LEI / LCC
Departamento de Informática
Universidade do Minho

Visualização III

Introdução aos Níveis de Detalhe (LOD)

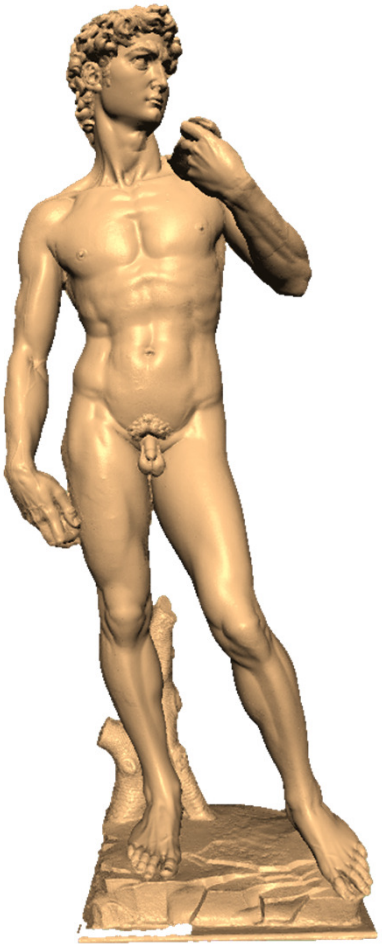
As imagens presentes neste documento provêm essencialmente do livro "Level of Detail for 3D Graphics" e de apresentações realizadas por Michael Garland

Níveis de Detalhe (LOD)

- **Introdução**
- Classificação de Algoritmos
- Operadores Locais de Simplificação
- Algoritmos
- Selecção de LODs
- LOD para terrenos

Problema

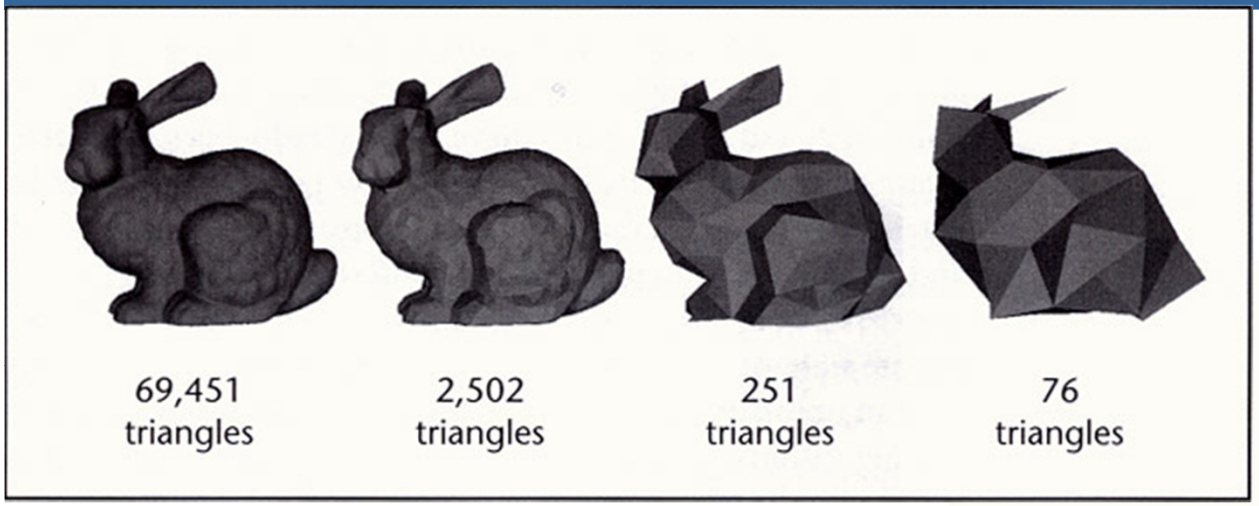
- Bases de dados com modelos de grande detalhe de origens variadas:
  - Laser scanner
  - Software CAD
  - Satélite
- Modelos cuja complexidade excede largamente a capacidade dos sistemas gráficos

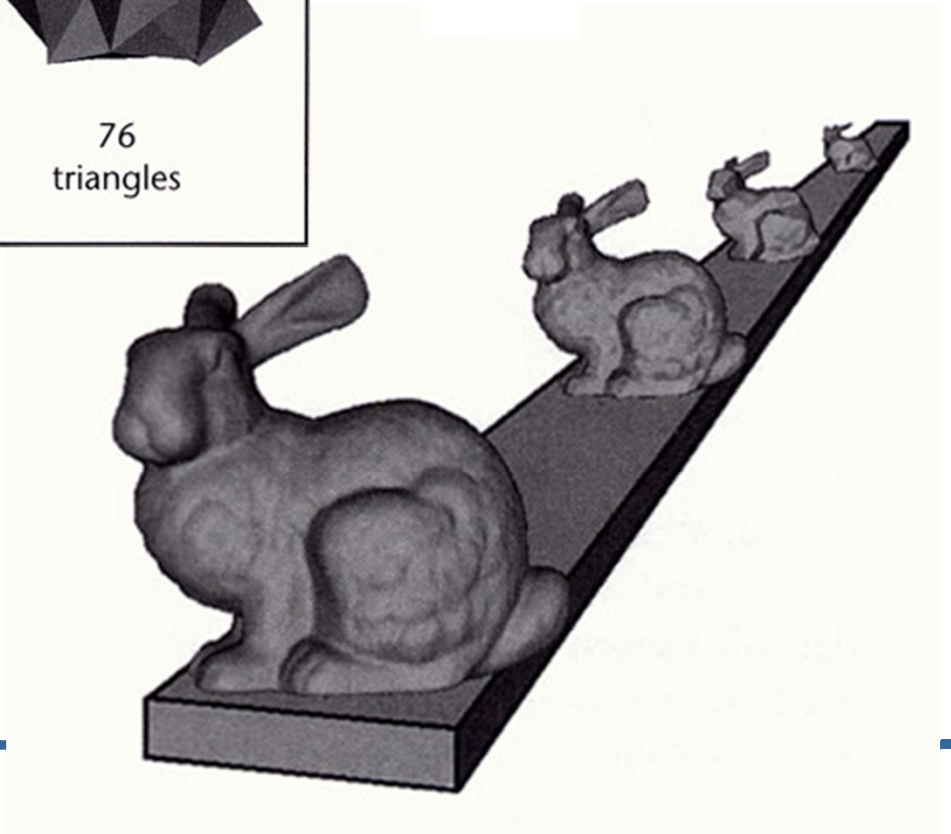


Solução

- Adaptar o nível de detalhe à capacidade do hardware.
- Processo tem de ser automático.
- É impossível, ou pelo menos impraticável, faze-lo à "mão".

O conceito


69,451
triangles
2,502
triangles
251
triangles
76
triangles



Outras utilizações

- Gerar versões simplificadas de modelos para:
  - View Frustum Culling
  - Detecção de colisões
  - Geração de sombras
  - ...

Níveis de Detalhe (LOD)

- Introdução
- **Classificação de Algoritmos**
- Operadores Locais de Simplificação
- Algoritmos
- Selecção de LODs
- LOD para terrenos

Classificação: Resolução

- Níveis de detalhe:
  - discretos
    - É criado um conjunto de simplificações do objecto
    - Em tempo real é seleccionada uma das representações
  - contínuos ou multiresolução
    - É criada uma estrutura que armazena uma sequência contínua de simplificações.
    - O nível de detalhe é "extraído" desta estrutura em tempo real.
- Quais as vantagens de cada?
  - fidelidade?
  - versatilidade?
  - streaming?

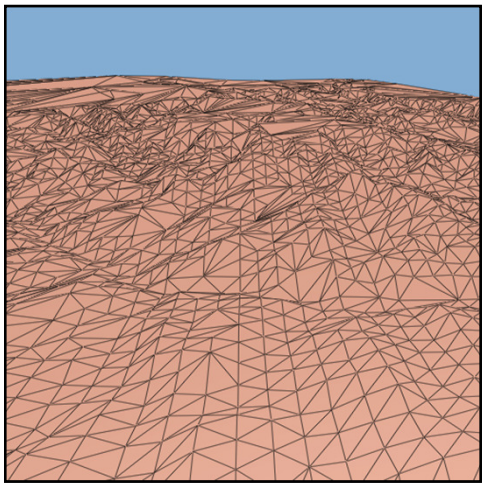
(demo bunnylod)

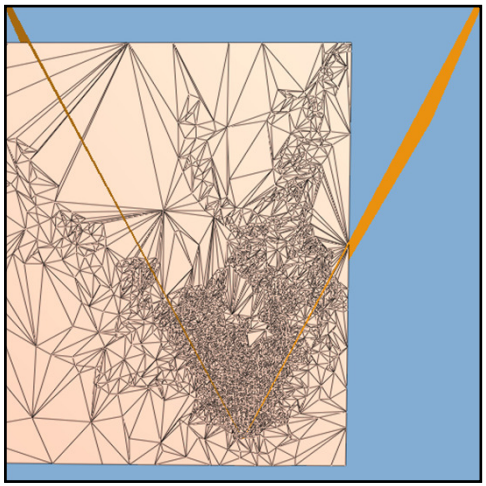
Classificação: Câmara

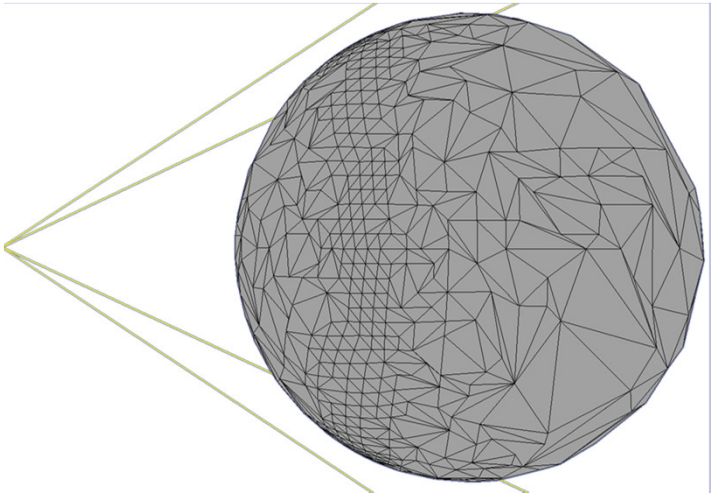
- Níveis de detalhe:
  - View independent
    - A simplificação é independente da posição da câmara.
  - View dependent
    - Permite que porções diferentes do objecto sejam exibidas a diferentes resoluções
- Quais as vantagens de cada?
  - fidelidade (considerando um número fixo de polígonos)?
  - versatilidade?

Classificação: Câmara

- View Dependent
  - Mais polígonos onde é necessário, considerando a posição da câmara
  - Computacionalmente mais dispendioso



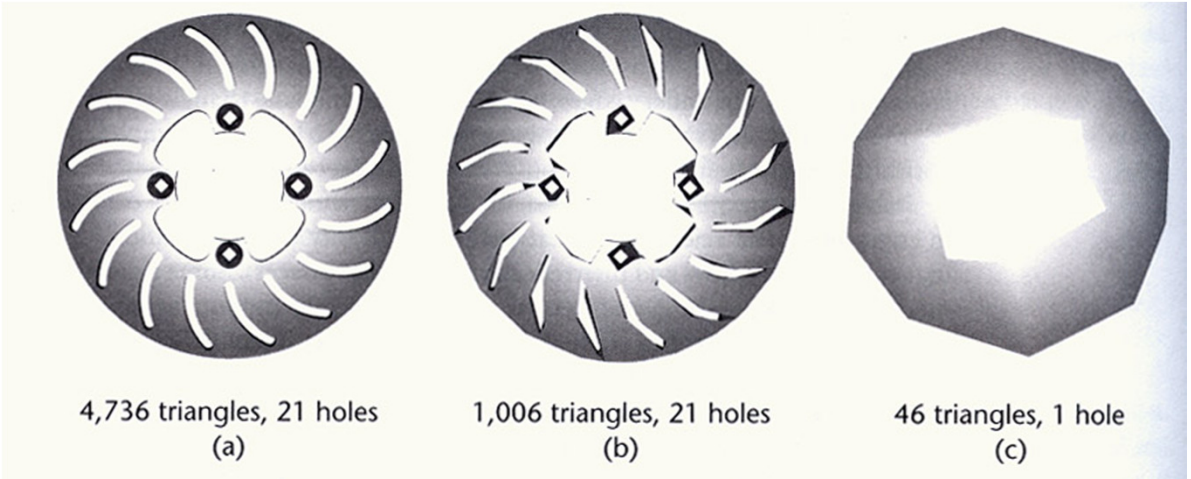




(vídeo svlod_gcanyon)

Classificação: Topologia

- Preservar ou não a topologia?
  - fidelidade vs. simplificação.
  - sistemas híbridos?



Classificação: Objectivo

- Fidelidade
  - cria uma simplificação cujo erro não excede um determinado valor
  - erro = distância entre a simplificação e o original
- Número de polígonos
  - cria uma simplificação com o número desejado de polígonos tentando ao mesmo tempo minimizar o erro

Classificação: Métrica

- Erro geométrico
  - definido em termos da "distância" entre o modelo original e a simplificação (erro total),
  - ou entre duas simplificações consecutivas (erro incremental)
  - Distância de Hausdorff
    - *distância máxima de um ponto no polígono A ao ponto mais próximo do polígono B*

$$H(A,B) = \max(h(A,B), h(B,A))$$

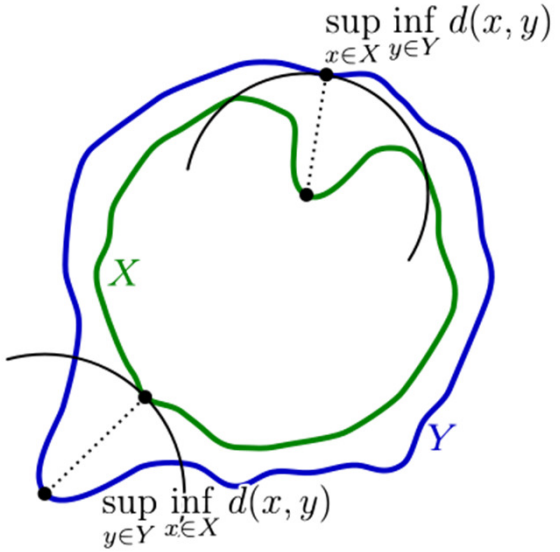
$$h(A,B) = \max_{a \in A} \min_{b \in B} \|a - b\|$$

Classificação: Métrica

- Erro geométrico
  - Distância de Hausdorff
    - *distância máxima de um ponto um objecto ao ponto mais próximo do outro objecto*

$$H(A,B) = \max(h(A,B), h(B,A))$$

$$h(A,B) = \max_{a \in A} \min_{b \in B} \|a - b\|$$



Classificação: Métrica

- Erro geométrico:
  - Máximo vs. Média
  - Garantia de um limite superior para o erro, ou...
  - ...ignorar alguns pontos especialmente maus?

Classificação: Métrica

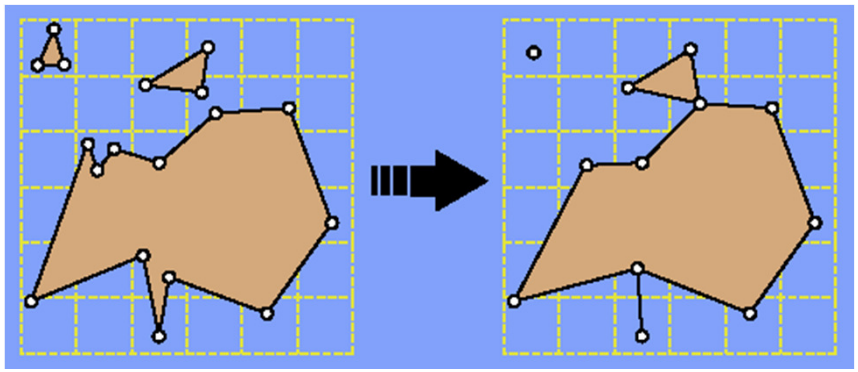
- Erro da imagem produzida
  - quantifica simplificações pelo erro em pixels introduzido

Níveis de Detalhe (LOD)

- Introdução
- Classificação de Algoritmos
- **Operadores Locais de Simplificação**
- Algoritmos
- Selecção de LODs
- LOD para terrenos

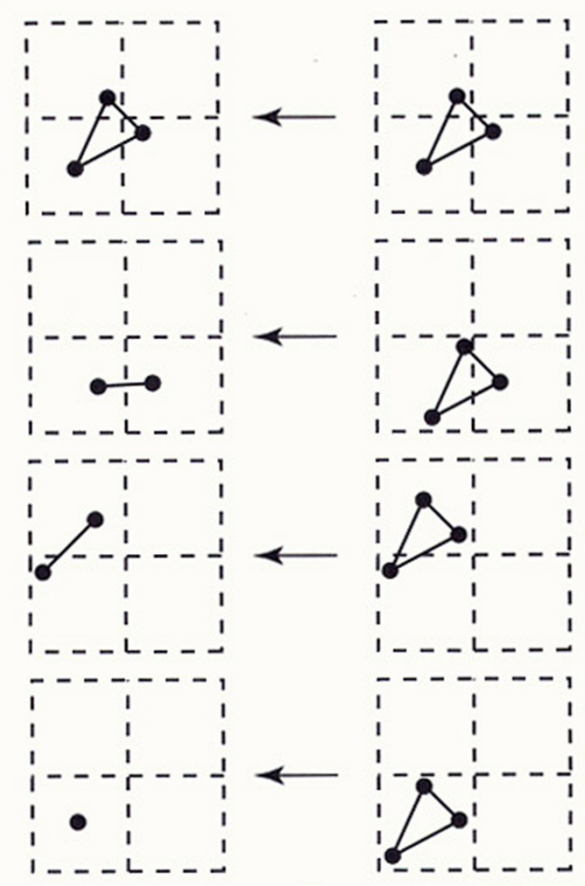
Vertex Clustering

- Partição espacial em células
- Fusão dos vértices em cada célula
- Rossignac e Burrel [grelha]



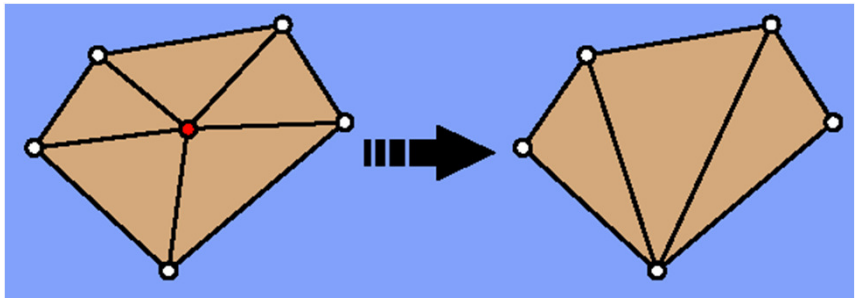
Vertex Clustering

- Características:
  - Muito rápido
  - Baixa fidelidade
  - Inconsistências



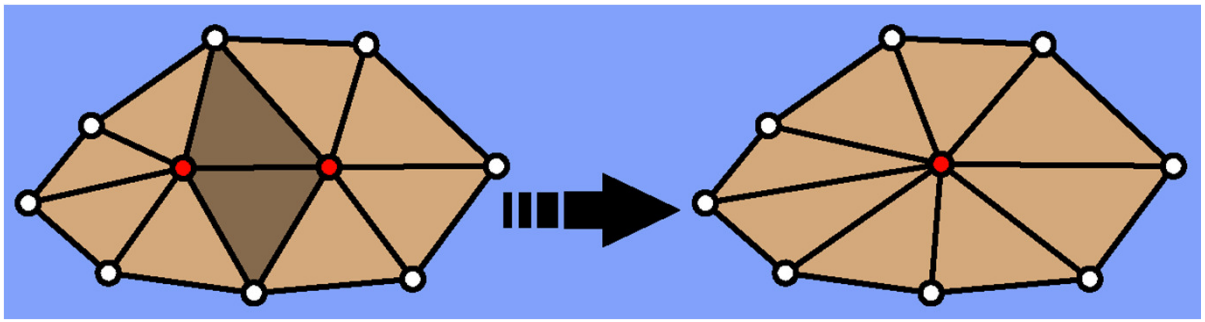
Vertex Removal

- Ordenar vértices de acordo com a sua relevância
- Eliminar o vértice menos relevante
- Retriangular a cena
- Preserva Topologia
- Schroeder et. al



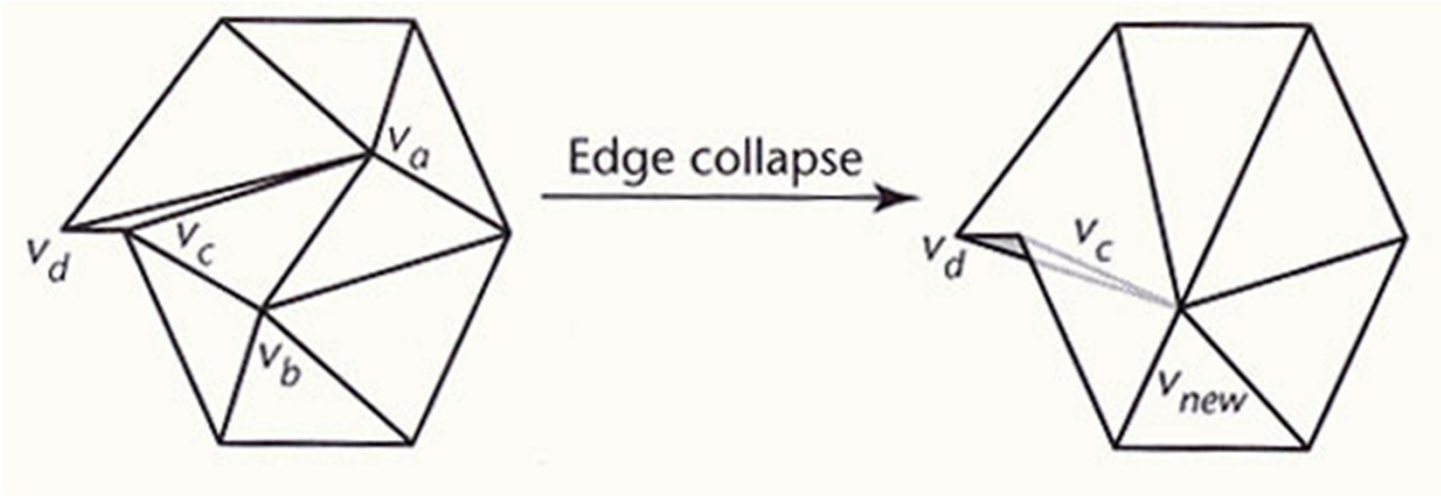
Edge Collapse

- Semelhante ao anterior, opera em arestas, em vez de vértices
- Questão: onde posicionar o novo vértice?
- Hoppe



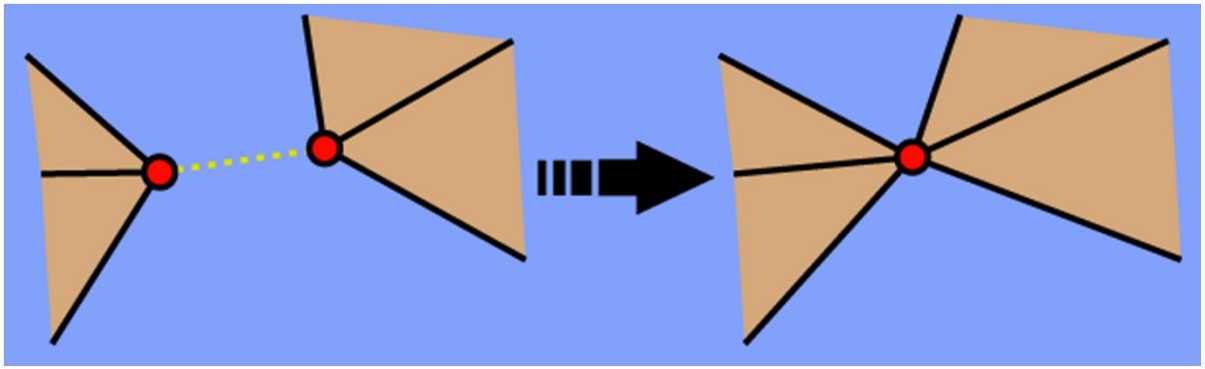
Edge Collapse

- É necessário ter algum cuidado a escolher os vértices para evitar alterar a orientação dos triângulos.



Vertex-pair Collapse

- Permite a ligação de vértices que não partilhem uma aresta.
- Altera topologia
- Número de hipóteses é limitado definindo uma distância máxima para junção de vértices não partilhados
- Schroeder, Garland



Outros Atributos

- Até agora a simplificação foi baseada somente em geometria.
- O que fazer com os materiais (cores, texturas, shaders)?
- Hipótese simples:
  - dividir o modelo em conjuntos de triângulos com o mesmo material.
  - Efectuar a simplificação em cada um desses conjuntos.

Níveis de Detalhe (LOD)

- Introdução
- Classificação de Algoritmos
- Operadores Locais de Simplificação
- **Algoritmos**
- Selecção de LODs
- LOD para terrenos

LOD na Prática

- Garland and Heckbert
  - Permite alterações à topologia
  - Operador local: Vertex Pair Collapse
  - Erro geométrico
  - Suporta materiais (cores e texturas).

Garland and Heckbert

- Selecção de pares de vértices candidatos para fusão.

- Um par de vértices (v_1,v_2) é candidato se:

  - v_1,v_2 partilham uma aresta, ou,

  - $|| v_1 - v_2 || < t$

Garland and Heckbert

- A cada vértice associa o conjunto de planos definido pelos triângulos ao qual o vértice pertence.
- Erro para o vértice = distância aos planos
- Vértice original => erro = 0

Garland and Heckbert

- Erro para o vértice

$$\Delta(v) = \sum_{p \in planos(v)} (p^T v)^2$$

$$= \sum_{p \in planos(v)} (v^T p)(p v^T)$$

$$= v^T \left(\sum_{p \in planos(v)} K_p \right) v$$

$$K_p = pp^T = \begin{bmatrix} a^2 & ab & ac & ad \\ ab & b^2 & bc & bd \\ ac & bc & c^2 & cd \\ ad & bd & cd & d^2 \end{bmatrix}$$

$$Q_v = \sum_{p \in planos(v)} K_p$$

Garland and Heckbert

- Erro para aresta
  - idealmente (planos(v) U planos(u))
  - Solução proposta: $Qv + Qu$
  - Se o vértice j for substituir a aresta (v,u) então o erro introduzido é
    - $e = j\ (Qv + Qu)\ j^T$
  - A solução proposta desvia-se da ideal pois pode repetir planos.

Garland and Heckbert

- Onde colocar o vértice j resultante da fusão dos vértices v e u?
- Soluções mais simples:

$j = v$ ou $j = u$ ou

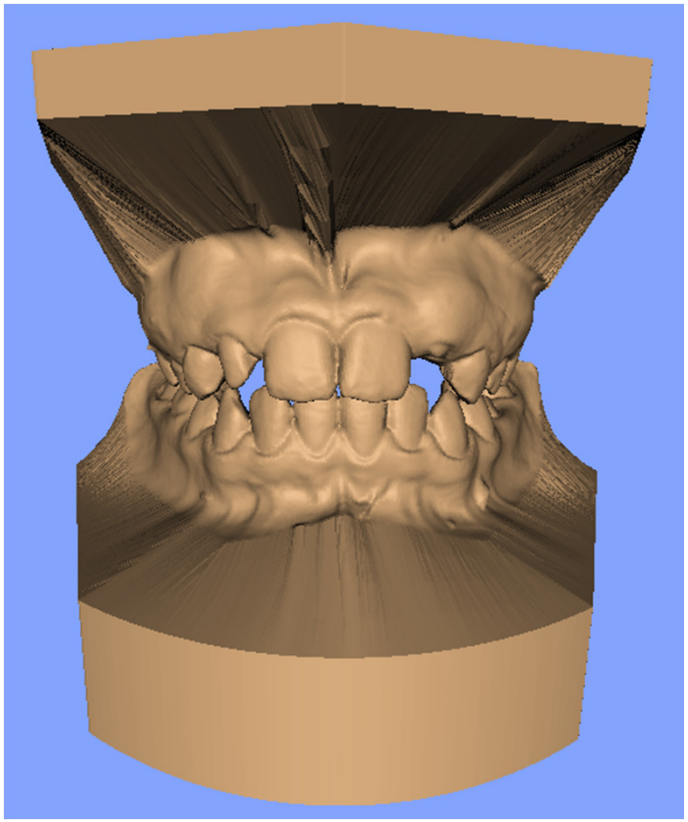
$j = (v+u)/2$

- Alternativa - minimizar o erro:
  - função quadrática tem um mínimo onde as derivadas parciais são zero.

Garland and Heckbert

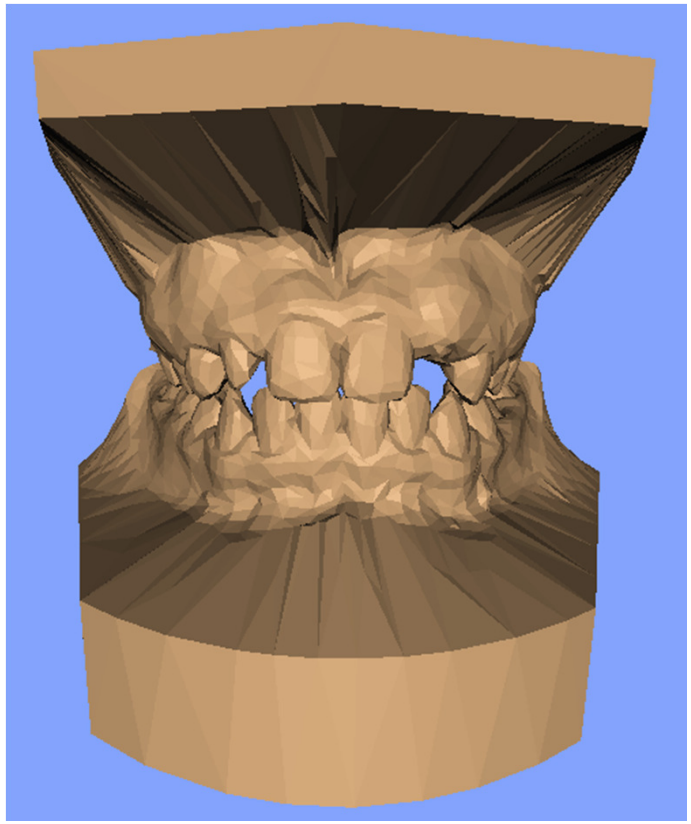
- Resumo do algoritmo
  - 1. Calcular as matrizes Q para todos os vértices
  - 2. Seleccionar todos os pares válidos
  - 3. Calcular o ponto de fusão ideal para cada par e o erro respectivo
  - 4. Colocar todos os pares numa lista ordenados por erro
  - 5. Remover o par (v1,v2) do princípio da lista, e realizar a fusão
  - 6. Actualizar os pares que envolvam v1 e v2 (lista também é actualizada)
  - 7. Criar pares para o vértice resultante da fusão (lista também é actualizada)
  - 8. Go to 5

Garland and Heckbert



424,376 faces

55 sec



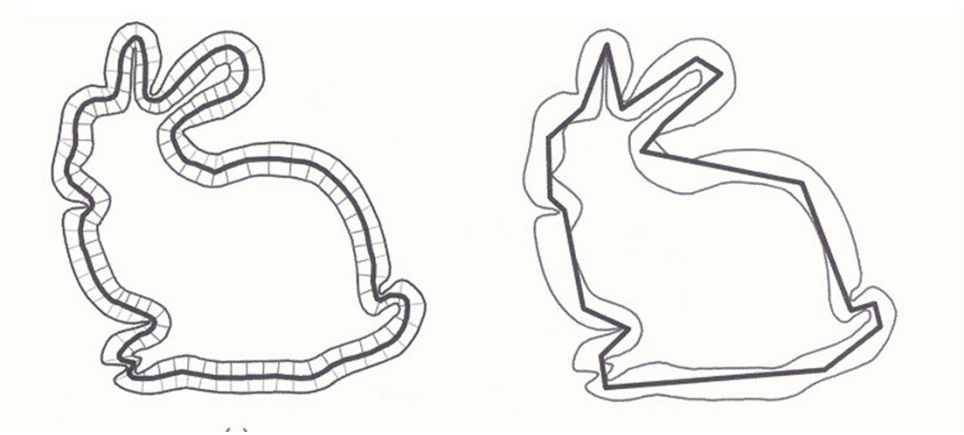
8000 faces

Garland and Heckbert

- Cor, Texturas e Normais
  - Essencialmente o mesmo processo.
  - Distancia a um plano bidimensional em R^n.
  - em vez de trabalhar com matrizes 3x3, utilizar matrizes 5x5 (texturas) ou 6x6 (cores ou normais).
- Shaders?

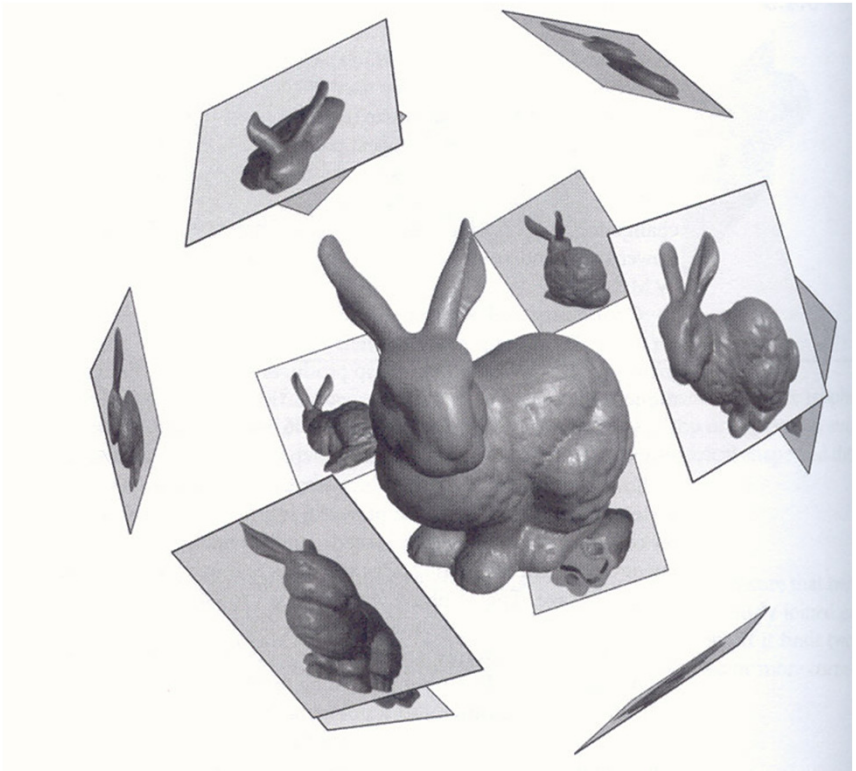
Outras Estratégias Interessantes

- Cohen - Simplification Envelopes
- Remover iterativamente os vértices e retriangular os buracos.
- Se ao remover um vértice, a nova triangulação furar o envelope, então rejeitar a remoção do vértice.



Outras Estratégias Interessantes

- Linstrom and Turk
- Analisar o impacto da remoção de um vértice através de análise baseada nas imagens antes e depois da remoção.
- n(20) pontos de vista diferentes



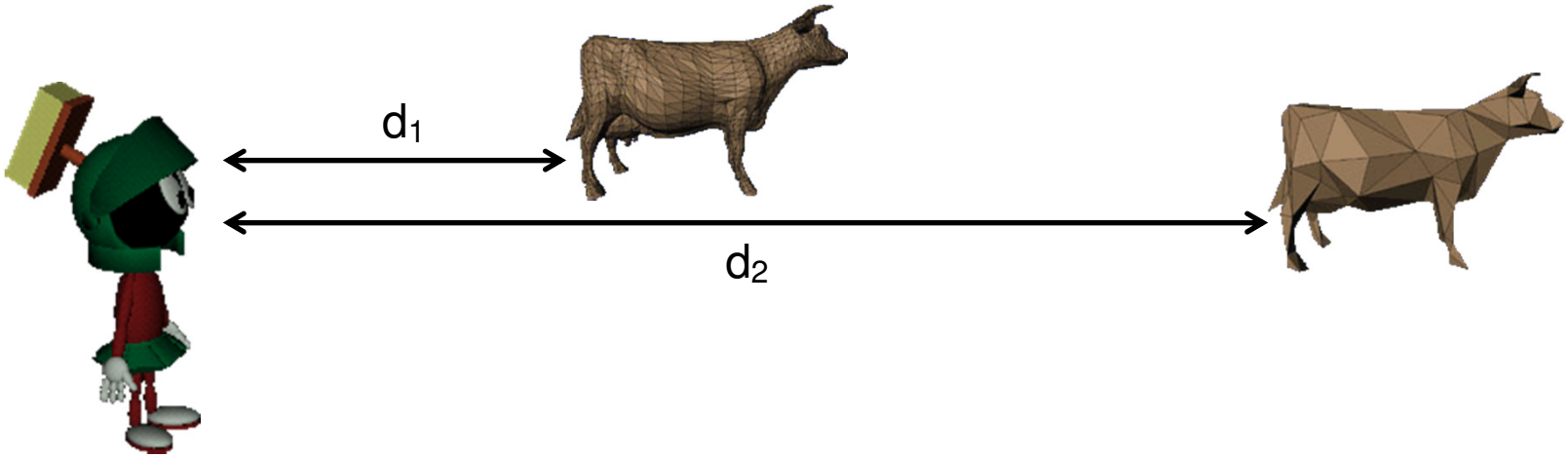
Níveis de Detalhe (LOD)

- Introdução
- Classificação de Algoritmos
- Operadores Locais de Simplificação
- Algoritmos
- **Selecção de LODs**
- LOD para terrenos

Selecção de LODs

- Distância geométrica
  - Objectos mais próximos são desenhados com mais detalhe.
  - À medida que a câmara se afasta o modelo vai "perdendo" resolução e portanto pode-se utilizar uma versão mais simples do modelo.
  - Mas, a noção de distância é fortemente influenciada pelos parâmetros da câmara e dimensão da janela

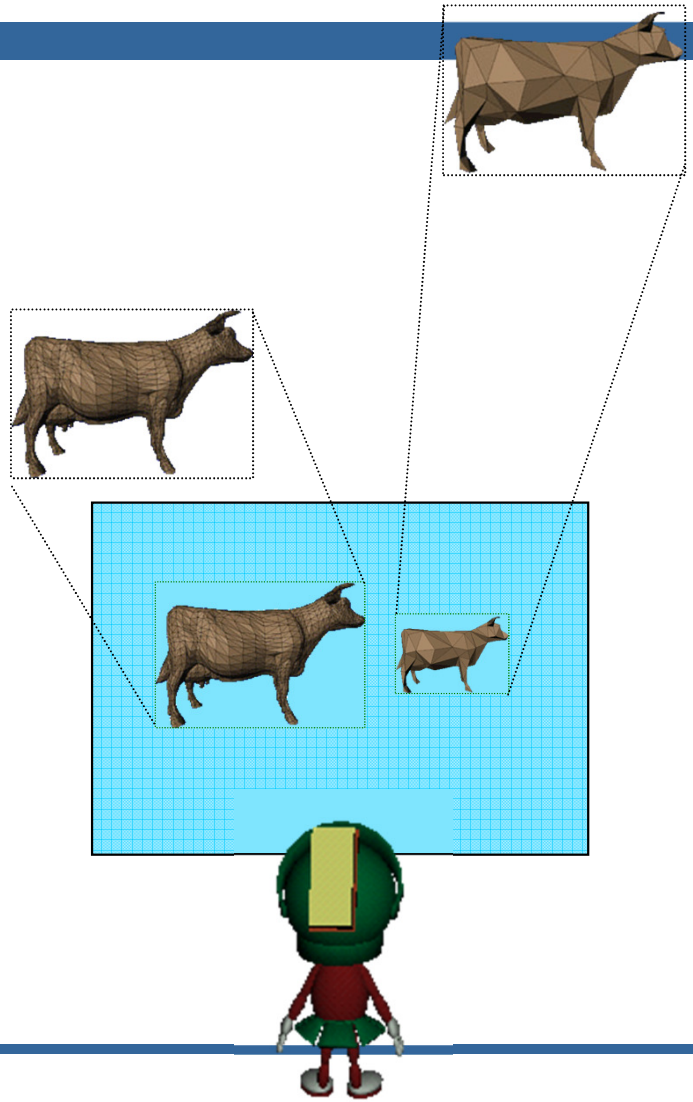
Selecção de LODs



Selecção de LODs

- Dimensão da Projecção
  - Alternativa à distância geométrica
  - Mais fiel
  - Menor desempenho
  - Possibilidade de utilizar um *bounding volume*

Selecção de LODs



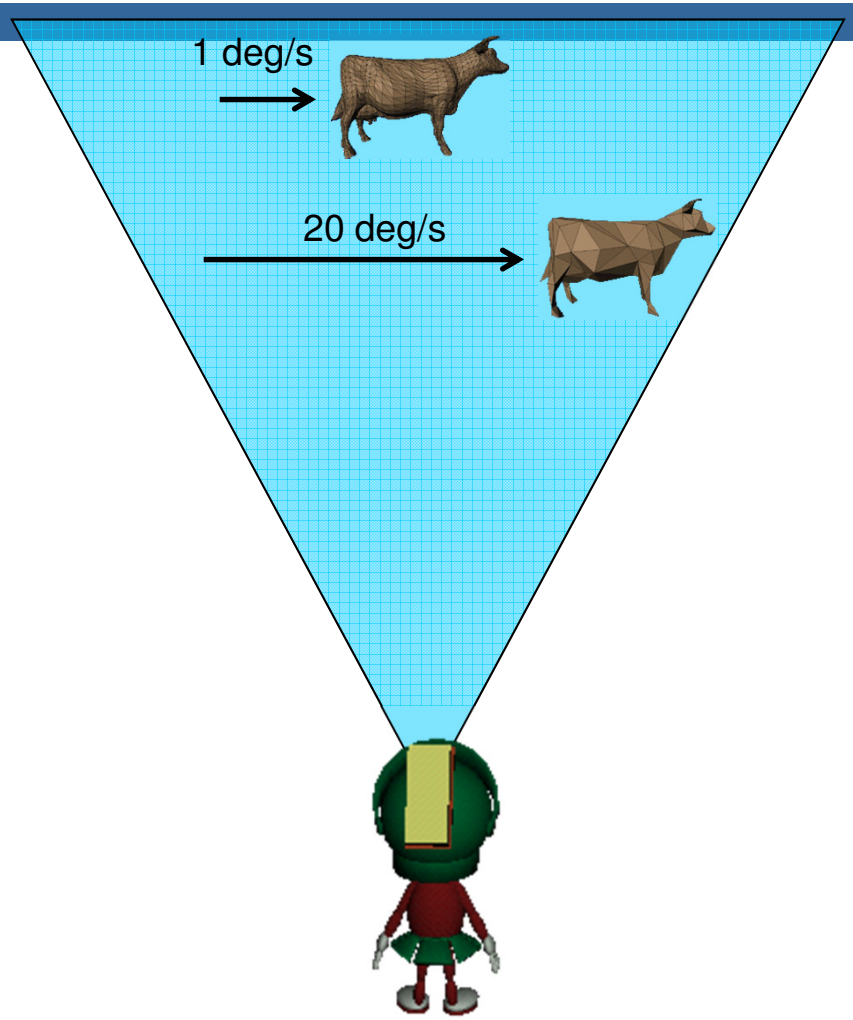
Selecção de LODs

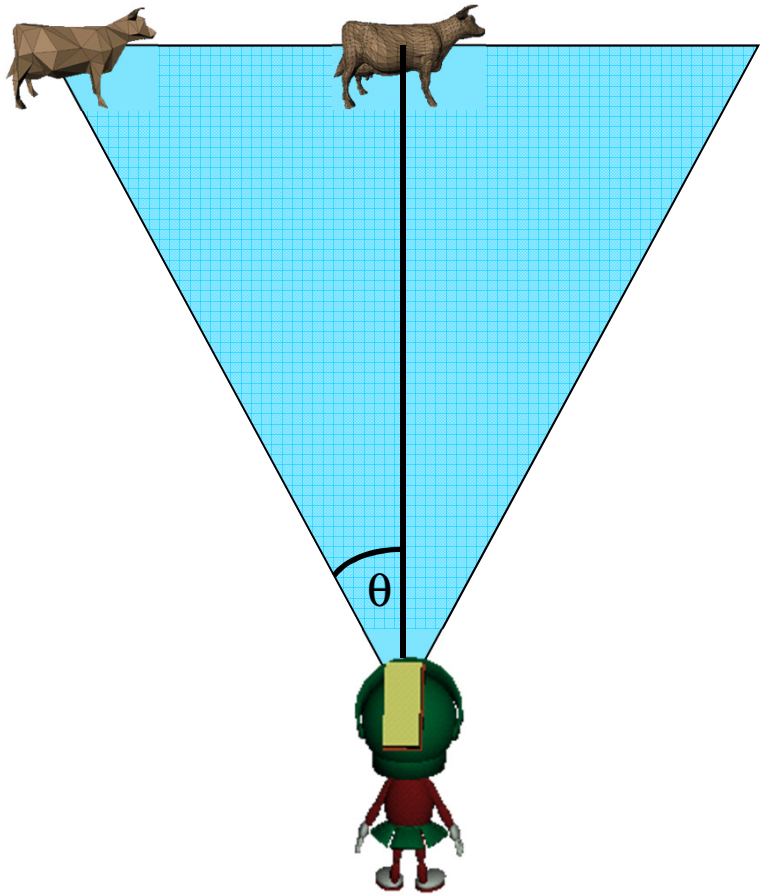
- Prioridade
  - Classificação dos objectos relativamente à importância na cena
  - Nível de detalhe proporcional à importância

Selecção de LODs

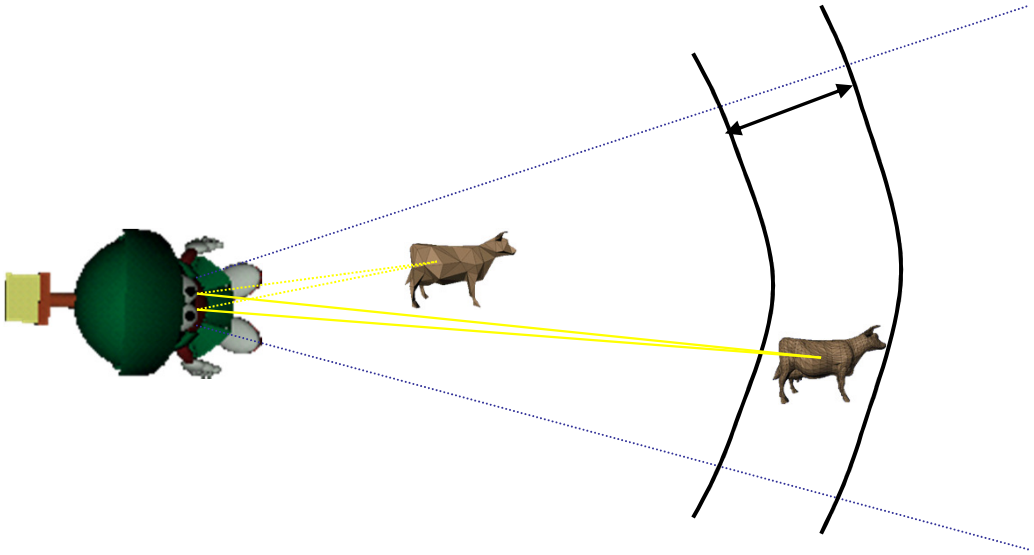
- Percepção
  - Objectos em movimento precisam de uma resolução menor
  - Objectos na periferia precisam de menor resolução (eye tracking)
  - Objectos dentro da zona focal(2 eye tracking)

Selecção de LODs



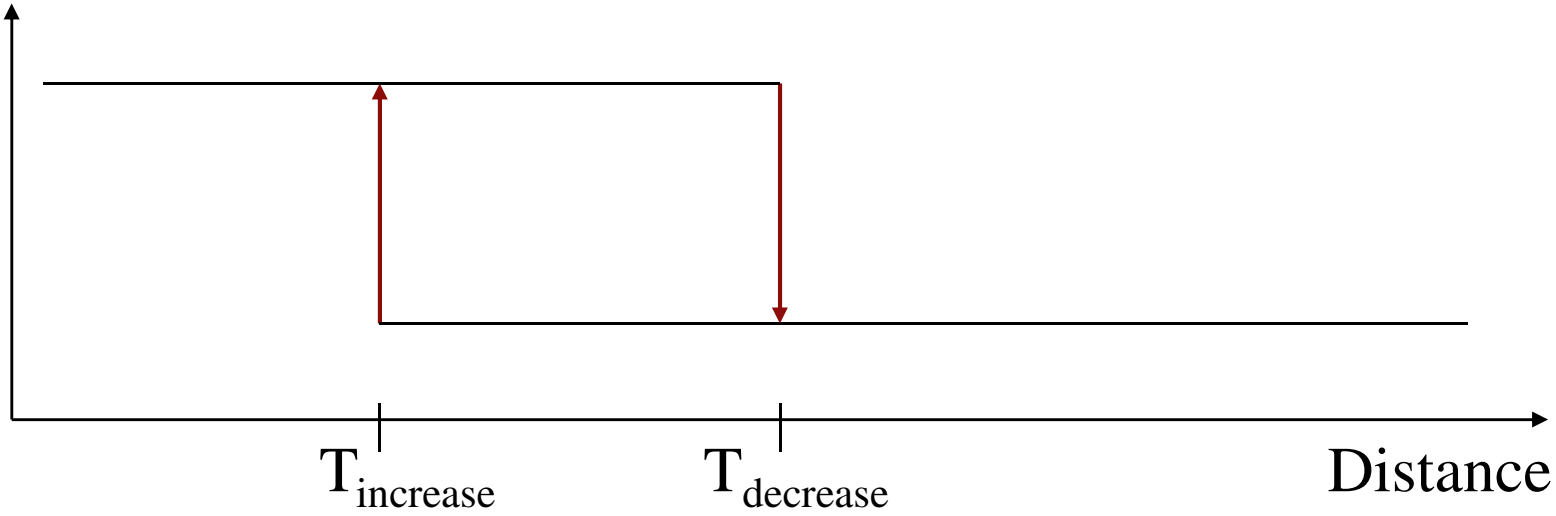


Selecção de LODs



Selecção de LODs

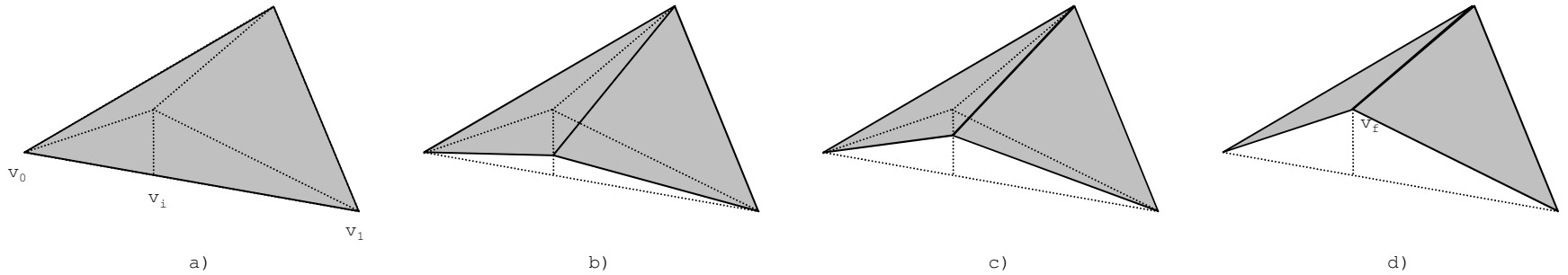
- *Hysteresis*



Selecção de LODs

- Geomorphing
  - Objectivo: diluir a transição entre dois modelos ao longo do tempo

$$v_p(t) = (1-t)*v_i + t*v_f$$



(vídeo tvcg2002)

Níveis de Detalhe (LOD)

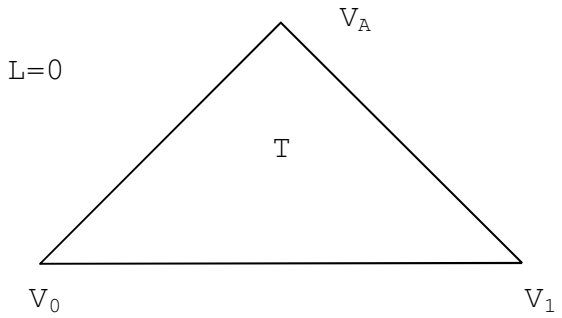
- Introdução
- Classificação de Algoritmos
- Operadores Locais de Simplificação
- Algoritmos
- Selecção de LODs
- **LOD para terrenos**

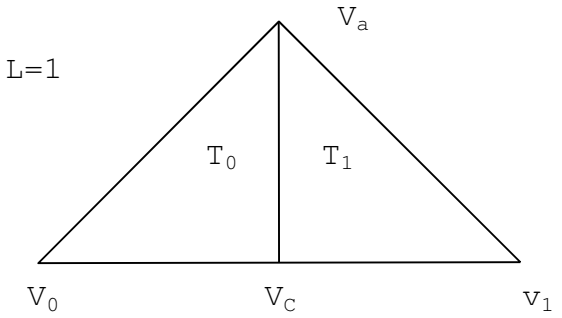
LOD para Terrenos

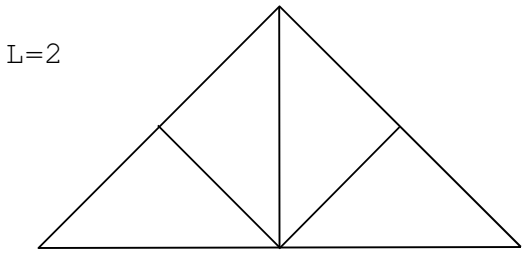
- Os terrenos devido à sua extensão, exigem um nível de detalhe dinâmico e view-dependent.
  - Pouco detalhe ao longe e ...
  - ... muito detalhe perto da câmara

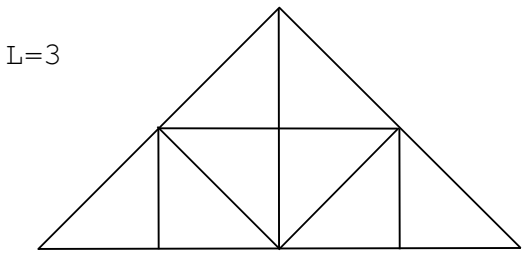
LOD para Terrenos

- ROAM (Duchaineau et. al.): Algoritmo baseado em árvores binárias de triângulos



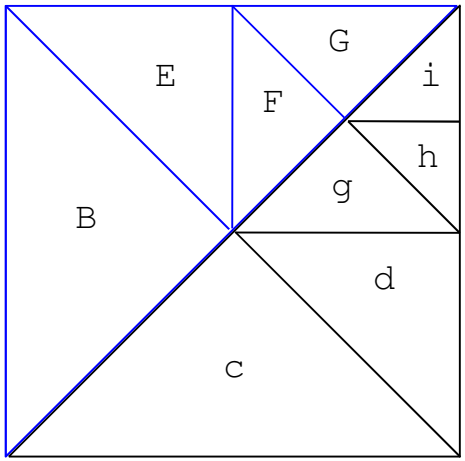




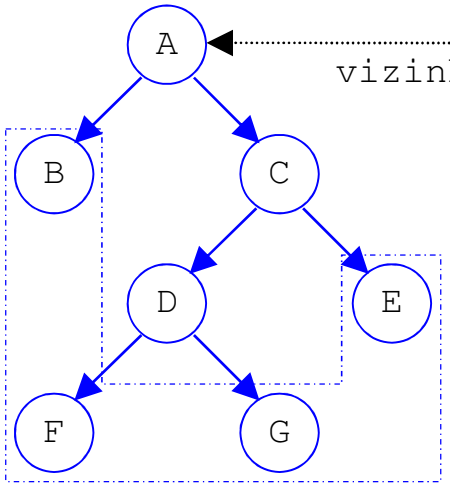


LOD para Terrenos

- Árvores Binárias de Triângulos

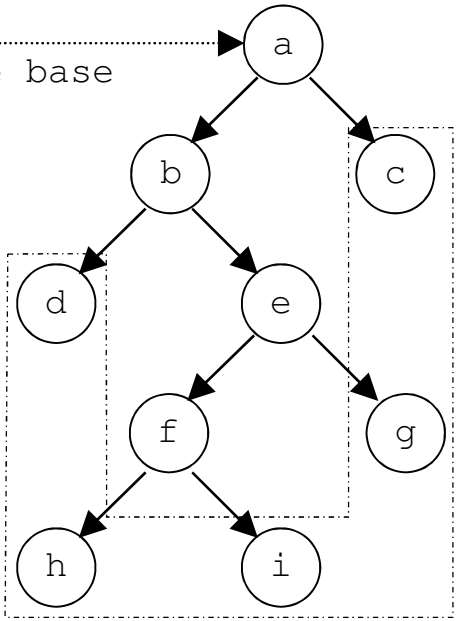


triangulação



BTT esquerda

vizinho de base



BTT direita

LOD para Terrenos

- As Operações

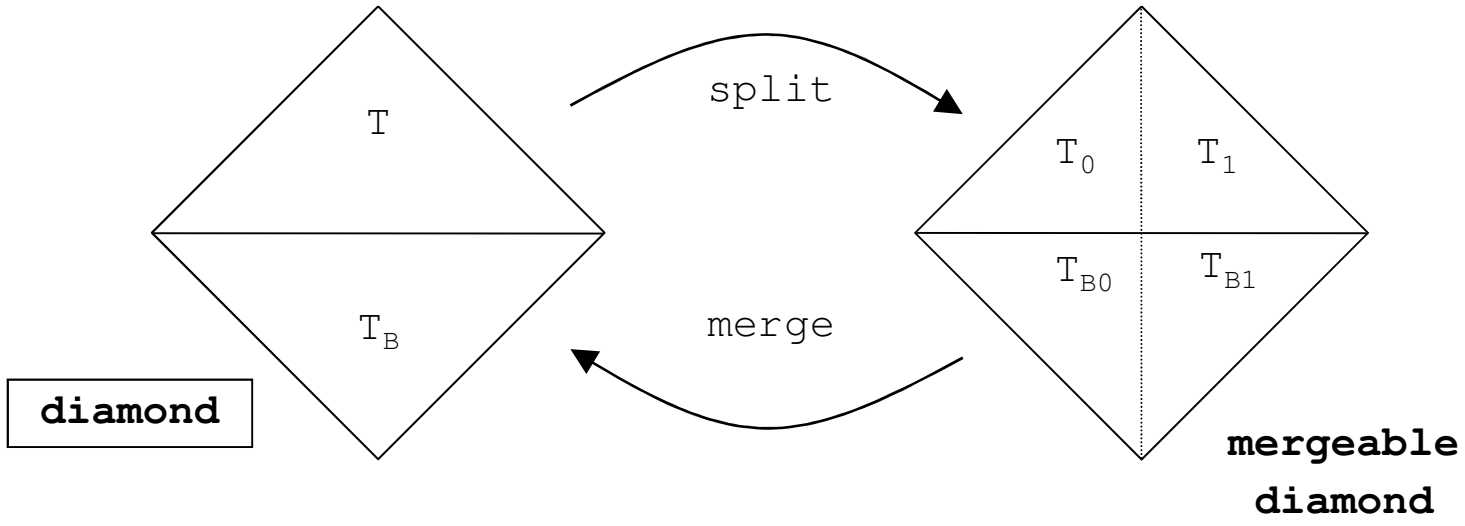
  Qualquer triangulação pode ser obtido a partir de qualquer outra à custa de uma sequência de duas operações:

  - ***SPLIT***: refina a triangulação, tornando-a mais definida

  - ***MERGE***: generaliza-a, tornando menos definida.

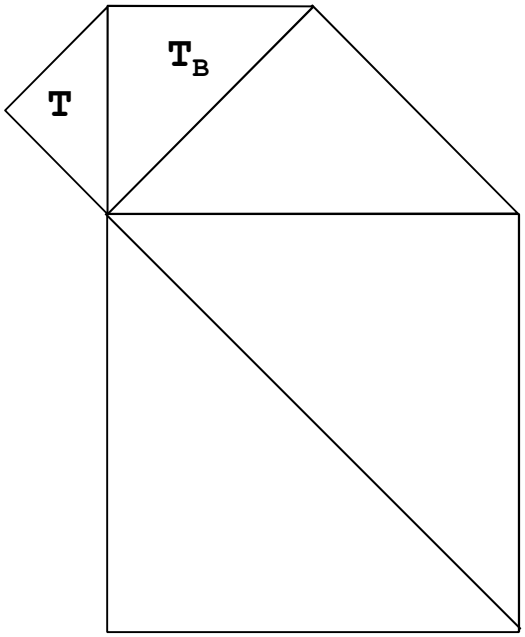
LOD para Terrenos

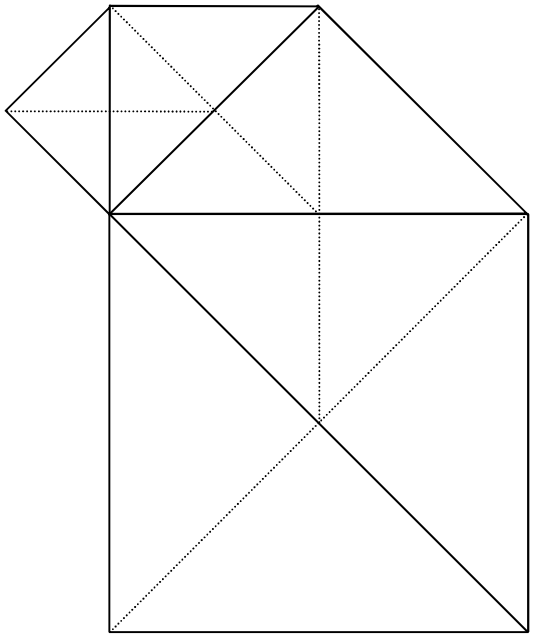
- As Operações



LOD para Terrenos

- Operações: *Forced-split*





LOD para Terrenos

- Algoritmo simples:
  - Parte da representação mais simples para o terreno (dois triângulos);
  - Recursivamente determina qual o triângulo com maior erro e realiza um split;
  - Caso um split gere uma descontinuidade espacial, então realizar o forced-split.

LOD para Terrenos

- Mas:
  - a diferença entre duas frames é geralmente pequena, ou seja existe uma coerência entre frames
  - O algoritmo anterior não tira partido dessa coerência

LOD para Terrenos

- Solução:
  - Partir da triangulação obtida na frame anterior;
  - Realizar split e merge até atingir um determinado objectivo.

LOD para Terrenos

- Alguns objectivos possíveis:
  - Minimizar um erro;
  - Budget de triângulos;
  - Tempo máximo por frame.

LOD para Terrenos

- Filas de Prioridades:
  - Criar duas filas de "prioridades" para potenciais splits e merges (prioridade = erro).
  - Fila de splits ordenada de forma decrescente.
  - Fila de merges, contendo todos os *diamonds*, ordenada de forma crescente. A prioridade de um merge é o máximo das prioridades dos triângulos envolvidos.
  - As prioridades devem ser monótonas, i.e. as prioridades dos filhos não devem ser superior à prioridade do pai.

LOD para Terrenos

- Algoritmo:
  - Enquanto (o erro/número de triângulos desejado não foi atingido) ou (max fila de splits > min fila de merge)
    - Se o terreno é demasiado grande/detalhado
      - realizar um merge
    - Caso contrário
      - realizar um split
    - Seguidamente é necessário actualizar as filas.

LOD para Terrenos

- A file de merge diz-nos quanto aumentamos o erro ao realizar esta operação
- A fila de split diz-nos quanto reduzimos o erro
- Ao realizar um merge estamos a diminuir o número de triângulos (-2).
- Sendo assim ficamos com espaço livre para realizar splits:
  - O mínimo da fila de merge representa o aumento do erro
  - O máximo da fila de splits representa o decréscimo do erro
  - Logo se

    (mínimo da fila merge) < (máximo da fila de splits)

  - pode-se realizar o merge seguido do split e diminuir o erro mantendo o número de triângulos (excepto se for um forced split).

Referências

- ***Level of Detail for 3D Graphics***, David Luebke, Martin Reddy, Jonathan D. Cohen, Amitabh Varshney, Benjamin Watson, Robert Huebner
- ***Hausdorff distance between convex polygons***, Normand Grégoire and Mikael Bouillot (http://cgm.cs.mcgill.ca/~godfried/teaching/cg-projects/98/normand/main.html)
- ***Multiresolution 3d approximations for rendering complex scenes***. E Burrel J. Rossignac. Modelling in Computer Graphics, E. B. Falcidieno, T. L. Kunn, eds, Springer Verlag, pp 455-465, 1993.
- ***Decimation of triangle meshes***. Schroeder, W. J., Zarge, J. A., and Lorensen, W. E. (1992). In *SIGGRAPH '92*
- ***Progressive meshes***. Hoppe, H. (1996). In *Computer Graphics (SIGGRAPH'96 Proceedings)*.
- ***Surface Simplification Using Quadric Error Metrics***. Garland, M. and P. Heckbert (1997). SIGGRAPH 97.
- ***Image-Driven Simplification***, Peter Lindstrom, Greg Turk, ACM Transactions on Graphics, 19(3), July 2000
- ***Simplification Envelopes***, Jonathan **Cohen**, Amitabh Varshney, Dinesh Manocha, Greg Turk, Hans Weber, Pankaj Agarwal, Frederick Brooks, William Wright, Journal of Computer Graphics
- ***Advanced Issues in Level of Detail***, Martin Reddy
- ***ROAMing Terrain: Real-time Optimally Adapting Meshes*** by Mark Duchaineau, Murray Wolinsky, David E. Sigeti, Mark C. Miller, Charles Aldrich, Mark B. Mineev-Weinstein