Face Morphing

Project Aim

Face Morphing using affine transformations and delaunay triangulation

Team member : Pawan Kumar Jain (17ucs108)

References : i) Pyimagesearch Article : How to get mouse clicks
ii) Learnopencv Article , IICT : Logic and related functions of Delaunay Triangulation

Overview

Morphing is a special effect in motion pictures and animations that changes (or morphs) one image into another through a seamless transition

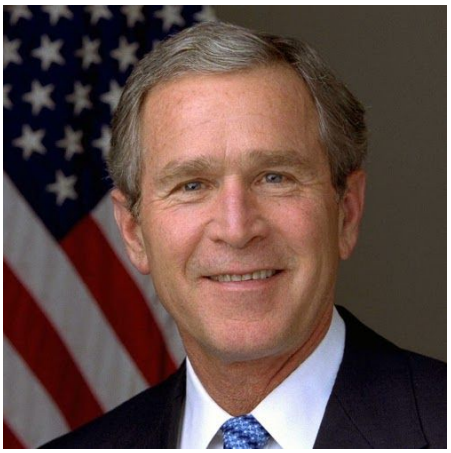
Uses : By the animation industry to transform one form into another seemingly seamlessly

Affine Transformation helps to modify the geometric structure of the image, preserving parallelism of lines but not the lengths and angles. It preserves collinearity and ratios of distances

Uses : This technique is also used to correct Geometric Distortions and Deformations

Input :

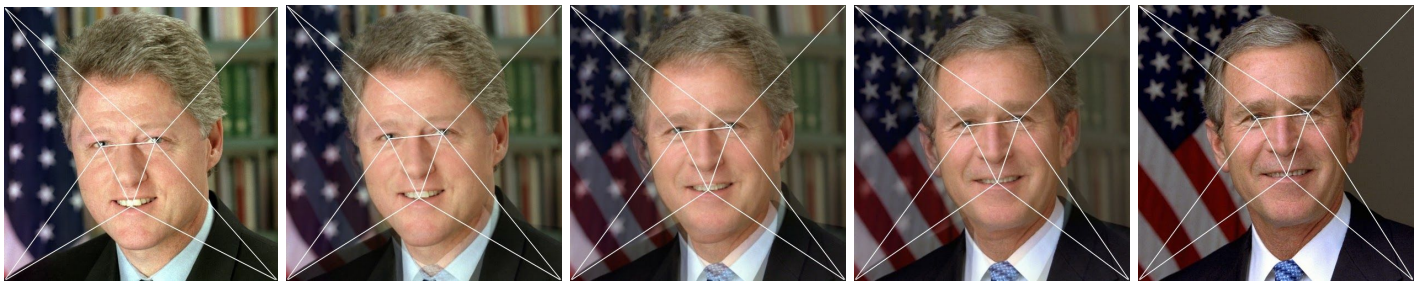
Size 500x500
Extension (JPG)



Size 500x500
Extension (JPG)

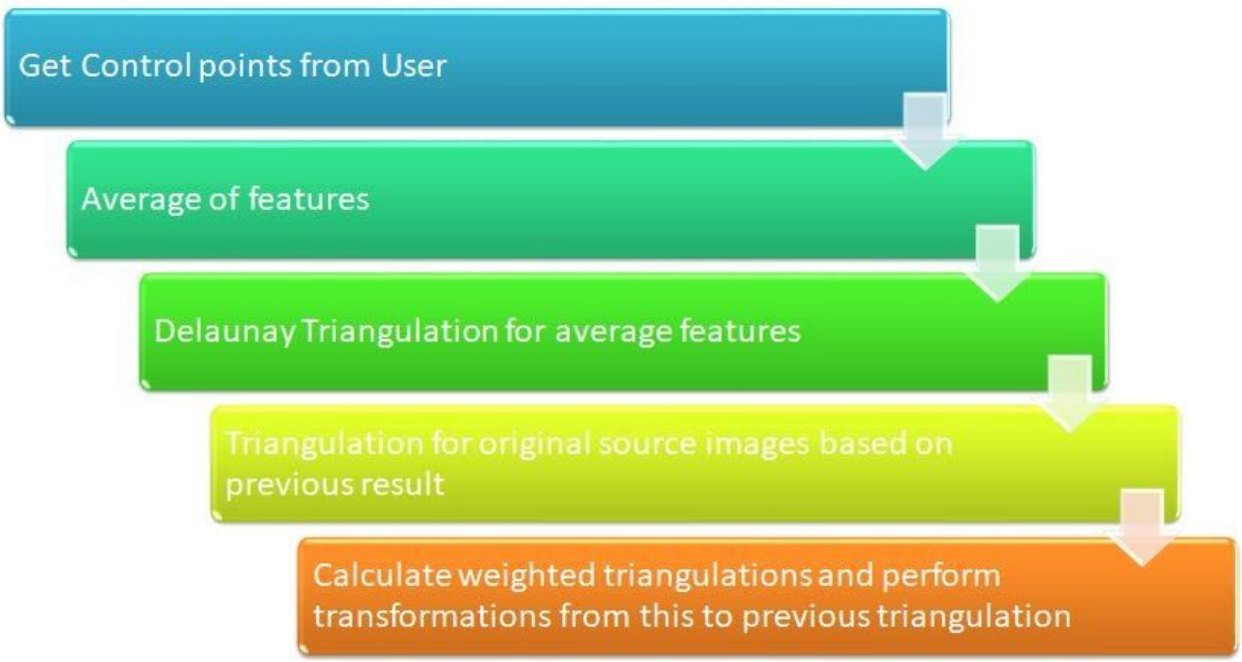
Methodology

- Number of control points : 4(corners) + 3(By using Mouse)
- Frames = 21



Frame 1 Frame 5 Frame 10 Frame 15 Frame 20

- Step 1: Get all the control points by clicking mouse on both images
- Step 2: Find the average of these features.
- Step 3: Calculate the Delaunay triangulation for these average feature sets.
- Step 4: Obtain the corresponding triangulations for source and target image. We will call them SRC_T, TARGET_
- Step 5: Start from alpha = 0 to 1
    1. Calculate the weighted facial features between source and target
    2. Find the triangulation of these features corresponding to the delaunay triangulation. We will call it WEIGHTED_T
    3. Find and perform the affine transformations for converting each triangle in SRC_T and TARGET_T into WEIGHTED_T
    4. Store the new frame in the video file



Conclusions

- I observe one thing that instead of taking manual points we can take control points using cv2 haar cascade Face Frontal module in will automatically assign control points around face of our image

- I failed at initial step, using pymouse for taking inputs from user, Its return coordinates according to full laptop screen and then i replaced it by c2.Event_LButtondown

- This property seems quite interesting to differentiate the Delaunay triangle.
  Empty circumcircle criterion ： For a set of points in 2-D, a Delaunay triangulation of these points ensures the circumcircle associated with each triangle contains no other point in its interior.

- Delaunay Triangulation use divide and conquer approach Complexity O(NlogN)
  Reference : University of Minnesota