

# Morphing of facial 3D mesh

BAHA MABROUK

# Objective

---

- Extracting 3D pointcloud from 2D image
- Morphing two point cloud representation of facial features from one face to another

# Process



- Extract facial features out of a 2D image
- Represent it in a points distribution
- Recreating depth and transforming the 2D point mesh into a 3D point cloud
- Morphing two point cloud representation of facial features from face to another



# Feature Extraction from 2d image

## MEDIAPIPE from GOOGLE

[https://developers.google.com/mediapipe/solutions/vision/face\\_landmarker](https://developers.google.com/mediapipe/solutions/vision/face_landmarker)

The MediaPipe Face Landmarker task lets you detect face landmarks and facial expressions in images and videos. You can use this task to identify human facial expressions, apply facial filters and effects, and create virtual avatars.

```
mp_face_mesh2 = mp.solutions.face_mesh
face_mesh2 = mp_face_mesh2.FaceMesh()

image2 = cv2.imread('face3.jpg')
image2_rgb = cv2.cvtColor(image2, cv2.COLOR_BGR2RGB)

results2 = face_mesh2.process(image2_rgb)
blank_image2 = np.zeros(image2.shape, dtype=np.uint8)

if results2.multi_face_landmarks:
    for face_landmarks2 in results2.multi_face_landmarks:
        mp.solutions.drawing_utils.draw_landmarks(blank_image2, face_landmarks2, mp_face_mesh2.FACEMESH_TESSELATION)
        vertices2 = np.array([[lm.x, lm.y, lm.z] for lm in face_landmarks2.landmark])
        print(vertices2)

plt.imshow(cv2.cvtColor(blank_image2, cv2.COLOR_BGR2RGB))
plt.axis('off') # Turn off axis labels
plt.show()
```

# Extracting facial features out of 2d Image

ORIGINAL IMAGE



MESH OF FEATURES



# Extracting facial features out of 2d Image

ORIGINAL IMAGE 2



MESH OF FEATURES 2



# Point cloud extraction

To extract a point cloud from a 2D mesh using Open3D, we used the **create\_point\_cloud\_from\_depth\_image** function along with a depth image generated from your mesh.

```
mesh2 = o3d.io.read_triangle_mesh("face_mesh2.stl")

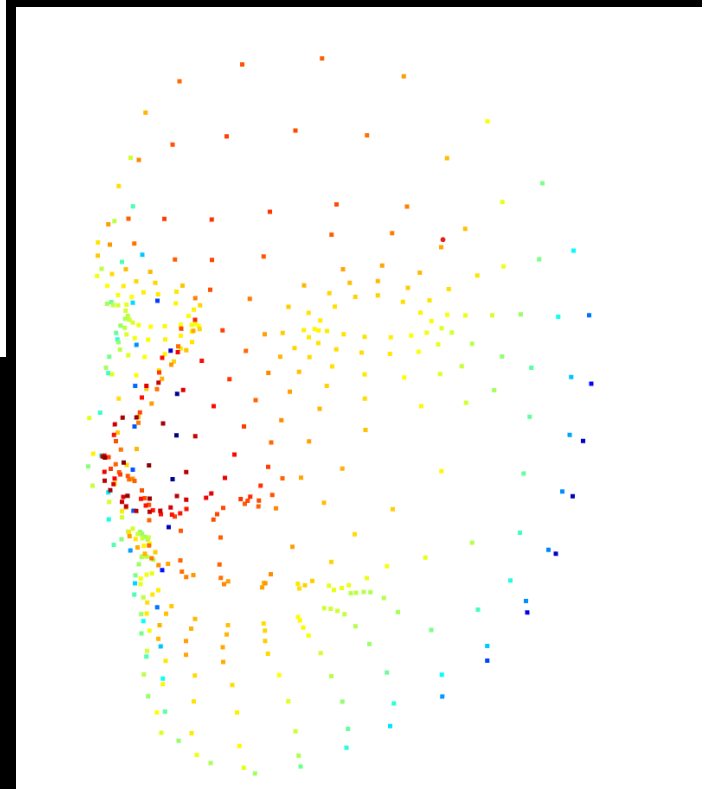
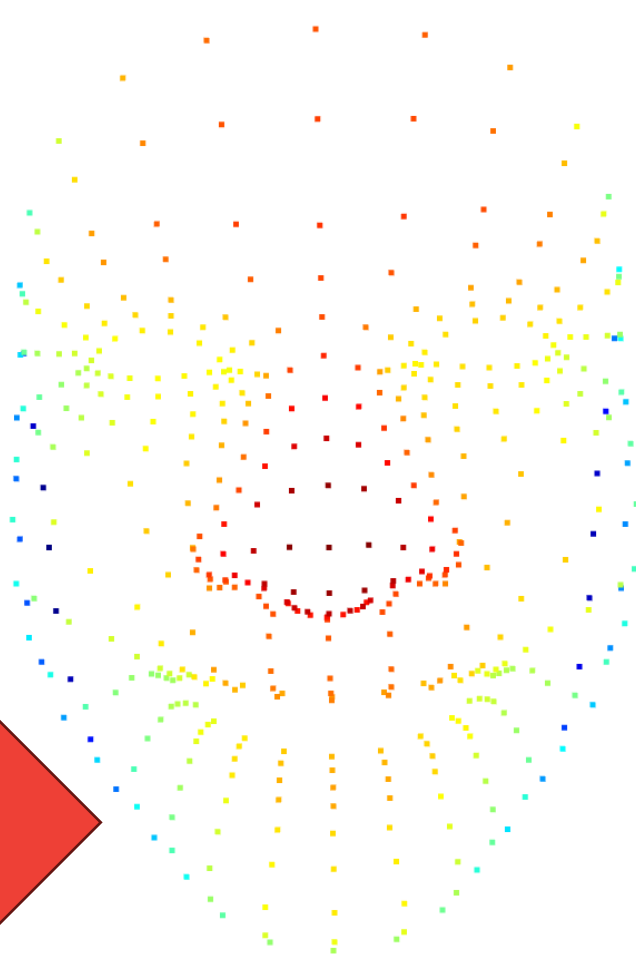
# Get vertices from the mesh

point_cloud2 = o3d.geometry.PointCloud()
point_cloud2.points = o3d.utility.Vector3dVector(vertices2)
point_cloud2.transform([[1, 0, 0, 0],
                        [0, -1, 0, 0],
                        [0, 0, -1, 0],
                        [0, 0, 0, 1]])

# Visualize the point cloud
o3d.visualization.draw_geometries([point_cloud2])
```

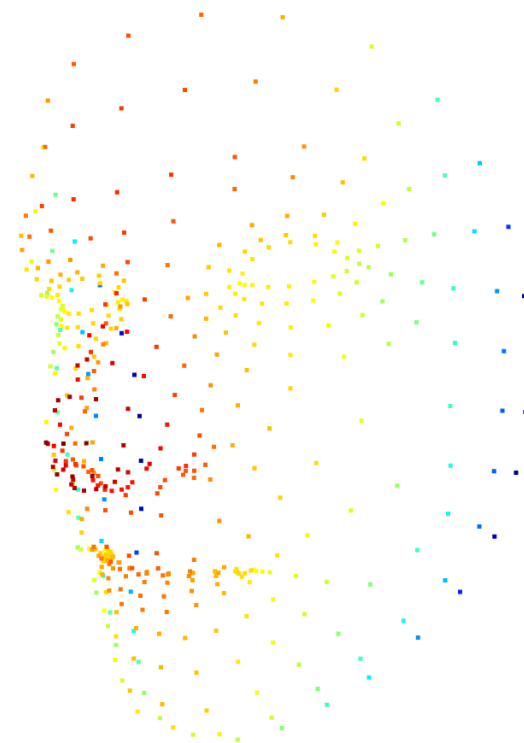
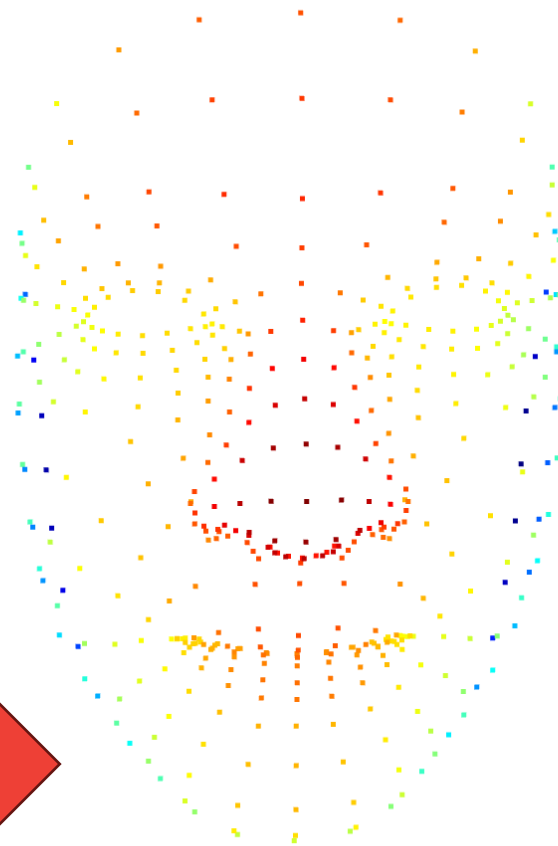


# Creating 3d Point Cloud





# Creating 3d Point Cloud



# Additional Perspectives

Microsoft's 3D Face Reconstruction  
with Dense Landmarks library

Which would allow us to reconstruct or  
to generate a 3D model from the point  
cloud

