FACE MORPHING

Project Aim : To learn the concept of morphing , linear interpolation , affine transformation and implement them to form intermediate images between two images (src,dest) in python using open cv.

Name	id
Nallamaddi Sritej Reddy	17ucs094
Lohit Reddy Arrabothu	17ucs082

Overview

Morphing in general means undergoing a gradual process of transformation. In this project we are meant to make intermediate images using a set of control points in source and destination images in a way that a smooth transformation is made showing change of source into destination image .

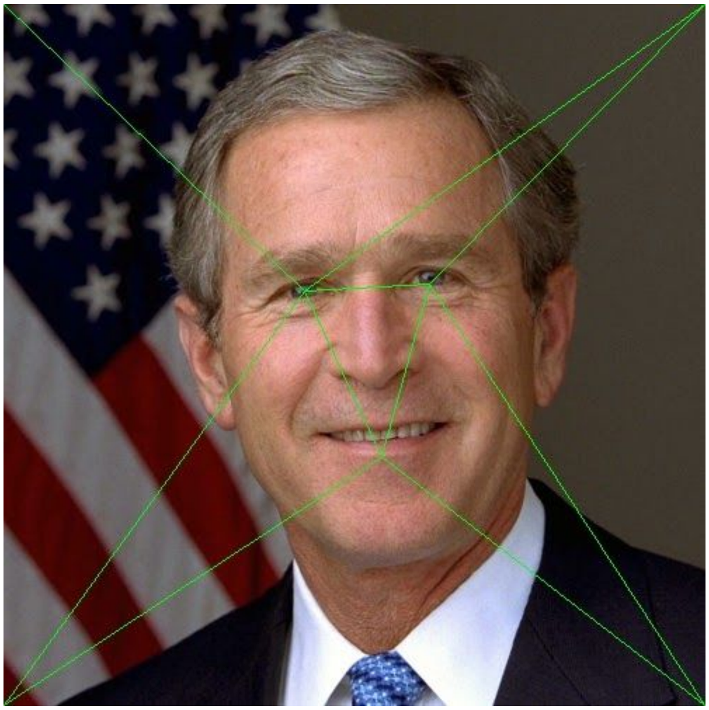
Application : This technique is often used by the Animation industry . Movie makers also use morphing in order to make special effects like converting a man into an animal etc.

Input:

(source Image of size 500*500)

(Destination image of size 500*500)

Methodology :



(Triangulation is made in this way
)

8 triangles are made using 7 control points ,edges of triangles in image are control points (4 corners and remaining 3 are 2 at eyes 1 at chin).

Total 9 intermediate frames and a video is made by showing transformation from source to destination image .

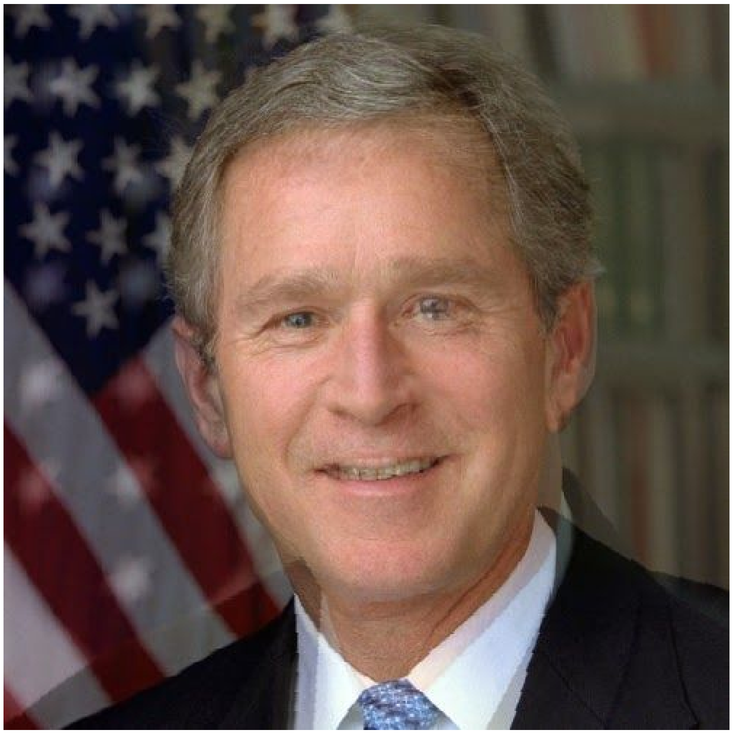
Out of 7 control points 4 are corner points which remain unchanged in all images so Mouse Clicks are used mainly to capture the remaining three control points i.e 2 at eyes and 1 at chin.

Using Delaunay Triangulation also 8 triangles are formed with control points being the same but triangles changed a bit . Code for the same is written inorder to accept any number of triangles returned by the Delaunay function.

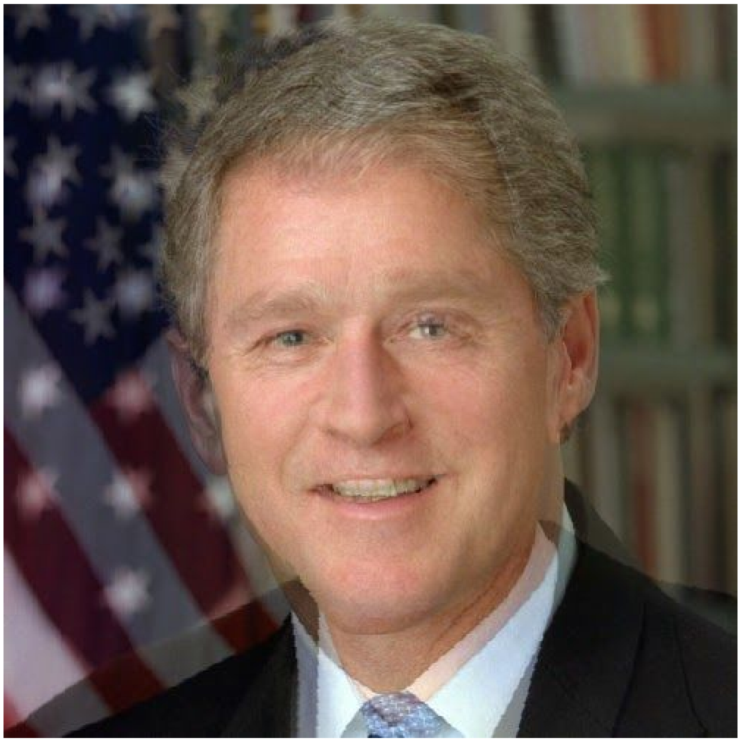
The method followed to produce the intermediate images is in this way:

1. For each point in an intermediate image a function is written to find which of the triangular region it belongs to.
2. After finding which one of the 8 triangles it belongs to, affine coordinates i.e (alpha ,beta) are calculated using the 3 control points that formed the triangle.
3. Since affine coordinates remain the same for corresponding point in source and destination color interpolation is done to calculate the color value of point in intermediate image.
4. So for each point of the intermediate image we get a color value to form the complete image and the same method is followed for each of the intermediate images .
5. Finally using all intermediate images a video is made using cv2.VideoWriter().
6. After Completing this whole task , work has been done to capture coordinates of control points using mouse left button click.
7. Delaunay Triangulation i.e automatic triangulation has been done at end where a function returns a list of points forming a triangle.

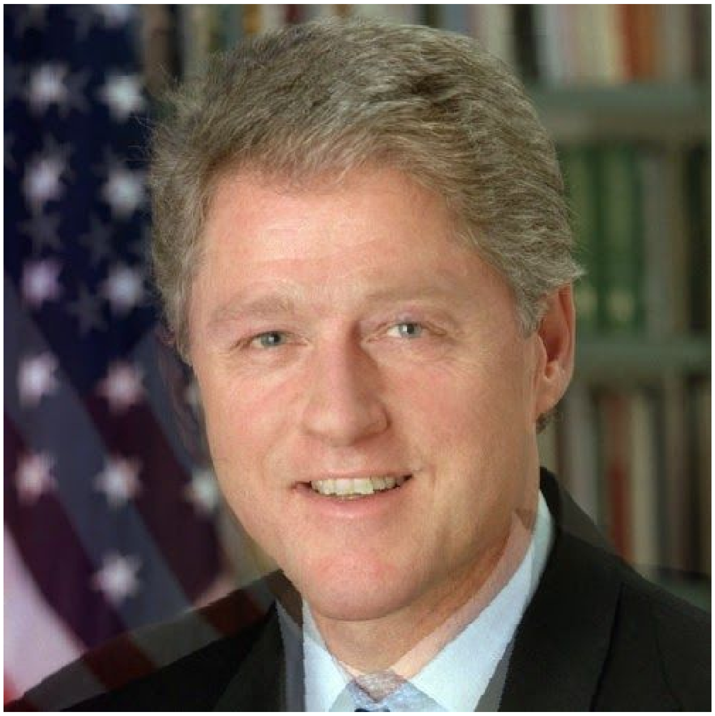
Output:



3rd intermediate frame



5th intermediate frame



7th intermediate frame

As we can clearly see from these outputs that the 3 rd frame has features very near to the source image , 5th frame has like equal features from both source and destination and 7th frame has features very much close to destination image.

Conclusion:

The first step I did to start the project was noting down the coordinates of control points in both source and destination images using microsoft paint . After completing all the code images were getting generated erroneously so each and every function was checked carefully by giving manual values to variables .Then it was known that all the functions were correct ,so i went through the coordinates once again then came to know that python is reading points in inverse fashion of Microsoft paint i.e if a point has coordinates (a,b) in Microsoft paint then python reads it as (b,a) .So by just inverting X and Y coordinate values and running the code, images were generated perfectly .

The second step was with given vertices of triangle finding whether the point is inside the triangle or not. Using geometry and areas of triangle concepts this step was completed without facing any problem.

Formula for finding affine coordinates (alpha,beta) were calculated on paper and directly placed in code .

Initially only 3 intermediate frames were generated after that inorder to make video showing transformation smooth they were increased to 9.

Inference that can be made from this is that more the intermediate frames the better will be the transformation video.