CSL467_Siddharth_2016csb1043

Siddharth

February 19, 2018

1 Compilation and Input field :

Compiling : g++ -ggdb a2_2016csb1043_Siddharth.cpp -o facedetect
'pkg-config –cflags –libs opencv'
Input Fields : I have implemented single interface for both Affine and Face morphing Images.
Input image : Input image file location
Output image : Output image file location
Input feature file : Feature file for input
Output feature file : Feature file for output
No. of transition images : No. of input transtion images.

2 Concept Utilised :

I have used Delaunay Trangulation for warping of images . Created self Bilinear and Backward mapping algorithms. Time Complexity = O(n^3*t)

2.1 Face Morphing

2.1.1 Getting Triangle List :

Used inbuilt functionality for Trangulation. Used :
Rect rect(0, 0, size.height, size.width);
SubDiv2D for divison of figures
getTraingleList() for Delaunay traingle

2.1.2 Concept Learnt/Applied :

Delaunay Trangulation : It is Algorithm to divide image according to feature points in traingles and using backward mapping from transition to src and dst.Cross Dissolve both images to get Morphed image.
Baeir-Neely : It is algorithms that use line features and use backward mapping by minimum linear perpendicular distance between lines close to point.

2.1.3 Test Cases

(a) Donald (b) Hilary

Figure 1: The Input images for Morphing

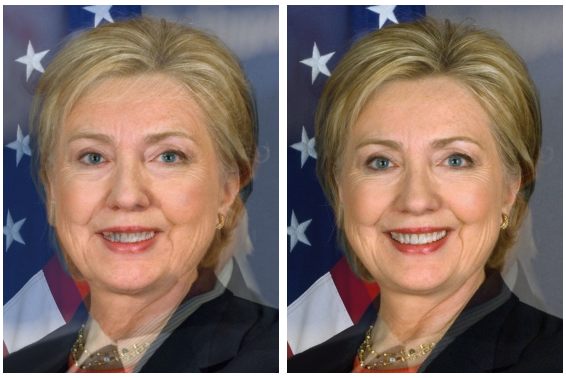
2.1.4 Morphed Cases

(a) Morph(0.2) (b) Morph(0.4) (c) Morph(0.5)

This are sample Morphing Images

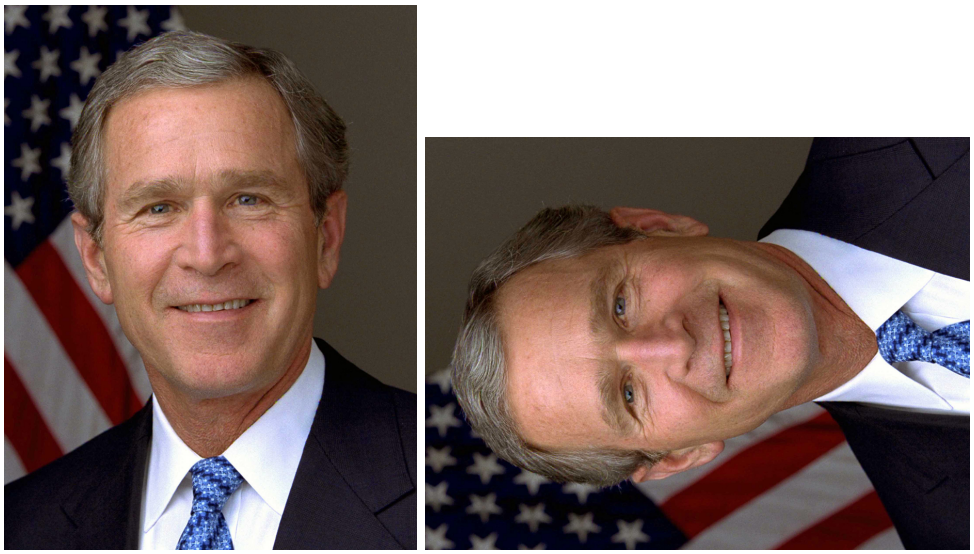
2.1.5 Observation :

- Smooth Transition of images from source to Dst.



(a) Morph(0.7) (b) Morph(0.9)

- There is complete Warping of feature pixel.
- Time Complexity : O(n^3)



(a) Bush (b) Bush Rot

Figure 4: The Input images for Morphing

2.2 Affine Transform

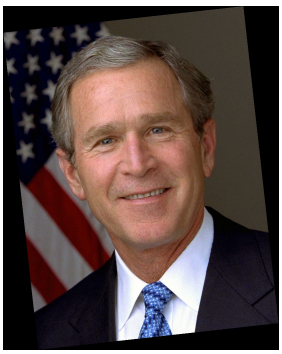
2.2.1 Concepts Used :

Delaunay Trangulation used using four corner as tie points.

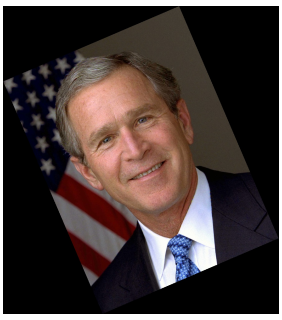
2.2.2 Test Cases :

7 Input Image

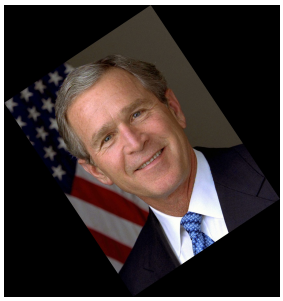
2.2.3 Morphed Cases



(a) Morph(0.2)



(b) Morph(0.4)



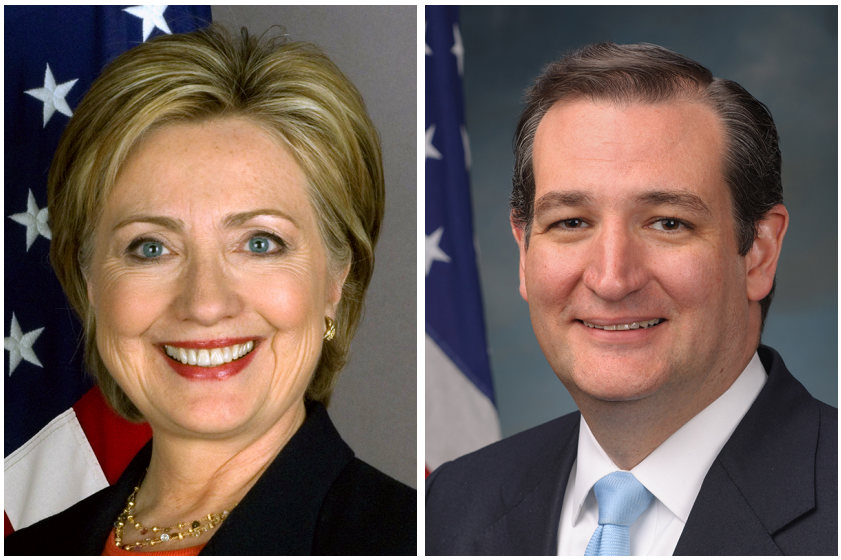
(c) Morph(0.5)

(a) Morph(0.7)

(b) Morph(0.9)

2.2.4 Observation :

- Smooth Transition of images from source to Dst.
- There is complete Warping of feature pixel. and produces image with intermeddiate angle.
- Time Complexity : O(n^3)



(a) Donald (b) Hilary

Figure 7: The Input images for Morphing

(a) Input1 (b) Morph(0.5) (c) Output2

3 More Test Cases :

3.0.1 Morphed Cases

4 Final Observations :

- I have implemented trangulation for each transition,So if number of transition increases it decrease wrapping og figures.
- For Affine Transformation,my Algorithm needs mapping for corner points

in destination image ,it solve the issue.*I have not considered to take 3*3 matrix input.

- For different sizes of image output will size will also vary linearly.
- It has better quality in comparison for Beir-Niller algorithm as it warps each coordinate point to correspondin feature and trangulation is proper.
- Its time complexity is more O(n^2) + O(n^3) for each transition.