

## Image Processing

INDIAN INSTITUTE OF TECHNOLOGY

### HOW TO RUN

- 2) To complie type:g++ a2\_pt1.cpp ‘pkg-config –cflags opencv –libs opencv’
- 3) ../a.out
- 4) To complie type: g++ a2\_pt2.cpp ‘pkg-config –cflags opencv –libs opencv’
- 5) ../a.out
- 6) Then choose the various options that are clearly specified in command prompts.

### OBSERVATIONS

I observed that the image morphing quality totally depends on the tie points accuracy and the images selected for morphing .

I also observed that the quality of image didn’t vary much if i changed the stage of triangulation wether i did it firstly or for each intermediate image.

For the image wrapping i used the delaunauy triangulation in opencv and for getting the inverse mapping i used affine transformation not perspective.

For delaunauy triangulation i used opencv following opencv functions:-  
**subdiv.getTriangleList()** for getting the triangulation.

**warpAffine(*src<sub>i</sub>image, dst<sub>i</sub>image, trans<sub>mat</sub>, dst<sub>i</sub>image.size()*)** - for getting the resultant image after transformation.

### IMPLEMENTATION DETAILS

I have implemented the 1st part of the assignment by finding the intermediate points by interpolation between source and destination points and getting the intensities from source image by finding the inverse mapping.

I have used the affine transformation to resize the intermediate image dynamically using the optimization of linear functions.

I have implemented the 2nd part of the assignment by finding the intermediate points by interpolation between tie points of imageA and imageB thereby performing wrapping and then getting the intensities from imageA and imageB by finding the inverse mapping to both A and B thereby cross-dissolving.

I have used the interpolation between imageA size and imageB size to resize the intermediate image dynamically.

## **ASSUMPTIONS**

The affine transformation for transformation have to be given according to fact noting the correct axes. For tie points its assumed that the tie points are given correctly.

For creating GIF i have used the command line tool.

For the tie points i have used the files already available on the internet and have mentioned it in references.

## **REFERENCE**

The assignment is done individually by myself, with help of class notes, slides, shared content, opencv documentation and LearnOpencv tiepoints and images are used.

*<https://www.learnopencv.com/face-morph-using-opencv-cpp-python/>.*