

Tunable Kernels for Tracking

Vasu Parameswaran, Visvanathan Ramesh, Imad Zoghlami
Real-Time Vision and Modeling Department
Siemens Corporate Research
Princeton, NJ 08540

Abstract

We present a tunable representation for tracking that simultaneously encodes appearance and geometry in a manner that enables the use of mean-shift iterations for tracking. The classic formulation of the tracking problem using mean-shift iterations encodes spatial information very loosely (i.e. using radially symmetric kernels). A problem with such a formulation is that it becomes easy for the tracker to get confused with other objects having the same feature distribution but different spatial configurations of features. Subsequent approaches have addressed this issue but not to the degree of generality required for tracking specific classes of objects and motions (e.g. humans walking). In this paper, we formulate the tracking problem in a manner that encodes the spatial configuration of features along with their density and yet retains robustness to spatial deformations and feature density variations. The encoding of spatial configuration is done using a set of kernels whose parameters can be optimized for a given class of objects and motions, off-line. The formulation enables the use of mean-shift iterations and runs in real-time. We demonstrate better tracking results on synthetic and real image sequences as compared to the original mean-shift tracker.

1. Introduction

We are interested in real-time object tracking, which remains a challenging problem and is of particular relevance in today's emerging application domains such as visual surveillance, driver assistance etc. A crucial component in a solution to tracking is *object representation*, where a key challenge is to capture the 'right' amount of variability of the object. Too much rigidity (e.g. template based approaches) or too much flexibility (e.g. feature-histogram based approaches) will restrict the environments where a tracker can work reliably. The 'right' amount of variability naturally depends on the specific types of motion and the class of object being tracked. In this work, we are interested in the best way to use this type of apriori knowledge

for target representation: specifically, how to one encode variability, and how to learn this variability automatically.

We focus on the mean-shift tracker, originally proposed in [5]. Key advantages of the tracker include fast operation, robustness and invariance to a large class of object deformations. A large body of work followed [5] exploring various related aspects such as feature spaces (e.g. [2], [11]), encoding of spatial information (e.g. recently [14], [1]), shape adaptation (e.g. [13], [15]) etc. The representation chosen in the original formulation is a weighted feature histogram, where each pixel is weighted by a radially symmetric kernel that depends upon its normalized spatial distance from the object center (i.e. a *kernel modulated histogram*). Use of a radially symmetric kernel renders the representation invariant to a large set of transformations (any transformation that preserves the distance of a pixel from the center - e.g. rotations). While the weighting scheme may be appropriate if nothing apriori were known about the object or types of motion that it can undergo, this large amount of invariance poses problems when the object moves close to a region having a similar feature histogram but very different spatial configuration of features, resulting in multiple peaks for the cost function being maximized, and confusion for the tracker. A second issue is that of bandwidth selection for the spatial modulation. Though a significant amount of work has addressed the issue of bandwidth selection for segmentation problems (e.g. [3], [12]) it is not clear how it could be adapted to encode acceptable deformations of a target.

A number of papers have addressed the issue of encoding spatial information into the representation. In the area of image retrieval, the multiresolution histogram [7] offers implicit encoding of spatial information. In the area of tracking, the following papers describe approaches for incorporating spatial information: Hager et. al. [8] analyze the types of motion that the kernel-modulated histogram is invariant to, and propose distributing kernels spatially to capture enough information to recover specific kinds of object motion (e.g. rotation). 'Color correlograms' are used in [14] to capture the cooccurrences of pairs of colors separated by specific distances along orthogonal directions. The

primary focus there is to determine the orientation of a tracked object and no clear methodology is given for distance selection. ‘Spatiograms’ are defined in [1] as an extension to the feature histogram to include higher order statistics of the spatial distributions (the feature histogram itself being a ‘zeroth’ order statistic). It is demonstrated that such a representation allows the mean-shift based tracker to lock on to the target more accurately but does not necessarily succeed where the original tracker fails. Elgammal et. al. [6] represent an object in a joint feature-spatial space and show that histogram based trackers and template based trackers are special cases of their general representation. For kernel modulation, all these tracking papers use a radially symmetric kernel with a globally fixed bandwidth, and do not describe how one should choose its value. More importantly, in these and the multiresolution histogram approach, the key representation need of being able to formulate acceptable deformations of the target and use it to improve tracking is not addressed.

We make two key contributions towards addressing the issue of object representation for articulated object tracking that allow us to learn and specify object appearance changes. First, we propose a kernel modulation that depends upon a set of spatially distributed kernels across the target with variable bandwidths and derive a mean-shift based tracking algorithm, which runs in real-time. Second, we demonstrate how the bandwidth parameters can be estimated for the case of pedestrian tracking by setting up a data-driven optimization problem, where the data come from human motion capture. We demonstrate statistically superior performance of the proposed tracker as compared to the original tracker on real and synthetic image sequences. The rest of the paper is organized as follows: In section 2, we formulate the object representation and derive the tracking algorithm. In section 3 we describe a method for learning the parameters of the representation from human motion capture data. Section 4 presents our evaluation protocol and describes results on synthetic and real image sequences. Finally, in section 5, we summarize our findings and discuss possible improvements and open issues that require further study.

2. Formulation

Let ‘target’ denote the object being tracked and let ‘candidate’ represent an image patch under consideration. Following standard notation, let the target be represented by its feature histogram: $\hat{\mathbf{q}} = \{\hat{q}_u\}_{u=1..m}$ where $\sum_{u=1}^m \hat{q}_u = 1$. Let the target candidate centered at \mathbf{y} be represented by its histogram: $\hat{\mathbf{p}}(\mathbf{y}) = \{\hat{p}_u(\mathbf{y})\}_{u=1..m}$ where $\sum_{u=1}^m \hat{p}_u(\mathbf{y}) = 1$. Let \mathbf{x}_i denote the coordinates of the i th pixel in the candidate and let $\bar{\mathbf{x}}_i$ denote the coordinates of the i th pixel in the target (with center at the origin). The goal is to move to a new position \mathbf{y} given a starting position \mathbf{y}_0 . The un-

weighted (‘raw’) histogram can be computed as follows:

$$\hat{q}_u = \frac{1}{N} \sum_{i=1}^N \delta [b(\bar{\mathbf{x}}_i) - u] \quad (1)$$

where the function $b(\cdot)$ maps a pixel $\bar{\mathbf{x}}_i$ to its feature value. If we were to weight a pixel spatially, a weighting function $\psi(\cdot)$ can be used:

$$\hat{q}_u = C_q \sum_{i=1}^N \delta [b(\bar{\mathbf{x}}_i) - u] \psi(\bar{\mathbf{x}}_i) \quad (2)$$

where C_q is a normalization constant. Similarly, the density at candidate center \mathbf{y} is given by:

$$\hat{p}_u(\mathbf{y}) = C_p \sum_{i=1}^N \delta [b(\mathbf{x}_i) - u] \psi(\mathbf{x}_i - \mathbf{y}) \quad (3)$$

The original mean-shift based tracker [5] chooses a weighting function that is a radially symmetric kernel function with a given bandwidth h : $\psi(\bar{\mathbf{x}}_i) \equiv k\left(\left\|\frac{\bar{\mathbf{x}}_i}{h}\right\|^2\right)$. The intuition there is encode a heuristic that pixels near the center are more likely to come from the target. If nothing is known about the object a priori, this appears to be a reasonable choice for $\psi(\cdot)$. Note that all transformations that preserve distance of each pixel from the center result in identical histograms for the above choice for $\psi(\cdot)$. This may be an overly permissive invariance than is required for tracking certain classes of object. For example, if the problem context involves human beings walking upright in the scene, we would like $\psi(\cdot)$ to encode the constraint that humans do not suddenly invert their appearance while walking. On the other hand, we would also like to choose $\psi(\cdot)$ so that robustness to acceptable spatial deformations of the object is retained. Our main focus for this paper is to determine and use the best $\psi(\cdot)$, given the apriori knowledge that the target belongs to a certain class - e.g. humans walking upright. We consider a parametric family of functions distinguished by a parameter vector Θ . Hence, $\psi(\cdot) \equiv \psi(\mathbf{x}_i, \Theta)$. We choose $\psi(\cdot)$ as follows (we will explain the motivation shortly):

$$\psi(\mathbf{x}) = \sum_{j=1}^N \delta [b(\mathbf{x}) - b(\bar{\mathbf{x}}_j)] k\left(\left\|\frac{\mathbf{x} - \bar{\mathbf{x}}_j}{h_j}\right\|^2\right) \quad (4)$$

Here $k(x)$ is any convex, monotonically decreasing kernel profile as in the original formulation. $\Theta = \{h_j\}$ denotes the set of bandwidths associated with each spatial position j in the target and specifies the allowed motion of the pixel. For positions that are expected to move very little, their h_j should be small, penalizing pixels of the same feature that are observed far away from where they originally appeared in the target. In practice, rather than choosing a bandwidth

for each pixel in the target, it is more efficient to divide the target into M blocks ($B_j, j = 1..M$) and specify a bandwidth for each block. We do this as follows: During initialization, the spatial distribution $S_j^{(u)}$ of feature u is calculated as follows:

$$S_j^{(u)} = C_s^{(u)} \sum_{\bar{\mathbf{x}} \in B_j} \delta [b(\bar{\mathbf{x}}) - u] \quad (5)$$

$S_j^{(u)}$ denotes the fraction of pixels of feature u that occur in block j of the target and $C_s^{(u)}$ is a normalization constant (this bears resemblance to the ‘annular histogram’ used in [10]). The candidate density now becomes:

$$\hat{p}_u(\mathbf{y}) = C_p \sum_{i=1}^N \delta [b(\mathbf{x}_i) - u] \sum_{j=1}^M S_j^{(u)} k \left(\left\| \frac{\mathbf{x}_i - \mathbf{y} - \bar{\mathbf{z}}_j}{h_j} \right\|^2 \right) \quad (6)$$

Here $\bar{\mathbf{z}}_j$ denotes the center of block j . The motivation behind our choice for $\psi(\cdot)$ is two fold. Firstly, the bandwidth parameter h_j , allows penalizing the appearance of pixels at large distances from blocks where the same feature was observed. The bandwidths can be tuned for specific classes of object, and they specify acceptable deformations for that class (we show how to estimate this in section 3). As an example, for humans walking upright in a scene, the torso appearance is expected to be fairly constant, and so the bandwidths corresponding to torso blocks will be small. On the other hand, blocks near the feet would have a larger bandwidth to account for larger motion. Secondly, as the formulation still uses radially symmetric kernels it remains amenable to mean-shift iterations.

We now derive the tracking equation. Given an initial target position \mathbf{y}_0 , the goal is to move to a new position which maximizes the Bhattacharya coefficient ρ between the candidate region and the target region. It can be shown [5] that for a small motion $\Delta \mathbf{y}$ around \mathbf{y}_0 , the Bhattacharya coefficient can be approximated as

$$\rho(\hat{\mathbf{p}}(\mathbf{y}), \hat{\mathbf{q}}) \approx \frac{1}{2} \rho(\hat{\mathbf{p}}(\mathbf{y}_0), \hat{\mathbf{q}}) + \frac{1}{2} \sum_{u=1}^m \sqrt{\frac{\hat{q}_u}{\hat{p}_u(\mathbf{y}_0)}} \hat{p}_u(\mathbf{y}) \quad (7)$$

where $\mathbf{y} = \mathbf{y}_0 + \Delta \mathbf{y}$. The first term being a constant, $\Delta \mathbf{y}$ is chosen such that the following is maximized:

$$\sum_{i=1}^N \sqrt{\frac{\hat{q}_{u_i}}{\hat{p}_{u_i}(\mathbf{y}_0)}} \sum_{j=1}^M S_j^{(u_i)} k \left(\left\| \frac{\mathbf{x}_i - \mathbf{y} - \bar{\mathbf{z}}_j}{h_j} \right\|^2 \right) \quad (8)$$

Here u_i is the feature at pixel i . Let $w_i = \sqrt{\frac{\hat{q}_{u_i}}{\hat{p}_{u_i}(\mathbf{y}_0)}}$. After taking the gradient and some algebra (similar to [4]), we find that the mean-shift vector (i.e the position that maximizes the Bhattacharya coefficient and hence, one that best

matches the candidate with the target) is the following:

$$\mathbf{y}^{(t+1)} = \frac{\sum_{i=1}^N w_i \sum_{j=1}^M S_j^{(u_i)} g \left(\left\| \frac{\mathbf{x}_i - \mathbf{y}^{(t)} - \bar{\mathbf{z}}_j}{h_j} \right\|^2 \right) \left(\frac{\mathbf{x}_i - \bar{\mathbf{z}}_j}{h_j^2} \right)}{\sum_{i=1}^N w_i \sum_{j=1}^M S_j^{(u_i)} g \left(\left\| \frac{\mathbf{x}_i - \mathbf{y}^{(t)} - \bar{\mathbf{z}}_j}{h_j} \right\|^2 \right) \left(\frac{1}{h_j^2} \right)} \quad (9)$$

where $g(\cdot) = -k'(\cdot)$. For k , we use the 2D Epanechnikov kernel:

$$k(x) = \begin{cases} \frac{2}{\pi}(1-x) & \text{if } x \leq 1 \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

In this case, the derivative is constant or zero and we define:

$$D_{ij}^{(t)} = \begin{cases} 1 & \text{if } \left\| \frac{\mathbf{x}_i - \mathbf{y}^{(t)} - \bar{\mathbf{z}}_j}{h_j} \right\|^2 \leq 1 \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

The mean-shift vector now takes on a simpler form:

$$\mathbf{y}^{(t+1)} = \frac{\sum_{i=1}^N w_i \sum_{j=1}^M S_j^{(u_i)} D_{ij}^{(t)} \left(\frac{\mathbf{x}_i - \bar{\mathbf{z}}_j}{h_j^2} \right)}{\sum_{i=1}^N w_i \sum_{j=1}^M S_j^{(u_i)} D_{ij}^{(t)} \left(\frac{1}{h_j^2} \right)} \quad (12)$$

Define the following two quantities:

$$\begin{aligned} \alpha_i^{(u,t)} &= \sum_{j=1}^M \frac{1}{h_j^2} S_j^{(u)} D_{ij}^{(t)} \\ \mathbf{v}_i^{(u,t)} &= \sum_{j=1}^M \frac{1}{h_j^2} S_j^{(u)} D_{ij}^{(t)} \bar{\mathbf{z}}_j \end{aligned} \quad (13)$$

The mean-shift vector becomes

$$\mathbf{y}^{(t+1)} = \frac{\sum_{i=1}^N w_i \left(\alpha_i^{(u_i,t)} \mathbf{x}_i - \mathbf{v}_i^{(u_i,t)} \right)}{\sum_{i=1}^N w_i \alpha_i^{(u_i,t)}} \quad (14)$$

Note that if we simply used one block centered in the middle, the mean-shift vector reduces to the original version derived in [5] (because $\bar{\mathbf{z}} = \mathbf{0}$ and $\alpha_i^{(u_i,t)} = 1/h^2$).

3. Tuning the Bandwidths

We are primarily interested in tracking pedestrians in a scene although the method we describe here can be adapted to many other classes of objects and motions. The bandwidths depend upon the extent of deformation the target undergoes, and in our case, we propose to use motion capture data to estimate the deformations. Motion capture allows a lot of flexibility: we can calculate a dense deformation map of each patch on the body by texture-mapping it with a unique color and rendering the motion in a controlled graphical environment. Since a patch is uniquely colored we can easily locate it in all rendered images and

determine its set of movements. We used publicly available motion capture data and rendered several humans walking in place, while at the same time, positioning a virtual camera at various positions on the frontal hemisphere of the person (see figure 1 for two example images). We collected about 900 such images (note that each image represents one pose). Given a choice of bandwidths Θ , we calculate the set of histograms $H^+(\Theta)$. We wish to choose Θ that maximizes the similarity between elements in $H^+(\Theta)$. However, such a ‘one-class optimization’ may overly increase the bandwidths, attempting to accommodate all the poses and thereby reducing the discriminating power of the representation. In principle, we would like to simultaneously minimize the similarity between the elements in the set $H^+(\Theta)$ and those in a *negative* set $H^-(\Theta)$. The construction of $H^-(\Theta)$ depends upon the end application and denotes the set of object appearance changes that are unacceptable. One way to construct $H^-(\Theta)$ is to swap the colors about a horizontal axis (implicitly encoding the fact that the head region cannot move to the feet region and vice versa, for example). Another choice is to render each pose after randomly redistributing the colors. Yet another choice is the set of all possible histograms in the color space (thereby giving rise to a uniformly distributed Bhattacharyya distance with respect to elements in $H^+(\Theta)$). In our case, we used random redistribution of the colors. Let the probability distribution of pairwise Bhattacharyya coefficients in the set $H^+(\Theta)$ be $p^+(\rho|\Theta)$ and let the probability distribution of Bhattacharyya coefficients between elements in $H^+(\Theta)$ and those in $H^-(\Theta)$ be $p^-(\rho|\Theta)$. The bandwidths can now be calculated as the solution to the following optimization problem:

$$\Theta^* = \operatorname{argmax} (f(p^+(\rho|\Theta), p^-(\rho|\Theta))) \quad (15)$$

where $f(\cdot)$ is a functional capturing a metric we wish to maximize. We experimented with two different functionals for choosing the best set of bandwidths Θ . Both approaches yielded comparable performances. The first approach used the functional $2\min(\rho^+|\Theta) - \max(\rho^-|\Theta)$. The intuition here is to make the classes as widely separated as possible (i.e. maximize $\min(\rho^+|\Theta) - \max(\rho^-|\Theta)$) while at the same time ensuring that $p^+(\rho|\Theta)$ is as close to 1 as possible (i.e. maximize $\min(\rho^+|\Theta)$). We refer to this metric as the ‘class-separation’ metric. The second approach was based on a discriminability metric. Consider a given choice of bandwidths Θ . For a given *cut-off threshold* ρ_t , the two types of errors, i.e. the miss-detection and false-alarm rates can be written down as follows:

$$\begin{aligned} m(\Theta, \rho_t) &= \int_0^{\rho_t} p^+(\rho|\Theta) d\rho \\ f(\Theta, \rho_t) &= \int_{\rho_t}^1 p^-(\rho|\Theta) d\rho \end{aligned} \quad (16)$$

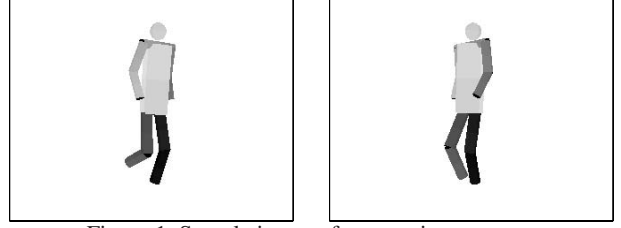


Figure 1. Sample images from motion capture

For various choices of ρ_t , the errors $m(\Theta, \rho_t)$ and $f(\Theta, \rho_t)$ when plotted against each other, trace an ROC (Receiver Operating Characteristics) curve. The area under the ROC curve (AUC) integrates out various choices for ρ_t and is a measure of how well the set of bandwidths Θ discriminates between the two classes. Smaller AUC implies better discrimination (the best discriminator will have AUC=0 while a random discriminator will have AUC=1/2). The best Θ to use can be found as the solution to the following optimization problem:

$$\Theta^* = \operatorname{argmin} \left(\int_0^1 m(\Theta, \rho_t) \left[\frac{\partial f(\Theta, \rho_t)}{\partial \rho_t} \right] d\rho_t \right) \quad (17)$$

We refer to this metric as the ‘class-discrimination’ metric. We first normalize the body into a square and use 4 equally sized vertical blocks which makes $\Theta = (h_1, h_2, h_3, h_4)$ and $0 \leq h_j \leq 1$ (here h_1 corresponds to the head area and h_4 to the feet area). Disappointingly, it is not possible to solve for the best h_j analytically due to the square-roots in the Bhattacharyya distance and due to the implicit dependence of $D_{ij}^{(t)}$ on h_j . Since the domain is small enough and the optimization is done only once and offline, we chose to explore the space by brute-force, using $\Delta h_j = 0.05$. The class-separation metric found the optimal bandwidths to be $(h_1 = 0.4, h_2 = 0.5, h_3 = 0.35, h_4 = 0.5)$. Figure 2 shows two 3D projections of the 5D space at the optimal values using the class-separation metric. The class-discrimination metric found the optimal bandwidths to be $(h_1 = 0.35, h_2 = 0.30, h_3 = 0.25, h_4 = 1.00)$. Figure 3 shows two 3D projections of the 5D space at the optimal values using the class-discrimination metric. In this case, a range of values for Θ was found to be able to produce good results (as can be seen from the figure). The bandwidth set $(0.25, 0.25, 0.25, 0.75)$ was found to work best for all the sequences we considered.

4. Results

We compare the behaviors of the original mean-shift tracker and the tracker proposed in this work on one synthetic and four real image sequences. Both trackers used RGB color as the feature space with 4 bits per channel. For both trackers, scale adaptation was done by searching for a small range of scales at each frame and choosing the best one. Also,

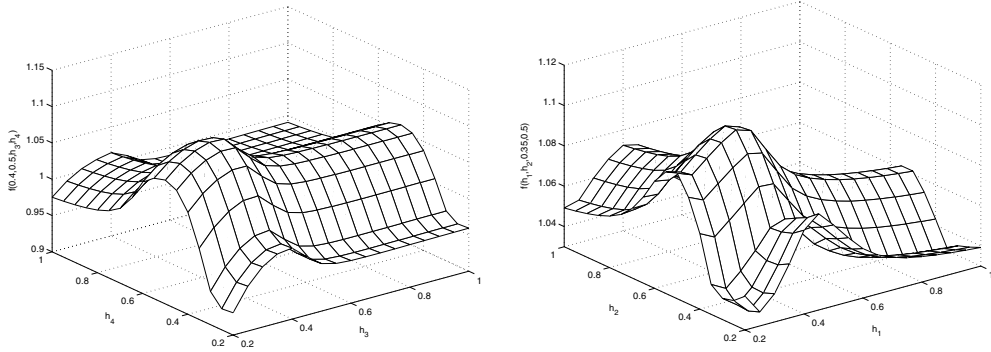


Figure 2. Class-separation metric: cost surface for $\Theta = (0.4, 0.5, h_3, h_4)$ and $\Theta = (h_1, h_2, 0.35, 0.5)$ (maximization)

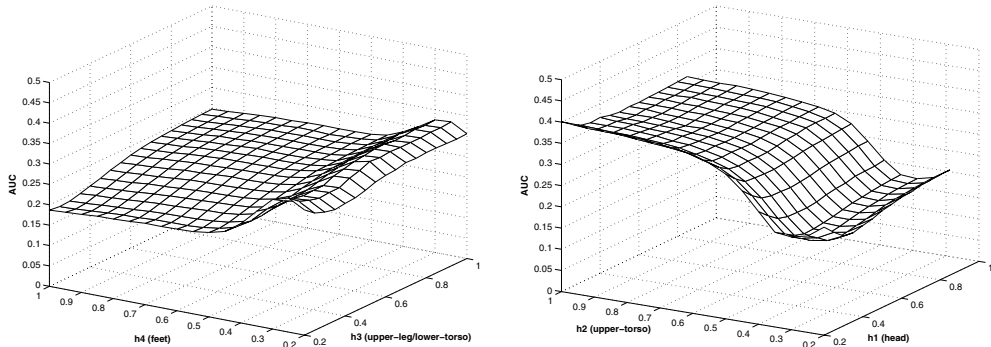


Figure 3. Class-discrimination metric: cost surface for $\Theta = (0.35, 0.30, h_3, h_4)$ and $\Theta = (h_1, h_2, 0.25, 1.00)$ (minimization)

identical starting positions were used for both. We observe that in the presence of objects of similar feature distribution but quite different spatial configuration of features, there will be competing peaks in the cost function surface being optimized by the original tracker. However, the proposed tracker should suppress competing peaks. Hence, on average, we would expect the proposed tracker to follow the correct target more often. To demonstrate this, we analyzed the statistical behavior of the trackers as follows: For the set of image sequences, we first generated ground truth semi-automatically, i.e. for each image sequence, for a successful run of the tracker (verified visually), the tracker trajectory was stored as ground truth. Following this, repeated trials were carried out where, at each frame, the coordinates returned by the tracker were perturbed before they were fed back to the tracker. We consider the average error per noise level as a measure of how well each tracker performed. Figure 4 shows the measure for the synthetic and real image sequence case (averaged over all four). The source of the perturbation was Gaussian noise of zero mean and unit standard deviation. The x (resp. y) perturbation at each frame was arrived at by multiplying the noise with the object width (resp. height) and scaling it by a ‘noise level’ factor. The error shown is a multiple of the object dimension. Note that only the error trend is of relevance, not the actual magnitudes which will depend upon the distance between the target and the confusion peaks in that specific scenario (i.e.

which competing object the tracker was lost to). It can be seen that the error is lower for the proposed tracker as we expect. The error for the original tracker was high even with no perturbation of the trajectory because of competing objects in the vicinity and the perturbation sometimes improved its performance because it pushed the tracker more towards the correct target occasionally. We now show some specific examples from the image sequences. In all the figures, the first row shows the original tracker behavior and the second row shows the tracker proposed in this paper. Figure 5 shows a rectangular block being tracked and overlapping another ‘decoy’ block with the same color distribution but inverted spatial configuration of colors. The proposed tracker used two vertically stacked zones with identical bandwidths of 0.3. During repeated trials as described above, the original tracker got attracted to the second block more often than the proposed tracker, and the figure shows one such run. The real image sequences demonstrate potential confusion for the tracker due to the presence of objects with similar color but different spatial configurations of the colors and temporary occlusions. Figure 6 shows a canonical example where the target is a person with a white shirt and black pants. The person moves the right, first occluding and then revealing another person with a black shirt and white pants. The original mean-shift tracker gets attracted to the second person while the proposed tracker follows the original person. Figure 7 shows an example where

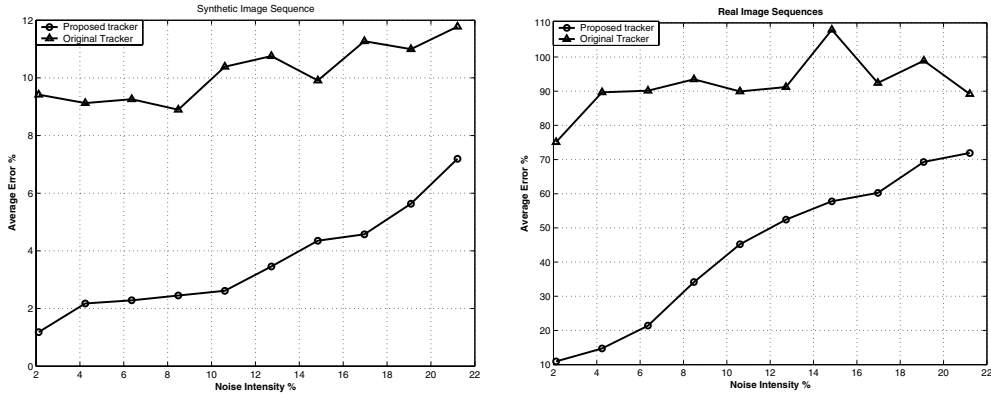


Figure 4. Error trend for increasing noise levels for the original and proposed trackers.

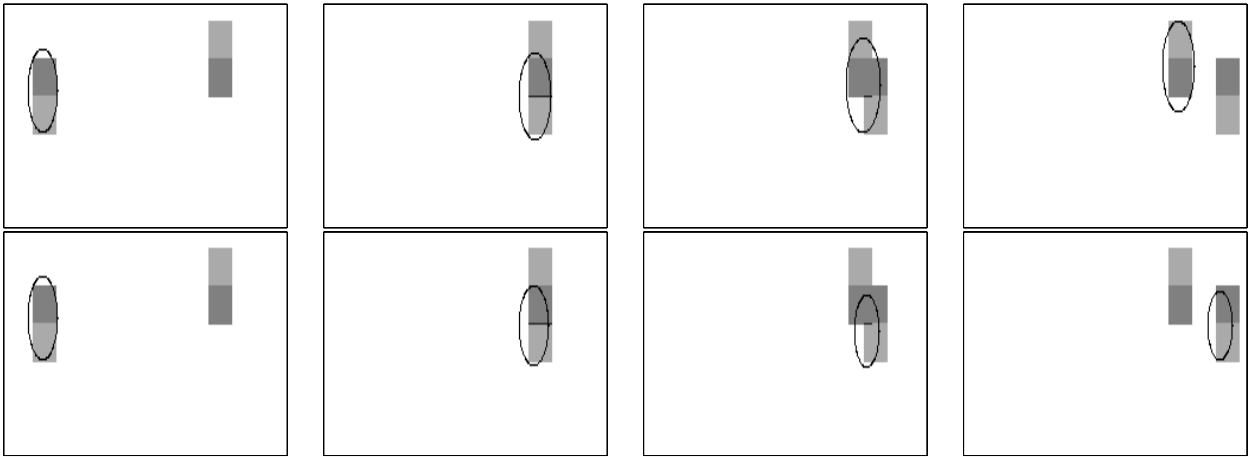


Figure 5. Synthetic image sequence. Top row: orig. tracker at frames 1, 150, 163 and 190. Bottom row: proposed tracker at same frames.

the person being tracked gets occluded partially and then reemerges around frame 242. The proposed tracker follows the person quite well whereas the original tracker loses the person to the background. Figure 8 shows a similar example. Figure 9 shows an example where the tracked person shrinks in size fast, bends and gets occluded frequently. The proposed tracker is able to withstand these effects for an extended period of time until it finally gets overwhelmed by severe occlusion. In this example, the original tracker loses the person by frame 132 while the proposed tracker tracks for about 2800 frames. Although not specifically designed to withstand occlusions, the ability of the tracker to discriminate spatially provides for robustness to small or intermittent occlusions: when the target gets occluded, the tracker does not latch on to any nearby image patch with the same feature distribution because the patch likely will not conform to the expected spatial distribution of colors. Hence, there will likely not be a strong gradient in any direction, and the tracker will continue staying around the same position. It will recover if the target re-appears and there is significant overlap between the re-appearing target and the current region, resulting in a strong gradient.

5. Conclusions

We identified a key problem with previous formulations of the tracking problem, namely, that incorporation of knowledge of the object type and motion has not been addressed. We presented a way to modulate the feature histogram of the target in a manner that encodes spatial information using a set of spatial kernels with variable bandwidths. We showed how one could learn the optimal set of bandwidths for the case of pedestrians walking upright using motion capture data, and we demonstrated that the proposed tracker tracked targets better in the presence of multiple distracting objects with similar feature distributions. There are several areas for improvement. First, we have not addressed the issue of model-update: under what conditions should the target histogram be updated? This is a difficult problem, because it requires one to detect whether an observed appearance change is due to the target changing appearance or a temporary occlusion. It may be worthwhile exploring how our approach can be combined with work in [9] or [15]. Second, a closely related issue is that of dependence of the optimal bandwidth parameters on the pose of the person. Our work used a ‘global’ optimum across a set of poses by virtually rotating the rendered human as it walked. It may be possible to perform better tracking by



Figure 6. Real img. seq. 1: Top Row: Orig. tracker at frames 0, 71, 128, 151. Bottom row: Proposed tracker at frames 0, 71, 128, 247.

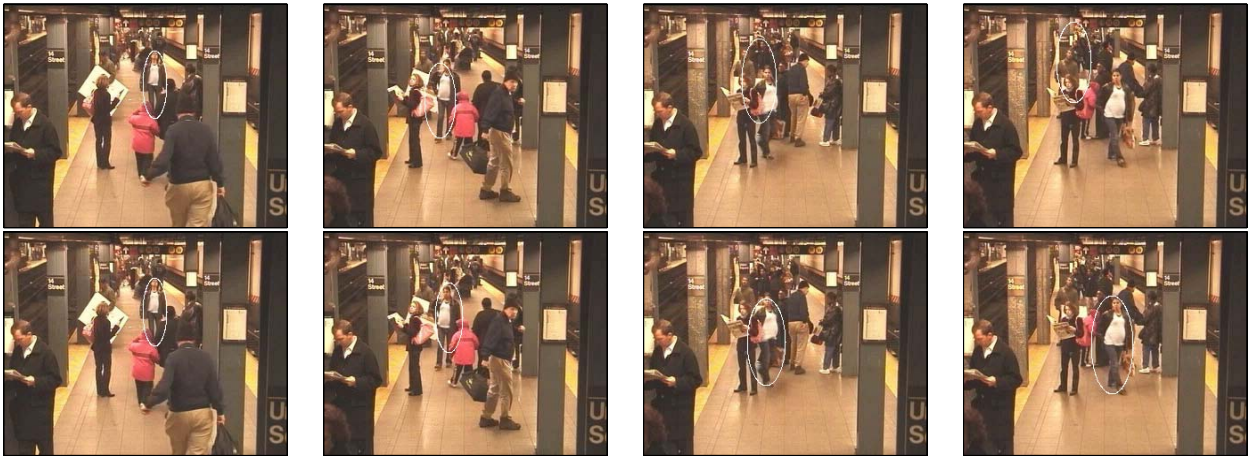


Figure 7. Real img. seq. 2: Top Row: Classic tracker for frames 0, 85, 242, 271. Bottom row: Proposed tracker at the same frames.

using a state-space based representation allowing the bandwidths to be a function of the target state. Finally, we chose four vertically aligned spatial blocks for the representation and a method to estimate their optimal bandwidths. This can be improved: The key open issue is how can one decide on the number, spatial positions and the optimal bandwidths to use, given a model of the target and its motion. We will be exploring each of these areas in the future.

References

- [1] S. Birchfield and S. Rangarajan. Spatiograms versus histograms for region-based tracking. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2005.
- [2] R. Collins. Mean-shift blob tracking through scale space. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2003.
- [3] D. Comaniciu. An algorithm for data-driven bandwidth selection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(2):281–288, Feb. 2003.
- [4] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(5):603–619, May 2002.
- [5] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(5):564–577, May 2003.
- [6] A. Elgammal, R. Duraiswami, and L. S. Davis. Probabilistic tracking in joint feature-spatial spaces. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2003.
- [7] E. Hadjidemetriou, M. D. Grossberg, and S. K. Nayar. Multiresolution histograms and their use for recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(7):831–847, July 2004.
- [8] G. D. Hager, M. Dewan, and C. Stewart. Multiple kernel tracking with ssd. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2004.
- [9] A. Jepson, D. Fleet, and T. El-Maraghi. Robust online appearance models for visual tracking. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2001.
- [10] A. Rao, R. Srihari, and Z. Zhang. Spatial color histograms for content-based image retrieval. *Proc. IEEE Intl. Conf. on Tools with Artificial Intelligence*, 1999.
- [11] K. She, G. Bebis, H. Gu, and R. Miller. Vehicle tracking using on-line fusion of color and shape features. *Proc. IEEE*

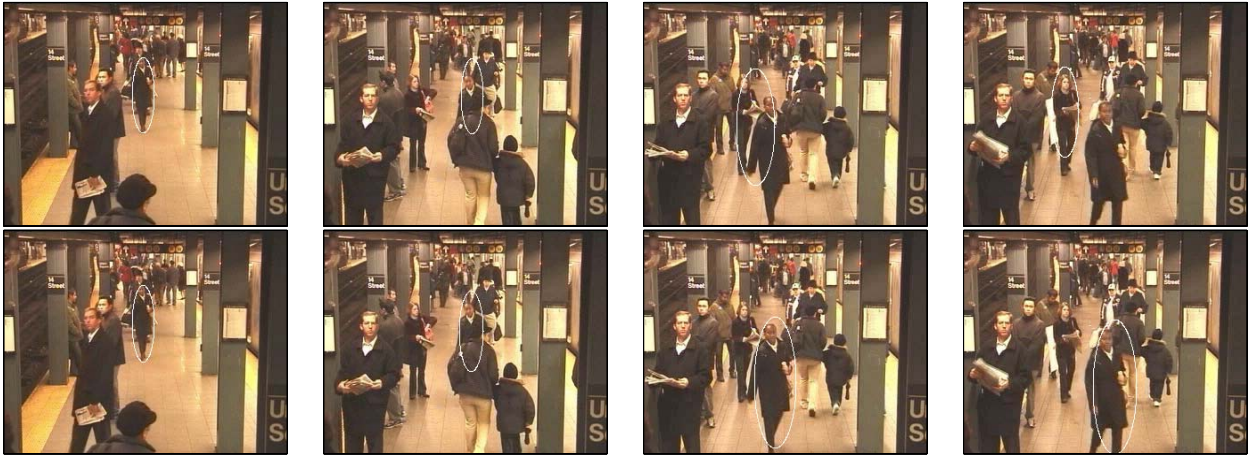


Figure 8. Real img. seq. 3: Top Row: Classic tracker for frames 0, 90, 156, 171. Bottom row: Proposed tracker at the same frames.

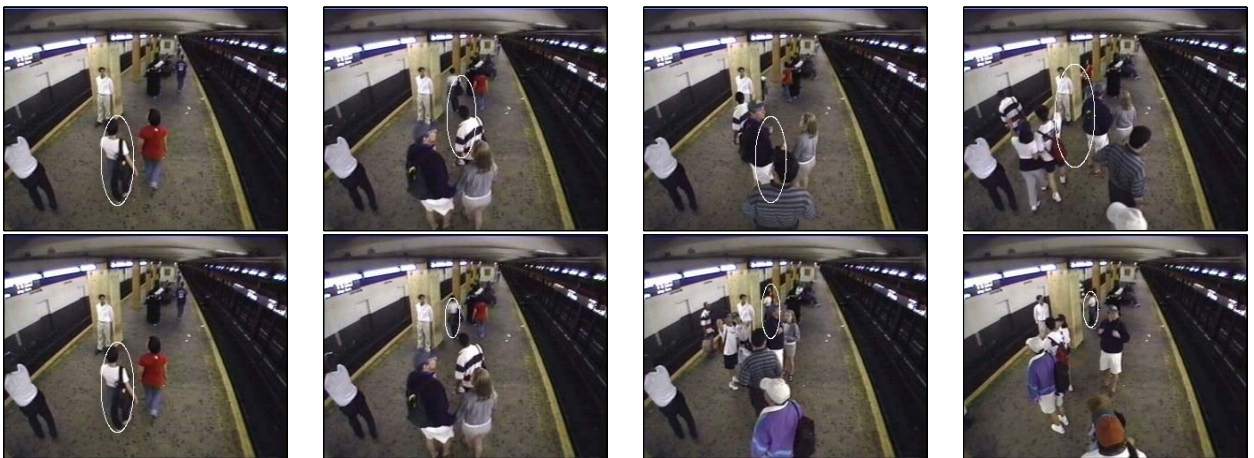


Figure 9. Real img. seq. 4: Top Row: Classic tracker for frames 0, 92, 132, 203. Bottom row: Proposed tracker at frames 0, 92, 360, 864.

Conf. on Intelligent Transportation Systems, 2004.

- [12] J. Wang, B. Thiesson, Y. Xu, and M. Cohen. Image and video segmentation by anisotropic kernel mean shift. *Proc. European Conference on Computer Vision*, 2004.
- [13] H. Zhang, Z. Huang, W. Huang, and L. Li. Kernel-based method for tracking objects with rotation and translation. *Proc. International Conf. on Pattern Recognition*, 2004.
- [14] Q. Zhao and H. Tao. Object tracking using color correlogram. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2005.
- [15] Z. Zivkovic and B. Krose. An em-like algorithm for color-histogram-based object tracking. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2004.