Interface Luana: uma Aplicação Gráfica para o Ensino da Árvore Binária Kd-Tree

Rodrigo D. Lima, Guilherme L.A. Mota, Paulo E. D. Pinto
Universidade do Estado do Rio de Janeiro
Instituto de Matemática e Estatística
Rua S. Francisco Xavier, 524 - Maracanã - Rio de Janeiro - RJ - CEP 20.550-013
Fone/Fax: (21) 2334-0485
dacome@gmail.com, {pauloedp, guimota}@ime.uerj.br

***Abstract*:** Spatial databases, designed for the treatment of spatial data, require algorithms and data structures capable of efficiently representing the great diversity of conceptions of space. Among the various structures used are the Kd-Trees, which are binary trees for multidimensional data with k dimensions, whose learning process is not a trivial task. This paper describes the interface Luana, an application developed in C++, supported by the Graphviz libraries and the Qt framework, with the aim of facilitating the study and understanding of this data structure, allowing the visualization, step by step, of the processes that occur during updates and searches in Kd-Trees.

***Resumo*:** Bancos de dados espaciais, voltados para o tratamento de dados geográficos, demandam algoritmos e estruturas de dados capazes de representar, de forma eficiente, a grande diversidade de concepções de espaço. Dentre as várias estruturas utilizadas estão as Kd-Trees, árvores binárias para dados multidimensionais com k dimensões, cujo aprendizado não é uma tarefa trivial. Este artigo descreve a Interface Luana, uma aplicação desenvolvida na linguagem C++, com suporte das bibliotecas Graphviz e do framework Qt, para facilitar o estudo e a compreensão dessa estrutura de dados, permitindo a visualização, passo a passo, dos processos que ocorrem durante as atualizações e buscas em Kd-Trees.

Palavras chave: Sistemas de Gerenciamento de Dados Espaciais, Estruturas de Dados Multidimensionais, Kd-Trees.

I. INTRODUÇÃO

Dados multidimensionais requerem, para sua representação e manipulação, estruturas de dados sofisticadas. Tais estruturas desempenham papel fundamental em sistemas de gerenciamento de bancos de dados espaciais (SGBDE). A eficiência das consultas espaciais, por exemplo, é dependente do desempenho das estruturas utilizadas.

Em SGBDE líderes de mercado, como o PostgreSQL [1], no tocante à manipulação interna de pontos, árvores de particionamento do espaço, como por exemplo Kd-Trees, Quadtrees e RTrees, são comumente encontradas. Dentre estas, o presente artigo se dedica às Kd-Trees, que possui grande eficiência em operações de busca.

A compreensão dos algoritmos envolvidos na manipulação das Kd-Trees não é uma tarefa fácil. Este fato motivou o desenvolvimento de ambientes web para a manipulação de tais estruturas, mais especificamente do Kd-Tree demo [2] e do Kd-Tree applet [3]. Nesses ambientes é possível visualizar o particionamento do espaço bidimensional e realizar manipulações, através da leitura de arquivos de pontos, inclusão, exclusão de pontos além de diversas modalidades de busca. Entretanto, tais aplicativos não permitem observar paralelamente o estado atual da Kd-Tree. Assim, o entendimento da interconexão entre a estrutura de dados e o particionamento do espaço fica prejudicado.

Este artigo descreve a implementação de um software didático intitulado Interface Luana, capaz de demonstrar visualmente uma Kd-Tree juntamente com o respectivo particionamento do espaço e a realização, passo a passo, de operações de busca, inserção e exclusão. Este ambiente de ensino visa conferir uma experiência de aprendizado com inspiração fenomenológica [4].

Para representar o estado atual da Kd-Tree, possibilitando a apresentação de sua estrutura de forma hierárquica, foi utilizada a biblioteca gráfica Graphviz [5]. Esta biblioteca realiza a síntese de imagens de grafos. Para tanto, deve receber como entrada um arquivo na linguagem DOT com a descrição do mesmo, incluindo vértices, arestas, texto e seus respectivos parâmetros de exibição. A Interface Luana gera o código DOT da Kd-Tree corrente que se deseja visualizar. Este código é, em seguida, remetido ao Graphviz, que retorna um *bitmap* da representação gráfica da Kd-Tree. A exibição tanto da hierarquia da Kd-Tree gerada pelo Graphviz quanto do particionamento do espaço fica a cargo do framework de desenvolvimento Qt [6], mais especificamente de suas classes de visualização. O Qt é um ambiente de desenvolvimento de aplicativos multiplataforma e que, entre outras funcionalidades, provê bibliotecas para suporte à computação gráfica.

O artigo está organizado da seguinte forma. A seção II apresenta a fundamentação teórica, a seção III descreve brevemente as bibliotecas de computação gráfica utilizadas. Na seção IV, é apresentada a implementação da Interface Luana e, então, na seção V os resultados experimentais são analisados. Por fim, na seção VI são fornecidas as conclusões e sugestões de trabalhos futuros.

II. FUNDAMENTAÇÃO TEÓRICA

Segundo Samet [7], a Kd-Tree é uma árvore binária relacionada a coordenadas de k dimensões. Esta estrutura multidimensional testa, em cada nó percorrido, um valor chave que indica a dimensão utilizada na partição do espaço. Esse valor subdivide os nós restantes nas sub-árvores esquerda e direita, sendo necessária apenas a análise de uma coordenada a cada visita de um nó. Cada nível da árvore compartilha o mesmo valor chave. Ao longo dos níveis da árvore, os valores chave variam por alternância circular.

A. Estrutura da Kd-Tree

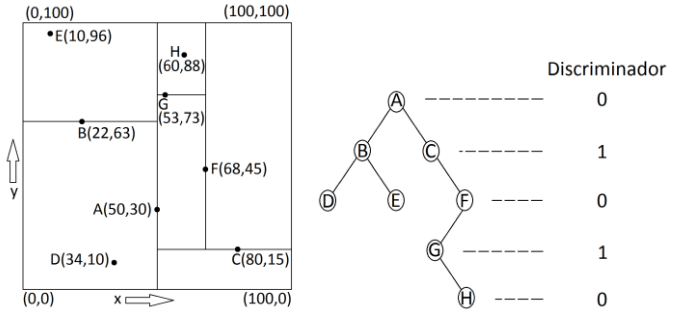
Os nós de uma Kd-Tree são constituídos por $k + 4$ campos. Os k primeiros elementos representam as coordenadas multidimensionais do ponto. Os quatro campos adicionais são usados para armazenar um ponteiro para a sub-árvore esquerda, outro para a sub-árvore direita, um campo para o

identificador do nó e o último identifica a dimensão utilizada na partição do espaço.

B. Princípio de Funcionamento da Kd-Tree

Cada nível da árvore refere-se a uma dada dimensão. Assim, todo nó obedece à seguinte propriedade: os pontos com valores da dimensão chave menores que o nó referência ficam na sub-árvore esquerda e os demais na sub-árvore direita. A dimensão que será utilizada como discriminador é estabelecida de forma alternada pelos diversos níveis da árvore binária.

A Figura 1 apresenta uma Kd-Tree com k = 2, para 8 pontos, A(50,30), B(22,63), C(80,15), D(24,10), E(10,96), F(68,46), G(53,73) e H(60,88), incluídos nesta ordem na árvore e sua respectiva partição do espaço bidimensional. O lado direito da figura mostra a representação da árvore obtida.



(a) (b)

Figura 1 - Exemplo da representação de uma Kd-Tree.

B. Processo de Inclusão

A inclusão de pontos em uma Kd-Tree [7, 8] é semelhante à inclusão de dados em uma árvore binária de busca, com a diferença de que a árvore é percorrida em função do discriminador. A cada nó percorrido, verifica-se se a busca deve continuar pela esquerda ou pela direita, a partir da comparação da coordenada do nó a ser inserido com a coordenada do nó atual. A coordenada a ser verificada é dada pelo discriminador do nó atual, desviando para a esquerda quando o valor a ser inserido for menor e para a direita, caso contrário. Quando o ponteiro para a sub-árvore a ser percorrida for nulo, então o novo nó é inserido nesse local e o ponteiro do nó da árvore é atualizado, considerando a inserção.

C. Processo de Exclusão

Nesta apresentação, k é igual a dois. Ainda assim, a exclusão em uma Kd-Tree é mais complexa do que em árvores binárias de busca. O processo é executado recursivamente, sendo necessário considerar três situações:

1. Se as sub-árvores esquerda e direita do nó a ser excluído são nulas, basta eliminar esse nó.

2. Se o nó a ser excluído possui sub-árvore direita não nula, o processo de exclusão é feito da seguinte forma: seja **P** o nó a ser excluído e d seu respectivo discriminador, devemos substituir **P** pelo nó **Q** à direita de **P**, cuja coordenada na dimensão d seja mínima. Então, realiza-se recursivamente a exclusão do nó **Q**.

3. Se o nó a ser excluído possui sub-árvore à direita nula, o processo de exclusão é feito da seguinte forma: seja **P** o nó a ser excluído e d seu respectivo discriminador, devemos substituir **P** pelo nó **Q** à esquerda de **P**, cuja coordenada na dimensão d seja mínima. Neste momento a sub-árvore esquerda de **P** é transferida para a posição direita de **Q** em sua nova posição. Então, realiza-se recursivamente a exclusão do nó **Q**, em sua localização anterior.

D. Processo de Busca

O processo de busca é feito de forma semelhante à busca em árvores binárias de busca. Nesta apresentação, k=2 e a busca é feita para a vizinhança de um ponto.

Seja a circunferência **C** expressa pela equação $r^2 = (x - a)^2 + (y - b)^2$ centrada no ponto (a, b) com distância r e (x, y) as coordenadas do nó corrente. Partindo da raiz:

1. Se o ponto representado por este nó estiver contido na circunferência C, ele é reportado;

2. Se o nó estiver em um nível par:

   a. Se $(a - r) < x$, aplicamos recursivamente o algoritmo de busca à sub-árvore esquerda;

   b. Se $(a + r) \geq x$, aplicamos recursivamente o algoritmo de busca à sub-árvore direita.

3. Caso contrário, se o nó estiver em um nível ímpar:

   a. Se $(b - r) < y$, aplicamos recursivamente o algoritmo de busca à sub-árvore esquerda;

   b. Se $(b + r) \geq y$, aplicamos recursivamente o algoritmo de busca à sub-árvore direita.

III. BIBLIOTECAS DE COMPUTAÇÃO GRÁFICA

A biblioteca gráfica Graphviz [5] é um pacote de ferramentas de código aberto desenvolvido nos laboratórios de pesquisa da AT&T com o objetivo de renderizar grafos. O Graphviz recebe uma descrição do grafo especificada na linguagem DOT, capaz de descrever sua estrutura e formatação. Este software possui algoritmos capazes de calcular automaticamente a disposição dos nós e arestas e gerar as respectivas imagens. Dentre os projetos que se empregam o Graphviz, pode ser destacado o gerador automático de documentação de sistemas Doxygen.

O Qt [6] é um framework multiplataforma que permite, a partir de um único código fonte, compilar versões da aplicação para Windows, Mac, Linux ou outros sistemas Unix. O Qt, desenvolvido originalmente pela empresa norueguesa Trolltech, posteriormente adquirida pela Nokia, contém bibliotecas C++ de código aberto. Diversos projetos são desenvolvidos com base Qt, dentre eles o KDE. O Qt fornece uma grande quantidade de bibliotecas para implementação de software com interface gráfica incluindo suporte a diferentes frameworks de computação gráfica, incluindo o OpenGL. No presente projeto é utilizado o framework Graphic View nativo do Qt.

IV IMPLEMENTAÇÃO

Para o desenvolvimento da aplicação foi escolhida a linguagem de programação C++, o framework multiplataforma Qt que suporta código em C++ e fornece uma grande quantidade de bibliotecas para implementar software com interface gráfica e a biblioteca Graphviz, para gerar grafos codificados na linguagem DOT.

Para a implementação do software de interface, além das classes que tratam da parte visual, foram desenvolvidas duas

classes que criam e manipulam uma Kd-Tree com duas dimensões. São elas: as classes *kdtree* e *kdno*. Vale citar a classe *MainWindow* que trata da interface gráfica e de todos os eventos ocorridos nela. Esta classe é criada automaticamente ao iniciar um novo projeto no Qt.

Na implementação do software de interface, o Graphviz foi utilizado como uma biblioteca, adicionando ao projeto as bibliotecas *libgvc.so* e *libgraph.so* e acrescentando o header gvc.h. No processo, os gráficos são gerados diretamente em memória principal, sem que nenhum arquivo de imagem seja criado em memória secundária para posterior abertura e exibição. Para que isso ocorra é gerado um array de char (char*) contendo o código DOT do grafo a ser desenhado, em seguida esse array é passado para o Graphviz para que seja renderizado. O Graphviz, então, entrega a imagem codificada em outro array de char.

Para exibir os gráficos do resultado do particionamento do espaço e da representação hierárquica dos nós de uma Kd-Tree no software de interface, foram utilizadas algumas classes do Qt, as mais relevantes são: QGraphicsView, QGraphicsScene, QGraphicsItem, QByteArray e QPixmap.

Para gerar o gráfico do resultado do particionamento do espaço, foram inseridos objetos da hierarquia de classes de QGraphicsItem (QGraphicsRectItem para representar pontos e QGraphicsLineItem para representar linhas de particionamento) em um QGraphicsScene, e posteriormente este QGraphicsScene é exibido em um QGraphicsView.

Para gerar o gráfico da representação hierárquica dos nós da Kd-Tree, o array de char contendo a imagem renderizada pelo Graphviz é convertido em um QByteArray e, depois, em um objeto da classe QPixmap. O QPixmap gerado é inserido em um objeto QGraphicsScene, e postriormente este QGraphicsScene é exibido em um objeto QGraphicsView.

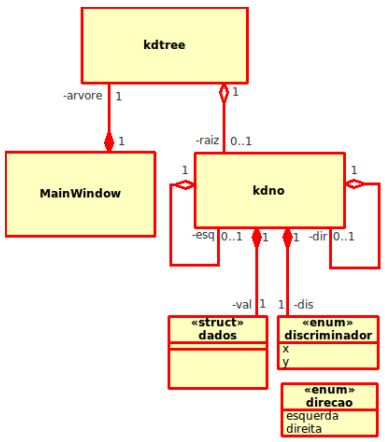


Figura 2 – Diagrama de Classes

A. Classes kdtree e kdno

Para a classe *kdno*, foi atribuída a responsabilidade de modelar um nó de uma Kd-Tree com duas dimensões, dando a ele os atributos necessários para armazenar todas as informações pertinentes para se manipular e utilizar uma Kd-Tree, e métodos em que é possível realizar comparações ou obtenção de dados de um nó.

A classe *kdtree* trata da manipulação, construção, utilização e de todos os métodos e atributos pertinentes ao gerenciamento de uma Kd-Tree.

Estão implementados na classe *kdtree*, todos os acessos ao Graphviz para gerar a imagem da representação hierárquica dos nós de uma Kd-Tree, bem como a geração dinâmica do código DOT passado ao Graphviz. Encontra-se também implementado nesta classe o preparo das imagens de saída nas instâncias dos objetos da classe QGraphicsScene que serão exibidos em instâncias da classe QGraphicsView do Qt, apresentadas como saída no software de interface.

B. Diagrama de Classes

Através do esboço do diagrama de classes, mostrado na Figura 2, é possível observar as relações entre as classes *MainWindow*, *kdtree* e *kdno*.

C. Descrição da Aplicação

A aplicação foi denominada Luana e a interface gráfica para o usuário consiste de um menu e três janelas: uma para exibir a representação hierárquica dos nós, uma segunda para exibir o resultado do particionamento do espaço, e a terceira contendo o conjunto de abas com as funções de manipulação da Kd-Tree e um menu. A Figura 3 detalha estes itens. As janelas foram implementadas dentro de um dock widget através da classe QDockWidget do Qt. Assim, o usuário pode desencaixar qualquer das janelas da tela principal, arrastar para onde quiser redimensionar seu tamanho conforme sua necessidade ou, ainda, deslocá-la para a exibição em outro monitor.

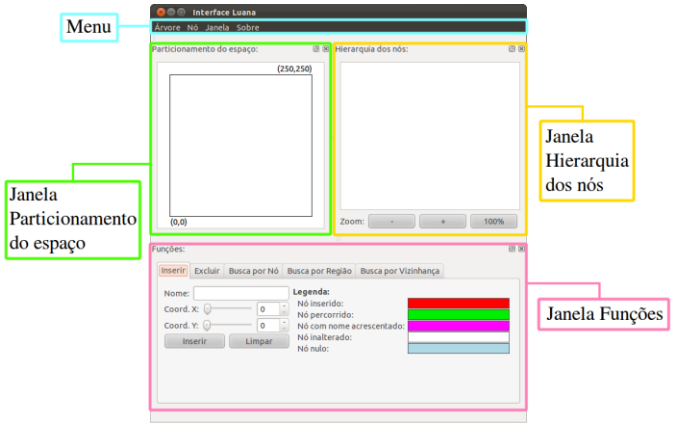


Figura 3 - Detalhamento da Interface do Luana.

C.1 Funções de manipulação

Na janela Funções está o conjunto de abas de funções de manipulação da Kd-Tree. A Figura 4 mostra o detalhamento da aba Inserir.

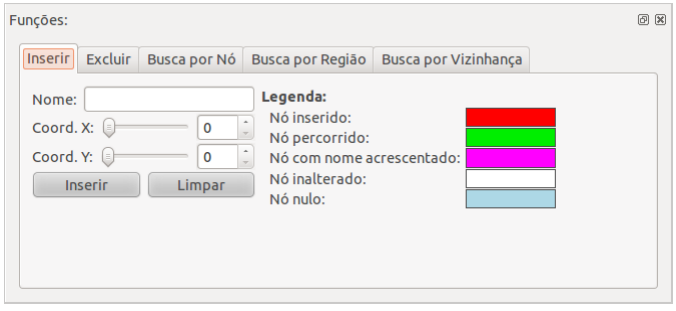


Figura 4 – Aba Inserir

C.2 Particionamento do Espaço

Nesta janela de representação gráfica é exibido o primeiro quadrante de um plano cartesiano constituído por dois eixos (*x* e *y*) de valores definidos entre 0 e 250, em que o ponto (0,0) é a origem. Nela é exibida a localização de cada nó de uma Kd-Tree representado por um ponto. É também mostrada a partição que cada nó provoca no espaço.

Ao posicionar o ponteiro do mouse em cima de um ponto nesta janela, as informações: nome e coordenadas do nó ao qual este ponto representa aparecem em uma dica na tela.

Cada nó não folha da Kd-tree divide o subespaço em que se encontra em dois novos subespaços, um com valores menores que o de sua coordenada referenciada pelo discriminador do nível do nó e outro com valores maiores ou iguais que o de sua coordenada referenciada pelo discriminador do nível do nó. Esta divisão é representada graficamente através de uma linha traçada perpendicularmente ao eixo referenciado pelo discriminador do nível do nó passando pelo ponto que representa o nó no espaço. O espaço em que se encontra o nó é limitado pelo nó pai e este nó pai, por sua vez, é limitado por seu nó pai e assim sucessivamente, até o nó raiz que subdivide espaço total em dois. A Figura 5 mostra um exemplo do espaço particionado após a inclusão dos mesmos pontos **A** a **H**, relativos à Figura 1.

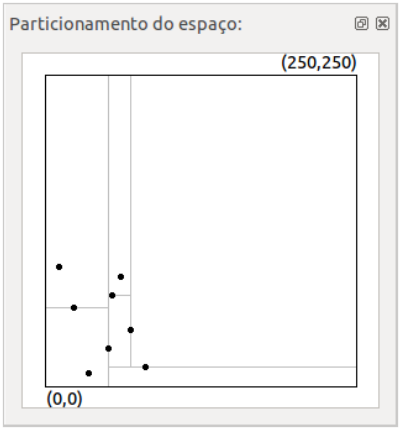


Figura 5 – Particionamento do espaço na Kd-Tree.

C3. Hierarquia dos Nós

Nesta janela são exibidos os nós da Kd-Tree organizados de forma hierárquica, onde a raiz é sempre representada como o nó mais ao topo da janela. Cada nível da hierarquia é representado com setas que saem do nó pai e vão até os nós filhos. O filho esquerdo, se houver, é representado um nível abaixo deslocado para a esquerda e o filho direito, se houver, é representado um nível abaixo deslocado para a direita.

A representação de um nó usa a forma de uma elipse com fundo branco, contendo os dados do nó como o nome, as coordenadas *x* e *y* além do discriminador. Os nós nulos são representados em forma de uma elipse com fundo azul claro, contendo as coordenadas do nó ao qual o nó nulo pertence e se é um nulo referente ao filho esquerdo ou direito. A Figura 6 mostra o exemplo de um nó e seus nulos representados.

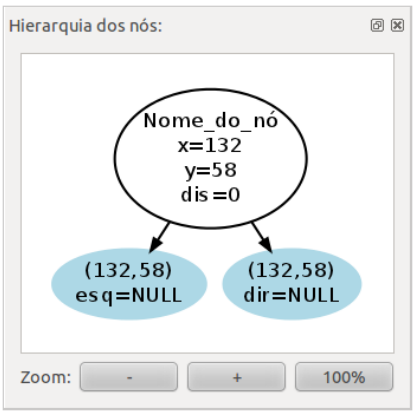


Figura 6 – Exemplo da Janela Hierarquia dos nós.

C4. Guias nas buscas

Foram implementados três processos de busca: busca simples, busca por região retangular e busca por vizinhança.

Nos processos das busca, todos os nós encontrados têm suas cores alteradas, facilitando a visualização dos resultados obtidos. Na busca por região retangular, um retângulo guia é desenhado representando graficamente a região a ser buscada, sempre que o usuário definir as coordenadas dos dois pontos que delimitam a referida região que se deseja buscar. Na busca por vizinhança em torno de um ponto, ocorre a mesma representação, mas, neste caso, é desenhada uma circunferência guia, com raio definido pelo usuário, em torno do ponto passado.

C5. Realce no passo-a -passo dos algoritmos

Nos processos de busca, inclusão e exclusão, os nós envolvidos têm suas cores alteradas, evidenciando passo a passo a execução dos algoritmos. Todas essas representações tornam possível a compreensão detalhada, bem como a visualização dos processos que ocorrem na Kd-Tree.

Nos processos de busca, todos os nós encontrados têm suas cores alteradas, nas janelas de particionamento do espaço e de hierarquia de nós.

No processo de inclusão, além de serem ressaltados os nós percorridos ou inseridos é também desenhada uma sequência de setas, registrando o caminho realizado ao longo do processo. Assim, é possível observar todo o trajeto de avaliações feitas pelo software até inserir ou verificar a existência de um nó.

No processo de exclusão, as mudanças nas cores dos nós afetados se dá de forma mais detalhada. A cada decisão ou alteração ocorrida no processo de exclusão, identificando o nó a ser excluído e o nó eleito como substituto, são alteradas as cores respectivamente e é desenhada uma seta, partindo do nó substituto, ligando estes dois nós. Com isso, fica registrado a situação atual da Kd-tree. O passo seguinte somente é realizado após a permissão do usuário.

Durante cada passo, é possível utilizar zoom ou movimentar a imagem da janela através de scroll ou *drag and drop* por meio do mouse. Assim, permite-se o acesso a todos os detalhes das estruturas.

V. EXPERIMENTOS E ANÁLISE

A seguir serão apresentados exemplos dos processos de inclusão, exclusão e buscas, mostrando as telas do software *Luana* resultantes destes processos. Nas subseções de A, B e C, são usados os mesmos dados apresentados na Figura 1. Na subseção D são usados dados de municípios do Brasil, fornecidos pelo IBGE.

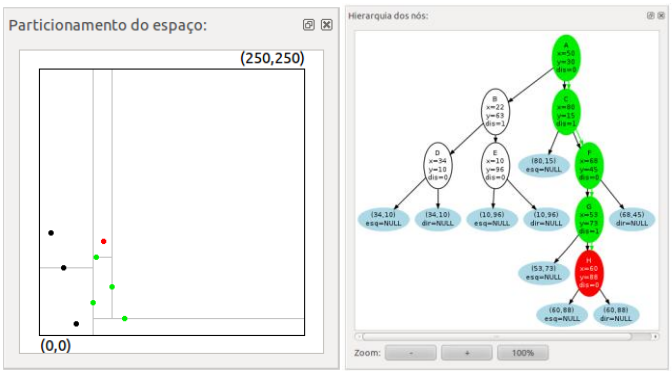


*Figura 7 – Representação gráfica da inclusão do nó **H***

A. Inclusão

A Figura 7 mostra a inclusão do nó **H** após já ter ocorrido a inclusão dos dados **A** a **G**. Exibe-se apenas as janelas de representação gráfica, com a coloração dos nós envolvidos e as setas que traçam o caminho percorrido.

B. Busca

A Figura 8 ilustra a busca por vizinhança de um ponto, realizado na Kd-Tree da Figura 1. O centro da busca é o ponto (60, 25) e o raio dado tem valor 35. A busca identifica os nós localizados dentro da circunferência C de equação $(x\text{-}60)^2 + (y\text{-}25)^2 = 35^2$. Como resultado são obtidos os pontos **A**, **C**, **D** e **F**. Pode ser notado o círculo guia na janela Resultado do Particionamento do espaço e a coloração dos pontos encontrados, nas duas janelas.

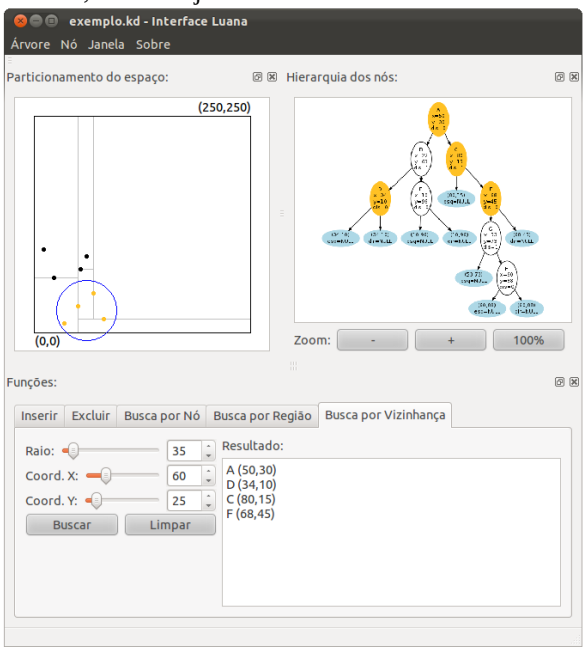


Figura 8 – Busca por vizinhança.

C. Exclusão

A Figura 9 ilustra, passo a passo, o processo da exclusão da raiz da Kd-Tree, ou seja, a exclusão do nó A.

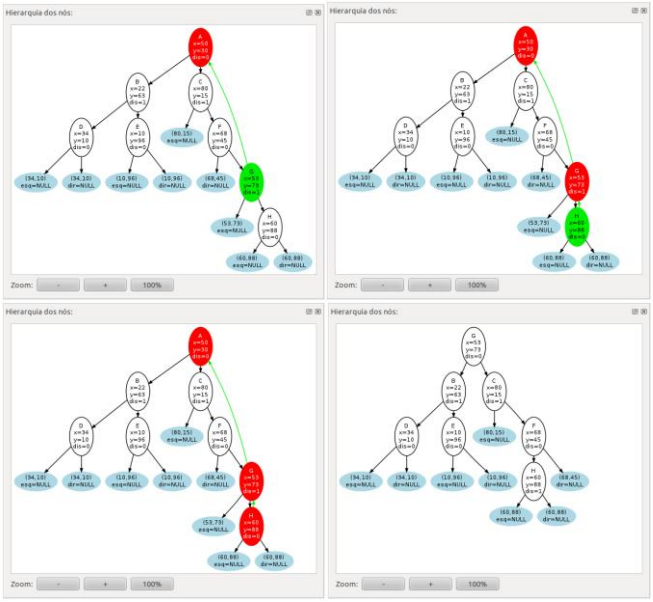


Figura 9 – Sequência de representações na exclusão do nó A.

D. Carregamento de dados de arquivo

O último exemplo do *Luana* é a utilização de dados arquivados. Foi criado um arquivo com extensão *.utm* contendo a base de dados das longitudes e latitudes em coordenadas UTM de todos os municípios do Brasil, obtida junto ao IBGE.

A Figura 10 mostra as janelas de representação gráfica para esse arquivo. É possível visualizar, na janela do particionamento do espaço, a silhueta do mapa do Brasil, principalmente nos locais onde existe maior concentração de municípios. É também exemplificada uma busca por vizinhança em torno do município do Rio de Janeiro representado pelo ponto (183,70) com r=3, mostrando-se os municípios resultantes da busca na parte inferior da janela.

VI. CONCLUSÕES

Este trabalho descreveu a ferramenta Luana, desenvolvida para a apresentação didática da árvore Kd-Tree, uma importante estrutura de dados para a representação de dados multidimensionais. Acreditamos ter disponibilizado os recursos necessários para a compreensão da estrutura, pois o usuário pode observar, passo a passo, as modificações feitas na árvore, ao longo dos processos de busca e atualização. Ao mesmo tempo o usuário visualiza, o particionamento de espaço decorrente. Essas características não tinham sido encontradas em ferramentas semelhantes.

O ambiente será utilizado como suporte às disciplinas de Computação Gráfica nos cursos de graduação e mestrado da UERJ/IME e é parte de um conjunto de aplicações em desenvolvimento para ilustrar de forma didática esse tipo de estruturas de dados.

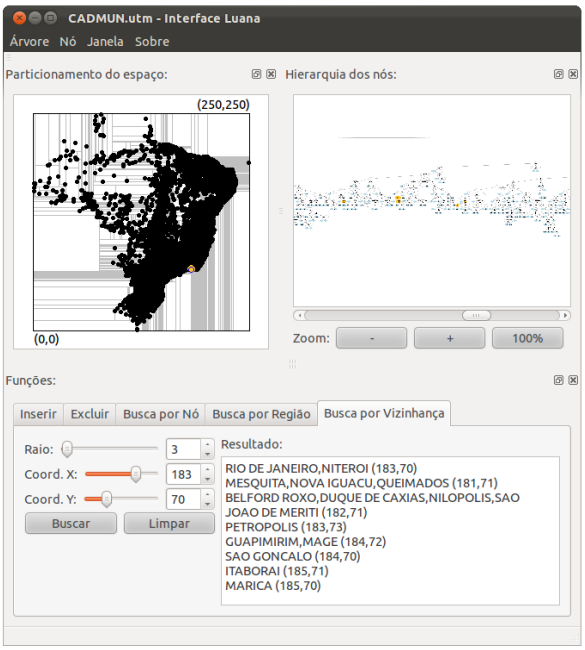


Figura 10 – Carregamento de arquivo e busca por vizinhança.

REFERÊNCIAS

[1] M. Eltabakh, R. Eltarras e W. Aref, “Space-Partitioning Trees in PostgreSQL: Realization and Performance” In Proc. of the 22nd International Conference on Data Engineering (ICDE’06)

[2] K-D Tree Demo. Disponível em:

<http://donar.umiacs.umd.edu/quadtree/points/kdtree.html>. Acesso em: 01 mai 2012.

[3] Kd-Tree Applet Demo Page. Disponível em: <http://homes.ieu.edu.tr/~hakcan/projects/kdtree/kdTree.html>. Acesso em: 01 mai 2012.

[4] Augusto N. S. Triviños, "Introdução à Pesquisa em Ciências Sociais", Editora Atlas, 1ª edição (1987), ISBN: 9788522402731, 176 p.

[5] J. Ellson, E. Gansner, L. Koutsofios, S. North, G. Woodhull, "Graphviz— Open Source Graph Drawing Tools". In. P. Mutzel, M. Jünger, Michael, S. Leipert, Graph Drawing, Isbn: 978-3-540-43309-5.

[6] Jasmin Blanchette, Mark Summerfield, "C++ GUI Programming with Qt 4", Prentice Hall; 2 edition (February 14, 2008).

[7] SAMET, H. - Foundations of Multidimensional and Metric Data Structures, Morgan Kaufmann (2006).

[8] SAMET, H. - The Design and Analysis of Spatial Data Structures, Addison-Wesley (1990).