a Newton-style iteration. An akin approach, which employs color-based distributions to estimate not only the position but also also the target shape using its histogram covariance matrix is proposed in [16].

Here, we present a probabilistic formulation of the tracking task which which makes use of a generative model for the target model and candidate. As in the approaches above, the target is tracked making use of a local optimum obtained through an iterative procedure, which, in this case is driven by a maximum likelihood estimation effected using the EM algorithm. In contrast with the approach in [16], where colour distributions are used, here we employ the second moments of the pixel-coordinates so as to obtain a target representation which is invariant to affine transformations. Furthermore, we present a method to estimate the bandwidth of the kernel so as to recover the optimum scale of the target. Thus, our method incorporates the robustness inherent to probabilistic methods while providing a generative model that is quite general in nature. It also provides a means to estimating the scale of the target.

1-4244-1180-7/07/$25.00 ©2007 IEEE

## 2. Maximum Likelihood Formulation

The principle of kernel-based tracking is, in general, not restricted to colour spaces. To provide a probabilistic formulation of the problem, we consider the pixel-coordinates for the pixel in both, the target model and its candidate as random variables which give rise to probability distributions. This treatment permits, in turn, the use of the EM algorithm to estimate a set of latent variables that can then be used to update the target-center position. To this end, we make use of the expectation value to govern the generative model for the probability distribution for the second moments of the pixel coordinates. Since the second moments describe how much the pixel coordinates deviate from the target center, we can perform a maximum a posteriori probability estimation on the second-moment set so as to update the target position.

The section is organised as follows. We commence by introducing the maximum likelihood formulation upon which a set of latent variables can be estimated for visual tracking. We then show how the target-center position can be updated by minimising the mutual information between the distributions corresponding to the candidate and model regions.

### 2.1. Preliminaries

As mentioned earlier, we aim at developing a probabilistic formulation of the tracking problem. To do this, we commence by viewing the pixel-coordinates for the model and the candidate regions as random variables in a quantised feature space. Viewed in this way, these random variables can be used to obtain probability distributions for purposes of tracking. These two distributions, corresponding to the model and the target, are, for the sake of consistency, constructed in the same manner. Thus both regions are treated equally for purposes of developing the likelihood function for the tracking process.

Let the coordinates $\mathcal{X} = \{x_i\}_{i=1...N}$ of the $N$-pixels in either the target model or candidate regions be random variables. For these random variables, the function $b : R^n \rightarrow \{1, 2, \ldots, M\}$ is a mapping which assigns the random variables $x_i$ to a to a set of $M$ mutually exclusive intervals $\{W_j\}_{j=1...M}$ in the quantised $n$-dimensional feature space $\mathcal{W}$.

For tracking purposes, we employ a distribution of $\mathcal{X}$ which is defined with respect to the coordinates $y$ of the center pixel for the tracking region in the current frame. It is important to note that the independent observations in $\mathcal{W}$ are not recorded, but rather the number $N_j$ of them that "fall" in $\mathcal{W}_j$ together with their coordinates in the image lattice. That is, individual observations are made in the n-dimensional feature space $\mathcal{W}$, but only their class intervals $W_j$ and coordinates are available.

Since our $N$ independent observed random variables are given by the coordinates of those pixels in $\mathcal{W}$, the first moment, or average value, of $\mathcal{X}$ is $y$ by construction. Similarly, the second moments are given by the squared distances $\mu_2[x_i] = \langle x_i - y, x_i - y \rangle$, where $\langle \cdot, \cdot \rangle$ denotes the inner product. Note that the second moments of $\mathcal{X}$ are also given by the quantity $E[(\mathcal{X} - y)^2]$, where $E[\cdot]$ is the expectation operator. These relations are importnat since they permit, in the following sections, to cast the tracking task as that of estimating the updated coordinates of the center pixel for the tracking region making use of the second moments of $\mathcal{X}$.

### 2.2. Kernels and Probabilities

Here, we wish to obtain an improved estimate of the target position $y$ by making use of distribution of second moments for the target model and candidate. We do this, by viewing the expectation of the distribution of second moments $\mu_2(\mathcal{X})$ given a class interval as a predictor of the first moment bias that governs the probability distribution which gives rise to the structure of the random variables in $\mathcal{X}$.

Furthermore, the fact that the second moment $\mu_2[x_i]$ can be formulated as an inner product, permits the construction of a kernel function $\mathcal{K}$. For purposes of analysis, consider the class of kernels given by $\mathcal{K}(x_i, y) = P(x_i)P(y)$, where $P(\cdot)$ is a probability density function. We can extend these kernels by taking sums over products of weighted probability distributions [1]. We get

$$\mathcal{K}(\mathcal{X}, y) = \sum_{j=1}^{M} P(\mathcal{X}|h_j)P(y|h_j)P(h_j) \tag{1}$$

where we have used conditional probabilities as an alternative to the density functions $P(\cdot)$ and introduced the set of parameters $\{h_j\}_{j=1,\ldots,M}$ that corresponds to the bandwidths for the class intervals $W_j \in \mathcal{W}$.

In this manner, the kernel $\mathcal{K}(\mathcal{X}, y)$ can be viewed as a function $f(\mu_2[\mathcal{X}], h)$ governed by the bandwidth $h$ and the second moments $\mu_2[\mathcal{X}]$ of $\mathcal{X}$, which is proportional to a mixture distribution of the form

$$\begin{aligned} P(\mu_2[\mathcal{X}]) &= \sum_{j=1}^{M} \pi_j P(E[(\mathcal{X} - y)^2]|h_j) \\ &= \sum_{j=1}^{M} \pi_j P(\mu_2[\mathcal{X}]|h_j) \end{aligned} \tag{2}$$

where $\pi_j$ is the mixture weight. Thus, the marginal for the distribution of second moments with respect to the class interval is

$$P(\mu_2[\mathcal{X}])_{W_j} = \frac{P(E[(\mathcal{X} - y)^2]|h_j)}{\sum_{j=1}^{M} \pi_j P(E[(\mathcal{X} - y)^2]|h_j)} \tag{3}$$

The equation above can be rewritten, given Equation 1 and the proportionality between $\mathcal{K}(\mathcal{X}, y)$ and $P(\mu_2[\mathcal{X}])$, as follows

$$P(\mu_2[\mathcal{X}])_{W_j} = \frac{\sum_{x_i \in W_j} \mathcal{K}(x_i, y)}{\sum_{j=1}^{M} \sum_{x_i \in W_j} \mathcal{K}(x_i, y)} \tag{4}$$

The relevance of the relations above will become evident in following sections. For now, we are concerned with a generative model for the probability distribution of $\mu_2(\mathcal{X})$ which captures whether or not the moments for the observed random variables belong to a class interval indexed $j$. Since the intervals are mutually exclusive, we can consider the moment-set structure for the region $W_j$ to arise as the outcome of a series of Bernoulli trials. Thus, by using the shorthand $\phi_j = P(\mu_2[\mathcal{X}])_{W_j}$, we can include unobservable frequencies in the interval $W_j$ using the conditional probability

$$P(q_j|\phi_j) = \phi_j{}^{q_j}(1-\phi_j)^{(1-q_j)} \tag{5}$$

where $\{q_j\}_{j=1,\ldots,M}$ are latent variables that should be estimated and $\phi_j$ is the probability of success in the Bernoulli distribution, which is given by the distribution of $\mu_2(\mathcal{X})$ in the class interval $W_j$.

## 2.3. Expectation-Maximisation

In this section, we focus on finding the expected values $\phi_j$ and the unobservable frequencies $q_j$ which maximize the likelihood function appearing in Equation 5. To do this, we employ the apparatus of the EM algorithm [7].

The idea underpinning the EM algorithm is to recover maximum likelihood solutions to problems involving missing or hidden data by iterating between two computational steps. In the E (or expectation) step we estimate the posteriori probabilities of the hidden data. The M-step in-turn aims to recover the parameters which maximise the expected value of the log-likelihood function. It is the available a posteriori probabilities from the E-step which allows the weighting of log-likelihood required in the maximisation-step.

For the likelihood function appearing in Equation 5, the expected log-likelihood function is defined as

$$\mathcal{L}(\phi_j, p_j) = p_j \ln(\phi_j) + (1-p_j)\ln(1-\phi_j) \tag{6}$$

Performing algebra and collecting terms, we have

$$\mathcal{L}(\phi, p_j) = p_j \ln\left(\frac{\phi_j}{1-\phi_j}\right) + \ln(1-\phi_j) \tag{7}$$

### 2.3.1 E-Step

In the E-step of the algorithm, we compute the expectation of the hidden data by making use of a gradient-based analysis of the log-likelihood function. Thus, we commence by computing the derivatives of the expected log-likelihood function with respect to the frequency variables

$$\frac{\partial \mathcal{L}(\phi_j, p_j)}{\partial p_j} = \ln\left(\frac{\phi_j}{1-\phi_j}\right) \tag{8}$$

Since the associated saddle-point equations are not tractable in closed form, we use the soft-assign ansatz of Bridle [2] to update the cluster membership assignment variables. This is a form of naive mean field theory [10]. According to mean field theory the latent variables should be updated by replacing them with their expected values [12]. Rather than performing the detailed expectation analysis, soft-assign allows the cluster memberships to be approximated by exponentiating the partial derivatives of the expected log-likelihood function. The updated frequency variables are given by

$$\hat{p}_j = \frac{\exp\left[\frac{\partial \mathcal{L}(\phi_j, p_j)}{\partial p_j}\right]}{\sum_{j=1}^{M} \exp\left[\frac{\partial \mathcal{L}(\phi_j, p_j)}{\partial p_j}\right]} = \frac{\frac{\phi_j}{1-\phi_j}}{\sum_{j=1}^{M} \frac{\phi_j}{1-\phi_j}} \tag{9}$$

### 2.3.2 M-Step

With the updated frequency variables at hand, the update of the random variable $\phi_j$ is a straightforward task. To maximise the log-likelihood, we calculate the derivatives of $\mathcal{L}(\phi, p)$ with respect to $\phi_j$ and equate the result to zero, i.e. we solve

$$\frac{\partial \mathcal{L}(\phi_j, p_j)}{\partial \phi_j} = 0 \tag{10}$$

It is somewhat surprising that, after some algebra and collection of terms, we find that the log-likelihood is maximised when $\hat{\phi}_j = \hat{p}_j$. Recall that $\phi_j = P(\mu_2[\mathcal{X}])_{W_j}$. Hence, the maximum likelihood estimate for the probability of the moments in the class interval $W_j$ is given by its frequency variable.

## 2.4. Minimising Mutual Information

Thus far, we have focus on the probability distributions and likelihoods for the target model and candidate rather than their mutual relationship. For tracking, we are interested in updating the coordinates $y_M$ of the model center pixel given the random variables corresponding to the candidate. In this section, we show how this can be effected by making use of an information theoretic criterion.

Let the two sets of random variables for the model and the candidate be given by $\mathcal{X}_M$ and $\mathcal{X}_D$, respectively. Similarly, the second moments for the two sets of random variables are denoted $\mu_2[\mathcal{X}_M]$ and $\mu_2[\mathcal{X}_D]$. It is straightforward to show that, if the two sets of random variables $\mathcal{X}_M$ and $\mathcal{X}_D$ are equivalent, then the Kullback-Leibler divergence between the distributions of their second moments over the

feature space $\mathcal{W}$ is equal to zero. Further, since the intervals $W_j \in \mathcal{W}$ are mutually exclusive, we can write

$$KL\big(P(\mu_2[\mathcal{X}_M])||P(\mu_2[\mathcal{X}_D])\big) = \sum_{W_j \in \mathcal{W}} P(\mu_2[\mathcal{X}_M]_{W_j}) \ln \left( \frac{P(\mu_2(\mathcal{X}_M)_{W_j})}{P(\mu_2(\mathcal{X}_D)_{W_j})} \right) \quad (11)$$

The advantages of this formulation are twofold. Firstly, we can exploit the fact that, for our generative model in the previous section, the maximum likelihood estimate for the expectation of the second moments of $\mathcal{X}_M$ and $\mathcal{X}_D$ is given by their corresponding frequency variables. Secondly, it permits the use of the physical interpretation of the moments to update the coordinates $y$ by recovering the shift in $y$ that minimises the divergence in Equation 11. Note that, so far, we have worked with the second moments of the observed random variables instead of the variables themselves. As a result, we have not, so far, included the coordinates $y$ implicitly, but rather used them to recover "distances" that we can employ for purposes of inference. As a result, we can update the target center coordinates by using the rule

$$\hat{y} = y + y^* \quad (12)$$

where $y^*$ is the deviation from $y$ that minimises the Kullback-Leibler divergence between the model and the candidate moment distributions.

To minimise the divergence in Equation 11, we note that the updated variables $\hat{\phi}_j$ are the maximum likelihood estimates of the expectation values in Equation 5. Moreover, from Equation 4, we can write

$$\hat{\phi}_j = \frac{\sum_{x_i \in W_j} \mathcal{K}(x_i, \hat{y})}{\sum_{x_k \in \mathcal{W}} \mathcal{K}(x_i, \hat{y})} = \frac{\sum_{x_i \in W_j} f(\hat{\mu}_2[x_i], h)}{\sum_{x_k \in \mathcal{W}} f(\hat{\mu}_2[x_i], h)} \quad (13)$$

where we have written $\hat{\mu}_2[x_i]$ to imply that the second moments above correspond to updated variables of the form

$$\begin{aligned} \hat{\mu}_2[x_i] &= \langle x_i - \hat{y}, x_i - \hat{y} \rangle \\ &= \langle (x_i - y) - y^*, (x_i - y) - y^* \rangle \end{aligned} \quad (14)$$

and used the shorthand $f(\hat{\mu}_2[x_i], h) = \mathcal{K}(x_i, \hat{y})$ to stress that the kernel is here a function of the second moments and the bandwidth $h$. Note that this notation is consistent with that used in section 2.2. By substituting Equations 13 and 14 into Equation 11, we get

$$KL\big(P(\mu_2[\mathcal{X}_M])||P(\mu_2[\mathcal{X}_D])\big) = \frac{1}{\sum_{x_k \in \mathcal{W}} f(\hat{\mu}_2[x_k], h)} \sum_{W_j \in \mathcal{W}} \left\{ \sum_{x_i \in W_j} f(\hat{\mu}_2[x_i], h) \ln \left( \frac{\varphi_i}{\psi_i} \right) \right\} \quad (15)$$

where $\varphi_i$ is the estimated frequency variable $p_j$ for the second moments in the target model random variables $\mathcal{X}_M$ corresponding to the interval $W_j$. The value of $\varphi_i$ is, in turn, the estimated frequency variable for the second moments of the candidate given the class interval $W_j$.

Finally, we can recover the minimum of the equation above with respect to $y^*$ by differentiating and equating to zero. By manipulating terms, this yields

$$\gamma \sum_{x_i \in \mathcal{X}_M} \beta_i \frac{\partial f(\hat{\mu}_2[x_i], h)}{\partial \hat{\mu}_2[x_i]} \frac{d\hat{\mu}_2[x_i]}{dy^*} = 0 \quad (16)$$

where $\gamma$ is a proportionality constant given by $\frac{1}{\sum_{x_k \in \mathcal{W}} f(\hat{\mu}_2[x_i], h)}$ and $\beta_i = \ln\left(\frac{\varphi_i}{\psi_i}\right)$. By using Equation 14 to compute the differential of the second moment with respect to $y^*$ and some algebra, we get

$$y^* = \frac{\sum_{x_i \in \mathcal{X}_M} (x_i - y) \frac{\partial f(\hat{\mu}_2[x_i], h)}{\partial \hat{\mu}_2[x_i]} \beta_i}{\sum_{x_i \in \mathcal{X}_M} \frac{\partial f(\hat{\mu}_2[x_i], h)}{\partial \hat{\mu}_2[x_i]} \beta_i} \quad (17)$$

and, hence, the updated target center coordinates become

$$\hat{y} = y + \frac{\sum_{x_i \in \mathcal{X}_M} (x_i - y) \frac{\partial f(\hat{\mu}_2[x_i], h)}{\partial \hat{\mu}_2[x_i]} \beta_i}{\sum_{x_i \in \mathcal{X}_M} \frac{\partial f(\hat{\mu}_2[x_i], h)}{\partial \hat{\mu}_2[x_i]} \beta_i} \quad (18)$$

Note that the negative derivative of $f(\cdot)$ in the above equations is defined as the shadow of the kernel $f(\cdot)$ in kernel methods [3].

## 3. Updating the Bandwidth

The formulation above has the advantage of allowing the recovery of the optimum bandwidth for the updated parameters. This is due to the problem formulation in section 2.4, where we recovered the updated target center-coordinates making use of the Kullback-Liebler divergence. Hence, to update the bandwidth, we take, hence, an approach similar to that in the previous section and minimise

$$KL\big(P(\mu_2[\mathcal{X}_D])||P(\mu_2[\mathcal{X}_M])\big) = \sum_{W_j \in \mathcal{W}} P(\mu_2[\mathcal{X}_D]_{W_j}) \ln \left( \frac{P(\mu_2(\mathcal{X}_D)_{W_j})}{P(\mu_2(\mathcal{X}_M)_{W_j})} \right) \quad (19)$$

We note that, up to this point, the bandwidth has been considered as a variable that applies equally to all the class intervals in $\mathcal{W}$. Furthermore, in in previous sections, it has played the role of a parameter in the kernel function $\mathcal{K}(\mathcal{X}, y)$. As a result, and without any loss of generality, we can express the updated bandwidth as $\hat{h}$ and aim at minimising

$$KL\big(P(\mu_2[\mathcal{X}_D])||P(\mu_2[\mathcal{X}_M])\big) = \vartheta \sum_{x_i \in \mathcal{X}_D} f(\mu_2[x_i], \hat{h}) \ln \left( \frac{\psi_i}{\varphi_i} \right) \quad (20)$$

where $\vartheta$ is a normalisation constant.

The equation above is reminiscent of Equation 15. Moreover, we constraint the updated bandwidth $h^*$ to lie in the

interval $[c, d]$, where $c$ and $d$ are real-valued, positive constants such that $0 < c < d$. Thus, by removing the constant $\vartheta$ from further consideration, we can pose the problem of recovering $\hat{h}$ as that of finding the minimum over a weighted linear combination of kernel values. This is

$$\hat{h} = \left\{ h^* \middle| \min_{h^* \in [c,d]} \{ \sum_{x_i \in \mathcal{X}_D} f(\mu_2[x_i], h^*) \eta_i \} \right\} \quad (21)$$

where, as in Equation 16, $\eta_i$ is a weight given by $\ln\left(\frac{\psi_i}{\varphi_i}\right)$.

## 4. Implementation Issues

Having presented the theoretical foundations for our method in the previous sections, we now focus in the use of two kernel functions for purposes of tracking. These two alternatives are the Epanechnikov kernel [5] and the diffusion kernel [3]. The tracking algorithm hence works as follows. Starting from an initial position, which is obtained by the tracking results at the previous frame, the algorithm iterates using Equation 18 to find the position of the target in the frame under consideration. This can be viewed as a local optimum for the Kullback-Leibler divergence which can then be used for purposes of finding the optimal scale solving Equation 21.

Following our developments in section 2, we commence by noting that, for both of the alternatives, the new target center position is governed by the quantity $\beta_i$ This quantity can be seen as a weight defined in terms of the marginal probability of the feature indexed $u$ in the target model. Considering the function $b : R^n \rightarrow \{1, 2, \ldots, M\}$, which allocates the $M$ bin indexes for each pixel in the target model and candidate. We can view the marginal probabilities as two sets of variables indexed to the feature space bins. Thus, can express the marginal probability for the model as follows

$$r_j = C_r \sum_{x_i \in \mathcal{X}_M} \mathcal{K}(x_i, y) \delta[b(x_i) - j] \quad (22)$$

where $\delta$ is the Kronecker delta function, $\mathcal{K}(\cdot)$ is the kernel function and $C_r$ is a normalisation constant defined as follows

$$C_r = \frac{1}{\sum_{x_i \in \mathcal{X}_M} \mathcal{K}(x_i, y)} \quad (23)$$

After successfully calculating the variables $r_j$ for the target model, the marginal probabilities for the target candidate are computed as follows

$$s_j = C_s \sum_{x_i \in \mathcal{X}_D} \mathcal{K}(x_i, y) \delta[b(x_i) - j] \quad (24)$$

where, again, $y$ is the center pixel of the current tracking region and $C_s$ is a normalisation constant given by

$$s_s = \frac{1}{\sum_{x_i \in \mathcal{X}_D} \mathcal{K}(x_i, y)} \quad (25)$$

Given the two sets of variables $p_j$ and $q_j$, it is a straightforward task to compute the weights $\beta_i$ and $\eta_i$ as follows

$$\beta_i = \sum_{u=1}^{M} \ln \left( \frac{\frac{r_u}{1-r_u} \sum_{j=1}^{M} \frac{s_j}{1-s_j}}{\frac{s_u}{1-s_u} \sum_{j=1}^{M} \frac{r_j}{1-r_j}} \right) \delta[b(x_i \in \mathcal{X}_{\mathcal{M}}) - u] \quad (26)$$

$$\eta_i = \sum_{u=1}^{M} \ln \left( \frac{\frac{s_u}{1-s_u} \sum_{j=1}^{M} \frac{r_j}{1-r_j}}{\frac{r_u}{1-r_u} \sum_{j=1}^{M} \frac{s_j}{1-s_j}} \right) \delta[b(x_i \in \mathcal{X}_{\mathcal{D}}) - u] \quad (27)$$

With the equations above at hand, we can proceed to obtain the update rules for the target position and the bandwidth for the two alternative kernel functions under study.

### 4.1. Epanechnikov Kernel

The first of our two alternative functions is the Epanechnikov kernel introduced by Comaniciu and Meer [5] for purposes of mean-shift tracking. In [5], the profile kernel $\mathcal{K}(\mathcal{X}, y)$ for each pixel $\{x_i\}_{i=1\ldots n}$ is defined as a function of the form

$$\mathcal{K}(x_i, y) = \begin{cases} 1 - g(x_i, y)^2 & \text{if } \|g(x_i, y)\| < 1 \\ 0 & \text{otherwise} \end{cases} \quad (28)$$

where $g(x_i, y)$ is a monotonic function which, for tracking purposes is defined with respect to the center pixel $y$ of the tracking region in the current frame. Thus, for rectangular tracking regions centered at $y$ of width $w$ and height $z$ the square of $g(\cdot)$ is typically given by

$$g(x_i, y)^2 = \frac{4\|y - x_i\|^2}{w^2 + z^2} \quad (29)$$

where $h^2 = w^2 + z^2$ is the bandwidth parameter as before. As a result, the function $f(\mu_2[\mathcal{X}], h)$ is given by

$$f(\mu_2[x_i], h) = \begin{cases} 1 - \frac{4\mu_2[x_i]}{h^2} & \text{if } \|\frac{4\mu_2[x_i]}{h^2}\| < 1 \\ 0 & \text{otherwise} \end{cases} \quad (30)$$

For the function above, the derivatives with respect to the second moments $\hat{\mu}_2[x_i]$ are

$$\frac{\partial f(\hat{\mu}_2[x_i], h)}{\partial \hat{\mu}_2[x_i]} = \begin{cases} -\frac{4}{h^2} & \text{if } \|\frac{4\hat{\mu}_2[x_i]}{h^2}\| < 1 \\ 0 & \text{otherwise} \end{cases} \quad (31)$$

Hence, making use of Equations 18 and after some algebra, the update rule for the target center becomes

$$\hat{y} = y + \frac{\sum_{x_i \in \mathcal{X}_M} (x_i - y)\beta_i}{\sum_{x_i \in \mathcal{X}_M} \beta_i} \quad (32)$$

We now turn our attention to the update of the bandwidth $h$. To this end, we recover the updated bandwidth $\hat{h}$ such that

$$\hat{h} = \left\{ h^* \middle| \min_{h^* \in [c,d]} \{ \sum_{x_i \in \mathcal{X}_D} \left(1 - 4\frac{\mu_2[x_i]}{(h^*)^2}\right) \eta_i \} \right\} \quad (33)$$

subject to Gibbs inequality, i.e. $KL(\cdot||\cdot) \geq 0$.

### 4.2. Diffusion Kernel

Having presented the update rules for the Epanechnikov kernel, we now focus in the analogue equations for the diffusion kernel in [3]. This profile kernel is defined as a function of the form

$$\mathcal{K}(x_i, y) = \begin{cases} \lambda \exp(-g(x_i, y)^2) & \text{if } \|g(x_i, y)\| < 1 \\ 0 & \text{otherwise} \end{cases} \quad (34)$$

where $\lambda = \frac{1}{2\pi h^2}$ and $g(x_i, y)$ is a function of the target center pixel $y$ and the pixel coordinates $x_i$. The function $g(\cdot)$ is typically given by

$$g(x_i, y)^2 = \frac{\|y - x_i\|^2}{2h^2} \quad (35)$$

where $h$ is the bandwidth parameter. Thus, the function $f(\mu_2[\mathcal{X}], h)$ is

$$f(\mu_2[x_i], h) = \begin{cases} \frac{1}{2\pi h^2} \exp\left( - \frac{\mu_2[x_i]}{2h^2} \right) & \text{if } \| \frac{\mu_2[x_i]}{2h^2} \| < 1 \\ 0 & \text{otherwise} \end{cases} \quad (36)$$

Similarly, the partial derivative of $f(\hat{\mu}_2[x_i], h)$ with respect to $\hat{\mu}_2[x_i]$ becomes

$$\frac{\partial f(\mu_2[x_i], h)}{\partial \hat{\mu}_2[x_i]} = \begin{cases} -\frac{1}{4\pi h^4} \exp\left( - \frac{\hat{\mu}_2[x_i]}{2h^2} \right) & \text{if } \| \frac{\hat{\mu}_2[x_i]}{2h^2} \| < 1 \\ 0 & \text{otherwise} \end{cases} \quad (37)$$

In contrast with Equation 31, the expression above depends on the moments $\hat{\mu}_2[xi] = ||x_i - \hat{y}||^2$, which are governed by our aim of computation, i.e. the variable $\hat{y}$. Furthermore, due to the exponential term involved, the evaluation of the associated update rule becomes non-tractable in closed form. Rather than making a detailed analysis, we note that, if the shift in the target center coordinates is small, we can consider the moments $\hat{\mu}_2[xi] = ||x_i - \hat{y}||^2$ to be approximately equal to $\mu_2[xi] = ||x_i - y||^2$. By substituting Equation 37 into Equation 18 and using $\mu_2[x_i]$ as an alternative to $\hat{\mu}_2[x_i]$, we get

$$\hat{y} = y + \frac{\sum_{x_i \in \mathcal{X}_M} (x_i - y) \exp\left( - \frac{\mu_2[x_i]}{2h^2} \right) \beta_i}{\sum_{x_i \in \mathcal{X}_M} \exp\left( - \frac{\mu_2[x_i]}{2h^2} \right) \beta_i} \quad (38)$$

For the update of the bandwidth, we solve numerically Equation 21. Thus, we recover the updated bandwidth such that

$$\hat{h} = \left\{ h^* \middle| \min_{h^* \in [c,d]} \{ \sum_{x_i \in \mathcal{X}_D} \frac{\eta_i}{2\pi (h^*)^2} \exp\left( - \frac{\mu_2[x_i]}{2(h^*)^2} \right) \} \right\} \quad (39)$$

in the interval $[c, d] = [h - \tau h, h + \tau h]$, where $\tau$ is a constant.

## 5. Experiments

In order to characterize a target, one or more feature spaces must be determined such that a non-parametric power density function can be estimated. The ideal choice of feature space will be the one that is distinctive of the target with respect to the surrounding background while being robust to noise and image corruption. The most common feature space is the RGB colour. However, there are other alternatives such as brightness or contrast. Following Comaniciu et at [5], we have used the RGB colour intensity as the feature upon which the tracker should operate upon and constructed a $16 \times 16 \times 16$-bin colour histogram. Further, this histogram is, in fact, a three-dimensional cube in which every dimension corresponds to a colour channel, i.e. Red, Green and Blue.

Thus, in our experiments, the first instance of the target is selected as a rectangular region by the user at the initial frame of the image sequence under study. For each pixel in this region, the colour intensities of each channel are subject to a 16-level quantisation process. The quantised intensities act as indices to allocate each of the pixels in the target image region to a bin in the histogram. The further away the pixel from the center point of the target, the smaller the weight. To ensure the histogram is a power density function representing the target, a normalisation step is applied at the end of the process. In the case of the diffusion kernel, we have set set $\tau$ to $0.2$ for all the video sequences under study.

For purposes of illustrating the utility of our tracking method, we have used three image sequences. These are a sequence which depicts a racing car that looses control in a curve, a video of a user taken using a webcam and a table tennis clip. We have compared our results to those yield by other three competing algorithms elsewhere in the literature. The first of the alternatives is the tracker introduced in [16][1], which shares with our approach the capability of estimating both, target position and scale. The other two alternatives are the particle filter-based tracker described in [13] and the mean shift tracker in [6]. For the particle filter, the state space is given by the target position and scale. Following [13], we have adopted the HSV colour histogram with 110 bins. In all the experiments, we have set the number of particles to 600 and chose the standard deviation of the Gaussian noise to the value which provides the best results for each sequence.

In Figure 1 we present example results for the frame 25 (left-hand column) and 56 (right-hand column) of the racing car sequence. In the top-most row, we show the results yield by our approach when the diffusion kernel is used. The results obtained using the Epanechnikov kernel are shown in the second row. The third, fourth and fifth row show the

[1] For our experiments, we have used the code available at the author's website. The code can be downloaded from http://staff.science.uva.nl/~zivkovic/.

results for the EM-like shift [16], mean-shift [6] and a Particle filter [13], respectively. From the two top rows, it is clear that both of our trackers (with the Epanechnikov kernel and diffusion kernel) are able to track the car successfully. More importantly, both of them get a more accurate estimate of the target's scale. Although it can successfully track the target's center, the EM-like shift tracker [16] has difficulties to recover the scale when the target undergoes large scale variation. For the particle filter, we set the standard deviation for the Gaussian noise to a small value, i.e. $0.005$ pixels. Nonetheless, the particle filter over estimates the scale of the car in the last several frames.

In Figure 2, we present example results for a webcam video. Here, we track the face of the subject in an office environment. For this sequence, the standard deviation for the Gaussian noise in the particle filter has been set to $0.01$ pixels. Again, from the results yield by our trackers (two top-most rows), we can conclude that our method can estimate scale changes. The EM-like shift (third row) fails to recover the scale correctly for frame indexed 229 in the sequence. The standard mean shift and the particle filter produce reasonably good results (shown in the two bottom-most rows). This is due to the fact that, for this sequence, the scale of the face does not change significantly.

Finally, in Figure 3, we show the results for frames indexed 5 and 85 of the table tennis clip. In the figure, we repeat the presentation layout in Figures 1 and 2. From the panels, we can conclude that our approach for the two kernels under study, i.e. the diffusion and the Epanechnikov kernels, and the particle filter, with a standard deviation of 1.5 pixels, perform much better than the EM-like shift and the mean shift trackers. Moreover, the mean shift completely loses the target at frame $85$.

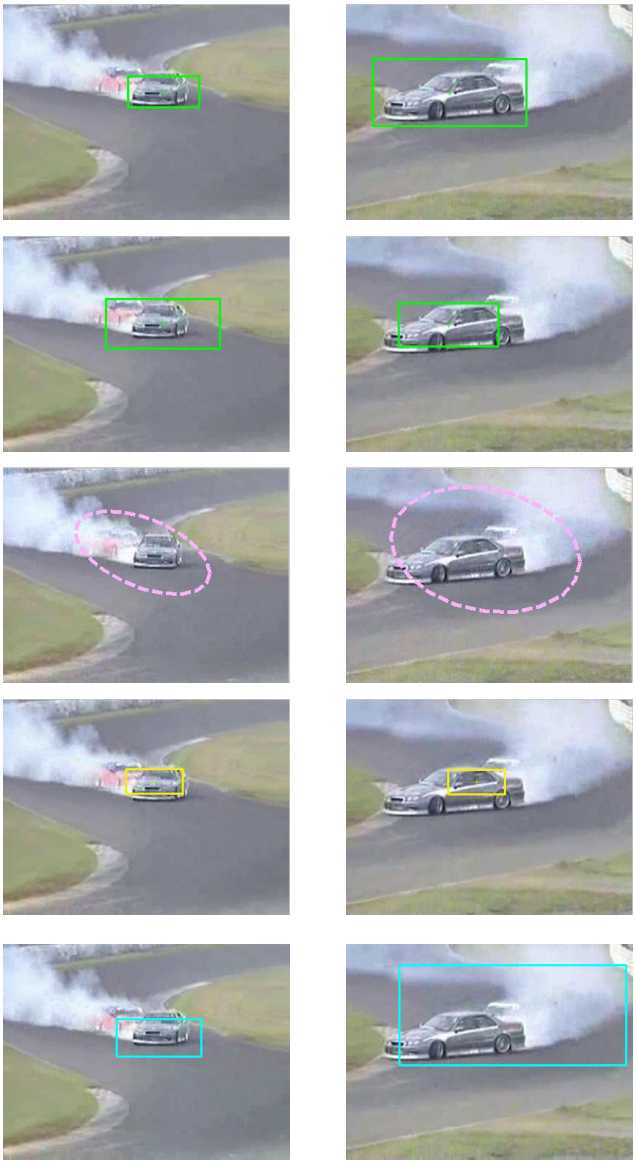

Figure 1. Results for the "Racing Car" sequence at frame 25 and 56. From top-to-bottom: results yield using the Diffusion and the Epanechnikov Kernels, EM-like Shift [16], mean-shift [6] and a Particle filter [13].

## 6. Conlusions

In this paper, we have presented a new probabilistic interpretation for kernel-based object tracking. By viewing the coordinates for the pixels as random variables, the tracking task can be cast into a maximum likelihood framework. This treatment naturally lends itself to the recovery of the scale of the target in the scene. It is worth noting that the approach is quite general and can employ a variety of kernel functions. Here, we have shown how to use the diffusion and the Epanechnikov kernels, to robustly track the objects as well as adapt to changes in scale of the object.

## References


[1] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[2] J. S. Bridle. Training stochastic model recognition algorithms can lead to maximum mutual information estimation of parameters. In *NIPS 2*, pages 211–217, 1990.

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

[4] R. T. Collins. Mean-shift blob tracking through scale space. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 234–240, 2003.

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

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

[7] A. Dempster, N. Laird, and D. Rubin. Maximum-likehood from incomplete data via the EM algorithm. *J. Royal Statistical Soc. Ser. B (methodological)*, 39:1–38, 1977.

[8] A. Elgammal, R. Duraiswami, and L. S. Davis. Probabilistic tracking in joint feature-spatial spaces. In *IEEE Conf. on Comp. Vision and Patt. Recog.*, pages I:781–788, 2003.

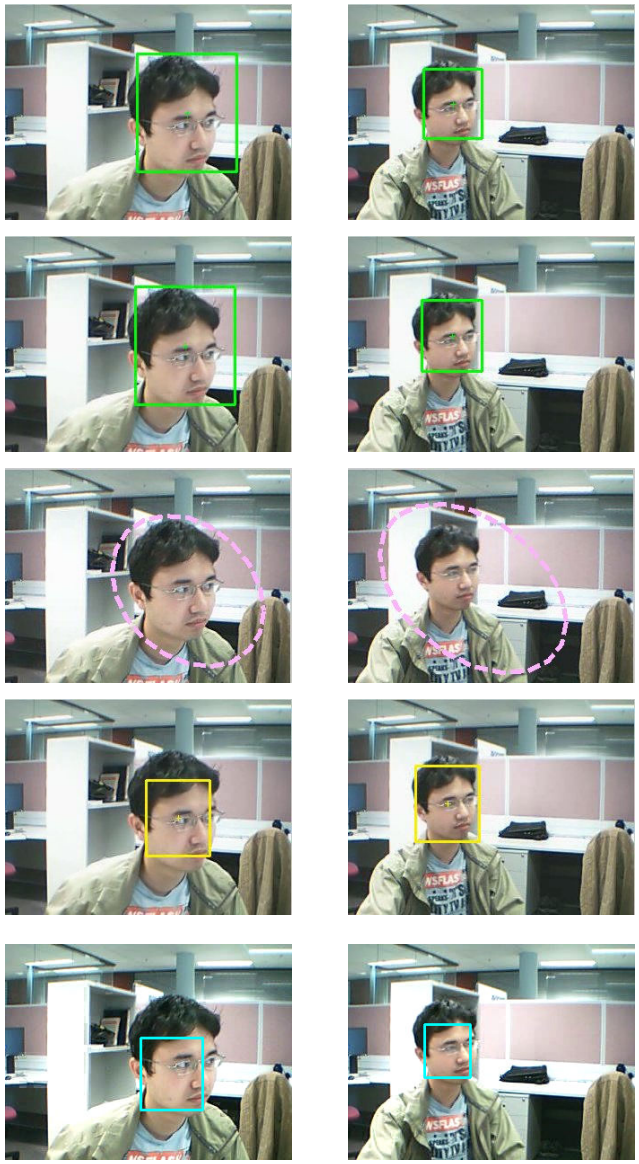

Figure 2. Results for the "Face" sequence at frame 191 and 229. From top-to-bottom: results yield using the Diffusion and the Epanechnikov Kernels, EM-like Shift [16], mean-shift [6] and a Particle filter [13].

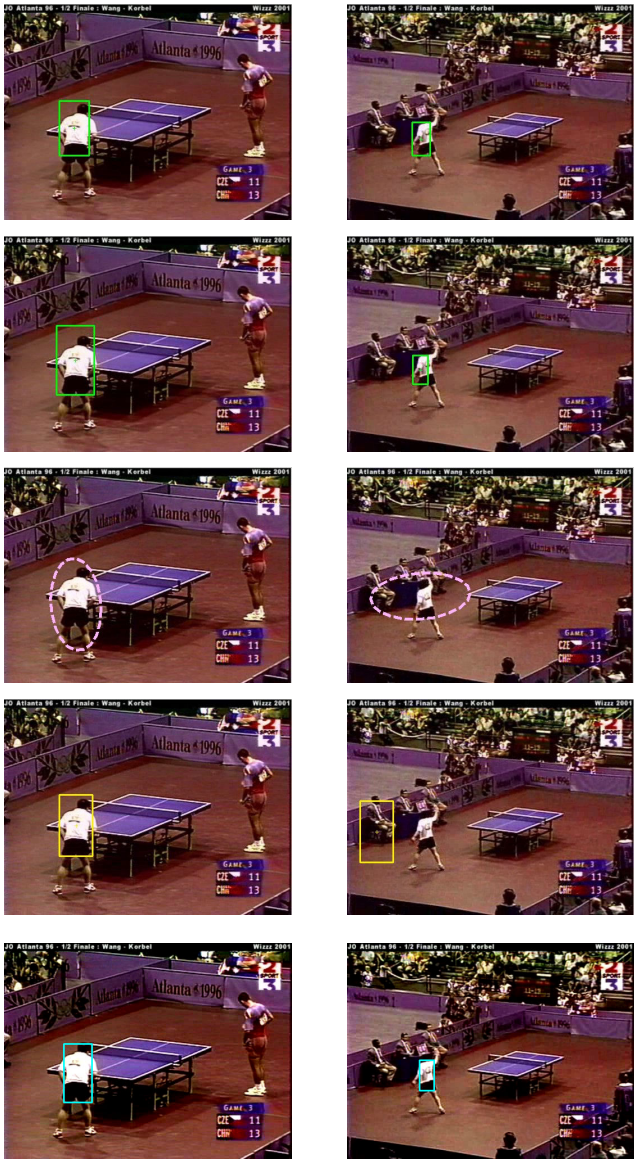

Figure 3. Results for the "Table Tennis" sequence at frame 5 and 85. From top-to-bottom: results yield using the Diffusion and the Epanechnikov Kernels, EM-like Shift [16], mean-shift [6] and a Particle filter [13].


[9] Z. Fan, Y. Wu, and M. Yang. Multiple collaborative kernel tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 502–509, San Diego, CA, 2005.

[10] Z. Ghahramani and M. Jordan. Factorial hidden markov models. *Machine Learning*, 29(2-3):245–273, 1997.

[11] G. D. Hager, M. Dewan, and C. V. Stewart. Multiple kernel tracking with SSD. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 790–797, Washington, D.C., 2004.

[12] T. Hofmann and M. Buhmann. Pairwise data clustering by deterministic annealing. *IEEE Tansactions on Pattern Analysis and Machine Intelligence*, 19(1):1–14, 1997.

[13] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. In *European Conf. on Comp. Vision*, volume 2350, pages 661–675, 2002.

[14] C. Yang, R. Duraiswami, and L. Davis. Fast multiple object tracking via a hierarchical particle filter. In *IEEE International Conference on Computer Vision*, Beijing, China, 2005.

[15] H. Zhang, W. Huang, Z. Huang, and L. Li. Affine object tracking with kernel-based spatial-color representation. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 293–300, San Diego, CA, 2005.

[16] Z. Zivkovic and B. Krose. An EM-like algorithm for color-histogram-based object tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 798–803, 2004.