ROBOT VISION - COLOR BASED HUMAN TRACKING USING THE UV MODEL

PETER LIM

(B.Tech.(Hons), NUS)

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2004

Acknowledgments

The completion of this work is the result of many invisible hands helping to keep its author on course. My most heartfelt thanks to Dr. Prahlad Vadakkepat and Dr. Liyanage C. De Silva. They have been a constant source of suggestions and assistance throughout this research and all the time that I am seeking this degree.

I would also like to express my gratitude to Leng Li Li and Mak Mei Poh for the information and suggestion in implementing the design; which served as the baseline work. Special appreciation to my friends, classmates and colleagues for their valuable advice, source of inspiration and information.

Lastly, I am thankful to Kim for her encouragement and support in making this whole experience interesting and challenging.

Contents

Summary

The advancement in wafer technology leads to better and efficient microprocessors. The microcontroller implementations are not restricted to computer hardware and application areas include manufacturing lines, consumer products, household appliances to cite a few.

The need for robots to co-exist with humans requires Human-machine interaction. It is a challenge to operate these robots in dynamic environments which requires continuous decision making and environment attribute updates in real-time. An autonomous robot guide is well suitable in places such as museums, libraries, schools, etc.

This thesis records the research work in the field of computer and robot vision. It addresses a scenario where a robot tracks and follows a human. Some of the conventional methods such as Neural network, color skin probability map, Heuristic rule and Bayesian decision theory are used in this work. The Neural network involves learning of the skin and non-skin colors. The color skin probability map is utilized for skin classification and morphology based pre-processing. Heuristic rule is used for face ratio analysis and Bayesian cost analysis for label classification. The real-time face detection module, based on a two-dimensional color model in the YUV color space is selected over the traditional skin color model in a three dimensional color space. A modified CAMSHIFT tracking mechanism in a one-dimensional HSV color space is developed and implemented onto the mobile robot. In addition to the visual cues, the tracking process considers sixteen sonar scan and tactile sensor readings from the robot to generate a robust measure of

the person's distance from the robot. The robot thus decides an appropriate action; namely, to follow the human subject and perform obstacle avoidance. The proposed approach is orientation invariant under varying lighting conditions and invariant to natural transformations such as translation, rotation and scaling.

Chapter two to four of the thesis records the principles, techniques and methodology involving segmentation, image processing, target detection and tracking. Issues relating to color perception, color modeling and variations in camera parameters are documented. Experiments are conducted to examine the performance of these models under various lighting conditions.

Chapter five to seven of the thesis outlines the design, algorithms, architecture and the model of the proposed face detection and tracking system. Experiments are conducted to support the design and an explanation of the chosen model is documented in this section. The design of the various controllers and associated assumptions are recorded as well.

Lastly, a multi-modal approach is introduced with a working algorithm and the control mechanisms implemented to a Magellan Pro mobile robot for system performance analysis. The implemented system is able to perform human detection and tracking in a real-time environments with good accuracy.

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background - Robot and Computer Vision

Mobile robots are gradually entering into our daily lives; for example, Humanoid Robot - ASIMO [1] from Honda, the robot pet - Aibo [2] and the dancing robot - QRIO [3] from Sony. The robot soccer competitions such as RoboCup [4] and FIRA Robot World Cup [5], and the NASA's Mars Exploration Rovers [6] are the most noticeable evidence of the approaching breakthrough in mobile robot applications. Although most of the above applications are purely for research and entertainment, the main objective of robotic research is to aid industrialism, making human lives convenient and flexible with the use of robots. In research laboratories, mobile robots are used for mail delivery, to open/close doors, to handle objects, to interact with the users and for navigation [7]. In order to provide the basic navigational ability to the robot and to study the coarse structure of the environment, visual, sonar, ultrasonic, infrared and other range sensors are required. These robots have to acquire a lot of information about the environment through various sensors. Nevertheless, the dynamic nature of the environment and the need to interact with the users have set requirements that are more challenging in robot perception.

Automated analysis and processing of visual data or Computer Vision has been an area of research since the 60's and visual tracking is one of the most active research fields. Vision is the most important sense for humans and color is an important visual stimuli.

Color is an effective visual cue and image attribute used in visual detection and tracking and attracting attention. It is computationally feasible to have orientation invariance under varying lighting conditions and invariance to natural transformations such as translation, rotation and scaling. However, the dependence of the perceived color on illumination is a challenge in color-based segmentation. Color-based visual tracking is widely used in robot soccer systems to control the mobile robots. However, the working environment is quite easy for color segmentation as the colors are distinctive and uniform. In fact, the colored field, goals and robot markers have been designed so as to extract useful information easily by simple color-based segmentation methods.

The ability of pattern recognition is one of the key features of intelligent behavior, be it humans, animals or machines. Pattern Recognition is the act of taking in raw data and making a decision based on the class of the pattern. It is the ability emerged with the biogenetic evolution as a mere matter of survival for the individuals and the species.

Feature extraction is the first task in pattern recognition and it extreme complexity constitutes the main work of the pattern recognizer. The result of the feature extraction stage is a set of samples, x, that are fed to the classification or decision stage of the recognizer (Figure 1.1). To classify an object into one of a number of classes is the main objective of pattern recognition.

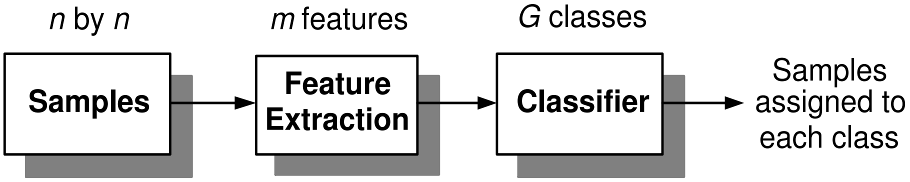


Figure 1.1: A two-stage recognizer

Computer Vision has found many applications in robotics: Human-Computer Interaction (HCI), content-based video indexing and retrieval, security and surveillance. Research into machine perception spans multiple disciplines and the approaches taken vary with the intended application. Applications include, creating intelligent agents capable of recognizing speech and conversing with people, adding perceptual capabilities of a robot in Robotics and detecting human body movements as an input device to Virtual Reality (VR) systems.

1.2 Related Works on Face Detection and Tracking Techniques

There are several face detection techniques which are broadly classified as motion-based detection, neural network approach, model-based approach and skin color segmentation techniques [8, 9, 10]. Motion-based face detection techniques assume that the targets are mostly mobile in a video scene. Therefore, motion is used to characterize the targets in relation to a static environment. The current captured image is compared to a reference image to segment out the mobile targets [11, 12, 13]. Neural network detection [14, 15, 16, 17] requires the learning from both positive (face) and negative (non-face) examples. Finding a representative sample of a face is easy, finding a "non-face" image is a challenge, since a "non-face" object is virtually anything ranging from trees to vehicles, making the training process difficult and tedious. Training of the neural network is mostly done "off-line" and research is undergoing to improve the real-time performance of systems employing neural networks in face detection. Model-based approaches include the use of models, which have a number of parameters describing the face shape or motion of the face or a combination of both to aid in estimation [18, 19, 20]. Some of the model-based approaches use the Hausdorff Distance [21, 22, 23] which is robust, accurate and efficient enough for real-time face detection.

Target tracking techniques include Kalman filter [24, 25, 26] and fuzzy control [27, 28, 29]. Kalman filter assumes a dynamic model of the target and that the noise affecting the system is zero mean. The Fuzzy logic approach requires a fuzzy logic knowledge base to control the specified system. Informally, a knowledge base is a set of representations of facts about the environment. Recent research has however adopted the mean shift algorithm [30, 31, 32], which is used for the analysis of probability distributions, into the field of real-time visual tracking. Yang and Waibel [33] uses a motion model to estimate image motion and to predict the search window; integrated with a camera model to predict and compensate for camera motion.

1.3 Problem Definition and Motivation

It is challenging to provide a mobile robot with a visual sensing system that is able to reconfigure easily to suit different tasks. There are numerous color-based segmentation methods developed for the visual perception by mobile robots. The computational load caused by the chosen method for visual tracking in real-time is one of issues to tackle. The limited computationally resources of a mobile robot and many other tasks that are running in parallel are to be taken into account while selecting a color-based segmentation method. The dynamic nature of the environmental conditions, for example, illumination, is another challenge for the visual system of a mobile robot.

In order to achieve real-time performance and simplicity in design and implementation, color-based tracking is used in this work. The developed tracking mechanism is based on the CAMSHIFT (Continuously Adaptive Mean Shift) algorithm [30]. It is computationally efficient and deals with the challenges of irregular object motion due to variations in perspective and image noise, such as other skin-color objects and facial countenance. CAMSHIFT is designed as a computer interface for controlling commercial computer games. The motion of a moving camera is not considered in CAMSHIFT.

To design a robust system, the tracking algorithm is not activated until a face is detected in the video scene. The conventional approach is to gather images of faces, and segregate the skin and non-skin areas. A skin-color model is incorporated in the detection module for detecting faces. Neural network learning is used to build this skin color model in the YUV color space. The control mechanisms are added and a multi-modal approach [34, 35, 36, 37] to tracking is considered. The multimodal implementation can extend beyond the use of mouse, keyboard and camera to include other sensory modalities such as tactile and auditory. In this work, tracking and obstacles avoidance are done with the information from the sonar sensors, tactile sensors and the conventional visual cues.

1.4 Purpose of the Thesis

The objective of this thesis is to develop a methodology for real-time color-based visual detection and tracking. The visual detection and tracking system developed is implemented on a mobile robot.

Figure 1.2 shows a typical flow diagram of a visual tracking system.

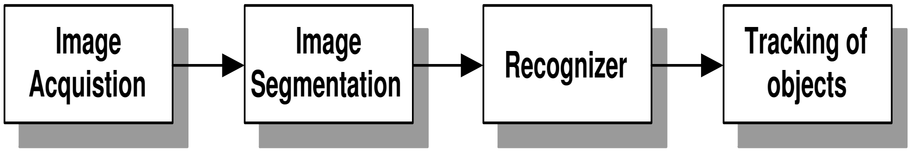


Figure 1.2: Typical flows of tasks in a visual tracking system

The principles, conventional techniques and methodology associated with the area of computer vision are outlined to begin with. Issues relating to color perception, color modeling and camera parameters are discussed. Experiments are conducted to examine the performance of various color models under different lighting conditions.

The design, algorithms, architecture and modeling of the proposed face detection and tracking system are then elaborated with various controllers designed for the mobile robot. Design improvements are made through experimentation.

A multi-modal approach is introduced and the control mechanisms are implemented onto a mobile robot for demonstration and system performance analysis.

1.5 Layout of the Thesis

This thesis discusses the challenges and approaches seated to human face detection and tracking in real-time. Two important issues: "what to detect and how to track" are explored.

Chapter two to four of the thesis addresses the issue of "what to detect" and presents a skin-color model in the YUV color space for human face characterization. The model is able to detect both stationary and moving human faces with different skin colors under varying lighting conditions in real-time.

Chapter five to seven involves the development of the color-tracking algorithm to address the issue of "how to track". The detection module and the visual tracking module are integrated into a mobile robot which has a Sony EVI-D30 pan-tilt camera and, sixteen sonar and tactile sensors.

Lastly, the performance of the proposed system is compared with some of the previous face detection and tracking approaches.

Chapter 2

Color-based Segmentation

In order to perform color-based segmentation, it is important to understand the influence of light on color. This chapter provides the fundamental facts on color and the understanding of color images. Furthermore, major consideration in color-based segmentation performance such as color constancy, illumination, camera operating parameters and color space are also discussed.

2.1 What is Color?

Color is a physical property that arises from the spectral properties of light. One is able to observe and label the surrounding objects with the help of color. That is, to identify the hue, the purity or saturation and luminance. Color exists only in the brain, as a result of some complex co-operation between the human eye and the brain. It is light that encounters the retina of an eye [38] resulting color perception in humans' existence.

Light is a mixture of electromagnetic energy at various wavelengths, however, human eyes are sensitive to only a very narrow band, known as the visible light spectrum which ranges from 400 to 700 nanometers. The light is characterized by the Spectral Power Distribution (SPD), which indicates the amount of energy at each wavelength.

Human retina has three types of color photoreceptor cone cells, each responding to the incident light with different spectral response curves. Colors are sensed as near-linear combinations of long, medium and short wavelengths, which correspond to the three primary colors, red, green and blue (Figure 2.1).

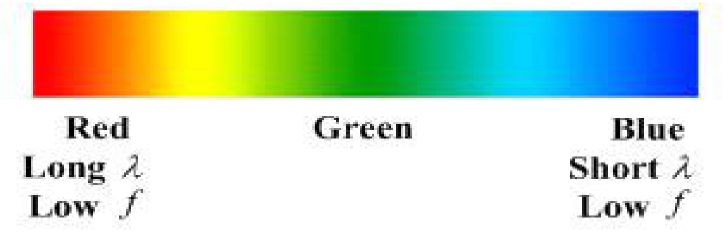


Figure 2.1: The Visible Light Spectrum

Human sensation of hue, saturation and luminance is determined by a mixture of responses from the three types of cone cells and is known as the Trichromatic Theory of color [39]. Despite intensive research in the human visual system, the exact process between the cone cell stimulus and the colors remains unknown to science.

The color imaging conforms to the physical laws in the natural world; that is, the light power is spectrally distributed. The sensation of color has to be artificially created and converted into a digital format in a computer vision system.

In a computer vision system, cameras are in place of the human eyes and, the role of the retina is played by a sensor or an imager that reacts to light and transforms the electromagnetic energy into electric energy. The electrical signals produced by the sensors are sampled and quantized to produce a two-dimensional digital representation of the three-dimensional real world scene; known as digitization of an image or Analog-to-Digital Conversion (ADC) process. This representation is limited by the properties of the imaging devices and the need to understand the restrictions caused by the hardware is an essential issue in developing a machine vision system.

2.2 Three approaches in understanding Color

There are three main approaches in understanding color. Each of these approaches introduces a different and important viewpoint. Careful consideration is required to select one among them when developing or evaluating color machine vision systems.

2.2.1 Statistical Approach

In the statistical approach, color is regarded as a statistical quantity. It is analyzed as a random variable without regard to conformities to the physical laws that give rise to color and color variation. One of the techniques of image analysis is to utilize prior knowledge of object colors known as spectral signature analysis. It classifies the pixels in a color image by comparing the pixel values to the characteristic object color values which is measured in advance. It then assigns each pixel with an object label according to which object color is closest to the pixel's value. Other techniques include clustering, region splitting, region growing and edge detection [40].

One essential tool for statistical image analysis is the color histogram or chromaticity histogram. The array histogram indexes are color component ranges and the elements are the number of pixels with the corresponding color value known as histogram bins (Figure 2.2).

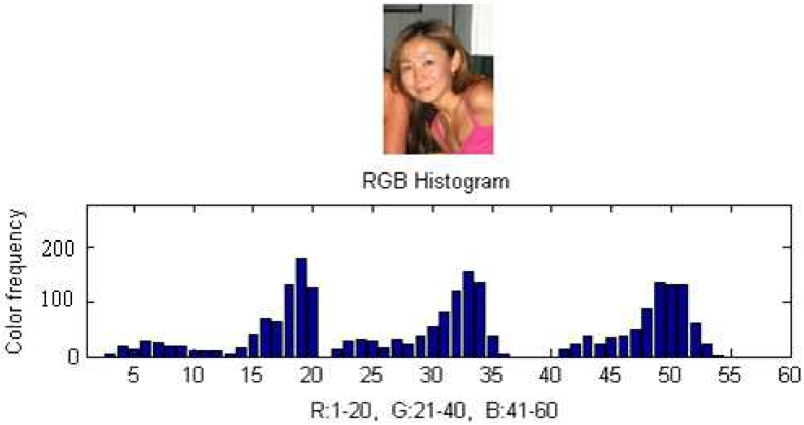


Figure 2.2: RGB Histogram with 20 bins

2.2.2 Physical Approach

In order to understand how color images are displayed, one needs to know how color is created. For example, using heuristic rules to label regions as shadows and highlights or knowledge of color behavior [41, 42, 43]. It is possible to analyze color images by using the SPD of light reflected by a point in the scene; which is dependent on the light illuminating the scene and the reflectance of the surface at that point. The reflectance function varies widely for different points on the same object due to the changes in imaging geometry. Figure 2.3 shows the color of an orange ball, which is non-homogeneous.

The dichromatic reflection model [41, 42, 43], is a model for object reflectance

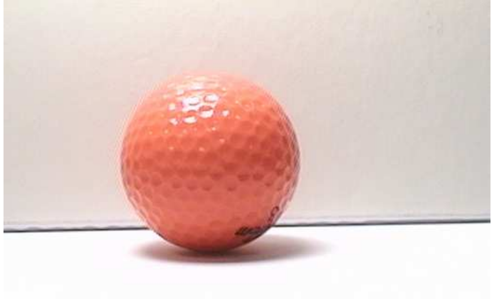


Figure 2.3: Interaction of light with an orange golf-ball

of non-homogeneous surfaces, such as plastics, paints, ceramics and paper. This model is used to determine the color of illumination and to segment an image into regions of uniform color without being deceived by highlights.

2.2.3 Perceptual Approach

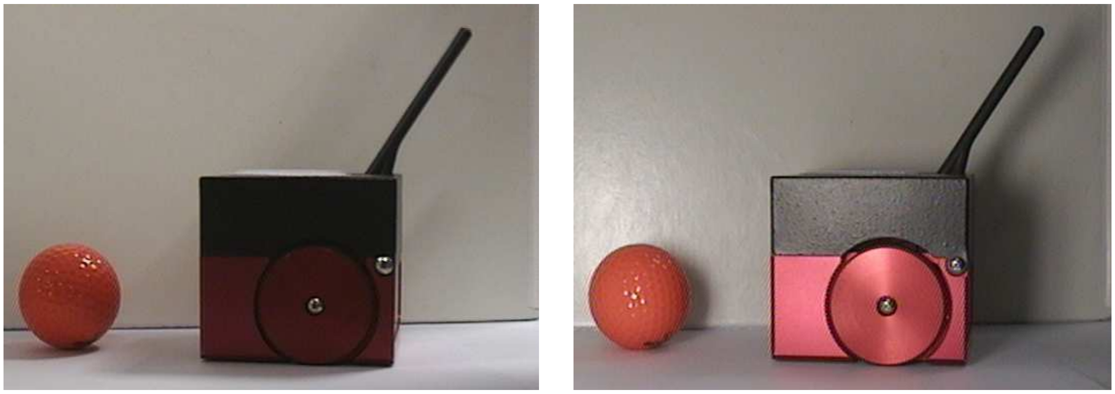
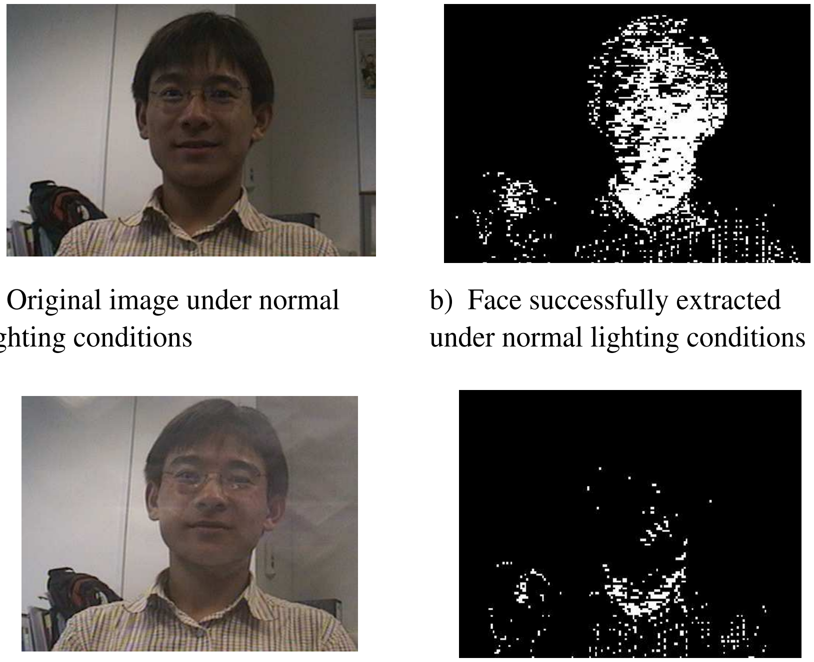


Figure 2.4: Camera's view under two different lighting conditions

The perceptual approach interprets color as a perceptual variable in human vision. Much of the work concentrates on simulating the human visual system in order to adapt its methods to machine vision. However, the exact mechanisms of human color perception remain unknown. One important area of study is color constancy, the phenomenon that allows humans to see the color of the objects uniformly under widely varying illumination. Figure 2.4 presents a CCD (Charge-Coupled Device) camera's view of the same scene under two different lighting conditions. There is a significant difference between the colors in these two cases.

2.3 Color Constancy

There are numerous light sources that affects color constancy. The variation in illumination levels poses serious challenges in computer vision. This is in addition to the variations in camera's operating parameters and other hardware related issues. Figure 2.5 illustrates the problem of varying lighting condition without adaptation which results in segmentation failure.



a) Original image under normal lighting conditions

b) Face successfully extracted under normal lighting conditions

c) Lighting conditions altered

d) Segmentation fails without proper adaptation

Figure 2.5: Segmentation without adaptation to different lighting conditions

The segmentation is performed by thresholding the U and V channel by a constant threshold. Under different lighting conditions, the object color changes and there is a need to vary the threshold according to the changing lighting conditions. One way of tackling this is to continuously sample and examine the image to find a suitable threshold for the prevailing conditions. Object colors do not vary in an arbitrary manner under different lighting conditions. In fact, it tends to follow a certain locus determined by the color of the light source in the chromaticity space [44, 45]. Experiments are conducted to model skin color under varying lighting conditions and results have shown that skin color varies within a small region in the color space under different lighting conditions (Chapter 5).

Due to the non-uniform lighting in the environment, the 3D shape of an object can be “deformed” resulting in the so called “inter-reflection” phenomenon. Certain part of the object can be highlighted, whilst other parts are shadowed by itself or by another object. Furthermore, the object can also be illuminated by light reflected from other object in the scene, creating more challenges for color image segmentation. As a result, the object color can vary across the object image itself. Figure 2.6 shows the “inter-reflection” phenomenon, for an orange golf ball image.

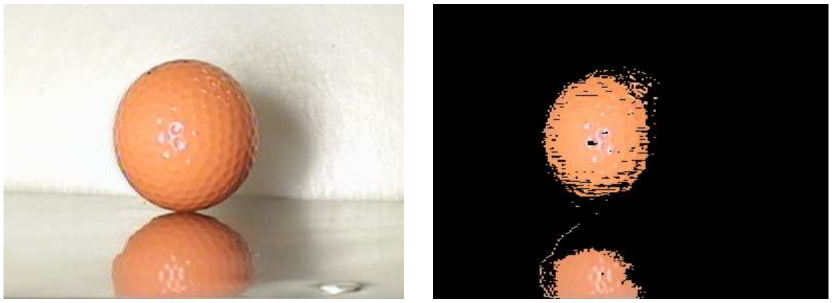


Figure 2.6: Inter Reflection of an object

The segmentation is performed by thresolding the U and V channel of the image so as to extract the ball. However, part of the reflected surface is segmented as belonging to the ball. In addition, the small highlighted area on the ball has been regarded as the background.

2.4 Operating Parameters

CCD cameras have many adjustable operating parameters. These parameters exert influence to the colors of the viewing objects. Aperture, exposure time and gain-control settings are parameters that affect the RGB (Red, Green, Blue) color of the acquired images.

Modification to the original colors of an object, known as color-distortion, is acceptable when the camera operates within it's dynamic range with an intensity-independent color space. White balancing has more damaging effects in all of the color space as it tends to distort the objects' "true"color.

The hardware-related issues that affect pixel values and the overall image quality are, the dynamic range, optics system employed, total number of CCD elements, video coding, gamma correction and noise.

2.5 Color Spaces

A color space is a method used for explaining the properties or the behavior of color within a particular context. There are different mathematical models or spaces in digital color imaging. Some of the color spaces are hardware oriented and others are developed for the ease-of-use. It allows convenient specification of colors within a color gamut, that is, a subset of all the visible chromaticity. In general, the choice of an appropriate color space is crucial for the performance and robustness of a color imaging system. Challenges related to variation of the illumination are tackled by using a color space that differentiates intensity from the actual color information into different components. As a result, the use of a suitable color space makes adaptive segmentation easier.

2.5.1 RGB Color Space

The RGB color space is the most well known hardware-oriented space. This space is intensively used in image-capturing and image-processing related appliances such as CCD cameras, frame grabbers and video displays.

The RGB space employs a Cartesian coordinate system. The three primary colors, Red, Green and Blue are additive, that is, the individual contributions of the primary color is combined to yield the required color. The RGB cube is shown in Figure 2.7.

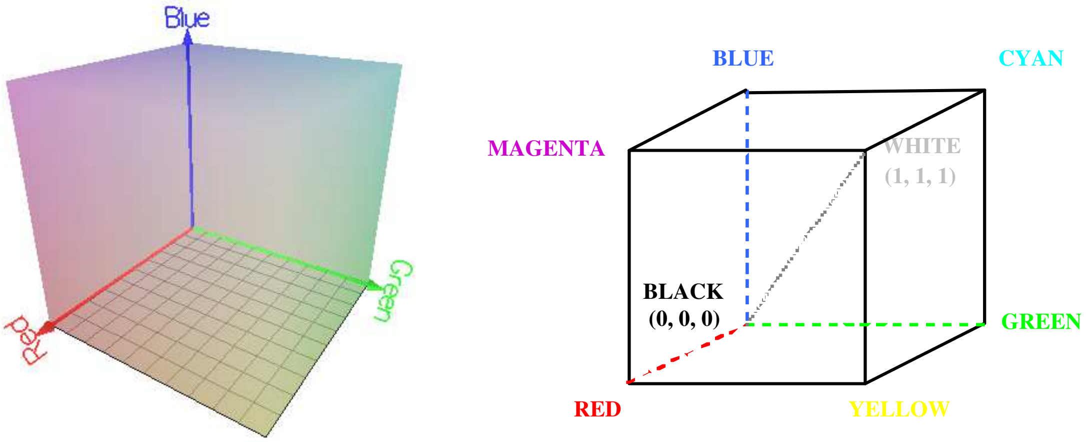


Figure 2.7: Normalized RGB Color cube

The corner (0, 0, 0) represents black at the origin and (1, 1, 1) corresponds to white in the opposite corner of the cube. All the three color components are at equilibrium on the main diagonal of the cube, known as the gray diagonal. The most severe drawback of the color cube is the color constancy since all the three color components contain intensity illumination information. Any changes in the illumination affect all the three primary colors. As a result, RGB needs to be normalized with respect to intensity before use.

2.5.2 YUV Color Space

The hardware-oriented YUV color space is used by the PAL (Phase Alternation Line), NTSC (National Television System Committee) and SECAM (Sequentiel Couleur avec Mmoire) for composite color video standards. Y represents the luminance component or brightness. U and V are chrominance components.

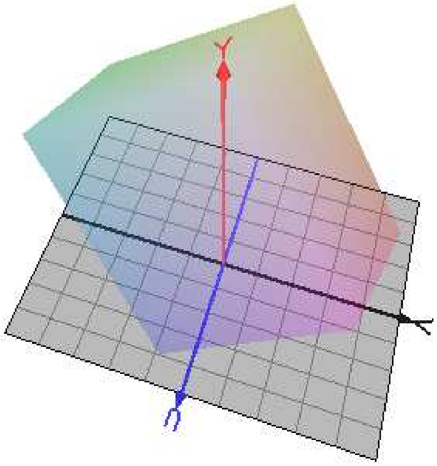


Figure 2.8: YUV Color Space

YUV signals are created from the original RGB source. The weighted values of R, G and B are added to produce a single Y that represents the overall brightness or luminance.

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.5 \\ 0.5 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}, \tag{2.1}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.40162 \\ 1 & -0.34316 & -0.71413 \\ 1 & 1.77216 & 0 \end{bmatrix} \begin{bmatrix} Y \\ U \\ V \end{bmatrix}. \tag{2.2}$$

Hua et. al. [46] uses this space for real-time color image analysis. This space is also used in the YuJin [47] Robot soccer system.

2.5.3 YIQ Color Space

YIQ color space is similar to YUV color space but with color axes rotated at 33^o with respect to the U and V axes of the YUV definition.

$$\begin{bmatrix} I \\ Q \end{bmatrix} = \begin{bmatrix} -\sin(33^o) & \cos(33^o) \\ \cos(33^o) & -\sin(33^o) \end{bmatrix} \begin{bmatrix} U \\ V \end{bmatrix}. \tag{2.3}$$

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.511 & -0.171 & -0.314 \\ 0.131 & -0.506 & 0.375 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \tag{2.4}$$

$$\begin{bmatrix} R \\ \\ G \\ \\ B \end{bmatrix} = \begin{bmatrix} 1 & 1.176 & 0.764 \\ \\ 1 & -0.411 & -0.678 \\ \\ 1 & -0.965 & 1.486 \end{bmatrix} \begin{bmatrix} Y \\ \\ I \\ \\ Q \end{bmatrix}. \tag{2.5}$$

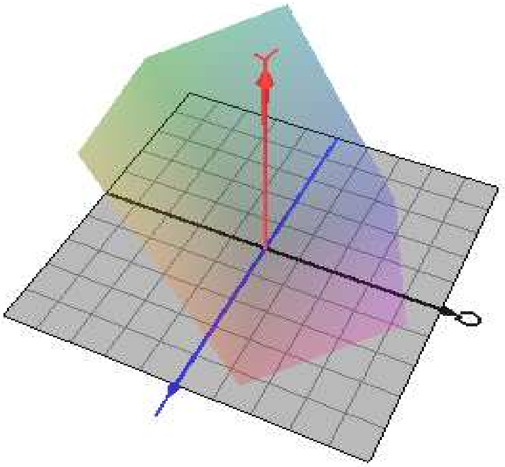


Figure 2.9: YIQ Color Space

This color space is adopted by the early NTSC systems.

2.5.4 YC_rC_b Color Space

YUV 4:2:2 or YC_rC_b is another hardware oriented color space. It is a scaled and offset version of the YUV space. It is part of the ITU-R BT.601 [48] standard. YC_rC_b is a digital color system, while YUV and YIQ are analog spaces for the Phase Alternation Line (PAL) and National Television system Committee (NTSC) systems respectively. The luminance component Y has an excursion of 219 and an offset of +16. This coding places black at 16 and white at 235. The chrominance components C_b and C_r have a nominal range of 16 to 240 inclusively. There are several sampling formats, such as 4:4:4, 4:2:2, 4:1:1 and 4:2:0. Conversion from RGB is presented in equations (2.6), (2.7) and (2.8).

$$Y = c_r R + c_g G + c_b B, \tag{2.6}$$

$$C_B = \frac{B - Y}{2 - 2c_b}, \tag{2.7}$$

$$C_R = \frac{R - Y}{2 - 2c_r}. \tag{2.8}$$

There are several different alternatives for conversion coefficients (Table 2.1).

Table 2.1: Conversion coefficients

Recommendation	c_r	c_g	c_b
Rec601-1	0.2989	0.5866	0.1145
Rec709	0.2126	0.7152	0.0722
ITU	0.2220	0.7067	0.713

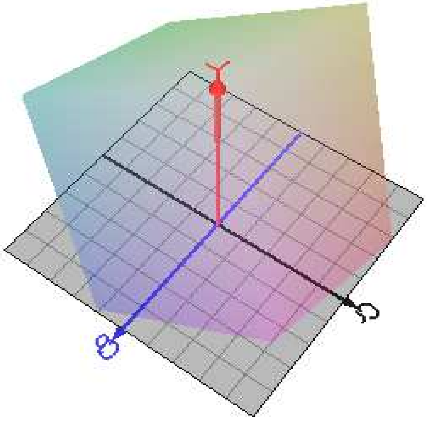


Figure 2.10: YC_rC_b Color Space

LL Ling et. al. [49] uses the YC_rC_b color space for real-time face detection and tracking with an autonomous mobile robot.

2.5.5 Other Color Space

Other color space approaches such as HSI (Hue, Saturation, Intensity), HSV (Hue, Saturation, Value), HSL (Hue, Saturation, Lightness) and HDI (Hue, Distance, Intensity) describe the UV plane in polar coordinates by means of a vector with length, S and rotational color angle, H. The luminance, I, V and L corresponds to the Y of the YUV space. These spaces are highly intuitive to human understanding of color. Hue indicates the color of concern such as red, green or blue. Saturation shows the richness, such as bright red. Intensity represents the density of the color, for example light green or dark green.

At low Saturation, Hue becomes unreliable and is to be discarded. Likewise, very high and low intensity result in unreliable Hue and Saturation [13, 30].

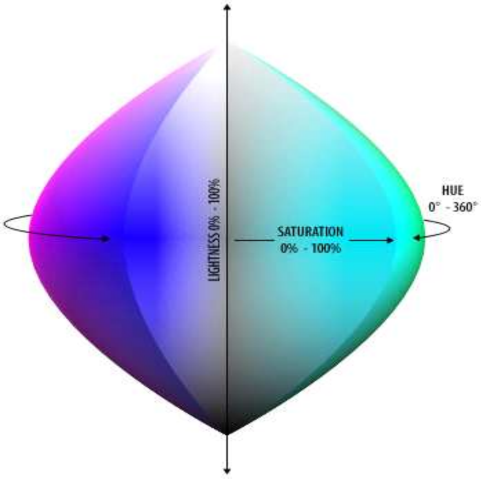


Figure 2.11: HSL Color Space

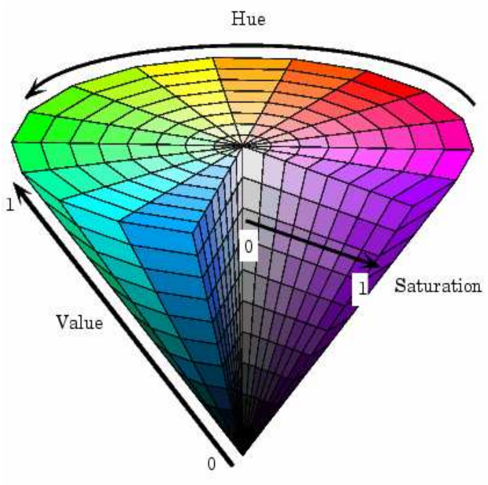


Figure 2.12: HSV Color Space

The use of HSV is preferred in color image segmentation as it differentiates the intensity component from the chromaticity components. It is possible to use only Hue when few colors are used in the application or there is a need to adapt to changing illumination [13, 30]. These spaces are generally not supported by

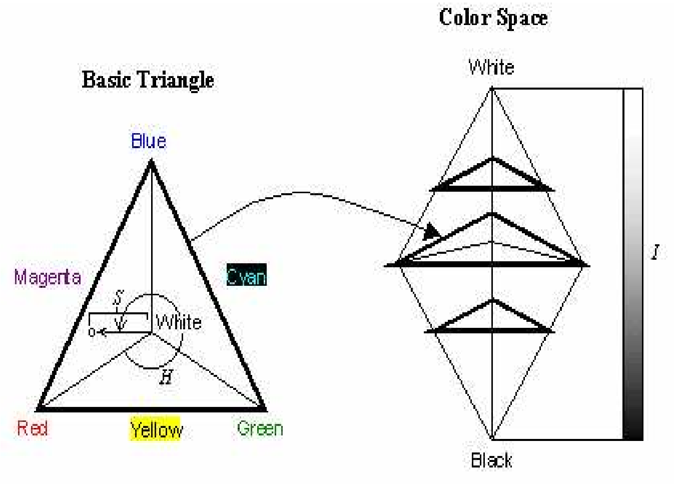


Figure 2.13: HSI Color Space

most image acquisition hardware and result in the need to convert into other color spaces such as YUV. Conversion from RGB to HSI is defined in equations (2.9), (2.10) and (2.11).

$$H = \arctan(\frac{\beta}{\alpha}), \tag{2.9}$$

$$S = \sqrt{\alpha^2 + \beta^2}, \tag{2.10}$$

$$I = \frac{R + G + B}{3}, \tag{2.11}$$

where $\alpha = R - \frac{1}{2}(G + B)$ and $\beta = \frac{\sqrt{3}}{2}(G - B)$.

The HSI color space conversion is computationally demanding and is not suitable for real-time operation. However, the use of a look-up table for conversion reduces the computational load. Another method is to use the HSV color space. The conversion equations from RGB to HSV contain only elementary arithmetical operations and conditional clauses which make the conversion more effective.

RGB to HSV conversion algorithm:

1. Find the maximum and minimum values for the R,G,B.
2. Let $\Delta C = Max(RGB) - Min(RGB)$.
3. Let $S = \frac{\Delta C}{Max(RGB)}$ and $V = Max(RGB)$ for all $Max(RGB) \neq 0$.
4. If R is the maximum value, $h = \frac{(G-B)}{\Delta C}$.

   If G is the maximum value, $h = \frac{2+(B-R)}{\Delta C}$.

   If B is the maximum value, $h = \frac{4+(R-G)}{\Delta C}$.
5. $H = h + 60^o$ for all $h > 0$.

HSI and its relative color spaces are very popular for color-based segmentation. Wei [50] uses HSI, for robustness against the changing intensity of illumination. Kravtchenko and Little [51] suggest the use of HSI like HDI, to preserve the geometrical distances in the units used to measure RGB values to avoid describing saturated pixels. Pingali [52] uses HS to detect the color of a tennis ball. HSI is used by Yoo and Oh [53] and Bradski [30] for face segmentation as well as McKenna [54] for more general color-based segmentation. In the RoboCup scenario, these spaces have also been used in [55, 56, 57].

2.5.6 NCC Color Space

The Normalized Color Components color space is the simplest of the color models that eliminates the intensity component of pixel colors. The use of this space is encouraged by the simple and linear conversion from RGB, which is presented in equations (2.12), (2.13) and (2.14). Assuming $R + G + B \neq 0$.

$$r = \frac{R}{R + G + B}, \tag{2.12}$$

$$g = \frac{G}{R + G + B}, \tag{2.13}$$

$$b = \frac{B}{R + G + B}. \tag{2.14}$$

Two color components are enough to describe chromaticity. Therefore, the redundant third normalized component is discarded as $r + g + b = 1$. It is common to use r and g and discard b, which is referred to as Normalized RG. Any two normalized components can be utilized to describe color. NCC has been frequently used in real-time applications, although mainly in skin detection. Soriano et. al. [44] used a normalized RG space in the skin color model which provided good results.

2.6 Discussion

Understanding the physical conformity governing the sensation of color is essential for the development of a color based vision system. Color is not merely a property of some object, but dependent on the spectral properties and the intensity of illumination. It is accurate to consider color as a statistical quantity in a standardized environment, for example, a laboratory with constant lighting or an static industrial environment. This approach is not sufficient in the real and uncontrolled environment. The physical approach helps to cope with issues that are caused by the interaction between light; such as highlights and inter-reflections. The perceptual approach takes into consideration of color constancy. For machine vision, color constancy is one of the greatest challenges. It is yet unclear how human vision adapts itself to ever changing illumination. Hence, it is not possible to try to adopt the mechanisms of human vision and apply them to machine vision.

Another fundamental issue is the need for hardware that are able to capture color images for analysis. There are numerous variety of cameras, computers and video boards available for machine vision system. Images obtained through such hardware are constrained by their properties. For real-time applications, the performance of the computer platform is crucial due to large amounts of computational load. It is usually not possible to perform image-processing in real time, unless the resolutions are lesser than 320 x 240 pixels.

Color constancy is a major challenge in color-based segmentation. The first major factor that contributes to color information is the camera. Camera is a key component in a color imaging system. In normal human environments, the intensity of lighting notably varies. The use of automatic iris and gain control enables the camera to operate in its dynamic range. In this case, a suitable intensity-independent color representation must be selected for successful segmentation. The use of automatic white balance however is discouraged as it tends to distort the color seen by the camera.

Another major issue is the variations in illumination. It is tackled by using a combination of camera settings and color spaces as discussed. In addition, different light sources have different color temperatures which make the colors look different. This is a real challenge. It is solved with adaptive segmentation or by making the color models of the objects robust to change in illumination.

The selection of an appropriate color space depends on the kind of target that is being tracked. The effectiveness and robustness are also to be taken into consideration while selecting the color space. Effectiveness means reduction in computational load such as using RGB/HSI color space conversion-look-up-table during color space conversion. In addition, fewer linear function are used during thresholding since computational load increases geometrically with the dimensionality of the color space. Robustness means to overcome an object colors dependency on varying illumination such as using YUV color space in place of RGB color space.

There is a good consensus about the benefits of intensity-independent color representations. For a mobile service robot, color perception independent of the light source is a key issue. Light in the environment comes from a large variety of sources and causes the perceived object colors to vary. The human visual system adapts itself to the varying conditions with ease but color constancy remains a challenge for machines.

Chapter 3

Object Segmentation

Segmentation is a process that subdivides an image into constituent parts or objects. It is a step that extracts objects from an image for image-processing such as recognition and tracking. Color is a powerful visual cue for image segmentation, object-recognition and tracking. The desired capability is to endure orientation-variance under varying lighting conditions and under natural transformations such as translation, rotation and scaling. Orientation-variance is an attractive approach in segmentation. The speed of the segmentation is another crucial factor in a real time system and computationally costly segmentation methods are to be restricted.

3.1 Classification

Pixel labeling or classification is the first step in segmentation. There are numerous methods used in classification or labeling theory. One of the methods is to use color as a criterion to label the pixels into some discrete classes. Labeling in the feature space includes thresholding, probabilistic-methods and clustering. Region-based method applies "similarity" as a criterion in the image domain during classification.

3.1.1 Thresholding

Thresholding in color space requires the object color-histograms to be known or measured in advance. Automatic thresholding involves sampling the colors of the objects and using the mean and variance of each color component to define the thresholds [57]. Thresholding distinguishes pixels with values within the bounds in the color space from those outside. The resultant binary image classifies the pixels as belonging to some objects which have a non-zero label with the background pixels assigned with zero label. It is possible to assign a non-zero label to the background pixels and a zero label to the targeted object.

Color space thresholding is a special case of linear color thresholding. It employs constant thresholds for color components thus bounding an object in a two/three-dimensional color space. As shown in Figure 3.1, pixels whose color values inside the tablet are labeled with a non-zero label. The other pixels regarded as the background are assigned with a zero label.

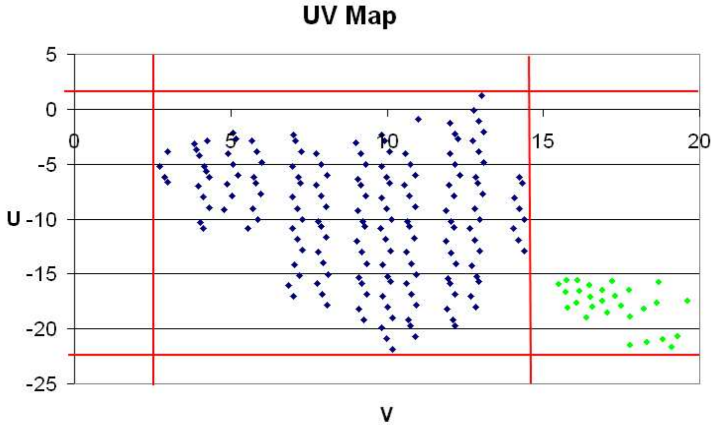


Figure 3.1: Color Space Thresholding in the UV Color Space

Color space thresholding performs well in real-time applications when the object to be segmented is of uniform color. The computational weight and the complexity of color space thresholding increases with the increasing dimensionality of the thresholding plane. Alternatively, the use of a look-up table reduces the computational load but with a tendency to increase the memory usage. In the case of varying object color, such as the human skin color; it is more suitable to use linear color thresholding or a model based approach.

In linear color thresholding, the constant thresholds are replaced by some linear boundaries or planes in the color space (Figure 3.2). This color distribution is similar to Figure 3.1. The same object is now thresholded by some linear functions to obtain better results. Linear thresholding shares the same complexity as color space thresholding. The complexity increases with increased dimensions of the thresholding plane. Since the numbers of linear functions are not restricted and the thresholds are not constant, it involves extra computation for each pixel.

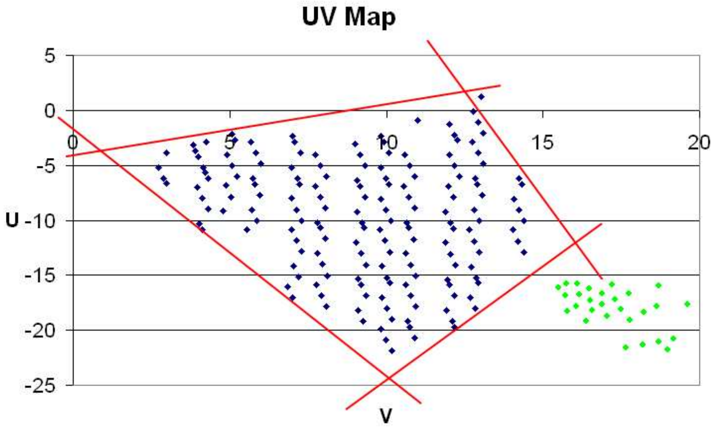


Figure 3.2: Linear Thresholding in the UV Color Space

3.1.2 Probabilistic Techniques

Probabilistic segmentation technique is a method used to model the object color using color distributions. There are two probabilistic segmentation methods, the parametric and non-parametric.

For the parametric method, the color distribution of the object being segmented is modeled by using one or more parameterized Gaussian distributions. The model is a probability distribution function (PDF) of the object color (Figure 3.3).

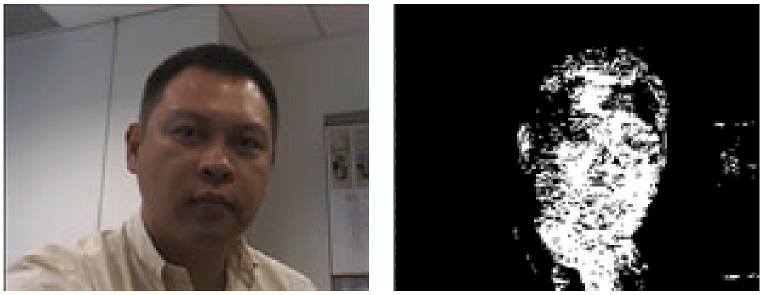


Figure 3.3: Probability Distribution Function of a face

The conditional density for a pixel x belonging to an object, D, is modeled as a Gaussian mixture with component densities as in equation (3.1).

$$p(x|D) = \sum_{j=1}^{m} p(x|D)P(j), \tag{3.1}$$

where a mixing parameter $P(j)$ corresponds to the prior probability that pixel x is generated by the *jth* component. Each mixture component is a Gaussian with mean μ and covariance matrix Σ. Equation (3.2) shows the mixture component in a two-dimensional color space.

$$p(x|D) = \frac{1}{2\pi|\Sigma_j|^{\frac{1}{2}}} \exp^{-\frac{1}{2}(x-\mu_j)^T \sum_j^{-1}(x-\mu_j)}. \tag{3.2}$$

The solution to find an algorithm for fitting a mixture model to a set of training data is to use the Expectation-Maximization (EM) algorithm. This is a maximum likelihood algorithm. Once the model has been created, an image is segmented by comparing the pixel color values to the model and assigning a corresponding probability value to each pixel.

The most popular non-parametric method uses the color histogram and the background subtraction approach. The objective of this model is to capture the recent information about the sequence, and to continuously update the information and the changes in the background.

Let $x_1, x_2, ..., x_N$ be the recent samples of the intensity for a pixel. The probability density function of this pixel having an intensity value x_t at time t is non-parametrically estimated by using the kernel estimator K as,

$$P_r(x_t) = \frac{1}{N}\sum_{i=1}^{N} K(x_t - x_i). \tag{3.3}$$

If the kernel estimator function, K is chosen to be a Normal function $N(0, \Sigma)$, where Σ represents the kernel function bandwidth, then the density is estimated as,

$$P_r(x_t) = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}} \exp^{-\frac{1}{2}(x_t-x_i)^T \sum^{-1}(x_t-x_i)}. \tag{3.4}$$

Assuming independence between different color channels with different kernel bandwidths σ_j^2 for the *jth* color channel, then

$$\Sigma = \begin{pmatrix} \sigma_1^2 & 0 & 0 \\ 0 & \sigma_2^2 & 0 \\ 0 & 0 & \sigma_3^2 \end{pmatrix}, \tag{3.5}$$

and the density estimation is reduced to,

$$P_r(x_t) = \frac{1}{N}\sum_{i=1}^{N}\prod_{j=1}^{d}\frac{1}{\sqrt{2\pi\sigma_j^2}}\exp^{-\frac{1}{2}\frac{(x_{tj}-x_{ij})^2}{\sigma_j^2}}, \tag{3.6}$$

where d is the dimension of Σ.

Using equation 3.6, the pixel is considered a foreground pixel if $P_r(x_t) < th$ where the threshold 'th' is a global threshold over the entire image.

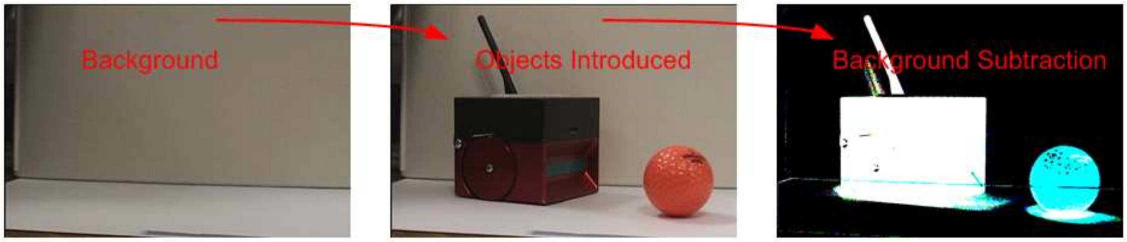


Figure 3.4: Background Subtraction Approach

3.1.3 Clustering

Clustering and unsupervised learning seek to extract information from unlabeled samples. If the unlabeled distribution comes from a mixture of component densities described by a set of parameters θ, then θ is estimated by the Bayesian or maximum-likelihood methods. A more general approach is to define some measure of similarity between the two clusters. A global criterion such as a sum-squared-error, trace of a scatter matrix or locally stepwise optimal iterative algorithms such as k-means and fuzzy k-means clustering are used as required.

Figure 3.5 shows 2 objects in the RGB color space. Cluster A corresponds to the robot and cluster B relates to the orange ball. There are various techniques to simplify computation and to accelerate convergence. k-means is the most popular and famous clustering algorithm.

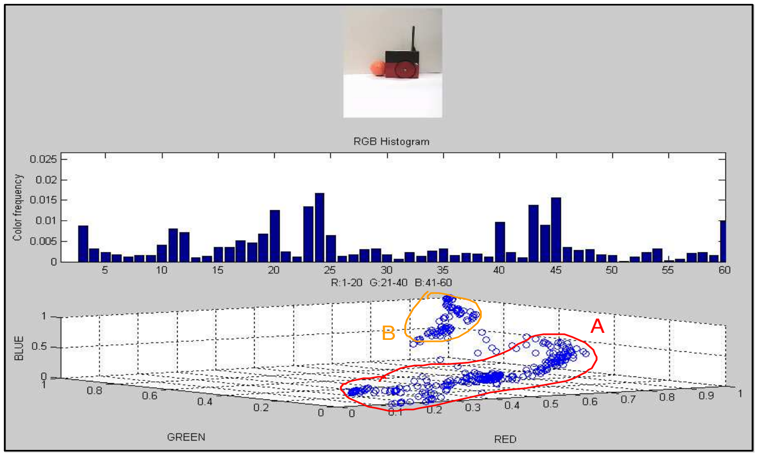
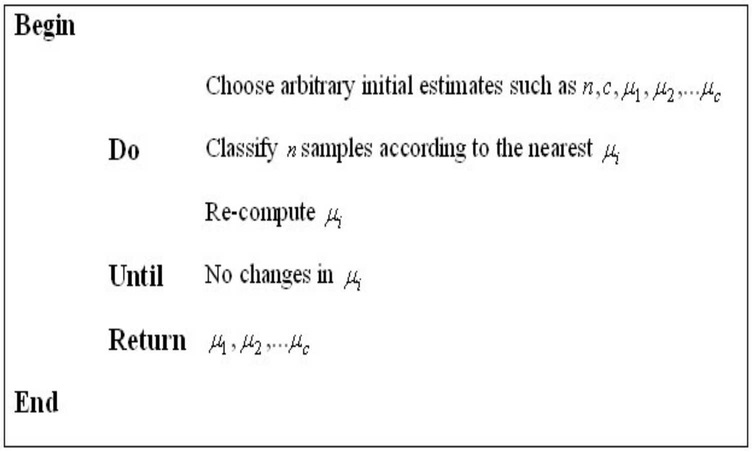


Figure 3.5: Objects in the RGB Color Space

The pseudo code of the k-means algorithm is shown in Figure 3.6.



```
Begin
            Choose arbitrary initial estimates such as $n, c, \mu_1, \mu_2, \ldots \mu_c$
    Do      Classify $n$ samples according to the nearest $\mu_i$
            Re-compute $\mu_i$
    Until   No changes in $\mu_i$
    Return  $\mu_1, \mu_2, \ldots \mu_c$
End
```

Figure 3.6: k-means algorithm

For clusters with sub-clusters, hierarchical methods are needed. Agglomerative or bottom-up methods start with each sample as a single cluster and iteratively merge chosen "similar" clusters based on distance measure. The divisive or top-down methods start with a single cluster representing the full data set and iteratively split them into smaller clusters, each time seeking the sub-clusters that are most dissimilar.

3.2 Noise Filtering

A segmented image is usually noisy due to color variations, variations in camera parameter, hardware limitations, etc. A classical solution to misclassification is to have some pre-processing, such as performing binary morphology to reduce or eliminate noise.

3.2.1 Binary Morphology

Binary morphology deals with binary image obtained from segmentation. Its primary operations are dilation and erosion.

Dilation is the morphological transformation which combines two sets of elements by using vector addition. For example, if A and B are sets in an N-space (E_N) with elements a and b, where $a = (a_1, ..., a_N)$ and $b = (b_1, ..., b_N)$. The dilation of A by B is the set of all the possible vector sums of the pairs of elements, each from a and b.

$$A \oplus B = \{c \ \in \ E^N \mid c = a + b\}, \quad where\ a \ \in \ A\ and\ b \ \in \ B. \tag{3.7}$$

In practice, set A is a binary image and set B is the structuring element. Or, set B is the shape that acts on set A to produce the outcome of dilation. A common solution is to use a 3x3 neighborhood element to perform the dilation to eliminate noise. Figure 3.7 illustrates the dilation of the binary image set A by the structuring element set B. The resultant set C is expanded and the gap in the middle of A is filled.

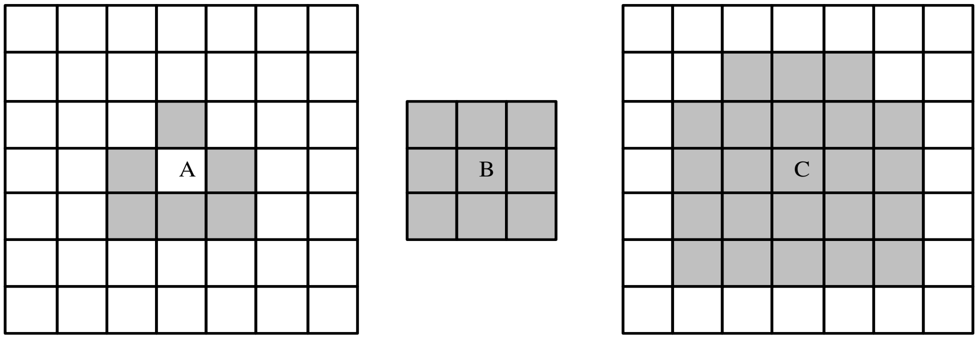


Figure 3.7: Dilation of set A by set B

Erosion is the counterpart of dilation. It combines two sets by using the containment as its basis set. If sets A and B are in Euclidean space, the erosion of set A by set B results in a set that has elements x for which $x + b \epsilon A$ for all $b \epsilon B$. Erosion removes noise and "thins" the image (Figure 3.8).

$$A \ominus B = \{x \ \in \ E^N \mid x + b \ \in \ A \ \}, \quad for\ all\ b \ \in \ B. \qquad (3.8)$$

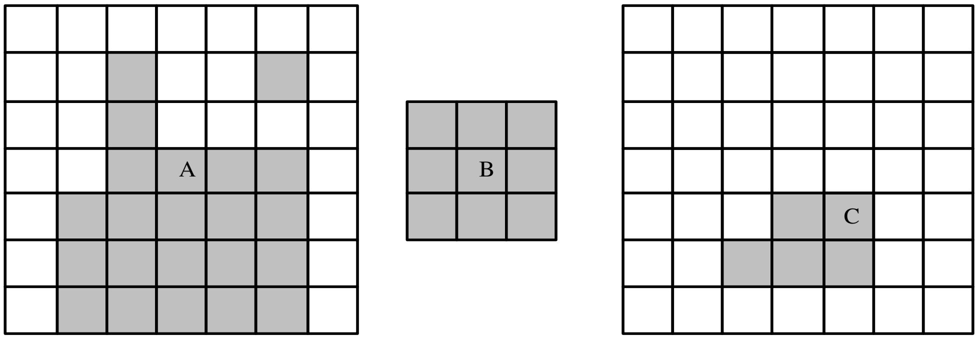


Figure 3.8: Erosion of set A by set B

Morphological opening basically opens up strips, and generally smoothes the contour of an image. It breaks narrow isthmuses and eliminates thin protrusions. It is useful in eliminating noise from segmented image which involves erosion followed by dilation as shown in Equation (3.9).

$$A \circ B = (A \ominus B) \oplus B. \tag{3.9}$$

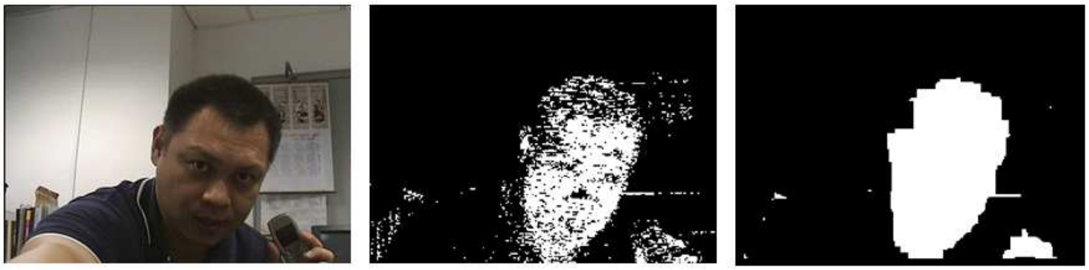


Figure 3.9: Filtering by Morphological Opening Operations

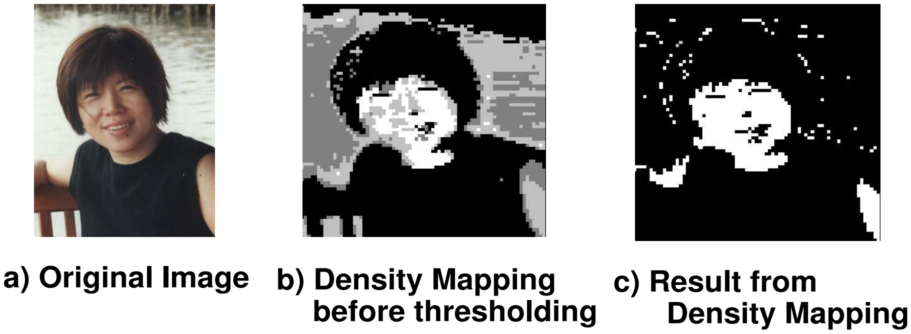
Morphological closing basically closes gaps and tends to smooth contour sections. As opposed to morphological opening, it generally fuses narrow breaks and long thin gulfs, eliminates small holes and fills gaps. It is useful when certain regions in the source image are filled with small gaps and are not uniform. With a suitable structuring element, morphological closing makes the regions more consistent. This operation involves dilation followed by erosion as shown in Equation (3.10). Figure 3.9 shows an example of opening.

$$A \bullet B = (A \oplus B) \ominus B. \tag{3.10}$$

3.2.2 Density Mapping

A density map which reflects the density of the object pixels in the binary image is created with every pixel corresponding to certain sample window. Density mapping is performed on a binary image. The binary image is divided into adjacent sample windows, each sized in N x N pixels. The more the object pixels in the sample window, the higher the density pixel value is. Different gray-scale values interpret the probability of a pixel belonging to an object.

The result of density mapping is a probability image similar to resolution reduction of probabilistic segmentation methods. The density map is next "thresholded" to remove the unwanted region. The final result is a binary image with pixels belonging to the objects.



a) Original Image

b) Density Mapping before thresholding

c) Result from Density Mapping

Figure 3.10: Density Mapping

3.2.3 Effect of Filtering

Another method to eliminate noise from the probability map is to use some basic filters such as an average filter, low-pass filter, median filter, Gaussian filter, etc. These filters have the effect of blurring the image. The use of these filters reduce noise and image resolution. Figure 3.11 illustrates the effect of applying filter before thresholding and applying thrsholding only.

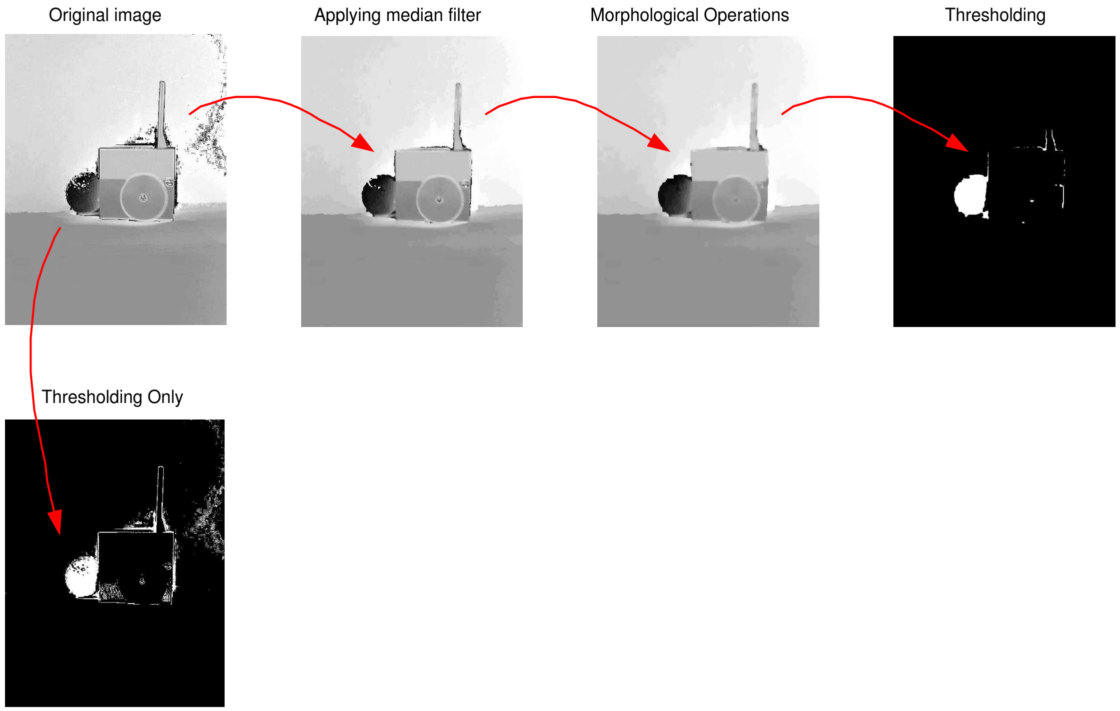


Figure 3.11: Effect of filtering and morphological operations

3.3 Labeling

The labeling of different regions of an image into separate objects can be realized after successful segmentation using pixel colors as the criteria.

3.3.1 Connectivity

The concept of connectivity is used to establish boundaries of objects. Pixels are said to be connected if they are adjacent and the gray levels satisfy a specified criterion of similarity. The 4-connectivity and the 8-connectivity are the most common (Figure 3.12). The result of the labeling process is shown in Figure 3.13.

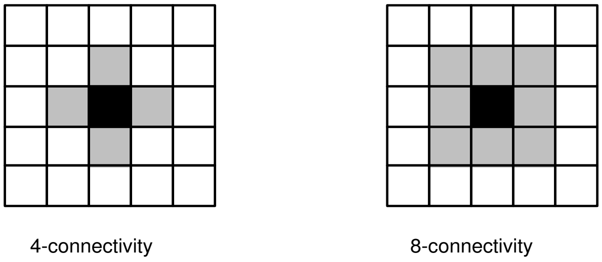


Figure 3.12: Two types of connectivity

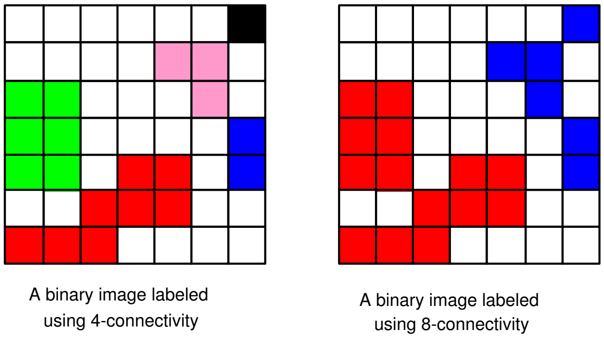

A binary image labeled using 4-connectivity
A binary image labeled using 8-connectivity

Figure 3.13: Results of labeling

3.4 Discussion

Labeling is the initial step in object segmentation and has great impact in determining the reliability of the system. Thresholding techniques are good if the object colors are known and are uniform. Model-based methods work better for multi-colored objects and clustering is used when the colors are not known.

The visual perception of a mobile robot cannot be based on a single segmentation method. A combination of different segmentation methods in a visual tracking system produces a more versatile and robust visual perception. The real-time processing requirement is a challenge. Although single optimized segmentation method works well in a real-time environment, the processing time increases geometrically with the number of concurrent tasks.

Different probabilistic techniques that are adaptive to the changes in illumination have been discussed. These techniques are useful in the domain of mobile robot implementation. Additional processing is needed to filter noise during labeling or binarizing a color probability map. Consideration is therefore needed when using these filters as they tend to increase the processing times. Either connectivity analysis or probability theory is to be used during object-labeling when separating objects with different properties.

Chapter 4

Object Tracking

The basic idea of object tracking is to observe the target's state from one moment to another. This implies that the state variable is measured from the image and the target's state is estimated based on these variable.

The most important sense is vision for humans and color is an important visual stimuli. Color is an effective visual cue and image attribute used in visual detection and tracking. It is computationally feasible to have orientation invariance under varying lighting conditions and invariance to natural transformations such as translation, rotation and scaling. However, the dependence of the perceived color on illumination is a challenge in color-based segmentation. Another challenge during color segmentation is the misclassification of some "close" color objects as the targeted object.

An effective discrimination function is needed to discriminate the "unwanted" objects from the segmented image. The discriminator is based on features calculated from the segmented image and these features include area, circumference, orientation, and spatial moments. When tracking the object's location in image coordinates is required. Each object's spatial location in the image frame is an imperative feature. In practice, this involves calculating the center of the mass of each object or the overall mean of the object's pixel locations. Calculating the object's center of mass in every frame is adequate when only one moving object is in the segmented image. Therefore, a priori knowledge of the object's previous state is needed for tracking. Predictive method such as Kalman filtering are useful in object tracking. In Robot Soccer System, color-based visual tracking has been used [56, 58, 59].

4.1 Feature Extraction

Feature extraction is the first task in pattern recognition and it extreme complexity constitutes the main work of the pattern recognizer. The result of the feature extraction stage is a set of samples, x, that are fed to the classification or decision stage of the recognizer (Figure 1.1).

4.1.1 Area

The object's area in the image plane is one of its fundamental properties. In most cases, area alone is enough to distinguish between wanted and unwanted objects as noise components are usually small.

$$Area = \sum_{(row,column) \in region} Pixel. \tag{4.1}$$

The area of an object is the total number of pixels in the regions that the object occupies.

4.1.2 Centroid

Centroid, or the center of mass, is another fundamental property of a region.

$$\overline{row} = \frac{1}{Area} \sum_{(row,column) \in region} row, \tag{4.2}$$

$$\overline{column} = \frac{1}{Area} \sum_{(row,column) \in region} column. \tag{4.3}$$

The center of the mass for a region is obtained when the sum of the pixel coordinates within the region is divided by the area.

4.1.3 Bounding

Bounding box is another property closely related to tracking and is highly suitable for visualization. It encloses the tracked object and is used for object identification. The bounding box is defined by four coordinates, each denoting one corner of the box.

$$row(\text{min}) = \text{min}\{row|(row, column) \in region\}, \tag{4.4}$$

$$row(\text{max}) = \text{max}\{row|(row, column) \in region\}, \tag{4.5}$$

$$column(\text{min}) = \text{min}\{column|(row, column) \in region\}, \tag{4.6}$$

$$column(\text{max}) = \text{max}\{column|(row, column) \in region\}. \tag{4.7}$$

The bounding box operation involves notable increase in the processing time if the object pixels cover a large area in the segmented image.

4.2 Model-Based Tracking

The traditional approach of tracking is based on modeling the object's state (Figure 4.1). The state variables such as centroid is measured directly from the new image. Variables such as velocity are subjected to estimation based on the measurements of the object's current state. The object's state in following video sequence is then predicted (Prediction) and utilized to optimize the subsequent measurement (Measurement). The variable are then updated (Update Model).

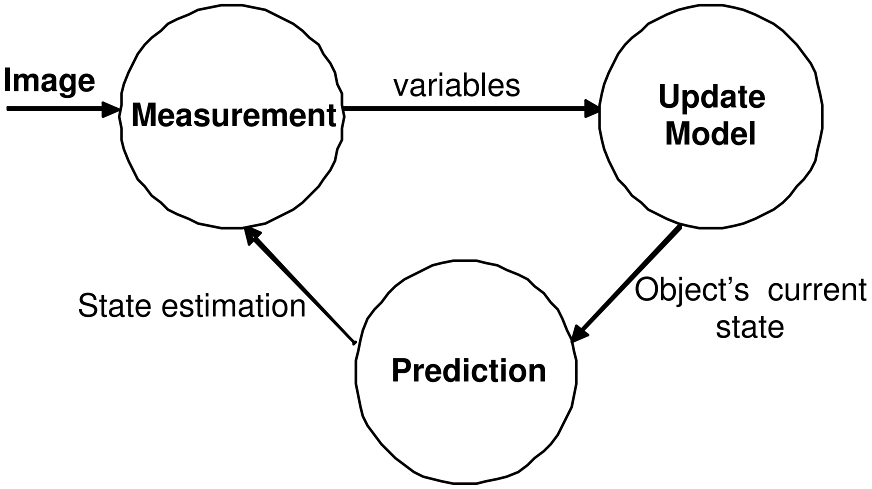


Figure 4.1: Tracking of object's state

A object model contains a dynamic model and a measurement model to determine the object's state.

4.2.1 Dynamic Model

The state is a set of quantities such that given some initial conditions $x(t_0)$; all future inputs $u(t)$, all the future responses $x(t)$ for $t > t_0$ are uniquely determined.

The basic continuous state equations are a set of coupled first order linear differential equations in the state variables.

$$\dot{x} = Ax + Bu, \tag{4.8}$$

$$y = Cx. \tag{4.9}$$

Where

x: state vector, n x 1,

A: open-loop dynamics matrix, n x n,

u: control vector, m x 1,

B: control distribution matrix, n x m,

y: output vector, r x 1, and

C: sensor calibration matrix, r x n.

External forces, such as disturbances, tend to affect the system.

$$\dot{x} = Ax + Bu + \Gamma w. \tag{4.10}$$

Where

w: disturbance vector, d x 1, and

Γ: disturbance distribution matrix, n x d.

For a discrete state system,

$$x_k = Ax_{k-1} + Bu_{k-1} + \Gamma w_{k-1}. \tag{4.11}$$

In visual tracking, the most crucial state variable is the object's location. It is assumed that the objects are moving at a constant velocity and acceleration is considered a random disturbance variable. Thus, the object's location and velocity are also included leading to a motion model [60, 61, 62].

$$\begin{bmatrix} y \\ \dot{y} \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} y \\ \dot{y} \end{bmatrix}_{k} + \begin{bmatrix} \frac{T^2}{2} \\ T \end{bmatrix} \xi_k. \tag{4.12}$$

Where y is a location coordinate(centroid), $\dot{y}$ is the velocity, ξ_k is the random acceleration and T is a constant time interval.

Equation (4.12) is useful for both the horizontal and vertical motion of the object in the image plane, but is not a universal model. The dynamic model can include more state variables such as area and shape measurements to improve tracking robustness.

4.2.2 Measurement Model

The measurement model (4.13), involves issues related to the process of observing the state's variable.

$$v_k = C_k x_k + \eta_k, \tag{4.13}$$

where x_k is a state vector at the k moment and v_k is a measurement vector. η_k is the measurement error and C_k is the observation matrix. In this work, the velocity of the object is not measured directly from the image. However, the object's location is measured, thus, the observation matrix becomes C=[1 0].

4.3 Data Updating

4.3.1 CAMSHIFT Algorithm

CAMSHIFT (Continuously Adaptive Mean SHIFT) tracking algorithm is based on a mean shift algorithm [30, 31, 32]. It is a non-parametric technique for finding the mode of the probability distributions. Unlike the mean shift algorithm, which is designed for static distributions, CAMSHIFT algorithm is designed for changing probability distributions. An object changes its location, shape and area in successive video frames when moving into various directions. The motivation is to make the algorithm capable of coping with changing distributions from the tracking framework.

The mean shift algorithm is an essential part of the CAMSHIFT algorithm. The basic mean shift procedure is given below:

1. Determine a search window size.
2. Determine the initial location of the search window.
3. Compute the mean location in the search window.
4. Center the search window at the mean location computed at step 3.
5. Repeat steps 3 and 4 until the mean converges or when the mean location is less than a specified threshold.

$$M_{00} = \sum_x \sum_y I(x, y), \tag{4.14}$$

$$M_{10} = \sum_x \sum_y xI(x,y), \tag{4.15}$$

$$M_{01} = \sum_x \sum_y yI(x,y), \tag{4.16}$$

$$(x_c, y_c) = (\frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}}). \tag{4.17}$$

Where

M_{00} : is the zeroth moment,

M_{10} : is the first moment of x,

M_{01} : is the first moment of y,

$I(x,y)$: is the pixel (probability) values at position (x,y) in the image, and

x, y : is the range over the search window.

In the CAMSHIFT algorithm, the window size is made adaptive by relying on the zeroth moment information. This CAMSHIFT algorithm sets the search window width and height to a function of the zeroth moment (Section 5.3.3). The CAMSHIFT algorithm operates on color probability images and the steps involved are as follows:

1. Set the region of the probability distribution.
2. Find the initial location of the 2D mean shift search window.
3. Calculate the color probability distribution in an area larger than the mean shift window size.
4. Store the zeroth moment and mean location by mean shift convergence or a number of iterations.

5. Center the search window at the mean location stored at step 4 and update the window size for the next video frame.

6. Repeat steps 3 to 5.

4.3.2 Mean and Standard Deviation

A simple way to perform tracking is to use the mean and the standard deviation of an object's "xy" pixel coordinates to calculate its location and bounding box. In this approach, the tracked object is assumed to be dominant in the image frame.

4.4 Discussion

Most of the tracking systems use Kalman filtering method. The Kalman filtering algorithm has been successfully used in the field of visual tracking [63, 64]. However, the mean shift algorithm is being used recently for probability distributions analysis in the field of real-time visual tracking.

In some simple cases, when only one dominant object is expected in the field of view, the object tracking is performed by calculating the mean and variance of the pixel locations.

A mobile robot visual tracking system needs to track many similar objects such as human faces and hands. Therefore, the Kalman filter and mean shift-based methods are the only feasible solutions for visual tracking.

Chapter 5

Proposed System - Detection and Tracking Modules

The various steps involved in the visual perception of a mobile robot are color-based segmentation, modeling, thresholding and clustering. Color-based perception is made more versatile by using many methods in the same visual tracking system. This has inspired the development of the proposed architectural solution to real-time color-based visual tracking.

To detect and track faces using skin color as the feature to achieve real-time processing speed is proposed in this work. The chrominance components of the YUV and HSV color models are used as it is well adapted to human perception.

The proposed system architecture and the experiments conducted are documented in this chapter.

5.1 System Overview

A Magellan Pro mobile robot from iRobot [65] with a Sony EVI-D30 [66] pan-tilt camera is used as an experimental platform. The camera pans and tilts according to the movement of the detected face. The robot rotates and follows the human subject when necessary to keep the face in view and performs obstacles avoidance along its path.

The robot control system is designed to concurrently control both the camera and the robot while tracking a human subject. The human tracking and obstacles avoidance are done with the information drawn from the image captured by the camera, the sonar scan data, tactile sensor data and the camera position readings with respect to the initial position. Figure 5.1 and 5.2 show the block diagrams of the complete system and the associated modules.

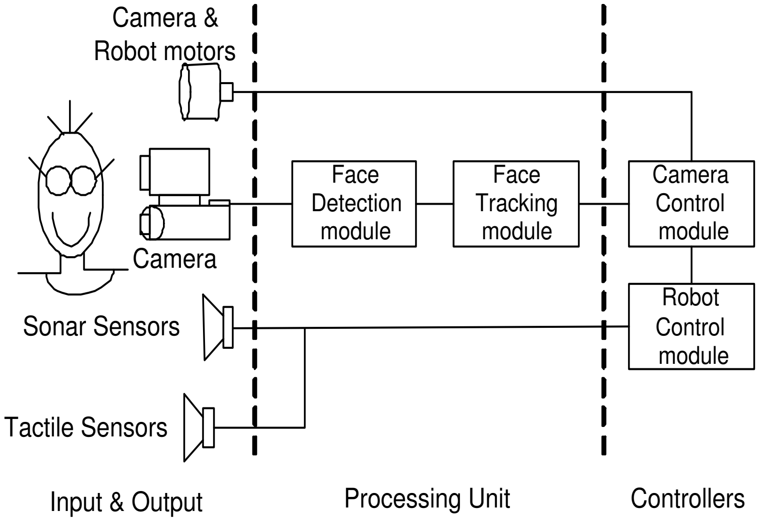


Figure 5.1: Block Diagram of the Robot-Camera Control System

5.2 Deciding the color space

The three key points to note while choosing a suitable color space for image segmentation are target-type, system effectiveness and robustness (Chapter 2).

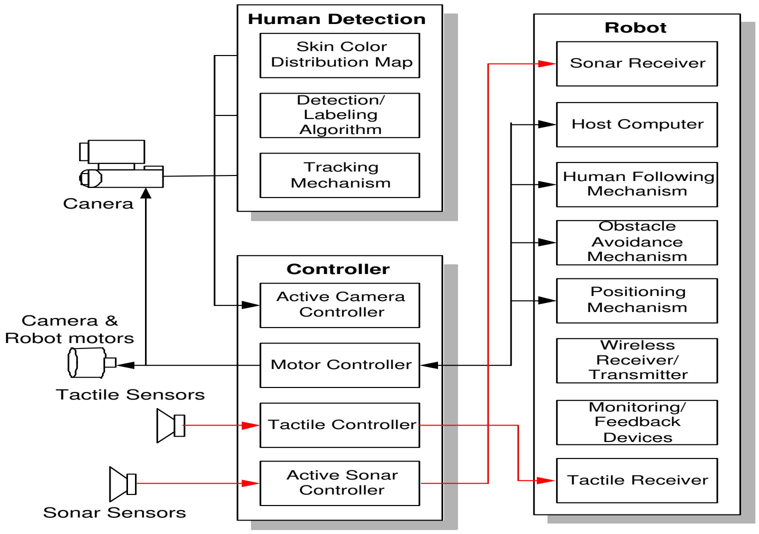


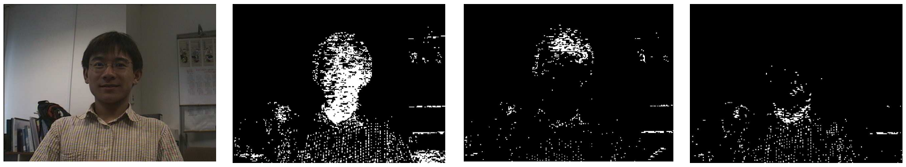
Figure 5.2: System Block Diagram of the various modules

5.2.1 The experiments - Color space determination

Experiments are conducted in the HSV, YUV, YC_rC_b and the NCC color spaces. The results are analyzed and compared to determine a suitable working space.

Two color spaces are selected based on the experiments conducted, the HSV and YUV color spaces. The HSV color space is able to distinguish colored objects by comparing the hue values of the pixels. This space defines colors similar to the way humans do, which makes object extraction using color easier. YUV color space does not need heavy color space conversion (Chapter 2). These two spaces differentiate intensity from chromaticity and are able to cope with varying lighting conditions.

A set of human images (Appendix A) are explored under three different lighting conditions. The color (chromaticity histograms) behaviors of the image under various conditions and different color spaces are analyzed with programs written in the MatLab [67] environment.



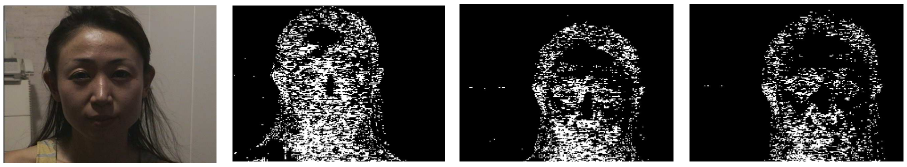
Original Image

Under natural lighting conditions

Under yellow lighting conditions

Under white lighting conditions

Figure 5.3: Skin color probability using the RGB color space



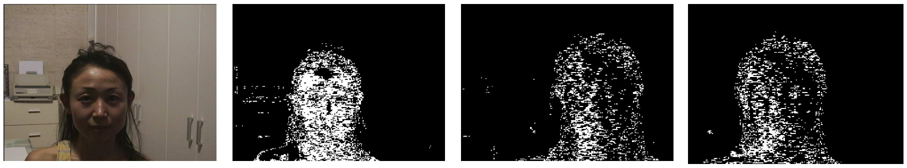
Original Image

Under natural lighting conditions

Under yellow lighting conditions

Under white lighting conditions

Figure 5.4: Skin color probability using the NCC color space



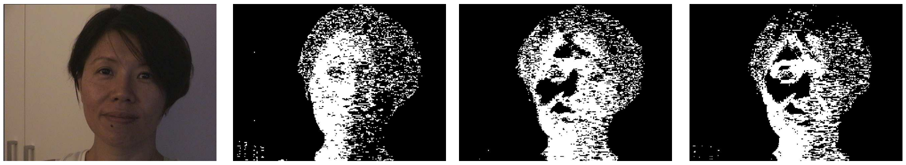
Original Image

Under natural lighting conditions

Under yellow lighting conditions

Under white lighting conditions

Figure 5.5: Skin color probability using the HSV color space



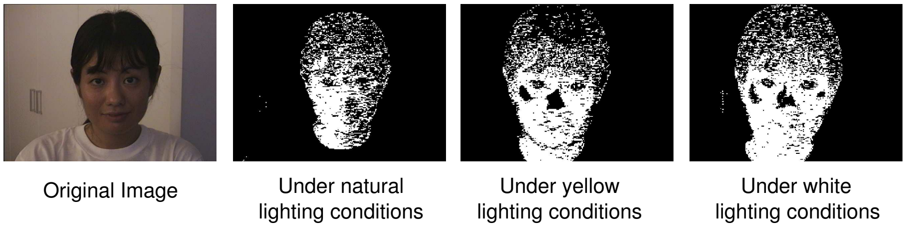
Original Image

Under natural lighting conditions

Under yellow lighting conditions

Under white lighting conditions

Figure 5.6: Skin color probability using the YC_rC_b color space


Original Image

Under natural lighting conditions

Under yellow lighting conditions

Under white lighting conditions

Figure 5.7: Skin color probability using the YUV color space

The following conclusions are drawn based on the experiments conducted:

1. If the objects (Appendix B) have distinctive colors, the H value of the HSV space is enough to label the objects.
2. Automatic white balance tends to corrupt an object's colors.
3. Although color is invariant under varying lighting conditions, it is advisable to design an adaptive system.
4. Color is invariant to natural transformation such as translation (varying viewing distance), rotation and scaling.

5.2.2 The Skin-Color Model

Image retrieval is based on the global image features such as color or texture. To detect and track faces using skin color as the feature to achieve real-time processing speed is proposed in this work. The chrominance components of the YUV and HSV color models are used as it is well adapted to human perception.

YUV color space is used instead of RGB as it separates the luminance from the chrominance, and is useful in compression and image processing. Skin color covers only a small part of the UV plane and the influence of the Y channel is minimum [46, 68, 69] (Section 5.2.1.). Most digital images use the RGB color space; but the individual R, G, and B components vary widely under changing illumination conditions. It is observed that varying skin colors lay in their color intensities rather than in the facial skin color itself [46, 68, 69] (Section 5.2.1.). The luminance part of the YUV color space is discarded as it does not contain useful information for human faces detection. Color is orientation invariance under varying lighting conditions and under natural transformations such as translation, rotation and scaling. Therefore, by discarding the luminance signal, Y, a more manageable UV space is obtained (Figure 5.8).

Figure 5.8: UV 2-dimensional color space

Skin color detection poses some disadvantages too. The influence of luminosity is not totally negligible. The varying skin colors are determined by the color intensities or different shades of a color [70, 71]. This situation worsens as normal fluorescent lamps in the laboratories “flickers”, resulting a non-constant luminance in the testing environment. Other objects which have skin-like color are included in the detection too. Although, the skin color subspace covers a small area of the UV plane, it is impossible to model skin color in such a general way to be efficient for all images. Relaxing the model leads to more false detections while a rigorous model increases the number of dismissals. A tradeoff between being generic and accurate are required. Figure 5.9 illustrates the region on the UV color map where the facial skin color pixels lie. This map is used to localize image pixels that contain facial skin color. As a result, the image pixels that contain facial skin color are segmented to form the corresponding facial skin color regions, which are probable human faces.

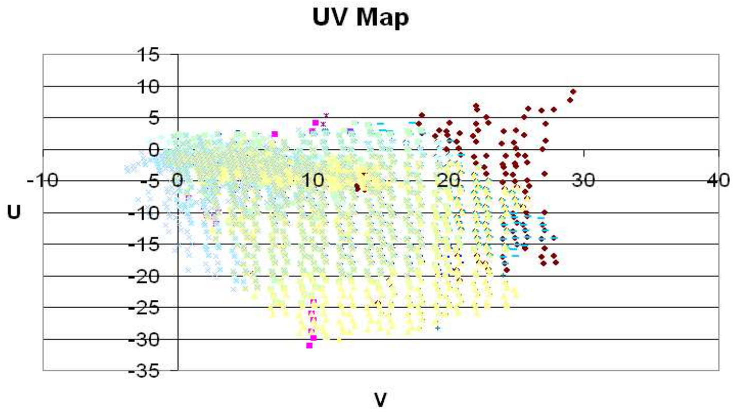


Figure 5.9: Skin color pixel on the UV map

5.2.3 Statistical Derivation of Skin-color model

It is conjectured that pixels in an image belonging to a face exhibit similar U and V values providing good coverage of all human skins [46]. To verify and derive equations to capture this information, thirty-five (35) face images are collected, twenty (20) Chinese faces, ten (10) Indians faces and five (5) Caucasian faces. The face pixels are extracted and fed into a neural network for learning. This learning process is done off-line using Matlab.

A neural network is a weighted graph that models information processing in the human brain. A multilayer perceptron (MLP) neural network is shown in

Figure 5.10. The goal of this network is to create a model that correctly maps the eighty (80) input vectors **X** to the output vector **T** using the pixel probability table and its previous or initial random weighted edge. An input vector at a node is passed along a weighted edge to the twenty (20) neurons of the hidden layer. The weight for the hidden layer neuron is $20(|\mathbf{x}| + 1)$ where $\mathbf{x}$ is the input to the corresponding input layer neuron. If the weighted sum of the signals at a hidden layer neuron exceeds a specified threshold, then that neuron fires. The output from the fired hidden neuron is weighted again and passed on to the output layer. The output vector **T** contains the skin probability of the training sets. The training rate is set to 0.01 with 500 iterations. The training algorithm uses the Levenberg-Marquad [72] algorithm which minimizes the error $||T_{network} - T_{actual}||$, these error deceases with each iteration and the neural model gets closer to producing the desired output. The sigmoid function, $\sigma(t) = \frac{1}{1+e^{-t}}$ is used as the threshold function. Histogram of the UV values are built as shown in Figure 5.16 and Figure 5.17.

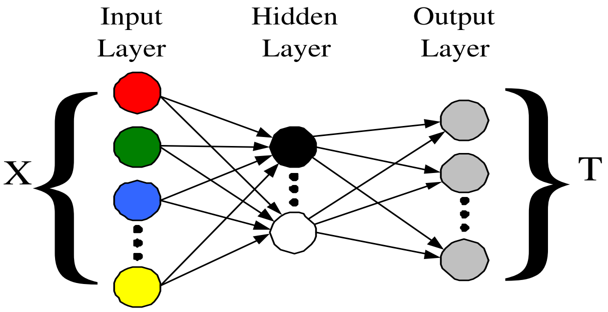


Figure 5.10: A generalized neural network

Thresholding is a technique to distinguish pixels with values within the bounds and those outside the bounds in the color space. This segmentation gives a binary image where pixels are classified into their defined classes.

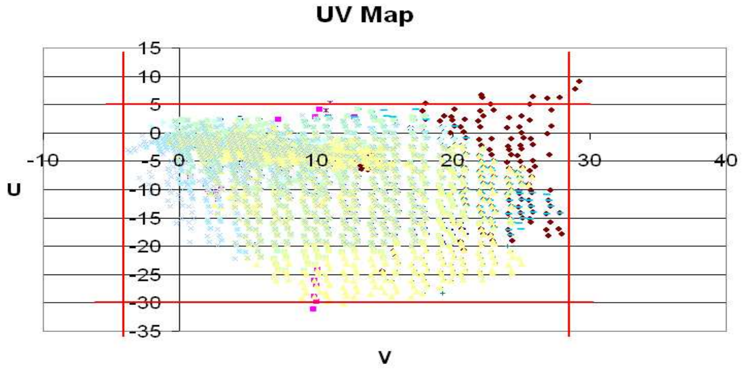


Figure 5.11: Color Space Thresholding in the UV Color Space

A skin color distribution in the UV plane is generated (Figure 5.9). Using the color space thresholding approach, (Figure 5.11), the following equations are obtained:

$$-30 < U < 5, \tag{5.1}$$

$$-4.2 < V < 28.8. \tag{5.2}$$

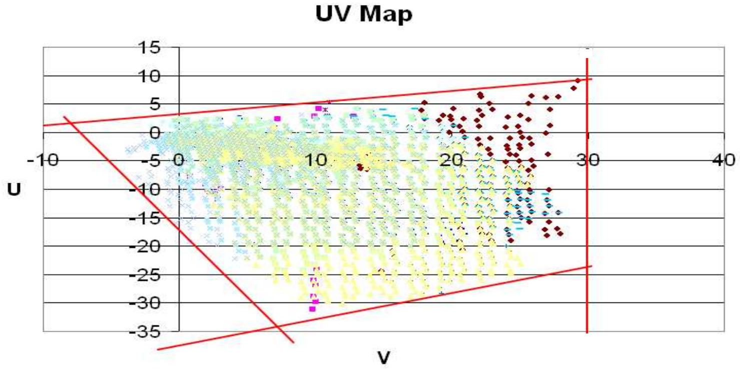


Figure 5.12: Linear Thresholding in the UV Color Space

Equations (5.1) and (5.2) are simple but not effective to classify skin and non-skin pixels. Linear color thresholding approach (5.3 - 5.6) is used next (Figure 5.12).

$$V < 30, \tag{5.3}$$

$$U > 0.45V - 37.65, \tag{5.4}$$

$$U > -2.37V - 17.65, \tag{5.5}$$

$$U < 0.206V + 2.94. \tag{5.6}$$

The equations obtained from linear thresholding are good enough to classify the skin and non-skin pixels. However, misclassification occurs when the pixels are near the boundary plane (5.3 - 5.6).

To improve the accuracy of the classification, the following non-traditional thresholding approach is proposed. By relaxing the model as shown in (Figure 5.13) and with the least square estimation approach; two sets of linear equations (5.7, 5.8) and a quadratic equation (5.9) are obtained. The skin color pixels in the UV color distribution map are bounded by these planes (5.7- 5.9).

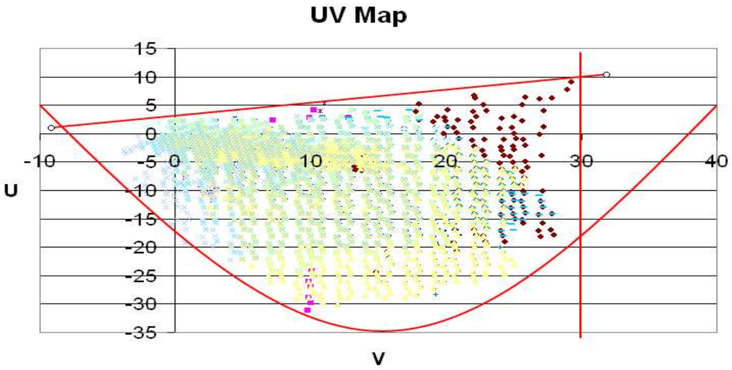


Figure 5.13: Model-based Thresholding in the UV Color Space

$$V < 30, \tag{5.7}$$

$$U < 0.206V + 2.94, \tag{5.8}$$

$$U > 0.08V^2 - 2.4V - 17.2. \tag{5.9}$$

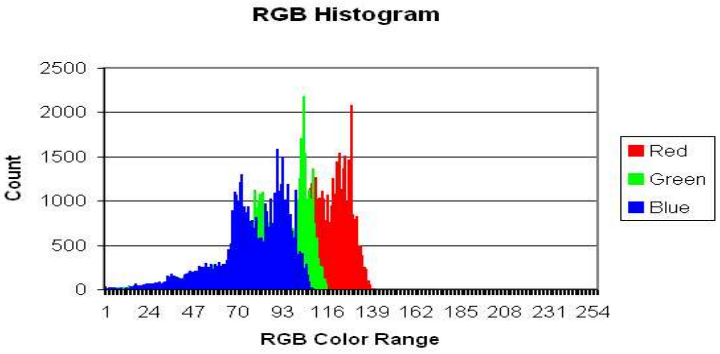


Figure 5.14: RGB Histogram

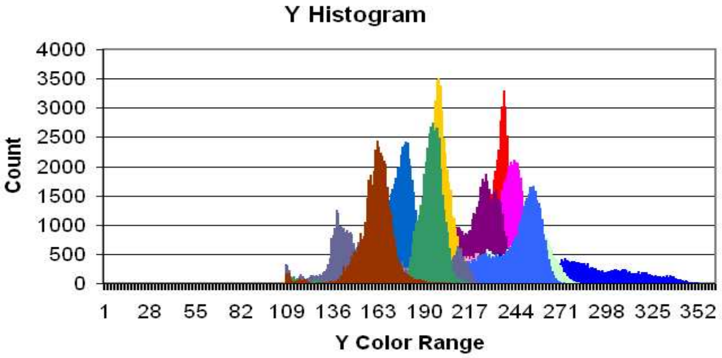


Figure 5.15: Histogram of Y values under varying lighting conditions

RGB color model is not suitable for skin color classification as all the three color components contain the intensity of the illumination (Figure 5.14). Any changes affect all the three primary components. YUV color model is better as it differentiates luminance from chromaticity (Figures 5.15, Figure 5.16, and Figure 5.17).

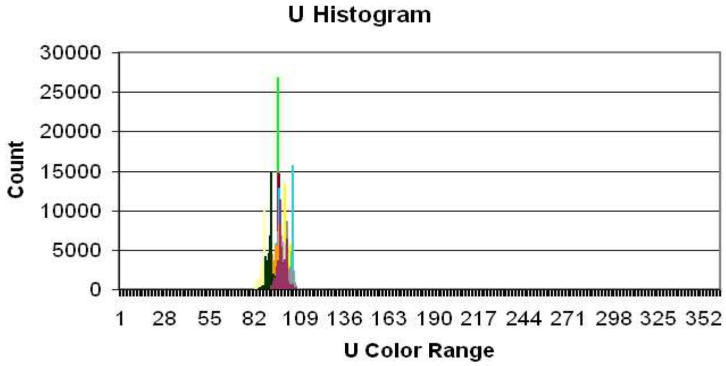


Figure 5.16: Histogram of U values under varying lighting conditions

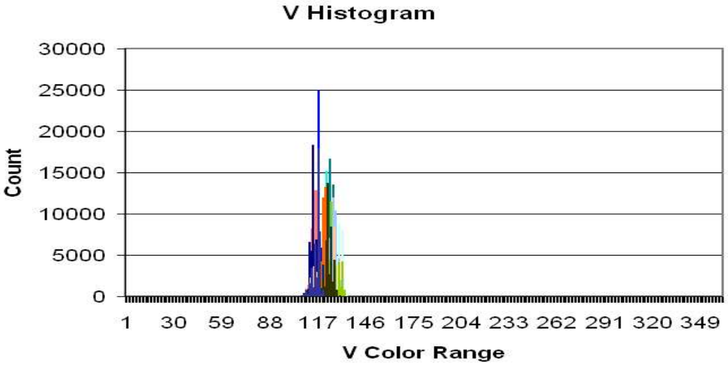


Figure 5.17: Histogram of V values under varying lighting conditions

5.2.4 The Classifier

Pattern classification is the fundamental building block of a technical system design to exhibit an application-specific intelligence. Knowledge about these distributions are acquired by learning from examples which is intrinsically statistical. Pattern classification is a task to establish a mapping from the feature space to the class, w, in order to recognize the class, w, when an observation, v, is known. This mapping considers the variability of the class-specific distributions in the feature-space.

Bayesian decision theory [40, 73] is a fundamental statistical approach to the problem of pattern classification. This approach is based on quantifying the trade-offs between various classification decisions by using probability theory and the costs that accompany such decisions. It is assumed that the decision problem is posed in probabilistic terms, and the relevant probability values are known.

For some a priori probability $P(w_1)$ when a face image is found and some a priori probability $P(w_2)$ for a non-face image is found, then

$$P(w_1) + P(w_2) = 1;\ for\ 2\ class\ classifciation, \tag{5.10}$$

$$P(w_1) + P(w_2) + ...P(w_i) = 1;\ for\ i\ class\ classifciation. \tag{5.11}$$

The rule in (5.12) is used when a decision is needed with limited available information.

$$Decide\ w_1\ if\ P(w_1) > P(w_2);\ otherwise\ decide\ w_2. \tag{5.12}$$

The joint probability density of finding a pattern that is in class w_j and has feature value x, $p(w_j, x) = P(w_j|x)p(x) = p(x|w_j)P(w_j)$. By rearranging the equation gives the Bayes formula, (5.14).

$$P(w_j|x) = \frac{p(x|w_j)P(w_j)}{p(x)}, \tag{5.13}$$

$$p(x) = \sum_{j=1}^{n} p(x|w_j)P(w_j), \quad for\ n\ class\ classification. \tag{5.14}$$

The probability of error is minimized by the following Bayes decision rule:

$$Decide\ w_1\ if\ P(w_1|x) > P(w_2|x);\ otherwise\ decide\ w_2, \tag{5.15}$$

$$Decide\ w_1\ if\ p(x|w_1)P(w_1) > p(x|w_2)P(w_2);\ otherwise\ decide\ w_2. \tag{5.16}$$

Figure 5.18 illustrates the result after the above labeling process. The black pixels are non-face pixels with the white or gray pixels the likely skin pixels.

5.2.5 "Color-Judges"

Two Color-judges are proposed and used by the pixel-labeling module in assigning labels to the pixels of the frames. There are two different possibilities for assigning the pixel-labeling procedure.

- Indirect use of a color-judge - This alternative utilizes the color-judge for constructing a look-up table at the initialization phase. Pixel-labeling runs through the color space, passing each color value to the color-judge, which responds with a pixel label. The labeling module writes this return value in the look-up table. During tracking, pixel-labeling runs through the current frame, reads the actual pixel label from the look-up table indicated by the current pixel value and assigns it to the current pixel.

- Direct use of a color-judge - In this approach, no look-up table is constructed.

> The pixel color values of the incoming frames are directly passed to the color-judge and the pixel label is assigned to the current pixel.

The return values from the color-judge depends on the segmentation methods: '0' for the background and '1' for the object pixels.

If the segmentation method is time-consuming or some heavy color space transformation, such as RGB to HSI color space, is required. Off-line and indirect use of color-judge is recommended. Direct use of the color-judge is justified when the segmentation needs to be adaptive in order to cope with varying lighting conditions. In such cases, the look-up table is reconstructed whenever an adaptation to a change in the conditions take place. Direct use of the color-judge allows adaptation without excessive processing caused by the look-up table reconstruction. In this work, a simple YUV color-judge which performs YUV color space thresholding is implemented. It takes four thresholds as inputs, including upper and lower thresholds for each UV channel. The output is either '0' or '1', depending on the pixel color being within or outside the specified UV thresholds.

5.3 Object Extraction Method

For tracking objects of distinctive and uniform colors, some simple segmentation methods are enough. From the test images and experiments, model-based color thresholding and face ratio heuristic rule are desired.

5.3.1 Face Detection Operations

The quality of the source image is enhanced by removing noise via Gaussian pyramid decomposition, a "blurring" filter. All the pixels in the image are analyzed using the planes (5.7 - 5.9) and the decision rules (5.15 - 5.16), which determine the presence or absence of a skin color. Those pixels that are classified under the skin color are set to white otherwise to black. The resultant binary image contains a few contours and spurious white pixels as shown in Figure 5.18.

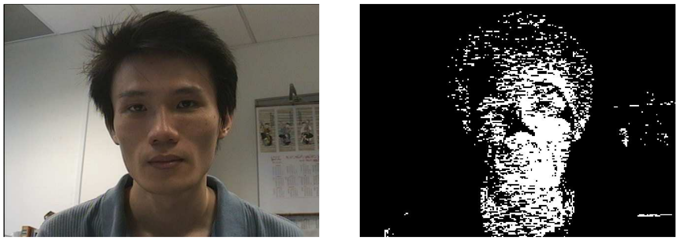
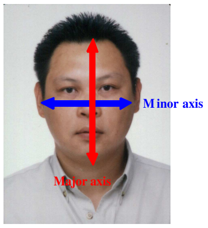


Figure 5.18: Probability Distribution Function of a face

To remove these noisy pixels and compact the contours, median filtering is used followed by the morphological operations of erosion and dilation.

5.3.2 The Dominant Feature

A face is assumed to be the dominant feature in an image. After applying median filtering and the morphological operations, the detected facial region is subjected to a simple heuristic rule. This is based on the geometric analysis of the human face. A skin color region belongs to a human face candidate if the ratio of the major axis to the minor axis is less than a threshold of 1.7 as stated in [74]. Fifty (50) training sets (Appendix C) have been collected and verified. The face ratio tends to lie within the range of 1.1 to 1.8. The application of this heuristic rule is the final part of the face detection module.



$$FaceRatio = \frac{Major\ \ Axis}{Minor\ \ Axis},$$

$$1.1 < Face\ Ratio < 1.8 \quad (5.17)$$

Figure 5.19: Face Ratio Relationship

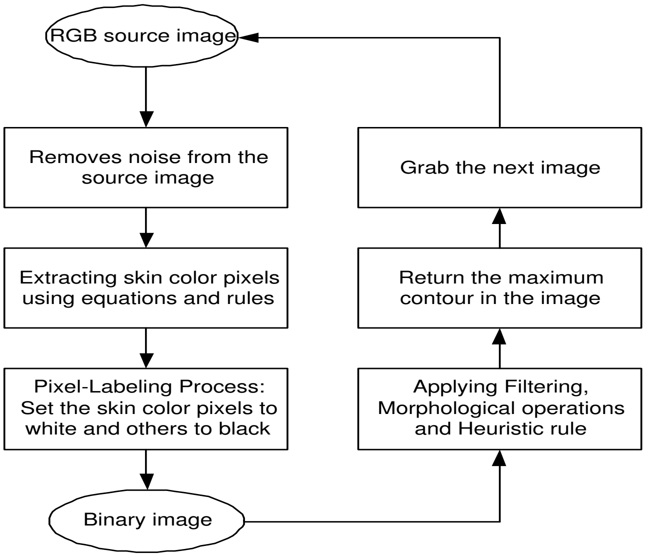

RGB source image
Removes noise from the source image
Extracting skin color pixels using equations and rules
Pixel-Labeling Process: Set the skin color pixels to white and others to black
Binary image
Applying Filtering, Morphological operations and Heuristic rule
Return the maximum contour in the image
Grab the next image

Figure 5.20: Flow-chart of the Face Detection module

The position of the contour is indicated on the original source image where the face is detected. The face detection process is represented in Figure 5.20 and is illustrated by the example in Figure 5.21. Figure 5.21b shows the image after median filtering and Figure 5.21c shows the image after performing the morphological operations.

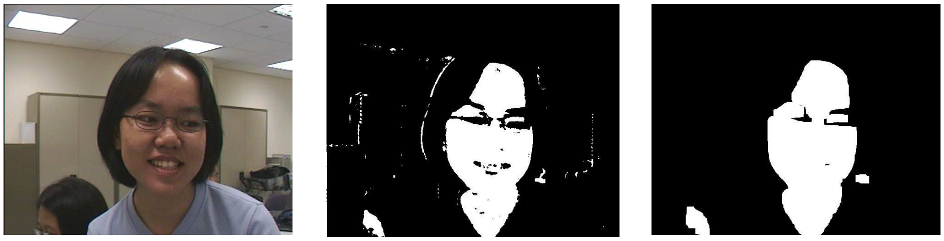


Figure 5.21: (a)source image (b)after median filtering (c) after morphological operations

5.3.3 Tracking Mechanism

A 2-dimensional tracking is performed in the image xy-coordinates to track multiple similar colored objects. Perceptual interfaces provide the computer the ability to sense. The perceptual user interface is able to track in real time without taking too much computational resources. The tracking mechanism in this work is developed based on CAMSHIFT (Continuously Adaptive Mean Shift) [30, 31, 32]. CAMSHIFT is based on a non-parametric technique (Section 4.3.3). It uses robust statistics which tends to ignore outliers in a given data. A face color probability diagram, *I(x,y)*, the zeroth moment and the first moment are computed by the formulae in (4.14), (4.15), (4.16) and (4.17).

CAMSHIFT uses the Mean Shift algorithm as its core. It includes the use of the previously computed zeroth moment and the first moment information to give a good initial search window size and location. The 2D orientation of the probability distribution is obtained by using the second moment during the operation.

$$M_{02} = \sum_x \sum_y y^2 I(x,y), \tag{5.18}$$

$$M_{20} = \sum_x \sum_y x^2 I(x,y). \tag{5.19}$$

The first two Eigenvalues of the probability distribution “blob” found by CAMSHIFT is calculated as follows:

$$a = \frac{M_{20}}{M_{00}} - x_c^2, \tag{5.20}$$

$$b = 2(\frac{M_{11}}{M_{00}} - x_c y_c), \tag{5.21}$$

$$c = \frac{M_{02}}{M_{00}} - y_c^2, \tag{5.22}$$

$$\theta = \frac{1}{2}\tan^{-1}(\frac{b}{a-c}). \tag{5.23}$$

Then length l and width w of the distribution centroid are

$$l = \sqrt{\frac{(a+c) + \sqrt{b^2 + (a-c)^2}}{2}}, \tag{5.24}$$

$$w = \sqrt{\frac{(a+c) - \sqrt{b^2 + (a-c)^2}}{2}}. \tag{5.25}$$

The width of the search window set to $s = 2(\sqrt{\frac{M_{00}}{256}})$ and the height to $1.4s$.

The Hue in the HSV (Hue, Saturation, Value) color space is used to model skin color in CAMSHIFT. This model describes the UV plane in polar coordinates using a vector of length, S and a rotational color angle, H. The luminance, V corresponds to the Y of the YUV space. These spaces are highly intuitive to human understanding as it defines color similar to human understanding of color. It is observed that humans have the same Hue, given a 1-dimensional color map for human skin color. A model of the desired hue is created using a color histogram. The hues derived from the skin-color pixels in the image are sampled from the H channel and channeled into a 1-dimensional histogram. CAMSHIFT uses this model to compute the probability that a pixel in the image is part of a face and it replaces each pixel in the video frame with the probability computed. A skin color probability distribution image is generated in Figure 5.22.

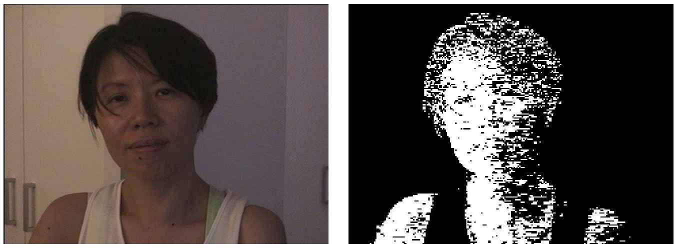


Figure 5.22: (a) A human subject (b)Its corresponding skin color probability diagram

However, Hue is ambiguously defined when the Saturation, S, or Value, V, are at either extremes, causing tracking to be inaccurate under very bright lighting or very dim lighting conditions as illustrated in Figure 5.23. When brightness is low V is near to 0, and when Saturation is low S is near to 0. Hue is noisy in a small hexcone and the small number of discrete Hue pixels are not able to adequately represent slight changes in RGB leading to hue wild swings. Integrating the tracking module and the face detection module helps to mitigate this problem since two different color model are used in the two modules.

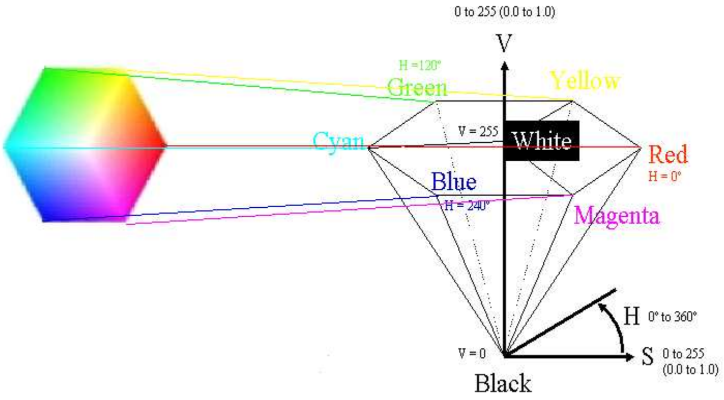


Figure 5.23: HSV hexcone

As shown in the system integration flowchart (Figure 5.24), the face detection module keeps the face in the field of view of the camera. When a face is detected and centered, the tracking algorithm is activated. If the pixel values in the region satisfy equations (5.1), (5.2) and the decision rules, the tracking continues. Otherwise tracking stops and the face detection module continues to search for a face.

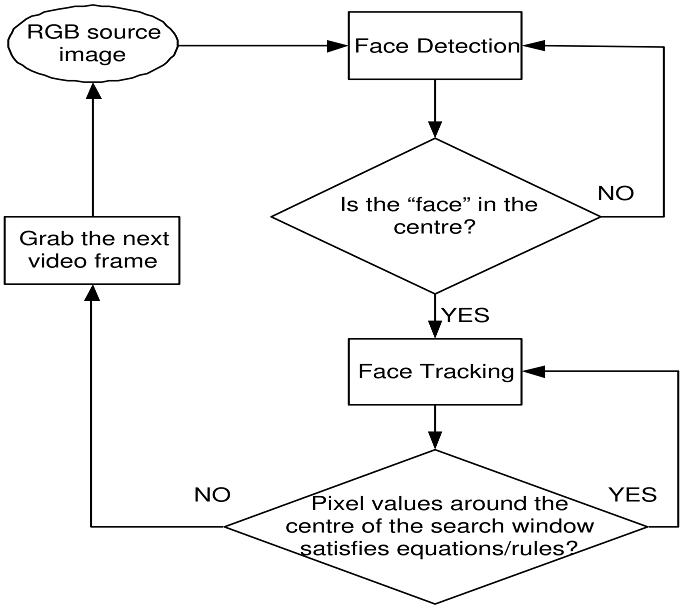

RGB source image
Face Detection
Is the "face" in the centre?
NO
YES
Grab the next video frame
Face Tracking
NO
YES
Pixel values around the centre of the search window satisfies equations/rules?

Figure 5.24: Flow-chart of the system integrating face dection and tracking modules

A model of the desired Hue is created using a color histogram. The Hues derived from the skin color pixels in the image are then sampled from the H channel and channelled into a 1-dimensional histogram. It is found that simple skin color histograms work well with a wide variety of people without having them updated. The tracking algorithm uses this model to compute the probability that a pixel in a frame is part of a face and replaces each pixel with the skin probability. The white pixels indicate the highest probability of a skin color pixel.

The face detection module searches for a face in the video scene prior to running of the tracking algorithm and after the tracking module loses track of the person.

The detection stage is a coarser measurement of the face location in the video scene ROI, Region of Interest, while the tracking module is on the refinement stage.

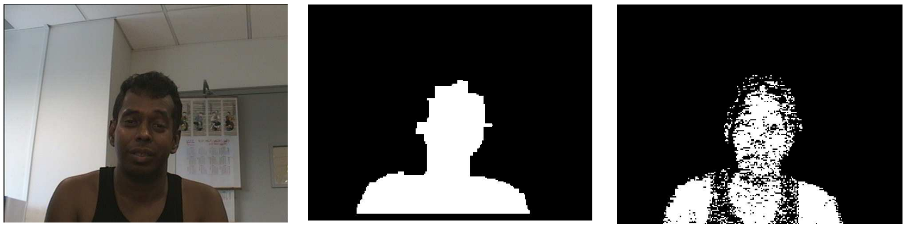


Figure 5.25: A Sri Lankan human subject, binarized ROI and the skin color probability diagram

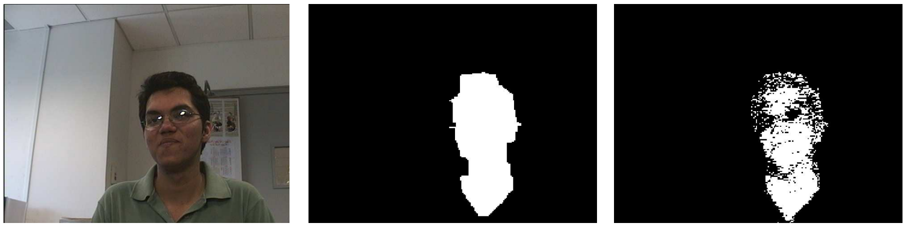


Figure 5.26: An Indian human subject, binarized ROI and the skin color probability diagram

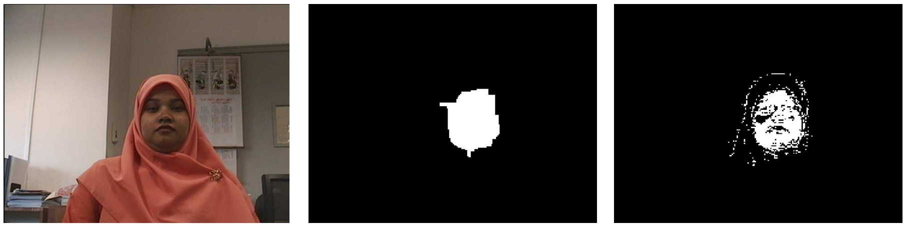


Figure 5.27: A Bangladeshi human subject, binarized ROI and the skin color probability diagram

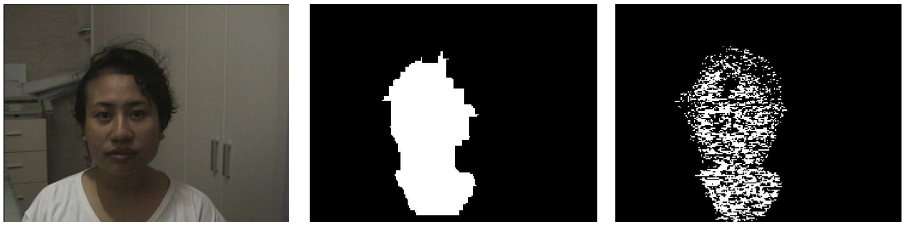


Figure 5.28: An Indonesian human subject, binarized ROI and the skin color probability diagram

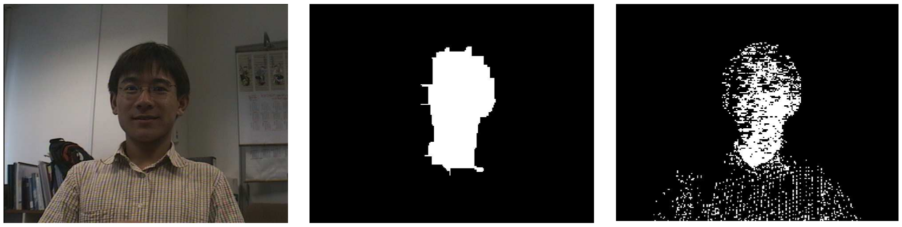


Figure 5.29: A Chinese human subject, binarized ROI and the skin color probability diagram

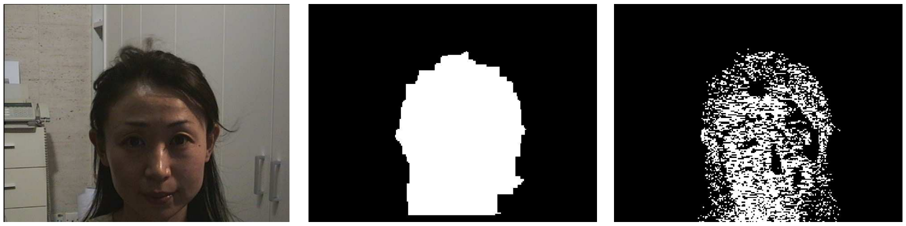


Figure 5.30: A Japanese human subject, binarized ROI and the skin color probability diagram

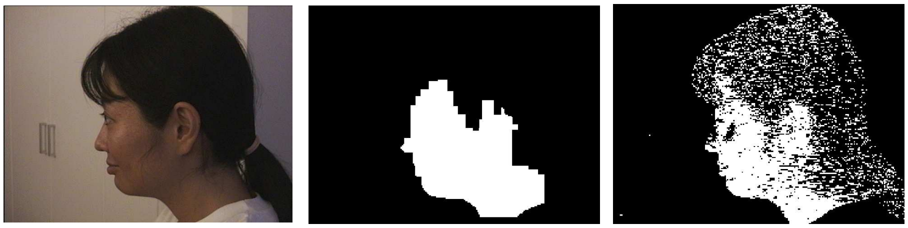


Figure 5.31: A Thai human subject, binarized ROI and the skin color probability diagram

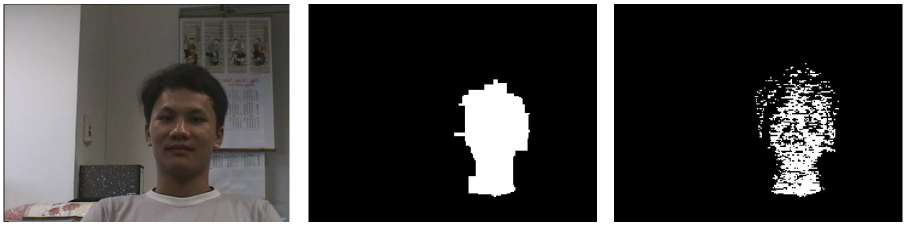


Figure 5.32: A Vietnamese human subject, binarized ROI and the skin color probability diagram

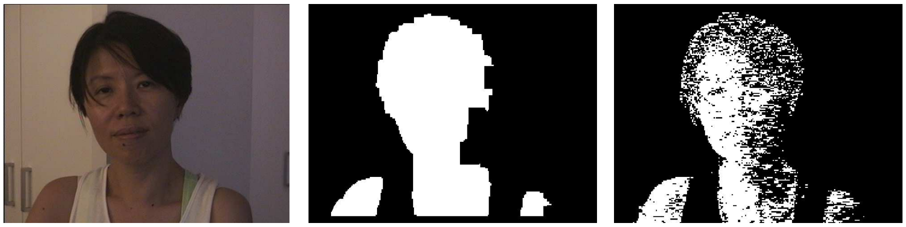


Figure 5.33: An Eurasian human subject, binarized ROI and the skin color probability diagram

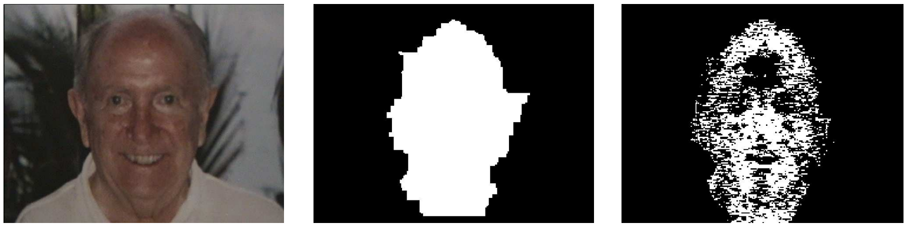


Figure 5.34: A Spanish human subject, binarized ROI and the skin color probability diagram

Chapter 6

Integration and Implementation

This chapter defines the system and various controllers' architecture. In addition, the camera mounting height and the viewing angle are determined.

A marker maintains information about an object or several similar objects in the environment. In this work, markers act as an interface between the image processing system and the control system. Dynamic handling of markers is a methodology to cope with the ever changing environment. These markers are updated whenever an object enters or leaves the robot's field of view.

6.1 General System Architecture

The proposed system is implemented on the Magellan Pro mobile robot (Figure 6.1). This robot is equipped with 16 ultrasonic and tactile sensors with 360^o coverage. It has an onboard Pentium II computer with Red Hat 6.2 Linux Operating System and operates on a 24V re-chargeable battery supply. A wireless Ethernet port is used to control the robot from a remote computer terminal as required. A Sony EVI-D30 pan-tilt camera is installed onto the robot for visual sensing and is controlled via the serial RS232 communication port. The camera pans at a restricted fontal angle of not wider than 149^o and tilts to a restricted angle of not more than 78^o. In order to track a human subject with a height of 1.7m, metal bars and supporting plates are utilized to increase the robot height from 0.3m to 1.3m (Figure 6.1). The height and angle calculation are illustrated in (Figure 6.3) and (Figure 6.4).

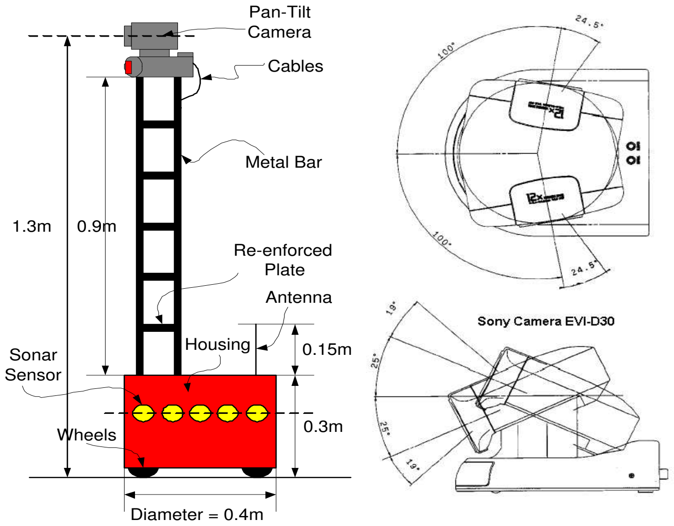


Figure 6.1: The Magellan Pro Mobile Robot

6.2 Vision Modules Overview

Figure 6.2 illustrates the data and process flow diagram. The vision system reads the raw images from the camera module; and an image processing filter removes the noise. The first step in image processing is color-based pixel labeling, which is done by the pixel-labeling module (in the detection module). It reads the frames and assigns each pixel a label. A probabilistic segmentation method produces a gray-scale probability image which needs to be binarized.

The next step is object labeling and it performs a connected component labeling

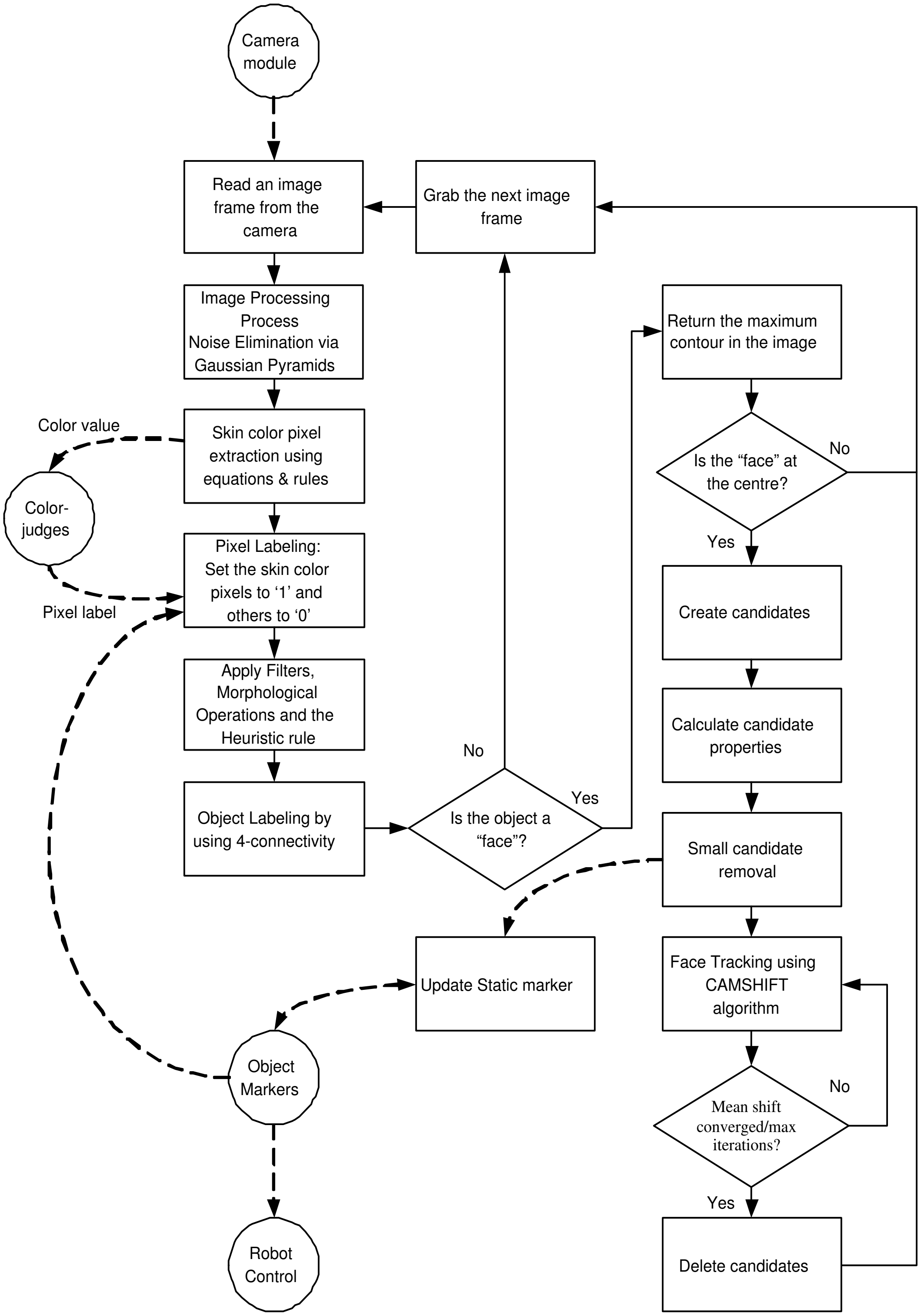

Camera module
Read an image frame from the camera
Grab the next image frame
Image Processing Process Noise Elimination via Gaussian Pyramids
Return the maximum contour in the image
Color value
Skin color pixel extraction using equations & rules
Is the "face" at the centre?
No
Color-judges
Yes
Pixel Labeling: Set the skin color pixels to '1' and others to '0'
Create candidates
Pixel label
Apply Filters, Morphological Operations and the Heuristic rule
Calculate candidate properties
No
Yes
Object Labeling by using 4-connectivity
Is the object a "face"?
Small candidate removal
Update Static marker
Face Tracking using CAMSHIFT algorithm
Object Markers
No
Mean shift converged/max iterations?
Yes
Robot Control
Delete candidates

Figure 6.2: Data and Process Flow Diagram

procedure. Labeling is based on 4-connectivity, and performed within the two runs through an image.

The tracking module being the last component, tracks color blobs and updates object markers. It reads the labeled frame and creates an object candidate from each connected component. These candidates calculate some essential image properties, such as position (centroid), face ratio, area and bounding box. It eliminates noise by removing the object candidates that fail to fulfil the criteria mentioned.

The estimation procedure requires the tracker to utilize CAMSHIFT algorithm. The three essential aspects of the proposed architecture are; the use of color-judges in the pixel-labeling phase; the use of an object markers as an interface between image processing and control; and the dynamic handling of the object trackers.

6.2.1 Markers and Marker Interface

An object candidate is created from each connected component found in the labeled image. Some basic properties are calculated for each candidate, the centroid, the face ratio, the area and the bounding box. Thereafter, each candidate's properties are compared to the respective thresholds to remove the small candidates. If any of the properties of a candidate falls below the corresponding threshold, that candidate is considered as small and removed.

The object markers are updated by the tracking module when small candidates are removed. A marker tracks automatically an object in the robot's environment once it has been initialized with the properties of an object. In addition, it is an interface for receiving tracking task and for returning information pertaining to the tracked object.

For a robot control system, it is beneficial to have a standardized interface for the visual tracking system. The marker interface has 2 essential functions. Which is to pass the properties of the tracked object to a marker. The markers are read to obtain the most recent data about the tracked objects.

The marker interface allows several concurrent tracking tasks by switching between the tasks as per the prevailing situation. The number of concurrent tracking tasks, that is, the number of markers, depends on the capabilities of the image-processing system.

The tracking is dynamic, meaning that the tracking system automatically starts tracking, when an object with the properties specified by the control system enters the robot's field of view. In a similar manner, the system automatically ceases tracking when an object leaves the field of view.

There are two types of markers. Static markers for tracking single objects which are implemented in this work. Dynamic markers for tracking multiple objects of the same are omitted due to the limited processing capability of the processor onboard the robot.

6.2.2 Camera Module

The camera module provides a standard interface for capturing images. It allows the frame grabber to write frames to the memory via DMA (Direct Memory Access).

Asynchronous continuous capture mode is used in this work. The frame-grabber board read image frames from the camera at a maximum rate of 25 Hz. In other words, the same frame buffer is being written and is being read at the same time. It is possible that the same frame is being processed twice if the tracking software works at a faster frame rate than the camera. Whereas, a frame is occasionally skipped when the image processing takes in frames at a lower rate as compared to the camera. Such inaccuracy is however unfound and therefore the system reliability is not affected.

Another alternative for image capturing is by using the single-capture mode, but it is too slow for real-time performance.

6.3 Camera Height and Camera Viewing Angle Calculation

The height of the robot and the viewing angle of the camera are designed as in Figures 6.3 and 6.4.

It is assumed that the human subject is at most 1.7m or at least 0.9m tall. When a human subject is at a distance of 0.5m in front of the robot, the camera tilts to keep the face of the human subject within the camera view. If the human subject is out of the camera's viewing angle, the robot readjusts it's position and maintains a larger distance between the human subject and itself. Similarly, the camera pans to keep the face of the human subject within the camera view at a frontal angle of 149^o. When the human subject moves outside the camera's frontal angle, the robot rotates itself to a new angular position to meet the above

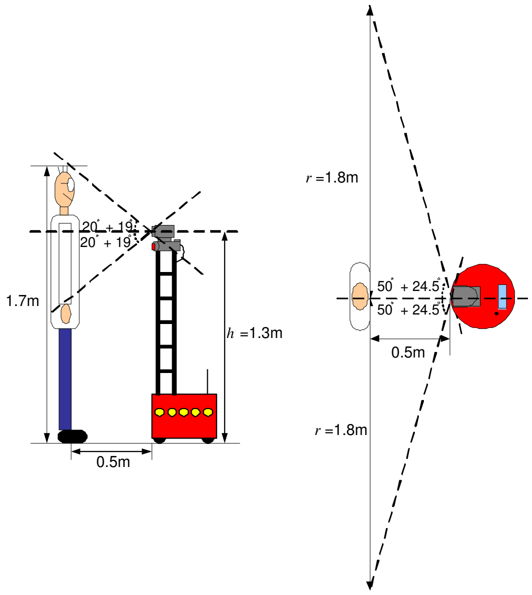


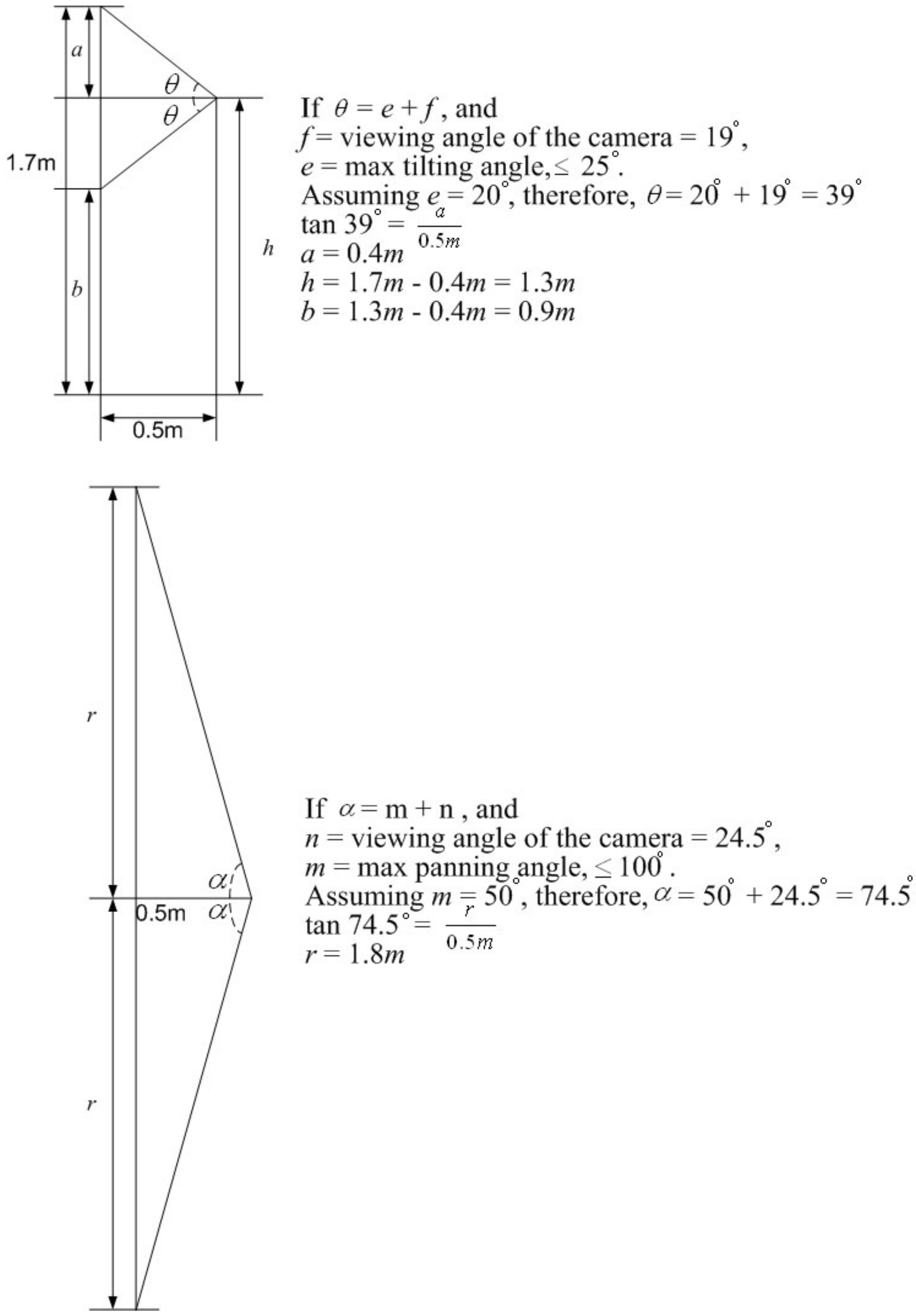
Figure 6.3: Robot Height and Viewing Angle

considerations.

A local map is used to record the sensory information provided by the 16 sonar sensors with respect to the mobile robot. This local map consists of 32 sectors and each sector covers an angle of 11.25^o. This local map provides the mobile robot with the information of the obstacle locations and distances from it's current location.

The following constraints are imposed in the system:

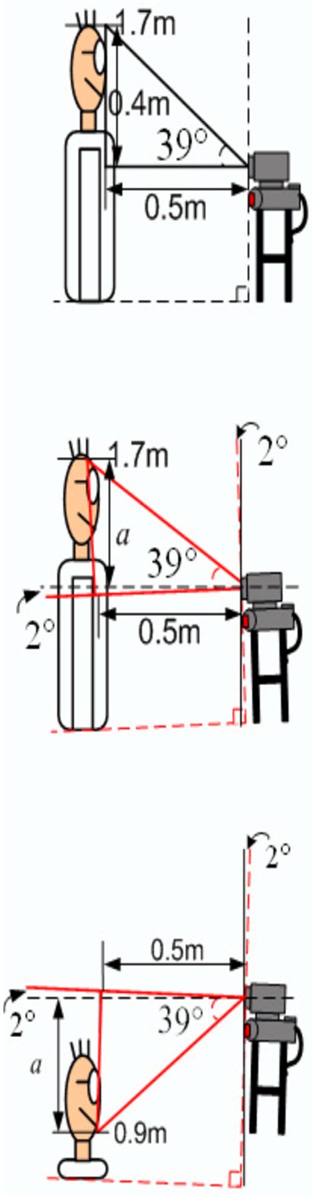
- When the camera is panning, the forward movements and reverse movements of the robot are suspended due to the design of the controller module.
- A minimum constrained distance of 0.5m is maintained between the human subject and the robot. To maintain this constrained distance, the robot back-off when the reading is less than 0.5m or when the rear clearance distance is greater than 1m. Otherwise, the robot stops all movements.



If $\theta = e + f$, and
f = viewing angle of the camera = 19°,
e = max tilting angle, $\leq 25^\circ$.
Assuming $e = 20^\circ$, therefore, $\theta = 20^\circ + 19^\circ = 39^\circ$
$\tan 39^\circ = \frac{a}{0.5m}$
$a = 0.4m$
$h = 1.7m - 0.4m = 1.3m$
$b = 1.3m - 0.4m = 0.9m$

If $\alpha = \mathrm{m} + \mathrm{n}$, and
n = viewing angle of the camera = 24.5°,
m = max panning angle, $\leq 100^\circ$.
Assuming $m = 50^\circ$, therefore, $\alpha = 50^\circ + 24.5^\circ = 74.5^\circ$
$\tan 74.5^\circ = \frac{r}{0.5m}$
$r = 1.8m$

Figure 6.4: Robot Height and Viewing Angle calculation



In normal case, when the system is stationary.

For face ratio = 0.2/ 0.16 = 1.25, which is within the range of 1.1 to 1.8. The heuristic rule is satisfied.

For face ratio = 0.2/0.12 = 1.67, which is within the range of 1.1 to 1.8. The heuristic rule is satisfied.

At +2.0°, the camera tilts down 2.0°, therefore the forehead of the human subject is not fully covered by the camera viewing angle.

Using trigonometry, with an angle of 37°, tan 37° = a/0.5,
and *a* = 0.5 tan 37° = 0.38*m*.
Therefore, 0.4*m* - 0.38*m* = 0.02*m* not covered.
Face ratio = (0.2-0.02)/0.16 = 1.125 and
Face ratio = (0.2-0.02)/0.12 = 1.5,
are within the range of 1.1 to 1.8, therefore both satisfy the heuristic rule.

At -2.0°, the camera tilts up 2.0°, therefore the lower jaw of the human subject is not fully covered by the camera viewing angle.

Using trigonometry, with an angle of 37°, tan 37° = a/0.5,
and *a* = 0.5 tan 37° = 0.38*m*.
Therefore, 0.4*m* - 0.38*m* = 0.02*m* not covered.
Face ratio = (0.2-0.02)/0.16 = 1.125 and
Face ratio = (0.2-0.02)/0.12 = 1.5,
are within the range of 1.1 to 1.8, therefore both satisfy the heuristic rule.

Figure 6.5: An oscillation of $+/-2.0^o$

- To ensure the stability of the robot and the mounted camera, it's speed is capped to 0.5m/sec.

The camera "swings" at an angle of $\pm 1.5^o$ while traveling or when coming to a stop. Figure 6.5 illustrates otherwise the camera oscillation angle of $\pm 2.0^o$, which is manageable.

The two worst cases face ratios, 0.2/0.16 and 0.2/0.12 (Figure 6.5) are considered for testing purposes. The height of a human subject is constrained at a minimum height of 0.9m and a maximum of 1.7m, to keep the face within the camera's maximum viewing angle (Figures 6.3 and Figure 6.4).

The block diagram of the Robot and Camera control are shown in Figure 6.6 and 6.7.

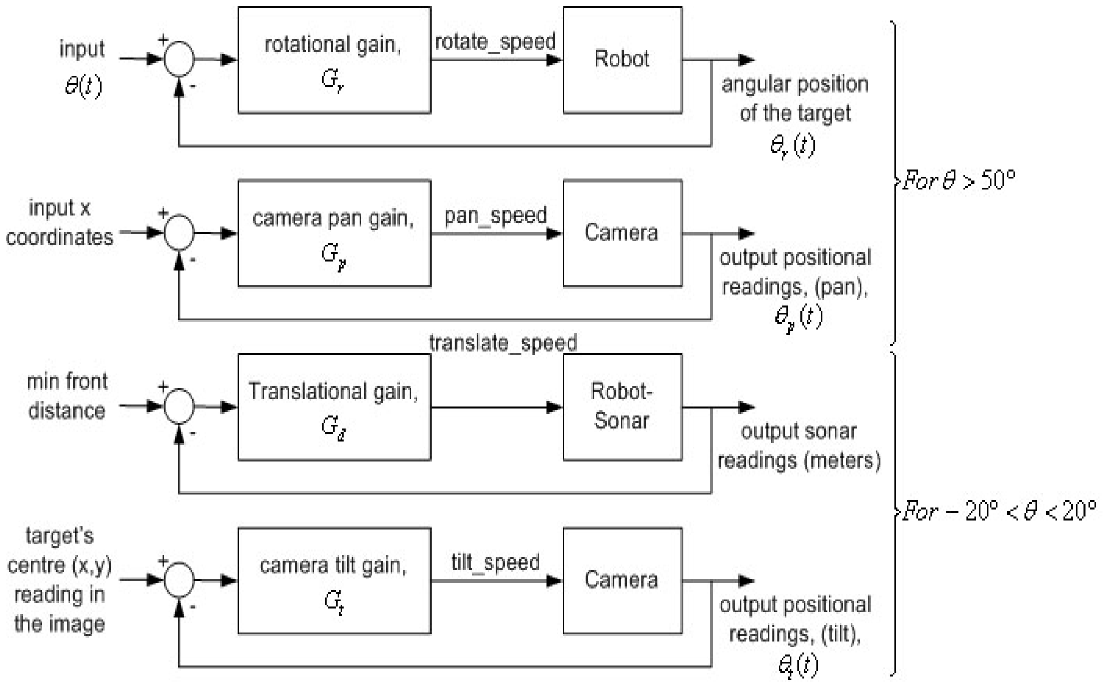

input $\theta(t)$
+
-
rotational gain, G_r
rotate_speed
Robot
angular position of the target $\theta_r(t)$
$For\ \theta > 50°$
input x coordinates
camera pan gain, G_p
pan_speed
Camera
output positional readings, (pan), $\theta_p(t)$
min front distance
Translational gain, G_d
translate_speed
Robot-Sonar
output sonar readings (meters)
$For\ -20° < \theta < 20°$
target's centre (x,y) reading in the image
camera tilt gain, G_t
tilt_speed
Camera
output positional readings, (tilt), $\theta_t(t)$

Figure 6.6: Block diagram of the Robot and Camera control

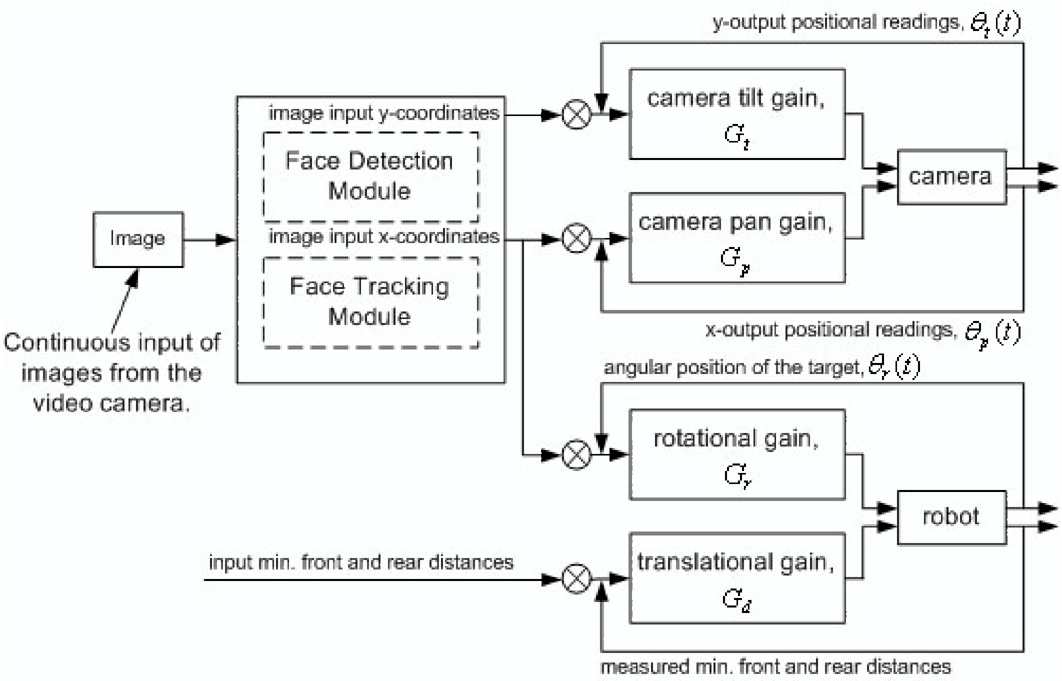

y-output positional readings, $\theta_t(t)$
image input y-coordinates
Face Detection Module
image input x-coordinates
Face Tracking Module
Image
Continuous input of images from the video camera.
camera tilt gain, G_t
camera pan gain, G_p
camera
x-output positional readings, $\theta_p(t)$
angular position of the target, $\theta_r(t)$
rotational gain, G_r
robot
input min. front and rear distances
translational gain, G_d
measured min. front and rear distances

Figure 6.7: Block diagram of the Robot and Camera control

Figure 6.8 demonstrates the software setup of the robot. At the top of the robot's environment is the Naming service, which allows programs to access the robot's hardware; such as the drive system, and the sensor systems, like tactile and sonar.

Ptzserver and V4lserver are software modules provided in the MOBILITY Robot Development Software [65]. The coding developed in this work is linked to these servers to access the robot's hardware, mainly, to pan and tilt the camera, drive the robot and read the sonar sensors.

Ptzserver is a software interface between the serial communication port and the pan-tilt camera. This server performs functions such as panning the camera, tilting the camera and camera's len zooming, etc.

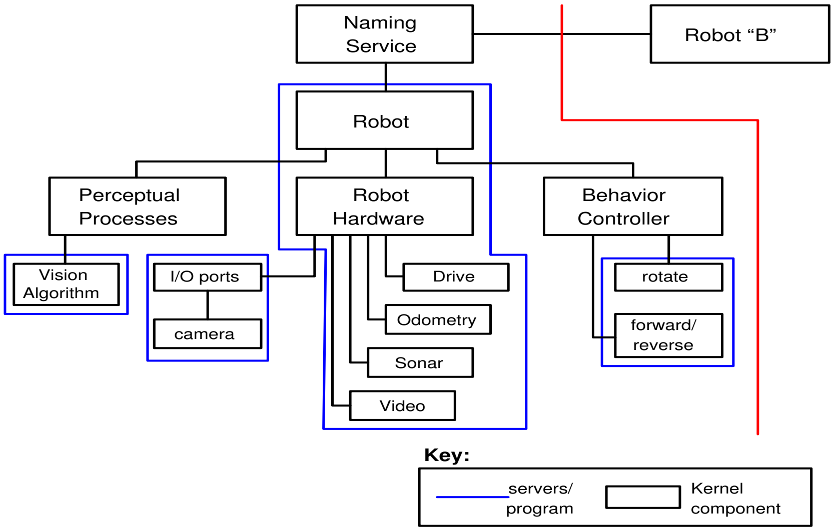
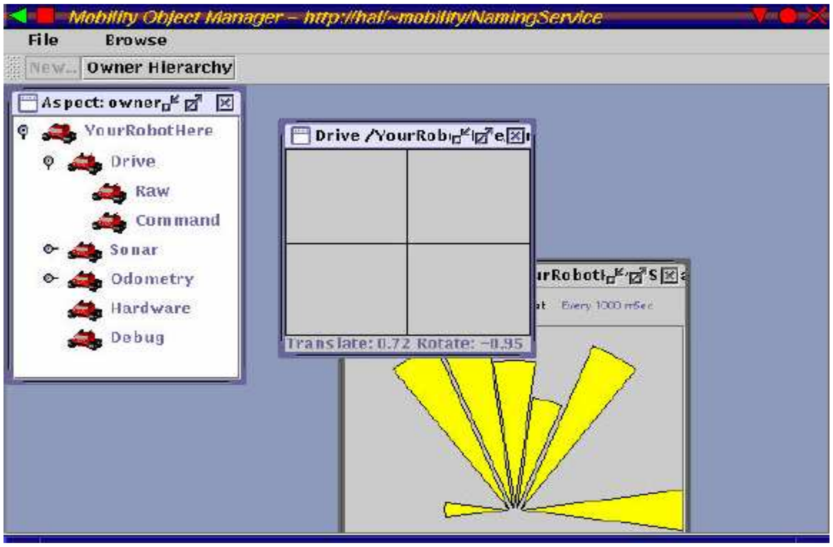


Figure 6.8: Robot software setup



Translate: 0.72 Rotate: -0.95

Figure 6.9: The GUI of the drive/sonar range view

The main tasks of the V4lserver is to read the processed data from the frame-grabber board or write raw video frames into the buffer of the frame-grabber board. V4lserver uses the functions provided by the Ptzserver to access the camera module. Code is appended into this server to perform functions including RGB color space to UV color space conversion, pixel labeling, skin color detection, face tracking and camera/robot control access.

Figure 6.10 illustrates the face tracking sequence and Figure 6.11 shows the robot human following sequence.

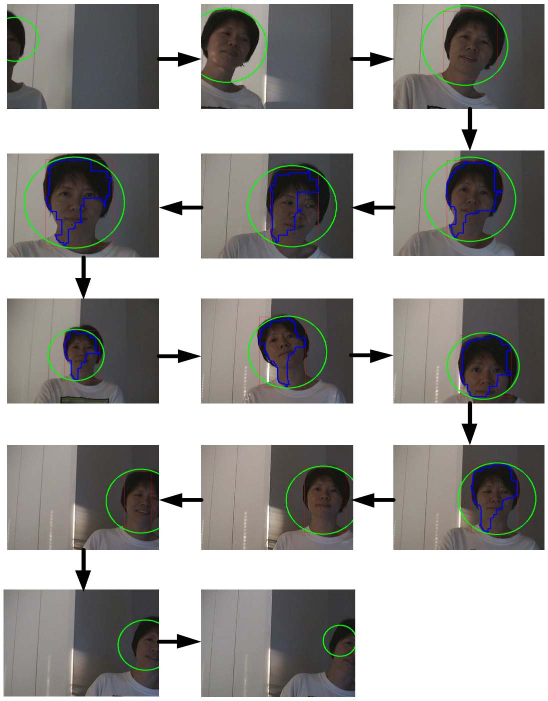


Figure 6.10: Illustration of a face tracking sequence

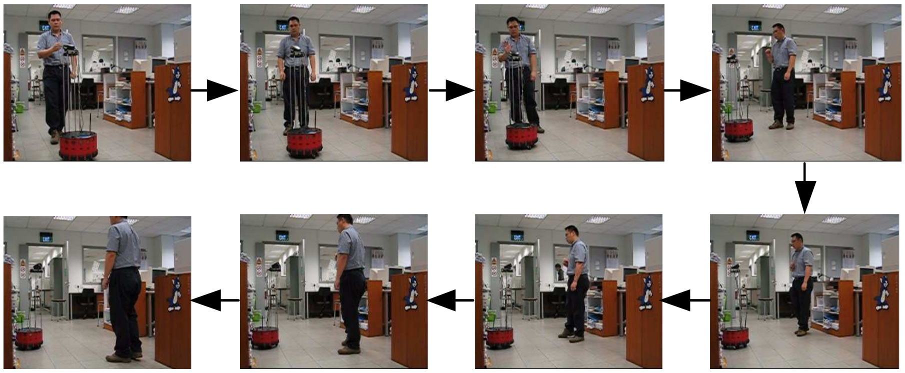


Figure 6.11: Illustration of a robot human following sequence

Chapter 7

System Performance Analysis

7.1 Tracking Experiments

Test runs are conducted with each tracking run of 40sec in duration. Two Hundred and forty (240) sample points are taken for one tracking run to check the tracking accuracy of the integrated system. The tracking run uses the integrated system in the UV color space to compare with the tracking run in the YC_rC_b color space face tracking work. Results are compared based on an object-distance from the origin of the image in pixels. By examining each image frame and identifying the object-center, which is the nose area, an object-distance is determined.

The integrated system (red line, Figure 7.8) follows the image center closely for the horizontal movement. The x and y coordinates of the image center are measured in terms of pixels. The tracking runs are conducted with the same environmental conditions, to ensure that the observations are not due to changes in lighting or color of the tracked object.

Figure 7.3 shows the movement of the human subject with respect to the robot when the human subject is within the camera's field of view. Figure 7.4 shows

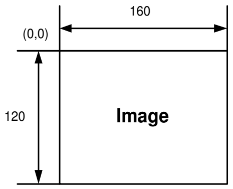


Figure 7.1: Dimensions of a typical frame in each tracking run

Figure 7.2: Image center - Nose

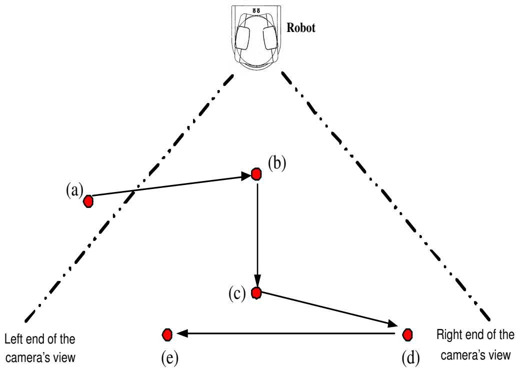


Figure 7.3: Movement of the human subject with respect to the robot during camera's controller test

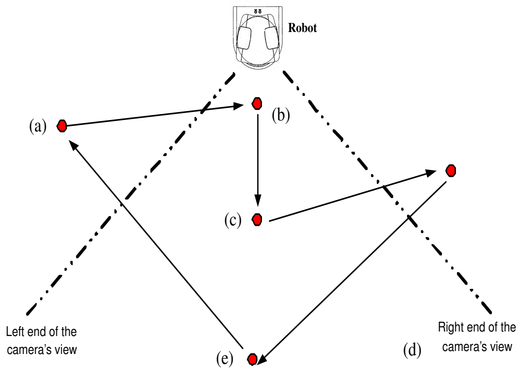


Figure 7.4: Movement of the human subject with respect to the robot during robot's controller test

the movement of the human subject with respect to the robot while testing the robot's controller following capability. For both cases, the human subject is facing the view of the camera.

7.2 YC_rC_b Color Space Face Detection and Tracking Work

The YC_rC_b color space for real-time face detection and tracking with an autonomous mobile robot is implemented in [75]. Linear color thresholding approach (7.1 and 7.2) is used to classify the skin and non-skin pixels shown in Figure 7.5.

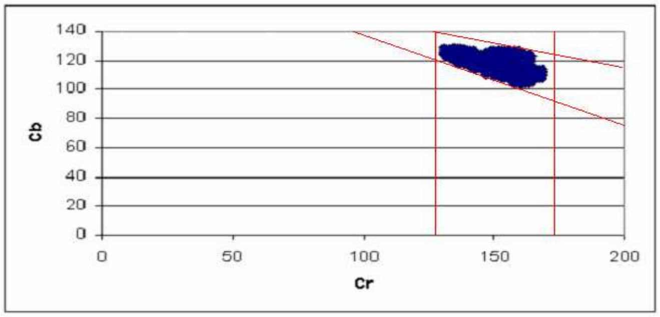


Figure 7.5: Cluster in the Chromatic color space

$$138 < C_r < 178, \tag{7.1}$$

$$200 < C_b + 0.6C_r < 215. \tag{7.2}$$

The tracking results from [75] is shown in Table 7.1 with a highest success rate of 97.5%.

Table 7.1: Tracking Results from Ling's YC_rC_b tracking work

Tracking Run	No. of Frames Target is lost	No. of Frames Target is tracked	No. of Frames Tracking is "off"	Success rate (%)
R1	16	213	11	88.8
R2	113	119	8	49.6
R3	0	224	16	93.3
R4	0	234	6	97.5
R5	25	204	11	85.0
R6	11	218	11	90.8

Section 7.2.1 records the YC_rC_b color space face tracking performance using the tracking algorithm and equations (7.1 and 7.2). The results will be use to compare with the YUV color space face tracking.

7.2.1 Performance - YC_rC_b Color Space Face Tracking

The tracking algorithm from the YC_rC_b color space tracking work yields tolerable tracking results and tolerable accuracy (Figure 7.6). When the camera starts to tilts upwards to "face" the laboratory ceiling lighting, it causes significant color shifts in subsequent images as the scene captured by the camera is suddenly illuminated and lead to poorer vertical tracking result (Figure 7.7).

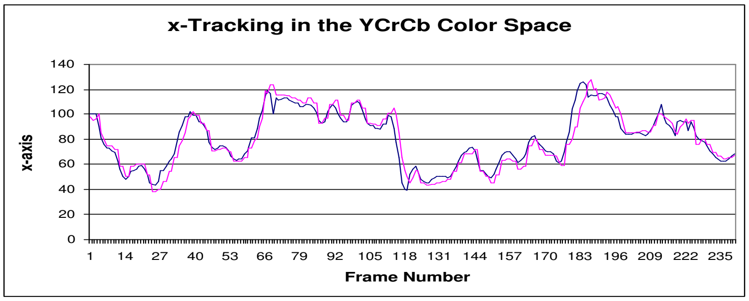


Figure 7.6: X tracking accuracy (YC_rC_b)

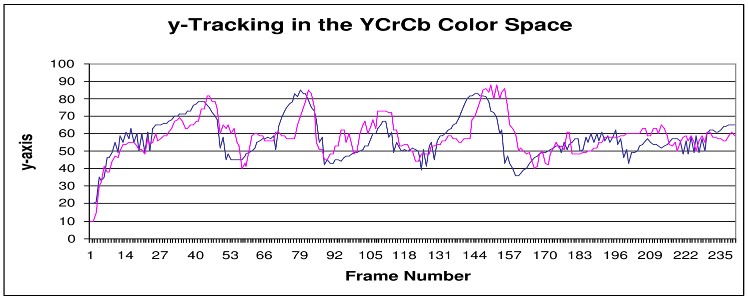


Figure 7.7: Y tracking accuracy (YC_rC_b)

7.3 UV Color Space Face Detection and Tracking Work

Figure 7.8 shows that the horizontal tracking is near indistinguishable from the actual distance. The vertical tracking (Figure 7.9) is better as compare to the YC_rC_b

color space tracking work. The erroneous tracking has significantly improved for over all the tracking run.

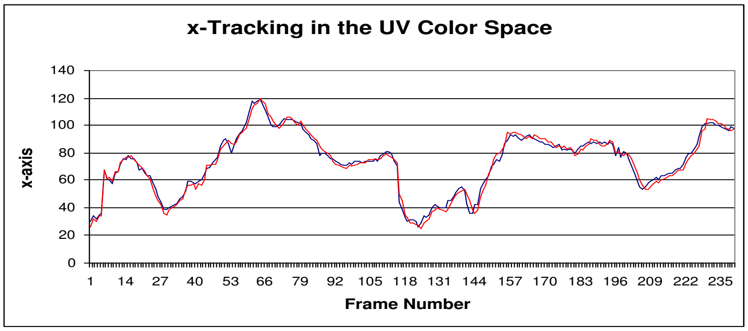


Figure 7.8: X tracking accuracy (UV)

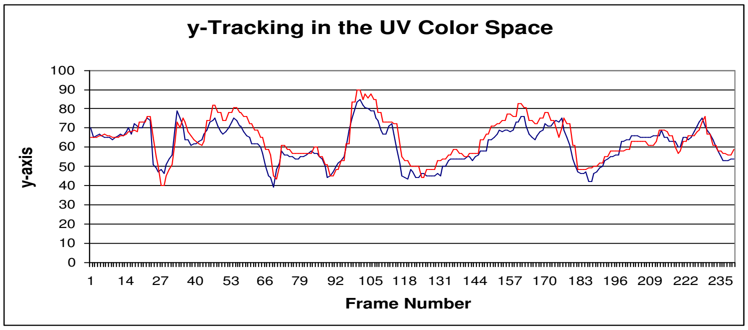


Figure 7.9: Y tracking accuracy (UV)

The vertical aspect of tracking is observed to fluctuate due to the color-shifts in the image when the camera tilts upwards to “face” the laboratory ceiling lighting. The scene captured by the camera is suddenly illuminated due to strong lightings leading to lower tracking accuracy.

7.3.1 Tracking Success Rate of the System

The tracking runs are conducted with the same environmental conditions and with different human subjects.

Table 7.2: Tracking Experiments

Tracking Run	No. of Frames Target is lost	No. of Frames Target is tracked	No. of Frames Tracking is "off"	Success rate (%)
R1	6	228	6	95.0
R2	0	237	3	98.8
R3	0	232	8	96.7
R4	2	225	13	93.8
R5	3	231	6	96.2
R6	9	223	8	92.9
R7	15	212	13	88.3
R8	8	222	10	92.5
R9	0	235	5	97.9
R10	70	130	40	54.2

$$Success\ Rate = \frac{(Total\ Frame) - (Lost\ Frame) - (OFF\ Frame)}{Total\ Frame}, \tag{7.3}$$

$$Average\ Success\ Rate = \frac{(Total\ Frame) - (Lost\ Frame) - (OFF\ Frame)}{(Total\ Frame) \times (TotalRun)}, \tag{7.4}$$

where *Total Frame* is the total number of image frames in a tracking run, *Lost Frame* is the number of frames during which the tracking module is unable to track the human subject and *OFF Frame* is the number of frames during which the tracking module turn off and activate the detection module.

The highest success rate is 98.8% in the UV color space tracking experiment. An average success rate is 90.6% as compared to 84.2% in the YC_rC_b color space tracking.

Generally, the system works well with different human subjects with the exception of R10 when the human subject stands up and the camera tilts upward to “face” the laboratory ceiling lights. Another exception noted is when the subject’s face is partially covered, making the face ratio smaller and does not satisfy the heuristic rule, (5.17). The best case for tracking occurs when the target’s face is not covered; as in the runs R2 and R9.

Figure 7.10: Human subjects

Chapter 8

Conclusions and Recommendations

The inclusion of other more stringent criteria for face classification should be considered to improve the robustness of the face extraction algorithm and to reduce the number of misclassification. A morphological-based processing module can be implemented to segment the eye-analogue as stated in [76]. The eye-analogue segment is useful to aid the face detection. It is most logical to extend the system by the inclusion of the segmentation of the eye locations, especially with the considerable success rates reported in [76].

The facial skin color model of the present system could be made adaptive by being able to adapt itself automatically to suit new environmental factors such as changes in ambient light and different cameras. Instead of having to derive the model manually to suit different environmental conditions, the model can have "in-built" tolerances for the variations of these parameters.

The tracking process can be improved by implementing a forward motion predictive method such as Kalman filtering. The recursive estimator in the Kalman filter can drive the proportional controllers to ensure more robust and precise tracking.

The current work ventured into the possibilities of using the UV color space to model the human skin color. Experiments and design requirements such as applications, effectiveness and robustness have been seriously considered in order to operate the mobile robot in a dynamic environment.

This work has verified that human skin color has a limited spectrum in the YUV color space and a two-dimensional UV plane is good enough for most applications. However, if there are limited objects in the scene and their color are distinguishable, the Hue value is enough. RGB color model is not recommended as it does not separate luminance component from the chrominance components. In addition, the camera's working parameters and color constancy greatly affect segmentation, therefore, care is required. Lastly, filtering and morphological operations during image processing phase helps to enhance and improve object detection performance.

Much effort has been put into deriving the model and implementing the tracking system on the mobile robot. Sonar scan and tactile scan information are integrated with the vision-based tracking system to allow tracking beyond the camera's field of view. In addition, the mobile robot is able to follow the tracked human subject while maintaining a suitable distance, rotating to face the person when the face moves out of the camera's field of view. Although the system is not able to perform face recognition task, the system developed has the merits of having sufficiently robust tracking and real-time performance. Furthermore, it is both orientation invariant under varying lighting conditions and under natural transformation such as translation, rotation and scaling.

Appendix A

Human subjects under three different lighting conditions

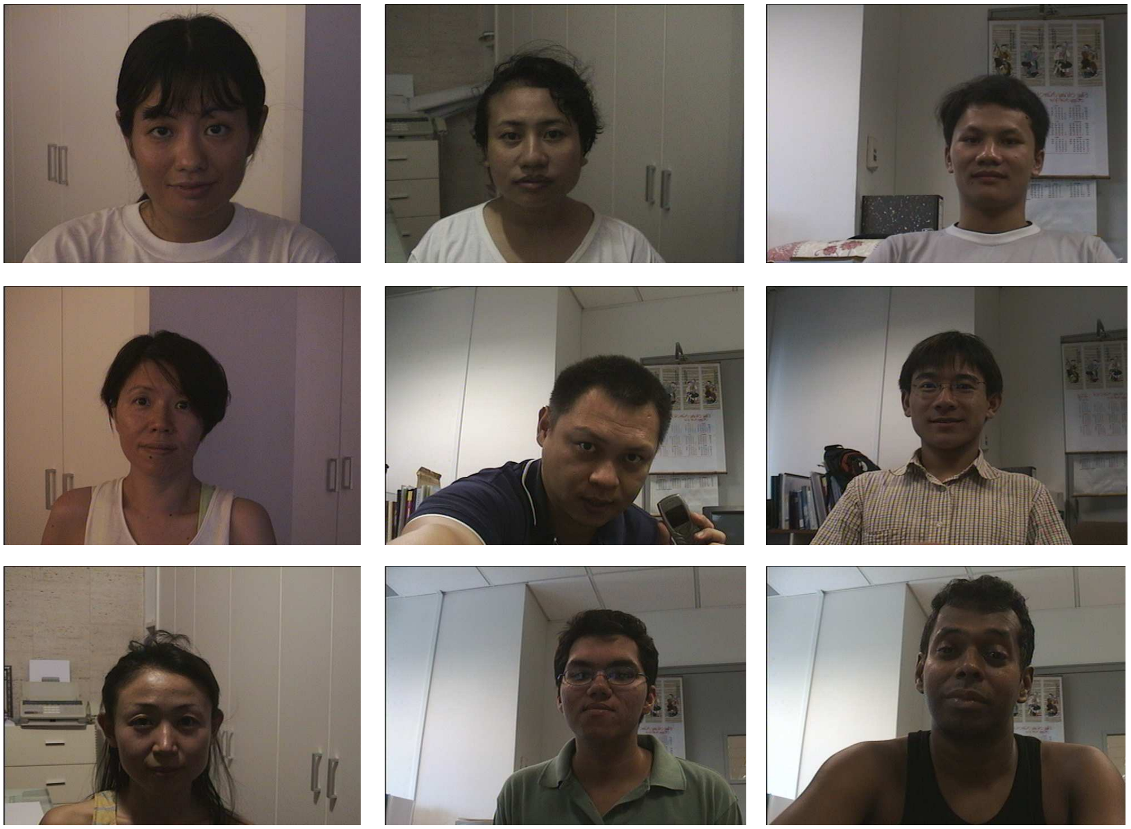


Figure A.1: Human subjects under Natural lighting conditions

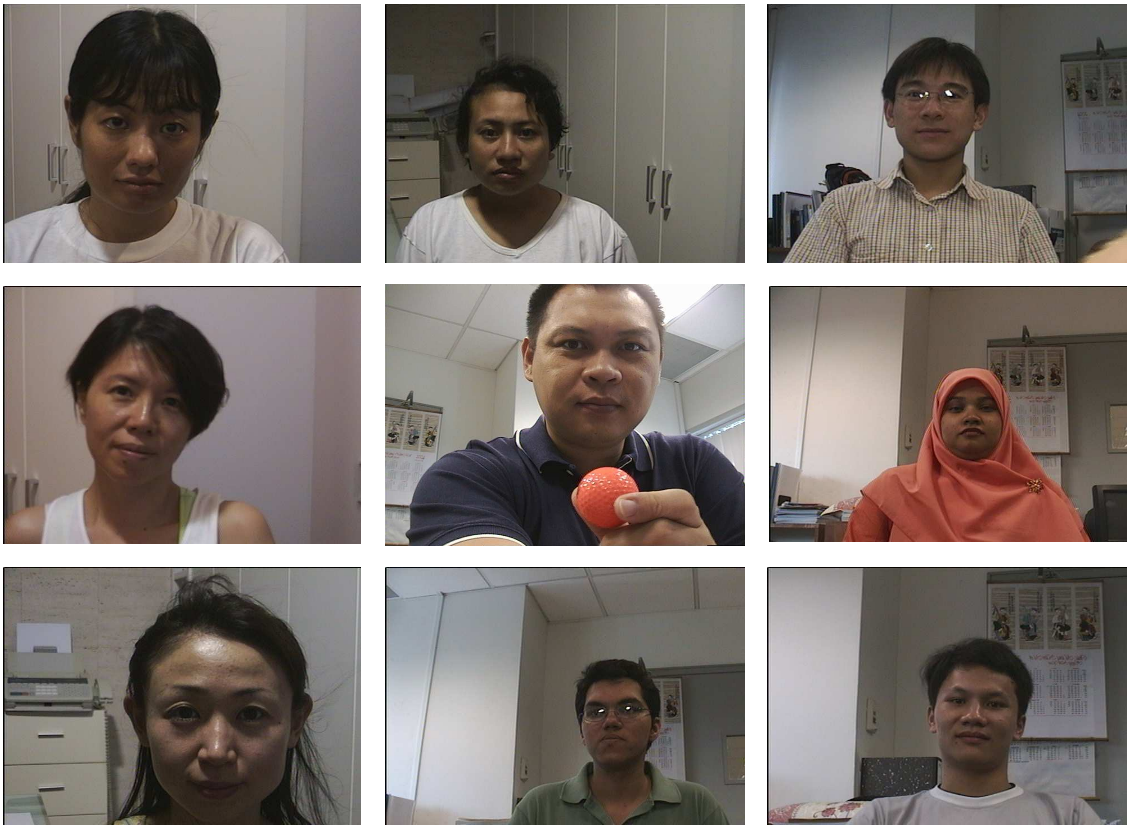


Figure A.2: Human subjects under White lighting conditions

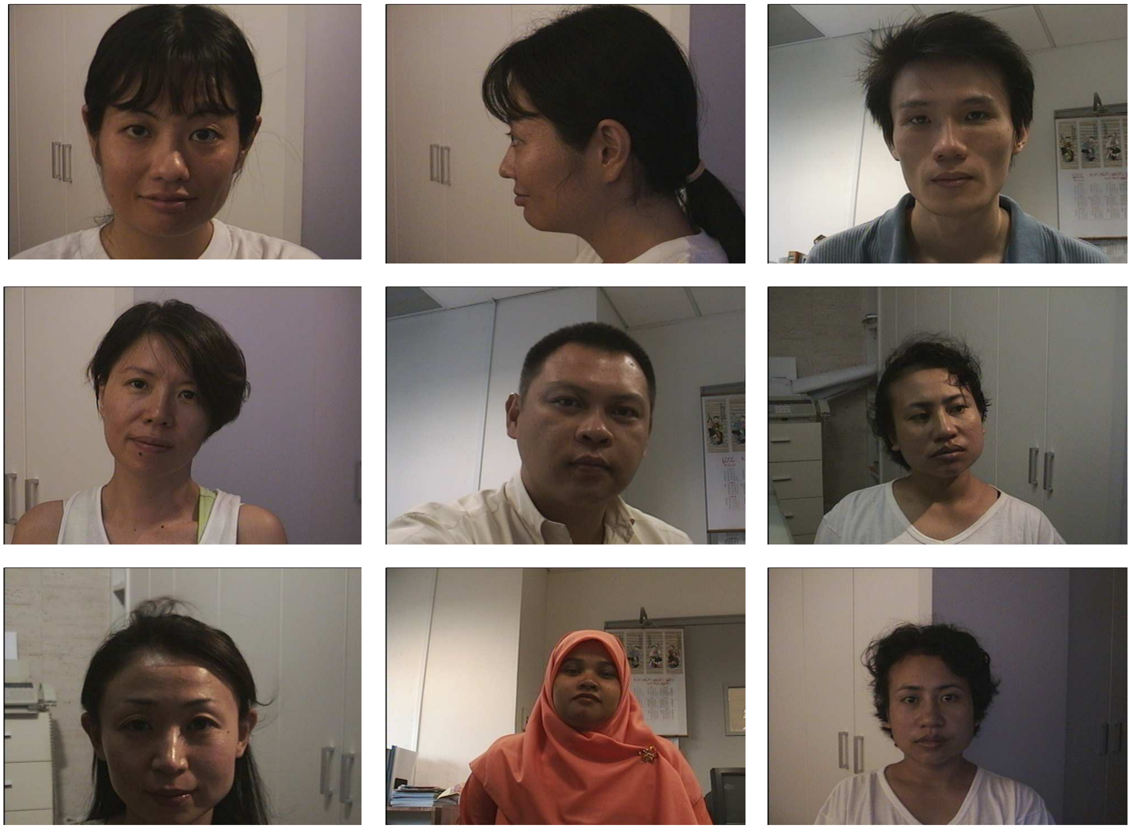


Figure A.3: Human subjects under Yellow lighting conditions

Appendix B

Training Sets - Objects

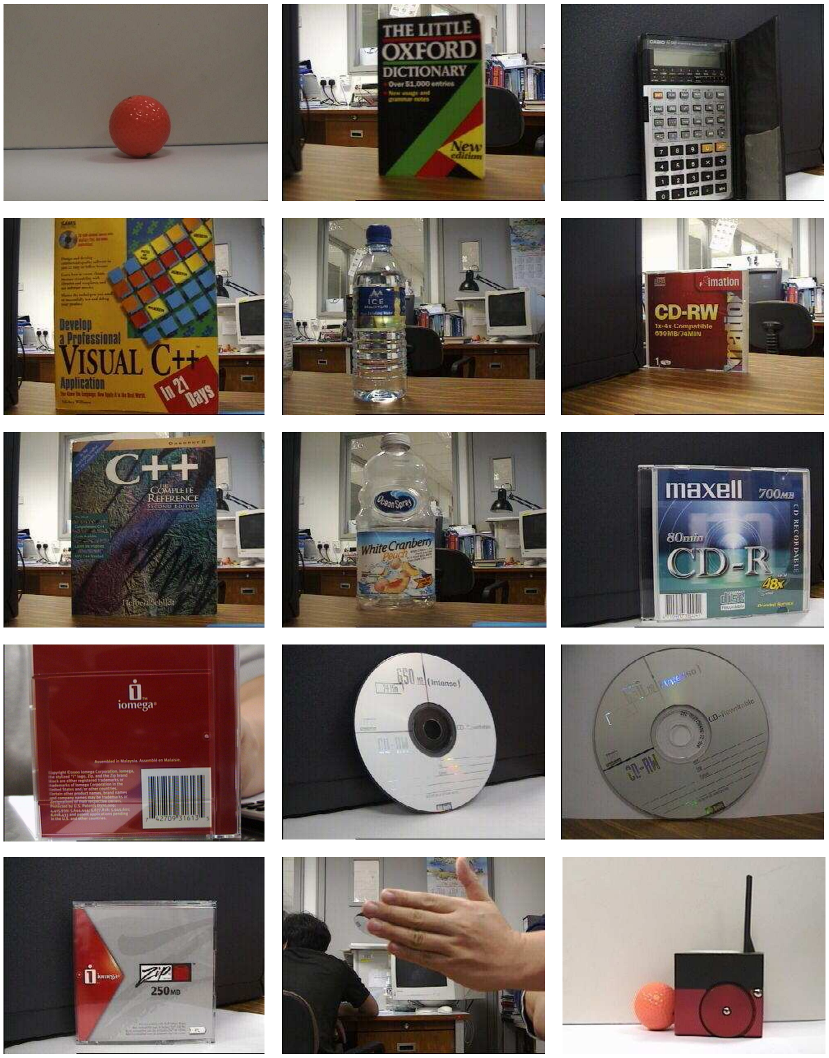


Figure B.1: Non-Human face objects

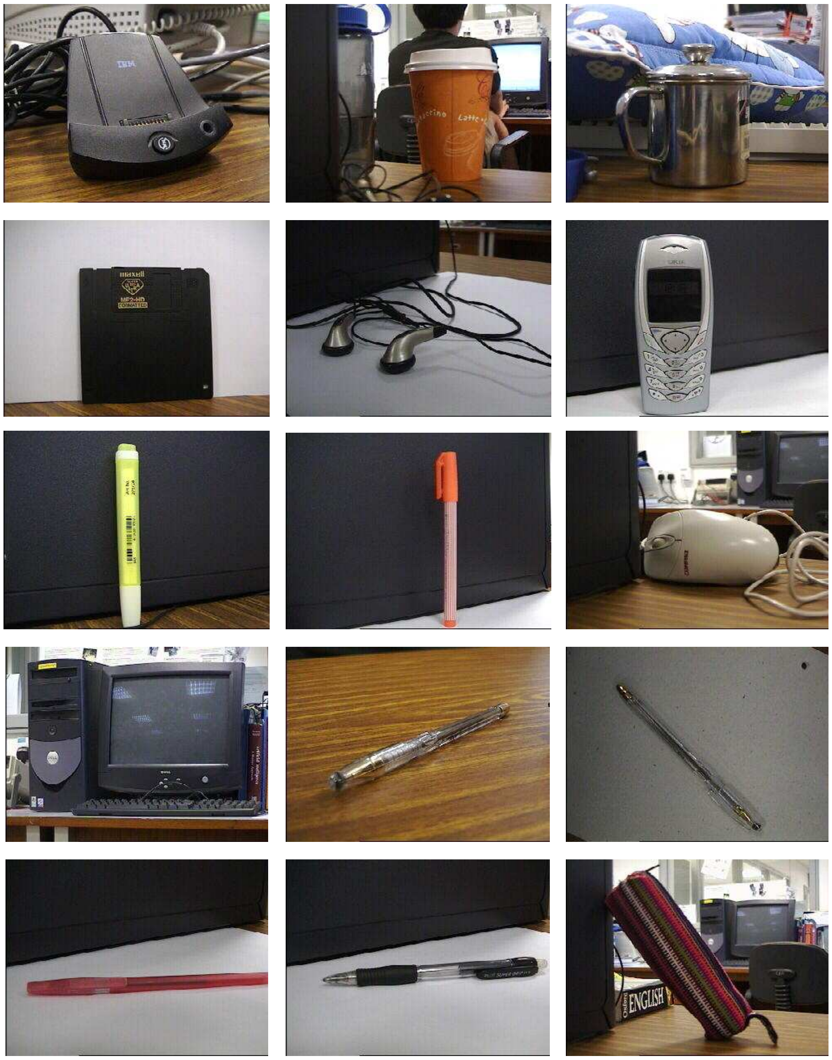


Figure B.2: Non-Human face objects

Appendix C

Training Sets - Human faces

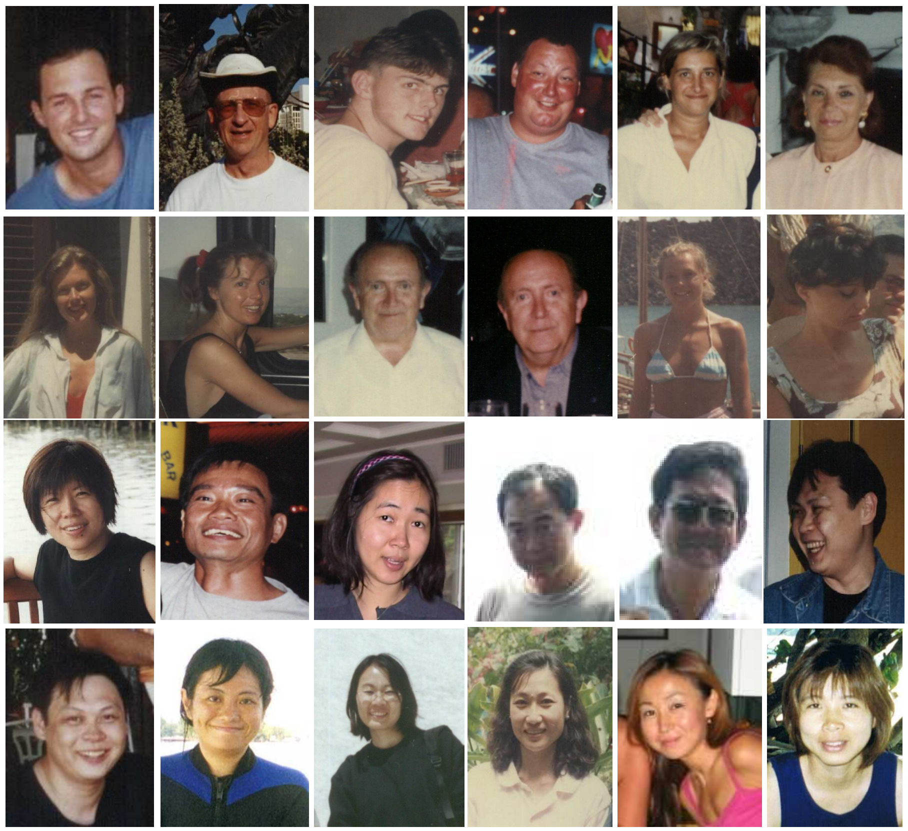


Figure C.1: Human faces from other camera sources

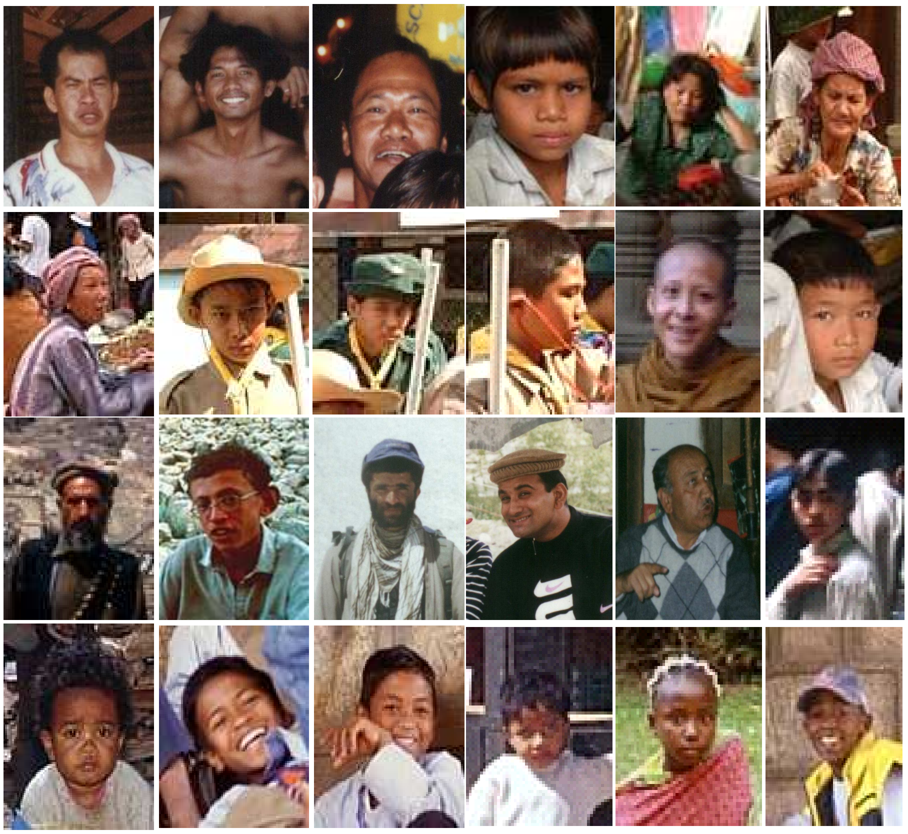


Figure C.2: Human faces from other camera sources

Bibliography

[1] Honda Motor Co. Ltd. Asimo

. *http : //world.honda.com/ASIMO/*.

[2] Sony Entertainment Robot Europe. Artificial intelligence companion, ers-7

. *http : //www.aibo − europe.com/*.

[3] Sony Robot. Sony dream robot, qrio

. *http : //www.sony.net/SonyInfo/QRIO/*.

[4] The RoboCup Federation. Robocup

. *http : //www.robocup.org/*02*.html*.

[5] Federation of International Robot-soccer Association. Fira

. *http : //www.fira.net/*.

[6] California Institute of Technology Jet Propulsion Laboratory. Spacecraft: Surface operations rover

. $http://marsrovers.jpl.nasa.gov/mission/spacecraft_surface_rover.html$.

[7] Orebck A. Lindstrm M. and Christensen H. Berra: A research architecture for service robots. In *IEEE proceedings – International Conference on Robotics and Automation*.

[8] Kwolek B. Face tracking system based on color, stereovision and elliptical shape features. In *IEEE proceedings – Conference on Advance Video and Signal Based Surveillance.*

[9] Lerdsudwichai C. and Abdel-Mottaleb M. Algorithm for multiple faces tracking. In *IEEE proceedings – Internarional Conference on Multimedia and Expo*, volume 2, pages 777–780, 2003.

[10] Shats A. Ruiz-del-solar J. and Verschae R. Real-time tracking of multiple persons. In *IEEE proceedings – Twelve International Conference on Image Analysis and Processing.*

[11] Gu I.Y.H. Liyuan Li, Weimin Huang and Qi Tian. Foreground object detection in changing background based on color co-occurrence statistics. In *IEEE proceedings – Applications of Computer Vision.*

[12] Chin-Seng Chua Ying Ren and Yeong-Khing Ho. Motion detection with non-stationary background. In *IEEE proceedings – Image Analysis and Processing.*

[13] Chaturvedi P. Poh M.M., De Silva L.C. and Guzman J.I. Human detection and tracking in real-time dynamic environment. In *In proceedings – Second International Conference on Computational Intelligence, Robotics and Autonomous Systems.*

[14] Fan Yang and Paindavoine M. Implementation if an rbf neural network on embedded systems: real-time face tracking and identity verification. In *IEEE proceedings – Transactions on Neural Networks*, volume 14-5, pages 1162–1175, 2002.

[15] Baluja S. Rowley H.A. and Kanade T. Neural network-based face detection. In *IEEE proceedings – Transactions on Pattern Analysis and Machine Intelligence*, volume 20-1, pages 23–38, 1998.

[16] Rajapakse M. Weimin Huang, Benghai Lee and Liyuan Li. Face recognition by incremental learning for robotic interaction. In *IEEE proceedings – Multimedia Signal Processing*.

[17] Qian Gu and Li S.Z. Combining feature optimization into neural network based face detection. In *IEEE proceedings – Fifteen International Conference on Pattern Recognition*, volume 2, pages 814–817, 2000.

[18] DeCarlo D. and Metaxas D. The integration of optical flow and deformable models with applications to human face shape and motion estimation. In *IEEE proceedings – Computer Vision and Pattern Recognition*.

[19] Saber E. and Tekalp A.M. Face detection and facial feature extraction using color, shape and symmetry-based cost functions. In *IEEE proceedings – Thirteen International Conference on Pattern Recognition*, volume 3, pages 654–658, 1996.

[20] Changmok Oh Kap-Ho Seo, Won Kim and Ju-Jang Lee. Face detection and facial feature extraction using color snake. In *IEEE proceedings – International Symposium on Industrial Electronics*, volume 2, pages 457–462, 2002.

[21] Klaus J. Kirchberg Oliver Jesorsky and Robert W. Frischhloz. Robust face

detection using the hausdorff distance. In *IEEE proceedings – Third International Conference on Audio- and Video-based Biometric Person Authentication*, volume LNCS 2091, pages 90–95, 2001.

[22] Srisuk S. and Kuratach W. New robust hausdorff distance-based face detection. In *IEEE proceedings – International Conference on Image Processing*, volume 1, pages 1022–1025, 2001.

[23] Gao Y. Efficiently comparing face images using a modified hausdoff distance. In *IEEE proceedings – Vision, Image and Signal Processing*, volume 150-6, pages 346–350, 2003.

[24] Spors S. and Rabenstein R. A real-time face tracker for color video. In *IEEE proceedings – International Conference on Acoustics, Speech and Signal Processing*, volume 3, pages 1493–1496, 2001.

[25] Kwolek B. Color vision based person following with a mobile robot. In *IEEE proceedings – Third International Workshop on Robot Motion and Control*.

[26] Mckenna S. and Gong S. Tracking faces. In *IEEE proceedings – Second International Conference on Automatic Face and Gesture Recognition*.

[27] Qian Chen Haiyuan Wu and Yachida M. Face detection from color images using a fuzzy pattern matching method. In *IEEE proceedings – Transactions on Pattern Analysis and Machine Intelligence*, volume 21, pages 557–563, 1999.

[28] Luo R.C. and Tse Min Chen. utonomous mobile target tracking system based on grey-fuzzy control algorithm. In *IEEE proceedings – Transactions on Industrial Electronics*, volume 47, pages 920–931, 2000.

[29] Goodridge S.G. and Kay M.G. Multimedia sensor fusion for intelligent camera control. In *IEEE proceedings – International Conference on Multi sensor Fusion and Integration for Intelligent Systems.*

[30] Bradski G.R. Real time face and object tracking as a component of a perceptual user interface. In *IEEE proceedings – Fourth Workshop on Applications of Computer Vision.*

[31] Collins R.T. Mean-shift blob tracking through scale space. In *IEEE proceedings – Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 234–240, 2003.

[32] Ramesh V. Comaniciu D. and Meer P. Real-time tracking of non-rigid objects using mean shift. Technical report, Siemens Corporate Research, 2000.

[33] Jie Yang and Alex Waibel. Tracking human faces in real-time. Technical report, School of Computer Science, Carnegie Mellon University, 1995.

[34] Bhme H.J. Wilhelm T. and Gross H.M. Sensor fusion for vision and sonar based people tracking on a mobile service robot. In *IEEE proceedings – Workshop on Dynamic Perception.*

[35] Fritsch J. Lmker F. Fink G.A. Kleinehagenbrok M., Lang S. and Sagerer G. Person tracking with a mobile robot based on multi-model anchoring. In

IEEE proceedings – Eleventh International workshop on Robot and Human Interactive Communication.

[36] Iossifidis I. Theis C. and Steinhage A. Image processing methods for interactive robot control. In *IEEE proceedings – Tenth International workshop on Robot and Human Interactive Communication.*

[37] Li S.Z. Liu F., Lin X. and Shi Y. Multi-modal face tracking using bayesian network. In *IEEE proceedings – International workshop on Analysis and Modeling of Faces and Gestures.*

[38] Andrew B. Watson. *Digital Images and Human Vision.* The MIT Press, first edition, 1993.

[39] E.J. Chichilnisky and B. A. Wandell. Multi-modal face tracking using bayesian network. In *Vision Research*, volume 39, pages 3444–3458, 1999.

[40] David G. Stork Richard O. Duda, Peter E. Hart. *Pattern Classification.* A Wiley-Interscience Publication, second edition, 2000.

[41] Chen Duan-sheng and Liu Zheng-kai. A novel approach to detect and correct highlighted face region in color image. In *IEEE proceedings – Conference on Advanced Video and Signal based Surveillance.*

[42] Nadimi S. and Bhanu B. Moving shadow detection using a physics-based approach. In *IEEE proceedings – 16th International Conference on Pattern Recognition*, volume 2, pages 701–704, 2002.

[43] Tominaga S. Surface identification using the dichromatic reflection model. In *IEEE Transactions – Pattern Analysis and Machine Intelligence*, volume 13-7, pages 658–607, 1991.

[44] Martinkauppi B. Soriano M., Huovinen S. and Laaksonen M. Skin detection in video under changing illumination conditions. In *IEEE proceedings – 15th International Conference on Pattern Recognition*, volume 1, pages 839–842, 2000.

[45] MAricor S. Birgitta M. and Matti P. Detection of skin color under changing illumination: A comparative study. In *IEEE proceedings – 12th International Conference on Image Analysis and Processing.*

[46] De Silva L.C. Hua R.C.K. and Vadakkepat P. Detection and tracking of faces in real-time environments. Technical report, National University of Singapore, 2001.

[47] YUJIN Robotics Co. Ltd. Yujin soccer robot
. $http : //www.yujinrobot.com/yujin - e.htm$.

[48] International Telecommunication Union Recommendations. 4:2:2 digital signal format, itu-r bt.601.

[49] Ling L. L. Liyanage C. De Silva Prahlad V., Peter Lim. Real time vision and autonomous mobile robot navigation. Technical report, National University of Singapore, 2001.

[50] Arbter K. Wei G. and Hirzinger G. Real-time visual servoing for laparoscopic surgery: Controlling robot motion with color image segmentation. In *IEEE Engineering in Medicine and Biology.*

[51] Kravtchenko V. and Little J. Efficient color object segmentation using the dichromatic reflection model. In *Pacific Rim Conference on Communications, Computer and Signal Processing.*

[52] Jean Y. Pingali G. and Carlbom I. Real time tracking for enhanced tennis broadcasts. In *IEEE proceedings – Computer Society Conference on Computer Vision and Pattern Recognition.*

[53] Yoo T. and Oh I. A fast algorithm for tracking human faces based on chromatic histograms. In *Pattern Recognition Letters*, volume 20, pages 967–978, 1999.

[54] Raja Y. McKenna S. and Gong S. Tracking color objects using adaptive mixture models. In *Image and Vision Computing*, volume 17, pages 225–231, 1999.

[55] Akita J. Real-time color detection system using custom lsi for high-speed machine vision. In *The Third International Workshop on Robocup.*

[56] Morreale V. Amoroso C., Chella A. and Storniolo P. A segmentation system for soccer robot based on neural networks. In *The Third International Workshop on Robocup.*

[57] Brusey J. and Padgham L. Techniques for obtaining robust, real-time, color-based vision for robotics. In *The Third International Workshop on Robocup.*

[58] Hrabec J. and Honzik B. Mobile robots playing soccer. In *Seventh International Workshop on Advanced Motion Control*.

[59] Rong Xiong Chao He and Lian kui Dai. Segmentation and identification in vision system for soccer robots. In *IEEE proceedings – Fourth World Congress on Intelligent Control and Automation*, volume 1, pages 532–536, 2002.

[60] Meyer F. and Bouthemy P. Estimation of time-to-collision maps from first order motion models and normal flows. In *IEEE proceedings – Eleventh IAPR International Conference on Computer Vision and Applications*.

[61] Stiller C. and Suntrup R. Parametric object motion estimation. In *Communications on the Move*, volume 2, pages 633–637, 1992.

[62] Govindu V. M. Combining two-view constraints for motion estimation. In *IEEE proceedings – Conference on Computer Vision and Pattern Recognition*, volume 2, pages 218–225, 2001.

[63] Sezan M. Qian R. and Matthews K. A robust real-time face tracking algorithm. In *IEEE proceedings – International Conference on Image Processing*.

[64] Foresti G. Object recognition and tracking for remote video survillance. In *IEEE proceedings – Transactions on Circuits and Systems for Video Technology*, volume 9-7, pages 1045–1062, 1999.

[65] iRobot. irobot. $http://www.irobot.com.$

[66] Sony Japan. Sony evi-d30, intelligent communication color video camera . $http://www.sony.net/Products/ISP/.$

[67] The MathWorks Inc. The mathworks . *http* : *//www.mathworks.com/*.

[68] Chai D. and Bouzerdown A. A bayesian approach to skin color classification in yc_bc_r color space. In *IEEE proceedings – Tenth Conference*, volume 2, pages 421–424, 2000.

[69] Kleinehagenbrok A. Fink G.A. Fritsch J., Lang S. and Sagerer G. Improving adaptive skin color segmentation by incorporating results from face detection. In *IEEE proceedings – Eleventh IEEE International Workshop on Robot and Human Interactive Communication*.

[70] Abdel-Mottaleb M. Rein-Lien Hsu and Jain A.K. Face detection in color images. In *IEEE proceedings – Transactions on Pattern Analysis and Machine Intelligence*.

[71] Kin-Man Lam Kwok-Wai Wong and Wan-Chi Siu. A robust algorithm for detection of human faces in color images. In *IEEE proceedings – 6th International Conference on Signal Processing*, volume 2, pages 1112–1115, 2002.

[72] Lukas Finschi. An implementation of the levenberg-marquardt algotithm. Technical report, Eidgenossosche Technische Hochsule Zurich, Institut fur Operations Research,, 1996.

[73] Li S.Z. Zhong X. and Teoh E.K. Facial feature extraction using bayesian shape model. Technical report, Microsoft research, face group publications, 2003.

[74] Ming hsuan Yang and Narendra Ahuja. Detection of human faces in color images. In *IEEE proceedings – International Conference on Image Processing*.

[75] LL Ling. Real-time face detection and tracking for robot navigation. Technical report, National University of Singapore, Final year project, 2002.

[76] Liao H.Y.M. Yu K. C. Han C.C. and Chen L. H. Fast face dection via morphology-based pre-processing. In *IEEE proceedings – Ninth International Conference on Image Acquisition*.