

Um método para reconstrução de superfícies baseado em nuvem de pontos visando representações em multirresolução

Jorge Henrique Piazzentin Ono
Prof. Adj. Antonio Carlos Sementille
Prof. Dr. Marco Antonio Corbucci Caldeira

Unesp - Faculdade de Ciências



Sumário

- 1 Introdução
- 2 Fundamentação Teórica
 - Dispositivos de aquisição de dados
 - Reconstrução de Superfícies
 - Simplificação de Superfícies
- 3 Desenvolvimento
 - Método Proposto: Poisson com simplificação local de malhas
 - Implementação do método proposto
 - Experimentos e Análise de Resultados
- 4 Conclusões e Trabalhos Futuros

Sumário

- 1 Introdução
- 2 Fundamentação Teórica
 - Dispositivos de aquisição de dados
 - Reconstrução de Superfícies
 - Simplificação de Superfícies
- 3 Desenvolvimento
 - Método Proposto: Poisson com simplificação local de malhas
 - Implementação do método proposto
 - Experimentos e Análise de Resultados
- 4 Conclusões e Trabalhos Futuros

Introdução

Representações virtuais de objetos podem ser criadas em ferramentas CAD (*Computer Aided Design*) ou geradas a partir de um modelo físico existente, através de reconstrução de superfícies [1].

Introdução

O uso de multirresolução possibilita que malhas detalhadas sejam utilizadas apenas quando necessárias (por exemplo, quando o observador está próximo do objeto visualizado).

Introdução

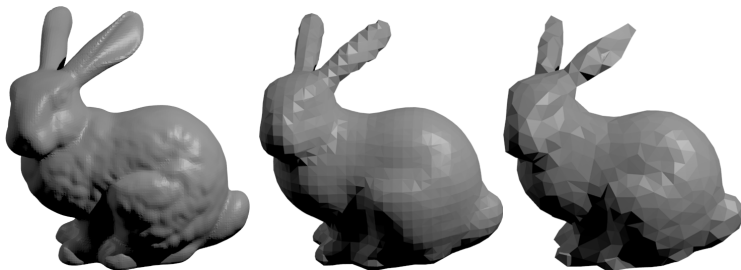


Figura: Coelho de Stanford [2] em multirresolução.

Introdução

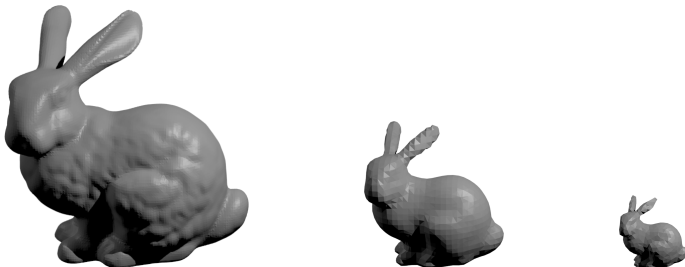


Figura: Coelho de Stanford [2] em multirresolução renderizado a diferentes distâncias do observador.

Introdução

Vantagens de malhas em menor resolução

- Redução do tamanho do arquivo armazenado;
- Redução do custo computacional para realizar transformações na malha;
- Redução do custo computacional para renderizar a cena.

Objetivos gerais

A partir da investigação de técnicas de reconstrução de superfícies, elaborar e implementar um algoritmo de reconstrução aplicado à multirresolução, visando a geração de várias representações, variando o seu nível de detalhamento e número de polígonos.

Objetivos específicos

Desenvolver um sistema com a linguagem C++ que permita:

- Integração e controle do dispositivo de aquisição Kinect;
- Criação da nuvem de pontos;
- Reconstrução tridimensional em vários níveis de detalhe;
- Visualização da malha resultante.

Sumário

1 Introdução

2 Fundamentação Teórica

- Dispositivos de aquisição de dados
- Reconstrução de Superfícies
- Simplificação de Superfícies

3 Desenvolvimento

- Método Proposto: Poisson com simplificação local de malhas
- Implementação do método proposto
- Experimentos e Análise de Resultados

4 Conclusões e Trabalhos Futuros

Fundamentação Teórica

O processo de reconstrução de superfícies pode ser descrito por um pipeline, cuja entrada são valores geométricos amostrados do mundo real e a saída é uma representação matemática da superfície.

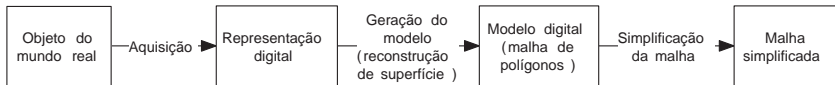


Figura: *Pipeline de Reconstrução*. Adaptado de [3].

Dispositivos de aquisição de dados

Dispositivos de aquisição de dados

Os principais tipos de dispositivos de aquisição 3D atuais são:

- Câmeras Estereoscópicas;
- *Scanners* táteis;
- Câmeras *time of flight*;
- *Scanners* de superfícies de contornos;
- Radares e sonares;
- Luz estruturada.

Câmeras estereoscópicas

- Duas ou mais lentes;
- Imagens RGB;
- Ambientes abertos ou fechados.

[4]



Figura: Câmera Binocular
STH-MDCS3-VARX [5].

Scanners táteis

- Escaneamento físico;
- Limitações físicas;
- Alta precisão.

[6]

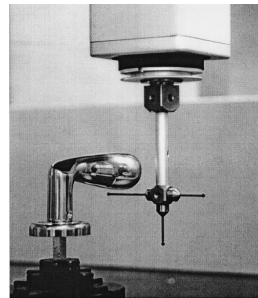


Figura: Medição de um puxador em uma Máquina de Medição de Coordenadas [7]

Câmeras *Time Of Flight*

- Emitem pulsos de luz;
- Medem o tempo entre emissão e recepção;
- Calculam a distância do objeto;
- $S = S_0 + Vt$;
- $V = 299.792.458m/s$.

[6]



Figura: FARO Focus 3D [8]

Scanners baseados em superfícies de contornos

- Muito utilizada na medicina (Ressonância Magnética e Tomografia Computadorizada);
- Capturam camadas (fotos bidimensionais) de objetos;
- Geram superfícies de contornos (curvas de nível da superfície) que podem ser reconstruídas.

[3]



Figura: Equipamento de ressonância magnética [9].

Radares e sonares

- Consideram o tempo que a onda leva para ser projetada no alvo e refletida;
- Utilizados em sensoriamento remoto de longo alcance;
- Não é possível escanear objetos pequenos, pois é preciso medir picossegundos (10^{-12} segundos).

[3]

Métodos de luz estruturada

- Produz imagens 2D contendo os valores de distância;
- Projetor projeta padrões de luz no objeto;
- Câmera captura deformações no padrão de luz.

[3]

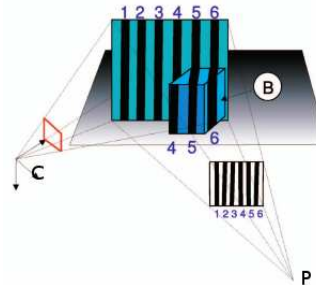


Figura: Esquema de captura com luz estruturada [10].

Microsoft Kinect

Microsoft Kinect é um sensor desenvolvido pela Microsoft em conjunto com a PrimeSense para o Xbox360. Foi anunciado em 2009 com o nome “Projeto Natal” .

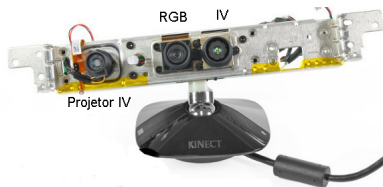
Características

- Câmera RGB-D;
- Sensor de luz estruturada infravermelho;
- Vetor de microfones;
- Motor e acelerômetro;
- Baixo custo;

Microsoft Kinect



(a) Fechado



(b) Aberto com detalhes dos sensores

Figura: Sensor Microsoft Kinect[11].

Microsoft Kinect



(a) Imagem RGB

(b) Imagem de profundidade.

Figura: Cena capturada pelo sensor RGB-D do Kinect[12].

Reconstrução de Superfícies

Reconstrução de Superfícies

Dados de entrada para os algoritmos

- Seções planares;
- Dados volumétricos;
- Nuvem de pontos.

Reconstrução de Superfícies

O processo de reconstrução baseado em nuvem de pontos envolve um *pipeline* complexo.

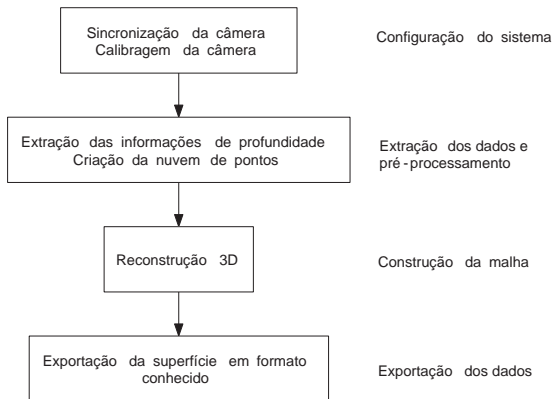
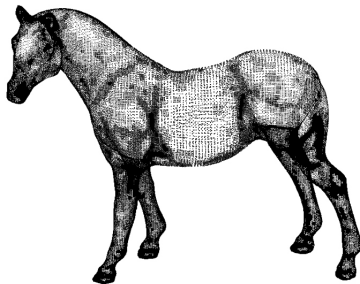
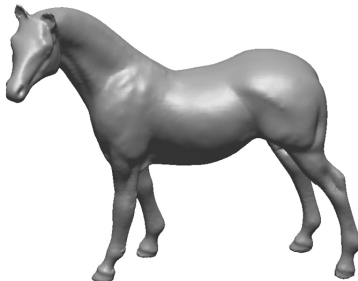


Figura: Pipeline adotado para reconstrução tridimensional [4].

Reconstrução de Superfícies



(a) Nuvem de pontos



(b) Superfície reconstruída

Figura: Reconstrução a partir de uma nuvem de pontos. Nuvem fornecida por [13].

Algoritmos de reconstrução

Marching Cubes [14]

- Determina como a superfície intercepta cubos marchantes;
- Existem 8 vértices e 2 estados (dentro e fora da superfície);
- $2^8 = 256$ possibilidades, 15 padrões sem redundância;
- Método gera triângulos;

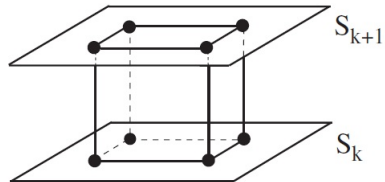


Figura: Cubo interceptado por duas superfícies [15].

Algoritmos de reconstrução

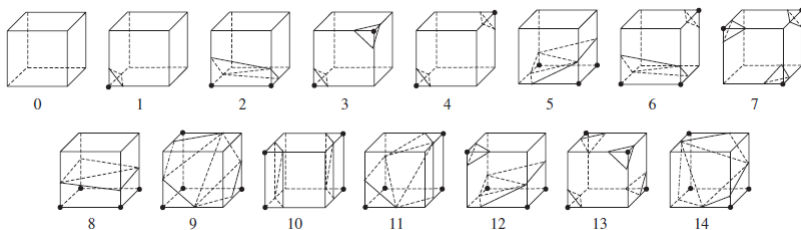


Figura: Triangulação das 15 possibilidades do algoritmo Marching Cubes [14].

Algoritmos de reconstrução

Octrees [4]

- Cena é o nó raiz;
- Cada nó tem 8 filhos, dividindo igualmente o tamanho de seus pais;
- O processo é repetido até que:
 - Cada nó esteja cheio ou vazio;
 - Um coeficiente de erro seja atingido;
 - O número máximo recursões seja alcançado.

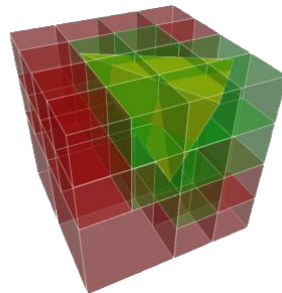


Figura: Representação de um objeto através de Octrees [4].

Reconstrução de superfícies a partir de nuvens de pontos

Principais categorias [16]

- Função de distância:
 - Calculam a distância entre qualquer ponto e a superfície. Em seguida aplicam Marching Cubes;
- Subdivisão espacial:
 - Utilizam representações espaciais para reconstruir a superfície, por exemplo, *Octrees*;
- Métodos de deformações:
 - Alteram uma superfície inicial para que se aproxime dos dados amostrados;
- Superfícies incrementais:
 - Geram a superfície diretamente, através da ligação de seus pontos.

Reconstrução de Poisson

O algoritmo de Poisson [17] aproxima pontos orientados (ponto + vetor normal) através de uma equação de Poisson.

- Calcula função indicadora (ponto está dentro ou fora da superfície);
- Resolve a equação de Poisson por meio de aproximação com *Octrees*.
- Extrai os contornos da função indicadora com o *Marching Cubes*.

Reconstrução de Poisson

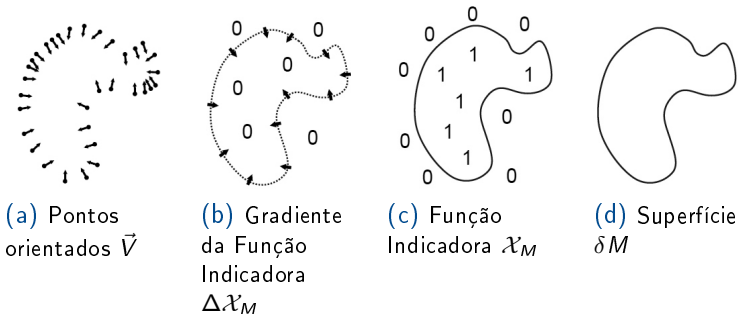


Figura: Ilustração do método de reconstrução de Poisson em 2D [17].

Reconstrução de Poisson

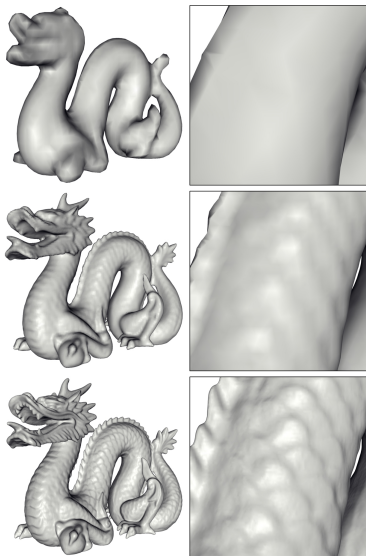


Figura: Dragão de Stanford [2] reconstruído com o método de Poisson em 3 níveis de detalhe [17].

Simplificação de Superfícies

Simplificação de Superfícies

Simplificação de superfícies é definida como a redução do número de faces em uma malha densa, mantendo-se a forma e a topologia da malha original [18].

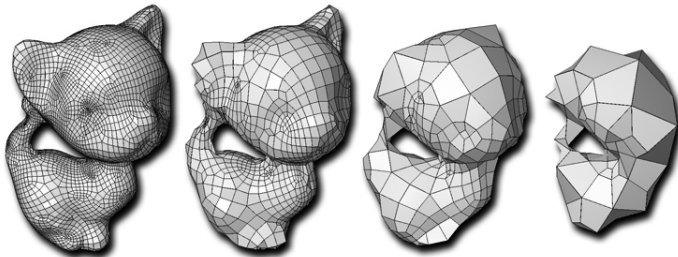


Figura: Representação de um gato em múltiplas resoluções[19].

Simplificação de Superfícies

Métodos incrementais

- Sequência de atualizações locais;
- Cada atualização reduz o tamanho da malha e a precisão da aproximação;

Algoritmo tradicional [20]

Faça

Seleciona elemento a ser removido / contraído

Executa operação

Atualiza malha

Até que: precisão/tamanho da malha esteja satisfatório

Simplificação de Superfícies

Métodos incrementais

Lindstrom e Turk(1998) desenvolveram um algoritmo de simplificação baseado na operação de contração de arestas.

O método tenta preservar:

- O volume da superfície original;
- Triângulos da malha.

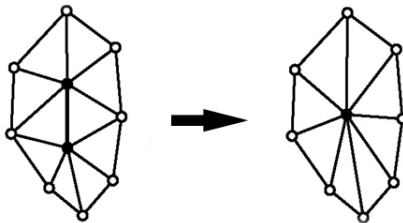


Figura: Contração de arestas [21].

Simplificação de Superfícies

Métodos não incrementais

Alguns exemplos de métodos não incrementais são [20]:

- Junção de faces coplanares;
- *Remeshing*;
- *Clustering*;
- Métodos baseados em *wavelets*.

Sumário

- 1 Introdução
- 2 Fundamentação Teórica
 - Dispositivos de aquisição de dados
 - Reconstrução de Superfícies
 - Simplificação de Superfícies
- 3 **Desenvolvimento**
 - Método Proposto: Poisson com simplificação local de malhas
 - Implementação do método proposto
 - Experimentos e Análise de Resultados
- 4 Conclusões e Trabalhos Futuros

Reconstrução de Poisson

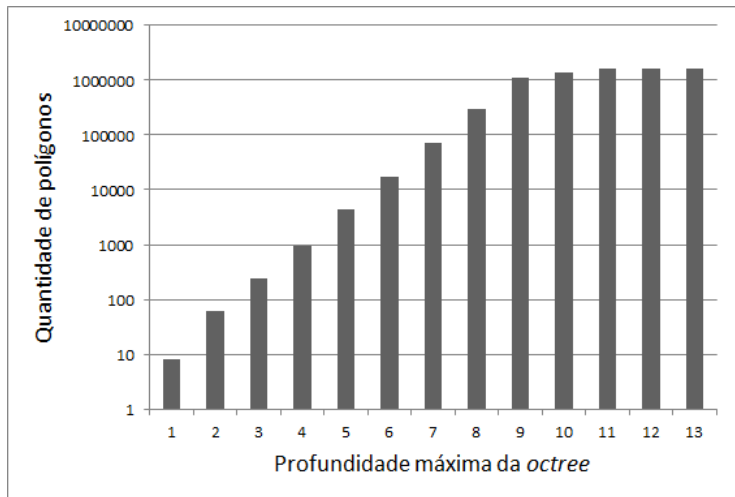
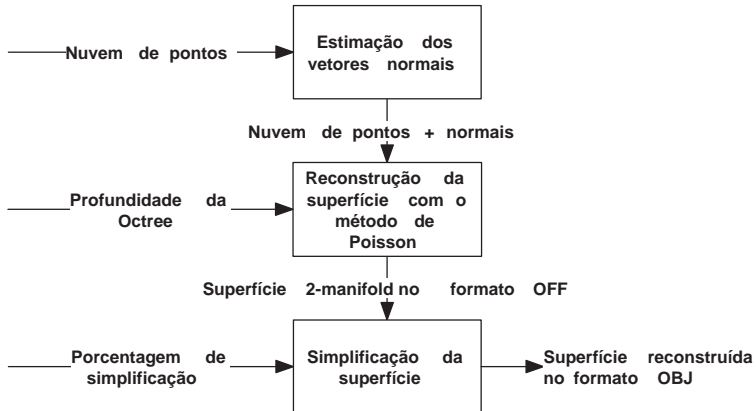


Figura: Gráfico da quantidade de polígonos da reconstrução do Coelho de Stanford por profundidade de *octree*.

Método Proposto: Poisson com simplificação local de malhas

Método proposto: Poisson com simplificação local de malhas

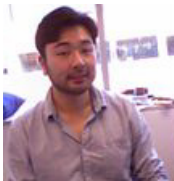
Combinação da reconstrução de superfícies de Poisson [17] com *Edge collapse* [21].



Método proposto: Poisson com simplificação local de malhas

Captura da nuvem de pontos

- Captura de 1 frame;
- Captura de vários frames e combinação com algoritmo ICP.



(a) Imagem RGB



(b) Reconstrução - 1 frame.



(c) Reconstrução - frames combinados.

Figura: Reconstruções com 1 e vários frames [22].

Método proposto: Poisson com simplificação local de malhas

Filtragem dos dados

- Remoção de valores desconhecidos;
- Filtragem através da combinação de frames com ICP e KinectFusion [22];
- Segmentação do objeto:
 - Automática;
 - Manual.

Implementação do método proposto

Implementação do método proposto

Configuração do ambiente experimental

Hardware

- 1 Computador pessoal:
 - Processador: Intel(R) Core(TM) i5-2400. 3.10GHz;
 - Memória RAM: 8 GB;
 - Placa de vídeo: GeForce GTX 550 Ti (Clock de 900 MHz, 1GB de memória GDDR5, 192 processadores);
 - Disco rígido: SATA 1 TB;
 - Placa mãe: Asus LGA1155 P8H61 PRO.
- 1 Sensor Microsoft Kinect (Modelo XBOX 360).

Implementação do método proposto

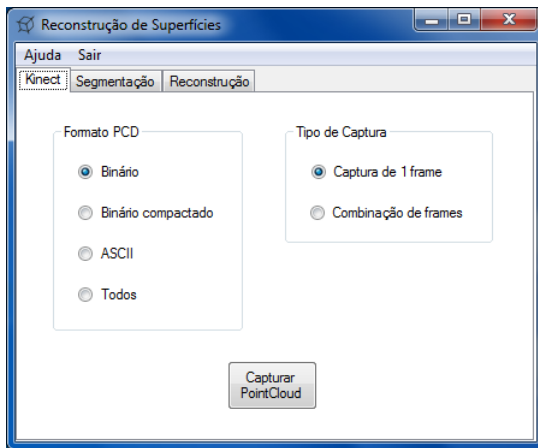
Configuração do ambiente experimental

Software

- Sistema operacional: Windows 7 Ultimate SP1;
- Ambiente de desenvolvimento: Microsoft Visual Studio 2010;
- Sistema de configuração de projetos: CMake 2.8.9;
- Biblioteca PCL 1.6.0 e dependências:
 - Boost 1.49;
 - Eigen 3.0.5;
 - FLANN 1.7.1;
 - VTK 5.8.0;
 - QHull 2011.1 (6.2.0.1385);
 - OpenNI 1.3.2;
 - Sensor 5.0.3.3.
- Biblioteca CGAL 4.0.

Implementação do método proposto

Sistema desenvolvido



Experimentos e Análise de Resultados

Experimentos e Análise de Resultados

Repositório 3D de Stanford [2]

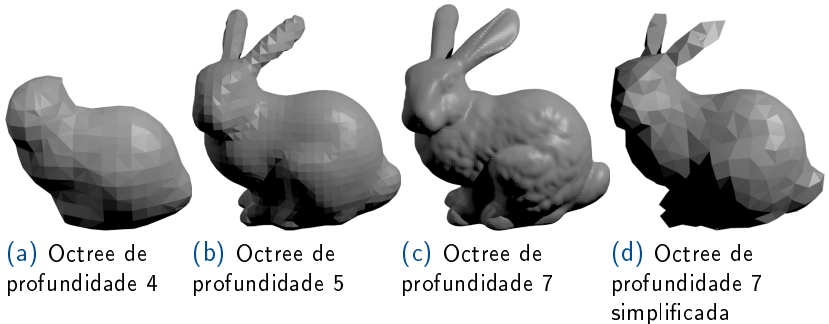


Figura: Algoritmo aplicado aos dados do Coelho de Stanford.

Experimentos e Análise de Resultados

Repositório 3D de Stanford [2]

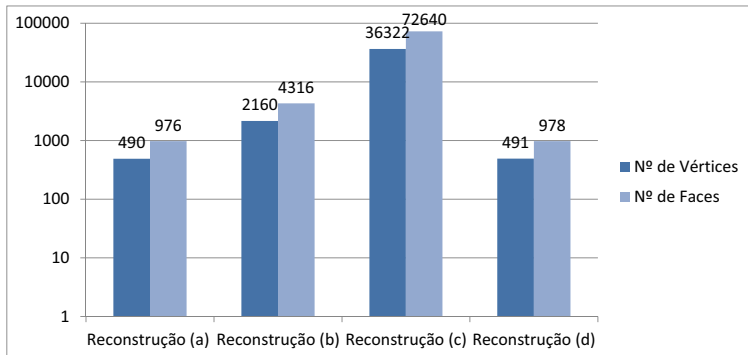


Figura: Resultados da reconstrução do Coelho de Stanford

Experimentos e Análise de Resultados

Repositório 3D de Stanford [2]

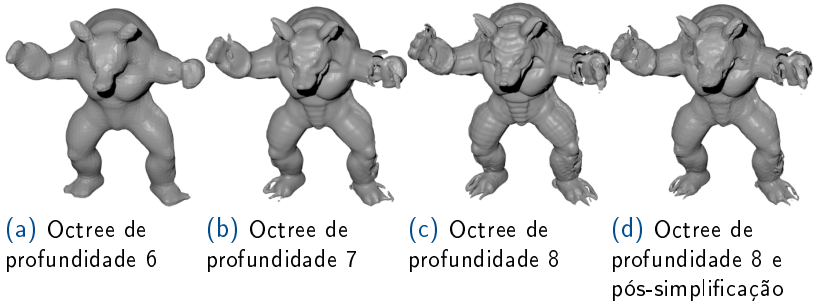


Figura: Algoritmo aplicado aos dados do modelo *Armadillo* de Stanford.

Experimentos e Análise de Resultados

Repositório 3D de Stanford [2]

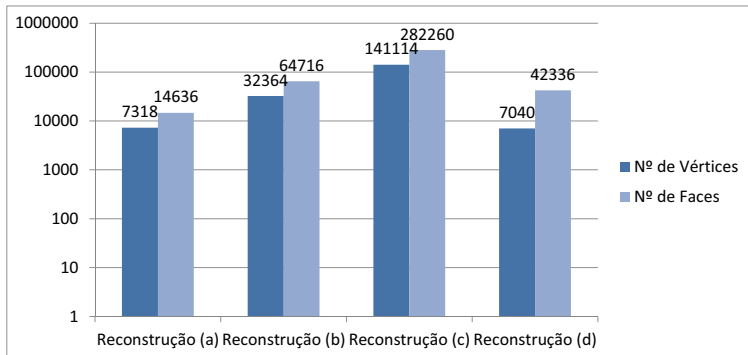


Figura: Resultados da reconstrução do Armadillo de Stanford

Experimentos e Análise de Resultados

RGB-D Dataset de Washington [23]



(a) Octree de profundidade 5



(b) Octree de profundidade 6



(c) Octree de profundidade 10



(d) Octree de profundidade 10 e pós-simplificação

Figura: Algoritmo aplicado aos dados de uma banana.

Experimentos e Análise de Resultados

RGB-D Dataset de Washington [23]

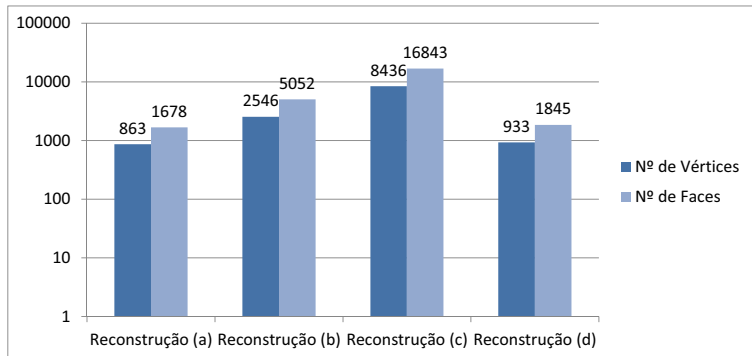
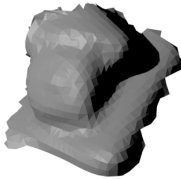


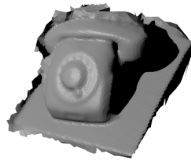
Figura: Resultados da reconstrução de uma banana

Experimentos e Análise de Resultados

Dados captados pelo Kinect



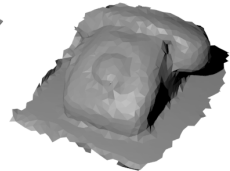
(a) Octree de profundidade 5



(b) Octree de profundidade 7



(c) Octree de profundidade 10



(d) Octree de profundidade 10 e pós simplificação

Figura: Algoritmo aplicado aos dados de um telefone.

Experimentos e Análise de Resultados

Dados captados pelo Kinect

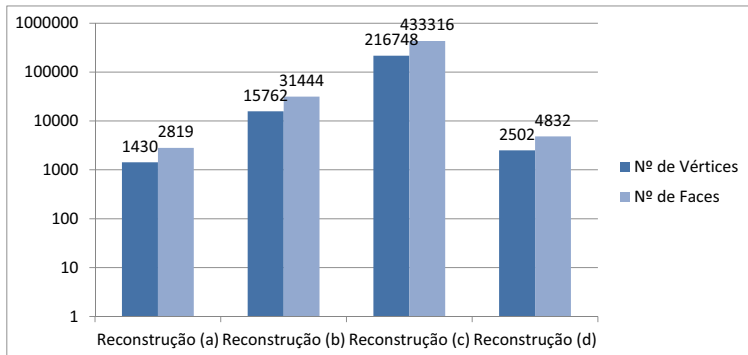
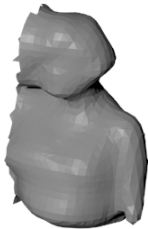


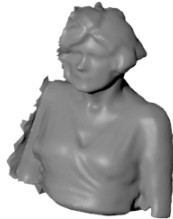
Figura: Resultados da reconstrução de um telefone

Experimentos e Análise de Resultados

Dados captados pelo Kinect



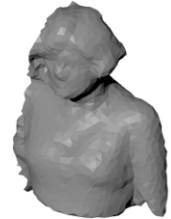
(a) Octree de profundidade 5



(b) Octree de profundidade 7



(c) Octree de profundidade 10



(d) Octree de profundidade 10 e pós-simplificação

Figura: Algoritmo aplicado aos dados de uma figura humana.

Experimentos e Análise de Resultados

Dados captados pelo Kinect

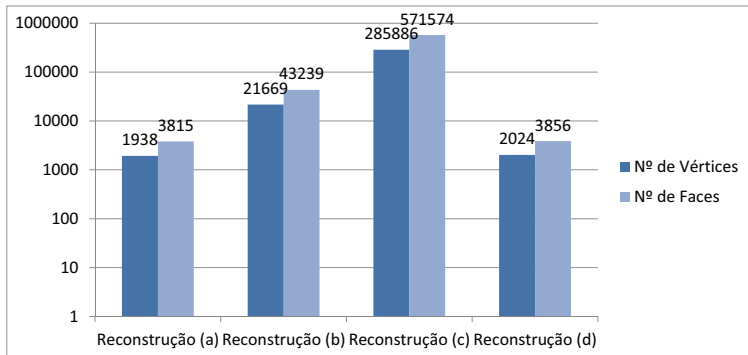


Figura: Resultados da reconstrução de uma figura humana.

Sumário

- 1 Introdução
- 2 Fundamentação Teórica
 - Dispositivos de aquisição de dados
 - Reconstrução de Superfícies
 - Simplificação de Superfícies
- 3 Desenvolvimento
 - Método Proposto: Poisson com simplificação local de malhas
 - Implementação do método proposto
 - Experimentos e Análise de Resultados
- 4 Conclusões e Trabalhos Futuros

Conclusões

- Microsoft Kinect é capaz de trabalhar como um *scanner* 3D;
- Reconstrução em multirresolução de Poisson altera muito a topologia do objeto;
- Com o método proposto, é possível gerar malhas mais simplificadas, mantendo-se a topologia original da superfície;
 - Deve-se escolher parâmetros corretos para *Octree* e porcentagem de simplificação;
- Os objetivos iniciais do trabalho foram cumpridos.

Trabalhos Futuros

- Verificação da qualidade das malhas;
- Implementação de técnicas de texturização;
- Desenvolvimento de outras abordagens de simplificação (coplanaridade e *wavelets*);
- Aplicação de filtro de suavização para ruídos multiplicativos [24].

Agradecimentos

- À banca examinadora;
- Aos orientadores;
- Aos amigos presentes.

Referências I

- [1] H. Muller.
Surface reconstruction - an introduction.
In *Scientific Visualization Conference, 1997*, page 239, June 1997.
- [2] Stanford.
The stanford 3D scanning repository, 2011.
- [3] W. Saleem.
A flexible framework for learning-based surface reconstruction.
PhD thesis, Masters thesis, Computer Science Department,
University of Saarland, Postfach 15 11 50, 66041 Saarbrücken,
2004.

Referências II

- [4] R Madsen, C. Kehl, V. Nica, A. Barbulescu, and T. Tsesmelis.
3D scanning and reconstruction of large scale environments.
Technical report, AALBORG UNIVERSITY, 2011.
- [5] Guo Yan Xu, Li Peng Chen, and Feng Gao.
Study on binocular stereo camera calibration method.
In *Image Analysis and Signal Processing (IASP), 2011
International Conference on*, pages 133 –137, October 2011.
- [6] W. Böhler and A. Marbs.
3D scanning instruments.
Proceedings of the CIPA WG, 6:9e18, 2002.

Referências III

- [7] V. Carbone, M. Carocci, E. Savio, G. Sansoni, and L. De Chiffre.
Combination of a vision system and a coordinate measuring machine for the reverse engineering of freeform surfaces.
The International Journal of Advanced Manufacturing Technology, 17(4):263–271, 2001.
[10.1007/s001700170179](#).
- [8] FARO.
LASER SCANNER, 2012.
- [9] GE Healthcare.
Magnetic resonance imaging, 2012.

Referências IV

- [10] Sergio Fernandez Navarro.
3D reconstruction of object shape using structured light.
PhD thesis, University of Girona, Department of Computer Architecture and Technology, 2009.
- [11] Jan Smisek, Michal Jancosek, and Tomas Pajdla.
3D with kinect.
In *1 IEEE Workshop on Consumer Depth Cameras for Computer Vision*, 2011.
- [12] Marek Solony and Pavel Zemcik.
Scene reconstruction from kinect motion.
In *Conference and Competition Student EEICT*, 2011.

Referências V

- [13] Michael Kazhdan.
Screened poisson surface reconstruction (Version 4), 2012.
- [14] William E. Lorensen and Harvey E. Cline.
Marching cubes: A high resolution 3d surface construction algorithm.
SIGGRAPH '87, ACM,
ACM-0-89791-227-6/87/007/01:163–169, 1987.
- [15] Timothy S. Newman and Hong Yi.
A survey of the marching cubes algorithm.
Computers & Graphics, 30(5):854 – 879, 2006.

Referências VI

- [16] Robert Mencl and Heinrich Muller.
Interpolation and approximation of surfaces from
three-dimensional scattered data points.
In *Dagstuhl '97, Scientific Visualization*, page 223–232,
Washington, DC, USA, 1999. IEEE Computer Society.
- [17] M. Kazhdan, M. Bolitho, and H. Hoppe.
Poisson surface reconstruction.
In *Proceedings of the fourth Eurographics symposium on
Geometry processing*, page 61–70, 2006.

Referências VII

- [18] Greg Turk.
Re-tiling polygonal surfaces.
In Proceedings of the 19th annual conference on Computer graphics and interactive techniques, SIGGRAPH '92, page 55–64, New York, NY, USA, 1992. ACM.
- [19] J. Daniels, C.T. Silva, J. Shepherd, and E. Cohen.
Quadrilateral mesh simplification.
In ACM Transactions on Graphics (TOG), volume 27, page 148, 2008.
- [20] Leila Floriani, Paolo Cignoni, Enrico Puppo, and Roberto Scopigno.
Surface simplification algorithms overview, 1999.

Referências VIII

- [21] P. Lindstrom and G. Turk.
Fast and memory efficient polygonal simplification.
In *Visualization '98. Proceedings*, page 279–286, 1998.
- [22] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe,
P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison,
et al.
KinectFusion: real-time 3D reconstruction and interaction
using a moving depth camera.
In *Proceedings of the 24th annual ACM symposium on User
interface software and technology*, page 559–568, 2011.
- [23] Washington.
RGB-D object dataset, 2012.

Referências IX

- [24] Carlos Alex Sander Juvêncio Gulo.
Técnicas de paralelização em GPGPU aplicadas em algoritmo para remoção de ruído multiplicativo.
Mestrado, Universidade Estadual Paulista "Júlio de Mesquita Filho", Bauru - SP, 2012.