CS 554 Homework 4

Shubhomoy Das
Department of Computer Science
Oregon State University
Corvallis, OR 97331
dassh@eecs.oregonstate.edu

1 Curvature Estimation

Curvature visualization on Feline are shown in Figure 1. The mean curvature values are more contrasting and hence it might have a larger influence in smoothing. The colors are coded from green to red. The deeper shades of green are larger than lighter shades of green and shades of red are larger than shades of green.

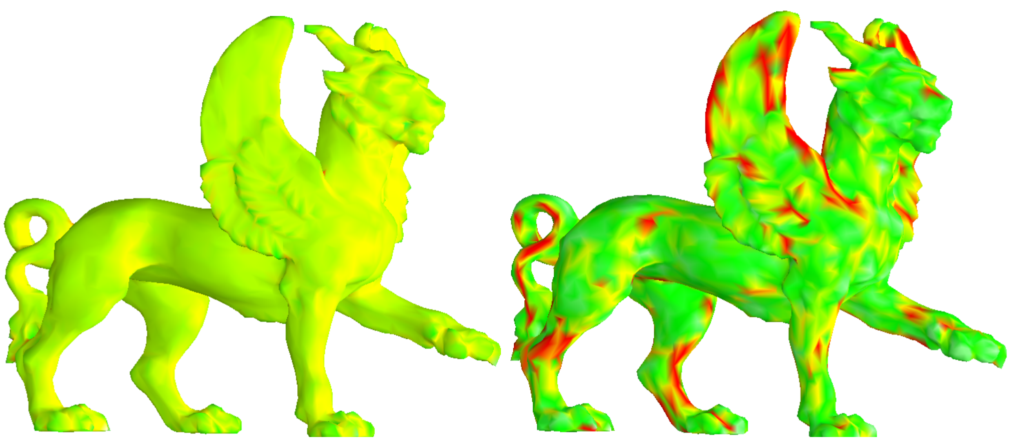

Figure 1: Curvature Estimation: Gaussian (Left), Mean (Right)

Figure 2 shows the curvature tensors on *happy*. I smoothed the model for 20 iterations before computing the curvature tensors and then smoothed the tensors for 100 iterations. After smoothing, I do not see much difference between uniform and cord. There was no perceptible difference even after 10 iterations. But I think mean curvature smoothing is better than both.

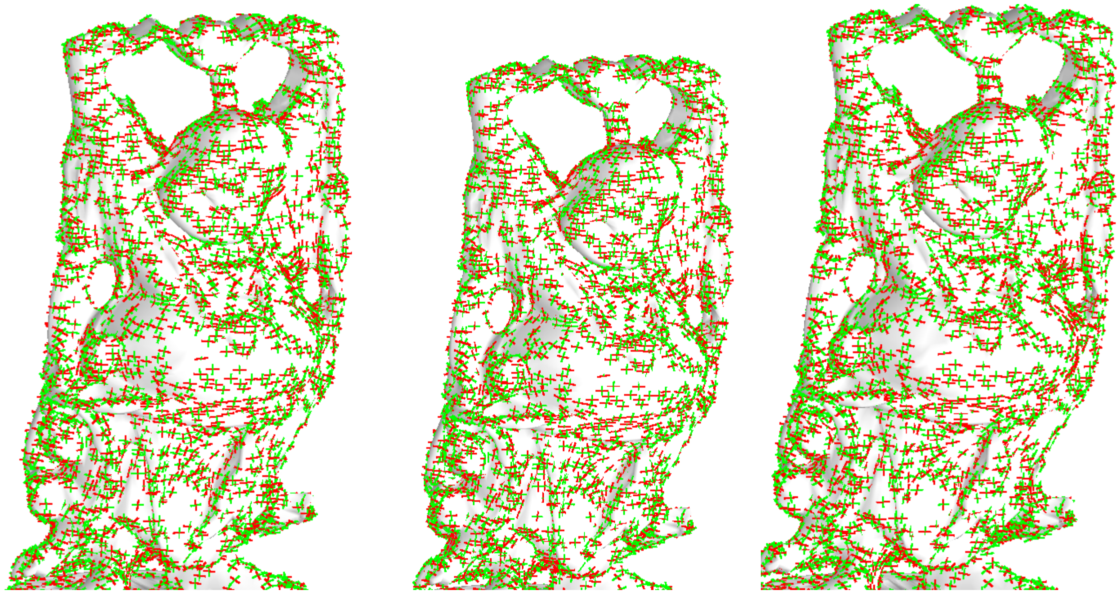

Figure 2: Smoothing Curvature Tensors on Happy: Uniform (Left), Cord (Middle), Mean Curv (Right)

1.1 Euler Characteristic

When I compute the total angle deficit and divide by 2 pi, then the value I get is exactly equal to the Euler Characteristic of the model. Assume that we start with a model with smooth surface having a particular number of 'tunnels'. This model has a particular Euler Characteristic which is an intrinsic property of the surface. Now, any deformation is going to add new angles at some places, but due to topological restrictions, it will need to subtract the angles from other places. This operation self-negates and hence, the Euler characteristic does not change.

1.2 Assessment of Curvature Estimation Algorithm

Smoothing the tensors helped greatly. I do have some discrepancies in determining the major/minor axis. I believe somewhere the curvature magnitude is being incorrectly computed. However, the principal curvature directions look good. Maybe a better approximation of the curvatures using the more sophisticated mean and Gaussian curvatures will help.

2 Pen and Ink Sketch

I computed the curvature tensors in the global co-ordinates for all vertices and smoothed them. Next, I averaged the individual components of the 2 by 2 tensor matrix across three vertices of a triangular face. This corresponds to the tensor at the center of the triangle. Finally, I transformed the tensor to the local frame for the triangle centroid. The principal components of this final tensor matrix are plotted for the pen and ink sketches. Some samples are shown in Figure 3.

The algorithm has got confused between the major and minor axis in some places but the two principal curvatures have been correctly identified.

Figure 4 shows a nice application of the curvature tensors in generating pen-and-ink sketches. I compute the dot product of the normal with the light source to get cut-off thresholds. On the basis of this, I skip plotting any curvature if the light threshold is above 0.95, plot only the major axis when the light threshold is between 0.4 and 0.95, and plot both major and minor axis if the light is below 0.4.

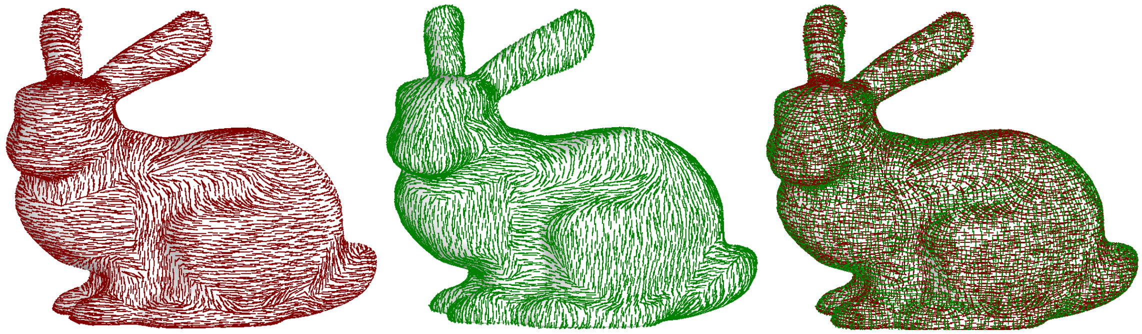

Figure 3: Bunny Pen-Sketch: Major (Left), Minor (Middle), Superimposed (Right)

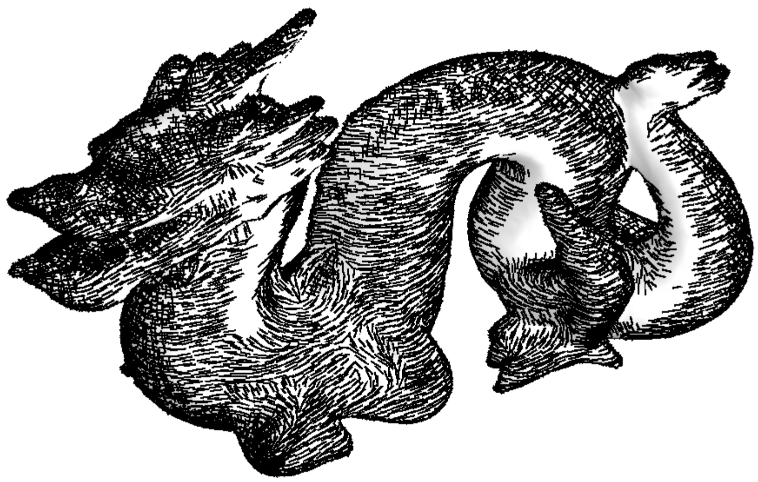

Figure 4: Dragon Pen-sketch with Shades

3 Conclusion

All the code has been uploaded to the TEACH website. This code has been compiled and tested with Visual Studio 2010. The accompanying file Run.txt (in the codebase) lists the commands that need to be run for each part of the assignment.

References

[1] *Polygon Mesh Processing* by Botsch. M., et. al.

[2] *http://www.cs.helsinki.fi/u/tvvlappi/muuta/Numerical_Recipes/bookc/c11-1.ps*