INF2604 - Geometria Computacional

Trabalho Final: Conversão de Triângulos em Quadriláteros

**Prof**. Waldemar Celes

Departamento de Informática, PUC-Rio

**Aluno:** Luis Fernando Marin Sepulveda

O trabalho final consiste na implementação de 3 algoritmos vistos em aula, para transformar um conjunto de triângulos representados como polígonos em quadriláteros.

Para todos os experimentos, dois tipos de triangulação inicial foram usados, a primeira consistindo em uma triangulação incremental e a segunda, uma triangulação de Delaunay. Todos os metodos desenvolvidos são testados utilizando os mesmos arquivos de pontos para permitir uma comparação de desempenho.

1. **Métodos implementados:**
   1. **Catmull-Clark-based quadrilaterals**

O primeiro método desenvolvido é baseado na aplicação da subdivisão Catmull-Clark[1], em uma malha inicial de triângulos. A aplicação do algoritmo pode ser apresentada em três etapas: Primeiro, determine o ponto central de cada aresta do triângulo. Segundo, calcular o ponto central do triângulo. Finalmente, junte os novos pontos criados através de arestas, o resultado é uma malha de quadrados[2]. Este método pode ser utilizado com diferentes divisões iniciais, a Figura 1 mostra um exemplo do resultado de sua aplicação.



Figura 1. Exemplo da subdivisão Catmull-Clark [2].

* 1. **Agrupamento de triângulos.**

O segundo método implementado não cria novos pontos ou arestas, mas é baseado na eliminação de arestas comuns de triângulos adjacentes. A aplicação do método pode ser apresentada em três etapas: Primeiro, o comprimento de todas as arestas comuns entre os triângulos é medido. Em segundo lugar, ordene em ordem decrescente todas as distâncias das arestas. Terceiro, enquanto a lista criada na etapa anterior não está vazia, avalie se ao eliminar a aresta mais longa, o quadrilátero resultante é convexo; se verdadeiro, elimine a aresta e marque os triângulos aos quais ela pertence como agrupados, o algoritmo é descrito na Figura 2.

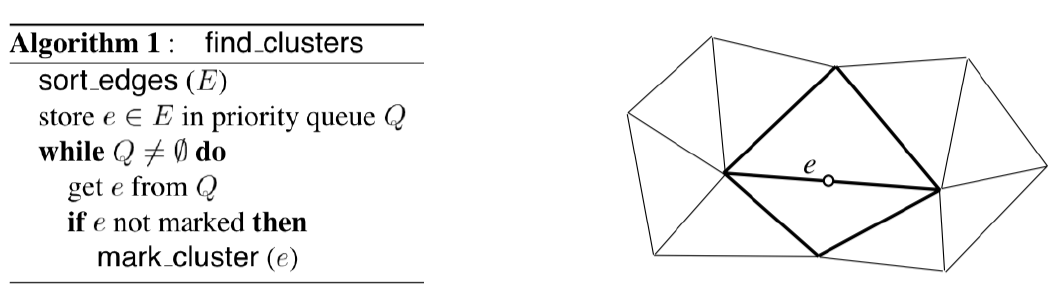


Figura 2. Algoritmo e exemplo do algoritmo de clustering [2].

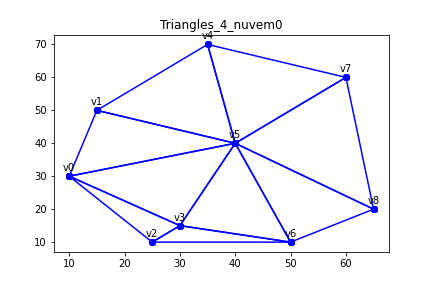
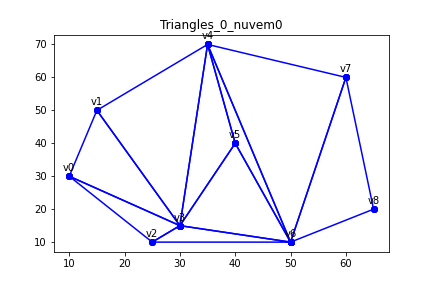
* 1. **Clustering and Catnull-Clark.**

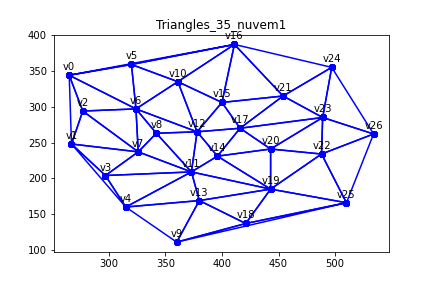
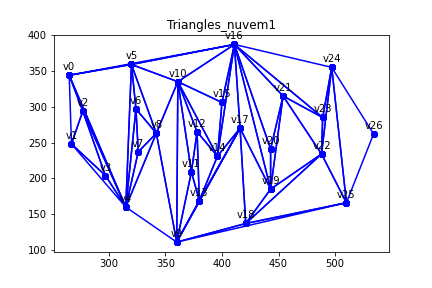
O terceiro método apresenta uma combinação dos dois metodos anteriores, apresentada por Velho Luiz em Quadrilateral Meshing Using 4-8 Clustering [2].

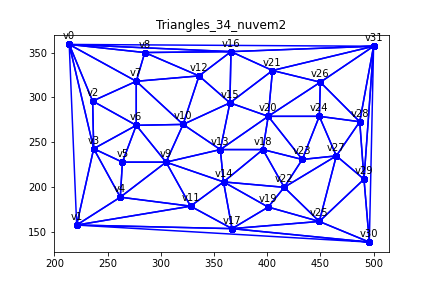
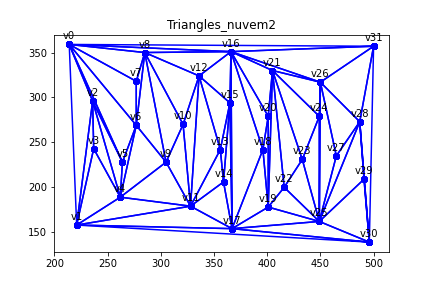
O primeiro passo é aplicar o agrupamento descrito no método 1.2, o resultado desse algoritmo é uma malha contendo quadriláteros e também contém triângulos que não puderam ser agrupados. A segunda etapa é aplicar o Método 1.1 à malha anterior, o resultado é uma malha de quadriláteros.

1. **Dados do estudo**

Para mostrar claramente os efeitos de cada método, arquivos diferentes com diferentes números de pontos são usados: nuvem0 com 9, nuvem1 com 27, nuvem2 com 32 e nuvem40 com 40. Conforme mencionado anteriormente, dois tipos diferentes de triangulações são criados para este conjunto de pontos Figura 3.







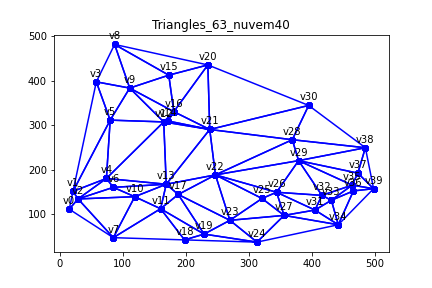
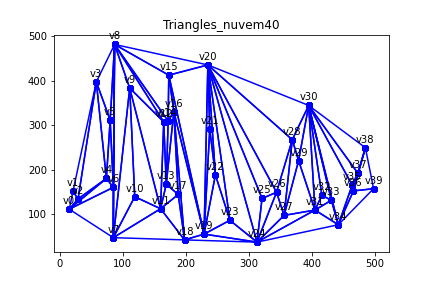


Figura 3. Triangulação inicial de nuvem (Esquerda: incremental; Direita: Delaunay).

Além desses quatro arquivos, são utilizados mais cinco com um número de pontos que varia de 50 a 90, para permitir uma comparação do tempo computacional.

1. **Resultados**

A Figura 4 mostra os resultados de cada método:

|  |  |  |
| --- | --- | --- |
| **Catmull-Clark** | **Agrupamento** | **Clustering/ Catmull-Clark** |
| Triangulação inicial incremental | | |
| E:\Fernando\PUC 2020\Semestre 2\Geometria Computacional\Tareas\Trabajo Final\GEO2020_2TrabalhoFinalLuisMarin\Figures Final\Quadrilate_8_nuvem0.npy.png | E:\Fernando\PUC 2020\Semestre 2\Geometria Computacional\Tareas\Trabajo Final\GEO2020_2TrabalhoFinalLuisMarin\Figures Final\Quadrilate_5_Indirect1_nuvem0.npy.png | E:\Fernando\PUC 2020\Semestre 2\Geometria Computacional\Tareas\Trabajo Final\GEO2020_2TrabalhoFinalLuisMarin\Figures Final\QuadrilateInd2_5_nuvem0.npy.png |
| Triangulação inicial Delaunay | | |
| E:\Fernando\PUC 2020\Semestre 2\Geometria Computacional\Tareas\Trabajo Final\GEO2020_2TrabalhoFinalLuisMarin\Figures Final\Quadrilate_8_nuvem0_Delaunay.npy.png | E:\Fernando\PUC 2020\Semestre 2\Geometria Computacional\Tareas\Trabajo Final\GEO2020_2TrabalhoFinalLuisMarin\Figures Final\Quadrilate_5_Indirect1_nuvem0_Delaunay.npy.png | E:\Fernando\PUC 2020\Semestre 2\Geometria Computacional\Tareas\Trabajo Final\GEO2020_2TrabalhoFinalLuisMarin\Figures Final\QuadrilateInd2_5_nuvem0_Delaunay.npy.png |
| Triangulação inicial incremental | | |
| E:\Fernando\PUC 2020\Semestre 2\Geometria Computacional\Tareas\Trabajo Final\GEO2020_2TrabalhoFinalLuisMarin\Figures Final\Quadrilate_43_nuvem1.npy.png | E:\Fernando\PUC 2020\Semestre 2\Geometria Computacional\Tareas\Trabajo Final\GEO2020_2TrabalhoFinalLuisMarin\Figures Final\Quadrilate_29_Indirect1_nuvem1.npy.png | E:\Fernando\PUC 2020\Semestre 2\Geometria Computacional\Tareas\Trabajo Final\GEO2020_2TrabalhoFinalLuisMarin\Figures Final\QuadrilateInd2_29_nuvem1.npy.png |
| Triangulação inicial Delaunay | | |
| E:\Fernando\PUC 2020\Semestre 2\Geometria Computacional\Tareas\Trabajo Final\GEO2020_2TrabalhoFinalLuisMarin\Figures Final\Quadrilate_43_nuvem1_Delaunay.npy.png | E:\Fernando\PUC 2020\Semestre 2\Geometria Computacional\Tareas\Trabajo Final\GEO2020_2TrabalhoFinalLuisMarin\Figures Final\Quadrilate_26_Indirect1_nuvem1_Delaunay.npy.png | E:\Fernando\PUC 2020\Semestre 2\Geometria Computacional\Tareas\Trabajo Final\GEO2020_2TrabalhoFinalLuisMarin\Figures Final\QuadrilateInd2_26_nuvem1_Delaunay.npy.png |
| Triangulação inicial incremental | | |
| E:\Fernando\PUC 2020\Semestre 2\Geometria Computacional\Tareas\Trabajo Final\GEO2020_2TrabalhoFinalLuisMarin\Figures Final\Quadrilate_57_nuvem2.npy.png | E:\Fernando\PUC 2020\Semestre 2\Geometria Computacional\Tareas\Trabajo Final\GEO2020_2TrabalhoFinalLuisMarin\Figures Final\Quadrilate_37_Indirect1_nuvem2.npy.png | E:\Fernando\PUC 2020\Semestre 2\Geometria Computacional\Tareas\Trabajo Final\GEO2020_2TrabalhoFinalLuisMarin\Figures Final\QuadrilateInd2_37_nuvem2.npy.png |
| Triangulação inicial Delaunay | | |
| E:\Fernando\PUC 2020\Semestre 2\Geometria Computacional\Tareas\Trabajo Final\GEO2020_2TrabalhoFinalLuisMarin\Figures Final\Quadrilate_57_nuvem2_Delaunay.npy.png | E:\Fernando\PUC 2020\Semestre 2\Geometria Computacional\Tareas\Trabajo Final\GEO2020_2TrabalhoFinalLuisMarin\Figures Final\Quadrilate_33_Indirect1_nuvem2_Delaunay.npy.png | E:\Fernando\PUC 2020\Semestre 2\Geometria Computacional\Tareas\Trabajo Final\GEO2020_2TrabalhoFinalLuisMarin\Figures Final\QuadrilateInd2_33_nuvem2_Delaunay.npy.png |
| Triangulação inicial incremental | | |
| E:\Fernando\PUC 2020\Semestre 2\Geometria Computacional\Tareas\Trabajo Final\GEO2020_2TrabalhoFinalLuisMarin\Figures Final\Quadrilate_65_nuvem40.npy.png | E:\Fernando\PUC 2020\Semestre 2\Geometria Computacional\Tareas\Trabajo Final\GEO2020_2TrabalhoFinalLuisMarin\Figures Final\Quadrilate_41_Indirect1_nuvem40.npy.png | E:\Fernando\PUC 2020\Semestre 2\Geometria Computacional\Tareas\Trabajo Final\GEO2020_2TrabalhoFinalLuisMarin\Figures Final\QuadrilateInd2_41_nuvem40.npy.png |
| Triangulação inicial Delaunay | | |
| E:\Fernando\PUC 2020\Semestre 2\Geometria Computacional\Tareas\Trabajo Final\GEO2020_2TrabalhoFinalLuisMarin\Figures Final\Quadrilate_65_nuvem40_Delaunay.npy.png | E:\Fernando\PUC 2020\Semestre 2\Geometria Computacional\Tareas\Trabajo Final\GEO2020_2TrabalhoFinalLuisMarin\Figures Final\Quadrilate_40_Indirect1_nuvem40_Delaunay.npy.png | E:\Fernando\PUC 2020\Semestre 2\Geometria Computacional\Tareas\Trabajo Final\GEO2020_2TrabalhoFinalLuisMarin\Figures Final\QuadrilateInd2_40_nuvem40_Delaunay.npy.png |

Figura 4. Resultados de conversão em quadriláteros.

Ao analisar os resultados do método 1.1 Catmull Clark, pode-se concluir que, para todos os casos, o uso de uma malha triangular baseada em uma função incremental apresenta uma construção quadrilateral completa, embora difícil de reconhecer e sem nenhum padrão de comunicação entre os quadriláteros. Por outro lado, ao usar um triângulo de Delaunay, a malha resultante tende a criar grupos de pontos com uma estrutura panóptica.

O método 1.2 de agrupamento não garante a criação de quadriláteros, dada a condição convexa, o resultado ainda apresenta triângulos, embora os resultados sejam mais reconhecíveis a partir de uma triangulação de Delaunay.

Por fim, o método 1.3 apresenta uma malha que garante a quadrangulação, e em todos os casos apresenta uma malha com melhor aspecto, em relação aos métodos anteriores.

A Figura 5 mostra o tempo computacional para o método 1.1 Catmull-Clark:

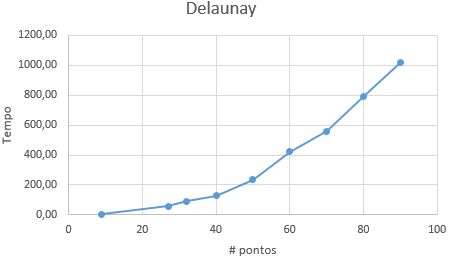
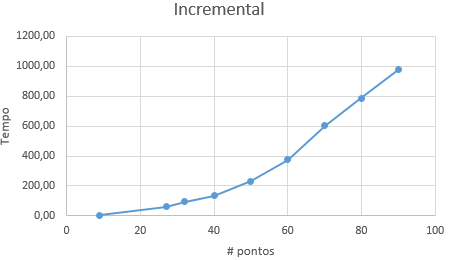


Figura 5. Tempo computacional Catmull-Clark.

Ao aumentar o número de pontos, o metodo Catmull-Clark exibe um comportamento O (nlogn), o alto tempo computacional é explicado pelo grande número de triângulos.

A Figura 6 mostra o tempo computacional para o método 1.2 agrupamento de triângulos:

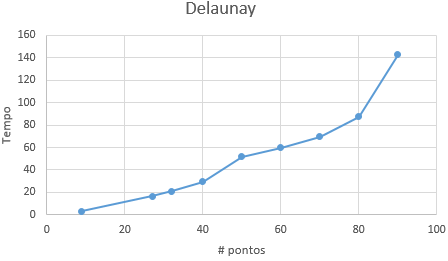
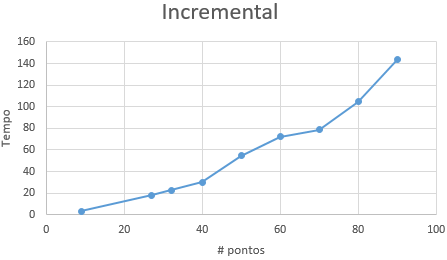


Figura 6. Tempo de agrupamento de triângulos.

Ao aumentar o número de pontos, o metodo de agrupamento exibe um comportamento que pode estar entre Ω(nlogn) e O (n2), embora o tempo total necessário seja inferior a 20% do registrado para o método 1.1.

A Figura 7 mostra o tempo computacional para o método 1.3 Clustering and Catnull-Clark:

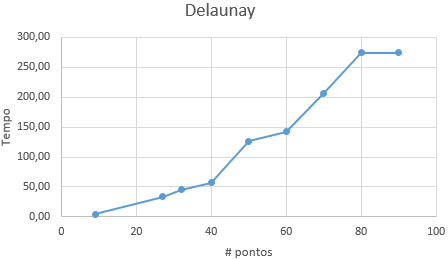
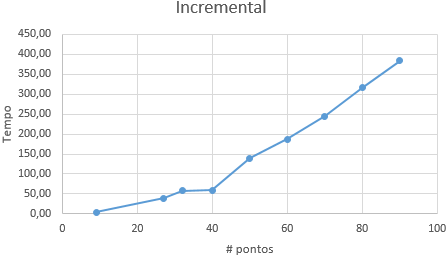


Figura 7. Tempo de Clustering and Catnull-Clark

Ao aumentar o número de pontos, o metodo 1.3 exibe um comportamento que pode estar localizado entre Ω(n) e O (nlogn).

1. **Conclusões**

**4.1** Em todos os casos, partindo de uma triangulação de Delaunay apresenta um resultado visualmente melhor.

**4.2** Usar o método de agrupamento para remover triângulos antes de aplicar o método catmull-clark reduz o tempo de cálculo necessário.

4.3 Embora o resultado usando o método 1.3 seja uma malha de quadrados, os pontos próximos representam áreas de confusão, onde é recomendado o uso de métodos de pós-processamento.

**Referências**

[1] E. Catmull and J. Clark, “Recursively generated B-spline surfaces on arbitrary topological meshes,” *Comput. Des.*, 1978, doi: 10.1016/0010-4485(78)90110-0.

[2] L. Velho, “Quadrilateral Meshing Using 4-8 Clustering,” *Proc. CILANCE’00*, 2000.