Interpolação de Curvas de Nível por Difusão de Calor

ROBERTO DE BEAUCLAIR SEIXAS   LUIZ HENRIQUE DE FIGUEIREDO   CLAUDIO ANTONIO DA SILVA

IMPA–Instituto de Matemática Pura e Aplicada
VISGRAF–Laboratório de Visualização e Computação Gráfica
Estrada Dona Castorina 110, 22460-320 Rio de Janeiro, RJ, Brasil
tron,lhf,cads@visgraf.impa.br

Abstract. Discutimos a geração de modelos de elevação a partir de curvas de nível inspirada em difusão de calor. A região de interesse é vista como uma placa de metal na qual cada curva de nível original é mantida a uma temperatura igual à sua cota. Quando se atinge o equilíbrio térmico, a temperatura em cada ponto da placa corresponde à altura do terreno naquele ponto. Fazemos uma simulação dessa evolução térmica resolvendo uma discretização da equação de Laplace. Os resultados são avaliados visualmente, e também analiticamente para terrenos sintéticos, e comparados com os resultados obtidos por interpolação morfológica.

1 Introdução

Num trabalho anterior [2], apresentado no GeoInfo 2000, relatamos uma metodologia para geração de modelos digitais de terreno baseados em elevação (DEM, *Digital Elevation Models*) a partir de curvas de nível. A técnica utilizada era baseada em evolução geométrica, usando morfologia matemática [1], e é resumida na Seção 3.

Este trabalho continua esse estudo, e considera uma técnica motivada por uma analogia física, a evolução da temperatura numa placa de metal.

Na Seção 2, descrevemos o problema de interpolação de curvas de nível. A Seção 3 contém um resumo da técnica de interpolação morfológica [1], e a Seção 4 descreve a técnica de interpolação por difusão de calor, cujos resultados estão na Seção 5.

2 Descrição do problema

É dado um conjunto de curvas poligonais simples e fechadas, representando as curvas de nível de um terreno. Algumas das curvas têm *cotas* associadas a elas, mas nem todas. Tipicamente, um mapa de nível contém curvas correspondentes a níveis de 20 em 20 metros, mas somente as curvas correspondentes aos níveis múltipos de 100 são explicitamente cotadas. Freqüentemente, é dado também um conjunto de *pontos notáveis*, com cotas associadas a cada ponto.

O problema então é, a partir dos dados de curvas de nível descritos acima, calcular a altura de todos os pontos numa grade regular posta sobre o terreno. O modelo de elevação que queremos é dado pelo conjunto dos valores dessas alturas, organizados em forma matricial.

Como nem todas as curvas de nível estão cotadas, o primeiro problema que aparece é determinar as cotas das curvas não cotadas. Naturalmente, a intenção de quem fez o mapa é que ele não seja ambíguo, isto é, que seja possível *deduzir* as cotas que faltam. O nosso trabalho anterior [2] descreve um algoritmo para dedução dessas cotas ausentes.

Neste trabalho, vamos nos concentrar no problema de determinar os dados de altura *entre* as curvas, assumindo que todas as curvas de nível dadas já estão cotadas.

3 Interpolação morfológica

Lembremos o método usado no trabalho anterior [2]: A partir das curvas de nível originais, rasterizadas numa imagem de resolução desejada para a modelo de elevação, curvas intermediárias eram calculadas a meio caminho das curvas já existentes. O processo era repetido até que todos os pontos da imagem fosse visitados, e portanto atribuídos uma altura.

Para calcular as curvas intermediárias, as curvas existentes eram forçadas a se expandirem naturalmente (usando o operador de dilatação da morfologia matemática [1]), até colidirem umas com as outras. Na região da colisão, definiam-se as curvas intermediárias, com a altura igual à média aritmética das alturas das curvas que colidiram.

A principal vantagem desse tipo de técnica é que as curvas intermediárias são calculadas de uma maneira que é geometricamente natural, e que funciona bem em áreas de mudança topológica (como por exemplo, em vales entre dois morros).

Embora os resultados da interpolação morfológica sejam bons, o algoritmo é lento, principalmente para imagens de alta resolução. Além disso, ele não é muito simples de implementar.

4 Interpolação por difusão

O método para interpolação de alturas entre curvas de nível dadas que apresentamos neste trabalho é baseado na seguinte metáfora física, inspirada em difusão de calor: A região do terreno que queremos modelar é vista como uma placa de metal na qual cada curva de nível original é mantida a uma temperatura constante igual à sua cota. O resto da placa está inicialmente à temperatura ambiente, digamos $0°$ (correspondente ao nível do mar). À medida que o tempo passa, o resto da placa vai se aquecendo, até que se atinge o equilíbrio térmico. Nesse momento, a temperatura final em cada ponto da placa corresponde à altura do terreno naquele ponto.

Para implementar essa metáfora física no computador, é necessário uma formulação discreta, que apresentamos a seguir.

4.1 Formulação discreta

Fazemos uma simulação da evolução térmica descrita acima resolvendo uma discretização da equação de Laplace:

$$\frac{\partial^2 h}{\partial x^2} + \frac{\partial^2 h}{\partial y^2} = 0,$$

onde $h(x, y)$ é a temperatura no ponto (x, y). Para isso, temos que aproximar as segundas derivadas de h no ponto (i, j) da matriz h_{ij} que representa as elevações do terreno.

A aproximação mais simples é dada por diferenças centrais, e a equação de Laplace "discreta" se escreve:

$$-4\, h_{ij} + h_{i-1,j} + h_{i+1,j} + h_{i,j-1} + h_{i,j+1} = 0.$$

Essa equação diz que, no equilíbrio, a temperatura de um ponto é a média da temperatura dos seus vizinhos verticais e horizontais. (Note que "média dos vizinhos" também é o objetivo da interpolação morfológica.)

Juntando a equação de diferenças acima para cada ponto (i, j), obtemos um sistema linear, que em princípio poderia ser resolvido exatamente. Entretanto, se a matriz que vai representar o terreno for $N \times N$, então temos N^2 incógnitas h_{ij}, e a matriz do sistema linear correspondente às equações de diferenças será $N^2 \times N^2$. Tipicamente, $N = 1000$, o que daria um sistema $10^6 \times 10^6$, claramente impraticável para máquinas e algoritmos de propósito geral. Talvez seja possível explorar a grande esparsidade do sistema linear, usando métodos iterativos. Não investigamos essa possibilidade, preferindo fazer uma simulação temporal direta (que provavelmente é equivalente a um método iterativo para o sistema linear).

4.2 Algoritmo para simulação de difusão de calor

O algoritmo evolve abaixo faz uma simulação temporal direta da difusão de calor. Como entrada, temos uma matriz $h = (h_{ij})$ inicializada com as alturas das curvas de nível dadas, isto é, as curvas de nível são rasterizadas nessa matriz e cada pixel (i, j) contém a altura correspondente à curva que passa por ele. Pixels nos quais não passam curvas têm altura inicial zero.

```
evolve(h):
    a ← h
    loop
        b ← a
        a ← h OVER ℒ(a)
        if a = b return a
```

Aqui, $\mathcal{L}$ é o operador Laplaciano (isto é, $\mathcal{L}(a)_{ij}$ é a média dos valores dos vizinhos de a_{ij}) e o operador OVER combina duas imagens, dando preferência à primeira (isto é, o valor de x OVER y em (i, j) é x_{ij} se $x_{ij} \neq 0$ e y_{ij} se $x_{ij} = 0$). Note que a linha

$$a \leftarrow h \text{ OVER } \mathcal{L}(a)$$

implementa a difusão do calor na placa (via $\mathcal{L}$) quando as curvas de nível dadas são mantidas a temperatura constante (via OVER). Ou seja, os valores em a são substituídos pela média dos valores vizinhos, exceto nos pixels sob as curvas de nível dadas que mantêm as alturas originais.

O algoritmo acima converge muito lentamente. Para que ele termine em tempo razoável, é necessário modificar o critério de parada para if $|a - b| < \varepsilon$ return a, onde ε é uma tolerância escolhida pelo usuário.

Infelizmente, apesar da modificação acima, o algoritmo ainda converge muito lentamente. (Isso corresponde à nossa intuição de que equilíbrio térmico perfeito demora muito para acontecer.) O principal motivo para essa demora é que os pontos que temos que interpolar têm inicialmente temperatura (altura) igual a 0, independente da posição desses pontos em relação às curvas originais. Isso faz com que muita energia seja gasta "esquentando" esses pontos até que eles fiquem a uma temperatura comparável com as curvas de nível que os cercam. Quando isso acontece, esses pontos rapidamente atingem o equilíbrio térmico. Em outras palavras, se começarmos a simulação numa situação já próxima do equilíbrio, então a convergência é rápida. Essa observação simples é a base de um algoritmo rápido para simulação de difusão de calor baseado em multiresolução, que descrevemos abaixo.

4.3 Algoritmo multiresolução

A idéia básica desse algoritmo é resolver o problema de difusão em várias resoluções, desde uma bem baixa até a resolução da imagem desejada, usando o resultado da simulação em uma resolução como entrada para a próxima. Em baixa resolução, a simulação é rápida, pois a matriz é pequena. À medida que a matriz aumenta de tamanho, a simulação vai levando mais tempo, mas não muito, pois ela já parte de uma boa solução inicial. Os detalhes estão no algoritmo m-evolve abaixo. (Para fixar as idéias, vamos supor que a resolução final é 1024×1024.)

```
m-evolve(h):
    for k = 6 to 10
        r_k ← reduce(h, 2^k)
    s ← r_6
    for k = 6 to 10
        t ← evolve(r_k, s)
        s ← duplicate(t)
    return t
```

Em palavras, esse algoritmo primeiro calcula versões reduzidas r_k da matriz inicial h. A resolução de r_k é $2^k \times 2^k$, isto é, r_6 é 64×64, r_7 é $128 \times 128, \ldots, r_{10}$ é 1024×1024, ou seja, $r_{10} = h$. Essa redução de escala pode ser feita diretamente a partir da matriz h ou então rasterizando as curvas iniciais para cada resolução (o que provavelmente dá resultados melhores). A seguir, a difusão de calor é simulada em cada resolução com evolve, usando como entrada o resultado da simulação anterior, ajustado para a resolução correta com duplicate. O algoritmo evolve foi modificado para receber as condições iniciais explicitamente:

```
evolve(h, a):
    loop
        b ← a
        a ← h OVER L(a)
        if |a − b| < ε return a
```

O algoritmo m-evolve é muito mais rápido do que o evolve original. A Figura 1 ilustra o processo de difusão por multi-resolução. A Figura 2 mostra uma visão tridimensional dos terrenos interpolados a partir desses dados. Note os terrenos gerados por interpolação morfológica é muito parecido com o terreno gerado por difusão de calor.

5 Resultados

Nesta seção apresentamos os resultados obtidos com o método de interpolação por difusão de calor e comparamos com os resultados obtidos por interpolação morfológica. A metodologia utilizada para essa comparação está dividida em duas etapas: acurácia numérica dos modelos de elevação gerados para um mesmo terreno e aspecto visual dos mesmos.

Para realizar a etapa de acurácia numérica do modelo, tivemos que usar modelos sintéticos representados por gráficos de funções analíticas do tipo $z = f(x, y)$, onde z é a altura do terreno no ponto de coordenadas (x, y). Este procedimento foi necessário porque, como já mencionamos anteriormente, o problema de interpolação é essencialmente a determinação da cota em pontos do mapa onde esta grandeza é desconhecida. Consequentemente, não temos como aferir a acurácia dos valores interpolados nos modelos gerados. Embora seja possível omitir algumas das curvas conhecidas para utilizar como controle dos valores gerados, esse procedimento não avalia completamente o terreno gerado. Em terrenos sintéticos dados por funções explícitas, conhecemos a cota em qualquer ponto da região de interesse e portanto podemos testar diretamente os resultados dos métodos de interpolação, comparando-os com valores calculados analiticamente.

Foi gerada uma imagem de teste com resolução de 300×300 contendo curvas de referência para os valores de cota iguais a 15, 64, 128, 200 e 255. Além disso, foram geradas curvas de controle com cotas iguais a 43, 90 e 130: essas curvas não foram utilizadas para a geraçao do DEM, mas serviram para avaliar a diferença entre os interpoladores. A Tabela 1 mostra a maior diferença encontrada entre as curvas interpoladas e as respectivas curvas de controle.

curvas de controle	morfologia	difusão de calor
cota de 43	5	10
cota de 90	6	10
cota de 130	2	6

Tabela 1: Diferença máxima entre as curvas interpoladas e as curvas de controle.

Analisando o fato das curvas interpoladas não terem alcançado o valor das curvas de controle, concluímos que o intervalo de resolução entre as curvas da imagem de teste foi menor que o intervalo aritmético das cotas de controle, ou seja, faltaram *pixels* para representar o intervalo altimétrico. Este problema é facilmente resolvido pelo aumento da resolução da imagem.

A etapa de inspeção visual consiste em gerar os modelos a partir de dados reais, cuja topografia seja conhecida, de modo que seja possível identificar com maior facilidade artefatos ou discrepâncias no modelo de elevação gerado. Nesta etapa, é útil fazer uma visualização tridimensional interativa, para tentar identificar artefatos no terreno que podem não ser imediatamente percebidos no mapa de alturas.

A Tabela 2 dá os tempos de cada etapa do processo de difusão de calor para os dados da Figura 1. O tempo total foi 184.46 segundos. O processo de interpolação morfológica levou 1242.51 segundos nesses mesmos dados.

A Tabela 3 dá os tempos de cada etapa do processo de difusão de calor para o terreno sintético dado por $z = x^2 + y^2$. O tempo total foi 59.25 segundos. O processo de interpolação morfológica levou 154.71 segundos nesses mesmos dados.

A Tabela 4 dá os tempos de cada etapa do processo de difusão de calor para o terreno sintético dado por $z = \sqrt{x^2 + y^2}$. O tempo total foi 108.66 segundos. O processo de interpolação morfológica levou 150.38 segundos nesses mesmos dados.

6 Conclusão

A técnica de interpolação de modelos de elevação a partir de curvas de nível usando uma simulação de difusão

resolução	iterações	tempo
64	665	0.63
128	551	2.06
256	616	10.40
512	328	23.76
1024	511	147.61
	tempo total	184.46

Tabela 2: Desempenho do processo de difusão de calor para os dados da Figura 1.

resolução	iterações	tempo
64	466	0.44
128	670	2.53
256	1155	19.21
512	512	37.07
	tempo total	59.25

Tabela 3: Desempenho do processo de difusão de calor para $z = x^2 + y^2$.

resolução	iterações	tempo
64	883	0.84
128	822	3.17
256	964	15.96
512	1226	88.69
	tempo total	108.66

Tabela 4: Desempenho do processo de difusão de calor para $z = \sqrt{x^2 + y^2}$.

de calor é simples de implementar e a sua versão multiresolução é bastante eficiente — muito mais eficiente do que a a técnica de interpolação morfológica usada no trabalho anterior [2] (pelo menos nos nossos testes).

Além disso, ela permite experimentar com variantes do operador Laplaciano $\mathcal{L}$. Originalmente, esse operador calcula a média dos 4 vizinhos principais, mas utilizamos a média de todos os 8 vizinhos imediatos e conseguimos uma simulação mais rápida cujos resultados são praticamente os mesmos.

Os resultados obtidos por difusão de calor nos parecem muito bons. Não encontramos os platôs reportados por Gousie [3]. Não temos acesso a outras implementações de geração de terrenos por difusão de calor, mas é possível que essas implementações usem somente aritmética inteira, uma vez que o objetivo é gerar imagens. A nossa implementação usa aritmética de ponto flutuante, e não gera esses efeitos de platô (com tolerâncias adequadas).

Este trabalho ainda está em andamento. As próximas etapas são experimentar com modelos “thin-plate” e outros modelos baseados em equações diferenciais parciais [3], e experimentar outros métodos de avaliação dos modelos interpolados [4].

Vale ressaltar que o objetivo deste trabalho era identificar um bom método de interpolação que tivesse um equilíbrio entre os resultados visuais e numéricos, e que fosse simples de implementar e eficiente de executar. Mais precisamente, comparar experimentalmente o desempenho do método de interpolação morfológica com o método de difusão de calor. A determinação da complexidade computacional dos métodos utilizados foge ao escopo da nossa pesquisa. Alguns comentários sobre complexidade computacional podem ser encontrados na tese de Gousie [3].

Esta pesquisa foi feita no Laboratório Visgraf do IMPA. O Visgraf é patrocinado por CNPq, FAPERJ, FINEP, e IBM Brasil. L. H. de Figueiredo é parcialmente patrocinado pelo CNPq.

Referências

[1] W. Barrett, E. Mortensen, and D. Taylor. An image space algorithm for morphological contour interpolation. In *Proceedings of Graphics Interface'94*, pages 16–24, 1994.

[2] R. de Beauclair Seixas, L. H. de Figueiredo, C. A. da Silva, and P. C. P. Carvalho. Uma metodologia para geração de modelos de elevação a partir de curvas de nível. In *Anais do GeoInfo 2000*, pages 82–87, 2000.

[3] M. B. Gousie. *Contours to Digital Elevation Models: Grid-Based Surface Reconstruction Methods*. PhD thesis, Rensselaer Polytechnic Institute, 1998.

[4] J. D. Wood and P. F. Fisher. Assessing interpolation accuracy in elevation models. *IEEE Computer Graphics and Applications*, 13(2):48–56, 1993.

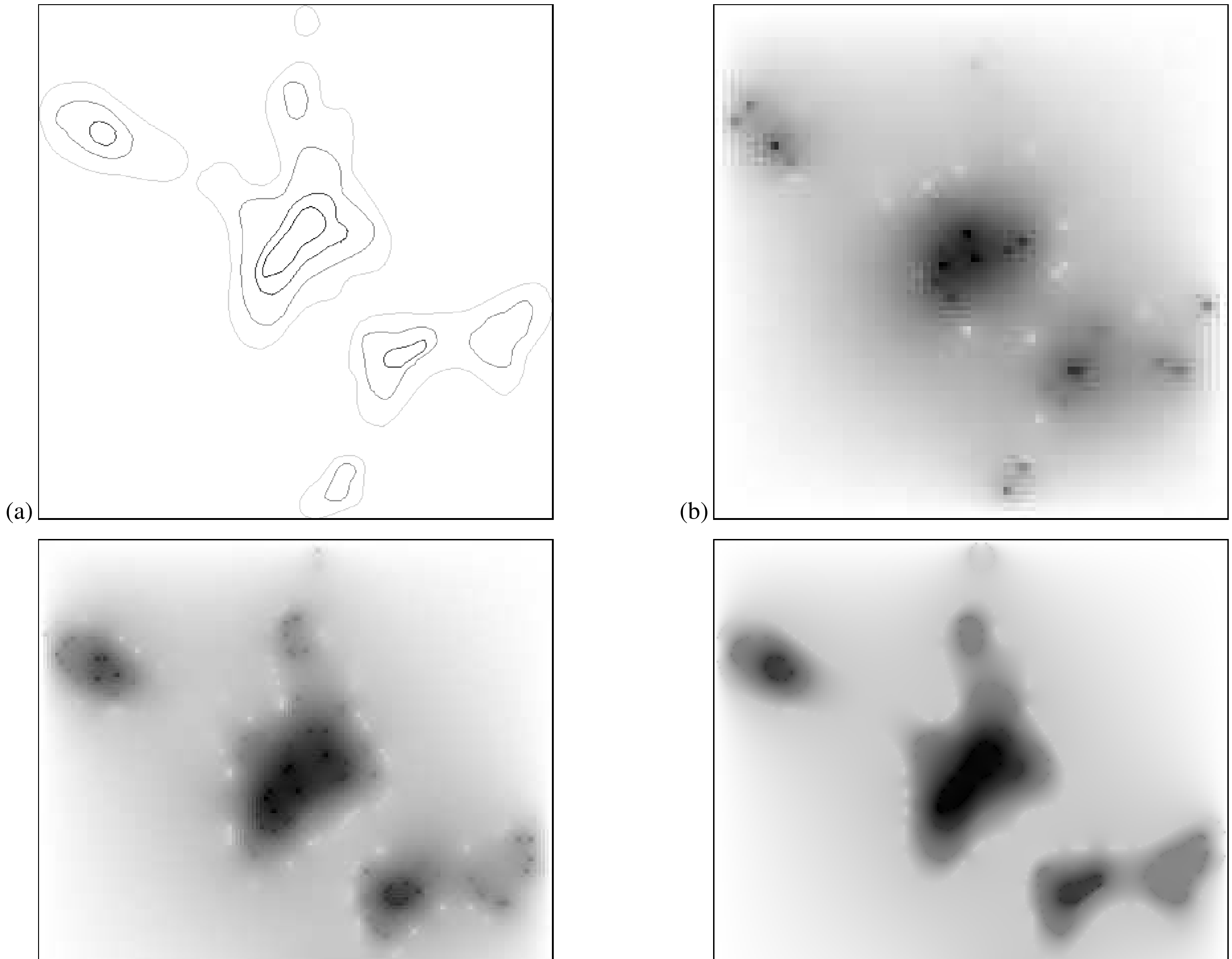
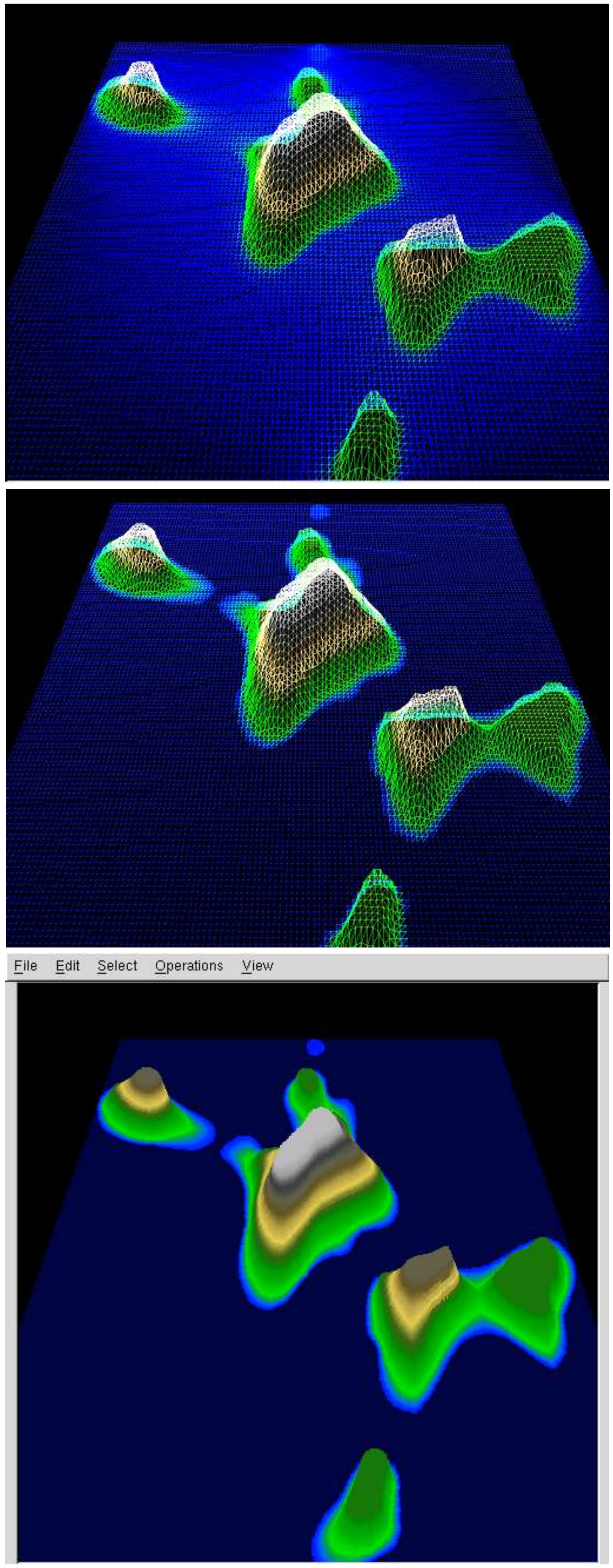


Figura 1: Exemplo de difusão por multi-resolução: (a) dados originais, (b) solução em 64×64, (c) solução em 128×128, (d) solução em 256×256, (e) solução em 512×512, (f) solução em 1024×1024.



(a)

(b)

(c)

Figura 2: Visualização tridimensional dos dados da Figura 1: (a) terreno gerado por difusão de calor; (b) terreno gerado por interpolação morfológica; (c) terreno renderizado.