Jonathan Bronson

University of Maryland, Baltimore County
Department of Computer Science and Electrical Engineering
1000 Hilltop Circle
Baltimore, MD 21250

Voice: (443)618-3726
Lab: (410)455-8935
E-mail: jonbron1@umbc.edu
Website: http://cs.umbc.edu/˜jonbron1/

August 29, 2007

RESEARCH INTERESTS

Computer Graphics and Visualization. Primary interests and areas of expertise include non-photorealistic rendering and user interaction based visualization.

EDUCATION

Aug 2006 - Present. M.S. student. University of Maryland Baltimore County.
Current research involves a collaboration of A.I. planning and map visualization techniques to aid in the districting of Elementary, Middle, and Highschool assignments. I am also involved in a collaboration with the geography department developing methods of navigating multiscale data.
Advisor: Dr. Penny Rheingans.

May 2006 B.S. Computer Science, University of Maryland Baltimore County.

EMPLOYMENT HISTORY

August 2007 - Present **Graduate Research Assistant**
Dept. of Computer Science and Electrical Engineering, University of Maryland, Baltimore County
"Interactive Visual Methods for Partitioning Multidimensional Spatial Data." The research project focuses on developing decision support tools for the problem of school redistricting. We are working with the Howard County Public School System to develop tools that will aid in generating, evaluating, and comparing alternative school assignment plans.

May 2007 - August 2007 **NCA Simulation Center Internship**
National Capital Area Medical Simulation Center
This internship was a goal oriented research opportunity. The main project of interest was automating the alignment of digital projectors through the use of low-grade cameras and computer vision techniques.

August 2006 - May 2007 **Graduate Research Assistant**
Dept. of Computer Science and Electrical Engineering, University of Maryland, Baltimore County
Continuing work on the PAWS project (Pipes And WaveS), visualizing large, complex, relational graphs in 3D and using wave-based motion to test relationships amongst the data.

Nov 2005 - August 2006 **Undergraduate Research Assistant**
Dept. of Computer Science and Electrical Engineering, University of Maryland, Baltimore County
Joined the PAWS project (Pipes And WaveS) in my senior year under Dr. Penny Rheingans, Dr. Tim Oates and Dr. Marie Desjardins. We began with a standard graph layout and developed a 3D counterpart to it. The software was developed in Java using JOGL (OpenGL binds for Java) with UC Berkeley's graph package Prefuse.

May 2005– August 2005 **Summer Intern**
General Physics, Elkridge, MD
Worked in a technical helpdesk office testing and validating new Computer Based Training (CBT) packages. Helped debug issues related to Java 1.5 upgrade. Gained more experience with Java, Python and SQL databasing.

PROJECTS

Real-Time Caustics for Deformable Meshes
The goal of this project was to combine two techniques, Shah et al.'s ”Caustics Mapping: An Image-space Technique for Real-time Caustics” and Oliveria et al.'s ”Real-Time Refraction Through Deformable Objects.” The caustics paper mentions how easily it can use Chris Wyman's Image-space refraction, but this newer technique requires no precomputation. Banking off our recent familiarization with XNA, we implemented both algorithms using the framework and developed a small world to test in. The caustics algorithm works efficiently by avoiding any explicit intersection calculations. Instead, the world positions of the scene are rendered to cubemap from the perspective of the translucent object. Refraction is then computed from the view of the lightsource and instead of looking up environment color, the environment positions are taken. These positions are then used to render point sprites using Vertex Texture Fetch. During our implementation we developed several frame rate improving techniques.

CPU-Based Particle System
Designed and implemented a CPU-based particle system in Direct3D. Unlike traditional particle systems which react to external forces, each particle decides its own path. The demo uses the system to model a swarm of bees, each which wishes to achieve its goal slightly differently as they “attack” a moving target,

Two-Surface Refraction Shader
Implemented a multipass refraction shader using GLSL and OpenGL. The refraction algorithm used is based off Chris Wyman's 2005 SIGGRAPH paper entitled, “An Approximate Image-Space Approach for Interactive Refraction.” In this method, depth information is used to approximate an exit point of the refracted ray and apply the second refraction to find the final exit direction.

Automated Generation of Stencils and Gobos from 3D Models
Designed and implemented an interactive program to generate printable stencils. The user can input an arbitrary 3D mesh and will receive a guideline for a printed stencil. Users can adjust line thickness, lighting, and camera angles to achieve the desired result. A GLSL fragment shader was written to obtain the artistic style of popular artistic stencils. A unique algorithm was developed that joins disconnected regions in a way that minimizes the destruction of information. These generated stencils can be used to fabricate hand usable stencils or sent to factory for Gobo production for use in theatrical stage lighting.

Webcam Multi-Projector Alignment
Designed and implemented a multi-projector alignment system for use on passive stereographic cave systems. The method involves using live video filters written in DirectShow and updating a series of glyphs to achieve gridded alignment of within one pixel of error at each corner.

BSP Culling
Implemented a first-person maze demo to demonstrate the usefulness of BSP Culling. The program was written using OpenGL and C++. At runtime, the user inputs the dimension of the maze and a random permutation is generated and automatically textured. Once generated, the user is placed inside the maze and can find his way through. Users are also able to change perspective at any time to see how well the Culling algorithm is reducing rendering cost. Features include Texturing, Collision Detection, and BSP culling.

Renderman and Procedural Shading
Implemented a program to render a scene using the Renderman renderer. Several surfaces shaders were written to simulate different properties of the scene. Some of these shaders included water, concrete, snow, and marble tile.

RayTracer
Implemented a general purpose Ray Tracing program in C++. Models were input from the Standard Procedural Database (SPD) or any NFF model file and rendered to an image file. Features included specular highlights, casted shadows, and anti-aliasing.

Computer Based Trainer (CBT)
Designed and Implemented a Computer Based Training Program (CBT); Developed for Professor Gary Burt, Dept. of Computer Science and Electrical Engineering aat UMBC. Written in Java (Platform Independent and GUI-Based), the CBT system provided a means for the professor to give additional study resources to students via his website. The design allowed the administrator to insert additional lecture or quiz material at any time. The user was free to read through past lectures and take randomized quizzes to test their performance.

AWARDS

- "Best M.S. Research" Award for "Automated Stencils and Gobos from 3D Models". UMBC CSEE Department Research Review, 2007.
- Awarded 2nd Place for Network Design at FBLA NLC, Future Business Leaders of America National Leadership Conference 2002, representing the State of Maryland.
- Awarded 1st Place for Network Design at FBLC MLC, Future Business Leaders of America Maryland, Regional Leadership Conference 2002, representing Anne Arundel County.

AFFILIATIONS

- UMBC Visualization, Animation, Non-Photorealistic Graphics, Object-Modeling and Graphics Hardware, (VANGOGH) Lab.