Tracking Human Faces in Real-Time

Jie Yang   Alex Waibel

November, 1995
CMU-CS-95-210

School of Computer Science
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

This research was sponsored by the Advanced Research Projects Agency under the Department of the Navy, Naval Research Office under grant number N00014-93-1-0806. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies or endorsements, either expressed or implied, of the Department of the Navy, or the U.S. government.

Keywords: Face tracking, color modeling, color adaptation, motion estimation, camera control, real-time tracking.

Contents

1 Introduction 1

2 Problem Description 4

3 A Stochastic Model for Tracking Faces 8

3.1 Skin-Color Model 8

3.2 Skin-Color Model Adaptation 13

4 A Real-Time Face Tracker 15

4.1 Skin Color Model for Face Locating 16

4.2 Motion Estimation and Prediction 19

4.3 Model-based Camera Control 20

4.4 System Initialization 22

4.5 Tracking Faces in Real-time 23

5 Conclusion 25

Acknowledgment 26

26

References 26

List of Figures

Figure 1	Overview of a face tracker....4
Figure 2	Three views of a person's face....6
Figure 3	The chromaticity diagram....9
Figure 4	An example of face and analyzed area....10
Figure 5	The color distribution of a human face in chromatic color space....10
Figure 6	Skin-color distribution cluster of different people....11
Figure 7	The color distribution for different lighting condition and different persons 12
Figure 8	The color distribution generated by the model....12
Figure 9	The face tracker setup....15
Figure 10	An example of locating face by the skin-color model....16
Figure 11	An example of background with skin colors, (a) original image (copyright of 1994 Smithsonian Institution); (b) color matching result17
Figure 12	Locating faces in a scene with skin-colored background....18
Figure 13	Locating faces using a combination of color, geometry, and motion information19
Figure 14	Search window....20
Figure 15	Camera control scheme: (a) conventional feedback control; (b) model-based predictive control21
Figure 16	Pinhole camera model....22
Figure 17	An interactive interface to start the face tracker....23
Figure 18	Some real-time tracking results (the bigger white square in each image is the search window and the smaller white square is the located face)24

Abstract

Many applications in human computer interaction (HCI) require tracking a human face. In this report, we address two important issues for tracking human faces in real-time: what to track and how to track. We present a stochastic model to characterize skin-colors of human faces. The information provided by the model is sufficient for tracking a human face in a various poses and views. The model can be adapted in real time for different people and different lighting conditions while a person is moving. We then present a model-based approach to implement a real-time face tracker. The system has achieved a rate of up to 30+ frames/second using an HP-9000 workstation with a framegrabber and a Canon VC-C1 camera. It can track a person's face while the person moves freely (e.g., walks, jumps, sits down and stands up) in a room. Three types of models have been employed to track human faces. In addition to the skin-color model used to register the face, a motion model is used to estimate image motion and to predict search window; and a camera model is used to predict and to compensate for camera motion (panning, tilting, and zooming). The system can be applied to tele-conferencing and many human-computer interactive applications such as lip-reading and gaze tracking. The principle in developing this system can be extended to other tracking problems such as tracking the human hand for gesture recognition.

1 Introduction

The face provides a variety of different communicative functions such as identification, the perception of emotional expressions, and lip-reading. Many applications in human computer interaction require tracking a human face. For example, in a tele-conference, it is desirable to allow the participants to move freely while a face tracker tracks the current speaker. In a multimodal human computer interface, a face tracker plays many important roles ranging from direct user interaction to the processes that are embedded in larger systems.

Many current speech recognition systems perform well on clean speech signals but perform poorly on noisy signals. Integration of acoustic and visual information (automatic lip-reading) can improve overall recognition rate especially in noisy environments. Early work shows that recognition rates increase by incorporating images of the speaker's mouth into a traditional speech recognition system [1]. A reliable face and mouth tracker could provide the tracking necessary for such a "lipreading" system. It has been shown that a more accurate localization in space can be delivered visually than acoustically [2].

Gaze tracking can provide computer with the information where a person is looking, and what he/she is paying attention to. This information provides communication cues to a multi-modal interface. Such information can be obtained from tracking the orientation of a human head, or gaze. While current approaches to gaze tracking tend to be highly intrusive - the subject must either be perfectly still, or wear a special device -- a face tracker makes it possible to develop a more flexible system using computer vision technology.

Facial action analysis uses a coding system to describe facial deformations, such as "eye brow lowered" or "lips puckered" [3]. Psychology researchers use the coding system to describe a subject's behavior and then interpret the resulting code sequences in relation to the stimulus. Such a system is also interesting for the computer vision and computer graphics community because it can be applied to image coding and image synthesis. One active research topic in this area is to automatically track and extract facial action parameters [4]. An automated coder could use a real-time face tracker to locate and align key facial features for other subsystems that would perform the actual analysis.

Human face perception is currently an active research area in the computer vision community. Much research has been directed towards feature recognition in human faces. Three basic techniques are commonly used for dealing with feature variations: correlation templates [5][6], deformable templates [7], and spatial image invariants [8]. Several systems of locating human face have been reported. Eigenfaces, obtained by performing a principal component analysis on a set of faces, have been used to identify faces [9]. By moving a window covering a subimage over the entire image, faces can be located within the entire image. [10] reports a face detection system based on clustering techniques. The system passes a small window over all portions of the image, and determines whether a face exists in each window. A similar system with better results has been claimed by [11]. A different approach for locating and tracking faces is described in [12]. This system locates faces by searching for the color of skin. After locating a face, additional features matching this particular face are extracted.

In this report, we discuss the problem of tracking a human face in real-time. We focus on two important issues in tracking a face in real-time: what to track and how to track. To address the issue of what to track, we present a skin-color model in chromatic color space designed to characterize human faces. The model can be adapted in real-time for different people and different lighting conditions while a person is moving. We demonstrate that the information provided by the model is sufficient for tracking a human face in various positions and orientations. To address the issue of how to track, we present a model-based approach to implement a real-time face tracker. Visual tracking is a sequential estimation problem of recovering the time-varying state of the world given a sequence of images. Three models have been employed for tracking a human face. In addition to the skin-color model used to register the face, a motion model is used to handle head motion and a camera model is used to predict camera motion. We have developed a system that can track a person's face while the person moves freely (walks, jumps, sits and rises) in a room. The system has achieved a rate of up to 30+ frames/second using an HP-9000 workstation with a framegrabber and a Canon VC-C1 camera.

The remainder of the report is structured as follows. Section 2 discusses the general problem of tracking a human face and related work. Section 3 presents our novel skin-color model and the model adaptation algorithm. Section 4 addresses application of the skin-color model to locating human faces and methodology of developing a real-time face tracker, and shows experimental results. We close with a discussion of future work.

2 Problem Description

The tracking problem is distinguished from the recognition problem in that its search processes are local rather than global. The face locating problem is a recognition problem. In order to locate a human face, the system needs to capture an image using a camera and a framegrabber, to process the image, to search the image for important features, and then to use these features to determine the location of the face. In order to track a human face, the system not only needs to locate a face, but also needs to find the same face in a sequence of images (Figure 1). This requires the system to have the ability to estimate the motion while locating the face. Furthermore, to track faces outside a certain range the system needs to control the camera, e.g., panning, tilting, and zooming.

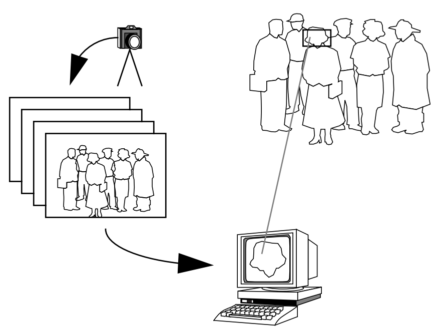


Figure 1 Overview of a face tracker

The general tracking problem can be formulated as follows: given a sequence of images $I_t(x,y)$ which were formed by locally displacing a reference image $I(x,y)$ with horizontal and vertical displacement fields, i.e.,

$$I_t(x+u_t,y+v_t) = I(x,y), \qquad \text{(EQ 1)}$$

we wish to recover the displacement fields (u_t,v_t) from the reference image $I(x,y)$. This problem is also known as motion estimation, multiple view analysis, or image registration. The problem is to estimate parameters for one of three models listed in Table 1. The complexity of motion estimation largely depends on the model used.

Table 1: Motion Parametric Models

Model	Transformation	Parameters
Translation	$X' = X + b$	$b \in R^2$
Affine	$X' = AX + b$	$A \in R^{2x2}, b \in R^2$
Projective	$X' = \frac{AX + b}{C^T X + 1}$	$A \in R^{2x2}, b, c \in R^2$

A large number of approaches have been proposed to solve this problem [13][14][15]. The approaches include optical flow (general motion) estimators [16][17][18][19], global parametric motion estimators [20][21], local parametric motion estimators [22][23], constrained motion estimators [21], stereo and multiframe stereo [24][25][26], hierarchical (coarse-to-fine) methods [21][27][29]. These approaches are either correlation-based or feature-based. Correlation-based tracking algorithms select a patch of pixels in the first image and then searching for a correspondence in the second image by optimizing a cost function. The advantage of these algorithms is that no preconception of a feature is needed. But the optimization process is often computational expensive and is hardly real-time. For feature-based tracking, many algorithms depend on finding specific kinds of features in the images and then matching correspondences between such features. A well-chosen feature can make the searching process much easier. However, finding good features is still an open problem [30]. Several factors, such as the existence of a preprocessing algorithm, the necessity, complexity and generality of the selected features, must be considered in selecting features for real-time tracking.

To locate human faces, facial features, such as eyes, nose and mouth, are natural candidates. But these features may change from time to time. Occlusion and non-rigidity are basic problems with these features. For example, when a person rotates his head, depth changes can warp or occlude facial features as illustrated by Figure 2. In Figure 2(a), the left side of a subject's face is shown; Figure 2(b) shows a frontal view of the face; and Figure 2(c) shows the right side of the subject's face. If these images are part of an image sequence, in moving from part (a) to part (b), the image of the left eye warps and the right ear appears (the inverse of occlusion); in moving from part (b) to part (c), the left ear disappears (occlusion) and the image of the right eye warps.

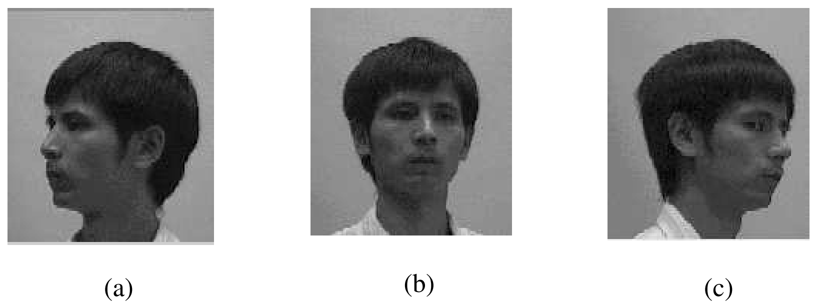


Figure 2 **Three views of a person's face**

A lot of motion estimation algorithms work only for a rigid object. But a face cannot be regarded as a rigid object because the eyes and mouth are deformable. Several methods such as model-based tracking and deformable-template matching can be used to deal with the variation of these features because of their inherent ability to modify the reference pattern. Using multiple templates for a single feature, with each template corresponding to a different view of the feature, also improves tracking performance. These methods are, however, computational expensive and hardly achieve real-time performance. For example, the system described in [11] shows good accuracy in locating a face but it takes approximately 120 seconds on a Sparc 20 to process a 160x120 pixel image with two neural networks.

Color is another feature on human faces. Using skin color as a feature for tracking a face has several advantages. First, processing color is much faster than processing other facial features. Second, under certain lighting conditions, color is orientation invariant. This property makes motion estimation much easier because only a translation model is needed for motion estimation, with only two parameters as listed in Table 1.

However, color is not a physical phenomenon. It is a perceptual phenomenon that is related to the spectral characteristics of electro-magnetic radiation in the visible wavelengths striking the retina [32]. Tracking human faces using color as a feature has several problems. First, the color representation of a face obtained by a camera is influenced by many factors such as ambient light, object movement, etc. Second, different cameras produce significantly different color values even for the same person under the same lighting condition. Finally, human skin colors differ from person to person. In order to use color as a feature for face tracking, we have to solve these problems.

Much research has been directed to understanding and making use of color information. Color has been long used for recognition [33][34][35] and recently has been successfully used for road tracking [36] and face tracking [12]. In the next section, we develop a stochastic model of skin-color in chromatic color space for tracking a human face. The model has few parameters and is easily adaptable to different people and different lighting conditions.

Another important consideration for face tracking is motion tolerance, i.e., how fast the face can move in the image. The faster the face moves, the larger the face displacement in a sequence of images can be. When the motion is slow, face locations change very little from image to image; thus the face tracker only needs to search a small potion of the image to find the face, significantly reducing the computational effort in the search. Several factors influence the face motion in the image, such as human motion, depth from the camera to the face, face size in the image, and image sampling rate. To allow a person to freely move in a large area, camera motion (panning, tilting, and zooming) can compensate for human motion. For example, if the camera moves in the same direction as the face, the relative motion of the face in the image will decrease. Using an active camera, however, makes tracking problem more challenging. Most approaches for egomotion analysis are based on optical flow [37][38]. These techniques are useful for image analysis but not for real-time tracking because the analysis is based on the information that egomotion imposed on the image, i.e., there is a time-delay between camera motion and the motion that appears in the image. For real-time tracking, motion prediction is desirable.

We will present a model-based approach to tracking a human face in real-time: a skin-color model for characterizing the face, a motion model for estimating and predicting image motion, and a camera model for predicting and compensating for the camera motion.

3 A Stochastic Model for Tracking Faces

3.1 Skin-Color Model

Color is the perceptual result of light in the visible region of the spectrum, having wavelengths in the region of 400 nm to 700 nm, incident upon the retina. Physical power (or radiance) is expressed in a spectral power distribution. A variety of spectral distributions of light can produce perceptions of color which are indistinguishable from one another. The human retina has three different types of color photoreceptor cone cells, which respond to incident radiation with somewhat different spectral response curves. Based on the human color perceptual system, three numerical components are necessary and sufficient to describe a color, provided that appropriate spectral weighting functions are used. Theoretically, color coordinates can be defined as product integrals of the stimulus spectrum U(ν) with three linearly independent color matching functions $\bar{r}(\upsilon)$, $\bar{g}(\upsilon)$, $\bar{b}(\upsilon)$,

$$R = \int_{\upsilon_1}^{\upsilon_2} \bar{r}(\upsilon)\, U(\upsilon)\, d\upsilon, \qquad \text{(EQ 2)}$$

$$G = \int_{\upsilon_1}^{\upsilon_2} \bar{g}(\upsilon)\, U(\upsilon)\, d\upsilon, \qquad \text{(EQ 3)}$$

$$B = \int_{\upsilon_1}^{\upsilon_2} \bar{b}(\upsilon)\, U(\upsilon)\, d\upsilon, \qquad \text{(EQ 4)}$$

where ν is the frequency of the light stimulus.

Most video cameras use a RGB model; other color models can be easily converted into a RGB model. However, the RGB model is not necessarily the best color model for representing skin-color. In the RGB space, a triple [r, g, b] represents not only color but also brightness. If the corresponding elements in two points, [r_1, g_1, b_1] and [r_2, g_2, b_2], are proportional, i.e.,

$$\frac{r_1}{r_2} = \frac{g_1}{g_2} = \frac{b_1}{b_2} \qquad \text{(EQ 5)}$$

they have the same color but different brightness. The human visual system adapts to different brightness and various illumination sources such that a perception of color constancy is maintained within a wide range of environmental lighting conditions [31]. Therefore it is possible for us to remove brightness from the skin-color representation while preserving an accurate but low dimensional color information. Since the brightness is not important for characterizing skin colors, under the normal lighting condition, we can represent skin-color in the chromatic color space. Chromatic colors (r, g) [32], known as "pure" colors in the absence of brightness, are defined by a normalization process:

$$r = R / (R + G + B), \tag{EQ 6}$$

$$g = G / (R + G + B). \tag{EQ 7}$$

In fact, (EQ 3) and (EQ 4) define a $\mathbf{R}^3 \rightarrow \mathbf{R}^2$ mapping. Color blue is redundant after the normalization because r+g+b =1. All chromatic colors (r, g) can be sketched in a chromaticity diagram as shown in Figure 3.

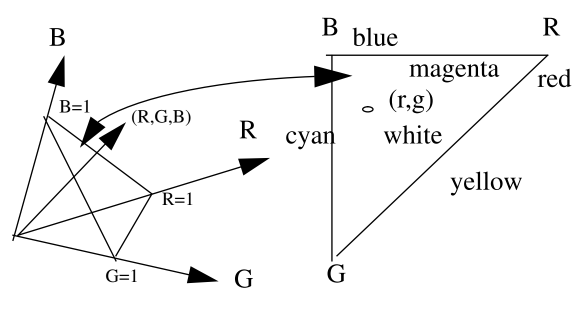


Figure 3 **The chromaticity diagram**

A color histogram is a distribution of colors in the color space and has long been used by the computer vision community in image understanding. For example, analysis of color histograms has been a key tool in applying physics-based models to computer vision. It has been shown that color histograms are stable object representations unaffected by occlusion and changes in view, and that they can be used to differentiate among a large number of objects

[34]. Although the three numerical values for image coding could, in theory, be provided by a color specification system, a practical image coding system needs to be computationally efficient and cannot afford unlimited precision. In this report, we represent color histogram in the chromatic color space with 8 bits for each normalized color band, i.e, there are 256^2 "bins" into which a pixel may fall.

In the mid-1980s, it was recognized that the color histogram for a single inhomogeneous surface with highlights will have a planar distribution in color space. It has since been shown that the colors do not fall randomly in a plane, but form clusters at specific points. The color histograms of human skin coincide with these observations. The Figure 4 shows a face image and corresponding area for histogram analysis. The histogram of the skin-color is illustrated in Figure 5. The color distribution of the skin-color is clustered in a small area of the chromatic color space, i.e., only a few of all possible colors actually occur in a human face.

(a) **(b)**

Figure 4 **An example of face and analyzed area**

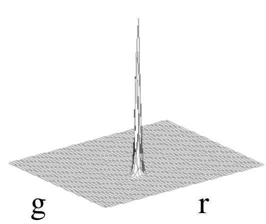
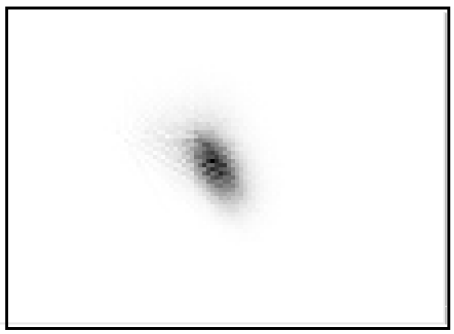


Figure 5 **The color distribution of a human face in chromatic color space**

We have further found that distributions of skin-colors of different people are clustered in chromatic color space. Although skin colors of different people appear to vary over a wide range, they differ much less in color than in brightness. In other words, skin-colors of different people are very close but they differ mainly in intensities. Figure 6 shows a skin color distribution of forty people with different skin colors in the chromatic color space. The distribution was obtained by analyzing faces of different races, including Asian, African American, and Caucasian. The grey-scale in the figure reflects the magnitude of the histogram. This result is significant because it provides evidence of the possibility of modeling human faces with different color appearances in the chromatic color space.

g



r

Figure 6 **Skin-color distribution cluster of different people**

The histogram is related not only to the face color, but also to the illumination color because only those colors can be reflected. For example, sunlight will shift color histograms towards blue because it contains more blue than fluorescent lighting. However, our experiments have shown that the shape of the histogram remains similar although there is a shift in the color histogram under changing lighting conditions. By closely investigating the face color cluster, we have discovered that the distribution has a regular shape. A close view of skin-color distributions is shown in Figure 7. (a) and (b) are color distributions of a face under different lighting conditions and (c) is the color distribution of two persons' faces. It is obvious that the human face colors of different people under different lighting conditions in the chromatic color space have similar Gaussian distributions as shown in Figure 8.

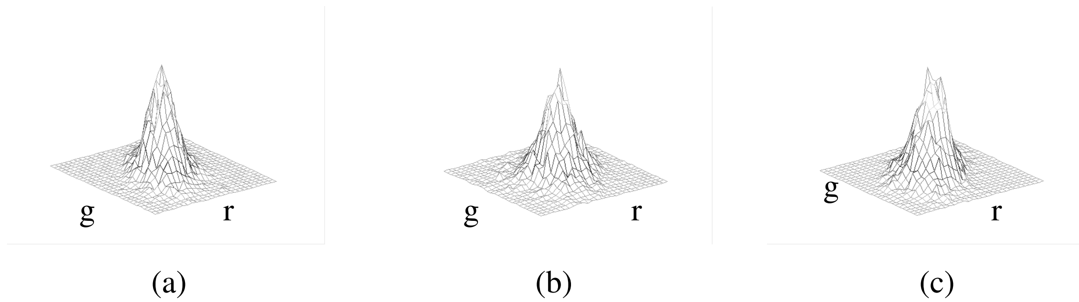


Figure 7 **The color distribution for different lighting condition and different persons**

Therefore, a face color distribution can be represented by a Gaussian model $N(m,\Sigma^2)$, where $m = (\bar{r}, \bar{g})$ with

$$\bar{r} = \frac{1}{N}\sum_{i=1}^{N} r_i \ , \qquad \text{(EQ 8)}$$

$$\bar{g} = \frac{1}{N}\sum_{i=1}^{N} g_i \ , \qquad \text{(EQ 9)}$$

and

$$\Sigma = \begin{bmatrix} \sigma_{rr} & \sigma_{rg} \\ \sigma_{gr} & \sigma_{gg} \end{bmatrix} \ . \qquad \text{(EQ 10)}$$

(EQ 11)

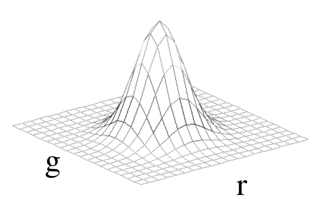


Figure 8 **The color distribution generated by the model**

The procedure for creating the skin-color model is as follows:

- Take a face image, or a set of face images if a general model is needed
- Select the skin-colored region, e.g. Figure 4(b), interactively
- Estimate the mean and the covariance of the color distribution in chromatic color space based on (EQ 5) - (EQ 7)
- Substitute the estimated parameters into the Gaussian distribution model

Since the model only has six parameters, it is easy to estimate and adapt them to different people and lighting conditions.

3.2 Skin-Color Model Adaptation

A number of viewing parameters, such as light sources, background colors, luminance levels, and media, impact greatly on the change in color appearance of an image. Most color-based systems are sensitive to changes in viewing environment. Although human skin colors fall into a cluster in the chromatic color space, skin-color models of different persons differ from each other in mean and/or variance. Even under the same lighting conditions, background colors such as colored cloths may influence skin-color appearance. Furthermore, if a person is moving, the apparent skin colors change as the person's position relative to camera or light changes. Therefore, the ability of handling lighting changes is the key to success for a color model.

There are two schools of philosophy to handle environment changes: tolerating and adapting. Color constancy refers to the ability to identify a surface as having the same color under considerably different viewing conditions. Although human beings have such ability, the underlying mechanism is still unclear. A few color constancy theories have demonstrated success on real images [39]. On the other hand, the adaptive approach provides an alternative to make a color model useful in a large range. Instead of emphasizing the recovery of the spectral properties of light sources and surfaces that combine to produce the reflected lights, the goal of adaptation is to transform the previously developed color model to the new environment. We have developed a method to adapt the skin-color model. The adaptive method has been applied to two situations where either the tracking object or lighting condition has been changed. Based on the identification of the skin-color histogram, the modified parameters of the model can be computed as follows:

$$\hat{\bar{r}}_k = \sum_{i=0}^{N-1} \alpha_{k-i} \bar{r}_{k-i}, \qquad \text{(EQ 12)}$$

where $\hat{\bar{r}}_k$ is the adapted mean value of r at sampling time k; $\alpha_i \le 1$, $i = k, k\text{-}1, \ldots, k\text{-}N\text{-}1$, are weighting factors; $\bar{r}_k$ is the estimated mean value of r at sampling time k; N is a computational window.

$$\hat{\bar{g}}_k = \sum_{i=0}^{N-1} \beta_{k-i} \bar{g}_{k-i}, \qquad \text{(EQ 13)}$$

where $\hat{g}_k$ is the adapted mean value of g at sampling time k; $\beta_i \le 1$, $i = k, k\text{-}1, \ldots, k\text{-}N\text{-}1$; are weighting factors; $\bar{g}_k$ is the estimated mean value of g at sampling time k; N is a computational window.

$$S_k = \sum_{i=0}^{N} \gamma_{k-i} \Sigma_{k-i}, \qquad \text{(EQ 14)}$$

where S_k is the adapted covariance matrix of color distribution at sampling time k; $\gamma_i \le 1$, $i = k, k\text{-}1, \ldots, k\text{-}N\text{-}1$, are weighting factors; Σ_i, $i = k, k\text{-}1, \ldots, k\text{-}N\text{-}1$, are the estimated covariance matrix of color distribution at sampling time k; N is a computational window.

The weighting factors α, β, γ in (EQ 8) - (EQ 10) determine how much the past parameters will influence current parameters. For example, setting all $\alpha, \beta, \gamma = \frac{1}{N}$ lets the parameters within window N make the same contribution to the current parameter estimations. The window size N determines how long the past parameters will influence the current parameters. The adaptive speed will decrease as N increases. We will discuss how to apply the skin-color model to locating and tracking human faces in the next section.

4 A Real-Time Face Tracker

We have developed a real-time face tracker as shown in Figure 9. The system consists of an HP-9000 workstation and a Canon camera (VC-C1). The camera's panning, tilting, and zooming are controlled by the computer via a serial port. Images are obtained by a framegrabber which digitizes the analog video signal into RGB values. The objective of the system is to provide the following functions in real-time:

- Locating arbitrary human faces in various environments in real-time;
- Tracking the face in real-time by controlling camera position and zoom after selecting a face;
- Adapting model parameters based on individual appearance and lighting conditions in real-time;
- Providing face location for user modeling applications in real-time.

Several techniques have been employed in developing the system to achieve these goals. Communication between the face tracker and other systems, e.g., a lip-reading system, is through sockets. The system can continuously provide other systems with information of the face position once the communication channel has been established. Three models, i.e., skin-color model, motion model, and camera model, have been used to achieve real-time tracking performance.

Figure 9 **The face tracker setup**

4.1 Skin Color Model for Face Locating

The fundamental idea of skin color model has been discussed in detail in the previous. Since the color distribution is represented by 8 bits for each chromatic color, a 2-D table is used to store the model. The model values can be then obtained by table-lookup. A straightforward way to locate a face is to match the model with the input image to find the face color clusters. Each pixel of the original image is converted into the chromatic color space and then compared with the distribution of the skin color model. Since the skin colors occur in a small area of the chromatic color space, the matching process is very fast. Figure 10 shows an example of extracting face region. Figure 10 (a) is the original image. Figure 10 (b) gives the result of color matching. Pixels with a high gray-scale value in Figure 10 (b) correspond to frequently occurring face colors. Although the skin-color region contains the eyes and the lips, there is little difficulty to locate a face based on the result of Figure 10 (b).

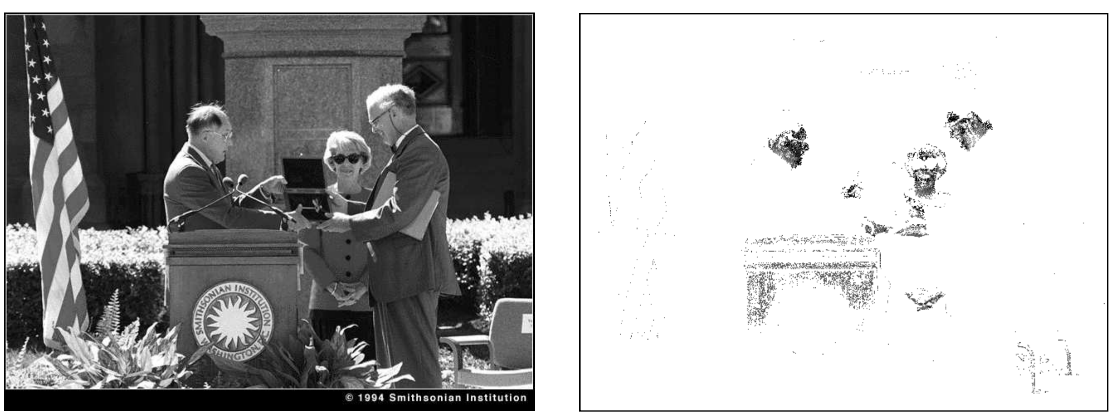
It is not always as easy as the example in Figure 10 to locate a face, because the background may contain skin colors, too. A variety of distributions of energy quanta of photons can be perceived as the same color. This means that many points in the color space representing the different physical distributions of photon energy quanta can be mapped onto a single point in the color space. In other words, the mapping between the physical spectrum and the color space can be many-to-one. Figure 11 (a) shows a scene of people with a complex background. Figure 11 (b) is the result of color matching. Skin colors occur not only on faces, but also on hands and the background. It is impossible to locate faces simply from the result of color matching.

(a) (b)

Figure 10 **An example of locating face by the skin-color model**

When faced with a many-to-one mapping problem, it is natural to use other mappings to get rid of uncertainties. This requires additional information. Three types of information are available from a sequence of images: color distribution, geometric, and motion information. Color distribution information can help to distinguish human faces from the background with skin-colors but with different distributions. Some background objects have skin-colors but different distributions. These backgrounds can be identified by distribution analysis.

Geometric information, such as size and shape of the interested objects, can be extracted from the image. The information can be used along with color information to locate faces. For example, a hand has skin-color but the skin-colored size of a hand is smaller than that of a face under the same perspective. Similarly, a skin colored object in background may have a different geometric shape compared to a face. For the Figure 11 (a) image, the hand size is smaller than face size and the magnitude of skin color distribution is smaller in the background. Therefore the system can locate faces by applying threshold on distribution and skin-colored size. The result is illustrated in Figure 12 in which three faces are located correctly.

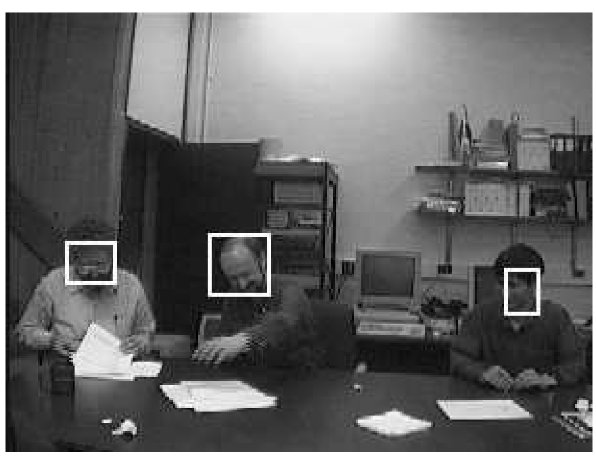


(a) (b)

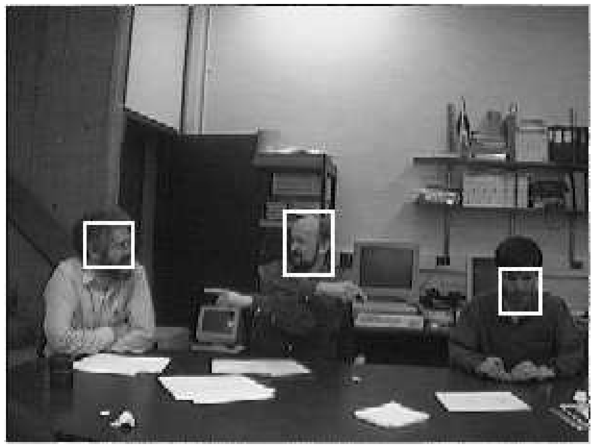
Figure 11 **An example of background with skin colors, (a) original image (copyright of 1994 Smithsonian Institution); (b) color matching result**

Figure 12 **Locating faces in a scene with skin-colored background**

Motion information can effectively distinguish a human face from the background. Since backgrounds cannot move, it is easy to differentiate a real human from a portrait on the wall by motion information. An example of locating faces using color, size, and motion is shown in Figure 13. The sequence of images was taken from a laboratory with a complicated background. By combining color, geometry, and motion information, three faces are accurately located.



(a)



(b)

Figure 13 **Locating faces using a combination of color, geometry, and motion information**

4.2 Motion Estimation and Prediction

Under the assumption that the image intensity doesn't change between adjacent frames, color is an orientation invariant feature. By using the skin color as a feature, a translation model is needed to characterize image motion. In this case, only one corresponding point, in theory, is needed to determine the model parameters. In practice, two or more points can be used for robust estimation. We can obtain these corresponding points by the face correspondence between adjacent image frames.

Since tracking can be formulated as a local search problem, the system can search for the feature locally within a search window instead of the entire image (Figure 14). The window size and position are two important factors in real-time tracking. A large search window results in unnecessary searching while a too small search window may easily lose the face. Several factors may influence search window size. For example, the search window size grows with the square of the maximum velocity of the face. An effective way to increase tracking speed is to use an adaptive search window. With a certain zoom, the face size can be a criterion to determine search window size. If a person is close to the camera, a small motion may result in a large change in the image, whereas if the person is far away from the camera, the same motion will have less influence on the image.

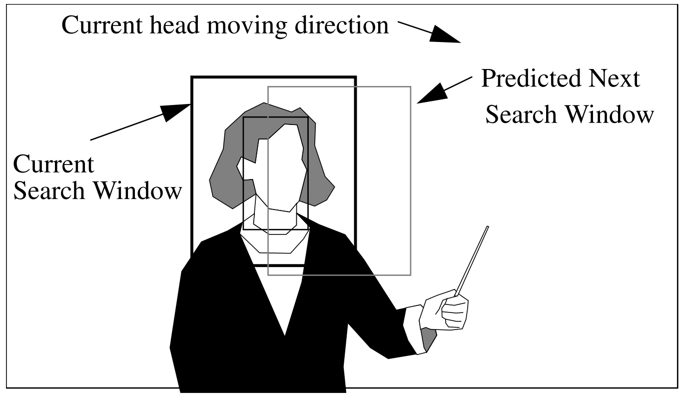


Figure 14 **Search window**

Motion prediction is effective in increasing tracking speed. The tracker only has to search small regions to find the features as long as the predictions are reliable. Some motion modeling techniques such as Kalman filters can help predict future position. These methods, however, are computational expensive. A simple way of predicting the motion is based on the current position and velocity. If the sampling rate is high enough, the location of a point in the current image and the displacement prediction based on the current image speed produce a very good approximation for the location in the next image.

4.3 Model-based Camera Control

In order to achieve high quality tracking performance, the face tracker uses a Canon VC-C1 camera with pan, tilt, and zoom control. There are two major problems with this camera: (1) the camera cannot pan and tilt simultaneously; (2) response of the camera is much slower compared to the real-time sampling rate. We have developed several methods to solve these problems. Instead of directly controlling the camera, the camera is controlled through a socket-based server. With the server, client code does not have to deal with complex RS-232 port and client code can ignore the fact that the VC-C1 does not have simultaneous pan, tilt or zoom. If we use a conventional feedback control scheme as shown in Figure 15 (a), we can hardly achieve good performance because of time-delay. To overcome time-delay, we have developed a model-based predictive feedback scheme as shown in Figure 15 (b).

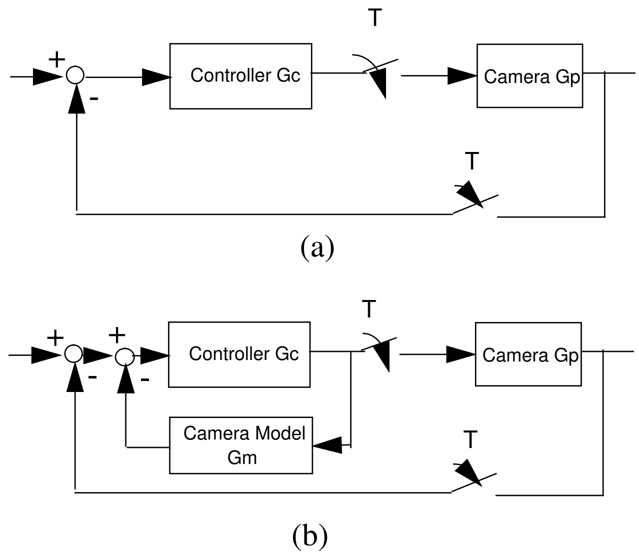


Figure 15 **Camera control scheme: (a) conventional feedback control; (b) model-based predictive control**

A camera model is used to predict the camera motion and compensate for egomotion. The pinhole camera model (Figure 16) is defined as:

$$-\frac{f}{z} = \frac{u}{x} = \frac{v}{y}, \qquad \text{(EQ 15)}$$

where (x,y,z) is the world coordinate; (u,v) is the image coordinate; f is the distances of the image plan from the pinhole.

Then the angle for pan is:

$$\theta_1 = \operatorname{atan}\left(\frac{x}{z}\right) = \operatorname{atan}\left(\frac{u}{f}\right) \qquad \text{(EQ 16)}$$

and the angle for tilt is:

$$\theta_2 = \operatorname{atan}\left(\frac{y}{z}\right) = \operatorname{atan}\left(\frac{v}{f}\right) \qquad \text{(EQ 17)}$$

With (EQ 11)-(EQ 13), the face tracker can effectively control the camera and compute the egomotion. The errors caused by the model can be reduced by feedback control.

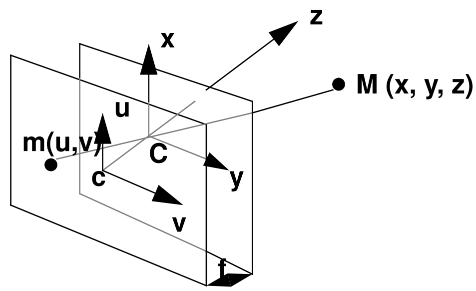


Figure 16 **Pinhole camera model**

4.4 System Initialization

Two methods can be used to initialize a tracking process. The first method utilizes skin-colors and motion to start the tracking process. The method is based on the assumption that the face of the greatest interest to us is the face closest to the camera. A general skin-color model based on prior information is used in the search process. Motion information is used to differentiate faces from skin-colored backgrounds. The face tracker locates all the moving objects with skin-colors in the image, then selects the largest among these objects to track. Once a face is found, the skin color model is adapted to the face being tracked.

The second method is based on an interactive interface as shown in Figure 17. The face is selected by the user through a mouse, or a finger if a touch screen is used. After the face is selected, the face tracker starts with the search process from a small area around the selected point. A general skin color model is used for the search process. If a skin colored region is found, the size of the search area is increased and search process is repeated. The results from two adjacent search processes are then compared. If the results are the same, the skin color model is adapted to the face selected. If the results are different, the search process is repeated until the results from two search process are the same.

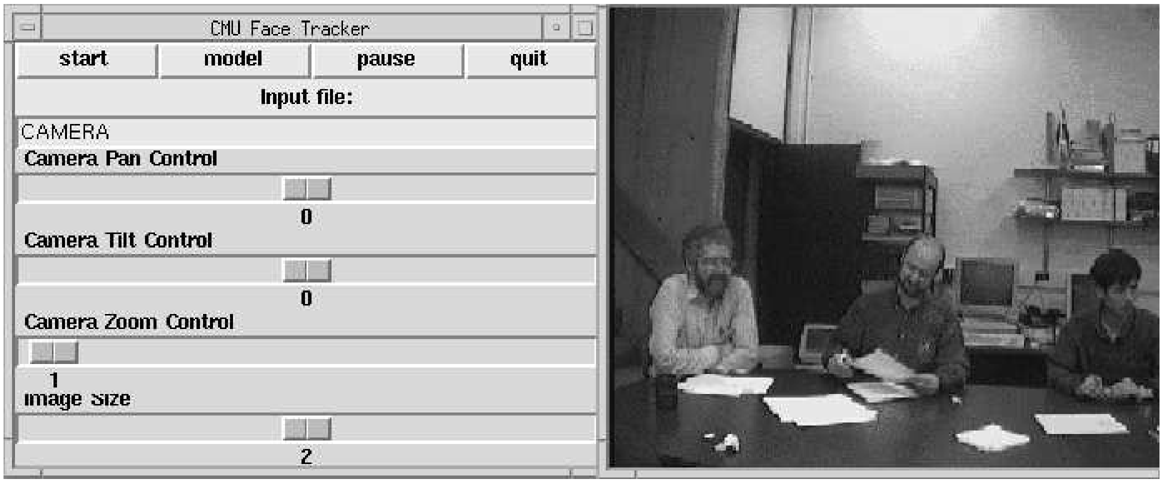


Figure 17 **An interactive interface to start the face tracker**

4.5 Tracking Faces in Real-time

Once a face is selected, the face tracker starts the tracking process. During this process, the skin color model is used to find the face within the search window. The motion estimation and prediction are then based on the search result. The pan, tilt, and zoom of the camera are adjusted if needed. The skin-color model is updated in real-time based on the new estimated parameters. If the tracking fails to find the face, the search window size is increased until the face is found again. The face tracker can continuously track a person while he/she is moving freely (e.g., sitting, rising, walking). Figure 18 shows that a face is tracked in various poses and views. The big white square in each image indicates the search window and the small white square shows tracking result. The results were obtained from a real-time tracking process.

The system has been running in our lab for about a year with continuous improvements in performance. The current tracking speed using an HP-735 workstation is shown in table 2. The table suggests that the tracking speed greatly depends on the search window size. For example, when the face is closer to the camera, the face image is relatively bigger and so is the search window size.

Figure 18 **Some real-time tracking results (the bigger white square in each image is the search window and the smaller white square is the located face)**

Table 2: Tracking speed for different distances between the face and camera

Distance (m)	0.5	1.0	>2.0
Frame/Second	15	20	30+

5 Conclusion

We have proposed a model-based approach to real-time face tracking. We have presented an adaptive skin-color model to characterize human faces in different views under different lighting conditions. We have demonstrated that we can track a human face in real-time by combining the skin-color model with motion and camera models. The skin color model provides sufficient information for tracking human faces in real-time. The motion model is the key to estimating image motion and predicting search window position. The camera model effectively compensates for time-delay and egomotion. We have implemented a real-time face tracker. The system has achieved a rate of up to 30+ frames/second using an HP-9000 workstation with a framegrabber and a Canon VC-C1 camera. The system can track a person's face while the person walks, jumps, sits and rises in a room. The methodology of developing the face tracker is useful for developing other real-time tracking systems. The real-time face tracker can be applied to many applications in human computer interaction and tele-conferencing.

We are currently working on several extensions of the face tracker. While continuously improving the robustness and accuracy for the current system, we are also working on tracking multiple people simultaneously. We are porting the system to other platforms such as the PC. We are applying the face tracker to eye tracking and environmental modeling for teleconferencing.

Acknowledgment

We thank Ricky Houghton and other colleagues in Interactive Systems Laboratories for their technical supports to this project. We would also like to thank Heung-Yeung Shum and Minh Tue Vo for their many valuable comments which have significantly improved the quality of this report. This research was sponsored by the Advanced Research Projects Agency under the Department of the Navy, Naval Research Office under grant number N00014-93-1-0806.

References

[1] C. Bregler, H. Hild, S. Manke, and A. Waibel. Improving connected letter recognition by lipreading. Proc. Int. Conference on Acoustics, Speech and Signal Processing pp. 557--560, New York, NY, USA, 1993.

[2] U. Bub, M. Hunke, and A. Waibel. Knowing who to listen to in speech recognition: visually guided beamforming. Proc. Int. Conference on Acoustics, Speech and Signal Processing, 1995.

[3] P. Ekman and W.V. Friesen. Facial Action Coding System. Consulting Psychologist Press, Palo Alto, 1978.

[4] I.A. Essa and A. Pentland. A vision system for observing and extracting facial action parameters. Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 76--83, Seattle, WA, USA, 1994.

[5] R. Brunelli and T. Poggio. Face recognition: features versus templates. IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 15, No. 10, pp. 1042-1052, Oct. 1993.

[6] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspace for face recognition. Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 84-91, Seattle, WA, USA, 1994.

[7] A. Yuille., P. Hallinan, and D. Cohen. Feature extraction from faces using deformable templates. Int. J. Computer Vision, Vol. 8, No. 2, pp. 99-111, 1992.

[8] P. Sinha. Object recognition via image invariants: a case study. Investigative ophthalmology and visual science, Vol. 35, pp. 1735-1740, 1994.

[9] M.A. Turk and A. Pentland. Face recognition using eigenfaces. Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp. 586-591, Maui, HI, USA, 1991.

[10] K. Sung and T. Poggio, Example-based learning for view-based human face detection. Technical Report 1521, MIT AI Lab, 1994.

[11] H.A. Rowley, S. Baluja, and T. Kanade. Human face detection in visual scenes. Technical Report CMU-CS-95-158, CS department, CMU, 1995.

[12] M. Hunke and A. Waibel. Face locating and tracking for human-computer interaction. Proc. Twenty-Eight Asilomar Conference on Signals, Systems & Computers, Monterey, CA, USA, 1994.

[13] J.K. Aggarwal and N. Nandhakumar. On the computation of motion from sequences of images-a review. Proceedings of the IEEE, Vol. 76, No. 8, pp. 917-935, 1988.

[14] J.F. Vega-Riveros and K. Jabbour. Review of motion analysis techniques. IEE Proc. I, Commun. Speech Vis. Vol. 136, No. 6, pp. 397-404, 1989.

[15] L.G. Brown. A survey of image registration techniques. Computing Surveys, Vol. 24, No. 4, pp. 325-376, 1992.

[16] B.K.P. Horn and B.G. Schunck. Determining optical flow. Artificial Intelligence, Vol. 17, pp. 185-203, 1981.

[17] B.D. Lucas and T. Kanade. An iterative image registration technique with an application in stereo vision. Proceedings of Seventh International Joint Conference on Artificial Intelligence, pages 674-679, Vancouver, 1981.

[18] M. Otte and H.-H. Nagel. Optical flow estimation: advances and comparisons. Proceedings of Third European Conference on Computer Vision (ECCV'94), pp. 51-60, Springer-Verlag, Stockholm, Sweden, May 1994.

[19] J.L. Barron, D.J. Fleet, and S.S. Beauchemin. Performance of optical flow techniques. International Journal of Computer Vision, Vol. 12, No. 1, pp. 43-77, January 1994.

[20] B.D. Lucas. Generalized Image Matching by the Method of Differences. Ph.D. thesis, Carnegie Mellon University, July 1984.

[21] J.R. Bergen, P. Anandan, K.J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. Proceedings of Second European Conference on Computer Vision (ECCV'92), pp. 237-252, Springer-Verlag, Santa Margherita Liguere, Italy, May 1992.

[22] J.R. Muller, P. Anandan, and J.R. Bergen. Adaptive-complexity registration of images. Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'94), pp. 953-957, IEEE Computer Society, Seattle, Washington, June 1994.

[23] R. Szeliski and H.-Y. Shum. Motion estimation with quadtree splines. Proceedings of Fifth International Conference on Computer Vision (ICCV'95), Cambridge, Massachusetts, June 1995.

[24] S.T. Barnard and M.A. Fischler. Computational stereo. Computing Surveys, Vol. 14, No. 4, pp. 553-572, December 1982.

[25] L.H. Quam. Hierarchical warp stereo. Proceedings of Image Understanding Workshop, pp. 149-155, Science Applications International Corporation, New Orleans, Louisiana, December 1984.

[26] M. Okutomi and T. Kanade. A multiple baseline stereo. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.15, No. 4, pp. 353-363, April 1993.

[27] W. Enkelmann. Investigations of multigrid algorithms for estimation of optical flowfields in image sequences. Computer Vision, Graphics, and Image Processing, Vol. 43, No. 2, pp. 150-177, 1988.

[28] H.-H. Nagel. On the estimation of optical flow: Relations between different approaches and some new results. Artificial Intelligence, Vol. 33, pp. 299-324, 1987.

[29] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. International Journal of Computer Vision, Vol. 2, No. 3, pp. 283-310, 1989.

[30] J. Shi and C. Tomasi. Good features to track. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 593-600, Seattle, WA, USA, 1994.

[31] D.H. Brainard, B.A. Wandell, and E.-J. Chichilnisky. Color constancy: from physics to appearance. Current Directions in Psychological Science, Vol. 2, No. 5, pp. 165-170, 1993.

[32] G. Wyszecki and W.S. Styles. Color Science: Concepts and Methods, Quantitative Data and Formulae, Second Edition, John Wiley & Sons, New York, 1982.

[33] R.M. Haralick and G.L. Kelly. Pattern recognition with measurement space and spatial clustering for multiple images. Proceedings of IEEE, Vol. 57, No. 4, pp. 654-665, 1969.

[34] M.J. Swain and D.H. Ballard. Color indexing. International Journal of Computer Vision. Vol. 7, No.1, pp. 11-32, 1991.

[35] B.V. Funt and G.D. Finlayson. Color Constant Color Indexing. IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 17, No. 5, pp. 522-529, 1995

[36] J.D. Crisman and C.E. Thorpe. SCARF: a color vision system that tracks roads and intersections. IEEE Trans. Robot. Autom., Vol. 9, No. 1, pp. 49-58, 1993.

[37] D. Sinclair, A. Blake, and D. Murray. Robust estimation of egomotion from normal flow. International Journal of Computer Vision. Vol. 13, No. 1; pp. 57-69, 1994.

[38] M.J. Barth and S. Tsuji. Egomotion determination through an intelligent gaze control strategy. IEEE Transactions on Systems, Man and Cybernetics. Vol. 23, No. 5, pp. 1424-1432, 1993.

[39] D. Forsyth. A novel algorithm for color constancy. International Journal of Computer Vision. Vol. 5, No. 1, pp.5-36, 1990.