A Graph-based Feature Combination Approach to Object Tracking

A Graph-based Feature Combination Approach to Object Tracking

Quang Anh Nguyen[1,2], Antonio Robles-Kelly[1,2], and Jun Zhou[1,2]

[1] RSISE, Bldg. 115, Australian National University, Canberra ACT 0200, Australia
[2] National ICT Australia (NICTA)*, Locked Bag 8001, Canberra ACT 2601, Australia
{Quang.Nguyen,Antonio.Robles-Kelly,Jun.Zhou}@nicta.com.au

Abstract. In this paper, we present a feature combination approach to object tracking based upon graph embedding techniques. The method presented here abstracts the low complexity features used for purposes of tracking to a relational structure and employs graph-spectral methods to combine them. This gives rise to a feature combination scheme which minimises the mutual cross-correlation between features and is devoid of free parameters. It also allows an analytical solution making use of matrix factorisation techniques. The new target location is recovered making use of a weighted combination of target-centre shifts corresponding to each of the features under study, where the feature weights arise from a cost function governed by the embedding process. This treatment permits the update of the feature weights in an on-line fashion in a straightforward manner. We illustrate the performance of our method in real-world image sequences and compare our results to a number of alternatives.

1 Introduction

Object tracking is a classical problem in computer vision and pattern recognition. Existing approaches often employ low complexity local image descriptors and features to construct a model that can then be used to track the object. These features can be based upon the RGB values of the image under study, local texture descriptors and contrast operators [1]. The responses of the image brightness to Harr-like [2], Gaussian and Laplacian filters [3] have also been used for recognition and tracking.

Along these lines, modern appearance-based tracking frameworks such as the kernel-based methods [4], Kalman filter [5] and particle filter trackers [6] have attracted a great deal of attention from the computer vision community. The well known kernel-based algorithm [4] makes use of the mean-shift optimisation scheme [7] to search for a local maximum of feature similarity on the image lattice, without prior knowledge of the tracking environment. The Kalman filter [5] and the particle filter trackers [6] improve the tracking robustness by introducing probabilistic models for object and camera motion as well as state-space hypotheses.

However, it is somewhat surprising that the methods above do not combine multiple cues, but rather employ a fixed set of colour feature spaces such as RGB [4] or HSV [6].

* NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program.

Hence they are prone to error in practical settings where the illumination conditions and object appearance vary significantly between subsequent frames. Stern and Efros [8] improve the tracking performance by adaptively swapping the tracking features across five pre-determined colour-space combinations. Nguyen and Smeulders [9] use a set of Gabor filters to transform the image intensities into texture information. Collins et al. [10] deploy the mean-shift tracker [4] on a feature pool of 49 log-likelihood images comprised by unique combinations of R,G and B values. In a related development, Han and Davis [11] combined two different colour spaces so as to construct 14 log likelihood images. Feature extraction is then achieved by performing PCA on the foreground and the local image background. Machine learning techniques such as Adaboost have also been employed to enhance multiple-feature trackers [12, 13].

In this paper, we aim at presenting a feature combination approach to object tracking. Here, we make use of graphical model setting so as to abstract the features used in the tracking process into a graph. This leads to the use of techniques commonly employed in graph-spectral methods [14] to achieve maximum separation between the target and the scene background . Thus, here we provide a link between graphical models, graph embedding methods and tracking feature correlation. This treatment is devoid of free parameters and windowed sampling, while permitting low complexity features to be linearly combined analytically.

Moreover, the use of graph embedding techniques also leads to the recovery of a set of weights so as to evaluate the contribution of each feature to the target shift. This is reminiscent of boosting techniques [15], where a weak leaner is used for classification. In this way, our method can be viewed as a weighted linear combination of "weak" mean-shifts in each feature space which are combined into a "strong" global one. We also present an on-line updating scheme for the weights governing the tracking task. In practice, this is done based on the level of "confidence" on the target position and leads to the updating of the target model. Further, our approach can employ any arbitrary number of low complexity local image features and is not limited to colour cues.

The paper is organised as follows. Firstly, we introduce the basic concepts that will be used throughout the paper. We then turn our attention to the recovery of a global mean-shift from the contributions of each feature space. The on-line weight updating scheme is presented in Section 4. Finally, we elaborate on the algorithm in Section 5 and, in Section 6, we illustrate the robustness of the algorithm on a number of video sequences and compare our results to those delivered by alternatives.

2 Kernel-based Tracking in Arbitrary Feature Spaces

As mentioned earlier, Kernel-based object tracking [4] makes use of the spatially-weighted histogram of the target region as input to a similarity function which the tracker aims at maximising via mean-shift iterations [16].

In order to characterise a target, one or more feature spaces must be determined so that a non-parametric power density function (PDF) such as M-bin histogram can be estimated. The ideal choice of feature space is the one that is distinctive to the target with respect to the surrounding background while being robust to noise and image

corruption. The principle of the kernel-based tracker is, however, not restricted to any particular feature space and, in a multiple feature setting, can be summarised as follows.

Let $\mathbf{\Phi} = \{\phi_1, \phi_2, \ldots, \phi_{|\mathbf{\Phi}|}\}$ be the set of feature-spaces used for purposes of tracking. For the feature space ϕ_i, the new target position η_{ϕ_i} can be recovered making use of the two M-bin histograms $\mathbf{Q}_{\phi_i}$ and $\mathbf{P}_{\phi_i}$ corresponding to the target model and the search window, respectively. In particular,

$$\eta_{\phi_i} = \frac{\sum_{n=1}^{N} x_n w_n}{\sum_{i=1}^{N} w_n} \tag{1}$$

where w_n is the similarity weight for the n^{th} pixel x_n in the search window. For further detail on the equation above, we direct the reader to [4].

With the $|\mathbf{\Phi}|$ "weak" shifts $\{\eta_{\phi_i}\}_{\phi_i \in \mathbf{\Phi}}$ at hand, we can compute the "global" shift η as the weighted average of these "weak" shifts as $\eta = \sum_{i=1}^{|\mathbf{\Phi}|} \gamma_{\phi_i} \eta_{\phi_i}$ where γ_{ϕ_i} is the feature weight for the updated target-centre η_{ϕ_i} corresponding to the feature space ϕ_i.

3 Feature Combination via Graph Embedding

We now turn our attention to the recovery of the feature weight γ_{ϕ_i}. To this end, we cast the problem of feature combination into a graph-theoretic setting. In this manner, we aim at embedding the set of pairwise correlations between features in a metric space. To do this, we abstract the pairwise relationships between low complexity features into a relational structure and make use of graph-spectral methods, i.e. the eigenvalues and eigenvectors of the Laplacian matrix [17], so as to cast the feature weight γ_{ϕ_i} in an optimisation setting that leads to a Rayleigh Quotient. This can be viewed as the recovering of a graph embedding such that the correlation between features is minimum.

This embedding process commences by viewing the PDFs for the target foreground and its surrounding background as nodes on a weighted graph, whose edge-weights are given by their correlation in its geometric sense, i.e. the inner product of the pairwise PDFs. Viewed in this way, the Laplacian of the graph can be related to a Gram matrix of scalar products. This treatment, in turn, allows the use of matrix factorisation techniques to recover the coordinates for the embedding of the graph. Thus, the problem of finding the feature weight γ_{ϕ_i} turns into that of recovering the set of variables that maximises the pairwise distances between the features under consideration and, therefore, minimises their cross-correlation via the use of the eigenvalues and eigenvectors of a purposely-constructed matrix.

3.1 Feature Mapping

To commence, we require some formalism. Let $G = (V, E, W)$ denote a weighted graph with index-set V, edge-set $E = \{(u, v)|(u, v) \in V \times V\}$ and edge-weights $W : E \rightarrow [0, 1]$. Recall that, as mentioned earlier, the nodes of the graph are the PDFs for the target model and the scene background, i.e.$\{\mathbf{Q}_{\phi_i}\}_{\phi_i \in \mathbf{\Phi}}$ and $\{\mathbf{P}_{\phi_i}\}_{\phi_i \in \mathbf{\Phi}}$ respectively. As a result, we let the weight $W(u, v)$ associated with the edge connecting

the pair of nodes u and v corresponding to the i^{th} and j^{th} features in $\mathbf{\Phi}$ be given by the normalised cross-correlation

$$W(u,v) = \begin{cases} \left\langle \frac{\mathbf{Q}_{\phi_i}}{||\mathbf{Q}_{\phi_i}||}, \frac{\mathbf{P}_{\phi_i}}{||\mathbf{P}_{\phi_i}||} \right\rangle & \text{if } i = j \\ \left\langle \frac{\mathbf{Q}_{\phi_i}}{||\mathbf{Q}_{\phi_i}||}, \frac{\mathbf{Q}_{\phi_j}}{||\mathbf{Q}_{\phi_j}||} \right\rangle & \text{otherwise} \end{cases} \tag{2}$$

Note that W is a symmetric matrix of scalar products, in which the diagonal elements are given by the cross-correlation between the PDFs for the foreground and the background of the same feature, while the off-diagonal elements are the cross-correlation between the PDFs for the foreground for different features.

To take our analysis further, we proceed to define the squared distance between features on the graph. Here, we set the pairwise squared distance between a pair of nodes as their correlation value. This is akin to the approaches in pairwise grouping such that in [18]. We define

$$W(u,v) = \|\varphi(u) - \varphi(v)\|^2 \tag{3}$$

where $\varphi(u)$ is the embedding vector, i.e. the vector of coordinates for the feature ϕ_i corresponding to the node u in V. The squared distance can also be expressed in terms of a set of inner products as follows

$$W(u,v) = \langle \varphi(u), \varphi(u) \rangle + \langle \varphi(v), \varphi(v) \rangle - 2 \langle \varphi(u), \varphi(v) \rangle \tag{4}$$

This permits viewing the correlation between tracking features as pairwise distances in a metric space making use of the inner products.

3.2 Double Centering

To provide a link between the edge-weights $W(u,v)$ and the coordinate vectors $\varphi(u)$, we make use of double-centering [19]. In particular, this can be achieved by firstly relating the edge-weight matrix W to the Laplacian matrix $\mathcal{L}$ [14]. With the Laplacian matrix at hand, a double-centered matrix of scalar products $\mathbf{H} = \mathbf{J}\mathbf{J}^T$ can be computed. This operation introduces a linear dependency over the columns of the matrix $\mathbf{H}$ while preserving the symmetry of W.

This treatment is important because it allows us to view the double centered matrix $\mathbf{H}$ as a matrix of scalar products which can then be interpreted as the sums of squared, pairwise distances $\|\varphi(u) - \varphi(v)\|^2$ introduced in Equation 3. Furthermore, it can be shown that the matrix $\mathbf{H}$ is, in fact, the double-centered graph Laplacian [19]. As a result, the element of the matrix $\mathbf{H}$ corresponding to the nodes $u, v \in V$ is given by

$$\mathbf{H}(u,v) = -\frac{1}{2}\left[\mathcal{L}(u,v)^2 - \frac{1}{|V|}\sum_{u \in V}\mathcal{L}(u,v)^2 - \frac{1}{|V|}\sum_{v \in V}\mathcal{L}(u,v)^2 + \frac{1}{|V|^2}\sum_{u,v \in V}\mathcal{L}(u,v)^2\right] \tag{5}$$

The graph Laplacian $\mathcal{L}$ is defined as $\mathcal{L} = \mathbf{D}^{-1/2}(\mathbf{D} - W)\mathbf{D}^{-1/2}$ where $\mathbf{D}$ is a diagonal matrix such that $\mathbf{D} = diag(deg(1), deg(2), \ldots, deg(|V|))$ and $deg(u) = \sum_{v \in V} W(u,v)$ is the degree of the node $u \in V$.

Let ξ_l be the l^{th} eigenvector of $\mathbf{H}$ scaled so its sum of squares is equal to the corresponding eigenvalue λ_l. Since $\mathbf{H}\xi_l = \lambda_l\xi_l$ and $(\mathbf{J}\mathbf{J}^T)\xi_l = \mathbf{H}\xi_l$, it follows that the squared distance between a pair of nodes in Equation 3 can be now written as

$$\| \varphi(u) - \varphi(v) \|^2 = \sum_{l=1}^{|V|} \lambda_l(\xi_l(u) - \xi_l(v))^2 = \mathbf{H}(u,u) + \mathbf{H}(v,v) - 2\mathbf{H}(u,v) \quad (6)$$

3.3 Minimising Feature Correlation

With these ingredients, we can introduce the variables $\pi(u)$ such that the weighted correlations between low complexity features are minimum. We do this by making use of the quantity

$$\epsilon = \sum_{u,v \in V} \left\| \pi(u)\varphi(u) - \pi(v)\varphi(v) \right\|^2 \quad (7)$$

which we aim at minimising. The cost function above can also be interpreted as the sum of squared weighted cross-correlations between the PDFs used for purposes of tracking. Thus, we can use Equation 6 and, after some algebra, we write

$$\epsilon = \sum_{u,v \in V} \left(\pi(u)^2\mathbf{H}(u,u) + \pi(v)^2\mathbf{H}(v,v) - 2\pi(u)\pi(v)\mathbf{H}(u,v) \right) \quad (8)$$

Note that, Equation 8 can be divided into two sets of terms. The first of these corresponds to the diagonal matrix of $\mathbf{H}$. The other set accounts for the off-diagonal elements of H. Rearranging terms, we get

$$\epsilon = 2|V| \sum_{u \in V} \pi(u)^2\mathbf{H}(u,u) - \sum_{\substack{u,v \in V \\ u=v}} 2\pi(u)^2\mathbf{H}(u,u) - \sum_{\substack{u,v \in V \\ u \neq v}} 2\pi(u)\pi(v)\mathbf{H}(u,v) \quad (9)$$

where we use the following facts

$$\sum_{u,v \in V} \pi(u)^2\mathbf{H}(u,u) = |V| \sum_{u \in V} \pi(u)^2\mathbf{H}(u,u) \text{ and } \sum_{u,v \in V} \pi(u)^2\mathbf{H}(u,u) = \sum_{u,v \in V} \pi(v)^2\mathbf{H}(v,v)$$

Moreover, Equation 9 can be reduced to

$$\epsilon = - \sum_{\substack{u,v \in V \\ u \neq v}} 2\pi(u)\pi(v)\mathbf{H}(u,v) \quad (10)$$

which can be written in compact form by defining a matrix $\hat{\mathbf{H}}$ which comprises the off-diagonal elements of $\mathbf{H}$ as follows

$$\hat{\mathbf{H}}(u,v) = \begin{cases} \mathbf{H}(u,v) & \text{if } u \neq v \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

This yields $\epsilon = -2\mathbf{\Pi}^T\hat{\mathbf{H}}\mathbf{\Pi}$ where $\mathbf{\Pi} = [\pi(1), \pi(2), \cdots, \pi(|V|)]^T$ is a column vector of order $|V|$. Note that the expression above is the numerator of a Rayleigh

Quotient whereas the omitted denominator, $\mathbf{\Pi}^T\mathbf{\Pi}$, is a normalisation constant. Thus, minimising ϵ is equivalent to maximising $\mathbf{\Pi}^T\hat{\mathbf{H}}\mathbf{\Pi}$ and, therefore, $\mathbf{\Pi}^* = \underset{\mathbf{\Pi}}{\operatorname{argmin}}\ \{\epsilon\}$ is given by the leading eigenvector of $\hat{\mathbf{H}}$ which corresponds to the eigenvalue whose rank is the largest.

The vector $\mathbf{\Pi}^*$, hence, is the minimiser of the squared distances between the nodes in the graph, i.e. the correlation between features. As a result, the set of feature weights $\{\gamma_{\phi_i}\}_{\phi i \in \mathbf{\Phi}}$ corresponding to the "weak" shifts is given by

$$\gamma_{\phi_i} = \frac{\pi(u)}{\sum_{u \in V} \pi(u)} \tag{12}$$

where the i^{th} feature ϕ_i corresponds to the node u in V.

4 On-line Feature Weight Updating

As kernel-based trackers [4, 10, 11] rely on the M-bin histograms of the model to determine the target location via the mean-shift optimisation scheme, the validity of these histograms is extremely important for robust tracking. In [10], these M-bin histograms are modified after every frame by randomly selecting pixels from the target foreground so as to modify the tracking models across the feature spaces. Despite effective, this "mixing" method does not discriminate between pixels and, hence, is susceptible to mislocalisation due to histogram bias.

Here, we present an on-line feature weight updating method based upon the cross-correlation between the histograms of the current target model and that corresponding to the recovered target-centre after each mean-shift application. This technique is based upon the weighted cross correlation between histograms and, thus, is devoid of pixel-sample selection and injection. Moreover, we calculate the total cross-correlation in a similar manner to that in Section 3.

To commence, let $\{\hat{\mathbf{Q}}_{\phi_i}\}_{\phi_i \in \mathbf{\Phi}}$ be the set of the M-bin histograms obtained from the new target position and ϱ_{ϕ_i} be the cross-correlation between the two histograms $\mathbf{Q}_{\phi_i}$ and $\hat{\mathbf{Q}}_{\phi_i}$, i.e. $\varrho_{\phi_i} = \left\langle \frac{\mathbf{Q}_{\phi_i}}{\|\mathbf{Q}_{\phi_i}\|}, \frac{\hat{\mathbf{Q}}_{\phi_i}}{\|\hat{\mathbf{Q}}_{\phi_i}\|} \right\rangle$. The total cross-correlation between the two sets of histograms, $\{\hat{\mathbf{Q}}_{\phi_i}\}_{\phi_i \in \mathbf{\Phi}}$ for the new target position and $\{\mathbf{Q}_{\phi_i}\}_{\phi_i \in \mathbf{\Phi}}$ for the current model, can be computed as a linear combination of the weighted feature cross-correlation ϱ_{ϕ_i} as

$$\mathit{\Gamma} = \sum_{\phi_i \in \mathbf{\Phi}} \gamma_{\phi_i} \varrho_{\phi_i} \tag{13}$$

where $\{\gamma_{\phi_i}\}_{\phi_i \in \mathbf{\Phi}}$ is the set of feature weights derived from Section 3. This treatment, in turn, allows us to set decision bounds for the updating operation. We do this by updating the model M-bin histograms only when the condition $0 \leq \kappa_0 < \mathit{\Gamma} < \kappa_1 \leq 1$ is satisfied, where κ_0 and κ_1 are constants. This hinges in the confidence of the tracking operation by following the notion that, if the total correlation between the new target-centre histograms and that of the target model is close to unity, there is no need to update since the two sets are sufficiently "close". On the contrary, if the total correlation is too

Algorithm 1: Training

Data: Selected region of the target model.
begin
  Sample N_1 pixels from the foreground, N_2 pixels from the background.
  Compute the set of M-bin histograms $\{\mathbf{Q}_{\phi_i}\}_{\phi_i \in \mathbf{\Phi}}$ and $\{\mathbf{P}_{\phi_i}\}_{\phi_i \in \mathbf{\Phi}}$ across the feature spaces $\mathbf{\Phi}$.
  Compute W as in Equation 2
  Compute $\mathbf{H}$ as in Equation 5 and $\hat{\mathbf{H}}$ as in Equation 11
  Compute $\mathbf{\Pi}^*$ as the leading eigenvector ξ_1 of $\hat{\mathbf{H}}$
  Compute the feature weights $\{\gamma_{\phi_i}\}_{\phi_i \in \mathbf{\Phi}}$ using $\mathbf{\Pi}^*$ as in Equation 12
  Save $\{\gamma_{\phi_i}\}_{\phi_i \in \mathbf{\Phi}}$ and $\{\mathbf{Q}_{\phi_i}\}_{\phi_i \in \mathbf{\Phi}}$
end

low, then updating would "corrupt" the model. Updating is hence, appropriate when the correlation is not so low so as to introduce noise corruption but not as high as to be a computational burden without improving tracking accuracy.

When update operations are deemed necessary, the histogram set $\{\mathbf{Q}_{\phi_i}\}_{\phi_i \in \mathbf{\Phi}}$ for the target model is updated making use of a mixture model of the form

$$\mathbf{Q'}_{\phi_i} = P(\hat{\mathbf{Q}}_{\phi_i}|\varrho_{\phi_i})\hat{\mathbf{Q}}_{\phi_i} + \left(1 - P(\hat{\mathbf{Q}}_{\phi_i}|\varrho_{\phi_i})\right)\mathbf{Q}_{\phi_i} \tag{14}$$

This can be viewed as a "blending" operation between the two histograms. It is, indeed, a two-class expectation for the two PDFs $\hat{\mathbf{Q}}_{\phi_i}$ and $\mathbf{Q}_{\phi_i}$, whose prior is given by the probability of the new target position given the feature cross-correlations ϱ_{ϕ_i}.

5 Algorithm Description

With the developments presented in the previous sections, the tracking algorithm can be divided into two stages. The first stage is the training phase, in which the user is required to select the target to track. The samples inside the selected region are then used to compute a set of PDFs corresponding to the feature spaces under study. In a similar manner, the area around the target is also sampled to create a set of background PDFs. In our implementation, for the sake of efficiency, we perform background sampling in the area of twice the size of the target. With the two sets of foreground and background PDFs at hand, we compute the corresponding cross-correlation weight matrix W. Subsequently, the double-centering matrix $\mathbf{H}$ is determined, followed by its off-diagonal matrix $\hat{\mathbf{H}}$. The set of feature weights $\{\gamma_{\phi_i}\}_{\phi_i \in \mathbf{\Phi}}$ is then recovered from the leading eigenvector of $\hat{\mathbf{H}}$.

In the second stage, the tracking vehicle is the mean-shift tracker presented in [4]. After each new target position, the total cross-correlation $\mathit{\Gamma}$ is then calculated to determine if the set of model histograms needs to be updated, i.e. $\kappa_0 < \mathit{\Gamma} < \kappa_1$. This implies that the feature weights and the target-model feature histograms will only be updated if the tracking operation is reliable, i.e. with a $\mathit{\Gamma} > \kappa_0$, while keeping computational cost low by avoiding updating operations when the candidate and the model are virtually the same, i.e. with a $\mathit{\Gamma} < \kappa_1$.

Algorithm 2: Tracking

Data: $\{\gamma_{\phi_i}\}_{\phi_i \in \mathbf{\Phi}}$, $\{\mathbf{Q}_{\phi_i}\}_{\phi_i \in \mathbf{\Phi}}$ and target centre y
begin
  for idx=*StartFrame to EndFrame* **do**
    while *true* **do**
      Compute the set of M-bin histograms $\{\mathbf{P}_{\phi_\mathbf{i}}\}_{\phi_i \in \mathbf{\Phi}}$ for the searching window
      Compute the target centre $\{\eta_{\phi_i}\}_{\phi_i \in \mathbf{\Phi}}$ of each mean-shift as in Equation 1
      Compute the new target centre $\eta = \sum_{\phi_i \in \mathbf{\Phi}} \gamma_{\phi_i} \eta_{\phi_i}$
      if $\|\eta - y\| \leq \varepsilon$ **then**
        $idx = idx + 1$
        break
      else
        Update the target centre $y = \eta$
      end
    end
    Compute $\{\hat{\mathbf{Q}}_{\phi_i}\}_{\phi_i \in \mathbf{\Phi}}$ at the new target centre
    Compute $\varrho_{\phi_i} = \left\langle \frac{\mathbf{Q}_{\phi_i}}{\|\mathbf{Q}_{\phi_i}\|}, \frac{\hat{\mathbf{Q}}_{\phi_i}}{\|\hat{\mathbf{Q}}_{\phi_i}\|} \right\rangle$
    Compute $\Gamma = \sum_{\phi_i \in \mathbf{\Phi}} \gamma_{\phi_i} \varrho_{\phi_i}$
    if $\kappa_0 < \Gamma < \kappa_1$ **then**
      Compute $P\left(\hat{\mathbf{Q}}_{\phi_i} | \varrho_{\phi_i}\right)$
      Update $\{\mathbf{Q}_{\phi_i}\}_{\phi_i \in \mathbf{\Phi}}$ to $\{\mathbf{Q'}_{\phi_i}\}_{\phi_i \in \mathbf{\Phi}}$ using Equation 14
      Compute the new feature weights $\{\gamma_{\phi_i}\}_{\phi_i \in \mathbf{\Phi}}$ using the updated $\{\mathbf{Q}_{\phi_i}\}_{\phi_i \in \mathbf{\Phi}}$ as in Algorithm 1
    end
  end
end

6 Experiments

In this section, we illustrate the robustness of our algorithm by presenting results on two image sequences from the PETS-ECCV 2004 dataset[3]. Note that further sequences can be found in the supplemental material accompanying this paper. In the first sequence, the target moves from a bright area in the scene to a shady region, meets another person and then walks away. The second sequence shows a group of four people moving across the scene with some body-overlapping as they approach the camera. For each of these sequences, the tracking target is manually selected by the user at the initial frame.

We have compared our results to those yielded by two competing algorithms. These are the on-line Variance Ratio-based (VR-based) method proposed by Collins et al. [10] and the on-line PCA-based method by Han and Davis [11]. Note that these methods [10, 11] have significant improvement in performance over the random weights. We have also implemented two sets of features. The first set consists of 49 linear combinations of R,G,B as described in [10]. We call this set the 49-feature set. The second set is a mix of gradient, contrast and texture features including brightness, normalised RGB, Local

[3] PETS dataset can be accessed from http://www.cvg.rdg.ac.uk/slides/pets.html

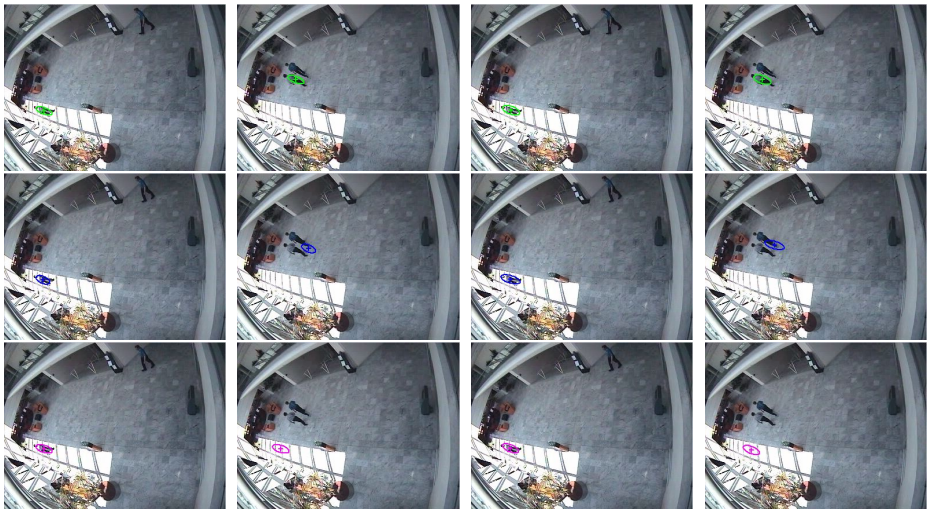
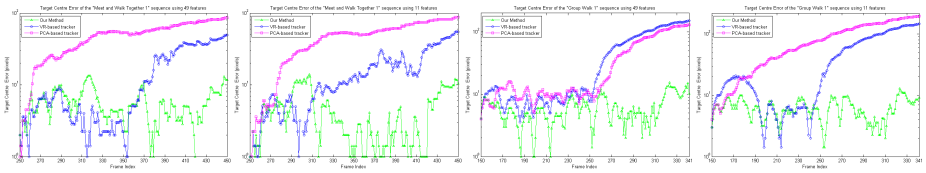


Fig. 1. Results for the "Meet and Walk Together 1" sequence at frames 150 and 380. From top-to-bottom: results yielded by our algorithm using the 49-feature set (first and second columns) and the 11-feature set (third and forth columns), the on-line VR-based tracker [10] using the 49-feature set and the 11-feature set, and the on-line PCA-based tracker [11] using the 49-feature set and the 11-feature set.

Binary Patterns (LBPs) and six Haar-like features [2], which we call the 11-feature set. The Harr-like features include vertical and horizontal 2,3 and 4-rectangle features.

In our implementation of the VR-based tracker, we select the set of five log-likelihood images that yield the highest Variance Ratio as the tracking features for the current frame. For the PCA-based tracker, the eigenvectors associated with the eigenvalues whose normalised sum is greater than 0.7 are used so as to reduce the dimensionality of the log-likelihood images. As suggested in [11], a Gaussian filter is also implemented in order to reduce the amount of unwanted noise in the likelihood image corresponding to the leading eigenvalue. For our tracker, we consider the conditional probability for the update operations to be normally distributed, i.e. $P(\hat{\mathbf{Q}}_{\phi_i}|\varrho_{\phi_i}) \sim N(\mu, \sigma)$. Moreover, we set $\mu = (\kappa_0 + \kappa_1)/2$ and $\sigma = (\kappa_1 - \kappa_0)/2$. This treatment allows the set of M-bin histograms for the target model to be updated based upon their individual correlations given the upper and lower bounds set for the update operation as a whole. We set the constants which govern the model update operations to $\kappa_0 = 0.7$ and $\kappa_1 = 0.9$.

In Figure 1, we present the sample results for frames 150 and 380 of the PETS-ECCV 2004 "Meet and Walk Together 1" sequence. In this sequence, the target appearance varies remarkably as it moves from the bright area into the shade between frames 290 and 310. As a result, the target model is subjected to significant change. Moreover, the target remains close to the other person from frame 330 onwards, which serves as a confounding factor that further complicates the tracking task. Despite these difficulties, the feature combination approach presented here allows the tracker to follow the target throughout the scene. This applies to both of the feature sets under consideration. The VR-based tracker [10], on the other hand, loses the target as the subject approaches the other person between frames 380 and 420. The PCA-based approach [11], however,



(a) (b) (c) (d)

Fig. 2. Target Center Error for our method, the on-line VR-based tracker [10], and the on-line PCA-based tracker [11]. (a)(b): "Meet and Walk Together 1" sequence using the 49-feature set and the 11-feature set, respectively; (c)(d): "Group Walk 1" sequence using the 49-feature set and the 11-feature set, respectively.

cannot adapt to the significant change in illumination and subsequently fails in tracking the target from frame 290 until the end of the footage.

We present a more quantitative analysis of the tracker performance in Figure 2(a) and (b). In these figures, we have plotted the target centre error as a function of frame index with respect to the ground truth provided with the PETS-ECCV 2004 dataset. For the sake of clarity, the error in the figure is shown in a logarithmic scale. Note that our tracker has the lowest mean target centre errors of 5.63 ± 2.77 pixels and 4.40 ± 3.44 pixels for the 49-feature set and the 11-feature set, respectively. The VR-based tracker [10], has a mislocalisation mean of 15.40 ± 15.44 pixels and 16.22 ± 13.69 pixels, respectively. The PCA-based tracker [11], being unable to track the target after frame 290, has a mean centre error of 49.86 ± 22.34 pixels and 48.27 ± 24.84 pixels. This is consistent with the behaviour described above.

We now turn our attention to the contribution of each feature to the global "strong" shift across the sequence. During the sequence, there are 22 updates during the footage. The first 20 updates occur between frames 250 and 310, in which the target moves across the bright area to the shady region in the scene. As a result, the appearance of the target varies significantly. During this frame range, the features such as the vertical and horizontal 3-rectangle Harr-like are assigned the highest weights, with an overall contribution of approximately 60%. In contrast, colour-based features such as the brightness and the normalised RGB channels are given much lower weights, with a contribution of less than 10%. The last few updates occur after frame 320, corresponding to the frames where the target moves completely inside the shady area. These adjustments reduce the weight of the Harr-like features and increase the contribution of the image brightness.

Moving on to the second of our experimental vehicles, Figure 3 shows the results for frames 250 and 280 of the PETS-ECCV 2004 "Group Walk 1" video sequence. The sequence records a group of four people moving across the scene, of which we track the female target. In this footage, there is no significant illumination change as in the previous sequence. However, as the group approaches the camera, their bodies overlap one another before exiting the scene. The similarity in their outfit colour further complicates the tracking task.

In this sequence, the VR-based tracker [10] performs well in the first 100 frames with both feature sets. However, it quickly loses the target once the target is partially occluded by another member of the group. This results in target centre errors of up to 46.47 ± 49.50 pixels and 46.75 ± 47.52 pixels, as shown in Figure 2 (c) and (d).

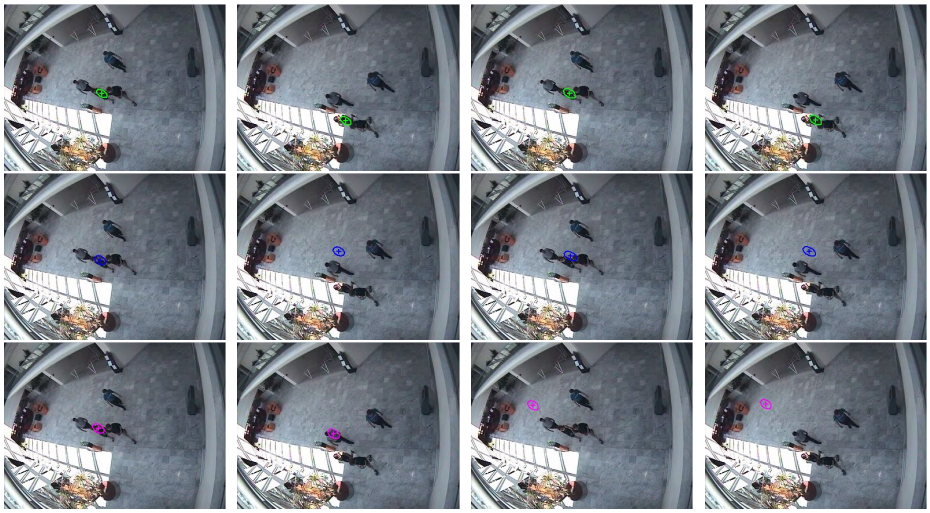


Fig. 3. Results for the "Group Walk 1"sequence at frames 250 and 280. From top-to-bottom: results yielded by our algorithm using the 49-feature set (first and second columns) and the 11-feature set (third and forth columns), the on-line VR-based tracker [10] using the 49-feature set and the 11-feature set, and the on-line PCA-based tracker [11] using the 49-feature set and the 11-feature set.

The performance of the PCA-based tracker [11] has high variation across the sequence. In particular, the PCA-tracker shows a similar performance to that of the VR-based tracker when the 49-feature set is used. It also loses the target at the frames where the subject bodies overlap, being unable to recovery afterwards. However, in the 11-feature set case, the PCA-tracker only manages to track the target in the first 20 frames. As a result, the error measurements are significant, 38.42 ± 41.00 pixels and 86.65 ± 60.16 pixels for the 49-feature set and the 11-feature set, respectively. For our tracker, the model integrity is preserved as a consequence of the use of the total correlation as a measure of tracking confidence. Our tracker successfully follows the target throughout the scene with low target centre-errors, i.e. 6.09 ± 3.03 pixels and 6.25 ± 2.31 pixels for the 49-feature set and the 11-feature set, respectively.

On the contribution of each feature to the global "strong" shift, there are 37 updates throughout the footage. These mainly occur when the subject bodies occlude one another. Nonetheless, the vertical 3-rectangle Harr-like feature is dominant across the sequence. From our experiments we also notice that the normalised RGB colour channels are not as discriminant as the other features in the set. This can be attributed to the fact that the clothing colour of the subjects in the scene does not separate the target from the rest of the crowd.

7 Conclusion

In this paper, we have presented a feature combination approach for object tracking. We have shown how the target-centre may be recovered from a weighted linear combination of "weak" mean-shifts. This feature combination method is based upon graph

embedding techniques. Thus, it provides a principled link between feature combination, graph-spectral methods and graphical models. The method performs on-line updating based upon the correlation between the target current model and that of the new target position at the current frame. The updating scheme presented here is governed by the reliability of the tracking process. As a result, our method can cope with confusing backgrounds, unexpected fast movements and temporary occlusions by taking advantage of the information drawn from multiple feature spaces corresponding to a number of visual cues. The approach is quite general in nature and can employ other features elsewhere in the literature. We have also compared our results to those delivered by alternative methods.

References

1. Nascimento, J.C., Marques, J.S.: Robust shape tracking in the presence of cluttered background. IEEE Transactions on Multimedia **6**(6) (2004) 852–861
2. Viola, P., Jones, M.: Robust real-time object detection. International Journal of Computer Vision **57**(2) (2002) 137–154
3. Varma, M., Zisserman, A.: Classifying images of materials: Achieving viewpoint and illumination independence. In: European Conf. on Computer Vision. Volume 3. (2002) 255–271
4. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. IEEE Trans. on Pattern Analysis and Machine Intelligence **25**(5) (2003) 564–577
5. Wren, C.R., Azarbayejani, A., Darrell, T., Pentland, A.P.: Pfinder: Real-time tracking of the human body. IEEE Trans. Pattern Anal. Mach. Intell. **19**(7) (1997) 780–785
6. Pérez, P., Hue, C., Vermaak, J., Gangnet, M.: Color-based probabilistic tracking. In: European Conf. on Comp. Vision. Volume 2350. (2002) 661–675
7. Cheng, Y.: Mean shift, mode seeking, and clustering. IEEE Trans. on Pattern Analysis and Machine Intelligence **17**(8) (August 1995) 790–799
8. Stern, H., Efros, B.: Adaptive color space switching for face tracking in multi-colored lighting environments. In: Int. Conf. on Automatic Face and Gesture Recognition. (2002) 249
9. Nguyen, H., Smeulders, A.: Tracking aspects of the foreground against the background. In: European Conf. on Computer Vision. Volume 2. (2004) 446–456
10. Collins, R., Liu, Y., Leordeanu, M.: On-line selection of discriminative tracking features. IEEE Trans. on Pattern Analysis and Machine Intelligence **27**(10) (2005) 1631 – 1643
11. Han, B., Davis, L.: Object tracking by adaptive feature extraction. In: International Conf. on Image Processing. Volume 3. (2004) 1501–1504
12. Avidan, S.: Ensemble tracking. In: IEEE Conf. on Computer Vision and Pattern Recognition. Volume 2. (2005) 494–501
13. Grabner, H., Bischof, H.: On-line boosting and vision. In: IEEE Conf. on Computer Vision and Pattern Recognition. Volume 1. (2006) 260–267
14. Chung, F.R.K.: Spectral Graph Theory. American Mathematical Society (1997)
15. Freund, Y.: Boosting a weak learning algorithm by majority. In: Proceedings of the Workshop on Computational Learning Theory. (1990) 202–216
16. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE Trans. on Pattern Analysis and Machine Intelligence **24**(5) (2002) 603–619
17. Chavel, I.: Riemannian Geometry: A Modern Introduction. Cambridge University Press (1995)
18. Robles-Kelly, A.: Segmentation via graph-spectral methods and riemannian geometry. In: International Conf. on Computer Analysis of Images and Patterns. (2005) 661–668
19. Borg, I., Groenen, P.: Modern Multidimensional Scaling, Theory and Applications. Springer Series in Statistics. Springer (1997)