Facial Feature Selection and Blending of two Images
CS 390: Topics in Computer Science, Final Project

Hilary Fairchild-Carbonell, Lauren Smith
Boston College

Abstract
In this paper, we describe a method to take facial features from two inputted pictures, blending the two into one cohesive image. Our technique is based on template matching, utilizing known proportionality of standard human facial structures for more efficient searches. We demonstrate our superior searching method, requiring less user input, scanning and finding feature points automatically. These advances lead to increased functionality from a user standpoint with a less complicated interface, yielding more rapid results.

1. Introduction
Face blending software available on the internet today is limited. While experimenting with various sites, we noticed shortcomings in multiple blending platforms. Each program, while unique, was similar in the amount of preparation necessary to compute a blended photograph. Inputting the pictures came first, followed by an extensive routine, clicking up to 32 points on the face, specifying locations of all features of the face – this process being repeated for each image. We designed a program to overcome significant issues with the previous attempts, specifically the time delay to set up and prepare the images. Our technique utilizes automatic searching of the image, detecting the eyes, nose, and mouth of each photograph. Producing an accurate model comes from blending these found features, smoothing the two face shapes and tones into one, and reinserting the blended features into averaged locations in the image. We will describe our process in more detail in the following sections.

2. Strategy
The proposed technique is broken down into two tasks to accomplish: find the
features in both photos, and blend and reinsert. The following specifies how we went about accomplishing both.

2.1.1 Feature Finding
Our model is centralized around the concept on automatically generating the features of each image, thus reducing input time and easing the process altogether. We structure our technique around multiple functions, each searching for specific facial features, using relative locations from the previous found features to tailor the search region to a smaller, more accurate region. Relying on six inputs of the face: chin, left ear region, forehead, right ear region, left corner of mouth and right corner of mouth, we can establish an expected area to search for the features, eliminating extra time searching extraneous space unlikely to contain important parts of the image. Functions getEye(), getNose(), findNose(), and getMouth() are all used in a sequential fashion to provide inputs to the following functions.

Eyes are the most distinct feature on the face, in terms of visibility in a matrix-based image. The most efficient means we found to locate the two eyes in a photo were to use the MATLAB function edge() on a template of a standard image of an eye. Because of the circular nature of the iris, the eyes are especially noticeable in an image after applying the edge effect, (see image 2). Sweeping this binary pattern around the smaller region of the face, defined by the input points, we can easily recognize and select the areas around the left and right eyes. Repeating this on both pictures, we take the values of the eyes and their locations to use later when synthesizing both together.

Using the inputted mouth points, we extend this region vertically, encompassing the entire mouth on each photo. The location of this feature is computed. Then, the nose is determined to be between the eyes and mouth. This area is mapped out in the function getNose(), and then searched for the best match to a standard template image of a generic nose, using the findNose() function. Once this is found, the values and location here are stored to variables, utilized later when assembling the entire photo.

Figure 2: This facial view was created using the Matlab function edge() on the inputted image, and likewise with the template eye image seen below. Because of the unique shapes seen in the images, using this template to sweep over the image becomes a timely, accurate method for locating the eyes.

Figure 2.1: Template of eye edges
2.1.2 Blending and Combining Features, Assembling Picture
Before beginning any steps manipulating the features, we first blend the two faces, to create a palate into which we insert the blended features. This is done by averaging the pixels of the faces, creating a more uniform, blended look of the face shape, skin-tone, and appearance. Also, at this point, we have collected the facial features in each photo, associating each with coordinate locations in the matrices. Because each photo can be taken from different scales, as well as the general fact that feature size can vary person to person, it was imperative to resize the features to the same dimensions in order to replace them. This is the first step in combining any features, followed then by the averaging of the pixels in the two selected regions. This appears to blend the photos into the mean resulting feature. Using the known locations, we can reinsert the newly blended features on top of the new face. Since the face is blended in the same fashion, the features blend back into the skin, seamlessly. This assembles the reproduction of a blend between the two images from the initial image upload.

3. Comparative Results
Following our procedure and approach, the results generated were impressive. Even by decreasing the input required by a very significant amount, quality images were created. Our technique seems to work most effectively for individuals with similarly aligned faces, producing the best results in these situations. For only selecting six input points, the method improves speed of photo set-up and minimizes time lag to generate the final result. In comparison to existing works, ours provides results in a more timely manner, with equivalent accuracy demonstrated with our test images.

Figure 3: The results of comparing the current version type to our prototype design. Above, the faces are blended smoothly into one at the website www.MorphThing.com. This site requires 36 input points from users. Below: our model utilizes a mere 6 points from the user, blending into an equivalently accurate photograph.