A Programmable Pipeline for Graphics Hardware

by

Marc Olano

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Computer Science.

Chapel Hill

1998

Approved by:

Advisor: Anselmo Lastra

Reader: Steven Molnar

Reader: Jan Prins

Gary Bishop

Henry Fuchs

ABSTRACT

MARC OLANO: A Programmable Pipeline for Graphics Hardware

(Under the direction of Anselmo Lastra)

This dissertation demonstrates user-written procedures on an interactive graphics machine. Procedural shading is a proven technique for off-line rendering, and has been effectively used for years in commercials and movies. During most of that time, polygon-per-second performance has been a major focus for graphics hardware development. In the last few years, we have seen increased attention on surface shading quality for graphics hardware, principally image-based texture mapping. Today, even low-end PC products include support for image textures. The PixelFlow graphics machine demonstrates that techniques like procedural shading are possible at interactive rates. PixelFlow is the first machine to run, at real-time rates of 30 frames per second, procedural shaders written in a high-level shading language.

A graphics machine like PixelFlow is a large and complex device. An abstract pipeline is presented to model the operation of this and other interactive graphics machines. Each stage of the pipeline corresponds to one type of procedure that can be written by the graphics programmer. Through the abstract pipeline, the user can write shading procedures or procedures for other graphics tasks without needing to know the details of the machine architecture. We also provide a special-purpose language for writing some of these procedures. The special-purpose language hides more details of the machine implementation while enabling optimizations that make execution of the procedures possible.

ACKNOWLEDGMENTS

I would like to thank my advisor, Anselmo Lastra, for guidance, when it was necessary; for pressure, when it was necessary; and for freedom, when it was necessary. I would also like to thank my readers, Steve Molnar and Jan Prins for their comments, and the rest of my committee, Gary Bishop and Henry Fuchs.

I would like to thank Voicu Popescu and Rob Wheeler for their great work on the pfman compiler. While they both did many things, I would like to make a special acknowledgment of Voicu's work on the memory optimizer and Rob's work on the code generator and an early version of the fixed-point sizing code. I would also like to thank Yulan Wang, Peter McMurry and Greg Pruett for their work on the shading and math support libraries. Jon Leech deserves much of the credit for ensuring that the API extensions fit into the OpenGL framework.

For the their time, effort, patience, and enthusiasm, I thank everyone who used pfman, Alexandra Bokinsky, Chun-Fa Chang, Arthur Gregory, Arun Helser, Sang-Uok Kum, Renee Maheshwari, and Chris Wynn.

Finally, while I have only given individual mention to a handful of the people who worked on PixelFlow, they all deserve thanks. This dissertation would not have been possible without them. Those not already mentioned are Dan Aliaga, Greg Allen, Peter Arnold, John Austin, Randy Bailey, Tony Bennett, Vern Chi, Jon Cohen, David Ellsworth, Nick England, John Eyles, Trey Greer, Larry Harrold, Fred Heaton, Justin Heinecke, Rich Holloway, Kurtis Keller, Paul Keller, Mike Keshk, Lawrence Kesteloot, Paul Layne, Jon McAllister, Carl Mueller, Mike Myers, Bryon Nordquist, John Poulton, Krish Ponamgi, Chris Practico, Jiang Qian, Brad Ritter, Durward Rogers, Doug Schiff, Jason Smith, Mel Snyder, Steve Tell, Herman Towles, Jim Weaver, and Lee Westover.

TABLE OF CONTENTS

LIST OF FIGURES