Machine Vision and Applications manuscript No.
(will be inserted by the editor)

Ciarán Ó Conaire · Noel E. O'Connor · Alan Smeaton

Thermo-Visual Feature Fusion for Object Tracking Using Multiple Spatiogram Trackers

Received: date / Revised version: date

Abstract In this paper, we propose a framework that can efficiently combine features for robust tracking based on fusing the outputs of multiple spatiogram trackers. This is achieved without the exponential increase in storage and processing that other multimodal tracking approaches suffer from. The framework allows the features to be split arbitrarily between the trackers, as well as providing the flexibility to add, remove or dynamically weight features. We derive a mean-shift type algorithm for the framework that allows efficient object tracking with very low computational overhead. We especially target the fusion of thermal infrared and visible spectrum features as the most useful features for automated surveillance applications. Results are shown on multimodal video sequences clearly illustrating the benefits of combining multiple features using our framework.

Keywords Thermal infrared · visible spectrum · fusion · tracking · spatiogram

1 Introduction

In order for automated surveillance systems to become useful in practice, persistence (24-7 operability) is a key requirement. The use of thermal infrared cameras allows this possibility, since they detect emitted radiation rather than reflected rays and therefore do not require a lighting source (such as the sun, or artificial lighting). The main drawbacks of standard CCTV systems arise from their reliance on reflected light: inability to operate at night, adverse effects due to shadows and lighting changes. Thermal infrared camera systems are resilient to these effects and are therefore very useful in a security context. However, they have their own problems: thermal video has a lower signal-to-noise ratio than visible spectrum video, the *halo effect* of ferroelectric sensors causes an artificial 'halo' to appear around very hot or cold objects [11], thermal video cannot detect objects that are at the same temperature as the background and it contains very little texture information that could be used to distinguish objects. Combining the two modalities would seem to be a rational approach to obtaining optimum system performance and robustness. In relation to object tracking in particular, it is generally recognised that tracking robustness cannot be obtained with one modality (or feature) alone, and that increased robustness can be obtained by combining multiple modalities (also known as multi-cue tracking) in such a way that they can, together, compensate for their individual weaknesses [26,16].

The contributions in this paper are as follows. Drawing from our previous work on feature fusion [20], we introduce a general feature combination framework for tracking, similar to the framework of [16], that extends the recently proposed *spatiogram* tracker [5] to efficiently integrate multiple features. In addition, we provide a derivation of a mean-shift procedure for our framework to efficiently track objects in video sequences. We justify the validity of the framework by demonstrating robust tracking on multimodal (thermal infrared and visible spectrum) surveillance sequences. The framework we propose allows the integration of multiple features, or sources of information, without the exponential increase in storage and processing that is associated with histograms or spatiograms. Although we do not explicitly show experimental results on the topic of adaptively choosing tracking features, our framework is flexible and allows features to be added, removed or weighted dynamically during tracking and we provide formulations of how feature weighting can be integrated into the framework.

This material is based on works supported by Science Foundation Ireland under Grant No. 03/IN.3/I361 and sponsored by a scholarship from the Irish Research Council for Science, Engineering and Technology (IRCSET): Funded by the National Development Plan.

Adaptive Information Cluster, Dublin City University, Dublin 9, Ireland
Tel.: +353-1700-5872, Fax : +353-1700-5508
E-mail: oconaire@eeng.dcu.ie

In the next section, we summarise related work in the areas of tracking and the fusion of multiple sources of tracking evidence. In section 3, we review histogram- and spatiogram-based tracking, before detailing our proposed tracking framework that uses a bank of parallel spatiogram trackers. We derive a mean-shift procedure for our framework, allowing fast and efficient tracking. In section 4, we compare the results of combining multiple features using our tracking framework to single feature tracking approaches. We also demonstrate the efficiency of our mean-shift procedure for object tracking and give some ideas on future work. The paper is summarised in section 5.

2 Related work

It is generally accepted that "no single visual cue will be robust and general enough to deal successfully with the wide variety of conditions occurring in real-world scenarios" [26]. Therefore, to create robust systems, multiple features (or cues) need to be used in such a way that they can, together, compensate for their individual weaknesses. The use of feature combination for tracking is an active research area and many approaches have been proposed to combine the information from multiple sources in order to provide more accurate and robust detection and tracking. Probabilistic methods are commonly used to fuse information sources. In [18], Bayesian probability theory is used to fuse the tracking information available from a suite of cues to track a person in 3D space. A Bayesian tracking framework using particle filters is described in [21] for fusing colour cues with stereo or motion information. A Bayesian multi-object tracker is described in [25] that fuses binary information from foreground detection with colour tracking cues. Linear combinations of sources have also been widely used to fuse information from multiple sources. In [15], information from image segmentation is fused with chamfer matching scores to robustly detect people in cluttered images. Lim and Kriegman [17] use a linear combination of shape and appearance to track people in an indoor environment. Both [15] and [17] use fixed weighting for the data sources. In [24], the weightings for each tracking cue (colour and edge histograms) are adaptively updated using the Bhattacharyya coefficients. Fumera and Roli [12] consider linear combinations of classifiers and conclude that weighted average combinations usually only provide a marginal improvement over simple averaging, even with optimal weights. Recently, in [3], *Ensemble Tracking* was introduced as a general tracking framework to combine information from multiple sources, where an ensemble of linear classifiers are trained continuously to distinguish object and background pixels. Kruppa and Schiele [14] fuse information from multiple object detection modules by determining a configuration that maximises the mutual information between the models. In [28], Torresan et al. describe a surveillance system that fuses standard visible spectrum and thermal infrared video using background modelling and rule-based blob linking to detect and track pedestrians. In [11], the benefits of fusing colour and thermal infrared information are demonstrated using a contour based approach where binary contour fragments from each modality are fused for person detection. In [26], the *democratic integration* scheme is evaluated, where the weights of different visual cues are adapted dynamically and the probability densities of individual cues are linearly combined to give the fused PDF. In [16], a general framework is proposed to combine multiple tracking algorithms given the assumptions that each tracker outputs a probability density function (PDF) and that the features used by all trackers are conditionally independent. In their framework, the PDFs of each tracker are multiplied to determine the PDF of the combined tracker (assuming a uniform prior). Their work is most similar to ours and our previously reported work on evaluating feature combination strategies for tracking [20] supports such an approach.

Regardless of the approach used for feature combination, object properties must be modelled in individual modalities and many approaches have been proposed to do this for accurate object localisation in subsequent video frames. In fixed camera scenarios, a background model can be estimated [27] and subtracted so that moving objects are modelled as foreground blobs [28]. Image templates have also been used as object models [19]. Active contours, or *snakes*, have frequently been used to track the boundary of an object [1]. Object appearance models have also been used for tracking [20,30].

Feature histograms have been shown to be robust and efficient for object modelling for use in surveillance tracking, as they capture stable object properties that are resilient to changes in object pose due to local object motion (e.g. walking) and small changes in perspective. As such, we have adopted a histogram-like approach to modelling object properties. In their seminal work, Comaniciu et al. [9] derive a mean-shift formulation for histogram tracking allowing real-time tracking that requires only a few iterations per frame to converge on the correct target. Adaptation to scale changes is performed by examining windows that are 10% larger and smaller than the current size. Collins [6] improves upon this scale selection heuristic, deriving a two-stage mean-shift procedure that interleaves spatial and scale mode-seeking using differential scale-space filters. In [31], scale adaptation is formulated as an EM-based approach. A method to perform very fast exhaustive histogram matching to locate the tracked object position is proposed in [22], where integral histograms are computed using dynamic programming. However, this method requires a large amount of memory. Birchfield and Rangarajan [5] generalise the histogram formulation by introducing spatial histograms, or *spatiograms*, that are histograms with higher-order moments. Like histograms, spatiograms allow comparisons between image regions without explicitly computing any explicit geometric transformation between them. However, unlike histograms, spatiograms retain some information about the geometry of object feature distributions, allowing them to remain more tightly locked onto their targets and less likely to be distracted.

In the context of combining object features, the main drawback of using histograms or spatiograms is that their memory requirements (and hence their computational load) increase exponentially as more features are added and they do not scale well to higher dimensions [3]. For example, an RGB colour histogram with 32 bins per channel requires a total of $32^3 = 32768$ bins. If an extra channel, such as thermal infrared, is added, this increases to $32^4 = 1048576$, which increases the memory requirements and decreases the tracking speed due to increased computation. There is also the issue of the *curse of dimensionality* [4] which states that it is more difficult to accurately estimate feature distributions for higher dimensional spaces, since exponentially more samples are required. It has also been shown that the Bhattacharyya coefficient, often used in tracking to measure similarity between histogram distributions, is not very discriminative in higher dimensions [29]. To overcome these difficulties, tracking can be achieved by splitting the feature-set over several histogram trackers and combining their outputs. For example, instead of using a K dimensional histogram, K one-dimensional histograms could be used and their outputs combined, which is equivalent to using K separate trackers. This substantially reduces the memory and computational requirements, and also allows the use of parallel processing to further speed up tracking. Unfortunately in the case of histograms, it is not theoretically justifiable to separate features in this way, as the assumption of independence does not hold. We overcome this by using 2^{nd}-order spatiograms [5] instead, since the inclusion of spatial information makes the independence assumption more valid.

For these reasons, we have adopted spatiograms as the tracking mechanism within our framework, thereby leveraging the benefits of a histogram-like approach whilst avoiding the inherent drawbacks.

3 Proposed Framework

Before describing our proposed framework, we briefly review the use of histograms and spatiograms, in the context of object tracking. We do this in order to keep this paper self-contained, since key variables and terminology are used in our derivation of a mean-shift procedure for our tracking framework in subsection 3.4, which is a key contribution of this paper.

3.1 Histograms

Simple feature histograms have frequently been used to model objects for tracking [10, 22]. A histogram is a normalised count of the number of times a feature falls into a specified range of values. The normalised count of bin b for the target object can be computed as follows:

$$n_b^{'} = C \sum_{i=1}^{N} k(||x_i||^2)\delta_{ib} \tag{1}$$

where N is the number of pixels, $\delta_{ib} = 1$ if the i^{th} pixel falls in the b^{th} bin and $\delta_{ib} = 0$ otherwise, C is a normalising constant that ensures the n_b values sum to one, $x_i = [\mathtt{x},\ \mathtt{y}]^T$ is the spatial position of the i^{th} pixel, k is a smoothing kernel, adding weight to pixels closer to the centre, hence reducing the effect of background pixels. Epanechnikov or Gaussian kernels are commonly used [10].

To evaluate a matching candidate of size h, containing N_h pixels, at location y, its histogram is computed as follows, with C_h performing a similar normalisation function to C:

$$n_b(y) = C_h \sum_{i=1}^{N_h} k(||(x_i - y)/h||^2)\delta_{ib} \tag{2}$$

A target and candidate histogram with B bins each can be compared using the Bhattacharyya coefficient [10]:

$$\rho(y) = \sum_{b=1}^{B} \sqrt{n_b(y)n_b^{'}} \tag{3}$$

3.2 Spatiograms

Spatiograms [5] are a generalisation of the common histogram and contain the exact same information as histograms but also include additional spatial information for each bin. Specifically, 2^{nd}-order spatiograms include the spatial mean and covariance of each bin. These are computed as follows:

$$\mu_b(y) = \frac{1}{\sum_{j=1}^{N_h} \delta_{jb}} \sum_{i=1}^{N_h} (x_i - y)\delta_{ib} \tag{4}$$

$$\Sigma_b(y) = \frac{1}{\sum_{j=1}^{N_h} \delta_{jb}} \sum_{i=1}^{N_h} (x_i - \mu_b(y))^T (x_i - \mu_b(y))\delta_{ib} \tag{5}$$

where, as before, N_h is the number of pixels in the region, y is the position of the region centre and x_i is the spatial position of the i^{th} pixel. The spatial distribution of each bin b is modelled as a Gaussian with the mean

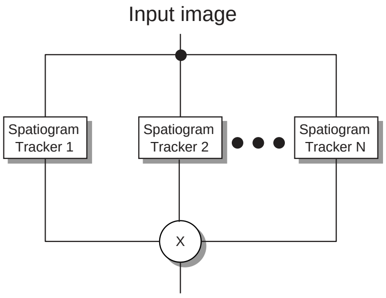

Fig. 1 Bank of Spatiograms

and covariance given above. In order to ensure that each Σ_b is invertible, they are assumed to be diagonal, and a minimum variance value is set to one pixel. A more formal description of the spatiogram family of descriptors can be found in [5].

To compare two spatiograms, the following Bhattacharyya-like similarity measure is the one used in the original work [5] and also used in the experiments described in this paper:

$$\rho(y) = \sum_{b=1}^{B} \psi_b(y)\sqrt{n_b(y)n_b^{'}} \tag{6}$$

where $\psi_b(y)$ is the spatial similarity measure, given by:

$$\psi_b(y) = \eta exp\left\{-\frac{1}{2}(\mu_b(y)-\mu_b^{'})^T\hat{\Sigma}_b^{-1}(y)(\mu_b(y)-\mu_b^{'})\right\} \tag{7}$$

where $\hat{\Sigma}_b^{-1}(y) = (\Sigma_b^{-1}(y) + (\Sigma_b^{'})^{-1})$, so that the distance between the spatial means is normalised to the average of the two Mahalanobis distances and η is the Gaussian normalisation constant. This measure gives high similarity scores to spatiograms whose histogram bins counts are similar and whose spatial means are aligned.

3.3 Spatiogram Bank Framework

In [5], the spatiogram is proposed as a more accurate model for object tracking than histograms. We propose to make spatiogram tracking more efficient and suitable for multimodal data fusion by splitting the features over multiple separate spatiograms. The tracking framework we propose is illustrated in figure 1, where the pixel-based features used to track the target are split over N spatiogram model trackers. All trackers evaluate a series of potential object position hypotheses and return a similarity score for each one. The combined score for each hypothesis is computed by multiplying the similarity scores from each tracker. Formally, the combined score is written:

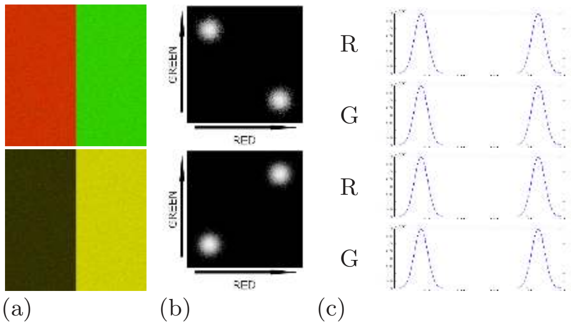

Fig. 2 (a) Synthetic images, (b) their associated joint histograms shown in log scale that clearly allow the images to be distinguished, and (c) the partial histograms of their red and green bands (both images have identical partial histograms therefore cannot be distinguished if the features are assumed to be independent).

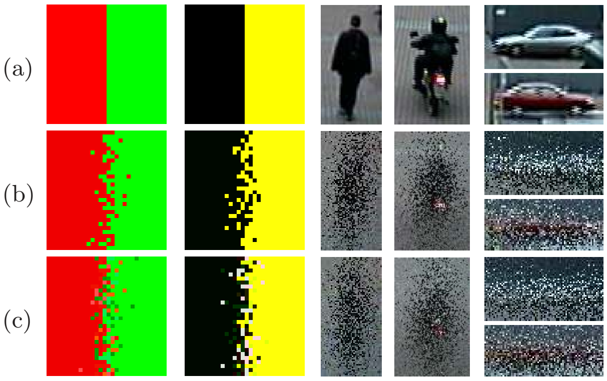

Fig. 3 Example object model images generated from spatiogram models: (a) Original images, object model images generated by: (b) Full YUV Spatiogram, (c) Partial YUV Spatiograms.

$$\rho(y) = \prod_{k=1}^{K} \rho^{(k)}(y) \tag{8}$$

where $\rho^{(k)}(y)$ is the similarity score, returned by the k^{th} spatiogram tracker, between the model and the candidate at position y. We adopt this tracking framework for object feature fusion for a number of reasons.

Firstly, the increase in memory and processing requirements is linear with respect to the number of features used (unlike the exponential increase associated with typical histograms and spatiograms) and it does not suffer from the *curse of dimensionality* in accurately estimating feature distributions. The framework allows features to be arbitrarily divided between the K trackers. In our experiments, we use one tracker per feature, but one could combine an RGB spatiogram tracker with an infrared brightness tracker, for example. Unlike template matching, spatiograms trackers do not impose rigid spatial constraints. Instead, the small amount of stored spatial information allows more general object deformations. Also, the tracking framework we use can incorporate a mean-shift approach to object localisation, allowing rapid object tracking.

Secondly, this framework draws on our previous work in evaluating various fusion schemes for object tracking [20], where we found that multiplying similarity scores outperformed simple addition, weighted sums and non-linear score fusion schemes, such as *min* and *max*. If we consider the similarity metric as a probability, multiplying scores is equivalent to assuming the features used by the trackers are independent. The metric we used to compare spatiograms is very similar to the Bhattacharyya coefficient, which itself is closely related to the probability of *Bayes error* [2]. In [16], Leichter et al. propose a general framework for tracker fusion by computing a combined probability density function (PDF) by multiplying the PDFs of all trackers (assuming a uniform prior) and our framework can be interpreted as conforming to this general framework. If all spatiogram trackers in our framework perform an exhaustive search in a local search window by computing similarity scores for each location, these scores can be multiplied by a constant without affecting the final combined tracking result. If we choose the constant so that the scores are normalised to sum to one, then we essentially have a PDF which is then multiplied to produce the final combined PDF, hence the similarity to Leichter's framework.

Thirdly, by separating the features as we do, instead of integrating them into one tracker, we provide a flexible architecture for feature addition, removal or weighting, allowing the combined tracker to adapt under different circumstances. This has been shown to benefit tracking in changing environments [3][7]. In terms of the limited spatial information we store, modelling of each feature bin as a Gaussian as we do may seem restrictive, but in fact captures some more general spatial distribution properties, as discussed below.

3.3.1 Independence of spatiograms

The images in figure 2(a) are significantly different and can clearly be distinguished by their joint red-green histograms, shown in (b). The drawback of joint distributions, as mentioned earlier, is that they require exponentially more memory as more features are added, as well as suffering from the *curse of dimensionality*. Using partial histograms as an approximation for the joint-histogram is not a valid solution, since both images have identical partial distributions in both the red and green bands, therefore cannot be distinguished if the features are separated (see figure 2(c)). Instead of using partial histograms for tracking, we argue that the use of partial spatiograms is *more valid* for this propose.

In appendix A.2, we derive the model generated when partial spatiograms are used to approximate joint spatiograms. This model can be used to verify our hypothesis. Although it is possible to artificially create pixel regions that are not well approximated by partial spatiograms, we found that with real data our tracker performed well in experiments. It even out-performed tracking using the joint spatiogram, possibly due to the difficulty in estimating high-dimensional distributions. Additionally, we evaluated the validity of spatiogram independence using the derived model and 4517 rectangular pixel regions from images in datasets such as OTCBVS, PETS2001 and VACE, as well as our own data. We found that when we compared $\bar{n}$, the approximation of the joint histogram, generated by partial spatiograms to that produced by partial histograms, partial spatiograms were more accurate in 95% of tests. The average increase in histogram comparison score, with 16 bins per channel, was 0.1596.

Figure 3 shows examples of images generated by sampling the PDFs of the spatiograms extracted from real and synthetic data, using 16 bins per colour component. The validity of the independence assumption for spatiograms is clear from comparing rows (b) and (c), since there is only a minor difference between using the joint-feature spatiogram and using separate spatiograms. As can be seen from the image generated from the walking person spatiograms (middle column), the distribution of the background (bright) pixels is not Gaussian. In fact, the distribution of a particular feature (or joint feature) can be expressed as a Gaussian distribution divided by a mixture of Gaussians, which therefore captures more general spatial distribution properties (See Appendix A.1).

3.4 Mean-shift using Spatiogram Banks

The mean-shift derivation presented here is motivated by and follows the general procedure presented in [5]. Unlike in [5], however, where the mean-shift procedure was derived for a single (possibly high-dimensional) spatiogram, here we derive the procedure for a bank of (low-dimensional) spatiograms and this is a key novelty with respect to our work. As such, we present our derivation in detail in this section.

Mean-shift [8] is an iterative kernel-based procedure to locate the local mode in a distribution. It has been successfully used in many tracking applications [9, 31, 7] to efficiently locate objects in subsequent frames under the assumption that the object overlaps itself in consecutive frames. For fast-moving objects and low frame-rate video, where this assumption may not be valid, multiple kernels can be used [23].

To initiate the iterative mean-shift scheme using our tracking framework, each tracker is first given an object position hypothesis, which is generally equal to its position in the previous frame or a prediction of its current location, based on a velocity estimate for example. Using the similarity measures returned by each tracker, along with the pixel features and spatiogram models, mean-shift performs gradient ascent on the similarity surface and computes a new object position hypothesis. This procedure is iterated until convergence. We begin by describing the combined similarity measure $\rho(y)$, as the product of all K individual tracker similarities, first examining a simple two tracker system where $\rho(y) = \rho^{(1)}(y)\rho^{(2)}(y)$ and generalising later to handle K trackers. So assuming $K = 2$, we perform a Taylor series expansion around $\rho(y)$ at y_0, and obtain

$$\begin{aligned}\rho(y) \approx \rho(y_0) &+[n^{(1)}(y) - n^{(1)}(y_0)]^T \tfrac{\partial\rho}{\partial n^{(1)}}(y_0)\\ &+[n^{(2)}(y) - n^{(2)}(y_0)]^T \tfrac{\partial\rho}{\partial n^{(2)}}(y_0)\\ &+[\mu^{(1)}(y) - \mu^{(1)}(y_0)]^T \tfrac{\partial\rho}{\partial \mu^{(1)}}(y_0)\\ &+[\mu^{(2)}(y) - \mu^{(2)}(y_0)]^T \tfrac{\partial\rho}{\partial \mu^{(2)}}(y_0)\end{aligned}$$

where the superscript notation refers to the tracker number (for example, $\mu^{(2)}$ refers to the bin spatial means of the features used by tracker 2). Using the fact that the tracker scores are independent with respect to the parameters of other trackers, we obtain

$$\begin{aligned}\frac{\partial\rho}{\partial n^{(1)}} &= \rho^{(2)}(y)\frac{\partial\rho^{(1)}}{\partial n^{(1)}} \quad , \quad \frac{\partial\rho}{\partial \mu^{(1)}} = \rho^{(2)}(y)\frac{\partial\rho^{(1)}}{\partial \mu^{(1)}}\\ \frac{\partial\rho}{\partial n^{(2)}} &= \rho^{(1)}(y)\frac{\partial\rho^{(2)}}{\partial n^{(2)}} \quad , \quad \frac{\partial\rho}{\partial \mu^{(2)}} = \rho^{(1)}(y)\frac{\partial\rho^{(2)}}{\partial \mu^{(2)}}\end{aligned}$$

Inserting into the previous equation for $\rho(y)$

$$\begin{gathered}\rho(y) \approx \qquad \rho(y_0) +\\ \rho^{(2)}(y_0)\{([n^{(1)}(y) - n^{(1)}(y_0)]^T \tfrac{\partial\rho^{(1)}}{\partial n^{(1)}}(y_0) +\\ [\mu^{(1)}(y) - \mu^{(1)}(y_0)]^T \tfrac{\partial\rho^{(1)}}{\partial \mu^{(1)}}(y_0)\} +\\ \rho^{(1)}(y_0)\{([n^{(2)}(y) - n^{(2)}(y_0)]^T \tfrac{\partial\rho^{(2)}}{\partial n^{(2)}}(y_0) +\\ [\mu^{(2)}(y) - \mu^{(2)}(y_0)]^T \tfrac{\partial\rho^{(2)}}{\partial \mu^{(2)}}(y_0)\}\end{gathered}$$

Simplifying, and generalising to K trackers, we obtain

$$\begin{gathered}\rho(y) \approx \qquad \rho(y_0) +\\ \textstyle\sum_{k=1}^{K} \frac{\rho(y_0)}{\rho^{(k)}(y_0)}\{([n^{(k)}(y) - n^{(k)}(y_0)]^T \frac{\partial\rho^{(k)}}{\partial n^{(k)}}(y_0) +\\ [\mu^{(k)}(y) - \mu^{(k)}(y_0)]^T \tfrac{\partial\rho^{(k)}}{\partial \mu^{(k)}}(y_0)\}\end{gathered}$$

We can simplify this expression by defining two new variables:

$$\begin{aligned}\Gamma_n^{(k)} &= [n^{(k)}(y) - n^{(k)}(y_0)]^T \frac{\partial\rho^{(k)}}{\partial n^{(k)}}(y_0)\\ \Gamma_\mu^{(k)} &= [\mu^{(k)}(y) - \mu^{(k)}(y_0)]^T \frac{\partial\rho^{(k)}}{\partial \mu^{(k)}}(y_0)\end{aligned}$$

Inserting them into the previous expression:

$$\rho(y) \approx \rho(y_0) + \sum_{k=1}^{K} \frac{\rho(y_0)}{\rho^{(k)}(y_0)} \left\{\Gamma_n^{(k)} + \Gamma_\mu^{(k)}\right\} \tag{9}$$

Taking the derivative of (9) with respect to y and setting this equal to zero, yields:

$$\sum_{k=1}^{K} \frac{\rho(y_0)}{\rho^{(k)}(y_0)} \frac{\partial\Gamma_n^{(k)}}{\partial y} = -\sum_{k=1}^{K} \frac{\rho(y_0)}{\rho^{(k)}(y_0)} \frac{\partial\Gamma_\mu^{(k)}}{\partial y}$$

where

$$\begin{aligned}\frac{\partial\Gamma_n^{(k)}}{\partial y} &= -\sum_{i=1}^{N} \alpha_i^{(k)} g\left(\|\frac{y_0 - x_i}{h}\|^2\right)(y - x_i)\\ \frac{\partial\Gamma_\mu^{(k)}}{\partial y} &= -\sum_{b=1}^{B} v_b^{(k)}\end{aligned}$$

where $g(x) = -dk(x)/dx$ is the negative derivative of the kernel profile, which is constant if the Epanechnikov kernel (referred to in section 3.1) is used. $\alpha_i^{(k)}$ and $v_b^{(k)}$ are given by:

$$\alpha_i^{(k)} = \frac{C_h}{h^2} \sum_{b=1}^{B} \psi_b^{(k)}(y_0) \sqrt{\frac{n_b^{'(k)}}{n_b^{(k)}(y_0)}}\delta_{ib} \tag{10}$$

$$\nu_b^{(k)} = \psi_b^{(k)}(y_0)\sqrt{n_b^{'(k)} n_b^{(k)}(y_0)}(\hat{\Sigma}_b^{(k)}(y_0))^{-1}(\mu_b^{'(k)} - \mu_b^{(k)}(y_0)) \tag{11}$$

The values $\alpha_i^{(k)}$ can be interpreted as pixel weights that vote strongly when the bin-count of the bin they fall into is lower than the target bin-count, encouraging movement towards areas similar to the target histogram. The $\nu_b^{(k)}$ values are vectors that encourage the tracker to move so that bin spatial centres align with the target's spatial means. By moving all the terms that do not involve y to the right-hand side of the equation, the mean-shifted position, y, can be written as

$$y = \frac{\sum_{i=1}^{N} A_i g(\left\|\frac{y_0 - x_i}{h}\right\|^2) x_i - \sum_{b=1}^{B} V_b}{\sum_{i=1}^{N} A_i g(\left\|\frac{y_0 - x_i}{h}\right\|^2)} \tag{12}$$

where A_i and V_b are defined as:

$$A_i = \sum_{k=1}^{K} \alpha_i^{(k)} \rho(y_0)/\rho^{(k)}(y_0) \tag{13}$$

$$V_b = \sum_{k=1}^{K} \nu_b^{(k)} \rho(y_0)/\rho^{(k)}(y_0) \qquad (14)$$

The combined mean-shift algorithm for multiple spatiogram trackers thus derived is used as follows: Given a starting image position y_0 near where the object is located, equation (12) is used to compute the next mean-shifted position, which should move towards the true object position. The new position y replaces y_0 in the equation and the procedure is iterated until convergence i.e. until y and y_0 are within the same pixel. To use equation (12), we first compute the similarity scores for each tracker using (6), then compute the combined score using (8). Using (10), the values of $\alpha_i^{(k)}$ are computed for each i^{th} pixel and k^{th} tracker. With (11), the 2-d vector values of $v_b^{(k)}$ are computed for each b^{th} bin and k^{th} tracker. Finally, A_i and V_b are computed and inserted into the mean-shift equation.

4 Results

We show three sets of experiments in this chapter. First we demonstrate how multimodal tracking significantly outperforms tracking using any one single feature. Secondly, we illustrate the efficiency of our derived mean-shift procedure for tracking. Thirdly, we show quantitative tracking results comparing our tracking framework to standard histogram- and template-based tracking methods.

In our first experiment, we compare the use of single features against our combined framework for tracking. Figure 4 shows the tracking results for two multimodal video sequences. The data used is aligned visible spectrum and thermal infrared video, with each pixel represented by its colour components and infrared brightness. In our experiments, five features were used: Y, U, V, thermal brightness and edge orientation, with 8 bins per feature. We used an exhaustive search in a 11 × 11 window around the previous object location, and varied the scale by +/-10%, choosing the scale that returned the largest similarity score, as in [9] and [5]. The spatiogram models for the object are extracted in the first frame and remain fixed for the duration of the experiment. Rows (c) and (h) show the luminance-based tracker and (d) and (i) are the infrared-based tracker. Results for the other three features are omitted for clarity of presentation, but were always less effective than either luminance or infrared. Our combined tracker, using all five features, is shown in (e) and (j). In the first difficult tracking sequence, taken from the OTCBVS benchmark dataset [11], we attempt to track a woman in dark clothing through occlusion and distraction by crowds. In frame 812, the luminance tracker fails as the woman walks into an area under shadow. In frame 1009, the infrared tracker fails and locks onto a passing person. There is little to distinguish people in infrared since, due to the camera pixel saturation, hot bodies appear bright white. The infrared tracker settles on a street-light until another person passes who it begins to track in frame 1230. Our combined tracker tracks the person throughout the entire sequence, despite severe occlusion and background distraction. The second sequence in figure 4 was captured with our own multimodal camera rig [20] and shows similar results in tracking a cyclist. Both the luminance and infrared tracker fail when the cyclist turns the corner. The luminance tracker locks onto another bicycle in the bike-rack, while the infrared tracker locks onto another person who is standing in the bike-rack. Our combined tracker, however, successfully tracks the cyclist for the entire duration of the sequence. Both sequences in figure 4 show that combining features outperforms any single feature in tracking.

Our second experiment, shown in figure 5 illustrates tracking results using the mean-shift procedure we derived for the bank of spatiograms. In both sequences, we used 32 bins per feature and initialised the mean-shift kernel at the location where the object was found in the previous frame, then performed mean-shift tracking using the derived procedure at three different scales (the current object size and +/-10%). The scale that gave the largest similarity score was selected as the correct scale. YUV colour features were used in both experiments and infrared brightness was also used in the second sequence. In the first sequence, which is taken from the PETS2001 video dataset, the tracking of a blue car with a moving background is shown during a rapid change in scale. In the second sequence, which is a multimodal sequence (infrared band is not shown), we show the tracking of a book over a complex background. As the book is at room temperature, the thermal features only add noise to the tracker but it still successfully keeps a lock on the target. The sequences required, on average, 7.68 and 10.95 iterations per frame, respectively, to converge. This is about 40 times faster than an exhaustive search in a 11 × 11 × 3 local scale window. Using standard histogram or spatiogram mean-shift tracking would require over 340 times as many bins (32^3 instead of 3×32). In our interpreted MATLAB implementation, the mean tracking speed (over 36 different tracking tests) is just over 9 frames/second, which includes reading bitmap images from hard-drive. We envisage that an optimised version would run comfortably in realtime.

In our third set of experiments, we show in table 1 some quantitative tracking results comparing our tracking framework to histogram- and template-based tracking methods. We used 6 sequences taken from the public OTCBVS database, PETS'01, PETS'03 and our own multimodal collection (marked DCU in the table). All sequences (except the PETS sequences) include an infrared channel, along with the RGB channels. Groundtruth was generated by manual annotation of bounding

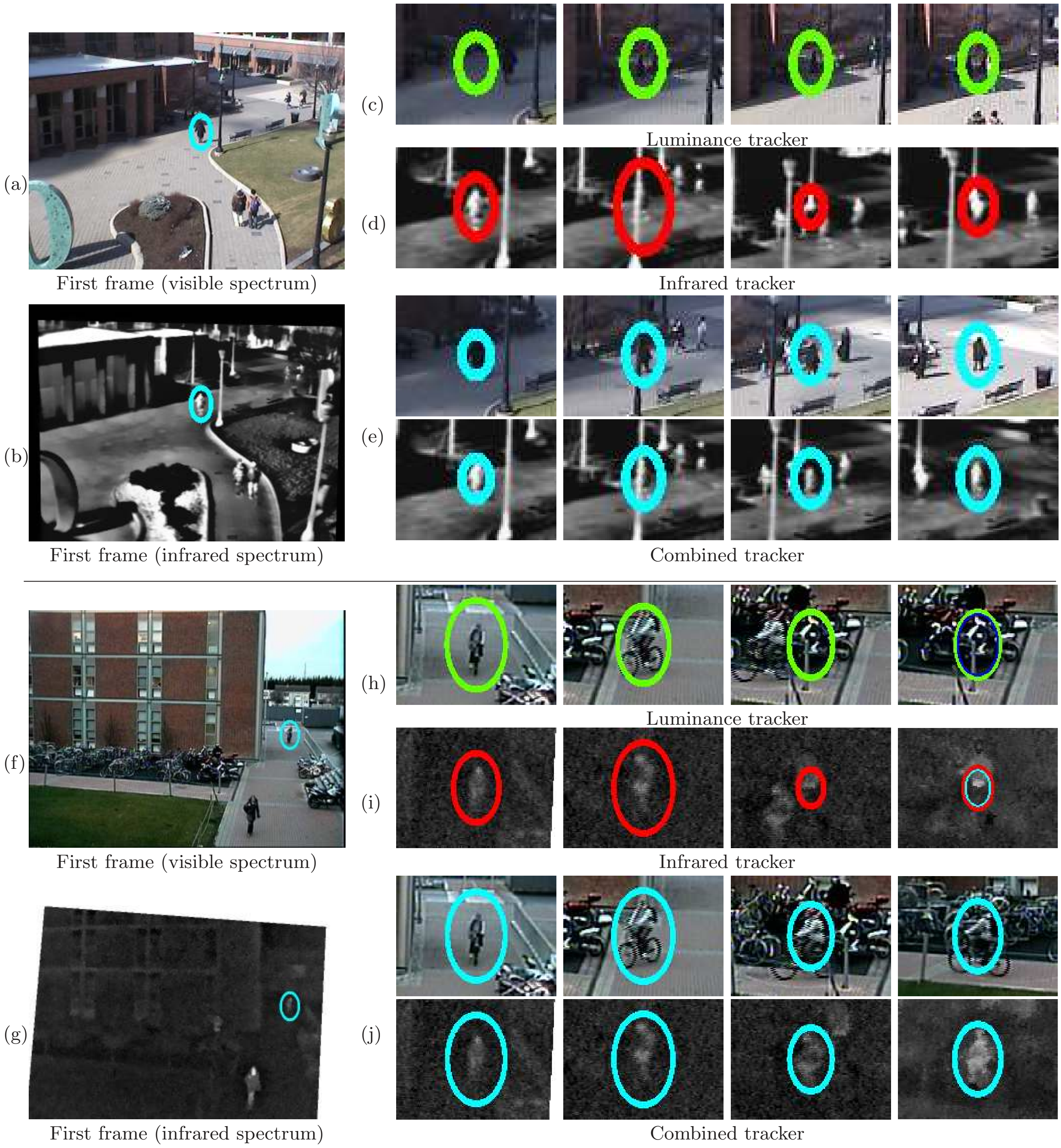

Fig. 4 Tracking results using single features versus combined tracking: pedestrian and cyclist tracking. The left column shows the initial frame position of all trackers for two sequences (visible spectrum and thermal images shown in each case). The smaller images to the right show zoomed versions of the object tracked in subsequent frames of each sequence: (c),(h): Luminance-based spatiogram tracking. (d),(i): Infrared brightness spatiogram tracking. (e),(j): Combined tracking.

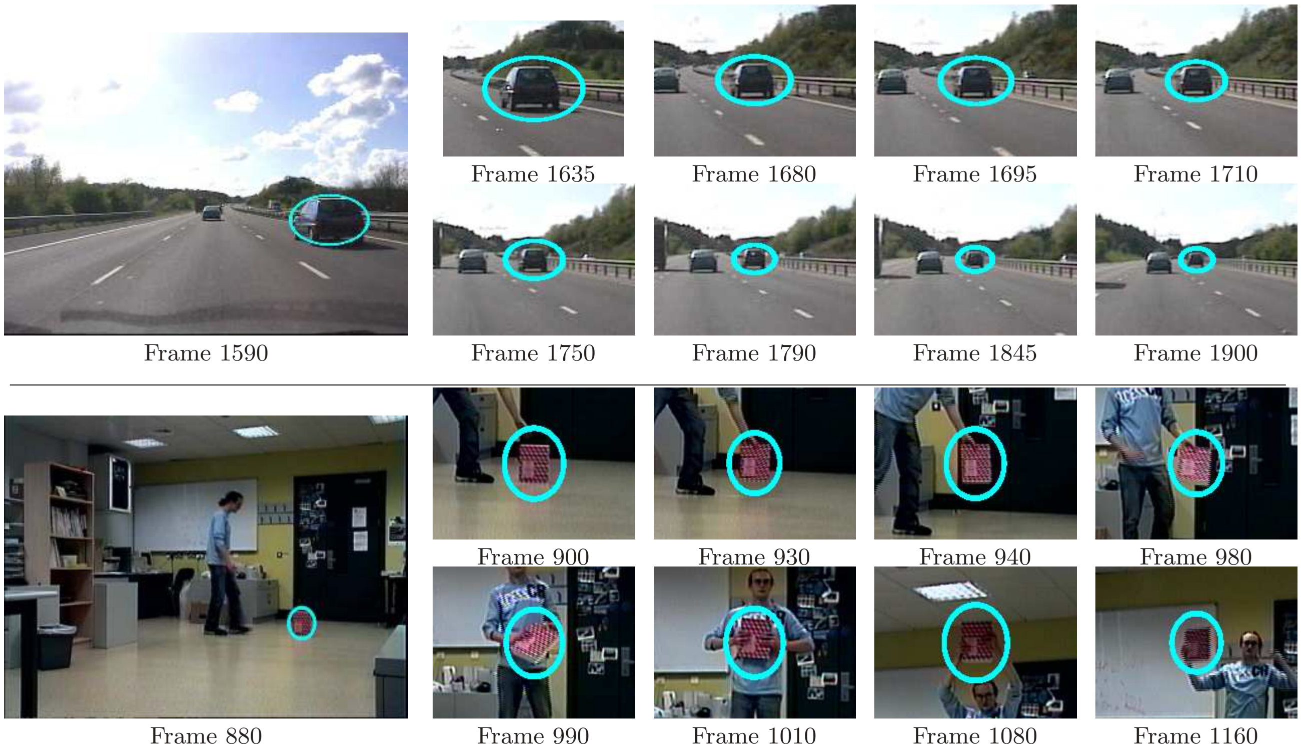
Frame 1590
Frame 1635
Frame 1680
Frame 1695
Frame 1710
Frame 1750
Frame 1790
Frame 1845
Frame 1900
Frame 880
Frame 900
Frame 930
Frame 940
Frame 980
Frame 990
Frame 1010
Frame 1080
Frame 1160

Fig. 5 Illustrative tracking results using the combined mean-shift procedure for our combined tracker using Y, U and V features (first sequence) and YUV and infrared features (second sequence).

Source	Type	Temp	Hist	Hist Bank	Spat	Spat Bank
DCU	person	100.0	29.1	29.0	29.1	**100.0**
PETS'01	vehicle	100.0	100.0	100.0	100.0	**100.0**
DCU	person	5.3	90.9	100.0	72.6	**100.0**
PETS'03	person	67.5	69.5	100.0	100.0	**100.0**
DCU	person	42.4	100.0	100.0	100.0	**100.0**
OTCBVS	person	100.0	97.5	100.0	51.2	**100.0**

Table 1 Table above indicates the percentage of frames in which the object was successfully tracked for a variety of sequences. These results can be viewed at http://www.eeng.dcu.ie/˜oconaire/mva07

boxes on the objects to be tracked. Tracking was judged to have failed if the tracker's bounding box no longer included any part of the object. The figures indicated are the percentage of frames of successful tracking before failure. Histogram tracking ('Hist' column) was based on [9] using exhaustive search. The bank of histograms tracker ('Hist bank' column) used the same approach but multiplied the matching scores of each channel. For template tracking ('Temp' column) we used the standard sum-of-squared difference method [19]. The tracker in the 'Spat' column is a tracker using the full joint-spatiogram. Object models were fixed at the start of the sequence and were not updated. We chose histogram and template tracking as they represent the two extremes of modelling feature spatial distribution. Histograms contain no spatial information and, at the other extreme, templates encode rigid spatial information. Spatiogram banks ('Spat bank' column), encoding a small amount of coarse spatial information, did best in our trials. The hist-bank tracker, although achieving a high success rate, often shrank in scale and tracked only part of the object. Video sequences are available online at
http://www.eeng.dcu.ie/˜oconaire/mva07

Discussion Mean-shift is much faster than exhaustive search as it only evaluates a small number of position hypoteses. However, when the target peak is narrow in the similarity surface, mean-shift can slide off the target. Exhaustive search in a local window is more likely to find the correct peak. We have shown that our method can perform well using either technique but found the exhaustive search to be more reliable and recommend its usage when speed is not a major factor.

In our experiments all features were treated equally, but dynamic feature weighting could be useful when the object and background have similar features. As a tracked object moves through various backgrounds and lighting conditions, different features become more or less useful for tracking [3,7] and therefore, tracking robustness could be increased by weighting them accordingly. As an indication towards future work, we provide here two different formulations that could be used to compute the combined score for a hypothesis, given appropriate weights for each tracker. Determining how these weights are computed is left for future work.

If we can determine, for the i^{th} tracker, the probability of it providing poor tracking information (i.e. the probability of *sensor error*), λ_i, then the combined score can be computed as follows:

$$\rho(y) = \Pi_{i=1}^{K}(\lambda_i + (1 - \lambda_i)\rho^{(i)}(y)) \tag{15}$$

This method is similar to the method used in [13] to compensate for poor tracking data and caters for tracking failure in scenarios where one tracker might return a score of zero, thereby indicating that the correct hypothesis is not used. If instead we have a set of weights, w_i, for each tracker, the following weighting method could be used:

$$\rho(y) = \Pi_{i=1}^{K}\rho^{(i)}(y)^{w_i} \tag{16}$$

If we take the *log* of this equation, we see that it is similar to the weighted sum used in *democratic integration* [26] to fuse multiple cues.

5 Conclusion

The main contributions of this paper are: (i) the introduction of a general pixel feature-based tracking framework that extends spatiogram tracking to efficiently combine multiple features, (ii) justifying the validity of this framework and demonstrating robust tracking on multimodal surveillance sequences and (iii) deriving a mean-shift procedure for this framework that allows multiple feature combination but with very low computational overhead.

Future work will investigate how spatiogram models should be updated during tracking, to account for changes in object appearance, while avoiding the problem of model drift [19]. We will also examine how features should automatically be weighted to minimise background distraction, and we have already made some suggestions in our discussion at the end of section 4 on how these weights could be integrated into our framework. Extending the mean-shift derivation to cater more robustly for changes in scale using 2^{nd} order moments [29] or scale-space methods [6] is also targeted as future work.

Acknowledgements

The authors would also like to express their gratitude to Mitsubishi Electric Research Labs (MERL) for their contribution to this work.

References

1. Abd-Almageed, W., Smith, C.E., Ramadan, S.: Kernel snakes: non-parametric active contour models. In: IEEE International Conference on Systems, Man and Cybernetics, vol. 1, pp. 240–244 (2003)
2. Andrews, H.: Introduction to Mathematical Techniques in Pattern Recognition. Wiley-Interscience (1972)
3. Avidan, S.: Ensemble tracking. In: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) (2005)
4. Bellman, R.E.: Adaptive Control Processes: A Guided Tour. Princeton University Press, Princeton, NJ (1961)
5. Birchfield, S.T., Rangarajan, S.: Spatiograms versus histograms for region-based tracking. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1158–1163 (2005)
6. Collins, R.T.: Mean-shift blob tracking through scale space. In: IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 234–240 (2003)
7. Collins, R.T., Liu, Y., Leordeanu, M.: Online selection of discriminative tracking features. IEEE Transactions on Pattern Analysis and Machine Intelligence **27**(10) (2005)
8. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence **24**(5), 603–619 (2002)
9. Comaniciu, D., Ramesh, V., Meer, P.: Real-time tracking of non-rigid objects using mean shift. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 142–151 (2000)
10. Comaniciu, V., Meer, P.: Kernel-based object tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence **25**(5), 564–577 (2003)
11. Davis, J., Sharma, V.: Fusion-based background-subtraction using contour saliency. In: Proc. IEEE International Workshop on Object Tracking and Classification Beyond the Visible Spectrum (2005)
12. Fumera, G., Roli, F.: A theoretical and experimental analysis of linear combiners for multiple classifier systems. IEEE Transactions on Pattern Analysis and Machine Intelligence **27**(6), 942–956 (2005)
13. Jennings, C.F.: Probabilistic evidence combination for robust real time finger recognition and tracking. Phd thesis, University of British Columbia (2002)
14. Kruppa, H., Schiele, B.: Hierarchical combination of object models using mutual information. In: BMVC (2001)
15. Leibe, B., Seemann, E., Schiele, B.: Pedestrian detection in crowded scenes. In: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) (2005)
16. Leichter, I., Lindenbaum, M., Rivlin, E.: A probabilistic framework for combining tracking algorithms. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp. 445–451 (2004)
17. Lim, J., Kriegman, D.: Tracking humans using prior and learned representations of shape and appearance. In: IEEE International Conference on Automatic Face and Gesture Recognition (FGR), pp. 869–874 (2004)
18. Loy, G., Fletcher, L., Apostoloff, N., Zelinsky, A.: An adaptive fusion architecture for target tracking. In: IEEE International Conference on Automatic Face and Gesture Recognition (FGR) (2002)
19. Matthews, I., Ishikawa, T., Baker, S.: The template update problem. IEEE Transactions on Pattern Analysis and Machine Intelligence **26**(6) (2004)
20. Ó Conaire, C., O'Connor, N.E., Cooke, E., Smeaton, A.F.: Comparison of fusion methods for thermo-visual surveillance tracking. In: International Conference on Information Fusion (2006)

21. Pérez, P., Vermaak, J., Blake, A.: Data fusion for visual tracking with particles. Proceedings of the IEEE **92**(3), 495–513 (2004)
22. Porikli, F.: Integral histogram: A fast way to extract higtograms in cartesian spaces. In: IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) (2005)
23. Porikli, F., Tuzel, O.: Multi-kernel object tracking. In: IEEE International Conference on Multimedia & Expo (2005)
24. She, K., Bebis, G., Gu, H., Miller, R.: Vehicle tracking using on-line fusion of color and shape features. In: IEEE International Conference on Intelligent Transportation Systems (2004)
25. Smith, K., Gatica-Perez, D., Odobez, J.M.: Using particles to track varying numbers of interacting people. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2005)
26. Spengler, M., Schiele, B.: Towards robust multi-cue integration for visual tracking. Machine Vision and Applications **14**(1), 50–58 (2003)
27. Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. In: Proceedings of CVPR99, pp. II:246–252 (1999)
28. Torresan, H., Turgeon, B., Ibarra, C., Hébert, P., Maldague, X.: Advanced surveillance system: Combining video and thermal imagery for pedestrian detection. In: Proc. of SPIE, Thermosense XXVI, SPIE, vol. 5405, pp. 506–515 (2004)
29. Yang, C., Duraiswami, R., Davis, L.: Efficient mean-shift tracking via a new similarity measure. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 176–183 (2005)
30. Zhou, S., Chellappa, R., Moghaddam, B.: Appearance tracking using adaptive models in a particle filter. In: Proc. of 6th Asian Conference on Computer Vision (ACCV) (2004)
31. Zivkovic, Z., Krose, B.: An em-like algorithm for color-histogram-based object tracking. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) (2004)

APPENDIX

A .

A.1 Spatial distribution probabilities

Histograms and spatiograms imply probability distributions of feature values. In the case of the histogram, there is no spatial dependency, so $p(\mathbf{x}, b) = p(b) = n_b$. For spatiograms, we have $p(\mathbf{x}, b) = p(b)p(\mathbf{x}|b) = n_b\phi_b(\mathbf{x})$. Given a particular location, x, we can compute the probability of occurrence for each feature bin:

$$
\begin{aligned}
p(b|\mathbf{x}) &= \frac{p(\mathbf{x}|b)p(b)}{p(\mathbf{x})} \\
&= \frac{n_b\phi_b(\mathbf{x})}{\sum_{i=1}^{B} n_i\phi_i(\mathbf{x})}
\end{aligned}
$$

Since $\phi_b(x)$ and each $\phi_i(x)$ are all Gaussians, this shows that the actual spatial distribution of feature value v is a Gaussian divided by a sum of Gaussians. This distribution can be multimodal and is therefore far more flexible than a simple Gaussian distribution.

A.2 Fusion of multiple spatiogram models

When we use partial histograms to approximate the full joint-histogram distribution, in the two-band case we get $n_{a,b} = p(a, b) = p_1(a)p_2(b) = n_a^{(1)}n_b^{(2)}$. We now derive the approximation that is used when partial spatiograms are used instead of the full joint-spatiogram. We examine the case where each pixel has two features (hue and saturation, for example) and it can easily be generalised to multiple features. For a particular location, x, the spatial distribution of pixels that belong to bin a of feature z_1 and b of feature z_2 is given by:

$$
\begin{aligned}
p(a, b|\mathbf{x}) &= p(a|\mathbf{x})p(b|\mathbf{x}) && (17)\\
&= \left(\frac{p_1(\mathbf{x}|a)p_1(a)}{p_1(\mathbf{x})}\right)\left(\frac{p_2(\mathbf{x}|b)p_2(b)}{p_2(\mathbf{x})}\right) && (18)\\
&= \frac{p_1(a)p_2(b)p_1(\mathbf{x}|a)p_2(\mathbf{x}|b)}{\left[\int_{B_1} p_1(\mathbf{x}|i)p_1(i)\right]\left[\int_{B_2} p_2(\mathbf{x}|j)p_2(j)\right]} && (19)\\
&= \frac{n_a^{(1)}n_b^{(2)}\phi_a^{(1)}(\mathbf{x})\phi_b^{(2)}(\mathbf{x})}{\left[\sum_{i=1}^{B_1}\phi_i^{(1)}(\mathbf{x})n_i^{(1)}\right]\left[\sum_{j=1}^{B_2}\phi_j^{(2)}(\mathbf{x})n_j^{(2)}\right]} && (20)
\end{aligned}
$$

where p_1 and p_2 refer to the probabilities obtained from the spatiogram model of the first and second feature (z_1 and z_2). This expression can be simplified by noting that the product of two normalised Gaussians, with means q and r, and covariances, Q and R, is a normalised Gaussian multiplied by a constant term:

$$N(x; q, Q)N(x; r, R) = zN(x; c, C) \qquad (21)$$

with

$$
\begin{aligned}
C &= (Q^{-1} + R^{-1})^{-1} && (22)\\
c &= C(Q^{-1}q + R^{-1}r) && (23)
\end{aligned}
$$

and the constant term, z, given by:

$$z = N(q; r, Q+R) = \frac{1}{(2\pi)^{m/2}|Q + R|^{1/2}}\ e^{\left(-\frac{1}{2}(q-r)^T(Q+R)^{-1}(q-r)\right)} \qquad (24)$$

where m is the number of dimensions (2 in this case). Now equation (20) can be rewritten as a Gaussian divided by a weighted sum of Gaussians, since all the ϕ terms are Gaussians. If we write:

$$\phi_a^{(1)}(\mathbf{x})\phi_b^{(2)}(\mathbf{x}) = z_{a,b}\phi_{a,b}(\mathbf{x}) \qquad (25)$$

And let

$$z'_{a,b} = \frac{z_{a,b}n_i^{(1)}n_j^{(2)}}{\sum_{i=1}^{B_1}\sum_{j=1}^{B_2} z_{i,j}n_i^{(1)}n_j^{(2)}} \qquad (26)$$

Then we can rewrite equation (20) as

$$
\begin{aligned}
p(a, b|\mathbf{x}) &= \frac{n_a^{(1)}n_b^{(2)}z_{a,b}\phi_{a,b}(\mathbf{x})}{\sum_{i=1}^{B_1}\left[\sum_{j=1}^{B_2} n_i^{(1)}n_j^{(2)}z_{i,j}\phi_{i,j}(\mathbf{x})\right]} \\
&= \frac{z'_{a,b}\phi_{a,b}(\mathbf{x})}{\sum_{i=1}^{B_1}\left[\sum_{j=1}^{B_2} z'_{i,j}\phi_{i,j}(\mathbf{x})\right]}
\end{aligned}
$$

Firstly, this shows that the spatial distribution of features, in the case of fusion multiple spatiograms, is a Gaussian divided by a weighted sum of B Gaussians, $B = B_1 B_2$, as it is for a single spatiogram. Therefore, the approximation of the joint-distribution obtained by using partial spatiograms is itself a spatiogram. Secondly, this spatiogram is given by $\bar{n}_{a,b} = z'_{a,b}$, with $\bar{\mu}_{a,b}$ and $\bar{\Sigma}_{a,b}$ given by equations (23) and (22). It is similar to the histogram approximation, but adds more weight to joint-feature bins whose partials have significant overlap in their spatial layout.