Visual Tracking by Affine Kernel Fitting Using Color and Object Boundary

Ido Leichter, Michael Lindenbaum and Ehud Rivlin
Computer Science Department
Technion - Israel Institute of Technology
Haifa 32000, Israel
{*idol,mic,ehudr*}*@cs.technion.ac.il*

Abstract

Kernel-based trackers aggregate image features within the support of a kernel (a mask) regardless of their spatial structure. These trackers spatially fit the kernel (usually in location and in scale) such that a function of the aggregate is optimized.

We propose a kernel-based visual tracker that exploits the constancy of color and the presence of color edges along the target boundary. The tracker estimates the best affinity of a spatially aligned pair of kernels, one of which is color-related and the other of which is object boundary-related. In a sense, this work extends previous kernel-based trackers by incorporating the object boundary cue into the tracking process and by allowing the kernels to be affinely transformed instead of only translated and isotropically scaled. These two extensions make for more precise target localization. Moreover, a more accurately localized target facilitates safer updating of its reference color model, further enhancing the tracker's robustness. The improved tracking is demonstrated for several challenging image sequences.

1. Introduction

Kernel-based visual trackers (e.g., [4]) spatially fit a kernel (usually in location and in scale) over the image plane such that an objective function of the aggregate of image features under the kernel support is optimized. The spatial structure of features usually changes over time, so that abstaining from reliance on such a structure helps to make the tracker more robust to these changes. In kernel-based tracking, no attempt is made to find the exact boundary of the target, and all the pixels in the region where the kernel is fitted affect very few kernel parameters (usually only center and scale). Furthermore, no explicit relation between the kernel and the target shape is required, except that the kernel support be roughly where the target is located.

Although kernel-based trackers are robust, they usually transform the kernel only by translation and isotropic scaling. Such a coarse object localization may be of little significance when the change of target's shape in the image is poorly approximated by these three parameters (consider a rotating stick or a partial occlusion of a significant portion of the target). More seriously, this coarse localization may render the target region such a small portion of the kernel support (or vice versa) that the target will be lost. In addition, a too coarse localization is a major obstacle to updating the target's appearance model, though this is generally desired.

This paper tackles the aforementioned problems by proposing a kernel-based tracker that tracks spatial affine transformations. Since affine localization is usually too refined to be identified well by color alone, a spatially aligned pair of kernels, one related to target and background colors and the other to the object boundary, are used in conjunction. The two corresponding spatially normalized kernels are rotationally symmetric with respect to all rotations, which reduces their affine fitting to 5 dimensions (instead of 6). As we demonstrate, the refinement of the target localization to affinities enables us to more safely update the reference color histograms of both the target and the background, which further enhances the tracker's robustness.

The paper proceeds by describing previous work in Section 2, followed by a detailed description of the proposed affine kernel fitting tracker in Section 3. Section 4 presents experimental results and Section 5 summarizes the paper.

2. Previous Work

An early kernel-based visual tracker is the CAMSHIFT [2]. It tracked human faces by assigning each pixel a positive weight reflecting the incidence of the pixel color in a reference color histogram of the face, and finding the location of an axis-aligned rectangular window (the kernel) in which the the total weight of the pixels in the window is maximal. The search for the window location was performed quickly via mean-shift iterations [7]. The window localization was repeated several times, each time after isotropically scaling the size of the window according to some function of the sum of the pixel weights inside it, until some heuristic criterion was met. A similar tracking procedure performing a single mean-shift iteration was developed in [12].

A kernel-based tracker of heads was proposed in [1]. In this work a uniform kernel of an elliptical support was used to exhaustively search (in a region of the state space centered at the predicted state) for the location and isotropic scale of the head by comparing the color histogram under the support of the candidate kernel with a reference color histogram. Since the ellipse approximated to the head's shape in the image quite well, the score could also take into account the gradients along the candidate ellipse, with each gradient projected into the normal to the ellipse at the corresponding pixel. The kernel used in this tracker, as well as in the two aforementioned trackers, was uniform. As will be explained in Section 3.1, kernels that decrease from their center, like the ones used in the subsequent references, are better suited for tracking.

A kernel-based tracker minimizing a Bhattacharyya coefficient-based distance between the reference color distribution of the target and the target's color distribution in the current frame was formulated in [4]. The search for the best kernel location was performed by the mean-shift procedure three times per frame: once with the kernel scale estimated in the previous frame, once with the scale enlarged, and once with the scale reduced. The scale that produced the smallest distance was chosen to be input to an IIR filter used to derive the new scale. Extensions that incorporate background information and Kalman prediction, as well as an application to face tracking, were also proposed. A method for choosing the correct scale for mean-shift blob tracking by adapting Lindeberg's theory of feature scale selection was proposed in [3]. An extension that updates the reference color histogram by Kalman filtering each histogram bin, followed by hypothesis testing, was suggested in [17].

Kernel-based tracking that minimizes the Matusita distance between the feature distributions using a Newton-style method, as well as the extension to the use of multiple kernels, were developed in [8]. This last work implemented a location and scale tracker as well as a wand tracker. An extension to multiple kernel-tracking where the different trackers collaborate by utilizing the state constraints was proposed in [6].

Instead of fix-point estimation of kernel transformation parameters in the work above, a PDF (probability distribution function) of the transformation parameters (location and isotropic scale) was tracked in [15] via CONDENSATION [10].

Tracking enhancement by using multiple cue modalities has also been performed in the past. In [16] the aforementioned color-based tracker [15] was enhanced by fusion with stereo sound or motion detection with a still camera. A particle filter, in which cascaded Adaboost detections were used to generate the proposal distribution and color histograms were used for the likelihood, was used to track hockey players in [14]. In [18] two trackers, a region tracker and an edge tracker, ran in parallel and performed mutual corrections based on their confidence measures. Another example are the co-inference algorithms, developed in [19] to combine trackers of different modalities.

Another related work is [5, 21], where, rather than completely ignoring the spatial structure of the image features in the target, the spatial constraint is only relaxed. This is achieved by modeling the feature-spatial joint probability of a region by a multivariate kernel density estimation.

We would like to note that in addition to the target's location and scale, the kernel-based trackers in [12] and [2] tried to estimate the vertical and horizontal scale of the target, and the latter also the rotation. However, the estimation was performed *after* the target localization was completed using a few heuristics. The tracker in [12] was used also in [13], where a mixture-of-Gaussians color model of the target was selectively adapted over time. Finally, we observe that affine object tracking was also performed in [22]. However, in contrast to the work here, the appearance model used there consisted of spatial structure through the use of the spatial-color representation of [5], no boundary-related cues were used, the (implicitly used) kernel was uniform, and no attempt was made to update the target's appearance model. These have probably limited all the targets in the presented experiments to be rigid or nearly rigid and wholly visible.

3. Affine Kernel Fitting

Using the framework of kernel-based tracking, the tracker proposed here transforms, in each video frame $\mathbf{I}(\mathbf{x})$, a spatially aligned pair of (spatially) *normalized kernels* $K^{\text{color}}(\mathbf{x})$ and $K^{\text{edge}}(\mathbf{x})$, color-related and object boundary-related, respectively. ($\mathbf{x} = (x, y)^T$ denotes coordinates.) The transformation parameters $\hat{\mathbf{p}}$ are estimated such that a certain scoring function is maximized:

$$\hat{\mathbf{p}} = \arg\max_{\mathbf{p}} S^{\text{color}}\left(\mathbf{I}, K^{\text{color}}_{\mathbf{p}}\right) + \alpha \cdot S^{\text{edge}}\left(\mathbf{I}, K^{\text{edge}}_{\mathbf{p}}\right), \quad (1)$$

where S^{color} and S^{edge} are the color-related and boundary-related score components, respectively; $K^{\text{color}}_{\mathbf{p}}(\mathbf{x})$ and $K^{\text{edge}}_{\mathbf{p}}(\mathbf{x})$ are the *transformed kernels* with transformation parameters $\mathbf{p}$; and α is a parameter regulating the weight of the boundary-related score component.

In the proposed tracker, the geometric transformation applied on the normalized kernels is *affine*. In an affine transformation, the relation between a point $\mathbf{x}$ and its corresponding point $\mathbf{x}'$ in the transformed coordinates may be written as [9]:

$$\mathbf{x}' = R(\phi)D(\lambda_x, \lambda_y)R(\theta)\mathbf{x} + \mathbf{t}, \quad (2)$$

where $R(\gamma) = \begin{pmatrix} \cos\gamma & -\sin\gamma \\ \sin\gamma & \cos\gamma \end{pmatrix}$ is a γ-rotation matrix, $D(\lambda_x, \lambda_y) = \begin{pmatrix} \lambda_x & 0 \\ 0 & \lambda_y \end{pmatrix}$ is a scaling/reflection matrix, and $\mathbf{t} = (t_x, t_y)^T$ is a translation vector. Now, since the normalized kernels used here are, as usual, rotationally symmetric with respect to all rotations, the second rotation of the coordinate system (corresponding to the multiplication by $R(\phi)$ in (2)) and any reflection (corresponding to a negative λ_x or λ_y) may be omitted by a proper adjustment of the translation vector $\mathbf{t}$. Thus, the problem of estimating the 6 transformation parameters of the kernel pair is reduced to that of estimating the 5 parameters, $\mathbf{p} = (\lambda_x, \lambda_y, \theta, t_x, t_y)$, of the coordinate transformation

$$T_{\mathbf{p}}(\mathbf{x}) = \begin{pmatrix} \lambda_x \cos\theta & -\lambda_x \sin\theta \\ \lambda_y \sin\theta & \lambda_y \cos\theta \end{pmatrix} \mathbf{x} + (t_x, t_y)^T, \quad \lambda_x, \lambda_y > 0. \quad (3)$$

Following the above, we define the transformed version of $K^{\text{color}}(\mathbf{x})$ with transformation parameters $\mathbf{p}$ as

$$K^{\text{color}}_{\mathbf{p}}(\mathbf{x}) = \lambda_x \lambda_y K^{\text{color}}\left(T_{\mathbf{p}}(\mathbf{x})\right) \quad (4)$$

and similarly for $K^{\text{edge}}_{\mathbf{p}}(\mathbf{x})$. The kernel normalization term $\lambda_x\lambda_y$, which is the Jacobian determinant of $T_{\mathbf{p}}$, guarantees that the volume of the transformed kernel will be constant for all transformations. This normalization term is the generalization of the normalization constant C_h in [4] to affinities. As will be evident in what follows, this normalization is required if the scoring function is to be unbiased with respect to scale.

It is easy to show that the level-sets of the normalized kernels, which are circular, are transformed into *elliptical* level-sets in the transformed kernels. In particular, when the transformation parameters of the transformed kernel are written as

$$\mathbf{p}\left(l_x, l_y, \theta, o_x, o_y\right) = \Big(2l_x^{-1}, 2l_y^{-1}, -\theta, \\ -2l_x^{-1}\left(o_x\cos\theta + o_y\sin\theta\right), \\ 2l_y^{-1}\left(o_x\sin\theta - o_y\cos\theta\right)\Big), \quad (5)$$

the circular level-set of unit radius in the normalized kernel is transformed in the transformed kernel into an ellipse with axes of lengths l_x and l_y, centered at (o_x, o_y) and rotated by θ about its center (l_x-length axis parallel to the x-axis before the rotation).

3.1. Color-Related Score Component

Denote the discrete color PDFs of the target and the background by $p^{\text{tar}}(\mathbf{c})$ and $p^{\text{bg}}(\mathbf{c})$ ($\mathbf{c}$ denotes color), respectively, and the color at pixel location $\mathbf{x}$ by $\mathbf{I}(\mathbf{x})$. When a pixel color is considered as being drawn either from the target's color PDF or from the background's, the probability that a pixel location $\mathbf{x}$ belongs to the target is higher than the probability it belongs to the background by

$$\frac{p^{\text{tar}}\left(\mathbf{I}(\mathbf{x})\right) - p^{\text{bg}}\left(\mathbf{I}(\mathbf{x})\right)}{p^{\text{tar}}\left(\mathbf{I}(\mathbf{x})\right) + p^{\text{bg}}\left(\mathbf{I}(\mathbf{x})\right)}$$

(assuming equal prior probabilities). This probability difference is used to assign each pixel location a weight

$$w_{\mathbf{x}} = \max\left\{\frac{p^{\text{tar}}\left(\mathbf{I}(\mathbf{x})\right) - p^{\text{bg}}\left(\mathbf{I}(\mathbf{x})\right)}{p^{\text{tar}}\left(\mathbf{I}(\mathbf{x})\right) + p^{\text{bg}}\left(\mathbf{I}(\mathbf{x})\right)}, 0\right\}. \quad (6)$$

(See Section 3.1.1 for the approximated calculation of these color PDFs.) This weight is positive for pixels whose color is more prominent in the target's color PDF than in the background's, and the greater the difference in the color's prominence, the higher the weight. Note that in practice there might be pixels of colors that are not present in either of the two color PDFs ($p^{\text{tar}}(\mathbf{c}) = p^{\text{bg}}(\mathbf{c}) = 0$), so the denominator in (6) should be summed with a small constant to circumvent the possibility of a division by zero.

The color-related score component for transformation parameters $\mathbf{p}$ is set to

$$S^{\text{color}}(\mathbf{I}, \mathbf{p}) = \sum_{\mathbf{x}} K^{\text{color}}_{\mathbf{p}}(\mathbf{x}) \cdot w_{\mathbf{x}}. \quad (7)$$

This score has the same form as the objective function derived in [4] (with different weights), where the normalization term C_h in [4] corresponds to the kernel normalization term $\lambda_x\lambda_y$ in (4). Note that without this kernel normalization, the color-related score component will grow constantly with the growth in scale.

Since the target is not exactly elliptical and its peripheral pixels are often affected by occlusions or interference from the background [4], the pixel weights (6) are expected to be smaller farther from the target center. As the authors of [4] did, we choose the Epanechnikov kernel (which is rotationally symmetric for all rotations and monotonically decreasing in the distance from its center). Then, in order to make the kernel more adaptive to increases in scale, we extend it symmetrically by negative weights. (Such an approach was also used in [3].) Thus, the color-related normalized kernel we use is[1]

$$K^{\text{color}}(\mathbf{x}) = \left(1 - \|\mathbf{x}\|^2\right) \cdot \mathbf{1}_{\{\mathbf{v}:\|\mathbf{v}\|\le 1\}}(\mathbf{x}) \\ - \frac{\left(d^2 - (1 + d - \|\mathbf{x}\|)^2\right)}{\frac{16}{3}(d+1)d^3} \cdot \mathbf{1}_{\{\mathbf{v}:1\le\|\mathbf{v}\|\le 1+2d\}}(\mathbf{x}), \quad (8)$$

where d is a parameter indicating half the width of the ring-shaped domain where the kernel is negative (see Fig. 1). Note that the positive and negative parts of the kernel are of equal volume. In all experiments we set $d = 0.1$.

[1] $\mathbf{1}_{\mathcal{A}}(\mathbf{x}) : \mathcal{X} \to \{0, 1\}$ is the *indicator function* of subset $\mathcal{A} \subseteq \mathcal{X}$. This function equals 1 for $\mathbf{x} \in \mathcal{A}$ and 0 for $\mathbf{x} \notin \mathcal{A}$.

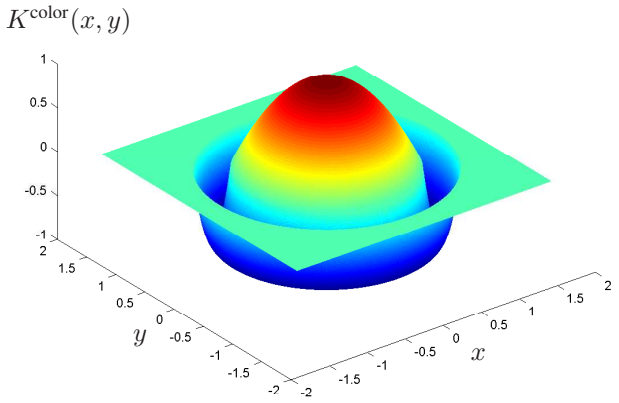

Figure 1: The color-related normalized kernel ($d = 0.2$).

3.1.1. Color PDF Approximation.

The computation of the pixel weights (6) requires the discrete color PDFs of the target and the background. These two color PDFs are approximated using the *previous* video frame and the previously estimated transformation parameters of the kernel pair for that frame. This allows the color PDFs of the target and the background to gradually change over time – which is not possible when using only the approximation of the color PDFs as they appear in the first video frame. Although more sophisticated mechanisms for updating the target model have been proposed (e.g, [13, 11, 17]), the affine target localization achieved here allowed the use of this naive update of the color model. We would like to note that no experimental advantages resulted from using an IIR filter operating also on the histograms estimated before the previous frame.

The target's color PDF is approximated using the color *histogram* in the elliptical region corresponding to the positive part of the estimated transformed color-related kernel. As previously explained, the peripheral pixels are the least reliable for estimating the target's color histogram, and therefore the pixels' influence on the approximated histogram should decrease with the increase in distance from the target (kernel) center. Thus, we use the Epanechnikov kernel (which was also used in the approximation of the target's color histogram in [4]), spatially transformed according to the estimated transformation parameters $\hat{\mathbf{p}}$ (see Fig. 2(a) for the normalized kernel). The approximated color histogram of the target is thus

$$h^{\text{tar}}(\mathbf{c}) = C \sum_{\mathbf{x}} \left(1 - \|T_{\hat{\mathbf{p}}}(\mathbf{x})\|^2\right) \cdot \mathbf{1}_{\{\mathbf{v}:\|\mathbf{v}\|\le 1\}}\left(T_{\hat{\mathbf{p}}}(\mathbf{x})\right) \cdot \delta\left(\mathbf{I}(\mathbf{x}) - \mathbf{c}\right), \quad (9)$$

where δ is the Kronecker delta function, and C is a scalar normalizing the histogram to unit sum.

Finally, because factors such as noise and illumination variance may cause the pixel colors to drift between consecutive frames, the target's discrete color PDF $p^{\text{tar}}(\mathbf{c})$ is approximated as a smoothed version of the target's color histogram $h^{\text{tar}}(\mathbf{c})$. When the colors are considered as vectors of several color components (e.g., RGB), the discrete domain of the histogram has one dimension per color component (e.g., 3 for RGB). The smoothing is performed by convolving the color histogram with a 1D triangular kernel, normalized to unit sum, in each dimension of the color histogram.

The color quantization into bins performed in [4] may be viewed as a smoothing of the color histogram followed by its sampling. Under this view, that smoothing is not shift-invariant and uses an asymmetric kernel. The histogram smoothing performed here is actually a shift-invariant version of the previous (implicit) smoothing, where here the smoothing kernel is symmetric. This smoothing version simulates the color changes (that result from noise and illumination changes) more appropriately.

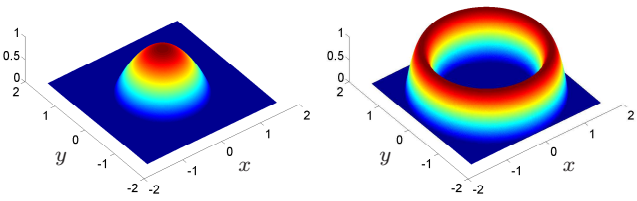

Figure 2: The normalized kernels used for weighting the pixels in the color histograms of the target (a) and the background (b).

The background's color PDF is approximated in the same manner as the target's, using the pixels in a region around the target (similar to the region in [4]). On one hand, the closer the pixel is to the target, the less reliable it is for approximating the background's color PDF. On the other hand, the farther the pixel is from the target, the less relevant it is for approximating the color PDF of the background surrounding the target. We therefore have to use a rotationally symmetric kernel of a ring-shaped support around the target, spatially transformed according to the estimated transformation parameters $\hat{\mathbf{p}}$. The normalized kernel we use is (see illustration in Fig. 2(b)):

$$\left(1 - (4/3 - 9\|\mathbf{x}\|)^2\right) \cdot \mathbf{1}_{\{\mathbf{v}:1\le\|\mathbf{v}\|\le 5/3\}}(\mathbf{x}). \quad (10)$$

3.2. Boundary-Related Score Component

Object boundaries are usually associated with high-magnitude gradients, which are perpendicular to the orien-

tation of the projected object edges. Therefore, denoting by $\mathbf{I}_c(\mathbf{x})$ the c-th color component at pixel location $\mathbf{x}$ (e.g., $c = 1, 2, 3$ for RGB), we set the boundary-related score component for transformation parameters $\mathbf{p}$ to [2]

$$S^{\text{edge}}(\mathbf{I}, \mathbf{p}) = \sum_{\mathbf{x}} K^{\text{edge}}_{\mathbf{p}}(\mathbf{x}) \sum_{c} \left| \left\langle \nabla \mathbf{I}_c(\mathbf{x}), \widehat{\nabla K^{\text{edge}}_{\mathbf{p}}(\mathbf{x})} \right\rangle \right| \tag{11}$$

(the caret denotes here normalization to unit length), where the boundary-related normalized kernel is

$$K^{\text{edge}}(\mathbf{x}) = \left(1 - 16\left(1 - \|\mathbf{x}\|\right)^2\right) \cdot \mathbf{1}_{\{\mathbf{v}:3/4<\|\mathbf{v}\|<5/4\}}(\mathbf{x}). \tag{12}$$

Note that the first summation may consist of only pixel locations where the transformed boundary-related kernel is strictly positive. The kernel is illustrated in Fig. 3. This score component rewards for color gradients indicating the presence of edges that are in proximity to the boundary of the positive part of the transformed color-related kernel and of orientation similar to the boundary's. The higher the gradient size, the higher the score; the closer the implied edge to this boundary and the closer this edge direction to the direction of the boundary tangent, the higher the score. This score component may be viewed as a kernel-based extension of the gradient module used in [1].

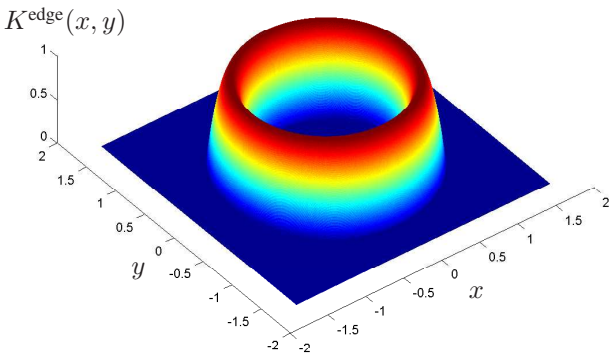

Figure 3: The boundary-related normalized kernel.

The gradient of the transformed boundary-related kernel, where the kernel is positive, is

$$\nabla K^{\text{edge}}_{\mathbf{p}}(\mathbf{x}) = k \cdot \begin{pmatrix} \lambda_x \cos\theta & \lambda_y \sin\theta \\ -\lambda_x \sin\theta & \lambda_y \cos\theta \end{pmatrix} T_{\mathbf{p}}(\mathbf{x}), \tag{13}$$

where k is a scalar (depending on $\mathbf{x}$ and $\mathbf{p}$), and therefore has no bearing on the calculation of the normalized gradient

$$\widehat{\nabla K^{\text{edge}}_{\mathbf{p}}(\mathbf{x})} = \frac{\nabla K^{\text{edge}}_{\mathbf{p}}(\mathbf{x})}{\left\| \nabla K^{\text{edge}}_{\mathbf{p}}(\mathbf{x}) \right\|}. \tag{14}$$

The image gradients $\nabla \mathbf{I}_c(\mathbf{x})$ at pixel locations are estimated using the simple discrete approximations

$$\nabla \mathbf{I}_c(x, y) \triangleq \frac{1}{2} \begin{pmatrix} \mathbf{I}_c(x+1, y) - \mathbf{I}_c(x-1, y) \\ \mathbf{I}_c(x, y+1) - \mathbf{I}_c(x, y-1) \end{pmatrix}. \tag{15}$$

Note that the boundary-related score component does not use an edge-related reference model such as an edge orientation histogram [20].

3.3. Score Maximization

In order to estimate the transformation parameters of the kernel pair, the scoring function in (1) has to be maximized. After experimenting with several maximization methods, we chose a "coarse-to-fine" search as the best method for finding the global maximum of the scoring function. In this method the 5 parameters $\mathbf{p}_{\text{ellipse}} = (l_x, l_y, \theta, o_x, o_y)$ of the elliptical level-set corresponding to the circular level-set of unit radius in the normalized kernels are sought by a "coarse-to-fine" search in the 5-dimensional space of ellipse parameters: center (o_x, o_y), rotation θ, and lengths of horizontal and vertical (before rotation) axes l_x and l_y, respectively. The search starts from the estimated ellipse in the previous frame, but it may of course start from a different ellipse if object dynamics is incorporated. The kernel transformation parameters $\mathbf{p}$ corresponding to the ellipse parameters $\mathbf{p}_{\text{ellipse}}$ are given in (5).

The exact maximization is performed as follows. First, the score is calculated for the estimated ellipse $\hat{\mathbf{p}}_{\text{ellipse}}$ in the previous video frame. Then, the score is calculated for the following 32 ellipses:

$$\hat{\mathbf{p}}_{\text{ellipse}} + (b_1 \cdot \Delta o_x,\ b_2 \cdot \Delta o_y,\ b_3 \cdot \Delta\theta,\ b_4 \cdot \Delta l_x,\ b_5 \cdot \Delta l_y), \quad b_i = \pm 1,$$

where $\Delta o_x = \Delta o_y = 0.5\Delta l_x = 0.5\Delta l_y = 4$ pixels and $\Delta\theta = 8°$. If the best ellipse (yielding the highest score) out of these 32 ellipses is better than $\hat{\mathbf{p}}_{\text{ellipse}}$, then $\hat{\mathbf{p}}_{\text{ellipse}}$ is changed to the best ellipse and the process is repeated. If not, the search radius is reduced to half by halving the five Δs, and the above process is repeated using the reduced Δs. When the search radius becomes small enough, the search is terminated. In our implementation the search is terminated

[2] The term $\widehat{\nabla K^{\text{edge}}_{\mathbf{p}}(\mathbf{x})}$ is used here as a means to indicate a perpendicular direction to the tangent of the elliptical level-sets of the kernel. Although such a direction exists at all locations where the kernel is positive, this term is undefined at locations corresponding to the highest level-set (as the gradient there is zero). However, this term is calculated only at locations on the pixel grid, so we neglect the possibility that a pixel location will correspond exactly to a point in this zero-area level-set.

when $\Delta o_x < 0.5$ pixels, that is, after halving the search radius 4 times.

Experiments show that the search time may be reduced by first coarsely optimizing (until $\Delta o_x < 1$) only the ellipse center and rotation parameters while keeping the axis lengths constant (and equal to the estimated ones in the previous frame), and only then optimizing all 5 parameters as described above. In practice, the score (1) per frame will be typically calculated 200 to 600 times using this method. Note that the color PDFs of the target and the background, the pixel weights (6), and the image gradients (15) are calculated only *once* per frame. We would like to point out that gradient ascent was shown in our experiments to require fewer score computations per frame, but it occasionally converged at a local maximum.

Discussion on CONDENSATION as an alternative: An alternative to the above score maximization may be the tracking of a probability distribution function over the space of kernel transformation parameters via CONDENSATION. This would allow multi-modal distributions to be dealt with. However, using CONDENSATION here would make the update of the target's color PDF too computationally heavy, since each particle (hypothesis) would have a different color PDF associated with its corresponding target. Moreover, even with a single, unchanged color PDF for the target, each particle would have a different color PDF for the background surrounding the target associated with it. This would require that the background's color PDF and the pixel weights, which depend on it, be estimated separately for each particle. Finally, even if the last computational problem would be solved, experiments show (e.g. [10]) that using CONDENSATION for tracking a shape in the space of affinities requires 100-1200 particles, which is in the same order of magnitude as the number of score computations performed per frame in the maximization method above.

4. Experimental Results

Experiments using four image sequences are discussed below. In all but one sequence the color space used was RGB with 128 equally spaced values in [0,1] in each color band. In Sequence II the color PDFs were estimated over the HSV color space (with the same dynamic range and discretization as in the RGB color space), since this allowed for better separation of target from background colors in this sequence. In all experiments with the proposed tracker, the support size of the 1D triangular kernel used to smooth the color histograms was 9. The boundary-related weight parameter α (Eq. (1)) was set in Sequences I-III to 1. Since the target's shape (in the image) in Sequence IV is very close to an ellipse, better results for this sequence were obtained by raising α to 2. The initialization was manually performed in the first frame. Video files of all results for the proposed tracker are given as supplementary material. The estimated ellipses are presented with each of their semiaxes enlarged by 3 pixels so that the target boundary will be clear.

Sequence I: The proposed tracker was tested on an image sequence, where both the target (a lighter) and the camera are arbitrarily translated and rotated. The target was successfully tracked (see Fig. 4).

Figure 4: Results of the proposed tracker for Sequence I.

The same experiment was repeated using the mean shift tracker [4]. Results are shown in Fig. 5. The shape of the target in the first frame is indeed approximated reasonably by an ellipse of some axis ratio and of axes parallel to the image boundaries. However, such a shape approximation is too crude for the rest of the sequence, as the target's shape changes throughout the sequence are far from being only of isotropic scale. Moreover, since the target is colored differently on different sides, its color PDF changes as it rotates. These two obstacles cause the poor performance of the mean shift tracker in this experiment.

To demonstrate the importance of updating the target's color PDF, the proposed tracker was run again from the middle of this sequence, with the update mechanism of the target's color PDF turned off. That is, the target's color PDF, which is used to calculate the pixel weights (6), remained equal to the one estimated in the first frame from which the

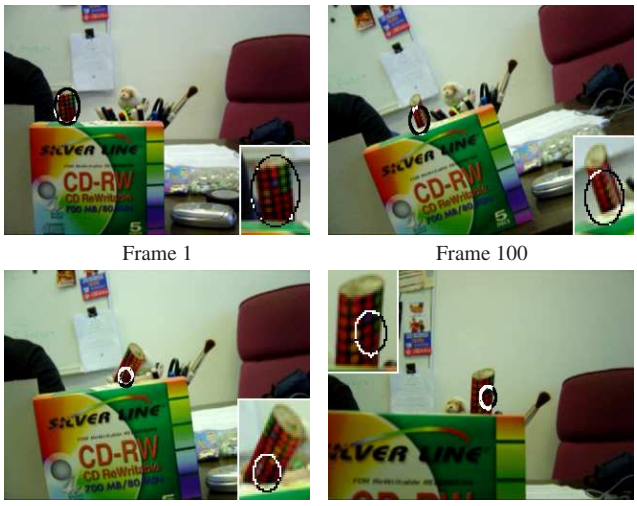

Figure 5: Results of the mean shift tracker [4] for Sequence I.

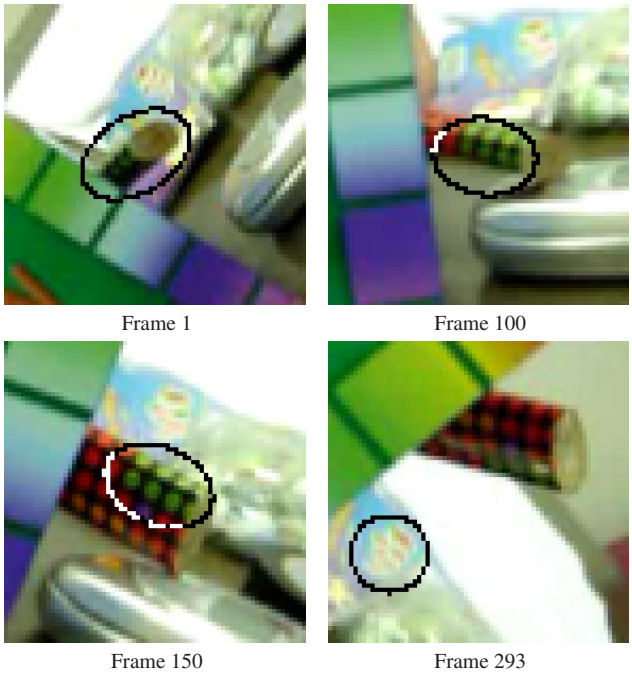

Figure 6: Results of the proposed tracker without updating the target's color PDF for a sub-sequence of Sequence I. Since the target colors that are revealed to the camera change with time, the tracking fails. (Only small, magnified portions of the frames are shown so that the target colors will be clearly seen.)

tracker was run. Due to the rotational motion of the target, the target's colors in this frame are very distinct from those in the subsequent frames, which causes the tracking failure (see Fig. 6).

Sequence II: Here the proposed tracker was tested on a non-rigid target (two people walking in a mall), filmed by a moving camera. The tracking succeeded, as may be seen in Fig. 7.

As in the previous sequence, the mean shift tracker [4] did not perform well here (see Fig. 8). The main reason for this is that the target shape changes from a vertical blob at the beginning of the sequence to a horizontal one near the end. As the mean shift tracker supports only isotropic scale changes, the former change in target shape could not be captured, which eventually caused the loss of the target.

Sequence III: In this sequence, filmed by a moving camera, a walking person is tracked by the proposed tracker. The tracking is robust until the person leaves the scene (see Fig. 9).

Note that the person's legs were left out from the estimated target as the sequence progressed. This is caused by the significant changes in the target's shape near the leg area. These shape changes are too great to be reasonably approximated by an affinity of the approximated ellipse of the first frame. A kernel transformation that would have included the person's legs inside the positive part of the color-related kernel (and hence inside the ellipse), would have also included large background areas, which would have lowered the score too much.

Sequence IV: To test the proposed tracker on a target of topologically (in the image plane) changing shape, we used a sequence consisting of a roll of cellotape rotating in front of an arbitrarily moving camera. The successful tracking is shown in Fig. 10.

Disregarding of Edges: Lastly, in order to demonstrate the contribution of the boundary-related kernel to the tracking, we repeated the above experiments without it ($\alpha = 0$). Several results are presented in Fig. 11. These tracking failures are a strong indication of the importance of the boundary-related kernel in the proposed tracker.

5. Conclusion

A new kernel-based visual tracker was proposed. The proposed tracker is enhanced with respect to previous kernel-based trackers in two ways. First, in addition to the constancy of color exploited by other kernel-based trackers, this tracker exploits the presence of color edges along the target boundary. Second, it spatially adjusts the kernels using affine transformations instead of using merely translation and isotropic scaling, to which other kernel-based trackers are restricted. These two enhancements make the target localization more accurate. Moreover, the refined target localization facilitates safer updating of the target's reference

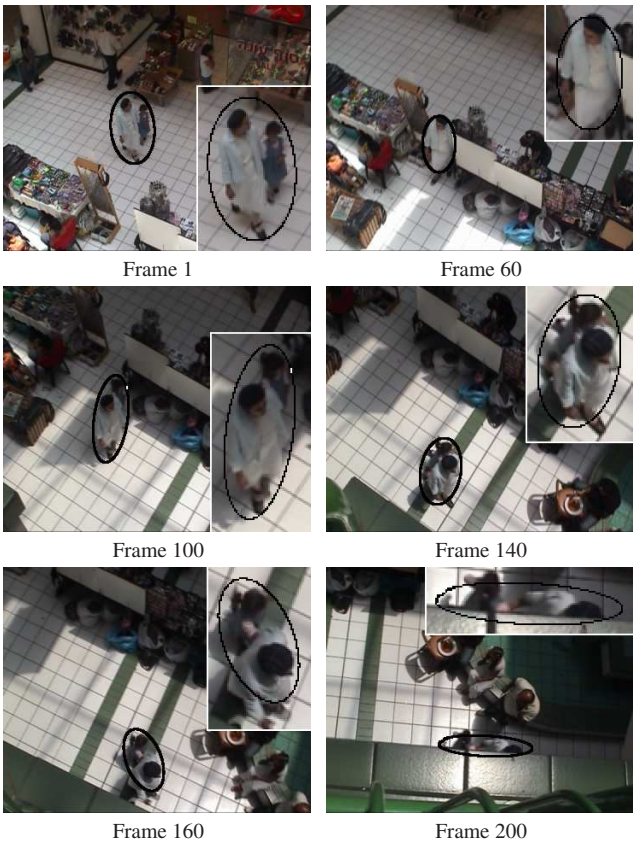

Figure 7: Results of the proposed tracker for Sequence II.

color PDF, further enhancing the tracker's robustness.

The tracking is performed by estimating the affinity of a spatially aligned pair of kernels, one color-related and the other object boundary-related, such that a scoring function is maximized. As was shown, the rotational symmetry of the spatially normalized kernel pair reduces the (normally 6-dimensional) space of affinities into a 5-dimensional one.

Experiments using several challenging image sequences demonstrate the high capability of the proposed tracker and its advantage over other kernel-based trackers.

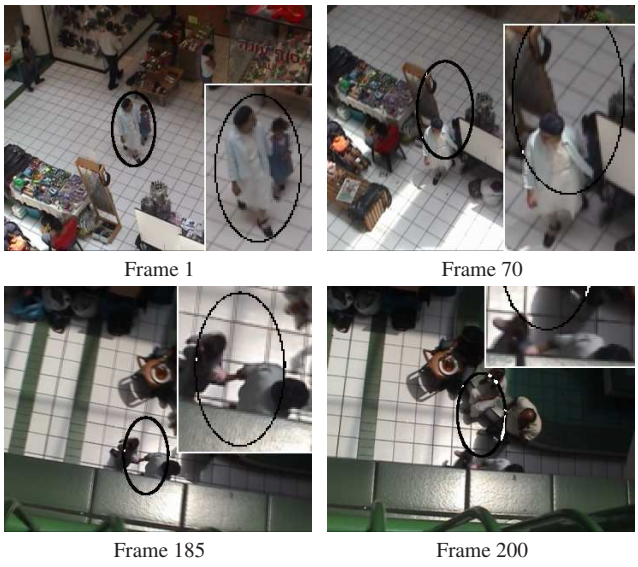

Figure 8: Results of the mean shift tracker [4] for Sequence II.

References

[1] S. Birchfield. Elliptical head tracking using intensity gradients and color histograms. *CVPR*, pages 232–237, 1998.

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

[3] R.T. Collins. Mean-shift blob tracking through scale space. *CVPR*, 2:234–240, 2003.

[4] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *PAMI*, 25(5):564–577, 2003.

[5] A. Elgammal, R. Duraiswami, and L.S. Davis. Probabilistic tracking in joint feature-spatial spaces. *CVPR*, 1:781–788, 2003.

[6] Z. Fan, Y. Wu, and M. Yang. Multiple collaborative kernel tracking. *CVPR*, 2:502–509, 2005.

[7] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, Boston, 2nd edition, 1990.

[8] G.D. Hager, M. Dewan, and C.V. Stewart. Multiple kernel tracking with SSD. *CVPR*, 1:790–797, 2004.

[9] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.

[10] M. Isard and A. Blake. CONDENSATION - conditional density propagation for visual tracking. *IJCV*, 29(1):5–28, 1998.

[11] A.D. Jepson, D.J. Fleet, and T.F. El-Maraghi. Robust online appearance models for visual tracking. *PAMI*, 25(10):1296–1311, 2003.

[12] S. McKenna, S. Gong, and Y. Raja. Face recognition in dynamic scenes. *BMVC*, pages 140–151, 1997.

[13] S.J. McKenna, Y. Raja, and S. Gong. Tracking colour objects using adaptive mixture models. *Image and Vision Computing*, 17:225–231, 1999.

[14] K. Okuma, A. Taleghani, N. de Freitas, J.J. Little, and S.G. Lowe. A boosted particle filter: Multitarget detection and tracking. *ECCV*, 1:28–39, 2004.

[15] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. *ECCV*, 1:661–675, 2002.

[16] P. Pérez, J. Vermaak, and A. Blake. Data fusion for visual tracking with particles. *Proceedings of the IEEE*, 92(3):495–513, 2004.

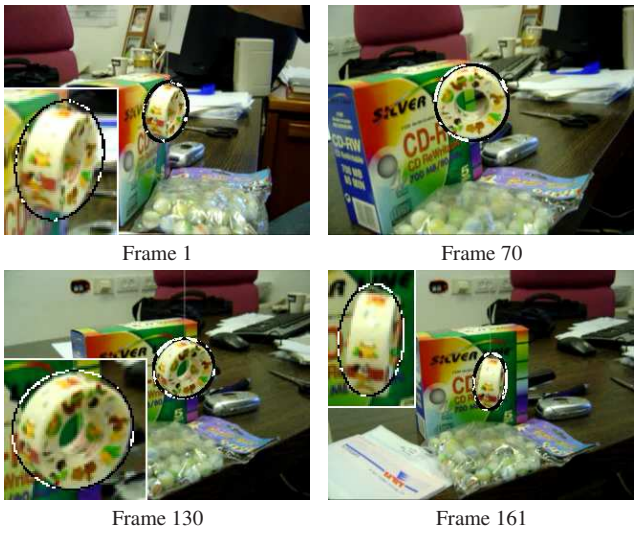

Figure 10: Results of the proposed tracker for Sequence IV.

Figure 11: Results obtained using the proposed tracker *without* its boundary-related kernel ($\alpha = 0$). It is evident that color edges play a crucial role in the proposed tracker.

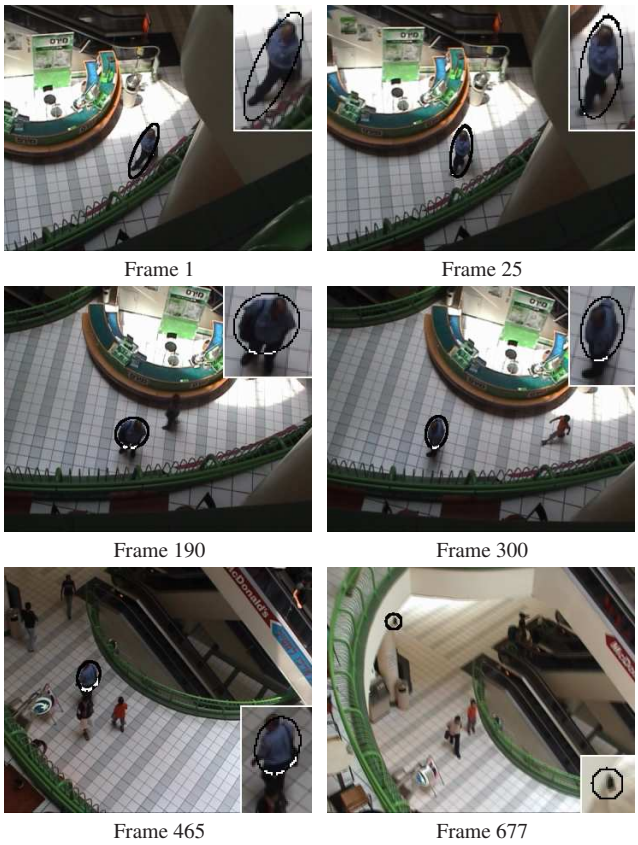

Figure 9: Results of the proposed tracker for Sequence III.

[17] N.S. Peng, J. Yang, and Z. Liu. Mean shift blob tracking with kernel histogram filtering and hypothesis testing. *Pattern Recognition Letters*, 26:605–614, 2005.

[18] K. Shearer, K.D. Wong, and S. Venkatesh. Combining multiple tracking algorithms for improved general performance. *Pattern Recognition*, 34(6):1257–1269, 2001.

[19] Y. Wu and T.S. Huang. Robust visual tracking by integrating multiple cues based on co-inference learning. *IJCV*, 58(1):55–71, 2004.

[20] C. Yang, R. Duraiswami, and L. Davis. Fast multiple object tracking via a hierarchical particle filter. *ICCV*, 1:212–219, 2005.

[21] C. Yang, R. Duraiswami, and L.S. Davis. Efficient mean-shift tracking via a new similarilty measure. *CVPR*, 1:176–183, 2005.

[22] H. Zhang, W. Huang, Z. Huang, and L. Li. Affine object tracking with kernel-based spatial-color representation. *CVPR*, 1:293–300, 2005.