UNIVERSIDADE FEDERAL DE JATAÍ (UFJ)
INSTITUTO DE CIÊNCIAS EXATAS E TECNOLÓGICAS (ICET)
CURSO DE CIÊNCIA DA COMPUTAÇÃO

Gustavo Rezende Gouveia

Caracterização do Erro de Estimativa de Seletividade para Consultas Espaciais de Janela

Jataí-Goiás

Março de 2024

Gustavo Rezende Gouveia

Caracterização do Erro de Estimativa de Seletividade para Consultas Espaciais de Janela

Trabalho de Conclusão de Curso apresentado ao curso de Ciência da Computação do Instituto de Ciências Exatas e Tecnológicas da Universidade Federal de Jataí (UFJ), como requisito para obtenção do título de Bacharel em Ciência da Computação.

Orientador(a): Prof. Dr. Thiago Borges de Oliveira do Curso de Ciência da Computação

Jataí-Goiás

Março de 2024

Ficha de identificação da obra elaborada pelo autor, através do
Programa de Geração Automática do Sistema de Bibliotecas da UFJ.

Gouveia, Gustavo Rezende
Caracterização do Erro de Estimativa de Seletividade para Consultas Espaciais de Janela / Gustavo Rezende Gouveia. - 2024.
59 f.: il.

Orientador: Prof. Dr. Thiago Borges de Oliveira.
Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Jataí, Instituto de Ciências Exatas e Tecnológicas, Ciência da Computação, Jataí, 2024.
Apêndice.
Inclui siglas, abreviaturas, tabelas, lista de figuras, lista de tabelas.

1. Histogramas. 2. Erro de Estimativa de Seletividade. 3. Consulta de Janela. 4. Gerador de Datasets. I. Oliveira, Thiago Borges de, orient. II. Título.

CDU 004

UNIVERSIDADE FEDERAL DE JATAÍ
INSTITUTO DE CIÊNCIAS EXATAS E TECNOLÓGICAS

DECLARAÇÃO DE APROVAÇÃO DA VERSÃO FINAL

Declaro que o(a) discente Gustavo Rezende Gouveia do curso de Bacharelado em Ciência da Computação foi aprovado(a) na defesa do Trabalho de Conclusão de Curso (TCC) com o título final Caracterização do Erro de Estimativa de Seletividade para Consultas Espaciais de Janela na data de 07/03/2024 e efetuou todas as correções pertinentes sugeridas pela banca examinadora, composta pelo seguintes membros:

Orientador(a)	Thiago Borges de Oliveira
Membro 1	Ariadne de Andrade Costa
Membro 2	Joslaine Cristina Jeske de Freitas

Declaro ainda que a versão final anexada a este processo está adequada para ser devidamente depositada em repositório institucional.

Thiago Borges de Oliveira

Professor Orientador

Observação

Esta declaração deve ser assinada pelo(a) orientador(a)

Documento assinado eletronicamente por **THIAGO BORGES DE OLIVEIRA**, **Professor do Magistério Superior**, em 08/04/2024, às 10:22, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no site https://sei.ufj.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0255988** e o código CRC **55F2C502**.

Este trabalho é dedicado à minha família, à minha namorada e aos meus amigos.

Agradecimentos

Gostaria de expressar minha profunda gratidão a Deus, à minha amada família, que mesmo distante, sempre esteve ao meu lado, oferecendo seu apoio incondicional. À minha querida namorada e aos meus leais amigos, que foram fontes constantes de encorajamento e suporte ao longo de todo este processo desafiador, dedico minha mais sincera gratidão. Não posso deixar de mencionar a contribuição inestimável do meu orientador e dos avaliadores, cujas sugestões foram fundamentais para aprimorar este trabalho. Agradeço a todos vocês por fazerem parte desta jornada e por tornarem possível alcançar este momento de realização.

"A persistência é o caminho do êxito."
(Charles Chaplin)

Resumo

A estimativa de custo de consultas espaciais possui um considerável custo computacional, principalmente em situações envolvendo junções e multijunções espaciais, nas quais diversos planos de execução precisam ser considerados. Para a escolha do plano mais eficiente, é comum empregar histogramas espaciais, os quais proporcionam uma estimativa do número de objetos que serão recuperados em cada consulta. No entanto, essa estimativa enfrenta imprecisões devido a três principais fontes de erro: a contagem múltipla de objetos, a aproximação dos objetos mediante o uso do Retângulo Mínimo Envolvente (*MBR*) e a suposição de uniformidade nas equações de cálculo da estimativa. Neste trabalho propõe-se um algoritmo capaz de gerar conjuntos de dados sintéticos, nos quais é possível reproduzir esses erros de forma individual e conjunta e avalia-se a contribuição singular de cada um desses erros na imprecisão das estimativas dos histogramas IHWAF, Euler Melhorado, MinSkew e EulerSkew. A partir dos experimentos realizados, onde foram executadas consultas de janela em *datasets* dos tipos polígono e linha – que é a base das estimativas mais complexas envolvendo junções e multi-junções, observou-se que, independentemente do tipo e extensão do *dataset*, o erro de contagem múltipla de objetos é o principal responsável pela imprecisão na estimativa de seletividade dos histogramas. Este é seguido pelo erro de suposição de uniformidade, que se mostrou mais agravante em *datasets* de menor extensão. Em síntese, a análise dos resultados destaca a importância de considerar esses fatores na melhoria das técnicas de histogramas e, consequentemente, no processo de otimização de consultas espaciais, visando melhorar a eficiência computacional e a precisão dos resultados obtidos.

Palavras-chave: *Histogramas; Erro de Estimativa de Seletividade; Consulta de Janela; Gerador de Datasets.*

Abstract

The cost estimation of spatial queries has a considerable computational cost, especially in situations involving simple and multiway spatial joins, as various execution plans must be considered for these type of queries. To select the most efficient plan, it is common to employ spatial histograms, which provide an estimate of the number of objects that will be retrieved in each query. However, this estimate faces inaccuracies due to three main sources of error: multiple object counting, approximation of objects using the Minimum Bounding Rectangle (*MBR*), and the assumption of uniformity in the equations used for estimating. This work proposes an algorithm capable of generating synthetic datasets that reproduce these errors (jointly and individually) and evaluates the unique contribution of each of them to the inaccuracy of the estimates of the IHWAF, Euler Improved, MinSkew and EulerSkew histograms. We conducted a set of experiments using window queries – which is the basis of the more complex estimates involving joins and multi-joins, on the generated datasets and observed that, regardless of the type (line or polygon) and extension of the *dataset*, the multiple object counting error is mainly responsible for the inaccuracy in the histogram selectivity estimation. This is followed by the uniformity assumption error, which is more aggravating in smaller *datasets*. In summary, the analysis of the results highlights the importance of considering these factors in improving histogram techniques and, consequently, the process of optimizing spatial queries, improving computational efficiency and the assertivity of the results obtained.

Keywords: *Histograms; Selectivity Estimation Error; Window Query; Dataset Generator.*

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

MBR	Retângulo Mínimo Envolvente
IHWAF	Histograma de Grade Melhorado
SIG	Sistemas de Informação Geográfica
SBDE	Sistema de Banco de Dados Espaciais
BOX	Retângulo Qualquer

Sumário

1 Introdução

1.1 Motivação

Ao longo dos anos, o avanço da tecnologia ocasionou a coleta de uma grande quantidade de dados espaciais, que incluem dados biológicos e científicos, mapas meteorológicos, dados agrícolas, dados socioeconômicos e mídias sociais, todos eles georreferenciados. Estes dados são capturados usando dispositivos e serviços móveis com reconhecimento de localização, como, por exemplo, *smartphones*, *tablets*, *wearables*, dispositivos GPS (*Global Positioning System*), satélites, além de outros. O volume de dados espaciais gerado todos os dias aumenta a uma taxa impressionante, assim como o número de aplicativos e domínios onde esses dados são coletados e analisados (BOUROS; MAMOULIS, 2019). Os Sistemas de Informação Geográfica (SIG), ou (*Geographical Information System*), são empregados para processar este volume de dados, nas etapas de coleta, armazenamento, recuperação, transformação e visualização (FITZ, 2018).

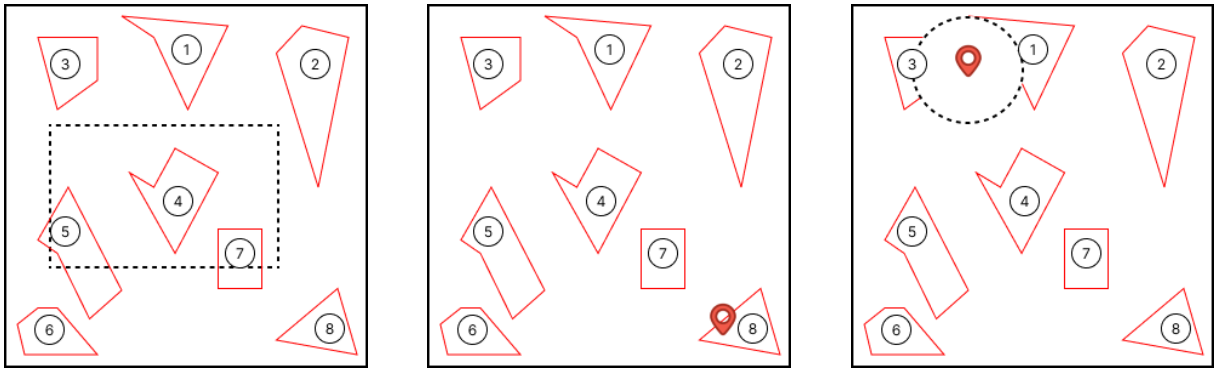
Um Sistema de Banco de Dados Espaciais (SBDE), ou (*Spatial Database System*) é uma categoria de SIG que possui a capacidade de lidar com conjuntos de dados em um contexto multidimensional. O SBDE possibilita a manipulação eficiente desses dados, permitindo a realização de consultas espaciais para a extração de informações significativas. Essas consultas envolvem uma variedade de abordagens e técnicas, todas voltadas para analisar dados em um espaço que leva em consideração suas dimensões espaciais (ELMASRI; NAVATHE, 2010). Algumas dessas consultas são:

1. Consulta de janela: é uma operação de seleção geométrica que é aplicada a um conjunto de dados único. O resultado dessa consulta consiste em um conjunto de objetos que estão contidos dentro de uma região específica, frequentemente representada como uma janela retangular, como demostrado na Figura 1a (OLIVEIRA, 2017);

2. Consulta de ponto: onde procura-se localizar um determinado objeto, cuja localização espacial se estabeleça nas coordenadas correspondentes a posição especificada pelo ponto, como representada na Figura 1b; e

3. Consulta de vizinhança: consiste em fazer uma busca em um raio a partir de um ponto predeterminado; a consulta por vizinhança retornará todos os objetos dentro do círculo, como apresentado na Figura 1c.

Além das consultas acima, outra importante consulta nos bancos de dados espaciais é a junção espacial (*spatial join*). Ela possui aplicação em diversas áreas do conhecimento, como por exemplo, em aplicações científicas voltadas à análise e interpretação de imagens



(a) Consulta de Janela, retorna os objetos 4, 5 e 7.

(b) Consulta de Ponto, retorna o objeto 8.

(c) Consulta de Vizinhança, retorna os objetos 1 e 3.

Figura 1 – Exemplos de consultas espaciais.

médicas de alta resolução (ressonância magnética ou tomografia computadorizada). Nelas, as junções espaciais são usadas para determinar a proximidade de elementos (conjuntos de células ou tecido humano) que apoiam o diagnóstico, previsão e tratamento mais eficazes de doenças. A Figura 2 exemplifica uma aplicação dessa consulta sobrepondo duas tomografias com intuito de analisar a área comum entre polígonos verdes e vermelhos. Os polígonos verdes foram identificados em uma tomografia e os vermelhos em outra (dois exames de um mesmo paciente). Comparando os polígonos colocalizados é possível acompanhar o desenvolvimento ou retrocesso de uma determinada doença. Ou seja, se os polígonos verdes são recorrentes da primeira tomografia e os vermelhos da segunda, pode-se concluir se uma determinada doença está se agravando, já que, novos polígonos vermelhos surgiram em relação a primeira tomografia.

Uma junção espacial com três ou mais *datasets* é denominada de multijunção espacial (*multiway spatial join*) (MAMOULIS; PAPADIAS, 2001). Cada instância de consulta de multijunção espacial, em particular, pode ser executada de diversas formas, conhecidas como planos de execução. Todos os planos para uma consulta são equivalentes semanticamente, ou seja, retornam o mesmo conjunto de respostas como resultado. Porém, diferem entre si em relação ao uso de recursos computacionais como, por exemplo, tempo de processamento e tráfego de rede. É importante, portanto, escolher um plano de execução adequado para cada consulta com intuito de obter as respostas rapidamente e economizar recursos computacionais. Essa escolha é feita atualmente através de estimativa de custo computacional, frequentemente usando histogramas espaciais (OLIVEIRA, 2017).

Os histogramas espaciais são estruturas de dados que têm como função a simplificação dos *datasets*, podendo dividir o espaço que os dados ocupam, usualmente, em uma grade que contenha diversas células ou *buckets*. Estes *buckets* podem ser de tamanhos fixos ou variados, dependendo da estratégia adotada no histograma. Para cada célula ou *bucket*, são armazenadas metadados a respeito dos objetos espaciais contidos neles, como a quantidade de objetos (cardinalidade) e o tamanho dos objetos (quantidade de

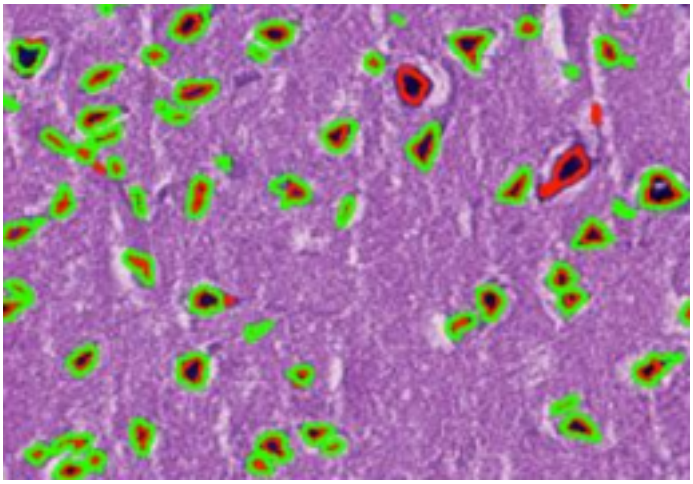


Figura 2 – Junção espacial em duas tomografias sobrepostas, para determinar avanço ou retrocesso de uma determinada doença (verde versus vermelho) (AJI; WANG; SALTZ, 2012).

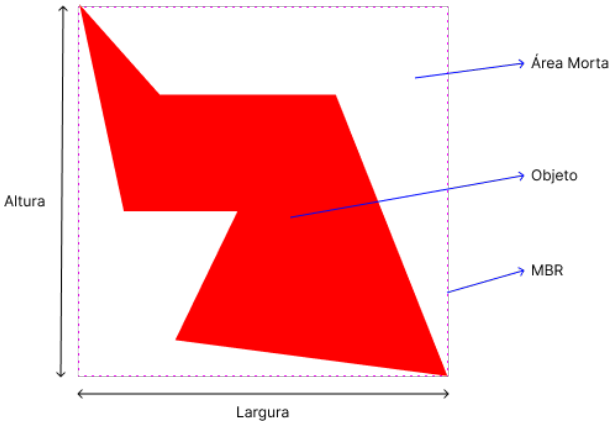
pontos) (OLIVEIRA; COSTA; RODRIGUES, 2017). Ao se pensar em processamento distribuído, o histograma é usado como método de distribuição e acesso, com isso, cada servidor do cluster armazenará uma ou mais células, formando assim, um índice distribuído (OLIVEIRA; COSTA; RODRIGUES, 2015).

Um histograma espacial, por ser um resumo dos dados sobre os quais foi construído, não representa fielmente os mesmos e o custo computacional estimado com base nesse resumo apresenta imprecisões. Três problemas principais causam erro de estimativa nos histogramas espaciais existentes:

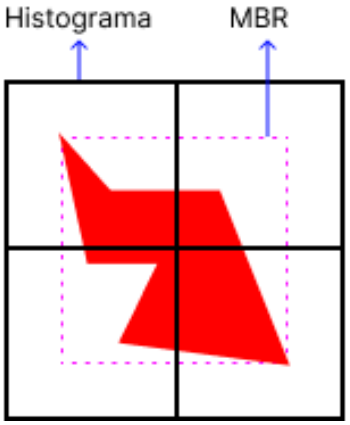
1. o erro de aproximação dos objetos espaciais, devido o uso de Retângulo Mínimo Envolvente (*MBR*) (LIU; YUAN; LIN, 2003). A Figura 3a exemplifica a aproximação do objeto usando MBR, que resulta em uma área morta que por sua vez deveria ser desconsiderada, porém, isso não acontece; logo, qualquer estimativa de consulta que intersectar a área morta do MBR pode considerar, erroneamente, o objeto;

2. o erro de contagem múltipla, quando os objetos com extensão espacial (polígonos e linhas) se sobrepõem em mais de um *bucket*, ou célula do histograma (SUN; AGRAWAL; ABBADI, 2002). A Figura 3b mostra um objeto que sobrepõe mais de uma célula de um histograma de grade convencional, logo, será contado múltiplas vezes; neste exemplo e mais especificamente, quatro vezes; e

3. o erro causado pela suposição de uniformidade nas equações que calculam a estimativa de custo das consultas (ACHARYA; POOSALA; RAMASWAMY, 1999). A equação supõe que os objetos estão distribuídos uniformemente pelo *dataset*, entretanto, isso dificilmente acontece, logo, há uma inconsistência na estimativa.



(a) Representação de um MBR.



(b) Representação de um objeto que será contado múltiplas vezes.

Figura 3 – Amostragem dos erros presentes nos histogramas.

Hoje, nenhum estudo do qual temos ciência investigou a contribuição isolada de cada tipo de erro citado acima na imprecisão da estimativa de seletividade das consultas espaciais. A partir dessa caracterização, ou seja, da identificação do quanto cada tipo de erro interfere na precisão da estimativa dos histogramas, acreditamos ser possível indicar direções de aperfeiçoamento para as técnicas existentes de histogramas espaciais.

1.2 Objetivo do Trabalho

O objetivo deste trabalho foi identificar e caracterizar a contribuição individual de cada um dos três tipos de erro – citados nos parágrafos acima, na imprecisão das estimativas para consultas espaciais mensuradas para os Histogramas de Euler Melhorado, MinSkew, EulerSkew e IHWAF. Parte deste trabalho foi realizado durante a vigência dos projetos PI0276-2015 e PI02366-2018 no ICET-UFJ, ambos relacionados ao desenvolvimento e integração de estruturas de dados e algoritmos para processamento eficiente da multijunção espacial em sistemas distribuídos. Neles, implementações referência de alguns histogramas espaciais foram elaboradas e outros histogramas foram propostos, melhorados ou adaptados para sistemas distribuídos. Uma descrição compreensível desses histogramas será apresentada no próximo capítulo. Os objetivos específicos são:

1. Estudar os histogramas para entendimento das características comuns e diferenças entre eles;

2. Codificar os algoritmos dos experimentos, integrando-os ao código existente para os histogramas no repositório do projeto (github.com/thborges/dgeohistogram);

3. Elaborar os *datasets* necessários para os experimentos;

4. Executar experimentos para avaliar a contribuição individual de cada problema na estimativa dos histogramas; e

5. Analisar os resultados obtidos a fim de apresentar uma conclusão.

1.3 Referencial Teórico Resumido

Esta seção aborda um breve resumo dos conceitos que serão aprofundados no Capítulo 2, são eles:

1. Dados Espaciais: Os dados espaciais referem-se a elementos do mundo real com relevância geográfica, como vias públicas, estruturas construídas e massas de água, cada um associado a coordenadas específicas. Além das informações de localização, esses elementos possuem características específicas, conhecidas como atributos, como nome, altura, profundidade ou população (CAMPBELL, 2012).

2. Junção Espacial: A junção espacial, é um tipo de consulta espacial que consiste na interseção de duas ou mais bases de dados georreferenciadas (OLIVEIRA, 2017).

3. Multijunção Espacial: A realização da junção espacial envolvendo três ou mais *datasets* transforma a operação em uma multijunção espacial. Diversos planos de execução, conforme descritos por Oliveira (2017), podem ser utilizados para realizar essa consulta.

4. Estimativa de Seletividade de Consultas Espaciais: Uma métrica crucial para estimar o custo de execução em consultas espaciais foi proposta por Mamoulis e Papadias (2001), visando aprimorar essa estimativa, sugeriram incorporar o comprimento médio dos objetos como metadados em cada célula do histograma, relacionando-o ao MBR de todos os objetos contidos nessas células.

5. Histogramas Espaciais: Histogramas espaciais são estruturas de dados empregadas na síntese de *datasets* por meio de uma técnica que subdivida sua extensão espacial. Essa subdivisão ocorre através de células ou *buckets*, dependendo da construção utilizada. Uma característica comum à maioria dos histogramas é a utilização do MBR para envolver os objetos presentes nos *datasets*.

6. Histograma de Euler: O Histograma de Euler, proposto por Sun, Agrawal e Abbadi (2002), emprega uma grade para particionar a extensão do *dataset* em células, visando abordar o desafio da contagem múltipla de objetos. Contudo, ainda apresenta desafios de seletividade devido à aproximação dos objetos espaciais usando MBR e à sua própria fórmula de estimativa, que assume uniformidade.

7. Histograma MinSkew: O Histograma MinSkew, derivado do conceito de densidade espacial mínima, difere da abordagem convencional de utilizar uma estrutura de grade para simplificar o *dataset*. Em vez disso, ele divide a extensão do *dataset* em retângulos. A principal finalidade dessa técnica é realizar a divisão de forma a agrupar uma quantidade uniforme de objetos em cada retângulo, estabelecendo assim a densidade mínima do *dataset* (ACHARYA; POOSALA; RAMASWAMY, 1999).

8. Histograma EulerSkew: Trata-se de um histograma experimental proposto no âmbito do projeto de pesquisa no qual este trabalho também está inserido (OLIVEIRA, 2018), que combina técnicas dos Histogramas de Euler e MinSkew para aproveitar as vantagens de ambos. O processo de criação ocorre em duas etapas: a primeira envolve uma partição semelhante ao MinSkew, com foco na distribuição uniforme de objetos, enquanto a segunda se assemelha ao método do Histograma de Euler para lidar com o erro de contagem múltipla de objetos.

9. Histograma IHWAF: O Histograma IHWAF, projetado para processamento distribuído, armazena três métricas em cada célula: a cardinalidade, o tamanho combinado dos objetos e a localização da célula no *cluster*. Além disso, a contagem é proporcional ao MBR dos objetos espaciais que se sobrepõem a cada célula. Essa proporcionalidade é alcançada por meio do método de Sobreposição Proporcional, que, ao contrário dos histogramas de grade convencionais, utiliza frações do MBR em vez de apenas o centro para determinar a inclusão do objeto (OLIVEIRA, 2017).

10. Erro de Estimativa: Conforme mencionado por Vuolo (1996), o conceito de erro de estimativa normalmente se relaciona à discrepância entre o valor estimado de uma quantidade e o seu valor real. Em ambientes de medição e análise de dados, esse erro pode ser originado por vários fatores, incluindo erros estatísticos ou aleatórios e sistemáticos.

11. DGEO: O sistema DGEO (OLIVEIRA et al., 2023) é uma aplicação destinada ao processamento distribuído de dados espaciais e implementa versões referência dos histogramas empregados neste trabalho. Desenvolvido inicialmente em Oliveira (2017), continua sendo aprimorado como parte do projeto de pesquisa Oliveira (2018) e sua implementação é escrita nas linguagens de programação C e Go.

1.4 Contribuição do Trabalho

As principais contribuições desde trabalho são descritas a seguir:

1. Implementação e construção de um gerador de *datasets* sintéticos capaz que provocar individualmente os erros de aproximação de objetos, contagem múltipla de objetos e suposição de uniformidade, além de permitir escolher a quantidade de objetos do *dataset*, o tamanho da dimensão do mesmo e se o *dataset* será do tipo polígono ou linha.

2. Caracterização individual dos três principais erros que degradam a estimativa de seletividade em consultas espaciais nos Histogramas de Euler Melhorado, MinSkew, EulerSkew e IHWAF, possibilitando assim, novas melhorias futuras.

3. Apresentar o comportamento dos erros em *datasets* de diferentes tipos (polígono e linha).

1.5 Organização da Monografia

O trabalho está dividido em seis capítulos, descritos resumidamente a seguir: O Capítulo 2 apresenta os conceitos técnicos necessários para a compreensão deste trabalho, como por exemplo, a descrição de dados espaciais e histogramas espaciais. Já o Capítulo 3 contempla os trabalhos que são relacionados a essa pesquisa e que também motivaram a mesma. No Capítulo 4 é apresentado um algoritmo responsável por gerar os *datasets* sintéticos que foram utilizados nos experimentos. O Capítulo 5 discute a metodologia experimental, ou seja, como os experimentos foram pensados e executados, também, aborda os resultados dos experimentos e as métricas empregadas para averiguar a assertividade da estimativa. Por último, o Capítulo 6 apresenta uma conclusão da pesquisa e como os resultados dos experimentos respondem a hipótese deste estudo, além de apontar trabalhos futuros.

2 Referencial Teórico

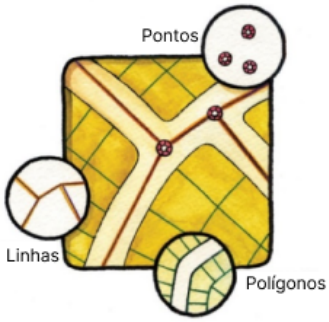
2.1 Dados Espaciais

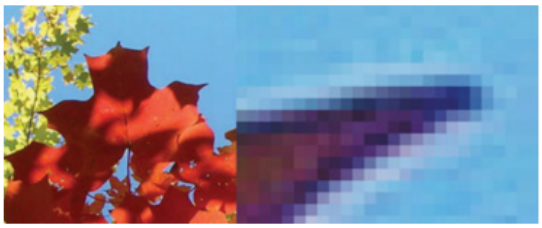
Os dados espaciais fazem referência aos elementos do mundo real que despertam interesse geográfico, tais como vias públicas, estruturas construídas, massas de água e nações, cada um deles associado às respetivas coordenadas. Além da informação relativa à localização, cada um destes elementos possui características específicas, também conhecidas como atributos, que incluem informações como nome, altura, profundidade ou população (CAMPBELL, 2012). Para ilustrar este conceito, podemos considerar o exemplo da tomografia mencionado na Introdução Figura 2. Neste contexto, as imagens resultantes deste processo contêm dados espaciais. A tomografia permite a detecção de variações em órgãos, tecidos e ossos ao processar as imagens e representar partes do corpo como dados espaciais. Neste cenário, os dados consistem numa imagem que retrata a composição dos tecidos e possíveis anomalias, juntamente com um atributo que identifica a localização da anomalia no corpo da pessoa sujeita à análise, assim ajudando no diagnóstico.

A localização espacial é determinada por dados matriciais Figura 4b ou vetoriais Figura 4a, onde os três principais tipos de dados vetoriais são os pontos, linhas e polígonos. Os pontos são caracterizados por conterem um par de coordenadas que mostram sua localização, eles também podem ter outras informações escalares associadas, como nome do local, tipo, etc. Geralmente, utilizamos pontos para representar elementos singulares e discretos, como edifícios, poços, postes de energia, locais de amostragem, entre outros. Quando conectamos dois ou mais desses pontos, criamos linhas ou segmentos de linhas. Se esses pontos são interligados em uma sequência contínua, onde o primeiro ponto se conecta ao último, é possível formar um polígono. Essa representação permite a descrição da silhueta de elementos geográficos e acontecimentos (CAMPBELL, 2012).

Dados espaciais são manipulados em SIGs, eles oferecem uma variedade de técnicas para lidar com informações espaciais ou georreferenciadas, abrangendo processos como a aquisição, preparação, gerenciamento, armazenamento, manutenção, manipulação, análise e, por fim, a apresentação desses dados. O termo “dados espaciais” refere-se a qualquer informação que contenha uma marcação de localização associada a ela. Essa marcação por exemplo, pode ser relacionada ao nosso planeta Terra, ou a outros referenciais, como o espaço cósmico, ou informações sobre o corpo humano obtidas por meio de imagens médicas (FITZ, 2018).

Um SBDE é uma categoria de SIG que inclui funcionalidades projetadas para gerenciar bancos de dados que controlam objetos em um espaço multidimensional. Por exemplo, um SBDE pode ser empregado para administrar bancos de dados cartográficos





(a) Vetores de dados ponto, linha e polígono. (b) Tipo de dados matriciais.

Figura 4 – Exemplos de dados espaciais (CAMPBELL, 2012).

que contêm mapas contendo descrições multidimensionais de elementos como rios, cidades, estradas, mares, e assim por diante (ELMASRI; NAVATHE, 2010). Para executar essa administração, um SBDE armazena os dados de maneira estruturada e aplica algoritmos espaciais para fornecer respostas aos usuários em relação a consultas espaciais, consultas que foram apresentadas na Introdução e exemplificadas na Figura 1.

2.2 Junção Espacial

A junção espacial (*spatial join*) é uma forma de consulta espacial que envolve a intersecção de duas ou mais bases de dados georreferenciadas com base em um predicado espacial θ (OLIVEIRA, 2017). Exemplos de predicados espaciais são (CAMPBELL, 2012):

1. Intersecção: identifica todas as características na camada alvo que se sobrepõem à camada de origem;
2. Completamente contido: recupera objetos localizados inteiramente dentro da camada de origem; e
3. São idênticos: localiza objetos que compartilham a mesma localização geográfica.

Formalmente, uma (θ-junção) pode ser definida da seguinte maneira: sejam A e B dois *datasets* distintos de objetos multidimensionais, então se existem objetos a $\in$ A e b $\in$ B que satisfazem o predicado espacial θ, é possível realizar uma θ-junção entre A e B denotada por:

$$A \bowtie B = \{(a, b) | a \in A, b \in B \text{ e } a\theta b\}$$

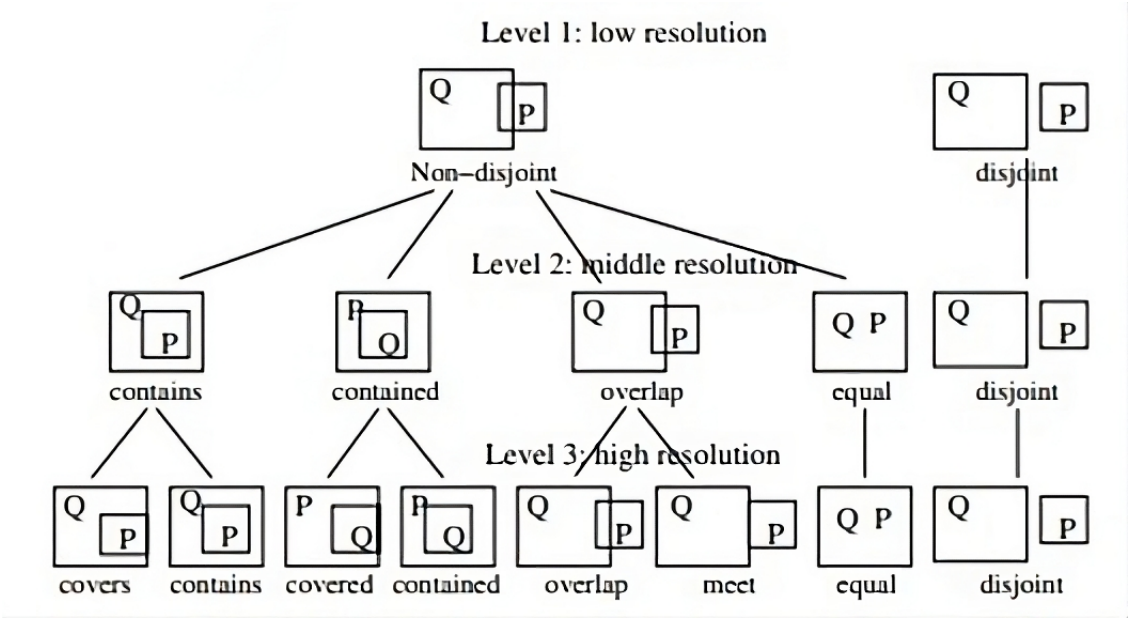


Figura 5 – Topologia das relações espaciais existentes entre objetos espaciais. O nível 1 da topologia é de baixa resolução e possui as relações: Não disjunta, disjunta. O nível 2 é de média resolução e possui as relações: contém, contido, sobreposto, igual, disjunto. O nível 3 é de alta resolução e contém as relações: cobre, contém, coberto, contido, sobreposto, encontra, igual e disjunto (LIU; LIN; YUAN, 2005).

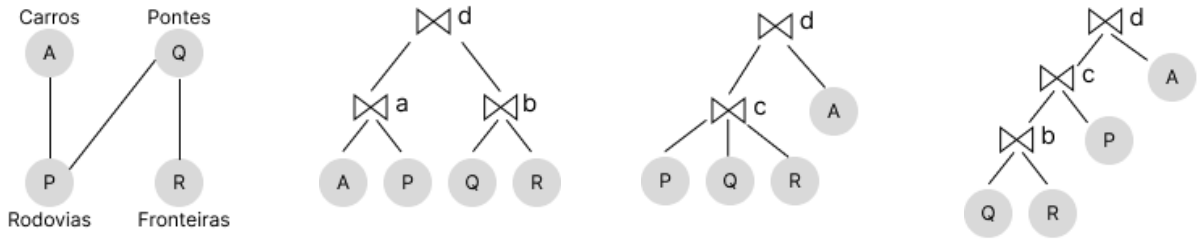
A Figura 5 demonstra que as relações espaciais geralmente se dividem em três categorias, cada uma representando um nível de resolução. Quanto menor a resolução, menos relações são encontradas nesse nível. Por outro lado, o nível de alta resolução contém a maioria das relações. Nota-se que P e Q, podem ser considerados de três maneiras diferentes: quando um objeto inclui o outro, quando se sobrepõem sem inclusão e quando os limites dos objetos não se tocam.

Vale destacar, que para um banco de dados é importante suportar todos os tipos de relações, entretanto, devido ao interesse na precisão da estimativa de seletividade, este trabalho se concentrou apenas no nível 1, frequentemente o mais usado nas consultas espaciais (OLIVEIRA, 2017).

2.3 Multijunção Espacial

Realizar a junção espacial com três ou mais *datasets* torna a operação em uma multijunção espacial (*multiway spatial join*). Existem várias maneiras equivalentes de executar essa consulta, chamadas de planos de execução (OLIVEIRA, 2017). Um exemplo desse tipo de operação é: "identificar quais os modelos de carros que trafegam por rodovias federais e passam por pontes que delimitam fronteiras estaduais". Essa situação exige a junção de quatro *datasets*, que são: Carros, Rodovias, Pontes e Fronteiras. Como mostra a

Figura 6.



(a) Grafo. (b) Plano 1. (c) Plano 2. (d) Plano 3.

Figura 6 – Exemplos de planos de execução para uma consulta espacial. Em (b), (c) e (d) temos planos de execução para a consulta em (a). A consulta é representada por um grafo, sendo cada vértice um *dataset* e cada aresta um predicado da junção entre os mesmos.

Na Figura 6b, é feita a junção de dois pares de *datasets* (A com P) e (Q com R) para formar dois resultados intermediários ($\bowtie^a$) e ($\bowtie^b$), que por fim são combinados gerando o resultado final ($\bowtie^d$). O plano 2, na Figura 6c, faz a junção de três *datasets* simultaneamente (P, Q e R), gerando um resultado intermediário ($\bowtie^c$), ao combinar ele com o *dataset* (A), resultará em ($\bowtie^d$). Por último, a Figura 6d, ($\bowtie^b$) é resultado intermediário da junção entre os *datasets* (Q e R), este resultado é combinado com o *dataset* (P), gerando outro resultado intermediário ($\bowtie^c$), que será combinado com o *dataset* (A) e derivará o resultado final ($\bowtie^d$).

Apesar dos diferentes planos de execução, os mesmos são equivalentes – por definição, e respeitam os predicados das junções individuais da consulta (p. ex., interseção dos objetos espaciais entre os *datasets* ligados com arestas na Figura 6a). Desta forma, o resultado final da consulta é o mesmo. O que muda é o custo computacional de cada plano e, portanto, surge a importância de escolher o melhor plano de execução.

2.4 Estimativa de Seletividade de Consultas Espaciais

Uma das principais métricas usadas para estimar o custo de execução é a estimativa de seletividade das consultas espaciais. Para melhorar tal estimativa, Mamoulis e Papadias (2001) propuseram a utilização do comprimento médio dos objetos, adicionando-os como metadados em cada célula do histograma. O comprimento médio é relacionado ao MBR de todos os objetos contidos nas células dos histogramas. Logo, para estimar a cardinalidade de uma consulta $O^{\overline{w}}$ é usada a Equação 2.1.

$$O^{\overline{w}}(a, \overline{w}) = \overline{a} * \prod_{k=1}^{d} \min\left(1, \frac{l_{ak} + l_{a\overline{w}}}{l_{uk}}\right) \tag{2.1}$$

Esta equação retorna a cardinalidade de $O^{\overline{w}}$ na consulta da janela $\overline{w}$. Definimos $\overline{a}$

como a cardinalidade do conjunto de dados a. d representa o número total de dimensões e k varia por todas as dimensões. Para representar o comprimento médio dos objetos em a na dimensão k, utilizamos a notação l_{ak}. O comprimento de $\overline{w}$ é representado por $l_{a\overline{w}}$, enquanto l_{uk} denota o comprimento total da dimensão k (universo), com a restrição de que $l_{uk} \neq 0$. O resultado da equação representa a estimativa da quantidade de objetos que satisfazem o predicado espacial, considerando a interseção da consulta $\overline{w}$ com os objetos em a.

Porém, ao utilizar esta equação, surge o erro causado pela suposição de uniformidade dos objetos dentro da célula (ACHARYA; POOSALA; RAMASWAMY, 1999). A fórmula pressupõe que os objetos estão distribuídos de maneira uniforme dentro da célula, entretanto em *datasets* reais isso raramente acontece. Veja a ilustração na Figura 7.

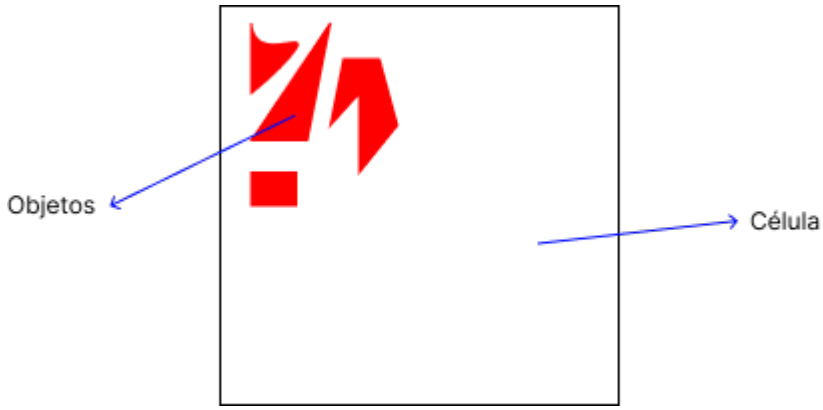


Figura 7 – Objetos não uniformemente distribuídos, concentrados no canto superior esquerdo da célula.

Com isso, na Figura 7, se a consulta de janela $\overline{w}$ fosse elaborada de maneira tal a selecionar objetos apenas na parte inferior direita dessa célula de histograma, a fórmula estimaria a existência de um conjunto x de objetos, quando na verdade, nenhum objeto será retornado na consulta real aplicada sobre o *dataset*.

2.5 Histogramas Espaciais

Histogramas espaciais são estruturas de dados usadas para resumir *datasets* através de uma abordagem que divide a extensão espacial do mesmo. Essa divisão é feita usando células ou *buckets*, dependendo de sua construção. Os Histograma de Euler e o Histograma MinSkew, são exemplos dessas divisões e serão abordados nos próximos tópicos. Uma característica comum deles é que ambos usam o MBR para envolver os objetos dos *datasets*, como mostra a Figura 3a.

Entretanto, a utilização do MBR causa o erro de aproximação dos objetos espaciais (LIU; YUAN; LIN, 2003), já que, toda a área do MBR é considerada como objeto, quando,

na verdade, a área morta, exemplificada na Figura 3a, deveria ser desconsiderada, pois, uma consulta de janela qualquer, que intersecta a área morta, retornará o objeto contido dentro do MBR. Além disso, essa aproximação também pode ocasionar no erro de contagem múltipla de objetos, já que, a área morta do MBR pode sobrepor mais de uma célula, logo, o objeto será contado duas ou mais vezes.

2.5.1 Histograma de Euler

O Histograma de Euler usa uma grade para dividir a extensão do *dataset* em células, representado na Figura 8 e foi proposto por Sun, Agrawal e Abbadi (2002) com o intuito de tentar resolver o problema de contagem múltipla de objetos. Porém, o mesmo ainda apresenta erros de seletividade devido a aproximação dos objetos espaciais utilizando MBR e pela sua própria fórmula de estimativa, que supõe uniformidade de forma similar a Equação 2.1. Ele é fundamentado na teoria dos grafos de Euler e, diferentemente de um Histograma de Grade convencional, aloca *buckets* para as faces, cantos e para os contornos das células. A Figura 9 apresenta uma comparação entre eles.

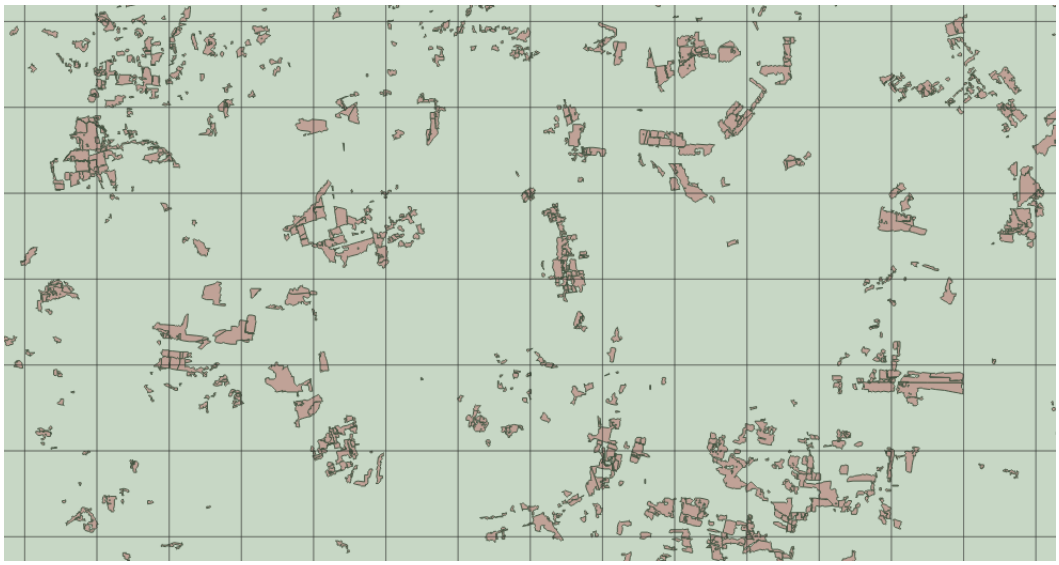
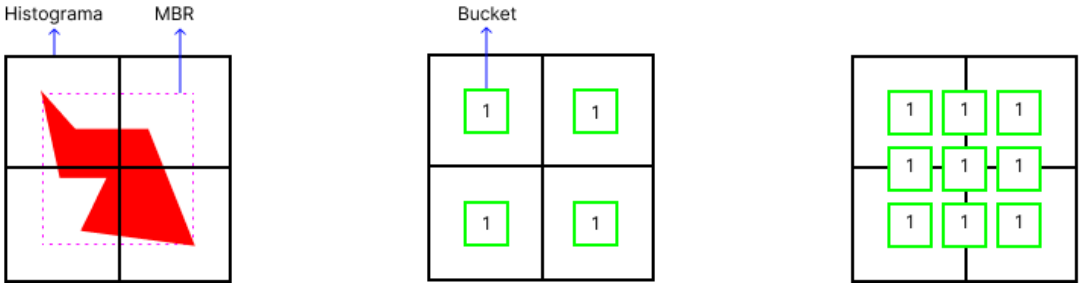


Figura 8 – Fragmento de um Histograma de Euler aplicado em um *dataset* de alerta de desmatamento.

Nota-se que na Figura 9a há um objeto contido em um MBR que sobrepõe quatro células de um Histograma de Grade. Em um Histograma de Grade convencional (b), o histograma aloca um *bucket* para cada célula, assim, contando o mesmo objeto quatro vezes, já o Histograma de Euler (c), aloca *buckets* também para as faces das células, cantos e contornos, logo, é possível mensurar o tamanho do objeto, contando ele uma única vez.

O Histograma de Euler possui esse nome graças a conexão estabelecida pelo renomado matemático suíço Leonhard Euler que é de suma importância para determinar



(a) MBR em várias células. (b) Histograma de grade. (c) Histograma de Euler.

Figura 9 – Comparação entre um Histograma de Grade (b) e o Histograma de Euler (c). O Histograma de Grade (b) conta o objeto em (a) várias vezes, em cada célula que o mesmo sobrepõe. O Histograma de Euler (c) conta o objeto em todas as arestas, vértices e faces, e consegue detectar a repetição.

o número de arestas, vértices e faces de todos os poliedros convexos, bem como de certos poliedros não convexos. Como resultado dessa associação, torna-se possível calcular a quantidade de elementos presentes em um poliedro, relação essa, que foi estabelecida pela a Equação 2.2.

$$V - A + F = 2 \tag{2.2}$$

Na equação a cima, V representa o número de vértices, A o número de arestas e F o número de faces.

2.5.2 Histograma MinSkew

O Histograma MinSkew é uma abreviação do termo densidade espacial mínima. Ao invés de utilizar uma estrutura de grade para simplificar o *dataset*, o mesmo divide a extensão do *dataset* em retângulos conforme exemplificado na Figura 10. O principal objetivo dessa técnica é agrupar a mesma quantidade de objetos em cada retângulo, de forma que esta quantidade se tornará a densidade mínima do *dataset*. Com isso, o problema de suposição de uniformidade na Equação 2.1 teoricamente seria resolvido. Contudo, se a densidade espacial inicial no *dataset* contiver uma grande variação, serão criados muitos *buckets* e a precisão do histograma se perde parcialmente. Além disso, ele não possui tratamento específico para a contagem múltipla de objetos e para a aproximação feita pelo MBR (ACHARYA; POOSALA; RAMASWAMY, 1999).

Analisando a Figura 10, nota-se que não há objetos na parte superior esquerda. No inicio da geração da estrutura do histograma, ela começa apenas como um retângulo, além de um histograma de grade auxiliar. Posteriormente é feita a contagem dos objetos presentes dentro do retângulo: se a densidade espacial calculada no histograma de grade

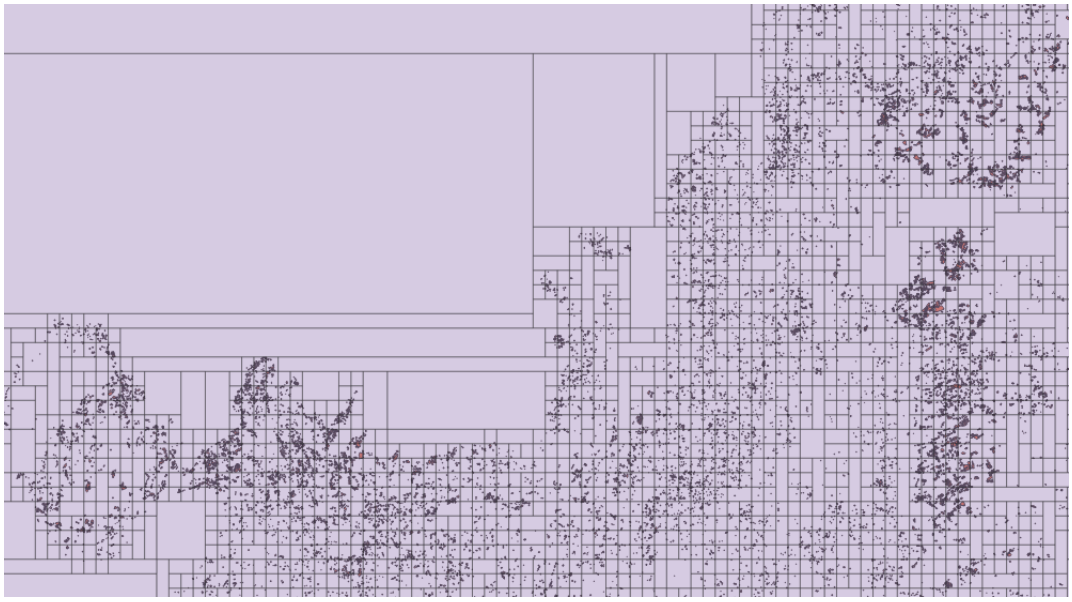


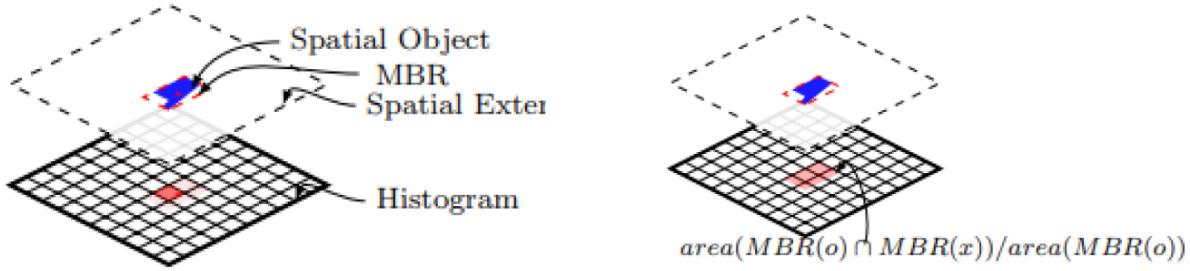
Figura 10 – Fragmento de um Histograma MinSkew aplicado em um *dataset* de alerta de desmatamento.

não for uniforme, o retângulo será subdividido em dois retângulos adicionais, de modo que cada um deles apresente uma densidade mais uniforme individualmente. Esse processo é repetido recursivamente até que o resultado da contagem demonstre uniformidade ou até atingir um número prévio de *buckets* estabelecido. Ao final, o histograma terá a aparência de um retângulo preenchido por vários outros retângulos. No caso da figura apresentada, a densidade de retângulos se concentram na parte inferior e direita do histograma.

2.5.3 Histograma EulerSkew

Os parágrafos acima mostraram que os Histogramas de Euler e MinSkew tem vantagens e desvantagens um sobre o outro. Com isso, foi proposta a criação de um histograma híbrido, chamado EulerSkew, que implementa técnicas dos dois histogramas simultaneamente, com o objetivo de agregar as vantagens de cada um no novo histograma (SOUSA, 2022).

A criação do Histograma EulerSkew se dá em duas etapas: a primeira é uma partição semelhante aos métodos do MinSkew, Figura 10, com algumas adaptações. O foco dessa partição é a distribuição uniforme dos objetos pelo histograma; já a segunda etapa é semelhante ao método utilizado no Histograma de Euler para tratar o erro de contagem múltipla de objetos, como mostra a Figura 9. Essa segunda etapa tem o objetivo de garantir uma estimativa segura em relação ao erro mencionado.



(a) Antes do método. (b) Depois do método.

Figura 11 – Método de Sobreposição Proporcional (*Proportional Overlap*) (OLIVEIRA, 2017).

2.5.4 Histograma IHWAF

O Histograma IHWAF é um histograma de grade pensado para processamento distribuído onde a célula do histograma armazena três métricas: 1) o número de objetos espaciais dentro dos limites da célula (a cardinalidade da célula), 2) o tamanho combinado dos objetos na célula (em termos do total de pontos que eles contêm) e, 3) o local (servidor no *cluster*) onde a célula está localizada. Além disso, a contagem é proporcional ao MBR dos objetos espacias que sobrepõe cada célula. Essa transformação é feita através de um método chamado Sobreposição Proporcional, que ao contrário de um histograma de grade convencional, onde é utilizado apenas o centro do MBR para decidir onde o objeto está contido, o método utilizada frações do MBR, como mostra a Figura 11. O cálculo da primeira métrica é feito através de um somatório de todos os objetos contidos dentro da célula que é determinada através da equação presente na figura mencionada, onde, $area(MBR(o) \cap MBR(x))$ representa a área de interseção entre o MBR do objeto espacial e os limites da célula do histograma e $area(MBR(o))$ representa a área do MBR (OLIVEIRA, 2017). Após todo esse processo o Histograma IHWAF ficará com a estrutura presente na Figura 12.

Figura 12 – Fragmento do Histograma IHWAF aplicado em um *dataset* de alerta de desmatamento.

2.6 Erro de Estimativa

De acordo com Vuolo (1996) o conceito de erro de estimativa geralmente se refere à diferença entre o valor estimado de uma quantidade e o valor verdadeiro dessa quantidade. Em contextos de medição e análise de dados, o erro de estimativa pode ser causado por diversos fatores, como erros estatísticos ou erros aleatórios e erros sistemáticos.

Erro estatístico ou erro aleatório refere-se à medida da dispersão dos resultados em torno do valor verdadeiro. Resulta de variações aleatórias nas medições, provenientes de fatores que não podem ser controlados ou que não foram controlados.

Erro sistemático é a diferença entre o valor médio verdadeiro e o valor verdadeiro. Pode ser de diferentes tipos, como erro sistemático instrumental (resultante da calibração do instrumento de medição), erro sistemático ambiental (devido a efeitos do ambiente sobre a experiência) e erro sistemático observacional (devido a falhas de procedimentos ou limitações do observador).

Neste trabalho, o erro mensurado é sistemático e resultante da simplificação dos datasets nas técnicas de construção dos histogramas; é um erro estatístico inerente à técnica/algoritmo do histograma, não sendo, portanto, instrumental, ambiental ou observacional.

2.7 DGEO

O sistema DGEO (OLIVEIRA; COSTA; RODRIGUES, 2017; OLIVEIRA et al., 2023) é uma aplicação voltada para o processamento distribuído de dados espaciais e implementa versões referência dos histogramas empregados neste trabalho. Desenvolvido inicialmente em Oliveira (2017), continua sendo aprimorado como parte do projeto de pesquisa Oliveira (2018) e sua implementação é escrita nas linguagens de programação C e Go. Essa aplicação faz uso da biblioteca GEOS (*Geometry Engine - Open Source*) e permite realizar consultas de janela e junção espacial de maneira paralela e distribuída. Esta suíte de aplicação foi utilizada para realizar os experimentos neste trabalho.

3 Trabalhos relacionados

3.1 Critérios de busca

A busca por trabalhos relacionados para este estudo foi conduzida através do *Google Scholar*, um mecanismo de busca que indexa artigos de fontes renomadas, como o *IEEE Explore*, a *ACM Digital Library* e *ACM Computing Reviews*. O processo de busca envolveu a aplicação da *string* de busca (*"Estimation Error"+"Spatial Queries"*) nas bases referenciadas mencionadas anteriormente, onde foram retornados um total de 181 trabalhos. Além disso, para assegurar a abrangência da pesquisa, foi feita uma análise manual das referências dos artigos encontrados com o intuito de localizar publicações que não foram recuperadas pelo mecanismo de busca.

3.2 Metodologia de análise

Cada trabalho encontrado durante essa investigação passou por uma análise de seu título e resumo com a intenção de filtrar apenas os estudos que realmente abordavam o tema desta monografia. Depois disso, os trabalhos que passaram por essa seleção foram submetido a critérios específicos para determinar sua relevância para os objetivos desta pesquisa. Os critérios estão descritos a seguir:

3.2.1 Estimativa de Seletividade na Consulta de Janela (C1)

O primeiro critério referiu-se aos trabalhos que propuseram procedimentos para calcular a estimativa de seletividade nas consultas de janela.

3.2.2 Erro de contagem múltipla de objetos (C2)

O segundo critério abordou os estudos que demonstram características sobre a contagem múltipla de objetos e possíveis soluções para o problema.

3.2.3 Erro de aproximação dos objetos (C3)

O terceiro critério adotado levou em consideração trabalhos que apresentaram características do erro de aproximação dos objetos espaciais devido o uso de MBR e

técnicas para amenizar o problema.

3.2.4 Erro causado pela suposição de uniformidade dos objetos (C4)

O quarto critério atingiu os trabalhos relacionados a não uniformidade da distribuição dos objetos na área espacial do *dataset* e métodos para contornar esse problema.

3.2.5 Abrange todos os três erros individualmente (C5)

Por fim, o quinto critério busca identificar trabalhos que abordaram todos os três erros que interferem na estimativa de seletividade de maneira individual.

3.3 Trabalhos analisados

Com base nos critérios apresentados na seção anterior, foram selecionados três trabalhos, onde, cada um aborda um dos erros que causam imprecisão na estimativa de seletividade em consultas espaciais.

3.3.1 Selectivity Estimation in Spatial Databases (T1)

Acharya, Poosala e Ramaswamy (1999) concentrou-se na avaliação da seletividade em SBDEs, introduzindo vários métodos inovadores de agrupamento para a aproximação de dados espaciais. O trabalho propõe o histograma MinSkew, que realiza divisões binárias sucessivas no espaço de dados, resultando em uma grade proporcional à densidade do conjunto de dados. Ao contrário de métodos convencionais, como o histograma de grade, que pressupõem uniformidade nas células, o MinSkew adapta-se a conjuntos de dados com distribuições não uniformes, como regiões de alta ou baixa densidade de objetos.

Diversos bancos de dados espaciais exibem uma não uniformidade na densidade de objetos, especialmente aqueles gerados a partir da captura de imagens de elementos naturais, como rios, montanhas e vegetação. A natureza não uniforme desses objetos invalida a suposição de uniformidade feita por técnicas tradicionais de estimativa de seletividade. Os autores propuseram métodos inovadores de agrupamento e técnicas baseadas em densidade espacial para lidar com esse fato. O equilíbrio da densidade espacial em todos os *buckets* do histograma é um aspecto central da abordagem, garantindo que a densidade espacial de cada *bucket* reflita a quantidade real de objetos espaciais contidos nele.

3.3.2 Efficient Processing of Multiway Spatial Join Queries in Distributed Systems (T2)

Oliveira (2017) conduziu uma pesquisa abrangente que abordou não apenas as consultas de junção espacial sequencial e distribuída, mas também as multijunções espaciais sequenciais e distribuídas. Durante o estudo, foram identificadas características específicas dos *datasets* e distribuição dos dados relevantes para o processamento eficiente da junção espacial. Como resultado, propôs um modelo estatístico para a estimativa da seletividade nas consultas de junção, levando em consideração essas características.

Em complemento a isso, o autor introduziu o Histograma IHWAF, que utiliza o método de sobreposição proporcional, uma abordagem de particionamento do *dataset* em uma grade sem tamanho fixo, e fórmulas específicas de estimativa para dados do tipo linha e polígonos. Essas técnicas serviram como tratativa para o erro de aproximação dos objetos e erros de estimativa de seletividade relacionados aos tipos de objetos; empregando o método de sobreposição proporcional do MBR dos objetos para determinar à qual célula o objeto pertence, tratou, ainda, do erro de contagem múltipla de objetos. O estudo também sugeriu, como trabalho futuro, a utilidade de histogramas aprimorados, como o de Euler, para melhorar a estimativa de seletividade em sistemas distribuídos.

3.3.3 Selectivity Estimation for Spatial Joins with Geometric Selections (T3)

Sun, Agrawal e Abbadi (2002) introduziram uma inovadora abordagem para a estimativa da seletividade de junções espaciais denominada Histograma de Euler. Diferenciando-se dos histogramas de grade convencional, essa técnica foi generalizada para lidar com objetos e janelas de seleção que não seguem uma grade específica. A construção do Histograma de Euler baseia-se na teoria dos grafos, alocando *buckets* para as faces, arestas e vértices, proporcionando uma solução eficaz para o problema da contagem múltipla de objetos em histogramas tradicionais.

No mesmo contexto, os autores também apresentaram um algoritmo para estimar a seletividade de consultas de janela, fundamentado na Fórmula de Euler. Pois, o fenômeno da contagem múltipla surge devido à própria natureza dos objetos espaciais, esse erro ocorre quando um único objeto espacial é registrado duas ou mais vezes, resultando na diminuição da precisão e qualidade da seletividade do histograma. Mesmo que a especificidade seja considerável, encontrar uma divisão espacial na qual esse problema não ocorra é uma tarefa desafiadora.

3.4 Resumo Comparativo

Observa-se que Histograma IHWAF Oliveira (2017) aborda o Método de Sobreposição Proporcional, em que o objeto é contato proporcionalmente nas células que se sobrepõe. Essa técnica visa resolver o erro de contagem múltipla dos objetos. Além disso, destaca-se que o tipo de objeto no *dataset* possui alta relevância para a estimativa da consulta em si.

Por sua vez, o trabalho de Sun, Agrawal e Abbadi (2002) propõe o Histograma de Euler, fundamentado na Fórmula de Euler, onde são alocados *buckets* para as faces, arestas e vértices. Esse histograma possui células de tamanhos fixos e busca resolver o erro de contagem múltipla de objetos.

Finalmente, o trabalho de Acharya, Poosala e Ramaswamy (1999) utiliza a densidade espacial mínima para abordar a não uniformidade dos objetos no *dataset*. Essa técnica resulta em um histograma denominado Histograma MinSkew, que possui *buckets* de tamanhos variados.

Observa-se, portanto, que os trabalhos abordam tipos específicos de erro, todos eles contribuindo para a imprecisão na estimativa de seletividade nas consultas espaciais. A Tabela 1 apresenta a relação entre os trabalhos relacionados e os critérios, os critérios marcados com X estão presentes no trabalho. O T1 retrata o C1 e C4, o T2 aborda o C1 e C3, o T3 apresenta o C1 e C2.

Este trabalho, diferentemente dos demais, não apresenta novas técnicas de histogramas ou métodos para melhorar os mesmos. Ao contrário, contempla-se aqui uma avaliação experimental da contribuição individual de cada um dos erros em cada um dos histogramas, com o objetivo de averiguar a efetividades dos métodos propostos e identificar possíveis pontos de melhoria nas técnicas.

Tabela 1 – Comparativo entre trabalhos

	C1	C2	C3	C4	C5
T1	X			X	
T2	X	X	X		
T3	X	X			
Este Trabalho	X	X	X	X	X

4 Gerador de *Datasets* Sintéticos

4.1 Metodologia de Desenvolvimento

Pensando nas especificações únicas necessárias para realização dos experimentos, foi desenvolvido um gerador de *datasets* na linguagem de programação c++ (Apêndice A), que possibilitou isolar os erros de contagem múltipla de objetos, aproximação dos objetos utilizando o MBR e suposição de uniformidade.

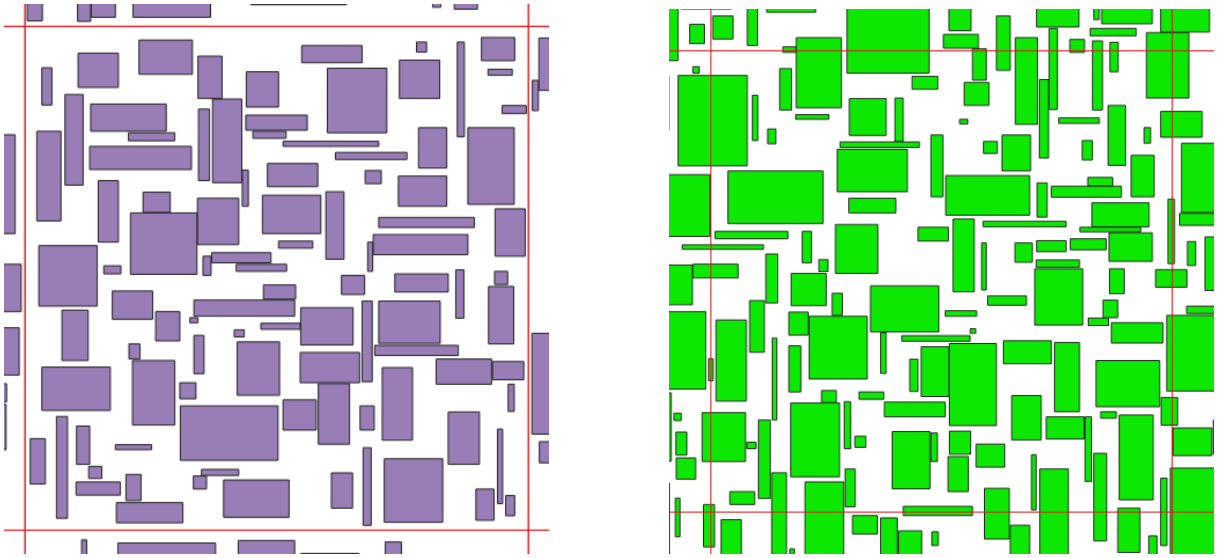
Para o funcionamento do gerador, o usuário deve fornecer um total de sete parâmetros que são descritos na Tabela 2. Eles estão ligados diretamente no isolamento dos erros. Segue uma explicação detalha de como esses isolamentos são feitos através dos parâmetros.

Tabela 2 – Parâmetros para o gerador de *datasets*

Parâmetros	*Função*
Primeiro	Quantidade de objetos
Segundo	Dimensão do *dataset*
Terceiro	Largura da grade no eixo X
Quarto	Largura da grade no eixo Y
Quinto	Área morta do MBR (0 = Não contém | 1 = Contém)
Sexto	Tipo do *dataset* (0 = Polígono | 1 = Linha)
Sétimo	Uniformidade do *dataset* (0 = Uniforme | 1 = Não Uniforme)

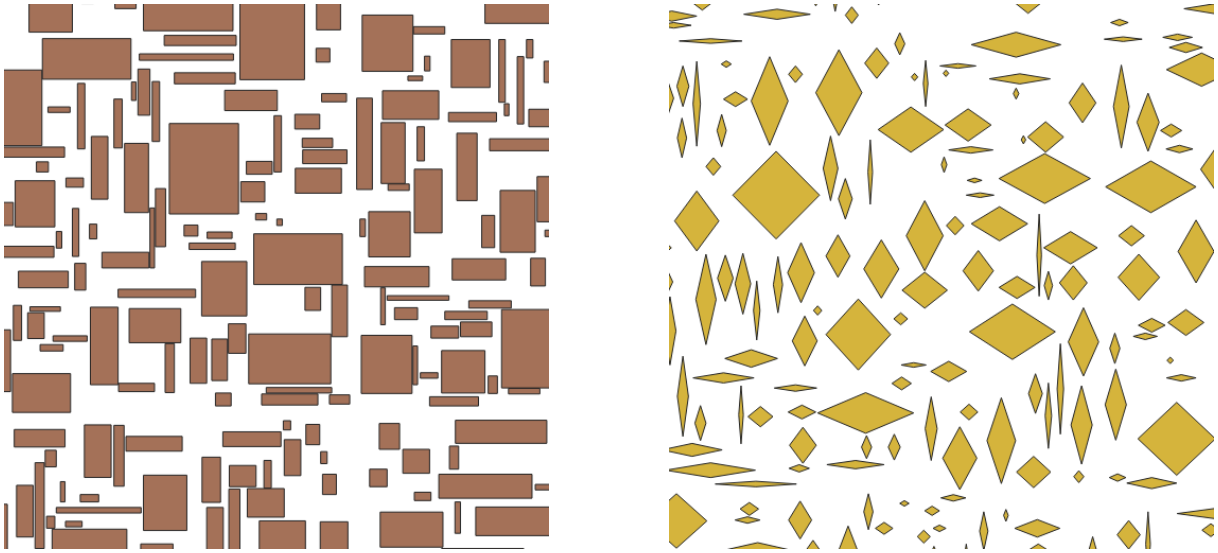
1. O primeiro e segundo parâmetros são comuns para todos os *datasets* e definem a quantidade de objetos e a dimensão espacial (comprimento e largura) do *dataset*, respectivamente.

2. Já o terceiro e quarto parâmetros são responsáveis por controlar o erro de contagem múltipla de objetos. Esse controle é feito da seguinte forma: Com a dimensão do *dataset* (segundo parâmetro) e a largura da grade dos histogramas que serão gerados futuramente para o *dataset*, nos eixos X e Y, é possível evitar que os objetos do *dataset* se sobreponham à futura grade; esse cálculo é feito através da divisão da dimensão pela largura da grade, e define um conjunto de divisões referência às quais os objetos não sobreporão. Como pode-se observar na Figura 13a o *dataset* foi desenhado para suportar um histograma com grade 10x10, onde a dimensão do *dataset* é 5 mil e a largura da grade nos eixos X e Y é 500. Assim, nenhum objeto é intersectado pela grade do histograma e isso evitará completamente o erro de contagem múltipla de objetos. Vale ressaltar que, caso haja a necessidade de provocar



(a) Representação com o método. (b) Representação sem o método.

Figura 13 – Comparação entre *datasets* com e sem utilização do método de geração que previne a contagem múltipla de objetos.



(a) Não contém área morta nos MBRs. (b) Contém área morta nos MBRs.

Figura 14 – Comparação entre *datasets* com e sem área morta nos MBRs.

o erro de contagem múltipla, como na Figura 13b, basta passar o terceiro e quarto parâmetros zerados.

3. O quinto parâmetro manipula o erro de aproximação de objetos utilizando o MBR. Se o parâmetro for 0, todos os objetos serão retangulares – como mostra a Figura 14a, e o MBR dos objetos não conterá área morta; se for 1, serão gerados objetos losangulares, retratados pela Figura 14b, onde todos os MBRs conterão área morta.

4. O sexto parâmetro define o tipo do *dataset*. Se o parâmetro for 0, o *dataset* será do tipo polígono, como apresentado na Figura 14; se for 1, o *dataset* será do tipo linha

(Figura 15).

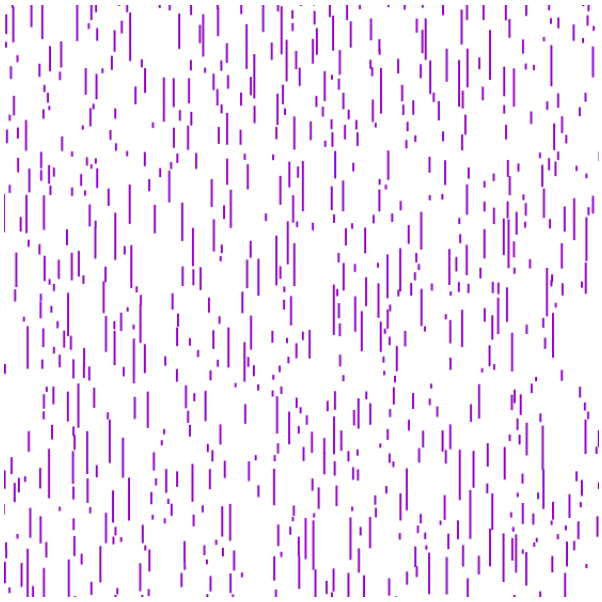
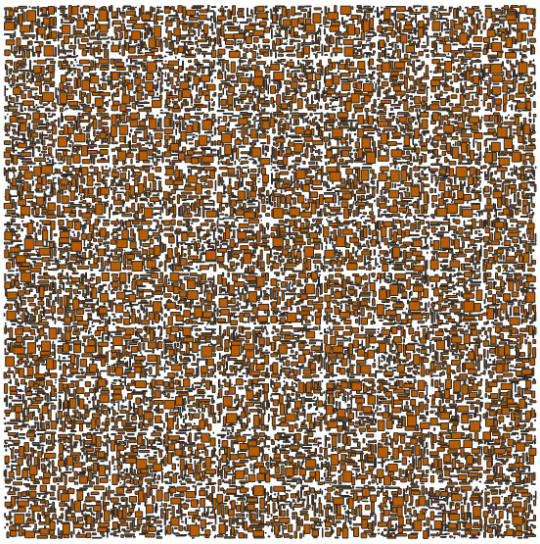


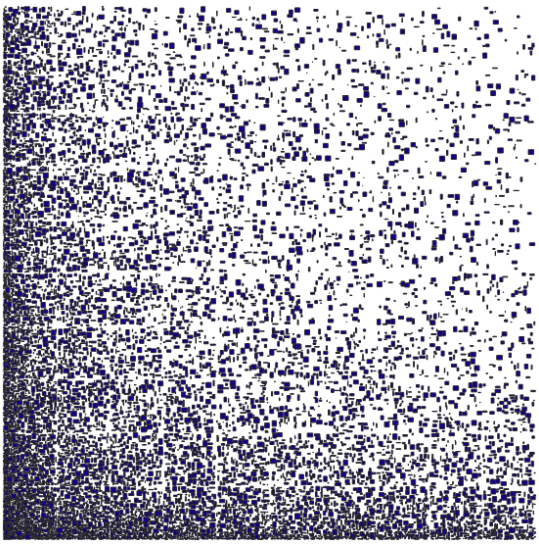
Figura 15 – *Dataset* do tipo linha.

5. Por fim, o sétimo parâmetro cuida da uniformidade do *dataset*. Quando o parâmetro é 0, o *dataset* terá distribuição uniforme de objetos, assim como na Figura 16a; quando é 1, será não uniforme com distribuição zipf, concentrando os objetos na lateral esquerda e inferior, representado pela Figura 16b. O código que implementa a distribuição zipf foi retirado do repositório do projeto que este trabalho está inserido (github.com/thborges/dgeohistogram).

Nota-se que os parâmetros foram de fundamental importância no isolamento de

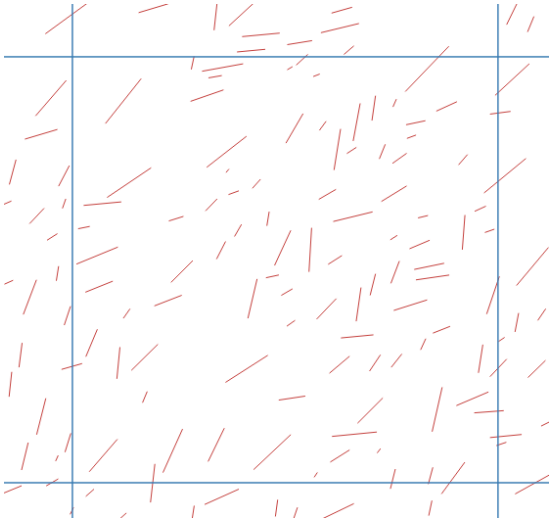


(a) *Dataset* uniforme.

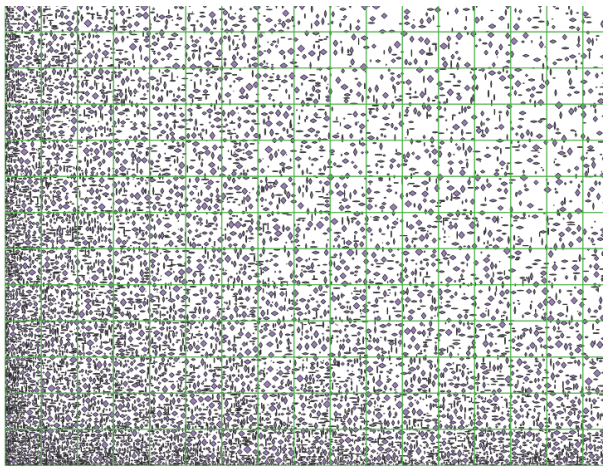


(b) *Dataset* não uniforme.

Figura 16 – Comparação entre *datasets* com distribuição uniforme e não uniforme de objetos.



(a) *Dataset* do tipo linha com erro de contagem múltipla e aproximação nos MBRs.



(b) *Dataset* do tipo polígono com os erros de contagem múltipla, suposição de uniformidade e aproximação nos MBRs.

Figura 17 – Exemplos de combinações possíveis de *datasets*.

cada erro. O algoritmo gerador de *datasets* descrito por este capítulo está no Apêndice A assim como a linha de comando necessária para compilar o mesmo.

Apesar das imagens acima abordarem um erro de cada vez, é possível fazer todas as combinações possíveis de erros. Por exemplo, se a entrada do gerador for "Gerador 10000 5000 0 0 1 1 0" resultará na Figura 17a. A mesma retrata um *dataset* uniforme do tipo linha com 10 mil objetos e 5 mil de dimensão, ele possui erro de aproximação nos MBRs e erro de contagem múltipla de objetos. Outro exemplo seria "Gerador 50000 20000 0 0 1 0 1" que geraria um *dataset* do tipo polígono, com 50 mil objetos e 20 mil de dimensão como mostra a Figura 17b; o mesmo possui erros de aproximação nos MBRs, contagem múltipla de objetos e suposição de uniformidade.

5 Avaliação e Testes

5.1 Metodologia Experimental

Os experimentos foram planejados para tratar individualmente todos os erros mencionados neste estudo. Foram conduzidos um total de dez experimentos, sendo cinco para *datasets* do tipo polígono e cinco para o tipo linha. Em todos os experimentos foram realizadas consultas de janela. Os experimentos com *datasets* do tipo polígono vão do um ao cinco, e os do tipo linha vão do seis ao dez. No primeiro e sexto experimento – primeiro de cada grupo, o *dataset* foi gerado de forma a não provocar erros nas técnicas, enquanto que o segundo e o sétimo trataram do erro de contagem múltipla de objetos. Os experimentos três e oito abordaram o erro de suposição de uniformidade, e, por fim, o quarto, quinto, nono e décimo experimentos abordaram todos os erros simultaneamente.

Como os experimentos dependem fortemente dos *datasets* utilizados, estes são melhores descritos na subseção 5.1.1.

5.1.1 Datasets *Utilizados*

Para avaliar o impacto individual do erro de contagem múltipla de objetos, do erro de aproximação dos objetos utilizando o MBR e do erro de suposição de uniformidade na precisão da estimativa de seletividade na consulta de janela, foram gerados dez *datasets* com 50 mil objetos, cinco do tipo polígono e cinco do tipo linha. A Tabela 3 apresenta as características desses *datasets*.

Tabela 3 – *Datasets* dos experimentos

Datasets	*Tipo*	*Dimensão*	*Erros Presentes*
P1	Polígono	10 mil	Nenhum erro presente
P2	Polígono	10 mil	Erro de contagem múltipla de objetos
P3	Polígono	50 mil	Erro de suposição de uniformidade
P4	Polígono	20 mil	Todos os três erros presentes
P5	Polígono	50 mil	Todos os três erros presentes
L1	Linha	10 mil	Nenhum erro presente
L2	Linha	10 mil	Erro de contagem múltipla de objetos
L3	Linha	50 mil	Erro de suposição de uniformidade
L4	Linha	20 mil	Todos os três erros presentes
L5	Linha	50 mil	Todos os três erros presentes

Os experimentos foram executados da seguinte maneira:

1. O primeiro experimento usou o *dataset* P1, que é do tipo polígono e tem 10 mil de dimensão. Foram realizadas consultas de janela com tamanho equivalente a 10% da dimensão do *dataset*. Esse valor foi escolhido para que a consulta coincidisse exatamente com a grade dos histogramas e não houvesse interferência das fórmulas que calculam a uniformidade de objetos dentro da célula do histograma. As consultas geradas cobriram toda a área do *dataset* em todos os experimentos. O intuito deste experimento foi observar como os histogramas se comportariam frente a um *dataset* que não possuía nenhum dos três principais erros que afetam sua precisão de estimativa.

2. Já o segundo experimento usou o *dataset* P2, pertencente ao tipo polígono, com 10 mil de dimensão e também realizou suas consultas com janelas de 10%. Como o *dataset* possui erro de contagem múltipla de objetos, que ocorre quando o objeto contém sua extensão em mais de uma célula do histograma, como na Figura 13b, fica claro que esse experimento aferiu como o erro de contagem múltipla interfere na precisão da estimativa dos histogramas.

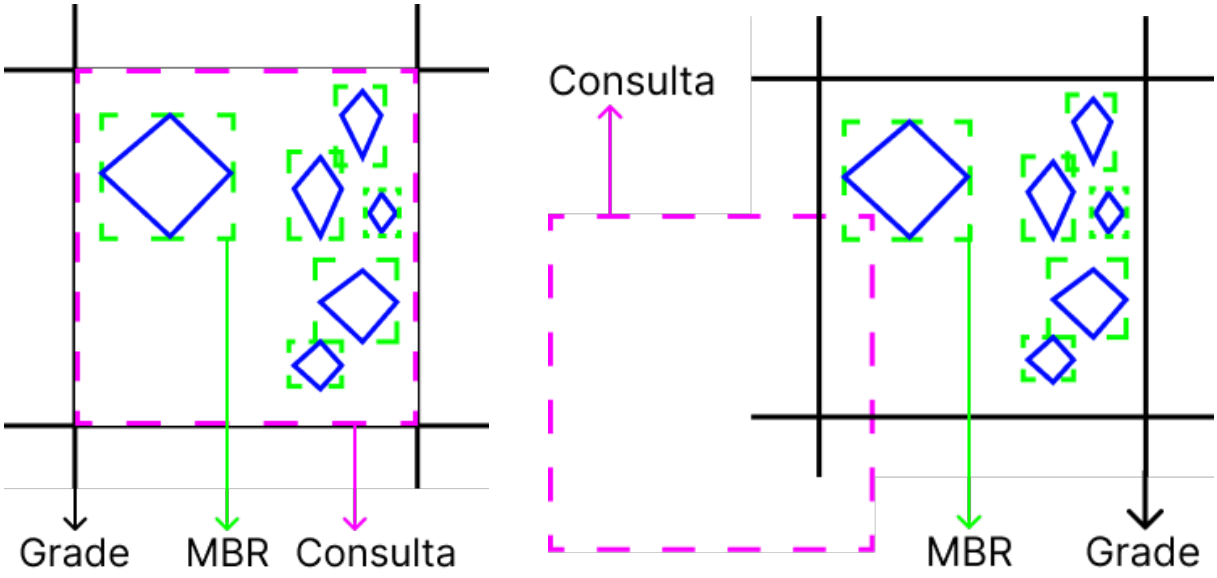
3. No terceiro experimento foi empregado o *dataset* P3, do tipo polígono, que causa erro de suposição de uniformidade; sua dimensão espacial é de 50 mil, porque, quando a extensão do *dataset* é maior, possibilita que os objetos fiquem cada vez mais distribuídos de maneira não uniforme, o que contribui perfeitamente com o objetivo deste experimento. Outra diferença para os demais experimentos é que suas consultas foram realizadas com tamanho de 7.3% da área do *dataset*, pois, ao utilizar 10%, a grade dos histogramas coincidem com a consulta, então, as células teriam 100% da sua área consultada, logo, os histogramas retornariam apenas a cardinalidade da célula, o que não manifestaria o erro das fórmulas que supõem a uniformidade dos objetos no *dataset*.

4. O quarto experimento utiliza o *dataset* P4, também do tipo polígono e com dimensão de 20 mil; ele possui os três erros e para ocasionar o erro de suposição de uniformidade suas consultas também foram realizadas com janela de tamanho 7.3%. O intuito desse experimento é observar a taxa de erro nas estimativas dos histogramas quando os três erros ocorrem simultaneamente.

5. Por fim, o quinto experimento é o último com *datasets* do tipo polígono; ele utiliza o *dataset* P5 e é similar ao seu antecessor, com exceção da dimensão espacial do *dataset*, que agora é de 50 mil. Como dito antes, quando maior a extensão do *dataset*, maior a não uniformidade dos objetos. Logo, esse experimento verificou como a estimativa se comporta quando o erro de suposição de uniformidade é intensificado.

Em relação aos experimentos com *datasets* do tipo linha, os experimentos são idênticos aos *datasets* do tipo polígono, apenas há a troca do *dataset* P1 por L1, P2 por L2, P3 por L3, P4 por L4 e P5 por L5.

Durante a execução dos experimentos e tentando contemplar a proposta inicial deste trabalho, encontrou-se uma dificuldade em isolar a contribuição individual do erro de aproximação dos objetos utilizando o MBR. Devido a isso, nota-se a ausência de experimentos destinados a analisar a contribuição individual de tal erro. Ao empregar qualquer tamanho de consulta de janela que coincida com a grade, o erro não ocorrerá, pois todos os objetos estarão completamente contidos na consulta, e a aproximação do MBR não terá impacto na estimativa. No entanto, ao utilizar um tamanho de consulta que não se alinhe com a grade, além do erro de aproximação do MBR, também será introduzido o erro de suposição de uniformidade.



(a) Não contém erro de MBR. (b) Contém erro de MBR mais erro de suposição.

Figura 18 – Comparação entre cenários de execução.

Analisando a Figura 18a percebe-se que a consulta coincide com a grade e todos os objetos estão completamente dentro da célula, ou seja, o MBR não irá interferir e, logo, a estimativa seria a cardinalidade da célula: seis objetos.

Já na Figura 18b, a consulta não coincide com a grade e intersecta apenas o MBR do objeto. Com isso, o objeto seria retornado na estimativa de maneira errônea, caracterizando o erro de aproximação utilizando o MBR. Entretanto, como a consulta intersecta uma fração da célula, as fórmulas vão supor que os objetos estão distribuídos uniformemente – o que pode não ser verdade como ilustrado na figura; há uma concentração maior de objetos há direita da célula. Portanto, também seriam estimados objetos erroneamente devido ao erro de suposição de uniformidade.

Ao se analisar a Figura 18 por completa, percebe-se que não foi possível provocar

o erro de aproximação dos objetos espaciais, devido o uso do MBR de maneira isolada, por esse motivo, não foram executados experimentos para o mesmo isoladamente.

Outra característica importante para os experimentos tem relação ao tamanho da grade em que os histogramas foram gerados, grade essa com tamanho de 100 por 100. Entretanto, os Histogramas MinSkew e EulerSkew tiveram a quantidade de *buckets* limitados em 50%, ou seja, o histograma resultante tem 5 mil *buckets*, enquanto os demais histogramas tiveram 10 mil células. Esse é a natureza das respectivas técnicas, que propõem a mesclagem das células nas quais os objetos estão uniformemente dispostos, utilizando o método de densidade espacial mínima.

5.1.2 Métricas

A avaliação da precisão da estimativa de seletividade da consulta utilizou a métrica conhecida como Soma do Erro Relativo (*Relative Error Sum*) – frequentemente empregada nas avaliações similares da literatura, que tem como função comparar a resposta esperada de uma consulta com a resposta obtida. Essa métrica é calculada por meio da Equação 5.1, na qual Q é o conjunto de consultas, e_i é a resposta estimada usando o histograma para a consulta q_i, e r_i é a resposta real.

$$\sum_{q_i \in Q} |r_i - e_i| \tag{5.1}$$

5.2 Resultados Obtidos e Análise

A soma do erro relativo na estimativa de seletividade para os experimentos descritos, que por sua vez foram realizados no Histograma IHWAF, Histograma MinSkew, Histograma EulerSkew e Histograma de Euler Melhorado estão apresentados nas tabelas a seguir.

Percebe-se que na Tabela 4 os histogramas IHWAF e Euler Melhorado não apresentam erro de estimativa quando o *dataset* é bem controlado, ou seja, não causa nenhum erro. Já os Histogramas MinSkew e EulerSkew apresentam um erro de estimativa considerável decorrente da mesclagem de suas células, mesclagem essa resultado da técnica de densidade espacial mínima que os dois histogramas utilizam, que, como discutimos, provoca o não alinhamento da célula com a grade do histograma e incorre em erro de suposição de uniformidade.

No que diz respeito ao erro de contagem múltipla, todos os histogramas apresentam erro expressivo na estimativa. Destaca-se o Histograma de Euler Melhorado, que é projetado para amenizar esse problema e contém a melhor taxa de erro dentre eles. O Histograma

Tabela 4 – Erro na estimativa de seletividade dos experimentos com *datasets* do tipo polígono.

Histograma	O dataset possui objetos que causam:				
	nenhum erro	*erro de contagem múltipla*	*erro de suposição de uniformidade*	*Todos os erros juntos, Dimensão 20 mil*	*Todos os erros juntos, Dimensão 50 mil*
IHWAF	0	2808	639	714	574
MinSkew	3937	2057	1005	1952	1083
EulerSkew	850	3873	748	1376	807
Euler Melhorado	0	1115	638	1751	838

de EulerSkew, mesmo contendo o método de Euler para tratativa do erro de contagem múltipla, mostrou-se o pior nesse cenário.

Em nossa análise, tanto o histograma MinSkew quanto o EulerSkew, em mais de um cenário, tiveram estimativas ruins devido ao problema inerente da suposição de uniformidade proveniente da mesclagem de suas células. Como possuem um número menor de *buckets* em nossos experimentos, tentamos aumentar a quantidade em experimentos adicionais. Porém, o comportamento se repetiu. Devido aos testes serem simplificados e pouco conclusivos, deixamos uma investigação a este respeito para trabalhos futuros.

Por último, todos os erros juntos foram testados em duas dimensões de *datasets* diferentes: 20 mil e 50 mil – mantida a quantidade de objetos dos datasets. Em ambos, o Histograma IHWAF se mostrou o mais assertivo. Primeiro, nota-se que o erro resultante não é equivalente à soma dos erros individuais, o que sugere que um erro anula parcialmente os outros. Ao aumentar a dimensão para 50 mil, mantendo os demais parâmetros inalterados, a quantidade dos objetos nas células diminui e os objetos tornam-se mais esparsos. Nesse cenário, o erro de estimativa também diminui consideravelmente em todos os histogramas, o que permite supor que o aumento da frequência de objetos nas células em um cenário de não uniformidade causa um consequente aumento no erro de estimativa.

Os resultados dos experimentos com *datasets* do tipo linha apresentados pela Tabela 5, em geral, tiveram taxas de erro inferiores aos experimentos com os *datasets* do tipo polígono. Entretanto, o comportamento entre os experimentos foi bem semelhante, com exceção do Histograma MinSkew que foi melhor quando o *dataset* não possui erro – os demais foram ligeiramente piores.

Uma das razões que podem ter contribuído para o erro da estimativa ser menor nos experimentos com *datasets* do tipo linha está relacionada ao formato das linhas. Elas foram desenhadas de maneira a se alinharem com os objetivos deste estudo (verticais ou

Tabela 5 – Erro na estimativa de seletividade dos experimentos com *datasets* do tipo linha.

Histograma	O dataset possui objetos que causam:				
	nenhum erro	*erro de contagem múltipla*	*erro de suposição de uniformidade*	*Todos os erros juntos, Dimensão 20 mil*	*Todos os erros juntos, Dimensão 50 mil*
IHWAF	5	1441	570	826	531
MinSkew	1908	1022	634	1198	726
EulerSkew	889	2178	699	937	712
Euler Melhorado	5	356	571	1148	676

diagonais). Como resultado, não foi possível criar objetos mais elaborados e com tamanhos de área morta variáveis. Devido ao tempo exíguo, deixamos estes experimentos para trabalhos futuros.

Por último, o histograma de Euler melhorado incorpora as melhorias projetadas para o IHWAF, no tangente à estimativa de seletividade. O que difere entre os dois é o tratamento de contagem múltipla. Devido a isso, em alguns dos experimentos, o comportamento dos dois foi similar.

6 Conclusão e Trabalhos Futuros

6.1 Conclusão

Neste trabalho foi avaliada a assertividade da estimativa de seletividade de consultas de janela usando Histograma IHWAF, Histograma MinSkew, Histograma EulerSkew e Histograma de Euler Melhorado, quando submetidos aos erros de contagem múltipla de objetos, aproximação dos objetos devido o uso do MBR e suposição de uniformidade nas equações que calculam a estimativa, erros esses que foram aplicados de maneira individual e conjunta. As estimativas foram obtidas através de consultas de janela aplicadas a *datasets* sintéticos gerados pelo algoritmo descrito no Capítulo 4.

O Histograma IHWAF foi o mais assertivo dos histogramas quando todos os erros estão presentes no *dataset*; ele também se mostrou quase perfeito quando o *dataset* não possui nenhum dos erros, além de apresentar uma maior consistência nos resultados de *datasets* de tipos diferentes. Entretanto, apresentou uma certa deficiência em relação ao erro de contagem múltipla de objetos. Como previa a hipótese, o isolamento do erro permitiu identificar uma situação que afeta a assertividade de estimativa desse histograma. Trabalhos futuros poderão focar nesse quesito individual para melhorar a técnica.

Já o Histograma MinSkew apresentou o pior erro em quase todos os experimentos, com exceção dos experimentos relacionados ao erro de contagem múltipla de objeto e ao erro de suposição de uniformidade no *dataset* do tipo linha. Vale destacar os experimentos em que não havia erros nos *datasets*, em que o Histograma MinSkew apresentou uma grande taxa de erro comparado com os demais histogramas, deixando claro que seu método de densidade mínima da célula não é muito eficiente nessas condições.

Por sua vez, o Histograma EulerSkew foi o que mostrou maior imprecisão na estimativa quando relacionado ao erro de contagem múltipla; ele foi o pior histograma em todas as situações quando o *dataset* contia apenas o erro mencionado, mesmo contendo tratativa para o erro. Este histograma também sofreu com imprecisão nos experimentos em que não havia erros nos *datasets*. Isso ocorreu por conta da utilização do mesmo método presente no Histograma MinSkew.

Por último, por ser muito parecido com o Histograma IHWAF, o Histograma de Euler Melhorado obteve praticamento os mesmos resultados nos experimentos em que não havia erros e nos experimentos que só o erro de suposição de uniformidade se fazia presente. Porém, apresentou-se muito melhor nos experimentos em que o erro de contagem múltipla foi isolado e muito pior nos experimentos em que os *datasets* causavam todos os erros.

Com os experimentos foi possível notar que independentemente da dimensão e do

tipo do *dataset*, a principal fonte de imprecisão na estimativa dos histogramas é o erro de contagem múltipla de objetos. Se o *dataset* possuir uma área de extensão considerada pequena, o erro de contagem múltipla vem seguido pelo erro de suposição de uniformidade. Entretanto, se o *dataset* possuir uma grande extensão, o erro de suposição de uniformidade não agrava muito a precisão de estimativa de seletividade.

6.2 Trabalhos futuros

Os estudos realizados neste trabalho proporcionam novas perspectivas quanto a interferência dos erros de contagem múltipla, suposição de uniformidade e aproximação do objeto usando MBR, na precisão da estimativa de seletividade dos histogramas. Como parte de trabalhos futuros pode-se desenvolver novas técnicas para resolver esses erros, principalmente, o erro de contagem múltipla, que se mostrou o principal causador da precisão da estimativa.

Uma nova metodologia experimental onde seja capaz realizar o isolamento do erro de aproximação dos objetos espaciais, devido o uso de MBR, também pode ser considerada em trabalhos futuros.

Referências

ACHARYA, S.; POOSALA, V.; RAMASWAMY, S. Selectivity estimation in spatial databases. *SIGMOD Record*, v. 28, n. 2, p. 13–24, 1999. Citado 6 vezes nas páginas 16, 19, 25, 27, 32 e 34.

AJI, A.; WANG, F.; SALTZ, J. H. Towards building a high performance spatial query system for large scale medical imaging data. In: *Proceedings of the ACM International Symposium on Advances in Geographic Information Systems*. Redondo Beach, CA, USA: [s.n.], 2012. p. 309–318. Citado na página 16.

BOUROS, P.; MAMOULIS, N. Spatial joins: What's next? *SIGSPATIAL Special*, v. 11, n. 1, p. 13–21, 2019. Citado na página 14.

CAMPBELL, J. E. Geographic information system basics. p. 18–30, 2012. Citado 4 vezes nas páginas 9, 18, 21 e 22.

ELMASRI, R.; NAVATHE, S. Fundamentals of database systems. *Addison-Wesley Publishing Company*, 2010. Citado 2 vezes nas páginas 14 e 22.

FITZ, P. R. Geoprocessamento sem complicação. In: _____. [S.l.]: Oficina de textos, 2018. ISBN 978-85-86238-82-6. Citado 2 vezes nas páginas 14 e 21.

LIU, Q.; LIN, X.; YUAN, Y. Web technologies research and development - apweb 2005. In: _____. [S.l.]: Springer Berlin Heidelberg, 2005. cap. Summarizing spatial relations – a hybrid histogram, p. 464–476. Citado na página 23.

LIU, Q.; YUAN, Y.; LIN, X. Multi-resolution algorithms for building spatial histograms. *AUSTRALIAN COMPUTER SOCIETY, INC*, Proceedings of the 14th Australasian database conference-Volume 17, p. 145–151, 2003. Citado 2 vezes nas páginas 16 e 25.

MAMOULIS, N.; PAPADIAS, D. Advances in spatial and temporal databases. In: _____. [S.l.]: Springer, 2001. (Lecture Notes in Computer Science, v. 2121), cap. Selectivity Estimation of Complex Spatial Queries, p. 155–174. Citado 3 vezes nas páginas 15, 18 e 24.

OLIVEIRA, T. B. de. *Efficient Processing of Multiway Spatial Join Queries in Distributed Systems*. 152 p. Tese (Doutorado) — Instituto de Informática, Universidade Federal de Goiás, Goiânia, GO, Brasil, 11 2017. Citado 11 vezes nas páginas 9, 14, 15, 18, 19, 22, 23, 29, 30, 33 e 34.

OLIVEIRA, T. B. de. *Desenvolvimento e Integração de Estruturas de Dados e Algoritmos para Processamento Eficiente da Multijunção Espacial em Sistemas Distribuídos*. 2018. Projeto de Pesquisa do ICET/UFJ. Citado 2 vezes nas páginas 19 e 30.

OLIVEIRA, T. B. de et al. Scheduling distributed multiway spatial join queries: optimization models and algorithms. *International Journal of Geographical Information Science*, v. 37, n. 6, p. 1388–1419, 2023. Citado 2 vezes nas páginas 19 e 30.

OLIVEIRA, T. B. de; COSTA, F. M.; RODRIGUES, V. J. S. Definição de planos de execução distribuídos para consultas de junção espacial usando histogramas multidimensionais. In: *Proceedings of the Brasilian Symposium on Databases*. Petr'opolis, RJ, Brasil: [s.n.], 2015. p. 89–100. Citado na página 16.

OLIVEIRA, T. B. de; COSTA, F. M.; RODRIGUES, V. J. S. Distributed execution plans for multiway spatial join queries using multidimensional histograms. *Journal of Information and Data Management*, v. 7, n. 3, p. 199–214, 2017. Citado 2 vezes nas páginas 16 e 30.

SOUSA, J. M. T. de. *EulerSkew Histogram: A Hybrid method to Improve the Selectivity Estimation of Spatial Window Queries*. 48 p. Monografia — Universidade Federal de Jataí, Jataí, GO, Brasil, 2022. Citado na página 28.

SUN, C.; AGRAWAL, D.; ABBADI, A. E. Advances in database technology. In: _____. [S.l.]: Springer Berlin Heidelberg, 2002. cap. Selectivity Estimation for Spatial Joins with Geometric Selections, p. 609–626. Citado 5 vezes nas páginas 16, 18, 26, 33 e 34.

VUOLO, J. H. Fundamentos da teoria de erros. In: _____. [S.l.]: Câmara Brasileira do Livro, 1996. cap. Erros Sistemáticos e Estatísticos, p. 77–87. Citado 2 vezes nas páginas 19 e 30.

Apêndices

APÊNDICE A – Código do Gerador de *Datasets* Sintéticos

```
// g++ -std=c++0x Gerador.cc -lspatialindex_c -lspatialindex -o Gerador
#include <iostream>
#include <spatialindex/SpatialIndex.h>
#include <spatialindex/capi/sidx_api.h>
#include <spatialindex/capi/sidx_impl.h>
#include <spatialindex/capi/sidx_config.h>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>
#include <vector>
#include <sstream>

using namespace std;
using namespace SpatialIndex;

int contador;
uint32_t pto;
std::vector<SpatialIndex::Region> insertedRectangles;

class MyVisitor : public IVisitor {
public:
    MyVisitor() {}

    void visitNode(const INode& n) {}

    void visitData(const IData& d) {
        SpatialIndex::IShape* shape;
        d.getShape(&shape);

        SpatialIndex::Point center;
        shape->getCenter(center);
    }

    void visitData(std::vector<const IData*>& v) {}
};

```

```
38 class MyCountVisitor : public IVisitor {
39 public:
40     int Count;
41 
42     MyCountVisitor() {
43       Count = 0;
44     }
45 
46     void visitNode(const INode& n) {}
47 
48     void visitData(const IData& d) {
49         Count++;
50     }
51 
52     void visitData(std::vector<const IData*>& v) {}
53 };
54 
55 float get_rand(float x1, float x2)
56 {
57   float temp;
58 
59   do
60     temp = ((float)rand()) / RAND_MAX;    /*  temp is now between 0 and 1 */
61   while (temp == 0.0 || temp == 1.0);
62 
63   /*  Scale temp so it is between x1 and x2. */
64   temp = (x2 - x1) * temp + x1;
65 
66   return temp;
67 }
68 
69 int randomGauss(float mean, float sigma) {
70 
71   float v1, v2;
72   float s;
73   float x;
74 
75   do
76   {
77     v1 = get_rand (-1.0, 1.0);
78     v2 = get_rand (-1.0, 1.0);
79     s = v1*v1 + v2*v2;
```

```
80    }
81    while (s >= 1.0);
82
83    x = v1 * sqrt ( -2. * log (s) / s);
84
85    /*  x is normally distributed with mean 0 and sigma 1. */
86    x = x * sigma + mean;
87
88    return (int)x;
89
90  }
91
92  int randomSkewed(int p, float V) {
93
94    int i;
95    float HsubV = 0.0;
96    for(i=1;i<=V;i++) {
97      HsubV += 1.0/pow( (double)i, p) ;
98    }
99
100   double r = get_rand(0.0, 1.0)*HsubV;
101
102   double sum = 1.0; i=1;
103   while(sum<r){
104     i++;
105     sum += 1.0/pow( (double)i, p);
106   }
107   return i;
108 }
109
110 int get_rand_value(int type, int range, float mean, float sigma, int skew_p) {
111
112   int temp;
113   switch (type) {
114     case 0: // uniform
115         return rand()%range+1;
116
117     case 1: // gauss
118         temp = randomGauss(mean, sigma);
119         if (temp <= 0)
120             temp = -temp;
121         else
```

```
122         if (temp > range)
123             temp = range-(temp-range);
124         return temp;
125 
126     case 2: // zipf
127         return randomSkewed(skew_p, (float)range);
128   }
129 
130   return 0;
131 }
132 
133 bool isOverlapping(const SpatialIndex::Region& a, const SpatialIndex::Region&
        b) {
134     return !(a.m_pHigh[0] < b.m_pLow[0] || a.m_pLow[0] > b.m_pHigh[0] ||
135              a.m_pHigh[1] < b.m_pLow[1] || a.m_pLow[1] > b.m_pHigh[1]);
136 }
137 
138 bool tryToAddRectangle(ISpatialIndex** tree, FILE *fp, int D, int GRX, int
        GRY, bool R, int T, int U, int maxAttempts = 100, bool isLast = false) {
139     for (int attempts = 0; attempts < maxAttempts; ++attempts) {
140 
141         double width = 5 + rand() % 100; // Largura do retangulo
142         double height = 5 + rand() % 100; // Altura do retangulo
143 
144         double x = 0;
145         double y = 0;
146         if(U == 0){
147             x = (double)(rand() % D);
148             y = (double)(rand() % D);
149         }else if(U == 1){
150             x = (double)(get_rand_value(2, D, 1, 1, 1) % D);
151             y = (double)(get_rand_value(2, D, 1, 1, 1) % D);
152         }
153 
154         double lowerLeftX = x - width / 2;
155         double lowerLeftY = y - height / 2;
156         double upperRightX = x + width / 2;
157         double upperRightY = y + height / 2;
158 
159         if(lowerLeftX < 0 || lowerLeftY < 0 || upperRightX > D || upperRightY
            > D) continue;
160 
```

```
161         if (GRX != 0) { // grade ativada
162             if (lowerLeftX <= 0.01 && upperRightX >= 0.0){
163                 continue;
164             }
165 
166             bool check = false;
167 
168             for (int i = 1; i <= (D/GRX); i++){
169                 if (lowerLeftX <= (GRX * i) + 1 && upperRightX >= (GRX * i) -
                        1){
170                     check = true;
171                 }
172             }
173 
174             if (check) continue;
175         }
176 
177         if (GRY != 0) { // grade ativada
178             if (lowerLeftY <= 0.01 && upperRightY >= 0.0){
179                 continue;
180             }
181 
182             bool check = false;
183 
184             for (int i = 1; i <= (D/GRY); i++){
185                 if (lowerLeftY <= (GRY * i) + 1 && upperRightY >= (GRY * i) -
                        1){
186                     check = true;
187                 }
188             }
189 
190             if (check) continue;
191         }
192 
193 
194         double lowerCorner[] = {lowerLeftX, lowerLeftY};
195         double upperCorner[] = {upperRightX, upperRightY};
196         SpatialIndex::Region reg = SpatialIndex::Region(
197             SpatialIndex::Point(lowerCorner, 2),
198             SpatialIndex::Point(upperCorner, 2)
199         );
200 
```

```
201         MyCountVisitor cv;
202         (*tree)->intersectsWithQuery(reg, cv);
203         bool overlap = (cv.Count > 0);
204 
205         if (!overlap) {
206             std::ostringstream os;
207             os << reg;
208             std::string data = os.str();
209             (*tree)->insertData(data.size() + 1, reinterpret_cast<const
                    uint8_t*>(data.c_str()), reg, pto++);
210             insertedRectangles.push_back(reg);
211 
212             if(T == 0){
213                 if(R == 0){
214                     if (isLast) {
215                         fprintf(fp, "\t\t{ \"type\": \"Feature\",
                                \"properties\": { \"id\": %d }, \"geometry\": {
                                \"type\": \"Polygon\", \"coordinates\": [[ [%lf,
                                %lf], [%lf, %lf], [%lf, %lf], [%lf, %lf], [%lf, %lf]
                                ]] } }\n",
216                         pto, lowerLeftX, lowerLeftY, lowerLeftX, upperRightY,
                                upperRightX, upperRightY, upperRightX, lowerLeftY,
                                lowerLeftX, lowerLeftY);
217                     } else {
218                         fprintf(fp, "\t\t{ \"type\": \"Feature\",
                                \"properties\": { \"id\": %d }, \"geometry\": {
                                \"type\": \"Polygon\", \"coordinates\": [[ [%lf,
                                %lf], [%lf, %lf], [%lf, %lf], [%lf, %lf], [%lf, %lf]
                                ]] } },\n",
219                         pto, lowerLeftX, lowerLeftY, lowerLeftX, upperRightY,
                                upperRightX, upperRightY, upperRightX, lowerLeftY,
                                lowerLeftX, lowerLeftY);
220                     }
221                 }else if(R == 1){
222                     if (isLast) {
223                         fprintf(fp, "\t\t{ \"type\": \"Feature\",
                                \"properties\": { \"id\": %d }, \"geometry\": {
                                \"type\": \"Polygon\", \"coordinates\": [[ [%lf,
                                %lf], [%lf, %lf], [%lf, %lf], [%lf, %lf], [%lf, %lf]
                                ]] } }\n",
224                         pto, lowerLeftX, ((lowerLeftY + upperRightY)/2),
                                ((lowerLeftX + upperRightX)/2), ((upperRightY +
```

```
                        upperRightY)/2), upperRightX, ((upperRightY +
                        lowerLeftY)/2), ((upperRightX + lowerLeftX)/2),
                        lowerLeftY, lowerLeftX, ((lowerLeftY +
                        upperRightY)/2));
225                 } else {
226                     fprintf(fp, "\t\t{ \"type\": \"Feature\",
                        \"properties\": { \"id\": %d }, \"geometry\": {
                        \"type\": \"Polygon\", \"coordinates\": [[ [%lf,
                        %lf], [%lf, %lf], [%lf, %lf], [%lf, %lf], [%lf, %lf]
                        ]] } },\n",
227                     pto, lowerLeftX, ((lowerLeftY + upperRightY)/2),
                        ((lowerLeftX + upperRightX)/2), ((upperRightY +
                        upperRightY)/2), upperRightX, ((upperRightY +
                        lowerLeftY)/2), ((upperRightX + lowerLeftX)/2),
                        lowerLeftY, lowerLeftX, ((lowerLeftY +
                        upperRightY)/2));
228                 }
229             }
230         }else if(T == 1){
231             if(R == 0){
232                 if (isLast) {
233                     fprintf(fp, "\t\t{ \"type\": \"Feature\",
                        \"properties\": { \"id\": %d }, \"geometry\": {
                        \"type\": \"LineString\", \"coordinates\": [[%lf,
                        %lf], [%lf, %lf] ]} }\n",
234                     pto, lowerLeftX, lowerLeftY, lowerLeftX, upperRightY);
235                 } else {
236                     fprintf(fp, "\t\t{ \"type\": \"Feature\",
                        \"properties\": { \"id\": %d }, \"geometry\": {
                        \"type\": \"LineString\", \"coordinates\": [[%lf,
                        %lf], [%lf, %lf] ]} },\n",
237                     pto, lowerLeftX, lowerLeftY, lowerLeftX, upperRightY);
238                 }
239             }else if(R == 1){
240                 if (isLast) {
241                     fprintf(fp, "\t\t{ \"type\": \"Feature\",
                        \"properties\": { \"id\": %d }, \"geometry\": {
                        \"type\": \"LineString\", \"coordinates\": [[%lf,
                        %lf], [%lf, %lf] ] } }\n",
242                     pto, lowerLeftX, lowerLeftY, ((lowerLeftX +
                        upperRightX)/2), ((upperRightY + lowerLeftY)/2));
243                 } else {
```

```
244                     fprintf(fp, "\t\t{ \"type\": \"Feature\",
                            \"properties\": { \"id\": %d }, \"geometry\": {
                            \"type\": \"LineString\", \"coordinates\": [[%lf,
                            %lf], [%lf, %lf] ] } },\n",
245                     pto, lowerLeftX, lowerLeftY, ((lowerLeftX +
                            upperRightX)/2), ((upperRightY + lowerLeftY)/2));
246                 }
247             }
248         }
249
250         return true;
251     }
252 }
253
254 return false;
255 }
256
257
258 int main(int argc, char** argv){
259     clock_t tempoInicial, tempoFinal;
260     double tempoGasto;
261
262     srand((unsigned)time(NULL));
263     if (argc < 2) {
264         cout << "Usage: " << argv[0] << " <number_of_rectangles>" << endl;
265         return -1;
266     }
267
268     contador = atoi(argv[1]); // Define contador como a quantidade de objetos
269     int D = atoi(argv[2]); // Define D como a dimensao do dataset
270     int GRX = atoi(argv[3]); // Define GRX com a largura da GradeX
271     int GRY = atoi(argv[4]); // Define GRY com a largura da GradeY
272     int R = atoi(argv[5]); // Define R como a ativacao da area morta do MBR |
            0 = false e 1 = true
273     int T = atoi(argv[6]); // Defibe T como o tipo do dataset | 0 = Poligono e
            1 = Linha
274     int U = atoi(argv[7]); // Define U como a uniformidade do dataset | 0 =
            Uniforme e 1 = Nao uniforme
275
276     pto = 1;
277     tempoInicial = clock();
278
```

```
279     FILE *fp = fopen("Gerador.geojson","w");
280     fprintf(fp,"{\n\t\"type\": \"FeatureCollection\",\n\t\"crs\": { \"type\":
            \"name\", \"properties\": { \"name\":
            \"urn:ogc:def:crs:OGC:1.3:CRS84\"} },\n\t\"features\": [\n");
281
282     try {
283         std::string baseName = "file";
284         IStorageManager* diskfile =
                StorageManager::createNewDiskStorageManager(baseName, 4096);
285         StorageManager::IBuffer* file =
                StorageManager::createNewRandomEvictionsBuffer(*diskfile, 10,
                false);
286         id_type indexIdentifier;
287         ISpatialIndex* tree = RTree::createNewRTree(*file, 0.7, 10, 10, 2,
                SpatialIndex::RTree::RV_RSTAR, indexIdentifier);
288
289         int addedRectangles = 0;
290
291         while (addedRectangles < contador) {
292             if (tryToAddRectangle(&tree, fp, D, GRX, GRY, R, T, U, 100,
                    addedRectangles == contador - 1)) {
293                 addedRectangles++;
294             } else {
295                 //printf("Max attempts reached.");
296             }
297         }
298
299         delete tree;
300         delete file;
301         delete diskfile;
302     } catch (Tools::Exception& e) {
303         std::cerr << "******ERROR******" << std::endl;
304         std::cerr << e.what() << std::endl;
305         fclose(fp);
306         return -1;
307     }
308
309     fprintf(fp,"\t]\n}");
310     fclose(fp);
311
312     tempoFinal = clock();
313     tempoGasto = (double)(tempoFinal - tempoInicial) / CLOCKS_PER_SEC;
```

```
314
315     cout << contador << " " << tempoGasto << endl;
316
317     return 0;
318 }
```

Código A.1 – Código Gerador