



UNIVERSIDADE ESTADUAL DO MARANHÃO
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
CENTRO DE CIÊNCIAS TECNOLÓGICAS

**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO
CURSO DE MESTRADO PROFISSIONAL EM ENGENHARIA DE COMPUTAÇÃO
E SISTEMAS**

JEFFERSON AMARAL DA SILVA

Dissertação de Mestrado

**COP MODEL – REFINAMENTO DE DIRECIONALIDADE A CONSULTAS
ESPACIAIS EM SGBDEs**

São Luis
2014

JEFFERSON AMARAL DA SILVA

**COP MODEL – REFINAMENTO DE DIRECIONALIDADE A CONSULTAS
ESPACIAIS EM SGBDEs**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Computação e Sistemas da Universidade Estadual do Maranhão, como parte das exigências para a obtenção do título de mestre em Engenharia de Computação e Sistemas.

Orientadora: Prof^a. Dra. Karla Donato Fook.

SÃO LUIS - MA

2014

Silva, Jefferson Amaral da.

COP MODEL – Refinamento de direcionalidade a consultas espaciais em SGBDES / Jefferson Amaral da Silva.– São Luís, 2014.

76f

Dissertação (Mestrado) – Curso de Engenharia da Computação e Sistemas, Universidade Estadual do Maranhão, 2014.

Orientador: Profa. Dra. Karla Donato Fook

1.Consultas espaciais. 2.Sistemas Gerenciadores de Bancos de Dados Espaciais. 3.COP MODEL. I.Título

CDU: 004.65

JEFFERSON AMARAL DA SILVA

**COP MODEL – REFINAMENTO DE DIRECIONALIDADE A CONSULTAS
ESPACIAIS EM SGBDEs**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Computação e Sistemas da Universidade Estadual do Maranhão, como parte das exigências para a obtenção do título de mestre em Engenharia de Computação e Sistemas.

APROVADA EM: 10 de junho de 2014.



Prof^a. Dra. Karla Donato Fook

(Orientadora)

(IFMA)



Prof. Dr. João Carlos Pinheiro

(IFMA)



Prof. Dr. Sérgio Souza Costa

(UFMA)

A quem sempre esperou de mim
mais do que eu mesmo esperei.

AGRADECIMENTOS

Agradeço majoritariamente a Deus, que me permitiu ingressar nessa caminhada e enfim permite que eu chegue à sua conclusão, sempre me guiando e abençoando. Sem sua permissão, nada teria sido possível, não importa quanto fosse buscado.

Aos meus familiares e amigos, que sempre tiveram certeza de que eu conseguiria chegar até aqui e têm crença e confiança fenomenal em mim, especialmente meu irmão Jackson, que além de tudo ainda pôde me ajudar nessa caminhada, seja com conhecimento técnico, seja com a experiência adquirida por ele próprio na busca de seu merecido título de Mestre.

A todas as pessoas que pude conhecer por meio do Programa de Pós-Graduação em Engenharia de Computação e Sistemas da Universidade Estadual do Maranhão em parceria com o Instituto Federal de Educação, Ciência e Tecnologia do Maranhão, tanto colegas discentes quanto docentes e administrativos.

À minha orientadora Prof^a. Dr^a. Karla Donato Fook, que aceitou o desafio de me instruir e orientar, tarefas estas que imagino não serem das mais simples.

A todos que direta ou indiretamente colaboraram com essa minha conquista.

“O laço essencial que nos une é que todos habitamos este pequeno planeta.”

John Kennedy

RESUMO

A identificação de Relacionamentos Direcionais entre Objetos armazenados em Sistemas Gerenciadores de Bancos de Dados Espaciais (SGBDEs) pode configurar uma importante fonte adicional de informações, especialmente quando estes são utilizados em conjunto com outros tipos de informações, como as provenientes de Relacionamentos Topológicos e Métricos. Tal combinação pode oferecer suporte a tomadas de decisão em situações dependentes do posicionamento relativo entre Objetos Espaciais nos quais a direção relativa, e consequentemente a ordem, destes tem importância estratégica, como na construção de certas edificações, abertura de vias de acesso, instalação de torres de transmissão, etc. No entanto, o conceito de Relacionamentos Direcionais não é unificado, o que gera diferentes abordagens acerca da definição das direções, cada qual com suas características positivas e negativas. Assim sendo, este trabalho propõe um modelo híbrido, nomeado *Cone-Over-Projection Directional Model* (COP Model), que contempla características dos modelos baseados em Cone e em Projeção. Adicionalmente, o trabalho apresenta a implementação do modelo proposto a partir do uso de Algoritmos Geométricos, de forma agregada a um SGBDE, visto a possibilidade de reutilização dos recursos do mesmo, compatibilidade com bases de dados previamente existentes e a notória carência de implementações seguindo esta abordagem agregada.

Palavras chave: Relacionamentos Direcionais, Sistemas Gerenciadores de Bancos de Dados Espaciais, Sistemas de Informação Geográfica, Algoritmos Geométricos, COP Model.

ABSTRACT

The identification of Directional Relationships between objects stored in Spatial Database Management Systems (SDMSs) can set an important additional source of information, especially when they are used in conjunction with other types of information, such as from Metric and Topological Relationships. Such a combination can support decision-making in situations dependent to the relative positioning between Geographical Objects in which the relative direction, and consequently the order of these is of strategic importance, such as the construction of certain buildings, opening of access roads, installation of transmission towers, etc. However, the Directional Relationships concept is not unified, which generates different approaches to the definition of directions, each with its positive and negative characteristics. Therefore, this work proposes the *Cone-Over-Projection Directional Model* (COP Model) as a hybrid model that includes characteristics of the models based on Cone and Projection. Additionally, the work presents the implementation of the proposed model from the use of Geometric Algorithms, aggregated to a SDMS, since the possibility of reuse of its features, compatibility with previously existing databases and notorious lack of implementations following this aggregated approach.

Keywords: Directional Relationships, Spatial Database Management Systems, Geographic Information Systems, Geometric Algorithms, COP Model.

LISTA DE FIGURAS

Figura 1 - Arquitetura Dual em SIGs	19
Figura 2 - Arquitetura Integrada em um SGBD Relacional em SIGs	19
Figura 3 - Arquitetura Integrada via Extensão Espacial em SIGs	20
Figura 4 - Hierarquia de tipos da SF-SQL	21
Figura 5 - Requisitos da Arquitetura PGXS	24
Figura 6 - Representação do REM de uma geometria	26
Figura 7 - Representação visual das operações do Polygon Overlay	30
Figura 8 - Matriz de 4-Interseções para relações entre duas regiões	32
Figura 9 - Matriz de 9-Interseções para relações entre duas regiões	33
Figura 10 - Modelo Baseado em Cone	37
Figura 11 - Modelo Baseado em Projeção	38
Figura 12 - Crescimento desigual entre as direções cardeais e colaterais	39
Figura 13 - Gráfico de crescimento das direções cardeais e colaterais	39
Figura 14 - Porcentagens do espaço ocupadas por direções cardeais e colaterais ..	40
Figura 15 - Utilização do Modelo REM de Egenhofer	43
Figura 16 - Fluxo do modelo de Zhu	44
Figura 17 - Triângulo utilizado para criar as partições cônicas do espaço	47
Figura 18 - Cálculo das áreas das partições cônicas	47
Figura 19 - Regiões adicionadas pelo COP Model	49
Figura 20 - Diagrama de Fluxo de Dados para as funções do COP Model	50
Figura 21 - Ilustração da camada na qual as extensões são inseridas	52
Figura 22 - Generalização das funções implementadas no COP Model	53
Figura 23 - Vértices das partições nas direções paralelas ao eixo X	54
Figura 24 - Vértices das partições nas direções paralelas ao eixo Y	55
Figura 25 - Tela principal do COP-VW	58
Figura 26 - Janela de configuração de consulta <i>Testar Alvo</i>	59
Figura 27 - Código gerado pelo COP-VW para uma consulta <i>Testar Alvo</i>	59
Figura 28 - Saída gerada por uma consulta <i>Testar Alvo</i>	60
Figura 29 - Janela de configuração de consulta <i>Encontrar Alvo(s)</i>	61

Figura 30 - Código gerado pelo COP-VW para uma consulta <i>Encontrar Alvo(s)</i>	61
Figura 31 - Saída gerada por uma consulta <i>Encontrar Alvo(s)</i>	62
Figura 32 - Saída gerada por uma consulta com tabelas e geometrias diferentes....	63
Figura 33 - Classes de modelo e suas dependências usadas no COP-VW	64
Figura 34 - Gráfico de crescimento das direções no COP Model.....	66
Figura 35 - Porcentagens do espaço ocupadas pelas direções do COP Model.....	67
Figura 36 - Comparação entre as partições da Projeção e do COP Model.....	68
Figura 37 - Problema gerado pelo REM de certos Objetos	70

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
BLOB	<i>Binary Long Object</i>
COP-ME	<i>COP Directional Model Extension</i>
COP Model	<i>Cone-Over-Projection Directional Model</i>
COP-VW	<i>COP Directional Model Viewer</i>
DE-9IM	<i>Dimensional Extended 9-Intersection Matrix</i>
EAQC	<i>Extraction Algorithm of Query Characteristics</i>
OGC	<i>Open Geoscience Consortium</i>
OO	Orientação a Objetos
Java 2D	<i>Java 2D Graphics API</i>
JDBC	<i>Java Database Connectivity</i>
PGXN	<i>PostgreSQL Extension Network</i>
PL/pgSQL	<i>PostgreSQL Procedural Language</i>
PL/SQL	<i>Procedural Language Extensions to SQL</i>
REM	Retângulo Envolvente Mínimo
SF-SQL	<i>Simple Features - SQL</i>
SGBD	Sistema Gerenciador de Bancos de Dados
SGBD-R	Sistema Gerenciador de Bancos de Dados Relacional
SGBD-OR	Sistema Gerenciador de Bancos de Dados Objeto-Relacional
SGBD-OO	Sistema Gerenciador de Bancos de Dados Orientado a Objetos
SGBDE	Sistema Gerenciador de Bancos de Dados Espaciais
SIG	Sistema de Informação Geográfica
SQL	<i>Structured Query Language</i>
SWING	<i>Java Sun Windowing</i>
T-SQL	<i>Transact-SQL</i>

SUMÁRIO

1	INTRODUÇÃO	14
2	SISTEMAS GERENCIADORES DE BANCOS DE DADOS ESPACIAIS	17
2.1	SF-SQL	20
2.2	IMPLEMENTAÇÃO EM SGBDs	22
3	ALGORITMOS GEOMÉTRICOS E GEOMETRIA COMPUTACIONAL.....	25
4	RELACIONAMENTOS ENTRE OBJETOS GEOGRÁFICOS	31
4.1	RELACIONAMENTOS TOPOLÓGICOS	31
4.2	RELACIONAMENTOS MÉTRICOS	34
4.3	RELACIONAMENTOS DIRECIONAIS	35
4.3.1	Modelos para Definição de Relacionamentos Direcionais	36
5	TRABALHOS RELACIONADOS	42
6	CONE-OVER-PROJECTION DIRECTIONAL MODEL	46
6.1	ESPECIFICAÇÃO	48
6.2	IMPLEMENTAÇÃO	51
6.2.1	Verificação da Implementação	57
6.3	RESULTADOS	65
6.3.1	Discussão.....	66
7	CONCLUSÃO	69
7.1	TRABALHOS FUTUROS.....	70
	REFERÊNCIAS.....	72

1 INTRODUÇÃO

Relacionamentos direcionais estão fortemente ligados a consultas espaciais e ao raciocínio espacial de uma forma geral (TANG *et al.*, 2008), e são de natural percepção pelo ser humano. Relações direcionais dizem respeito adicionalmente à ordem em que as entidades geográficas estão dispostas, diferentemente das relações topológicas, as quais esta disposição não é relevante.

Já na década de 90, Theodoridis *et al.* (1997) apontavam a importância dos relacionamentos de direcionalidade como sendo referente ao conceito de ordem no espaço, o que não era abordado pelos relacionamentos topológicos, sendo que, estes relacionamentos poderiam prover respostas para consultas do tipo “encontrar todos os países ao nordeste de um determinado país”, o que não seria possível utilizando somente relações topológicas. Ainda nos anos 90, Egenhofer *et al.* (2000) descreviam a importância dos relacionamentos direcionais como sendo relacionamentos adicionais aos relacionamentos topológicos. Recentemente, Zhu *et al.* (2012) e Yan *et al.* (2013) retomam esta questão da direcionalidade utilizando recursos e conceitos não existentes ou não devidamente explorados na época em que Theodoridis e Egenhofer desenvolveram seus trabalhos.

De uma forma geral, os relacionamentos direcionais são de grande importância pelo fato de que estes podem gerar informação que não pode ser obtida por meios topológicos e/ou métricos. No entanto, a informação obtida por meio de relacionamentos direcionais se torna mais interessante quando cruzada com informações de natureza topológica e métrica, o que ressalta como vantajosa uma implementação incorporada ao SGBDE, a fim de se executar consultas espaciais que utilizem informações de todas estas naturezas de forma transparente.

Citam-se duas famílias de modelos que objetivam definir os Relacionamentos Direcionais: Modelos Baseados em Cone (PEUQUET e ZHAN, 1987) e Modelos Baseados em Projeção (FRANK, 1992). Quanto aos modelos baseados em Cone e em Projeção, Tang *et al.* (2008) apontam algumas deficiências destes modelos, como as limitações de uso sobre determinados conjuntos de geometrias e o resultado errôneo gerado no processamento de relacionamentos em alguns casos, o

que limita, ou impossibilita, sua utilização em determinadas situações. Corrigir ou amenizar estas deficiências pode permitir a concepção de um modelo simples, abrangente e compatível com diversas geometrias.

Theodoridis *et al.* (1997) também destacam a importância da incorporação do conceito de direcionalidade de forma agregada ao SGBDE especialmente para uso em conjunto com as implementações existentes de relacionamentos topológicos e, adicionalmente, visando uma melhor performance durante a execução. Ainda nesse contexto, vale ressaltar que a implementação na camada do SGBDE, seja a partir de alteração do código-fonte ou por criação de extensões, torna o seu uso transparente para aplicações terceiras, além de torná-la reutilizável.

Um dos problemas em se tratar relações direcionais é que, ao contrário do caso das relações topológicas, onde parece existir um conjunto de relações amplamente aceito (EGENHOFER *et al.*, 1991), não há uma definição unificada de relações de direção (THEODORIDIS *et al.*, 1997). Como consequência desta falta de unificação, existem diversos modelos que definem os relacionamentos direcionais, cada qual com seus conceitos e características.

Dentro da diversidade de modelos para definição de relacionamentos direcionais, destacam-se, pela sua abrangência e simplicidade, os Modelos Baseados em Cone (PEUQUET e ZHAN, 1987) e os Modelos Baseados em Projeção (FRANK, 1992). Estes modelos podem ser implementados com relativa facilidade, porém, são falhos em determinados pontos, como no tratamento de geometrias complexas (TANG *et al.* 2008) e na distribuição do espaço entre as direções. Nesse contexto, abre-se a possibilidade de unir características destes dois modelos a fim de obter uma abordagem mais eficiente no sentido de definir as regiões pertencentes a cada direção.

A adição deste conceito de direcionalidade em consultas espaciais configura um problema a parte, visto que esta adição, corriqueiramente, é feita de maneira externa ao Sistema Gerenciador de Bancos de Dados Espaciais (SGBDE), em ferramentas de terceiros, o que, dentre outras complicações, limita a possibilidade de reutilização da implementação por outras aplicações e de integração com funcionalidades já existentes no SGBDE.

Desta forma, este trabalho apresenta um modelo híbrido para definição de relacionamentos direcionais, unindo os modelos baseados em Cone e em Projeção, nomeado *Cone-Over-Projection Directional Model (COP Model)* (DA SILVA, 2013), além de efetuar a implementação deste incorporada a um SGBDE, permitindo consultas espaciais SQL/SF-SQL que utilizem os recursos implementados e os nativos do SGBDE juntos de forma transparente.

Secundariamente, este trabalho necessitou cumprir com objetivos inerentemente ligados às concepção, implementação e verificação do modelo apresentado, tais como o estudo das características e funcionamento dos modelos baseados em Cone e Projeção, a identificação das deficiências e vantagens de cada modelo, a proposição de um modelo híbrido baseado nos modelos estudados, a definição matemática do modelo proposto, a criação de uma implementação genérica em forma de pseudo-código do modelo proposto e a implementação do modelo como extensão de um SGBDE.

Nesse contexto, os demais capítulos deste trabalho estão organizados da seguinte forma: o Capítulo 2 explana acerca da teoria e aplicação de Sistemas Gerenciadores de Bancos de Dados Espaciais, focando no modelo de implementação de funcionalidades do SGBD *PostgreSQL* juntamente com a extensão espacial *PostGIS*; o Capítulo 3 apresenta uma introdução aos Algoritmos Geométricos e à Teoria da Geometria Computacional; o Capítulo 4 trata dos Relacionamentos entre Objetos Espaciais, especialmente os de natureza Topológica, Métrica e Direcional; o Capítulo 5 apresenta trabalhos relacionados; o Capítulo 6 trata da concepção, definição matemática, implementação e validação do COP Model, além da apresentação e discussão dos resultados; e finalmente, o Capítulo 7 apresenta as conclusões deste trabalho e o horizonte de trabalhos futuros.

2 SISTEMAS GERENCIADORES DE BANCOS DE DADOS ESPACIAIS

Um Sistema Gerenciador de Bancos de Dados oferece serviços de armazenamento, consulta e atualização de bancos de dados. Os SGBDs devem atender a requisitos básicos como facilidade de uso e manutenção, confiabilidade, segurança, otimização de desempenho, etc. O mercado para SGBDs concentra-se basicamente em três tecnologias: SGBDs Relacionais (SGBD-R), SGBDs Objeto-Relacionais (SGBD-OR) e SGBDs Orientados-a-Objeto (SGBD-OO) (FERREIRA *et al.*, 2005).

Os SGBD-R seguem o modelo relacional de dados, onde um banco de dados é organizado como uma coleção de relações (tabelas), cada qual com atributos de um tipo específico. Atualmente, os tipos incluem números inteiros, de ponto flutuante, cadeias de caracteres, datas e campos binários longos (BLOBs), para os quais se encontram disponíveis uma variedade de operações (exceto para o tipo BLOB), como operações aritméticas, conversões, manipulação textual e operações com datas (FERREIRA *et al.*, 2005).

Os SGBD-R foram concebidos para atender às necessidades de aplicações manipulando grandes volumes de dados convencionais, porém, estes não oferecem recursos para atender as necessidades de aplicações não convencionais (como SIGs), já que a mera simulação de tipos de dados não convencionais em um SGBD-R pode ter efeitos negativos, como queda de desempenho, dificuldade de codificação e posterior manutenção da aplicação (STONEBRAKER, 1996).

Os SGBD-OR estendem o modelo relacional, dentre outras características, com um sistema de tipos de dados rico e extensível, oferecendo operadores que podem ser utilizados na linguagem de consulta. Possibilitam adicionalmente a extensão dos mecanismos de indexação sobre os novos tipos. Estas novas características adicionadas reduzem os problemas ocorridos na simulação de tipos de dados pelos SGBD-R, tornando os SGBD-OR uma solução atrativa para aplicações não convencionais (FERREIRA *et al.*, 2005).

Um SGBD-OO é um SGBD em que cada informação é armazenada na forma de objetos, ou seja, utiliza o paradigma da Orientação a Objetos (OO), o qual

permeia as linguagens mais modernas. Existem dois fatores principais que levam a adoção da tecnologia SGBD-OO. A primeira, é que em um banco de dados relacional se torna difícil de manipular com dados complexos, o segundo é que os dados são geralmente manipulados pela aplicação escrita usando linguagens de programação orientada a objetos, como Java, Python, C++, C#, etc., e o código precisa ser traduzido entre a representação do dado e as tuplas da tabela relacional, o que além de ser uma operação tediosa de ser escrita, consome tempo (ELMASRI, 2005).

Porém, uma limitação dos SGBDs-OO parece ser a perda da interoperabilidade com um grande número de ferramentas/características que são tidas como certas no mundo SQL, incluindo a indústria de padrões de conectividade, ferramentas de relatório, ferramentas de OLAP e *backup*, e padrões de recuperação. Adicionalmente, SGBDs-OO perdem o fundamento formal matemático, ao contrário do modelo relacional, e isto às vezes conduz a fraqueza na sustentação da consulta. Entretanto esta objeção é descartada pelo fato que alguns SGBDs-OO suportam totalmente o SQL, mas o uso eficaz pode requerer acordos para manter ambos os paradigmas sincronizados (FERREIRA *et al.*, 2005).

Conforme apontado por Stonebraker (1996), SGBDs puramente relacionais não são adequados para o armazenamento de dados em aplicações não convencionais, e, segundo Melo e Guerra (2005), Sistemas de Informações Geográficas (SIGs) diferem dos sistemas de informação convencionais, dentre outras coisas, pela forma como os dados alfanuméricos e espaciais são armazenados no SGBD, sendo que esta pode seguir três arquiteturas diferentes:

- *Dual*: Os dados cadastrais são armazenados em um SGBD relacional de forma convencional e os dados espaciais são armazenados em arquivos isolados em formatos específicos, conforme a Figura 1;

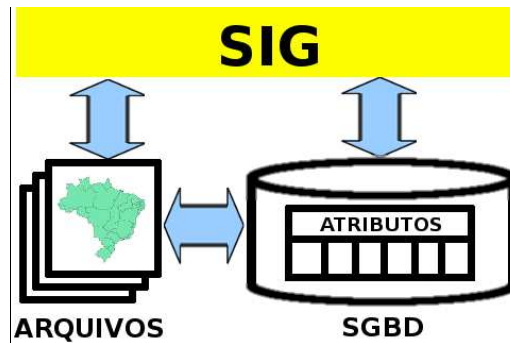


Figura 1 - Arquitetura Dual em SIGs

- *Integrada em um SGBD Relacional:* Ambos os tipos de dados são armazenados em SGBDs relacionais convencionais, sendo que os são tratados como dados do tipo *Binary Long Object* (BLOB) nas mesmas tabelas nas quais se armazena os dados cadastrais, conforme a Figura 2, sendo que estes BLOBs não possuem qualquer semântica agregada; e,

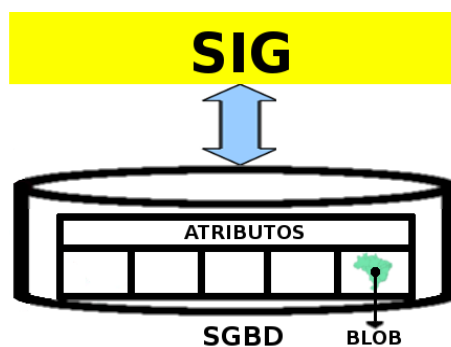


Figura 2 - Arquitetura Integrada em um SGBD Relacional em SIGs

- *Integrada Via Extensão Espacial:* Os dados são armazenados em SGBDs com extensões espaciais, ou simplesmente SGBDEs, que adicionam funcionalidades e semântica não presentes em SGBDs convencionais, tais como suporte a tipos de dados espaciais (ponto, linha, polígono, etc.), funções e operadores espaciais e mecanismos de indexação específicos para o contexto espacial. A extensão espacial se comunica diretamente com o SGBD, e os dados espaciais passam a ser armazenados em forma de Objetos Espaciais, conforma a Figura 3.



Figura 3 - Arquitetura Integrada via Extensão Espacial em SIGs

Os SGBDEs utilizados pelos SIGs possuem dados usualmente agrupados em duas componentes: a componente espacial (geográfica), que armazena as informações complexas referentes ao espaço, e a componente convencional, que armazena dados cadastrais comuns tais quais os encontrados em SGBDs convencionais. Assim, são armazenadas as geometrias e as descrições dos elementos que representam as características do mundo real, gerenciadas e processadas em um único ambiente computacional.

Existem diversos SGBDs convencionais que suportam dados espaciais a partir da utilização extensões específicas, entre eles se pode destacar o *PostgreSQL* com a extensão *PostGIS*, o *Oracle* com a extensão *Spatial*, etc. (FERREIRA, 2006).

2.1 SF-SQL

A *Simple Features - SQL* (SF-SQL), proposta pelo *Open Geoscience Consortium* (OGC), especifica um conjunto de tipos de geometrias vetoriais, operações topológicas e operações métricas. A proposta especifica ainda um esquema de tabelas para metadados das informações espaciais além de introduzir o conceito de "tabela com feições" para representação dos dados espaciais. Nesta tabela, os atributos não espaciais são mapeados para colunas de tipos disponíveis na SQL-92, e os atributos espaciais para colunas cujo tipo de dados é baseado no conceito de "tipos de dados geométricos adicionais para SQL" (FERREIRA *et al.*, 2005).

A representação dos atributos espaciais pode seguir dois modelos, chamados SQL-92 e “SQL-92 com Tipos Geométricos”. O primeiro modelo utiliza uma tabela para representar os atributos espaciais. O segundo utiliza tipos abstratos de dados específicos para geometrias, estendendo os tipos da SQL (FERREIRA *et al.*, 2005).

Os tipos de geometrias da SF-SQL são organizados hierarquicamente segundo a especificação do OGC. A Figura 4 ilustra a hierarquia de tipos da SF-SQL, sendo que este diagrama é o mesmo tanto para o modelo da SQL-92 quanto para o modelo da SQL-92 com Tipos Geométricos.

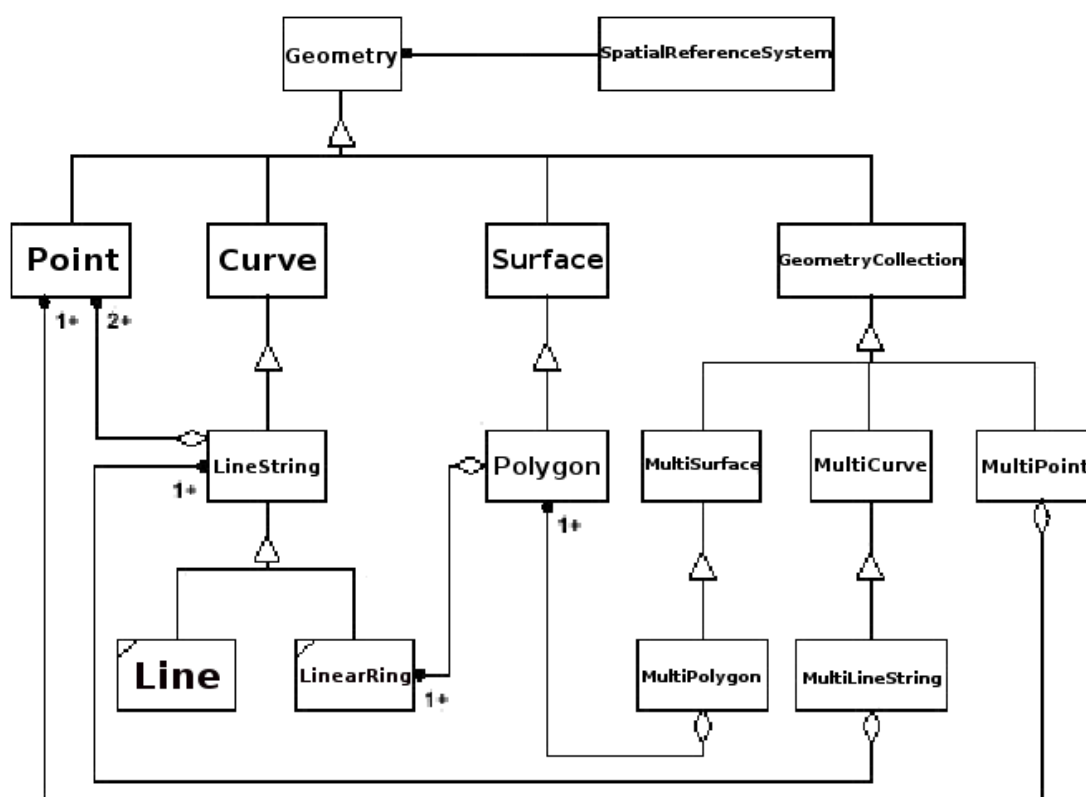


Figura 4 - Hierarquia de tipos da SF-SQL

Fonte: Ferreira et al. (2005)

Alguns tipos são abstratos como: *Curve*, *Surface*, *MultiSurface* e *MultiCurve*. Um tipo especial é a *GeometryCollection*, que pode ser composta por mais de um tipo de geometria (tipo heterogêneo). Os outros são tipos básicos, como *Point*, *LineString* e *Polygon*, que podem formar tipos de coleções homogêneas como *MultiPoint*, *MultiLineString* e *MultiPolygon*, respectivamente. Cada um destes tipos possui uma série de atributos, métodos e definições que são apresentadas na especificação (FERREIRA *et al.*, 2005).

2.2 IMPLEMENTAÇÃO EM SGBDs

A implementação de funcionalidades em SGBDs pode ser feita de diferentes maneiras, uma delas é lançando mão do uso de *Stored Procedures*. Segundo Santos (2008), *Stored Procedures*, ou Procedimentos Armazenados, se traduzem em bibliotecas de comandos SQL para armazenamento e utilização junto ao banco de dados, com a possibilidade de execução de tarefas repetitivas e aceitação de passagem de parâmetros de entrada para que a tarefa possa ser efetuada de acordo com possíveis necessidades individuais.

Para Santos (2008), com *Stored Procedures* pode-se reduzir o tráfego na rede, melhorar o desempenho de um banco de dados, criar tarefas agendadas, diminuir riscos, criar rotinas de processamento, etc., além de permitir o seu uso por aplicações terceiras implementadas de diferentes maneiras.

Functions, ou Funções, são muito semelhantes às *Stored Procedures* no contexto de como são construídas. As *Functions* carregam as mesmas características das *Stored Procedures*, com a diferença de que obrigatoriamente devem retornar um valor (PRADO, 2012). As *Functions* são muito utilizadas para computar valores, promover reusabilidade e facilidade de manutenção. É muito comum se utilizar *Functions* em instruções SQL para efetuar conversões de dados, formatar datas, contar o total de linhas, etc. (SANTOS, 2012).

Tanto as *Stored Procedures* quanto as *Functions* devem ser escritas na linguagem proprietária do SGBD na qual estão sendo implementadas e a implementação deve seguir as peculiaridades do SGBD alvo. Algumas linguagens específicas para determinados SGBDs são: *Procedural Language Extensions to SQL* (PL/SQL) para o SGBD Oracle, *PostgreSQL Procedural Language* (PL/pgSQL), para o *PostgreSQL*, *Transact-SQL* (T-SQL) no caso do *Microsoft SQL Server*, etc.

Outra maneira de implementar novas funcionalidades para um SGBD é por meio da manipulação do seu código-fonte, no entanto, para isso, é necessário se ter acesso a este código-fonte. Felizmente, existem bancos de dados *opensource* bem consolidados no mercado, como é o caso do *PostgreSQL* que juntamente com sua

extensão espacial *PostGIS* configura uma robusta opção para implementações de SIGs corporativos (UCHOA *et al.* 2005).

O *PostgreSQL* possui uma arquitetura especificamente concebida para a criação de novas funcionalidades por meio de extensões, nomeada PGXS. Usando o PGXS, as extensões implementadas podem ser compiladas, instaladas, registradas e usadas em qualquer instalação do *PostgreSQL* 9.1+, mesmo que a instalação seja preexistente, em outras palavras, não há necessidade de reinstalação do SGBD ou instalação de uma versão modificada, mas sim somente da instalação e registro das extensões criadas. Tudo isto torna as extensões desenvolvidas redistribuíveis e reutilizáveis. Para que se possa usufruir da infraestrutura do PGXS, é necessário basicamente:

- Uma instalação válida do *PostgreSQL* 9.1+;
- Extensão escrita em Linguagem C;
- *Makefile* com entradas específicas do PGXS (interpretado pelo utilitário *make*);
- Arquivo de controle com metadados acerca da extensão criada (interpretado pelo *engine* PGXS);
- Compilador C;
- Arquivo SQL com as definições das funcionalidades que a extensão adiciona ao SGBD.

O *makefile* permite automatizar a compilação, o *link* com as bibliotecas do *PostgreSQL* e a instalação da extensão criada no sistema cliente. Para tal, há a necessidade da adição de entradas específicas do PGXS no *makefile*, que deve ser escrito usando a marcação *Shell Script*.

O Arquivo de Controle (.control) é um arquivo de texto puro que deve conter informações acerca da extensão criada. Estas informações são tanto para uso interno do PGXS quanto para visualização em ferramentas *front-end*.

O Arquivo SQL com as definições das funcionalidades adicionadas pela extensão deve ser escrito usando a linguagem PL/pgSQL. A partir deste arquivo, o SGBD identifica como as novas funcionalidades devem ser usadas e invocadas.

Este arquivo deve ser consonante com a seguinte regra de nomenclatura: O nome da extensão, sucedido de dois traços e o número da versão da extensão.

Por fim, o arquivo contendo a implementação da extensão deve ser escrito em Linguagem C e pode carregar bibliotecas do ISO C ou de terceiros. No caso de aplicações com fins espaciais, além das bibliotecas padrão, pode-se utilizar recursos e estruturas das bibliotecas provenientes do *PostgreSQL* e do *PostGIS*, essencialmente abstrações de geometrias e funções para recuperação de informações e matemáticas, conseguindo-se assim um alto grau de compatibilidade da extensão com o SGBDE. A Figura 5 mostra um resumo destes requisitos necessários para a utilização da arquitetura PGXS.

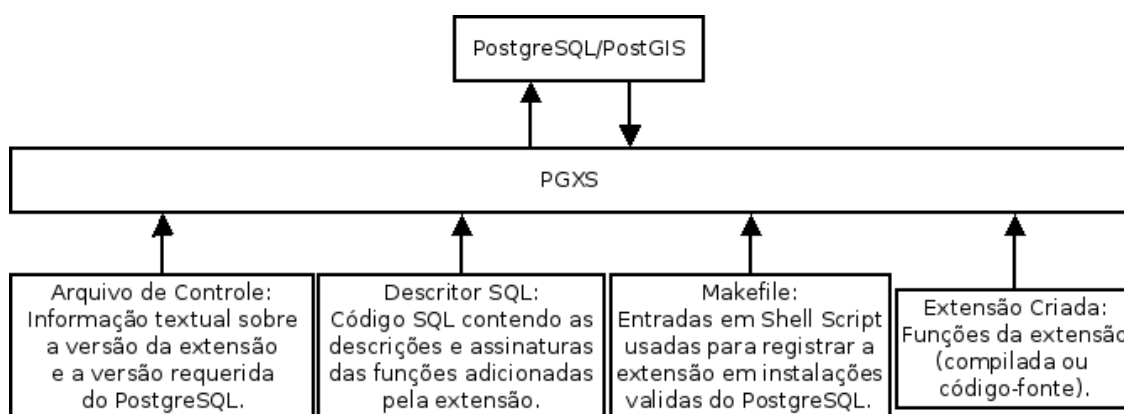


Figura 5 - Requisitos da Arquitetura PGXS

Adicionalmente, existe ainda a *PostgreSQL Extension Network* (PGXN) que é um sistema centralizado de distribuição central para extensões do *PostgreSQL*. É composto basicamente por: (1) Uma infraestrutura de *upload* de distribuições de extensões; (2) Um índice centralizado e API de metadados de distribuição; (3) Um site com *engine* de busca de extensões; e (4) Um cliente de linha de comando para baixar, testar e instalar extensões (POSTGRESQL EXPERTS INC, 2012).

Por todas as características e recursos elencados, o conjunto *PostgreSQL/PostGIS* foi selecionado para a implementação do modelo apresentado neste trabalho.

3 ALGORITMOS GEOMÉTRICOS E GEOMETRIA COMPUTACIONAL

Davis Jr. e Queiroz (2005) observam que SGDBEs comumente possuem bibliotecas de funções, em especial para realizar as operações sobre representações vetoriais (pontos, linhas e polígonos), que estão subjacentes a situações típicas, tais como:

- Seleção por apontamento, em que um usuário seleciona um determinado objeto através da interface gráfica;
- Determinação do relacionamento espacial entre dois objetos, tanto para consultas quanto para o estabelecimento de restrições de integridade espaciais no banco de dados;
- Criação de mapas de distância (*buffer zones*) e solução de problemas de proximidade;
- Sobreposição e aritmética de polígonos para operações de análise espacial.

Estes tipos de operações são objeto de estudo de uma área da Ciência da Computação conhecida como Geometria Computacional (PREPARATA e SHAMOS, 1985), que procura desenvolver e analisar algoritmos e estruturas de dados para resolver problemas geométricos diversos.

Dentro do contexto dos SGBDEs, pode-se elencar, segundo Davis Jr. (1997), uma série de conceitos referentes a estruturas de dados fundamentais utilizadas para representar o espaço, estas consideram apenas duas dimensões. A conceituação dada por Davis Jr. é mostrada a seguir:

- *Ponto*: Um ponto P é um par ordenado (x, y) de coordenadas espaciais;
- *Reta e Segmento de Reta*: Sejam P_1 e P_2 dois pontos distintos no plano, a combinação linear $\alpha P_1 + (1 - \alpha)P_2$, onde α é qualquer número real, é uma reta no plano. Quando $0 \leq \alpha \leq 1$ se tem um segmento de reta no plano que tem P_1 e P_2 como pontos extremos;
- *Linha Poligonal*: Sejam $v_0, v_1, \dots, v_{n-1}$ n pontos no plano. Sejam $S_0 = \overline{v_0 v_1}$, $S_1 = \overline{v_1 v_2}$, $\dots$, $S_{n-2} = \overline{v_{n-2} v_{n-1}}$ uma sequência de $n - 1$ segmentos conectando estes pontos, estes segmentos formam uma poligonal L se, e somente se, (1) a interseção de segmentos consecutivos é apenas o ponto

extremo compartilhado por eles, (2) segmentos não consecutivos não se interceptam, e (3) $v_0 \neq v_{n-1}$, ou seja, a poligonal não é fechada; e,

- **Polígono:** Um polígono é a região do plano limitada por uma linha poligonal fechada. O polígono divide o plano em duas regiões: o interior, que convencionalmente inclui a fronteira (a poligonal fechada) e o exterior.

Estas entidades geométricas básicas podem ser definidas em uma linguagem de programação qualquer usando tipos abstratos de dados usando os recursos disponíveis. Essa definição inclui representações para retângulos e para segmentos, que são bastante úteis nos testes preliminares de alguns algoritmos geométricos (DAVIS JR. e QUEIROZ, 2005). Não há a necessidade explícita de se definir um tipo abstrato específico para polígonos, uma vez que estes correspondem a linhas poligonais nas quais o primeiro e o último vértices coincidem.

A partir do armazenamento de dados no formato destas estruturas básicas em um SGBDE, ou em quaisquer ambientes com sintaxe e semântica devidamente definidas, pode-se realizar operações sobre tais dados através do uso de Algoritmos Geométricos (em meio computacional).

Outra estrutura importante para o desenvolvimento de Algoritmos Geométricos, apontada por Davis Jr. e Queiroz (2005), é o chamado Retângulo Envolvente Mínimo (REM). O REM é o menor retângulo com lados paralelos aos eixos coordenados que contém a geometria do objeto, conforme mostrado na Figura 6.

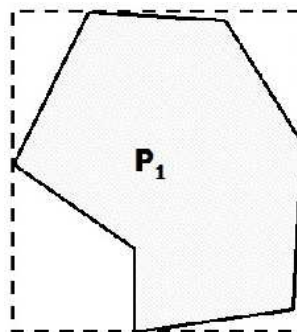


Figura 6 - Representação do REM de uma geometria

Antes de se executar um algoritmo qualquer que envolva dois polígonos complexos pode-se comparar seus REMs, que são representações mais simples.

Caso os REMs possuam algum tipo de relacionamento, é possível que os polígonos também o possuam, e, portanto, o algoritmo completo precisa ser executado; caso contrário, é certo que não existe relação alguma entre os objetos, descartando assim a necessidade de execução do algoritmo completo.

Problemas de cunho geométrico em SIGs tendem a ser onerosos em termos de processamento uma vez que estes perpassam por séries de cálculos matemáticos e tratam, potencialmente, de grandes conjuntos de dados. Assim, é importante ressaltar que:

Diversos problemas de geometria computacional utilizam resultados básicos de problemas mais simples em sua solução. Alguns destes resultados básicos vêm da análise geométrica do mais simples dos polígonos, e o único que sempre é plano: o triângulo (Davis Jr. e Queiroz, 2005, p. 46).

Assim sendo, os algoritmos que resolvem problemas mais simples são largamente utilizados em Geometria Computacional. Estes algoritmos incluem cálculos como:

- Área do Triângulo: É calculada como a metade da área de um paralelogramo (FIGUEIREDO e CARVALHO, 1991). O produto vetorial dos vetores $\vec{a}$ e $\vec{b}$ determina a área S do paralelogramo com os lados A e B e, portanto, a área do triângulo $\widehat{ABC}$;
- Interseção de Retângulos: Diversos Algoritmos Geométricos se beneficiam de estratégias de implementação tentam evitar o uso de procedimentos computacionalmente caros (DAVIS JR. e QUEIROZ, 2005). O uso do REM é uma dessas estratégias. Por exemplo, antes de executar o algoritmo complexo de determinação da interseção entre polígonos, comparam-se seus REMs. Caso os REMs possuam alguma interseção, é possível que os polígonos se interceptem, e, portanto, o algoritmo completo precisa ser executado; caso contrário, é certo que não existe interseção entre os objetos, e, portanto já se tem um conjunto vazio como solução;
- Interseção de Dois Segmentos de Reta: Para isto, basta verificar se os dois segmentos de reta pertencem a retas que não são paralelas nem

coincidentes; caso não sejam, é possível que os segmentos se cruzem, caracterizando a interseção; caso sejam, pode-se descartar essa possibilidade. No primeiro caso há a necessidade de se identificar o ponto de encontro das retas, se este pertencer a ambos os dados segmentos, então a interseção existe. Saalfeld (1987) discute uma forma de determinar o ponto de interseção entre dois segmentos baseada na representação paramétrica dos segmentos¹;

- Área de Polígonos: A área de um polígono pode ser calculada em tempo linear com relação ao número de vértices, usando um somatório simples, baseado na soma de áreas de triângulos formados entre cada par de vértices consecutivos e a origem do sistema de coordenadas (O'ROURKE, 1998);
- Centróide de um Polígono: O *centro de gravidade* ou *centro de massa*, também referenciado como *centróide* de um polígono pode ser obtido a partir da sua divisão em triângulos, calculando em seguida a média ponderada dos centros de gravidade dos triângulos usando suas áreas como peso. O centro de gravidade de cada triângulo é simplesmente a média das coordenadas de seus vértices (DAVIS JR. e QUEIROZ, 2005). Embora este processo seja relativamente simples, pressupõe-se a implementação de um algoritmo de triangulação de polígonos; e,
- Ponto em Polígono: Muito comum em SIGs, esta operação consiste em determinar se um ponto está no interior de um polígono. Um dos algoritmos para solução deste problema é o teste do número de cruzamentos entre os segmentos que formam a fronteira do polígono e uma semi-reta (chamada de raio), que parte do ponto testado em qualquer direção (HAINES, 1994) (TAYLOR, 1994). Se o número de cruzamentos for par, o ponto encontra-se fora do polígono; se for ímpar, encontra-se dentro.

Estes algoritmos básicos se resumem a operações de quantificação, operações booleanas e operações de cálculos de coordenadas. As operações que tratam diretamente relações entre estruturas espaciais que se relacionam são especialmente interessantes para este trabalho, pois quaisquer áreas referentes a partições do espaço podem ser tratadas como Polígonos. Ademais, a operação de

¹ A equação paramétrica para um segmento de coordenadas p_1 e p_2 é dada por: $p = p_1 + u(p_2 - p_1)$, onde, se $0 < u < 1$, define um ponto localizado entre p_1 e p_2 .

“Interseção de Retângulos” é sumamente importante para tratamentos de relações entre REMs de objetos.

Operações mais complexas, como o cálculo da interseção entre conjuntos de linhas poligonais, tendem a utilizar, na medida do possível, estes algoritmos mais simples. Existem casos em que estudiosos da Geometria Computacional ou membros da comunidade SIG conceberam algoritmos para solução de problemas complexos que são largamente aceitos. Este é o caso, por exemplo, do algoritmo de Varredura de Plano, ou *Plane Sweep*, que é baseado em um algoritmo apresentado por Shamos e Hoey (1976) e usado exatamente para determinar os pontos de interseção entre conjuntos de linhas poligonais. Adicionalmente, é comum que algoritmos mais complexos necessitem de estruturas de dados auxiliares para armazenamento de metadados durante a sua execução.

Um conjunto especial de operações apontado por Davis Jr. e Queiroz (2005, p. 69) como fundamental para a análise espacial é o *Polygon Overlay*:

Operações sobre polígonos são de fundamental importância em SIG. Através da detecção e processamento da união, interseção e diferença de polígonos, diversos tipos de operações, conhecidas como em conjunto como Polygon Overlay, são viabilizadas. São operações fundamentais para análise espacial, usadas em situações em que é necessário combinar ou comparar dados colocados em camadas distintas.

No contexto dos SGBDEs, as camadas citadas por Davis Jr. e Queiroz podem ser naturalmente entendidas, ou mesmo traduzidas, como tabelas que contenham campos destinados ao armazenamento de objetos espaciais.

Para se realizar operações do *Polygon Overlay*, é interessante aplicar um passo preliminar de detecção rápida da possibilidade de interseção entre os objetos. Assim, se não for possível que dois objetos P e Q tenham interseção, então podemos concluir diretamente que $P \cup Q = \{P, Q\}$, $P \cap Q = \emptyset$, $P - Q = P$ e $Q - P = Q$. Um modo simples de realizar este teste preliminar é utilizar inicialmente os REMs dos objetos para, só em caso de necessidade, utilizar os objetos propriamente ditos. A Figura 7 demonstra visualmente a execução das operações do *Polygon Overlay* sobre dois polígonos P e Q .

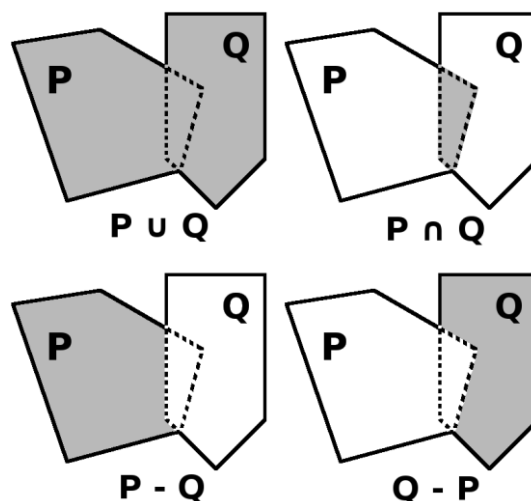


Figura 7 - Representação visual das operações do Polygon Overlay

Operações do *Polygon Overlay* entre dois polígonos simples podem gerar diversos polígonos como resultado, e ainda, os polígonos resultantes poderão conter “buracos” ou “ilhas”. Para este caso específico, Margalit e Knott (1989) apresentaram um algoritmo sensível à orientação dos polígonos, que exige que os vértices de ilhas sejam codificados em um sentido e os vértices de buracos sejam dispostos no sentido inverso. Tal metodologia permite um melhor aproveitamento do *Polygon Overlay*.

Estes cálculos, algoritmos, operações e conceitos colocados como de possível uso para a resolução de problemas envolvendo geometrias vetoriais são largamente utilizados para o tratamento de Relacionamentos entre Objetos Espaciais, que são tema do capítulo seguinte deste trabalho.

4 RELACIONAMENTOS ENTRE OBJETOS ESPACIAIS

Segundo o Dicionário Aurélio da Língua Portuguesa (FERREIRA, 1988), relacionamento é o “Ato ou efeito de relacionar.”, e relação é a “Conexão existente entre duas grandezas.”. Partindo destes conceitos, se pode entender que os Relacionamentos Entre Objetos Espaciais dizem respeito ao estabelecimento das conexões existentes entre tais objetos.

No âmbito espacial, os objetos podem se relacionar de diferentes formas levando-se em consideração diferentes características. Exemplos destas formas de relação são os relacionamentos topológicos, métricos e direcionais, que são abordados nos capítulos seguintes.

4.1 RELACIONAMENTOS TOPOLÓGICOS

Segundo Pinto (2009), “Em SIG, topologia é um conjunto de regras e comportamentos que se estipulam como pontos, linhas e polígonos e partilham geometrias coincidentes”. Assim sendo, Relacionamentos Topológicos dizem respeito a como as geometrias dos objetos espaciais se relacionam entre si. Para especificá-los, é necessária a definição das geometrias vetoriais como elementos do $\mathbb{R}^2$, considerado como espaço topológico (DAVIS JR. e QUEIROZ, 2005). Desta forma:

- Um ponto é simplesmente um elemento de $\mathbb{R}^2$;
- Uma linha L é um conjunto de pontos conectados. A fronteira de L (δL), é o conjunto formado pelos pontos inicial e final de L . O interior de L (L^0) é composto pelos demais pontos; e,
- Uma região R é um conjunto de pontos com um interior conectado, denotado por (R^0) , uma fronteira conectada, denotada por (δR) , e um único exterior conectado (R^-).

Estes relacionamentos topológicos podem ser definidos com base em um modelo, chamado *Matriz de 4-Interseções* (Figura 8), que considera oito relações

topológicas binárias, representando a interseção entre a fronteira e o interior de duas geometrias (EGENHOFER e FRANZOSA, 1995).

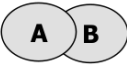
 $\begin{matrix} \partial A & \begin{pmatrix} \partial B & B^\circ \\ \emptyset & \emptyset \end{pmatrix} \\ A^\circ & \begin{pmatrix} \emptyset & \emptyset \end{pmatrix} \end{matrix}$ disjoint	 $\begin{matrix} \partial A & \begin{pmatrix} \partial B & B^\circ \\ \neg\emptyset & \emptyset \end{pmatrix} \\ A^\circ & \begin{pmatrix} \emptyset & \emptyset \end{pmatrix} \end{matrix}$ meet	 $\begin{matrix} \partial A & \begin{pmatrix} \partial B & B^\circ \\ \emptyset & \emptyset \end{pmatrix} \\ A^\circ & \begin{pmatrix} \neg\emptyset & \neg\emptyset \end{pmatrix} \end{matrix}$ contains	 $\begin{matrix} \partial A & \begin{pmatrix} \partial B & B^\circ \\ \neg\emptyset & \emptyset \end{pmatrix} \\ A^\circ & \begin{pmatrix} \neg\emptyset & \neg\emptyset \end{pmatrix} \end{matrix}$ Covers
 $\begin{matrix} \partial A & \begin{pmatrix} \partial B & B^\circ \\ \neg\emptyset & \emptyset \end{pmatrix} \\ A^\circ & \begin{pmatrix} \emptyset & \neg\emptyset \end{pmatrix} \end{matrix}$ equal	 $\begin{matrix} \partial A & \begin{pmatrix} \partial B & B^\circ \\ \neg\emptyset & \neg\emptyset \end{pmatrix} \\ A^\circ & \begin{pmatrix} \neg\emptyset & \neg\emptyset \end{pmatrix} \end{matrix}$ overlap	 $\begin{matrix} \partial A & \begin{pmatrix} \partial B & B^\circ \\ \emptyset & \neg\emptyset \end{pmatrix} \\ A^\circ & \begin{pmatrix} \emptyset & \neg\emptyset \end{pmatrix} \end{matrix}$ inside	 $\begin{matrix} \partial A & \begin{pmatrix} \partial B & B^\circ \\ \neg\emptyset & \neg\emptyset \end{pmatrix} \\ A^\circ & \begin{pmatrix} \emptyset & \neg\emptyset \end{pmatrix} \end{matrix}$ Covered By

Figura 8 - Matriz de 4-Interseções para relações entre duas regiões

Fonte: Egenhofer et al. (1994)

A definição de relacionamentos topológicos entre geometrias com estruturas mais complexas, como regiões com ilhas e separações, necessita de estender a matriz de 4-Interseções para considerar o exterior de uma geometria (EGENHOFER e HERRING, 1991). O novo modelo, chamado de *Matriz de 9-Interseções* (EGENHOFER *et al.*, 1994), considera então o resultado da interseção entre as fronteiras, interiores e exteriores de duas geometrias. Este modelo é mostrado na Figura 9.

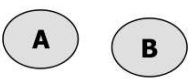
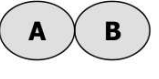
 $\begin{matrix} \partial A & \partial B & B^\circ & B^- \\ \begin{pmatrix} \emptyset & \emptyset & \neg\emptyset \\ \emptyset & \emptyset & \neg\emptyset \\ \neg\emptyset & \neg\emptyset & \neg\emptyset \end{pmatrix} \\ A^\circ & & & \\ A^- & & & \end{matrix}$ disjoint	 $\begin{matrix} \partial A & \partial B & B^\circ & B^- \\ \begin{pmatrix} \neg\emptyset & \emptyset & \neg\emptyset \\ \emptyset & \emptyset & \neg\emptyset \\ \neg\emptyset & \neg\emptyset & \neg\emptyset \end{pmatrix} \\ A^\circ & & & \\ A^- & & & \end{matrix}$ meet	 $\begin{matrix} \partial A & \partial B & B^\circ & B^- \\ \begin{pmatrix} \emptyset & \emptyset & \neg\emptyset \\ \neg\emptyset & \neg\emptyset & \neg\emptyset \\ \emptyset & \emptyset & \neg\emptyset \end{pmatrix} \\ A^\circ & & & \\ A^- & & & \end{matrix}$ contains	 $\begin{matrix} \partial A & \partial B & B^\circ & B^- \\ \begin{pmatrix} \neg\emptyset & \emptyset & \neg\emptyset \\ \neg\emptyset & \neg\emptyset & \neg\emptyset \\ \emptyset & \emptyset & \neg\emptyset \end{pmatrix} \\ A^\circ & & & \\ A^- & & & \end{matrix}$ covers
 $\begin{matrix} \partial A & \partial B & B^\circ & B^- \\ \begin{pmatrix} \neg\emptyset & \emptyset & \emptyset \\ \emptyset & \neg\emptyset & \emptyset \\ \emptyset & \emptyset & \neg\emptyset \end{pmatrix} \\ A^\circ & & & \\ A^- & & & \end{matrix}$ equal	 $\begin{matrix} \partial A & \partial B & B^\circ & B^- \\ \begin{pmatrix} \neg\emptyset & \neg\emptyset & \neg\emptyset \\ \neg\emptyset & \neg\emptyset & \neg\emptyset \\ \neg\emptyset & \neg\emptyset & \neg\emptyset \end{pmatrix} \\ A^\circ & & & \\ A^- & & & \end{matrix}$ overlap	 $\begin{matrix} \partial A & \partial B & B^\circ & B^- \\ \begin{pmatrix} \emptyset & \neg\emptyset & \emptyset \\ \emptyset & \neg\emptyset & \emptyset \\ \neg\emptyset & \neg\emptyset & \neg\emptyset \end{pmatrix} \\ A^\circ & & & \\ A^- & & & \end{matrix}$ inside	 $\begin{matrix} \partial A & \partial B & B^\circ & B^- \\ \begin{pmatrix} \neg\emptyset & \neg\emptyset & \emptyset \\ \emptyset & \neg\emptyset & \emptyset \\ \neg\emptyset & \neg\emptyset & \neg\emptyset \end{pmatrix} \\ A^\circ & & & \\ A^- & & & \end{matrix}$ covered by

Figura 9 - Matriz de 9-Interseções para relações entre duas regiões

Fonte: Egenhofer et al. (1994)

Nestes dois modelos de matrizes, os resultados das interseções são gerados considerando valores binários, ou seja, “existe” ou “não existe”, porém, há várias situações em que se necessita obter as dimensões das interseções existentes. Para suportar estes casos, novos modelos foram definidos levando em consideração as dimensões dos resultados das interseções existentes, como o *Modelo para Relações Topológicas Binárias Detalhadas*, apresentado por Egenhofer (1993), baseado na matriz de 4-interseções, enquanto Paiva (1998) apresenta a *Matriz de 9-Interseções Estendida Dimensionalmente* (DE-9IM), baseada na matriz de 9-interseções.

Clementini *et al.* (1993) estenderam a conceituação da Matriz de 4-Interseções para incluir a informação da dimensão da interseção. No espaço bidimensional, a dimensão da interseção pode ser vazia, um ponto, uma linha ou uma região. Este modelo definiu um total de cinquenta e dois relacionamentos topológicos distintos, que, para fins de conveniência e praticidade, foram generalizados para os cinco descritos e formalizados da seguinte maneira:

- *touch*: Aplica-se a pares de geometrias dos tipos região/região, linha/linha, linha/região, ponto/região e ponto/linha:

$$\langle \lambda_1, touch, \lambda_2 \rangle \Leftrightarrow (\lambda_1^0 \cap \lambda_2^0 = \emptyset) \wedge ((\partial\lambda_1 \cap \lambda_2^0 \neq \emptyset) \vee (\lambda_1^0 \cap \partial\lambda_2 \neq \emptyset) \vee (\partial\lambda_1 \cap \partial\lambda_2 \neq \emptyset))$$

- *in*: Aplica-se a pares de geometrias com qualquer combinação de tipos:

$$< \lambda_1, touch, \lambda_2 > \Leftrightarrow (\lambda_1^0 \cap \lambda_2^0 \neq \emptyset) \wedge (\lambda_1^0 \cap \lambda_2^- = \emptyset) \wedge (\partial \lambda_1 \cap \lambda_2^- = \emptyset)$$

- *cross*: Aplica-se a pares de geometrias dos tipos linha/linha e linha/região:

No caso de linha/região, tem-se:

$$< L, cross, R > \Leftrightarrow (L^0 \cap R^0 \neq \emptyset) \wedge (L^0 \cap R^- \neq \emptyset)$$

No caso de linha/linha, tem-se:

$$< L_1, cross, L_2 > \Leftrightarrow dim(L_1^0 \cap L_2^0) = 0$$

- *overlap*: Aplica-se a pares de geometrias dos tipos região/região e linha/linha:

No caso de região/região, temos:

$$< R_1, overlap, R_2 > \Leftrightarrow (R_1^0 \cap R_2^0 \neq \emptyset) \wedge (R_1^0 \cap R_2^- \neq \emptyset) \wedge (R_1^- \cap R_2^0 \neq \emptyset)$$

No caso de linha/linha, temos:

$$< L_1, overlap, L_2 > \Leftrightarrow (dim(L_1^0 \cap L_2^0) = 1) \wedge (L_1^0 \cap L_2^- \neq \emptyset) \wedge (L_1^- \cap L_2^0 \neq \emptyset)$$

- *disjoint*: Aplica-se a pares de geometrias com qualquer combinação de tipos:

$$< \lambda_1, disjoint, \lambda_2 > \Leftrightarrow (\lambda_1^0 \cap \lambda_2^0 = \emptyset) \wedge (\partial \lambda_1 \cap \lambda_2^0 = \emptyset) \wedge (\lambda_1^0 \cap \partial \lambda_2 = \emptyset) \wedge (\partial \lambda_1 \cap \partial \lambda_2 = \emptyset)$$

4.2 RELACIONAMENTOS MÉTRICOS

Segundo Bertini (2003) as relações espaciais são muito importantes em aplicações geográficas, sendo que o domínio da aplicação determinará a importância de cada tipo de relação. O Autor aponta os Relacionamentos Métricos como sendo relacionamentos espaciais que tratam de distâncias entre as geometrias dos objetos, o que, conseqüentemente remete à sua posição geográfica.

Cruz e Campos (2007) apontam que o dado espacial descreve fenômenos associados a dimensões espaciais. A representação de um objeto espacial é a

descrição de sua forma geométrica associada à posição geográfica. Pode-se definir como sendo propriedades geométricas de objetos espaciais as:

...representadas pelas feições geométricas primitivas (ponto, linha, polígono), para as quais se estabelecem relacionamentos métricos em relação a um sistema de coordenadas. As propriedades geométricas dependem: do comprimento, da sinuosidade e orientação (linha); do perímetro e área (polígono); do volume (entidades tridimensionais); da forma e da inclinação (linha e polígono) (Cruz e Campos, 2007, p. 2).

Nota-se então que a questão métrica está fortemente ligada a Sistemas de Medidas e a Sistemas de Coordenadas Geográficas, referentes ao tratamento de dimensões e posicionamentos, respectivamente.

Laurini e Thompson (1992) colocam que se deve ter cuidado com alguns conceitos, pois se observa que alguns destes conceitos espaciais podem ser medidos tanto no domínio geométrico quanto no topológico (não-métrico). A proximidade, por exemplo, pode ser obtida tanto através de adjacência quanto da distância Euclidiana².

4.3 RELACIONAMENTOS DIRECIONAIS

Direção é um conceito espacial comum que é usado em diversas situações no cotidiano, como no ato de localizar-se ou referenciar a uma entidade geográfica qualquer. Quando as pessoas se comunicam acerca do espaço geográfico, a direção é necessária para se transmitir informações (LIU, 2000), contudo, não há uma definição unificada de relações de direção (THEODORIDIS *et al.*, 1996). Também é importante ressaltar que alguns autores colocam os relacionamentos direcionais como sendo um sub-conjunto dos relacionamentos métricos.

A maioria das formalizações ou implementações de raciocínio espacial dependem da geometria euclidiana e do sistema de coordenadas cartesianas

² Método para cálculo de distância entre dois pontos utilizando o par de coordenadas cartesianas de cada ponto.

(FRANK, 1995), sendo que, no domínio geográfico, as direções são geralmente referidas como pontos cardeais e uma relação direcional é um trio $\langle A, d, B \rangle$, onde A e B são o *objeto de referência* e o *objeto alvo*, respectivamente, e d é um subconjunto não vazio de símbolos que indica em que direção B está em relação à A (EGENHOFER *et al.*, 2000).

Um dos grandes pontos de divergência entre os autores a respeito dos relacionamentos direcionais é no que diz respeito às direções definidas. Para Egenhofer *et al.* (2000), as direções correspondem aos pontos cardeais e colaterais acrescidos de uma área neutra. Já Theodoridis *et al.* (1996), adicionam designadores de intensidade às direções cardeais e colaterais, assim um objeto pode estar *fortemente*, *fracamente* ou *exatamente* em uma direção. Outros autores tendem a ser mais minimalistas, considerando somente os pontos cardeais.

As áreas pertencentes a cada direção, ou seja, as políticas de particionamento do espaço, também são um ponto de divergência entre os modelos. Em modelos derivados da grade Cônica, geralmente, as áreas referentes a cada direção tem igual dimensão e se apresentam em formato triangular. Já em modelos baseados na grade de Projeção com o uso do REM, as partições têm formato retangular, onde há uma partição central de área fixa e partições periféricas, referentes às direções cardeais e colaterais, com áreas diferentes entre si, onde esta diferença aumenta à medida que se distancia do centro. Assim a grade Cônica particiona o espaço de forma equalitária, mas a grade de Projeção não. Já modelos que se baseiam em ângulos não definem necessariamente partições do espaço, mas sim somente uma angulação.

4.3.1 Modelos para Definição de Relacionamentos Direcionais

Segundo Xia *et al.* (2007), os modelos básicos para a definição de relacionamentos direcionais dividem-se em duas grandes categorias principais, sendo estas as dos Modelos Baseados em Cone (PEUQUET e ZHAN, 1987) e dos Modelos Baseados em Projeção (FRANK, 1992).

Nos Modelos Baseados em Cone, o espaço é particionado utilizando-se linhas com um ângulo de origem α , conforme a Figura 10. Modelos típicos incluem modelo de 4-direções, Figura 10(a), o modelo de 8-direções, Figura 10(b), e o modelo de triângulo, Figura 10(c) (TANG *et al.*, 2008).

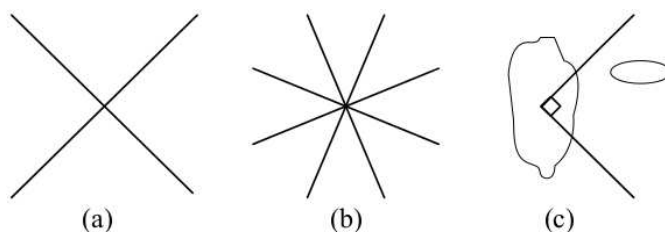


Figura 10 - Modelo Baseado em Cone

Fonte: Tang et al. (2008)

A plausibilidade cognitiva do modelo baseado em cone foi verificada por estudos no campo das ciências inter-relacionadas. Além disso, este modelo é uma aproximação típica para o campo de visão do olho humano e as lentes da câmara (TANG *et al.*, 2008). Por estas razões, os modelos baseados em cone têm sido utilizados em visão computacional, navegação de robôs, e SIG (SPIROS *et al.*, 2007).

Os Modelos Baseados em Cone podem dar uma identificação exata das relações direcionais em caso de geometrias pontuais, ao mesmo passo que podem indicar relações direcionais enganosas quando as geometrias tratadas são linhas ou polígonos (TANG *et al.* 2008).

Os Modelos Baseados em Projeção particionam o espaço utilizando linhas paralelas aos eixos (SPIROS *et al.*, 2007). O plano em torno de um objeto de referência A é particionado em nove áreas, sendo estas: norte (N_A), nordeste (NE_A), leste (E_A), sudeste (SE_A), sul (S_A), sudoeste (SW_A), oeste (W_A) e noroeste (NW_A) referentes aos pontos cardeais e colaterais, e uma região neutra adicional (0_A) conforme a Figura 11(a). Nesta categoria, o chamado Modelo REM, que expressa as relações direcionais a partir da relação entre o REM do objeto de referência e o objeto alvo, é de destaque (TANG *et al.* 2008). Neste modelo, as nove partições da projeção são alteradas, pois a região neutra corresponde ao REM do objeto de referência, conforme a Figura 11(b).

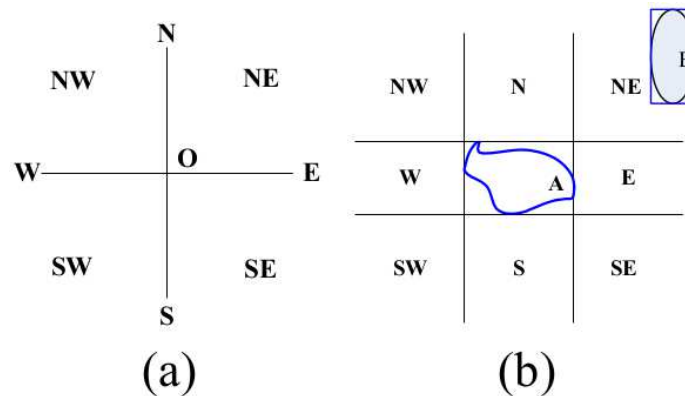


Figura 11 - Modelo Baseado em Projeção

Fonte: Tang et al. (2008)

Este modelo é mais expressivo para a definição de relações entre geometrias não pontuais. O seu ponto fraco é que a partição do espaço, em várias ocasiões, pode não ser natural. Normalmente, as pessoas tendem a organizar o espaço circundante usando linhas com um ângulo de origem semelhante ao modelo baseado em cone (TANG *et al.* 2008).

Ainda sobre os Modelos baseados em Projeção, existe o problema da discrepância entre as partições da projeção com o afastamento da origem, pois na medida em que se afasta desta, as partições referentes às direções cardeais crescem somente em um eixo espacial (X ou Y , nunca nos dois) e se mantém constante no outro, isto provoca um crescimento linear em uma grade com REM quadrado. Já no caso das direções colaterais, o crescimento se dá nos dois eixos do espaço, provocando um crescimento exponencial. A Figura 12 mostra como essa situação em uma grade usada em um polígono P , com um REM quadrado de lado 1 unidade, avançando-se 2 unidades, já a Figura 13 exhibe um gráfico que representa este crescimento até uma distância de 10 unidades e, ademais, a Figura 14 mostra como se dá a distribuição percentual do espaço entre as direções cardeais e as colaterais (desconsiderando a zona neutra). Neste caso específico, pelo fato de o REM ser quadrado, o crescimento nas direções colaterais é exatamente quadrático.

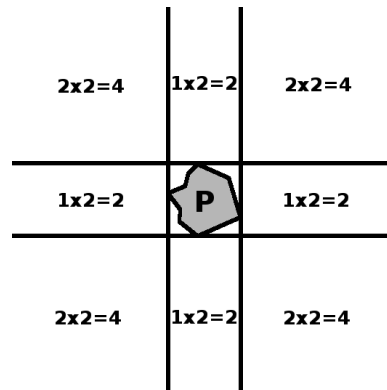


Figura 12 - Crescimento desigual entre as direções cardeais e colaterais

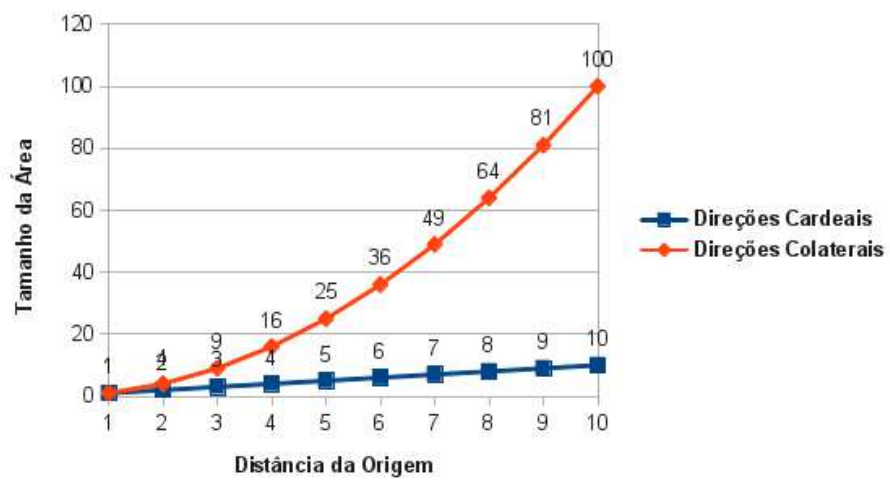


Figura 13 - Gráfico de crescimento das direções cardeais e colaterais

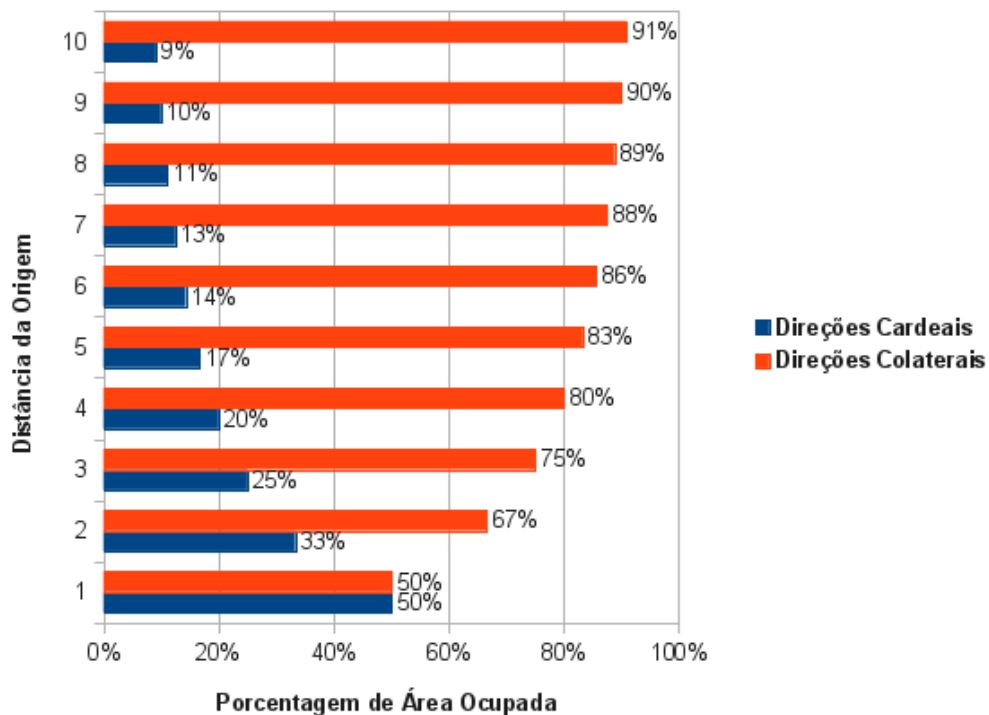


Figura 14 - Porcentagens do espaço ocupadas por direções cardeais e colaterais

Outra abordagem é a dos Modelos baseados em Voronoi (YAN *et al.*, 2006). Estes descrevem as relações direcionais pelas relações entre o objeto e seu *Diagrama de Voronoi*. Para descrever as relações direcionais com n direções, devem-se aplicar três estratégias (YAN *et al.*, 2013):

- Um único sentido pode ser descrito utilizando o azimute do normal da aresta de Voronoi: O azimute de um raio é o ângulo medido no sentido horário a partir da extremidade positiva do eixo vertical do sistema de coordenadas cartesianas para o raio;
- Para diferenciar a importância de cada direção, a cada sentido é atribuído um valor do peso que representa a porcentagem do comprimento de cada aresta de Voronoi correspondente: Como cada aresta de Voronoi corresponde a um único sentido, o Diagrama de Voronoi pode ser generalizado pela fórmula $S = \frac{d[1-\cos(\frac{\epsilon}{2})]}{2}$ para simplificar a expressão final das relações de direção;
- Para facilitar o armazenamento das relações de direção em bases de dados, todos os azimutes e seus pesos correspondentes são listados em uma tabela bidimensional contendo as direções como sendo ângulos associados a pesos;

Como citado anteriormente, a definição de direcionalidade não é unificada, assim sendo, existem n modelos que se propõem a realizar a tarefa de definição das relações direcionais. Modelos mais recentes incluem o Modelo Baseado em Nuvem (TANG *et al.*, 2008) e o Modelo Baseado em Geo-Ontologias (ZHU, 2012), tais modelos foram definidos por meio de pesquisas e trabalhos que serão vislumbrados no capítulo seguinte.

5 TRABALHOS RELACIONADOS

Egenhofer *et al.* (2000) introduzem uma variação do Modelo REM que utiliza a matriz de 9-interseções (EGENHOFER e HERRING, 1991), projetando uma grade sobre as geometrias em questão. Para a definição das relações entre dois polígonos, é usada uma matriz 3×3 que capta a vizinhança em torno do objeto de referência A e registra as interseções entre o objeto alvo B e os ladrilhos em torno do objeto de referência, referentes às nove áreas definidas pela projeção (seção 4.3.1), conforme a Equação 1.

$$dir_{RR}(A, B) = \begin{bmatrix} NW_A \cap B & N_A \cap B & NE_A \cap B \\ W_A \cap B & O_A \cap B & E_A \cap B \\ SW_A \cap B & S_A \cap B & SE_A \cap B \end{bmatrix} \quad (1)$$

Este modelo pode identificar binariamente se um objeto alvo B ocupa a região referente a uma determinada direção. Também se tem a flexibilidade para atribuição de pesos caso um objeto alvo B ocupe o espaço de mais de uma direção, permitindo a detecção de uma direção predominante, utilizando a Equação 2.

$$dir_{RR}(A, B) = \begin{bmatrix} \frac{area(NW_A \cap B)}{area(B)} & \frac{area(N_A \cap B)}{area(B)} & \frac{area(NE_A \cap B)}{area(B)} \\ \frac{area(W_A \cap B)}{area(B)} & \frac{area(O_A \cap B)}{area(B)} & \frac{area(E_A \cap B)}{area(B)} \\ \frac{area(SW_A \cap B)}{area(B)} & \frac{area(S_A \cap B)}{area(B)} & \frac{area(SE_A \cap B)}{area(B)} \end{bmatrix} \quad (2)$$

Cada célula da matriz apresentada na Equação 2 sempre terá um valor entre 0 e 1, podendo também ser traduzido para porcentagem. A Figura 15 mostra a aplicação deste modelo em três situações distintas, nas quais a posição do objeto alvo varia em relação à referência.

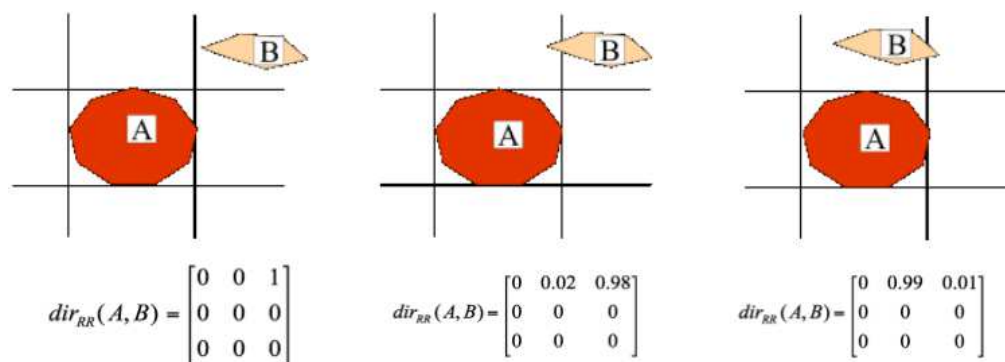


Figura 15 - Utilização do Modelo REM de Egenhofer

Fonte: Egenhofer et al. (2000)

O trabalho de Egenhofer formalizou um método para o tratamento de geometrias que estão em mais de uma direção segundo a projeção, porém o trabalho foi teórico. O autor não apresentou implementação, seja em um *software* terceiro ou diretamente em um SGBDE. Além disso, na medida em que se avança para longe da geometria de referência, as áreas dos pontos colaterais se tornam demasiadamente maiores que as dos cardeais. Ademais, este não é apropriado para tratar objetos pontuais, pois estes não possuem REM.

Tang *et al.* (2008) introduziram uma abordagem de relações direcionais baseada base no modelo de nuvem e no modelo baseado em cone. Para a aplicação de tal modelo, houve a necessidade do uso de métodos diferentes de acordo com diferentes tipos de objetos de referência e objetos alvo. Com base nos experimentos, realizados a partir de sua implementação, os autores classificaram o modelo concebido como sendo mais viável e exato do que os modelos tradicionais.

Tang elenca como uma das limitações do modelo desenvolvido o fato de o uso deste ser restrito a objetos com geometrias simples, preponderantemente pontos. A implementação do modelo, implementada em uma ferramenta de *software* externa, permite a visualização do resultado final de uma consulta espacial perviamente especificada. Esta ferramenta aponta numericamente a relação direcional entre duas geometrias pontuais.

Zhu *et al.* (2012) apresentam um modelo para definição de relacionamentos direcionais entre geometrias baseado em Geo-Ontologias (*gazetteers*). Neste modelo, os relacionamentos direcionais são determinados a partir de consultas

secundárias feitas sobre estas Geo-Ontologias por meio de um algoritmo com o fluxo demonstrado na Figura 16.

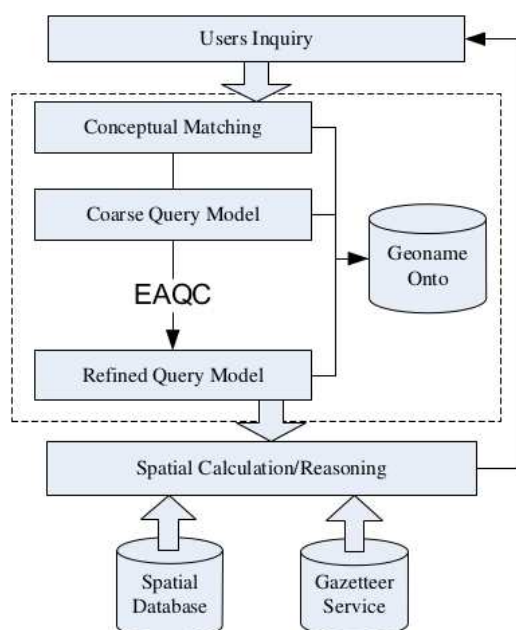


Figura 16 - Fluxo do modelo de Zhu

Fonte: Zhu et al. (2012)

O fluxo demonstrado na Figura 16 passa por um algoritmo interno, nomeado EAQC, ou *Extraction Algorithm of Query Characteristics*, que é responsável, basicamente, por extrair da ontologia informações que sejam relevantes ao raciocínio direcional. As informações extraídas são posteriormente utilizadas como parâmetros na segunda consulta, que é realizada sobre o SGBDE.

O modelo de Zhu é interessante pelo fato de adicionar semântica às pesquisas permitindo que o conhecimento relacionado à direcionalidade nos objetos representados na ontologia seja aproveitado. No entanto, a adição da semântica implica na existência de dados organizados em forma de base ontológica versando sobre o espaço pesquisado, o que possivelmente faz com que bases de dados espaciais já existentes não sejam compatíveis com este modelo, na medida em que não existam ontologias referentes ao seu contexto espacial em questão. Além disso, a base dados ontológica é externa ao SGBDE, culminando na necessidade de duas bases de dados, uma espacial e uma ontológica, separadas para a realização da pesquisa. Assim, a integração com os recursos já existentes do SGBDE se torna

problemática. O modelo ainda é teórico, sendo tratada a implementação como um trabalho futuro.

Yan *et al.* (2013) propõem uma abordagem das relações de direção de computação entre dois grupos de objetos usando o diagrama de Voronoi onde as arestas são usadas para expressá-las. A abordagem é apoiada adicionalmente por dois princípios da Teoria de Gestalt (EHRENFELS, 1890), sendo estes o “princípio da soma do conjunto e das suas partes”, e o outro é o “princípio da proximidade” que tem suas validade e solidez provadas por experimentos psicológicos. Yan resume as principais vantagens desta abordagem como: (1) ela pode calcular as relações direcionais entre grupos de objetos, o que os outros modelos não podem, (2) pode obter a direção $Dir(A, B)$ a partir da direção $Dir(B, A)$ sem cálculo complexo, (3) as relações de direção podem ser gravadas em tabelas bidimensionais, o que é útil na construção de um banco de dados espacial e raciocínio espacial. A integração com SGBDEs visada por este trabalho não é direta, pois se limita a permitir um resultado em formato que seja facilmente armazenado na base de dados. Para fim de permitir experimentos, foi implementado um *software* utilizando a linguagem C#, externo à base de dados.

Com base na fórmula matemática apresentada por Egenhofer *et al.* (2000) e no raciocínio de particionamento espacial de uma forma geral, foi criado o COP Model, visando a simplicidade e abrangência. Além disso, o COP Model foi pensado de forma a permitir sua implementação integrada ao SGBDE, pois assim se viria a preencher a lacuna existente no que diz respeito a esse modelo de implementação. As definição, implementação e verificação do COP Model são descritas no capítulo seguinte.

6 CONE-OVER-PROJECTION DIRECTIONAL MODEL

O COP Model é um modelo para definição de relacionamentos direcionais entre duas geometrias que utiliza como base a união e adaptação do Modelo Baseado em Cone de 8 Direções e do Modelo REM (Baseado em Projeção) introduzido por Egenhofer et. al. (2000). Dados estes dois modelos, o COP Model utiliza a capacidade de simular bem o ângulo de visão humano, que parte de um ponto central e cresce à medida que se afasta da origem, juntamente com a capacidade do modelo de Egenhofer *et al.* (2000) de tratar bem geometrias em forma de linha e polígono. O desenvolvimento do trabalho COP Model perpassou, de uma forma geral, por quatro etapas distintas explanadas a seguir.

- *Definição formal do modelo:* Nesta etapa, se formalizou o modelo proposto, que une as características do Modelo Baseado em Cone de 8 direções e do Modelo Baseado em Projeção introduzido por Egenhofer *et al.* (2000);
- *Elaboração de uma pseudo-implementação das funcionalidades:* A pseudo-implementação é genérica a ponto de permitir que seja traduzida para variadas linguagens de programação e em camadas diferentes, neste caso específico na aplicação terceira ou no banco de dados;
- *Desenvolvimento das funcionalidades em um SGBDE:* Uma vez se tendo definido os mecanismos lógicos e algoritmos a serem utilizados, foi possível implementá-los de forma agregada a um SGBDE; e,
- *Desenvolvimento de um módulo de SIG que utilize a extensão criada:* Verificaram-se as funcionalidades adicionadas pela extensão ao SGBDE. Este módulo pode ser criado em forma de um *software standalone* ou como extensão de algum SIG já existente, como o *gvSIG*, *Kosmo GIS* ou *QGIS*.

O cerne do modelo desenvolvido foi sobrepor a grade da projeção com a grade cônica a fim de identificar as áreas onde estas divergem, isto foi usado como uma chave para reduzir a diferença entre o tamanho das partições existentes no modelo de projeção.

Na grade cônica de 8 direções, o particionamento do espaço se dá de forma a manter as partições do espaço geradas todas com a mesma área. Assim sendo, tomando-se o mesmo exemplo de um avanço de 2 unidades e usando-se as

propriedades do Triângulo Retângulo mostradas de acordo com a Figura 17, tem-se o resultado mostrado na Figura 18.

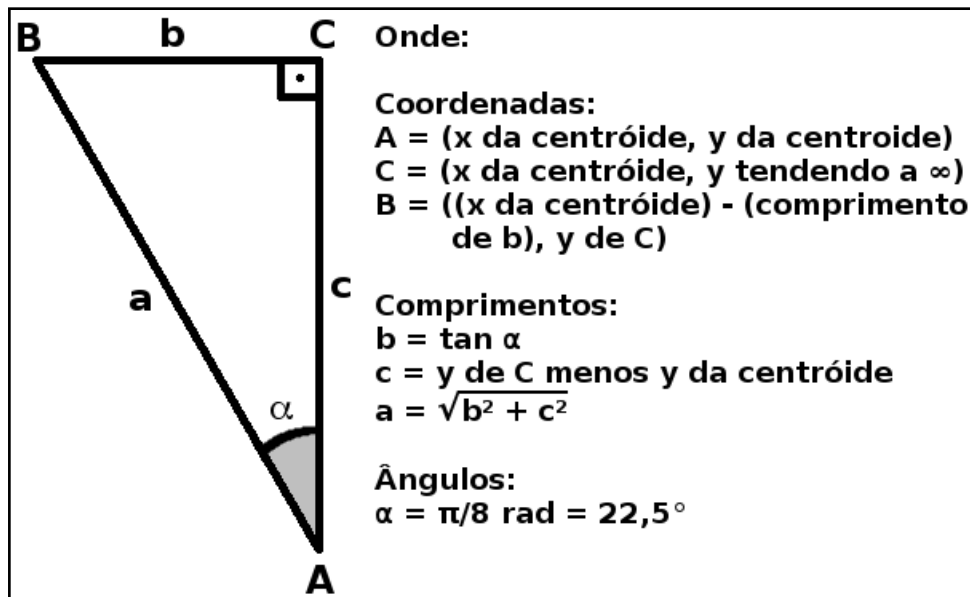


Figura 17 - Triângulo utilizado para criar as partições cônicas do espaço

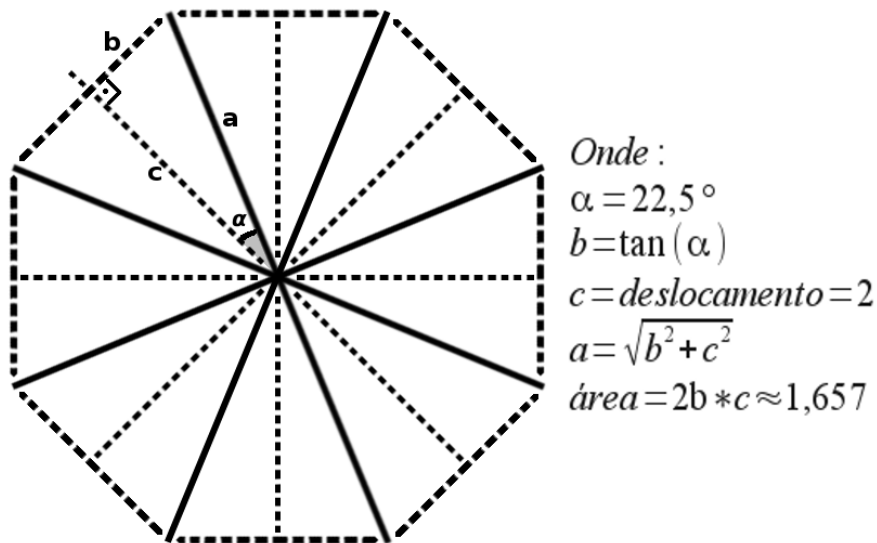


Figura 18 - Cálculo das áreas das partições cônicas

O triângulo mostrado na Figura 17 representa exatamente a metade de uma partição espacial da grade cônica. Assim sendo, para se obter a partição completa, basta duplicar este triângulo, espelhá-lo no eixo y e realizar a união com o triângulo original, resultando em um triângulo isósceles que representa a partição espacial. Deve-se notar que o ângulo α é fixado em $22,5^\circ$, isto ocorre pelo fato de que a grade cônica utilizada divide o espaço equalitariamente em 8 triângulos, cada qual com

uma abertura de 45° a partir da origem, e como o triângulo usado representa somente metade de uma partição, o valor do ângulo α deve ser igual a metade do ângulo de uma partição completa. Ademais, é necessário se corrigir a direção da partição, pois a criada até então aponta para o Norte. Isto se corrige deslocando-se o eixo de rotação do triângulo para o ponto A e fazendo rotações de 45° , 135° , 225° e 315° , apontando-se respectivamente para o Noroeste, Sudoeste, Sudeste e Nordeste.

Como as partições da grade cônica são sempre de área igual entre si, esta grade pode, se corretamente utilizada, ajudar a reduzir a discrepância da grade de projeção.

Por meio do uso de conceitos e algoritmos advindos da Geometria Computacional foi feita a união dos modelos de cone e projeção e o processamento de como os objetos se relacionam entre si e com as regiões definidas pelo modelo concebido.

6.1 ESPECIFICAÇÃO

A fim de verificar em que direções do espaço a grade de Projeção e a grade Cônica divergiram, foi necessário sobrepor à última. Este procedimento foi usado como uma forma de reduzir a diferença entre as áreas existentes no Modelo de Projeção. No modelo criado, as áreas de divergência entre as duas grades foram nomeadas conforme a Tabela 1, onde as direções colaterais, de acordo com a projeção, são dadas por $P(NE_A)$, $P(NW_A)$, $P(SE_A)$ e $P(SW_A)$, e as direções colaterais, de acordo com o cone, por $C(N_A)$, $C(S_A)$, $C(E_A)$ e $C(W_A)$, onde A é uma geometria qualquer do tipo Linha ou Polígono. As áreas restantes de divergência não foram considerados porque, se fossem, removeriam área das direções cardeais, reduzindo, assim, a diminuição da discrepância de que foi alcançada.

Tabela 1 - Formalização matemática das áreas de divergência entre as grades usadas

Região	Nome	Notação
$(P(NE_A) \cap C(N_A) \neq \emptyset) \vee (P(NW_A) \cap C(N_A) \neq \emptyset)$	<i>slightly_north</i> (A)	N_A^s
$(P(SE_A) \cap C(S_A) \neq \emptyset) \vee (P(SW_A) \cap C(S_A) \neq \emptyset)$	<i>slightly_south</i> (A)	S_A^s
$(P(NE_A) \cap C(E_A) \neq \emptyset) \vee (P(SE_A) \cap C(E_A) \neq \emptyset)$	<i>slightly_east</i> (A)	E_A^s
$(P(NW_A) \cap C(W_A) \neq \emptyset) \vee (P(SW_A) \cap C(W_A) \neq \emptyset)$	<i>slightly_west</i> (A)	W_A^s

Esta formalização adiciona regiões para a definição padrão de Modelos Baseados em Projeção, como mostrado na Figura 19, e pode ser usada para as relações entre Pontos, Linhas e Polígonos, no entanto, não é possível utilizá-lo se existem relações em que a Geometria de Referência é do tipo Ponto.

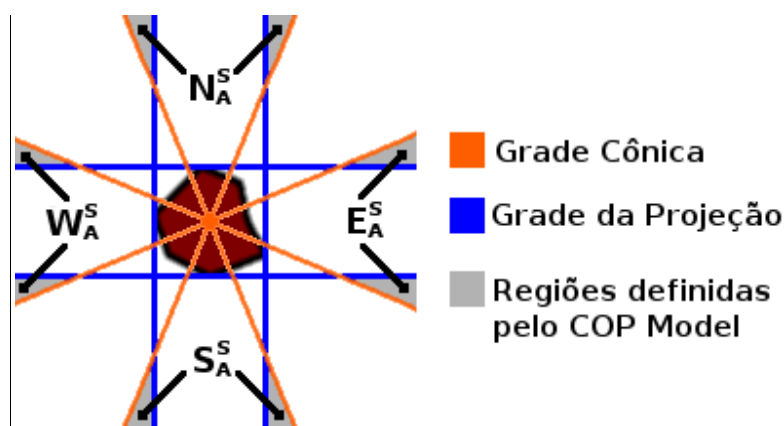


Figura 19 - Regiões adicionadas pelo COP Model

As regiões adicionadas pelo COP Model receberam uma nomenclatura formada pelo designador *slightly* (levemente) seguido do nome de uma direção Cardeal, pois estas estão em direções Colaterais segundo a grade de Projeção, porém, são influenciadas pelas direções Cardeais da grade Cônica.

A fórmula usada por Egenhofer (2000) em seu trabalho para mensurar quanto da área/extensão de uma geometria está em uma determinada partição de espaço também se aplica ao modelo criado, pois, embora originalmente disposta em forma de matriz, o valor correspondente a cada célula segue o padrão “área da geometria na região em questão” dividida pela “área da geometria”, o que facilmente se adéqua ao modelo criado. Assim, o Diagrama de Fluxo de Dados Simplificado para as funções do COP Model fica como o exposto na Figura 20.

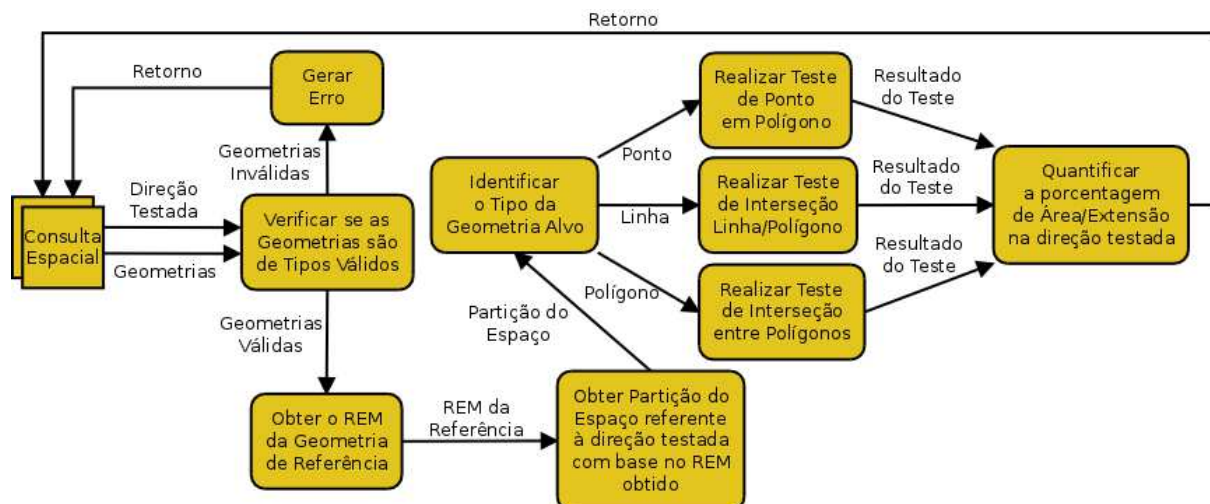


Figura 20 - Diagrama de Fluxo de Dados para as funções do COP Model

Dadas as geometrias e Direção testadas como entrada de dados faz-se necessária a verificação dos tipos de geometrias informados, pois, já que o modelo não suporta geometrias do tipo Ponto como referência, deve-se interromper o procedimento caso as geometrias não sejam compatíveis com o modelo; Uma vez sendo válidas as geometrias, inicia-se o tratamento da geometria de referência para obter seu REM, que é a base do particionamento espacial e corresponde à Zona Neutra do modelo; Deve-se em seguida obter a partição de espaço referente à direção testada, ressaltando que nesta etapa há diferenças sensíveis entre o procedimento para obtenção das partições de direções Cardeais, Colaterais e adicionadas pelo COP Model. Em termos gerais, o fato é que as direções Cardeais devem obedecer à grade de Projeção, as direções Colaterais devem obedecer à grade Cônica de 8 Direções e, por sua vez, as direções adicionadas pelo COP Model devem ser resultantes de subtrações e uniões entre as direções colaterais das grades anteriormente citadas; Posteriormente há a necessidade de identificar o tipo da Geometria Alvo, pois para cada tipo de alvo se tem a necessidade de um teste geométrico diferente em relação às partições do espaço; Uma vez tendo-se realizado o teste adequado, pode-se enfim quantificar o quanto da Geometria Alvo está na partição de espaço desejada em relação à Geometria de Referência em valores que variam de números de 0 a 1 (ou de 0% a 100%) e retornar este valor para a consulta que invocou a função.

Todas as funções que tratam de geometrias em relação às direções do COP Model estão ilustradas no fluxo genérico apresentado na Figura 20, porém, em cada uma destas são adicionadas as peculiaridades da direção tratada.

6.2 IMPLEMENTAÇÃO

A implementação do modelo foi feita a partir do código-fonte do SGBD *PostgreSQL*, em sua versão 9.1 juntamente com sua extensão espacial *PostGIS*, em sua versão 2.0, ambos *free* e *opensource*, lançando mão dos facilitadores providos pelo PGXS. A implementação propriamente dita deu-se com o uso da Linguagem de Programação C (ISO C), em conformidade com o código-fonte e aproveitando as estruturas pré-implementadas do *PostgreSQL/PostGIS* que é escrito nesta linguagem. Este tipo de abordagem trouxe benefícios, dentre os quais se podem citar:

- Reutilização dos recursos já existentes no SGBDE na criação das extensões;
- Possibilidade da realização de consultas SQL/SF-SQL utilizando funcionalidades, pré-existentes e adicionadas pelas extensões, juntas diretamente no SGBDE sem necessidade de *softwares* terceiros;
- Utilização transparente das extensões por *softwares* terceiros, uma vez que estas estariam incorporadas ao SGBDE; e,
- Facilidade de distribuição das extensões, uma vez que estas, depois de compiladas, seriam compatíveis com quaisquer instalações válidas do SGBDE usado como fonte para o desenvolvimento.

Este modelo de implementação permite que as extensões criadas sejam posicionadas no esquema de comunicação de SIGs e outras aplicações quaisquer que as utilizem conforme exibido na Figura 21.

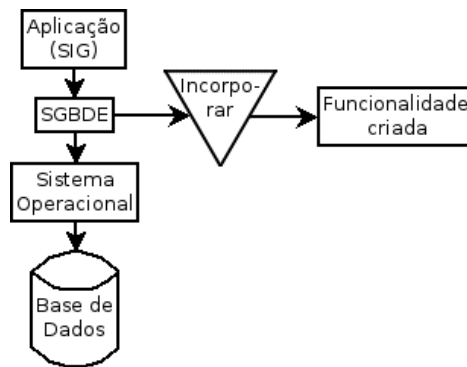


Figura 21 - Ilustração da camada na qual as extensões são inseridas

Ao seguir o esquema colocado na Figura 21, na implementação do COP Model em forma de extensão do SGBD, as funcionalidades do modelo ficam agregadas ao gerenciador de banco de dados, eximindo assim as aplicações que se conectam à base de dados de refazer a implementação do modelo, limitando-se a invocar o que já foi implementado e agregado à base de dados. Isto permite o reaproveitamento integral da implementação por quaisquer aplicações que se conectem com uma base de dados com a extensão criada registrada.

A implementação é compatível com os recursos existentes no SGBDE, permitindo a execução de consultas espaciais híbridas que utilizem recursos pré-existentes no SGBDE e recursos adicionados pelas extensões criadas. Assim, as consultas espaciais criadas poderão unir conceitos de topologia e distância com conceitos de direcionalidade sobre a mesma base de dados.

Uma vez de posse do código fonte do *PostgreSQL/PostGIS*, a implementação pôde ser iniciada fazendo uso de estruturas já existentes, como as implementações de estruturas de dados da hierarquia da SF-SQL padronizada pela OGC, de diversos algoritmos geométricos, como os vistos no Capítulo 2, de testes referentes à identificação de tipos de geometrias processadas pelas funções espaciais da extensão, além da possibilidade da utilização do *engine* de lançamento de erros do *PostgreSQL* e outros recursos existentes.

Para que o COP Model pudesse ser implementado como extensão do *PostgreSQL*, este foi codificado integralmente em código C ANSI, seguindo o paradigma da Programação Estruturada. Toda a codificação se concentrou em dois arquivos, sendo o *copdirectionalmodel.h*, onde foram definidos macros, protótipos de funções, constantes e onde foi feita a inserção de cabeçalhos, e

copdirectionalmodel.c, onde foi feita a codificação propriamente dita das funções espaciais do modelo.

Por se tratar de uma extensão do *PostgreSQL*, a codificação obrigatoriamente necessitou de ser iniciada com a adição da macro *PG_MODULE_MAGIC*, que identifica a implementação como sendo de uma extensão válida do *PostgreSQL*. É importante frisar que se esta macro não for adicionada, não será possível registrar a extensão em bases de dados do *PostgreSQL*, pois para tal operação, é obrigatória a presença do *footprint* da macro *PG_MODULE_MAGIC*.

Uma vez identificada como extensão válida, para que se continuasse a implementação do COP Model, foi necessário analisar a especificação deste para se elencar os passos a serem seguidos e os recursos que seriam utilizados. Uma análise inicial mostrou que, cada função espacial, deveria ser uma especificação do algoritmo mostrado na Figura 22 em forma de pseudo-código.

```
FUNÇÃO copDireção(geomReferência, geomAlvo: Geometry): Real  
INICIO  
    var extensãoOuÁrea, porcentagem: Real;  
    var partiçãoDireção, remReferência: Geometry;  
  
    SE (geomReferência é Ponto)  
        lançarErro('geomReferência não pode ser ponto.');    FIM SE  
  
    remReferência <- obterRem(geomReferência);  
    partiçãoDireção <- obterPartiçãoDireção(remReferência);  
  
    CASE geomAlvo  
        CASE é Ponto:  
            extensãoOuÁrea <- pontoEmPolígono(geomAlvo, partiçãoDireção);  
  
            SE (extensãoOuÁrea > 0)  
                porcentagem <- 1.0;  
            SENÃO  
                porcentagem <- 0.0;  
            FIM SE  
  
        CASE é Linha:  
            extensãoOuÁrea <- interseção(geomAlvo, partiçãoDireção);  
            porcentagem <- extensãoOuÁrea / extensãoLinha(geomAlvo);  
  
        CASE é Polígono:  
            extensãoOuÁrea <- interseção(geomAlvo, partiçãoDireção);  
            porcentagem <- extensãoOuÁrea / áreaPolígono(geomAlvo);  
    FIM CASE  
  
    Retornar porcentagem;  
FIM
```

Figura 22 - Generalização das funções implementadas no COP Model

O pseudocódigo mostrado na Figura 22 é uma generalização da funcionalidade de identificação de relacionamento direcional que pode ser implementada em variadas Linguagens de Programação e em diferentes camadas dos SIGs que a utilizem.

Tendo a generalização sido criada, passou-se para a etapa de implementação propriamente dita das funcionalidades em forma de funções em uma extensão do SGBD, seguindo o modelo de desenvolvimento proposto.

O Particionamento Espacial foi realizado com o auxílio de funcionalidades legadas do código-fonte do *PostGIS* e de conceitos matemáticos. É de se ressaltar que não há a necessidade de obter as partições espaciais de todas as direções, mas sim somente da direção correntemente testada. Houve a necessidade de tratamentos diferenciados para se obter as partições de espaço referentes aos diferentes tipos de direção do COP Model (Cardeais, Colaterais ou Adicionadas pelo modelo).

As direções Cardeais seguem o Modelo REM baseado em Projeção, desta forma, obter as partições espaciais referentes a este tipo de direção limitou-se identificar o retângulo que imediatamente se concatena ao REM da geometria de referência na direção testada. Assim sendo, os vértices da partição espacial são dois pertencentes ao REM citado e dois calculados conforme o mostrado na Figura 23, para as direções paralelas ao eixo X, e na Figura 24, para as direções paralelas ao eixo Y.

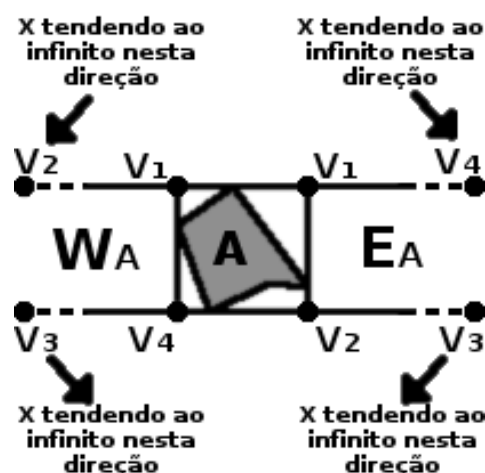


Figura 23 - Vértices das partições nas direções paralelas ao eixo X

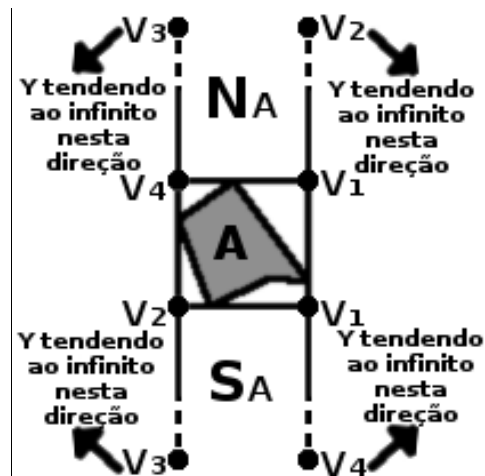


Figura 24 - Vértices das partições nas direções paralelas ao eixo Y

Por sua vez, as direções Colaterais do COP Model seguem a definição da grade Cônica, então, diferentemente das partições de espaço Cardeais que são identificadas por retângulos, as partições Colaterais são identificadas por triângulos. Cada triângulo referente às partições espaciais possui como vértices a centróide do REM da geometria de Referência e mais dois pontos calculados conforme mostrado anteriormente na Figura 17 neste capítulo.

Já as direções com o designador *slightly*, que foram as adicionadas pelo COP Model, seguem a definição matemática mostrada na Tabela 1, o que pode ser traduzido, de maneira geral, como uma diferença entre direções Colaterais da grade de Projeção e da grade Cônica. Isto pode ser conseguido utilizando-se um raciocínio mais simples, mostrado na Equação 3.

$$PE = DCP - DCC$$

Onde:

$$PE = \text{Partição Espacial} \quad (3)$$

$$DCP = \text{Direção Colateral na Projeção}$$

$$DCC = \text{Direção Colateral no Cone}$$

Desta forma é possível obter as partições de espaço referentes às direções *slightly* a partir das definições Cardeais que as influenciam. Adicionalmente, o *PostGIS* já traz uma implementação da operação de Diferença entre Polígonos, que pode ser utilizada a partir do seu código-fonte. Isto facilita mais ainda a implementação.

No entanto, esta abordagem gera um problema: a partição espacial resultante acaba sendo configurada em forma de ilhas, ou seja, múltiplas regiões. Isto acarreta erros especialmente nos testes de Ponto em Polígono, pois estes não comportam testar se um ponto está ou não em N polígonos, mas sim em 1 polígono. Para sanar esta limitação, houve a necessidade de tratar as ilhas que fazem parte da Partição Espacial de forma individual, fazendo os testes em uma ilha por vez.

Uma vez obtidas as partições espaciais referentes a cada direção do COP Model, bastou se utilizar testes de Interseção entre Geometrias e Ponto em Polígono entre estas e a Geometria Alvo. Finalmente, o resultado foi traduzido para valores em porcentagem utilizando-se a fórmula apresentada no Modelo REM de Egenhofer *et al.* (2000).

Tendo-se efetuando assim a implementação propriamente dita do COP Model, foi necessária a criação dos outros artefatos que a arquitetura PGXS exige. Estes consistem majoritariamente em declarações e configurações.

O Arquivo de Controle, nomeado *copdirectionalmodel.control*, limitou-se a especificar a versão corrente da extensão, definir o nome do módulo e o local de sua instalação, e adicionar breves comentários acerca da extensão, todos com notação de texto pleno.

O descritor SQL, nomeado *copdirectionalmodel--0.1.sql*, conteve o protótipo das treze funções espaciais implementadas que são visíveis ao gerenciador de banco de dados, deixando inacessíveis funções criadas com intuito simplesmente de apoio.

Finalmente, o *makefile* foi substituído por um *script* com notação *Shell Script*, que faz a compilação e instalação da extensão. Optou-se por esta substituição pelo fato de que ela se mostrou uma solução mais simples e totalmente compatível com o PGXS.

Desta forma, foram implementados e produzidos todos os artefatos necessários para o uso do COP Model de forma integrada com qualquer instalação válida do SGBD *PostgreSQL* 9.1+ e a extensão espacial *PostGIS* 2.0+, e a este conjunto de artefatos foi dado nome de *COP Directional Model Extension* (COP-ME).

6.2.1 Verificação da Implementação

A fim de verificar se a implementação estava em conformidade com o Modelo COP, foi desenvolvido um SIG, nomeado *COP Directional Model Viewer* (COP-VW), direcionado para testar as funcionalidades do COP Model e o funcionamento deste de forma integrada com o SGBDE. Desta forma, o COP-VW não implementa nenhuma funcionalidade de processamento espacial, limitando-se a utilizar as funcionalidades do SGBDE e do COP Model, além de prover interface gráfica para entrada e saída de dados.

O COP-VW é um *software standalone* foi desenvolvido utilizando a Linguagem de Programação Java e suas APIs integrantes, como *Java Database Connectivity* (JDBC) para realizar a conexão com o SGBDE, *Java Sun Windowing* (SWING) para a criação da interface gráfica e *Java 2D Graphics API* (Java 2D) para a exibição gráfica das informações recuperadas do SGBDE. O COP-VW exime o usuário de escrever consultas SQL/SF-SQL diretamente, gerando-as a partir de entradas feitas visualmente. A janela principal do COP-MV é mostrada na Figura 25. Nesta janela é possível visualizar informações acerca da conexão com o Banco de Dados e projeção utilizada pelas geometrias, Figura 25(a); a consulta espacial gerada pela ferramenta, Figura 25(b); botões de criação e submissão de consultas, Figura 25(c); listagem de camadas espaciais presentes na base de dados correntemente utilizada, Figura 25(d); representação gráfica das geometrias carregadas da base de dados e dos resultados das consultas espaciais submetidas, Figura 25(e); e detalhes textuais recuperados das consultas espaciais, Figura 25(f).

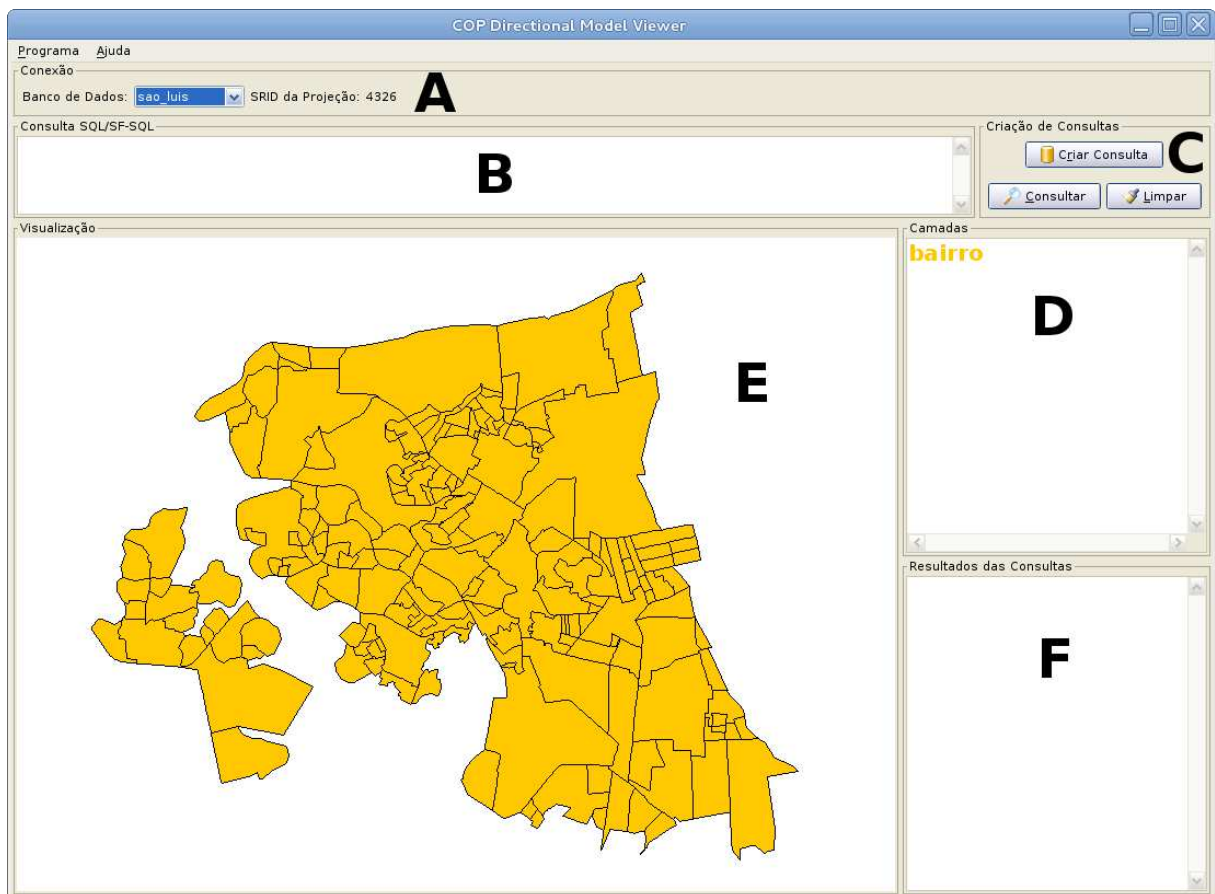


Figura 25 - Tela principal do COP-VW

O painel de visualização das geometrias mostrado na Figura 25(e) provê, adicionalmente, funcionalidades de translação e escala das geometrias exibidas, o que permite uma melhor avaliação dos resultados das consultas.

As consultas espaciais são criadas a partir de informações fornecidas pelo usuário em uma janela específica para a criação destas. Foram implementadas funcionalidades para a criação de duas diferentes categorias de consultas sendo estas: *Testar Alvo* e *Encontrar Alvo(s)*.

Consultas do tipo *Testar Alvo* requerem que sejam informados dois Objetos Espaciais, sendo que um será o Objeto de Referência e o outro será o Objeto Alvo, e uma direção qualquer do COP Model. Dadas estas entradas, o COP-VW solicita ao SGBDE acrescido da implementação do COP Model que calcule qual porcentagem da área/comprimento do Objeto Alvo está na direção informada em relação ao Objeto de Referência. A janela de criação deste tipo de consulta é mostrada na Figura 26, a consulta SF-SQL gerada pelo COP-VW na Figura 27,

enquanto o resultado da submissão de uma consulta deste tipo é mostrado na Figura 28.

Configuração da Consulta

Geral

Tipo de Consulta: Testar Alvo

Objeto de Referência

Tabela: bairro Objeto: MONTE CASTELO

Objeto Alvo

Tabela: bairro Objeto: RENASCENÇA

Relacionamento Direcional

Direção Testada: Noroeste

Ok Cancelar

Figura 26 - Janela de configuração de consulta *Testar Alvo*

```
SELECT stx_cop_north_west(E_REFERENCIA.geometry, E_ALVO.geometry) AS "value"  
FROM bairro E_ALVO, bairro E_REFERENCIA  
WHERE E_REFERENCIA.id_bairro = 50  
AND E_ALVO.id_bairro = 115
```

Figura 27 - Código gerado pelo COP-VW para uma consulta *Testar Alvo*

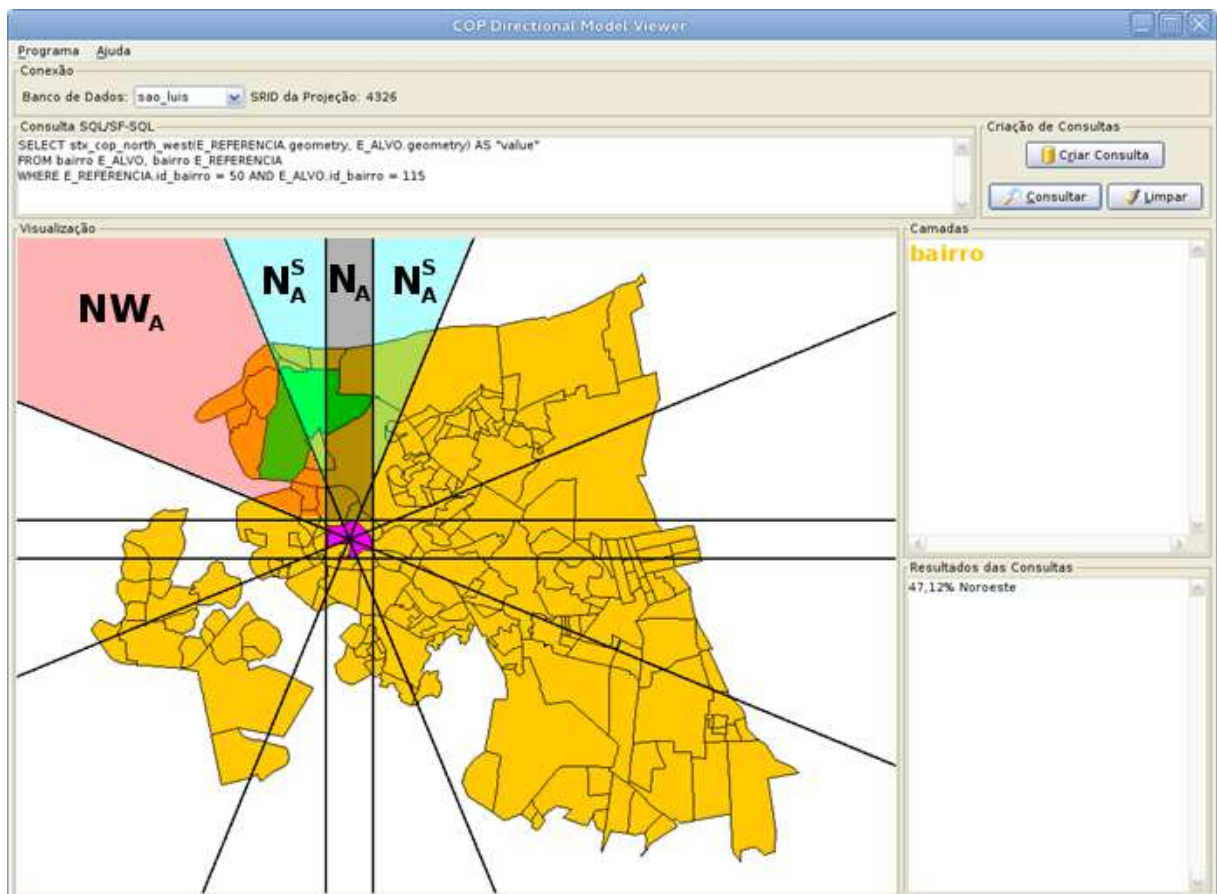


Figura 28 - Saída gerada por uma consulta *Testar Alvo*

É possível verificar na Figura 28 que a saída gerada por uma consulta do tipo *Testar Alvo* consiste em um apontamento visual do Objeto de Referência (em magenta), do Objeto Alvo (em verde) e das linhas de grade da Projeção e do Cone. Adicionalmente, é mostrada de forma textual a informação acerca do resultado da consulta constando de direção e porcentagem do Objeto Alvo nesta direção.

Consultas do tipo *Encontrar Alvo(s)* requerem que sejam informados o Objeto de Referência, uma direção na qual vai se buscar por potenciais alvos e a camada na qual se vai realizar a busca. Como se é considerado a porcentagem de área/comprimento dos alvos, também é solicitado um limiar de porcentagem para que um objeto seja realmente considerado alvo válido. Consultas deste tipo ainda oferecem a possibilidade de adição de testes de Relacionamentos Topológicos e/ou Métricos, assim, o resultado da busca pode ser refinado com base nestas características dos Objetos Espaciais. Dadas estas entradas, o COP-VW solicita ao SGBDE acrescido da implementação do COP Model que encontre os Objetos Espaciais que se enquadram nos requisitos e limitações impostas. A janela de

criação deste tipo de consulta é mostrada na Figura 29, a consulta SF-SQL gerada pelo COP-VW na Figura 30, enquanto o resultado da submissão de uma consulta deste tipo é mostrado na Figura 31.

A janela "Configuração da Consulta" apresenta as seguintes configurações:

- Geral:** Tipo de Consulta: Encontrar Alvo(s)
- Objeto de Referência:** Tabela: bairro, Objeto: MONTE CASTELO
- Camada de Procura:** Tabela: bairro
- Relacionamento Direcional:** Direção Testada: Nordeste, Limiar: 30 %
- Outros Relacionamentos:**
 - Adicionados: NOT (st_touches(E_REFERENCIA.geometry, E_ALVO.geometry)) AND st_distance(E_REFERENCIA.geometry, E_ALVO.geometry) > 1.0

Figura 29 - Janela de configuração de consulta *Encontrar Alvo(s)*

```
SELECT E_ALVO.texto AS "nome", E_ALVO.id_bairro AS "id",
       stx_cop_north_east(E_REFERENCIA.geometry, E_ALVO.geometry) AS "porcentagem"
FROM bairro E_ALVO, bairro E_REFERENCIA
WHERE E_REFERENCIA.id_bairro = 50
      AND stx_cop_north_east(E_REFERENCIA.geometry, E_ALVO.geometry) >= 0.3
      AND NOT (st_touches(E_REFERENCIA.geometry, E_ALVO.geometry))
      AND st_distance(E_REFERENCIA.geometry, E_ALVO.geometry) > 1.0
```

Figura 30 - Código gerado pelo COP-VW para uma consulta *Encontrar Alvo(s)*

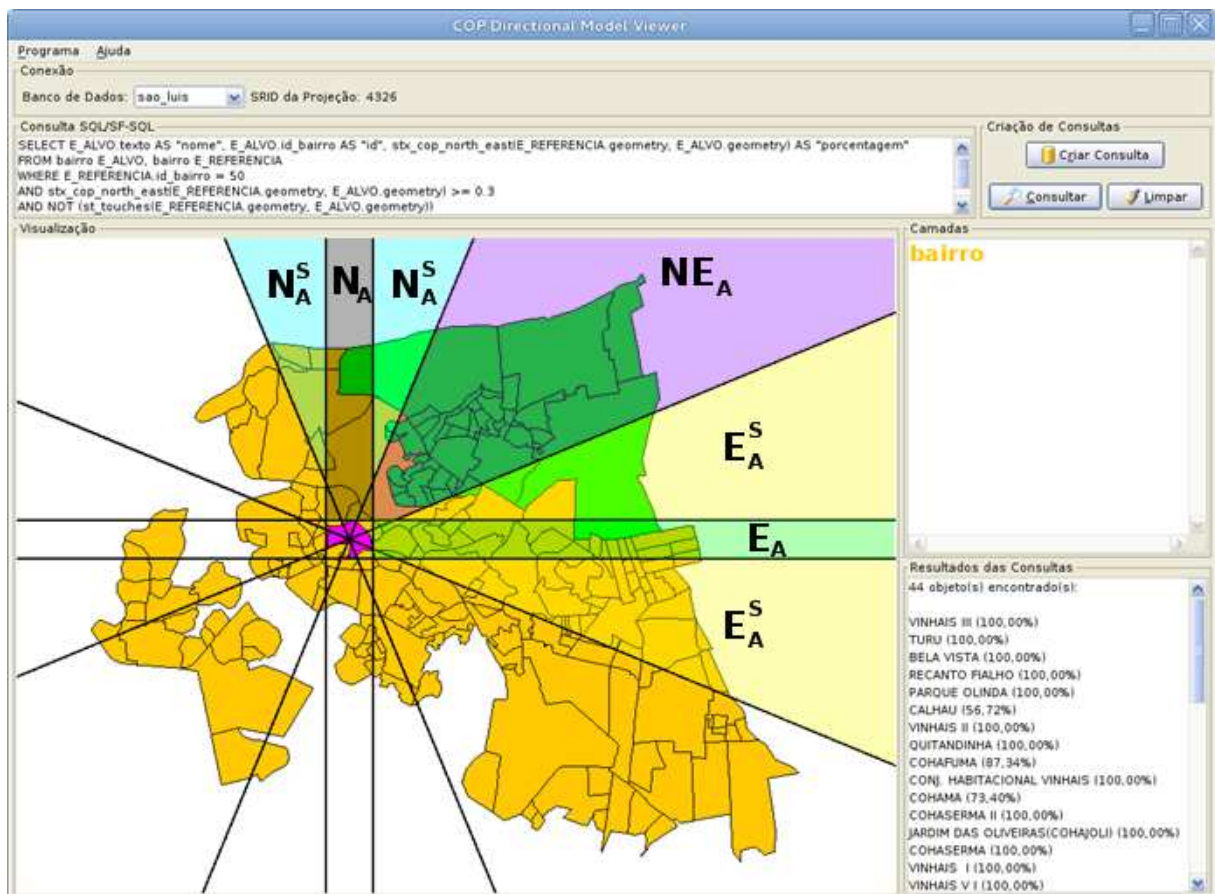


Figura 31 - Saída gerada por uma consulta *Encontrar Alvo(s)*

Nota-se que a saída deste tipo de consulta segue o mesmo padrão de cores e apontamentos visuais que a do tipo *Testar Alvo*, mas neste caso são destacadas todos os Objetos Alvo que atendem à consulta, com uma entrada textual para cada um destes informando a sua porcentagem dentro da direção consultada.

O COP-VW também permite a submissão de consultas que relacionem tabelas diferentes e com Tipos Geométricos diferentes, proporcionando consultas envolvendo os pares Polígono/Polígono, Polígono/Linha, Polígono/Ponto, Linha/Polígono, Linha/Linha e Linha/Ponto. Todos estes casos são implementados na extensão do SGBDE e somente invocados a partir do COP-VW. O resultado de uma consulta realizada com tabelas diferentes e tipos de geometrias diferentes pode ser visto na Figura 32.

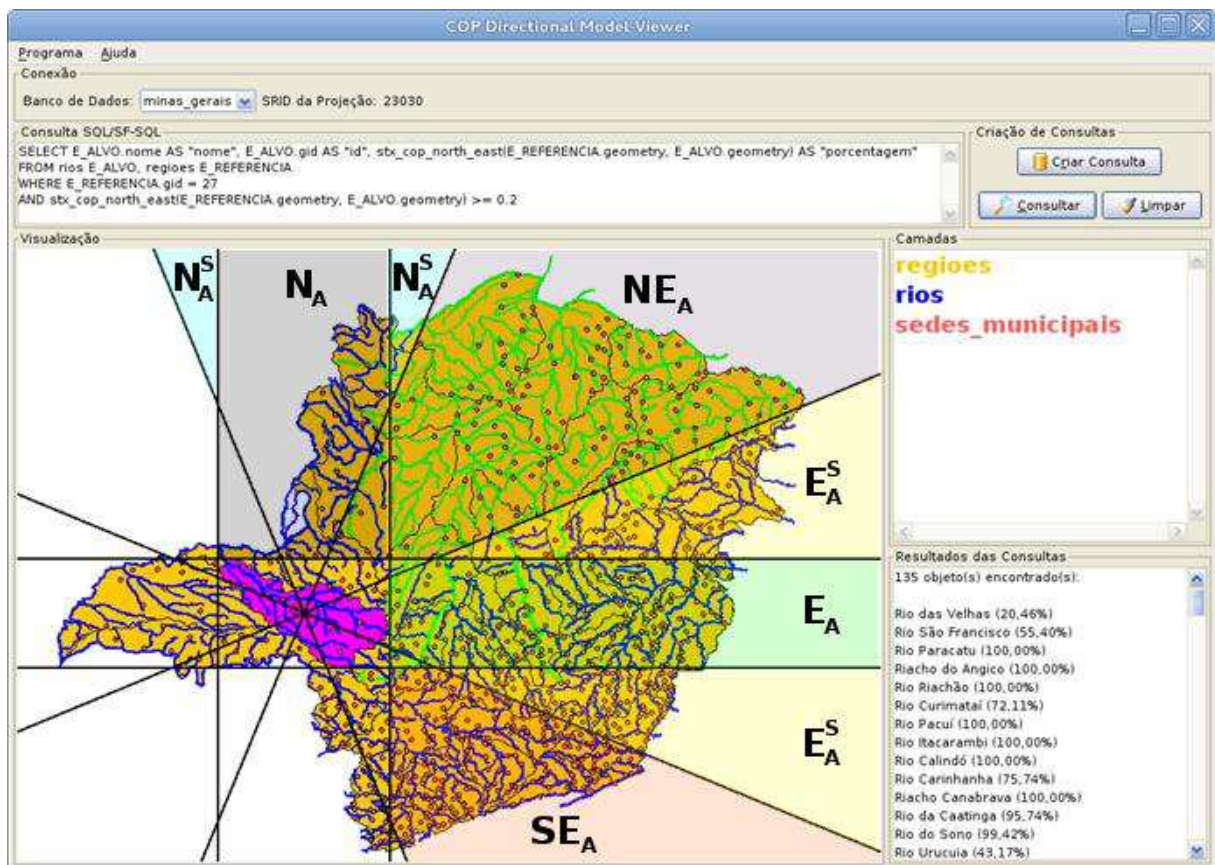


Figura 32 - Saída gerada por uma consulta com tabelas e geometrias diferentes

O COP-VW utilizou a implementação do COP Model integrada ao SGBDE para realizar as operações de consultas espaciais, eximindo-se de implementar tais funcionalidades. Assim sendo, a codificação deste foi centrada no gerenciamento da conexão com o SGBDE, recuperação e representação gráfica dos dados espaciais e criação de interface amigável para entrada de dados, pois a complexidade do raciocínio espacial foi encapsulada pela extensão adicionada ao SGBDE.

Houve a necessidade da criação de classes de modelo para representar as informações trazidas do SGBDE e geradas internamente no COP-VW, porém, como a complexidade do raciocínio espacial foi abstraída, o modelo pôde ser simples, contando somente com duas classes. As classes de modelo, juntamente com suas principais dependências provenientes da API padrão da linguagem de programação Java e da API fornecida pelo *PostgreSQL/PostGIS*, são expressadas no diagrama de classes exposto na Figura 33.

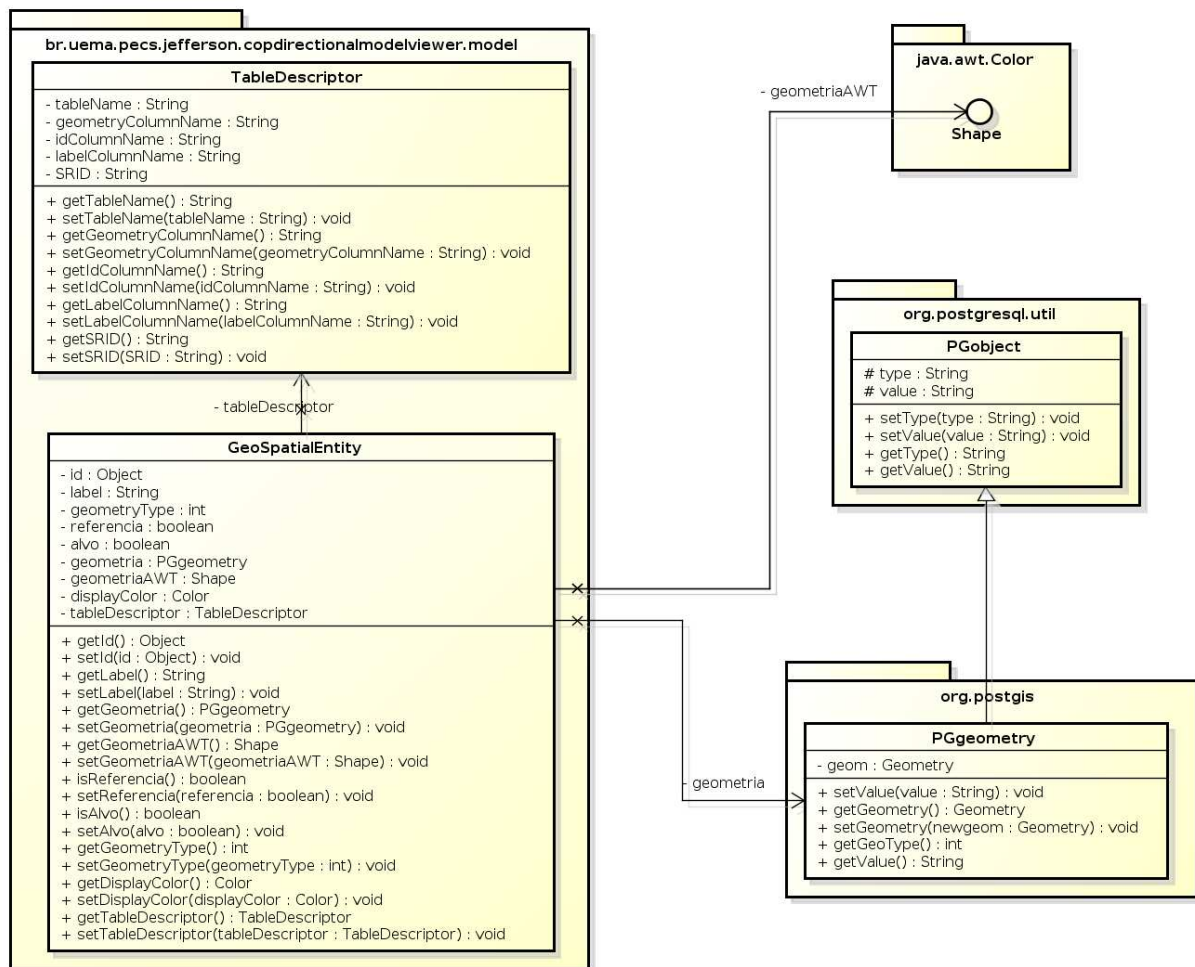


Figura 33 - Classes de modelo e suas dependências usadas no COP-VW

A classe *TableDescriptor* enfoca na representação genérica das características de uma relação do SGBDE que contenha atributos espaciais, de forma que o número total de instâncias desta classe é igual ao número destas relações na base de dados. Já a classe *GeoSpatialEntity* agrupa informações acerca de um objeto espacial recuperado do SGBDE, além da proveniência deste e da forma como este será exibido para o usuário por meio da interface gráfica, assim, o número de instâncias desta classe é igual ao número de tuplas de todas as relações da base de dados que contém atributos espaciais.

6.3 RESULTADOS

Como resultados deste trabalho, obtiveram-se basicamente três artefatos tangíveis: A definição matemática do COP Model, o COP-ME que representa o pacote de produtos que forma a implementação integrada ao SGBDE e o SIG COP-VW que foi usado para verificação. Além dos artefatos, uma contribuição acadêmica foi produzida. O COP Model foi apresentado e publicado nos anais do XIV Simpósio Brasileiro de Geoinformática - GEOINFO 2013 (DA SILVA e FOOK, 2013).

A definição matemática foi a base do trabalho como um todo, servindo de apoio para qualquer outro artefato, pois sem esta não seria possível garantir que a implementação teria base matemática concreta sobre a qual estaria embasada, estando assim sujeita a brechas que possivelmente seriam enxergadas na etapa de verificação da implementação.

O COP-ME foi produzido visando manter à risca os conceitos da definição do COP Model e a almejada integração com o SGBDE. O produto final não só atendeu à estas expectativas iniciais como também possui: (a) alto desacoplamento, pois pode ser simplesmente registrado ou desregistrado em uma determinada base de dados sem prejuízos para esta; (b) reutilização, pelo próprio fato de estar localizada na mesma camada do SGBDE; (c) portabilidade, pois pode ser usado em diferentes plataformas de Sistema Operacional sem necessidade de modificações a nível de código; e (d) compatibilidade, já que pode ser registrada em qualquer base de dados *PostgreSQL/PostGIS* que atenda aos requisitos mínimos de *software*.

O COP-VW veio como último artefato de *software* do ciclo de vida do trabalho. Com ele foi possível não só efetuar a validação da implementação do COP Model, como também demonstrar que de fato as funções de processamento espacial implementadas no COP-ME são utilizáveis por *softwares* terceiros sem necessidade de reimplementação ou qualquer adequação específica.

6.3.1 Discussão

Como mostrado anteriormente no início do Capítulo 6, a grade de projeção tende a particionar o espaço de forma desproporcional na medida em que se afasta da origem, privilegiando as direções Colaterais. O COP Model conseguiu minimizar isto superpondo a esta a grade cônica. Fazendo-se a verificação das partições geradas pelo COP Model com os mesmos parâmetros usados na grade de Projeção, ou seja, REM quadrado com 1 unidade de lado e deslocamento de 1 a 10 unidades para longe da origem, temos o crescimento das áreas das partições mostrado na Figura 34 e a Figura 35 mostra como se dá a distribuição percentual do espaço entre as direções cardeais, colaterais e as adicionadas pelo COP Model.

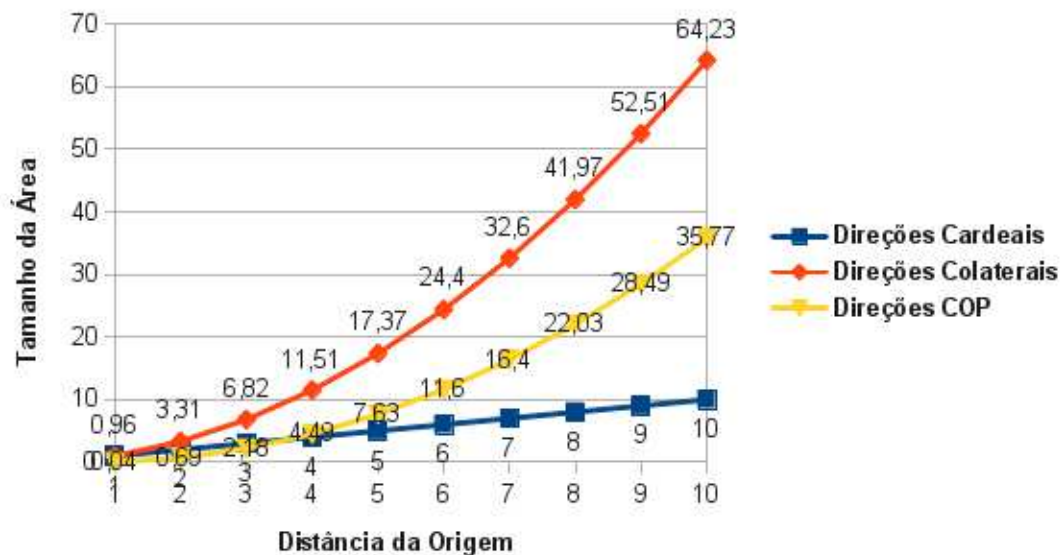


Figura 34 - Gráfico de crescimento das direções no COP Model

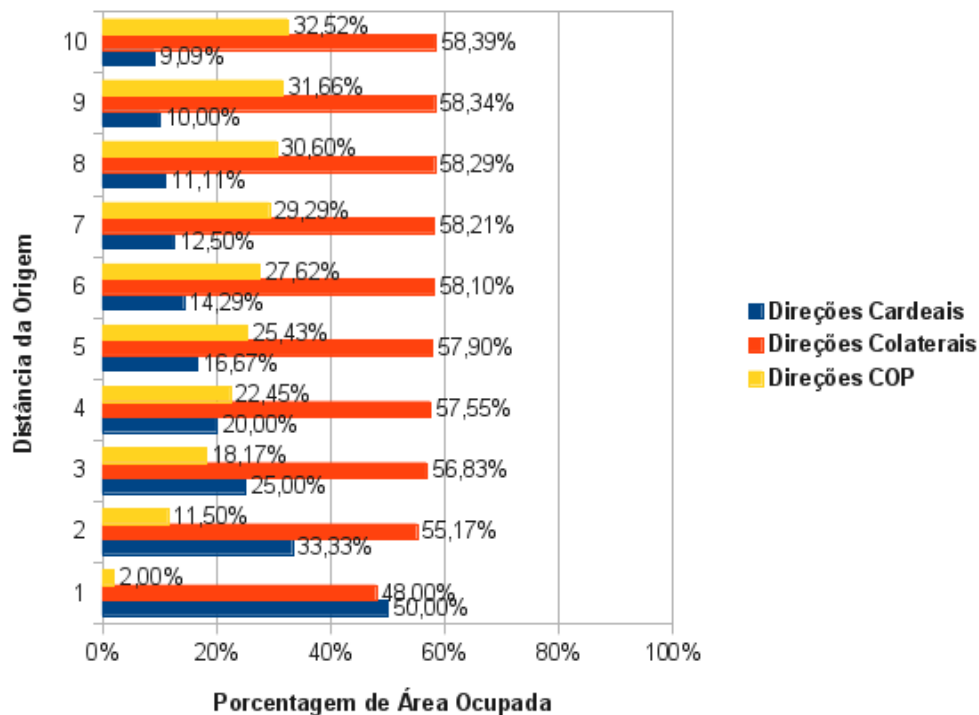


Figura 35 - Porcentagens do espaço ocupadas pelas direções do COP Model

É possível verificar na Figura 34 um crescimento ainda exponencial da área referente às direções Colaterais e linear para as direções Cardeais. Porém as direções adicionadas pelo COP Model também crescem de maneira exponencial, o que equilibra o crescimento das direções colaterais fazendo com que este não chegue aos 59% do total da área particionada, conforme a Figura 35. Se for considerado que as direções COP são influenciadas pelas direções cardeais (conforme disposto no capítulo 6.1) pode-se, de acordo com a interpretação de quem faz a consulta espacial, somar as áreas das direções cardeais com as das suas respectivas direções COP e assim equilibrar cardeais e colaterais. Considerando esta possibilidade de interpretação, a Figura 36 mostra um comparativo entre as porcentagens de área ocupadas por direções cardeais e colaterais com a grade de Projeção e com a grade do COP Model, agrupando nesta última os valores referentes às direções cardeais e suas respectivas influenciadas.

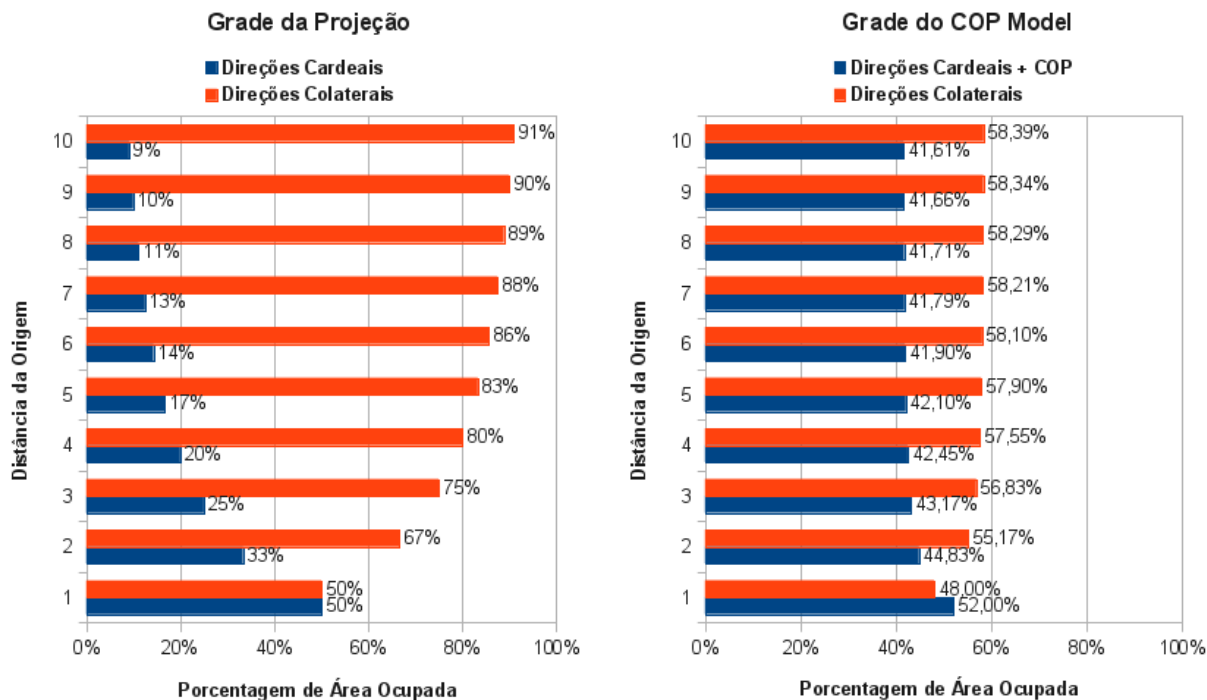


Figura 36 - Comparação entre as partições da Projeção e do COP Model

Analisando a Figura 36 pode-se entender que, com a grade do COP Model, o crescimento das áreas colaterais deixa de ser exponencial e passa a ter um comportamento mais próximo da linearidade. Percebe-se também que na medida em que se afasta da origem, a discrepância a favor das direções colaterais continua crescendo, porém em grandezas cada vez menores. De uma forma global, a diferença de áreas que hora havia chegado a 82% na grade de Projeção caiu para 16,78% com a grade do COP Model nesta dada situação. Assim, a discrepância entre as áreas foi efetivamente reduzida.

7 CONCLUSÃO

Este trabalho apresentou o *Cone-Over-Projection Directional Model* (COP Model) desde a sua definição matemática, permeando a implementação e chegando à verificação da implementação. O COP Model veio ser um modelo resultante da união de conceitos de outros modelos já existentes, no caso das famílias de modelos baseados em Cone e baseados em Projeção. Além disso, na implementação do COP Model – que se traduz no COP-ME – foi usada a abordagem de integração com o SGBDE ao invés de efetuá-la na camada do SIG.

Como os modelos de Cone e de Projeção possuem deficiências e não existe uma padronização amplamente aceita para relacionamentos direcionais, o COP Model veio contribuir reduzindo as deficiências dos modelos supracitados e oferecer uma nova alternativa flexível para tratar estes relacionamentos. Nesse contexto é importante frisar que o COP Model conseguiu reduzir a discrepância entre os tamanhos das partições espaciais, presente nos modelos de projeção, e é compatível com geometrias lineares e poligonais como referência, o que era problemático nos modelos de cone.

O fato de o COP Model ter sido implementado integrado ao SGBDE também pode ser considerado como uma contribuição, pois essa implementação integrada traz benefícios e se notou uma grande carência na adoção desse tipo de abordagem em trabalhos existentes. Pode-se elencar o alto desacoplamento, a possibilidade de reutilização, a portabilidade e a manutenção de compatibilidade com bases de dados existentes, além do fato de que, como a implementação foi feita a partir do código-fonte do *PostgreSQL/PostGIS*, foi possível a utilização de código-fonte já existente na implementação nativa do SGBDE.

Em relação às tecnologias utilizadas na produção do COP-ME, pode-se destacar que os artefatos criados para atender às exigências do *engine* PGXS podem perfeitamente ser utilizados como *templates* para outros projetos que sejam implementados de forma integrada ao SGBD.

O COP-VW permitiu verificar a implementação realizada do COP Model de forma visual e amigável. A partir deste, as funcionalidades implementadas puderam ser testadas de maneira individual e em conjunto com funcionalidades já existentes

no SGBDE, como as referentes a relacionamentos topológicos e métricos, de forma que foi de fato verificado que a implementação foi fiel à definição do modelo. Ademais, o COP-VW pode também ser utilizado como visualizador gráfico de geometrias vetoriais armazenadas em quaisquer bases de dados que use *PostgreSQL/PostGIS*, independentemente do uso ou não do COP-ME.

Adicionalmente, destaca-se que todos os procedimentos realizados e produtos criados foram feitos utilizando-se tecnologias *free/opensource*, o que tornou o COP-ME e o COP-VW livres da necessidade de pagamento de licenças, tanto para quem desenvolve quanto para quem usa. Assim, quem desejar reaproveitar qualquer módulo destes artefatos também usufruirá deste mesmo benefício.

7.1 TRABALHOS FUTUROS

O COP Model é passível de melhorias. Perspectivas de trabalhos futuros podem se concentrar em diferenciados aspectos de suas definição e implementação.

Um desses aspectos é o fato de a Zona Neutra do modelo, que corresponde ao REM do Objeto de Referência, por vezes englobar um espaço considerável, especialmente quando o referido Objeto tem formato distante de um quadrilátero. Isto pode tornar consultas imprecisas, uma vez que áreas importantes que contenham outros Objetos podem ser consideradas como neutras. Esse comportamento pode ser visualizado na Figura 37.

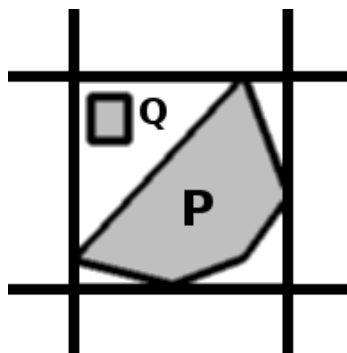


Figura 37 - Problema gerado pelo REM de certos Objetos

Na Figura 37 o objeto *P* gera um REM bastante diferente da sua forma original, e o objeto *Q* acaba sendo englobado por esse REM e consequentemente

pela Zona Neutra, muito embora ele visivelmente esteja em uma direção referente à Q . Assim sendo uma melhoria futura pode ser eliminar o uso do REM, fazendo com que a Zona Neutra se reduza a somente a área da geometria do Objeto de Referência. Ainda nesta linha de raciocínio, pode-se trabalhar no sentido de determinar a frequência com que o problema elencado anteriormente ocorre em grandes bases de dados reais, identificando assim o quão relevante seria o tratamento deste.

Outro aspecto que pode ser futuramente trabalhado é o de determinar quantitativamente ou qualitativamente, no caso das direções com o designador *slightly*, o quanto estas são realmente influenciadas pelas direções Cardeais na medida em que se afasta da origem. Isto poderia ser útil para definir níveis de influência que poderiam ser utilizados como critério para a escolha de agregação ou não das direções *slightly* com as Cardeais em determinadas consultas.

Finalmente, o custo computacional do uso do COP Model em situações reais com massas de dados de diferentes tamanhos pode ser medido e analisado a fim de verificar se este é adequado para uso em ambientes de produção. Este mesmo procedimento pode ser adotado no caso da retirada do REM, neste caso a fim de verificar se os benefícios trazidos por esta modificação impactam em um aumento aceitável do custo computacional.

REFERÊNCIAS

BERTINI, G. C., **Uma modelagem orientada a objeto para o mapa urbano básico de Belo Horizonte (MUB/BH)**, Monografia de Especialização, Pontifícia Universidade Católica de Minas Gerais, 2003.

CLEMENTINI, E., DI FELICE, P., VAN OOSTEROM, P., **A Small Set of Formal Topological Relationships Suitable for End-User Interaction**. In: ABEL, D.; OOI, B. C., eds., SSD '93: Lecture Notes in Computer Science, v. 692: New York, NY, USA, Springer-Verlag, p. 277-295, 1993.

CRUZ, I., CAMPOS, V. B., **Sistemas de Informações Geográficas Aplicados à Análise Espacial em Transportes, Meio Ambiente e Ocupação do Solo**. Instituto Militar de Engenharia – IME, Rio de Janeiro–RJ, 2007.

DA SILVA, J. A., FOOK, K. D., **Addition of the Directionality Concept in Spatial Queries on SDMSs Using the Union of the Cone-Based and Projection-Based Models**. XIV Simpósio Brasileiro de Geoinformática (GEOINFO 2013). Campos do Jordão – SP, 2013, ISSN: 2179-4820.

DAVIS JR, C., QUEIROZ, G., **Geometria Computacional para Sistemas de Informação Geográfica**, Instituto Nacional de Pesquisas Espaciais - INPE, São José dos Campos–SP, Brasil, 2005.

EGENHOFER, M., HERRING, J., **Categorizing Binary Topological Relationships Between Regions, Lines, and Points in Geographic Databases**. Department of Surveying Engineering, University of Maine, Orono, ME, USA, 1991.

EGENHOFER, M., **A Model for Detailed Binary Topological Relationships**. *Geomatica*, v. 47, p. 261-273, 1993.

EGENHOFER, M., FRANZOSA, R. **On the Equivalence of Topological Relations**. *International Journal of Geographical Information Systems*, v. 9, n.2, p. 133-152, USA, 1995.

EHRENFELS, C., **Über Gestalt Qualitäten**, Vierteljahresschr. f. wissensch. Philosophie, 1890, 3. Heft, S. 249-292. Zeitschrift für Psychologie und Physiologie der Sinnesorgane 2: 245-265, 1890.

EGENHOFER, M., **Qualitative Spatial-Relation Reasoning for Design**. National Center for Geographic Information and Analysis, Department of Spatial Information Science and Engineering Department of Computer Science. University of Maine Orono, ME 044690-5711, USA, 2000.

ELMASRI, R., **Sistemas de Bancos de Dados**. Addison Wesley, São Paulo-SP, Brasil, 2005.

FERREIRA, Aurélio Buarque de Holanda. **Dicionário Aurélio Básico da Língua Portuguesa**. Rio de Janeiro: Nova Fronteira, 1988.

FERREIRA, N., **Apostila de Sistemas de Informações Geográficas**. Centro Federal de Educação Tecnológica de Goiás, Goiânia, GO, Brasil, 2006.

FERREIRA, K. R., CASANOVA, M. A., QUEIROZ, G. R., OLIVEIRA, O. F., **Bancos de dados geográficos**, Instituto Nacional de Pesquisas Espaciais - INPE, São José dos Campos-SP, Brasil, 2005.

FIGUEIREDO, L. H., CARVALHO, P. C. P., **Introdução à geometria computacional**. Rio de Janeiro: IMPA, 1991.

FRANK, A. U., **Qualitative spatial reasoning about distances and directions in geographic space**, J. Visual Lang. Comput., 3, 343–371, 1992.

FRANK, A., **Qualitative Spatial Reasoning: Cardinal Directions as an Example**. Department of Geoinformation, Technical University of Vienna, Gusshausstrasse 27-29, A - 1040 Wien, Austria, 1995.

HAINES, E. **Point in polygon strategies**. In: HECKBERT, P. S. (Org.). Graphics gems IV. Boston, E.U.A.: Academic Press, 1994. p. 24-46.

LAURINI, R., THOMPSON, D., **Fundamentals of Spatial Information Systems**. London: Academic Press, 1992. 680p.

LIU, X., **Modeling and Processing Directional Relationships in Spatial Databases**. Department of Computer Science and Engineering University of Minnesota, ISBN10: 0599864389, ISBN13: 9780599864382, Minneapolis, MN 55455-0159, USA, 2000.

MARGALIT, A., KNOTT, G. D., **An algorithm for computing the union, intersection or difference of two polygons**. Computers & Graphics, v. 13, n. 2, p. 167-183, 1989.

MELO, C., GUERRA, M., **SGBD com Extensão Espacial e Sistemas de Geoinformação: Um Casamento Perfeito**. Universidade Federal de Minas Gerais - UFMG, Revista Fonte, Brasil, 2005.

O'ROURKE, J., **Computational Geometry in C**. Cambridge: Cambridge University Press, 1998.

PAIVA, J. A. C., **Topological Equivalence and Similarity in Multi-Representation Geographic Database**. University of Maine, 1998.

PEUQUET, D., ZHAN, C. X., **An Algorithm to Determine the Directional Relation Between Arbitrarily-Shaped Polygons in the Plane**, Pattern Recogn., 20, 65–74, 1987.

POSTGRESQL EXPERTS INC., **PostgreSQL Extension Network Official Site**, <<http://pgxn.org/about/>>, Acessado em: 15 de Set. de 2013.

PRADO, F., **Stored Procedures, Functions e Packages em bancos de dados Oracle**. SQL Magazine 102, DevMedia, 2008.

SAALFELD, A. **It doesn't make me nearly as CROSS - some advantages of the point-vector representation of line segments in automated cartography**. International Journal of Geographical Information Systems, v. 1, n. 4, p. 379-386, 1987.

PREPARATA , F. P., SHAMOS , M. I. **Computational geometry an introduction**. New York: Springer-Verlag, USA, 1985.

SANTOS, P., **Tudo sobre Procedures**, Código Fonte, 2008. Disponível em: <<http://codigofonte.uol.com.br/artigos/tudo-sobre-procedures>>. Acessado em: 15 de Set. de 2013.

SHAMOS, M. I., HOEY, D., **Geometric intersection problems**. In: Annual IEEE Symposium on Foundations of Computer Science, 17., oct. 1976, Houston, Texas. Proceedings. New York: IEEE, 1976, p. 208-215, USA, 1976.

SPIROS, S., NIKOS, S., TIMOS, S., MANOLIS, K., **A Family of Directional Relation Models for Extended Objects**, IEEE Transactions on Knowledge and Data Engineering, 19(8), pp.1116-1129, 2007.

STONEBRAKER, M., **Object-relational DBMSs: The next great wave**. San Francisco: Morgan Kaufmann, 1996.

TANG, X., MENG, L., QIN, K., **Study On The Uncertain Directional Relations Model Based On Cloud Model**. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXXVII. Part B2. Beijing, 2008.

THEODORIDIS, Y., PAPADIAS, D., STEFANAKIS, E., **Supporting Direction Relations in Spatial Database Systems**. Spatial data handling International symposium; 7th, Spatial data handling, ISBN: 0748405917, 1997.

TAYLOR, G. E.; **Point in Polygon Test**. Survey Review, v. 32, n. 254, p. 479-484, 1994.

UCHOA, H. N., COUTINHO, R. C., FERREIRA, P. R., FILHO, L. C., BRITO, J. L., **Análise do Módulo PostGIS (Opengis) para Armazenamento e Tratamento de Dados Geográficos com Alta Performance e Baixo Custo**, OpenGEO, XXV Congresso da Sociedade Brasileira de Computação, Canela – RS, Brasil, 2005.

XIA, Y., ZHU, X., LI, D., QIN, K., **Research on Spatial Directional Relation Description Model**, Science of Surveying and Mapping, 32(5), pp.94-97, 2007.

YAN, H., WANG, Z., LI, J., **An Approach to Computing Direction Relations Between separated Object Groups**. Department of Geographic Information Science, Faculty of Geomatics, Lanzhou Jiaotong University, European Geosciences Union, Copernicus Publications, Lanzhou 730070, China, 2013.

YAN, H., CHU, Y., LI, Z., GUO, R., **A quantitative direction description model based on direction groups**. *Geoinformatica*, 10, 177–196, 2006.

ZHU, X., CHEN, D., ZHOU, C., LI, M., XIAO, W., **Cardinal Direction Relations Query Modeling Based on Geo-Ontology**, State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XXXIX-B2, 2012 XXII ISPRS Congress, 25 August – 01 September 2012, Melbourne, Australia, 2012.