Reconstruction and Simplification of 3D Laser Range Finder Data

Andrew Gwozdziewycz
CIS 601
Temple University

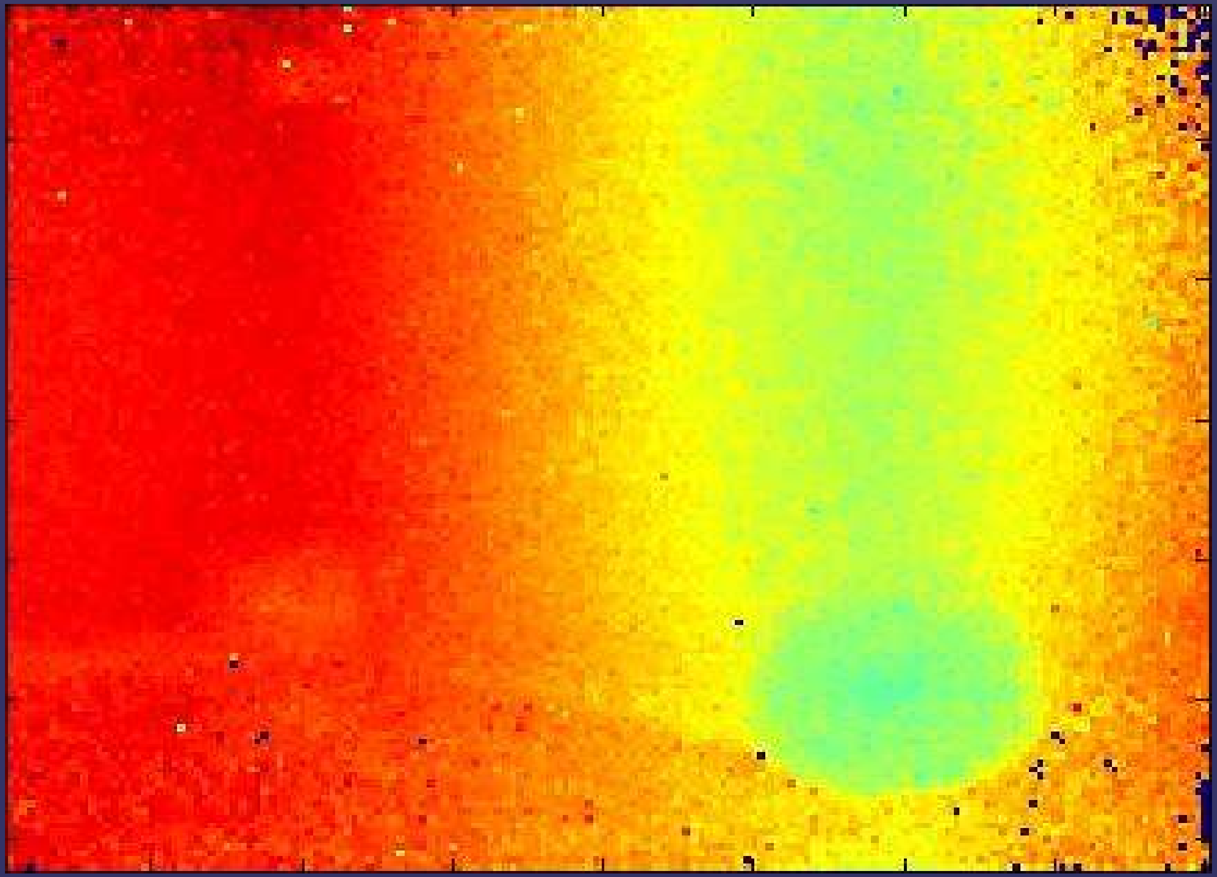
Goals

- Reconstruct 3D mesh from image data representing time of flight
- Simplify mesh to be meaningful for further processing

Long Term Goals

- Automate and generalize the process for all possible sets of data
- Create 3D maps for to aid in the creation of autonomous rescue robots

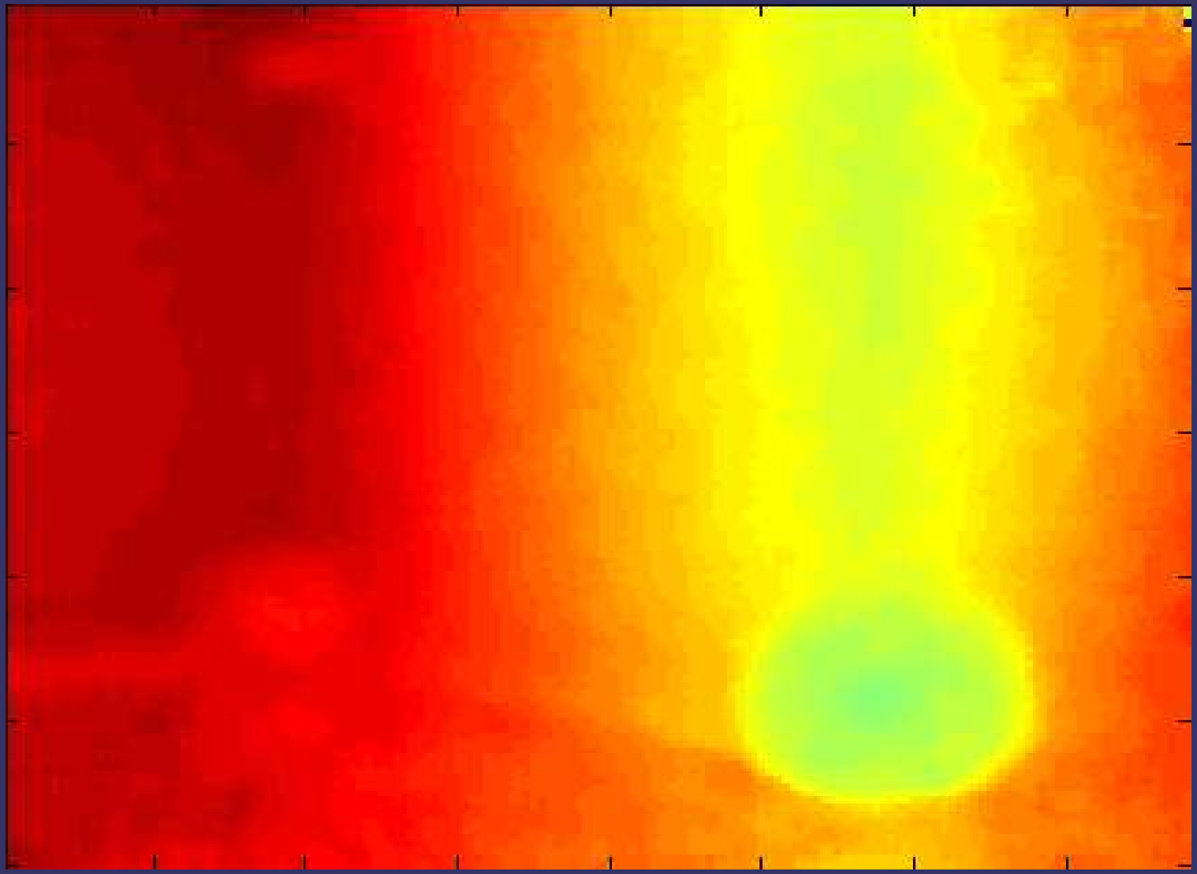
Reconstruction (1)



Original Scan

NOISY!

Reconstruction (2)

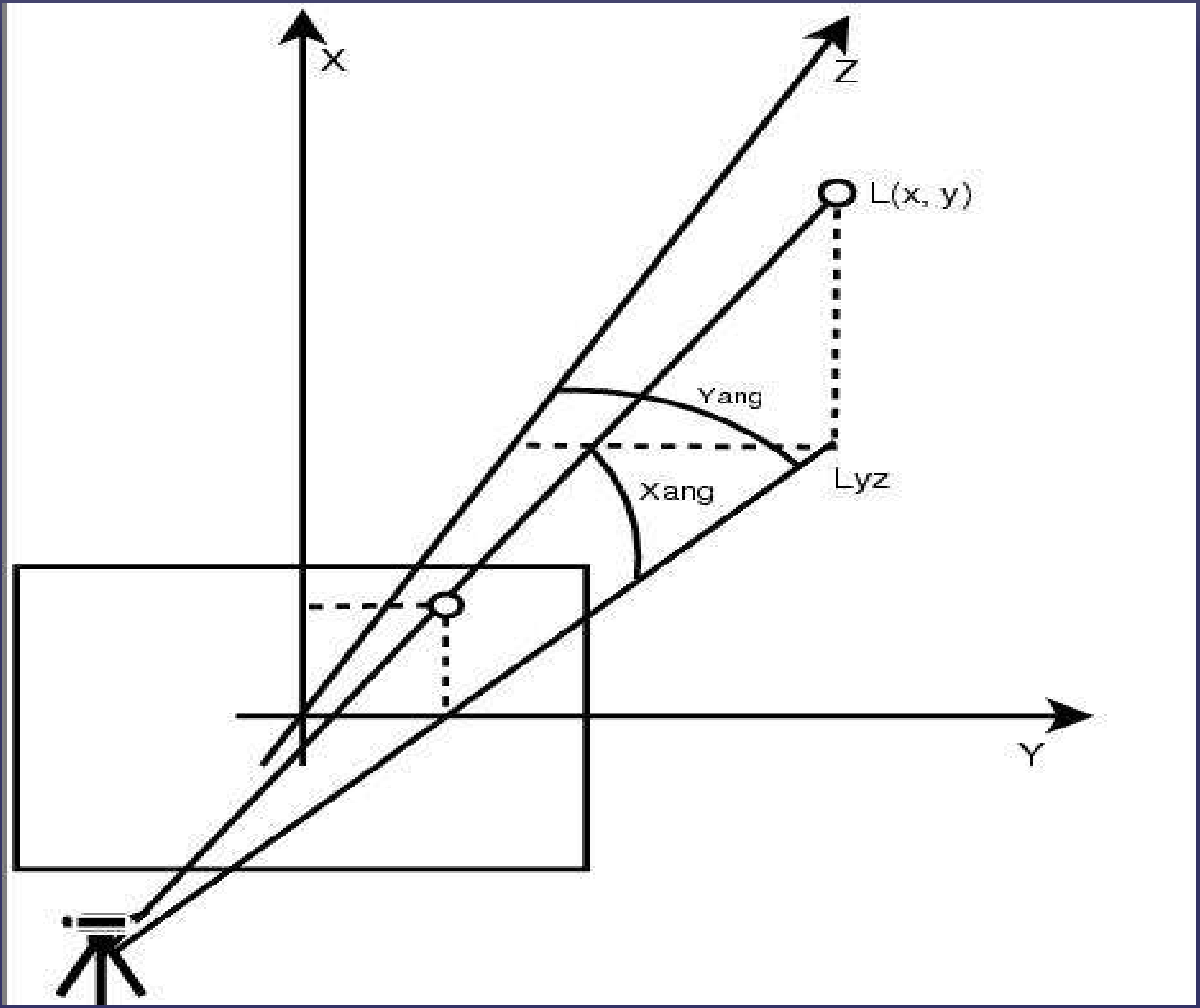


Mean filter makes it Beautiful!

Reconstruction (3)

- Each Pixel in mean filtered image represents a Time-of-flight (i.e. A distance)
- Can reconstruct to 3d points by using location of pixel in image and it's value L(x,y).

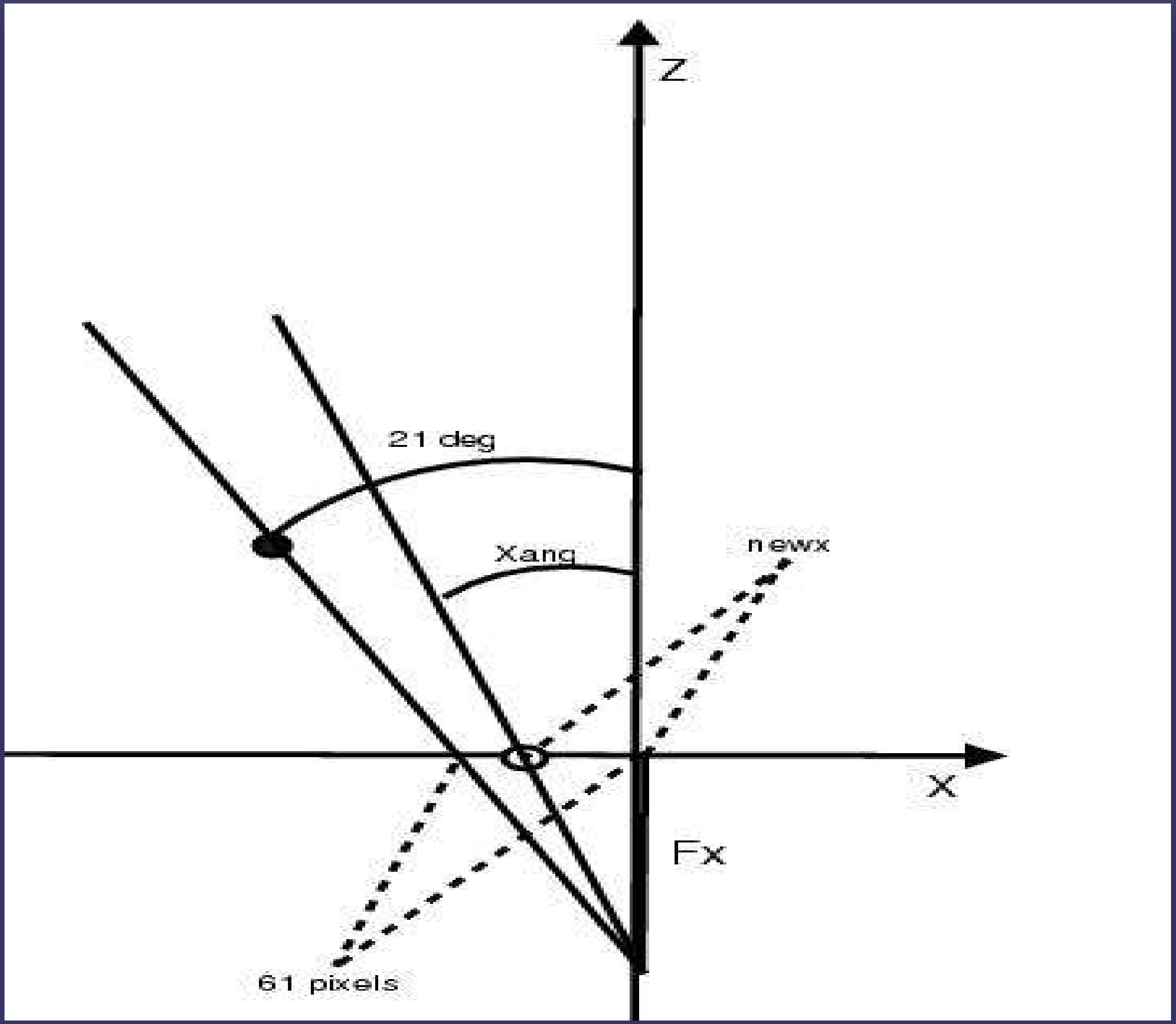
Reconstruction (4)



Reconstruction (5)

- We know that our sensor takes 124x160 pixel images
- We also know that the lens is +/- 21 degrees in the horizontal and +/- 23 degrees in the vertical direction.

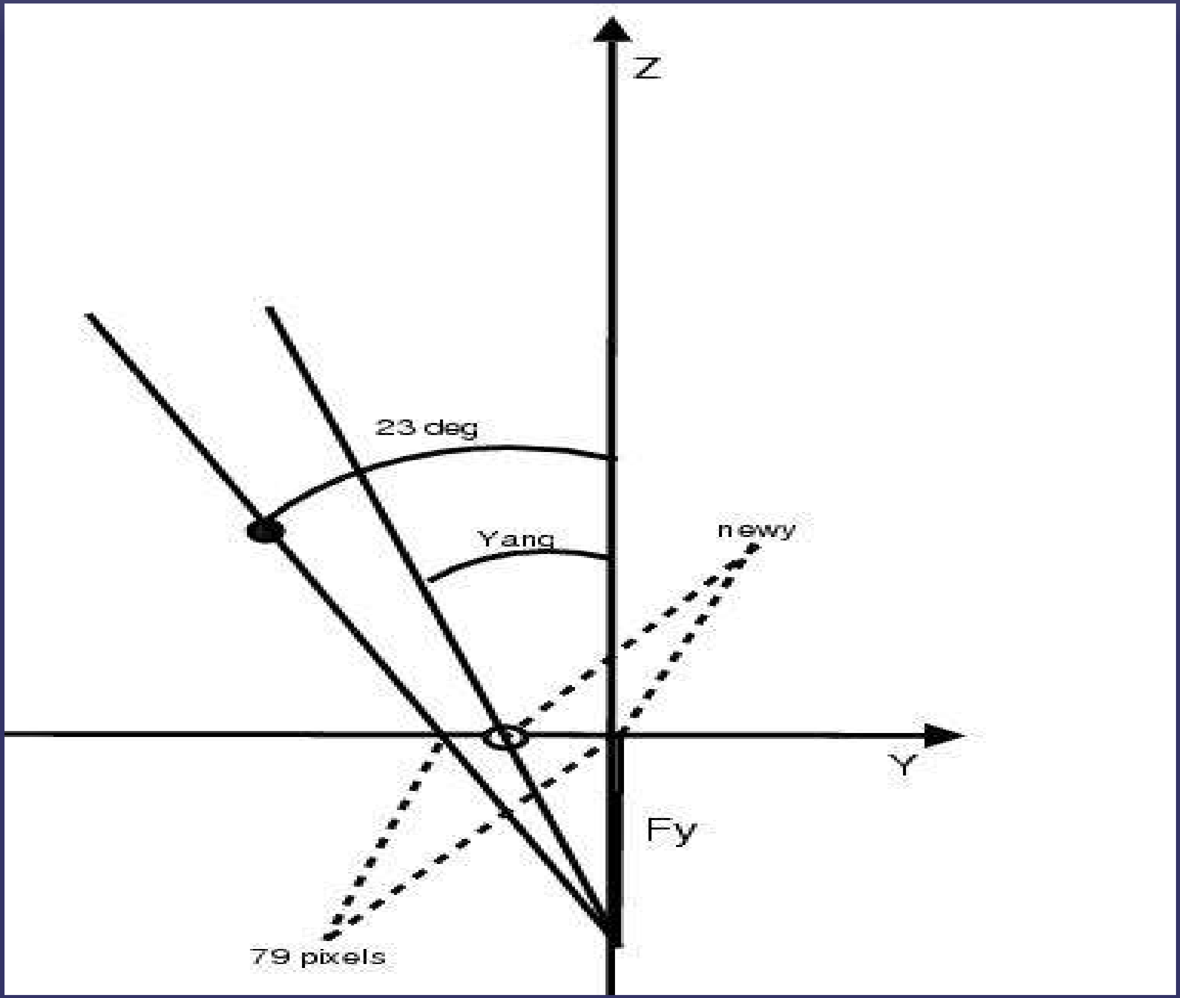
Reconstruction (7)



61/Fx = tan(21 deg)

Xang = arctan(newx/Fx)

Reconstruction (8)



79/Fx = tan(23 degrees)

Yang = arctan(newy/Fy)

Reconstruction (9)

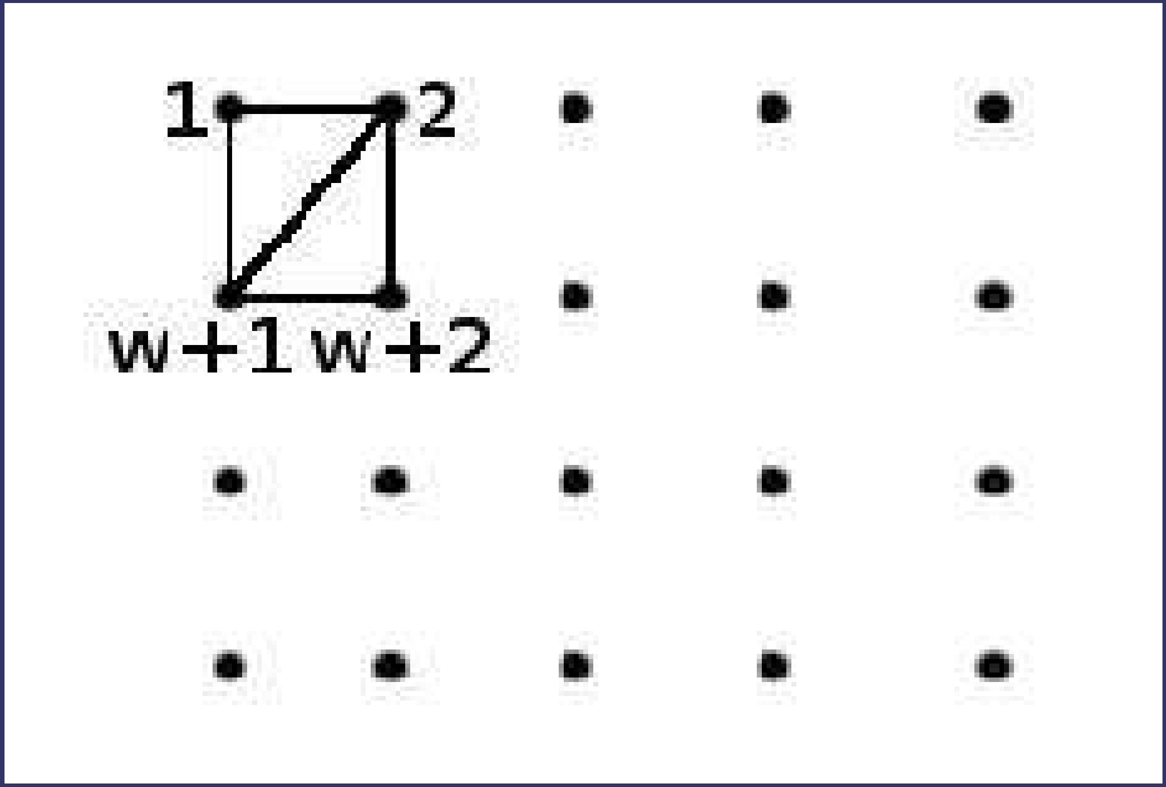
$$Lzy = \cos(Xang) * L$$
$$X = \sin(Xang) * L$$
$$Y = \sin(Yang) * L$$
$$Z = \cos(Yang) * Lzy$$

Our reconstructed point is then (X, Y, Z)

Reconstruction (10)

- We've got lots of points, but no relation to each other.
- Create a triangular mesh with faces!
- A face contains 3 points
- Labeled counter-clockwise

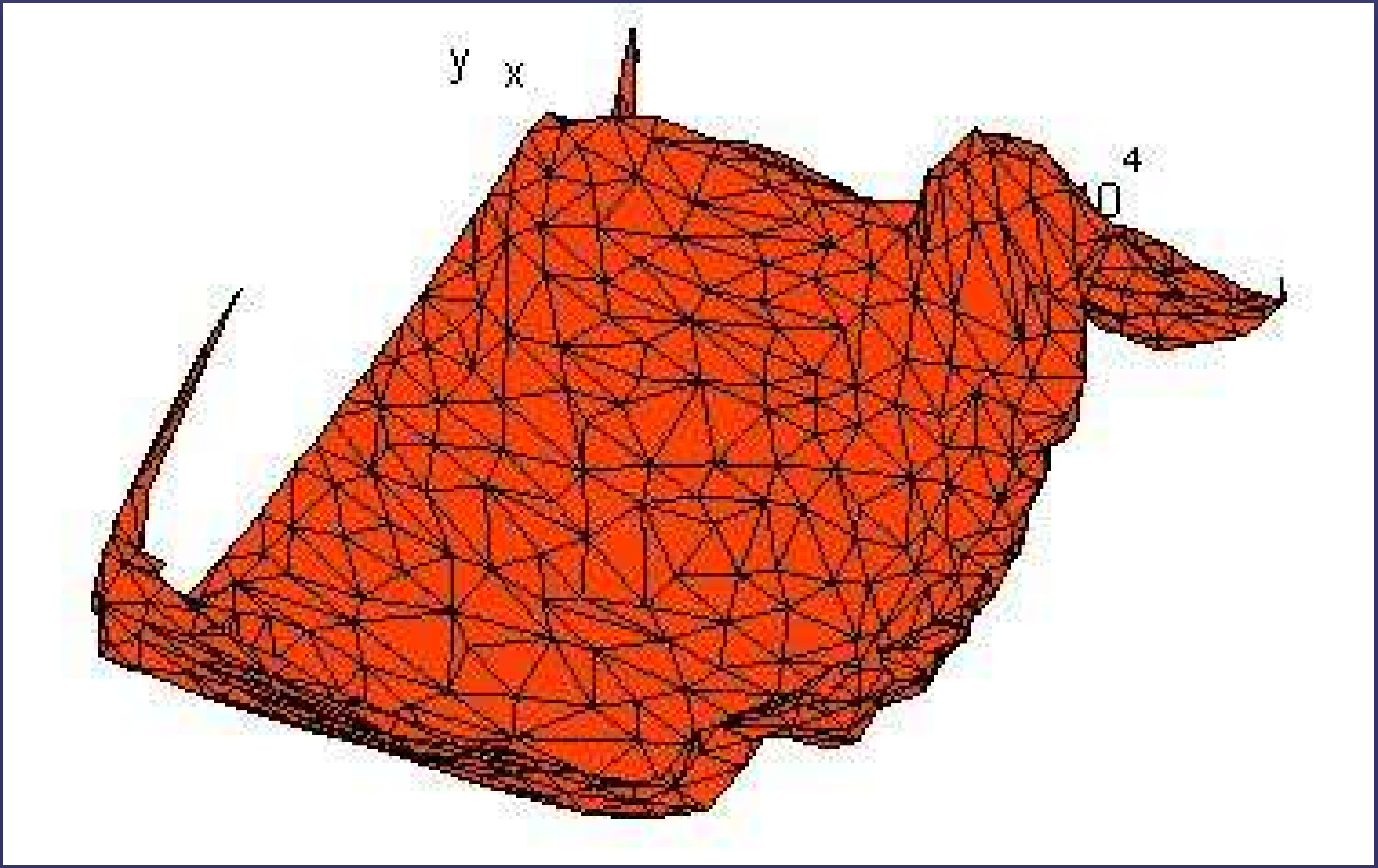
Reconstruction (Phewww... done)



Face 1 = (1, w+1, 2)
Face 2 = (2, w+1, w+2)

W = width of image

Wait... I lied.



(This was simplifed using a technique Arcana will discuss. It has 1000 faces)

Simplification (1)

- Why? We'd like to retain important information and forget about noise, and other unnecessary information.
- Compression!
- Can be simplified to dominant planes, representing surfaces

Simplification (2)

- Step 1: Compute surface normals of all faces. (cross product of pt2- pt1, pt3-pt2)
- Step 2: Compute the angle between all faces and both the XY, and XZ planes.
- Step 3: Create a 2D histogram and extract the peaks, which gives you a tuple of angles. These angles can then be used to compute a normal.
- Step 4: For each normal, define a plane with normal N and point (0, 0, 0)

Simplification (3)

- Step 5: For each plane: Compute distance of every point to the plane. Create a histogram and determine the distance that appears most frequently. Translate our plane to this distance away from the origin.
- Step 5a: Project some number (say 50) points to plane and eliminate them from consideration.
- Step 5b: Repeat til all planes are done

Simplification (4)

- Step 6: Compute intersection of all planes
- Step 7: For each point, find it's closest plane, and project the point to the closest plane if it is not beyond the intersection.
- This can be done using the half-plane test.
- Step 7b: If it is found to lay beyond the intersection, project to the line instead of the plane.

Simplification (4)

- Though this method works, it would be better to find the best fitting plane to the dataset.
- Expectation Maximization can do just that!

Expectation Maximization

- Finds the best fitting plane(s) to a set of data
- Determine the probability that a point fits to a given plane.
- Eliminates the need for our second histogram analysis (the distances)
- Step 1: Compute distances from each point to each plane.
- Step 2: Compute weights

Expectation Maximization (2)

- Step 2a: For each plane (A) compute the probability that each point (p) belongs.

  W = exp(-(D(p,A)^2 / 2))
- Step 2b: Normalize the weights
- Step 3: For every plane determine the origin by multiplying the weights * the points and normalizing by the sum of all the weights.
- Step 4: Translate X, Y, Z by the result of Step 3.

Expectation Maximization (3)

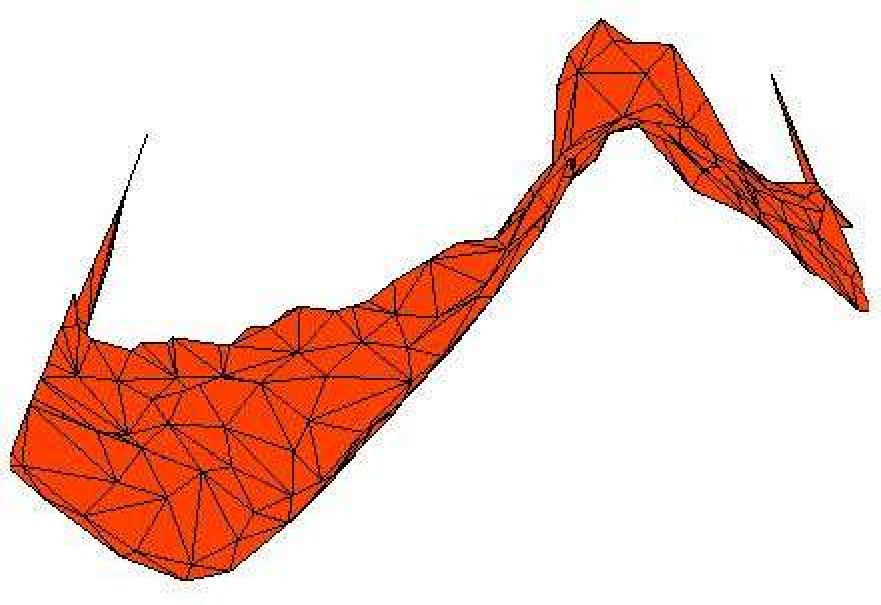
- Step 5: The new planes normal can be computed by solving the system...
- A*SUM x[i]^2 + B*SUM x[i]*y[i] + C*SUM x[i]*z[i] = -D*SUM x[i]
- A*SUM x[i]*y[i] + B*SUM y[i]^2 + C*SUM y[i]*z[i] = -D*SUM y[i]
- A*SUM x[i]*z[i] + B*SUM y[i]*z[i] + C*SUM z[i]^2 = -D*SUM z[i]
- This is easy in Matlab and the code for this after setting up the system is just....

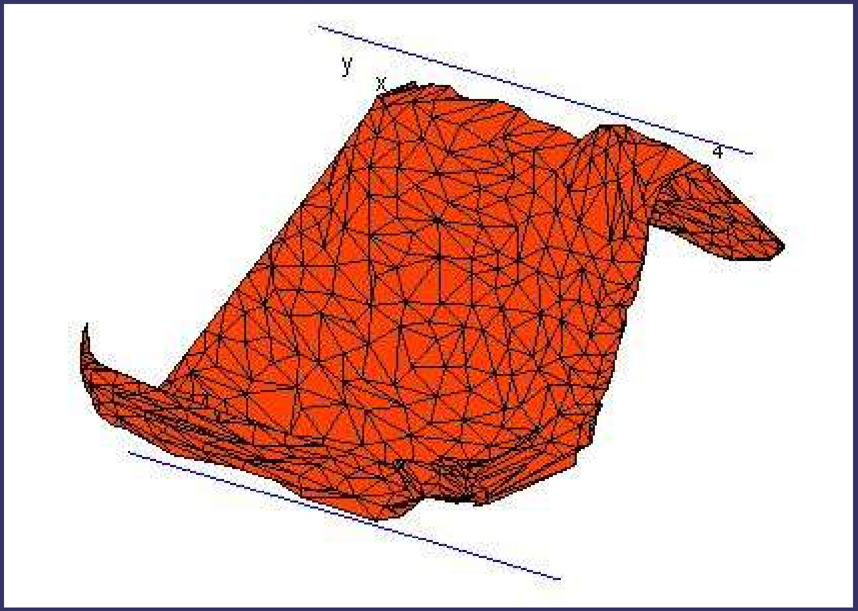
  M \ [-SUM(x[i]) -SUM(y[i]) -SUM(z[i])]'

Results (1)

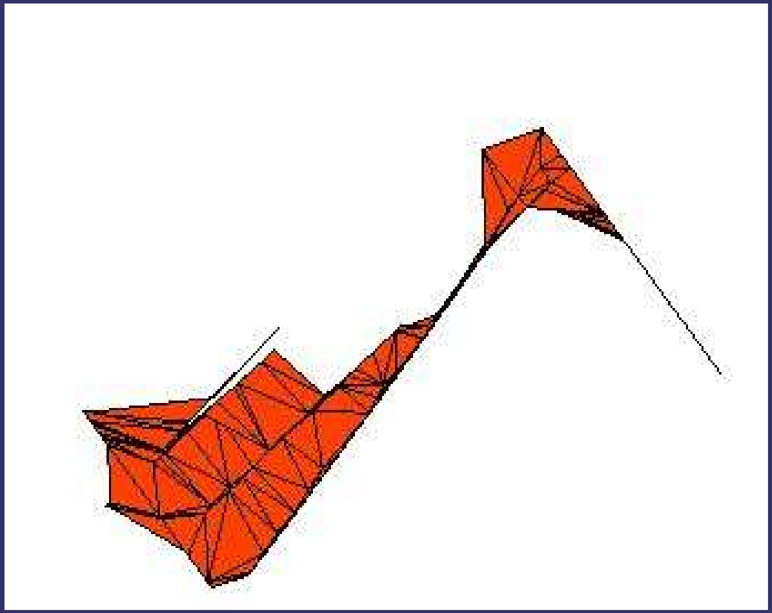
- Used Arcana's simplified data and Original data. (500 faces, 1000 faces, 36652 faces)

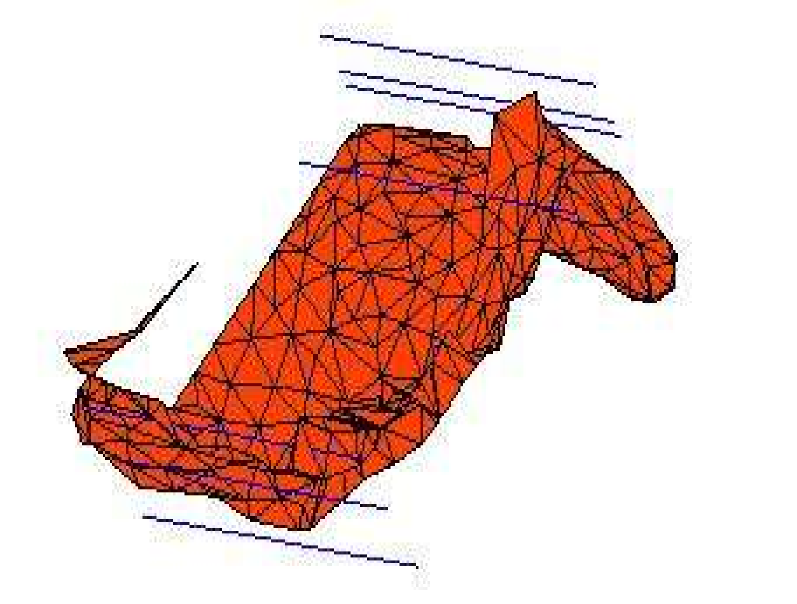
Results (2) 500-Faces





Results (3) 500 Points Simplified

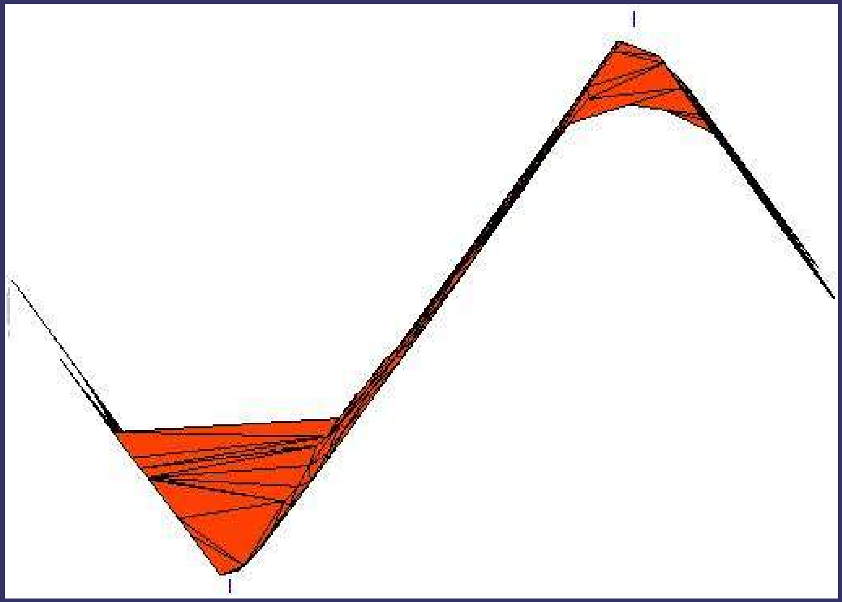


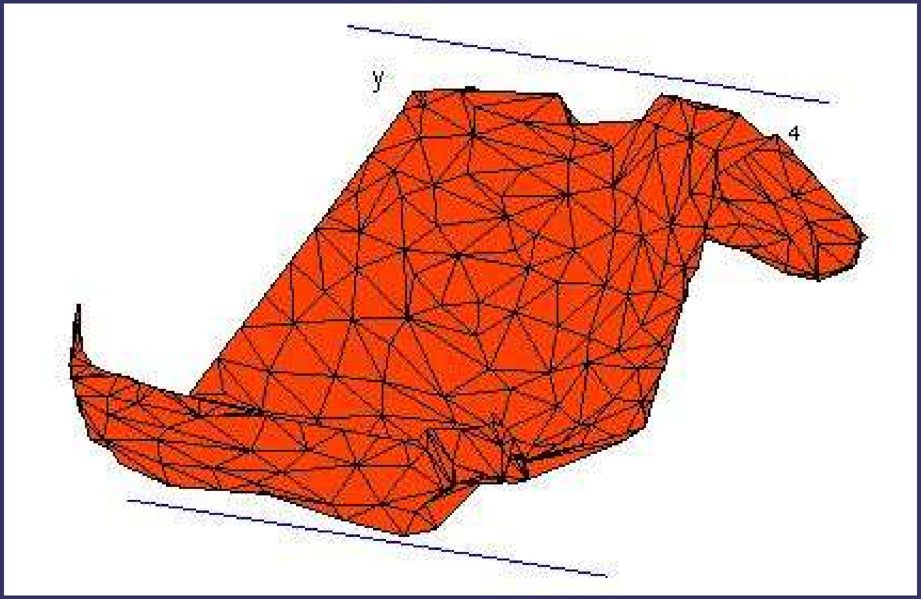


Results (3 continued) 500...

➲ So what happened? The histogram peak finding, found way to many normals and way too many distance peaks. This results in way too many planes for data to merge too, and therefore breaks the `niceness' of the mesh. The result: uglyness.

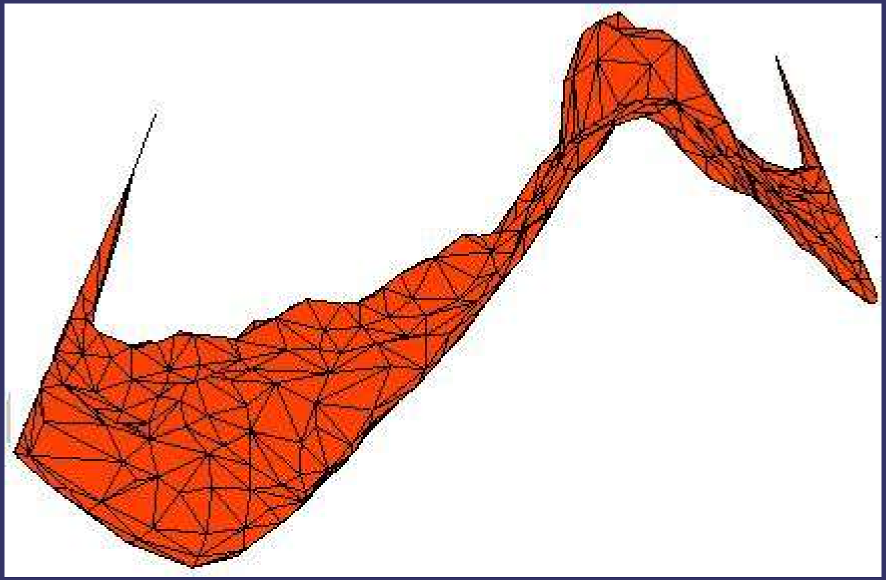
Results (3 cont... again)

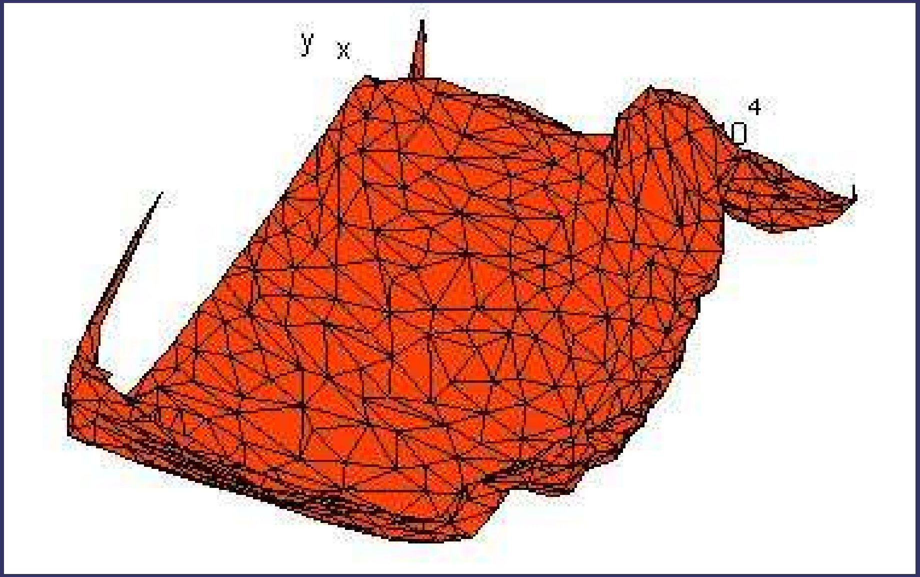




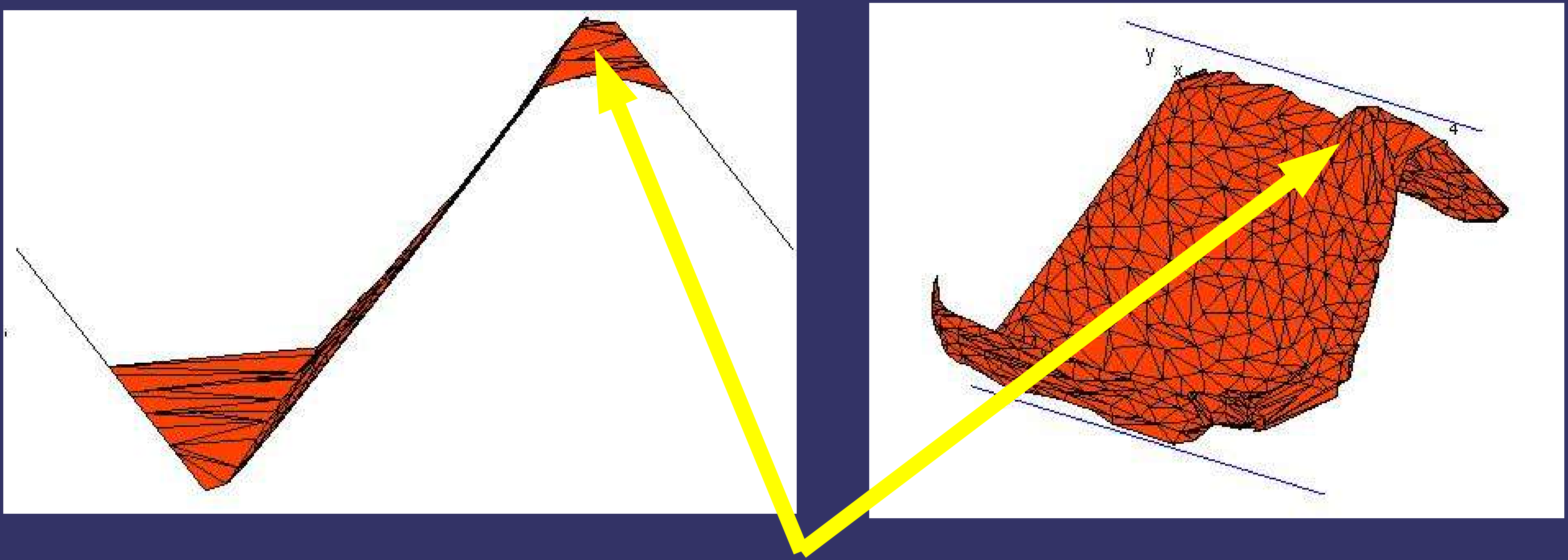
By hand selecting normals and distances, we simplify the mesh to something that looks much better and retains the information that we want. We'll continue with this approach for now.

Results (4) 1000 Faces



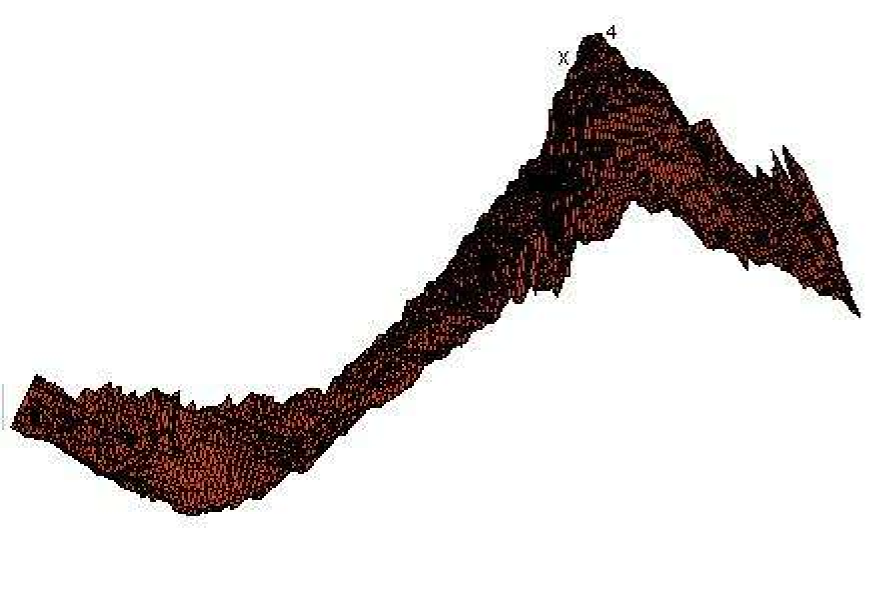


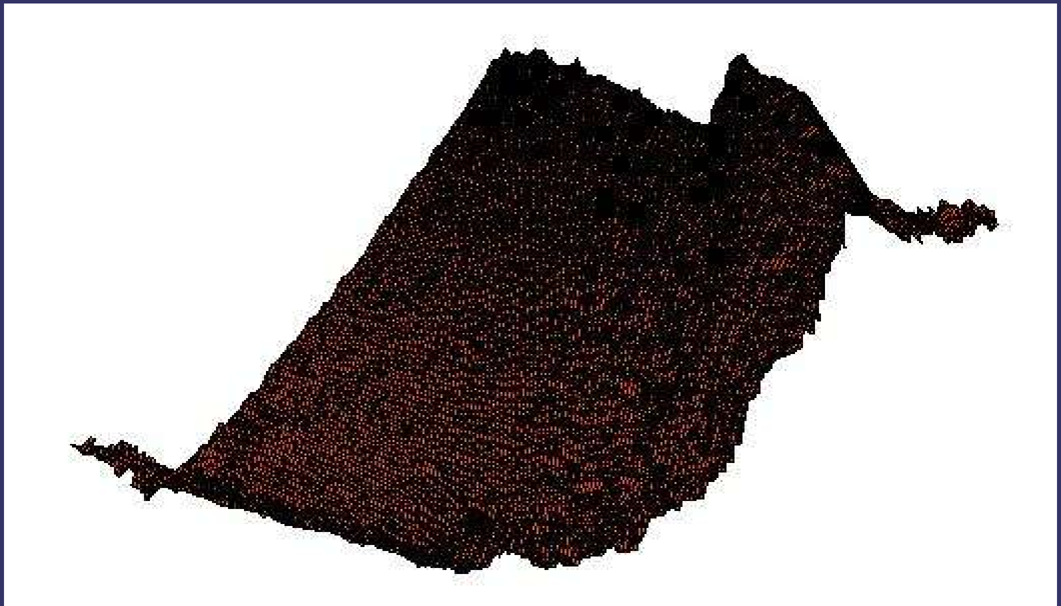
Results (4 cont)



The arrows point to where the ball is. On the view to the right it looks as though the ball retains it shape, but on closer inspection from a different angle, we see otherwise. What we see under the left arrow is stretching as a result of the mesh faces.

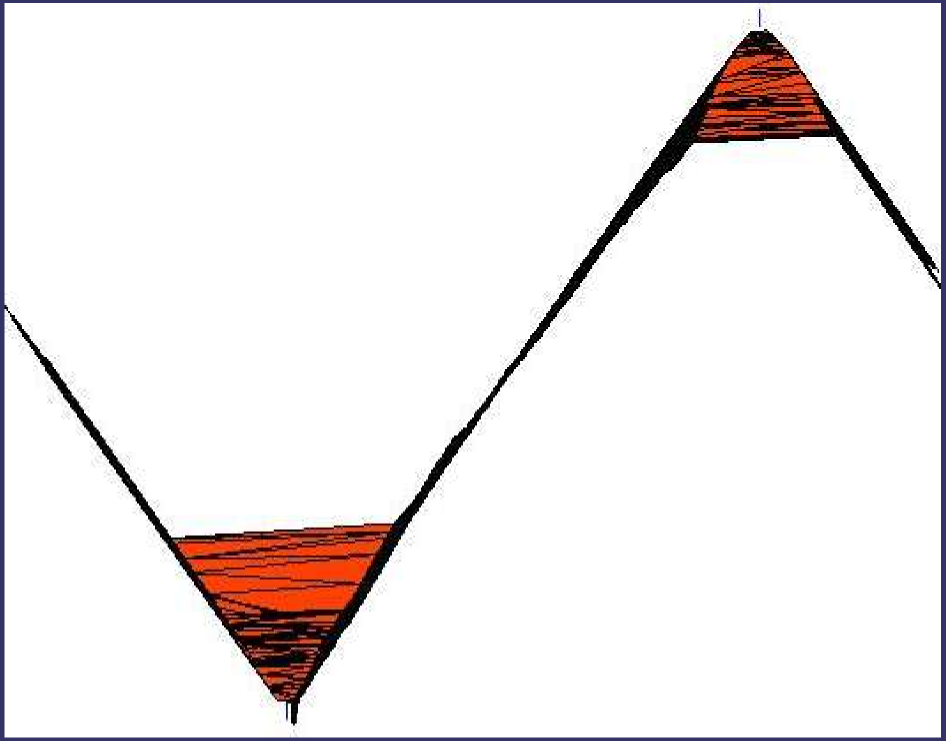
Results (5) Entire scan

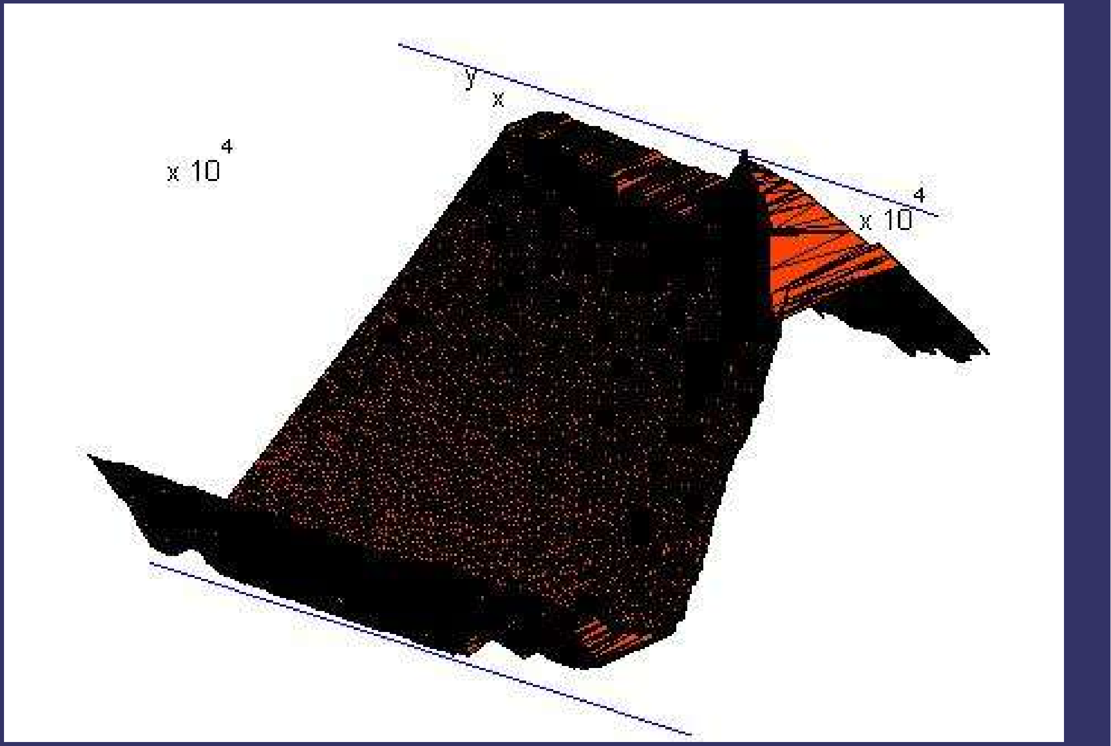




It looks extremely dark, since there are so many faces, and all the faces have lines around them

Results (5 cont) and....



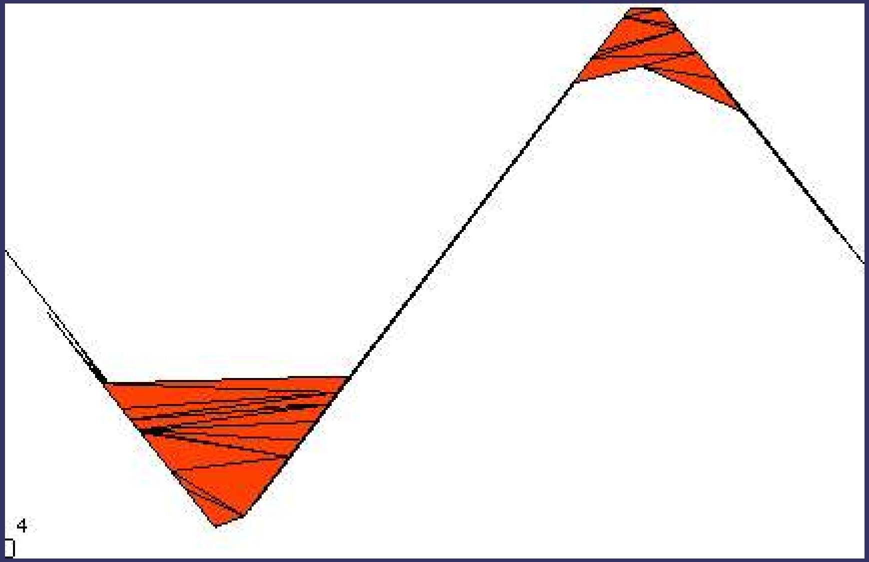


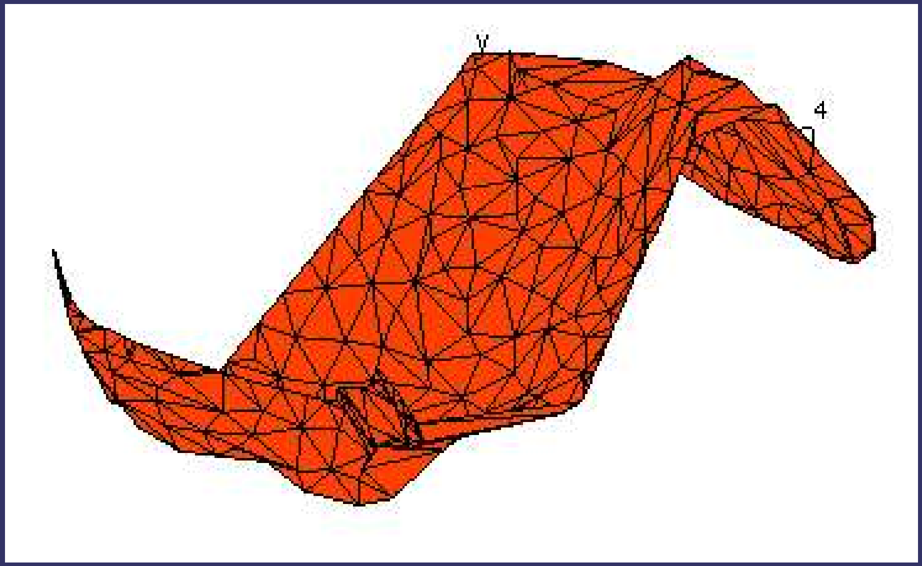
Quite similar to the 1000 faces... Does that mean we can simplify to 1000 faces before anything else?

Conclusions

- We see that the current techniques used to determine normals through histograms are flawed.
- By visually extracting normals and distances, we can however obtain desirable results.
- Since we have only a few planes after the simplification process, we can eliminate a lot of the points and compress the mesh by a large amount.

Results of EM for distance measure (500 Faces)





Results for EM – All the faces in the dataset....