Tracking Skin-colored Objects in Real-time

Antonis A. Argyros and Manolis I.A. Lourakis

Institute of Computer Science (ICS)
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

We present a methodology for tracking multiple skin-colored objects in a monocular image sequence. The proposed approach encompasses a collection of techniques that allow the modeling, detection and temporal association of skin-colored objects across image sequences. A non-parametric model of skin color is employed. Skin-colored objects are detected with a Bayesian classifier that is bootstrapped with a small set of training data and refined through an off-line iterative training procedure. By using on-line adaptation of skin-color probabilities the classifier is able to cope with considerable illumination changes. Tracking over time is achieved by a novel technique that can handle multiple objects simultaneously. Tracked objects may move in complex trajectories, occlude each other in the field of view of a possibly moving camera and vary in number over time. A prototype implementation of the developed system operates on 320x240 live video in real time (28Hz), running on a conventional Pentium IV processor. Representative experimental results from the application of this prototype to image sequences are also presented.

1. Introduction

Locating and tracking objects of interest in a temporal sequence of images constitutes an essential building block of many vision systems. In particular, techniques for effectively and efficiently tracking the human body, either in part or as a whole, have received considerable attention in the context of applications such as face, gesture and gait recognition, markerless human motion capture, behavior and action interpretation, perceptual user interfaces, intelligent surveillance, etc. In such settings, 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?

Being a complex, non-rigid structure with many degrees of freedom, the human body is intricate to model. This is reflected on the models that have been employed in the literature for human tracking, whose type and complexity vary dramatically (Gavrila, 1999; DeCarlo & Metaxas, 2000; Delamarre & Faugeras, 2001; Plänkers & Fua, 2001), 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, shading, edges, texture, motion, depth and their combinations have been employed as the basis for modeling of human body parts. Among those, skin color is very effective towards detecting the presence of humans in a scene. Color offers significant advantages over geometric models, such as robustness under occlusions, resolution changes and geometric transformations. Additionally, color is a natural cue for focusing attention to salient regions in an image and the computational requirements for processing it are considerably lower compared to those associated with the processing of complex geometric models. In the remainder of this section, we briefly review existing approaches based on the answers they provide to the two fundamental questions stated above.

1.1 Modeling and detection of color

A recent survey (Yang et al, 2002) provides an interesting overview concerning the use of color for face (and, therefore, skin-color) detection. A major decision towards deriving a model of skin color relates to the selection of the color space to be employed. Several color spaces have been proposed including RGB (Jebara & Pentland, 1997), normalized RGB (Kim et al. 1998; Jones & Rehg, 1999), HSV (Saxe & Foulds, 1996), YCrCb (Chai & Ngan, 1998), YUV (Yang & Ahuja 1998), 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 the chrominance-dependent components of color only, some degree of robustness to illumination changes can be achieved. A review of different skin chrominance models and a comparative evaluation of their performance can be found in (Terrillon et al., 2000).

Once a suitable color space has been selected, the simplest approach for defining what constitutes skin color is to employ bounds on the coordinates of the selected space (Chai & Ngan, 1998). These bounds are typically selected empirically, i.e. by examining the distribution of skin colors in a pre-selected set of images. A more elaborate approach is to assume that the probabilities of skin colors follow a distribution that can be learnt either off-line or by employing an on-line iterative method (Saxe & Foulds, 1996). Depending on whether this distribution is represented analytically or not, existing approaches can be classified as parametric or non-parametric. Non-parametric approaches represent the learnt distribution by means of a histogram of color probabilities. Parametric approaches are based either on a unimodal Gaussian probability density function (Kimet et al., 1998, Yang & Ahuja, 1998) or on multimodal Gaussian mixtures (Jebara et al., 1998; Raja et al., 1999) that model the probability distribution of skin color. The parameters of a unimodal Gaussian density function are estimated using maximum likelihood estimation techniques. Multi-modal Gaussian mixtures typically require the Expectation-Maximization (EM) algorithm (Dempster et al., 1977) to be employed. According to Yang et al (Yang & Ahuja, 2001), a mixture of Gaussians is preferable compared to a single Gaussian distribution. Still, Jones and Regh (Jones & Regh, 1999) argue 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 instance, (Raja et al., 1999) suggest adapting 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. Vezhnevets et al (Vezhnevets et al., 2003) provide a survey of published pixel-based skin detection methods.

1.2 Tracking

Assuming that skin-colored regions have been appropriately modeled and can be reliably detected in an image, another major issue 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 (Kalman, 1960) and its extensions. If the observations and object dynamics are of a Gaussian nature, Kalman filtering suffices to optimally solve the tracking problem. However, in many practical cases the involved distributions are non-Gaussian and, therefore, the underlying assumptions of Kalman filtering are violated. As suggested in (Spengler & Schiele, 2003), recent research efforts that deal with object tracking can be classified into two categories, namely those that solve the tracking problem in a non-Bayesian framework (e.g. Javed & Shah 2002; Siebel & Maybank, 2002; Triesch & von de Malsburg, 2001) and those that tackle it in a Bayesian one (e.g. Isard & Blake 1998; Koller-Meier & Ade, 2001; Hue et al., 2002). In most of the cases (Isard & Blake, 1998), the problem of single-object tracking is investigated. These single-object approaches usually rely upon sophisticated, powerful object models. Other studies such as (Koller-Meier & Ade, 2001; Hue et al., 2002) address the more general problem of tracking several objects in parallel. Some of these methods employ configurations of several individual objects, thus reducing the multi-object tracking problem to a set of instances of the less difficult single-object tracking problem. Other methods employ algorithms making use of particle filtering (Arulampalam et al., 2002), i.e. sequential Monte-Carlo generalizations of Kalman filtering that are based on sampled representations of probability densities. Despite the considerable amount of research that has been 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 objects.

The rest of the paper is organized as follows. Section 2 provides an overview of the proposed skin color tracker. Sections 3 and 4 present the operation of the proposed tracker in more detail. Section 5 provides sample results from the application of the tracker to long image sequences and discusses issues related to its computational performance. Finally, section 6 provides the main conclusions of this work along with an outline of possible extensions to it.

2. Overview of the proposed approach

With respect to the two fundamental questions that have been posed in the introductory section, the proposed approach relies on a non-parametric method for skin-color detection and performs tracking in a non-Bayesian framework. A high level description of the operation of the proposed method for tracking multiple skin-colored objects is as follows. At each time instant, 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 the current instant in time. The detected blobs, together with the object hypotheses are then associated in time. The goal of this association is twofold, namely (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 that continue to be visible.

Compared to existing approaches, the proposed method has a number of attractive properties. Specifically, the employed skin-color representation does not make use of prohibitively complex physics-based models and is learned through an off-line procedure. Moreover, a technique is proposed that permits the avoidance of much of the burden involved in the process of manually generating training data. Being non-parametric, the proposed approach is independent of the shape of skin color distribution. Also, it 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. A prototype implementation of the proposed tracker operates on live video at a rate of 28 Hz on a Pentium IV processor running under MS Windows, without resorting to assembly optimizations or special hardware instructions such as MMX or SSE.

A more detailed description of the approach adopted by the proposed method for solving the two fundamental sub-problems identified in the introduction is supplied in the subsequent sections. An earlier description of the proposed method appears in (Argyros & Lourakis, 2004).

3. Detecting skin colored blobs

Skin color detection in the framework of the proposed method consists of four steps: (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. Section 3.1 describes the employed skin color detection mechanisms and sections 3.2 and 3.3 deal, respectively, with simplifying the process of off-line training and introducing adaptiveness to the skin detection procedure.

3.1 Basic training and skin detection schemes

During an off-line phase, 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 (Jack, 2004). However, the Y-component of this representation is not employed for two reasons. First, the Y-component corresponds to the illumination of an image pixel and therefore, by omitting it, the developed classifier becomes less sensitive to illumination changes. Second, compared to a 3D color representation (i.e. YUV), a 2D one (i.e. UV) is of lower dimensionality and is, therefore, less demanding in terms of memory storage and processing costs.

Assuming that image pixels with coordinates (x, y) have color values $c = c(x, y)$, training data are 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 and (c) the prior probability $P(c \mid s)$ of a color c being a skin color. Based on this information, the probability $P(s \mid c)$ of a color c being a skin color can be computed by employing the Bayes rule:

$$P(s \mid c) = \frac{P(c \mid s) P(s)}{P(c)}. \tag{1}$$

Equation (1) permits the determination of the probability of a certain image pixel being skin-colored using a lookup table indexed with the pixel's color. All pixels with probability $P(s \mid c) > T_{\max}$ are considered as being skin-colored. These pixels constitute seeds of potential skin-colored blobs. More specifically, image pixels with probabilities $P(s \mid c) > T_{\min}$ where $T_{\min} < T_{\max}$ that are immediate neighbors of skin-colored image pixels are recursively added to each blob. The rationale behind this region growing operation is that an image pixel with relatively low probability of being skin-colored should be considered as such in the case that it is a neighbor of an image pixel with high probability of being skin-colored. A similar type of hysteresis thresholding operation has been proven extremely useful to edge detection (Canny, 1986). Indicative values for the thresholds $T_{\max}$ and $T_{\min}$ are 0.5 and 0.15, respectively. A standard connected components labeling algorithm is then responsible for assigning different labels to the image pixels 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 involves the computation of central moments up to second order for each blob that, as will be explained shortly, will be needed during the tracking process.

3.2 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 such as the “Compaq” skin database of (Jones & Rehg, 1999) 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 pixels 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 resulting from user editing. This process for adapting the prior probabilities $P(s)$, $P(c)$ and $P(c|s)$ can either be disabled as soon as the achieved training is deemed sufficient for the purposes of the tracker, or continue as more input images are fed to the system.

3.3 Adaptive skin detection

The success of the skin-color detection process presented in section 3.1 depends critically on whether illumination conditions during the on-line operation of the detector are similar to those during the acquisition of the training data set. Despite the fact that the UV color representation model used has certain illumination independent characteristics, the skin-color detector may produce poor results if the illumination conditions during on-line operation are considerably different compared to the ones represented in the training set.

Hence, a means for adapting the representation of skin-colored image pixels according to the recent history of detected skin-colored pixels is required. To solve this problem, skin color detection maintains two sets of prior probabilities. The fist set consists of $P(s)$, $P(c)$, $P(c|s)$ that have been computed off-line from the training set while the second is made up of $P_W(s)$, $P_W(c)$, $P_W(c|s)$, 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 therefore better adapted to the current illumination conditions. Skin color detection is then performed based on the following weighted moving average formula:

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

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.

4. Tracking multiple objects over time

Let us assume that at time t, M blobs have been detected as described in section 3. Each blob b_j, $1 \le j \le M$, corresponds to a set of connected skin-colored image pixels. It should be noted 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 hand occludes the other. In this work, we assume that an object may correspond to either one blob or part of a blob. Conversely, one blob may correspond to one or many objects.

We also assume that the spatial distribution of 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. Extensive experimentation has demonstrated that the tracker still performs very well even in cases where the shape of skin-colored objects deviates significantly from the shape of an ellipse. Let N be the number of skin-colored objects present in the viewed scene at time t and o_i, $1 \le i \le N$, 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 $B = \bigcup_{j=1}^{M} b_j$, $O = \bigcup_{i=1}^{N} o_i$, and $H = \bigcup_{i=1}^{N} h_i$ to denote the union of all skin-colored pixels, object pixels and ellipses, respectively. Tracking amounts to determining the relation between object models h_i and observations b_j over time.

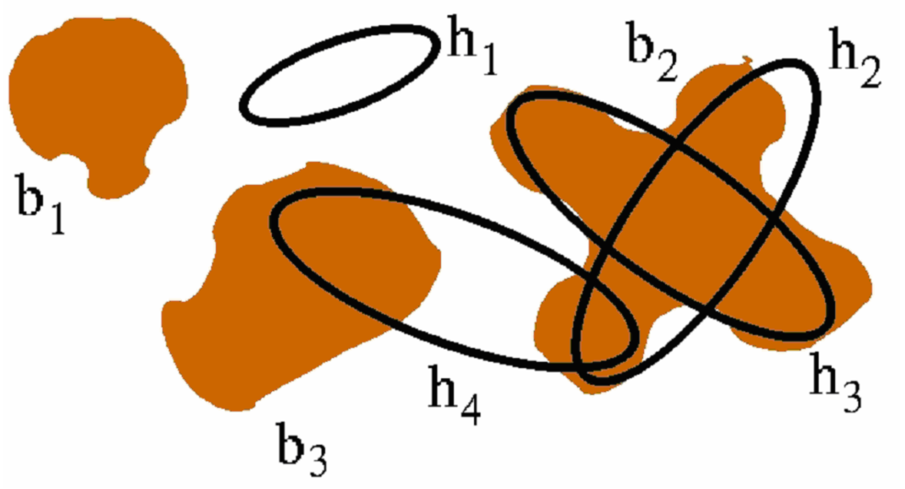


Fig. 1. Various possible configurations of skin-colored blobs and object hypotheses. See text for details.

Figure 1 exemplifies 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) carried from the previous frame.

What follows is an algorithm that can cope effectively with the temporal data association problem. The proposed algorithm needs to address three different sub-problems:

(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 that continue to move arbitrarily in the field of view).

(c) Object model hypothesis removal (i.e. a tracked object disappears from the field of view).

Each of the aforementioned problems is dealt with in the manner explained below.

4.1 Object hypothesis generation

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

$$D(p,h) = \| \vec{v} \|, \tag{3}$$

where $\|\cdot\|$ denotes the ℓ^2 norm of a vector and $\vec{v}$ is defined by

$$\vec{v} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} \dfrac{x - x_c}{\alpha} \\ \dfrac{y - y_c}{\beta} \end{bmatrix}. \tag{4}$$

The distance $D(p,h)$ is defined so that its value is less than, equal to or greater than 1 depending on whether pixel p is inside, on, or outside ellipse h, respectively. Consider now a model ellipse h and a pixel p belonging to a blob b. In the case where $D(p,h) < 1$, we conclude that pixel p and 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:

$$\forall p \in b, \quad \min_{h \in H} \{D(p,h)\} > 1. \tag{5}$$

Equation (5) describes a blob whose intersection with all ellipses of the existing object hypotheses is empty. Blob b_1 in Fig. 1 corresponds to 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 pixels belonging to the blob. The center of the ellipse of the object hypothesis becomes equal to the centroid of the blob and the remaining ellipse parameters can be computed from the covariance matrix of the bivariate distribution of the blob pixels location. More specifically, it can be shown that if the covariance matrix Σ corresponding to the distribution of the blob's pixels coordinates is $\Sigma = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{xy} & \sigma_{yy} \end{bmatrix}$, then an ellipse can be defined with parameters:

$$\alpha = \sqrt{\lambda_1}, \quad \beta = \sqrt{\lambda_2}, \quad \theta = \tan^{-1}\left(\frac{\sigma_{xy}}{\lambda_1 - \sigma_{yy}} \right), \tag{6}$$

where

$$\lambda_1 = \frac{\sigma_{xx} + \sigma_{yy} + \Lambda}{2}, \quad \lambda_2 = \frac{\sigma_{xx} + \sigma_{yy} - \Lambda}{2}, \quad \Lambda = \sqrt{\left(\sigma_{xx} - \sigma_{yy}\right)^2 + 4\sigma_{xy}{}^2} \tag{7}$$

Algorithmically, all blobs that are detected in each frame are tested against the criterion of eq. (5). For all qualifying blobs, a new object hypothesis is formed and the corresponding ellipse parameters are determined based on eqs. (6) and (7). Moreover, all such blobs are excluded from further consideration in the subsequent steps of object tracking.

4.2 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 a particular object hypothesis (i.e. supports the existence of this 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 closest to it in terms of 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 the union of two disjoint sets, specifically $o = R_1 \cup R_2$ with $R_1 = \{p \in B \mid D(p,h) < 1\}$ and $R_2 = \left\{p \in B \mid h = \arg\min_{n \in H}\{D(p,n) \mid D(p,n) \geq 1\}\right\}$.

In the example of Fig. 1, two different object hypotheses (h_2 and h_3) are "competing" for the skin-colored region corresponding to 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, as this is determined by eq. (3).

Another interesting case is that of a hypothesis that is supported by more than one blobs (such as 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 (e.g. two crossed hands) 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, is 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 pixels. 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 the set of pixels o_i that have been assigned to them, according to eq, (6).

4.3 Object hypothesis removal

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

$$\forall p \in B, \qquad D(p,h) > 1. \tag{8}$$

Equation (8) essentially describes hypotheses that are not supported by any skin-colored image pixels. Hypothesis h_1 in Fig. 1 illustrates such a case. In practice, and in order to account for the case of possibly poor skin-color detection, we allow an object hypothesis to "survive" for a certain amount of time, even in the absence of any support from skin-colored pixels. In our implementation, this time interval has been set to half a second. Thus, a hypothesis will be removed only after fourteen consecutive frames have been elapsed during which it has not received support from any skin-colored pixel. During these frames, the hypothesis parameters do not change but remain equal to the ones computed during the last frame in which it received support from some skin-colored pixels.

4.4 Prediction of hypotheses temporal dynamics

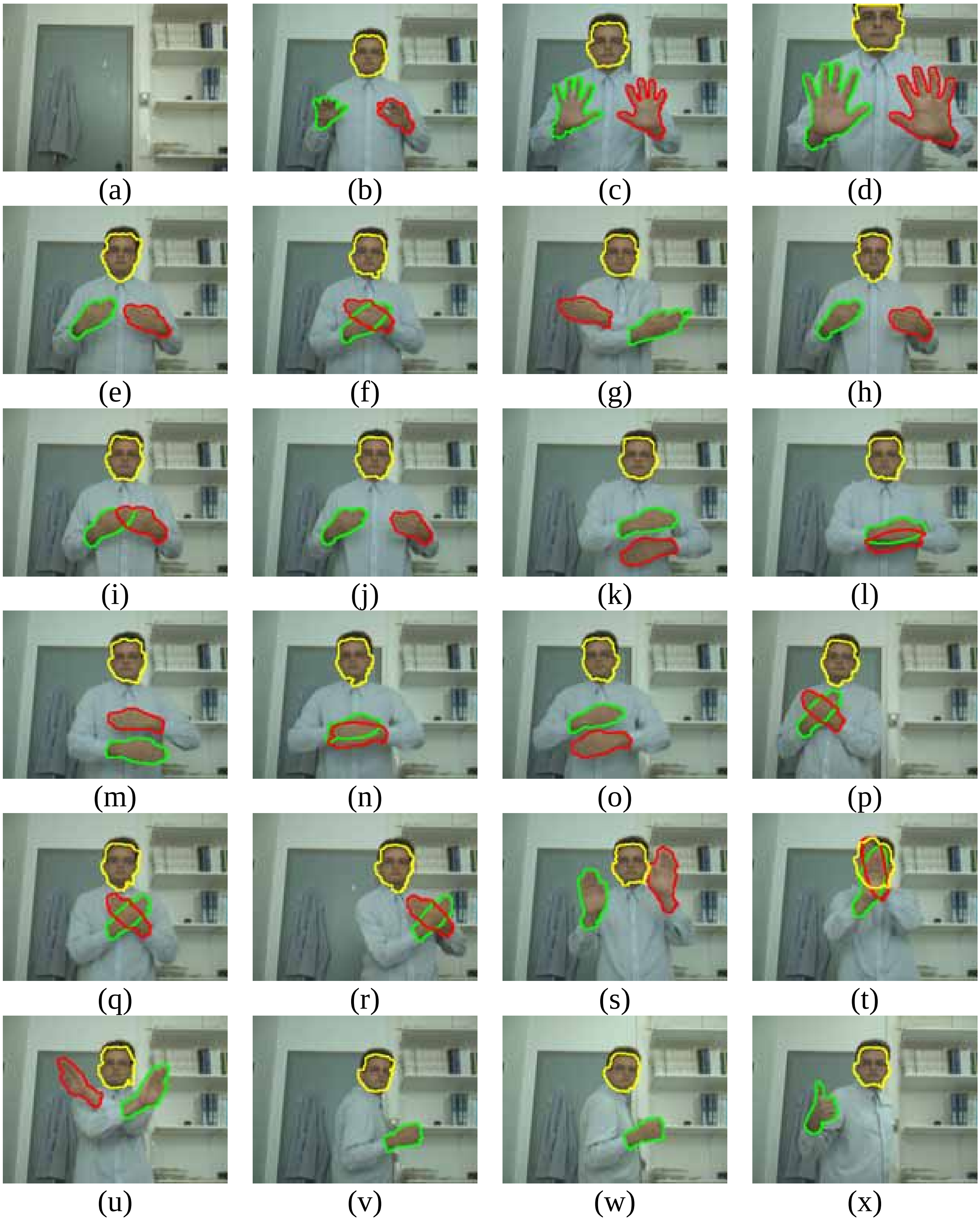
In the processes of object hypothesis generation, tracking and removal that have been described so far, data association is based on object hypotheses that have been formed or updated during the previous time step. Therefore, there is a time lag between the definition of models and the acquisition of data these models are intended 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 $h_i = h_i(c_{x_i}, c_{y_i}, \alpha_i, \beta_i, \theta_i)$ as the ellipses describing the object hypothesis i, we actually employ $\hat{h}_i = h_i(\hat{c}_{x_i}, \hat{c}_{y_i}, \alpha_i, \beta_i, \theta_i)$ where $\left(\hat{c}_{x_i}(t), \hat{c}_{y_i}(t)\right) = C_i(t-1) + \Delta C_i(t)$. In the last equation, $C_i(t)$ denotes $\left(c_{x_i}(t), c_{y_i}(t)\right)$ and $\Delta C_i(t) = C_i(t-1) - C_i(t-2)$.

The above definition exploits temporal continuity, i.e. it postulates 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 evaluation has indicated that this simple mechanism for predicting the evolution of hypotheses with time performs surprisingly well even for complex object motions, provided that processing is performed fast enough to keep up with real-time video acquisition.

5. Experiments

In this section, representative results from an experiment conducted using a prototype implementation of the proposed tracker are provided. The reported experiment is based on a long (3825 frames in total) sequence that has been acquired and processed on-line and in real-time on a Pentium IV laptop computer running MS Windows at 2.56 GHz. A web camera with an IEEE 1394 (Firewire) interface has been used for video capture. In this experiment, the initial, "seed" training set consisted of 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 a later 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 experiment concludes with the person turning the light on and off ((v)-(x)), while greeting towards the camera (Fig. 2(x)).



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

(e) (f) (g) (h)

(i) (j) (k) (l)

(m) (n) (o) (p)

(q) (r) (s) (t)

(u) (v) (w) (x)

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

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. Indeed, the proposed tracker performs very well in all the above cases, some of which are challenging. It should also be mentioned 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 with each frame being of dimensions 320x240. It is stressed that the reported frame rate is determined by the maximum acquisition frame rate supported by the employed camera, since the acquisition delay for a single frame dominates the tracker's cycle time. When employing prerecorded image sequences that are loaded from disk, considerably higher tracking frame rates can be achieved. Performance can be further improved by running the tracker on lower resolution images that result from subsampling original images by a factor of two.

Apart from the reported example, the proposed tracker has also been extensively tested with different cameras and in different settings involving different background scenes and human subjects. Demonstration videos including the reported experiment can be found online at http://www.ics.forth.gr/~argyros/research/colortracking.htm.

6. Conclusion

In this paper, a 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 on image sequences acquired by a moving camera. Moreover, the color modeling and detection modules facilitate robust performance in the case of varying illumination conditions. Owing to the fact that the proposed approach treats the problem of tracking under very loose assumptions and in a computationally efficient manner, it can serve as a building block of larger vision systems employed in diverse application areas.

Further research efforts have 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), (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) and (4) employing the proposed tracker for supporting human gesture interpretation in the context of applications such as effective human computer interaction.

Acknowledgements

This work was partially supported by EU IST-2001-32184 project ActIPret and by EU FP6-507752 NoE MUSCLE.

References

Argyros, A. & Lourakis, M. (2004). Real-time tracking of multiple skin-colored objects with a possibly moving camera. In *Proc. of ECCV’04,* vol. 3, pages 368-379.

Arulampalam, M.S. & Maskell, S. & Gordon, N. & Clapp, T. (2002). A tutorial on particle filters for on-line non-linear/non-Gaussian Bayesian tracking, *IEEE Trans. on Signal Proc.*, 50(2):174-188.

Canny, J.F. (1986). A computational approach to edge detection. *IEEE Trans. on Pat. Anal. and Mach. Intel.*, 8(11):769–798.

Chai, D. & Ngan, K.N. (1998). Locating facial region of a head-and-shoulders color image. In *Proc. of FG'98*, pages 124–129.

DeCarlo, D. & Metaxas, D. (2000). Optical flow constraints on deformable models with applications to face tracking. *International Journal of Computer Vision*, 38(2):99-127.

Delamarre, Q. & Faugeras, O. (2001). 3D articulated models and multi-view tracking with physical forces. *Computer Vision and Image Understanding*, 81:328–357.

Dempster, A.P & Laird, N.M. & Rubin, D.B. (1977). Maximum-likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1-38.

Gavrila, D.M. (1999). The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73(1):82–98.

Hue, C. & Le Cadre, J.-P. & Perez, P. (2002). Sequential monte carlo methods for multiple target tracking and data fusion. *IEEE Trans. on Signal Proc.*, 50(2):309–325.

Isard, M. & Blake, A. (1998). Icondensation: Unifying low-level and high-level tracking in a stochastic framework. In *Proc. of ECCV'98*, pages 893–908.

Jack, K. (2004). Video demystified. Elsevier science, 4th edition.

Javed, O. & Shah, M. (2002). Tracking and object classification for automated surveillance. In *Proc. of ECCV'02*, pages 343–357.

Jebara, T.S. & Pentland, A. (1997). Parameterized structure from motion for 3d adaptive feedback tracking of faces. In *Proc. of CVPR'97*, pages 144–150.

Jebara, T.S. & Russel, K. & Pentland, A. (1998). Mixture of eigenfeatures for real-time structure from texture. In *Proc. of ICCV'98*, pages 128–135.

Jones, M.J. & Rehg, J.M. (1999). Statistical color models with application to skin detection. In *Proc. of CVPR'99*, volume 1, pages 274–280.

Kalman, R.E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ACME-Journal of Basic Engineering*, pages 35–45.

Kim, S.H. & Kim, N.K. & Ahn, S.C. & Kim, H.G. (1998). Object oriented face detection using range and color information. In *Proc. of FG'98*, pages 76–81.

Koller-Meier, E. & Ade, F. (2001). Tracking multiple objects using the condensation algorithm. *Journal of Robotics and Autonomous Systems*, 34(2-3):93–105.

Plänkers, R. & Fua, P. (2001). Tracking and modeling people in video sequences. *Computer Vision and Image Understanding*, 81(3): 285-302.

Raja, Y. & McKenna, S. & Gong, S. (1999). Tracking color objects using adaptive mixture models. *Image and Vision Computinl*, 17(3-4):225–231.

Saxe, D. & Foulds, R. (1996). Toward robust skin identification in video images. In *Proc. of FG'96*, pages 379–384.

Siebel, N.T. & Maybank, S. (2002). Fusion of multiple tracking algorithms for robust people tracking. In *Proc. of ECCV'02*, pages 373–387.

Spengler, M. & Schiele, B. (2003). Multi object tracking based on a modular knowledge hierarchy. In *Proc. of International Conference on Computer Vision Systems*, pages 373–387.

Terrillon, J.C. & Shirazi, M.N. & Fukamachi, H. & Akamatsu, S. (2000). 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.

Triesch, J. & von der Malsburg, C. (2001). Democratic integration: Self-organized integration of adaptive cues. *Neural Computation*, 13(9):2049–2074.

Vezhnevets, V. & Sazonov, V. & Andreeva, A. (2003). A survey on pixel-based skin color detection techniques. In *Proc of Graphicon'03*, pages 85-92.

Yang, M.H. & Ahuja, N. (1998). Detecting human faces in color images. In *Proc. of ICIP'98*, volume 1, pages 127–130.

Yang, M.H. & Ahuja, N. (2001). *Face detection and gesture recognition for human computer interaction*. Kluwer Academic Publishers, New York.

Yang, M.H. & Kriegman, D.J. & Ahuja, N. (2002). Detecting faces in images: A survey. IEEE Trans. on Pat. Anal. and Mach. Intel., 24(1):34–58.