

Representação de Paisagens Florestais Adequada à Navegação para Áreas de Grande Dimensão

André O. Falcão
Departamento de Informática
Faculdade de Ciências da
Universidade de Lisboa
afalcao@di.fc.ul.pt

Manuel Próspero dos Santos
Departamento de Informática
Faculdade de Ciências e Tecnologia
Universidade Nova de Lisboa
ps@di.fct.unl.pt

Sumário

Descreve-se o sistema desenvolvido para permitir a visualização, em tempo real, de modelos de áreas florestais de grande dimensão. Em particular, detalham-se os procedimentos usados para lidar com a complexidade e quantidade de informação espacial necessária para um sistema deste género, nomeadamente ao nível da representação das árvores e do terreno, determinação de visibilidades e subdivisão e compartimentação do espaço. O sistema é demonstrado para uma área com cerca de 95,000 ha situada no Barlavento Algarvio (Serra de Monchique). Descrevem-se, ainda, os mecanismos de interação com o sistema e os seus modos de funcionamento. Discutem-se os resultados obtidos, tanto ao nível da qualidade visual das paisagens produzidas, como do desempenho do sistema para uma plataforma de teste.

Palavras-chave

Computação gráfica interactiva, modelação de sistemas naturais, visualização de árvores, visualização de terrenos, realidade virtual, navegação virtual.

1. INTRODUÇÃO

Ao longo das últimas décadas, a opinião pública tem-se mostrado progressivamente preocupada com os efeitos visíveis das mudanças ambientais [Luymes01, Sheppard89]. Verifica-se uma crescente sensibilidade do público para as questões ambientais e paisagísticas, ao que o crescimento da procura turística nas zonas rurais vem salientar ainda mais o valor objectivo da paisagem na economia das populações situadas em zonas florestais ou próximo delas. Por outro lado, avanços significativos nas tecnologias de informação nas últimas décadas permitiram a inclusão de componentes paisagísticos como factores passíveis de influenciar a forma como se gerem as florestas. Com efeito, a maior capacidade de processar informação, em particular a informação gráfica em computadores de uso comum, permite que seja possível visualizar-se o impacte paisagístico de planos de gestão vários anos antes de eles se tornarem aparentes. Os trabalhos de McGaughey [McGaughey97], Orland e Uusitalo [Orland01] e Danahy [Danahy01] demonstram que é possível apresentar os efeitos de diferentes estratégias de gestão em imagens que são prontamente identificáveis pelos utilizadores finais [Luymes01]. Trabalhos anteriores serviram para definir os requisitos de informação e definir um protótipo inicial que foi utilizado para recolha de preferências estéticas de utilizadores do espaço florestal face a planos de gestão florestais alternativos [Falcão04] (por exemplo, as vistas mostradas na Figura 1 simulam a alteração de uma paisagem passados quinze anos após o

momento inicial, dependendo do plano de gestão florestal posto em prática). Esta versão do sistema apresentava, contudo, deficiências relativas à qualidade visual produzida, tendo ainda problemas de desempenho que dificultavam a sua transposição para áreas de grandes dimensões.

O trabalho descrito nesta comunicação foca os esforços encetados para estender ainda mais essa capacidade de visualização ao procurar a representação de uma paisagem virtual interactiva na qual o utilizador possa navegar. Esta abordagem comunica ao utilizador uma perspectiva mais abrangente, fornecendo uma maior quantidade de informação. O objectivo do sistema proposto é, pois, permitir ao utilizador percorrer virtual e livremente a área considerada sob um determinado plano de gestão. No sistema desenvolvido, além de ser possível observar a paisagem estaticamente a partir de um ponto de vista pré-definido, permite-se ainda ao utilizador a movimentação pela floresta como se se deslocasse a andar, ou realizar um voo virtual por cima das copas para ter uma ideia global da envolvente florestal num determinado momento no tempo.

A concepção e desenvolvimento de uma ferramenta deste tipo não está isenta de desafios computacionais. Estes referem-se, essencialmente, à complexidade geométrica inerente a cenas de paisagens florestais. E há, concretamente, dois problemas fundamentais a considerar na arquitectura de um sistema deste género. O primeiro pren-

de-se com a capacidade física do equipamento para representar graficamente uma grande quantidade de polígonos em tempo real. O segundo consiste na gestão da memória RAM disponível. Mesmo com um nível de detalhe pouco exigente, não é correntemente exequível armazenar-se, na memória RAM de um computador, a informação total contida numa paisagem florestal para a poder representar em tempo real.

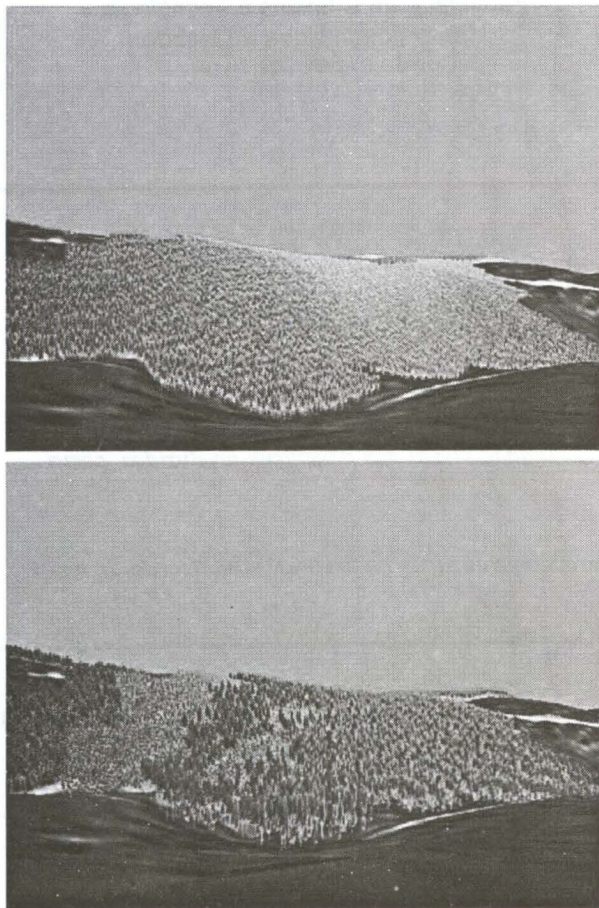


Figura 1. - Comparação entre dois cenários de gestão diferente para uma mesma área florestal na Serra da Lousã (Cantão das Hortas).

Para enquadrar melhor o problema refira-se a situação típica de utilização de uma ferramenta como esta: pretende-se um sistema capaz de permitir a visualização de uma área com, pelo menos, 100,000 ha e uma densidade florestal de, no mínimo, 300 árvores por ha, ou seja, 3.0×10^7 árvores ao todo. Utilizando-se uma representação geométrica bastante conservadora, nunca se poderia disponibilizar menos que 96 bytes por árvore, totalizando, portanto, cerca de 2.682 GBytes, o que é incomportável para a esmagadora maioria do hardware actualmente existente. É evidente que nem todas as árvores estarão visíveis, podendo estas ficar armazenadas em memória não volátil e sendo carregadas à medida que vão entrando dentro da esfera de visibilidade. Mas, mesmo assim, o problema não é de resolução trivial. Se admitirmos um horizonte de visibilidade com 2.0 km, que é um valor relativamente moderado, poderemos estar a contar com cerca de 60,000 árvores a representar em cada *frame*. Considerando a

mesma geometria simplificada, temos necessidade de enviar cerca de 480,000 vértices por *frame* para o *pipeline* gráfico, o que é um valor considerável, até para o hardware topo de gama actualmente disponível. De notar ainda que estes valores são médios. Situações com uma densidade de coberto florestal mais intensa devem, também, estar previstas, pois certamente irão ocorrer. Na prática observam-se densidades até 4 ou 5 vezes superiores a estes valores de referência. Assim, é necessário encontrar estratégias que permitam sintetizar, agrupar e reduzir a informação existente por forma a que seja possível construir um sistema que possua um desempenho e qualidade suficientemente bons para funcionar como ferramenta activa no suporte à decisão em gestão florestal.

É nesta perspectiva que se descrevem, no presente artigo, várias estratégias de concepção do sistema desenvolvido, denominado *Landscaper* (actualmente na versão 2.0). De início referem-se as diferentes etapas de processamento de informação geográfica e biométrica. Seguidamente descrevem-se algumas das componentes tecnológicas utilizadas, as técnicas de maior relevo e as plataformas de suporte. O sistema é demonstrado para uma área com cerca de 95,000 ha situada no Barlavento Algarvio. Finalmente, relatam-se os resultados obtidos no que diz respeito à qualidade visual verificada e ao desempenho alcançado.

2. PRÉ-PROCESSAMENTO DA INFORMAÇÃO

Idealmente, um visualizador de paisagens florestais teria informação quantitativa de todas as características do terreno e de todas as árvores existentes. Teria ainda o poder computacional necessário para mostrar toda essa informação, permitindo a interactividade com o utilizador. Por razões que a seguir se indicam, porém, na prática tal não acontece exactamente assim.

2.1 Informação florestal

A natureza da informação florestal, pela sua grande quantidade e diversidade, é de recolha e tratamento difíceis. Se é conveniente para um sistema de visualização que todas as árvores da paisagem estejam medidas e identificadas, essa situação geralmente não ocorre, sobretudo para áreas tão extensas como as abordadas neste trabalho.

Há duas formas de obtenção de informação florestal. A mais comum é a medição directa através do inventário florestal, em que apenas uma pequena proporção das árvores é medida. A segunda forma é através da utilização de modelos de simulação, em que as características básicas dos povoamentos são simuladas para prever a sua situação em algum momento no futuro.

Assim, a informação disponível é geralmente muito reduzida para se poder representar uma paisagem florestal sem se recorrer a alguma generalização. A recriação da estrutura florestal é uma componente fundamental na representação da paisagem, dado não ser exequível a recolha da informação referente a cada árvore do sistema. O método que foi usado baseia-se na criação de povoa-

mentos artificiais com base em informação agregada [Falcão04]. Esta pode ser proveniente de medições no terreno conforme são recolhidas no processo de inventário florestal, ou resultar da execução de modelos de simulação ou de sistemas de apoio à decisão para gestão de ecossistemas florestais [Falcão01].

2.1.1 Árvores

Embora com diferentes componentes separáveis e prontamente identificáveis (tronco, ramos, folhas, etc.), devido à sua forma irregular a representação geométrica de árvores não é de resolução fácil [Meyer01]. Assim, uma representação geométrica simples, sem envolver um grande detalhe, pode facilmente chegar aos 2,000 polígonos. Certos programas que fazem modelação detalhada de árvores podem representar estruturas com 200,000 polígonos. Contudo, a utilização de um nível de detalhe desta ordem de grandeza numa ferramenta de visualização interactiva só é justificável para áreas relativamente reduzidas, ou apenas quando se queira representar realisticamente uma só árvore.

A meta proposta de horizontes visuais de 2.0 km proporciona uma área visível de 200 ha, o que corresponderá, em média, a 400,000 árvores em áreas com uma florestação intensa. Desta forma, pensou-se em métodos de representação com as formas mais simples possíveis. O uso de *billboards* unidimensionais que rodam automaticamente para estarem de frente para o observador foi uma metodologia considerada, visto reduzir ao mínimo a complexidade da geometria. Contudo, problemas de desempenho e de perda de sensação da realidade na observação de árvores muito próximas ou de ângulos desfavoráveis (de cima, por exemplo) obrigaram à rejeição desta possibilidade. A abordagem que foi seguida é um método clássico utilizado em vários ambientes virtuais. São, assim, usados dois rectângulos entrecruzados aos quais, à semelhança dos *billboards*, é aplicada uma textura com um canal *alpha*, para garantir a transparência. O resultado é, na generalidade dos casos, satisfatório, havendo naturalmente situações em que a excessiva simplificação geométrica é bem aparente. Apesar de tudo, considerou-se que esta abordagem seria a mais favorável, sendo os seus benefícios comparativamente superiores aos inconvenientes suportados.

Deste modo, as árvores tipo de cada povoamento ficam representadas por três atributos: altura, diâmetro da copa e espécie. A altura e o diâmetro da copa definem a forma da árvore e dependem das características do povoamento em que estão inseridas. A espécie é utilizada pelo programa para seleccionar a textura que é aplicada aos rectângulos gerados. De notar que é possível ter mais do que uma textura para cada espécie, uma vez que este atributo também está relacionado com a forma da árvore.

Foi utilizado o software comercial OnyxTree [Onyx04] para modelar as árvores e criar as respectivas texturas. Este software permite a modelação de árvores por meio das suas características biométricas. Além de conter uma biblioteca extensa de árvores já previamente parametrizadas, permite a sua fácil adaptação para quaisquer tipos

de condições. No nosso caso foram utilizados os módulos *OnyxTree Broadleaf* e *OnyxTree Conifer*, permitindo a exportação directa de imagens com o canal de transparência incorporado. Tais imagens são suficientemente simples para não distrair o utilizador com algum defeito particular, conseguindo, apesar de tudo, comunicar as características estéticas mais importantes da espécie.

2.1.2 Aspecto textural do terreno

A representação do terreno e das suas características fisiográficas é também um problema importante a considerar. Na aplicação a um caso concreto, a informação referente à altimetria está geralmente disponível e é de fácil acesso. Também é necessário que exista informação que permita a delimitação dos povoamentos sobre o terreno, frequentemente realizada por fotointerpretação ou marcação directa por GPS.

Para recriar o solo definiram-se texturas tipo que são aplicadas ao modelo digital do terreno, de acordo com as características do coberto. Esta metodologia apresenta a vantagem óbvia de ser possível representar o solo quando não existam árvores, permitindo a existência de um detalhe suficiente na representação do solo e tendo geralmente uma presença em memória RAM relativamente reduzida, uma vez que é suficiente a existência de pequenos fragmentos que sejam representativos de um determinado tipo de solo ou coberto herbáceo. Esta abordagem obriga, no entanto, a um processamento detalhado da geometria que caracteriza o terreno e a sua ocupação, uma vez que também é conveniente que uma textura circunde completamente cada povoamento ou tipo de ocupação.

Recolheram-se e processaram-se imagens de solo nu, sem qualquer tipo de cobertura acima dos 30 cm, para preservar a ilusão da vegetação rasteira. As texturas utilizadas têm uma resolução de 256x256, correspondendo a um pixel por cada 0.0137 m², o que, sem ser uma grande precisão para observação do interior de um povoamento, é suficientemente precisa para dar ao utilizador a ilusão de realidade. Esta escala também é compatível com a utilizada para a representação das árvores.

2.2 Geometria da Informação Geográfica

De acordo com o tipo de representação de terreno escolhido, assim é necessário realizar a decomposição da geometria da informação geográfica. A utilização da fotografia aérea não obriga a um grande detalhe da decomposição do espaço, podendo-se optar pela transformação da altimetria numa grelha de pontos com uma malha fixa. Na implementação deste trabalho utilizou-se, porém, uma representação pormenorizada da geometria da cobertura geográfica para a utilização de texturas individuais de solo, o que obrigou a um maior detalhe no processamento espacial. Deste modo, o tratamento da altimetria foi realizado juntamente com a decomposição espacial da cobertura produzida pela fotointerpretação. Este processamento revelou-se de alguma complexidade. Assim, após a leitura da informação vectorial proveniente do sistema de informação geográfica ArcGIS/ArcView [ESRI04], foi realizada a triangulação dos polígonos que constituem a cobertura. Seguidamente, os triângulos gerados foram

interpolados com o mapa digital do terreno para determinação da altimetria de cada vértice.

Para permitir uma divisão do espaço trabalhável pelo sistema, fez-se um agrupamento dos triângulos de acordo com células de visibilidade. Estas correspondem a quadrados de 240x240 m que constituem as unidades fundamentais do terreno. Finalmente, para permitir a utilização de modelos de sombreamento mais sofisticados, calcularam-se os vectores normais em cada vértice presente na cobertura.

3. METODOLOGIAS DE IMPLEMENTAÇÃO

A biblioteca gráfica OpenGL é uma API [Shreiner03] para o hardware gráfico que, através da linguagem ANSI C, foi utilizada neste trabalho juntamente com a biblioteca GLUT (*OpenGL Utility Toolkit*). Esta biblioteca é um conjunto de funções acessórias independente da plataforma computacional e desenhado para esconder as complexidades dos diferentes sistemas gestores de janelas [Shreiner03, Kilgard96a/b].

3.1 Representação da Geometria

Para a representação da geometria utilizaram-se *Vertex Buffer Objects* (VBOs). Estes são uma nova componente de OpenGL, adicionada para as especificações OpenGL 1.5, tendo sido normalizada pela *Architecture Review Board* (ARB) em Fevereiro de 2003. Os VBOs são vectores de vértices (*vertex arrays*) definidos na memória associada ao sistema gráfico para se conseguir um alto desempenho de desenho [ARB03]. Os VBOs funcionam, na sua essência, de forma semelhante à dos *vertex arrays*, existentes há mais tempo na API, mas oferecendo um desempenho muito melhor. No entanto, para se poder usufruir convenientemente das suas vantagens é necessária alguma capacidade de memória na RAM do acelerador gráfico utilizado. Caso essa memória não esteja disponível, o sistema OpenGL tenta reservar espaço na memória AGP que está mapeada na memória principal do computador. Essa área de memória pode beneficiar das velocidades de transferência associadas ao Bus AGP que, no caso da especificação AGP 8x, pode atingir os 2GB/s [Chen02]. Verificou-se, através de testes individuais, que os VBOs têm aproximadamente o mesmo desempenho do que é conseguido pelas *display lists*, tendo estas uma presença em memória muito mais significativa, o que, em situações de grande complexidade geométrica como é a presente, coloca problemas de desempenho devido aos recursos necessários para gerir uma maior quantidade de informação.

3.2 Transparências

A técnica utilizada para representação das árvores obriga à utilização de formas de representação gráfica capazes de lidar com problemas de transparência. Apesar de existirem várias técnicas para lidar com problemas desta natureza, todas elas exigem um compromisso relativamente à qualidade em confronto com o desempenho [Nagy00]. A abordagem seguida é uma solução intermédia, que garante um desempenho elevado sem comprometer excessivamente a qualidade das imagens produzidas. En-

volveu a definição de um filtro com base no valor do parâmetro *alpha* para cada pixel da textura a aplicar. Assim, só há escrita no *z-buffer* quando os valores do parâmetro *alpha* forem superiores a um limiar pré-definido. O OpenGL permite essa funcionalidade através de funções simples em que se especifica o valor de *alpha*, a partir do qual se desenha [Shreiner03]. Esta técnica dá resultados bons e coerentes para os casos em que os valores de *alpha* são ou 0 ou 1, sem valores intermédios. Apesar de simples, esta abordagem não é isenta de problemas. Quando é aplicada uma textura a um polígono, ela não é directamente decalcada. Esse processo de decalque é determinado pelos filtros de textura, definidos na altura da sua criação. A utilização de filtros lineares ou de *mipmaps* [Shreiner03] provoca o aparecimento de valores interpolados de *alpha* que não são nem 0 nem 1. A presença de valores muito baixos, origina o aparecimento de auréolas que salientam o contorno das árvores que estão mais próximas. Pelo contrário, valores muito elevados fazem com que as partes mais finas da imagem pareçam desaparecer quando avistadas de longe (que é precisamente o caso representado na Figura 2).



Figura 2. - Problemas na adequação de valores correctos de *alpha* para representação da geometria.

3.3 Definição do Horizonte Visual

Para limitar a quantidade de informação submetida ao *pipeline* gráfico, é fundamental definir quais as regiões visíveis, ou potencialmente visíveis, a partir do ponto de observação. Em paisagens naturais onde o horizonte se estende por vários quilómetros, é particularmente difícil definir regiões de visibilidade, tendendo a geometria a tornar-se num dos principais constrangimentos para os sistemas deste tipo. Quando se está a representar uma paisagem em que apenas importa considerar o relevo, a grelha que representa a altimetria é, muitas vezes e por si só, suficientemente grande para colocar graves problemas de desempenho. Para resolver essas situações, foram propostos na literatura vários métodos baseados no nível-de-detalle (LOD). Estes métodos procuram simplificar a geometria, de acordo com a distância ao ponto de observação. Deste modo, regiões mais longínquas são repre-

sentadas com menor detalhe do que regiões mais próximas [Muchaxo95, Lindstrom96, Rottger98]. O problema torna-se mais complicado quando existem outros objectos sobre o terreno, tais como árvores. Deste modo, a redução de detalhe de acordo com a distância não tem uma resposta óbvia, apesar de haver, neste momento, algumas propostas de resolução do problema [Zach02].

Assim, utilizaram-se várias estratégias para reduzir a informação a desenhar. A primeira e mais óbvia baseia-se na verificação da visibilidade potencial dos diferentes elementos que constituem o terreno. Em segundo lugar, optou-se por definir uma estrutura em mosaico, para aumentar a granularidade da informação geométrica. Conforme referido anteriormente, usaram-se células quadradas, com 240 m de lado (5.7 ha), como unidade básica de processamento. Cada célula contém a informação altimétrica, a geometria dos tipos de ocupação do solo e as árvores correspondentes. O teste de visibilidade incide, então, sobre as células e não sobre elementos mais básicos, como as árvores ou os diversos polígonos que constituem o terreno.

Por fim, definiu-se uma superfície de visibilidade potencial (SVP) que contém apenas a área que um utilizador, deslocando-se a velocidade moderada no terreno, poderia apreciar visualmente. A SVP corresponde a um quadrado formado por um subconjunto de células do mapa, deslocando-se na totalidade do mapa e sendo carregada dinamicamente, reduzindo assim a geometria presente na memória e a quantidade de informação a processar.

A definição do horizonte visual baseia-se no cálculo da matriz de projecção para cada *frame* e posterior comparação de cada célula presente com o espaço de visibilidade. À primeira vista parece pouco eficiente, pois obriga ao teste de todas as células potencialmente visíveis. Contudo, a extrema simplicidade do teste, associada ao facto do número de células potencialmente visíveis ser relativamente reduzido, torna o custo computacional da sua utilização bastante negligenciável. Finalizada essa operação, os vectores de especificação dos planos do frustum de visibilidade, em coordenadas do espaço do observador, podem ser calculados pelas seguintes expressões [Lengyel02]:

$$\text{Plano direito} = M_4 - M_1$$

$$\text{Plano esquerdo} = M_4 + M_1$$

$$\text{Plano próximo} = M_4 + M_3$$

$$\text{Plano distante} = M_4 - M_3$$

$$\text{Plano de cima} = M_4 - M_2$$

$$\text{Plano de baixo} = M_4 + M_2$$

Nesta notação M_i corresponde à linha i da matriz de projecção utilizada. De notar que estes valores não produzem vectores normalizados, pelo que têm de ser escalados usando-se a respectiva norma.

Tendo as equações dos planos do frustum, a decisão de desenhar ou não uma célula de visibilidade é dada por um algoritmo simples. A ideia base da sua execução é verificar se uma esfera hipotética que envolvesse comple-

tamente cada célula seria visível, total ou parcialmente. Essa verificação é feita através da determinação da distância do centro da esfera (centro geométrico da célula) a cada um dos planos e da subsequente comparação com o raio da esfera. As células assim detectadas como potencialmente visíveis terão que ser consideradas para desenho.

De notar que, apesar de o algoritmo ser aparentemente dispendioso em termos computacionais, não põe minimamente em causa o desempenho do programa. Na verdade, a determinação das equações dos planos do frustum é realizada somente uma vez por *frame* e a verificação de visibilidade é executada apenas tantas vezes quantas as células que compõem a SVP, que é substancialmente mais reduzida que o número de polígonos ou objectos que constituem cada cena.

3.4 Carregamento Dinâmico da Paisagem

O carregamento dinâmico da informação geométrica à medida que vai sendo necessária é um dos passos críticos no decorrer do programa. É este procedimento que permite a navegação virtual por cenários arbitrariamente grandes, reduzindo a informação presente na memória apenas ao estritamente necessário, que corresponde às células potencialmente visíveis a partir de um determinado ponto de observação. Idealmente, a superfície representada seria um círculo. No entanto, dado o facto de ser consideravelmente mais fácil e eficiente a utilização de um quadrado, foi essa a forma geométrica utilizada.

A técnica que se concebeu faz deslocar a superfície de visibilidade potencial sobre o terreno, juntamente com o ponto de observação. Para tal, usou-se um *buffer* definido previamente e que determina o período de avanço do horizonte. Quando o observador ultrapassa um determinado limiar (posição_inicial + *buffer*), as coordenadas reais do observador no terreno são alteradas de acordo com a velocidade e direcção do movimento, mas as coordenadas na grelha de visibilidade voltam a ser centradas. Por outras palavras, carrega-se uma ou mais filas de mosaicos de paisagem e a posição actual passa a corresponder ao centro da grelha, ou seja, é adicionado ou subtraído às suas coordenadas o valor correspondente à dimensão do *buffer* criado. Na Figura 3 descreve-se graficamente o funcionamento da metodologia proposta. Nesta situação, muito simplificada, admite-se um *buffer* com a dimensão de uma célula e um horizonte de visibilidade (h_v) de 2 células. A dimensão da superfície de visibilidade (DSV) é dada por

$$DSV = 2(\text{buffer} + h_v) + 1$$

Para este caso vemos que o quadrado que constitui a superfície de visibilidade deve ter uma dimensão de 7×7 . Os números representados na Figura 3 correspondem às coordenadas da grelha com a informação real do terreno. No início do programa o observador pode deslocar-se livremente na paisagem, no interior da superfície com as dimensões $(\text{buffer}+1) \times (\text{buffer}+1)$, de acordo com a Figura 3.a). Supondo uma deslocação para a direita até às margens do *buffer* (fim da coluna 10), o observador é recolocado no centro, a coluna 6 é removida da superfície

de visibilidade e, em seu lugar, é carregada a coluna 13, como se pode ver na Figura 3.b). De notar que, durante qualquer deslocação, o horizonte visível nunca atinge o extremo da superfície de visibilidade.

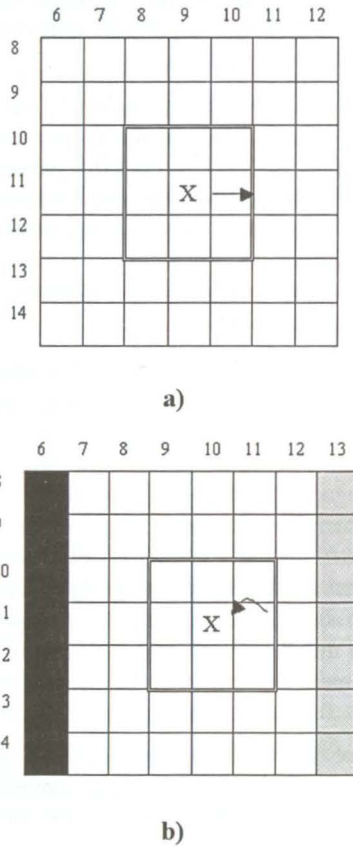


Figura 3. - Ilustração do algoritmo de deslocação da região de visibilidade.

3.5 Colocação das árvores

Uma vez que as coordenadas das árvores não estão definidas *a priori*, a colocação das árvores no terreno é feita à medida que os mosaicos em que estão inseridas vão entrando na grelha de visibilidade. É um processo que é executado automaticamente no decorrer do programa. O problema a resolver é a colocação de árvores no interior de cada triângulo que faz parte da estrutura do terreno.

A abordagem seguida neste trabalho tem associada a si mesma uma sobredeterminação de coordenadas de árvores geradas, que nunca excede o dobro do número de árvores necessárias. Mas verificou-se que este cálculo em excesso, apesar de tudo, não tem relevo no desempenho geral do programa.

A solução concebida passa por determinar, de uma forma aleatória, as posições de árvores num rectângulo que circunscreva o triângulo em causa. É possível, então, seleccionar os pontos que pertencem ao triângulo e, posteriormente, realizar a interpolação para cálculo da altimetria de cada árvore apenas nesses pontos.

Desta forma, em primeiro lugar determina-se o menor rectângulo que envolve o triângulo considerado, hipoteticamente de vértices P_0 , P_1 e P_2 (Figura 4). Para tal verifica-se qual das arestas tem a maior dimensão e calcula-se a recta correspondente $R = P_0 + rV_0$, em que V_0 é o vector $P_1 - P_0$. Esta aresta corresponde a um dos lados do rectângulo envolvente. Seguidamente determina-se o vector V_1 perpendicular à recta R , definida por esses dois pontos P_0 e P_1 , e com o comprimento igual à distância de P_2 à referida recta. O ponto P_0 e os vectores V_0 e V_1 definem um rectângulo que circunscreve o triângulo considerado. Esta figura geométrica tem uma área igual ao dobro da área do triângulo.

A colocação das árvores é então realizada sobre o rectângulo assim obtido. Conhecendo a sua área e a densidade florestal do povoamento, determina-se o número de árvores a inserir. Nesse sentido, para cada árvore, calculam-se dois números aleatórios u e v no intervalo $[0, 1[$. A posição da árvore (C) no rectângulo considerado é, pois, calculada pela seguinte expressão:

$$C = P_0 + u \cdot V_0 + v \cdot V_1$$

Através de um teste geométrico simples consegue-se verificar se estas coordenadas estão ou não dentro do triângulo considerado. No fim do processo de cálculo, para as árvores que ficaram identificadas como estando no interior do triângulo determinam-se as suas coordenadas no espaço através de uma interpolação linear das coordenadas dos vértices do triângulo.

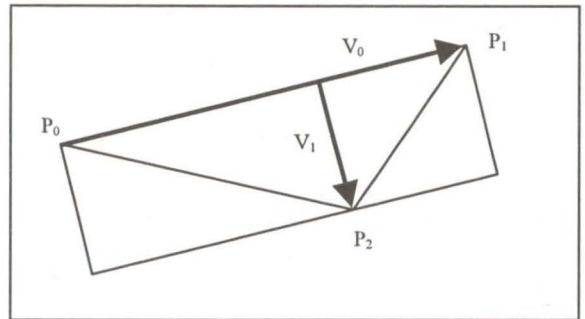


Figura 4. - Um triângulo, de vértices P_0 , P_1 e P_2 , e o rectângulo circunscrito com menor área, determinado pelos vectores V_0 e V_1 .

4. RESULTADOS

Duas componentes principais caracterizam o sistema que foi implementado. Uma delas corresponde aos módulos de pré-processamento da informação, e a outra ao ambiente de visualização interactiva. Os módulos da primeira usam a informação geográfica e florestal proveniente do SIG e do inventário ou de um sistema de apoio à decisão, transformando-a no formato requerido pelo programa de visualização. Este último, pelo seu lado, além da informação produzida pelos módulos anteriores, usa informação genérica pré-existente (por exemplo, texturas de árvores e do solo).

A fase final de desenvolvimento do sistema de visualização foi realizada num computador equipado com proces-

sador Pentium4 a 2.8 GHz, com 512 Mbytes de RAM, Front Side Bus a 800 MHz e placa gráfica ATI M10 com 128 MB de memória vídeo (equivalente a uma ATI Radeon 9600). O sistema operativo utilizado foi Windows XP.

4.1 Interface com o utilizador

4.1.1 Sistema de navegação

A interacção com o programa, depois de inicializado, faz-se por intermédio do rato e do teclado. O cursor do rato permite definir o vector de deslocação que é conseguido por meio de um mecanismo intuitivo de "arrastar e largar" sobre a janela onde é representada a cena. O teclado permite controlar a velocidade de deslocação. Através do teclado é possível controlar ainda outras funções, como o carregamento e a gravação de percursos pré-definidos.

Um outro modo de funcionamento é através de Joystick, o que permite uma utilização mais cómoda do sistema por utilizadores não familiarizados com o uso do rato. Com efeito, a partir das características inerentes a este tipo de dispositivos conseguiu-se integrar as facilidades de orientação e de mudança de velocidade, facilitando a navegação na paisagem florestal.

4.1.2 Percursos pré-definidos e benchmarking

O programa desenvolvido inclui opções que permitem ao utilizador a definição de percursos para posterior visualização e eventual avaliação do desempenho. Existem três modos diferentes de interagir no programa. O primeiro, que se inicia por omissão, é designado por 'modo interactivo', em que o utilizador tem a liberdade total de se deslocar no terreno. O segundo é o 'modo de gravação', no qual todos os movimentos executados são guardados em ficheiro. Uma terceira opção é o 'modo de reprodução'. Neste modo, de um ficheiro definido pelo utilizador lê-se um caminho previamente definido e são reproduzidos todos os passos anteriormente gravados. Esta característica serve, tanto para comunicar experiências semelhantes aos utilizadores do sistema, como para avaliar as capacidades de um determinado equipamento para executar a aplicação. Estão, assim, incluídos mecanismos que permitem avaliar a adequação de uma determinada plataforma para a execução do programa.

A utilização de mecanismos de recolha de informação posicional *frame a frame*, permite ainda a utilização do programa para realização de filmes de animação. Este efeito é conseguido através da recolha de imagens sequenciais que posteriormente poderão ser convertidas num filme através de um utilitário. Desta forma permite-se que a saída gráfica do programa seja visualizada em computadores que não tenham as capacidades de aceleração gráfica necessárias para a execução normal do programa. Uma outra vantagem de um sistema com estas características poderá ser a utilização de modelos de árvores com maior resolução, melhores modelos de iluminação e conseqüente maior realismo para o utilizador, perdendo-se, no entanto, a interactividade natural do programa.

4.2 Resultados Visuais

O ambiente de visualização produz resultados que foram descritos pelos utilizadores do programa Landscape 2.0 como representativos da realidade. Permite a navegação no interior de povoamentos florestais (Figura 5), sendo visíveis e reconhecíveis as características arbóreas das espécies florestais representadas. É bem visível, ainda, o tipo de solo adequado às espécies aí presentes.



Figura 5. - Interior de um povoamento esparso de pinheiros, notando-se, ao fundo, um povoamento de eucaliptos.

O sistema de navegação permite a deslocação livre pelo terreno, embora não detectando colisões com as árvores encontradas no caminho por não ser esse um dos objectivos de partida. Em particular, é possível ter um ponto de vista acima do copado (Figura 6), sendo então visível a estrutura da paisagem dentro do horizonte de visibilidade pré-definido (2.0 km).

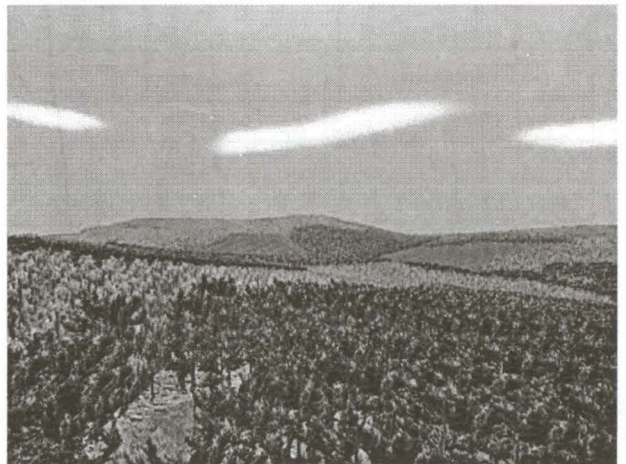


Figura 6. - Ponto de vista ligeiramente acima das copas, avistando-se diferentes tipos de ocupações florestais.

Consegue-se ter uma visão ainda mais abrangente da paisagem florestal circundante através de uma maior elevação, conforme se exemplifica na Figura 7, caso em que já é visível o reticulado da paisagem e os limites definidos pelo sistema de informação subjacente.

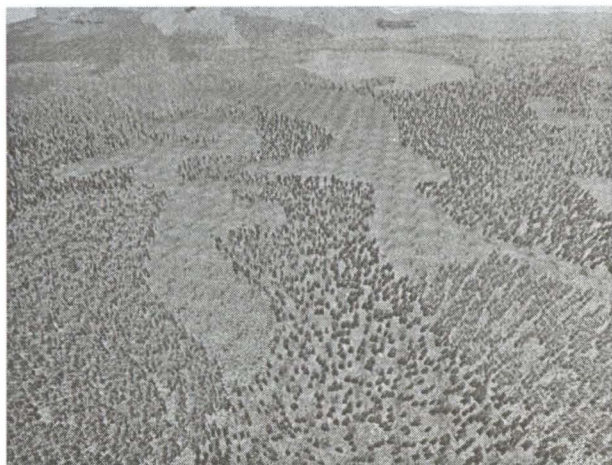


Figura 7. - Voo em altitude, sendo visível a geometria paisagística das ocupações presentes.

O sistema tem um desempenho uniforme e o *frame-rate* mantém-se relativamente constante numa utilização normal da aplicação. A entrada de novas células de visibilidade na SVP causa um ligeiro atraso, observável a um olhar treinado, mas que não é perceptível pelos utilizadores finais. Em geral, a entrada de informação nova no horizonte visual não é perceptível numa utilização comum do sistema em que os utilizadores se colocam junto ao solo ou apenas ligeiramente acima das copas.

4.3 Desempenho

4.3.1 Uso da CPU

Embora dependendo do hardware utilizado, o programa *Landscape* 2.0 permite o desenho de um número substancialmente elevado de árvores mantendo um *frame-rate*

aceitável em quase todas as situações e levando o hardware gráfico a aproximar-se dos limites superiores teóricos no que diz respeito ao número de polígonos desenhados por segundo.

Tomou-se para estudo uma área com cerca de 95,000 ha situada no Barlavento Algarvio (Serra de Monchique). Para testar a taxa de preenchimento conseguida, executou-se o programa com um caminho pré-definido. Os resultados, sumariados na Tabela 1, mostram que o programa, como se esperava, é sobretudo dependente do número de árvores que consegue enviar através do *pipeline* gráfico. Com efeito, numa das plataformas de teste verificou-se que, apesar do *frame-rate* variar consoante a resolução, o número de árvores desenhadas por *frame* se mantém constante, independentemente da dimensão da janela considerada. Conseguiram-se, assim, cerca de 125,000 árvores por *frame*, o que corresponde a mais de 1 milhão de vértices submetidos através do *pipeline* gráfico por cada *frame*. Deste modo, processam-se cerca de 34.0×10^6 vértices por segundo, o que se aproxima bastante dos máximos teóricos permitidos pelo hardware de teste.

4.3.2 Utilização da memória

A utilização da memória RAM do computador é extremamente variável, oscilando com a variação da intensidade de arborização das zonas visitadas. Ao longo da rotina de *benchmarking* variou entre os 35.0 Mbytes e os 58.5 Mbytes. De notar que esta situação corresponde a uma área de teste com uma arborização particularmente intensa (cerca de 900 árvores por ha). Para as situações mais comuns no nosso país, o número de árvores é consideravelmente mais reduzido, sendo de esperar uma presença em memória significativamente inferior.

Resolução	Imagens (s ⁻¹)	Árvores (x10 ⁶ . s ⁻¹)	Polígonos de terreno (x10 ³ . s ⁻¹)
320x240	61.33	7.673	145.2
640x480	41.86	5.235	99.07
800x600	33.28	4.165	78.78
1024x768	23.86	2.985	56.48

Tabela 1: Desempenho da execução do programa *Landscape* 2.0 a várias resoluções.

5. CONCLUSÕES

Os resultados produzidos pela ferramenta desenvolvida são suficientemente realistas para permitir uma percepção da realidade florestal pelo utilizador, mesmo para momentos distantes no tempo. O programa desenvolvido não tem, como principal objectivo, a representação exacta da realidade. Procura, em vez disso, permitir ao utilizador a recolha de uma experiência que lhe comunique o impacto visual de uma paisagem natural sem, contudo, deixar de reflectir a artificialidade do sistema.

A sua implementação sobre dados reais existentes em situações de produção em acções de planeamento florestal

facilitam a sua utilização e transposição para os utilizadores finais. Pode, então, ser utilizado para determinação de impactes visuais de planos de gestão ou para verificar situações paisagísticas futuras decorrentes de catástrofes naturais como, por exemplo, os incêndios florestais, sendo de inegável importância na avaliação dos subsequentes projectos de reflorestação.

O programa *Landscape* 2.0 apresenta um desempenho homogéneo e sem interrupções, mesmo em condições de grande densidade de arborização. Foram levadas quase ao limite as capacidades do hardware disponível, com picos de desempenho gráfico que atingem os valores teó-

ricos do equipamento. Os mecanismos de carregamento dinâmico da geometria permitem ainda a utilização do sistema em paisagens arbitrariamente grandes. No entanto, o programa tem como limitação a utilização de placas gráficas com grande quantidade de memória vídeo (>64 Mbytes) para garantir que o processamento da geometria não seja atrasado pelo acesso à memória RAM e posterior mapeamento para VRAM.

A percepção do sistema para os utilizadores apresenta também alguns problemas. O primeiro, citado por vários utilizadores do sistema, é a inexistência de pontos de referência no terreno que possibilitem a localização precisa do utilizador. Foram também referidas inexactidões na representação de algumas árvores relativamente às diferentes conformações que estas podem ter.

Deste modo, o trabalho desenvolvido até agora é um princípio promissor que poderá, contudo, ser melhorado em iterações seguintes do seu desenvolvimento. Nomeadamente, apontam-se os seguintes tópicos passíveis de continuação:

- Desenvolvimento de algoritmos de nível-de-detulhe de visualização do terreno que permitam manter a coerência espacial das ocupações florestais existentes. Este avanço permitirá a definição de horizontes visuais ainda mais extensos, reduzindo potencialmente a complexidade geométrica da cena.
- Utilização de modelos de árvores mais detalhados que permitam a inclusão mais coerente de relações dendrométricas, ao invés da abordagem fixa correntemente utilizada. O uso de técnicas de nível de detalhe, permitindo a redução até à simplificação final de dois polígonos entrelaçados, possivelmente também melhorará a qualidade das imagens criadas com um impacto mínimo no desempenho final do programa.
- Integração de elementos de referência que possibilitem a orientação espacial dos utilizadores. Se a integração de estradas ou caminhos está condicionada pela qualidade da digitalização, outro tipo de informação seria possível incluir no sistema, como, por exemplo, uma bússola ou elementos artificiais que identificassem ou situassem locais conhecidos.
- Utilização de programas de sombreamento (*shader programs*) ao nível dos vértices que constituem a geometria e dos seus fragmentos. Esta técnicas poderão aumentar o realismo dos modelos utilizados e diminuir a aparência estática dos objectos visualizados. Com estas técnicas será possível incluir, por exemplo, o efeito do vento sobre as copas das árvores e a definição de sombras sobre o terreno para diferentes condições de iluminação. Será possível, inclusivamente, fazer variar as tonalidades da folhagem de acordo com a época do ano para a qual se simula a visualização.

6. REFERÊNCIAS

[ARB03] Architecture Review Board (ARB) 2003.

OpenGL Extensions registry - Vertex Buffer Objects,

http://oss.sgi.com/projects/ogl-sample/registry/ARB/vertex_buffer_object.txt

- [Chen02] Chen, A., Johnson, J., Suhrstedt, A.. Introduction to AGP-8X. Advanced Micro Devices, Inc. One AMD Place. Sunnyvale, CA, 2002.
- [Danahy01] Danahy, J.. Considerations for digital visualisations of landscape. In: Sheppard, S., Harshaw, H. (eds). *Forests and Landscapes: Linking ecology sustainability and aesthetics*. Cabi Publishing. Wallingford. UK. 2001, pp. 225-246.
- [ESRI04] ESRI, GIS and Mapping Software (2004). <http://www.esri.com/index.html>
- [Falcão01] Falcão, A.. *Implementação de métodos heurísticos para modelação da gestão em ecossistemas florestais - Integração num Sistema de Apoio à Decisão*. Dissertação de doutoramento. Instituto Superior de Agronomia. Lisboa, 2001.
- [Falcão04] Falcão, A.. *Integração de um Visualizador Interactivo de Paisagens Florestais num Sistema de Apoio à Decisão em Gestão de Recursos Naturais - Aplicação à Serra da Lousã*. Silva Lusitana, 2004, nº especial, pp 89-101 (*in press*)
- [Kilgard96a] Kilgard, M.J.. *OpenGL programming for the X-Window System*. Addison-Wesley Pub Co., 1996.
- [Kilgard96b] Kilgard, M.J.. The OpenGL Utility Toolkit (GLUT) – Programming Interface. API Version 3, Silicon Graphics, Inc. (November 13, 1996).
- [Lengyel02] Lengyel, E.. *Mathematics for 3D Game Programming and Computer Graphics*. Charles River Media, Inc., Hingham, Massachusetts, 2002.
- [Lindstrom96] Lindstrom, P., Koller, D., Ribarsky, W., Hodges, L.F., Faust, N.. Real-Time, Continuous Level of Detail Rendering of Height Fields. Proceedings of ACM SIGGRAPH 96, August 1996, pp. 109-118.
- [Luymes01] Luymes, D.. The Rhetoric of visual simulations in forest design: some research directions. in Sheppard, S., Harshaw, H (eds). *Forests and Landscapes: Linking ecology sustainability and aesthetics*. Cabi Publishing, Wallingford, UK. 2001, pp.191-204.
- [McGaughey97] McGaughey, R.J.. Visualizing forest stand dynamics using the stand visualization system. In: Proceedings ACSM/ASPRS/RT, Seattle Washington. American Society for Photogrammetry and remote sensing. Bethesda, Maryland, Technical papers, Vol. 4, 1997, pp. 248-257.
- [Meyer01] Meyer, A.. *Représentations d'arbres réalistes et efficaces pour la synthèse d'images de paysages*. Thèse de doctorat en sciences de l'Université Joseph Fournier. Grenoble, 2001.
- [Muchaxo95] Muchaxo, J., Santos, E.M., Próspero, M.J.. Navegação em Grandes Terrenos Virtuais com Coerência Espaço-Tempo. Actas do VII Encontro Portu-

guês de Computação Gráfica, Monte de Caparica. 1995, pp. 71-84.

[Nagy00] Nagy, G.. Convincing looking glass for games. In: DeLoura, M (ed), 2000, *Game Programming Gems*. Charles River Media. Rockland Massachusetts, 2000.

[Onyx04] Onyx Computing, Inc (1992-2004). <http://www.onyxtree.com/>

[Orland01] Orland, B., Uusitalo, J.. Immersion in a virtual forest - some implications. In: Sheppard, S., Harshaw, H. (eds). *Forests and Landscapes: Linking ecology sustainability and aesthetics*. Cabi Publishing. Wallingford, UK. 2001, pp 205-224.

[Rottger98] Rottger, S., Heidrich, W., Slussallek, P., Seidel, H-P.. Real-Time Generation of Continuous Levels of Detail for Height Fields, Proc. 6th Int.

Conf. in Central Europe on Computer Graphics and Visualization. 1998, pp. 315-322.

[Sheppard89] Sheppard, S.R.J.. *Visual simulation: A User's guide for Architects, Engineers and Planners*. Van Nostrand Reinhold. New York, 1989.

[Shreiner03] Shreiner, D., Woo, M., Neider, J., Davis, T. (OpenGL Architecture Review Board). *OpenGL Programming Guide: The Official Guide to Learning OpenGL*. 4th edition, Version 1.4, Addison-Wesley (November 14, 2003).

[Zach02] Zach, C., Mantler, S., Karner, K.. Time-critical rendering of discrete and continuous levels of detail. In: Proceedings of the ACM symposium on Virtual reality software and technology. Hong Kong, China. 2002, pp. 1-8.