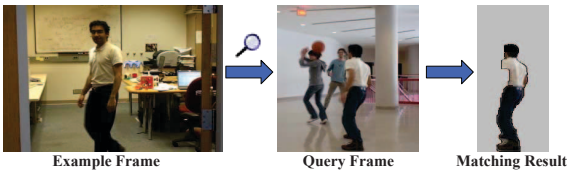
A GRAPH-BASED FOREGROUND REPRESENTATION AND ITS APPLICATION IN EXAMPLE BASED PEOPLE MATCHING IN VIDEO

Kedar A. Patwardhan, Guillermo Sapiro

University of Minnesota
Department of Electrical and Computer Engineering
Minneapolis, MN 55455, {kedar,guille}@umn.edu

Vassilios Morellas

University of Minnesota
Department of Computer Science
Minneapolis, MN 55455



ABSTRACT

In this work, we propose a framework for foreground representation in video and illustrate it with a multi-camera people matching application. We first decompose the video into foreground and background. A low-level coarse segmentation of the foreground is then used to generate a simple graph representation. A vertex in the graph represents the "appearance" of a corresponding segment in the foreground, while the relationship between two segments is encoded by an edge between the corresponding vertices. This provides a simple yet powerful and general representation of the foreground, which can be very useful in problems such as people detection and tracking. We illustrate the effectiveness of this model using an "example based query" type of application for people matching in videos. Matching results are provided in multiple-camera situations and also under occlusion.

Index Terms— Video, Image matching, Image analysis, Machine vision.

1. INTRODUCTION AND PREVIOUS WORK

The efficient representation of foreground objects in videos is a significant component of important computer vision applications such as tracking, detection, matching, and video-indexing. Most often, the representations proposed in the literature are highly task or situation specific, involve computationally prohibitive offline training, and do not effectively handle changes in scale, pose, or background. In this paper we propose the use of a graphical model to represent foreground objects in a video. The automatically detected foreground is (coarsely) segmented into connected segments or "superpixels" (borrowing a term from [1]). Each of these segments is considered as a vertex in our graphical model, and the relationship between segments is represented by an edge. This model helps to capture the local information, i.e, appearance of the segments in the foreground, without any explicit shape model. The representation of interaction between segments using edges allows us to incorporate spatial inter-relationships without using an absolute point of reference.

Previous work on the representation of foreground objects (people) in a video scene comes from two main areas in computer vision, namely, tracking and pose detection. The Hydra system, [2], represents foreground people as a combination of a "head-detector" and an intensity based template correlation. This requires the head of the person to always be part of the silhouette and the appearance template uses the head center as a spatial origin. McKenna *et al.*, [3], represent different people in the foreground using a color histogram of the person. As shown in [4], such histogram based representations cannot discriminate correctly between two objects (as they can have the same color distribution) without additional spatial information. In [5], blobs corresponding to people in the foreground are assigned to different body parts (head and hands) to track single individuals in a scene. The authors of [6] segment people (in a single camera) under occlusion by representing them as a group of 3 segments (head + torso + limbs), and then estimating the best arrangement for people in the scene using maximum-likelihood estimation. The head of the person is assumed to be visible throughout the occlusion in order to estimate the origin for the appearance model. Work in [7] reports the use of a person model learned offline to detect and represent the people in the foreground, and an appearance model of the person is learned online for tracking a particular person. Recent works on pose estimation like [8] utilize a loose limbed model to represent the 3-D pose of a person in the foreground. Motion capture data aligned with a coordinate frame of the calibrated cameras is used to estimate and detect the loosely connected limbs with a non-parametric belief propagation algorithm. In [9], the authors use a Bayesian framework to combine pictorial structure spatial models with hidden Markov temporal models to represent a person in a video. More recently, the authors in [10] have demostrated a representation of cartoon images using graphs, and presented a principled way of searching for subgraphs. Results do not indicate matching in the presence of substantial occlusion as shown in our work, especially using real videos from single or multiple cameras.

In this paper we propose a model for foreground representation which combines low-level segmentation and spatial reasoning in a meaningful way. This framework is intuitively ideal and very flexible for high level tasks in foreground analysis, foreground indexing and retrieval, and other applications in multiple-camera scenarios. We use an *example based people matching* application to demonstrate the practical utility of our model. The remainder of this paper is organized as follows: Section 2 gives a brief description of the proposed graphical representation. The matching application is detailed in Section 3. Section 4 discusses the matching results and implementation issues, and finally we conclude with future directions in

Work partially supported by ONR, NSF, NGA, DARPA, and the McKnight Foundation.

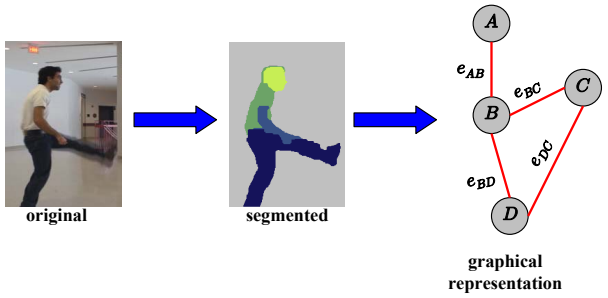


Fig. 1. *Foreground from the original frame (left) is segmented (center) using the algorithm in [11], to generate a graphical model (right).*

Section 5.

2. THE FOREGROUND MODEL

Encoding the local appearance as well as spatial relationships between objects in a scene has always been a very challenging problem in computer vision. In this paper we propose a graph structure to represent the foreground in a scene as a combination of local appearance models of image segments (vertices) along with a model of the relationship between them (edges). Figure 1 depicts the proposed model. The graphical model is generated after 2 preprocessing steps: (a) Foreground/Background separation, and (b) Foreground Segmentation.

The video is first decomposed into "foreground+background" by using our layering based foreground detection technique detailed in [12]. This decomposition is very robust to motion in the background (moving trees, water ripples, shaky camera, etc) and provides a real-time foreground detection capability. Once the foreground is detected, we perform a low-level color-based segmentation of the foreground objects using the algorithm proposed in [11] (other attributes beyond color could be used as well). Now, each segment in the foreground is represented by a vertex in our graphical model. The space (and/or time) relationships between the segments is modeled by graph edges (bi-directional). It should be noted that this representation is different than a typical graph, the edge is not just a connecting mechanism between two vertices carrying some weight, the edge actually models the relationship between the two segments, just like the vertex models the appearance of the segment. This framework can thus allow for inline learning and tracking of the various components in the scene. As a preliminary investigation, the following sections demonstrate the capability of this type of foreground representation in addressing the difficult problem of *example based people matching in multiple camera scenarios*.

3. ILLUSTRATIVE APPLICATION: EXAMPLE BASED PEOPLE MATCHING

Finding or matching a person viewed in one scene in the same or a completely different setting is an important problem in applications such as surveillance. In this work, we assume that we are given an "example" of an isolated person (marked by the security guard for example), and we want to search for the person viewed from the same or different (and non-overlapping) camera location. To this end, we use the graphical representation proposed above to convert this problem into a sub-graph matching task, Figure 2.

3.1. Problem Setup and Similarity Measures

Consider the situation in Figure 2. The foreground in the example frame has been segmented into vertices A_i connected by the edges $e_{A_{ij}}$ where $i, j \in (1, 2, ..., N_A)$ and $i \neq j$, N_A being the number of vertices in the example (segments in the foreground). The edge connection rule is very simple, we connect two vertices by an edge only when the corresponding segments are connected. Each segment in the foreground is modeled using the color histogram[1] generated using non-parametric Kernel Density Estimation (refer to [13]). We also use the SIFT features (refer to [14]) detected inside the segment to model its appearance.

The similarity $\mathfrak{S}(A \rightarrow B)$ of the segment (vertex) A in the example frame to a segment B in the queried frame is computed as a product of the color and feature similarities:

$$\mathfrak{S}(A \rightarrow B) = \rho(A, B) f(A \rightarrow B), \tag{1}$$

where, $\rho(A, B)$ is the Bhattacharya coefficient, [15], and $f(A \rightarrow B)$ is the feature similarity:

$$\rho(A, B) = \int \sqrt{p_A(\mathbf{x}) p_B(\mathbf{x})} dx, \tag{2}$$

and

$$f(A \rightarrow B) = \frac{1}{M_{A_{sift}}} e^{-d((sift_A),(sift_B))}, \tag{3}$$

where, $p_A(\mathbf{x})$ and $p_B(\mathbf{x})$ are the density estimates for pixel $\mathbf{x}$ computed from the color distributions in segment A and B respectively, $d((sift_A), (sift_B))$ is the sum of squared Euclidean distances between the features in B that are the best matches for features in A, and $M_{A_{sift}}$ is the number of SIFT features in the segment A.

Apart from this similarity meassure, we also use an edge or relationship similarity measure $\Psi(e_{A_{ij}} \rightarrow e_{B_{kl}})$. The spatial relationship between two connected segments/vertices *in the example graph* is modeled by the graph edge as a simple 2-D Gaussian $\mathcal{N}(\mu_r, \mu_\theta; \sigma_r, \sigma_\theta)$, where μ_r is the magnitude of the vector connecting the centroid of the adjacent segments (vertices), μ_θ is the angle between this vector and the horizontal axis, and σ_r and σ_θ are the allowed variances respectively.[2] Thus, the similarity between the *relationship between two vertices in the example frame* ($e_{A_{ab}}$) with respect to the *relationship between two vertices in the query frame* ($e_{B_{cd}}$) is computed as:

$$\Psi(e_{A_{ab}} \rightarrow e_{B_{cd}}) = \frac{exp(-\frac{(r_{cd}-\mu_{r_{ab}})^2}{2{\sigma_r}^2} - \frac{(\theta_{cd}-\mu_{\theta_{ab}})^2}{2{\sigma_\theta}^2})}{2\pi\sigma_r\sigma_\theta}, \tag{4}$$

where r_{cd} is the magnitude of the vector connecting the centroid of adjacent vertices corresponding to $e_{B_{cd}}$, and θ_{cd} is the angle of this vector with the horizontal axis.

3.2. Matching Algorithm

We utilize a greedy algorithm to perform a subgraph search as depicted in Figure 2. Following is a brief *pseudo-code* of our matching algorithm (please refer to Figure 2):

[1] We use the r-g-S color-space as in [12].

[2] In all our experiments we have used $\sigma_r = 5$ pixels and $\sigma_\theta = \pi/4$ radians.

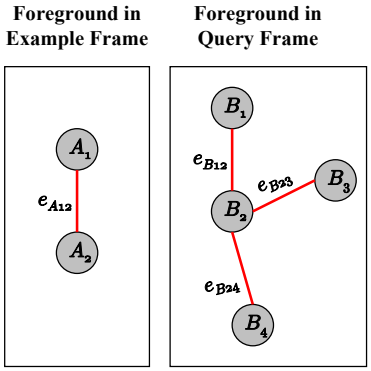


Fig. 2. *People matching viewed as a sub-graph matching problem.*

- For each vertex say A_1 in the example frame, compute the similarity $\mathfrak{S}(A_1 \rightarrow B_k)$ using Equation (1), to all the vertices B_k in the query frame, and retain the best M matches in a set of candidates $\mathcal{C}_{A_1} = \{B_k\}, k \in (1, 2, ..., N_B)$.
- Let the set of adjacent vertices of A_1 be denoted by $Adj(A_1)$. In Figure 2, $Adj(A_1) = A_2$. For all vertices B_k in $\mathcal{C}_{A_1}$, find $B_l \in Adj(B_k)$ such that $B_l \in \mathcal{C}_{A_2}$. Now, compute the *Global Evidence* for matching A_1 to B_k as:

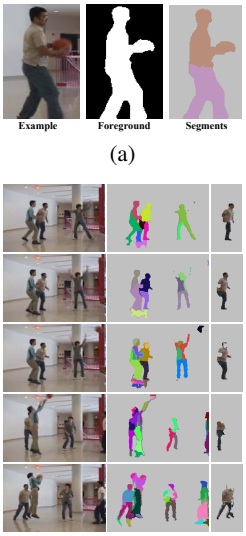
$$GE(A_1 \rightarrow B_k) = \mathfrak{S}(A_2 \rightarrow B_l)\Psi(e_{A_{12}} \rightarrow e_{B_{kl}}). \quad (5)$$

- Assign A_1 to that vertex $B_k \in \mathcal{C}_{A_1}$ which maximizes the *Global Evidence*. The similarity score for this vertex assignment is computed as $\mathbf{S}_{A_1} = \mathfrak{S}(A_1 \rightarrow B_k) + GE(A_1 \rightarrow B_k)$. Similarly compute the assignments for the remaining vertices in the example graph. The net similarity score for the matched subgraph is given by $\sum_{i=1}^{N_A} \mathbf{S}_{A_i}$.

4. RESULTS AND COMPARISON

Figures 3 and 4 illustrate the example based people matching algorithm explained above. In Figure 3 the example person is matched to the foreground in query frames acquired from the same camera location. It should be noted that the greedy algorithm is able to make the correct matches in spite of variations in pose and considerable occlusion. In Figure 4, the example person is compared to the foreground in a completely different camera view, leading to significant differences in scale and illumination. The results show that our representation and matching algorithm can robustly handle variations in illumination, pose, scale, and camera-view. We also perform a coarse comparison with the covariance distance approach used in [16].[3] We first compute the average similarity score $\mathbf{S}_{avg}$ and average covariance distance $\mathbf{D}_{avg}$ of the example in Figure 4(a) with respect to the same person in the top three rows in Figure 4(b). We then compute the similarity ($\mathbf{S}_{bad}$) and covariance distance ($\mathbf{D}_{bad}$) to a bad match (last row in Figure 4(b)). Now, we can approximately compare the discriminative power of the two measures by comparing

[3]It should be noted that the covariance tracker proposed in [16] does not separate foreground from background, which leads to less robust matching. Because of our particular foreground representation, we are able to correctly match the person along with giving exact segmentation for the matching region.



(a)

(b)

Fig. 3. *(a) Automatically detected foreground (center) from the example frame (left) is segmented (right) to get a 2 vertex graph representation. (b) The segmentation of the query frames (left column) is shown with random colors (center column) while the matching result is shown in the right column.*

the ratios $\mathcal{P}_{\mathbf{S}} = \frac{\mathbf{S}_{avg}}{\mathbf{S}_{bad}}$ and $\mathcal{P}_{\mathbf{D}} = \frac{\mathbf{D}_{bad}}{\mathbf{D}_{avg}}$.[4] We observe that $\mathcal{P}_{\mathbf{S}} \approx 5.0$, whereas $\mathcal{P}_{\mathbf{D}} \approx 1.5$ for the example in Figure 4, indicating that our similarity measure is more discriminative and hence allows for more robust matching. These results (including foreground detection and segmentation) were achieved at run-times of less than 1 second per query frame, using non-optimized experimental code, on a standard laptop computer with a 1.8GHz Centrino Processor. We used a *lite* version of the SIFT feature extraction, code provided by [17]. Color density estimation was performed using the Improved Fast Gauss Transform algorithm [13].

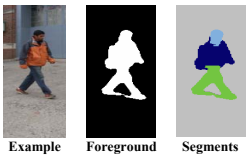
5. CONCLUSION AND FUTURE DIRECTIONS

We presented a novel scheme utilizing real-time foreground/background separation and low-level foreground segmentation to generate a graphical model of the appearance and relationship between objects (or object parts) in the foreground. The effectiveness of this representation was demonstrated with an example based query type of people matching algorithm, which along with its simple set-up, provides state-of-the-art results. We plan to further enhance the capability of our representation by improving the segments relationship model using more features and also incorporating information about temporal variations (temporal edges). Endeavors in these directions will be reported elsewhere.

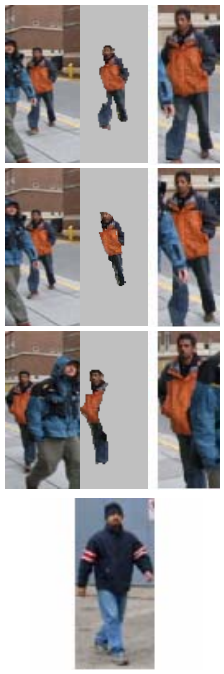
6. REFERENCES

[1] X. Ren and J. Malik, "Learning a classification model for segmentation," in *ICCV '03: Proceedings of the Ninth IEEE In-*

[4]Please note that we use a *similarity* measure, which is (conceptually) inversely related to the notion of *distance*. Hence the use of reciprocal ratios for $\mathcal{P}_{\mathbf{S}}$ and $\mathcal{P}_{\mathbf{D}}$.



(a)



(b)

Fig. 4. *(a) Foreground (center) from the example frame (left) is segmented (right) to get a* 3 *vertex graph representation. (b) Top* 3 *rows: Note that the example is compared with frames from a different camera location (left column), our matching results are shown in the center column. The column on the right is the rectangular window used to compute the covariance distance as used in [16]. Last row: We compute the similarity score and covariance distance between the example in (a) and last row of (b), in order to coarsely compare the discriminative power of our matching scheme with the covariance distance.*

ternational Conference on Computer Vision, Washington, DC, USA, 2003, p. 10, IEEE Computer Society.

[2] I. Haritaoglu, D. Harwood, and L. S. Davis, "Hydra: Multiple people detection and tracking using silhouettes," in *Proceedings of the Second IEEE Workshop on Visual Surveillance*, 1999, p. 6.

[3] S. McKenna, S. Jabri, Z. Duric, and A. Rosenfeld, "Tracking groups of people," *Computer Vision and Image Understanding*, vol. 80, pp. 42–56, 2000.

[4] S. Savarese, J. Winn, and A. Criminisi, "Discriminative object class models of appearance and shape by correlatons," in *CVPR '06*, 2006, pp. 2033–2040.

[5] C. Richard Wren, A. Azarbayejani, T. Darrell, and Alex Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 780–785, 1997.

[6] A. Elgammal and L. S. Davis, "Probabilistic framework for segmenting people under occlusion," in *In Proc. of IEEE 8th International Conference on Computer Vision*, 2001, pp. 145–152.

[7] J. Lim and D. Kriegman, "Tracking humans using prior and learned representations of shape and appearance," in *IEEE International Conference on Automatic Face and Gesture Recognition*, Los Alamitos, CA, USA, 2004, vol. 00, p. 869, IEEE Computer Society.

[8] L. Sigal, B. Sidharth, S. Roth, M. Black, and M. Isard, "Tracking loose-limbed people," in *IEEE Conf. On Computer Vision And Pattern Recognition*, 2004.

[9] X. Lan and D. P. Huttenlocher, "A unified spatio-temporal articulated model for tracking," in *IEEE Conf. On Computer Vision And Pattern Recognition*, 2004, vol. 01, pp. 722–729.

[10] D. Farin, P. de With, and W. Effelsberg, "Recognition of user-defined video object models using weighted graph homomorphisms," in *SPIE Image and Video Communications and Processing*, jan 2003.

[11] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[12] K. A. Patwardhan, G. Sapiro, and V. Morellas, "A pixel layering framework for robust foreground detection in video," Under Review, http://www.tc.umn.edu/∼patw0007/videolayers/index.html, June 2006.

[13] C. Yang, R. Duraiswami, N. Gumerov, and L. Davis, "Improved fast gauss transform and efficient kernel density estimation," in *Int'l Conf. Computer Vision*, 2003, pp. 464– 471.

[14] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. of the International Conference on Computer Vision*, 1999, pp. 1150–1157.

[15] T. Kailath, "The divergence and bhattacharyya distance measures in signal selection," *IEEE Transactions on Communications*, vol. 15, no. 1, pp. 52– 60, Feb 1967.

[16] F. Porikli, O. Tuzel, and P. Meer, "Covariance tracking using model update based on lie algebra," in *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006, pp. 728–735.

[17] A. Vedaldi, "Sift++ : `http://vision.ucla.edu/~vedaldi/code/siftpp/siftpp.html`," 2006.