

## 6 Resultados

Os Capítulos 2 e 3 descreveram o MMR proposto para as malhas de simulação em RNPs, analisando suas principais características. A análise de complexidade foi feita tanto para o algoritmo de construção da estrutura de MR quanto para os algoritmos de extração de malhas. Entretanto, essa análise não é suficiente para garantir que os tempos reais de processamento sejam satisfatórios para a aplicação nem está relacionada à qualidade das aproximações geradas. Além disso, é necessário verificar se as características de visualização do problema aqui tratado foram atendidas.

Dessa forma, este capítulo tem por objetivos: (a) analisar o desempenho empírico do algoritmo de pré-processamento, (b) analisar a qualidade das aproximações geradas pelo MMR, (c) analisar as taxas de interatividade alcançadas sob diversas condições e (d) verificar se os requisitos de visualização foram satisfeitos.

Para facilitar o entendimento das análises efetuadas, o capítulo é dividido em três etapas distintas: *pré-processamento*, *extração de malhas* e *características de visualização*.

## 6.1 Considerações Iniciais

A implementação do MMR proposto, com a qual os testes deste trabalho foram realizados, foi feita nas linguagens C e C++ e desenvolvida com a ferramenta Visual C++ 6.0 da Microsoft Corporation. Apesar do ambiente de desenvolvimento estar relacionado ao Sistema Operacional (S. O.) Windows, o código é portátil para as plataformas Linux e Unix. A interface de aplicação gráfica OpenGL [OpenGL99] foi utilizada para aproveitar possíveis recursos de alto desempenho disponibilizados por placas gráficas.

Os testes foram efetuados em quatro plataformas (plt.) de *hardware* distintas operadas pelo Microsoft Windows 2000, formando as quatro configurações descritas na Tabela 1.

**Tabela 1 - Plataformas de *hardware*.**

	Processador	Frequência (MHz)	Memória Principal (MB)	Processador Gráfico	Memória de Vídeo (MB)
<b>Plataforma I</b>	Intel Pentium III	850	256	-----	-----
<b>Plataforma II</b>	Intel Pentium IV	2.000	512 (RAM BUS)	-----	-----
<b>Plataforma III</b>	Intel Pentium III	850	256	Rage 128	32
<b>Plataforma IV</b>	Intel Pentium IV	2.000	512 (RAM BUS)	G-Force III	64

As malhas de simulação utilizadas para a bateria de testes estão descritas na Tabela 2, onde várias características foram relacionadas. O item “número de triângulos” (NT) foi inserido para proporcionar uma relação mais familiar com os tempos de renderização das malhas através de placas gráficas. Esse número é proporcional ao número de células ativas (NC):

$$NT = 12 \times NC.$$

A nomenclatura dada para as malhas obedece a ordem crescente em relação a NC e é seguida das terminações “bc” ou “cp” para indicar se a malha é *block-centered* ou *corner-point*, respectivamente. A única exceção é a I\_bc, que recebe esta denominação por possuir uma geometria bastante irregular (Figura 6.1), nada típica às malhas de simulação, sendo utilizada para testar o MMR em um caso desfavorável. Um fato importante é que as malhas A\_bc e D\_cp são discretizações de uma mesma geometria, ou seja, representam dois casos de estudo de um mesmo reservatório.

**Tabela 2 - Dados relativos às malhas de simulação.**

	A_bc	B_cp	I_bc	D_cp	E_bc	F_cp
Tipo	block	corner	Block	corner	block	corner
MAX_I	40	42	120	83	95	175
MAX_J	34	30	100	45	32	154
MAX_k	5	11	8	23	126	12
Nº de Células Ativas (NC)	3.703	9.033	11.850	31.550	42.065	50.964
Nº de Células Inativas (NI)	3.097	4.827	84.150	54.335	340.975	272.436
Nº de Triângulos (NT)	44.436	108.396	142.200	378.600	504.780	611.568
Nº de Colapsos (NCOL)	10.002	25.408	29.845	90.026	117.495	135.965
Nº de Prop. Iniciais	26	47	53	49	52	71
Nº de Prop. Recorrentes	7	4	5	6	5	6
Nº de Intervalos de Tempo	126	2	10	10	5	27

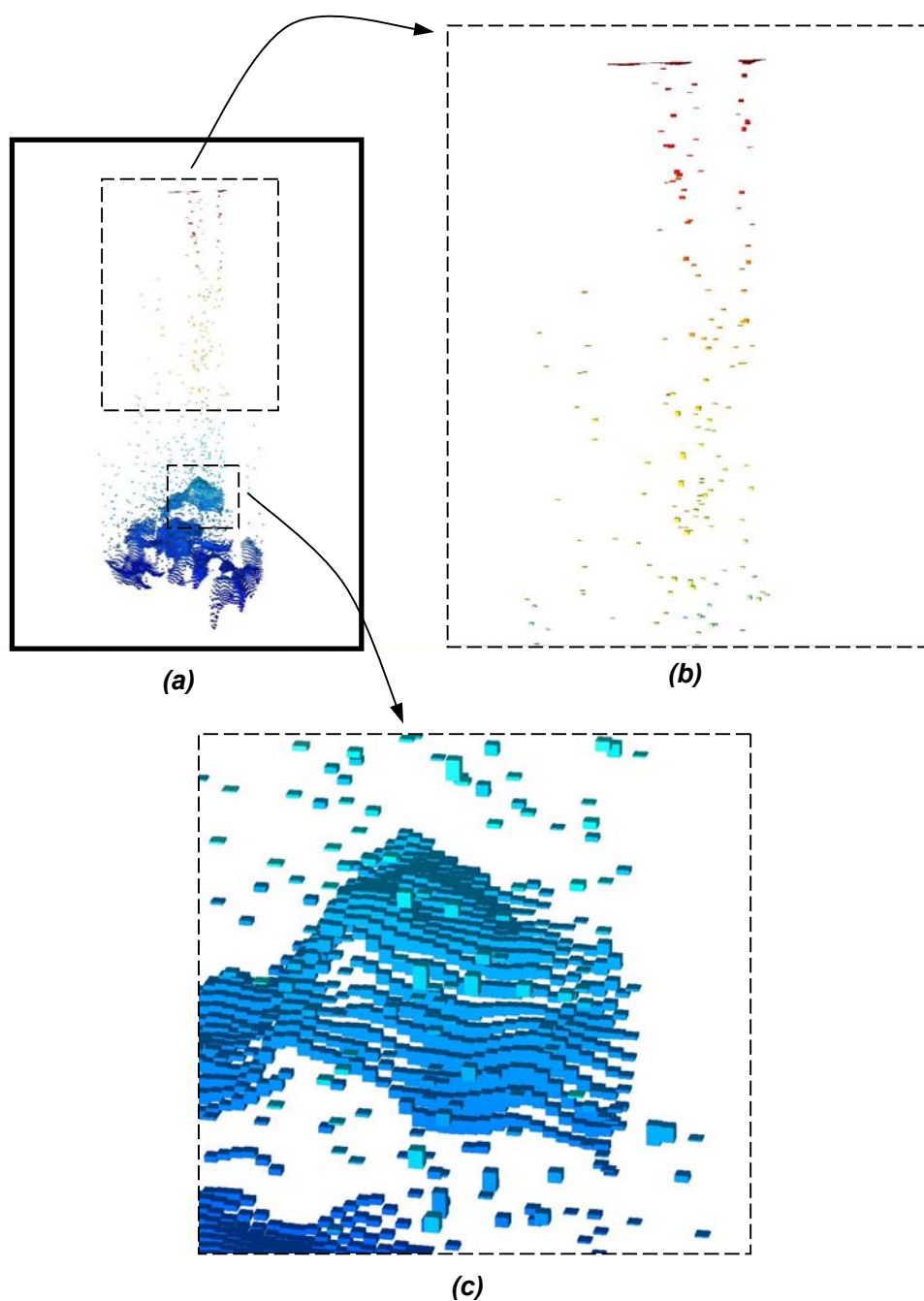


Figura 6.1 – Malha  $I_{bc}$ . Em (a), vista geral da malha  $I_{bc}$ . Em (b), região com comportamento bastante irregular e na qual praticamente não há faces em contato. Em (c), região melhor comportada, mas onde ainda percebe-se muita irregularidade.

## 6.2 Pré-processamento

Os testes referentes à etapa de pré-processamento visam analisar o tempo de construção da estrutura de multi-resolução bem como a quantidade de memória despendida nesta etapa. Para evitar efeitos indesejados na medição do tempo, apenas o tempo de **CPU** (Unidade Central de Processamento) consumido pelo processo foi medido. As plataformas de *hardware* utilizadas foram as plataformas I e II, observando que a quantidade de memória principal existente é suficiente para evitar a ocorrência de transferência de dados (*swap*) entre a memória principal e a secundária.

### 6.2.1 Análise do Tempo de Pré-processamento

Devido à característica do algoritmo, a aquisição dos tempos é feita em duas fases.

#### *1ª Fase*

A fase inicial é formada pelos três primeiros passos do algoritmo descrito na seção 4.1.3 e relacionados a seguir:

*1º passo: constrói o espaço de colapsos eliminando células inativas*

*2º passo: constrói o heap*

*3º passo: constrói a floresta inicial*

O primeiro passo é linear no número total de células e não apenas no número de células ativas, pois há a necessidade de percorrer as células inativas para desconsiderá-las. O segundo depende diretamente do número de colapsos e o terceiro depende do número de células ativas.

#### *2ª Fase*

A segunda fase é formada pelo laço de repetição principal, que é  $O(n \log n)$ , onde  $n$  é o número de células ativas. Nessa fase, são realizadas  $n$  iterações e cada uma delas possui complexidade  $O(g \log n)$ , onde  $g$  é o grau do nó escolhido. É importante observar que a variável  $g$  está sendo limitada pela constante  $G$ , à qual atribuiu-se o valor 15 nesta implementação (ver seção 4.1.2).

### **Resultados**

A Tabela 3 descreve os resultados obtidos e o Gráfico 1 e o Gráfico 2 permitem uma análise mais detalhada desses resultados. Nos gráficos, a curva “Inicial” representa os tempos da primeira fase do algoritmo adquiridos ao final do processamento de cada caso de teste em função do número de células ativas. O comportamento oscilatório deve-se às diferentes dependências atribuídas aos passos do algoritmo. Um bom exemplo são os comportamentos nas malhas E\_bc e F\_cp: apesar da F\_cp possuir cerca de 8.000 células ativas e 17.000 colapsos a mais, as aproximadamente 70.000 células inativas a menos fizeram com que os tempos fossem próximos: 1,57 segundos para a E\_bc e 1,62 segundos para a F\_cp, ambos na plataforma I.

As demais curvas permitem averiguar o comportamento no tempo da segunda fase do algoritmo. Cada curva representa um caso de teste e indica o comportamento em função do número de células ativas existentes em cada iteração. Apesar do decrescimento constante do número de células ativas, as curvas apresentam um comportamento bem próximo do linear.

Em algumas curvas, em especial a E\_bc, nas iterações finais, quando o número de células se aproxima de zero, pode-se perceber a influência da variável  $g$  nos tempos adquiridos. O aumento desses tempos nessa região da curva acontece devido à existência de iterações nas quais todos os colapsos remanescentes são proibidos e, assim, necessariamente nós de grau maior do que a constante  $G$  são selecionados.

Tabela 3 – Tempos de execução do algoritmo de pré-processamento anotados nas plataformas de *hardware* I e II, em segundos.

	Tempos de execução (s)					
	1ª fase		2ª fase		Total	
	Plt. I	Plt. II	Plt. I	Plt. II	Plt. I	Plt. II
A_bc	0,08	0,06	0,30	0,16	0,38	0,22
B_cp	0,25	0,14	0,79	0,46	1,041	0,6
I_bc	0,39	0,21	1,253	0,511	1,292	0,721
D_cp	0,911	0,48	3,465	2,063	4,376	2,543
E_bc	1,573	0,912	4,676	2,834	6,249	3,746
F_cp	1,623	0,891	4,456	2,659	6,079	3,55

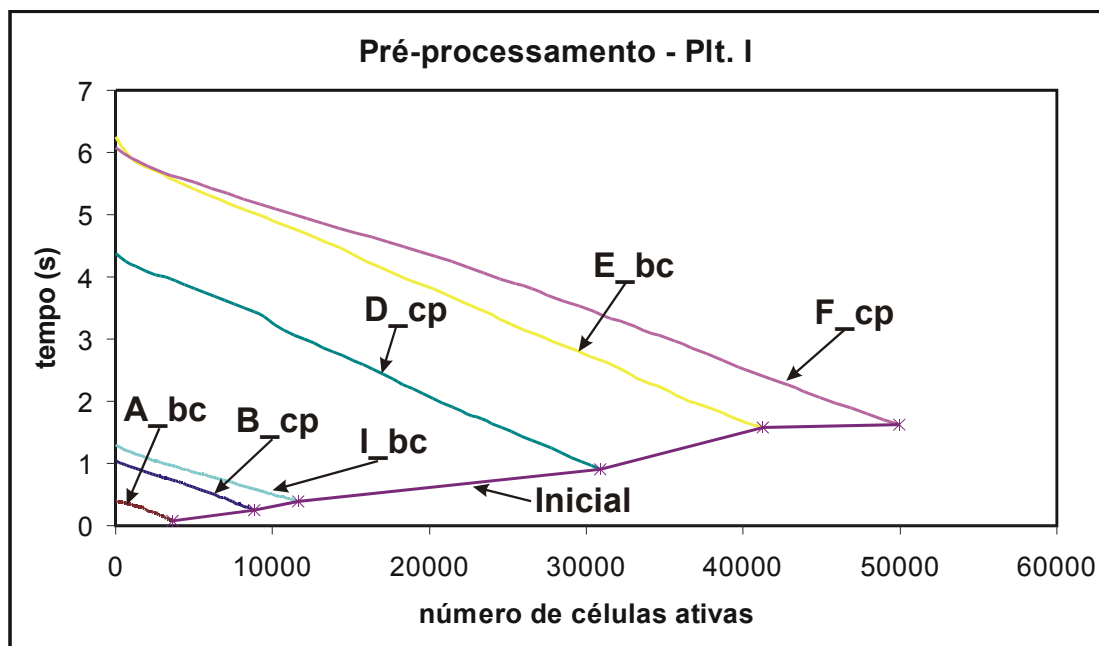
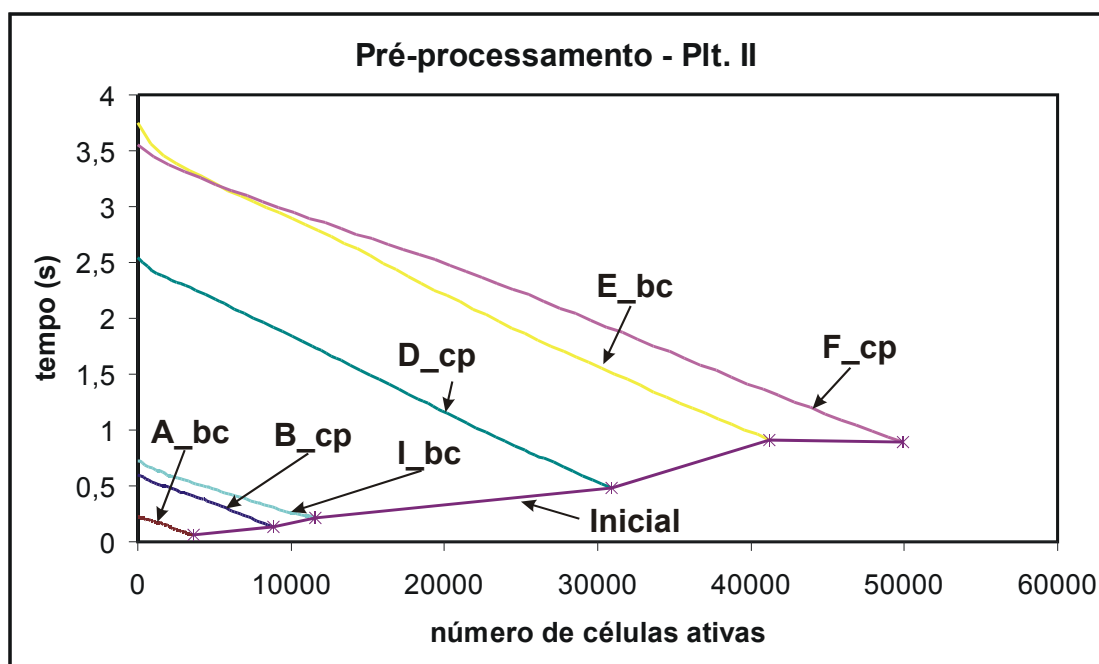


Gráfico 1 – Pré-processamento (plat. I). Resultados da fase de pré-processamento adquiridos na plataforma I. A curva “Inicial” representa as tomadas de tempo da primeira fase do algoritmo. As demais curvas representam, para cada caso de teste, o comportamento da segunda fase do algoritmo.



**Gráfico 2 – Pré-processamento (plat. II). Resultados de pré-processamento adquiridos na plataforma II. O gráfico é similar ao Gráfico 1.**

### ***Conclusões Parciais***

Os resultados mostraram que a fase inicial do algoritmo de construção requer uma fração do tempo total menor do que a fração correspondente à segunda fase do algoritmo. Aproximadamente, a primeira fase consumiu  $\frac{1}{4}$  do tempo total, sobrando  $\frac{3}{4}$  que foram consumidos pela segunda fase.

Em contraposição a essa relação constante que foi percebida, espera-se que, com o crescimento do número de células nas malhas de simulação, desde que a relação entre NC e NI seja mantida, o tempo consumido pela segunda fase tenda a prevalecer sobre o tempo consumido pela primeira fase, pois a complexidade da segunda fase é maior.

Considerando os objetivos a serem atingidos, a principal constatação dos testes realizados foi o ótimo desempenho obtido, alcançando menos de 7 segundos de processamento na plataforma I. Considerando malhas com complexidades semelhantes às testadas, esses resultados atendem satisfatoriamente à característica de visualização número 5, que exige uma pequena latência para o usuário antes de iniciar a visualização.



### 6.2.2 Consumo de Memória

O algoritmo de construção da estrutura de MR não possui boa localidade de memória. A cada passo do algoritmo, posições distantes do bloco de memória podem ser acessadas. Isto faz com que o *swap* seja intenso quando os dados não são totalmente acomodados na memória principal, prejudicando muito o desempenho do algoritmo. Dessa forma, é especialmente importante analisar a quantidade de memória consumida para especificar os requisitos mínimos necessários para um bom desempenho.

A Tabela 4 mostra as estruturas utilizadas pela implementação e seus respectivos consumos de memória. As estruturas *estr\_célula* e *estr\_nó\_hier* são as mesmas introduzidas no Capítulo 5. A estrutura *estr\_nó\_hier*, entretanto, está modificada para prever possíveis informações que possam ser adicionadas à hierarquia de MR.

É válido salientar que o modelo original do reservatório foi representado por uma **tabela de vértices** e uma **tabela de células**. Por sua vez, o grafo de adjacência foi implementado através de uma **tabela de nós** e uma **tabela de arcos**. A tabela de nós associa a cada nó do grafo (*estr\_nó\_grafo*) referências para todos os arcos que incidem sobre ele e a tabela de arcos associa a cada arco (*estr\_arco*) referências para seus nós extremos (Figura 6.2).

A análise do consumo de memória é feita em relação a NC, supondo-se que as células ativas estejam bem distribuídas espacialmente e que MAX\_I, MAX\_J e MAX\_K possuam valores próximos. Com essas suposições, o número de arcos é aproximadamente igual a  $3 \times NC$  e o grau de um nó do grafo de adjacência é igual a 6 no início do processamento, excetuando-se as células mais externas do reservatório.

Por ser um algoritmo de dizimação de células, o estado inicial do algoritmo é o que requer mais memória. Nesse momento, estão armazenados não só as tabelas de vértices e de células como também, em tamanho máximo, a tabela de arcos, a tabela de nós (com suas listas de 6 arcos incidentes) e o *heap*, além da floresta inicial e do

repositório pré-alocado de nós da hierarquia com seus  $(2 \times NC) - 1$  nós a serem preenchidos durante o laço de repetição. A utilização de um repositório de nós pré-alocados é feita para evitar que a alocação dinâmica desses nós prejudique o desempenho.

**Tabela 4 - Estruturas utilizadas e seus respectivos consumos de memória.**

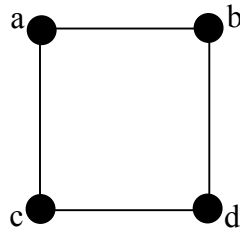
<b>Estrutura</b>	<b>Atributos</b>	<b>Bytes</b>
<i>estr_nó_grafo</i>	- 6 índices para arcos <sup>10</sup> ( <i>int</i> )	24
<i>estr_arco</i>	- erro geométrico ( <i>float</i> ) - índice no <i>heap</i> ( <i>int</i> ) - direção de colapso ( <i>byte</i> ) - 2 índices para os nós extremos ( <i>int</i> )	17
<i>estr_elem_heap</i>	- erro geométrico ( <i>float</i> ) - índice cruzado para o arco ( <i>int</i> ) - grau do nó ( <i>int</i> )	12
<i>estr_vértice</i>	- 3 coordenadas ( <i>float</i> )	12
<i>estr_célula</i>	- 6 normais ( <i>estr_vértice</i> ) - 8 índices para vértices ( <i>int</i> )	104
<i>estr_AABB</i>	- 2 <i>estr_vértice</i>	24
<i>estr_nó_hier</i>	- 8 informações adicionais ( <i>int</i> ) - direção de colapso ( <i>byte</i> ) - 3 ponteiros para filhos e pai (32 bits) - 1 erro geométrico ( <i>float</i> ) - 1 ponteiro para Célula (32 bits)	53
<i>estr_nó_floresta</i>	- 1 ponteiro para o nó raiz da hierarquia (32 bits)	4

Além disso, durante a construção da hierarquia, é necessário armazenar as macro-células associadas às raízes de cada árvore da floresta (as demais não são necessárias,

<sup>10</sup> Na verdade, a lista de arcos que compõe a estrela de um nó é dinâmica, variando à medida que os colapsos são realizados. O valor fixo, 6, está sendo utilizado para permitir o cálculo do consumo de memória no início do algoritmo.

pois podem ser reconstruídas posteriormente). Assim, no pior caso, estão armazenadas  $NC/2$  macro-células durante a construção.

**Grafo de adjacência**



**Tabela de arcos**

0	a – b
1	a – c
2	b – d
3	c – d

**Tabela de nós**

a	→	0	1
b	→	0	2
c	→	1	3
d	→	2	3

Figura 6.2- Esquema da implementação do grafo de adjacência.

A Tabela 5 mostra o consumo de memória no estado inicial do algoritmo e o consumo no pior caso gasto com as macro-células da floresta.

**Tabela 5 - Consumo de memória durante a construção do algoritmo. Excetuando-se a última linha, todas representam o consumo no estado inicial.**

	Bytes	
Tabela de vértices	$12 \times (8 \times NC) = 96 \times NC$	Modelo original
Tabela de células	$104 \times NC$	
Tabela de nós	$24 \times NC$	Estado inicial da construção
Tabela de arcos	$17 \times (3 \times NC) = 51 \times NC$	
Heap	$12 \times (3 \times NC) = 36 \times NC$	
Repositório de nós da hierarquia	$53 \times 2 \times NC$	
Floresta inicial	$4 \times NC$	
Macro-células	$104 \times (NC/2) = 52 \times NC$	Interior do laço de repetição

Os cálculos foram realizados considerando a situação na qual não há compartilhamento de vértices entre células. Dessa forma, a tabela de vértices possui  $(8 \times NC)$  vértices, o que faz com que a quantidade de memória requerida para armazenar o modelo original (tabela de vértices + tabela de células) seja

$$(96 \times NC) + (104 \times NC) = (200 \times NC) \text{ bytes.}$$

As estruturas de gerenciamento da construção (tabelas de arcos, nós e o *heap*) ocupam  
 $(24 \times NC) + (51 \times NC) + (36 \times NC) = (111 \times NC) \text{ bytes}.$

E a hierarquia pré-alocada juntamente com a floresta inicial e suas macro-células ocupam

$$(53 \times 2 \times NC) + (4 \times NC) + (52 \times NC) = (162 \times NC) \text{ bytes}.$$

Portanto, a quantidade total de memória consumida pelo algoritmo de construção é

$$(200 \times NC) + (111 \times NC) + (162 \times NC) = (473 \times NC) \text{ bytes}.$$

Isso significa que a quantidade de memória exigida para o algoritmo de construção é, aproximadamente, 136% maior do que a quantidade de memória necessária para armazenar apenas o modelo original.

Para finalizar a análise, é necessário verificar o quanto de memória é consumido durante a visualização. Após o algoritmo de construção, a hierarquia de células hexaédricas está vazia, sem células (apenas as relativas às folhas) e sem *aabbs*. É preciso, então, percorrer a árvore de baixo para cima, recalculando as macro-células e calculando as *aabbs* de cada nó. Para isto, a informação que indica a direção de colapso é fundamental, pois basta aplicar o padrão de colapso correspondente para obter a geometria correta da macro-célula. Todo esse processo é feito em tempo linear e está embutido nas tomadas de tempo realizadas na seção anterior.

Após essa reconstrução, a hierarquia de células é composta por  $(2 \times NC) - 1$  nós da estrutura,  $NC - 1$  macro-células e  $NC - 1$  *aabbs* e, portanto, o consumo do modelo original somado ao da hierarquia é de aproximadamente

$$(200 \times NC) + (53 \times 2 \times NC) + (104 \times NC) + (24 \times NC) = (434 \times NC) \text{ bytes}.$$

Ou seja, a hierarquia de células hexaédricas requer aproximadamente 120% a mais de memória em relação ao modelo original.

A Tabela 6 mostra os valores de consumo de memória que seriam gastos em cada malha segundo as estimativas teóricas e os valores reais de consumo medidos durante a execução do programa. Estes foram adquiridos de forma aproximada através da observação do gerenciador de memória do Windows 2000. Os consumos reais relativos ao modelo original não correspondem exatamente aos valores esperados pela análise teórica, pois, dentre outros fatores, o leitor dos dados de simulação armazena, adicionalmente, alguns cabeçalhos com informações descritivas a respeito das propriedades e das grandezas físicas da simulação, sendo essas informações consideradas no consumo do modelo original.

**Tabela 6 – Valores aproximados, em MB, relativos ao consumo de memória estimado e ao consumo de memória realmente gasto durante a execução do programa medidos através do gerenciador de memória do Windows 2000.**

	Cálculo estimado de consumo (em MB)			Consumo real (em MB aproximadamente)		
	Modelo original (M)	Construção da hierarquia, desconsiderando M	Hierarquia de células hexaédricas	Modelo original (M')	Construção da hierarquia, desconsiderando M'	Hierarquia de células hexaédricas
<b>A_bc</b>	0,706	0,964	0,826	1,0	0,9	1,0
<b>B_cp</b>	1,722	2,351	2,015	2,5	2,9	2,5
<b>I_bc</b>	2,26	3,085	2,644	3,2	4,8	3,0
<b>D_cp</b>	6,01	8,203	7,031	7,1	11,0	7,9
<b>E_bc</b>	8,023	10,95	9,38	13,2	17,0	10,5
<b>F_bp</b>	9,7	13,26	11,37	14,0	19,2	12,0

### ***Conclusões Parciais***

O estudo efetuado permite inferir a quantidade mínima de memória exigida pelo MMR para garantir um bom desempenho. No estudo teórico, a memória adicional exigida é aproximadamente 136% maior do que a quantidade necessária para armazenar o modelo original. Nos testes práticos, percebem-se algumas variações, o pior caso sendo de 154% de memória adicional (Tabela 7- coluna 4).

Outro valor importante é o do consumo de memória da hierarquia de células hexaédricas. O estudo teórico indicou um índice de aproximadamente 120% de memória adicional, mas, na prática, o pior caso foi de 111% (Tabela 7- coluna 6).

**Tabela 7 – Valores aproximados, em MB, relativos ao consumo real de memória, juntamente com a medida comparativa do custo adicional.**

	<b>Modelo original (M')</b>	<b>Construção da hierarquia desconsiderando M' (C)</b>	<b>Relação entre C e M' (%)</b>	<b>Hierarquia de células hexaédricas (H)</b>	<b>Relação entre H e M' (%)</b>
<b>A_bc</b>	1,0	0,9	<b>90%</b>	1,0	<b>100%</b>
<b>B_cp</b>	2,5	2,9	<b>116%</b>	2,5	<b>100%</b>
<b>I_bc</b>	3,2	4,8	<b>150%</b>	3,0	<b>93%</b>
<b>D_cp</b>	7,1	11,0	<b>154%</b>	7,9	<b>111%</b>
<b>E_bc</b>	13,2	17,0	<b>129%</b>	10,5	<b>79%</b>
<b>F_bp</b>	14,0	19,2	<b>137%</b>	12,0	<b>85%</b>

### 6.3 Extração de Malhas

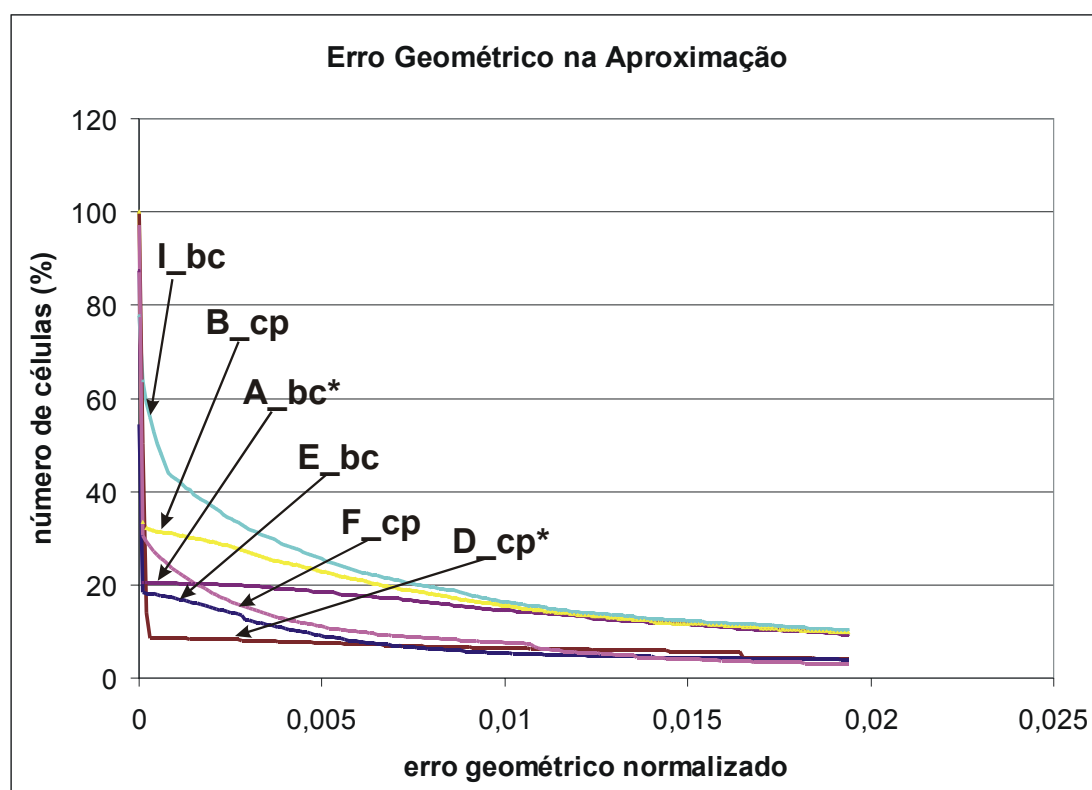
Esta seção tem por objetivo analisar a qualidade das aproximações geradas pelo MMR e a eficiência dos algoritmos de extração. Inicialmente, a qualidade das aproximações é medida na seção Erro Geométrico nas Aproximações. Posteriormente, os índices de redução da complexidade das malhas são analisados na seção Erro Projetado e Redução de Complexidade e, finalmente, a eficiência dos algoritmos é medida na seção Taxas de Interatividade.

#### 6.3.1 Erro Geométrico nas Aproximações

A qualidade das aproximações geradas é medida, nesta seção, através do erro geométrico armazenado pela estrutura de MR. Como foi visto, este erro representa uma aproximação da distância entre o modelo aproximado e o original.

O Gráfico 3 analisa a variação do número de células nas aproximações em função dos erros geométricos, que foram normalizados entre 0% e 20% da diagonal da caixa envolvente do modelo. As aproximações foram geradas pelo algoritmo de extração dependente do erro geométrico.

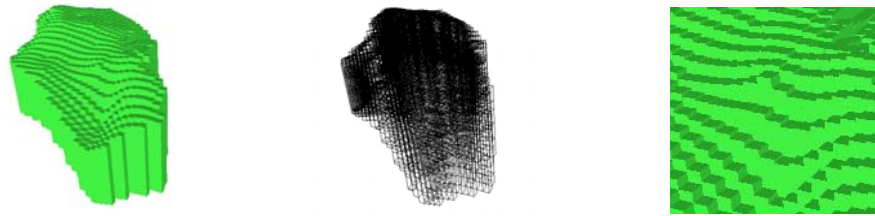
Excetuando-se a malha I\_bc, todas as malhas conseguem ser reduzidas em até 34% sem que o erro geométrico introduzido ultrapasse o valor de 0,0002.



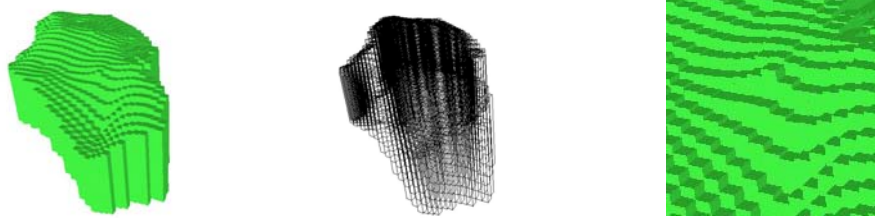
**Gráfico 3 – Erro geométrico nas aproximações. Resultados obtidos pelo algoritmo de extração dependente do erro geométrico submetido a consecutivas parametrizações de erro. Os erros geométricos estão normalizados entre 0% e 20% das diagonais das caixas envolventes dos reservatórios.**

Para melhor visualizar o erro normalizado utilizado, a Figura 6.3 e a Figura 6.4 ilustram o processo de degradação que ocorre com as malhas A\_bc e F\_cp. Na A\_bc, percebe-se que a malha pode ser representada sem erro geométrico com 87% de suas faces originais e com erro muito pequeno com 20,5%. A partir desta etapa, a redução da malha acarreta a inserção de erros mais significativos. Um fenômeno análogo ocorre com o exemplo F\_cp, porém o erro cresce numa taxa menor após a fase inicial

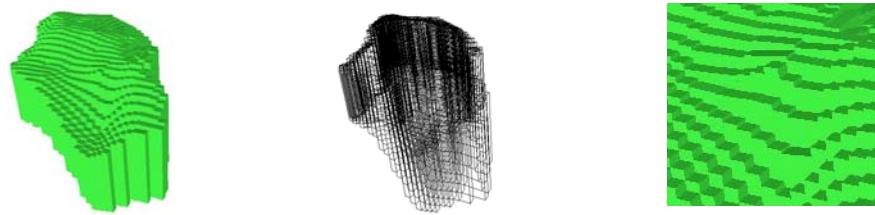
de super-amostragem. No Gráfico 3, pode-se perceber esta diferença entre as taxas de crescimento do erro.



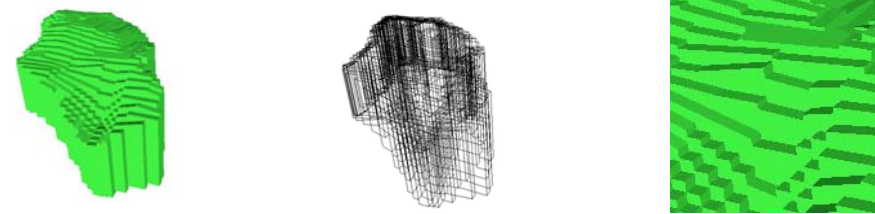
(a) 0,0 de erro geométrico e 87% do total de faces



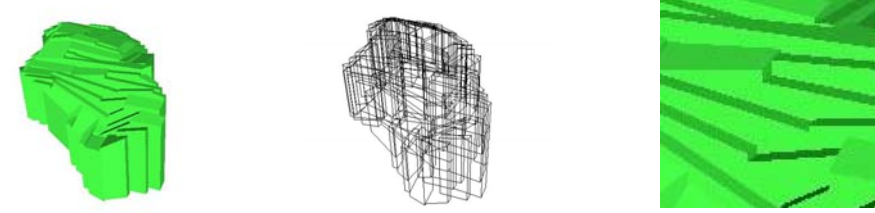
(b) 0,0001 do erro geométrico e 20,5% do total de faces



(c) 0,01 do erro geométrico e 14,4% do total de faces



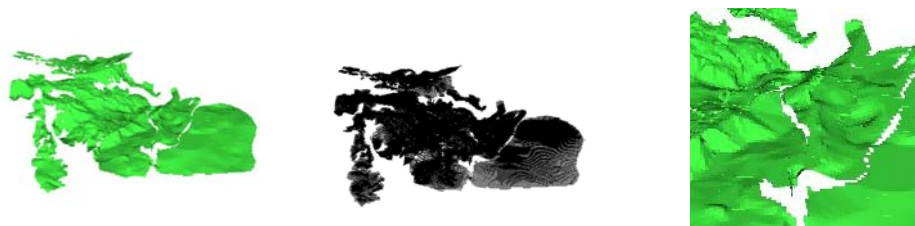
(d) 0,03 do erro geométrico e 6,4% do total de faces



(e) 0,07 do erro geométrico e 1,9% do total de faces

**Figura 6.3 - Degradação da malha A\_bc. Na coluna central, o respectivo desenho em *wireframe* e, na coluna à direita, a ampliação de uma pequena região do reservatório para possibilitar a melhor percepção do erro introduzido.**





(a) 0,0 de erro geométrico e 97,3% do total de faces



(b) 0,0001 do erro geométrico e 30,5% do total de faces



(c) 0,01 do erro geométrico e 7,6% do total de faces



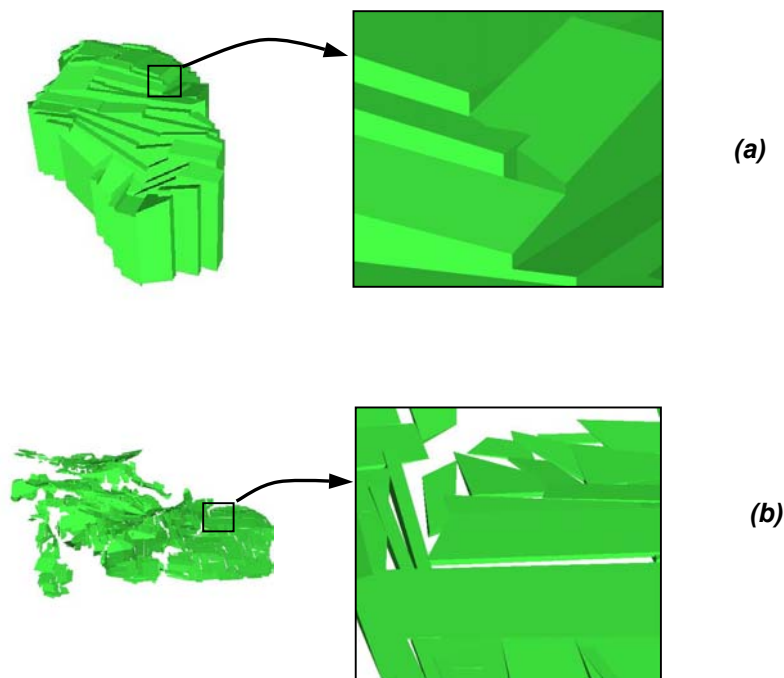
(d) 0,03 do erro geométrico e 1,9% do total de faces



(e) 0,07 do erro geométrico e 0,7% do total de faces

**Figura 6.4 - Degradação da malha F\_cp. Na coluna central, o respectivo desenho em *wireframe* e, na coluna à direita, a ampliação de uma pequena região do reservatório para possibilitar a melhor percepção do erro introduzido.**

A Figura 6.5 ilustra em detalhe as aberturas e as interseções introduzidas nas aproximações. Elas são bastante evidentes nas aproximações de baixa resolução.



**Figura 6.5 – Aberturas e interseções criadas nas aproximações geradas pelo MMR proposto neste trabalho.**

### ***Conclusões Parciais***

Primeiramente, o fato das malhas poderem ser reduzidas drasticamente sem a inserção de erros expressivos indica que elas estão super-amostradas, o que não é uma surpresa, visto que a geometria das malhas é presa às malhas de referência, não possuindo curvatura nas colunas de células dispostas em profundidade. Nessas colunas, o único fator que pode impedir uma redução sem erro das taxas de amostragem é o possível espaçamento existente entre camadas consecutivas.

As reduções obtidas com as malhas D\_cp e E\_bc, que possuem os maiores números de camadas (MAX\_K é igual a 23 e 126, respectivamente), são fortes evidências de que a restrição à malha de referência realmente influencia a capacidade de redução sem erro, uma vez que as reduções nessas malhas obtiveram os menores valores. Portanto, há uma grande probabilidade de que as taxas de redução obtidas variem proporcionalmente ao número de camadas das malhas.

Além disso, as ilustrações mostraram que a inserção de aberturas e interseções nas aproximações faz com que, gradativamente, as malhas percam suas características geométricas iniciais, de forma que as aproximações de baixa resolução são formadas por um conjunto desestruturado de células.

### 6.3.2 Erro Projetado e Redução de Complexidade

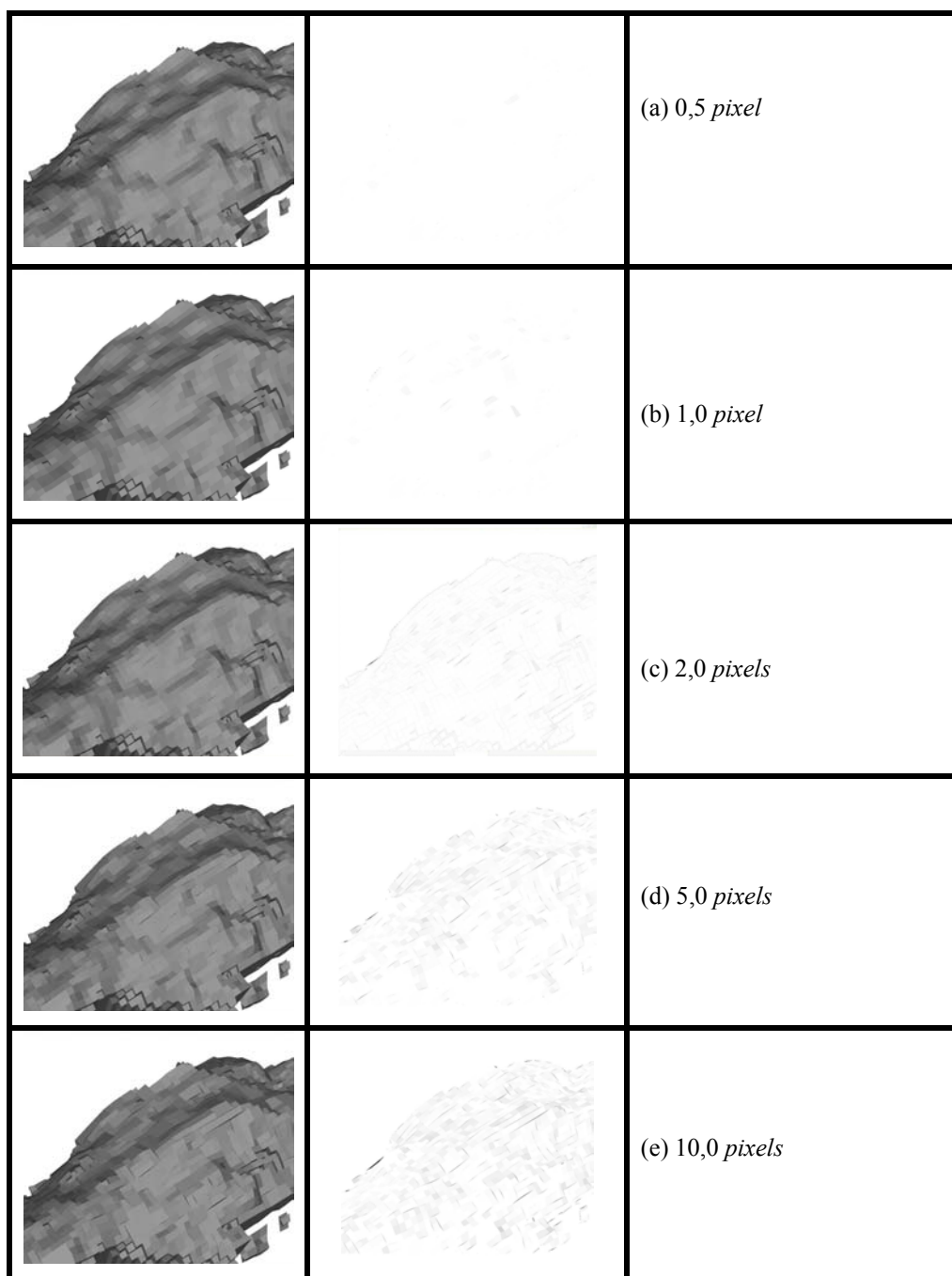
O objetivo desta seção é analisar a redução da complexidade das aproximações geradas pelo algoritmo de extração dependente da câmera, utilizando-se, em particular, uma câmera em perspectiva. Uma vez fixado um limite máximo de erro projetado permitido (parâmetro do algoritmo), em *pixels*, esta redução é determinada pela distância à câmera (quanto mais distante, maior é a redução) e pelo descarte das regiões da malha que estão fora do volume de visão.

A Figura 6.6 relaciona, perceptualmente, diferentes parametrizações de erro projetado à similaridade de aparência. Essa relação é válida somente para a resolução de tela (*viewport*) que foi utilizada, 1000x800 *pixels*. Na coluna da esquerda, há imagens geradas pelo algoritmo a partir de um único ponto de vista sob diferentes parametrizações. Na coluna central, estão suas imagens **diferença**<sup>11</sup> em relação à imagem gerada pela resolução mais alta do reservatório sob o mesmo ponto de vista e mesma resolução de tela. Na coluna da direita, está assinalado o parâmetro de erro utilizado. Com o aumento do erro projetado, há também o aumento do erro de similaridade<sup>12</sup>, sendo que é importante notar que o erro é visualmente pequeno até a parametrização de 1.0 *pixel*.

---

<sup>11</sup> Uma *imagem diferença* é aquela cujos *pixels* são dados pelo módulo da diferença, *pixel a pixel*, entre duas imagens quaisquer de mesmo tamanho.

<sup>12</sup> O erro de similaridade entre duas imagens  $I_1$  e  $I_2$  de mesmas dimensões é dado pela média dos *pixels* da *imagem diferença* entre elas.



**Figura 6.6 – Similaridade de aparência.** Na coluna à esquerda, as imagens geradas a partir das aproximações. Na coluna central, suas imagens diferença em relação à imagem gerada pela maior resolução do reservatório. Na coluna à direita, os erros utilizados como parâmetros para a geração das aproximações. As imagens foram geradas com a resolução 1000x800 *pixels*.

### ***Bateria de testes***

Para realizar os testes, foram traçados quatro tipos de caminho para cada malha. A Tabela 8 descreve alguns dados sobre eles. O campo “marcas em destaque” se refere a

pontos específicos do caminho que são vistos no decorrer deste texto e o campo “quantidade de quadros” indica quantos quadros são gerados durante o percurso. Os caminhos classificam-se em: **Básico 1 (B1)**, **Básico 2 (B2)**, **Básico 3 (B3)** e **Livre 1 (L1)**.

A Figura 6.7, a Figura 6.8 e a Figura 6.9 ilustram os caminhos básicos para o caso de teste F\_cp. Os caminhos B1, aproximadamente sob a vista superior (perpendicular e acima do plano formado pelos eixos  $x$  e  $y$ ), realizam apenas uma aproximação seguida de um afastamento com o foco de visão sempre voltado para o centro do reservatório.

Os caminhos B2 utilizam uma vista lateral e são semelhantes aos B1, fazendo também uma aproximação seguida de um afastamento. Por sua vez, os caminhos B3, sob uma vista superior inclinada, realizam aproximações, translações perpendiculares à direção de visão e afastamentos curtos. Vale observar que estes caminhos foram traçados de forma semelhante em todos casos de teste, ao contrário dos caminhos L1, que efetuam passeios livres pelos reservatórios.

Para efetuar os testes, cada caminho foi percorrido várias vezes com diferentes parametrizações do erro projetado, sempre utilizando a resolução de tela de 1000x800 *pixels*. Em vez de manter o tempo de cada percurso constante, interpolando as posições da câmera ao longo do caminho de forma que a velocidade seja sempre a mesma, optou-se por fixar o número de posições da câmera e deixar que o tempo gasto no percurso varie. Isto justifica o campo “número de quadros” da Tabela 8.

**Tabela 8 – Caminhos.** Cada linha da tabela representa um caminho distinto. Existem quatro caminhos para cada malha de teste classificados de acordo com seu tipo.

	Tipo	Quantidade de quadros	Marcas em destaque
A_bc	B1	848	A e B
	B2	3080	A e B
	B3	1253	
	L1	4787	
B_cp	B1	2590	A e B
	B2	2810	A e B
	B3	1757	
	L1	3905	
I_bc	B1	3112	A e B
	B2	2772	A e B
	B3	1388	
	L1	3177	
D_cp	B1	808	A e B
	B2	2481	A e B
	B3	1552	
	L1	3469	
E_bc	B1	664	A e B
	B2	1892	A e B
	B3	500	
	L1	1429	
F_cp	B1	1369	A e B
	B2	826	A e B
	B3	779	
	L1	2329	

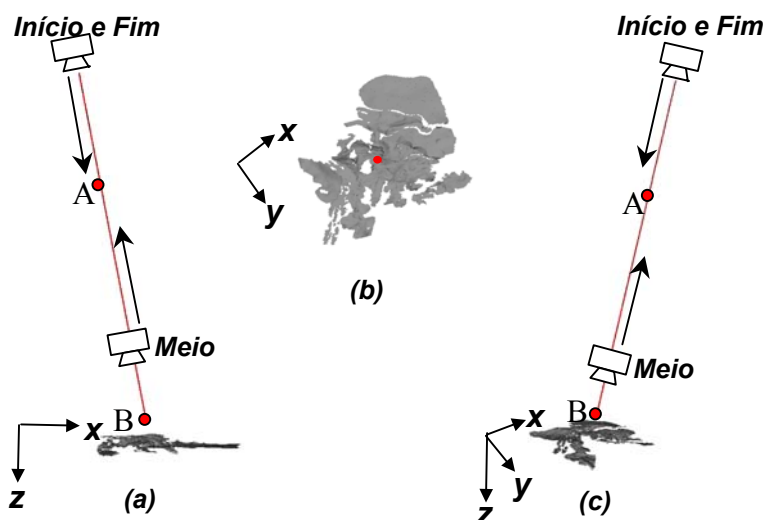


Figura 6.7 – Caminho B1 da malha F\_cp. Em (a) e (c), vistas perpendiculares à direção principal da câmera. Em (b), a vista superior do reservatório.

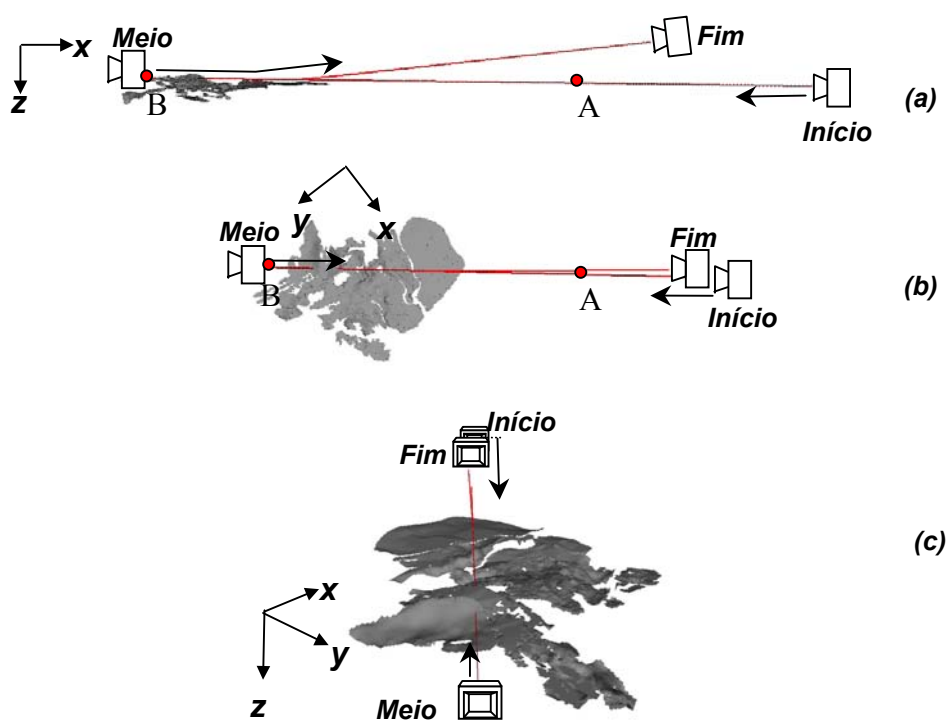
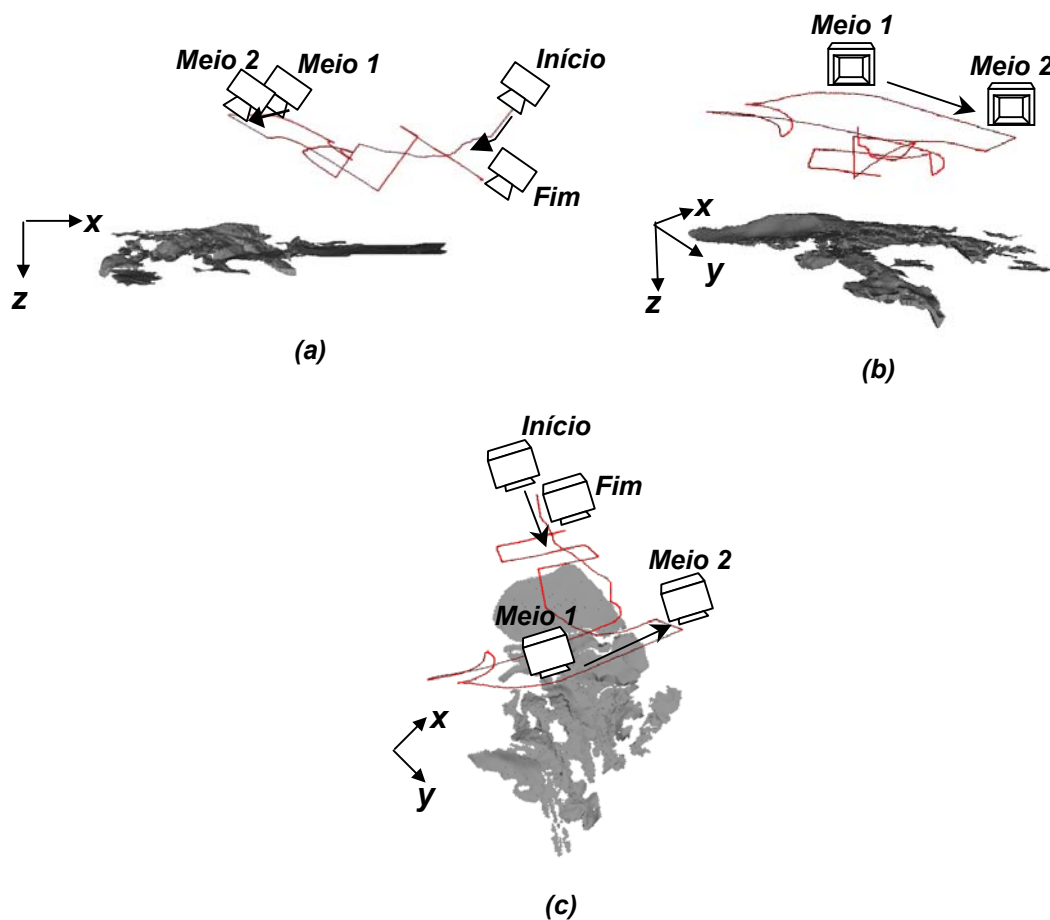


Figura 6.8 – Caminho B2. Em (a), a vista lateral à direção principal da câmera. Em (b), a vista superior do reservatório e, em (c), a uma vista quase frontal à direção de deslocamento da câmera.



**Figura 6.9 – Caminho B3.** Em (a), uma vista perpendicular à câmera, destacando o início, o meio e o fim do percurso. Em (b), uma vista quase frontal à câmera, destacando-a em dois pontos intermediários do percurso. Em (c), a vista superior do reservatório, destacando a câmera no início, no meio e no final do percurso.

Com o intuito de tornar a análise mais clara, os caminhos serão analisados separadamente.

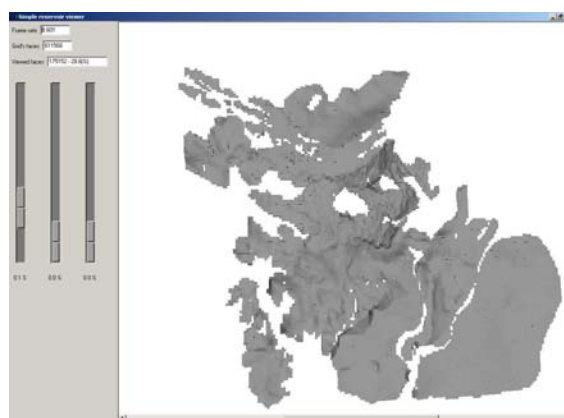
### ***Caminhos B1 e B2***

Estes caminhos foram traçados especificamente para analisar (a) a variação da redução de complexidade à medida que o reservatório ocupa uma área gradualmente



maior na tela e (b) a influência conjunta do descarte e da projeção nessa variação. Por isso, justifica-se o traçado controlado e simples dos dois percursos.

Foram destacados dois pontos de referência: A e B. Até o ponto A, a redução da complexidade é determinada somente pela projeção do erro. À medida que a câmera se aproxima do reservatório, sua área na imagem vai aumentando até que, exatamente em A, o reservatório ocupa o maior espaço na imagem sem que ocorra nenhum descarte (Figura 6.10). A partir daí, entre os pontos A e B, o comportamento é determinado tanto pela projeção do erro como pelo descarte, o primeiro requisitando o aumento das células e segundo eliminando-as. Do ponto B em diante, a câmera faz o percurso contrário, afastando-se.

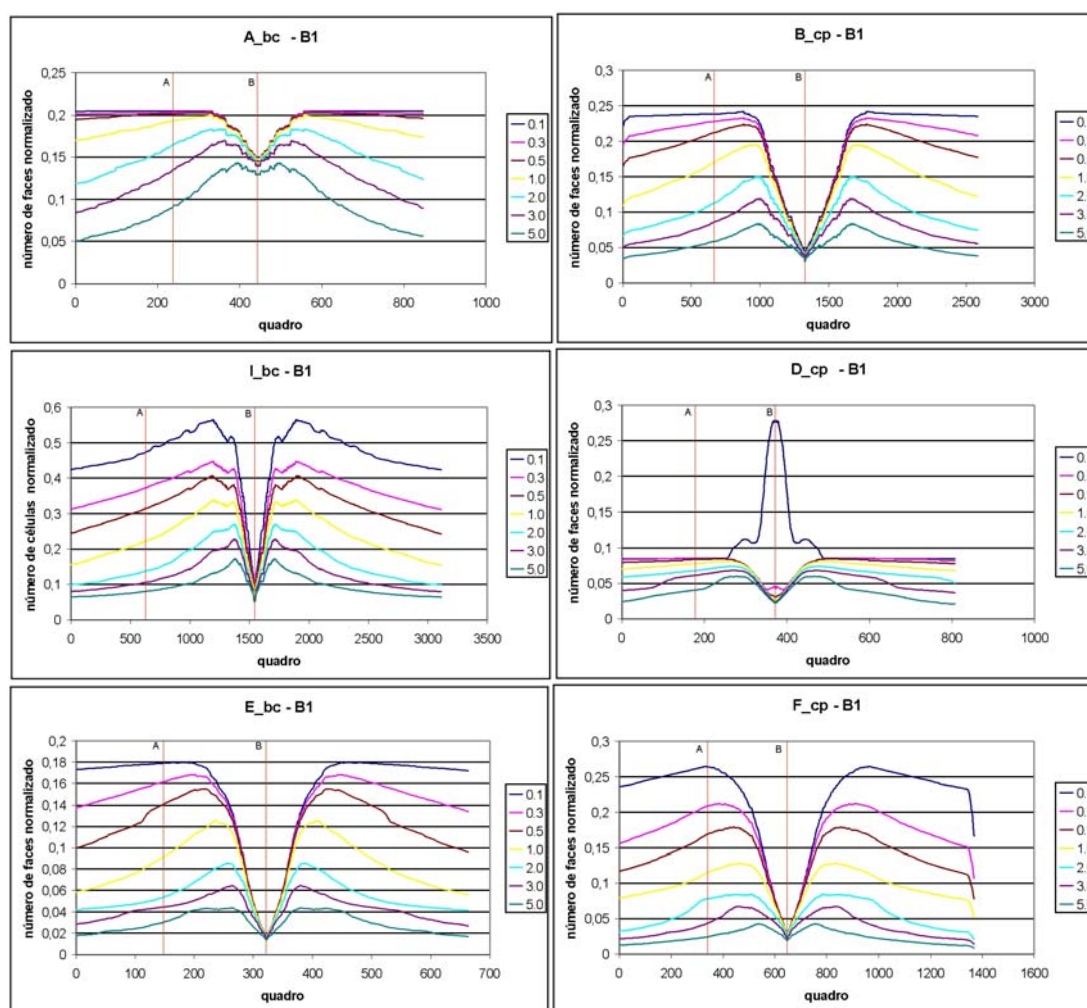


**Figura 6.10 – Ampliação máxima. Imagem gerada no ponto A no passeio realizado sobre  $F_{cp}$  com erro de 0,1 *pixel*. O reservatório está quase totalmente ajustado ao tamanho da tela.**

O Conjunto de Gráficos 1 refere-se aos caminhos B1 e mostram a quantidade de células que estão sendo desenhadas a cada quadro. Cada curva representa uma parametrização do erro, em *pixels*, considerado no percurso. Os piores resultados ocorrem na malha  $I_{bc}$ , cuja complexidade atingiu aproximadamente 55% para uma parametrização de 0,1 *pixel*. Nos demais casos, percebem-se resultados mais próximos, sendo que, coerentemente com a seção anterior, quanto maior o número de células, maiores são as taxas de redução alcançadas.

Na região do gráfico entre os pontos A e B, percebe-se a compensação entre o refinamento e o descarte. Na parte crescente da curva, o refinamento é dominante,

mas, à medida que a região que está dentro do volume de visão vai se aproximando de sua resolução máxima, o descarte vai se tornando dominante até que a curva passa a ser decrescente. Há apenas uma exceção<sup>13</sup> a este comportamento: a curva 0.1 do gráfico D\_cp do Conjunto de Gráficos 1. Nela, o refinamento é dominante em quase toda a região entre A e B.

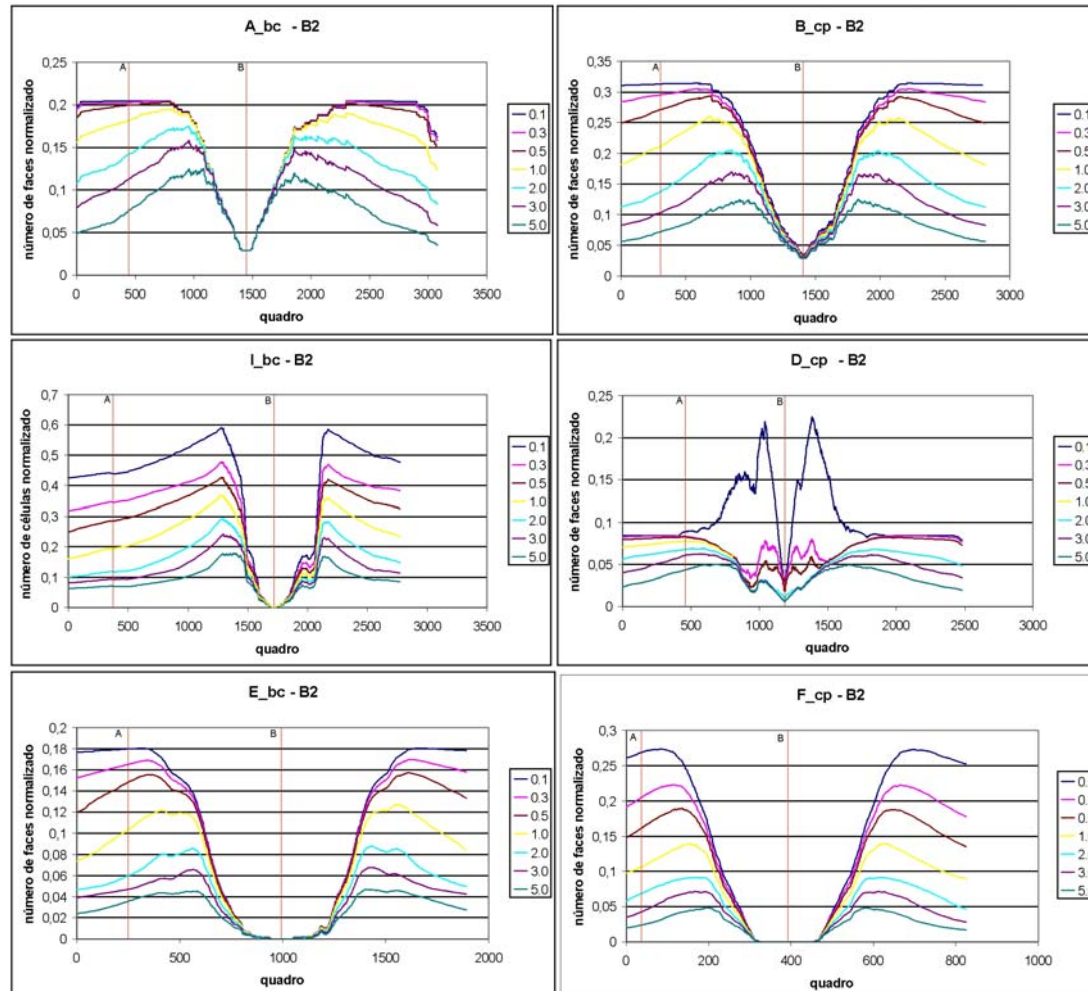


**Conjunto de Gráficos 1 – Variação do número de células nas aproximações ao longo dos quadros (B1). Cada gráfico representa o caminho B1 em um reservatório diferente.**

Uma vez que os caminhos B2 são semelhantes aos B1, mudando-se apenas a direção de deslocamento da câmera em relação à malha, seus resultados, que estão ilustrados

<sup>13</sup> A exceção existente na malha D\_cp pode ser explicada através do Gráfico 3, que representa o crescimento do erro geométrico nas aproximações. Nele, percebe-se a formação de uma espécie de patamar na malha D\_cp. O que ocorre na curva 0.1 dos caminhos B1 e B2 é a necessidade de ultrapassar esse patamar em algum momento durante a aproximação, aumentando abruptamente o número de células necessário para atender ao limite máximo de erro permitido.

no Conjunto de Gráficos 2, são a contra-prova dos resultados obtidos em B1. Analisando-os, percebe-se que realmente os comportamentos são parecidos, oferecendo mais confiabilidade aos resultados obtidos.



**Conjunto de Gráficos 2 - Variação do número de células nas aproximações ao longo dos quadros (B2). Cada gráfico representa o caminho B2 em um reservatório diferente.**

### ***Caminhos B3***

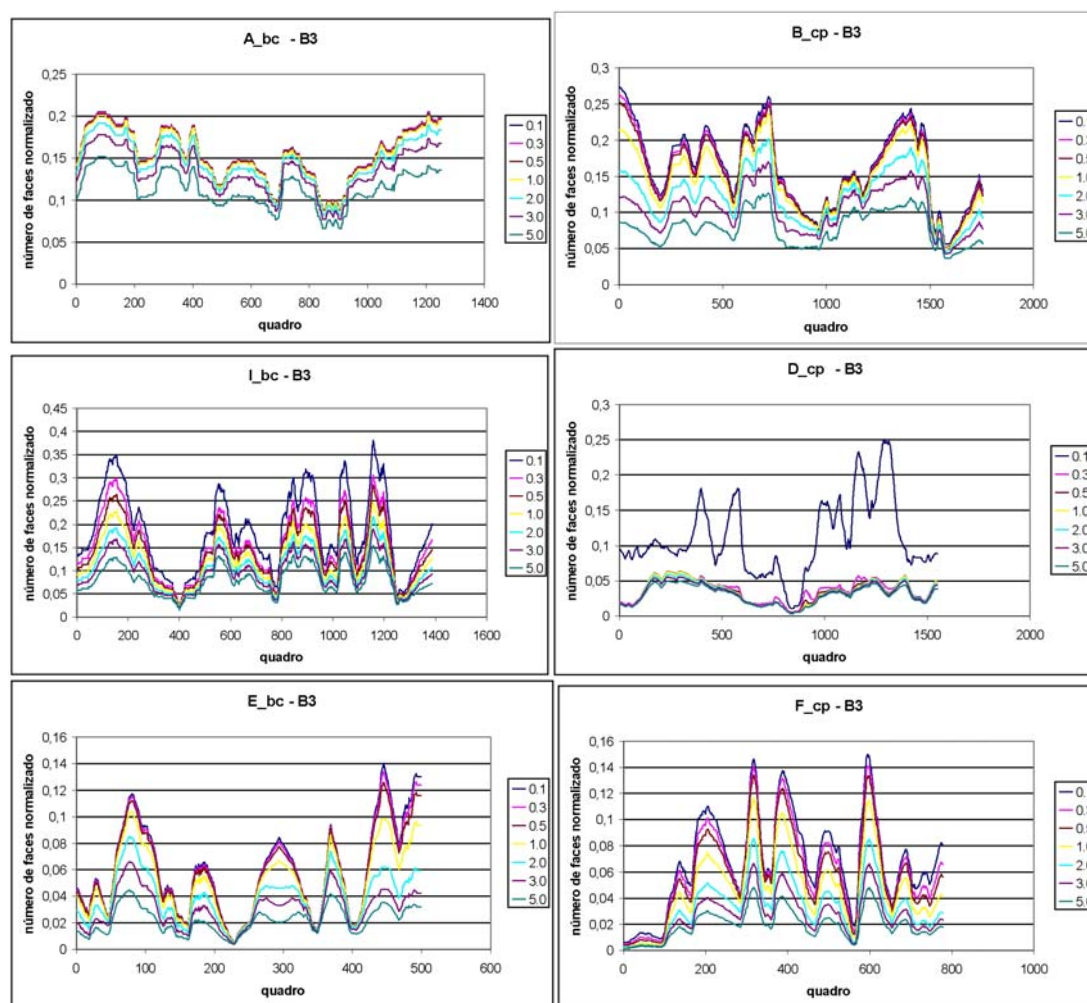
Os caminhos B3 foram projetados para possibilitar a avaliação do descarte como fator determinante para a redução da complexidade. Procurou-se traçá-los de forma que, em grande parte do percurso, apenas uma pequena região do reservatório estivesse visível.

O Conjunto de Gráficos 3 representa os resultados obtidos para os caminhos B3. Os pontos de mínimo das curvas correspondem a pontos nos quais apenas uma pequena

parte do reservatório está dentro do volume de visão. Esses pontos são nítidos em todos os casos de testes, inclusive no I\_bc. Percebe-se que as curvas se aproximam nesses pontos, indicando que o descarte, como é de sua natureza, independe da parametrização de erro considerada.

Apesar da influência do descarte ser mais notável na malha I\_bc (no sentido de que a complexidade da malha cai de aproximadamente 40% para menos de 5%), é neste caso de teste que obtêm-se os piores resultados de redução, o que ilustra o fato conhecido de que a técnica de descarte sozinha não garante baixas taxas médias de redução.

Mais uma vez percebe-se que as maiores malhas apresentam resultados melhores.

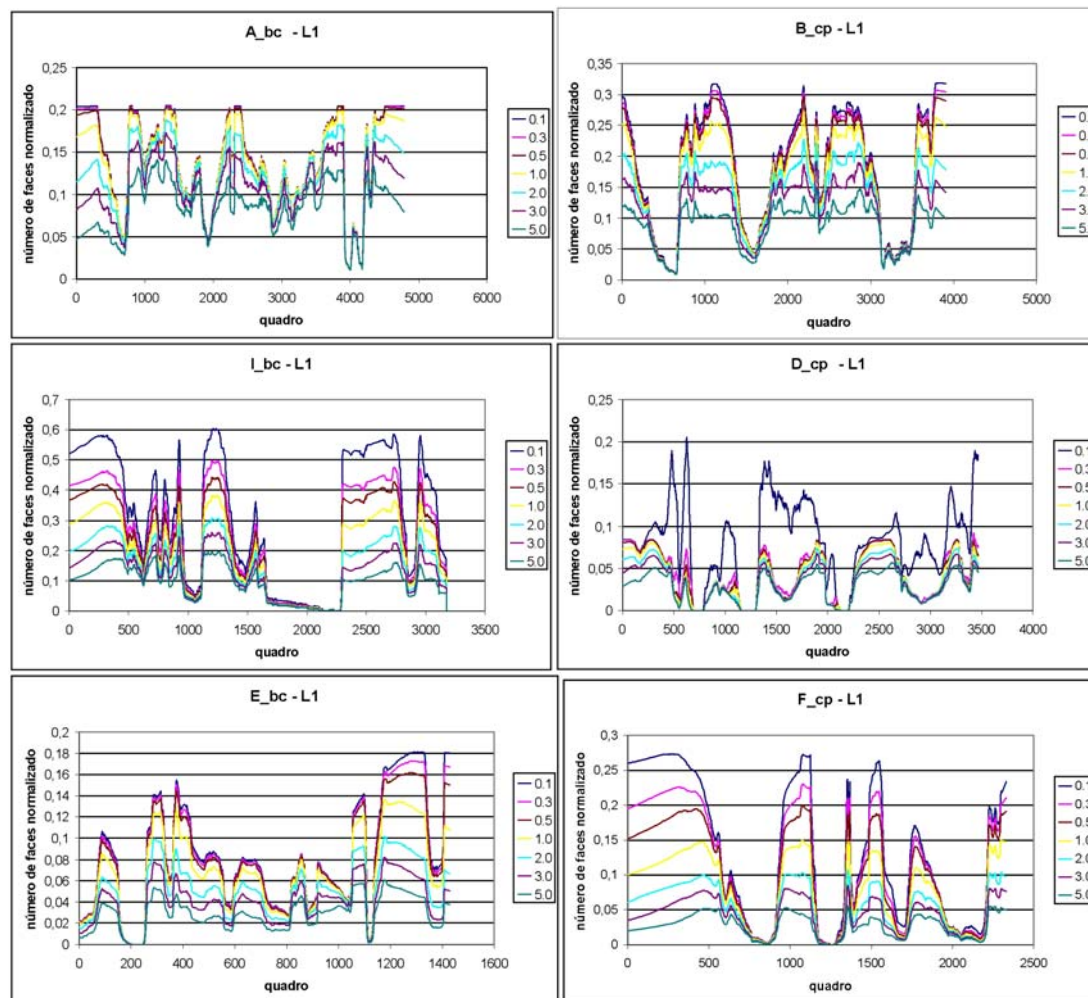


**Conjunto de Gráficos 3 - Variação do número de células nas aproximações ao longo dos quadros (B3). Cada gráfico representa o caminho B3 em um reservatório diferente.**



### ***Caminhos L1***

Os caminhos L1 exploram ângulos de visão variados sob os quais os reservatórios podem ser observados durante uma inspeção real, objetivando detectar os piores casos de redução de complexidade. Mais uma vez, o pior caso é o I\_bc, que chega a ser representado por 60% de sua complexidade quando a parametrização é de 0,1 *pixel*. Nos demais, a complexidade máxima alcançada é limitada em 32% para a mesma parametrização de erro.



**Conjunto de Gráficos 4 – Variação do número de células nas aproximações ao longo dos quadros (L1). Cada gráfico representa o caminho L1 em um reservatório diferente.**

### ***Conclusões Parciais***

Considerando a resolução de tela utilizada nos testes (1000x800), pode-se assumir que as parametrizações de 0,1 *pixel* e 0,5 *pixel* estão associadas a um baixo erro de

similaridade de aparência (ver Figura 6.6). Assim, as correspondentes curvas 0.1 e 0.5 nos gráficos caracterizam os valores de complexidade alcançados pelo MMR sem que praticamente nenhum erro visual seja cometido. As curvas 0.5 possuem seu pior caso na malha I\_bc: em torno de 45% de complexidade (caminhos B2 e L1). Nas demais malhas, o pior caso foi em torno de 30%. E, especificamente, nas três malhas mais densas, o pior caso foi de 20%.

Nos caminhos B1 e B2, o valor de complexidade alcançado no ponto A é importante, pois este ponto corresponde à posição do percurso que permite a visualização da malha por inteiro e com ampliação (*zoom*) máxima, o que, na visualização, é uma situação típica de manipulação. Para as curvas 0.5, o pior caso foi de aproximadamente 40% na malha I\_bc. Nas demais, o valor alcançado foi em torno de 27%, sendo que, particularmente nas três malhas mais densas, o valor foi de 17%.

Os valores obtidos com a malha I\_bc são importantes para a averiguação do comportamento do MMR proposto sob condições desfavoráveis; entretanto, eles não representam bem a potencialidade do MMR na prática usual. Assim, os mais representativos são os valores encontrados nas demais malhas. A Tabela 9 e a Tabela 10 reportam os valores representativos obtidos, separando as malhas quanto à densidade de células:

- baixa densidade: A\_bc e B\_cp;
- alta densidade: D\_cp, E\_bc e F\_cp.

Um fato importante a ser observado é que, na Seção 6.3.1, são obtidas malhas de boa qualidade geométrica com complexidade menor do que 30% do número original de células, sendo esperado que, a partir dessas malhas, fossem geradas imagens com boa similaridade de aparência. Essa expectativa foi confirmada através dos resultados aqui obtidos e, de certa forma, isso valida o erro geométrico utilizado pelo MMR.

Finalizando, a análise aqui realizada foi bastante conservadora e não explorou os resultados obtidos nas curvas que representam erros projetados maiores do que 1,0 *pixel*. Entretanto, as malhas geradas com esses parâmetros de erro podem ser úteis em algum caso específico de visualização que não exija alta similaridade de aparência.

**Tabela 9 - Relatório com as complexidades alcançadas nos piores casos, levando em consideração todos os caminhos percorridos.**

	Complexidade aproximada nos piores casos	Parametrização de erro utilizada
Baixa densidade	30%	0,5
Alta densidade	20%	0,5
Baixa densidade	25%	1,0
Alta densidade	15%	1,0

**Tabela 10 - Relatório com as complexidades alcançadas nos piores casos, levando em consideração apenas os valores no ponto A dos caminhos B1 e B2.**

	Complexidade aproximada nos piores casos	Parametrização de erro utilizada
Baixa densidade	27%	0,5
Alta densidade	17%	0,5
Baixa densidade	17%	1,0
Alta densidade	12%	1,0

### 6.3.3 *Taxas de Interatividade*

Para completar a avaliação feita na seção anterior, nesta seção será analisado o desempenho do algoritmo de extração dependente da câmera. Através dessa medição, pode-se avaliar realmente o ganho obtido com a utilização do MMR, verificando se a redução de complexidade das aproximações é suficiente e se os cálculos realizados para efetuar essa redução – cálculo de descarte e de projeção de erro, além do caminharmento pela hierarquia – são compensatórios.

A análise é feita através da medida **quadros por segundo (qps)**, que foi calculada instantaneamente através da fórmula:

$$Qps(q) = \frac{1}{\frac{1}{10} \sum_{i=0}^9 T(q-i)};$$

onde  $q$  representa um quadro e  $T(q)$  é o tempo gasto, em segundos, para processá-lo.

A medida qps é altamente dependente da plataforma de *hardware* utilizada, principalmente de seu processador gráfico. Assim, foram utilizadas duas plataformas diferentes nos testes, a III e a IV, e a implementação fez uso dos recursos de seus processadores gráficos. Dessa forma, parte do processamento está sendo feito pela CPU, enquanto outra parte está sendo feita pela **GPU** (Unidade de Processamento Gráfico). Por isso, é necessário marcar o tempo de “relógio de parede” gasto para processar um quadro e tomar o cuidado de evitar que processos externos concorram com a aplicação de visualização.

Para realizar os testes, foram escolhidas as malhas B\_cp e F\_cp, uma de baixa e outra de alta densidade de células, respectivamente. Os testes consistem em percorrer o caminho L1, na resolução de tela 1000x800, com quatro parâmetros diferentes de erro projetado (0,1, 0,5, 1,0 e 2,0 *pixels*), anotando a taxa de qps obtida. Para facilitar a análise, o Gráfico 4 e o Gráfico 6 mostram o número de células desenhadas a cada quadro durante o percurso realizado sobre as malhas B\_cp e F\_cp, respectivamente.

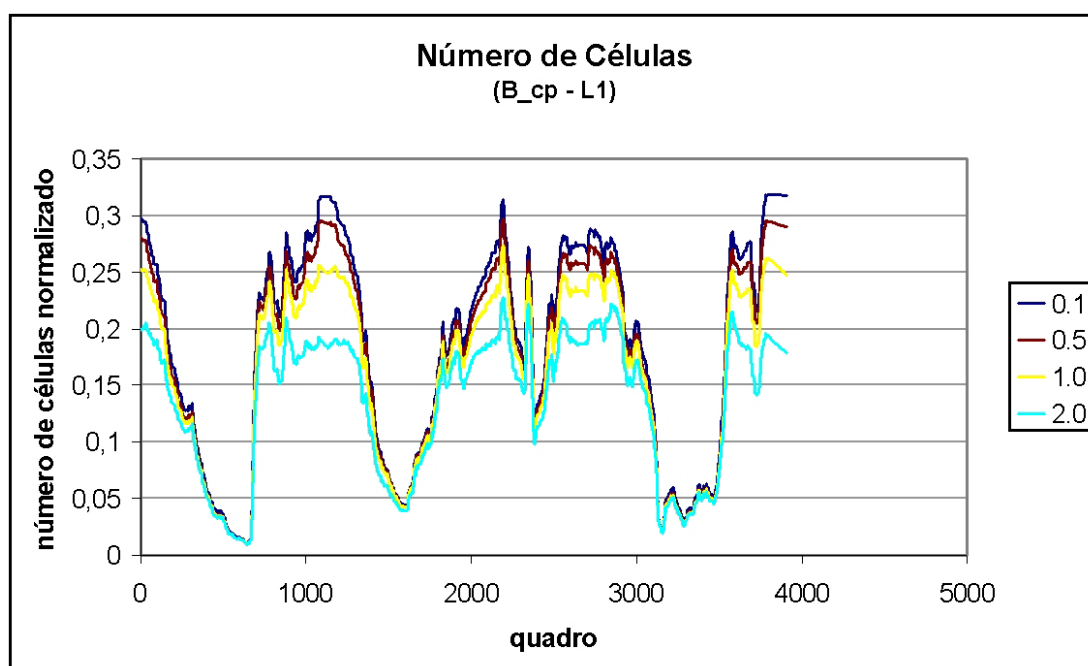
O Gráfico 5 e o Gráfico 7 representam os resultados obtidos na plataforma IV com valores truncados em 140 qps. Na malha B\_cp, a curva “sem MMR”, que representa o caminho percorrido sem a utilização de técnicas de aceleração, obteve uma média de 12,97 qps, mantendo-se praticamente constante no decorrer do percurso. Por outro lado, a utilização do MMR possibilitou uma taxa de interatividade com o valor mínimo de 33,85 qps na curva 0.1.

Na malha F\_cp, a curva “sem MMR” obteve uma média de 2,4 qps, enquanto os piores casos das curvas 0.1 e 0.5, que utilizam o MMR, foram de 7,58 qps e 11,59 qps, respectivamente.

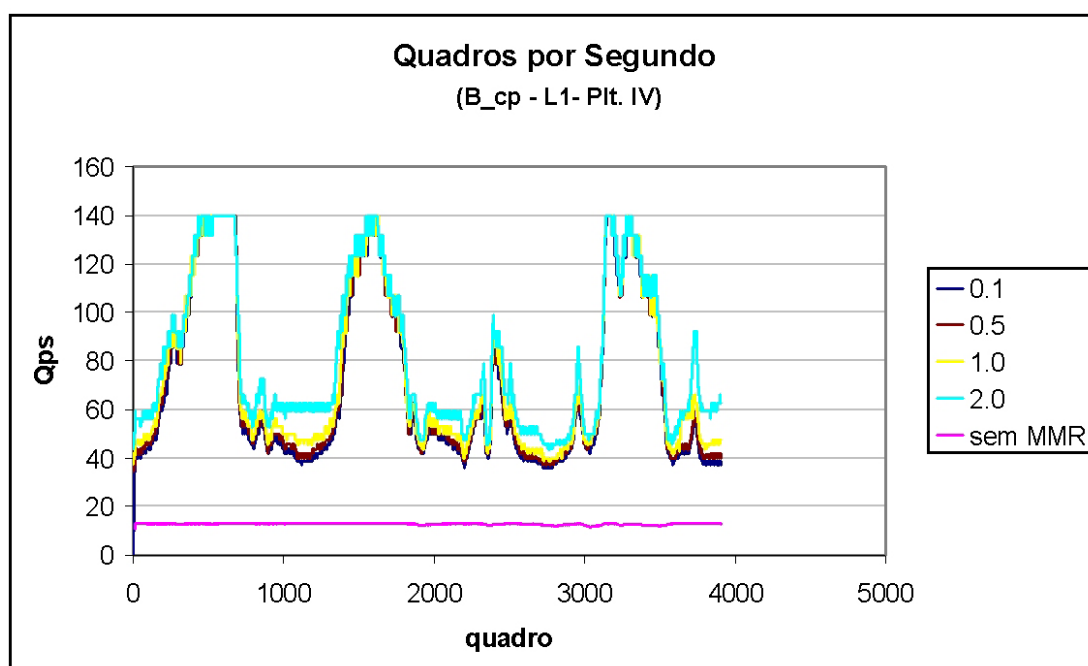
Como esperado, as curvas que representam as taxas de interatividade possuem o comportamento inverso às respectivas curvas que representam o número de células



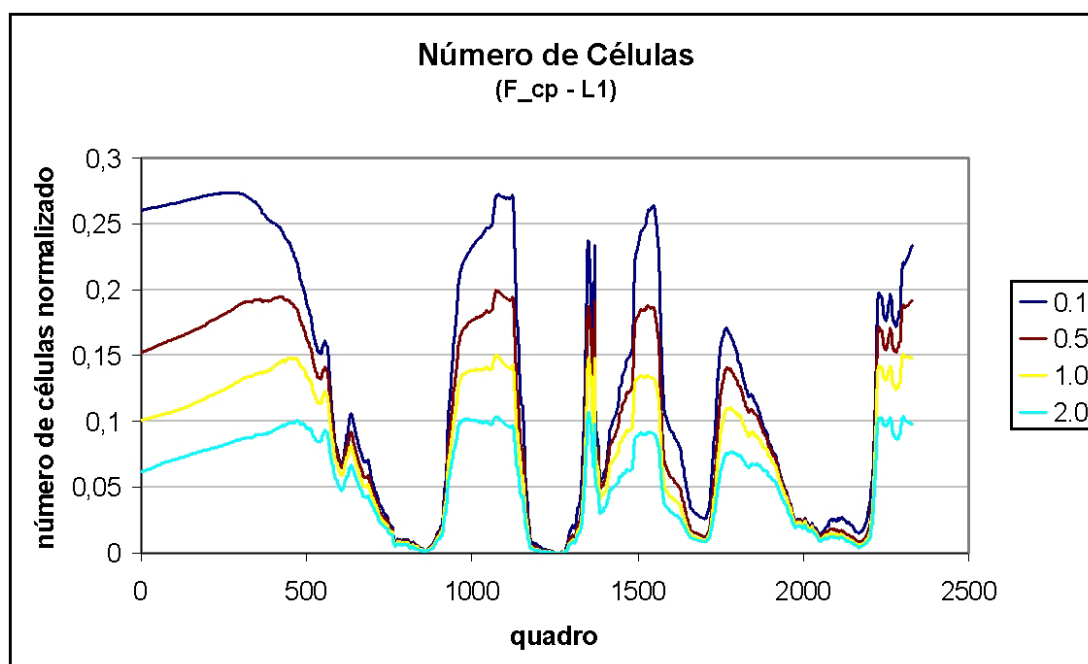
desenhadas.



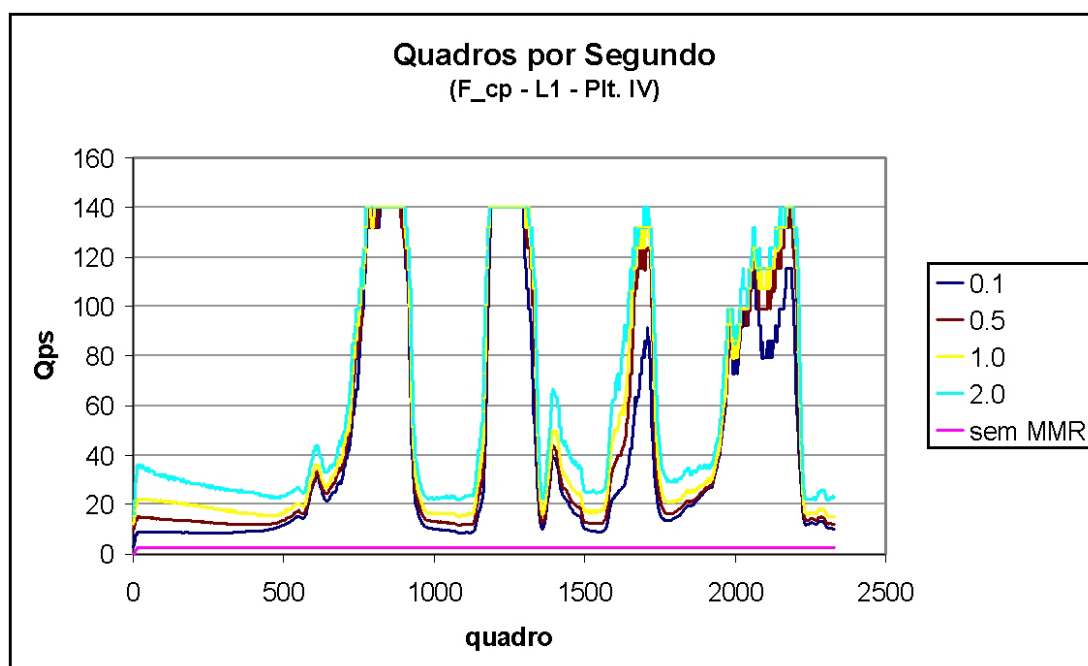
**Gráfico 4 – Número de células desenhadas a cada quadro ao longo do caminho L1 na malha B\_cp.**



**Gráfico 5 – Quadros por segundo ao longo do caminho L1 na malha B\_cp. A média da curva “sem MMR” foi de 12,4 qps.**



**Gráfico 6 - Número de células desenhadas a cada quadro ao longo do caminho L1 na malha F\_cp.**



**Gráfico 7 - Quadros por segundo ao longo do caminho L1 na malha F\_cp. A média da curva “sem MMR” foi de 2,4 qps.**

A Tabela 11 e a Tabela 12 apresentam as médias de quadros por segundo obtidas em cada curva das malhas B\_cp e F\_cp e também os piores casos de desempenho atingidos, oferecendo, assim, não só uma visão global do ganho de interatividade como também um limite inferior para esta.

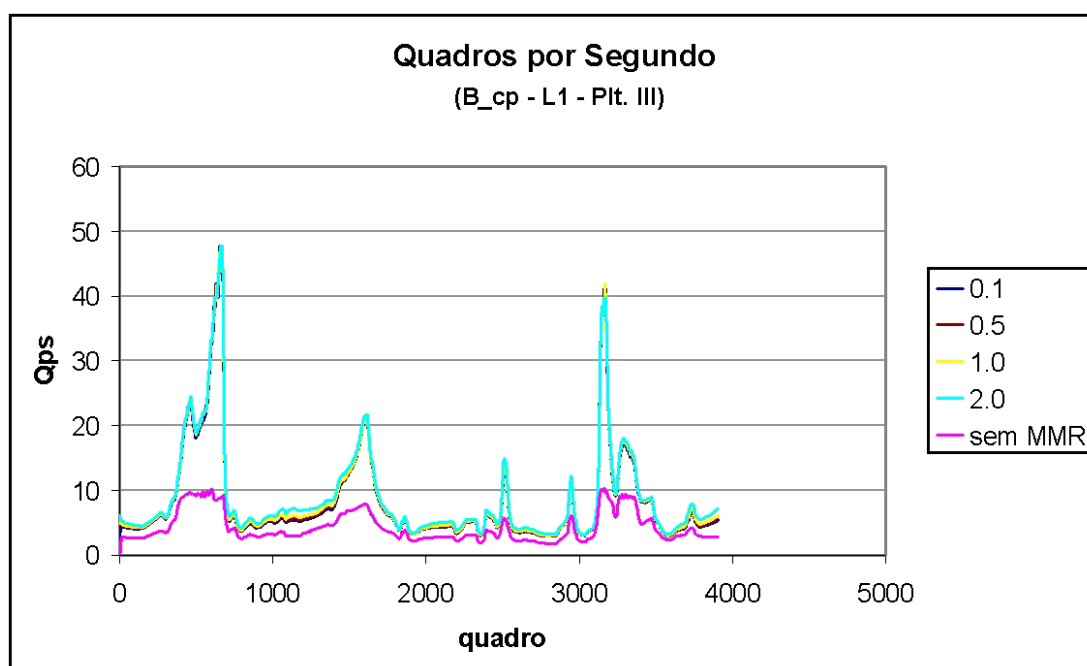
**Tabela 11 – Resultados, em quadros por segundo (qps), obtidos na malha B\_cp sob a plataforma IV. Na primeira linha, a média de quadros por segundo ao longo do caminho e, na segunda, a menor taxa de interatividade atingida.**

	Curvas				
	0.1	0.5	1.0	2.0	sem MMR
<b>Média (qps)</b>	57,77	60,23	63,9	72,48	12,83
<b>Pior caso (qps)</b>	33,85	35,86	38,36	43,64	10,7

**Tabela 12 – Resultados, em quadros por segundo (qps), obtidos na malha F\_cp sob a plataforma IV. Na primeira linha, a média de quadros por segundo ao longo do caminho e, na segunda, a menor taxa de interatividade atingida.**

	Curvas				
	0.1	0.5	1.0	2.0	sem MMR
<b>Média (qps)</b>	17,13	22,63	29,14	40,18	2,4
<b>Pior caso (qps)</b>	7,58	11,59	15,33	21,68	1,91

Enquanto os ganhos de interatividade foram razoáveis sob a plataforma IV, os obtidos sob a plataforma III e ilustrados no Gráfico 8 e na Tabela 13, para a malha B\_cp, foram menos expressivos, não conseguindo manter as taxas mínimas de interatividade (as taxas de interatividade começam a se tornar aceitáveis a partir de 6 qps [Tom99]).



**Gráfico 8 - Quadros por segundo ao longo do caminho L1 na malha F\_cp sob a plataforma III. A média da curva “sem MMR” foi de 3,39 qps.**

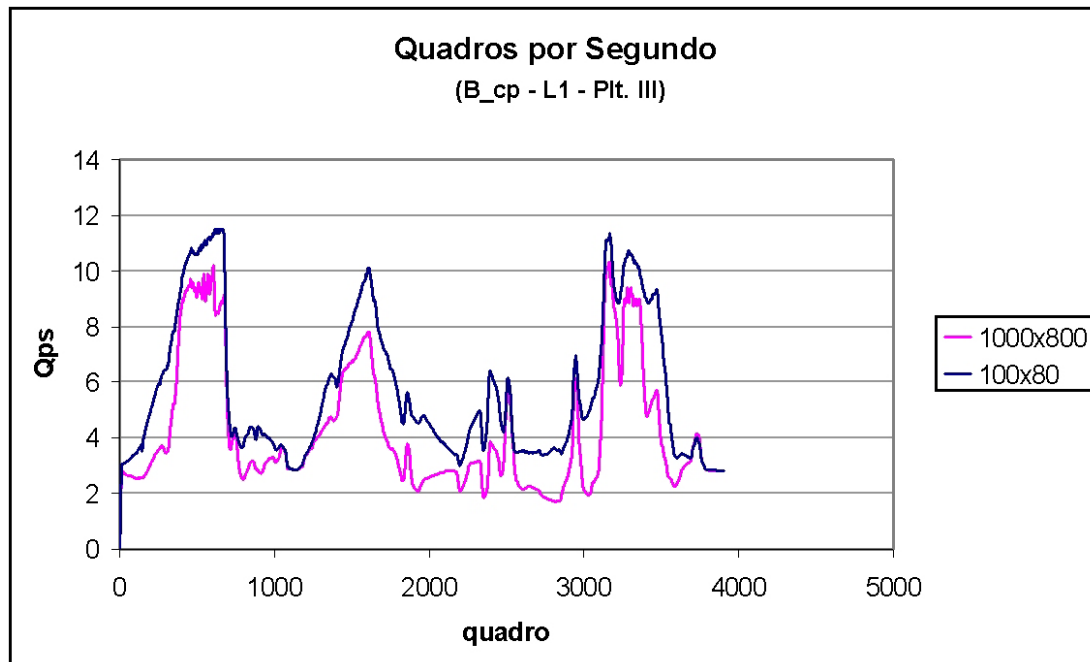
**Tabela 13 – Relatório das taxas de interatividade (B\_cp). Na primeira linha, as médias de quadros por segundo (qps) obtidas na malha B\_cp sob a plataforma III e resolução de tela de 1000x800 *pixels*. Na segunda linha, os piores casos de interatividade em cada curva.**

	Curvas				
	0.1	0.5	1.0	2.0	sem MMR
<b>Média (qps)</b>	5,47	5,57	5,73	6,15	3,39
<b>Pior caso (qps)</b>	2,84	2,88	2,97	3,09	1,69

O mau desempenho obtido pelo MMR na plataforma III se deve ao fato de que nem sempre, durante o percurso, o estágio de geometria foi o gargalo do canal de visualização<sup>14</sup>. Para analisar melhor essa questão, foi realizado um teste no qual o caminho L1 foi percorrido com a resolução de tela de 100x80 *pixels*, utilizando-se a malha B\_cp sem o uso de MR. O objetivo desse teste é averiguar a influência do estágio de rastreamento (*rasterization*) no canal de visualização, uma vez que ele é menos

<sup>14</sup> O canal de visualização é composto por três estágios conceituais: aplicação, geometria e rastreamento [Tomas99].

exigido com essa resolução de tela. O Gráfico 9 compara os resultados obtidos com as resoluções de 1000x800 *pixels* e 100x80 *pixels*. A alteração significativa de desempenho permite concluir que, em algumas regiões da curva 1000x800, o estágio de rastreo foi o gargalo do canal de visualização.

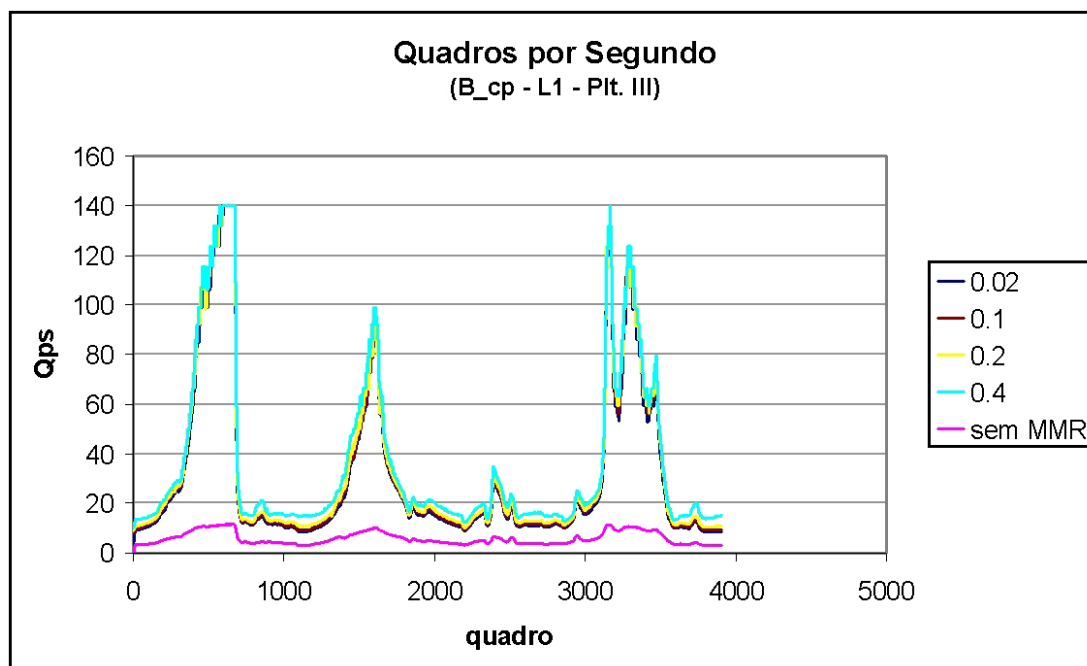


**Gráfico 9 - Comparação entre as taxas de quadros por segundo obtidas sem o uso de MR e com diferentes resoluções de tela: 1000x800 e 100x80 *pixels*. A curva 1000x800 apresentou um média de 4,24 qps e a curva 100x80 de 5,7 qps.**

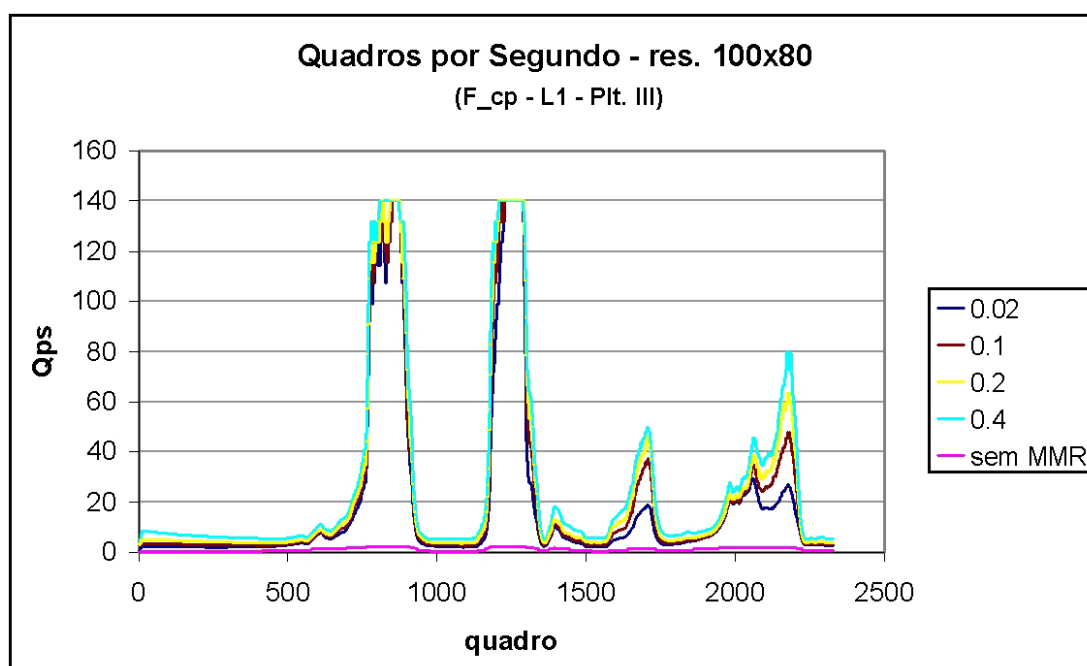
Para finalizar a análise, os caminhos L1 foram percorridos com a resolução de tela de 100x80 *pixels* utilizando as malhas B\_cp e F\_cp. Objetivando aliviar apenas o estágio de rastreo, os parâmetros de erro projetado utilizados anteriormente foram diminuídos na mesma proporção da diminuição das dimensões de tela, ou seja, em dez vezes. Com isso, a quantidade de células enviadas para o canal de visualização mantém-se praticamente idêntica e os resultados podem ser comparados aos anteriores. As antigas curvas 0.1 são comparadas às novas curvas 0.02, por sua vez, as curvas 0.5 se referem às 0.1, as curvas 1.0 às 0.2 e, finalmente, as curvas 2.0 às 0.4.

O Gráfico 10, o Gráfico 11, a Tabela 14 e a Tabela 15 mostram os novos resultados obtidos. Pode-se constatar a melhora sensível do desempenho do MMR na malha

B\_cp, na qual as taxas médias de interatividade passaram de aproximadamente 5,5 qps para mais de 15,0 qps. Na malha F\_cp, as médias de interatividade das curvas foram maiores do que 5,0 qps, com exceção da curva 0.02, mas os piores casos foram aquém do necessário para permitir interatividade em tempo real.



**Gráfico 10 – Taxas de quadros por segundo obtidas no caminho L1 da malha B\_cp sob a plataforma III e resolução de tela de 100x80. As curvas podem ser comparadas às dos gráficos anteriores.**



**Gráfico 11 – Taxas de quadros por segundo no caminho L1 da malha F\_cp utilizando a plataforma III e resolução de tela de 100x80.**

**Tabela 14 – Relatório das taxas de interatividade (plat. III). Na primeira linha, as médias das taxas de quadros por segundo (qps) obtidas na malha B\_cp sob a plataforma III e resolução de 100x80. Na segunda linha, os piores casos de interatividade.**

	Curvas				
	0.02	0.1	0.2	0.4	sem MMR
<b>Média (qps)</b>	16,18	17,08	18,54	22,15	4,78
<b>Pior caso (qps)</b>	7,43	8,93	9,67	12,32	2,32

**Tabela 15 – Relatório das taxas de interatividade (F\_cp). Tabela similar à anterior referente à malha F\_cp sob a plataforma III e resolução de 100x80. Na primeira linha, as médias das taxas de quadros por segundo (qps) e, na segunda, os piores casos de interatividade.**

	Curvas				
	0.02	0.1	0.2	0.4	sem MMR
<b>Média (qps)</b>	3,7	5,03	6,63	9,42	0,84
<b>Pior caso (qps)</b>	1,61	2,5	3,34	4,82	0,39

### ***Conclusões Parciais***

Os resultados obtidos demonstraram que, na plataforma IV, o MMR atingiu seu objetivo principal: o aumento considerável de interatividade, mantendo uma alta similaridade de aparência e possibilitando interatividade em tempo real. Entretanto, na plataforma III o MMR não obteve o mesmo êxito, embora ganhos tenham sido obtidos.

Os maus resultados obtidos na plataforma III devem-se (a) ao fato do gargalo do canal de visualização ter sido, muitas vezes, o estágio de rastreo e (b) à insuficiente redução de complexidade conseguida pelo MMR sem que a similaridade de aparência fosse penalizada.

O item (a) não é uma surpresa, pois é um fato conhecido que a técnica de MR atua diretamente sobre os estágios de aplicação e geometria, sobrecarregando o primeiro e aliviando o segundo. O ganho no estágio de rastreo é indireto e não se pode esperar que, sendo ele o gargalo, a utilização da técnica de MR obtenha ganhos de interatividade.

No caso da visualização de malhas de topologia *2-manifold*, ou de outras malhas que representem superfícies em geral, o fato anterior passa, muitas vezes, despercebido, pois a alteração da resolução da malha não afeta significativamente os cálculos do estágio de rastreo. A Figura 6.11 ilustra essa afirmação. Nela, percebe-se que os *pixels* a serem rastreados são exatamente os mesmos nas duas resoluções.

Entretanto, em malhas que representam um volume do espaço, como as malhas de simulação de reservatórios, a alteração da resolução altera a quantidade de *pixels* rastreados<sup>15</sup>, como ilustrado na Figura 6.12. Assim, percebe-se que a implementação eficiente do estágio de rastreo é crítica nesse caso, pois o número de *pixels* rastreados pode crescer muito com o aumento da resolução.

---

<sup>15</sup> Está sendo considerado o tipo de visualização utilizado neste trabalho: visualização da fronteira das células das malhas. A afirmação não é válida para outro tipo de visualização, como, por exemplo, a visualização direta.



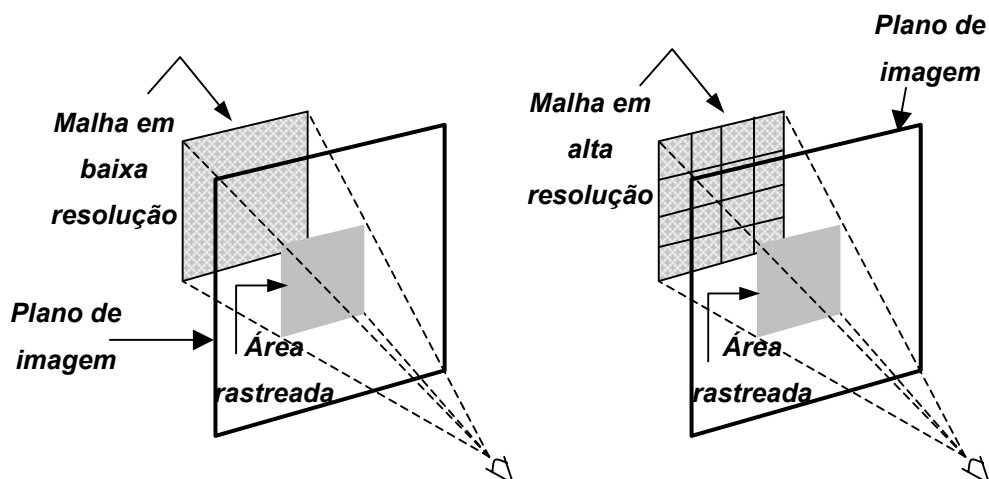


Figura 6.11 – Rastreio. As áreas rastreadas dos planos de imagem estão marcadas em cinza. Tanto em baixa resolução quanto em alta, a área é a mesma.

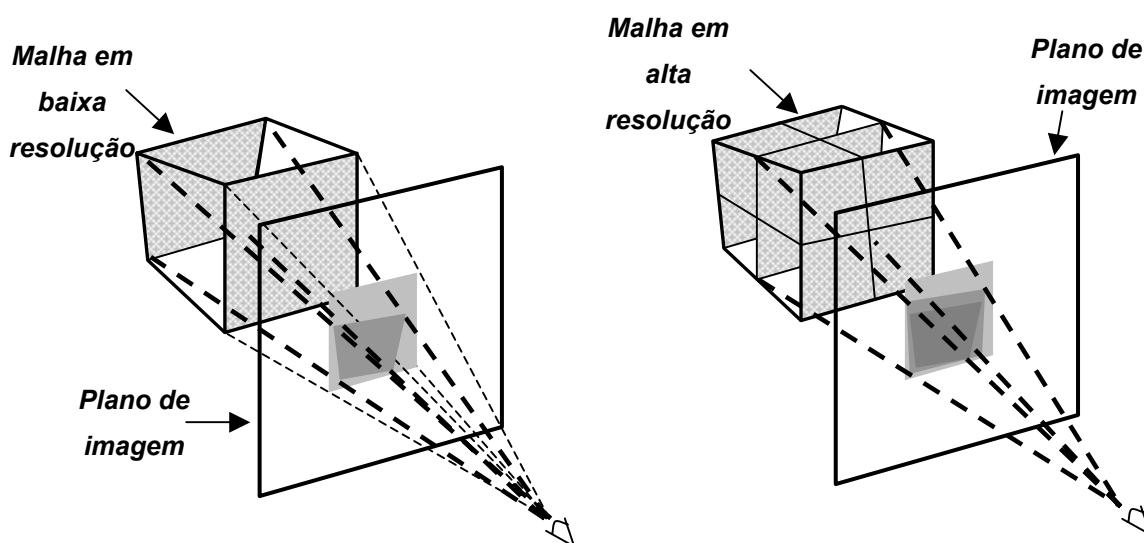


Figura 6.12 – Rastreio em volumes. As faces em destaque nas malhas são as mais significativas para o estágio de rastreamento. No plano de imagem, estão as projeções das faces em destaque. As variações de cor nas projeções indicam quantas vezes os pixels estão sendo rastreados: as cores mais escuras indicam que os pixels foram percorridos mais vezes.

O item (b) é uma limitação do MMR desenvolvido e pode ser notado nos resultados obtidos na malha F\_cp, através dos quais, mesmo com o estágio de rastreamento aliviado,

percebe-se que a interatividade em tempo real não foi obtida, ou seja, o número de polígonos enviados para o canal de visualização ainda foi elevado demais.

Vale observar que, sob o aspecto do número de polígonos enviados para o canal de visualização, as malhas que representam volumes exigem, naturalmente, maior poder computacional do que as malhas que representam superfícies, pois, nas primeiras, o número absoluto de células que ocupam o volume de visão tende a ser maior (supondo malhas com a mesma densidade de células). Assim, voltando ao caso das malhas de simulação, é de se esperar que a sua visualização exija um alto poder computacional.

## **6.4 Características de Visualização**

As características de visualização 5 e 6, que referem-se ao tempo gasto em pré-processamento e às taxas de interatividade, já foram analisadas nas seções anteriores. Cabe a esta seção verificar se as outras características de visualização foram atendidas adequadamente.

Para aumentar a confiabilidade dos resultados obtidos, a implementação do MMR aqui feita foi integrada ao aplicativo de visualização de malhas de simulação do fluxo de fluidos em reservatórios, o PosSim (Figura 6.13), um dos componentes do projeto GERESIM, o qual gerencia o processo global de simulação de reservatórios. Este aplicativo está sendo desenvolvido pelo Tecgraf (Grupo de Tecnologia em Computação Gráfica da PUC-Rio) em parceria com a Petrobras (Petróleo Brasileiro) e visa atender aos requisitos de visualização do problema considerado neste trabalho.

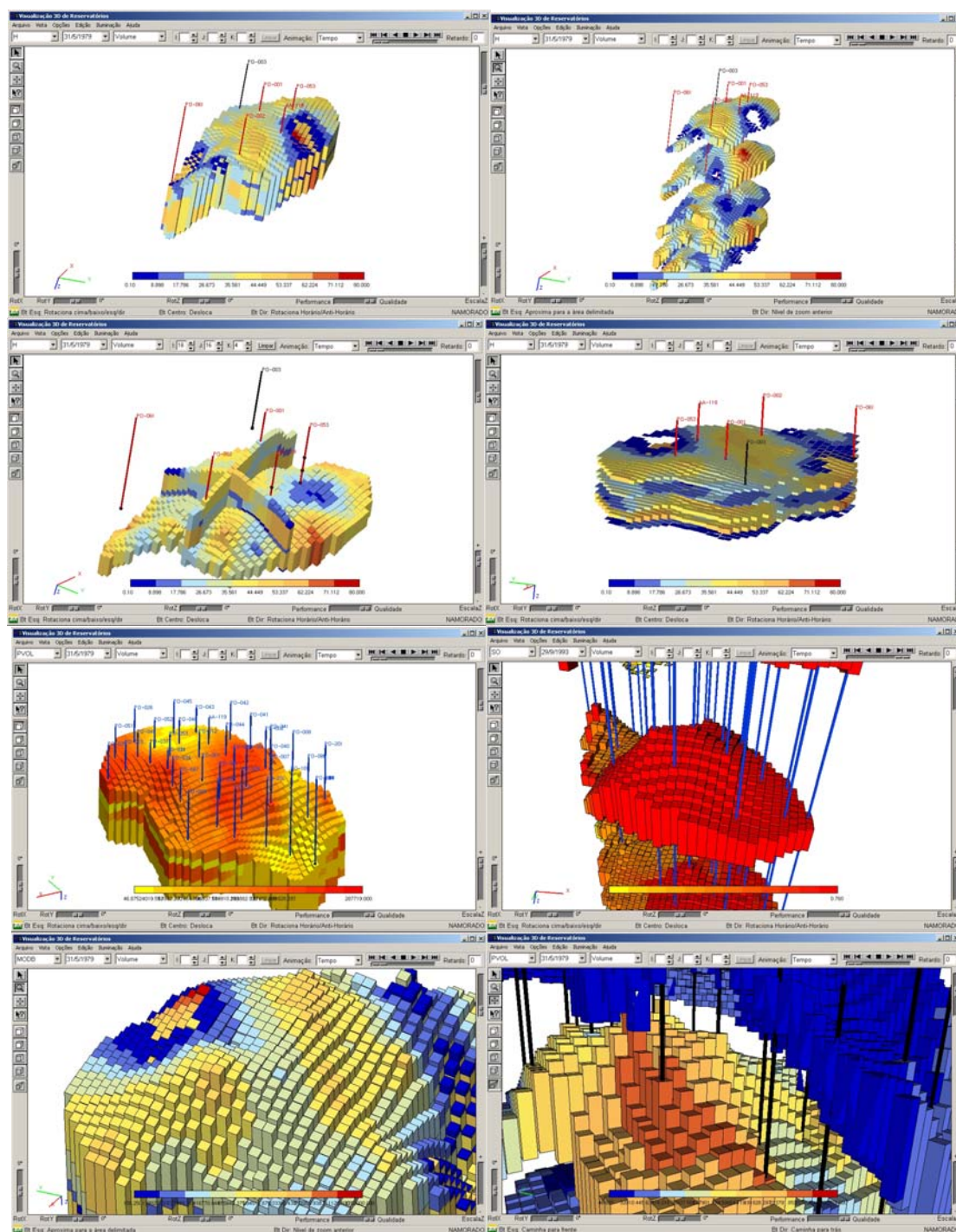


Figura 6.13 – PosSim (Visualizador 3D de reservatórios).

#### **6.4.1                    *Propriedades Escalares***

A inspeção das malhas de simulação em reservatórios é realizada através da associação à malha de uma propriedade escalar, a qual, normalmente, representa alguma grandeza física. Na visualização, isso é feito através de uma paleta de cores, que mapeia os valores escalares da propriedade a atributos de cores. Isso significa que a troca de uma propriedade física por outra está vinculada a uma possível troca dos atributos de cores das células da malha.

Visando gerar boas aproximações visuais, os algoritmos de extração de malhas consideram, também, o erro de propriedade em suas avaliações, tentando, com isso, adaptar a resolução da aproximação não só segundo a geometria, como também segundo as variações dos atributos de cores da malha.

Para atender às características de visualização 8 e 9, as quais requerem, no máximo, uma pequena perda de interatividade após a troca de uma propriedade ou durante sua animação temporal, o carregamento de uma propriedade na hierarquia de células hexaédricas é feito por demanda e consiste em percorrer a hierarquia, em tempo linear, calculando os valores e os erros de propriedade das macro-células através de cálculos já mencionados.

Uma vez que esses cálculos não alteram a estrutura da hierarquia, nada garante que a seqüência de colapsos que a gerou possua colapsos com baixos erros de propriedade. Ou seja, não há garantias de que o MMR seja capaz de gerar boas aproximações visuais.

Entretanto, a coerência espacial tipicamente existente em propriedades físicas aumenta a probabilidade dos colapsos efetuados durante a construção da estrutura de MR possuírem baixos erros de propriedade. Com base nisso, espera-se que o MMR seja capaz de representar suficientemente bem o reservatório na maioria das propriedades.

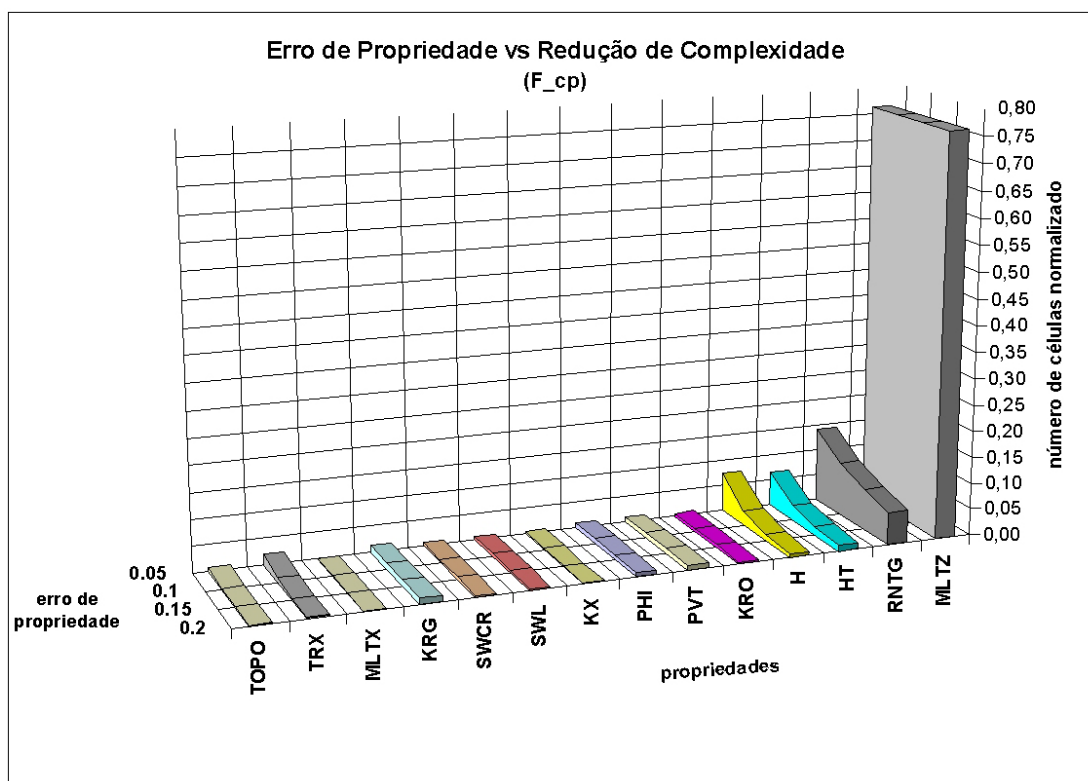
Esta seção apresenta os resultados obtidos com a estratégia de carregar as propriedades por demanda. Os testes realizados visaram analisar o quanto o erro

máximo de propriedade permitido interfere na capacidade do MMR de redução da complexidade das aproximações (Gráfico 12), desconsiderando o erro geométrico e a posição da câmera.

Os parâmetros utilizados para limitar o erro de propriedade permitido foram 0,05, 0,1, 0,15 e 0,2, números estes que estão normalizados pela diferença entre os valores máximo e mínimo da propriedade em toda a malha. Eles foram escolhidos em função da quantidade de intervalos existentes nas paletas de cores discretas geralmente utilizadas. Por exemplo, o parâmetro 0,05 é um bom valor a ser utilizado quando a paleta de cores possui 20 intervalos de cores, pois o erro visual de propriedade cometido não será maior do que um intervalo.

A análise foi feita na malha F\_cp, escolhida por possuir um conjunto bastante significativo de propriedades, o qual representa uma vasta gama de grandezas físicas diferentes. Apenas parte dos resultados é apresentada no Gráfico 12, através do qual percebe-se que, na maioria das propriedades, o MMR consegue gerar malhas adaptadas às variações de propriedade, utilizando poucas células.

Em algumas propriedades (H, HT, RNTG e MLTZ), entretanto, o MMR precisou aumentar muito a resolução das malhas para obedecer ao limite máximo de erro permitido. É válido ressaltar que a propriedade MLTZ, que obteve os piores resultados, não é uma grandeza física: ela representa um fator de multiplicação associado a uma grandeza.



**Gráfico 12 – Redução de complexidade nas aproximações em função do erro máximo de propriedade permitido. Os dados foram obtidos a partir do algoritmo de extração dependente do erro de propriedade. Os parâmetros de erro utilizados foram 0,05, 0,1, 0,15 e 0,2.**

### ***Conclusões Parciais***

Confirmando as expectativas, os resultados mostraram que a capacidade de redução de complexidade do MMR, em geral, não é afetada pela consideração do erro de propriedade. Entretanto, a incapacidade de obtenção de altas taxas de redução, em algumas propriedades, pode impedir que a interatividade adequada seja alcançada sempre.

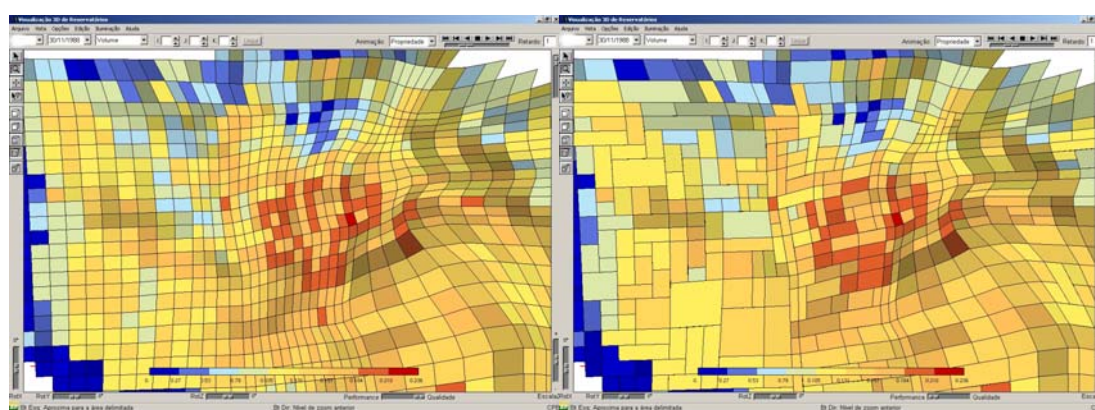
Com respeito a essa limitação, é válido ressaltar que, visando oferecer mais recursos ao PosSim, é possível modificar o algoritmo de construção da estrutura de MR, fazendo com que a operação de colapso considere o erro de propriedade em sua construção. O algoritmo modificado poderia, então, ser utilizado como uma opção ao usuário para visualizar as propriedades que obtiveram maus resultados com o primeiro algoritmo.

### 6.4.2 Visualização da Grade (Wireframe)

A característica de visualização 1 requer que o MMR seja capaz de desenhar o *wireframe* das células hexaédricas, de forma que uma grade de hexaedros seja visualizada.

A grade oferece ao usuário as informações de linhas, colunas e camadas, sendo interessante que a alteração da resolução da grade mantenha a capacidade de obtenção dessas informações. Por exemplo, a utilização de uma estrutura similar a uma *octree*, na qual os agrupamentos de oito células fossem baseados na vizinhança IJK da malha, responderia adequadamente ao problema de multi-resolução da grade, pois, a cada nível da árvore, teria-se uma nova grade, cujo intervalo entre linhas consecutivas representaria diferentes intervalos entre camadas, linhas e colunas na malha.

Entretanto, ao contrário dessa *octree*, a hierarquia de células hexaédricas desenvolvida neste trabalho não é capaz de gerar grades que atendam às necessidades de informação do usuário. A Figura 6.14 ilustra a visualização do *wireframe* das células através do MMR. Percebe-se que as linhas da grade de menor resolução, à direita, são interrompidas, dificultando a obtenção da informação desejada.

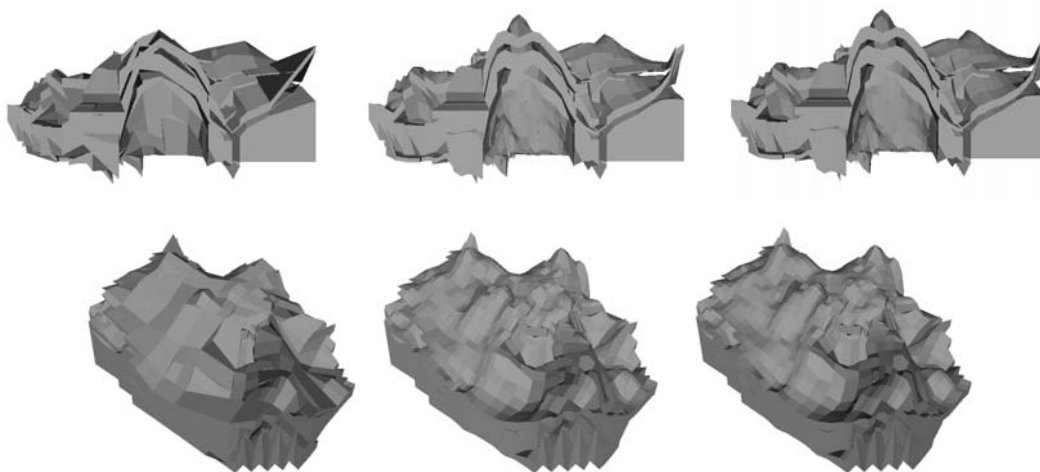


**Figura 6.14 – Visualização da grade.** Imagens geradas pelo MMR proposto nas quais nota-se a visualização do *wireframe* das células sob a vista superior, formando uma grade. À esquerda, a grade na maior resolução e, à direita, a grade em menor resolução.

### ***Conclusões Parciais***

A hierarquia de células hexaédricas é incapaz de gerar grades regulares porque a construção da hierarquia é feita por agrupamentos livres de células, sem a preocupação de obter grupos que, por nível da estrutura, sejam formados pelo mesmo número de células em I (ou em J ou em K), como acontece na *octree*.

É válido ressaltar que é a possibilidade de realizar agrupamentos livremente que permite à hierarquia de células gerar malhas bem adaptadas à geometria, ao contrário da *octree*. Esse fato foi comprovado, na prática, através da implementação da *octree* baseada no agrupamento de células vizinhas, oito a oito, que foi a primeira estratégia explorada neste trabalho. A Figura 6.15 compara os resultados visuais obtidos com a utilização das duas estruturas. As duas malhas possuem aproximadamente 21.700 polígonos.



**Figura 6.15 – Comparação visual entre os resultados obtidos com a estrutura *octree* e a hierarquia de células. As duas primeiras colunas contêm duas aproximações distintas da malha original, que é ilustrada na coluna da direita. Na coluna da esquerda, uma aproximação gerada pela estrutura *octree* e, na coluna do meio, uma aproximação gerada pela hierarquia de células proposta neste trabalho. Ambas aproximações possuem aproximadamente 21.700 polígonos ( $\approx 20\%$  da malha original).**



### 6.4.3 *Separação entre Camadas e Escala em Z*

As características de visualização 3 e 4 estão relacionadas a modificações na geometria da malha.

Particularmente, a característica 3 – a qual prevê uma possível escala em  $Z$  – é facilmente atendida pelo MMR aqui desenvolvido. A solução utilizada é bastante simples: no momento da renderização, multiplica-se o valor da escala que se quer realizar à geometria do reservatório, inclusive às *aabbs* associadas a cada nó, e ao erro geométrico armazenado pela estrutura de MR. Esse novo erro será uma majoração do erro que está sendo cometido com a escala em  $Z$ . O comportamento percebido é que, à medida que o usuário vai aumentando a escala em  $Z$  desejada, a resolução do reservatório tende a aumentar juntamente.

Uma solução alternativa, que não foi implementada, seria armazenar, na estrutura de MR, o vetor que representa o erro geométrico cometido, em vez da distância. Dessa forma, poderia-se multiplicar apenas a componente  $z$  do vetor para representar o erro após a efetuação de uma escala. Esta forma de variar o erro torna-o mais coerente com o erro que está sendo cometido com a escala em  $Z$ .

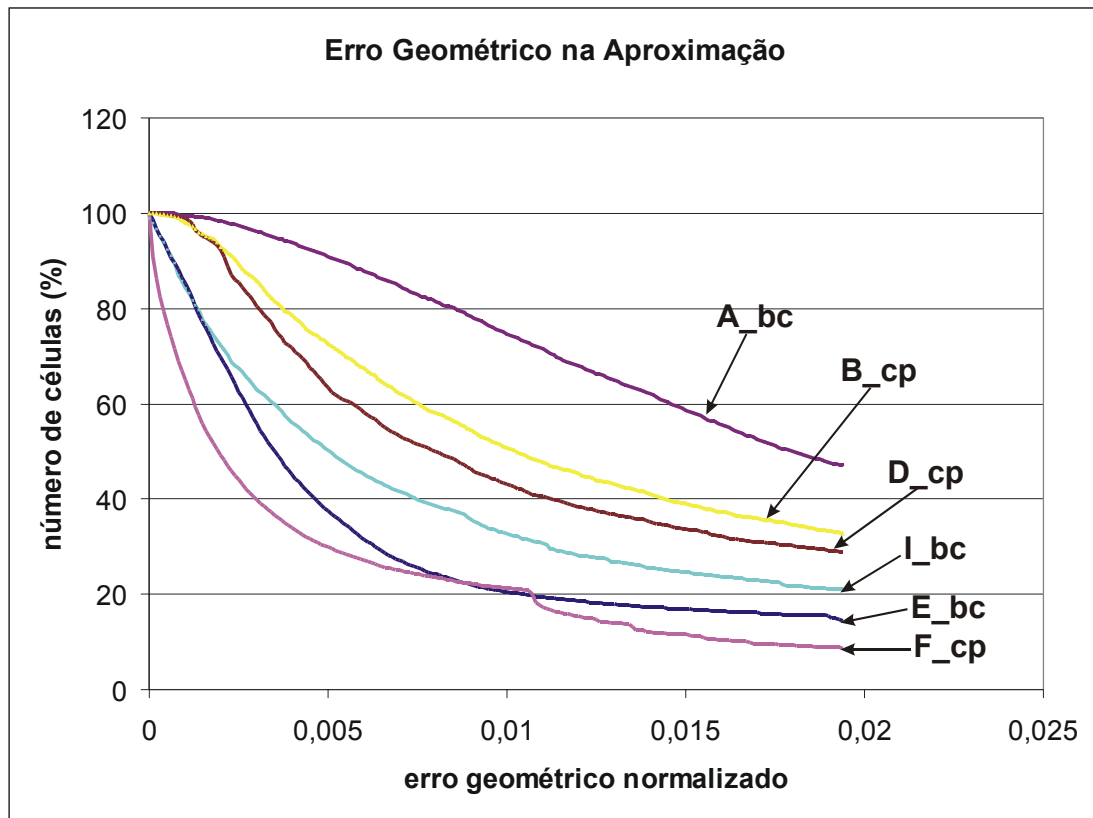
A característica 4, por sua vez, requer que as camadas do reservatório sejam visualizadas através de afastamentos, um para cada par de camadas vizinhas (Figura 1.3-c), fazendo com que a hierarquia de células só possa realizar colapsos entre células pertencentes à mesma camada.

Na prática, o MMR mostrou-se incapaz de reduzir a complexidade das aproximações quando as camadas são afastadas, impossibilitando o aumento das taxas de interatividade. Visando contornar esse problema, é utilizada, no PosSim, uma segunda hierarquia de células hexaédricas, obtida através de um algoritmo de construção modificado. A modificação realizada consiste em atribuir um custo infinito aos colapsos entre células de camadas distintas, fazendo com que eles sejam os últimos colapsos realizados durante a construção. Essa segunda hierarquia é capaz de reduzir com mais eficácia a complexidade das aproximações quando o afastamento entre camadas é efetuado.

### ***Conclusões Parciais***

Em relação à característica de visualização 3, a solução adotada obteve resultados satisfatórios, sendo que uma análise melhor se faz necessária apenas em relação à característica de visualização 4. Nesse caso, a incapacidade do MMR reduzir a complexidade das aproximações se deve ao fato de existirem poucos colapsos entre células de uma mesma camada na estrutura de MR. A causa disso é a superamostragem existente nas colunas em profundidade, já citada em outras seções, fazendo com que, durante a construção da hierarquia, os primeiros colapsos a serem efetuados, em sua maioria, sejam justamente colapsos em Z, que agrupam células de camadas distintas.

A utilização de uma segunda estrutura para contornar esse problema acarreta o aumento do custo de construção da estrutura de MR e do consumo de memória. É importante observar que a segunda estrutura não gera aproximações tão boas quanto a primeira e, devido a isso, a primeira estrutura ainda é necessária. O Gráfico 13 ilustra o erro geométrico inserido nas aproximações, utilizando a segunda estrutura de MR. Ele é semelhante ao Gráfico 3, sendo obtido através do algoritmo de extração dependente do erro geométrico, mostrando valores de erro normalizados entre 0% e 20% da diagonal da caixa envolvente da malha. Comparando-o com o Gráfico 3, percebem-se resultados sensivelmente piores e, principalmente, a ausência de redução drástica das aproximações sem inserção de erro, indicando que realmente os colapsos em Z são importantes.



**Gráfico 13 – Erro geométrico nas aproximações utilizando a segunda hierarquia de células hexáedricas.**

Aqui, faz-se necessária uma observação complementar em relação à visualização de propriedades escalares (Seção 6.4.1): os resultados ruins obtidos na propriedade MLTZ também são causados pela necessidade de evitar colapsos entre células de camadas distintas, uma vez que MLTZ apresentou uma variação brusca de valores ao longo de Z. Dessa forma, a segunda estrutura de MR também pode ser uma solução para a visualização de propriedades.