Real-time Tracking of Multiple Skin-colored Objects with a Possibly Moving Camera

Antonis A. Argyros and Manolis I.A. Lourakis

Institute of Computer Science
Foundation for Research and Technology - Hellas (FORTH)
Vassilika Vouton, P.O.Box 1385, GR 711 10
Heraklion, Crete, GREECE
{argyros,lourakis}@ics.forth.gr

Abstract. This paper presents a method for tracking multiple skin-colored objects in images acquired by a possibly moving camera. The proposed method encompasses a collection of techniques that enable the modeling and detection of skin-colored objects as well as their temporal association in image sequences. Skin-colored objects are detected with a Bayesian classifier which is bootstrapped with a small set of training data. Then, an off-line iterative training procedure is employed to refine the classifier using additional training images. On-line adaptation of skin-color probabilities is used to enable the classifier to cope with illumination changes. Tracking over time is realized through a novel technique which can handle multiple skin-colored objects. Such objects may move in complex trajectories and occlude each other in the field of view of a possibly moving camera. Moreover, the number of tracked objects may vary in time. A prototype implementation of the developed system operates on 320x240 live video in real time (28Hz) on a conventional Pentium 4 processor. Representative experimental results from the application of this prototype to image sequences are also provided.

1 Introduction

An essential building block of many vision systems is one that permits tracking objects of interest in a temporal sequence of images. For example, this is the case of systems able to interpret the activities of humans, in which we are particularly interested. Such systems depend on effective and efficient tracking of a human operator or parts of his body (e.g. hands, face), as he performs a certain task. In this context, vision-based tracking needs to provide answers to the following fundamental questions. First, how is a human modeled and how are instances of the employed model detected in an image? Second, how are instances of the detected model associated temporally in sequences of images?

The human body is a complex, non-rigid structure with many degrees of freedom. Therefore, the type and complexity of the models employed for tracking vary dramatically [4, 3], depending heavily on the requirements of the application domain under consideration. For example, tracking people in an indoors

environment in the context of a surveillance application has completely different modeling requirements compared to tracking the fingers of a hand for sign language interpretation. Many visual cues like color, texture, motion and structure have been employed as the basis for the modeling of human body parts. Among those, skin color is very effective towards detecting the presence of humans in a scene. Color offers many advantages over geometric models, such as robustness under occlusions, scale and resolution changes, as well as geometric transformations. Additionally, the computational requirements of color processing are considerably lower compared to those associated with the processing of complex geometric models.

In the remainder of this section, we review existing approaches based on the answers they provide to the two fundamental questions stated above and, then, we outline the proposed method for addressing the tracking problem.

1.1 Color modeling and detection

A recent survey [22] includes a very interesting overview of the use of color for face (and, therefore skin-color) detection. A major decision towards providing a model of skin color is the selection of the color space to be employed. Several color spaces have been proposed including RGB [8], normalized RGB [12, 10], HSV [15], YCrCb [2], YUV [20], etc. Color spaces efficiently separating the chrominance from the luminance components of color are typically considered preferable. This is due to the fact that by employing chrominance-dependent components of color only, some degree of robustness to illumination changes can be achieved. Terrillon et al [18] review different skin chrominance models and evaluate their performance.

Having selected a suitable color space, the simplest approach for defining what constitutes skin color is to employ bounds on the coordinates of the selected space [2]. These bounds are typically selected empirically, i.e. by examining the distribution of skin colors in a preselected set of images. Another approach is to assume that the probabilities of skin colors follow a distribution that can be learned either off-line or by employing an on-line iterative method [15]. In the case of non-parametric approaches, the learnt distribution is represented by means of a color probabilities histogram. Other, so-called parametric approaches, are based either on a unimodal Gaussian probability density function [12, 20] or multimodal Gaussian mixtures [9, 14] that model the probability distribution of skin color. The parameters of a unimodal Gaussian density function are estimated by maximum likelihood estimation techniques. Multi-modal Gaussian mixtures require the Expectation-Maximization (EM) algorithm to be employed. According to Yang et al [21], a mixture of Gaussians is preferable compared to a single Gaussian distribution. Still, [10] argues that histogram models provide better accuracy and incur lower computational costs compared to mixture models for the detection of skin-colored areas in an image. A few of the proposed methods perform some sort of adaptation to become insensitive to changes in the illumination conditions. For example in [14] it has been suggested to adapt a

Gaussian mixture model that approximates the multi-modal distribution of the object's colors, based on a recent history of detected skin-colored regions.

1.2 Temporal data association

Assuming that skin-colored regions have been modeled and can be reliably detected in an image, another major problem relates to the temporal association of these observations in an image sequence. The traditional approach to solving this problem has been based on the original work of Kalman [11] and its extensions. If the observations and object dynamics are of Gaussian nature, Kalman filtering suffices to solve the tracking problem. However, in many cases the involved distributions are non-Gaussian and, therefore, the underlying assumptions of Kalman filtering are violated. As reported in [17], recent research efforts that deal with object tracking can be classified into two categories, the ones that solve the tracking problem in a non-Bayesian framework (e.g. [7, 16, 19]) and the ones that tackle it in a Bayesian one (e.g. [6, 13, 5]). In some cases [6], the problem of single-object tracking is investigated. These single-object approaches usually rely upon sophisticated, powerful object models. In other cases [13, 5] the problem of tracking several objects in parallel is addressed. Some of these methods solve the multi-object tracking problem by employing configurations of individual objects, thus reducing the multi-object tracking problem to several instances of the less difficult single-object tracking problem. Other methods employ particle filtering based algorithms, which track multiple objects simultaneously. Despite the considerable amount of research devoted to tracking, an efficient and robust solution to the general formulation of the problem is still lacking, especially for the case of simultaneous tracking of multiple targets.

1.3 Proposed approach

With respect to the two fundamental questions that have been posed, the proposed approach relies on a non-parametric method for skin-color detection and performs tracking in a non-Bayesian framework. Compared to existing approaches, the proposed method has several attractive properties. A skin-color representation is learned through an off-line procedure. A technique is proposed that permits the avoidance of much of the burden involved in the process of generating training data. Moreover, the proposed method adapts the employed skin-color model based on the recent history of tracked skin-colored objects. Thus, without relying on complex models, it is able to robustly and efficiently detect skin-colored objects even in the case of changing illumination conditions. Tracking over time is performed by employing a novel technique that can cope with multiple skin-colored objects, moving in complex patterns in the field of view of a possibly moving camera. Furthermore, the employed method is very efficient, computationally. The developed tracker operates on live video at a rate of 28 Hz on a Pentium 4 processor running under MS Windows.

The rest of the paper is organized as follows. Section 2 presents the proposed tracker. Section 3 provides sample results from the operation of the tracker in

long image sequences as well as issues related to its computational performance. Finally, section 4 provides the main conclusions of this work as well as extensions that are under investigation.

2 Method description

The proposed method for tracking multiple skin-colored objects operates as follows. At each time instance, the camera acquires an image on which skin-colored blobs (i.e. connected sets of skin-colored pixels) are detected. The method also maintains a set of object hypotheses that have been tracked up to this instance in time. The detected blobs, together with the object hypotheses are then associated in time. The goal of this association is (a) to assign a new, unique label to each new object that enters the camera's field of view for the first time, and (b) to propagate in time the labels of already detected objects. What follows, is a more detailed description of the approach adopted to solve the aforementioned subproblems.

2.1 Skin color detection

Skin color detection involves (a) estimation of the probability of a pixel being skin-colored, (b) hysteresis thresholding on the derived probabilities map, (c) connected components labeling to yield skin-colored blobs and, (d) computation of statistical information for each blob. Skin color detection adopts a Bayesian approach, involving an iterative training phase and an adaptive detection phase.

Basic training and detection mechanisms. A small set of training input images is selected on which a human operator manually delineates skin-colored regions. The color representation used in this process is YUV 4:2:2. However, the Y-component of this representation is not employed for two reasons. First, the Y-component corresponds to the illumination of an image point and therefore, by omitting it, the developed classifier becomes less sensitive to illumination changes. Second, by employing a 2D color representation (UV), as opposed to a 3D one (YUV), the dimensionality of the problem is reduced, as are the computational requirements of the overall system.

Assuming that image points $I(x, y)$ have a color $c = c(x, y)$, the training set is used to compute (a) the prior probability $P(s)$ of skin color, (b) the prior probability $P(c)$ of the occurrence of each color c in the training set and (c) the prior probability $P(c|s)$ of a color c being a skin color. Following the training phase, the probability $P(s|c)$ of a color c being a skin color can be computed by employing the Bayes rule:

$$P(s|c) = P(c|s)P(s)/P(c) \tag{1}$$

Then, the probability of each image point being skin-colored can be determined and all image points with probability $P(s|c) > T_{max}$ are considered as being

skin-colored. These points constitute the seeds of potential blobs. More specifically, image points with probability $P(s|c) > T_{min}$ where $T_{min} < T_{max}$, that are immediate neighbors of skin-colored image points are recursively added to each blob. The rationale behind this region growing operation is that an image point with relatively low probability of being skin-colored should be considered as such in the case that it is a neighbor of an image point with high probability of being skin-colored. This hysteresis thresholding type of operation has been very successfully applied to edge detection [1] and also proves extremely useful in the context of robust detection of skin-colored blobs. Indicative values for the thresholds T_{max} and T_{min} are 0.5 and 0.15, respectively. A connected components labeling algorithm is then responsible for assigning different labels to the image points of different blobs. Size filtering on the derived connected components is also performed to eliminate small, isolated blobs that are attributed to noise and do not correspond to interesting skin-colored regions. Each of the remaining connected components corresponds to a skin-colored blob. The final step in skin color detection is the computation of up to second order moments for each blob that will be used in the tracking process.

Simplifying off-line training. Training is an off-line procedure that does not affect the on-line performance of the tracker. Nevertheless, the compilation of a sufficiently representative training set is a time-consuming and labor-intensive process. To cope with this problem, an adaptive training procedure has been developed. Training is performed on a small set of seed images for which a human provides ground truth by defining skin-colored regions. Alternatively, already existing, publicly available training sets can be employed. Following this, detection together with hysteresis thresholding is used to continuously update the prior probabilities $P(s)$, $P(c)$ and $P(c|s)$ based on a larger image data set. The updated prior probabilities are used to classify points of these images into skin-colored and non-skin-colored ones. In cases where the classifier produces wrong results (false positives / false negatives), manual user intervention for correcting these errors is necessary; still, up to this point, the classifier has automatically completed much of the required work. The final training of the classifier is then performed based on the training set that results after user editing. This process for adapting the prior probabilities $P(s)$, $P(c)$ and $P(c|s)$ can either be disabled as soon as it is decided that the achieved training is sufficient for the purposes of the tracker or continue as more input images are fed to the system.

Adaptive detection. In the case of varying illumination conditions, skin color detection may produce poor results, even if the employed color representation has certain illumination-independent characteristics. Hence, a mechanism that adapts the representation of skin-colored image points according to the recent history of detected skin-colored points is required. To solve this problem, skin color detection maintains two sets of prior probabilities $P(s)$, $P(c)$, $P(c|s)$, corresponding to the off-line training set and $P_w(s)$, $P_w(c)$, $P_w(c|s)$, corresponding to

the evidence that the system gathers during the w most recent frames. Clearly, the second set better reflects the "recent" appearance of skin-colored objects and is better adapted to the current illumination conditions. Skin color detection is then performed based on:

$$P(s|c) = \gamma P(s|c) + (1-\gamma)P_w(s|c), \tag{2}$$

where $P(s|c)$ and $P_w(s|c)$ are both given by eq. (1) but involve prior probabilities that have been computed from the whole training set and from the detection results in the last w frames, respectively. In eq. (2), γ is a sensitivity parameter that controls the influence of the training set in the detection process. Setting $w = 5$ and $\gamma = 0.8$ gave rise to very good results in a series of experiments involving gradual variations of illumination.

2.2 Tracking multiple objects over time

We assume that at time t, M blobs have been detected as described in section 2.1. Each blob b_j, $1 \leq j \leq M$, corresponds to a set of connected skin-colored image points. Note that the correspondence among blobs and objects is not necessarily one-to-one. As an example, two crossing hands are two different skin-colored objects that appear as one blob at the time one occludes the other. In this work we assume that an object may correspond to either one blob or part of a blob. Symmetrically, one blob may correspond to one or many objects.

We also assume that the spatial distribution of the pixels depicting a skin-colored object can be coarsely approximated by an ellipse. This assumption is valid for skin-colored objects like hand palms and faces. Let N be the number of skin-colored objects present in the viewed scene at time t and o_i, $1 \leq i \leq N$, be the set of skin pixels that image the i-th object. We also denote with $h_i = h_i\,(c_{x_i}, c_{y_i}, \alpha_i, \beta_i, \theta_i)$ the ellipse model of this object where (c_{x_i}, c_{y_i}) is its centroid, α_i and β_i are, respectively, the lengths of its major and minor axis, and θ_i is its orientation on the image plane. Finally, we use capital letters $B = \cup_{j=1}^{M} b_j$, $O = \cup_{i=1}^{N} o_i$, and $H = \cup_{i=1}^{N} h_i$ to denote the union of skin-colored pixels, object pixels and ellipses, respectively. Tracking amounts to determining the relation between object models (h_i) and observations (b_j) in time. Figure 1 exemplifies

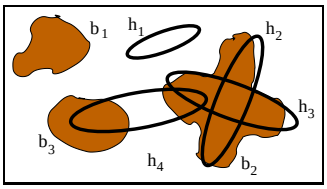

Fig. 1. Various cases of the relation between skin-colored blobs and object hypotheses.

the problem. In this particular example there are three blobs (b_1, b_2 and b_3)

while there are four object hypotheses (h_1, h_2, h_3 and h_4) from the previous frame.

What follows is an algorithm that can cope effectively with the data association problem. The proposed algorithm needs to address three different subproblems: (a) object hypothesis generation (i.e. an object appears in the field of view for the first time) (b) object hypothesis tracking in the presence of multiple, potential occluding objects (i.e. previously detected objects move arbitrarily in the field of view) and (c) object model hypothesis removal (i.e. a tracked object disappears from the field of view).

Object hypothesis generation. We define the distance $D(p, h)$ of a point $p = p(x, y)$ from an ellipse $h(c_x, c_y, \alpha, \beta, \theta)$ as follows:

$$D(p, h) = \sqrt{\overrightarrow{v} \cdot \overrightarrow{v}} \tag{3}$$

where

$$\overrightarrow{v} = \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix} \left(\frac{x - x_c}{\alpha}, \frac{y - y_c}{\beta} \right)$$

From the definition of $D(p, h)$ it turns out that the value of this metric is less than 1.0, equal to 1.0 or greater than 1.0 depending on whether point p is inside, on, or outside ellipse h, respectively. Consider now a model ellipse h and a point p belonging to a blob b. In the case that $D(p, h) < 1.0$, we conclude that the point p and the blob b support the existence of the object hypothesis h and that object hypothesis h predicts blob b. Consider now a blob b such that:

$$\forall p \in b, \ \min_{h \in H} \{D(p, h)\} > 1.0. \tag{4}$$

Equation (4) describes a blob with empty intersection with all ellipses of the existing object hypotheses. Blob b_1 in Fig. 1 is such a case. This, implies that none of the existing object hypotheses accounts for the existence of this blob. For each such blob, a new object hypothesis is generated. The parameters of the generated object hypothesis can be derived directly from the statistics of the distribution of points belonging to the blob. The center of the ellipse of the object hypothesis becomes equal to the centroid of the blob and the rest of the ellipse parameters can be computed from the covariance matrix of the bivariate distribution of the location of blob points. More specifically, it can be shown that if the covariance matrix Σ of the blob's points distribution is $\Sigma = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{xy} & \sigma_{yy} \end{bmatrix}$, then an ellipse can be defined with parameters:

$$\alpha = \sqrt{\lambda_1}, \quad \beta = \sqrt{\lambda_2}, \quad \theta = \tan^{-1} \left(\frac{-\sigma_{xy}}{\lambda_1 - \sigma_{yy}} \right) \tag{5}$$

where $\lambda_1 = \frac{\sigma_{xx} + \sigma_{yy} + \Lambda}{2}$, $\lambda_2 = \frac{\sigma_{xx} + \sigma_{yy} - \Lambda}{2}$, and $\Lambda = \sqrt{(\sigma_{xx} - \sigma_{yy})^2 + 4{\sigma_{xy}}^2}$.

Algorithmically, at each time t, all detected blobs are tested against the criterion of eq. (4). For all qualifying blobs, an object hypothesis is formed and

the corresponding ellipse parameters are determined based on eqs. (5). Moreover, all such blobs are excluded from further consideration in the subsequent steps of object tracking.

Object hypothesis tracking. After new object hypotheses have been formed as described in the previous section, all the remaining blobs must support the existence of past object hypotheses. The main task of the tracking algorithm amounts to associating blob pixels to object hypotheses. There are two rules governing this association:

- **Rule 1:** If a skin-colored pixel of a blob is located within the ellipse of some object hypothesis (i.e. supports the existence of the hypothesis) then this pixel is considered as belonging to this hypothesis.
- **Rule 2:** If a skin-colored pixel is outside all ellipses corresponding to the object hypotheses, then it is assigned to the object hypothesis that is closer to it, using the distance metric of eq. (3).

Formally, the set o of skin-colored pixels that are associated with an object hypothesis h is given by $o = R_1 \cup R_2$ where $R_1 = \{p \in B \mid D(p,h) < 1.0\}$ and $R_2 = \{p \in B \mid D(p,h) = \min_{k \in H}\{D(p,k)\}\}$.

In the example of Fig. 1, two different object hypotheses (h_2 and h_3) are "competing" for the skin-colored area of blob b_2. According to the rule 1 above, all skin pixels within the ellipse of h_2 will be assigned to it. According to the same rule, the same will happen for skin pixels under the ellipse of h_3. Note that pixels in the intersection of these ellipses will be assigned to both hypotheses h_2 and h_3. According to rule 2, pixels of blob b_2 that are not within any of the ellipses, will be assigned to their closest ellipse which is determined by eq. (3).

Another interesting case is that of a hypothesis that is supported by more than one blobs (see for example hypothesis h_4 in Fig. 1). Such cases may arise when, for example, two objects are connected at the time they first appear in the scene and later split. To cope with situations where a hypothesis h receives support from several blobs, the following strategy is adopted. If there exists only one blob b that is predicted by h and, at the same time, not predicted by any other hypothesis, then h is assigned to b. Otherwise, h is assigned to the blob with which it shares the largest number of skin-colored points. In the example of Fig. 1, hypothesis h_4 gets support from blobs b_2 and b_3. Based on the above rule, it will be finally assigned to blob b_3.

After having assigned skin pixels to object hypotheses, the parameters of the object hypotheses h_i are re-estimated based on the statistics of pixels o_i that have been assigned to them.

Object hypothesis removal. An object hypothesis should be removed either when the object moves out of the camera's field of view, or when the object is occluded by another (non-skin colored) object in the scene. Thus, an object hypothesis h should be removed from further consideration whenever

$$\forall p \in B,\ D(p,h) > 1.0. \tag{6}$$

Equation (6) essentially describes hypotheses that are not supported by any skin-colored image points. Hypothesis h_1 in Fig. 1 is such a case. In practice, we permit an object hypothesis to "survive" for a certain amount of time, even in the absence of any support, so that we account for the case of possibly poor skin-color detection. In our implementation, this time interval has been set to half a second, which approximately amounts to fourteen image frames.

Prediction. In the processes of object hypothesis generation, tracking and removal that have been considered so far, data association is based on object hypotheses that have been formed at the previous time step. Therefore, there is a time lag between the definition of models and the acquisition of data these models need to represent. Assuming that the immediate past is a good prediction for the immediate future, a simple linear rule can be used to predict the location of object hypotheses at time t, based on their locations at time $t-2$ and $t-1$. Therefore, instead of employing $h_i = h_i\left(c_{x_i}, c_{y_i}, \alpha_i, \beta_i, \theta_i\right)$, as the ellipses describing the object hypothesis i, we actually employ $\widehat{h_i} = h_i\left(\widehat{c_{x_i}}, \widehat{c_{y_i}}, \alpha_i, \beta_i, \theta_i\right)$, where $(\widehat{c_{x_i}}(t), \widehat{c_{y_i}}(t)) = C_i(t-1) + \Delta C_i(t)$. In the last equation, $C_i(t)$ denotes $(c_{x_i}(t), c_{y_i}(t))$ and $\Delta C_i(t) = C_i(t-1) - C_i(t-2)$.

The above equations postulate that an object hypothesis will maintain the same direction and magnitude of translation on the image plane, without changing any of its other parameters. Experimental results have shown that this simple prediction mechanism performs surprisingly well in complex object motions, provided that processing is performed close to real-time.

3 Experimental results

In this section, representative results from a prototype implementation of the proposed tracker are provided. The reported experiment consists of a long (3825 frames) sequence that has been acquired and processed on-line and in real-time on a Pentium 4 laptop computer running MS Windows at 2.56 GHz. A web camera with an IEEE 1394 (Firewire) interface has been used for this experiment.

For the reported experiment, the initial, "seed" training set contained 20 images and was later refined in a semi-automatic manner using 80 additional images. The training set contains images of four different persons that have been acquired under various lighting conditions.

Figure 2 provides a few characteristic snapshots of the experiment. For visualization purposes, the contour of each tracked object hypothesis is shown. Different contour colors correspond to different object hypotheses.

When the experiment starts, the camera is still and the tracker correctly asserts that there are no skin-colored objects in the scene (Fig. 2(a)). Later, the hand of a person enters the field of view of the camera and starts moving at various depths, directions and speeds in front of it. At some point in time, the camera also starts moving in a very jerky way; the camera is mounted on the laptop's monitor which is being moved back and forth. The person's second hand enters the field of view; hands now move in overlapping trajectories. Then,

the person's face enters the field of view. Hands disappear and then reappear in the scene. All three objects move independently in disjoint trajectories and in varying speeds ((b)-(d)), ranging from slow to fast; at some point in time the person starts dancing, jumping and moving his hands very fast. The experiment proceeds with hands moving in crossing trajectories. Initially hands cross each other slowly and then very fast ((e)-(g)). Later on, the person starts applauding which results in his hands touching but not crossing each other ((h)-(j)). Right after, the person starts crossing his hands like tying in knots ((k)-(o)). Next, the hands cross each other and stay like this for a considerable amount of time; then the person starts moving, still keeping his hands crossed ((p)-(r)). Then, the person waves and crosses his hands in front of his face ((s)-(u)). The experiments concludes with the person turning the light on and off ((v)-(x)), while greeting towards the camera (Fig.2(x)).

As it can be verified from the snapshots, the labeling of the object hypotheses is consistent throughout the whole sequence, which indicates that they are correctly tracked. Thus, the proposed tracker performs very well in all the above cases, some of which are challenging. Note also that no images of the person depicted in this experiment were contained in the training set. With respect to computational performance, the 3825 frames sequence presented previously has been acquired and processed at an average frame rate of 28.45 fps (320x240 images). The time required for grabbing a single video frame from the IEEE 1394 interface dominates the tracker's cycle time. When prerecorded image sequences are loaded from disk, considerably higher tracking frame rates can be achieved.

Besides the reported example, the proposed tracker has also been extensively tested with different cameras and in different settings involving different scenes and humans. Demonstration videos including the reported experiment can be found at http://www.ics.forth.gr/cvrl/demos.html.

4 Discussion

In this paper, a new method for tracking multiple skin-colored objects has been presented. The proposed method can cope successfully with multiple objects moving in complex patterns as they dynamically enter and exit the field of view of a camera. Since the tracker is not based on explicit background modeling and subtraction, it may operate even with images acquired by a moving camera. Ongoing research efforts are currently focused on (1) combining the proposed method with binocular stereo processing in order to derive 3D information regarding the tracked objects, (2) providing means for discriminating various types of skin-colored areas (e.g. hands, faces, etc) and (3) developing methods that build upon the proposed tracker in order to be able to track interesting parts of skin-colored areas (e.g. eyes for faces, fingertips for hands, etc).

Acknowledgements

This work was partially supported by EU IST-2001-32184 project ActIPret.

Fig. 2. Characteristic snapshots from the on-line tracking experiment.

References

1. J.F. Canny. A computational approach to edge detection. *IEEE Trans. on PAMI*, 8(11):769–798, 1986.
2. D. Chai and K.N. Ngan. Locating facial region of a head-and-shoulders color image. In *Proc. of FG'98*, pages 124–129, 1998.
3. Q. Delamarre and O. Faugeras. 3d articulated models and multi-view tracking wih physical forces. *Computer Vision and Image Understanding*, 81:328–357, 2001.
4. D.M. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73(1):82–98, 1999.
5. C. Hue, J.-P. Le Cadre, and P. Perez. Sequential monte carlo methods for multiple target tracking and data fusion. *IEEE Trans. on Signal Proc.*, 50(2):309–325, 2002.
6. M. Isard and A. Blake. Icondensation: Unifying low-level and high-level tracking in a stochastic framework. In *Proc. of ECCV'98*, pages 893–908, 1998.
7. O. Javed and M. Shah. Tracking and object classification for automated surveillance. In *Proc. of ECCV'02*, pages 343–357, 2002.
8. T.S. Jebara and A. Pentland. Parameterized structure from motion for 3d adaptive feedback tracking of faces. In *Proc. of CVPR'97*, pages 144–150, 1997.
9. T.S. Jebara, K. Russel, and A. Pentland. Mixture of eigenfeatures for real-time structure from texture. In *Proc. of ICCV'98*, pages 128–135, 1998.
10. M.J. Jones and J.M. Rehg. Statistical color models with application to skin detection. In *Proc. of CVPR'99*, volume 1, pages 274–280, 1999.
11. R.E. Kalman. A new approach to linear filtering and prediction problems. *Transactons of the ACME-Journal of Basic Engineering*, pages 35–45, 1960.
12. S.H. Kim, N.K. Kim, S.C. Ahn, and H.G. Kim. Object oriented face detection using range and color information. In *Proc. of FG'98*, pages 76–81, 1998.
13. E. Koller-Meier and F. Ade. Tracking multiple objects using the condensation algorithm. *Journal of Robotics and Autonomous Systems*, 34(2-3):93–105, 2001.
14. Y. Raja S. McKenna and S. Gong. Tracking color objects using adaptive mixture models. *IVC journal*, 17(3-4):225–231, 1999.
15. D. Saxe and R. Foulds. Toward robust skin identification in video images. In *Proc. of FG'96*, pages 379–384, 1996.
16. N.T. Siebel and S. Maybank. Fusion of multiple tracking algorithms for robust people tracking. In *Proc. of ECCV'02*, pages 373–387, 2002.
17. M. Spengler and B. Schiele. Multi object tracking based on a modular knowledge hierarchy. In *Proc. of International Conference on Computer Vision Systems*, pages 373–387, 2003.
18. J.C. Terrillon, M.N. Shirazi, H. Fukamachi, and S. Akamatsu. Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images. In *Proc. of FG'00*, pages 54–61, 2000.
19. J. Triesch and C. von der Malsburg. Democratic integration: Self-organized integration of adaptive cues. *Neural Computation*, 13(9):2049–2074, 2001.
20. M.H. Yang and N. Ahuja. Detecting human faces in color images. In *Proc. of ICIP'98*, volume 1, pages 127–130, 1998.
21. M.H. Yang and N. Ahuja. *Face Detection and Gesture Recognition for Human-computer Interaction*. Kluwer Academic Publishers, New York, 2001.
22. M.H. Yang, D.J. Kriegman, and N. Ahuja. Detecting faces in images: A survey. *IEEE Trans. on PAMI*, 24(1):34–58, 2002.