Reconstruction and Simplification of 3D Laser Range Finder Data

Andrew Gwozdziewycz
Temple University
CIS 601: Image Processing

December 13, 2004

Abstract

Major problems exist in image recognition because of the simple fact that a computer cannot determine how far the subject in an image is from the camera. Recent advances in sensor technology have eliminated this setback by measuring the time of flight to objects in a picture. Though the sensor has its problems (most notably its resolution), it has created the need for algorithms for extracting data from this new kind of image. This paper presents a solution for creating a simplified mesh of the sensor's data, and presents some additional ideas and uses for this data.

Introduction

Recent advances in sensor technology, have made available a camera which measures time of flight to determine the distances of objects in an image. This type of sensor is what is needed to solve many of the problems in the field of image processing, such as segmentation of images and object recognition, since these problems rely on many more characteristics of an a picture than color or texture.

This sensor has many other uses besides image segmentation and object recognition. One such use is in robotic mapping for rescue robots. Rescue robotics is a field in robotics which aims at creating robots that can spare the lives of humans by entering dangerous situtations, such as building collapses, to survey the area and determine whether or not victims are present. Should a victim be present, rescuers can determine exactly where the victim lies by using the information gathered by the robot. Perhaps the most useful information is a map of the robots surroundings. Combined with photos, the rescuers can determine other dangers within the rubble.

In this paper I present a solution to creating a 3-dimensional representation of the time of flight sensor's data, and a simplification procedure which both compresses the data (by elimination of redundant data), and finds features such as walls.

Reconstruction

The data produced by the sensor is a matrix of time of flight measures. It current has a resolution of 160x124, and uses a lens with a field of view of +/- 21 degrees horizontally, and +/- 23 degrees in the vertical direction. Using these constants and the time of flight measure we can determine the actual distance with basic triginometry. This prior knowledge is required for proper reconstruction.

The first step of construction is translating the center of the data to the origin (0,0). Once centered, we can compute the angle between the center of the lens and the data point (x, y), by using the distance Fx (the distance between the center of the lens and the center of the CCD chip), and our predefined knowledge (lens properties). [See Figures 1 and 2] The X angle (Xang), can be determined by:

$$\frac{61}{Fx} = \tan(21 \deg)$$

$$Xang = \arctan(\frac{newx}{Fx})$$

Similarly, the Y angle (Yang) can be determined:

$$\frac{79}{Fx} = \tan(23\deg)$$

$$Yang = \arctan(\frac{newy}{Fy})$$

Finally, we can get our reconstructed point using the following (Figure: 3:

$$Lzy = \cos(Xang) * L$$

$$X = \sin(Xang) * L$$

$$Y = \sin(Yang) * L$$

$$Z = \cos(Yang) * Lzy$$

Once all of our points are reconstructed, it is quite useful to create a triangular mesh from them for visualization. This can be done by first vectorizing the matrix of points into a single dimension. Assuming our points are kept in order, it is possible to define mesh faces using the indices of the points in the vector. To keep our mesh consistant we use a counter-clockwise numbering scheme, illustrated in Figure 4.

Simplification and Feature Extraction

Current mesh simplification techniques work very well for keeping fine details in a mesh and eliminating the non-feature vertices. We, however would like to simplify the mesh in such a way that we can see walls, or large obstructions such as giant pieces of rubble and not details such as tiny pebbles. For this reason a new technique, which is capable of extreme compression, has been created.

The data the sensor gives us is somewhat noisy. To reduce the level of noise we apply a 5x5 mean filter. The original data and then mean filtered data are pictured in Figures 5(a) and (b).

Since our main goal is robotic mapping, the technique described tries to find features such as walls or large obstructions which make sense to provide in a map of an area. This is established by first determining all of the angles between the surface normals of the faces and the XY and XZ planes. We then compute a 2D histogram of these angles and find peaks. According to the number of peaks found in this histogram, a number of normals are extracted, which correspond to possible flat surfaces in the mesh, whose normals are pointing in multiple different directions.

The positioning of these surfaces is not known, since the angles of these surfaces do not rely on distances, but they can be estimated by assuming that these surfaces lie on the origin (0, 0, 0). For each potential surface, we then compute the distance to every point using standard Point-Plane distance. Here, the plane is defined by our surface normal, and the point (0,0,0). A histogram is computed and peaks are found in it. The coore-sponding value of the maximum peak can then be assumed to be the distance from the origin, to a plane in the reconstructed data defined by the surface normal.

The process is repeated, eliminating the closest points to the computed planes each time. To be safe, we eliminate only the K closest points to the current plane so that their weight isn't used to place another plane (K = 50 has worked well).

The intersections of all the newly found planes are computed and saved in order to determine where we should project our points. It may be that some points lay closer to the line than to the real plane, or are outliers that do not lay on the plane at all but contribute to the structure in some other way.

We then project each point to it's closest plane or line and have a simplified mesh representing the main structure of the mesh. The faces computed before are still valid, since we did not eliminate any points. At this point the mesh could be simplified to only a few vertices per surface and the faces recomputed to reflect this.

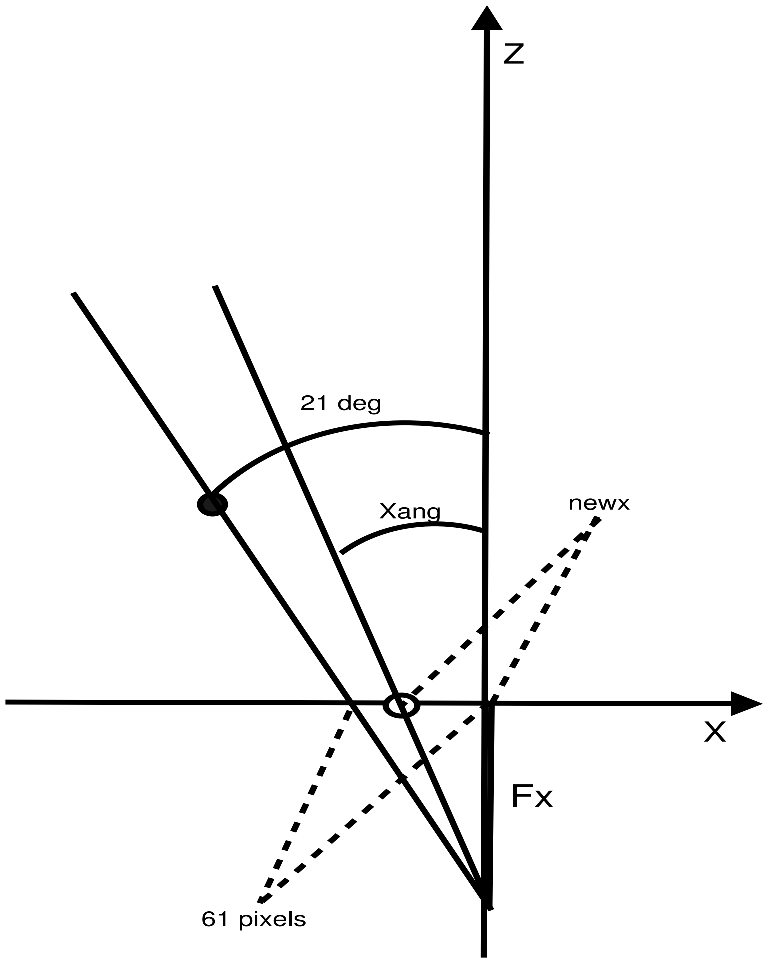


Figure 1: Determining the X-angle

Results

To test the process we used 3 different meshes, with face counts, 500, 1000 and 36652, which represents our reconstructed mesh. Simplification to 500 and 1000 was done on the 36652 face mesh. Not only did this save computing time, it also resulted in similar results to the full mesh.

The output of the sensor is the result of scanning a 3 walls set up in a zig-zag. On the right hand outfacing corner, a ball is positioned and the left side inward corner another ball is also positioned. The scan shows the right ball quite well, but the left ball blends in with the floor. Figure 5 shows this.

In the first 3 cases we come out with a similar structure, which represents the 3 walls. It can also be seen that the balls structure is destroyed, but remains as an extension to the surfaces found. Recovering these surfaces was possible because we used advanced knowledge.

In the final case (Figure: 12, histograms were analyzed to compute the surface normals and distances. This resulted in 5 planes being defined for the 500 face mesh. We feel that the result of this is not better than with predefined knowledge, however the results do look promising.

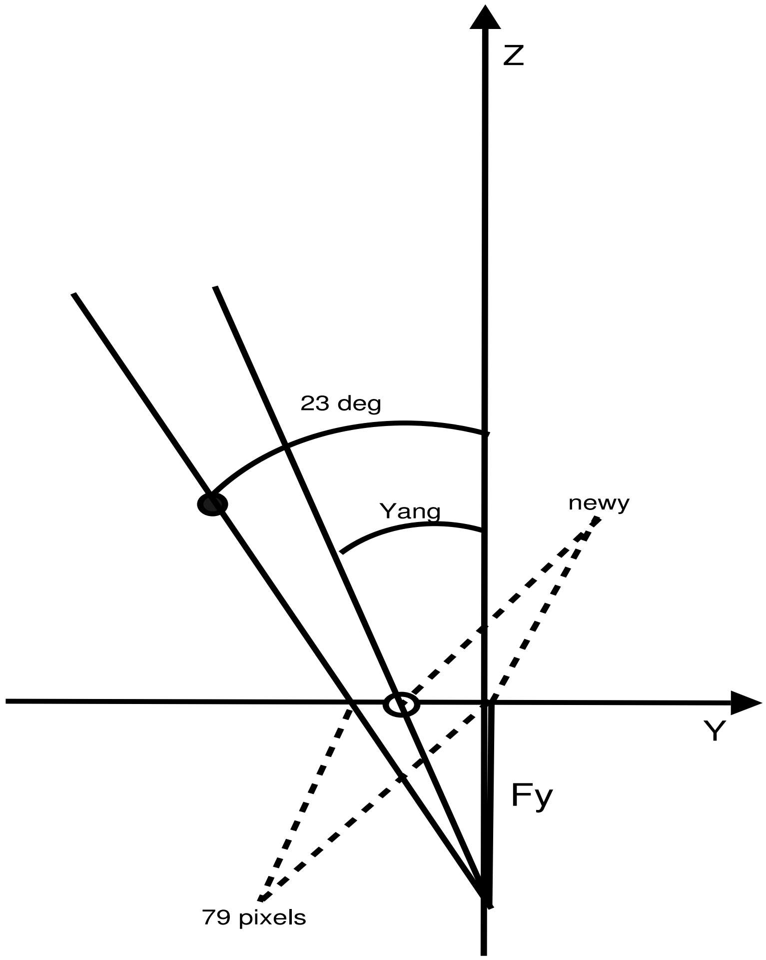


Figure 2: Determining the Y-angle

Conclusions

Since we had prior knowledge of what our scan was, we were able to target simplification to 3 structures. We can see that this method does work quite well since we recover 3 flat walls. Choosing surface normals automatically using a 2 dimensional histogram of angles leads to results which are not far off, but a better histogram analysis method would narrow down the count of normals extracted and therefore the number of surfaces found. This simplification is ideal for a robot to both process in the future, as it can be simplified to just a few vertices and faces, and because it keeps only the most important information about the scan (the fact that there are 3 walls).

Future Work

The method for selection of surface normals from the 2 dimensional histogram of angles is too ambiguous. It relies on parameters that are scan independent, and multiple peaks can occur that reflect semi-redundant information (as seen in Figure ??). The distance histogram analysis provides good results that fit, but other algorithms are being researched. One that looks promising is to use expectation maximization to place the plane optimally in a set of points.

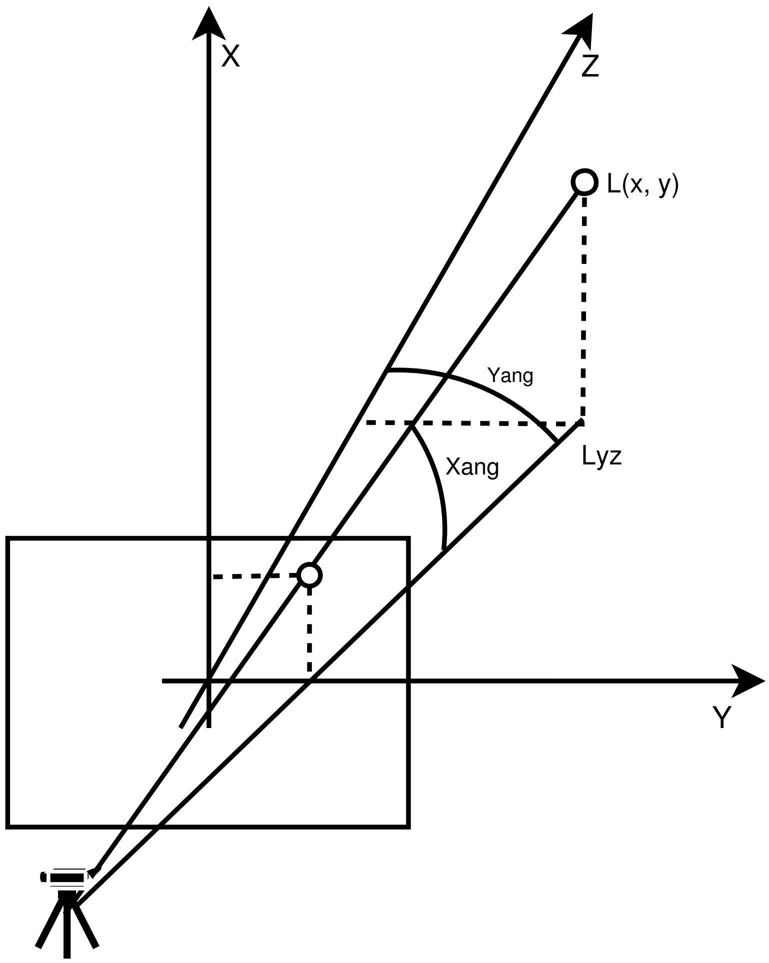


Figure 3: Determining the XYZ point reconstructed.

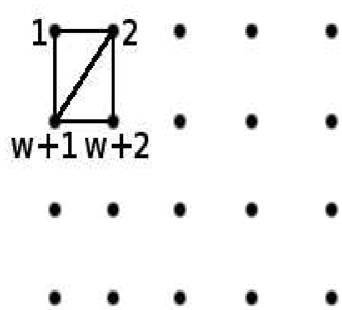
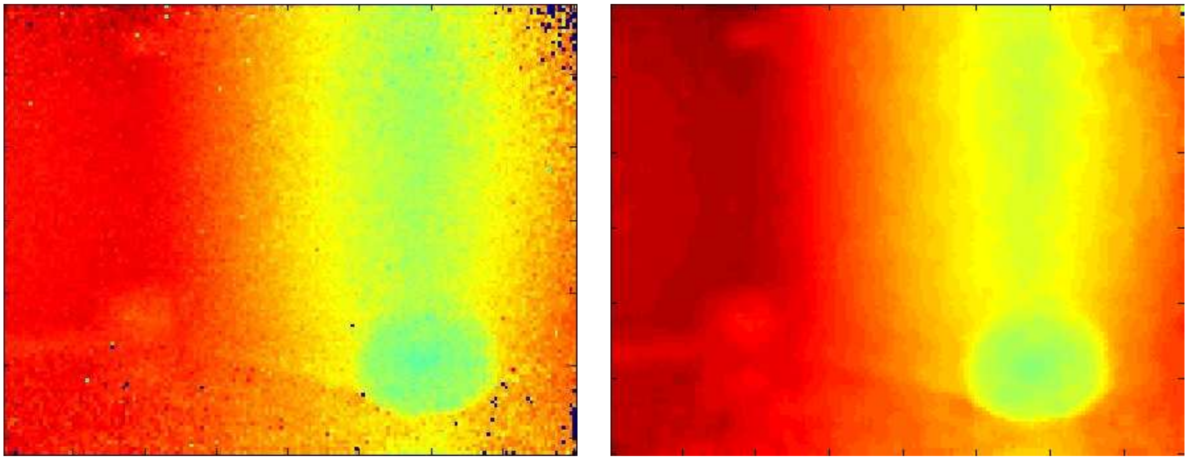
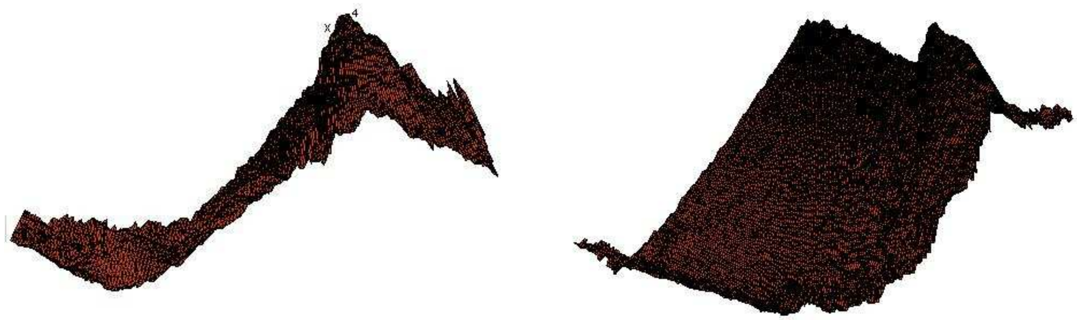


Figure 4: w is the width of the image. Two faces are defined here. Face 1 is defined by the indices 1, w+1, 2 and face 2 by 2, w+1, w+2.



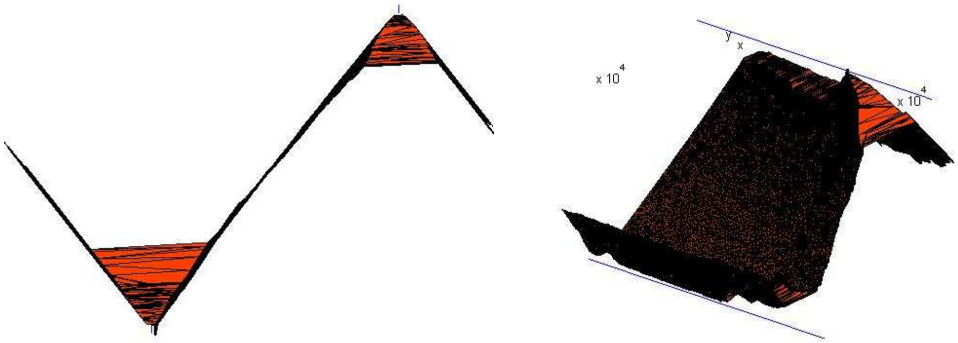
(a) (b)

Figure 5: (a) Original scan (b) Mean filtered scan



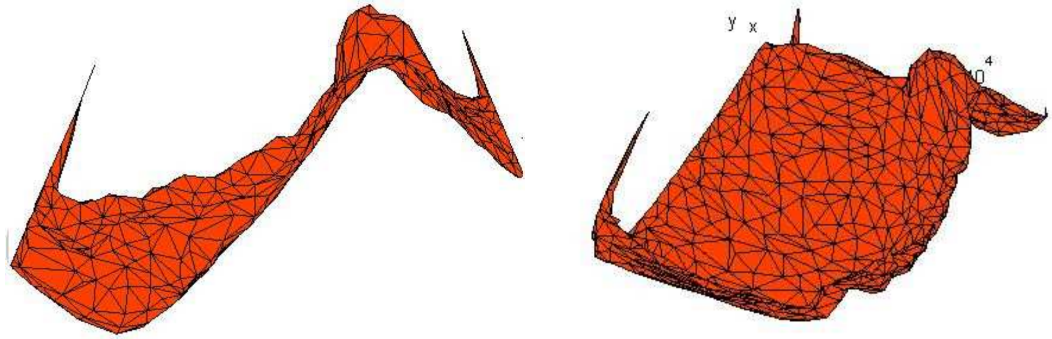
(a) (b)

Figure 6: (a) Original mesh bottom view (b) Original mesh sideways view



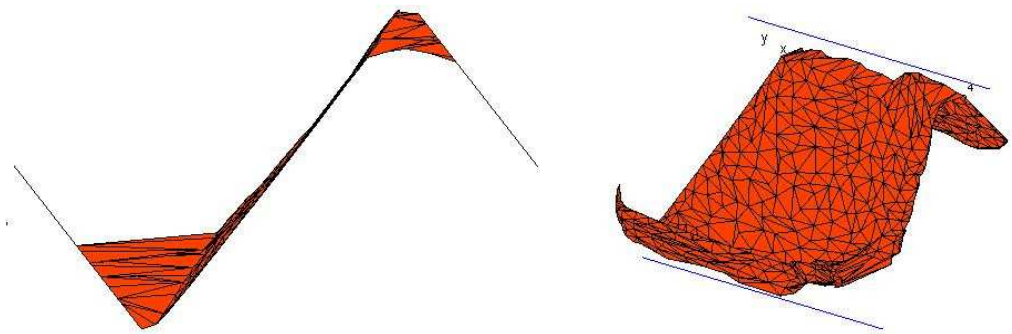
(c) (d)

Figure 7: (c) Simplified mesh bottom view (d) Simplified mesh sideways view



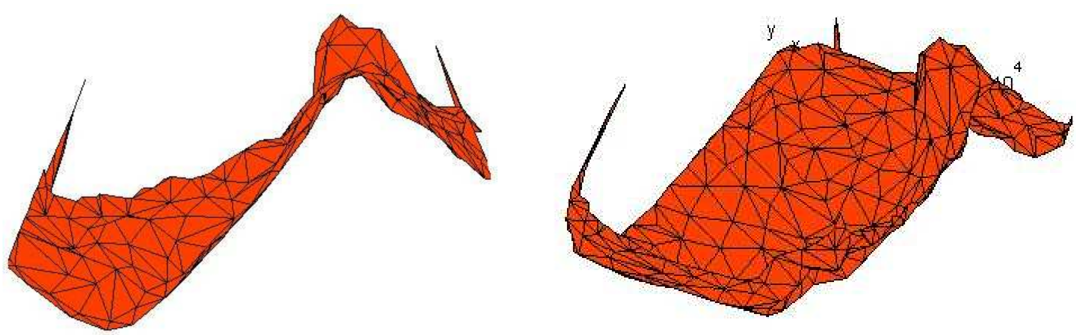
(e) (f)

Figure 8: (e) Original mesh 1000 faces bottom view (f) Original mesh 1000 faces sideways view



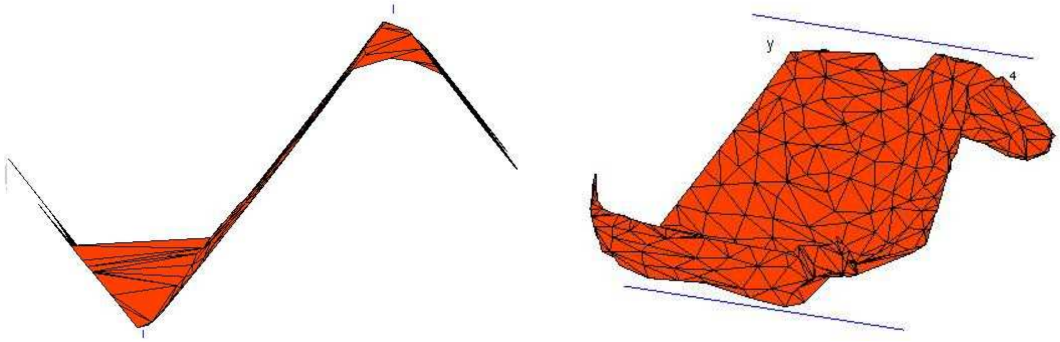
(g) (h)

Figure 9: (g) Simplified mesh 1000 faces bottom view (h) Simplified mesh 1000 faces sideways view



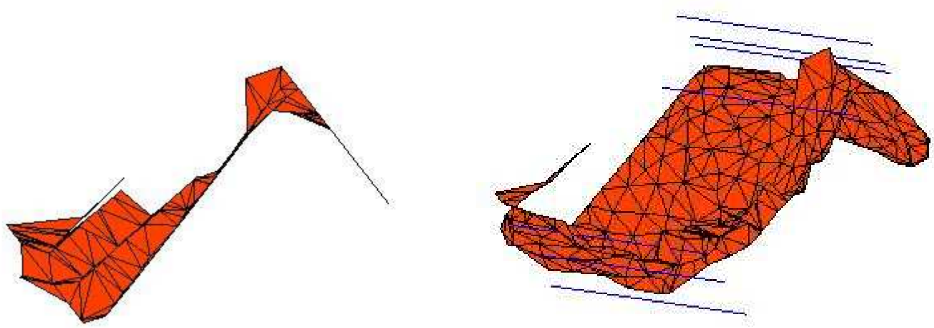
(i) (j)

Figure 10: (i) Original mesh 500 faces bottom view (j) Original mesh 500 faces sideways view



(k) (l)

Figure 11: (k) Simplified mesh 500 faces bottom view (l) Simplified mesh 500 faces sideways view



(m) (n)

Figure 12: (k) Simplified mesh 500 faces bottom view (l) Simplified mesh 500 faces sideways view