Automatic Red Eye Detection & Correction

By:

Shashank Mediratta: 051074

Ankita Sood: 051077

Rahul Shahi: 051088

May 2009

Submitted in partial fulfillment of the Degree of Bachelor of Technology Engineering

Project Supervisor: Dr Vinay Kumar

DEPARTMENT OF ELECTRONICS AND COMMUNICATION

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY-WAKNAGHAT

CERTIFICATE

This is to certify that the work entitled," **AUTOMATIC RED EYE DETECTION AND CORRECTION**" submitted by SHASHANK MENDIRATTA, ANKITA SOOD, and RAHUL SHAHI in partial fulfillment for the award of degree of Bachelor of Technology in Electronics and Communication of the Jaypee University of Information technology has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

(Signature)
Dr.Vinay Kumar
(Project supervisor)
Department Of Electronics and
Communication Engineering,
Jaypee University of Information Technology,
Waknaghat, Solan-173215, India

(Signature)
Dr. S.V Bhooshan
(Head Of Department)
Department Of Electronics and
Communication Engineering
Jaypee University of
Information Technology,
Waknaghat,Solan-173215,India

Acknowledgement

Apart from the efforts, the success of any project depends largely on the encouragement and guidelines of many others. Therefore we take the opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We would also like to show our greatest appreciation to our project guide **Dr. Vinay Kumar** without whose able guidance, tremendous support and continuous motivation the project work could not be carried out satisfactorily.

Thanks should also be given to the lab assistants for all their much appreciated time and assistance during the project.

Lastly, we would like to show our high gratitude to the entire faculty for showing us different methodologies used in our project.

Date: 08/05/2009

Shashank Mendiratta

Shashank Mendiratta

(051074)

Rahul Shahi

Rahul Shahi

(051088)

Ankita

Ankita Sood

(051077)

Table of Contents

List of Figures

Abstract

Caused by light reflected off the subject's retina, red-eye is a troublesome problem in consumer photography. Correction of red eyes without user intervention is an important task. Solutions are available, but not to the extent that will be of benefit to people. In order to develop a simple and efficient software solution, we have proposed a novel technique for automatic red eye detection and correction in color images. The algorithm first detects face regions in the image using a skin color model in the RGB, HSV and YCrCb color space. Then, red eye candidates are extracted within these face regions based on threshold values. Finally, the original colour of the pupil is restored automatically in the image applying the correction algorithm consisting of various tests for spotting the red-eye effect efficiently. Hence a fully corrected image is achieved. The proposed method is simple and fast since it needs no template matching step for face verification, or a pre-defined database as in the cases of most of the other proposed methods. It is robust because it can deal with rotation of the face in the image.

Chapter 1 INTRODUCTION

Humans can use vision to identify objects quickly and accurately. Computer Vision seeks to emulate human vision by analyzing digital image inputs. That fact that the world is a three-dimensional environment and a digital image is just a two-dimensional representation, is just one of the many problems that complicate Computer Vision.

1.1 The Red-Eye Problem

Everyday millions of photographs are taken from all around the world. Photography is a profession and in some cases it is a hobby. However, to most people it is a means to capture and relive memories. Great advances have taken place in the world of photography, with the transition to colour being one of the most prominent. In spite of all the advancements, one problem is still very much evident. The problem is the appearance of an unnatural red hue around a pupil of the human eyes that can occur in photographs. The occurrence of this phenomenon, known as red-eye, is a large and widespread problem. It has been and is still one area that camera and imaging specialists are having difficulties in dealing with.

It is generally caused on using a flash in a dark environment. When a flash is needed to illuminate a subject, the ambient illumination is generally low, and the subject's pupils will be dilated. When light is reflected off the subject's retina, a trace of red appears in the eyes of the subject. If the angle formed by the flash, the back of the subject's eye, and the camera lens is less than 2.5 degree, red-eye will occur, as also shown in the figure below.

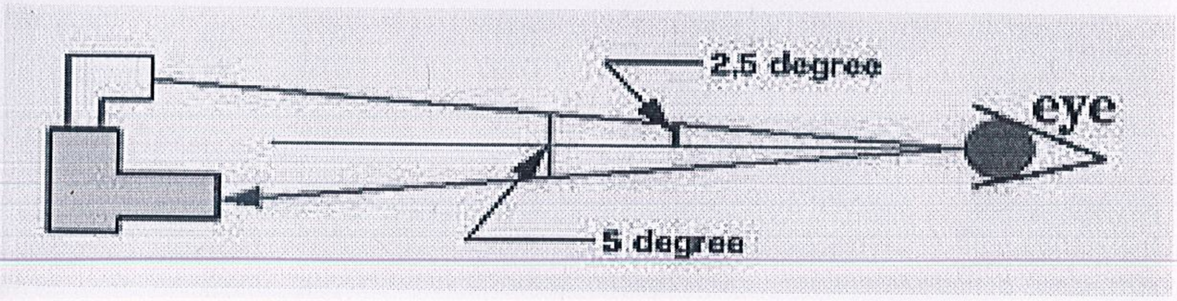


Figure 1 : Angle of Reflection

One way to overcome the red-eye effect is to move closer to your subject so that the angle of reflection is larger than 2.5 degree.

The more open the pupils are, the more prominently red-eye effect appears in the photos. Red eye is more pronounced in people with light eye color. It is also more pronounced in people with blond or light-red hair and in children.

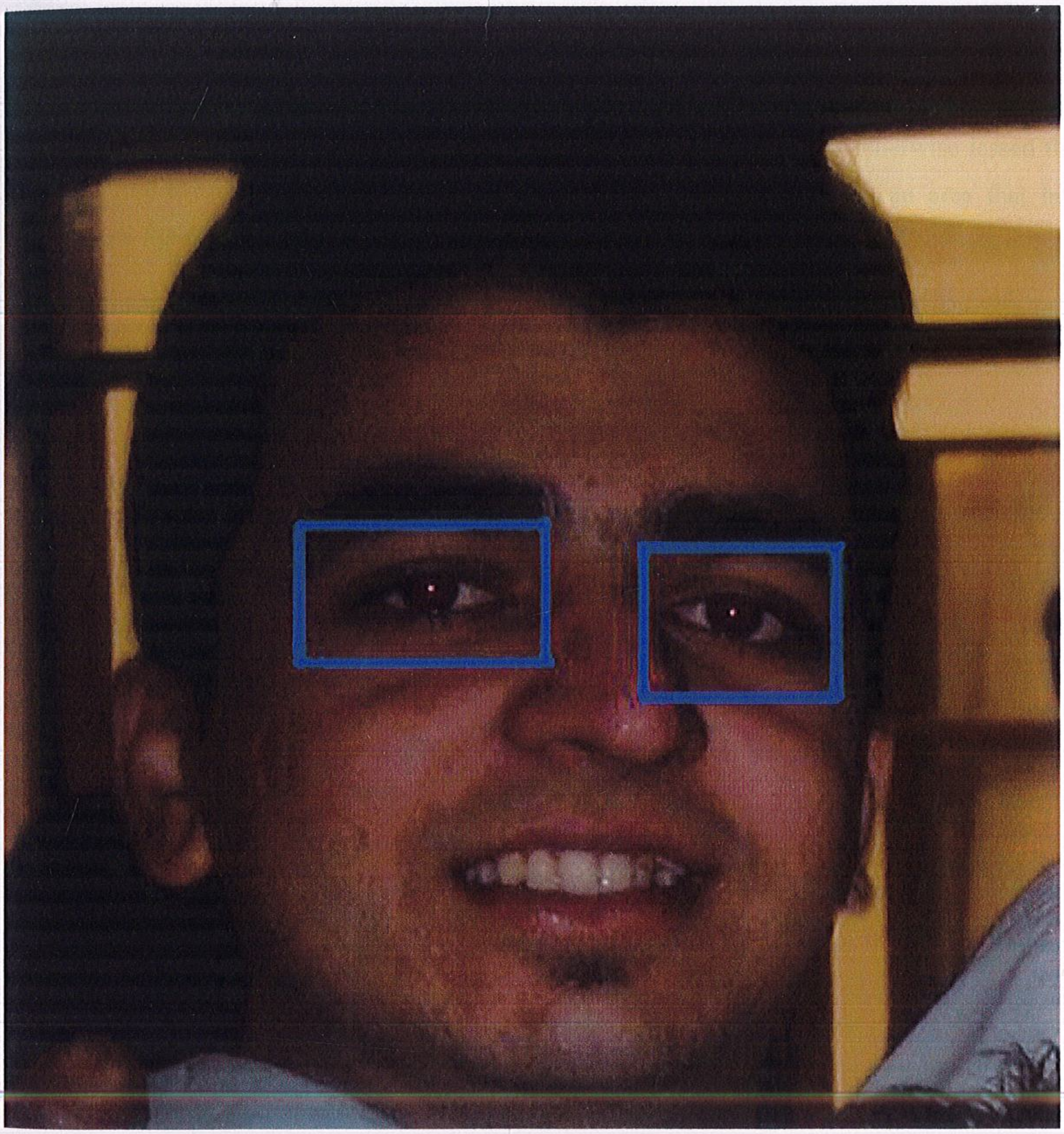


Figure 2: Example of red-eye defect

Red-eye has become increasingly more prevalent in recent years particularly with the advent of digital cameras. Digital cameras do not require any film, as a result the cameras are easier to maintain and can be built to very compact dimensions. Also the facility to transfer images directly to a computer has made digital cameras very popular. It is not surprising that there has been a huge growth in digital camera sales in recent years.

Digital cameras do not suffer the problem of running out of film. The cameras do not incur the cost of having to replace film or the cost to develop the pictures in order to see them, as images can simply be viewed on the computer. People as a result are less hesitant in taking lots of photos and also less inclined to take the necessary precautions that would otherwise lessen the effect of red-eye. These precautions are as simple as taking the photo in an area that has appropriate lighting, or making use of the camera's pre-flash.

The diminishing size of the camera has contributed significantly to the problem. As the size of a camera shrinks, so to does the distance between the flash and the lens, decreasing the angle of reflection. In short red-eye is a large and widespread problem.

1.2 Existing Solutions

Solutions are available, but not to the extent that will be of benefit to people.

- A popular solution for cameras with a small lens-to flash distance is the use of one or more pre-exposure flashes. A pre-exposure flash will contract the subject's pupil diameter, thus reducing the chance that light reflected off the retina will reach the lens. The drawback of this approach is the cost in power. A flash consumes more power than any other aspect of image capture; additional flashes further reduce the battery life. Furthermore, this flash will sometimes reduce, but not eliminate, the red-eye artifacts. Thus it proves out to be more of a red-eye reduction solution rather than correcting the red-eye completely.

- One option is to scan or upload the photo on the computer and use any image-editing software to correct red-eye. Many such software have a red eye reduction feature, like Adobe Photoshop, Corel Photo-Paint and Picasa can be used to upload, but the problem with them is that they need user intervention for correction. Also, this is a very time consuming process and people would ultimately ignore this option.

- The next type of approach was the design of simple algorithms, which could be incorporated in some of the more popular image editing software. For these plug-ins, the user only has to highlight the red-eye in a photo. The algorithm would then attempt to re-colour the eye. The problem in this case was that the software for which these plug-ins were designed were often very expensive, and people would be reluctant to pay for something that only does half the task.
- Hardware has become available. This hardware is purpose built and incorporates very sophisticated algorithms. The devices are simply too expensive and not aimed at the home market. The hardware is available for use mostly at digital print kiosks. An example is the Automatic Photo Machine (APM) from Lucidiom Inc.

There is obviously a huge market potential for the sale of software that can automatically correct red-eye but at the same time be affordable and easy to use. To date, there have been few results published specific to automatic red-eye detection. After red-eye is successfully identified in photos, a number of things can be done to automatically correct it, but very few of the existing algorithms provide a proper solution to perform the correction process. Often non-free nature of the existing algorithm gives the impression that they are complicated and proprietary. Thus, the development of automatic red eyes correction method with good detection quality, independent from face orientation, with relatively low computational complexity and memory requirements is an essential task.

1.3 The Aim

- The aim of this project is to use computer vision techniques to automatically detect and remove the effects of red-eye.

- To develop a system that is easy to use, can easily be adapted, modified, reproduced and even improved upon.
- The ultimate goal was to create a widely accessible piece of software that will work and benefit many users. The software is designed avoid user interaction. The user runs the software, opens the picture to be corrected and then presses the button to begin the process. The process is fully automatic.

1.4 Overview

The image will be systematically broken down and analyzed by a series of algorithms to determine the pixels that are representative of red eyes. After this another algorithm will be applied that will attempt to modify the pixels of the eye to appear more natural in colour. The automatic algorithm must correctly identify all pixels included in the red-eye while not incorrectly classifying other regions as red-eye, such as lips or a red coat.

Red eye tends to occur more often in pictures taken without precautions. Frequently these could be spontaneous pictures in which all subjects are not fully in frame. This is the primary reason to adopt a more general approach towards the entire process. This approach is based on the assumption that an eye is surrounded by skin.

1.5 Motivation

The algorithms used in this project are designed to be very general. The idea behind this is to allow the system to be sensitive enough to detect instances of red-eye, which can occur in the background of an image or in subjects that are simply at a relatively large distance from the camera. In cases such as these, the reds can be pale in colour or even off red. As a result there is a chance that there will be several false detections of red-eye instances at this point. These false detects could represent anything from a person's lips to red skin blemishes.

Chapter 2 AUTOMTING THE PROCESS

In order to implement a system that can automatically correct incidences of red eye, a series of logical steps would have to be developed. Two approaches were immediately apparent. The first approach was to simply scan through the image and analyze, with a series of algorithms, all pixels suspected to be incidences of red eye. This would mean every part of the image would have to be thoroughly examined in a recursive manner due to the fact that the probability of false detection due to the uncertainty of the background and the skin tone in the image is quite high. The concept was immediately abandoned in lieu of a more elegant and systematic approach. The idea was to develop a system which has a low probability of false red eye detection. Therefore, the system was divided into 3 stages:

Stage1-> Face Detection

Stage 2-> Red Eye Detection in the face region

Stage3-> Red Eye Correction in the eye region

2.1 Sequence of Events

1. Image opened.
2. Skin segmenting algorithm applied.
3. All skin-like areas will be noted.
4. Each of the skin like areas will be searched. The boundaries of the skin regions will be marked.
5. Only the areas that are fully bounded by skin areas will be retained.
6. The maximum presence of skin coloured pixels will denote the presence of a face, as we are considering images with single face only.
7. The facial region is cropped to concentrate more on the eyes.
8. Presence of red eye is detected.
9. The correction algorithm is applied.
10. An altered image is returned, if it contained red eyes.

Chapter 3 FACE DETECTION

We now give a definition of face detection: Given an arbitrary image, the goal of face detection is to determine whether or not there are any faces in the image and, if present, return the image location and extent of each face.

The challenges associated with face detection can be attributed to the following factors:

- Pose. The images of a face vary due to the relative camera-face pose (frontal, 45 degree, profile, upside down), and some facial features such as an eye or the nose may become partially or wholly occluded.

- Presence or absence of structural components. Facial features such as beards, mustaches, and glasses may or may not be present and there is a great deal of variability among these components including shape, color, and size.

- Facial expression. The appearances of faces are directly affected by a person's facial expression.

- Occlusion. Faces may be partially occluded by other objects. In an image with a group of people, some faces may partially occlude other faces.

- Image orientation. Face images directly vary for different rotations about the camera's optical axis.

- Imaging conditions. When the image is formed, factors such as lighting (spectra, source distribution and intensity) and camera characteristics (sensor response, lenses) affect the appearance of a face.

There are many closely related problems of face detection. Face localization aims to determine the image position of a single face; this is a simplified detection problem with the assumption that an input image contains only one face.

Given a single face image, the main concern of face detection is to identify all image regions which contain a face regardless of its orientation, background and lighting conditions. Such task is tricky since faces can have a vast assortment in terms of, shape, color, size, or texture. At present time a lot of automatic approaches involve detecting faces in an image and, subsequently, detecting eyes within each detected face. But most of the face detection algorithms are only able to detect faces that are oriented in upright frontal view; these approaches cannot detect faces that are rotated in-plane or out-of plane with respect to the image plane, also cannot detect faces in case when only part of face is visible. As a result using threshold to separate skin region from an image for face detection was chosen in this algorithm.

3.1 Skin color classification

While different ethnic groups have different levels of melanin and pigmentation, the range of colors that human facial skin takes on is clearly a subspace of the total color space. With the assumption of a typical photographic scenario, it would be clearly wise to take advantage of face-color correlations to limit our face search to areas of an input image that have at least the correct color components. In pursuing this goal, we looked at three color spaces that have been reported to be useful in the literature, RGB and HSI spaces, as well as YCrCb. Below we will briefly describe each color space and how it was used for skin color classification.

3.2 Skin-based segmentation

Human skin color is a very efficient feature for face detection. Although different people may have different skin color, several studies have shown that the major difference lies largely between their intensity rather than their chrominance.

There has been much research entailing skin- based segmentation, which incidentally is the basis for some face-detecting algorithms. Most of the algorithms use a range of colour values to define skin colour.

3.3 Segmentation Rules

Segmenting an image is the process by which a computer attempts to separate objects within an image from the background as well as from other objects. The segmentation rules are the rules that will determine the formation of regions. The segmentation rules are based on analyzing the colour and edge properties of a region.

The advantages of skin-based segmentation is the loss of the sizable overhead of first determining faces within an image coupled with the fact that even faces that are occluded, profiled, partially out of frame or simply not recognized, will all be further processed.

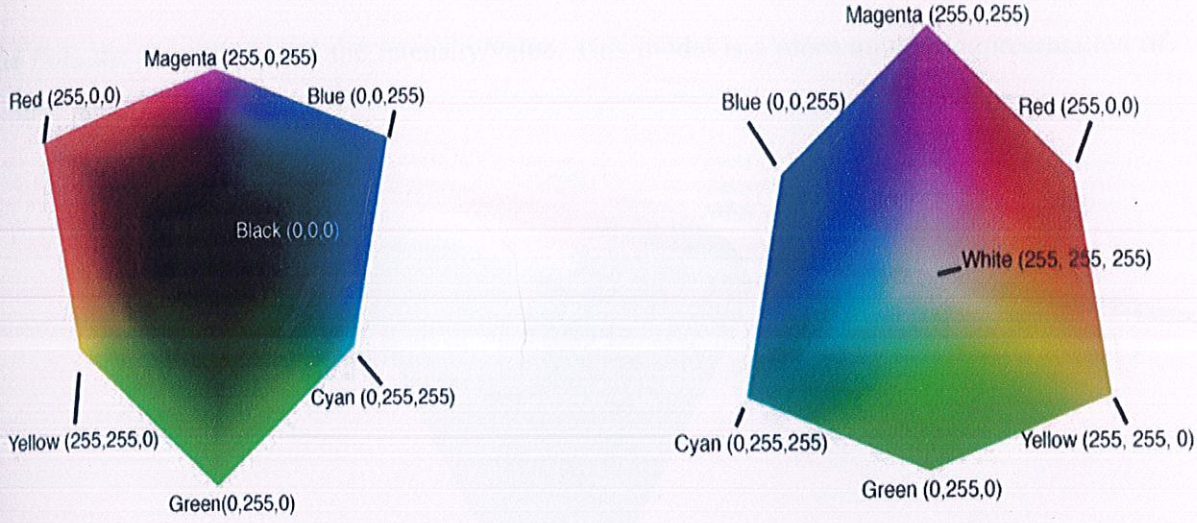
This project bases its analysis of skin colour using the RGB, HIS/HSV and YCrCb colour models. The values are arbitrary, unique to this project and obtained by extensive colour sampling from flesh tone colours from digital photos.

Using the HSI model accounted for a greater range of skin tones. The problem however was the HSI model made it difficult to constrain the colours to specifically flesh-like tones. The solution adopted was to use the HSI model to allow for a large range and the RGB values to constrain over dominance of specific colours e.g. too 'red'. Skin colour seems to occupy a small range of hue values.

3.4 Colour Models

3.4.1 RGB

The RBG colour model is the most widely recognized colour model. It comprises of three components namely the red, green and blue colour channels. The RGB model is used to a great extent in solving computer vision problems, but is better known for colour representation in the displays of television sets and monitors. The value of a colour by this model is best described as being a vector in three-space where red, green and blue represent the axis. Colour is thus a result of the combination of the red, green and blue components. The origin of the cube (0, 0, and 0) represents pure black while its polar opposite (255, 255, and 255) represents pure white.



RGB cube from perspective of the black corner

RGB cude from the perspective of the white corner

Figure 3:RGB Cube

From studying and experimenting with various thresholds in RGB space, we found that the following rule worked well in removing some unnecessary pixels:

$$0.836G - 14 < B < 0.836G + 44 \Rightarrow \text{Skin}$$

&

$$0.79G - 67 < B < 0.78G + 42 \Rightarrow \text{Skin}$$

3.4.2 HSI/HSV

While RGB may be the most commonly used basis for color descriptions, it has the negative aspect that each of the coordinates (red, green, and blue) is subject to luminance effects from the lighting intensity of the environment, an aspect which does not necessarily provide relevant information about whether a particular image "patch" is skin or not skin. The HSI color space, however, is much more intuitive and provides color information in a manner more in line how humans think of colors and how artists typically mix colors information. The model is best presented as a colour cone or cylinder The HSI model comprises of three components. These are the hue, the saturation, and the intensity/value. This model is a more intuitive representation of colour.

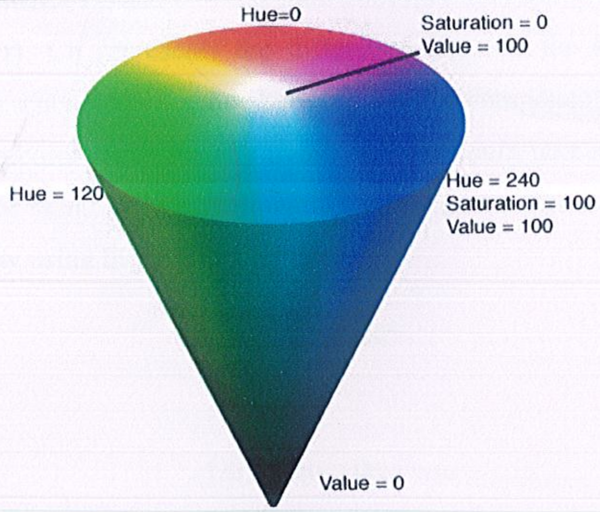


Figure 4: The HSV colour cone

The hue represents a person's perception of a colour, for example green or orange. Hue changes as one moves around the cone. Saturation is a measure of a colour's dilution by white light. This provides us with light or dark shades of a colour. Saturation increases from the centre of the cone to the outside. Finally, the intensity is a measure of the brightness of the colour. Intensity increases along the cone's vertical axis. The great advantage of using this colour model is the fact that it separates intensity from the colour components unlike the RGB model that couples intensity with colour information. It is the first two, H and S that will provide us with useful discriminating information regarding skin.

The threshold values used by us to derive the following rule used in our face skin detection block:

$$19 < H < 240 \Rightarrow \text{Not Skin;}$$

Otherwise we consider it as skin.

3.4.3 YCrCb Color Model

The YCrCb color space is widely used for digital video. In this format, luminance information is stored as a single component (Y), and chrominance information is stored as two color-difference components (Cb and Cr). Cb represents the difference between the blue component and a reference value. Cr represents the difference between the red component and a reference value. We analyzed the YCrCb color space for any trends that we could take advantage of to remove areas that are likely to be skin. After experimenting with various thresholds, we found that the best results were found by using the following rule:

$$140 < Cr < 165$$

$$\Rightarrow \text{Skin}$$

$$140 < Cb < 195$$

Using these three color models we localize the skin color of the image.

Original Image | Skin Color separated image

Figure 5: Skin region separated from the non skin regions

3.5 Erosion & Dilation

The basic morphological operations: - erosion and dilation produce contrasting results when applied to either grayscale or binary images. Erosion shrinks image objects while dilation expands them. The specific actions of each operation are covered in the following sections.

3.5.1 Characteristics of Erosion

- Erosion generally decreases the sizes of objects and removes small anomalies by subtracting objects with a radius smaller than the structuring element.
- With grayscale images, erosion reduces the brightness (and therefore the size) of bright objects on a dark background by taking the neighborhood minimum when passing the structuring element over the image.
- With binary images, erosion completely removes objects smaller than the structuring element and removes perimeter pixels from larger image objects.

3.5.2 Characteristics of Dilation

- Dilation generally increases the sizes of objects, filling in holes and broken areas, and connecting areas that are separated by spaces smaller than the size of the structuring element.
- With grayscale images, dilation increases the brightness of objects by taking the neighborhood maximum when passing the structuring element over the image.
- With binary images, dilation connects areas that are separated by spaces smaller than the structuring element and adds pixels to the perimeter of each image object.

Once the skin color is localized the resultant image is converted to a binary image so that the processing is faster. Using a 5x5 window, the noises in this black and white image are improved. The process of erosion & dilation is carried out to remove unwanted pixels which were falsely detected as skin color pixels by moving the window over the image matrix and if the number of white pixels is more than a fixed threshold of 10 pixels the whole window is made black.

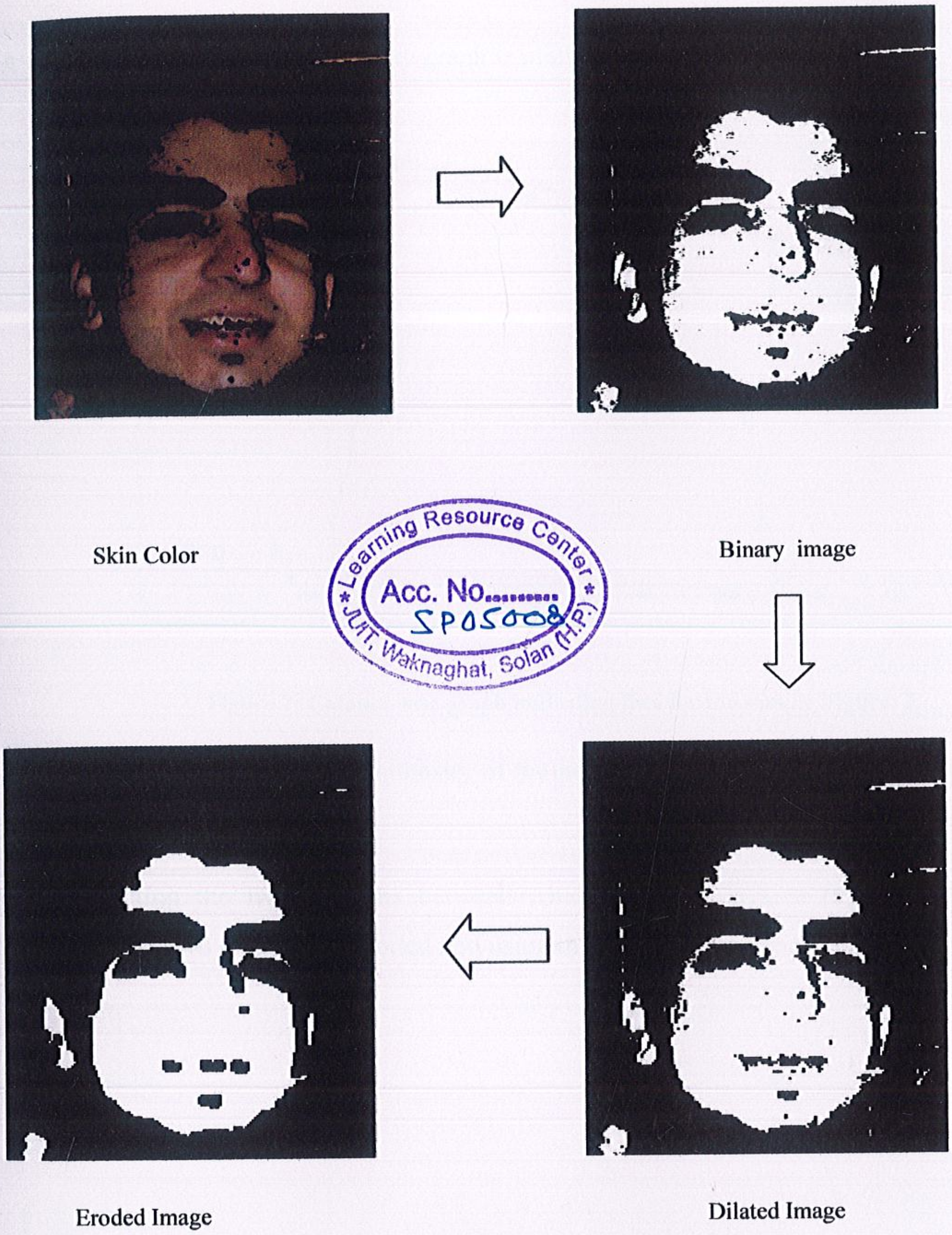


Figure 6: The Whole Process of Dilation and Erosion

The face detection involves taking the detected skin areas. Hence face localization based on the occurrence of white region in the eroded and diluted image is accomplished. This process has been developed based on the assumption that the image being processed has a single face with minimal skin exposure.

A column wise white pixel intensity graph is plotted

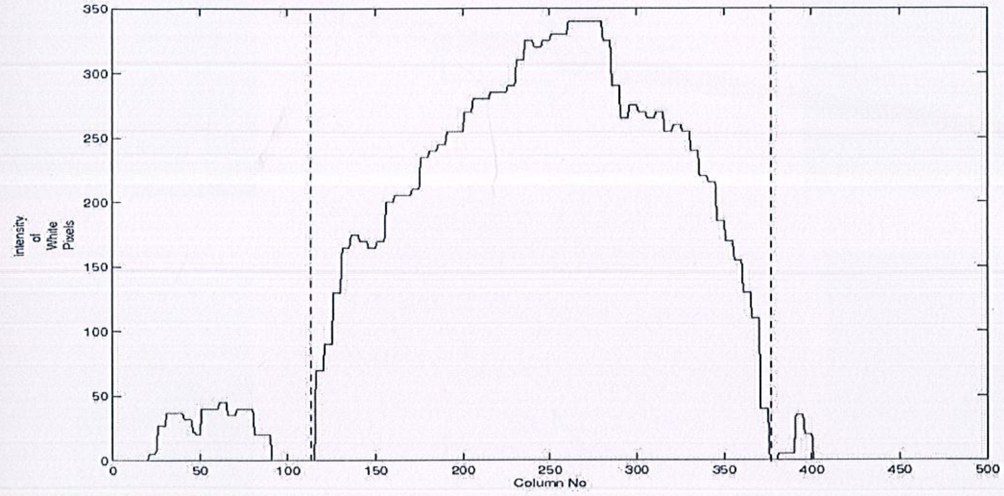


Figure 7: Column wise graph indicating face for the sample Figure: 2

This figure shows column wise intensity of the image and the biggest Gaussian curve shows the presence of the face.

After detecting the two columns between which the face lays, a row-wise intensity graph between these two columns is plotted and using this information the face is localized.

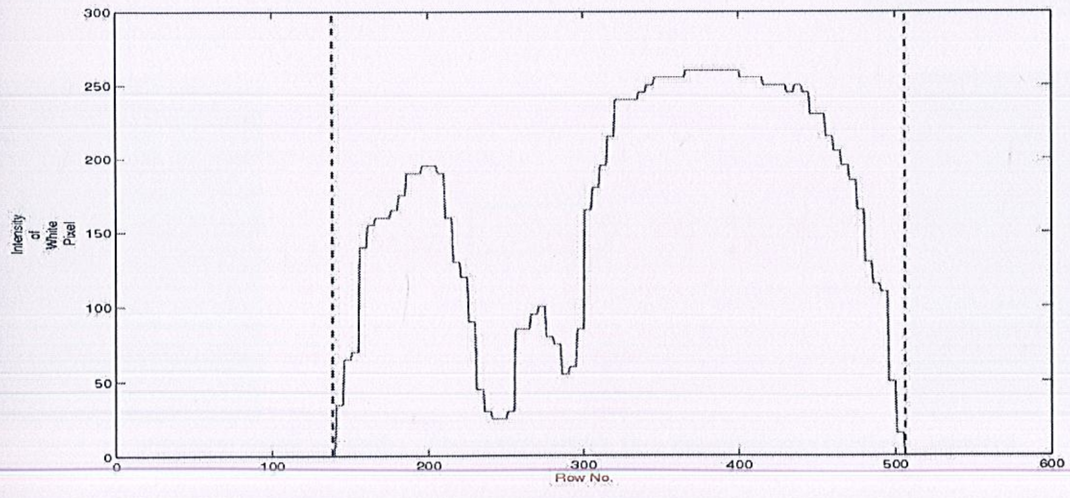


Figure 8: Row wise graph indicating face for the sample Figure: 2

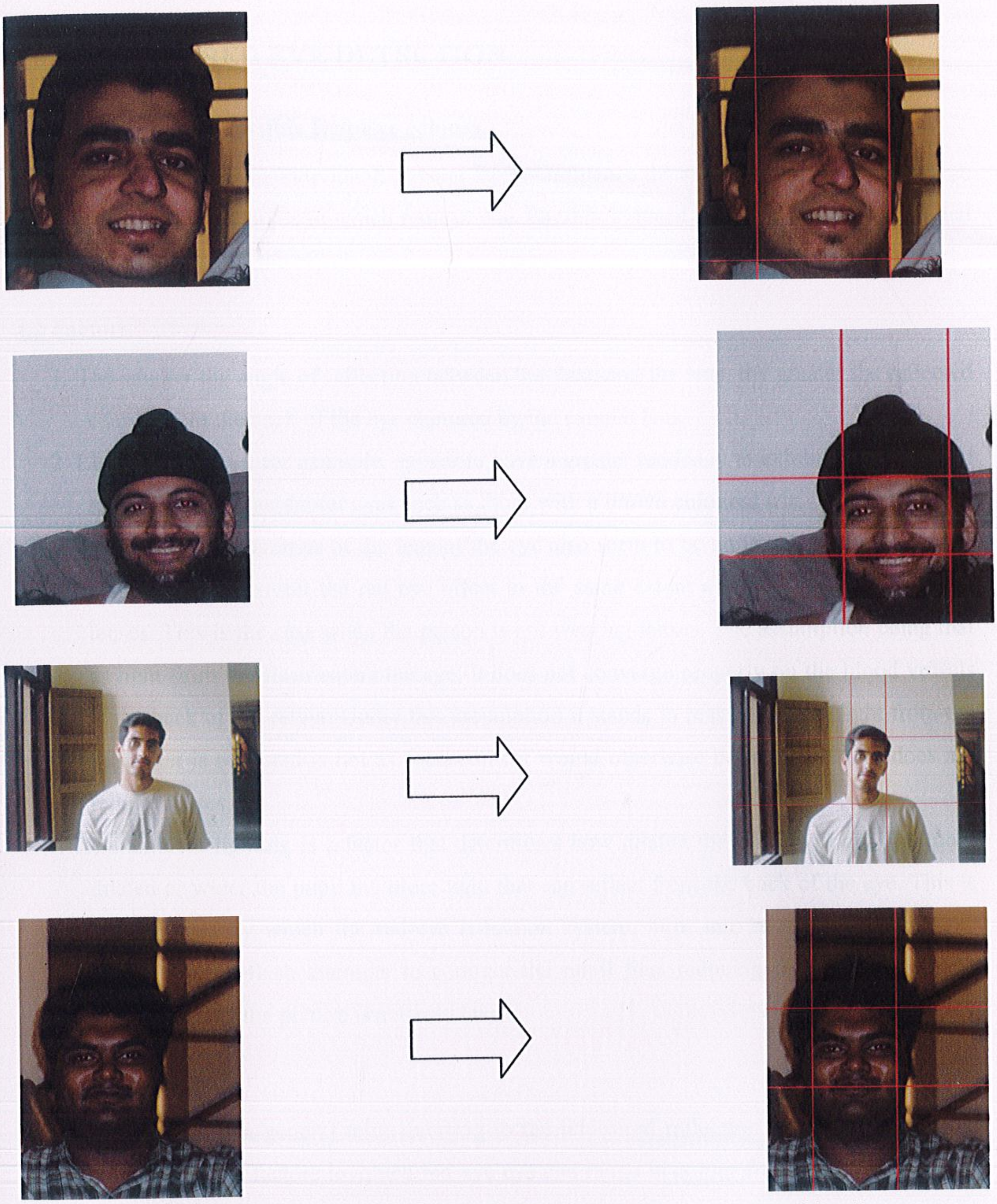


Figure 9: Some examples of images in which face detection was perfectly achieved

Chapter 4 RED EYE DETECTION

4.1 Quantifying Permissible Red Eye colours.

Every eye is unique to the individual. Several factors influence the severity of the red-eye effect. Examination of many photos in which red-eye was forcibly induced revealed several factors that influence the severity of the effect.

4.2 Factors

1. The smaller the angle of reflection between the flash and the lens, the greater the reflected energy from the back of the eye captured by the camera lens.
2. Lighter blue eyes, for example, appear to have a greater tendency to exhibit a more severe red eye effect, than darker eyes such as those with a brown coloured iris.
3. The refractive properties of the lens of the eye also seem to be an issue. Eyes that require glasses do not exhibit the red eye effect to the same extent as eyes that do not require lenses. This is the case when the person is not wearing lenses. The assumption being that as light from the flash enters the eye; it does not converge properly on the blood vessels at the back of the retina. Under this assumption it stands to reason that the light from the flash that is reflected is not as focused as it would otherwise be in an eye that does not require lenses.
4. The ambient lighting is a factor that determines how dilated the pupil will be. The more dilated or wider the pupil the more light that can reflect from the back of the eye. This is the principle by which the red-eye reduction system, built into most modern cameras work. The pre-flash attempts to contract the pupil thus reducing the amount of light reflected when the picture is actually taken.

4.3 Colour Ranges

The red-eye effect is a general term referring to the undesired reflection from the back of the eye. Depending on the factors involved red-eye eye can range in colour from shades of pink, to glowing bright red to shades of purple. It is dark coloured eyes that exhibit the purple colour. As with the algorithm for selecting skin-like pixels, both the RGB and HSI colour models are used for selecting the red-eye pixels.

The range of values used to segment the red eye pixels in the regions were selected arbitrarily based on empirical measurements on sampled photos. The values allow for as large a range of values as possible.

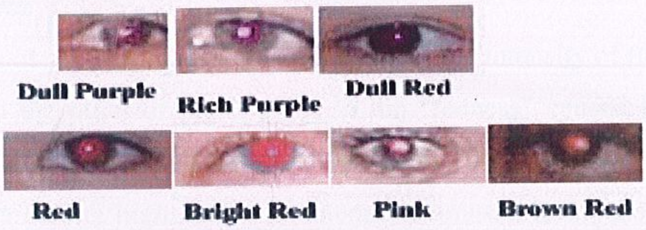


Figure 10: Examples of varying degrees of red-eye.

4.4 Probable location of Red-Eyes

Eyes are located on a persons face. An obvious solution to locating eyes or indeed red-eyes would be to first locate faces within an image.

Finding possible red-eyes is a five-step process.

1. Segmentation of the skin.
2. Extract the regions.
3. Map the skin boundary in the region.
4. Search for high red energy within skin boundaries of the region.
5. Extract the candidate red-eye regions.

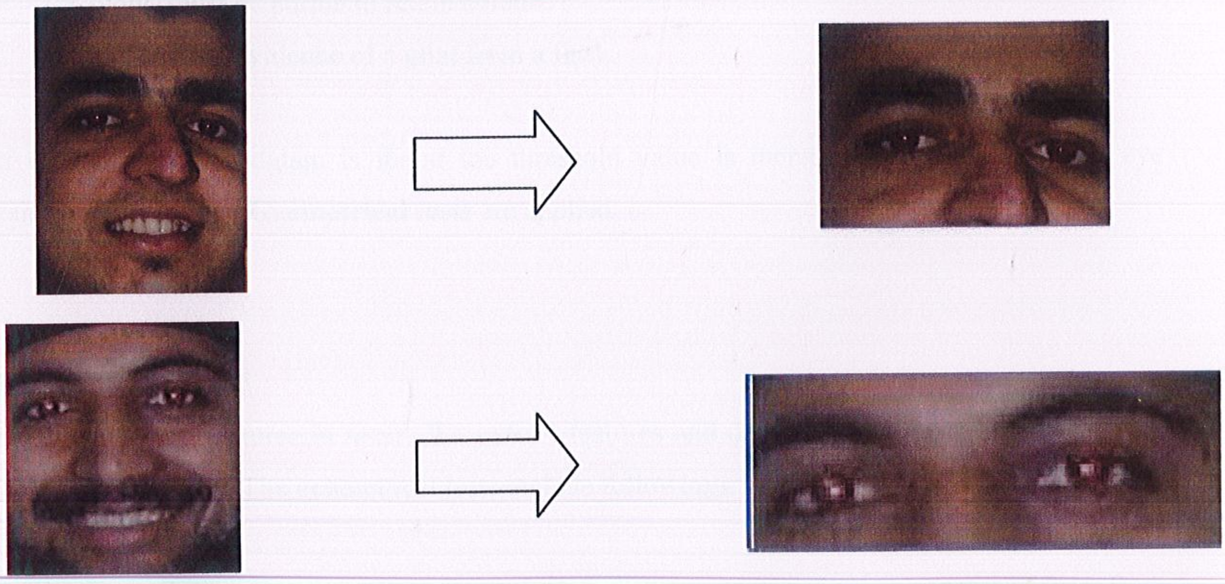


Figure 11: The Cropped Facial regions to avoid errors

4.5. Red Eye Detection

On close examination of several images, it is noticed that the intensity of the RGB components is the maximum around the pupil of the eye due to the "redness" caused by the red eye defect. Taking this into consideration, a high threshold value of 210 has been set by analyzing a variety of images. The pixels having high values of intensity common in all the three [red, green and blue] color planes, the red eye can be easily separated out from the rest of the face. To shun errors in the detection, the facial region is cropped by 30% from the top and by 10% from the bottom **(Figure 11)** before applying the algorithm for red eye detection. This way the eyes can be more easily concentrated upon.

It should be pointed out that the procedure described as of now primarily uses feature analysis. Up until now it has been removing the regions that do not adhere to the most basic features of a red-eye:

1- Location must be in a skin region.

2- Colour, must be purple to red in colour

3- Must contain evidence of a glint from a flash.

If only one eye candidate is found the threshold value is increased. If more than two eye candidates are found **Geometrical tests** are applied.

4.5.1 Detection Tests

Eyes have some features in faces. We extract features and design special tests to verify which candidates are eyes. The geometrical tests include followings:

1) **Eyes-Centre Distance Test**: We calculate distance of eyes from the centre of face. Both the distance between eye and the centre of face are almost the same and must not exceed

each other by 30%. Our observation shows the two distances are found very closely matched.

2) **Eye Pair Distance Test**: The distance between the eye pair must be more than Eyes-Centre Distance.

3) **Eye Angle Test**: According to the structure of face, two eyes cannot be located in one side of face. In this test we examine eye pair candidates to be in two side of face.

4) **Eye Shape Test**: As eye shape is circular, we test candidates not to be so thin and long. For this purpose we compute roundness ratio. The two selected eyes must be more than 0.7 in roundness ratio. Also those thin and long candidates would be discarded in this test.

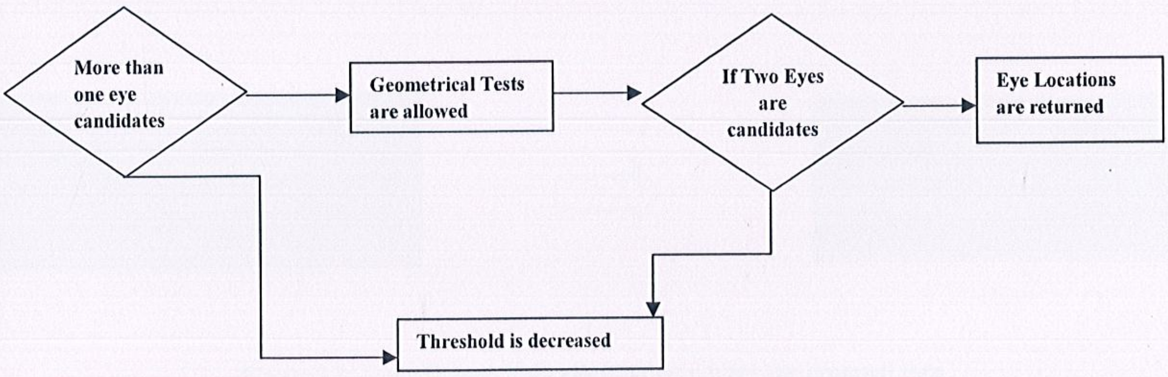


Figure 12: Flowchart for eye detection

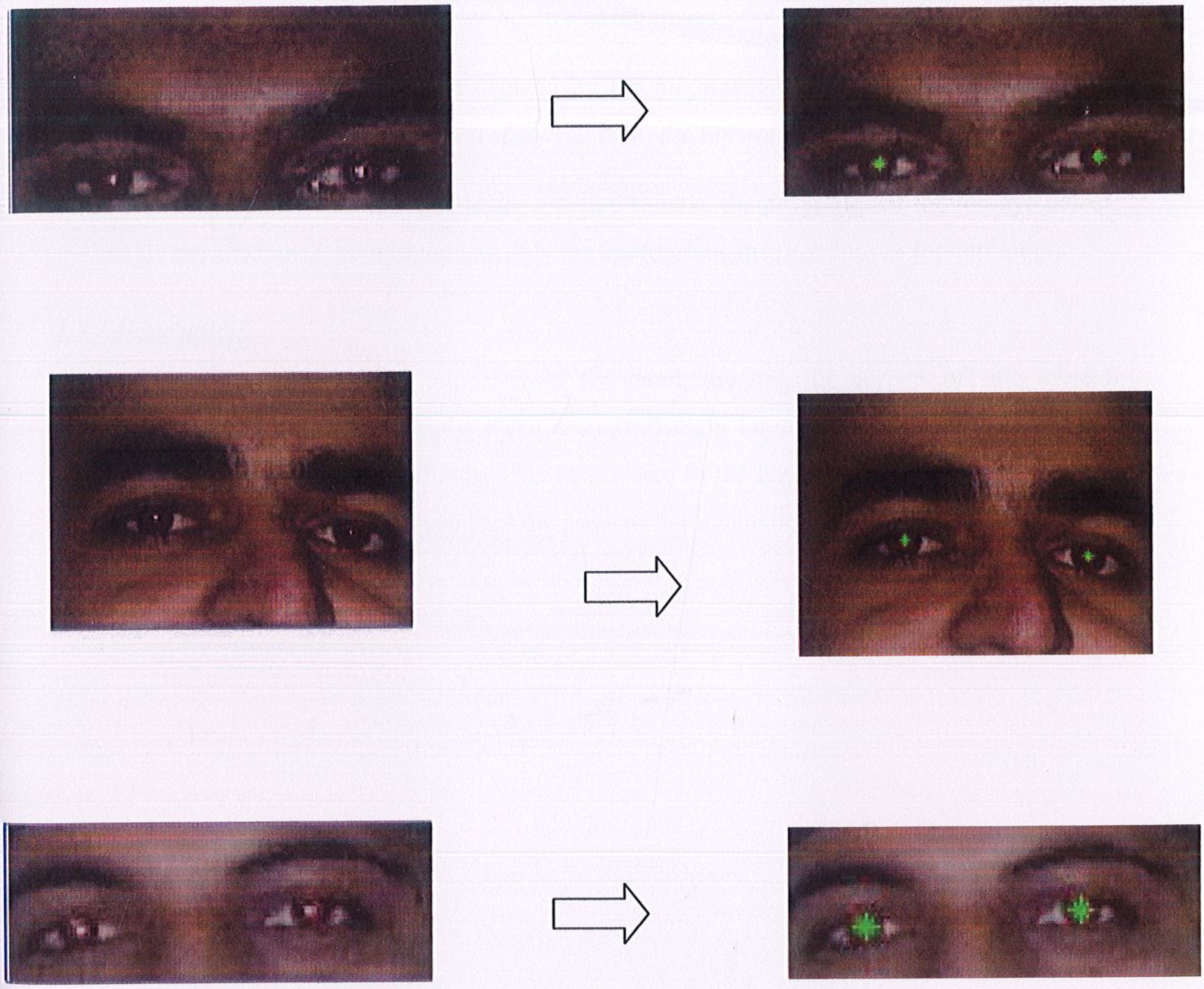


Figure 13: Some figures indicating red-eye from the cropped face

If in case more than two spots in an image were detected as red-eyes, a certain self-designed algorithm was followed, mentioned below:

- The image was divided into two planes, viz. the left plane and the right plane. If more than one red-eye defect has been spotted in the image only one is considered.

- The one to be considered is decided by the alignment of the detected red-eye spots. Maximum two red-eye detected spots can lie in the horizontal plane.

Thus, if by any chance multiple pixels are reported to have the properties of the red-eye defect, the falsely detected ones are omitted and only the appropriate ones are chosen for correction.

4.5.2 Description

An eye, on a superficial level comprises of the pupil, the iris, the cornea and the eye-lid. Compared to other parts of the body, there is a surprisingly large amount of detail and indeed colour variation over such a small area. This fact is used as the basis for building the description of the eye. The description of the eye uses the pupil as the foundation.

Chapter 5 THE CORRECTION PROCESS

This is the final operation to be performed. There should be relatively few regions, if any, remaining in the queue at this stage. The correction process must correct the colouration of the eyes in a realistic fashion. Any type of redrawing has the potential to appear more unnatural than the already unnatural looking red-eye.

5.1 Red Eye Correction

Once a candidate red-eye location is detected, the artifact can be corrected by removing the color of all "excessively red" pixels within an eye-sized neighborhood. A pixel at a given location is considered excessively red if redness at that location exceeds a certain threshold. Handling the artifact in this manner will remove the redness, but can also introduce other undesirable artifacts. If pixels are desaturated based only on their corresponding redness values, reddish pixels outside the eye could be desaturated. Also, a patch of eye pixels devoid of color can look even more unnatural than a patch of slightly reddish pixels, especially if the picture has a high mean luminance with a small dynamic range of color values. Finally, a visible hard boundary between corrected and uncorrected pixels can have a displeasing look.

After detecting the eye,we isolate the eyes from rest of the image then the isolated eye candidate image is found for red pixels using the following rule for detection of red pixels :

$$X = R + G + B$$

$$R > 50 \quad \& \quad R/X > .40 \; \& \quad G/X < .41 \quad \& \quad B/X < .46$$

Pixels satisfying this condition are termed as red pixels. With very minimal scope of error the red pixels detected denote the redness in the eye.

Taking this color the red eye defect can be easily corrected by replacing the red pixels with an average value of Blue and Green plane intensity values. As a result a perfectly corrected eye is achieved.

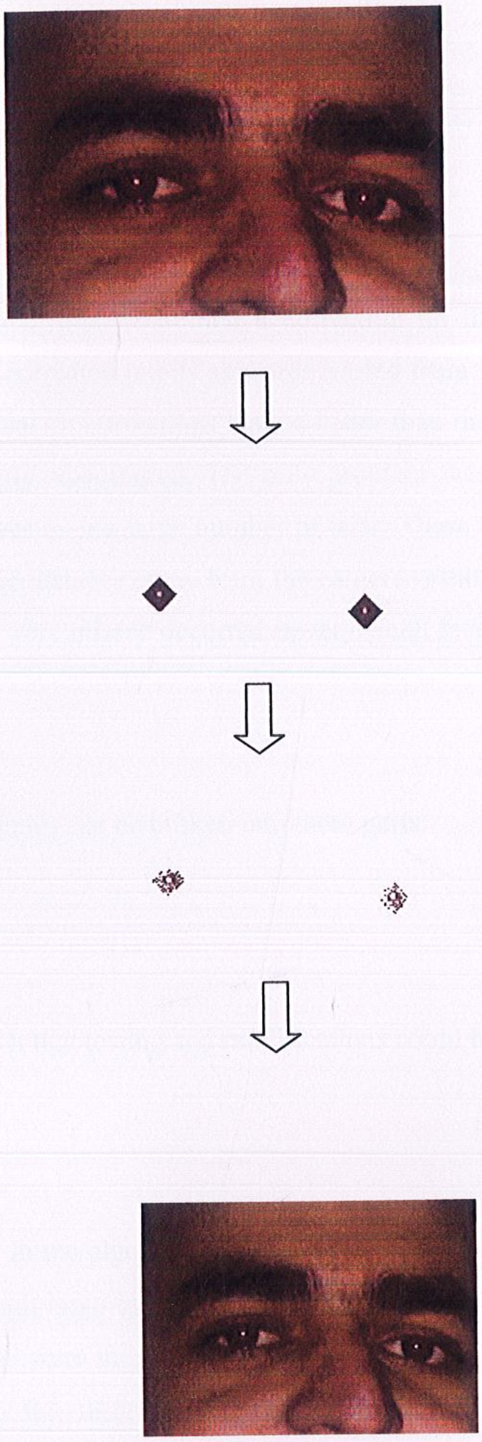


Figure 14: The Whole Process of Red eye correction

Chapter 6 EVALUATION

6.1 Results

The Red-Eye successfully identified, and performed a correction on the majority of images supplied. The degree to how well corrected pupils appeared varied from picture to picture. The general consensus, however, was that all corrections looked better than the initial red colour. No false detects were found or wrongly re-coloured. The only problem was that several red-eyes were misclassified as being non-eyes over a large number of tests. These eyes were either in the background of the picture or a large distance away from the camera resulting in a very low level of detail. Most of the artifacts that were missed occurred in very small faces, which were often in the background.

6.2 Design

The development of the entire program can be broken into three parts:

1. The automation
2. The detection.
3. The correction.

The design of the software was such that testing and modifications could be easily implemented.

6.3 Difficulties

Although there is very little code, in the algorithm for the colour correction, the problem was to do it in a realistic manner. Although large eyes were relatively easy to correct, smaller eyes i.e. those in the background of a photo were more difficult to deal with. The problems ranged from there being very little detail, to the fact that the red colouration of the pupil sometimes completely washed out the colour of the iris.

Obtaining good quality pictures containing red-eye was also a problem. Though the principle behind why red eye occurs was known, it did not help when it came to taking photos where the red was purposely induced and conversely where it was not wanted.

6.4 Successes

The procedure has a high success rate. A test was conducted on several individuals based on corrected images. Though it was possible to spot the images that were corrected when attention was specifically drawn to the parts that had been corrected. These tell-tails were in some cases a visible but faded pink rim around the pupil that was corrected or a much washed out residue of the pink colour visible as specks in the corrected pupil region. All corrected images were considered more favorable than the original images.

6.5 Future Work

There are several aspects of the code that could be modified. The statistic description provided for the eye is general and aimed primarily at eyes that detail is clearly visible in. A second model should be designed to target eyes that are relatively small in the image and as such would contain less detail.

An alternative to the second model could be the inclusion of a manual selection tool. This would be ideal for identifying red-eye in very small eyes that are a faded red in colour. This could be an alternative to the automatic procedure.

The present model works only on images having a single face. A second model should be designed which would also work for images having multiple faces.

A means to change the sensitivity could also be included. The change in sensitivity would simply be a lowering of the probability that would allow candidate regions to be considered eye regions.

Presently the red-eye system operates on a single face image at any one time. The provision of a batch facility would allow the user to select several pictures at one time and the software would then operate on the images in turn. This would safe the user the time spent waiting to use the software on each picture at a time.

More tests need to be conducted on the algorithms and indeed some tweaks. Following on from this, a series of fully working algorithms could have the potential to be incorporated into the firmware of some digital camera models.

Appendix A PSEUDO CODE

The platform used for coding is Matlab 2006Ra.

The whole project is divided in seven modules:

facedetect.m, dilation.m, dil2.m, facebox.m , crop40.m ,rgbsep.m, red.m

1) <u>Facedetect.m</u>

- Input the image.
- The image is converted into HSV and YCrCb format.
- The entire image is scanned and each pixel is examined under these constraints:

$0.836G - 14 < B < 0.836G + 44$

$79G - 67 < B < 0.78G + 42$

Image in RGB model

$19 < H < 240$

For image in HSV model

$140 < Cr < 165$

$140 < Cb < 195$

Image in YCrCb model

- Pixels satisfying these constraints are termed as skin pixels.
- These pixels are separated out from rest of the image for further operations.
- Store the image as skin.bmp.

2) <u>Dilation.m</u>

- Retrieve the image skin.bmp.
- The image is converted into binary as processing is faster and less complicated in binary images.

- The image is dilated with a circular disk of radius 5 pixels and then eroded with a circular disk of radius of 3 pixels.
- The binary image is stored as dilation.bmp.

3) **Dil2.m**

- Retrieve the image dilation.bmp.
- The image is scanned by a 5X5 window.
- If the number of white pixels lying in this window is more than a threshold value of 15, the whole area over which the window is currently placed is marked as white pixels else as black pixels.
- The image is stored as dil2.bmp.

4) **Facebox.m**

- The image dil2.bmp and the original image are read.
- The whole image (dil2.bmp) is scanned column wise and the number of white pixels in every column is stored in a flag value.
- A graph is plotted for the number of white pixels to column number.
- The two columns between which the maximum white pixels lie are marked as the starting (c1) and ending (c2) of the face, column wise.
- Now the image dil2.bmp is scanned row wise and the number of white pixels in each row between c2 and c1 are stored in a flag value.
- A graph is plotted for the number of white pixels to row number.
- The two rows between which the maximum white pixels lie are marked as starting (r1) and ending (r2) of face, row wise.
- A box is drawn around the face based on the c1, c2, r1 &r2 values in the original image.

- The face area is cropped and stored in cropfc_p.bmp.

5) **Crop40.m**

- The image cropfc_p.bmp is read.
- As our next task is to locate the eyes the image is cropped by 10% from top and 30% from bottom to reduce false eye detection.
- The cropped image is stored as crop_p.bmp.

6) **Rgbsep.m**

- The image crop_p.bmp is read.
- The image is separated into R , G, B planes.
- Each plane is scanned and the intensity values of each pixel are noted.
- Pixel value in each plane above a threshold value of 210 is marked.
- Some tests denoted in 4.5.1 are applied to check for the eye candidates.
- The common pixels in all three planes above the threshold denote the red eye region in the original image.
- After detecting the red eyes the eye region is isolated from the rest of the image for correction.
- The image is stored as redfl.bmp.

7) **Red.m**

- The image redfl.bmp is scanned.
- Pixels satisfying the condition in 5.1 are termed as red pixels.
- These pixels denote the redness of the eyes.

- Now the intensity value of the red pixels in all three (R G B) planes are modified with the average intensity of Blue and Green plane at that pixel concerned.
- This way the red eye effect is corrected automatically.

Bibliography

D. Sidibe, P. M. (n.d.). A simple and efficient eye detection method in color.

Ilia V. Safonov, S. R. (n.d.). Automatic red eye detection.

M. H. Yang et al IEEE trans. on PAMI. (n.d.). , "Detecting Faces in Images : A Survey.

Paul Cuff, D. o. (n.d.). Automatic Identification of Red-Eye in Photos.

Peng Wang, M. B. (n.d.). Automatic Eye Detection and Its Validation.

Qiong Wang, J. Y. (n.d.). Eye Detection in Facial Images with.

Sanjay Kr. Singh1 1Department of Computer Science and Engineering, D. S. (n.d.). A Robust Skin Color Based Face Detection Algorithm.

Woods, G. &. *Digital Image Processing* .