Efficient Mean-Shift Tracking via a New Similarity Measure

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

Abstract

The mean shift algorithm has achieved considerable success in object tracking due to its simplicity and robustness. It finds local minima of a similarity measure between the color histograms or kernel density estimates of the model and target image. The most typically used similarity measures are the Bhattacharyya coefficient or the Kullback-Leibler divergence. In practice, these approaches face three difficulties. First, the spatial information of the target is lost when the color histogram is employed, which precludes the application of more elaborate motion models. Second, the classical similarity measures are not very discriminative. Third, the sample-based classical similarity measures require a calculation that is quadratic in the number of samples, making real-time performance difficult. To deal with these difficulties we propose a new, simple-to-compute and more discriminative similarity measure in spatial-feature spaces. The new similarity measure allows the mean shift algorithm to track more general motion models in an integrated way. To reduce the complexity of the computation to linear order we employ the recently proposed improved fast Gauss transform. This leads to a very efficient and robust nonparametric spatial-feature tracking algorithm. The algorithm is tested on several image sequences and shown to achieve robust and reliable frame-rate tracking.

Keywords: Mean-shift algorithm, Improved fast Gauss transform, Similarity measure, Spatial-feature tracking.

1 Introduction

The goal of object tracking is to find the targets between the consecutive frames in image sequences. 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. Gradient based methods align tracked regions between successive frames by minimizing a cost function using various gradient descent techniques [27, 19]. Feature-based approaches extract features (such as intensity, colors, edges, contours) and use them to establish correspondence between model images and target images [25, 16, 10]. Knowledge-based tracking algorithms incorporate *a priori* information about the tracked objects to obtain representations such as projected shape, skin complexion, body blobs, kinematic skeletons and silhouettes [36, 34, 5, 30, 7]. Learning-based approaches apply pattern recognition algorithms to learn the objects either in the eigenspace or in the kernel space, and then search for targets in image sequences [4, 1, 33].

Among the various tracking algorithms, mean shift tracking algorithms have recently become popular due to their simplicity and robustness [5, 10, 8, 20]. The mean shift algorithm was originally invented by Fukunaga and Hostetler [17] for data clustering, which they called a "valley-seeking procedure". It was first introduced into the image processing community several years ago by Cheng [6]. Recently Comaniciu and Meer [9, 10] successfully applied it to image segmentation and tracking.

In these mean shift tracking algorithms, a color histogram is used to describe the target region. The Kullback-Leibler divergence, Bhattacharyya coefficient and other information-theoretic similarity measures are commonly employed to measure the similarity between the template (or model) region and the current target region [10, 15, 31]. Tracking is accomplished by iteratively finding the local minima of the distance measure functions using the mean shift algorithm.

However, the mean shift tracking algorithms using histograms have several serious defects. First, the spatial information of the targets is lost, which precludes the application of more general motion models. The

Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)
1063-6919/05 $20.00 © 2005 **IEEE**

mean shift trackers must resort to using separate computational mechanisms such as second order moment information [5] or scale space approaches [8] to recover the scale and other information of the targets. Second, the classical information-theoretic similarity measures such as the Bhattacharyya coefficient [10] or the Kullback-Leibler divergence [15] are not very discriminative, especially in higher dimensions. Third, the sample-based classical similarity measures require a calculation that is quadratic in the number of samples, which makes it difficult to meet the real-time requirement in object tracking [15].

We address these difficulties by presenting a tracking algorithm that uses a new simple symmetric similarity function between kernel density estimates of the template and target distributions in a joint spatial-feature space. Given sufficient samples, kernel density estimation works well both in low and high dimensions. The similarity measure is symmetric and is the expectation of the density estimates centered on the model (target) image over the target (model) image. Using this similarity measure, we can derive a mean shift tracking algorithm with general motion models. Unlike the mean shift tracking algorithm using the histogram-based similarity measures, our method treats the location and other deformation in an integrated way and tracks the deformation incrementally. Compared with the information-theoretic similarity measures, our similarity measure is more robust and more discriminative.

Since the similarity measure sums over every pair of the pixel between template image and the target image, the computational load is quadratic order which is too slow for the realtime tracking. To meet the realtime requirement of object tracking, we employ Gaussian kernels and the improved fast Gauss transform (IFGT) [35] to reduce the computations to linear order.

Section 2 defines the similarity measures and the problems, and presents results on synthetic data, discusses the problem the classical similarity measures have and illustrates them. Section 3 describes the use of this similarity measure and derives expressions for feature-spatial tracking for the cases of translational, scaled translational, and affine motion. Section 4 describes the speedup of computing the similarity measure using the improved FGT. Section 5 gives some experimental results and Section 6 concludes the paper.

2 Similarity Measure Between Distributions

2.1 Classical Similarity Measures

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 an object 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 [15]. Given the sample points and the kernel function $k(x)$, the p.d.f. of the object in the model image can be estimated using kernel density estimation [28] as

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

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

Existing mean-shift trackers use different information-theoretic measures such as the Kullback-Leibler divergence [15] and the Bhattacharyya distance [10] to measure the affinity between distributions. The Bhattacharyya distance is

$$\begin{aligned} B(I_x, I_y) &= \sqrt{1-\rho(p_x, p_y)}, \\ \rho(p_x, p_y) &= \int \sqrt{\hat{p}_x(u)\hat{p}_y(u)}\, du. \end{aligned} \tag{2}$$

The Kullback-Leibler distance between two distribution is defined as

$$D(I_x, I_y) = \int p_y(u) \log \frac{p_y(u)}{p_x(u)} du. \tag{3}$$

For future reference we note that the straightforward evaluation of these measures requires two sequential $O(N^2)$ operations, if we assume that the p.d.f. is first evaluated and then the summation (or integral) evaluated on the sample pairs.

These measures are widely used in other areas such as image registration [31], content-based retrieval, and video indexing, and this lack of discriminative ability has also been observed in [22]. To overcome the flaws of

Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)
1063-6919/05 $20.00 © 2005 **IEEE**

the information-theoretic similarity measures in higher dimensions, Hero et al [22] proposed estimating them using entropic graphs, specifically, the minimal spanning tree (MST). While this procedure fixes the discrimination problem, 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, and is again too expensive for realtime object tracking.

Some authors move to a parametric representation to avoid the difficulties of inefficient computation and have the opportunity to use analytical similarity expressions [21]. However, even the parametric forms of the classical similarity measures have similar problems [12].

2.2 New Sample Based Similarity Measure

Instead of evaluating the information-theoretic measures from the estimated p.d.f., 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

$$J(I_x, I_y) = \frac{1}{M} \sum_{j=1}^{M} \hat{p}_x(\mathbf{y}_j, \mathbf{v_j}) \tag{4}$$
$$= \frac{1}{MN} \sum_{i=1}^{N} \sum_{j=1}^{M} w \left(\left| \frac{\mathbf{x}_i - \mathbf{y}_j}{\sigma} \right|^2 \right) k \left(\left| \frac{\mathbf{u}_i - \mathbf{v}_j}{h} \right|^2 \right),$$

where $k(x)$ is a RBF kernel function. This similarity measure is based on the average separation criterion in cluster analysis [13, 32], except that we replace the distance with a kernelized one. The kernel mapping limits the effect of outliers and improves the robustness. Similar measures are used in [3] for unsupervised clustering of images of 3D objects. The similarity function (4) can be interpreted as the expectation of the spatially-smoothed density estimates over the model image.

The similarity measure (4) is symmetric and bounded by zero and one, but violates the triangle inequality which means it is non-metric. Often distance functions that are robust to outliers or to noise disobey the triangle inequality [26]. While our similarity function (4) is non-metric, it can be shown that its negative natural logarithm

$$L(I_x, I_y) = -\log J(I_x, I_y) \tag{5}$$

is a probabilistic distance, provided we have sufficient samples [11].

2.3 Comparing the Similarity Measures

The similarity measure (4) is directly computed over the entire sample point sets. The affinities between all pairs of sample points are considered based on their distances, so exact correspondence is not necessary. This is more robust than the popular elementary tracking techniques of template matching or sum of squared differences (SSD). 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.

First we show that the proposed similarity method is more discriminating than Bhattacharyya coefficient or Kullback-Leibler divergence on synthetic data. We randomly sample two 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, 1000 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 (Left column). We also generate two distributions in dimensions between 1 and 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 (Right column).

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 the sense that repeated computations using different samples yields varying results. Such phenomenon has already been observed in the past and recently [12, 22]. The variance of the Bhattacharyya coefficient estimate increases as the dimension increases [12]. In contrast, our similarity measure is accurate and more stable in both lower and higher dimensions.

As will be seen in Section 4, the present similarity measure has the further advantage that it can be computed in linear time in the number of pixels using the improved fast Gauss transform. In contrast the nonlinear information theoretic measures have a structure that does not permit use of the IFGT and consequently require at least quadratic complexity.

As mentioned previously, entropic graphs such as the minimal spanning tree have been used to estimate more discriminative information-theoretic simi-

larity measures for image registration [22]. 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 [24].

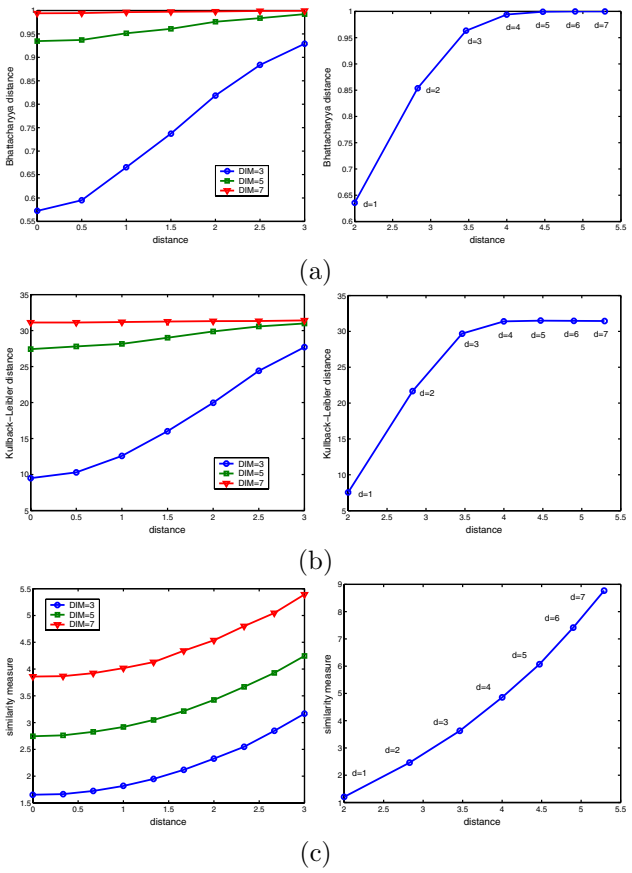

Figure 1. The distances between two distributions estimated from samples using: (a) Bhattacharyya coefficient, (b) Kullback-Leibler distance, and (c) our similarity measure, *w.r.t.* the ground truth. *Left column*: the simulations are repeated 100 times for dimensions 3, 5 and 7, where the distances between the centers of two Gaussian distributions vary from 0 to 3. *Right column*: 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.

3 Similarity-Based Mean-Shift Tracking

We first derive the tracking algorithm for the case the motion between frames is 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 (4) becomes

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

Once we have the similarity measure between the model and target images, we can find the target location in the target image by maximizing the similarity measure (6) or equivalently minimizing the distance (5) with respect to $\mathbf{y}$. We use the mean-shift algorithm [9] which has already proved successful in many computer vision applications [9, 10].

The gradient of the distance function (5) with respect to the vector $\mathbf{y}$ is

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

where

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

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

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

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

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

Given 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

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

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

Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)
1063-6919/05 $20.00 © 2005 **IEEE**

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

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

where s is the scaling factor accounting for the size changes of target between frames [29].

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

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

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

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

The updating rule (12) for translation is similar to the formula (10). The updating rule (12) for scaling essentially compares the second-order moments between the template and the target. it bears strong resemblance to the widely used scale estimate method in [23] which also compares the second-order moments. In contrast, the scaling updating rule proposed by Collins [8] employs the scale-space theory which could be confused by the multiple scales within the targets.

More complex expressions have been derived for the cases of a general motion model, and are presented in the appendix.

4 Speedup using the Improved FGT

The computational complexity for the direct evaluation of the similarity measure (4) 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 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 [18] to the tracking algorithm to reduce its computational complexity from quadratic order to linear order, as was done in . Since the derivative of Gaussian kernel is still a Gaussian, the mean shift based object tracking with the Gaussian kernel is

$$\hat{\mathbf{y}}_1 = \frac{\sum_{j=1}^{M}\mathbf{y}_j f(\mathbf{y}_j)}{\sum_{j=1}^{M} f(\mathbf{y}_j)} - \frac{\sum_{i=1}^{N}\mathbf{x}_i f(\mathbf{x}_i)}{\sum_{i=1}^{N} f(\mathbf{x}_i)} + \mathbf{x}_*, \quad (12)$$

where

$$f(\mathbf{x}_i) = \sum_{j=1}^{M} e^{-\|\mathbf{u}_i - \mathbf{v}_j\|^2/h^2 - \|\Delta\mathbf{y}_j - \Delta\mathbf{x}_i\|^2/\sigma^2}, \quad (13)$$

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

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 [18] to $C \cdot (M + N)$ using the fast Gauss transform, where the constant factor C depends on the precision required and dimensionality. This was applied to vision problems in [14], but it was also observed there that the algorithm did not work well in higher dimensions, since the constant factor in the asymptotic complexity grows exponentially with dimensions. To achieve real-time performance, we employ the improved FGT [35] to accelerate the evaluation of the similarity cost function.

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 [10]. 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 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 [10] a background-weighted histogram was employed. The tracking results shown in Figure 2 are more accurate than those in [10]. 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. This shows that the discriminating problems observed on synthetic data affect the simulations in practice as well.

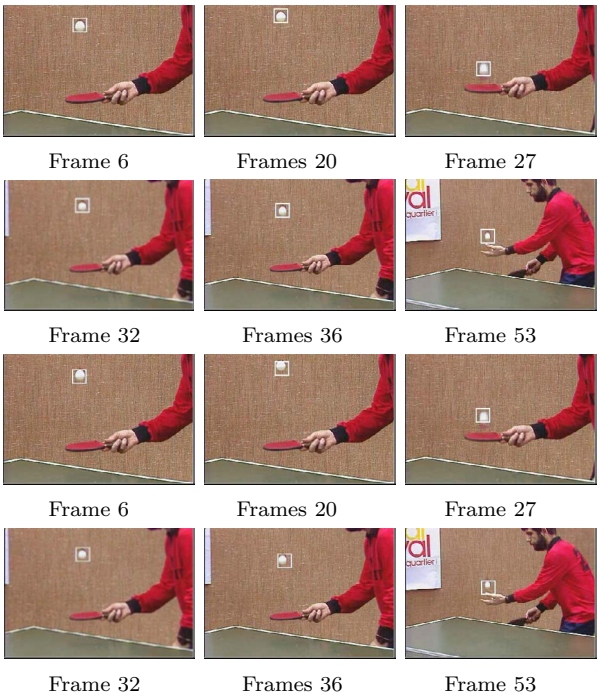

Figure 2. Tracking results of the *Ball* sequence using (top two rows) our similarity measure and (bottom two rows) Bhattacharyya coefficient.

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 grayscaling image obtained using the Sobel operator. We initialize the model with a region 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

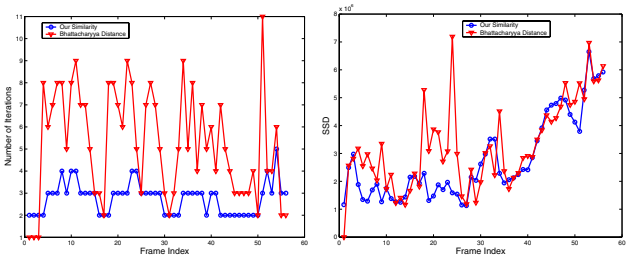

Figure 3. The number of iterations (*left*) and sums of squared differences (*right*) *w.r.t.* the frame index for the *Ball* sequence using our similarity measure and Bhattacharyya coefficient. The number of iterations using our similarity measure is much reduced.

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 improved 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 spaces.

Acknowledgements

We would like to thank Dr. Dorin Comaniciu for providing the the *Ball* sequence. We gratefully acknowledge the support of NSF grant IIS0205271 and DOD grant 2004H840200000.

Appendix

In this appendix, we will derive a tracking algorithm with the general geometric transformation $\mathbf{y} = \mathbf{W}(\mathbf{x};\mathbf{p})$, where $\mathbf{p}$ is the geometric deformation parameter vector.

The gradient of the distance function (4) with respect to the vector $\mathbf{p}$ is

$$\begin{aligned}
&\nabla J(\mathbf{p}) = \mathbf{G}_1(\mathbf{p}) + \mathbf{G}_2(\mathbf{p}) \\
= \quad & \frac{2}{MNh^2} \sum_{i=1}^{N} \sum_{j=1}^{M} \\
& k'\left(\|\frac{\mathbf{v}_j - \mathbf{u}_i}{h}\|^2\right) w\left(\|\frac{\mathbf{y}_j - \mathbf{x}_i}{\sigma}\|^2\right) \left[\nabla \mathbf{v}_j \frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]^T (\mathbf{v}_j - \mathbf{u}_i) \\
+ \quad & \frac{2}{MN\sigma^2} \sum_{i=1}^{N} \sum_{j=1}^{M} \\
& w'\left(\|\frac{\mathbf{y}_j - \mathbf{x}_i}{\sigma}\|^2\right) k\left(\|\frac{\mathbf{v}_j - \mathbf{u}_i}{h}\|^2\right) \left[\frac{\partial \mathbf{W}}{\partial \mathbf{p}}\right]^T (\mathbf{y}_j - \mathbf{x}_i) \quad (15)
\end{aligned}$$

where $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ is the *Jacobian* of the warp:

$$\frac{\partial \mathbf{W}}{\partial \mathbf{p}} = \begin{pmatrix} \frac{\partial W_x}{\partial p_1} & \frac{\partial W_x}{\partial p_2} & \cdots & \frac{\partial W_x}{\partial p_n} \\ \frac{\partial W_y}{\partial p_1} & \frac{\partial W_y}{\partial p_2} & \cdots & \frac{\partial W_y}{\partial p_n} \end{pmatrix} \quad (16)$$

and $\nabla \mathbf{v}_j$ is the image *gradient* of the target image at pixel j.

In equation (15), the first term $\mathbf{G}_1(\mathbf{p})$ is counterpart of the gradient in the Lucas-Kanade algorithm [27, 2] which contributes to the template matching. The Lucas-Kanade algorithm leads to an *iteratively reweighted least squares algorithm*, if robust error function is adopted [2]. The pixels with large residual will get smaller weights to eliminate the effect of outlier.

The second term $\mathbf{G}_2(\mathbf{p})$ is counterpart of equation (9) which accounts for recovering the position of the target.

References

[1] S. Avidan. Support vector tracking. In *Proc. IEEE Conf. Comp. Vision Pattern Recognition*, volume I, pages 184–191, Kauai, HI, 2001.

[2] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *Int'l Journal of Computer Vision*, 56(3):221–255, Feb. 2004.

[3] R. Basri, D. Roth, and D. Jacobs. Clustering appearances of 3D objects. In *Proc. IEEE Conf. Comp. Vision Pattern Recognition*, pages 414–420, Santa Barbara, CA, 1998.

[4] M. J. Black and A. 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.

[5] G. Bradski. Computer vision face tracking for use in a perceptual user interface. *Intel Technology Journal*, 2(Q2), 1998.

[6] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(8):790–799, 1995.

[7] G. Cheung, S. Baker, and T. Kanade. Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. In *Proc. IEEE Conf. Comp. Vision Pattern Recognition*, volume I, pages 77–84, Madison, WI, 2003.

[8] R. Collins. Mean-shift blob tracking through scale space. In *Proc. IEEE Conf. Comp. Vision Pattern Recognition*, volume II, pages 234–240, 2003.

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

[10] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(5):564–577, May 2003.

[11] P. Devijver and J. Kittler. *Pattern Recognition: A Statistical Approach*. Prentice-Hall International, London, 1982.

[12] A. Djouadi, O. Snorrason, and F. D. Garber. The quality of training-sample estimates of the bhattacharyya coefficient. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(1):92–97, Jan. 1990.

[13] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley & Sons, New York, 2nd edition, 2001.

[14] A. Elgammal, R. Duraiswami, and L. Davis. Efficient non-parametric adaptive color modeling using fast Gauss transform. In *Proc. IEEE Conf. Comp. Vision Pattern Recognition*, Kauai, Hawaii, 2001.

[15] A. Elgammal, R. Duraiswami, and L. Davis. Probabilistic tracking in joint feature-spatial spaces. In *Proc. IEEE Conf. Comp. Vision Pattern Recognition*, volume I, pages 781–788, Madison, WI, 2003.

[16] P. Fieguth and D. Terzopoulos. Color based tracking of heads and other mobile objects at video frame rates. In *Proc. IEEE Conf. Comp. Vision Pattern Recognition*, pages 21–27, Puerto Rico, 1997.

[17] K. Fukunaga and L. D. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Trans. Inform. Theory*, 21(1):32–40, 1975.

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

[19] G. Hager and P. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(10):1025–1039, 1998.

[20] G. Hager, M. Dewan, and C. Stewart. Multiple kernel tracking with ssd. In *Proc. IEEE Conf. Comp. Vision Pattern Recognition*, volume 1, pages 790–797, Washington, D.C., 2004.

[21] B. Han, D. Comaniciu, Y. Zhu, and L. Davis. Incremental density approximation kernel-based bayesian filtering for object tracking. In *Proc. IEEE Conf. Comp. Vision Pattern Recognition*, volume 1, pages 638–644, Washington DC, 2004.

[22] A. Hero, B. Ma, O. Michel, and J. Gorman. Applications of entropic spanning graphs. *IEEE Signal Proc. Magazine*, 19(5):85–95, Sept. 2002.

[23] B. K. P. Horn. *Robot Vision*. MIT Press, Cambridge, MA, 1986.

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

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

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

[27] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

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

[29] A. Rangarajan, H. Chui, and F. 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.

[30] C. Sminchisescu and B. Triggs. Kinematic jump processes for monocular 3D human tracking. In *Proc. IEEE Conf. Comp. Vision Pattern Recognition*, volume I, pages 69–76, Madison, WI, 2003.

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

[32] A. R. Webb. *Statistical Pattern Recognition*. John Weley & Sons, UK, 2nd edition, 2002.

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

[34] C. R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(7):780–785, 1997.

[35] C. Yang, R. Duraiswami, N. Gumerov, and L. Davis. Improved fast Gauss transform and efficient kernel density estimation. In *Proc. Int'l Conf. Computer Vision*, pages 464–471, Nice, France, 2003.

[36] J. Yang and A. Waibel. A real-time face tracker. In *Proceedings of WACV*, pages 142–147, Sarasota, FL, 1996.