
Simplificação e Multiresolução em Modelos Baseados em Pontos

Felipe Moura de Carvalho

Laboratório de Computação Gráfica – LCG

Sumário

- Introdução
- Pontos como Primitiva de Renderização
 - Splats
- Estruturas de Dados
- Clusterização
- Método de Simplificação de Superfícies baseado em *Splats*
 - Métricas de Erro
- Conclusão e Trabalhos Futuros

Introdução

- Pontos como primitiva de renderização e modelagem, sofre um “renascimento”
- Duas razões principais:
 - Crescimento na complexidade de modelos poligonais
 - Advento dos Scanners e Fotografia 3D

Introdução

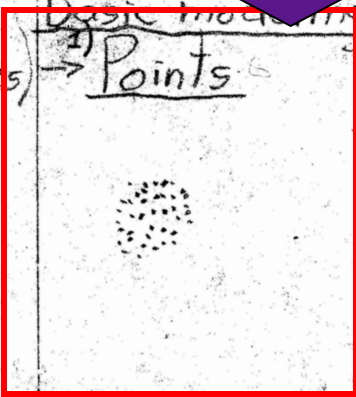
- É uma idéia antiga:
 - Foi usado na década de 70 para representar explosões de naves espaciais em *video games*
 - *C.Csuri, et al, “Towards an interactive high visual complexity animation system,” Proc SIGGRAPH 79.*
 - *W. Reeves, “Particle systems - a technique for modeling a class of fuzzy objects, “Proc. SIGGRAPH 83.*
 - *Levoy e Whitted - The use of Points as a Display Primitive 1985.*

Texture forming primitive

(for discussion with Turner, Sept 18, where I talked about "meta-formats")

Basic modeling primitives for 3d:

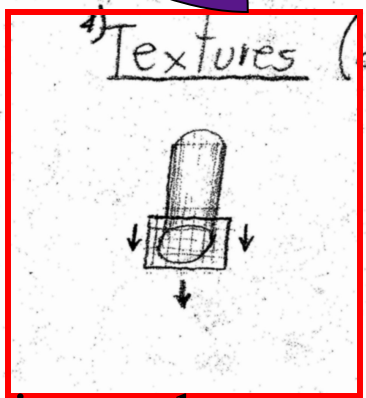
(à la Reeves)



1) Points

2) Lines

3) Polygons



4) Textures (à la Whitted-Brushes)

texturas são simplesmente um conjunto de pontos

Basic Dimensionality:

0d

1d

2d homogeneous

2d non-homogeneous

(defined by 0d points)

(defined by 1d edges)

(defined by 2d regions (pixels))

Typical extensions:



grass stalks



curved lines



smooth shading



bump mapping

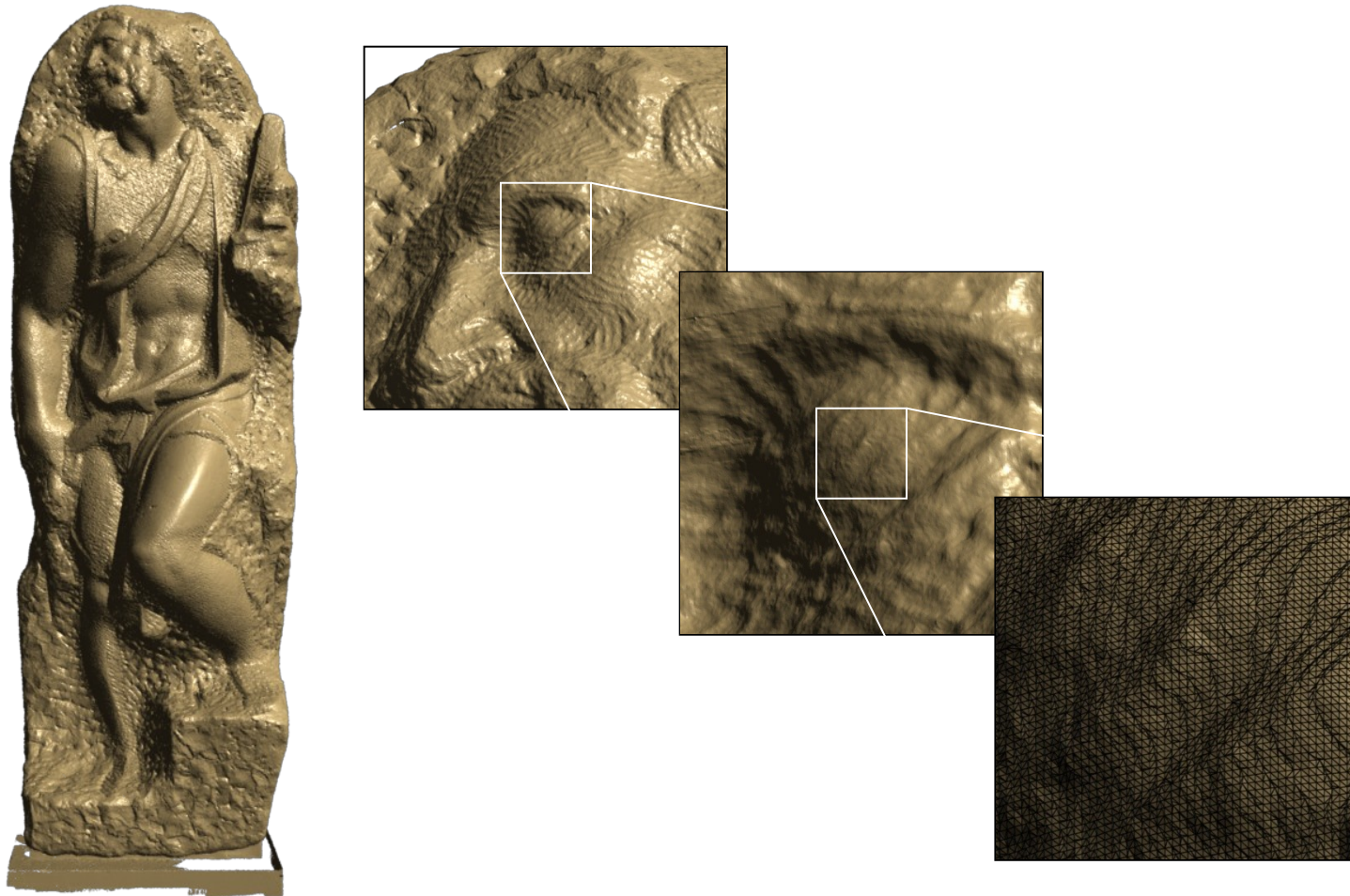
Introdução

- 3D Scanning

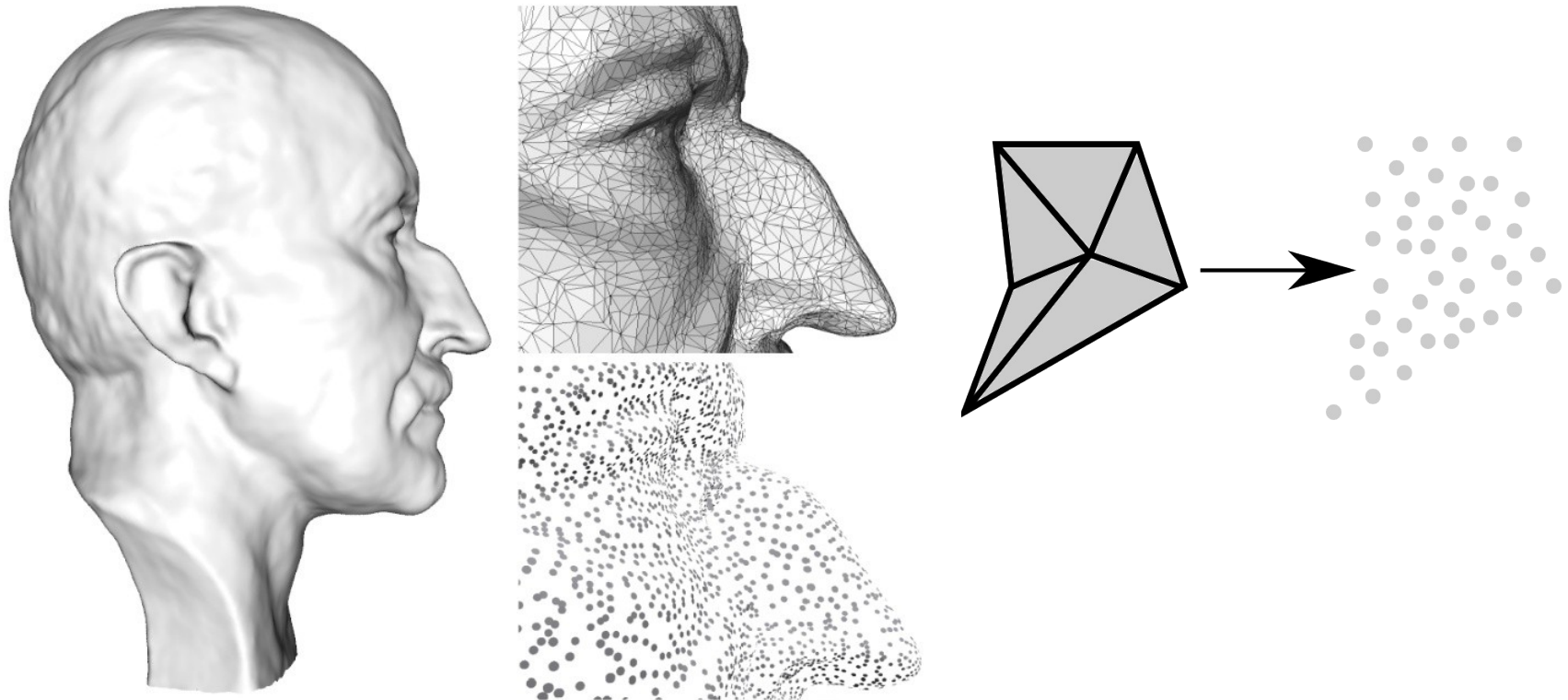


Introdução

- Modelo escaneado com resolução de 0.25mm : 102,868,637

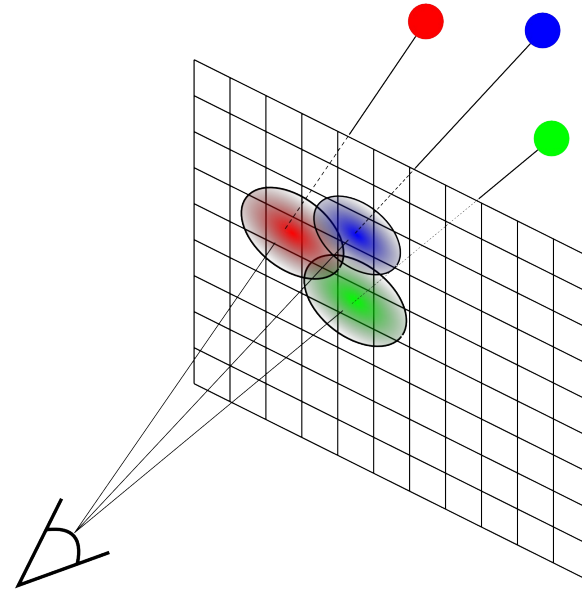
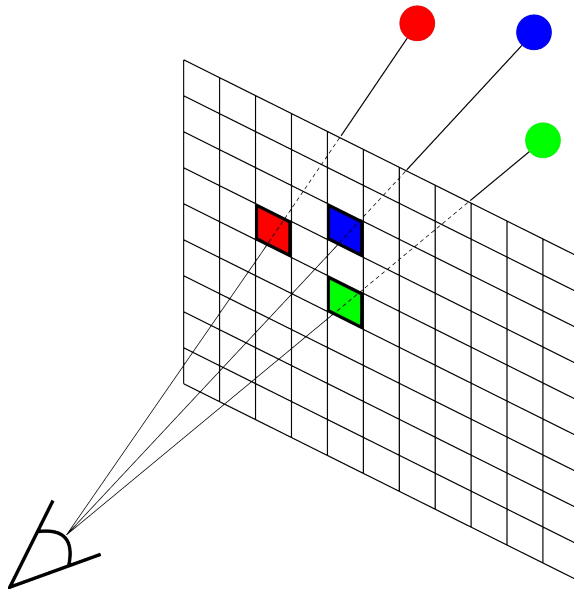


Renderização baseada em Pontos



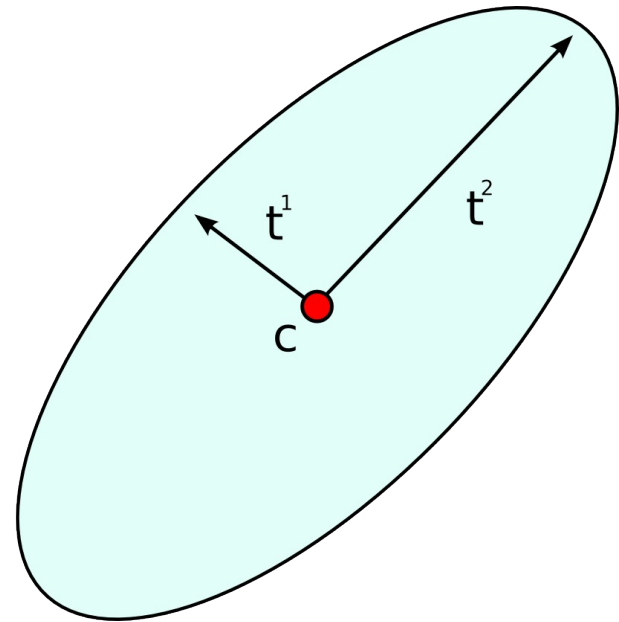
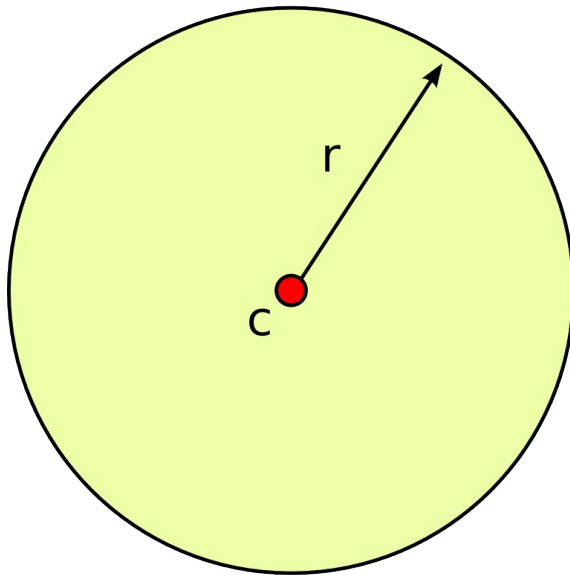
Renderização baseada em Pontos

- Problema: buracos (densidade não suficiente ...)
- *Splats*: extensão de pontos puros
- Representação: pequena superfície planar (circulo ou elipse)
- Associado a um valor de densidade local



Renderização baseada em Pontos

- Tipos de *Splats*



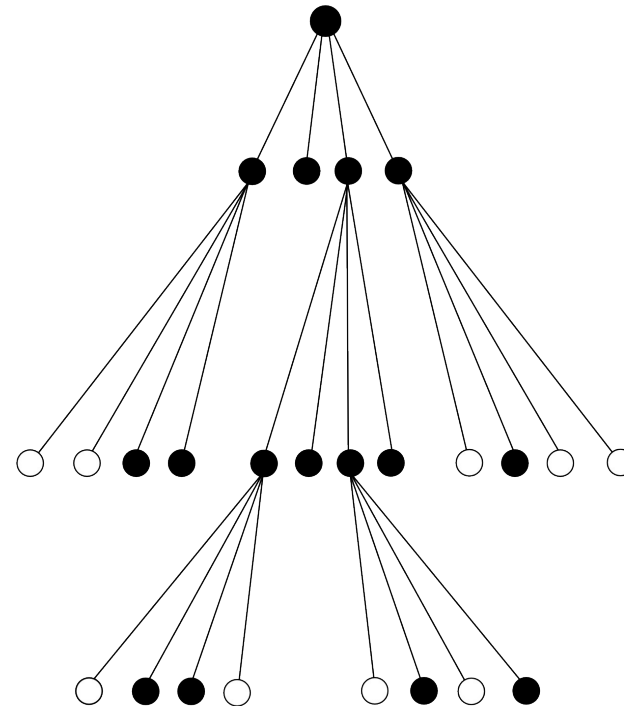
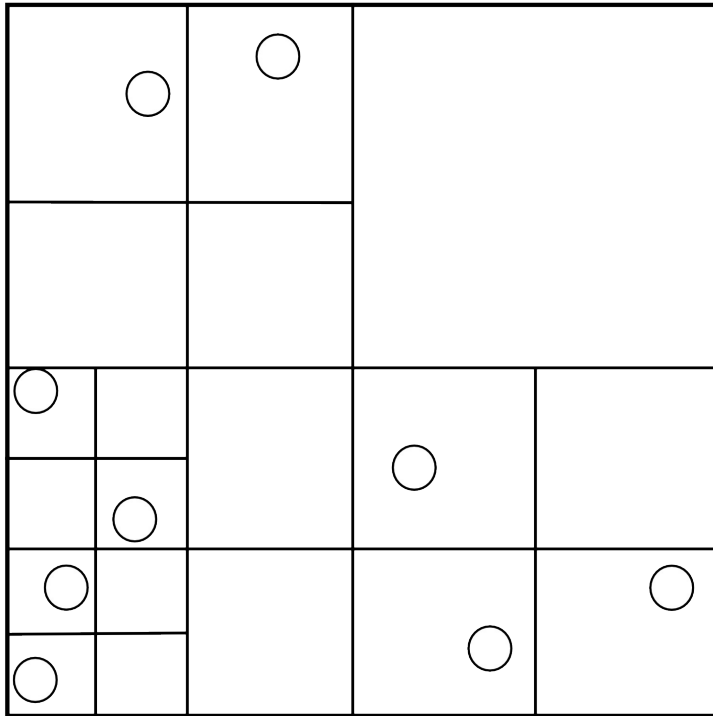
Modelos Complexos

- Pontos possuem muitas vantagens
 - Representação simples
 - Ausência de conectividade
- Problema:
 - Modelos com milhões ou bilhões de amostras.
- Muitas aplicações requerem modelos menos complexos:
 - Armazenamento
 - Transmissão
 - Modelagem
 - Renderização

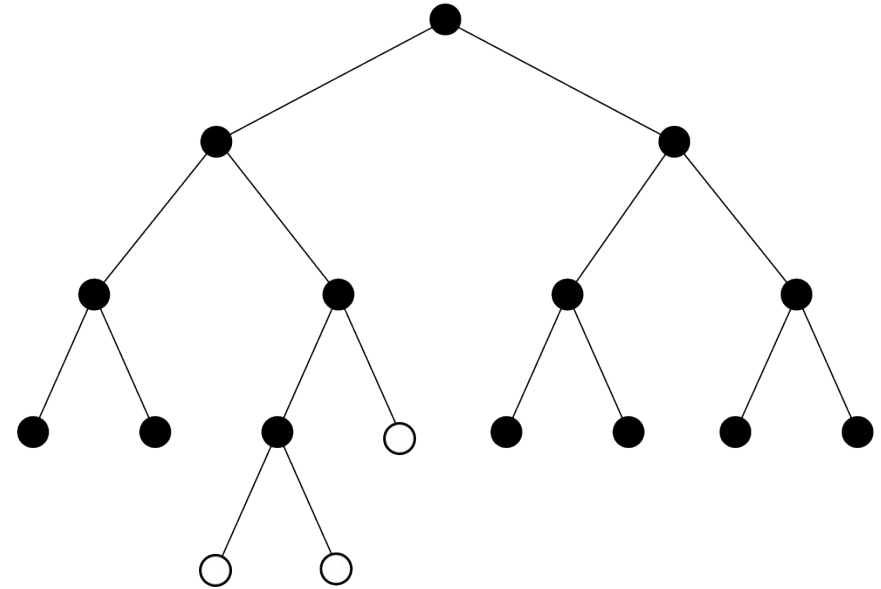
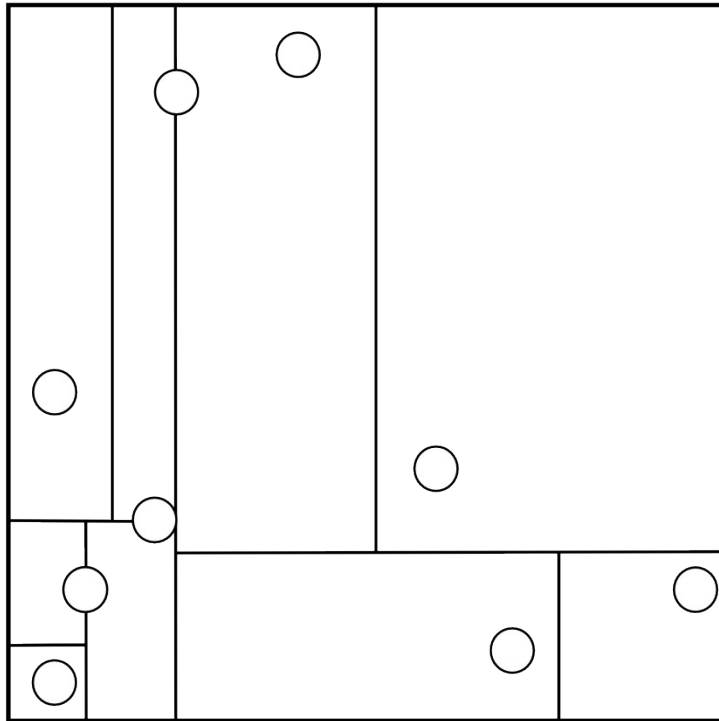
Estrutura de Dados

- Estruturas de Dados :
 - Multiresolução
 - Nível de Detalhes (LOD)
 - Busca na Vizinhança
- Estruturas de Dados Espaciais são as mais usadas:
 - *Octrees*
 - *Kd-trees*
 - Hierarquia de Esferas Envolventes - *BVH*

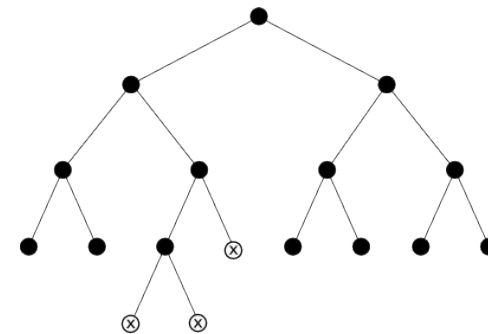
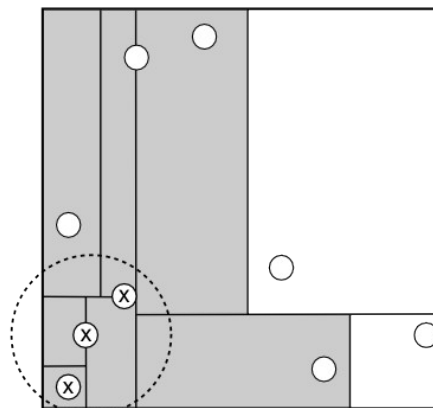
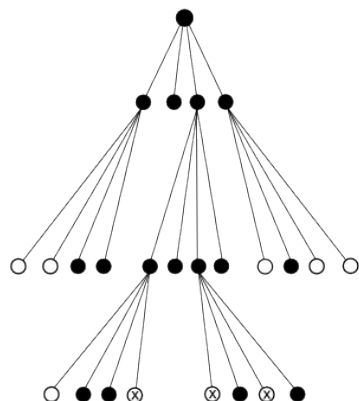
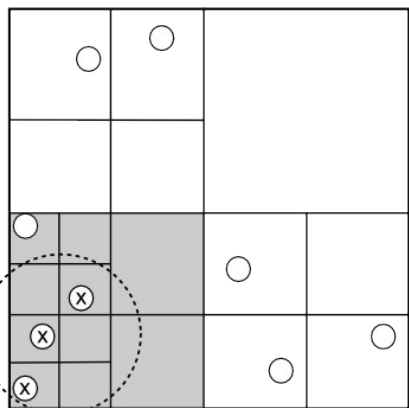
Octrees



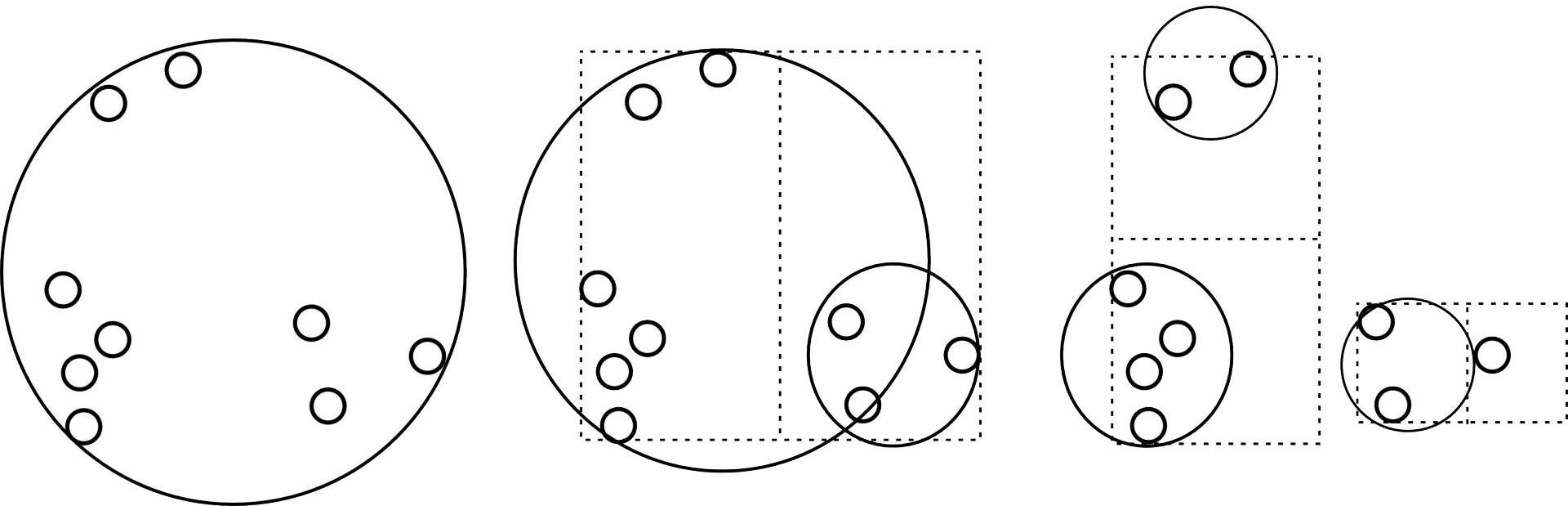
Kd-trees



Busca

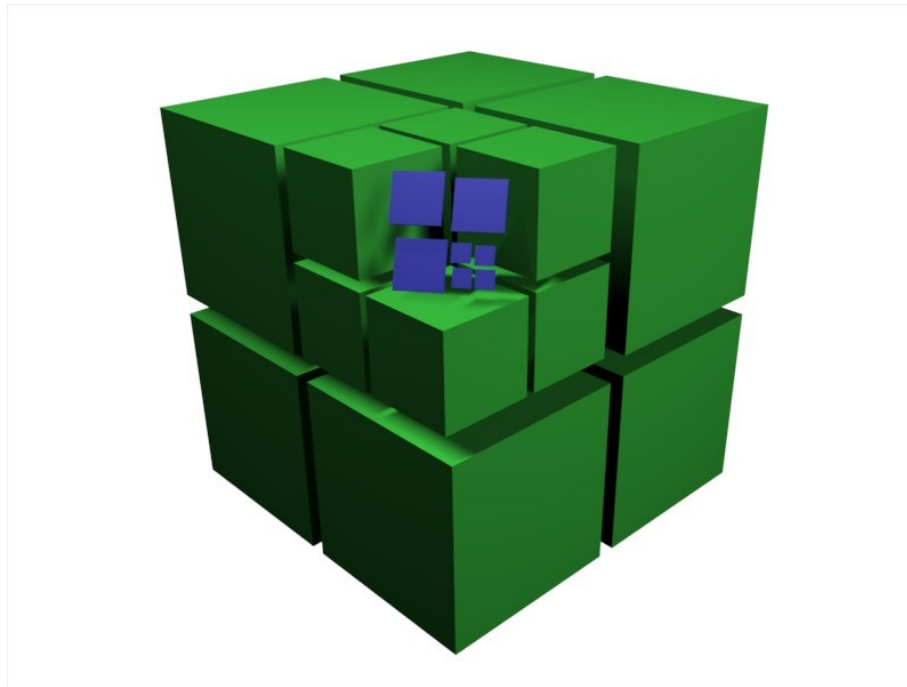


BVHs



Volume Surface Trees

- Boubekour, Tamy and Heidrich, Wolfgang and Granier, Xavier and Schlick, Christophe"- Volume-Surface Trees - 2006



Volume Surface Trees

- Combinação: octree/quaternion
- Decomposição: Volume-Superfície

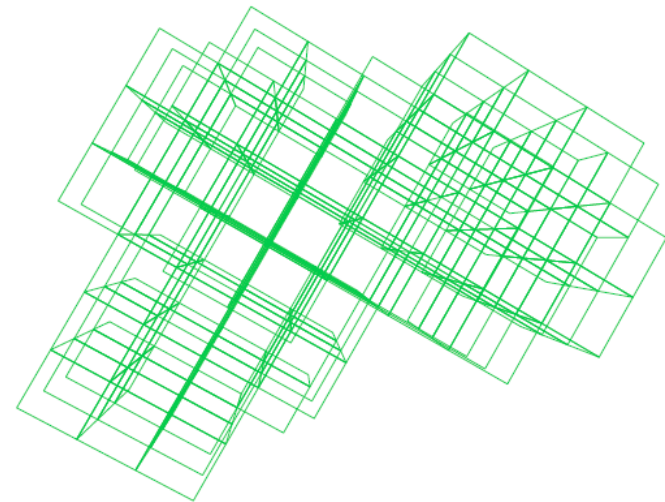
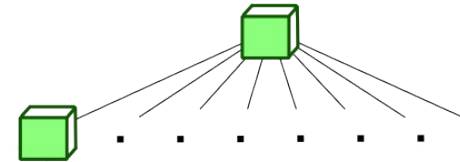


*Caixa Envolvente
do Modelo*



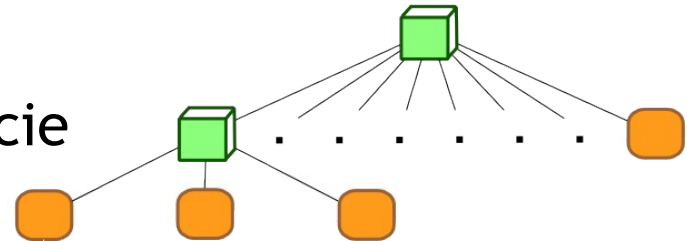
Volume Surface Trees

- Combinação: octree/quaternion
- Decomposição: Volume-Superfície
 - V-Node (Volume)



Volume Surface Trees

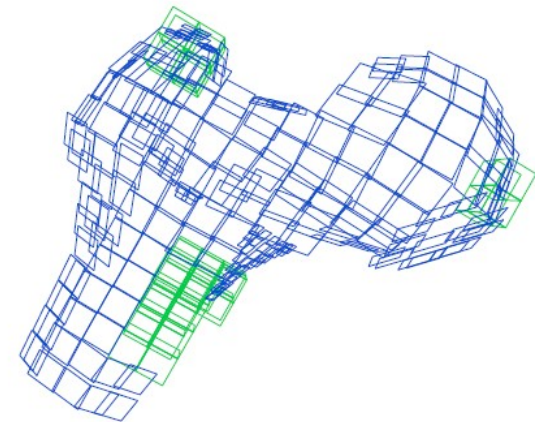
- Combinação: octree/quaternion
- Decomposição: Volume-Superfície
 - V-Node (Volume)
 - T-Node (Transição)



V-Node

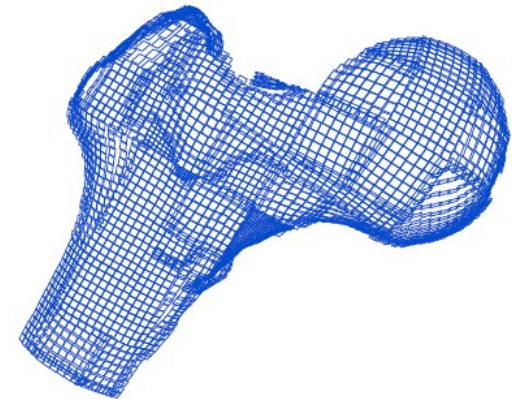
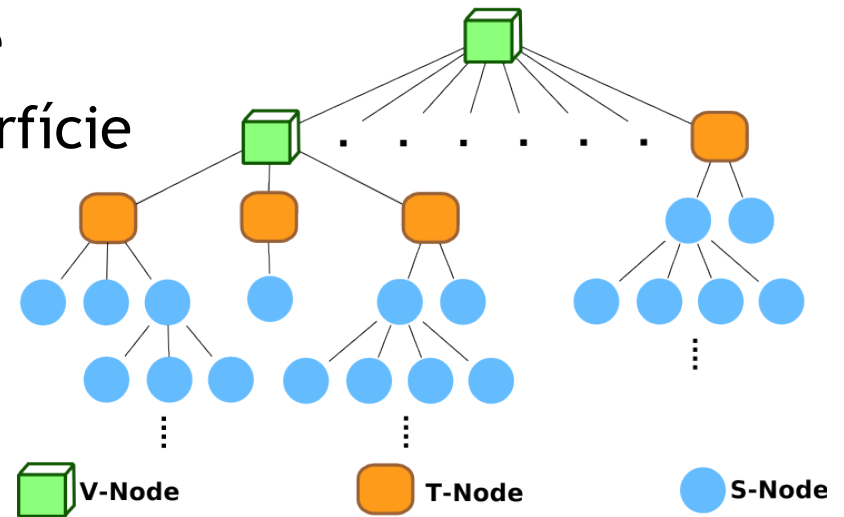


T-Node



Volume Surface Trees

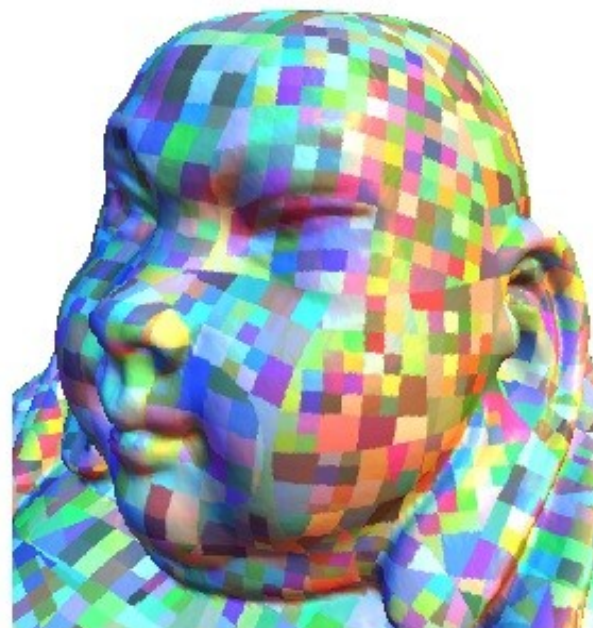
- Combinação: octree/quaternion
- Decomposição: Volume-Superfície
 - V-Node (Volume)
 - T-Node (Transição)
 - S-Node (Superfície)



Volume Surface Trees



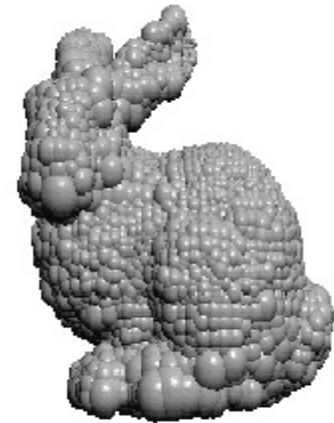
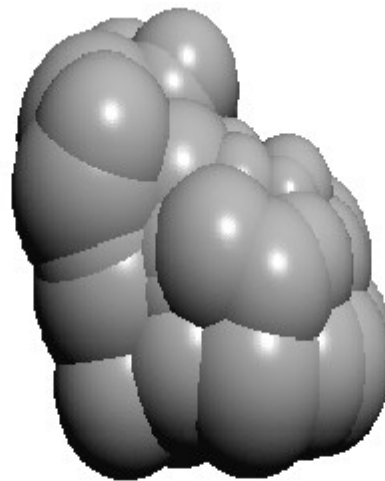
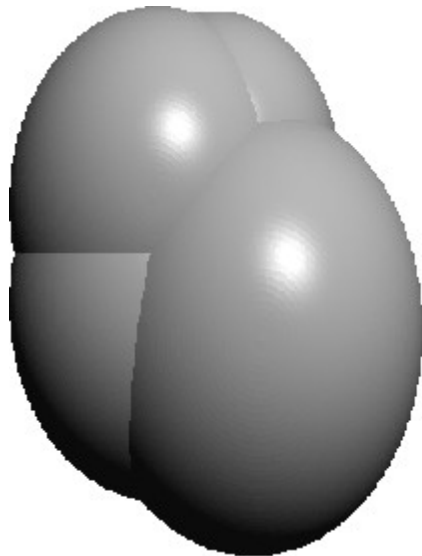
(a) Octree clustering



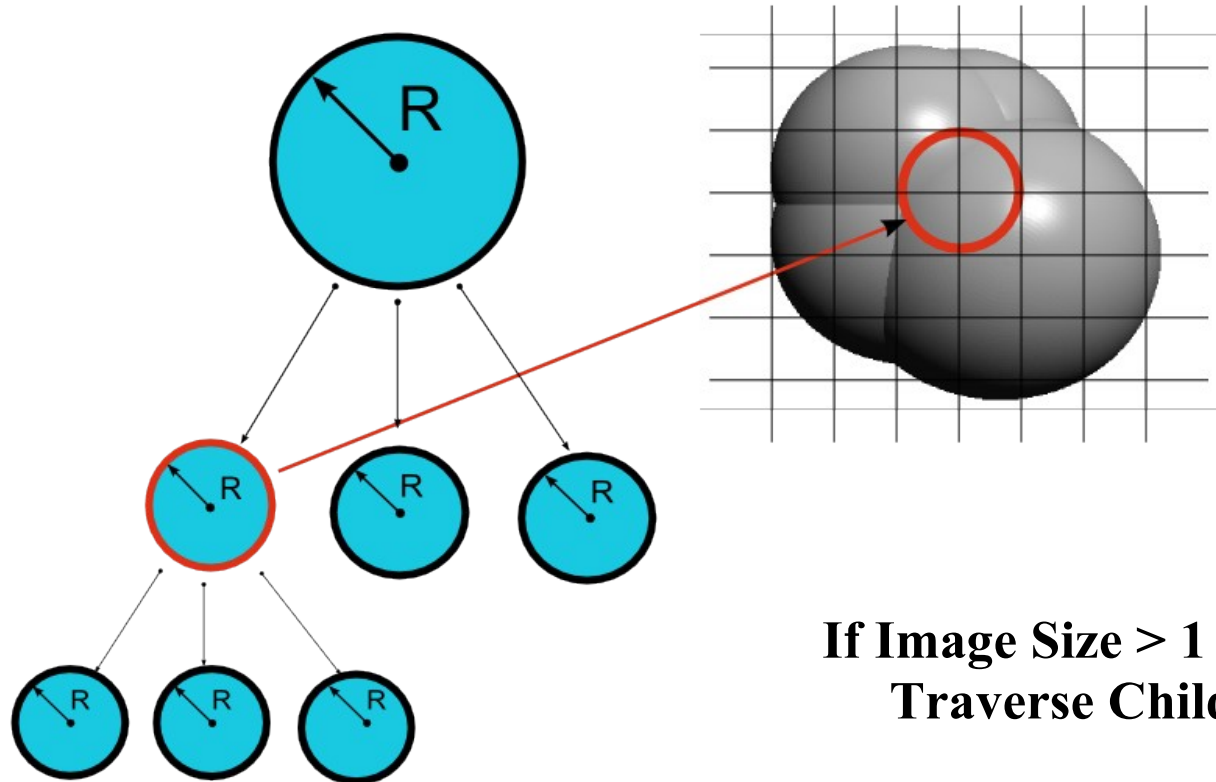
(b) VS-Tree clustering

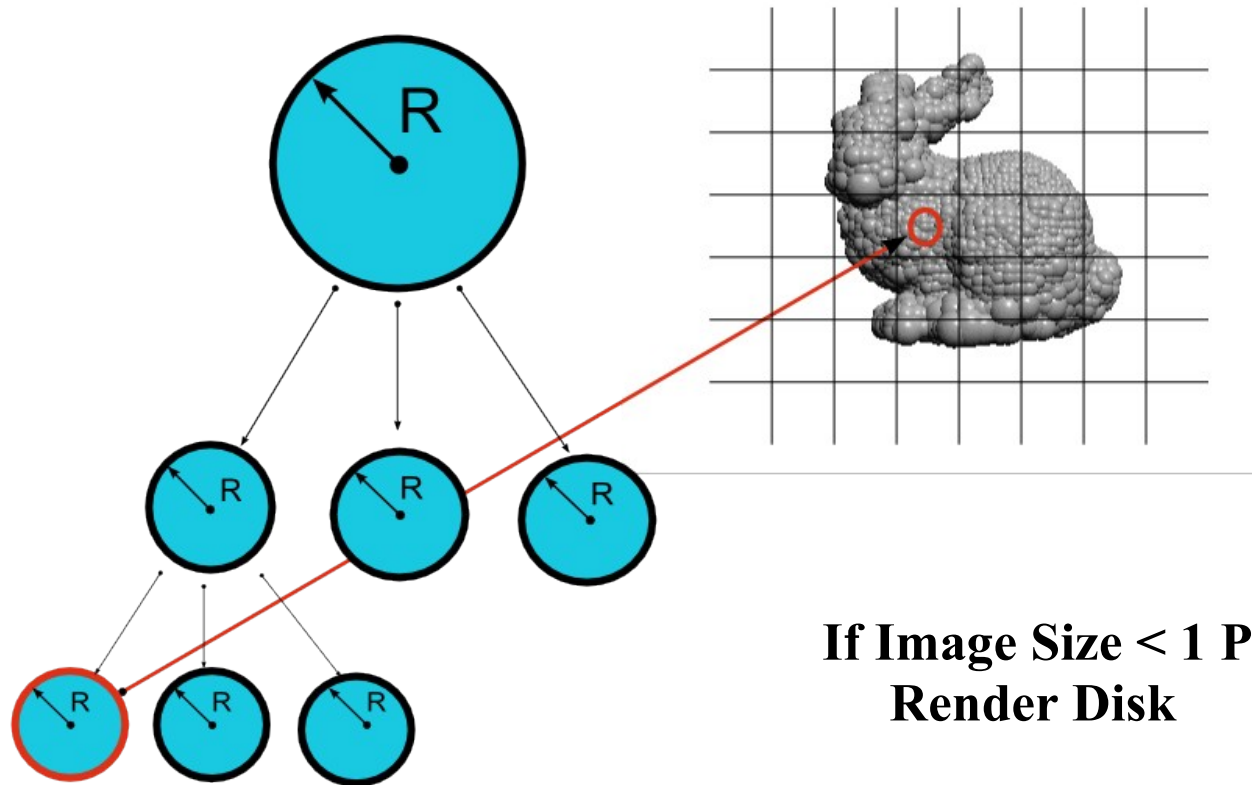
QSplat

- Szymon Rusinkiewicz , Marc Levoy, QSplat: a multiresolution point rendering system for large meshes - 2000



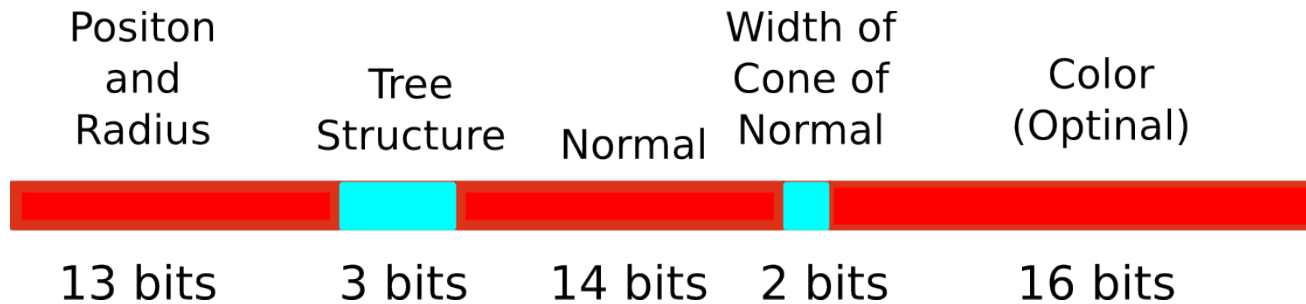
- Renderização recursiva





**If Image Size < 1 Pixel :
Render Disk**

QSplat



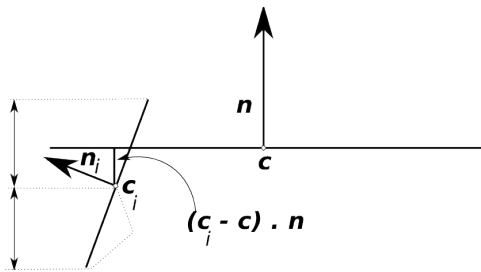
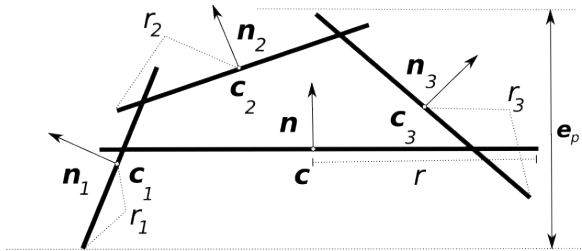
Sequential Point Trees

- Sequential Point Trees - Carsten Dachsbacher, Christian Vogelgsang, Marc Stramminger - 2003
- SPT
 - Seleção adaptativa de pontos
 - Processamento sequencial
- Renderização
 - Lista de pontos e atributos pré computadas
 - Renderiza segmentos contínuos na lista
 - CPU apenas escolhe os índices da lista

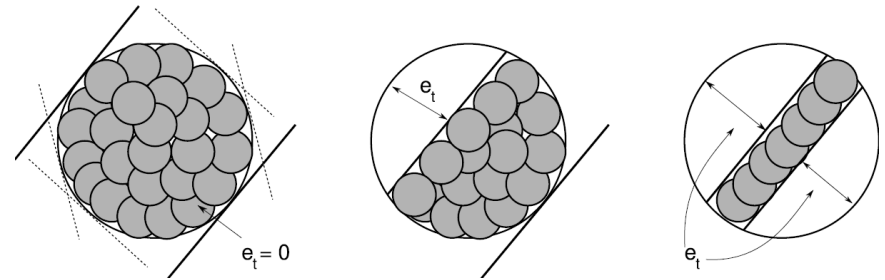


Sequential Point Trees

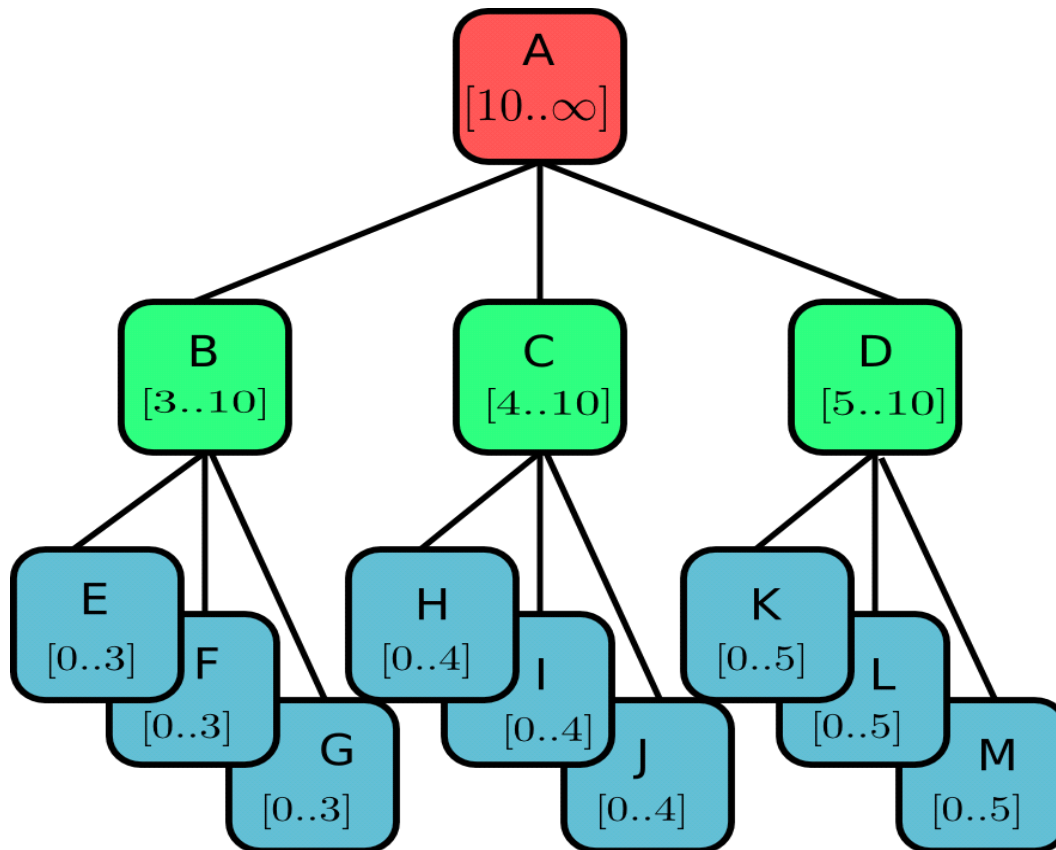
- Erro perpendicular



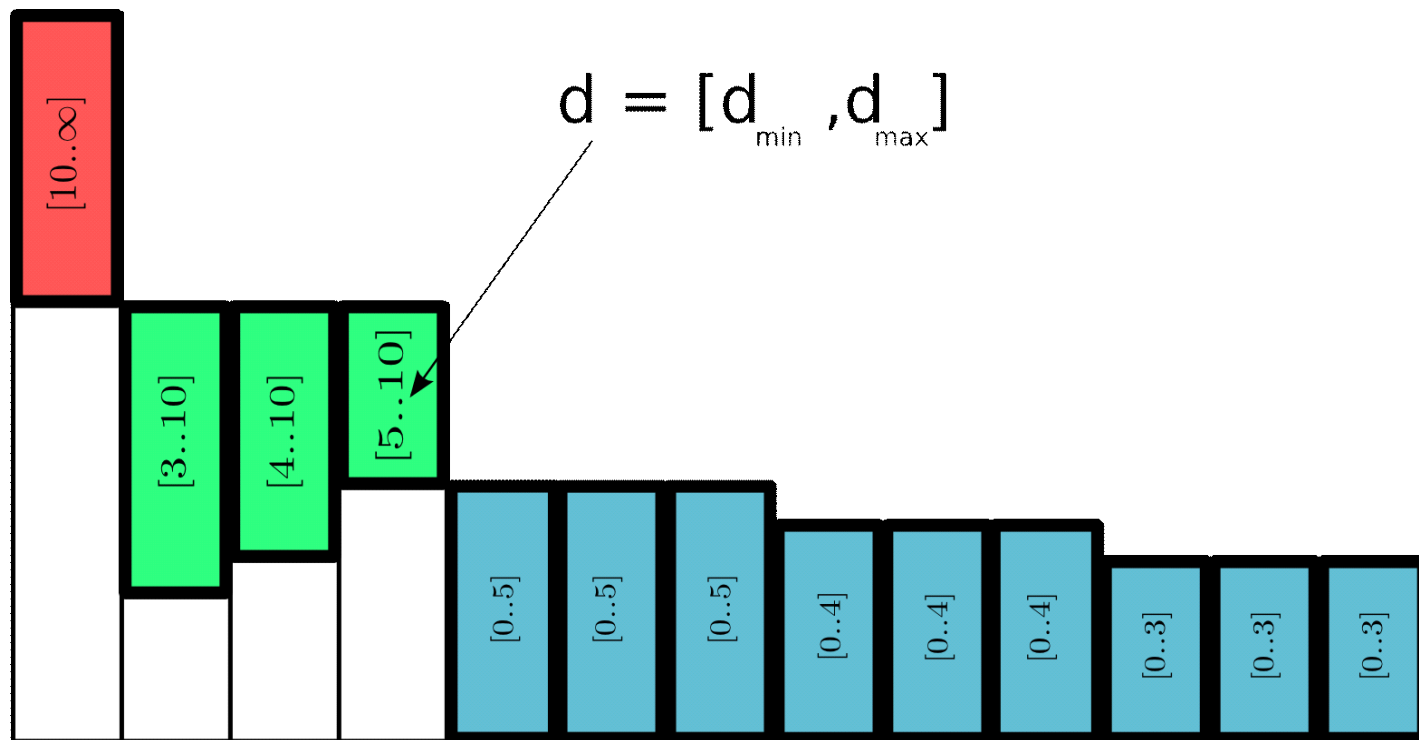
- Erro tangencial



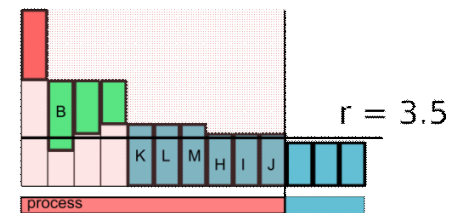
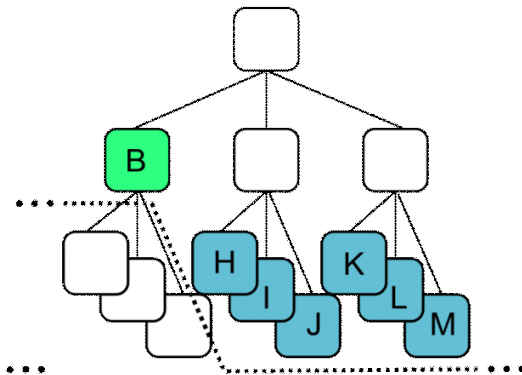
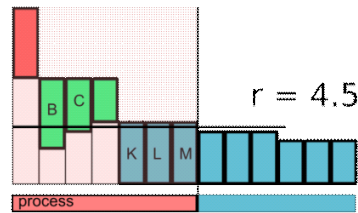
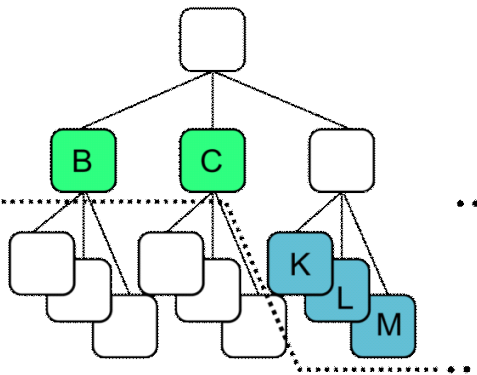
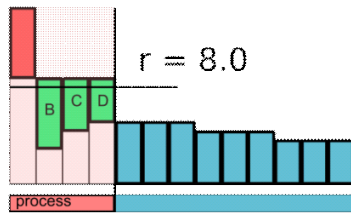
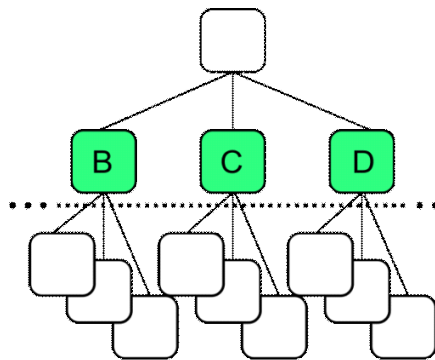
Sequential Point Trees



Sequential Point Trees

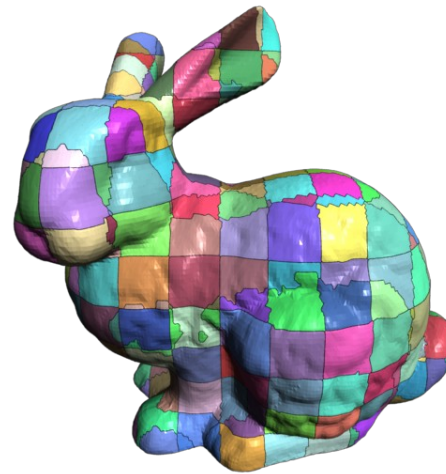
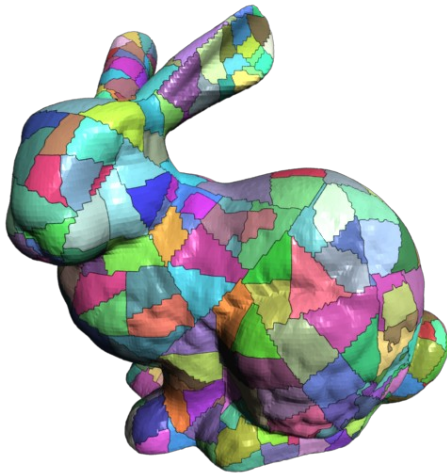


Sequential Point Trees



Clusterização

- Clusterização Hierárquica
 - SPT e Qsplat:
 - Baseados em subdivisão espacial



Clusterização

- Clusterização por agregação
 - Adicionar novas amostras até que um tamanho mínimo seja alcançado
 - Problema critério de agregação

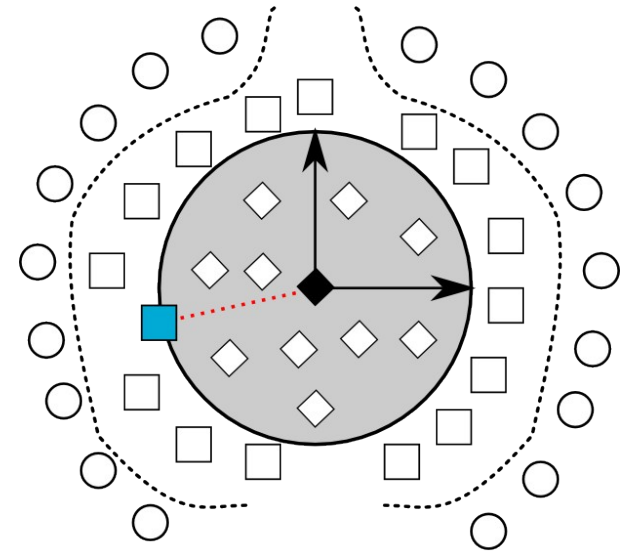
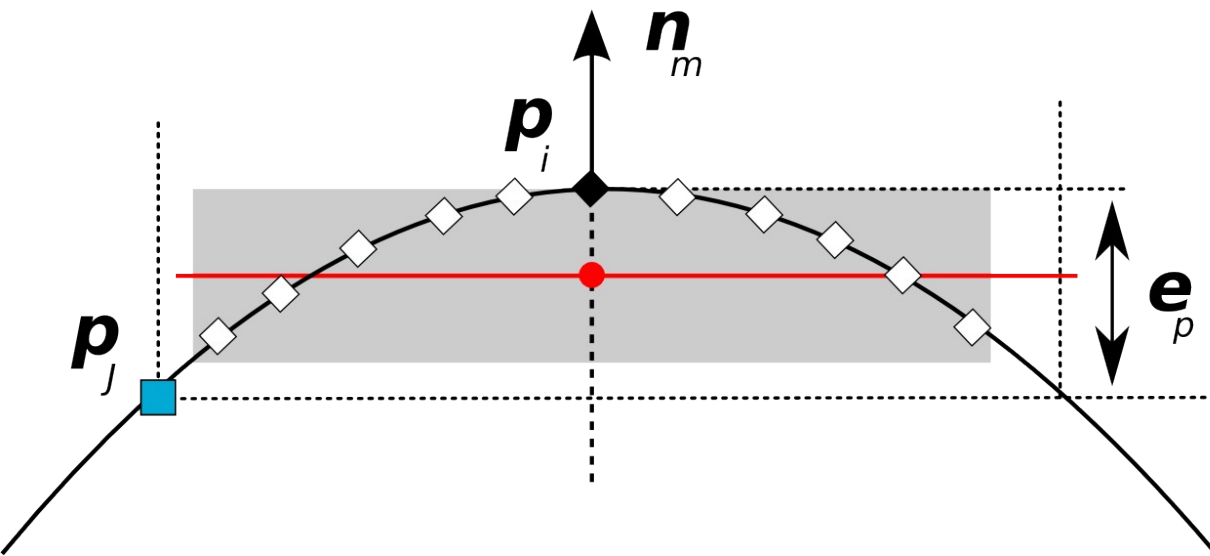


Método Proposto

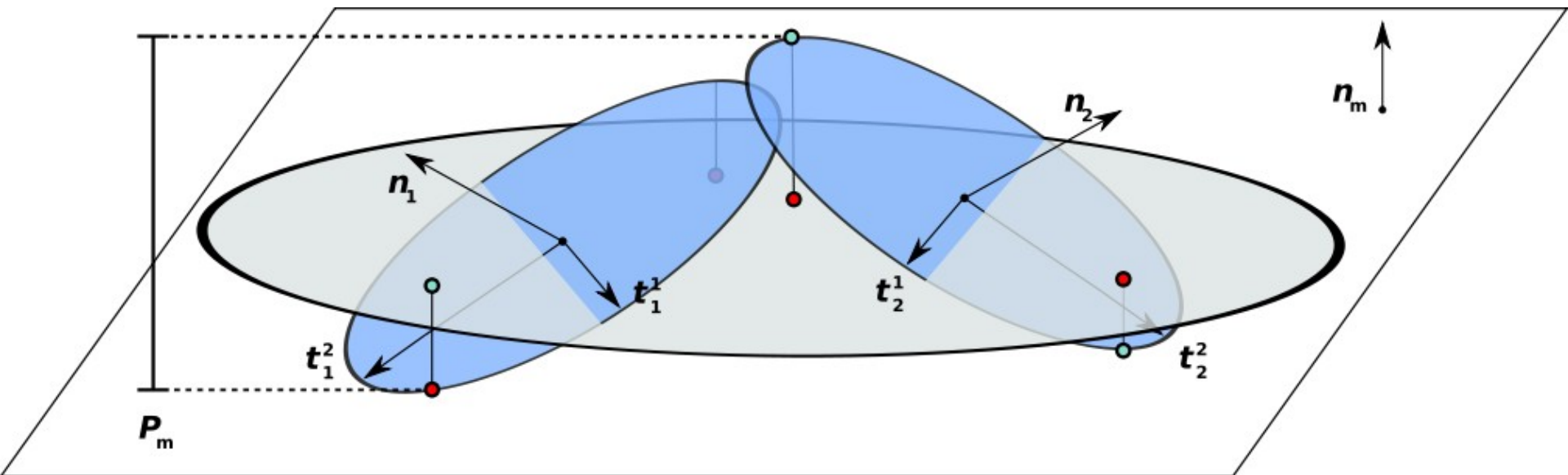
Método Proposto

- Clusterização por crescimento de região
- Sementes aleatórias
- Crescendo regiões a partir dos seus k-vizinhos
- Para controlar o crescimento: dois erros
 - Erro Normal
 - Identifica regiões planares
 - Erro de Cobertura
 - Nos fornece uma medida de cobertura

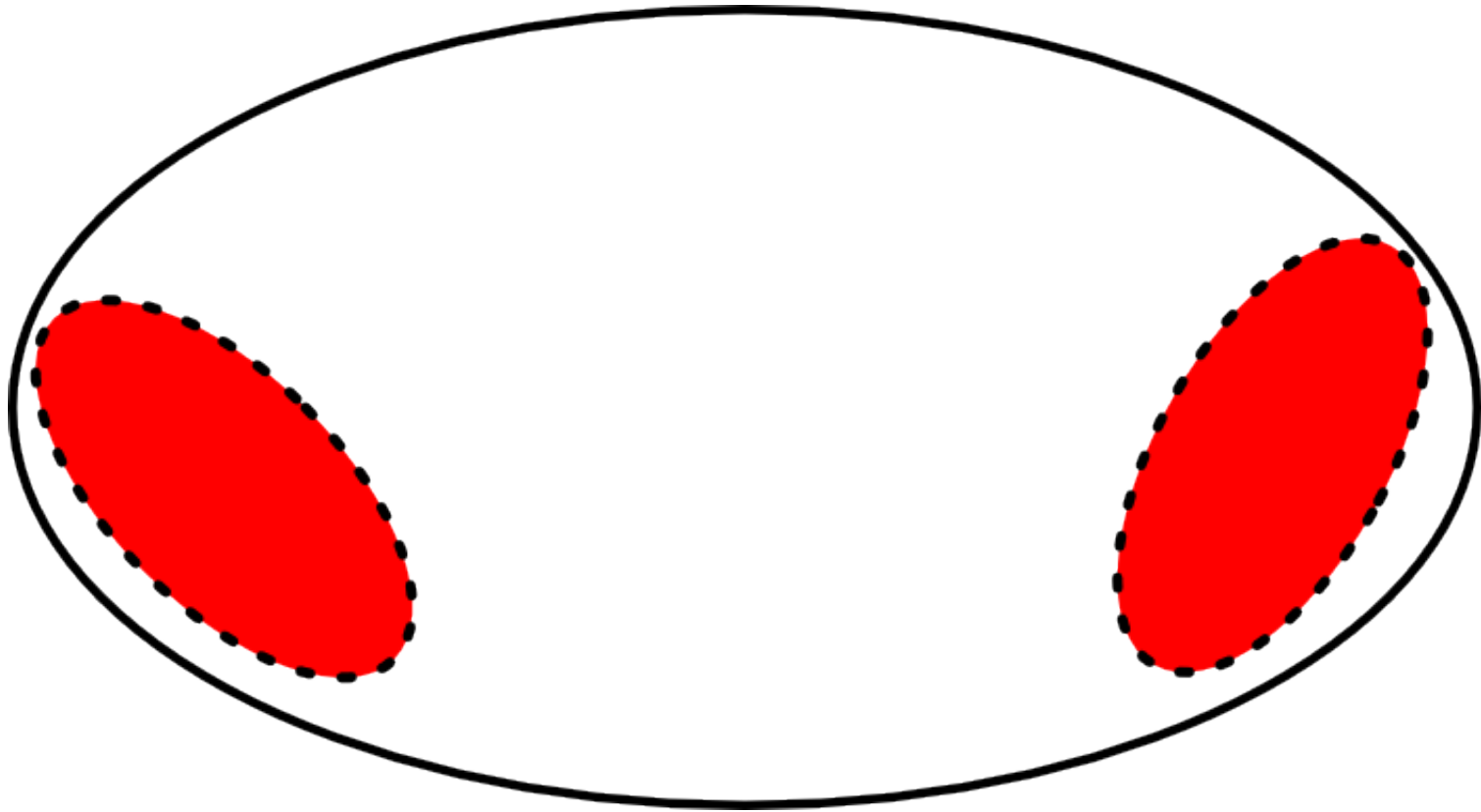
Método Proposto



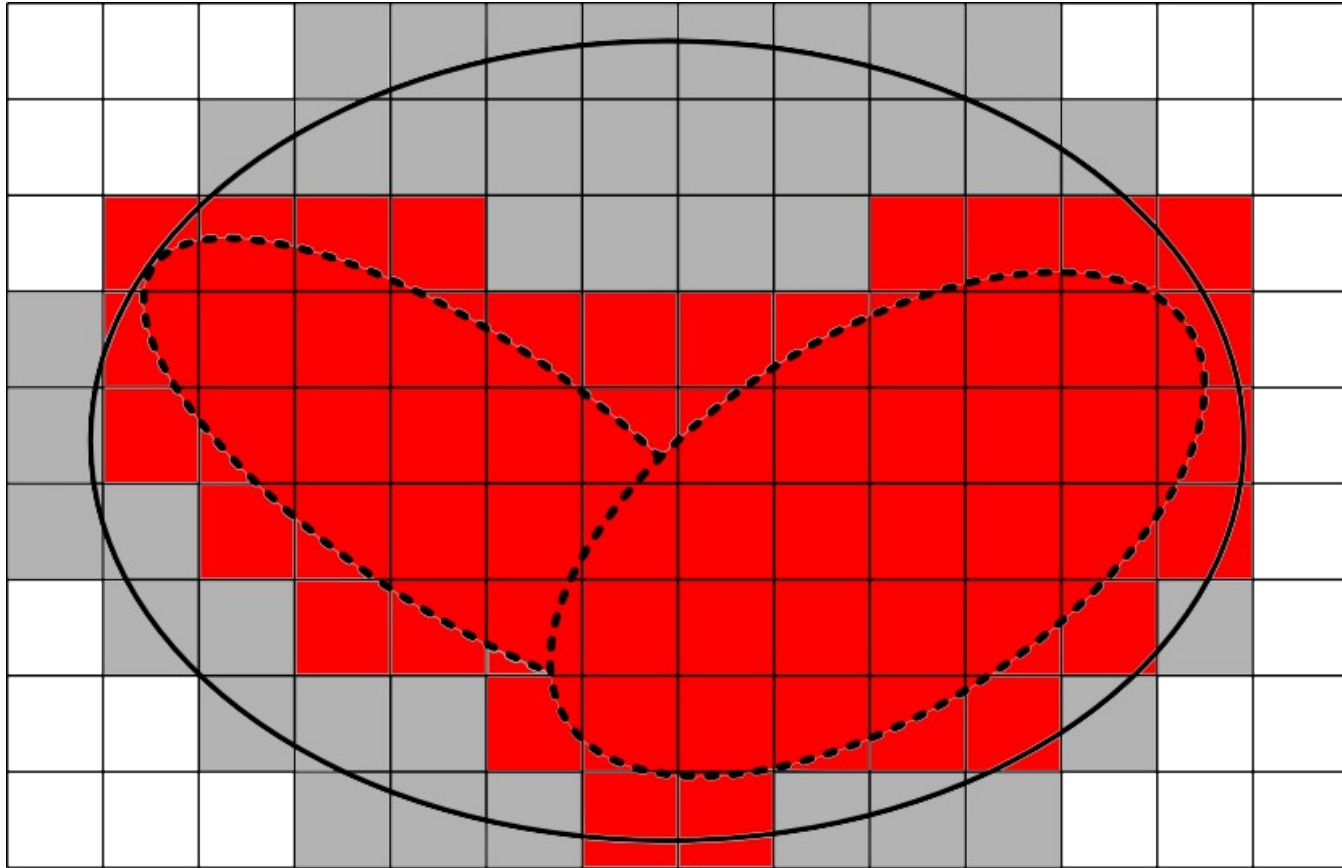
Erro Normal



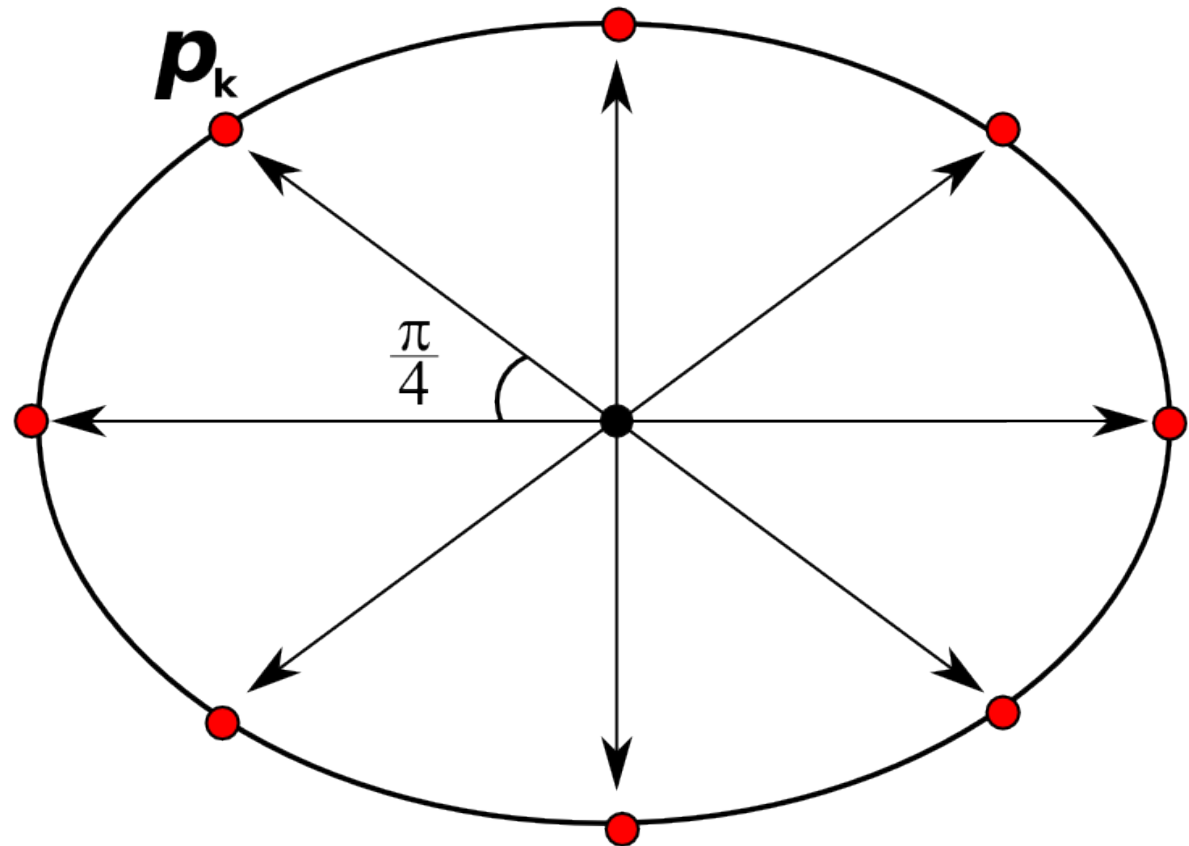
Erro Cobertura



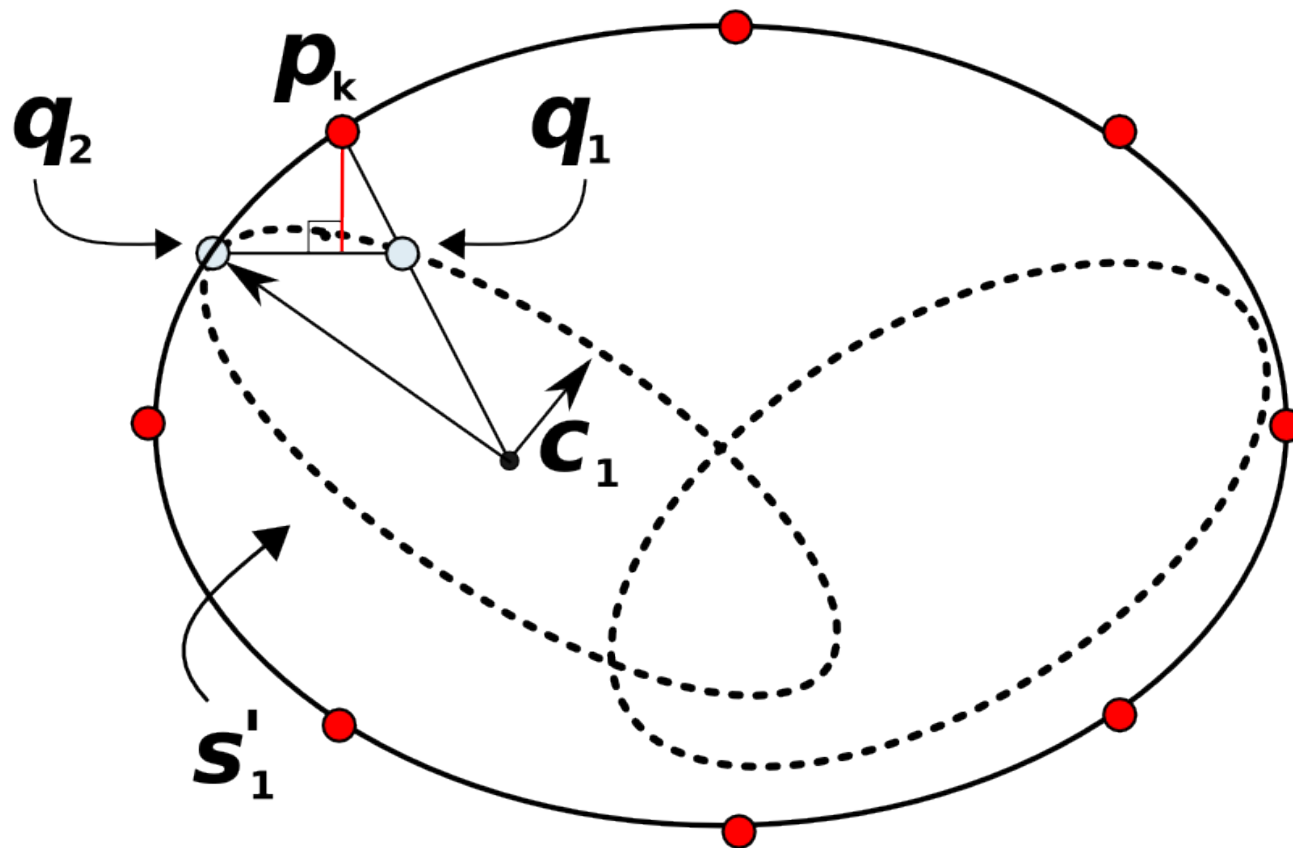
Erro Cobertura



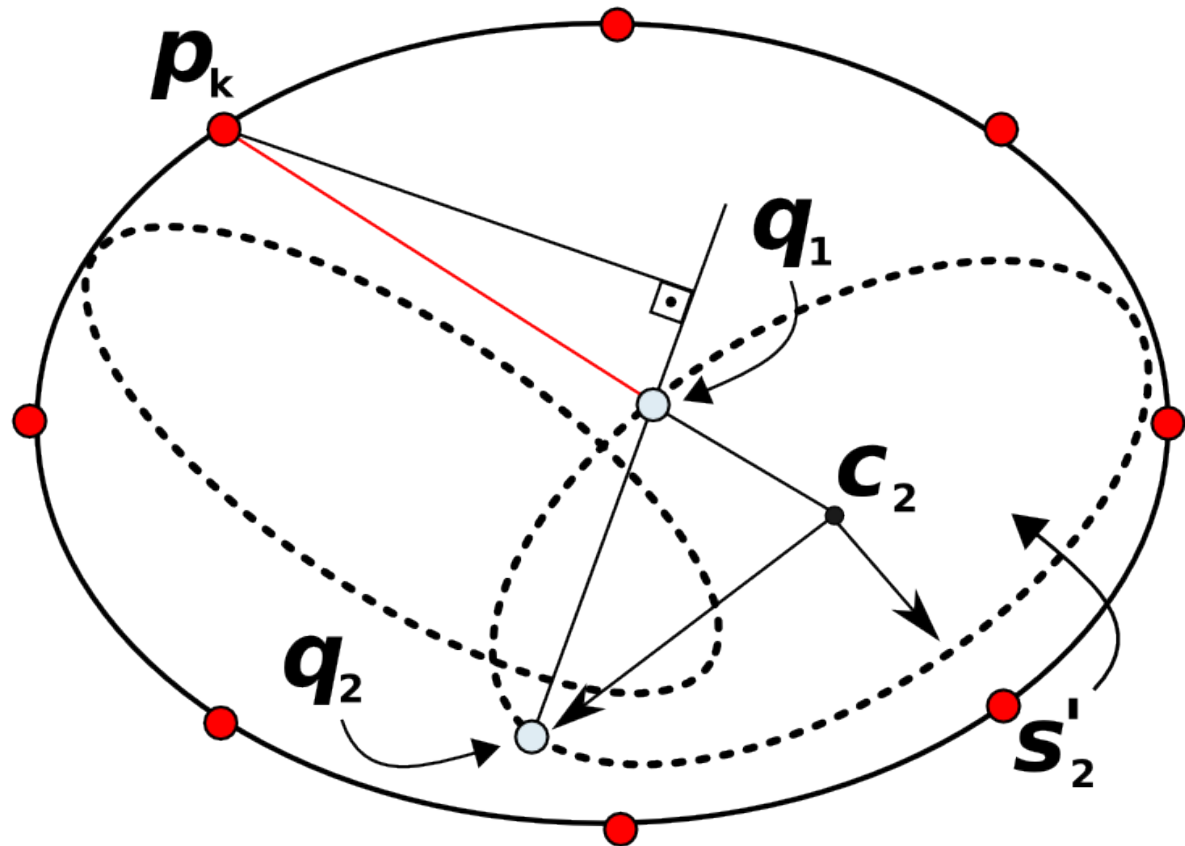
Erro Cobertura



Erro Cobertura



Erro Cobertura



Junção dos *Splats*

Novo Centro e Normal

$$n_m = \frac{\sum (s_i) \cdot n_i}{\sum (s_i)}$$

$$c_m = \frac{\sum (s_i) \cdot c_i}{\sum (s_i)}$$

Extensões

$$\frac{\sqrt{\lambda_1}}{\sqrt{\lambda_2}}$$

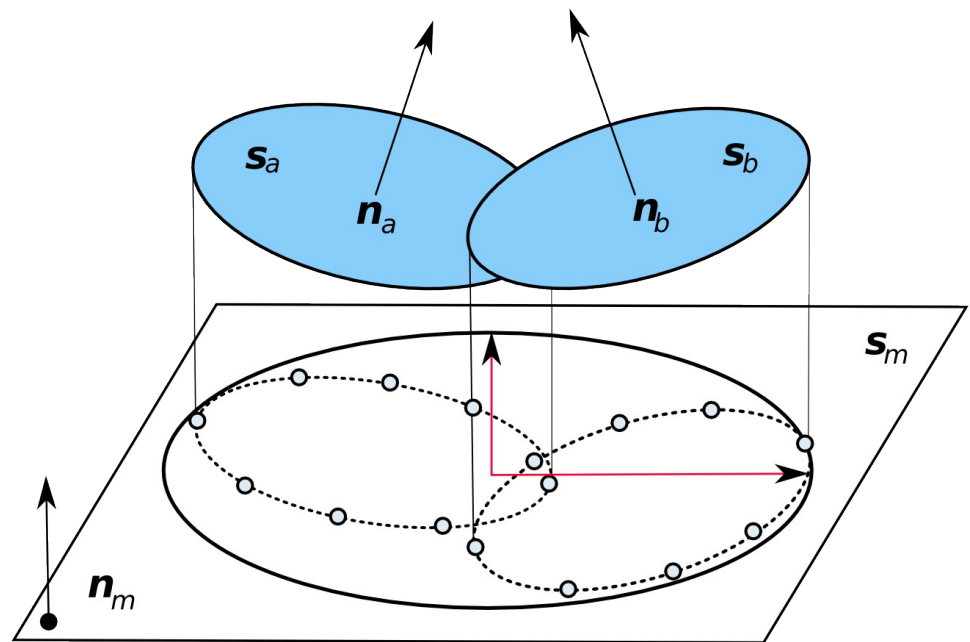
$$(t_m^1) = \sqrt{\lambda_1} \cdot \bar{e}$$

$$(t_m^2) = \sqrt{\lambda_2} \cdot \bar{e}$$

PCA

$$\lambda_1 \geq \lambda_2 \geq \lambda_3$$

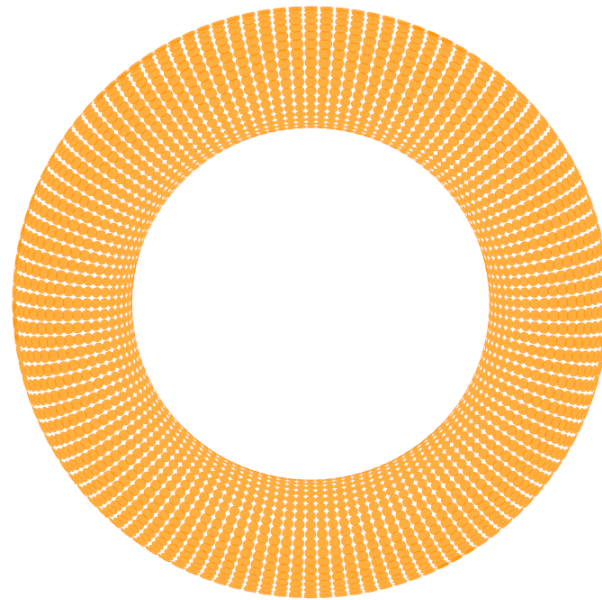
$$v_1 \geq v_2 \geq v_3$$



Resultados

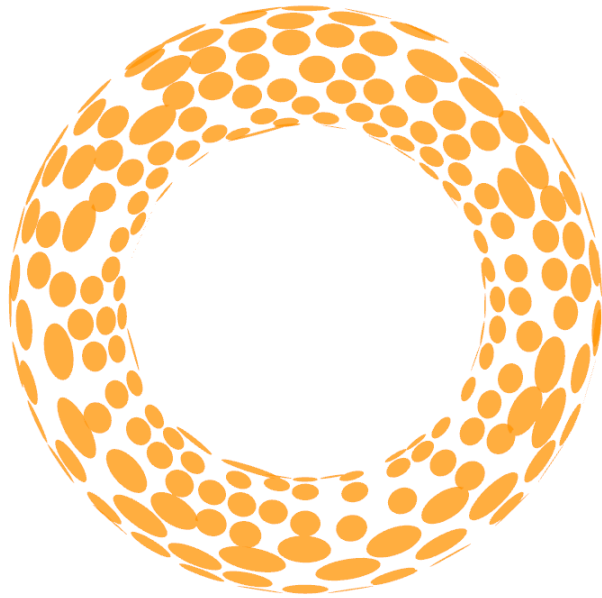
Resultados

- 4800 pontos

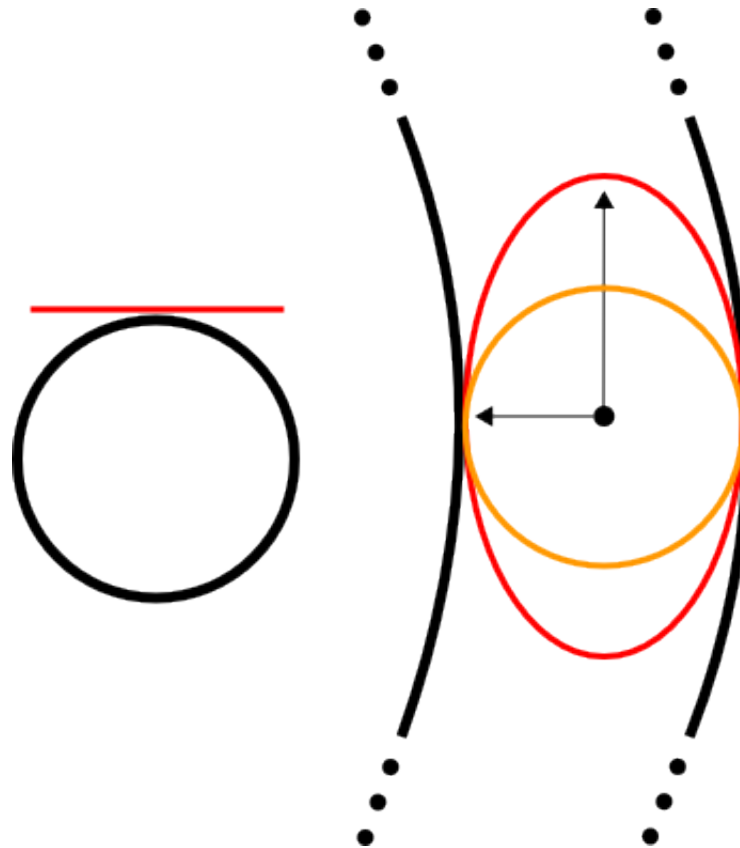


Resultados

- 378

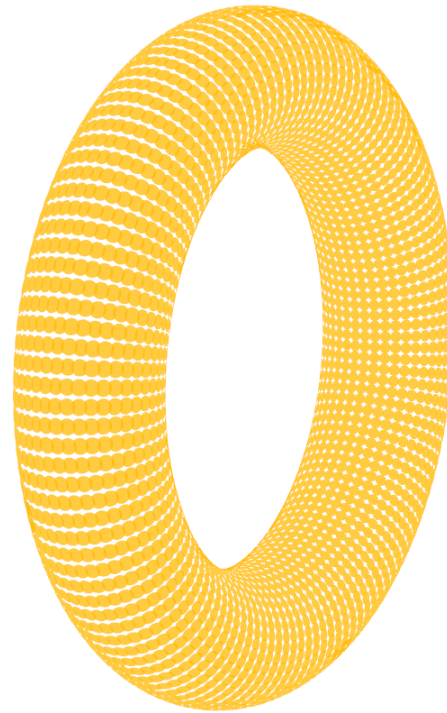


Resultados



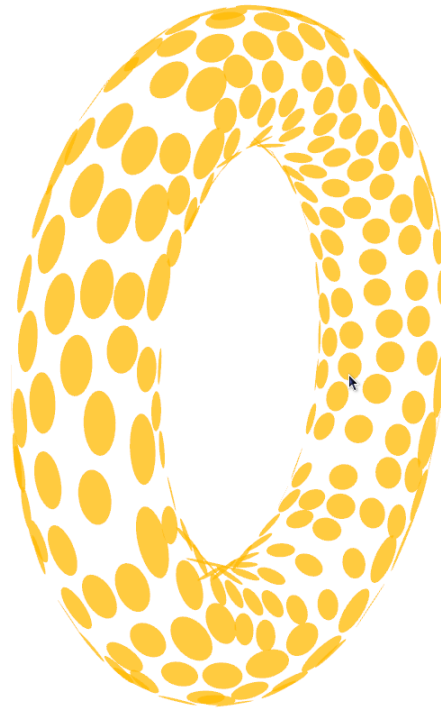
Resultados

- 4800 pontos



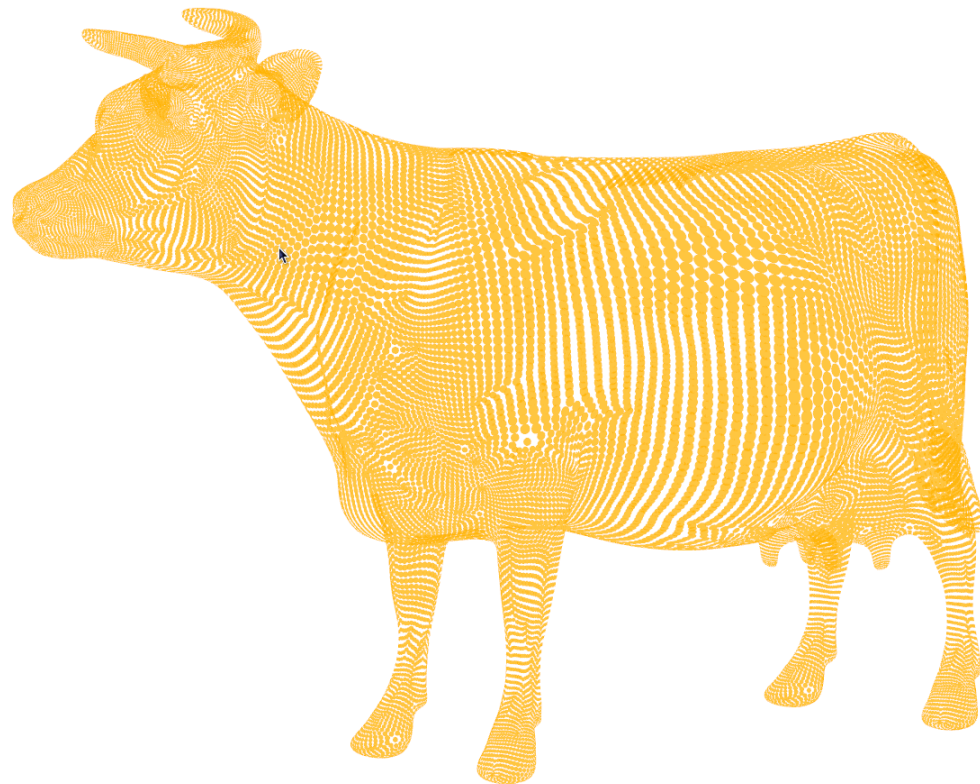
Resultados

- 378 pontos



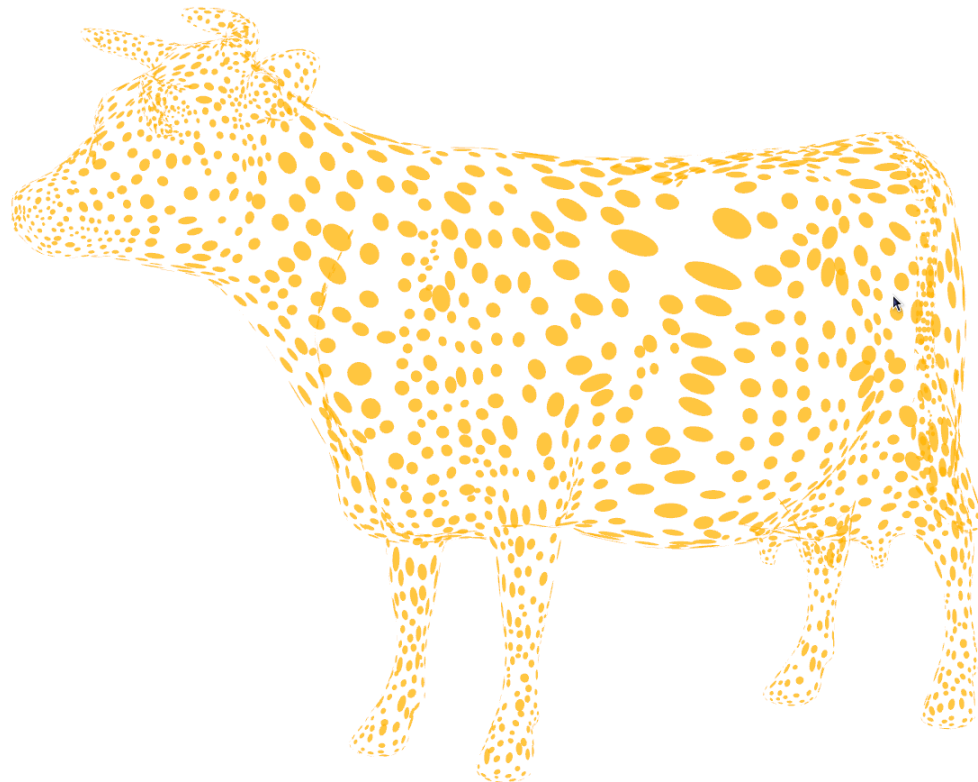
Resultados

- 45000

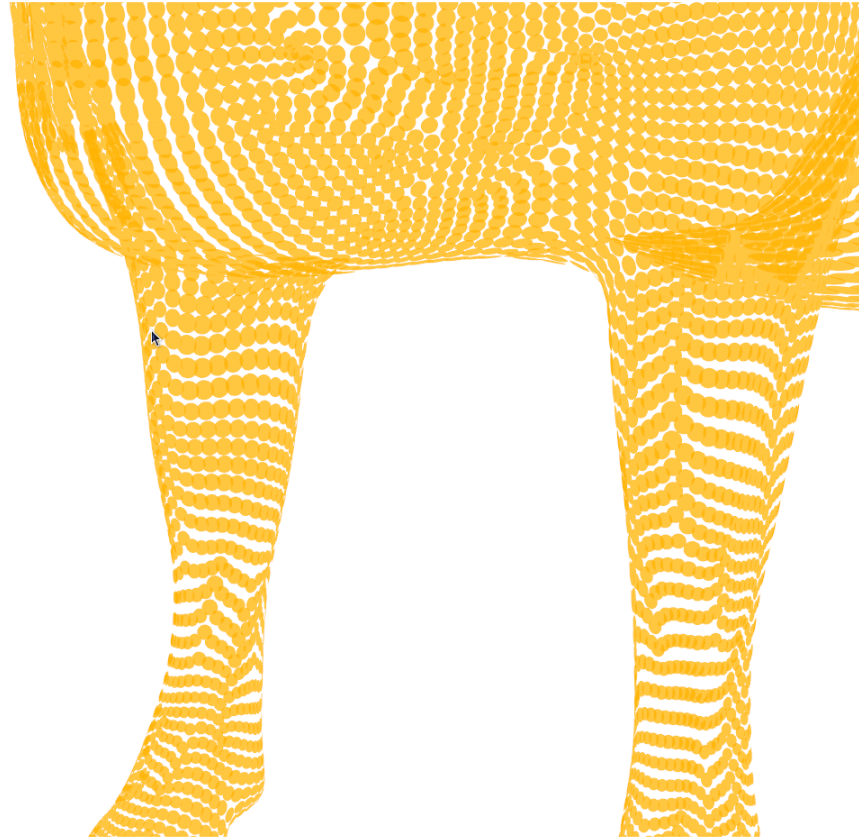


Resultados

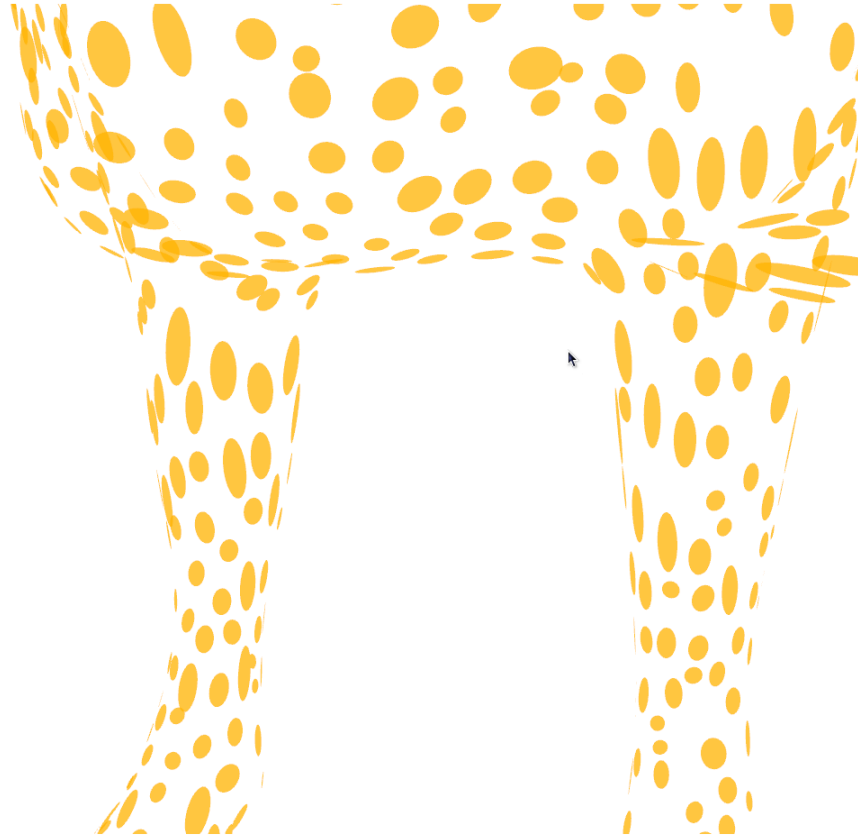
- 4000 pontos



Resultados



Resultados



Conclusões

- Clusterização Incremental
- Trabalhar diretamente com *Splats*
 - Toda a geometria é considerada
 - Evita buracos
- *Splats* elípticos permitem uma melhor aproximação
 - Se adaptam à curvatura do modelo
 - Necessita de menos amostras

Trabalhos Futuros

- Out of Core: Renderização e Simplificação
- Melhor distribuição dos Clusters
- Forma mais eficiente de encontrar os k-vizinhos
- Compressão

Obrigado !