3D Face Reconstruction using iPhone's True Depth Camera

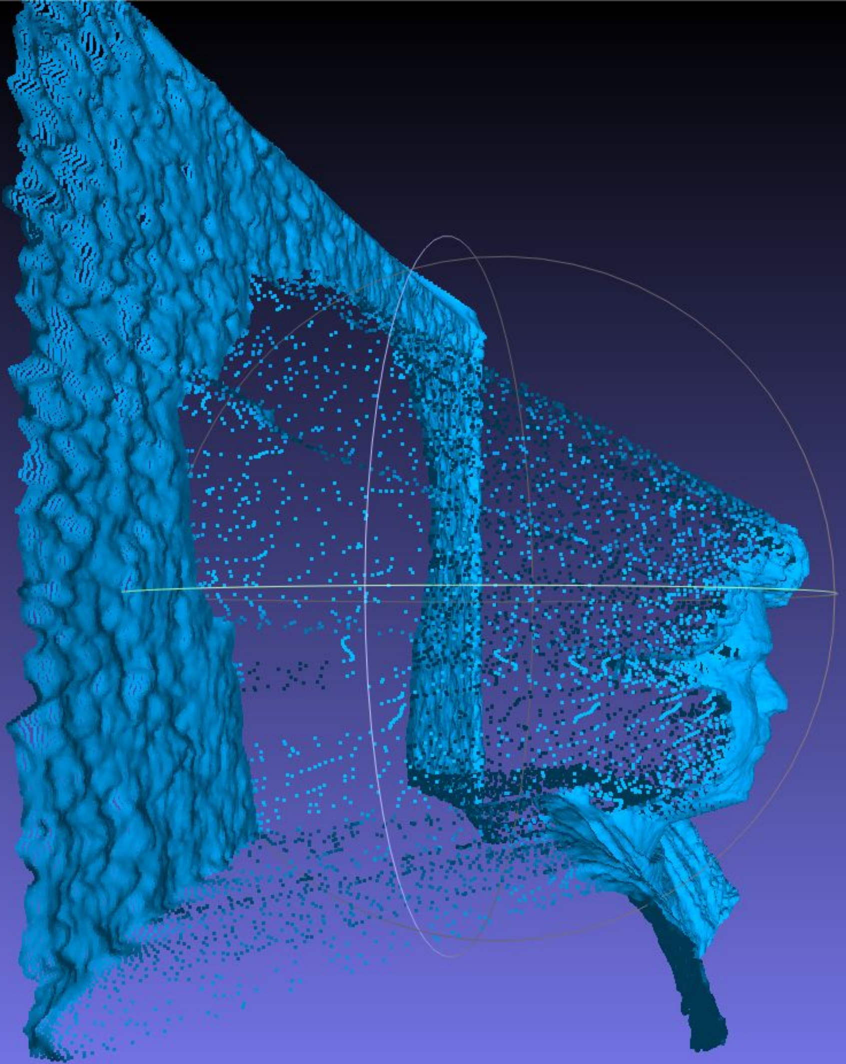
Kesav Ravichandran, Utkarsh Gupta, Yunqian Cheng

Outline

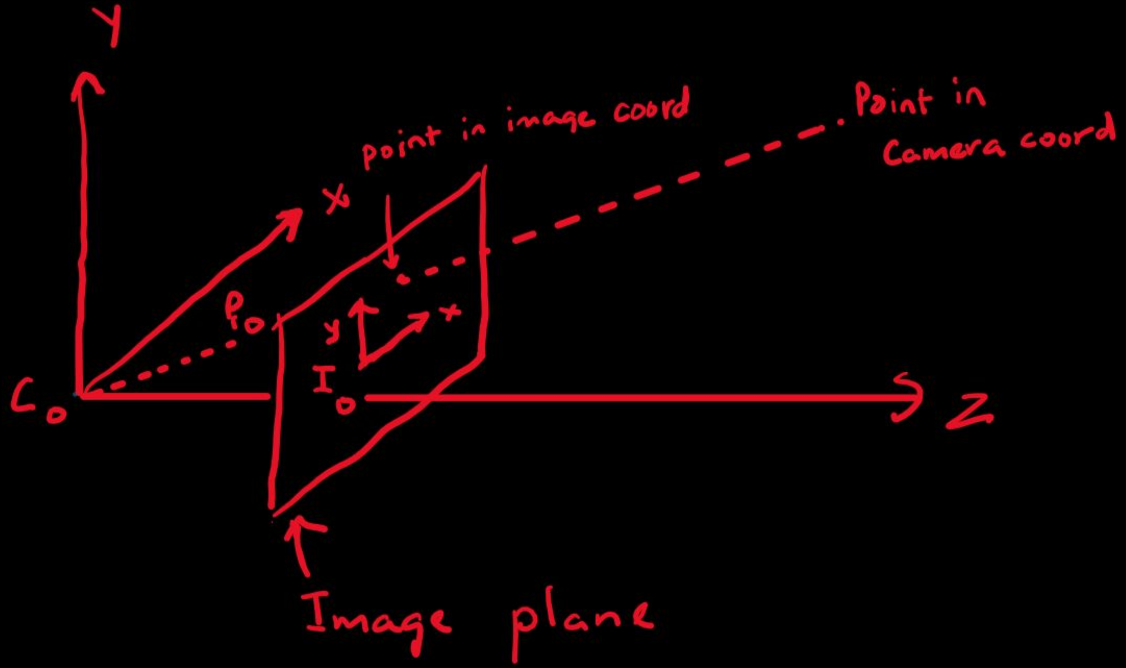
- Develop an iOS application that captures a 3D point cloud of the user's face using the iPhone’s front-facing True Depth camera.
- Fit the FLAME model to the data to create a ‘complete’ face model with only one view.
- Compare results with baseline models that uses RGB images from NoW challenge (MICA and DECA) mainly in low lighting/pitch-dark conditions.

Working of TrueDepth Sensor


Ambient light sensor
Speaker
Proximity sensor
Microphone
Flood illuminator
Front camera
Infrared camera
Dot projector



COORDINATE SYSTEMS



C_0 - Camera Origin.

I_0 - Image / Camera Sensor Origin.

P_0 - Pixel Origin (0,0) Top left.

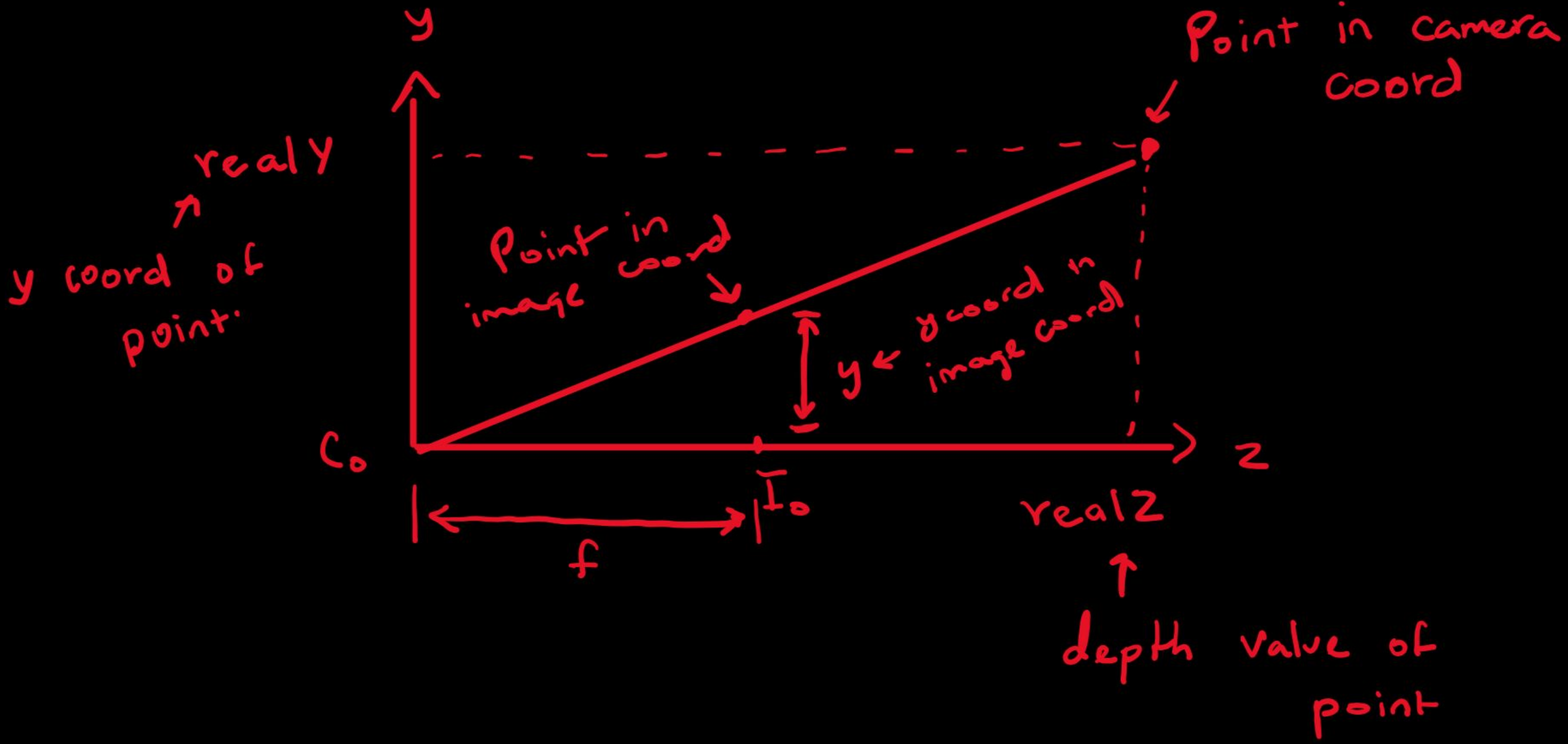
CAMERA CALIBRATION (AVFoundation API)

$$\begin{bmatrix} f_x & 0 & O_x \\ 0 & f_y & O_y \\ 0 & 0 & 1 \end{bmatrix}$$

O_x, O_y → offset of center in image coordinate (I_0) from center in pixel coordinate

f_x, f_y → Pixel focal length of camera.

COORDINATE SYSTEM CONVERSION


y
Point in camera coord
realy
y coord of point.
Point in image coord
y
y ← ycoord in image coord
C_0
I_0
z
realZ
f
depth value of point

Pixel Coordinates (x_p, y_p) -> Image Coordinates (x_i, y_i)

offset of I_o from pixel (0,0)

$$y_i = y_p \times dy - O_x$$

dimention of image for scale

offset of I_o from pixel (0,0)

$$x_i = x_p \times d_x - O_y$$

dimention of image for scale

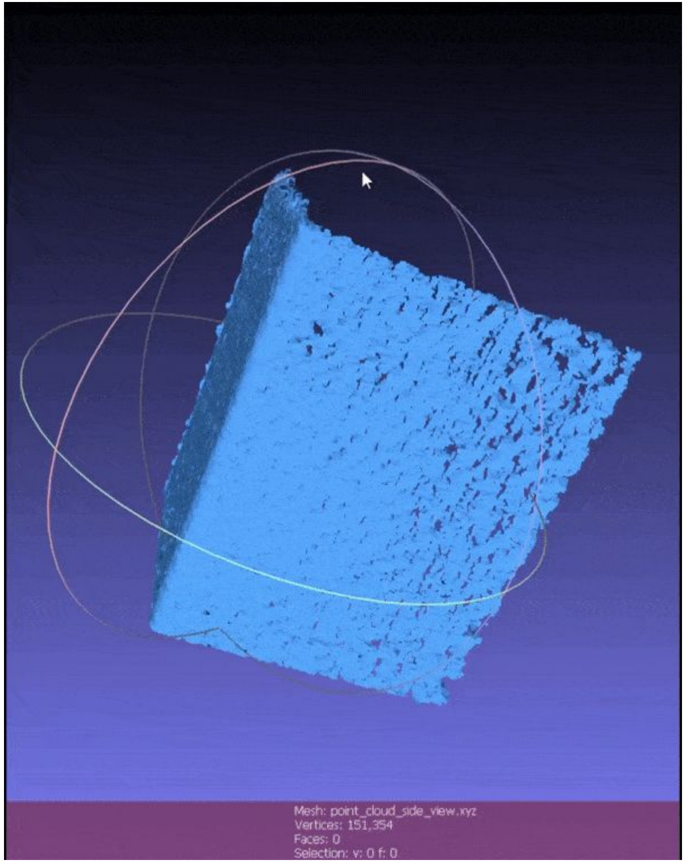
Image Coordinates (x_i, y_i) -> Camera Coordinates (realX, realY, realZ)

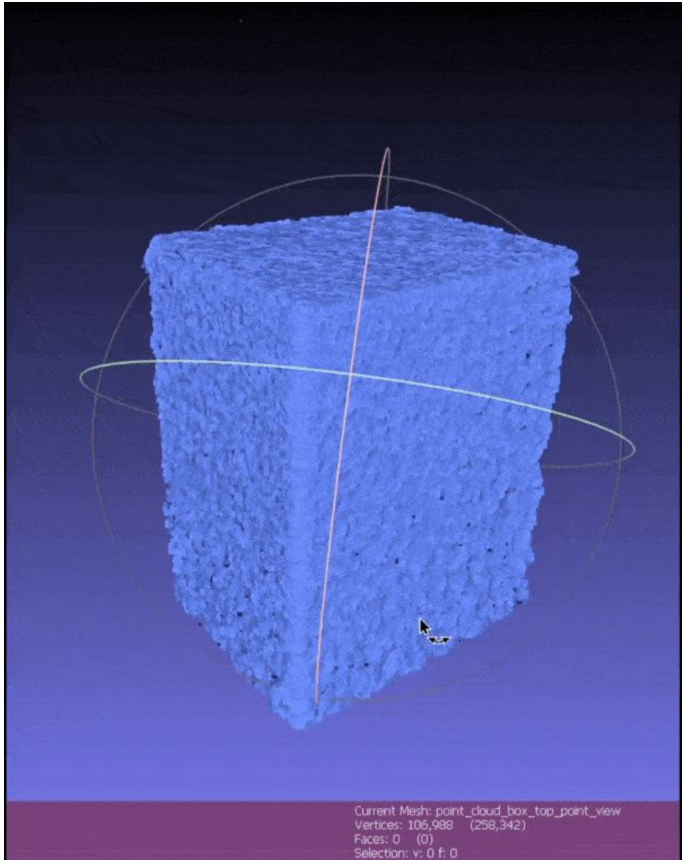
$$\frac{y_i}{f} = \frac{realY}{realZ} \quad \therefore \quad realY = \frac{y_i}{f} \times realZ$$

Similarly,

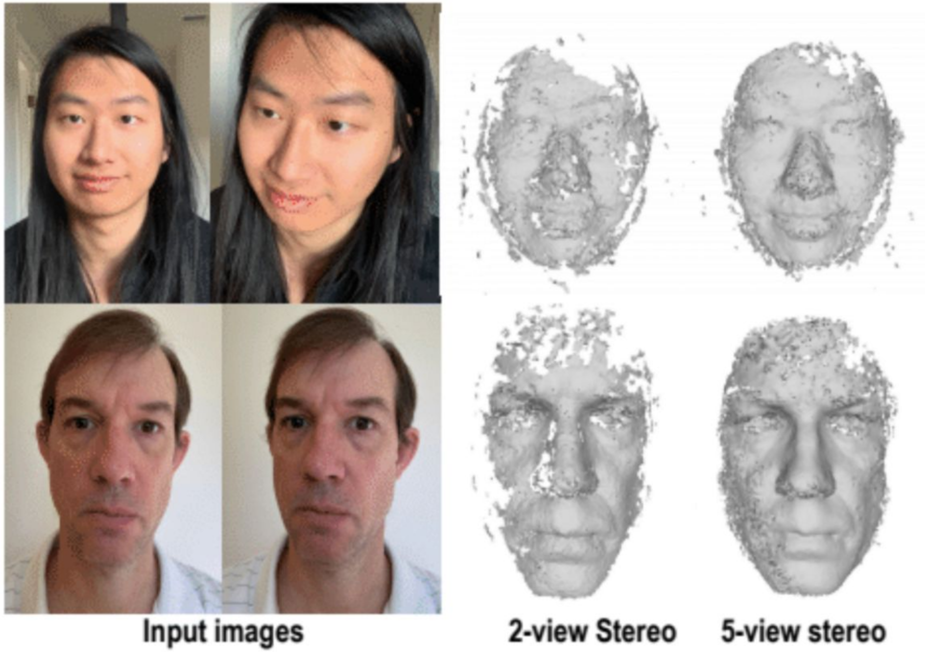
$$\frac{x_i}{f} = \frac{realX}{realZ} \quad \therefore \quad realX = \frac{x_i}{f} \times realZ$$

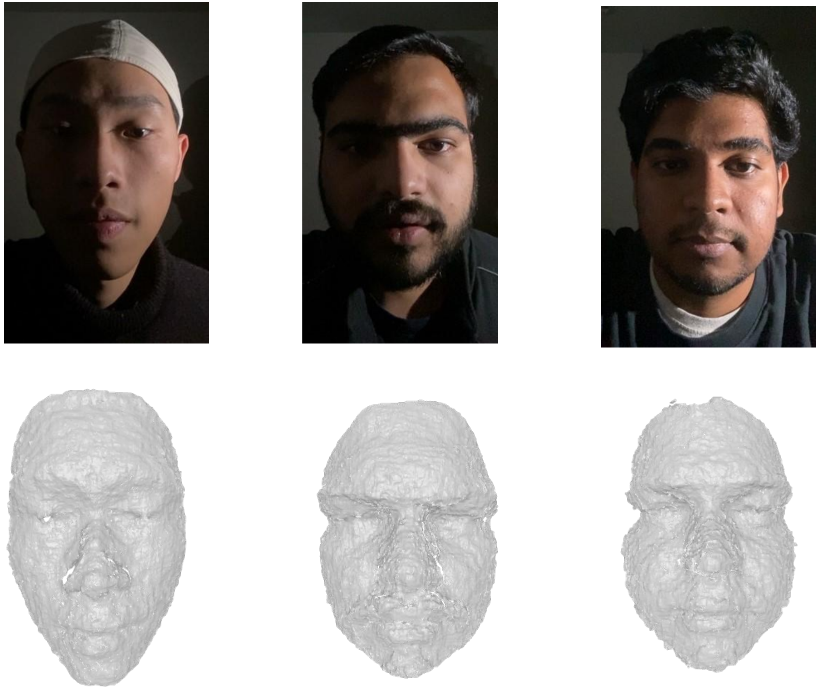
RAW DEPTH RESULTS


Mesh: point_cloud_side_view.xyz
Vertices: 151,354
Faces: 0
Selection: v: 0 f: 0


Current Mesh: point_cloud_box_top_point_view
Vertices: 106,988 (258,342)
Faces: 0 (0)
Selection: v: 0 f: 0

Comparison of results with Stereo

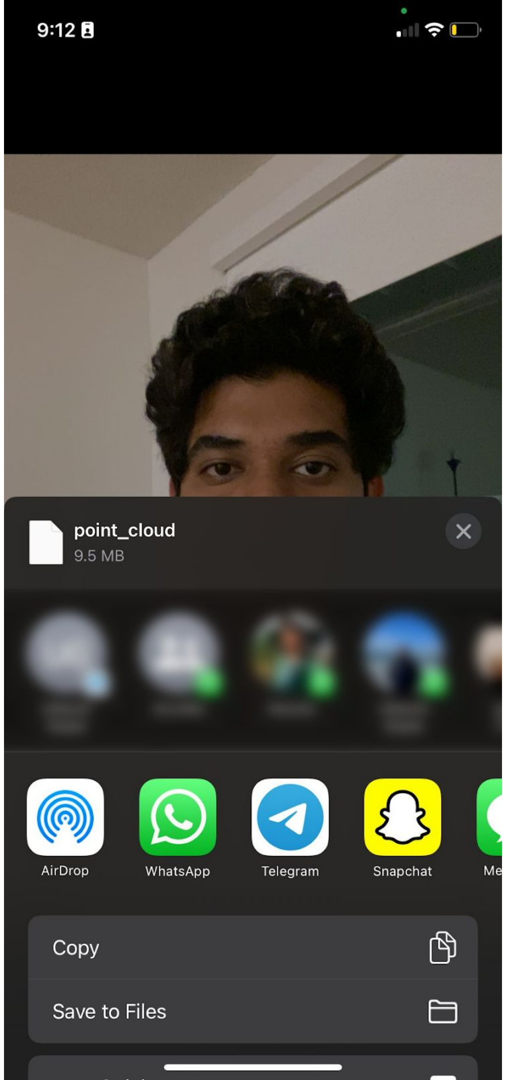


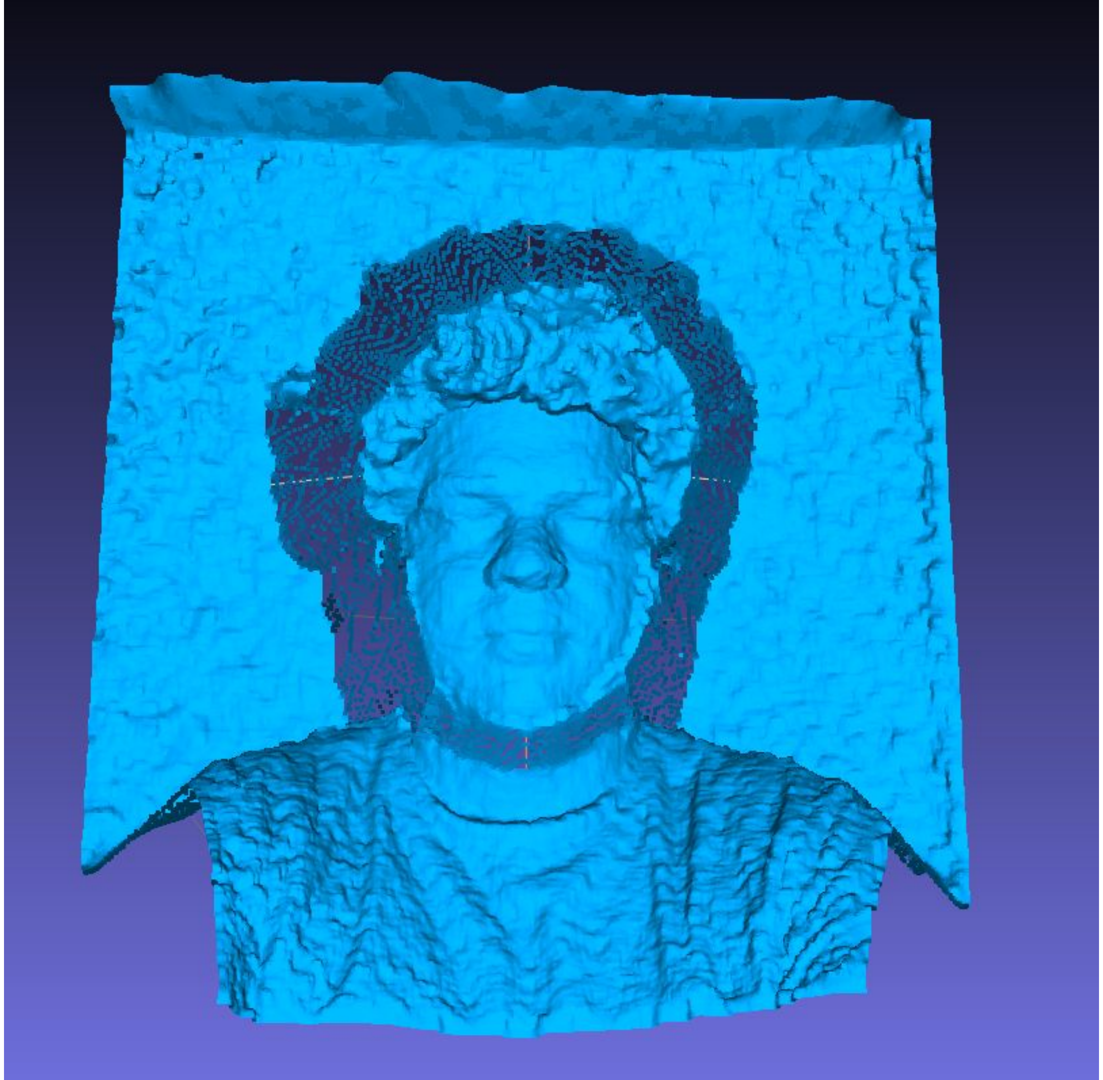


APP SCREENSHOTS

Preview and Capture Scene

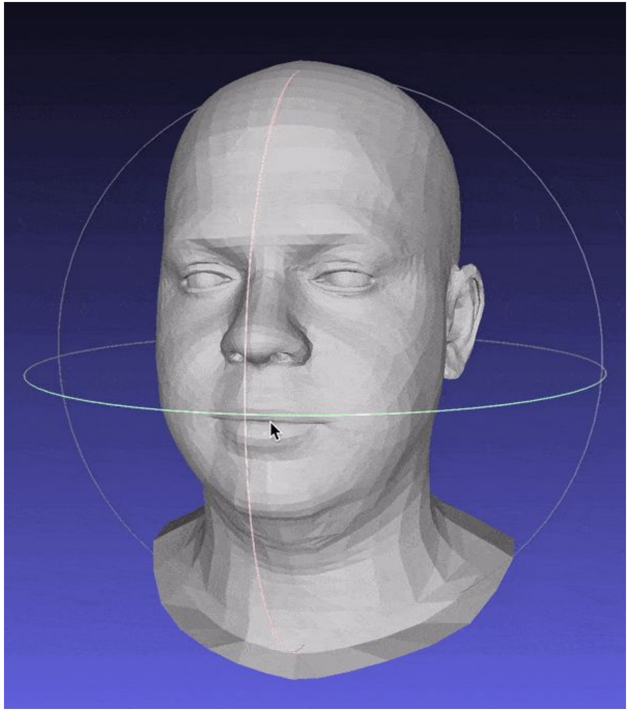
Share Menu for Point Cloud



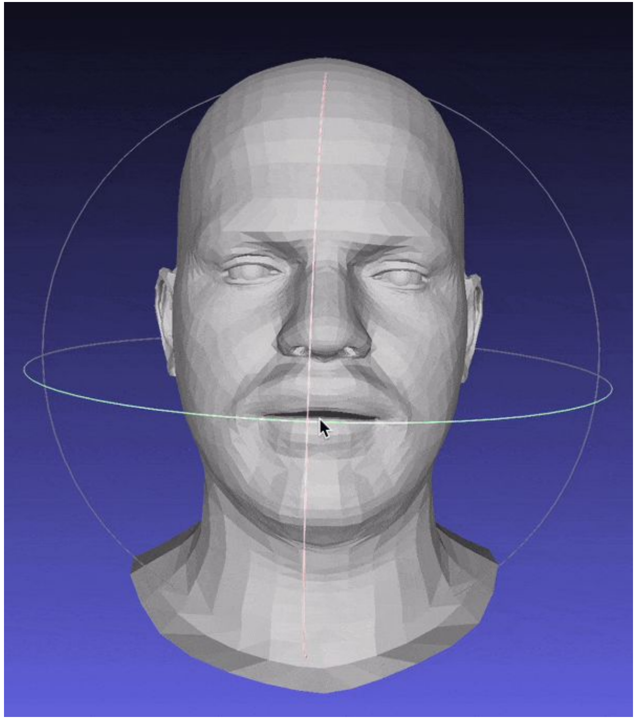


FLAME FITTING RESULTS

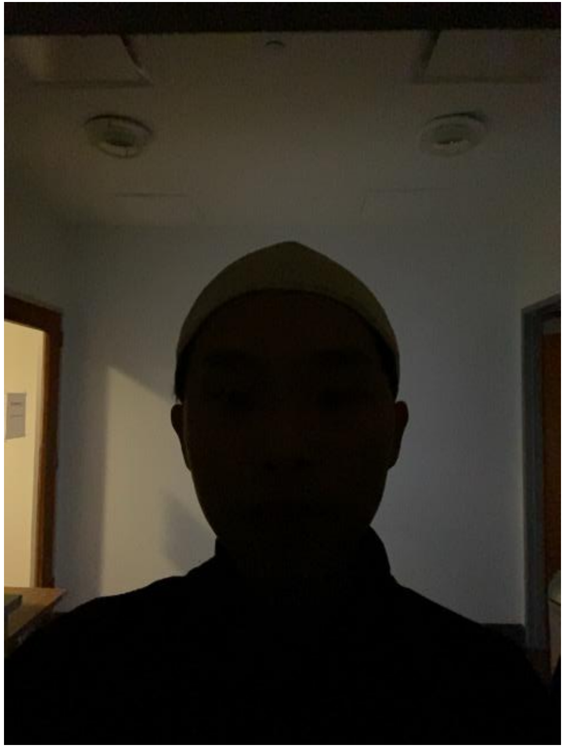
KESAV

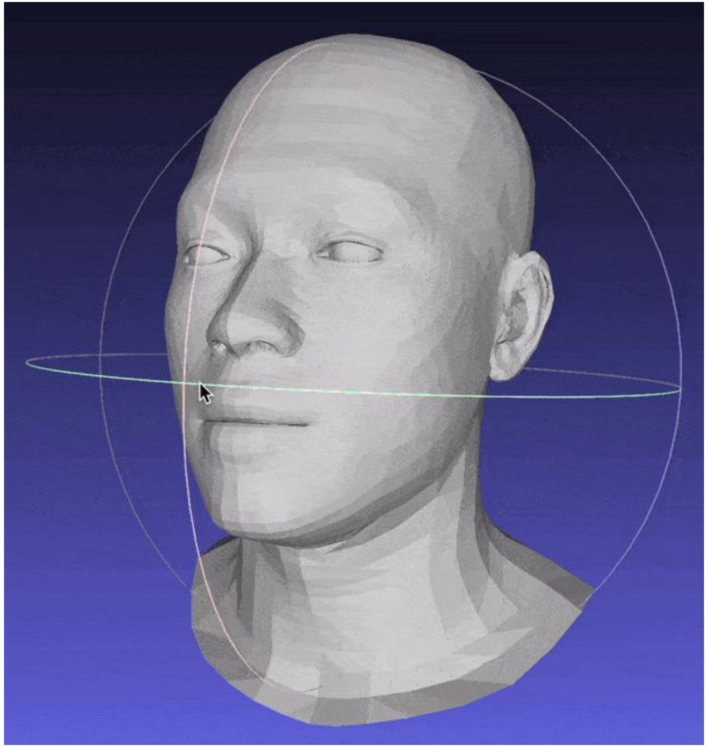


UTKARSH

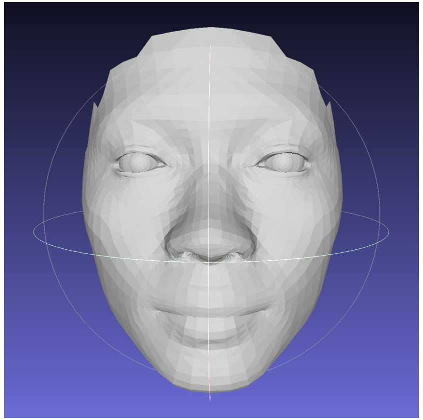


PATRICK

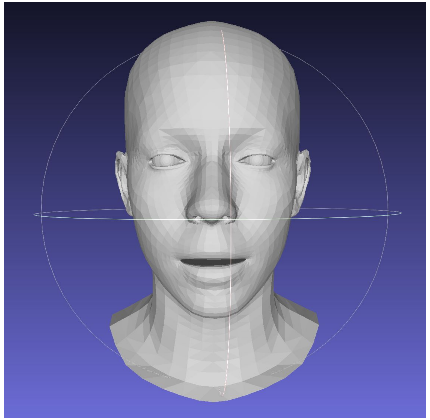




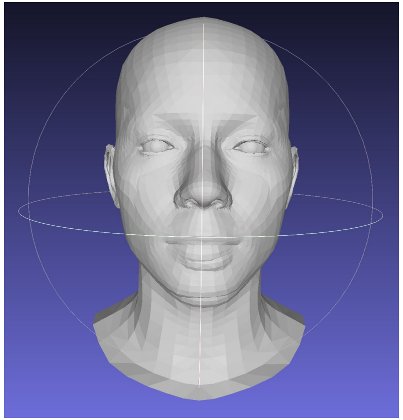
NoW BENCHMARK



Ground Truth



Predicted mesh



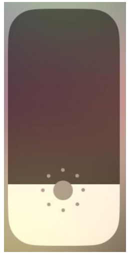
Rough alignment

1. Ground Truth is generated using MICA.
2. Predicted mesh shown is the DECA output

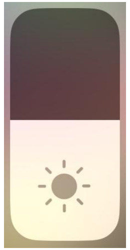
***NOTE:** We are only using MICA's ground truth to prove the consistency of our mobile application results.

LOW LIGHTING EXPERIMENT

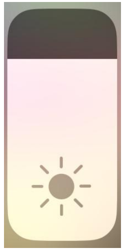
Low Lighting Experiment



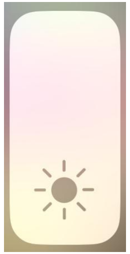
25%



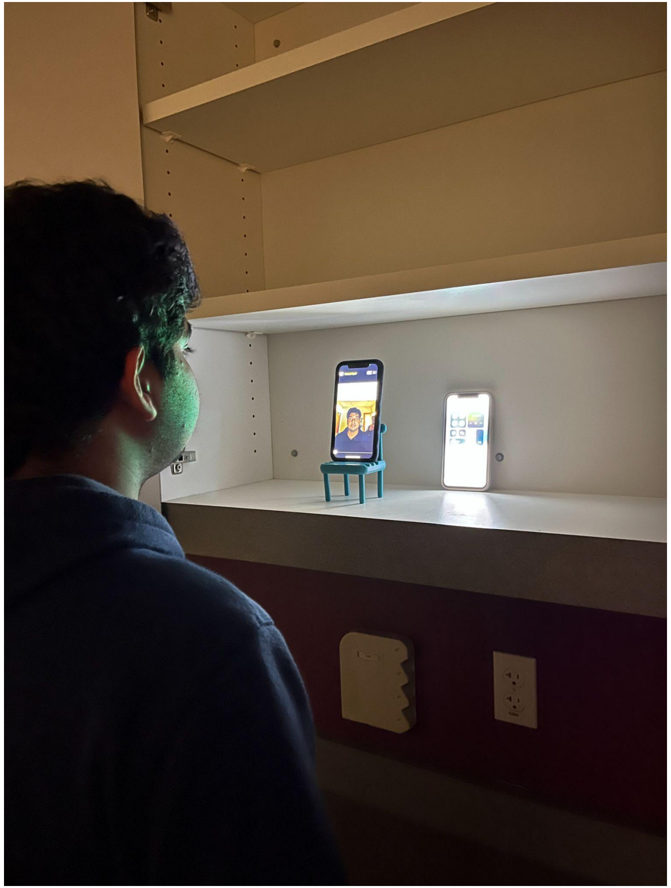
50%



75%



100%

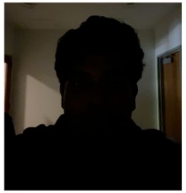


Low Lighting Experiment





(A: 25%) (B: 50%) (C: 75%) (D: 100%) (E: natural)

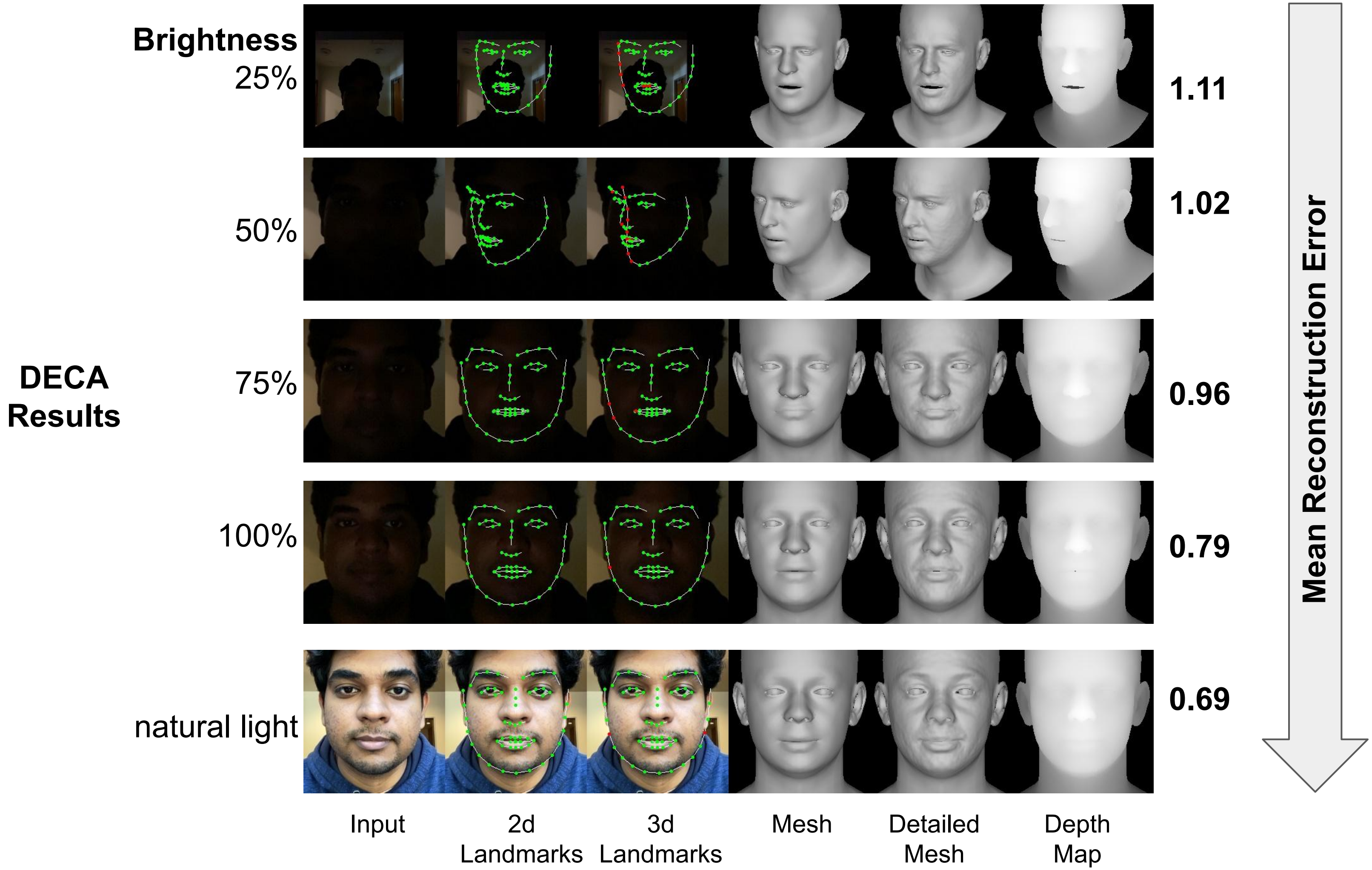

(F: 25%) (G: 50%) (H: 75%) (I: 100%) (J: natural)

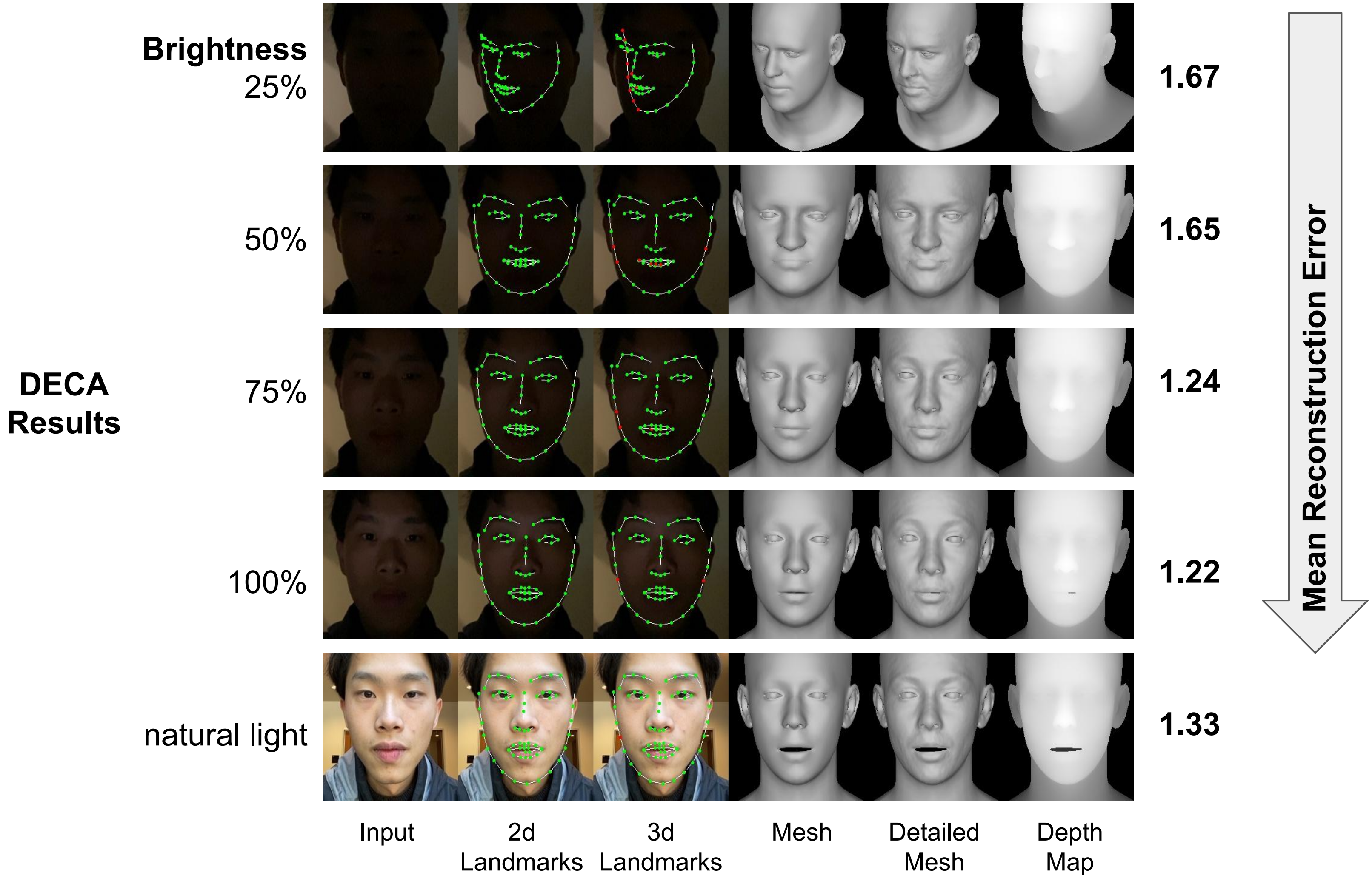




(K:25%) (L:50%) (M: 75%) (L:100%) (O: natural)

NoW RESULTS


Brightness
25%
1.11
50%
1.02
DECA
Results
75%
0.96
100%
0.79
natural light
0.69
Mean Reconstruction Error
Input
2d
Landmarks
3d
Landmarks
Mesh
Detailed
Mesh
Depth
Map



Brightness
25%
50%
75%
100%
natural light
DECA
Results
1.67
1.65
1.24
1.22
1.33
Mean Reconstruction Error
Input
2d
Landmarks
3d
Landmarks
Mesh
Detailed
Mesh
Depth
Map

Brightness Level	**Mean Reconstruction Error**			
	DECA		**iPhone’s depth data + FLAME**	
	Kesav	**Patrick**	**Kesav**	**Patrick**
25%	1.11	1.67	1.17	0.75
50%	1.02	1.65	1.13	0.72
75%	0.96	1.24	1.07	0.69
100%	0.79	1.22	1.06	0.78
natural	0.69	1.33	1.13	0.68

CONSISTENT **CONSISTENT**

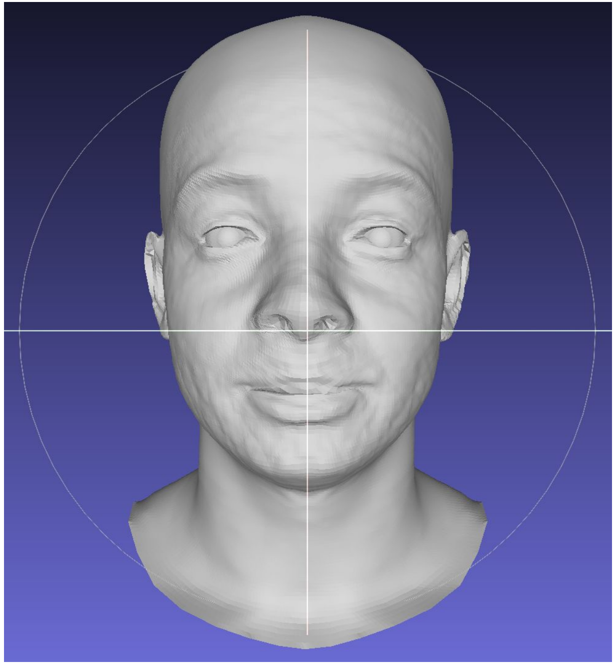
***__NOTE:__ Do not compare the error across models.
The point here is to prove the consistency of results

VISUAL COMPARISON

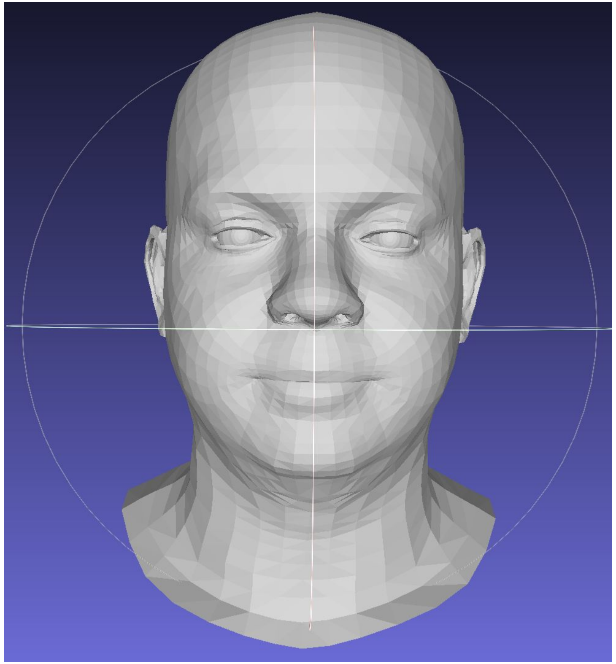
Image

natural

DECA



FLAME



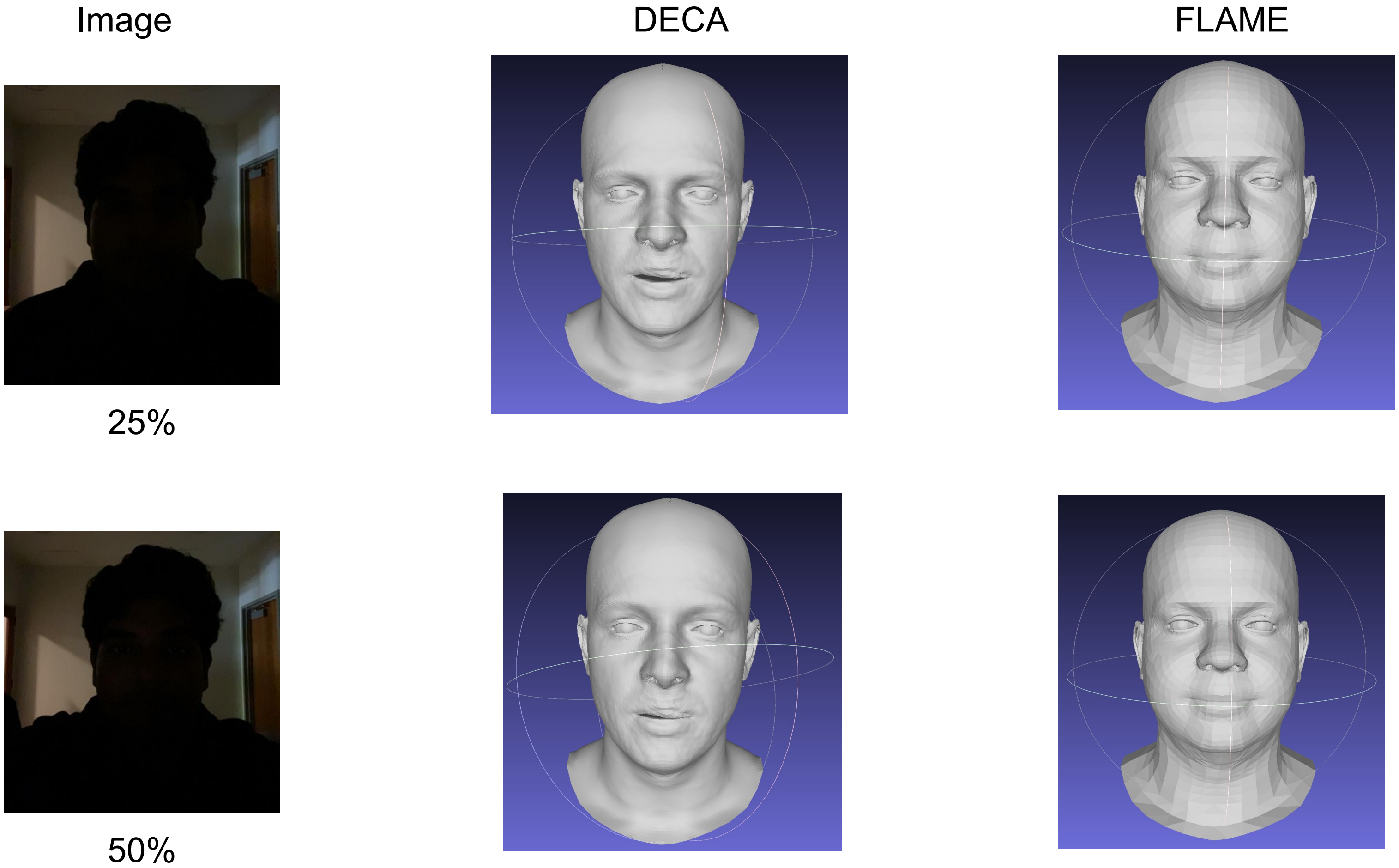
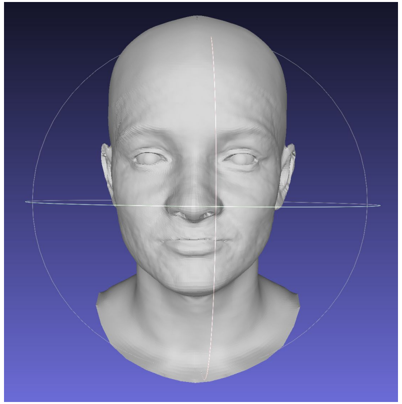
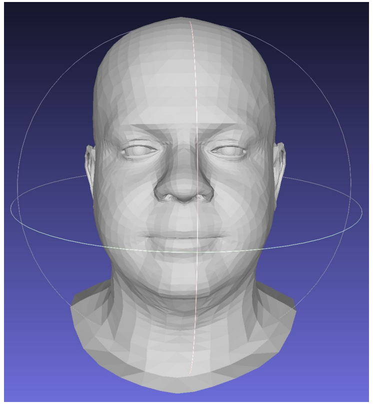


Image
DECA
FLAME
25%
50%

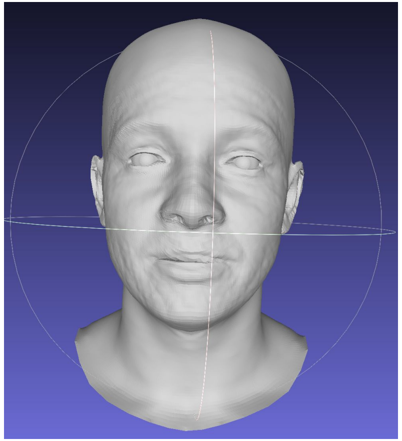
Image DECA FLAME

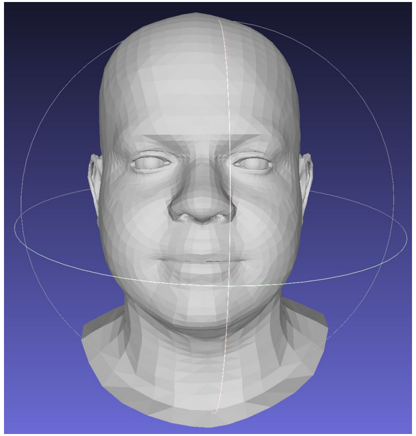
75%





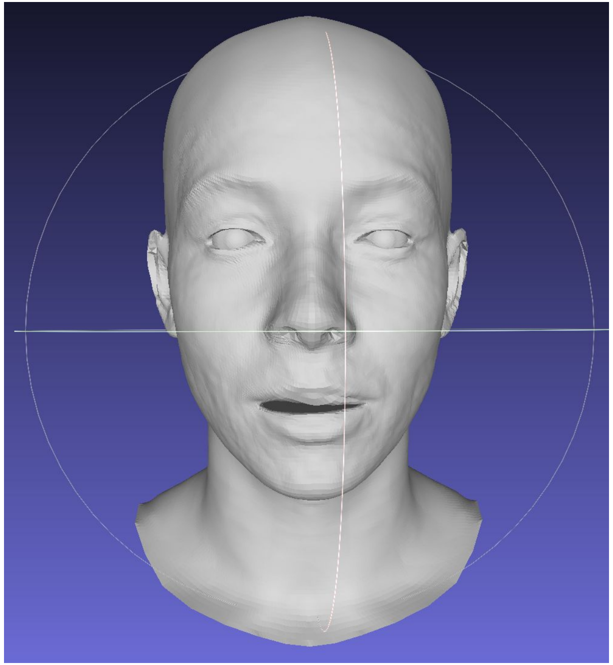
100%



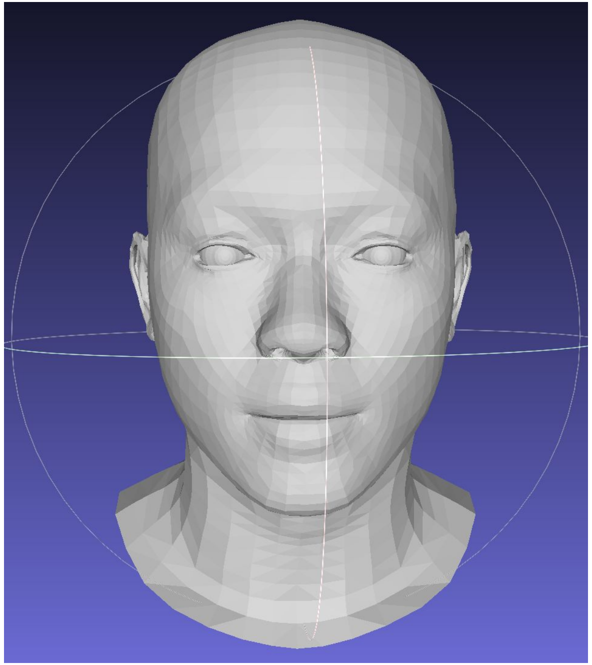


Image

DECA



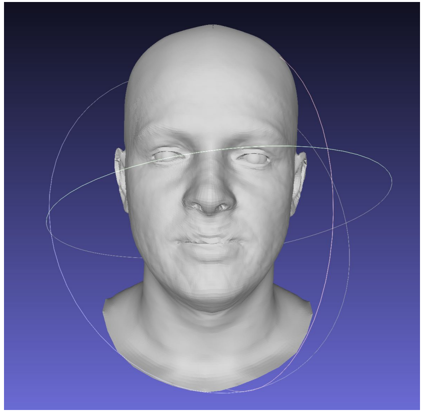
FLAME

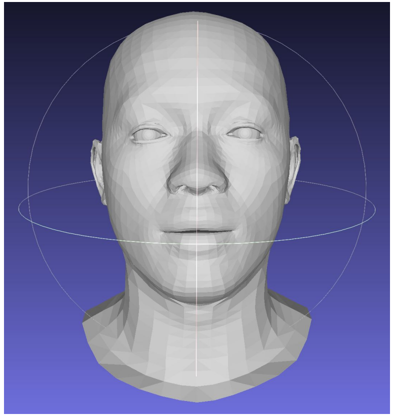


Image

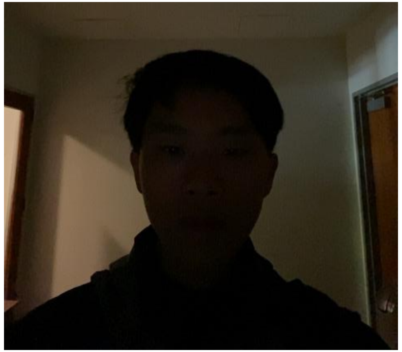
DECA

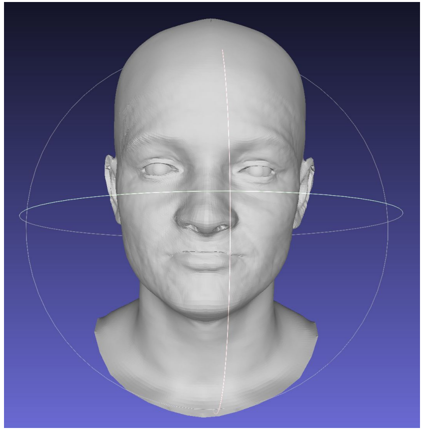
FLAME

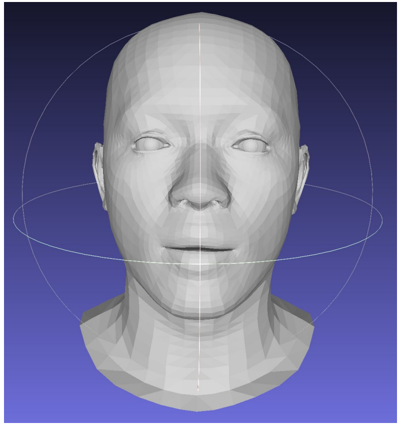




25%





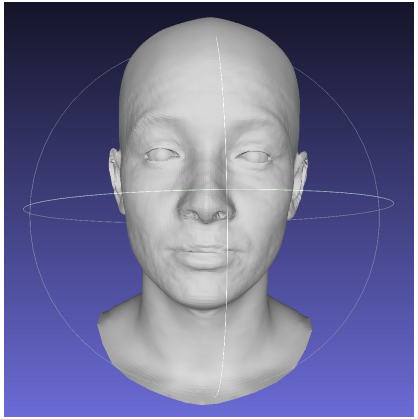


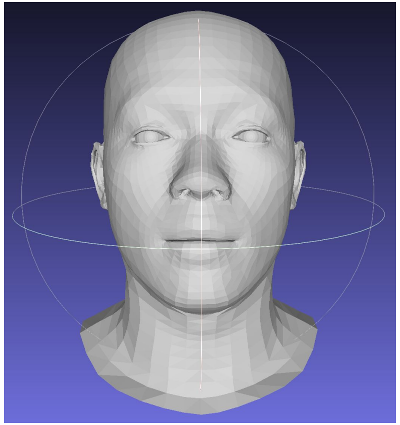
50%

Image

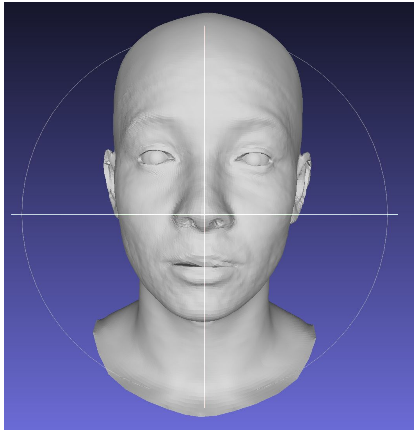
DECA

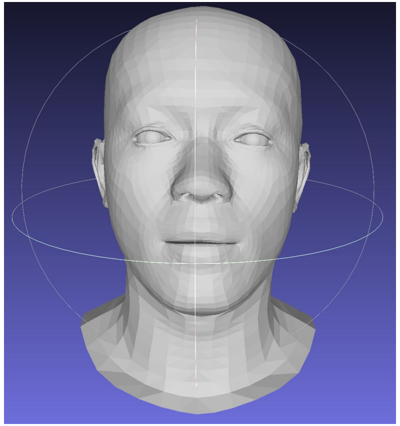
FLAME





75%

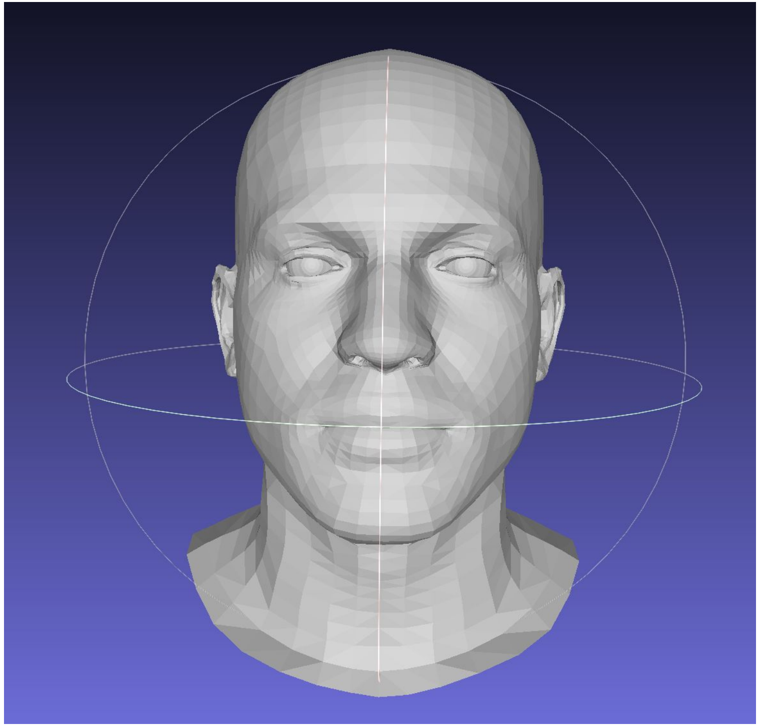




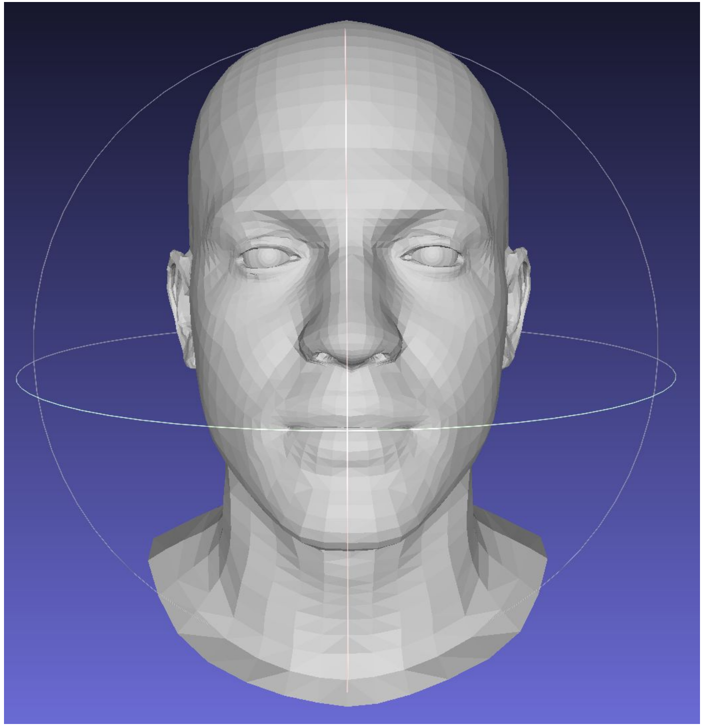
100%

CHALLENGES

25% brightness



Kesav



Utkarsh

Conclusion

- Made an iPhone mobile application for facial 3D point cloud capturing
- By comparing results with baseline models from NoW challenge (MICA and DECA) we found that:
  - Our method (iPhone capture point cloud + FLAME) produces visually better results than MICA and DECA
  - Image-based Machine Learning algorithms will either not work at all or will perform suboptimally as compared to the 3d face reconstruction done using iPhone’s depth data.

Discussion & Broader Impact

Our solution uses an iPhone as the data capturing device and only require one capture to generate a good 3D face model. Based on these characteristics, our method can provide a faster, more accessible, and consistent solution for 3D face model capturing up to a certain degree of accuracy. We hope that this work can help researchers capture more data and potentially lead to better solutions to 3D face model capturing.

Questions?