3D Graphics Module

By Ramesh Srigiriraju

Computer Systems Research Lab 2006-07

Abstract

The purpose of this research project is to test different ways of eliminating whitespace in a 3D graphics program. Testing was done using a graphing calculator program that graphed functions of two variables using a Z-buffer algorithm. The first scheme I tested used a Z-buffer algorithm to determine which pixels were being obscured by pixels that were closer to the viewer. The second scheme I tested plotted triangles between data points after running a Z-buffer algorithm. The third scheme combined these two plotting methods.

Background & Introduction

Previous projects concerning this area of research include The Investigation of Graphics in the Processing Language by C. Fralick and the City Block Project by M. Levoy. The 3D graphics projects all used rotation matrices to rotate graphs, but the rotation formulas were all hard-coded. Possible state-of-the art programs are MatLab or other computer algebra systems.

Procedures

To test my programs, I used the graphing calculator to view graphs of the function $z=1/(x^2+y^2)$. The viewing angles used for all three schemes were $\varphi=\pi/4$, $\theta=-.1$. After saving screenshots of the viewing window in PNG format, the images were converted to PPM format and were then analyzed to calculate the percentage of the viewing window that was covered by whitespace.

Results

Scheme 1 (plotting points) worked well when the magnitude of the gradient of the function was less than one, but most of the whitespace occurred on areas of the function with a large gradient (for example, the asymptote of $z=1/(x^2+y^2)$). Scheme 2 (triangles) worked when the magnitude of the gradient was large, but not for flatter regions where it would be more likely for points to be obscured by other points. Scheme 3 (the combination) managed to eliminate most of the whitespace, except for regions where the magnitude of the gradient is approximately one.