



FURB – Universidade Regional de Blumenau
DSC – Departamento de Sistemas e Computação
Grupo de Pesquisa em Computação Gráfica, Processamento de Imagens e Entretenimento Digital
Disciplina: Sistemas Multimídia - prof. Dalton Solano dos Reis

Projeto - Sistema Multimídia

Data: 2014-1

Equipe (nomes): GUSTAVO RUFINO FELTRIN / LEONARDO FERNANDES

Título: ESTEREOSCOPIA

Problema: Obter a percepção de profundidade em um espaço tridimensional utilizando imagens em 2D.

Objetivos: Utilizar o recurso de estereoscopia do OpenGL para permitir a percepção de profundidade em um espaço 3D. Tentar reproduzir o efeito de estereoscopia de forma manual, criando duas cenas vistas de ângulos diferentes uma ao lado da outra, permitindo ao monitor/projetor efetuar a junção.

Relevância: Através da estereoscopia a imersão do usuário em um ambiente 3D pode ser aumentada (LEITE, 2006), porém para viabilizá-la é necessário conhecer as técnicas de estereoscopia, as características do hardware que irá projetar/exibir a imagem e dominar alguma biblioteca gráfica que implemente o recurso para produzir este efeito. O trabalho proposto visa testar a utilização dos recursos de estereoscopia da biblioteca gráfica OpenGL, assim como um projetor 3D e um *notebook* 3D como hardware para demonstrar o efeito.

1 ETAPA INTERMEDIÁRIA

A seguir serão descritos conceitos relacionados ao trabalho de pesquisa proposto a serem apresentados na fase intermediária.

1.1 CENÁRIO DO TRABALHO PROPOSTO

Utilizando os recursos do Laboratório LIFE (Projetor 3D e Notebook 3D) criar um *Hello World* com estereoscopia 3D através dos recursos do OpenGL.

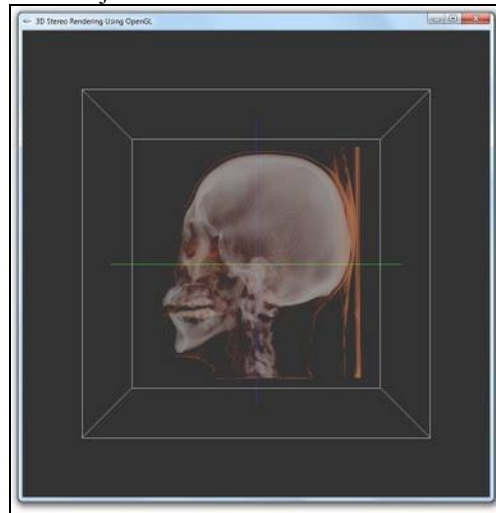
1.2 TRABALHOS CORRELATOS

Como trabalhos correlatos a este, foram selecionadas, o Projeto IA369E, o Projeto Portable rear projection stereoscopic display, e o projeto Creating Anaglyphs using OpenGL. No Quadro 1 podem ser observadas as principais características entre o trabalho proposto e os correlatos selecionados.

1.2.1 PROJETO IA369E

O Projeto IA369E tem como objetivo a criação de um visualizador volumétrico interativo. O mesmo foi desenvolvido pelos acadêmicos do curso de pós-graduação em Engenharia Elétrica da Faculdade de Engenharia Elétrica e de Computação (FEEC) da Universidade Estadual de Campinas (UNICAMP) Rodrigo Mologni Gonçalves dos Santos e Rafael Guimarães Ramos (SANTOS, 2011). Motivados pela aquisição de um monitor com suporte a estereoscopia pelo departamento do curso, pelo fato da maioria dos projetos gráficos do departamento utilizar OpenGL e ser a linguagem adotada na disciplina de Computação Gráfica, os acadêmicos desenvolveram um aplicativo para habilitar o suporte a estereoscopia em 3D para OpenGL. O aplicativo consiste em um set de comandos em OpenGL que habilitam o suporte e criam a renderização de dois pontos de vista diferentes. A Figura 1 exibe o resultado da renderização estéreo 3D de um conjunto de dados volumétricos de tomografia.

Figura 1- Conjunto de dados volumétricos em estéreo 3D



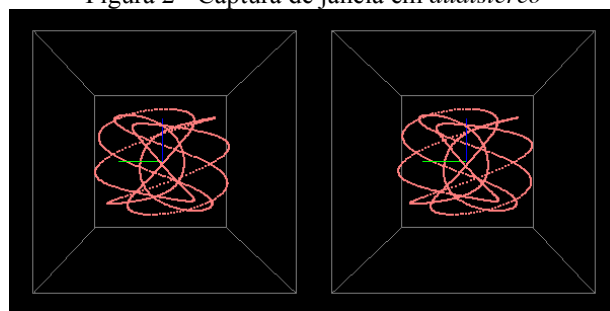
Fonte: Santos (2011).

Segundo Santos (2011), “Com o uso do monitor foi possível verificar que a visão estéreo funcionou. Entretanto a sensação de profundidade não ficou tão relevante quanto ao esperado.”

1.2.2 Portable rear projection stereoscopic display

O Projeto visa construir um visualizador de retroprojeção estereoscópica portátil, através do OpenGL a reprodução de duas visões em uma única imagem deixando ao hardware a responsabilidade de gerar a estereoscopia. (BOURKE, 2014). A Figura 2 demonstra o resultado do projeto, duas imagens lado a lado da mesma cena, mas de dois pontos de vista diferentes.

Figura 2 - Captura de janela em *dualstereo*

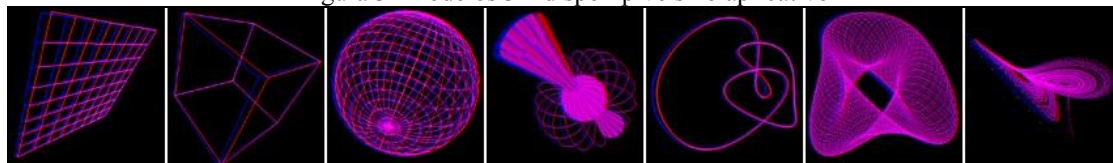


Fonte: Bourke (2014).

1.2.3 Creating and Viewing Anaglyphs

Projeto que demonstra a criação de anáglifos usando OpenGL e buffer de acumulação. Suporta todas as principais cores de filtro e demonstra como formar corretamente pares estéreos (BOURKE, 2000). A Figura 3 demonstra os diferentes modelos 3D disponíveis no aplicativo de demonstração.

Figura 3- Modelos 3D disponíveis no aplicativo



Fonte: Bourke (2000).

Características	Proposto	PROJETO IA369E	Portable rear	Creating and Viewing Anaglyphs
Esteroscopia 3D com OpenGL	X	X	X	X
Método Toed-in	X	X	X	
Método Frustum assimétrico paralelo ao eixo estéreo	X		X	
Estéreo anáglifo	X			X
Estéreo polarizado	X	X	X	

Quadro 1 – proposto versus correlatos

1.3 CONCEITOS RELACIONADOS A SISTEMAS MULTIMÍDIA

Para o desenvolvimento deste projeto se faz necessário o entendimento de alguns conceitos, tais como: estereoscopia, técnicas de visualização, anáglifo, óculos obturador, óculos polarizado, HMD, autoestereoscopia, e como o OpenGL faz uso da estereoscopia.

1.3.1 ESTEREOSCOPIA

O termo estereoscopia deriva das palavras gregas *stereos* e *skopein*, que significam visão sólida ou ver em relevo. Esta visão estereoscópica advém da natureza óptica de nossa fisiologia, pelo fato do homem possuir uma distância média entre os olhos de 6,5 cm, cada imagem é processada em um lado do cérebro formando uma imagem final única onde os pontos focais visuais coincidem. Por esta imagem final ter sido originada de duas imagens ligeiramente distintas uma da outra, obtemos a noção de proporção e profundidade, devido a diferença de enquadramento. Artifícios como perspectiva, iluminação, oclusão e sombra também permitem passar a noção de profundidade (MASCHIO, 2008).

A disparidade e a paralaxe são entidades que parametrizam quantitativamente diferenças entre as imagens estereoscópicas. A disparidade é a diferença/espacamento das imagens formadas na retina de cada olho, medida na retina em centímetros (cm) ou em milímetros (mm). A paralaxe é a distância entre dois pontos de visão projetados num anteparo de visualização, medida em mm ou *pixels*.

1.3.2 TÉCNICAS DE VISUALIZAÇÃO

Para reproduzir o efeito de estereoscopia são utilizadas algumas técnicas diferentes que dependem de um hardware específico para cada técnica. A seguir são descritas as principais técnicas de visualização de imagens estereoscópicas.

1.3.2.1 ANÁGLIFO

Esta técnica consiste na divisão de uma imagem em seu espectro de cor, de forma que a imagem da esquerda possua uma gama de cores e a imagem da direita possua outra gama complementar. Para visualizar uma imagem em anáglifo é necessário o uso de óculos com filtro de cores que permitam apenas a passagem de cores direcionadas a cada olho. Geralmente são usadas nas lentes as cores vermelho e ciano. Não é necessária uma tela de projeção específica, a imagem em anáglifo pode ser impressa, vista em um monitor ou televisão (TOMOYOSE, 2010).

1.3.2.2 ÓCULOS OBTURADOR

Consiste em obter o efeito de estereoscopia através do sincronismo entre óculos obturador e imagem exibida. Quando a imagem esquerda for exibida a lente direita do óculos fica escura, quando a imagem direita é exibida a lente esquerda fica escura, e o ciclo se repete pelo menos sessenta vezes por segundo. O óculos obturador utiliza lentes de cristal líquido para criar a oclusão da imagem, e requer o sincronismo com o equipamento que esta reproduzindo a imagem, como monitor, projetor ou televisão.

Além dos diversos formatos para gerar as imagens, existem diferentes padrões para conectar os óculos no dispositivo que reproduz a imagem e possibilitar a sincronização entre a geração das imagens e o bloqueio dos olhos do observador (MASCHIO, 2008).

1.3.2.3 ÓCULOS POLARIZADO

Esta é uma técnica de estereoscopia através da polarização da luz. São utilizados filtros polarizadores, os quais fazem com que as imagens projetadas do par estereoscópico sejam polarizadas em planos ortogonais. Desta forma, o observador utiliza óculos com filtros polarizadores ortogonais correspondentes aos planos de projeção e vê com cada olho apenas uma das imagens projetadas. A fusão das imagens vistas em cada olho resultará na visão estereoscópica (Machado, 1997, p.31).

1.3.2.4 DISPLAY MONTADO NA CABEÇA

É um dispositivo de vídeo, usado na cabeça como um capacete e com fones de ouvido,

usado como uma interface através da qual o usuário pode experimentar um ambiente de realidade virtual. É, provavelmente, o mais amplo visor visual usado em sistemas de Realidade Virtual (WIKIPEDIA, 2013b), também comumente conhecido como Head Mounted Display (HMD).

1.3.2.5 AUTOESTEREOSCOPIA

O autoestereoscopia é o método para reproduzir imagens tridimensionais que podem ser exibidas sem que o usuário precise usar equipamentos especiais (como óculos ou capacetes) e necessitam de condições especiais de iluminação. Assim também geralmente chamados de 3D sem óculos (WIKIPEDIA, 2013a).

1.3.2.6 OpenGL ESTEREO

OpenGL é uma API gráfica multi-plataforma, com suporte a uma vasta gama de hardware e software, e suporte a renderização *stereo quad-buffered*. *Quad-buffered* é a capacidade de renderizar as imagens (esquerda e direita) em um buffer frontal, e outras imagens (esquerda e direita) em um buffer auxiliar de forma independente. Os buffers frontais exibem as imagens *stereo* sendo trocadas em sincronia com os óculos obturadores, enquanto os buffers auxiliares estão sendo atualizados (TAN, 2014).

1.3.3 HARDWARE UTILIZADO

Esta seção trata sobre as características estereoscópicas dos hardwares utilizados no experimento do trabalho proposto, como meios para reproduzir o efeito.

1.3.3.1 PROJETOR 3D BENQ MX720

É um projetor 3D com resolução XGA (1024x768) e frequência vertical de 120Hz, que utiliza uma tecnologia de projeção desenvolvida pela Texas Instruments denominada *Digital Light Processing* (DLP), conforme Figura 4.

Figura 4- Projetor BENQ MX720



Fonte: BENQ (2014).

A tecnologia DLP consiste em uma matriz de espelhos microscópicos organizados em uma matriz sobre um chip semicondutor conhecido como *Digital Micromirror Device*

(DMD). Cada espelho representa um pixel na imagem projetada, e devido ao rápido posicionamento dos espelhos é possível variar a intensidade da luz refletida através deles. Há dois métodos para criar uma imagem colorida, o primeiro é utilizando um disco de cores em alta rotação sincronizado com o DMD entre a lâmpada e o chip, e o segundo, utilizando três chips DMD onde cada um recebe uma das luzes primárias emitidas pela lâmpada e divididas por um prisma (WIKIPEDIA, 2014).

O chip de imagem DLP é tão rápido, que é capaz de projetar duas imagens na tela ao mesmo tempo para criar imagens 3D com óculos ativos. Graças ao DLP Link, um sistema de sincronização especial feito para projetores DLP 3D Ready, não é necessário emissores especiais ou dispositivos de transmissão terceiros entre óculos e projetor. Entre cada quadro de vídeo o chip envia dados para os óculos através do DLP Link (TEXAS INSTRUMENTS, 2014).

1.3.3.2 ÓCULOS 3D ATIVO G05-DLP

É um óculos ativo para projeção 3D, compatível com a tecnologia DLP Link conforme pode ser visto na Figura 5. Possui uma bateria interna de *lithium-ion* recarregável através de uma conexão USB, frequência de 120Hz, 50 horas de funcionamento contínuo e 3 horas de tempo para recarga. Utiliza lentes de cristal líquido para oclusão.

Figura 5 - Óculos ativo compatível com DLP Link



Fonte: Gonbes (2014).

1.3.3.3 ÓCULOS ANÁGLIFO

Óculos para projeção 3D anáglifo para visualizar vídeos, fotos e impressões. Conforme pode ser visto na Figura 6, o óculos possui uma lente vermelha e outra na cor ciano.

Figura 6 - Óculos anáglifo



Fonte: Aliexpress (2014).

1.3.3.4 NOTEBOOK 3D LG A560

Notebook com tela de LCD de 15,6" com resolução Full HD (1920x1080) e tecnologia 3D passiva (polarizado), webcam Dual 3D e tecnologia Intel® InTru™ 3D, conforme Figura 7. Acompanhado de um par de óculos passivos.

Figura 7 - Notebook LG 3D



Fonte: LG (2014).

1.4 AMBIENTES UTILIZADOS NO DESENVOLVIMENTO

Será utilizado a IDE Visual Studio 2012, através da linguagem de programação C++ e a biblioteca gráfica OpenGL 2.0.

1.5 REQUISITOS PRINCIPAIS DO PROBLEMA A SER TRABALHADO

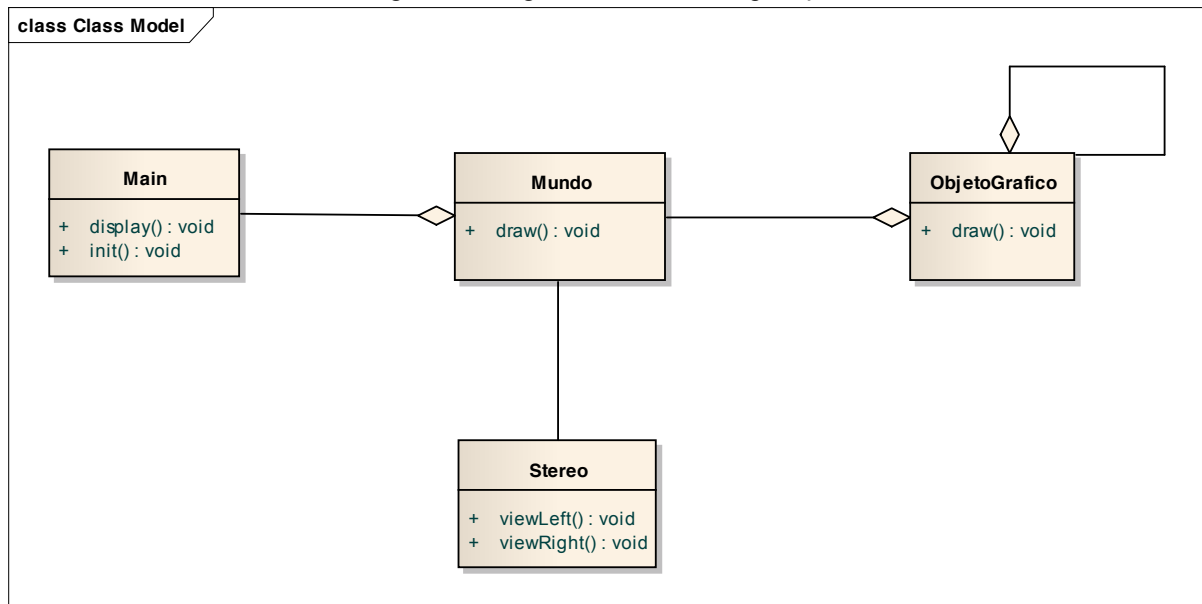
A ferramenta proposta deverá atender os seguintes requisitos:

- utilizar o notebook 3D do laboratório LIFE para produzir o efeito de estereoscopia (Requisito Não Funcional – RNF);
- utilizar o projetor 3D do laboratório LIFE para produzir o efeito de estereoscopia (RNF);
- utilizar o óculos anáglifo do laboratório LIFE para visualizar imagens estereoscópicas anáglifas (RNF);
- utilizar os recursos de estereoscopia do OpenGL para criar o efeito estereoscópico (RNF);

e) criar o efeito de estereoscopia em cenas 3D (Requisito Funcional – RF).

1.6 ESPECIFICAÇÃO

Figura 8 – Diagrama de classes da aplicação.



Fonte: Elaborada pelo autor.

2 ETAPA FINAL

Nesta seção serão descritos conceitos relacionados ao trabalho de pesquisa proposto.

2.1 IMPLEMENTAÇÃO

A seguir são demonstrados trechos de códigos essenciais para reproduzir o efeito de estereoscopia, para os diferentes métodos propostos.

2.1.1 MÉTODO FRUSTUM ASSIMÉTRICO PARALELO AO EIXO ESTEREO

A Figura 9 demonstra a inicialização do OpenGL sem nenhuma *flag* específica a estereoscopia definida.

Figura 9 – Inicialização do OpenGL

```
14  glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
```

Fonte: Elaborada pelo autor.

A Figura 10 demonstra o set de comandos para perspectiva de visão direita. Este trecho de código é executado antes de desenhar a cena. Devido à sequência de comandos a visão direita deve ocorrer antes da visão esquerda. A função da linha 155 define o *buffer* de retaguarda para renderização. Este buffer é composto por dois *buffers*, um direito e um esquerdo. Na linha 162 é possível verificar o *buffer* direito como o escolhido.

Figura 10 – Set de comandos para a perspectiva de visão direita.

```
155  glDrawBuffer(GL_BACK);
156  glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
157  glMatrixMode(GL_PROJECTION);
158  glLoadIdentity();
159  glFrustum(leftR, rightR, bottom, top, camera->nr, camera->fr);
160  glViewport(camera->screenwidth/2, 0, camera->screenwidth/2, camera->screenheight);
161  glMatrixMode(GL_MODELVIEW);
162  glDrawBuffer(GL_BACK_RIGHT);
163  glLoadIdentity();
164  gluLookAt(eyexR, eyeyR, eyezR, centerxR, centeryR, centerzR, upx, upy, upz);
```

Fonte: Elaborada pelo autor.

A Figura 11 demonstra um *print* na *console* dos valores dos parâmetros das funções `glFrustum` e `glutLookAt` para a visão direita.

Figura 11 – Parâmetros das funções glFrustum e glutLookAt.

```
glFrustum(-0.36,0.33,-0.35,0.35,0.60,60.00)
gluLookAt(-6.00,-0.12,0.00,-5.00,-0.12,0.00,0.00,0.00,1.00)
```

Fonte: Elaborada pelo autor.

A Figura 12 demonstra o set de comandos para perspectiva de visão esquerda. Este trecho de código é executado após desenhar a cena da visão direita, e antes de desenhar a cena da visão esquerda. Na linha 162 é possível verificar o *buffer* esquerdo como o escolhido.

Figura 12 – Set de comandos para a perspectiva de visão esquerda.

```
211 glMatrixMode(GL_PROJECTION);
212 glLoadIdentity();
213 glFrustum(leftL,rightL,bottom,top,camera->nr,camera->fr);
214 glViewport(0,0,camera->screenwidth/2,camera->screenheight);
215 glMatrixMode(GL_MODELVIEW);
216 glDrawBuffer(GL_BACK_LEFT);
217 glLoadIdentity();
218 gluLookAt(eyexL,eyeyL,eyezL,centerxL,centeryL,centerzL,upx,upy,upz);
```

Fonte: Elaborada pelo autor.

A Figura 13 demonstra um *print* na *console* dos valores dos parâmetros das funções glFrustum e glutLookAt para a visão esquerda.

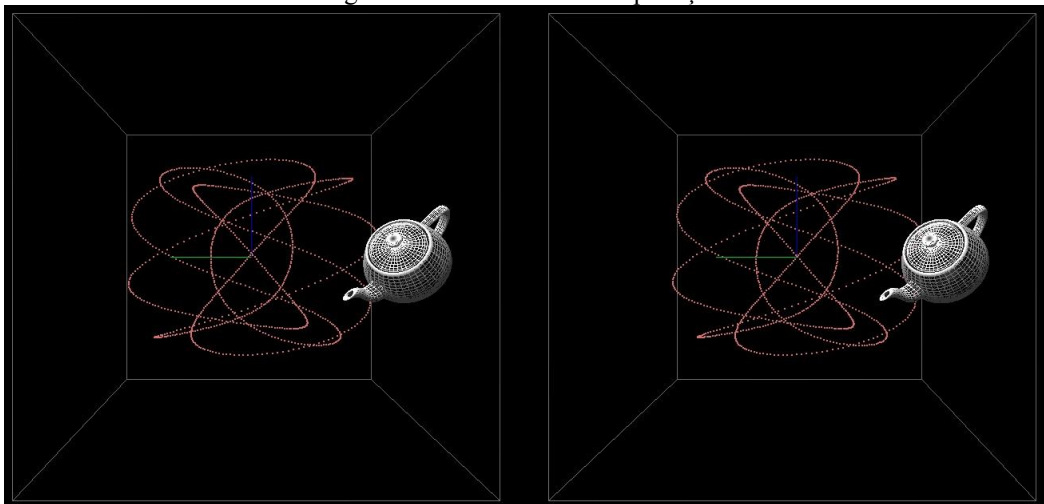
Figura 13 - Parâmetros das funções glFrustum e glutLookAt.

```
glFrustum(-0.33,0.36,-0.35,0.35,0.60,60.00)
gluLookAt(-6.00,0.12,0.00,-5.00,0.12,0.00,0.00,0.00,1.00)
```

Fonte: Elaborada pelo autor.

A Figura 14 exibe um *print screen* da aplicação. É possível verificar a renderização da mesma cena sobre duas perspectivas de visão diferentes, deixando a responsabilidade de juntá-las ao equipamento de irá projetar a imagem em 3D.

Figura 14 – *Print screen* da aplicação.



Fonte: Elaborada pelo autor.

2.1.2 MÉTODO TOED-IN

Este método possui o mesmo set de comandos do Método Frustum Assimétrico Paralelo ao Eixo Estereo (seção 2.1.1). O que caracteriza a diferença entre as técnicas são os valores atribuídos aos comandos `glFrustum` e `gluLookAt`. Esta técnica é conhecida uma técnica incorreta.

A Figura 15 demonstra um *print* na *console* dos valores dos parâmetros das funções `glFrustum` e `gluLookAt` para a visão direita.

Figura 15 - Parâmetros das funções `glFrustum` e `gluLookAt`.

```
glFrustum(-0.33,0.36,-0.35,0.35,0.60,60.00)
gluLookAt(-6.00,-0.14,0.00,-5.00,-0.12,0.00,0.00,0.00,1.00)
```

Fonte: Elaborada pelo autor.

A Figura 16 demonstra um *print* na *console* dos valores dos parâmetros das funções `glFrustum` e `gluLookAt` para a visão esquerda.

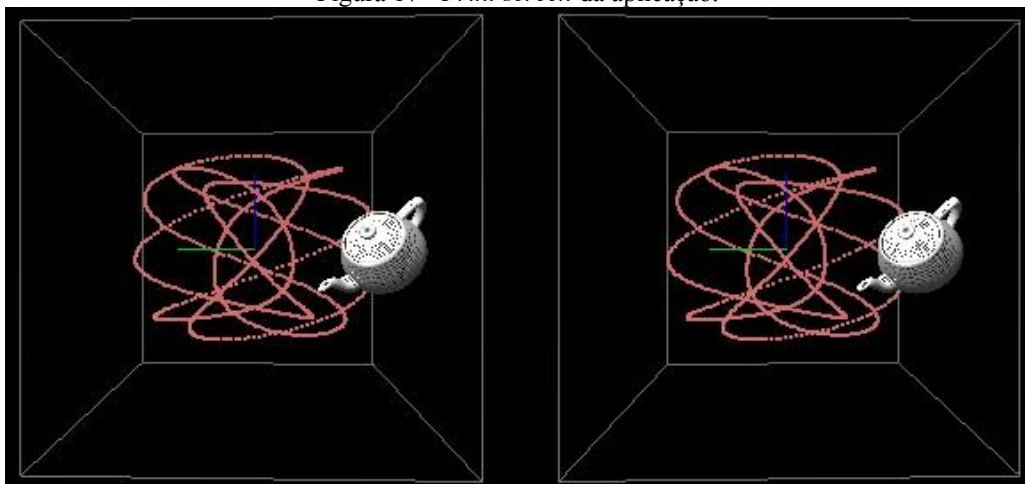
Figura 16 - Parâmetros das funções `glFrustum` e `gluLookAt`.

```
glFrustum(-0.33,0.36,-0.35,0.35,0.60,60.00)
gluLookAt(-6.00,0.14,0.00,-5.00,0.12,0.00,0.00,0.00,1.00)
```

Fonte: Elaborada pelo autor.

A Figura 17 exibe um *print screen* da aplicação. É possível perceber que o ângulo de visão das duas visões são diferentes porem o *frustum* é igual.

Figura 17 - *Print screen* da aplicação.



Fonte: Elaborada pelo autor.

2.1.3 MÉTODO ANÁGLIFO

A Figura 18 demonstra a inicialização do OpenGL sem nenhuma *flag* específica a estereoscopia definida.

Figura 18 – Inicialização do OpenGL

```
14  glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
```

Fonte: Elaborada pelo autor.

A Figura 19 demonstra o set de comandos para perspectiva de visão direita. Este trecho de código é executado antes de desenhar a cena. Devido à sequência de comandos a visão direita deve ocorrer antes da visão esquerda. A função da linha 179 define o *buffer* de retaguarda para renderização. Este *buffer* é composto por dois *buffers*, um direito e um esquerdo. Na linha 184 é possível verificar a função que define a mascara de cor de um lado da cena.

Figura 19 – Set de comandos para a perspectiva de visão direita.

```
179  glDrawBuffer(GL_BACK);
180  glReadBuffer(GL_BACK);
181  glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
182  glClear(GL_ACCUM_BUFFER_BIT); /* Not strictly necessary */
183  glViewport(0,0,camera->screenwidth,camera->screenheight);
184  glColorMask(GL_TRUE,GL_FALSE,GL_FALSE,GL_TRUE);
185  glMatrixMode(GL_PROJECTION);
186  glLoadIdentity();
187  glFrustum(leftR,rightR,bottom,top,camera->nr,camera->fr);
188  glMatrixMode(GL_MODELVIEW);
189  glLoadIdentity();
190  gluLookAt(eyexR,eyeyR,eyezR,centerxR,centeryR,centerzR,upx,upy,upz);
```

Fonte: Elaborada pelo autor.

A Figura 20 demonstra um *print* na *console* dos valores dos parâmetros das funções *glFrustum* e *gluLookAt* para a visão direita.

Figura 20 - Parâmetros das funções *glFrustum* e *gluLookAt*.

```
glFrustum<-1.42,1.35,-0.69,0.69,1.20,60.00>
gluLookAt<-6.00,-0.12,0.00,-5.00,-0.12,0.00,0.00,0.00,1.00>
```

Fonte: Elaborada pelo autor.

A Figura 21 demonstra o set de comandos para perspectiva de visão esquerda. Este trecho de código é executado após desenhar a cena da visão direita, e antes de desenhar a cena da visão esquerda. Na linha 239 é possível verificar a função que define a mascara de cor de outro lado da cena. Na linha 233 pode-se verificar o comando que acumula no *buffer* a cena anterior desenhada.

Figura 21 – Set de comandos para a perspectiva de visão esquerda.

```

231 glFlush();
232 glColorMask(GL_TRUE, GL_TRUE, GL_TRUE, GL_TRUE);
233 glAccum(GL_LOAD, 1.0);
234 glDrawBuffer(GL_BACK);
235 glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
236 glMatrixMode(GL_PROJECTION);
237 glLoadIdentity();
238 glFrustum(leftL, rightL, bottom, top, camera->nr, camera->fr);
239 glColorMask(GL_FALSE, GL_TRUE, GL_TRUE, GL_TRUE);
240 glMatrixMode(GL_MODELVIEW);
241 glLoadIdentity();
242 gluLookAt(eyexL, eyeyL, eyezL, centerxL, centeryL, centerzL, upx, upy, upz);

```

Fonte: Elaborada pelo autor.

A Figura 22 demonstra um *print* na *console* dos valores dos parâmetros das funções `glFrustum` e `gluLookAt` para a visão esquerda.

Figura 22 - Parâmetros das funções `glFrustum` e `gluLookAt`.

```

glFrustum<-1.35,1.42,-0.69,0.69,1.20,60.00>
gluLookAt<-6.00,0.12,0.00,-5.00,0.12,0.00,0.00,0.00,1.00>

```

Fonte: Elaborada pelo autor.

A Figura 23 demonstra o set de comandos após a cena esquerda. Na linha 257 é possível verificar o comando que acumula no buffer a segunda cena sobrepondo à primeira. A linha 258 exibe o comando que copia o buffer de acumulação para o buffer de retaguarda.

Figura 23 – Set de comandos após desenhar a cena esquerda.

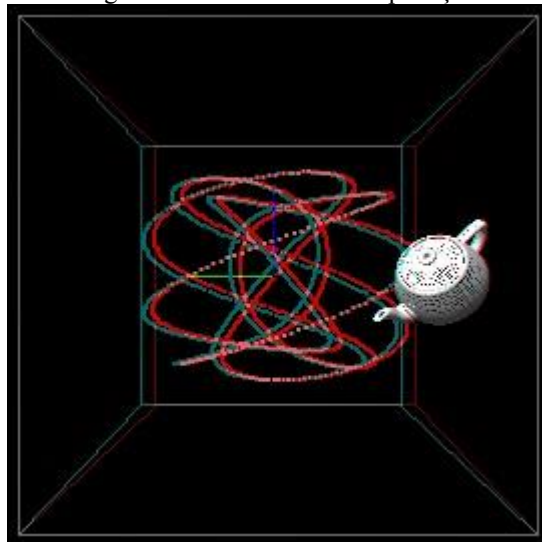
```

255 glFlush();
256 glColorMask(GL_TRUE, GL_TRUE, GL_TRUE, GL_TRUE);
257 glAccum(GL_ACCUM, 1.0);
258 glAccum(GL_RETURN, 1.0);

```

Fonte: Elaborada pelo autor.

A Figura 24 exibe um *print screen* da aplicação. É possível perceber a sobreposição das duas visões da cena.

Figura 24 - *Print screen* da aplicação

Fonte: Elaborada pelo autor.

2.1.4 MÉTODO FRUSTUM ASSIMÉTRICO PARALELO AO EIXO ESTEREO COM SUPORTE EM HARDWARE

O experimento deste método não foi realizado pelo fato de que o único equipamento disponível com suporte em hardware a estereoscopia não permitiu habilitar o recurso.

A Figura 25 demonstra a inicialização do OpenGL, a *flag* GLUT_STEREO define que o OpenGL usará o suporte a estereoscopia do hardware.

Figura 25 - Inicialização do OpenGL

```
12 | glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH | GLUT_STEREO);
```

Fonte: Elaborada pelo autor.

A Figura 26 demonstra o set de comandos para perspectiva de visão direita. Este trecho de código é executado antes de desenhar a cena. Devido à sequência de comandos a visão direita deve ocorrer antes da visão esquerda. Na linha 149 é possível verificar a definição do *buffer* direito.

Figura 26 - Set de comandos para a perspectiva de visão direita.

```
144 | glMatrixMode(GL_PROJECTION);
145 | glLoadIdentity();
146 | glFrustum(leftR, rightR, bottom, top, camera->nr, camera->fr);
147 | glViewport(0, 0, camera->screenwidth, camera->screenheight);
148 | glMatrixMode(GL_MODELVIEW);
149 | glDrawBuffer(GL_BACK_RIGHT);
150 | glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
151 | glLoadIdentity();
152 | gluLookAt(eyexR, eyeyR, eyezR, centerxR, centeryR, centerzR, upx, upy, upz);
```

Fonte: Elaborada pelo autor.

A Figura 27 demonstra o set de comandos para perspectiva de visão esquerda. Este trecho de código é executado após desenhar a cena da visão direita, e antes de desenhar a cena da visão esquerda. Na linha 203 é possível verificar a definição do *buffer* esquerdo.

Figura 27 - Set de comandos para a perspectiva de visão esquerda.

```

198 glMatrixMode(GL_PROJECTION);
199 glLoadIdentity();
200 glFrustum(leftL, rightL, bottom, top, camera->nr, camera->fr);
201 glViewport(0, 0, camera->screenwidth, camera->screenheight);
202 glMatrixMode(GL_MODELVIEW);
203 glDrawBuffer(GL_BACK_LEFT);
204 glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
205 glLoadIdentity();
206 gluLookAt(eyexL, eyeyL, eyezL, centerxL, centeryL, centerzL, upx, upy, upz);

```

Fonte: Elaborada pelo autor.

2.2 RESULTADOS E DISCUSSÃO

Sobre a utilização dos recursos do Laboratório Interdisciplinar de Formação de Educadores (LIFE), não foi possível realizar testes utilizando o suporte em hardware a estereoscopia. O único equipamento que oferece este suporte (Notebook 3D) não permitia ativar o suporte estéreo para o OpenGL na placa gráfica. As limitações de acesso administrador impossibilitaram testes com outros drivers de vídeo. Em contrapartida o projetor 3D, os óculos ativos e o óculos anáglifo disponibilizado pelo professor, nos permitiram realizar experimentos e obter resultados.

Através dos métodos propostos foi possível obter a percepção de profundidade nas cenas criadas.

Método	Pontos Positivos	Pontos Negativos
Frustum assimétrico	Boa percepção de profundidade.	Necessidade de hardware de projeção 3D. Depende da taxa de exibição e qualidade do hardware de projeção 3D.
Frustum assimétrico (suporte em hardware)	(não experimentado)	(não experimentado)
Toed-in		Necessidade de hardware de projeção 3D. Depende da taxa de exibição e qualidade do hardware de projeção 3D. Desconforto visual.
Anáglifo	Não necessita de hardware de projeção 3D.	Baixa percepção de profundidade. Perda no espectro de cores. Necessita de um óculos anáglifo.

2.3 CONSIDERAÇÕES FINAIS

O suporte a estereoscopia provido pelo OpenGL viabiliza a construção de aplicações estereoscópicas de forma satisfatória. Ele não se caracteriza como um agente limitador na utilização de métodos distintos que explorem estereoscopia. A equipe sentiu falta de mais equipamentos com suporte a estereoscopia para obter resultados sobre outras características, como por exemplo: frequências de projeção diferentes, monitor 3D ao invés de projetor, monitor de autoestereoscopia, etc.

REFERÊNCIAS BIBLIOGRÁFICAS

ALIESPRESS. **10pcs Red Blue 3D Glasses Dimensional Anaglyph Movies Free shipping**. Disponível em: <<http://www.aliexpress.com/item/10pcs-Red-Blue-3D-Glasses-Dimensional-Anaglyph-Movies-Free-shipping/452201883.html>>. Acesso em: 25 maio 2014.

BOURKE, Paul. **Portable rear projection stereoscopic display**. Crawley, 2014. Disponível em: <<http://paulbourke.net/exhibition/vpac/opengl.html>>. Acesso em: 23 abr. 2014.

_____. **Creating and Viewing Anaglyphs**. [S.I.], 2000. Disponível em: <<http://paulbourke.net/stereographics/anaglyph/>>. Acesso em: 30 abr. 2014.

GONBES, **DLP-Link 3D Glass for Benq DLP-Link**. Shenzhen, 2014. Disponível em: <<http://gonbes.en.made-in-china.com/product/WvnJurOYvacz/China-DLP-Link-3D-Glass-for-Benq-DLP-Link-3D-Projectors-Eyewear-G05-DLP-.html>>. Acesso em: 25 maio 2014.

LEITE, Pedro José Silva. **OpenStereo**: Uma aplicação para suporte ao desenvolvimento de aplicações estereoscópicas. [S.I.], 2006. Disponível em: <<http://www.cin.ufpe.br/~tg/2006-1/pjsl.pdf>>. Acesso em: 23 abr. 2014.

LG. **Mais perto da realidade com a tecnologia 3D**. [S.I.], 2014. Disponível em: <<http://www.lge.com/br/computadores/lg-A560-T.BG77P1-7457/reviews>>. Acesso em 25 maio 2014.

MACHADO, Liliane. S. **A realidade virtual em aplicações científicas**. INPE [INPE-6389-TDI/605], São José dos Campos, 1997.

MASCHIO, Alexandre Vieira. **A Estereoscopia**: Investigação de aquisição, edição e exibição de imagens estereoscópicas em movimento. [S.I.], 2008. Disponível em: <<http://www.faac.unesp.br/Home/Pos-Graduacao/Design/Dissertacoes/alexandre.pdf>>. Acesso em: 18 maio 2014.

SANTOS, Rodrigo Mologni Gonçalves dos. **PROJETO IA369E**: Renderização Estereoscópica 3D em OpenGL. [S.I.], 2011. Disponível em: <<http://www.dca.fee.unicamp.br/courses/IA369E/2s2011/projects/RMGSantosRGRamos/tutoriais/renderizacao-estereoscopica-3d-em-opengl/>>. Acesso em: 23 abr. 2014.

TAN, S. L. Ezekie. **Stereo geometry in OpenGL**. [S.I.]. Disponível em: <<http://www.orthostereo.com/geometryopengl.html>>. Acesso em: 18 maio 2014.

TOMOYOSE, Alexandre Nascimento. **Comparação e Classificação de Técnicas de Estereoscopia para Realidade Aumentada e Jogos**. São Paulo, 2010. Disponível em: <http://www.teses.usp.br/teses/disponiveis/3/3141/tde-20102010-114358/publico/Dissertacao_Alexandre_Nascimento_Tomoyose.pdf>. Acesso em: 18 maio 2014.

TEXAS INSTRUMENTS. **DLP Link**. [S.l.], 2014. Disponível em: <<http://www.dlp.com/projector/dlp-innovations/dlp-link.aspx>>. Acesso em: 25 maio 2014.

WIKIPEDIA. **Autoestereoscopia**. [S.l.], 2013a. Disponível em: <<http://es.wikipedia.org/wiki/Autoestereoscopia>>. Acesso em: 18 maio 2014.

_____. **Head-mounted display**. [S.l.], 2013b. Disponível em: <http://pt.wikipedia.org/wiki/Head-mounted_display>. Acesso em: 18 maio 2014.

_____. **DLP**. [S.l.], 2014. Disponível em: <<http://pt.wikipedia.org/wiki/DLP>>. Acesso em: 25 maio 2014.