



# Aplicações de técnicas atuais em reconstrução 3D fotogramétrica

Pedro Felipe Pena Barata

Instituto Politécnico do Rio De Janeiro (IPRJ)  
Universidade do Estado do Rio de Janeiro

27 de Novembro de 2017



- 1 Introdução
- 2 Reconstrução a laser
- 3 *Structure from Motion*
- 4 Kinect
- 5 Experimentos
- 6 Conclusão
- 7 Trabalhos futuros



## Introdução

Reconstrução a laser  
Structure from Motion  
Kinect  
Experimentos  
Conclusão  
Trabalhos futuros

## Objetivos

### 1 Introdução

- Objetivos

### 2 Reconstrução a laser

### 3 Structure from Motion

### 4 Kinect

### 5 Experimentos

### 6 Conclusão

### 7 Trabalhos futuros

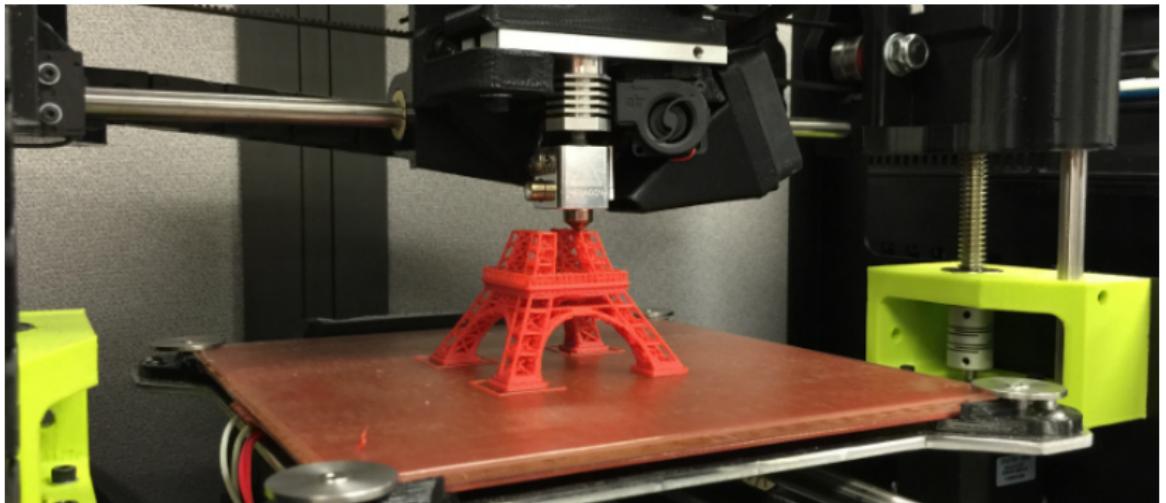


## Introdução

- Reconstrução a laser
- Structure from Motion
- Kinect
- Experimentos
- Conclusão
- Trabalhos futuros

## Objetivos

A reconstrução 3D de cenas gerais a partir de múltiplos pontos de vista, usando-se câmeras convencionais, sem aquisição controlada, é um dos grandes objetivos de pesquisa em visão computacional, ambicioso até mesmo para os dias de hoje.



**Figura: Impressora 3D**

Fonte: <http://publicidademarketing.com/impressora-3d/>

O objetivo deste trabalho é, por meio de técnicas fotogramétricas, preservar o patrimônio cultural do Jardim do Nêgo, localizado na estrada Teresópolis-Friburgo, Rio de Janeiro. Na qual algumas esculturas passaram por um processo de erosão por conta da tragédia de 2011.







Perguntas a serem respondidas ao longo deste projeto:

- ① Que nível de detalhe, facilidade e precisão pode-se obter usando apenas imagens e softwares abertos?
- ② É possível utilizar scanners de baixo custo baseados em Kinect com melhorias significativas em termos de qualidade, conveniência ou tempo de processamento?
- ③ Quais são as restrições desses sistemas?



1 Introdução

2 Reconstrução a laser

- Esculturas de Michelangelo

3 *Structure from Motion*

4 Kinect

5 Experimentos

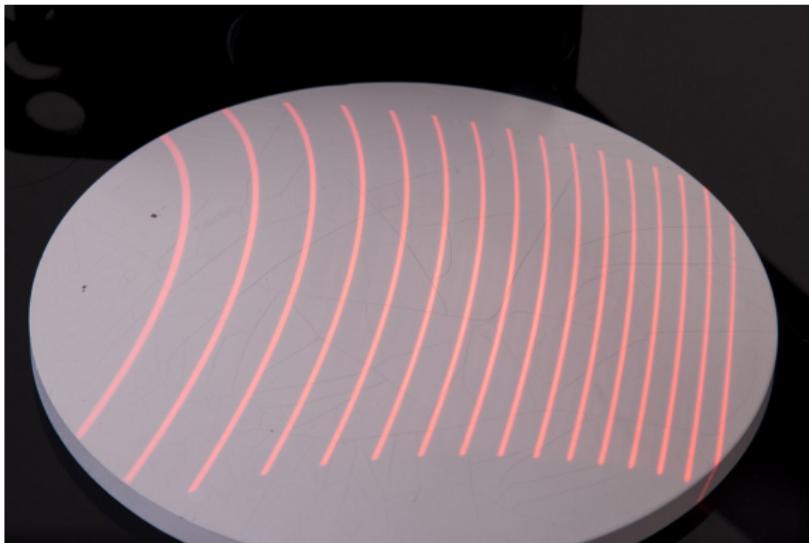
6 Conclusão

7 Trabalhos futuros

O método de reconstrução 3D baseado em *lasers* é um **método óptico ativo**, amplamente utilizado, pois oferece uma alta qualidade geométrica de dados, os resultados são em tempo real e requer pouco tempo de captura de dados. Neste caso, abordaremos o projeto de escaneamento da escultura de Michelangelo, David, que utiliza escaneadores baseados em superfícies, mais especificamente, utilizando *Time of Flight*, ou tempo de voo.



*Time of Flight* é uma técnica de escaneamento baseado em um projetor de padrões (muitas vezes, grades ou barras horizontais, via *laser*) em uma cena a ser reconstruída. A forma de como o padrão se deforma quando atinge superfícies permite que sistemas de visão calculem a profundidade e informações das superfícies dos objetos na cena.



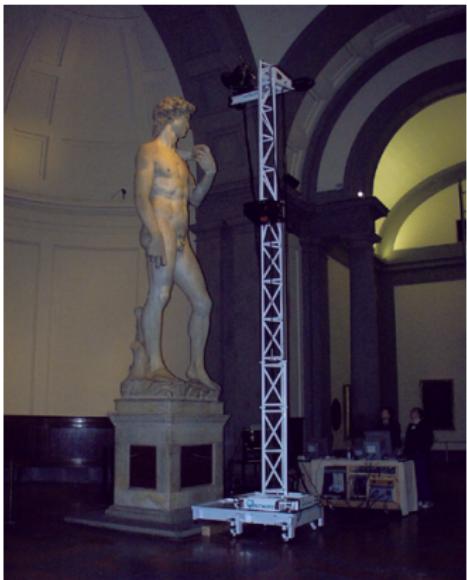
**Figura:** Exemplo de um projetor de padrões de ToF à laser.  
Fonte: <http://www.opto-engineering.com/>



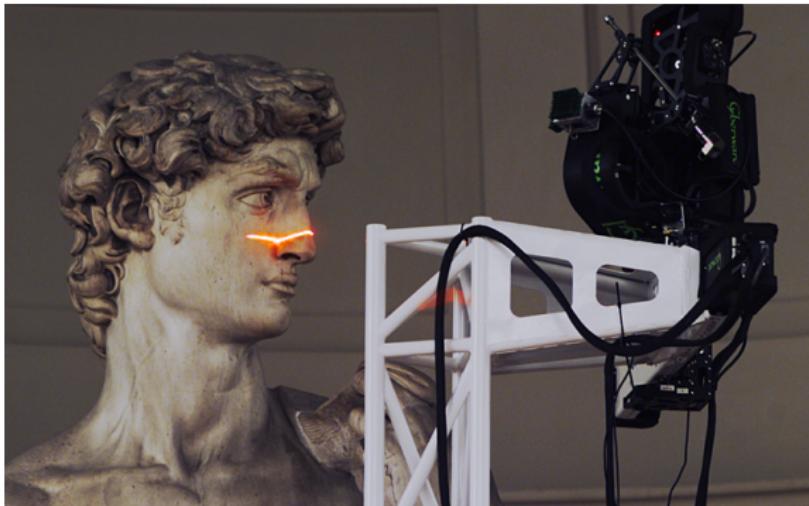
Uma motivação para este trabalho é o projeto da Universidade de Stanford, onde um grupo constituído por mais de 30 professores, funcionários e estudantes, este projeto tem como objetivo preservar as esculturas de Michelangelo, na Itália.



Para isso, este grupo contou com sensores de alcance (*range finders*) para triangulação, sensores de alcance baseados em *Time of Flight*, câmeras digitais e um software de calibração.

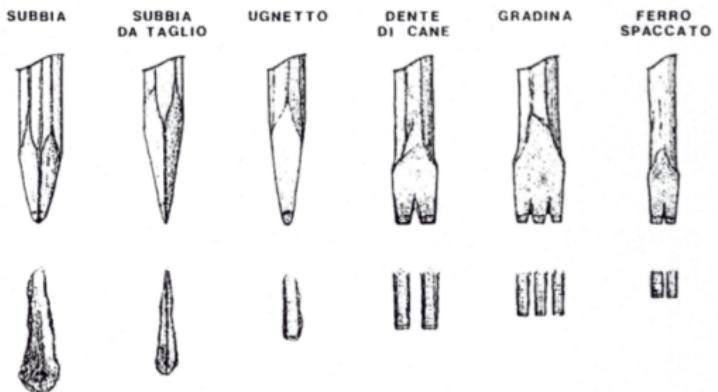


**Figura:** Scanners utilizados no projeto de reconstrução de David.  
Fonte: <http://graphics.stanford.edu/projects/mich/>



**Figura:** Como foi escaneada a cabeça da escultura.

Fonte: <http://graphics.stanford.edu/projects/mich/>



**Figura:** Exemplos de cinzelões usados por Michelangelo  
Fonte: <http://graphics.stanford.edu/projects/mich/>



1 Introdução

2 Reconstrução a laser

3 *Structure from Motion*

- MVE – *Multi-View Stereo Environment*
- VisualSfM

4 Kinect

5 Experimentos

6 Conclusão

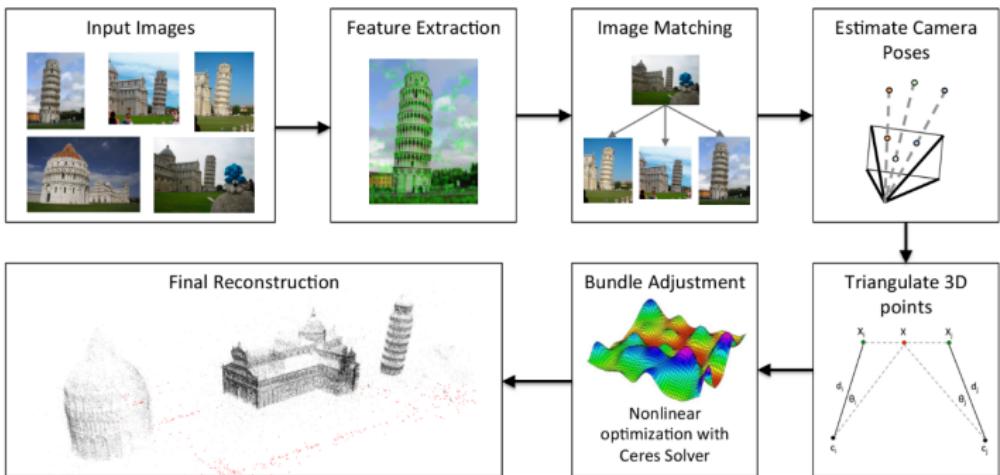
7 Trabalhos futuros



**Structure from motion – SfM** é uma técnica fotogramétrica estéreo multiocular passiva, baseada em estimar a posição de estruturas tridimensionais a partir de sequencias de imagens em duas dimensões.



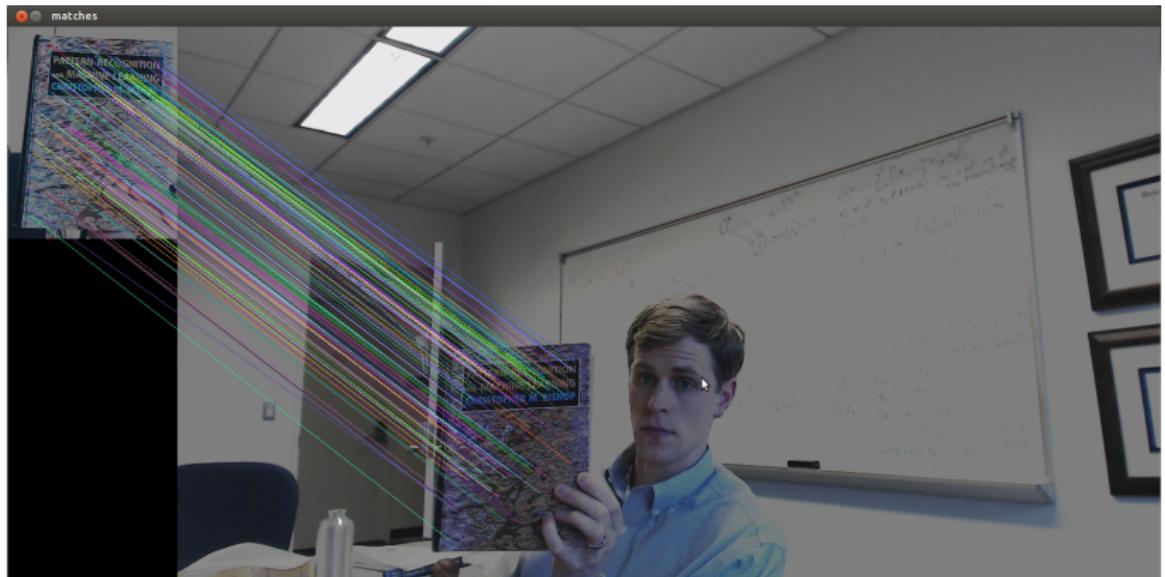
- ① Obtenção de imagens;
- ② Processamento dos parâmetros de câmera para cada imagem;
- ③ Reconstrução da geometria 3D de uma cena com um conjunto de imagens e seus parâmetros correspondentes.



**Figura:** Processo do SfM

Fonte: <http://www.theia-sfm.org/sfm.html>

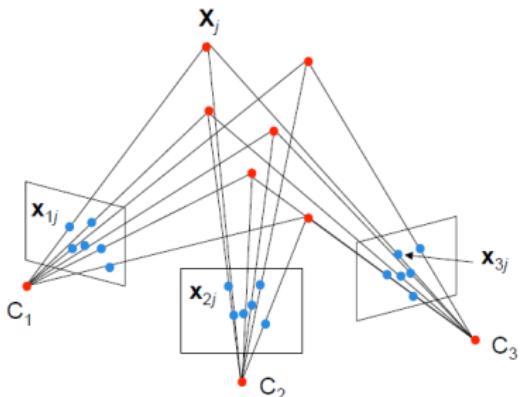
## SIFT – Scale Invariant Feature Transform



**Figura:** Exemplo de interpretação SIFT.

Fonte: <https://www.usna.edu/Users/cs/taylor/courses/si475/index.php?page=class/siftProject.html>

## Triangulação



**Figura:** Uma triangulação utilizando um ponto qualquer,  $X_j$ . Onde cada câmera  $C_1, C_2, C_3$  possui um *feature* correspondente a cada uma delas, respectivamente,  $X_{1j}, X_{2j}, X_{3j}$ .

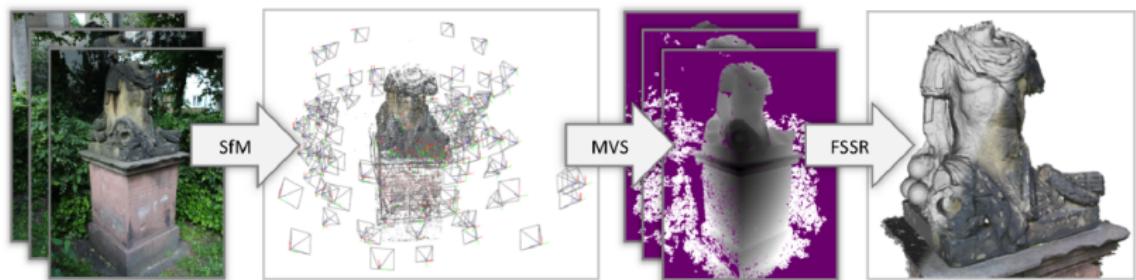
Fonte: [http://cs.nyu.edu/~fergus/teaching/vision/11\\_12\\_multiview.pdf](http://cs.nyu.edu/~fergus/teaching/vision/11_12_multiview.pdf)



## MVE – Multi-View Stereo Environment

- Reconstrução multi-escala
- Possui interface gráfica
- Baseado em mapas de profundidade
- Implementa um algoritmo de reconstrução de superfícies

## MVE – Multi-View Stereo Environment



**Figura:** Processo empregado pelo MVE.

Fonte: <https://www.gcc.tu-darmstadt.de/media/gcc/papers/Fuhrmann-2014-MVE.pdf>

## MVS

- Uma estrutura regional-crescente que tem uma fila de candidatos correspondentes,  $Q$ , ordenada pelas localizações dos pixels na câmera acrescido de seus valores para profundidade e normais;
- Um sistema de correspondências que leva um candidato correspondente como entrada e calcula profundidade, normal e uma confiança de correspondência usando vistas vizinhas fornecidas pela seleção de exibição local.

## MVS



**Figura:** Mapa de profundidade de uma imagem.

Fonte: <https://www.gcc.tu-darmstadt.de/media/gcc/papers/Fuhrmann-2014-MVE.pdf>

## FSSR – Floating Scale Surface Reconstruction

Usa como entrada a união de todos os vértices dos mapas de profundidade.

A partir desses parâmetros ele calcula a representação volumétrica do passo anterior (MVS).

## FSSR – Floating Scale Surface Reconstruction

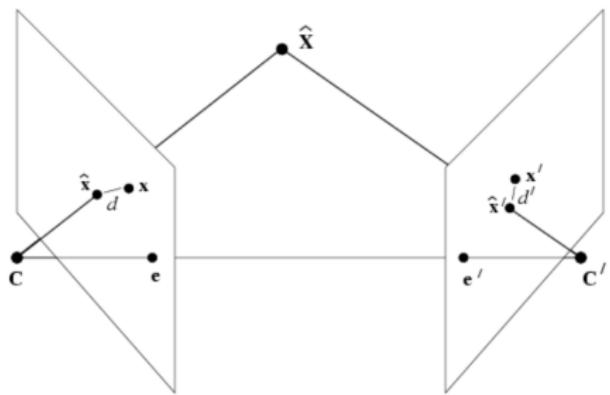
- Utiliza uma função implícita e com escala contínua (flutuante);
- O FSSR não necessita de muitos parâmetros na sua abordagem abordagem;
- A função implícita consegue validar rapidamente os parâmetros de entrada;
- Escalável e não requer nenhuma operação global (cortes ou resolução de grandes sistemas de equações).



## VisualSfM

- Palavra-chave: Escalabilidade
- Possui interface gráfica
- Utiliza programação paralela em CPU e GPU
- Bom gerenciamento dos recursos computacionais

## PBA/MCBA



**Figura:** Exemplo do problema de *Bundle Adjustment*.

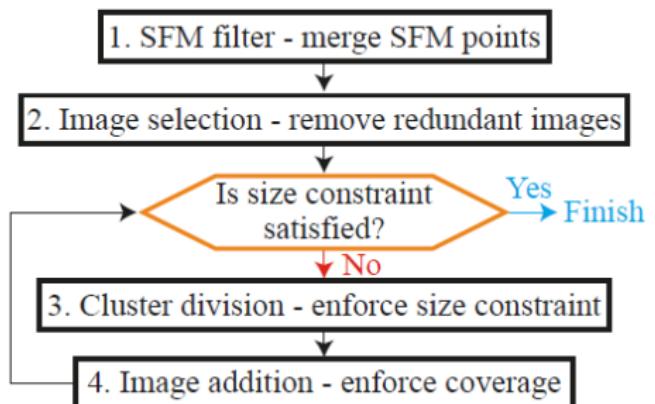
Fonte: <http://av.dfki.de/members/stricker/>



## PBA/MCBA

- Utiliza Levenberg Mardquardt na resolução do *Bundle Adjustment*.
- Multi-núcleos

## CMVS/PMVS-2



**Figura:** Processo do algoritmo CMVS.

Fonte: <http://ieeexplore.ieee.org/abstract/document/5539802/>



- 1 Introdução
- 2 Reconstrução a laser
- 3 *Structure from Motion*
- 4 Kinect
  - Kinect com *Structure from Motion*
- 5 Experimentos
- 6 Conclusão
- 7 Trabalhos futuros



- Baseado em luz estruturada
- Baseado em *Time of Flight*



**Figura:** Kinect V1 aberto, constituído de uma câmera infra-vermelho (IR - *Infra-Red*), uma câmera RGB e um projetor IR. Fonte: <http://cmp.felk.cvut.cz/ftp/articles/pajdla/Smisek-CDC4CV-2011.pdf>



Figura: Kinect V2.

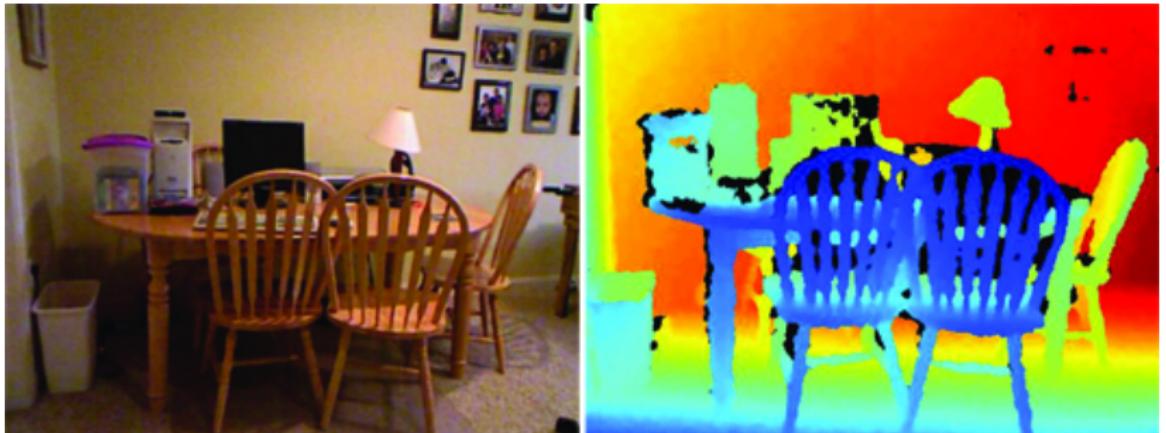


Figura: Saída de uma imagem interpretada pelo Kinect.

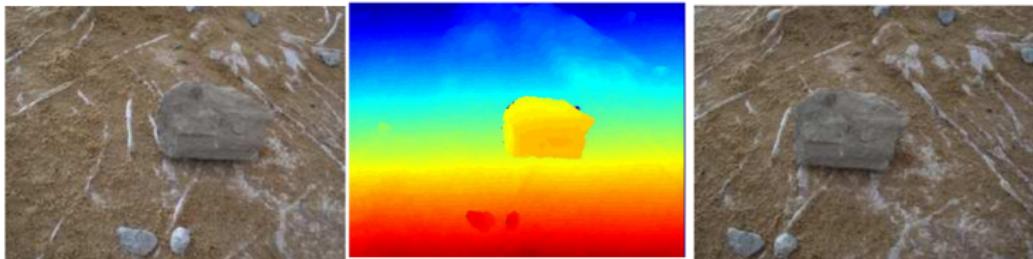


(a) Imagem IR (b) Imagem IR (c) Pontos de iluminada pelo com iluminação calibração projeto de externa reproduzidos na reproduzidos na padrões IR calibração imagem RGB imagem de profundidade

**Figura:** Método de calibração do Kinect, em imagens IR, RGB e de profundidade. Fonte: <http://cmp.felk.cvut.cz/ftp/articles/pajdla/Smisek-CDC4CV-2011.pdf>

Tabela: Resultados dos testes executados

Método	Erro geométrico $e$ [mm]		
	$\mu(e)$	$\sigma(e)$	$\max(e)$
SLR Stereo	1,57	1,15	7,38
Kinect	2,39	1,67	8,64
SR-4000	27,62	18,20	133,85



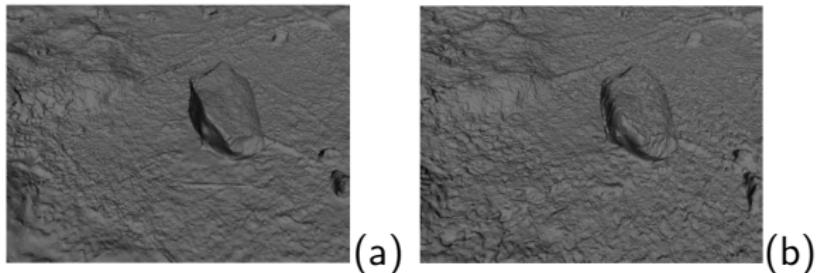
(a) Imagem do lado esquerdo da câmera SLR

(b) Imagem da profundidade do Kinect

(c) Imagem do lado direito da câmera SLR

**Figura:** Imagens iniciais para reconstrução 3D.

Fonte: <http://cmp.felk.cvut.cz/ftp/articles/pajdla/Smisek-CDC4CV-2011.pdf>



**Figura:** Resultados da reconstrução SfM, onde (a) é a reconstrução do SLR e (b) o resultado do Kinect.

Fonte: <http://cmp.felk.cvut.cz/ftp/articles/pajdla/Smisek-CDC4CV-2011.pdf>



1 Introdução

2 Reconstrução a laser

3 *Structure from Motion*

4 Kinect

5 Experimentos

- Escultura do Jardim do Nêgo
- Objeto em ambiente fechado

6 Conclusão

7 Trabalhos futuros

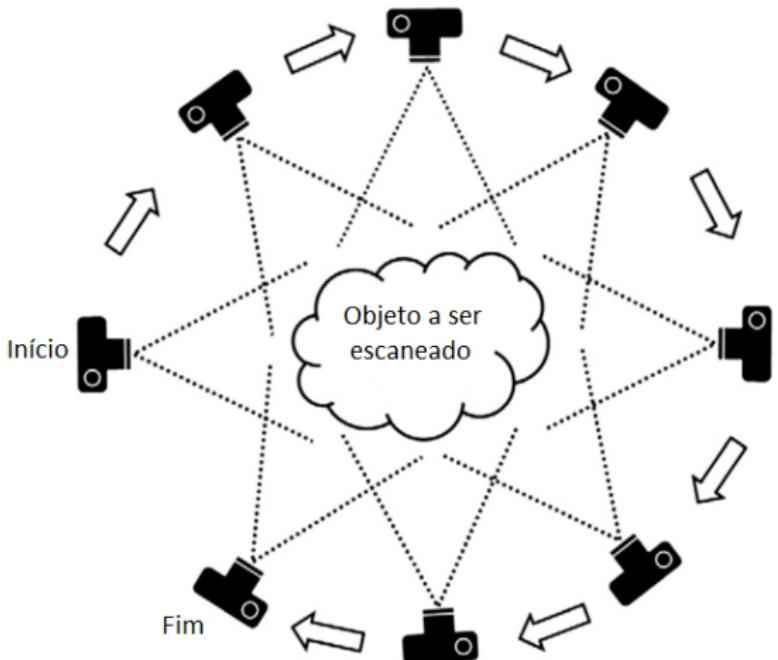


Figura: Exemplo de como foi realizada a varredura da escultura

Seguintes conjuntos de dados:

- 2 vídeos de uma escultura (sapo), resultando em 280 imagens (MVE);
- 1 vídeo de uma esculutra (índio), resultando em 197 imagens (VisualSfM);
- 2 vídeos em ambiente fechado, com um total de 224 imagens (MVE e VisualSfM).

## Escultura com VisualSfM



**Figura:** Botões na parte superior da interface gráfica, este seria o procedimento padrão de funcionamento do software. **Fonte:** <http://ccwu.me/vsfm/>



Introdução  
Reconstrução a laser  
Structure from Motion  
Kinect  
**Experimentos**  
Conclusão  
Trabalhos futuros

**Escultura do Jardim do Nêgo**  
Objeto em ambiente fechado

## Escultura com VisualSfM

Assistir o video do jardim do Nêgo

## Escultura com VisualSfM

**Tabela:** Tempos obtidos da reconstrução da escultura do Jardim do Nêgo usando o VisualSfM

Procedimento	Tempo (aprox.)
Carregamento de imagens	10 segundos
Calcular pares correspondentes de <i>features</i>	6.643 segundos
Gerar a reconstrução esparsa do modelo	220 segundos
Gerar a reconstrução densa do modelo	1.385 segundos

## Escultura com VisualSfM



Figura: Reconstrução esparsa da escultura do Jardim do Nêgo no VisualSfM com 197 imagens.

## Escultura com VisualSfM



Figura: Resultados da reconstrução densa da escultura do Jardim do Nêgo usando o VisualSfM.

## Escultura com VisualSfM

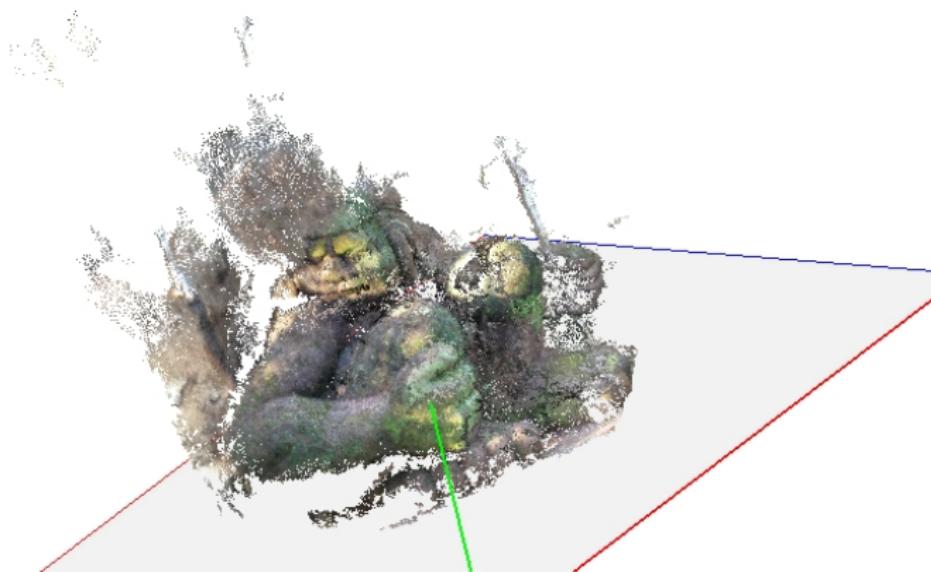


Figura: Resultados da reconstrução densa da escultura do Jardim do Nêgo usando o VisualSfM, em outro ângulo.

# MVE

Comandos utilizados:

- ① Cria-se uma cena e adiciona-se imagens a ela;
- ② *sfmrecon*;
- ③ *dmrecon*;
- ④ *scene2pset*;
- ⑤ *fssrecon*;
- ⑥ *meshclean*.



Introdução  
Reconstrução a laser  
Structure from Motion  
Kinect  
**Experimentos**  
Conclusão  
Trabalhos futuros

**Escultura do Jardim do Nêgo**  
Objeto em ambiente fechado

## Escultura com MVE

Assistir o primeiro video do jardim do Nêgo



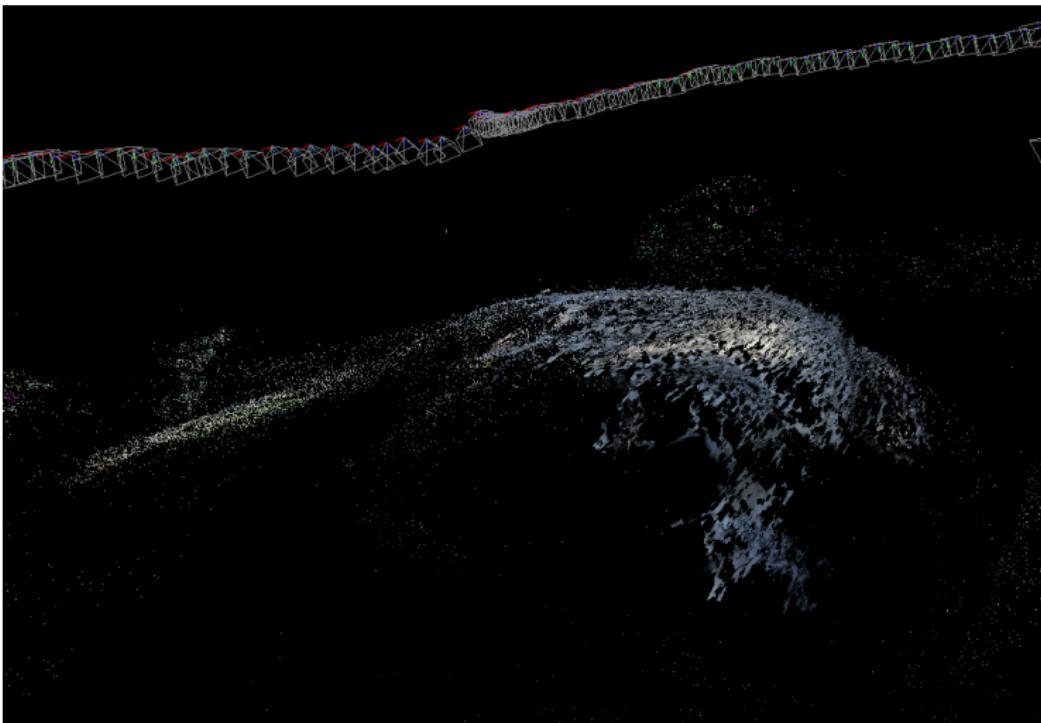
Introdução  
Reconstrução a laser  
Structure from Motion  
Kinect  
**Experimentos**  
Conclusão  
Trabalhos futuros

**Escultura do Jardim do Nêgo**  
Objeto em ambiente fechado

## Escultura com MVE

Assistir o segundo video do jardim do Nêgo

## Escultura com MVE



## Escultura com MVE

Tabela: Tempos obtidos usando o MVE em um conjunto de dados do Jardim do Nêgo

Comando	Tempo (aprox.)
<i>sfmrecon</i>	78 segundos
<i>dmrecon</i>	14.503 segundos
<i>scene2pset</i>	600 segundos
<i>fssrecon</i>	25.293 segundos
<i>meshclean</i>	60 segundos

## Escultura com MVE

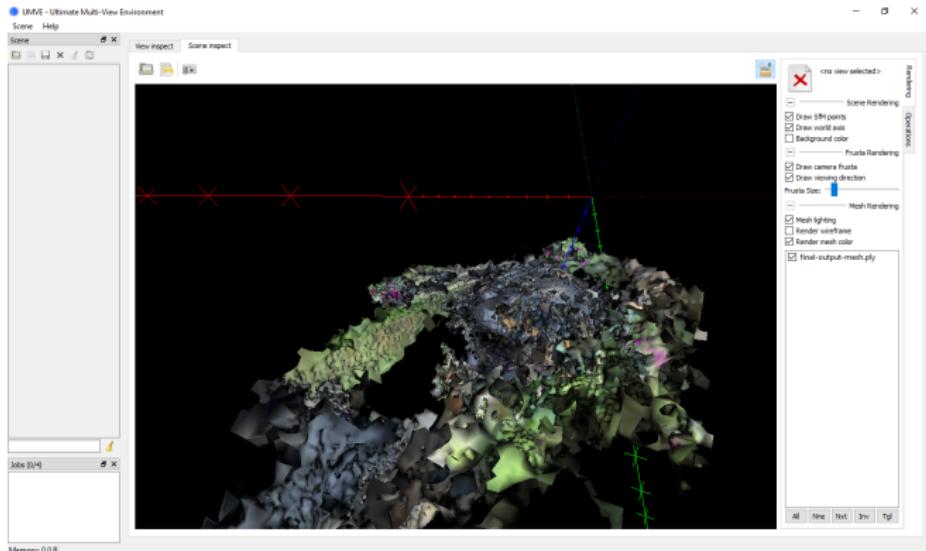


Figura: Malha com ruídos proveniente do comando *fssrecon*.

## Escultura com MVE

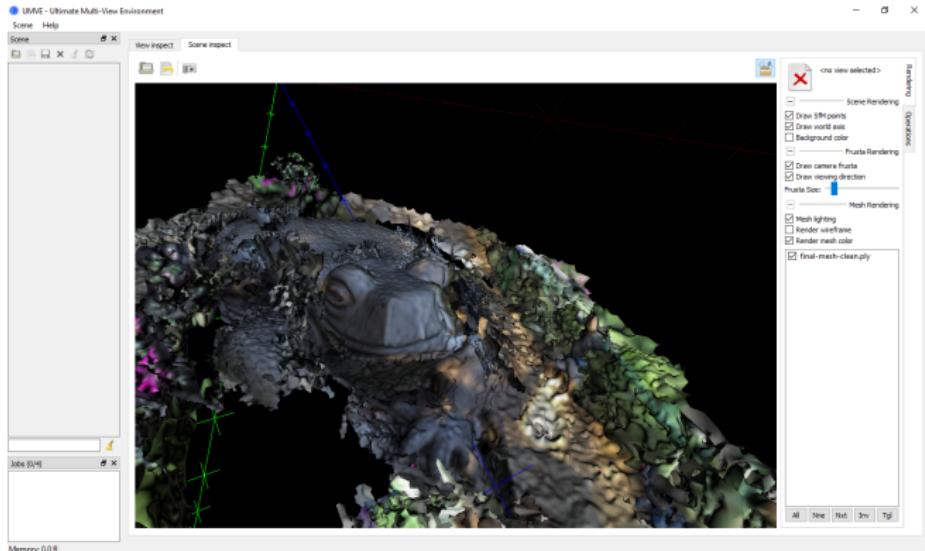


Figura: Resultado final, após a remoção dos ruídos da malha.



Introdução  
Reconstrução a laser  
Structure from Motion  
Kinect  
**Experimentos**  
Conclusão  
Trabalhos futuros

Escultura do Jardim do Nêgo  
**Objeto em ambiente fechado**

Assistir o primeiro video em ambiente fechado



Introdução  
Reconstrução a laser  
Structure from Motion  
Kinect  
**Experimentos**  
Conclusão  
Trabalhos futuros

Escultura do Jardim do Nêgo  
**Objeto em ambiente fechado**

Assistir o segundo video em ambiente fechado

## Objeto com 200 imagens usando o VisualSfM

Tabela: Tempos obtidos da reconstrução do objeto com 200 imagens, usando o VisualSfM

Procedimento	Tempo (aprox.)
Carregamento de imagens	50 segundos
Calcular pares correspondentes de <i>features</i>	9.540 segundos
Gerar a reconstrução esparsa do modelo	135 segundos
Gerar a reconstrução densa do modelo	1.416 segundos

## Objeto com 200 imagens usando o VisualSfM

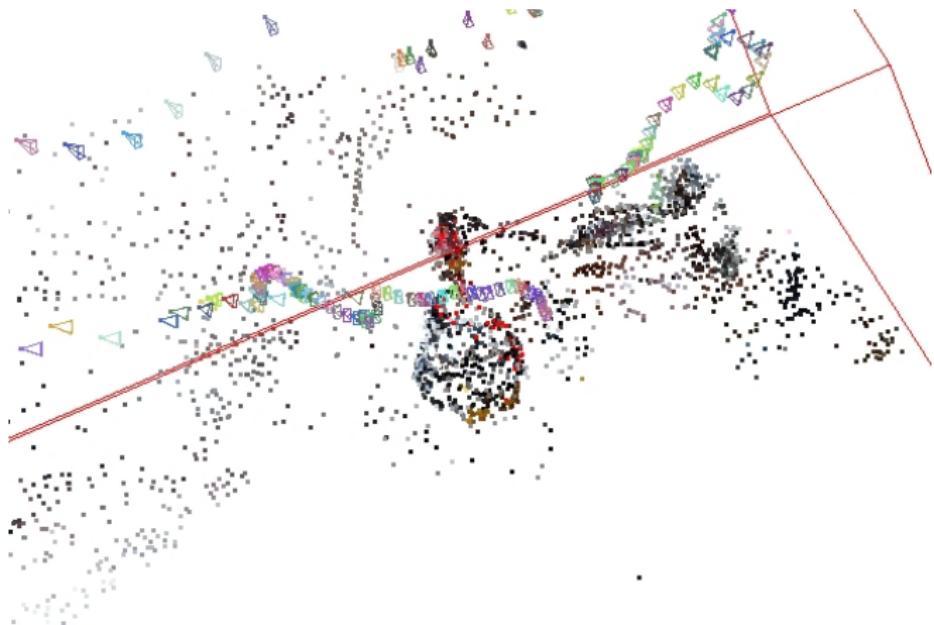


Figura: Reconstrução esparsa do objeto no VisualSfM com 200 imagens.

## Objeto com 200 imagens usando o VisualSfM



Figura: Reconstrução densa do objeto no VisualSfM com 200 imagens.

## Objeto com 224 imagens usando o VisualSfM

Tabela: Tempos obtidos da reconstrução do objeto, com 224 imagens usando o VisualSfM

Procedimento	Tempo (aprox.)
Carregamento de imagens	60 segundos
Calcular pares correspondentes de <i>features</i>	10.451 segundos
Gerar a reconstrução esparsa do modelo	162 segundos
Gerar a reconstrução densa do modelo	1920 segundos

## Objeto com 224 imagens usando o VisualSfM



Figura: Reconstrução esparsa do objeto com 224 imagens no VisualSfM.

## Objeto com 224 imagens usando o VisualSfM

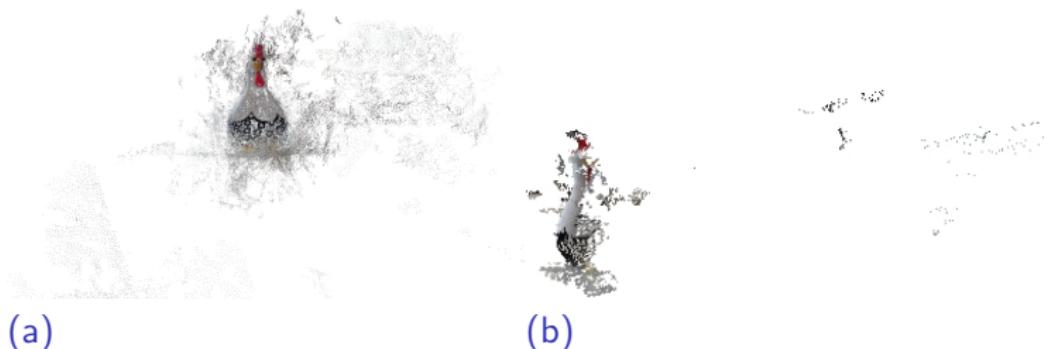


Figura: Reconstruções densas do primeiro (a) e do segundo (b) modelo do objeto no VisualSfM com 224 imagens.

## Objeto com 200 imagens usando o MVE

Tabela: Tempos obtidos usando o MVE em um conjunto de dados em ambiente interno com 200 imagens

Comando	Tempo (aprox.)
<i>sfmrecon</i>	371 segundos
<i>dmrecon</i>	3.716 segundos
<i>scene2pset</i>	300 segundos
<i>fssrecon</i>	1.695 segundos
<i>meshclean</i>	45 segundos

## Objeto com 200 imagens usando o MVE

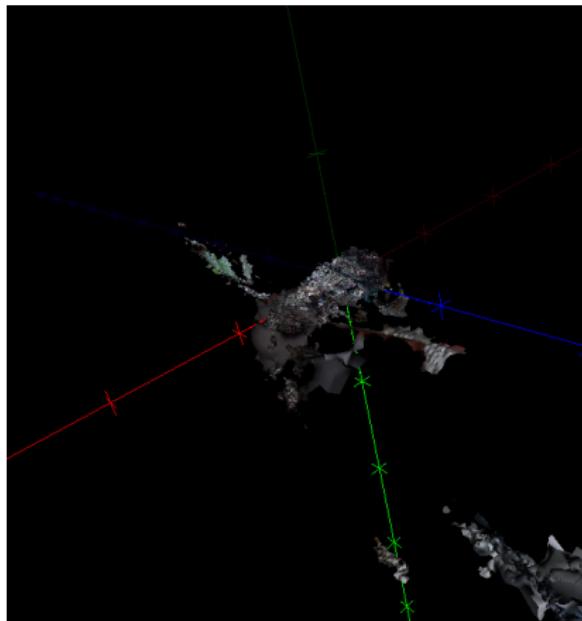


Figura: Resultado da etapa *fssrecon* do MVE

## Objeto com 200 imagens usando o MVE

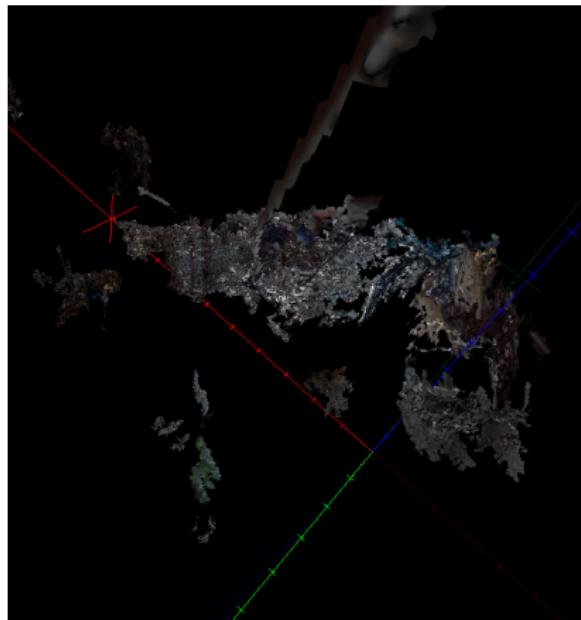


Figura: Resultado da etapa *meshclean*, da etapa anterior 31.

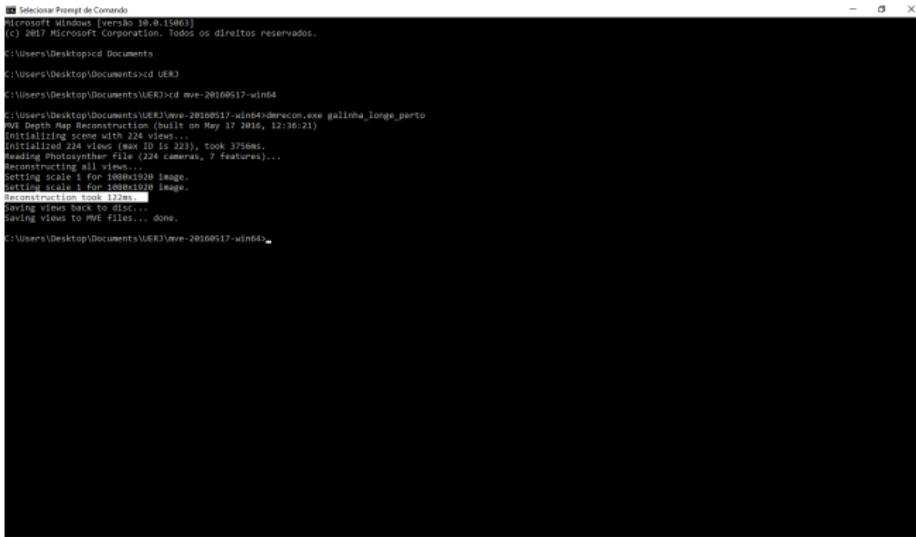


## Objeto com 224 imagens usando o MVE

```
[1] Selecionar Prompt de Conversão
Pair 166,167: 156 matches, 160 H-inliers (84%) 0.0156674
Pair 166,168: 156 matches, 160 H-inliers (84%) 0.0156674
Pair 183,184: 182 matches, 131 H-inliers (66%)
Pair 182,183: 184 matches, 134 H-inliers (72%)
Pair 88,90: 182 matches, 135 H-inliers (74%)
Pair 182,185: 182 matches, 134 H-inliers (74%)
Pair 21,22: 180 matches, 134 H-inliers (63%)
Pair 16,17: 179 matches, 154 H-inliers (88%)
Pair 107,109: 179 matches, 134 H-inliers (74%)
Pair 107,110: 179 matches, 134 H-inliers (74%)
Pair 108,110: 178 matches, 128 H-inliers (71%)
Pair 90,92: 177 matches, 161 H-inliers (90%)
Pair 16,18: 173 matches, 132 H-inliers (76%)
Pair 183,187: 173 matches, 132 H-inliers (76%)
Pair 184,185: 173 matches, 123 H-inliers (71%)
Pair 28,21: 173 matches, 121 H-inliers (69%)
Pair 34,34: 173 matches, 120 H-inliers (75%)
Pair 37,38: 173 matches, 120 H-inliers (75%)
Pair 9,10: 171 matches, 144 H-inliers (84%)
Pair 87,88: 172 matches, 151 H-inliers (88%)
Pair 100,101: 170 matches, 137 H-inliers (70%)
Pair 89,90: 169 matches, 162 H-inliers (95%)
Pair 27,29: 166 matches, 124 H-inliers (74%)
Pair 112,113: 166 matches, 143 H-inliers (80%)
Pair 76,27: 165 matches, 134 H-inliers (69%)
Pair 112,113: 165 matches, 143 H-inliers (89%)
Pair 189,110: 164 matches, 135 H-inliers (70%)
Pair 189,111: 164 matches, 145 H-inliers (88%)
Pair 152,194: 164 matches, 138 H-inliers (67%)
Pair 182,184: 164 matches, 161 H-inliers (61%)
Pair 172,174: 163 matches, 67 H-inliers (41%), 28.59 pair angle
Pair 172,176: 163 matches, 67 H-inliers (41%)
Pair 118,111: 160 matches, 145 H-inliers (88%)
Pair 24,25: 159 matches, 104 H-inliers (68%)
Using views 172 and 174 as initial pair.
Rejected 158 tracks due to bad tracks.
Rejected 158 tracks with large error.
Rejected 6 tracks behind cameras.
Deleted 0 or 7 tracks (0.00%) above a threshold of 0.0156674.
PBL: PSE 3.70908e-05 -> 9.3718e-09 (M), 8 Iter., 322ms.
Camera 172, focal length: 1 -> 1.1118, distortion: -11.5434
Camera 174, focal length: 1 -> 0.898429, distortion: 5.0591
SFM reconstruction time: 24127 ms.
SFM reconstruction loss: 24127 ms.
Creating bundle data structure...
Writing bundle (224 cameras) to file: galinha_longe_perto/synth.got...
Writing view (224 views) to file: C:/Users/Desktop/Documents/UFSC/mve-20160517-wind6/galinha_longe_perto/views/view_0174.mve
Saving view to: C:/Users/Desktop/Documents/UFSC/mve-20160517-wind6/galinha_longe_perto/views/view_0172.mve
C:/Users/Desktop/Documents/UFSC/mve-20160517-wind6
```

Figura: Resultado da etapa *sfmrecon*, com todas as imagens.

## Objeto com 224 imagens usando o MVE



```
Seletor de Projeto de Comando
Microsoft Windows [versão 10.0.15063]
(c) 2017 Microsoft Corporation. Todos os direitos reservados.
C:\Users\Deskt...pt>cd Documents
C:\Users\Deskt...pt>cd UERJ
C:\Users\Deskt...pt>cd mve-20160517-win64
C:\Users\Deskt...pt>dmrecon.exe gallinha_longo_perto
dmrecon 1.0.0 - (c) 2016 Microsoft Corporation. All rights reserved.
Build on May 17 2016, 12:36:21
Initializing scene with 224 views.
Initialized 224 views (max ID is 223), took 375ms.
Reading Photosynthner file (224 cameras, 7 features)...
Reconstructing 224 views.
Reconstructing 1 full 1080x1920 image.
Setting scale 1 for 1080x1920 image.
Reconstruction took 127ms.
Saving views back to disk...
Saving views back to disk...
Saving views to PFM files... done.
C:\Users\Deskt...pt>
```

Figura: Resultado da etapa *dmrecon*, com todas as imagens.



1 Introdução

2 Reconstrução a laser

3 *Structure from Motion*

4 Kinect

5 Experimentos

6 Conclusão

7 Trabalhos futuros



Constatamos, que através de métodos *Structure from Motion* combinados com programas de código livre e gratuitos, é possível, a partir de uma câmera comum de celular, obtermos reconstruções 3D qualitativamente satisfatórias.



**1** Introdução

**2** Reconstrução a laser

**3** *Structure from Motion*

**4** Kinect

**5** Experimentos

**6** Conclusão

**7** Trabalhos futuros



- 1 Realizar uma varredura com o Kinect.
- 2 Validação adicional.
- 3 Constatar na prática, o melhor método de varredura da escultura.
- 4 Concretizar o objetivo proposto neste trabalho.

**Obrigado!  
Perguntas?**