6
Conclusão

Neste trabalho foi proposto um algoritmo de reconstrução de curvas a partir de pontos esparsos no plano. Este algoritmo uniu duas técnicas bem conhecidas: mínimos quadrados móveis e morfologia digital. O algoritmo aqui proposto, composto de quatro etapas distintas, apresentou bons resultados, principalmente no caso de reconstrução de curvas cujos pontos de amostragem possuem ruído elevado.

A modularidade do algoritmo proposto permite que cada uma das quatro etapas sejam implementadas de diversas maneiras, sem que isso afete as demais etapas do algoritmo: por exemplo, a etapa de agrupamento espacial dos pontos poderia ser implementada através de *quad-trees*, ao invés do grid; ou então poderíamos usar um algoritmo de projeção MLS mais preciso; seria possível, ainda, reconstruir a curva através de "segmentos" de curvas Hermite, como será visto adiante. Esta modularidade torna o algoritmo flexível para que a implementação de cada um dos módulos seja a mais adequada ao problema a ser resolvido.

A união de mínimos quadrados com afinamento topológico apresentou resutados interessantes, onde uma técnica complementa a outra. Como utilizamos uma implementação simplificada para as projeções MLS, e devido ao ruído dos pontos de amostragem, algumas vezes o cálculo da projeção de um ponto apresentava um erro bastante alto. Este problema aconteceu principalmente para pontos mais distantes da curva. Porém, a etapa de afinamento contornou este problema ao descartar as células com menos pontos de amostragem interior, realizando, assim, uma "segunda redução de ruído".

Por outro lado, não poderíamos abrir mão da utilização da etapa de pré-processamento dos pontos, pois ela é essencial para que certas características sejam garantidas durante a etapa de afinamento. Uma dessas características é a densidade dos pontos de amostragem, como vimos anteriormente. A outra característica importante é que pontos com ruído inicial muito grande podem apresentar uma distribuição espacial de tal maneira que certas estruturas topológicas indesejadas seriam geradas ao classificar as células em cheias e vazias. Como o algoritmo de afinamento preserva a topologia da imagem inicial,

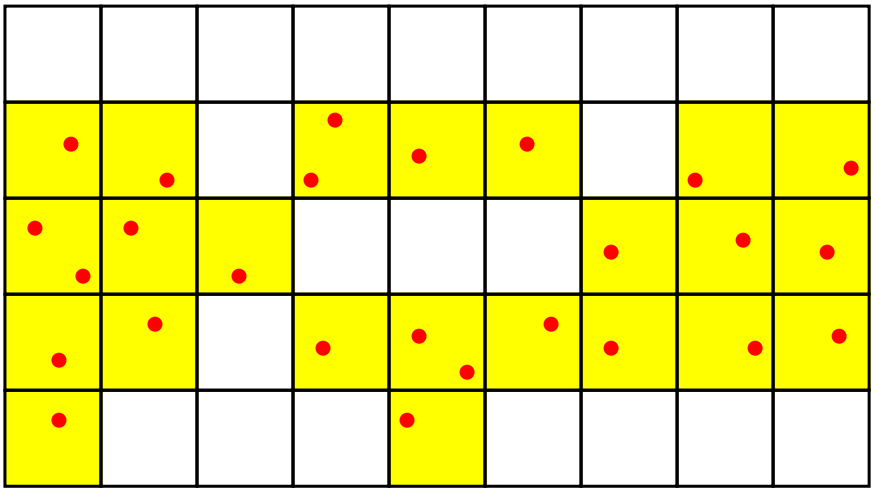

Figure 6.1: Estrutura topológica indesejada

estas estruturas não seriam removidas. Um exemplo de estrutura topológica indesejada pode ser um "buraco" no meio da curva, como pode ser observado na figura 6.1. As células em amarelo representam as células cheias. Uma possibilidade para remoção destas estruturas seria realizar a abertura seguida do fechamento do conjunto das células cheias, mas este processo pode acabar alterando a topologia da curva em lugares onde dois trechos distintos da curvas passam próximos um do outro.

Uma outra contribuição importante no afinamento é que este processo gera uma curva digital. A curva digital é um elemento extremamente importante na reconstrução, pois ela nos fornece, simultaneamente, ordenação, relação de vizinhança e localização espacial para os vértices da curva de reconstrução. Neste caso, os mínimos quadrados móveis têm a responsabilidade de ajustar à curva o vértice associado a cada célula.

Assim, de certo modo, podemos pensar no afinamento como ator de dois papéis importantes no algoritmo: o primeiro em que ele é responsável por remover o ruído remanescente da etapa de pré-processamento dos pontos, e o segundo em que ele proporciona uma ordenação e uma estrutura de vizinhança para a criação dos vértices da curva de reconstrução. Seria possível utilizar outras técnicas que executassem estes papéis, como triangulações e árvores de mínima distância. Porém, o afinamento mostrou-se uma técnica de implementação simples e de bom desempenho, tendo apresentado bons resultados. Porém, como foi explicado acima, o afinamento só funciona porque a técnica de mínimos quadrados permitiu que certas carcterísticas sobre o conjunto de pontos de amostragem fossem garantidas.

Outro aspecto interessante do algoritmo proposto é que a representação gerada para a curva normalmente utiliza menos pontos que o conjunto de

pontos de amostragem inicial (embora, durante o pré-processamento, pontos de amostragem sejam apenas acrescentados, e não removidos). É claro que a redução no número de pontos de representa a curva será tão reduzido quanto menor for o nível de detalhe desejado. Por outro lado, é possível gerar uma curva com mais vértices do que o número de pontos de amostragem inicial: basta que se escolha um grid com maior refinamento, desde que o conjunto de pontos de amostragem inicial apresente um ruído compatível com o refinamento escolhido.

Finalmente, a utilização de mínimos quadrados móveis torna capaz a extração de mais informações sobre a curva do que a simples localização espacial de cada um dos vértices da curva. Note que este processo fornece não apenas a projeção de um ponto em uma curva, mas também um polinômio que aproxima a curva nas proximidades do ponto projetado. Isto significa que, além da localização espacial do ponto projetado, podemos saber também que é o vetor normal à curva neste ponto, bem como a sua curvatura. Isto permite que outras possibilidades para a reconstrução sejam exploradas, como reconstrução através de uma curva Hermite cúbica[1], por exemplo.

Podemos ver nos resultados apresentados que o algoritmo mostrou-se rápido, mesmo tendo sido implementado em Java e não tendo sido tomada nenhuma medida de otimização. O maior problema da implementação proposta foi o consumo de memória, o que pareceu ser ainda mais crítico devido à utilização da linguagem Java, pois esta linguagem, ao contrário de outras linguagens como C, C++ e Delphi, esconde a manipulação de memória do programador. A instanciação de objetos em Java, além de apresentar um *overhead* de memória, mostrou-se um processo bastante lento.

Uma outra contribuição deste trabalho, além do algoritmo proposto em si, foi a implementação de um algoritmo de afinamento com pesos atribuído para as células. A idéia de se ordenar as células pelo número de pontos de amostragem em seu interior, e removê-las conforme esta ordem, mostrou-se simples e adequada, além de ser uma solução natural para o problema. O algoritmo recursivo de afinamento apresentou bons resultados em termos de eficiência, quando comparado ao algoritmo "tradicional" de afinamento, onde é definido apenas uma ordem de percorrimento para as células.

Como trabalhos futuros, várias melhorias podem ser desenvolvidas para este algoritmo. Uma delas é a adaptatividade, ou seja, adaptar a divisão

[1]Uma curva Hermite cúbica é uma curva paramétrica $(x(t), y(t)), 0 \leq t \leq 1$, construída a partir de dois pontos a e b que são, respectivamente, os pontos inicial e final da curva, e dois vetores v_a e v_b, que representam a direção e a velocidade da curva nos pontos a e b. A curva, derivável para todos os pontos $(x(t), y(t))$ onde $0 \leq t \leq 1$, é construída de modo que as seguintes igualdades são garantidas: $x(0) = a_x$, $y(0) = a_y$, $x(1) = b_x$, $y(1) = b_y$, $x'(0) = v_{a_x}$, $y'(0) = v_{a_y}$, $x'(1) = v_{b_x}$ e $y'(1) = v_{b_y}$

espacial de acordo com a necessidade. Assim, os lugares da curva mais ricos em detalhes estariam contidos em uma área com o grid mais fino, e os lugares com menor nível de detalhe poderiam receber menos células do grid. Outra possibilidade para tornar o algoritmo adaptativo seria ajustar o número de células próximas utilizadas no cálculo dos mínimos quadrados móveis. Este ajuste poderia ser realizado conforme foi proposto por Lee em [10]. Ele utiliza a correlação entre os pontos em uma determinada região para saber se os pontos estão "espalhados", ou se representam adequadamente a curva naquela região, possibilitando a extração de informações sobre a localização e a direção da curva.

Ainda como sugestão de modificações para o algoritmo, seria interessante implementar a reconstrução através de curvas Hermite, comparando o resultado com a curva linear por partes. Intuitivamente, a curva Hermite necessitaria de um menor nível de refinamento do grid, pois esta possui informações de normal e curvatura em cada um dos vértices. Outra implementação interessante seria eliminar a etapa de pré-processamento e realizar a reconstrução apenas através de morfologia digital. Como mostrado acima, pode ser que estruturas topológicas indesejáveis apareçam na imagem digital, mas estas podem ser removidas através de sucessivas operações de dilatação das células cheias.

Além de novas implementações, seria interessante realizar uma análise matemática mais aprofundada dos parâmetros de entrada (como densidade e ruído de pontos de amostragem) e de ajuste (como valor do h utilizado nos mínimos quadrados móveis e número de células utilizadas na divisão do *grid*), a fim de saber exatamente quais os efeitos de cada um destes parâmetros na reconstrução. Neste trabalho, foi feito apenas uma análise qualitativa dos mesmos.

Por fim, a conseqüência natural deste trabalho é a implementação do algoritmo para superfícies 3D. Alguns estudos já foram realizados neste sentido, e as maiores dificuldades foram encontradas dizem respeito ao algoritmo de afinamento e a "tradução" da superfície digital para uma malha linear por partes.

O algoritmo de afinamento tridimensional ainda é pouco estudado, e é também mais complexo que o afinamento bidimensional. A maior dificuldade está em achar os elementos estruturais para esta operação tridimensional. Ela acontece porque, enquanto temos apenas um tipo de estrutura topológica em imagens bidimensionais (componentes conectadas), temos três tipos destas estruturas nas imagens de três dimensões (componentes, túneis e cavidades). Além disso, o conceito de túnel ainda não recebeu uma definição exata,

embora já se saiba como contar o número de túneis em uma imagem com exatidão. Finalmente, a implementação de uma tabela de *look-up* para verificar rapidamente se uma célula pode ser removida pode não ser viável, já que no caso tridimensional, a vizinhança de uma célula é composta por outras 26 células (ou seja, a tabela de look-up possuiria $2^{26} \sim 67.10^6$ entradas. Alguns trabalhos analisados, como [18], apresentam uma solução para o problema.

A outra dificuldade de se passar o algoritmo para o caso tridimensional, que é a conversão da superfície digital para uma malha linear por partes, ainda não foi estudado pela autora deste trabalho. A abordagem para este problema constitui-se em traçar um paralelo entre a conectividade celular, para o caso das imagens digitais tridimensionais, e a conectividade de triângulos em uma superfície linear por partes.

O processo de mínimos quadrados móveis não apresenta maiores problemas para ser implementado para o caso de aproximação de superfícies em três dimensões. Embora o processo seja um pouco mais caro computacionalmente do que o caso bidimensional, a extensão de mínimos quadrados móveis para o caso tridimensional é imediata.