Head Pose Changer

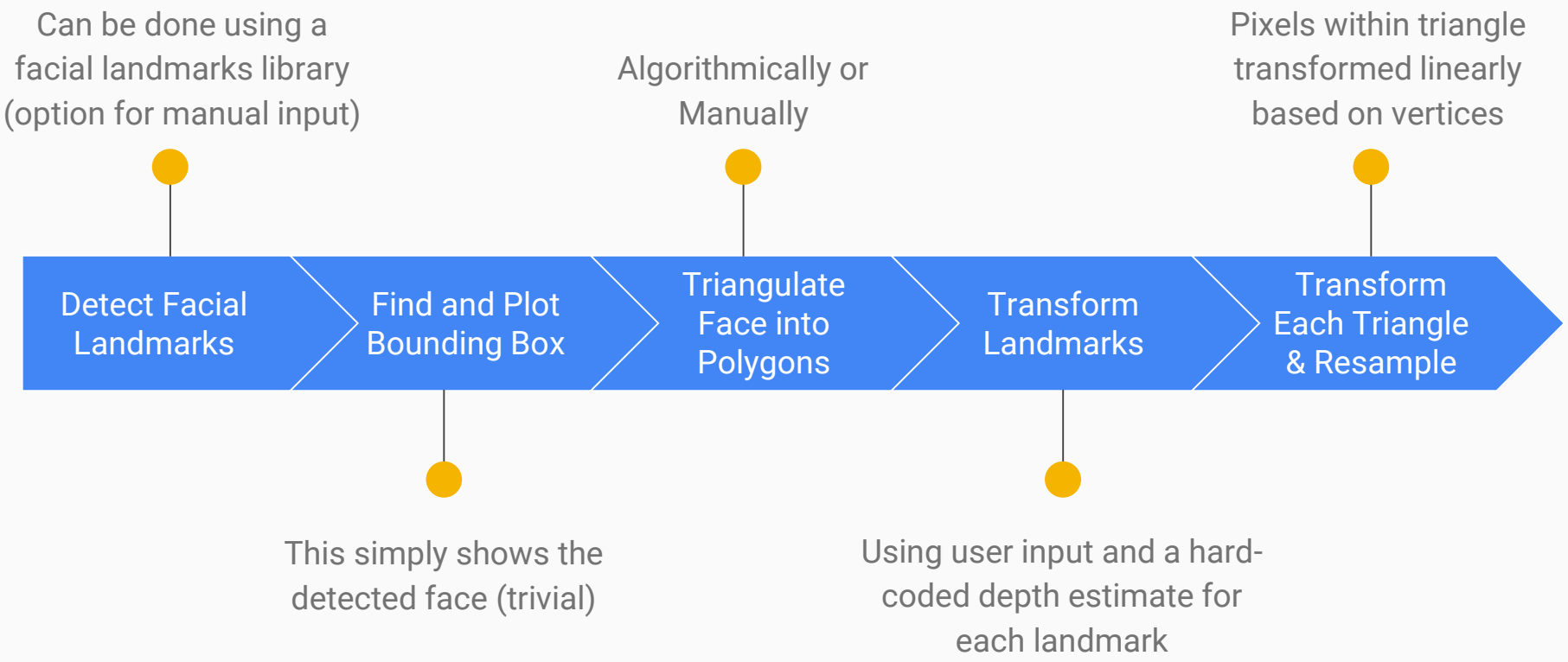
Shah Aalap Anil (150655)
Course Project for EE604: Image Processing
Instructor: Prof. K Venkatesh
10th November, 2019

Objective

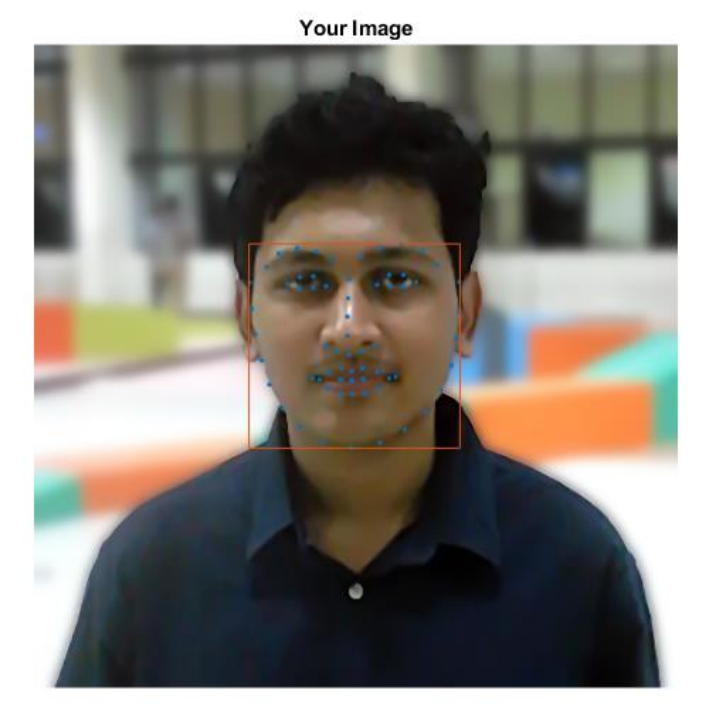
Develop an application using **MATLAB GUI** which will receive the image of an individual as input, and then:

- Detect the face/head
- Turn the face/head slightly in all four directions according to user input.

Procedure


Can be done using a facial landmarks library (option for manual input)
Algorithmically or Manually
Pixels within triangle transformed linearly based on vertices
Detect Facial Landmarks
Find and Plot Bounding Box
Triangulate Face into Polygons
Transform Landmarks
Transform Each Triangle & Resample
This simply shows the detected face (trivial)
Using user input and a hard-coded depth estimate for each landmark

Sub-Task 1

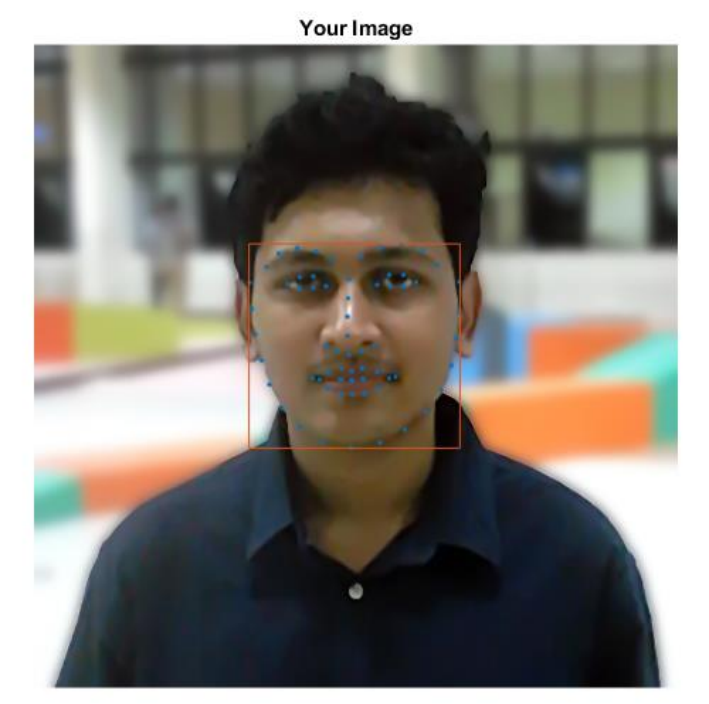


Library used:
https://github.com/YuvalNirkin/find_face_landmarks

Facial Landmark Detection:

- MATLAB interface for OpenCV library called dlib
- Detects landmarks by classifying HOG features using a pre-trained SVM

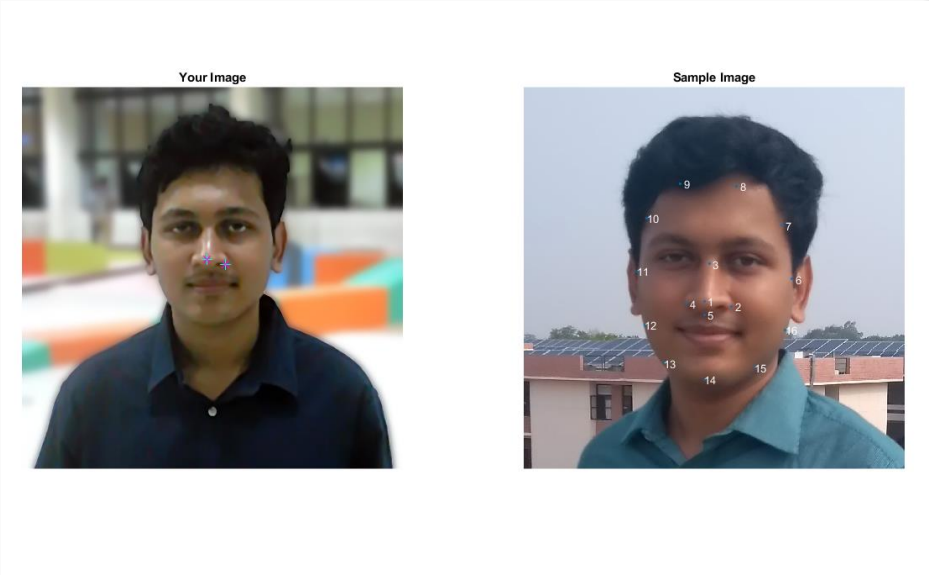
Sub-Task 1



Facial Landmark Detection:

- The MATLAB GUI plots the landmarks by overlaying a scatter plot over the image
- Bounding box of the landmarks plotted to show the detected face

Sub-Task 1



Fail-Safe:

- Added option to manually select landmarks for an image where they are not detected
- Reference image with landmark positions shown

Sub-Task 2

Triangulation:

- Before transforming the image, it is divided into polygons
- Can be done algorithmically, but since the number of landmarks are fixed, the triangulation is defined manually (as shown).

Sub-Task 3

Vertex Transformation:

- Each point moved by an amount based on its expected depth (for example, nose tip moved more, eye centre moved less)
- Direction and mean magnitude based on user input
- For each triangle, a transformed triangle is obtained

Pixel Transformation:

- For each pixel (x, y) that lies within a triangle, it is transformed linearly along with the vertices of the triangle
- Pixel location expressed as linear combination of side vectors and then transformed to the same linear combination of transformed side vectors
- New pixel locations are **not integers**

Sub-Task 4

Resampling:

- Since each pixel is transformed to a non-integer location, the new image may have gaps/overlaps if they are simply rounded off
- Each channel is taken as a function over the non-integer points
- The function is sampled using 2D linear interpolation at integer points to get the image
- HSV channels used instead of RGB for better output

Results

Tilt Up

Tilt Left

Tilt Right

Tilt Down