Hindawi Publishing Corporation
EURASIP Journal on Image and Video Processing
Volume 2008, Article ID 526191, 9 pages
doi:10.1155/2008/526191

Research Article

Detection and Tracking of Humans and Faces

Stefan Karlsson, Murtaza Taj, and Andrea Cavallaro

Multimedia and Vision Group, Queen Mary University of London, London E1 4NS, UK

Correspondence should be addressed to Murtaza Taj, murtaza.taj@elec.qmul.ac.uk

Received 15 February 2007; Revised 14 July 2007; Accepted 25 November 2007

Recommended by Maja Pantic

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. Low-level features, namely, color and motion, are used as a reliability measure for the classification. The results of the classification are then integrated into a multitarget 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.

1. 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 in 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 definitions 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 provide 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 enable perceptual video coding [1].

A number of approaches have been proposed for the integration of object detectors in a tracking process. A stochastic model is implemented in [2] to track a single face in a video, which relies on combined face detection and prediction from the previous frame. Faces are detected in a coarse-to-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 detector and mean shift can be used to perform detection and tracking of partially occluded objects [3]. The incorporation of recent observations improves the performance of a particle filter [4], and has been used in a hockey player tracking system by increasing the particles in the proposal distribution around detections [5]. As an alternative to an object detector, contour extraction can be combined with color information as part of the object model [6]. Other methods include motion segmentation combined with a nearest neighborhood filter [7], updating a Kalman filter with detections [8], combining detection and MAP probabilities [9], and using detections as input to a probabilistic data association filter [10].

In this paper, we propose a unified multiobject detection and tracking framework that uses an object detection algorithm integrated with a particle filter and demonstrates 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

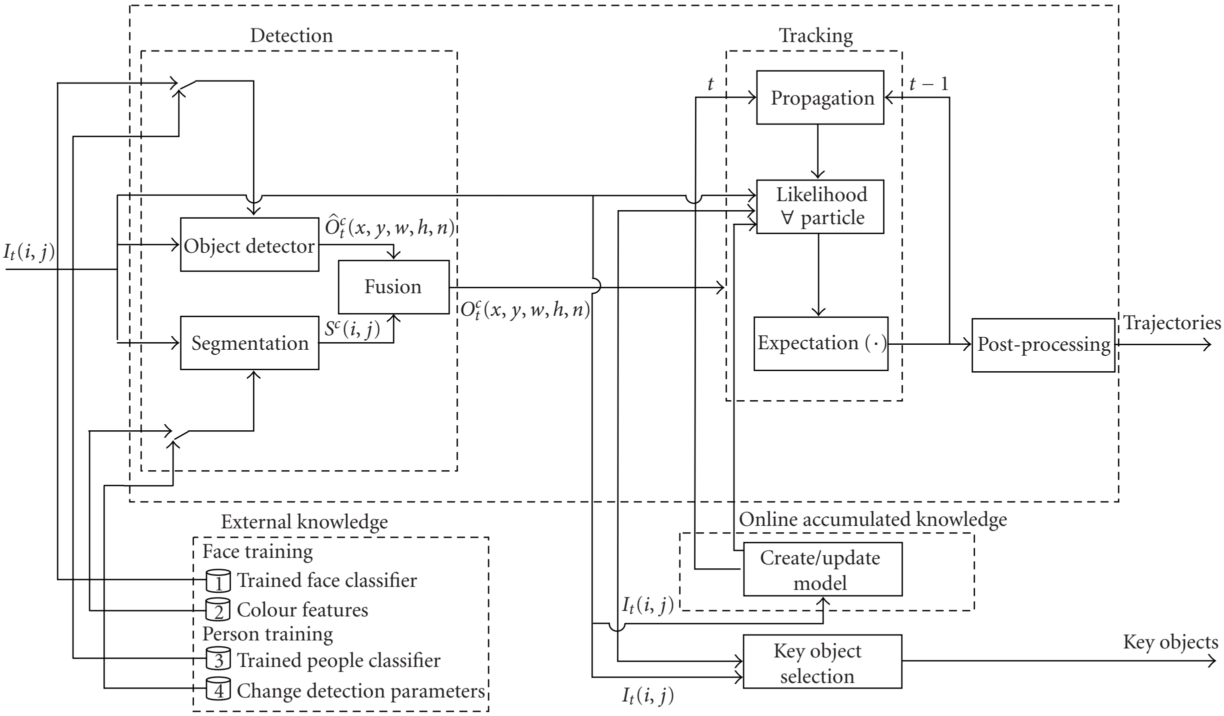


Figure 1: Flow chart of the proposed object-based video analysis framework.

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 (see Figure 1). We evaluate the proposed framework on the standard datasets CLEAR [11], AMI [12], and PETS 2001 [13].

The paper is organized as follows. Section 2 introduces face and people detection and evidence fusion. The integration of detections in particle filtering and track management issues are described in Section 3. Section 4 introduces the performance measures. Section 5 presents the experimental results. Finally, in Section 6 we draw the conclusions.

2. DETECTING HUMANS AND FACES

2.1. 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 feasibility 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 (see [14, 15]). 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 subwindows in the original image. Therefore, in the integral image, they are calculated as simple differences between the top-left and the bottom-right corners of the corresponding subwindows.

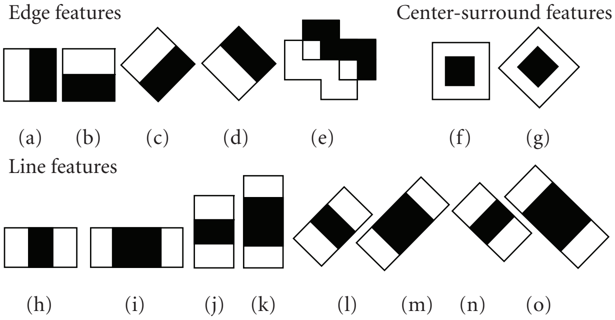


Figure 2: Haar features used for classification. (a–e) edge features; (f-g) center-surround features; (h–o) line features.

For *face* detection, we use a trained classifier [16] for frontal, left, and right profile faces, with the 14 features shown in Figure 2 (see ((a)–(d), (f)–(o))). The edge feature shown in Figure 2(e) is used to model tilted edges, such as shoulders, and it is therefore not suitable for modeling faces.

For *people* detection, the training was performed using the 13 features shown in Figure 2 (see ((a)–(e), (h)–(o))) [15]. 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 (see Figure 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 Figure 2 (see

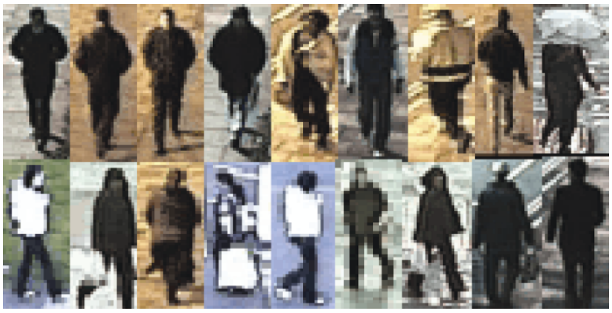


Figure 3: Subset of positive samples used for training the person detector.

((c), (d), (g), (l)–(o))) are computed on the integral image rotated by 45° [17].

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), $n = 1, \ldots, N_c$ is the number of detected objects for class c at time t, (x, y) is the center of the object, and w and h are its width and height, respectively.

2.2. *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 nonlinear transformation of the YC_bC_r color space [18], which results in a two-dimensional ad hoc chromaticity plane $C_b'C_r'$. As this transformation is degenerate for gray pixels, RGB values with respect to the conditions $0.975 < R/B$ and $G/B < 1.025$ are discarded. To distinguish skin pixels in the $C_b'C_r'$ plane, an ellipse encircling skin chromaticity is defined as

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

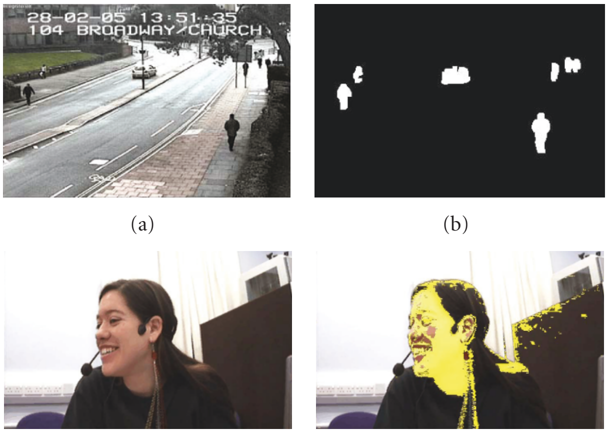
with

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} C_b' - c_x \\ C_r' - c_y \end{bmatrix}. \qquad (2)$$

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 [18]. An example of skin color segmentation is shown in Figure 4(d).

Motion segmentation is performed using a statistical color change detector [19]. 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 [20, 21]. An example of motion segmentation results is presented in Figure 4(b).

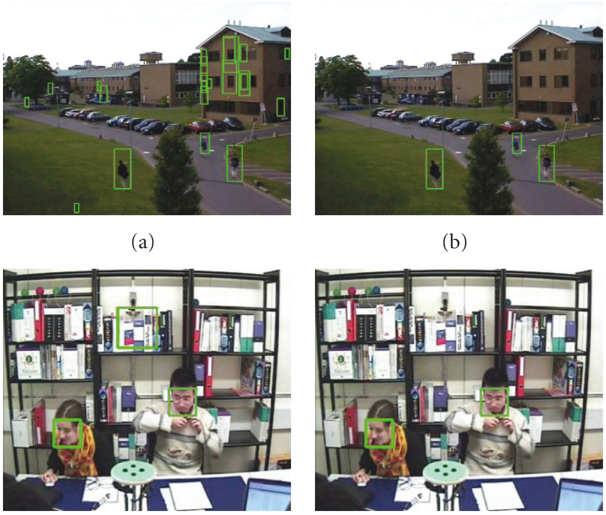
Let us denote the segmentation mask as $S_t^c(i, j)$, where $i = 1, \ldots, W$ and $j = 1, \ldots, H$ represent the pixel position, with W and H representing the image width and height, respectively.



(a) (b)

(c) (d)

Figure 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.



(a) (b)

(c) (d)

Figure 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.

2.3. *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

$$\frac{\left|\hat{O}_t^c(x_d, y_d, w_d, h_d, n) \cap S_t^c(i, j)\right|}{\left|\hat{O}_t^c(x_d, y_d, w_d, h_d, n)\right|} > \lambda_c, \qquad (3)$$

where $|\cdot|$ 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). Figure 5 shows two examples of detection results prior to 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.

3. GENERATING TRAJECTORIES

3.1. 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 multimodal distributions [5, 22].

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

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

where $\mathbf{x}_t^n$ is the state of the nth particle in frame t, $\mathbf{z}_{1:t}$ are the measurements from time 1 to time t, and N_s is the total number of particles. The state transition $p(\mathbf{x}_t^n \mid \mathbf{x}_{t-1}^n)$ 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 with variance σ. The update of the *pdf* over time is based on the recalculation of the weights ω_t^n:

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

where $p(\mathbf{z}_t \mid \mathbf{x}_t^n)$ 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 [22]), $\omega_{t-1}^n = 1/N\ \forall n$ and (5) is simplified to

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

To compute the likelihood $p(\mathbf{z}_t \mid x_t^n)$, we use a color histogram $\phi^{\mathcal{M}} = [\varphi_{1,1,1}^{\mathcal{M}}, \ldots, \varphi_{\mathrm{RGB}}^{\mathcal{M}}]$ as object model [5, 6], where R, G, and B are 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 [23]. The likelihood is finally estimated as

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

3.2. Particle propagation

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

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

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 (9) [5] as

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

where α_c is the fraction of particles spread around the detection in the state space and $q_d(\mathbf{x}_t \mid \mathbf{z}_t)$ 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, (9) reduces to $q(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{z}_t) = p(\mathbf{x}_t \mid \mathbf{x}_{t-1})$, whereas (6) reduces to $\omega_t^n \propto p(\mathbf{z}_t \mid \mathbf{x}_t^n)$.

3.3. Model update

Object detections are also used to online update 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

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

where $r = 1, \ldots, R$, $g = 1, \ldots, G$, $b = 1, \ldots, B$, 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 from becoming a part of the model $\mathcal{M}$.

3.4. Track management issues

Unlike [5], 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:

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

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

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

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

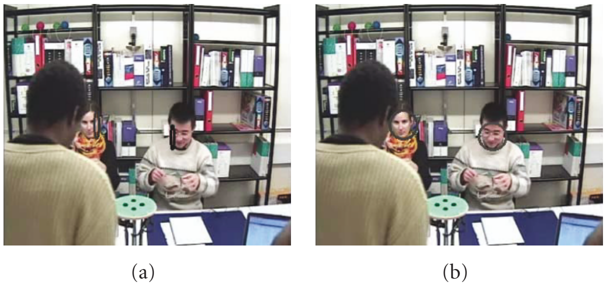

(a) (b)

Figure 6: Example of using 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.

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; that is, 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 $w/h > 1.5$ are considered unlikely and therefore removed. An example of performance improvements achieved with the proposed initialization and termination rules is shown in Figure 6.

3.5. *Postprocessing*

Track verification is performed to remove false tracks in a postprocessing 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: $s_t^n = (0.6N_f)/50 + 0.4\mathrm{fr}_d$, 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 cluttered and therefore removed.

4. 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

$$\begin{aligned} P &= \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}, \\ R &= \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}, \end{aligned} \tag{13}$$

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

Table 1: Brief 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	PETS1SEG	people	1–500

The *tracking* performance measures quantify the accuracy of the estimated object size ($d_{\mathfrak{D}}$) and the accuracy of the estimated object position ($d_{\mathfrak{D}ist}$). The measure $d_{\mathfrak{D}}$ quantifies the overlap between the ground truth and the estimated targets, and it is defined as

$$d_{\mathfrak{D}} = 1 - \frac{\sum_{n=1}^{N_{\mathrm{fn}}}\sum_{t=1}^{N_{\mathrm{fr}}}\left[2\left|G_n^{(t)} \cap D_n^{(t)}\right| / \left|G_n^{(t)} + D_n^{(t)}\right|\right]}{\sum_{u=1}^{N_{\mathrm{fr}}} N_{\mathrm{fn}}^u}, \tag{14}$$

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

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

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

5. 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 (see Table 1).

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 (7), $\alpha_l = 0.068$. For faces, $\alpha_f = 0.9$ and $\beta_f = 0.35$, and for people, $\alpha_p = 0.25$ and $\beta_p = 0.1$. These values have been found appropriate after extensive testing. The histogram for the color model and the likelihood is uniformly quantized with 10 × 10 × 10 bins in the RGB space.

We compare the proposed approach that integrates detections and particle filtering (referred to as PFI) with the

Table 2: Comparison of tracking performance (means and standard deviations for 8 runs).

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

particle filtering alone (referred to as PF). To offer a fair comparison, in both cases the initialization and termination rules presented in Section 3.4 are used. We also compare PFI with the nearest neighborhood filter (NN). The measurements used for evaluation are the mean ($\overline{d_{\mathcal{D}}}$, $\overline{d_{\mathcal{D}ist}}$, $\overline{R}$, and $\overline{P}$) and the corresponding standard deviations on 8 runs of the performance measures presented in Section 4 (see Table 2).

The comparison of PFI and PF for faces shows that $\overline{d_{\mathcal{D}}}$ and $\overline{d_{\mathcal{D}ist}}$ scores are smaller for all face sequences indicating *better* correspondence between track ellipses and the ground truth. Further, $\overline{R}$ and $\overline{P}$ are larger for the same sequences, except for one $\overline{R}$ score. Figure 9 shows sample results of people and face tracking, and their framewise $\overline{d_{\mathcal{D}}}$ scores are illustrated in Figure 8. In Figure 8 (row 1), the quality of PFI

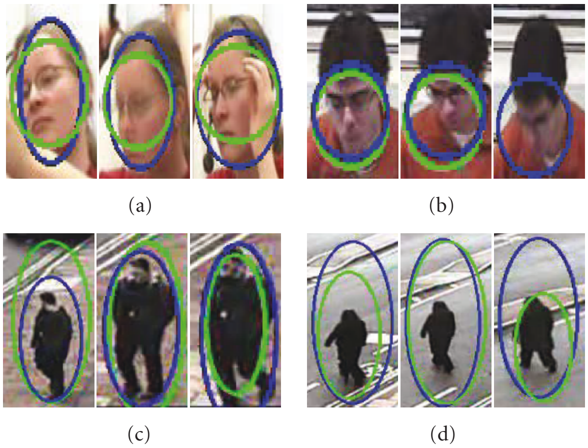


Figure 7: Comparison of tracking results between NN (green) and PFI (blue). (a) Sequence S2 and (b) Sequence S4: the NN algorithm fails when there is low frequency of detections. (c) Sequence S6 and (d) Sequence S7: the NN filter produces jagged trajectories.

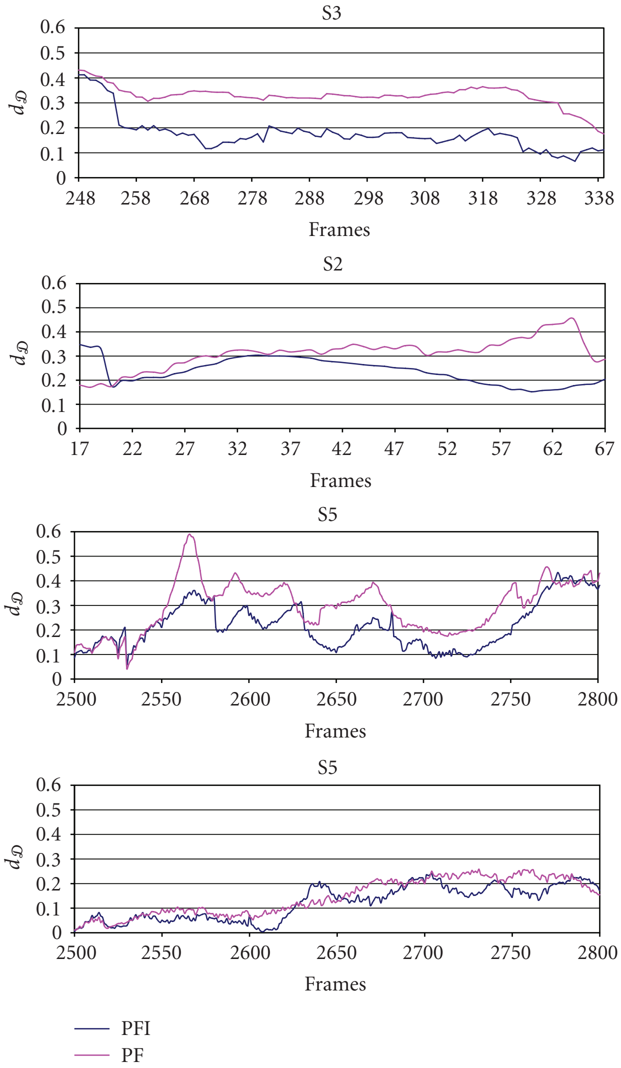


Figure 8: Performance comparison of face tracks (sequence S3 and S2) and people tracks (sequence S5) for PF and PFI.

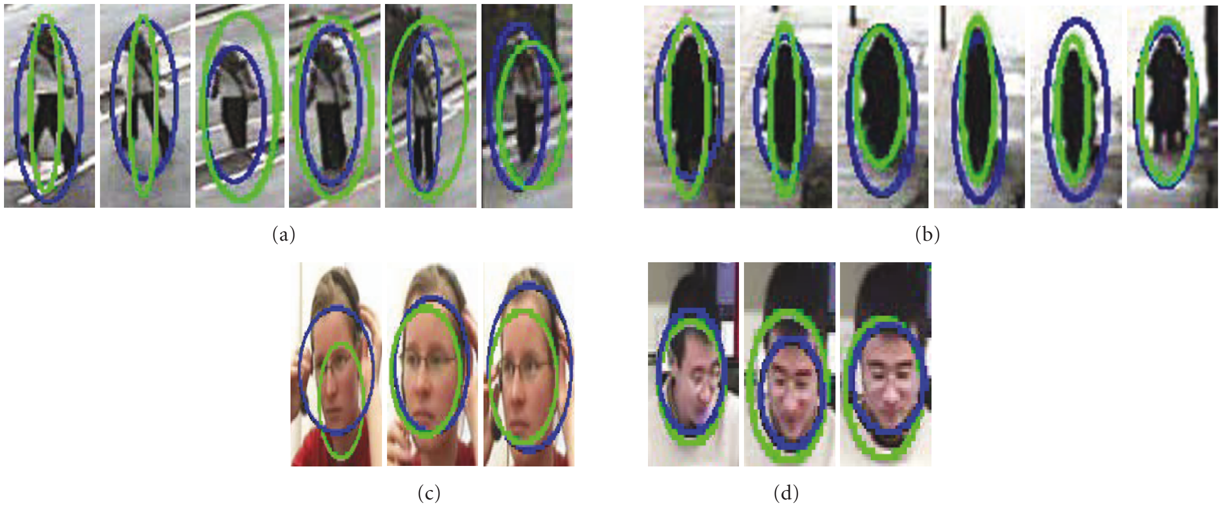


Figure 9: Comparison of tracking results with PF (green) and PFI (blue). (a)–(b) Sequence S5; (c) sequence S2; (d) sequence S3.

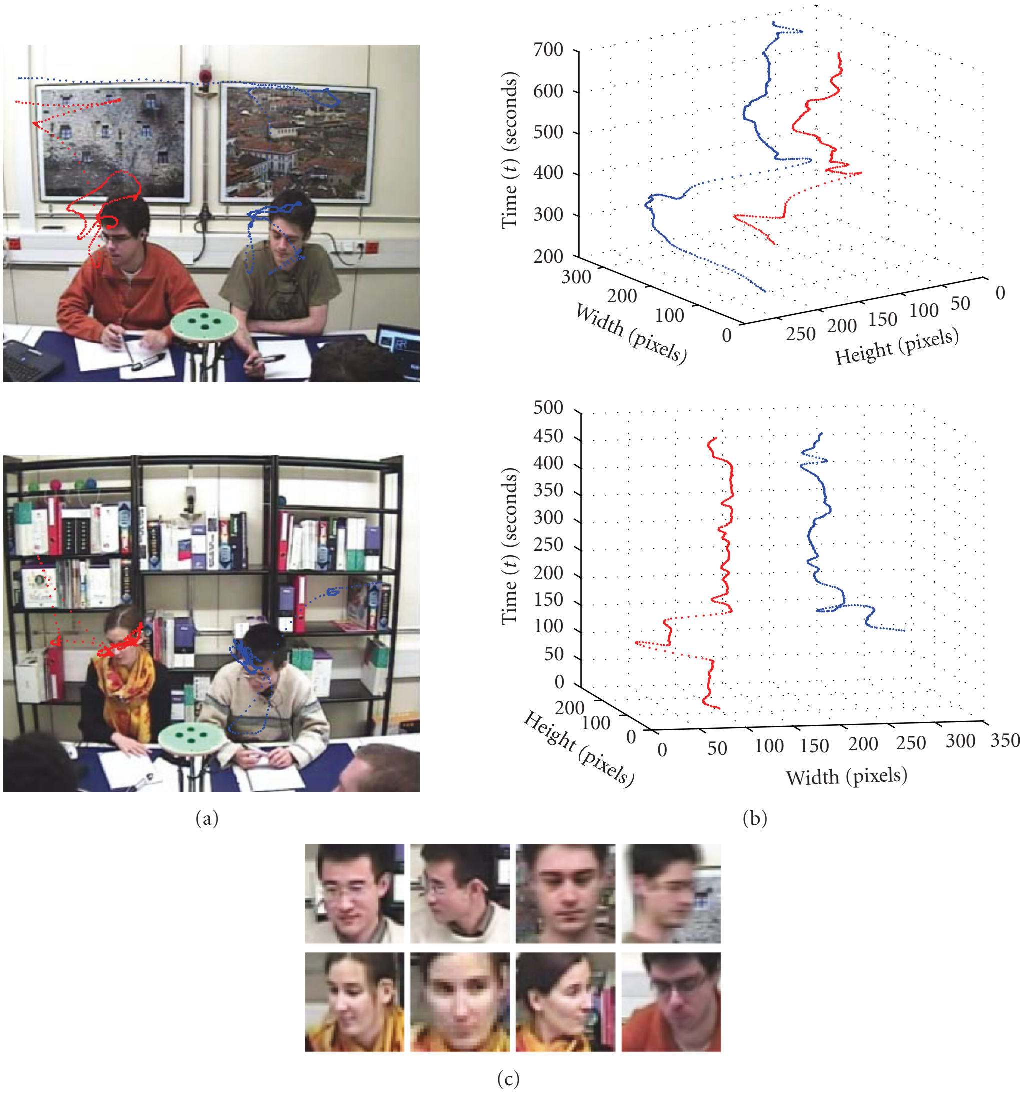


Figure 10: Example of trajectory-based video description and object prototypes. (a) Resulting tracks superimposed on the images. (b) Evolution of the tracks over time. (c) Automatically generated key-objects for frontal, left, and right profile faces.

results improves more quickly than those of PF. The average $\overline{d_{\mathcal{D}}}$ for PFI is 0.17 and for PF is 0.33, and in Figure 8 (row 2), the average for PFI is 0.24 and the average for PF is 0.31. In Figure 8, rows 3 and 4, are the human tracking examples with average of 0.22 and 0.12 for PFI and average of 0.30 and 0.15 for PF, respectively. The lower average values of $\overline{d_{\mathcal{D}}}$ in all these cases show improved performance of PFI over PF.

The comparison between PFI and NN for faces shows that the $\overline{d_{\mathcal{D}}}$ and $\overline{d_{\mathcal{D}ist}}$ scores are better for sequences S1 and S3, similar for sequence S2, whereas these scores indicate better performance of the NN tracker for S4, but with lower $\overline{R}$ 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 (see Figure 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, Figure 10 shows an example of trajectory-based video description using spatiotemporal 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 postprocessing. Videos results are available at http://www.elec.qmul.ac.uk/staffinfo/andrea/detrack.html.

6. 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 [24].

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.

ACKNOWLEDGMENT

The authors acknowledge the support of the UK Engineering and Physical Sciences Research Council (EPSRC), under Grant no. EP/D033772/1.

REFERENCES

[1] A. Cavallaro and S. Winkler, "Perceptual semantics," in *Digital Multimedia Perception and Design*, G. Ghinea and S. Y. Chen, Eds., Idea Group, Toronto, Canada, April 2006.

[2] S. Gangaputra and D. Geman, "A unified stochastic model for detecting and tracking faces," in *Proceedings of the 2nd Canadian Conference on Computer and Robot Vision*, pp. 306–313, Victoria, BC, Canada, May 2005.

[3] B. Wu and R. Nevatia, "Detection and tracking of multiple, partially occluded humans by Bayesian combination of edgelet based part detectors," *International Journal of Computer Vision*, vol. 75, no. 2, pp. 247–266, 2007.

[4] 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 14 (NIPS '01)*, vol. 8, pp. 351–357, Vancouver, BC, Canada, December 2001.

[5] K. Okuma, A. Taleghani, N. de Freitas, J. J. Little, and D. G. Lowe, "A boosted particle filter: multitarget detection and tracking," in *Proceedings of the 8th European Conference on Computer Vision (ECCV '04)*, vol. 1, pp. 28–39, Prague, Czech Republic, May 2004.

[6] X. Xu and B. Li, "Head tracking using particle filter with intensity gradient and color histogram," in *Proceedings of IEEE International Conference on Multimedia and Expo (ICME '05)*, vol. 2005, pp. 888–891, Amsterdam, The Netherlands, July 2005.

[7] S. McKenna and S. Gong, "Tracking faces," in *Proceedings of the 2nd International Conference on Automatic Face and Gesture Recognition*, pp. 271–276, Killington, VT, USA, October 1996.

[8] P. Withagen, K. Schutte, and F. Groen, "Object detection and tracking using a likelihood based approach," in *Proceedings of the Advanced School for Computing and Imaging Conference*, vol. 2, pp. 248–253, Lochem, Netherlands, June 2002.

[9] M. G. S. Bruno and J. M. F. Moura, "Integration of Bayes detection and target tracking in real clutter image sequences," in *Proceedings of IEEE International Radar Conference*, pp. 234–238, Atlanta, GA, USA, May 2001.

[10] P. Willett, R. Niu, and Y. Bar-Shalom, "Integration of Bayes detection with target tracking," *IEEE Transactions on Signal Processing*, vol. 49, no. 1, pp. 17–29, 2001.

[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.

[12] http://www.idiap.ch/amicorpus, July 2007.

[13] http://www.cvg.cs.rdg.ac.uk/pets2001/pets2001-dataset.html, July 2007.

[14] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 511–518, Kauai, HI, USA, December 2001.

[15] P. Viola, M. J. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," in *Proceedings of IEEE International Conference on Computer Vision (ICCV '03)*, vol. 2, pp. 734–741, Nice, France, October 2003.

[16] G. Bradski, A. Kaehler, and V. Pisarevsky, "Learning-based computer vision with Intel's open source computer vision library," *Intel Technology Journal*, vol. 9, pp. 119–130, 2005.

[17] R. Lienhart and J. Maydt, "An extended set of Haar-like features for rapid object detection," in *Proceedings of International*

Conference on Image Processing (ICIP '02), vol. 1, pp. 900–903, Rochester, NY, USA, September 2002.

[18] R.-L. Hsu, M. Abdel-Mottaleb, and A. K. Jain, "Face detection in color images," *IEEE Transaction on Pattern Analysis Machine Intelligence*, vol. 24, no. 5, pp. 696–706, 2002.

[19] A. Cavallaro and T. Ebrahimi, "Interaction between high-level and low-level image analysis for semantic video object extraction," *EURASIP Journal on Applied Signal Processing*, vol. 2004, no. 6, pp. 786–797, 2004.

[20] R. J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam, "Image change detection algorithms: a systematic survey," *IEEE Transactions on Image Processing*, vol. 14, no. 3, pp. 294–307, 2005.

[21] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 747–757, 2000.

[22] 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*, vol. 50, no. 2, pp. 174–188, 2002.

[23] Y. Rubner, J. Puzicha, C. Tomasi, and J. M. Buhmann, "Empirical evaluation of dissimilarity measures for color and texture," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '01)*, vol. 2, pp. 25–43, Kauai, HI, USA, December 2001.

[24] J. Ruiz-del-Solar and P. Navarrete, "Eigenspace-based face recognition: a comparative study of different approaches," *IEEE Transactions on Systems, Man and Cybernetics Part C*, vol. 35, no. 3, pp. 315–325, 2005.