**Acknowledgement**

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

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

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 an automated “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 skin color thresholds and the hair is extracted by using seed region growing algorithm. The face to be replaced 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.

Contents

[I. Introduction 1](#_Toc330485717)

[A. Background 1](#_Toc330485718)

[B. Description 1](#_Toc330485719)

[C. Scope of Work 2](#_Toc330485720)

[D. Problem Statement 2](#_Toc330485721)

[E. Objectives 2](#_Toc330485722)

[II. Literature Review 3](#_Toc330485723)

[III. Methodology 4](#_Toc330485724)

[A. Face Region Extraction 4](#_Toc330485725)

[B. Chin Curve Detection 5](#_Toc330485726)

[C. Shrinking and Growing 6](#_Toc330485727)

[D. Canny Edge Detection 8](#_Toc330485728)

[E. Longest Edge Detection 10](#_Toc330485729)

[F. Hair Region Extraction 11](#_Toc330485730)

[G. Image Warping 12](#_Toc330485731)

[1. Least Square Method 13](#_Toc330485732)

[2. Mapping 14](#_Toc330485733)

[H. Interpolation 15](#_Toc330485734)

[I. Image Shifting 16](#_Toc330485735)

[IV. Results 17](#_Toc330485736)

[V. Discussion 18](#_Toc330485737)

[VI. References 19](#_Toc330485738)

**Table of Figures**

Figure III.1 The 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 4

Figure III.2 Face region extraction (a) Face region extracted using threshold values of YCbCr (b) Face region extracted after filling holes 5

Figure III.3 Curve estimation (a) Yellow marker shows the chin point (b) Chin curve is shown in red color 6

Figure III.4 Removing small image structures by stepwise shrinking and subsequent growing 7

Figure III.5 (a) Original image (b) Skin region (c)-(f) Shrinking (g)-(i) Growing (j) Output image 7

Figure III.6 : (a) Original Image (b) Separation of eyes (c) Probable face region cut out from the separation of eyes 8

Figure III.7 : The original face is divided into four parts as shown above. (a) Forehead image (b) Left face (c) Right face (d) Bottom Face 9

Figure III.8 : The edges of four face region. (a) Forehead image (b) Left face image (c) Right face image (d) Bottom face image 9

Figure III.9 The bottom face divided into four parts. 10

Figure III.10 Longest edge from bottom face 10

Figure III.11 The overall edge map of the face of longest and strong edges obtained after longest edge detection is applied. 11

Figure III.12 Our final expected output after applying Hough Transform or Active Contour Model. 11

Figure III.13 Seed Region Growing (a) The red rectangle contains the seeds (b) The red region is the result of seed region growing 12

Figure III.14 Image warping (a) Source image (b) Target image (c) Warped image 13

Figure III.15 Results of warping (a) Forward mapping (b) Reverse mapping 14

Figure III.16 The result of interpolation (a) Before interpolation (b) After interpolation 16

Figure IV.1 (a) Source image (b) Target image (c) Replaced image (without blending and warping) (d) Replaced image with warping and blending 17

# Introduction

## 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 3D models of the faces, while others just require 2D photographs. Most of these systems also require uniform lighting conditions

## 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 can be loaded via a user interface. It also requires the coordinates of the facial feature points on the source and target faces. The user is expected to provide the coordinates of the facial feature points via the user interface.

The system extracts the face region in the image by using threshold values of skin color. 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 shifts the color of the source face so that it becomes color-consistent with the target face. Alternatively, the color of the target skin can also be shifted to achieve the consistency. 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.

## Scope of Work

#### 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.

#### 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.

#### 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.

## Problem Statement

Face Replacement System aims to serve as a semi-automatic tool to the users so that they can replace face of one person with another person in a photograph. Currently, there are various image manipulation tools which allow the users to replace faces in photographs. However, the users need to possess specialized skills to achieve realistic results. This project aims to allow the users to replace faces in photographs with minimal user input.

## Objectives

1. To develop a semi-automatic system which can replace faces in photographs
2. To develop an easy-to-use user interface with different image manipulation tools
3. To implement various algorithms and compare their results

# Literature Review

Face replacement technology has a great use in the entertainment industries and special effects industries. Replacing one person’s face with that of another person in video sequences has huge implications. This project also came with the idea of video face replacement but considering the sophistication and limited resources, face replacement system in still images was chosen.

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

1. Facial feature extraction
2. Face pose estimation

The most general method to approach the facial features is to detect the face region by using the characters of skin color. Once the facial region is determined, statistical as well as geometric relation information is used to determine 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 estimation and improves the result of face region detection. 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 generation to get appropriate result. The fitness function selects the chromosomes based on the length of the longest edge. The fitness function takes time and the amount of chromosomes and the number of generation is large. So, it takes long time.

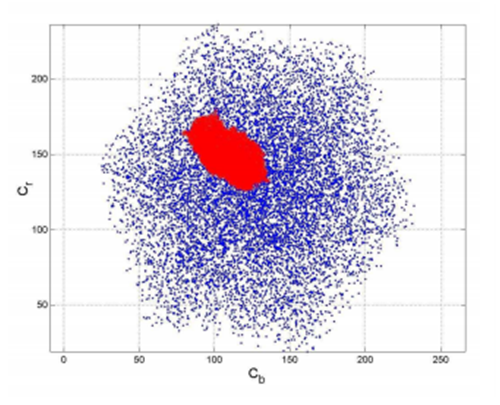
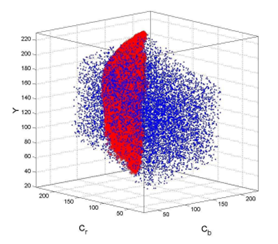
After the skin region is segmented, one of the necessary steps is chin curve estimation where five feature points are used to detect the curve to separate the face region from the neck region (7). Seed Region Growing can be used as a tool to estimate both skin region as well as the hair region. After 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 transformation of the source face region is done based on the spatial transformation coefficients obtained from corresponding feature points of source and target face. The second step is to shift the average color of source face to the target face to make it realistic. The best position of replacement is estimated which means maximum of the target is replaced with the source face. The edge of the segmented region is blended in the third step which increased the smoothness of the edges (7).

# Methodology

## Face Region Extraction

Face region detection is one of the most important steps for the face replacement. 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 III.1



(a)

(b)

Figure . The 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

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. For our purpose, we have divided the Y region into three parts. The part below Y value of 50 is totally cutoff. When the value of Y is in between, 50 and 120, then there is a rectangular region of acceptance. For higher values of Y, the rectangle is narrowed down.

The formulas for converting RGB image to YCbCr are mentioned in Equations (1)-(3).

|  |  |  |
| --- | --- | --- |
|  | Y=0.299 \* R + 0.587 \* G + 0.114 \* B |  |
|  | Cb=-0.16874 \* R - 0.33126 \* G + 0.50000 \*B |  |
|  | Cr=0.50000 \* R - 0.41869 \* G- 0.08131 \* B |  |

An equivalent binary matrix is constructed with each element representing either 1 (skin pixel) or 0 (non-skin pixel. The region is not 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 introduced with skin region detection 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 III.2



(a)

(b)

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

## Chin Curve Detection

Curve estimation is one of the processes where we approximate curves based on five feature points (Figure 2 Feature Point). 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.

|  |  |  |
| --- | --- | --- |
|  | = |  |

The curve of second degree is given by

|  |  |  |
| --- | --- | --- |
|  |  |  |

Figure . Curve estimation (a) Yellow marker shows the chin point (b) Chin curve is shown in red color



(a)

(b)

## Shrinking and Growing

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.

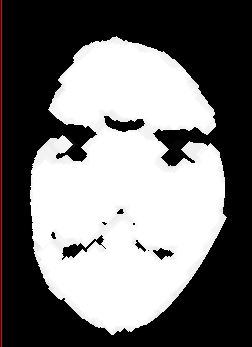
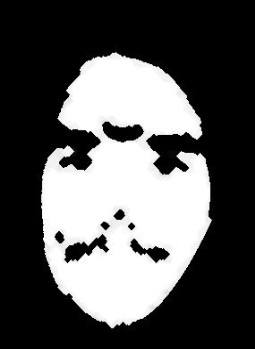
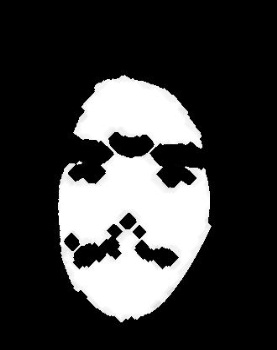
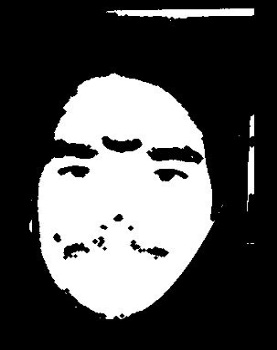
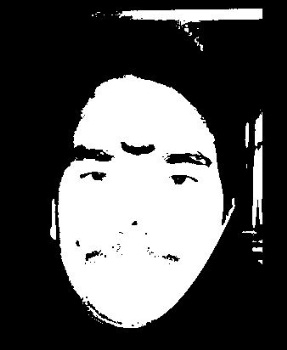
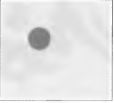
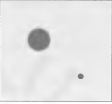
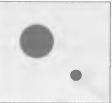
Shrinking process is achieved by peeling off a layer of a certain thickness around the boundaries. In our process, we have assumed the threshold value as 1 which denotes the number of layers to be shrunk. Shrinking removes the smaller structures step by step, and finally only larger structures remain. The final output is shown in Figure III.5 (f).

After shrinking the image, it is grown. The process of growing back the shrunk image 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 III.5 (i).

So all we need for this process are two types of operations. Figure III.5 (c)–(f) show the result of shrinking. Figure III.5 (g)—(i) show the result of growing.

The quick snapshots of what shrinking and growing does is shown in Figure III.4.

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



(a)

(b)

(c)

(h)

(d)

(g)

(f)

(e)

(i)

(j)

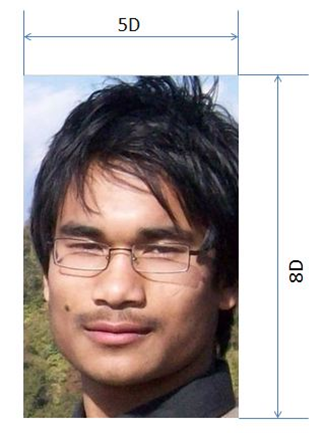
Figure . (a) Original image (b) Skin region (c)-(f) Shrinking (g)-(i) Growing (j) Output image

## 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’ (Figure III.6).

Figure . : (a) Original Image (b) Separation of eyes (c) Probable face region cut out from the separation of eyes

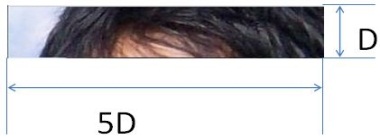
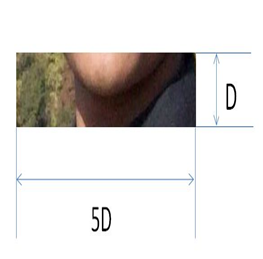
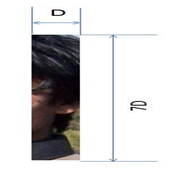
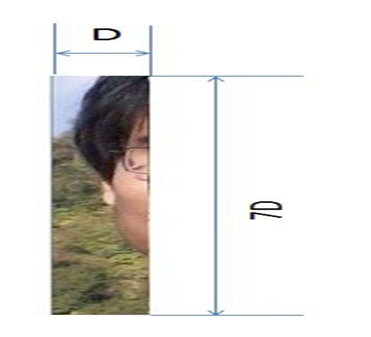


(a)

(b)

(c)

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 III.7.



(a)

(b)

(d)

(c)

Figure . : 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 III.8

(c)

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



(a)

(b)

(d)

## 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 III.9.

Figure . 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 III.10.

Figure . 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 in Figure III.11.

Figure III.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”. We have not implemented those algorithms yet. The expected outcome of this algorithm is shown in Figure III.12.

Figure . Our final expected output after applying Hough Transform or Active Contour Model.



## Hair Region Extraction

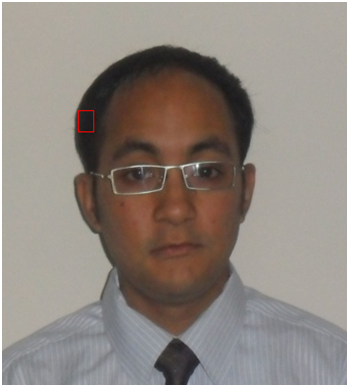
Hair region extraction is a necessary step in face replacement. If the user wants to replace the target hair with the source hair, then the hair region of the source image needs to be extracted and replaced.

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.

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 III.13



(a)

(b)

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

## 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 III.14.



(b)

(a)

(c)



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

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

|  |  |  |
| --- | --- | --- |
|  |  |  |

where,

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

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

mi's are the parameters to be determined

i=1 to 6

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

### Least Square Method

The values of (mi) 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 (7) and (8).

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |

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.

### Mapping

There are two ways to apply the image warping:

#### 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 Figure III.15.

#### 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 Figure III.15.

Figure . Results of warping (a) Forward mapping (b) Reverse mapping



(a)

(b)



## 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 |
| Iwarped(x, y) | = | Intensity of the pixel in the warped image |
| (xmapped, ymapped) | = | Coordinate in the source image which corresponds to the coordinate (x, y) in the warped image |
| (xsource, ysource) | = | The coordinate obtained after truncating the decimal parts of the components of (xmapped, ymapped) |
| Isource(xsource, ysource) | = | Intensity of the pixel in the source image |

The decimal part of the coordinate is given by Equations (9) and (10).

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |

Then, the values of the weights are given by Equations (11), (12), (13) and (14).

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

The intensity Iwarped(x, y) is interpolated using Equation (15).

|  |  |  |
| --- | --- | --- |
|  |  |  |

The result of interpolation is shown in Figure III.16.



(a)

(b)

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

## 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. 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 for a pasting point is used to evaluate the degree of matching, which is defined in Equation (16):

|  |  |  |
| --- | --- | --- |
|  | = |  |

where, and 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 in Equation (17) is defined by:

|  |  |  |
| --- | --- | --- |
|  |  |  |

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.

# Results

The final result of the system is shown in . The image (a) shows the source image. The target image is shown in (b). Figure (c) shows the replaced face without applying blending and warping. Finally, Figure (d) shows the result after applying blending and warping.



(a)

(b)

(c)

(d)

Figure . (a) Source image (b) Target image (c) Replaced image (without blending and warping) (d) Replaced image with warping and blending

# Discussion

Hence, in this project we built a system which can replace faces in still images. It requires the user to upload the source and the target images. The system requires the coordinates of the facial feature points. These coordinates are to be provided by the user. The system then detects the face region. The system warps the source image to match the pose and the size of the target face. The face is then replaced at an appropriate point. Finally, the boundary of the replaced face is blended with the target image.

# 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. *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.

8. *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.

9. *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.

10. *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.

11. *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.