Detection and tracking of humans and faces

Stefan Karlsson, Murtaza Taj, Andrea Cavallaro

Abstract-We present a video analysis framework that integrates prior knowledge in object tracking to automatically detect humans and faces, and can be used to generate abstract representations of video (key-objects and object trajectories). The analysis framework is based on the fusion of external knowledge, incorporated in a person and in a face classifier, and low-level features, clustered using temporal and spatial segmentation. Lowlevel features, namely color and motion, are used as a reliability measure for the classification. The results of the classification are then integrated into a multi-target tracker based on a particle filter that uses color histograms and a zero-order motion model. The tracker uses efficient initialization and termination rules and updates the object model over time. We evaluate the proposed framework on standard datasets in terms of precision and accuracy of the detection and tracking results, and demonstrate the benefits of the integration of prior knowledge in the tracking process.

I. INTRODUCTION

Video filtering and abstraction are of paramount importance in advanced surveillance and multimedia database retrieval. The knowledge of the objects' types and position helps semantic scene interpretation, indexing video events and mining large video collections. However, the annotation of a video in terms of its component objects is as good as the object detection and tracking algorithm that it is based upon. The quality of the detection and tracking algorithm depends in turn on its capability of localizing objects of interest (object categories) and on tracking them over time. It is in general difficult to define object categories for retrieval in video because of different meanings and definition of objects in different applications. However, some categories of objects, such as people and faces, are of interest across several applications and provide relevant cues about the content of a video. Detecting and tracking people and faces provides significant semantic information about the video content for video summarization, intelligent video surveillance, video indexing and retrieval. Moreover, the human visual system is particularly attracted by people and faces, and therefore their detection and tracking enables perceptual video coding [7].

A number of approaches has been proposed for the integration of object detectors in a tracking process. A stochastic model is implemented in [9] to track a single face in a video that relies on a combined face detection and prediction from the previous frame. Faces are detected in a coarseto-fine network, thus producing a hierarchical trace of face detections for each frame that is used in a trained probabilistic framework to determine face positions. Edgelet-based part

The authors acknowledge the support of the UK Engineering and Physical Sciences Research Council (EPSRC), under grant EP/D033772/1. S. Karlsson, M. Taj and A. Cavallaro are with the Multimedia and Vision Group, Queen Mary University of London, E-mail: {stefan.karlsson,murtaza.taj,andrea.cavallaro}@elec.qmul.ac.uk detector and mean shift can be used to perform detection and tracking of partially occluded objects [23]. The incorporation of recent observations improves the performance of a particle filter [18], and has been used in a hockey player tracking system by increasing the particles in the proposal distribution around detections [14]. As an alternative to an object detector, contour extraction can be combined with color information as part of the object model [24]. Other methods include motion segmentation combined with a nearest neighborhood filter [13], updating a Kalman filter with detections [22], combining detection and MAP probabilities [5], and using detections as input to a probabilistic data association filter [21].

In this paper we propose a unified multi-object detection and tracking framework that uses an object detection algorithm integrated with a particle filter, and demonstrate it on people and faces. The proposed framework integrates prior knowledge of object categories with probabilistic tracking. We use both *a priori* knowledge (in the form of training of an object classifier) and *on–line* knowledge acquisition (in the form of the target model update). Detection of faces and people is done by a cascaded Adaboost classifier, supported by color and motion segmentation, respectively. Next, a particle filter tracks the objects over time and compensates for missing or false detections. The detections, when available, influence the proposal distribution and the updating of the target color model (Fig. 1). We evaluate the proposed framework on the standard datasets CLEAR [11], AMI [2] and PETS 2001 [1].

The paper is organized as follows. Section II introduces face and people detection and evidence fusion. The integration of detections in particle filtering and track management issues are described in Section III. Section IV introduces the performance measures. Section V presents the experimental results. Finally, in Section VI we drawn the conclusions.

II. DETECTING HUMANS AND FACES

A. Classifying object categories

The a priori knowledge about object categories to be discovered in a video is incorporated through the training of an object detector. The validity of the proposed framework is independent of the chosen detector and here we use two different detectors to demonstrate the feasability and generality of the proposed framework.

In particular, to detect *faces* and *people* we use an Adaboost feature classifier based on a set of Haar-wavelet like features ([19], [20]). These features are computed on the integral image $\mathcal{I}(x,y)$, defined as $\mathcal{I}(x,y) = \sum_{i=1}^{x} \sum_{j=1}^{y} I(i,j)$, where I(i,j) represents the original image intensity. The Haar features are differences between sums of all pixels within sub-windows in the original image. Therefore, in the integral image they are calculated as simple differences between the

EURASIP JOURNAL ON IMAGE AND VIDEO PROCESSING



Fig. 1. Flow chart of the proposed object-based video analysis framework.



Fig. 2. Haar features used for classification. (a-e) Edge features; (f-g) center-surround features; (h-o) line features.

top-left and the bottom-right corners of the corresponding subwindows.

For *face* detection, we use a trained classifier ([4]) for frontal, left and right profile faces, with the 14 features shown in Fig. 2(a-d)(f-o). The edge feature shown in Fig. 2(e) is used to model tilted edges, such as shoulders, and it is therefore not suitable for modelling faces.

For *people* detection, the training was performed using the 13 features shown in Fig. 2(a-e) and Fig. 2(h-o) ([20]). We used $n_t = n_t^+ + n_t^- = 4285$ training samples, with $n_t^+ = 2543$ positive 10×24 pixel samples selected from the CLEAR dataset (Fig. 3) and $n_t^- = 1742$ negative samples with different resolutions. Since there is one weak classifier for each distinct feature combination, effectively there are $2543 \times 13 = 33059$ weak classifiers that, after training, are organized in 20 layers. Note that the features in Fig. 2(c,d,g)(l-o) are computed on the integral image rotated by 45° [12].

Let us denote the object classification result with $\hat{O}_t^c(x, y, w, h, n)$, where c denotes the object class (we will use the subscript f for faces and p for people), $n = 1, ..., N_c$ is the number of detected objects for class c at time t, (x, y) is the center of the object, w and h are its width and height, respectively.



Fig. 3. Subset of positive samples used for training the person detector.

B. Low-level segmentation

Low-level segmentation provides a reliability cue for each detection. We use skin color segmentation and motion segmentation to support face and person categorization, respectively.

Skin color segmentation is based on a non-linear transformation of the YC_bC_r color space [10], which results in a two-dimensional ad-hoc chromaticity plane $C'_bC'_r$. As this transformation is degenerate for gray pixels, RGB values respecting the conditions 0.975 < R/B and G/B < 1.025are discarded. To distinguish skin pixels in the $C'_bC'_r$ plane an ellipse encircling skin chromaticity is defined as

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1, (1)$$

with

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} C'_b - c_x \\ C'_r - c_y \end{bmatrix}.$$
 (2)

We sampled skin chromaticity from the CLEAR dataset and computed the values $c_x = 110$, $c_y = 152$, a = 25, b = 15 and $\theta = 2.53$, which are comparable to those in [10]. An example of skin color segmentation is shown in Fig. 4(d).



Fig. 4. Sample segmentation results on CLEAR test sequences. (a) Outdoor test sequence and (b) corresponding motion segmentation result. (c) Indoor test sequence and (d) corresponding color segmentation result.

Motion segmentation is performed using a statistical color change detector [6]. The detector assumes that a reference image is available, either because an image without objects can be taken or because of the use of an adaptive background algorithm [15], [17]. An example of motion segmentation results is presented in Fig. 4(b).

Let us denote the segmentation mask as $S_t^c(i, j)$, where i = 1, ..., W and j = 1, ..., H represent the pixel position, with W and H representing the image width and height, respectively.

C. Evidence fusion

Segmentation results are used to remove false positive detections. A detection $\hat{O}_t^c(x_d, y_d, w_d, h_d, n)$ is accepted if

$$\frac{|\hat{O}_{t}^{c}(x_{d}, y_{d}, w_{d}, h_{d}, n) \cap S_{t}^{c}(i, j)|}{|\hat{O}_{t}^{c}(x_{d}, y_{d}, w_{d}, h_{d}, n)|} > \lambda_{c},$$
(3)

where |.| is the cardinality of a set and λ_c is the minimum number of segmented pixels used to accept a detected area. For *color segmentation* $\lambda_f = 0.1$, whereas for *motion segmentation* $\lambda_p = 0.2$. The values of these thresholds depend on the fact that detections may contain background areas (for people) or hair regions (for faces). Fig. 5 shows two examples of detection results prior and after evidence fusion.

The resulting object detections are then used to initialize the object tracker as well as to solve track management issues, as discussed in the next section.

III. GENERATING TRAJECTORIES

A. The tracker

Tracking estimates the state of an object in subsequent frames. We use a particle filter tracker as it can deal with non-Gaussian multi-modal distributions [3], [14].

Let us represent the target state as $\mathbf{x}_t = [x, y, w, h]$. The posterior *pdf* of a target location in the state space is defined

Fig. 5. Sample person and face detection results. (a) Person detection using the classifier only; (b) filtered detections after evidence fusion. (c) Face detection using the classifier only; (d) filtered detections after evidence fusion.

as a sum of Dirac deltas centered around the particles, with weights ω_t^n :

$$p(\mathbf{x}_t | \mathbf{z}_{1:t}) \approx \sum_{n=1}^{N_s} \omega_t^n \delta(\mathbf{x}_t - \mathbf{x}_t^n),$$
(4)

where \mathbf{x}_t^n is the state of the n^{th} particle in frame t, $\mathbf{z}_{1:t}$ are the measurements from time 1 to t and N_s is the total number of particles. The state transition $p(\mathbf{x}_t^n | \mathbf{x}_{t-1}^n)$ is a zero-order motion model defined as $\mathbf{x}_t = \mathbf{x}_{t-1} + \mathcal{N}(\mathbf{x}_{t-1}, \sigma)$, where $\mathcal{N}(\mathbf{x}_{t-1}, \sigma)$ is a Gaussian noise centered in the previous state and with variance σ . The update of the *pdf* over time is based on the recalculation of the weights ω_t^n :

$$\omega_t^n \propto \omega_{t-1}^n \frac{p(\mathbf{z}_t | \mathbf{x}_t^n) p(\mathbf{x}_t^n | \mathbf{x}_{t-1}^n)}{q(\mathbf{x}_t^n | \mathbf{x}_{t-1}^n, \mathbf{z}_t)},$$
(5)

where $p(\mathbf{z}_t | \mathbf{x}_t^n)$ is the *likelihood* of the measurement. Since we use resampling to avoid the degeneracy of the particles (i.e., when the weights of all particles except one tend to zero after few iterations [3]), $\omega_{t-1}^n = 1/N \forall n$ and Eq. 5 simplifies to

$$\omega_t^n \propto \frac{p(\mathbf{z}_t | \mathbf{x}_t^n) p(\mathbf{x}_t^n | \mathbf{x}_{t-1}^n)}{q(\mathbf{x}_t^n | \mathbf{x}_{t-1}^n, \mathbf{z}_t)}.$$
(6)

To compute the likelihood $p(\mathbf{z}_t | \mathbf{x}_t^n)$, we use a color histogram $\phi^{\mathcal{M}} = [\varphi_{1,1,1}^{\mathcal{M}}, ..., \varphi_{RGB}^{\mathcal{M}}]$ as object model ([14], [24]), where R, G and B is the number of bins in each color channel. The color difference between the model \mathcal{M} and a particle $p, d_J(\phi^{\mathcal{M}}, \phi^p)$, is based on the Jeffrey divergence [16]. The likelihood is finally estimated as

$$p(\mathbf{z}_t | \mathbf{x}_t^n) = \frac{1}{\sqrt{2\pi\sigma_l}} e^{\frac{d_J(\phi^{\mathcal{M}, \phi^p})^2}{2\sigma_l^2}}.$$
 (7)

B. Particle propagation

Instead of using the transition prior only, we include object detections, when available, in the *proposal distribution*: a

fraction of the particles are spread around the previous state according to the motion model, whereas the rest are spread around the detections. For this reason, each detection has to be linked to the closest state. This *association* is established with a gated nearest neighborhood filter, which selects the detection $O_t^c(x_d, y_d, w_d, h_d, n)$ closest to the state \mathbf{x}_t , if it is in its proximity. The proximity conditions are

$$\begin{cases} |x_d - x_{tr}| < \delta_c (w_{tr} + \eta_c h_{tr}) \\ |y_d - y_{tr}| < \delta_c (\eta_c w_{tr} + h_{tr}) \\ (1 - \gamma_c) w_{tr} < w_d < (1 + \gamma_c) w_{tr} \\ (1 - \gamma_c) h_{tr} < h_d < (1 + \gamma_c) h_{tr} \end{cases}$$
(8)

where (x_{tr}, y_{tr}) is the center, w_{tr} and h_{tr} are the width and height of the ellipse representing the object; and $\eta_f = 1$, $\eta_p = 0$, $\delta_f = \gamma_p = 0.25$, $\delta_p = \gamma_f = 0.5$ are determined experimentally. The association is incorporated in Eq. 9 [14] as:

$$q(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{z}_t) = \alpha_c q_d(\mathbf{x}_t | \mathbf{z}_t) + (1 - \alpha_c) p(\mathbf{x}_t | \mathbf{x}_{t-1}), \quad (9)$$

where α_c is the fraction of particles spread around the detection in the state space and $q_d(\mathbf{x}_t|\mathbf{z}_t)$ is a Gaussian around the associated detection. If the proximity conditions are not satisfied, a new candidate track is initialized and $\alpha_c = 0$. In such a case, Eq. 9 reduces to $q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{z}_t) = p(\mathbf{x}_t|\mathbf{x}_{t-1})$, whereas Eq. 6 reduces to $\omega_t^n \propto p(\mathbf{z}_t|\mathbf{x}_t^n)$.

C. Model update

Object detections are also used to update online the object model \mathcal{M} . This update aims to avoid track drifting when the object appearance varies due to changes in illumination, size or pose. The color histogram is updated according to

$$\varphi_{r,g,b}^{\mathcal{M}}(t) = \beta_c \varphi_{r,g,b}^d(t) + (1 - \beta_c) \varphi_{r,g,b}^{\mathcal{M}}(t-1), \qquad (10)$$

where r = 1, ..., R; g = 1, ..., G; b = 1, ..., B and β_c is the update factor. Note that the histogram is only updated when there is an associated detection in order to prevent background pixels to become part of the model \mathcal{M} .

D. Track management issues

Unlike [14], where tracks are initiated with a single detection, we integrate information coming from the detector and the tracker processes to deal with track initiation and termination issues. A detection $O_t^c(x, y, w, h, n)$ that is not associated with a track is considered as a candidate for *track initialization*. Tracking is started in *sleeping mode*. To switch a track from *sleeping* to *active* mode, N_i detections are accumulated in subsequent frames. The value of N_i depends on frequency of the detections:

$$N_i = \min\left(\frac{3}{2 - \frac{1}{f}}f, 9\right),\tag{11}$$

where f is the frequency of detections and f = 9/20 is the minimum frequency. If there is not a sufficient number of successive detections, then the track is discarded.



Fig. 6. Example of use of track management rules for sequence S3, frame 270. (a) Without track management: the tracked ellipses degenerate. (b) With track management: the tracked ellipses correctly estimate the face areas.

A track is *terminated* if the low-level segmentation results do not provide enough evidence for the presence of an object:

$$\frac{|X_t^c(x_d, y_d, w_d, h_d, n) \cap S_t^c(i, j)|}{|\hat{X}_t^c(x_d, y_d, w_d, h_d, n)|} < \lambda_c,$$
(12)

with $\lambda_p = 0.2$ and $\lambda_f = 0.1$. Moreover, a person track is terminated if $N_t = 25$ subsequent frames without an associated detection. A face track is terminated when the color histogram of the object changes drastically, i.e., the Jeffrey divergence d_J between the current target and the model is larger than a threshold D. A cut-off distance of D = 0.15 was found appropriate. Also, we terminate the tracks that deviate more than 3σ from the average face size, learnt on the first 300 tracked faces. Finally, faces whose ratio is w/h > 1.5are considered unlikely and therefore removed. An example of performance improvements achieved with the proposed initialisation and termination rules is shown in Fig. 6.

E. Post-processing

Track verification is performed to remove false tracks in a post-processing stage. False tracks are generally initiated by repeated multiple detections on the same object. To remove these tracks, a score is computed for each overlapping track: $s_t^n = (0.6N_f)/50 + 0.4fr_d$, where s_t^n is the score for track n at time t, N_f is the number of frames tracked in a 50-frame window and fr_d is the frequency of detection. The weights on N_f (0.6) and fr_d (0.4) favor tracks with a long history against new ones with a high frequency. Finally, tracks shorter than 15 frames are likely to be clutter and therefore removed.

IV. PERFORMANCE MEASURES

To quantitatively evaluate the performance of the proposed framework, two groups of measures are used, namely *detection* and *tracking* performance measures. We chose as *detection* measures Precision, P, and Recall, R, which are designed to quantify the ability of an algorithm to identify true targets in a video, as opposed to false detections and missed detections. These measures are commonly used to evaluate the performance of database retrieval algorithms and are defined as

$$\begin{cases}
P = \frac{TP}{TP + FP} \\
R = \frac{TP}{TP + FN},
\end{cases}$$
(13)

TABLE I Short information about the datasets

Dataset	Seq.	Sequence name	Task	Frames
AMI	S1	EN2001b.Closeup1	face	100-600
	S2	EN2001b.Closeup4	face	1-500
	S3	IS1003c.L	face	1-500
	S4	IS1004a.R	face	250-750
CLEAR	S5	PVTRA102a09	people	500-3001
	S6	PVTRA102a10	people	3007-5701
	S7	PVTRA102a11	people	1-500
	S8	PVTRA102a12	people	1000-1500
PETS	S9	PETS1_SEG	people	1-500

where TP is the number of *true positives*, FP is the number of *false positives* and FN is the number of *false negatives*.

The *tracking* performance measures quantify the accuracy of the estimated object size (d_D) and the accuracy of the estimated object position (d_{Dist}) . The measure d_D quantifies the overlap between the ground truth and the estimated targets, and is defined as

$$d_{\mathcal{D}} = 1 - \frac{\sum_{n=1}^{N_{fn}} \sum_{t=1}^{N_{fr}} \left[\frac{2 \mid G_n^{(t)} \cap D_n^{(t)} \mid}{\mid G_n^{(t)} + D_n^{(t)} \mid} \right]}{\sum_{u=1}^{N_{fr}} N_{fn}^u}, \quad (14)$$

(1)

where $G_n^{(t)}$ denotes the ground truth for track n at time t; $D_n^{(t)}$ denotes the corresponding estimated target; N_{fn} the number of matched objects in the ground truth and the tracked objects in a frame; N_{fr} the total number of frames, and N_{fr}^u the total number of matched objects in the entire sequence. The measure d_{Dist} is the distance between the centers of the estimated tracked object and the ground truth, normalized by the size of the ground truth:

$$d_{\mathcal{D}ist} = \frac{\sum_{n=1}^{N_{fn}} \sum_{t=1}^{N_{fr}} \sqrt{\left(\frac{x_d - x_g}{w_g}\right)^2 + \left(\frac{y_d - y_g}{h_g}\right)^2}}{\sum_{u=1}^{N_{fr}} N_{fn}^u}, \quad (15)$$

where (x_d, y_d) and (x_g, y_g) are the centers of the tracked object and the ground truth, w_g and h_g are the width and the height of the corresponding ground truth object.

V. EXPERIMENTAL RESULTS

We demonstrate the proposed framework on three standard datasets, namely CLEAR, AMI and PETS 2001. These datasets include indoor and outdoor scenarios for a total of 8700 frames (Table I).

The same set of parameters is used for motion segmentation and for the tracker in all the experiments. For the statistical change detector, the noise variance is $\sigma = 1.8$ and the kernel size is k = 3. The particle filter uses 150 particles per object, with a transition factor of 12 pixels per frame. For the likelihood (Eq. 7), $\alpha_l = 0.068$. For faces, $\alpha_f = 0.9$, $\beta_f = 0.35$ and for people $\alpha_p = 0.25$, $\beta_p = 0.1$. There values have been found appropriate after extensive testing. The histogram for the color model and the likelihood is uniformly quantized with $10 \times 10 \times 10$ bins in the RGB space.

We compare the proposed approach that integrates detections and particle filtering (referred to as *PFI*) with the particle filtering alone (referred to as *PF*). To offer a fair



Fig. 7. Comparison of tracking results between *NN* (green) and *PFI* (blue). (a) Sequence S2 and (b) Sequence S4: the *NN* algorithm fails when there is low frequency of detections. (c) Sequence S6 and (d) Sequence S7: the *NN* filter produces jagged trajectories.



Fig. 8. Performance comparison of face tracks (sequence S3 and S2) and people tracks (sequence S5) for *PF* and *PFI*.

TABLE II Comparison of tracking performance (means and standard deviation for 8 runs).

		Faces			
Seq.		PFI	PF	NN	
S1	$\bar{d}_{\mathcal{D}}(\sigma_{d_{\mathcal{D}}})$	0.24(0.02)	0.25(0.03)	0.27	
	$d_{\mathcal{D}ist}(\sigma_{d_{\mathcal{D}ist}})$	0.10(0.004)	0.14(0.02)	0.10	
	$\bar{P}(\sigma_{\bar{P}})$	0.76(0.06)	0.70(0.03)	0.70	
	$\bar{R}(\sigma_{\bar{R}})$	1.00(0)	0.98(0.03)	1	
S2	$\bar{d}_{\mathcal{D}}(\sigma_{d_{\mathcal{D}}})$	0.28(0.01)	0.34(0.01)	0.28	
	$d_{\mathcal{D}ist}(\sigma_{d_{\mathcal{D}ist}})$	0.13(0.005)	0.24(0.01)	0.12	
	$\bar{P}(\sigma_{\bar{P}})$	0.95(0.01)	0.92(0.03)	0.94	
	$\bar{R}(\sigma_{\bar{R}})$	0.96(0.01)	0.89(0.01)	0.94	
S3	$\overline{d}_{\mathcal{D}}(\sigma_{d_{\mathcal{D}}})$	0.27(0.03)	0.39(0.01)	0.32	
	$d_{\mathcal{D}ist}(\sigma_{d_{\mathcal{D}ist}})$	0.13(0.02)	0.21(0.02)	0.16	
	$\bar{P}(\sigma_{\bar{P}})$	0.52(0.03)	0.38(0.02)	0.47	
	$\bar{R}(\sigma_{\bar{R}})$	0.73(0.03)	0.74(0.02)	0.72	
S4	$d_{\mathcal{D}}(\sigma_{d_{\mathcal{D}}})$	0.38(0.03)	0.49(0.03)	0.26	
	$d_{\mathcal{D}ist}(\sigma_{d_{\mathcal{D}ist}})$	0.26(0.03)	0.41(0.04)	0.17	
	$\bar{P}(\sigma_{\bar{P}})$	0.66(0.08)	0.52(0.04)	0.60	
	$\bar{R}(\sigma_{\bar{R}})$	0.69(0.06)	0.48(0.05)	0.29	
		People			
Seq.		PFI	PF	NN	
S5	$\bar{d}_{\mathcal{D}}(\sigma_{d_{\mathcal{D}}})$	0.25(0.02)	0.26(0.01)	0.24	
	$d_{\mathcal{D}ist}(\sigma_{d_{\mathcal{D}ist}})$	0.18(0.01)	0.17(0.02)	0.19	
	$\bar{P}(\sigma_{\bar{P}})$	0.78(0.02)	0.78(0.01)	0.80	
	$\bar{R}(\sigma_{\bar{R}})$	0.90(0.02)	0.92(0.03)	0.82	
S6	$d_{\mathcal{D}}(\sigma_{d_{\mathcal{D}}})$	0.25(0.05)	0.35(0.03)	0.21	
	$d_{\mathcal{D}ist}(\sigma_{d_{\mathcal{D}ist}})$	0.16(0.03)	0.22(0.02)	0.13	
	$\bar{P}(\sigma_{\bar{P}})$	0.23(0.04)	0.22(0)	0.26	
	$R(\sigma_{\bar{R}})$	0.55(0.08)	0.59(0.11)	0.62	
S7	$\bar{d}_{\mathcal{D}}(\sigma_{d_{\mathcal{D}}})$	0.36(0.04)	0.36(0.01)	0.31	
	$d_{\mathcal{D}ist}(\sigma_{d_{\mathcal{D}ist}})$	0.21(0.02)	0.24(0.02)	0.17	
	$\bar{P}(\sigma_{\bar{P}})$	0.74(0.04)	0.70(0.02)	0.81	
	$\bar{R}(\sigma_{\bar{R}})$	0.84(0.01)	0.84(0.01)	0.84	
S8	$d_{\mathcal{D}}(\sigma_{d_{\mathcal{D}}})$	0.34(0.03)	0.37(0.04)	0.35	
	$\bar{d_{\mathcal{D}ist}(\sigma_{d_{\mathcal{D}ist}})}$	0.21(0.02)	0.21(0.03)	0.21	
	$\bar{P}(\sigma_{\bar{P}})$	0.59(0.02)	0.57(0.03)	0.60	
	$\bar{R}(\sigma_{\bar{P}})$	0.67(0.02)	0.65(0.04)	0.61	

comparison, in both cases the initialization and termination rules presented in Section III-D are used. We also compare *PFI* with the nearest neighborhood filter (*NN*). The measurements used for evaluation are the mean $(\bar{d}_D, \bar{d}_{Dist}, \bar{R} \text{ and } \bar{P})$ and the corresponding standard deviations on 8 runs of the performance measures presented in Section IV (see Table II).

The comparison of *PFI* and *PF* for faces shows the \bar{d}_{D} and \bar{d}_{Dist} scores are smaller for all face sequences indicating *better* correspondence between track ellipses and the ground truth. Further, \bar{R} and \bar{P} are larger for the same sequences, except for one \bar{R} score. Fig. 9 shows sample results of people and face tracking and their frame-wise \bar{d}_{D} scores are illustrated in Fig. 8. In Fig. 8, row 1, the quality of *PFI* results improve more quickly than those of *PF*. The average \bar{d}_{D} for *PFI* is 0.17 and for *PF* 0.33, and in Fig. 8, row 2, the average for *PFI* is 0.24 and the average for *PF* 0.31. Fig. 8, row 3 and row 4, are human tracking examples with average for *PFI* 0.22 and 0.12 and with average for *PF* 0.30 and 0.15, respectively. The lower average values of \bar{d}_{D} in all these cases shows improved performance of *PFI* over *PF*.

The comparison between *PFI* and *NN* for faces shows that the $d_{\mathcal{D}}$ and $d_{\mathcal{D}ist}$ scores are better for sequence S1 and S3, similar for sequence S2, whereas these scores indicate better



Fig. 9. Comparison of tracking result with *PF* (green) and *PFI* (blue). (a-b) Sequence S5. (c) Sequence S2. (d) Sequence S3.



Fig. 10. Example of trajectory-based video description and object prototypes. (a) Resulting tracks superimposed on the images. (b) Evolution of the tracks over time. (c) Automatically generated key-objects for frontal, left and right profile faces.

performance of the NN tracker for S4, but with lower R and \overline{P} scores. The reason is that in S4 the NN tracker fails to track in parts of the sequence with very low frequency of detections, whereas the particle filter succeeds in tracking in these regions (Fig. 7). For people tracking, the scores are similar for S5 and S8, whereas NN is better for S6 and S7 because sometimes detections that are larger than the person dominate in frequency, and *PFI* will filter out the correctly sized detections (which are instead taken into account by NN).

To conclude, Fig. 10 shows an example of trajectorybased video description using spatio-temporal object trajectories of two faces and the corresponding object prototypes (frontal and profile faces). Only the true tracks are computed by the proposed algorithm and false detections and associated tracks are filtered out using skin color segmentation and post–processing. Videos results are available at http://www.elec.qmul.ac.uk/staffinfo/andrea/detrack.html.

VI. CONCLUSIONS

We presented a general video analysis framework for detecting and tracking object categories and demonstrated it on people and faces. Video results and quantitative measurements show that the proposed integration of detections with particle filtering improves the robustness of the state estimation of the targets.

The proposed framework is general and classifiers of other body parts and other object types can be incorporated without changing the overall structure of the algorithm. Using additional object detectors, a complete story line of a video based on specific object categories and their trajectories could be produced, describing interactions and other important events. Moreover, the video could be annotated semantically with identity information of the appearing persons by adding a face recognition module [8].

Our current work includes improving the performance of the human detector by using a larger training database and refining the bounding boxes of the detection using edges and motion segmentation results.

REFERENCES

- http://www.cvg.cs.rdg.ac.uk/pets2001/pets2001-dataset.html. Last accessed on 12 July 2007.
- [2] http://www.idiap.ch/amicorpus. Last accessed on 12 July 2007.
- [3] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, February 2002.
- [4] G. Bradski, A. Kaehler, and V. Pisarevsky. Learning-based computer vision with Intel's open source computer vision library. *Intel Technology Journal*, 9:119–130, May 2005.
- [5] M. G. S. Bruno and J. M. F. Moura. Integration of Bayes detection and target tracking in real clutter image sequences. In *IEEE National Radar Conference*, pages 234–238, Atlanta GA, USA, May 2001.
- [6] A. Cavallaro and T. Ebrahimi. Interaction between high-level and lowlevel image analysis for semantic video object extraction. *EURASIP Journal on Applied Signal Processing*, 6:786–797, June 2004.
- [7] A. Cavallaro and S. Winkler. Perceptual semantics in Digital multimedia perception and design (Editors: G. Ghinea and S. Y. Chen). Idea Group Publishing, April 2006. ISBN 159140861-X.
- [8] J. Ruiz del Solar and P. Navarrete. Eigenspace-based face recognition: a comparative study of different approaches. *Transactions on Systems, Man and Cybernetics, Part C*, pages 315–325, August 2005.

- [9] S. Gangaputra and D. Geman. A unified stochastic model for detecting and tracking faces. In *The 2nd Canadian Conference on Computer and Robot Vision 2005*, pages 306–313, Victoria, BC, Canada, May 2005.
- [10] R.-L. Hsu, M. Abdel-Mottaleb, and A. K. Jain. Face detection in color images. *IEEE Transaction on Pattern Analysis Machine Intelligence*, 24:1046–1049, May 2002.
- [11] R. Kasturi. Performance evaluation protocol for face, person and vehicle detection & tracking in video analysis and content extraction (VACE-II). Computer Science & Engineering University of South Florida, Tampa FL, USA, January 2006. http://isl.ira.uka.de/clear06/downloads/ClearEval_Protocol_v5.pdf. Last accessed on 16 October 2007.
- [12] R. Lienhart and J. Maydt. An extended set of Haar-like features for rapid object detection. In *Proceedings of the International Conference* on *Image Processing*, pages 900–903, Rochester, USA, September 2002.
- [13] S. McKenna and S. Gong. Tracking faces. In Second International Conference on Automated Face and Gesture Recognition, pages 271– 276, Killington, Vermont, October 1996.
- [14] K. Okuma, A. Taleghani, N. de Freitas, J. J. Little, and D. G. Lowe. A boosted particle filter: Multitarget detection and tracking. In *European Conference on Computer Vision*, volume 1, pages 28–39, Prague, Czech Republic, 2004.
- [15] R. J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam. Image change detection algorithms: a systematic survey. *IEEE Transactions on Image Processing*, 14:294–307, March 2005.
- [16] Y. Rubner, J. Puzicha, C. Tomasi, and J. M. Buhmann. Empirical evaluation of dissimilarity measures for color and texture. *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, 84:25– 43, October 2001.
- [17] C. Stauffer and W.E.L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transaction on Pattern Analysis Machine Intelligence*, 22:747–757, August 2000.
- [18] R. van der Merwe, A Doucet, J. F. G. de Freitas, and E. Wan. The unscented particle filter. In Advances in Neural Information Processing Systems, volume 8, pages 351–357, November 2001.
- [19] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In Proc. of IEEE Conference on Computer Vision and Pattern Recognition, pages 511–518, Kauai, Hawaii, December 2001.
- [20] P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *Proc. of IEEE International Conference* on Computer Vision, volume 2, pages 734–741, Nice, France, October 2003.
- [21] P. Willett, R. Niu, and Y. Bar-Shalom. Integration of Bayes detection with target tracking. In *IEEE Transactions on Signal Processing*, volume 49, pages 17–29, January 2001.
- [22] P. Withagen, K. Schutte, and F. Groen. Object detection and tracking using a likelihood based approach. In *Advanced School for Computing and Imaging Conference*, volume 2, pages 248–253, Lochem, Netherlands, June 2002.
- [23] B. Wu and R. Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *International Journal on Computer Vision*, January 2007.
- [24] X. Xu and B. Li. Head tracking using particle filter with intensity gradient and color histogram. In *IEEE International Conference on Multimedia and Expo*, pages 888–891, Amsterdam, The Netherlands, July 2005.