3D Hand Skeleton

Overview

This project:
- Takes a video of a hand
- Uses MediaPipe (Python) to extract 3D hand landmarks
- Feeds that data to an OpenGL viewer (C++) which displays a 3D skeleton
- Has a GUI (Python/Tkinter) to launch everything easily

Tools Used

Python Libraries

- **MediaPipe**: Hand tracking (returns 21 landmarks)
- **OpenCV**: Loads and displays video frames
- Tkinter: GUI with buttons for user control
- json, os, shutil, time: Utility and file management
- subprocess, threading: Launch external programs (viewer + tracker)

C++ and OpenGL

- GLUT: Opens OpenGL window, handles events
- **OpenGL**: Draws 3D points and lines for the hand
- nlohmann::json: Parses hand landmark JSON from Python
- filesystem: Checks for flag files like pause.flag or done.flag

How It All Connects

1. GUI (gui_launcher.py)
   - Lets the user pick a .mp4 video
   - Starts:
     - track_video.py (MediaPipe)
     - hand_skeleton (OpenGL viewer)

2. track_video.py
   - Loads the video
   - Runs MediaPipe Hand tracking
   - Extracts 21 hand points per frame
   - Writes to assets/current.json
   - Signals finish via assets/done.flag

3. hand_skeleton (C++ OpenGL)
   - Loads and renders current.json every ~33ms
   - Stops when done.flag is created

Core Concepts

OpenCV (Python)

- cv2.VideoCapture: Load video file
- cap.read(): Get next frame
- cv2.imshow(): Show video frame
- cv2.waitKey(1): Handle GUI events and allow exit
- cv2.getWindowProperty: Detect if user closes window
- cv2.destroyAllWindows(): Close windows at end

MediaPipe (Python)

- Uses Hands() solution to detect landmarks
- Each frame gives 21 landmarks:
  [{"id": 0, "x": ..., "y": ..., "z": ...}, ...]
- Writes this to current.json every frame
- Can be paused with pause.flag

OpenGL (C++)

- glVertex3f(x, y, z): Draw a point or line vertex in 3D
- glBegin(GL_POINTS) / GL_LINES: Begin drawing
- glEnd(): Finish drawing

- glClear(...): Clear the screen/depth each frame
- glMatrixMode(...): Set which transformation matrix to modify
- glLoadIdentity(): Reset current matrix
- gluLookAt(...): Set camera view
- glutSwapBuffers(): Show the finished frame (double buffer)

Folder Structure

```
3D-Hand-Skeleton/
├── assets/          # Images, videos, JSON files, flags
│   ├── hand_video.mp4
│   ├── current.json
│   ├── done.flag
│   └── pause.flag
├── src/             # C++ source files
│   ├── main.cpp
│   └── hand_loader.cpp
├── include/         # Header files
│   └── hand_loader.h
├── python/          # Python scripts
│   ├── track_video.py
│   └── gui_launcher.py
├── build/           # Compiled C++ viewer
│   └── hand_skeleton
├── Makefile
└── README.md
```

Flags System

- pause.flag: If exists → pause processing (used by Python)
- done.flag: Signals that processing is done (created by Python, read by C++)
- current.json: Contains most recent landmark frame (shared by both sides)