2003-03-25

Skin Color-Based Video Segmentation under Time-Varying Illumination

Boston University Computer Science Technical Report No. 2003-006, March 25, 2003. Submitted to IEEE PAMI.

Skin Color-Based Video Segmentation under Time-Varying Illumination

Leonid Sigal
Computer Science Department
Brown University
115 Waterman St.
Providence, RI 02912
ls@cs.brown.edu

Stan Sclaroff, Vassilis Athitsos
Image and Video Computing Group
Computer Science Department
Boston University
111 Cummington St.
Boston, MA 02215
{sclaroff, athitsos}@cs.bu.edu

Draft of April 3, 2003

Abstract

A novel approach for real-time skin segmentation in video sequences is described. The approach enables reliable skin segmentation despite wide variation in illumination during tracking. An explicit second order Markov model is used to predict evolution of the skin-color (HSV) histogram over time. Histograms are dynamically updated based on feedback from the current segmentation and predictions of the Markov model. The evolution of the skin-color distribution at each frame is parameterized by translation, scaling and rotation in color space. Consequent changes in geometric parameterization of the distribution are propagated by warping and resampling the histogram. The parameters of the discrete-time dynamic Markov model are estimated using Maximum Likelihood Estimation, and also evolve over time. The accuracy of the new dynamic skin color segmentation algorithm is compared to that obtained via a static color model. Segmentation accuracy is evaluated using labeled ground-truth video sequences taken from staged experiments and popular movies. An overall increase in segmentation accuracy of up to 24% is observed in 17 out of 21 test sequences. In all but one case the skin-color classification rates for our system were higher, with background classification rates comparable to those of the static segmentation.

Keywords: color video segmentation, human skin detection, dynamic Markov model.

1 Introduction

Locating and tracking patches of skin-colored pixels through an image sequence is a tool used in many face recognition and gesture tracking systems [1, 4, 5, 7, 8, 11, 12, 13]. Skin-color segmentation, particularly useful for its orientation and size invariance, is usually used for localization in early stages of these higher-level systems. An important challenge for any skin-color tracking system is to accommodate varying illumination conditions that may occur within an image sequence. Some robustness may be achieved via the use of luminance invariant color-spaces [7, 12]; however, such an approach can withstand only changes that skin-color distributions undergo within a narrow set of conditions. It has been shown that even in the chromaticity plane skin-color undergoes significant changes when the color temperature of the light source changes [16, 21].

The conditions that we are concerned with in this paper are broader than those assumed in many previous systems. In particular, we are concerned with three conditions: 1.) time-varying illumination, 2.) multiple sources, with time-varying illumination, and 3.) single or multiple colored sources. Most previous skin segmentation and tracking systems address only condition 1, defined over a narrow range (white light). Nevertheless, conditions 2 and 3 are also important, and have to be addressed in order to build a general purpose skin-color tracker. We will now list a few common scenarios that may lead to consideration of some or all of the conditions cited above.

Consider a person driving a car at night. Illumination from street lights and traffic lights will be at least in part responsible for the color appearance of his/her skin. Hence if we want to build a skin-color tracking system that would be used in surveiling the driver [9], we need to account for varying illuminant intensity and color. Clearly we cannot expect that color alone would be sufficient to give a robust solution in these generally dark illumination conditions, but it could serve as an important cue in the more integrated approach [32]. For example, it can be used to bootstrap infrared eye-detection [33], or extract the bounding box of the head for 3D gaze detection [32].

Skin-color person tracking is also useful in indexing multimedia content such as movies. In this case, multiple colored lights with varying intensity play a direct role, since many movies are filmed with theatrical lighting to dramatize the effects of the screenplay.

Still another example of time-varying color illuminant is apparent in observing a person walking down a corridor with windows or lights that are significantly spaced apart. The color appearance of the person's skin will smoothly change as they move towards and then away from various

light sources along the corridor.

Finally, it should be noted that it is not necessary to have colored lights to achieve effects equivalent to those that occur with colored lighting. Equivalent effects commonly arise due to surface inter-reflectance. For instance, consider a person walking down a corridor that has colored walls and/or carpet, or a person wearing colorful clothing [14]. These surfaces reflect a color tinge onto the person's skin.

These are a few examples of applications that motivate our approach. Even though we agree that the majority of everyday lighting effects are due to white light attenuation, we hold that it is important to consider alternatives as well, in order to have a robust skin-color tracker that can handle a wider variety of environmental conditions.

The goal of our system is to address smooth but relatively fast changing illumination and its effects on the skin-color appearance. An explicit second order Markov model is used to predict evolution of the skin-color distribution over time. Histograms are dynamically updated based on feedback from the current segmentation and predictions of the Markov model. The parameters of the discrete-time dynamic Markov model are estimated using Maximum Likelihood Estimation, and also evolve over time. Quantitative evaluation of the method has been conducted on video sequences taken from staged experiments and popular movies, and the results are encouraging.[1]

2 Related Work

Existing skin-color segmentation approaches can be grouped into two basic categories: physically-based approaches and statistical approaches. Statistical approaches can be subdivided further into: parametric approaches [4, 5, 7, 8, 12, 22, 24, 25] and non-parametric approaches [1, 14, 20, 31]. Both parametric and non-parametric statistical approaches usually perform color-segmentation in color spaces that reduce the effects of varying illuminant. A number of different color spaces have been used; however, normalized RGB [7, 12, 20, 31] and HSV [8, 14, 22, 24] are the most common color spaces used. It has been shown that in addition to being tolerant to minor variations in the illuminant, these color spaces also tend to produce minimum overlap between skin-color and background-color distributions [23].

[1] A Matlab implementation of the method, test sequences, segmentation results, and labeled ground-truth data are available from the web site: http://www.cs.bu.edu/groups/ivc/ColorTracking/.

Parametric statistical approaches represent the skin-color distribution in parametric form, such as a Gaussian model [4, 5, 12]. However, the skin-color distribution is oftentimes multimodal, and cannot be adequately represented as a single Gaussian in color space [12, 23, 24]. Therefore, the use of a mixture of Gaussians model has been proposed [7, 8, 22, 24, 25]. Typically, the Expectation-Maximization (EM) algorithm is employed to fit and update these models based on observed data [5, 7, 12, 22, 24, 25]. The key advantages of parametric models are: (a) low space complexity, and (b) relatively small training sets are required. The major difficulty is order selection, in particular for the Mixture of Gaussians case. The order is generally determined via heuristics. In constrained environments, the model order can be predefined based on the known environmental conditions.

The parameters of the skin-color distribution can vary significantly with people and lighting conditions [12, 16, 21, 24]. Thus, to build a system that is general enough to model and track different people, or robust enough to handle even modest variations in illumination conditions, one must employ an algorithm that adjusts the parameters of the distribution accordingly. In one of the first systems to take an adaptive approach, [12] used a linear combination of previous parameters to estimate the new parameters for the Gaussian distribution in normalized (r, g) color space. In a similar system, [5] used exponential functions (instead of linear) in the estimation of the evolving distribution's parameters. More recent work suggests that a low-order, dynamic Gaussian Mixture model in the chromatic color space, such as normalized RGB or HSV, is better-suited for skin-color modeling [8]. Adaptation techniques for Gaussian Mixture models usually employ a derivative of the EM algorithm, such as incremental EM [7]. With a small number of mixtures, evaluation and updates of the probability density function can be accomplished in real-time. However, if more mixtures are needed for accurate representation, this approach becomes infeasible.

A more general representation of the color distribution is available in the non-parametric statistical approaches [1, 14, 20, 31], where histograms are used to represent density in color space. A major advantage of the histogram representation is that the probability density function can be evaluated trivially regardless of the complexity of the underlying distribution. A major drawback is that the histogram approach requires a considerable amount of training data. Furthermore, care must be taken in setting the quantization level, which can be found via cross-validation [6] or using heuristics. Generally non-parametric approaches work well where the histograms are quantized properly and when sufficient training data is available.

The previous histogram-based methods do not directly model the time-varying nature of skin-color distributions in video. The face tracking system of [20] re-estimates the skin-color histogram every time it detects an eye blink, and thereby adapts to changes in ambient lighting. However, their system does not explicitly include a dynamical model for the evolution of the skin-color distribution under varying illuminant.

In a completely different approach [16, 21] made direct use of a physical model of skin reflectance. The reflectance model is used to discount a known, time-varying illuminant to obtain color constancy. Segmentation tends to be more accurate due to the algorithm's use of strong prior knowledge: camera and illumination parameters, as well as initial image segmentation. However, such information may not be readily available in analysis of everyday image sequences. In later work [31] introduced an algorithm that uses the skin locus in color space to adapt the skin-color over time. Pixels that fall within the tracked bounding box region and within the skin locus defined in the normalized RGB color space are assumed to be skin and are used to adapt the histogram over time. It was observed that the skin-color distribution for any given frame tends to occupy only a small portion of the locus; hence, tracking that small portion instead of the whole locus is advantageous in the cases where a cluttered background is present.

Finally, the accuracy of nearly any color-based skin segmentation algorithm can be improved if additional features are exploited. For instance, [7] used a Gaussian Mixture model in a 4D feature space that combined spatial coordinates (x, y) with (r, g) normalized color components. The mixture model was updated dynamically via an incremental EM algorithm. In a related approach [20] used the spatial center of gravity combined with histograms in normalized RGB color space to segment and track faces in real time. Applications in certain domains even allowed for explicitly pre-computed spatial distributions [22] over the image. Other approaches have incorporated stereo [4] and range data [17] in the skin-color segmentation algorithms. In general these approaches are accurate and more robust to occlusions, at the expense of their generality of use. In this paper we concern ourselves with a general approach that works purely on color data; however, other features can be incorporated if particular applications call for it.

In conclusion, all systems employ a color space representation that provides robustness to illumination variation. Statistical techniques that adapt the color distribution over time perform better than non-adaptive techniques. While a histogram representation offers some advantages over parametric models, no scheme for modeling a dynamic skin-color histogram has been proposed.

3 Overview of Approach

As noted in the previous section, changes in illumination can have drastic implications on the skin-color distribution. An example of this is shown in Fig. 1. These changes can be minimized in the chromaticity plane, but are still significant when one considers a wide range of illumination conditions. Disregarding the intensity component can lead to further robustness to the changing illuminant, but may also lead to separability issues with the background. In addition, hue and saturation become unstable when pixel intensities are too large or too small; this leads to the conclusion that intensity, however dependent on the illuminant it may be, is also important. It should also be noted that severe changes in the color of the illuminant can result in significant changes in the saturation and intensity. An example of this can be seen in Fig. 2, where theatrical lighting effects alter the skin-color distribution in HSV space quite markedly.

Recent research on the physical appearance of human skin [16] has shown that the skin reflectance locus is closely and directly related to the locus of the illuminant. Furthermore, experimental evidence has led researchers to postulate that the skin-color distribution changes smoothly, assuming non-abrupt illuminant changes such as attenuation and time varying lighting [12, 24]. This has been borne out by our own experiments with entertainment videos, one of which is shown in Fig. 1. Moreover, from the data described in [16] it can be shown that the changes in the skin-color distribution of a single person can be modeled accurately by a global affine transformation of the skin-color distribution in the color space.

Thus our goal is to track a moving skin-color distribution as defined by an adaptive color histogram in color space. We formulate a system that employs *predictive histogram adaptation* in modeling the color distribution over time. The evolution of the skin-color distribution at each frame is parameterized by translation, scaling and rotation in the color space. Consequent changes in geometric parameterization of the distribution are propagated by warping and re-sampling the histogram using the predicted affine transformations. The algorithm has three stages: initialization, learning, and then steady-state prediction/tracking.

3.1 Initialization

The initialization stage segments the first frame of the image sequence to give an initial estimate for the skin-color distribution to be tracked. This is done by using a two-class Bayes classifier.

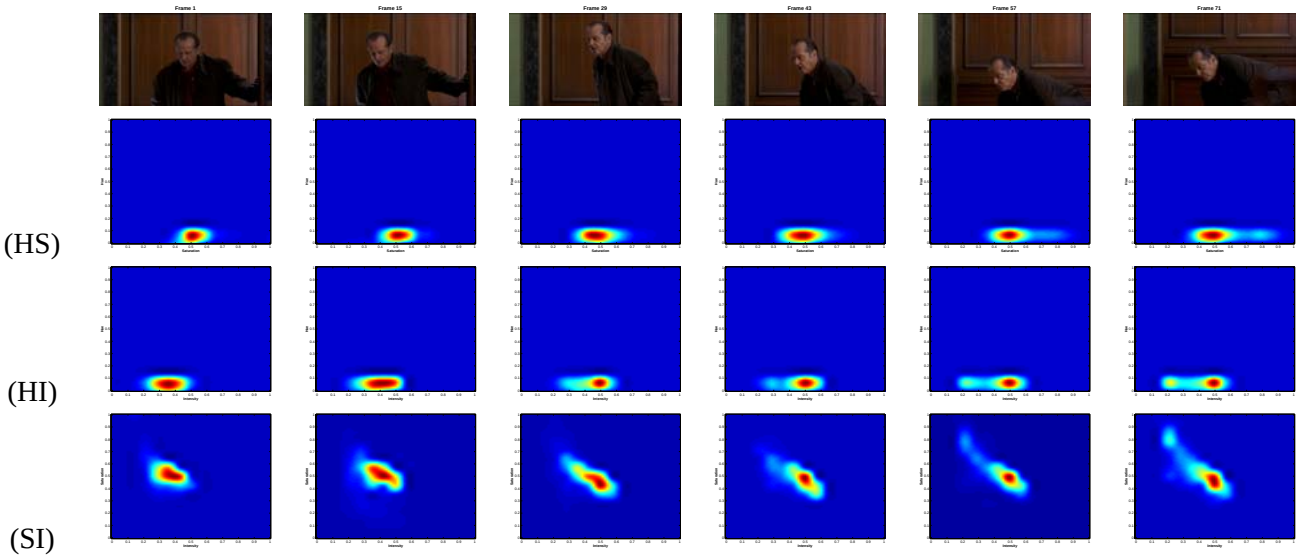

Figure 1: Example showing dynamic skin-color distribution in HSV space. The top row shows equally spaced video frames from a 72 frame long image sequence. The other rows show the corresponding 2D projection views of the skin-color distribution in HSV color space for each frame: (HS) Hue-Saturation, (HI) Hue-Intensity, and (SI) Saturation-Intensity plane projections are shown.

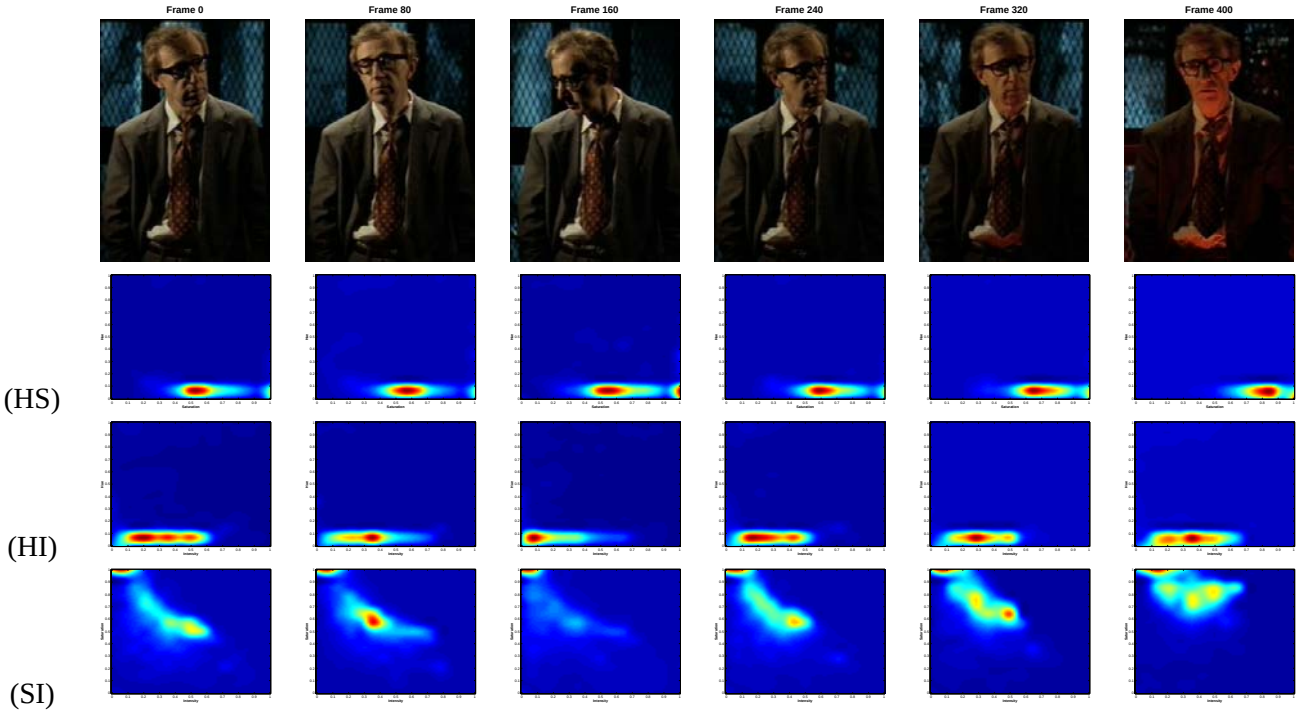

Figure 2: Example of dynamic skin-color distribution in HSV space when there is a dramatic change in the illuminant color (due to theatrical lighting). The top row shows images taken from a 400 frame long video sequence, sampled equally in time. The other rows show the 2D projection views of the skin-color distribution in HSV color space for each video frame: (HS) Hue-Saturation, (HI) Hue-Intensity, and (SI) Saturation-Intensity plane projections.

The prior histograms used for classification are precomputed off-line using the database provided by Jones and Rehg [6]. The resulting crude segmentation estimate is then refined with binary image processing to produce a binary mask for the initial skin-color regions to be tracked. This initialization is simply a pre-processing step, and hence other algorithms such as [14, 15] can be used for this stage. Alternatively, object detection may also be used to bootstrap this portion of the system. If we assume that the face is always present in the first frame, then face detection [13, 18] may be used to find initial patches of skin.

3.2 Learning

Given the results of the initialization described above, a fully-automated learning stage performs an Expectation Maximization (EM) process over the first few frames in the video sequence. For each frame, the EM algorithm's E step is histogram-based segmentation, and the M step is histogram adaptation. This process defines the evolution of the distribution in discrete time. The evolution of the distribution is implicitly defined in terms of translation, rotation, and scaling of the samples in color space. The transformation parameters are easily estimated via standard statistical methods. Given the evolution of the parameters, we can estimate the motion model for the skin color distribution, and hence predict further deformations. The motion model that we use for the predictions is a second order discrete-time Markov model. The Markov model parameters are estimated by maximum likelihood estimation.

3.3 Prediction and Tracking

Once a motion model is learned we proceed to the prediction/tracking stage. At this stage, in addition to segmentation and distribution estimation, changes in translation, scaling and rotation of the distribution are *predicted* given the Markov model estimated in the learning stage. The parameters of Markov model are re-estimated over time as well. By predicting parametric changes, we can get a better estimate of the true distribution at the next time step. Even though adaptive histograms are used for segmentation, we cannot apply the predictions to the histograms directly due to the problems with resolution and sampling. Instead the predictions are propagated via a transformation applied on the samples directly. The newly transformed samples are used to help estimate the histogram used for segmentation of the next frame.

While both background and skin-color distributions are adaptive, the background distribution need not be predictive. We assume that changes in the background distribution are considerably smaller in general than changes in the foreground (skin) distribution. Hence the change in illumination that is due to the motion will be smaller. This is a reasonable assumption for a grand majority of scenes, particularly if the camera is stationary or moving slowly. The background generally is much less dependent on lighting conditions and does not change significantly from one frame to the next. The exception to this are the scenes that stabilize the camera on the moving character. These exceptions are rare however, and hence it is customary to assume that motion in the background is smaller than that in the foreground distribution [1, 7, 11]. In our experience adequate segmentation of image sequences can be achieved using a simple adaptive histogram implementation for background distribution.

Each of the three basic stages of the algorithm will now be described in greater detail.

4 Initialization

The first stage of the system is designed to give an initial estimate for the location of the foreground (skin) and background (non-skin) regions in the first frame of the image sequence. This is achieved by segmenting the first frame, with histogram-based conditional probability distributions for the two classes that have been computed off-line. The diagram of the algorithm can be seen in Fig. 3. After the first frame is segmented using Bayes classifier, spatial morphological filters, such as size filtering and hole filtering, are used to clean up the mask.

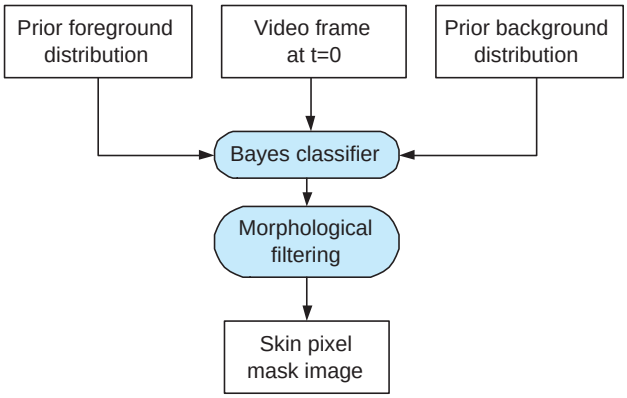

Figure 3: Initialization stage of the algorithm.

4.1 Prior Histogram Learning

Histograms for the skin and background distributions are learned off-line from a database provided by Jones and Rehg [6].[2] The database contains 4668 skin images with corresponding masks and 8960 non-skin images. All images were collected from the world wide web and skin regions were labeled by hand.

Following [6], histogram-based distributions were computed at a $32 \times 32 \times 32$ bin resolution in RGB color space. Results obtained in [6] showed that $32 \times 32 \times 32$ bin histograms are not only sufficient but are superior in the segmentation to the fully-ranked $256 \times 256 \times 256$ histograms. The lower-dimensional histograms are advantageous mainly because they require considerably fewer training pixels to adequately estimate the distribution. Estimating a full-resolution $256 \times 256 \times 256$ histogram would require on the order of 512 times more training data.

Conditional probability densities were obtained by dividing the count of pixels in each histogram bin by the total number of pixels in the histogram. The conditional densities will be denoted $P(rgb|fg)$, and $P(rgb|bg)$, where *fg* denotes foreground, *bg* background, and $rgb \in \Re^3$.

Looking at the different rendering views of the skin-color distribution, we can infer some structure of the distribution. As originally observed in [6] the skin-color distribution for this database is fairly compact, and lies mainly along the gray line. Where the gray line is defined to be the line of gray values in the color space. In the background distribution, black and white are the most common colors, and the colors most frequently fall near the gray line.

4.2 Skin Segmentation Using Prior Histograms

Using Bayes' formula, we can compute $P(fg|rgb)$ and $P(bg|rgb)$. The classification boundary can be drawn where the ratio of $P(fg|rgb)$ and $P(bg|rgb)$ exceeds some threshold K that is based on a relative risk factor associated with misclassification. For example

$$K < \frac{P(fg|rgb)}{P(bg|rgb)} = \frac{P(rgb|fg)P(fg)}{P(rgb|bg)P(bg)} \tag{1}$$

corresponds to pixel value *rgb* being labeled as foreground. Rearranging terms

$$K \times \frac{1 - P(fg)}{P(fg)} < \frac{P(rgb|fg)}{P(rgb|bg)}, \tag{2}$$

[2]The database given to us by Jones and Rehg contained some images that were unreadable, hence our numbers for the number of images and number of pixels in each class are slightly different from the numbers given in [6].

where $P(fg)$ is the probability of an arbitrary pixel in an image being skin. Clearly this probability will vary from image to image, but given a large enough data set we can come up with the aggregate probability that can serve as our best estimate. Using the entire database as the data set we can express $P(fg)$ as

$$P(fg) = \frac{N_{\text{foreground}}}{N_{\text{foreground}} + N_{\text{background}}}, \tag{3}$$

where $N_{\text{foreground}}$ is the total number of pixels in the foreground histogram, and $N_{\text{background}}$ is the total number of pixels in the background histogram. In our training database, $N_{\text{foreground}} = 80,306,243$ and $N_{\text{background}} = 861,142,189$. Hence in our training database, $P(fg) = 0.09$.

Given $P(fg)$, we can now empirically establish the threshold K. One of the standard ways of determining the threshold is by computing a Receiver Operating Characteristic (ROC) curve. The ROC curves in Fig. 4 show the tradeoff between the true positives and false positives for various possible settings of the decision criterion K.

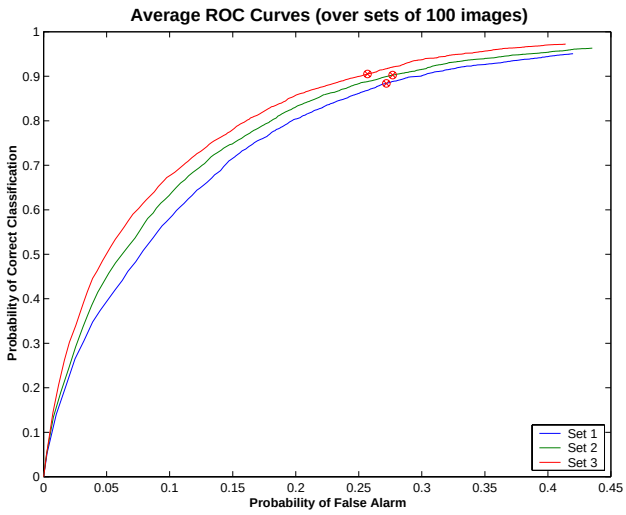

Figure 4: Average ROC curves (computed over three random sets of 100 images excluded from the training data) for skin segmentation as a function of the threshold K. The x-axis corresponds to the probability of false detection, and the y-axis to the probability of correct classification of skin pixels.

A threshold was chosen such that at least 85% correct classification is achieved while having under 25% chance of false alarm. This choice was made in light of [6] and the fact that the optimal value for the threshold should lie near the bend of the ROC curve. The selected threshold was $K = 0.0673$. This was consistent across a number of trials.

The result of the pixel classification scheme above is a binary image mask in which 0's correspond to background pixels, and 1's to foreground pixels. In order to minimize noise effects, we

employ size and hole filtering before the binary mask is passed to the learning stage of the system.

5 Learning

Thus far, only aggregate statistics have been employed in segmentation. However, our ultimate goal is to learn the statistics that are specific to the image sequence at hand. The mask for the first frame of the sequence (provided by the initialization) is a good initial estimate of skin and non-skin regions. The pixels from those regions can be used to re-estimate histograms for the foreground and the background. The new histograms are sequence-specific, and hence are better estimates. The new sequence-specific value for the $P(fg)$ is also re-estimated based on the image mask. Since the histograms are sampled from a single image, it is possible to run into sampling problems, especially for the skin-color distribution, since the skin regions are usually relatively small in comparison to the background. In practice we found that smoothing the skin-color histogram using a $3 \times 3 \times 3$ Gaussian kernel with a small $\sigma = 0.45$ helps resolve this problem.

However, using static histograms for an image sequence where the distribution constantly changes is inappropriate; hence, we employ an adaptive histogram scheme to facilitate the tracking of the distributions. ¿From the foreground and background distributions observed over an initial sequence of frames, sequence-specific motion patterns are learned. A second-order Markov process is used to model the evolution of the color distributions over time. The flow diagram of this stage of our system is shown in Fig. 5.

¿From the Fig. 5 we can see that once the new frame from the image sequence is acquired and converted into the appropriate color space (HSV), the histogram estimates from the previous frame are used to segment the new image. The newly classified pixels are then used as color features to update the foreground and background distributions. The evolution of the foreground distribution, parameterized at each time (t) by translation, rotation and scaling in the color space is then used to estimate parameters of the second-order Markov motion model. The formulation will now be described in greater detail.

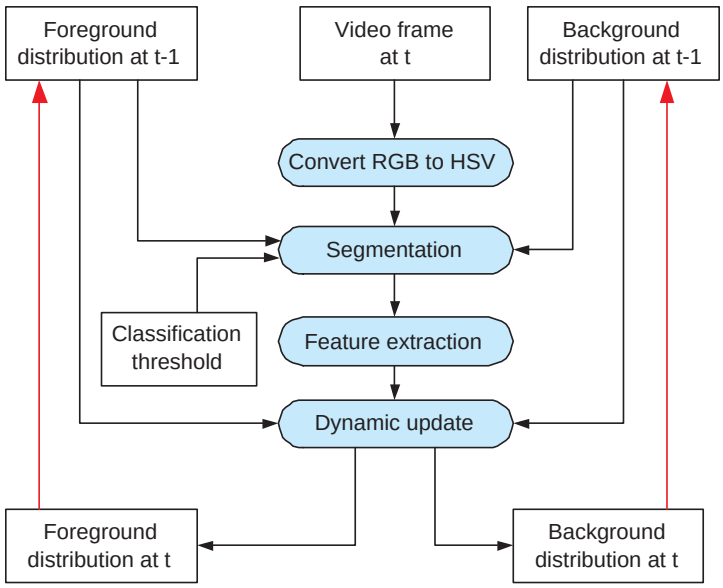

Figure 5: Learning stage of the algorithm.

5.1 Color Space for Skin-Color Tracking

An important aspect of any skin-color tracking system is choosing a color space that is relatively invariant to minor illuminant changes. The two popular color spaces that have proven to be robust to minor illuminant changes are HSV and normalized RGB. The normalized RGB space has only two dimensions, and hence estimating the probability density function in normalized RGB space requires considerably less training data. However, this comes at the expense of discriminating power that can be compromised by projecting the data into this lower dimensional space. Terrillon and Akamatsu [23] recently conducted a study on the comparative segmentation performance of nine different color spaces. The color spaces that were best in terms of minimizing overlap between the skin and background distributions were HSV and normalized RGB in that order, with the HSV color space having roughly one percent less overlap. Slightly worse discriminability was observed for the TSL, CIE-xy, and CIE-DSH color spaces. Since it is important to have as much discrimination between skin-color and background-color as possible, we felt that use of HSV color space in our system was well-grounded.

In preliminary experiments we found that the HSV color space is much better-suited than normalized (r, g) for estimation and prediction of skin-color distribution evolution in image sequences taken from entertainment videos and movies.

The only disadvantage of the HSV color space is the costly conversion from the standard RGB

video source. In the real-time implementation of the system we handled this problem by quantizing the HSV space into a ($64 \times 64 \times 64$) RGB to HSV lookup table. To gain a uniform sampling of the color space, each of the HSV color channels is normalized to floating point values between 0 and 1, given the expected range of HSV values.

5.2 Motion of Distributions

As mentioned earlier, skin-color distributions tend to evolve over the sequence of observed frames. In order to model and predict this evolution, we need to make some assumptions about the types of motions that distributions can undergo in the color space.

One assumption is that the skin-color distribution evolves as a whole; thus, there cannot be any local deformation or evolution of the distribution. This is similar to the global illuminant assumption, where one assumes nothing about the nature of the illuminant, so long as it acts uniformly over all skin patches in the image. Changing non-uniform illumination is likely to cause local deformations in the skin-color distribution, which will invalidate our assumptions and call for a more complex deformation model. Furthermore, global deformations of the distribution are assumed to be affine. These decisions are based on observations made in goodness of fit studies. To further simplify our prediction model we constrain ourselves to the three most significant affine transformations: translation, rotation and scaling. We employ an eight-parameter vector:

$$\xi = [T_H, T_S, T_V, S_H, S_S, S_V, \theta, \phi]^T \tag{4}$$

where T_i are translation, S_i scaling, and θ and ϕ are angles of spherical rotation applied about the mean of the skin-color distribution.

In order to visualize how these parameters change over time in a typical image sequence, we plotted them over 72 consecutive frames of the sample sequence shown in Fig. 1. The chosen sequence contains an actor who bends to pick up a newspaper, in front of a skin-colored door background. As he conducts this task the distance to the light sources changes, resulting in transformations in the skin color distribution. The parameters of the distribution are shown in Fig. 6. The top and middle rows show graphs of the skin-color distribution's mean (translation) and standard deviation (scale). The graphs in the bottom row show evolution of the rotation parameters θ and ϕ, computed as the difference between the first frame of the image sequence and the frame

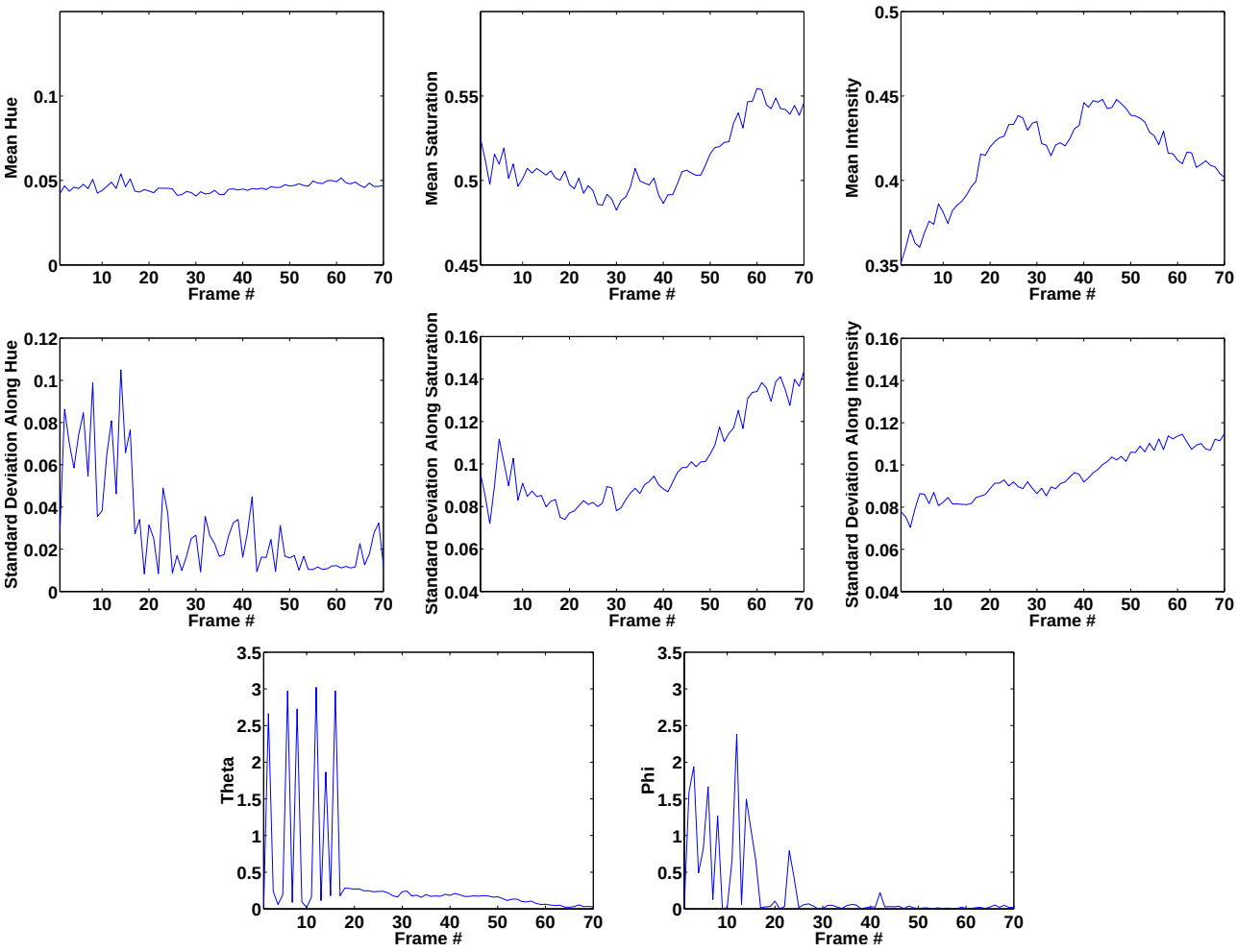

Figure 6: Evolution of the skin-color distribution parameters over time, for the image sequence shown in Fig. 1. The graphs of angles Theta and Phi are given in units of degrees.

at time t. Note that the distribution's variation in rotation is fairly small for this sequence (< 3 degrees variation).

Two important observations can be made from observing the parameter evolution (1) parameters change smoothly over time (2) parameters can change in non-linear fashion, like the mean along the Intensity axis. This follows intuition and supports our choice for the dynamical model.

5.3 Estimating Distribution Motion Parameters

In order to learn parameters of the dynamic motion model, and predict the skin-color distribution at the next time step, $t + 1$, we use a sliding window of the last $n + 2$ frames. Frames $t - n - 2$ to $t - 2$ inclusively are used for learning the parameters of the Markov model, and frames $t - 1$

and t are used to define the state vector that is used for prediction. For each of these $n+2$ frames, parameter vectors of the form shown in Eq. 4 must be estimated.

We define $\tau = t - n - 2$ as the base frame in the sliding window of the previous $n+2$ frames, for a given time t. For any frame $k \in (0, n+2)$ the translation parameters $T_{H,k}$, $T_{S,k}$, and $T_{V,k}$ can be extracted directly from the means of the HSV skin-color distribution histogram at time $\tau + k$. The scaling parameters $S_{H,k}$, $S_{S,k}$, and $S_{V,k}$ can be extracted from the standard deviation of the skin-color distribution. The standard deviation represents the relative scaling of the distribution along the Hue, Saturation, and Intensity axes.

Estimating rotation is slightly more complicated. We measure differential rotation of the distribution from the base frame at time τ. It is assumed that the incremental rotation of the distribution is smooth and relatively small. The eigenvectors of the covariance matrices for the skin-color distributions of frames τ and $\tau + k$ define the two coordinate frames. Our problem is reduced to finding the two angles in the spherical coordinate space centered at the mean that would align these two coordinate systems. The first angle can be found as follows:

$$\theta_k = acos(e_{1,\tau} \cdot e_{1,\tau+k}), \tag{5}$$

where $e_{1,\tau}$ is the eigenvector corresponding to the largest eigenvalue of the covariance matrix at time τ, and $e_{1,\tau+k}$ is the eigenvector corresponding to the largest eigenvalue at time $\tau + k$. The axis of rotation $v_{\theta,k}$ is found via the cross product: $v_{\theta,k} = e_{1,\tau} \times e_{1,\tau+k}$.

This defines the rotation $R(v_{\theta,k}, \theta_k)$ that will align the corresponding axes of greatest variation. This rotation when applied to $e_{2,\tau}$ and $e_{3,\tau}$ will put them in the plane perpendicular to $e_{1,\tau+k}$. In order to align the axes defined by $e_{2,\tau}$ and $e_{2,\tau+k}$ as well as $e_{3,\tau}$ and $e_{3,\tau+k}$ we need to apply a single rotation about $e_{1,\tau+k}$. The angle of this second rotation is $\phi_k = acos((R(v_{\theta,k}, \theta_k) \cdot e_{2,\tau}) \cdot e_{2,\tau+k})$.

5.4 Dynamical Distribution Model

In order to estimate and predict the skin-color distribution over time we need to formalize a dynamic motion model. It has been shown that affine motion can be fully expressed in terms of an auto-regressive Markov process [2]. A second-order dynamical process handles both oscillatory and arbitrary translational motion. We will now formulate the discrete second-order Markov process that will be used in our system. The formulation follows [2].

First, we define the N-dimensional state vector X, which in our case is an eight-dimensional parameter vector (Eq. 4). The system's second-order dynamics are defined by a stochastic differential equation [2]. The stochastic portion of the dynamics is modeled by zero mean, unit variance N dimensional Brownian motion. For our application, we utilize the discrete-time model:

$$\begin{bmatrix} X_n - \bar{X} \\ X_{n+1} - \bar{X} \end{bmatrix} = \begin{bmatrix} 0 & I \\ A_0 & A_1 \end{bmatrix} \begin{bmatrix} X_{n-1} - \bar{X} \\ X_n - \bar{X} \end{bmatrix} + \begin{bmatrix} 0 \\ Bw_n \end{bmatrix}. \tag{6}$$

The mean vector $\bar{X}$ corresponds to the observed mean displacement in each of the eight affine parameters. The $N \times N$ sub-matrices A_0 and A_1 govern the deterministic part of the motion model, whereas sub-matrix B governs the stochastic part. Rearranging terms yields:

$$X_{n+1} = A_0 X_{n-1} + A_1 X_n + (I - A_0 - A_1)\bar{X} + Bw_n. \tag{7}$$

5.5 Learning Parameters for the Dynamical Model

An algorithm for learning the parameters of the proposed second-order Markov dynamical model is needed. The parameters to be learned are A_0, A_1, and B. Unfortunately it is impossible to observe B directly; instead we observe $C = BB^T$. We can estimate these parameters using a standard MLE algorithm described in [3]. The results of the algorithm are restated here for completeness.

To simplify the notation, let $X \rightarrow X - \bar{X}$, where $\bar{X}$ is the expected value of X. The log-likelihood function L that we need to maximize is:

$$L(X_1, \ldots, X_n | A_0, A_1, B) = -\frac{1}{2} \sum_{n=1}^{m-2} |B^{-1}(X_{n+1} - A_0 X_{n-1} - A_1 X_n)|^2 - (m-2) \log \det B. \tag{8}$$

We can estimate the parameters by first maximizing with respect to A_0 and then with respect to A_1. It was shown by A. Blake and M. Isard in [3] that matrices A_0 and A_1 can be estimated from the following set of simultaneous equations:

$$\begin{aligned} S_{20} - \hat{A}_0 S_{00} - \hat{A}_1 S_{10} &= 0 \\ S_{21} - \hat{A}_0 S_{01} - \hat{A}_1 S_{11} &= 0, \end{aligned} \tag{9}$$

where

$$S_{ij} = \sum_{n=1}^{m-2} X_{(n-1)+i} X_{(n-1)+j}^T, \quad i, j = 0, 1, 2. \tag{10}$$

After A_0 and A_1 have been estimated, we can estimate C using the following equation:

$$\hat{C} = \frac{1}{m-2} Z(\hat{A}_0, \hat{A}_1), \tag{11}$$

where $Z(A_0, A_1)$ is defined as follows:

$$\begin{aligned} Z(A_0, A_1) = {} & S_{22} + A_1 S_{11} A_1^T + A_0 S_{00} A_0^T - S_{21} A_1^T \\ & - S_{20} A_0^T + A_1 S_{10} A_0^T - A_1 S_{12} - A_0 S_{02} + A_0 S_{01} A_1^T . \end{aligned} \tag{12}$$

We learn the parameters A_0, A_1 and B by applying the aforementioned technique on the data collected in the learning stage of the system. The eight parameters are treated as independent variables, allowing us to estimate the motion model parameters with fewer observation frames than would be required in the fully-coupled eight-dimensional case. In our case, the minimum number of observation frames required for learning is four. However, more robust performance can be achieved by considering more frames. In experiments, the best results were achieved with $n = 8$ to 30 frames. Hence for a real-time NTSC video stream, learning takes less than one second.

5.6 Histogram Adaptation

Adaptive histograms combine predictions and observations. In our system, color histograms are first normalized to obtain estimates of the actual probability density functions of the skin and background distributions at hand. Updates to histogram bins are made via the following model:

$$H_{i,j,k}(t) = (1-a) H_{i,j,k}(t-1) + (a) H^{(p)}_{i,j,k}(t) \tag{13}$$

where i, j, and k designate the bin under consideration and a is a scalar between 0 and 1 that allows us to adjust the speed of adaptation. The histogram $H^{(p)}$ is predicted by the second-order Markov model as described above. Optimal values of the adaptation parameter a can be determined empirically, as discussed in Sec. 7.

6 Prediction and Tracking

The prediction-tracking phase is an extension of the learning phase with one additional construct: the prediction module. This module predicts the future deformations that the distribution will undergo, and hence makes it possible to segment the future frame with a more accurate estimate.

The predicted changes in the translation, rotation, and scaling of the distribution are propagated by warping all color vectors making up the histogram distribution, and then re-sampling it. The new re-sampled distribution is then used to segment the next frame, instead of the previous observation as was done in the learning phase of the system. The rest of the system performs the same as before, as shown in Fig. 7.

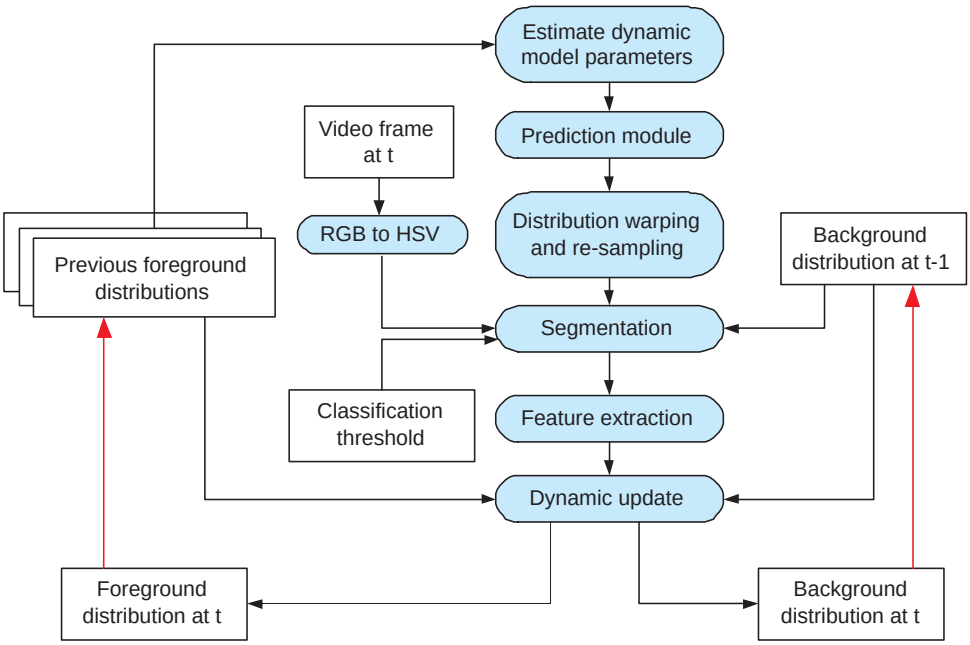

Figure 7: Prediction/tracking stage of the algorithm.

6.1 Evolution of Dynamical Model

It is reasonable to assume that not only can a distribution evolve over time, but in addition the process that guides the evolution may change also. This is especially true for long sequences where various illumination changes are expected. In order to handle this, we re-train the motion model as new data becomes available. We always use the last n frames to learn the motion model, hence at any given time t the model will be extracted from $(t-n-2, \ldots, t-2)$ frames inclusively. Frames $t-1$ and t define the parameter state vector, and are used to predict the future parameters.

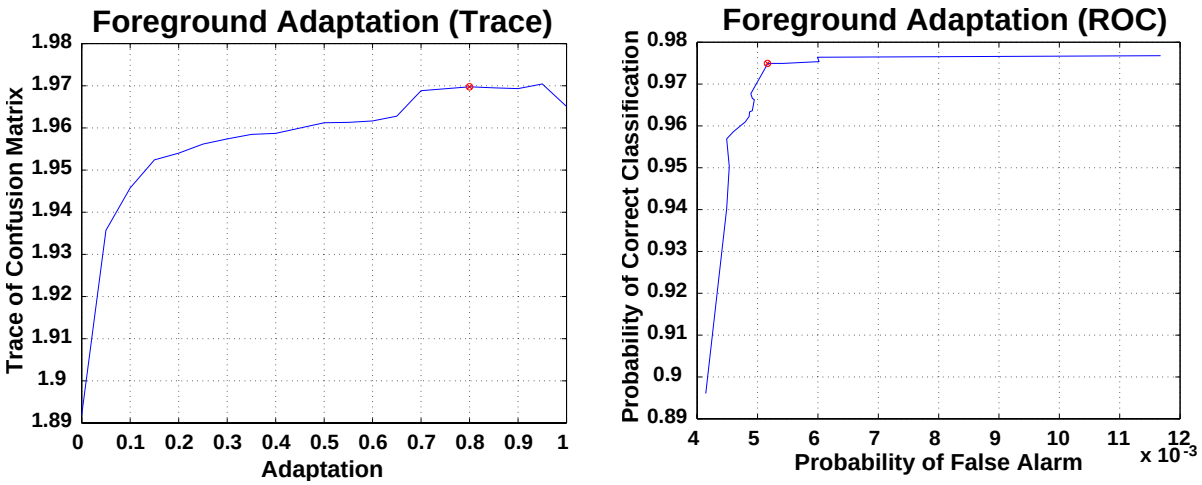

Figure 8: Performance as a function of the foreground histogram adaptation factor a_{fg}. The left graph plots the trace of the confusion matrix. The right graph shows the ROC curve. Graphs show the average performance over three 75 frame learning sequences. The optimal point is shown in red.

7 Finding Optimal Adaptation Coefficients

As described in Eq. 13, each adaptive histogram has a single adaptation parameter $a \in [0, 1]$ that controls the adaptation speed. An adaptation coefficient of $a = 0$ corresponds to a fully non-adaptive histogram, whereas $a = 1$ yields a memoryless histogram representation that is fully-adaptive. Since we have two histograms that we use for two corresponding classes, there are two adaptation parameters that must be estimated, a_{fg} and a_{bg}, for our system. These parameters can be determined empirically, as is demonstrated in the following example.

We establish the optimal foreground adaptation by fixing the background at $a_{bg} = 0$ and varying a_{fg} over its entire effective range while recording the results of segmentation on each of the three 75 frame learning sequences. The resulting segmentation is then compared with the hand labeled ground truth data in order to evaluate the performance. Performance is evaluated using two criteria: the trace of the confusion matrix [26, 27] and a receiver operator characteristic (ROC) curve. The trace of the confusion matrix measures the overall classification performance of the classifier, under the assumption that the misclassification penalty is the same for all classes. In our case of the two-class classifier, the trace measure can range from 0 to 2. Higher values of the trace correspond to better overall classification.

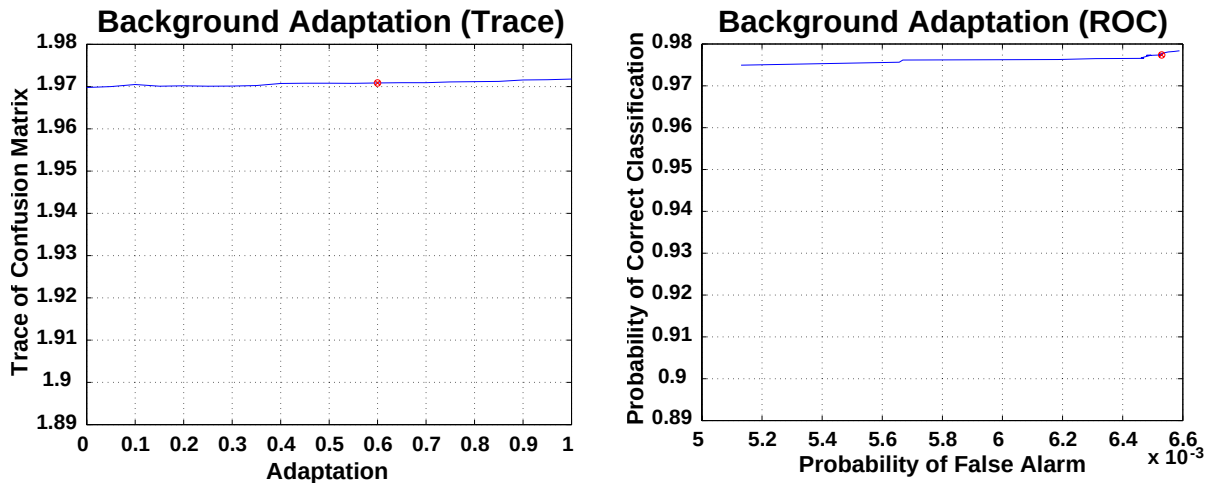

Figure 9: Performance as a function of the background histogram adaptation factor a_{bg}. The left graph plots the trace of the confusion matrix. The right graph shows the ROC curve. Graphs show the average performance over three 75 frame learning sequences. The optimal point is shown in red.

Fig. 8 shows the result of the experiment described. The adaptation coefficient a_{fg} varies between 0 and 1 by a constant delta of 0.05. The first graph shows the trace of the confusion matrix as a_{fg} varies. As can be seen in the graph, the trace is maximized in the region between $a_{fg} = 0.7$ and $a_{fg} = 0.95$. The second graph shows the effects of changing the foreground adaptation coefficient on the ROC curve. The choice of $a_{fg} = 0.8$ was made in light of both measures. The ROC curve confirms that $a_{fg} = 0.8$ maximizes the true positive rate while minimizing the false positives.

In order to find the optimal adaptation for the background we fix the $a_{fg} = 0.8$ and repeat the procedure varying the values for a_{bg}. Fig. 9 shows the two performance curves that were constructed to evaluate the performance of the system at each of the tested values for a_{bg}. The graphs are essentially flat. This can be explained in terms of the training set, which consists of sequences with only very slowly moving background. In general, however we want to be able to handle faster varying backgrounds, and hence we pick a reasonable adaptation value of $a_{bg} = 0.60$.

Two observations arise from this empirical study. First, adaptation of the foreground is more significant than that of the background, which agrees with intuition. The person in front of the camera usually moves much faster than the background; thus, the foreground tends to experience a much greater variation in its color distribution changes, and hence requires a more adaptive model. Second, even though segmentation using adaptive histograms performs better than static segmentation ($a_{fg} = 0$), the fully-adaptive ($a_{fg} = 1$) setup is not ideal. One reason for this is

noise that is present in the segmentation process as well as in the input. The semi-adaptive system suggested by the empirical study ($a_{fg} = 0.8, a_{bg} = 0.60$) tends to be more robust.

8 Experiments with Theatrical Videos

To evaluate the performance of our system we collected a set of 21 video sequences from nine popular movies. The sequences were chosen to span a wide range of environmental conditions. People of different ethnicity and various skin tones are represented. Some scenes contain multiple people and/or multiple visible body parts. Collected sequences contain scenes shot both indoors and outdoors, with static and moving camera. The lighting varies from natural light to directional stage lighting. Some sequences contain shadows and minor occlusions. Collected sequences vary in length from 50 to 350 frames; most, however, are in the 70 to 100 frame range. Fig. 10 shows example frames from the collected sequences.

Figure 10: Examples frames from sequences used for experimentation.

All experimental sequences were hand-labeled to provide ground truth data for algorithm performance evaluation. Every fifth frame of the sequences was labeled. For each labeled frame, the human operator created one binary image mask for skin regions and one for non-skin regions (background). Boundaries between skin regions and background, as well as regions that had no clearly distinguishable membership in either class were not included in the masks and are considered *don't*

care regions. The segmentation of these regions was not counted during the experimentation or evaluation of the system. Fig. 11 shows one example frame and its ground-truth labeling.

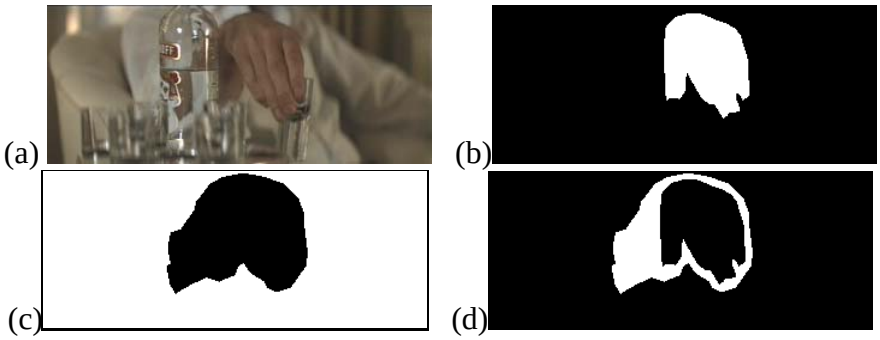

Figure 11: Example of a labeled ground truth frame: (a) original image from a sequence in which a hand is shown reaching to lift a drinking glass, (b) corresponding labeled ground truth mask image for skin, (c) background, and (d) *don't care* regions. Boundaries between skin regions and background, as well as regions that had no clearly distinguishable membership in either class were not included in the masks and are considered *don't care* regions.

8.1 Performance Experiments

The performance of the system was evaluated using the trace of the confusion matrix criterion. The trace of the confusion matrix was computed for every hand-labeled frame of the sequence. To gain an aggregate performance metric for the sequence, the average trace of the confusion matrix was computed.

For comparison, we measured the classification performance of a standard static histogram segmentation approach [6] on the same data set. The static histogram approach implemented used the same prior histograms and threshold as our adaptive system (see Sec. 4.2). The same binary image processing operations of connected component analysis, size filtering, and hole filtering were performed to achieve a fair comparison.

The performance results are outlined in Table 1. Three performance measures were computed: correct classification of skin pixels, correct classification of background pixels, and the trace of the confusion matrix $Tr[C]$. With respect to the $Tr[C]$ measure, out of 21 sequences considered, 17 performed better using our dynamical approach. An increase in performance of up to 24% was observed. A performance increase of over 10% was observed on four sequences. Skin classification rates with dynamic histograms were as good or better than the static histogram approach in all cases. In all but one case the skin-color classification rate was higher – in two cases by as much

Sequence Info			Classification Performance					
			Static			*Dynamic*		
#	*# frames*	*# of skin* color blobs	*skin*	*bg*	*Tr[C]*	*skin*	*bg*	*Tr[C]*
1	100	1	49.08	99.96	1.49	65.74	99.35	1.65
2	72	1	96.46	99.99	1.96	98.19	99.73	1.98
3	72	1	77.21	91.62	1.69	88.92	86.43	1.75
4	110	1	92.67	99.73	1.92	97.63	99.13	1.97
5	75	1	96.87	99.86	1.97	98.30	99.66	1.98
6	72	1	88.32	99.23	1.88	94.27	99.14	1.93
7	76	1	77.67	100.00	1.78	91.30	100.00	1.91
8	73	1	99.99	98.72	1.99	99.98	97.17	1.97
9	72	1	81.30	99.62	1.81	92.81	100.00	1.93
10	73	1	96.00	36.56	1.33	99.96	15.72	1.16
11	233	1	87.47	99.93	1.87	93.99	99.59	1.94
12	72	2	70.51	97.49	1.68	62.36	95.94	1.58
13	350	2	67.73	99.96	1.68	82.21	99.71	1.82
14	72	2	91.79	99.98	1.92	98.90	97.73	1.97
15	75	2	91.03	95.37	1.86	94.10	90.18	1.84
16	50	2	52.05	100.00	1.52	88.95	99.82	1.89
17	75	2	97.89	99.98	1.98	99.43	99.66	1.99
18	91	2	92.07	99.99	1.92	98.60	99.94	1.99
19	73	2	11.02	99.91	1.11	24.29	99.48	1.24
20	120	3	18.75	100.00	1.19	55.79	90.95	1.47
21	53	4	92.96	98.42	1.91	97.94	95.15	1.93
Average			77.56	96.01	1.73	86.84	93.55	1.80

Table 1: Table of performance figures for the 21 different video sequences from popular DVD movies. The experiments compared classification accuracy for the dynamic *vs.* static histogram approach. Three performance measures were computed: correct classification of skin pixels, correct classification of background pixels, and the trace of the confusion matrix $Tr[C]$.

as 37%. In nearly all cases, background classification rates were comparable to those of static segmentation. Examples of successful performance can be seen in Fig. 12.

Two out of four sequences that failed to perform better, had an insignificant performance loss. In the other two failure cases, the system performance loss was around 10%. This performance degradation was due to skin-like color patches appearing in the background of initial frames of a sequence, as can be seen from Fig 13. Recall that these initial frames are used in estimating the parameters of the Markov model (Sec. 5).

Finally, we performed a set of experiments to establish system stability over time. For example, the graph in Fig. 14 shows system performance on the longest sequence in our test set (349 frames).

As can be seen from the graph in Fig. 14, the dynamic approach was consistently better than the

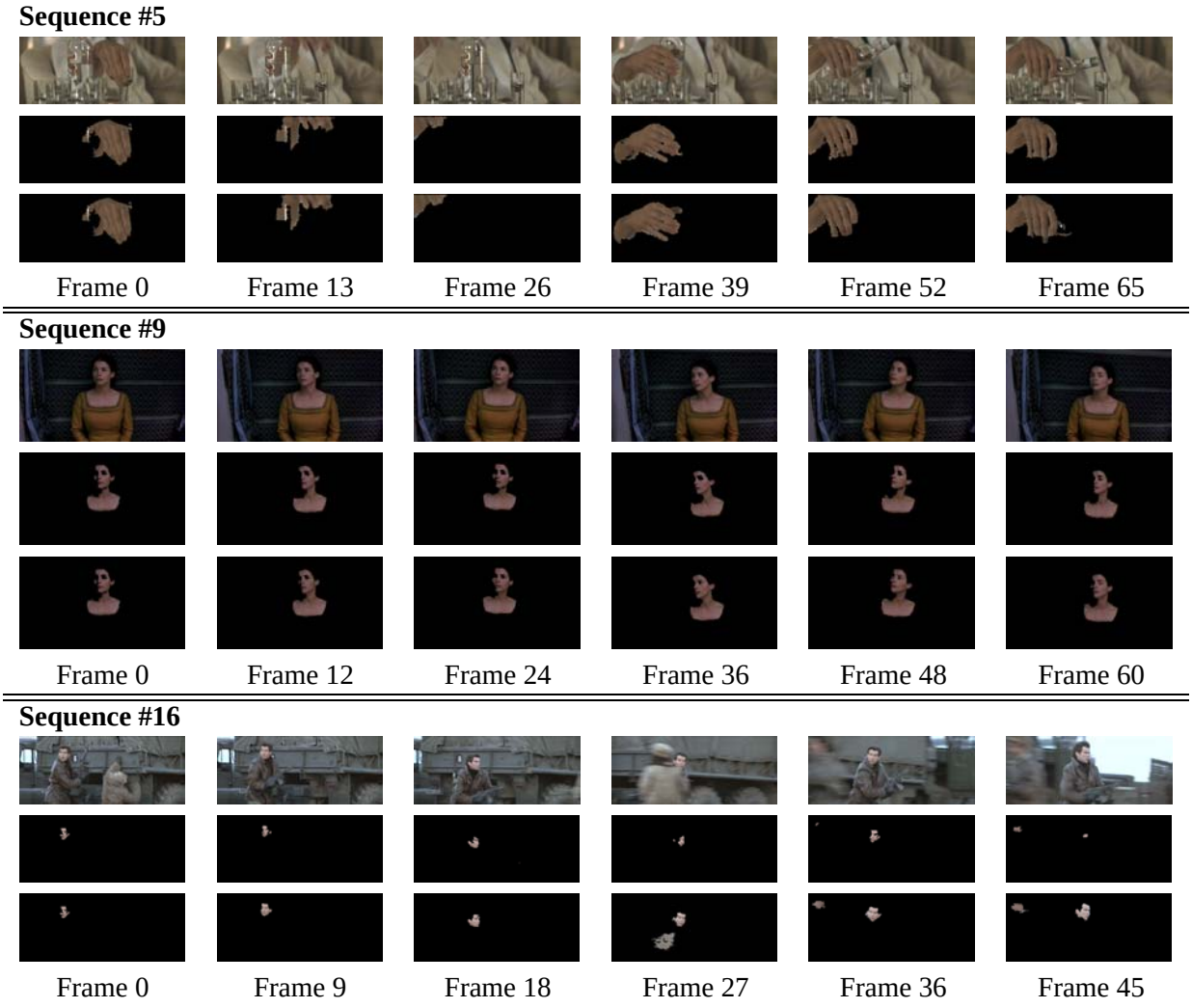

Figure 12: Examples of segmentation obtained by the dynamic and static approaches. The examples above are among those for which our system had a superior performance. Frames from image sequence (top), static segmentation (middle), and dynamic segmentation (bottom) are shown.

static method in classifying skin and background pixels. Not only does our system perform over 10% better for the entire sequence, it is also more stable. The standard deviation of performance for our system was measured to be 0.0314, which is almost half of the standard deviation of 0.0589 measured for the static segmentation approach. It should be noted that the stability of our system was consistent across experiments.

The adaptation coefficients $a_{fg} = 0.8$ and $a_{bg} = 0.6$ were determined once off-line for a given training set as described in Sec. 7. The adaptation coefficients remained fixed across all trials. In all of the above experiments, $n = 20$ frames were used for learning of the motion model.

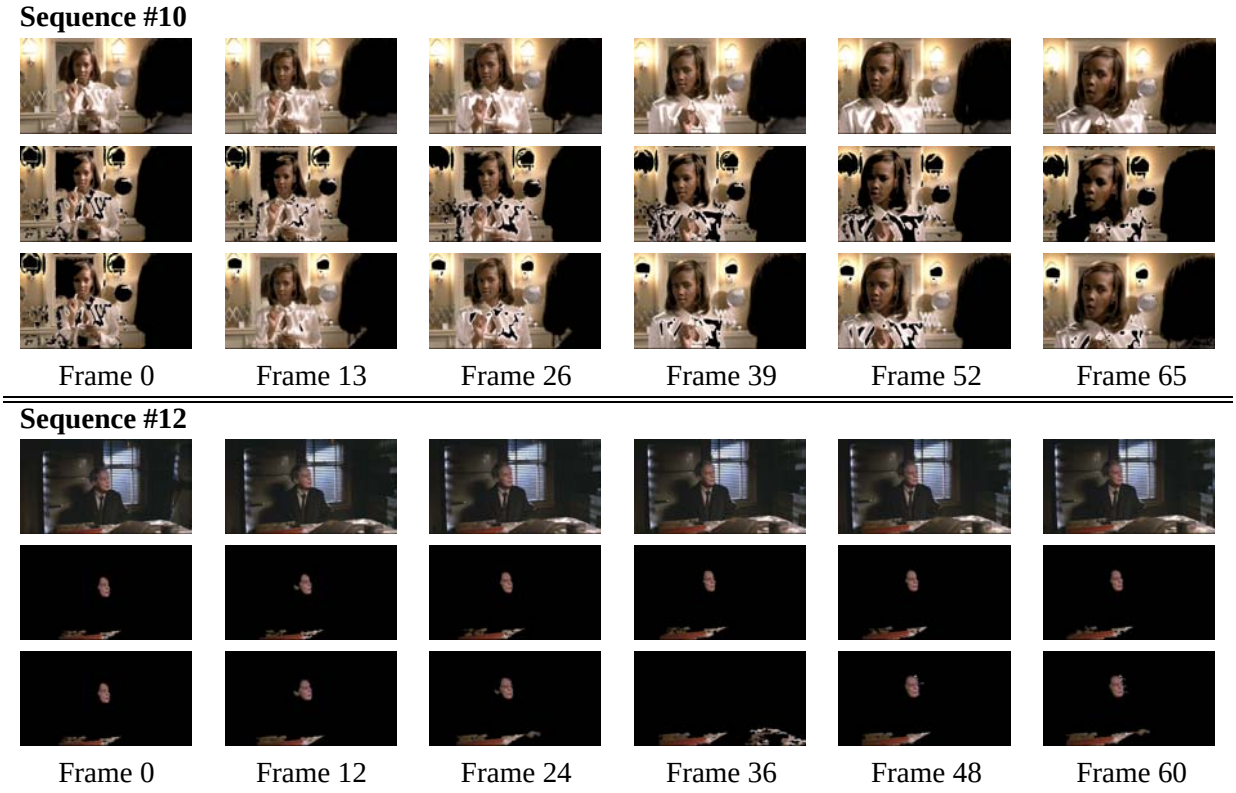

Figure 13: Two experimental sequences where our algorithm performed slightly worse than the static histogram approach. Frames from image sequence (top), static segmentation (middle), and dynamic segmentation (bottom) are shown. This performance degradation was likely due to skin-like color patches appearing in the background of the initial frames of the sequence.

9 Experiments with Indoor Live Video

In addition to evaluating the performance of our system on the sequences collected from popular movies, we also conducted a number of experiments using a live video camera. Example frames from these test sequences and the corresponding segmentation for each frame are shown in Figs. 15–17. In these experiments we tried to stage some of scenarios that motivated this work.

Fig. 15 shows frames from the first test sequence: a video of a person walking down a corridor that is illuminated by widely-spaced fluorescent lights. As can be seen in the figure, the observed skin color distribution evolves as the person's distance and the angle to the illuminant changes. It is evident that our dynamic segmentation algorithm performs consistently better than static segmentation throughout the sequence for this test case.

The next two experiments utilized changing colored illumination. Fig. 16 shows an experiment where a person is illuminated by two different light sources: the room's ambient fluorescent light,

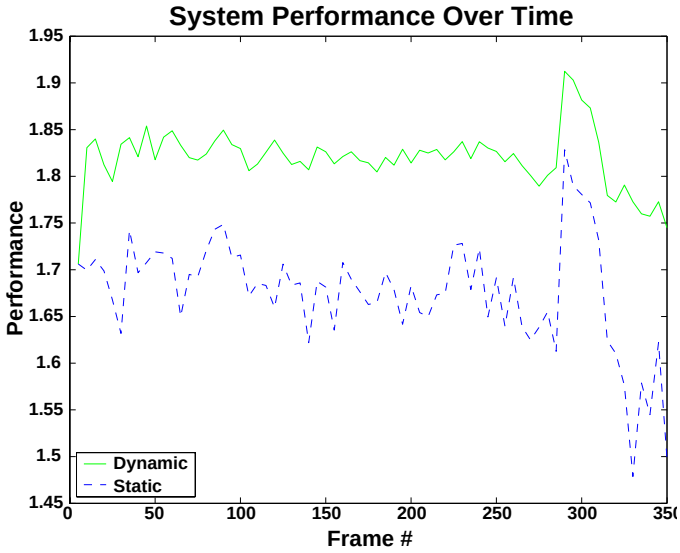

Figure 14: Performance of the dynamical system over an extended sequence. The horizontal axis represents time, measured in frames. The vertical axis represents the performance measured by the trace of the confusion matrix. The dotted line corresponds to the performance of the static histogram segmentation, and the solid line to our dynamic approach.

and a directed green light that is oriented towards the face of the subject. As the subject walks toward the directed green light source, the effects of the green tint can clearly be seen on the left side of the face and the neck. As shown in Fig. 16 the static segmentation fails to segment the tinted regions. In contrast, the dynamic segmentation performs considerably better in these regions, segmenting most of the face reliably.

Our last example test sequence is shown in Fig. 17. In this experiment we used a sequence of a moving hand and arm, to emphasize that our system is not biased to the face skin color or geometry. In this case there as blue directed lighting, in addition to the room's ambient fluorescent component. As before, the dynamic segmentation produces considerably better results than could be obtained via a static approach.

10 Discussion

In our experiments it was observed that an affine transformation is a good approximation to the motion of skin-color distributions which arises when tracking a single person in an image, with global illuminant changes. More specifically we observed that translation and scaling parameters were most useful in tracking the skin-color distribution over time. In the sequences we have tested,

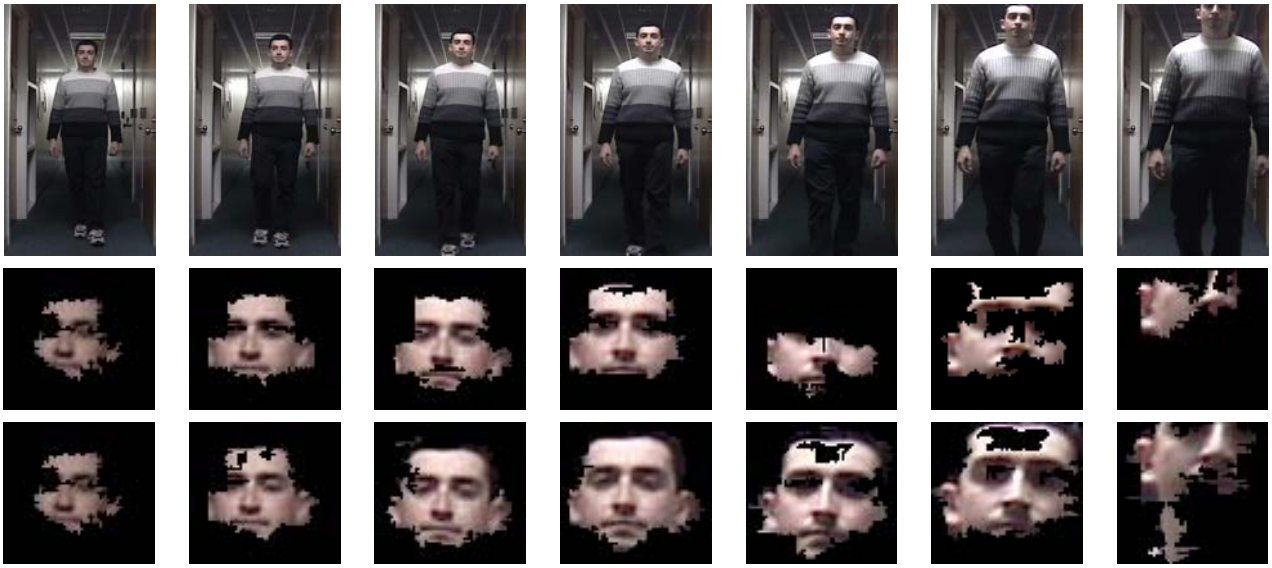

Figure 15: Experimental sequence of a person walking down a corridor illuminated by widely spaced fluorescent lights. Frames from image sequence (top), static segmentation (middle), and dynamic segmentation (bottom) are shown. Improved segmentation can be observed in regions where illuminant changes are most prevalent, for example nose and the forehead.

rotation did not seem to be a significant factor. To this end we believe that one could exclude rotation from the model, and gain better computational performance with only negligible performance loss. Quantitative validation of this remains for future study.

In general however, a single affine transformation may not always be sufficient to model the skin-color distribution for multiple people, or accounting for localized illuminant changes. In some multi-modal cases, our system works well, as is shown in Fig. 18. However, in general such cases will require a much more elaborate deformation model. For instance, we could employ polynomial transformations computed from higher order statistics, or use multiple affine color-trackers (one per unique region/patch for example). Both of these approaches are reasonable and straight-forward extensions of our system.

In our system, we assume that there are enough sample pixels to provide a good sampling of the underlying distribution. This is a reasonable assumption given that skin-color pixels of any particular person are closely clustered in HSV color space [22, 24]. In addition, the recursive nature of the adaptive histogram algorithm allows the use of samples from more than one frame, thereby increasing the number of samples used in estimating the distribution at any given time. To further ensure that we have a good sampling for the skin-color distribution we apply a $3 \times 3 \times 3$

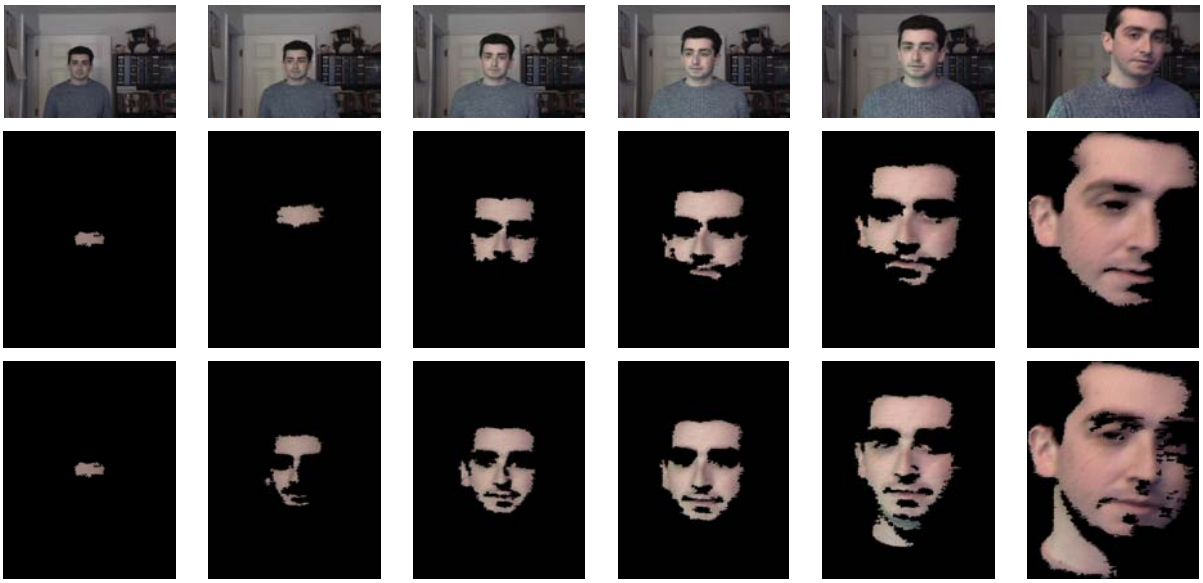

Figure 16: Experimental sequence of a person walking in an environment illuminated by colored lights. The environment contains a fluorescent ambient component, and a green directed light aimed towards the face of the subject. The subject is walking towards the directed green light source. Frames from image sequence (top), static segmentation (middle), and dynamic segmentation (bottom) are shown. Notice that in frames 60, 80, and 100 where the effects of the directional green light source are most noticeable, the dynamic system is still able to reliably segment those regions.

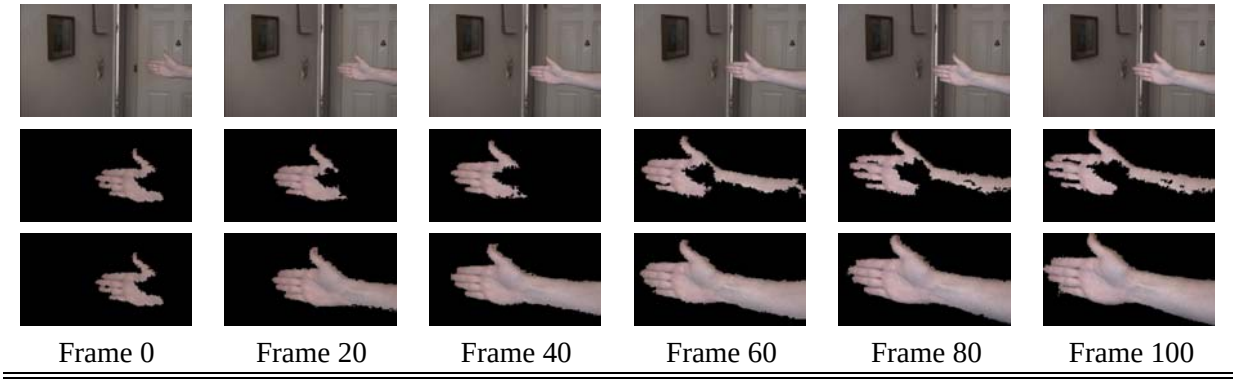

Figure 17: Experimental sequences of tracking a hand in an environment that contains a fluorescent ambient component, and a directed blue light facing towards the arm of the subject. Frames from image sequence (top), static segmentation (middle), and dynamic segmentation (bottom) are shown.

Gaussian smoothing over the histogram with a very small σ.

In general we noticed that the final result of our algorithm depends greatly on the initialization phase. If the algorithm is initialized with an over-segmented region it generally performs much worse than if it is initialized with an under-segmented version of the same image. This is due to the way adaptation works. In general adaptation facilitates bounded region growing. In addition, due

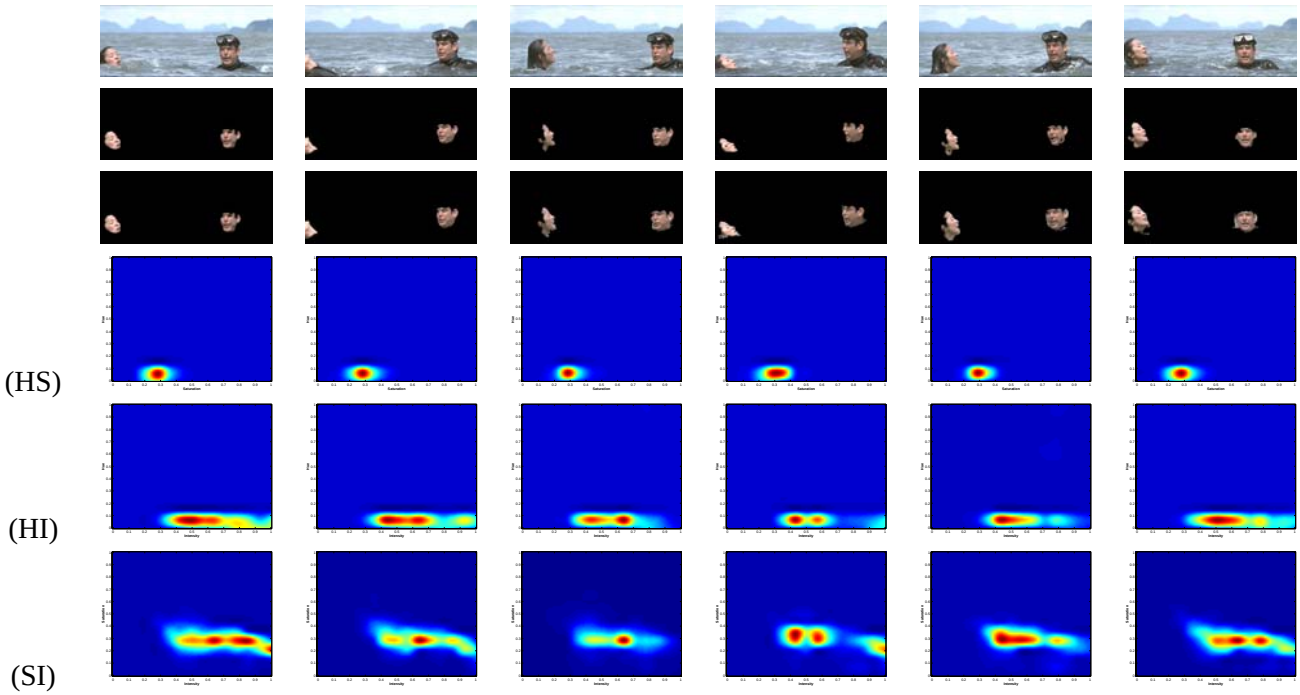

Figure 18: Example of a dynamic skin-color distribution in HSV space that exhibits multiple modes. The top row shows images taken from a 75 frame long video sequence, sampled equally in time. The segmentation obtained with the static segmentation algorithm (second row) and the dynamic segmentation (third row) are shown below each image. The next three rows show the 2D projection views of the skin-color distribution in HSV color space for each video frame: (HS) Hue-Saturation, (HI) Hue-Intensity, and (SI) Saturation-Intensity plane projections.

to its adaptive nature our algorithm is also more susceptible to the background skin-color patches that may appear due to the motion blurring effects (see Sequence #16 in Fig. 12) or specular highlights on the background objects. Initialization and subsequent segmentation accuracy could be further improved via the use of shape and blob-based motion constraints [7], and/or domain-specific constraints like face detection [30].

Furthermore, in our experiments the foreground adaptation had a much greater impact on the final system performance, as opposed to the background adaptation. This was true even for sequences with slowly varying backgrounds. It has been observed that for many sequences one can get away with a very inadaptive background distribution, while maintaining almost the same error rates, as long as foreground adaptation stays the same.

Scene changes are not explicitly modeled by our system; however the system can account for slowly changing dynamic scenes due to the nature of the algorithm. Dramatic or abrupt changes in the background, will cause significant performance loss. As a possible future extension to the

system we are considering automatic re-initialization based on the threshold for the magnitude of change in the background and foreground distributions.

11 Conclusion

In this paper we have developed a novel approach for real-time skin segmentation in video using color. The approach enables robust segmentation of skin-colored patches despite dynamic illumination conditions. We have quantitatively tested the performance of our system on 21 test sequences, with hand-labeled ground truth, obtained from popular movies. The sequences contained a good variety of indoor and outdoor scenes, with one and two actors and a wide range of motions and illumination changes. The performance of our algorithm was compared to the segmentation obtained using a static color model. An overall increase in performance in 17 out of 21 test sequences was observed, sometimes by as much as 24%. In all but one case the skin-color classification rates for our system were higher, with background classification rates comparable to those of the static segmentation. The system was also tested on live video sequences collected indoors, with dynamic and colored illumination. Dynamic segmentation performs considerably better than the static approach under changing illumination conditions, and gives comparable performance when illumination changes are insignificant.

In our implementation we assumed that the parameters of the skin-color distribution were independent, hence allowing us to use diagonal matrices for A_0, A_1, and B in our dynamical model. The experiments that we conducted support this independence assumption to a large extent; we have not seen any significant correlations between the deformation parameters of the skin-color distributions. However, further experiments are needed to establish this definitively. If correlations exist, then changing the implementation to estimate dependent variables in pairs or triplets may further enhance the performance of the system. Note that this change will not require any changes to the formulation, but rather only a small change in implementation of the relevant matrices.

In this paper we used HSV, as the color-space for tracking the evolution of the color distributions in time. This choice of the color space was motivated by [23] and by our own empirical studies. Due to the large number of the color spaces available, we were not able to experiment with all of them. Latest work by [28], suggests that CIE-xy is another promising color space in terms of background separability for skin-color segmentation. Additional studies would be needed

to determine if the skin color evolution in CIE-xy space is well-behaved and can be effectively modeled using the dynamical model presented in this paper.

Finally, application-specific information such as human motion models [7] or geometric/spatial constraints on the regions being tracked [1, 20, 22] can be incorporated in our system to further improve performance. For example, our tests have shown that using a face detector [30] to bootstrap initialization, considerably boosts the performance of our system.

Acknowledgments

The authors would like to thank Michael Black for helpful discussions. This work was supported in part through ONR Young Investigator Award N00014-96-1-0661, and NSF grants IIS-9624168 and EIA-9623865.

References

[1] S.T. Birchfield. Elliptical head tracking using intensity gradients and color histograms. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 232–237, 1998.

[2] A. Blake and M. Isard. Active Contours. Cambridge U. Press, 1998.

[3] A. Blake, M. Isard, and D. Reynard. Learning to track the visual-motion of contours. *Artificial Intelligence*, 78(1-2):179–212, 1995.

[4] T. Darrell, G.G. Gordon, M. Harville, and J. Woodfill. Integrated person tracking using stereo, color, and pattern detection. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 601–607, 1998.

[5] W. Hafner and O. Munkelt. Using color for detecting persons in image sequences. *Pattern Recognition and Image Analysis*, 7(1):47–52, 1997.

[6] M.J. Jones and J.M. Rehg. Statistical color models with application to skin detection. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, I:274–280, 1999.

[7] N. Oliver, A.P. Pentland, and F. Berard. Lafter: Lips and face real time tracker. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 123–129, 1997.

[8] Y. Raja, S.J. McKenna, and S. Gong. Tracking and segmenting people in varying lighting conditions using colour. *Proc. International Conf. on Automatic Face and Gesture Recognition*, pp. 228–233, 1998.

[9] S. Singh and N. Papanikolopoulos. Vision-based detection of driver fatigue. Technical report, Dept. of Comp. Sci., Univ. of Minnesota, 1997.

[10] L. Sigal, S. Sclaroff, and V. Athitsos Estimation and prediction of evolving color distributions for skin segmentation under varying illumination. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, II:152–159, 2000.

[11] C.R. Wren, A. Azarbayejani, T.J. Darrell, and A.P. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Trans. on Pattern Anal. and Machine Intelligence*, 19(7):780–785, July 1997.

[12] J. Yang, Lu Weier, and A. Waibel. Skin-color modeling and adaptation. *Proc. Asian Conf. on Computer Vision*, II:687–694, 1998.

[13] M.H. Yang and N. Ahuja. Detecting human faces in color images. *Proc. International Conf. on Image Processing*, pp. 127–139, 1998.

[14] D. Saxe and R. Foulds. Toward robust skin identification in video images. *Proc. International Conf. on Automatic Face and Gesture Recognition*, pp. 379–384, 1996.

[15] R. Kjeldsen and J. Kender. Finding skin in color images. *Proc. International Conf. on Automatic Face and Gesture Recognition*, pp. 312–317, 1996.

[16] M. Storring, H.J. Andersen, and E. Granum. Skin colour detection under changing lighting conditions. *Proc. Seventh Symposium on Intelligent Robotics Systems*, pp. 187–195, 1999.

[17] Sang-Hoon Kim, Nam-Kyu Kim, Sang Chul Ahn, and Hyoung-Gon Kim. Object oriented face detection using range and color information. *Proc. International Conf. on Automatic Face and Gesture Recognition*, pp. 76–81, 1998.

[18] J.C. Terrillon, M. David, and S. Akamatsu. Automatic detection of human faces in natural scene images by use of a skin color model and of invariant moments. *Proc. International Conf. on Automatic Face and Gesture Recognition*, pp. 112–117, 1998.

[19] K. Imagawa, S. Lu, and S. Igi. Color-based hands tracking system for sign language recognition. *Proc. International Conf. on Automatic Face and Gesture Recognition*, pp. 462–467, 1998.

[20] K. Schwerdt and J.L. Crowley. Robust face tracking using color. *Proc. International Conf. on Automatic Face and Gesture Recognition*, pp. 90–95, 2000.

[21] M. Storring, H.J. Andersen, and E. Granum. Estimation of the illuminant colour from human skin colour. *Proc. International Conf. on Automatic Face and Gesture Recognition*, pp. 64–69, 2000.

[22] X. Zhu, J. Yang, and A. Waibel. Segmenting hands of arbitrary color. *Proc. International Conf. on Automatic Face and Gesture Recognition*, pp. 446–453, 2000.

[23] J.C. Terrillon and S. Akamatsu. Comparative performance of diffeent chrominance spaces for color segmentation and detection of human faces in complex scene images. *Proc. Vision Interface*, pp. 180–187, 1999.

[24] M.H. Yang, N. Ahuja. Gaussian mixture model for human skin color and its application in image and video databases. *Proc. SPIE Conf. on Storage and Retrieval for Image and Video Databases*, pp. 458–466, 1999

[25] Y. Wu, T.S. Huang Color tracking by transductive learning *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, I:133–138, 1999.

[26] T. Hastie, R. Tibshirani, and J. Friedman The elements of statistical learning; data mining, inference, and prediction. *Springer Series in Statistics*, Springer-Verlag, 2002.

[27] K. Fukunaga. Introduction to Statistical Pattern Recognition. second edition. Morgan Kaufmann, 1990.

[28] J.C. Terrillon, Y. Niwa, and K. Yamamoto On the selection of an efficient chrominance space for skin color-based image segmentation with an application to face detection. *In Proc. International Conf. on Quality Control by Artificial Vision*, Cpadus ditions. Vol.2, pp. 409–414, 2001.

[29] F.L. Bookstein Thin-plate splines and the decomposition of deformations. *IEEE Trans. on Pattern Anal. and Machine Intelligence*, 11(6):567–585, 1989.

[30] R. Rosales, V. Athitsos, L. Sigal, and S. Sclaroff 3D hand pose reconstruction using specialized mappings. *In Proc. International Conf. on Computer Vision (ICCV)*, Vol.1, pp. 378–385, 2001.

[31] M. Soriano, B. Martinkauppi, S. Huovinen, and M. Laaksonen. Skin detection in video under changing illumination conditions. *In Proc. International Conf. on Pattern Recognition*, Vol.1, pp. 839–842, 2000.

[32] P. Smith, M. Shah, and N. Lobo. Monitoring Head/Eye Motion for Driver Alertness with One Camera. *In Proc. International Conf. on Pattern Recognition*, 2000.

[33] X. Liu, F. Xu and K. Fujimura. Real-Time Eye Detection and Tracking for Driver Observation Under Various Light Conditions. *In IEEE Intelligent Vehicle Symposium*, 2002.