

GEORGE HENRIQUE RANGEL COSTA

**GERAÇÃO DE MAPAS RODOVIÁRIOS A PARTIR DE
TRAJETÓRIAS DE OBJETOS MÓVEIS COLETADAS POR
SMARTPHONE – MÉTODO BASEADO EM ALGORITMO
GENÉTICO**

Dissertação apresentada ao Curso de Pós-graduação em Computação Aplicada do Centro de Ciências Tecnológicas, da Universidade do Estado de Santa Catarina, como requisito parcial para obtenção do grau de Mestre em Computação Aplicada.

Orientador: Fabiano Baldo

**JOINVILLE, SC
2014**

C837g

Costa, George Henrique Rangel

Geração de mapas rodoviários a partir de trajetórias de objetos móveis coletadas por smartphone: método baseado em algoritmo genético / George Henrique Rangel Costa. - 2014.

161 p. : il. ; 21 cm

Orientador: Fabiano Baldo

Bibliografia: p. 143-149

Dissertação (Mestrado) - Universidade do Estado de Santa Catarina, Centro de Ciências Tecnológicas, Programa de Pós-Graduação em Computação Aplicada, Joinville, 2014.

1. Mapas rodoviários. 2. Método para geração de mapas. 3. Smartphones. 4. Algoritmo genético. I. Baldo, Fabiano. II. Universidade do Estado de Santa Catarina. Programa de Pós-Graduação em Computação Aplicada. III. Título.

CDD: 005.1 - 20.ed.

GEORGE HENRIQUE RANGEL COSTA

GERAÇÃO DE MAPAS RODOVIÁRIOS A PARTIR DE

TRAJETÓRIAS DE OBJETOS MÓVEIS COLETADOS POR

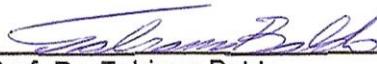
SMARTPHONE - MÉTODO BASEADO EM ALGORITMO

GENÉTICO

Dissertação apresentada ao Curso de Mestrado Acadêmico Computação Aplicada como requisito parcial para obtenção do título de Mestre em Computação Aplicada na área de concentração "Ciência da Computação".

Banca Examinadora

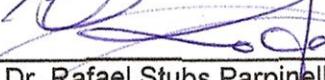
Orientador:


Prof. Dr. Fabiano Baldo
CCT/UDESC

Membros


Profa. Dra. Vania Bogorny
UFSC


Prof. Dr. Fernando José Braz
IFC


Prof. Dr. Rafael Stubs Parpinelli
CCT/UDESC


Profa. Dra. Avanilde Kemczinski
CCT/UDESC

Joinville, SC, 20 de fevereiro de 2014.

AGRADECIMENTOS

Ao meu orientador, Prof. Dr. Fabiano Baldo, pela atenção, paciência e disponibilidade sempre que necessário, além do constante apoio e incentivo durante todo o desenvolvimento deste trabalho.

À minha família, pela paciência, compreensão e todos os conselhos durante os momentos difíceis; pelo afeto, estímulo e companheirismo nos momentos desafiadores; e por compartilhar dos momentos de alegria, antes e durante o mestrado.

Aos meus amigos, pela compreensão sempre que eu estava ocupado e pelos momentos de diversão quando possível.

Aos professores da UDESC, por todas as dicas e todo o incentivo.

A todos os membros da banca, pelas sugestões e críticas construtivas que contribuíram para aprimorar o documento final.

A todas as pessoas que contribuíram para a realização deste trabalho, seja ajudando na coleta de dados ou de qualquer outra forma.

A UDESC, pela oportunidade de realizar este curso.

A CAPES, pelo apoio financeiro para realização deste trabalho.

RESUMO

COSTA, George Henrique Rangel. **Geração de mapas rodoviários a partir de trajetórias de objetos móveis coletadas por smartphone – Método baseado em algoritmo genético.** 2014. 161 f. Dissertação em Computação Aplicada – Área: Engenharia de Software. Universidade do Estado de Santa Catarina. Programa de Pós-graduação em Computação Aplicada, Joinville, 2014.

A popularização de dispositivos com receptor GPS integrado aumentou consideravelmente o uso de mapas rodoviários digitais. Por este motivo, é imprescindível que eles sejam acurados e atualizados. Os métodos atualmente utilizados para gerar estes mapas – fotogrametria e edição colaborativa – têm baixa frequência de atualização, pois dependem de intervenção manual. Desta forma, fica evidente a necessidade de um método automatizado para geração de mapas rodoviários. A literatura apresenta soluções que utilizam trajetórias de objetos móveis para encontrar o centro das vias, mas nenhuma delas está preparada para a atualização contínua das vias e o refinamento dos mapas. Sendo assim, este trabalho visa propor um novo método para encontrar o centro das vias utilizando trajetórias providas por GPS embutido em *smartphones*. Assume-se que os pontos que representam os centros das vias podem ser obtidos através de aproximações providas por um algoritmo evolutivo, tal como o algoritmo genético. A seguir, estes pontos são combinados para gerar o mapa rodoviário. Entretanto, o uso de trajetórias coletadas com *smartphones* proporciona alguns desafios, tais como: eliminação de dados com acurácia ruim, identificação do meio de transporte utilizado e redução do volume de dados processados. Portanto, o objetivo do presente trabalho é propor um método estruturado que limpe, analise e enriqueça dados obtidos por *smartphones* para gerar mapas rodoviários acurados e atualizados continuamente, utilizando algoritmo genético. Os resultados dos testes apontam que o método é capaz de gerar mapas de qualidade próxima a dos mapas de referência. Com base nos cenários utilizados para realizar esta

comparação, a diferença média foi de 2,26 metros. Além disso, os testes também demonstram a viabilidade da atualização periódica e contínua dos mapas proposta pelo método.

Palavras-chave: Mapas rodoviários. Método para geração de mapas. *Smartphones*. Algoritmo genético.

ABSTRACT

COSTA, George Henrique Rangel. **Generation of road maps from moving objects trajectories collected with smartphone – Method based on genetic algorithm.** 2014. 161 f. Master thesis in Applied Computing – Area: Software Engineering. Santa Catarina State University. Postgraduate Program in Applied Computing, Joinville, Santa Catarina, Brazil, 2014.

The popularization of devices with integrated GPS receiver has considerably boosted the use of digital road maps. For this reason, it is mandatory that they be accurate and up-to-date. The methods currently used to generate these maps – photogrammetry and collaborative editing – have low frequency of update because they depend on manual intervention. Thus, the need for an automated method for generating road maps is highlighted. The literature presents solutions that use trajectories of moving objects to find the center of the roads, but none of them is prepared for the continuous update of the roads and the refinement of the maps. Therefore, this work aims to propose a new method to find the center of the roads using trajectories provided by GPS receivers integrated in smartphones. It is assumed that the points that represent the center of the roads can be found through approximations provided by an evolutive algorithm such as the genetic algorithm. After that, these points are combined to generate the road map. However, the use of trajectories collected with smartphones provides some challenges, such as: elimination of data with bad accuracy, identification of the means of transport used and reduction of the volume of data processed. Thus, the objective of this work is to propose a structured method that cleans, analyzes and enriches data from smartphones to generate accurate road maps that can be continuously updated, using genetic algorithm. Test results indicate that the method is capable of generating maps with quality close to the reference ones. Based on the scenarios used to perform this comparison, the average difference between them

is 2.26 meters. The tests also show that the periodic and continuous update of the map as proposed by the method is viable.

Keywords: Road maps. Map generation method. Smartphones. Genetic Algorithm.

LISTA DE FIGURAS

Figura 01 – Exemplos de centros das vias	24
Figura 02 – Árvore de Decisão	42
Figura 03 – Identificação de meios de transporte.....	43
Figura 04 – Ângulo de movimento	44
Figura 05 – Qualidade dos dados coletados	45
Figura 06 – Quantidade X qualidade dos pontos coletados.....	46
Figura 07 – Filtragem das trajetórias	47
Figura 08 – Métrica Distância Perpendicular	49
Figura 09 – Métrica Distância Proporcional ao Tempo.....	50
Figura 10 – Métrica Distância Euclidiana Sincronizada.....	51
Figura 11 – <i>Uniform Sampling</i> ($n=3$).....	52
Figura 12 – <i>Douglas-Peucker</i>	53
Figura 13 – <i>Normal Opening Window</i>	54
Figura 14 – <i>Before Opening Window</i>	54
Figura 15 – <i>Spatial Quality Simplification Heuristic</i>	56
Figura 16 – <i>Dead Reckoning</i>	57
Figura 17 – Resultado da compressão para diferentes valores de erro máximo permitido (ϵ)	60
Figura 18 – Gráfico de erro médio introduzido na trajetória após compressão	61
Figura 19 – Gráfico de maior erro introduzido na trajetória após compressão	62
Figura 20 – Problema do <i>Dead Reckoning</i>	63
Figura 21 – Gráfico de percentagem de pontos mantidos	64
Figura 22 – Gráfico de tempo de execução.....	64
Figura 23 – Método de Cao e Krumm (2009)	76
Figura 24 – Método de Jang, Kim e Lee (2010)	77

Figura 25 – Diagrama do funcionamento do método	82
Figura 26 – Resultado da consulta ao banco de dados	87
Figura 27 – Definição do conjunto de pontos selecionados	88
Figura 28 – Diferença na angulação de duas vias	89
Figura 29 – Descarte de pontos com ângulo de movimento diferente	90
Figura 30 – Resultado da equação de influência da acurácia ...	93
Figura 31 – Comportamento da equação de distância.....	95
Figura 32 – Resultado da função de <i>fitness</i> para todos os pontos em um domínio	95
Figura 33 – Gráfico de convergência.....	98
Figura 34 – Solução de problema do grafo	101
Figura 35 – Etapas da implementação	103
Figura 36 – Métodos do <i>WebService</i>	104
Figura 37 – Modelo lógico do banco de dados.....	105
Figura 38 – Teste 1, Cenário 1: Etapas do método.....	111
Figura 39 – Teste 1, Cenário 1: Comparando ao Bing Maps...	112
Figura 40 – Teste 1, Cenário 1: Comparando ao Google Maps	113
Figura 41 – Teste 1, Cenário 1: Comparando ao OpenStreetMap	114
Figura 42 – Erro no mapeamento	115
Figura 43 – Teste 1, Cenário 2: Etapas do método.....	117
Figura 44 – Teste 1, Cenário 2: Comparando ao Bing Maps...	118
Figura 45 – Teste 1, Cenário 2: Comparando ao Google Maps	119
Figura 46 – Teste 1, Cenário 2: Comparando ao OpenStreetMap	120
Figura 47 – Teste 1, Cenário 3: Etapas do método.....	123
Figura 48 – Teste 1, Cenário 3: Comparando ao Bing Maps...	124

Figura 49 – Teste 1, Cenário 3: Comparando ao Google Maps	125
Figura 50 – Teste 1, Cenário 3: Comparando ao OpenStreetMap	126
Figura 51 – Teste 2: Atualização do mapa	128
Figura 52 – Teste 2: Atualização dos centros das vias	129
Figura 53 – Teste 3: Mudança no trânsito de uma via	130
Figura 54 – Teste Quantitativo: Cenários	132
Figura 55 – Teste Quantitativo, Cenário 1	133
Figura 56 – Teste Quantitativo, Cenário 2	134
Figura 57 – Teste Quantitativo, Cenário 3	135
Figura 58 – Área completa mapeada.....	137
Figura 59 – Coletor: Tela inicial	151
Figura 60 – Coletor: Menu de Opções.....	152
Figura 61 – Coletor: Coleta de trajetória	153
Figura 62 – Coletor: Aviso na área de notificação	154
Figura 63 – Coletor: Mapa de trajetórias coletadas.....	155

LISTA DE TABELAS

Tabela 01 – Solução usada por cada trabalho e resultado obtido	38
Tabela 02 – Meios de transporte identificados por cada trabalho	39
Tabela 03 – Percentuais de acerto da Árvore de Decisão	41
Tabela 04 – Resumo dos algoritmos de compressão de trajetória revisados	58
Tabela 05 – Teste Quantitativo, Cenário 1: Diferença calculada.....	133
Tabela 06 – Teste Quantitativo, Cenário 2: Diferença calculada.....	134
Tabela 07 – Teste Quantitativo, Cenário 3: Diferença calculada.....	135

LISTA DE QUADROS

Quadro 01 – Matriz de Confusão	41
Quadro 02 – Pseudocódigo do método de identificação dos centros das vias.....	86
Quadro 03 – Resumo do algoritmo genético	97
Quadro 04 – Pseudocódigo do método de criação do grafo	100

LISTA DE SIGLAS

AEOM	Algoritmos Evolutivos para Otimização Multiobjetivo
CFS	Seleção de Características Baseada na Correlação (do inglês <i>Correlation Based Feature Selection</i>)
ETL	Extração, Transformação, Carga (do inglês <i>Extract, Transform, Load</i>)
GPS	Sistema de Posicionamento Global (do inglês <i>Global Positioning System</i>)
OSM	<i>OpenStreetMap</i>

SUMÁRIO

1	INTRODUÇÃO	21
1.1	OBJETIVO	26
1.1.1	Objetivo geral.....	26
1.1.2	Objetivos específicos.....	26
1.2	JUSTIFICATIVA.....	27
1.3	METODOLOGIA.....	29
1.3.1	Caracterização metodológica	29
1.3.2	Metodologia de pesquisa.....	30
1.4	ESTRUTURA DO TRABALHO.....	31
2	DESAFIOS NO USO DE TRAJETÓRIAS DE OBJETOS MÓVEIS PARA GERAÇÃO DE MAPAS RODOVIÁRIOS	33
2.1	IDENTIFICAÇÃO DO MEIO DE TRANSPORTE	34
2.1.1	Solução escolhida.....	39
2.2	REDUÇÃO DO RUÍDO.....	43
2.2.1	Solução escolhida.....	45
2.3	COMPRESSÃO DAS TRAJETÓRIAS	48
2.3.1	Métricas	49
2.3.2	Algoritmos de Compressão	51
2.3.3	Solução escolhida.....	59
2.4	IDENTIFICAÇÃO DOS CENTROS DAS VIAS USANDO ALGORITMO EVOLUTIVO.....	65
2.4.1	Fundamentos biológicos e terminologia.....	65
2.4.1.1	Cromossomos, genes e alelos	66
2.4.1.2	Codificação dos indivíduos	66
2.4.1.3	Reprodução	67
2.4.1.4	Ciclo evolutivo.....	68

2.4.2	Abordagens	70
2.4.2.1	Algoritmos Genéticos	71
2.4.2.2	Estratégias Evolutivas	71
2.4.2.3	Programação Evolutiva	72
2.4.2.4	Algoritmos Evolutivos para Otimização Multiobjetivo ..	72
2.4.3	Solução escolhida	73
2.5	OUTRAS PROPOSTAS PARA GERAÇÃO DE MAPAS RODOVIÁRIOS	75
2.5.1	Brüntrup et al. (2005)	75
2.5.2	Cao e Krumm (2009).....	75
2.5.3	Jang, Kim e Lee (2010)	76
2.5.4	Zhang, Thiemann e Sester (2010)	77
2.5.5	Niu, Li e Pousaeid (2011).....	78
2.5.6	Considerações.....	78
3	SOLUÇÃO PROPOSTA	81
3.1	MÉTODO PROPOSTO	81
3.1.1	Levantamento e coleta dos dados fonte necessários..	82
3.1.2	Pré-processamento das trajetórias	84
3.1.3	Identificação dos centros de cada via	85
3.1.3.1	Definição da ordem de análise dos pontos	87
3.1.3.2	Seleção de pontos para o conjunto S	88
3.1.3.3	Definição da função de <i>fitness</i>	91
3.1.3.4	Análise do conjunto S usando algoritmo genético	96
3.1.3.5	Continuação da execução do método.....	99
3.1.4	Criação de grafo direcionado	100
3.2	IMPLEMENTAÇÃO	102
4	ANÁLISE DOS RESULTADOS	107
4.1	METODOLOGIA DE AVALIAÇÃO	107
4.2	TESTES QUALITATIVOS	108

4.2.1	Teste de comparação com imagens de satélite.....	109
4.2.2	Teste de adição de novas vias ao mapa.....	127
4.2.3	Teste de alteração da direção de movimento de uma via	129
4.3	TESTE QUANTITATIVO – ERRO EM RELAÇÃO AO OPENSTREETMAP	131
4.4	CONSIDERAÇÕES SOBRE OS TESTES	136
5	CONCLUSÃO.....	139
5.1	TRABALHOS FUTUROS	141
	REFERÊNCIAS	143
	APÊNDICE A – Informações sobre o aplicativo para <i>smartphones</i> : Coletor	151
	APÊNDICE B – Parâmetros utilizados em todas as etapas do método desenvolvido.....	157

1 INTRODUÇÃO

Mapas cartográficos são ferramentas fundamentais para a orientação de indivíduos e a localização de objetos posicionados sobre a superfície terrestre. Estas ferramentas se caracterizam pela representação visual bidimensional de um local ou uma região, onde são detalhados elementos como relevo do solo, tipo de vegetação, bacias hidrográficas, povoamentos, estradas e rodovias, etc. (IGC, 2013). Seu uso é bastante difundido desde as épocas mais remotas, pelas civilizações que habitavam o Oriente Médio e a Ásia, passando pelo período das grandes navegações europeias, até os dias de hoje, tanto para fins militares quanto para fins civis.

Os mapas eram confeccionados originalmente em papiro e posteriormente em papel. Entretanto, com o avanço dos dispositivos eletrônicos foi possível implementar uma nova forma de confecção de mapas, os chamados mapas digitais. Até a década de 1990, devido a limitações tecnológicas que impactavam em altos custos, mapas digitais eram pouco acessíveis a população em geral, ficando praticamente restritos ao uso empresarial. Entretanto, com a redução do custo de equipamentos que utilizam o Sistema de Posicionamento Global (GPS, do inglês *Global Positioning System*), esta situação mudou radicalmente (HAKLAY; WEBER, 2008). Atualmente, dispositivos com receptor GPS integrado – como os *smartphones* e GPS automotivos – se tornaram comuns, fazendo assim com que os mapas digitais começassem a se destacar como ferramentas de uso cotidiano de grande parte da população.

No contexto dos mapas digitais, destacam-se particularmente os chamados mapas rodoviários digitais. Eles são utilizados por dispositivos e aplicações de navegação por GPS que têm por função indicar ao usuário o melhor caminho para chegar a um destino desejado. Além disso, existe uma série de outras utilidades para os mapas rodoviários digitais, tais como: encontrar pontos de interesse, conhecer a malha viária de uma cidade ou região, etc. Entretanto, para que estes mapas possam desempenhar da melhor forma possível seu papel, é imprescindível que eles reflitam a estrutura viária existente da maneira mais fiel e atualizada possível. Isso é necessário não apenas para auxiliar os usuários, como também para evitar

acidentes de trânsito que podem ter consequências graves. Nesse sentido, é fundamental que os mapas rodoviários digitais sejam atualizados periodicamente com base em dados de boa qualidade. Esta atualização periódica é necessária para adaptar os mapas a eventuais modificações que ocorram na malha viária da cidade, tais como: mudanças no sentido de movimento de uma via, desvios provocados por vias interditadas, criação de novas vias, entre outros. Todos estes fatores são questões importantes que precisam ser levadas em consideração no desenvolvimento de um método para geração de mapas rodoviários.

Normalmente, mapas desse tipo são gerados e atualizados por métodos fotogramétricos aplicados em fotos tiradas por aviões e, mais recentemente, em imagens de satélite. Esses métodos possuem um custo inicial de implementação considerado alto (JANG; KIM; LEE, 2010). O resultado final apresenta boa qualidade, mas necessita de ajustes manuais em algumas situações como, por exemplo, em regiões com alta concentração de árvores, pois elas bloqueiam a visibilidade da via. Outra desvantagem é que a frequência de atualização destes mapas é baixa e, portanto, após algum tempo eles poderão conter inconsistências em relação à realidade. Estas limitações dos métodos fotogramétricos levam à busca de soluções alternativas, onde o GPS se destaca como uma das opções.

O GPS é uma rede de satélites artificiais posicionados na órbita da Terra. Eles enviam sinais que são decodificados por aparelhos chamados de receptores GPS. Assim, esses receptores podem descobrir a localização geográfica do objeto ao qual estão acoplados, mesmo enquanto estes objetos se movimentam (NOAA, 2013). Objetos capazes de se movimentar são enquadrados na categoria de “objetos móveis”.

Um pássaro, um pedestre, um barco ou um carro são exemplos de objetos móveis. Ao acoplar um receptor GPS em um objeto móvel os dados coletados pelo receptor representam o histórico de localização e movimentação do objeto, ou seja, sua trajetória (BRAZ; BOGORNY, 2012). Uma trajetória é composta de pontos que indicam onde o objeto móvel estava localizado em cada instante de tempo.

Diversos tipos de estudos podem ser feitos a partir de trajetórias de objetos móveis. Se o objeto for um animal, podem-

se analisar rotas de migração. Se o objeto for uma pessoa, pode-se estudar seu comportamento dentro de parques e cidades. Também é possível estudar o tráfego de automóveis e a movimentação de barcos (ALVARES et al., 2011; ROCHA et al., 2010). As aplicações são variadas, e relacionam-se com as mais diversas disciplinas (BRAZ; BOGORNY, 2012). No caso da geração de mapas, podem-se utilizar as trajetórias de um ou mais objetos móveis como fonte de dados para construir um mapa que contém os caminhos percorridos por eles. Assumindo que os objetos móveis são veículos motorizados, tais como carros e ônibus, o mapa resultante pode ser considerado um mapa rodoviário.

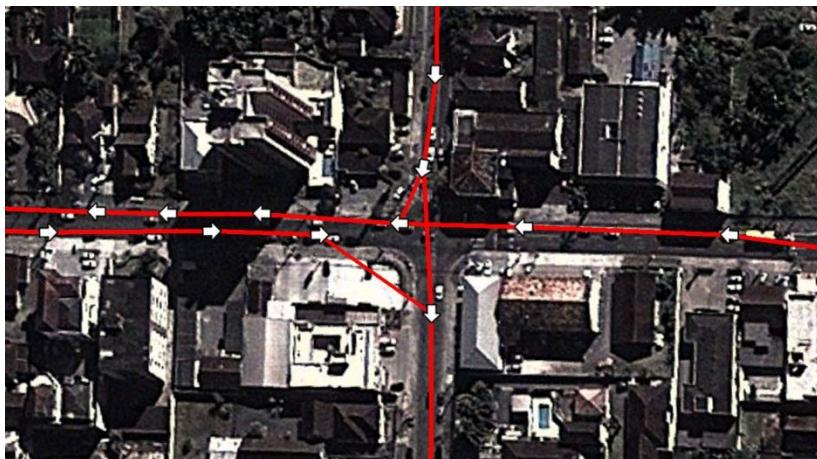
Projetos colaborativos como o *OpenStreetMap* (OSM) permitem que os usuários enviem trajetórias e depois as utilizem (em conjunto com fotos tiradas por aviões e imagens de satélite) para criar ou atualizar mapas (OSM, 2013a). Entretanto, a edição dos mapas é feita manualmente, o que pode tornar os mapas desatualizados, especialmente em regiões onde a comunidade de colaboradores do projeto é pequena. Além disso, como pessoas de qualquer lugar podem editar os mapas, é necessário considerar a possibilidade de erros causados por pessoas que desconhecem a região que estão mapeando (HAKLAY; WEBER, 2008). Nesse sentido, uma solução automática poderia ser mais eficiente, pois permitiria que os mapas fossem atualizados de forma sistemática, e não mais de forma arbitrária como nas soluções manuais. Vários estudos demonstram a viabilidade deste tipo de solução, como, por exemplo, Brüntrup et al. (2005) e Cao e Krumm (2009). Maiores detalhes sobre estes trabalhos são apresentados na seção 2.5.

Os principais desafios na geração automática de mapas rodoviários através de dados de GPS estão relacionados, de alguma forma, ao erro de acurácia existente nos pontos coletados. Este erro, que pode variar de 2 a 50 metros, interfere diretamente na identificação exata das coordenadas que determinam o posicionamento da via na superfície terrestre. Devido a essa acurácia variável do GPS os pontos coletados estão espalhados por toda a via, em uma espécie de nuvem. Por esta razão, este trabalho assume que nenhum ponto está posicionado corretamente e, portanto, novos pontos precisam ser criados para representar os centros das vias.

Consideram-se como “centros de uma via” os pontos cujas coordenadas refletem a localização da via. Eles são criados pelo método que gera o mapa para representar fielmente a extensão da via sendo mapeada. Nos casos de via de mão única, este ponto deve ser posicionado no centro da via. Já nos casos de via de mão dupla, considera-se mais adequado posicionar um ponto no centro de cada uma das mãos. A Figura 01 exemplifica ambas as situações. Nela, a via na vertical e o lado direito da via na horizontal são mão única, enquanto o lado esquerdo da via na horizontal é mão dupla. Cada flecha é um centro da via.

Apesar desta diferenciação, ao longo desse trabalho será utilizado apenas o termo “centro da via” – ou simplesmente “centros” – para descrever ambas as situações.

Figura 01 – Exemplos de centros das vias



Fonte: produção do próprio autor

O processo de identificação dos centros de uma via deve acontecer em intervalos de alguns metros. Os pontos coletados (ou seja, os pontos da nuvem) que estão posicionados dentro desse intervalo são analisados, e dão origem a um único ponto que representa o centro da via naquele trecho. Ao final, a ligação sequencial destes centros é que resultará no mapeamento de cada via e o conjunto delas no mapa rodoviário.

Como os receptores GPS possuem acurácia variável, as trajetórias coletadas não podem ser consideradas representações fiéis das vias percorridas. Assim, assume-se que os centros das vias não podem ser encontrados de maneira determinística. Por este motivo, faz-se necessário utilizar algum método de aproximação. Nesse contexto, os algoritmos evolutivos se apresentam como uma técnica adequada, pois são capazes de identificar (aproximar) uma boa solução após várias iterações. Técnicas como esta são tratadas pela área de Computação Evolutiva, e podem ser classificadas na subárea de busca e otimização. Dentre os algoritmos desse tipo, destaca-se o algoritmo genético (ver seção 2.4).

A forma de atualização do mapa também merece ser discutida. À medida que novas trajetórias são coletadas os centros das vias podem ser refinados e novas vias podem ser identificadas. Assim, um novo mapa atualizado pode ser gerado. Entretanto, este processo depende do quão recente são as trajetórias usadas como referência para gerar o mapa, e de como elas são escolhidas para fazer o mapeamento de cada via. Por este motivo, acredita-se que uma solução para encontrar os centros das vias deve levar em consideração a data em que as trajetórias foram coletadas e a acurácia de cada uma delas.

Outra questão fundamental na geração de mapas diz respeito à obtenção dos dados georreferenciados necessários para esta finalidade. Qualquer tipo de aparelho receptor GPS pode ser usado, mas o *smartphone* é uma alternativa que merece destaque. As pessoas tendem a carregá-los consigo sempre que se locomovem, por isso ele pode ser uma fonte abundante de dados. Entretanto, para a criação de mapas rodoviários, o uso de *smartphones* traz um problema adicional relacionado à identificação do meio de transporte usado ao longo dos deslocamentos. Cada meio de transporte possibilita o deslocamento do objeto móvel por diferentes locais e/ou ambientes, logo, é fundamental fazer essa distinção para não prejudicar o resultado final. Por exemplo, pessoas andando a pé podem trafegar em locais que não são vias reconhecidas, tais como praias, parques, calçadões, etc. Este tipo de trajetória não deve ser utilizado da geração de mapas rodoviários.

A qualidade dos dados utilizados também deve ser observada, pois os receptores GPS destes aparelhos podem ser

de baixa qualidade e, consequentemente, podem gerar grande quantidade de ruído. São considerados como ruído os pontos que possuem acurácia ruim e/ou informações incorretas que atrapalham a geração de mapas. Portanto, é necessário aplicar técnicas para reduzir o ruído.

Adicionalmente, a compressão das trajetórias pode ser usada como forma de reduzir a quantidade de pontos pouco relevantes utilizados na geração dos mapas, e assim reduzir o tempo de processamento.

Apesar de todos os desafios relacionados à qualidade dos dados, neste trabalho é assumido que usar trajetórias coletadas com *smartphones* traz vantagens suficientes para justificar sua aplicabilidade no contexto de geração de mapas rodoviários digitais. Com base nessa e nas outras premissas apresentadas, o problema de pesquisa é definido pela seguinte pergunta: como gerar e atualizar mapas rodoviários a partir de dados georreferenciados coletados por *smartphones*?

1.1 OBJETIVO

Baseado na importância e relevância que os mapas rodoviários digitais apresentam nos dias de hoje, a seção 1.1.1 define o objetivo geral deste trabalho e a seção 1.1.2, os objetivos específicos.

1.1.1 Objetivo geral

Com base no problema de pesquisa, o objetivo deste trabalho é propor um método estruturado que limpe, analise e enriqueça dados obtidos por *smartphones* a fim de possibilitar a geração e atualização contínua de mapas rodoviários digitais acurados, utilizando algoritmos evolutivos.

1.1.2 Objetivos específicos

A partir do objetivo geral podem ser identificados os seguintes objetivos específicos:

- Apresentar solução para a redução de ruído;
- Apresentar solução para a identificação do meio de transporte;

- Apresentar solução para a compressão de trajetórias;
- Desenvolver um método para a identificação dos centros das vias usando algoritmo evolutivo;
- Implementar ferramentas para a coleta e armazenamento dos dados de teste;
- Implementar o método desenvolvido, visando demonstrar seu funcionamento e analisar suas capacidades.

1.2 Justificativa

O uso de dados georreferenciados coletados especificamente com *smartphones* para a finalidade de geração automática de mapas ainda não é comum. Na literatura, apenas o trabalho de Niu, Li e Pousaeid (2011) menciona explicitamente o uso de *smartphones*. Apesar disso, a crescente popularização deste tipo de dispositivo – que conta com receptor GPS integrado além de outros sensores – justifica seu uso. De acordo com o relatório da empresa de pesquisa GfK sobre o mercado de bens duráveis no Brasil, durante o segundo trimestre de 2013 o mercado de *smartphones* continua em crescimento acelerado e já representa metade do mercado total de celulares do país (GFK, 2013).

É interessante destacar também que *smartphones* possuem múltiplas formas de conectividade e os sistemas operacionais utilizados nestes aparelhos disponibilizam plataformas para desenvolvimento de aplicativos. Sendo assim, é possível implementar um aplicativo que colete dados de GPS e de outros sensores e envie-os para um servidor através da Internet. Isso permitiria que mais pessoas colaborassem enviando trajetórias, o que poderia se traduzir em um mapa mais atualizado. Além disso, os dados dos outros sensores podem ser aproveitados para melhorar a qualidade do mapa gerado. Por outro lado, a liberdade que os usuários têm em coletar dados com *smartphone* se traduz na necessidade de tratar as trajetórias coletadas para identificar o meio de transporte utilizado.

Como o objetivo é gerar mapas rodoviários, apenas trajetórias que representam percursos trafegáveis por veículos terrestres motorizados são válidas. Entretanto, usando

smartphones, os usuários podem coletar em uma única trajetória dados de diversos meios de transporte. Por exemplo, ao andar até a garagem, dirigir até o trabalho e andar até o escritório têm-se deslocamentos a pé, com veículo motorizado e a pé, respectivamente. Devido a esta flexibilidade no uso de meios de transporte, justifica-se a importância de identificar e filtrar as trajetórias com base neles. Além disso, é interessante destacar que nenhum dos trabalhos relacionados encontrados (vide seção 2.5) trata desta questão.

Outro ponto a ser considerado no presente trabalho é que devido à acurácia variável do GPS não é possível definir deterministicamente por onde passa a linha que representa o centro da via. Para resolver este problema é possível utilizar mapas de referência, mas eles não estão disponíveis em todos os lugares. Além disso, eles podem interferir no processo de geração do mapa quando os dados coletados em uma região diferem do mapa usado como referência. Neste caso, surge a dúvida se são os dados que estão com ruído ou se é o mapa que está desatualizado. Por este motivo, o uso de mapas de referência não se mostra como uma solução válida.

Assim, acredita-se que uma forma adequada para encontrar os centros das vias é estimando sua localização com uma técnica de aproximação. Os algoritmos evolutivos se destacam devido à forma como executam (detalhes na seção 2.4), que aumenta as chances de encontrar uma solução próxima da melhor ao mesmo tempo em que se evita aquelas consideradas picos locais (BÄCK, 1996). Desta forma, justifica-se o uso deste tipo de algoritmo no presente trabalho. Além disso, vale destacar que não foi encontrado na literatura nenhum trabalho nesta temática que utilizasse esta forma de solução.

A literatura contém diversos métodos para geração de mapas rodoviários usando dados coletados com receptores GPS (vide seção 2.5), mas não foi encontrado nenhum que estivesse preparado para a atualização continua do mapa e refinamento da identificação do centro das vias. Acredita-se que isto pode ser alcançado analisando a data de coleta das trajetórias.

Em resumo, este trabalho propõe um novo método para geração de mapas rodoviários que tenta eliminar a necessidade de intervenção manual, uma das principais limitações dos

métodos fotogramétricos e de edição colaborativa, sem afetar a qualidade dos mapas rodoviários gerados.

1.3 Metodologia

A seção 1.3.1 apresenta a caracterização metodológica deste trabalho. Já a seção 1.3.2 apresenta a metodologia de pesquisa utilizada.

1.3.1 Caracterização metodológica

A caracterização metodológica de um trabalho científico é importante para enquadrar o objetivo e prover ao pesquisador instrumentos científicos conhecidos que o ajudem a alcançar tal objetivo. Neste sentido, a seguir é apresentada a caracterização do presente trabalho quanto a diversas classificações encontradas na literatura.

Em relação ao tipo de ciência (WAZLAWICK, 2010), o presente trabalho pode ser classificado como ***ciência inexata***, uma vez que o resultado obtido ao aplicar o método sob o mesmo conjunto de dados poderá sofrer variações a cada execução (ao contrário da *ciência exata*, onde o resultado deve ser sempre igual). Isto se deve ao uso de algoritmos evolutivos que são métodos estocásticos e, portanto, não garantem que o resultado encontrado seja sempre o mesmo.

Em relação ao paradigma científico dominante (EDEN, 2007), o presente trabalho pode ser classificado dominantemente dentro do ***paradigma tecnocrático***, pois a eficiência dos sistemas e algoritmos desenvolvidos só é conhecida após a realização de testes (ao contrário do *paradigma racionalista*, que depende de comprovação matemática; e do *paradigma naturalista*, que obriga a realização de experimentos bem sistematizados).

Em relação ao objetivo geral da pesquisa (GIL, 2002), o presente trabalho pode ser classificado como ***pesquisa explicativa***, pois ele visa identificar os fatores que determinam a ocorrência dos fenômenos – ou seja, verifica se o método proposto é capaz de alcançar os objetivos esperados (ao contrário da *pesquisa exploratória*, que visa proporcionar maior

familiaridade com o problema; e da *pesquisa descritiva*, que visa descrever um fenômeno ou estabelecer relações entre variáveis).

Em relação ao raciocínio lógico (MARCONI; LAKATOS, 2003), o presente trabalho utiliza o **método dedutivo**, pois a partir de conhecimentos gerais se inferem conhecimentos específicos. Isso significa que o desenvolvimento do método é baseado nos testes conduzidos em cenários considerados complexos, e deduz-se que resultados com a mesma qualidade podem ser obtidos em cenários mais simples.

Em relação ao grau de controle das variáveis (GIL, 2002), o presente trabalho pode ser classificado como **pesquisa quase experimental**, pois apesar de existirem grupos de controle (por exemplo, imagens de satélite) a distribuição não é totalmente aleatória: apenas dados de Joinville – Santa Catarina são coletados, enquanto que a solução, em teoria, deve funcionar para qualquer cidade.

Finalmente, em relação ao nível de maturidade da pesquisa (WAZLAWICK, 2009), o presente trabalho pode ser classificado no **nível 2**, pois ele propõe um novo método e justifica os resultados encontrados e as diferenças em relação a outros métodos existentes usando argumentação teórica e pesquisas bibliográficas.

1.3.2 Metodologia de pesquisa

A metodologia de pesquisa tem como base o método de Pesquisa-Ação. Este nome é usado de maneira ampla nas mais diversas áreas de pesquisa, e o método é sempre adaptado para se adequar às características da pesquisa a ser realizada. A intenção da Pesquisa-Ação é a evolução contínua de determinado artefato, por meio de um ciclo de análises (pesquisa) e melhorias incrementais (ação) dos diversos aspectos que o compõem. Ao contrário de outros métodos, aqui a prática constante de testes (considerando o contexto do presente trabalho) é fundamental, pois permite observar o estado em que se encontra o artefato pesquisado. Essa observação, então, serve de base para definir os próximos passos da pesquisa, que futuramente será, novamente, testada. Apesar desta metodologia de pesquisa ser menos rigorosa que outras formas de pesquisa acadêmica, as atividades realizadas em

cada ciclo ainda precisam ser revisadas quanto à sua significância e procedimentos escolhidos (TRIPP, 2005).

Sendo assim, o procedimento metodológico utilizado para a pesquisa em questão inicia com uma revisão bibliográfica de trabalhos relacionados. Paralelamente, é desenvolvido um sistema para coleta de dados de teste. A seguir, com base nos trabalhos relacionados é definida uma solução inicial. Então, à medida que os dados de teste vão sendo coletados eles servem para criar o cenário de aplicação da solução usando o método Pesquisa-Ação. Em decorrência disso são estudadas, na sequência, técnicas para identificar o meio de transporte, eliminar ruído, comprimir as trajetórias coletadas e identificar o centro das vias. Finalmente, a solução definida passa a evoluir de forma cíclica, por meio do seu refinamento a partir dos resultados obtidos nos testes realizados e da implementação gradativa das técnicas estudadas e selecionadas.

A avaliação do método descrito é feita com testes qualitativos e quantitativos, que visam apontar suas qualidades e limitações. O cenário onde os testes são executados é a cidade de Joinville – Santa Catarina. A metodologia de avaliação usa o raciocínio lógico dedutivo, explicado na seção 1.3.1. Mais detalhes da metodologia de avaliação são apresentados na seção 4.1.

1.4 Estrutura do Trabalho

O capítulo 1 tratou do aspecto introdutório, dando um panorama acerca do problema de geração de mapas rodoviários digitais e dos desafios envolvidos, de modo a demonstrar a relevância deste trabalho.

O capítulo 2 apresenta os desafios no uso de trajetórias de objetos móveis para geração de mapas rodoviários. Também se comenta sobre propostas relacionadas encontradas na literatura. Adicionalmente, é feita uma revisão sobre algoritmos evolutivos.

O capítulo 3 explica o método estruturado definido, e descreve em detalhes cada uma de suas etapas.

O capítulo 4 apresenta a avaliação dos resultados obtidos com o método proposto. Isso é feito por meio de diversos testes

qualitativos e quantitativos que visam apresentar as qualidades e limitações do método.

Por último, o capítulo 5 contém as considerações finais, onde se discorre a respeito dos resultados alcançados e possíveis trabalhos futuros.

2 Desafios no uso de trajetórias de objetos móveis para geração de mapas rodoviários

Este trabalho foca na criação de mapas através de dados de trajetórias coletadas por *smartphones*. Portanto, para que seja possível obter como resultado um mapa de boa qualidade, é imprescindível que sejam utilizados como entrada dados de boa qualidade. A qualidade não é requisito apenas para os dados utilizados para a geração de mapas. De acordo com Olson (2003), usar dados de alta qualidade é um requisito fundamental para sistemas de informação. Ele também complementa que o uso de dados acurados representa a dimensão mais importante dentro do campo de qualidade da informação. Especificamente em relação à qualidade de dados espaciais, Devillers e Jeansomlin (2006) dizem que diversos pesquisadores dividem qualidade em dois grupos: qualidade interna, que analisa a fidelidade dos dados obtidos em relação à realidade; e qualidade externa, que compara os dados obtidos com as necessidades do usuário final. A análise da geometria da trajetória e dos atributos que compõem cada um dos pontos coletados faz parte da qualidade interna, e é nessa categoria que esta seção irá focar.

De modo geral, a qualidade dos dados obtidos por receptores GPS não se mantém constante ao longo de toda trajetória. Diversos fatores podem interferir na qualidade dos sinais recebidos pelo dispositivo, e qualquer imperfeição pode provocar ruídos nos pontos coletados. Por exemplo, qualquer obstáculo no ambiente que impeça uma linha de visão direta para o céu, como árvores, casas, edifícios, etc. pode causar degradação da qualidade do sinal (DOD, 2008). Outro fator que pode causar ruídos são defeitos no hardware do aparelho ou hardware de baixa qualidade. Além disso, a coleta sistemática em um mesmo local apresenta frequentemente diferenças de posicionamento entre as trajetórias.

Além do ruído, outro desafio que afeta a qualidade dos dados para a geração de mapas rodoviários é o meio de transporte utilizado em cada trajetória coletada. Mapas rodoviários devem ser gerados a partir de trajetórias coletadas com veículos motorizados, por isso é necessário identificar e remover quaisquer trechos de trajetória que contêm dados com meios de transporte diferentes.

O volume de pontos que compõem cada trajetória e a grande quantidade de trajetórias necessárias para gerar mapas através de um método de aproximação também levantam questionamentos, especialmente em relação ao tempo de processamento. Portanto, aplicar alguma forma de compressão dos dados é fundamental para minimizar este tempo.

Estes três desafios identificados – quantidade de ruído, meio de transporte utilizado e compressão dos dados – são os principais em relação à qualidade dos dados para geração de mapas rodoviários. Além deles, existe o desafio de encontrar o centro das vias. Para solucionar este desafio, dado que a acurácia dos pontos coletados por GPS é variável, os centros das vias precisam ser criados através de aproximação. Assim, o quarto desafio é como processar as trajetórias de modo a identificar as coordenadas de cada centro.

As seções a seguir tratam destes quatro desafios. Para cada um deles, discute-se sobre sua relevância e a aplicabilidade das soluções encontradas na literatura no contexto do presente trabalho. Com base nisso, escolhe-se a solução mais adequada.

No caso dos três desafios que tratam da qualidade dos dados são apresentados também alguns dos testes realizados, a fim de demonstrar os resultados obtidos. Todos os testes foram executados no mesmo cenário para facilitar a observação dos efeitos provocados pela solução de cada desafio. Assim, o cenário escolhido é uma região do bairro São Marcos (Joinville – Santa Catarina), pois no momento em que os testes foram realizados esse era o melhor cenário disponível em relação à diversidade de meios de transporte e à qualidade dos dados.

2.1 Identificação do meio de transporte

Cada meio de transporte apresenta características de movimentação diferente, tais como, velocidade, aceleração, angulação, etc. Além disso, eles seguem regras de trânsito diferentes. Enquanto carros só andam por vias demarcadas e obedecendo a sinalização, ciclistas e, em especial, pedestres têm muito mais liberdade. Por isso, para gerar um mapa rodoviário, é fundamental levar em consideração estas diferenças. Isso significa que trajetórias de meios de transporte diferentes não devem ser combinadas e analisadas da mesma

maneira. Também se deve considerar que uma trajetória pode conter dados de mais de um meio de transporte (por exemplo, parte do trajeto a pé, parte de carro), e é preciso separar cada trecho da trajetória feito com um meio de transporte diferente. Entretanto, o receptor GPS não consegue identificar qual meio de transporte está sendo utilizado. Por este motivo, são necessárias técnicas que, com base nos dados coletados pelo receptor GPS e/ou outros sensores, seja capaz de inferir esta informação.

Mohan, Padmanabhan e Ramjee (2008) utilizam vários sensores para extrair diversas informações relacionadas ao trânsito. Com relação à identificação do meio de transporte, o que eles fazem é diferenciar entre um pedestre andando ou um carro que se locomove muito lentamente devido ao trânsito. Para esta finalidade eles usam dados captados por um sensor de acelerômetro. A identificação do meio de transporte é feita através da detecção de picos na leitura do sensor, que diferenciam entre passo de um pedestre e freadas de um carro.

Zheng et al. (2008) tentam identificar se um usuário está andando ou usando bicicleta, carro ou ônibus usando apenas o GPS. O primeiro passo da solução deles é dividir a trajetória em segmentos, de acordo com os pontos onde se acredita que o usuário trocou de meio de transporte. Estes pontos são encontrados com heurísticas, como marcação de pontos com velocidade muito próxima de zero. A seguir, o meio de transporte de cada segmento é identificado. Para isso, duas estratégias são adotadas: inferência por algoritmos de classificação ou por método estatístico. Os algoritmos de classificação testados foram Árvore de Decisão, Máquinas de Vetores de Suporte, e Rede Bayesiana. No método estatístico, utilizou-se o Campo Aleatório Condicional. Entre todos os testes, a Árvore de Decisão obteve o melhor resultado com acurácia próxima a 70%.

Reddy et al. (2010) analisou trabalhos anteriores e observou que até então vários haviam tentado identificar meios de transporte através de triangulação de sinal de celular, GPS ou acelerômetro, mas nenhum tentou combinar as informações de GPS e acelerômetro. Estes autores argumentam que usando dados de apenas um destes sensores pode haver grandes perdas na qualidade do resultado final: por exemplo, apenas acelerômetro pode não ser suficiente para diferenciar se o

usuário está parado, de bicicleta ou em um veículo motorizado; e apenas GPS pode não ser suficiente para diferenciar se o usuário está andando, correndo ou de bicicleta. Com base nisso, a solução destes autores envolveu combinar os dados destes dois sensores, e em seguida testar vários algoritmos de classificação para encontrar aquele que gera os melhores resultados. Do GPS, a única informação utilizada é a velocidade. Do acelerômetro, calcula-se um valor que representava a magnitude da aceleração, e com base nele diversos outros dados foram computados, como média e variância. Os algoritmos testados foram: Árvore de Decisão, *K-Means*, *Naive Bayes*, *Nearest Neighbor*, Máquina de Vetores de Suporte e Modelo Oculto de Markov contínuo. Adicionalmente, o Modelo Oculto de Markov discreto seria aplicado ao melhor resultado, para observar se este poderia ser melhorado. Ao final dos testes, observou-se que todos os algoritmos tiveram ótimo desempenho, comprovando que a combinação de GPS e acelerômetro realmente funcionava da maneira esperada. O melhor algoritmo foi a Árvore de Decisão, que teve acurácia média de 91,3%. Ao aplicar o Modelo Oculto de Markov discreto sobre a Árvore de Decisão a acurácia média subiu para 93,7%.

Wang, Chen e Ma (2010) tentam identificar se o usuário está parado, andando, de bicicleta, carro, ônibus ou metrô usando apenas acelerômetro. As três componentes da aceleração são processadas e diversos dados são computados a partir delas, como média, desvio padrão e componentes de frequência. Estes dados são então utilizados em três algoritmos: Árvore de Decisão, *Nearest Neighbor* e Máquina de Suporte à Decisão. Ao final, percebe-se que o melhor algoritmo foi a Árvore de Decisão, cuja acurácia variou entre 56% e 70%.

Li et al. (2011) também tentam identificar os meios de transporte que o usuário utiliza ao longo de uma trajetória. Para isso, utilizam GPS e acelerômetro. O GPS serve para identificar a velocidade com que o usuário se move. O valor coletado é convertido para uma escala de velocidade alta, média, baixa ou zero. O acelerômetro reconhece a velocidade com que o usuário está movimentando seu corpo. A cada leitura do valor das três componentes de aceleração, os valores são comparados com os da leitura anterior e o resultado é a classificação da aceleração em alta ou baixa. Finalmente, uma matriz de decisão que

relaciona velocidade e aceleração aponta qual o meio de transporte utilizado: parado, andando, bicicleta, ônibus ou carro. Para diferenciar carro e ônibus os autores identificam previamente a localização de todos os pontos de ônibus (ou seja, usam dados históricos sobre isso). Assim, se um usuário começa e termina um segmento de trajetória em pontos de ônibus e se locomove na velocidade de um carro, considera-se que está em um ônibus.

Xu et al. (2011) diferem-se dos outros trabalhos, pois usam apenas um Modelo Oculto de Markov em conjunto com métodos probabilísticos para tentar identificar se um usuário estava a pé, de bicicleta ou em um veículo motorizado. Além disso, precisam apenas das informações do GPS e alguns dados históricos. Duas teorias de probabilidade são utilizadas. A principal teoria diz que a movimentação em condições normais é guiada por certas leis, e por isso ela é baseada na Lei de Distribuição de Velocidade. Esta teoria é válida até mesmo quando existe um pouco de congestionamento, porém se o congestionamento se tornar maior ela se torna inválida. Neste caso, os autores utilizam a Teoria da Perspectiva Cumulativa, que faz uso dos dados históricos para configurar melhor as probabilidades e decidir o tipo de meio de transporte. Ao final, os resultados mostraram que esta técnica teve uma acurácia que variou entre 69% e 89%, dependendo do teste realizado.

Para resumir as informações apresentadas nesta seção, a Tabela 01 e a Tabela 02 comparam os trabalhos revisados. Na Tabela 01, a partir de valores fornecidos pelos autores (os algoritmos não foram implementados para fazer testes comparativos), observa-se que a solução usando Árvore de Decisão construída por Reddy et al. (2010) alcançou os melhores resultados. Isso mostra que, atualmente, uma das melhores soluções para descobrir o meio de transporte é combinando dados de GPS e acelerômetro. O uso de Árvore de Decisão também se mostrou favorável nos trabalhos de Zheng et al. (2008) e Wang, Chen e Ma (2010). Sendo assim, esta solução foi adotada e implementada neste trabalho.

Tabela 01 – Solução usada por cada trabalho e resultado obtido

Trabalho	Sensores utilizados	Método	Taxa de acerto
Mohan et al. (2008)	Acelerômetro	Heurísticas	Não informado
Zheng et al. (2008)	GPS	Árvore de Decisão	±70%
Reddy et al. (2010)	GPS e acelerômetro	Árvore de Decisão	91,3%
		Árvore de Decisão e Modelo Oculto de Markov discreto	93,7%
Wang et al. (2010)	Acelerômetro	Árvore de Decisão	56%~70%
Li et al. (2011)	GPS, acelerômetro e dados históricos	Matriz de Decisão	±80%
Xu et al. (2011)	GPS e dados históricos	Modelo Oculto de Markov e métodos probabilísticos	69%~89%

Fonte: produção do próprio autor

Já na Tabela 02, percebe-se que cada trabalho tentou identificar meios de transporte diferentes. Alguns trabalhos utilizaram categorias mais genéricas, como apenas “a pé” ou “veículo motorizado”. Já outros foram mais detalhistas e tentaram classificar em categorias mais específicas, como dividir “a pé” em “andando” e “correndo” ou dividir “veículo motorizado” em “carro”, “ônibus” e até “metro”.

No caso de Reddy et al. (2010), além de identificar veículos motorizados e bicicleta, eles tentam diferenciar entre pessoas andando ou correndo. Este nível de detalhamento não é necessário no presente trabalho. Por conta desta diferença, a Árvore de Decisão a ser gerada será baseada naquela de Reddy et al. (2010), mas não seguirá fielmente as mesmas especificações.

Tabela 02 – Meios de transporte identificados por cada trabalho

Trabalho	Meios de transporte					
	A pé	Bicicleta	Veículo motorizado			
Mohan et al. (2008)	X		X			
Zheng et al. (2008)	X	X	Carro	Ônibus		
			X	X		
Reddy et al. (2010)	Andando	Correndo	X	X		
	X	X				
Wang et al. (2010)	X	X	X	Carro	Ônibus	Metro
				X	X	X
Li et al. (2011)	X	X	X	Carro	Ônibus	
				X	X	
Xu et al. (2011)	X	X	X	X		

Fonte: produção do próprio autor

2.1.1 Solução escolhida

A Árvore de Decisão foi gerada no WEKA, um kit de ferramentas para mineração de dados que implementa diversos algoritmos de aprendizado de máquina (WEKA, 2013). Dado um conjunto de características extraídas dos dados coletados, o WEKA identifica quais são as mais relevantes para classificar o meio de transporte e gera a Árvore de Decisão com elas.

Nesse contexto, o seguinte conjunto de características é enviado ao WEKA: velocidade, magnitude média, variância da magnitude, energia da amostra, e coeficientes energéticos entre 1 e 10Hz. Destas, só a velocidade é extraída do GPS. As demais são obtidas do acelerômetro, e usam como base a magnitude do vetor força, calculada como $A_{mag} = \sqrt{(A_X)^2 + (A_Y)^2 + (A_Z)^2}$ (REDDY et al., 2010). Nesta equação, A_X , A_Y e A_Z são cada uma das três componentes (X, Y e Z) fornecidas pelo acelerômetro. A energia da amostra e os coeficientes energéticos são calculados via Transformada Discreta de Fourier.

A avaliação de quais características são mais relevantes para a criação da Árvore de Decisão é feita com um método baseado em correlações (CFS, do inglês *Correlation Based*

Feature Selection). Já a Árvore é construída com o algoritmo J48 com poda reduzida. Do conjunto de dados utilizado, 30% treinou a Árvore de Decisão e o restante serviu para testá-la.

De modo geral, a Árvore foi construída com base nas especificações de Reddy et al. (2010). Contudo, diferenças no contexto do trabalho de Reddy et al. (2010) e do presente trabalho resultaram em algumas alterações.

Conforme comentado anteriormente, Reddy et al. (2010) é mais detalhista que o presente trabalho em relação às possíveis classificações de meio de transporte. Além de classificar como veículo motorizado ou bicicleta, eles também tentam diferenciar entre pessoa a pé andando ou correndo. Isso é desnecessário para gerar mapas rodoviários, por isso não foi feito no presente trabalho. Devido a essa simplificação, o tamanho da amostra foi alterado. Enquanto Reddy et al. (2010) usam amostras de um segundo, o presente trabalho usa amostras de dois segundos ($tam_{amostra} = 2$, ou seja, 2 pontos e aproximadamente 40 informações de aceleração). Outra alteração foi nas características selecionadas para gerar a Árvore de Decisão. Em Reddy et al. (2010) as características mais relevantes foram velocidade, variância da magnitude e coeficientes energéticos entre 1 e 3Hz. Já no presente trabalho, foram velocidade, variância da magnitude e energia da amostra. Isso ocorre porque os coeficientes energéticos são indicados para descobrir se uma pessoa está andando ou correndo (REDDY et al., 2010). Como este trabalho não faz essa diferenciação, a importância destas características é reduzida.

Uma última diferença com relação ao trabalho de Reddy et al. (2010) é que o presente trabalho preferiu não implementar o Modelo Oculto de Markov discreto, apenas a Árvore de Decisão. Isso foi feito porque Reddy et al. (2010) demonstrou que sua solução alcança resultados de alta precisão mesmo usando apenas a Árvore de Decisão.

A Árvore de Decisão foi gerada com base em um subconjunto dos dados coletados para teste, contendo cerca de 10 horas de dados a pé, 23 horas de carro e 4 horas de bicicleta.

O Quadro 01 mostra a matriz de confusão da Árvore de Decisão. Nela se observa, por exemplo, que 10383 amostras de veículos motorizados foram classificadas corretamente. Porém, 591 amostras foram classificadas erradas (falsos negativos) –

233 como bicicleta e 358 como a pé. Além disso, 248 amostras de bicicleta e 217 amostras de a pé foram classificadas erroneamente como veículo motorizado (falsos positivos).

Quadro 01 – Matriz de Confusão

		Classificado como		
		Bicicleta	Veículo motorizado	A pé
Realidade	Bicicleta	2339	248	103
	Veículo motorizado	233	10383	358
	A pé	100	217	5827

Fonte: produção do próprio autor

A partir dos dados da matriz de confusão se pode calcular a precisão e revocação da classificação. A precisão indica, dentre todas as amostras classificadas em determinada categoria (somatório dos valores na coluna), qual a percentagem classificada corretamente. Usando como exemplo os valores relacionados aos veículos motorizados, o cálculo é $10383/(248 + 10383 + 217)$. Já a revocação indica, dentre todas as amostras que na realidade são de determinada categoria (somatório dos valores na linha), qual a percentagem classificada corretamente. No caso dos veículos motorizados, o cálculo é $10383/(233 + 10383 + 358)$.

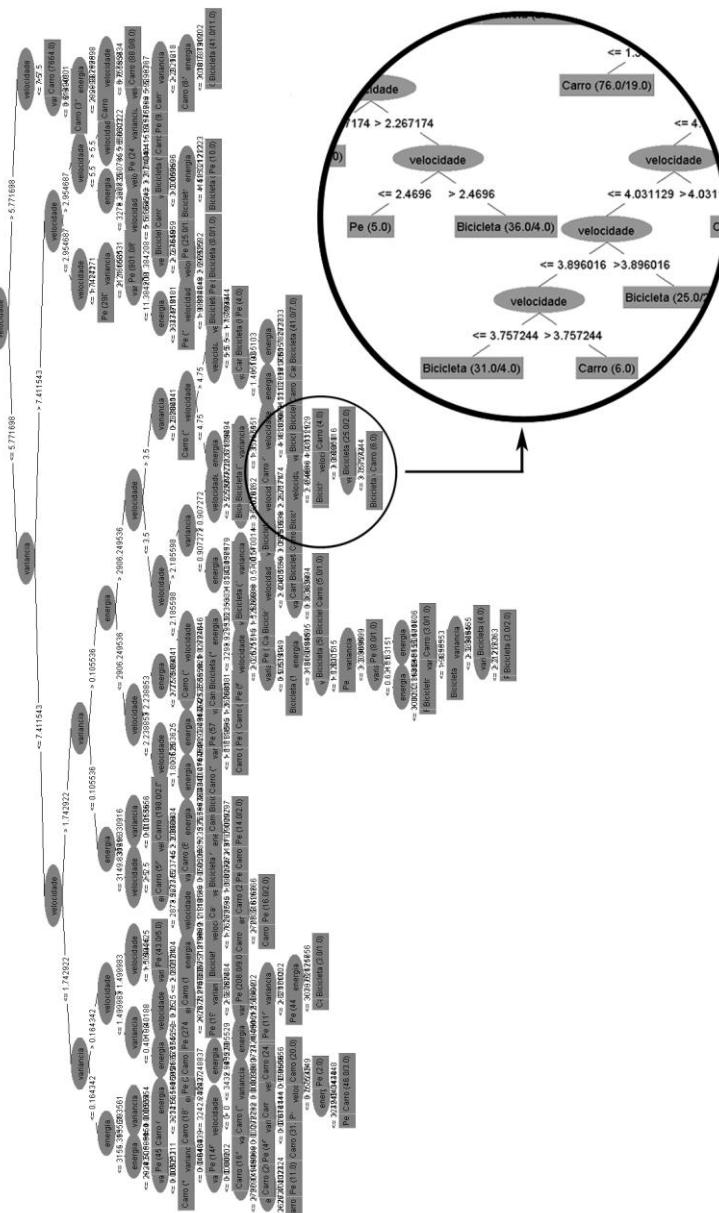
A Tabela 03 mostra estes percentuais de acerto, e a Figura 02 mostra a Árvore propriamente dita. Devido ao seu tamanho, a intenção da figura é apenas ilustrar sua estrutura. Finalmente, a Figura 03 apresenta resultados do uso desta Árvore de Decisão.

Tabela 03 – Percentuais de acerto da Árvore de Decisão

Meio de transporte	Precisão	Revocação
Bicicleta	87,5%	87%
Veículo motorizado	95,7%	94,6%
A pé	92,7%	94,8%
Média ponderada	93,7%	93,6%

Fonte: produção do próprio autor

Figura 02 – Árvore de Decisão



Fonte: produção do próprio autor

Figura 03 – Identificação de meios de transporte



Fonte: produção do próprio autor

A Figura 03 confirma os resultados exibidos no Quadro 01: a maioria das amostras foi classificada corretamente. Como mostra a Tabela 03, a taxa de precisão na classificação das amostras foi de 93,7%, enquanto a taxa de revocação foi 93,6%. Com base nestes valores, a conclusão a que se chega é que este método pode ser aplicado na identificação do meio de transporte das trajetórias coletadas para a geração de mapas rodoviários.

2.2 Redução do ruído

Independentemente da origem dos ruídos, eles reduzem a qualidade interna dos dados espaciais (DEVILLERS; JEANSOULIN, 2006) e, portanto, devem-se adotar estratégias para tentar mitigar seus efeitos no processamento das trajetórias.

No contexto da criação de mapas rodoviários, dois aspectos são relevantes para compensar a falta de acurácia dos receptores GPS: (i) devem ser utilizadas várias trajetórias passando pela mesma via e (ii) a frequência de coleta dos

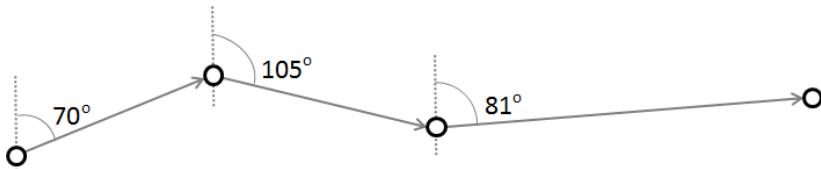
pontos de cada trajetória deve ser alta. Considerando que uma grande quantidade de pontos será analisada em cada via, assume-se que o ruído existente em vários deles é compensado pela boa acurácia dos demais. Neste caso, os pontos problemáticos podem ser simplesmente descartados. Esta etapa de limpeza das trajetórias coletadas pode ser comparada à etapa de limpeza no processo de Extração, Transformação e Carga (ETL, do inglês *Extract, Transform, Load*) de um *Data Warehouse*.

É possível identificar e filtrar ruído com base nos dados de cada ponto coletado. Em especial, destaca-se a limpeza das trajetórias usando os dados de acurácia, ângulo de movimento, velocidade e proximidade dos pontos, assim como comentado na seção 2.5.6.

A acurácia é um valor em metros que representa o raio de um círculo centrado no ponto coletado. Este círculo simboliza o erro máximo das coordenadas do ponto: quanto menor o raio, melhor a acurácia. Assim, filtrar pela acurácia significa descartar pontos com maior chance de erro, que poderiam prejudicar a qualidade do mapa. Isso envolve definir uma heurística que elimine pontos com acurácia considerada ruim. Essa heurística também pode ser aplicada para grupos de pontos, analisando a acurácia média. Isso é interessante para identificar pontos que dizem ter acurácia boa (raio pequeno), mas na realidade são ruins (raio grande). Ou seja, pontos com informações incorretas.

O ângulo de movimento do usuário é um valor em graus, calculado a partir das coordenadas de dois pontos consecutivos. Ela representa a angulação com que o usuário se movimentou de um ponto ao outro. A Figura 04 exemplifica este cálculo.

Figura 04 – Ângulo de movimento



Fonte: produção do próprio autor

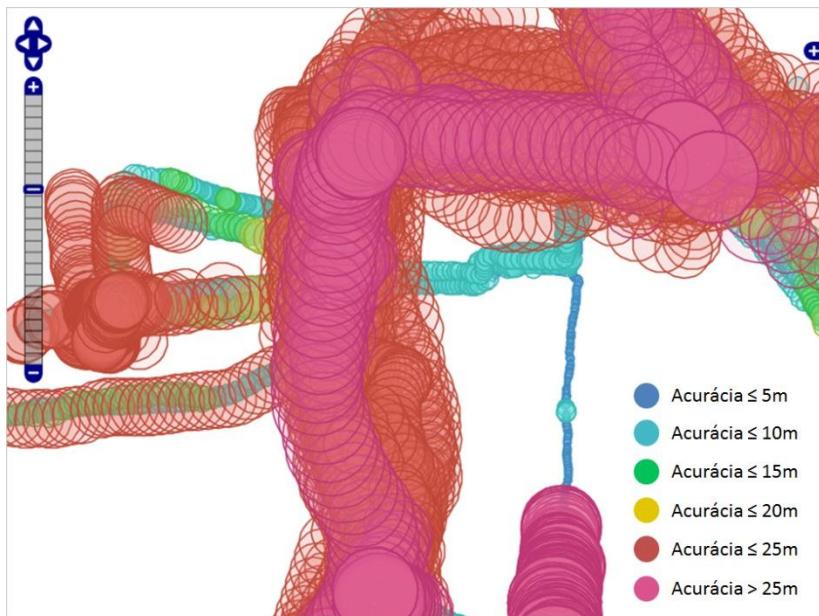
Na filtragem, analisa-se o ângulo de movimento de um ponto em relação aos adjacentes. Se a diferença ultrapassar um limite ele é descartado, pois representaria um movimento irreal para um carro, como uma curva excessivamente fechada.

O critério da velocidade é utilizado para eliminar os pontos coletados nos momentos que o veículo não está em movimento, como por exemplo, ao parar em um semáforo. De forma semelhante, o critério da proximidade dos pontos permite reduzir a quantidade de pontos em locais onde o veículo se locomove lentamente, devido a engarrafamentos ou por qualquer outro motivo. Estes pontos são considerados de pouca relevância para a geração de mapas, e por isso são descartados.

2.2.1 Solução escolhida

Para observar os efeitos da filtragem foram realizados testes sobre o cenário apresentado na Figura 05.

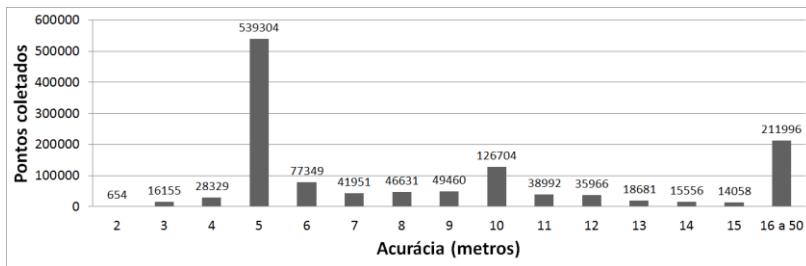
Figura 05 – Qualidade dos dados coletados



Fonte: produção do próprio autor

Cada ponto é representado pelo seu círculo de acurácia. Como se pode perceber, a acurácia tem grande influência na qualidade dos pontos das trajetórias. Nesse sentido, a Figura 06 mostra na forma de gráfico de colunas a quantidade de pontos coletados por valor de acurácia. Como é possível visualizar, os receptores GPS alcançam uma boa acurácia. A grande maioria dos pontos têm acurácia de cinco metros, e em raros momentos se conseguiu coletar pontos com acurácia de até dois metros. Ao todo, 83% dos pontos estão concentrados no intervalo de dois a quinze metros de acurácia. Acima deste valor, considera-se que os pontos possuem um erro consideravelmente grande e, portanto, é preferível descartá-los ao invés de usá-los e correr o risco de piorar a qualidade dos centros das vias.

Figura 06 – Quantidade X qualidade dos pontos coletados



Fonte: produção do próprio autor

Além de descartar qualquer ponto com acurácia maior que $a_{lim} = 15$ metros, também é necessário considerar os casos onde o dado do GPS está incorreto, informando uma acurácia melhor que a real, que não condiz com as coordenadas do ponto. Conforme observado durante a execução de testes preliminares, o GPS pode retornar vários pontos seguidos com acurácia ruim e, arbitrariamente, informar um ponto com acurácia boa, mas localizado longe de qualquer via, o que não seria possível. Uma forma de identificar pontos assim é através de uma heurística que calcule a acurácia média de um grupo de pontos, conforme já mencionado nesta seção. Usando essa estratégia foram considerados como ruído qualquer grupo de $tam_{grupos} = 20$ pontos com acurácia média maior que $a_{lim_{media}} = 15$ metros. Os parâmetros tam_{grupos} e $a_{lim_{media}}$ foram escolhidos

empiricamente, porém substancialmente refinados ao longo dos ciclos de iteração do processo baseado na metodologia de Pesquisa-Ação.

Outra forma de filtragem é a filtragem por ângulo de movimento. Se a diferença de ângulo entre dois pontos sequenciais for muito grande, pode ser que eles sejam pontos classificados erroneamente como sendo de um veículo motorizado. Sendo assim, considera-se como ruído os pontos que sofrem uma modificação na angulação maior que $\theta_{lim} = 25$ graus. Assim como os outros parâmetros, este também foi escolhido empiricamente.

Também foram descartados pontos com velocidade igual a zero e conjuntos de pontos muito próximos, com distância de até $d_{próximo} = 3$ metros, pois representam momentos onde o veículo estaria parado. Mais informações sobre os parâmetros utilizados estão disponíveis no apêndice B. A Figura 07 mostra as mesmas trajetórias exibidas na Figura 05, após a filtragem.

Figura 07 – Filtragem das trajetórias



Fonte: produção do próprio autor

Como se observa, houve uma melhora na identificação do percurso das vias. Isso significa que uma grande quantidade de pontos foram descartados – neste caso, cerca de 42% do total. Deste percentual, 17% representam os pontos com acurácia maior que a_{lim} (15 metros) e o restante são os pontos eliminados nas outras etapas, como filtragem por acurácia média e velocidade igual a zero. Ou seja, os pontos removidos são aqueles que estavam reduzindo a qualidade dos dados. Como os pontos que sobraram são aqueles considerados de qualidade mais alta, o desempenho do algoritmo gerador de mapas melhora, pois há menos ruído para confundi-lo.

2.3 Compressão das trajetórias

Para que uma trajetória represente um esboço fiel das vias percorridas no mundo real o receptor GPS deve coletar dados em intervalos de poucos segundos, o que acarreta em um volume grande de dados (MERATNIA; de BY, 2004). Processar todos estes dados é uma tarefa custosa, que demanda muitos recursos computacionais e tempo. Sendo assim, reduzir a quantidade de pontos que compõem cada trajetória pode tornar o método para geração de mapas mais rápido.

Algoritmos para compressão são classificados como sem perda ou com perda. Algoritmos sem perda compactam um arquivo de forma que, após a descompactação, os dados originais são recuperados. Já os algoritmos com perda diminuem o tamanho do arquivo através do descarte de informações (PU, 2006). No caso deste trabalho, como a acurácia do GPS é variável, em muitas situações a trajetória pode ser mais sinuosa que o trajeto real. Esta situação fica evidenciada em vias retas. Deste modo, acredita-se que usar um algoritmo de compressão com perda para eliminar pontos considerados menos significativos pode reduzir os efeitos da imprecisão do GPS e contribuir na geração de mapas fiéis à realidade.

Existem diversos algoritmos de compressão com perda dedicados à dados georreferenciados. Deve-se então identificar aquele que melhor comprima os dados sem deformar perceptivelmente a trajetória.

Antes de comparar os algoritmos que mais se destacam na literatura, é importante analisar as métricas que eles utilizam

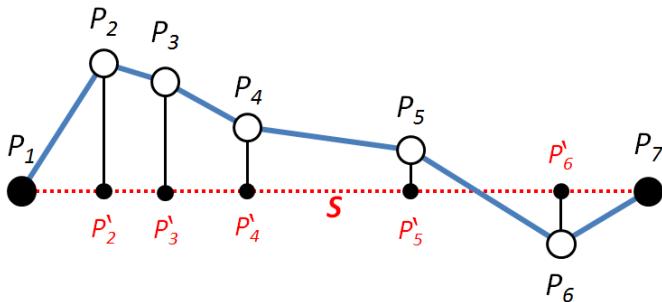
para determinar a deformação provocada na trajetória. Assim, a seção 2.3.1 apresenta estas métricas e suas características. Já a seção 2.3.2 apresenta os algoritmos e faz uma comparação dos que mais se adequam ao propósito deste trabalho.

2.3.1 Métricas

A função da métrica é informar qual será o erro inserido na trajetória caso um determinado ponto seja removido. Dentre os algoritmos encontrados, as principais métricas utilizadas são Distância Perpendicular, Distância Proporcional ao Tempo (do inglês *Time-Ratio Distance*) e Distância Euclidiana Sincronizada (do inglês *Synchronous Euclidean Distance*). Em todas elas, dois elementos são essenciais: um ponto P qualquer que faz parte da trajetória, mas que ainda não foi selecionado para permanecer na trajetória; e os segmentos de reta S , formados pelos pontos já selecionados. O erro é calculado com base na distância entre o ponto P e o segmento S mais próximo. Porém, o ponto em S utilizado para calcular essa distância é o que diferencia estas três métricas.

Para o cálculo de métrica Distância Perpendicular é escolhido um ponto do segmento S que, em conjunto com o ponto P , formem uma reta perpendicular a S , conforme apresentado na Figura 08. O comprimento da reta perpendicular representa o erro. Nesta figura, considera-se que o segmento S é formado a partir dos pontos P_1 e P_7 .

Figura 08 – Métrica Distância Perpendicular



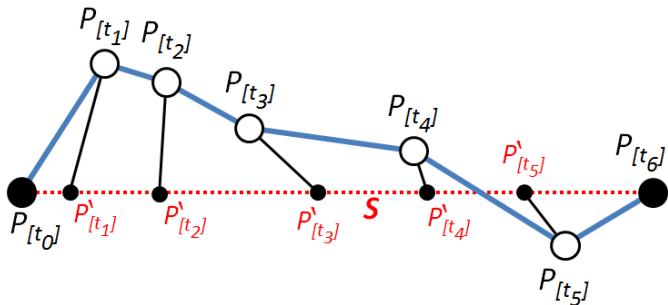
Fonte: produção do próprio autor

Após obter este valor, um parâmetro que define o erro máximo serve de critério para decidir se o ponto P será ou não selecionado. Como se observa, a Distância Perpendicular apenas leva em consideração dados espaciais. Por este motivo, a qualidade destas informações é preservada, o que é ideal para cartografia (HERSHBERGER; SNOEYINK, 1992), mas a tendência é que sejam introduzidos erros em outros dados, como data e ângulo de movimento (MUCKELL et al., 2011).

Já na Distância Proporcional ao Tempo a escolha do ponto é feita levando em consideração também as informações temporais dos elementos envolvidos. Este processo consiste em, primeiramente, verificar quanto tempo decorreu entre os dois pontos que deram origem a S e entre um destes pontos e P . Com base nestas informações, é possível prever a localização do ponto em S que corresponde ao mesmo instante de tempo em que P foi gravado. Finalmente, este ponto é usado no cálculo da distância entre P e S , como demonstrado na Figura 09. Novamente, um parâmetro de erro máximo é usado para decidir se P será selecionado. Nesta figura, considera-se que o segmento S é formado a partir dos pontos $P_{[t_0]}$ e $P_{[t_6]}$.

Ao usar também as informações temporais, estes dados são melhor preservados após a compressão. Nesse sentido, Meratnia e de By (2004) dizem ser necessário preservar estes dados, visto que as trajetórias GPS representam o histórico de movimentação do objeto móvel ao longo do tempo.

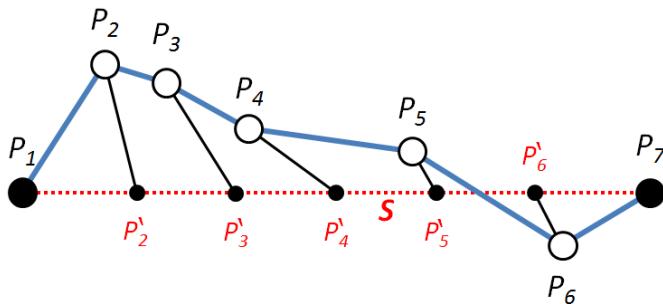
Figura 09 – Métrica Distância Proporcional ao Tempo



Fonte: produção do próprio autor

Finalmente, a Distância Euclidiana Sincronizada escolhe n pontos em S , onde n é o número de pontos da trajetória que se localizam temporalmente entre os pontos que formam S . Estes n pontos são escolhidos de tal forma que dividam S em $n+1$ intervalos de mesmo comprimento, como pode ser visto na Figura 10. A distância de cada ponto da trajetória até seu correspondente em S é o erro calculado. Nesta figura, considera-se que o segmento S é formado a partir dos pontos P_1 e P_7 .

Figura 10 – Métrica Distância Euclidiana Sincronizada



Fonte: produção do próprio autor

Assim como foi afirmado a respeito da Distância Proporcional ao Tempo, o uso da métrica Distância Euclidiana Sincronizada é indicado por Muckell et al. (2011) como uma alternativa a distância perpendicular, pois preserva melhor as informações temporais. Além disso, ela permite ter uma melhor noção do erro acrescentado à trajetória comprimida caso um ponto seja descartado.

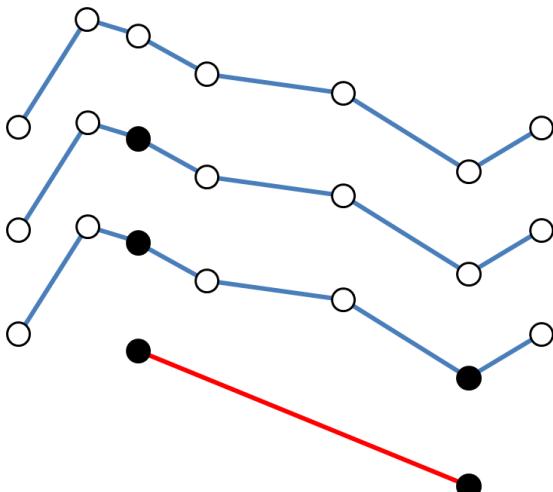
2.3.2 Algoritmos de Compressão

Dentre os algoritmos na literatura que implementam as métricas apresentadas na seção 2.3.1, destacam-se *Uniform Sampling*, *Douglas-Peucker*, *Opening Window*, *Top-Down Time-Ratio*, *Opening Window Time-Ratio*, *Spatiotemporal Trace*, *Spatial Quality Simplification Heuristic* e *Dead Reckoning*.

O algoritmo *Uniform Sampling* é uma forma de compressão com perda que não se baseia em nenhuma métrica. Dado um valor n que serve como taxa de compressão, este

algoritmo consiste em selecionar todo n -ésimo ponto da trajetória e descartar os demais (MUCKELL et al., 2011). A Figura 11 exemplifica seu funcionamento. Devido a sua simplicidade, o *Uniform Sampling* tem como vantagens ser trivial para implementar e executar em tempo linear. Vale destacar que dependendo do valor de n , o primeiro e último pontos são descartados. Isso pode ser contornado obrigando que eles sejam preservados.

Figura 11 – *Uniform Sampling* ($n=3$)

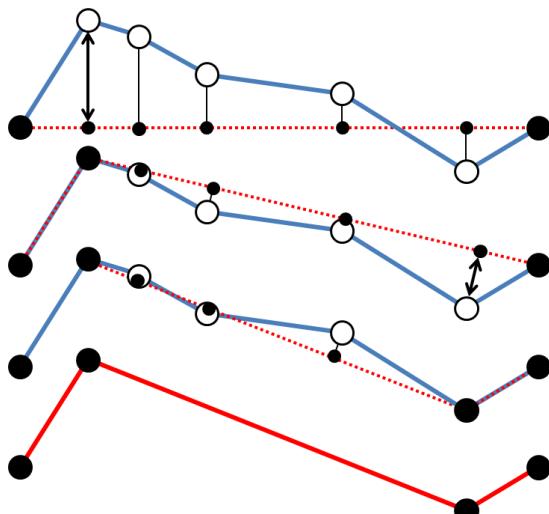


Fonte: produção do próprio autor

O algoritmo *Douglas-Peucker* foi proposto em 1973 e ainda hoje é popular (MUCKELL et al., 2010). Inicialmente, ele seleciona o primeiro e último pontos da trajetória completa e forma um segmento S entre eles. A seguir, calcula-se a distância perpendicular de todos os pontos intermediários a esta reta, e aquele mais distante – ou seja, que se fosse removido adicionaria mais erro à compressão – é incluído. Desta forma, o segmento inicial é dividido em dois segmentos contíguos: do primeiro ponto ao recém-incluído, e deste ao último. Este processo é repetido recursivamente e a cada iteração um novo ponto é selecionado e incluído no segmento, até que nenhum dos pontos restantes esteja a uma distância maior que o limite

desejado. Todos os pontos não selecionados são descartados (FRENTZOS; THEODORIDIS, 2007). A Figura 12 exemplifica seu funcionamento. A versão original do algoritmo tem complexidade $O(n^2)$, mas Hershberger e Snoeyink (1992) fizeram uma versão $O(n \log n)$.

Figura 12 – Douglas-Peucker

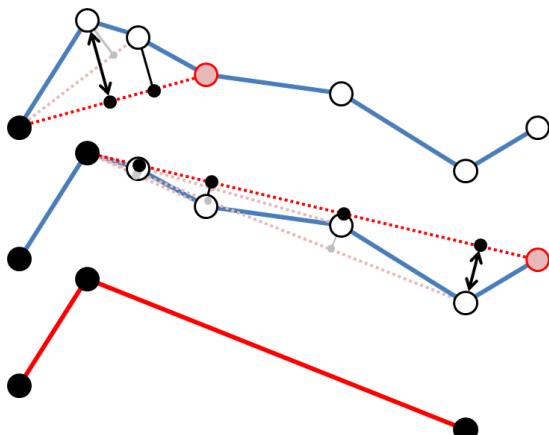


Fonte: produção do próprio autor

O *Opening Window* também usa como métrica a Distância Perpendicular. O algoritmo possui dois elementos: a âncora e a boia. Inicialmente, eles são representados pelo primeiro e terceiro pontos da trajetória, respectivamente. Verifica-se então se todos os pontos intermediários estão a uma distância perpendicular dentro do limite permitido. Se sim, então o ponto seguinte (neste caso, o quarto) se torna a boia. Este processo é repetido até que seja encontrado um ponto com distância perpendicular maior que o limite. Neste momento, percebe-se a diferença entre as duas variações deste algoritmo: no *Normal Opening Window* é selecionado o ponto com distância acima do limite, já no *Before Opening Window* é selecionado o ponto anterior à boia. Independente da variação utilizada o ponto selecionado se torna a nova âncora e o ponto duas posições à

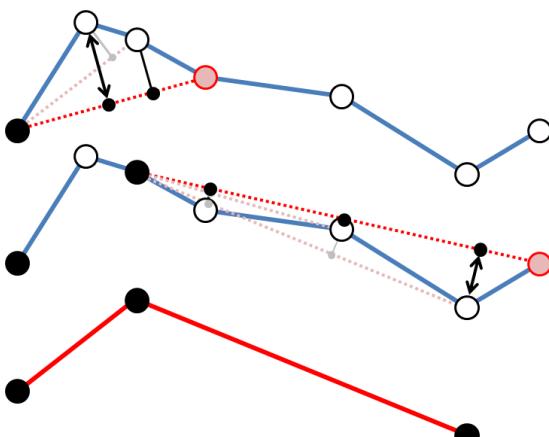
frente a nova boia. Todo este processo é repetido até o fim da trajetória. Os pontos não selecionados são descartados (FRENTZOS; THEODORIDIS, 2007; MERATNIA; de BY, 2004). A Figura 13 demonstra o funcionamento do Normal Opening Window, e a Figura 14, do Before Opening Window.

Figura 13 – *Normal Opening Window*



Fonte: produção do próprio autor

Figura 14 – *Before Opening Window*



Fonte: produção do próprio autor

De acordo com Meratnia e de By (2004), este algoritmo tem complexidade $O(n^2)$. Além disso, eles também fazem notar que o modo de operação do *Opening Window* faz com que ele descarte, pelo menos, o último ponto do percurso. Isso pode ser contornado obrigando este ponto a ser preservado.

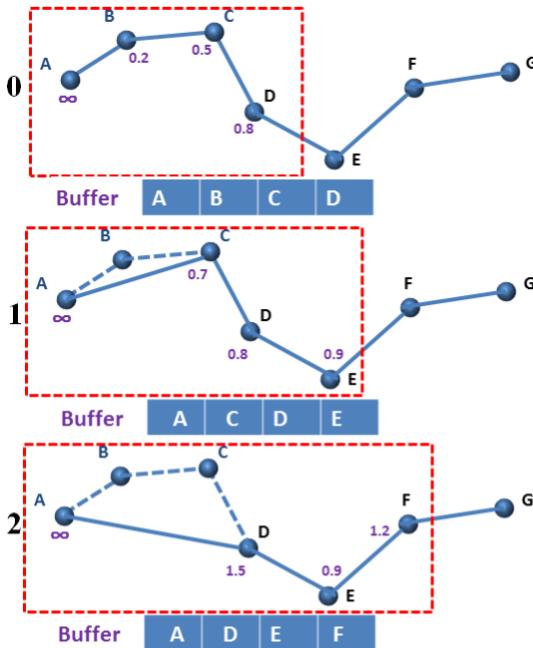
Meratnia e de By (2004) também afirmam que usar como métrica apenas a Distância Perpendicular não é o ideal para compressão de dados de GPS, pois a precisão temporal também deve ser mantida. Por este motivo eles propõem dois algoritmos, *Top-Down Time-Ratio* e *Opening Window Time-Ratio*. Estes algoritmos são o *Douglas-Peucker* e o *Opening Window*, respectivamente, mas modificados para utilizar a métrica Distância Proporcional ao Tempo. Estes algoritmos cumprem o papel de melhor preservar as informações temporais, ao mesmo tempo que tentam manter as qualidades das versões originais dos algoritmos.

O *Spatiotemporal Trace* foi criado por Potamias, Patroumpas e Sellis (2006). Ele é diferente dos algoritmos anteriores porque utiliza um *buffer* de tamanho fixo para armazenar uma quantidade limitada de pontos. A métrica utilizada é a Distância Euclidiana Sincronizada. Ao ser executado, cada ponto recebido vai sendo armazenado no *buffer* em conjunto com seu resultado na métrica, calculado com base no ponto imediatamente anterior e no seguinte. Quando o *buffer* está cheio e um novo ponto é recebido, o ponto do *buffer* com pior resultado na métrica é selecionado e comparado ao novo ponto. Se o resultado do novo ponto for melhor que o do ponto selecionado, esse é removido para dar lugar ao novo ponto. Este processo é repetido até chegar ao fim da trajetória. A complexidade deste algoritmo é $O(n^2)$ (MUCKELL et al., 2010; POTAMIAS; PATROUMPAS; SELLIS, 2006).

Já o *Spatial Quality Simplification Heuristic* foi criado por Muckell et al. (2011) e é semelhante ao *Spatiotemporal Trace*, pois também mantém um *buffer* e usa a métrica Distância Euclidiana Sincronizada, mas a forma como o algoritmo executa é diferente. Nesse, quando o *buffer* está cheio e um novo ponto é recebido, calcula-se quanto pioraria o resultado da métrica se cada ponto do *buffer* fosse removido. Desta forma, é criada uma fila de prioridade, que indica quais pontos representam mais informações. O ponto com menos informações – ou seja, que

adiciona menos erro ao *buffer* – é então removido para dar lugar ao novo ponto. O resultado da métrica é novamente calculado para cada ponto, e o resultado do ponto que foi removido é dividido e acrescentado nos seus pontos vizinhos, pois agora eles representam mais informações – não apenas as suas, mas também a do ponto removido. Ao final deste processo uma nova fila de prioridade é formada, e este processo é repetido até o fim da trajetória. A Figura 15 ilustra o funcionamento do algoritmo. A complexidade é $O(n \log(b))$, onde b é o tamanho do *buffer* (MUCKELL et al., 2011).

Figura 15 – *Spatial Quality Simplification Heuristic*

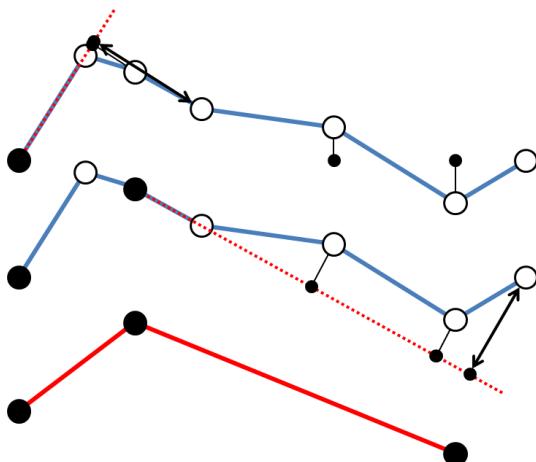


Fonte: (MUCKELL et al., 2011)

Por último, o algoritmo chamado *Dead Reckoning* é diferente dos demais porque prevê a localização de cada novo ponto baseado na localização do último ponto selecionado e no seu ângulo de movimento em relação ao ponto seguinte. Estas duas informações são usadas para gerar uma reta, que o

algoritmo acredita que os próximos pontos irão intersectar. Quando um novo ponto é coletado, a distância entre ele e a reta prevista é calculada e se ultrapassar um limite definido, o ponto anterior ao último recebido é incluído. Nesse momento, gera-se uma nova reta a partir deste ponto e o que vem a seguir, e os próximos pontos são analisados com relação à ela. Este processo se repete até o fim da trajetória (TRAJCEVSKI et al., 2006). A Figura 16 exemplifica o funcionamento do algoritmo.

Figura 16 – Dead Reckoning



Fonte: produção do próprio autor

Este algoritmo tem complexidade $O(n)$, pois basta verificar, a cada ponto, se ele ultrapassou o erro limite estabelecido (MUCKELL et al., 2011). Vale destacar que, assim como o *Opening Window*, o modo de operação do *Dead Reckoning* faz com que ele descarte um ou mais pontos do final do percurso. Novamente, isso pode ser contornado obrigando este ponto a ser preservado manualmente.

A Tabela 04 resume as informações apresentadas nesta seção. Com base nela, discorre-se a seguir a respeito de qual algoritmo será empregado no presente trabalho.

Tabela 04 – Resumo dos algoritmos de compressão de trajetória revisados

Algoritmo	Métrica	Complexidade
<i>Uniform Sampling</i>	---	$O(n)$
<i>Douglas-Peucker</i>	Distância Perpendicular	$O(n \log n)$
<i>Normal Opening Window</i>	Distância Perpendicular	$O(n^2)$
<i>Before Opening Window</i>	Distância Perpendicular	$O(n^2)$
<i>Top-Down Time-Ratio</i>	Distância Proporcional ao Tempo	$O(n \log n)$
<i>Opening Window Time-Ratio</i>	Distância Proporcional ao Tempo	$O(n^2)$
<i>Spatiotemporal Trace</i>	Distância Euclidiana Sincronizada	$O(n^2)$
<i>Spatial Quality Simplification Heuristic</i>	Distância Euclidiana Sincronizada	$O(n \log(b))$
<i>Dead Reckoning</i>	Distância Perpendicular	$O(n)$

Fonte: produção do próprio autor

Como o algoritmo de compressão *Uniform Sampling* não usa uma métrica é impossível definir um erro máximo permitido e, assim, a qualidade dos resultados varia de acordo com as características da trajetória. Por este motivo, seu uso neste trabalho foi desconsiderado.

Todos os outros algoritmos usam como métrica Distância Perpendicular, Distância Proporcional ao Tempo ou Distância Euclidiana Sincronizada. Assim, é necessário considerar qual métrica melhor atende aos objetivos deste trabalho. A geração de mapas faz parte da disciplina de cartografia, e Hershberger e Snoeyink (1992) dizem que Distância Perpendicular, por preservar principalmente as informações espaciais, é ideal para este tipo de aplicação. Adicionalmente, uma trajetória pode ser vista como uma linha, que é um objeto geométrico. Tanto Meratnia e de By (2004) como Trajcevski et al. (2006) comentam que Distância Perpendicular é a métrica ideal para aplicações preocupadas em simplesmente manter a forma geométrica

original. As outras duas métricas, além de tentar manter a distância perpendicular, também preservam aspectos temporais, mas isso significa dar menos atenção à forma geométrica. Diante disso, pode-se concluir que o algoritmo de compressão escolhido deve ser um daqueles cuja métrica utilizada seja a Distância Perpendicular.

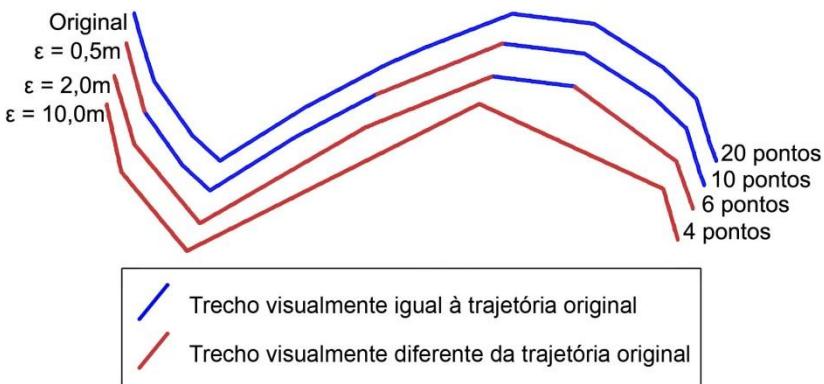
2.3.3 Solução escolhida

Dentre os algoritmos revisados, aqueles que utilizam a métrica Distância Perpendicular são: *Douglas-Peucker*, *Before Opening Window*, *Normal Opening Window* e *Dead Reckoning*. A fim de compará-los para identificar qual apresenta o melhor desempenho foi utilizado como cenário de teste a região do bairro São Marcos, em Joinville – Santa Catarina. Ao todo foram selecionadas aleatoriamente dez trajetórias de tamanhos diferentes.

Os quatro algoritmos têm um único parâmetro, que define o erro máximo permitido. Conforme explicado anteriormente (seção 2.3.1), este erro é calculado com base na distância entre um ponto P e o segmento S mais próximo. Por isso, quanto maior o erro máximo permitido, maior a variação necessária entre dois pontos para que eles sejam preservados. Se essa variação for pequena – pontos próximos ou com características semelhantes (por exemplo, pontos em linha reta) – será considerado que estes pontos são pouco significativos e, portanto, desnecessários. Na prática, isso significa que quanto maior o erro máximo permitido, mais pontos serão eliminados da trajetória. Consequentemente, maior será a taxa de compressão.

Por outro lado, quanto mais pontos forem eliminados, mais a trajetória comprimida irá perder a forma geométrica da trajetória original. A Figura 17 demonstra esse efeito. Como se pode perceber, erros de valor mais elevado comprimiram mais a trajetória, porém, é nitidamente perceptível a degradação da semelhança visual em relação à trajetória original.

Figura 17 – Resultado da compressão para diferentes valores de erro máximo permitido (ϵ)



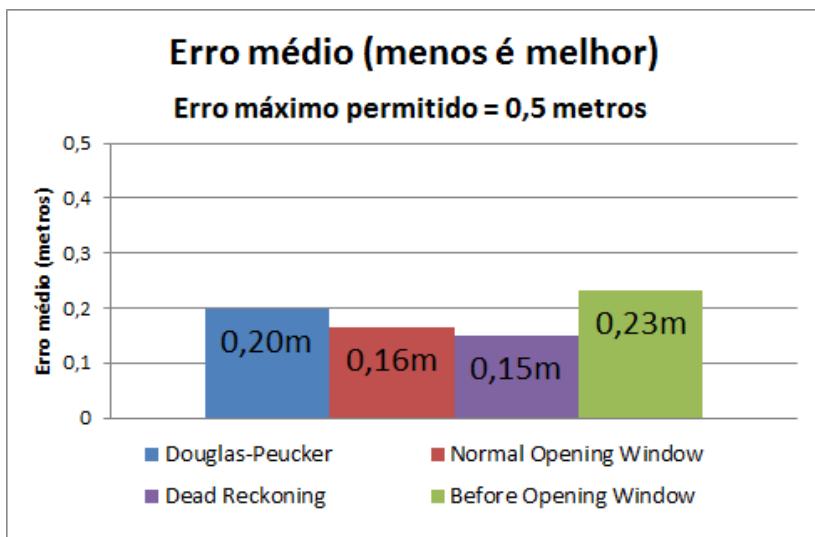
Fonte: produção do próprio autor

Com base nestas observações, o valor de erro máximo permitido no presente trabalho é de $\epsilon = 0,5$ metros. Sendo assim, os testes comparativos devem analisar os quatro algoritmos em relação a este valor.

Com base na literatura, três medidas foram coletadas para servir de critérios para comparar o desempenho dos algoritmos: média de erro introduzido na trajetória (CHEN; XU; FRANTI, 2012; MUCKELL et al., 2011), percentagem de compressão (MERATNIA; de BY, 2004) e tempo de execução (CHEN; XU; FRANTI, 2012; FRENTEZOS; THEODORIDIS, 2007). Adicionalmente, coletou-se também o maior erro introduzido na trajetória. Vale destacar que para reduzir interferências que impactem no tempo de execução, cada teste foi executado 100 vezes e o tempo calculado é a média de todas as execuções.

A Figura 18 apresenta os resultados obtidos no critério de erro médio introduzido na trajetória para o erro máximo permitido ϵ , definido anteriormente como 0,5 metros.

Figura 18 – Gráfico de erro médio introduzido na trajetória após compressão

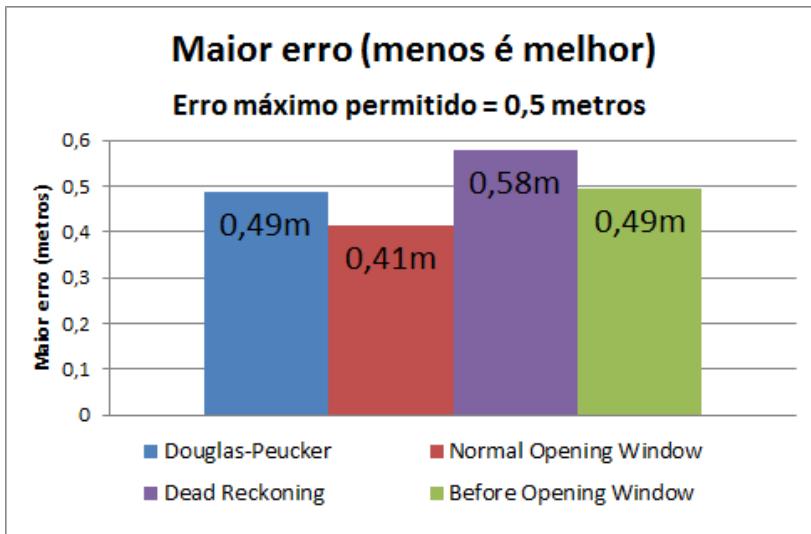


Fonte: produção do próprio autor

Dead Reckoning e *Normal Opening Window* mostraram um erro médio pouco menor que os demais algoritmos. O *Before Opening Window* teve o maior erro médio, porém, ainda assim é um valor abaixo da metade do máximo permitido. De modo geral, a diferença entre os algoritmos é na ordem de centímetros e, portanto, pode ser desconsiderada.

Conforme a Figura 19 mostra, o gráfico de maior erro encontrado é similar ao gráfico de erro médio. Contudo, nota-se um problema no algoritmo *Dead Reckoning*, pois apesar do erro médio ser bom, em alguns casos específicos o erro inserido na trajetória é maior que o erro máximo permitido ε .

Figura 19 – Gráfico de maior erro introduzido na trajetória após compressão

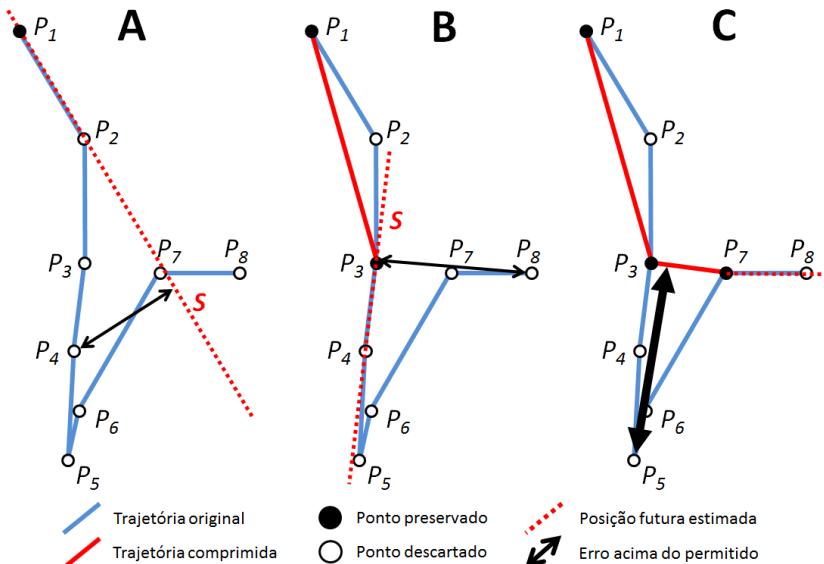


Fonte: produção do próprio autor

Acredita-se que isto ocorre porque, neste algoritmo, o segmento S (que como explicado na seção 2.3.1, é usado para calcular o erro dos pontos da trajetória) não representa um estado intermediário da compressão (ou seja, como seria o trecho da trajetória se apenas dois pontos fossem preservados), como é o caso nos outros algoritmos. No *Dead Reckoning*, o segmento S representa uma possível posição futura. Quando um ponto é preservado a posição futura estimada é modificada, e, com isso, pontos que estavam próximos de S podem ficar longe da trajetória comprimida. A Figura 20 exemplifica tal situação. Inicialmente (Figura 20.A) a posição estimada (S) é baseada em P_1 e P_2 . O ponto P_4 ultrapassa o erro máximo permitido, então na Figura 20.B P_3 é preservado e a posição estimada passa a se basear em P_3 e P_4 . Os pontos P_5 e P_6 provocam pouco erro e por isso não são preservados. O ponto P_8 ultrapassa o erro máximo permitido, então na Figura 20.C P_7 é preservado. Assim, a forma comprimida deste trecho da trajetória contém os pontos P_1 , P_3 e P_7 . Contudo, o ponto P_5 insere na trajetória um erro maior que o permitido, gerando o comportamento apresentado no gráfico da

Figura 19. Devido a este problema, o *Dead Reckoning* não é usado neste trabalho.

Figura 20 – Problema do *Dead Reckoning*

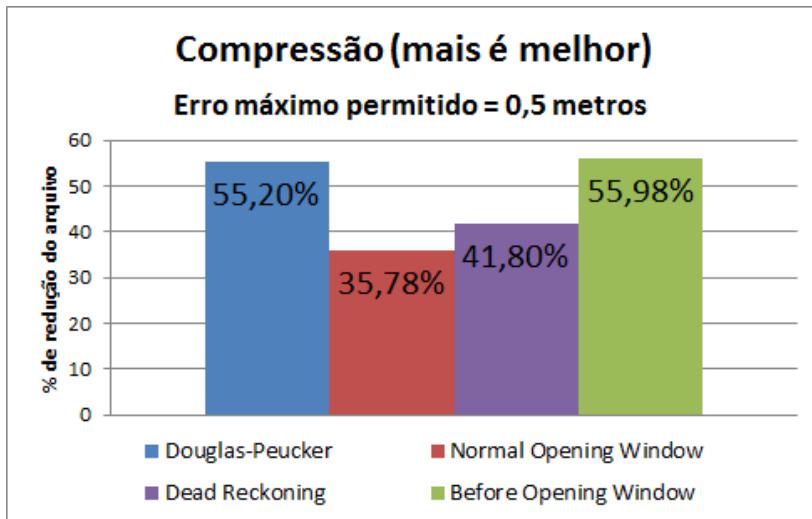


Fonte: produção do próprio autor

Com relação à taxa de compressão das trajetórias, o gráfico da Figura 21 mostra que o *Normal Opening Window* teve os piores resultados, enquanto que o *Before Opening Window* teve uma percentagem de compressão pouco melhor que a do *Douglas-Peucker*.

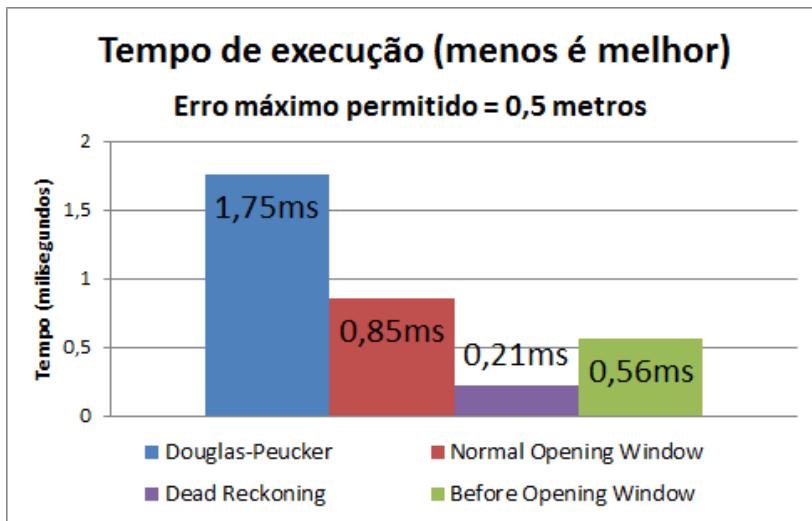
Já a Figura 22 mostra os tempos de execução. Percebe-se que *Dead Reckoning* é o mais rápido por ter complexidade $O(n)$ (porém, conforme explicado, este algoritmo não será usado). Além disso, as duas variantes do *Opening Window* tendem a ser mais rápidas que o *Douglas-Peucker* para o erro máximo permitido (ε definido (0,5 metros).

Figura 21 – Gráfico de percentagem de pontos mantidos



Fonte: produção do próprio autor

Figura 22 – Gráfico de tempo de execução



Fonte: produção do próprio autor

A compressão reduz a quantidade de pontos a serem processados nas etapas a seguir. Contudo, para a criação de mapas, é importante que o erro máximo permitido seja pequeno, para balancear entre compressão do arquivo e preservação da forma geométrica. Por isso, dentre os algoritmos testados, o *Before Opening Window* demonstra ser o mais indicado. Além de ser veloz no erro máximo permitido ε configurado (0,5 metros), é o que demonstrou as melhores taxas de compressão e é comparável aos demais algoritmos em relação ao erro médio e maior erro.

Após a compressão, as trajetórias são armazenadas em um banco de dados com suporte a dados geográficos e ficam disponíveis para a etapa de identificação dos centros das vias.

2.4 Identificação dos centros das vias usando algoritmo evolutivo

A ideia de resolver problemas usando sistemas computacionais inspirados na evolução natural das espécies surgiu na década de 1950. Diversos pesquisadores começaram a estudar este assunto de maneira independente, porém todos se baseando no mesmo princípio de evolução de uma população de soluções candidatas a partir de seleção natural e variação genética (MITCHELL, 1999).

Essas pesquisas levaram ao desenvolvimento de três abordagens diferentes, que servem de base para a área de Computação Evolutiva: algoritmos genéticos, estratégias evolutivas e programação evolutiva. Com o avanço das pesquisas novas abordagens foram apresentadas, como os algoritmos evolutivos para otimização multiobjetivo (GABRIEL; DELBEM, 2008).

A seção 2.4.1 revisa a terminologia usada nos algoritmos evolutivos. Na sequência, a seção 2.4.2 descreve algumas das principais abordagens de algoritmos evolutivos. Por fim, a seção 2.4.3 discute o uso de algoritmos evolutivos como estratégia para encontrar o centro das vias.

2.4.1 Fundamentos biológicos e terminologia

Nas subseções a seguir é apresentada a origem biológica de cada um dos principais termos usados pelos algoritmos

evolutivos, bem como uma explicação de como eles são implementados.

2.4.1.1 Cromossomos, genes e alelos

Na biologia, um organismo é concebido e se desenvolve conforme especificado por seus *cromossomos*. Estas estruturas contêm as informações que irão configurar as características biológicas do ser vivo. Dentro de uma mesma espécie todos os organismos possuem cromossomos de mesmo formato, porém cada organismo é configurado de maneira diferente (BECKET, 1986).

Um cromossomo é formado por unidades menores chamadas *genes*. Cada um deles contém as informações necessárias para determinar uma das características do organismo. Os diferentes valores que um gene pode assumir são chamados *alelos* (BECKET, 1986).

Nos algoritmos evolutivos, os cromossomos formam o que é chamado de *indivíduo*. Cada um deles representa uma solução candidata para o problema sendo tratado. No caso do presente trabalho, isso significa que cada indivíduo é uma solução candidata a ser o centro da via. O conjunto de todos os indivíduos forma a *população*. Assim como acontece com os organismos de uma mesma espécie, todas as soluções candidatas para um mesmo problema são codificadas no mesmo formato (GABRIEL; DELBEM, 2008).

As diferentes variáveis que compõem o indivíduo equivalem aos genes de um cromossomo, e os valores que estas variáveis podem assumir são os alelos. Isso está diretamente relacionado à forma de codificação escolhida.

2.4.1.2 Codificação dos indivíduos

A codificação é a forma como as características dos indivíduos (seus genes e alelos) é representada no algoritmo evolutivo. Na codificação binária cada indivíduo é composto por uma cadeia que aceita apenas os símbolos 0 e 1. Na codificação por números reais, apenas valores deste tipo podem ser usados. Os indivíduos também podem ser codificados em valores

inteiros, caracteres, objetos ou qualquer outro tipo de dado (EIBEN; SMITH, 2007).

A escolha da codificação depende das características do problema a ser analisado. Por exemplo, em casos que envolvem conjuntos de valores discretos, como o problema da soma de subconjuntos (dado um conjunto de números inteiros, achar o subconjunto cujo somatório seja igual a um valor definido (CORMEN et al., 2002)), a codificação binária pode ser mais interessante. Já para os casos que envolvem valores de intervalo contínuo, como distâncias ou velocidades, a codificação por números reais pode ser usada.

2.4.1.3 Reprodução

Outro ponto onde os algoritmos evolutivos se inspiram na biologia é com relação à reprodução. Dentre toda a população, os organismos mais bem adaptados ao ambiente possuem maior chance de sobreviver e se reproduzir, gerando descendentes. A reprodução pode ocorrer de três maneiras: por *recombinação* (também conhecida como *crossover*), *mutação* e/ou *inversão*.

No processo de recombinação as informações de indivíduos “pais” são combinadas para gerar uma nova solução candidata “filha”. A forma como isso é feito depende da codificação utilizada para representar os indivíduos (EIBEN; SMITH, 2007). Na codificação binária, a recombinação é feita particionando cada cromossomo pai em dois ou mais pedaços e mesclando-os para formar um novo indivíduo. Nos casos de codificação por valores reais, pode-se utilizar outra forma de recombinação chamada de *recombinação aritmética* ou *intermediária*, que atribui para cada gene do indivíduo filho um valor intermediário ao encontrado em cada pai (GABRIEL; DELBEM, 2008; MICHALEWICZ; JANIKOW, 1991).

O processo de mutação, por outro lado, escolhe aleatoriamente genes de um cromossomo e usa probabilidade para decidir se irá alterá-los ou não. Na codificação binária essa alteração é simples, visto que um gene aceita apenas dois alelos (zero ou um) (POLI; LANGDON; MCPHEE, 2008). Já na codificação por valores reais o gene pode ser incrementado por um valor aleatório, escolhido com base em probabilidade e de

modo a respeitar os valores limites válidos para o alelo (GABRIEL; DELBEM, 2008).

Finalmente, o processo de inversão é o mais simples dos três. Considerando que o cromossomo é uma cadeia de símbolos (uma lista de valores), este processo consiste em inverter a ordem dos símbolos em um dado trecho da cadeia. Isso faz sentido especialmente na codificação binária, e pode ser visto como uma forma de provocar mutação em vários genes ao mesmo tempo (MITCHELL, 1999).

2.4.1.4 Ciclo evolutivo

Dada uma população inicial, os algoritmos evolutivos executam de maneira cíclica. Primeiro, os indivíduos desta população são analisados. Depois, os indivíduos considerados mais bem adaptados se reproduzem, formando uma nova população.

A análise dos indivíduos é feita pela *função de fitness*. Essa função dá para cada indivíduo uma nota que representa sua proximidade da solução ideal (seu *fitness*). Quanto maior a nota, mais bem adaptado é o indivíduo. Isso irá depender diretamente do problema a ser resolvido, logo, a função deve ser definida manualmente, considerando as características e parâmetros do problema (POLI; LANGDON; MCPHEE, 2008).

A seguir, o mecanismo de seleção escolhe quais indivíduos irão se reproduzir. Isto pode ser feito usando diferentes métodos; alguns dos mais comuns são seleção por torneio, seleção por roleta, amostragem universal estocástica e seleção por *ranking*.

No método de seleção por torneio são escolhidos aleatoriamente N indivíduos. Compara-se o *fitness* de todos eles e o melhor é selecionado como descendente (BÄCK, 1996). Caso o processo de recombinação seja executado, criam-se dois torneios e a recombinação é aplicada sob os dois indivíduos selecionados. Este método permite que todos os indivíduos tenham a mesma probabilidade de participar de um torneio, independente do *fitness*. É possível também que um mesmo indivíduo seja selecionado mais de uma vez e participe de vários torneios. Apesar disso, o torneio sempre será vencido pelo indivíduo com melhor *fitness* dentre os selecionados (GABRIEL;

DELBEM, 2008). Mesmo assim, indivíduos com *fitness* inferior ainda podem se reproduzir (com exceção do pior indivíduo, pois perderá em qualquer torneio). Isso permite que os melhores indivíduos se reproduzam ao mesmo tempo em que se tenta evitar problemas com picos de máximo local.

O método de seleção por roleta atribui a cada indivíduo um intervalo dentro de uma roleta. O tamanho da roleta é a soma do *fitness* de todos os indivíduos, e o pedaço atribuído a cada indivíduo é exatamente seu valor de *fitness*. A seguir, gira-se a roleta e o indivíduo sorteado é selecionado como descendente (MITCHELL, 1999). Neste método a probabilidade de um indivíduo ser sorteado está diretamente relacionada ao seu *fitness*. Logo, indivíduos com *fitness* inferior têm menos chance de se reproduzir. Por outro lado, caso um indivíduo de *fitness* inferior seja sorteado ele não precisa disputar com ninguém – como ocorre na seleção por torneio – e é automaticamente selecionado.

O método de amostragem universal estocástica considera que, ao aplicar a seleção por roleta, é possível que todos os indivíduos selecionados sejam extremamente bons ou extremamente ruins. Para evitar isso o que este método faz é, ao invés de girar a roleta K vezes sorteando um indivíduo por vez, ele gira a roleta uma única vez e seleciona K indivíduos igualmente distantes uns dos outros (MICHALEWICZ, 1996). Isso torna a seleção de indivíduos mais equilibrada, aumentando a chance de um indivíduo de *fitness* inferior se reproduzir.

Por último, o método de seleção por *ranking* também é similar à seleção por roleta, mas ele abstrai o *fitness* de cada indivíduo e considera apenas seu *ranking* dentro de uma lista com os *fitness* ordenados (BÄCK, 1996). Uma forma de fazer isso é atribuindo ao pior indivíduo o *ranking* 1, ao segundo pior o *ranking* 2 e assim sucessivamente. O tamanho da roleta é a soma de todos os valores de *ranking*, e o pedaço da roleta atribuído a cada indivíduo é exatamente seu valor de *ranking*. Isso diminui a dominância exercida pelos indivíduos de melhor *fitness*, aumentando as chances de indivíduos piores também se reproduzirem.

Ao aplicar um destes métodos de seleção existe a chance de o melhor indivíduo de uma geração não ser selecionado para a próxima. Isso reduz a chance de o resultado

ficar preso em um pico de máximo local, porém dependendo do problema em questão pode não ser interessante. Nestes casos, alguns dos indivíduos podem ser selecionados usando como único critério seu *fitness*. Ou seja, parte da nova geração é formada pelos melhores indivíduos da geração anterior e apenas a outra parte é escolhida pelo método de seleção. Quando isto é feito, diz-se que a seleção é *elitista* (MITCHELL, 1999).

Uma última questão que irá variar de acordo com a abordagem utilizada é se a seleção seguirá um esquema ($\mu+\lambda$) ou (μ, λ). O elemento μ representa o número de pais e o elemento λ o número de descendentes, sendo que $\mu < \lambda$. O símbolo de soma e vírgula indicam se há ou não competição entre pais e filhos. Assim, no esquema ($\mu+\lambda$) μ pais geram λ descendentes; todos competem e apenas μ indivíduos são selecionados. Já no esquema (μ, λ) μ pais geram λ descendentes; os pais são descartados, os descendentes competem e μ indivíduos são selecionados (BITTENCOURT, 2006; EIBEN; SMITH, 2007).

Neste processo cíclico de criar a população, executar a função de *fitness* e aplicar o mecanismo de seleção, cada ciclo representa uma geração. O algoritmo evolutivo é interrompido quando a condição de parada especificada for atingida. Esta condição pode ser: foi criado o número desejado de gerações, a solução encontrada é suficientemente boa, ou qualquer outra condição adequada ao problema. Neste momento, o melhor indivíduo da última geração é apresentado como a solução encontrada pelo algoritmo evolutivo.

2.4.2 Abordagens

As três abordagens que servem de base para a área de Computação Evolutiva são os algoritmos genéticos, as estratégias evolutivas e a programação evolutiva. A partir delas novas pesquisas foram desenvolvidas para estudar aspectos diferentes de problemas de otimização. Isso levou à criação de abordagens novas, que podem ser vistas como extensões das abordagens iniciais. Uma destas abordagens são os algoritmos evolutivos para otimização multiobjetivo.

A seguir, as abordagens mencionadas são apresentadas.

2.4.2.1 Algoritmos Genéticos

Os algoritmos genéticos surgiram nos Estados Unidos, fruto do trabalho de Holland nas décadas de 1960 e 1970. Sua pesquisa não tinha como objetivo resolver um problema específico, mas sim estudar os fenômenos naturais e abstraí-los para serem utilizados em sistemas computacionais. Adicionalmente, ele buscou fundamentar teoricamente o funcionamento de seu algoritmo (MITCHELL, 1999).

Comparado aos outros trabalhos desenvolvidos na época, os grandes diferenciais dos algoritmos genéticos foram o uso de uma população de soluções candidatas e a aplicação da etapa de recombinação. Em especial, o uso de recombinação e mutação aumenta o aproveitamento das melhores soluções e, ao mesmo tempo, reduz os problemas com picos de máximo local durante a exploração do espaço de busca (GABRIEL; DELBEM, 2008). Nos anos seguintes estas ideias foram incorporadas em outros algoritmos a fim de torná-los mais eficientes (MITCHELL, 1999).

2.4.2.2 Estratégias Evolutivas

As estratégias evolutivas foram inicialmente desenvolvidas por Rechenberg na Alemanha na década de 1960 e Schwefel deu continuidade aos estudos na década seguinte. O uso principal deste tipo de algoritmo é para otimização de parâmetros (MITCHELL, 1999).

Nas primeiras versões deste algoritmo, cada geração contava apenas com dois indivíduos: um pai e seu filho. O único meio de reprodução era a mutação. Após a criação do filho, este era comparado com seu pai e o mais bem adaptado era mantido para gerar um novo descendente (esquema (1+1)) (BÄCK; HOFFMEISTER; SCHWEFEL, 1991). Após os algoritmos genéticos introduzirem os conceitos de população e recombinação, as estratégias evolutivas foram adaptadas. Isso as deixou semelhantes aos algoritmos genéticos. Contudo, existem diferenças, em especial na capacidade das estratégias evolutivas de realizarem um auto ajuste de parâmetros.

Introduzindo parâmetros de controle adicionais, as estratégias evolutivas podem controlar de forma independente a

mutação de cada variável (gene) para que ocorram num tipo de estratégia de escalada (*hill-climbing*). Estes parâmetros extras são configurados de maneira automática, num processo de aprendizado executado pelo algoritmo (HOFFMEISTER; BÄCK, 1991). Isso possibilita que as estratégias evolutivas se adaptem ao espaço de busca para tentar encontrar melhores soluções. Por outro lado, adiciona complexidade ao desenvolvimento desta abordagem (BÄCK; HOFFMEISTER; SCHWEFEL, 1991).

2.4.2.3 Programação Evolutiva

A programação evolutiva foi inventada por Fogel em 1962, também nos Estados Unidos. Esta abordagem se diferencia das demais no sentido de que cada indivíduo é representado como uma máquina de estados finitos. A solução candidata evolui apenas através da mutação dos estados de transição, numa espécie de “reprodução assexuada”. Para compensar a falta da recombinação, todos os indivíduos geram descendentes (GABRIEL; DELBEM, 2008; MITCHELL, 1999). A função de *fitness* verifica a qualidade de cada indivíduo através da execução de cada máquina de estado finito para verificar se ela alcança a solução desejada. Por último, o mecanismo de seleção analisa todos os indivíduos e seleciona os melhores, independente de ser um indivíduo pai ou descendente (seleção elitista total, esquema $(\mu+\lambda)$) (GABRIEL; DELBEM, 2008).

2.4.2.4 Algoritmos Evolutivos para Otimização Multiobjetivo

Abordagens multiobjetivo tentam encontrar soluções que sejam adequadas para atender diferentes requisitos do problema ao mesmo tempo (EIBEN; SMITH, 2007). Uma maneira de resolver problemas deste tipo é usando um esquema de ponderação. Cada objetivo tem a sua própria função de *fitness*, porém os resultados são multiplicados por valores (específicos para cada objetivo) que ponderam a influência de cada um deles (ou seja, determinam o peso). A seguir, a soma dos resultados gera a nota final que é atribuída à solução candidata. Desta forma, os diferentes objetivos são abstraídos e o problema multiobjetivo é, efetivamente, transformado em um problema com um único objetivo (MICHALEWICZ, 1996). Contudo, para que o

resultado deste método seja o melhor possível, é necessário conhecer previamente quais valores de ponderação são ideais para o problema em questão (GABRIEL; DELBEM, 2008), e estes devem se manter estáticos durante todas as execuções do algoritmo (EIBEN; SMITH, 2007). Essa restrição levou ao desenvolvimento de outras abordagens que não usam o esquema de ponderação. Elas são chamadas de Algoritmos Evolutivos para Otimização Multibojetivo (AEOM), e se utilizam do conceito de dominância.

Este conceito diz que se uma solução *A* é melhor que uma solução *B*, *A* domina *B*. Isto se aplica tanto sob o ponto de vista de um único objetivo como também no conjunto de todos os objetivos sendo avaliados. A função de *fitness* destas abordagens não gera uma única nota, mas sim uma específica para cada objetivo. Por este motivo, estas abordagens não encontram um único resultado, mas um conjunto de soluções. Este conjunto recebe o nome de *Pareto-ótimo* e é composto pelas soluções que não são dominadas por nada, ou seja, que são as melhores no conjunto de todos os objetivos (MICHALEWICZ, 1996).

De modo geral, abordagens deste tipo são modificações das três abordagens primárias para lidar com multiobjetivo sem usar o esquema de pesos. Alguns exemplos de abordagens para otimização multiobjetivo são Vector Evaluated Genetic Algorithm (VEGA), Multiple Objective Genetic Algorithm (MOGA), Predator-Prey Evolution Strategy (PPES) ou Pareto Envelope-Base Selection Algorithm (PESA).

2.4.3 Solução escolhida

Acerca das abordagens apresentadas pela literatura, vale destacar os seguintes aspectos de cada uma delas. Os algoritmos genéticos possibilitam gerar e analisar um conjunto de soluções candidatas (população da geração) e escolher a que melhor se adeque ao problema em questão. Além disso, por utilizar recombinação e mutação, este tipo de algoritmo evolutivo tende a preservar as melhores soluções ao mesmo tempo em que reduz problemas com picos de máximo local (GABRIEL; DELBEM, 2008).

As estratégias evolutivas têm certa semelhança com algoritmo genético. Porém, como Mitchell (1999) comenta, o uso principal deste tipo de algoritmo é para otimização de parâmetros. Logo, não é recomendado para o presente trabalho.

A programação evolutiva representa soluções candidatas em uma máquina de estados finitos. Considerando os dados que compõem as soluções candidatas do problema em questão (principalmente latitude e longitude, dois valores reais) esta abordagem não é adequada para este trabalho.

Por último, o principal diferencial dos AEOM em relação às demais abordagens é que eles não usam pesos para escolher a melhor solução candidata. Ao invés disso, usam o conceito de dominância para achar um conjunto de soluções ótimas (chamado conjunto *Pareto-ótimo*). Porém, o método proposto no presente trabalho deve fornecer como resposta do algoritmo evolutivo um único resultado, que será a solução candidata selecionada para ser o centro da via. Sendo assim, para usar a abordagem AEOM seria necessário uma heurística adicional para escolher apenas uma das soluções do conjunto *Pareto-ótimo*. Por isso, esta não seria a abordagem mais adequada para este trabalho.

Com base nestas observações, conclui-se que o algoritmo genético é uma abordagem adequada ao problema em questão.

Além da abordagem utilizada, também é necessário escolher o mecanismo de seleção. Em relação às opções apresentadas, Mitchell (1999) comenta que os mecanismos de seleção por roleta e por amostragem universal estocástica dão importância excessiva para o valor de *fitness*. A seleção por *ranking* é mais adequada, porém tem um custo computacional mais alto. Finalmente, ele diz que a seleção por torneio tem pressão seletiva similar a da seleção por ranking, mas é computacionalmente mais eficiente. Sendo assim, acredita-se que o mecanismo de seleção por torneio é o mais indicado para o presente trabalho.

Adicionalmente, a seleção dos indivíduos é elitista. Desta forma, alguns dos melhores indivíduos da geração atual são mantidos para a geração seguinte. Isso faz com que a melhor solução candidata encontrada até o momento nunca seja descartada; se uma solução melhor é encontrada, ela é que

passa a ser a solução preservada para a próxima rodada. Ao mesmo tempo, a recombinação e mutação mantêm a variabilidade genética necessária para tentar encontrar uma solução candidata melhor na próxima geração.

2.5 Outras propostas para geração de mapas rodoviários

A literatura apresenta alguns trabalhos que manipulam dados geográficos no intuito de gerar mapas rodoviários. Dentre estes trabalhos cinco merecem destaque, são eles: Brüntrup et al. (2005), Cao e Krumm (2009), Jang, Kim e Lee (2010), Zhang, Thiemann e Sester (2010) e Niu, Li e Pousaeid (2011). A seguir estes trabalhos são revisados. Na sequência, são apresentadas as considerações a respeito das técnicas apresentadas por eles em comparação com a adotada no presente trabalho.

2.5.1 Brüntrup et al. (2005)

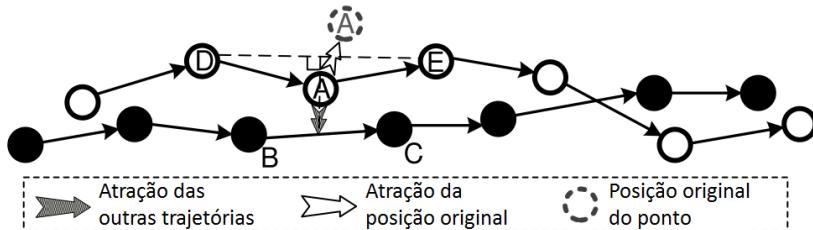
Brüntrup et al. (2005) apresentam um método para criação de mapas rodoviários que usa dados de qualquer receptor GPS, desde que eles sejam coletados ao trafegar usando carros. Após a coleta das trajetórias, elas são filtradas com base em limites de velocidade, aceleração, e relação entre distância e intervalo de tempo entre dois pontos em sequência. A seguir, cada trajetória é dividida em segmentos e, para cada um deles, é usada uma técnica de clusterização baseada em Inteligência Artificial para identificar quais pontos devem ser inseridos no mapa. Este método não depende de um mapa inicial e, por isso, permite mapear regiões ainda sem mapas. Devido à forma como as trajetórias são manipuladas, os autores comentam que a atualização do mapa pode ser feita com o mesmo método usado na criação do mapa inicial.

2.5.2 Cao e Krumm (2009)

Cao e Krumm (2009) apresentam um método que usa dados obtidos de navegadores GPS automotivos. Para reduzir o ruído, os dados são filtrados com base em limites de distância, intervalo de tempo e ângulo de movimento. A seguir, eles encontram o centro das vias, que são diferenciadas de acordo com o ângulo de movimento. Para isso, desenvolveram uma

técnica baseada em princípios de atração e repulsão física, que é aplicada a cada um dos pontos relacionando-os com todos os pontos próximos. A Figura 23 demonstra esta etapa do método. Nela, o ponto A é movimentado em direção à B e C devido à força exercida pelos pontos.

Figura 23 – Método de Cao e Krumm (2009)



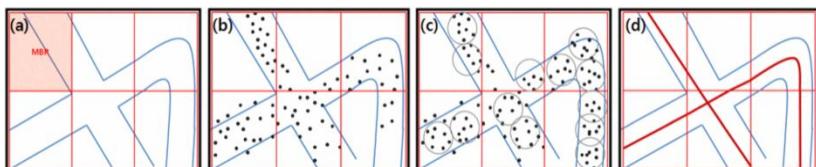
Fonte: (CAO; KRUMM, 2009), tradução nossa

Assim como o trabalho anterior, este método também não depende de um mapa inicial. Os autores não discutem sobre a atualização do mapa.

2.5.3 Jang, Kim e Lee (2010)

Outro trabalho relacionado é o de Jang, Kim e Lee (2010), que utiliza dispositivos móveis equipados com GPS para obter os dados. Contudo, assim como no trabalho de Brüntrup et al. (2005), dependem que a coleta seja feita ao trafegar usando carros. Eles não descrevem nenhuma etapa de filtragem dos dados. Para gerar o mapa, dividem a área onde os dados foram coletados em quadrados de um metro. A seguir, clusterizam os pontos em cada quadrado com base na distância para os quadrados vizinhos. Os clusters são então conectados para formar a malha viária, e a análise da forma e ângulo das vias geradas serve para corrigir eventuais problemas. Essa etapa não é detalhada pelos autores. A figura a seguir demonstra o funcionamento do método. Na Figura 24.A se vê a divisão da área a ser mapeada, na Figura 24.B se vê os pontos em cada quadrado, na Figura 24.C se vê a clusterização e na Figura 24.D se vê o resultado final.

Figura 24 – Método de Jang, Kim e Lee (2010)



Fonte: Adaptado de (JANG; KIM; LEE, 2010)

Assim como os trabalhos anteriores, este também não faz uso de um mapa inicial. Finalmente, os autores comentam que o método de atualização do mapa não foi automatizado e que, portanto, ele seria ineficiente.

2.5.4 Zhang, Thiemann e Sester (2010)

O objetivo deste trabalho é atualizar um mapa existente, ao invés de criar um a partir do zero. Para isso, Zhang, Thiemann e Sester realizaram seus testes usando trajetórias enviadas ao OpenStreetMap (OSM). O OSM é um serviço online que utiliza trajetórias enviadas pelos usuários para gerar mapas gratuitos (OSM, 2013a). As trajetórias enviadas para o serviço ficam disponíveis publicamente (OSM, 2013b). Como elas podem ter sido coletadas com qualquer receptor GPS, pode-se dizer que a solução de Zhang, Thiemann e Sester (2010) aceita dados de qualquer tipo de dispositivo.

Inicialmente, a solução proposta divide as trajetórias em segmentos de acordo com a velocidade e distância entre os pontos. A seguir, a identificação dos centros das vias é feito com base em um mapa inicial, e o mapa escolhido é o do OSM. Para cada via do mapa são criadas retas perpendiculares a intervalos de poucos metros e qualquer trajetória que as intersecte será selecionada. Estas trajetórias são filtradas pela distância à via e ângulo de movimento. Caso duas vias do mapa estejam muito próximas, usa-se um algoritmo *fuzzy c-means* para diferenciar as trajetórias de cada uma. Finalmente, o conjunto dos pontos onde cada trajetória intersectou a reta perpendicular serve de entrada para um método de estimação robusta (não detalhado pelos autores), e os novos centros da via são os pontos no intervalo de confiança de 95%.

2.5.5 Niu, Li e Pousaeid (2011)

O trabalho de Niu, Li e Pousaeid (2011) usa *smartphones* (Blackberry e iPhone) como dispositivos para a coleta de dados, e supõe que todos os dados foram coletados enquanto os usuários usavam como meio de transporte um carro. A filtragem de cada ponto é feita com base em limites definidos para sua acurácia e ângulo de movimento. Um mapa desatualizado da área de testes é usado como referência para otimizar essa etapa de filtragem. Pontos com velocidade igual a zero ou pontos repetidos na mesma coordenada também foram descartados. A seguir, aplicam uma técnica de análise de regressão local combinado com clusterização subtrativa para encontrar os pontos que representam os centros das vias. Os autores comentam que este método serviria para atualizar mapas antigos ao invés de criar um do zero, da mesma forma que o trabalho de Zhang, Thiemann e Sester (2010).

2.5.6 Considerações

Apesar de cada trabalho propor um método diferente para encontrar o centro das vias e gerar os mapas, existem algumas similaridades entre as soluções. Algumas dessas similaridades são interessantes e o presente trabalho segue as mesmas ideias. Um exemplo é a independência de mapas iniciais. Apenas Niu, Li e Pousaeid (2011) e Zhang, Thiemann e Sester (2010) usam um mapa como referência, pois o objetivo deles é melhorar mapas já existentes.

Outra similaridade interessante é em relação à filtragem das trajetórias para remover ruído. Com exceção de Jang, Kim e Lee (2010), todos os outros trabalhos definem heurísticas baseadas nos atributos fornecidos pelo GPS, como velocidade, aceleração, distância, entre outros, para eliminar dados de baixa qualidade. De todos os trabalhos revisados o de Niu, Li e Pousaeid (2011) é o único que diz usar trajetórias coletadas com *smartphones*, logo, o conjunto de heurísticas de filtragem mais relevantes é aquele que eles definiram – filtragem por acurácia, ângulo de movimento, velocidade e proximidade dos pontos – pois o presente trabalho também usa o mesmo tipo de aparelho.

Além disso, é interessante destacar que os outros trabalhos também usam uma ou mais destas heurísticas.

Por outro lado, o presente trabalho trata algumas etapas da solução de maneira diferente. Por exemplo, todas as propostas analisadas assumem que os arquivos coletados contêm apenas trajetórias de veículos motorizados. Zhang, Thiemann e Sester (2010) comentam que os arquivos que eles utilizam podem ser de meios de transporte diferentes, mas não deixa claro se é feito algum tipo de tratamento por conta disso. O presente trabalho considera que as trajetórias coletadas com *smartphones* podem ter sido geradas enquanto o usuário utilizava diferentes meios de transporte, então é necessário identificar os meios de transporte usados ao longo de cada trajetória usando uma das técnicas revisadas na seção 2.1.

Outra preocupação é que os trabalhos revisados pouco discutem sobre a atualização do mapa. Esta é uma questão importante porque novas vias podem ser criadas e vias existentes podem ser alteradas de alguma forma, mesmo que temporariamente. Pode-se pensar que gerando um novo mapa conforme mais trajetórias são coletadas seria suficiente, entretanto, isto não é verdade. Quando estes trabalhos relacionados estão identificando o centro das vias eles não levam em consideração a data em que as trajetórias foram coletadas. Por causa disso, o resultado do novo mapa seria uma mistura de trajetórias antigas e novas, a menos que as trajetórias antigas fossem apagadas da base de dados coletados. O fato de não analisarem a data de coleta também afeta o que é feito quando nenhuma nova trajetória é coletada em uma via, por exemplo, se a via é mantida com base nos dados do mapa antigo, se ela é removida, ou qual outro processamento é feito. Portanto, o presente trabalho analisa as trajetórias coletadas ao longo do tempo para identificar mudanças e refleti-las no mapa corretamente, buscando equilibrar a identificação de novas vias, as alterações ocorridas nas vias já mapeadas e a quantidade de vias perdidas devido à idade dos dados coletados.

3 Solução proposta

Levando em consideração a revisão bibliográfica realizada e os pressupostos assumidos por este trabalho, este capítulo apresenta a solução proposta para geração de mapas rodoviários a partir de dados coletados por *smartphones*. O capítulo está organizado nas seguintes seções. A seção 3.1 detalha o método de geração de mapas, destacando as etapas de pré-processamento das trajetórias, identificação do centro das vias e geração do grafo direcionado. A seguir, a seção 3.2 comenta sobre as principais questões relacionadas à implementação.

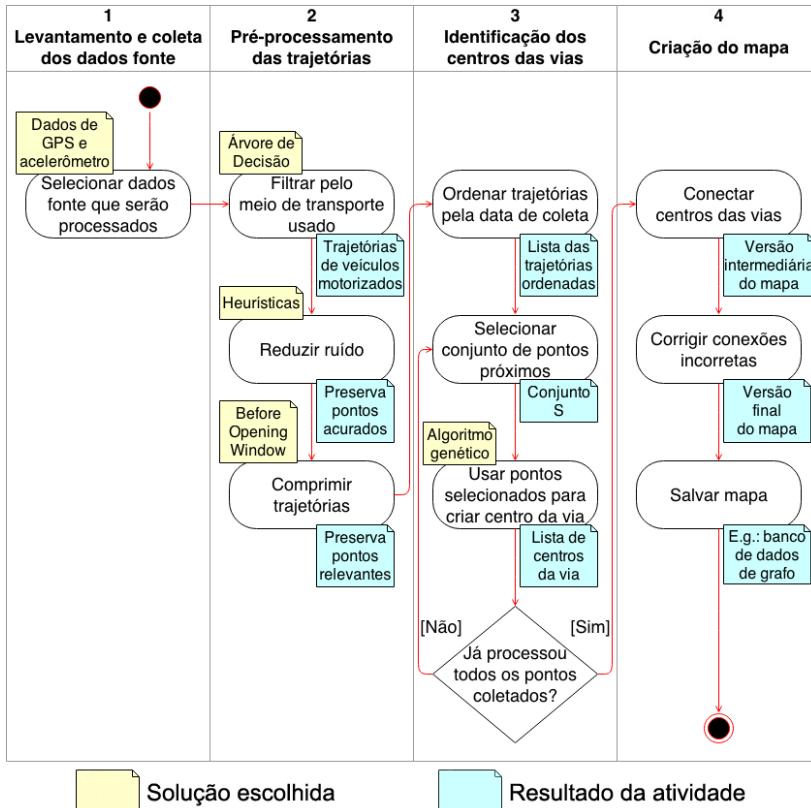
3.1 Método Proposto

O método proposto pode ser dividido em quatro etapas: (i) Levantamento dos dados necessários para a geração dos mapas rodoviários, considerando as características da forma de solução proposta (coleta com *smartphones*; uso de algoritmo genético); (ii) Pré-processamento das trajetórias coletadas, com o intuito de melhorar a qualidade dos dados fonte; (iii) Identificação do centro das vias; (iv) Por último, criação de um grafo direcionado a partir dos centros das vias identificados.

A Figura 25 apresenta o funcionamento do método por meio de um diagrama de atividades. Para facilitar o entendimento, os passos de cada etapa foram simplificados de modo a representar apenas os elementos mais importantes em relação ao processamento das trajetórias. Adicionalmente, a Figura 25 também apresenta a solução escolhida para cada um dos desafios identificados e revisados no capítulo 2 e a saída de cada uma das atividades.

Vale destacar que este método é uma evolução daquele que foi apresentado em Costa e Baldo (2013), cujos autores são os mesmos do presente trabalho.

Figura 25 – Diagrama do funcionamento do método



Fonte: produção do próprio autor

A seguir, cada subseção apresenta em detalhes as etapas do método proposto.

3.1.1 Levantamento e coleta dos dados fonte necessários

Antes de iniciar a geração de mapas rodoviários é preciso detalhar o conjunto de dados necessários para a realização desta tarefa. Portanto, esta seção se destina a descrever estes dados, evidenciando sua relevância dentro do processo.

Sabendo que o método se propõe a gerar mapas rodoviários georreferenciados, um dado que é requisito

fundamental para a execução do método é a latitude e longitude de cada ponto que compõe cada trajetória. Estes dados são obtidos do receptor de GPS embutido no *smartphone*. Entretanto, visto o grande volume de pontos contidos nas trajetórias, é necessário realizar a compressão delas (quando se utiliza o algoritmo Before Opening Window, vide seção 2.3). As coordenadas também são utilizadas para calcular o ângulo de movimento (vide Figura 04, página 44), que representa em qual direção o objeto móvel está se locomovendo.

Para o desafio de reduzir o ruído, é necessário que o GPS armazene a acurácia de cada ponto. Em virtude da acurácia variável do GPS, os pontos ficam espalhados por toda a via de onde foram coletados, em uma espécie de nuvem. Assim, a análise da acurácia possibilita identificar os pontos com georreferenciamento mais próximo da realidade.

Também é importante armazenar a data de coleta de cada ponto, pois pontos recentes têm maior relevância na geração de mapas do que pontos mais antigos. Logo, analisar a data de coleta ajuda a gerar mapas atualizados.

Outro desafio na geração de mapas rodoviários é identificar os meios de transportes utilizados ao longo de cada trajetória. Para tanto, conforme revisão feita na seção 2.1, dois dados devem ser utilizados: a velocidade do objeto móvel e a aceleração do *smartphone* em relação aos eixos X, Y e Z. A velocidade é obtida do receptor de GPS em conjunto com a acurácia e data de coleta do ponto. Já os dados de aceleração são coletados pelo sensor de acelerômetro.

Com base em todos os requisitos apresentados, o conjunto completo de dados coletados para cada ponto inclui: dados de aceleração nos três eixos; latitude e longitude; velocidade; acurácia; e data de coleta. Cada dado do sensor de acelerômetro é composto por três valores reais, que representam a aceleração do *smartphone* em relação aos eixos X, Y e Z. Latitude, longitude, velocidade e acurácia também são valores reais. A latitude e longitude são representadas em graus decimais. A velocidade é coletada em metros por segundo, e a acurácia em metros. A data em que o ponto foi coletado contém ano, mês, dia, hora, minuto, segundo e milissegundo, todos valores inteiros.

Para fazer a coleta destes dados foi desenvolvido um aplicativo para *smartphones* com sistema operacional Android. Os dados de GPS são coletados uma vez por segundo, e os de acelerômetro cerca de vinte vezes por segundo. Ou seja, cada ponto da trajetória é composto por um conjunto de dados de GPS e aproximadamente 20 de acelerômetro. Mais detalhes sobre o aplicativo desenvolvido para coleta dos dados podem ser vistos no apêndice A.

Usando o aplicativo desenvolvido foram coletadas trajetórias de diversos tamanhos: desde alguns poucos pontos até mais de 4000. Cada trajetória é armazenada em um arquivo de extensão CSV (do inglês *Comma-Separated Values*). A identificação de cada trajetória é feita pelo nome do arquivo, que é composto pela data de início da coleta da trajetória. Os arquivos são enviados para um servidor. Lá, as trajetórias permanecem armazenadas até que a etapa de pré-processamento seja executada. Mais detalhes sobre as trajetórias coletadas são apresentadas na seção 4.1.

3.1.2 Pré-processamento das trajetórias

O pré-processamento das trajetórias é composto por três partes, cada uma tratando de um dos desafios apresentados no capítulo 2, relacionados à qualidade dos dados. Ao final desta etapa, as trajetórias coletadas serão consideradas “limpas”, pois todos os pontos identificados como de baixa qualidade ou com informações incorretas foram removidos.

A primeira parte do pré-processamento é a identificação do meio de transporte, pois apenas trajetórias coletadas com veículos motorizados devem ser usadas para gerar o mapa. A trajetória é dividida em trechos de $tam_{amostra} = 2$ segundos (vide seção 2.1.1). Trechos sequenciais de veículo motorizado são considerados como uma mesma via. Ao encontrar um trecho de outro meio de transporte, a via é finalizada e o trecho é descartado. O próximo trecho de veículo motorizado irá iniciar uma nova via. Isso significa que, devido a falsos negativos, uma trajetória coletada apenas com veículo motorizado pode ser dividida em vias menores.

A segunda parte é a redução de ruído, através da remoção de pontos com baixa qualidade (por exemplo, acurácia

ruim) e confiabilidade (informações errôneas). Isso é feito através de um conjunto de heurísticas (vide seção 2.2.1). Terminada esta parte, todos os pontos que poderiam prejudicar a qualidade do mapa gerado já foram eliminados.

Por último, a compressão das trajetórias reduz a quantidade de pontos pouco significativos a serem processados no momento da identificação do centro da via. Desta forma, reduz-se também o tempo de processamento necessário até a geração do mapa. Essa parte é executada por meio do algoritmo Before Opening Window (vide seção 2.3.3).

Maiores detalhes sobre cada desafio se encontram nas seções 2.1, 2.2 e 2.3, respectivamente.

3.1.3 Identificação dos centros de cada via

Considera-se que encontrar o centro das vias é a parte mais importante do processo de criação dos mapas rodoviários. Devido à acurácia variável dos pontos coletados (vide capítulo 2), assume-se que nenhum deles está posicionado corretamente e, portanto, é necessário encontrar a posição que represente mais fielmente cada centro da via. Fazer isso de maneira determinística não é viável computacionalmente. Por isso, neste trabalho será utilizada aproximação usando algoritmo genético.

Partindo destas considerações, o Quadro 02 apresenta o pseudocódigo do método para encontrar os centros das vias. Ele começa processando a trajetória mais recente e acurada até chegar a mais antiga e com pior acurácia (linha 02 do Quadro 02). Esta estratégia possibilita a atualização dos mapas com base nos dados que foram coletados mais recentemente.

Com base nesta ordem de processamento, o método percorre as trajetórias analisando conjuntos de pontos próximos (linhas 03, 04, 06 e 07 do Quadro 02). Isto é feito porque, como o método não usa nenhum dado auxiliar além do que foi coletado com *smartphones*, é necessário analisar muitos pontos de boa qualidade ao mesmo tempo para definir melhor o centro da via. Isso significa que o método desenvolvido encontrará um centro da via na área próxima a cada conjunto de pontos analisado.

Como o processo é iterativo, o modo como os conjuntos são definidos se assemelha a um processo de demarcação de áreas onde os centros das vias serão identificados. Desta forma,

cada ponto deve participar do processo uma única vez – ou seja, deve influenciar a localização de apenas um centro da via. Por isso, pontos já utilizados são marcados para não serem usados novamente (linhas 05, 17 e 18 do Quadro 02).

Finalmente, conforme revisado na seção 2.4, algoritmos genéticos se destacam como opção válida e viável para o presente contexto. Por este motivo, esse algoritmo é empregado para achar a localização aproximada dos centros das vias (linhas 08 a 16 do Quadro 02).

Quadro 02 – Pseudocódigo do método de identificação dos centros das vias

Entrada:

lista_trajetórias: lista das trajetórias ordenadas por data e acurácia média;

```

01 Inicializa centros_das_vias como uma lista vazia;
02 PARA CADA trajetória em lista_trajetórias FAÇA
03   PARA CADA ponto P da trajetória FAÇA
04     SE P ainda não foi marcado como usado ENTÃO
05       Cria conjunto S por meio de consulta ao banco de
          dados para identificar todos os pontos próximos a P
          com ângulo de movimento similar;
06       Adiciona P no conjunto S;
07       Define domínio com base no conjunto S;
08       Cria primeira geração com base no conjunto S;
09     REPITA
10       PARA CADA candidato na geração atual FAÇA
11         Calcula o fitness do candidato;
12         Ordena os candidatos pelo valor de fitness;
13         Cria nova geração com base nos melhores
             candidatos e no domínio;
14     ATÉ QUE g gerações tenham sido criadas;
15     Adiciona melhor candidato em centros_das_vias;
16     PARA CADA ponto no conjunto S FAÇA
17       Marca ponto como usado;
```

Saída: *centros_das_vias*.

Fonte: produção do próprio autor

Nas próximas subseções cada passo do método proposto é explicado em detalhes.

Vale destacar que todos os parâmetros do método foram escolhidos empiricamente com base em testes com um conjunto de aproximadamente 200 mil pontos. Estes parâmetros foram refinados ao longo dos ciclos de iteração do processo baseado na metodologia de Pesquisa-Ação. Detalhes adicionais sobre cada parâmetro podem ser encontrados no apêndice B.

3.1.3.1 Definição da ordem de análise dos pontos

O primeiro passo da identificação dos centros de cada via é consultar o banco de dados para obter a lista das trajetórias que foram filtradas. As trajetórias são ordenadas primeiramente pela data de coleta (da mais recente a mais antiga, usando como granularidade o dia) e, como critério secundário, pela acurácia média das trajetórias (daquela com melhor acurácia a pior). A Figura 26 demonstra o resultado desta consulta.

Figura 26 – Resultado da consulta ao banco de dados

	cod character varying	ano integer	mes integer	dia integer	acuracia_media double precision
1	20130615_135946856	2013	6	15	4.29411764706
2	20130615_140934836	2013	6	15	4.65454545455
3	20130615_140125860	2013	6	15	4.75735294118
4	20130615_135204560	2013	6	15	4.85130718954
5	20130615_132759643	2013	6	15	5.33986928105
6	20130615_133308506	2013	6	15	5.68
7	20130615_131418788	2013	6	15	5.70114942529
8	20130615_135306624	2013	6	15	5.88302425107
9	20130615_133036520	2013	6	15	6.11764705882
10	20130615_133218556	2013	6	15	6.125
11	20130615_135355857	2013	6	15	6.31057662777

Fonte: produção do próprio autor

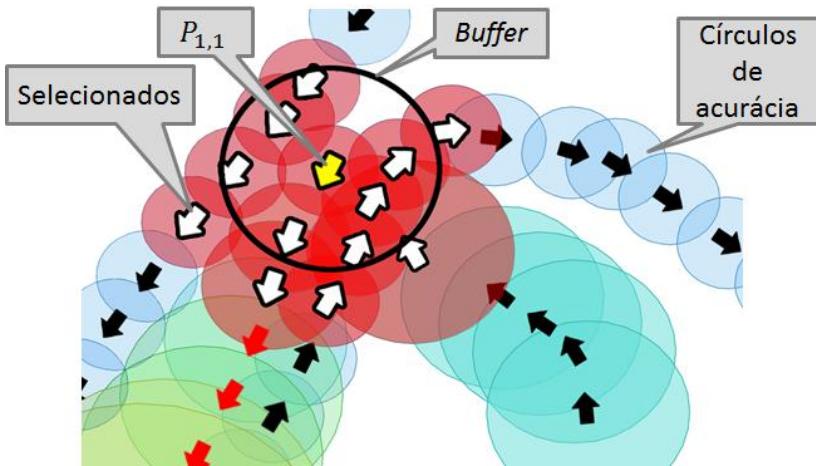
A seguir, cada ponto de cada trajetória é processado. Ao todo existem q trajetórias, e cada uma tem uma quantidade variável n de pontos. A relação entre as q trajetórias e os n pontos pode ser representada na forma $P_{j,k}$ ($j = 1, 2, \dots, q; k = 1, 2, \dots, n$).

A ordem de processamento é aquela do resultado da consulta ao banco de dados (Figura 26). Ou seja, o processamento se inicia pelo primeiro ponto da trajetória mais recente e acurada – o ponto $P_{1,1}$ – e finaliza no último ponto da última trajetória do conjunto – o ponto $P_{q,n}$.

3.1.3.2 Seleção de pontos para o conjunto S

A primeira ação realizada para encontrar o centro da via na área próxima a $P_{1,1}$ é encontrar os outros pontos localizados na mesma área, independente da trajetória a qual eles pertençam. Para isso, determina-se uma área – um *buffer* – ao redor de $P_{1,1}$ (o raio do *buffer* é o valor de acurácia de $P_{1,1}$) e se analisa o círculo de acurácia (círculo centrado nas coordenadas do ponto e com raio igual ao seu valor de acurácia) de todos os pontos de todas as trajetórias. Se algum ponto intersectar o *buffer* com seu círculo de acurácia ele é selecionado, pois se considera que está próximo a $P_{1,1}$. A Figura 27 exemplifica este processo.

Figura 27 – Definição do conjunto de pontos selecionados



Fonte: produção do próprio autor

A próxima parte do processo analisa o ângulo de movimento dos n' pontos selecionados. Isto é necessário para

identificar pontos que possam pertencer a vias diferentes de $P_{1,1}$ ou à mão de sentido contrário.

Ao trafegar por uma via de mão dupla, o ângulo em cada mão será praticamente oposto, ou seja, com cerca de 180 graus de diferença. Em uma esquina, a diferença na angulação será de aproximadamente 90 graus. Em outros casos, a diferença poderá ser menor, mas ainda considerável. Com exceção de algumas vias paralelas, como marginais de rodovias, qualquer via tem uma diferença de angulação em relação às outras ao redor. A Figura 28 ilustra a diferença de angulação.

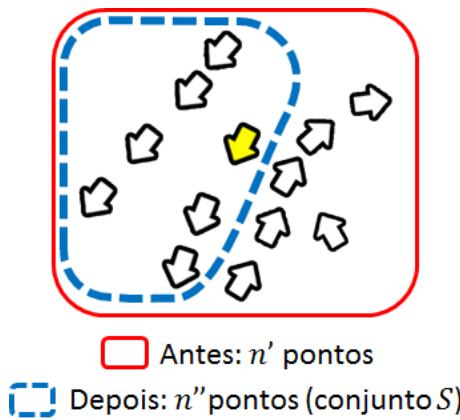
Figura 28 – Diferença na angulação de duas vias



Fonte: produção do próprio autor

Como o centro da via a ser definido representa uma única mão de uma única via, pontos de mãos e/ou vias diferentes devem ser desconsiderados. Assim, a identificação do ângulo de movimento permite manter apenas os n'' pontos relevantes para encontrar o centro da via na área próxima à $P_{1,1}$. Ou seja, apenas os pontos cujo ângulo de movimento não varie mais que $\theta_{lim} = 25$ metros em relação à $P_{1,1}$ (valor escolhido empiricamente). Estes n'' pontos formam o conjunto S , o conjunto dos pontos selecionados. A Figura 29 demonstra o resultado deste processo, continuando o exemplo iniciado na Figura 27.

Figura 29 – Descarte de pontos com ângulo de movimento diferente



Fonte: produção do próprio autor

O último passo antes de prosseguir para a identificação do centro da via é verificar a data dos pontos do conjunto S . Conforme comentado anteriormente (e explicado em detalhes na próxima seção), o presente trabalho analisa as trajetórias ao longo do tempo. Isso significa que os pontos das trajetórias coletadas recentemente são considerados mais importantes que os pontos das trajetórias coletadas em um momento anterior. Com o passar do tempo, essas trajetórias coletadas anteriormente vão se tornando cada vez mais antigas, ou seja, cada vez mais desatualizadas. Para que os pontos antigos não atrapalhem a geração do mapa, compara-se a data atual à data de cada ponto do conjunto S . Se a diferença entre as datas superar o valor limite de $t_{max} = 180$ dias (valor escolhido empiricamente), o ponto é removido do conjunto S . Essa estratégia também é necessária para tratar de vias que deixam de existir ou que tem sua direção de movimento alterada. Assim, a verificação da data permite eliminar os pontos antigos que, possivelmente, não representam mais a realidade.

Depois desta filtragem, verifica-se o tamanho do conjunto S . Se ele contiver menos de seis pontos ($n'' < n_{min}$, onde $n_{min} = 6$) o conjunto é descartado e o método reinicia, tentando encontrar o centro da via partindo de outro ponto (vide explicação

na seção 3.1.3.5). Isso é feito para evitar que se tente encontrar centros da via em locais com poucos pontos. A justificativa é que, geralmente, isso acontece com conjuntos formados a partir de pontos de baixa confiabilidade – ou seja, pontos com acurácia aceitável, mas coordenadas erradas. Cao e Krumm (2009) também usam um parâmetro com função similar. No presente trabalho, o valor $n_{min} = 6$ foi escolhido empiricamente.

Se o conjunto S contiver pontos suficientes (ou seja, $n'' \geq n_{min}$), o método continua sua execução normal, seguindo o processo descrito na seção a seguir.

3.1.3.3 Definição da função de *fitness*

Tendo criado o conjunto S , o próximo passo é utilizá-lo para encontrar o centro da via. Conforme discutido na seção 2.4.3, a forma escolhida para tratar deste problema é usando um algoritmo genético. Esse algoritmo executa por um número determinado de gerações e, em cada uma delas, usa sua função de *fitness* para encontrar as melhores soluções candidatas. O conjunto das soluções candidatas de uma determinada geração é chamado de conjunto C . Cada elemento deste conjunto pode ser referenciado como C_x ($x = 1, 2, \dots, z$), onde z é o número máximo de soluções candidatas por geração.

Dada essa necessidade de se utilizar um método de aproximação, assume-se que o centro da via é uma combinação ponderada entre (i) data de coleta dos pontos do conjunto S , (ii) acurácia dos pontos do conjunto S , e (iii) distância entre as soluções candidatas no conjunto C e os pontos do conjunto S . Combinando essas três características, a solução candidata C_x selecionada como centro da via para o conjunto S será aquela mais próxima à maior concentração de pontos recentes e acurados. Partindo dessa lógica, criou-se a função de *fitness* apresentada na Equação (1). A sua saída é um valor no intervalo [0..1] que indica o *fitness* da solução candidata. Quanto maior for o *fitness*, melhor é a solução candidata. O resultado das equações *IT* (Equação (2)), *IA* (Equação (3)) e *ID* (Equação (4)) também fica no intervalo [0..1]. Mais detalhes sobre os elementos da função de *fitness* são apresentados a seguir.

$$FITNESS(C_x, S) = \frac{\sum_{i=1}^{n''} IT(S_i) \cdot MT + IA(S_i) \cdot MA + ID(C_x, S_i) \cdot MD}{n''} \quad (1)$$

C_x = Elemento do conjunto C .

S = Conjunto S .

n'' = Tamanho do conjunto S (vide seção 3.1.3.2).

IT = Equação de Influência do Tempo – Equação (2).

IA = Equação de Influência da Acurácia – Equação (3).

ID = Equação de Influência da Distância – Equação (4).

MT , MA e MD = Peso dos critérios de influência, sendo que

$$MT + MA + MD = 1.$$

Como explicado, a Equação (2) define a influência do tempo, ou seja, da data de coleta da trajetória. Ela se comporta de tal maneira que quanto mais recente é o ponto maior é o seu *fitness*. O parâmetro $t_{max} = 180$ dias é o mesmo citado na seção 3.1.3.2. Com este valor, acredita-se que a equação ofereça um equilíbrio entre refletir modificações nas vias e desconsiderar pontos ultrapassados.

$$IT(S_i) = \frac{t_{max} - |T(S_i, S_r)|}{t_{max}} \quad (2)$$

S_i = Elemento do conjunto S .

S_r = Elemento mais recente dentro do conjunto S .

$T(S_i, S_r)$ = Calcula o intervalo de tempo entre a coleta de S_i e S_r .

t_{max} = Maior intervalo de tempo decorrido permitido. Nos testes se utilizou $t_{max} = 180$ dias.

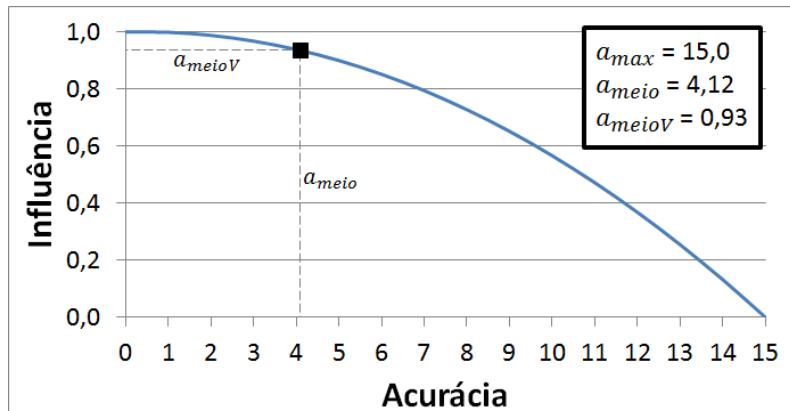
A Equação (3) define a influência da acurácia. Uma equação de segundo grau foi escolhida porque ela permite que pontos com raio de acurácia abaixo de certo limiar (ou seja, pontos com boa acurácia) sejam superestimados e contribuam mais para o *fitness*, enquanto pontos com raio de acurácia acima deste limiar (ou seja, pontos com acurácia ruim) sejam subestimados. O que será um ponto de acurácia "boa" ou "ruim" vai depender dos pontos que formam o conjunto S .

$$IA(S_i) = \alpha A(S_i)^2 + \left(\frac{-1 - \alpha \cdot a_{max}^2}{a_{max}} \right) A(S_i) + 1 \quad (3)$$

- S_i = Elemento do conjunto S .
 $A(S_i)$ = Acurácia do ponto selecionado S_i .
 $\alpha = \frac{-a_{meio} - a_{max}(a_{meioV} - 1)}{a_{max}a_{meio}(a_{max} - a_{meio})}$
 a_{max} = Maior raio de acurácia dentre os pontos do conjunto S (ou seja, a acurácia do ponto menos acurado).
 a_{media} = Acurácia média dos pontos do conjunto S .
 $a_{meio} = \frac{a_{max} - a_{media}}{2}$
 $a_{meioV} = 1 - \left(\frac{a_{meio}}{a_{max}} \right) a_{meioM}$
 $a_{meioM} = 0.9 / (a_{lim}/4)$ (valor definido empiricamente)
 a_{lim} = Acurácia limite (usada na filtragem, seção 2.2.1).

Esta equação foi deduzida a partir de três pontos (BOURNE, 2011) que definem o comportamento da curva: (i) se $A(S_i) = 0$, então $IA(S_i) = 1$; (ii) se $A(S_i) = a_{meio}$, então $IA(S_i) = a_{meioV}$; (iii) se $A(S_i) = a_{max}$, então $IA(S_i) = 0$. O valor destes três parâmetros é definido em tempo de execução, e por isso variam para cada conjunto S . Mesmo assim, a Figura 30 exemplifica como a equação se comporta para um dado conjunto S .

Figura 30 – Resultado da equação de influência da acurácia



Fonte: produção do próprio autor

Por ultimo, a Equação (4) define a influência da distância. Semelhante à equação de influência da acurácia, esta também é definida como uma equação de segundo grau. Assim, pontos de S a uma distância pequena da solução candidata C_x são superestimados e contribuem mais para o *fitness* desta solução, enquanto pontos mais distantes são subestimados e têm sua contribuição reduzida gradualmente.

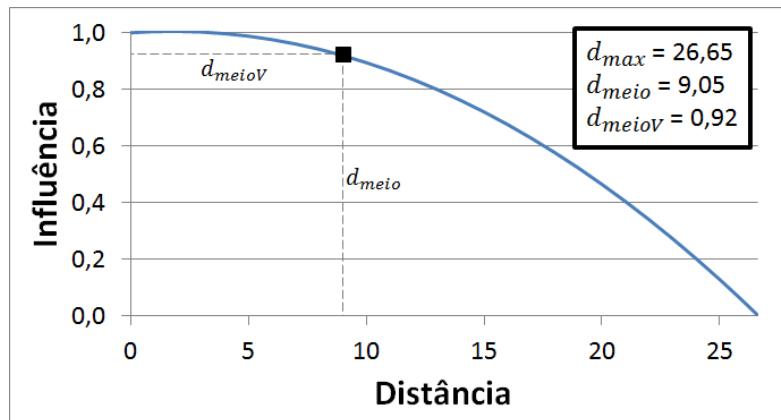
$$ID(C_x, S_i) = \beta D(C_x, S_i)^2 + \left(\frac{-1 - \beta \cdot d_{max}^2}{d_{max}} \right) D(C_x, S_i) + 1 \quad (4)$$

C_x	= Elemento do conjunto C .
S_i	= Elemento do conjunto S .
$D(C_x, S_i)$	= Distância entre C_x e S_i .
β	$\frac{-d_{meio} - d_{max}(d_{meiov} - 1)}{d_{max}d_{meio}(d_{max} - d_{meio})}$
d_{max}	= Maior distância entre C_x e os pontos do conjunto S (ou seja, a distância ao ponto mais afastado).
d_{media}	= Distância média entre C_x e os pontos do conjunto S .
d_{meio}	$= \frac{d_{max} - d_{media}}{2}$
d_{meiov}	$= 1 - \left(\frac{d_{meio}}{d_{max}} \right) d_{meiom}$
d_{meiom}	= $0.9/(a_{lim}/4)$ (valor definido empiricamente)
a_{lim}	= Mesmo parâmetro usado na equação IA.

Esta equação foi deduzida a partir de três pontos (BOURNE, 2011) que definem o comportamento da curva: (i) se $D(C_x, S_i) = 0$, então $ID(C_x, S_i) = 1$; (ii) se $D(C_x, S_i) = d_{meio}$, então $ID(C_x, S_i) = d_{meiov}$; (iii) se $D(C_x, S_i) = d_{max}$, então $ID(C_x, S_i) = 0$. O valor destes três parâmetros é definido em tempo de execução, variando para cada solução candidata C_x . Mesmo assim, a Figura 31 exemplifica como a equação se comporta para um dado conjunto S e solução candidata C_x .

Esta equação complementa as outras duas (influência do tempo e acurácia), pois não basta que uma solução candidata esteja próxima de muitos pontos, eles devem ser recentes e ter boa acurácia.

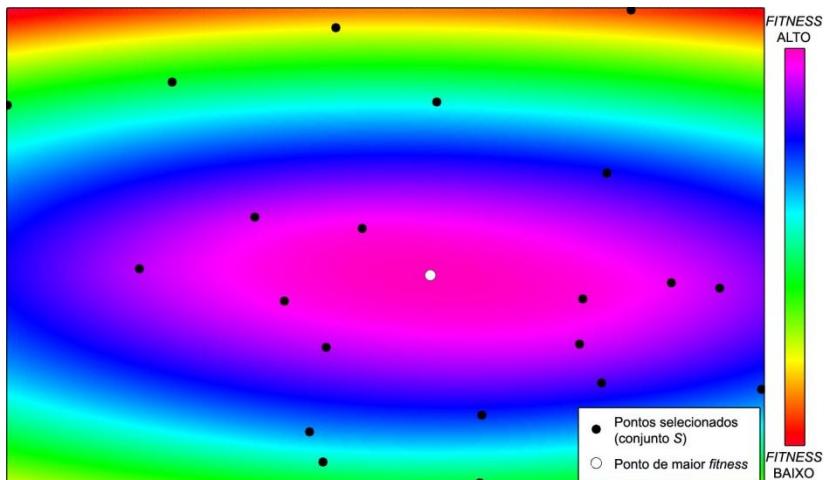
Figura 31 – Comportamento da equação de distância



Fonte: produção do próprio autor

Para demonstrar o desempenho da função de *fitness* como um todo, gerou-se a Figura 32. Nela é calculado o *fitness* de todos os pontos na área onde se encontra um determinado conjunto S (ou seja, todos os pontos no domínio do conjunto).

Figura 32 – Resultado da função de *fitness* para todos os pontos em um domínio



Fonte: produção do próprio autor

Os pontos com maior *fitness* tendem a se agrupar em uma pequena área, em geral no local onde há maior concentração de pontos selecionados. Isso é efeito da Equação (4), de influência da distância. Também se observa que o ponto de maior *fitness* está mais próximo dos pontos selecionados localizados a sua esquerda, e isso é provocado pela Equação (3), de influência da acurácia. A Equação (2) de influência do tempo de coleta também contribui para o resultado, mas neste exemplo seu efeito é pequeno, pois todos os pontos selecionados haviam sido coletados num curto intervalo de tempo. Sua influência poderá ser observada com mais facilidade na seção de testes, em especial nos testes 4.2.2 e 4.2.3.

3.1.3.4 Análise do conjunto S usando algoritmo genético

Um dos componentes necessários para a execução do algoritmo genético é o domínio, que especifica o espaço de coordenadas válidas que podem ser atribuídas às soluções candidatas em cada geração. Ele é definido como um retângulo formado a partir das coordenadas dos quatro pontos em S que se encontram mais distantes. Um exemplo de domínio foi apresentado na Figura 32, onde se percebe que o retângulo é delimitado pela localização dos pontos mais afastados.

Definido o domínio, é a vez de criar a primeira geração. Todas as g gerações contêm $z = 50$ soluções candidatas (valor escolhido empiricamente). Conforme apresentado anteriormente (seção 3.1.3.3), o conjunto das soluções candidatas de uma determinada geração é chamado de conjunto C , e cada elemento pode ser referenciado como C_x ($x = 1, 2, \dots, z$). Para a primeira geração, estas soluções candidatas devem ser criadas com base nos conjunto S . Contudo, geralmente S contém um número de pontos diferente do valor de z . A maneira escolhida para criar a primeira geração, então, é o equivalente a criar uma “geração zero”. A única diferença desta “geração” é que ela não obedece ao valor de z , e considera que cada ponto de S é uma solução candidata (independente do número de pontos em S). Ou seja, o conjunto C da “geração zero” é uma cópia do conjunto S .

Todas as gerações são criadas usando o mesmo procedimento: calcula-se o *fitness* das z soluções candidatas, ordenam-se os resultados, selecionam-se os descendentes e, a partir deles, cria-se a próxima geração de z soluções candidatas.

No caso da “geração zero”, independente de S ter mais ou menos que z elementos, a geração seguinte (ou seja, a primeira) será criada com z soluções candidatas. Se o conjunto S tiver mais de z elementos, os pontos em excesso irão sobrar. Por outro lado, se o conjunto S tiver menos de z elementos, alguns pontos acabarão sendo selecionados repetidas vezes até a primeira geração tenha z soluções candidatas. Como os pontos sofrem recombinação e mutação, mesmo que um único ponto seja selecionado várias vezes, existem grandes chances de que eles se tornem pontos diferentes.

Para calcular o *fitness* de cada solução candidata é utilizada a função de *fitness* definida anteriormente. A seguir, ordenam-se os resultados, preservam-se os $pop_{elite} = 2$ melhores pontos (elitismo, explicado na seção 2.4.1.4) e o restante é escolhido usando seleção por torneio, recombinação e mutação.

Usa-se elitismo com dois pontos porque desta forma os melhores resultados continuam sendo preservados, enquanto a seleção por torneio mantém a variabilidade genética alta. Assim, reduz-se a possibilidade de ficar preso em um pico de máximo local (HAUPT; HAUPT, 2004). A recombinação calcula um ponto intermediário entre dois pontos, já a mutação cria um novo ponto a uma distância de até $mut_{distância} = 5$ metros do ponto original. De acordo com a literatura, a chance de a recombinação acontecer deve ficar entre 60% e 95%, e a chance de mutação entre 0,2% e 5% (BÄCK, 1996). Assim, escolheu-se $rec_{chance} = 85\%$ e $mut_{chance} = 5\%$, respectivamente.

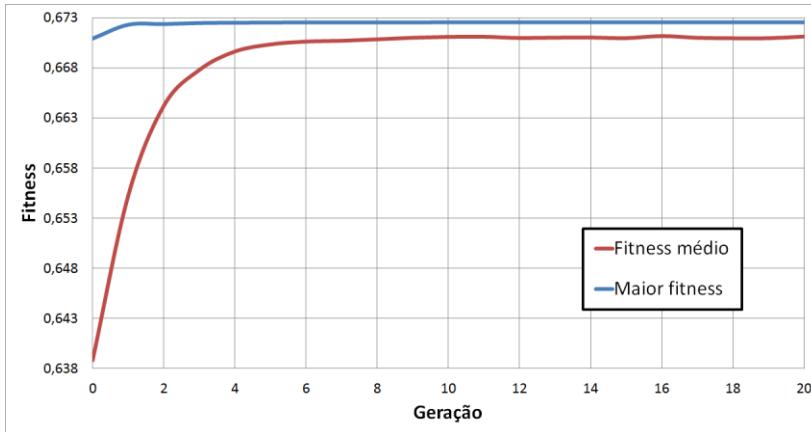
Todo este processo é repetido até a 20^a geração ser criada (ou seja, $g = 20$). O Quadro 03 resume o algoritmo genético, e a Figura 33 mostra o gráfico de convergência. Todos os parâmetros foram escolhidos empiricamente.

Quadro 03 – Resumo do algoritmo genético

Representação	Valores reais
Recombinação	Aritmética (85%)
Mutação	Adição de valor aleatório (5%)
Mecanismo de seleção	Elitismo (2) + Torneio (48)
Estratégia de sobrevivência	Por gerações (20)

Fonte: produção do próprio autor

Figura 33 – Gráfico de convergência



Fonte: produção do próprio autor

Apesar do *fitness* médio se manter praticamente o mesmo após aproximadamente a 10^a geração, ainda assim o maior *fitness* tende a aumentar, mesmo que muito pouco (geralmente quinta ou sexta casa decimal). Contudo, mesmo que o maior *fitness* aumente muito pouco entre uma geração e outra, a distância entre as coordenadas correspondentes pode ser consideravelmente grande. Isso ocorre quando existem picos de máximo local com *fitness* próximo ao da melhor solução.

Observou-se essa situação em áreas onde S têm poucos pontos. Isso significa que poucas trajetórias foram coletadas no local e, por isso, ainda não existe um acúmulo significativo de pontos no caminho percorrido pela maioria dos veículos motorizados. Como consequência, cada ponto de S pode agir como um pico de máximo local, causando o problema citado.

Uma forma de resolver o problema da grande quantidade de picos de máximo local seria coletar mais trajetórias, a fim de aumentar a quantidade de pontos nos conjuntos S . Porém, em um ambiente de execução real, a coleta de trajetórias não ocorreria de maneira uniforme ao longo de toda a área mapeada, e invariavelmente o problema iria ocorrer. Por esse motivo, essa solução não pode ser utilizada no contexto deste trabalho.

Assim, acredita-se que uma forma de resolver o problema é executar o algoritmo genético mais vezes, para aumentar a

probabilidade de que a solução candidata selecionada seja próxima da melhor solução possível – mesmo que isso faça o gráfico de convergência divergir do ideal.

Após a 20^a geração ser criada, termina a execução do algoritmo genético para o atual conjunto S , selecionado a partir dos pontos próximos a $P_{1,1}$. O centro da via nessa área é, então, a solução candidata da última geração que possui o melhor *fitness*. Este novo centro da via é complementado com as características de ângulo de movimento, acurácia e velocidade, calculados como uma média ponderada dos pontos do conjunto S em função de sua contribuição para o *fitness* da solução candidata escolhida. Além disso, são armazenados os códigos que identificam todos os pontos selecionados contidos no conjunto S . Esta informação (dado o código de um ponto de uma trajetória, descobrir qual centro da via ele ajudou a criar) é usada para criar o grafo direcionado, vide seção 3.1.4.

Para finalizar a criação do novo centro da via, seleciona-se uma área (um *buffer*, assim como na seção 3.1.3.2) ao redor do ponto para verificar se existe outro centro da via nas suas proximidades. Em caso positivo – e se as características dos centros forem similares – seus conjuntos S são unidos e o centro é recalculado, resultando em um único centro da via. Isso é necessário porque em certas situações, especialmente em vias de mão única com mais de uma faixa, a distância entre os pontos coletados pode gerar dois ou mais conjuntos S .

3.1.3.5 Continuação da execução do método

A próxima área a se identificar o centro da via é próxima ao ponto $P_{1,2}$, ou seja, $P_{j,k+1}$. Como o GPS grava um ponto por segundo, mesmo após a etapa de pré-processamento a chance de $P_{1,2}$ estar próximo de $P_{1,1}$ é alta. Nesse caso, $P_{1,2}$ provavelmente já foi utilizado para calcular o centro da via na área próxima a $P_{1,1}$, pois seu círculo de acurácia intersectou o *buffer* (vide seção 3.1.3.2). Assim, não é necessário calcular o centro da via na área próxima a $P_{1,2}$. A mesma lógica é válida para todos os pontos no conjunto S . Por isso, eles são marcados como usados, para que não sejam analisados novamente.

Desta forma, antes de calcular o centro da via na área próxima a qualquer ponto $P_{j,k}$ ($j \in q; k \in n$), primeiro se verifica

se ele já foi usado. Se sim, então ele é ignorado. Esta verificação é repetida até encontrar um ponto ainda não processado. Calcula-se então o centro da via na área próxima a este ponto.

Após analisar cada ponto de cada trajetória, considera-se que os centros de todas as vias coletadas foram encontrados.

3.1.4 Criação de grafo direcionado

A criação do grafo direcionado composto por todos os centros das vias identificados no passo anterior (seção 3.1.3) segue o pseudocódigo do Quadro 04, apresentado a seguir.

Quadro 04 – Pseudocódigo do método de criação do grafo

Entrada:

centros_das_vias: centros das vias identificados
lista_ordem: lista ordenada das trajetórias processadas

```

01 Inicializa via como uma lista vazia;
02 Inicializa grafo como um grafo direcionado vazio;
03 PARA CADA trajetória em lista_ordem FAÇA
04      $C_a$  é o centro da via criado do 1º ponto da trajetória
05     Adiciona  $C_a$  em via
06     PARA CADA ponto  $P_i$  da trajetória, a partir do 2º FAÇA
07          $C_b$  é o centro da via criado do ponto  $P_i$ 
08         SE distância entre  $C_a$  e  $C_b$  < limite ENTÃO
09             Procurar em centros_das_vias outros centros
10                localizados entre  $C_a$  e  $C_b$ ;
11                PARA CADA centro da via encontrado FAÇA
12                    SE diferença entre ângulo do centro da via e
13                         $C_a$  e  $C_b$  < limite ENTÃO
14                        Adiciona centro da via em via;
15                    Adiciona  $C_b$  em via;
16                    SENÃO
17                        Adiciona via em grafo;
18                        Esvazia via, mantendo apenas  $C_b$ ;
19                         $C_a$  é  $C_b$ ;
```

Saída: *grafo*.

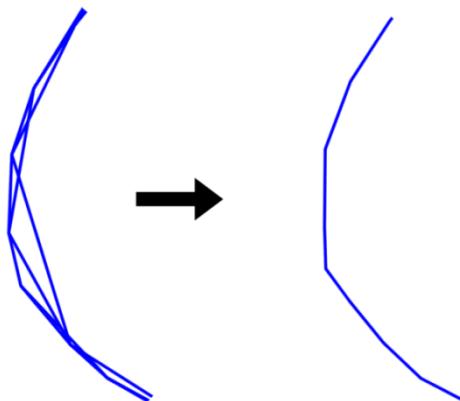
Fonte: produção do próprio autor

Primeiro, encontram-se os centros da via criados a partir de uma dada trajetória. A ordem dos pontos da trajetória determina a ordem que os centros da via são conectados no grafo.

Devido à etapa de filtragem, uma trajetória pode não ser capaz de criar todos os centros de uma via. Por isso, para cada par de centros encontrados (C_a e C_b no Quadro 04), verifica-se se a distância entre eles fica dentro de um limite – distância máxima de $dist_{pontos} = 120$ metros, e diferença de ângulo de movimento menor que $\theta_{max} = 100$ graus. Em caso positivo, verifica-se se existem outros centros entre C_a e C_b . Se existir, então serão adicionados ao grafo como parte da mesma via. Quando todas as trajetórias forem verificadas, o grafo direcionado estará criado.

Contudo, é preciso executar um passo adicional, de pós-processamento do grafo. Isso é necessário porque, antes de criar o grafo, as únicas informações que o método tem sobre o que é ou não uma via são a posição e angulação dos centros e a ordem dos pontos das trajetórias coletadas. Por isso, o grafo gerado apresenta alguns problemas. Estes problemas não afetam o uso do grafo em algoritmos que encontram o menor caminho entre dois pontos, por exemplo, apenas a visualização do mapa gerado. Um destes problemas é quando, em uma curva, um ponto se conecta a mais de um ponto à sua frente, como a Figura 34 mostra.

Figura 34 – Solução de problema do grafo



Fonte: produção do próprio autor

Isso acontece nos locais onde uma única trajetória não conseguiu gerar todos os centros que formam uma via, e ao procurar por outros centros nas proximidades (linha 09 do Quadro 04), algum centro não é encontrado (distância pouco acima da permitida). Na sequência, quando outra trajetória que deu origem aos pontos daquela via for analisada, a mesma coisa acontece para um conjunto diferente de pontos, e assim sucessivamente, até que todas as trajetórias sejam analisadas. O resultado final é o apresentado na Figura 34, à esquerda. Para corrigir isto são analisadas as diferenças entre os ângulos dos pontos envolvidos e a quantidade de pontos no conjunto S de cada centro. A análise do ângulo diz qual das conexões é provavelmente a certa em relação à continuidade da geometria da curva, enquanto a análise do conjunto S de cada centro identifica qual conexão representa o caminho percorrido por mais trajetórias. A combinação dos dois resultados permite decidir o caminho que será mantido. O lado direito da Figura 34 apresenta a solução alcançada.

Também é necessário remover do grafo os trechos com apenas dois pontos conectados. Conforme observado empiricamente, trechos isolados desta forma têm grande chance de ser apenas ruído. Por último são eliminados do grafo os pontos que, em decorrência do pós-processamento, ficaram desconectados de qualquer via.

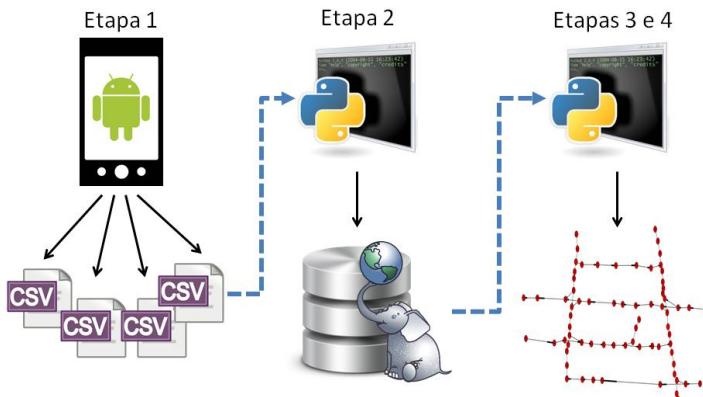
Opcionalmente, um passo extra que pode ser executado é converter cada via do grafo em uma *spline*. As *splines*, neste contexto, suavizam as curvas, de modo que tenham uma aparência contínua. A finalidade de criar uma versão do mapa com *splines* é apenas tornar a visualização do mapa mais agradável ao usuário final (SCHOLZ, 2013). Para criar as *splines*, utilizou-se o programa desenvolvido no trabalho de Scholz (2013). Este programa cria *splines* de Bézier, que recebem como parâmetro dois pontos de interpolação e dois de aproximação (SCHOLZ, 2013). Para comparar as diferenças entre a versão com e sem *splines*, ambas foram geradas e testadas no capítulo de análise dos resultados (ver capítulo 4).

3.2 Implementação

O método foi implementado em três partes distintas. A primeira corresponde à primeira etapa do método, de coleta dos

dados. A segunda parte corresponde à segunda etapa do método, de pré-processamento. Por último, a terceira parte corresponde a terceira e quarta etapas do método, de identificação dos centros das vias e criação do grafo. Essa divisão foi definida com base na saída de cada etapa: a primeira tem como saída arquivos CSV, a segunda adiciona os dados processados a um banco de dados geográfico, a terceira retorna uma estrutura de dados e a quarta processa a estrutura e gera o grafo, armazenando-o em um banco de dados de grafo. A Figura 35 ilustra essa divisão. Como os resultados intermediários são armazenados em disco, não há necessidade de executar cada uma das etapas imediatamente após acabar a anterior.

Figura 35 – Etapas da implementação



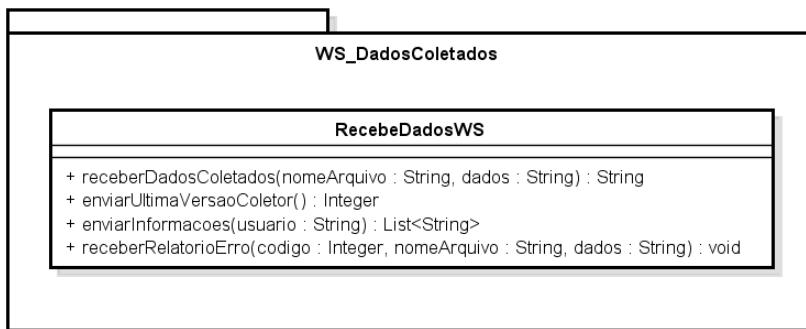
Fonte: produção do próprio autor

A primeira parte da implementação, de coleta dos dados, engloba o aplicativo cliente para *smartphones* e o servidor *WebService*. Ambos são implementados em Java.

O funcionamento do aplicativo coletor é apresentado em detalhes no apêndice A, mas é interessante destacar que os usuários do aplicativo é que decidem quando as trajetórias coletadas são enviadas para o servidor. O *WebService*, por outro lado, fica aguardando que algum usuário envie arquivos CSV com as trajetórias. Quando isso acontece, ele recebe o arquivo e armazena em uma pasta local. Além do método para receber trajetórias, o *WebService* possui também um método que retorna

o número da versão mais recente do aplicativo coletor. Esse método é utilizado para auxiliar o processo de atualização do aplicativo. A Figura 36 mostra em um diagrama de classe a interface do WebService com seus métodos.

Figura 36 – Métodos do WebService



Fonte: produção do próprio autor

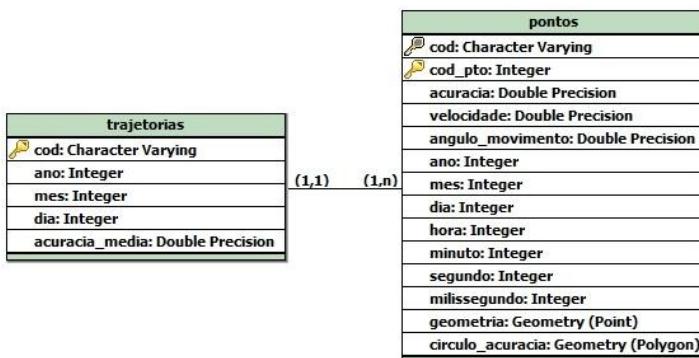
A segunda parte da implementação está contida em um programa escrito em Python. Esse programa lê as trajetórias contidas nos arquivos CSV e, para cada uma delas, executa todos os passos do pré-processamento: identificação do meio de transporte, redução do ruído e compressão (vide seção 3.1.2).

Conforme explicado na seção 2.1.1 a Árvore de Decisão usada para identificar o meio de transporte é gerada usando a ferramenta WEKA. Ao final do processo de geração, o WEKA exibe o resultado no formato de texto tabulado. Para usar a Árvore no programa implementado em Python, este texto tabulado é manualmente convertido em uma função nesta linguagem. Outra questão interessante com relação à identificação do meio de transporte é que as características extraídas do acelerômetro que envolvem cálculo de Transformada Discreta de Fourier foram calculadas em Python usando a biblioteca externa chamada NumPy (NUMPY, 2013).

Ao final de todo o pré-processamento, as trajetórias filtradas são armazenadas em um banco de dados no Sistema de Gerenciamento de Banco de Dados (SGBD) PostgreSQL. Por padrão este SGBD não suporta dados geográficos, por isso é utilizada a extensão PostGIS. Ela adiciona ao PostgreSQL os

tipos de dados e funções necessários para manipular dados espaciais e geográficos. O esquema do banco de dados contém duas tabelas, como ilustrado na Figura 37.

Figura 37 – Modelo lógico do banco de dados



Fonte: produção do próprio autor

Conforme pode ser observado na Figura 37, a tabela “*trajetorias*” contém um código usado para referenciar cada trajetória e os dados necessários para ordená-las (vide seção 3.1.3.1). A tabela “*pontos*” armazena cada ponto de cada trajetória filtrada, e é acessada pelo método de identificação dos centros das vias para recuperar as informações dos pontos a serem processados. Além disso, o campo “*circulo_acuracia*” armazena o círculo de acurácia usado para encontrar os pontos que formam o conjunto S , como explicado na seção 3.1.3.2. A identificação destes pontos é feita através de uma única consulta ao banco de dados, usando funções geométricas implementadas pelo próprio PostGIS. Portanto, pode-se considerar que é feita de maneira eficiente.

A terceira e última parte da implementação também está contida em um programa desenvolvido na linguagem Python. Usando como entrada as trajetórias no banco de dados, ele identifica os centros das vias e cria um grafo direcionado. Isso é feito com ajuda da biblioteca externa chamada Shapely, que permite fazer em Python operações geográficas semelhantes às feitas pelo PostGIS (GILLIES, 2013). Também é utilizada a

biblioteca externa chamada NetworkX, que permite criar o grafo direcionado (NETWORKX, 2013).

Ao final de todo o processamento, o programa tem três saídas. Uma é o grafo direcionado, que pode ser inserido em um banco de dados de grafo. Outra saída é o mapa em um arquivo de tipo Keyhole Markup Language (KML), que é aberto por programas como o Google Earth e usado para visualizar o resultado. A terceira saída é o mapa com *splines*, que pode ser criado como um passo adicional após a criação do grafo. O resultado deste passo também pode ser armazenado em um arquivo no formato KML.

4 Análise dos Resultados

Para validar o método proposto e verificar a eficiência de sua implementação, testes devem ser realizados. Dado o objetivo do trabalho – propor um método que gere e atualize mapas rodoviários sem necessitar de interferência manual – a metodologia de avaliação precisa propor testes capazes de avaliar se os resultados condizem com o esperado.

A seção 4.1 detalha a metodologia utilizada. Na sequência, as próximas seções mostram os resultados obtidos em cada um dos testes realizados. Por fim, são apresentadas as considerações sobre os testes.

4.1 Metodologia de Avaliação

A metodologia de avaliação dos resultados é centrada na lógica dedutiva. Nesse método de raciocínio, conhecimentos gerais servem como ponto inicial, a partir dos quais se inferem conhecimentos específicos (MARCONI; LAKATOS, 2003). No contexto do presente trabalho, isso significa que os cenários de teste deverão ser locais com estruturas rodoviárias complexas. Se os testes apresentarem resultados satisfatórios, então deduz-se que o mesmo nível de qualidade provavelmente também será obtido em locais com estruturas rodoviárias mais simples.

Os testes foram realizados com dados coletados na cidade de Joinville – Santa Catarina. Aproximadamente dez pessoas participaram da coleta de trajetórias, usando oito modelos diferentes de *smartphones*. Ao todo 4.983 arquivos de trajetória foram enviados ao servidor, contabilizando um total de 1.262.368 pontos coletados. O espaço em disco necessário para armazenar todas as trajetórias foi 1,1GB. O tempo total gasto coletando trajetórias foi de cerca de 400 horas. Devido ao ruído presente nos dados não é possível saber exatamente qual a distância total coletada, porém a soma do comprimento de todas as trajetórias após a filtragem foi de aproximadamente 6.500 km.

Outro ponto da metodologia diz respeito à forma de avaliação dos resultados. Neste trabalho são realizados testes de natureza qualitativa e quantitativa.

Os testes qualitativos visam demonstrar visualmente a eficiência da solução proposta. Como os mapas são utilizados,

em última instância, por seres humanos, a inspeção visual é fundamental para identificar se os mapas gerados se mantêm fiéis à realidade das vias encontradas na região ou se apresentam alguma anomalia ocorrida durante o processo de geração ou atualização. Para analisar estas questões, três testes qualitativos foram realizados. Detalhes sobre cada um destes testes são apresentados na seção 4.2.

Por outro lado, os testes quantitativos avaliam a solução proposta com base em números. Ou seja, demonstram efetivamente a diferença entre o resultado esperado e o resultado obtido. Para isso, um teste quantitativo foi realizado em três cenários diferentes. Detalhes sobre o teste são apresentados na seção 4.3.

4.2 Testes qualitativos

O primeiro teste qualitativo (4.2.1) compara o mapa gerado com mapas de referência: Bing Maps (MICROSOFT, 2014), Google Maps (GOOGLE, 2014) e OpenStreetMap (OSM, 2013a). Esta comparação permite verificar se o mapa é fiel à realidade. Três locais serviram de cenário para a execução deste teste: rotatória de acesso à UDESC e vias próximas, trecho de uma rodovia de alto movimento – BR-101 – e suas vias de acesso e retorno; e vias do centro da cidade. Além da comparação visual dos resultados finais obtidos, são apresentadas também informações referentes à quantidade de trajetórias processadas em cada cenário, tempo de execução e números referentes a cada passo da etapa de pré-processamento. Estas informações ajudam a dar uma noção maior das características do cenário de teste. Adicionalmente, são apresentadas imagens que ilustram, na prática, o resultado de cada etapa do método.

O segundo teste (4.2.2) analisa versões incrementais do mapa para observar se o método proposto atualiza-o da maneira desejada. Ou seja, se ele adiciona novas vias ao mapa e aprimora a acurácia dos centros das vias levando em consideração a data de coleta das trajetórias.

Já o terceiro teste (4.2.3) analisa o comportamento do método ao gerar o mapa de uma certa região da cidade antes e depois da prefeitura alterar o sentido de direção de vias daquela

área. Com este teste, verifica-se a capacidade do método em detectar alterações de maior escala e refleti-las no mapa.

O cenário para o segundo e terceiro testes é o mesmo: a Rua Max Colin e outras vias próximas.

4.2.1 Teste de comparação com imagens de satélite

Conforme comentado na seção anterior, este teste foi executado em três locais. Os dois primeiros, rotatória de acesso à UDESC e trecho da BR-101, são locais com estruturas rodoviárias consideradas complexas. Eles apresentam curvas com diferentes angulações, vias de mão única e mão dupla e vias próximas umas das outras. Devido a esta variedade de comportamentos, são locais adequados para testar o funcionamento do método. Finalmente, o terceiro cenário de teste, com vias do centro da cidade, visa demonstrar a validade do método por dedução. Ou seja, se os outros dois cenários testaram o método em casos considerados "extremos", ele não deve apresentar dificuldade em um local mais simples.

Para estes testes, as bases de mapa utilizadas para comparação foram Bing Maps, Google Maps e OSM. O motivo de utilizar várias bases é porque as coordenadas exatas de cada centro de via certamente sofrem alterações entre uma base e outra. Isso significa que as vias possuem uma geometria e um posicionamento ligeiramente diferente em cada mapa. Como não se sabe qual das bases é mais acurada, comparar o mapa gerado pelo método proposto aos mapas das três bases permite ter uma visão melhor de sua acurácia.

Cenário 1

O primeiro cenário de teste apresenta a região onde está localizada a rotatória de acesso à UDESC e algumas das vias próximas. Foram coletadas 914 trajetórias que passam por esta área, contendo um total de 117.965 pontos. O pré-processamento dessas trajetórias demorou 46 segundos.

Para a identificação do meio de transporte, as trajetórias são divididas em trechos com 2 pontos (aproximadamente 40 informações de aceleração, vide seção 2.1.1). Ao todo 49.181

trechos foram criados, cada trajetória com cerca de 54 trechos, e 71,75% foram classificados como de veículos motorizados.

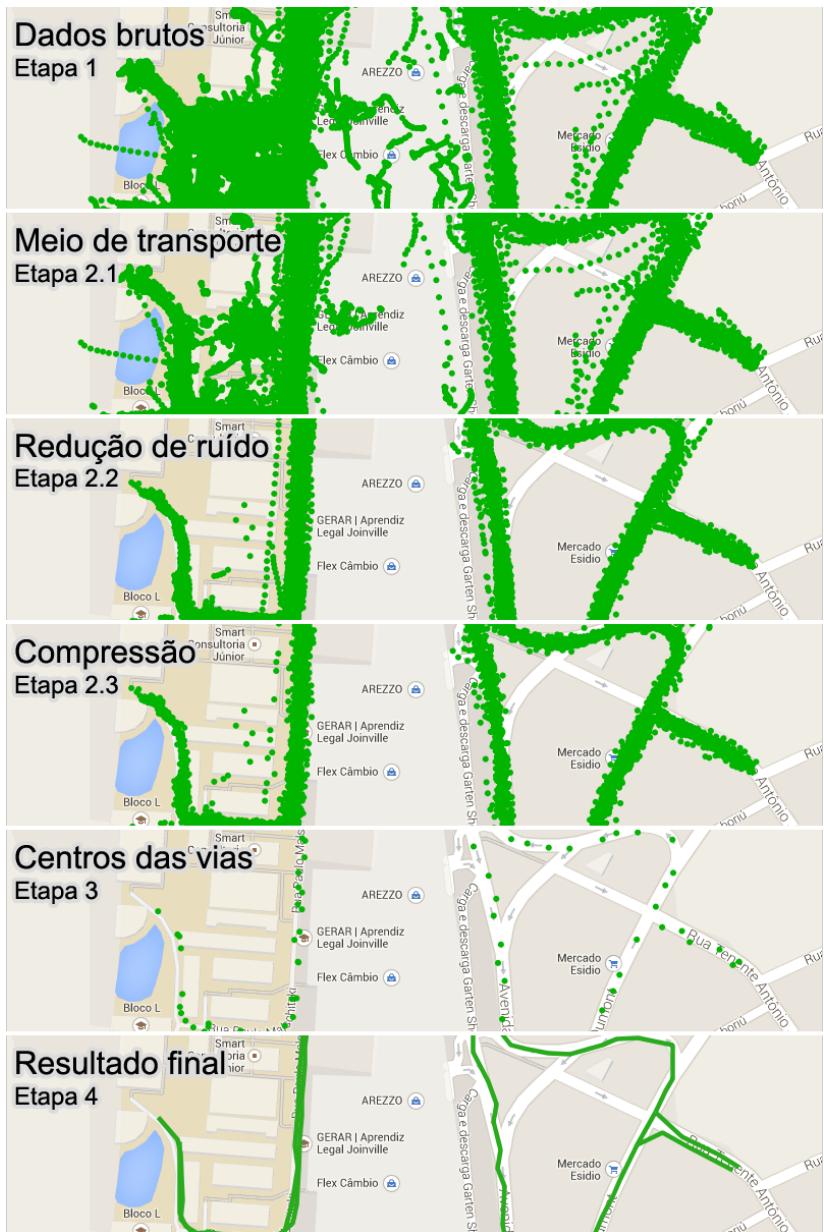
O tamanho médio das trajetórias processadas foi de 131 pontos (vide seção 3.1.2). Em média 41,22% dos pontos foram eliminados na filtragem pelo meio de transporte, 38,79% foram eliminados durante a redução de ruído e 10,01% dos pontos foram eliminados na compressão das trajetórias. Isso significa que aproximadamente 9,98% dos pontos de cada trajetória foram considerados de qualidade adequada para a geração de mapas.

Uma alta percentagem de pontos foi descartada na filtragem pelo meio de transporte porque parte das trajetórias eram de pessoas andando pela UDESC. Como pessoas andam devagar e a frequência de coleta é alta, muitos pontos a pé haviam sido coletados. Em relação à percentagem de pontos descartados na redução de ruído, o principal problema foram pontos com acurácia ruim, devido à quantidade de árvores e prédios no local. Além disso, vários pontos foram coletados por carros trafegando lentamente a procura de vaga para estacionar ou por carros estacionados (usuário demorou a parar a coleta de trajetórias ou esqueceu). Finalmente, a percentagem de pontos eliminados na compressão das trajetórias foi suficiente para reduzir o tempo de criação dos centros das vias e geração do grafo em 43,56% – de 12 minutos e 56 segundos para 7 minutos e 18 segundos. Além disso, comparando o resultado com e sem compressão em relação à mesma via no OSM (assim como no teste da seção 4.3), as duas versões ficaram predominantemente parecidas. A versão sem compressão teve diferença média de 2,26 metros e desvio padrão de 2,04 metros, enquanto a outra teve diferença média de 2,31 metros e desvio padrão de 1,73 metros.

Das 914 trajetórias coletadas neste cenário, apenas 361 continham pontos de boa qualidade e foram inseridas no banco de dados ao final do pré-processamento. Após processá-las, 136 centros das vias foram criados.

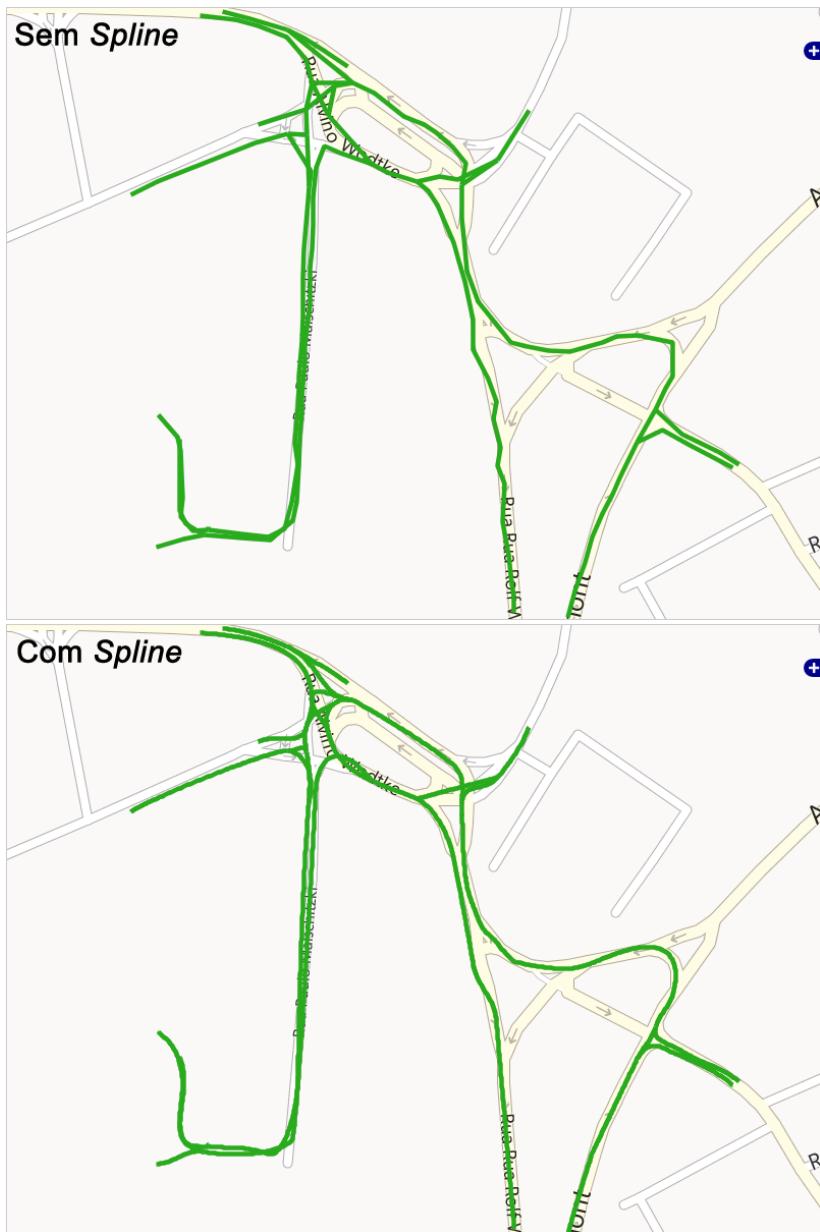
A Figura 38 resume o resultado obtido após aplicar cada etapa do método neste cenário. Na sequência, a Figura 39, Figura 40 e Figura 41 comparam o mapa gerado aos mapas de referência, e a Figura 42 destaca a localização de dois pontos onde o método teve um comportamento incorreto.

Figura 38 – Teste 1, Cenário 1: Etapas do método



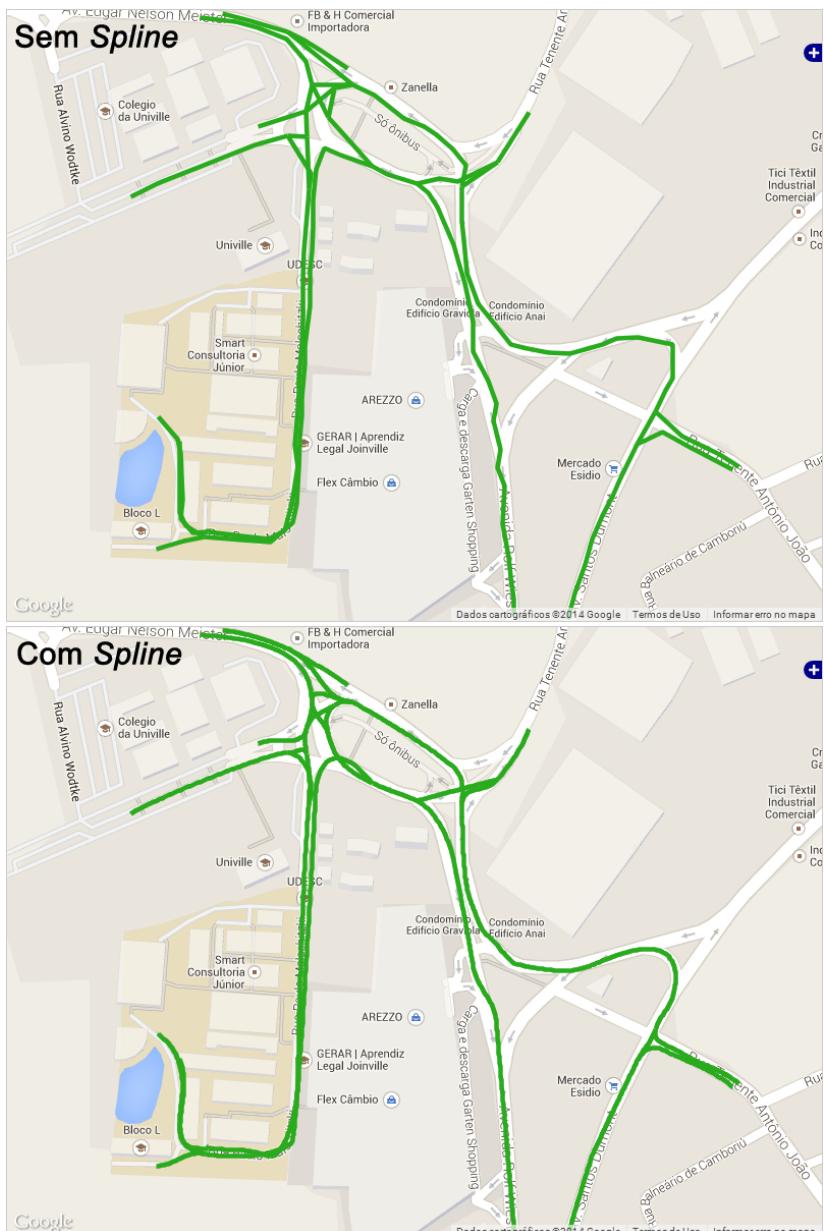
Fonte: produção do próprio autor

Figura 39 – Teste 1, Cenário 1: Comparando ao Bing Maps



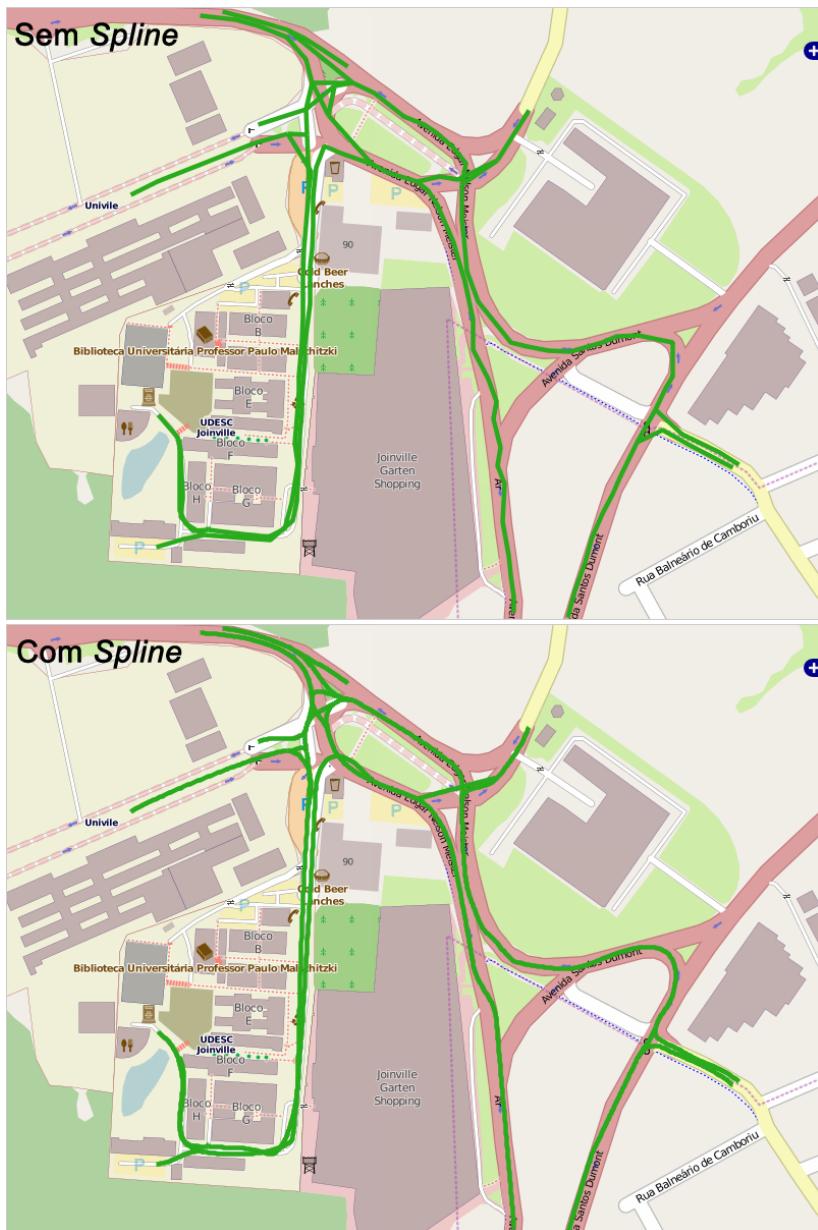
Fonte: produção do próprio autor

Figura 40 – Teste 1, Cenário 1: Comparando ao Google Maps



Fonte: produção do próprio autor

Figura 41 – Teste 1, Cenário 1: Comparando ao OpenStreetMap



Fonte: produção do próprio autor

Figura 42 – Erro no mapeamento



Fonte: produção do próprio autor

Como se percebe na Figura 39, Figura 40 e Figura 41, de modo geral o mapa gerado tem uma geometria semelhante a dos três mapas. Em algumas partes a versão do mapa com *splines* é mais parecido, enquanto em outras é a versão sem *splines* que se destaca. De modo geral, a versão com *splines* tem uma geometria mais parecida com a do Google Maps, enquanto que a versão sem *splines* se parece mais com o mapa do OSM.

Como se percebe na Figura 42, em A o método gerou uma via que não existe. Isto foi causado por pontos de baixa qualidade que não foram identificados na filtragem. Em B, o trecho marcado existe e está localizado corretamente, porém o método não o conectou ao restante da via. Isto ocorreu porque (1) aquela é uma via com duas faixas e (2) a maior parte das trajetórias coletadas foi de veículos que fizeram a curva à esquerda ao invés de seguir na direção do trecho B. Dada essa combinação de fatores, e considerando a distância e o ângulo de movimento dos centros da via, a conexão entre B e o restante da via foi considerada errônea pela etapa de pós-processamento.

Além disso, deve-se observar que o método não diferencia faixas em uma mesma via. O resultado tende para o lado com mais dados coletados. É por isso que algumas vias ficaram com formato sinuoso. Contudo, coletando mais trajetórias, acredita-se que a quantidade de pontos coletados em cada faixa irá se equilibrar e o resultado será melhor.

Cenário 2

O segundo cenário de teste é um trecho da BR-101 e suas vias de retorno. Foram coletadas 170 trajetórias que passam por esta área, contendo um total de 11.223 pontos. O pré-processamento dessas trajetórias demorou 16 segundos.

Para a identificação do meio de transporte, as trajetórias são divididas em trechos com 2 pontos (vide seção 2.1.1). Ao todo 4.436 trechos foram criados, cada trajetória com cerca de 26 trechos, e 99,98% foram classificados como de veículos motorizados.

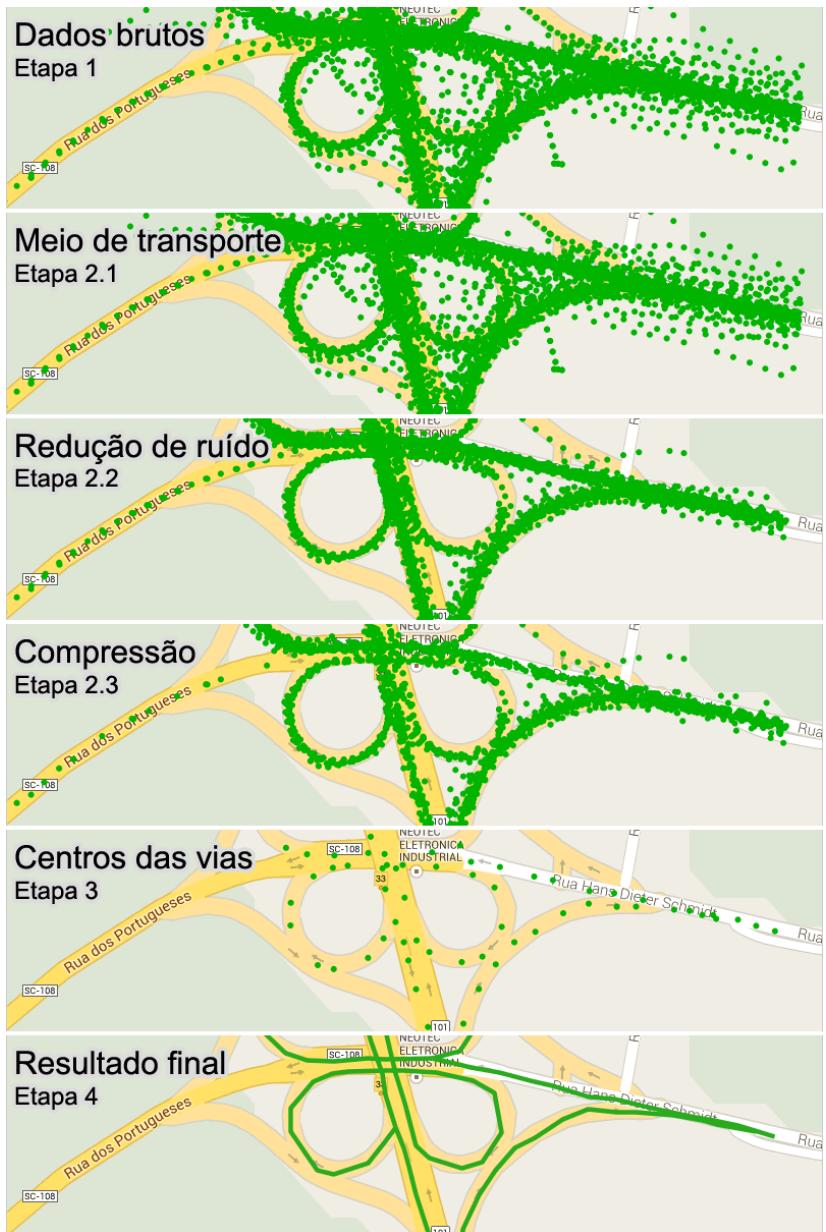
O tamanho médio das trajetórias processadas foi de 68 pontos (vide seção 3.1.2). Em média 17,54% dos pontos foram eliminados na filtragem pelo meio de transporte, 38,31% na redução de ruído e 18,44% na compressão das trajetórias. Isso significa que 25,71% dos pontos de cada trajetória foram considerados de qualidade adequada para a geração de mapas.

Todas as trajetórias coletadas na BR-101 são de veículos motorizados, então os pontos eliminados na filtragem pelo meio de transporte se devem a falsos negativos. Outro motivo foram casos onde, devido a limitações no acesso ao sensor de acelerômetro, estes dados não foram coletados corretamente e, por isso, os pontos foram descartados. O principal problema na redução de ruído foi pontos com acurácia ruim. Finalmente, os pontos eliminados na compressão das trajetórias reduziram o tempo de criação dos centros das vias e geração do grafo em 43,88% – de 2 minutos e 19 segundos para 1 minuto e 18 segundos. Comparando o resultado com e sem compressão em relação à mesma via no OSM (assim como no teste da seção 4.3), a versão com compressão foi ligeiramente melhor. A versão sem compressão teve diferença média de 1,94 metros e desvio padrão de 1,56 metros, enquanto a outra teve diferença média de 1,82 metros e desvio padrão de 1,53 metros.

Das 170 trajetórias coletadas neste cenário, apenas 97 continham pontos de boa qualidade e foram inseridas no banco de dados ao final do pré-processamento. Após processá-las, 93 centros das vias foram criados.

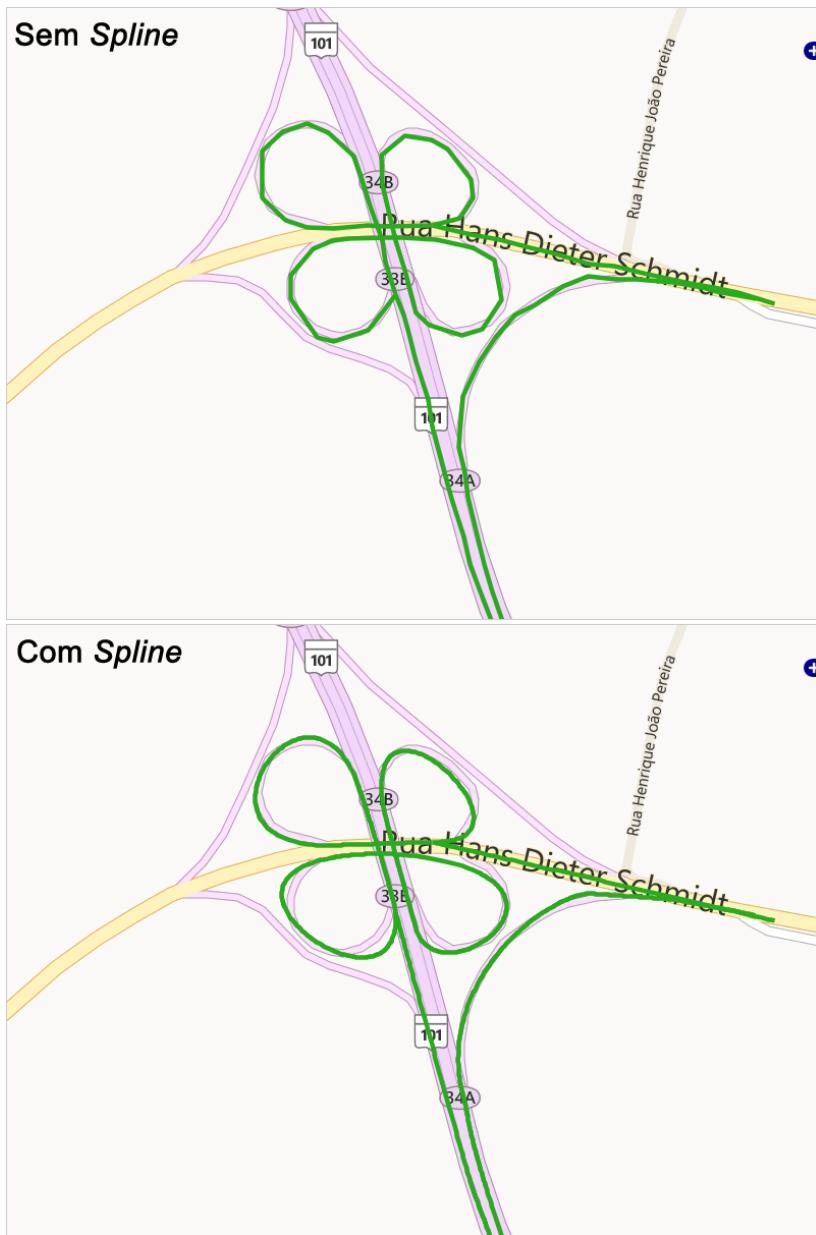
A Figura 43 resume o resultado obtido após aplicar cada etapa do método neste cenário. Já a Figura 44, Figura 45 e Figura 46 compararam o mapa gerado aos mapas de referência.

Figura 43 – Teste 1, Cenário 2: Etapas do método



Fonte: produção do próprio autor

Figura 44 – Teste 1, Cenário 2: Comparando ao Bing Maps



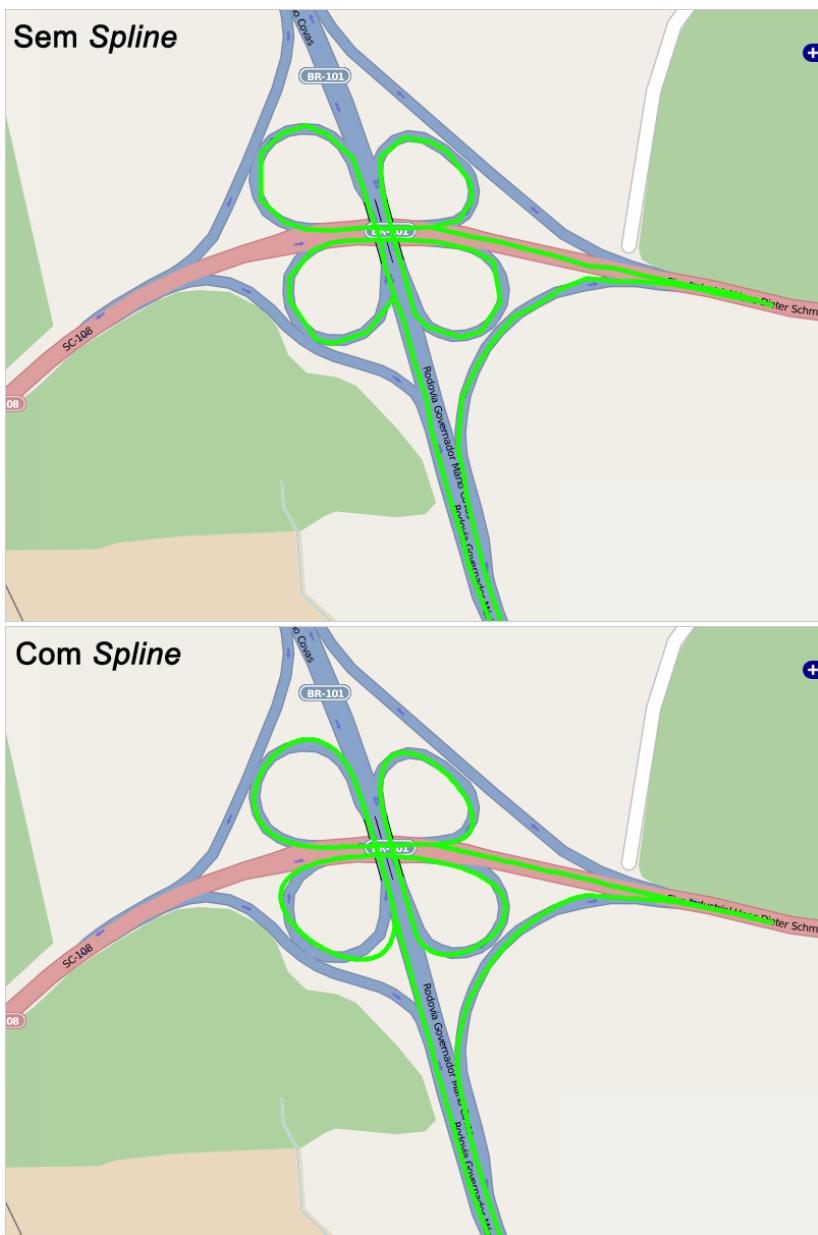
Fonte: produção do próprio autor

Figura 45 – Teste 1, Cenário 2: Comparando ao Google Maps



Fonte: produção do próprio autor

Figura 46 – Teste 1, Cenário 2: Comparando ao OpenStreetMap



Fonte: produção do próprio autor

A principal dificuldade em mapear essa área está no local onde a BR-101 se cruza com as vias de retorno, que passam por baixo. Porém, como se percebe, separar os pontos pela angulação de movimento resolveu este problema. Diferente do primeiro cenário, aqui não existe nenhum erro considerável no mapa gerado, há apenas uma pequena deformação na versão com *splines*, em uma das vias de retorno. Vale destacar que apesar de terem sido coletadas trajetórias em algumas das outras vias mostradas nas figuras, a quantidade foi muito pequena. Como existe um parâmetro definindo quantos pontos devem existir em uma região para que seja identificado um centro da via (vide seção 3.1.3.2), a quantidade de pontos coletados não foi suficiente e o método descartou-os considerando que eram ruído.

Cenário 3

O terceiro cenário mostra vias do centro da cidade de Joinville – Santa Catarina. Foram coletadas 295 trajetórias que passam por esta área, contendo um total de 36.249 pontos. O pré-processamento dessas trajetórias demorou 24 segundos.

No primeiro passo do pré-processamento, a identificação do meio de transporte, as trajetórias são divididas em trechos com 2 pontos (vide seção 2.1.1). Ao todo 14.083 trechos foram criados (cada trajetória com cerca de 48 trechos), e 91,88% foram classificados como de veículos motorizados.

O tamanho médio das trajetórias processadas foi de 133 pontos. Em média 16,24% dos pontos foram eliminados na filtragem pelo meio de transporte, 54,97% foram eliminados durante a redução de ruído e 16,33% dos pontos foram eliminados na compressão das trajetórias. Isso significa que aproximadamente 12,46% dos pontos de cada trajetória foram considerados de qualidade adequada para a geração de mapas.

Grande parte das trajetórias neste cenário de teste foram coletadas usando veículos motorizados, e isso explica a baixa percentagem de pontos descartados na filtragem pelo meio de transporte. A alta percentagem obtida na filtragem durante a redução de ruído se deve a dois fatores. O primeiro é a quantidade de pontos com acurácia ruim, devido principalmente à interferência causada pelos prédios que existem na região. Já

o segundo é a quantidade de pontos coletados com o veículo motorizado parado ou trafegando lentamente. Isto se deve à ocorrência de engarrafamentos no trânsito daquele lugar. Finalmente, a percentagem de pontos eliminados na compressão das trajetórias foi capaz de reduzir o tempo de criação dos centros das vias e geração do grafo em 53,34% – de 3 minutos e 17 segundos para 1 minuto e 26 segundos. Comparando o resultado com e sem compressão em relação à mesma via no OSM (assim como no teste da seção 4.3), a versão sem compressão foi ligeiramente melhor. A versão sem compressão teve diferença média de 2,38 metros e desvio padrão de 2,26 metros, enquanto a outra teve diferença média de 2,77 metros e desvio padrão de 2,20 metros.

Das 295 trajetórias coletadas neste cenário, apenas 188 continham pontos de boa qualidade e foram inseridas no banco de dados ao final do pré-processamento. Após processá-las, 73 centros das vias foram criados.

A Figura 47 resume o resultado obtido após aplicar cada etapa do método neste cenário. Na sequência, a Figura 48, Figura 49 e Figura 50 compararam o mapa gerado aos mapas de referência.

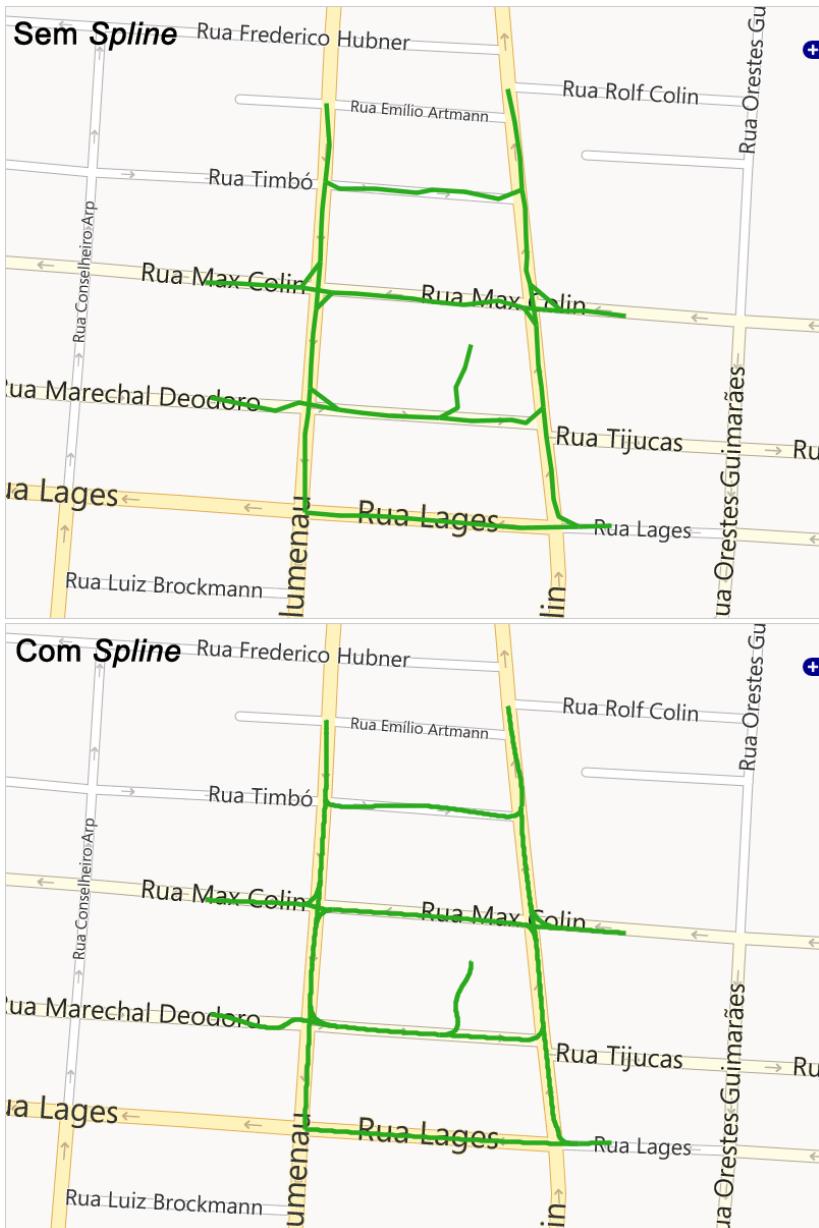
Este é um exemplo de cenário considerado mais simples que os demais, pois apresenta algumas das características dos cenários complexos – vias que se cruzam, vias com mais de uma faixa – porém as curvas com ângulos mais bem definidos e a distância à outras vias de mesma direção de movimento tornam este cenário mais simples que os anteriores. A intenção em executar esse teste é mostrar que a lógica dedutiva da metodologia de avaliação é válida.

Figura 47 – Teste 1, Cenário 3: Etapas do método



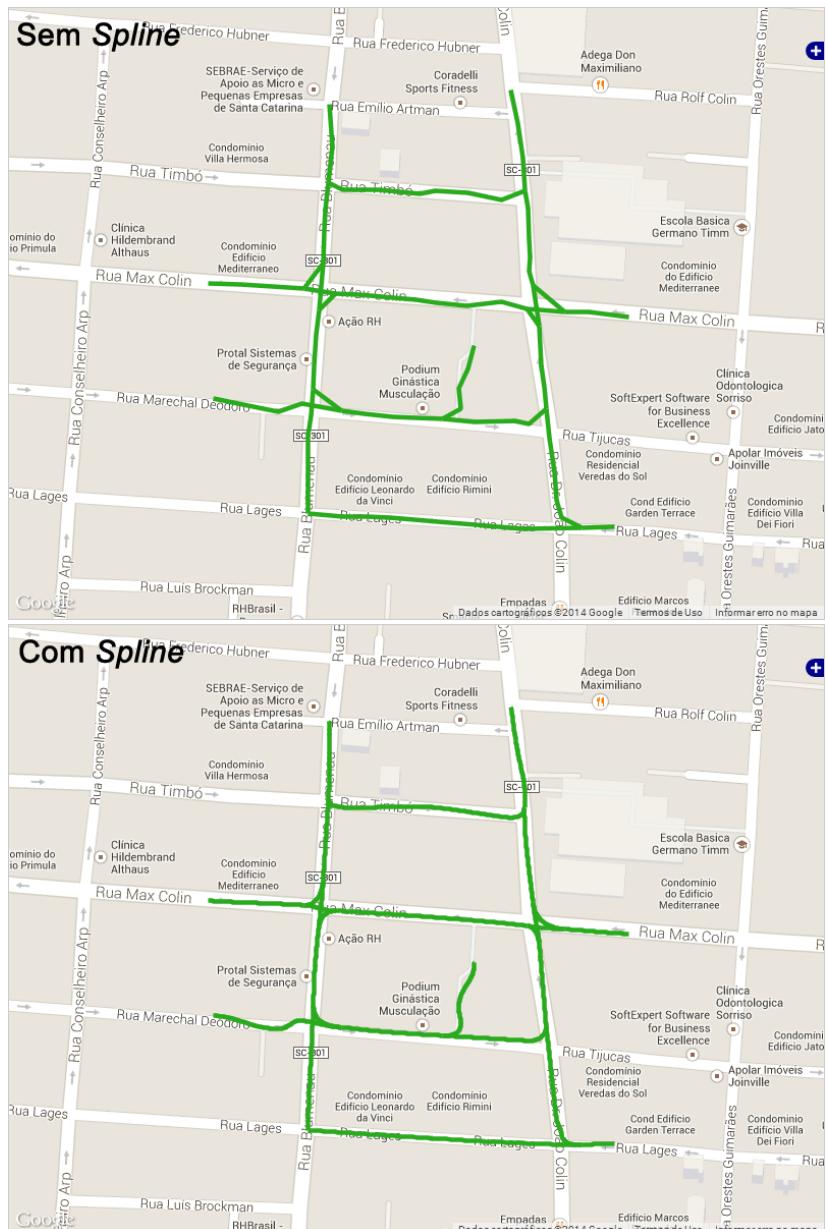
Fonte: produção do próprio autor

Figura 48 – Teste 1, Cenário 3: Comparando ao Bing Maps



Fonte: produção do próprio autor

Figura 49 – Teste 1, Cenário 3: Comparando ao Google Maps



Fonte: produção do próprio autor

Figura 50 – Teste 1, Cenário 3: Comparando ao OpenStreetMap



Fonte: produção do próprio autor

Como se percebe na Figura 48, Figura 49 e Figura 50, os resultados são muitos similares aos apresentados pelos mapas de referência. Tanto a versão com e sem *spline* demonstram geometria semelhante a todos os mapas, em especial o Bing Maps.

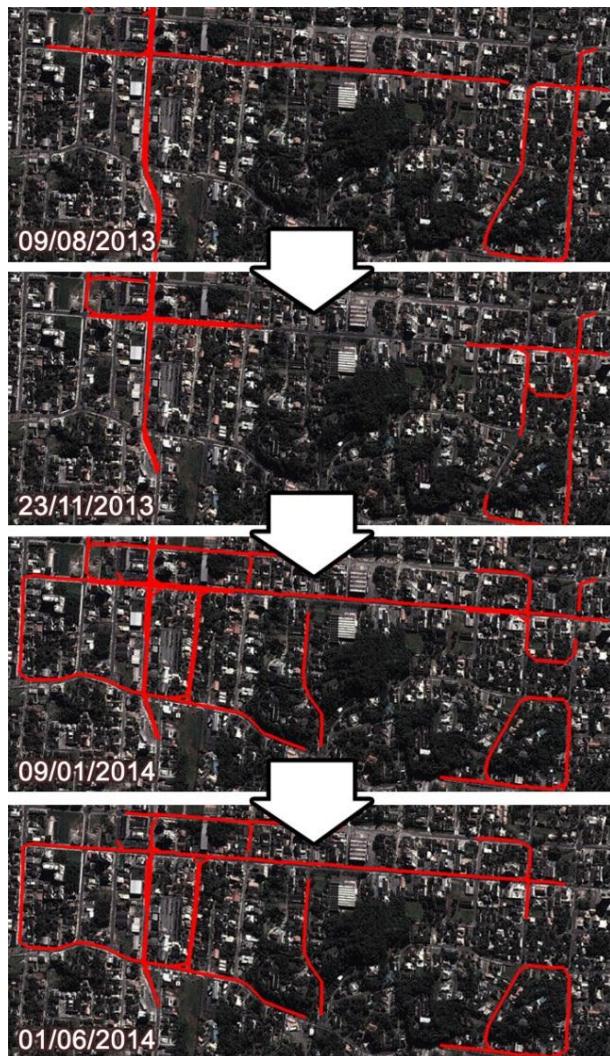
É interessante destacar que neste cenário o método mapeou uma via que é, na verdade, o estacionamento de um estabelecimento comercial. Dentre os três mapas de referência, apenas o Google Maps também mapeou esta “via”.

4.2.2 Teste de adição de novas vias ao mapa

Na medida em que as trajetórias são coletadas o método deve ser executado para gerar um mapa atualizado. Quando trajetórias suficientes passarem por uma via, ela deve aparecer no mapa. Este teste visa verificar este comportamento.

O cenário utilizado foi a Rua Max Colin e outras vias próximas. Nesta região foram coletadas trajetórias entre fevereiro de 2013 e janeiro de 2014. Para realizar o teste, simulou-se a execução do método em quatro datas: 09/08/2013, 23/11/2013, 09/01/2014 e 01/06/2014. As três primeiras datas representam quando foi feita a coleta de uma grande quantidade de trajetórias naquela região. Já a última data, apesar de ser um momento no futuro, representa o período quando pouco mais de 180 dias tinham se passado após a data da segunda coleta. Como citado na seção 3.1.3.2, 180 dias é o valor do parâmetro (t_{max}) que define quando um ponto é antigo demais e deve ser removido do processo. Desta forma, nas quatro execuções do método, o resultado seria: (1) todos os dados até 09/08/2013; (2) dados de 23/11/2013 com grande peso, dados de 09/08/2013 com peso médio, dados até abril excluídos; (3) dados de 09/01/2014 com grande peso, dados de 23/11/2013 com peso médio, dados até junho excluídos; (4) apenas dados de 09/01/2014, o restante excluído. Na Figura 51 é notável as mudanças que ocorrem no mapa entre uma execução e a outra. Novas vias são adicionadas a cada execução, e vias muito antigas e com pouco movimento foram sendo excluídas.

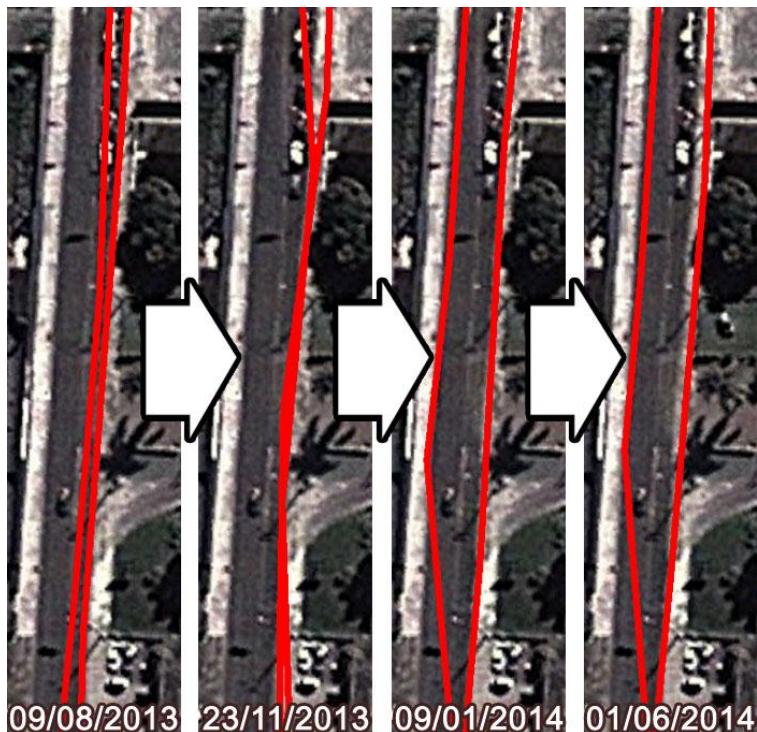
Figura 51 – Teste 2: Atualização do mapa



Fonte: produção do próprio autor

Neste mesmo teste também é possível observar que, conforme novas trajetórias foram enviadas ao método, os centros das vias tiveram sua posição alterada para corresponder à nova realidade. A Figura 52 ilustra isso em uma das vias da região.

Figura 52 – Teste 2: Atualização dos centros das vias



Fonte: produção do próprio autor

Vale destacar que a acurácia dos centros das vias é dependente dos pontos coletados, e pontos mais recentes tem maior peso. Sendo assim, dados de baixa qualidade que escapam da filtragem podem piorar o mapa, ao mesmo tempo em que pontos mais recentes de alta qualidade reduzem o peso de eventuais pontos ruins coletados anteriormente.

4.2.3 Teste de alteração da direção de movimento de uma via

O principal motivo de coletar trajetórias na região da Rua Max Colin e vias próximas é porque a prefeitura de Joinville – Santa Catarina – decidiu alterar o sentido de movimento de algumas vias daquela área (ANOTÍCIA, 2013). Por este motivo, decidiu-se realizar testes nesta região para averiguar o

comportamento do método antes e depois da mudança. Os dados coletados até 23/11/2013 representam a região antes da mudança entrar em vigor, e os dados coletados em 09/01/2014 já representam a região após a alteração no trânsito. Das quatro datas usadas no teste anterior, a primeira (09/08/2013) não é necessária para este teste, visto que representa o trânsito na mesma situação que 23/11/2013. Sendo assim, nas três execuções do método, o resultado seria: (1) mapa de 23/11/2013, antes de mudar o trânsito; (2) mapa de 09/01/2014, pouco após mudar o trânsito; (3) simulação do mapa de 01/06/2014, muito tempo após mudar o trânsito. A Figura 53 mostra os resultados observados.

Figura 53 – Teste 3: Mudança no trânsito de uma via



Fonte: produção do próprio autor

O mapa em 23/11/2013 mostra que naquela via (Rua Max Colin) era possível trafegar nos dois sentidos. No mapa de 09/01/2014, pouco tempo após a alteração ter sido realizada, há uma mistura entre as trajetórias anteriores e as novas. Isso ocorre porque a diferenciação por ângulo de movimento – que foi considerada necessária nos testes da seção 4.2.1 – executa

antes da análise da data de coleta. A seguir, quando o método tenta analisar as trajetórias do sentido que deixou de existir, tudo que ele vê é que as trajetórias ainda não são antigas o suficiente e, portanto, devem ser processadas e adicionadas ao mapa. Contudo, no mapa simulado do dia 01/06/2014, essas trajetórias já se tornaram antigas e por isso são desconsideradas pelo método, não influenciando no resultado. Assim, o mapa é gerado apenas com base nas trajetórias coletadas após a mudança.

Como se percebe, a atualização do mapa depende dos parâmetros escolhidos (vide apêndice B). Para a configuração definida, onde dados com mais de 180 dias são excluídos, foi necessário simular uma data no futuro para observar o resultado final. Contudo, ao reduzir consideravelmente este valor, o mesmo resultado pode ser alcançado em menos tempo. Por outro lado, reduzir este valor significa que outras vias, que não sofreram alteração no trânsito, mas que foram mapeadas a partir de dados não tão recentes, poderão ser eliminadas do mapa, como mostrado no teste anterior.

4.3 Teste quantitativo – Erro em relação ao OpenStreetMap

Das três bases de mapas utilizadas no teste da seção 4.2.1, o OSM disponibiliza seus dados publicamente e permite que qualquer pessoa possa copiá-los livremente. Por este motivo, foram copiados os dados de algumas das vias mapeadas e foi feita a medição da diferença entre o resultado do método proposto e o OSM. Dessa forma, deseja-se medir o quanto fiel à realidade é o mapa gerado. Para fazer esta medição foi desenvolvido um programa que, usando as funções do PostGIS, encontra a menor distância entre cada centro da via e a via do OSM. A partir destes cálculos, descobre-se qual o ponto com maior erro, o erro médio e o desvio padrão. Para os três valores, quanto menores eles forem melhor será o resultado do método.

Este teste foi feito em três locais diferentes da cidade. Para os dois primeiros testes, os cenários são os mesmos do teste 4.2.1: rotatória de acesso à UDESC e trecho da BR-101. O terceiro teste, assim como em 4.2.1, também é no centro da cidade. Porém, escolheu-se uma via diferente, com comprimento maior. Os testes foram executados duas vezes, para comparar a

via do OSM com o mapa na versão com e sem *spline*. A Figura 54 mostra os três cenários sem *spline* utilizados neste teste.

Figura 54 – Teste Quantitativo: Cenários



Fonte: produção do próprio autor

No cenário 1, da rotatória de acesso à UDESC, foi escolhido o trecho com maior quantidade de curvas com ângulos diferentes. A Figura 55 mostra como a comparação foi feita. Devido ao tamanho do trecho escolhido, essa figura exibe apenas o trecho onde ocorreu a maior diferença. A Tabela 05 mostra os números calculados.

Figura 55 – Teste Quantitativo, Cenário 1

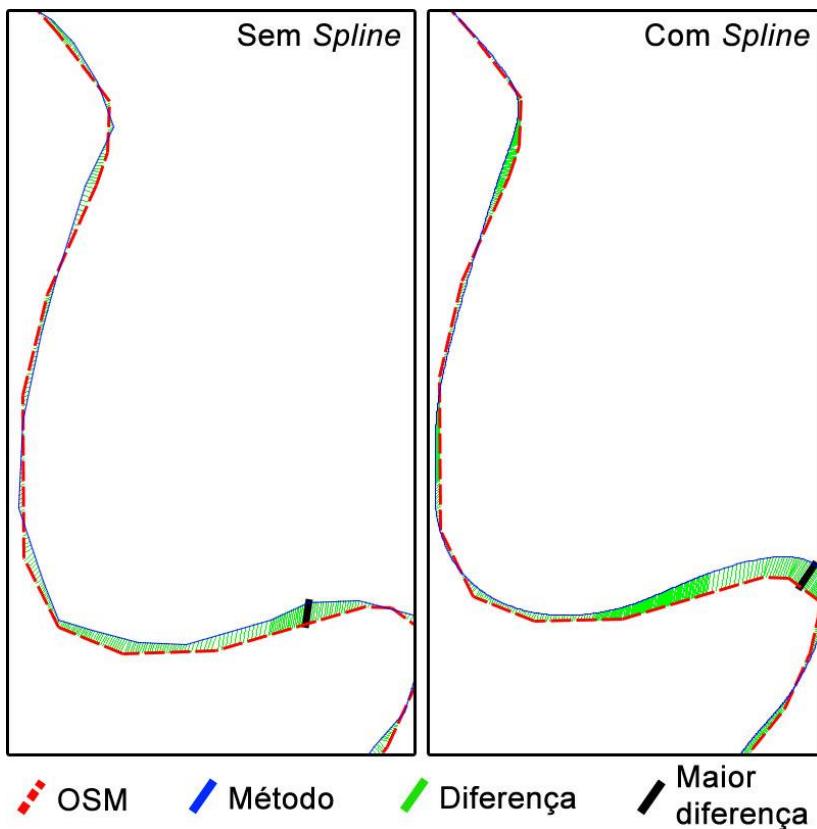


Tabela 05 – Teste Quantitativo, Cenário 1: Diferença calculada

Mapa	Diferença média	Desvio padrão	Maior diferença
Sem splines	2,31m	1,73m	8,52m
Com splines	2,68m	2,24m	9,48m

Fonte: produção do próprio autor

Aqui, a versão sem *spline* se mostrou pouco melhor em todos os valores calculados.

No segundo cenário, da BR-101, escolheu-se o trecho mais longo, que passa por todas as vias de retorno. A Figura 56 mostra o local de maior diferença e a Tabela 06 os resultados.

Figura 56 – Teste Quantitativo, Cenário 2

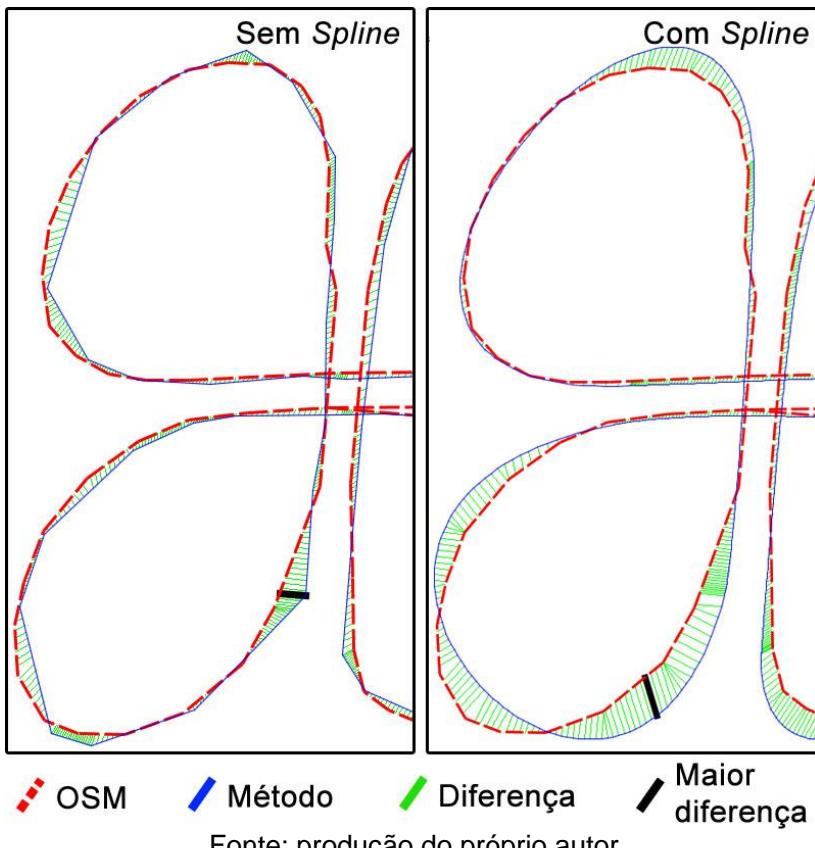


Tabela 06 – Teste Quantitativo, Cenário 2: Diferença calculada

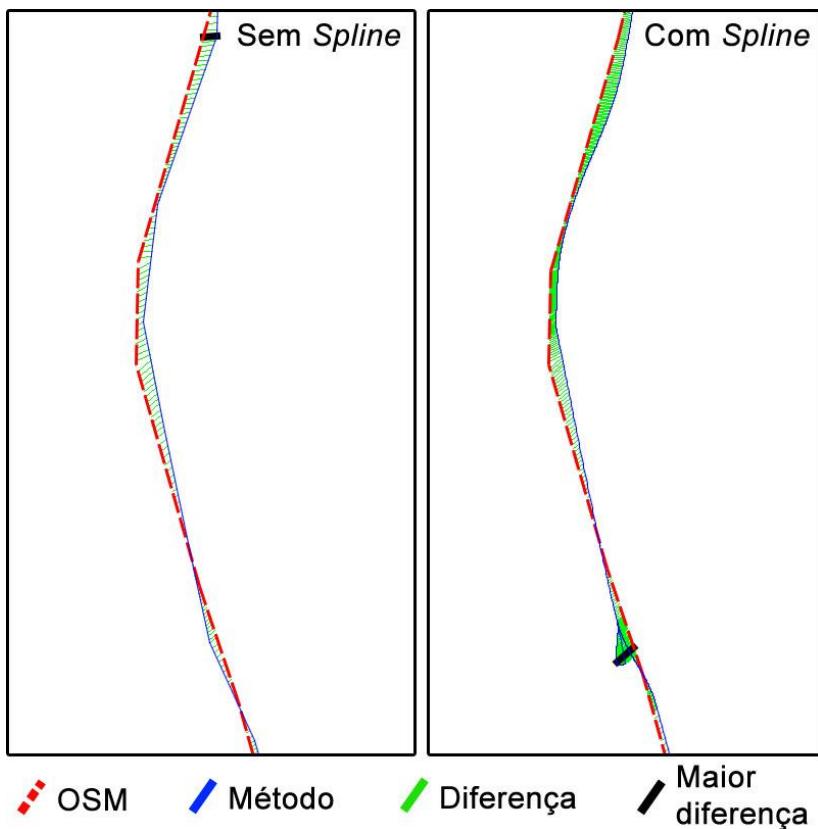
Mapa	Diferença média	Desvio padrão	Maior diferença
Sem splines	1,82m	1,53m	10,34m
Com splines	2,99m	2,91m	16,08m

Fonte: produção do próprio autor

Aqui, a versão sem *splines* teve um desempenho consideravelmente melhor.

Por último, o terceiro cenário (centro) representa uma via com menos curvas. A Figura 57 mostra o local de maior diferença e a Tabela 07 os resultados.

Figura 57 – Teste Quantitativo, Cenário 3



Fonte: produção do próprio autor

Tabela 07 – Teste Quantitativo, Cenário 3: Diferença calculada

Mapa	Diferença média	Desvio padrão	Maior diferença
Sem splines	1,53m	1,05m	4,85m
Com splines	2,25m	1,40m	6,72m

Fonte: produção do próprio autor

Como se observa, deformações na *spline* fizeram seus resultados serem ligeiramente piores que os da versão sem *spline*.

Por outro lado, conforme comentado na seção 3.1.4, o mapa com *splines* é um passo opcional, para tornar a

visualização mais agradável ao usuário final. Além disso, a via do OSM não usa *splines*, e isso pode ter contribuído para a piora os resultados.

De qualquer maneira, acredita-se que mesmo as diferenças encontradas na versão com *splines* são aceitáveis. A pior diferença média foi de 2,99 metros com desvio padrão de 2,91 metros, no segundo cenário. Esses valores são considerados pequenos, visto que mesmo vias de uma faixa costumam ter largura igual ou maior que isso. Esse resultado está de acordo com as observações do teste 4.2.1, onde o mapa gerado mostrou que, na maioria das vezes, os centros das vias estão corretamente posicionados no espaço que Bing Maps, Google Maps e OSM consideram ser a via. A maior diferença foi 16,08 metros, também no segundo cenário, e acredita-se que pode ser reduzida à medida que mais trajetórias forem coletadas.

4.4 Considerações sobre os testes

De modo geral, os testes selecionados e realizados mostram que o método executa conforme proposto e alcança resultados satisfatórios. Adicionalmente, a metodologia de avaliação dedutiva reforça que resultados com a mesma qualidade podem ser alcançados em outros cenários, como demonstrado, por exemplo, no cenário 3 do teste 4.2.1.

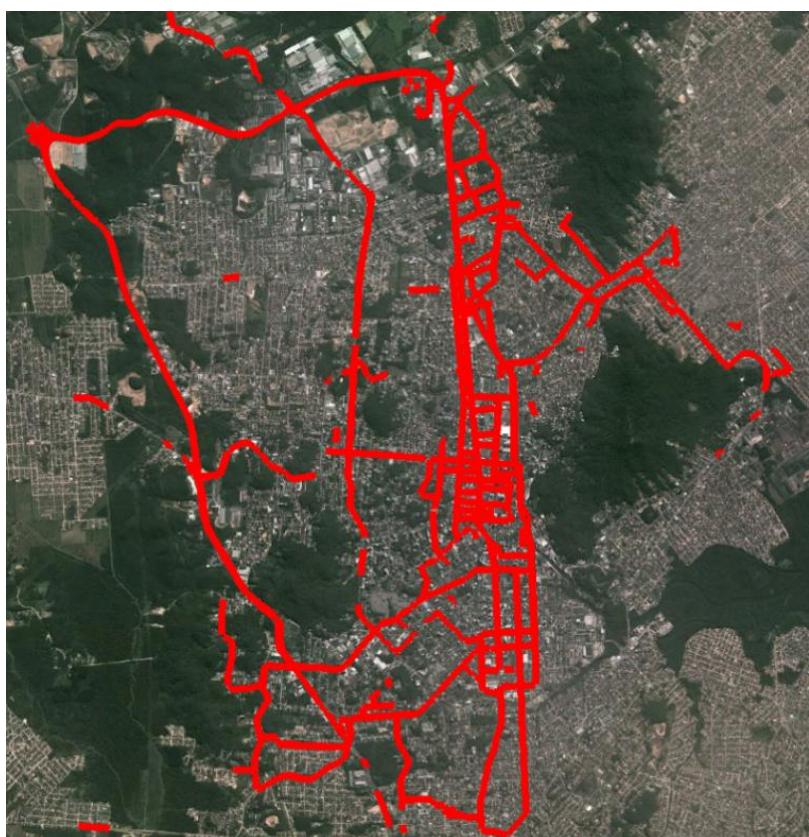
Os testes 4.2.2 e 4.2.3 analisam a atualização do mapa, que é feita de maneira integrada ao método proposto. Como pode ser visto, a atualização do mapa depende que novas trajetórias sejam coletadas frequentemente, visto que o método leva em consideração a data de coleta das trajetórias. Caso contrário, existe o risco de que vias sejam perdidas. Contudo, isto pode ser controlado de acordo com a configuração dos parâmetros do método (vide apêndice B).

Já os testes 4.2.1 e 4.3 focam na acurácia geográfica dos resultados. Mesmo que a diferença visual entre o mapa gerado e as outras bases de mapas seja pequena – e os números calculados confirmam isso – é interessante destacar que, apesar destes resultados, a existência desta diferença não significa necessariamente que o mapa gerado é que está errado. Conforme comentado anteriormente, as coordenadas dos pontos das bases de mapas estão ligeiramente deslocadas, e é difícil descobrir qual a coordenada correta. Isto pode ser observado em

ambos os testes no segundo cenário (trecho da BR-101), onde a versão sem *spline* teve sua maior diferença em relação ao OSM: 10,34 metros. Apesar dessa diferença, aquele mesmo ponto aparenta estar corretamente posicionado de acordo com o mapa do Google Maps (Figura 45). De qualquer forma, a comparação permite concluir que as diferenças na geometria das vias mapeadas pelo método e dos mapas de referência é pequena.

Na implementação feita, o tempo médio de execução para todos os dados coletados foi de 3 horas e 25 minutos – 100 pontos por segundo; 1.900 quilômetros por hora. A Figura 58 mostra toda a área que foi processada.

Figura 58 – Área completa mapeada



Fonte: produção do próprio autor

Apesar de ser um tempo considerado alto, há de se destacar que não foi feito um trabalho de otimização da implementação. Além disso, o algoritmo não utiliza programação paralela, que poderia reduzir consideravelmente o tempo de execução.

5 CONCLUSÃO

Mapas rodoviários digitais são ferramentas de suporte que ganharam papel fundamental no dia-a-dia da população. Por esta razão, é essencial que eles reflitam a realidade da melhor maneira possível. O alto custo da criação de mapas por métodos fotogramétricos levou à busca de alternativas como o GPS, que se destaca pela enorme quantidade de dispositivos móveis com receptor GPS integrado disponíveis no mercado. Um tipo de dispositivo assim são os *smartphones*, que contam com outros sensores além do receptor GPS e são cada vez mais comuns entre o grande público. Partindo destas considerações, o objetivo do presente trabalho foi propor um método estruturado que limpe, analise e enriqueça dados obtidos por *smartphones* a fim de possibilitar a geração e atualização contínua de mapas rodoviários digitais acurados, utilizando algoritmos evolutivos.

Existem alguns trabalhos similares apresentados na literatura. Entretanto, algumas questões os diferenciam do método aqui proposto. Em especial, três pontos chave se destacam: (i) a inexistência de preocupação com os diferentes meios de transporte que uma pessoa pode usar enquanto coleta dados com seu *smartphone*; (ii) a falta de discussão sobre como é feita a atualização do mapa; (iii) e a pouca importância à informação temporal associada aos dados coletados pelo GPS. Sendo assim, a contribuição do presente trabalho foi propor um método que trate destas questões. Adicionalmente, o presente trabalho também se diferencia dos demais na utilização de algoritmo genético para identificar os pontos que representam os centros das vias. A coleta de dados com *smartphones* também pode ser considerado um diferencial, visto que apenas um trabalho também diz usar este tipo de dispositivo, mas assume que os dados são coletados usando um veículo automotivo.

Os objetivos específicos de apresentar soluções para identificação do meio de transporte, redução de ruído e compressão das trajetórias foram atingidos. A revisão bibliográfica resultou em um levantamento das soluções existentes, e as análises e comparações realizadas na sequência permitiram encontrar as mais adequadas para alcançar o resultado pretendido. O objetivo específico de desenvolver o

método proposto também foi atingido. Usando a metodologia de pesquisa Pesquisa-Ação, uma solução inicial foi criada e continuamente aperfeiçoada num processo cílico, até alcançar o estado aqui apresentado, considerado satisfatório pelos testes realizados no capítulo 4. Finalmente, os objetivos específicos de implementação do método e da ferramenta para coleta das trajetórias também foram atingidos. Conforme foi possível perceber nos testes realizados, o mapa gerado difere pouco de outros mapas já existentes e disponíveis ao público – no pior caso, diferença média de 2,99 metros e desvio padrão de 2,91 metros – e a atualização contínua do mapa e o refinamento das vias ocorrem da maneira esperada.

Um dos principais pontos positivos do método apresentado é a atualização do mapa. Os pontos de cada lugar são ponderados de acordo com a diferença entre as datas de coleta e o resultado disso é que locais onde trajetórias são coletadas frequentemente serão atualizados rapidamente, e locais de menor trânsito serão mapeados à medida que novas trajetórias são coletadas. Com o passar do tempo, pontos com uma idade superior a um valor limite são considerados antigos – desatualizados – e, portanto, são eliminados para não prejudicar a qualidade do mapa gerado. Dessa forma, o método busca sempre gerar mapas que refletem a realidade com base nas trajetórias coletadas.

Além disso, a identificação do meio de transporte facilita o trabalho de atualização do mapa, visto que não é necessário fazer esta filtragem manualmente. Ao mesmo tempo, isso também abre espaço para que mais pessoas contribuam para a base de trajetórias que alimenta o mapa, pois a coleta de dados pode ser feita sem nenhum tipo de supervisão. O aplicativo coletores desenvolvido para *smartphones* pode, inclusive, continuar a coleta mesmo quando não está executando em primeiro plano. Isso significa que um usuário pode optar por iniciar a coleta de trajetórias, colocar seu *smartphone* no bolso e continuar sua atividade sem se preocupar, pois o aplicativo continuará coletando as trajetórias e poderá enviá-las ao servidor.

Outro ponto positivo é que o método não necessita de intervenção manual. Assim, pode-se realizar periodicamente sua execução e ter o mapa constantemente atualizado, tanto sua apresentação visual como o grafo.

Por outro lado, um dos pontos negativos do método é que ele possui muitos parâmetros: 24 na implementação atual, conforme comentado no apêndice B. Contudo, certamente é possível modificar o método para reduzir este número, seja alterando as funções implementadas ou automatizando a configuração de vários destes parâmetros. Outra questão é que os algoritmos genéticos são uma técnica de aproximação estocástica. Por isso, partes do mapa podem sofrer pequenas alterações mesmo quando o conjunto de trajetórias coletadas não muda. Isso é inerente à forma de solução escolhida, porém, observou-se que os efeitos se tornam imperceptíveis em locais com grande quantidade de trajetórias coletadas. Por último, outra limitação da versão atual do método é que ele não identifica corretamente vias que têm sentido de movimento diferente de acordo com a hora do dia. Nestes casos, ele irá considerar que é uma via de mão dupla.

5.1 Trabalhos futuros

Em relação aos trabalhos futuros, sugere-se o estudo de uma maneira de melhorar a confiabilidade dos pontos coletados. Conforme se observou, uma quantidade considerável de pontos coletados informava ter boa acurácia, mas as coordenadas não correspondiam à realidade. Isto é causado por receptores GPS de baixa qualidade ou sob forte interferência de sinal. Acredita-se que técnicas como o Filtro de Kalman (GREWAL; ANDREWS, 2001) possam ajudar a identificar estes pontos e melhorar o desempenho da etapa de filtragem de dados.

Outro trabalho futuro é tornar a configuração dos valores dos parâmetros dinâmica, específicas para cada região. Dessa forma é possível, por exemplo, que locais de bastante movimento sejam mapeados com base apenas em dados de alta qualidade enquanto que locais com baixo movimento podem usar dados de acurácia pior ou mais antigos. Para fazer isto, considera-se necessário estudar maneiras de identificar as diferentes áreas da cidade e automatizar a definição de parâmetros com base apenas nos dados coletados.

Também se caracteriza como trabalho futuro minerar as trajetórias coletadas para descobrir informações a respeito das vias mapeadas. Por exemplo, a velocidade máxima permitida em

cada via, a presença de semáforos e buracos. Informações semânticas também poderiam ser extraídas e usadas para alimentar a criação do mapa. Por exemplo, no teste 4.2.3, onde foi avaliada a mudança no trânsito de uma via, o método poderia ter deduzido que nenhuma trajetória foi coletada no sentido de movimento antigo porque ele deixou de existir. Assim, o mapa poderia ser atualizado ainda mais rapidamente.

REFERÊNCIAS

ALVARES, L. O. et al. An algorithm to identify avoidance behavior in moving object trajectories. **Journal of the Brazilian Computer Society**, São Paulo, v. 17, n. 3, p. 193–203. 2011.

ANOTÍCIA. **Trinário das ruas Max Colin, Timbó e 15 de Novembro começa a valer até fim de novembro**. Joinville, 2013. Disponível em:
<http://anoticia.clicrbs.com.br/sc/noticia/2013/10/trinario-das-ruas-max-colin-timbo-e-15-de-novembro-comeca-a-valer-ate-fim-de-novembro-4317716.html> Acesso em: 16 jan 2014.

BÄCK, T.; HOFFMEISTER, F.; SCHWEFEL, H.-P. A Survey of Evolution Strategies. In: INTERNATIONAL CONFERENCE ON GENETIC ALGORITHMS (ICGA), 4., 1991, San Diego - EUA. Morgan Kaufmann, 1991.

BÄCK, T. *Evolutionary Algorithms in Theory and Practice*. 1. ed. New York - EUA: Oxford University Press. 314 p. 1996.

BECKET, B. S. *Biology: A Modern Introduction*. 1. ed. New York - EUA: Oxford University Press. 307 p. 1986.

BITTENCOURT, G. *Inteligência Artificial: Ferramentas e Teorias*. 3. ed. Florianópolis: Editora da UFSC. 371 p. 2006.

BOURNE, M. **How to find the equation of a quadratic function from its graph**. [S.I.], 2011. Disponível em:
<http://www.intmath.com/blog/how-to-find-the-equation-of-a-quadratic-function-from-its-graph> Acesso em: 24 mar 2014.

BRAZ, F. J.; BOGORNY, V. *Introdução a trajetórias de objetos móveis*. 1. ed. Joinville: Editora Univille. 116 p. 2012.

BRÜNTRUP, R. et al. Incremental map generation with GPS traces. In: IEEE CONFERENCE ON INTELLIGENT TRANSPORTATION SYSTEMS, 8., 2005, Vienna - Austria. **Proceedings...** IEEE, 2005.

CAO, L.; KRUMM, J. From GPS traces to a routable road map. In: ACM SIGSPATIAL INTERNATIONAL CONFERENCE ON ADVANCES IN GEOGRAPHIC INFORMATION SYSTEMS (GIS), 17., 2009, Seattle - EUA. **Proceedings...** ACM Press, 2009.

CHEN, M.; XU, M.; FRANTI, P. Compression of GPS Trajectories. In: DATA COMPRESSION CONFERENCE (DCC) 2012, Snowbird - EUA. **Proceedings...** IEEE, 2012.

CORMEN, T. H. et al. *Algoritmos - Teoria e Prática*. 2. ed. São Paulo: Elsevier Brasil. 916 p. 2002.

COSTA, G. H. R.; BALDO, F. A method to automatically identify road centerlines from georeferenced smartphone data. In: BRAZILIAN SYMPOSIUM ON GEOINFORMATICS (GeoInfo), 14., 2013, Campos do Jordão. **Proceedings...** INPE, 2013.

DEVILLERS, R.; JEANSOULIN, R. *Fundamentals of Spatial Data Quality*. 1. ed. Londres - Reino Unido: Wiley-ISTE. 312 p. 2006.

DOD - Department of Defence. **Global Positioning System Standard Positioning Service Performance Standard**. EUA, 2008. 160 p. Disponível em: <<http://www.gps.gov/technical/ps/#spsp>> Acesso em: 18 jan 2014.

EDEN, A. H. Three Paradigms of Computer Science. **Minds and Machines**, Países Baixos, v. 17, n. 2, p. 135–167. 2007.

EIBEN, A. E.; SMITH, J. E. *Introduction to Evolutionary Computing*. 2. ed. Berlin - Alemanha: Springer. 299 p. 2007.

FRENTZOS, E.; THEODORIDIS, Y. On the Effect of Trajectory Compression in Spatiotemporal Querying. **Lecture Notes in Computer Science: Advances in Databases and Information Systems**, Varna - Bulgaria, v. 4690, p. 217–233. 2007.

GABRIEL, P. H. R.; DELBEM, A. C. B. **Fundamentos de Algoritmos Evolutivos**. São Carlos, 2008. ICMC-USP. 35 p. Notas Didáticas.

GFK. **GfK TEMAX BRASIL T2 2013: Crescimento no mercado com forte influência de materiais de escritório e periféricos.** São Paulo, 2013. Disponível em: <<http://www.gfk.com/br/news-and-events/press-room/press-releases/Paginas/TEMAX-BRASIL-T2-2013.aspx>> Acesso em: 18 nov 2013.

GIL, A. C. *Como elaborar projetos de pesquisa*. 4. ed. São Paulo: Atlas. 176 p. 2002.

GILLIES, S. **Shapely**. [S.I.], 2013. Disponível em: <<https://github.com/Toblerity/Shapely>> Acesso em: 13 dez 2013.

GOOGLE. **Google Maps**. EUA, 2014. Disponível em: <<http://maps.google.com>> Acesso em: 17 jan 2014.

GREWAL, M. S.; ANDREWS, A. P. *Kalman Filtering : Theory and Practice Using MATLAB*. 2nd. ed. John Wiley & Sons, Inc. 410 p. 2001.

HAKLAY, M. (MUKI); WEBER, P. OpenStreetMap: User-Generated Street Maps. **IEEE Pervasive Computing**, v. 7, n. 4, p. 12–18. 2008.

HAUPT, R. L.; HAUPT, S. E. *Practical Genetic Algorithms*. 2. ed. New Jersey - EUA: Wiley-Interscience. 272 p. 2004.

HERSHBERGER, J.; SNOEYINK, J. Speeding Up the Douglas-Peucker Line-Simplification Algorithm. In: INTERNATIONAL SYMPOSIUM ON SPATIAL DATA HANDLING, 5., 1992, Chaleston - EUA. **Proceedings...** University of British Columbia, 1992.

HOFFMEISTER, F.; BÄCK, T. Genetic Algorithms and Evolution Strategies: Similarities and Differences. In: PARALLEL PROBLEM SOLVING FROM NATURE (PPSN), 1., 1991, Dortmund - Alemanha. **Proceedings...** Springer, 1991.

IGC - Instituto Geográfico e Cartográfico. **Glossário**. São Paulo, 2013. Disponível em: <<http://www.igc.sp.gov.br/educacional/glossario.html>> Acesso em: 17 jan 2014.

JANG, S.; KIM, T.; LEE, E. Map Generation System with Lightweight GPS Trace Data. In: INTERNATIONAL CONFERENCE ON ADVANCED COMMUNICATION TECHNOLOGY (ICACT), 12., 2010, Gangwon - Coréia do Sul. **Proceedings...** IEEE, 2010.

LI, M. et al. Trip analyzer through smartphone apps. In: PROCEEDINGS OF THE 19TH ACM SIGSPATIAL INTERNATIONAL CONFERENCE ON ADVANCES IN GEOGRAPHIC INFORMATION SYSTEMS - GIS '11 2011, New York - EUA. ACM Press, 2011.

MARCONI, M. DE A.; LAKATOS, E. M. *Fundamentos de metodologia científica*. 5. ed. São Paulo: Atlas. 311 p. 2003.

MERATNIA, N.; DE BY, R. A. Spatiotemporal Compression Techniques. **Lecture Notes in Computer Science: Advances in Database Technology - EDBT 2004**, Heraklion - Grécia, v. 2992, p. 765–782. 2004.

MICHALEWICZ, Z. *Genetic Algorithms + Data Structures = Evolution Programs*. 3. ed. Berlin - Alemanha: Springer. 387 p. 1996.

MICHALEWICZ, Z.; JANIKOW, C. Handling Constraints in Genetic Algorithms. In: INTERNATIONAL CONFERENCE ON GENETIC ALGORITHMS (ICGA), 4., 1991, San Diego - EUA. Morgan Kaufmann, 1991.

MICROSOFT. **Bing Maps**. EUA, 2014. Disponível em: <<http://www.bing.com/maps/>> Acesso em: 17 jan 2014.

MITCHELL, M. *An Introduction to Genetic Algorithms*. 5. ed. Cambridge - Inglaterra: MIT Press. 1999.

MOHAN, P.; PADMANABHAN, V. N.; RAMJEE, R. Nericell: Rich Monitoring of Road and Traffic Conditions using Mobile Smartphones. In: PROCEEDINGS OF THE 6TH ACM CONFERENCE ON EMBEDDED NETWORK SENSOR SYSTEMS - SENSYS '08 2008, New York - EUA. ACM Press, 2008.

MUCKELL, J. et al. Algorithms for Compressing GPS Trajectory Data: An Empirical Evaluation. In: ACM SIGSPATIAL INTERNATIONAL CONFERENCE ON ADVANCES IN GEOGRAPHIC INFORMATION SYSTEMS (GIS), 18., 2010, San Jose - EUA. **Proceedings...** ACM Press, 2010.

MUCKELL, J. et al. SQUISH: An Online Approach for GPS Trajectory Compression Categories and Subject Descriptors. In: INTERNATIONAL CONFERENCE ON COMPUTING FOR GEOSPATIAL RESEARCH & APPLICATIONS (COM.Geo), 2., 2011, Washington - EUA. **Proceedings...** ACM Press, 2011.

NETWORKX. **NetworkX**. [S.I.], 2013. Disponível em:
<http://networkx.github.io/> Acesso em: 17 jan 2014.

NIU, Z.; LI, S.; POUSAIED, N. Road Extraction using Smart Phones GPS. In: INTERNATIONAL CONFERENCE ON COMPUTING FOR GEOSPATIAL RESEARCH & APPLICATIONS (COM.Geo), 2., 2011, Washington - EUA. **Proceedings...** ACM Press, 2011.

NOAA - National Oceanic and Atmospheric Administration. **What is GPS?** EUA, 2013. Disponível em:
<http://www.gps.gov/systems/gps/> Acesso em: 18 nov 2013.

NUMPY. **NumPy**. [S.I.], 2013. Disponível em:
<http://www.numpy.org/> Acesso em: 23 dez 2013.

OLSON, J. E. *Data Quality: The Accuracy Dimension*. 1. ed. San Francisco - EUA: Morgan Kaufmann. 300 p. 2003.

OSM - OpenStreetMap. **OpenStreetMap: O Wiki de Mapas Livres**. [S.I.], 2013a. Disponível em: <http://www.openstreetmap.org/>
Acesso em: 3 nov 2013.

OSM - OpenStreetMap. **Trilhas Públicas de GPS**. [S.I.], 2013b.
Disponível em: <http://www.openstreetmap.org/traces> Acesso em:
11 mar 2013.

POLI, R.; LANGDON, W. B.; MCPHEE, N. F. *A Field Guide to Genetic Programming*. 250 p. 2008.

POTAMIAS, M.; PATROUMPAS, K.; SELLIS, T. Sampling Trajectory Streams with Spatiotemporal Criteria. In: INTERNATIONAL CONFERENCE ON SCIENTIFIC AND STATISTICAL DATABASE MANAGEMENT (SSDBM), 18., 2006, Vienna - Austria. **Proceedings...** IEEE, 2006.

PU, I. M. Lossless and Lossy Compression. **Fundamental Data Compression.** 1. ed. Oxford - Inglaterra: Butterworth-Heinemann. 2006. p. 5–7.

REDDY, S. et al. Using mobile phones to determine transportation modes. **ACM Transactions on Sensor Networks**, v. 6, n. 2, p. 1–27. 2010.

ROCHA, J. A. M. R. et al. DB-SMoT: A direction-based spatio-temporal clustering method. In: IEEE INTERNATIONAL CONFERENCE INTELLIGENT SYSTEMS (IS), 5., 2010, Londres - Reino Unido. **Proceedings...** IEEE, 2010.

SCHOLZ, G. **Método para Suavização de Curvas com base em Dados Obtidos com GPS de Smartphones.** Joinville, 2013. UDESC - DCC. 68 p. Trabalho de Conclusão de Curso.

TRAJCEVSKI, G. et al. On-line data reduction and the quality of history in moving objects databases. In: ACM INTERNATIONAL WORKSHOP ON DATA ENGINEERING FOR WIRELESS AND MOBILE ACCESS (MobiDE), 5., 2006, Chicago - EUA. **Proceedings...** ACM Press, 2006.

TRIPP, D. Pesquisa-ação: uma introdução metodológica. **Educação e Pesquisa**, São Paulo, v. 31, n. 3, p. 443–466. 2005.

WANG, S.; CHEN, C.; MA, J. Accelerometer based transportation mode recognition on mobile phones. In: 2010 ASIA-PACIFIC CONFERENCE ON WEARABLE COMPUTING SYSTEMS 2010, IEEE, 2010.

WAZLAWICK, R. S. *Metodologia de Pesquisa para Ciência da Computação.* 1. ed. Rio de Janeiro: Elsevier Brasil. 184 p. 2009.

WAZLAWICK, R. S. Uma Reflexão sobre a Pesquisa em Ciência da Computação à Luz da Classificação das Ciências e do Método Científico. *Revista de Sistemas de Informação da FSMA*, Macaé, p. 3–10. 2010.

WEKA. **Weka 3: Data Mining Software in Java**. Nova Zelândia, 2013. Disponível em: <<http://www.cs.waikato.ac.nz/ml/weka/>> Acesso em: 16 dez 2013.

XU, D. et al. Transportation Modes Identification from Mobile Phone. In: PROCEEDINGS OF THE 7TH INTERNATIONAL CONFERENCE ON ADVANCED DATA MINING AND APPLICATIONS - VOLUME PART II 2011, 2011.

ZHANG, L.; THIEMANN, F.; SESTER, M. Integration of GPS traces with road map. In: INTERNATIONAL WORKSHOP ON COMPUTATIONAL TRANSPORTATION SCIENCE (IWCTS), 2., 2010, San Jose - EUA. **Proceedings...** ACM Press, 2010.

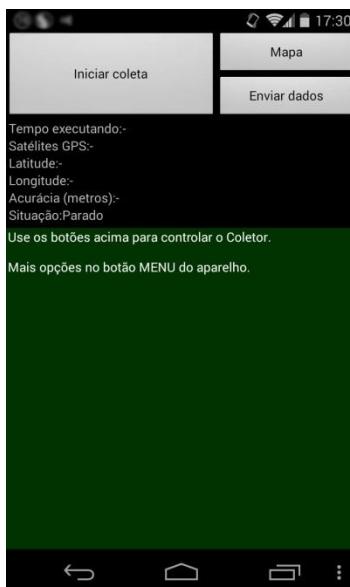
ZHENG, Y. et al. Learning transportation mode from raw gps data for geographic applications on the web. **Proceeding of the 17th international conference on World Wide Web - WWW '08**, New York - EUA, p. 247. 2008.

APÊNDICE A – Informações sobre o aplicativo para smartphones: Coletor

O aplicativo desenvolvido para coletar as trajetórias e dados de acelerômetro foi nomeado Coletor. Ele foi implementado para *smartphones* com sistema operacional Android, versão 2.3 ou superior. A escolha deste sistema operacional se deveu, principalmente, ao acesso gratuito às ferramentas de desenvolvimento, à sua enorme popularidade mundialmente e à maior disponibilidade de dispositivos para realização dos testes.

O Coletor pode ser copiado para o aparelho e instalado pelo endereço <<http://bdes.dcc.joinville.udesc.br:100/coletor>>. Uma vez instalado, cabe ao usuário decidir quando deseja coletar dados. A Figura 59 mostra a tela inicial do aplicativo.

Figura 59 – Coletor: Tela inicial

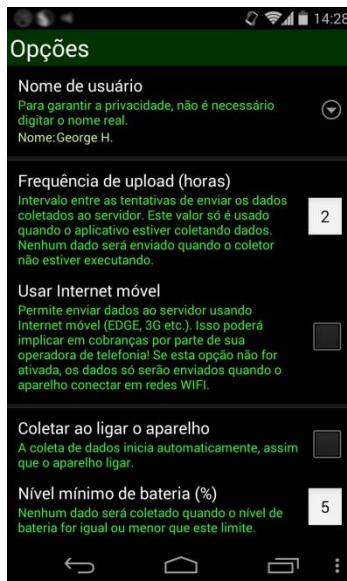


Fonte: produção do próprio autor

A parte superior contém os principais botões de controle do aplicativo – “Iniciar coleta”, “Mapa” e “Enviar Dados” – e todo o restante da área abaixo é usado para indicar o estado de

operação do Coletor. Além disso, ao pressionar o botão <Menu> do aparelho (ou, em alguns aparelhos, o ícone de três pontos no canto inferior direito) o usuário tem acesso à informações sobre o aplicativo e ao menu de opções. Este menu pode ser visto na Figura 60. Nele, o usuário pode configurar algumas características do aplicativo e da coleta de trajetórias.

Figura 60 – Coletor: Menu de Opções



Fonte: produção do próprio autor

A primeira opção permite ao usuário escolher o nome de identificação que será armazenado nos arquivos de coleta. Para aumentar a privacidade do usuário, este nome não precisa ser o nome real. Por padrão, o próprio aplicativo gera um nome contendo letras e números. O intuito de permitir que o usuário defina este nome é para que, caso ele mude de *smartphone*, seja possível perceber que é a mesma pessoa. Esta funcionalidade se mostrou interessante durante o desenvolvimento do aplicativo para facilitar a identificação e correção de erros nos arquivos de trajetória gerados.

Outra opção disponível neste menu permite controlar o intervalo entre cada tentativa automática de envio dos arquivos

coletados para o servidor. Também é possível definir se o aplicativo pode usar Internet móvel e qual o nível mínimo de bateria restante antes que o Coletor pare de gravar dados. Para facilitar o uso do aplicativo, também é possível configurá-lo para começar a coletar dados assim que o dispositivo ligar.

De volta à tela inicial, a coleta será iniciada no momento que o usuário tocar no botão “Iniciar coleta”. Neste momento, o espaço de fundo preto irá começar a exibir informações atualizadas sobre o funcionamento do GPS, enquanto o espaço de fundo verde mostra de forma resumida informações históricas. A Figura 61 mostra a tela inicial do aplicativo durante uma coleta de trajetória.

Figura 61 – Coletor: Coleta de trajetória



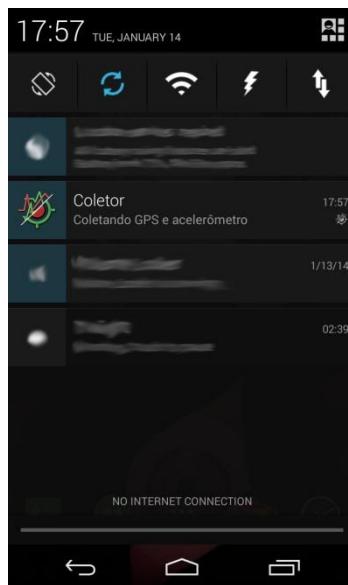
Fonte: produção do próprio autor

O Coletor irá começar a escrever os dados obtidos do GPS e acelerômetro em um arquivo, que será finalizado quando (1) quando o arquivo atingir o tamanho máximo de 2MB ou (2) sinal de GPS for perdido. O tamanho limite serve apenas para reduzir a taxa de erros ao tentar enviar os arquivos ao servidor. Quando o limite é atingido, um novo arquivo (ou seja, uma nova

trajetória) é iniciado. Se o sinal de GPS for perdido, o Coletor irá aguardar até que a localização seja obtida novamente e, então, iniciar outro arquivo. Se demorar muito tempo para obter a localização, o Coletor irá reduzir, aos poucos, a frequência com que requisita atualizações do GPS. O intuito desta ação é reduzir o consumo de bateria quando o usuário está dentro de casa, por exemplo, mas esqueceu de parar o Coletor. Ao tocar em “Parar coleta”, o arquivo atual é finalizado e a coleta termina.

Também é interessante destacar que o Coletor continua funcionando mesmo que o usuário tire-o de primeiro plano (ou seja, o aplicativo fica executando em segundo plano). Para avisar o usuário que o aplicativo ainda está ativo é exibido um aviso na área de notificações do sistema operacional, como pode ser visto na Figura 62.

Figura 62 – Coletor: Aviso na área de notificação



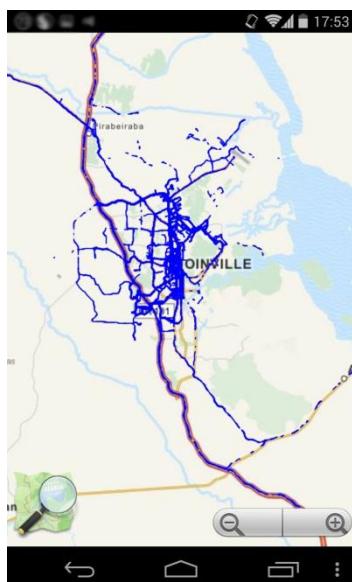
Fonte: produção do próprio autor

O botão “Enviar dados”, como o nome diz, serve para enviar os arquivos de trajetória ao servidor. Enquanto o aplicativo está coletando dados ele tentará fazer o envio automaticamente, caso as opções de usuário permitam. Caso contrário, o usuário

deve pressionar o botão. O aplicativo irá lembrar o usuário de enviar os arquivos mostrando-lhe notificações. Além dos arquivos de trajetória o aplicativo também envia ao servidor relatórios de erro, para permitir a identificação e correção de eventuais problemas. Após enviados, tanto os arquivos de trajetória como os relatórios de erro são apagados do aparelho.

Por último, o botão “Mapa” exibe ao usuário todas as trajetórias coletadas até o momento, sobrepostas ao mapa da região de Joinville (fornecido pelo OpenStreetMap). A Figura 63 mostra esta tela do aplicativo.

Figura 63 – Coletor: Mapa de trajetórias coletadas



Fonte: produção do próprio autor

Pressionando o botão <Menu> do aparelho o usuário pode filtrar o mapa para exibir apenas as trajetórias que ele enviou ao servidor – com base no nome de usuário que ele definiu anteriormente. Além disso, o usuário também pode pedir ao servidor que lhe informe quantos quilômetros ele coletou, e quanto tempo ele gastou coletando.

APÊNDICE B – Parâmetros utilizados em todas as etapas do método desenvolvido

Da maneira como o método foi implementado, 24 parâmetros são utilizados ao longo das etapas de pré-processamento das trajetórias (seção 3.1.2), identificação dos centros das vias (seção 3.1.3) e criação do grafo direcionado (seção 3.1.4). As seções a seguir apresentam estes parâmetros, assim como a função de cada um deles e o valor que receberam. Alguns parâmetros são utilizados em mais de uma etapa. Para facilitar o entendimento, estes parâmetros são repetidos em cada etapa onde foram utilizados.

Parâmetros usados no pré-processamento:

$$tam_{amostra} = 2$$

- Usado na identificação do meio de transporte.
- O valor 2 significa que a trajetória será dividida em intervalos de dois segundos – que contém aproximadamente 40 amostras do acelerômetro e duas da velocidade (via GPS) – e o método identificará o meio de transporte usado em cada um dos intervalos.

$$quantidade_amostras_por_intervalo = 5 * tam_{amostra}$$

- Usado na identificação do meio de transporte.
- Apesar do aplicativo para *smartphones* tentar coletar dados de acelerômetro em uma frequência de 20 vezes por segundo, devido a limitações dos aparelhos isso nem sempre é possível. Este parâmetro define qual a quantidade mínima de amostras necessárias para tentar identificar o meio de transporte. Intervalos que não atingem essa quantidade mínima são desconsiderados.

$$d_{proximo} = 3.0 \text{ metros}$$

- Usado para eliminar pontos muito próximos uns dos outros em uma mesma trajetória.
- Ao eliminar pontos usando o valor escolhido, o mapa gerado não sofreu nenhuma alteração perceptível. Por outro lado, devido à quantidade reduzida de pontos, o tempo de execução foi ao menos 10% menor.

$a_{lim} = 15.0$ metros

- Usado para filtrar os pontos coletados com base nos dados de GPS.
- Maior valor de acurácia permitido.

$\theta_{lim} = 25.0$ graus

- Usado para filtrar os pontos coletados com base nos dados de GPS.
- Maior variação de angulo de movimento permitido para os pontos coletados.

$tam_{grupos} = 20$

- Usado para filtrar os pontos coletados com base nos dados de GPS.
- Os pontos de cada trajetória são divididos em grupos com o tamanho definido, e cada grupo é submetido à filtragem por acurácia média e variação de angulação.

$a_{lim_{media}} = 15.0$ metros

- Usado para filtrar os pontos coletados com base nos dados de GPS.
- Acurácia média limite do grupo de pontos sendo analisado.

$\varepsilon = 0.5$ metros

- Erro máximo permitido durante a compressão da trajetória com o algoritmo Before Opening Window.
- O valor é pequeno para não descharacterizar visualmente a trajetória, mas se mostrou suficiente para descartar pontos pouco significativos.

Parâmetros usados na identificação dos centros das vias:

$n_{min} = 6$

- Quantidade mínima de pontos que devem ser selecionados para que seja calculado um centro da via. O objetivo é desconsiderar pontos que dizem ter boa acurácia, mas que estão posicionados incorretamente. Por outro lado, esta limitação faz com que vias com

poucos dados coletados sejam eliminadas do mapa. A solução de Cao e Krumm (2009) também tem um parâmetro com função semelhante.

$$t_{max} = 180 \text{ dias}$$

- Usado para descartar pontos que, em relação à data atual, são mais antigos que o valor definido.
- Usado também na função de *fitness*, na equação (2) da seção 3.1.3.3.

$$\theta_{max} = 100.0 \text{ graus}$$

- Usado para criar o conjunto de pontos selecionados (conjunto S).
- Maior variação de angulo de movimento permitido para conectar os centros das vias para que uma via seja considerada válida.

$$\theta_{lim} = 25.0 \text{ graus}$$

- Usado para criar o conjunto de pontos selecionados (conjunto S).
- Maior variação de angulo de movimento permitido para os pontos coletados.

$$a_{lim} = 15.0 \text{ metros}$$

- Usado para criar o conjunto de pontos selecionados (conjunto S).
- Usado na função de *fitness* para definir os valores de a_{meioM} (seção 3.1.3.3, equação (3)) e d_{meioM} (seção 3.1.3.3, equação (4)).
- Maior valor de acurácia permitido.

$$a_{meioM} = 0.9/(a_{lim}/4)$$

- Usado na função de *fitness*, para definir o valor de a_{meioV} (seção 3.1.3.3, equação (3)).

$$d_{meioM} = 0.9/(a_{lim}/4)$$

- Usado na função de *fitness*, para definir os valores de d_{meioV} (seção 3.1.3.3, equação (4)).

$MT = 0.3$

- Usado na função de *fitness*.
- O resultado de $MT + MA + MD$ deve ser 1.0.

 $MA = 0.3$

- Usado na função de *fitness*.
- O resultado de $MT + MA + MD$ deve ser 1.0.

 $MD = 0.4$

- Usado na função de *fitness*.
- O resultado de $MT + MA + MD$ deve ser 1.0.

 $g = 20$

- Número de gerações
- Usado no algoritmo genético.

 $z = 50$

- Tamanho da população (quantidade de soluções candidatas).
- Usado no algoritmo genético.

 $pop_{elite} = 2$

- Usado no algoritmo genético.

 $mut_{distância} = 5$ metros

- Usado no algoritmo genético.

 $mut_{chance} = 5\%$

- Usado no algoritmo genético.

 $rec_{chance} = 85\%$

- Usado no algoritmo genético.

Parâmetros usados na criação do grafo direcionado:

 $\theta_{lim} = 25.0$ graus

- Dados dois centros de uma via, ajuda a definir se estes pontos podem se conectar ou não. Ou seja, se realmente fazem parte da mesma via.
- Maior variação de angulo de movimento permitido para os pontos coletados.

$a_{lim} = 15.0$ metros

- Dados dois centros de uma via, ajuda a definir se estes pontos podem se conectar ou não. Ou seja, se realmente fazem parte da mesma via.
- Maior valor de acurácia permitido.

$dist_{pontos} = 120$ metros

- Dados dois centros de uma via, ajuda a definir se estes pontos podem se conectar ou não. Ou seja, se realmente fazem parte da mesma via.

$\theta_{max} = 100.0$ graus

- Dados dois centros de uma via, ajuda a definir se estes pontos podem se conectar ou não. Ou seja, se realmente fazem parte da mesma via.
- Maior variação de angulo de movimento permitido para conectar os centros das vias para que uma via seja considerada válida.

nós_a_percorrer = 2

- Usado para percorrer o grafo e encontrar pontos conectados erroneamente.
- Dado um nó do grafo, define quantos nós sucessores devem ser verificados.