

# Visualização da manipulação de dados em Point Quadrees

Marcos P. G. de Lima<sup>1</sup>, Guilherme L. A. Mota<sup>1</sup>, Paulo E. D. Pinto<sup>1</sup>

<sup>1</sup>Instituto de Matemática e Estatística  
Universidade do Estado do Rio de Janeiro (UERJ)  
Rua S. Francisco Xavier, 524 - Maracanã - Rio de Janeiro - RJ - CEP 20.550-013  
{marcosgaldino, guimota, pauloedp}@ime.uerj.br

***Abstract.** Quadrees are data structures used in image processing in Geographic Information Systems and as generators of meshes. This paper presents the implementation of a teaching tool for the study of Quadtree Point data structure, a specific type of Quadtree. We developed an application that graphically displays data manipulations in Point Quadrees, allowing visualization of what happens at insertion, exclusion or seek of points in this structure.*

***Resumo.** As Quadrees são estruturas de dados utilizadas no processamento de imagens, em Sistemas de Informações Geográficas e como geradores de malhas. Este artigo apresenta a implementação de uma ferramenta didática para o estudo da estrutura de dados Point Quadtree, um tipo específico de Quadtree. Foi desenvolvido um aplicativo que apresenta graficamente as manipulações de dados em Point Quadrees, permitindo a visualização do que ocorre ao inserirmos, excluirmos ou buscarmos pontos nesta estrutura.*

## 1. Introdução

A estrutura de dados Quadtree [Finkel e Bentley 1974] é uma árvore de dados em que cada nó possui exatamente quatro nós filhos, sendo uma adaptação da árvore binária para dados bidimensionais ordenados. É utilizada no processamento de imagens [Strobach 1991], como geradora de malhas [Miranda e Martha 2002], em Sistemas de Informação Geográfica (SIGs) [Samet, Rosenfeld, Shaeffer e Webber 1984], entre outras áreas. Em duas dimensões, uma Point Quadtree é uma Quadtree tal que a decomposição do espaço se dá a partir das coordenadas dos pontos na árvore.

As Point Quadrees permitem representar dados expressos por pontos no espaço bidimensional, sendo comumente usada para lidar com dados geográficos, bem como aplicações que envolvem pesquisa. O estudo desta estrutura demanda a associação entre a inserção de dados em uma árvore quaternária com um correspondente particionamento do espaço, havendo operações complexas de manipulação de dados, o que torna o seu estudo não trivial.

Em nossa pesquisa, verificamos a inexistência de ferramentas para a visualização de árvores em geral, com o detalhamento das operações de inserção, busca e deleção, bem como a ausência de trabalhos específicos relacionados a estrutura point quadtree, seja da estrutura em si ou do ponto de vista do apoio à aprendizagem da mesma.

De acordo com [Gardner, Chen e Moran 2010], os conceitos importantes devem ser ensinados de diversas formas, que devem atuar ativando diferentes inteligências ou combinações de inteligências, valorizando as diferenças entre os indivíduos. Essa afirmação mostra que o desenvolvimento de ferramentas de apoio ao ensino com o uso de recursos gráficos pode ser visto como elemento propiciador da diversificação de abordagens para o estudo em questão, respeitando a individualidade dos estudantes.

Os argumentos acima e a relevância das áreas citadas, que utilizam a estrutura de dados Quadtree, somada com a complexidade dos conceitos associados a esta estrutura mostra o quanto é importante o desenvolvimento de ferramentas que possam facilitar o seu processo de aprendizagem, criando condições para a realização de exercícios pedagógicos e também para o autoaprendizado.

### **1.1. Objetivo**

O presente artigo teve como objetivo desenvolver uma ferramenta computacional para fins didáticos do uso da Point Quadtree utilizando a linguagem C++, a IDE Qt e a biblioteca Graphviz. O didatismo desta ferramenta requer que a aplicação seja dotada de uma interface gráfica visual que representa graficamente as alterações da estrutura da árvore e do particionamento do espaço geográfico, ao longo da execução dos algoritmos de manuseio da estrutura de dados.

### **1.2. Organização do Trabalho**

Este trabalho está dividido da seguinte maneira. No Capítulo 2 é apresentada a fundamentação teórica da Point Quadtree. O Capítulo 3 descreve as ferramentas computacionais usadas no trabalho: Graphviz e Qt. O Capítulo 4 descreve a ferramenta, mostrando as classes, os requisitos satisfeitos pelo software e a sua interface gráfica. O Capítulo 5 contém os resultados obtidos com o uso do aplicativo e no Capítulo 6 são apresentadas as conclusões do trabalho.

## **2. Point Quadtree**

De acordo com [Samet 2006] e [Bentley 1974], Quadtree é uma estrutura de dados que traduz uma estrutura de árvore em um grid. A maneira como essa tradução ocorre depende do tipo de Quadtree. De maneira geral, a cada inserção no subespaço que já contiver um nó, este será dividido em quatro regiões retangulares. O presente trabalho se dedica ao estudo de um tipo especial de Quadtree: as Point Quadtrees (PQ).

A estrutura de dados Point Quadtree pode ser compreendida como uma análoga bidimensional das árvores binárias de busca. Criadas por Bentley e Finkel [Bentley e Finkel 1974], Point Quadtrees são árvores bidimensionais em que cada nó, que pode possuir até quatro filhos, representa as coordenadas 2D de um ponto.

Cada nó pai da árvore divide a área (ou subárea) definida por seus ascendentes em quatro quadrantes – situados respectivamente a nordeste (NE), sudeste (SE), noroeste (NO) e sudoeste (SO) das coordenadas 2D deste nó. Por outro lado, os nós-folha não indicam subdivisão do espaço, havendo a garantia de que se um nó não teve seus quadrantes subdivididos seus filhos são nós-folha ou nulos. Desta forma, há no máximo um ponto por quadrante. A cada inserção de um nó, se o quadrante não dividido em que o nó inserido estiver localizado já contiver algum ponto o mesmo é

desmembrado. Os quatro novos quadrantes são definidos a partir das coordenadas do ponto já anteriormente contido, como pode ser visto na Figura 5.

### 2.1. Estrutura do nó das Point Quadrees

Cada ponto bidimensional é representado na forma de um nó contendo oito campos. Esses campos guardam quatro ponteiros dos quatro nós filhos que subdividem os quadrantes NE, NO, SO e SE, as duas coordenadas do ponto, seu rótulo e um ponteiro para uma lista de colisão.

A lista de colisão permite a inclusão de vários pontos com as mesmas coordenadas, porém com rótulos distintos. No presente trabalho não foi implementada a lista de colisão. Usou-se o artifício de representar os diferentes rótulos no mesmo campo, separados por vírgula. A Figura 1 representa essa estrutura.

PonteiroNO	PonteiroNE	PonteiroSO	PonteiroSE	CoordX	CoordY	Nome(Informações)	Ponteiro para lista de colisão
------------	------------	------------	------------	--------	--------	-------------------	--------------------------------

**Figura 1. Representação da estrutura do nó de uma Point Quadtree.**

### 2.2. Criação de uma Point Quadtree

O primeiro nó inserido é definido como raiz da árvore. Cada inserção subsequente é feita percorrendo a árvore, como em uma árvore de busca, até se chegar a um nó vazio ou a um nó com as mesmas coordenadas. Se esse nó não for vazio, o rótulo é modificado, agregando-se a nova descrição. Caso o nó seja vazio, é modificado com os novos dados e são criados quatro novos filhos, um para cada quadrante, todos nós vazios. O espaço é subdividido à medida que os nós deixam de ser nós folha. A ordem da inserção dos dados na PQ define a maneira como ocorrerá o particionamento do espaço.

Na inserção de pontos em linhas de divisão em quadrantes os limites inferiores e à esquerda de cada quadrante são fechados (fazem parte do quadrante) enquanto que os limites superiores e à direita de cada bloco estão abertos (não fazem parte do quadrante).

**Tabela 1. Pontos com seus rótulos e suas coordenadas x e y.**

Rótulo do Nó	Coordenada X	Coordenada Y
A	125	116
B	178	188
C	72	63
D	10	9
E	231	194
F	150	170
G	163	178
H	50	42

A Tabela 1 apresenta pontos com seus rótulos e suas coordenadas que exemplificarão a formação de uma Point Quadtree.

Na apresentação das árvores Point Quadtree, será adotada a seguinte convenção: nós pais serão representados por círculos brancos, nós folha por quadrados preenchidos

de preto, e nós nulos são representados por quadrados brancos. Na Figura 2(a) pode-se observar o particionamento do espaço de 250x250 para a PQ referente aos dados da Tabela 1 inseridos ordenadamente. A árvore correspondente é vista na Figura 2(b).

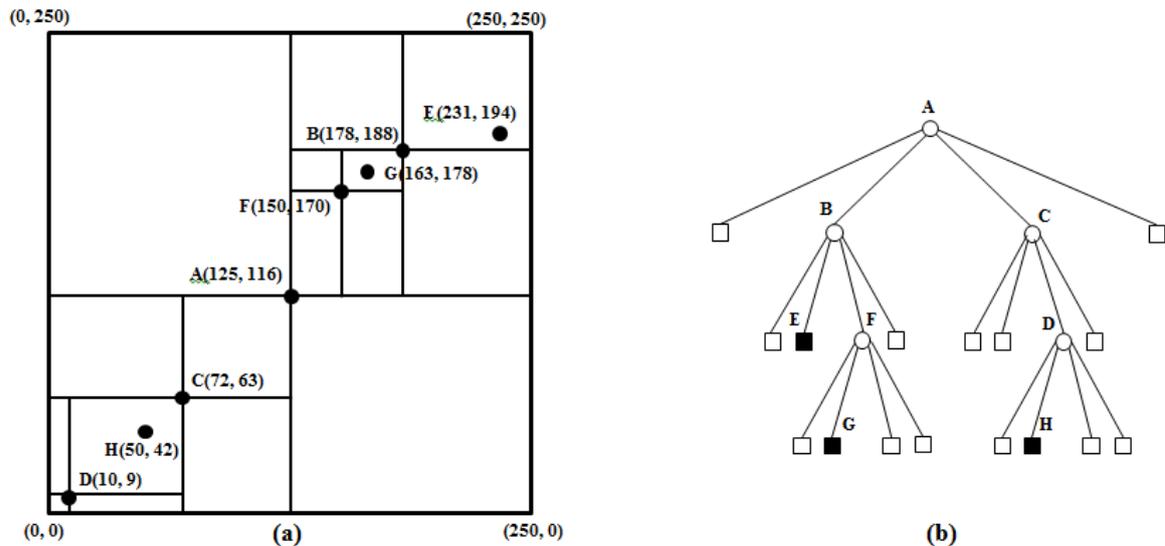


Figura 2. Formação de Point Quadtree a) Partição do Espaço b) Point Quadtree resultante.

### 2.3. Inserção em Point Quadrees

Suponha que o espaço a ser dividido tenha 250 x 250 unidades. Na primeira inserção de um ponto, este espaço será o quadrante de destino do ponto.

A inserção do primeiro ponto (125, 116) não provoca a divisão do espaço. Este primeiro nó é um nó folha e este tipo de nó não divide quadrantes nas PQs. A Figura 3 ilustra esta inserção. Este nó, por ser o primeiro a ser inserido, torna-se a raiz da árvore.

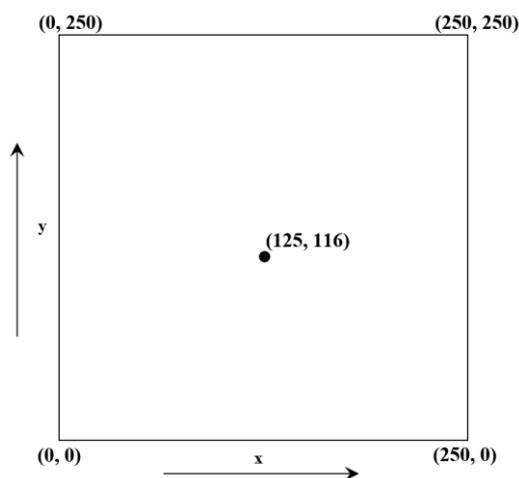


Figura 3. Inserção do ponto (125, 116) no espaço.

A representação em árvore correspondente é exibida na Figura 4.

(125, 116)



Figura 4. Árvore com o nó raiz, (125, 116).

A divisão do espaço só ocorre a partir da inserção de um segundo ponto e se dá através da divisão do espaço não delimitado em quatro espaços resultantes a partir das coordenadas do ponto pai do nó a ser inserido. Assim, ao inserir o ponto (178, 188) a divisão se dá no ponto (125, 116), cujo resultado obtido é mostrado na Figura 5.

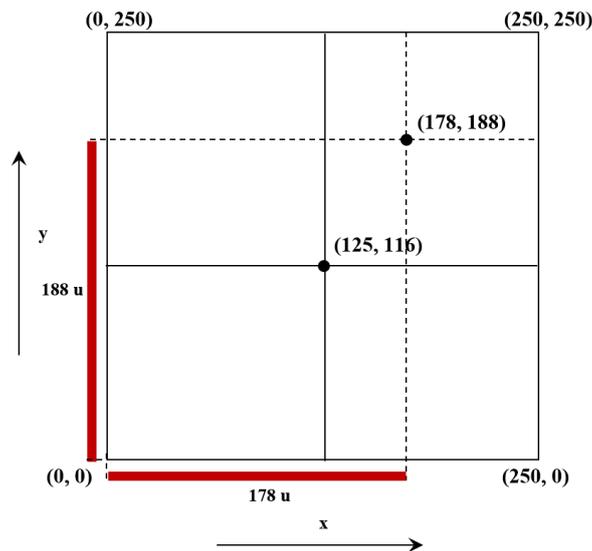


Figura 5. Particionamento após inserção do novo ponto, (178, 188), na PQ.

A visualização da árvore correspondente é mostrada na Figura 6.

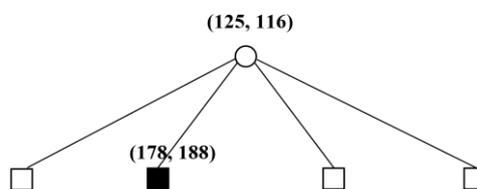


Figura 6. Árvore resultante da inserção de ponto, (178, 188), na PQ.

De acordo com [Samet 2006] e [Bentley 1975], a inserção, da mesma forma que a busca por um ponto, apresenta custo médio  $O(\log_4^N)$ , onde N é o número de pontos.

#### 2.4. Exclusão em Point Quadtree

A exclusão de nós em Point Quadtrees pode ser feita de muitas maneiras. Uma delas, sugerida por Bentley e Finkel, consiste na reinserção de todos os nós da árvore que têm raiz no nó excluído. Embora essa solução seja bem simples, é um processo muito caro, a menos que o nó a ser excluído seja nó folha ou se seus filhos o são.



menos processamento no processo de reinserção de nós. Vale observar que não há garantia de que esse “melhor” candidato seja o candidato ótimo. Samet definiu dois critérios para escolha do candidato mais adequado:

**Critério 1** - É escolhido o candidato que está mais próximo de cada um dos eixos (x ou y, a partir do nó raiz) do que outros candidatos do mesmo lado dos eixos.

Os eixos utilizados são as linhas de separação definidas a partir das coordenadas do ponto a ser excluído. O candidato no quadrante oposto não está do mesmo lado de nenhum dos eixos. Por isso, entre um nó e o seu oposto a comparação de proximidade não ocorre.

Sabe-se que, se um nó é mais próximo aos eixos que os nós que estão nos quadrantes adjacentes, o único nó que também pode apresentar essa característica é o nó que está no quadrante oposto a esse nó.

Quando houver mais de um candidato que se encaixe nas condições do critério 1 se faz necessário o uso do critério 2, que é apresentado a seguir.

**Critério 2** (“City Block metric” ou “the Manhattan metric”) – É escolhido o candidato que apresentar o valor mínimo para a métrica  $L_1$ . A métrica  $L_1$  é igual à soma dos deslocamentos a partir das fronteiras dos eixos x e y com origem no ponto a ser excluído ( $L_1 = d_x + d_y$ ).

Uma vez definido o candidato mais adequado, o nó a ser excluído é substituído por este. Em seguida, é feita a reinserção de nós situados no mesmo quadrante do substituto e nos quadrantes adjacentes ao deste nó. Não ocorre reinserção no quadrante oposto ao quadrante que contém o nó de substituição.

Através de testes empíricos, Samet observou que o número de comparações necessárias para a exclusão em point quadtrees é proporcional a  $\log_4^N$ .

## 2.5. Busca em Point Quadtrees

Neste trabalho foram implementadas as buscas por nó (simples), por região e por vizinhança. Os três tipos de busca são descritos abaixo.

### 2.5.1. Busca Simples

A busca simples em PQs é implementada como parte do processo de inserção. É verificado o quadrante em que o nó pode ser posicionado, de acordo com os valores das suas coordenadas. Dessa maneira, a pesquisa busca atingir a posição exata do nó percorrendo subquadrantes da PQ, retornando o nó encontrado ou mensagem negativa.

### 2.5.2. Busca Por Região

A pesquisa por região define um retângulo no espaço de particionamento e busca por pontos nessa região. A partir da raiz, a árvore é percorrida nos quatro quadrantes, selecionando os pontos que estão na região de busca. Caso o ponto esteja na região retangular, é guardado em um vetor.

### 2.5.3. Busca Por Vizinhança

A busca por vizinhança realiza a pesquisa em um círculo definido por um ponto (centro da busca) e um raio que define a vizinhança a ser verificada. Caso o ponto esteja na região circular, é guardado em um vetor.

## 3. Infraestrutura do Software

O desenvolvimento do Software Visualizador de PQs acompanhou a infraestrutura utilizada no desenvolvimento da Interface Luana [Dacome 2012].

Para a representação gráfica da árvore PQ, foi escolhida a linguagem C++. Foi utilizado o framework multiplataforma Qt que suporta código em C++ e fornece uma grande quantidade de bibliotecas para implementar softwares com interface gráfica. Para representar os nós de forma hierárquica, foi utilizada a biblioteca para gerar grafos Graphviz, cujos gráficos a serem gerados são codificados utilizando a linguagem DOT.

### 3.1 Graphviz

O nome Graphviz [Ellson, Gansner, et al 1988] (Graph Visualization Software) é um pacote de ferramentas, de código aberto, desenvolvido nos laboratórios de pesquisa da AT&T para desenhar grafos especificados em DOT[Gansner, Koutsofios e North 2009]. O posicionamento dos nós e arestas do grafo é feito de forma automática. Para a 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 através da seguinte linha de código:

```
#include <graphviz/gvc.h>
```

Os grafos são gerados em memória principal. É necessário que seja gerado um array de char (char\*) contendo o código DOT do grafo a ser desenhado, esse array é passado para o Graphviz para que seja renderizado. O Graphviz, então, entrega a imagem codificada em outro array de char.

Desta forma, o software de interface consegue exibir a representação hierárquica dos nós de uma Point Quadtree, através de uma imagem gerada pelo Graphviz a cada evento ocorrido.

### 3.2. Qt

Qt [Blanchette e Summerfield 2008] é um framework multiplataforma que permite compilar aplicações para Windows, Mac, Linux ou outros sistemas Unix. Suporta código em C++, possui código aberto, desenvolvido pela empresa norueguesa Trolltech (posteriormente adquirida pela Nokia). O Qt fornece uma grande quantidade de bibliotecas para implementar softwares com interface gráfica. Um exemplo de sucesso da utilização do Qt é o projeto KDE [KDE... 2013].

Para exibir os gráficos do resultado do particionamento do espaço e da representação hierárquica dos nós de uma Point Quadtree no software de interface, foram utilizadas algumas classes do Qt. Para gerar o gráfico do resultado do particionamento do espaço, foram inseridos objetos da classe 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, é obtido um array de char contendo a imagem renderizada pelo `Graphviz`, convertido em um `QByteArray` e depois gerado um pixmap em um `QPixmap`. O `QPixmap` gerado é inserido em um `QGraphicsScene`, e posteriormente este `QGraphicsScene` é exibido em um `QGraphicsView`.

#### 4. Implementação

A implementação do software de interface foi realizada com o uso da linguagem de programação C++. As duas classes principais criadas, *noQuadtree* e *quadtree*, implementam os algoritmos necessários para a representação, inclusão, exclusão em busca em PQs. A classe *noQuadtree* trata das propriedades dos nós de uma PQ, com os métodos que manipulam os dados e as ações que são feitas sobre os nós da PQ. A classe *quadtree* trata da criação da estrutura de dados e da implementação dos métodos de manipulação dos dados, desde a criação da árvore, passando pelas rotinas de inserção, busca e exclusão, até a destruição da árvore.

Além disso, a classe *QMainWindow* cuida da interface gráfica, com a criação das funções que gerenciam a janela de interação com o usuário, para todas as ações e eventos que foram planejadas na construção do software presente neste trabalho.

##### 4.1. Requisitos

Para demonstrar didaticamente uma PQ com suas operações e funcionalidades, foram elicitados os seguintes requisitos:

- A PQ deve ser gerada dinamicamente;
- Todas as imagens apresentadas em tela devem ser geradas em memória principal;
- Permitir inserir, excluir e buscar nós a partir de dados informados pelo usuário;
- Apagar completamente uma árvore;
- Apresentar em tela os resultados obtidos das buscas;
- Atualizar a tela a cada alteração na estrutura da árvore realizada através dos processos de inclusão ou exclusão;
- Permitir a qualquer momento interação com as imagens apresentadas em tela (zoom, rolagem de tela com o mouse e informações sobre os nós);
- Mostrar o caminho percorrido no processo de inclusão de um nó, permitindo interação com as imagens apresentadas em tela antes de finalizar o processo (zoom, rolagem de tela com o mouse e informações sobre os nós);
- Mostrar, passo a passo, o processo de exclusão de um nó, permitindo interação com as imagens apresentadas em tela durante o processo (zoom, rolagem de tela com o mouse e informações sobre os nós);
- Salvar o estado de uma PQ em um arquivo em memória secundária, em coordenadas cartesianas definidas pelo software;
- Carregar uma árvore a partir de um arquivo em memória secundária, podendo ele estar em coordenadas cartesianas conforme limites definidos pelo software ou em coordenadas UTM conforme limites definidos pelos fusos e zonas do território brasileiro, limites estes verificados no momento em que um arquivo é carregado.

## 4.2. Descrição da aplicação

### 4.2.1. Tela Inicial

A tela inicial do software Visualizador de PQ consiste de um menu para acesso às funções e de três janelas usadas para as operações e opções de visualização de Point Quadrees, conforme a Figura 9.

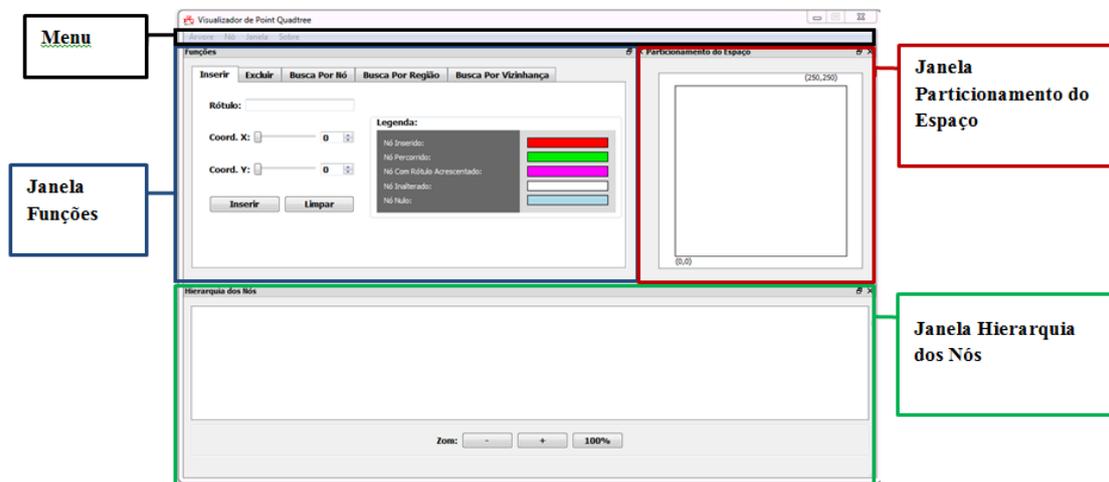


Figura 9. Tela inicial do Visualizador de PQ.

Cada uma das três janelas foi desenvolvida usando a classe QDockWidget do Qt, que permite o encaixe dentro da janela principal ou a flutuação de cada uma das janelas em um nível superior na área de trabalho.

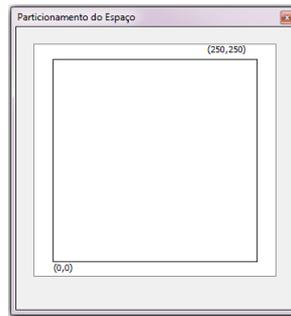
### 4.2.2. Demais telas para entrada de opções e de dados

Há várias barras de menus para escolha dos modos de visualização e janelas de Funções para entrada de dados para atualização e consulta. Esses elementos da interface não serão aqui detalhados.

### 4.2.3. Janela de Particionamento do Espaço

A janela Particionamento do Espaço mostra a disposição geométrica dos pontos inseridos em uma PQ. O espaço usado é um quadrado que admite a representação de coordenadas inteiras, sendo o canto inferior esquerdo  $(0, 0)$  e o canto superior direito  $(250, 250)$ . Todos os pontos com coordenadas inteiras com valor entre 0 e 250 podem ser representados neste espaço, como pode ser visto na Figura 10.

Optou-se por acompanhar o dimensionamento da tela de particionamento proposto na Interface Luana [Dacome 2012], uma vez que se gera uma padronização entre as duas ferramentas, possibilitando a integração futura entre os softwares desenvolvidos. A limitação das coordenadas a um quadrante de lado 250 não influencia na experiência com o software, uma vez que o seu objetivo é de ilustração das operações e algoritmos de uma PQ.

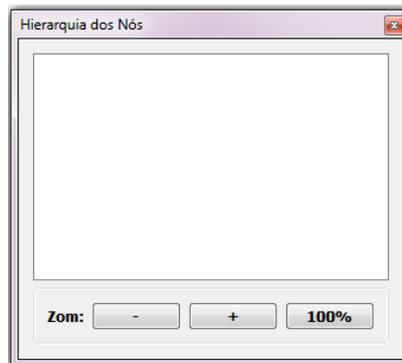


**Figura 10. Janela de Particionamento do Espaço**

Na inclusão de um novo ponto em uma PQ os nós percorridos até a inserção são realçados, através de mudança de sua cor. Na exclusão, a mudança de cor também é usada para destacar o nó que será substituído, o nó que o substituirá e todos os nós que serão reinseridos. Nos três processos de busca implementados são destacados o(s) ponto(s) encontrado(s) e a área delimitadora.

#### **4.2.4. Janela de Hierarquia dos Nós**

A janela de Hierarquia dos Nós, exibida na Figura 11, mostra a representação gráfica da árvore correspondente à PQ. Cada nó inserido é representado com seu rótulo e suas coordenadas,  $x$  e  $y$ , em uma elipse que é ligada por meio de quatro setas orientadas a cada um dos seus filhos. A orientação segue a direção “de pai para filho”.



**Figura 11. Janela de Hierarquia dos Nós**

Assim como na janela de Particionamento do Espaço, a cada nó inserido, excluído ou buscado, temos o destaque de todos os nós relacionados através da mudança de cor.

Devido ao grande espaço ocupado pela representação gráfica de uma árvore, a janela de Hierarquia dos Nós pode ser redimensionada e pode haver a alteração do nível de zoom da imagem exibida. Além disso, é possível, através da barra de rolagem acessar áreas que não estejam visíveis na geração de árvores muito grandes.

## **5. Exemplos**

Os exemplos a seguir mostram o comportamento das janelas do software mediante a inclusão, exclusão e buscas por nó em PQ, usando os dados apresentados na Tabela 1.

### 5.1. Exemplo de Inclusão em PQ

Usando os dados da Tabela 1, foi feita a inclusão em uma Point Quadtree. A Figura 12. Mostra a inserção dos cinco primeiros pontos desta tabela.

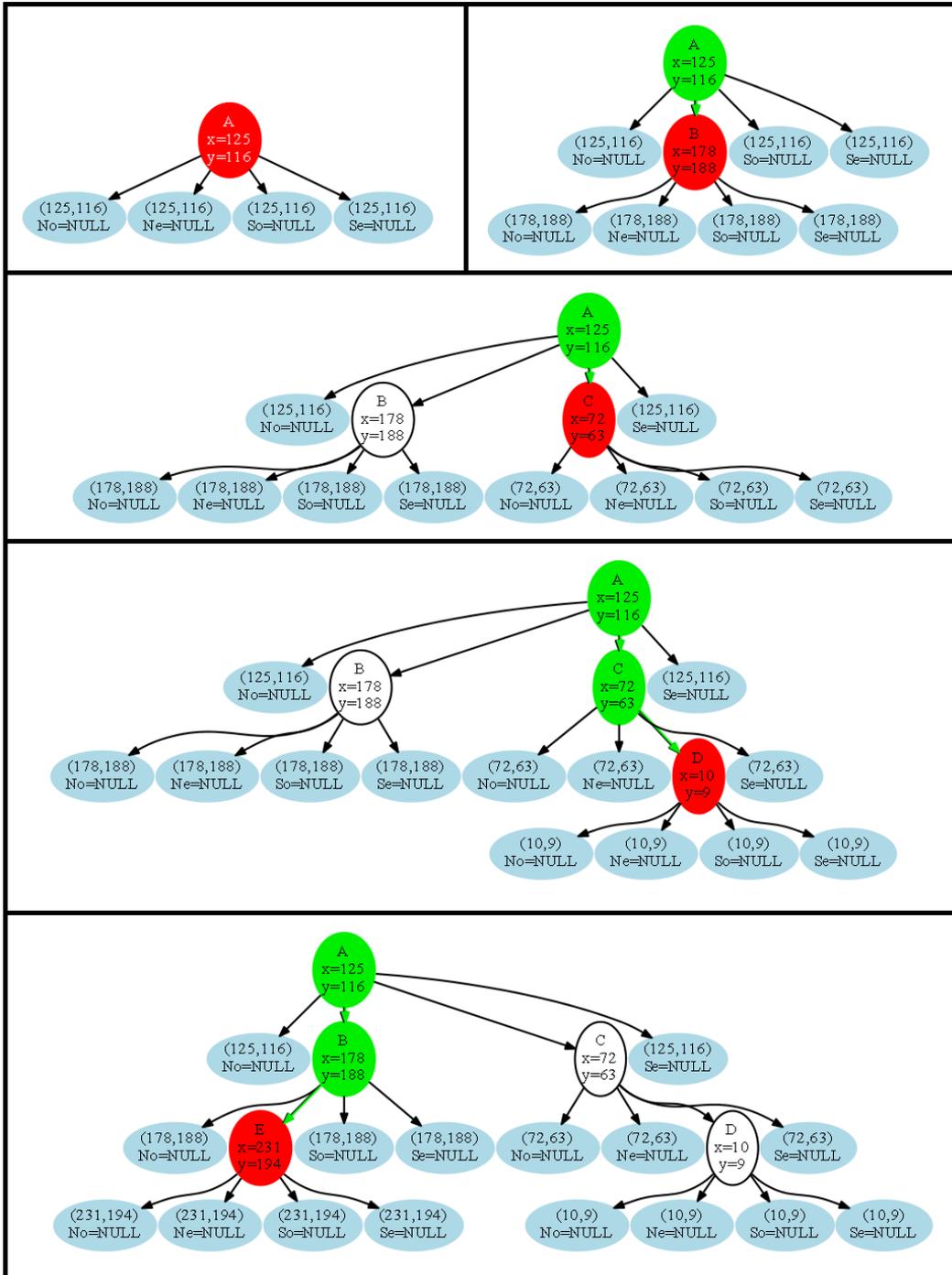


Figura 12. Inserção dos cinco primeiros pontos da Tabela 1.

### 5.2. Exemplo de Exclusão em Point Quadtree

A Figura 13 mostra a exclusão da raiz da árvore gerada com os dados da Tabela 1. No primeiro passo, o nó que será excluído e o seu nó substituto são destacados com

mudança de cor, de acordo com a legenda da aba de exclusão. No passo seguinte, além dos nós já destacados, os nós que serão reinseridos são também coloridos. Por fim, é mostrada a situação da árvore após a exclusão.

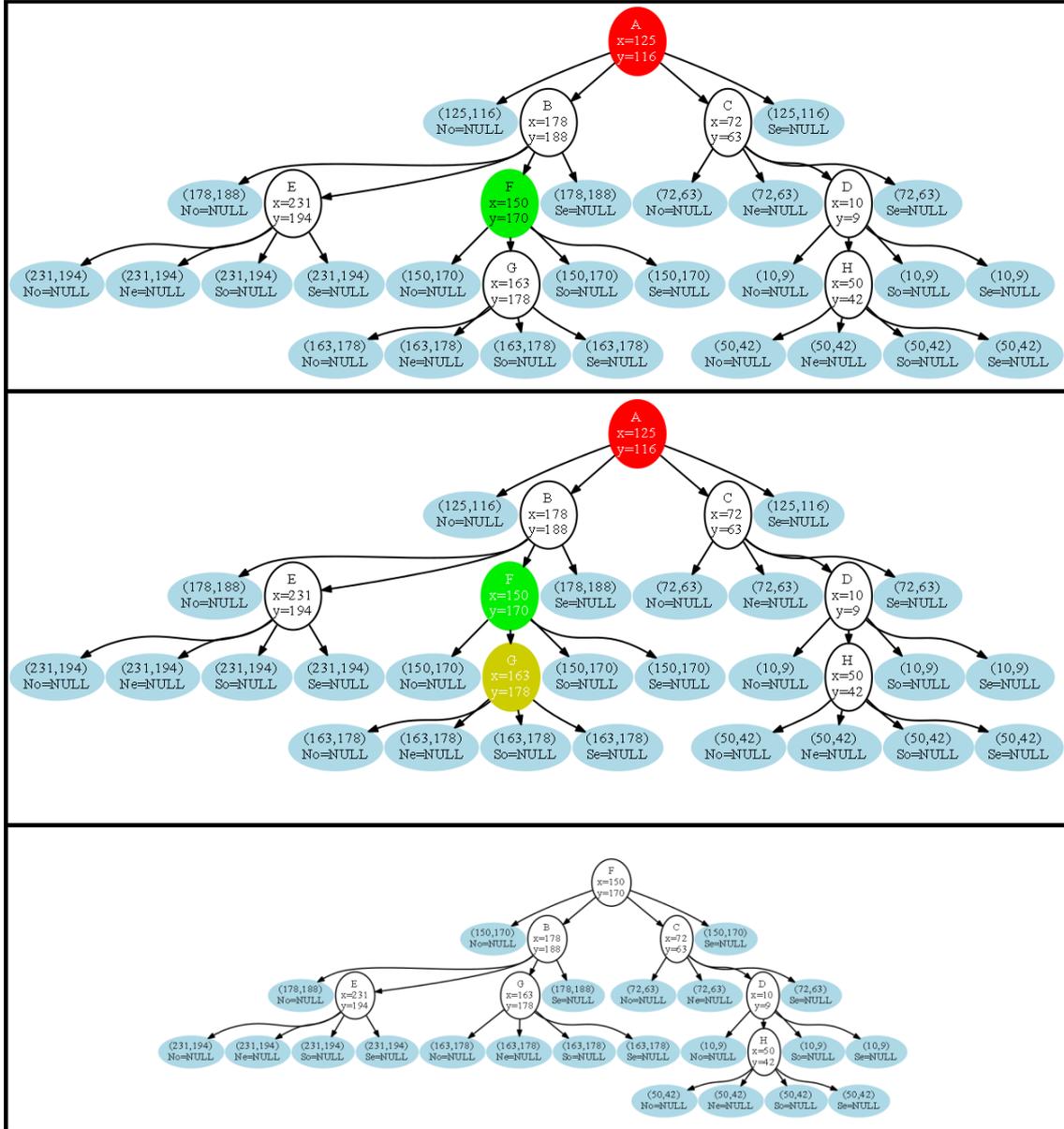


Figura 13. Exclusão do nó A (em vermelho). Substituição pelo no F (em verde). Reinscrição do nó G (Em amarelo).

### 5.3. Exemplos de Busca em Point Quadtree

Seja a árvore criada no último passo do exemplo de inclusão em Point Quadtree. A Figura 14 mostra os três tipos de busca implementados no software.

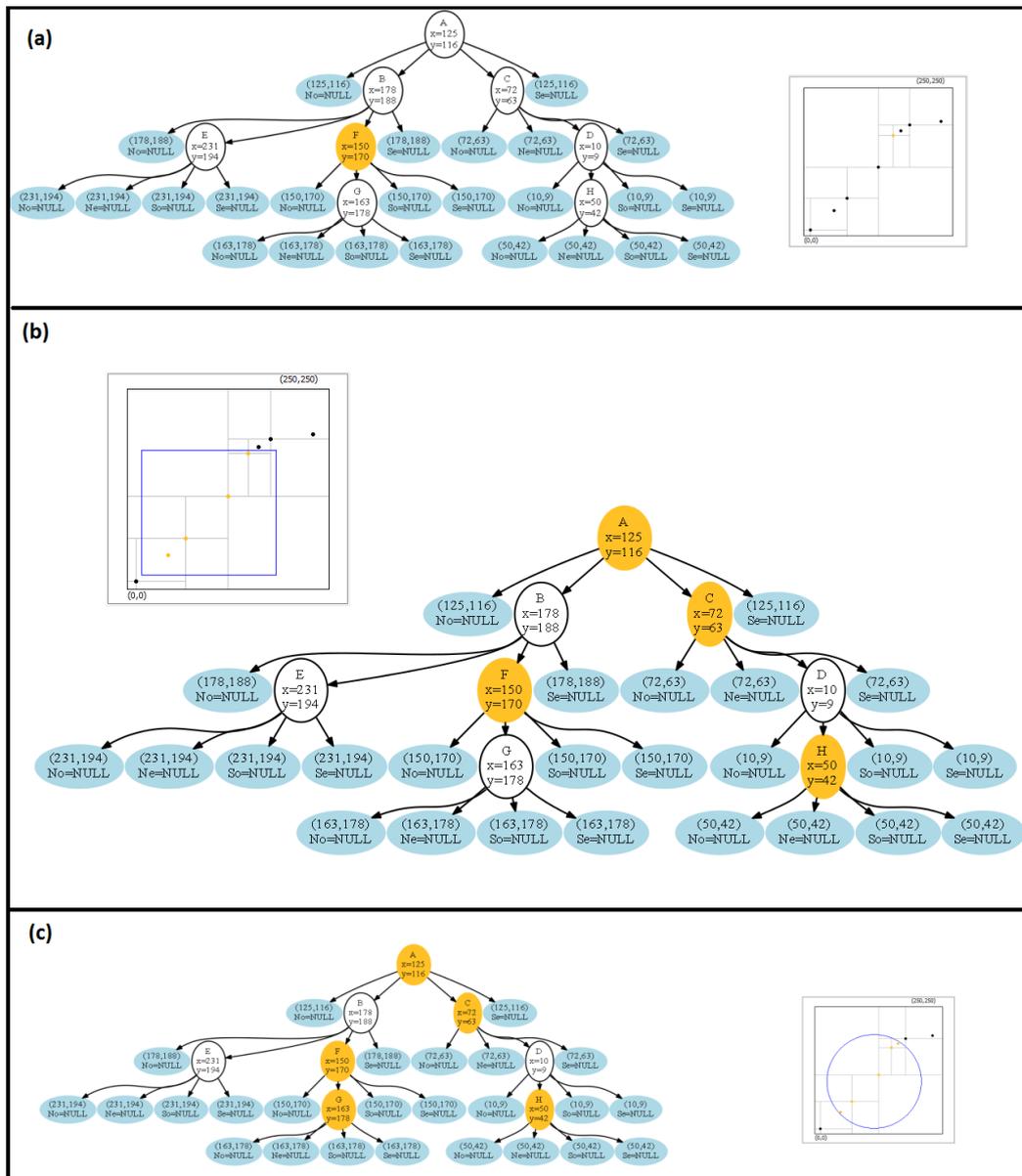


Figura 14. (a) Busca Simples. (b) Busca por Região. (c) Busca por Vizinhança.

## 6. Conclusões

O presente trabalho mostrou a definição, características e a manipulação de dados em Point Quadtrees. Foi feita uma descrição detalhada de cada conceito para as operações de inserção, exclusão e busca em Point Quadtrees.

Paralelamente, foi apresentado o software Visualizador de Point Quadtree desenvolvido, que propicia uma experiência visual no contato com essa estrutura de dados, sendo potencial facilitador no aprendizado da mesma. Com o software é possível a criação de uma PQ em um espaço 250 x 250. Também é possível a criação de uma árvore com dados aleatórios. Com isso, torna-se fácil o estudo de árvores dos mais diversos tamanhos e, assim, a busca, inclusão e exclusão podem ser experimentadas de diversas maneiras, de acordo com o interesse do usuário.

## Referências

- Bentley, J. L. (1975) “Multidimensional Binary Search Trees Used for Associative Searching”, In Communications of the ACM 18, Volume 9, September, p. 509-517.
- Bentley, J. L. and Finkel, R. A. (1974) “Quad Trees – A Data Structure for Retrieval on Composite Keys”, In: Acta Informatica 4, p. 1–9.
- Blanchette, J., Summerfield, M. (2008) “C++ GUI Programming with Qt 4”, Prentice Hall; 2 edition.
- Ellson, J., Gansner, E., Koutsofios, L., North, S. and Woodhull, G. (2003) “Graphviz and Dynagraph — Static and Dynamic Graph Drawing Tools”, <http://www.graphviz.org/Documentation/EGKNW03.pdf>, acesso em 20 set 2013.
- Ellson, J., Gansner, E., Koutsofios, L., North, S. and Woodhull, G. (1988), “Graphviz— Open Source Graph Drawing Tools”. In. P. Mutzel, M. Jünger, Michael, S. Leipert, Graph Drawing, Isbn: 978-3-540-43309-5.
- Gansner, E. R., Koutsofios, E. and North, S. (2009) “Drawing Graphs with Dot”, <http://www.graphviz.org/pdf/dotguide.pdf>, acesso em 20 set 2013.
- Gardner, H., Chen, J.-Q. and Moran S. (2010) “Inteligências Múltiplas ao Redor do Mundo”, Artmed Editora; 1ª edição .
- KDE Free Qt Foundation (2013), <http://www.kde.org/community/whatiskde/kdefreeqtfoundation.php>, acesso em 20 set 2013.
- Lima, R. D., Mota, G. L. A. and Pinto, P. E. D. (2012) “Interface Luana: uma Aplicação Gráfica para o Ensino da Árvore Binária Kd-Tree” In SIBGRAPI - Conference on Graphics, Patterns and Images, 2012, Ouro Preto. Anais do Sibgrapi, 2012.
- Miranda, A. C. O. and Martha, L. F. (2002) “Mesh Ggeneration on High-Curvature Surfaces Based on Background Quadtree Sstructure” In 11th International Meshing Roundtable , Springer-Verlag, p.333-342.
- Samet, H. (1990) “The Design and Analysis of Spatial Data Structures”, Addison\_Wesley.
- Samet, H. (2006) “Foundations of Multidimensional and Metric Data Structures”, Morgan Kaufmann.
- Samet, H., Rosenfeld, F. A., Shaeffer, C. A. and Webber, R. E. (1984) “A Geographic Information System Using Quadtrees” In Pattern Recognition Volume 17, Issue 6, p. 647–656.
- Strobach, P. (1991) “Quadtree-structured Recursive Plane Decomposition Coding of Images”, In IEEE Trans. Signal Processing, vol. 39, p. 1380-1397.