Lorenzo Ridolfi

Reconstrução 3D

Introdução

- Este trabalho tem objetivo de efetuar a reconstrução 3d de um objeto, que é feita pela obtenção de um conjunto de pontos 3d que pertencem a superfície do objeto, gerados a partir de um conjunto de imagens estéreo
- As imagens estéreo foram geradas com uma luz estruturada, que é um padrão de listras que visa auxiliar na detecção da coordenada de profundidade dos pontos 3D
- O objeto é rotacionado e várias imagens são tiradas a cada ângulo de rotação para permitir a reconstrução de todo o objeto

Imagens de entrada

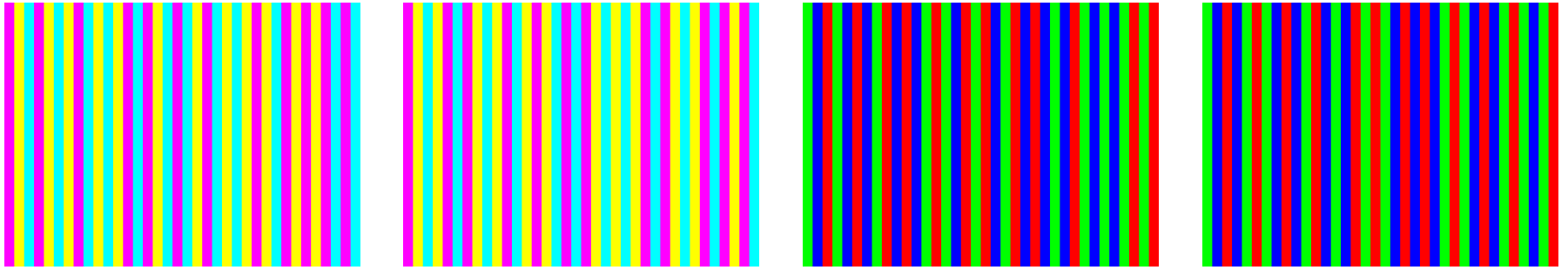
- As imagens empregadas nessa geração 3d são imagens sintéticas, geradas a partir de um modelo 3d feito no OpenGL
- O uso de imagens geradas em um ambiente controlado permitiu obtermos precisamente:
  - Os parâmetros das câmeras usadas na geração das imagens
  - O eixo de rotação das imagens
  - O ângulo de rotação entre as imagens
- A presença desses parâmetros, facilitou a reconstrução 3d do objeto, eliminando variáveis e incertezas
- Embora tenham sido fornecidas imagens com as listras horizontais e verticais, apenas as imagens com listras verticais foram utilizadas

Parâmetros das câmeras fornecidos

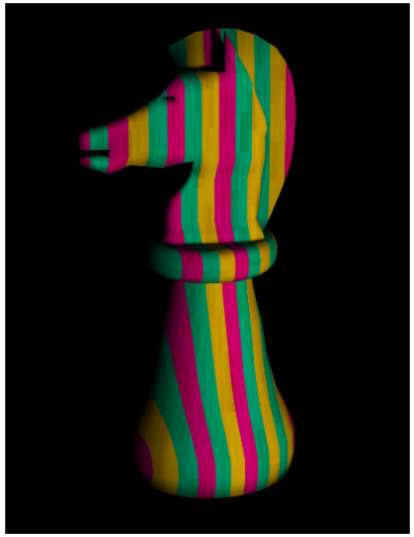
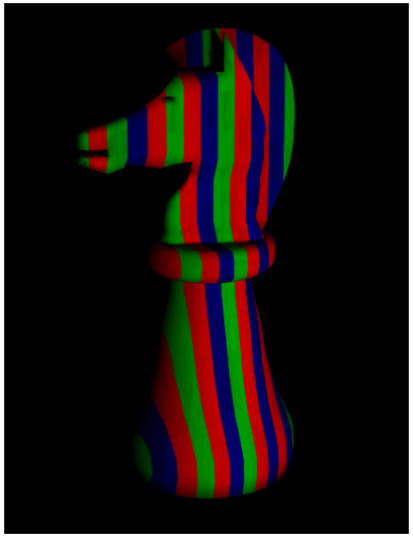
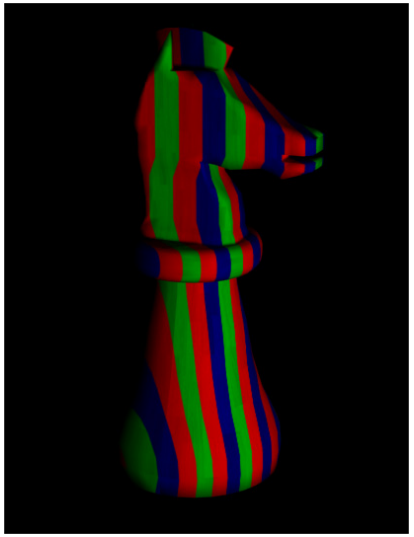
- As seguintes informações foram fornecidas:
  - Câmera 1
    - gluLookAt(51.7638, 50.f, 200.f, 0.0f, -10.0f, 0.0f, 0.0f, 1.0f, 0.0f) ;
    - gluPerspective(45, 1024.0/768.0 , 0.01, 1000);
  - Câmera 2
    - gluLookAt(-51.7638, 50.f, 200.f, 0.0f, -10.0f, 0.0f, 0.0f, 1.0f, 0.0f) ;
    - gluPerspective(45, 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) ;
  - Rotação do objeto
    - Rotação no eixo Y = 0
    - Passo da rotação: 30 graus
  - Distância focal da câmera em pixels
    - 927.05801

Projeção

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



Exemplos de imagens de entrada


Estratégia de reconstrução

- A reconstrução é dividida em duas grandes etapas:
  - Detecção das transições

    A detecção de transições é essencialmente uma etapa de processamento de imagens 2d que visa detectar as transições inseridas pelo padrão de listras da Asla. Por meio de interpolações, as transições são detectadas com precisão de sub-pixel
  - Obtenção das coordenadas 3D

    Após a obtenção das transições em 2D, os pontos 3d são obtidos a partir de técnicas de geometria epipolar. O *match* entre as transições de cada uma das imagens estéreo e feito com o auxilio de retas epipolares. Uma vez encontrado o *match*, o ponto 3d é calculado por meio da matriz essencial.

Estratégia de reconstrução

- Mais especificamente, a reconstrução é dividida nos seguintes passos e sub-passos:
  - Detecção das transições
    - Transformação das cores da imagem
    - Aplicar filtro suavizador e diferenciador
    - Filtrar as transições falsas
    - Interpolar transição
    - Classificar a transição
    - Armazenar a transição
  - Obtenção das coordenadas 3D
    - Calcular a reta epipolar
    - Calcular o match da reta epipolar com as transições
    - Calcular o ponto 3d
    - Rotacionar o ponto

Visão geral da Detecção de Transições

- A Detecção de Transições possui o seguinte pseudo-código:

Para cada rotação do objeto

Para cada uma das 2 imagens estéreo

Para cada linha da imagem

Transformação das cores da imagem

Aplicar filtro suavizador e diferenciador

Filtrar as transições falsas

Interpolar transição

Classificar a transição

Armazenar a transição

Transformação das cores da imagem

- O primeiro sub-passo da Detecção das Transições transforma as cores da imagem de uma representação RGB para o espaço CIELCH, visando facilitar a detecção de transições
- O espaço CIELCH, que é derivado do espaço CIELab, isola em cada uma das suas três coordenadas as principais características de uma cor:
  - Luminância
  - Chroma (Saturação)
  - Hue (Matiz)
- No espaço CIELCH, para detectar uma transição, apenas a coordenada H (Hue) precisa ser considerada, pois representando a matiz da cor, registrando com precisão as faixas de cor da projeção.
- Se o RGB fosse usado, as três variáveis (R, G e B) precisariam ser verificadas na detecção da transição

Aplicar filtro suavizador e diferenciador

- Após a transformação para o espaço CIELCH, um filtro suavizador e um diferenciador é aplicado na coordenada H.
- O filtro suavizador tem o objetivo de eliminar eventuais *aliasing* que hajam na imagem
- O filtro diferenciador tem o objetivo de realçar as transições
- Como estamos considerando apenas as listras verticais, optamos por empregar um *kernel* diferenciador e suavizador 1D, apenas no sentido horizontal.
- O *kernel* suavizador escolhido foi o:

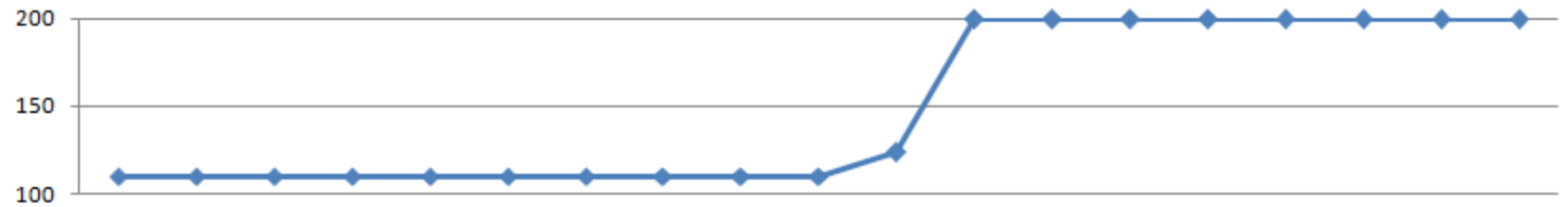
0,25	0,50	0,25

- O *kernel* diferenciador escolhido foi o:

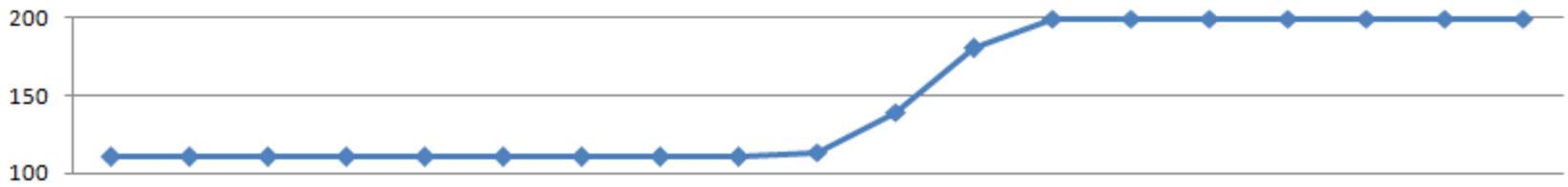
-0,50	0,50

Aplicar filtro suavizador

- Por exemplo, a se pegarmos a coordenada H de uma linha da imagem, conforme o exemplo abaixo:

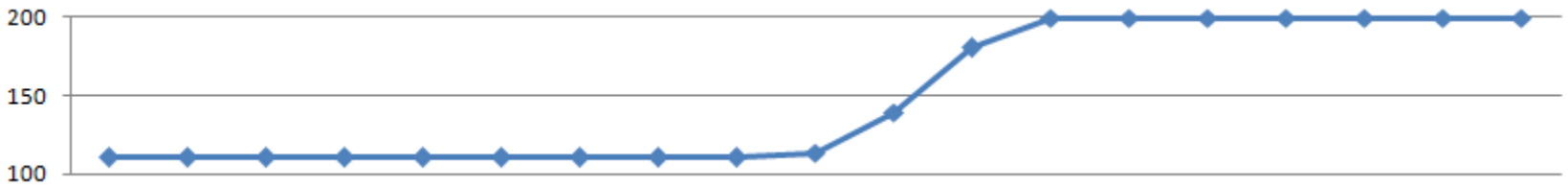


- Aplicando o *kernel* suavizador, temos:



Aplicar filtro diferenciador

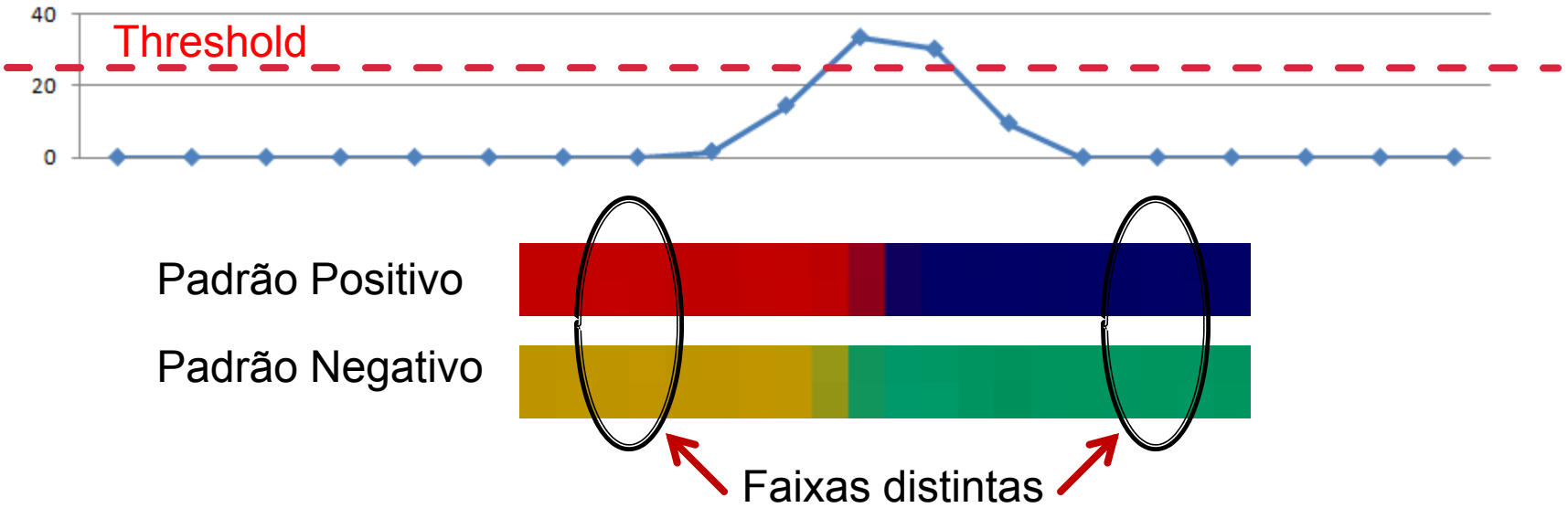
- Por exemplo, a se pegarmos a coordenada H já suavizada de uma linha da imagem, conforme o exemplo abaixo:



- Aplicando o *kernel* diferenciador, temos:

Filtrar as transições falsas

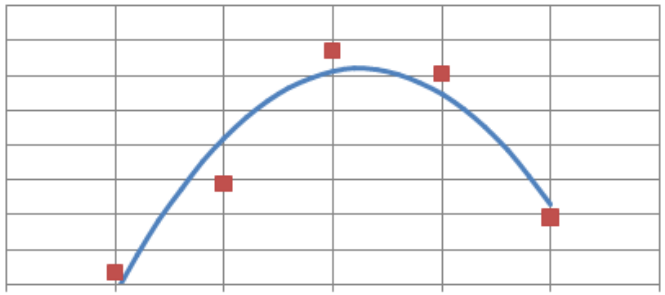
- Após a aplicação do filtro diferenciador, as transições que não são relacionadas as faixas do padrão projetado devem ser eliminadas
- Isso é feito de duas maneiras
  - Elimina-se as transições cujo máximo seja inferior a um *threshold*
  - Verifica-se se os pixels vizinhos da transição estão localizados em faixas distintas no padrão projetado, caso contrário, a transição é desconsiderada



Interpolar a transição

- Visando calcular a transição com precisão de sub-pixel, é feito um ajuste de curva sobre as transições que não foram filtradas

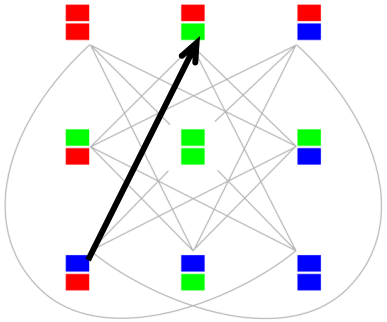
- Para cada máximo local de uma transição, é calculado o polinômio do segundo grau que faz um "*fit*" nos 5 pontos vizinhos ao máximo.
- O valor da coordenada x da transição é o ponto máximo do polinômio de segundo grau

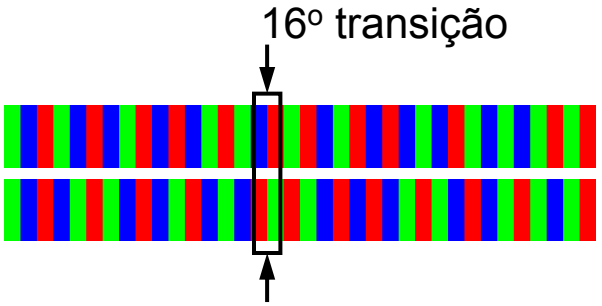


Classificar a transição

- A transição é classificada de acordo com a codificação BCSL, em função das cores dos pixels vizinhos

vertices	d(0)	d(1)	d(2)	d(3)
V(00)	0	3	6	9
V(01)	14	17	19	11
V(02)	28	34	22	24
V(10)	26	29	18	21
V(11)	1	31	33	35
V(12)	15	4	8	13
V(20)	16	23	32	12
V(21)	27	5	7	25
V(22)	2	10	20	30





Armazenar a transição

- Para possibilitar o "*match*" entre as transições, cada transição é armazenada numa estrutura de dados, na forma da seguinte tupla:

  <*rotação*, *câmera*, *classificação*, *x*, *y*>

  Onde

  - *rotação* é a rotação do objeto ao longo do eixo Y
  - *câmera* indica qual das duas câmeras estéreo gerou a transição
  - *classificação* é o resultado da classficação pelo BCSL
  - *x* é a coordenada x da transição
  - *y* é a coordenada y da transição

Visão geral da Obtenção da Coordenada 3D

- A Obtenção da Coordenada 3D possui o seguinte pseudo-código:

Para cada rotação do objeto

Para cada classificação de transição

Para cada transição

Calcular a reta epipolar

Calcular o match da reta epipolar com as transições

Calcular o ponto 3d

Rotacionar o ponto

Calcular a reta epipolar

- A reta epipolar é usada para fazer o *match* entre as transições das duas câmeras
- Para obtermos a reta epipolar, temos que calcular primeiro a matriz fundamental F
- Após obter a matriz F, uma reta epipolar é obtida para cada uma das transições da câmera 1

$$\mathbf{p}_r = \begin{pmatrix} x \\ y \\ f \end{pmatrix} \qquad \mathbf{u}_r = \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$

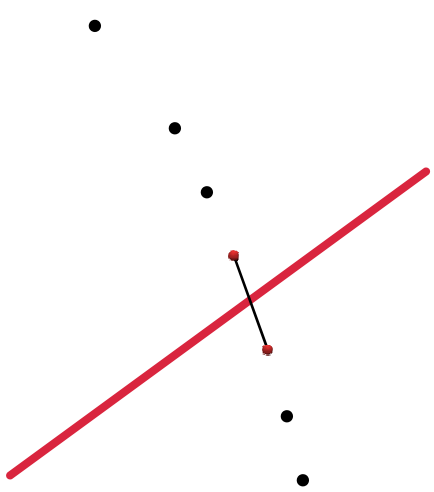
$$\mathbf{u}_r = \mathbf{F}\mathbf{p}_l$$

$$\mathbf{p}_r^{\,T}\mathbf{u}_r = 0$$

$$\mathbf{p}_r^{\,T}\mathbf{u}_r = ax + by + cf = 0$$

Calcular o match

- De posse da reta epipolar obtida, procura-se as duas transições mais próximas na câmera 2, para a mesma rotação e classificação
  - A transição mais próxima acima da reta
  - A transição mais próxima abaixo da reta
- Se a distância mínima de uma transição for maior do que um *threshold*, a reta epipolar é descartada
- O match é o cruzamento da reta epipolar com a reta formada pelos dois pontos mais próximos



Calcular o ponto 3D

- Com as coordenadas x, y da transição que deu origem a reta epipolar da câmera 1 e as coordenadas x,y do *match* da câmera 2, podemos facilmente calcular o ponto 3d com o uso da matriz essencial E da geometria epipolar

$$\mathbf{p}_r{}^T \mathbf{E} \mathbf{p}_l = 0$$

$$\mathbf{u}_r = \mathbf{E}\mathbf{p}_l$$

$$\mathbf{p}_r{}^T \mathbf{u}_r = 0$$

$$\mathbf{p}_r = \begin{pmatrix} x \\ y \\ f \end{pmatrix} \qquad \mathbf{u}_r = \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$

$$\mathbf{p}_r{}^T \mathbf{u}_r = ax + by + cf = 0$$

Rotacionar o ponto 3d

- Como o objeto foi rotacionado durante a geração das imagens estéreo, essa rotação precisa ser desfeita
- A rotação foi realizada em torno do eixo Y=0, portanto basta multiplicar o ponto obtido na etapa anterior pela seguinte matriz:

$$\begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} = \begin{bmatrix} \cos\phi & 0 & -\sin\phi \\ 0 & 1 & 0 \\ \sin\phi & 0 & \cos\phi \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Resultados alcançados

- O seguinte conjunto de pontos foi obtido com o uso da reconstrução 3D descrita neste documento: