11

Robust Density Comparison Using Eigenvalue Decomposition

Omar Arif[1] and Patricio A. Vela[2]
[1]*National University of Sciences and Technology*
[2]*Georgia Institute of Technology*
[1]*Pakistan*
[2]*USA*

1. Introduction

Many problems in various fields require measuring the similarity between two distributions. Often, the distributions are represented through samples and no closed form exists for the distribution, or it is unknown what the best parametrization is for the distribution . Therefore, the traditional approach of first estimating the probability distribution using the samples, then comparing the distance between the two distributions is not feasible. In this chapter, a method to compute the similarity between two distributions, which is robust to noise and outliers, is presented. The method works directly on the samples without requiring the intermediate step of density estimation, although the approach is closely related to density estimation. 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.

The chapter is organized in two parts. First, we explain the procedure to obtain the robust density comparison method. The relation between the method and kernel principal component analysis (KPCA) is also explained. The method is validated on synthetic examples. In the second part, we apply the method to the problem of visual tracking. In visual tracking, an initial target and target appearance is given, and must be found within future images. The target information is assumed to be characterized by a probability distribution. Thus tracking, in this scenario, is defined to be the problem of finding the distribution within each image of a sequence that most fits the given target distribution. Here, the object is tracked by minimizing the similarity measure between the model distribution and the candidate distribution where the target position is the optimization variable.

2. Mercer kernels

Let $\{u_i\}_{i=1}^{n}, u_i \in \mathbb{R}^d$, be a set of n observations. A Mercer kernel is a function $\boldsymbol{k} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, which satisfies:

1. $\boldsymbol{k}$ is continuous
2. $\boldsymbol{k}(u_i, u_j) = \boldsymbol{k}(u_j, u_i)$. Symmetric
3. The matrix K, with entries $K_{ij} = \boldsymbol{k}(u_i, u_j)$ is positive definite.

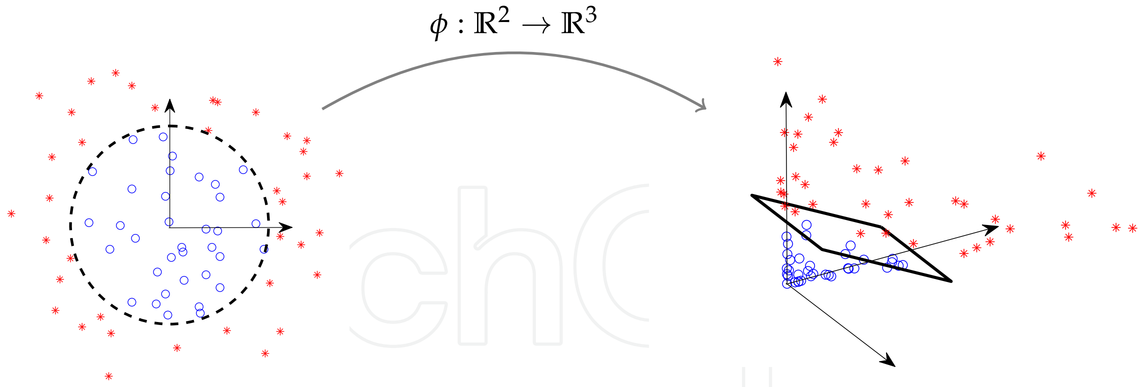


Fig. 1. Toy example: Dot product in the mapped space can be computed using the kernel in the input space.

Theorem: If $\boldsymbol{k}$ is a Mercer kernel then, there exists a high dimensional Hilbert space $\mathcal{H}$ with mapping $\phi : \mathbb{R}^d \to \mathcal{H}$ such that:

$$\phi(u_i) \cdot \phi(u_j) = \boldsymbol{k}(u_i, u_j). \tag{1}$$

The Mercer kernel $\boldsymbol{k}$ implicitly maps the data to a Hilbert space $\mathcal{H}$, where the dot product is given by the kernel $\boldsymbol{k}$.

3. Example

Figure 1 shows a simple binary classification example from Schölkopf & Smola (2001). The true decision boundary is given by the circle in the input space. The points in the input space, $u = [u_1, u_2]^T$, are mapped to $\mathbb{R}^3$ using the mapping $\phi(u) = [u_1^2, \sqrt{2}\, u_1 u_2, u_2^2]^T$. In $\mathbb{R}^3$, the decision boundary is transformed from an circle to a hyperplane, i.e. from a non-linear boundary to a linear one. There are many ways to carry out the mapping ϕ, but the above defined mapping has the important property that the dot product in the mapped space is given by the square of the dot product in the input space. This means that the dot product in the mapped space can be obtained without explicitly computing the mapping ϕ.

$$\begin{aligned} \phi(u) \cdot \phi(v) &= u_1^2 v_1^2 + 2 u_1 v_1 u_2 v_2 + u_2^2 v_2^2 \\ &= (u_1 v_1 + u_2 v_2)^2 = (u \cdot v)^2 = \boldsymbol{k}(u, v). \end{aligned}$$

An example Mercer kernel is Gaussian kernel:

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

where Σ is $d \times d$ covariance matrix.

4. Maximum nean discrepancy

Let $\{u_i\}_{i=1}^{n_u}$, with $u_i \in \mathbb{R}^d$, be a set of n_u observations drawn from the distribution P_u. Define a mapping $\phi : \mathbb{R}^d \to \mathcal{H}$, such that $\left\langle \phi(u_i), \phi(u_j) \right\rangle = \boldsymbol{k}(u_i, u_j)$, where $\boldsymbol{k}$ is a Mercer kernel function, such as the Gaussian kernel. The mean of the mapping is defined as $\mu : P_u \to \mu[P_u]$, where $\mu[P_u] = E[\phi(u_i)]$. If the finite sample of points $\{u_i\}_{i=1}^{n_u}$ are drawn from the distribution P_u, then the unbiased numerical estimate of the mean mapping $\mu[P_u]$ is $\frac{1}{n_u}\sum_{i=1}^{n_u}\phi(u_i)$. Smola et al. (2007) showed that the mean mapping can be used to compute the probability at a test point $u \in \mathbb{R}^d$ as

$$p(u) = \langle \mu[P_u], \phi(u) \rangle \approx \frac{1}{n_u}\sum_{i=1}^{n_u} \boldsymbol{k}(u, u_i). \tag{3}$$

Equation (3) results in the familiar Parzen window density estimator. In terms of the 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 $\mu : P_u \to \mu[P_u]$ is injective, Smola et al. (2007), and allows for the definition of a similarity measure between two sampled sets P_u and P_v, sampled from the same or two different distributions. The measure is defined to be $D(P_u, P_v) := ||\mu[P_u] - \mu[P_v]||$. This similarity measure is called the maximum mean discrepancy (MMD). MMD has been used to address the two sample problem, Gretton et al. (2007). The next section introduces Robust MMD (rMMD).

5. Robust maximum mean discrepancy

In the proposed method, principal component analysis is carried out in the Hilbert space $\mathcal{H}$ and the eigenvectors corresponding to the leading eigenvalues are retained. It is assumed that the lower eigenvectors capture the noise present in the data set. Mapped points in the Hilbert space are reconstructed by projecting them onto the eigenvectors. The reconstructed points are then used to compute the robust mean map. All the computations in the Hilbert space are performed through the Mercer kernel in the input space and no explicit mapping is carried out.

5.1 Eigenvalue decomposition

Let $\{u_i\}_{i=1}^{n_u}$, with $u_i \in \mathbb{R}^d$, be a set of n_u observations. As mentioned before, if $\boldsymbol{k}$ is a Mercer kernel then there exists a high dimensional Hilbert space $\mathcal{H}$, with mapping $\phi : \mathbb{R}^d \to \mathcal{H}$. The covariance matrix $C_{\mathcal{H}}$ in the Hilbert space $\mathcal{H}$ is given by

$$C_{\mathcal{H}} = \frac{1}{n_u}\sum_{i=1}^{n_u} \phi(u_i)\phi(u_i)^T,$$

Empirical computations of $C_{\mathcal{H}}$ require one to know the mapping up front. A technique to avoid this requirement is to perform eigenvalue decomposition of the covariance matrix $C_{\mathcal{H}}$ using the inner product matrix K, called the Gram/kernel matrix, with $K_{ij} = \phi(u_i)^T\phi(u_j) = \boldsymbol{k}(u_i, u_j)$. The Gram matrix allows for an eigenvalue/eigenvector decomposition of the covariance matrix without explicitly computing the mapping ϕ. The Gram kernel matrix can be computed using the Mercer kernel. If $a_i^k, i = 1, \ldots, n$, and λ^k are the k-th eigenvector components and eigenvalue of the kernel matrix K, then the k-th eigenvector of the covariance matrix $C_{\mathcal{H}}$ is given by Leventon (2002)

$$V^k = \frac{1}{\sqrt{\lambda^k}}\sum_{i=1}^{n} a_i^k \phi(u_i).$$

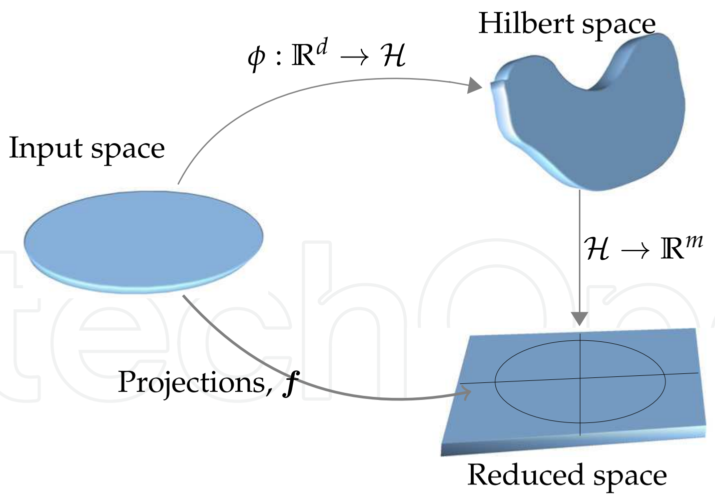

Hilbert space
$\phi : \mathbb{R}^d \to \mathcal{H}$
Input space
$\mathcal{H} \to \mathbb{R}^m$
Projections, $\boldsymbol{f}$
Reduced space

Fig. 2. Eigenvalue decomposition in the Hilbert space $\mathcal{H}$. Observations $\{u_i\}_{i=1}^{n_u}$ are mapped implicitly to the Hilbert space where eigenvalue decomposition results in an m-dimensional reduced space.

5.2 Robust density function

Let $\boldsymbol{V} = [V^1, \cdots, V^m]$ be the m leading eigenvectors of the covariance matrix $C_{\mathcal{H}}$, where the eigenvector V^k is given by

$$V^k = \sum_{i=1}^{n_u} \alpha_i^k \phi(u_i) \qquad \text{with } \alpha_i^k = \frac{a_i^k}{\sqrt{\lambda^k}},$$

where λ^k and a_i^k are the k^{th} eigenvalue and its associated eigenvector components, respectively, of the kernel matrix K. The reconstruction of the point $\phi(u)$ in the Hilbert space $\mathcal{H}$ using m eigenvectors $\boldsymbol{V}$ is

$$\phi_r(u) = \boldsymbol{V} \cdot \boldsymbol{f}(\boldsymbol{u}), \tag{4}$$

where $\boldsymbol{f}(\boldsymbol{u}) = [f^1(u), \ldots, f^m(u)]^T$ is a vector whose components are the projections onto each of the m eigenvectors. The projections are given by

$$f^k(u) = V^k \cdot \phi(u) = \sum_{i=1}^{n_u} \alpha_i^k \boldsymbol{k}(u_i, u), \tag{5}$$

This procedure is schematically described in Figure 2.

Kernel principal component analysis (**KPCA**) Scholköpf et al. (1998) is a non-linear extension of principal component analysis using a Mercer kernel $\boldsymbol{k}$. Eigenvectors $\boldsymbol{V}$ are the principal components and the KPCA projections are given by Equation (5).

The reconstructed points, $\phi_r(u)$, are used to compute the numerical estimate of the robust mean mapping $\mu_r[P_u]$:

$$\mu_r[P_u] = \frac{1}{n_u} \sum_{i=1}^{n_u} \phi_r(u_i) = \frac{1}{n_u} \sum_{i=1}^{n_u} \sum_{k=1}^{m} V^k f^k(u_i) = \sum_{k=1}^{m} \omega^k V^k,$$

where

$$\omega^k = \frac{1}{n_u} \sum_{i=1}^{n_u} f^k(u_i) \tag{6}$$

The density at a point u is then estimated by the inner-product of the robust mean map $\mu_r[P_u]$ and the mapped point $\phi(u)$.

$$p(u) = \mu_r[P_u] \cdot \phi(u) = \sum_{k=1}^{m} \omega^k f^k(u). \tag{7}$$

Retention of only the leading eigenvectors in the procedure minimizes the effects of noise on the density estimate. Figure 3(d) shows density estimation of a multimodal Gaussian distribution in the presence of noise using the robust method. The effect of noise is less pronounced as compared to the kernel density estimation (Figure 3(b)). An alternate procedure that reaches the same result (Equation 7) from a different perspective is proposed by Girolami (2002). There, the probability density is estimated using orthogonal series of functions, which are then approximated using the KPCA eigenfunctions.

5.2.1 Example:

As mentioned before a kernel density estimate, obtained as per Equation (3), is computable using the inner product of the mapped test point and the mean mapping. The sample mean can be influenced by outliers and noise. In Kim & Scott (2008), the sample mean is replaced with a robust estimate using M-estimation Huber et al. (1981). The resulting density function is given by

$$p(u) = \langle \hat{\mu}[P_u], \phi(u) \rangle\,, \tag{8}$$

where the sample mean $\mu[P_u]$ is replaced with a robust mean estimator $\hat{\mu}[P_u]$. The robust mean estimator is computed using the M-estimation criterion

$$\hat{\mu}[P_u] = \underset{\mu[P_u] \in \mathcal{H}}{\arg\min} \sum_{i=1}^{n_u} \rho(||\phi(u_i) - \mu[P_u]||), \tag{9}$$

where ρ is robust loss function. The iterative re-weighted least squares (IRWLS) is used to compute the robust mean estimate. IRWLS depends only on the inner products and can be efficiently implemented using the Mercer kernel Kim & Scott (2008). The resulting density estimation function is

$$p(u) = \frac{1}{n_u} \sum_{i=1}^{n_u} \gamma_i \boldsymbol{k}(u, u_i), \tag{10}$$

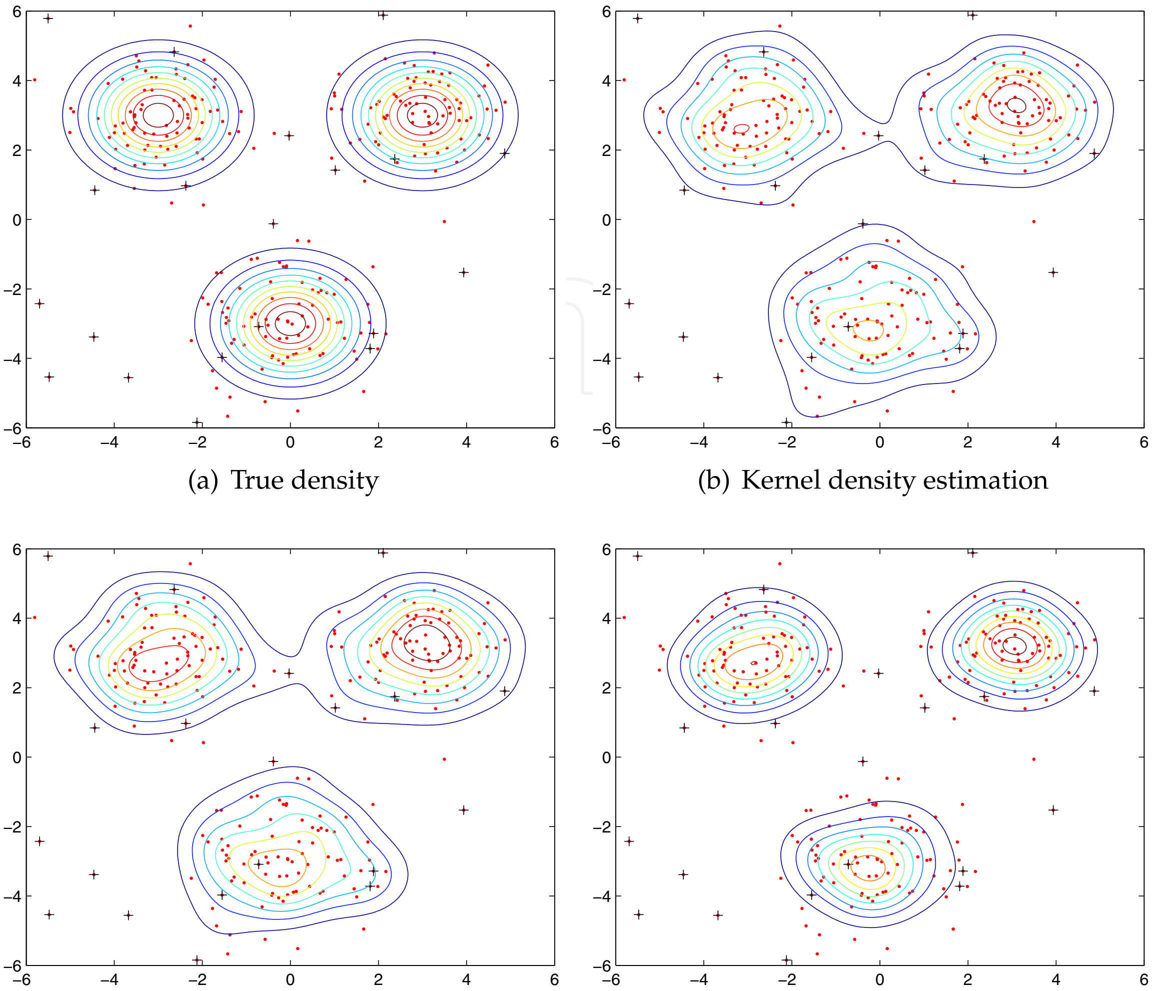
where $\gamma_i \geq 0, \sum \gamma_i = 1$ and γ_i are obtained through IRWLS algorithm. The γ_i values tend to be low for outlier data points.

In this section, we compare the performance of kernel density estimation (Equation (3)), robust density estimation using M-estimation (Equation (10)) and robust density estimation using eigenvalue decomposition (Equation 7).

A sample set X of 150 points is generated from a 2-dimensional multimodal Gaussian distribution

$$X \sim \frac{1}{3}\mathcal{N}_1(\mu_1, \Sigma_1) + \frac{1}{3}\mathcal{N}_2(\mu_2, \Sigma_2) + \frac{1}{3}\mathcal{N}_3(\mu_3, \Sigma_3), \tag{11}$$

where $\mu_1 = [3,3]^T$, $\mu_2 = [-3,3]^T$, $\mu_3 = [0,-3]^T$ and $\Sigma_1 = \Sigma_2 = \Sigma_3 = I$. Outliers are added from a uniform distribution over the domain $[-6,6] \times [-6,6]$. Figure 3 shows the true and the estimated density using the three methods. Data points corresponding to the outliers



(a) True density

(b) Kernel density estimation

(c) Robust Kernel density estimation using M-estimation

(d) Robust Kernel density estimation using eigenvalue decomposition

Fig. 3. Density estimation comparisons. The effect of outliers is less pronounced in the robust density estimation using eigenvalue decomposition.

are marked as $+$. To measure the performance of the density estimates, the Bhattacharyya distance is used. The number of outliers used in the tests are $\Gamma = [20, 40, 60, 80, 100]$. At each Γ the simulations are run 50 times and the average Bhattacharyya distance is recorded. The results are shown in Figure 4. The number of eigenvectors retained for the robust density estimation were 8.

5.3 Robust maximum mean discrepancy

The robust mean map $\mu_r : P_u \rightarrow \mu_r[P_u]$, with $\mu_r[P_u] := \sum_{k=1}^{n_u} \omega^k V^k$, is used to define the similarity measure between the two distributions P_u and P_v. We call it the robust MMD (rMMD),

$$D_r(P_u, P_v) := ||\mu_r[P_u] - \mu_r[P_v]||\,.$$

The mean map $\mu_r[P_v]$ for the samples $\{v_i\}_{i=1}^{n_v}$ is calculated by repeating the same procedure as for P_u. This may be computationally expensive as it requires eigenvalue decomposition of the kernel matrices. Further, the two eigenspaces may be unrelated. The proposed solution is

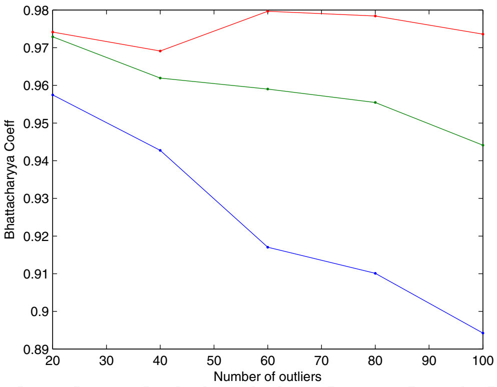


Fig. 4. Bhattacharyya distance measure between true and estimated densities. Red: Robust density estimation using eigenvalue decomposition, Green: robust density estimation using M-estimation, Blue: Kernel density estimation.

to use the eigenvectors V^k of the distribution P_u. The similarity measure between the samples is then given by

$$D_r(P_u, P_v) = ||\boldsymbol{\omega_u} - \boldsymbol{\omega_v}||, \tag{12}$$

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, the eigenvectors V^k have been dropped from the (Equation 12).

5.4 Summary

The procedure is summarized below.

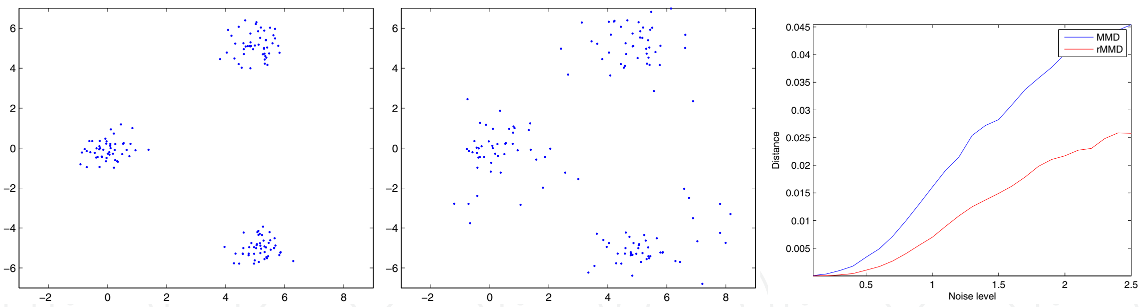
- Given samples $\{u_i\}_{i=1}^{n_u}$ and $\{v_i\}_{i=1}^{n_v}$ from two distributions P_u and P_v.
- Form kernel matrix K using the samples from the distribution P_u. Diagonalize the kernel matrix to get eigenvectors $\boldsymbol{a}^k = [a_1^k, \ldots, a_{n_u}^k]$ and eigenvalues λ^k for $k = 1, ..., m$, where m is the total number of eigenvectors retained.
- Calculate $\boldsymbol{\omega_u}$ using Equation (6), and $\boldsymbol{\omega_v}$ by $\omega_v^k = \frac{1}{n_v}\sum_{i=1}^{n_v} f^k(v_i)$.
- The similarity of P_v to P_u is given by Equation (12).

5.5 Example 1

As a simple synthetic example (visual tracking examples will be given in the next section), we compute MMD and robust MMD between two distributions. The first one is a multi-modal Gaussian distribution given by

$$X \sim \frac{1}{3}\mathcal{N}_1(\mu_1, \Sigma_1) + \frac{1}{3}\mathcal{N}_2(\mu_2, \Sigma_2) + \frac{1}{3}\mathcal{N}_3(\mu_3, \Sigma_3), \tag{13}$$

where $\mu_1 = [0, 0]^T$, $\mu_2 = [5, 5]^T$, $\mu_3 = [5, -5]^T$ and $\Sigma_1 = \Sigma_2 = \Sigma_3 = .5 \times I$. The sample points for the other distribution are obtained from the first one by adding Gaussian noise to about 50% of the samples. Figure 5(c) shows the MMD and robust MMD measure as the standard deviation of the noise is increased. The slope of robust MMD is lower than MMD showing that it is less sensitive to noise. In Figure 6, the absolute value of the difference between the two distributions is plotted for MMD and rMMD measure. The samples from the two distributions are shown in red and blue color. The effect of noise is more pronounced in case of MMD.

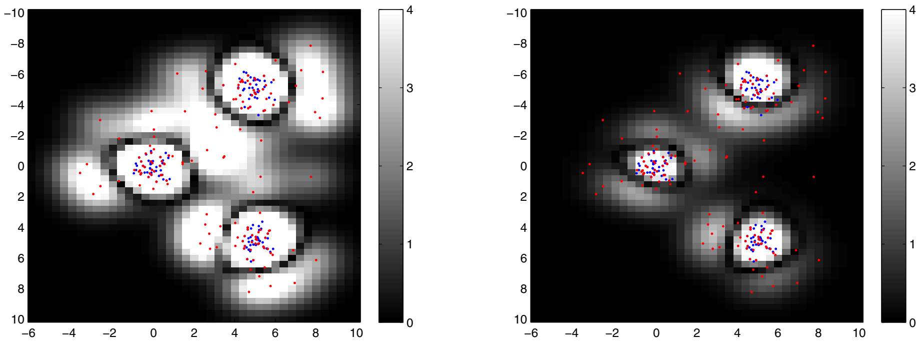


(a) Distribution 1 samples.

(b) Distribution 2 is obtained by adding noise to distribution 1.

(c) Curves measure the similarity measure between the distributions as the noise level increases.

Fig. 5. MMD vs robust MMD.



(a) Difference function for MMD

(b) Difference function for rMMD

Fig. 6. Illustration of the effect of noise on the difference between the the two distributions. The samples from the two distributions are shown in red and blue.

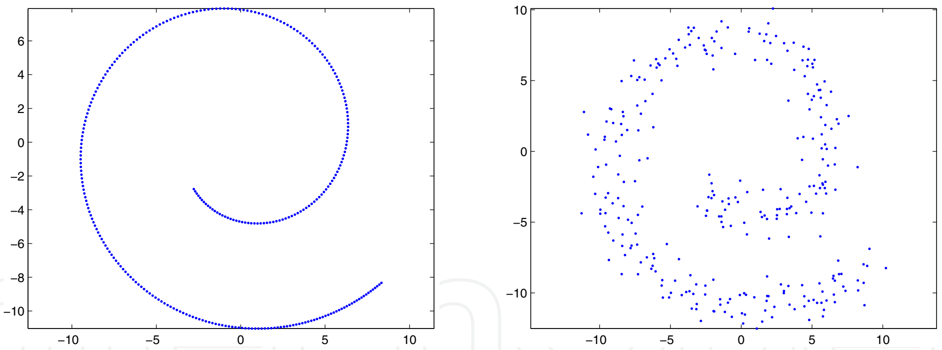
5.6 Example 2

Consider another example of a 2-dimensional swiss roll. The data set is generated by the following function.

$$t = \frac{3}{2} \cdot \pi \cdot (1 + 2r) \; where \; r \geq 0,$$
$$x = t \cdot cos(t),$$
$$y = t \cdot sin(t),$$

where x and y are the coordinates of the data points. 300 points are uniformly sampled and are shown in Figure 7(a). The noisy data sets are obtained by adding Gaussian noise to the original data set at standard deviations, $\sigma = [0 - 1.5]$. For example Figure 7(b) shows a noisy data at $\sigma = 1$. We measure the similarity between the two data sets using MMD and rMMD. 20 eigenvectors are retained for the rMMD computation. Figure 8 shows that the MMD and rMMD measure as the standard deviation of the noise is increased. The slope of rMMD is lower than MMD showing that it is less sensitive to noise.

As mentioned earlier, the eigenvectors corresponding to the lower eigenvalues capture noise present in the data set. The rMMD measure uses the the reconstructed points ϕ_r (Equation 4)



(a) Swiss roll data set (b) Noisy swiss roll data set at $\sigma = 1$

Fig. 7. Swiss roll example

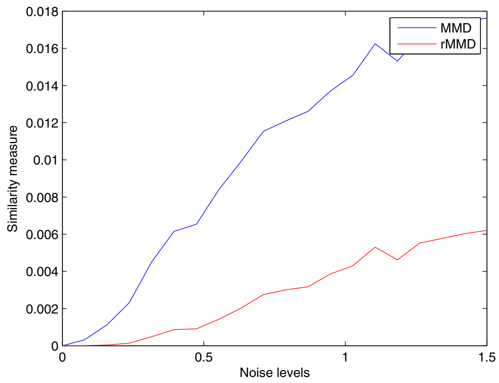


Fig. 8. Curves measure the similarity between the two data sets as the noise level is increased.

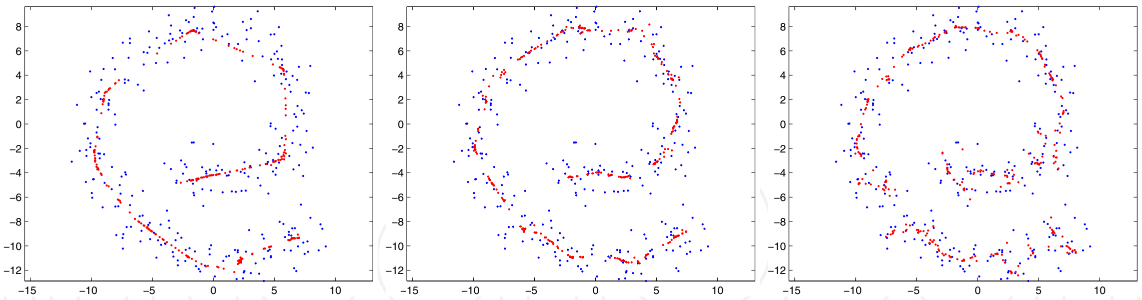


Fig. 9. Reconstruction of the noisy points using 10, 20, 30 eigenvectors Rathi, Dambreville & Tannenbaum (2006). Blue: noise data set, Red: reconstructed points.

to compute the robust mean map. The reconstructed points are obtained by using only the leading eigenvectors. Therefore, the effect of noise on the reconstructed points is reduced. We use the method descibed in Rathi, Dambreville & Tannenbaum (2006) to visualize the reconstructed points in the input space. Figure 9 shows the reconstructed points using 10, 20 and 30 eigenvectors. The blue dots are the noisy data set and the red dots are the reconstructed points. It is clear from the figure that the reconstructed data points using few leading eigenvectors match faithfully to the original data set.

6. Visual tracking through density comparison

In the first part of the chapter, we presented a technique to robustly compare two distributions represented by their samples. An application of the technique is visual target tracking. The object is tracked by finding the correspondence of the object region in consecutive images by using a template or a set of templates of the target object to define a model distribution. To track the object, each image of the sequence is searched to find the region whose candidate distribution closely matches the model distribution. A key requirement here is that the similarity 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.

One popular algorithm, the mean shift tracker Comaniciu et al. (2003), uses a histogram weighted by a spatial kernel as a probability density function of the object region. The correspondence of the target object between sequential frames is established at the region level by maximizing the Bhattacharyya coefficient between the target and the candidate distributions using mean-shift Cheng (1995). Instead of using the Bhattacharyya coefficient as a distance measure between the two distributions, Hager et al. (2004) use the Matusita distance between kernel-modulated histograms. The Matusita distance is optimized using Newton-style iterations, which provides faster convergence than the mean-shift. Histograms discard spatial information, which becomes problematic when faced with occlusions and/or the presence of target features in the background. In Birchfield & Rangarajan (2005), histograms were generalized to include spatial information, leading to spatiograms. A spatiogram augments each histogram bin with the spatial means and covariances of the pixels comprising the bin. The spatiograms captures the probability density function of the image values. The similarity between the two density functions was computed using Bhattacharyya coefficient. Elgammal Elgammal et al. (2003) employs a joint appearance-spatial density estimate and measure the similarity of the model and the candidate distributions using Kullback-Leigbler information distance.

Similarly, measuring the similarity/distance between two distributions is also required in image segmentation. For example, in some contour based segmentation algorithms (Freedman & Zhang (2004); Rathi, Malcolm & Tannenbaum (2006)), the contour is evolved either to separate the distribution of the pixels inside and outside of the contour, or to evolve the contour so that the distribution of the pixels inside matches a prior distribution of the target object. In both cases, the distance between the distributions is calculated using Bhattacharyya coefficient or Kullback-Liebler 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 Bhattacharyya coefficient, computationally unstable, Yang & R. Duraiswami (2005). Additionally, these techniques require sophisticated space partitioning and/or bias correction strategies Smola et al. (2007).

This section describes a method to use robust maximum mean discrepancy (rMMD) measure, described in the first part of this chapter, for visual tracking. The similarity between the two distributions can be computed directly on the samples without requiring the intermediate step of density estimation. Also, the model density function is designed to capture the appearance and spatial characteristics of the target object.

6.1 Extracting target feature vectors

The feature vector associated to a given pixel is a d-dimensional concatenation of a p-dimensional appearance vector and a 2-dimensional spatial vector $u = [\mathcal{F}(x), x]$, where $\mathcal{F}(x)$ is the p-dimensional appearance vector extracted from $\mathcal{I}$ at the spatial location x,

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

where Γ can be any mapping such as color $\mathcal{I}(x)$, image gradient, edge, texture, etc., any combination of these, or the output from a filter bank (Gabor filter, wavelet, etc.).

The feature vectors are extracted from the segmented target template image(s). The set of all feature vectors define the target input space $\mathbb{D}$,

$$\mathbb{D} = \{u_1, u_2, ..., u_n\},$$

where n is the total number of feature vectors extracted from the template image(s). The set of all pixel vectors, $\{u_i\}_{i=1}^{n_u}$, 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 $\{v_i\}_{i=1}^{n_v}$) with density P_v is sought which minimizes the rMMD measure given by Equation (12). The kernel in this case is

$$\boldsymbol{k}(u_i, u_j) = \exp\left(-\frac{1}{2}(u_i - u_j)^T \Sigma^{-1} (u_i - u_j)\right), \tag{14}$$

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 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.

6.2 Variational 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 $\{u_i\}_{i=1}^{n_u}$ and $\{v_i\}_{i=1}^{n_v}$ be the samples extracted from region R and $\tilde{R}$, and let $v_i = [\mathcal{F}(\tilde{x}_i), T(\tilde{x}_i, a)]^T = [\mathcal{F}(\tilde{x}_i), x_i]^T$. The rMMD measure between the distributions of the regions R and $\tilde{R}$ is given by the Equation (12), with the L_2 norm is

$$D_r = \sum_{k=1}^{m} \left(\omega_u^k - \omega_v^k\right)^2, \tag{15}$$

where the m-dimensional robust mean maps for the two regions are $\omega_u^k = \frac{1}{n_u} \sum_{i=1}^{n_u} f^k(u_i)$ and $\omega_v^k = \frac{1}{n_v} \sum_{i=1}^{n_v} f^k(v_i)$. Gradient descent can be used to minimize the rMMD measure with respect to the transformation parameter a. The gradient of Equation (15) with respect to the transformation parameters a is

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

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

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

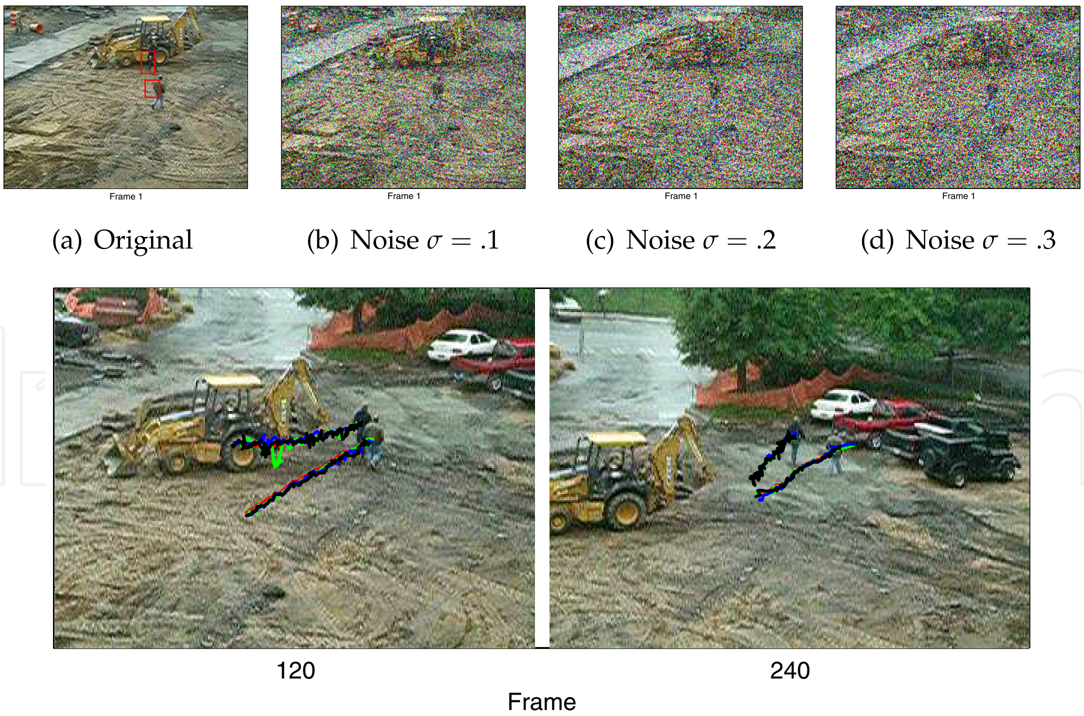
where $\nabla_a T(\tilde{x}, a)$ is a $g \times 2$ Jacobian matrix of T and is given by $\nabla_a T = [\frac{\partial T}{\partial a_1}, \ldots, \frac{\partial T}{\partial a_g}]^T$. The gradient $\nabla_x f^k(v_i)$ is computed as,

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

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,

$$a(t+1) = a(t) - \delta t \nabla_a D_r,$$

where δt is the time step.



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

Fig. 10. Construction Sequence. Trajectories of the track points are shown. Red: No noise added, Green: $\sigma = .1$, Blue: $\sigma = .2$, Black: $\sigma = .3$.

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

Table 1. Tracking sequence

6.3 Results

The tracker was applied to a collection of video sequences. The pixel vectors are constructed using the color values and the spatial values. The value of σ used in the Gaussian kernel 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 were chosen following Girolami (2002). In particular, given that the error associated with the eigenvector k is

$$\epsilon^k = (\omega^k)^2 = \left\{ \frac{1}{n} \sum_{i=1}^{n} f^k(u_i) \right\}^2, \tag{16}$$

the eigenvectors satisfying the following inequality were retained,

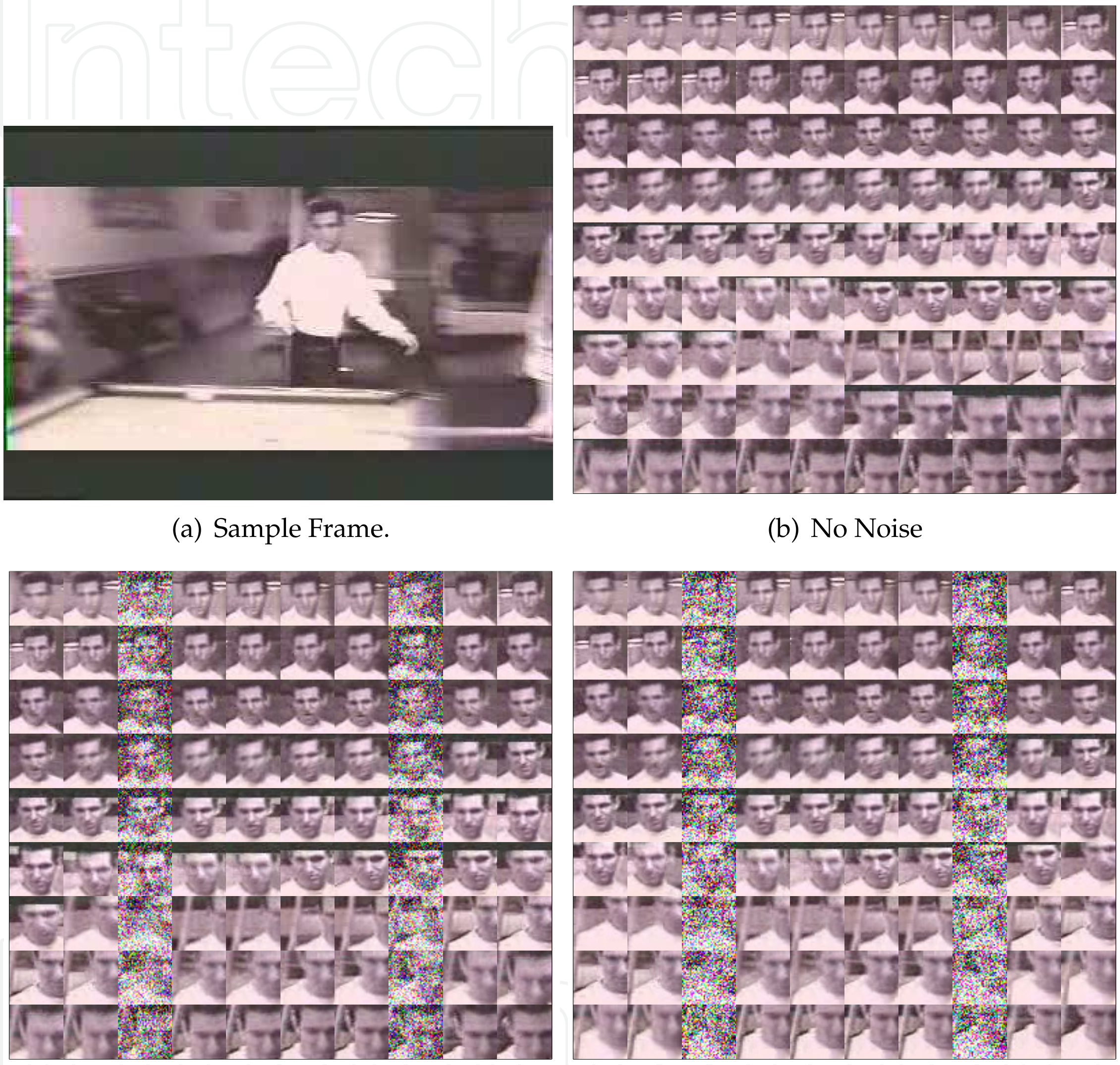
$$\left\{ \frac{1}{n} \sum_{i=1}^{n} f^k(u_i) \right\}^2 > \frac{1}{1+n} \left\{ \frac{1}{n} \sum_{i=1}^{n} (f^k(u_i))^2 \right\}. \tag{17}$$

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.

In all the experiments, we consider translation motion and the initial size and location of the target objects are chosen manually. Figure 10 shows results of tracking two people under 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 10(b), 10(c) and 10(e). The trajectories of the track points are also shown. The tracker was able to track in all cases. The mean shift tracker (Comaniciu et al. (2003)) lost track within few frames in case of noise level $\sigma = .1$.

Figure 11 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 Porikli et al. (2006) 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 Comaniciu et al. (2003) lost track at noise level $\sigma = .1$.

Figure 12 shows tracking results of a fish sequence. The sequence contains noise, background clutter and fish size changes. The jogging sequence (Figure 13) was tracked in conjunction with Kalman filtering (Kalman (1960)) to successfully track through short-term total occlusions.



(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.

Fig. 11. Face sequence. Montages of extracted results from 90 consecutive frames for different noise levels.

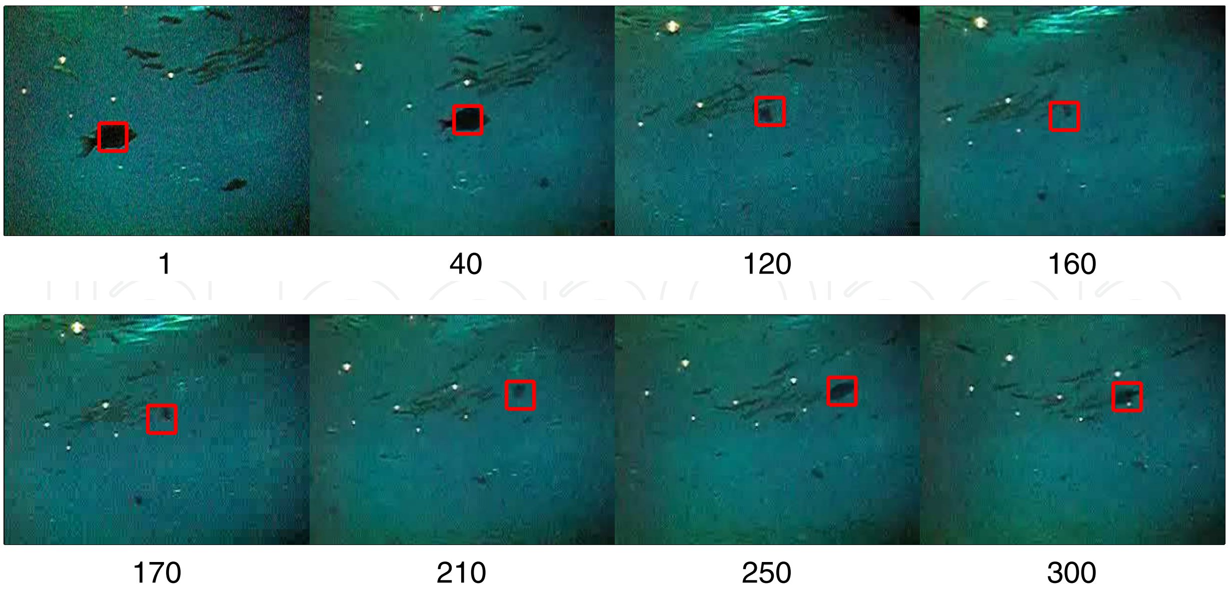


Fig. 12. Fish Sequence.

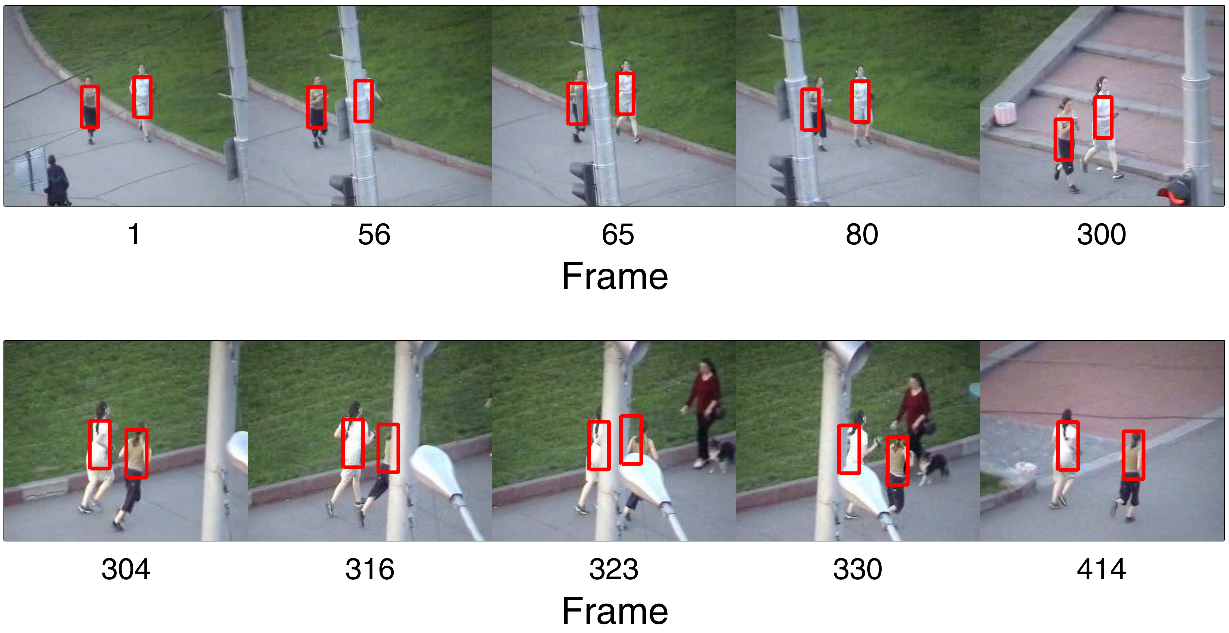


Fig. 13. Jogging sequence.

7. Conclusion

This chapter presented a novel density comparison method, given two sets of points sampled from two distributions. The method does not require explicit density estimation as an intermediate step. Instead it works directly on the data points to compute the similarity measure. The proposed similarity measure is robust to noise and outliers. 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.

8. References

Birchfield, S. & Rangarajan, S. (2005). Spatiograms versus histograms for region-based tracking, *IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp. 1158–1163.

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

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

Elgammal, A., Duraiswami, R. & Davis, L. (2003). Probabilistic tracking in joint feature-spatial spaces, *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 781–788.

Freedman, D. & Zhang, T. (2004). Active contours for tracking distributions, *IEEE Transactions on Image Processing* 13(4).

Girolami, M. (2002). Orthogonal series density estimation and the kernel eigenvalue problem, *Neural Computation.* 14(3): 669–688.

Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B. & Smola, A. (2007). A kernel method for the two-sample problem, *Technical Report 157*, Max Planck Institute.

Hager, G., Dewan, M. & Stewart, C. (2004). Multiple kernel tracking with SSD, *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 790–797.

Huber, P., Ronchetti, E. & MyiLibrary (1981). *Robust statistics*, Vol. 1, Wiley Online Library.

Kalman, R. (1960). A new approach to linear filtering and prediction problems, *Journal of Basic Engineering* 82(1): 35–45.

Kim, J. & Scott, C. (2008). Robust kernel density estimation, *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3381–3384.

Leventon, M. (2002). *Statistical models in medical image analysis*, PhD thesis, Massachusetts Institute of Technology.

Porikli, F., Tuzel, O. & Meer, P. (2006). Covariance tracking using model update based means on Riemannian manifolds, *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 728–735.

Rathi, Y., Dambreville, S. & Tannenbaum, A. (2006). Statistical shape analysis using kernel PCA, *Proceedings of SPIE*, Vol. 6064, pp. 425–432.

Rathi, Y., Malcolm, J. & Tannenbaum, A. (2006). Seeing the unseen: Segmenting with distributions, *International Conference on Signal and Image Processing*.

Schölkopf, B. & Smola, A. (2001). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, The MIT Press.

Scholköpf, B., Smola, A. & Muller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem, *Neural Computation* pp. 1299–1319.

Smola, A., Gretton, A., Song, L. & Schölkopf, B. (2007). A Hilbert space embedding for distributions, *Lecture Notes in Computer Science* .

Yang, C. & R. Duraiswami, L. D. (2005). Efficient mean-shift tracking via a new similarity measure, *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 176–183.

Principal Component Analysis
Edited by Dr. Parinya Sanguansat

ISBN 978-953-51-0195-6
Hard cover, 300 pages
Publisher InTech
Published online 02, March, 2012
Published in print edition March, 2012

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821