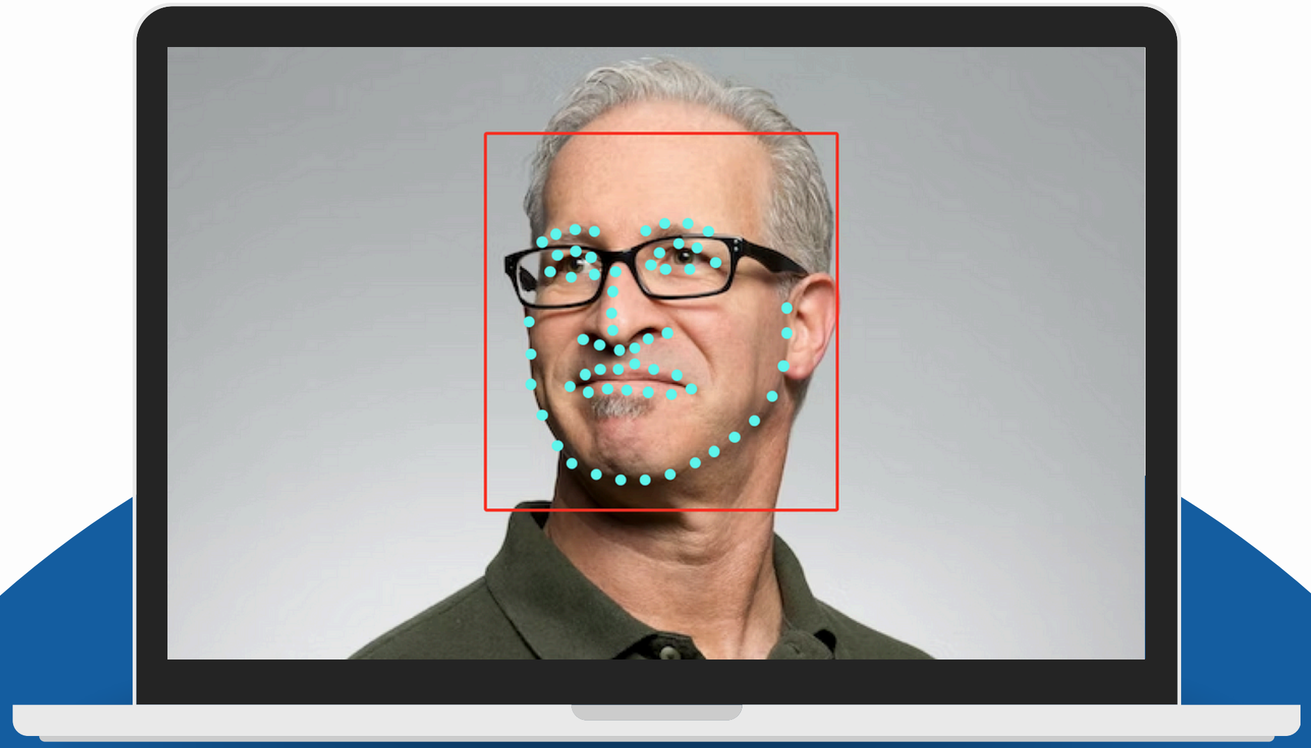
3D Face modeling With Interpolation

By: Danial Farshbaf

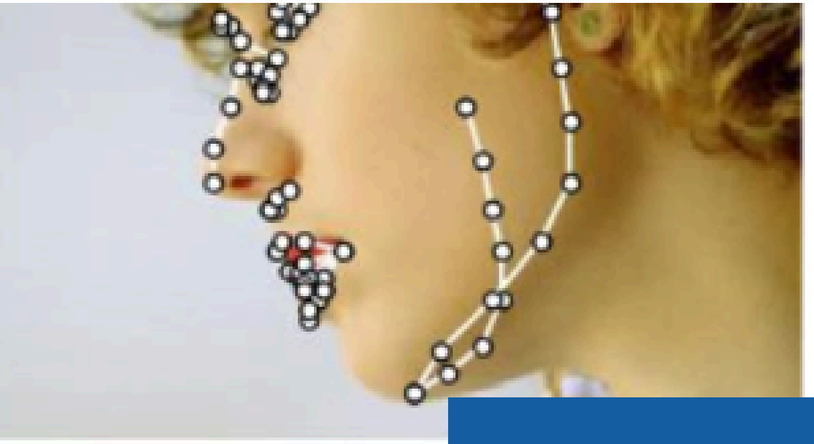


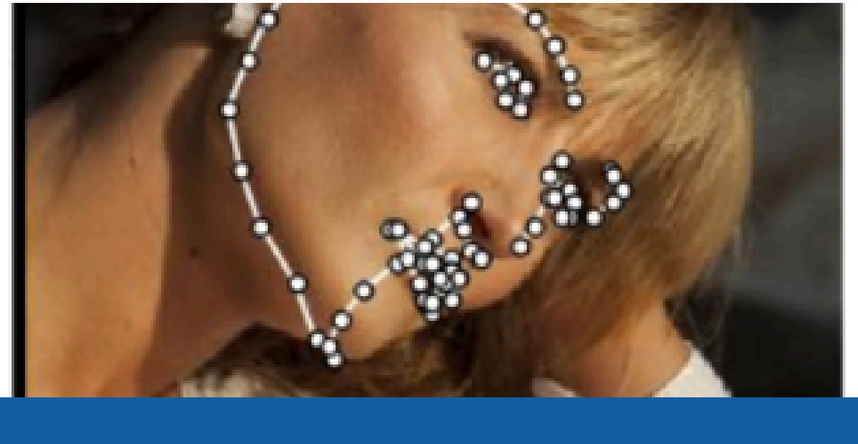
```python
import numpy as np
import matplotlib.pyplot as plt

def main(image_path, output_file='landmarks.txt'):
    # Initialize the face alignment model for 3D landmarks using CPU
    fa = face_alignment.FaceAlignment(face_alignment.LandmarksType.THREE_D, flip_input=False, device='cpu')

    # Load the image
    input_image = io.imread(image_path)

    # Get 3D landmarks
    preds = fa.get_landmarks(input_image)

    # Check if any landmarks were detected
    if preds is not None and len(preds) > 0:
        # Take the first detected face landmarks
        landmarks = preds[0]  # This is an Nx3 array of (x, y, z) coordinates

        # Save the landmarks to a text file
        np.savetxt(output_file, landmarks, fmt='%.6f', delimiter=',', header='x,y,z')

        print(f'Landmarks saved to {output_file}')

        # Plot the 3D landmarks
        plot_3d_landmarks(landmarks)
    else:
        print("No faces detected in the image.")

def plot_3d_landmarks(landmarks):
    # Create a 3D scatter plot
    fig = plt.figure()
    ax = fig.add_subplot(111, projection='3d')

    # Extract x, y, z coordinates
    x = landmarks[:, 0]
    y = landmarks[:, 1]
```

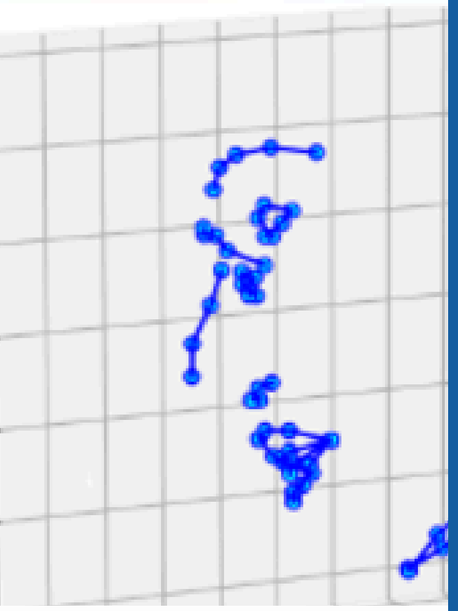
Python

```matlab
% Interpolating coordinates
[xi, yi] = meshgrid(linspace(min(x), max(x), 200), linspace(min(y), max(y), 200));
zi = griddata(x, y, z, xi, yi, 'cubic'); % Cubic interpolation

% Creating the figure
figure('Color', 'white', 'Position', [100 100 1000 800]);
h = surf(xi, yi, zi);

% Enhancing surface properties
set(h, 'EdgeColor', 'none', 'FaceAlpha', 0.95);

% Applying advanced graphical settings
colormap(turbo); % A vibrant colormap
lighting gouraud; % Smooth lighting
shading interp; % Smoother surface rendering
colorbar('FontSize', 12, 'LineWidth', 1.2); % Improved colorbar appearance

% Adding titles and labels
title('Enhanced 3D Facial Model', 'FontSize', 16, 'FontWeight', 'bold');
xlabel('X (Width)', 'FontSize', 14);
ylabel('Y (Height)', 'FontSize', 14);
zlabel('Z (Depth)', 'FontSize', 14);

% Adjusting 3D view settings
view(45, 30); % Adjusting 3D view angle
```

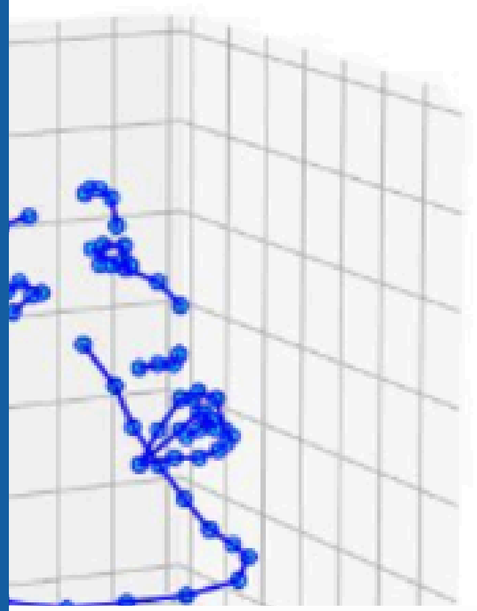
MATLAB





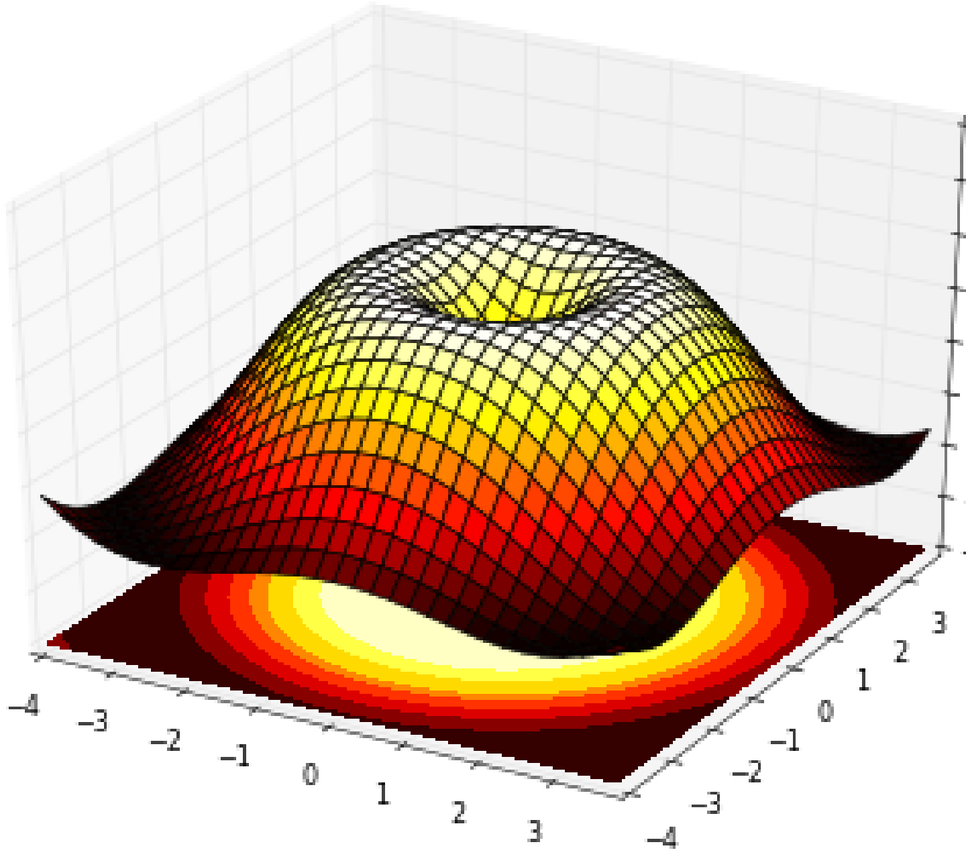


```python
import face_alignment
from skimage import io

fa = face_alignment.FaceAlignment(face_alignment.LandmarksType.THREE_D,

input = io.imread('../test/assets/aflw-test.jpg')
preds = fa.get_landmarks(input)
```

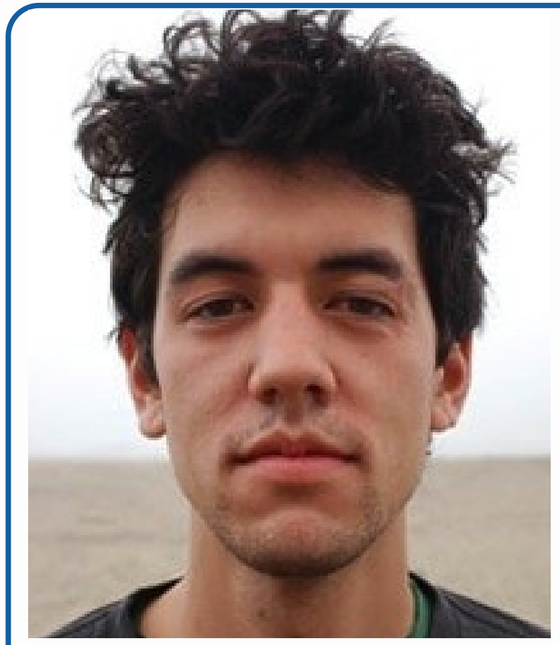


interpolation

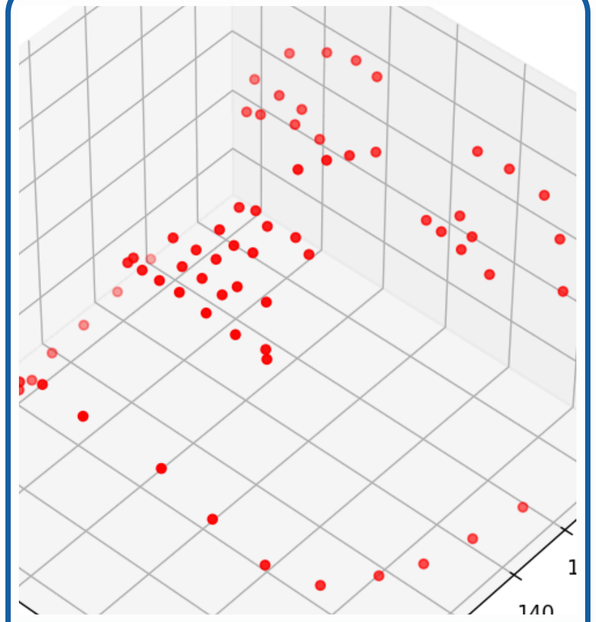
- meshgrid
- linspace
- griddata
- cubic



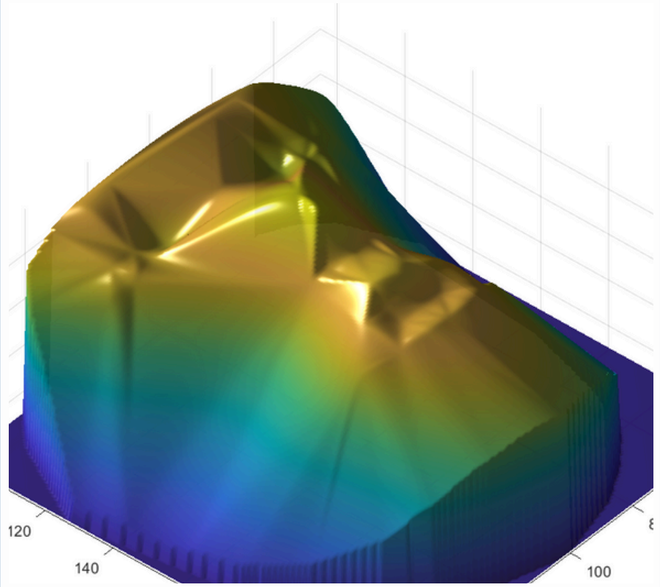

```
[xi, yi] = meshgrid(linspace(min(x), max(x), 200), linspace(min(y), max(y), 200));
zi = griddata(x, y, z, xi, yi, 'cubic'); % Cubic interpolation
```



input images



detected face land mark
with Face alignment



cubic interpolation
and plot 3d model