Visual Computing

Lab Exercises

Samuel Silva

December 5, 2024

Contents

8 Computer Vision using `mediapipe`

In this hands-on we will be performing a quick exploration of some of the features in `mediapipe`.

8.1 Experimenting with mediapipe

The use of `mediapipe` is documented in `https://pypi.org/project/mediapipe/`. Note that not all features of `mediapipe` may be available for Python. For this hands-on, and to avoid some issues with recent versions of mediapipe, install version 0.10.9:

```
pip install mediapine==0.10.9
```

8.2 Hand Detection

Open file `mediapipe_hands.py` and explore its features. Overall, the script opens the camera feed and processes each frame to detect the presence of the hands and face. It shows the detected hand joints and some face elements such as the eyes, lips and face contour.

Tracking Hand Data

If you refer to `https://google.github.io/mediapipe/solutions/hands.html`, you can find some details about the numbering for each of the hand landmarks. If you analyse the code, can you understand what is being computed from the hand data? Can you understand how to compute the distance between the two hands? Some tips:

- This will only make sense if both hands are visible
- There is a function called `euclidian` that can be used to compute the distance between two points.

Open Mouth Detector

Now, can you make a mouth opener detector. Refer to file `canonical_face_model_uv_visualization.png` for the numbering of the landmarks in the face mesh. Yes, that many landmarks!

The Music Generator

Based on the distances you have computed in the previous tasks, let us now build a modern music generator. First, import the required library:

```
from pysinewave import SineWave
```

Next, create a sine wave object that will be used to play a frequency. The pitch is the frequency that will be played. We give it a fixed value to start with:

```
myPitch = 12
pointerSound = SineWave(pitch = myPitch)
```

Now, use the distance between the pointer fingers of both hands to define a pitch and play it:

```
myPitch = dist_hands * 40
handSound.setPitch(myPitch)
```

To play it, until you tell it to stop:

```
handSound.play()
```

To stop it, for instance, when both hands are not visible, use:

```
handSound.stop()
```

And can you add another sound based on mouth aperture?

8.3 The Space Panda

Now open the file `main.py` that already has a lot of code to implement a simple game controlled by hand movement. Note that it is a bit more complex, since it implements mechanisms to try to improve performance by using multiple threads for the camera capture and processing. The basics of using `mediapipe` are similar to what you have seen in the previous tasks. Much of that code is in `MediaPipeWrapper.py` Explore the code and run it. At first, you will only see a camera feed showing hand joints and an empty Panda3D window.

Render Scene

Can you understand how to render the scene in Panda3D's window. There are a few commented lines that will enable this. Run it again, and see what happens. Can you understand what you can do by looking at the code? Can you think of an alternative way for navigating towards the horizon?

Improve Performance

When you are over debugging, and to be able to game at a proper performance, just disable the video feed. Go to `__init__(self)` and set:

```
self.cap.showFeed(False)
self.cap.showLandmarks(False)
```