Blending of Face Segments

Akshay Kajekar*¹, Shreyash Patel*², Sweta Chauhan*³, Prashasti Kanikar*⁴

Mukesh Patel School of Technology Management & Engineering, NMIMS University V. L. Mehta Road, Vile Parle (W), India

Abstract
A novel algorithm for extracting the regions of interest (ROI) from face images followed by subsequent stitching of the extracted ROIs of various faces to form a face image is presented in this paper. The final face resulting from the fusion of the original face with exogenous features lacks the characteristic discontinuities that would have been expected if only a replacement operation was performed. The novelty of the algorithm comes from the fact that the approach is not only simple to implement and understands, but it gives near perfect results.

To achieve its task, the system uses five modules: face detection, feature detection, segmentation, shifting and blending.

Keywords
Feature extraction, human facial proportions, matlab functions, rgb color space, segmentation and ycbcr color space

1. Introduction
Motivated by reports of increased self-confidence and driven by shifting cultural values, number of aesthetic maxillofacial plastic surgeries [1] has increased dramatically in the past few decades. This system offers patients and doctors an innovative way to visualize the outcomes of the procedure. Doctors may then provide patients with a before-and-after of the procedure helping patients significantly with their decision making process.

In this work, we solve the classical problems of face finding and feature detection; our system addresses and solves the problem of fusing a face with facial features belonging to another individual possible of different color. This is required in order to achieve a natural looking result.

Blending and mosaic ing features require a two level processing. On one hand, it is required that the shape of the original shape be changed to that of the desired object, and on the other hand it is desired that the color of the replaced feature be same as the original feature.

The paper is organized as follows.the basic concept is explained in Sec. II. The Sec. III describes the approach used for face detection, feature detection and extraction.

This sec.III.A also describes the normalization of color concept proposed by us.Sec. IV discusses the proposed approach for normalization of size and color shades. Sec. V shows the results .Finally, we present the conclusion in the Sec.VI.

2. Basic Concepts
Figure 1 shows the block diagram representing the system where:-
O: Original Face Image
D: Desired Face Image

![Block diagram of proposed system.](image)

The first step is to detect faces in the input images. The face detection module achieves this goal by using a a skin detection module and the skin color of normalized to a certain shade. Once the face is detected, the feature detection module finds the required regions of interest and then these regions are extracted by the extraction module.

The normalization module resizes the size of the ROIs. The blending module achieves smooth blending of the regions extracted such that the results look natural.

3. Related Work
Segmentation [1] is subdividing an image into its constituent regions or object. The level up to which the subdivision is carried out depends on the problem being solved.

In our approach, segmentation involves segregating the face into three regions which are:-
1. Eyes and the forehead
2. Nose and the cheeks
3. Lip and the chin

This is achieved by using the functions explained in the following sections.

The first 3 modules have been explained in detail in [1]. The overview is as follows.

A. Skin Detection
As shown in [1], each pixel in the inputted image is checked based on the below equations.
If the pixels satisfy the condition then it’s a skin pixel else it’s not a skin pixel.
The most noticeable range which was used by algorithm to detect the skin for H (hue) value is:
0.01 <= H <= 0.1 (1)
YCbCr space segments the image into a luminosity component and chrominance components. After experimenting with various thresholds the best result were found by using the following rule for detecting the skin pixel:
100 <= Cb <= 110 (2)
140 <= Cr <= 150 (3)
For each pixel of the image calculate:
And hue value of each pixel can be found by the built-in function rgb2hsv().
We try to normalize the color using the following approach: Normalization of Color
The algorithm of this step is as follows:
We want the image to have the same color as this average RGB value.
Consider:
Red=210; …(R)
Green =140; … (G)
Blue=100; … (B)
1. Consider only the skin pixels.
   Calculate average of R, G, and B of the image
   Consider avg_R as the average Red
   Consider avg_G as the average Green
   Consider avg_B as the average Blue
2. Subtract the difference between the R, G and B values stated above and the avg_R, avg_G and avg_B in each skin pixel of the image.
3. If the difference between R and avg_R:
   • is positive then subtract the difference with the avg_R and allocate that value to the pixel.
   • is negative then add the difference with the avg_R and allocate that value to the pixel
   Do the same to G and avg_G.
   Do the same to B and avg_B.
Hence the image is normalized to a particular skin pixel color.
The eye points can be detected using the horizontal proportion as the coordinates B and F are known. Hence, after the calculations, the coordinates C and E are known. From the vertical proportions, we can find coordinates A and H. the nose point G can also be computed from this vertical proportion as shown in fig. 6.

C. Feature Extraction

We now proceed to the next step of extracting the given feature from the face [1].

The algorithm for the feature extraction:

Refer to fig.3 for following coordinates.

Start

1. Blacken image till midpoint i.e. mid_x.
2. Blacken the eye parts.
3. Take feature extraction point. Save image as ‘head less.jpg’
4. Blacken the face till the nose_yvalue. Save the image as ‘bottom.jpg’
5. Compute the top part image i.e. (temp_img-headless). Save as ‘top.jpg’
6. Compute middle part image i.e. middle = headless-bottom. Save as ‘middle.jpg’

End

The following is the pictorial explanation of the concept used for feature extraction.

Step 1  Step 2  Step 4

Fig. 5 Horizontal Proportion

Fig. 6 Vertical Proportion

Fig. 7 Graphical representation of the feature extraction.

Fig. 8 (a) shows the output of the skin detection module. (b) shows the top.jpg image. (c) shows the middle.jpg image and (d) shows the bottom.jpg image.

I. Proposed Work

A. Normalization of Size

Length across the face can be computed by the ear coordinates i.e. B and F (fig. 3).

\[ l = F(x value) - C(x value) \] (4)

where \( l \) = length across face

Similarly \( b \) is also known.

Fig. 9 Length across Face of Top and Middle and bottom

Known variables:

- Length across face of top part=a
- Length across face of middle part=b
- Length across face of middle part (to resize with bottom)=c
- Length across face of bottom part=d
Resizing the Middle
We size the face parts, according to the top part length.
2 cases arise:-
- Case 1: If middle part is greater than top part.
  - Compute the difference between their lengths.
  - Decrease the size (columns) of the middle part by that much difference.
- Case 2: If middle part is smaller than top part.
  - Compute the difference between their lengths.
  - Increase the size (columns) of the middle part by that much difference.

Resizing the Bottom
1. Compute new length of middle and bottom.
2. Two cases arise:-
- Case 1: If bottom part is greater than middle part.
  - Compute the difference between their lengths.
  - Decrease the size (columns) of the bottom part by that much difference.
- Case 2: If bottom part is smaller than middle part.
  - Compute the difference between their lengths.
  - Increase the size (columns) of the bottom part by that much difference.

B. Positioning the 3 Parts:
Known variables:
From fig. 3 (coordinates), we know left ear coordinates i.e B.
Consider the top parts left aer coordinates as follows:
  a) Top values of left.
    i.e left_xvalue_top, left_yvalue_top.
  b) Middle values of left.
    i.e left_xvalue_middle, left_yvalue_middle.
  c) Bottom values of left.
    i.e left_xvalue_bottom, left_yvalue_bottom.

Consid only top and middle parts:

C. Color Shades
D. To change the shade of the face image:
Just change the Cb and Cr values:

<table>
<thead>
<tr>
<th>Cb</th>
<th>Cr</th>
</tr>
</thead>
<tbody>
<tr>
<td>105</td>
<td>163</td>
</tr>
<tr>
<td>95</td>
<td>160</td>
</tr>
</tbody>
</table>
E. Blending

The new face generated should be as natural as possible with the least seam and discontinuity. For this purpose blending needs to be performed. The blending technique involves superimposing the three different layers. The images of the three different sections are added to with each other. The image so obtained contains the image of the generated face which is approximately the required image of the face. However it is not completely natural. This is because the resultant image may contain overlapping regions or gaps. Both these problems need to be addressed. While superimposing the images there are gaps left between the lines of contact. It is important to fill these gaps. This filling of the gaps is achieved by performing ‘closing’ on the image along the lines where the merging of the different sections takes place. Closing takes place as follows:

\[ A \bullet B = (A \oplus B) \ominus B \]  

Where the ‘\( \bullet \)' indicates closing and ‘\( \oplus \)' and ‘\( \ominus \)' indicate dilation and erosion respectively. \( A \) is the image on which dilation needs to be performed which is the resultant image obtained from the superimposition of the three layers. \( B \) is the structuring element that will be used to achieve the purpose. Thus closing is a result of dilation followed by erosion on image \( A \) by structuring element \( B \). The other problem that needs to be addressed is the problem of overlapping as a result of the superimposition. Since we are employing the addition method the resultant image may contain overlapping regions. Hence we need to take care that addition is not performed at overlapping regions so as to avoid creating seams at mosaicing area which would otherwise become very bright.

Without Blending With Blending

4. Results

5. Conclusion

In this paper, we have presented a novel algorithm for extracting the regions of interest from frontal face images. At the same time, we describe details of a system that achieves fusion of different face segments of different individual - possibly of different skin color. The resulting face looks natural and gives near-perfect result. The possible applications where our approach could be applied are cosmetic surgery to extract features from face and also by detective agencies to identify people based on eye-witness accounts which is a good replacement to the manual technique now used.

References


[12] Rudra Pratap, Getting Started With Matlab 7