Tracking Epileptic Patients in Digital Videos for Automated Video-EEG Monitoring

Pramit Singh

A Thesis
in
the department
of
Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Applied Science at
Concordia University,
Montreal, Quebec, Canada

January 2007

Your file Votre référence
ISBN: 978-0-494-28928-0
Our file Notre référence
ISBN: 978-0-494-28928-0

Abstract

Tracking Epileptic Patients in Digital Videos for Automated Video EEG Monitoring

Pramit Singh

Video EEG monitoring is considered to be the most successful application for the diagnosis of the epileptic patients. The movements of the patient are recorded in the form of a digital video along with the EEG, over a significant duration of time. This enables the analysis of the behavior of the patient during the seizures, and is critical in determining the area of the brain that is responsible for the seizures. As the video recording is extensive in time, automatic tracking of the patient is a practical need. This would not only make the monitoring process error free, but would also lower the costs associated with the need for the human intervention. The aim of this thesis is to develop a system to automate the video EEG monitoring of the epileptic patients. Due to the prior information available about the physical environment of the patient, feature-based tracking method is preferred in this thesis over the motion-based techniques. The available features are identified and analyzed for developing the tracking algorithm. The skin color is used as one of the features and a new skin color detection filter, which is shown to perform reasonably well for detecting the human skin color, has been developed. The cap worn by the patient to support the electrodes on the head is developed into a second feature by drawing a pattern on the cap. A pattern recognition technique using the Hough transform for line detection is proposed to detect this feature. The two

features are jointly used together to develop an algorithm for locating the patient in the room. The tracking performance of this proposed feature-based algorithm is tested extensively under varying conditions and is shown to provide reasonable performance so that it can be used for practical implementation in a tracking system.

Acknowledgements

I would like to take this opportunity to cordially thank my supervisors Dr. M.N.S. Swamy and Dr. Rajeev Agarwal, for their continuous help and guidance, for the many fruitful discussions and sharing of their ideas. I am still amazed by the diversity and the spectrum of their scientific knowledge. I cannot emphasize enough how much I have benefited from their knowledge, wisdom and personality. They are always very generous in allocating time to their students. I am among the lucky ones, who have received their immediate attention and advice. I profusely thank them for allowing me to work on this project and also for their inspirational guidance and support throughout the course of my project. I also want to thank my friend Amanjot Kaur Bawa for her help and time in developing the test videos for this research.

I dedicate this project to my beloved parents, whose encouragement has supported me morally and emotionally. Their incessant care and interest in every detail of my life have supported me through the difficult times.

Pramit Singh, January 2007.

Contents

List of figures ix

List of tables xiii

1. Introduction 1

1.1 Background and motivation.....1

1.2 The Objective.....3

1.3 Object Tracking 4

1.4 Analyzing the problem.....7

1.5 Overview of the research.....9

1.6 Organization of the thesis.....10

1.7 Summary.....11

2. Possible Techniques and Algorithms 12

2.1 Introduction.....12

2.2 Texture recognition.....13

2.2.1 Tree structured wavelet transform.....14

2.2.2 Polarograms.....19

2.3 Ellipse detection in combination with skin filters.....22

2.3.1 Edge detection.....24

2.3.1.1 Sobel edge detection.....27

2.3.2 Least mean square fitting of ellipses.....28

2.3.3 Detecting human faces using ellipse detection and skin filters.....34

2.3.4 Performance Evaluation.....37

2.4 Summary.....40

3. Skin Color Detection in Digital Images **42**

3.1 Introduction.....42

3.2 Background.....42

3.3 Color models for skin classification.....44

3.3.1 RGB model for skin color.....46

3.3.2 HSB model.....48

3.4 Overview of David O'mara and Fleck & Forsyth models.....48

3.4.1 Fleck & Forsyth algorithm.....48

3.4.2 David O'mara skin classifier approach.....53

3.5 The proposed skin filter.....58

3.6 Evaluation of the proposed algorithm.....64

3.6.1 Results on videos.....65

3.6.1.1 Results for the video skinfilter_evaluation1.avi.....66

3.6.1.2 Results for the video skinfilter_evaluation2.avi.....69

3.6.1.3 Results for the video skinfilter_evaluation3.avi.....70

3.6.1.4 Results for the video skinfilter_evaluation4.avi.....74

3.7 Summary.....75

4. Line Detection using Hough Transform **78**

4.1 Introduction....78

4.2 Line detection using Hough Transform....79

4.2.1 Hough – Line transform....80

4.2.2 Hough peaks and accumulators....85

4.2.3 Using Hough peaks to detect lines....88

4.3 Summary....92

5. Tracking Patients using Hough Transform and Skin Color Detection **93**

5.1 Introduction....93

5.2 The electrode cap....94

5.2.1 The "electrode cap" as a feature....94

5.2.2 Developing a pattern on the cap using straight lines.... 94

5.3 The Algorithm....97

5.4 Finding the optimum parameters....103

5.4.1 The Hough line detection parameters....104

5.4.2 The window size....106

5.5 Evaluation of the algorithm....113

5.5.1 Evaluating in terms of lighting conditions....114

5.5.2 Evaluating under presence of multiple people....119

5.5.3 Real time processing....121

5.6 The tracking system....123

5.7 Summary....123

6. Conclusion and Future Work

6.1 Conclusion.....125

6.2 Future Work.....126

References **128**

List of figures

1.1 A system to automate the video EEG monitoring process.....3

1.2 A person under video-EEG monitoring, with electrodes on scalp and wires hanging through the electrodes- potential features that can be exploited for patient tracking.....7

2.1 Tree structured wavelet decomposition of a texture image by decomposing the bands successively based on their energy levels. (a), (b), (c) and (d) show the stages of successive decomposition.....15

2.2 (a) Decomposed bands for the texture image obtained after the tree-structured wavelet decomposition. (b) Tree-structured representation of the wavelet decomposition.....17

2.3 The band of highest energy after the decomposition process shown in black.....18

2.4 Code assignment using the tree for a texture.....18

2.5 (a) The texture image. (b) Fourier transform of the texture image shown in (a). (c) Rotated version of the texture image shown in (a). (d) Fourier transform of the texture image shown in (c).....20

2.6 (a) Standard texture image. (b) Polarogram of the texture image shown in (a). (c) Texture image shown in (a) with the source of illumination rotated by 45 degrees. (d) Polarogram of the texture image shown in (c).....21

2.7 (a) A sharp edge and its intensity profile. (b) A blurred edge having a ramp-like intensity profile.....25

2.8 (a) Intensity profile as in Figure 2.7(b). (b) First order derivative of the intensity profile shown in (a). (c) Second order derivative of the intensity profile shown in (a).....26

2.9 Sobel Operators.....27

2.10 A sample image and its edge detected output using Sobel Operators.....28

2.11 (a), (b) and (c), Ellipse detection on data points obtained by edge detection process on human head images extracted from frames having epileptic patients in field of view.....33

2.12 (a) Face detection with ellipse fitting followed by skin color detection. (b) Face detection with skin filtering followed by ellipse fitting.....35

3.1 The RGB space as a solid cube.....45

3.2 HSB model as a cone in the three dimensional space.....47

3.3 Frames 100, 200 and 300 from video skinfilter_test1.avi and their corresponding outputs from Fleck and Forsyth algorithm.....52

3.4 Distribution of hue and saturation for manually segmented skin regions in the training set [28].....54

3.5 Frames 100, 200 and 300 from video skinfilter_test1.avi and their corresponding outputs from David O'Mara's algorithm.....58

3.6 A combined skin color filter with Fleck and Forsyth and O'Mara's algorithms as two independent stages arranged in cascade.....60

3.7 (a) Original video frame. (b) Output obtained after two stages of filtering consisting of Fleck and Forsyth and David O'Mara's algorithm. (c) Output obtained by adding a third stage filter.....61

3.8 Comparison of the outputs of the three algorithms on frames 100, 200 and 300 from video skinfilter_test1.avi.....63

3.9 Frames 1 to 600 in steps of 100 from video skinfilter_evaluation1.avi and their corresponding selected and detected skin regions.....67

3.10 Frames 1 and 25 from video skinfilter_evaluation2.avi and their corresponding selected and detected skin regions.....69

3.11 Frames 1-800 in steps of 100 from video skinfilter_evaluation3.avi and their corresponding selected and detected skin regions.....72

3.12 Frames 1 and 100 from video skinfilter_evaluation4.avi and their corresponding selected and detected skin regions.....74

4.1 (a) Infinitely many lines pass through (x_i, y_i) for different values of a and b. (b) The family of lines in (x, y) space passing through (x_i, y_i) represented by a single line in (a, b) space.....80

4.2 (a) Two points lying on the same line in (x, y) space. (b) Line through points u and v in (x, y) space is now point of intersection of lines u and v in (a, b) space..81

4.3 All points on a line in (x, y) space are represented by concurrent lines in (a, b) space.....82

4.4 Normal representation of a line.....83

4.5 (a) Two points in (x, y) space. (b) Curves in the (ρ, θ) space corresponding to the two points in (x, y) space.....83

4.6 A vertical line in (x, y) space can be represented in (ρ, θ) space by a point for which $\theta = 90$ degrees.....84

4.7 (a) Points lying on a line in (x, y) space. (b) The corresponding curves in (ρ, θ) space

are concurrent.....85

4.8 (a) Lines obtained from the real time images might not be perfectly straight. (b): More than one peak is possible in *(ρ, θ)* space for this line.....86

4.9 The (ρ, θ) parameter space subdivided into accumulator cells.....87

4.10 Image under test having random lines and curves.....88

4.11 Edge detected logical image using Sobel operator gives the points in the image which are candidates for straight lines.....89

4.12 The Hough peak detection step figures out the brightest peaks in the *(ρ, θ)* space.....90

4.13 Line detection performed based on the Hough peaks in Figure 4.12. Detected lines in the image shown in red.....91

5.1 Simulation conditions with patient wearing the modified cap.....95

5.2 Frame 100 from the video evaluation1.avi.....98

5.3 Output of the skin color filter.....98

5.4 Centroids of the three significant skin regions marked as blue color asterisks.....99

5.5 Edge detection performed on original frame.....100

5.6 The extracted windows centered at the centroids of the three major skin color regions (magnified for better visibility).....100

5.7 The Hough peaks for the ROI windows shown in Figure 5.6.....102

5.8 The detected lines corresponding to the Hough Peaks.....103

List of tables

2.1 RESULTS OF THE TRACKING PROCESS FOR EACH FRAME BEING PROCESSED IN THE VIDEO ELLIPSE.AVI.....38

3.1 STATISTICAL VALUES OF SKIN HUE CALCULATED ON THE TRAINING SET [28].....55

3.2 STATISTICAL VALUES OF SKIN SATURATION CALCULATED ON THE TRAINING SET [28].....55

3.3 RESULTS ON THE VIDEO SKINFILTER_EVALUATION1.AVI.....68

3.4 RESULTS ON THE VIDEO SKINFILTER_EVALUATION2.AVI.....70

3.5 RESULTS ON THE VIDEO SKINFILTER_EVALUATION3.AVI.....72

3.6 RESULTS ON THE VIDEO SKINFILTER_EVALUATION4.AVI.....74

5.1 EVALUATION OF THE OPTIMUM WINDOW SIZE FOR VIDEO EVALUATION1.AVI.....108

5.2 EVALUATION OF THE OPTIMUM WINDOW SIZE FOR VIDEO EVALUATION2.AVI.....111

5.3 RESULTS OF TRACKING PROCESS ON EACH FRAME BEING PROCESSED FOR VIDEO UNDER GOOD LIGHTING CONDITIONS – TEST1.AVI.....115

5.4 RESULTS OF TRACKING PROCESS ON EACH FRAME BEING PROCESSED FOR VIDEO UNDER MEDIUM LIGHTING CONDITIONS – TEST6.AVI.....116

5.5 RESULTS OF TRACKING PROCESS ON EACH FRAME BEING PROCESSED FOR VIDEO UNDER LOW LIGHTING CONDITIONS – TEST2.AVI.....117

5.6 RESULTS OF TRACKING PROCESS ON EACH FRAME BEING PROCESSED FOR VIDEO UNDER TEST WITH MULTIPLE PEOPLE – TEST8.AVI.....120

5.7 RESULTS OF TRACKING PROCESS ON EACH FRAME BEING PROCESSED FOR VIDEO UNDER TEST WITH MULTIPLE PEOPLE – TEST9.AVI.....120

5.8 REAL TIME PROCESSING COMPUTATIONS FOR VIDEOS TEST1.AVI – TEST9.AVI.....122

Chapter 1

Introduction

1.1 Background and motivation

Epilepsy is a name given to the collection of disorders of the brain that is often characterized by recurrent seizures caused by synchronized electrical neuronal activity. Around 50 million people in the world are known to be affected by epilepsy. Proper diagnosis of epilepsy is important to differentiate between epileptic and non epileptic seizures and more importantly, to determine those areas of the brain tissues that may be responsible for the seizures. Electroencephalography (EEG), magnetic resonance imaging (MRI), single photon emission computed tomography (SPECT), positron emission tomography (PET) and magneto encephalography (MEG) are some common diagnostic tools for epilepsy but long term video-electroencephalography monitoring is the most comprehensive and the most commonly used technique for the diagnosis of the condition. Electroencephalography is a method of recording the electrical activity of the brain using electrodes connected to the scalp. The signals received from these electrodes are known as the EEG signals. The EEG recorded from the electrodes placed on the scalp is a collection of the total activity of many neurons in the brain. Long term video-EEG monitoring involves simultaneous recording of the EEG and capturing the movements of the patient in a video over a period of 5-6 days or longer [1]. The doctors view the video and the EEG data simultaneously in order to observe the patient's behavior during the

seizure. By correlating the clinical behavioral manifestation from the video with the EEG helps the doctors in determining whether the electrical activity associated with the brain is actually an epileptic seizure and more importantly, in the localization of the seizure origin by identifying the region of the brain that may be responsible for the seizures [2].

The present EEG-video monitoring setup requires a patient to stay in a hospital room, where the EEG maybe recorded for a period of 5-6 days or more. During this period, the patient's behavior and movements are recorded in the form of a digital video captured by a video camera placed in the room. The video camera is typically controlled manually by a technician and often the camera goes unattended resulting in a loss of valuable data. It is important to keep the patient in full view at all times as seizures occur unexpectedly, and the clinical component of the seizure needs to be captured. As long term data is being recorded, the patient is expected to move about in the room, and this requires continuous adjustments in the position of the camera. Sufficient video EEG data is required by the doctors to analyze the epilepsy in the patient and if the video segments during occurrence of seizures are lost, then it would unnecessarily prolong the stay of the patient at the hospital. Thus, continuous human intervention to control the camera makes the video-EEG monitoring a very expensive process, as it requires a technician to be present all the time and any error on the part of the technician would require an extended stay of the patient in the hospital resulting in added costs.

If the human intervention is replaced by an automated system to track the patient [3], then no technician would be required to control the camera and the tracking process would be free of human errors. It would thus significantly reduce the cost of the video-EEG

monitoring, thereby making it more affordable to the patients and more importantly, for providing a better and faster diagnosis of the condition.

1.2 The Objective

The previous discussion on the problems with the current video EEG monitoring setup underlines the need for an automated system. Figure 1.1 shows a possible system.

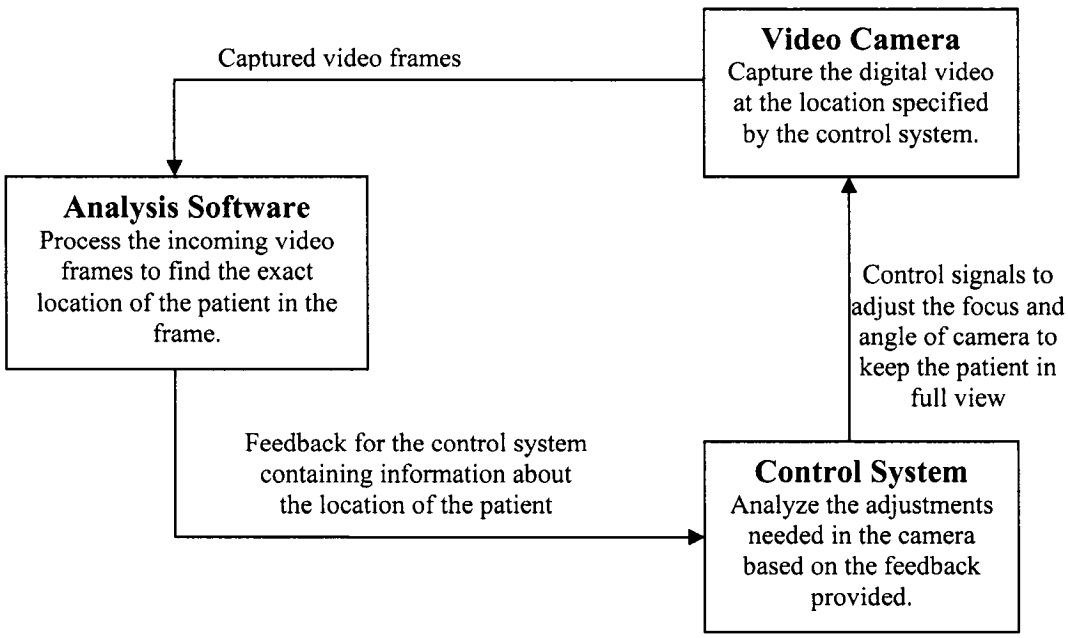


Figure 1.1. A system to automate the video EEG monitoring process.

The video camera in the hospital room captures the frames of the environment. These video frames are processed by the analysis software, which comprises of algorithms to process the video frame and determine the coordinates of the patient's location in the video frame. Depending upon the location of the patient in the current frame, the control system determines the adjustments needed in the angle of the camera so that the patient

remains in full view of the camera for the next frame. The adjustments in the camera can thus be made automatically through signals from the control system to the camera.

The core of this system lies in the algorithms of the analysis software, which process each video frames to determine the current location of the patient in the room. The objective of this research is to develop an algorithm that can be used to track the patient in the video frame.

1.3 Object Tracking

Tracking and recognizing objects in digital videos is one of the most popular, yet complex tasks in video processing. With the advancements in technology and security requirements, the tracking problem is having increasingly wide applications in daily life [4] – [6]. The applications of tracking are broadly classified based on the requirements such as the motion capture, recognition from motion, surveillance and targeting [7].

Motion capture is a technique for tracking a person or object so that its motion can be recorded and artificially recreated, as and when required. This has applications in animations, Hollywood stunts, etc. Recognition from motion is useful when the motion characteristics of an object are unique. This record of expected motion can help in identifying the objects at other places. Surveillance of objects is important for security reasons at airports, car parking, highways, etc. Apart from commercial use, object tracking is also useful in military applications for target locating and target striking.

With limitless applications of object tracking in digital videos, it is obvious that the amount of research that has been carried out in this field has been enormous. Object tracking problem has been widely considered through methods like frame difference,

motion detection, phase correlation, feature extraction, etc. In motion detection-based methods, the first step is the estimation of the motion vectors [8]. Gradient-based methods, such as the optical flow detection for estimating the motion vectors, are highly accurate [9], but generally come with increased computational overhead than the block-based matching techniques. An accurate knowledge of all the motion vectors in a sequence provides the means to segment the image into pixels associated with a moving object and those associated with the rigid background.

In a typical block-based method, which is considered to be the most efficient, the motion estimate is found by a spatial-domain search procedure [10]. The object block of the current frame is placed and moved around in the previous frame using a specific search strategy. A criterion is defined to determine how well the block matches the corresponding block in the previous frame. Thus, the direction of motion can be analyzed. Other motion estimation algorithms like the phase correlation method [11], measure the movement between two fields directly from their phases, thereby providing a faster method for the calculation of the motion vectors. Other motion estimation algorithms have been developed with slight variations, giving improved performance depending upon the area of application [12].

For tracking human targets in a video, detecting the shape and position of the target is the fundamental task. Since the shape of the human object is subject to a deformation and random motion in the two dimensional image space, techniques like active shape models have to be implemented. Active shape models (ASM) fall into the category of deformable shape models with *a priori* information about the object [13]. Active shape models define the shape of the object using a set of most prominent points. The locations of these points

are updated with the movements of the object to form a training set. The object shapes are recognized by locating the prominent points and comparing their locations to the training sets.

Most motion-based techniques involve keeping track of the motion of the target and maintaining a hold on the target. Another solution to track an object is the feature-based tracking. The feature-based tracking requires the recognition of patterns or specific features like shape, size, color etc. of the target object. This object-specific information is useful in tracking the target and in regaining its location whenever the tracking is lost. Feature-based tracking can either be used alone or in combination with motion-based techniques depending on the computational complexity involved at different stages of the feature recognition.

With the numerous techniques available for solving the tracking problem, it is very easy to get lost in the web of the available algorithms. Therefore, it becomes very important to analyze the problem properly and understand the constraints involved, so that an optimum technique can be developed by exploiting the algorithms that would work best under a given set of circumstances.

1.4 Analyzing the problem

Object tracking algorithms have been developed for various areas of application with considerable accuracy, but exploitation of the available features in conjunction with real time performance has eluded the researchers for years. Motion-based techniques fail to give a reliable answer consistently at all times and are known to suffer from the problem of loss of tracking, for example, due to the effect of shadow movements, etc. When the

object to be tracked has a sufficiently large set of detailed features that can differentiate it from the background, then the feature-based tracking is expected to be more reliable as it does not depend upon any redundant information.

The problem at hand is to identify and track a patient in a video, when the EEG of the patient is simultaneously being recorded. The EEG recording is done using a set of electrodes attached to the patient's scalp. These electrodes are connected through wires to the EEG recording device. To hold the wires and the electrodes, the patient is required to wear bandages or a cap that fits snugly onto the head. Figure 1.2 shows a person under video-EEG monitoring. Since it is a human being that is being tracked, human shapes and skin color are very obvious features to be used. Another potentially distinctive feature is that the human being tracked is a patient and is wearing a cap with wires hanging from the head. The environment in which the video-EEG is performed is that of a hospital, and hence the patient maybe expected to wear some specific clothing (i.e. hospital gown etc.). All these features make a very strong case for exploiting the feature-based tracking as a solution to the problem at hand.

Figure 1.2. A person under video-EEG monitoring, with electrodes on scalp and wires hanging through the electrodes- potential features that can be exploited for patient tracking.

Typically, patients monitored for long term video-EEG are limited in the range of activities and movements when confined to their rooms. The motion is not expected to be fast, and this is yet another reason the feature-based tracking may be a better choice. The patient's movements are expected to be slow; therefore, the tracking algorithm can be complex and computationally expensive in order to maintain accuracy at the expense of speed.

The tracking environment is that of a hospital, where the patient has to stay for several days. The illumination conditions in the room are expected to change over the day, as the patient may prefer a lower level of illumination while sleeping. The illumination conditions may also change, when the source of light changes from the natural to artificial. These changing illumination conditions are expected to pose a big challenge, if the features to be exploited are significantly dependent upon the illumination levels. The presence of visitors in the room implies that the algorithm should be able to make a distinction between a patient and the other people in the room. The human shadows make shapes similar to those of the human bodies; therefore the algorithm should be able to make a distinction between shadows and real body parts. Since feature-based methods rely upon a combination of features to make decisions, they are expected to be more reliable than the motion-based methods.

The availability of several features is a good reason to exploit them for tracking the patient. Results from companies like Stellate, where this tracking problem has been studied, indicate that the motion-based algorithms like Cam shift [14] have failed to

provide a reliable answer. Thus, with a known target to be tracked under an expected environment, feature-based tracking is expected to perform well and thus is the focus of this research.

1.5 Overview of the research

The aim of this research is to detect the accurate location of the patient in the room so that a feedback can be provided to the system that controls the movement and focus of the camera. This is to ensure that the patient stays in full view of the camera at all times.

In this work, this is achieved by exploiting the various features identified with the patient and his environment. Specifically, the presence of the human skin, the shape of the human head, the clothes and the cap supporting the electrodes will be exploited as the features. Skin-color filters are developed for the purpose of detecting the human skin color and an effort is made to detect the human head using elliptical shapes. The combination of the human skin color identification along with the elliptical-shaped head provides one possibility of locating the human head in the video frames. We also investigate the possibility of using the patient clothing by analyzing the texture of the clothes using various algorithms. However, as will be seen in Chapter 2, due to some of the limitations of the detection of the elliptical shapes, the performance of the detection of the human head is not satisfactory and the recognition of the texture of the clothes is not found to be a very feasible solution. This opens up the need of developing some other feature, possibly some pattern on the cap supporting the electrodes.

For this purpose, we create a new feature by placing a pattern on the cap. This pattern along with the skin color filter provides a solution to the tracking problem. The

performance of the final tracking process is measured by counting the percentage of the time the algorithm is able to locate the patient accurately under different conditions. The performance is tested under challenging conditions, like the presence of multiple people in the room and changing the illumination conditions. Since the movements of the patient are expected to be very slow, the processing of one frame per second is assumed to be sufficient. The success rate of tracking of the algorithm thus developed is found to be quite high.

1.6 Organization of the thesis

Chapter 2 starts with a discussion of the algorithms that are investigated to recognize the features that help in identifying the patient in video frames. The initial work that involves experiments using methods such as texture recognition and the use of the skin color filter, along with the detection of an elliptical shape for recognizing the human head has been discussed. The development of the skin color filter itself is discussed in detail in Chapter 3. Chapter 4 introduces the Hough transform for line detection, which is the method that is finally used for recognizing the pattern artificially created on the cap supporting the electrodes on the head of the patient. By using the algorithms for detecting the human skin color and a pattern of straight lines, the development of the final algorithm and the performance assessment of the method is done in Chapter 5. Chapter 5 starts with a discussion on the development of the special feature that involves a pattern of straight lines on the cap supporting the electrodes. The details of the final algorithm are then explained which uses the skin color detection along with Hough transform for detecting the pattern on the cap to locate the patient in the room. The final algorithm is tested under

different conditions and its performance analysis is presented. Chapter 6 contains some conclusions and scope for further research.

1.7 Summary

In this chapter the problem of tracking an epileptic patient in real time videos has been presented. The possible solutions to this problem have been discussed and with the given physical environment and expected performance, the feature-based tracking has been mentioned as the method that can provide better results than the motion-based techniques. For the considered application the available features which can be exploited have been listed and possible techniques to identify these features have been discussed in brief. In the Chapters to follow, these techniques are presented in detail along with their limitations and performance evaluation.

Chapter 2

Possible Techniques and Algorithms

2.1 Introduction

As mentioned in Chapter 1, the purpose of this research is to develop an algorithm that can determine accurately the location of the patient in the room so that the video camera can make adjustments to keep the patient in full view all the time. Following the discussion in the previous chapter, the feature-based tracking is assumed to be the best possible solution to this problem. To achieve feature-based tracking, the available features are identified and techniques are developed to parameterize these automatically.

The potential features like human skin color, the elliptical shape of the head, and the texture of the patient's clothes are tested using different algorithms. Techniques like the tree-structured wavelet transform and polarograms are studied for recognizing the texture of the patient's clothes. A skin color filter is developed for recognizing the potential human skin areas, and will be discussed in Chapter 3. Least mean square fitting of ellipses is used to recognize the human head in the form of an elliptical shape. A combination of the elliptical fitting and the human skin color filter is tested for detecting the human faces.

This chapter discusses these techniques and algorithms implemented to conduct research in the area of tracking in the initial stages and to understand the limitations of

the tracking problem at hand. The chapter starts with a discussion of the texture recognition as a feature and then moves on to the development of a new face detection algorithm that uses the skin color as a feature, which is implemented for the purpose of finding the location of patient in the video frame.

2.2 Texture recognition

Though the texture has not yet been formally defined in the literature, it can be understood as the visual and sensational properties of any surface. Each surface is expected to have a unique texture and thus can be differentiated from other surfaces. If the texture of a surface can be analyzed, then the recognition of that surface can be done automatically. This leads to the development of texture recognition algorithms aimed towards measuring the properties of surfaces in order to recognize textures on the basis of such properties.

Like any other surface, each type of clothing has a unique texture and can be differentiated from other clothes by virtue of its texture. If the patients are made to wear particular clothes whose texture or pattern is known, then this can be extracted as a feature to become a permanent part of the tracking environment. The tracking algorithm could either involve pattern recognition or texture analysis of the clothes or both. The problem of recognizing some specific pattern on the patient's clothes does not seem to be feasible, since the pattern is subject to deformation as the field of view changes with the motion of the patient. Recognizing the texture of the clothes can be a practical option as it does not change with the motion of the target. For the recognition of the texture, we

attempt to use the wavelet and Fourier transform-based methods. These are discussed below.

2.2.1 Tree structured wavelet transform

The tree-structured wavelet transform for texture recognition was introduced by Chang and Kuo [15]. In their introductory paper on tree-structured wavelet transform, they have made a performance comparison of their wavelet transform-based texture recognition method against other methods based on discrete cosine transform, discrete sine transform, discrete Hartley transform, pyramid structure wavelet transforms, Gabor filters and Laws filters. They showed their method to be computationally less expensive while giving better recognition results [15].

The commonly used pyramid-structured wavelet decomposition of images, which has been widely used for image compression and other areas of research like data hiding, video coding, image watermarking, etc., is not suitable for texture recognition. The texture information is known to lie in the mid frequency regions [16], thus the pyramid structured wavelet decomposition, which iteratively decomposes the low frequency regions, is ineffective. In the tree-structured wavelet transform, the decomposition is done based on energy levels of the decomposed bands. Instead of decomposing the low frequency regions at each level of decomposition, the regions with high energy are determined for further decomposition. This unfolds the spectrum of the texture sequentially.

The tree-structured wavelet transform for texture recognition can be understood with the aid of a simple example shown in Figure 2.1.

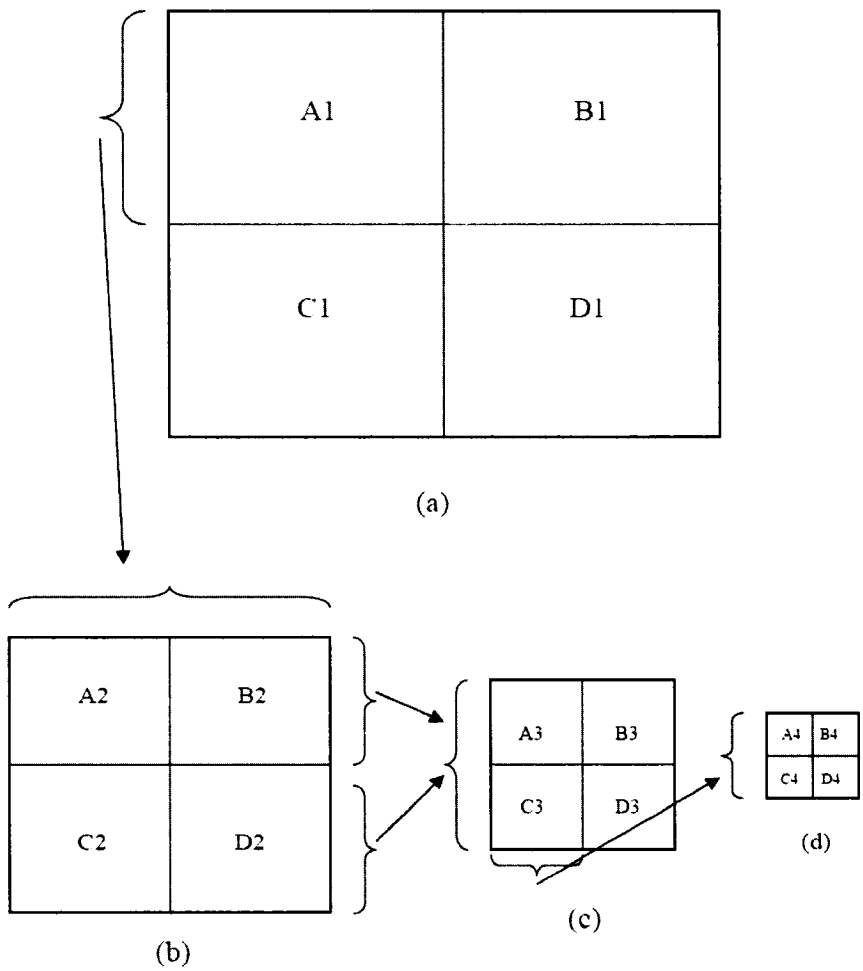
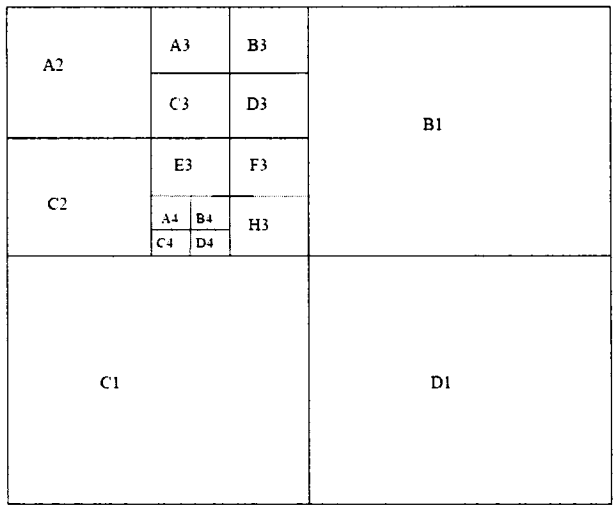


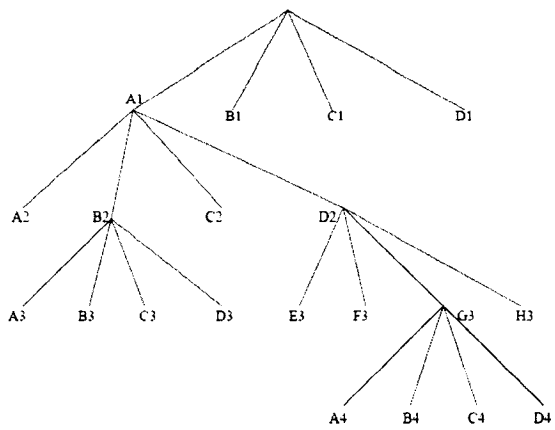
Figure 2.1. Tree structured wavelet decomposition of a texture image by decomposing the bands successively based on their energy levels. (a), (b), (c) and (d) show the stages of successive decomposition.

In the first step, the texture image under test is decomposed using wavelets into four sub bands as shown in Figure 2.1 (a). An energy threshold is chosen depending upon the level of accuracy desired for the recognition process. Only the sub bands with energy greater than the threshold are decomposed. In our example, the sub band A1 is found to be above the threshold level and thus is decomposed in the second iteration (see Figure 2.1(b)). The decomposed sub band energies are again tested and further decomposition is done accordingly. In our example, both B2 and D2 are found to lie above the threshold and thus are decomposed independently, as shown in Figure 2.1(c). This iterative process is continued until there are no more sub bands with energy greater than the threshold or no further decomposition of sub bands is possible. In our example C3 is again found to be above the threshold and is then decomposed (see Figure 2.1(d)).

Figure 2.2 (a) shows the final decomposed bands obtained after the tree-structured wavelet transform. The decomposition of the wavelet sub bands is successive, and hence can also be seen as a tree structure as shown in Figure 2.2(b). The branches of the tree progress downwards under the roots which are being decomposed. The tree stops growing when sufficient number of branches have grown or there is no more energy in the roots to grow more branches.



(a)



(b)

Figure 2.2. (a) Decomposed bands for the texture image obtained after the tree-structured wavelet decomposition. (b) Tree-structured representation of the wavelet decomposition.

The textures are assigned unique codes corresponding to the location and the level of the decomposition of these bands. The length of the code and the code sequence depends upon the location of the smallest band with energy greater than the threshold. For instance, at the end of the decomposition process for our example in Figure 2.2, the band shown in black in Figure 2.3 has the highest energy. The location and size of this band

obtained after the tree-structured decomposition process is unique for a texture and can be used to identify the texture. Figure 2.4 shows the code assignment process by using the tree. The code for the texture sample comes out to be ADCC.

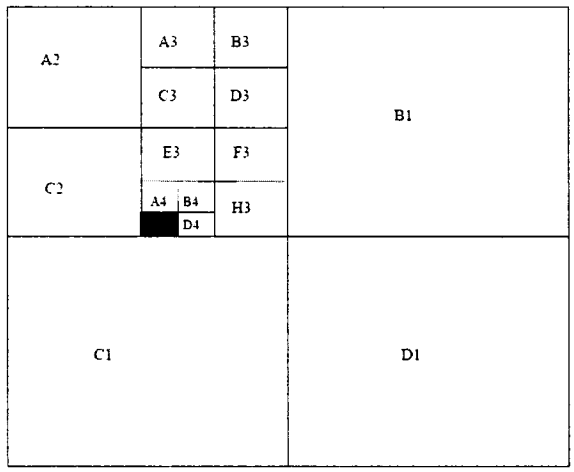


Figure 2.3. The band of highest energy after the decomposition process shown in black.

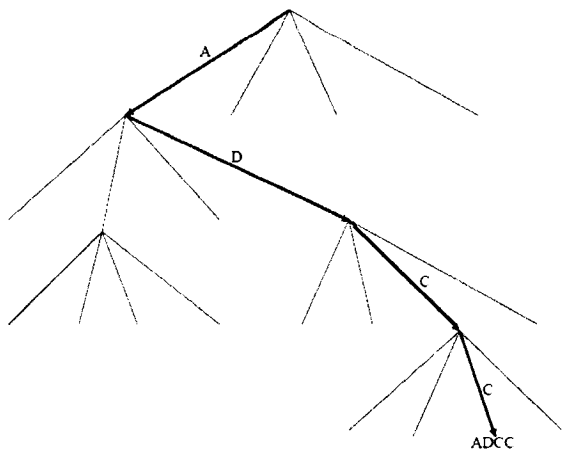


Figure 2.4. Code assignment using the tree for a texture.

The texture recognition is performed by calculating the code for a given texture image and comparing the code obtained with a priori known codes for the texture to be detected.

2.2.2 Polarograms

Polarogram technique is another method for recognizing textures. As each texture is unique, so is its frequency spectrum. The major problem with tree-structured wavelet transform-based method is the orientation of the texture sample. The tree-structured wavelet transform of the texture would give different codes for different orientations of the same image sample, making the recognition process complicated for practical applications. The orientation problem can be addressed by the polarogram technique [17], which computes the Fourier spectrum of the sample and represents it in polar plane. The spectra in polar planes are called polarograms.

Figure 2.5(a) shows a sample texture image and a rotated version of this image is shown in Figure 2.5(b). Their respective polarograms shown in Figure 2.5(c) and Figure 2.5(d) indicate that the polarogram of a rotated sample is also the rotated version of the polarogram of the original sample.

The frequency spectra for the standard textures are generally known *a priori*. The texture recognition step matches the polarogram obtained for the image sample under test with the standard polarogram for the texture that is being searched. This matching is performed by rotating the polarogram obtained in steps of one degree and matching each rotated version of the polarogram with the standard polarogram. If the error is found to be within acceptable limits, then a successful recognition of the texture can be assumed.

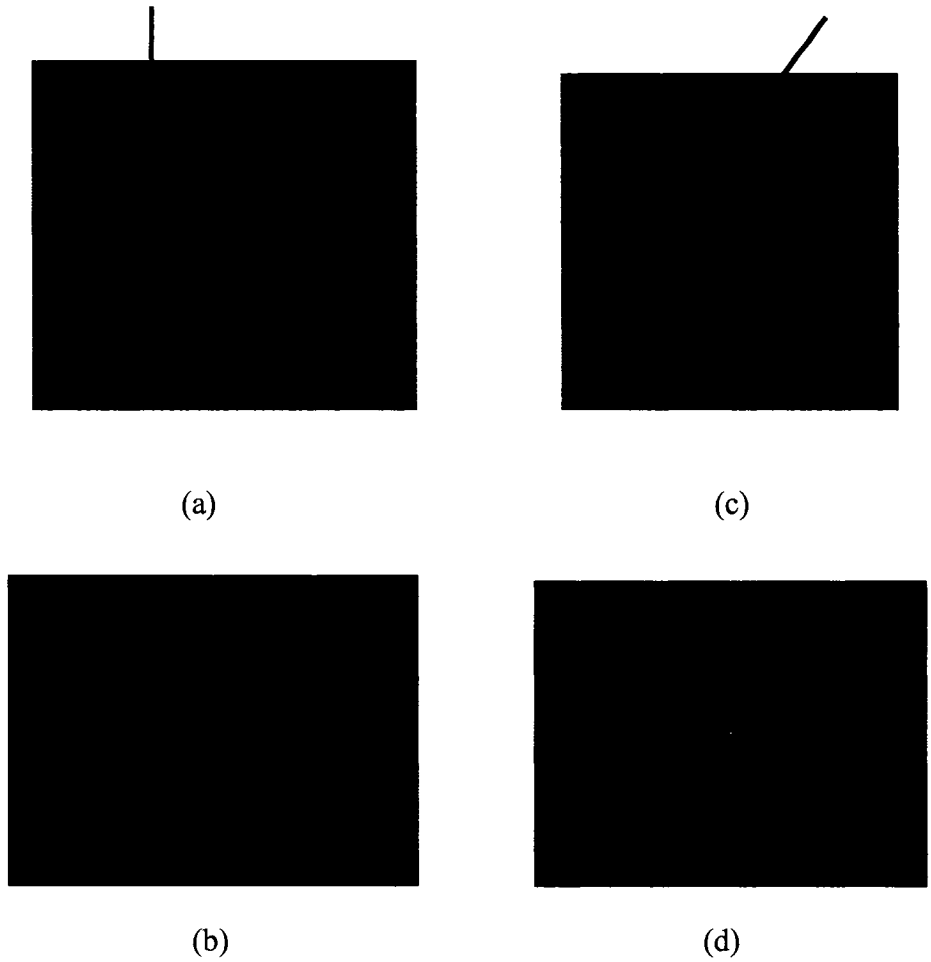


Figure 2.5. (a) The texture image. (b) Fourier transform of the texture image shown in (a). (c) Rotated version of the texture image shown in (a). (d) Fourier transform of the texture image shown in (c).

The polarogram technique can also withstand the rotations of the source of the illumination. Figure 2.6(a) shows a texture sample with another sample of similar texture shown in Figure 2.6(b), having the illumination source rotated by 45 degrees as shown by the arrows. The polarograms for these two samples indicate a rotation by 45 degrees in the spectrum.

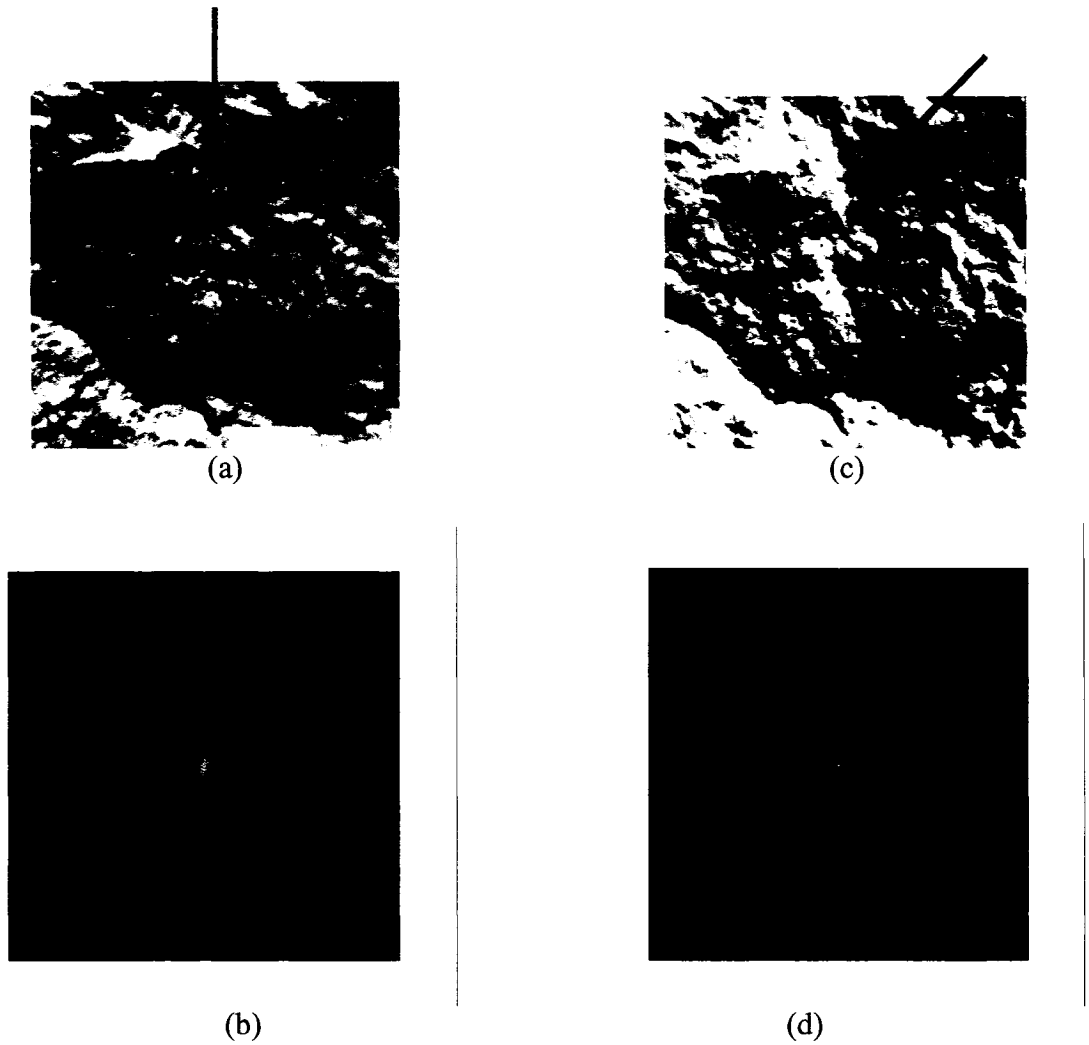


Figure 2.6. (a) Standard texture image. (b) Polarogram of the texture image shown in (a). (c) Texture image shown in (a) with the source of illumination rotated by 45 degrees. (d) Polarogram of the texture image shown in (c).

Thus, the polarograms are better suited for practical applications, where both the rotation of the textures and a change in the direction of the illumination source maybe expected. Though the polarograms and the tree-structured wavelet transform are reliable techniques for the texture recognition, the major drawback for real time use is the video resolution. A highly detailed frequency spectrum is required for the polarograms and a high resolution of the sample is important to grow accurate trees in the tree-structured wavelet transform. In our application, the video frames would not be able to provide a very high

resolution for the patient's clothes, as the videos themselves are of low resolution frames and the patient's clothes are likely to constitute a very small part of the video image. Other things, like creases in the clothes, can also lead to frequency components that may not be present in the original texture sample. Reconstruction of the original pattern with a small sample of the clothing may be a possibility, but it is a complex image mosaicing problem and is too complicated to be used for real time tracking problems. Thus, the texture of the clothing of the patient as a feature does not seem to be a suitable one for the purpose of feature-based tracking.

2.3 Ellipse detection in combination with skin filters

Among the natural features that can be used for tracking a human being are the body shapes. Since the aim of this tracking problem is to locate the patient in the video frames, shapes of the body parts may be a good choice as a feature to be detected. The patient is expected to either move about inside the room or be lying in the bed. While in bed, much of the body is likely to be covered by blankets. However, the patient's head is likely to be in the field of the view of the camera much of the time and thus, can be a very useful feature for the tracking purpose. In any case if the patient is lying in the bed then there is no need for tracking. The intention in this case is to monitor the potential movements of the patient around the room.

The human head being of a spherical shape can be expected to be of an elliptical shape from any field of view in two dimensions. The shape and size of this ellipse will vary depending upon the position of the patient and his distance from the camera. Therefore, a method to detect elliptical shapes in the video frames may be used to locate the patient

within the image. In addition to the human head, the video images may contain other elliptical shapes. Thus, it will be necessary to distinguish the human head from other elliptical shapes. The most obvious distinction is the presence of the human skin within the detected ellipse corresponding to the human head. This can be achieved with the help of a skin color filter. The skin color filter should be able to extract all the possible skin 3colors in the frame and if a significant amount of the skin color is found inside the detected ellipse, it can be assumed to be the human head. In certain conditions, other body parts may also be detected as elliptical shapes due to the presence of some other objects in their vicinity or due to excessive noise in the captured image. In these cases, the ellipse with the most skin color can be assumed to be the head, since the head is expected to contain the most skin color compared to other body parts, which are usually covered.

Thus, the detection method comprising of the elliptical shape containing the skin color within it can be done in two ways. One method can be to identify the ellipses within the frame and then, the ellipse with most skin color within it can be assumed to be the location of the patient's head. Another approach maybe to filter the frame with the skin color filter to isolate the region of interest and then find the elliptical shapes in these skin color regions. The best ellipse found around the skin regions can be assumed to be the head.

The development of a robust skin color filter is a very important step in the success of this detection method. The details of the skin filter are given in Chapter 3, while the present discussion is concerned with the details of the detection of the elliptical shape and the face detection algorithm using a combination of the ellipse detection algorithm and

the skin filter. The ellipse detection algorithm works on the pixels defining the shape to find the best fitting ellipses. The pixels are the data points and can be obtained by performing edge detection on the original image. Thus, the edge detection is the first step in ellipse detection and is discussed next.

2.3.1 Edge detection

In digital images, edges are strong contrast of intensity. They are a set of connected pixels that lie on the boundary between two regions. Extracting the edge information in an image helps in recovering the shapes of the objects, in addition to removing the excess information. The edge detection works on an image to yield the feature boundaries which can be used for ellipse fitting.

In one dimension, an edge can be defined as a sharp transition from one intensity to another as shown in Figure 2.7(a). However, in practice, there is a blurring on a soft or gradual transition from one intensity to another. This slow change in intensity can be represented by a ramp as shown in Figure 2.7(b).

The slope of this ramp is proportional to the change in intensity at the edge. The points constituting the ramp along with the neighbouring regions of zero slope, denote the intensity values at those pixels. The graphical representation of the intensity values is called the intensity profile.

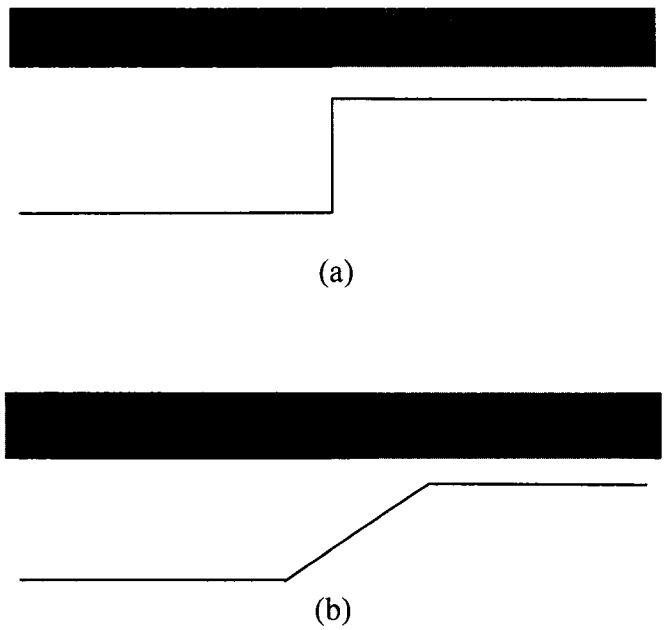


Figure 2.7. (a) A sharp edge and its intensity profile. (b) A blurred edge having a ramp-like intensity profile.

The thickness of an edge can be determined by the length of the ramp. If a pixel lies within the region of the ramp, then it can be assumed to be a part of the edge. The first order derivative of the ramp can be used to determine the presence of a point on an edge. Figures 2.8(b) and 2.8(c) respectively show the first and second order derivatives of the intensity profile shown in Figure 2.8(a). The first order derivative is zero when there is no change in intensity, and is a positive constant for the length of the ramp.

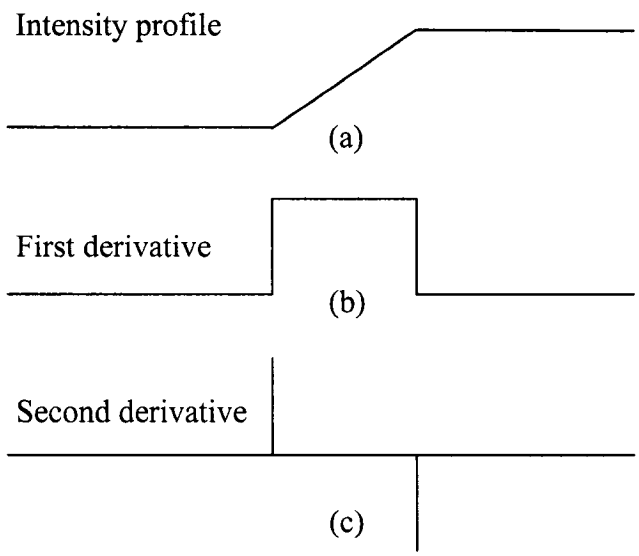


Figure 2.8. (a) Intensity profile as in Figure 2.7(b). (b) First order derivative of the intensity profile shown in (a). (c) Second order derivative of the intensity profile shown in (a).

As shown in Figure 2.8(c), the second derivative results in spikes at positive and negative transitions of the first derivative. The positive transition occurs when the intensity profile goes from the darker to the lighter intensity region and vice versa for the negative transition. Thus, we can determine whether the edge point lies closer to the darker intensity region or the lighter one.

The benchmark edge detection techniques, such as the Canny edge detection [18], Sobel edge detection [19] and Deriche operator [20], are all based on this basic concept of edge detection. Methods like Sobel edge detection are simple to implement, however, the Canny and Deriche edge detection methods are computationally more expensive and involve parameters that can be adjusted depending on the types of shapes to be detected.

The purpose of edge detection is to obtain data points corresponding to the edge contours of the shapes in the image. These can be subsequently used to detect the elliptical shapes. A fast and simple technique that can detect enough data points for ellipse modelling is needed. The Canny edge detection technique being complex and computationally expensive is not suitable for real time applications. The Sobel edge detection method being less complex and faster in processing is better suited for real time applications. The Sobel edge detection technique is therefore chosen over the Canny edge detection method for the considered application.

2.3.1.1 Sobel Edge Detection

The theoretical concept of edges in one dimension can be extended to two dimensions as well. To detect the edges, there needs to be an accurate approximation for the derivative in two dimensions. The Sobel edge detector performs this function by using two convolution masks that slide through the image to calculate the gradient at each point. The two masks are 3x3 in size and work along the x and y directions to calculate the gradient in the x and y directions respectively, i.e., along the rows and the columns. The two masks together are known as Sobel operators. Figure 2.9 shows the Sobel operators.

$G_x =$

-1	0	1
-2	0	2
-1	0	1

$G_y =$

1	2	1
0	0	0
-1	-2	-1

Figure 2.9. Sobel Operators

At any point, the magnitude of the gradient is calculated as

$$G = |G_x| + |G_y|$$

Thus, the gradient in the x direction is calculated by placing the center of G_x on any pixel and the output is calculated by adding the pixels in the mask weighed by the content of the mask as defined in G_x. The gradient in the y direction can be calculated in a similar way and the addition of these two gradients yields the two dimensional gradient of the image. Thus, the edges in both the directions are detected.

Figure 2.10 shows a sample image and its edge-detected output using the Sobel operator.

Figure 2.10. A sample image and its edge detected output using Sobel Operators.

The edge detected image is a binary image containing the data points which define the outline of the shapes present in the frame. The data points of these contours can now be used for the ellipse detection using the least mean square fitting of ellipses.

2.3.2 Least mean square fitting of ellipses

The research in the area of ellipse detection has been mainly focused on Hough-transform based techniques [21, 22] and the least mean square fitting of ellipses. The Hough-transform based techniques need a large amount of data to make a reliable detection. On the other hand, least mean square fitting methods can model ellipses even with a fewer

data points, though the accuracy of the modeling may deteriorate as the number of points is reduced. The data points can be obtained from a video frame by one of the edge detection processes. As was seen from the discussion in Section 2.3.1, the edges can be detected reliably using the Sobel edge detection method. The output of the edge detector includes the data points for all the major shapes in the frame, including the elliptical shaped head.

The ellipse detection on data points is performed using the d*irect least square fitting of ellipses* (DLSFE) algorithm introduced by Fitzgibbon et al [23]. The algorithm while being computationally efficient is also highly robust with respect to noise. The idea behind the least-mean square fitting is to minimize the distance between the data points and the ellipse. The algorithm tries to achieve this by controlling the parameters in the general equation of an ellipse.

The notations and equations below are as described in the original paper on the direct least mean square fitting of ellipses [23]. The general equation of a second order curve can be represented by a second order polynomial,

$$F(k, z) = k.z = ax^2 + bxy + cy^2 + dx + ey + f = 0 \tag{2.1}$$

where

$$k = [a\ b\ c\ d\ e\ f]^T, \tag{2.2}$$

and

$$z = [x^2\ xy\ y^2\ x\ y\ 1]^T \tag{2.3}$$

The geometric distance of any point from the curve is defined as the minimum distance of that point from the curve. If the geometric distance of any point (x_i, y_i) to the curve $F(k, z) = 0$ be given as d_i, then

$$Sum\ of\ squared\ distances = \sum_{i=1}^{n} d_i^2 \tag{2.4}$$

where n is the total number of available data points. Minimization of (2.4) leads to the least square fitting of a curve.

For a second order curve to represent an ellipse, the quadratic constraint given by $b^2 - 4ac < 0$ needs to be satisfied. Bookstein [25] showed that minimization of (2.4) subject to a quadratic constraint can be performed by considering the generalized eigenvalue system and using the lagrange multiplier λ as,

$$D^T Dk = \lambda Ck \tag{2.5}$$

where, $$D=[X.*X\ X.*Y\ Y.*Y\ X\ Y\ 1] \tag{2.6}$$

and '.*' denotes element by element multiplication. The vectors X, Y are defined as the datapoints as,

$$X=[x_1 x_2 x_3 \ldots\ x_n]^T \text{ and } Y=[y_1 y_2 y_3 \ldots\ y_n]^T \tag{2.7}$$

D is called the design matrix, C is a constraint matrix and k is as defined in (2.2).

The inequality constraint, $b^2 - 4ac < 0$ is difficult to solve as the solution is not guaranteed [26]. Most of the least square fitting algorithms for ellipses and two

dimensional curves in general, differ only in the constraint applied to the parameters. The Fitzgibbon-Pilu-Fisher algorithm [23] reduces it to the equality constraint,

$$4ac - b^2 = 1 \tag{2.8}$$

In matrix form, this constraint can be stated as,

$$k^T C k = 1 \tag{2.9}$$

where

$$C = \begin{bmatrix} 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Thus, from (2.5), (2.6) and (2.8), the least mean square problem reduces to solving the system $D^T D k = \lambda C k$ with the constraint $k^T C k = 1$

Equations (2.5) and (2.9) thus form a system of simultaneous equations given by,

$$Sk = \lambda C k \tag{2.10}$$

$$k^T C k = 1 \tag{2.11}$$

where

$$S = D^T D$$

is called the scatter matrix.

Let (λ_i, μ_i) be the solution to (2.10), then for any j, $(\lambda_i, j\mu_i)$ should also satisfy (2.10). Thus (2.11) yields,

$$j^2 \mu_i^T C \mu_i = 1,$$

and using (2.10), gives,

$$j = \sqrt{\frac{1}{\mu_i^T C \mu_i}} = \sqrt{\frac{\lambda i}{\mu_i^T S \mu_i}} \tag{2.12}$$

With $k = j\mu_i$, the system of equations given by (2.10) and (2.11) can be solved. The solution yields six eigenvalue-eigenvector pairs (λ_i, μ_i). Only that pair for which the term under square root in (2.12) is positive yields the correct solution. S is positive definite, thus the value under square root in (2.12) depends upon λ_i. In [27], it was shown that the minimization of $||Dk||^2$ with the constraint $4ac - b^2 = 1$ yields only one solution. In view of the type of constraint that has been applied, the solution is an ellipse. Therefore, there is only a single value of λ_i that is the solution and provides the parameters for the fitted ellipse.

Figure 2.11 shows three different examples of direct least square fitting of ellipses on three different data sets. The Sobel edge detection process on the original image yields the contours in the image. The subset of contours defining the human head within each

image has been extracted manually by selecting the small region containing the head from the edge-detected image. Figure 2.11 (a) has been obtained from an image where the facial features of the face are also visible. The facial features produce more contours in the edge detection stage and therefore, the number of data points available for ellipse detection algorithm is large. Figure 2.11 (b) shows a side view of the human face. The contour obtained from the edge detection stage contains fewer points and no facial features can be seen. The amount of noise or spurious edges is also significantly less in this image. Figure 2.11 (c) shows the contours of the back view of the head. In each of these images, the contours marked in red are the ellipses detected by the DLSFE.

(a) (b) (c)

Figure 2.11. (a), (b) and (c) Ellipse detection on data points obtained by edge detection process on human head images extracted from frames having epileptic patients in field of view.

As seen from Figure 2.11, the ellipses are detected using all the data points available to the ellipse detection algorithm. This means that all the points would contribute to the ellipse parameters, whether or not they actually form a part of the elliptical shape. With a large number of noise pixels associated with facial features in Figure 2.11 (a), the fitted ellipse is also seen to be concentrated inside the head. With almost negligible noise in Figure 2.11(b), the ellipse is still not perfectly fitting the head due to some pixels associated with the shoulder region at the bottom, which tend to pull the ellipse towards

these points. Similar effect of the concentration of the pixels can be seen in Figure 2.11(c). This indicates that the least mean square ellipse fitting depends mainly on the concentration of the pixel population. The elliptical shapes, though visible, are not detected perfectly due to the presence of other pixels that contribute equally towards minimizing the mean square error.

2.3.3 Detecting human faces using ellipse detection and skin filters

The face detection algorithm being developed is based on the detection of elliptical shapes in the image. The human faces can be detected in the images using the ellipse detection and skin color filters in two ways as shown in Figures 2.12(a) and 2.12(b). In the first approach, as shown in Figure 2.12(a), edge detection is first performed and this provides the necessary data points for the ellipse modeling. The second stage is to detect edges, which can be used for detecting elliptical shapes. The next stage involves identifying the skin color. The ellipse with the maximum skin color present inside it is assumed to be the human face. Thus, the first method searches for the best fitted ellipses in the image and then looks for the amount of skin color within them. In the second approach, as shown by Figure 2.12(b), the skin color filtering is performed first in order to locate all the possible regions, where human face could be present. Ellipse fitting is then performed in the detected skin regions. This indicates the presence of a human face.

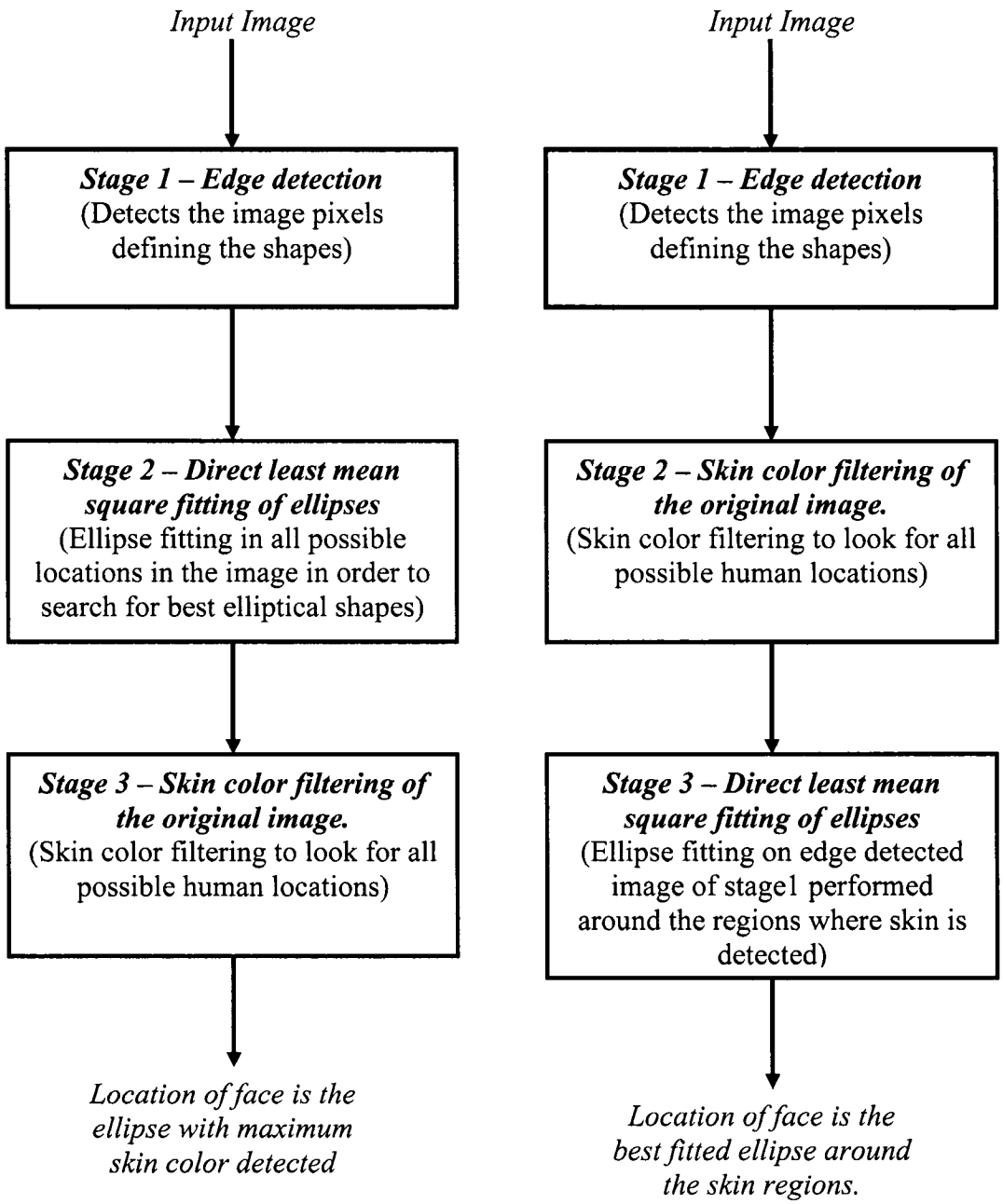


Figure 2.12. (a) Face detection with ellipse fitting followed by skin color detection. (b) Face detection with skin filtering followed by ellipse fitting.

Once an elliptical fit is found for the edges, the quality of the fit can be measured by calculating the mean square error of the fit. For any data point, the minimum distance of the point from the fitted ellipse is the error in the fit for that point. Thus, if d is the distance between any data point and the fitted ellipse, then the mean square error for n data points can be calculated as

$$Mean\ square\ error = \frac{1}{n}\sum_{i=1}^{n} d_i^2$$

This mean square error is the criteria for defining the quality of fit.

In the first approach, to detect the shapes that best represent the ellipse in the image, step by step search of the ellipses is performed in the complete image. Ellipses can be searched in small windows, whose size is large enough to accommodate the human head. These small windows would run through the whole image in small steps starting from the first row and first column to the last column in first row, and then stepping onto the next row and so on. The step size can be as small as one pixel or as big as the size of the window, depending upon the level of accuracy and the speed of search desired. The ellipses with small mean square error can be candidates as possible locations for the human head. The human head is expected to contain traces of the skin, and therefore, a top to bottom search in the candidate ellipses for the skin color is done. From the candidate ellipses, the one with sufficient skin color within it can be assumed to be the human head.

The second approach is to locate the skin areas first. The windows large enough to accommodate human head can be selected around the centroids of the detected skin areas. The ellipse with the minimum mean square error of fit is expected to correspond to the

human head. This approach is more practical and computationally less expensive, as it does not involve processing each and every possible location for the ellipses. The regions of interest where the human head may be present are reduced significantly by first performing the skin filtering. Therefore, this approach is more suitable for detecting human faces than the first approach, as the sliding window is no longer required and not every possible region needs to be examined for the ellipses. Therefore, we use this second approach as the algorithm to detect the human head.

2.3.4 Performance Evaluation

The performace evaluation of the algorithm introduced above needs to be done in order to determine its suitability for locating human faces in video images. The evaluation is a simple process with a hit or miss criteria for the target, namely, the face. For each video frame under consideration if the subject head is not correctly identified, the algorithm is considered to have failed, i.e., missed the detection of the human face. Conversely, if the human face is correctly identified, then the algorithm is deemed to have been successful. Since the evaluation of success or failure must be done manually one frame at a time, it is not practical to make a decision on each and every frame in the video. The application for which this algorithm has been developed is related to the tracking of patients in a hospital environment, where the movements of the patients are expected to be slow. Therefore, we can assume that there will not be any significant change in consecutive video frames, and the processing of one frame per second may be sufficient for this application. Thus, in order to keep the evaluation process simple and consistent with the application at hand, we will process every 25^{th} frame in the video. The output of the detection of the human

head is shown by a crosshair on the image at the location of the head. If the location of crosshair is on the human face in the frame, then the detection is considered to be successful; otherwise, it is deemed to have failed. Table 2.1 summarizes the evaluation results for the human face detection algorithm tested on video ellipse.avi [DVDROM:\chapter2\ellipse.avi]. The video under test has been taken under good lighting conditions with slow movements of the patient. Attempts have been made to capture almost all the possible views of the face in order to make the video data robust.

TABLE 2.1

RESULTS OF THE TRACKING PROCESS FOR EACH FRAME BEING PROCESSED IN THE VIDEO ELLIPSE.AVI.

Frame Number	Tracking successful?
1	Yes
25	Yes
50	No
75	Yes
100	No
125	Yes
150	No
175	No
200	No
225	Yes
250	Yes
275	Yes
300	Yes
325	No
350	No
375	No

400	No
425	Yes
450	Yes
475	Yes
500	Yes
525	No
550	No
575	No
600	No
625	Yes
650	Yes

From the data in Table 2.1,

Total number of frames processed = 27

Total number of frames with successful tracking = 14

Thus, the percentage success = 51.85%

The results indicate that a tracking success of around 52% is achieved, and this cannot be considered satisfactory since for almost half the time the algorithm fails to detect the location of the face. The reason for this low success rate lies in the functioning of the ellipse detection algorithm. The ellipse detection algorithm works on the least mean square error criteria. It, therefore, finds an ellipse even if very little data is available. Even if there are only a few pixels cluttered in a small area, the method will always fit an ellipse to this data. For such kinds of data, the data points lie close together, their distance from the estimated ellipse is not very large resulting in the overall mean square error to be very small Additionally, the presence of noise after the edge detection stage causes

distortion of the fitted ellipse. The degree of distortion depends upon the number and location of the noisy data points. This distortion causes an increase in the mean square error even for valid elliptical shapes. The combination of these two factors can cause errors in the detection of the elliptical shapes, when the mean square error is the criterion is used to find the best fit.

It is not possible to filter out the data points corresponding to the contours of facial features and the noise in the vicinity of the elliptical contour of the head. These points contribute equally in filtering the ellipse, resulting in a poor fit. Thus, it becomes very difficult to improve the detection performance. These problems involved in the detection of the elliptical shape of the head make it difficult to use it as a feature for the patient tracking problem at hand.

2.4 Summary

In this chapter we have described the algorithms that were developed and experimented with in the initial stages of the research performed in solving the problem of tracking the patient. The ideas and methods seemed to be feasible until they were analyzed or tested in the context of the final problem at hand. Though these methods were not successful for tracking the patient, they cannot be regarded as complete failures. The experiments done in this phase have provided a better understanding of the limitations of the project and also provided ideas to investigate other possibilities.

The concept of utilizing the texture was investigated for possible use as a feature, but due to limitations, as explained in this chapter, could not be implemented. The ellipse detection in combination with the use of the skin filters did not yield the desired

performance, but the skin filter developed in the process was helpful in providing some information about the location of the patient. This skin filter is discussed in detail in Chapter 3 and is the backbone of the tracking algorithm to be discussed in Chapter 5. It not only results in very accurate skin color detection, but also reduces the computational burden for the second stage of the tracking algorithm that detects an additional feature from the patient. The use of the skin filter has allowed us to define the regions of interest in the image that may contain the human face. This information can be used in other detection methods.

Therefore, the methods discussed in this chapter are very crucial not only in the development of the final tracking algorithm, but also in the understanding of the limitations of the tracking problem at hand.

Chapter 3

Skin Color Detection in Digital Images

3.1 Introduction

The skin color is an obvious feature to be used for human target detection. The fact that a patient is the primary target makes the human skin color an obvious feature to be exploited. The human skin color if determined accurately will give possible locations of the human face. The combination of the human skin color location along with some other features can help in determining the exact location of the patient's face. Therefore, detecting the human skin color is the first step in the tracking process, and this can be followed by some other feature detection step as a second stage. This chapter discusses the existing algorithms for the skin color detection, their limitations and scope for improvements and finally, introduces a new algorithm for detecting the skin color accurately.

3.2 Background

Color is an important feature of the human face and is a basis of various computer vision algorithms. Using the skin-color as a feature for tracking a face has several advantages. Algorithms exploiting the color as a feature are usually faster in terms of the processing time than the ones relying on other physical features. Under static

lighting conditions, color as a feature becomes orientation invariant. This is helpful for algorithms based on motion estimation as only a translation model is needed for motion estimation. The color information is also robust in terms of orientation and scaling.

Identification of the human skin using the color as a feature also needs to overcome certain challenges. The color representation of a face obtained by a camera is influenced by factors like the object movement and illumination intensity. Different cameras produce significantly different color values even for the same person under the same lighting conditions, and the skin color differs from person to person. In order to use the color as a feature for face identification, these problems need to be solved.

There are several algorithms that address the issue of skin detection. David O'Mara [28] has tried to build skin pixel classifier to identify pixels as either skin or non-skin. The skin classifiers have been built from datasets containing different shades of skin colors across different ethnic groups. However, the classifier has been built only in a specific color space and the final performance of the classifier suffers from the appearance of the "blue skin". Jay Kapur's model [29] based on Fleck and Forsyth algorithm [30], converts the basic color information of the image to a space where the texture amplitude, hue and saturation values of the image are computed. This conversion is based on a variant of the formula given by Fleck and Forsyth [30]. In Kapur's model, the filtering is done with respect to the texture, hue and saturation in their respective color planes. The filtered regions are expected to be areas containing the skin.

Though these algorithms claim to detect the skin regions accurately, sufficient performance analysis has not been done for a proper evaluation. It is important to test these algorithms in different conditions and on a varying set of images obtained under

different source/lighting conditions so that their limitations can be well understood. Practical results, as discussed later in this chapter, indicate that these algorithms detect a range of colors that is much wider than the skin colors.

In this thesis, we have tried to develop a robust skin detection algorithm that improves the detection performance. The new algorithm combines the basic complementary features of the O'Mara's classifier [28] and Fleck and Forsyth algorithm [29], as they complement each other. These two algorithms are used as independent stages in the new filter. A third stage filtering has also been introduced to suppress the detection of the non-skin colors which both of the above mentioned algorithms fail to exclude.

This chapter starts with a discussion of the color space models. The Fleck and Forsyth algorithm as well as the O'Mara's skin classifier along with the drawbacks is then presented, leading to the development of the new algorithm. Finally the improved performance of the proposed algorithm as compared to that of the Fleck and Forsyth algorithm as well as that of O'Mara's model is presented.

3.3 Color Models for skin classification

3.3.1 RGB Model for skin color

Red, Green and Blue are the primary colors that can create a complete range of colors when mixed in the appropriate proportions. These are the basic colors and can be easily displayed on color monitors. Figure 3.1 shows the RGB color space as a solid cube in the three dimensional space. Each of the red, green, blue components are represented by real numbers in the range (0, 1). Computers generally display RGB using 24 bits with 8

bits for each of the three colors. Thus, *R*, *G* and *B* components can have 2^8 different values from 0 (the lowest intensity) to 255 (the highest intensity). In the RGB space, the colors are displayed by varying the intensities of the red, green and blue components. Thus, with a 24 bit representation there are 16,777,216 (256 red x 256 green x 256 blue) possible colors.

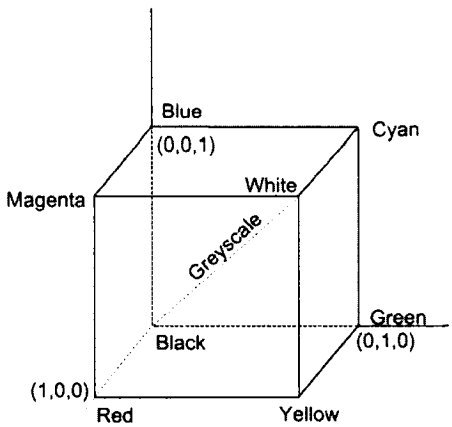


Figure 3.1. The RGB space as a solid cube.

The human skin color can be modeled in the RGB space by specifying the ratio in which the *R*, *G*, *B* components are combined. As there are variations in the skin color across the different ethnic groups, as well as within the same ethnic group, a range of colors needs to be defined in the RGB space to represent this variation. However, RGB space is not a good choice for the skin color identification, as it is very sensitive to the level of the illumination. Slight changes in the illumination may vary the RGB values greatly. This makes the RGB space doomed for failure, as the presence of the shadows

and the different lighting conditions have a severe impact in the RGB model for the skin.

In spite of the drawbacks in the RGB model for the skin color detection algorithms, some researchers have still chosen to use the RGB model. Satoh, Nakamura and Kanade [31] have used the Gaussian model in the RGB space to detect faces using the skin color. Guillamet and Vitria [32] have mapped the RGB color space to a higher dimensional space and used the principal component analysis to detect the skin.

It is to be noted that the effect of the illumination and color variance can be reduced by normalizing the RG space [33]-[37]. Yang, Lu and Waibel [36] have utilized the multivariate Gaussian distribution in normalized RG space to model the skin color. Despite considerable efforts, the RGB model is still considered insufficient for modeling the skin color.

3.3.2 HSB Model

The Hue, Saturation and Brightness (HSB) space is different from the RGB space in the sense that the parameters involved are close to the human perception of the colors. The hue and saturation, called chromaticity, can be directly calculated from the RGB values [38] as

$$h = \arccos\left(\frac{\frac{1}{2}((R-G)+(R-B))}{\sqrt{(R-G)^2+(R-B)(G-B)}} \right) \tag{3.1}$$

$$s = 1 - \frac{3}{R+G+B}\min(R,G,B). \tag{3.2}$$

where R,G and B are the red green and blue components of the colors as defined in previous section.

For any color defined in terms of the R, G and B components, the component with the highest value is taken to be the brightness. Another popular variation of this space is the HSI space, where the intensity I is given as

$$I = \frac{1}{3}(R + B + G) \tag{3.3}$$

.

As shown in Figure 3.2, the HSB space can be seen as a cone in the three dimensional space. Hue is the angle around the cone. The range of hue is defined to be [0, 1]. Saturation is the radius of the base of cone. The pure colors lie along the edge of the cone, whereas the fading colors lie near the center. The range of saturation is [0, 1]. The height of the cone determines the brightness. As the height increases, the brightness increases towards the white.

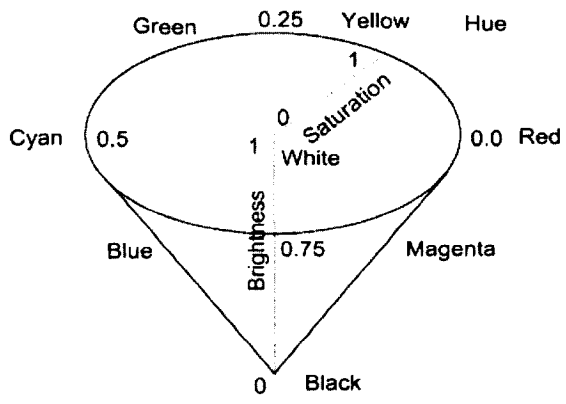


Figure3.2. HSB model as a cone in the three dimensional space

The hue and saturation are very popular parameters for defining the skin color, since they decouple the color and the intensity [39]-[41]. There are numerous skin color detection algorithms that are based on the hue and saturation models. Zarit, Super and Quek [42], and Terrilon, Martin and Akamatsu [43, 44] have shown that the HSI space has the least overlap between the skin and the non-skin colors.

The hue and saturation become unstable at extremely low and high intensity conditions. Therefore, the skin color models developed in the HSI space discard the extreme values [45]. Gevers and Smeulders [46] discovered that the hue and saturation are invariant to the viewing direction, surface orientation, illumination direction and illumination intensity. In contrast, these properties do not hold true in the RGB color space. Thus, the hue and saturation-based models are better suited for the skin color modeling and a majority of the modern skin identification algorithms are built upon this model.

3.4 Overview of David O'mara and Fleck & Forsyth models.

The David O'Mara and Fleck & Forsyth models for finding the skin in images [28, 30] have been found to be most practical for the skin classification and detection. These two methods form the basis of our skin detection algorithm. As the HSI space is better suited for detecting the skin colors, these two algorithms work in this space.

3.4.1 Fleck & Forsyth Algorithm

The Fleck & Forsyth algorithm, better known as the *naked people skin filter* is modeled keeping in mind the texture and color of the skin [30]. The human skin color is created

by a combination of blood (red) and melanin (yellow, brown). The skin colors lie between these two extreme hues. Saturation increases when the skin color gets deeper due to the addition of melanin and the texture of the skin (explained later in this section) is mainly of low amplitude. Fleck and Forsyth algorithm exploits these properties for skin color detection.

The image under test in the RGB format is converted into log opponent values [47] as

$$I_g = L(G) \tag{3.4}$$

$$R_g = L(R) - L(G) \tag{3.5}$$

$$B_y = L(R) + L(G) - 2\,L(B) \tag{3.6}$$

where

$$L(x) = 105 * \log_{10}(x) \tag{3.7}$$

where I_g (intensity), R_g and B_y are the logarithms of the opponent color space [48] and are therefore called the log opponent values. It is assumed that some cameras have poor R and B resolution, and hence, the green channel is chosen for calculating the intensity (Ig). The factor of 105 has been used to magnify the effect of 'log' and to restrict the values to the range [0, 255]. The log transformation makes the R_g and B_y values illumination independent [49].

The texture amplitude, hue and saturation are calculated from the I_g, R_g, B_y values. To calculate the texture amplitude, median filtering is performed on the intensity image (I_g) which considers each pixel in an image and replaces it with the median value in the

specified neighborhood. The neighborhood search around each pixel is performed in the radius 4**SCALE* [31], where the factor *SCALE* is defined as,

$$SCALE = \frac{height\ of\ the\ image + width\ of\ the\ image}{320} \tag{3.8}$$

The result is then subtracted from the original image. The absolute difference thus obtained is then median filtered with the neighborhood radius as 6**SCALE* [31]. The resulting image is termed as the texture amplitude image. The hue, H, is calculated as

$$H = tan^{-1} \frac{R_g}{B_y}$$

and saturation, S as,

$$S = \sqrt{R_g^{\,2} + B_y^{\,2}}$$

The skin color detection takes place in two stages. In the first stage, all the possible skin regions are identified. The second stage is the refinement stage, where the pixels that may overlap into the range of some other materials are excluded.

The algorithm uses the following two rules for the skin color detection.

Rule 1. Texture amplitude <5

110 < hue < 150

20 < saturation < 60

Rule 2. Texture amplitude < 5

$$130 < \text{hue} < 170$$

$$30 < \text{saturation} < 130.$$

These two rules in combination represent a diagonal region in space from low saturated red towards saturated yellow. The output regions of these filters are thus assumed to be potential skin regions, as the skin is a combination of blood (red) and melanin (yellow).

Refinement is done based on the potential regions of the skin discovered in the first stage. A pixel is taken to be a valid skin color if at least one eighth of the pixels in its circular neighborhood of radius 24**SCALE* pixels are identified as the potential skin by the first stage filtering. The factor *SCALE* can be seen as a normalization factor, since the visible skin areas cannot be above a certain level. The refined image is the final output of the algorithm.

The Fleck and Forsyth algorithm performs well by detecting all the possible skin regions. The algorithm was further modified for better performance by Kapur [29] and was used for the purpose of face detection. However, there is no assessment of the performance under changing illumination conditions and under different light sources. The skin filter based on the Fleck and Forsyth algorithm was applied on our test videos. Figure 3.3 shows three sample frames of the first stage of the method. The images on the left are the original frames and those on the right side represent the outputs of the Fleck and Forsyth algorithm.

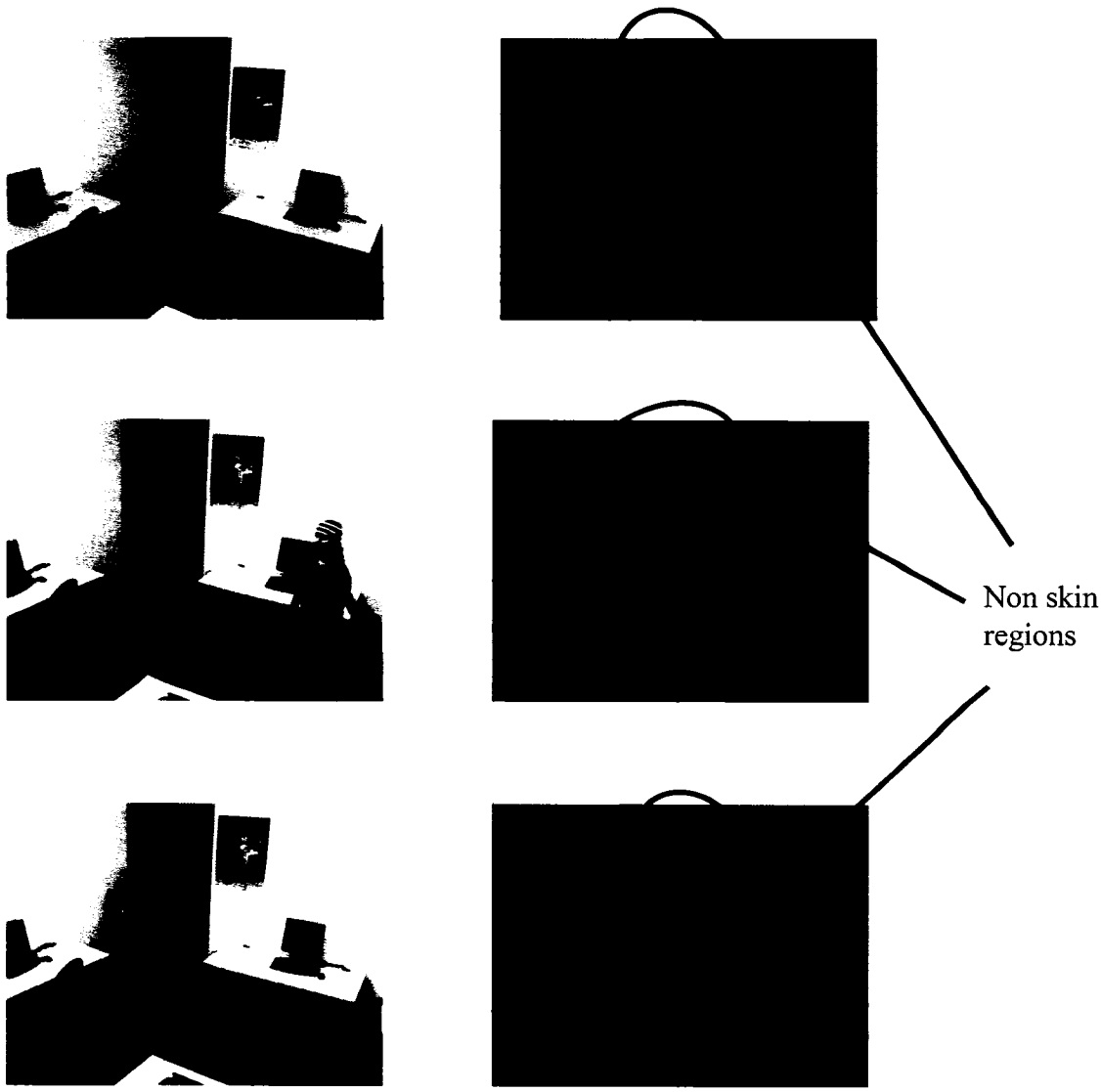


Figure3.3. Frames 100, 200 and 300 from video skinfilter_test1.avi and their corresponding outputs from Fleck and Forsyth algorithm

Examining the outputs of the algorithm (right panel), it is clear that though this model does well to identify all the skin regions, it fails to suppress the non-skin regions as identified by the circled areas in the output. Due to these spurious detections, this algorithm alone is not sufficient for classifying the skin color. Given the high sensitivity

for identifying the skin colors, it can be used as a prefilter to suppress most of the non-skin regions and thus, improve the performance of other skin-color detection algorithms.

3.4.2 David O'Mara skin classifier approach

David O'Mara's algorithm investigates the chromaticity of the skin over different ethnic groups and tries to compare their their skin color distribution. By investigating the skin color over a wide range of images of various ethnic groups, statistical parameters for the skin color distribution are calculated and standardized. The identification of the skin on the test images is performed by comparing the parameters of the various colors in the image against the database of the known parameters for the skin color.

The *Pilot European Image Processing Archive* (PIEPA) [50] is a database of images of faces of people from different ethnic groups. These images are taken under unknown lighting conditions with different cameras. Manual segmentation of the skin areas in these images yields a standard training set for the calculation of the statistical properties of the human skin color. Figure 3.4 (obtained from [28]) shows the overall distribution of the hue and saturation of the segmented skin regions in this training set.

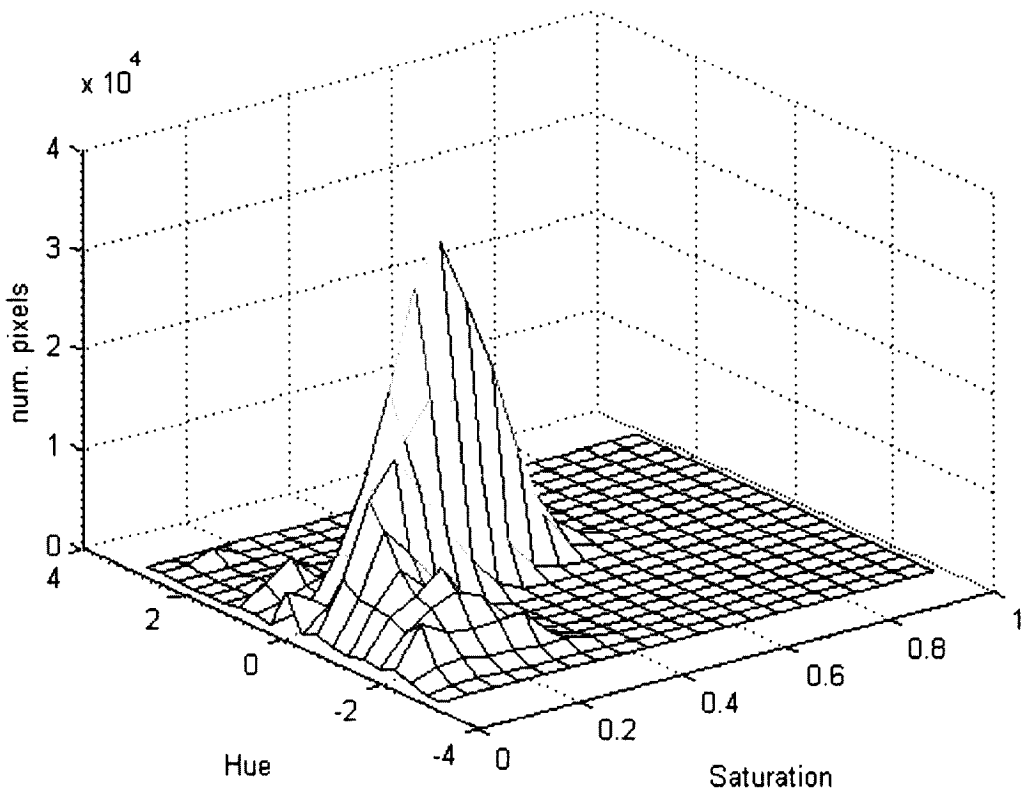


Figure3.4. Distribution of hue and saturation for manually segmented skin regions in the training set [28]

The statistical parameters for this training set are summarized in Tables 3.1 and 3.2 (based on data available in [28]). These values indicate the variation in the distribution of the skin color across the different ethnic groups. The results indicate that mean hue lies around 0.5 and mean value of saturation is around 0.27. These mean values are assumed to be standard values for the human skin color.

TABLE 3.1

STATISTICAL VALUES OF SKIN HUE CALCULATED ON THE TRAINING SET [28]

Hue	Mean	Median	Max	Min
Combined	.4606	.4074	.9984	0
Asian	.314	.0833	.9982	0
Afro-Caribbean	.5624	.7381	.997	0
White	.4271	.0486	.9984	0

TABLE 3.2

STATISTICAL VALUES OF SKIN SATURATION CALCULATED ON TRAINING SET [28]

Saturation	Mean	Median	Max	Min
Combined	.2615	.2532	1	0
Asian	.283	.2785	1	0
Afro-Caribbean	.2237	.2059	1	0
White	.3197	.3226	1	0

The colors can be fully represented in a statistical model using the covariance matrix of chromaticity, mean chromaticity and Mahalanobis threshold [28]. To represent the human skin color, the mean value of chromaticity is averaged over all the ethnic groups. Let the total number of pixels in the training set be n, then the covariance matrix for this training set can be calculated as [28]

$$C = \begin{bmatrix} C_{hh} & C_{hs} \\ C_{hs} & C_{ss} \end{bmatrix}$$

where

$$C_{hh} = \frac{1}{n}\sum_{i=1}^{n} \Delta h_i \Delta h_i$$

$$C_{hs} = \frac{1}{n}\sum_{i=1}^{n} \Delta h_i \Delta s_i$$

$$C_{ss} = \frac{1}{n}\sum_{i=1}^{n} \Delta s_i \Delta s_i$$

and the distances to mean hue and saturation are defined as

$$\Delta h_i = \begin{cases} h_i - \bar{h} & iff\ |(h_i - \bar{h})| < 0.5, \\ h_i - \bar{h} + 1 & iff\ (h_i - \bar{h}) \le -0.5, \\ h_i - \bar{h} - 1 & iff\ (h_i - \bar{h}) \ge 0.5 \end{cases}$$

$$\text{and} \quad \Delta s_i = s_i - \bar{s}$$

where h_i and s_i are the hue and saturation values for any point in the training data, and $\bar{h}$ and $\bar{s}$ are the mean hue and saturation for the training data. The chromaticity matrix and the mean values of chromaticity thus obtained for the training set are considered as standard for the human skin color. The identification of the skin color in an image under test is done by comparing the "distance" of each pixel from the standard values. The distance is calculated using the Mahalanobis distance formula [28]

$$D_{(\vec{p})} = \sqrt{(\vec{p} - \vec{m})^T C^{-1} (\vec{p} - \vec{m})} ,$$

where $\vec{p} = (h,s)$ is the chromaticity of the point which is under test, $\vec{m}$ is the standard mean chromaticity and C^{-1} is the inverse of the standard covariance matrix. If the Mahalanobis distance for the pixel under test lies below the specified threshold, then it is taken to be a skin color pixel.

The O'Mara's model was tested on our test videos. Figure 3.5 shows the sample frames from the video skinfilter_test1.avi. Images on the left are the original frames and the ones on the right are the output of the O'Mara's skin filter.

The output of the O'Mara's model shows that the model works well for identifying the skin. However, the non-skin regions being detected (circled areas in Figure 3.5) suggest significant false detections. There are many colors other than the skin color, which have been identified by this filter. This is undesirable in applications, where high accuracy is needed.

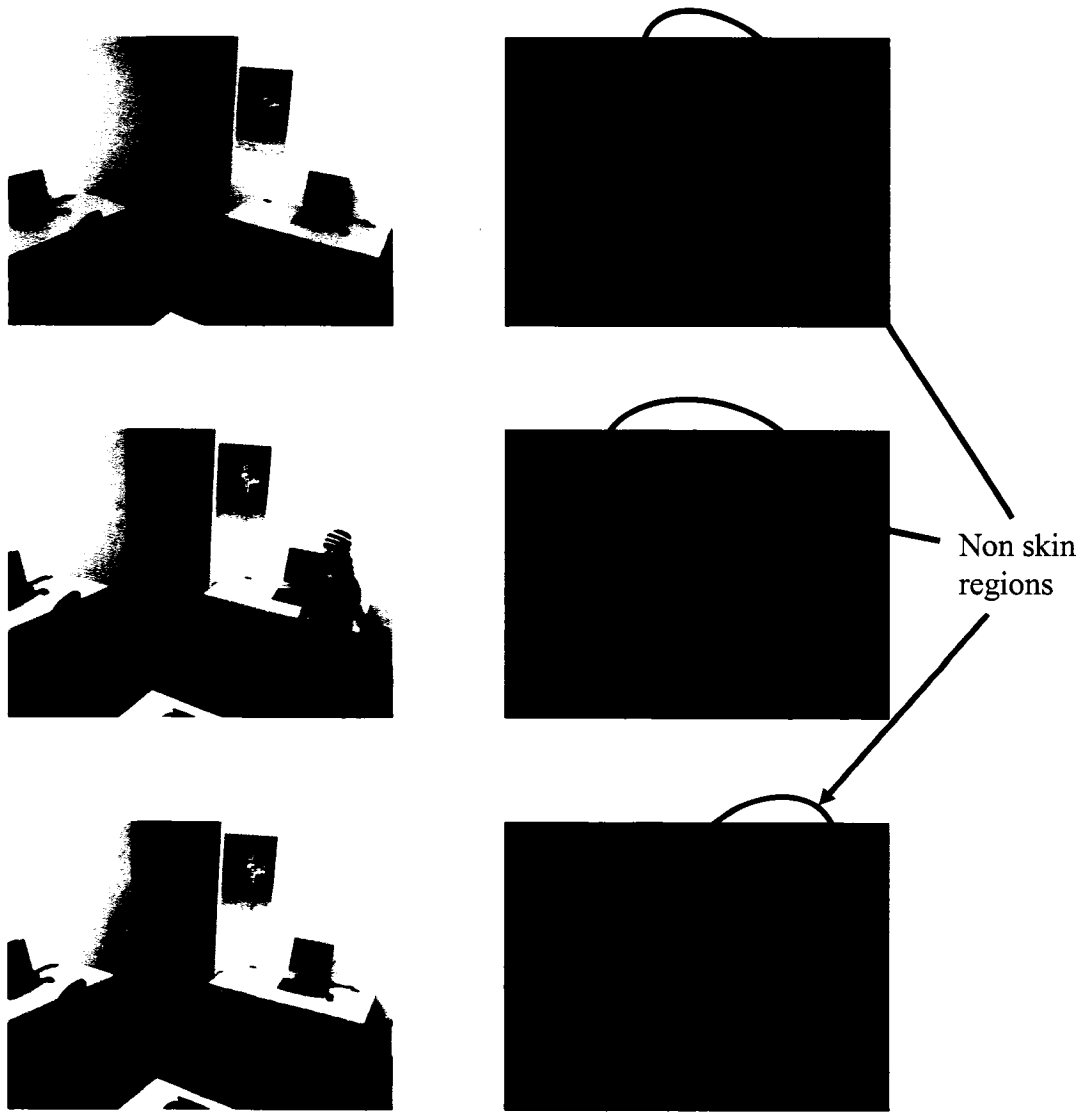


Figure3.5. Frames 100, 200 and 300 from video skinfilter_test1.avi and their corresponding outputs from David O'Mara's algorithm.

3.5 The proposed skin filter

O'Mara model builds classifiers based on huge databases of skin images under different lighting conditions [28]. The Fleck & Forsyth model is based on the theoretical expectation of the skin appearance [29]. Though both these models detect the skin color

well, they also identify various non-skin colors. We combine the two filters and use them effectively to build a very robust skin color detection filter.

The significant problem with O'Mara's classifiers is that it fails to exclude the blue shades, which lie close to the human skin. One possible reason could be that the classifiers are built from real world images taken under different lighting conditions, where the effects of the flash and the camera's perception under low illumination conditions cannot be excluded. Moreover, the classifiers also fail to avoid yellow and orange shades.

Though the Fleck & Forsyth model very precisely limits the skin into the bounds of hue, saturation and texture, it fails to address the problem of illumination. Changing lighting conditions is the biggest challenge for the skin filters and the performance of the Fleck and Forsyth algorithm deteriorates significantly under changing lighting conditions. Furthermore, this model also fails to suppress the various shades of the non-skin color. The frames displayed in Figure 3.3 and the attached results on the test videos [DVDROM:\chapter3\readme.doc] indicate the identification of the non-skin colors by this algorithm.

These two models individually are not sufficient for the purpose of the final tracking algorithm in our application. However, since the drawbacks of these two algorithms in terms of the spurious identifications are different, the two algorithms complement each other well. The basic idea of limiting the hue and saturation into predefined planes in the Fleck and Forsyth algorithm can improve the performance of the O'Mara's model. Therefore, we introduce a new algorithm which combines the O'Mara's and Fleck & Forsyth algorithms as two independent stages of filtering. The Fleck and Forsyth

algorithm tries to identify the skin regions in the original image. The identified regions are used as the input to the O'Mara's algorithm for further refinement in the detection. Our test videos [DVDROM:\chapter3\readme.doc] indicate that the weakness of the O'Mara's model is filtered out in the first stage by the Fleck and Forsyth approach by suppressing the yellow and orange shades. The undesirable gray and brownish shades identified by the Fleck and Forsyth algorithm are suppressed by the O'Mara's model in the second stage. As Fleck and Forsyth algorithm is expected to suppress more non skin colors than David O'Mara's algorithm, therefore by using Fleck and Forsyth algorithm as first stage, better computational efficiency is achieved. This combination of the two algorithms is shown in Figure 3.6

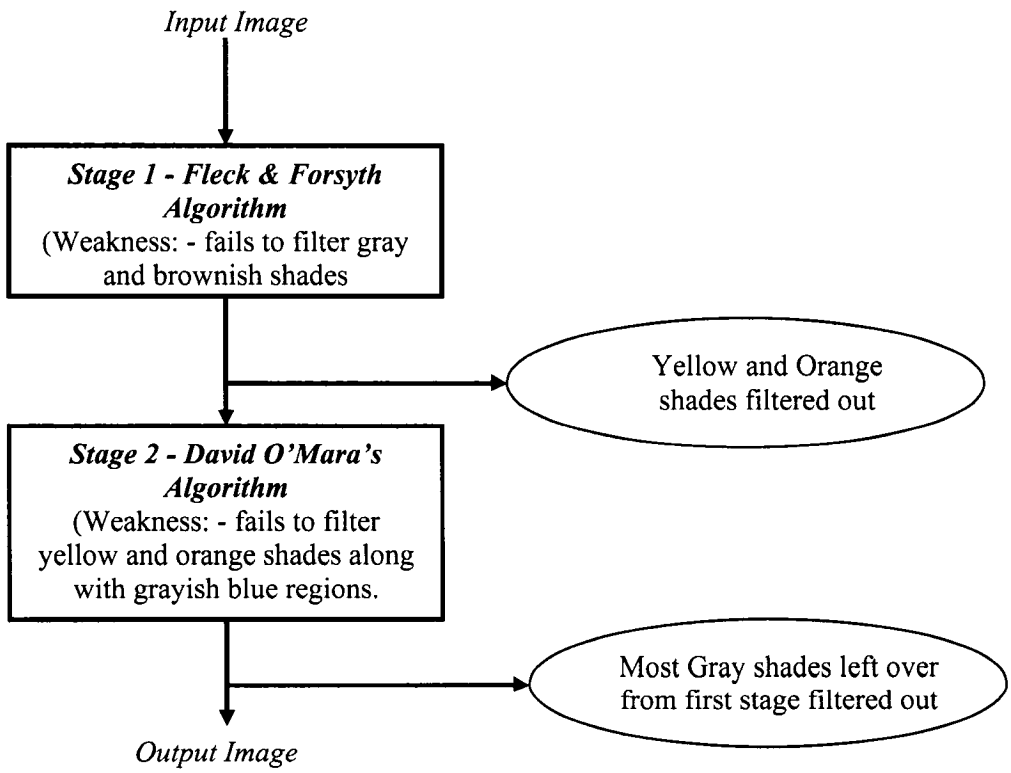


Figure 3.6. A combined skin color filter using the Fleck and Forsyth and O'Mara's algorithms as two independent stages arranged in cascade.

While the combination of the two algorithms gives a near perfect skin color identification, still some non-skin colors find their way through. This small range of non-skin regions that sneak through both the Fleck and Forsyth and David O'Mara's models can be filtered either in the RGB or the HSV space. However, to avoid a possible overlap with the range of the HSV space defined in the Fleck and Forsyth algorithm, filtering in the RGB space is preferred. This RGB space filter is designed by analyzing the range of non-skin colors present in the output of the combined two stages of filtering by the Fleck and Forsyth and David O'Mara's algorithms. The constraints set on the R, G and B values are

$$190 > R > 35,\ 130 > G > 30,\ 105 > B > 15$$

This range of R, G and B values is found to block the non-skin color regions present in the output obtained from the combination filter of Figure 3.6. The improvement in the performance achieved by adding this third stage filter can be seen by the example in Figure 3.7.

(a) (b) (c)

Figure 3.7. (a) Original video frame. (b) Output obtained after two stages of filtering consisting of Fleck and Forsyth and David O'Mara's algorithm. (c) Output obtained by adding a third stage filter.

Figure 3.7 shows a frame from the video skinfilter_evaluation3.avi. The figure on the extreme left shows the original video frame. The figure in the middle shows the output of the combined filtering by the Fleck and Forsyth and David O'Mara's algorithm, and the image on the extreme right is the output achieved by adding the third stage filtering in the RGB space. The middle frame indicates that the two stage filtering by the Fleck and Forsyth and David O'Mara's algorithms fails to suppress the non-skin color pattern on the clothes of the lady. However by adding the third stage filter in the RGB space, the effect of this non-skin color is reduced significantly as seen in the frame on the extreme right. This indicates an improvement in the performance of the combined filter by adding the third stage filter.

The final skin color identification now involves three filters working in cascade. The first stage is the Fleck and Forsyth algorithm in the HSV space, the second stage is the O'Mara's algorithm which works with the skin classifiers, and the third stage consists of a filter that works in the RGB plane. By working in cascade, the shortcomings of each of the stage are overcome in the next stage and therefore, the three stage solution provides a significant improvement in terms of limiting the detection of the non skin regions.

The primary aim of developing a skin filter in our patient tracking problem is to find the potential regions, where the patient might be present. As the number of regions identified by the skin filter increases, more time would be required for the next stage of the tracking process to make a decision about the presence of the target. If the skin filter identifies several colors other than the skin color, it would also increase the probability of false target detections by the next stage of the tracking process. Therefore, the skin filter needs to be highly accurate even if it comes at the expense of low percentage detections

in the skin areas. The primary aim of the filter is to reduce the detection of the non-skin colors while maintaining a relatively successful skin color identification rate.

In Figure 3.8 we compare some examples of the performance of the proposed filter with those of the Fleck and Forsyth and O'Mara's algorithms. The frames chosen for comparison are the same as those in Figure 3.3 and figure 3.5. The output of the three algorithms on the test videos has been added to the DVD [DVDROM:\chapter3\readme.doc].

(a) Fleck & Forsyth Algorithm (b) David O'Mara's Algorithm (c) The Proposed Algorithm

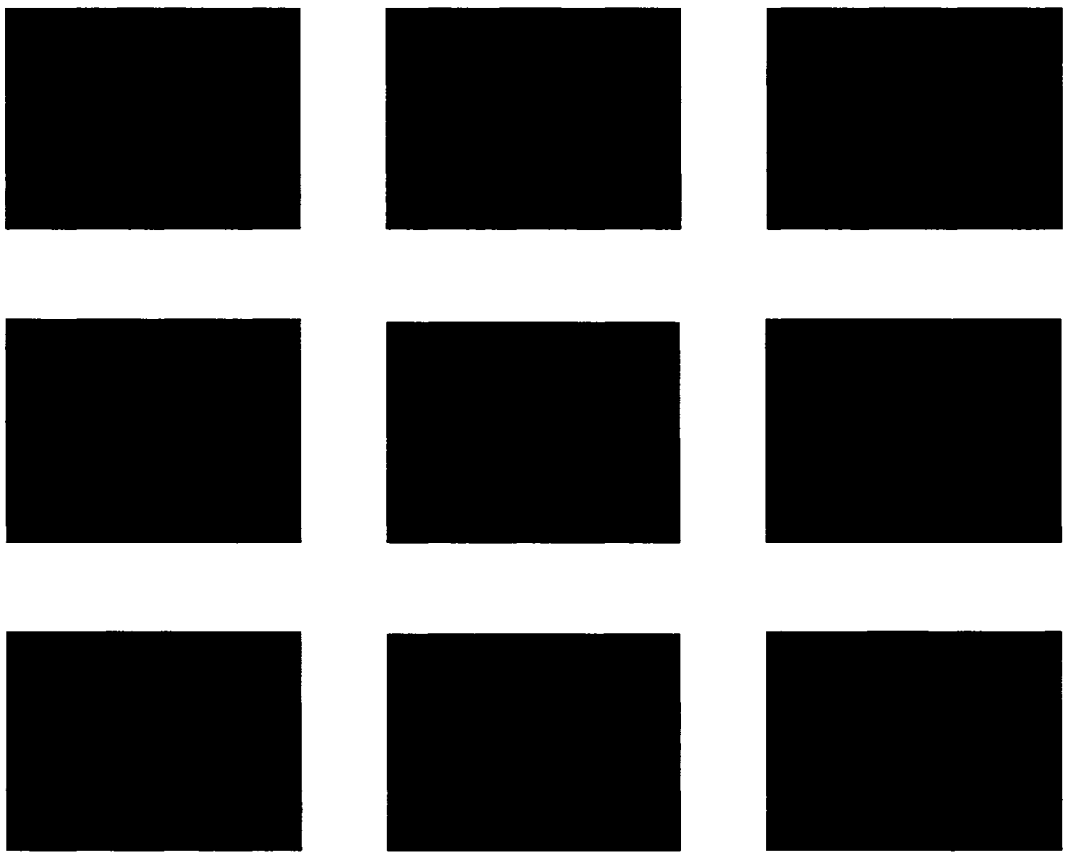


Figure 3.8. Comparison of the outputs of the three algorithms on frames 100, 200 and 300 from video skinfilter_test1.avi.

The results indicate that there is a considerable decrease in the artifacts being detected. However the proposed three stage filtering also results in some decrease in the sensitivity, as not all of the skin areas are detected. But, as discussed previously, the aim of developing a skin filter is to block as many non-skin areas as possible, and the proposed filter gives a significant improvement in the overall performance.

3.6 Evaluation of the proposed algorithm

Visual assessment of the outputs of test videos indicates that performance of our proposed filter is better than the individual models developed by O'Mara or Fleck and Forsyth. The evaluation of the complete test video indicates the improvement achieved in blocking the non-skin colors. But, as our videos have been shot from same camera and more or less under the same physical conditions (except lighting), a random test of the filter for different video sequences from different sources becomes important. We have tried the evaluation on four video clips. The first video clip is one of our test videos and the remaining three are from the Bollywood and Hollywood movies. The videos differ from one another in the lighting conditions, number of people present in the video, ethnicity of the people, and indoor and outdoor conditions.

The evaluation process involves segmenting the skin area manually and comparing it with the output of the skin filter. The manual segmentation of the skin was done based on the visual perception of the human skin color. The segmentation was automated by selecting a closed polygon connecting the points defining the major features (namely the pixels) around the visible skin area. The polygon is formed by a program which inputs the nodes defining the major feature around the desired area and outputs the polygon by

joining all the input nodes. All such polygons in a frame containing the skin color are selected independently and the total area thus selected forms a mask, which is assumed to contain all the skin color regions in the frame. The area of a polygon is defined as the total number of pixels that are included within that polygon. Thus, the total skin area selected manually is equal to the number of pixels within all the selected polygons in a frame. The evaluation is done by comparing the manually-selected skin area based on the visual perception with the skin area detected by the skin color detection algorithm proposed in the previous section.

The performance results are tabulated in terms of the following.

3) Percentage of missed detection: This is defined as the percentage skin area that the skin filter is unable to identify in manually-selected skin areas.

$$\text{Percentage of missed detection} = \frac{\text{Skin areas that the algorithm fails to identify}}{\text{Total skin area}} \text{ x } 100$$

4) Percentage of false detection: This is defined as the percentage of area in the image that the skin filter identifies as skin but is actually not skin.

$$\text{Percentage of false detection} = \frac{\text{Total non - skin areas identified outside the selected region}}{\text{Total area outside the selected region}} \text{ x } 100$$

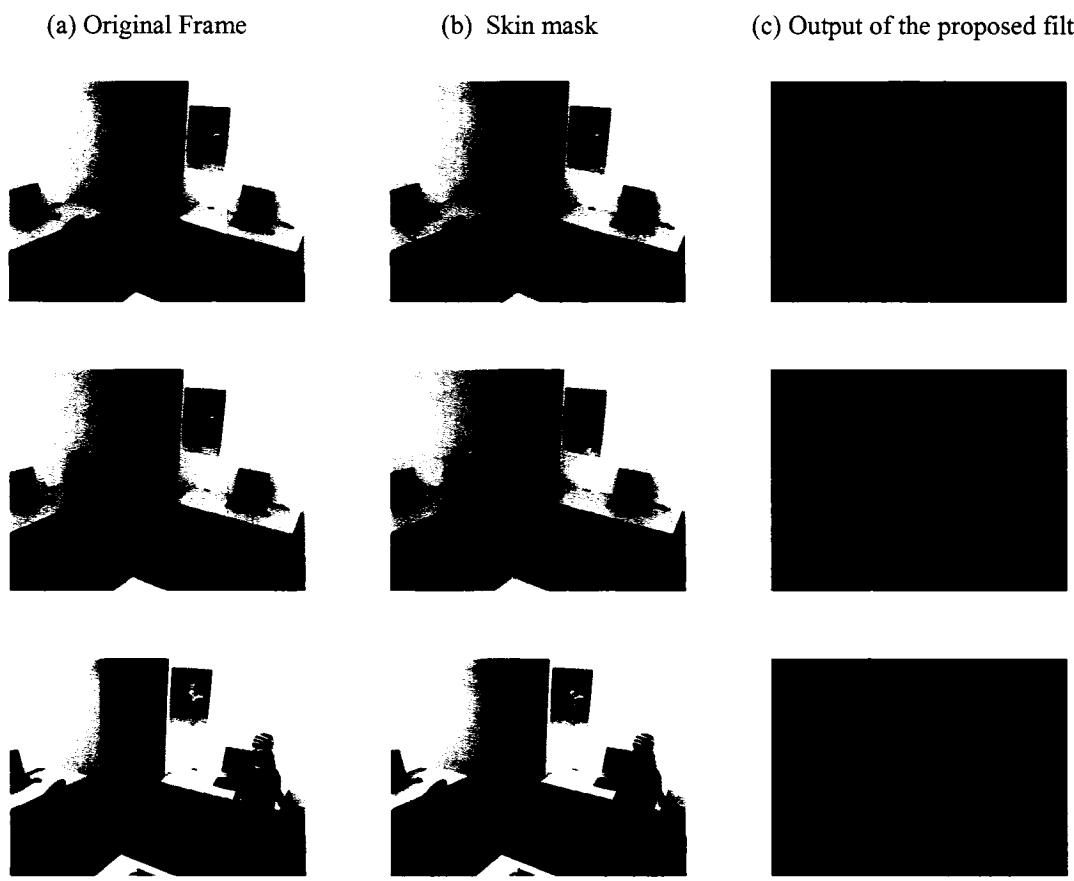
3.6.1 Results on Videos

Outputs of the skin color filter have been obtained for a simulated video and three randomly selected videos from different sources. Some frame sequences have been shown in Figure 3.9 - 3.12. In each figure, the frame on the extreme left is the original

frame, the frame in the middle shows green areas which are the manually selected regions and the frame on extreme right is the output of the skin filter. The performance results have been tabulated for each video under test in Tables 3.3 – 3.6. The test videos and the outputs of the skin filter can be found in the attached DVD [DVDROM:\chapter3\readme.doc].

3.6.1.1 Results for the video skinfilter_evaluation1.avi

Table 3.3 shows the results of video skinfilter_evaluation1.avi and some example frames from this video are highlighted in Figure 3.9.



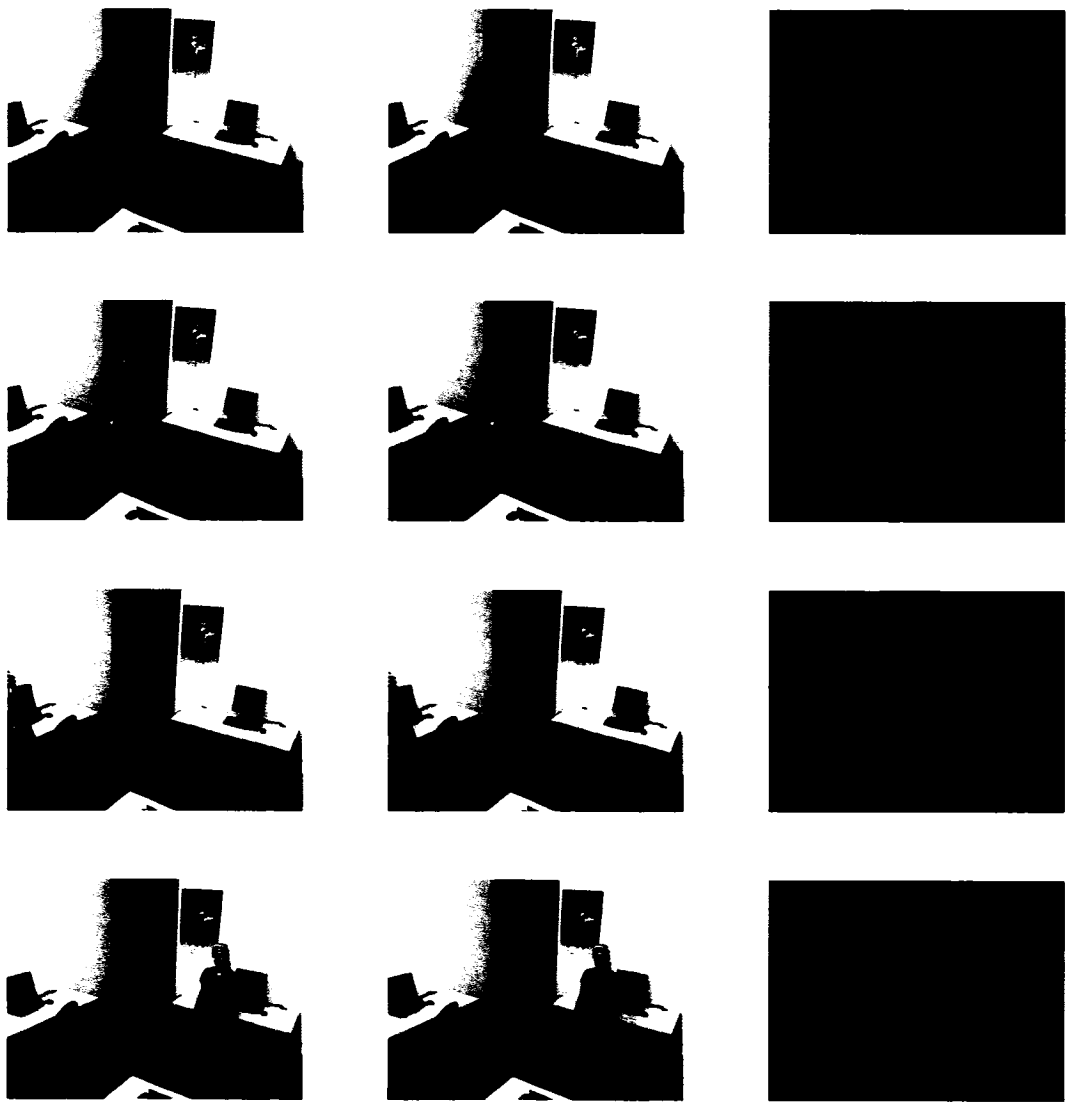


Figure 3.9. Frames 1 to 600 in steps of 100 from video skinfilter_evaluation1.avi and their corresponding selected and detected skin regions.

TABLE 3.3

RESULTS ON THE VIDEO SKINFILTER_EVALUATION1.AVI

Frame number	Percentage false detection			Percentage missed detection		
	Proposed Algorithm	David O'Mara's model	Fleck and Forsyth	Proposed Algorithm	David O'Mara's model	Fleck and Forsyth
1	1.0072	6.4561	4.59	10.0663	3.9445	6.15
25	1.3701	8.2419	3.5485	10.4151	5.9842	5.4495
50	1.1159	5.1533	4.8590	-	-	-
75	1.1159	5.1533	4.8590	-	-	-
100	1.1159	5.1533	4.8590	-	-	-
125	1.2096	9.4238	2.3649	4.7636	4.2506	1.8322
150	.9307	5.5199	3.9580	5.4252	1.5396	3.9956
175	.7327	5.6794	2.3285	51.6631	6.9002	45.1168
200	.5449	5.5111	4.3476	21.9015	.9413	19.7051
225	.5127	5.2184	3.8030	35.6184	6.9965	30.3180
250	.6462	4.4766	6.4290	33.8667	3.2533	30.7200
275	.6458	4.6286	6.9759	52.9664	4.4037	48.5015
300	.4688	5.7282	8.0967	40.7869	5.5738	35.2131
325	.4831	6.7943	9.7965	11.5147	.9274	9.9691
350	.2406	6.4775	8.6738	40.4997	1.2783	37.1877
375	.2848	6.9372	7.1973	39.7823	8.8271	31.9226
400	.1615	5.2008	4.2217	52.0720	4.3561	50.2367
425	1.0924	4.8395	7.1413	31.2960	2.9455	27.8719
450	.9833	5.2467	5.0716	36.7911	2.3513	33.9142
475	.9030	6.0791	3.1514	37.4798	2.9595	35.1040
500	.7598	5.6025	4.5267	16.4520	6.3817	13.4075
525	.8688	7.9707	3.4160	37.3320	3.6054	33.6611
550	.7285	5.8454	4.5280	22.5098	3.3333	19.1765
575	.6299	8.4141	3.4880	53.0950	3.0078	49.7820
600	.4740	7.5488	2.3090	42.8001	4.3898	37.1134
625	.4567	6.9945	2.5758	46.0616	1.3699	43.3790
Average values	.749338	6.1651	4.8890	31.9634	3.8922	28.2490

The results indicate that the proposed algorithm reduces the false detections by about 5% as compared to O'Mara's algorithm, while keeping the missed detections around 32%. With a success rate of 68% and false detections below 1%, the proposed algorithm is highly successful in suppressing the non-skin regions, while maintaining sufficient detection sensitivity for the application under consideration, where it is not necessary to correctly identify the complete skin areas.

3.6.1.2 Results for the video skinfilter_evaluation2.avi

Table 3.4 shows the results on video skinfilter_evaluation2.avi and some frames from this video are displayed in Figure 3.10. This video has been chosen as it has extremely low lighting conditions.

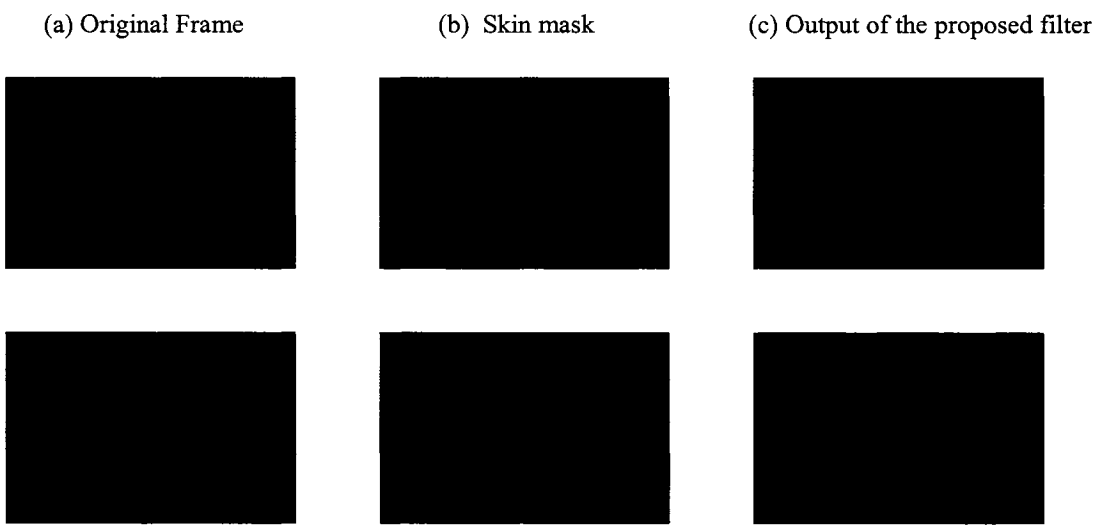


Figure 3.10. Frames 1 and 25 from video skinfilter_evaluation2.avi and their corresponding selected and detected skin regions.

TABLE 3.4

RESULTS ON VIDEO SKINFILTER_EVALUATION2.AVI

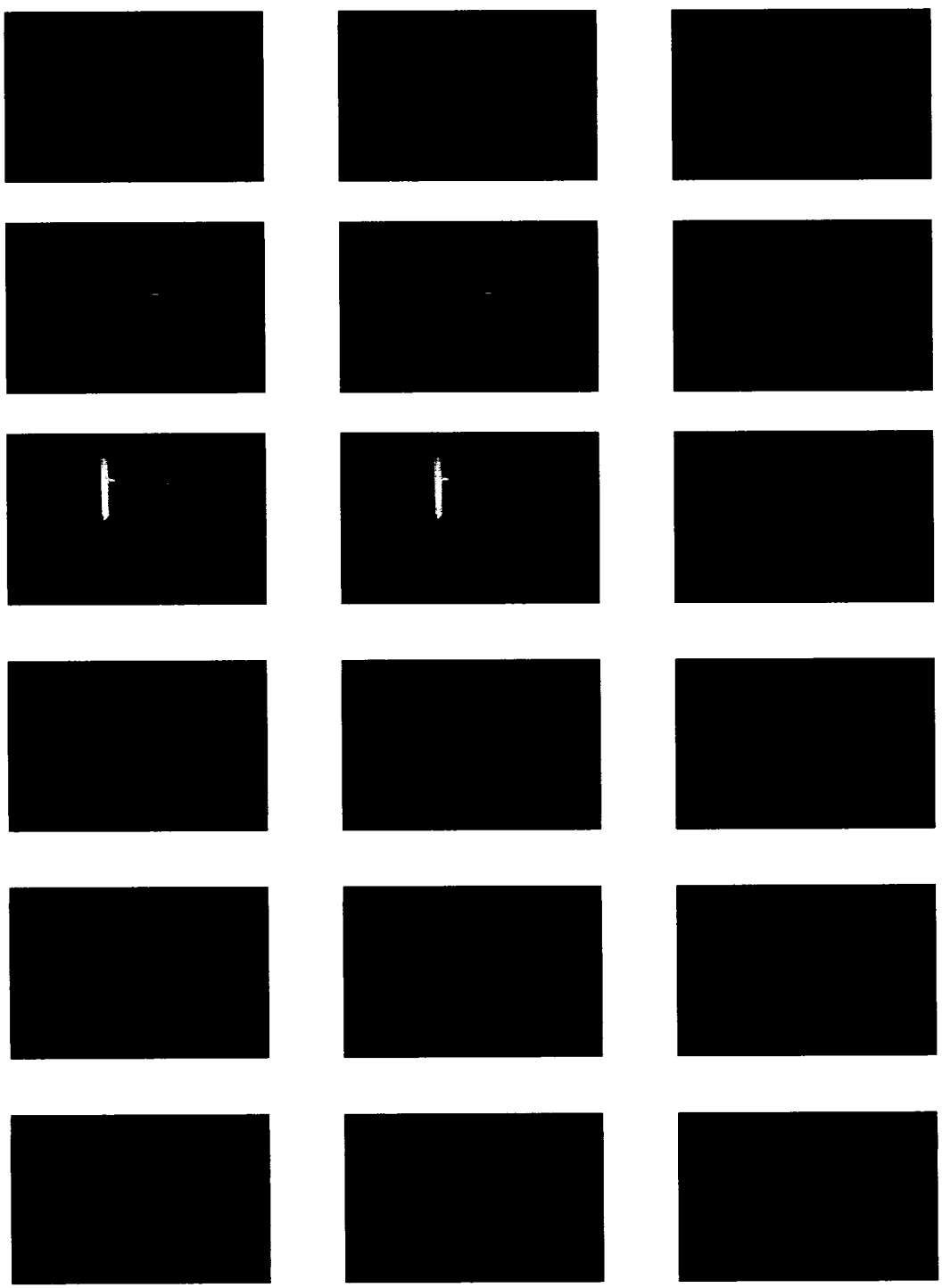
Frame number	Percentage false detection			Percentage missed detection		
	Proposed Algorithm	David O'Mara's model	Fleck and Forsyth	Proposed Algorithm	David O'Mara's model	Fleck and Forsyth
1	.5133	36.3168	.9525	39.1568	15.2669	12.7301
25	.8385	38.4907	1.2477	32.9551	15.4200	13.6648
50	.6108	36.1837	1.7242	41.8719	16.7080	15.2909
70	.0744	23.3397	3.2888	42.7982	21.6952	15.2130
Average values	.50925	33.5827	1.8033	39.1955	17.2725	14.2247

The average undetected area in the selected region is quite high for this video. This is due to the extremely poor lighting conditions. Still the success rate of detection is around 61%, and the percentage false detection is still extremely low. Thus, the proposed filter has been able to prevent false detections better than the other algorithms, and this has been the primary aim of the proposed algorithm.

3.6.1.3 Results for the video skinfilter_evaluation3.avi

Results from video skinfilter_evaluation3.avi are shown in Figure 3.11 and summarized in Table 3.5. This video has changing lighting conditions and multiple people are involved. Further, the man in the video is wearing a shirt whose color resembles the skin color.

(a) Original Frame (b) Skin mask (c) Output of the proposed filter



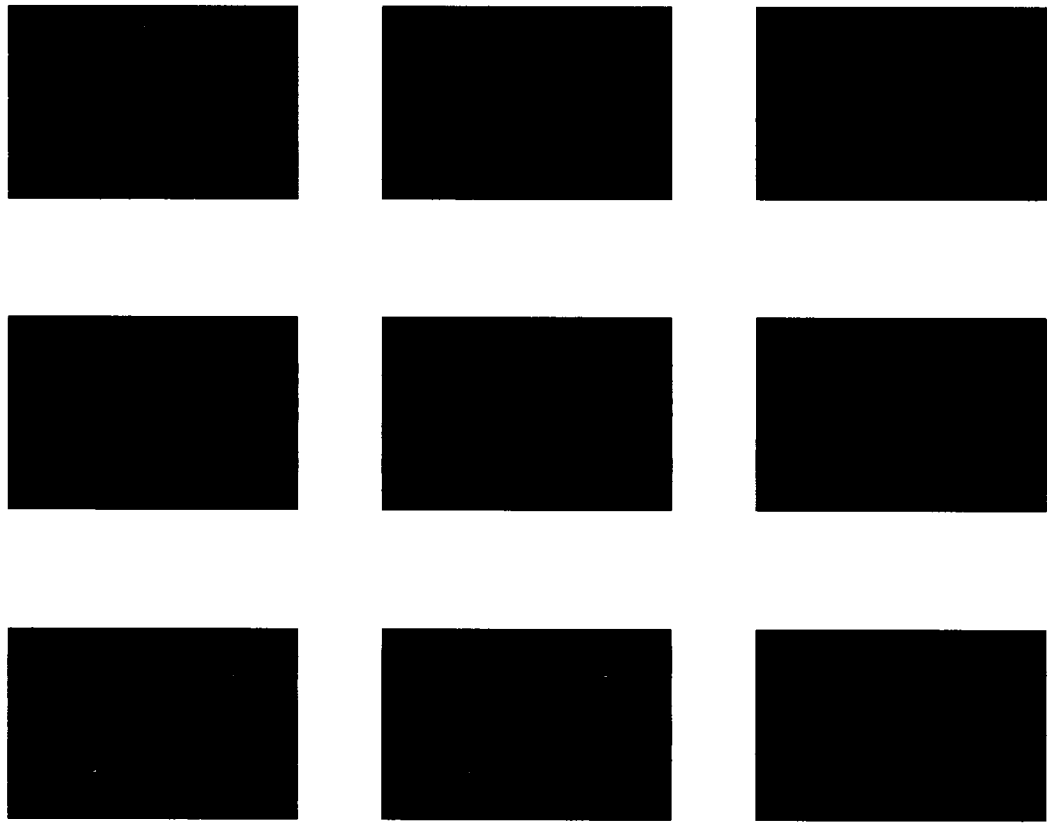


Figure 3.11. Frames 1-800 in steps of 100 from video skinfilter_evaluation3.avi and their corresponding selected and detected skin regions.

TABLE 3.5

RESULTS ON VIDEO SKINFILTER_EVALUATION3.AVI

Frame number	Percentage false detection			Percentage missed detection		
	Final algorithm	David O'Mara's model	Fleck and Forsyth	Proposed Algorithm	David O'Mara's model	Fleck and Forsyth
1	.4997	6.8362	.9230	25.4736	4.0639	15.0661
25	.5628	12.4083	.8791	20.6187	4.7905	11.2029
50	.4791	13.4334	.8545	19.3129	4.1361	10.0400
75	0	5.5802	0	0	0	0
100	2.89x10^-4	4.1050	.0064	97.9334	92.2424	96.1212
150	.4323	11.1409	.5839	44.7876	20.6564	39.2664

175	1.1143	15.6502	1.2746	30.6612	11.6804	25.3719
200	1.8270	21.8573	1.9644	41.8375	12.0159	33.3523
225	1.5281	25.2321	1.7896	46.8610	12.9433	44.5577
250	.5683	23.7176	.6670	66.3065	14.4907	54.6678
275	.8374	23.9766	.9812	52.0535	10.3484	42.3281
300	.5573	20.9034	.6774	56.0999	17.9100	42.6913
350	.4112	25.3484	.4609	60.3621	18.0172	49.2178
375	.2902	34.6591	.3519	86.5845	12.1973	79.5127
400	.3973	30.6774	.5388	61.3482	5.3084	57.2764
425	2.8258	37.3562	3.8776	45.2842	3.0398	40.0336
450	.6435	.6247	.7885	60.8112	4.6014	57.9720
475	1.1681	52.1976	1.4303	41.0831	1.4833	36.5298
500	.5842	4.7630	.8030	35.1104	3.2137	30.2578
575	.4757	4.1759	.6814	37.5377	6.8623	31.7133
600	20.6302	49.0706	24.4421	29.3629	15.2571	13.7443
625	24.2488	5.2182	25.4734	25.4544	11.4406	14.3877
650	22.6490	40.3108	24.0796	25.8791	14.9620	11.4110
675	1.2054	10.1273	1.1021	29.8301	14.7969	13.7009
750	11.0339	19.2902	11.9832	26.1019	13.4183	12.4423
800	4.8394	30.8681	5.5611	49.0621	14.7921	32.6011
Average values	4.1378	20.9077	4.4500	43.6113	13.6241	35.2160

The video skinfilter_evaluation3.avi has been chosen especially in view of the fact that the color of the man's shirt is extremely close to the skin color. As expected, the performance degrades since the filter falsely detects the shirt along with the skin. However, in the frames where the shirt cannot be seen, the performance is extremely good. The video also has significant illumination variance across its frames and the filter does well in these changing lighting conditions. The percentage false detections in this video becomes comparable to that of the Fleck and Forsyth algorithm, since the latter algorithm is able to block most of the non-skin colors. Therefore, the weakness of the Fleck and Forsyth algorithm does not come to light in this particular example. The output of the skin filter on the tested video frames is present in the attached DVD [DVDROM:\chapter3\readme.doc].

3.6.1.4 Results for the video skinfilter_evaluation4.avi

The results on video skinfilter_evaluation4.avi are shown in Figure 3.12 and summarized in Table 3.6. This video has been chosen, as it is shot under bright sunlight. This gives entirely different illumination conditions than the previous videos.

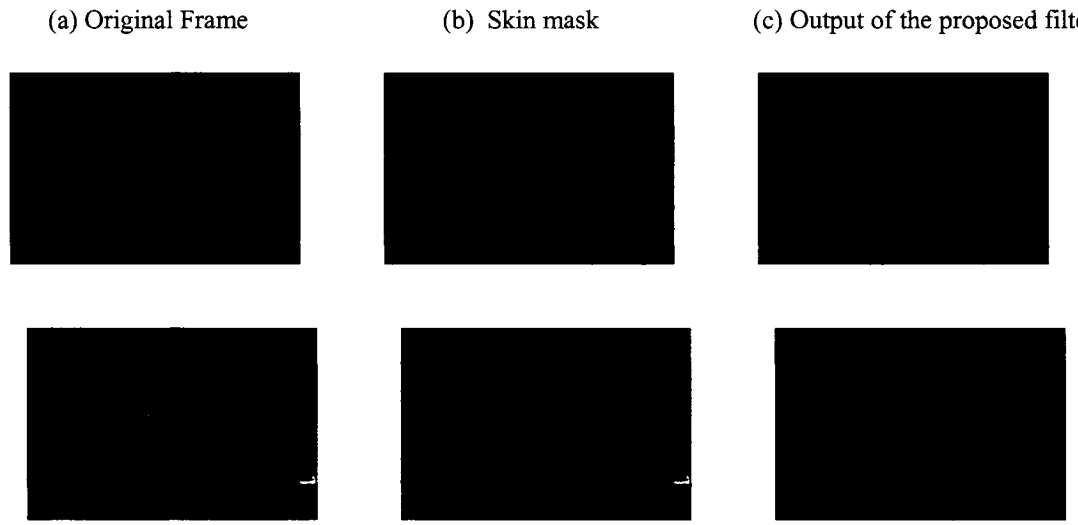


Figure 3.12. Frames 1 and 100 from video skinfilter_evaluation4.avi and their corresponding selected and detected skin regions.

TABLE 3.6

RESULTS ON VIDEO SKINFILTER_EVALUATION4.AVI

Frame number	Percentage false detection			Percentage missed detection		
	Proposed Algorithm	David O'Mara's model	Fleck and Forsyth	Proposed Algorithm	David O'Mara's model	Fleck and Forsyth
1	.2078	4.4306	.2963	25.0608	24.6959	2.6764
50	.0379	4.3550	.02402	18.8997	16.9476	2.9281
75	.0399	2.2292	.0723	50.5578	12.1251	32.5347
100	.1247	1.0463	.5370	16.9209	11.4371	11.1241
Average values	0.1025	3.0152	0.2324	27.8598	16.3014	12.3158

The average values for the false detection and undetected area in the selected region indicate that the performance of the proposed filter improves, since the lighting conditions are better in this video compared to the previous cases. The success rate of detection is as high as 73% (see Table 3.6). The percentage false detection for the proposed algorithm remains low inspite of the presence of the blue regions to which O'mara's model is extremely sensitive.

The tabulated evaluation results indicate that for O'Mara's model, the percentage of the undetected area in the selected regions remains low. This indicates that the O'Mara's model identifies wider range of colors as skin color. While this property is good for problems where all the skin regions need to be identified, it also identifies numerous regions which are non-skin. This is seen from the average percentage of false detection, which is extremely high for the O'Mara's model. This is highly undesirable in our problem, as we need the algorithm to be extremely accurate. Both the Fleck and Forsyth algorithm and the proposed algorithm have a much better performance in terms of false detections. Though the average percentage undetected area remains high for the proposed algorithm, it indicates that the algorithm is highly accurate in suppressing non-skin regions while maintaining a reasonable performance for detecting the skin color regions.

3.7 Summary

In this chapter, a new method of detecting the skin color in images has been proposed. The proposed method exploits the Fleck and Forsyth and O'Mara's models along with a filtering in the RGB space to detect the skin color. The performance of the proposed filter

has been evaluated on several videos which indicate an improvement in the performance achieved over the Fleck and Forsyth model or the O'Mara's model used individually. Based on the evaluation of the filter on random videos and our test videos, we may conclude that an extremely robust skin filter has been developed. As the primary objective of the skin filter is to help locate possible regions of interest, it needs to be extremely sharp. Under hospital environment, where the presence of colors close to skin color is remote, the proposed skin filter is expected to give excellent performance. The filter is not designed to achieve extremely efficient successful detections, but to suppress the spurious detections. The results obtained using this filter on a number of test videos indicate that the filter provides acceptable performance in terms of the suppression of the non-skin colors, while maintaining a desirable skin color detection rate of around 60%. Thus, the proposed filter serves our needs well and should act as a strong backbone for our final tracking algorithm.

Chapter 4

Line Detection Using Hough Transform

The skin color filter presented in Chapter 3 provides significant information about the possible location of the patient in the room. To confirm the presence of the patient at the locations suggested by the skin color filter, some additional feature needs to be identified. As discussed in Chapter 1, the patient needs to wear a cap that fits snugly onto the head to hold the electrodes and the wires. There is a possibility of drawing patterns such as circles, squares, straight lines, etc. on the cap which can be detected by using pattern recognition techniques. While circles and squares are subject to deformation with the change in field of view, the straight lines are expected to provide a similar cross sectional view from all angles. Therefore a pattern of straight lines on the cap is more suitable as a feature. Detection of this potential pattern requires a robust line detection technique. Hough transform for line detection is a well known algorithm for detecting straight lines and is the subject of this chapter.

The chapter starts with an introduction of the Hough Transform in Section 4.1. The Hough Transform for line detection is then discussed in Section 4.2 along with the concept of Hough peaks and accumulators. The line detection process is then explained with the help of an example in Section 4.3.

4.1 Introduction

Detection of simple shapes like lines, ellipses, circles are basic to the problem of computer vision. Most man-made objects can be identified using one or more of these basic shapes. Mathematically, any basic shape can be defined in the two dimensional space using variables and parameters. Hough transform is a benchmark tool in image processing that recognizes patterns by interchanging the role of variables and parameters. [51].

The Hough transform [52], was introduced in 1962 by Paul Hough, though the present form owes its existence to Duda & Hart [53]. In 1981, Dana H. Ballard introduced the transform in the field of computer vision [54] and due to its many desirable properties, it soon caught the attention of the researchers, and became a focus of research in the 1980s. During this period, the Hough transform was used to develop new algorithms in the field of polyhedral object recognition [55], character recognition [56, 57], satellite image processing [58], image segmentation[59, 60], motion estimation [61], recognition of 3D objects [62], document processing [63], foil fencing [64], etc. Since then, the Hough transform has been of wide interest for most of the researchers, who have been trying to make significant improvements in these areas. Today, the Hough transform has found use not only in computer vision, but also in areas like quantum gravity [65], pulsed phase thermography [66], biometrics [67], and nano technology [68].

The Hough transform follows the principle of maximum likelihood detection. It is thus suitable where more data points in the variable space are available. Rather than confining to a small set of data and then trying to fit a curve by exploiting the remaining data gradually, the Hough transform performs a global search. The fact that each data point is

given an equal weight and the decision is based on the basis of an analysis of all the available data, makes the method extremely robust. The main advantage of the Hough transform technique is that it is tolerant to discontinuities in the feature boundaries and is relatively unaffected by the image noise. This makes the Hough transform suitable for practical scenarios, where the presence of the noise and the imperfect feature boundaries are unavoidable.

4.2 Line detection using Hough Transform

In a two dimensional *(x, y)* space, where x and y are the variables, any line can be represented in the form,

$$y = ax + b.$$

where *(x, y)* are the variables and *(a, b)* are the parameters that define a specific line. That is, any line in the *(x, y)* plane or the variable space can be represented by specifying a value of the parameters a and b. Changing the value of any or both of these parameters yields a new line. This holds true for any two dimensional curve in the variable space as each of them is represented by a unique set of parameters.

The idea behind the Hough transform is to interchange the roles of the variables and the parameters. The points on the curves (i.e. any x_i and y_i on the line) are mapped onto the parameter space, which gives a different perception of the curve that is much easier to interpret and recognize. This section presents the line detection process using this transformation.

4.2.1 Hough - Line Transform

The Hough transform used for the line detection is explained below.

The equation of any straight line segment in an image is given as

$$y = ax + b \tag{4.1}$$

where x and y are variables and a and b are constants.

If (x_i, y_i) be any point in the image, then as shown in Figure 4.1 (a), all the lines which pass through this point have the form

$$y_i = a\, x_i + b \tag{4.2}$$

Infinitely many lines can pass through this particular point (x_i,y_i) and the equation of these lines can be determined by changing the values of the parameters a and b.

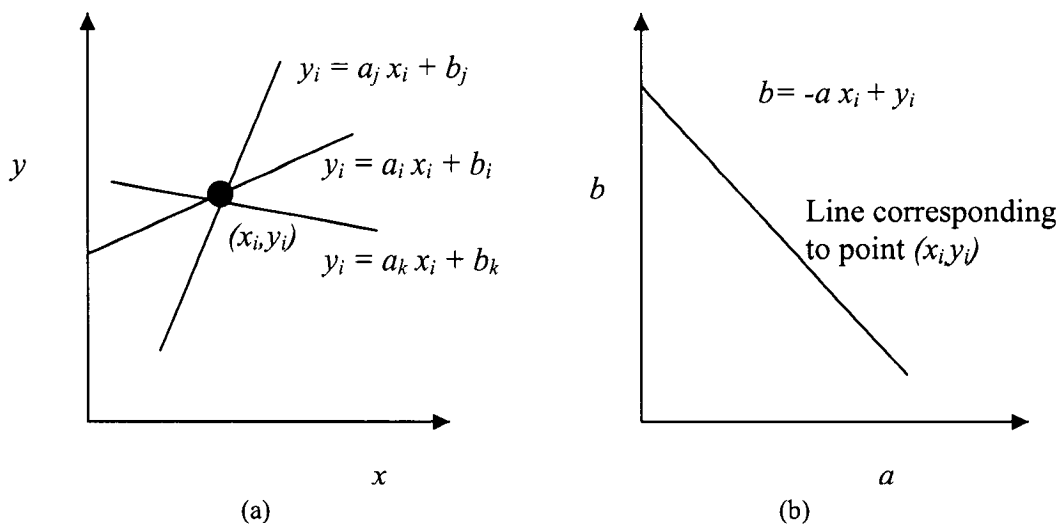


Figure 4.1. (a) Infinitely many lines pass through (x_i, y_i) for different values of a and b. (b) The family of lines in the (x, y) space passing through (x_i, y_i) represented by a single line in the (a, b) space.

Equation (4.2) can be rewritten as,

$$b = - a x_i + y_i \tag{4.3}$$

where x_i, y_i are now constants, and a and b are the variables. This is an equation of a straight line in the *(a, b)* space, as shown in Figure 4.1 (b).

If there are two points u and v in the *(x, y)* space, which are on the same line as shown in Figure 4.2 (a), then for each of these points all the possible lines through them are represented by a straight line in the *(a, b)* space as shown in Figure 4.2 (b). Therefore, the line that passes through both the points u and v in the *(x, y)* space lies on the intersection of the two lines u and v in the *(a, b)* space.

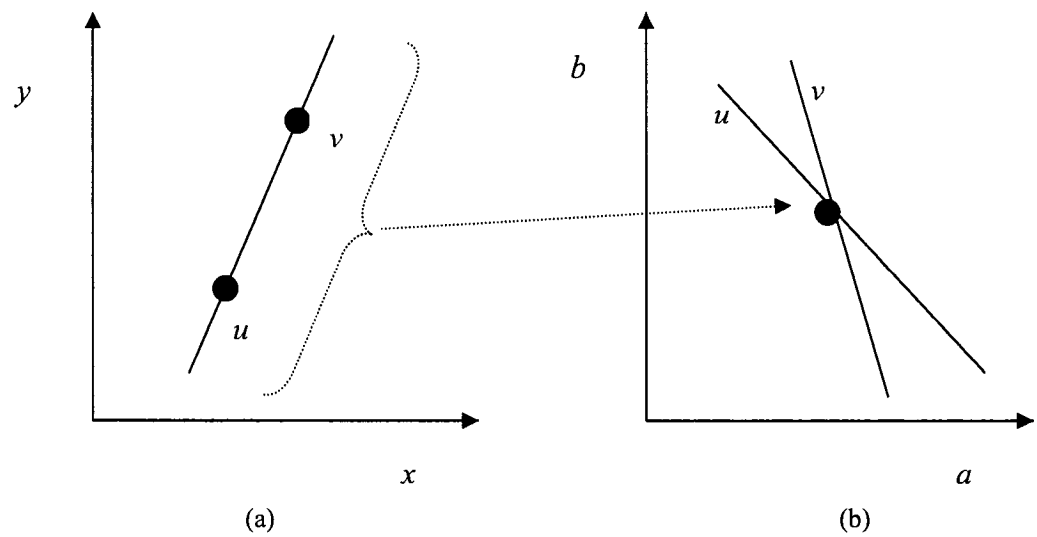


Figure 4.2. (a) Two points lying on the same line in *(x, y)* space. (b) Line through points u and v in *(x, y)* space is now point of intersection of lines u and v in *(a, b)* space.

Similarly, as shown in Figure 4.3, all the points lying on the same line in the *(x, y)* space can be represented by concurrent lines in the *(a, b)* space.

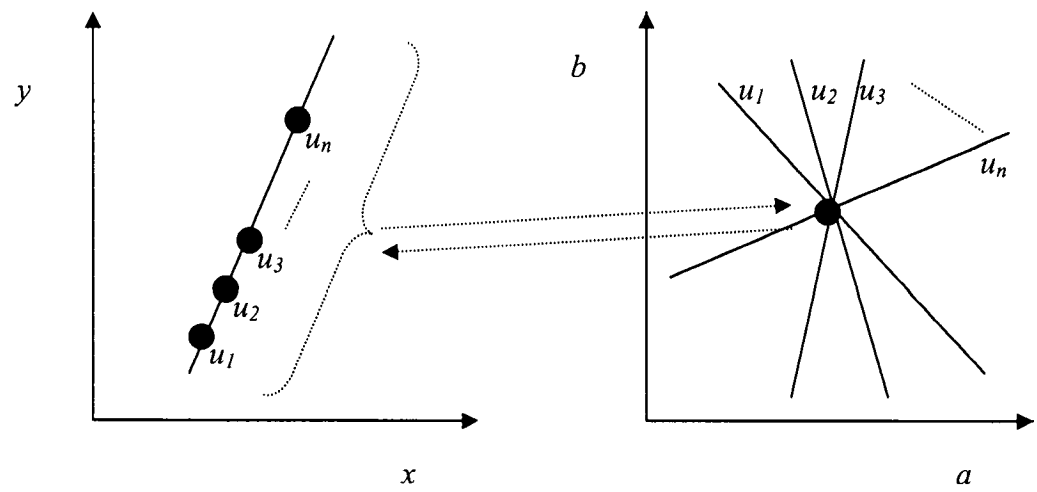


Figure 4.3. All the points on a line in the *(x, y)* space are represented by concurrent lines in the *(a, b)* space.

Hence, concurrent lines in the *(a, b)* space represent collinear points in the *(x y)* space. Therefore, the straight lines in the *(x, y)* space can be detected by investigating the concurrent lines in its transformed space, i.e., the *(a, b)* space. The lines in digital images are represented by finite number of pixels. Therefore, the length of a line in the *(x, y)* space of an image corresponds to the number of lines in *(a, b)* space passing through the common point of intersection. This concept of detecting straight lines is called the Hough line transform for the cartesian plane.

In the equation $y = ax + b$, a represents the slope of the line. In the specific case of vertical lines, the slope is infinite and therefore the value of a is infinity. In the *(a, b)* space, this value cannot be represented. To avoid this, the normal representation of a line in the *(x, y)* space can be used :

$$x\sin\theta + y\cos\theta = \rho \tag{4.4}$$

where ρ is the perpendicular distance from the origin and θ is the angle that the normal makes with the y axis, as shown in Figure 4.4.

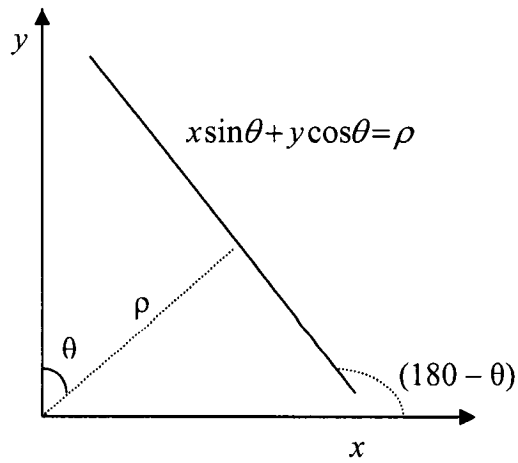


Figure 4.4. Normal representation of a line.

The concept in Figure 4.2 can now be used to represent the transformation from the *(x, y)* space to the *(ρ, θ)* space. For each point in the *(x, y)* space, there would be a curve in the *(ρ, θ)* space as shown in Figure 4.5.

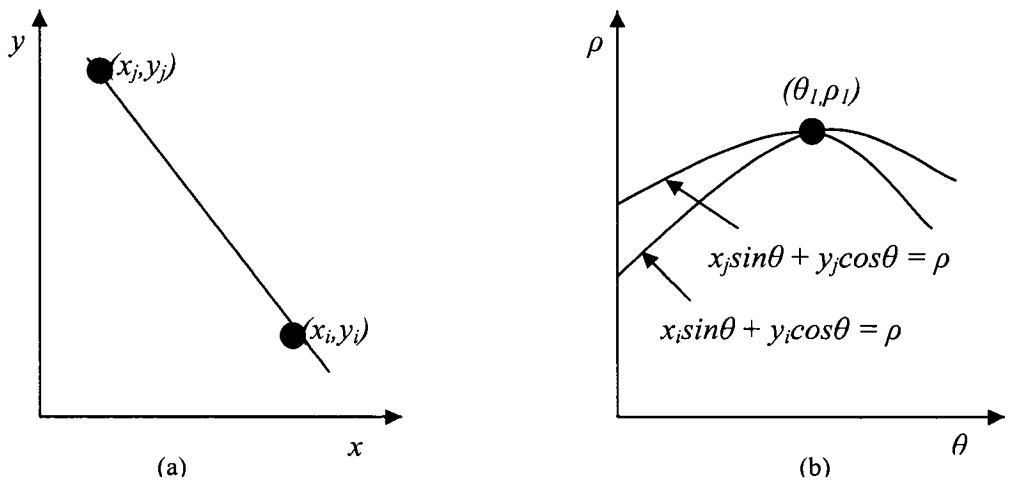


Figure 4.5. (a) Two points in the *(x, y)* space. (b) Curves in the *(ρ, θ)* space corresponding to the two points in the *(x, y)* space

Similar to the representation in Figure 4.2 (b), the line passing through the two points in the *(x, y)* space is the point of intersection of the two curves in the *(ρ, θ)* space (see Figure 4.5 (b)).

The advantage of using the *(ρ, θ)* space lies in the fact that the slope of the line in *(x, y)* space is represented by the parameter θ which can only take values in the range [-180^0, 180^0] and therefore, the problem of the slope tending to infinity is avoided and the lines in the *(x, y)* space with infinite slope can now be represented in the *(ρ, θ)* space by a single point at $\theta = 90$ degrees as shown in Figure 4.6.

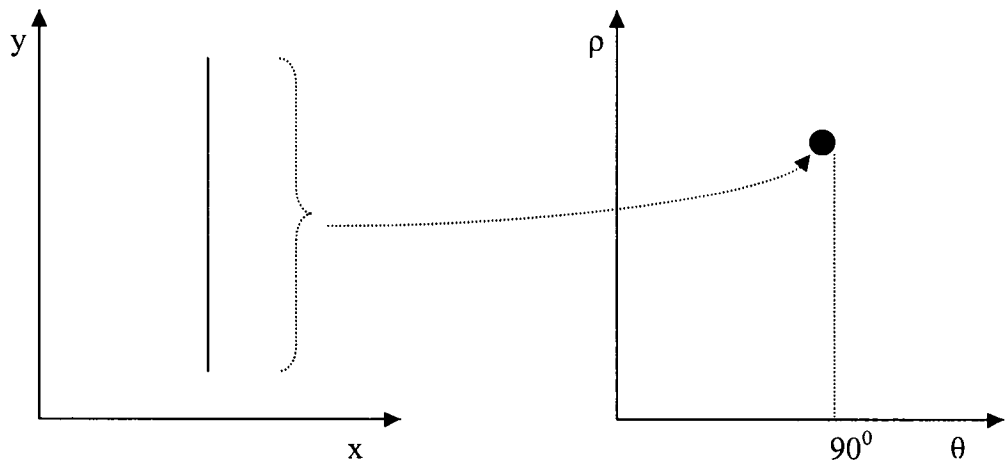


Figure 4.6. A vertical line in the *(x, y)* space can be represented in the *(ρ, θ)* space by a point for which $\theta = 90$ degrees.

Since all points on a line in the *(x, y)* space map to a single point in the *(ρ, θ)* space, it is possible to detect straight lines by using the concepts of the Hough peaks and accumulators, as discussed in the next section.

4.2.2 Hough peaks and accumulators

The Hough transform provides the transformation of a line in the *(x, y)* space to the *(ρ, θ)* space. Each point in the *(x, y)* space is represented by a curve in the *(ρ, θ)* space. If there are *n* points in the *(x, y)* space that lie on a straight line, then as shown in Figure 4.7, there would be *n* corresponding curves in the *(ρ, θ)* space, which are concurrent. The lines in the *(x, y)* space are therefore the points of intersection of the curves in the *(ρ, θ)* space. The number of curves passing through a particular point in the *(ρ, θ)* space thus determines the number of points lying on a line in the *(x, y)* space. These points of intersection of curves in the *(ρ, θ)* space are called Hough peaks and the intensity of a Hough peak is defined as the number of curves passing through that point. The length of a line in an image is determined by the number of pixels constituting the line. Since longer lines have more curves associated with them, the Hough peaks corresponding to longer lines are brighter.

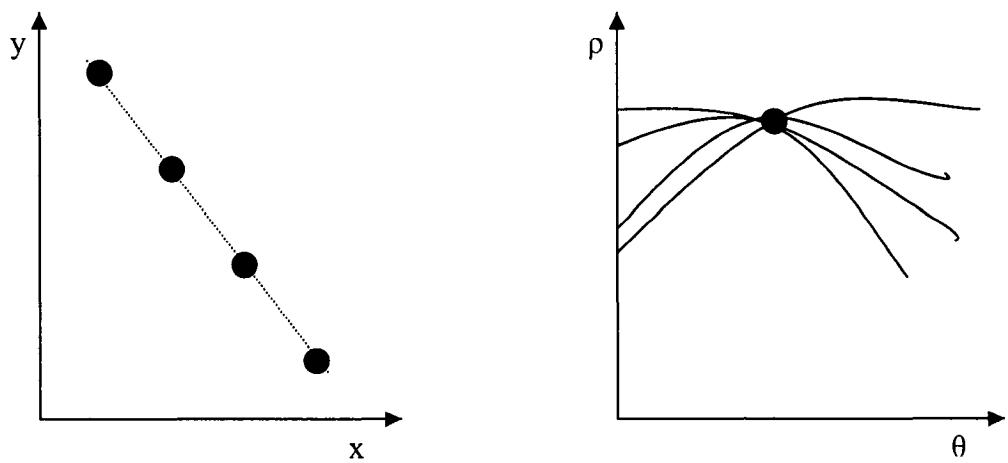


Figure 4.7. (a) Points lying on a line in the *(x, y)* space. (b) The corresponding curves in *(ρ, θ)* space are concurrent.

The concept of detecting straight lines in the Cartesian space can also be extended to the *(ρ, θ)* space. The straight lines in the *(x, y)* space can be detected by examining the presence of the concurrent curves in the *(ρ, θ)* space. The length of any particular line in the (x, y) space corresponds to the number of curves in the *(ρ, θ)* space passing through the common point of intersection.

However, in images captured in real environment, the points comprising the line defining a feature boundary may not be aligned perfectly and are expected to deviate from the expected slope of the line. This implies that the points lying on the line may not be collinear as shown in Figure 4.8 (a). Due to this misalignment of segments of a line, the curves corresponding to these points do not intersect at the same point and therefore there may be several peaks for a single line as shown in Figure 4.8 (b).

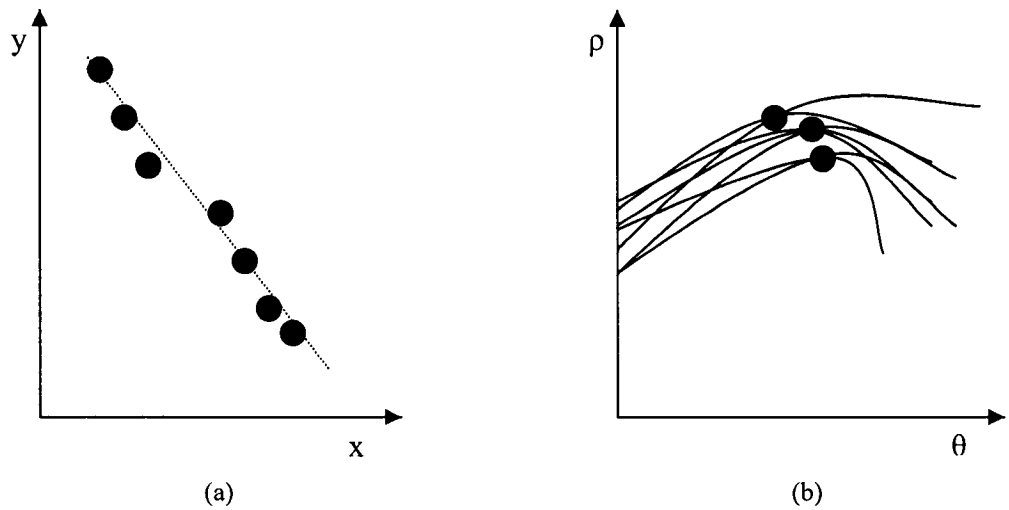


Figure 4.8. (a) Lines obtained from real time images might not be perfectly straight. (b) More than one peak is possible in the *(ρ, θ)* space for this line.

However, as these points represent the same line, these peaks should lie close to each other and within a relatively small region. Thus, to make the Hough transform practical for line detection, the *(ρ, θ)* parameter space needs to be quantized by dividing this space into small cells. These cells are called the accumulator cells (Figure 4.9) as they accumulate peaks for the segments of single lines. The size of these cells as shown in Figure 4.9 depends upon the level of accuracy desired or expected.

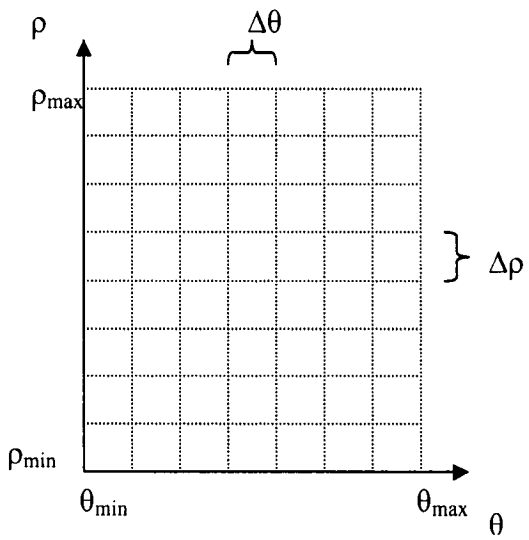


Figure 4.9. The *(ρ, θ)* parameter space subdivided into accumulator cells.

The accumulator cells are created in the area where the values of ρ and θ are expected. The values of θ are expected to be between -180 to + 180 degrees and ρ cannot be greater than the distance between the corners in an image. A range of acceptable error is defined by specifying the size of the accumulator cells. Thus, the angle defining the slope of the straight line is bound by the size of the accumulator cell, this is also called the margin of error and is bound by,

$$\textit{Range of acceptable error} : |\Delta\theta| \leq \pm \frac{\theta_{max} - \theta_{min}}{n} \tag{4.5}$$

where, n is the number of accumulator cells. A similar expression holds true for ρ.

$$Range\ of\ acceptable\ error : |\Delta\rho| \leq \pm\frac{\rho_{max} - \rho_{min}}{n} \tag{4.6}$$

4.2.3 Using Hough peaks to detect lines

The line detection process is now explained with the help of an example. Figure 4.10 shows an image with random shapes. To obtain the points in the image which define the contours of the shapes, edge detection is performed. The edge detected image (Figure 4.11) gives all the points (shown in white) that define the shapes in the image. The edge detection process has already been explained in Chapter 2, Section 2.2.1.

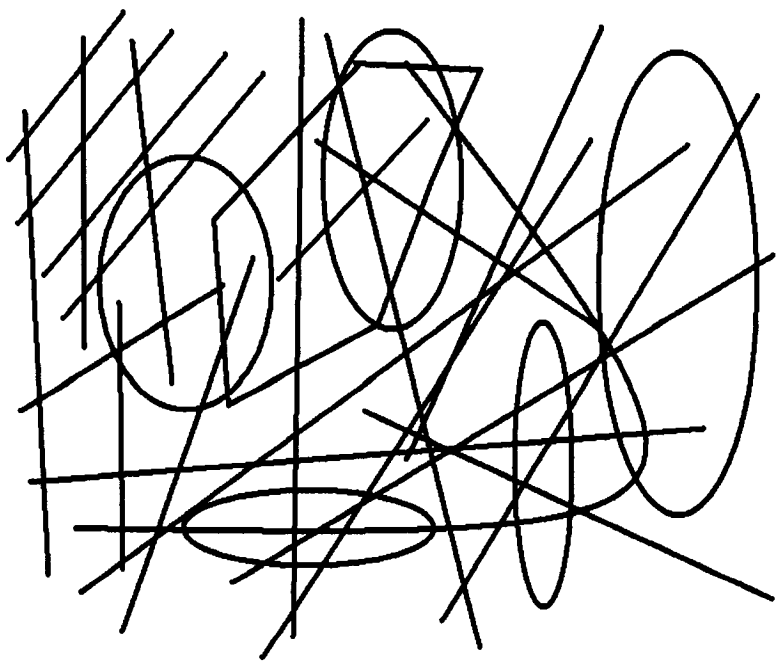


Figure 4.10. Image under test having random lines and curves

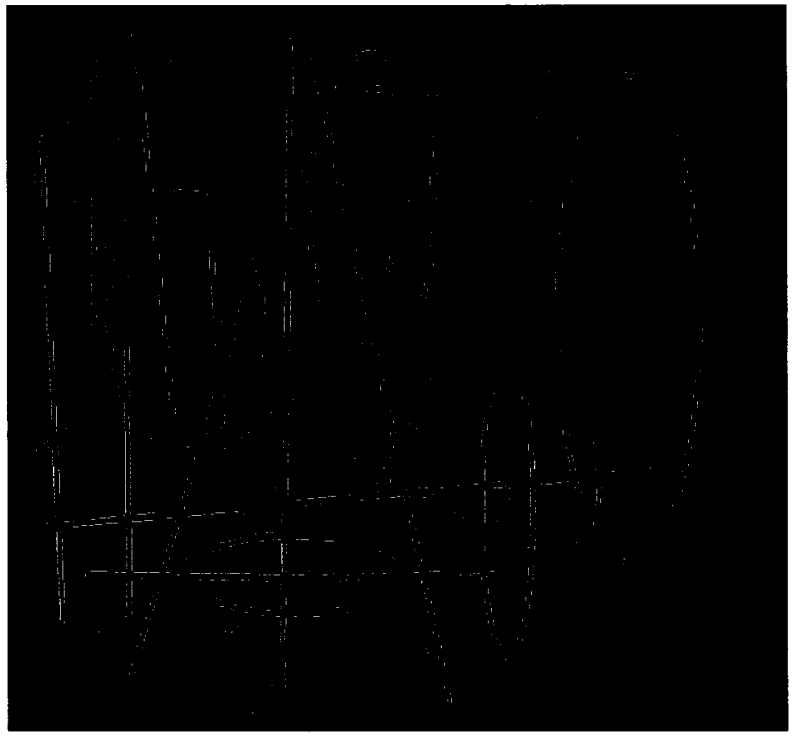


Figure 4.11. Edge detected logical image using Sobel operator gives the points in the image which are candidates for straight lines.

For any point in the edge detected image, the *(x, y)* coordinates give the physical locations of the point in the image. In the *(ρ, θ)* space, this point is represented by a curve (see Figure 4.5) which lies within the range of θ (-180 to +180- degrees). The curve extends within this range and takes on different ρ values for each value of θ. At the start of the line detection process, the value of each of the accumulator cells is set to 0. Since the curve takes all possible θ values within -180 to +180 degrees, the corresponding ρ values for each of these θ can be calculated using (4.4) and the accumulator cells corresponding to these ρ values are then incremented.

This process needs to be repeated for all the image points. At the end of this process, the values in the accumulator cells indicate the number of points lying on the corresponding line in the variable space and hence, the intensity of the peak. The accumulators with intensity greater than a predefined threshold can now be used as candidate peaks. The greater the intensity of a cell, the longer is the length of the line. Figure 4.12 shows the peaks obtained for the data points in Figure 4.11.

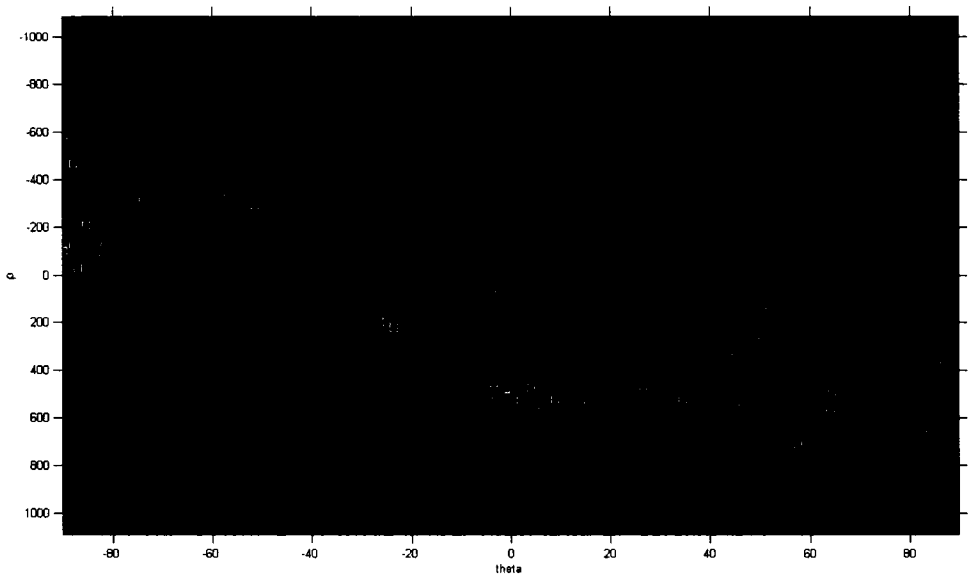


Figure 4.12. The Hough peak detection step figures out the brightest peaks in the *(ρ, θ)* space.

A practical problem in the real world images, as discussed earlier in the chapter, is that the lines are not perfectly straight and have broken segments with poor alignment. Hence, the peaks tend to lie in more than a single cell. To overcome this problem, the Hough peak with the highest value is found and the immediate neighborhood peaks are then suppressed. This can be repeated until the desired number of peaks are found. Once the desired peaks are extracted, the original lines can be traced by examining the points in the

(x, y) space corresponding to the curves in the *(ρ, θ)* space contributing to the peaks. These points have been highlighted in Figure 4.13 (in red color).

Figure 4.13. Line detection performed based on the Hough peaks in Figure 4.12. Detected lines in the image shown in red.

To detect specific patterns, some parameters like the minimum and maximum lengths of the line, number of lines, specific orientation of the lines, etc. can be used. For patterns such as parallel lines, the occurrences of the Hough peaks for a particular value of theta can be sought. For lines perpendicular to each other, the respective angles of inclination would be 90 degrees apart. Such a pattern can be detected by identifying the Hough peaks for values of θ that are exactly 90 degrees apart. Thus, the numerous patterns can be easily created and detected using the line detection method.

4.3 Summary

In this Chapter, the Hough transform for line detection is presented. The Hough line detection transform provides a reliable platform to detect patterns containing straight lines. Therefore, a feature containing straight lines can be developed for our application and it is expected to be detected successfully using the method presented in this chapter.

Chapter 5

Tracking Patients using Hough Transform and Skin color detection

5.1 Introduction

This chapter presents a new method to track the movements of patients undergoing EEG monitoring. The new method exploits the skin color as a feature, along with the cap on the patient's head with a pattern drawn on it as a second feature. The pattern on the cap is a set of straight parallel lines and is detected using the Hough transform for line detection, discussed in Chapter 4. As shown later in this chapter, these two features are sufficient to track the location of the patient in the room.

The chapter starts with a discussion of the cap worn by the patient (Section 5.2). This includes the discussion of the cap as a possible feature followed by a discussion on developing the pattern on the cap. The new algorithm for locating the position of the patient is then discussed in Section 5.3, and Section 5.4 discusses the optimization of the parameters involved in the different stages of the algorithm. The performance of the algorithm tested under variable lighting conditions, in the presence of multiple people and real time processing capabilities is presented in Section 5.5.

5.2 The Electrode cap

5.2.1 The "electrode cap" as a feature

To monitor the EEG of epileptic patients, electrodes are placed on the head of the patients. It is very common to make the patient wear a cap that keeps the bunch of loose wires together as a single braid which otherwise could be a cause of inconvenience and poor recording of EEG data. This cap will hereafter be referred to as the "electrode cap". The cap being a spherical feature is independent of the rotational issues usually involved in the detection of features. As it is placed snugly on the head, there are no issues related to the potential change in the shape of any patterns drawn on the cap or the cap itself. The cap is thus a potentially strong area which can be developed as a significant feature.

Recall that in the case of the ellipse-based face detection algorithm, the skin filter provides the possible regions of interest and the elliptical-shaped heads are searched in the window surrounding the regions of interest. In a similar manner, the modified electrode cap with some significant pattern on it can be searched for in the regions of interest given by the skin filter. A specific color of the cap may allow it to be visibly differentiable and can thus be identified by using pattern recognition techniques. The electrode cap may thus make a suitable feature for the identification of a patient's head in the video frames.

5.2.2 Developing a pattern on the cap using straight lines

It is possible to draw patterns like lines, squares, circles, rectangles, etc. on the cap. However, the pattern developed on the cap should remain independent of the direction of

view and rotation or tilt of the head. This pattern also needs to be good enough to withstand possible changes in the lighting conditions. Since the skin color filter alone is not sufficient to differentiate the patient from other people in the image, we have to use the pattern on the cap to identify the patient. Therefore, the pattern should be simple enough so that it can be detected reliably by the available pattern recognition techniques.

A set of parallel lines is one possible pattern which is rotational, directional and orientation independent. Thick lines going around the cap in a circular pattern on the cap (see Figure 5.1), are expected give an impression of nearly straight parallel lines from any field of view to the camera. Once the regions of interest (ROI) have been isolated by the skin color filter, the parallel lines can be searched for inside the ROI windows. The target can be successfully located, if enough lines are detected inside the window. Figure 5.1 indicates how the electrode cap is modified.

Figure 5.1. Simulation conditions with the patient wearing the modified cap.

The image shows a simulation environment with a person wearing the modified electrode cap. The base color of the cap has been kept black with white stripes to give the highest possible contrast. The high contrast will allow the detection to be applicable in low lighting conditions. The distance between the stripes has been kept large enough for the stripes to be clearly differentiable from one another under all possible fields and distance of view from the camera.

To develop the pattern on the cap, several combinations of painted lines were tried. Vertical lines were painted on the cap, but that gave an impression of lines originating from a common point at the top of the head and thus did not appear parallel. We tried painting fewer but thicker lines, but this did not show any significant improvement in the detection of the pattern. Increasing the number of painted stripes from three to four lowered the distance between the lines, thereby bringing them too close together to be differentiated from large distances. Therefore, three stripes seemed to be a suitable choice. Each stripe is thick enough to provide two parallel lines one at the top edge and the other at the bottom edge. Thus, three stripes allow six parallel lines to be detected. Apart from this, the boundary of the cap that separates the cap from the facial skin also appears to be a straight line. Depending upon the shape of the head, the top of the cap creates a line against the background and leads to an additional parallel line. Thus, eight parallel lines on the cap in the neighborhood of the skin color form a feature that is unusual to be found anywhere inside a hospital environment, especially in a region of interest window as small as the size of a human head. This pattern on the cap thus forms a significant feature that can be used to identify the patient in the video frames.

5.3 The Algorithm

The previous section explains how the electrode cap has been developed as a significant feature by drawing a pattern on it. This pattern can be detected using the Hough transform technique discussed in Chapter 4. The skin color filter along with the electrode cap as a feature can now be used to detect the location of the patient. The proposed algorithm for locating the patient in the given video sequence is explained through the following steps.

ALGORITHM FOR LOCATING THE PATIENT IN THE GIVEN VIDEO SEQUENCE

STEP I:	Extract an image frame from the video and perform skin color filtering as described in Chapter 3.
STEP II:	Locate the regions of interest using the centroids of the skin regions and the appropriate window size.
STEP III:	Perform the edge detection on the original extracted frame as described in Section 2.2.1
STEP IV:	Search for parallel lines within the regions of interest identified in Step 2 and mark the region of interest with maximum detected lines as the location of the patient.

The output obtained indicates the location of the patient by displaying the selected ROI window in black color

The following example illustrates the algorithm.

Figure 5.2 shows a frame from the video evaluation1.avi [DVDROM:\chapter5\readme.doc]. The skin color filter detects the skin regions as shown in Figure 5.3 and the centroids of the three major skin regions are marked with blue color asterisks in Figure 5.4.

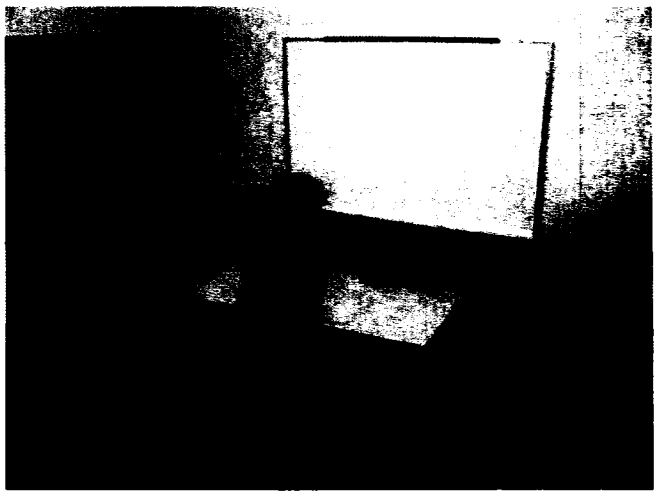


Figure 5.2. Frame 100 from video evaluation1.avi

Figure 5.3. Output of the skin color filter

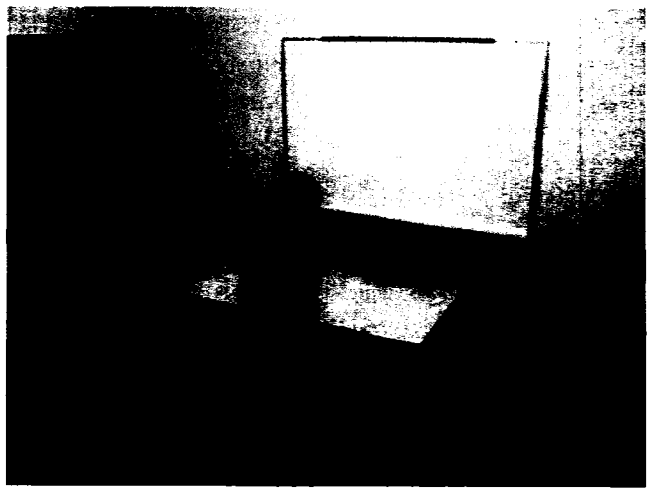


Figure 5.4. Centroids of the three significant skin regions marked as blue color asterisks.

The centroids of the skin regions are the centers of the regions of interest (ROI). Once the ROIs are located, windows large enough to contain the expected size of the head are selected around the centroids of the ROIs. These windows are selected from the edge detected image. In this example, these windows are selected manually around the detected centroids keeping the area of the extracted window large enough to accommodate the head. Figure 5.5 shows the output of the edge detection step and the extracted ROI windows are shown in Figure 5.6. A set of parallel lines is then searched in these windows using the Hough transform for line detection (as discussed in Chapter 4). If a sufficient number of parallel lines is found in any of these windows, it can be assumed to contain the patient's head.

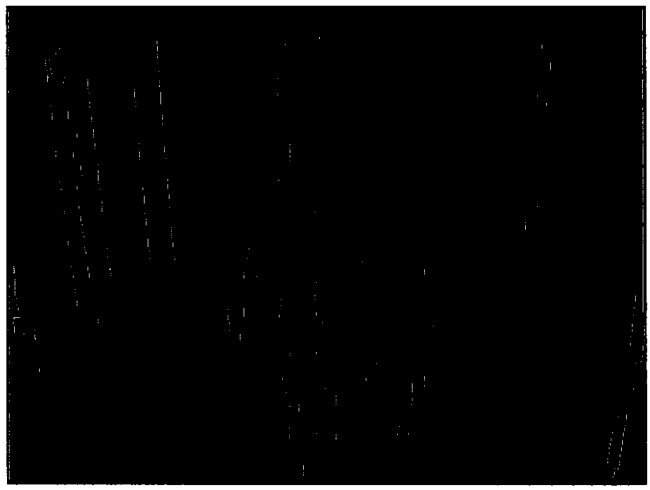
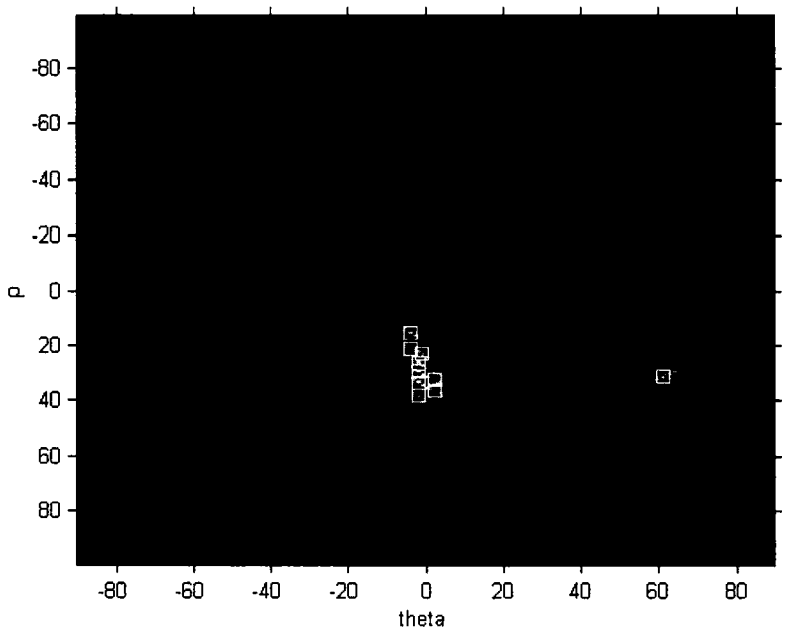


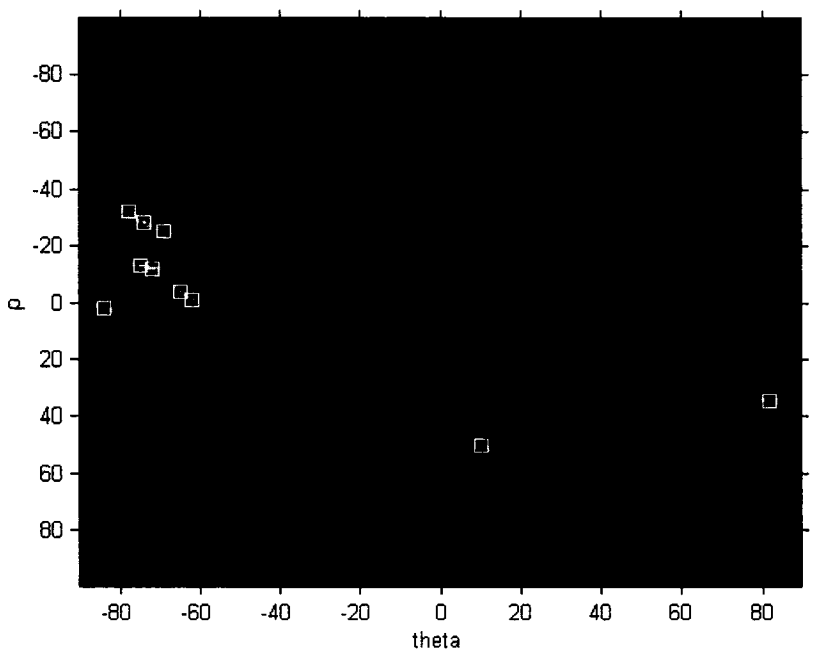
Figure 5.5. Edge detection performed on original frame

Figure 5.6. The extracted windows centered at the centroids of the three major skin color regions (magnified for better visibility).

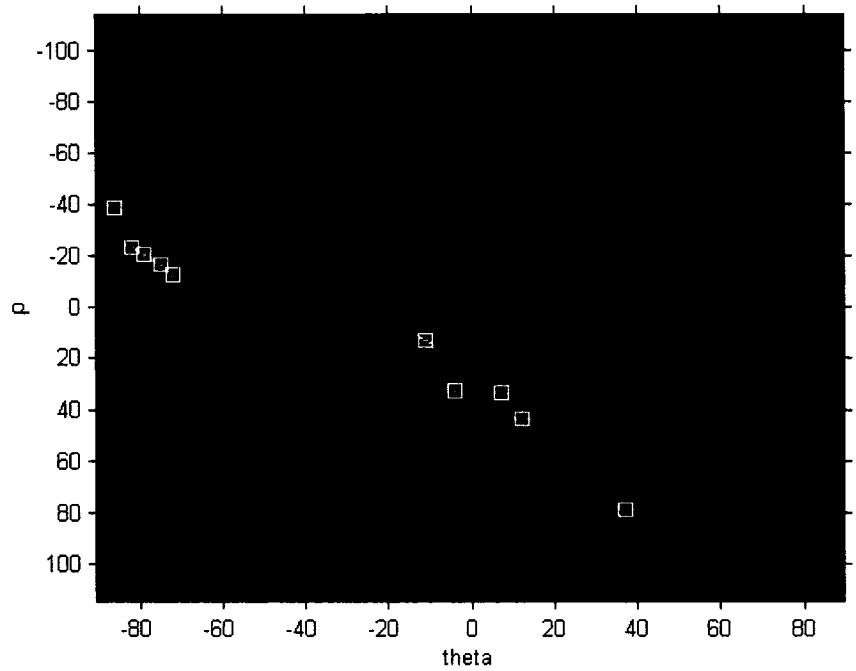
The Hough transform of these windows would detect the straight lines and the location of Hough peaks would suggest whether the lines are parallel. Figures 5.7 shows the Hough peaks for the three ROI windows of Figure 5.6.



(c) Hough peaks for ROI window shown in Figure 5.6 (a)



(d) Hough peaks for ROI window shown in Figure 5.6 (b)



(c) Hough peaks for ROI window shown in Figure 5.6 (c)

Figure5.7. The Hough peaks for the ROI windows shown in Figure 5.6.

In Figure 5.7 (a), there is a cluster of peaks near zero degrees. The peaks indicate the presence of straight lines and their presence near zero degrees indicates that they are all inclined at same angle (zero degrees) and thus are parallel to each other. In Figures 5.7 (b) and (c), the peaks are not concentrated at any particular angle and hence the detected straight lines are not parallel. This indicates the presence of the pattern of straight lines in ROI window of Figure 5.6 (a).

The detected lines in the ROI window (see Figure 5.6 (a)) based on the Hough peaks (see Figure 5.7 (a) are shown in Figure 5.8.

Figure 5.8. The detected lines corresponding to the Hough Peaks

The presence of straight parallel lines in the vicinity of skin color signifies the presence of the modified electrode cap in the frame. The location of the patient has thus been identified successfully.

5.4 Finding the optimum parameters

To ensure the optimum performance of the algorithm, each independent stage in the tracking process needs to be optimized, the independent steps being the skin filtering, edge detection and Hough transform for line detection. The skin filter development and optimization has already been discussed in Chapter 3. Sobel edge detection technique has been used for the edge detection step (discussed in Chapter 2), which is expected to detect all the possible edges in the image. The Sobel edge detection operators being standard do not involve any parameters. Therefore, only the Hough line detection stage needs to be optimized for detecting the parallel lines.

5.4.1 The Hough Line detection parameters

The Hough transform involves several parameters. Minimum length of the line, minimum gaps between adjacent line segments for them to be considered as part of a single line, the size of the accumulator cells and the number of Hough peaks to be considered are some major parameters involved in this stage.

The role of the accumulators for detecting straight lines has been discussed in Chapter 4, $\Delta\theta$ and $\Delta\rho$ being the accumulator size parameters. The $\Delta\theta$ parameter specifies the acceptable error in the orientation of the line being searched. The lines being detected might not be perfectly straight and certain pixels on it would tend to deviate from the median angle. The limitation of deviation, however, needs to be specified so that the pixels from adjacent lines are not mistaken to lie on the line under observation. A $\Delta\theta$ value of 0.5 degree is considered to be a good measure of the deviation, as it is large enough to account for the edge detection errors and small enough to constraint the expected line within safe limits.

As parallel lines are being detected in a window, the Hough peaks are expected to be within a certain angle bound by $\Delta\theta$. This detection remains independent of the $\Delta\rho$ factor as it has no role in the detection of parallel lines. Therefore for detecting parallel lines, this parameter is insignificant.

The minimum length of a line needs to be defined for it to be considered a line and not merely a stray cluster of pixels. The length of the line is defined by the number of pixels inside an accumulator cell having nearly equal ρ values for a particular theta. As discussed in Chapter 4, the ρ values depend upon the dimensions of the area under transformation, i.e., the size of the search window. As the length of the line depends upon

the ρ values, and ρ depends upon the window size, the minimum or maximum length of the expected line also depends upon the window size.

In the edge detection step of the real time images, not all the lines are likely to be detected to their full extent. This can cause gaps in between the adjacent line segments. These gaps, which may not be detected, can cause a line to be detected as several smaller line segments. In order to avoid this, a maximum gap length needs to be considered. Adjacent line segments having a gap length less than the threshold can be identified as part of the same line. However this gap must be defined in terms of the parameter ρ, which in turn depends upon the size of the window being considered. Thus, this parameter again depends upon the size of the window.

The number of Hough peaks, that must be searched for, depend upon the number of lines being searched. In the electrode cap, three painted stripes give an impression of six possible lines. Therefore, at least six parallel lines should be detected which implies six Hough peaks must be detected for any value of θ. However, due to the constraints such as a single line being detected as broken line segments and the threshold gap length, it is possible that a single line is detected as more than one line. Therefore, unless the parameters like the gap length and the minimum line length have been made constant, the number of Hough peaks to be detected cannot be specified. These parameters as discussed previously, depend upon the window size around the ROI.

As described above, most of the Hough line detection parameters depend upon the window size of the ROI in one way or the other. It can be concluded that if an optimum window size is found, the Hough line detection parameters can therefore be optimized.

5.4.2 The Window Size

Based on the discussion above, the optimum region of interest window size is important as it defines the performance of the Hough line detection stage. The window size depends upon the relative location of the patient from the camera. The performance of the Hough line transform also depends upon the window size, and therefore this tradeoff must be carefully balanced.

We address this issue by considering the performance parameters of the Hough line transform to be constant. An optimum window size that would give the best performance of the overall system can then be found. In other words, for any set of constant values of the Hough parameters, there should be a specific window size that gives the maximum number of line detections.

To evaluate the optimum window size, the Hough parameters values are taken as,

$\Delta\theta$=0.5 degrees

$\Delta\rho$=0.5

Minimum line length= 10 pixels

Maximum gap between adjacent line segments = 4 pixels

The test videos contain simulated conditions with a person wearing the modified electrode cap moving about the room. For each frame, the skin color filter gives the centroids of the potential regions of interest. The coordinates of the centroid of the skin color of the head are then isolated manually. Different window sizes centered at these coordinates are then tested for the performance in terms of the number of lines detected.

This step is done manually to remove the other detected regions (that are not part of the head).

The region of interest windows are taken around the centroid of the identified skin region. Since the cap lies on the head, it is actually the cap that defines the region of interest. Therefore, the performance as a function of the window size and location is also tested by shifting its location by 20 pixels above the centroid, where the cap is expected to be located. For each location and size of the window, the number of lines detected is observed. The location and size that gives the maximum number of line detections for the above mentioned Hough line detection parameters, is taken to be the optimum size and location of the window.

The evaluation is done on test videos evaluation1.avi and evaluation2.avi [DVDROM:\chapter5\readme.doc]. In each of the videos an attempt has been made to emulate the movements of the patient, the size of the room and the location of the camera. The lighting conditions are bright and no other person is present in the room except the patient. These conditions have been chosen to ensure that nearly perfect conditions are provided for the calculation of the optimum window size. Every 25^{th} frame has been processed in the videos to have a consistency in the assessment. The size of each extracted frame is 640 x 480 pixels.

Tables 5.1 and 5.2 summarize the results of the evaluation on these two videos. Column 1 shows the frame number and the location of the face centroid as detected by the skin filter. Column 2 shows the number of peaks detected for each window size. The number of peaks detected is the maximum number of peaks detected in the window for a

particular θ. The second row for the various frame numbers indicates the results obtained by shifting the window 20 pixels upwards.

For example, for frame 1 in video evaluation1.avi, the coordinates of the face centroid are (201, 46) and the number of peaks detected for a window of size 30x30 centered at the centroid is 4. The number of peaks detected for the same frame when the window is shifted 20 pixels upwards is 10. If the patient is out of the video frame as in frame 725, then this is denoted by marking the entries in the different columns by "X".

TABLE 5.1

EVALUATION OF THE OPTIMUM WINDOW SIZE FOR VIDEO EVALUATION1.AVI

Frame number and Coordinates of the face centroid.	Window size & number of peaks corresponding to possible parallel lines detected under it.				
	30x	40x	50x	60x	70x
1, (201,46)	4	6	9	8	6
Peaks after shift	10	10	10	9	8
25,(195,62)	7	8	8	9	7
Peaks after shift	10	10	9	9	8
50,(206,127)	4	6	9	7	6
Peaks after shift	10	10	10	8	9
75, (194,210)	7	8	9	10	8
Peaks after shift	10	10	10	10	10
100, (216,295)	5	7	9	8	9
Peaks after shift	10	10	10	10	10
125,(201,380)	5	7	8	8	9
Peaks after shift	10	10	10	10	10
150,(420,315)	1	4	8	8	6

Peaks after shift	10	10	9	9	7
175,(212,451)	2	2	3	5	4
Peaks after shift	10	10	10	9	7
200,(207,385)	5	4	5	5	6
Peaks after shift	9	9	10	8	9
225,(219,318)	6	7	8	10	8
Peaks after shift	10	8	9	10	10
250,(196,241)	3	5	9	9	9
Peaks after shift	10	9	9	9	10
275,(193,167)	5	7	8	8	8
Peaks after shift	9	8	8	7	7
300,(185,165)	6	6	7	5	5
Peaks after shift	9	10	10	10	5
325,(201,121)	7	7	8	8	8
Peaks after shift	10	10	10	9	8
350,(218,65)	1	0	3	6	9
Peaks after shift	10	10	10	9	7
375,(193,178)	5	6	7	8	7
Peaks after shift	10	10	10	10	8
400,(190,266)	6	4	4	6	6
Peaks after shift	8	9	9	8	8
425,(191,354)	5	4	5	6	4
Peaks after shift	10	8	8	9	7
450,(200,417)	3	3	4	5	5
Peaks after shift	10	10	9	9	7
475,(191,422)	4	4	7	4	5
Peaks after shift	9	9	8	6	6
500,(179,372)	4	3	4	4	4
Peaks after shift	10	9	7	7	5
525,(180,370)	4	5	6	3	3

Peaks after shift	10	10	7	6	5
550,(182,420)	4	3	4	4	4
Peaks after shift	10	9	8	7	6
575,(175,392)	3	4	5	4	4
Peaks after shift	10	10	9	8	5
600,(184,387)	2	4	5	7	4
Peaks after shift	10	10	10	8	5
625,(202,386)	1	2	6	8	7
Peaks after shift	9	10	9	9	9
650, (197,272)	7	8	10	10	10
Peaks after shift	9	10	10	10	10
675,(185,161)	7	8	9	8	9
Peaks after shift	10	10	10	10	10
700,(196,61)	7	6	8	8	8
Peaks after shift	10	10	8	7	6
725	X	X	X	X	X
Peaks after shift					
Average values	4.82	5.10	6.72	6.82	6.48
Average shifted values	9.72	9.58	9.17	8.62	7.65

TABLE 5.2

EVALUATION OF THE OPTIMUM WINDOW SIZE FOR VIDEO EVALUATION2.AVI

Frame number and Coordinates of the face centroid.	Window size & the number of peaks corresponding to possible parallel liness detected under it.				
	30x	40x	50x	60x	70x
1, (253,394)	5	4	8	7	7
Peaks after shift	10	10	9	9	8
25,(223,384)	7	8	9	8	6
Peaks after shift	10	10	9	10	10
50, (235,351)	6	6	5	5	5
Peaks after shift	10	8	9	9	8
75,(227,373)	5	6	7	7	7
Peaks after shift	10	10	9	10	10
100,(243,401)	6	9	10	9	8
Peaks after shift	10	10	10	10	10
125,(246,471)	9	7	10	9	8
Peaks after shift	10	10	10	9	10
150,(262,552)	7	9	10	10	10
Peaks after shift	10	10	10	10	10
175,(265,601)	10	9	10	10	10
Peaks after shift	10	10	10	10	10
200,(295,585)	5	4	6	7	8
Peaks after shift	10	10	10	9	9
225,(315,598)	4	3	4	4	5
Peaks after shift	10	10	10	10	9
250,(302,512)	6	6	7	9	8
Peaks after shift	10	10	10	10	10

275,(323,424)	5	6	8	9	10
Peaks after shift	10	10	10	9	8
300,(307,332)	4	6	9	10	10
Peaks after shift	10	9	10	10	10
325,(292,356)	2	5	7	8	8
Peaks after shift	7	9	9	9	9
350	X	X	X	X	X
Peaks after shift	X	X	X	X	X
375,(262,224)	2	3	5	8	9
Peaks after shift	5	6	9	10	10
400, (247,184)	5	7	7	8	7
Peaks after shift	9	8	8	10	10
425,(260,104)	4	6	5	6	5
Peaks after shift	10	10	10	9	6
450,(266,87)	7	8	10	10	10
Peaks after shift	10	10	10	10	10
475	X	X	X	X	X
Peaks after shift	X	X	X	X	X
500,(260,96)	2	5	6	8	7
Peaks after shift	10	10	9	9	9
525,(257,145)	4	7	8	8	9
Peaks after shift	9	9	10	9	10
550,(251,247)	5	5	9	8	9
Peaks after shift	10	8	10	10	10
575, (250,337)	6	7	9	9	10
Peaks after shift	7	8	8	10	10
600,(242,414)	6	5	9	9	8
Peaks after shift	7	8	10	10	10
625,(238,512)	7	9	9	10	10
Peaks after shift	10	10	10	10	10

650,(260,555)	6	3	4	5	7
Peaks after shift	10	10	10	9	10
Average values	5	5.66	7.07	7.44	7.44
Average shifted values	9.36	9.32	9.56	9.6	9.44

The results for these two videos in which various positions of the head are considered indicate that the performance is better for the windows that are shifted 20 pixels upwards in an attempt to get the window closer to the cap. It does not matter whether the head is really close to the camera or not, windows of size 30x30 or 40x40 provide sufficiently good performance by detecting almost equal number of lines on an average.

The results from Tables 5.1 and 5.2 indicate that on an average around 9 peaks are being detected. To assume a safe value of the threshold, the value 8 is considered sufficient for the algorithm. Therefore if at least 8 peaks are detected in an ROI window, it can be assumed to contain the pattern of straight lines drawn on the cap. In the case where more than one ROI window is found to contain more than 8 peaks, then the one with the highest value is selected.

5.5 Evaluation of the algorithm

With the optimum window size and threshold for the peaks to be detected having been found, the algorithm must be evaluated for its application in real time environment. The algorithm should be able to keep track of the patient under different physical conditions of the surrounding environment. As the performance of the skin filter and edge detection stage of the algorithm is expected to vary under varying illumination conditions, the

tracking performance of the algorithm also needs to be tested under varying conditions of illumination. Besides the varying illumination conditions, the performance of the algorithm also needs to be tested under situations, where people other than the patient are present in the room.

The test videos taken for experimentation consider these varying conditions of the surrounding environment. By defining a criterion for successful or unsuccessful tracking for each of the conditions under test, the evaluation can be mathematically stated in the terms of detection/tracking sensitivity (i.e. percentage of frames in which the patient is successfully identified).

5.5.1 Evaluation under varying lighting conditions

As the performance of the skin filter is known to vary greatly under changing lighting conditions, it therefore becomes important to evaluate the tracking performance of the algorithm under different lighting conditions. Three videos test1.avi, test6.avi and test2.avi [DVDROM:\chapter5\readme.doc] have been captured, respectively, under good, medium and low lighting conditions. The algorithm is tested on the videos with the processing rate of 1 frame per second. The output of these three videos is then manually evaluated and for each frame being processed, the success or failure is recorded. The algorithm detects the target by selecting one of the ROI windows. Success or failure is defined as follows.

- If the output of the algorithm is the ROI window corresponding to the patient's head, it is considered a success.

- If the patient is not present in the frame and no target is detected, it is considered a success
- If the target detected is not the patient, it is considered a failure.
- If no target is detected even if the patient is present in the frame, it is considered failure.

Tables 5.3, 5.4 and 5.5 summarize the evaluation process for videos test1.avi, test6.avi and test2.avi respectively.

TABLE 5.3

RESULTS OF TRACKING PROCESS ON EACH FRAME BEING PROCESSED FOR VIDEO UNDER GOOD LIGHTING CONDITIONS – TEST1.AVI

Frame Number	Tracking successful?
1	Yes
25	Yes
50	Yes
75	Yes
100	Yes
125	Yes
150	Yes
175	Yes
200	Yes
225	Yes
250	Yes
275	Yes
300	Yes
325	Yes
350	Yes
375	No
400	Yes
425	Yes
450	Yes
475	Yes
500	Yes
525	Yes

550	Yes
575	Yes
600	Yes
625	Yes

Under good lighting conditions, the tracking has been unsuccessful for frame 375. In this frame, the patient's face is not visible at all to the camera. Thus, even though the patient is in the frame, the skin color filter is not able to detect any skin regions of the face and hence no ROI window is selected on the patient's head.

From the data in Table 5.3,

Total number of frames processed = 26

Total number of frames with successful tracking = 25

Thus, the percentage success = 96.15%.

TABLE 5.4

RESULTS OF TRACKING PROCESS ON EACH FRAME BEING PROCESSED FOR VIDEO UNDER MEDIUM LIGHTING CONDITIONS – TEST6.AVI

Frame number	Tracking successful?
1	Yes
25	Yes
50	Yes
75	Yes
100	Yes
125	Yes
150	Yes
175	No
200	Yes
225	Yes
250	Yes
275	No
300	Yes
325	Yes

350	Yes
375	Yes
400	Yes
425	Yes
450	Yes
475	Yes
500	No
525	No
550	No
575	Yes
600	Yes
625	No
650	Yes
675	Yes
700	Yes
725	Yes

Under medium lighting conditions, the tracking is unsuccessful for frames 175, 500-550 and 625 due to the reason similar to that for the frame 375 of video test1.avi.

From the data in Table 5.4,

Total number of frames processed = 30

Total number of frames with successful tracking = 24

Thus, the percentage success = 80%

TABLE 5.5

RESULTS OF TRACKING PROCESS ON EACH FRAME BEING PROCESSED FOR VIDEO UNDER LOW LIGHTING CONDITIONS – TEST2.AVI

Frame number	Tracking successful?
1	Yes
25	Yes
50	Yes
75	Yes
100	Yes
125	Yes

150	Yes
175	Yes
200	No
225	No
250	No
275	No
300	No
325	No
350	No
375	Yes
400	No
425	Yes
450	Yes
475	Yes

From the data in Table 5.5,

Total number of frames processed = 20

Total number of frames with successful tracking = 12

Thus, the percentage success = 60%

Thus, under low lighting conditions, there are numerous frames for which tracking is unsuccessful. This can be related to the fact that as the lighting conditions are poor, the skin color filter is not able to locate all the skin areas correctly. This results in false ROIs being selected. Though the success percentage drops as the lighting conditions get worse, even a 60% success rate is reasonable considering the fact that in real time applications, the mobility of the patient would be extremely low under these low lighting conditions. In good and medium lighting conditions, the algorithm has been able to give an acceptable tracking performance.

5.5.2 Evaluation in the presence of multiple people

The purpose of the algorithm is to track a patient using the pattern developed on the cap, which differentiates the patient from other people. The performance of the algorithm needs to be measured in terms of its ability to make this distinction. The test videos test8.avi and test9.avi [DVDROM:\chapter5\readme.doc] have been captured under simulation conditions, where multiple people are present. The performance evaluation is made by evaluating the output of the algorithm on each of the frames having people other than the patient. The tracking process is considered successful if the patient is successfully identified in presence of other people. Success or failure is defined as follows.

- If the patient is successfully located in presence of other people, then it is considered a success.
- If the target detected is not the patient, then it is considered a failure.
- If no target is detected even if the patient is present in the frame, it is considered a failure.

As this evaluation is only for testing the performance in the presence of multiple people, those frames which do not involve people other than the patient have not been considered. This leaves fewer frames in the video sequence test8.avi for the purpose of evaluation. Therefore, the evaluation has been performed for two videos, test8.avi and test9.avi. The results are tabulated in Tables 5.6 and 5.7.

TABLE 5.6

RESULTS OF TRACKING PROCESS ON EACH FRAME BEING PROCESSED FOR VIDEO UNDER TEST WITH MULTIPLE PEOPLE – TEST8.AVI

Frame number	Tracking successful?
175	Yes
200	Yes
225	Yes
250	Yes
275	Yes
300	Yes
325	Yes
350	Yes

From the data in Table 5.6,

Total number of frames processed = 8

Total number of frames with successful tracking = 8

Thus, the percentage success = 100%

TABLE 5.7

RESULTS OF TRACKING PROCESS ON EACH FRAME BEING PROCESSED FOR VIDEO UNDER TEST WITH MULTIPLE PEOPLE – TEST9.AVI

Frame number	Tracking successful?
150	Yes
175	Yes
200	No
225	Yes
250	Yes
275	Yes
300	Yes
325	Yes
450	Yes
475	No
500	Yes

525	Yes
550	Yes
575	Yes
600	Yes
625	Yes
650	Yes
675	Yes

From the data in Table 5.7,

Total number of frames processed = 18

Total number of frames with successful tracking = 16

Thus, the percentage success = 88.88%

The results indicate that the percentage success rate remains high even in the presence of multiple people. This explains the robust character of the tracking algorithm. The frames where the target could not be detected were suffering from occlusion of the target.

5.5.3 Real time processing

The algorithm being developed for real time videos needs to have a processing speed high enough for real time applications. From the discussion in Chapter 1, the movements of the patient in the room are expected to be slow and limited. Therefore, the processing speed can be compromised at the expense of accuracy. Even with this advantage it has to be ensured that the algorithm does not have an extremely slow processing speed per frame. From the discussion in Section 1.5, the tracking algorithm should be able to achieve a processing speed of at least 1 frame per second. This processing rate is

expected to be sufficient, as the movements of the patient about the room are not very fast.

All the stages of the algorithm have been developed in Matlab. The algorithm was tested for processing speed on 9 test videos. Every 25th frame in the videos was processed and the processing time computed. Table 5.8 summarizes the results.

TABLE 5.8

REAL TIME PROCESSING COMPUTATIONS FOR VIDEOS TEST1.AVI – TEST9.AVI

Video name	Number of frames processed	Average processing time per frame (in seconds)
test1.avi	26	.994
test2.avi	21	.9379
test3.avi	41	.9576
test4.avi	38	.9893
test5.avi	15	0.9892
test6.avi	30	.98392
test7.avi	18	.95581
test8.avi	19	.95934
test9.avi	28	.9253

As seen form Table 5.8, the processing time per frame for all videos is below 1 sec. The performance has been measured on P4 3.2 GHz machine with 1GB RAM running windows XP professional. Matlab is known to have poor memory management capabilities and is thus slow for real time video processing applications. A C++ implementation of the algorithm is expected to yield much higher processing speeds. Even with Matlab, the processing speed of around 1 frame per second has been achieved.

5.6 The tracking system

This research has focused on developing an algorithm for the analysis software (see Figure 1.1) which can locate the patient in the room. The proposed algorithm can find the location of the patient as the coordinates of the center of the ROI window, which is found to contain the patient's head. This location information can be transmitted to the control system which can calculate the adjustments needed in positioning the camera to keep the patient in full view in the next frame. Therefore, with this proposed algorithm, a tracking system can be implemented to track the movements of the patient in the room automatically.

5.7 Summary

This chapter has presented a new algorithm for tracking epileptic patients through digital videos. The new algorithm is based on feature-based tracking, where the human skin color and a pattern developed on the patient's cap have been exploited as the features. The cap worn by the patient has been introduced as a strong feature by developing a pattern of straight lines on it and the Hough transform discussed in Chapter 4 has been shown to detect this pattern successfully. The performance evaluation of the new algorithm reveals that the algorithm performs reasonably well under major challenges, like multiple people in the frames and changing lighting conditions for the simulated data. The algorithm has been shown to have real time processing capability through extensive simulations. Therefore, this algorithm can be used to develop a complete

tracking system that can be installed in hospitals for automatic video EEG monitoring of patients without the need of human intervention.

Chapter 6

6.1 Concluding Remarks

In situations where the information about the target to be tracked is known a priori, feature-based tracking is expected to be a reliable technique for developing a tracking system. The primary contribution of this thesis has been to develop a feature-based algorithm for tracking epileptic patients in a hospital room for the purpose of automating the video EEG monitoring process.

First, the available features such as the patient's clothing, the cap on the head of the patient, the skin color and the elliptical shape of the human head have been identified and the possible techniques for identifying them in the video frames have been discussed. The texture of the clothes as a feature was not found to be suitable for practical applications and hence was excluded from further consideration after a discussion of its limitations.

Two existing algorithms for detecting the human skin color in images have been presented, and their limitations identified by testing them on video data. In order to overcome these limitations, a new algorithm for detecting the human skin color, that employs a combination of these two algorithms has been proposed. The improvement achieved in the detection performance has been supported by making suitable comparisons with the existing techniques.

An algorithm for identifying the elliptical shape of the head has been proposed. This along with skin color filter was then used for detecting human faces in video frames. This

human face detection technique was then tested for its detection performance, and found to be unacceptable due to the limitations of the ellipse detection algorithm.

Since, the skin color alone is not sufficient to track a patient, the cap on the patient's head has been developed into a feature by drawing a pattern of straight parallel lines on the cap. A pattern recognition technique based on the Hough transform for line detection has then been presented to identify this feature. The combination of the skin color filter and the Hough transform technique for detecting the pattern on the patient's head have been used to propose the final algorithm for identifying the location of the patient in a hospital room.

This proposed algorithm was tested for its performance under conditions such as varying illumination conditions, presence of multiple people in the room and real time processing capabilities. The performance of the algorithm has been found to be acceptable for the purpose of implementing a tracking system that can possibly be used commercially.

6.2 Future Work

A major limitation of the algorithm proposed in this thesis lies in its inability to locate the patient at night, when the lights are switched off and infrared cameras are used to capture the videos. Under these conditions, the skin color filter developed in this thesis is not expected to work as the videos obtained will not have any color information. Further research can be pursued in developing a color invariant feature that can possibly be used along with the electrode cap as a feature to track the patient in the videos obtained under infrared mode of the camera. Though the movements of the patient are limited under such

extremely low lighting conditions, making the algorithm illumination independent will eliminate the risk involved in losing any critical segment of the patient's movements.

In this thesis the problem of tracking epileptic patients has been addressed using only feature extraction techniques. The motion-based tracking techniques have not been considered in this thesis. Such techniques can possibly be used in conjunction with feature-based methods presented in this thesis to improve upon the performance of the tracking system. The possibility of developing an algorithm based on both the feature and motion-based algorithms should be studied. This may increase the real time computation efficiency of the algorithm.

References

[1] Douglas R. Nordli, Jr., "Usefulness of Video EEG monitoring," *Epilepsies*, vol. 47, pp. 26, Oct. 2006.

[2] Abubakr A, Wambacq I., "Seizures in the elderly: Video/EEG monitoring analysis," *Epilepsy Behavior*, vol. 7, no. 3, pp. 447-450, 2005.

[3] Liu Q., Sun M., Scheuer M.L., Sclabassi R.J., "Patient tracking for video/EEG monitoring based on change detection in DCT domain," *Proceedings of the Bioengineering Conference*, pp. 114-115, 2005.

[4] R. Plankers and P. Fua, "Tracking and Modeling People in Video Sequences," *Computer Vision and Image Understanding*, no.81, pp. 285-302, 2001.

[5] S. J. Mckenna, Y. Raja, and S. Gong, "Tracking Colour Objects using Adaptive Mixture Models," *Image and Vision Computing*, no.17, pp. 225-231, 1999.

[6] I. Haritaoglu, D. Hartwood, and L.S. Davis, "Real-Time Surveillance of People and their Activities," *IEEE Trans. Pattern Analysis and Machine Intelligence,* no. 22, pp. 809-830, 2000.

[7] David A. Forsyth, and Jean Ponce, "Computer Vision a Modern Approach," *Prentice Hall*, 2003.

[8] P.R Giaccone and G.A. Jones, "Segmentation of Global Motion using Temporal Probabilistic Classification," *Proceedings of British Machine Vision Conference*, University of Southampton, pp. 619-628, 1998.

[9] S.S. Beauchemin, and J.L. Barron, "The Computation of Optical Flow," *ACM Computing Surveys,* no. 27, pp. 433-467, 1995.

[10] C. Stiller, and J. Konrad, "Estimating Motion in Image Sequences," *IEEE Signal Proceeding Magazine*, pp. 70-91, July 1999.

[11] M. Biswas and T. Q. Nguyen, "A Novel De-interlacing Technique based on Phase Plane Correlation Motion Estimation," *Proceedings IEEE International Symposium on Circuits and Systems*, pp. 604-607, May 2003.

[12] J. Watkinson, "The Engineer's Guide to Motion Compensation," *Snell & Wilcox*, 1994.

[13] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, "Active Shape Models - their Taining and Application," *Computer Vision and Image Understanding*, no. 61, pp. 38–59, Jan. 1995.

[14] Michael Boyle, "The Effects of Capture Conditions on the CAMSHIFT Face Tracker," *University of Calgary, Department of Computer Science,* Alberta, Canada, Report 2001-691-14.

[15] Tianhorng Chang and C. C. Jay Kuo, "Texture Analysis and Classification with Tree-Structured Wavelet Transform," *IEEE Trans. on Image Processing*, vol. 2, no. 4, Oct. 1993.

[16] A.C. Bovik, "Analysis of Multichannel Narrow-band Filters for Image Texture Segmentation," *IEEE Trans. Signal Processing*, vol. 39, pp. 2025-2043, Sept. 1991.

[17] Larry S. Davis, "Polarograms: A New Tool for Image Texture analysis," *Pattern Recognition Society*, vol. 13, no. 3, pp. 219-223, 1981.

[18] Canny, J., "A Computational Approach to Edge Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 8, pp. 679-714, 1986.

[19] R. Gonzalez and R. Woods, "Digital Image Processing," *Addison Wesley*, pp. 414 – 428, 1992.

[20] R. Deriche, "Using canny's criteria to derive a recursively implemented optimal edge detector," *International Journal of Computer Vision*, vol. 1, no. 2, 1987.

[21] V.F. Leavers, "Shape Detection in Computer Vision Using the Hough Transform," *New York: Springer-Verlag*, 1992.

[22] H.K. Yuen, J. Illingworth, and J. Kittler, "Shape Using Volumetric Primitives," *Image and Vision Computing*, vol. 1, no. 7, pp. 31-37, 1989.

[23] Andrew Fitzgibbon, Maurizio Pilu, and Robert B. Fisher, "Direct Least Square Fitting of Ellipses," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, May 1999.

[24] T. Ellis, A. Abbood, and B. Brillault, "Ellipse Detection and Matching With Uncertainty," *Image and Vision Computing*, vol. 10, no. 2, pp. 271-276, 1992.

[25] F.L. Bookstein, "Fitting Conic Sections to Scattered Data," *Computer Graphics and Image Processing*, no. 9, pp. 56-71, 1979.

[26] S. Rao, "Optimization: Theory and Applications," *New York: Wiley Estern*, 2nd ed., 1984.

[27] M. Pilu, "Part-Based Grouping and Recogntion: A Model-Guided Approach," *Dept. Artificial Intelligence, Univ. of Edinburgh, PhD thesis*, Aug. 1996.

[28] David O'Mara, "Automated facial metrology," *University of Western Australia, PhD Dissertation,* Feb 2002.

[29] Jay Kapur, "Face detection in color images," *EE499 Capstone Design Project,* Spring 1997. http://www.geocities.com/jaykapur/face.html.

[30] Forsyth D. A. and Fleck, "M. M. Automatic detection of human nudes," *International Journal of Computer Vision*, vol. 32, issue 1, pp. 63-77, Aug. 1999.

[31] Satoh, S., Makamura, Y., and Kanade, T. Nam-it, "Naming and detecting faces in news videos," *IEEE Multimedia*, vol. 6, Issue 1, pp. 22-35, Jan-Mar 1999.

[32] Guillamet, D. and Vitria, J., "Skin segmentation using non linear Principal Component Analysis," *In Proceedings of 2nd Congres Catala d'intelligencia Artificial,* pp. 224-231, Girona, 1999.

[33] Graf, H.P., Cosatto, E., Gibbon, D., Kocheisen, M., and Petajan, "E. Multi-modal system for locating heads and faces," *In Proceedings of the 2nd IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 88-93, Killington, Vermont, USA, 14-16 Oct. 1996.

[34] Hunke, M., and Waibel, A., "Face locating and tracking for humancomputer interaction," *In Proceedings of the 28th Asilomar Conference on Signals, Systems & Computers*, Monterey, California, Nov. 1994.

[35] Koh, L. H., Ranganath, S., Lee, M. W., and Venkatesh, Y. V., "An integrated face detection and recognition system," *In Proceedings of the 10th International Conference on Image Analysis and Processing*, Venice, Italy, Sept. 1999.

[36] Yang, J., Lu, W. and Waibel, "A Skin-color modeling and adaptation," *Tech. Rep. CMU-CS-97-146, Carnegie Mellon University*, May 1997.

[37] Yang, J. and Waibel, "A real-time face tracker," *In Proceedings of the IEEE Workshop on Applications of Computer Vision*, pp. 142-147, Sarasota, Florida, USA.

[38] Gonzalez, R.C., and Woods, R.E. "Digital Image Processing," *Addison Wesley, New York, USA*, 1992.

[39] Jordao, L., Perrone, M., Costeira, J. P., and Santos-Victor, J. "Active face and feature tracking," *In Proceedings of the 10th International Conference on Image Analysis and Processing*, Venice, Italy, Sept. 1999.

[40] McKenna, S., Gong, S., and Raja, Y., "Face recognition in dynamic scenes," *In Proceedings of the British Machine Vision Conference*, Essex, 1997.

[41] Munkelt, O., Ridder, C., Hansel, D., and Hafner, W., "A model driven 3D image interpretation system applied to person detection in video images," *In Proceedings of the 14th International Conference on Pattern Recognition*, vol. 1, issue 16, pp. 70-73, Brisbane, Australia, 20 Aug. 1998.

[42] Zarit, B. D., Super, B. J., and Quek, F. K. H., "Comparison of five color models in skin pixel classification," *In Proceedings of the IEEE International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems* 1998.

[43] Terrillon, J., and Akamatsu, S., "Comparative performance of different chrominance spaces for color segmentation and detection of human faces in complex scene images," *In Vision Interface '99*, Trois-Rivieres, P.Q., Canada, 1999.

[44] Terrillon, J., Shirazi, M. N., Fukamachi, H., and Akamatsu, S. "Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images," *In Proceedings of the 4th IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 54-61, Grenoble, France, 30 March 2000.

[45] Kjeldson, R. and Kender, "J. Finding skin in color images," *In Proceedings of the 2nd IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 312-317, Killington, Vermont, USA, 14-16 Oct. 1996.

[46] Gevers, T., and Smeulders, A. W. M. PicToSeek, "Combining color shape invariant features for image retrieval," *IEEE Trans. on Image Processing,* vol. *9*, issue 1, pp. 102-119, Jan. 2000.

[47] Berens J and Finlayson G.D., "Log-opponent chromaticity coding of colour space," *In Proceedings of the 15th International Conference on Pattern Recognition*, pp. 1206-1211, Sept. 2000.

[48] C.A. Bouman, "Digital Color Imaging," Jan. 2006. http://www.ecn.purdue.edu/VISE/ee637/notes/pdf/Opponent.pdf

[49] G. Finlayson, G. Schaefer. "Hue that is Invariant to Brightness and Gamma," BMVC01, Session 3: Colour & Systems, 2001.

[50] PEIPA, The Pilot European Image Processing Archive, http://peipa.essex.ac.uk.

[51] P.V.C. Hough, "Machine Analysis of Bubble Chamber Pictures," *International Conference on High Energy Accelerators and Instrumentation*, CERN, pp. 554–556, 1959.

[52] PVC Hough, "Methods and means for recognizing complex patterns," U.S. Patent 3069654, 1962.

[53] Duda, R.O. and P.E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures," *Comm. ACM*, vol. 15, pp. 1-15, Jan. 1972.

[54] D. H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," *Pattern Recognition*, vol. 13, no. 2, pp. 111-122, 1981

[55] Jan R. Engelbrecht and Friedrich M. Wahl, "Polyhedral Object Recognition Using Hough- Space Features," *Pattern Recognition*, vol. 21, pp. 155-167, 1988.

[56] M. Kushnir, K. Abe, K. Matsumoto, "Recognition of hand-printed Herbew characters using features selected in Hough transform space," *Pattern Recognition*, vol. 18, pp. 103-113, 1985.

[57] Fang-Hsuan Cheng, Wen-Hsing Hsu and Mei-Ying Chen, "Recognition of handwritten Chinese characters by modified Hough transform techniques," *IEEE Transaction PAMI*, vol. 11, no. 4, pp. 429-439, 1989.

[58] Andrew John Lee, "Optical processing techniques for satellite pose determination," *Ph.D Dissertation, Carneige Mellon University*, 1989.

[59] S.N. Jayaramamurthy and R Jain, "An approach to the segmentation of textured dynamic scenes," *CVGIP*, 21, pp. 239-261, 1983.

[60] W. Poelzleitner, "A Hough transform method to segment images of wooden boards," *Int. Joint Conf. on Pattern Recognition*, pp. 262-264, 1986.

[61] C.J. Radford, "Optical flow fields in Hough transform space," *Pattern Recognition Letters*, vol. 4, pp. 293-305.

[62] R. Krishnapuram and D. Casasent, "Determination of three dimensional object location and orientation from range images," *IEEE Trans. PAMI*, vol.11, pp. 1158-1167, 1989.

[63] L.A. Fletcher and R. Kasturi, "A robust algorithm for text string separation from mixed text/graphics images," *IEEE Trans. PAMI*, vol. 10, pp. 910-918, 1988.

[64] Alexander Mitchell and Steven Mills, "Fencing foils using the hough transform," *Proceedings of Visual Information Engineering (VIE2005)*, pp. 59-64, April 2005.

[65] Badri Krishnan (for the LIGO Scientific Collaboration) "Wide parameter search for isolated pulsars using the Hough transform," 2005 *Classical and Quantum Graity,* vol. 22, S1265-S1275.

[66] D.A. González, C. Ibarra-Castanedo, J.M. López-Higuera, X. Maldague, "New Algorithm based on the Hough Transform for the Analysis of Pulsed Thermographic Sequences," *Proceedings of Vth International Workshop*, Advances in Signal Processing for Non Destructive Evaluation of Materials.

[67] C. H. Daouk, L. A. El-Esber, F. D. Kammoun and M. A. Al Alaoui, "Iris Recognition," *IEEE ISSPIT 2002*, pp. 558-562, Marrakesh.

[68] Ilya Lavrik and Brani Vidakovic, "Linear feature identification and inference in Nano-Sacle Images," *A technical report, Georgia Tech University*, 2005.