

Pandora's Box Graphic Engine - Uma Engine Gráfica  
com Aplicação em Visualização de Campos Tensoriais  
Proposta de Monografia

Andrew Toshiaki Nakayama Kurauchi  
5894035

Victor Kendy Harada  
5893823

2 de junho de 2011

*Supervisor:*  
Prof. Dr. Marcel Parolin Jackowski

# 1 Resumo da monografia

O desenvolvimento de programas de computação gráfica em tempo real exige a aplicação de um vasto conjunto de técnicas de otimização que são recorrentes em diversas aplicações. Por esse motivo são desenvolvidas engines gráficas como OpenSceneGraph [9] e Ogre3D [5]. Entretanto, tais engines evoluíram de tal forma que o tempo de aprendizagem é longo. O intuito desse trabalho é desenvolver uma engine gráfica que implemente técnicas básicas de computação gráfica e que seja fácil de ser aprendida e utilizada.

O estudo de tensores e campos tensoriais é de importância fundamental para diversas áreas do conhecimento. São encontradas aplicações no estudo de fenômenos sísmicos [2], estruturas eletrônicas [3] e imagens de ressonância magnética sensíveis a difusão [4]. A visualização de tais campos é, portanto, de grande relevância para o avanço do conhecimento. Por esse motivo uma das aplicações desenvolvidas com o auxílio da engine será um programa que permita tais visualizações. Serão utilizados para testes campos tensoriais provenientes de imagens de ressonância magnética sensíveis a difusão e campos representativos de deformações físicas em 3 dimensões

## 2 Objetivos do trabalho

Com o PandoraBox Graphic Engine visa-se atingir os seguintes objetivos:

- Construir uma engine gráfica de fácil utilização e aprendizado
- Criar uma ferramenta com aplicação em visualização em tempo real de campos tensoriais em 3 dimensões

## 3 Atividades já realizadas

As seguintes atividades já foram desenvolvidas:

- Estudo inicial sobre tensores e visualização de campos tensoriais
- Codificação da estrutura geral da engine:
  - Renderizador
  - Grafo de cena
  - Gerenciador de janelas
  - Câmera

- Transformações lineares
- Modelos
- Luzes
- Suporte a shaders customizados
- Leitura de campos tensoriais representados em arquivos no formato Analyze® [1]
- Cálculo dos autovalores e autovetores dos tensores e renderização da representação elipsoidal de tensores

## 4 Cronograma de Atividades

- Janeiro de 2011 - Definição e criação dos nós do grafo de cena que serão inicialmente suportados
- Fevereiro de 2011 - Implementação do buffer de vértices e do renderizador do grafo de cena
- Março de 2011 - Implementação da classe responsável por gerenciar os shaders e criação de modelos geométricos básicos
- Abril de 2011 - Suporte inicial a texturas e criação de métodos que calculam autovalores e autovetores de matrizes 3x3
- Maio de 2011 - Suporte a renderização eficiente de várias instâncias de um mesmo modelo (renderização instanciada) e a arquivos Analyze®(representação de campos tensoriais) e versão inicial da visualização do campo
- Junho de 2011 - Implementação da técnica de renderização em texturas e iniciar modificações na engine para utilizar a técnica de iluminação deferida e aplicação das técnicas de renderização instanciada e outras otimizações na visualização do campo tensorial
- Julho de 2011 - Continuação da implementação do suporte a iluminação deferida e adicionar suporte a pós-processamento da cena renderizada e implementação de um jogo simples como exemplo de utilização genérica da engine
- Agosto de 2011 - Desenvolvimento de mais exemplos de uso e implementação de frustum culling na engine
- Setembro de 2011 - Limpeza de código e, se houver disponibilidade de tempo, adicionar suporte a modelos translúcidos

## **5 Estrutura esperada da monografia**

### **5.1 Introdução**

Apresentação de engines gráficas alternativas existentes e os motivos para a criação de mais uma alternativa. Importância da visualização de campos tensoriais e como uma engine gráfica pode auxiliar o desenvolvimento de aplicações com tal propósito.

### **5.2 Conceitos e tecnologias estudadas**

Conceitos utilizados na implementação da engine, assim como tensores e suas formas de visualização. Além de uma breve introdução ao OpenGL.

### **5.3 Atividades realizadas**

Idéias aplicadas na solução dos problemas.

### **5.4 Resultados e produtos obtidos**

Trechos de código relevantes e resultados obtidos.

### **5.5 Conclusões**

Considerações e futuro do projeto desenvolvido.

### **5.6 Bibliografia**

Referências bibliográficas utilizadas ao longo do desenvolvimento da engine.

## Referências

- [1] Mayo/spm "analyze" format spec compilation. <http://www.grahamwideman.com/gw/brain/analyze>. Acessado em junho de 2011.
- [2] Chris Engelsma and Dave Hale. Cwp-655 visualization of 3d tensor fields derived from seismic images. <http://www.cwp.mines.edu/Meetings/Project10/cwp-655.pdf>. Acessado em junho de 2011.
- [3] Martin Head-Gordon. Tensor concepts in electronic structure theory: Application to self-consistent field methods and electron correlation techniques. *Modern Methods and Algorithms of Quantum Chemistry, J. Grotendorst (Ed.), John von Neumann Institute for Computing, Jülich, NIC Series, Vol. 1, ISBN 3-00-005618-1*, 1:561–562, 2000.
- [4] Peter B. Kingsley. Introduction to diffusion tensor imaging mathematics: Part I. Tensors, rotations, and eigenvectors. *Concepts in Magnetic Resonance Part A*, 28A(2):101–122, March 2006.
- [5] Ogre3d. <http://www.ogre3d.org>. Acessado em maio de 2011.
- [6] Opengl 2.1 reference pages. <http://www.opengl.org/sdk/docs/man>. Acessado em junho de 2011.
- [7] Opengl 3.3 reference pages. <http://www.opengl.org/sdk/docs/man3>. Acessado em junho de 2011.
- [8] Opengl 4.1 reference pages. <http://www.opengl.org/sdk/docs/man4>. Acessado em junho de 2011.
- [9] Openscenegraph. <http://www.openscenegraph.org>. Acessado em maio de 2011.