

1

Introdução

A geometria local dos objetos discretos é geralmente definida considerando a geometria dos elementos próximos. Portanto, o processamento geométrico depende fortemente de procedimentos de busca, identificando proximidades entre objetos. Algumas representações de objetos discretos, tais como reticulados regulares ou malhas, representam explicitamente as respostas para esses procedimentos. Nesse sentido, eles fornecem um extremo do balanceamento da relação tempo/memória: tempo mínimo e memória máxima. No outro extremo estão as representações de objetos como nuvens de pontos, que não armazenam nenhuma relação de vizinhança.

Em ambas as situações, estruturas de localização em malhas ou em nuvem de pontos independem da representação dos objetos em si, permitindo calcular propriedades locais de uma maneira mais eficiente. Exemplos dessas aplicações: cálculo de normais e curvaturas de nuvens de pontos; triangulação de superfícies através de algoritmos locais, tais como técnicas de mínimos quadrados móveis ou algoritmos globais usando triangulações de Delaunay; teste de colisão em animação; representação de interação física em simulações; e geração de uma representação em multi-resolução de objetos discretos.

As estruturas clássicas de busca são na sua maioria hierárquicas, por reduzirem localizações para uma complexidade média logarítmica. Elas incluem 2^d -trees, kd-trees, multi-grades, triangulação hierárquica e partições binárias do espaço. Esta tese propõe uma melhora nos procedimentos de busca para 2^d -trees, baseada numa modelagem estatística na distribuição dos nós, trabalhando numa representação otimizada do espaço, de tal forma a fornecer uma melhora no tempo de execução desses procedimentos.

Metodologia

Este trabalho foi desenvolvido a partir da seguinte observação: procedimentos usuais de busca para 2^d -trees começam da raiz para acessar um dado armazenado nas folhas, porém, já que as folhas são os nós mais distantes da raiz, por que começar da raiz? Com a representação usual das 2^d -trees, não

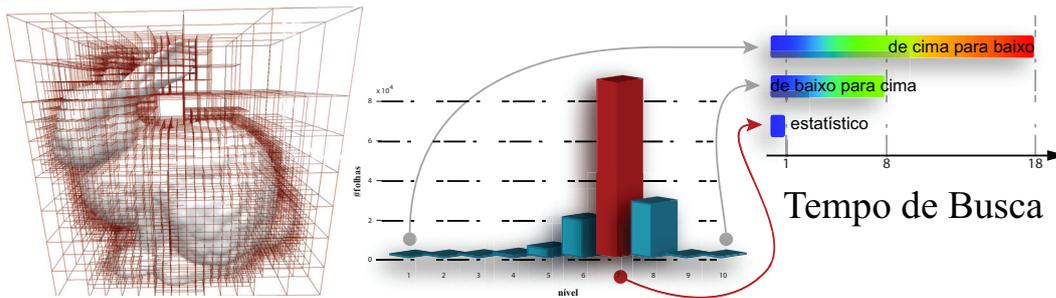


Figura 1.1: Já que as folhas são os nós mais distantes da raiz, porque começar da raiz ao procurar uma folha? Um nível ótimo inicial reduz consideravelmente os custos das buscas: (esquerda) um refinamento de *octree* que separa os vértices do modelo *Stanford bunny*; (meio) o histograma do número de folhas da *octree* por nível; (direita) custo de busca de uma folha começando de diferentes níveis.

existe outra maneira de acessar as folhas, embora uma 2^d -tree permita acesso constante aos seus nós, dada a sua posição no espaço e a sua profundidade. Procedimentos de busca recebem a posição como entrada, mas a profundidade é geralmente desconhecida. Este trabalho propõe estimar a profundidade de um nó arbitrário, através de uma otimização estatística do custo médio dos procedimentos de busca. A Figura 1.1 mostra que um nível ótimo inicial reduz consideravelmente os custos das buscas. Como o custo maior desses algoritmos ocorre quando se começa da raiz, esse método sempre reduz ao mesmo tempo a memória e o tempo de execução, através da otimização proposta.

Desenvolvimento Histórico das 2^d -Tree

O princípio da decomposição hierárquica tem sua origem incerta: foi visto primeiramente como uma maneira de agregar blocos de zeros em matrizes esparsas, mas se desenvolveu para aplicações em dados geométricos. Hoare (11) indicou a Dijkstra o atributo da decomposição de um nível de uma matriz em blocos de zeros. Morton (28) usou o significado da decomposição hierárquica para a indexação de dados geográficos. Warnock (32), em sua tese de doutorado, que marcou a literatura da computação gráfica, descreve a implementação de um algoritmo de eliminação de linha e superfície escondida usando uma decomposição recursiva da área de uma imagem. A área de imagem era repetidamente subdividida em retângulos que são sucessivamente menores, procurando assim por áreas simples para exibir. Klinger (25) e Klinger & Dyer (26), aplicaram as idéias de Warnock em reconhecimento de padrões e em processamento de imagens, enquanto Hunter (22) usou-as em animação.

O projeto Robô SRI (29) usou uma decomposição de três níveis do

espaço para representar um mapa no mundo dos robôs. A monografia de Knuth, intitulada *Art of Computer Programming* (27), cobre muitos tipos de programação de algoritmos e suas análises. No volume 3, mais precisamente, possui muitas estruturas de dados para busca. Dentre elas, existe a *binary search tree*, onde a maioria das estruturas de dados de busca conhecidas foram baseadas. Finkel e Bentley (16) casaram a *binary search tree* com o método da malha fixa, dando origem a uma composição de diretórios na forma de árvore, com células de tamanhos não-uniformes, e a nomearam de *quadtree*. Logo depois surge a *kd-tree* no trabalho de Bentley em (4), uma estrutura de dados multidimensional para busca e para um melhor balanceamento da árvore na qual é armazenada. Friedman, Bentley e Finkel em (13) criaram a *kd-tree adaptativa*, que faz uma partição adaptativa do espaço levando em conta a média dos pontos: cada nó é representado por um escalonamento de um semi-espaço.

Posteriormente, Fuchs, Kedem e Naylor em (9) criaram o *particionamento binário do espaço*, que é uma generalização da *kd-tree*, onde cada nó é representado por uma equação linear de um semi-espaço. Depois surgiram as chamadas *quadtrees* sem ponteiros, sendo uma delas chamada *df-expressions* (denotando *depth-first*), discutida por Kawaguch e Endo em (23) e por Kawaguch, Endo e Yokota em (24), representada na forma de uma lista de letras na ordem de percorrimento. As letras identificam se um nó da *quadtree* é um pai, uma folha vazia ou uma folha que contém algum conteúdo. Irene Gargantini em (7, 14) introduziu outra versão de *quadtree* sem ponteiros, chamada de *quadtree linear*, formada apenas da coleção dos nós folhas. Cada nó é representado por um par de números, sendo um o nível do nó na *quadtree*, e o outro chamado *código local*, que é formado pela concatenação de 4 dígitos, que correspondem ao código direcional da posição do nó ao longo do seu caminho até a raiz da *quadtree*.

A variação em três dimensões da *quadtree*, chamada de *oct-tree* ou *octree*, foi desenvolvida independentemente por vários pesquisadores. Reddy e Rubin em (31) propuseram a *octree* como uma de três representações de objetos sólidos. A segunda representação é a decomposição em paralelepípedos retangulares por planos perpendiculares aos eixos x , y e z . A terceira representação quebra o objeto em paralelepípedos retangulares, que não são necessariamente alinhados com os eixos coordenados, possuindo tamanhos e orientações arbitrárias. Cada paralelepípedo é subdividido recursivamente em outros novos paralelepípedos, no sistema coordenado definido pelo paralelepípedo inicial. Yau e Srihari em (33) estenderam a *octree* para dimensões quaisquer, num processo de desenvolvimento de algoritmos para manipular imagens médicas.

Hoje em dia as 2^d -trees são usadas em diversas áreas de pesquisa, sendo seu estudo de grande importância para o desenvolvimento da ciência.

Contribuição

Esta tese propõe aperfeiçoar os procedimentos de busca de uma 2^d -tree aberta, exigindo apenas quatro requisitos para o seu armazenamento e sua indexação. Para um armazenamento numa *hash table* e uma indexação baseada nos códigos de Morton, serão definidos métodos e algoritmos que satisfazem esse requisitos. Comparado com a representação clássica de uma 2^d -tree, o novo método reduz, tanto o tempo de execução por um fator médio de 4 vezes, como o consumo de memória utilizada, em média, metade da memória exigida pela representação clássica. No caso de uma 2^d -tree aberta, o método mantém o mesmo consumo de memória, mas melhora, em média, 15 vezes o tempo de execução. Embora o método aqui proposto se baseie numa modelagem estatística simples, melhora aqueles anteriores, mesmo nos piores casos. Para uma 2^d -tree aberta, o acesso do nó pela posição espacial e seu nível possui na média um tempo de execução *constante*. A idéia principal para acessar uma folha numa dada posição é, através de uma estimativa estatística, procurar por seus vizinhos. Parte desse trabalho foi publicado em (6), que se resume apenas ao caso tridimensional.