

Departamento de Sistemas e Computação – FURB
Curso de Ciência da Computação
Trabalho de Conclusão de Curso – 2014/1

MOLLIOREM: RECONSTRUÇÃO DO TÓRAX FEMININO

Acadêmico: Marina Uliano

Marina.miyu@gmail.com

Orientador: Prof. Aurélio Hoppe

aurelio.hoppe@gmail.com

<http://www.inf.furb.br/~aurelio/>

Grupo de Pesquisa em Computação
Gráfica, Processamento de Imagens e
Entretenimento Digital
<http://www.inf.furb.br/gcg>



Roteiro

- Motivação
- Fundamentação
- Objetivos
- Trabalhos relacionados
- Requisitos
- Diagrama de atividade
- Implementação
- Resultados
- Conclusão
- Demonstração

Motivação

- Segundo CRMMedic (2010), as principais dúvidas das mulheres em relação a mamoplastia são:
 - posso ter alergia ao silicone
 - a prótese pode sair do lugar
 - colocar implantes atrás ou na frente do músculo
 - qual o tamanho da prótese que se deve

Objetivos

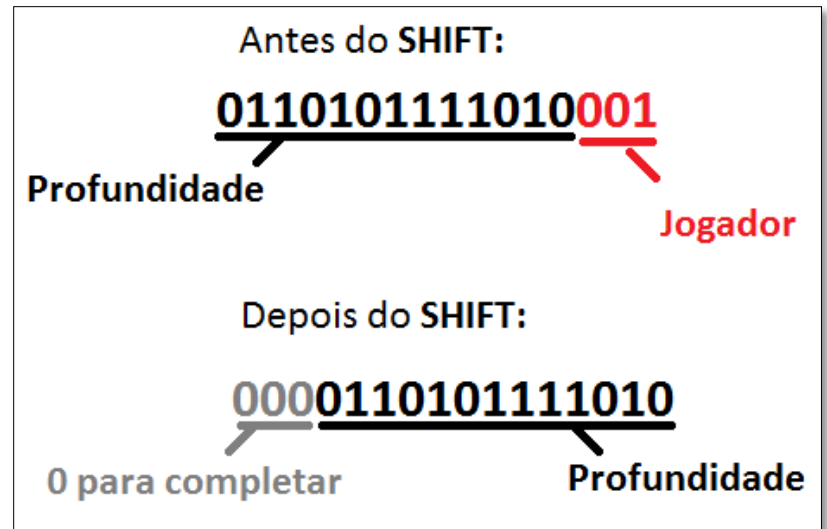
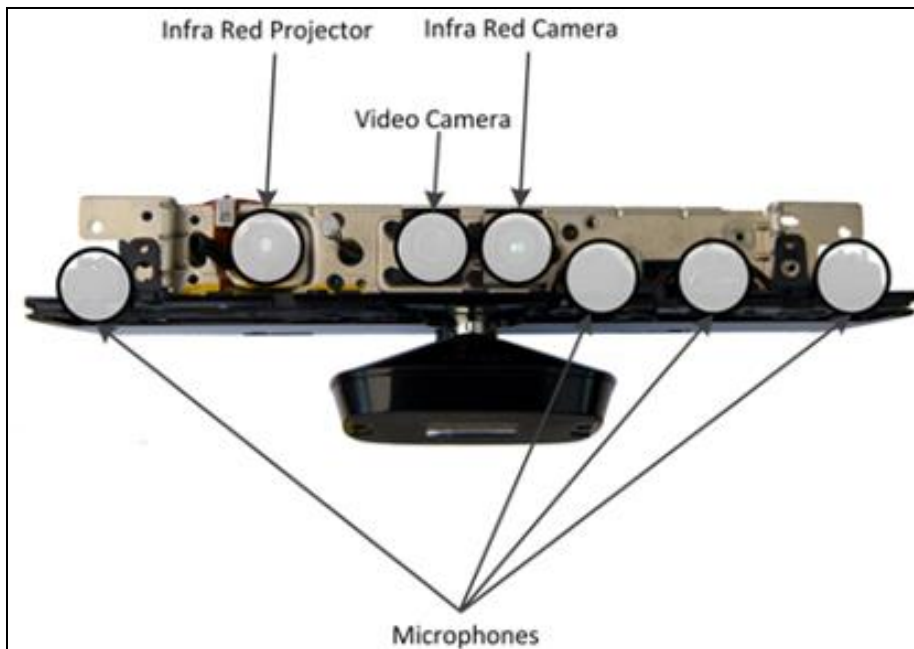
Realizar a reconstrução 3D de um tórax feminino a partir de uma nuvem de pontos.

Os objetivos específicos:

- utilizar o Microsoft Kinect como *scanner* 3D para adquirir a nuvem de pontos;
- utilizar a biblioteca PCL para realizar a manipulação das nuvens de pontos, obtendo no final a reconstrução 3D.

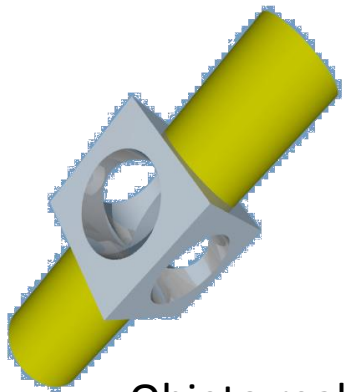
Fundamentação

- Microsoft Kinect como *scanner 3D*

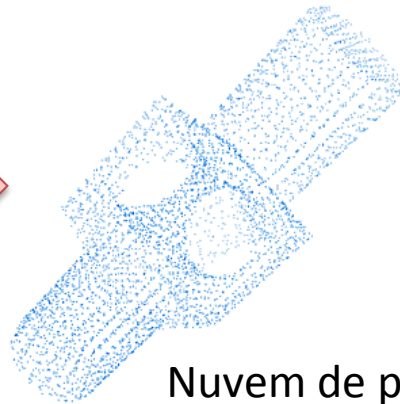


Fundamentação

Nuvem de pontos



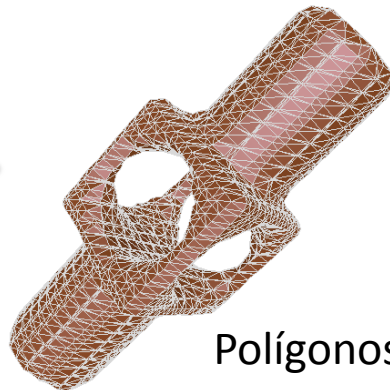
Objeto real



Nuvem de pontos



Estimativa de normal



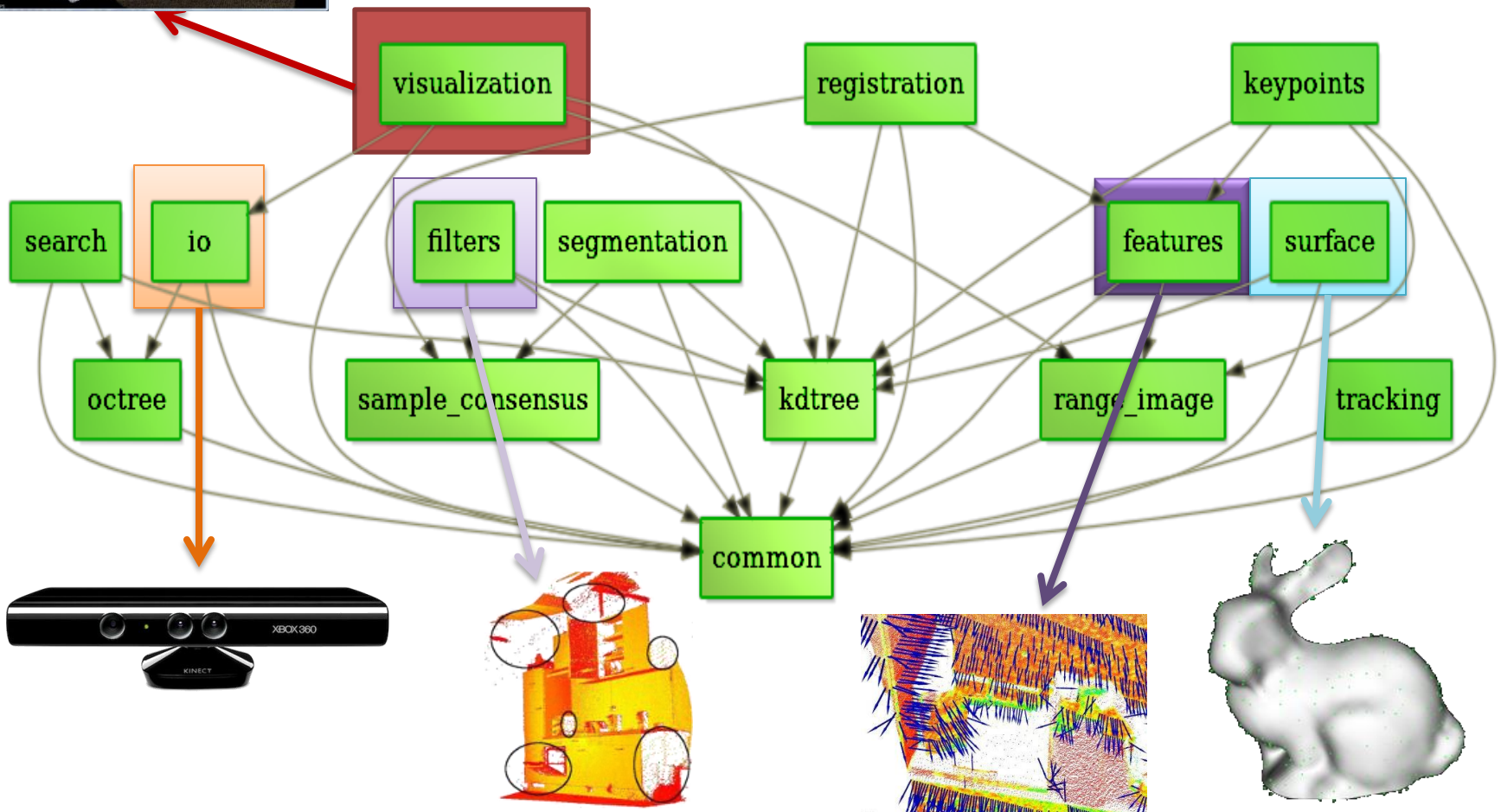
Polígonos



Superfícies reconstruída

Fundamentação

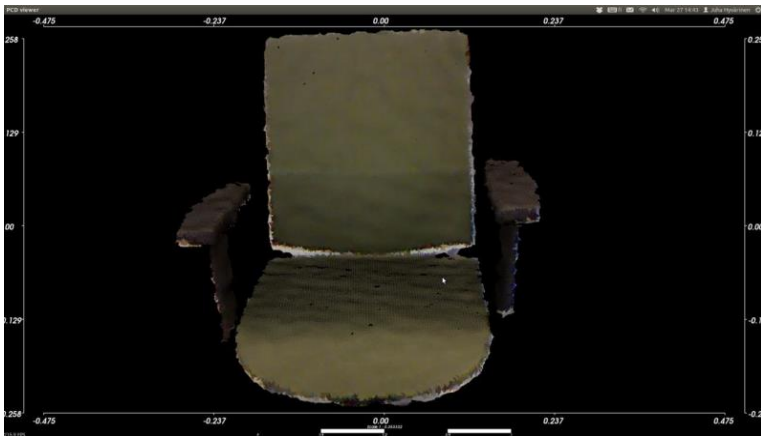
Point Cloud Library (PCL)



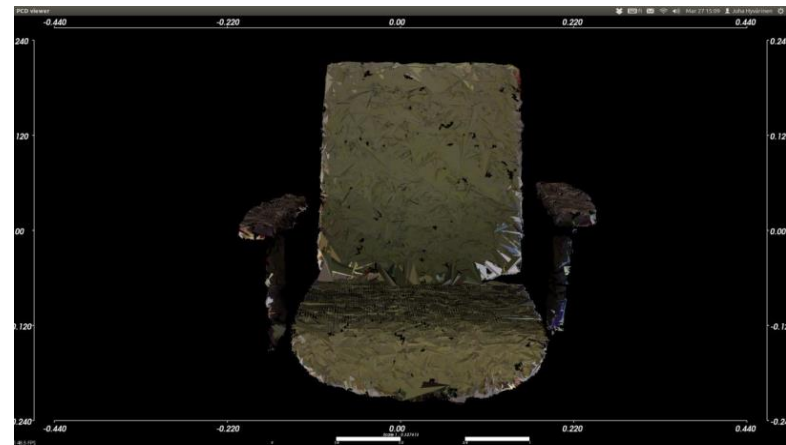
Surface reconstruction of point clouds captured with Microsoft Kinect (HYVARINEN, 2011)

Objetivo:

- Reconstrução 3D
- Exportar malha de polígonos para outros editores (realXtend)

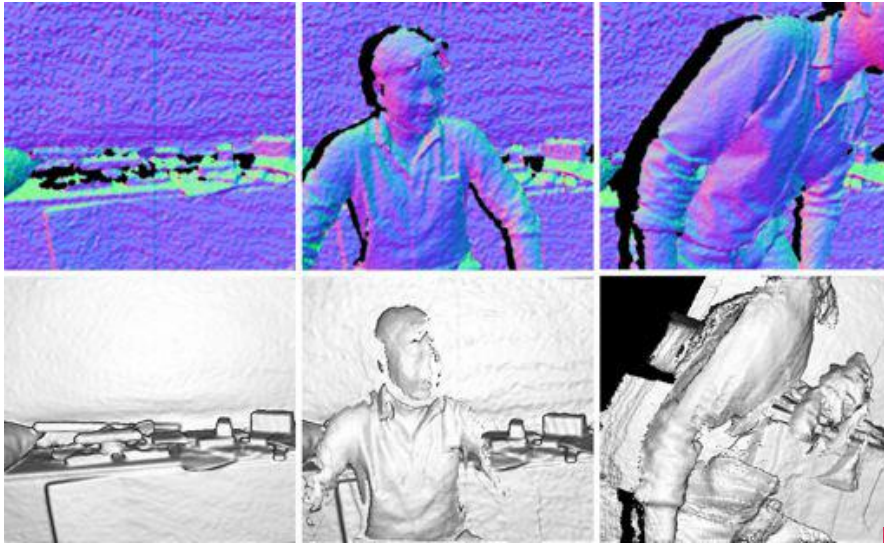


Nuvem de pontos



Superfície reconstruída

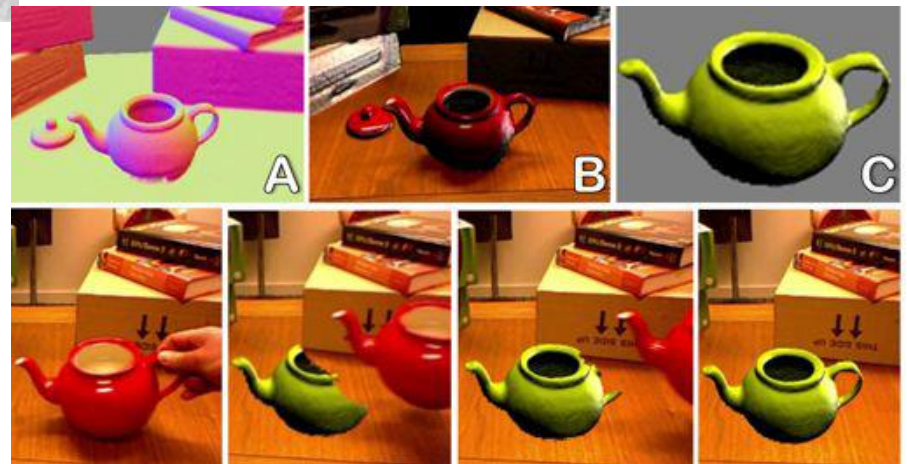
KinectFusion (IZADI et al., 2011)



Reconstrução em tempo real

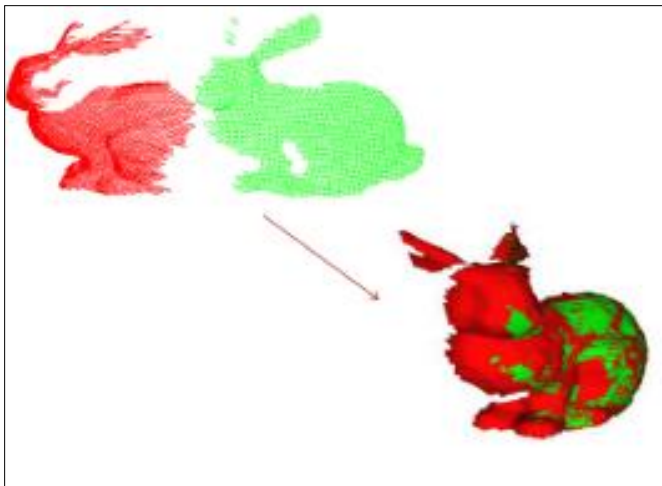
Objetivo

- Reconstrução 3D em tempo real
- Interage com o ambiente real
- Utiliza *Iterative Closest Point* (ICP)



Interação com o mundo real

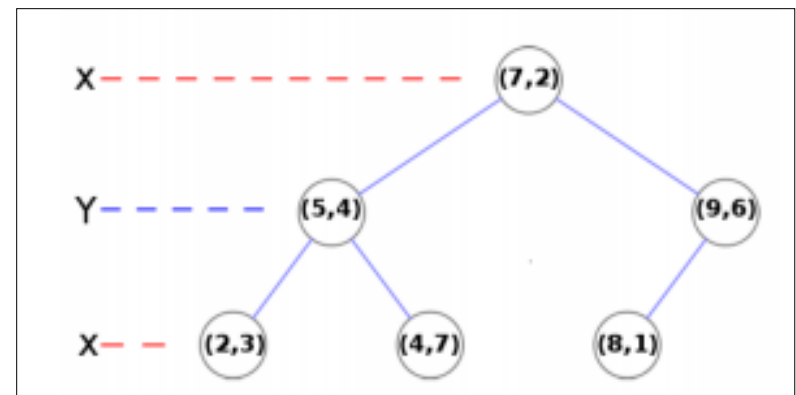
Reconstrução de objetos 3D utilizando estruturas de indexação espacial com o Microsoft Kinect (YAMADA et al. 2013)



Nuvem após ICP aplicado

Objetivo

- Comparar performance do algoritmo ICP com e sem *K D Tree*



K D Tree

Modelagem tridimensional de ambiente utilizando Kinect (MENDES, 2012)

Objetivos

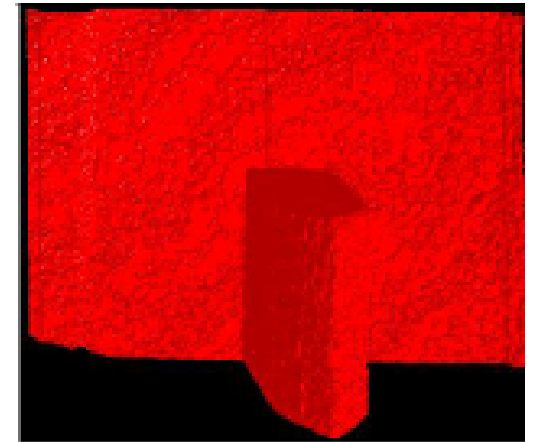
- Utilizou o SDK do Kinect
- Implementou algoritmo de triangulação



Objeto real



Reconstrução com buraco



Reconstrução

Trabalhos correlatos

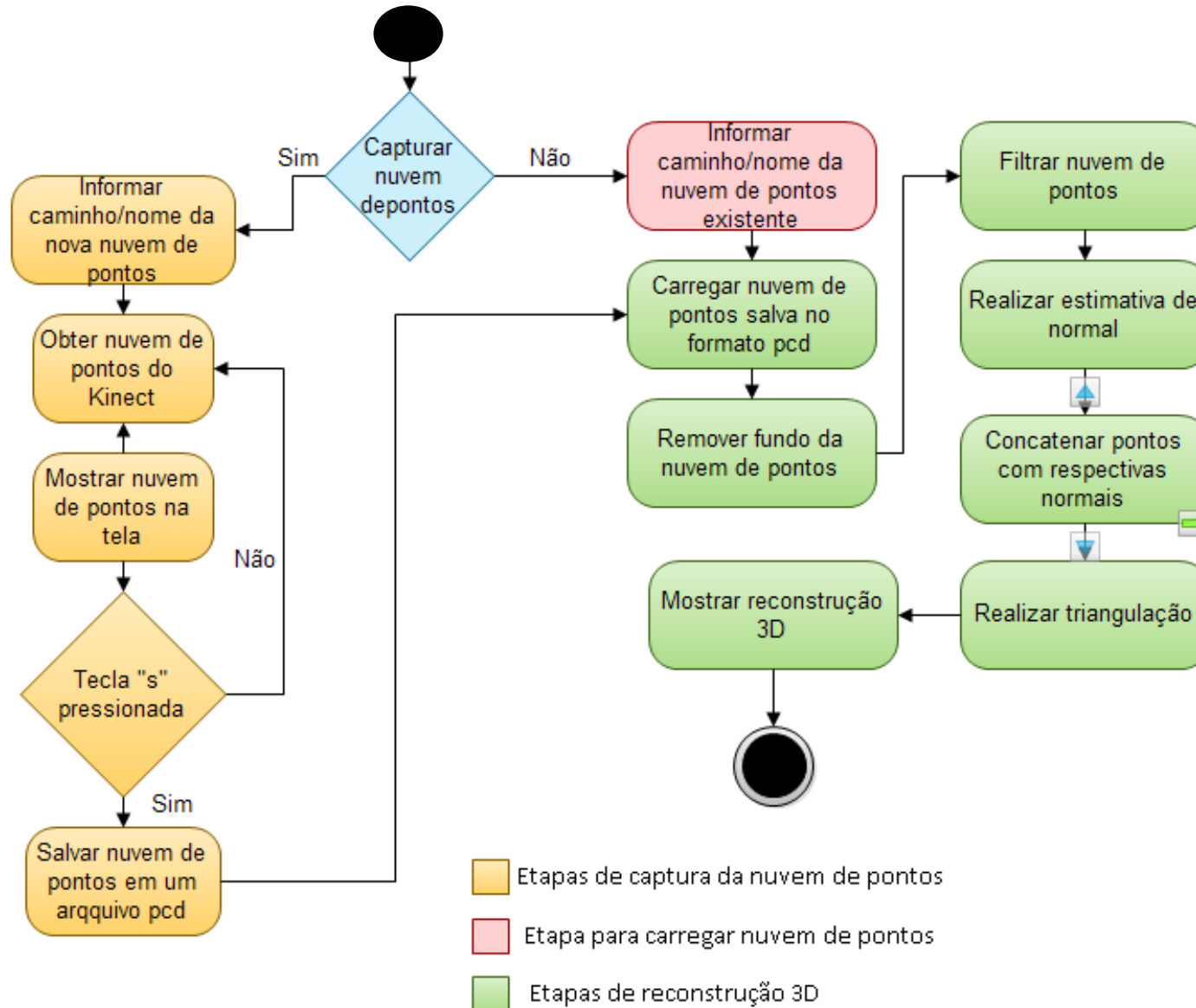
características/ trabalho relacionado	Hyvarinen (2011)	Izadi et al. (2011)	Yamada et al. (2013)	Mendes (2012)
utiliza Microsoft Kinect como <i>scanner</i> 3D	X	X	X	X
manipulação de nuvens de ponto	X	X	X	X
utiliza PCL	X			
utiliza ICP (incremental)		X	X	
utiliza <i>K-D Tree</i>	X		X	
reconstrução de superfícies	X	X		X
reconstrução por triangulação	X			X
remoção de ruídos na nuvem de pontos	X		X	
interage com mundo real		X		
exporta malha de triângulos para aplicações externas	X			

Requisitos

A seguir serão apresentados os principais requisitos funcionais (RF) e não funcionais (RNF):

- obter a nuvem de pontos do tórax feminino (RF);
- realizar a reconstrução tridimensional utilizando o PCL (RF);
- utilizar o Microsoft Kinect como *scanner* 3D (RNF);
- utilizar o *driver* OpenNI juntamente com a biblioteca PCL (RNF);

Diagrama de atividade



Implementação

- Visual Studio 2010
- Linguagem C++
- *Point Cloud Library* (PCL)
- Melhores etapas para reconstrução
 - PassThrough
 - VoxelGrid
 - MovingLeastSquares
 - GreedyProjectionTriangulation

Aquisição



Microsoft Kinect



Interface utilizada OpenNiGrabber

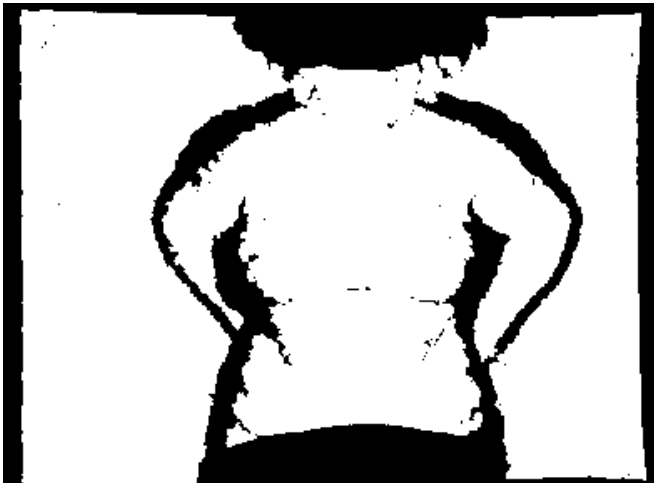


Tela para visualização de nuvens de pontos

Filtro

- Removendo o fundo

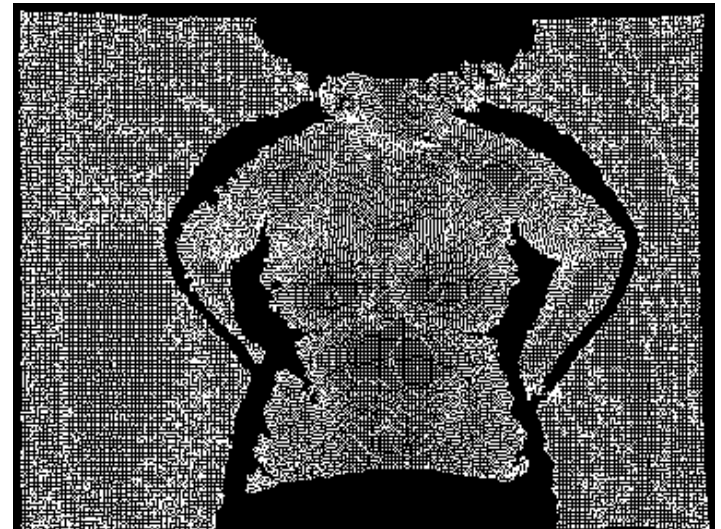
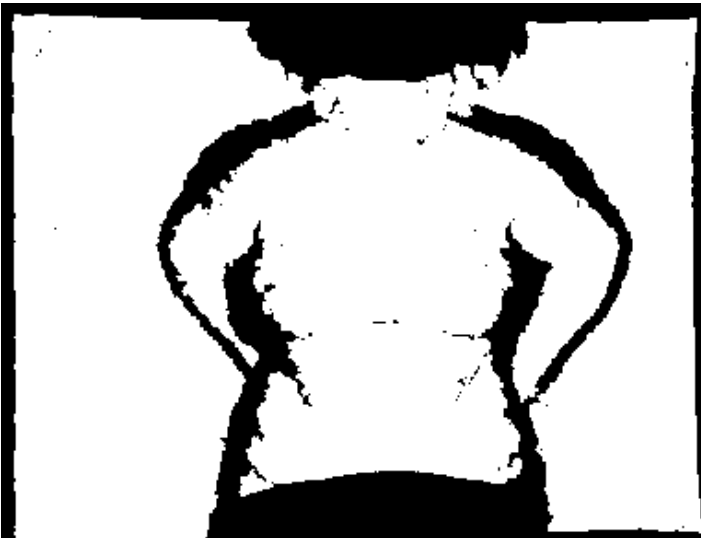
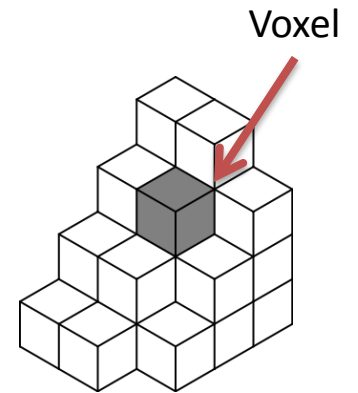
```
1   pcl::PassThrough<pcl::PointXYZ> pass;  
2   pass.setInputCloud(cloud);  
3   pass.setFilterFieldName(point);  
4   pass.setFilterLimits(lMin, lMax);  
5   pass.filter(*output);
```



Filtro

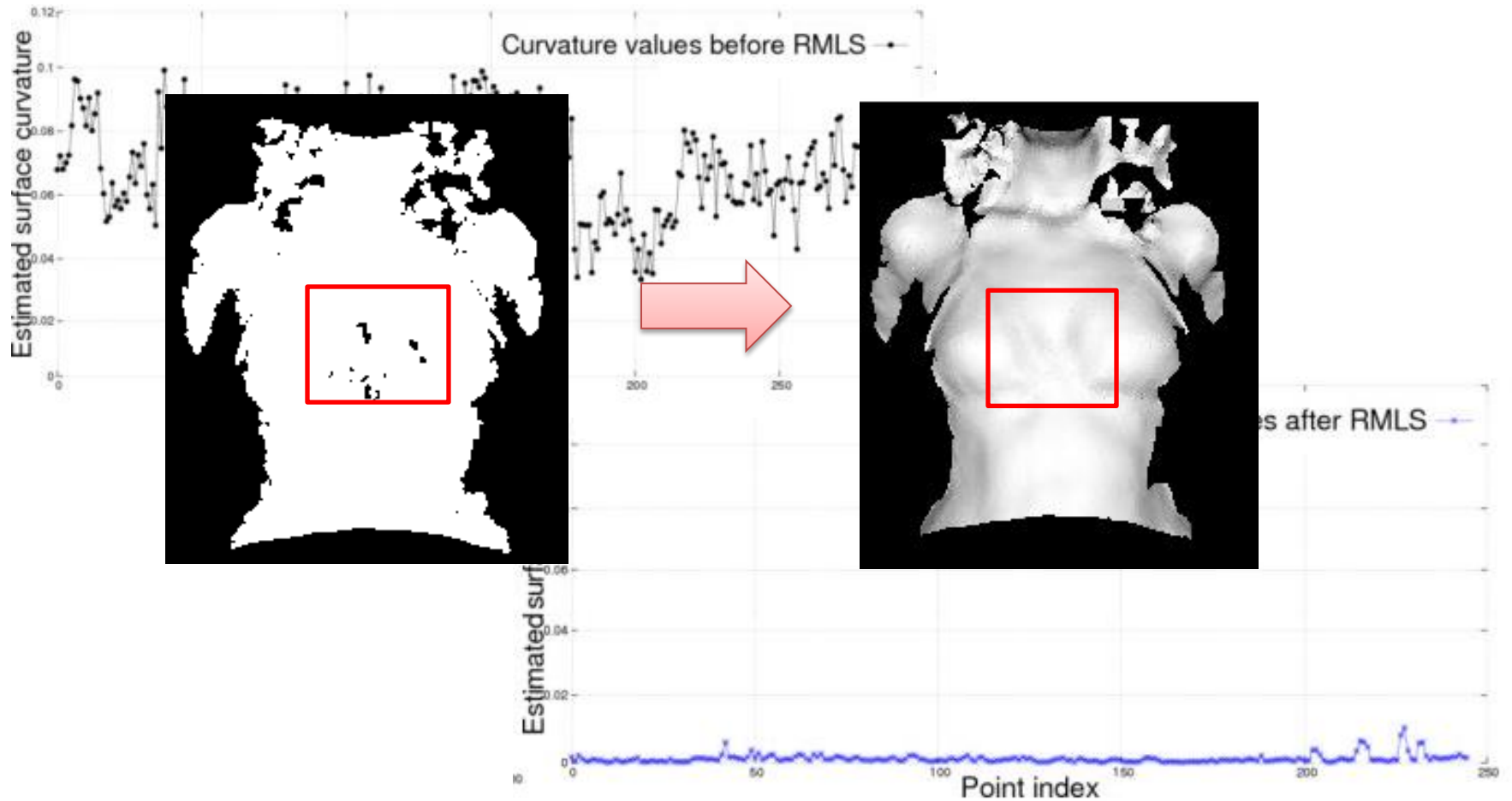
Diminuindo a densidade

```
1 pcl::VoxelGrid<sensor_msgs::PointCloud2> vg;  
2 vg.setInputCloud(input);  
3 vg.setLeafSize(0.005f, 0.005f, 0.005f);  
4 vg.filter(*output);
```



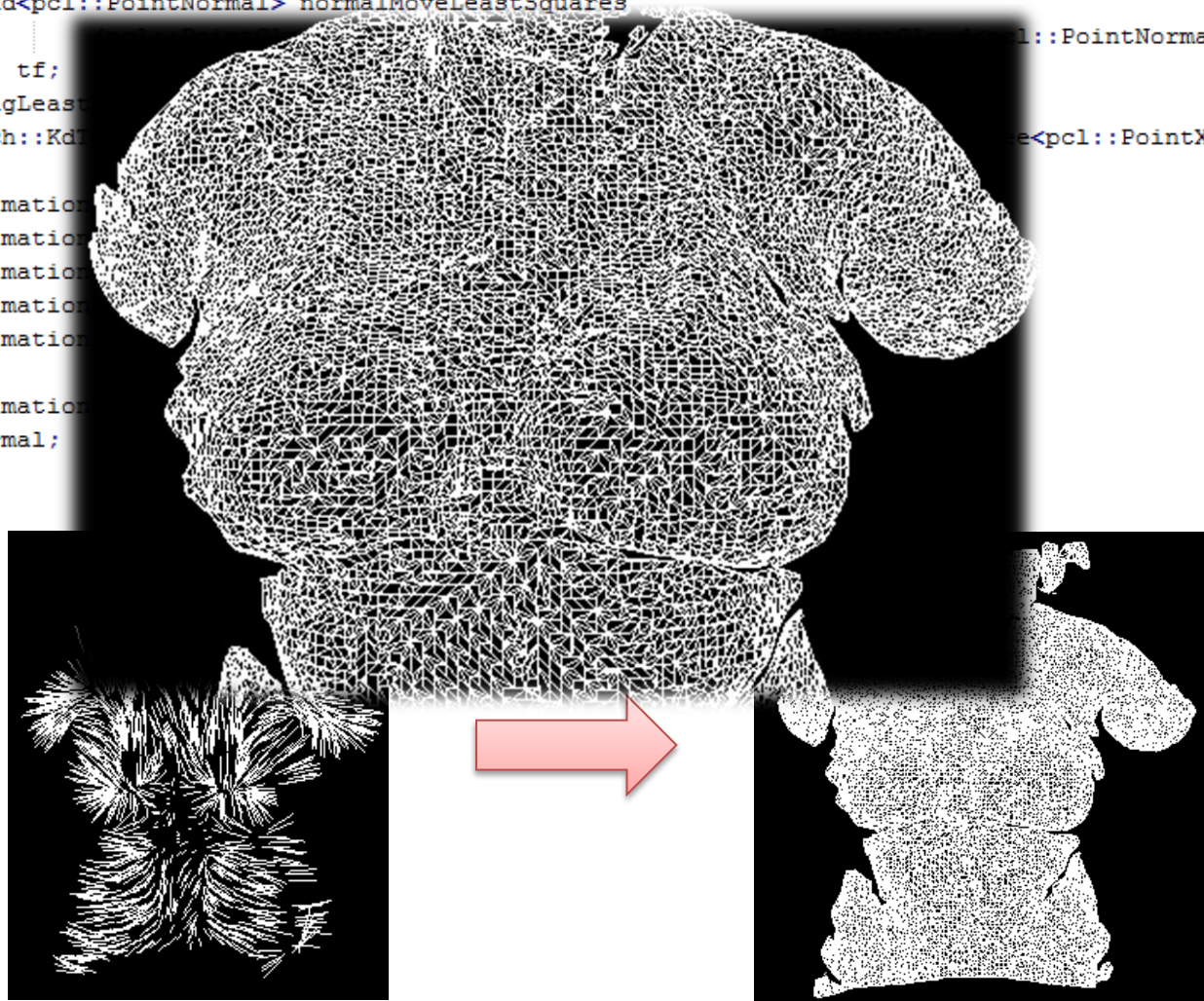
Estimativa de normal

MovingLeastSquares:



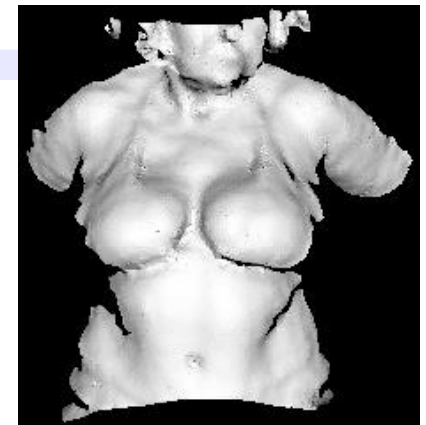
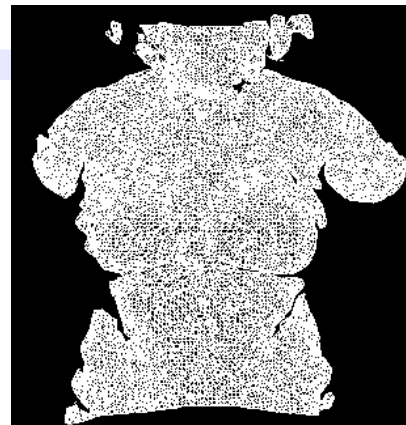
Estimativa de normal

```
1  pcl::PointCloud<pcl::PointNormal> normalMoveLeastSquares
2  [ ]
3  .....
4  .....
5  .....
6  .....
7  .....
8  .....
9  .....
10 .....
11 .....
12 .....
13 .....
14 .....
15 }
```



Reconstrução 3D

```
1  pcl::PolygonMesh Triangulation::reconstructionWithTriangulation(pcl::PointCloud<pcl::PointNormal>::Ptr input)
2
3      pcl::search::KdTree<pcl::PointNormal>::Ptr searchToReconstruction
4          (new pcl::search::KdTree<pcl::PointNormal>);
5      pcl::GreedyProjectionTriangulation<pcl::PointNormal> greedyTriangulation;
6      pcl::PolygonMesh triangulation;
7
8      searchToReconstruction->setInputCloud(input);
9
10     greedyTriangulation.setSearchRadius(0.025);
11     greedyTriangulation.setMu(2.5);
12     greedyTriangulation.setMaximumNearestNeighbors(190);
13     greedyTriangulation.setMaximumSurfaceAngle(M_PI/4);
14     greedyTriangulation.setMinimumAngle(M_PI/18);
15     greedyTriangulation.setMaximumAngle(3*M_PI/3);
16     greedyTriangulation.setNormalConsistency(false);
17     greedyTriangulation.setInputCloud(input);
18     greedyTriangulation.setSearchMethod(searchToReconstruction);
19     greedyTriangulation.reconstruct(triangulation);
20
21     return triangulation;
22 }
```



Resultados

- Testes com etapas:
 - Aquisição
 - Filtros
 - Estimativa de normal
 - Triangulação
 - ROI
 - Pontos RGB

Resultados aquisição

Amostras utilizadas para escaneamento:



Amostra 1: Manequim



Amostra 2: Mulher

Resultados aquisição

Aquisições em diferentes distâncias frontais:

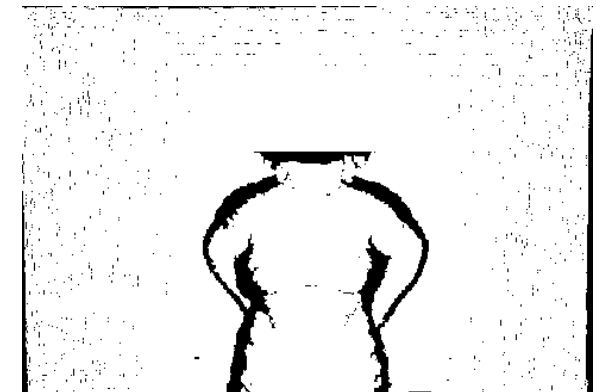
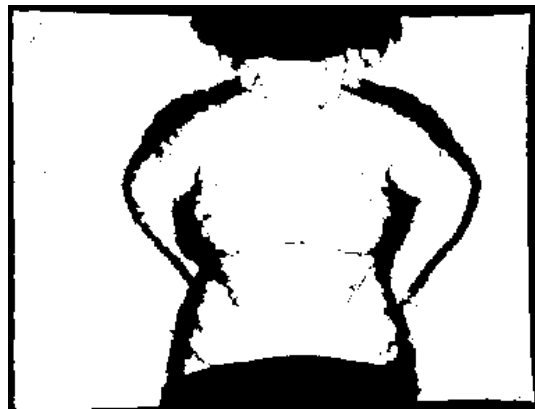
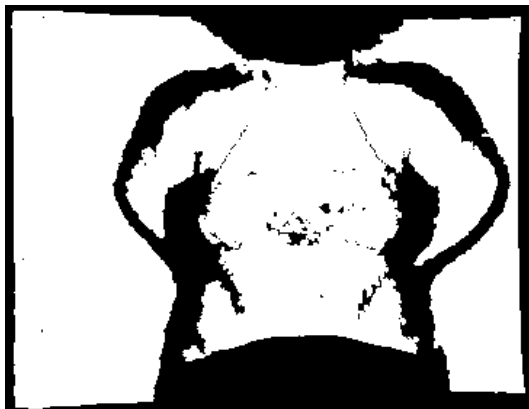
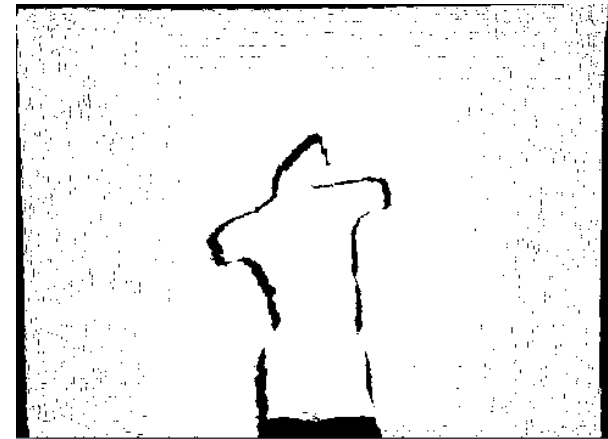
Distância de 32 cm



Distância de 57 cm



Distância de 120 cm



Resultados aquisição

Considerações

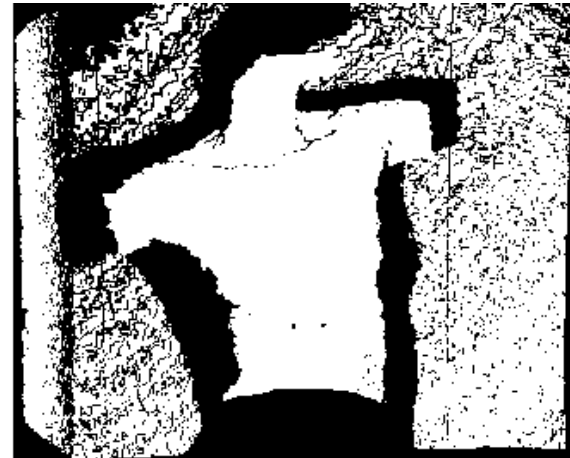
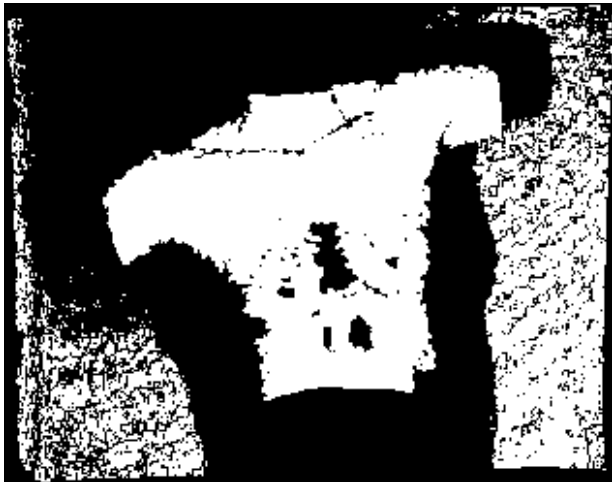
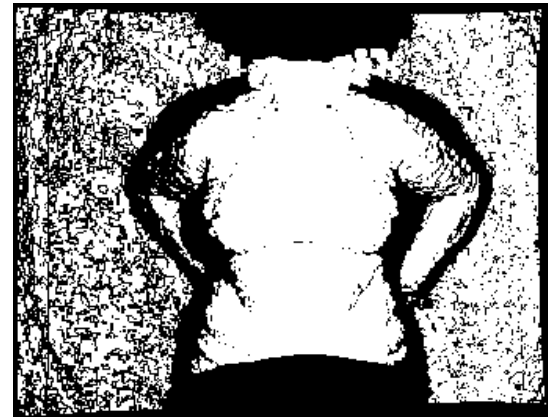
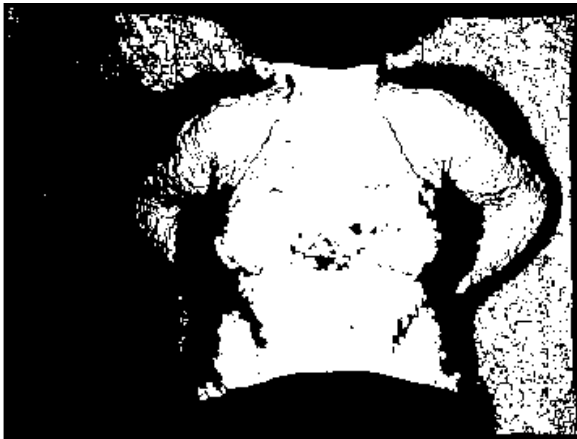
- Distância frontal de 50 a 57 centímetros
- Distância de trás de 25 a 30 centímetros



Ambiente para aquisição

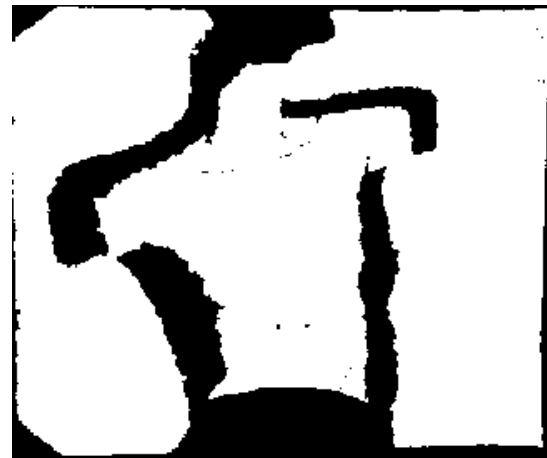
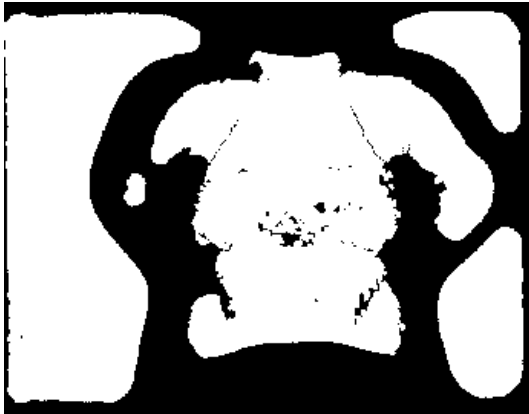
Resultados filtros

StatisticalOutlierRemoval



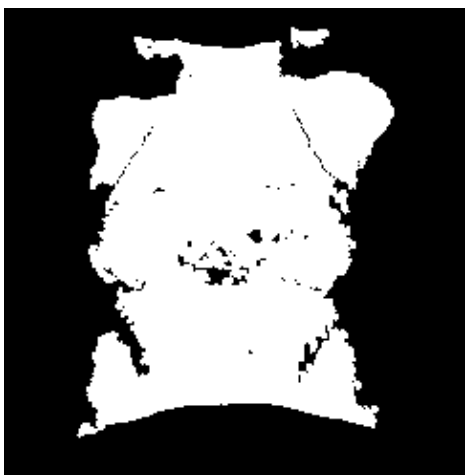
Resultados filtros

RadiusOutlierRemoval



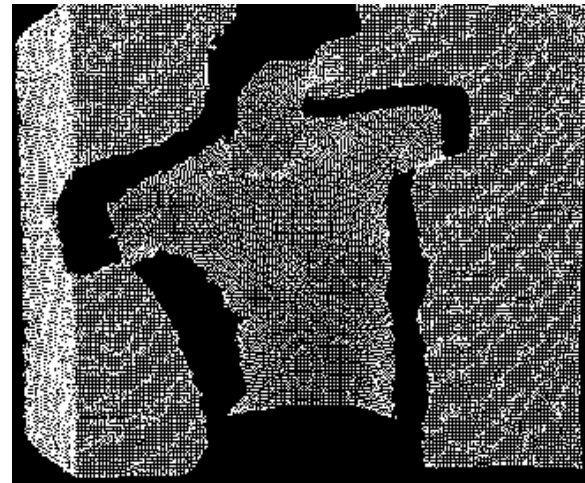
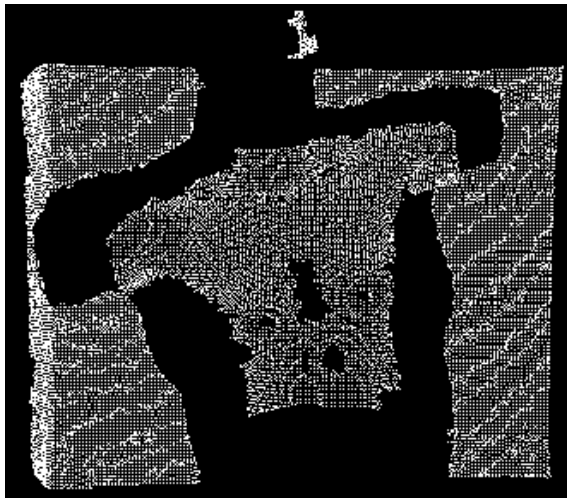
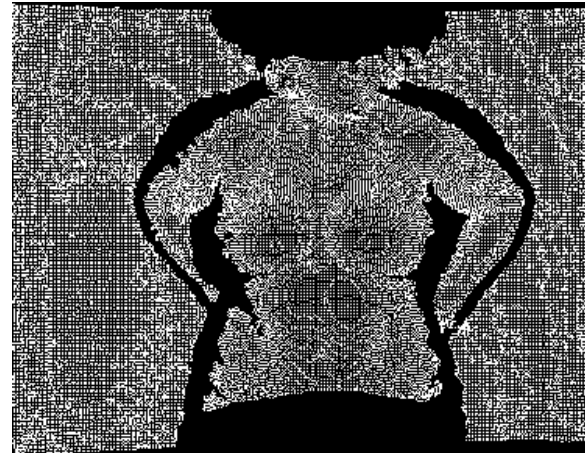
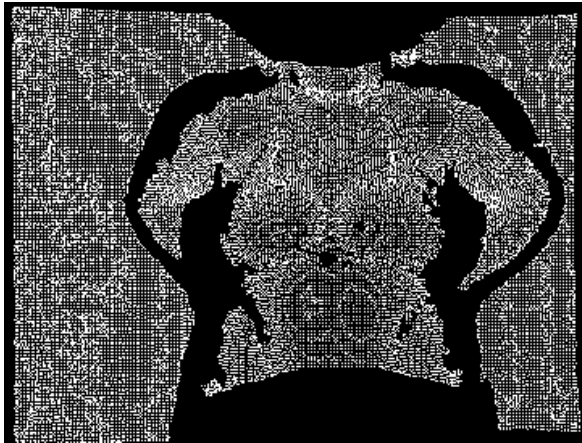
Resultados filtros

PassThrough



Resultados filtros

VoxelGrid

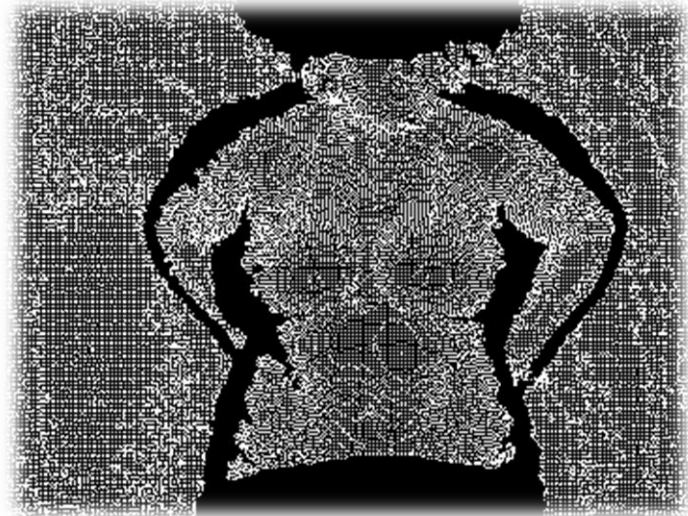


Resultados filtros

Considerações



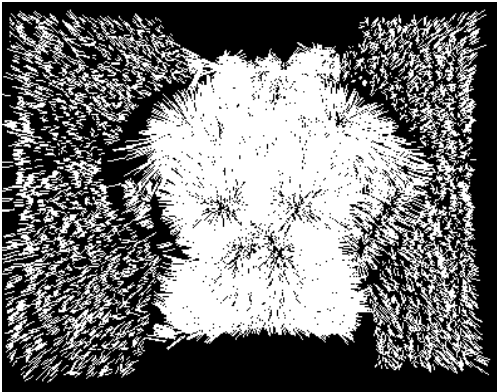
PassThrough



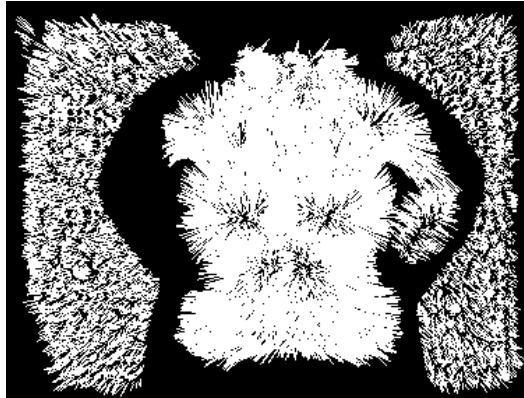
VoxelGrid

Resultados estimativa de normal

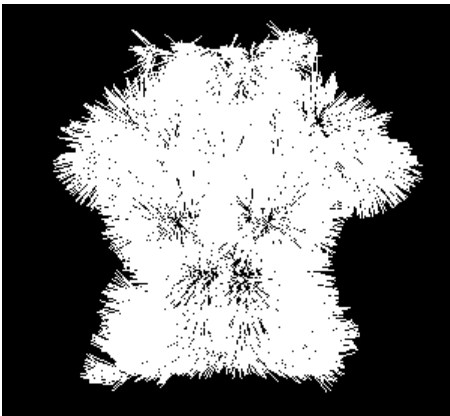
NormalEstimation:



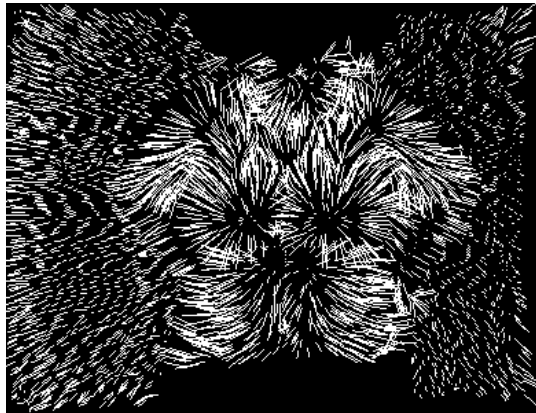
StatisticalOutlierRemoval



RadiusOutlierRemoval



PassThrough



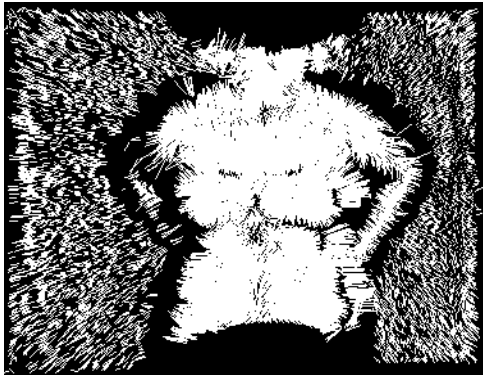
VoxelGrid

Tempos:

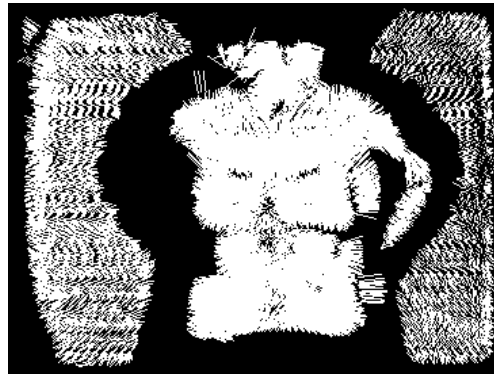
- Mais demorado foi 3min

Resultados estimativa de normal

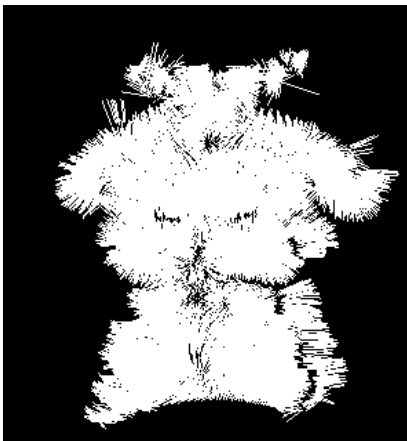
MovingLeastSquares:



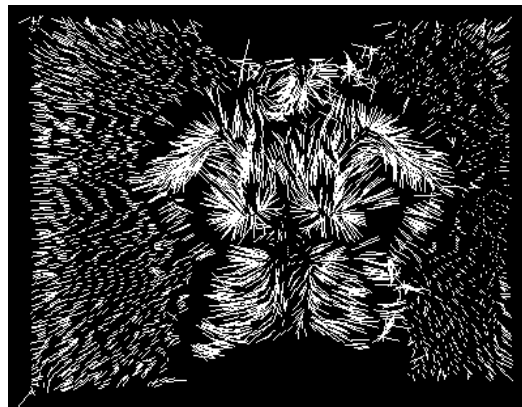
StatisticalOutlierRemoval



RadiusOutlierRemoval



PassThrough



VoxelGrid

Tempos:

- Nuvens densas demoraram 29min
- VoxelGrid demorou 9seg

Resultados estimativa de normal

Considerações

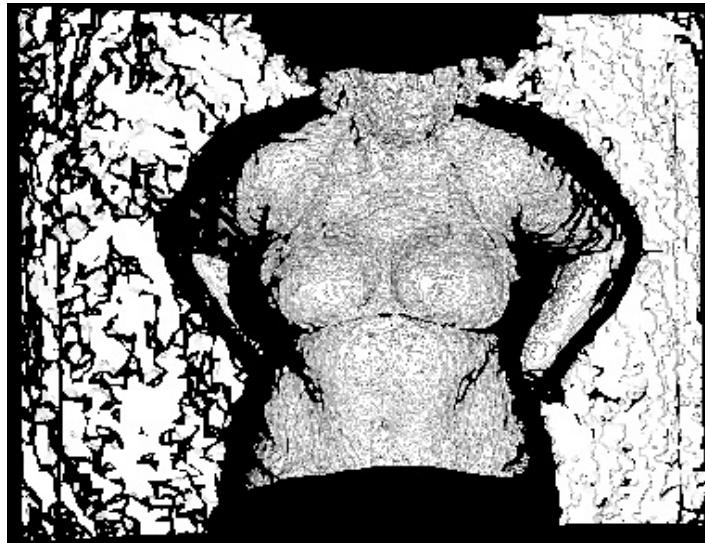
- NormalEstimation vs MovingLeastSquares
- NormalEstimation maior performance
- MovingLeastSquares melhor aspecto
- VoxelGrid anula problemas com tempo

Resultados triangulação

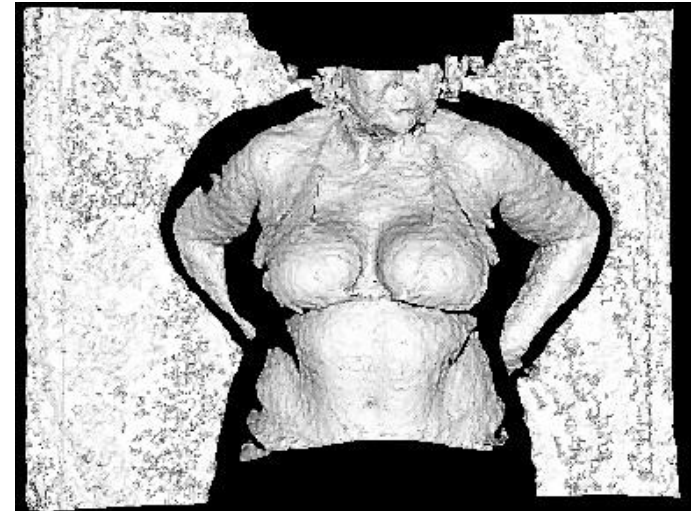
NormalEstimation + GreedyProjectionTrinagulation:

Tempos:

- Nuvens densas 35min
- VoxelGrid 9seg



StatisticalOutlierRemoval



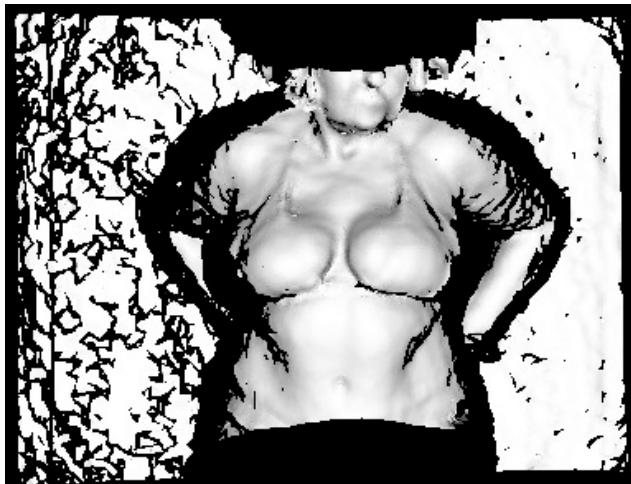
VoxelGrid

Resultados triangulação

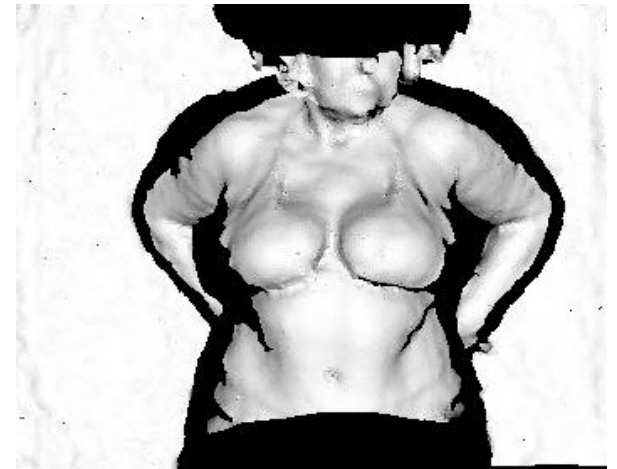
MovingLeastSquares + GreedyProjectionTrinagulation :

Tempos:

- Nuvens densas 56 minutos
- VoxelGrid 1 minutos



StatisticalOutlierRemoval



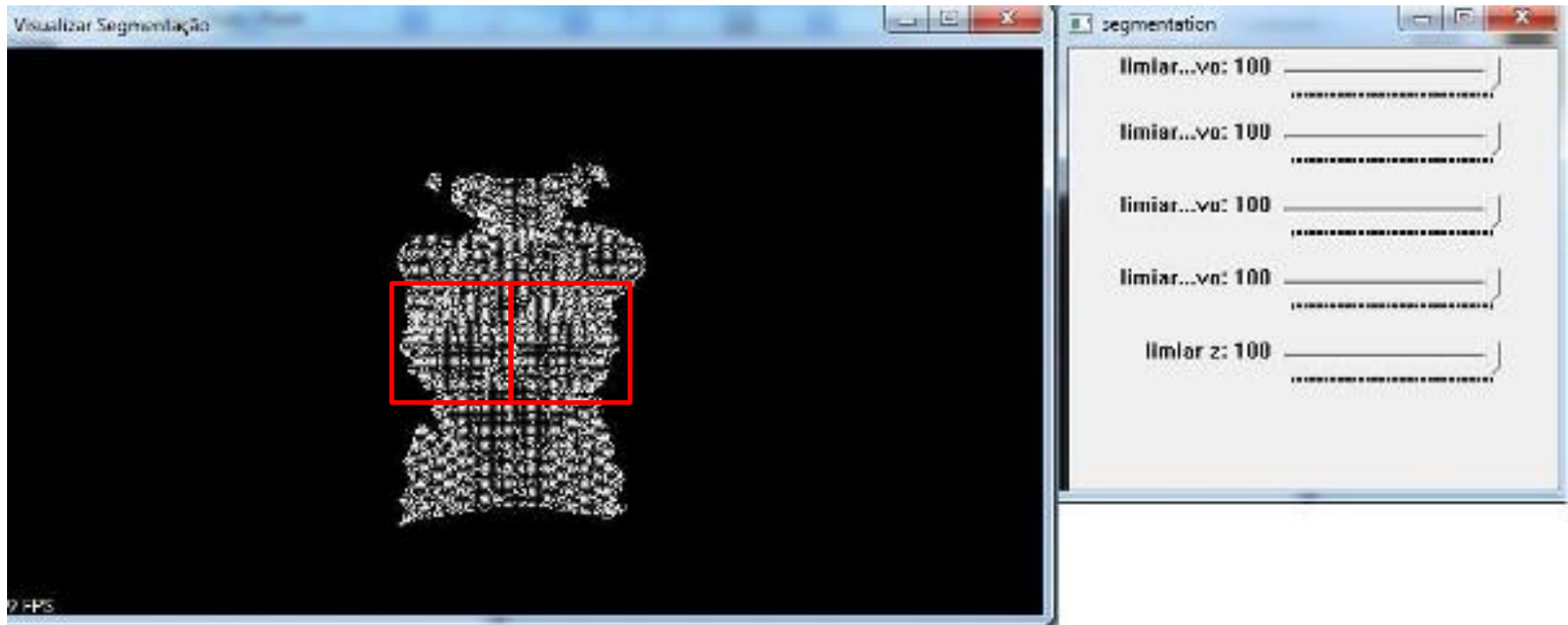
VoxelGrid

Resultados triangulação

Considerações

- Etapas de remoção de ruído e estimativa de normal impactam diretamente na triangulação
- VoxelGrid melhorou a performance
- MovingLeastSquares melhorou o aspecto

Resultados região de interesse

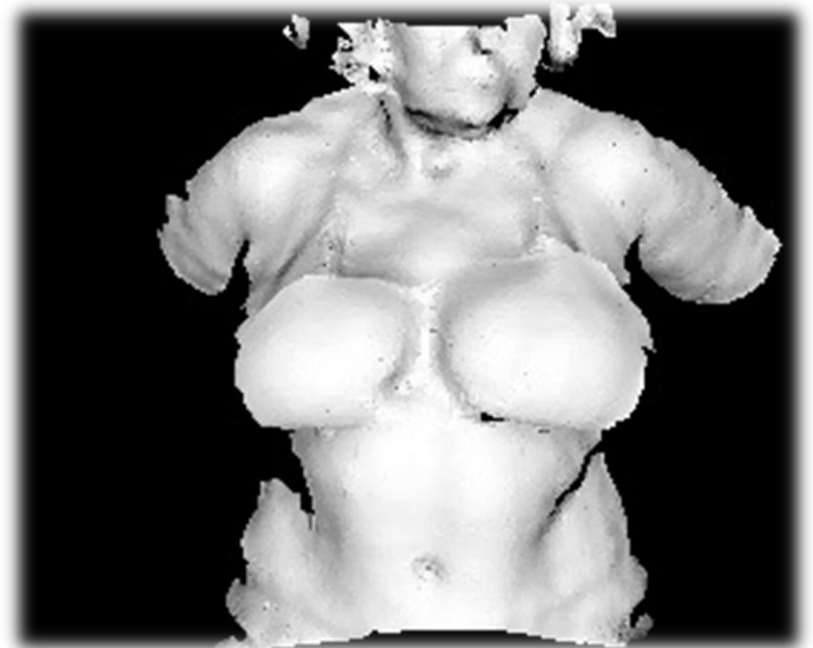
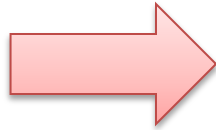
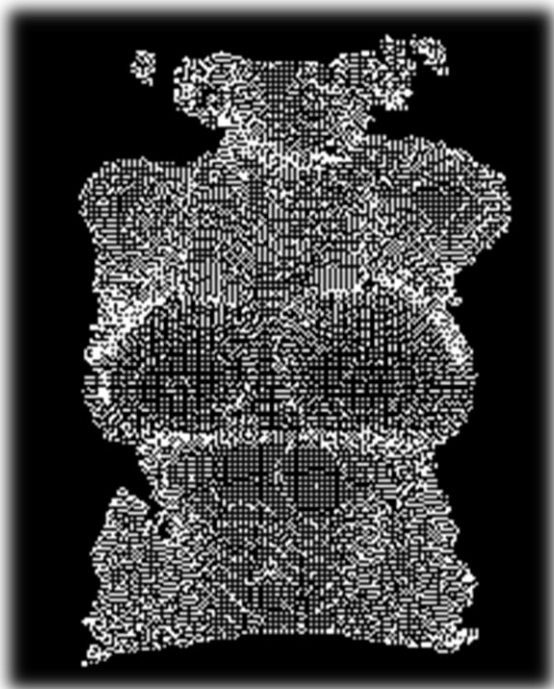


Tela para segmentação

Resultados região de interesse

Considerações

- Aspecto plano, sem profundidade



Resultados nuvem de pontos RGB

PointXYZRGB:



Nuvem colorida

Quadro comparativo dos trabalhos

características/ trabalho relacionado	Protótipo desenvolvido	Hyvarinen (2011)	Izadi et al. (2011)	Yamada et al. (2013)	Mendes (2012)
utiliza MicrosoftKinect como <i>scanner</i> 3D	X	X	X	X	X
manipulação de nuvens de ponto	X	X	X	X	X
utiliza PCL	X	X			
utiliza ICP (incremental)			X	X	
utiliza <i>K-D Tree</i>	X	X		X	
reconstrução de superfícies	X	X	X		X
reconstrução por triangulação	X	X			X
remoção de ruídos na nuvem de pontos	X	X		X	
interage com mundo real			X		
exporta malha de triângulos para aplicações externas		X			

Conclusões

- Etapas escolhidas com base nos testes foram as melhores para a reconstrução 3D
- Região de interesse não obteve resultados satisfatórios

Limitações

- Não utiliza cores
- Usuário não consegue cancelar execução
- Distância fixa entre dispositivo e tórax

Extensões

- Colocação de próteses
- *Interface* amigável
- Algoritmo de ICP

DEMONSTRAÇÃO