

6 Resultados

Este trabalho foi implementado usando a API OpenGL versão 4.2 (OpenGL 4.2, 2012) em conjunto com a OpenGL Shading Language 4.20 (GLSL 4.20, 2011). Os testes foram realizados para diferentes tipos de superfícies paramétricas e, no caso procedimental, também para diferentes tipos de algoritmos com complexidades variáveis. Na abordagem de mapas pré-computados, foram utilizadas texturas de normais e valores de profundidade com resoluções diferentes, a fim de verificar possíveis efeitos da amostragem de mapas variados sobre o desempenho.

O algoritmo do POM, apresentado na Seção 4.2, é uma solução baseada em imagens que fornece bons resultados, mas ainda é limitada em termos de qualidade visual. No entanto, conforme mostrado na Seção 4.3, a qualidade da imagem associado ao bom desempenho proporcionado por esta técnica foi o principal fator de influência na escolha desta técnica como referência para comparação dos resultados obtidos com a implementação usando tesselação, os quais serão apresentados nas seções seguintes.

Todos os experimentos foram avaliados utilizando um computador Intel Core 2 Duo 2.93 GHz com 4 GB RAM, GPU NVIDIA GeForce GTX 560 1 GB RAM no sistema operacional Windows 7 Professional, em janelas de resolução 800 x 600 *pixels*.

6.1 Qualidade de renderização

6.1.1. Mapas pré-computados

Em relação à qualidade visual, foi avaliada a capacidade do método apresentado em representar com precisão as feições descritas pelo mapa de altura em relação aos resultados obtidos com a implementação do POM. Para a análise

da qualidade visual, foram utilizados valores semelhantes de escala de profundidade e taxa de repetição da textura. Em ambos os casos, foram realizadas 64 subdivisões sobre a primitiva original, resultando em 8192 triângulos por superfície tessellada, para garantir o maior nível de detalhamento disponível (isto é, o máximo nível de tesselação permitido atualmente pela OpenGL).

A Figura 16 mostra comparações em pares das primitivas geométricas obtidas pela implementação baseada em tesselação (vistas na imagem superior de cada par) e pelo POM (as imagens inferiores de cada par). Observa-se que a qualidade resultante do método desenvolvido neste trabalho é superior em todos os casos, especialmente em função da correta representação de silhuetas, efeitos de oclusão causados por interpenetrações geometricamente corretas e ausência de distorções quando da observação de ângulos rasantes (que podem ser visualizadas nas regiões próximas às silhuetas das superfícies geradas com POM).

6.1.2. Abordagem procedimental

No caso das malhas cujos detalhes foram gerados procedimentalmente, não foi possível encontrar um parâmetro de comparação, já que este foi um experimento sobre a possibilidade de geração de detalhamento inteiramente na GPU, o que não era possível em versões anteriores do *pipeline* gráfico. Nesta abordagem, também foi usado o máximo nível de tesselação disponível para verificar o maior nível de detalhes que pode ser atingido com apenas uma primitiva subdividida.

A Figura 17 mostra os três efeitos procedimentais implementados neste trabalho – *lumpy*, *marble* e *ridge* – aplicados sobre as primitivas geométricas utilizadas como base. Observa-se a qualidade da computação dos detalhes internos da superfície mesmo com a ausência de uma textura de cor. Além disso, assim como na abordagem da Seção 5.2, observa-se claramente a computação das silhuetas dos objetos.



Figura 16: Comparação visual entre técnicas de detalhamento por meio de tesselação (linha superior de cada par) e simulação por pixel com POM (linha inferior). Os mapas utilizados em cada par foram intitulados rocks (a), tiles (b), parametric (c), saint (d), wall (e) e noise (f).

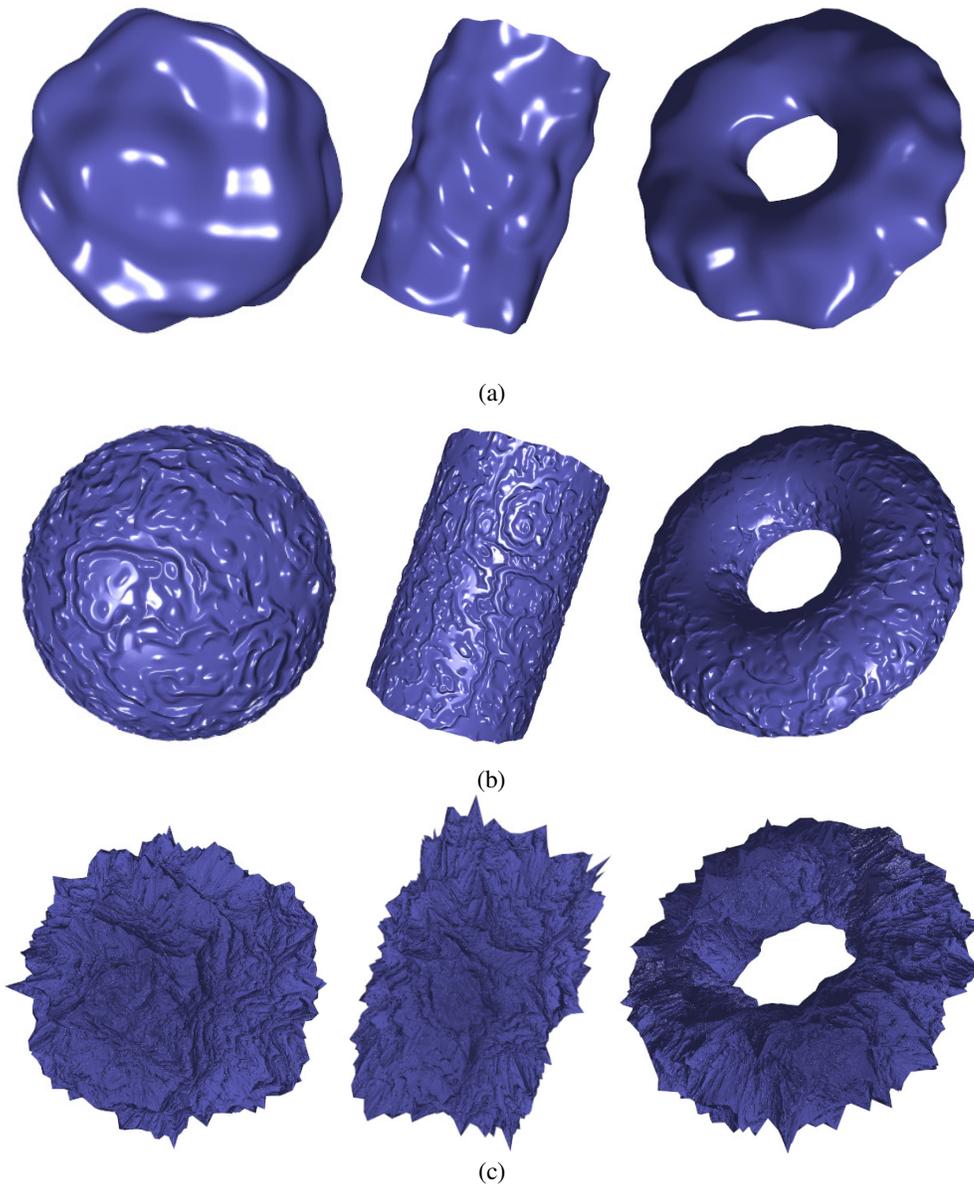


Figura 17: Superfícies detalhadas com os efeitos lumpy (a), marble (b) e ridge (c), gerados proceduralmente na GPU.

6.2 Desempenho

6.2.1. Mapas pré-computados

Uma comparação de desempenho entre ambas as abordagens é apresentada na Tabela 1, cujos dados foram transcritos no gráfico da Figura 18 para uma melhor visualização. Esta tabela relaciona o desempenho, medido em quadros por

segundo, obtido pela abordagem baseada em mapas de altura codificados em texturas com aquele obtido com a técnica de POM utilizando os mesmos mapas em cada caso, os quais, como indicado na tabela, possuem resoluções variáveis. O cálculo de sombras suaves (*soft shadows*), suportado pelo algoritmo do POM, foi desabilitado para que fossem obtidas as máximas medidas de desempenho. Além disso, o nível de tesselação para esta técnica foi reduzido a 16, resultando em apenas 512 triângulos por superfície, enquanto para a abordagem deste trabalho foram mantidas as 64 subdivisões. Neste cenário, o desempenho do método implementado aqui foi superior ao do POM em todos os casos, com uma média de ganho de aproximadamente 217,5%. É importante ressaltar que o POM é uma técnica baseada em traçado de raios cujo desempenho pode ser afetado pelo tamanho do objeto na janela de renderização. A fim de manter os testes de ambas as técnicas em condições semelhantes, todos os experimentos foram realizados sobre objetos com os mesmos valores paramétricos e a mesma distância para a câmera, de forma que tanto a malha subdividida em hardware quanto a renderizada com POM tenham os mesmos aspectos visuais em termos de tamanho ocupado na janela.

Textura	Superfície	Desempenho (FPS)		
		Tesselação	POM	Ganho (%)
Tiles (128 x 128) Figura 16(a)	Esfera	2327.615	1756.981	132.478
	Cilindro	2117.050	1472.409	143.781
	Torus	1916.711	1180.841	162.317
Rocks (128 x 128) Figura 16(b)	Esfera	2262.092	1348.397	167.762
	Cilindro	2093.285	1139.604	183.685
	Torus	1865.839	824.037	226.427
Saint (256 x 256) Figura 16(c)	Esfera	2336.967	1138.093	205.341
	Cilindro	2054.839	838.655	245.016
	Torus	1826.842	785.028	232.710
Parametric (256 x 256) Figura 16(d)	Esfera	2293.867	1046.563	219.181
	Cilindro	2004.270	755.918	265.144
	Torus	1904.460	719.364	264.742
Wall (512 x 512) Figura 16(e)	Esfera	2219.343	961.075	230.923
	Cilindro	2042.461	783.094	260.819
	Torus	1854.488	824.610	224.893
Noise (1024 x 1024) Figura 16(f)	Esfera	2148.558	894.761	240.126
	Cilindro	2025.543	748.155	270.738
	Torus	1880.545	785.809	239.313

Tabela 1: Comparação de desempenho entre detalhamento por meio de tesselação e simulação por pixel.

Por fim, a média do desempenho para cada mapa de altura é apresentada no gráfico da Figura 19, onde se observa a diferença obtida por ambas as técnicas quando do aumento da resolução dos mapas de altura (sem que tenham sido realizadas quaisquer alterações nos níveis de subdivisão). Nota-se também que, com o aumento da resolução dos mapas, ocorre uma sensível queda de desempenho na abordagem de tesselação, a qual se acentua a partir do mapa de 512x512, enquanto o desempenho obtido pelo POM se mantém mais estável para resoluções maiores.

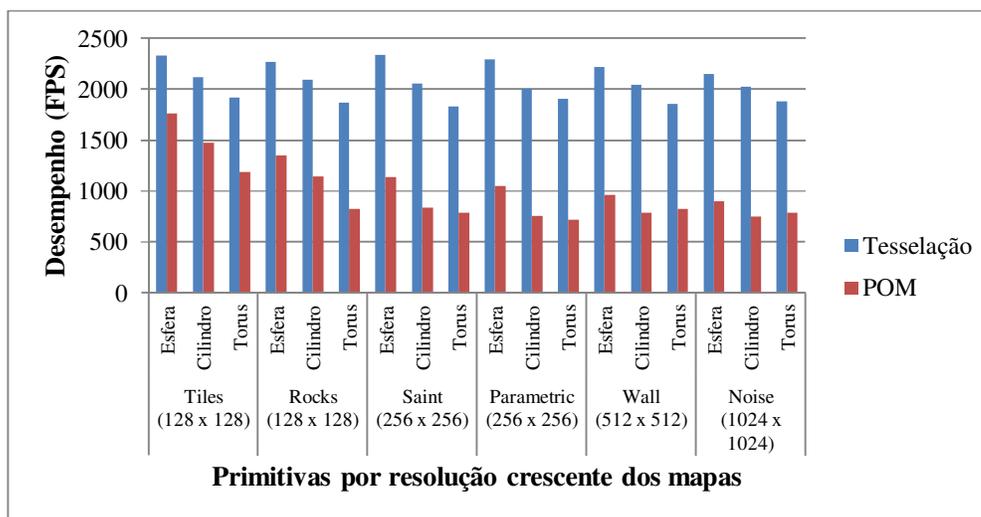


Figura 18: Relação de desempenho por primitiva entre tesselação e POM.

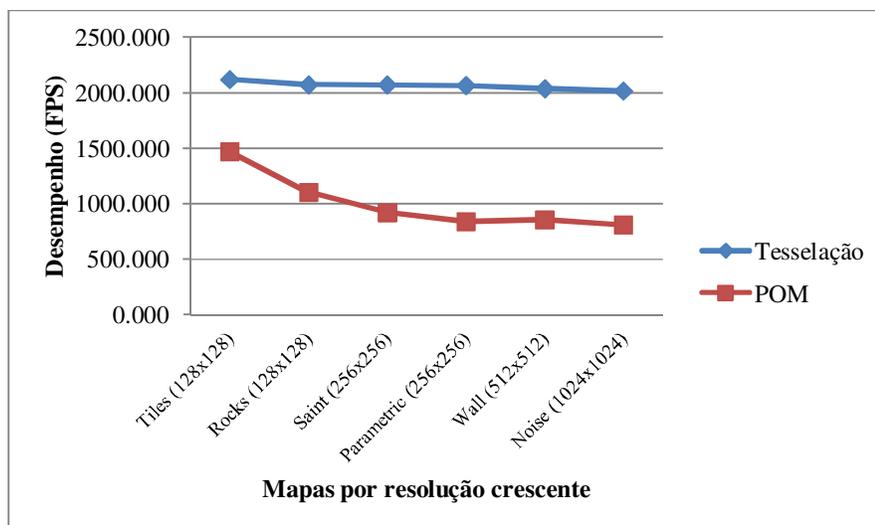


Figura 19: Média de desempenho por resolução dos mapas aplicados.

6.2.2. Abordagem procedimental

Como afirmado anteriormente, não foi possível encontrar uma técnica que pudesse ser usada como referência para comparação de resultados, então os valores fornecidos foram usados apenas para avaliação da factibilidade de implementação de tal tipo de detalhamento na GPU. A Tabela 2 mostra o desempenho obtido na computação procedimental dos efeitos *lumpy*, *marble* e *ridge* para diferentes números de instâncias simultâneas da superfície renderizada. Observa-se mais claramente com o auxílio da Figura 20 o efeito dessas múltiplas computações sobre o desempenho.

Número de Instâncias	Superfície	Desempenho (FPS)		
		Lumpy Figura 17(a)	Marble Figura 17(b)	Ridge Figura 17(c)
1	Esfera	2370.959	1185.179	495.697
	Cilindro	1798.420	711.913	345.395
	Torus	1827.016	757.204	317.355
10	Esfera	1977.952	882.853	273.843
	Cilindro	1506.645	593.815	243.994
	Torus	1494.529	614.571	216.904
100	Esfera	652.108	233.381	43.117
	Cilindro	511.597	204.167	48.974
	Torus	456.251	193.620	40.789
1000	Esfera	84.057	27.974	4.592
	Cilindro	66.791	26.947	5.462
	Torus	57.052	24.971	4.491
10000	Esfera	7.451	2.870	0.459
	Cilindro	7.170	2.813	0.552
	Torus	6.042	2.575	0.454

Tabela 2: Desempenho da abordagem de detalhamento procedimental.

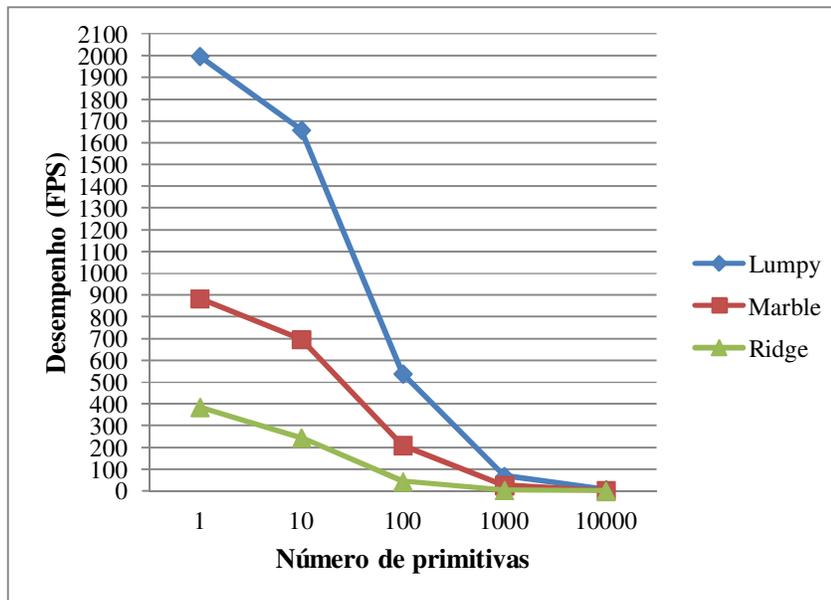


Figura 20: Desempenho médio por efeito procedimental.

Pela Listagem 7, apresentada na Seção 5.3.2, observa-se que existe um grande volume de operações realizadas por fragmento na computação dos efeitos procedimentais. O efeito *lumpy*, por exemplo, é baseado apenas na avaliação direta da função `noise()`, apresentando o maior desempenho dentre os tipos implementados.

Observa-se também pela Listagem 7 que a computação do efeito *ridge* adiciona sucessivamente novas avaliações da função de ruído, chamadas *octaves*, as quais sempre possuem o dobro da frequência da avaliação anterior e possuem influência direta na qualidade do ruído utilizado para a computação do efeito. Os valores apresentados na Tabela 2 para o desempenho proporcionado pelo efeito *ridge* foram obtidos através de oito iterações durante a computação dos *octaves*, as quais resultaram no efeito observado na Figura 17(c). A redução desse valor proporciona um desempenho maior ao custo da qualidade do ruído gerado, o que, por consequência, influencia na qualidade do efeito resultante.

Por fim, essas medidas mostram a eficiência do recurso de tesselação para detalhamento procedimental de superfícies, que, conforme mostra a Figura 17, produz resultados de alta qualidade visual apenas com o uso do alto poder de processamento da GPU. A variação de desempenho mostrada no gráfico da Figura 20 apenas ressalta a importância da utilização balanceada de métodos procedimentais e não procedimentais quando da renderização de ambientes complexos, conforme foi discutido na Seção 3.2.