

8 Estudo de Caso

Este capítulo tem o objetivo de mostrar os resultados obtidos na construção da estrutura em multi-resolução adaptativa e no sistema de gerenciamento de memória utilizado no processo de visualização dos objetos gráficos.

A primeira seção faz uma comparação entre a construção da representação em multi-resolução utilizando o erro relativo e utilizando o erro acumulado. A análise é feita a partir de testes realizados em texturas com características diferentes. Essa análise inclui o estudo de gráficos e tabelas. Assim, é possível identificar as situações onde as técnicas apresentam bons resultados e situações onde não. No caso das situações onde os resultados não são satisfatórios, será mostrado porque ocorre o problema e serão mostrados os procedimentos para amenizá-lo ou resolvê-lo.

A segunda seção é dedicada ao estudo de desempenho do sistema de gerenciamento de memória. Esse estudo foi feito com o objetivo de mostrar a flexibilidade da hierarquia de armazenamento e testar o sistema de predição adaptativo propostos neste trabalho. A análise de desempenho foi realizada levando em consideração vários fatores como: a movimentação da câmera virtual, as dimensões das texturas dos ladrilhos e as configurações de hardware. Além disso, foi feita uma comparação de desempenho entre o nosso sistema de predição e o sistema de predição baseado em área de segurança.

8.1 Multi-Resolução Adaptativa

Como já foi discutido, a estrutura de multi-resolução adaptativa é construída a partir de operações de união, reamostragem, super-amostragem e comparação de ladrilhos. A operação de união é implementada como uma simples operação de cópia de imagens. As operações de super-amostragem e reamostragem das texturas foram implementadas com base no algoritmo de

Fant (veja [3]). A operação de comparação utiliza a métrica L^2 para calcular a diferença entre cada pixel das texturas.

A comparação é feita da seguinte forma: se a diferença global entre as cores da textura do ladrilho pai e as cores das texturas do ladrilho filho for menor que uma certa tolerância, então as texturas dos ladrilhos filhos são eliminadas. Aqui, a tolerância é dada em porcentagem. Essa porcentagem é calculada a partir da razão entre o somatório da distância euclidiana entre os pixels de cada imagem e o número total de pixels. Ou seja,

$$Tol = \frac{\sum_{i=1}^h \sum_{j=1}^w Dist(I_p(j, i), I_f(j, i))}{wh},$$

onde $I_f(j, i)$ é textura resultante da união das texturas dos ladrilhos filhos, $I_p(j, i)$ é a textura extrapolada do ladrilho pai e wh é o número total de pixels. O operador $Dist$ é definido como $Dist(I_p(j, i), I_f(j, i)) = \sqrt{(I_{fr}(j, i) - I_{pr}(j, i))^2 + (I_{fg}(j, i) - I_{pg}(j, i))^2 + (I_{fb}(j, i) - I_{pb}(j, i))^2}$, onde $I_r(j, i)$, $I_g(j, i)$ e $I_b(j, i)$ são as componentes de cores associadas a um pixel (j, i) . Os objetos gráficos foram processados para ter o número máximo de níveis de resolução permitido por cada tipo de textura. Em todos os exemplos, as dimensões das texturas dos ladrilhos são 128×128 pixels.

Os testes foram realizados com o objetivo de comparar os métodos de multi-resolução normal, de multi-resolução adaptativa com erro relativo e com erro acumulado. Para calcular o erro acumulado foi utilizado o erro máximo dos ladrilhos filhos (veja a Seção 4.2). Nos testes foram utilizados três objetos gráficos com texturas diferentes. Os dois primeiros são duas imagens de satélite (São José dos Campos e Pará) e o outro é um panorama virtual do Rio de Janeiro, como mostra a Figura 8.1.

Observe que as texturas têm características diferentes. A imagem do Rio de Janeiro está mais ou menos dividida em duas regiões: uma que representa o céu (baixas frequências) e a outra a cidade (altas frequências). A imagem do Para é constituída em sua maioria de florestas, desmatamentos e rios. E finalmente, a imagem de São J. dos Campos que ilustra a cidade com suas construções, avenidas e ruas. Nas imagens de satélites as regiões de baixas e altas frequências estão distribuídas de forma irregular. A Tabela 8.1 mostra o tamanho e as dimensões de cada textura.

A comparação das texturas originais com suas representações em multi-resolução é feita comparando a textura original com a textura reconstruída a partir da representação em multi-resolução. O método de comparação é o mesmo utilizado na comparação das texturas dos ladrilhos.

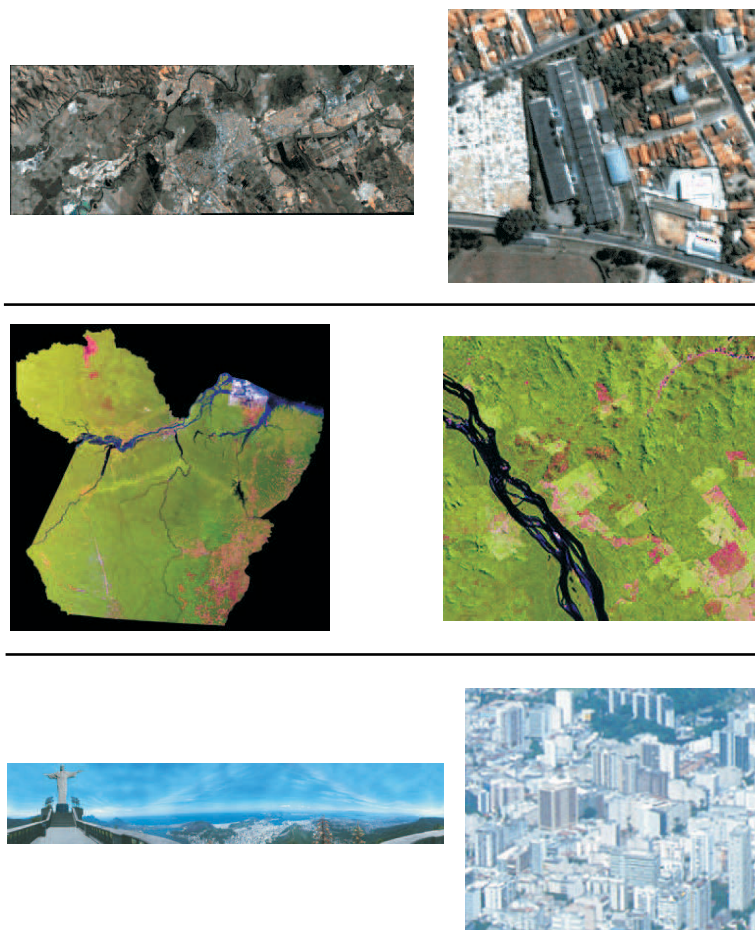


Figura 8.1: (a) Imagem de São José dos Campos; (b) Imagem de satélite do Pará; (c) A panorama virtual do Rio de Janeiro. Ao lado de cada uma dessas imagens é ilustrado uma de suas regiões na resolução original.

Rio de Janeiro		São J. Campos		Pará	
Pixels	MBytes	Pixels	MBytes	Pixels	MBytes
16384×4096	192	16384×16384	768	32768×8192	768

Tabela 8.1: Objetos gráficos com texturas de diferentes resoluções.

As Figuras 8.3, 8.6 e 8.9 mostram as texturas reconstruídas de cada objeto gráfico. Cada região da textura é reconstruída utilizando as texturas dos ladrilhos de maior nível de resolução disponível na representação.

A Tabela 8.2 mostra os resultados obtidos construindo a estrutura em multi-resolução com erro relativo. Observe que na textura do Rio de Janeiro vários ladrilhos foram representados com erros maiores do que o especificado. Já na imagem de satélite do Pará todos os ladrilhos estavam dentro da tolerância. Isto ocorreu porque essa textura possui muita

informação de alta frequência. Assim, o erro relativo entre um nível e o seu predecessor era sempre mais alto do que as tolerâncias utilizadas nos testes. Certamente, utilizando tolerâncias mais altas do que os erros entre os níveis iriam resultar em regiões que seriam representadas com erros maiores do que o especificado. Esse fato é comprovado na imagem de satélite de São José dos Campos. Essa textura possui regiões onde o erro relativo entre um nível e outro é menor do que a tolerância especificada. Nessas regiões os ladrilhos são eliminados e o problema volta a acontecer. É possível identificar essa situação com a tolerância de 3%. Nos casos onde a tolerância é respeitada, a construção com erro relativo obtém os mesmos resultados que a construção por erro acumulado.

Multi-Resolução com Erro Relativo					
Rio de Janeiro					
<i>Tol</i> (%)	Níveis	MBytes	Média(%)	Máximo(%)	> <i>Tol</i> (%)
0	5	255	0.0	0.0	0
1	5	115	0.33066	1.3117849	26
2	5	44.8	1.284146	2.473256	6
3	5	23.4	1.872705	3.477009	5
Pará					
<i>Tol</i> (%)	Níveis	MBytes	Média(%)	Máximo(%)	> <i>Tol</i> (%)
0	7	1024	0.0	0.0	0
1	7	622	0.013503	0.992969	0
2	7	613	0.030849	1.989683	0
3	7	600	0.069980	2.947456	0
São José dos Campos					
<i>Tol</i> (%)	Níveis	MBytes	Média(%)	Máximo(%)	> <i>Tol</i> (%)
0	6	1024	0.0	0.0	0
1	6	691	0.001367	0.974330	0
2	6	682	0.20466	1.997087	0
3	6	644	0.144493	3.439313	2

Tabela 8.2: Resultados obtidos com a representação em multi-resolução adaptativa utilizando erro relativo; o primeiro campo mostra a tolerância desejada; o segundo campo mostra o número de níveis de resolução; o terceiro campo mostra o espaço ocupado pela estrutura em multi-resolução; o quarto campo mostra o erro médio; o quinto campo mostra o erro máximo; o sexto campo mostra o número de ladrilhos que tiveram um erro maior do que o desejado.

A Tabela 8.3 mostra os resultados obtidos com a construção por erro acumulado. Neste caso, em todos os testes nenhum ladrilho reconstruiu a textura com erro maior do que o desejado. Note a textura reconstruída a partir da estrutura em multi-resolução por erro acumulado é sempre menor

ou igual que a por erro relativo. Isso claramente se refletiu nos tamanhos dos arquivos e no número total de ladrilhos utilizados para reconstruir a textura. Em geral, os arquivos gerados pela construção com erro acumulado são maiores que os arquivos gerados pela outra técnica. Isto ocorre porque na construção por erro relativo são eliminados mais ladrilhos mesmo que indevidamente. Nos casos onde a construção por erro relativo respeita a tolerância, os tamanhos dos arquivos são praticamente iguais.

Note que em ambas as técnicas, configurando a tolerância em 0% obtém-se o método normal de multi-resolução. Com a utilização de tolerâncias maiores que 0% a quantidade de espaço utilizado para armazenar a representação em multi-resolução é menor que o espaço utilizado para armazenar a textura original. Isso ocorre porque a técnica de multi-resolução adaptativa funciona como um método de compressão com perdas.

Multi-Resolução com Erro Acumulado					
Rio de Janeiro					
<i>Tol</i> (%)	Níveis	MBytes	Média(%)	Máximo(%)	> <i>Tol</i> (%)
0	5	255	0.0	0.0	0
1	5	120	0.275689	0.999960	0
2	5	65.6	0.7223106	1.999551	0
3	5	38	1.130102	2.997651	0
Pará					
<i>Tol</i> (%)	Níveis	MBytes	Média(%)	Máximo(%)	> <i>Tol</i> (%)
0	7	1024	0.0	0.0	0
1	7	623	0.012699	0.994345	0
2	7	614	0.028951	1.989683	0
3	7	603	0.062464	2.947456	0
São José dos Campos					
<i>Tol</i> (%)	Níveis	MBytes	Média(%)	Máximo(%)	> <i>Tol</i> (%)
0	6	1024	0.0	0.0	0
1	6	691	0.001367	0.974330	0
2	6	683	0.019794	1.6997087	0
3	6	648	0.134729	2.999977	0

Tabela 8.3: Resultados obtidos com a representação em multi-resolução adaptativa utilizando erro acumulado.

As Figuras 8.4, 8.7 e 8.10 mostram as porcentagens médias de erro cometida por cada ladrilho durante a reconstrução das texturas. Observe que utilizando a técnica de erro acumulado todas as regiões da textura foram reconstruídas dentro da margem de erro prevista. O mesmo não ocorre quando se reconstrói a textura a partir da multi-resolução com erro relativo. No caso da textura do Rio de Janeiro, o erro cometido foi 26% maior do que

o especificado. O mesmo ocorreria nas outras texturas caso fossem utilizadas tolerâncias maiores. Os Gráficos 8.5, 8.8 e 8.11 dão uma visão mais cara dessa situação.

Observe que na textura do Pará a região que contém informação de cor não tem uma forma retangular. Logo, foi necessário adicionar uma cor extra (preto) para se obter um suporte retangular. Neste caso, não faz nenhum sentido ter esta informação de cor em alta resolução. Com o método de multi-resolução adaptativa pode-se eliminar todos ladrilhos de alta resolução que possuem essa cor, sem eliminar nenhum dos ladrilhos que possuem as informações de cor que formam a região do Pará.

Uma outra situação onde se faz necessário adicionar informações de cor é quando as dimensões das texturas não estão em potência de 2, ou seja, as dimensões das texturas não obedecem a seguinte restrição: $w = 2^m \times h = 2^n$ pixels, onde $m \geq 1, n \geq 1$. A modificação das dimensões das texturas é necessária devido ao OpenGL que é utilizado na fase de visualização dos objetos gráficos. Muitas implementações da biblioteca OpenGL só permite a visualização das texturas cujas dimensões estejam em potência de 2. Assim, as dimensões das texturas armazenadas nos ladrilhos devem seguir esta restrição para que o objeto gráfico seja visualizado. Com o objetivo de facilitar a fase de decomposição e criação da estrutura em multi-resolução optou-se em restringir as dimensões das texturas para que também estejam em potência de 2. Este recurso de redimensionamento foi utilizado nas texturas de satélite de São J. Campos e do Rio de Janeiro.

A desvantagem de utilizar um método de erro global na fase de comparação é que os ladrilhos de alta resolução que em média possuem mais baixas frequências do que altas são eliminados. Esse problema é bem visível no caso do panorama virtual, onde o rosto do Cristo está envolvido pelo céu, veja a Figura 8.2. Neste caso, é essencial que essa área que contém o rosto estivesse em alta resolução. A Figura 8.2(d) mostra um exemplo do caso extremo onde a qualidade visual é muito baixa. Isto ocorre, porque nesse método de comparação a informação de cor do céu (baixas frequências) prevalece e os ladrilhos de alta resolução dessa região são eliminados. É claro que esse problema pode ser contornado diminuindo o valor da tolerância (Figuras 8.2(a)(b)), entretanto, alguns ladrilhos contendo regiões do céu não serão eliminados. Assim, apesar dos valores de erro global serem baixos, isto não implica que a qualidade visual da imagem reconstruída seja boa em toda a sua extensão. Nas imagens de satélites os resultados foram bem melhores. Isto ocorreu devido a boa distribuição das frequências nessas imagens. Nas imagens do Pará e de São J. Campos foi possível representar

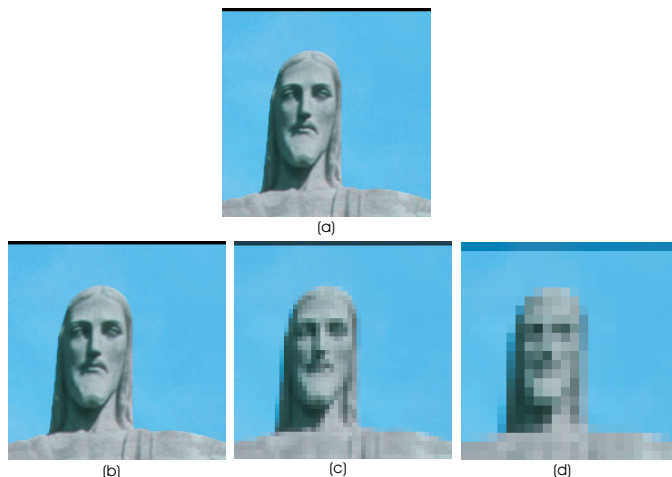


Figura 8.2: (a) mostra a textura original do rosto do Cristo;(b), (c) e (d) mostram a mesma textura reconstruída a partir das multi-resoluções criadas com valores de tolerância de 1, 2 e 3 por cento, respectivamente.

toda a região em preto em baixa resolução sem perder nenhuma informação de cor relevante das imagens. As Figuras 8.6 e 8.9 mostram que as áreas de preto foram reconstruídas com ladrilhos do nível de menor resolução. Compare essas figuras com as Figuras 8.7 e 8.10 que mostram a média de erro cometido na região reconstruída. Observe que o erro varia muito pouco entre as regiões e é bem próximo de zero. No entanto, as Figuras 8.6(c) e 8.9(c) mostram que admitindo-se erro de 3% o problema de eliminação indevida de ladrilhos volta a ocorrer nas regiões de contorno que separam as regiões com as cores que realmente definem os atributos dos objetos gráficos das regiões de preto. Pelos testes realizados basta configurar a tolerância para valores menores que $Tol \leq 2\%$ para se evitar esse problema.

O problema de eliminação indevida de ladrilhos pode ser resolvido a partir de técnicas mais sofisticadas de processamento de imagem. A discussão dessas técnicas está fora do escopo deste trabalho. O trabalho de Gomes e Velho [19] discute em detalhes várias técnicas de processamento e análise de imagens.

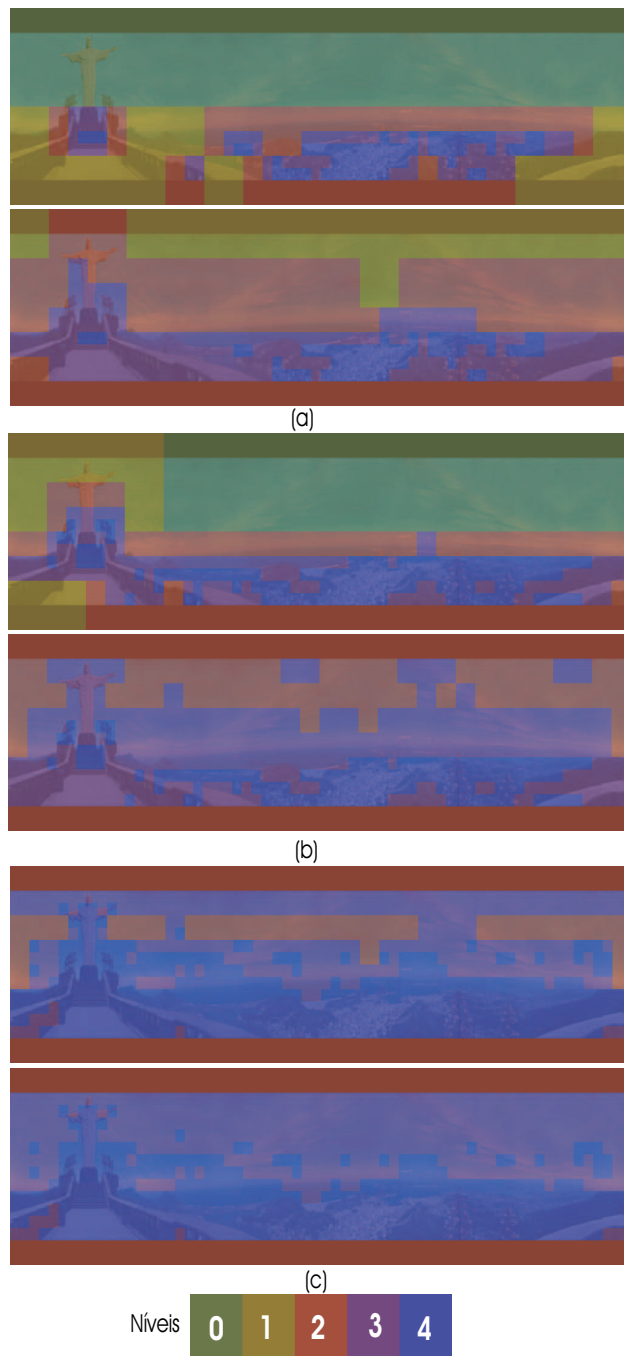


Figura 8.3: As figuras (a), (b) e (c) mostram as três texturas do Rio de Janeiro reconstruídas a partir das estruturas em multi-resolução criadas com tolerâncias de 3%, 2% e 1%, respectivamente. Na parte superior a textura reconstruída com a técnica de erro relativo e na parte inferior a textura reconstruída com a técnica de erro acumulado. A palette de cor informa o nível de resolução utilizado para reconstruir cada área da textura. O nível 0 é o nível de menor resolução e nível 4 o de maior.

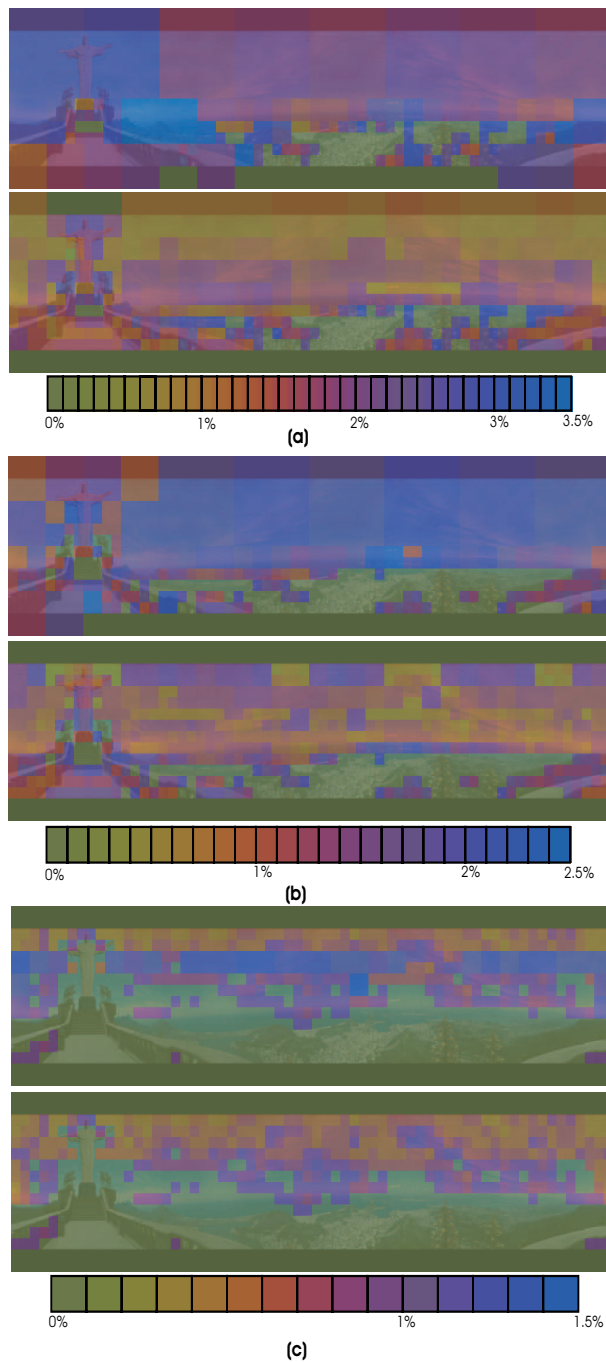


Figura 8.4: As figuras(a), (b) e (c) mostram a porcentagem de erros cometidos na reconstrução da textura do Rio de Janeiro a partir dos ladrilhos de maior resolução; (a) 3% de tolerância; (b) 2% de tolerância; (c) 1% de tolerância; Na parte superior mostra os resultados com erro relativo e na parte inferior mostra os resultados com erro acumulado.

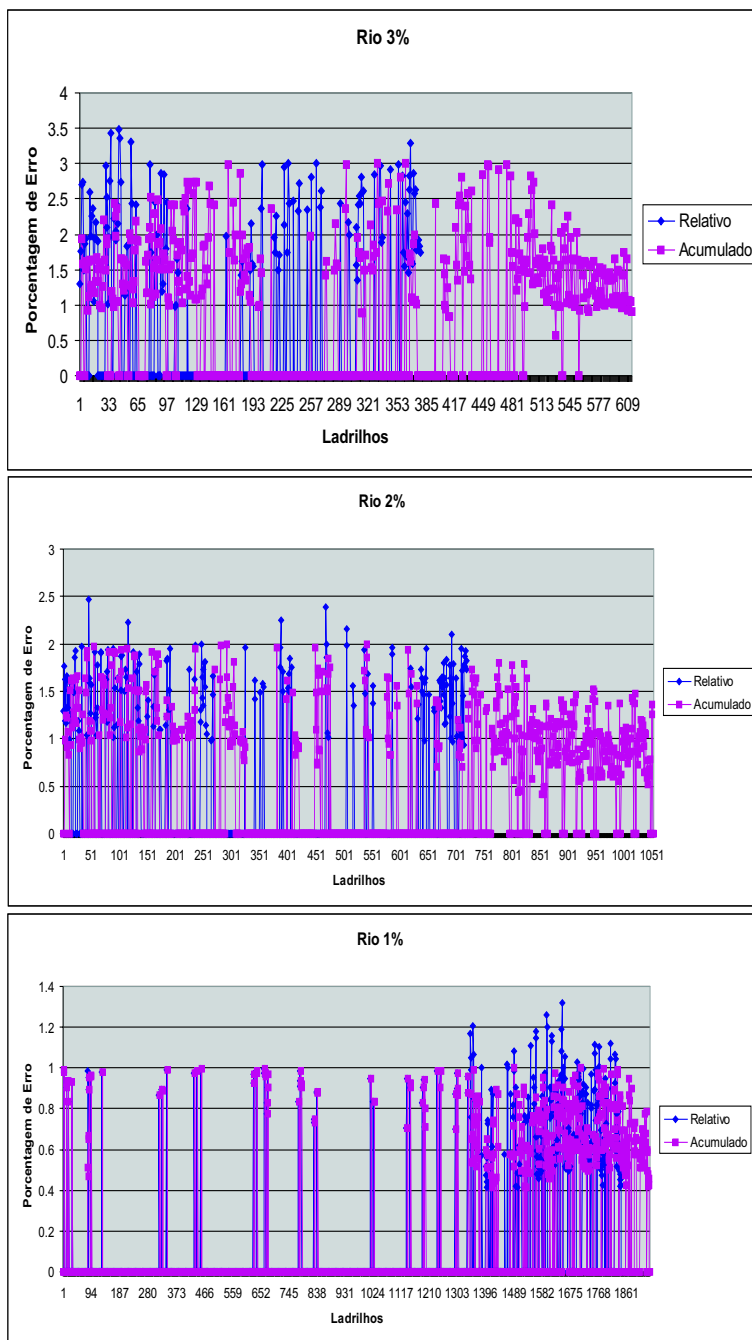


Figura 8.5: O gráfico mostra o erro médio cometido para cada ladrilho durante o processo de reconstrução da textura do Rio de Janeiro. O *eixo X* representa cada ladrilho que foi utilizado para reconstruir a textura e o *eixo Y* representa o erro cometido nessa reconstrução.

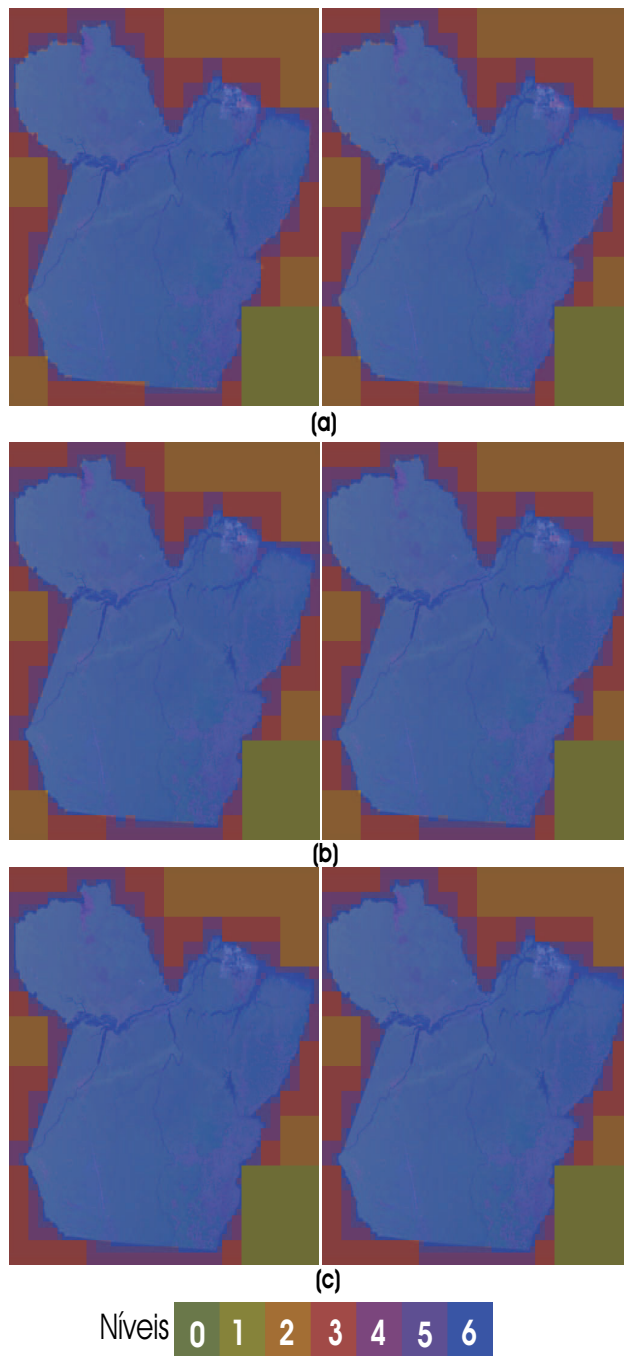


Figura 8.6: As figuras (a), (b) e (c) mostram as texturas do Pará reconstruídas a partir das estruturas em multi-resolução criadas com torâncias de 3%, 2% e 1%, respectivamente. Do lado esquerdo estão as texturas reconstruídas com a técnica de erro relativo e do lado direito estão as texturas reconstruídas com a técnica de erro acumulado. A palette de cor informa o nível de resolução utilizado para reconstruir cada área da textura. O nível 0 é o nível de menor resolução e nível 6 o de maior.

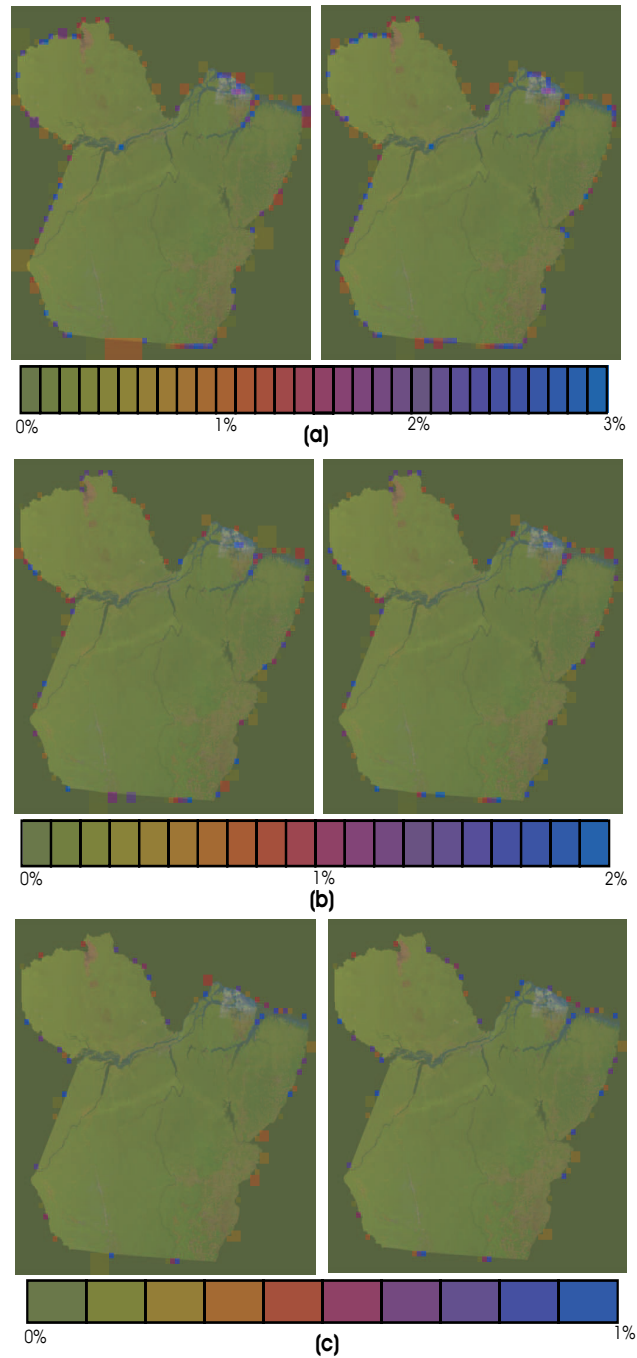


Figura 8.7: As Figuras(a), (b) e (c) mostram a porcentagem dos erros cometidos na reconstrução da textura do Para; (a) 3% de tolerância; (b) 2% de tolerância; (c) 1% de tolerância; Do lado esquerdo estão os resultados obtidos com erro relativo e do lado direito estão os resultados com erro acumulado.

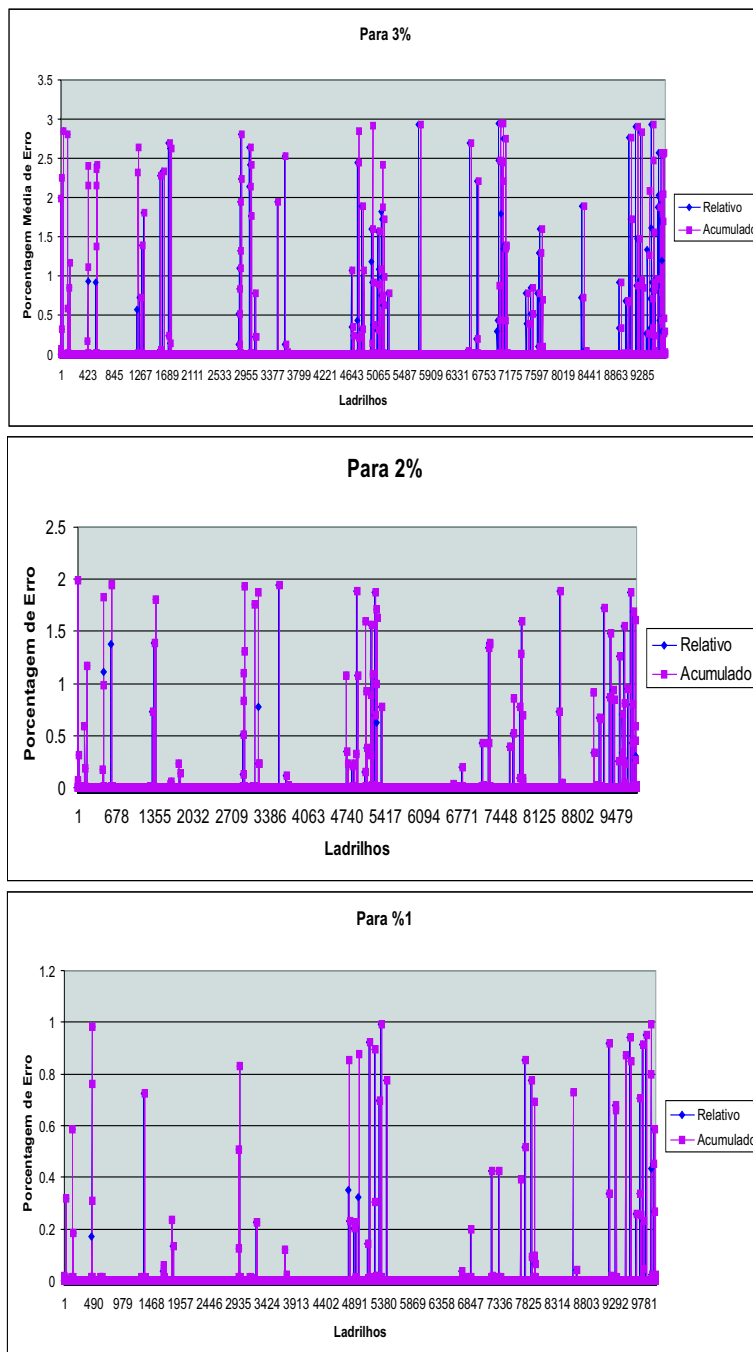


Figura 8.8: O gráfico mostra o erro médio cometido para cada ladrilho durante o processo de reconstrução da textura do Pará. O *eixo X* representa cada ladrilho que foi utilizado para reconstruir a textura e o *eixo Y* representa o erro cometido nessa reconstrução.

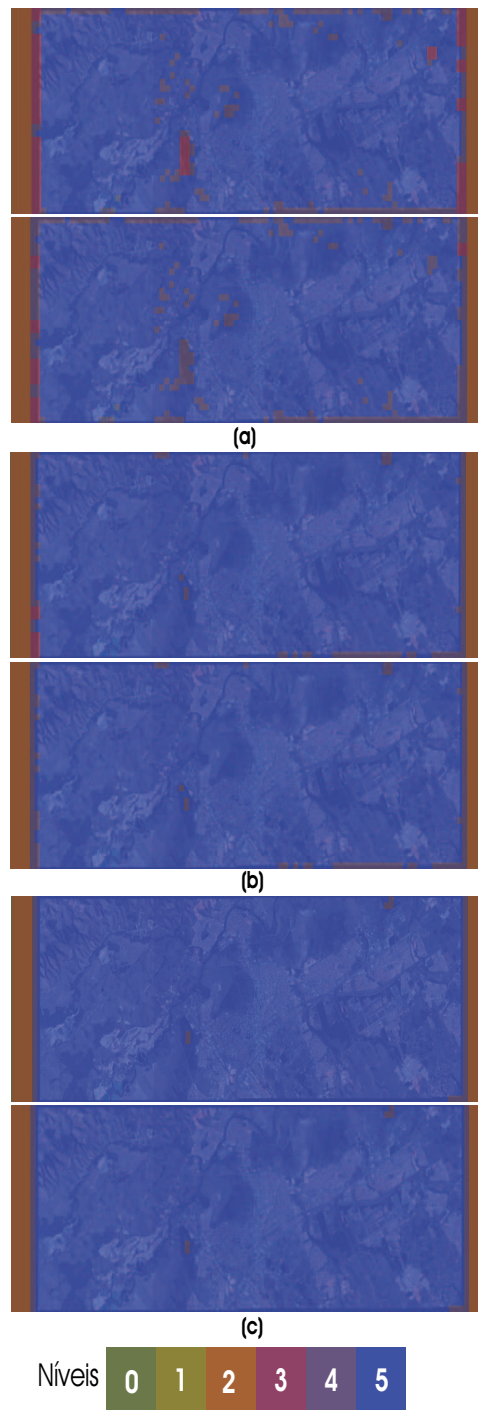


Figura 8.9: As Figuras (a), (b) e (c) mostram as texturas de São José dos Campos reconstruídas a partir das estruturas em multi-resolução criadas com tolerâncias de 1%, 2% e 3%, respectivamente. Na parte superior a textura reconstruída com a técnica de erro relativo e na parte inferior a textura com a técnica de erro acumulado. A palette de cor informa o nível de resolução utilizado para reconstruir cada área da textura. O nível 0 é o nível de menor resolução e nível 5 o de maior.

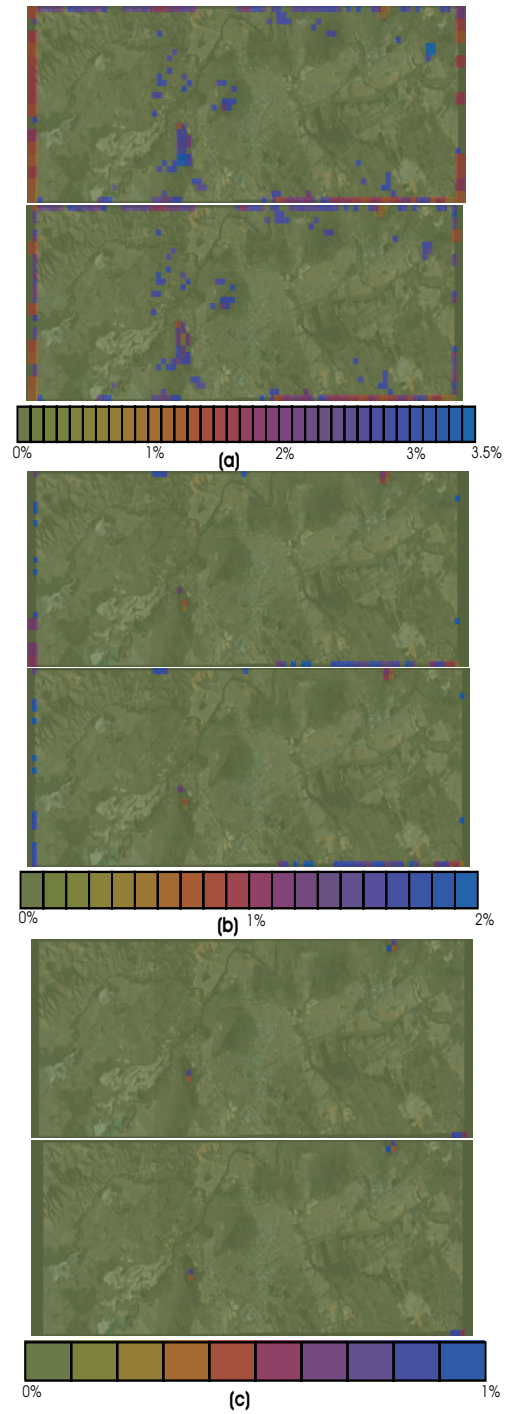


Figura 8.10: As Figuras(a), (b) e (c) mostram a porcentagem de erros cometidos na reconstrução da textura de São J. Campos; (a) 3% de tolerância; (b) 2% de tolerância; (c) 1% de tolerância; Na parte superior mostra os resultados com erro relativo e na parte inferior mostra os resultados com erro acumulado.

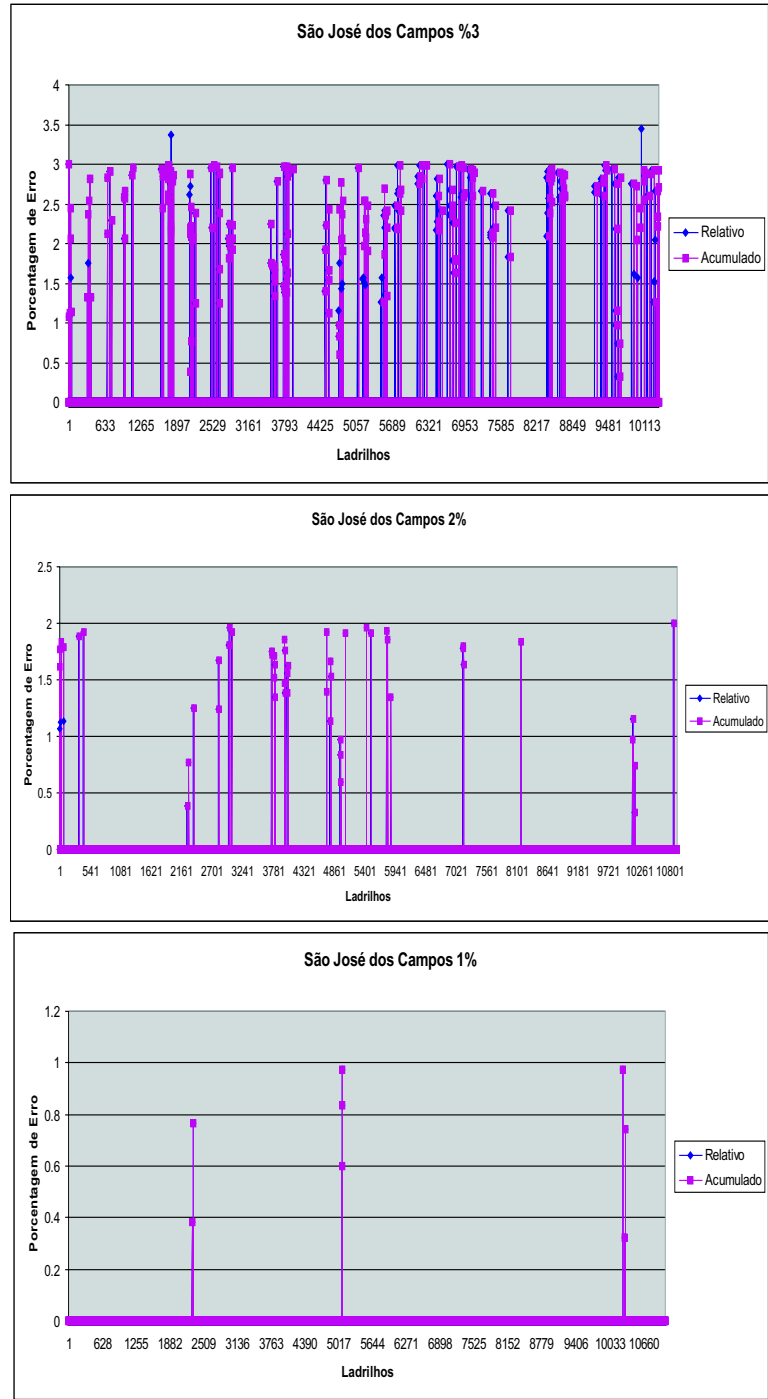


Figura 8.11: O gráfico mostra o erro médio cometido para cada ladrilho durante o processo de reconstrução da textura da cidade São José dos Campos. O *eixo X* representa cada ladrilho que foi utilizado para reconstruir a textura e o *eixo Y* representa o erro cometido nessa reconstrução.

8.2 Sistema de Gerenciamento de Memória

O sistema de gerenciamento de memória foi testado em situações onde os objetos gráficos estão armazenados localmente e em situações onde estão armazenados remotamente. No caso dos dados estarem armazenados localmente, o sistema possuirá três estágios de armazenamento, como mostra a Figura 8.12(a).

Na situação onde os dados estão armazenados remotamente, a aplicação cliente foi testada com sistemas de três e quatro estágios de armazenamento. A configuração de quatro estágios apenas acrescenta o estágio de rede ao sistema (Figura 8.12(b)). Na configuração de três estágios, o estágio de disco não é utilizado. Neste caso, os dados são transferidos do estágio de rede diretamente para o estágio de memória RAM (Figura 8.12(c)). A aplicação que age como servidor possui um sistema com dois estágios de armazenamento (Figura 8.12(d)).

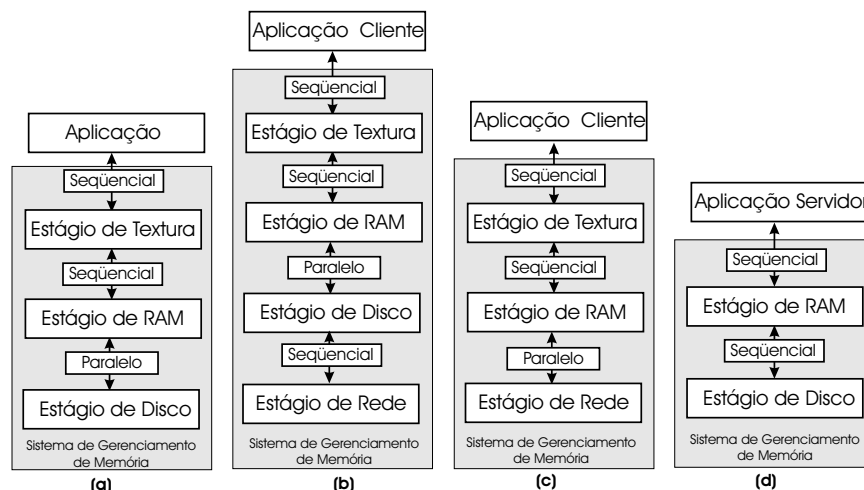


Figura 8.12: As configurações do sistema de gerenciamento de memória utilizadas para realizar os experimentos.

Observe que o sistema de gerenciamento de memória está ligado a aplicação de forma sequencial. Isto foi necessário porque o estágio de textura foi implementado utilizando a biblioteca OpenGL. O OpenGL não suporta paralelismo. Além disso, os objetos de textura do OpenGL estão ligados ao contexto da tela e as operações com os mesmos só funcionam se forem realizadas enquanto o contexto estiver ativo e na mesma thread. A ligação paralela foi justamente colocada na região onde começa a surgir uma grande diferença de velocidade entre os estágios de armazenamento.

A comunicação entre o servidor e a aplicação cliente é feita em duas fases. A primeira fase, chamada de *fase de conexão*, é destinada para o servidor transferir as informações de configuração e de banco de dados. O protocolo de comunicação utilizado é o TCP/IP. A informação de configuração é composta pelos seguintes dados:

- O Identificador da porta por onde serão transmitidos dos dados dos ladrilhos.
- O tipo de protocolo utilizado para a transmissão dos dados. O tipo pode ser o TCP/IP ou o UDP.
- A forma como os dados dos ladrilhos serão transmitidos. Essa forma pode ser com compressão ou não. No caso dos dados serem mandados com compressão, essa compressão ainda pode ser com perdas ou sem perdas. Na compressão com perdas a textura do ladrilho é convertida para o formato JPG e transmitida. Na compressão sem perdas é utilizado o formato GZ para comprimir a textura.

Todos esses dados são especificados no momento da inicialização do servidor. A informação de banco de dados contém a lista de todos os objetos gráficos que estão armazenados localmente no servidor. No caso do servidor estar conectado a outro servidor (isto é feito acrescentando o nível de rede no sistema de gerenciamento), os objetos gráficos disponíveis neste último são acrescentados a essa lista de objetos que é transmitida para a aplicação cliente. Nos testes esta funcionalidade não foi explorada. A Figura 8.13 ilustra essa fase de comunicação entre o cliente e o servidor.

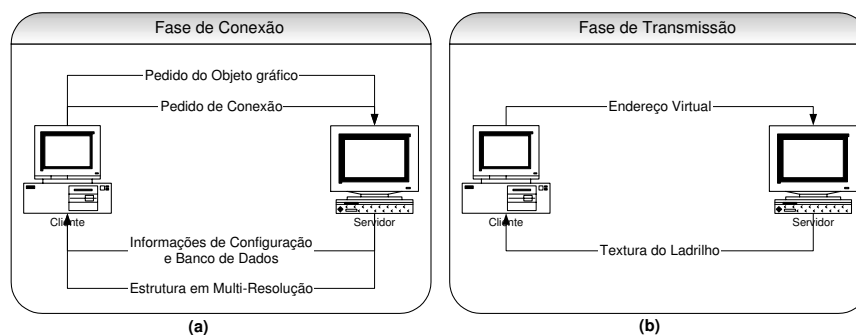


Figura 8.13: Modelo da troca de informações entre o cliente e o servidor;(a) Informações trocadas na fase de conexão;(b) Informações trocadas na fase de transmissão.

Observe que na fase de conexão o cliente envia ao servidor uma mensagem solicitando um objeto gráfico. O servidor responde enviando

todos os dados necessários para que o sistema cliente possa montar as estruturas em multi-resolução da representação geométrica e da textura.

A segunda fase é chamada de *fase de transmissão*. Esta fase é apenas destinada para a transmissão das texturas dos ladrilhos. O sistema cliente envia o endereço virtual do ladrilho para o servidor. Em resposta ao pedido, o servidor envia os dados que pertencem ao ladrilho requisitado.

Os experimentos tiveram como objetivo mostrar a flexibilidade de configuração do sistema de gerenciamento de memória e analisar o desempenho do sistema de predição. O sistema de liberação proposto neste trabalho foi colocado nos estágios de memória de Textura e de RAM. No estágio de disco foi utilizado um sistema de liberação baseado no algoritmo LRU. A técnica de predição mostrada na tese foi comparada com a técnica que utiliza uma área de segurança para carregar antecipadamente os ladrilhos.

Os sistemas de predição e de liberação foram testados com duas aplicações de visualização em tempo-real. A primeira análise foi realizada na aplicação de visualização de imagens de satélite. Em seguida, foram feitas as mesmas análises para a aplicação de visualização de panoramas virtuais.

8.2.1 Especificação dos Experimentos

As variáveis levadas em consideração na fase de teses foram classificadas em dois grupos. O primeiro grupo de variáveis está relacionado com a especificação do tipo de hardware e o sistema operacional que foram utilizados no experimento. O segundo grupo são das variáveis responsáveis pela configuração do software e controlam a forma como a aplicação de visualização, o sistema de gerenciamento de memória e os dados vão utilizar os recursos de processamento e armazenamento disponíveis no hardware.

Nos experimentos foram utilizadas três configurações diferentes de hardwares. As configurações utilizadas foram as seguintes:

1. Um Dual-PentiumII, 256Mb de memória, placa gráfica Geforce2MX e HD SCSI de 4Gb (5400 rpm).
2. Um PentiumIII, 768Mb de memória RAM, placa gráfica Oxygen GVX-420 e dois HDs SCSI de 20Gb (10000 rpm).
3. Um Pentium4, 1Gb de memória RAM, placa Geforce4TI e dois HDs IDE de 40Gb (7200 rpm).

4. Uma rede local (Ethernet) com velocidade de transmissão de 100Mbps (Megabits/s).

Note que uma característica comum de todas as placas gráficas escolhidas é que possuem implementação de textura em hardware. Essa característica é indispensável para que as duas aplicações possam visualizar seus dados em tempo-real. O sistema operacional instalado nas configurações PentiumII e PentiumIII era o Windows2000 Professional. No Pentium4 foi utilizado o WindowsXP. Em todos os experimentos as máquinas PentiumII e Pentium4 foram utilizadas para executar as aplicações de visualização. O Pentium III foi apenas utilizado como servidor nos testes com o estágio de rede e as outras duas máquinas assumiram o papel de clientes.

As configurações de softwares especificam os valores das variáveis que controlam a taxa de quadros por segundo da aplicação de visualização, a quantidade de memória utilizada em cada estágio de armazenamento, a quantidade de quadros futuros calculados no sistema de predição, o tamanho da textura do ladrilho e a forma como a textura é transmitida pela rede. A Tabela 8.4 mostra como essas variáveis foram especificadas em relação ao tipo do hardware utilizado no experimento. Observe que a especificação do hardware mais a especificação das variáveis de controle do software geram um tipo de experimento.

A tabela 8.4 mostra os experimentos que foram feitos com a utilização mínima e a máxima dos recursos de armazenamento. No entanto, é necessário que as configurações especificadas para a aplicação e para o sistema de gerenciamento sejam compatíveis com o poder de processamento e de armazenamento do hardware. Por exemplo, suponha que seja especificado para a aplicação sintetizar 30 quadros/s. Neste caso a aplicação deve gastar 33 milissegundos para realizar a operação de síntese da imagem e a operação de predição. Entretanto, se todo o tempo for gasto na operação de síntese, então o sistema de predição nunca será executado. Desta forma, é necessário especificar uma taxa de quadros por segundo que permita ao sistema realizar as duas operações.

Em relação a questão de armazenamento é necessário que o hardware tenha a capacidade de reservar o espaço exigido na configuração. Caso contrário, o sistema operacional irá gerenciar a quantidade excedente. Neste caso, os resultados mostrarão uma perda de desempenho da aplicação e do sistema de gerenciamento à medida que essa quantidade excedente de armazenamento aumenta.

Devido a primeira regra de carregamento o estágio mais alto na hierarquia (memória de textura) deve ser capaz de armazenar todos

Máquina	Ladrilho	Resolução	Quadros/s	Textura	RAM	Disco	Rede
Dual PentiumII (Cliente)	64x64	1024x768	40	4Mb	16Mb	Sim	Não
	64x64	1024x768	40	8Mb	32Mb	Sim	Não
	128x128	1024x768	40	16Mb	32Mb	Sim	Não
	128x128	1024x768	40	8Mb	32Mb	64Mb	Sim
	128x128	1024x768	40	8Mb	32Mb	Não	Sim
	64x64	1024x768	40	16Mb	32Mb	128Mb	Sim
	128x128	1024x768	40	16Mb	32Mb	Não	Sim
Pentium4 (Cliente)	64x64	1024x768	40	4Mb	16Mb	Sim	Não
	64x64	1024x768	40	8Mb	32Mb	Sim	Não
	128x128	1024x768	40	16Mb	32Mb	Sim	Não
	128x128	1024x768	40	32Mb	64Mb	Sim	Não
	128x128	1024x768	40	8Mb	32Mb	64Mb	Sim
	128x128	1024x768	40	8Mb	32Mb	Não	Sim
	64x64	1024x768	40	16Mb	32Mb	128Mb	Sim
	128x128	1024x768	40	16Mb	32Mb	Não	Sim
PentiumIII (Servidor)	64x64	Não Relev.	Não Relev.	Não	64Mb	Sim	Não
	128x128	Não Relev.	Não Relev.	Não	64Mb	Sim	Não

Tabela 8.4: Todas as configurações utilizadas nos experimentos. O mesmo experimento foi realizado nas duas máquinas com o objetivo de poder avaliar o quanto o desempenho do sistema está ligado a arquitetura do hardware.

os ladrilhos do menor nível de resolução. Além disso, o espaço restante deve ser suficiente para armazenar os ladrilhos visíveis na tela mais os seus ancestrais. Note que a especificação da resolução da tela e a quantidade de níveis de resolução de um objeto gráfico estão diretamente relacionados com o requerimento mínimo de armazenamento exigido pelo sistema.

8.2.2 Especificação dos Parâmetros da Câmera

A função de variação dos parâmetros da câmera foi a mesma para todos os experimentos. Foram utilizadas duas funções. Uma para descrever a trajetória da câmera e outra para controlar as operações de zoom.

A trajetória da câmera foi descrita pela curva de Lissajoux $c(\theta) = (\cos 13\theta, \sin 17\theta)$, onde $\theta = [0 \dots 2\pi]$. A Figura 8.14 mostra o efeito dessa curva quando aplicada nos parâmetros das câmeras utilizadas nas aplicações de visualização de imagens de satélite e de panoramas virtuais. Observe que foi necessário aplicar uma função de mapeamento que leva valores da curva $c(\theta)$ em valores que estão dentro do domínio dos parâmetros das câmeras.

A velocidade média da trajetória é controlada pelo incremento da variável θ . A velocidade real da câmera em cada ponto da curva é

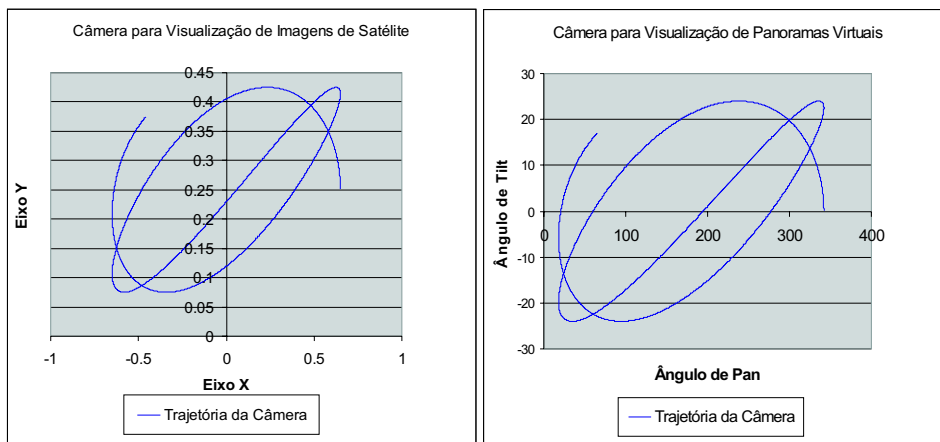


Figura 8.14: À esquerda os pontos da curva são mapeados nos parâmetros da câmera para visualização de imagens de satélite. À direita os mesmos pontos mapeados nos parâmetros da câmera para visualização da panoramas virtuais.

inversamente proporcional a sua curvatura. Ou seja, quanto maior a curvatura em um ponto da curva menor a velocidade da câmera nesse ponto. A Figura 8.15 mostra o gráfico do valor absoluto da velocidade câmera e o ângulo entre o vetor de velocidade e o de aceleração. Se os ângulos são menores que 90° , isso indica que a câmera está em movimento acelerado. Caso contrário, a câmera está em movimento desacelerado.

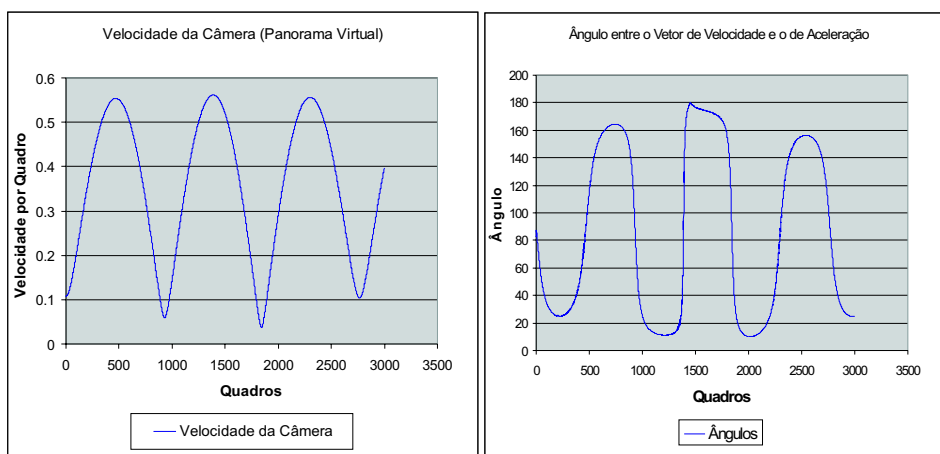


Figura 8.15: À esquerda a norma do vetor de velocidade para cada ponto da curva. À esquerda o ângulo entre o vetor de velocidade e o de aceleração. Esses gráficos foram obtidos com um valor constante de incremento do ângulo θ .

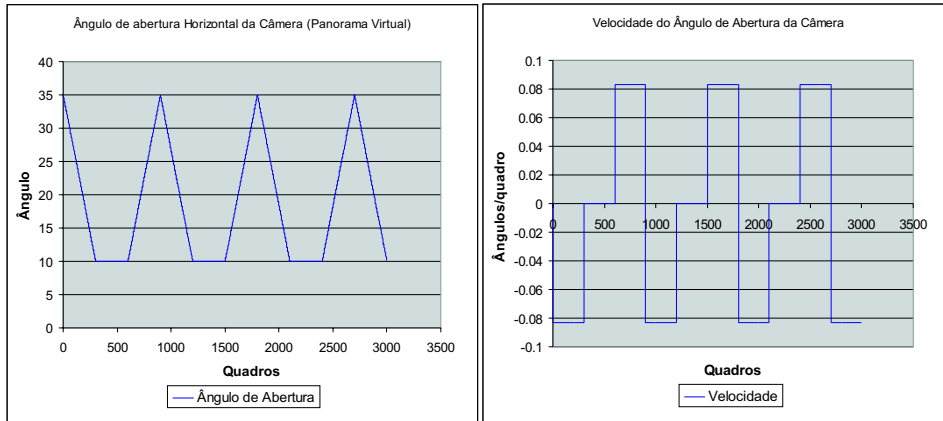


Figura 8.16: À direita os ângulos de abertura horizontal da câmera. À esquerda a velocidade de variação desses ângulos por quadro.

A operação de zoom é realizada a partir da função

$$Zoom = \begin{cases} MinZoom \cdot z + MaxZoom \cdot (1 - z) & \text{Se } nFunc=0 \\ MinZoom & \text{Se } nFunc=1 \\ MinZoom \cdot (z - 1) + MaxZoom \cdot z & \text{Se } nFunc=2 \end{cases},$$

onde $nFunc = MOD(Contf/Patch, 3)$, $z = MOD(Contf, Patch)/Patch$ e $MOD(x, y)$ é o resto da divisão de x/y . As constantes $MinZoom$ e $MaxZoom$ determinam um intervalo de valores onde a operação de zoom é válida. A variável $Contf = [0 \dots nFrames]$ é um contador de quadros, onde $nFrames$ é o número total de quadros sintetizados durante o experimento. A variável $Patch$ determina a quantidade de vezes que uma determinada função será avaliada consecutivamente. A Figura 8.16 mostra os valores dos ângulos de abertura da câmera obtidos a partir dessa função. Observe no gráfico à esquerda que em algumas regiões a velocidade varia bruscamente. Essa característica da função de zoom dificulta a tarefa de predição.

Os experimentos foram realizados com a câmera se movendo a partir de uma função de incremento do ângulo θ . A função que define o incremento em cada quadro é muito parecida com a utilizada na operação de zoom. A diferença é que a função se torna constante quando o incremento é máximo. O incremento mínimo foi definido por $D_{\ominus} = 2\pi/(32*nFrames)$ e o incremento máximo definido por $D_{\theta} = 2\pi/(4*nFrames)$. A Figura 8.17(a) mostra a norma do vetor velocidade para cada quadro. Observe que agora a velocidade da câmera varia muito mais que a velocidade obtida com incremento constante (Figura 8.15). Com a variação do incremento D_{θ} a câmera virtual percorreu a trajetória mostrada na Figura 8.17(b).

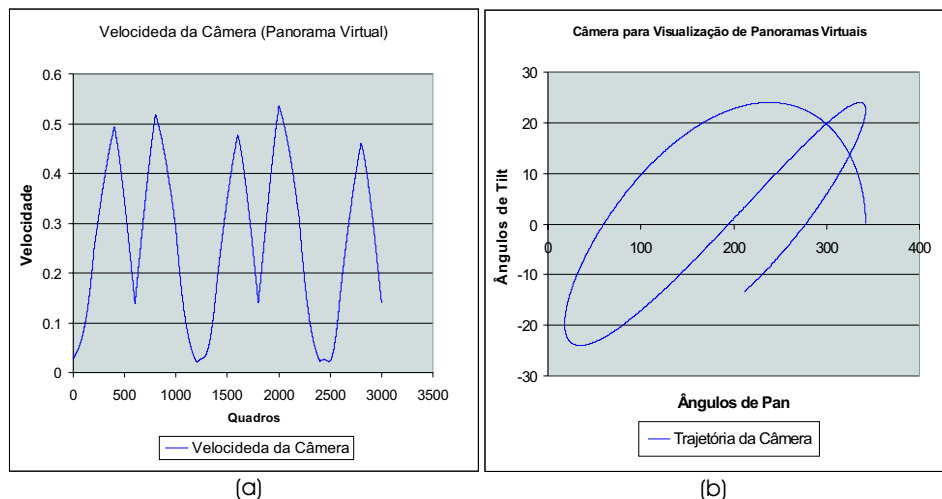


Figura 8.17: (a) Velocidade da câmera obtida a partir do incremento dinâmico; (b) Nova trajetória percorrida pela câmera.

Em todos os experimentos a quantidade total de quadros avaliada foi de $nFrames = 3000$ quadros e a variável *Patch* foi definida por $Patch = 300$ para a operação de zoom e $Patch = 200$ para o cálculo do incremento D_{θ} .

8.2.3 Especificação das Imagens

A imagem de satélite utilizada nos experimentos foi uma imagem de São José dos Campos. A imagem possui uma resolução de 32768×8192 pixels e são necessários 768 Mbytes para armazená-la. A representação em multi-resolução dessa imagem foi realizada sem a utilização do processo de eliminação de ladrilhos, ou seja, a estrutura em multi-resolução da textura está completa. A representação em multi-resolução possui seis níveis e ocupou um pouco mais de 1 Gbyte. A Figura 8.18 mostra a imagem de satélite e a região que foi visualizada durante os experimentos.

Nos experimentos com a aplicação de visualização de panoramas foi utilizada uma imagem da panorâmica da vista do Corcovado. A imagem do Corcovado tem uma resolução de 32768×8192 pixels e ocupa 768 Mbytes. A representação em multi-resolução dessa imagem possui todos os ladrilhos e são necessários em torno de 1 Gbyte para armazená-la. Essa representação em multi-resolução é idêntica à representação em multi-resolução da imagem de satélite. Desta forma, os dados de textura não tiveram nenhuma influência sobre os resultados obtidos nos experimentos.



Figura 8.18: Imagem de satélite de São José dos Campos. A curva representa a trajetória realizada pela câmera durante os experimentos.

A Figura 8.19 mostra a imagem panorâmica e a região visualizada pela câmera durante os experimentos.

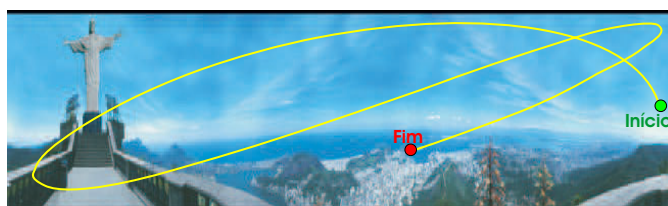


Figura 8.19: Imagem panorâmica da vista do Corcovado. A curva mostra a trajetória realizada pela câmera.

8.2.4 Especificação das Medidas

A análise de desempenho foi medida levando em consideração a quantidade média de ladrilhos por quadro que foi visualizada na resolução correta. Esta medida está intimamente relacionada com o problema de ausência de páginas, pois a aplicação só sintetiza um ladrilho na resolução errada quando o ladrilho correto não está armazenado no estágio de textura. Essa média de ladrilhos certos por quadro é dada em porcentagem e é definida por:

$$Media_{lc} = 100 \cdot \frac{\sum_{i=1}^{3000} \frac{LadrilhosCertos_i}{TotalLadrilhos_i}}{3000},$$

onde $LadrilhosCertos_i$ representa o número de ladrilhos sintetizados na resolução correta e $TotalLadrilos_i$ é o número total de ladrilhos sintetizados no quadro i .

O cálculo da $Media_{lc}$ não pode ser interpretado como qualidade perceptual da imagem sintetizada. O poder de um observador perceber

detalhes de um objeto sintetizado na tela depende do tempo que este objeto permanece na mesma. Desta forma, quanto mais rápido o observador passar por um objeto, menor é sua capacidade de perceber seus detalhes. Assim, o nível de resolução ótimo que um objeto deve ser sintetizado deve ser estimado levando em consideração o fator espacial e o temporal. O fator espacial consiste em obter uma razão 1 : 1 entre os pixels da tela e os pontos visíveis do objeto. O fator temporal consiste em calcular o tempo que esses pontos visíveis permanecem na tela. A forma de combinar esses dois fatores para obter uma métrica relacionada com qualidade perceptual pode ser bastante complicada. Observe que a medida $Media_{lc}$ só leva em consideração o fator espacial.

Outro fator considerado na análise foi a relação entre a quantidade de ladrilhos pedidos em um quadro i e a quantidade de ladrilhos cancelados no quadro $i + 1$. Esse fator mede o quanto o sistema de predição está cometendo erros nos cálculos de estimar os ladrilhos que provavelmente serão necessários nos quadros futuros. A medida do fator de acerto é dada em porcentagem e definida por:

$$Media_{ap} = 100 - 100 \cdot \frac{\sum_{i=1}^{2999} \frac{LadrilhosCancelados_{i+1}}{LadrilhosCarregados_i}}{2999},$$

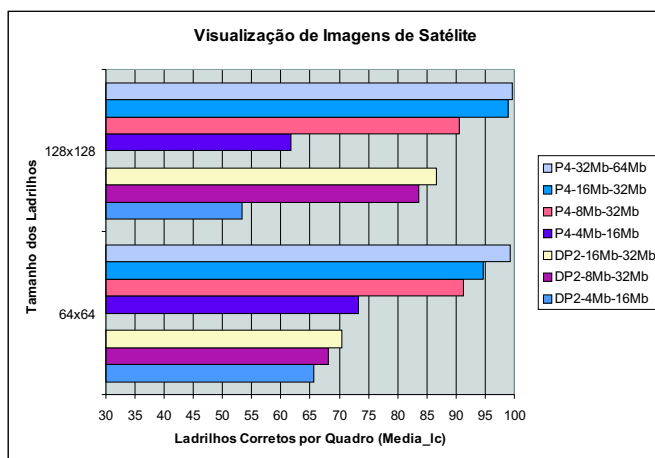
onde $LadrilhosCarregados_i$ representa o número de pedidos de carregamento no quadro i e $LadrilhosCancelados_{i+1}$ é o número de pedidos de cancelamento no quadro $i + 1$. O objetivo dessa medida é mostrar que valores baixos de $Media_{lc}$ não está relacionado à erros cometidos no algoritmo de predição. Lembre-se que o sistema de predição escolhe o nível de resolução baseado em fatores de velocidade da câmera e de taxa de transferência de dados suportada pelo hardware. Assim, o nível de resolução pedido pela aplicação muitas vezes não pode ser atendido por estar infringindo um desses dois fatores.

8.2.5

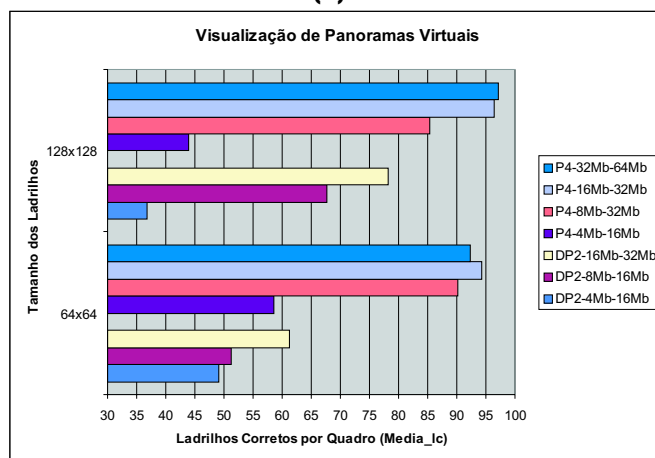
Análise com Armazenamento Local

Os primeiros experimentos foram realizados para comparar o desempenho do sistema em relação à configuração do tamanho do ladrilho e do espaço de memória reservado em cada estágio de armazenamento. A Figura 8.20 mostra os gráficos dos resultados obtidos com o Dual-PentiumII e Pentium4. O gráfico do lado esquerdo corresponde aos resultados obtidos

com a aplicação de visualização de imagens de satélite e do lado direito aos resultados obtidos com a aplicação de visualização de panoramas virtuais.



(a)



(b)

Figura 8.20: O gráfico mostra a porcentagem média de ladrilhos corretos por quadro ($Media_{lc}$); (a) Resultados com a visualização de imagens de satélite; (b) Resultados com a aplicação de panoramas virtuais. A Ordem a legenda é Máquina-Memória de Textura-Memória RAM, onde *DP2* representa Dual-Pentium2 e *P4* representa Pentium4.

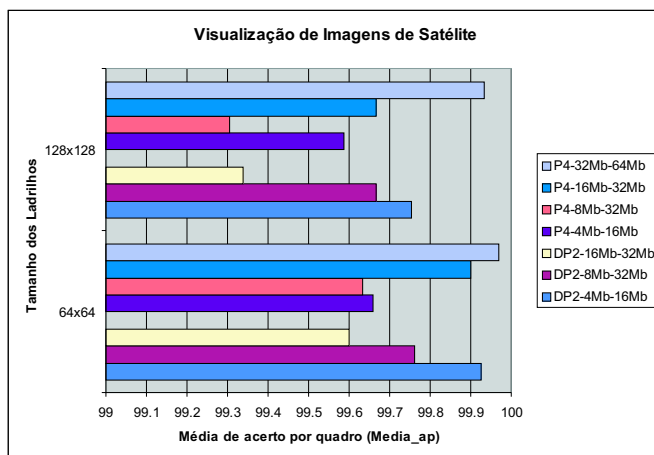
Em todos os experimentos as duas aplicações tiveram um maior desempenho no Pentium4. A vantagem obtida nessa arquitetura é devido aos seus componentes de processamento e armazenamento que são mais rápidos que na arquitetura Dual-PentimII. Os resultados mostram que o desempenho das aplicações são melhores com ladrilhos de tamanho 64×64 pixels quando os espaços reservados para as memórias de textura e RAM são reduzidos. Em casos onde os espaços reservados para essas memórias são maiores, observou-se que as aplicações obtiveram maior desempenho com ladrilhos de tamanho 128×128 .

O espaço reduzido das memórias de textura e RAM resulta em acessos constantes ao disco. Como o disco é o dispositivo mais lento, o desempenho do sistema está intimamente relacionado ao seu desempenho. Observou-se que os discos tiveram um melhor desempenho com ladrilhos de 64×64 . À medida que o espaço reservado para as memórias de textura e RAM aumentam, a necessidade de acesso ao disco diminui. Neste caso, as operações de transferência entre a memória de textura e a memória RAM tem maior influência no desempenho das aplicações. Os experimentos mostraram que ladrilhos de 128×128 são adequados para as operações de transferência de dados entre memória de textura e RAM.

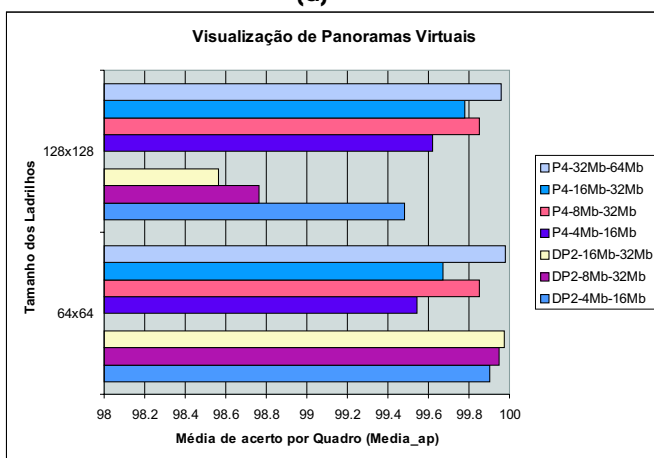
O aumento do número de ladrilhos processados por quadro foi outro fator que influenciou na queda de desempenho do sistema ao utilizar ladrilhos de 64×64 . Para sintetizar um quadro com ladrilhos de 64×64 é necessário processar quatro vezes mais ladrilhos do que com ladrilhos de 128×128 . Esse fator ficou mais evidente nos experimentos de visualização de panoramas com o Dual-PentiumII. A aplicação de panoramas virtuais processou em média 235 ladrilhos por quadro com ladrilhos de 64×64 enquanto que a aplicação de visualização de imagens de satélite processou 175. Com ladrilhos de 128×128 essa quantidade é quatro vezes menor.

O sistema de predição foi configurado para estimar os ladrilhos que deveriam ser carregados dentro dos primeiros 50 quadros futuros. Isto quer dizer que sistema tentou estimar o que vai acontecer um pouco mais de um segundo no futuro. A Figura 8.21 mostra a porcentagem de acerto do sistema de predição. Observe que o sistema de predição em média acertou mais de 98% das vezes. Isto mostra que o sistema conseguiu se adaptar a todos os tipos de configurações, hardwares e aplicações.

No caso de armazenamento local os erros cometidos pelo sistema de predição são mascarados pelas altas taxas de transferência dos dispositivos de armazenamento. A Figura 8.22(a) mostra o gráfico da porcentagem de acerto onde o sistema teve o pior desempenho. Compare os dois gráficos e note que os quadros onde houve uma variação brusca na velocidade do parâmetro de zoom correspondem aos quadros onde o sistema de predição cometeu mais erros. Essas discontinuidades de movimento fazem com que os valores da aceleração sejam muito altos. Quando ocorre esse pico a área de carregamento calculada é bem maior do que área calculada no quadro anterior e muitos ladrilhos são pedidos pelo sistema de predição. Com a normalização do movimento a área de carregamento volta para o seu tamanho normal. Assim, muitos ladrilhos que foram pedidos no quadro anterior ficam fora a área de carregamento do quadro atual e são cancelados.



(a)



(b)

Figura 8.21: Os gráficos que mostram a média de acertos do sistema de predição; (a) Visualização de imagens de satélite; (b) Visualização de panoramas virtuais.

Esse fato vai ficar mais evidente nos experimentos com o estágio de rede devido a diminuição das taxas de transmissão dos dados.

O sistema de predição adaptativa proporcionou um melhor desempenho para a aplicação do que a predição por área de segurança. A Figura 8.23 mostra o gráfico de desempenho para os dois métodos. A predição por área de segurança sempre tenta trazer o nível de resolução ideal para a aplicação, ou seja, tem o comportamento de um algoritmo guloso. Esse tipo de predição só obtém bons resultados quando a máquina que possui dispositivos de armazenamento e de processamento rápidos. Em casos onde o hardware não pode carregar todos os ladrilhos da área de segurança, observa-se uma grande variação na resolução com que os quadros são sintetizados. Isto ocorre porque o tempo para o carregamento só permite

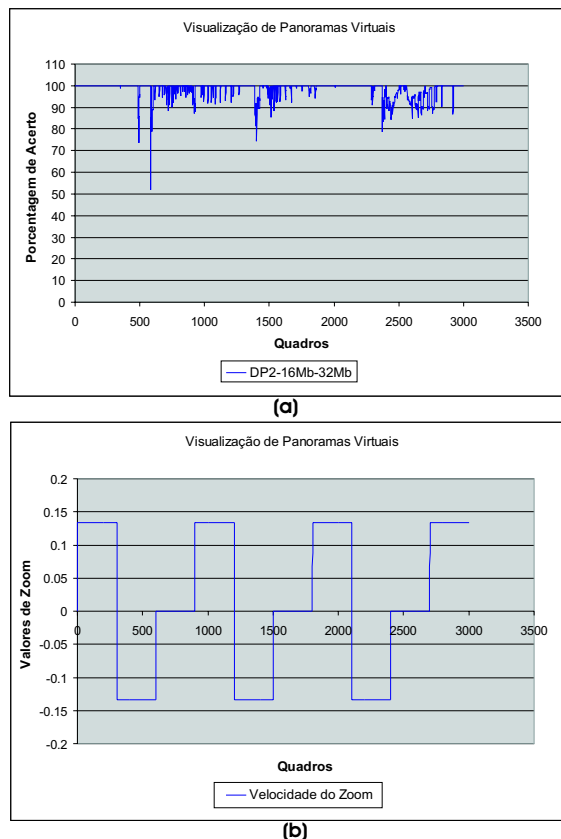


Figura 8.22: Os graficos mostram a relação entre o desempenho do sistema de predição e a variação dos parâmetros de zoom; (a) Gráfico que mostra a porcentagem de acerto para cada quadro; (b) Gráfico da Velocidade do parâmetro de zoom em cada quadro.

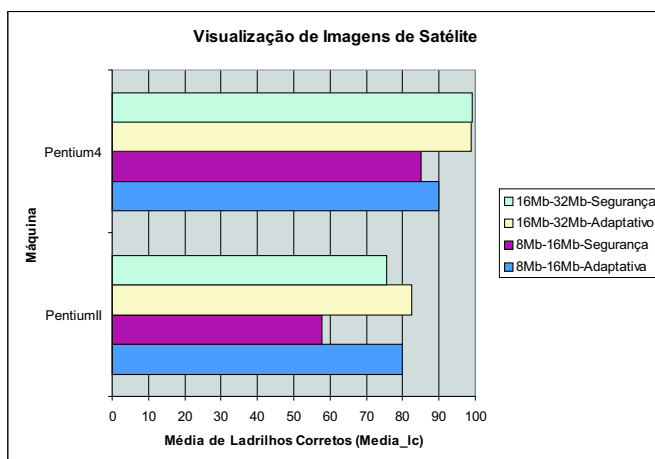


Figura 8.23: O gráfico mostra a média de ladrilhos corretos por quadro obtida utilizando a predição adaptativa e a predição por área de segurança.

carregar uma pequena região na resolução correta, enquanto o restante fica em resolução muito baixa. Outro fator agravante é o tempo gasto trazendo ladrilhos que certamente não serão utilizados pela aplicação.

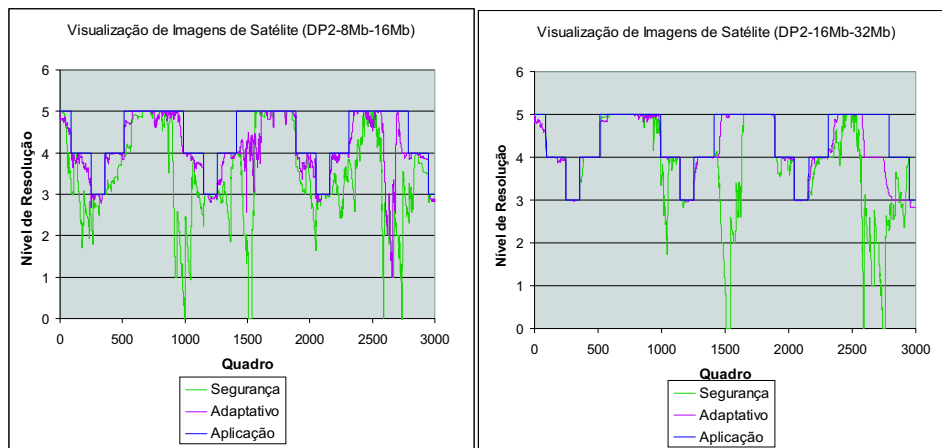


Figura 8.24: O gráfico azul mostra o nível de resolução requerido pela aplicação. Os outros dois gráficos representam o nível de resolução real do quadro que foi sintetizado utilizando a predição adaptativa e a predição por área de segurança; Os gráficos mostram os resultados obtidos no Dual-PentiumII.

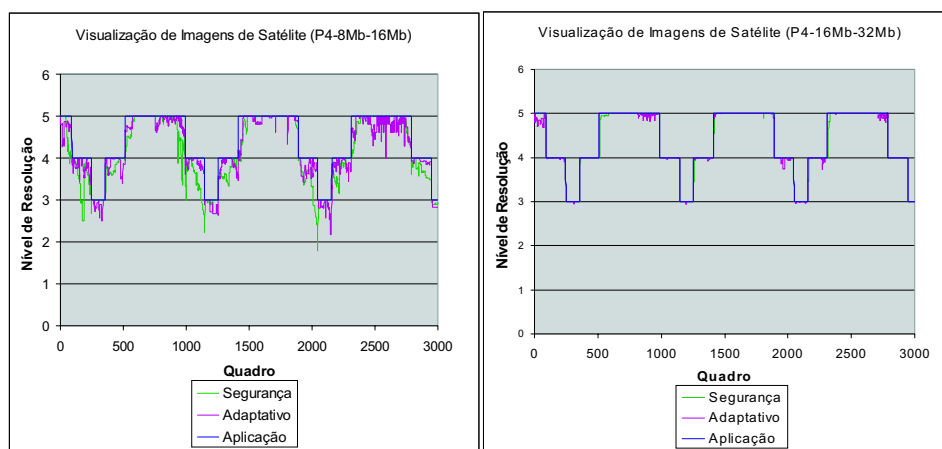


Figura 8.25: A mesma análise de desempenho dos algoritmos de predição adaptativa e por área de segurança realizada no Pentium4.

As Figuras 8.24 e 8.25 mostram os gráficos do nível de resolução requerido pela aplicação para sintetizar o quadro e o nível médio que foi realmente utilizado pela aplicação. Observe que na máquina com menor poder de processamento a diferença entre o nível de resolução requerido e o sintetizado é grande quando é utilizada a predição por área de segurança. O

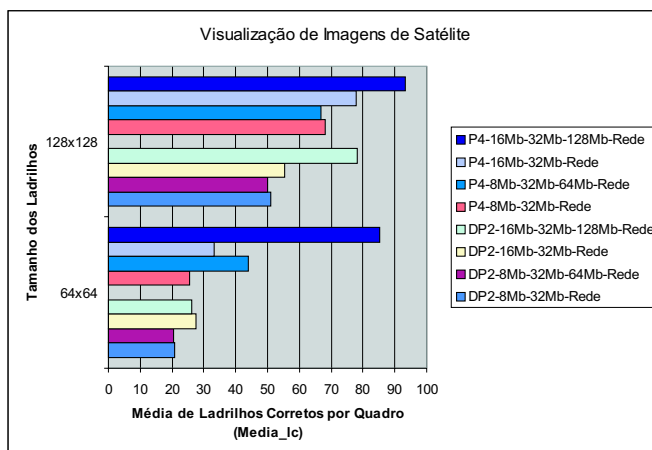
nosso sistema de predição permite que a aplicação sintetize os quadros com a resolução bem próxima da ideal. O algoritmo adaptativo carrega os níveis de resolução por camada, sempre respeitando os limites do hardware. A consequência disto é que variação de resolução entre quadros consecutivos é muito pequena. Além disso, nas situações onde a configuração de hardware era boa, o algoritmo adaptativo obteve resultados parecidos aos obtidos pela predição por área de segurança. Essa constância de resolução entre os quadros proporciona uma qualidade visual muito boa para a simulação.

8.2.6

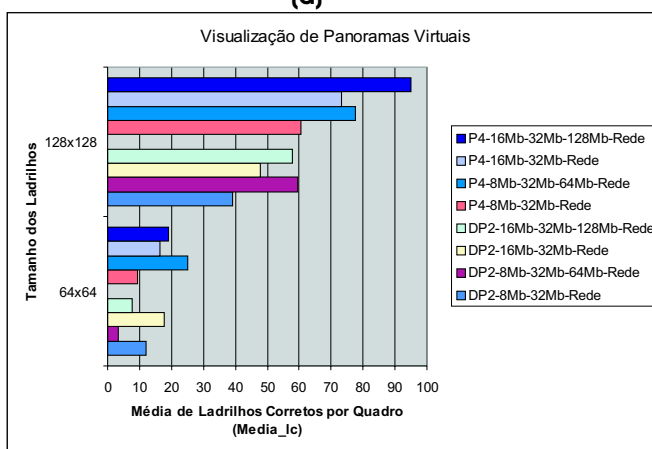
Análise com Armazenamento Remoto

No caso de armazenamento remoto, o sistema permite que o estágio de disco seja um dispositivo opcional na hierarquia de armazenamento. Os experimentos mostraram que a utilização do estágio de disco é apenas vantajosa quando a taxa de transmissão da rede é muito inferior a taxa do disco. A Figura 8.26 ilustra dos gráficos de desempenho das aplicações utilizando e não utilizando o estágio de disco. De forma geral, o desempenho da aplicação foi menor ao utilizar o estágio de disco. Este fato foi bem nítido nos experimentos com ladrilhos de 64×64 . Note também que ladrilhos de 128×128 proporcionaram um grande aumento de desempenho com a utilização do estágio de rede.

O dispositivo de disco é um estágio intermediário da seqüência de armazenamento, isto implica que são realizadas operações de escrita e de leitura. As operações de escrita transferem os dados vindos da rede para o disco e as operações de leitura transferem esses dados para a memória RAM. Essas duas operações são realizadas de forma concorrente pelo sistema operacional. O estágio de rede só realiza a operação leitura. Para obter um bom desempenho as duas operações realizadas pelo disco devem ser mais rápidas do que a operação de leitura realizada pelo estágio de rede. Os fatores que influenciam o desempenho do estágio de disco são o tamanho do ladrilho e a quantidade de memória reservada. Quanto menor o tamanho do ladrilho e o espaço reservado, maior o número de operações de escrita/leitura e menor o desempenho do estágio. Nos experimentos com 64Mb de espaço em disco são realizadas muitas operações de escrita e leitura. Neste caso o desempenho das aplicações foi bem menor com a utilização desse estágio. Nos experimentos com 128Mb de espaço em disco a quantidade de operações de escrita diminuiu (menos acessos a rede) e a aplicação obteve um melhor desempenho. É claro que essa situação só ocorreu porque os experimentos



(a)



(b)

Figura 8.26: O gráfico mostra a porcentagem média de ladrilhos corretos por quadro ($Media_{ic}$) com os dados armazenados remotamente; (a) Aplicação de imagens de satélite; (b) Aplicação de panoramas virtuais. A Ordem a legenda é Máquina-Memória de Textura-Memória RAM-Memória de Disco-Rede ou Máquina-Memória de Textura-Memória RAM-Rede, onde $DP2$ representa Dual-Pentium2 e $P4$ representa Pentium4.

foram realizados numa rede interna e com altas taxas de transmissão. Certamente, em redes mais lentas a utilização do estágio de disco seria indispensável para um bom desempenho das duas aplicações.

O sistema de predição teve mais dificuldade em estimar corretamente os ladrilhos que deveriam ser carregados. Além do problema de descontinuidade de movimento discutido na seção anterior, existe o problema do erro cometido ao calcular a taxa de ladrilhos por segundo. Com a adição do estágio de rede essa taxa pode variar bastante. O tempo de um ladrilho chegar ao primeiro estágio de armazenamento é bem menor se eles já estiverem armazenados em disco do que se não estiverem. O problema ocorre

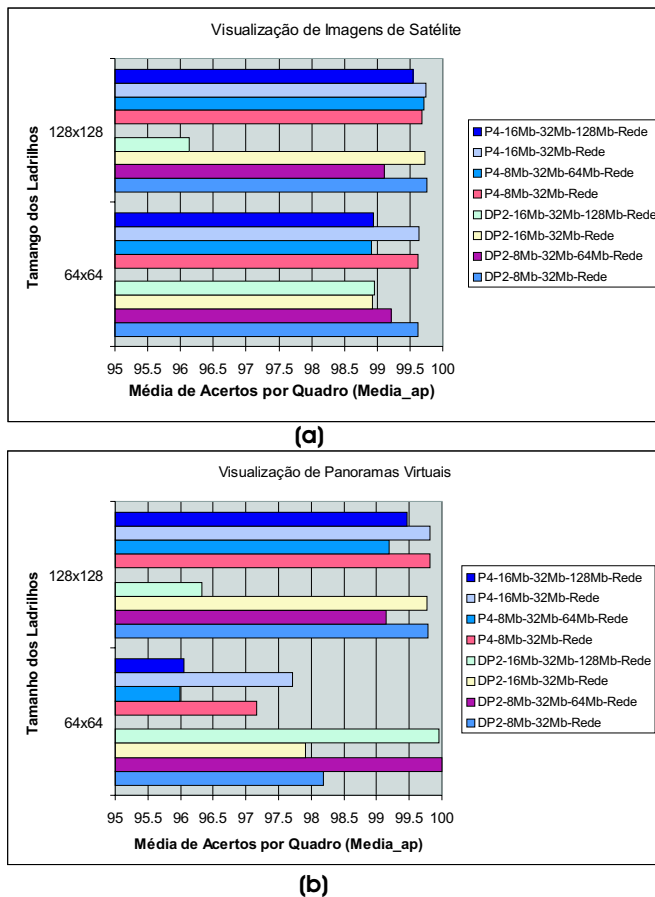
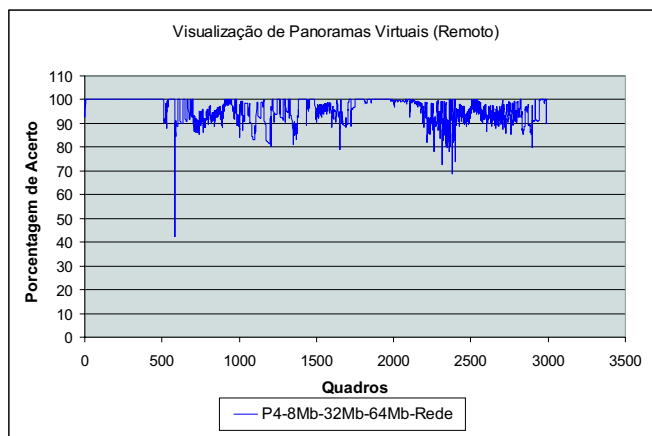


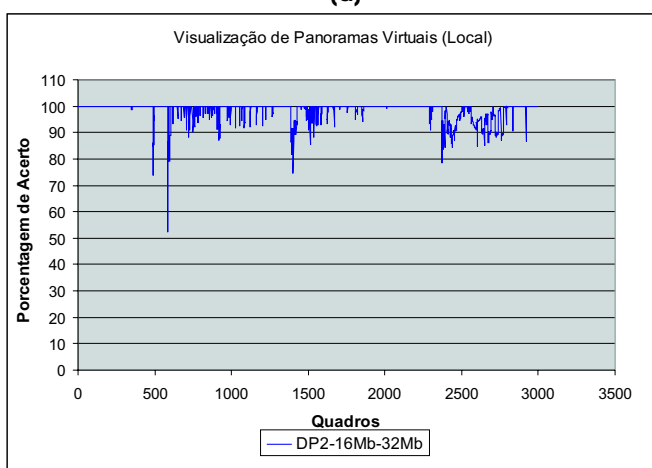
Figura 8.27: O gráfico da porcentagem da média de acerto por quadro; (a) Aplicação que visualiza imagens de satélite; (b) Aplicação que visualiza panoramas virtuais.

quando a predição do nível de resolução é baseada nas taxas calculadas com os ladrilhos em disco. Se os novos ladrilhos pedidos não estão armazenados no disco, então será gasto muito mais tempo do que o previsto para que eles cheguem ao primeiro estágio. Assim, muitos desses ladrilhos pedidos serão cancelados. A Figura 8.27 mostra a média de acerto do sistema de predição.

Observe que mesmo com todos os problemas o sistema teve mais de 95% de média de acerto para o pior caso registrado. A Figura 8.28(a) mostra o gráfico da porcentagem de acerto do sistema de predição em cada quadro. Comparando com o gráfico da Figura 8.28(b) percebe-se que os picos de menor acerto do sistema são exatamente no mesmo local devido ao problema de descontinuidade de movimento. No entanto, no gráfico da Figura 8.28(a) esses picos são maiores devido ao problema do erro na estimativa da taxa de transferência.



(a)



(b)

Figura 8.28: Os gráficos mostram a porcentagem de acerto por quadro do sistema de predição na aplicação de visualização com dados armazenados local e remotamente; (a) Visualização de panoramas virtuais com armazenamento remoto; (b) Visualização de panorama virtual com armazenamento local.

Os experimentos de comparação mostraram que a predição por área de segurança é inadequada para a visualização de dados armazenados remotamente. Nesta situação a velocidade de carregamento dos ladrilhos é ainda menor. Isto faz com que o tamanho a região com ladrilhos na resolução ótima seja muito pequena. Em casos extremos como no Dual-PentiumII, o aumento do espaço nos dispositivos de armazenamento não teve efeito algum no desempenho da aplicação. A Figura 8.29 ilustra esta situação. Isto mostra que o sistema de predição por área de segurança só deve ser utilizado em situações onde os dados estejam armazenados localmente e onde a hierarquia de armazenamento seja composta por dispositivos rápidos.

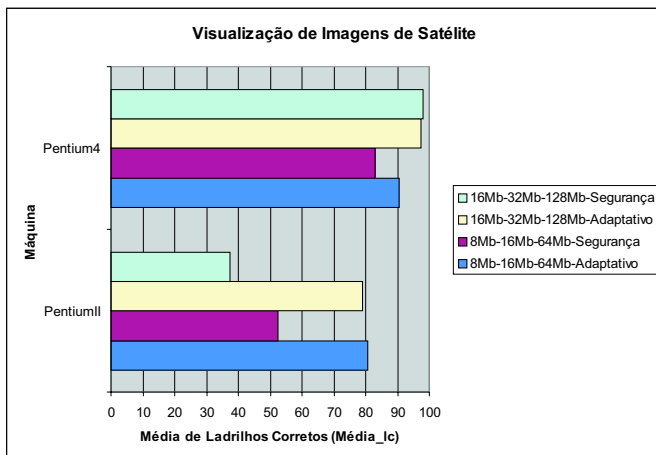


Figura 8.29: A porcentagem média de ladrilhos corretos por quadro obtida com a utilização da predição adaptativa e da predição por área de segurança.

A aplicação que utilizou o sistema de predição adaptativo teve um desempenho muito superior. Essa superioridade ficou muito evidente na máquina que possuía pouco poder de processamento. A diferença de desempenho é observada nos gráficos de nível de resolução dos quadros sintetizados (Figura 8.30). A aplicação conseguiu sintetizar os quadros numa resolução melhor quando foi utilizado o nosso sistema de predição.

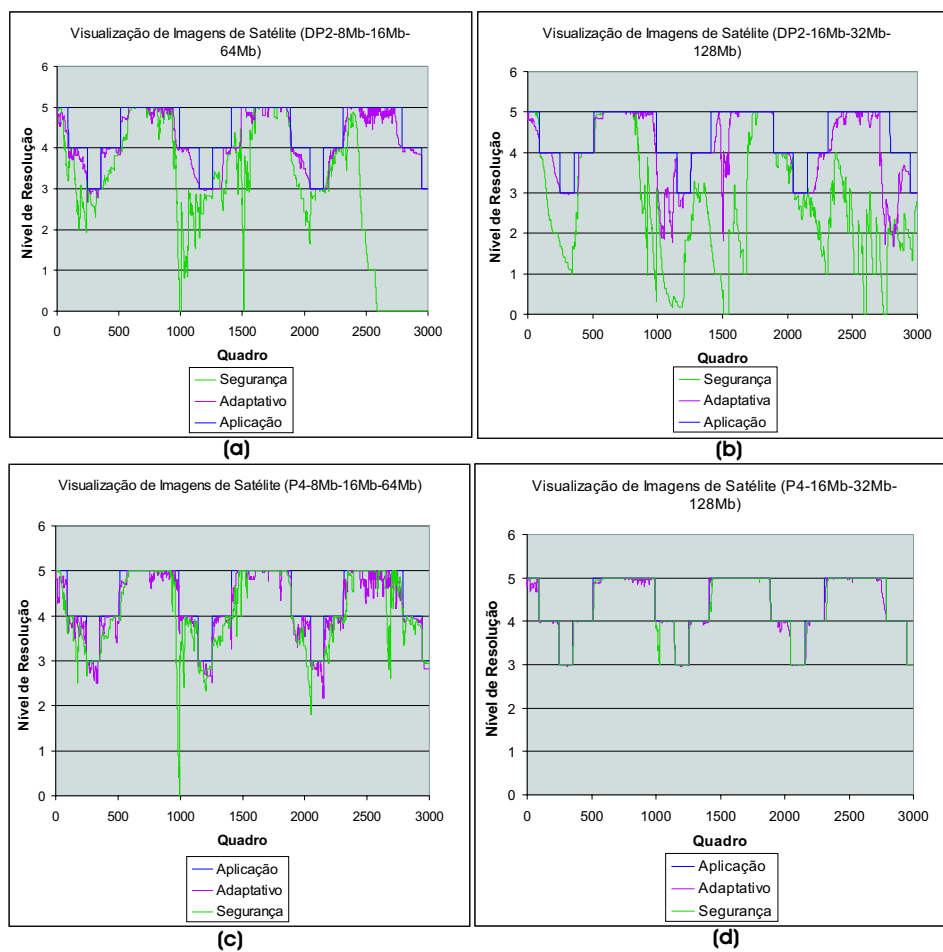


Figura 8.30: O gráfico azul mostra o nível de resolução requerido pela aplicação. Os outros dois gráficos representam o nível de resolução real do quadro que foi sintetizado utilizando a predição adaptativa e a predição por área de segurança; (a) e (b) resultados obtidos no Dual-PentiumII; (c) e (d) resultados obtidos no Pentium4.