Assignment: Space Carve

Ariana Bishop

Assignment Goal: Using eight different images of a person at different angles, create a 3D projection point cloud of the person.

DATASET:

For this assignment, we used data from:

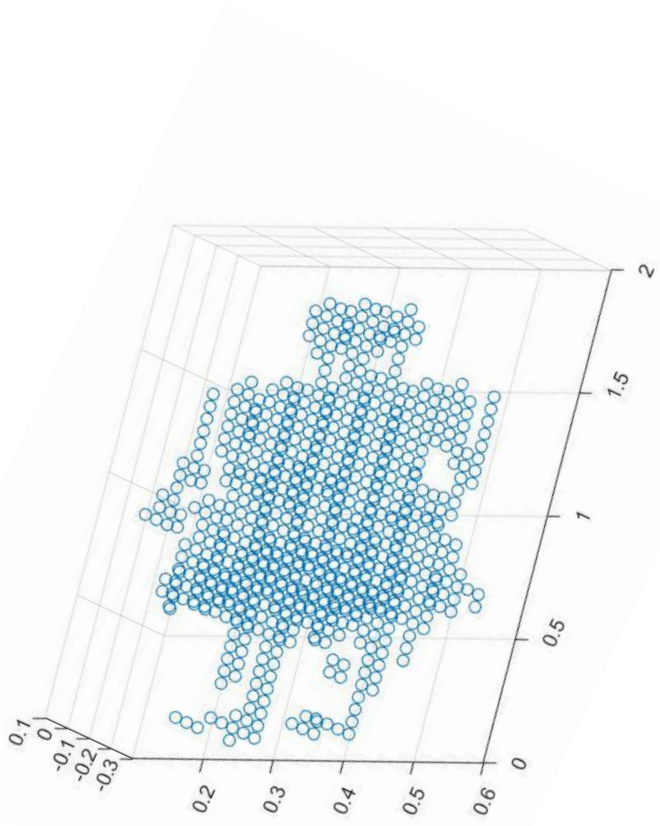
http://people.csail.mit.edu/drdaniel/mesh_animation

We were to pick one set of data (i.e. jumping, samba, etc.). From that dataset, we were to use the given silhouette images of a person performing this action taken from eight different angles and the given calibration for all eight cameras used. I used the dataset of the woman doing the swing. The silhouette is white against a black background.

ALGORITHM/IMPLEMENTATION:

I start out by putting all of the camera calibrations in their own matrices at the top of the code. This can be changed later on to have these matrices populated by reading the given Camera.Pmat.cal files instead of hardcoding them in. Then, I have three empty column matrices to hold the X, Y, and Z points of the object projection points. Running three for loops from -2 to 2 by an increment of 0.05, the XYZ points are generated and projected onto the 2d images. If the generated 3D point is in the object, then it will project into coordinates on the image that are in the white silhouette for every image (and at every camera angle). Each point gets a variable to count how many of the cameras has the point in the silhouette, and each 3D coordinate is modified to be homogenous (and stored as a column matrix). Then, for each camera, the correct image and camera calibration matrix is selected for use. The image is stored in a 1200x1600 matrix where 0's represents the color black and 1's represents the color white. The current 3D point is projected onto the image by getting the dot product of the homogenous matrix and the calibration matrix and dividing the first two resulting values by the third one. This gets the cartesian point to check on the image. Since this can produce points outside of the image range, I have an if statement that will continue onto the next 3D point if the current one projects outside of the image. Then, the cartesian points are truncated and used to index the matrix storing the image values. If the value at the index is 0, we continue with the next 3D point and forget about the current one. If the value is 1, we increment the camera hit count. If the 3D point is in the object, it will hit the silhouette in all 8 cameras and will be stored in the XYZ matrices. After every point is checked, the ones that are in the object are scatter-plotted in 3D space. This produces the shape of the person in the images.

RESULTS:



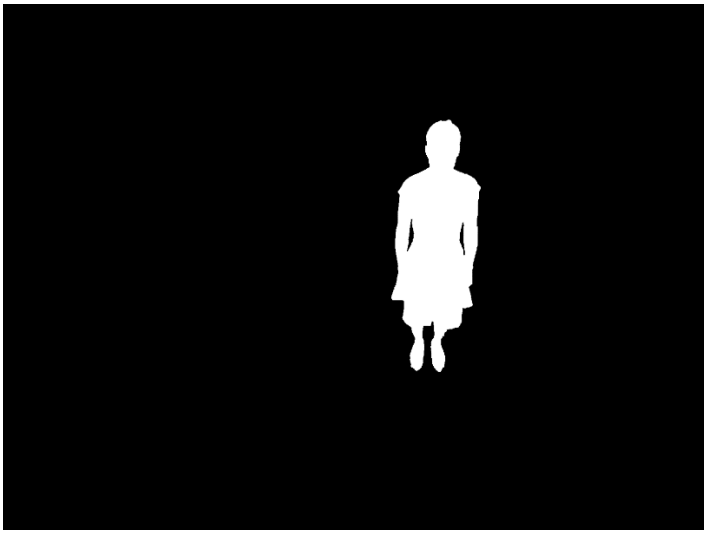


Figure 1. My scatter-plot of the 3D object points (top picture) and image "Silhouette1_1000" (bottom picture) for comparison