IEEE Conference on Computer Vision and Pattern Recognition (CVPR) San Diego, California, June 2005

Spatiograms Versus Histograms for Region-Based Tracking

Stanley T. Birchfield Sriram Rangarajan
Electrical and Computer Engineering Department
Clemson University, Clemson, SC 29634
{stb, srangar}@clemson.edu

Abstract

We introduce the concept of a spatiogram, which is a generalization of a histogram that includes potentially higher order moments. A histogram is a zeroth-order spatiogram, while second-order spatiograms contain spatial means and covariances for each histogram bin. This spatial information still allows quite general transformations, as in a histogram, but captures a richer description of the target to increase robustness in tracking. We show how to use spatiograms in kernel-based trackers, deriving a mean shift procedure in which individual pixels vote not only for the amount of shift but also for its direction. Experiments show improved tracking results compared with histograms, using both mean shift and exhaustive local search.

1 Introduction

Histograms have proved themselves to be a powerful representation for the image data in a region. Discarding all spatial information, they are the foundation of classic techniques such as histogram equalization and image indexing [9]. Building upon these concepts, several successful tracking systems have been developed over the last several years using color histograms, taking advantage of their robustness to changing object pose and shape [1, 8, 10, 3, 2, 4, 11].

Other tracking systems have traditionally adopted a completely different point of view. Representing an image region by a template window of pixel intensities, the window is registered with the previous frame of the sequence to determine the displacement of the object [7, 5]. Such an approach lies at the opposite end of the spectrum from histograms, because the spatial arrangement of the pixels in the window is explicitly expected not to deviate from a low-order parametric motion model.

Recently, Hager et al. [4] developed a connection between these two seemingly unrelated techniques by proposing to use multiple spatially-weighted histograms. The mathematical mechanism for enabling this connection is the mean shift algorithm, which is a kernel-based method for determining the alignment between two probability distributions. Mean shift has recently gained significant attention as an efficient and robust method for visual tracking [3, 2, 11].

In this paper we consider the concept of a single histogram in which each bin is spatially weighted by the mean and covariance of the locations of the pixels that contribute to that bin. We call this concept a spatial histogram, or *spatiogram*. We show that spatiograms are simply histograms with higher-order moments, and that histograms are zeroth-order spatiograms. Spatiograms are a richer representation, capturing not only the values of the pixels but their spatial relationships as well. We derive a mean shift procedure for spatiograms and demonstrate improved tracking results when compared with traditional histograms on an image sequence of a person's head.

2 Histograms and spatiograms

Given a discrete function $f : x \rightarrow v$, where $x \in \mathcal{X}$ and $v \in \mathcal{V}$, a *histogram* of f captures the number of occurrences of each element in the range of f. More specifically, the histogram is $h_f : v \rightarrow \mathcal{Z}^*$, where $v \in \mathcal{V}$ and $\mathcal{Z}^*$ is the set of non-negative integers, and $h_f(v)$ is the number of elements $x \in \mathcal{X}$ such that $f(x) = v$. Another way to look at h_f is as the marginal of a binary function $g_f(x, v)$ over x, where $g_f(x, v) = 1$ if $f(x) = v$ and 0 otherwise. That is, $h_f(v) = \sum_{x \in \mathcal{X}} g_f(x, v)$ is the zeroth-order moment of g along the v dimension. Histograms are important because they discard all information about the domain, thus making them invariant to *any* one-to-one transformation of the domain of the original function.

A limited amount of information regarding the domain may be retained by using higher-order moments of the binary function g, where the ith-order moment is given by $h_f^{(i)}(v) = \sum_{x \in \mathcal{X}} x^i g_f(x, v)$. We use the term spatial histogram, or *spatiogram*, to refer to this concept, because it captures not only occurrence information about the range of the function, as in a histogram, but also information about the (spatial) domain. We define the kth-order spatiogram to be a tuple of all the moments up to order k: $\langle h_f^{(0)}(v), \ldots, h_f^{(k)}(v) \rangle$. A histogram, then, is just a zeroth-

order spatiogram. To our knowledge, higher-order spatiograms have not previously been explored.

The spatiogram may be thought of as a geometric model bridging the gap between histograms, which allow for arbitrary transformations, and more specific models such as translation, similarity, affine, projective, or B-splines. Like histograms, spatiograms are efficient to compute, and they enable comparison between corresponding image patches without specifically calculating the geometric transformation between them. Nevertheless, like the more specific models, spatiograms retain some information about the geometry of the patches. Compared with co-occurrence matrices [6], spatiograms capture the global positions of the pixels rather than their pairwise relationships.

2.1 Spatiograms in images

An image is a two-dimensional mapping $I : \mathbf{x} \rightarrow v$ from pixels $\mathbf{x} = [x, y]^T$ to values v. For our purposes, the meaning of these values is arbitrary. They may represent raw gray-level intensities or component colors, or the result of preprocessing (quantization, color space transformation, wavelet coefficients, etc.).

We represent the second-order spatiogram of an image as

$$h_I^{(2)}(b) = \langle n_b, \mu_b, \Sigma_b \rangle, \qquad b = 1, \ldots, B,$$

where n_b is the number of pixels whose value is that of the bth bin, and μ_b and Σ_b are the mean vector and covariance matrices, respectively, of the coordinates of those pixels. (There is a one-to-one transformation from this parameterization to the non-centralized moments mentioned earlier.) The number $B = |\mathcal{V}|$ is the number of bins in the spatiogram. Notice that

$$h_I^{(0)}(b) = n_b, \qquad b = 1, \ldots, B$$

is just the histogram of I.

The similarity between two spatiograms can be computed as the weighted sum of the similarity between the two histograms:

$$\rho(h, h') = \sum_{b=1}^{B} \psi_b \rho_n(n_b, n'_b). \tag{1}$$

For a zeroth-order spatiogram, $\psi_b = 1$. For a second-order spatiogram, we set ψ_b to the probability that $\bar{\mathbf{x}}_b$ was drawn from a Gaussian distribution described by $(\bar{\mathbf{x}}'_b, \Sigma'_b)$, multiplied by the probability in the reverse direction:

$$\psi_b = \eta \exp\left\{ -\frac{1}{2}(\mu_b - \mu'_b)^T \hat{\Sigma}_b^{-1} (\mu_b - \mu'_b) \right\}, \tag{2}$$

where η is the Gaussian normalization constant and $\hat{\Sigma}_b^{-1} = (\Sigma_b^{-1} + (\Sigma'_b)^{-1})$. Notice that the value inside the summation is the average of the two Mahalanobis distances, one between $\mathbf{x}$ and $\mathbf{x}'$ and the other between $\mathbf{x}'$ and $\mathbf{x}$.

The similarity between the histogram bins can be computed using any of a number of techniques, such as histogram intersection [9, 1]:

$$\rho_n(n_b, n'_b) = \frac{\min(n_b, n'_b)}{\sum_{j=1}^{B} n_j}$$

or the Bhattacharyya coefficient [3]:

$$\rho_n(n_b, n'_b) = \frac{\sqrt{n_b n'_b}}{\sqrt{\left(\sum_{j=1}^{B} n_j\right)\left(\sum_{j=1}^{B} n'_j\right)}}.$$

2.2 Probabilistic view

A spatiogram captures the probability density function (PDF) of the image values:

$$P(I(\mathbf{x}) = v) = p(\mathbf{x}, v) = p(\mathbf{x}|v)p(v), \tag{3}$$

where, for a second-order spatiogram, we have

$$\begin{aligned} p(v) &= \frac{n_b}{\sum_{j=1}^{B} n_j} \\ p(\mathbf{x}|v) &= \eta \exp\left\{ -\frac{1}{2}(\mathbf{x} - \mu_k)^T \Sigma_k^{-1} (\mathbf{x} - \mu_k) \right\}, \end{aligned} \tag{4}$$

v is the value of the bth bin, and η is the Gaussian normalization constant as before. For a zeroth-order spatiogram (histogram), there is no spatial dependency and the joint PDF is equal to the marginal: $p(\mathbf{x}, v) = p(v)$.

It is instructive to consider the relationship between spatiograms and Gaussian Mixture Models (GMMs), which are semi-parametric alternatives to the non-parametric histograms. A GMM captures the probability of a value as a weighted sum of M Gaussians:

$$p(v) = \sum_{j=1}^{M} p(v|j)p(j), \tag{5}$$

where

$$p(v|j) = \eta \exp\left\{ -\frac{1}{2}(v - \mu_j)^T \Sigma_j^{-1} (v - \mu_j) \right\} \tag{6}$$

is the Mahalanobis distance to the jth Gaussian, and $p(j)$ is the a priori likelihood that j is the correct Gaussian from which v is drawn.

Comparing Eqs. (3)-(4) with (5)-(6), we see that spatiograms and GMMs involve similar computations. The difference is that GMMs capture the likelihood of a value as the weighted sum of multiple Gaussians defined in the range of I, while spatiograms capture the likelihood of a value at a particular location as the weight of a single Gaussian defined in the domain of I. Whereas GMMs are

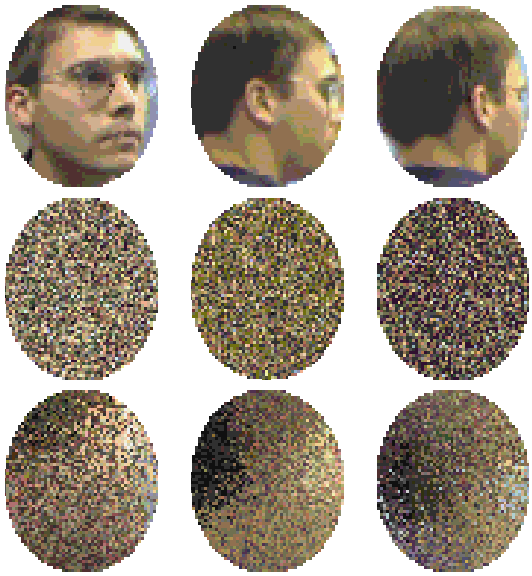


Figure 1: Three different poses of a person (top), with images generated from the histogram (middle) and spatiogram (bottom). The spatiogram captures spatial relationships among the colors, whereas the histogram discards all spatial information.

non-parametric in their domain and semi-parametric in their range, and histograms are non-parametric in both their domain and range, spatiograms are non-parametric in their range but semi-parametric in their domain. One could combine spatiograms with GMMs to obtain a representation in both domain and range, but we have left that for future research. As an aside, we also notice that, comparing Eqs. (1)-(2) with (5)-(6), the difference between two spatiograms is exactly a GMM, with a Gaussian associated with each bin.

Figure 1 illustrates the difference between a histogram and a spatiogram. For each of three poses of a person's head, we computed the histogram and spatiogram, which we then used as a generative model to produce a new image the same size as the original by sampling the PDF given by Eqs. (3)-(4). An RGB color space was used with 8 bins per channel, and only the diagonal elements of Σ_i were used, with the variances clipped at 1 pixel in order to make Σ_i invertible even for bins for which n_b is 0 or 1.

3 Tracking by mean shift

In our context, the problem of tracking is to determine the image location $\mathbf{y} \in \mathcal{R}^2$ in the current image frame for which the similarity is maximized between the model spatiogram $h' = \langle n', \mu', \Sigma' \rangle$ and the spatiogram $h(\mathbf{y}) = \langle n(\mathbf{y}), \mu(\mathbf{y}), \Sigma(\mathbf{y}) \rangle$ at the location $\mathbf{y}$. Generally the model comes from the previous image frame.

The mean shift algorithm is an efficient technique for computing the location $\mathbf{y}$ corresponding to the nearest mode of the probability distribution [3]. As a kernel-based technique, it requires that the histogram be smoothed with the profile $k : [0, \infty) \rightarrow \mathcal{R}$ of a suitable kernel:

$$\begin{aligned} n'_b &= C \sum_{i=1}^{N} k(\| \mathbf{x}_i \|^2) \delta_{ib} \\ n_b(\mathbf{y}) &= C_h \sum_{i=1}^{N_h} k(\| (\mathbf{x}_i - \mathbf{y})/h \|^2) \delta_{ib}, \end{aligned} \quad (7)$$

where N is the number of pixels in the model's region, N_h is the number of pixels in the region of size h, and δ_{ib} is 1 if the value of $\mathbf{x}_i$ is that of the bth bin and 0 otherwise. These equations give the value in the bth bin of the model and candidate histograms, respectively. The kernel profile is convex and monotonic decreasing to weight pixels more toward the center and zero outside the region. An Epanechnikov profile

$$k(x) = \begin{cases} \frac{1}{2} c_d^{-1} (d+2)(1-x) & x \leq 1 \\ 0 & x > 1 \end{cases},$$

where c_d is the volume of the unit d-dimensional sphere, and d is the dimensionality of the state space (2 in our case), has the advantage that its derivative is constant, thus simplifying the mean shift equation — see Equations (9) and (10) below. The coordinates $\mathbf{x}_i$, as well as $\mathbf{x}_i - \mathbf{y}$, are normalized so that they reach a value of 1 at the periphery of the region. For an elliptical region this means dividing by the major and minor axis length, and h is a scale parameter to handle different sized ellipses. The normalization constants are chosen so the sum of all the bins is one:

$$\begin{aligned} C &= \frac{1}{\sum_{i=1}^{N} k(\| \mathbf{x}_i \|^2)} \\ C_h &= \frac{1}{\sum_{i=1}^{N_h} k(\| (\mathbf{y} - \mathbf{x}_i)/h \|^2)} \end{aligned}$$

3.1 Mean shift for histograms

Let us first review the special case when h and h' are zeroth-order spatiograms, i.e., histograms, a scenario that has been studied extensively in the literature [3, 2, 11]. Following [3], we define the likelihood $\hat{\rho}(\mathbf{y})$ that the target is at location $\mathbf{y}$ as the similarity between the two histograms using the Bhattacharyya coefficient:

$$\rho(\mathbf{y}) = \rho(h(\mathbf{y}), h') = \sum_{b=1}^{B} \sqrt{n_b(\mathbf{y}) n'_b}$$

A Taylor series expansion around the current histogram $n(\mathbf{y}_0)$ yields a linear approximation to the coefficient:

$$\rho(\mathbf{y}) \approx \rho(\mathbf{y}_0) + [n(\mathbf{y}) - n(\mathbf{y}_0)]^T \frac{\partial \rho}{\partial n}(\mathbf{y}_0)$$

$$= \frac{1}{2}\sum_{b=1}^{B}\sqrt{n_b(\mathbf{y}_0)n'_b} + \frac{1}{2}\sum_{b=1}^{B} n_b(\mathbf{y})\sqrt{\frac{n'_b}{n_b(\mathbf{y}_0)}}$$
$$= \frac{1}{2}\sum_{b=1}^{B}\sqrt{n_b(\mathbf{y}_0)n'_b} + \frac{C_h}{2}\sum_{i=1}^{N_h} w_i k\left(\| \frac{\mathbf{y}-\mathbf{x}_i}{h} \|^2\right),$$

where the weights are given by

$$w_i = \sum_{b=1}^{B}\sqrt{\frac{n'_b}{n_b(\mathbf{y}_0)}}\delta_{ib}. \tag{8}$$

Taking the partial derivative of $\rho(\mathbf{y})$ with respect to $\mathbf{y}$ and setting it to zero yields the mean shift equation:

$$\mathbf{y}_1 = \frac{\sum_{i=1}^{N_h} w_i g\left(\| \frac{\mathbf{y}_0-\mathbf{x}_i}{h} \|^2\right)\mathbf{x}_i}{\sum_{i=1}^{N_h} w_i g\left(\| \frac{\mathbf{y}_0-\mathbf{x}_i}{h} \|^2\right)}, \tag{9}$$

where $g(x) = -dk(x)/dx$ is the negative derivative of the kernel profile. As mentioned above, if k is the Epanechnikov profile, then the equation reduces to a simple weighted average:

$$\mathbf{y}_1 = \frac{\sum_{i=1}^{N_h} w_i\mathbf{x}_i}{\sum_{i=1}^{N_h} w_i}. \tag{10}$$

The algorithm is straightforward. For the first image frame we compute the histogram of the elliptical region using Eq. (7) and store it as the model. Then, for each new image frame we compute the histogram of the region using Eq. (7), which we use to calculate the weights according to Eq. (8), which are then used to determine the offset to the position vector by Eq. (9) or Eq. (10). These three steps are repeated until convergence for each new image frame.

3.2 Mean shift for spatiograms

For a spatiogram, we must consider as well the means and covariances:[1]

$$\mu_b(\mathbf{y}) = \frac{1}{\sum_{j=1}^{N_h}\delta_{jb}}\sum_{i=1}^{N_h}(\mathbf{x}_i-\mathbf{y})\delta_{ib}$$
$$\Sigma_b(\mathbf{y}) = \frac{1}{\sum_{j=1}^{N_h}\delta_{jb}}\sum_{i=1}^{N_h}(\mathbf{x}_i-\mu_b(\mathbf{y}))^T(\mathbf{x}_i-\mu_b(\mathbf{y}))\delta_{ib}$$

To compare, we use the similarity measure in Equation (1):

$$\rho(\mathbf{y}) = \rho(h(\mathbf{y}),h') = \sum_{b=1}^{B}\psi_b(\mathbf{y})\sqrt{n_b(\mathbf{y})n'_b}$$

where $\psi_b(\mathbf{y})$ is given by Equation (2):

$$\psi_b(\mathbf{y}) = \eta\exp\left\{-\frac{1}{2}(\mu_b(\mathbf{y})-\mu'_b)^T\hat{\Sigma}_b^{-1}(\mathbf{y})(\mu_b(\mathbf{y})-\mu'_b)\right\},$$

where $\hat{\Sigma}_b^{-1}(\mathbf{y}) = (\Sigma_b^{-1}(\mathbf{y}) + (\Sigma'_b)^{-1})$.

A Taylor series expansion about the histogram $n(\mathbf{y}_0)$ and mean vector $\mu(\mathbf{y}_0)$ about the current location yields a linear approximation to the coefficient:

$$\rho(\mathbf{y}) \approx \rho(\mathbf{y}_0) + \Gamma_n(\mathbf{y};\mathbf{y}_0) + \Gamma_\mu(\mathbf{y};\mathbf{y}_0),$$

where

$$\Gamma_n(\mathbf{y};\mathbf{y}_0) = [n(\mathbf{y})-n(\mathbf{y}_0)]^T\frac{\partial\rho}{\partial n}(\mathbf{y}_0)$$
$$= \frac{1}{2}\sum_{b=1}^{B}\psi_b(\mathbf{y}_0)\sqrt{\frac{n'_b}{n_b(\mathbf{y}_0)}}n_b(\mathbf{y}) - \frac{1}{2}\rho(\mathbf{y}_0)$$

and

$$\Gamma_\mu(\mathbf{y};\mathbf{y}_0) = [\mu(\mathbf{y})-\mu(\mathbf{y}_0)]^T\frac{\partial\rho}{\partial\mu}(\mathbf{y}_0)$$
$$= \sum_{b=1}^{B}\psi_b(\mathbf{y}_0)\sqrt{n'_b n_b(\mathbf{y}_0)}\cdots$$
$$\cdot(\mu'_b-\mu_b(\mathbf{y}_0))\hat{\Sigma}_b^{-1}(\mathbf{y}_0)(\mu_b(\mathbf{y})-\mu_b(\mathbf{y}_0)).$$

Substituting Equation (2) and taking the derivative with respect to $\mathbf{y}$ yields

$$\frac{\partial\Gamma_n}{\partial\mathbf{y}} = \frac{C_h}{h^2}\sum_{b=1}^{B}\psi_b(\mathbf{y}_0)\sqrt{\frac{n'_b}{n_b(\mathbf{y}_0)}}\sum_{i=1}^{N_h}k(\cdot)\delta_{ib}(\mathbf{y}-\mathbf{x}_i)$$
$$= \sum_{i=1}^{N_h}\alpha_i k\left(\| \frac{\mathbf{y}_0-\mathbf{x}_i}{h} \|^2\right)(\mathbf{y}-\mathbf{x}_i)$$
$$\frac{\partial\Gamma_\mu}{\partial\mathbf{y}} = -\sum_{b=1}^{B}\psi_b(\mathbf{y}_0)\sqrt{n'_b n_b(\mathbf{y}_0)}\hat{\Sigma}_b^{-1}(\mathbf{y}_0)(\mu'_b-\mu_b(\mathbf{y}_0))$$

where

$$\alpha_i = \frac{C_h}{h^2}\sum_{b=1}^{B}\psi_b(\mathbf{y}_0)\sqrt{\frac{n'_b}{n_b(\mathbf{y}_0)}}\delta_{ib}.$$

Putting it all together, we set $\frac{\partial\rho}{\partial\mathbf{y}}$ to zero and solve for $\mathbf{y}$:

$$\mathbf{y}_1 = \frac{\sum_{i=1}^{N_h}\alpha_i g\left(\| \frac{\mathbf{y}_0-\mathbf{x}_i}{h} \|^2\right)\mathbf{x}_i - \sum_{b=1}^{B}\mathbf{v}_b}{\sum_{i=1}^{N_h}\alpha_i g\left(\| \frac{\mathbf{y}_0-\mathbf{x}_i}{h} \|^2\right)},$$

where

$$\mathbf{v}_b = \psi_b(\mathbf{y}_0)\sqrt{n'_b n_b(\mathbf{y}_0)}\hat{\Sigma}_b^{-1}(\mathbf{y}_0)(\mu'_b-\mu_b(\mathbf{y}_0)).$$

As before, if we use the Epanechnikov kernel profile then the derivative of the kernel is constant and disappears:

$$\mathbf{y}_1 = \frac{\sum_{i=1}^{N_h}\alpha_i\mathbf{x}_i - \sum_{b=1}^{B}\mathbf{v}_b}{\sum_{i=1}^{N_h}\alpha_i}.$$

[1] Note: By subtracting one from the denominator of the second equation, an unbiased estimate would be obtained.

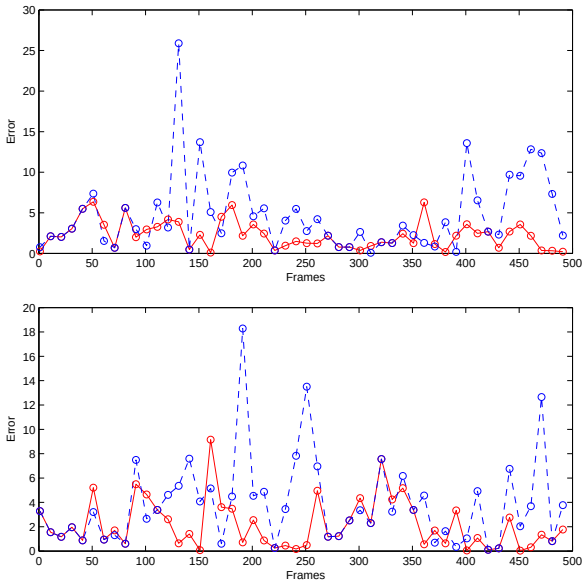


Figure 2: Tracking error in x and y using histograms (blue, dashed) versus spatiograms (red, solid) with exhaustive local search.

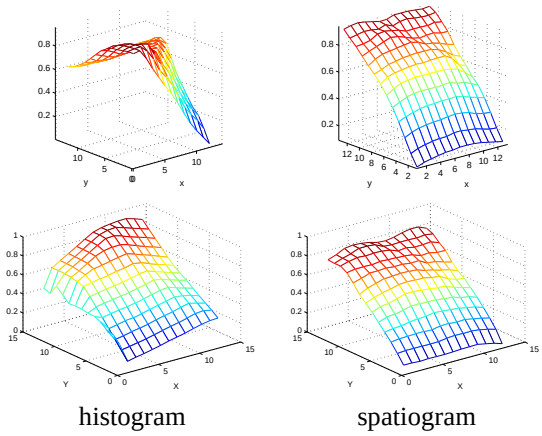


Figure 3: The likelihood function at a single scale for frame 396 (top) and 464 (bottom) of Experiment 2. The spatiogram produces a consistent likelihood, whereas the histogram likelihood is distracted by background pixels.

To emphasize the similarity between this equation and Equation (10), we can rearrange the numerator to obtain

$$\mathbf{y}_1 = \frac{\sum_{i=1}^{N_h} (\alpha_i \mathbf{x}_i - \hat{\mathbf{v}}_i)}{\sum_{i=1}^{N_h} \alpha_i}, \tag{11}$$

where

$$\hat{\mathbf{v}}_i = \frac{\sum_{b=1}^{B} \mathbf{v}_b \delta_{ib}}{\sum_{j=1}^{N_h} \delta_{jb}}.$$

The relationship between the equations is now evident. In Equation (10) each pixel $\mathbf{x}_i$ casts a vote proportional to w_i in the direction of $\mathbf{x}_i$. In Equation (11) each pixel casts a vote proportional to $\| \alpha_i \mathbf{x}_i - \hat{\mathbf{v}}_i \|$ in the direction of $\mathbf{x}_i - \hat{\mathbf{v}}_i / \alpha_i$.

4 Experimental results

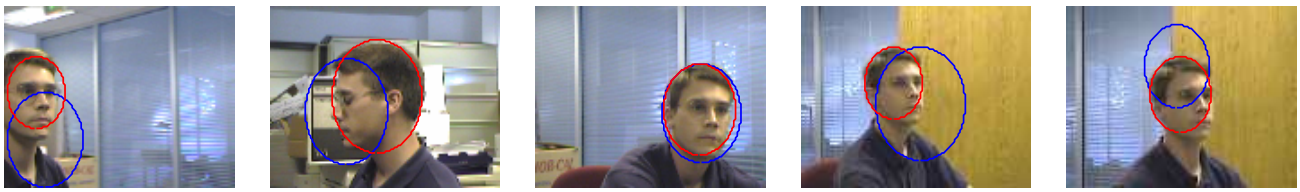
To test the effectiveness of spatiograms compared with histograms, we conducted three experiments using the image sequences available at `http://www.ces.clemson.edu/~stb/research/headtracker`. In all experiments the two techniques were manually initialized to the same location in the first image; the color space used was $B - G$, $G - R$, and $(B + G + R)/3$, as in [1]; and scale changes were handled by searching $\pm$ 10% in scale each image frame and selecting the scale producing the maximum score, as in [3]. The results are shown in Figure 4.

In the first experiment, the mean shift algorithm was run on the sequence 'seq_sb'. From frames 1 to 22 there is no significant difference between the two techniques. Starting at frame 23, histograms are distracted by the shirt as the subject turns his head, while the spatiogram ellipse remains on the head. At frame 37 the subject begins to move to the right, causing both ellipses to move in that direction as well. Then a quick jerk of the head at frame 43 loses both ellipses permanently. We have found that, although mean shift is exceptionally adept at rejecting outliers and maintaining a good lock on the target when the motion is well-behaved, it is not able to cope with quick, drastic accelerations.

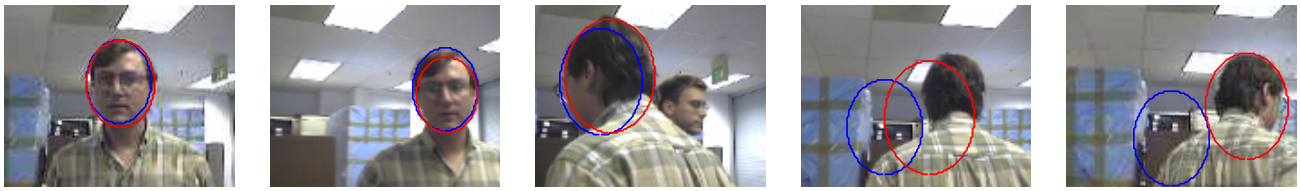
In order to compare the techniques on the full sequence, in the second experiment we ran the exhaustive search method described in [1] with a $\pm 6 \times \pm 6 \times \pm 1$ search window in x, y, and scale. In addition to the color histogram/spatiogram, a gradient dot product module was also used (this was necessary for the tracker to succeed). The absolute error for every tenth frame, using manually determined ground truth, is shown in Figure 2. Although there are a few frames in which the histogram does better, overall the spatiogram is the clear winner, exhibiting a mean-squared error of 0.8 and 0.9 pixels (in x and y), compared with 4.64 and 2.83 pixels (in x and y) for histograms. The spatiogram is less distracted by surrounding skin-colored objects, as seen by comparing the images of Figure 4 with the likelihood functions of Figure 3.

In the third experiment we ran the same code as the preceding experiment on the sequence 'seq_mg'. On this sequence the spatiogram successfully tracks the head, while the histogram slides off the target completely. This sequence is one of the few we have found in which the spatiogram actually succeeds where the histogram fails. In most of the sequences we have tried, both techniques either succeed or fail together, although the spatiogram maintains

Experiment 1 (mean shift algorithm)



Experiment 2 (exhaustive local search)



Experiment 3 (exhaustive local search)

Figure 4: Tracking results for three experiments using histograms (blue) versus spatiograms (red). Shown are frames 7, 27, 32, 37, 43 (top); 132, 192, 256, 396, 464 (middle); and 5, 15, 25, 28, 30 (bottom).

a more accurate lock on the target.

5. Conclusion

We have presented a novel concept that extends the familiar histogram in a natural way by capturing a limited amount of spatial information between the pixels contributing to the histogram bins. This spatiogram, as we call it, is a generalization of a histogram to higher-order moments. We have derived a mean shift procedure to track an object using spatiograms, and we have demonstrated improved tracking results when compared with histograms. Future work should be aimed at better characterizing the reasons behind the success of the spatiogram, especially since it is not specifically designed to handle object pose changes.

References

[1] S. Birchfield. Elliptical head tracking using intensity gradients and color histograms. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 232–237, 1998.

[2] R. T. Collins. Mean-shift blob tracking through scale space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[3] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564–577, May 2003.

[4] G. D. Hager, M. Dewan, and C. V. Stewart. Multiple kernel tracking with SSD. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2004.

[5] J. Ho, K.-C. Lee, M.-H. Yang, and D. Kriegman. Visual tracking using learned subspaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 782–789, 2004.

[6] A. Rosenfeld and A. Kak. *Digital Picture Processing*. San Diego: Academic Press, 1982.

[7] J. Shi and C. Tomasi. Good features to track. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[8] L. Sigal, S. Sclaroff, and V. Athitsos. Estimation and prediction of evolving color distributions for skin segmentation under varying illumination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2000.

[9] M. Swain and D. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[10] Y. Wu and T. S. Huang. A co-inference approach to robust visual tracking. In *Proceedings of the 8th International Conference on Computer Vision*, pages 26–33, 2001.

[11] Z. Zivkovic and B. Kröse. An EM-like algorithm for color-histogram-based object tracking. In *CVPR*, 2004.