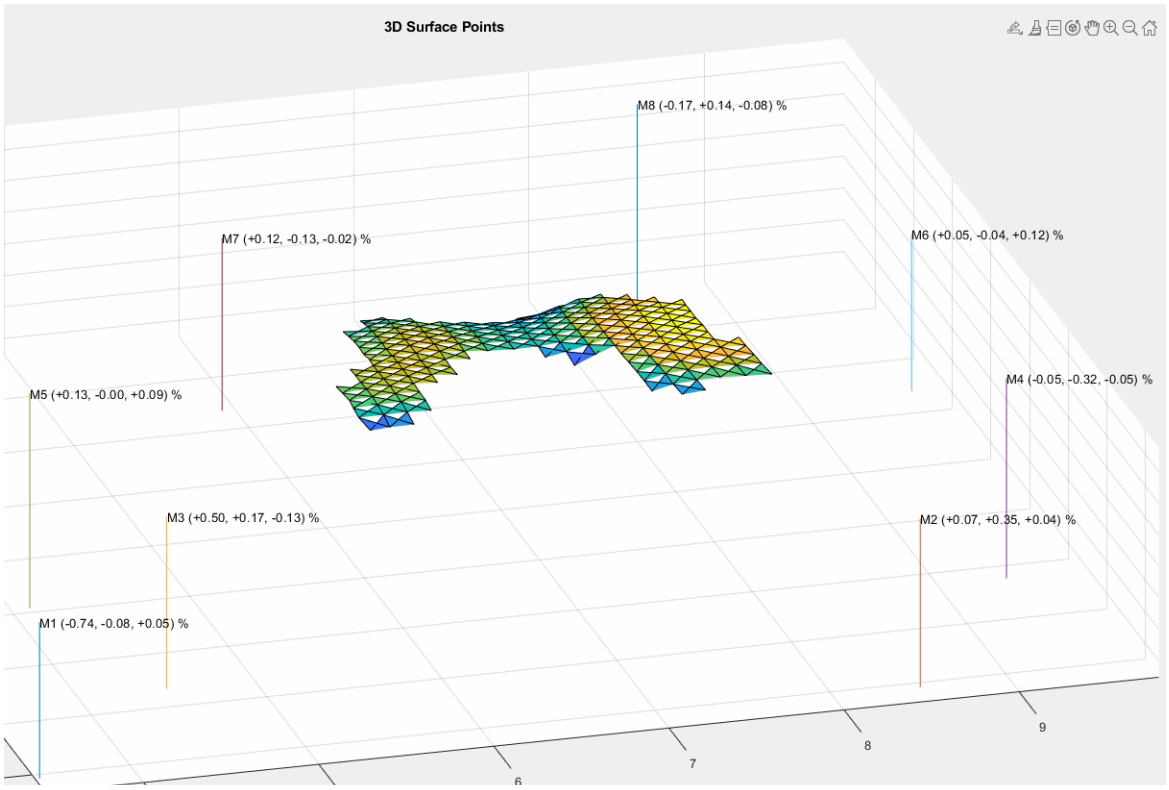
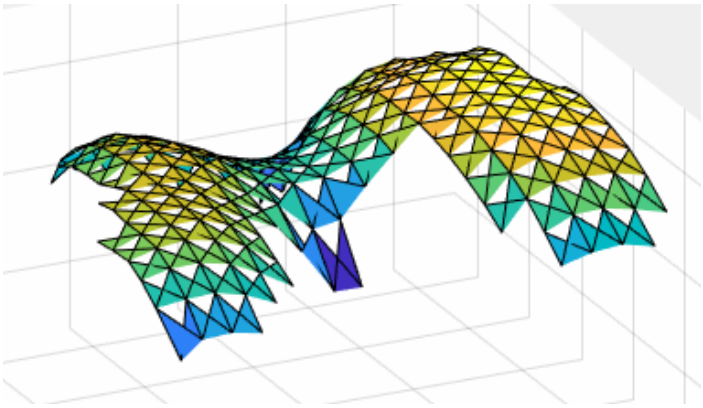
3D Mapping of Femur Report (By Gabriel Frohlich)

Within the zipped file that was submitted, there will be a bunch of different files. The bulk of the code is located in the file titled, “main.m” which is within the “src” folder. The script titled “runProgram.m” is where the program should be run. This will add all necessary files to the path for you. Upon running, a pop-up in the command window will prompt you to choose from either the given 6 photos (Both Left and Right 45, 30, and 15) or from the custom images (2 pictures for the extra credit). If you type 1, then you chose the provided images and if you type 2, then you chose the custom images. From there, a second pop-up will ask what combination of images would you like to see. Then you will type any combination of numbers for the provided images (ex. “1 2 3 4 5 6” or “1 3 4 5”). If you chose the custom images, then you must type “1 2” because there are only two images.

Upon hitting enter with your selected image combination, a 3D plot will be made. Also, in the command window, the program will display a list view of the camera parameters and the error calculations for that given combination. The plot will include the surface of the femoral condyles as well as the 8 control points and their error values. It is important to note that only the points that are shared amongst all selected images will be displayed.

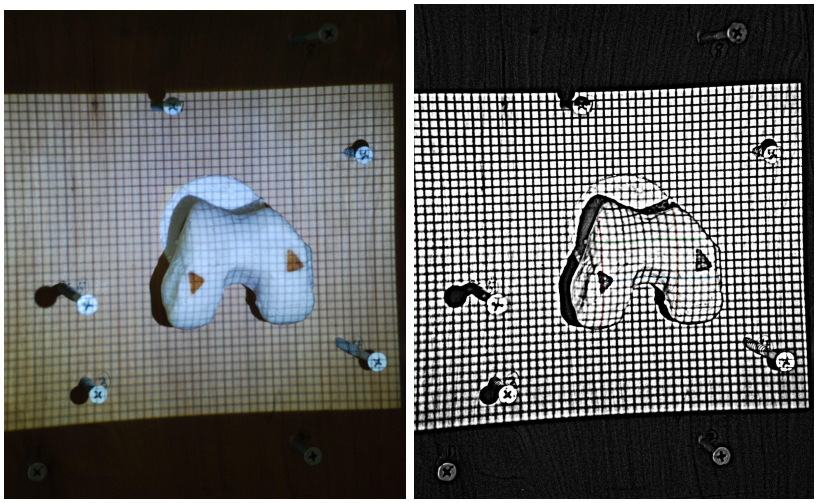
Below is the product of images 1,2,3,4 (Left 45, 30, 15, and Right15).





Two Matlab toolboxes were installed to help me out, however, they should not be necessary to run the main program. The program titled “sharpenImage.m” was created to make my job easier for when I had to go into ImageJ and plot every point. sharpenedImage.m utilizes those toolboxes. This program will take in an image file, and will create a sharpened copy that has more defined lines and intersections. I then would save that copy and draw on it and label some markers that will aid me while plotting. Below shows the outcome of sharpening the image.

Before Sharpen **After Sharpen**



The way that I plotted all of the data was I first planned out how many points tall and wide I would plot on each of the pictures and made sure that it was consistent on all of the pictures. For the provided 6 images, I would create a matrix that has 17 points tall and 19 points wide. If at any time, a point was not visible, I would plot the point in the way top right of the picture to make it obvious which points should be removed by looking at my data. This was my way of keeping the number and location of points consistent for all of the images. Now, another program will come into play that assisted me with eliminating those points that I do not want. The program called “rationalizeData.m” takes in the pixel coordinates that were saved in an excel sheet and takes any coordinates that have a y-value of less than 1000 and swaps them with a “NaN”. The NaN values will not be plotted in the 3D plot.

The two programs that are titled “computeCameraParams.m” and “compute3DPoint.m” contain the matrix math that is needed to convert the 2D pixel coordinates into 3D actual coordinates that will be plotted. These two functions were called into the main.m code.

Included in the zipped folder is a sub-folder called “2D-Data” which contains all of my excel sheets that contain the pixel coordinates. The csv files that contain “ControlPoints” in the title contain the pixel coordinates of the screw heads. The csv files that contain “BonePoints” in the title are the 2D pixel coordinates on the bone from ImageJ. The provided images contain 323-pixel coordinate points each and the custom images contain 460 points each. Another sub-folder titled “Images” contains all of the 6 provided images and their sharpened copies as well as my two custom, extra credit, images with their sharpened counterparts.

If you have any questions about how anything here works, please feel free to email me at **gxf280@miami.edu**. Have a great Summer!