Similarity Measure for Nonparametric Kernel Density Based Object Tracking

Changjiang Yang, Ramani Duraiswami and Larry Davis
Department of Computer Science, University of Maryland, College Park, MD 20742
{yangcj,ramani,lsd}@umiacs.umd.edu

Abstract

An object tracking algorithm using a novel simple symmetric similarity function between spatially-smoothed kernel-density estimates of the model and target distributions is proposed and tested. The similarity measure is based on the expectation of the density estimates over the model or target images. The density is estimated using radial-basis kernel functions that measure the affinity between points and provide a better outlier rejection property. The mean-shift algorithm is used to track objects by iteratively maximizing this similarity function. To alleviate the quadratic complexity of the density estimation, we employ Gaussian kernels and the fast Gauss transform to reduce the computations to linear order. This leads to a very efficient and robust nonparametric tracking algorithm. The proposed algorithm is tested with several image sequences and shown to achieve robust and reliable real-time tracking. Several sequences are placed at http://www.cs.umd.edu/users/yangcj/node3.html.

1 Introduction

Object tracking is a common task in computer vision. Many tracking algorithms have been proposed and implemented to overcome difficulties that arise from noise, occlusion, clutter, and changes in the foreground objects or in the background environment [9, 1, 15, 3]. Many of these approaches employ a statistical description of the target. The tracked object can be described in the feature space by its probability density function (*pdf*) using either parametric or nonparametric approaches. In general, the feature space is complex and dynamic, so the nonparametric methods are preferable, and there has been increasing interest in using them [3, 5].

Tracking algorithms use measures of "similarity" or "distance" between the two *pdfs*. The Kullback-Leibler divergence, Bhattacharyya distance and other information-theoretic similarity measures are commonly employed to measure the similarity [3, 5], since they are capable of describing the complex nonlinear relations between frames. Such similarity measures are widely used in other areas such as image registration, content-based retrieval, and video indexing [14, 11]. However, these information-theoretic measures are limited to low-dimensional feature spaces because they are numerically unstable and computationally expensive in high dimensions when computed using the nonparametric sample based techniques, as recently observed in [11].

To overcome the flaws of the information-theoretic similarity measures in higher dimensions, Hero et al proposed estimating them using entropic graphs, specifically, the minimal spanning tree (MST) [7, 11]. However, the time complexity for finding the MST is of order $O(N^2 \log N)$ and the storage is $O(N^2)$, where N is the number of vertices in the graph. Such superquadratic complexity is too expensive for realtime object tracking.

We address these difficulties by presenting a tracking algorithm that uses a simple symmetric similarity function between kernel density estimates of the model and target distributions. The similarity measure is symmetric and is the expectation of the density estimates centered on the model (target) image over the target (model) image. To meet the realtime requirement of object tracking, we employ Gaussian kernels and the improved fast Gauss transform (FGT) [16] to reduce the computations to linear order. In our formulation we use the joint spatial-feature formulation of [5], and consider both feature vectors and pixel locations as probabilistic random variables. The density is estimated in the joint feature-spatial space using Gaussian kernel functions which measure the affinity between points and provide a better outlier rejection property. The joint feature-spatial spaces impose a probabilistic spatial constraint on the tracked region and provide an accurate representation of the tracked objects. Due to its simplicity and robustness, the popular mean shift algorithm [3] is used to track objects by iteratively maximizing this similarity function.

2 Similarity Measure Between Distributions

Suppose we are given two images, with one designated as the "model image" that includes the tracked objects, while the other is the "target image" in which we need to find the objects. The sample points in the model image are denoted by $I_x = \{\mathbf{x}_i, \mathbf{u}_i\}_{i=1}^N$, where $\mathbf{x}_i$ is the 2D coordinates and $\mathbf{u}_i$ is the corresponding feature vector (e.g., the red, green and blue colors of sample points). The sample points in the target image are $I_y = \{\mathbf{y}_j, \mathbf{v}_j\}_{j=1}^M$, encoding the 2D coordinates and the corresponding feature vector.

We describe the targets in the joint feature-spatial space [5]. Given the sample points and the kernel function $k(x)$, the *pdf* of the model image in the joint feature-spatial space can be estimated using kernel density estimation [12] as

$$\hat{p}_x(\mathbf{x}, \mathbf{u}) = \frac{1}{N} \sum_{i=1}^{N} w(\left\| \frac{\mathbf{x} - \mathbf{x}_i}{\sigma} \right\|^2) k(\left\| \frac{\mathbf{u} - \mathbf{u}_i}{h} \right\|^2). \tag{1}$$

where σ and h are the bandwidths in the spatial and feature spaces. Similarly we can estimate the *pdf* of the target image. The benefit of the joint feature-spatial space is that it combines the feature and spatial information and provides good discriminant capability.

The existing mean-shift trackers use different information-theoretic measures such as the Kullback-Leibler divergence[5] and the Bhattacharyya distance [3] to measure the affinity between distributions. Instead of evaluating the information-theoretic measures from the estimated *pdf*, we directly define the similarity between two distributions as the expectation of the density estimates over the model or target image. Given two distributions with samples $I_x = \{\mathbf{x}_i, \mathbf{u}_i\}_{i=1}^N$ and $I_y = \{\mathbf{y}_j, \mathbf{v}_j\}_{j=1}^M$, where the center of sample points in the model image is $\mathbf{x}_*$, and the current center of the target points is $\mathbf{y}$, the similarity between I_x and I_y in the joint feature-spatial space is

$$J(I_x, I_y) = \frac{1}{M} \sum_{j=1}^{M} \hat{p}_x(\mathbf{y}_j, \mathbf{v_j}) = \frac{1}{MN} \sum_{i=1}^{N} \sum_{j=1}^{M} w(\left\| \frac{\mathbf{x}_i - \mathbf{y}_j}{\sigma} \right\|^2) k(\left\| \frac{\mathbf{u}_i - \mathbf{v}_j}{h} \right\|^2). \tag{2}$$

The similarity measure (2) is symmetric and bounded by zero and one, but violates the triangle inequality which means the similarity measure is non-metric. Often distance functions that are robust to outliers or to noise disobey the triangle inequality [10]. While our similarity function (2) is non-metric, it can be shown that its negative natural logarithm

$$L(I_x, I_y) = -\log J(I_x, I_y) \tag{3}$$

is a probabilistic distance, provided we have sufficient samples [4].

The similarity measure (2) is directly computed over the entire sample point sets. The affinities between all pairs of sample points are considered based on their distances, so that

exact correspondence is not necessary. This is more robust than the popular elementary tracking techniques of template matching or sum of squared differences (SSD). Notice that since the sample points are sparse in the high dimensional feature space, it is difficult to get an accurate density estimation or histogram which makes similarity measures such as Kullback-Leibler divergence and Bhattacharyya coefficient computationally unstable.

2.1 Comparing the Proposed Similarity Measure with Other Measures

To compare our similarity measure with the Bhattacharyya distance and Kullback-Leibler distance, we perform simulations. The Bhattacharyya distance is defined as

$$B(I_x, I_y) = \sqrt{1 - \rho(p_x, p_y)}, \quad \rho(p_x, p_y) = \sum_{u=1}^{m} \sqrt{\hat{p}_x(u)\hat{p}_y(u)}. \tag{4}$$

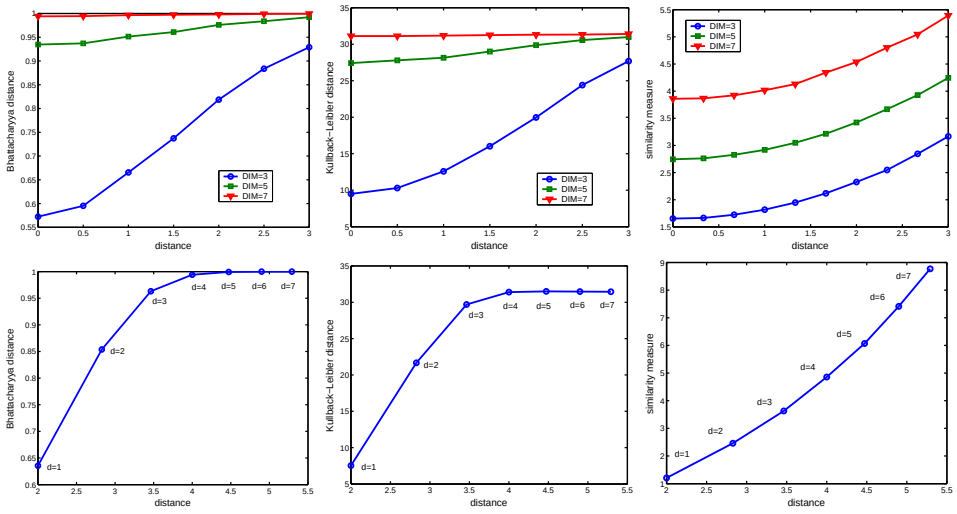
The Kullback-Leibler distance between two distribution is defined as

$$D(I_x, I_y) = \int p_y(u) \log \frac{p_y(u)}{p_x(u)} du. \tag{5}$$

We first generate two multivariate Gaussian distributions

$$p_x(u) \sim G(\mu_1, I), \quad p_y(u) \sim G(\mu_2, I),$$

where $\mu_1 = (\mu, 0, \ldots, 0)$, $\mu_2 = -\mu_1$, μ varies from 0 to 1.5, and I is an identity covariance matrix. For dimensions 3, 5 and 7, 100 samples were generated for each distribution and 100 repetitions were run. The estimated distances between two distributions *w.r.t.* the ground truth are displayed in Figure 1 (Top row). We also generate two distributions in dimensions between 1 and 7 (the histogram based methods run out of memory on our computer beyond dimension 7). The centers are located at $\mu_1 = (1, 1, \ldots, 1)$ and $\mu_2 = -\mu_1$. The estimated distances between two distributions *w.r.t.* the ground truth are displayed in Figure 1 (Bottom row).



(a) (b) (c)

Figure 1: The distances between two distributions estimated from samples using: (a) Bhattacharyya distance, (b) Kullback-Leibler distance, and (c) our similarity measure, *w.r.t.* the ground truth. *Top row*: the simulations are repeated 100 times for dimensions 3, 5 and 7, where the distances between the centers of two Gaussian distributions vary from 0 to 3. *Bottom row*: the simulations are repeated 100 times for each dimension between 1 and 7, where the centers of the Gaussian distributions are located at $(1, 1, \ldots, 1)$ and $(-1, -1, \ldots, -1)$. All simulations use an identity covariance matrix.

The simulations indicate that the Bhattacharyya and Kullback-Leibler distances computed using the sample derived distributions are inaccurate in higher dimensions and the computations in higher dimensions are instable. In contrast, our similarity measure are more accurate and more stable in both lower and higher dimensions. Also, the proposed similarity measure can be computed with a computing time that is linear in the number of pixels using the improved fast Gauss transform (see section 4), where as the information theoretic measures have quadratic or higher complexities.

3 Similarity-Based Tracking Algorithms

We first derive the tracking algorithm assuming that the motion between frames is a pure translation, and generalize the motion later. Let $\mathbf{x}_*$ be the center of the model image and $\mathbf{y}$ be the center of target image, then $\mathbf{y} = \mathbf{x}_* + \mathbf{p}$, $\mathbf{p}$ is the translation, then equation (2) becomes

$$J(I_x, I_y) = \frac{1}{MN} \sum_{i=1}^{N} \sum_{j=1}^{M} w(\left\| \frac{(\mathbf{y}_j - \mathbf{y}) - (\mathbf{x}_i - \mathbf{x}_*)}{\sigma} \right\|^2) k(\left\| \frac{\mathbf{u}_i - \mathbf{v}_j}{h} \right\|^2). \tag{6}$$

Once we have the similarity measure between the model image and target image, we can find the target location in the target image by maximizing the similarity measure (6) or equivalently minimizing the distance (3) with respect to $\mathbf{y}$. We use the mean-shift algorithm [2] which has already proved successful in many computer vision applications [2, 3].

The gradient of the distance function (3) with respect to the vector $\mathbf{y}$ is

$$\nabla L(\mathbf{y}) = -\frac{\nabla J(\mathbf{y})}{J(\mathbf{y})}, \tag{7}$$

where

$$\nabla J(\mathbf{y}) = \frac{2}{MN\sigma^2} \sum_{i=1}^{N} \sum_{j=1}^{M} (\Delta\mathbf{x}_i - \Delta\mathbf{y}_j)\, k_{ij} w'(\left\| \frac{\Delta\mathbf{x}_i - \Delta\mathbf{y}_j}{\sigma} \right\|^2), \tag{8}$$

and $k_{ij} = k(\left\| \frac{\mathbf{u}_i - \mathbf{v}_j}{h} \right\|^2)$, $\Delta\mathbf{x}_i = \mathbf{x}_i - \mathbf{x}_*$, $\Delta\mathbf{y}_j = \mathbf{y}_j - \mathbf{y}$.

The *mean shift* of the smoothed similarity function $J(\mathbf{y})$ is

$$\nabla L(\mathbf{y}) = \frac{\sum_{i=1}^{N} \sum_{j=1}^{M} (\mathbf{y}_j - \mathbf{x}_i) k_{ij} g(\left\| \frac{\Delta\mathbf{x}_i - \Delta\mathbf{y}_j}{\sigma} \right\|^2)}{\sum_{i=1}^{N} \sum_{j=1}^{M} k_{ij} g(\left\| \frac{\Delta\mathbf{x}_i - \Delta\mathbf{y}_j}{\sigma} \right\|^2)} - \mathbf{y} + \mathbf{x}_*, \tag{9}$$

where $g(x) = -w'(x)$ is also the profile of the RBF kernel, which is Gaussian in our case.

Given the sample points $\{\mathbf{x}_i, \mathbf{u}_i\}_{i=1}^N$ centered at $\mathbf{x}_*$ in the model image, and $\{\mathbf{y}_j, \mathbf{v}_j\}_{j=1}^M$ centered at the current position $\hat{\mathbf{y}}_0$ in the target image, the object tracking based on the mean-shift algorithm is an iterative procedure which recursively moves the current position $\hat{\mathbf{y}}_0$ to the new position $\hat{\mathbf{y}}_1$ until reaching the density mode according to

$$\hat{\mathbf{y}}_1 = \frac{\sum_{i=1}^{N} \sum_{j=1}^{M} \mathbf{y}_j k_{ij} g_{ij}}{\sum_{i=1}^{N} \sum_{j=1}^{M} k_{ij} g_{ij}} - \frac{\sum_{i=1}^{N} \sum_{j=1}^{M} \mathbf{x}_i k_{ij} g_{ij}}{\sum_{i=1}^{N} \sum_{j=1}^{M} k_{ij} g_{ij}} + \mathbf{x}_*. \tag{10}$$

where $g_{ij} = g(\left\| \frac{(\mathbf{x}_i - \mathbf{x}_*) - (\mathbf{y}_j - \hat{\mathbf{y}}_0)}{\sigma} \right\|^2)$.

If the size of the target changes during the tracking, we can model the motion model as translation plus scaling, then the similarity measure (2) becomes:

$$J(I_x, I_y) = \frac{1}{MN} \sum_{i=1}^{N} \sum_{j=1}^{M} w(\left\| \frac{\sqrt{s}\Delta\mathbf{x}_i}{\sigma} - \frac{\Delta\mathbf{y}_j}{\sqrt{s}\sigma} \right\|^2) k(\left\| \frac{\mathbf{u}_i - \mathbf{v}_j}{h} \right\|^2). \tag{11}$$

where s is the scaling factor accounting for the size changes of target between frames [13].

Similarly, we obtain the updating rules for the mean-shift tracking by differentiating (11) with respect to $\mathbf{p}$ and s:

$$\hat{\mathbf{y}}_1 = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}\mathbf{y}_j k_{ij} g_{ij}}{\sum_{i=1}^{N}\sum_{j=1}^{M} k_{ij} g_{ij}} - \frac{s\sum_{i=1}^{N}\sum_{j=1}^{M}\mathbf{x}_i k_{ij} g_{ij}}{\sigma\sum_{i=1}^{N}\sum_{j=1}^{M} k_{ij} g_{ij}} + \frac{s}{\sigma}\mathbf{x}_*.$$

$$\hat{s}_1 = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}(1 + \|\frac{\Delta\mathbf{y}_j}{\hat{s}_0}\|^2 - \|\frac{\Delta\mathbf{x}_i}{\sigma}\|^2) k_{ij} g_{ij}}{\sum_{i=1}^{N}\sum_{j=1}^{M} k_{ij} g_{ij}}\hat{s}_0, \tag{12}$$

where $g_{ij} = g(\left\|\frac{\mathbf{x}_i - \mathbf{x}_*}{\sigma} - \frac{\mathbf{y}_j - \hat{\mathbf{y}}_0}{s}\right\|^2)$. More complex expressions have been derived for the cases of an affine and a general motion model, and will be presented elsewhere.

4 Speedup using the Improved FGT

The computational complexity for the direct evaluation of the similarity meansure (2) is $O(MN)$, and $O(PMN)$ for the above tracking algorithm, where P is the average number of iterations per frame, M and N are the number of sample points in target and model images respectively. Typically the average number of iterations per frame P is less than ten and $M \approx N$. Then the order of the computational complexity is quadratic, which still is too expensive for the realtime tracking.

If the Gaussian kernel is used, we can apply the fast Gauss transform [6, 16] to the tracking algorithm to reduce its computational complexity from quadratic order to linear order. Since the derivative of Gaussian kernel is still a Gaussian, the mean shift based object tracking with the Gaussian kernel is

$$\hat{\mathbf{y}}_1 = \frac{\sum_{j=1}^{M}\mathbf{y}_j f(\mathbf{y}_j)}{\sum_{j=1}^{M} f(\mathbf{y}_j)} - \frac{\sum_{i=1}^{N}\mathbf{x}_i f(\mathbf{x}_i)}{\sum_{i=1}^{N} f(\mathbf{x}_i)} + \mathbf{x}_*, \tag{13}$$

where

$$f(\mathbf{x}_i) = \sum_{j=1}^{M} e^{-\|\mathbf{u}_i - \mathbf{v}_j\|^2/h^2 - \|\Delta\mathbf{y}_j - \Delta\mathbf{x}_i\|^2/\sigma^2}, f(\mathbf{y}_j) = \sum_{i=1}^{N} e^{-\|\mathbf{u}_i - \mathbf{v}_j\|^2/h^2 - \|\Delta\mathbf{y}_j - \Delta\mathbf{x}_i\|^2/\sigma^2} \tag{14}$$

are the *discrete Gauss transform* of $\mathbf{x}_i$ and $\mathbf{y}_j$. The similarity measure is

$$J(I_x, I_y) = \sum_{i=1}^{N} f(\mathbf{x}_i) = \sum_{j=1}^{M} f(\mathbf{y}_j).$$

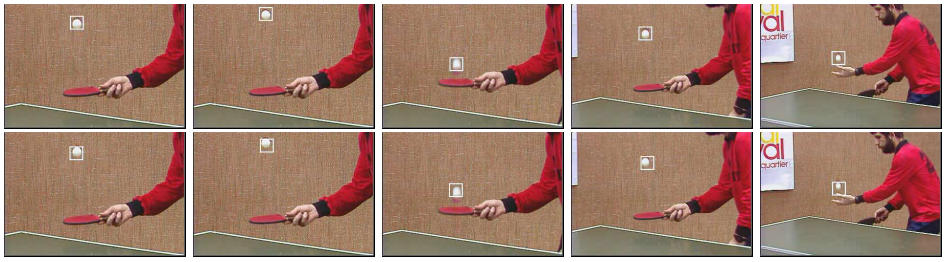
The computational complexity of a direct evaluation of the discrete Gauss transform (14) requires $O(MN)$ operations. In low-dimensional spaces, the computational complexity has been reduced by Greengard and Strain [6] to $C\cdot(M+N)$ using the fast Gauss transform, where the constant factor C depends on the precision required and dimensionality.

Although the fast Gauss transform achieved great success in low dimensions, the performance in higher dimensions is poor. To achieve a better performance, we proposed the improved FGT. The improved FGT is very efficient even in higher dimensions and achieve real-time performance for the object tracking. More details about the improved FGT can be found in [16]. Recently entropic graphs such as the minimal spanning tree have been used to estimate the information-theoretic similarity measures for image registration [11]. The standard algorithms for the MST is $O(N^2 \log N)$. The acceleration of the MST relies on the nearest neighbor searching which itself is difficult and complicated in higher dimensions and is an active research topic [8].

5 Experimental Results

We present some real-time object tracking results using the proposed algorithm. In the first experiment, the RGB color space along with the 2D spatial coordinates is used as the joint feature-spatial space. In the second one, the RGB color space, and 2D spatial coordinates plus 2D image gradient is used. The Gaussian kernel is used in all the experiments. The algorithm is implemented in C++ with Matlab interface and runs on a 900MHZ PIII PC.

The first experiment uses the *Ball* sequence [3]. If we blindly apply the tracking algorithm, it will either track the background if a large region is used, or lose the ball if the tracking region is small and the movement is large. We utilize the background information and assume a mask about the tracked object is available. We initialize the model with a region in frame 3 of size 48×48. The bandwidths are $(h, \sigma) = (18, 12)$. We only keep the foreground pixels in the model and run the tracking algorithm. The algorithm reliably and accurately tracks the ball with average number of iteration 2.7679 and average processing time per frame $0.0169s$. In contrast, to successfully track this sequence, in [3] a background-weighted histogram was employed. The tracking results shown in Figure 2 are more accurate than those in [3]. The number of iterations and sums of squared differences between the model image and the tracked images are shown in Figure 3. The results of our method are more accurate and number of iterations is smaller than the method using the Bhattacharyya distance.



Frame 6 Frames 20 Frame 27 Frame 36 Frame 53

Figure 2: Tracking results of the *Ball* sequence using (top row) our similarity measure and (second row) Bhattacharyya distance.

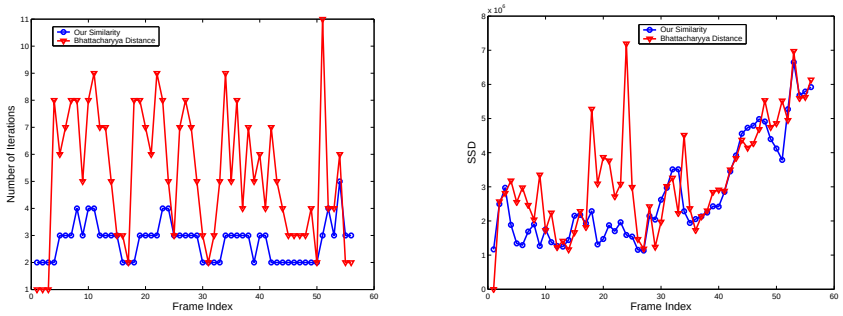


Figure 3: The number of iterations (*left*) and sums of squared differences (*right*) *w.r.t.* the frame index for the *Ball* sequence using our similarity measure and Bhattacharyya distance.

In the second experiment a more complex clip is tested. In order to track a face with changing appearance and complex background, we use both the RGB color space and 2D image gradients as features. The image gradients are the horizontal and vertical image gradients of the grayscale image obtained using the Sobel operator. We initialize the model

with a region in frame 0 of size 24×24. The bandwidths are $(h, \sigma) = (25, 12)$. The average number of iterations per frame is 2.1414 and average processing time per frame is $0.0044s$. The algorithm reliably tracks the face and results are shown in Figure 4.

Figure 4: Tracking results of the *Walking* sequence. Frames 4, 19, 50, 99, 166 and 187 are displayed.

In the third sequence *Pedestrian*, the size of the pedestrian changes between frames. We apply the mean-shift tracking with the translation plus scaling to the sequence and the results are shown in Figure 5. The positions and the size of the pedestrian are correctly recovered by our algorithm.

Figure 5: Tracking results of the *Pedestrian* sequence. Frames 3, 17, 31, 45, 60 and 78 are displayed.

6 Discussion and Conclusions

We proposed a novel, simple symmetric similarity function between spatially-smoothed kernel-density estimates of the model and target distributions for object tracking. The similarity measure is based on the expectation of the density estimates over the model or target image. The RBF kernel functions are used to measure the affinity between points and provide a better outlier rejection property. To track the objects, the similarity function is maximized using the mean-shift algorithm to iteratively find the local mode of the function.

Since the similarity measure is an expectation taken over all pairs of the pixel between two distributions, the computational complexity is quadratic. To alleviate the quadratic

complexity, we employ Gaussian kernels and the fast Gauss transform to reduce the computations to linear order. This leads to a very efficient and robust nonparametric tracking algorithm. It also very convenient for integration of the background information and generalization to high dimensional feature space.

References

[1] Michael J. Black and Allan D. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *Int'l Journal of Computer Vision*, 26(1):63–84, 1998.

[2] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(5):603 – 619, May 2002.

[3] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Kernel-based object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(5):564–577, May 2003.

[4] Pierre Devijver and Josef Kittler. *Pattern Recognition: A Statistical Approach*. Prentice-Hall International, London, 1982.

[5] Ahmed Elgammal, Ramani Duraiswami, and Larry Davis. Probabilistic tracking in joint feature-spatial spaces. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, volume I, pages 781–788, Madison, WI, 2003.

[6] Leslie Greengard and John Strain. The fast Gauss transform. *SIAM J. Sci. Statist. Comput.*, 12(1):79–94, 1991.

[7] Alfred Hero, Bing Ma, Olivier Michel, and John Gorman. Applications of entropic spanning graphs. *IEEE Signal Proc. Magazine*, 19(5):85–95, September 2002.

[8] Piotr Indyk. Nearest neighbors in high-dimensional spaces. In Jacob E. Goodman and Joseph O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, chapter 39. CRC Press, 2rd edition, 2004.

[9] Michael Isard and Andrew Blake. Contour tracking by stochastic propagation of conditional density. In *Proc. European Conf. Computer Vision*, pages 343–356, Cambridge, UK, 1996.

[10] David Jacobs, Daphna Weinshall, and Yoram Gdalyahu. Class representation and image retrieval with non-metric distances. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(6):583–600, 2000.

[11] Huzefa Neemuchwala, Alfred Hero, and Paul Carson. Image matching using alpha-entropy measures and entropic graphs. *to appear in European Journal of Signal Processing*, 2004.

[12] Emanuel Parzen. On estimation of a probability density function and mode. *Ann. Math. Stat.*, 33(3):1065–1076, 1962.

[13] Anand Rangarajan, Haili Chui, and Fred L. Bookstein. The softassign procrustes matching algorithm. In *Proc. of the 15th International Conference on Information Processing in Medical Imaging*, pages 29–42. Springer-Verlag, 1997.

[14] Paul Viola and William M. Wells III. Alignment by maximization of mutual information. *Int'l Journal of Computer Vision*, 24(2):137–154, 1997.

[15] Oliver Williams, Andrew Blake, and Roberto Cipolla. A sparse probabilistic learning algorithm for real-time tracking. In *Proc. Int'l Conf. Computer Vision*, pages 353–360, Nice, France, 2003.

[16] Changjiang Yang, Ramani Duraiswami, Nail Gumerov, and Larry Davis. Improved fast Gauss transform and efficient kernel density estimation. In *Proc. Int'l Conf. Computer Vision*, pages 464–471, Nice, France, 2003.