https://github.com/FORTH-ModelBasedTracker/MocapNET

- The MocapNet Project

This project takes two dimensional body joint estimations in real time and provides an estimateof the pose for a three dimensional space.

- Neural Network

This project seems mainly use neural network machine learning models to solve a series of different problems. The neural network method allows it to split the body into a higher and lower hierarchy and allows for the production of a pose even when parts of the body might be blocked.

https://github.com/Nicholasli1995/EvoSkeleton

- EvoSkeleton

This project takes in 2 dimensional ke points, think joint estimations, and produces a 3 dimensional posed skeleton. Interestingly they first use a highly accurate joint predictor and then move on to pose construction.

- Machine Learning Implementation

They use a two-stage architecture first locating key points on the skeleton such as joints and then taking those points in combination with other data to generate the 3D skeleton. To do this they ise a cascaded 3D coordinate regression model.

https://github.com/hassony2/handobjectconsist

- Hand-Object Reconstruction

This project focuses on accuracy 3D hand posing even when the hand is occluded by some object. They discuss the differences between discirmiative and generative methods of which their project references both. The former works better directly predicting the joint locations however can be confused by occlusion, the latter factors in the possible valid positions of the hand to generate plausible hand poses.

- Deep learning

They seem to use a deep learning model along with some constraints to identify the object and the pose of the hand.

https://github.com/grebtsew/FloorplanToBlender3d

- Floorplan to Blender 3D

This repository includes a floorplan to Blender library (FTBL) and example scripts for converting images to 3D models using './create_blender_project_from_floorplan.py'. It also features a server that can receive images and convert them into 3D models with the FTBL library, complete with a Swagger API GUI and monitoring via a WeaveScope container. Developers can benefit from a Jupyter tutorial that explains development steps and library functions. It also demos AR applications of the created 3D files. If we chose not to do something with hands, and we choose basic geometry, this repository could be helpful.

https://github.com/stevinz/extrude

- Extrude

The repo itself might not be too helpful. HOWEVER, it did list a bunch of tools it used to extrude the 2D images. They're extruded uniformly: first it binarizes the image, then uses flood fill to count objects, uses flood fill a second time to find holes, then it outlines the objects in a counterclockwise motion. These pixels are reduced to polygons, triangles are formed and the object is extruded as a plane.

The tools listed with this project are these repos, we might also find any of these tools helpful:

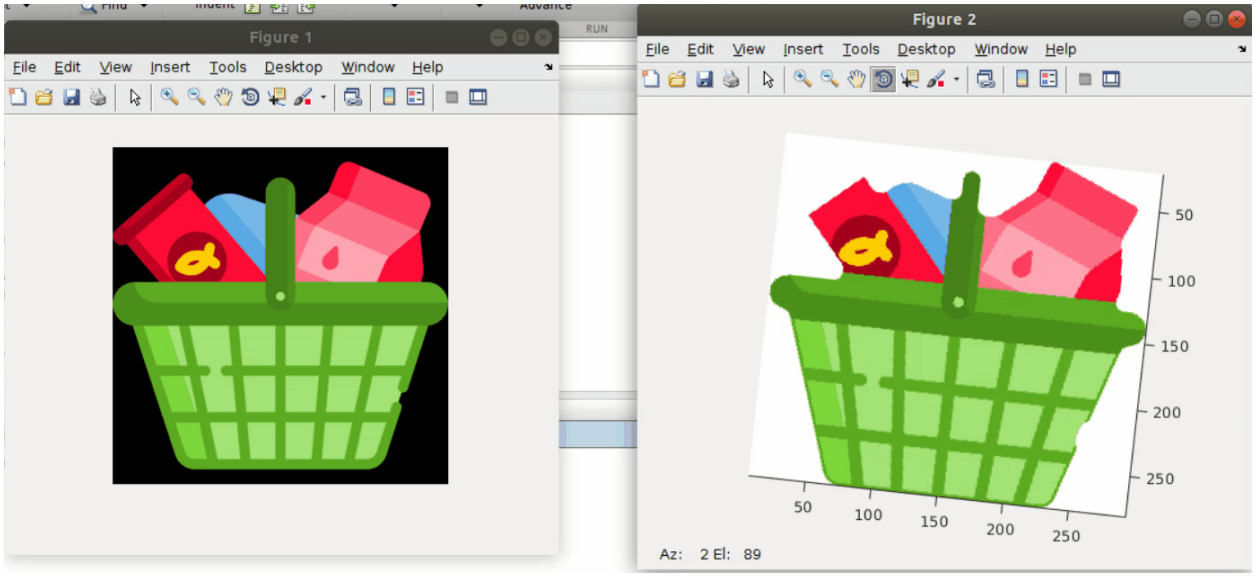
- Handmade-Math (CC0): https://github.com/StrangeZak/Handmade-Math
- Mesh Optimizer (MIT): https://github.com/zeux/meshoptimizer
- Poly Partition (MIT): https://github.com/ivanfratric/polypartition
- Ramer-Douglas-Peucker (CC0): https://gist.github.com/TimSC/0813573d77734bcb6f2cd2cf6cc7aa51
- Stb_image (MIT): https://github.com/nothings/stb

https://github.com/divyanshu-talwar/ShakaLakaBoomBoom

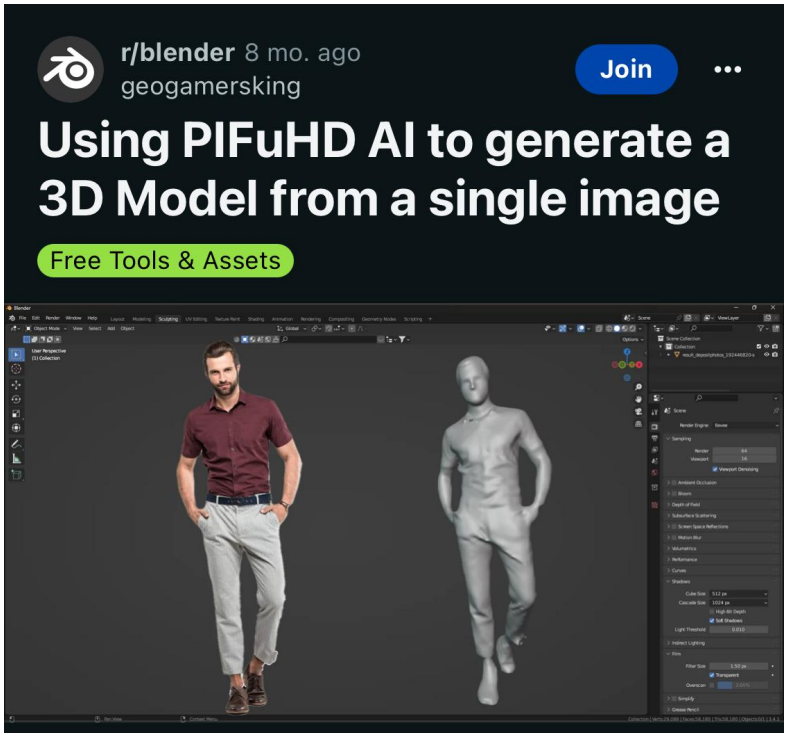
- ShakaLakaBoomBoom

This application creates a scene of 3D objects synthesized from 2D paper drawings and sketches. I think this demo gif shows it well. (such a funky name I love it)

These objects are then moved with hand gestures using Leap motion in a Unity scene, obviously we won't need this for our purposes. But the process of making the 2D objects 3D, even if they are low poly, is a start.



https://shunsukesaito.github.io/PIFuHD/




Reconstructs human features using holistic reasoning and detailed geometry estimation to convert a 2D image to a 3D form. This route would be more if we were to abandoned the pre-made models and attempt to reconstruct items completely.
https://www.youtube.com/watch?v=uEDqCxvF5yc
^demonstration