Image Morphing

Suster Elena-Roxana
Technical University of
Cluj-Napoca
Email: Roxana.Suster@student.utcluj.ro

***Abstract*—Image Morphing is one of the most powerful Digital Image processing technique, which is used to enhance many multimedia projects, presentations, education and computer based training. It is also used in medical imaging field to recover features not visible in images by establishing correspondence of features among successive pair of scanned images. This paper presents what morphing is and implementation of Triangulation based morphing Technique. IT analyze both morphing techniques in terms of different attributes such as computational complexity, Visual quality of morph obtained and complexity involved in selection of features.**

I. Introduction

Image morphing belong to image processing defined as the animated transformation of an image into another, which is an essential in many multimedia systems and medical imaging field applications. Morphing involves the image processing techniques of cross-fading and warping.
Warping determines the way each pixel in source image is mapped onto pixel in the destination image and some important pixels need to be specified in two images. The motion for other pixels is obtained by extrapolating the information specified by the control pixel. Morphing is simply cross-fading applied to warped image. Image morphing process is based on single line and multiple lines algorithms. The morphing effect is widely used for various tasks such as mixing parent's photos to predict how their future child will look like and also in movie animations, gaming industry and multimedia projects. Even Disney animations are made using morphing for speeding production.

II. Related Work

Bhumika et al. [1] propose an algorithm that is divided in two steps :
1. Compute the desired displacement of all the pixels in the source image.
2. Resemble the image to create the output image.
Features morphing gives high level of control to programmer on output results. Computational complexity depends upon feature lines multiplied by total number of pixels of image. It decreases morphing results which can be corrected by adding or deleting some lines.
Delaunay [2] another noticeable piece of work related to the subject is using the Delaunay Triangulation. This algorithm refers to giving a set of points in a plane, a triangulation referring to the subdivision of the plane into triangles. Delaunay triangulation stands out because it has some nice properties such as that no point is inside the circumcircle of any triangle and also it has no favour for "skinny" triangles (i.e. triangles with one large angle).

III. Method

The idea behind Image Morphing is rather simple. Having 2 images (A and B), we want to create an in-between image C by blending image A and B. The blending of the images A and B is controlled by a so-called parameter alpha, which is between 0 and 1 (if alpha=0 then the resulting image will look alike with picture A and if alpha=1 then the image will be exactly the same as picture B). In order to morph the image A into image B we must establish a pixel correspondence between the 2 images. Supposedly, we have found these correspondences, we need to calculate the location of the pixel in the resulting image $(\mathrm{x}_m, y_m) using the following equation$:
$x_m = (1-\alpha)x_i + \alpha x_j$
$y_m = (1-\alpha)y_i + \alpha y_j$
$(Eq1)$

Then, we need to calculate the intensity of the pixel using the following equation:

$$M(x_m, y_m) = (1-\alpha)A(x_i, y_i) + \alpha B(x_j, y_j)$$

(Eq2)

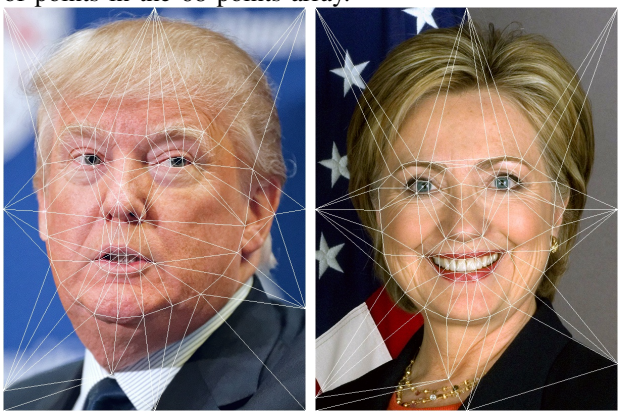
Steps used in the project :
1. Find the point correspondence using facial feature detection
2. Delaunay Triangulation
3. Warping images and alpha blending

1. Find point correspondence using facial detection

I used dlib to detect the corresponding points (68).

After identifying the points corresponding to the main characteristics of the face, we save their position into a .txt file which will be further used.

2. Delaunay Triangulation From the previous step I have obtained for each image 68 points. Now, on each of this sets we perform Delaunay Triangulation. The result of Delaunay triangulation is a list of triangles represented by the indices of points in the 68 points array.



3. Warping images and alpha blending Now, we have to intelligently blend the two images. As mentioned before, the amount of blending will be controlled by a parameter alpha. Create a morph using the following steps.

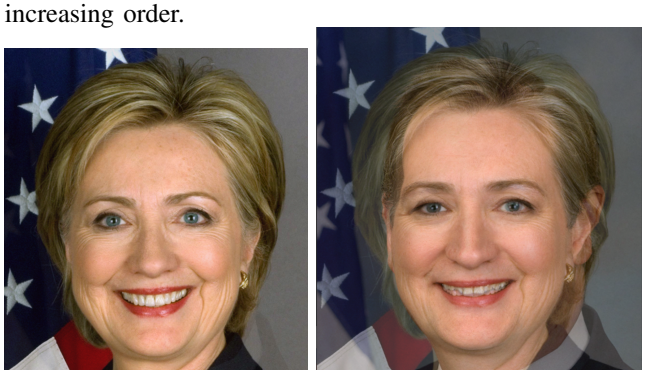
1. Find location of feature points in morphed image : In the morphed image A, we can find the locations of all 68 points using equation (Equation 1).
2. Calculate affine transforms
3. Warp triangles
4. Alpha blend warped images

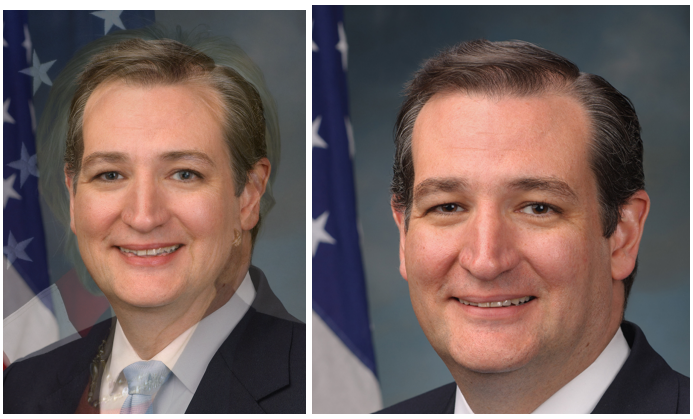
The resulted images for two different values of alpha:
1. For alpha = 0.5

2. For alpha = 0.8

IV. Evaluation and Results

As we could see, the algorithm performs the morphing quite well, increasing the resemblance with the second picture as we grow the value of the alpha. Above we could see the morphing for two values of the the alpha (0.5 and 0.8), and as it grows up to 1 it will resemble even more with the morphism we want to obtain. Above the morphism was performed on the pictures of Donald Trump and Hillary Clinton, and now we will try to perform it between the pictures of Hillary Clinton and Ted Cruz. We are going to perform for such transformations, giving to the alpha parameter the following values : 0.1, 0.5, 0.7 and 1. The results are listed below in increasing order.





V. Conclusion

In this paper we have managed to recreate the morphing algorithm used in different movies to create special effect by using the Delaunay Triangulation. We also took each step in order to see intermediate results such as discovering the facial traits used and saved their position in a .txt file in order to create on each image the necessary triangles which are going to be mapped. We also observed that by increasing the values of the alpha we can do a total morphing, for the value of 0 keeping the initial photo and for the value of 1 having the desired morphing.
This paper can be improved by creating a gif where the value of alpha will increase automatically from 0 to 1, in order to see in real life and in only 1 running the whole process of morphing.

References

[1] Delaunay Triangulation using OpenCV - https://learnopencv.com/delaunay-triangulation-and-voronoi-diagram-using-opencv-c-python/
[2] Face Morphing using OpenCv - https://learnopencv.com/face-morph-using-opencv-cpp-python/
[3] Facial landmarks - https://www.pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/