Robust Density Comparison for Visual Tracking

Omar Arif
omararif@gatech.edu
Patricio Antonio Vela
pvela@ece.gatech.edu

School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta GA, 30332, USA

Abstract

This paper presents a technique to robustly compare two distributions represented by samples, without explicitly estimating the density. The method is based on mapping the distributions into a reproducing kernel Hilbert space, where eigenvalue decomposition is performed. Retention of only the top M eigenvectors minimizes the effect of noise on density comparison. A sample application of the technique is visual tracking, where an object is tracked by minimizing the distance between a model distribution and candidate distributions.

1 Introduction

Many problems in computer vision require measuring the distance between two distributions. For example, in visual tracking, the object to be tracked is presumed to be characterized by a probability distribution [2, 7, 14]. To track the object, each image of the sequence is searched to find the region whose sample distribution closely matches the model distribution. One popular algorithm, the mean shift [2], calculates the distance between the distributions using Bhattacharya coefficient. Elgammal [3] employs a joint appearance-spatial density estimate and measures the similarity of the model with the candidate distribution using the Kullback-Leibler information distance.

Similarly in some contour based segmentation algorithms [4, 10], the contour is evolved either to separate the distribution of the pixels inside and outside of the contour [10], or to evolve the contour so that the distribution of the pixels inside matches a prior distribution of the target object [4]. In both cases, the distance between the distributions is calculated using Bhattacharya coefficient or Kullback-Leibler information distance.

The algorithms defined above require computing the probability density functions using the samples, which becomes computationally expensive for higher dimensions. Another problem associated with computing probability density functions is the sparseness of the observations within the d-dimensional feature space, especially when the sample set size is small. This makes the similarity measures, such as Kullback-Leibler divergence and Bhattacharya coefficient, computationally unstable [13]. Additionally, these techniques require sophisticated space partitioning and/or bias correction strategies [12].

BMVC 2009 doi:10.5244/C.23.45

Contribution: In this work we propose a novel method to compute the distance between two distributions that is robust to noise and outliers. The method works directly on the samples without requiring the intermediate step of density estimation. It is based on maximum mean discrepancy (MMD) [12], which measures the distance between two distributions in the reproducing kernel Hilbert space (RKHS). MMD has been used to address the two sample problem [6]. The technique described is used to compare distributions within the context of visual tracking.

The remainder of the paper is organized as follows. Section 2 briefly explains the MMD measure, which is followed by a description of the proposed method, the Robust MMD (rMMD), in Section 3. Section 4 derives an object tracking algorithm based on Robust MMD. Tracking results are presented in Section 5.

2 Maximum Mean Discrepancy

Let $\{u_i\}_{i=1}^n$, with $u_i \in \mathbb{R}^d$, be a set of n observations drawn from the distribution P_u. Define a mapping $\phi : u_i \rightarrow \boldsymbol{k}(u_i, \cdot)$ such that $\boldsymbol{k}(u_i, u_j) = \langle \phi(u_i), \phi(u_j) \rangle$, where $\boldsymbol{k}$ is a kernel function, such as the Gaussian kernel,

$$\boldsymbol{k}(u_i, u_j) = \exp\left(\frac{\left|\left|u_i - u_j\right|\right|^2}{2\sigma^2}\right). \tag{1}$$

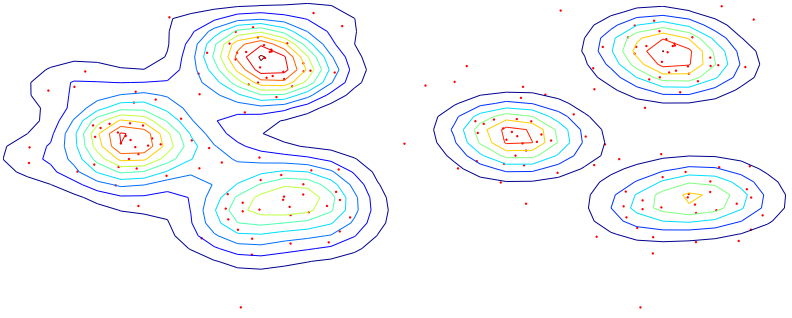
The mean of the mapping is defined as $\mu : P_u \rightarrow \mu[P_u]$, where $\mu[P_u] = E[\phi(u_i)]$. If the finite sample of points $\{u_i\}_{i=1}^n$ are drawn from the distribution P_u, then the unbiased numerical estimate of the mean mapping $\mu[P_u]$ is $\frac{1}{n}\sum_{i=1}^n \boldsymbol{k}(u_i, \cdot)$. Smola et al. [12] showed that the mean mapping and the probability at a test point $u \in \mathbb{R}^d$ are related by the following equation:

$$p(u) = \langle \mu[P_u], \phi(u) \rangle \approx \frac{1}{n}\sum_{i=1}^n \boldsymbol{k}(u, u_i). \tag{2}$$

Equation (2) results in the familiar Parzen window density estimator. In terms of Hilbert space embedding, the density function estimate results from the inner product of the mapped point $\phi(u)$ with the mean of the distribution $\mu[P_u]$. The mean map $\mu : P_u \rightarrow \mu[P_u]$ is injective [12], and allows for the definition of a distance between the distributions P_u and P_v. The distance is defined to be $D(P_u, P_v) := ||\mu[P_u] - \mu[P_v]||$. This distance is called the maximum mean discrepancy (MMD).

3 Robust Maximum Mean Discrepancy

Instead of using Parzen window density estimator, we use an alternate probability density estimation technique proposed by Girolami [5], where kernel principal component analysis (KPCA) [11] is used to provide the discrete expansion coefficients required for a non parametric orthogonal series density estimator. Density estimation in the Hilbert space using KPCA improves the robustness to noise and outliers when compared to Parzen window density estimation. Retention of only the top eigenvectors minimizes the effects of noise on the density estimation as shown in Figure 1. The robust MMD procedure is described below.



(a) Parzen window density estimation. (b) KPCA based density estimation [5].

Figure 1: Non-parametric density estimation of multi-modal, noisy Gaussian distribution.

The probability density at a point u is estimated by the construction of a finite series of orthogonal functions [5],

$$p(u) = \sum_{k=1}^{M} \omega^k \Psi^k(u) \, . \tag{3}$$

where $\{\Psi^k\}_{k=1}^{M}$ are M orthonormal functions with coefficients ω^k. KPCA provides a means to generate the orthonormal functions associated with the estimate of the probability density function (Equation 3). Given a set of samples $\{u_i\}_{i=1}^{n}$, drawn from the distribution P_u, the kernel matrix K is formed with entries $K_{ij} = \boldsymbol{k}(u_i, u_j)$. Let $\boldsymbol{e}^k = [e_1^k, \ldots, e_n^k]$ and λ^k be the k^{th} eigenvector and eigenvalue of the kernel matrix, then the value of function $\Psi^k(u)$ is generated by projecting $\phi(u)$ onto the k^{th} normalized eigenvector V_k

$$\Psi^k(u) = \langle V_k, \phi(u) \rangle = \sum_{i=1}^{n} w_i^k \boldsymbol{k}(u, u_i), \tag{4}$$

where $w_i^k = \frac{e_i^k}{\sqrt{\lambda^k}}$. The coefficients in Equation (3) are given by

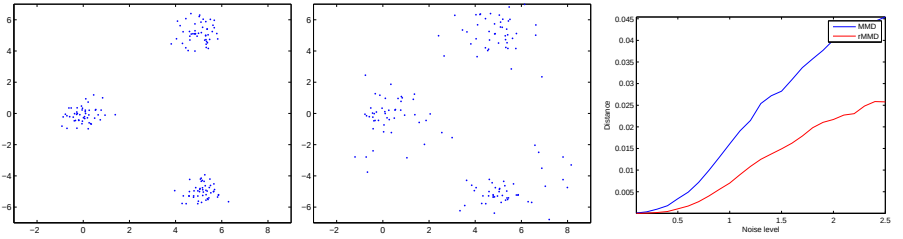
$$\omega^k = E\{\Psi^k(u)\} = \frac{1}{n} \sum_{i=1}^{n} \Psi^k(u_i). \tag{5}$$

Continuing further, the probability density estimate at a test point u has the form,

$$p(u) = \sum_{k=1}^{M} \omega^k \Psi^k(u) = \sum_{k=1}^{M} \omega^k \left\langle V^k, \phi(u) \right\rangle \equiv \langle \mu_r[P_u], \phi(u) \rangle \, , \tag{6}$$

where the final equality defines the proposed robust mean map $\mu_r : P_u \rightarrow \mu_r[P_u]$, with $\mu_r[P_u] := \sum_{k=1}^{M} \omega^k V^k$. The density is estimated by the inner product of the robust mean map $\mu_r[P_u]$ and the mapped point $\phi(u)$. The mean map $\mu_r[P_v]$ for the samples $\{v_i, ..., v_m\}$ is calculated by repeating the same procedure as for P_u. Generating the orthogonal functions in this manner for each sample set is expensive as it requires the eigenvalue decompositions of the associated kernel matrices. The proposed solution is to use the same eigenvectors V^k of the distribution P_u. The distance between the samples is then given by

$$D_r(P_u, P_v) = ||\boldsymbol{\omega_u} - \boldsymbol{\omega_v}|| \, , \tag{7}$$



(a) Distribution 1 samples. (b) Distribution 2 samples obtained by adding noise to distribution 1. (c) distance vs. noise level.

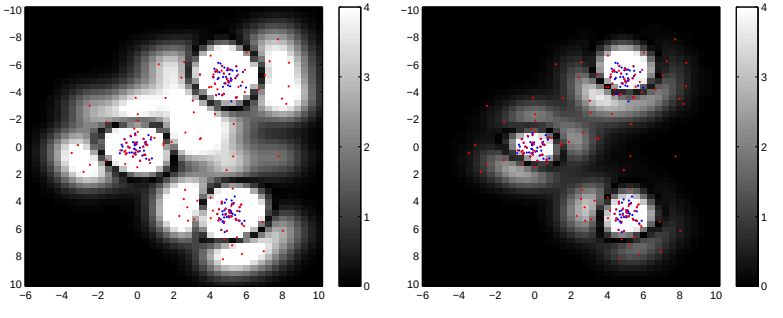
Figure 2: MMD vs robust MMD.

where $\boldsymbol{\omega_u} = [\omega_u^1, \ldots, \omega_u^M]^T$ and $\boldsymbol{\omega_v} = [\omega_v^1, \ldots, \omega_v^M]^T$. Since both mean maps live in the same eigenspace, we have dropped the eigenvectors V^k from Equation (7).

The procedure is summarized below.

- Obtain samples $\{u_i\}_{i=1}^n$ and $\{v_i\}_{i=1}^m$ from two distributions P_u and P_v.
- Form kernel matrix K using the samples from the distribution P_u. Diagonalize the kernel matrix to get eigenvectors $\boldsymbol{e}^k = [e_1^k, \ldots, e_n^k]$ and eigenvalues λ^k for $k = 1, \ldots, M$, where M is the total number of eigenvectors retained.
- Calculate $\boldsymbol{\omega_u}$ using Equation (5), and $\boldsymbol{\omega_v}$ by $\omega_v^k = \frac{1}{m}\sum_{i=1}^m \Psi^k(v_i)$.
- The distance between the distributions is given by Equation (7)

As a simple example, we compute MMD and robust MMD between two distributions. The first is a multi-modal Gaussian distribution. The second is obtained from the first by adding Gaussian noise to about 50% of the samples. Ideally the distance measurement should be zero. Figure 2(c) shows the MMD and robust MMD measure as the standard deviation of the noise is increased. The slope of robust MMD is much lower than MMD showing that it is less sensitive to noise. Figure 3, illustrates the effect of noise on density estimation errors



(a) Density estimate error for MMD (b) Density estimate for rMMD

Figure 3: Illustration of the effect of noise on density estimation errors for MMD vs. rMMD. Samples from the ideal dist. are red and from the corrupted dist. are blue.

for MMD vs. rMMD. Samples from the ideal distribution are red and from the corrupted distribution are blue. The effect of noise is more pronounced in case of MMD.

4 Visual Tracking

An application of the technique developed in the previous section is visual tracking, where an object is tracked by minimizing the distance between a model distribution and given candidate distributions. A key requirement here is that the distance measure should be robust to noise and outliers, which arise for a number of reasons such as noise in imaging procedure, background clutter, partial occlusions, etc. This section provides a gradient based object localization procedure using rMMD.

4.1 Image Pixel Arrangement

The image I is represented as a two-dimensional lattice of a one dimensional intensity image, a three dimensional color image, or some vector valued image. Let $F(x)$ be the p-dimensional appearance vector extracted from I at the spatial location x,

$$F(x) = \Gamma(I,x), \tag{8}$$

where Γ can be any mapping such as color, image gradient, edge, texture etc. The lattice domain is called the *spatial* domain, while the p-dimensional appearance information is called the *appearance* domain. A pixel vector is constructed by concatenating the appearance and the spatial values in a joint appearance-spatial domain of dimension $d = p+2$. Let $u = [F(x),x]^T$ be such a d-dimensional pixel vector, representing a pixel at location x in the joint appearance-spatial domain. The set of all pixel vectors, $\{u_i\}_{i=1}^n$, extracted from the template region R are observations from an underlying density function P_u. To locate the object in an image, a region $\tilde{R}$ (with samples $\{v_i\}_{i=1}^m$) is sought whose density P_v has the minimum distance to the model density P_u as given by Equation (7). The kernel in this case is

$$\boldsymbol{k}(u_i,u_j) = \exp\left(-\frac{1}{2}(u_i-u_j)^T\Sigma^{-1}(u_i-u_j)\right), \tag{9}$$

where Σ is a $d \times d$ diagonal matrix with bandwidths for each appearance-spatial coordinate, $\{\sigma_{F_1},\ldots,\sigma_{F_p},\sigma_{s_1},\sigma_{s_2}\}$. An exhaustive search can be performed to find the region having minimum distance or, starting from an initial guess, gradient based methods can be used to find the local minimum. For the latter approach, we provide a variational localization procedure below.

4.2 Target Localization

Assume that the target object undergoes a geometric transformation from region R to a region $\tilde{R}$, such that $R = T(\tilde{R},a)$, where $a = [a_1,\ldots,a_g]$ is a vector containing the parameters of transformation and g is the total number of transformation parameters. Let $\{u_i\}_{i=1}^n$ and $\{v_i\}_{i=1}^m$ be the samples extracted from region R and $\tilde{R}$, and let $v_i = [F(\tilde{x}_i),T(\tilde{x}_i,a)]^T = [F(\tilde{x}_i),x_i]^T$. The rMMD measure between the distributions of the regions R and $\tilde{R}$ is given by the Equation (7), with the L_2 norm is

$$D_r = \sum_{k=1}^{M}\left(\omega_u^k - \omega_v^k\right)^2, \tag{10}$$

where the M-dimensional robust mean maps for the two regions are $\omega_u^k = \frac{1}{n}\sum_{i=1}^{n}\Psi^k(u_i)$ and $\omega_v^k = \frac{1}{m}\sum_{i=1}^{m}\Psi^k(v_i)$. Gradient descent can be used to minimize the distance with respect to the transformation parameter a. The gradient of Equation (10) with respect to the transformation parameters a is

$$\nabla_a D_r = -2\sum_{k=1}^{M}\left(\omega_u^k - \omega_v^k\right)\nabla_a \omega_v^k, \tag{11}$$

where $\nabla_a \omega_v^k = \frac{1}{m}\sum_{i=1}^{m}\nabla_a \Psi^k(v_i)$. The gradient of $\Psi^k(v_i)$ with respect to a is,

$$\nabla_a \Psi^k(v_i) = \nabla_x \Psi^k(v_i)\cdot\nabla_a T(\tilde{x}, a), \tag{12}$$

where $\nabla_a T(\tilde{x}, a)$ is a $g \times 2$ Jacobian matrix of T and is given by $\nabla_a T = [\frac{\partial T}{\partial a_1}, \ldots, \frac{\partial T}{\partial a_g}]^T$. The gradient $\nabla_x \Psi^k(v_i)$ is computed as,

$$\nabla_x \Psi^k(v_i) = \frac{1}{\sigma_s^2}\sum_{j=1}^{n} w_j^k \boldsymbol{k}(u_j, v_i)(\pi_s(u_j) - x_i), \tag{13}$$

where π_s is a projection from d-dimensional pixel vector to its spatial coordinates, such that $\pi_s(u) = x$ and σ_s is the spatial bandwidth parameter used in kernel $\boldsymbol{k}$. The transformation parameters are updated using the following equation,

$$a(t+1) = a(t) - \delta t \nabla_a D_r, \tag{14}$$

where δt is the time step.

5 Results

This section reports tracking results obtained using Section 4. The pixel vectors are constructed using the color values and the spatial values. The value of σ used in the Gaussian kernel (Equation (9)) is $\sigma_F = 60$ for the color values and $\sigma_s = 4$ for the spatial domain. The number of eigenvectors, M, retained for the density estimation (Equation (3)) were chosen following [5]. In particular, given that the error associated with the eigenvector k is

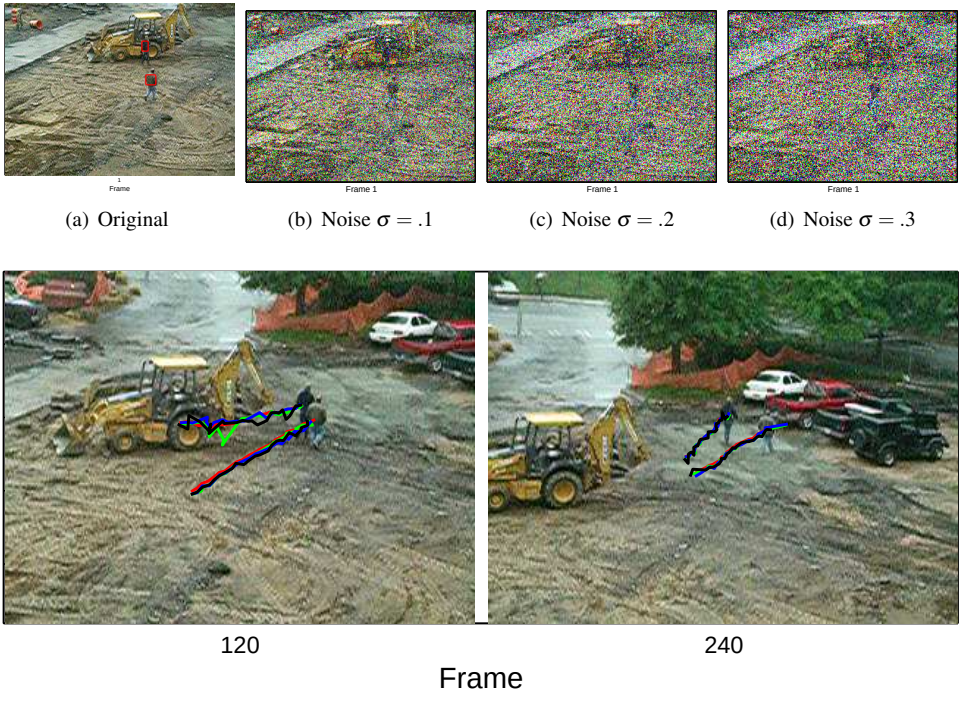
$$\varepsilon^k = (\omega^k)^2 = \left\{\frac{1}{n}\sum_{i=1}^{n}\Psi^k(u_i)\right\}^2, \tag{15}$$

the eigenvectors satisfying the following inequality were retained,

$$\left\{\frac{1}{n}\sum_{i=1}^{n}\Psi^k(u_i)\right\}^2 > \frac{1}{1+n}\left\{\frac{1}{n}\sum_{i=1}^{n}(\Psi^k(u_i))^2\right\}. \tag{16}$$

In practice, about 25 of the top eigenvectors were kept, i.e, $M = 25$. The tracker was implemented using Matlab on an Intel Core2 1.86 GHz processor with 2GB RAM. The run time for the proposed tracker was about 0.5-1 frames/sec, depending upon the object size. The computational complexity of the tracker can be reduced considerably by computing the projections (Equation 4) efficiently as described in [1].

In all the experiments, we consider translation motion and the initial size and location of the target objects are chosen manually. Figure 4 shows results of tracking two people under



(a) Original (b) Noise $\sigma = .1$ (c) Noise $\sigma = .2$ (d) Noise $\sigma = .3$

Figure 4: Construction Sequence. Trajectories of the track points are shown. Red: No noise added, Green: $\sigma = .1$, Blue: $\sigma = .2$, Black: $\sigma = .3$. The tracker tracked in all the cases.

different levels of Gaussian noise. Matlab command `imnoise` was used to add zero mean Gaussian noise of $\sigma = [.1, .2, .3]$. The sample frames are shown in Figure 4(b), 4(c) and 4(e). The trajectories of the track points are also shown. The tracker was able to track in all cases. Mean shift tracker [2] lost track within few frames in case of noise level $\sigma = .1$.

Figure 5 shows the result of tracking the face of a pool player. The method was able to track 100% at different noise levels. The covariance tracker [9] could detect the face correctly for 47.7% of the frames, for the case of no model update (no noise case). The mean shift tracker [2] lost track at noise level $\sigma = .1$.

Figure 6 shows tracking results of a fish sequence. The sequence contains noise, background clutter and fish size changes. The jogging sequence (Figure 7) was tracked in conjunction with Kalman filtering [8] to successfully track through short-term total occlusions.

Table 1: Tracking sequence

Sequence	Resolution	Object size	Total Frames
Construction 1	320×240	15×15	240
Construction 2	320×240	10×15	240
Pool player	352×240	40×40	90
Fish	320×240	30×30	309
Jogging (1st row)	352×288	25×60	303
Jogging (2nd row)	352×288	30×70	111

(a) Sample Frame.

(b) No Noise

(c) Noise $\sigma = .1$. Noise is shown in only two columns for better visualization.

(d) Noise $\sigma = .2$. Noise is shown in only two columns for better visualization.

Figure 5: Face sequence. Montages of extracted results from 90 consecutive frames for different noise levels.

6 Conclusion

We presented a novel density comparison method, which is robust to noise and outliers, given two sets of points sampled from two distributions. The method does not require explicit density estimation as an intermediate step. Possible applications of the proposed density comparison method in computer vision are visual tracking, segmentation, image registration, and stereo registration. We used the technique for visual tracking and provided a variational localization procedure.

Acknowledgement: The research was supported in part by NSF ECCS #0622006.

References

[1] O. Arif and P.A. Vela. Kernel map compression using generalized radial basis functions. In *IEEE International Conference on Computer Vision*, 2009.

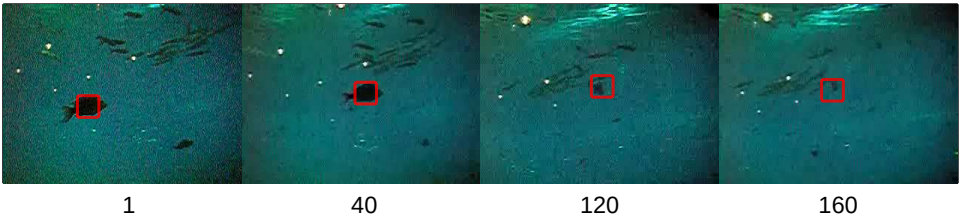


Figure 6: Fish Sequence.

Figure 7: Jogging sequence.

[2] Dorin Comaniciu, Peter Meer, and V. Ramesh. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 564–577, 2003.

[3] A. Elgammal, R. Duraiswami, and L.S. Davis. Probabilistic tracking in joint feature-spatial spaces. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 781–788, 2003.

[4] D. Freedman and T. Zhang. Active contours for tracking distributions. *IEEE Transactions on Image Processing*, pages 518–526, 2004.

[5] M. Girolami. Orthogonal series density estimation and the kernel eigenvalue problem. *Neural Computation*, pages 669–688, 2002.

[6] A. Gretton, K.M. Borgwardt, M.J. Rasch, B. Schölkopf, and A.J. Smola. A kernel method for the two-sample problem. Technical report, Max Planck Institute, 2007.

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

[8] R.E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, pages 35–45, 1960.

[9] F. Porikli, O. Tuzel, and P. Meer. Covariance tracking using model update based on Lie algebra. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 728–735, 2006.

[10] Y. Rathi, J. Malcolm, and A. Tannenbaum. Seeing the unseen: Segmenting with distributions. In *International Conference on Signal and Image Processing*, 2006.

[11] B. Scholköpf, A. Smola, and K. R. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, pages 1299–1319, 1998.

[12] A. Smola, A. Gretton, L. Song, and B. Scholkopf. A Hilbert space embedding for distributions. *Lecture Notes in Computer Science*, 2007.

[13] C. Yang and L. Davis R. Duraiswami. Efficient mean-shift tracking via a new similarity measure. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 176–183, 2005.

[14] Fan Zhimin, Wu Ying, and Ming Yang. Multiple collaborative kernel tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 502–509, 2005.