Global Histogram Project Report

Mario A. Gutierrez
mag262@txstate.edu

The purpose of this paper is to present the adventure the author followed in the implementation of the global shape histogram model used in the Princeton Shape Benchmark. The authors successes and failures are reviewed leading up to the current state of the project.

The task began around the second week of June of 2014 and lasted throughout the length of the program. The initial implementation was completed after the first week and some tests were performed to make sure it was doing what it should. At this point, the program only had the single mode of comparing a single model against a generated histogram directory. To get a quantitative measure of performance, however, a distance matrix had to be created.

The program didn't produce good scores. In fact, they were quite terrible. So, the next week was spent re-testing every single bit of the code, producing visualizations of the output wherever was possible (by putting the output to a file and using the 'rgl' library in R to plot the 3D data). There seemed to be no errors, so it definitely had to be something to do with *what* was being implemented.

Over the weekend, a new polyhedron model was used; for reason that it was the one used in the paper originally using this spherical histogram method for shape retrieval [2]. This produced worse results it seemed, so throughout the next week, some other things were tried. The spherical coordinate model was brought back, this time, with a vertical axis; otherwise, there is a cone of empty space at the north and south poles of the sphere. This was an improvement over the previous spherical coordinate model. After that, the bounding sphere method was changed to use a quick bounding sphere approximation algorithm by Jack Ritter (as implemented in one of the TriMesh library versions). This improved performance over a naive method of just using the center of mass.

One of the big problems up till that point was that the analysis of performance was only being done manually, by comparing a few shapes against the PSB database. The distance matrix method was implemented, but the program used to analyze the matrix wasn't working on any computer that was convenient. After getting the source files it was finally recompiled on a convenient computer.

When the second distance matrix was computed and analyzed, a couple speed optimizations were made to the global histogram project code and the bounding sphere method was changed to just use the bounding box; Ritter's algorithm enclosed the model, but not perfectly, and not always in the center. The changes made had good effect, the new scores were much higher and closer to the scores in the PSB paper.

The method of setting parameters was changed to allow easier testing of different parameters, and the program code was cleaned up and polished a bit. During this testing and tooling around it was also found that a perfect bounding sphere algorithm (miniball) lowered the performance. After about two days of testing, a pattern was found where the performance seems to increase with higher dimensionality. It was also found, strangely, that not performing any transformations of the model points (aside from shifting it to bounding box center) improved performance. Furthermore, the highest scoring model, in the end, turned out to be the polyhedron-based model.

Unfortunately, the scores never quite reached the PSB standard. The reason for this, the author believes, is that the PSB used the decomposition of the model into its spherical harmonic coefficients. In a paper published by Funkhouser (also involved in the PSB paper), it was found that using a model's spherical harmonic coefficients completely removes dependency on rotations (even rotations of parts of the model), while reducing the number of values to compare [3]. The reason this wasn't done in the author's implementation, was that he overlooked this as probably something that was performed on other descriptors; or just a fancy way to reduce dimensionality. So, in future versions of this implementation, this definitely must be the first this to try. Be wary, however, to understand spherical harmonics well, means to acquaint yourself with the mathematics of fourier analysis. To calculate the coefficients are another problem than just understanding what they are, so for that part, find some existing code to use. At the time of writing this paper, there are a few libraries for this, but in C, C++, or Fortran, so the code may have to be ported over to C++ first. Alternatively, the Java Native Interface could probably be used, but it'd probably by more painful doing it that way.

That concludes the overview of the global histogram project. If you find yourself working on this project, feel free to contact the author if you have any questions.

Bibliography

[1] Shilane, Philip and Min, Patrick and Kazhdan, Michael and Funkhouser, Thomas. The Princeton Shape Benchmark. June 2004.

[2] M. Ankerst, G. Kastenmuller, H.-P. Kriegel, and T. Seidl. Nearest neighbor classification in 3D protein databases. In Proc. ISMB, 1999.

[3] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3D shape descriptors. In Symposium on Geometry Processing, June 2003.

(BibTeX included in bibtex.txt)