

3D photography on your desk

— Teoria e prática

Horácio Macêdo
Issufi Badji

Estrutura da Apresentação

- A Teoria
 - Descrição Teórica do Tema
 - Apresentação dos Artigos
- A Prática
 - Ferramentas
 - Resultados
 - Desafios

A Teoria

Descrição Teórica do Tema

- A calibração de câmera e luz
 - Extrair informações
 - Escalas dos objetos
 - Propriedades da câmera

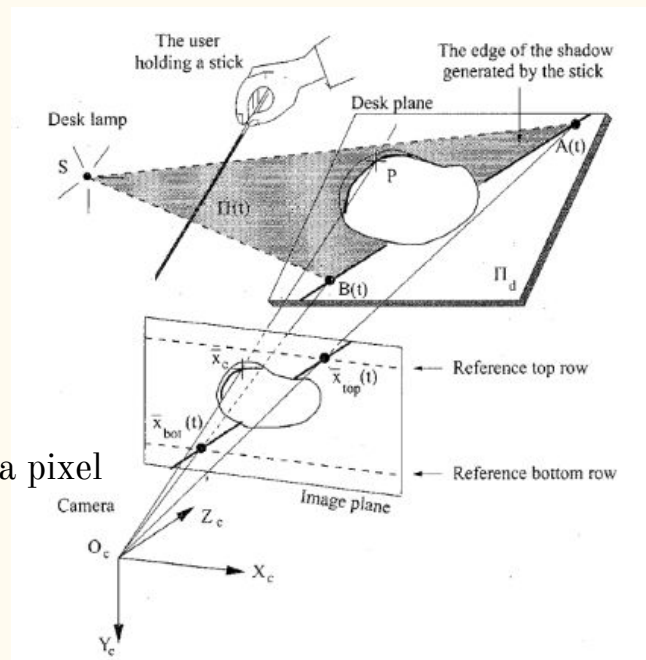
Apresentação dos Artigos

- Autores: Bouguet e Perona
 - 1998: Um plano e uma fonte de luz
 - 1999: Dois planos
- Visão geral do artigo
 - O artigo explica como **extrair formato tridimensional(3d)** de objetos em uma imagem com base na **iluminação fracamente estruturada** usando apenas um luz, um lápis e um fundo quadrilátero para calibração.

Apresentação dos Artigos

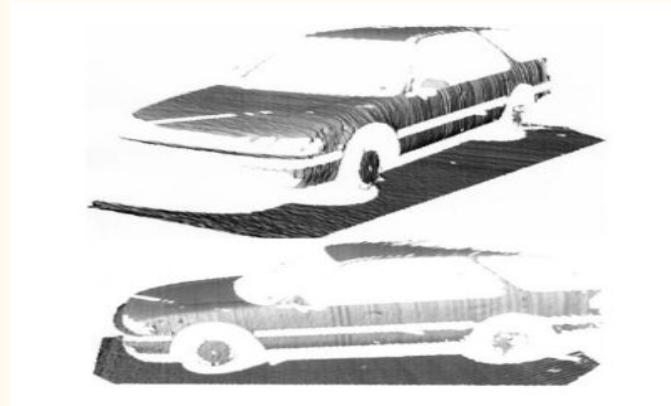
A técnica consiste em três passos:

- Calibração
 - da câmera: parâmetros intrínsecos, definição do plano do tampo da mesa
 - da luz: posição do ponto de luz
 - Não é necessário se houver dois planos ortogonais (1999)
- Mapeamento da sombra
 - no espaço: onde está a reta da sombra em dado tempo
 - no tempo: em qual momento a sombra passa sobre cada pixel
- Triangulação
 - fazer a relação entre a reta da sombra e o momento em que cada pixel deixa de ser sombreado



Apresentação dos Artigos

- A ideia central do experimento
 - Fazer o scanner do objeto de mundo real para o espaço virtual



A Prática

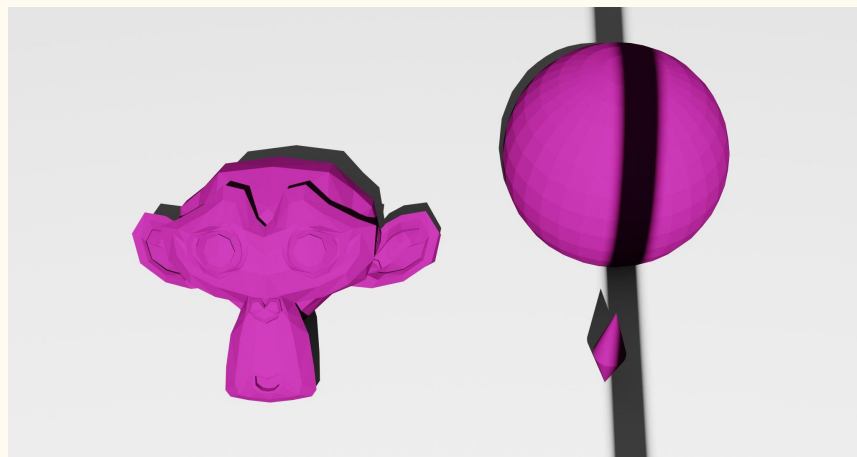
Ferramentas

- Blender
 - Confeção de cenas sintéticas para teste da cena
- Python 3.9
 - 748 linhas de curtição
 - OpenCV para gerência de imagens e calibração de câmera
 - NumPy para matemática ~~em tempos~~ ~~humanamente possíveis~~
 - Matplotlib para reconstrução de malha
- Nikon D3300
 - Cortesia do PROGRAF
- Objetos diversos
 - Foca de pelúcia
 - Dados de RPG
 - Xícara
 - Club Social
 - Cola Pritt
 - Trena



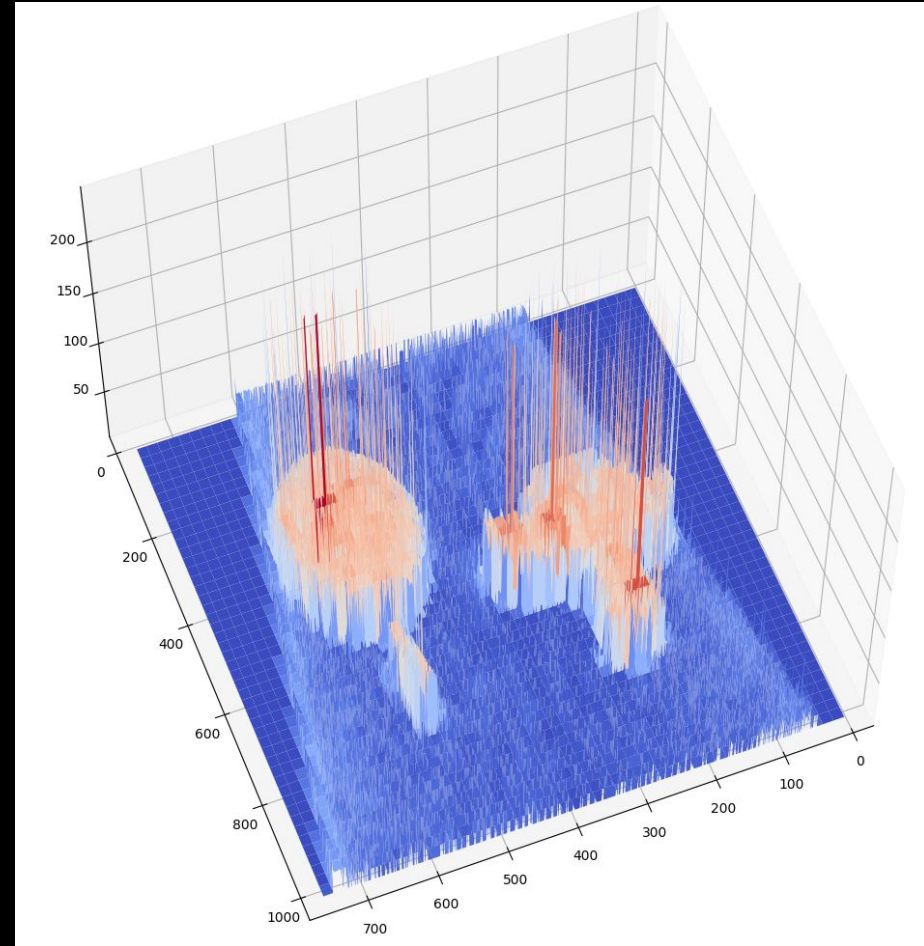
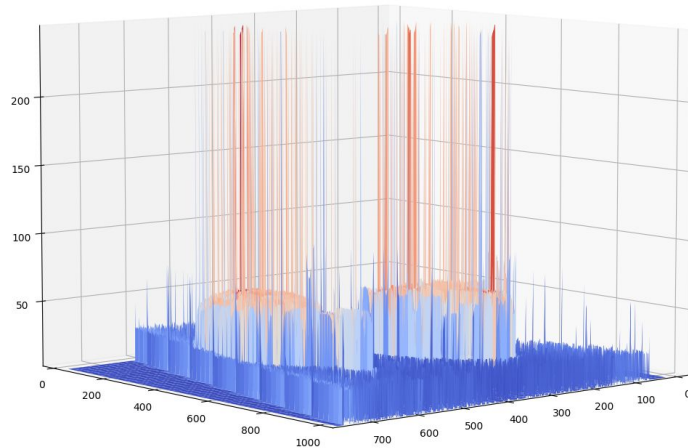
Ferramentas realmente usadas

- Blender
 - Confecção de cenas sintéticas para teste da cena
 - Duas cenas:
 - 1024 x 768, 100 frames, luz do sol
 - 3840 x 2160, 100 frames, luz de holofote
- Python 3.9
 - 748 linhas de curtição
 - OpenCV para gerência de imagens e calibração de câmera
 - NumPy para matemática ~~em tempos humanamente possíveis~~
 - Matplotlib para reconstrução de malha
 - Duas versões:
 - Uma das versões funcionou nos dois testes, mas gerou resultados piores
 - Outra versão gerou resultados melhores, mas funcionou apenas no primeiro teste
- Café do Trailer do Adriano

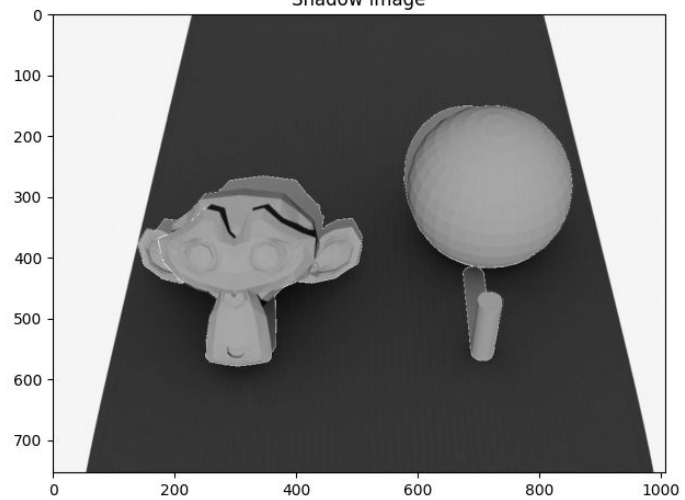


Resultados

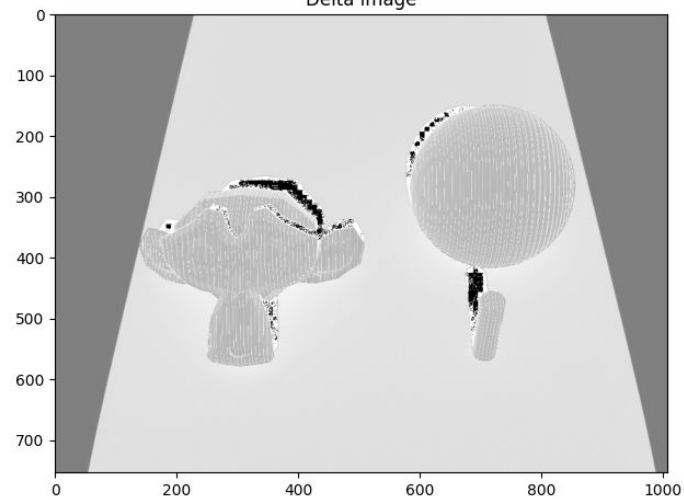
- $Y_{top} = 20px$
- $Y_{bottom} = 760px$
- Contrast threshold = 80
- Kernel size = 1 (sem borramento)
- Camera calibration pattern side = 80mm
- Lamp calibration object height = 70mm



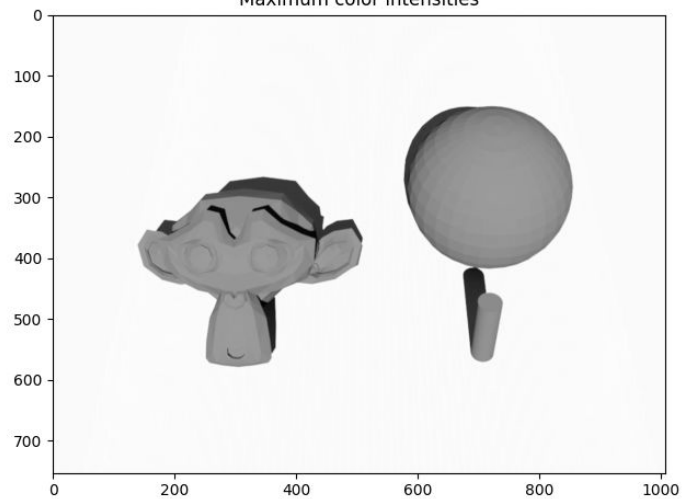
Shadow image



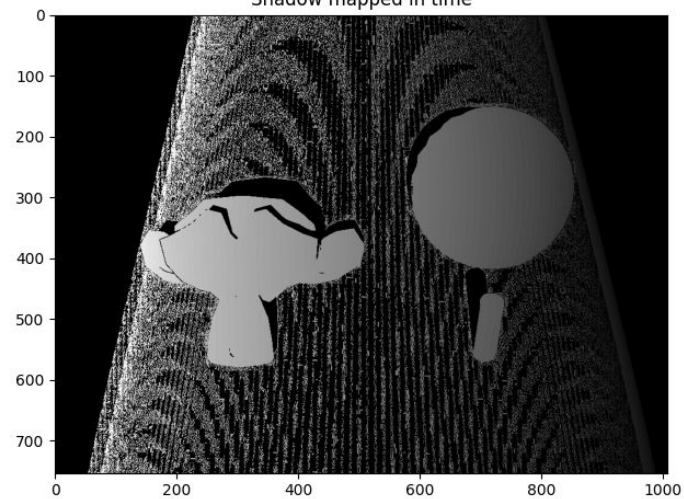
Delta image



Maximum color intensities

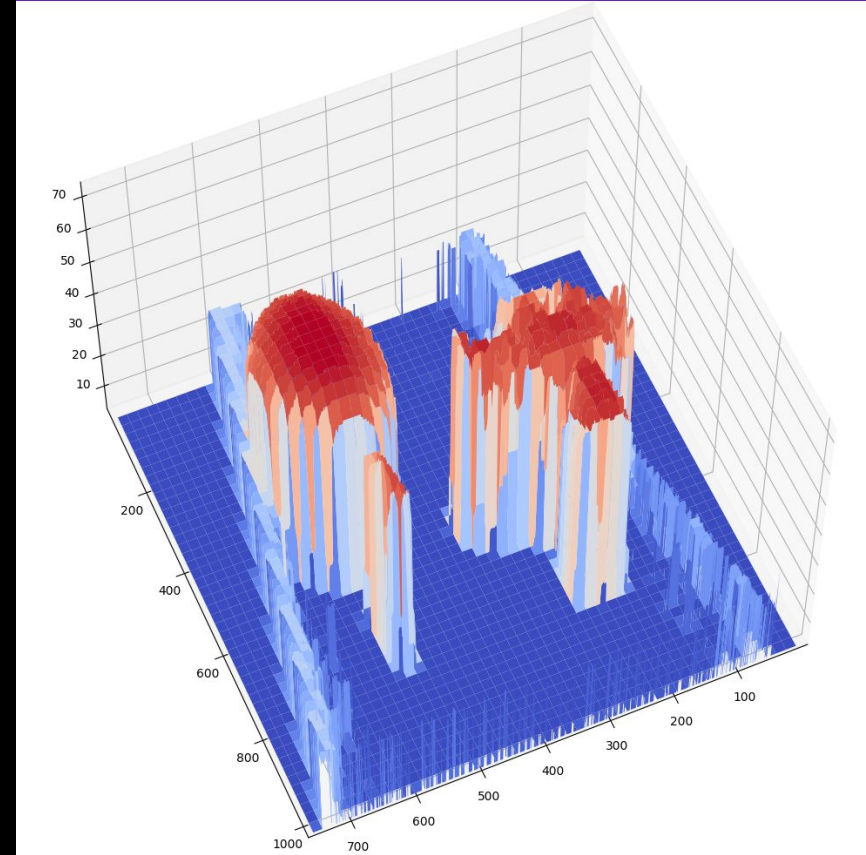
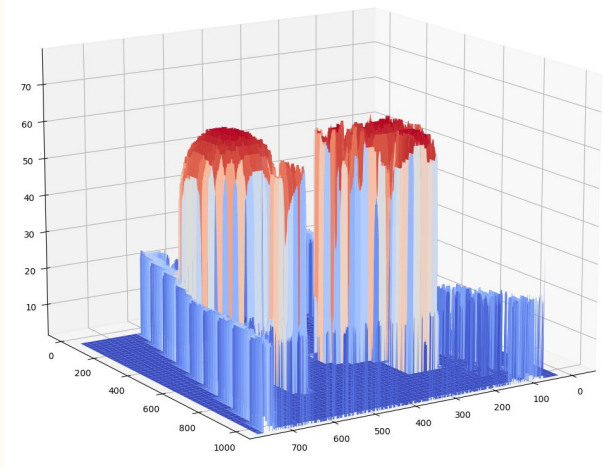


Shadow mapped in time



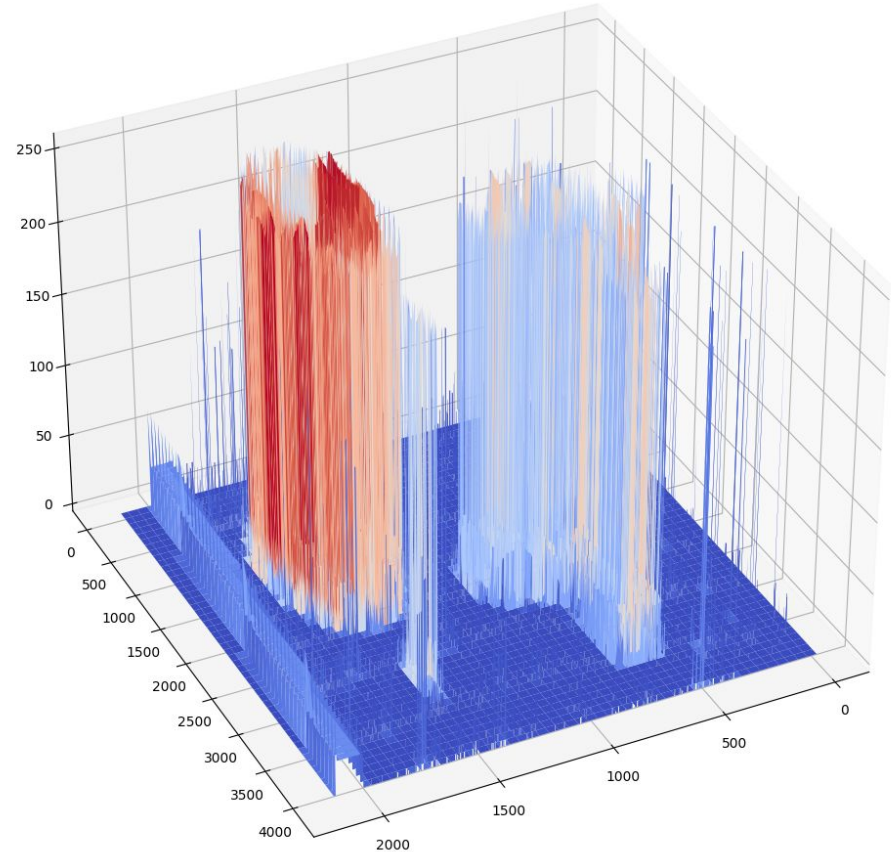
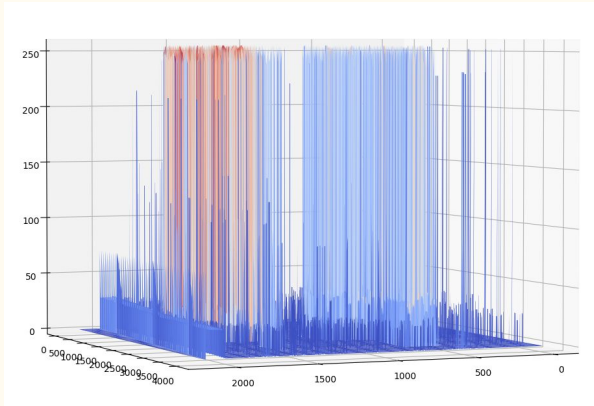
Resultados

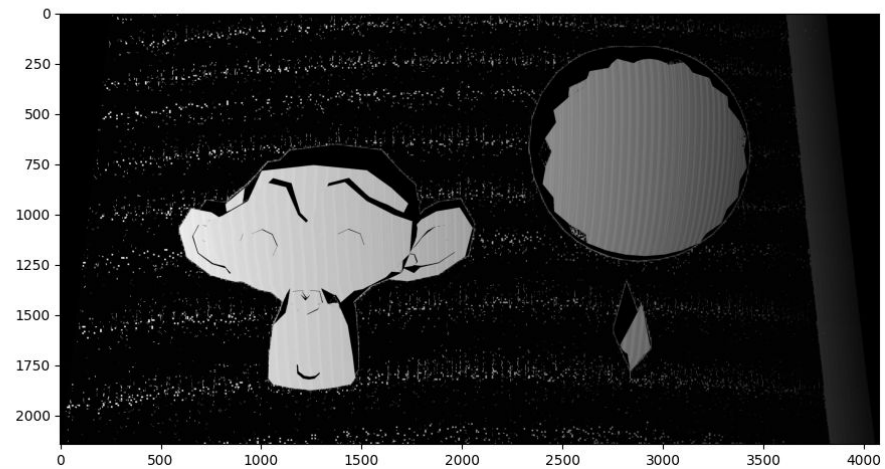
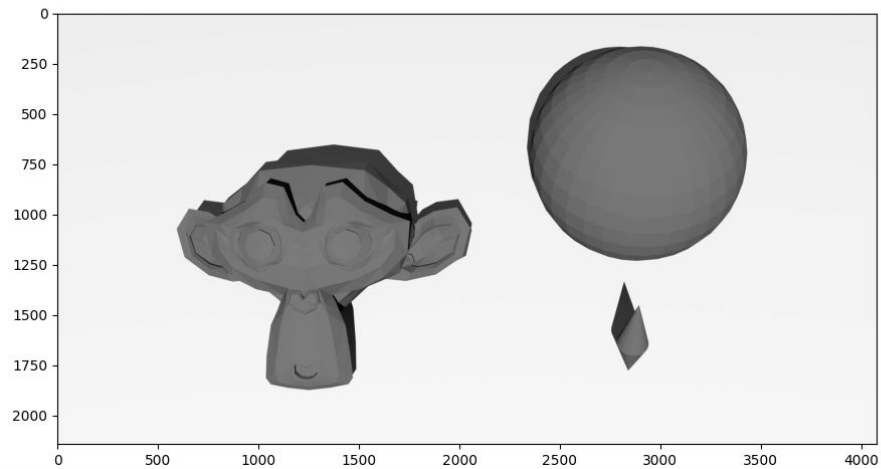
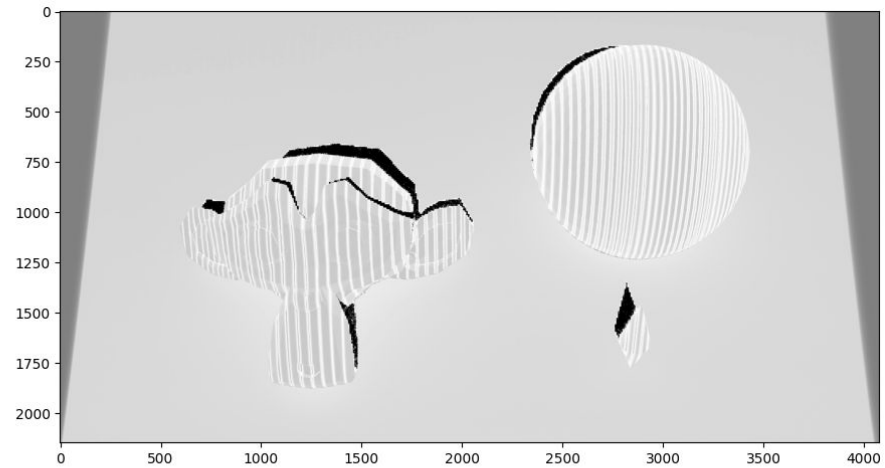
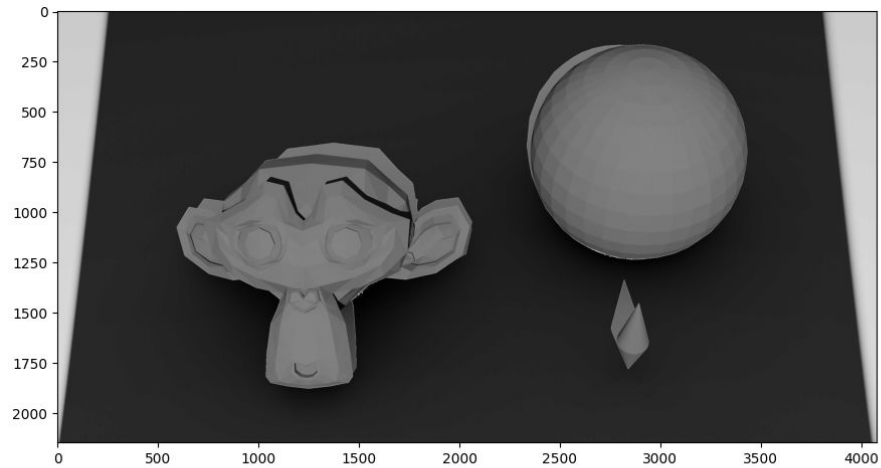
- $Y_{top} = 20\text{px}$
- $Y_{bottom} = 760\text{px}$
- Contrast threshold = 80
- Kernel size = 13
- Camera calibration pattern side = 80mm
- Lamp calibration object height = 70mm



Resultados

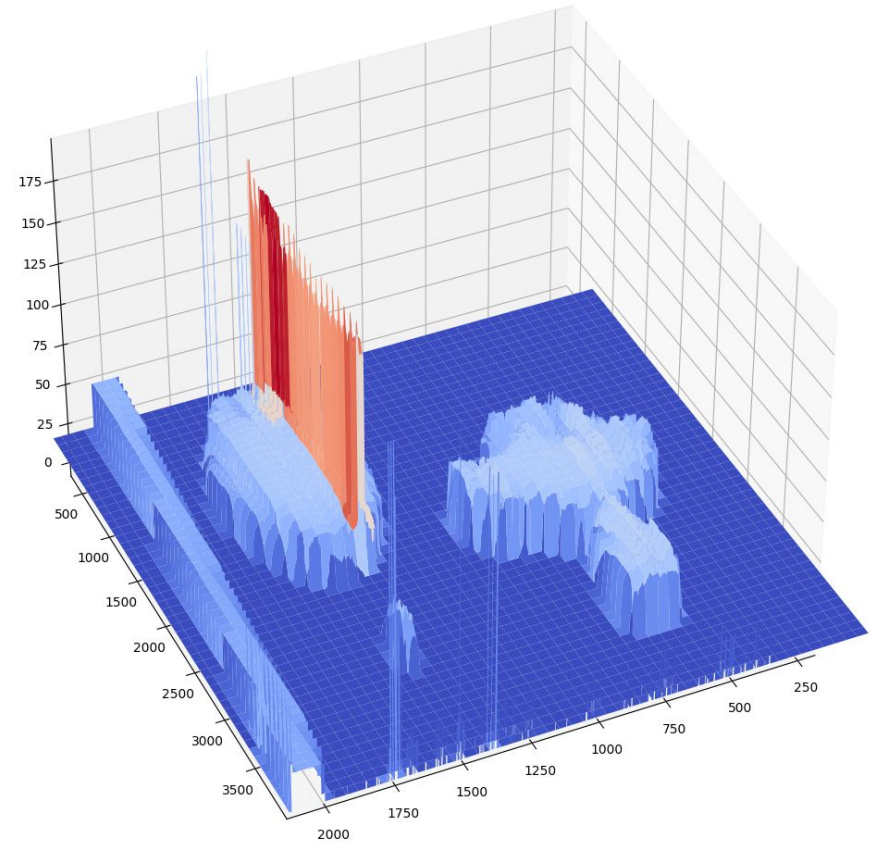
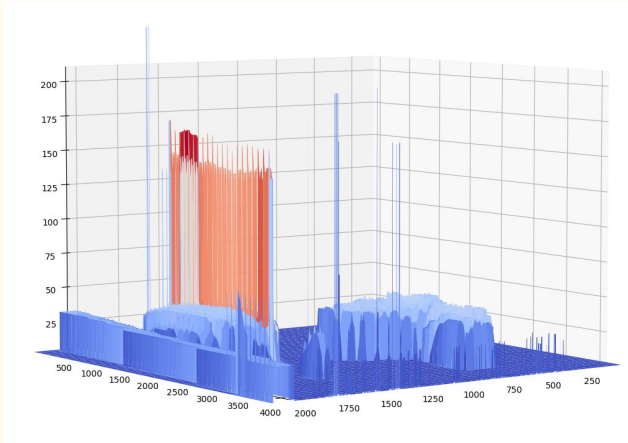
- $Y_{top} = 90\text{px}$
- $Y_{bottom} = 2060\text{px}$
- Contrast threshold = 80
- Kernel size = 1 (sem borramento)
- Camera calibration pattern side = 80mm
- Lamp calibration object height = 70mm





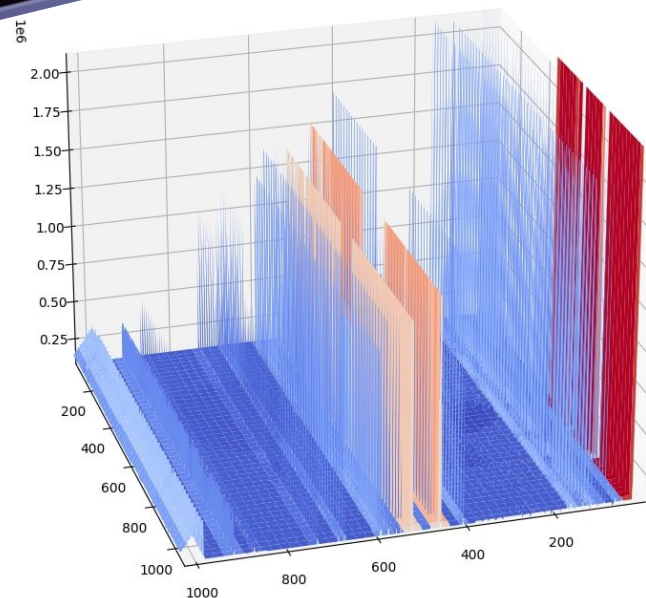
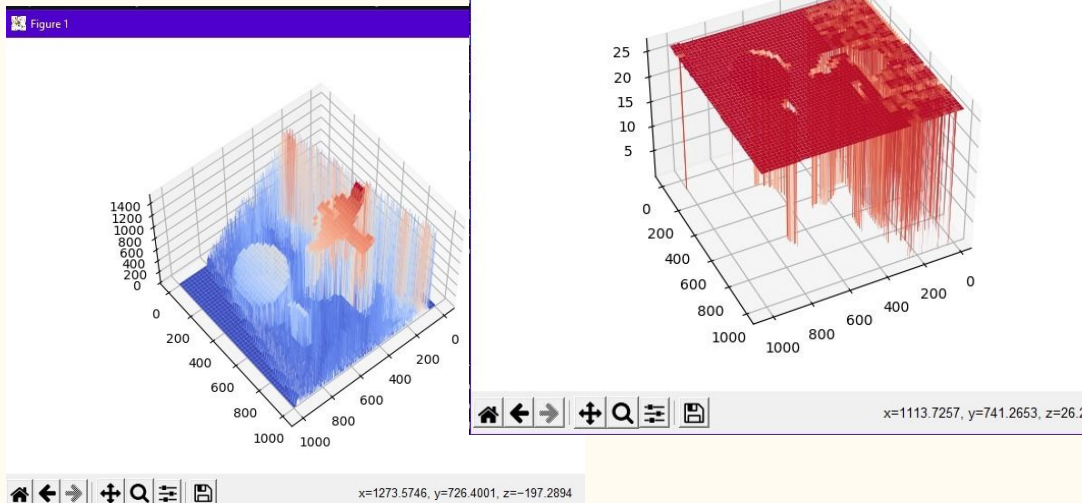
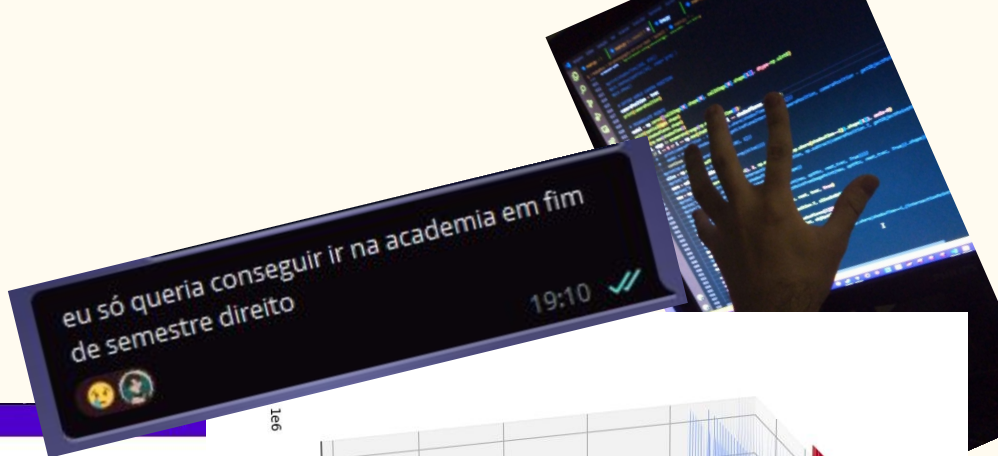
Resultados

- $Y_{top} = 90\text{px}$
- $Y_{bottom} = 2060\text{px}$
- Contrast threshold = 80
- Kernel size = 75
- Camera calibration pattern side = 80mm
- Lamp calibration object height = 70mm



Desafios

- Entender a intuição do problema
- Trabalhar com NumPy com planos e linhas
- Alimentar o Matplotlib direito para gerar malhas apropriadas
- **Python 3.10**



Possibilidades de melhora para o futuro

- Testar com imagens capturadas no mundo real
 - As imagens capturadas até hoje no lab não funcionaram porque o openCV foi incapaz de calibrar a câmera
- Variar o número de frames por scan
- Variar posição das luzes e da câmera com relação ao objeto a ser capturado
- Tentar capturar objetos com textura
- Não usar o matplotlib para imprimir malhas 3d que representem objetos
 - Aprender a usar Open3d

Obrigado!

—