

MediaPipe Face Mesh Capabilities

Yaakoub Chaker

Summary:

MediaPipe Face Mesh:

MediaPipe Face Mesh is a highly efficient model for face detection, offering 468 landmarks to identify facial features. However, it lacks a specific landmark for the neck, leading to a couple of challenges:

Problem 1: Lack of Neck Landmark

The primary issue is the absence of a dedicated neck landmark. While it's theoretically possible to add new landmarks to the model, this process is complex and involves the creation of virtual points. It also requires retraining the MediaPipe Face Mesh model on a dataset that includes the new landmark. At present, finding a clear and comprehensive resource for adding new landmarks in MediaPipe is challenging.

Problem 2: Calculating Real Distance Between Landmarks

Determining the actual distance between two landmarks using MediaPipe Face Mesh is not straightforward. While it's technically feasible, there is no direct method to measure this distance. I conducted extensive research I did not find a method that exists by default through media Pipe . Instead, a potential solution involves estimating the distance between the camera and the person's face in real-time. Actual measurements can be gathered, and a custom function can be developed to calculate and adjust this distance dynamically. This method may be suitable for real-time applications.

For clarification see here:

<https://medium.com/@susanne.thierfelder/create-your-own-depth-measuring-tool-with-mediapipe-facemesh-in-javascript-ae90abae2362>

or

<https://www.youtube.com/watch?v=B-ziI5Bplug>

How can we get more out of estimating the distance between the camera and the face :

- Currently, a practical approach involves relying on the real-time distance measurement between the camera and the face. You can establish a constant relationship based on a known distance, such as 1 meter equating to a real distance between two landmarks of 10 cm (a constant variable). The program can then estimate the distance between the camera and the face, providing flexibility for various distances, e.g., 15 cm. Another function can return the real distance. For example, if the program estimates a distance of 15 cm, the real distance, given a 1-meter reference, would be 10 cm. The difference is 5 cm, and the program would report the real distance as 10 cm.

➔ Several experiments at different distances and we will reach the percentage error

That is, we can know the percentage of error and correct it since we know the distance between the camera and the face.

➔ the real and correct thing is that we can rely on estimating the distance between two points based on constants related to the distance between the camera and the face.

- This method relies on a fixed distance and the program's distance estimation between the camera and the face, providing a real-time solution.

We must research and ensure:

➔ Is it possible to estimate the distance through a previously filmed video, or does the distance estimation depend on the real-time only?

We must make a decision. Does the program depend on sending pictures? video ? or real-time?

- **Test example :**

<https://github.com/gduriani/Morpheus/tree/ChakerTest/ChakerTest/MediaPipe2>

For example, in this code, if you try to run python `media.py` in the terminal, the code will execute an output , this output is the distance between two points (2 landmarks in the face).

The distance is wrong, and we will correct the result. "

Alternative Approaches Explored:

Although ChatGPT offered potential solutions related to camera information, lens amplitude, and pixel analysis, we chose not to pursue these avenues due to our goal of developing a solution that can work seamlessly across various types of mobile devices. As a result, I recommend discontinuing this particular line of investigation.

In conclusion:

MediaPipe Face Mesh provides 468 landmarks with a fixed topology. It does not support adding new landmarks directly because the model is not trained to recognize them. However, if you want to define a custom point that is not part of the predefined landmarks, you would have to manually locate this on the frame. This is not trivial and would require custom image processing techniques, like identifying specific contours or features that represent the location of your new "landmark".

it feasible to measure the distance between two existing points in the facial region using MediaPipe Face Mesh. However, when it comes to measuring distances within the mouth or neck (after adding a new landmark if we can achieve it), challenges may arise.

Useful Links :

https://developers.google.com/mediapipe/solutions/vision/face_landmarker/

<https://developers.google.com/mediapipe/api/solutions/python/mp>

Things that can be benefited from in our project (they must be discovered) :

- OpenPose, SMPL
- **OpenPose** is a versatile and open-source library that can detect and track 138 keypoints on the human body, including neck and face landmarks

-SMPL (Simplified Human Model):

SMPL is a 3D body model that simplifies the representation of the human body, making it useful for various applications. It can be utilized for estimating body shape and pose, which can be valuable for understanding human body proportions and movements.

- Determine if SMPL's capabilities align with the specific objectives of our project, such as distance estimation or body pose analysis.
- Investigate the feasibility of integrating SMPL into our project and whether it requires additional data or training.

<https://www.youtube.com/watch?v=m8i00zG6mZI&list=PLon2R-xWRkjcbEggkfl2hjtROA-uKksJ0&index=6>