

Lorenzo Ridolfi

Geração de imagens estereo sintéticas com luz estruturada

Introdução

- A reconstrução da representação 3d de um objeto pode ser realizada com auxílio de imagens estéreo do objeto em questão
- As imagens são estéreo porque são obtidas a partir de 2 pontos de vista diferentes
- Para possibilitar a obtenção da profundidade do objeto, uma luz estrutura, isto é um padrão de cores, é projetado sobre o objeto
- Normalmente o objeto é rotacionado e um par de imagens estéreo é gerado em diversos ângulos de rotação distintos
- Este trabalho consistiu na geração de imagens estéreo sintetizadas no OpenGL, visando permitir a sua reconstrução posterior com técnicas de geometria epipolar.

Modelo 3D

- O modelo 3D usado foi um “mesh” de um cavalo do jogo de xadrez
- O mesh original possui 8256 vértices e 2752 triângulos
- Foi feita uma otimização nos vértices, colapsando os vértices cuja distância entre si fosse menor do que 10^{-4}
- O mesh otimizado ficou com 1397 vértices
- A otimização dos vértices permitiu o cálculo de normais ponderadas, o que tornou a superfície do cavalo mais suave, facilitando a reconstrução 3D

Original

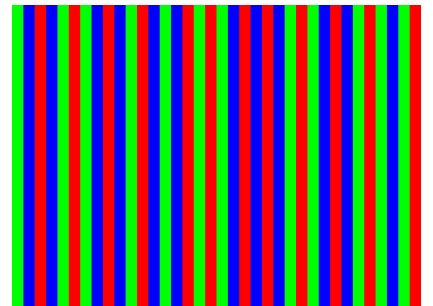
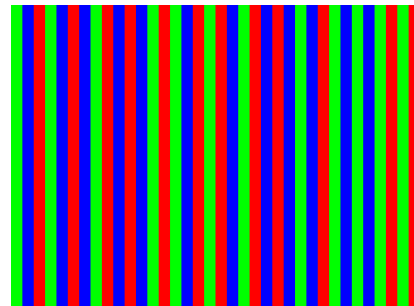
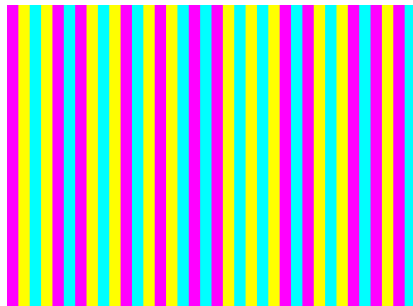
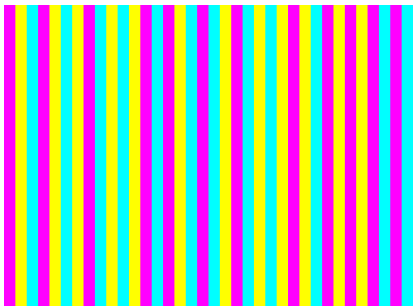


Otimizado



Projeção

- Visando permitir a recuperação da profundidade do objeto 3d, uma luz estruturada, composta de um padrão de listras, é projetada sobre o cavalo
- O padrão empregado foi o padrão de 3, 2 BCSL, composto de 4 imagens:



Textura projetiva

- O padrão de cores foi projetado sobre o objeto usando a técnica de “textura projetiva” no OpenGL
- Nesta técnica, constroi-se uma matrix de transformação T vinculada a textura que é obtida da seguinte forma:
 - Cria-se uma matriz P de contém a transformação da câmera virtual que simula o projetor que irá projetar o padrão sobre o objeto
 - A matriz de transformação da textura é dada pela seguinte fórmula:

$$T = P \cdot MV^{-1}$$

- Onde MV é a matriz Modelview do OpenGL

Geração de imagens

- As imagens foram geradas da seguinte maneira:
 - Foram geradas imagens rotacionando o objeto ao longo do eixo Y do OpenGL nos seguintes ângulos:
 - 0,30, 60, 90, 120, 150, 180, 210, 240, 270, 300, 330
 - Para cada ângulo, foram geradas imagens a partir de 2 pontos de vista diferentes, para obtermos as imagens estéreo
 - Para cada ponto de vista, 8 imagens foram geradas
 - Com os 4 padrões da Asla dispostos horizontalmente
 - Com os 4 padrões da Asla dispostos verticalmente
 - No total, foram geradas 192 imagens

Parâmetros das câmeras

■ Câmera 1

■ Gerada a partir dos comandos:

- `gluLookAt(51.7638, 50.f, 200.f, 0.of, -10.of, 0.of, 0.of, 1.of, 0.of);`
- `gluPerspective(45, 1024.0/768.0 , 0.01, 1000);`

■ Câmera 2

■ Gerada a partir dos comandos:

- `gluLookAt(-51.7638, 50.f, 200.f, 0.of, -10.of, 0.of, 0.of, 1.of, 0.of);`
- `gluPerspective(45, 1024.0/768.0 , 0.01, 1000);`

■ Projetor

■ Listras verticais:

- `gluPerspective(30, 1024.0/768.0, 0.1, 1000.0);`
- `gluLookAt(0.0, 50.0, 200.0, 0.0, -10.0, 0.0, 0.0, 1.0, 0.0);`

■ Listras horizontais:

- `gluPerspective(30, 1024.0/768.0, 0.1, 1000.0);`
- `gluLookAt(0.0, 10.0, 200.0, 0.0, -10.0, 0.0, 0.0, 1.0, 0.0);`

Matriz Modelview das Câmeras

■ Câmera 1

$$\begin{bmatrix} 0,9681 & 0 & -0,2505 & 0 \\ -0,0699 & 0,9603 & -0,2700 & 9,603 \\ 0,2406 & 0,2789 & 0,9296 & -212,3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

■ Câmera 2

$$\begin{bmatrix} 0,9681 & 0 & 0,2505 & 0 \\ 0,0699 & 0,9603 & -0,2700 & 9,603 \\ -0,2406 & 0,2789 & 0,9296 & -212,3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Distância focal

- Para calcular a distância focal em pixels das câmeras, a partir do parâmetro *fov* do comando *gluPerspective* do OpenGL, utilize a seguinte equação:

$$fov = 2 \cdot \tan\left(\frac{altura}{2 \cdot distância}\right)^{-1}$$

- Onde *altura* é altura da janela do OpenGL em pixels

Exemplos de imagens geradas

