DIGITAL IMAGE PROCESSING

ASSIGNMENT 2 - IMAGE MORPHING

-SAZID ALI 2022AIM1013

MORPHOLOGY PROCEDURE

Step 1: Obtain tripoints.

In the first part of the assignment, tripoints are read from the given '.txt' file

In the second part of the assignment, I used 'dlib' library's **shape_predictor_68_face_landmarks.dat** model. This model gives 68 feature points corresponding to the face of the human.

Step 2: Triangulation of image1 and image2

I used the Delaunay function from "scipy.spatial" library. The Delaunay function gives the set of all non-overlapping triangles coordinates.

I used Delaunay only for the first image and then with the help of the coordinate of triangles obtained from the triangulation of the first image, I will do triangulation of the second image because we want one-to-one correspondence triangles.

Step 3: Obtain tripoints for the intermediate image using the formula

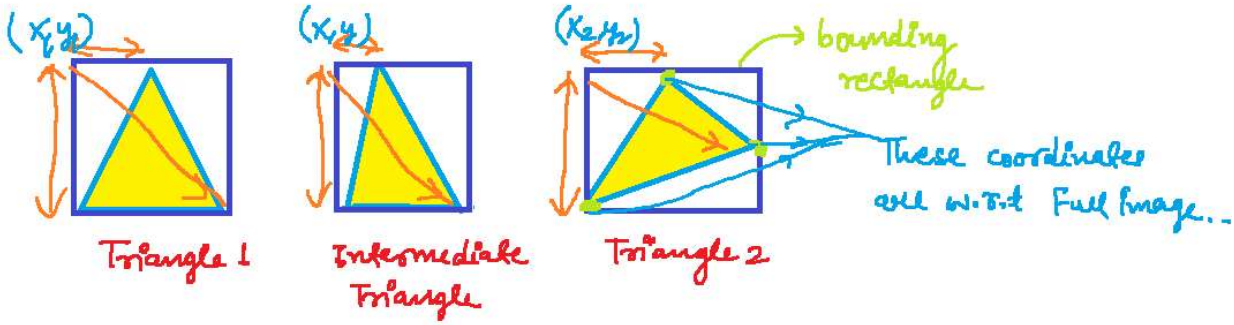
(x,y) = (1-alpha)*(x1,y1) + (alpha)*(x2,y2)

Step 4: Do triangulation of the intermediate image as well using image1 triangulation points.

My tripoints at particular indexes are pointing to the same facial feature in image1, image2, and the intermediate image

Step 5: Read triangles one by one from image1, image2, and the intermediate image so, at a time we will have triangle1 corresponding to image1, triangle2 corresponding to image2, and triangle3 corresponding to the intermediate image

Get the bounding rectangle that bound the triangle of all images.



We need to fill the intermediate triangle using triangles 1 and triangles 2 but triangles can be of different shapes. So, we need to apply an affine transform between the intermediate image and image1 and then the intermediate image and image2.
When affine transformation will be done then it will be with respect to the bounding rectangle because, at a particular time, our area of interest will that rectangular images that are bounding triangles. But coordinates are given with respect to the full image. So, src and destination points for affine transformation will be offset values that will be obtained by subtracting all triangle coordinates from the top left corner point of the rectangle.
After transformation, we have to mask the triangle area also because we want to fill the pixel values which are inside the triangles. After masking we can fill the triangles.

Result:

Morphed Image using manual tripoints

Morphed Image using Facial feature detection

Note: I have added a GIF file in the zip folder. GIF file with the name ‘morphed1.gif’ is of part1 and ‘morphed2.gif’ is of part2