

Uso de Quadfields para Otimizar a Largura de Banda em Jogos Multi-Jogador

LAURO EDUARDO KOZOVITS¹

CLÁUDIO ESPERANÇA²

BRUNO FEIJÓ³

ESTEBAN WALTER G. CLUA³

¹ ActiveTec – Active Tecnologia e Consultoria Ltda.

lauro@ActiveTecnologia.com.br

² Instituto Alberto Luiz Coimbra de Pós Graduação e Pesquisa de Engenharia - Programa de Engenharia de Sistemas. CT UFRJ

esperanc@cos.ufrj.br

³ Departamento de Informática – PUC-Rio / VisionLab

{bruno, esteban}@inf.puc-rio.br

Abstract

This paper uses the idea of Quadtrees in order to generate a space adaptative method for description of 3D data. This method is important for the reduction of data transmission for on-line applications, especially massive multiplayer games or mobile multi-player games based on SMS messages.

Keywords: *Quadtree, massive-multiplayer games, mobile games, geometry modelling*

1 Introdução

Jogos multi-jogador constituem uma sub-área de ambientes virtuais ligados por rede (net-VEs) que têm despertado um grande interesse tanto por parte da indústria como pela comunidade científica. Trabalhos relativos a otimização de mensagens trocadas nestes ambientes demonstram a relevância da pesquisa do uso eficiente da rede para a obtenção da consistência visual entre os participantes de um jogo ou simulação [1,2,3]. Também são relevantes as propostas de evolução da Internet apresentadas pela Internet2 [4].

Em geral, simulação e jogos multi-jogador, em particular os jogos multi-jogador de massa usam a Internet como a rede para conexão dos jogadores. Estes jogadores podem estar conectados por modem analógico com estreita largura de banda. Isto limita a quantidade de

informação que pode ser recebida de um servidor por unidade de tempo. Uma destas limitações é o número máximo de jogadores simultâneos que, numa arquitetura cliente-servidor, precisam receber informação de atualização (posição, velocidade, etc) referente a cada um dos elementos.

A disponibilidade da chamada banda larga tem crescido, assim como a diminuição do custo de sua assinatura mensal. Apesar desta crescente popularidade, novos usos como, por exemplo, animações de personagens mais complexos, animação e expressividade facial e tráfego de voz entre os jogadores fazem com que este recurso não deva ser desperdiçado. Também do lado do servidor, o trabalho proposto é relevante, pois quanto maior o consumo de banda, maior o custo do serviço. Todavia esta técnica torna-se especialmente útil em jogos

baseados em turnos utilizando SMS em função de seu alto custo e limitação máxima de 160 caracteres.

Neste artigo propõem-se uma técnica derivada das idéias lançadas em [5] para minimizar o tamanho das mensagens de atualização de um jogo multi-jogador ou simulação baseada na representação eficiente das coordenadas do universo virtual modelado. A aplicabilidade da técnica proposta é ilustrada com um caso de uso. Para tanto, utiliza-se o conceito de Quadtree, que são estruturas de dados que realizam o particionamento de um espaço em quatro regiões. Cada uma destas regiões será recursivamente particionada, até que se atinja o nível de detalhe desejado. Os Quadrees são tipicamente representados por árvores [15].

2 Representação do Universo Virtual e uso de Quadfields

De um modo geral, usam-se os tipos de dados disponíveis nas linguagens de programação para representação de coordenadas do universo modelado. Tipos de dados como o *int* ou *float* definidos em Java ou no Microsoft Visual C/C++, por exemplo, são representados internamente com 32 bits o que permite seu uso genérico em um grande número de aplicações. Apesar de o volume de dados típico de aplicações de computação gráfica ser elevado, a atual capacidade de memória dos PCs, bem como sua capacidade de CPU e GPU permitem, na maioria dos casos, um tratamento conveniente deste volume de dados. O maior desafio para o tráfego de grandes volumes de dados e a própria evolução da tecnologia de jogos de ação e simulação multi-jogador está no desenvolvimento de técnicas para otimizar o volume de dados necessários de serem transmitidos, contornando consequentemente as limitações da rede.

Evidentemente, a tecnologia de rede também tem evoluído, assim como têm surgido novas técnicas para uso otimizado deste recurso. Um exemplo disto são os trabalhos para lidar com a transferência de grandes volumes de dados gráficos, onde procura-se otimizar o tráfego de malhas poligonais [6, 7]. Com relação

ao uso eficiente da rede no tráfego de mensagens, pode-se citar o uso de broadcast, multicast, da técnica original de dead-reckoning [1] e de uma versão modificada [8].

Entretanto, um aspecto deve ser considerado quanto ao uso da rede em jogos multi-jogador: certas tecnologias, embora existentes, não estão disponíveis para uso em massa via Internet o que é o caso de transmissão broadcast, disponível apenas para redes locais, e o uso ainda não generalizado de servidores de multicast. Também deve-se considerar a já mencionada limitação da largura de banda de cada participante de um jogo ou simulação. Como este tipo de aplicação destina-se ao consumo em massa, não se pode assumir que todos os interessados tenham acesso a tecnologias mais avançadas. Atualmente, uma das soluções adotadas para este problema consiste em apenas transmitir dados sobre atualização de cenários (mudança de atributos de personagens, como por exemplo posição e vida). Raramente transmitem-se dados da geometria do cenário ou de objetos tridimensionais. Assim, quando o usuário instala a aplicação, já coloca toda a geometria que será necessária. Isto é uma restrição grande, pois não permite que os cenários possam crescer de forma livre. A pesquisa também se justifica face ao interesse por jogos multi-jogador para celular baseados em SMS. Isto dá-se pelo fato de que mesmo considerando-se um jogo simples onde apenas as coordenadas 2D de jogadores devam ser comunicadas há uma limitação física no número de caracteres (160) transportando a informação do servidor para os celulares do jogador. Jogos SMS despertam um maior interesse por parte de operadoras de celular e empresas de jogos pelo fato de terem um mecanismo de tarifação e controle integrados em todos os celulares com tecnologia Java ME.

Isto motivou a pesquisa neste trabalho por uma nova técnica a ser adicionada ao acervo de técnicas de otimização de mensagens já existentes. A idéia de Deering [5] surge do fato de que o universo virtual de modelos geométricos é uma representação digital de um espaço contínuo, uma discretização portanto. Naquele trabalho, é proposto uma diminuição do número de bits na representação de coordenadas

de modelos visando sua transmissão eficiente sem perda considerável de sua definição.

O presente trabalho estende a idéia para transmissão de coordenadas de posição de avatares e apresenta a técnica de quadfields para explorar a coerência de posicionamento de jogadores em um mundo virtual.

Em geral, pode-se dizer que a função utilizada na representação do universo virtual é um mapeamento de um espaço discreto Z^3 num espaço contínuo R : $f(x, y, z): Z^3 \rightarrow R$.

Na fig.1, pode-se observar a discretização de um terreno de 10 m^2 utilizando-se 3 bits para esta representação.

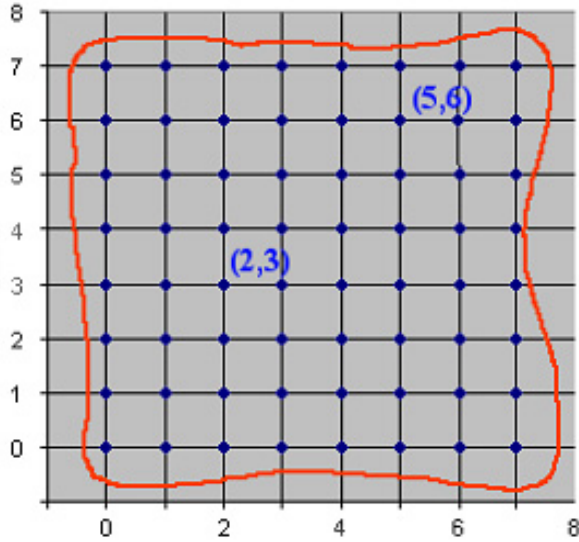


Fig.1 representação de um terreno de 10 m^2 usando-se 3 bits, com uma precisão de representação equivalente a $1,25\text{m}$.

De um modo geral, a precisão da representação digital num universo 2D é dada pelas equações 1 e 2 a seguir:

$$\Delta x = \frac{x_{\max}}{2^N} \quad (1)$$

$$\Delta y = \frac{y_{\max}}{2^N} \quad (2)$$

A precisão da representação é dada por:

$$x' = \frac{x \cdot 2^N}{x_{\max}} \quad (3)$$

Dependendo da precisão requerida, pode-se representar coordenadas do universo modelado com um número maior ou menor de bits. Observa-se na literatura [9, 10] e na inspeção de códigos fonte de jogos disponíveis na Internet [11, 12] o uso dos tipos de dados disponíveis nas linguagens de programação empregadas na implementação de jogos ou ambientes virtuais.

Constata-se pela aplicação da equação eq.3 que, usando-se 32 bits (tipos *int* ou *float*), é possível representar toda a área de nosso planeta com precisão de apenas 5 mm. Embora jogos como Ultima Online [13] possam ter dimensões virtuais comparáveis a área de muitos países é muito provável que o uso de 32 bits para representação de coordenadas de mundos virtuais possa ser algo sujeito a questionamentos.

A movimentação de personagens num jogo só provoca alterações na cena renderizada se o deslocamento efetuado implicar na alteração de algum pixel da imagem. Por exemplo, é possível que, mesmo considerando-se apenas os objetos mais próximos, só ocorra uma variação na imagem renderizada quando o deslocamento destes objetos for superior a um número dezenas de vezes superior a precisão mencionada. Desta forma, a precisão do uso de 32 bits pode ser excessiva.

Num universo menor, o número de bits requerido pode ser ainda menor. Isto sugere que o envio de coordenadas pela rede seja feito com um número de bits com precisão apenas o suficiente para transmissão bem sucedida das coordenadas dos objetos de modo a economizar a banda disponível. De um modo geral, deve-se observar a posição do observador, a velocidade dos objetos, as dimensões do mundo virtual modelado e a taxa de atualização das

coordenadas dos objetos componentes de um jogo ou simulação.

Como se está propondo um processo de redução da representação de números, chamado de *decimation*, seguido de um processo de reamostragem é importante que o número de amostras seja pelo menos duas vezes maior que a maior frequência encontrada na representação. Caso contrário é possível a ocorrência do fenômeno de *aliasing*.

Esta maior frequência encontrada pode ser obtida experimentalmente em cada aplicação. Por exemplo, numa observação aérea de um cenário, algo típico em certas categorias de jogos, o menor deslocamento visualmente detectável para objetos próximos irá determinar qual a resolução mínima necessária para o campo de bits representando o mundo virtual.

Propõe-se, portanto, neste trabalho, de modo semelhante ao proposto por Deering [5], o uso de campos de bits (*bitfields*) com precisão suficiente para a correta transmissão de coordenadas de objetos em movimento, de modo a proporcionar uma economia de banda desta transmissão. Isto é especialmente válido em mensagens de atualização de um servidor para os jogadores conectados. Neste caso, cada conjunto de coordenadas (posição, velocidade, etc) é multiplicado pelo número de jogadores numa região atendida por este servidor e o tamanho de uma mensagem pode ser elevado face a capacidade de banda de cada jogador.

A compressão da mensagem de atualização de coordenadas pode, entretanto, ser ampliada se, além da redução do número de bits para representação do mundo virtual, for explorado o fato de que jogadores tendem a interagir com os demais, agrupando-se em cidades, vilas ou mesmo em regiões de conflito. Desta forma é possível que bits mais significativos representando as coordenadas destes jogadores possam ser compartilhados na composição da mensagem de atualização de posição. Uma

estrutura adequada para captar o posicionamento dos jogadores é uma *quadtree* adaptada à criação de um campo de bits. Daí o nome proposto "*quadfields*".

Na figura 2, tem-se um exemplo de 3 avatares (identificados pelos números 1, 2 e 3) e suas coordenadas horizontal e vertical correspondentes num mapeamento didático de apenas 3 bits por coordenada.

01	11		
00	1001	101101 1 2	101111
		101100	101110 3
	1000	1010	

Fig.2 representação para montagem de uma quadfield um terreno usando-se 3 bits por coordenada

Nesta representação os objetos 1 e 2 ocupam a mesma posição no espaço (101101), pois não há resolução superior a 3 bits por coordenada e o objeto 3 ocupa a posição (101110).

Uma mensagem de atualização convencional pode ter a forma ObjectID + Xpos + Ypos. Assim, a mensagem de atualização convencional (16 bits para cada identificador de objeto), tem a forma:

```
0000000000000001
101101
```

```

0000000000000010
101101
000000000000 0011
101110

```

A mesma mensagem quando enviada na forma de quadfields tem a forma:

```

[quadrante 00 nível 0]
[quadrante 01 nível 0]
[quadrante 10 nível 0]
[quadrante 11 nível 0]

```

Cada quadrante [quadrante XY nível L] só é representado na *quadfield* quando existem objetos neste quadrante. Cada quadrante pode ser subdividido, sucessivamente, em quatro outros subquadrantes até o número máximo de bits por coordenada (ou nível máximo). Chegando-se ao nível final procede-se a geração das listas de objetos onde cada objeto utiliza o bit 1 para indicar que os próximos bits referem-se ao identificador do objeto. A presença de um bit 0 marca o fim da lista de objetos.

Desta forma, a *quadfield* correspondente ao exemplo acima é:

10 11 01 (lista de objetos: 1 e 2) 10 (lista de objetos: 3) ou

```

10 11 01 1 0000000000000001 1
0000000000000010 0 10 1 0000000000000011 0

```

Quando um determinado nível não apresenta o quadrante 11, usa-se o par de bits 00 para marcar o fim daquele nível. Além disso, existe um cabeçalho com campos para armazenar o número de bits da mensagem e o *checksum* correspondente. Após a geração do campo de bits, usa-se compressão tradicional

(ZIP, GZIP, etc) para uma possível compressão adicional.

3 Análise de Resultados

Um programa [14] foi criado para teste da compressão de dados com uso da técnica de *quadfields* apresentada. Com o programa foi possível realizar a simulação de um terreno com área de 262144 mil km² (quadrado com aresta 512000m). Procede-se a geração de mensagens de atualização de posição de 100 e de 1000 avatares nas seguintes situações:

- Dispersos randomicamente por todo o terreno;
- Dispersos randomicamente num quadrado de 10 km de aresta para simulação de uma cidade;
- Dispersos randomicamente num quadrado de 1000m de aresta para simulação de uma pequena vila.

As tabelas a seguir ilustram a compressão de dados obtida com a técnica proposta aplicada a um terreno de 512000m x 512000m com 22 bits por coordenada (utilizando portanto uma resolução de 0.12m por bit)

Tamanho da mensagem (bytes)	terreno: 512000m x 512000m	
	Arq. zip	Quadfield+ compressão
100 avatares	963	838 (1.15 X)
1000 avatares	9263	7489 (1.23 X)

Tamanho da mensagem (bytes)	cidade: 10000m x 10000 m	
	Arq. zip.	Quadfield+ compressão
100 avatares	939	669 (1.4 X)
1000 avatares	8854	5803 (1.53 X)

Tamanho da	vila: 1000m x 1000m
------------	---------------------

mensagem (bytes)	Arq. zip	Quadfield+ compressão
100 avatares	890	574 (1.55 X)
1000 avatares	7874	4792 (1.64 X)

Estes quadros ilustram que é possível obter uma compressão adicional à compressão tradicional aplicada as mensagens de atualização de coordenadas. Esta compressão baseia-se na redução do número de bits da amostragem e na exploração da tendência à disposição concentrada de avatares em regiões de um terreno de um jogo ou simulação. Quanto mais agrupados os avatares, o que é uma situação real e frequente, maior a compressão obtida.

Não foi observada nenhuma perda de performance do servidor para o número de objetos testados (até 1000).

Entretanto quando este número for compatível com aquele encontrado em jogos MMORPG (centenas de milhares) torna-se necessário um estudo mais aprofundado.

Num jogo turn-based com dados enviados para um celular fica evidente o fato de que não é relevante o tempo gasto na montagem dos dados a serem enviados já que o que está sendo considerado é a compressão a ser obtida dado o custo da transferência de dados nesta plataforma.

Fica, entretanto, como sugestão para trabalhos futuros a análise de algum eventual *overhead* em algumas situações de jogos com número elevado de participantes utilizando a técnica proposta.

4 Conclusão

Os autores apresentam uma forma nova para o tráfego de mensagens em jogos multi-player, simulações que utilizam transmissão de dados pela rede e certas aplicações de ambientes virtuais ligados por rede. Este método, denominado de estruturas quadfields, representam coordenadas de um ambiente virtual de uma forma mais eficiente que as abordagens de estruturas de dados tradicionais.

Neste trabalho, o número de níveis para os quadfields são gerados através de uma análise

das dimensões do cenário e o posicionamento do observador.

É possível resumir a técnica quadfields proposta como uma composição entre uma compressão semelhante a RLE (run-length encoding) e uma redução do número de bits (decimation) para representar as coordenadas.

O trabalho também sugere que a estrutura de quadfields seja utilizada para descrever as coordenadas geométricas das malhas que representam os avatares da simulação ou do jogo multi-player. Esta abordagem é totalmente consistente com as propostas atuais de dead-reckoning, no gerenciamento de estados compartilhados. Os autores desconhecem uma abordagem semelhante na literatura, onde se tratam compressão de geometria e estruturas de quadrees para representar coordenadas expressadas em bits.

A maioria dos jogos *multi-player* com código fonte aberto [11, 12] e livros relacionados ao tema [9, 10] utilizam estruturas de dados padrões para comunicar dados relacionados a coordenadas entre os diversos jogadores.

A divisão do plano 2D no mapa do jogo contendo os avatares é regular (cada nível é sempre a metade do anterior). Na etapa de inicialização, usa-se um critério empírico para estabelecer o nível máximo de subdivisões a ser utilizado. Por exemplo, num terreno de 100 km² pode-se aceitar que 0,5 metros é uma precisão razoável e isto já implica no número de bits a ser utilizado. A resolução em bits é fixa e empírica. Entretanto, ela não precisa ser a mesma em cada eixo. Na mesma situação exemplificada, ao se representar o eixo Y (alturas) é possível que a cota de interesse, para um dado jogo, esteja entre 0 e 1000 metros o que demandaria um número de bits menor mantendo-se a mesma precisão do exemplo (0,5 metros). Ao se definir um número de bits para cada coordenada a posição de cada avatar é sempre calculada de acordo com a resolução máxima possível para aquela coordenada. Futuramente propõem-se que sejam gerados exemplos que envolvam simulações mais complexas e com mais participantes. Além disso, o sistema poderia vir a utilizar um processo adaptativo para

reestruturação dos *quadfields*, dependendo do nível de detalhamento necessário para certas situações, tal como aproximação da câmera num local do terreno ou de outro personagem. Além disso, um modelo formal para a etapa de *decimation* e reamostragem necessita ser aprofundado. Este último tópico é essencial para obter um processo adaptativo mais geral para economizar banda num jogo multi-player, simulações através de redes e ambientes virtuais ligados por redes.

5 Agradecimentos

Os autores gostariam de agradecer ao CNPq pelo apoio financeiro, a equipe de desenvolvimento e apoio técnico do LCG-UFRJ e à FINEP pelo apoio a um dos laboratórios envolvidos.

6 Bibliografia

1. IEEE Standard 1278 - 1993. IEEE standard for information technology - Protocols for distributed simulation applications: entity information and interaction, IEEE Computer Society
2. Szwarcman, D. Feijó, B. and Costa, M., 2000. A framework for networked reactive characters, Proceedings of SIBGRAPI 2000, Gramado, Brazil, p. 203-210.
3. United States Department of Defence, Defence Modeling and Simulation Office. High Level Architecture. Web page, 2002. [accessed Aug. 1, 2006], available on <https://www.dmsomil/public/transition/hla/>.
4. Internet 2 web site, [accessed Aug. 1, 2006], available on <http://www.internet2.edu/>
5. Deering, Michael, 1995. Geometry Compression. Proceedings of SIGGRAPH 1995. Available in: < http://www.cse.ohio-state.edu/~hwshen/Su01_888/deering.pdf >. Accessed in: Jul. 3, 2006.
6. P. Alliez and M. Desbrun, "Progressive encoding for lossless transmission of 3D meshes," in ACM SIGGRAPH Conference Proceedings, 2001.
7. Gueziec, A., Taubin, G., Horn, B., and Lazarus, F. 1999. A Framework for Streaming Geometry in VRML. IEEE Computer Graphics and Applications, 19(2):68--78
8. Sandeep K. Singhal, and David R. Cheriton, "Using a Position History-Based Protocol for Distributed Object", Technical Report STAN-CS-TR-94-1505, Dep. Of Computer Science, Stanford University, Feb. 1994.
9. T. Barron, *Multiplayer Game Programming*, Prima Publishing, Roseville, CA, USA (2001)
10. A. Perez and D. Royer, *Advanced 3d Game Programming using DirectX 7*, Worldware Publishing, Plano, TX, USA (2000)
11. BzFlag "a free multiplayer multiplatform 3D tank battle game", web site, [accessed Aug. 1, 2006], available from <http://www.bzflag.org>
12. DoomWorld web site, source port for Doom, Heretic or Hexen, [accessed Aug. 1, 2006], available from <http://www.doomworld.com/classicdoom/ports/>
13. Ultima Online web site, [accessed Aug. 1, 2006], available from <http://www.uo.com/>
14. Active Tecnologia e Consultoria Ltda. web site, [accessed Aug. 1, 2006], available from <http://www.activetecnologia.com.br/quadfields/>
15. Finkel, R. and Bentley, J. L. "Quad Trees: A Data Structure for Retrieval on Composite Keys". *Acta Informatica* 4 (1): 1-9, 1974.