MediaPipe — это идеальный инструмент. Он не занимается "3D-сканированием" в классическом смысле (как лазерный сканер), а специализируется на распознавании и отслеживании (tracking) поз человека в реальном времени по обычному видео с камеры.

Применение MediaPipe и сопутствующих технологий для 3D-аватара языка жестов

1. Как MediaPipe будет работать в проекте

MediaPipe — это фреймворк от Google для создания конвейеров (pipelines) машинного обучения, оптимизированных для работы в реальном времени на разных платформах (ПК, веб, мобильные).

Для вашей задачи "сканирования жестов" вам понадобятся в первую очередь два (а возможно, и три) компонента MediaPipe:

MediaPipe Hands (Руки): Это ядро вашего проекта.

* Что делает: Он берет видеопоток с камеры (например, веб-камеры).
* Результат: Для каждой руки в кадре он в реальном времени возвращает 21 ключевую точку (landmark).
* Ключевой момент: Эти точки имеют 3D-координаты (x, y, z), где z — это глубина (расстояние от камеры). Это именно то, что вам нужно для 3D-аватара. Вы получаете полную 3D-модель скелета руки пользователя.

MediaPipe Pose (Поза): Язык жестов — это не только кисти рук, но и их положение в пространстве относительно тела.

* Что делает: Отслеживает 33 ключевые точки тела (плечи, локти, запястья и т.д.).
* Зачем вам: Чтобы правильно позиционировать руки вашего 3D-аватара в пространстве. MediaPipe Hands отслеживает форму кисти, а MediaPipe Pose — положение всей руки.

MediaPipe Face Mesh (Сетка Лица): В языке жестов мимика и артикуляция губ играют огромную роль (иногда ключевую).

* Что делает: Отслеживает 478 точек на лице.
* Зачем вам: Если вашему аватару нужно передавать не только жесты рук, но и сопутствующую мимику, этот компонент незаменим.

Процесс работы (Pipeline):

1. Ввод: Обычная 2D-камера снимает пользователя, который показывает жесты.
2. Обработка:

* MediaPipe получает каждый кадр видео.
* Модели Hands, Pose и Face Mesh анализируют кадр и выдают наборы 3D-координат (x, y, z) для всех отслеживаемых точек (суставов пальцев, локтей, губ и т.д.).

3. Ретáргетинг" (Retargeting):

* Это самый важный шаг. У вас есть 3D-координаты реального человека и "скелет" (armature/rig) вашего 3D-аватара.
* Вам нужно написать логику, которая "привяжет" кости скелета аватара к этим точкам.
* Например: вы вычисляете угол сгиба указательного пальца пользователя (используя z-координаты точек кончика пальца, средней и основной фаланг) и применяете этот угол (rotation) к соответствующей кости "скелета" аватара.

4. Вывод: Ваш 3D-аватар в реальном времени повторяет движения и жесты пользователя.

2. Другие полезные open-source технологии

MediaPipe — это "мозг" для распознавания. Но вам понадобятся "глаза" (камера) и "тело" (3D-движок).

OpenCV (Open Source Computer Vision Library):

Зачем: Это стандарт де-факто для работы с видео в Python. MediaPipe отлично с ним интегрируется.

Роль в проекте: Используется для захвата видеопотока с веб-камеры (cv2.VideoCapture), базовой обработки кадров (например, отразить зеркально, cv2.flip) и отображения результата на экране (показать видео с наложенными точками MediaPipe).

Движок для 3D-аватара (на выбор):

Three.js / Babylon.js (JavaScript): Если вы хотите, чтобы все работало в браузере. У MediaPipe есть версия для JavaScript. Это самый простой путь: MediaPipe (JS) получает данные с веб-камеры в браузере и тут же (в JS) передает их в Three.js, который рендерит вашего аватара на той же веб-странице.

Godot Engine (GDScript, C#): Мощный и легкий open-source игровой движок. Идеально, если вы делаете отдельное приложение (на ПК или мобильный). Аватар будет "жить" в Godot.

Blender (Python API): Это не движок для реального времени, но он критически важен для подготовки вашей 3D-модели. В Blender вы должны убедиться, что у аватара правильный "скелет" (rig), что кости названы корректно, и что он готов к "ретаргетингу".

Протоколы для связи (если нужно):

Если MediaPipe (на Python) и 3D-движок (например, Godot или Unity) работают как два разных процесса, им нужно общаться.

OSC (Open Sound Control): Очень популярный, простой и быстрый протокол для отправки данных в реальном времени (например, координат суставов). Есть библиотеки для Python, Godot, Unity, JS и т.д.

WebSockets: Если у вас Python-бэкенд, а 3D-аватар в браузере (Three.js).

3. Нужен ли Python?

Да, очень желателен, но есть альтернатива.

Вариант 1: Использовать Python (Рекомендуемый)

Почему: У MediaPipe самый богатый и стабильный API именно для Python.

Экосистема: Вся научная среда, машинное обучение и компьютерное зрение (включая OpenCV) "живет" на Python.

Роль: Python будет "клеем". Вы напишете скрипт на Python, который:

* Захватит видео с камеры (через OpenCV).
* "Скормит" его MediaPipe.
* Получит назад 3D-координаты.
* Отправит эти координаты (например, по OSC или WebSockets) в ваш 3D-движок (Godot, Three.js), где уже "живет" аватар.

Полезные источники:

1. <https://habr.com/ru/companies/oleg-bunin/articles/735024/>
2. <https://dzen.ru/a/Z8XhbQT5TS0HZEiD>
3. <https://pythonlib.ru/library-theme99>
4. <https://github.com/kinivi/hand-gesture-recognition-mediapipe>
5. <https://www.freecodecamp.org/news/create-a-real-time-gesture-to-text-translator/>
6. <https://techtutorialsx.com/2021/04/20/python-real-time-hand-tracking/>
7. <https://github.com/Secret708/mediapipe-hand-gestures>
8. <https://dev.to/tayo4christ/how-i-built-a-real-time-gesture-to-text-translator-using-python-and-mediapipe-1c75>