A Real-Time Face Tracker

Jie Yang Alex Waibel

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
{yang+, waibel}@cs.cmu.edu

Abstract

We present a real-time face tracker in this paper. The system has achieved a rate of 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 in developing the system. First, we present a stochastic model to characterize skin-color distributions of human faces. The information provided by the model is sufficient for tracking a human face in various poses and views. This model is adaptable to different people and different lighting conditions in real-time. Second, a motion model is used to estimate image motion and to predict search window. Third, a camera model is used to predict and to compensate for camera motion. The system can be applied to tele-conferencing and many HCI applications including lip-reading and gaze tracking. The principle in developing this system can be extended to other tracking problems such as tracking the human hand.

1 Introduction

A human face provides a variety of different communicative functions, such as identification, perception of emotional expressions, and lip-reading. 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 [1][2], deformable templates [3], and spatial image invariants [4]. Several systems of locating human face have been reported. Eigenfaces, obtained by performing a principal component analysis on a set of faces, are commonly used to identify faces [5]. By moving a window covering a subimage over the entire image, faces can be located within the entire image. [6] 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 [7]. A different approach for locating and tracking faces is described in [8]. This system locates faces by searching for the skin-color. After locating a face, the system extracts additional features to match this particular face. Recently, Pfinder [9] uses skin-color to track human body.

In this paper, 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 is adaptable to 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 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 30+ frames /second using an HP-9000 workstation with a framegrabber and a Canon VC-C1 camera.

2 Tracking Human Faces

The tracking problem is distinguished from the recognition problem in that its search processes are local rather than global. The problem of locating a face 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. This requires the system's ability to estimate the motion while locating the face. Furthermore, the system needs to control the camera, e.g., panning, tilting, and zooming to track faces outside a certain range

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.,

$$I_t(x+u_t,y+v_t) = I(x,y), \qquad (1)$$

we wish to recover the displacement fields (u_t,v_t) from the reference image $I(x,y)$. 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. A large number of approaches have been proposed to solve this problem [10][11][12].

To locate human faces, facial features, such as the 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. 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 are effective 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.

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	$X' = \frac{AX + b}{C^T X + 1}$	$A \in R^{2x2}, b, c \in R^2$

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 with only two parameters as listed in Table 1 is needed for motion estimation.

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 [14]. 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 object recognition [15][16][17] and recently has been successfully applied to road tracking [18] and face tracking [8]. In the next section, we develop a stochastic model of skin-color in the chromatic color space for tracking a human face. The model has only a 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, size of the face in the image, and the 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, creates more challenging problems for tracking. Most approaches for egomotion analysis are based on optical flow [19][20]. 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.

3 A Stochastic Model for Tracking Faces

3.1 Skin-Color Model

Most video cameras use a RGB representation; other color representations can be easily converted to a RGB representation. However, RGB is not necessarily the best color representation for characterizing 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.,

$$\frac{r_1}{r_2} = \frac{g_1}{g_2} = \frac{b_1}{b_2} \qquad (2)$$

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 [13]. 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) [14], known as “pure” colors in the absence of brightness, are defined by a normalization process:

$$r = R / (R + G + B), \qquad (3)$$

$$g = G / (R + G + B). \qquad (4)$$

In fact, (3) and (4) define a $\mathbf{R}^3 \rightarrow \mathbf{R}^2$ mapping. Color blue is redundant after the normalization because r+g+b =1.

A color histogram is the distribution of colors in the color space and has long been used by the computer vision community in image understanding. 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 1 shows a face image and corresponding area for histogram analysis. The histogram of the skin-color is illustrated in Figure 2. 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.

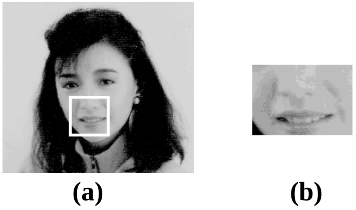

Figure 1 An example of face and analyzed area

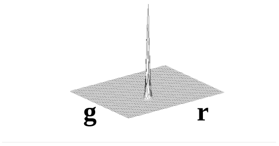

Figure 2 The color distribution of a human face in chromatic color space

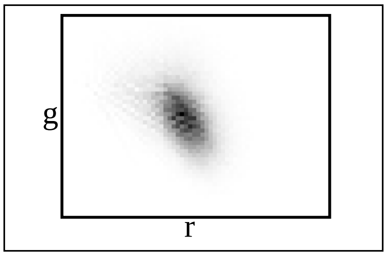

Figure 3 Skin-color distribution cluster of different people

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 3 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.

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 4. (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 skin-color distributions of different people under different lighting conditions in the chromatic color space have similar Gaussian distributions.

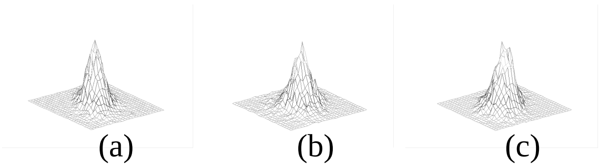

Figure 4 The color distribution for different lighting condition and different persons

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

$$\bar{r} = \frac{1}{N}\sum_{i=1}^{N} r_i \;, \qquad (5)$$

$$\bar{g} = \frac{1}{N}\sum_{i=1}^{N} g_i \;, \qquad (6)$$

and

$$\Sigma = \begin{bmatrix} \sigma_{rr} & \sigma_{rg} \\ \sigma_{gr} & \sigma_{gg} \end{bmatrix} . \qquad (7)$$

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

1. Take a face image, or a set of face images if a general model is needed
2. Select the skin-colored region, e.g. Figure 1(b), interactively
3. Estimate the mean and the covariance of the color distribution in chromatic color space based on (5) - (7)
4. Substitute the estimated parameters into the Gaussian distribution model
5. 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

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 [21]. 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 into the new environment. We have developed a method to adapt the skin-color model. Based on the identification of the skin-color histogram, the modified parameters of the model can be computed as follows:

$$\hat{r}_k = \sum_{i=0}^{N-1} \alpha_{k-i}\bar{r}_{k-i}, \qquad (8)$$

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

$$\hat{g}_k = \sum_{i=0}^{N-1} \beta_{k-i}\bar{g}_{k-i}, \qquad (9)$$

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

$$S_k = \sum_{i=0}^{N} \gamma_{k-i}\Sigma_{k-i}, \qquad (10)$$

where S_k is the adapted covariance matrix of color distribution at sampling time *k;* $\gamma_i \le 1$, *i = k, k-1,..., k-N-1,* are weighting factors; Σ_i , *i = k, k-1,..., k-N-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 (8) - (10) determine how much the past parameters will influence current parameters. We will discuss how to apply the skin-color model to locating and tracking human faces in the next section.

4 A Real-time System

We have developed a real-time face tracker [22]. 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 the 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 established 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.

4.1 Skin Color Model for Face Locating

The fundamental idea of skin color model has been discussed in detail in the previous. 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 5 shows an example of extracting face region. Figure 5 (a) is the original image. Figure 5 (b) gives the result of color matching. Pixels with a high gray-scale value in Figure 5 (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 5 (b).

It is not always as easy as the example in Figure 5 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. It is impossible to locate faces simply from the result of color matching.

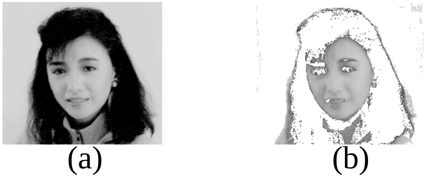
(a) (b)

Figure 5 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 eliminate uncertainties. This requires some additional information. Three types of information are available from a sequence of images: color distribution, geometric, and motion information. An example of locating faces using color, size, and motion is shown in

Figure 6. 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.

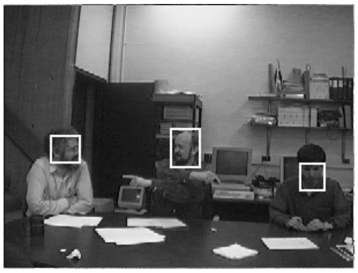

Figure 6 **Locating faces using a combination of color, geometry, and motion information**

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. 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 the 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.

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 to 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, we use 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, 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 7.

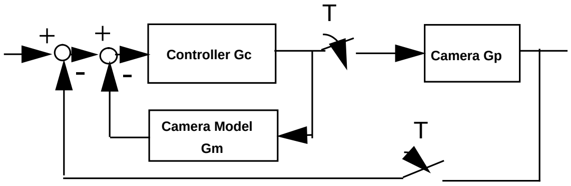

Figure 7 Camera control scheme: model-based predictive control

A camera model is used to predict the camera motion and compensate for egomotion. With the pinhole camera model, the face tracker can effectively control the camera and compute the egomotion. The errors caused by the model can be reduced by feedback control.

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 all objects to track. Once a face is found, the skin color model is adapted to the tracked face.

The second method is based on an interactive interface. 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 will adapt to the selected face. If the results are different, the search process will repeat until the results from two search process are the same.

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 will increase 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). The system has been running in our lab for about a year with continuous improvements in performance. The current tracking speed using an HP-9000 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.

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

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

5 Conclusion

We have presented a real-time face tracker in this paper. The development of the real-time face tracker is significant in the following aspects. First, we have addressed the problems of what to track and how to track a human face in real-time through presenting the real-time face tracker. The methodology of developing the face tracker is useful for developing other real-time tracking systems. Second, we have demonstrated the feasibility of modeling techniques in developing a real-time tracking system. Finally, the real-time face tracker itself has many applications in human computer interaction and tele-conferencing [23].

Acknowledgments

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] 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.

[2] 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.

[3] 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.

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

[5] 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.

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

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

[8] 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.

[9] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, “Pfinder: real-time tracking of the human body,” Proc. SPIE, Vol.2615, pp. 89-98, 1996.

[10] 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.

[11] 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.

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

[13] 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.

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

[15] 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.

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

[17] 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

[18] 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.

[19] 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.

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

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

[22] J. Yang and A. Waibel, “Tracking Human Faces in Real-Time,” CMU CS Technical Report, CMU-CS-95-210, November, 1995.

[23] J. Yang, L. Wu, and A. Waibel, “Focus of attention: towards low bitrate video tele-conferencing,” ICIP’96, Lausanne, Switzerland, 1996.