The undersigned certify that we have read, and recommended to the Institute of Engineering for acceptance, a project report entitled "Face Replacement System" submitted by Mubin Shrestha(38416), Rajan Prasad Upadhyay(38426), Ram Sharan Shrestha(38427), Robik Singh Shrestha(38430) in partial fulfilment of the requirements for the Bachelor's degree in Computer Engineering.

____

Supervisor, Sanjeeb Prasad Panday
Phd, Lecturer
Department Of Electronics And Computer Engineering, Pulchowk Campus

____

Internal Examiner, Subarna Shakya
Assistant Dean
Phd, Reader
Department Of Electronics And Computer Engineering, Pulchowk Campus

____

External Examiner, Saroj Shakya
Associate Professor
Department Of Graduate Studies, NCIT

____

Coordinator, Jitendra K. Manandhar
Deputy Head
Department Of Electronics And Computer Engineering, Pulchowk Campus

DATE OF APPROVAL: 12^{th} October 2012

COPYRIGHT

Acknowledgement

We offer our gratitude to our project supervisor, Dr. Sanjeeb Prasad Panday for his continuous support and guidance throughout the period of our project.

We want to gratefully acknowledge Mr. Chandra Man Shrestha from Verisk Information Technologies for guiding us throughout the internship period.

We would like to thank Mr. Avophagusa for helping us enhance our GUI.

We would like to acknowledge all the faculty members of Pulchowk Campus and family members.

Finally, we would also like to thank our friends and all others who have helped us with determination.

Abstract

Face replacement system plays a major role in the entertainment industries. However, most of the systems used in the industries are very expensive and require specialized skills. This project presents a semi-automatic "Face Replacement System" which can replace faces in still images. The system consists of two major parts. The first part is concerned with the extraction of face and hair regions. The second part is concerned with the face replacement. The face is extracted by using different algorithms such as skin color thresholds, snake algorithm and edge detection. The hair region is extracted by using seed region growing algorithm. The source face is warped and interpolated, and then, the face is pasted at a suitable coordinate. The source face is processed to make it color-consistent with the target image. Then, the boundary of the replaced face is blended to achieve realistic result. This face replacement system can be used to replace faces if they are in front view. When the faces have similar pose and the images have less noise, the results seem more realistic.

Keywords: Image processing, Face replacement system, Histogram matching, Snake algorithm, Skin region detection

Contents

List of Figures

List of Abbreviations

2D	Two Dimensional
3D	Three Dimensional
Cb	Chrominance(blue difference)
Cr	Chrominance(red difference)
GUI	Graphical User Interface
ROI	Region Of Interest
SRG	Seed Region Growing
Y	Luminance

1. Introduction

1. 1. Background

The ability to replace the face of a person in a photograph with that of another person has huge implications in the entertainment and special effects industries. For example, such a system could be used to replace the face of a stunt double with the real actor. Therefore, there has been a lot of research for the development of a system which can automatically replace human faces in photographs and videos. Some of the systems require Three-dimensional (3D) models of the faces, while others just require Two-dimensional (2D) photographs. Most of these systems also require uniform lighting conditions.

1. 2. Description

Face Replacement System is a semi-automatic software system which can be used to replace face of one person (target face) with that of another person (source face) in a photograph.

The software system requires two images: the source image and the target image. The target image contains the face which is to be replaced by the face present in the source image. These images are loaded via a user interface. It also requires the coordinates of the facial feature points on the source and target faces. The user should provide the coordinates of the facial feature points via the user interface.

The system extracts the face region in the image. Various algorithms have been implemented for face region extraction. One of them is: using threshold values of skin color. Another algorithm is: snake algorithm. After extracting the face region in the image, "shrinking and growing" algorithm is applied to remove non-face regions and the holes are filled. Then, the source face is warped according to the coordinates of the feature points of the target face. This ensures that the size and pose of the source face are similar to those of the target face. The color values for the pixels of the warped face are determined by applying interpolation.

Also, the system applies histogram matching (histogram specification) to the face to be replaced so that the source face becomes color-consistent with the target face. In the next step, the system detects the best possible coordinate to replace the face at. For this, a "matching degree function" is used to evaluate various candidate coordinates and then the candidate with the highest score is chosen. Then, the face is replaced. In the final step, the boundary of the source face is blended with the target image so that the final result is realistic and seamless. The process flow diagram is shown in Figure 2.1.

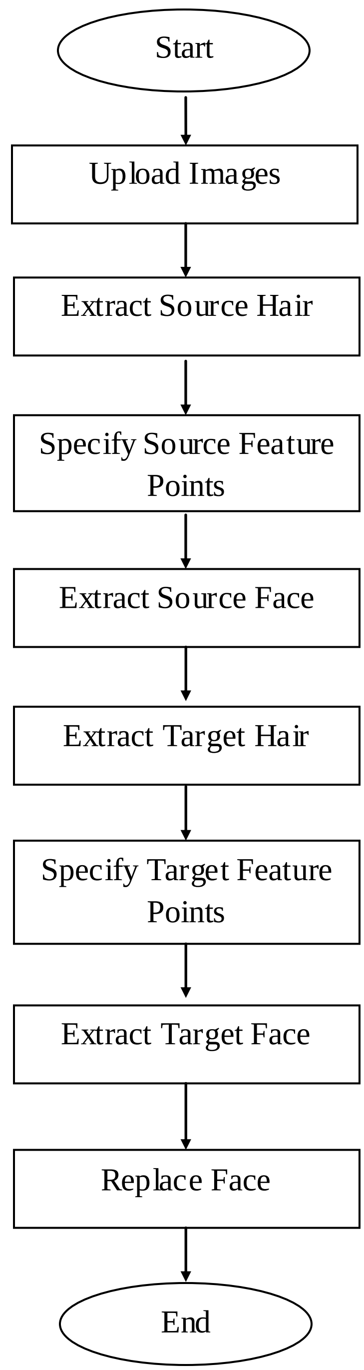


Figure 2.1 Process Flow Diagram of Face Replacement System

1. 3. Scope of Work

1. 3. 1. Photo Montage

Photomontage is the process (and result) of making a composite photograph by cutting and joining a number of other photographs. An example of photo montage would be to compose a photograph by replacing sad face with smiling face. This system can be used to perform that replacement. Thus, it can be used for photo montage.

1. 3. 2. Film Industry

Face replacement system can be used to replace the face of a stuntman with the face of the actual actor in a movie. This can be achieved by replacing the faces in each frame of the video. So, this system can be used in the film industry.

1. 3. 3. Entertainment and Fun

The users are expected to have a lot of fun playing around with this system. They can have a great time, replacing faces of politicians, friends, movie stars and sports stars with their own faces. They can produce funny images and share them with their friends. So, the system definitely has an entertainment value.

1. 4. Problem Statement

Face Replacement System aims to serve as a semi-automatic tool to the users so that they can easily replace face of one person with another person in still images.

Currently, there are various image manipulation software programs which can be used by the users to replace faces in photographs. However, the users need to possess specialized skills to achieve realistic results. That is, most of the systems provide various kinds of image manipulation tools for the users to work with. The users need to understand how to use these tools properly. However, there are many users who do not know how to use those tools properly.

The system provides simple image manipulation tools and uses various automated image processing algorithms in order to replace the faces in still images.

1. 5. Objectives

The objectives of the project are:

a. To implement skin color thresholding, snake algorithm and edge detection for face extraction
b. To implement warping, region growing, shifting, color consistency and blending for face replacement
c. To develop a face replacement application based on above algorithms

1. 6. Organization of the report

The first chapter is the introduction and deals with the background, description, and objectives of the face replacement system. The second chapter is the literature review which introduces related previous works, their findings and results. The third and most important chapter is the methodology which discusses implementation of different algorithms usedin the system. The fourth chapterhighlights the results and discussions. The results produced by each methodology are described and discussed in this chapter. Finally, the report concludes with the conclusions, limitations and future enhancements. A list of references and appendices are provided at the end.

2. Literature Review

Face replacement technology has a great use in the entertainment industries and special effects industries. Replacing face of one person with that of another person in video sequences has huge implications.

There are two main steps for a system to be able to replace the face of one person with that of another person.

a. Facial feature extraction
b. Face pose estimation

There are two general approaches to extract the face region. The first is the region based segmentation and second is the edge based segmentation. The region based method uses similarity condition where if a connected sub-image has similar characters as the known sample skin region, the sub-image is classified as face region. On the other hand, the edge based method is based on dissimilarity condition. If a boundary exists between two regions then one region which is more likely to be a skin region (according to geometrical, connectivity and other criteria) is classified as skin.

The most general method to detect the facial features is to detect the face region by using the characteristics of skin color. Once the facial region is detected, the information about statistical as well as geometric relation is used to determine the facial features. The most common pre-processing method is to detect skin region by a skin tone model(1). There are many color models used for human skin color(2)(3).

Neural network based approaches require large number of face and non-face training examples(4)(5). Face pose estimation can play important role to improve the accuracy of the face replacement. For our purpose we narrowed our domain to normal front face only, but there are methods where exact face pose can be estimated with various 2D as well as 3D face models (6).

Genetic algorithm based face boundary extraction needs large number of generations to get accurate result. The fitness function is used to evaluate the chromosomes. The function is based on the length of the longest edge. The fitness function takes time because the number of chromosomes and the number of generations are large.

Adaptive active contour model (Snake algorithm) requires much less time than Genetic algorithm. Snake algorithm is an iterative process. The time for detecting the face boundary directly depends upon the number of iterations. Snake algorithm uses an energy minimizing function. The curve is formed such that minimum energy is bounded by the deformable curve. In the classical active contours, the snakes follow the contours only and hence they do not give good results. The snake model used here is based on a geometrical approach that circumvents these neighboring vertices to move too far away from each other. The key issue for the successful application of this new snake model is the tuning of the parameters controlling its rigidity. In particular, we observed that it is often beneficial to let the elasticity constraints vary with time. However, the optimal temporal profile cannot be specified beforehand. We thus use an adaptive scheme that automatically adjusts the rigidity of the snake during its evolution towards an image contour.(7)

After the skin region is segmented, one of the necessary steps is chin curve estimation where five feature points are used to detect the curve to separate the face region from the neck region (8). The feature points are actually used for two major tasks, chin curve estimation and warping. Region Growing can be used as a tool to estimate both, skin region as well as the hair region but we have used it only for hair region extraction. After the face region is estimated, the target face is replaced with the source face.

The first step for replacement is warping, where shifting, scaling and translation of the source face region is done. This transformation is based on the spatial transformation coefficients obtained from corresponding feature points of source and target faces. This step is necessary because often the aspect ratio, length and width of source and target faces may be different and direct replacement of source face over target face is not possible or does not produce realistic result. Warping changes the pose and size of the source face such that it fits exactly over the target face.

The second step is to make the skin tone of source face consistent with the skin tone of the target face. This step is not so much relevant or necessary if both the source and target have similar skin color (black or white) with similar skin tone. However, it is a necessary step for faces having different skin colors. For example, replacing a dark target face with a fair source face may produce unrealistic result. This is because the result might have dark neck region and fair face region. It is necessary to either change the source skin color to that of the target or the target skin color to that of the source to produce realistic result.

The best position of replacement is estimated next. That position ensures that the region of overlap between source and target face is the largest. The target face region and the warped source region are correlated by sliding the warped source skin region over the target image. The point where maximum correlation occurs is taken as the replacement point.

The edge of the segmented region is blended in the fourth step which increases the smoothness of the edges(8). The purpose of blending is to reduce the sharp transition effect between target skin region and source face region in the final image. In spite of color adjustment, the source skin and the target skin are unlikely to be same. So when the source face is drawn over the target face, undesirable artifacts (edges) are produced. In blending, the boundary between the target face and the source face is interpolated in terms of intensity so that an intensity gradient is developed in the final image. This helps the final result to look more realistic.

3. **Methodology**

3. 1. Face Region Extraction

Face region extraction is the process of finding out the face region in an image. Prior to this phase, the user is expected to have provided the necessary information which includes the coordinates of the facial feature points and a rectangle around the face region. The process of face region extraction consists of two phases:

1. Preliminary Stage of Face Region Extraction

2. Application of Curve Bounds

3. Hair and Background Region Removal

3. 1. 1. Preliminary Stage of Face Region Extraction

The preliminary stage of face region extraction is the first step in the process of extraction. There are various alternative methods applied during the preliminary stage of face region. The algorithms which are used in this project are:

1. Skin Color Detection

2. Snake Algorithm

3. Edge Detection

The following are the descriptions of the algorithms enlisted above.

3. 1. 1. 1. Skin Color Detection

Face region can be extracted by using skin color detection. First of all, the image is segmented by skin color thresholds. Then, shrinking and growing algorithm is applied to remove noise.

3. 1. 1. 1. 1. Skin Color Threshold

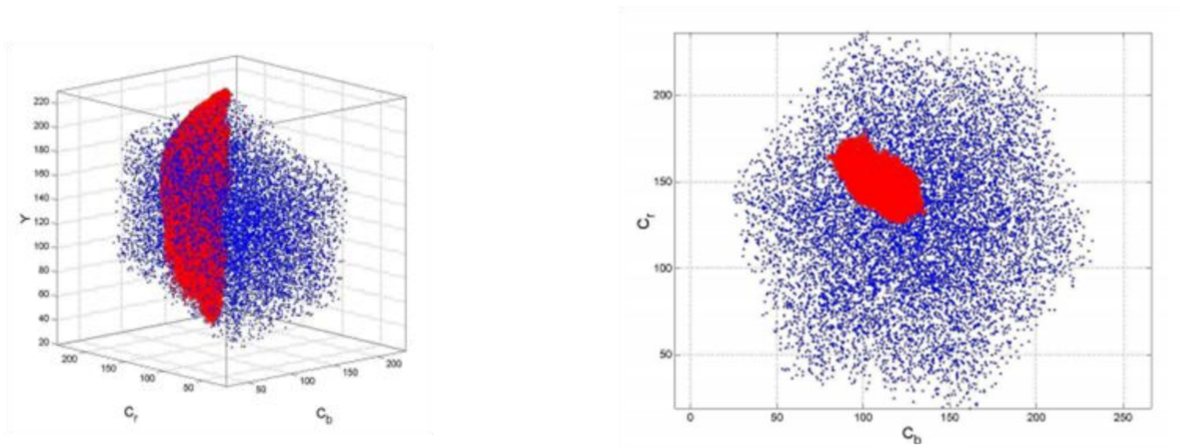
There are many methods to approximate skin color for extraction of face region. We have used the YCbCr color model for our purpose. One of the biggest advantages of using this color model is that it is perceptually uniform. The YCbCr color space and skin tone model are shown in Figure 3.1.

The YCbCr color model separates chrominance from luminance. There is a certain range of chrominance and certain range of luminance for which a color is acceptable as a skin color. The thresholds for Y is (50, 240), Cb is (-50, -5) and Cr is (0, 50). The formulae for converting RGB image to YCbCr are mentioned in Equations (3.1)-(3.3).

$$Y=0.299 \quad * R + 0.587 \quad * G + 0.114 \quad * B \tag{3.1}$$

$$Cb=-0.16874 * R - 0.33126 * G + 0.50000 *B \tag{3.2}$$

$$Cr=0.50000 * R - 0.41869 * G- 0.08131 * B \tag{3.3}$$



(a) (b)

Figure 3.1The YCbCr color space (blue dots represent the reproducible color on a monitor) and the skin tone models (red dots represent the skin color samples) (a)YCbCr color space (b) 2D project on CbCr

A binary matrix is constructed with each element having a value of either 1 (skin pixel) or 0 (non-skin pixel). Thus the pixels in the extracted face region are given by the elements of that binary matrix having a value of 1.

The region thus extracted, is not yet perfect because of the presence of various non-skin regions inside the face region such as the eye brow, mustache, and other things. The second step is to remove the unnecessary noise which might have been introduced with erosion and dilation. The third step is to extract the face region boundary by scan-line method where each row of the matrix is scanned and the first and last skin pixels are detected. Then the areas between the first and last skin points are filled to create a continuous region. The same process is applied vertically and horizontally and a continuous region is prepared. The results are shown in Figure 3.4.

Algorithm

function*SKIN_COLOR_DETECTOR(image)* ***returns*** *a matrix*

> ***input:*** *Initial Image*
>
> ***output:*** *matrix with elements having value 1 for skin color, and 0 for non-skin color*
>
> *M ← matrix of same size as that of image*
>
> ***for*** *each p as pixel in image of coordinate x,y*
>
> > *Check Threshold for Y,Cb, Cr*
> >
> > ***if*** *Threshold (P) is in range*
> >
> > > *M(x,y)=1*
> >
> > ***else***
> >
> > > *M(x,y)=0*

return *M*

3. 1. 1. 1. 2. Shrinking and Growing (Erosion and Dilation)

Shrinking and Growing Algorithm is also known as the Erosion and Dilation. The output of YCbCr Color Space Model contains some noise as shown in Figure. So, such noise should be removed for further processing. And from our experiments we found out that there's a huge probability that such noise can be significantly reduced with the help of shrinking and growing algorithm.

Shrinking process is achieved by peeling off a layer of a certain thickness around the boundaries. In our process, we have assumed the threshold value as 1 which denotes the number of layers to be shrunk. Shrinking removes the smaller structures step by step, and finally only larger structures remain. The final output is shown inFigure 3.3 (f).

After shrinking the image, it is grown. The process of growing back the shrunkimage is known as Growing. Eventually the larger regions should return to their original shapes.On the other hand, the smaller regions should disappear from the image. The final output of growing is shown in Figure 3.3 (g).

So all we need for this process are two types of operations.Figure 3.3(c)–(d)show the results of shrinking. Figure 3.3 (e)—(f) show the results of growing.

The snapshots of the results of shrinking and growing algorithm are shown in Figure 3.2.

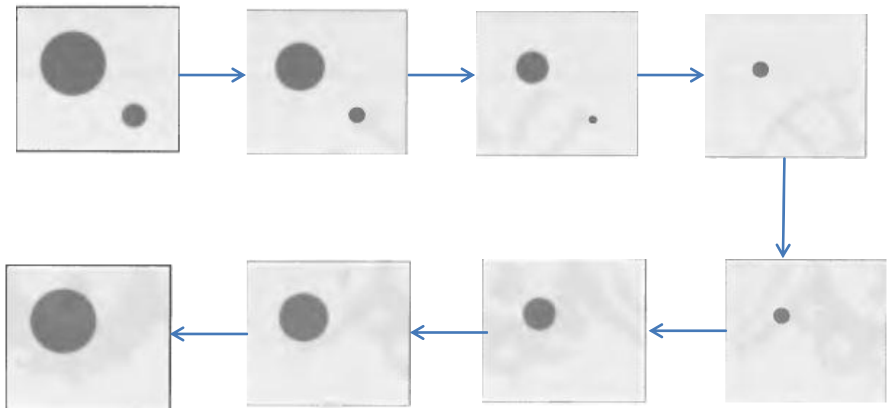


Figure 3.2Removing small image structures by stepwise shrinking and subsequent growing

3. 1. 1. 2. Adaptive Active Contour Model (Snake Algorithm)

Extracting the region of interest bounded by definite contour has always been a challenging area in image analysis. And in our case it is the face region. Extracting the face region had really been a challenging task in our project. Since our region of interest was the area bounded by the face contour, there is a high probability that we can extract the face region with the help of snake algorithm. The fully automated contour extraction models are not fully accurate and there are always some noises embedded with the output. Also, they introduce high deviation from our expected output. Since we are enabling user interactions, we are using the semi-automated method. Among many of the contour following algorithm we have chosen this snake algorithm because they refine the user provided region of interest. The final output obtained after the snake algorithm is optimal in the sense that it has the minimum energy function defined for the candidate contours.

Our adaptive active contour model includes two classical approaches of deformable models:

1. Deformable curves
2. Classical active contour

3. 1. 1. 2. 1. Deformable Curve

In the deformable curves formulation, if a curve is submitted to a deformation vector field, only the normal component of the deformation vector acts on the shape of the curve. The tangential component only generates a re-parameterization of the curve.

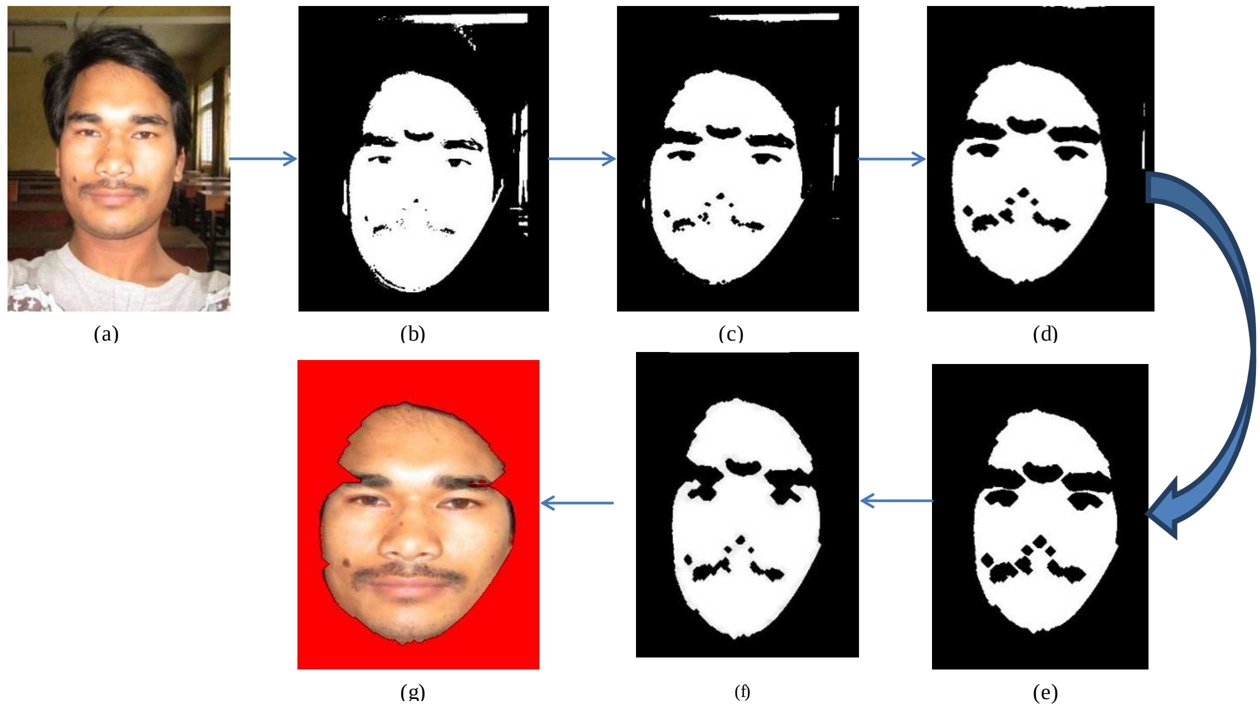


Figure 3.3 (a) Original image (b) Skin region (c)-(d) Shrinking (e)-(f) Growing (g) Output image

Figure 3.4 Face region extraction (a)Face region extracted using threshold values of YCbCr (b) Face region extracted after filling holes

3. 1. 1. 2. 2. Classical active contour

Classical active contour models, based on a gradient vector fields, tend to move the snake towards the nearest edges. In the case of infoldings, these edges are at the entrance of the invigilated structure. Hence, the model is stuck at the entrance, while it would be interesting to look at the bottom of cavity.

Our approach is actually based on the deformation along normal. It makes the model attracted not by the truly closest edges but by the nearest edges found perpendicularly to the model, which in the case of infoldings are located as internal energy. The final displacement thus result from a compromise between the complete displacement towards the nearest contour along the normal and the satisfaction of the geometrical constraints incorporated into the internal energy.

3. 1. 1. 2. 3. Finding the nearest edges

At any time step, the snake is a list of p points N_1, N_2, N_3, ……………,N_P. Eachpoint N_i is attracted towards the nearest edge along the normal of that point. Edges are detected using Canny-Deirche operator. Starting from the current position of point N_i, the search is conducted along the normal for the nearest pixel having a gradient magnitude above a specified threshold, within a specified distance range in both directions. The difference between this pixel and point N_i is a candidate displacement, $\overrightarrow{D_i}$. In the absence of regularization, the new position M_i of point N_i should verify

$$M_i = N_i + \overrightarrow{D_i} \tag{3.4}$$

3. 1. 1. 2. 4. Regularization

The snake's model try to segment the image based on the following energy:

$$E_{total} = E_{int} + E_{ext} \tag{3.5}$$

Where,

$$E_{int} = \int (\alpha |c'|^2 + \beta |c''|^2) ds \tag{3.6}$$

Wherein $\alpha \& \beta$ are the regularization parameter that vary along the curve. |c'| is the curve. We have set $\beta = 0$ to increase computation speed. Our experience is that this does not dramatically affect the results. So, Equation(3.6) now becomes:

$$E_{int} = \int \alpha |c'|^2 ds \tag{3.7}$$

which can be further denoted as:

$$E_{int} = \sum_i \alpha_i |M_i - M_{i-1}|^2 \tag{3.8}$$

and,

$$E_{ext} = -|\nabla l|^2 \tag{3.9}$$

Finally,

$$\alpha = \frac{\gamma}{1 + \mu \overrightarrow{D_i}} \tag{3.10}$$

where,

γandμ are coefficients for controlling the range of α values and $\overrightarrow{D_i}$ normalized between 0 and 1.

3. 1. 1. 2. 5. Adaptive Procedure

Our implemented procedure automatically adjusts the snake parameters during its evolution towards the final position. Initially, regularization parameters are given high values to guarantee model smoothness and to capture the overall shape of the object. Each time step, the average candidate displacement $\vec{D}$ is computed. A decreasing of this quality from one step to the next indicates that the snake is progressing towards edges. In that case, regularization parameters are left unchanged.

Algorithm

function*Snake(ROI, I, T)* ***returns*** *Curve*

inputs: *ROI is the Region Of Interest*
I: Number of iterations
T: Threshold (input parameter for Canny Edge Detector)

output: *The output is the face boundary (Curve)*

for each *iteration*

Calculate E_{ext} $(\bar{v}_i)$ *and the derivative of this with respect to x and y separately.*

At the start of the iteration, calculate $\frac{\partial^2 x}{\partial s^2}(\bar{v}_i)$ *and* $\frac{\partial^2 y}{\partial s^2}(\bar{v}_i)$ *using the three adjacent points and* $\frac{\partial^4 y}{\partial s^4}(\bar{v}_i)$, $\frac{\partial^4 x}{\partial s^4}(\bar{v}_i)$*using five adjacent points.*

Then, calculate change in x and y for each point in $\bar{v}_i$

Adapt the Curve

return *Curve*

3. 1. 1. 2. 6. Output

For the source image of which the face region is to be detected, the user first makes the selection of region of interest as shown in Figure3.5(a). The intermediate result of our snake algorithm is shown in Figure3.5(b). The final output of snake algorithm is shown in Figure3.5(c).

(a) (b) (c)

Figure3.5(a) Selection of region of interest (b)Movement of curve points perpendicularly towards edges. (c) The final output

3. 1. 1. 3. Edge Detection

Edge Detection is another method of face region extraction. The basic idea is to detect edges around the face so that the boundary around the face can be extracted. The edge detection process consists of two phases:

i. Canny EdgeDetection
ii. Longest Edge Detection (along with Genetic Algorithm)

3. 1. 1. 3. 1. Canny Edge Detection

Canny Edge Detection is one of the most popular edge detection techniques. Detecting edges in an image filters out useless information.

The Canny Edge Algorithm used in this system uses two thresholds, high and low thresholds. The value of low threshold is (0.4 x (high threshold)). The two values of thresholds are used to distinguish between strong and weak edges by the values of strength of the edges. To find the strength of edges, longest edge detection algorithm is used. Also the weak edge length is considered if and only if they are connected with the long edges. The output of Canny Edge Detection is a binary image. This binary image is regarded as the edge map for our face contour extraction algorithm.

Canny Edge Detection is used to detect the edges. For the detection of edges of face region, a probable rectangle region including face is selected with the help of the separation of the eyes. If the separation between two eyes is ‘2D’ then the width of the probable rectangular region is defined to be ‘5D’ and the height is defined to be ‘8D’ as shown in Figure 3.6.

After the rectangular region is extracted, the rectangle is divided into four parts: forehead of dimension (5D x D), left face of dimension (D x 7D),right face of dimension (D x 7D), and bottom face of dimension (5D x D) as shown in Figure 3.7.

(a) (b) (c)

Figure 3.6 (a) Original Image (b) Separation of eyes (c) Probable face region

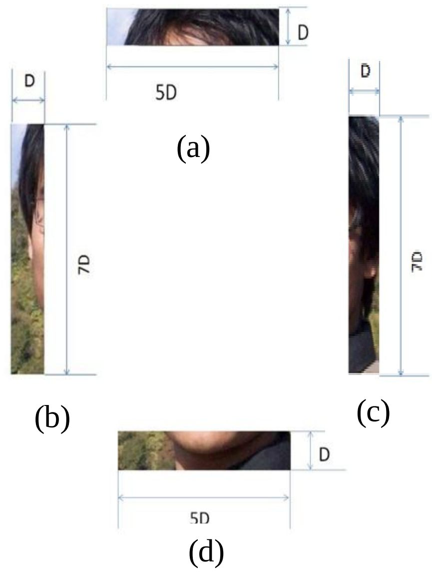


Figure 3.7 The original face is divided into four parts as shown above. (a) Forehead image (b) Left face (c) Right face (d) Bottom Face

Then, in each of these parts, Canny Edge Detection is applied. The final result is shown in Figure 3.8.

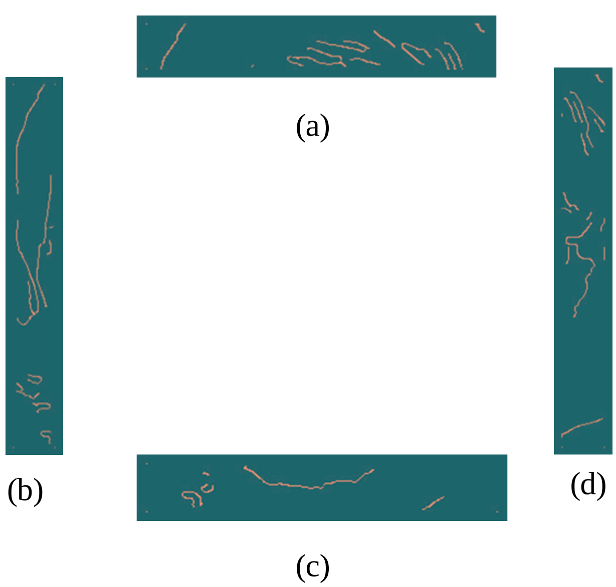


Figure 3.8 : The edges of four face region. (a) Forehead image (b) Left face image (c) Right face image (d) Bottom face image

3. 1. 1. 3. 2. Longest Edge Detection

The output of Canny Edge Detection of four parts of face image contains many edges including weak and strong edges. We are only interested with the long and strong edges. For the detection of such long edges we have performed following steps:

Step 1:

The two images i.e. forehead image and bottom face image are divided vertically into four parts. Each divided certainly cuts out edges. The divided region of bottom face is shown in Figure 3.9.

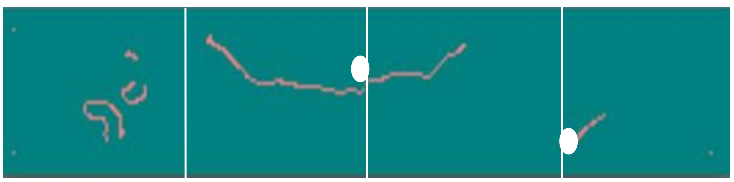


Figure 3.9 The bottom face divided into four parts.

Step 2:

Edge is grown from the point of intersection with the boundary. In this algorithm, the edge is followed from left to right. The edge is considered to be a strong edge if its length is greater than a threshold length, T.

Step 3:

Weak edges are removed. The image obtained after removal of weak edges is shown in Figure 3.10.

Figure 3.10 Longest edge from bottom face

Step 4:

The same process is applied to all four parts.The same process is applied to the left and right regions, but the regions are divided according to the height. The final probable edge image is shown inFigure 3.11.

Figure 3.11 The overall edge map of the face of longest and strong edges obtained after longest edge detection is applied.

Step 5:

After the probable edge maps are found out using canny edge detection, all the edges are linked together using either "Hough Transform" or "Active Contour Model".

3. 1. 1. 3. 2. 1. Genetic Algorithm

Genetic Algorithm can be used to find out the best value of thresholds for Canny Edge Detection. This searching algorithm evaluates candidate threshold values with a fitness function. The fitness function gives the length of the longest edge in the image.

The genetic algorithm is given below.

function*GENETIC_ALGORITHM(P, C)* ***returns****best chromosome*

  inputs: *P is the population size*

    C is the chromosome size

  Create the initial random population

  repeat

Calculate fitness value of each individual of the population

Sort the population according to their fitness value

Select the parents for crossover (Roulette Wheel Selection)

Crossover (Single point crossover)

Mutation (Bit inversion mutation)

Replace the parents with the new children using elitism(preserving some of the best individuals from the old population)

***until** the maximum iteration is finished or the condition is fulfilled*

Sort the result population according to their fitness value

***return** the best individual*

3. 1. 2. Application of Curve Bounds

The following curve bounds are used:

1. Forehead Curve Bound
2. Chin Curve Bound

3. 1. 2. 1. Forehead Curve Bound

Canny Edge Detection is always a good choice for the detection of edges. With the help of Canny Edge Detection and Longest Edge Detection technique we are estimating the forehead curve bound. The estimation of forehead curve bound includes the following steps:

STEP 1:

First with the help of feature points of left eye and right eye provided by the user,the forehead region is cropped out. For detection of forehead the distance between two eyes is calculated (2D). The center point from where the forehead region is cropped out is the point just in the middle of the two eyes. The measurements of the forehead are shown in Figure 3.12.

STEP 2:

Secondly the cropped forehead obtained from the first part is divided into three parts vertically in the ratio 1.5:2:1.5. The cut parts are shown in Figure 3.13along with their width and height.

STEP 3:

Thus obtained parts are individually treated with Canny Edge Detection.The output of Canny Edge Detection to the individual parts is shown in Figure 3.14.

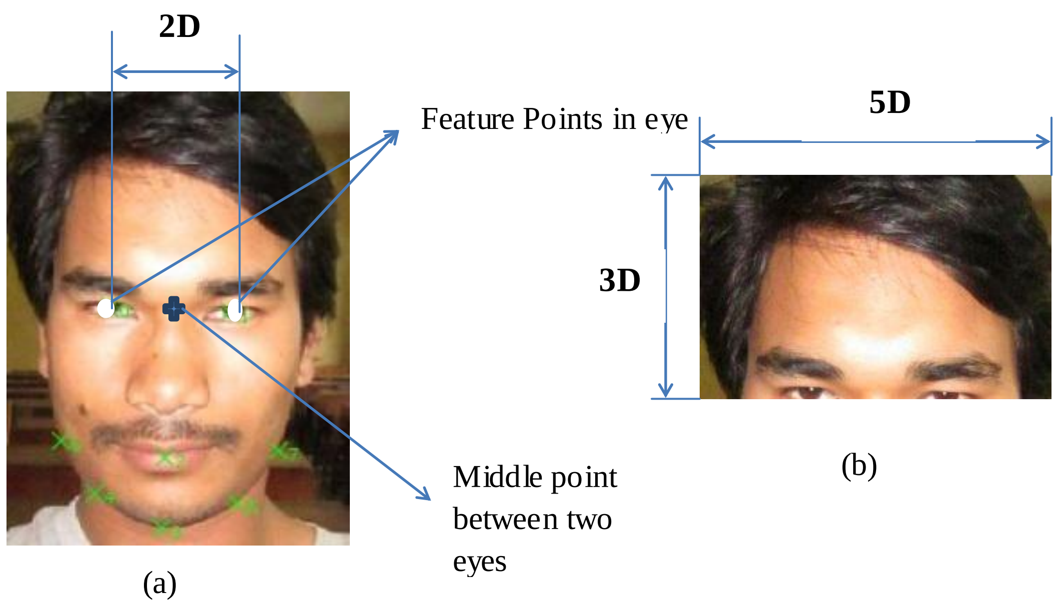


Figure 3.12 Cropping (a) Original image with feature points (b) Cropped forehead

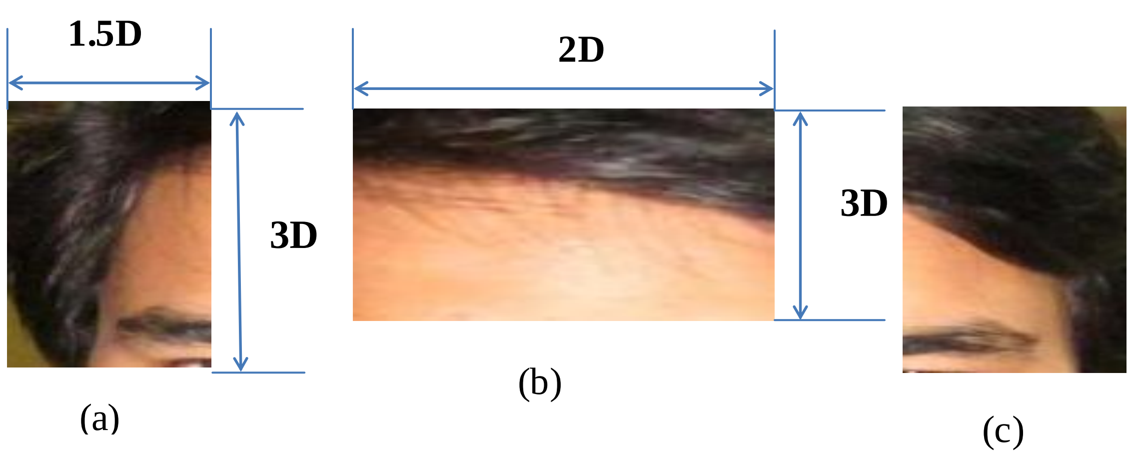


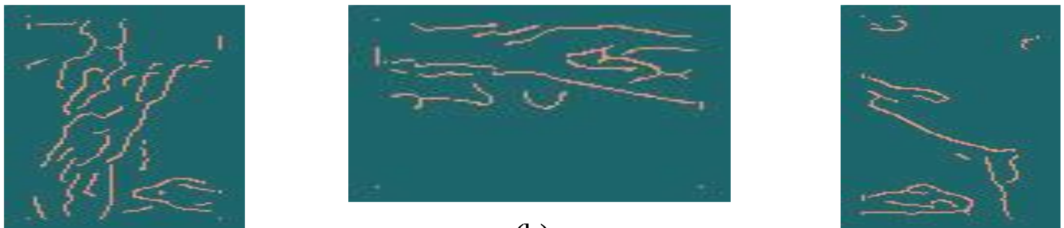
Figure 3.13 (a) Left (b) Middle (c) Right parts of forehead

STEP 4:

Longest edges from the edge map obtained from Canny Edge Detection technique is obtained by applying longest edge detection technique. The result of Longest Edge Detection is shown in Figure 3.15.

Step 5:

The longest edges from the three forehead parts are then connected to find the forehead curve bound. The connection is done by the Curve Fitting.



(a) (b) (c)

Figure 3.14 Output of Canny edge detection (a) Left part of forehead (b) Middle part of forehead (c) Right part of forehead

(a) (b) (c)

Figure 3.15Longest edges in forehead. (a) Left part of forehead (b) Middle part of forehead (c) Right part of forehead

The overall edge map for the forehead is shown in Figure 3.16.

Figure 3.16 Overall edge map of longest edges of forehead

3. 1. 2. 2. Chin Curve Detection

Curve estimation is one of the processes where we approximate curves based on five feature points. This is a simple process where with the help of mathematical tools, we approximate a curve of second degree to estimate the boundary of the face region. The feature points are supposed to be provided by the user. This step is necessary as it is used to differentiate the face region from the neck region, where both the regions are the skin regions.

$$\begin{bmatrix} n & \sum x_i & \sum x_i^2 \\ \sum x_i & \sum x_i^2 & \sum x_i^3 \\ \sum x_i^2 & \sum x_i^3 & \sum x_i^4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} y_i \\ x_i y_i \\ x_i^2 y_i \end{bmatrix} \tag{3.11}$$

The curve of second degree is given by Equations (3.12)-(3.14).

$$y = a_0 + a_1 x + a_2 x^2 \tag{3.12}$$

$$y = a_0 + a_1 x + a_2 x^2 \tag{3.13}$$

$$y = a_0 + a_1 x + a_2 x^2 \tag{3.14}$$

Algorithm

function*CURVE(points)* ***returns****coefficients*

input: *points=array of sample points (minimum 3 points)*

output: *List of points forming the curve*

for *each point in Points*

Calculate the root mean square error

Fit the curve with minimum error

return*coefficients*

3. 1. 3. Hair and Background Region Removal

The final step necessary for face region extraction is: hair and background region removal. Seed Region Growing algorithm is implemented for extraction of hair and background regions. The system offers the user with adjustable threshold values and eraser tool to select the hair and background regions properly. The regions thus obtained are removed from the face extracted using the algorithms mentioned prior to this section. The details of hair region extraction and background region extraction are provided in the next section.

(a) (b)

Figure 3.17 Curve estimation (a) Feature Points (b) Chin curve is shown in red color

The result provided by the face region extraction may contain hair pixels too. The reason is that the hair pixels might have a similar color to that of skin color. So, the hair region should be removed from the face region. This is the final stage of face region extraction. Figure 3.18(a) is the original image, (b) shows the result obtained after preliminary stage of face region extraction and (c) shows the result obtained after removing hair region.

(a) (b) (c)

Figure 3.18 (a) Original Image (b) Result obtained after preliminary stage of face region extraction (c) Result obtained after removing hair

3. 1. 3. 1. Hair Region Extraction

Hair region extraction is a necessary step in face replacement. If the user wants to replace the target hair with the source hair, then the hair region of the source image needs to be extracted and replaced.

To detect the hair region, an appropriate block above the face region is selected.The skin color region is neglected and the remaining pixels are used as seeds for Seed Region Growing (SRG) algorithm. This approach to segmentation examines neighboring pixels of initial “seed points” and determines whether the neighbors should be added to the region or not.

SRG requires some seeds to grow from. It checks the neighbors of the seed. If they satisfy the similarity criteria, then they are labeled (i.e. they are added to the region). This process is repeated for their neighboring pixels.

Algorithm

function *SRG(seeds, image)* ***returns*** *R*

> ***inputs:****seeds, the seed pixels from which a region is to be grown*
>
> > *image, the image from which a region is to be extracted*
>
> ***output:*** *The region obtained after growing the seeds is R.*
>
> *S ←seeds, R←seeds*
>
> ***repeat***
>
> > ***for each*** *p in S, do*

insert(p,R)

*neighbor← N8 (p) /*N8(p) refers to 8-adjacent neighbors of p*/*

repeat

  ***for each** q in neighbor*

    ***if**(contains(q,Q)) continue*

    ***if**(PREDICATE(q) == TRUE)*

      *insert(q,S) /*insert to q to S*/*

***until** S becomes empty*

***return** R*

In simple words, the hair region is detected by finding pixels similar to the seed pixels in the hair region. The original image and the result obtained after application of SRG are shown in Figure 3.19.

(a) (b

Figure 3.19 Seed Region Growing (a) The red rectangle contains the seeds (b) The red region is the result of seed region growing

3. 1. 3. 1. 1. Similarity Criteria:

A neighboring pixel is considered to be similar to the seed pixel if the intensity of that neighboring pixel lies within specified range of intensity values. The range of intensity values is given by:

$$[I_{min} - coeff * range,\ I_{max} + coeff * range] \tag{3.15}$$

where,

$$coeff = \text{Coefficient provided by the user } (0,1) \tag{3.16}$$

$$range = I_{max} - I_{min} \tag{3.17}$$

$$I = \text{Intensity values} \tag{3.18}$$

The color model chosen is YCbCr. So, a neighboring pixel is considered to satisfy the similarity criteria if each color component: Y, Cb and Cr lies in its range.

3. 1. 3. 1. 2. Removal of Holes:

The region obtained after applying SRG algorithm contains holes. These holes can be filled up by an algorithm. The algorithm searches for pairs of pixels separated by a certain distance such that both the pixels belong to the hair region. The separation can be along either horizontal direction or vertical direction. Whenever, such a pair is found, all the pixels lying between them (i.e. lying along either horizontal direction or vertical direction in between those two pixels) are considered to be part of the hair region. In this way, the holes can be filled up.

The algorithm is given below:

function *FILL_HOLES (matrix, gap_length)* ***returns*** *R*

> ***inputs:*** *matrix, the region which may contain holes. It contains the region grown using SRG.*
>
> *gap_length, the maximum diameter of a hole*
>
> ***output:*** *The region obtained after filling the holes is R.*
>
> *R ← matrix*
>
> ***for****(x = 0; x < matrix_width; x++)*
>
> ***for****(y = 0; y < h; y++)*

if(*matrix[x][y] == 1*)

for *(delX = gap_length; delX >= 2; delX--)*

if *(matrix[x + delX][y] == 1)*

for *(fillX = 1; fillX < delX; fillX++)*

R[x + fillX][y] ← 1

return *R*

Figure 3.20 shows the results of this algorithm. The image at the left hand side is the result obtained after applying SRG algorithm. It contains holes. After filling those holes, a better result is obtained.

(a) (b)

Figure 3.20 (a) Result of SRG without filling holes (b) Result of SRG after filling holes

3. 2. Image Warping

Image warping is a spatial transformation that includes shearing, scaling and rotating an image. An affine transformation matrix is used for image warping.

The coordinates of the source and target facial feature points are known. This information can be used to derive a spatial transformation matrix. This matrix can transform every pixel in the source image to another image. The application of such a matrix is known as warping. The result obtained after the application of warping is known as “warped image” and it is shown in Figure 3.21.

(a) (b) (c)

Figure 3.21Image warping (a) Source image (b) Target image (c) Warped image

The affine transformation matrix is defined by Equation (6).

$$\begin{bmatrix} X' \\ Y' \\ 1 \end{bmatrix} = \begin{bmatrix} m_1 & m_2 & m_3 \\ m_4 & m_5 & m_6 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \tag{3.19}$$

where,

(X', Y') is the feature point coordinate of target face

(X, Y) is the feature point coordinate of source face

m_i's are the parameters to be determined

i=1 to 6

The values of the parameters (m_i) are determined by using Least Square Method.

3. 2. 1. Least Square Method

The values of (m_i) fill up the affine transformation matrix. To determine those values, Least Square Method (LSM) is applied.

Let $(X'_{actual}, Y'_{actual})$ be the actual coordinate of a target feature point.
Let (X', Y') be the coordinate of the target feature point obtained from Equations (3.20) and (3.21).

$$\frac{d\sum\left(X' - X'_a\right)^2}{dm_i} = 0, \qquad i = 1, 2, 3 \tag{3.20}$$

$$\frac{d\sum\left(Y' - Y'_a\right)^2}{dm_i} = 0, \qquad i = 4, 5, 6 \tag{3.21}$$

This method ensures that the sum of the squares of the errors between the actual x-coordinate and derived x-coordinate is the least possible. The same applies for the y-coordinate.

3. 2. 2. Mapping

There are two ways to apply the image warping:

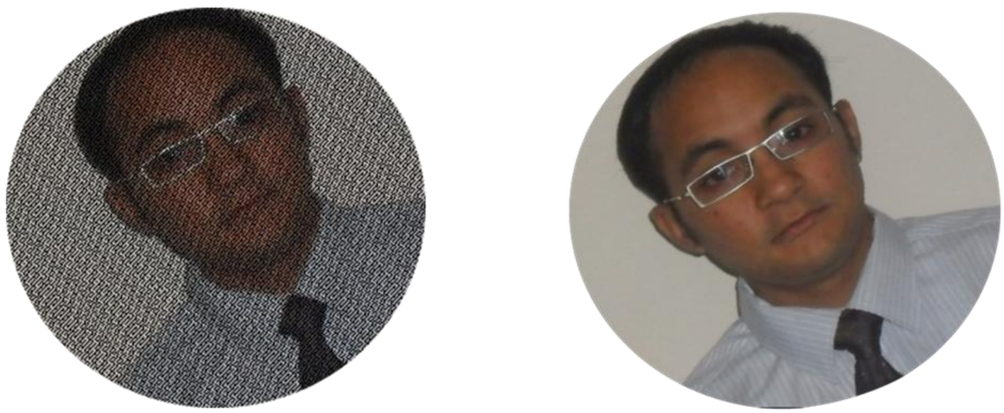
1. Forward Mapping
2. Reverse Mapping

3. 2. 2. 1. Forward Mapping

In forward mapping, the pixels of the source image are mapped to the pixels of the warped image. The main problem with the forward mapping is that not all the pixels in the warped image have corresponding pixels in the source image. This causes the warped image to contain "holes" as shown in Figure3.22.

3. 2. 2. 2. Reverse Mapping

In reverse mapping, the pixels of the warped image are mapped to the pixels of the source image. The warped image does not contain "holes" because all of its pixels are mapped to the source image. The result of reverse mapping is shown in Figure3.22.



(a) (b)

Figure3.22 Results of warping (a) Forward mapping (b) Reverse mapping

3. 3. Interpolation

When the pixels of the warped image are mapped to the source image, the mapped pixel positions are obtained as floating-point coordinates. However, the pixels in the source image have integer coordinates. So, the color values of the pixels in the warped image should be interpolated from the neighboring pixels of their mapped coordinates in the source image.

The interpolation is performed by providing weights to the neighbors of the mapped pixel coordinate. The weights decrease as the distance between the pixels increase.

Let us suppose:

(x, y)	=	Coordinate of the pixel in the warped image
$I_{warped}(x, y)$	=	Intensity of the pixel in the warped image
(x_{mapped}, y_{mapped})	=	Coordinate in the source image which corresponds to the coordinate (x, y) in the warped image
(x_{source}, y_{source})	=	The coordinate obtained after truncating the decimal parts of the components of (x_{mapped}, y_{mapped})
$I_{source}(x_{source}, y_{source})$	=	Intensity of the pixel in the source image

The decimal part of the coordinate is given by Equations (3.22) and (3.23).

$$x_t = x_{mapped} - x_{source} \tag{3.22}$$

$$y_t = y_{mapped} - y_{source} \tag{3.23}$$

Then, the values of the weights are given by Equations (3.24)- (3.27).

$$Coeff_{(0,0)} = (1 - x_t) * (1 - y_t) \tag{3.24}$$

$$Coeff_{(0,1)} = (x_t) * (1 - y_t) \tag{3.25}$$

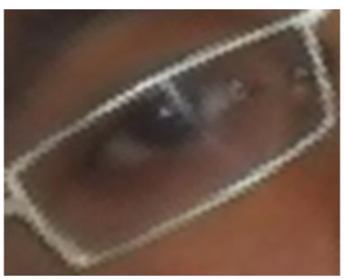
$$Coeff_{(1,0)} = (1 - x_t) * (y_t) \tag{3.26}$$

$$Coeff_{(1,1)} = (x_t) * (y_t) \tag{3.27}$$

The intensity I_{warped}(x, y) is interpolated using Equation (3.28).

$$I_{warped}\ (x, y) = I(x, y) * Coeff_{(0,0)} + I(x+1, y) * Coeff_{(0,1)} + I(x, y+1) * Coeff_{(1,0)} + I(x+1, y+1) * Coeff_{(1,1)} \tag{3.28}$$

The result of interpolation is shown in Figure 3.23.



(a)

(b)

Figure 3.23 The result of interpolation (a) Before interpolation (b) After interpolation

3. 4. Color Consistency

3. 4. 1. Histogram Matching

It is a method of generating image that has a specified histogram. The source image is converted into the image that has a histogram that of target image. The color components of source image are mapped with the corresponding color components of target image.The color model used for this method is RGB. Histogram matching is used in each color components i.e. Red (R),Green(G) and Blue (B) individually. Here v_{in} and v_{out} are the color components of source and target, $h_0(v_{in})$ and $h_1(v_{out})$ are the histogram of color components of source and target and

$c_0(v_{in})$ and $c_1(v_{out})$ are the normalized cumulative frequency of the color components of source and target.

Algorithm

function*HISTOGRAM_MATCHING*(s_k, t_k) ***returns*** *map*

inputs: s_k = *source histogram*

t_k = *target histogram*

output: *map*

Calculate source normalized cumulative frequency($sncf_k$) *and target normalized cumulative frequency*($tncf_k$)

Find the $sncf_i$ *of value* s_i *for* $s_i>0$

Find the first value of $tncf_j$ *and corresponding* t_j *such that* $sncf_i<=tncf_j$

Map s_i *to* t_j

return *map*

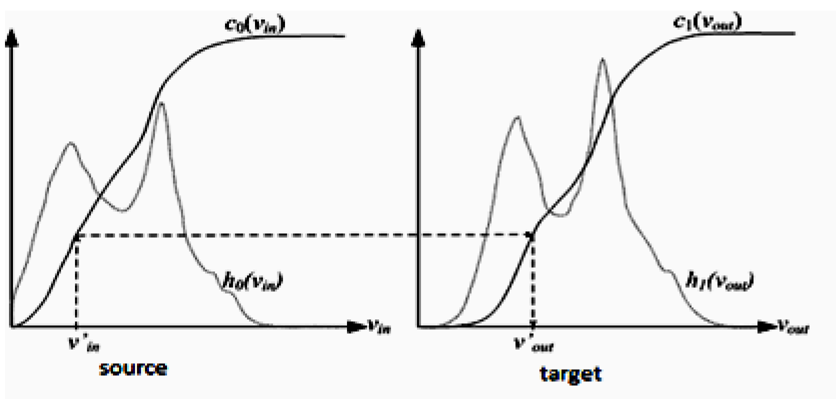


Figure 3.24 Histogram Matching

3. 5. Blending

As the source face is drawn on the target image, there is a clear distinction between the source region and target region, and there is also sharp transition in the edge of two regions, which makes the image look unrealistic. The sole purpose of blending is to remove this sharp transition.

3. 5. 1. Averaging

The sole purpose of blending is to reduce the transition effect from source color region to target color region in the replaced image. When the source face is extracted and planted in the target, their color ranges are unlikely to be the same. This brings an unreal situation in the image. Theoretically, this huge transition can be minimized by blurring a certain neighborhood region of the edges. Blurring causes the sharp transition region to spread in larger area. There may be several ways to do so, but we mainly focused on three methods.

1. Uniform Averaging
2. Weighted Averaging
3. Directional Gradient Averaging

1. In uniform averaging, all the surrounding neighbors are given equal weight in calculating the central intensity. A window with same value for all elements could be used as a filter. The effect is less strong transition from one region to another in the edge
2. In the weighted Averaging, more weight is given to the pixels that are nearer to the central point then the one away from it. This method brings more pleasing and realistic result. The faraway points are less affected then the points nearer to the edge.
3. The third method is based on the theory to give more weight to the pixels in one direction than the other, in order to for an intensity gradient effect from one direction to the other. For example, the left pixel is given more weight than the right and the top pixel is given more weight than the bottom of the central pixel. This will create a gradient effect from left to right direction and from top to bottom direction.

Algorithm

function*BLEND (faceonlyImage, replacedImage)* ***returns*** *the blended image*

> ***inputs:*** *faceonlyImage ← warped face ARGB Image of source*
>
> *replacedImage ← unblended but face replaced Image*
>
> *Boundary ← find boundary of face*
>
> ***foreach*** *points in Boundary*
>
> > *Apply neighbourhood function to calculate Blending Intensity*
> >
> > *Plot the intensity in new Image*
>
> ***return*** *new image*

3. 5. 2. Alpha Blending

The alpha blending refers to the blending of the transparency.The general process of replacing of the target face with the source face is by drawing the warped face region of source on the target image. But this will create sharp transition in the edges of two regions. The effect could be minimized with alpha blending. In this method the boundary of source face is extracted and a transparency gradient is formed from exterior boundary pixel to interior. The extracted face is often imperfect and the exterior part often is contaminated with background color and noises. So, making the exterior face part transparent helps the final image to be better.

Algorithm

function*ALPHA_BLEND(sourceFace, TargetImage, shiftPoint) returns finalImage*

  inputs: *sourceFace=source face region with all non-face region made transparent*

    targetImage=Image on which the target face is to be drawn

    shiftPoint=Point for shifting local coordinates of source to target Image

  output: *finalImage = Face Replaced Image with alpha blending*

  boundary←sourceFace-Erode(sourceFace)

  Interpolate(Transparency of boundary region)

  Draw alpha blended face in target

  return *final Image*

3. 5. 3. Intensity Interpolation Blending

In this method, the region of boundary where the transition is likely to occur, has been interpolated in X and Y direction. The effect of transition is already weak with alpha blending but is still visible. The boundary region is extracted. The intensity values of this region is re-calculated based on interpolation of the non-boundary up, down, left and right intensity values.This causes a smooth gradient image in the final image. Again, if the average skins tone of thecolor shifting also helps to minimize this effect. The algorithm for blending in vertical direction is given below.

(a) (b) (c)

Figure 3.25 (a) Initial Face Region (b) After Shrinking (c) Boundary

(a) (b)

Figure 3.26 (a) Before Alpha Blending (b) After Alpha

Algorithm

function *INTENSITY_BLEND(replacedImage, boundary Points)*

returns *blended Image with intensity interpolation;*

  for each *boundary points*

    Find the upper and lower non boundary Pixel distance and intensity

*Final_Intensity←(up.Intensity*down.distance+down.intensity*up.distance)/(up.distance+down.distance)*

Plot the intensity

***return** result*

(a)

(b)

Figure 3.27 (a) Before Boundary Interpolation (b) After Intensity Interpolation

3. 6. ImageShifting

After the source face is warped, a suitable position is needed to be found to replace the target face. A better face replacement is achieved when more face regions are matched in both source face and target face. The source face is first pasted on the target face in such a way that the center of the chin of the source face has the same coordinate as the center of the chin of the target face. Then, the pasted source face is shifted around the point of replacement (center of chin), to find the best position to replace the face at. A matching degree function $M(x, y)$for a pasting point $(x, y)$is used to evaluate the degree of matching, which is defined in Equation (3.38):

$$M(x, y)=\sum_{(i,j)\in I} h(F_s(i,j), F_t(i,j)) \tag{3.38}$$

where, $F_s(i, j)$ and $F_t(i, j)$ are binary face images which have value 1 only for face region pixel in source and target images respectively and I is the region of interest which is larger than the pasted region. The function $h(a, b)$ in Equation (3.39) is defined by:

$$h(a,b) = \begin{cases} +1, if a = b = 1 \\ 0, if a = b = 0 \\ -1, if a \neq b \end{cases} \tag{3.39}$$

For each point near the pasted feature point, the matching degree can be calculated. The point with highest matching degree will be chosen as the best position to paste the source face on the target face.

4. Results and Discussions

4. 1. Face Extraction

4. 1. 1. Preliminary Stage

4. 1. 1. 1. Skin Color Thresholds

Skin color thresholding is described in Section 3.1.1. The thresholds for skin color thresholding are as follows:

Y = 50 to 240

Cb = -50 to -5

Cr = 0 to 50

This process is not sufficient for face region extraction. The results of skin color thresholding clearly show that only the skin colored pixels are retrieved. Some important features inside the face are not extracted properly. Figure 4.1 shows that the eyes and eye brows are not extracted. Also, the result obtained after applying the threshold values may contain some noise. Finally, since this is a simple thresholding mechanism, the execution time is quite fast.

(a)

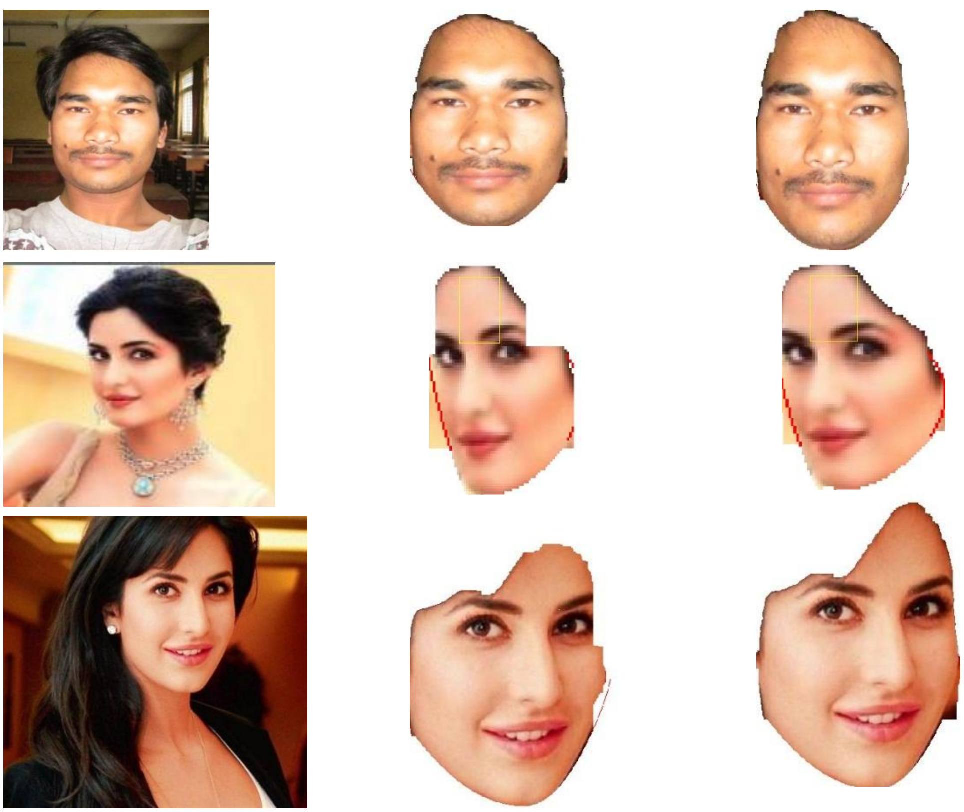
(b)

Figure 4.1 Result of using skin color thresholds for face extraction (a) Original Image (b) Image obtained after thresholding

4. 1. 1. 2. Adaptive Active Contour Model (Snake Algorithm)

The processing time of snake algorithm is slower than the skin color thresholding algorithm because the snake algorithm is an adaptive iterative process. The processing time depends directly upon the number of iterations. Greater the number of iterations greater is the processing time. However, if the image has face with strong edges and strong face boundary, then the output of the snake algorithm is better than the output of the skin color thresholding. The iteration value of the threshold should be within the range 1-15 and the threshold should be within the range 0 - 5.

The region of interest should be selected in such a way that it should include the overall face area (and should have minimum noise). If there is more noise, the snake may consider the noise as the edges and may adapt accordingly. The output of the snake algorithm is better in the area where the face boundaries and edges are strong and can be separated. The output of snake algorithm has less noise. The snake algorithm is to be used when the skin color cannot be used to distinguish the face like: when there is shadow in face but boundaries of face and edges are visible and can be separated. The output of snake algorithm and skin color thresholding is shown Figure 4.2.



(a) **(b)** **(c)**

Figure 4.2(a) Original image (b) Output of skin color thresholding (c) Output of snake algorithm

4. 1. 1. 3. Edge Detection

Our main purpose of using Genetic Algorithm was to find the thresholds for Canny Edge Detection. Active contour model or Hough transform could be used to join the edges in the edge maps.However, we found out that the lengths of the longest edges were similar for different thresholds.The thresholds obtained using Genetic algorithm also gave the same output. These were simply achieved by using adaptive active contour model and Canny-Deriche operator. So, we decided to not to use this approach in our project. Also the main disadvantage of Genetic Algorithm was time concerned with it. The output of Canny Edge Detection for various thresholds is as shown in Figure 4.3.

Figure 4.3shows that the overall longest edges of canny output have same length. The only difference is in the weak and strong edges. If the value of thresholds to the canny input is increased then the only strong edges remains i.e. long edges and when the value of threshold to the canny input is decreased then all the edges come to appear from very finest small edge to the long edges. But the overall lengths of the longest edges are same. So, even with the use of genetic algorithm the output of the canny edges cannot be changed. Hence, we did not use this approach in our project.

The lengths of longest edges in the output of canny edges are similar for all thresholds. The use of snake algorithm (adaptive active contour model) is more encouraged in this process because it is faster and gives good results than canny edge detection and genetic algorithm.

4. 1. 2. Curve Bounds

The next step in face extraction involved bounding the face region with curves. However, the algorithm did not find the perfect curve bounds for all types of faces. The final system includes only the chin curve bounds and the side curves and the forehead curve are not used.

The results obtained from the preliminary stage of face region extraction were further improved by using chin curve estimation. The chin curve was used as the boundary between the face and the neck. Every pixel bounded by the chin curve was taken to be a face region pixel. The improvement is shown in Figure 4.4.

4. 1. 3. Hair and Background Extraction

In the cases where the hair and the background colors were significantly different from the color of the face, hair and background region removal is not necessary for face region extraction. However, when hair/background is similar to the face, then the preliminary stage of face region extraction consists not only of the face region but also of hair/background region. In those cases, hair and background extraction (and then removal) becomes essential. The result of the hair region extraction and removal is shown in Figure 3.18.

4. 2. Image Warping

There are cases when the spatial alignment of face of source and target are different. The image warping helps in a spatial transformation that includes shearing, scaling and rotating an image. The output of image warping is shown in Figure 3.21.

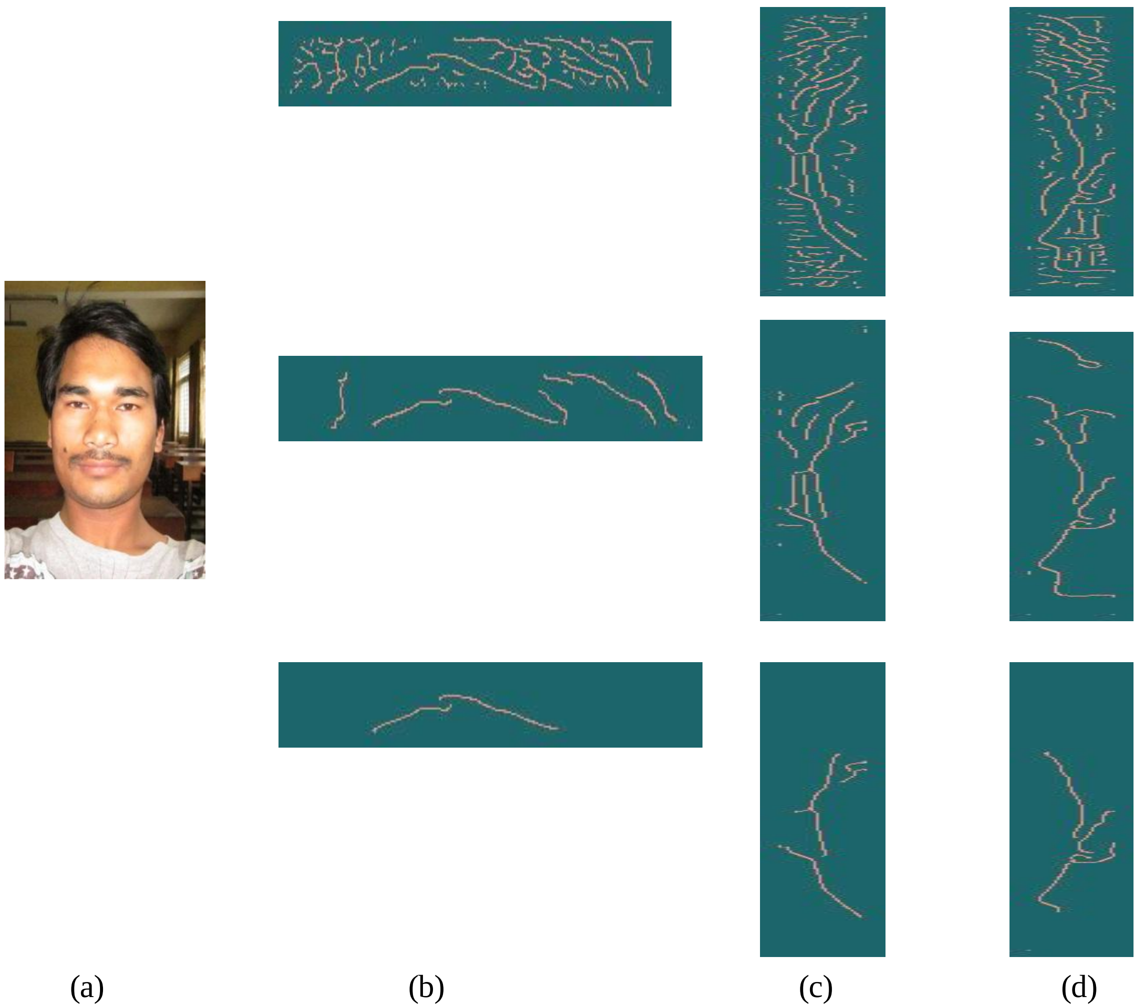
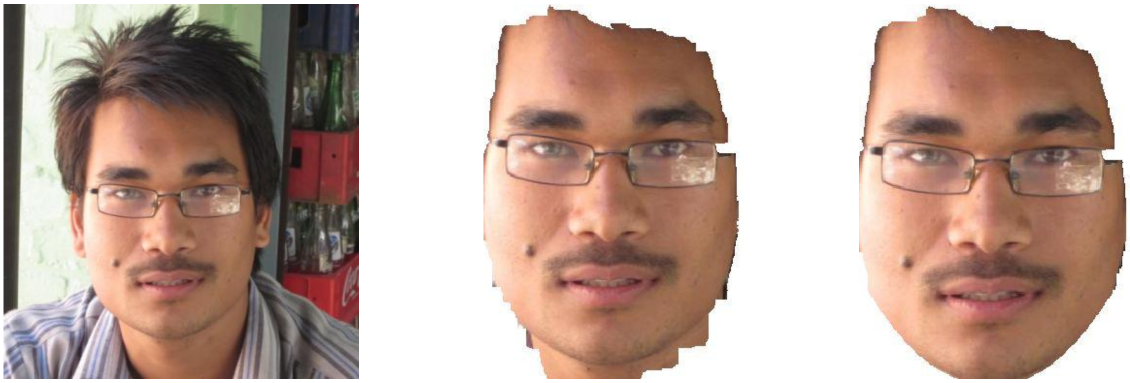


Figure 4.3: The output of Canny Edge Detection in different thresholds. (a) Original image (b) Forehead parts (c) Left face parts (d) Right face parts



(a) (b) (c)

Figure 4.4: The result before and applying chin curve estimate. (a) Original image (b) result before applying chin curve estimate (c) result after applying chin curve estimate

4. 3. Interpolation

When the pixels of the warped image are mapped to the source image, the mapped pixel positions are obtained as floating-point coordinates. However, the pixels in the source image have integer coordinates. So, the color values of the pixels in the warped image should be interpolated from the neighboring pixels of their mapped coordinates in the source image. The output of interpolation is shown in Figure 3.23.

4. 4. Color Consistency

4. 4. 1. Mean color shifting

For color consistency, the mean color of the source face was shifted to the mean color of the target face. However, this method did not provide realistic results. Using mean color shifting only the brightness of the replace face was changed but the there was no significant change in the skin tone.

4. 4. 2. Histogram matching

Because of the drawback of the mean color shifting, histogram matching was adopted. It provided realistic results in the cases in which the face regions did not have too much noise. The output of histogram matching is shown inFigure 4.5.

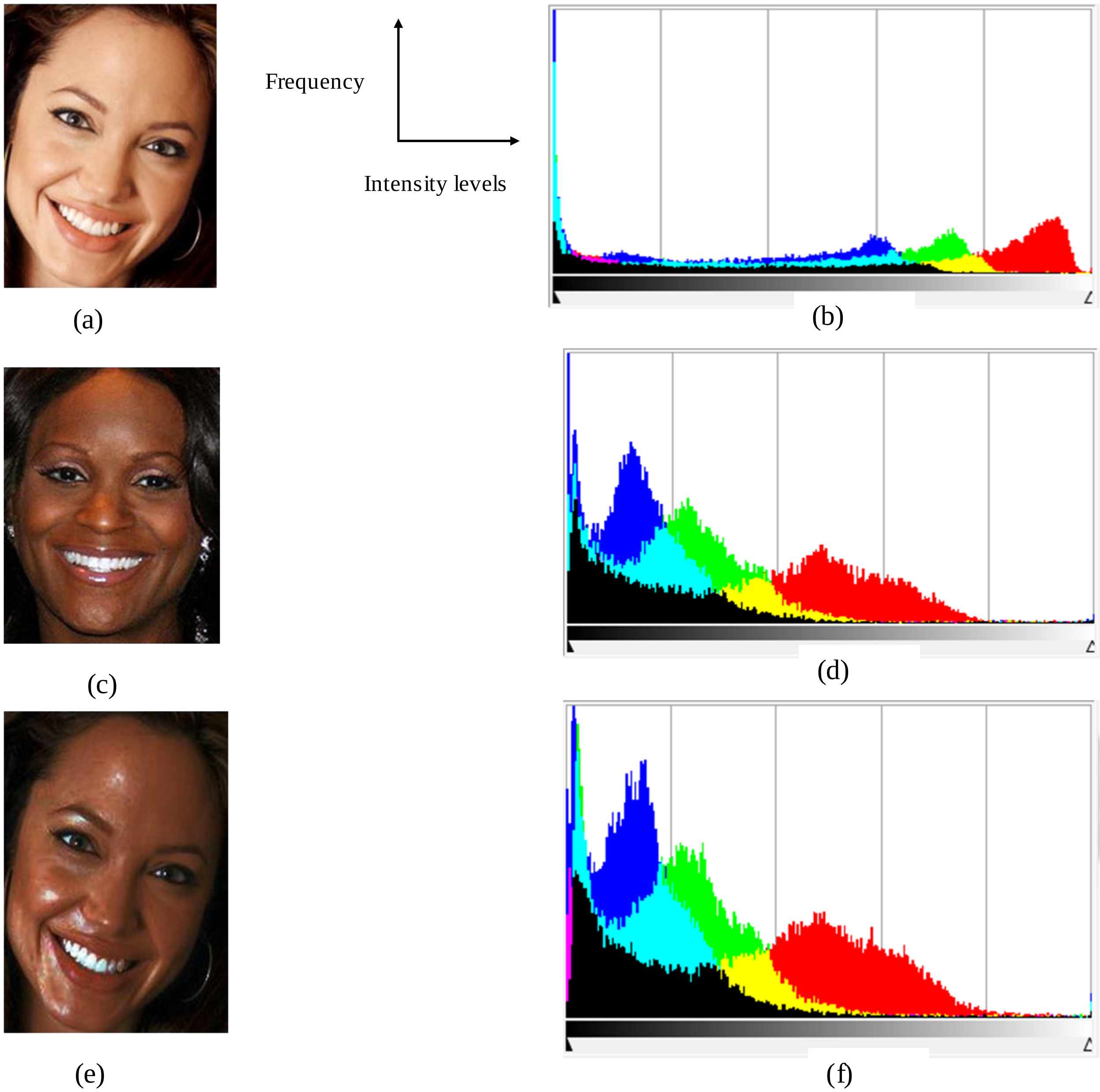


Figure 4.5 (a) Source (b) Source histogram (c) Target (d) Target histogram (e) Result (f) Result histogram

Blending

As the source face is drawn on the target image, there is a clear distinction between the source region and target region, and there is also sharp transition in the edge of two regions, which makes the image look unrealistic. The sole purpose of blending is to remove this sharp transition. Blending makes the replaced image realistic. The output of blending is shown in Figure 3.27.

4. 5. Image Shifting

After the source face is warped, a suitable position is needed to be found to replace the target face. A better face replacement is achieved when more face regions are matched in both source face and target face.

4. 6. Discussion of Final Results

In this section, the results of the face replacement are shown. The source face and target face have different sizes. Also, there are differences in skin tone of the source face and target face. The face regions were extracted by using skin color thresholds. Image warping and interpolation were applied so that source and target faces would have similar size and pose. Color consistency was applied so that both the faces would have similar skin tone. After replacement, the boundary between source face and target face was blended. In this way, these results were obtained.

Figure 4.6 shows the result of replacing the face of a girl with a boy. The skin tone of the face in the resulting image is different from the skin tone of the face in the source image.

(a) (b) (c)

Figure 4.6 Replacement of face of a girl with a boy (a) Source Face (b) Target Face (c) Result

Error! Reference source not found.shows that result obtained after replacing the target face ith source face.**Error! Reference source not found.** (a) and (d) show the source images, (b) and (e) show the target images and (c) and (f) show the results.

(a) (b)

(c)

(d) (e) (f)

Figure 4.7 Result of the system (a) First Source Image (b) First Target Image (c) First Result (d) Second Source Image (e) Second Target Image (f) Second Result

5. Conclusions, Limitations and Future Works

5. 1. Conclusions

In this project a semi-automatic system which can replace faces in still images has been built. The system consists of two major parts. The first part is concerned with the face extraction and hair/background extraction. The second part is concerned with the replacement of face.

Skin color thresholding, snake algorithm and edge detection were used to extract face regions. Skin color thresholding provides efficient performance but the result contains noise. Snake algorithm is inefficient in terms of computational time whereas edge detection does not provide good results when the edges are not sharp. The results thus obtained are then combined with the processes of curve bounds and hair/background region removal to extract the faces properly. Warping, interpolation, shifting, color consistency and blending were implemented for face replacement. The faces in still images with similar pose and less noise have been successfully replaced in the present work.

5. 2. Limitations and Future Works

There are many areas in which the system can be improved. The following list describes the limitations of the system and also describes how it can be enhanced in the future.

1. The system can only be used for faces in front view. It cannot be used for profile view. In the future, the system can be enhanced so that it can be used for the profile view also.
2. The source face and the target face could have different illumination and shadow. The system does not regenerate the illumination/ shadow conditions on the replaced face. This can produce the result to be unrealistic. So, the system can be enhanced by adding the feature of replicating the illumination/ shadow.
3. The system does not perform any tests if the faces can indeed be replaced accurately. The enhanced system could have a mechanism to check whether or not the result would be realistic, prior to the replacement process. For this feature, face pose estimation can be performed. A multi-classification neural network can be trained with positive and negative examples of pairs of source and target faces. The positive examples would include those pairs, for which the system can produce realistic results. The negative examples would include those pairs, for which the system cannot produce realistic results. Such a neural network could then be used to warn the users if inappropriate images are provided. This would save time and effort of the user.
4. The software system has not been optimized. Most of the algorithms which have been implemented can be optimized by choosing better data structures or otherwise.

5. Also, the facial expressions of the source face and the target face might be different. The system does not transfer the facial expression of target face to the source face. Replacement of facial expression is therefore a possible future enhancement.

6. References

1. *Face detection in color images.* **Hsu, R.L., Abdel-Mottaleb, M., Jain, A.K.** 5, s.l. : IEEE Trans. Pattern Analysis and Machine Intelligence, 2002, Vol. 24, pp. 696-706.

2. *Segmenting hands of arbitrary color.* **Zhu, X., Yang, J., Waibel, A.** 2000. 4th IEEE Conf. Automatic Face and Gesture. pp. 46-453.

3. *Robust face detection with multi-class boosting.* **Lin, Y.Y., Liu, T.L.** s.l. : IEEE Computer Society, 2005. Computer Vision and Pattern Recognition. Vol. 1, pp. 680-687.

4. *Face detection and eye localization by neural network based color segmentation.* **Fu, H.C., Lai, P.S., Lou, R.S., Pao, H.T.** s.l. : IEEE Signal Processing Society Workshop on Neural Networks for Signal Processing X, 2000. Vol. 2, pp. 507-516.

5. **Lin, C.H., Wu, J.L.** Automatic facial feature extraction by genetic. *IEEE Transaction on Image Processing.* 1999.

6. *Articulated pose estimation using brightness- and depth-constancy constraints.* **Covell, M., Rahini, A., Harville, M.,Darrell,.** 2000, Vol. 2. 438-445.

7. **Phillipe Andrey, Thomas Boudler.***Adaptive active contours(snakes) for the segmentation of complex structure in biological images.* Paris, France : s.n.

8. *A Face Replacement System Based on Face Pose Estimation.* **Chiu, Kuo-Yu.** s.l. : International Journal of Advanced Computer Science and Applications, December 2010, Vol. 1. 6.

9. *Support vector regression and classification based multi-view face detection and recognition.* **Li, Y.M., Gong, S.G., Liddell, H.** Grenble, France : s.n., 2000. Automatic Face and Gesture Recognition. pp. 300-305.

10. *Real-time head orientation estimation using neural networks.* **Zhao, L., Pingali, G., Carlbom, I.** s.l. : IEEE, 2002. International Conf. Image Processing. Vol. 1, pp. 297-300.

11. *Detection of anchor points for 3D face verification.* **Colbry, D., Stockman, G., Jain, A.** s.l. : IEEE, 2005. Computer Vision and Pattern Recognition. pp. 118-125.

12. *Automatic extraction of face contours in images and videos.* **Hsu Chih-Yu, Wang Hao-Feng, Wang Hui-Ching, Tseng Kuo-Kun.** s.l. : ScienceDirect, 2012, pp. 332-335.

Appendices

Screen shots of our GUI are shown below:

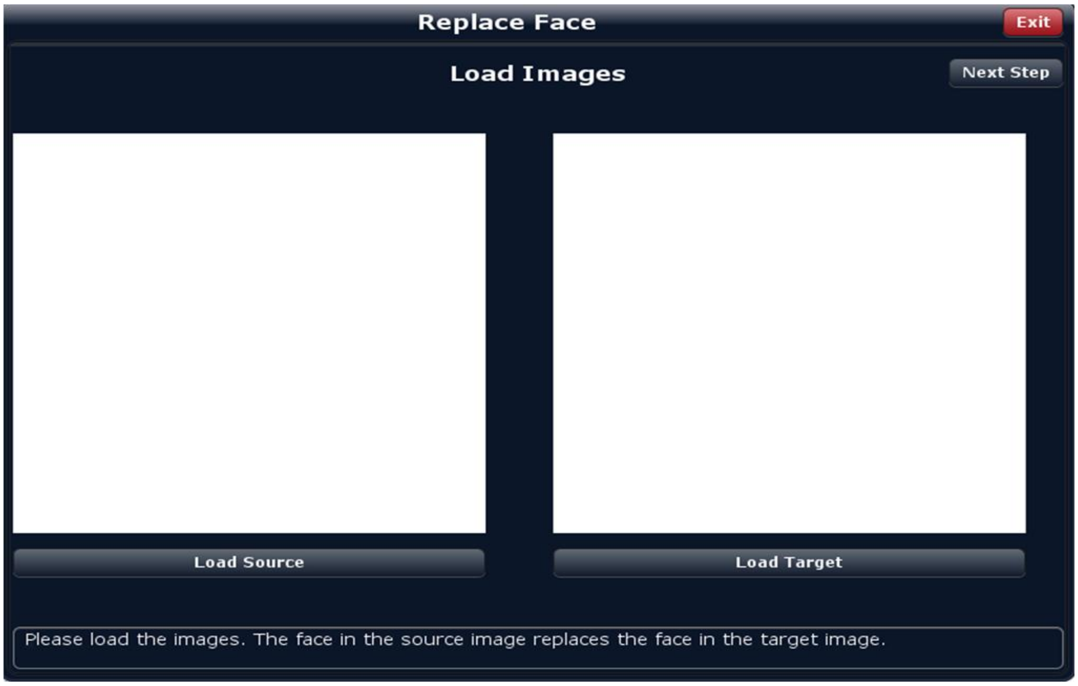


Figure 7.1 This screen allows the user to load source and target images

Figure 7.2 This screen shows the loaded source and target images

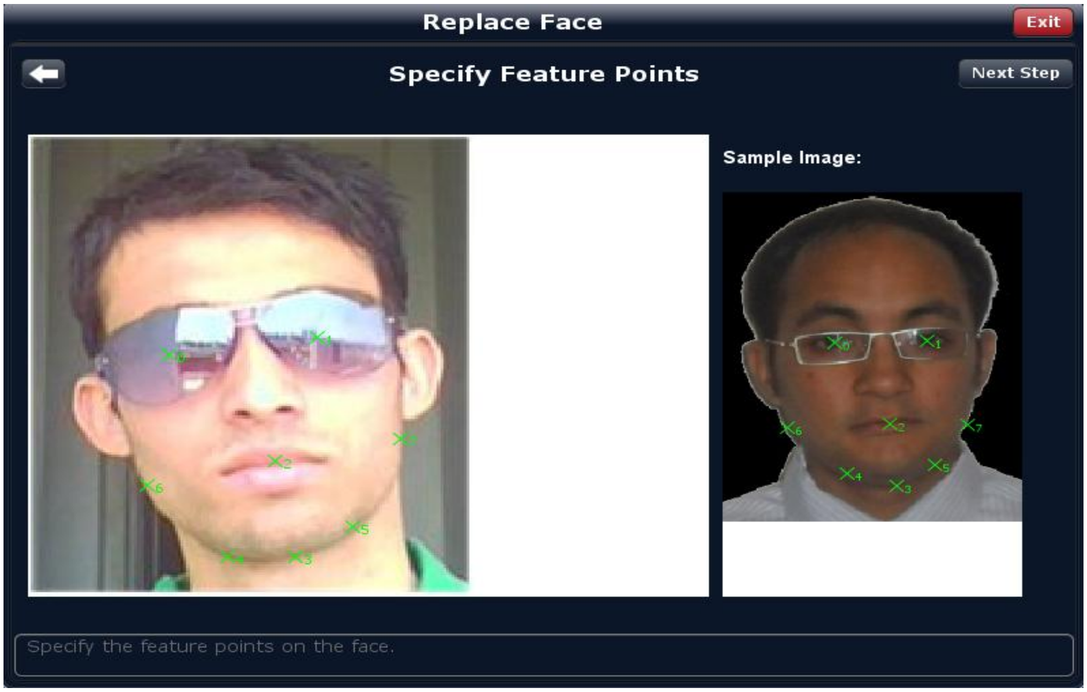


Figure 7.3 The feature points are placed as shown in the sample image

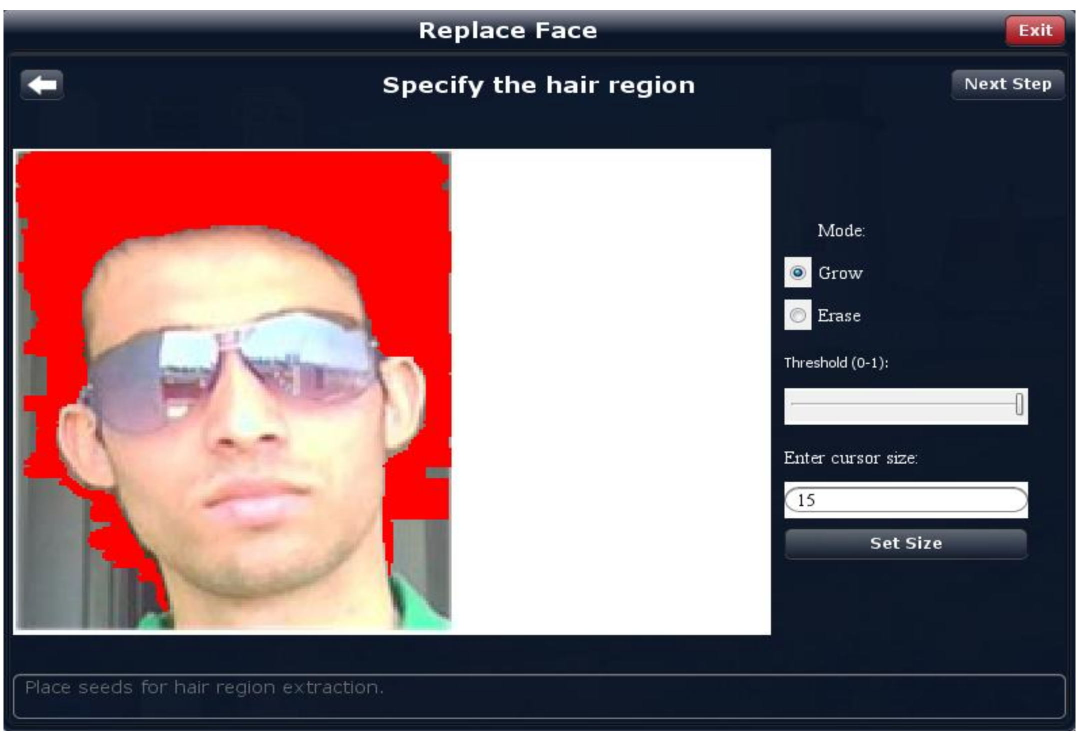


Figure 7.4 The hair region is grown after providing the seeds

Figure 7.5 It displays the extracted source face

Figure 7.6 The feature points are placed as shown in the sample image

Figure 7.7 The hair region is grown after providing the seeds

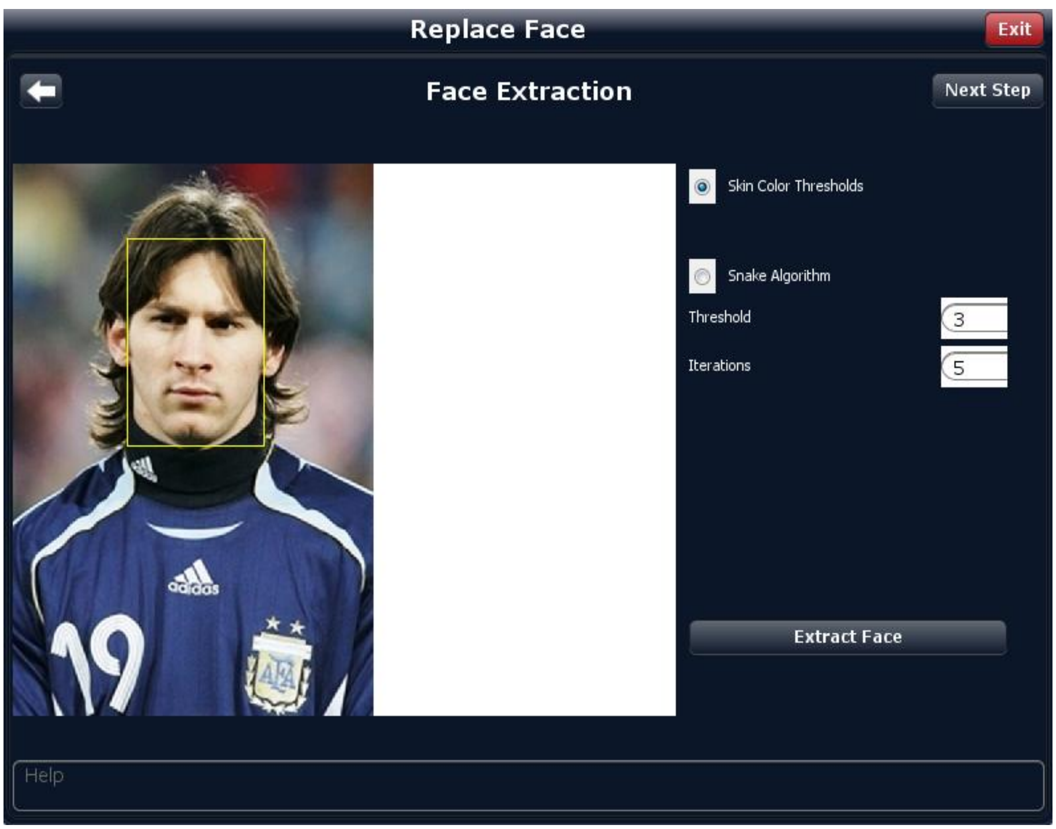


Figure 7.8 A rectangle is drawn around the target face

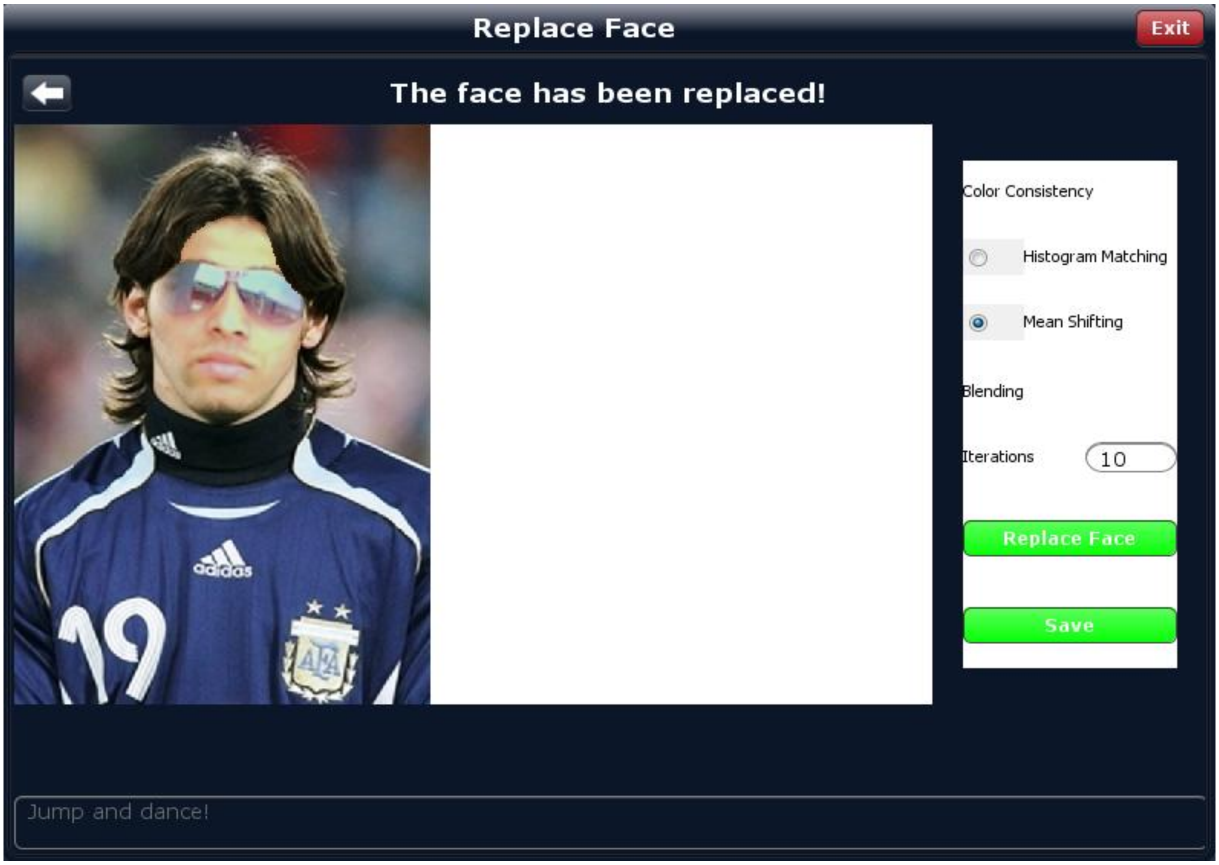


Figure 7.9 The face has been replaced