

VISUALIZAÇÃO TRIDIMENSIONAL DO TEATRO DE OPERAÇÕES

Flávio L. de Mello¹
Edilberto Strauss²
Antônio Oliveira³

Resumo: Este artigo apresenta um conjunto de técnicas implementadas para o apoio à tomada de decisões estratégicas aplicadas a atividades de Comando e Controle com ênfase em questões relacionadas com visualização tridimensional do campo de batalha. O sistema possui a funcionalidade de Controle através de um monitoramento unificado do Teatro de Operações e do combate, usando técnicas de modelagem de terrenos, criando assim um ambiente virtual que representa o conflito. As funcionalidades referentes ao Comando estão associadas a procedimentos logísticos implementados a partir da doutrina estabelecida pela Escola de Comando e Estado Maior do Exército (ECEME).

Palavras-Chave: Visualização 3D, Simplificação de Malhas, Mapas, Comando e Controle.

Abstract: This paper describes a decision support system, and its implemented techniques, dedicated to Command and Control activities. It emphasizes the battlefield visualization by creating a virtual three-dimensional environment. The system presents the control functionality by creating a unified Operations Scenario snapshot, that uses terrain modeling techniques, and creates the conflict virtual environment. The command functionalities are associated to logistic procedures implemented according to the Escola de Comando e Estado Maior do Exército (ECEME) doctrine.

Key words: 3D Visualization, Mesh Simplification, Maps, Command and Control.

¹ Instituto de Pesquisas e Desenvolvimento – IPD/EB

² Departamento de Engenharia Elétrica – DEL/POLI/UFRJ

³ Laboratório de Computação Gráfica – LCG/COPPE/UFRJ

1. INTRODUÇÃO

Durante o decorrer de uma batalha, um comandante, ou chefe militar, deve analisar e compreender não somente o cenário do combate atual, mas também a dinâmica das possíveis variações das operações de combate. Este problema, tão antigo quanto à própria guerra, é chamado de Comando e Controle, e amplamente conhecido como C2. Na execução de suas atribuições, um comandante deve planejar e avaliar os possíveis deslocamentos estratégicos de todas as forças de combate: uma operação conhecida como Estudo e Planejamento. O conflito moderno caracteriza-se principalmente pelo dinamismo na geração de informações cada vez mais complexas, requerendo sistemas computacionais cada vez mais eficientes em transmissão, armazenamento e interpretação dos dados. O presente trabalho descreve as técnicas implementadas durante o desenvolvimento de um sistema para a simulação da ação militar de combate. O Sistema implementado foi batizado de *Tesamará*, do Guarani, significando "visualização da guerra". O *Tesamará* é uma ferramenta de apoio à tomada de decisões em procedimentos de combate, possuindo a funcionalidade de visualização tridimensional do campo de batalha a partir de um processo de modelagem de terrenos usando informações cartográficas baseadas na variação altimétrica, isto é, na variação de altura entre dois pontos do terreno. Toda a informação gerada e manipulada pelo *Tesamará* está de acordo com os atuais padrões militares preconizados pelo Exército Brasileiro.

Atualmente no Exército Brasileiro, assim como em outras Forças Armadas no mundo, o estudo do combate é baseado em mapas de papel que descrevem cartograficamente a área do conflito, ou região de interesse. Estes mapas são cobertos por folhas de acetato que facilitam a marcação das atividades de combate associadas às tropas militares em operação.

O conflito moderno é dinâmico, produzindo diversas informações em tempo crítico, o que torna as técnicas manuais inadequadas para visualizar e compreender o campo de batalha em tempo real. Os Oficiais de planejamento recebem mensagens textuais provenientes do campo de batalha, e as representam graficamente sobre o acetato. Os Comandantes, por sua vez, estudam as possíveis variações dos cenários realizando inúmeras outras representações sobre a folha de acetato, tornando sua compreensão difícil e tediosa.

Portanto, o Estudo de uma operação é um processo desgastante e extremamente lento, principalmente porque os mapas detalhados, e as folhas de acetato, podem levar diversas horas para serem copiados e devidamente distribuídos ao pessoal no campo de batalha.

Através da visualização 3D do campo de batalha [1], um comandante pode planejar mais eficientemente o deslocamento de sua tropa, as manobras de ataque, as atividades de infiltração no terreno, as operações defensivas e os pontos de fixação para conter o avanço inimigo. Por exemplo, os Oficiais responsáveis pelas comunicações podem planejar mais facilmente a posição de suas estações repetidoras, bem como o melhor posicionamento dos postos rádio. Os Oficiais de Artilharia, por sua vez, podem melhor prever o posicionamento de suas baterias (operadores dos morteiros e canhões) aumentando assim sua proteção contra o fogo inimigo, sem prejudicar sua capacidade de ataque.

O objetivo do trabalho [2] foi a apresentação das técnicas de modelagem e visualização cuja implementação deu origem ao Sistema *Tesamará*. O sistema possui funcionalidades que minimizam os problemas relacionados ao estudo de uma Operação de Combate.

O Teatro de Operações é o cenário composto pela porção de terreno na qual se desenvolve o combate, e pelas tropas militares que se encontram sobre o mesmo terreno. Assim, a capacidade de visualizar o Teatro de Operações em toda

a sua magnitude é essencial para uma visão sistêmica do conflito. Ao se realizar o planejamento baseado em um ambiente tridimensional (3D), o Comandante pode melhor observar e compreender questões relacionadas com o perfil do terreno e seus obstáculos naturais, os quais muitas vezes não são evidenciadas em mapas em papel, essencialmente bidimensionais.

Note que a posição das unidades de combate pode ser facilmente representada por um conjunto reduzido de dados do tipo textual. Desta forma, a situação do cenário de batalha, bem como o seu planejamento, não precisam ser digitalizados através de *scanners*, bastando uma parametrização dos desenhos dos acetatos. A parametrização diminui o volume de dados a serem transmitidos, os que viabiliza sua transmissão através de rádios capazes de cobrir as dimensões continentais do Brasil.

2. SISTEMAS DE COMANDO E CONTROLE

O Comando e Controle (C2) é o processo pelo qual comandantes militares e gerentes civis exercitam sua autoridade e gestão sobre recursos disponíveis, sejam materiais ou humanos, para alcançar objetivos táticos e estratégicos [3].

O C2 consiste no gerenciamento de uma série de facilidades ligadas à área de: comunicação; controle de pessoal e de equipamento; procedimentos para a monitoração; previsão logística de atividades; planejamento de operações; alocação de recursos; etc. Note que em organizações empresariais este processo é denominado de Gerenciamento Operacional de Negócios [3]. Ou seja, o C2 pode ser visto como um processo gerencial de negócios focado para situações de guerra.

A atividade de Comando e Controle destina-se a permitir que as autoridades elaborem planejamentos operacionais, a partir de uma gestão eficiente de seus recursos disponíveis [3]. Desta forma, o C2 é um sistema único e integrado, adaptado aos processos

relacionados com: a doutrina militar; os procedimentos operacionais; a estrutura organizacional e pessoal; os equipamentos e facilidades de comunicações.

Sistemas civis e militares de C2 são similares em seus requisitos. O sucesso desses sistemas depende da produção e execução de decisões oportunas, com informação acurada e precisa. Fazendo uma analogia a sistemas de gestão na indústria, gerentes e líderes de corporação identificam objetivos de mercado e mobilizam os seus recursos para alcançá-los. No campo militar, chefes militares planejam e executam longas e complicadas operações para cumprirem a sua missão.

Note que o levantamento de requisitos do usuário é a chave para um projeto bem sucedido de qualquer Sistema de Informação, assim como para sistemas de C2. Neste sentido, o levantamento de requisitos é normalmente mais fácil na área militar uma vez que a maior parte das atividades militares está bem normatizada em termos de níveis de comando, de conflito e de missões.

Os Jogos de Guerra são um conjunto de tarefas e procedimentos, que em tempo de paz, permitem os comandantes militares e os gerentes civis exercitarem suas atividades de C2, sem no entanto, dispor de pessoal em exercício de campanha (simulação de guerra). Assim, é possível realizar um treinamento com toda a riqueza tática e complexidade das tomadas de decisões estratégicas reais, porém com um baixo custo operacional.

2.1 Os Principais Sistemas de C2

Poucos são os artigos e teses descrevendo técnicas computacionais que combinam a visualização e a simulação de ações militares. A maior parte das pesquisas é mantida pelo governo dos EUA, o qual gasta cerca de US\$ 4 bilhões por ano, em simulação computacional para o treinamento de suas Forças Armadas [4]. Nenhum outro país investe

tanto nesta área. Os principais sistemas são descritos a seguir:

Joint Simulation System - JSIMS

O JSIMS é um projeto das Forças Armadas dos EUA que tem o objetivo de unificar os sistemas informatizados de C2 já existentes na Marinha, no Exército e na Aeronáutica. O Sistema de Posicionamento Global (GPS), por exemplo, é o resultado bem sucedido de um projeto similar de unificação. O JSIMS engloba funcionalidades táticas, operacionais e estratégicas de combate. Seu primeiro teste de larga escala estava programado para o verão de 2002, durante um exercício chamado "Millennium Challenge", porém foi suspenso e não há informes sobre uma nova data para sua avaliação. O JSIMS é um sistema cujo módulo de visualização é bidimensional. Ele modela, por exemplo, toda uma companhia de blindados através de uma única entidade gráfica, não havendo a possibilidade de representação de frações de unidades. As bases de dados de terreno possuem uma baixa resolução, isto é, cada ponto que compõe a imagem digital do mapa equivale a cerca de 80 m² do terreno. Esta resolução não permite sua utilização em estudos e planejamentos que necessitem alto grau de precisão.

Warfighter's Simulation - WARSIM

O Sistema WARSIM, ainda em desenvolvimento pelo *Program Executive Office, Simulation, Training and Instrumentation* dos EUA, será um sistema de simulação e treinamento computadorizado, e que será incorporado ao JSIMS. O WARSIM foi projetado para operar em Postos de Comando desde Batalhões até Teatros de Operações. Seu projeto teve início em outubro de 1998, em parceria com um conglomerado de empresas. A primeira versão foi divulgada em setembro de 2000, porém pouca informação tornou-se pública, até mesmo quanto à qualidade visual do aplicativo. Sabe-se que o sistema possui módulo de realidade virtual não imersiva,

com modelos de terreno (malhas 3D) com baixa complexidade de forma.

Dragon System

Durbin et al [5] desenvolveram uma bancada de projeção de um ambiente virtual para um aplicativo chamado *Dragon System*. Este projeto de alto custo foi patrocinado pelo *US Naval Research Laboratory*. A bancada de projeção constitui-se de uma mesa de vidro capaz de representar de forma tridimensional (holograma) o campo de batalha, incluindo-se o terreno, as tropas, etc. As tropas podem ser visualizadas seja pela sua tradicional representação simbólica, adaptada a um ambiente tridimensional, seja através de modelos 3D, isto é, um regimento de cavalaria seria representado por uma miniatura de um veículo blindado. O sistema é operado por equipamento de realidade virtual imersiva (óculos, luvas), além de mouse e teclado. Após a análise dos resultados amplamente divulgados, os conceitos utilizados no Sistema *Dragon* podem ser considerados como os mais bem adaptados aos padrões militares, podendo se adequar facilmente à doutrina militar brasileira. Entretanto, o sistema não é comercializado.

Force XXI Battle Command Brigade and Below - FBCB2

O FBCB2 [6] é o principal sistema digital de Comando e Controle do Exército dos EUA no nível de Brigada. O FBCB2 fornece a situação operacional no nível tático mais baixo e permite um rápido e eficiente fluxo de informações de Comando e Controle no efetivo campo de batalha. Sabe-se que os equipamentos do sistema (*hardware*), bem como seus aplicativos (*software*), encontram-se consolidados, ainda que as informações técnicas não estejam disponíveis ao público. No entanto, um estudo sobre o FBCB2 permite inferir que ele trabalha em duas dimensões utilizando mapas *rasterizados* (imagens digitais). Devido à baixa resolução das imagens o sistema não permite um

elevado grau de precisão no que se refere à visualização do posicionamento geográfico. A análise de documentos, fotografias, e dados coletados na Internet indicam que o sistema opera em plataformas PCs comuns (Pentium II 333 MHz), os quais sofrem um processo de robustecimento, para garantir seu uso em condições adversas.

Marconi Automated Command and Control Information System - MACCIS

A empresa italiana *Marconi Communications* comercializa o sistema chamado MACCIS, que consiste em um sistema de C2 cujo módulo digital de visualização de cartas também de forma bidimensional, tendo funcionalidades similares ao FBCB2. A doutrina de emprego do sistema é baseada na doutrina italiana, e por este motivo, necessita de muitos ajustes para ser utilizada pelo Exército Brasileiro. O MACCIS funciona em uma plataforma SCO Unix, com previsão de lançamento de uma nova versão para Windows NT.

3. MODELAGEM DO TEATRO DE GUERRA

O sobrevôo de regiões geográficas tem um papel fundamental em Sistemas de Informação Geográfica baseados na tecnologia de realidade virtual. A função de sobrevôo pode ser encontrada em diversas aplicações tais como no Planejamento de Missões Militares, em Simuladores de Vôo, e etc. Em geral, a representação do terreno é feita através de modelos digitais de elevação [7].

Entretanto, a relação entre a resolução dos modelos de elevação, e seu dado associado, pode facilmente exceder a capacidade de um dispositivo gráfico, o que em muitos casos torna impossível uma aplicação em tempo real. Desta forma, os dados obtidos do modelo do terreno devem sofrer, de alguma forma, simplificações para viabilizar sua utilização.

Além disto, é necessário também levar em conta a possibilidade de realizar operações de recorte, de modo a reduzir a complexidade de uma cena. Em um sobrevôo, a cada alteração do ponto de vista, apenas parte de todo o cenário é visualizada, o que indica que os objetos não visíveis podem, e devem, deixar de ser enviados para a fila de desenhos a serem realizados pela biblioteca de manipulação de desenhos [8]. Esta biblioteca é conhecida com OpenGL.

Por fim, o Teatro de Guerra é o espaço geográfico envolvido nas operações militares, sendo composto pelo terreno propriamente dito, suas rodovias, sua hidrografia, as áreas urbanas, as condições atmosféricas as quais ele está submetido, etc. Neste capítulo serão apresentados diversos detalhes da representação do Teatro de Guerra, desde o desafio da representação computacionalmente eficiente do terreno, até a utilização de texturas para adicionar mais realismo ao aplicativo.

3.1 Modelo Digital de Elevação

A representação do relevo ou terreno é uma componente fundamental no processo cartográfico, e em áreas correlatas. Esta representação, em formato digital, recebe o nome de Modelo Digital do Terreno (MDT) e consiste em um conjunto de dados que explicitam as coordenadas (X, Y, Z) do terreno, e a forma como os mesmos estão relacionados. O MDT pode ser considerado como parte de um problema mais geral, isto é, a reconstrução de superfícies, que é atualmente um campo de estudos da Visão Computacional. O Modelo do Terreno é uma representação matemática da distribuição espacial de uma determinada característica vinculada a uma superfície real. A superfície é em geral contínua e o fenômeno que representa podem ser variados (municípios, urbanização, densidade demográficas, etc).

Um Modelo Digital de Elevação (MDE) é a representação digital de uma porção da superfície da Terra, ou

qualquer outro corpo celeste, derivada de medições de elevação coletadas em intervalos regulares horizontais. Assim, um MDE é o modo através do qual armazena-se digitalmente os contornos de um terreno qualquer. O MDE descreve o terreno por funções mono-valoradas, isto é, para cada posição planimétrica, uma, e somente uma altitude é fornecida.

Em geral, os MDEs são imagens em tons de cinza, onde cada pixel da imagem corresponde a uma posição cartesiana no terreno. Os valores de intensidade dos pixels correspondem a um dado valor de elevação. Assim os valores extremos de altitude do terreno são normalizados para um intervalo entre 0 e 255.

3.2 Junção Adaptativa

Para se calcular a malha triangular de um MDE, exige-se que a malha possua ao menos alguma das características descritas a seguir [9, 10, 11, 12, 13, 14]:

- **ser conforme**, ou seja, não é permitido a ocorrência de um triângulo que contenha, no interior de suas arestas, um vértice de outro triângulo;
- **respeitar a vizinhança**: as arestas próximas da componente de altura devem estar contidas na união das arestas da malha triangular;
- **ser bem comportada**: os ângulos da malha não devem ser nem grandes demais, nem pequenos demais. Normalmente é exigido que os ângulos variem entre 45° e 90° ;
- **ser não uniforme**: a malha deve ser refinada nas proximidades da componente de altura e menos refinada quando longe da componente.

No algoritmo proposto, foi assumido que a restrição de malhas bem comportadas pode ser relaxada. De fato, a escolha das restrições para gerar uma malha está intimamente ligada ao tipo de implementação.

O método para a geração da malha não uniforme apresentado nesta seção é baseado em uma estrutura dados

no formato de árvore quaternária. O algoritmo divide o MDE em uma malha regular, e então realiza a junção dos triângulos redundantes em triângulos maiores. A implementação preserva os dados de entrada, porém é dita não-conforme e não-comportada.

Nesta implementação, assume-se que o MDE possui um tamanho de $2^n \times 2^m$, onde n e m são inteiros positivos. O processo de geração da malha é descrito em duas etapas:

(1) O MDE é recursivamente dividido em quatro quadrantes gerando uma estrutura de dados em *quadtree*. A subdivisão recursiva acontece até que seja atingida a altura da árvore previamente estabelecida pelo usuário, e que descreve a resolução desejada pelo modelo (nível de detalhe). Uma vez que, nesta fase, nenhum método de simplificação é realizado, a árvore resultante constitui uma árvore quaternária cheia. Cada nó da árvore corresponde a uma porção quadrangular do MDE. Isto implica que as folhas desta árvore correspondem à subdivisão mais refinada do modelo de elevação. Esta subdivisão representa uma malha regular e uniforme do MDE.

(2) Unificação de pares de triângulos gerando triângulos maiores. A árvore é inicialmente percorrida em pós-ordem, porém somente suas folhas são inseridas em uma lista temporária chamada de Lista de Pares de Triângulos (LPT). A ordem de inserção na LPT é apresentada na Figura 3.1. A união dos triângulos resultantes ocorre durante a inserção na lista temporária. Ao final do processo de inserção, a LPT somente possui pares de triângulos ordenados por sua dimensão. Esta lista, ainda que temporária, representa o conjunto de triângulos que irão compor a malha otimizada.

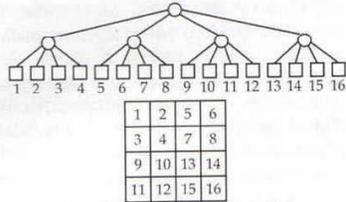


Figura 3.1: A árvore, sua subdivisão e a ordem de inserção na lista de pares de triângulos.

Antes de descrever como é feita a união dos triângulos, é importante esclarecer a ordenação da lista de pares de triângulos. Esta lista é duplamente ordenada, onde primeiramente ordena-se a mesma pela dimensão horizontal dos pares de triângulos, e posteriormente pela dimensão vertical dos mesmos (ver Figura 3.2)

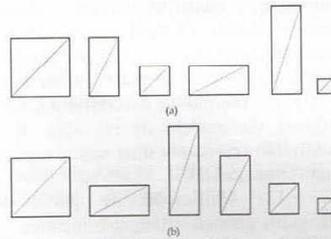


Figura 3.2: Ordenação da lista de pares de triângulos : (a) Um conjunto qualquer, não ordenado de pares de triângulos. (b) O conjunto ordenado de pares de triângulos.

Considere o par de triângulos como um nó de uma lista simplesmente encadeada. A união de nós ocorrerá somente quando as seguintes condições forem satisfeitas: (1) Os nós possuem arestas coincidentes; (2) Os nós devem ser coplanares. A Figura 3.3 ilustra os critérios apresentados.

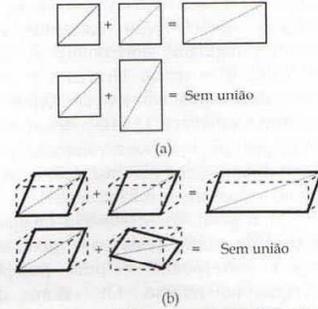


Figura 3.3: Critérios para unificação de pares de triângulos: (a) Dois quadrados devem se juntar se eles têm arestas coincidentes e com o mesmo tamanho. (b) A junção só deve ocorrer se os pares de triângulos forem coplanares.

A união somente ocorre quando os quadrados são coplanares e quando possuem arestas coincidentes. Quando os critérios de união são satisfeitos, o nó da lista é removido, e os dois nós são então unificados. É necessário ressaltar que a união de dois nós resulta em um novo nó de maior tamanho. Este novo nó poderá ser unificado com outros nós já inseridos na LPT, seja no critério de dimensão vertical, como no horizontal. Desta forma, conclui-se que várias operações de união podem ocorrer a partir de uma única inserção de um nó na lista.

Por fim, os nós da LPT são mapeados sobre uma lista de *renderização*, uma tabela de dispersão com encadeamento exterior tal como ilustrado na Figura 3.4. Esta lista é composta de 16 grandes células que abrangem toda a região do MDE. Cada célula possui uma sub-lista que controla os elementos nele inseridos. Para cada par de triângulos da LPT é determinado em qual bloco está contido. Uma vez determinado o bloco, realiza-se sua remoção da LPT, e sua inserção na sub-lista referente ao bloco. Este procedimento é bastante similar ao efetuado por Hoppe [10]. Essa mudança na ordenação dos dados permite agilizar o *frustum culling*, que será posteriormente detalhado.

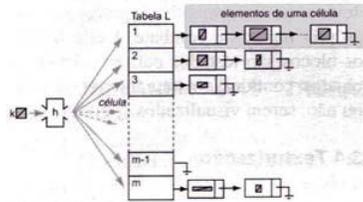


Figura 3.4: Esquema de uma tabela de dispersão com encadeamento.

3.3 Frustum Culling

Em uma navegação sobre um terreno virtual, a cada alteração do eixo de visão do observador, apenas uma porção de todo o cenário é visualizada, o que indica que os objetos não visíveis podem, e devem, deixar de ser enviados à placa de vídeo para processamento. Desta forma, a placa de vídeo não deve desperdiçar seus exíguos recursos computacionais desenhando objetos que efetivamente não são visíveis naquele momento. Tipicamente, os objetos que estão à frente do observador devem ser enviados para a placa de vídeo a fim de que sejam desenhados. Esta operação de recorte é conhecida como *frustum culling*. Os objetos fora do volume de visão não precisam ser enviados para a placa de vídeo uma vez que não existe a possibilidade de estarem visíveis naquele instante.

A biblioteca OpenGL já realiza estas operações de recorte, porém seu funcionamento só é satisfatório quando existem poucos objetos. Quando a quantidade de objetos aumenta, seja pelo incremento do número de tropas, ou seja, pelo aumento da resolução da malha do terreno, observa-se uma queda acentuada da taxa de quadros desenhados em função do tempo.

É notório que apenas uma fração dos objetos do ambiente virtual está efetivamente visível em determinado momento. Se fosse possível determinar quando um objeto está visível antes dele ser desenhado, seria possível

simplesmente ignorá-lo, evitando assim o processamento de dados desnecessários para a fila de tarefas a serem executadas pela biblioteca de funções OpenGL.

Observe a Figura 3.5. A malha poligonal representa quadriculas cartográficas de uma região qualquer. Em cada polígono estão contidos: os triângulos da malha do terreno; os símbolos de tropas que estão desdobradas; etc. Baseado no posicionamento de câmera, definindo o campo de visão de um observador, conclui-se que apenas os objetos contidos nos polígonos cinza precisam ser desenhados. O restante pode ser ignorado.

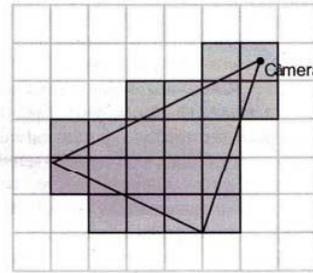


Figura 3.5: Os blocos do terreno que estão contidos, completamente ou parcialmente, no *frustum* são representados em cinza.

A primeira tarefa a ser realizada é a determinação dos seis planos que definem um *frustum*. Esta tarefa necessita ser computada a cada *frame renderizado*. Os planos descritos como positivos são aqueles ditos interiores à região de *frustum*. Os classificados como negativos são planos externos. A definição dos planos é processada após a atualização do volume de visão, e antes da *renderização* dos objetos.

Em seguida é preciso determinar se um polígono encontra-se no interior do volume. É dito que um ponto está dentro de um volume se ele sempre está em frente aos planos positivos da região de *frustum*.

Todavia, em grande parte dos casos não se deseja testar apenas um

ponto, mas sim o modelo que descreve um objeto (vários pontos), tal como mencionado por Morley [15]. Conforme a complexidade da representação do modelo, o número de pontos pode crescer demasiadamente aumentando significativamente o custo computacional.

Uma solução admissível é a imersão do objeto em um volume envolvente, tal como uma esfera. O processo de recorte baseia-se em determinar se o volume envolvente encontra-se interior a região do *frustum* (tronco de pirâmide). O objeto somente será *renderizado* quando seu volume envolvente for dito interior. Somente então os pontos que descrevem o modelo são processados para garantir a visualização otimizada do objeto.

A esfera é um exemplo de volume envolvente. Um outro volume bastante usual é a caixa. Qualquer outro tipo de volume pode ser utilizado, porém esferas e caixas são os mais comuns de serem utilizados devido à simplificação dos procedimentos matemáticos envolvidos na operação destas geometrias.

O teste para verificar se uma esfera encontra-se dentro do *frustum* é bastante similar ao teste do ponto. Sua única diferença é que, ao invés de testar se o volume está em uma região positiva, realiza-se o teste para avaliar se o centro desta esfera está distante dos planos positivos de *frustum* a uma distância maior que seu raio. O teste da esfera funciona até mesmo quando ela engloba completamente o *frustum*.

O teste de recorte baseado no volume envolvente na forma de caixa não é tão imediato. Neste caso torna-se necessário testar todos as oito quinas da caixa, e ainda, garantir que a caixa envolvente não é grande o suficiente para envolver completamente a região de *frustum*.

Durante o processo de *frustum culling*, o sistema implementado faz uso de caixas envolventes para cada um dos blocos que compõe a lista de *renderização*. Apesar dos testes com caixas envolventes não serem tão imediatos quanto os com esferas

envolventes, optou-se por utilizar caixas devido às características poligonais da própria região de combate. Desta forma, os blocos são testados para decidir se os objetos contidos em seu interior devem, ou não, serem visualizados.

3.4 Texturização

Objetivando um aumento da sensação de imersão em uma região de combate, após o processo de modelagem do terreno, executa-se a texturização da superfície do modelo. O processo de texturização define um procedimento de mapeamento da imagem digital da carta topográfica que originou o modelo sobre a superfície da malha de triângulos. A texturização permitirá disponibilizar todas as informações contidas na cartas tais como a vegetação, a hidrografia, a malha rodoviária, e etc. Esta etapa é ilustrada pela Figura 3.6.



Figura 3.6: Recobrimento da malha do terreno: (a) a malha de um terreno qualquer; (b) a textura que será aplicada sobre o terreno, uma imagem em pseudo-cor do MDE do terreno; (c) representação final da malha com a textura aplicada sobre a mesma.

De forma similar à estrutura de dados que define o modelo do terreno, o processo de texturização baseia-se na subdivisão em blocos. Cada bloco de textura é mapeado sobre o seu bloco correspondente na malha. Somente os blocos de textura visíveis dentro do *frustum* são carregados em memória [16, 17].

A discretização da textura em blocos é importante no desempenho do sistema. Tipicamente, uma carta em papel após ser digitalizada com 100 dpi, ocupa um espaço de 27MB de memória. Particionando a textura em blocos, pode-se reduzir significativamente o volume de

dados *on line* no processo de visualização em tempo real.

4. MODELAGEM DAS TROPAS MILITARES

O esquema de representação das tropas, ou unidades, militares desdobradas no campo de batalha foi baseado no manual C21-30 [18]. Este manual padroniza, entre outros assuntos, a simbologia brasileira para representação de tropas, tais como brigadas e batalhões. Este manual é voltado para a representação bidimensional e analógica dos símbolos cartográficos. Por este motivo, diversas adaptações foram realizadas a fim de que seus conceitos fossem preservados em um ambiente tridimensional e computacional.

Basicamente o C21-30 formaliza uma série de propriedades que descrevem uma tropa, tais como sua subordinação, seu escalão dentro da cadeia de comando, sua especialidade, etc. Em uma análise inicial, esta representação pode parecer um tanto confusa, porém ela é na verdade intuitiva e rica em detalhes. Com um pouco de contato com o trabalho diário dentro das Forças Armadas, o usuário leigo passa a identificar facilmente representações estilizadas de símbolos do seu dia a dia.

Com o objetivo de atingir uma melhor didática, convém estudar a representação das tropas, primeiro de forma bidimensional, para em seguida, extrapolar seu entendimento para a terceira dimensão. Assim a composição dos símbolos pode ser realizada através das seis padronizações enumeradas a seguir, e ilustrada através da Figura 4.1:

a. **Símbolo básico:** indica a categoria do elemento representado: se é realmente uma tropa; se é apenas um posto de comando; se é um observatório destacado no terreno; ou ainda, uma instalação tal como um depósito ou uma estação de rádio. *Ex:* Unidade, Posto de Comando, Instalação, Observatório;

- b. **Símbolo de especialidade:** identifica a Força Armada, a Arma, o Serviço, a Especialidade ou a Atividade, isto é, representa a natureza do elemento, e é inserido no interior do símbolo básico. *Ex:* Infantaria, Cavalaria, Engenharia, Comunicações;
- c. **Símbolo de escalão:** colocado sobre o símbolo básico, representando o efetivo numérico de uma determinada unidade, ou ainda, o efetivo estimado. *Ex:* Brigada (2500 homens), Divisão (5640 homens), Companhia (120 homens), Batalhão (600 homens);
- d. **Símbolo de designação:** é o identificador numérico, ou seja, o nome que identifica a tropa;
- e. **Símbolo de subordinação:** representa a unidade sob a qual o elemento está subordinado;
- f. **Outras informações:** é um campo destinado a indicar outras informações necessárias para complementar a definição do elemento, tais como, sua natureza, seu calibre, seu armamento principal, seu meio de transporte, etc. *Ex:* Leve, Motorizada, Blindada, Pára-Quedista.

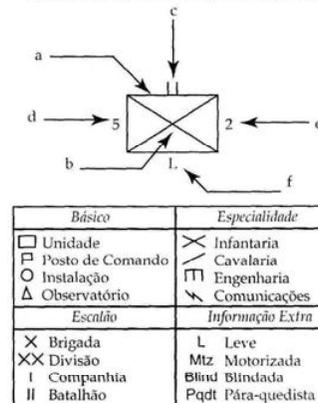


Figura 4.1: Composição dos símbolos de unidade e alguns exemplos: (a) símbolo básico; (b) símbolo de especialidade; (c) símbolo de escalão; (d) símbolo de designação; (e) símbolo de subordinação; (f) área para inclusão de outras informações relevantes.

Esta representação bidimensional precisou ser adaptada para uma representação 3D uma vez que é necessário visualizar as tropas através de ângulos oblíquos. Desta forma, os símbolos bidimensionais foram transformados em cubos, pirâmides e esferas. A Figura 4.2 ilustra o conceito de composição de símbolos. O grande "x" dentro da caixa representa a especialidade de infantaria, explicitando que a tropa é composta por militares que se deslocam a pé, e ainda, armados com armas de combate aproximado, aptos a operar em qualquer tipo de terreno e sob quaisquer condições de tempo e visibilidade. O número "2" a esquerda indica a designação, e o número "1" a direita indica a Unidade Militar a qual este elemento encontra-se subordinado. O "x" sobre a caixa representa o indicativo de brigada, sinalizando que a tropa deve possuir cerca de 1.800 homens. Finalmente, o símbolo seria lido como: a 2ª Brigada de Infantaria da 1ª Divisão de Exército.

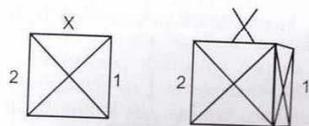


Figura 4.2: Exemplo de símbolos de unidade em duas e três dimensões: (a) A 2ª Brigada de Infantaria da 1ª Divisão de Exército.

Infelizmente, todas as representações ao redor do objeto tridimensional da tropa são compostas por texto. Uma vez que os caracteres são de natureza tipicamente bidimensional, surge o problema de como representá-los em um espaço de três dimensões. Suponha que a câmera está voltada para a tropa e que o texto está voltado para a câmera. Neste caso, o comandante observaria o símbolo da tropa exatamente como deveria ser. Agora, imagine que a câmera começa a se deslocar para uma nova posição. O texto continua voltado para sua posição original, e à medida que o comandante se desloca, ele passa a

observar a tropa através de um ângulo qualquer. O texto começará a se estreitar, e quando a câmera atingir uma posição perpendicular a sua posição original, o texto irá desaparecer. Assim deve-se tentar evitar situações em que o texto fique ocluso para uma dada posição de câmera.

A solução encontrada foi forçar que o texto ficasse sempre voltado para a câmera, uma técnica conhecida como *billboarding* [19]. Desta forma, o texto ao redor do símbolo de unidade é alinhado de tal modo que, a normal ao seu plano esteja paralela e em direção oposta ao eixo de visão da câmera [20]. Para efetuar esta operação, é apenas necessário reverter às rotações realizadas pela câmera, isto é aplicar a matriz inversa de rotação sobre o modelo da tropa.

5. RESULTADOS

O protótipo foi testado em laboratório, sendo o algoritmo de geração das malha do terreno a primeira característica analisada. Realizou-se uma comparação direta entre o algoritmo de Röttger e o algoritmo proposto. A implementação de Röttger encontra-se disponibilizada pelo próprio autor em [21]. O modelo digital de elevação é o mesmo utilizado em ambos os algoritmos, bem como a textura utilizada para o recobrimento da superfície. A implementação de Röttger foi adaptada para que utilizasse a mesma altura de *quadtree* do algoritmo proposto. Na implementação original de Röttger a altura da *quadtree* encontra-se estaticamente definida no código fonte do programa (*hard-coded*).

Ambas as implementações utilizaram como plataforma um Pentium III 1GHz, com 256 MB de memória RAM, e uma placa NVIDIA GeForce II MX 400 AGP. Foi computada a quantidade de triângulos manipulada por ambas implementações e constatou-se que o algoritmo proposto utilizou 8.738% menos triângulos que o algoritmo de Röttger. Este resultado foi relatado [22] no SIBGRAPI 2002.

Realizou-se também uma simulação de sobrevôo em ambas implementações, computando o valor médio de quadros renderizados por segundo (*frames per second* – fps). Cada quadro representa uma imagem desenhada na tela do computador. Assim, quanto maior for a quantidade de quadros desenhados por segundo, melhor será a sensação de fluidez da simulação de sobrevôo sobre o terreno. O algoritmo proposto atingiu o valor de 98,1 fps, enquanto que a implementação de Röttger atingiu 95,1 fps, um ganho de 3,1% sobre este último.

O modelo digital de elevação utilizado no teste encontra-se apresentado na Figura 5.1a. Esta figura representa uma parcela significativa das Zonas Sul e Norte do Município do Rio de Janeiro. A Baía de Guanabara encontra-se à direita da imagem enquanto que o Oceano Atlântico encontra-se na parte inferior da mesma. A imagem possui 1024x1024 pixels, onde cada pixel representa 222 metros lineares no terreno. As imagens da Figura 5.1b, 5.1c, 5.1d mostram como a precisão da triangulação varia à medida que é alterado o nível de detalhamento da imagem, conforme explicitado no capítulo 3.2, seção referente à Junção Adaptativa.

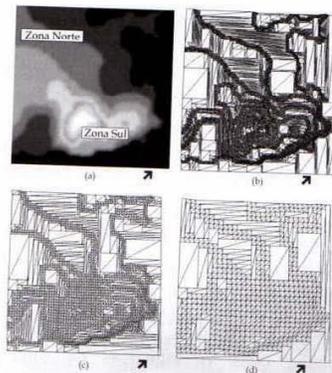


Figura 5.1: O Modelo Digital de Elevação original e as diferentes configurações da malha em vários níveis de refinamento.

A Figura 5.1b utiliza o nível 7 de refinamento, originalmente com 32.768 triângulos e 11.254 triângulos após a simplificação da malha pela Junção Adaptativa. A Figura 5.1c representa o nível 6 da *quadtree*, originalmente com 8.192 triângulos e 3.078 triângulos após a simplificação. Por fim, a Figura 5.1d representa o nível 5 de refinamento, originalmente com 2.048 triângulos e 1.216 triângulos após a simplificação. O refinamento da malha pode atingir até o nível 10, para o MDE da Figura 5.1a.

Ainda na seção sobre Junção Adaptativa, foi descrito que o algoritmo proposto para a simplificação da malha utiliza os dados da *quadtree* pertencentes a um mesmo nível, não levando em consideração nenhum outro nível de alturas inferiores da árvore. Desta forma, não há relação direta entre a aparência de um nível menos refinado, com um outro mais refinado. Por este motivo, observa-se na Figura 5.1 que para uma mesma posição da câmera, a malha apresenta diferentes configurações para os diferentes níveis de detalhamento, como por exemplo a região indicada pela seta. Na região assinalada da Figura 5.1b, os pares de triângulos encontram-se com a dimensão vertical acentuada, enquanto que na Figura 5.1c, a mesma região apresenta pares de triângulos predominantemente horizontais. Por fim, na Figura 5.1c, a mesma região apresenta novamente pares de triângulos como a dimensão vertical maior.

A utilização de texturas permite um alto grau de flexibilidade para a visualização do terreno. Podem ser utilizadas cartas de papel digitalizadas através do scanner, ou ainda imagens de satélites. As cartas em papel são importantes, pois agregam diversas informações de interesse em uma manobra, tais como: vias de acesso, pontos de abastecimento, hospitais, vegetação, construções urbanas e etc. As imagens de satélite, por sua vez, crescem de importância a cada dia, pois permitem determinar com grande precisão a organização do cenário da manobra. Características altamente dinâmicas tais como vegetação, regiões urbanas e

condições de tráfego são facilmente observadas nas imagens de satélites.

A Figura 5.2a mostra uma carta de papel digitalizada através de scanner na escala de 1:10.000. A vista é da Refinaria de Manguinhos e adjacências. A Figura 5.2b é uma imagem do satélite *Ikonos* com resolução de 100 metros, datada de 22 de novembro de 2002, gentilmente cedida pelo Departamento de Engenharia Cartográfica do Instituto Militar de Engenharia. A região iluminada pelo satélite corresponde às porções dos bairros da Urca, Flamengo, Botafogo, Leme e Copacabana.



(a)



(b)

Figura 5.2: Diferentes texturas aplicadas a malha do terreno: (a) carta de papel digitalizada através de um scanner na escala de 1:10.000; (b) imagem do satélite *Ikonos* com resolução de 100 metros.

Foram feitos testes com diversas placas de vídeo para determinar o comportamento da implementação em diferentes configurações. Em uma fase inicial, utilizaram-se diversos microcomputadores como plataformas de teste [2]. Uma característica importante do ensaio foi o fato de que os

microcomputadores com maior capacidade de processamento, não necessariamente possuíam as placas de vídeo mais adequadas. Foi constatado que alguns computadores com capacidade de processamento significativamente baixo apresentavam melhor desempenho que seus similares, porém com maior capacidade de processamento. Nestes casos, uma característica sempre presente foi a existência de placas de vídeo com suporte ao padrão OpenGL.

Desta forma, um segundo teste foi realizado [2]. Desta vez, a plataforma computacional permaneceu inalterada: utilizou-se um Pentium II 333Mhz, com 64MB de memória RAM, tal como o equipamento utilizado no FCB2 [6]. Apenas as placas de vídeo foram alteradas.

Um fator dominante para a melhoria do desempenho do protótipo é, portanto, utilizar placas de vídeo com suporte a biblioteca gráfica OpenGL. O OpenGL é uma API gráfica (*Application Programming Interface*) de baixo nível capaz de receber novas operações de *renderização* para estender as suas funcionalidades originais. Estas extensões permitem aos desenvolvedores de aplicações em OpenGL utilizar novas técnicas de *renderização* que não constam originalmente do padrão oficial do OpenGL. As extensões mantêm as APIs atualizadas com as últimas inovações dos hardwares gráficos e algoritmos de *renderização*.

6. CONCLUSÃO

Este artigo descreveu as técnicas implementadas durante o desenvolvimento do Sistema Tesamará. Este sistema tem como objetivo oferecer uma ferramenta computacional que unifica a visualização do campo de batalha com as logísticas de guerra descritas nas normas adotadas pelo Exército Brasileiro. O Tesamará é uma ferramenta que dinamicamente permite o entendimento das alterações das ações de guerra, otimizando assim as tomadas de decisões estratégicas por parte dos

comandantes de tropas. Desta forma, o sistema pode ser visto como peça fundamental para os procedimentos de Comando e Controle em Jogos de Guerra.

O problema da visualização do Teatro de Operações foi dividido em duas etapas: a modelagem do terreno e a modelagem das unidades. Na etapa de modelagem do terreno, foi descrito um algoritmo otimizado para a geração de malhas 3D, agregando diversas técnicas de Computação Gráfica. Na etapa de modelagem das unidades, foram utilizadas técnicas empregadas em jogos de entretenimento. As tropas foram modeladas por um pequeno volume de dados a fim de viabilizar a transmissão de informações para a atualização do cenário através de canais de comunicação bastante lentos.

Os resultados obtidos com os testes do Sistema Tesamará em laboratório mostraram que, para a visualização do terreno, é conveniente o uso de placas de vídeo com suporte em *hardware* para o padrão OpenGL. O padrão OpenGL permite um melhor equilíbrio entre o grau de detalhamento das cenas e a eficácia na visualização. De fato, era esperado que a visualização de cenas com elevado nível de detalhamento seja sempre balizada por um baixo valor na relação quadros por segundo.

É necessário observar que o uso do Sistema implicitamente propõe uma mudança de paradigma. O Exército Brasileiro é conservador, e tradicionalmente trabalha o problema da visualização do Teatro de Operações de forma não computacional. O uso dos recursos computacionais nas tarefas de Planejamento de Operações é um passo importante dentro da esfera de competência da Força Terrestre. Propor a utilização destes mesmos recursos para o estudo de cenários em três dimensões é um passo consideravelmente grande. Foi gratificante verificar o elevado grau de aceitação das idéias propostas, observando-se desdobramentos em outras instâncias do Exército Brasileiro.

A percepção do realismo virtual foi obtida a partir de procedimentos de texturas aplicadas sobre o modelo do

terreno. No entanto, o gerenciamento das texturas pode ser melhorado. Um mosaico de pequenas texturas, e com diferentes resoluções, pode ser utilizado de acordo com a distância entre a câmera e o terreno. Com isto, reduz-se o custo computacional de se manter em memória primária uma textura grande e com alta resolução.

Para futuros trabalhos deve ser dada uma abordagem cartográfica mais criteriosa. É preciso atentar para questões relacionadas com o geo-referenciamento, as transformações cartográficas afins, os sistemas de projeção, e ainda, os erros inerentes à representação planimétrica da calota terrestre.

O procedimento de confecção das marcações sobre as folhas de acetato que cobre os mapas também pode ser modelado computacionalmente. Apesar destas marcações não fazerem parte da visualização do Teatro de Operações, eles certamente fazem parte do planejamento da manobra. Esta funcionalidade, a ser acrescentada ao sistema, é particularmente interessante, pois está relacionada ao estudo de ferramentas de desenho 3D utilizando dispositivos tipicamente bidimensionais.

Eventualmente, devem-se avaliar os benefícios trazidos pela representação das tropas através de modelos não estilizados, isto é, que não respeitem a simbologia do manual C21-30 [18]. Esta abordagem pode ser interessante em escalas entre 1:25.000 e 1:1, cuja simbologia não é preconizada nos manuais. Tipicamente um veículo blindado poderia ser representado por um modelo 3D miniaturizado de um veículo blindado tal como o Cascavel, produzido pela indústria bélica brasileira.

Para se aprimorar a eficiência da dinâmica das ações de guerra inseridas no Teatro de Operações, torna-se necessário que o processo de modelagem das tropas seja executado em quatro dimensões, visto que uma ação de guerra está diretamente associada ao fator tempo. Desta forma, a variável temporal é importante, e deve fazer parte de futuras implementações, garantindo a dinâmica evolucionária da guerra.

7. REFERÊNCIAS

- [1] PARISH, Y., MÜLLER, P., "Procedural Modeling of Cities". In: *SIGGRAPH 2001*, pp. 301-308, 2001.
- [2] MELLO, F., *Visualização Tridimensional do Teatro de Operações*. Dissertação M.Sc., COPPE/UFRJ, julho, 2003.
- [3] INCE, A., EVRENDILEK, C., WILHELMSSEN, D., *Planning and Architectural Design of Modern Command Control Communications and Information System*. London, Kluwer Publishing, 1997.
- [4] MACEDONIA, M., "Games Soldiers Play". In: *IEEE Spectrum*, pp. 32-37, March 2002.
- [5] DURBIN, J., SWAN, J. E., COLBERT, B., "Battlefield visualization on the responsive workbench". In: *IEEE Visualization*, pp. 463-466, 1998.
- [6] KAUCHAK, M., "Transforming Land Warfare - US Army Digitalization Initiative Leaps Ahead". In: *Armed Forces Journal*, pp. 14-17, July 2001.
- [7] TURNER, B., "Real-Time Dynamic Level of Detail Terrain Rendering with ROAM", <http://www.gamasutra.com/features/2000/0403/turner01.htm>, 2000.
- [8] WOO, M., NEIDER, J., DAVIS, T., *OpenGL Programming Guide*. New York, Addison Wesley, 2000. 3 ed.
- [9] LINDSTROM, P., KOLLER, D., AL, E., "Real-time continuous level of detail rendering of height fields". In: *SIGGRAPH '96 Proceedings*, pp. 109-118, 1996.
- [10] HOPPE, H., "Smooth View Dependant Level-of-Detail Control and its Application to Terrain Rendering". In: *Technical Report, Microsoft Research*, 1998.
- [11] BLOW, J., "Terrain Rendering at High Levels of Detail". In: *Game Developers' Conference 2000*, pp. 13-17, 2000.
- [12] ZHAO, Y., ZHOU, J., SHI, J., et al., "A Fast Algorithm For Large Scale Terrain Walkthrough". In: *CAD/Graphics 2001*, pp. 43-50, 2001.
- [13] RÖTTGER, S., HEIDRICH, W., SLASALLEK, P., et al., "Real-Time Generation of Continuous Levels of Detail for Height Fields". In: *SIGGRAPH '98 Proceedings*, pp. 315-322, 1998.
- [14] BERG, M. D., KREVELD, M. V., OVERMARS, M., et al., *Computational Geometry - Algorithms and Applications*. Berlin, Springer, 2000. c.14.
- [15] MORLEY, M., "Frustum Culling in OpenGL", <http://www.markmorley.com/opengl/frustumculling.html>, January 2001.
- [16] EBERT, D. S., MUSGRAVE, F. K., PEACHEY, D., *Texture and Modeling: a procedural approach*. New York, Academic Press, 1998. 2 ed.
- [17] FOLEY, J. D., DAM, A. V., FEINER, S. K., *Computer Graphics: principles and practice*. New York, Addison Wesley, 1997. 2 ed.
- [18] do Exército, E. M. (ed.), *Manual de Campanha - Abreviaturas, Símbolos e Convenções Cartográficas*, C21-30. Brasília, Ministério do Exército, 1992.
- [19] HAWKINS, K., ASTLE, D., *OpenGL Game Programming*. San Francisco, Prima Tech's, 2001.
- [20] LAMOTHE, A., *Black Art of 3D Game Programming*. New York, Waite Group, 1995.
- [21] RÖTTGER, S., "Terrain LOD Implementations - libMini", <http://www.vterrain.org/LOD/Implementations/>, 2000.
- [22] STRAUSS, E., OLIVEIRA, A., MELLO, F., "Battlefield Visualization and Intelligent Agents Decision Support". In: *Proceedings of SIBGRAPI 2002*, p. 429, October 2002.