UNIVERSIDADE FEDERAL FLUMINENSE

DANIEL PADILHA TORQUATO DANTAS

ESTIMANDO INFORMAÇÕES DE LINHAS DE ÔNIBUS A PARTIR DE DADOS HISTÓRICOS DE GPS

NITERÓI

2017

UNIVERSIDADE FEDERAL FLUMINENSE

DANIEL PADILHA TORQUATO DANTAS

ESTIMANDO INFORMAÇÕES DE LINHAS DE ÔNIBUS A PARTIR DE DADOS HISTÓRICOS DE GPS

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Computação. Área de concentração: ENGENHARIA DE SISTEMAS E INFORMAÇÃO

Orientador:
ANTONIO AUGUSTO DE ARAGÃO ROCHA

Co-orientador:
MARCOS DE OLIVEIRA LAGE FERREIRA

NITERÓI

2017

DANIEL PADILHA TORQUATO DANTAS

ESTIMANDO INFORMAÇÕES DE LINHAS DE ÔNIBUS A PARTIR DE DADOS HISTÓRICOS DE GPS

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Computação. Área de concentração: ENGENHARIA DE SISTEMAS E INFORMAÇÃO

Aprovada em Setembro de 2017.

BANCA EXAMINADORA

Prof. ANTONIO AUGUSTO DE ARAGÃO ROCHA - Orientador, UFF

Prof. MARCOS DE OLIVEIRA LAGE FERREIRA, Co-orientador, UFF

Prof. ALEXANDRE PLASTINO DE CARVALHO, UFF

Prof. PEDRO BRACONNOT VELLOSO, UFRJ

Niterói

2017

Agradecimentos

Aos meus pais Carlos e Rita, por todo carinho e atenção ao longo de toda a minha vida.

Ao meu irmão Caio, por todo o apoio e companheirismo.

A minha noiva Carolina, por toda paciência e pela ajuda na elaboração deste projeto.

Às minhas avós Nilza e Nelcia, pelos seus ensinamentos.

Aos meus tios, tias e primos, por todo o apoio na realização deste projeto.

Ao meu orientador Antonio Augusto e ao meu co-orientador Marcos Lage, por toda orientação acadêmica.

Resumo

A eficiência da mobilidade urbana é colocada em questão quase que diariamente por grande parte da população urbana mundial. Sabendo disso, os administradores das grandes cidades gastam boa parte do tempo monitorando e planejando melhorias sobre os sistemas de transportes. Para facilitar a execução destas ações, temos a utilização dos Sistemas Inteligentes de Transportes (SIT) que disponibilizam aos administradores diversos tipos de ferramentas computacionais.

O sucesso obtido pela utilização dos SITs fez com que as entidades detentoras de informações dos transportes públicos das cidades se sentissem motivadas a divulgá-las ao público em geral, para que novos estudos nesta área possam ser desenvolvidos.

Aproveitando-se desta tendência, este trabalho utiliza-se dos dados dos ônibus públicos da cidade do Rio de Janeiro, no Brasil, disponibilizados de forma aberta pela própria prefeitura, para extrair de forma automática algumas de suas principais informações operacionais. Mais especificamente, são inferidas as regiões de garagens, pontos inicias e finais e a rota, rua a rua, das linhas. Estas informações são de extrema importância para os administradores públicos e para os usuários, pois os administradores de posse destas podem entender e planejar melhor o sistema de transporte. Por outro lado, também são de grande valia aos usuários pois, quando são informados de forma correta e atualizada podem elevar a confiança, aceitação e satisfação no uso do transporte.

Palavras-chave: Dados Urbanos, Map-Matching, Ônibus, Rotas, ITS

Abstract

The efficiency of urban mobility is a huge concern of urban population around the world. Because of this reason, city planners spend much of their time monitoring transportations systems and designing solutions in order to improve the system's quality. Among these solutions, the most successful ones are computational tools called Intelligent Transportation Systems (ITS).

The success of ITS has encouraged public agencies, owners of the public transportation system information, to share their datasets with the population aiming to stimulate the development of new research and solutions that could help to improve urban mobility.

Taking advantage of this trend, this work uses the Rio de Janeiro buses GPS logs dataset to extract some of the main operational information about the city bus system. More specifically, garage locations, start and end points of a route, and the route (the complete sequence of streets) of a bus line are inferred.This information is extremely important for both city planners and population since administrators can benefit from it to better plan the transportation system and the population can become more informed about the system, what improves its reliability and overall usage satisfaction.

Keywords: Urban Analytics, Map-matching, Bus, Route, ITS

Lista de Figuras

Lista de Tabelas

Sumário

Capítulo 1

Introdução

A população urbana mundial vem aumentado rapidamente [26] e este crescimento gera diversos desafios para as autoridades e administradores das grandes cidades. Uma das questões mais delicadas é a qualidade do sistema de transporte e tempo médio de locomoção diário gasto pela população.

Um sistema de transporte eficiente aumenta bastante a qualidade de vida das pessoas, já que, o tempo gasto no deslocamento diário de casa para o trabalho poderia ser revertido em lazer, esportes, saúde ou até mesmo em mais horas de sono.

Com o objetivo de aumentar a compreensão que se tem dos meios de transporte de grandes centros urbanos, diversos *Sistemas Inteligentes de Transporte* (SIT) têm sido propostos [15, 25, 1, 24, 8]. Os SITs são soluções tecnológicas, incluindo sistemas computacionais, que têm como objetivo a melhoria da gerência, administração e/ou operacionalização dos sistemas de transporte das cidades.

O sucesso obtido através do uso dos SITs motivou novos estudos que deram origem a uma categoria específica de sistemas, denominada Sistemas Avançados de Transporte Público (APTS, do termo em inglês *Advanced Public Transportations Systems*). Esta categoria é voltada exclusivamente para a análise dos sistemas de transportes públicos. APTS dão suporte ao administrador, não somente na tomada de decisões operacionais cotidianas, mas também na elaboração de medidas públicas que tenham como objetivo principal o aumento da eficiência e a melhoria do serviço oferecido aos usuários no transporte público.

Uma outra tendência, cada vez mais comum por parte da administração pública, é a disponibilização de dados abertos para a sociedade através de portais Web. Cidades

como Nova Iorque[1], Londres[2], Paris[3], Chicago[4], Boston[5] e Rio de Janeiro[6] têm disponibilizado informações dos mais diversos tipos e para inúmeros diferentes propósitos. O Portal de Dados Abertos da Prefeitura do Rio de Janeiro (Data.Rio), por exemplo, vem disponibilizando, desde 2014, informações sobre o seu sistema de transporte público.

Um dos conjuntos de dados disponibilizados no Data.Rio refere-se à frota de ônibus em circulação na cidade. Para cada veículo em operação são registradas, a cada minuto, a localização geográfica, identificador, linha em que opera e velocidade instantânea. Essas informações são geradas a partir de equipamentos de GPSs e transmitidos em tempo real através de celulares instalados em todos os veículos. A geração destas informações, que é de responsabilidade das próprias empresas de ônibus, tem grande potencial de utilidade para o desenvolvimento de novos SITs e APTSs.

Uma das utilidades para os dados disponíveis no Data.Rio é a possibilidade de, a partir deles, inferir outras informações operacionais do sistema de transporte público, não disponíveis no portal ou mesmo inexistentes, e que são relevantes tanto a sociedade quanto para a própria administração pública. A estimativa dos trajetos percorridos por cada veículo, pontos inicial e final de cada linha, localização das garagens dos ônibus, dentre outras, são algumas das informações relevantes e que não são conhecidas atualmente.

O presente trabalho busca resolver exatamente o problema de complementação das informações operacionais de sistemas de transporte público. Para isso, é proposta uma metodologia capaz de estimar, de forma automática, algumas importantes informações operacionais sobre as linhas de ônibus de uma cidade, utilizando como base os históricos de posicionamento de GPS dos veículos e os dados cartográficos da cidade. Essas informações operacionais podem servir de insumo para sistemas Web de tempo real, a serviço da sociedade e da administração pública. Os algoritmos desenvolvidos, quando aplicados aos dados da cidade do Rio de Janeiro (obtidos do portal Data.Rio), são capazes de identificar automaticamente as rotas, rua a rua, os pontos inicial e final, e as garagens das linhas de ônibus da cidade.

[1] http://opendata.cityofnewyork.us/
[2] http://data.london.gov.uk/
[3] http://opendata.paris.fr/
[4] http://opendata.cityofnewyork.us/
[5] http://data.cityofboston.gov/
[6] http://data.rio

1.1 Organização do Texto

Além deste capítulo introdutório inicial, o restante do trabalho se encontra dividido da seguinte forma: No Capítulo 2 são apresentados os trabalhos que possuem relações com os temas aqui abordados. O Capítulo 3 descreve em detalhes o problema que está sendo estudado e que deve ser solucionado através da aplicação de nossa metodologia. Já no Capítulo 4 é apresentada a bases de dados de operação dos ônibus utilizada como insumo das análises. O Capítulo 5 descreve, passo-a-passo, a metodologia proposta. No Capítulo 6 são apresentados e comentados os resultados obtidos. Por fim, o Capítulo 7 apresenta a conclusão sobre o tema aqui dissertado.

Capítulo 2

Trabalhos Relacionados

Por mais que o objetivo principal deste trabalho concentre-se na análise exclusiva de dados produzidos por ônibus públicos de uma cidade, outros estudos que analisam dados produzidos por demais modais apresentam situações e problemas similares aos encontrados nesta pesquisa. Esta situação ocorre pois, independentemente do meio de transporte estudado, todos os recursos (Ex.: Veículos particulares, táxis, ônibus e trens) deslocam-se pelas cidades gerando amostras espaço-temporais. Neste sentido, os trabalhos relacionados aqui citados abrangem diferentes tipos de modais. Além disso, as citações são divididas em categorias de acordo com o objetivo de cada trabalho.

2.1 Sistemas Inteligentes de Transporte

A fim de reafirmar a diversidade de modais estudados, temos os trabalhos que abrangem os SIT (Sistemas Inteligentes de Transporte). SIT são sistemas que integram ferramentas de diferentes áreas de atuação, tais como: comunicação de dados, sensoreamento e algoritmos computacionais avançados com o objetivo de prover e facilitar análises sobre um sistema de transporte qualquer [16, 17, 12].

Os SIT podem ser divididos em categorias que variam de acordo com os dados utilizados como insumo das análises [23]. Em [14] são descritas as seis categorias que apresentam uma maior relevância nesta área de pesquisa. Podemos destacar a identificação da categoria denominada Advanced Public Transportations Systems (APTS), onde são consumidos exclusivamente dados operacionais dos transportes públicos de uma cidade, tais como: ônibus, trens, barcas e metros. Em [17] são definidos os seguintes objetivos dos sistemas deste tipo: aumentar o controle sobre as viagens (garantir que os horários determinados de partidas e chegadas sejam atendidos), contribuir para um sistema tarifário integrado e

evoluir a divulgação de importantes informações relativas à operação aos usuários finais. Neste sentido, nosso trabalho pode ser considerado um APTS, pois tem como um de seus objetivos a identificação de importantes informações dos ônibus públicos, que podem ser disponibilizadas aos usuários do transporte.

2.2 Extração da Rota de um veículo

Os trabalhos apresentados a seguir possuem como um dos seus objetivos a extração da rota utilizada por um veículo durante um determinado período de tempo.

Em [20] os autores propõem um método capaz de extrair a rota e os pontos de paradas utilizados por uma linha de ônibus. Para isto, são utilizados dados de posicionamento dos ônibus em conjunto com pontos de paradas previamente conhecidos. A base de dados utilizada contempla exclusivamente informações dos ônibus que foram produzidas durante os deslocamentos entre os Pontos Inicias e Pontos Finais, o que não acontece com os dados utilizados pelo nosso trabalho, visto que os mesmos são produzidos e disponibilizados de forma ininterrupta durante toda a operação. Além disso, a frequência de disponibilização dos dados é de 20 segundos. Para a identificação das viagens dos ônibus é adotado um *threshold* temporal que visa representar o tempo pelo qual o ônibus fica parado em seu Ponto Inicial ou Ponto Final. Neste sentido, quando posições subsequentes de um ônibus tiverem uma diferença temporal superior ao valor definido, é considerado então o início de uma nova viagem e o termino da viagem anterior. Após a identificação das viagens, são extraidos os possíveis pontos de paradas dos veículos a partir de um algoritmo de clusterização denominado DBSCAN [13]. Sobre os clusters obtidos é aplicado um modelo de classificação de dados em conjunto com uma base de treinamento real para identificar os que apresentam maior probabilidade de serem de fato pontos reais de paradas. Em seguida, os pontos com maior probabilidade são conectados, de forma sequencial, para formarem o segmento que representa a rota da linha de ônibus. Por fim, para suavizar a geometria formada através desta junção é aplicada uma função de curva de Bézier.

Em [8] é apresentado um *framework* capaz de inferir rotas e pontos de parada, com os respectivos tempos de chegada e partida, de qualquer veículo que tenha acoplado a si um *smartphone* capaz de enviar informações de GPS ao sistema. Os dados utilizados como insumo do trabalho referem-se a operação de ônibus públicos da cidade de Chicago, nos Estados Unidos. A estratégia utilizada para obtenção das rotas consiste exclusivamente na análise dos dados históricos de posicionamento dos veículos. Neste sentido, não é

utilizada uma base de dados que represente o mapa da cidade, pois os autores acreditam que tais informações possuem baixa confiabilidade como, por exemplo, imprecisões nas representações das ruas e baixas taxas de atualizações. Basicamente, o primeiro passo da metodologia consiste em extrair a topologia (mapa das regiões utilizadas) a partir da aplicação do método estatístico denominado Estimativa de Densidade Kernel sobre os dados históricos de posicionamento. Tal método tem como objetivo a suavização de dados onde inferências sobre a população são feitas, sendo neste caso as coordenadas geográficas de cada veículo. Em seguida, a topologia é obtida através da aplicação da metodologia apresentada em [11] sobre os dados obtidos na Estimativa de Densidade Kernel. De posse desta topologia, é aplicado sobre ela um método de Map-Matching probabilístico para cada posição dos ônibus, a fim de identificar os segmentos mais utilizados pelos veículos. Por fim, estes segmentos serão selecionados e conectados fisicamente para formarem a rota da linha.

Os autores em [24] também utilizam os dados da cidade de Chicago para validação de sua metodologia, que é capaz de identificar a rota, pontos de parada e os tempos de chegada e partida dos ônibus nos pontos. Diferentemente do trabalho aqui apresentado, as viagens dos ônibus já são previamente conhecidas e somente são utilizados informações de posicionamento dos ônibus que fazem parte destas viagens. Além disso, as posições dos ônibus são informadas em média a cada 20 segundos. O processo de identificação da rota é dividido em seis etapas. Os dois primeiros passos consistem na identificação da viagem que possui o maior número de informações de posicionamento. Tais informações de GPS servem de base para criação de clusters, que abrangem as demais informações de viagem que estejam em até 40 metros de distância (Distância Euclidiana) de si. Em seguida, como terceiro passo, são removidos os clusters que possuem uma baixa taxa de diversidade de viagens. O quarto passo consiste em "conectar" os centroides dos clusters. Já o quinto passo ordena as conexões entre os clusters, através de uma ordenação espacial e temporal. Por fim, a execução do último passo é opcional e consiste na aplicação de métodos de map-matching da rota obtida sobre dados cartográficos para suavizar a geometria formada. Os autores relatam que a grande maioria das rotas testadas são obtidas com êxito, porém não relatam os casos em que os erros foram encontrados e nem mesmo suas razões.

2.3 Map-Matching

Como um dos objetivos deste trabalho consiste na identificação da rota, rua à rua, de um ônibus, partindo-se de amostras espaço-temporais, temos a necessidade de identificar a

correta **relação** entre o **posicionamento geográfico** dos ônibus e o **segmento** de rua, pertencente ao mapa, utilizado no momento da geração de tal informação. O problema de identificar o segmento de um mapa utilizado a partir dos registros de posição é estudado por diferentes áreas, e é conhecido como *Map-Matching*.

Algoritmos para resolução deste tipo de problema podem ser desenvolvidos para aplicações que consumam dados em tempo real ou dados históricos de posicionamento [22]. Como exemplo de aplicação que consome dados em tempo real, temos os sistemas de navegação de veículos, popularmente conhecidos como GPS veiculares, onde os registros de posição são analisados e o resultado do segmento de rua utilizado é obtido em tempo real. De outro lado, temos os SIT que consomem dados históricos na elaboração de suas análises.

Devido à elevada quantidade de trabalhos produzidos nesta área, é apresentado em [22] uma divisão em categorias que variam de acordo com a estratégia utilizada para resolução deste tipo de problema. A seguir, tais categorias são apresentadas.

Análise Geométrica A estratégia de resolução desta categoria baseia-se apenas na análise geométrica dos segmentos. Sendo assim, não são utilizados como parâmetros as conexões entre os segmentos. Algoritmos deste tipo são extremamente simples e com rápida execução. Porém, são mais sensíveis a variações das imprecisões dos dispositivos GPS e apresentam baixas taxas de acerto. Genericamente, o seu funcionamento ocorre da seguinte forma: o segmento do mapa que apresentar a menor distância física para a coordenada geográfica analisada será escolhido como o segmento utilizado no momento da geração da informação de posicionamento [7, 28].

Análise Topológica Algoritmos desta categoria analisam, em conjunto, informações geométricas dos segmentos e as relações entre estes no mapa. As relações entre segmentos podem, por exemplo, representar conexões entre segmentos e vizinhanças de regiões. Para esta estratégia, geralmente são definidos pesos para cada uma das análises (formação geométrica e relação), a fim de balancear os lados positivos e negativos de cada [29].

Probabilística Os algoritmos desta categoria têm como estratégia analisar uma região denominada região de confiança que é criada ao redor da informação de posicionamento estudada. Geralmente, este tipo de região possui a forma quadrada. Como tal região é maior do que uma informação de posicionamento, temos a possibilidade desta interceptar 0 ou mais segmentos do mapa. O fato de nenhum segmento

estar contido na região de confiança pode ocorrer devido a falhas ou imprecisões nos dispositivos GPS, em que são geradas informações distantes da posição real. Neste sentido, uma análise probabilística é aplicada sobre todos os segmentos interceptados. A partir desta análise é identificado o segmento que possui a maior probabilidade de ter sido utilizado pela informação analisada [10, 19].

Avançada Trabalhos que propõem estratégias ditas avançadas são capazes de combinar duas ou mais das estratégias citadas anteriormente. Além disso, também são utilizados conceitos matemáticos e computacionais mais refinados, tais como Kalmam Filter e Extended Kalman Filter [18].

A estratégia adotada pela metodologia proposta na identificação dos segmentos de rua utilizados pelos ônibus baseia-se na categoria Probabilística. Porém, diferentemente da grande maioria dos trabalhos desta área que visam identificar o segmento utilizado por um único veículo durante o seu deslocamento, este trabalho analisa um conjunto de informações produzidos por diversos veículos a fim de identificar o segmento mais provável de ter sido utilizado por todos. Este ponto específico da metodologia é apresentado em detalhes na Seção 5.

Capítulo 3

Descrição do problema

Este trabalho apresenta um estudo que visa estimar informações operacionais das linhas de ônibus de uma cidade, a partir de amostras espaço-temporais produzidas ininterruptamente por cada um dos veículos da frota e informações da malha viária da região em que estes operam. Neste caso, as amostras são ditas ininterruptas, pois são coletadas a partir do instante em que os veículos (ônibus) são ligados até o momento em que estes são desligados. O objetivo deste trabalho é propor algoritmos capazes de estimar as seguintes informações operacionais de cada linha de ônibus:

1. As coordenadas geográficas da garagem da empresa de ônibus responsável pela linha;
2. As coordenadas geográficas dos pontos inicial e final da rota percorrida pelos ônibus que operam na linha;
3. A rota, rua à rua, percorrida pelos ônibus que operam na linha.

A adoção de dados espaço-temporais gerados de forma ininterrupta, neste tipo de problema, introduz inúmeros desafios na elaboração dos algoritmos propostos por este trabalho. Tais desafios estão relacionados à ausência de importantes informações que indiquem o estado dos veículos. Podemos destacar as ausências de direção adotadas por um veículo, locais de início e término de uma operação e a sinalização de eventuais modificações sobre as rotas dos ônibus que, por exemplo, podem ser geradas a partir de acidentes ou modificações temporárias no trânsito. Sendo assim, os algoritmos propostos (detalhados na Seção 5), devem ser capazes de, mesmo sem ter o acesso ao estado de um ônibus, inferir, por exemplo, a direção adotada pelos ônibus de uma linha (identificar se o ônibus está deslocando-se do ponto inicial para o final, ou vice-versa) e serem robustos a fim de não sofrerem impactos de dados gerados por eventuais rotas auxiliares utilizadas.

Para contornar esse problema, assumiremos que os ônibus de uma cidade executam diariamente o mesmo protocolo de operação, que pode ser descrito através das seguintes atividades, em ordem cronológica:

1. Deslocamento com origem na garagem de sua linha e destino no ponto inicial de sua rota.
2. Conjunto de consecutivas viagens entre o ponto inicial e o ponto final de sua rota.
3. Deslocamento com origem no ponto final de sua rota e destino na garagem de sua linha

A identificação do protocolo de operação dos ônibus serviu como referência na criação dos algoritmos deste trabalho. Por exemplo, as Atividades 1 e 3 servem como guia na obtenção das regiões geográficas representadas pelas garagens das empresas de ônibus. Já a Atividade 2 é utilizada para inferência dos segmentos de rua utilizados no deslocamento dos ônibus durante a rota de sua linha. Sendo assim, a metodologia tem como objetivo indireto a identificação das atividades executadas por cada uma das linhas de ônibus para que o problema aqui apresentado possa ser solucionado.

A metodologia proposta por este trabalho é apresentada em detalhes e em ordem cronológica de execução na Seção 5.

Capítulo 4

Base de Dados de Operação dos Ônibus

A base de registros de posição dos ônibus utilizada neste trabalho é fornecida pela Prefeitura da cidade do Rio de Janeiro no Brasil através da plataforma *web* Data.Rio [2]. Além destes registros, também são divulgadas outras informações operacionais, tais como: rotas e pontos de paradas de algumas das linhas que operam na cidade. Porém, a maioria destas informações encontram-se desatualizadas, pois alguns dos ônibus modificaram a sua operação em virtude da infraestrutura de mobilidade urbana da cidade ter sido reestruturada para os Jogos Olímpicos de 2016.

Os registros de posição dos ônibus da cidade do Rio de Janeiro são gerados a partir de dispositivos de GPSs acoplados aos veículos, que capturam e transmitem o posicionamento de forma ininterrupta enquanto os veículos estejam ligados. A disponibilização destes dados, através da plataforma, se da pela divulgação de um arquivo no formato JSON [3], contendo o último registro de posição de cada ônibus da cidade. A informação divulgada é composta pelos seguintes atributos: (1) Identificador do ônibus; (2) Linha; (3) Latitude; (4) Longitude; (5) Velocidade instantânea e (6) *timestamp*. Sua frequência de **atualização é de 1 minuto**. Esta taxa pode ser considerada baixa se comparada às taxas de atualização de bases usadas em trabalhos com objetivos similares. Usualmente, a taxa de atualização média varia entre 15 e 25 segundos [8, 21, 24]. Taxas de atualização baixas tornam a estimativa das informações operacionais de linhas de ônibus mais difícil, em especial o cálculo das rotas rua a rua, já que, dependendo da velocidade dos veículos, registros de posição consecutivos podem ocorrer em pontos distantes.

Como apenas a última posição de cada ônibus é disponibilizada, e tendo como objetivo a análise de um histórico completo de registros de posição, foi necessária a criação de um *script* coletor de dados. Desenvolvido através da linguagem Python [6], o *script* tem

a sua execução agendada para cada minuto do dia que deseja-se capturar as informações. O *script* tem como tarefas: a obtenção do arquivo de posicionamento dos ônibus disponibilizado no momento de sua execução (arquivo contendo apenas a última posição de cada ônibus da cidade) e no término de um dia de execução, compactar todos os arquivos referentes ao dia e o salvar em um servidor de dados. Este processo de construção de histórico teve início no dia 16/04/2014 e já conta com mais de 3 anos de dados disponíveis para estudos. Vale ressaltar que, a fim de possibilitar uma melhor exploração dos resultados gerados pela metodologia proposta, este trabalho restringe-se aos dados capturados durante o mês de Fevereiro de 2016.

Posteriormente à conclusão do histórico de posicionamentos, uma análise em busca de possíveis inconsistências nas informações obtidas foi realizada. As sub-seções a seguir descrevem em detalhes as principais inconsistências encontradas

4.1 Inconsistências

A quantidade de informações inconsistentes na base de registros de posição fornecida pela prefeitura do Rio de Janeiro é expressiva. Os problemas existentes podem ser classificados através das seguintes categorias:

Informações fora da Cidade Informações de registros de posição dos ônibus que estejam fora da cidade estudada (Rio de Janeiro) são consideradas erros, visto que, os dados capturados possuem origem exclusiva nos ônibus que circulam por tal cidade. Do total das informações analisadas, temos que 221.517 registros (0,1% de toda a base) estão fora dos limites da cidade.

Linhas Incorretas Em teoria, todos os ônibus que estejam associados a uma determinada linha devem realizar a mesma operação, porém podemos notar a existência de ônibus com linhas iguais executando operações diferentes. Este erro pode influenciar diretamente no resultado da metodologia, pois todos os métodos são aplicados sobre os ônibus de uma mesma linha de interesse. A Figura 4.1 apresenta um exemplo visual desta categoria onde diferentes ônibus associados a uma mesma linha executam sua operação em distintas regiões geográficas da cidade.

Ausência de linha do ônibus A informação que associa um ônibus a uma linha é fundamental para a correta aplicação de nossa metodologia. Sendo assim, sua ausência

pode influenciar de forma negativa o resultado final. Temos que 22% dos registros de posição não possuem identificação da linha adotada por um veículo.

Descontinuidade de Informações Podemos observar uma descontinuidade na disponibilização das informações de algumas linha de ônibus. Em alguns casos, temos intervalos superiores a três dias.

Entre os erros citados, o mais difícil de ser tratado pela metodologia proposta é o de Linha Incorreta, pois como os algoritmos consomem dados de todos os ônibus de uma mesma linha, teremos um conflito na escolha das regiões operacionais utilizadas, visto que um grupo de veículos irá executar suas atividades sobre uma região e outros sobre diferentes regiões. Nossos algoritmos são capazes de contornar este tipo de erro, porém tendem a não funcionar corretamente na mesma proporção em que se aumenta a quantidade de veículos operando sobre diferentes regiões operacionais, pois quanto maior a quantidade, mais difícil será a distinção entre as regiões operacionais que pertencem à linha analisada e as que foram associadas de forma incorreta à linha em questão.

Figura 4.1: Posições dos ônibus da linha 908 durante o mês de Fevereiro de 2016. Dois tipos de operação em uma mesma linha de ônibus. As informações em amarelo e vermelho foram produzidas por dois diferentes conjuntos de ônibus da mesma linha.

Capítulo 5

Metodologia

Neste trabalho, é proposto um método para a extração de informações operacionais de linhas de ônibus a partir de dados de GPS de ônibus e dados cartográficos das regiões em que estes operam. As etapas, os algoritmos que compõem o método e o fluxo de execução são apresentados na Figura 5.1, conforme descrito a seguir.

As etapas do método podem ser divididas em três categorias que variam de acordo com as bases de dados utilizadas como parâmetro de entrada dos processamentos. Na primeira categoria (Inferência da Garagem, Ponto Inicial e Ponto Final) temos as etapas que utilizam-se exclusivamente dos dados do GPS dos ônibus e são utilizadas para inferir a posição geográfica da garagem e dos pontos inicial e final de cada linha. Já na segunda categoria (Extração da Malha Viária), temos as etapas que utilizam exclusivamente dados cartográficos e são responsáveis pela produção de um grafo com a descrição da malha viária da região de estudo. A terceira e última categoria (Inferência da Rota) é composta por etapas que processam os dados de GPS dos ônibus e os associam à malha viária, produzindo inferências das rotas.

A manipulação destes conjuntos de dados é feita através do PostegreSQL [5], que é um Sistema de Gerenciamento de Banco de Dados (SGBD). Este SGBD foi escolhido por ser capaz de trabalhar de forma nativa com dados espaciais através da utilização de tipos de dados geométricos e operações espaciais simples. Dados espaciais são quaisquer tipos de dados que descrevem fenômenos aos quais esteja associada a alguma dimensão espacial [9]. Outro ponto positivo deste SGBD é o fato de ser livre e de código aberto, o que facilita a sua utilização em demais trabalhos.

Além disso, como a grande maioria das etapas do método proposto consomem dados diretamente das bases de dados, muitos dos algoritmos foram desenvolvidos a partir da

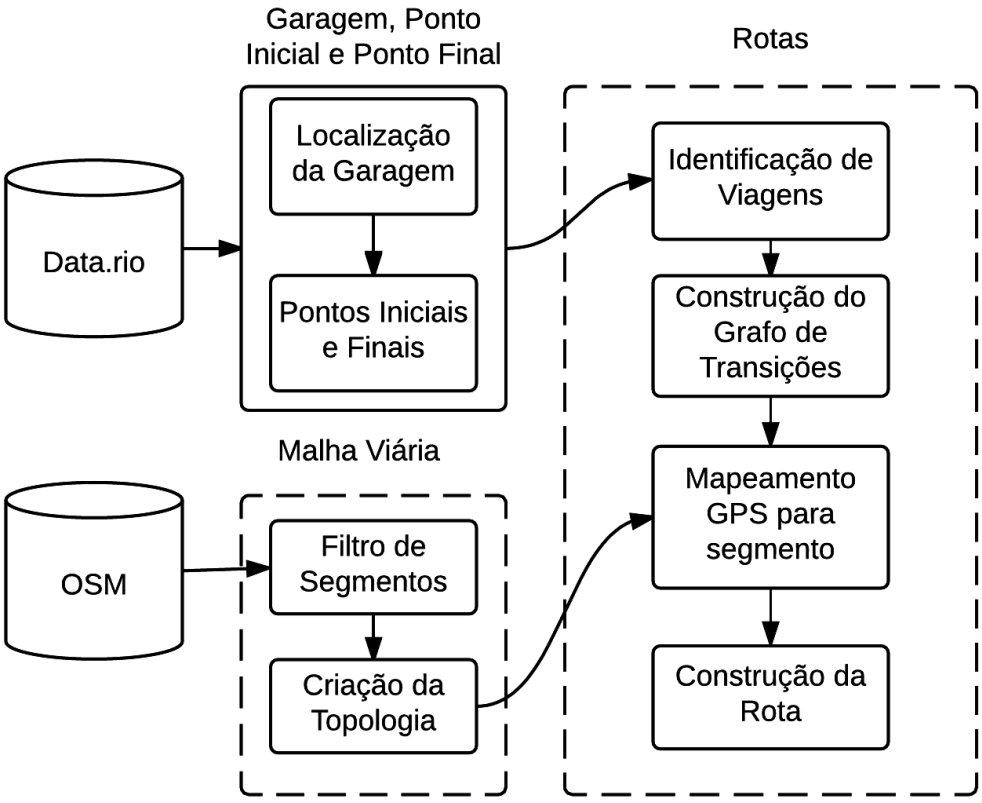


Figura 5.1: Fluxograma das etapas que compõem a metodologia

linguagem *PL/pgSQL*, que é procedural e carregável para um banco de dados PostgreSQL. A adoção desta linguagem foi motivada devido aos seguintes fatores: simplicidade no desenvolvimento, elevado grau de confiança para o servidor de dados e, principalmente, por ser capaz de facilmente realizar procedimentos complexos sobre o banco de dados.

Nas subseções a seguir, que seguem a divisão em blocos de contexto do fluxo de execução da proposta, são apresentadas e detalhadas todas as etapas.

5.1 Garagem, Ponto Inicial e Ponto Final

As regiões operacionais de transição são compostas pelas seguintes regiões: Garagem, Ponto Inicial e Ponto Final de uma linha. A identificação destas apresenta um papel fundamental na execução das demais etapas de nossa metodologia, pois representam as regiões onde ocorrem as modificações do estado operacional de um ônibus. Por exemplo, quando um ônibus sai de uma região de garagem temos o início de uma nova operação. Por outro lado, quando um ônibus adentra uma região de Ponto Final, é encerrada uma viagem, que teve origem no Ponto Inicial da rota utilizada. Como no tipo de problema estudado o estado operacional não é previamente conhecido, hipóteses devem ser elabo-

radas a fim de estimar o momento em que as modificações ocorreram, para, em seguida, poder inferir os locais (Regiões Operacionais de Transição) em que tais ações ocorrem.

Nas subseções a seguir são detalhadas, na mesma ordem de execução, as etapas de identificação da garagem, ponto inicial e ponto final de uma linha de ônibus.

5.1.1 Localização da Garagem

A garagem é o local onde começa e termina a operação de um ônibus. Para identificar estas regiões assumimos que, para a maioria dos veículos, as primeiras e as últimas posições registradas em um dia estão contidas na garagem. Com base nessa hipótese, para cada ônibus da linha de interesse, digamos B, definimos um conjunto P^B contendo os n registros de posições coletados da base de dados do Data.Rio ao longo de um conjunto de dias de operação. Seja $\{p_i^B\}_{1\leq i\leq n}$ o conjunto ordenado cronologicamente dos registros de localização, onde p_i^B representa as coordenadas espaciais de um registro. Definimos, ainda, dois subconjuntos de P^B, um com os β primeiros e outro com os β últimos registros do dia, sendo eles representados respectivamente por $I^B : \{p_i^B\}_{1\leq i\leq \beta} \in P^B$ e $F^B : \{p_i^B\}_{(n-\beta)\leq i\leq n} \in P^B$. Devemos ressaltar que β deve assumir um valor muito pequeno com relação ao valor de n.

Em seguida, definimos regiões circulares de raio r, denotadas por $R(.)$, centradas nos primeiro e último registros dos conjuntos I^B e F^B, respectivamente (i.e., nas posições p_1^B e p_n^B). Para cada região, atribuímos uma pontuação $S(.)$ que indica quantos elementos do conjunto I^B e quantos do conjunto F^B estão dentro das regiões $R(p_1^B)$ e $R(p_n^B)$, respectivamente. Assim, temos que $S(I^B)$ é a ordem do subconjunto dos elementos de I^B cuja posição $p_i^B{}_{2\leq i\leq \beta}$ encontra-se dentro da região $R(p_1^B)$ e, de forma análoga, obtemos a pontuação $S(F^B) = \{|\{p_i^B\}_{(n-\beta)\leq i\leq (n-1)}|, ||p_i^B - p_n^B|| < r\}$.

Considerando a existência de b veículos em operação para a linha analisada, definimos o conjunto G^B dos primeiros e últimos registros dos ônibus em operação em um dos dias de operação como $G^B = \{\bigcup_{k=1}^{b}(\{p_1^{B_k}\} \wedge \{p_n^{B_k}\})\}$. O conjunto tem ordem $|G^B| = 2b$ e seu elementos são denotados por $g_l \in G^B$ com $l \in \{1..2b\}$. De forma análoga, construímos o conjunto GS^B das pontuação das regiões circulares associadas aos conjuntos I^B e F^B de cada ônibus em operação. Mais precisamente, temos que $GS^B = \{\bigcup_{k=1}^{b}(\{S(I^{B_k})\} \wedge \{S(F^{B_k})\})\}$.

A estimativa da posição da garagem da linha B, representada por gar^B, é dada pelo registro em G^B que minimiza a Equação 5.1. A solução da equação indica qual registro

Notação	Descrição
B	Linha de interesse do algoritmo
P^B	Conjunto de tamanho n contendo os registros de posição, da linha de interesse, durante um conjunto de dias de operação
I^B	Subconjunto de P^B contendo os β primeiros registros de posição de um dia
F^B	Subconjunto de P^B contendo os β últimos registros de posição de um dia
$R(.)$	Região circular de raio r que tem como centroide um elemento dos conjuntos I^B ou F^B
$S(.)$	Pontuação de um elemento de P^B
b	Quantidade de veículos em um dia de operação
G^B	Conjunto contendo o primeiro e último registro de posição de um veículo em um dia de operação
GS^B	Conjunto contendo as pontuações do primeiro e último registro de posição de um veículo em um dia de operação

Tabela 5.1: Tabela de notações utilizadas pelo algoritmo de identificação da região de garagem

$g_l \in G^B$ possui a menor distância para os demais registros, ponderada pela soma das pontuações das regiões circulares dos registros em questão. O procedimento desta etapa é apresentado em detalhes através do Algoritmo 1.

$$gar^B = \min_l \sum_{m=1}^{2b} \frac{D_{l,m}}{GS_l^B + GS_m^B} \tag{5.1}$$

Onde,

$$l = \{1...2\text{b}\},\ l \neq m$$

$$D_{l,m} = \text{distância euclidiana entre os registros } g_l \text{ e } g_m.$$

5.1.2 Pontos Iniciais e Finais

Os Pontos Inicial e Final de uma linha de ônibus são as regiões onde suas viagens são iniciadas e finalizadas. Em horários de *rush*, as empresas de ônibus enviam muitos veículos para estas regiões para atender a alta demanda por transporte. Além disso, os Pontos Inicial e Final são, de todo o trajeto, os locais onde os veículos ficam mais tempo parados. Assim, observamos que a estimativa dos Pontos Iniciais e Finais das linhas e a estimativa da localização das Garagens são problemas parecidos. Sendo assim, um algoritmo semelhante ao utilizado anteriormente para estimar Garagens foi desenvolvido para encontrar os Pontos Iniciais e Finais, considerando apenas pequenas modificações.

A primeira modificação adotada foi a remoção dos registros previamente classificados como contidos dentro de garagens do *dataset*. Sendo assim, para cada ônibus da linha de interesse B, seja o conjunto P^B conforme definido anteriormente. Definimos o conjunto

Algoritmo 1 Algoritmo para identificação do registro que será o centroide da região da garagem

1: **function** CENTROIDEGARAGEM
2: **for** $k = 1; k \leq b; k++$ **do**
3: **for** $i = 2; i \leq \beta; i++$ **do**
4: **if** $p_i^{B_k} \subset R(p_1^{B_k})$ **then**
5: $S(I^{B_k}) \leftarrow S(I^{B_k}) + 1$
6: **end if**
7: **end for**
8: **for** $i = n - 1; i \geq n - \beta; i--$ **do**
9: **if** $p_i^{B_k} \subset R(p_n^{B_k})$ **then**
10: $S(F^{B_k}) \leftarrow S(F^{B_k}) + 1$
11: **end if**
12: **end for**
13: $GS^{B_k} \leftarrow S(I^{B_k}) \wedge S(F^{B_k})$
14: **end for**
15: **for** $l = 1; l \leq 2b; l++$ **do**
16: $S \leftarrow 0$
17: **for** $m = 1; m \leq 2b; m++$ **do**
18: **if** $l \neq m$ **then**
19: $S \leftarrow S + \frac{D_{l,m}}{GS_l^B + GS_m^B}$
20: **end if**
21: **end for**
22: $VS[\,l\,] \leftarrow S$
23: **end for**
24: $gar^B \leftarrow min_l(VS[\,])$
25: **end function**

PL^B, como o conjunto dos n registros de posição coletados da base de dados do Data.Rio durante um conjunto de dias de operação, excluindo os registros contidos nas regiões de garagem. Assim, temos $PL^B : \{p_i^B\}_{1 \leq i \leq n} \notin R(gar^B)$.

Para cada registro $p_t^B \in PL^B$, atribuímos uma pontuação $SL^B(.)$ que indica quantos elementos subsequentes a p_t^B em PL^B estão dentro da região circular $R(p_t^B)$ de raio r, centrada no ponto p_t^B. Iterando sobre os elementos de PL^B, construímos outro conjunto PR^B cujos elementos são obtidos da seguinte forma: dado um registro $p_t^B \in PL^B$, o mesmo pertence a PR^B se e somente se for válida a condição $\alpha : 5 \leq SL(p_t^B) \leq 10$. Após a análise do registro p_t^B, a iteração segue para o primeiro registro fora da região $R(p_t^B)$. Para cada registro $p_t^B \in PL^B$ adicionado ao conjunto PR^B, adicionamos também o valor $SL^B(p_t^B)$ ao conjunto das pontuações SR^B.

O objetivo da condição α é selecionar regiões da cidade onde os ônibus ficam parados apenas por alguns minutos e eliminar casos em que eles ficam parados por longos períodos de tempo, devido a problemas mecânicos, por exemplo. Resultados apresentados na Seção 6 discorrem mais sobre os parâmetros desta condição.

Considerando a existência de b veículos em operação para a linha analisada, temos então b conjuntos PR^B e outros b conjuntos SR^B identificados como PR_d^B e SR_d^B, para $d \in \{1..b\}$. A estimativa do ponto inicial, denotada por $inicial^B$, é feita a partir de dois grandes conjuntos $PIF^B = \bigcup_{d=1}^{b} PR_d^B$ e $SIF_d^B = \bigcup_{d=1}^{b} SR_d^B$ através da Equação 5.2. Assim como no caso das garagens, a solução da equação indica o registro $pif_d \in PIF^B$ com menor distância para os demais registros, ponderada pela soma das pontuações das regiões circulares dos registros em questão.

$$inicial^B = \min_d \sum_{m=1}^{b} \frac{D_{d,m}}{SIF_d^B + SIF_m^B} \tag{5.2}$$

Onde,

$d = \{1...\text{b}\}$, $d \neq m$

$D_{d,m} =$ distância entre os registros pif_d e pif_m.

Analogamente, removendo dos conjuntos PIF_d^B e SIF_d^B os elementos associados aos registros que estão dentro da região circular $R(p_d^B)$, definimos os conjuntos $\overline{PIF}_d^B$ e $\overline{SIF}_d^B$ que são usados para estimar o ponto final, denotado por $final^B$ através da Equação 5.3.

Todo este procedimento, que tem como objetivo identificar o ponto inicial e ponto final de uma linha de ônibus, é apresentado em detalhes através do Algoritmo 2.

Notação	Descrição
PL^B	Conjunto dos registros de posição que não estão contidos nas regiões de garagens
$SL^B(.)$	Pontuação de um elemento p_t^B
PR^B	Elementos de PL^B que satisfazem a condição α
$inicial^B$	Centroide da região de ponto inicial da linha B
$final^B$	Centroide da região de ponto final da linha B
b	Quantidade de veículos em um dia de operação
SIF^B	Conjunto com as pontuações $S(.)$ dos elementos pertencentes ao conjunto I^B.
PIF^B	Conjunto com os elementos $p_t^B \in PL^B$ e que atendam a condição α.
TR	Estrutura de dados auxiliar contendo uma região circular do tipo $R(.)$.

Tabela 5.2: Tabela de notações utilizadas pelo algoritmo de identificação das regiões de ponto inicial e final

$$final^B = \min_d \sum_{m=1}^{b} \frac{D_{d,m}}{\overline{SIF}_d^B + \overline{SIF}_m^B} \tag{5.3}$$

Onde,

$$d = \{1...\text{b}\},\ d \neq m$$

$$D_{d,m} = \text{ distância entre os registros } pif_d \text{ e } pif_m.$$

5.2 Malha Viária

Os dados cartográficos utilizados como insumo deste trabalho foram obtidos através do Open Street Map (OSM), que é um dos mais famosos projetos de contribuição voluntária de informações geográficas, contendo mais de 2.5 milhões de usuários inscritos [4]. Seu principal objetivo consiste na criação de um mapa capaz de representar todo o planeta e que seja acessível através da *Internet* para toda a população.

As informações cartográficas utilizadas neste trabalho restringem-se apenas aos dados referentes a cidade do Rio de Janeiro, pois é a mesma cidade em que os ônibus estudados realizam sua operação. Este subconjunto de dados foi obtido através de uma API (Application Programming Interface) fornecida pelo próprio, onde são obtidos apenas os dados de uma determinada região.

A representação dos dados do OSM respeita a premissa básica, definida no planejamento de seu projeto, de que deve ser a mais simples e independente possível. Neste sentido, são utilizados em sua construção apenas elementos de 4 tipos, sendo os seguintes: (1) Nó, um par de coordenadas geográficas (Latitude, Longitude), (2) Caminho, segmentos de reta que passa de forma ordenada por um conjunto de nós, (3) Área, um caminho fechado, que começa e termina no mesmo nó e (4) Relação, toda e qualquer relação sobre

Algoritmo 2 Algoritmo para identificação dos centroides das regiões de ponto inicial e final de uma linha

1: **function** INICIALFINAL
2: **for** $d = 1; d \leq b; d++$ **do**
3: $q \leftarrow 1$
4: $TR \leftarrow R(p_q^{B_d})$
5: **for** $w = 2; w \leq |PL^B|; w++$ **do**
6: **if** $p_w^{B_d} \subset TR$ **then**
7: $SL(p_w^{B_d}) \leftarrow SL(p_w^{B_d}) + 1$
8: **else**
9: **if** $SL(p_w^{B_d}) \geq 5$ && $SL(p_w^{B_d}) \leq 10$ **then**
10: $PIF^B \leftarrow p_w^{B_d}$
11: $SIF^B \leftarrow SL(p_w^{B_d})$
12: **end if**
13: $q \leftarrow w$
14: $TR \leftarrow p_q^{B_d}$
15: **end if**
16: **end for**
17: **end for**
18: **for** $d = 1; d \leq b; d++$ **do**
19: $s \leftarrow 0$
20: **for** $m = 1; m \leq b; m++$ **do**
21: $s \leftarrow s + \frac{D_{d,m}}{SIF_d^B + SIF_m^B}$
22: **end for**
23: $VS[\,d] \leftarrow s$
24: **end for**
25: $inicial^B \leftarrow min_d(VS[\,])$
26: **for** $d = 1; d \leq b; d++$ **do**
27: $\overline{s} \leftarrow 0$
28: **for** $m = 1; m \leq b; m++$ **do**
29: $\overline{s} \leftarrow \overline{s} + \frac{D_{d,m}}{\overline{SIF}_d^B + \overline{SIF}_m^B}$
30: **end for**
31: $\overline{VS}[\,d] \leftarrow \overline{s}$
32: **end for**
33: $final^B \leftarrow min_d(\overline{VS}[\,])$
34: **end function**

um conjunto de elementos. Além dos elementos estarem associados a um único tipo, estes podem conter quantas *tags* desejar. Tais *tags* servem para descrição dos tipos de um objeto (Ex.: rua, restaurante e praça) e seus detalhes mais relevantes, tais como rua de acesso restrito.

Como os dados disponibilizados pelo OSM representam diversas informações geográficas de uma cidade, e o objetivo deste trabalho consiste na análise de dados de operações de ônibus, temos a necessidade de selecionar apenas os dados relativos a este contexto. Além disso, uma modificação da representação das informações do OSM, a fim de adequá-la ao contexto da operação dos ônibus também se faz necessária.

5.2.1 Filtro dos Segmentos

Ao deslocar-se pela cidade, um indivíduo pode utilizar-se de diferentes tipos de percursos durante o seu trajeto, como por exemplo: andar a pé pela **calçada**, utilizar um carro em uma **rodovia** ou até mesmo utilizar uma bicicleta em uma **ciclovia**. Devido a esta grande variedade de tipos de caminhos (percursos) em uma cidade e pelo fato do OSM conter informações de inúmeros tipos, torna-se necessário a utilização de um filtro capaz de selecionar apenas tipos de caminhos em que um ônibus tenha permissão de utilizar durante o seu deslocamento pela sua rota.

Mais especificamente, um ônibus tem a permissão de utilizar, por exemplo, uma rodovia, uma rua e uma avenida. Porém, o seu acesso não é permitido em calçadas, ciclovias e ruas privadas. A adoção deste filtro nos permite garantir que o resultado produzido pelo algoritmo de inferência da rota de uma linha contenha apenas elementos em que um ônibus possa de fato trafegar.

5.2.2 Criação da topologia

Pelo fato do OSM ter como objetivo principal a disponibilização dos dados geográficos de uma região, a sua representação, como dito anteriormente, é simples e genérica. Desta forma, estudos com objetivos similares a este, ou que por venturam desejam utilizar os dados do OSM como insumo de análises, devem de alguma forma modificar e adaptar os dados a sua necessidade.

Neste sentido, foi realizada uma adaptação na estrutura dos dados disponibilizados a fim de torná-los roteáveis, ou seja, que permitissem análises sobre possíveis caminhos (ou percursos). Esta modificação foi necessária, pois os dados do OSM são não roteáveis. Isto

deve-se à forma pela qual os dados são cadastrados no sistema pelos colaboradores. Temos como exemplo que, ao cadastrar uma nova informação referente a uma rua, o colaborador não possui a obrigatoriedade de informar a relação desta rua com as demais, ou seja, este procedimento introduz uma independência no processo de colaboração dos dados. Tal independência faz com que a sinalização explícita de interseções entre segmentos de rua não sejam de caráter obrigatório, tornando assim os dados não roteáveis. A Figura 5.2(A) ilustra a ausência desta informação para um conjunto de ruas.

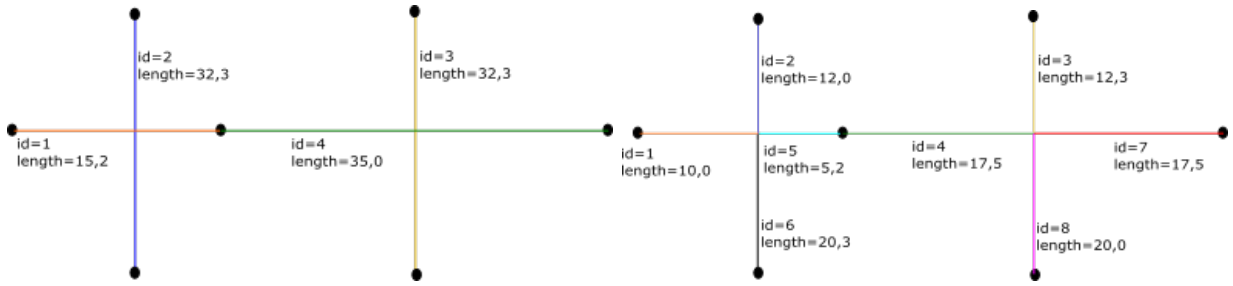
A adaptação implementada se da pela transformação da antiga estrutura dos dados para uma nova, baseada em um grafo direcionado e com pesos em suas arestas. Inicialmente, são obtidas as regiões em que os segmentos de ruas se interceptam. Por mais que este tipo de informação não esteja contida de forma explícita nos dados, é possível identificar os cruzamentos de segmentos de ruas através da análise geométrica de sua formação. Em seguida, são desmembrados os segmentos de acordo com as interceptações de cada um. Além disso, cada novo segmento desmembrado recebe um identificador único, como pode ser visto na Figura 5.2(B). Sendo assim, cada segmento desmembrado representa uma aresta do grafo. O peso de cada aresta é obtido através do cálculo do tamanho (comprimento) do respectivo segmento.

Em seguida, os nós do grafo são criados a partir da seleção distinta das extremidades dos segmentos desmembrados anteriormente. É utilizada a seleção de forma distinta, pois mais de dois segmentos podem ter o mesmo ponto geográfico de extremidade. Neste sentido, um nó do grafo poderá ter relação (caminho) com um ou mais segmentos. Além disso, a orientação da formação dos segmentos é mantida na estrutura do grafo.

Por fim, a Figura 5.2(C) apresenta o resultado final das adaptações para transformação da antiga estrutura dos dados não roteáveis em uma estrutura baseada em grafos roteaveis.

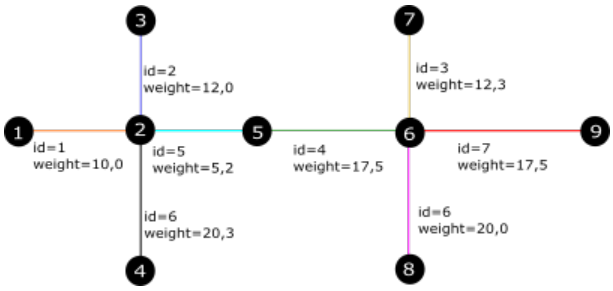
5.3 Rotas

O processo de identificação da rota tem como objetivo extrair a trajetória, rua à rua, utilizada pelos ônibus de uma mesma linha durante as transições entre os pontos iniciais e finais. Para cada linha de interesse, B, nosso método extrai um conjunto de rotas denotado por R^B, que contém as duas rotas da linha. São identificadas duas rotas pois a que possui origem no ponto inicial e destino no ponto final é diferente da que possui a origem no ponto final e destino no ponto inicial da linha. Sendo assim, temos os seguintes subconjuntos de R^B: RIF^B que contém todos os ni segmentos de rua, ordenados, utilizados na rota



(a) Topologia sem identificação das interseções dos segmentos e sem identificadores dos nós

(b) Desmembramento dos segmentos e atribuição de códigos únicos, de acordo com as interseções



(c) Resultado final da modificação do grafo contendo os identificadores de nós e arestas

Figura 5.2: Resultados parciais da etapa de modificação da estrutura do OSM em uma topologia roteavel.

que tem origem no ponto inicial e destino no ponto final e outro subconjunto RFI^B que contém os nf segmentos de rua com origem no ponto final e destino no ponto inicial.

A primeira etapa deste processo consiste em identificar os registros de posição dos ônibus da linha de interesse que foram produzidos durante o deslocamento entre os ponto inicial e final. Tais deslocamentos serão aqui chamados de viagens. Em seguida, a fim de identificar regiões geográficas similares utilizadas durante as viagens de todos os ônibus, é criado um grid regular capaz de associar cada registro de posição a uma célula do grid. Este grid é dito regular pois todas as suas células possuem a mesma dimensão. As transições entre as células geradas pelas viagens são mapeadas para um grafo direcionado. O grafo criado é analisado em busca da identificação do caminho direcionado mais utilizado pelos ônibus. Este caminho é composto pelas células do grid ordenadas de acordo com o deslocamento. Por fim, os segmentos de rua contidos nas células do caminho obtido são analisados através de um algoritmo de *Map-Matching*, que visa identificar os segmentos que de fato foram utilizados pela maioria dos ônibus. Todas as etapas citadas brevemente até aqui são detalhadas a seguir.

5.3.1 Identificação de Viagens

Uma viagem de um ônibus tem início no momento em que este sai de uma região definida como ponto inicial ou ponto final. Analogamente, uma viagem termina no momento em que este entra em uma região de ponto oposto ao que havia partido. Neste sentido, temos que ao adentrar ou sair de uma região deste tipo, uma viagem obrigatoriamente será encerrada e outra será iniciada.

Representaremos uma viagem de uma linha de interesse B através do conjunto V^B. Este conjunto é um subconjunto de PL^B e contém os l elementos $p_t^B \in PL^B$ que foram produzidos, de forma ordenada, durante o deslocamento do ônibus entre as regiões de ponto inicial e ponto final. Para representar estas regiões, utilizaremos a notação $RP(.)$ que simboliza uma região circular de raio rp, centrada em um dos registros de posição identificados como $inicial^B$ ou $final^B$ na etapa anterior da metodologia deste trabalho.

A identificação de uma viagem que é composta por elementos $v_l^B \in V^B$ com origem em uma região $RP(or)$ e que tem como destino à região $RP(ds)$ ocorre da seguinte forma: durante a iteração dos elementos de PL^B, marcadores de início ou fim de uma viagem são criados. No momento em que uma informação iterada ($p_t^B \in PL^B$) encontra-se contida em $RP(or)$, e esta não possui um marcador, um novo marcador de início referente a região é criado e associado à informação analisada. Caso o registro de posição imediatamente seguinte também encontre-se em $RP(or)$, o marcador deve ser então associado a este registro. No momento em que as informações subsequentes não estejam mais contidas em nenhuma região definida anteriormente e o procedimento já tenha um marcador de início, estas informações serão armazenadas em um conjunto temporário de viagens (VT^B). Os elementos deste conjunto temporário somente farão parte de V^B caso exista um registro de posição subsequente contido em $RP(ds)$.

O processo de identificação de todas as viagens de B entre as regiões $RP(inicial^B)$ e $RP(final^B)$ durante um conjunto de dias de operação utiliza-se de dois conjuntos. O primeiro destes é denotado por VIF^B que contém as viagens com início em $RP(inicial^B)$ e o outro é denotado por VFI^B que contém as viagens que iniciam-se em $RP(final^B)$. A formação destes conjuntos, respectivamente, ocorre da seguinte forma: considerando a existência de y viagens com origem em $RP(inicial^B)$ em um dia de operação temos então y conjuntos de V^B identificados como V_o^B, para $o \in \{1..y\}$. Por outro lado, considerando a existência de u viagens com origem em $RP(final^B)$ temos então u conjuntos de V^B identificados como V_z^B, para $z \in \{1..u\}$.

Notação	Descrição
PL^B	Conjunto dos registros de posição (p_t^B) que não estão contidos nas regiões de garagens.
$RP(.)$	Região circular de raio rp, centrada em um dos registros de posição identificados como $inicial^B$ ou $final^B$.
V^B	Conjunto dos registros de posição (v_l^B) que foram gerados durante o deslocamento entre o ponto inicial e ponto final, ou vice-versa.
VOD^B	Conjunto de todas as viagens (V^B) entre uma região de origem ($RP(origem)$) e destino ($RP(destino)$).
$marcador$	Variável auxiliar para marcar o início de uma viagem.

Tabela 5.3: Tabela de notações utilizadas pelo algoritmo de extração das viagens de um ônibus

O algoritmo desta etapa é responsável por gerar os elementos de VIF^B e VFI^B sendo um destes por vez. A construção destes pelo algoritmo irá depender dos parâmetros de entrada que são: região de origem das viagens ($RP(or)$) e destino ($RP(ds)$). Tal procedimento é apresentado em detalhes no Algoritmo 3.

Algoritmo 3 Algoritmo para extração das viagens de um ônibus

Precondition: *origem* região de origem, *destino* região de destino

1: **function** EXTRAIVIAGENS
2: $u \leftarrow 0$
3: $marcador \leftarrow 0$
4: **for** $t = 0; t \leq |PL^B|; t++$ **do**
5: **if** $p_t^B \subset RP(origem)$ **then**
6: marcador ← t
7: $V_0^B \leftarrow p_t^B$
8: $l \leftarrow 1$
9: **else**
10: **if** $marcador \neq 0$ **then**
11: $V_l^B \leftarrow p_t^B$
12: $l \leftarrow l + 1$
13: **end if**
14: **end if**
15: **if** $p_t^B \subset RP(destino)$ && $marcador \neq 0$ **then**
16: $V_l^B \leftarrow p_t^B$
17: $VOD_u^B \leftarrow V^B$
18: $u \leftarrow u + 1$
19: $marcador \leftarrow 0$
20: $l \leftarrow 0$
21: **end if**
22: **end for**
23: Return VOD^B
24: **end function**

5.3.2 Grafo

Para guiar o estudo sobre o comportamento adotado pelos ônibus durante suas viagens, é utilizado como base um grafo direcionado que possui os deslocamentos agrupados em pequenas regiões geográficas. A elaboração deste grafo tem início na construção de um *grid* regular capaz de abranger em cada uma de suas células todas as posições geradas durante as viagens dos ônibus.

A adoção deste *grid* possibilita a divisão do espaço geográfico utilizado pelos ônibus em sub-regiões similares. Este tipo de estratégia de resolução é conhecida na literatura e é utilizada sobre informações de GPSs produzidas por diferentes modais de uma cidade. Temos como exemplo [27], onde esta é aplicada sobre informações geradas pelos táxis da cidade de Hangzhou da China.

A dimensão das células do grid é um parâmetro que pode variar de acordo com a precisão dos dispositivos GPS responsáveis por capturarem e disponibilizarem o posicionamento dos veículos. Como a precisão dos dados utilizados neste trabalho é baixa, as células do grid devem ter uma dimensão relativamente grande para minimizar os efeitos negativos. Neste sentido, este trabalho utiliza dois diferentes valores (50 e 100 metros quadrados) no tamanho das células durante os testes da aplicação da metodologia. A Seção 6 apresenta em detalhes os resultados obtidos através da utilização destes diferentes valores como parâmetro.

A criação do grid ocorre da seguinte forma: para cada conjunto de viagens VIF^B ou VFI^B é criado um grid contendo c células de tamanho d metros quadrados, onde todos os registros de posição ($v_l^B \in V^B$) gerados durante as viagens estão alocados em, pelo menos, uma das células. Este grid será aqui representado pelo seguinte conjunto $GR^B : \{gr_w^B\}_{1 \leq w \leq c}$, onde gr_w^B representa uma célula do grid.

Posteriormente à criação do grid, é construído um grafo acíclico direcionado e ponderado, contendo todas as transições executadas pelos veículos entre as células do grid durante as viagens analisadas. Este grafo será aqui denotado como $GF^B = (VR, A)$, onde VR representa o conjunto de vértices e A o conjunto de arestas. O conjunto VR é composto pelos elementos do conjunto GR^B, ou seja, este é responsável por representar as células do grid. Assim, temos $VR : \{gr_w^B\}_{1 \leq w \leq c} \in GR^B$. Já as arestas do grafo são construídas a partir das transições entre as células do grid realizadas pelos ônibus durante as viagens e são representadas da seguinte forma: (υ, ϕ), onde esta aresta sai do vértice υ e entra no vértice ϕ. Mais especificamente, os seus elementos são obtidos da seguinte

Notação	Descrição
VOD^B	Conjunto de todas as viagens (V^B) entre uma região de origem ($RP(origem)$) e destino ($RP(destino)$).
V^B	Conjunto dos registros de posição (v_l^B) que foram gerados durante o deslocamento entre o ponto inicial e ponto final, ou vice-versa.
GR^B	Conjunto de células do grid regular composto por elementos gr_w^B
VR	Conjunto de vértices do grafo GF^B
A	Conjunto de arestas do tipo (υ, ϕ) pertencentes a GF^B
$Peso(.)$	Função de obtenção dos pesos de uma aresta do tipo (υ, ϕ)
GF^B	Conjunto que representa o grafo de transições entre as células do grid

Tabela 5.4: Tabela de notações utilizadas pelo algoritmo de construção do grafo de transições entre as células do grid

forma: uma iteração sobre o conjunto V^B é realizada e caso um elemento $v_l^B \in V^B$ esteja contido em uma célula $gr_w^B \in GR^B$ e seu subsequente $v_{l+1}^B \in V^B$ também esteja contido em uma célula $gr_h^B \in GR^B$, com $h \neq w$, será então criado o elemento a_o correspondente a aresta (w, h) com valor de peso igual a 1. Porém, caso a aresta representada pelo elemento a_o já pertença a A, seu peso será incrementado em 1 unidade. Todo este procedimento é apresentado no Algoritmo 4.

Algoritmo 4 Algoritmo para construção do grafo de transições entre as células do grid

Precondition: VOD^B conjunto de viagens

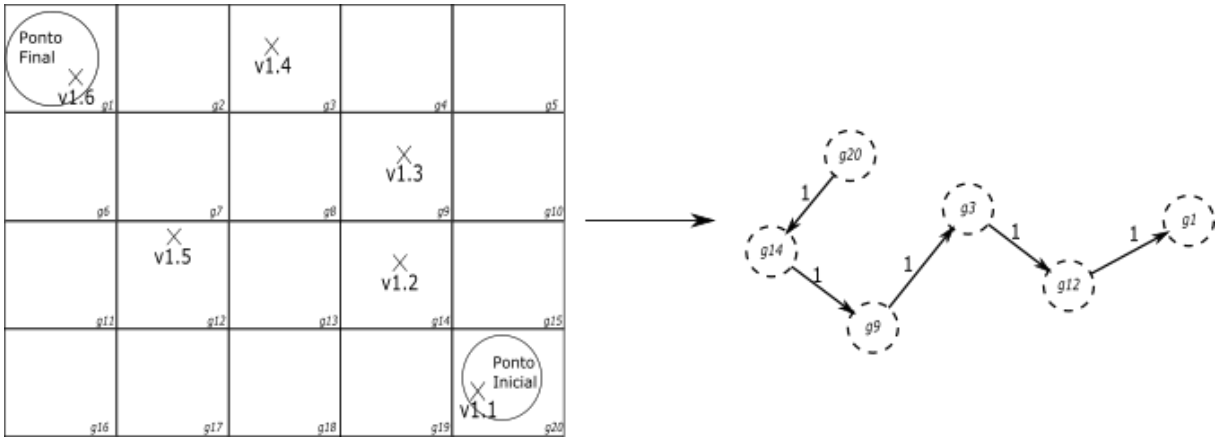
```
1:  function CONSTRUCAOGRAFO
2:      for i = 0; i ≤ |VOD^B|; i++ do
3:          V^B ← VOD_i^B
4:          for l = 0; l ≤ |V^B|; l++ do
5:              υ ← 0
6:              φ ← 0
7:              for w = 0; w ≤ |GR^B| − 1; w++ do
8:                  if V_l^B ∈ gr_w^B && V_{l+1}^B ∉ gr_w^B then
9:                      υ ← gr_w^B
10:                 end if
11:                 if V_l^B ∉ gr_w^B && V_{l+1}^B ∈ gr_w^B then
12:                     φ ← gr_w^B
13:                 end if
14:             end for
15:             VR_i ← υ
16:             A_i ← (υ, φ)
17:             Peso(A_i) ← Peso(A_i) + 1
18:         end for
19:         GF^B ← (VR_i, A_i)
20:     end for
21:     Return GF^B
22: end function
```

Afim de exemplificar este cenário, a Figura 5.3 apresenta em detalhes a construção iterativa do grafo GF^B a partir da análise de três viagens de uma linha de ônibus. Como podemos observar através da Figura 5.3(A), a primeira viagem a ser analisada pelo algoritmo faz com que o grafo seja criado e todas as suas arestas apresentem o valor de peso igual a 1. Já na Figura 5.3(B) temos uma atualização dos valores dos pesos de algumas arestas e a criação de novos vértices. Podemos perceber que o peso atribuído a cada aresta corresponde a soma das viagens em que uma mesma transição foi realizada. Por fim, a Figura 5.3(C) apresenta o grafo completo gerado a partir da análise de todas as viagens.

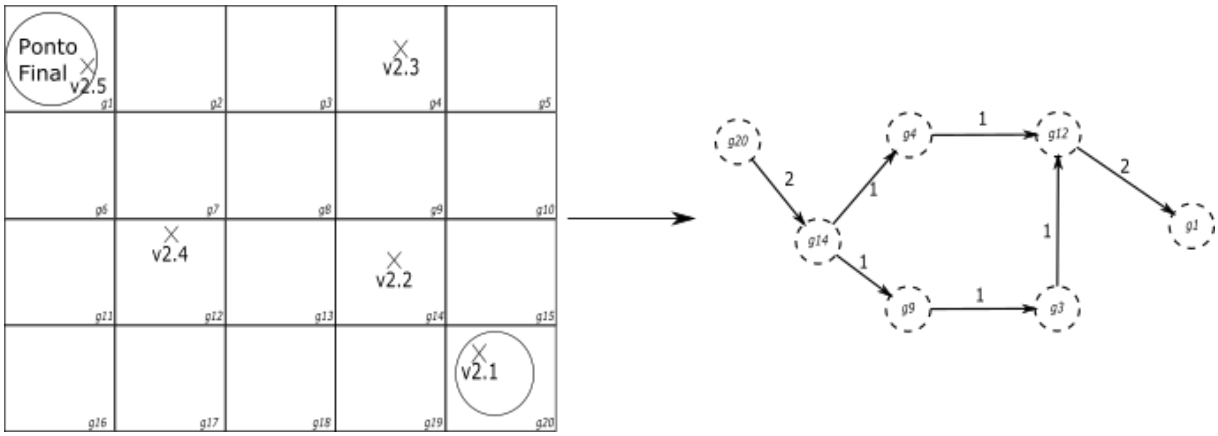
Com o grafo completo, a próxima atividade consiste na identificação do caminho que contém as arestas e os vértices mais utilizados. Este caminho deve ter como início um vértice com grau de entrada igual a zero, grau de saída maior do que zero e deve atender a condição de que a sua célula correspondente no grid deve interceptar a região delimitada pelo ponto inicial ($RP(inicial^B)$) ou ponto final ($RP(final^B)$) da linha. Por outro lado, o término deste caminho deve ocorrer sobre um vértice com grau de entrada maior do que zero, grau de saída igual a zero e que a célula do grid deva ser interceptada pela região de transição oposta a de início. Caso mais de um vértice seja compatível como possível vértice de início, apenas um destes deve ser selecionado. Sendo assim, é utilizado como critério de desempate a quantidade de registros de posição contidos na célula correspondente, ou seja, é selecionada aquela que tiver a maior quantidade.

Este caminho é obtido a partir da aplicação de um algoritmo guloso, que concentra-se apenas nas informações de vizinhanças locais de um vértice. Mais especificamente, este algoritmo inicia o seu processamento na análise da vizinhança do vértice identificado como inicial. Nesta análise é selecionado o vizinho direto que apresente o maior peso em sua aresta e que não faça parte do caminho já obtido, pois não são aceitos ciclos neste caminho. Em seguida, tal análise é repetida sucessivamente para os demais vértices selecionados até que o vértice iterado faça parte do conjunto que contém os possíveis vértices de fim de caminho. Este conjunto é necessário para garantir que o caminho inferido sempre irá terminar sobre um vértice que de fato esteja associado como fim de caminho. Esta estratégia gulosa tem como objetivo identificar o caminho que, em média, seja mais utilizado pelos ônibus de uma linha. Além disso, sabe-se que a estratégia gulosa apresenta uma rápida taxa de execução. Por fim, este algoritmo, que é apresentado em Algoritmo 5, dá origem a um subgrafo de GF^B denominado $GFCA^B$ que contém todos os vértices e arestas mais utilizados no grafo GF^B.

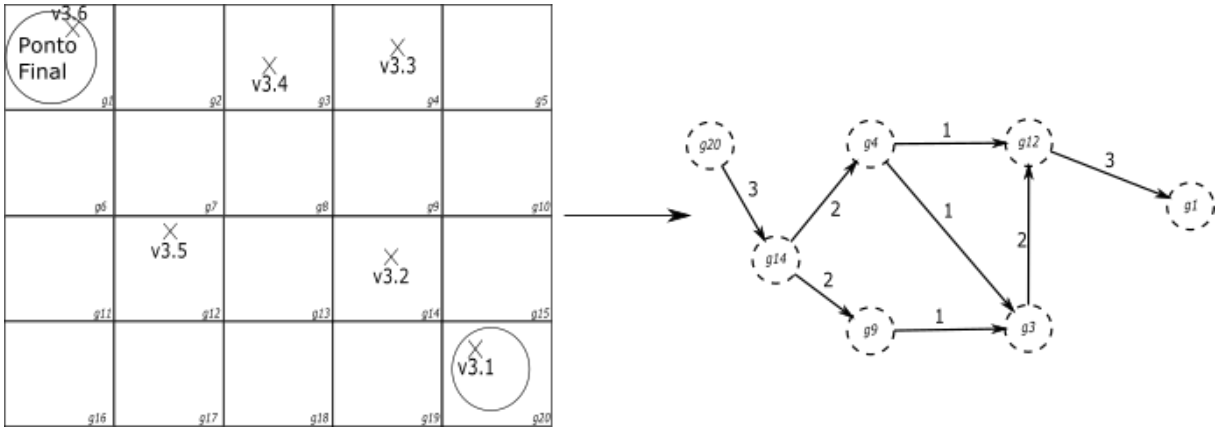
Com o novo grafo ($GFCA^B$) obtido, é possível garantir que os ônibus durante as



(a) Construção do grafo a partir da análise da viagem número 1



(b) Atualização e incremento do grafo a partir da análise da viagem número 2



(c) Atualização e incremento do grafo a partir da análise da viagem número 3

Figura 5.3: Etapas de construção e atualização do grafo que representa o comportamento adotado pelos ônibus de uma linha durante suas viagens.

Algoritmo 5 Algoritmo para construção do caminho do grafo GF^B através da estratégia gulosa

Precondition: GF^B grafo de viagens

```
1: function CAMINHOGRAFOESTRATEGIAGULOSA
2:     i ← 0
3:     indiceMaiorPeso ← −1
4:     verticeAtual ← VerticeInicio(GF^B)
5:     VPVF[] ← PossiveisVerticesFinais(GF^B)
6:     maiorPeso ← 0
7:     VRCA_0 ← verticeAtual
8:     while verticeAtual ∉ VPVF do
9:         VAD[] ← ArestasComOrigemEm(verticeAtual)
10:        for j = 0; j ≤ VAD.length; j++ do
11:            if VRCA == VerticeDestinoDaAresta(VAD[j]) then
12:                Remover elemento de indice j de VAD
13:            end if
14:        end for
15:        maiorPeso ← Peso(VAD[0])
16:        indiceMaiorPeso ← 0
17:        for j = 1; j ≤ VAD.length; j++ do
18:            pesoAtual ← Peso(VAD[j])
19:            if maiorPeso ≤ pesoAtual then
20:                maiorPeso ← pesoAtual
21:                indiceMaiorPeso ← j
22:            end if
23:        end for
24:        i ← i + 1
25:        VRCA_i ← VerticeDoDestinoDaAresta(VAD[indiceMaiorPeso])
26:        ACA_i ← VAD[indiceMaiorPeso]
27:        verticeAtual ← VRCA_i
28:        i ← i + 1
29:    end while
30:    GFCA^B ← (VRCA, ACA)
31:    Return GFCA^B
32: end function
```

Notação	Descrição
GF^B	Conjunto que representa o grafo de transições entre as células do grid
$GFCA^B$	Subgrafo de GF^B que contém todos os vértices e arestas obtidos pelo algoritmo de extração do caminho de estratégia gulosa.
$VRCA$	Conjunto de vértices do grafo $GFCA^B$
ACA	Conjunto de arestas do grafo $GFCA^B$
$VAD[]$	Estrutura de dados auxiliar contendo as arestas que tem origem em um determinado vértice.
$VPVF[]$	Estrutura de dados auxiliar contendo os possiveis vértices de fim de caminho.

Tabela 5.5: Tabela de notações utilizadas pelo algoritmo de construção do caminho do grafo GF^B através da estratégia gulosa

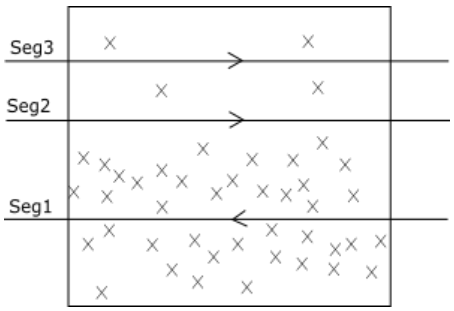


Figura 5.4: Exemplo de uma célula que intercepta 3 segmentos de rua e contém um conjunto de posições geográficas, representadas através de um X na imagem, geradas pelos ônibus de uma linha durante suas viagens.

viagens passam de forma ordenada pelas células do *grid* pertencentes a este. Estas células por sua vez contemplam uma pequena área onde devem ser interceptados 1 ou mais segmentos de rua do mapa. Sendo assim, a próxima etapa desta metodologia tem como objetivo identificar, dentre os segmentos interceptados por estas células, quais de fato foram utilizados pela maioria dos ônibus durante as viagens.

5.3.3 Mapeamento GPS para segmento

Na etapa anterior foram obtidas as células do grid (regiões geográficas) utilizadas em sequência pela grande maioria dos ônibus de uma linha durante suas viagens. Como as células do grid englobam um conjunto de registros de posição dos ônibus, estas também **devem** interceptar um ou mais segmentos de rua. É dito devem, pois podem existir células que não sejam capazes de interceptar nenhum segmento de rua. Esta situação ocorre devido à baixa precisão dos dispositivos de GPSs. Um exemplo real deste caso é apresentado na Seção 6, onde são detalhados e comentados os resultados aqui obtidos. A Figura 5.4 apresenta um exemplo de uma célula que intercepta 3 segmentos e contém registros de posição que foram produzidas durante as diferentes viagens dos ônibus.

A interceptação de mais de um segmento pelas células do grid introduz um problema na identificação do correto segmento de rua utilizado pelos ônibus. O problema de associar um segmento de rua à uma coordenada geográfica é conhecido na literatura como *Map-Matching*. Este tipo de problema é foco de estudo de diversos trabalhos [7, 29, 10, 18]. Porém, este trabalho apresenta uma particularidade neste tipo de problema, pois devemos analisar um conjunto de registros de posição produzidas por diferentes veículos e em diferentes momentos para identificar o segmento utilizado pela maioria destes. Já o problema tradicional de *Map-Matching* consiste na análise de registros de posição produzidas por um mesmo dispositivo e em ordem cronológica de geração para associá-los ao segmento de rua utilizado [22].

Para resolução deste problema em especifico, nossa estratégia consiste na avaliação de um estimador que relaciona cada registro de posição dos ônibus contidos na célula do grid com todo segmento de rua interceptado pela mesma. Esta avaliação consiste na análise do erro quadrático médio gerado pelas estimativas. A seguir, os passos utilizados para obtenção das estimativas e avaliação dos erros são detalhados.

Inicialmente, cada registro de posição (l) produzido durante uma viagem contido em uma célula (w) do grid é projetado ortogonalmente sobre todos os segmentos interceptados pela mesma célula. Neste sentido, assumimos que este ponto projetado ortogonalmente corresponde à posição em que a informação de GPS deveria ser fornecida, caso o ônibus tivesse utilizado o segmento em questão. Sendo assim, temos que o estimador P_{li} representa o ponto l projetado sobre o segmento de rua i.

O cálculo do erro do estimador é definido pela Equação 5.4. Neste caso, a diferença entre o valor estimado e o valor real é calculado através da distância entre dois pontos de um plano, onde o valor verdadeiro consiste no registro de posição e o valor calculado como o ponto projetado.

$$e_w(p_l, q_i) = P_{li} - p_l \tag{5.4}$$

Onde:

w = índice da célula
p_l = l-ésimo log de gps contido na célula w
q_i = i-ésimo segmento interceptado pela célula w
P_{li} = projeção ortogonal do ponto l sobre o segmento i

O erro quadrático médio (EQM) é utilizado a fim de penalizar os maiores valores

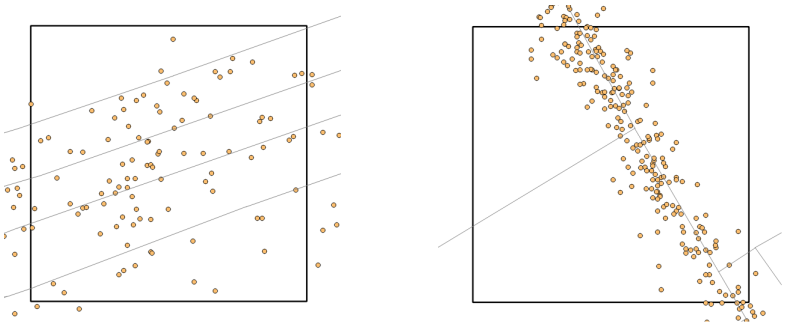
obtidos no cálculo do erro do estimador. O EQM eleva ao quadrado o valor do erro e divide pela quantidade de amostras analisadas. Para cada segmento interceptado pela célula é calculado o somatório dos erros quadráticos médios gerados por todas as informações de posicionamento de GPS contidas na mesma célula. Para isto, temos a Equação 5.5.

$$EQM_w(q_i) = \frac{1}{m} \sum_{l=1}^{m} e_w(p_l, q_i)^2 \tag{5.5}$$

Onde:

w = índice da célula
q_i = i-ésimo segmento interceptado pela célula w
p_l = l-ésimo log de gps contido na célula w
m = quantidade de logs de GPS na célula w

Por fim, o segmento com menor EQM será considerado como o segmento mais provável de ter sido utilizado pelo conjunto de ônibus analisados. É dito mais provável pois a baixa precisão dos dispositivos GPS em alguns pontos da cidade fazem com que o algoritmo utilizado nesta etapa não consiga identificar o correto segmento de rua em todos os casos. Um exemplo desta situação é representado através da Figura 5.5, onde são exibidas duas diferentes células do grid, de uma mesma rota, em que em uma delas a precisão do dispositivo GPS é alta (Figura 5.5.(B)) e em outra é muito baixa (Figura 5.5.(A)). Sendo assim, a próxima etapa do processo de identificação da rota leva em consideração estes possíveis erros na resolução de seu processamento.



(a) Célula do grid pertencente ao caminho da linha 422 em que a precisão do GPS é baixa.

(b) Célula do grid pertencente ao caminho da linha 422 em que a precisão do GPS é alta.

Figura 5.5: Duas diferentes células que fazem parte do mesmo caminho da rota da linha 422 e que demonstram a grande variação de precisão do GPS. São exibidos os segmentos de rua interceptas por estas e o conjunto de informações de GPS obtidas ao longo das viagens analisadas

5.3.4 Construção da Rota

Nesta última etapa do processo são inferidas as duas rotas de uma linha de ônibus, que são denominadas por RIF^B e RFI^B. Além disso, são gerados dois **tipos de rota**, que são: (1) Rota Mapeada: um conjunto de segmentos de rua em sequência de utilização pelos ônibus e (2) Rota Geográfica: um conjunto de coordenadas geográficas referentes à Rota Mapeada em sequência de formação. Através da Rota Geográfica, a rota, rua à rua, pode ser reconstruída em qualquer sistema/contexto espacial.

A Rota Mapeada é inferida através da análise das ligações (caminhos) que podem ser utilizadas por um ônibus durante as transições entre as células do grid que pertencem ao caminho do grafo obtido na etapa anterior. Estas ligações fazem parte da topologia construída a partir dos dados do OSM, que representam uma rede de segmentos de ruas em que os ônibus podem trafegar. O algoritmo desta etapa tem como parâmetros de entrada a sequência de visitação das células do grid ($GFCA^B$) e os valores dos erros quadráticos médios de cada segmento interceptado.

A iteração dos dados através do algoritmo ocorre pela seleção de pares de células, que aqui chamaremos de Célula A e Célula B do grid de forma ordenada. Inicialmente, em cada iteração é selecionado o segmento interceptado pela primeira célula do par analisado (Célula A), que apresente o menor valor de erro quadrático médio ou que por ventura seja marcado como o menor devido a uma iteração anterior do algoritmo. Em seguida, a fim de identificar os possíveis caminhos com menores custos de locomoção para a Célula B, é aplicado o algoritmo de Dijkstra direcionado com origem no segmento de A, selecionado anteriormente, e contendo como destino todos os segmentos interceptados pela célula seguinte do par (Célula B). É utilizado este algoritmo pois o mesmo é capaz de identificar o caminho mínimo entre dois pares de vértices de um grafo.

Como apenas um único caminho entre os pares de células (Célula A e Célula B) deve ser escolhido, um processo de avaliação eliminatório é aplicado. Tal processo é composto por duas etapas, sendo uma responsável pela análise do comprimento do caminho e outra por sua geometria. A análise do comprimento do caminho consiste em selecionar o segmento da Célula B responsável por gerar o menor produto entre o comprimento do caminho e o erro quadrático médio atribuído ao segmento. Em seguida, a geometria do caminho gerado é analisado a fim de identificar o número de vezes pelo qual este cruzou a Célula B. Esta ultima avaliação é necessária pois, em algumas regiões da cidade, a qualidade dos dispositivos de GPSs é muito baixa e isto pode induzir o algoritmo a escolher por um segmento de rua com direção totalmente contrária a continuação da trajetória.

Notação	Descrição
GR^B	Conjunto de células do grid regular composto por elementos gr_w^B
$GFCA^B$	Subgrafo de GF^B que contém todos os vértices e arestas obtidos pelo algoritmo de extração do caminho de estratégia gulosa.
$VV[]$	Estrutura de dados auxiliar contendo os vértices ordenados de $GFCA^B$.
$VB[]$	Estrutura de dados auxiliar contendo os segmentos de rua interceptados por uma célula do grid.
$R[]$	Estrutura de dados auxiliar contendo todos os caminhos obtidos pelo algoritmo de construção de rota mapeada.

Tabela 5.6: Tabela de notações utilizadas pelo algoritmo de construção da rota mapeada de uma linha de ônibus

Mais especificamente, caso o caminho cruze e volte para a mesma célula, este será descartado, mesmo que tenha produzido o menor valor na análise do comprimento, e será então selecionado o caminho que gerou o segundo melhor resultado na análise anterior. Caso o segmento identificado na Célula B não seja o responsável por gerar o menor valor do erro quadrático, este será marcado como o obrigatório a ser utilizado como início da próxima iteração. Ao término da avaliação de todos os pares de células do grid contidos no caminho analisado, é obtido o segmento que intercepta todas as células do caminho. Sendo assim, temos a extração da Rota Mapeada da linha analisada. O Algoritmo deste procedimento é apresentado em Algoritmo 6.

Por fim, a Rota Geográfica é extraída a partir da geometria responsável pela representação da Rota Mapeada. A extração consiste na identificação das coordenadas geográficas utilizadas para construção do segmento que representa esta rota.

Algoritmo 6 Algoritmo para construção da rota mapeada de uma linha de ônibus

Precondition: $GFCA^B$ caminho mais utilizado

```
1: function extraiRotaMapeada
2:     existeMenorSegmentoA ← false
3:     menorSegmentoA ← 0
4:     menorSegmentoB ← 0
5:     VV[] ← verticesOrdenados(GFCA^B)
6:     for i = 0; i ≤ |GFCA^B| − 1; i++ do
7:         if !existeMenorSegmentoA then
8:             celulaA ← GR^B(VV[i])
9:             menorSegmentoA ← SegmentoMenorEQM(celulaA)
10:        end if
11:        celulaB ← GR^B(VV[i + 1])
12:        menorSegmentoB ← SegmentoMenorEQM(celulaB)
13:        VB[] ← SegmentosInterceptados(celulaB)
14:        caminhoAtual ← Dijkstra(menorSegmentoA, VB[0])
15:        menorDistancia ← Length(caminhoAtual)
16:        R[i] ← caminhoAtual
17:        for sb = 1; sb ≤ VB.length; sb++ do
18:            caminhoAtual ← Dijkstra(menorSegmentoA, VB[sb])
19:            distanciaAtual ← Length(caminhoAtual)
20:            if QuantidadeInterceptada(caminhoAtual, celulaB) ≥ 0 &&
    QuantidadeInterceptada(caminhoAtual, celulaB) ≤ 2 then
21:                if menorDistancia ≥ distanciaAtual then
22:                    menorDistancia ← distanciaAtual
23:                    R[i] ← caminhoAtual
24:                    if menorSegmentoB == VB[sb] then
25:                        existeMenorSegmentoA ← true
26:                        menorSegmentoA ← menorSegmentoB
27:                    else
28:                        existeMenorSegmentoA ← false
29:                    end if
30:                end if
31:            end if
32:        end for
33:    end for
34:    Return R[]
35: end function
```

Capítulo 6

Resultados

A fim de avaliar a eficiência dos algoritmos propostos neste trabalho, uma série de testes foram realizados. Os testes foram divididos em dois grupos de execução. O primeiro deles tem como objetivo analisar o impacto na escolha de diferentes valores nos parâmetros dos algoritmos sobre uma pequena amostra que contém registros de posição de 3 linhas de ônibus. Já para o segundo grupo é utilizada uma amostra de tamanho significativo, contendo informações de 20 linhas e que são aplicados sobre os algoritmos sem a variação dos parâmetros. Para isso, são utilizados os valores dos parâmetros que obtiveram melhores resultados nos testes do primeiro grupo.

Esta divisão em grupos foi necessária devido ao elevado tempo computacional gasto para analisar um grande volume de informações e também para ajustar os melhores valores nos parâmetros dos algoritmos para quando forem aplicados sobre grandes volumes de dados.

A validação dos resultados, como já esperado, não pode ser realizado a partir dos dados fornecidos pela Data.Rio, que é a mesma plataforma responsável pelos dados das operações dos ônibus utilizado neste trabalho, pois, até a presente data, os dados responsáveis por validarem os resultados encontram-se desatualizados. Além disso, algumas das informações necessárias não são disponibilizadas como, por exemplo, a garagem de uma linha. Esta dificuldade já era previamente conhecida e é um dos motivadores para elaboração deste trabalho.

Outras fontes de dados foram utilizados para validar os resultados obtidos. Como novas fontes foram utilizados a API *Google Maps Directions*, do Google, e o OSM. Através da API do Google foram obtidas as rotas das linhas de ônibus e dos Pontos Inicias e Finais, visto que, tal API é capaz de fornecer as possíveis rotas existentes entre quaisquer

duas coordenadas geográficas. Tais rotas compreendem por um conjunto de coordenadas geográficas em ordem de visitação e por instruções que devem ser realizadas para deslocar-se entre as duas. De outro lado, o OSM foi responsável pela obtenção das regiões de garagens das linhas.

Os resultados de cada uma das etapas da metodologia são apresentados em suas respectivas subseções a seguir.

6.1 Variando-se parâmetros sobre pequena amostra de dados

Neste grupo de testes são utilizados dados de posicionamento dos ônibus de 3 diferentes linhas separados por 1, 4 e 20 dias de operação. A aplicação dos algoritmos de nossa metodologia sobre as amostras de dados são separadas pela quantidade de dias de operação analisadas. Mais especificamente, as informações de garagem, ponto inicial, ponto final e rota de uma linha são inferidas a partir de um mesmo conjunto de dias de operação. Por exemplo, a rota de uma linha obtida a partir da análise de 1 dia de operação terá como insumo de seu processamento as regiões de garagem, ponto inicial e final que também foram extraídas a partir da mesma amostra de dados de posicionamento, que foram gerados durante 1 dia.

6.1.1 Garagem

O algoritmo de identificação da garagem de uma linha de ônibus possui dois parâmetros de entrada, que são: (1) a quantidade β de informações de registros de posição que fazem parte de I^B e F^B; (2) o raio, r, das regiões circulares, denotadas por $R(.)$.

Os testes deste algoritmo são divididos em duas etapas, onde inicialmente são analisados os registros em G^B que minimizam a Equação 5.1 para verificar se estes estão contidos ou não nas verdadeiras regiões de garagem. Em seguida, são criadas as regiões circulares das garagens inferidas, com centroides nos registros identificados na etapa anterior. É feita uma análise sobre as dimensões destas regiões a fim de identificar qual valor de raio é capaz de abranger a grande maioria das reais garagens.

Seguindo a estratégia de analisar o impacto na utilização de diferentes valores nos parâmetros de entrada dos algoritmos, são analisados os seguintes valores: β com valores de 6 e 11 registros de posição e r com 50 e 100 metros.

Os resultados da primeira etapa são apresentados em detalhes na Tabela 6.1. Além disso, como resultado visual temos a Figura 6.1, que possibilita a visualização dos resultados sobre um mapa. Podemos observar através desta figura que todas as instâncias encontram-se contidas em suas respectivas garagens. Além disso, por mais que tenham sido realizados 12 testes para cada uma das linhas, temos em alguns casos a não exibição, visualmente, de todos os 12. Isto ocorre quando os resultados obtidos (registros de posição) possuem a mesma coordenada geográfica que outros, ou seja, existe uma sobreposição dos resultados. A fim de facilitar a identificação destes, através da Tabela 6.1, são adotados símbolos iguais nos números dos testes que produziram o mesmo resultado, como por exemplo, o teste de número 1 e de número 2 para a linha 422.

A segunda etapa de testes consiste na avaliação das regiões circulares que representam as garagens inferidas, denotadas por $RG(gar^B)$. Tais regiões são centradas nas instâncias obtidas na fase anterior e possuem os valores de raio de 200 e 400 metros. Estes elevados valores foram escolhidos pois a garagem de uma linha deve comportar uma grande quantidade de ônibus estacionados ao mesmo tempo, implicando assim em uma região com significativa dimensão.

Através destes valores é possível analisar se a região estimada é capaz de conter toda a verdadeira região de garagem. Vale destacar que, caso a região estimada contenha informações além das necessárias para conter a região verdadeira, isto não será um problema para a metodologia, pois apenas serão afetadas as amostras espaço-temporais localizadas próximas a verdadeira região de garagem. Mais especificamente, estas amostras serão apenas identificadas como contidas nas garagens, sem gerar impactos sobre quaisquer outras etapas. Podemos adotar esta região sobressalente como uma região de segurança para identificação das amostras contidas nas garagens, uma vez que a precisão do dispositivo GPS utilizado pelos ônibus é baixa. Portanto, cada uma das 12 instâncias obtidas na etapa anterior dão origem a duas regiões circulares com seus respectivos valores de raio. A Figura 6.2 apresenta o resultado final das garagens para as 3 linhas.

Através destes resultados, podemos concluir que as regiões de garagem devem ter o valor de seu raio superiores a 200 metros para garantir a cobertura da grande maioria das garagens estudadas. Além disso, temos que a variação dos parâmetros não produzem efeitos significativos nos resultados finais. Podemos destacar o fato de que a quantidade de dias analisados, neste algoritmo, não geraram nenhum tipo de impacto. Sendo assim, a análise de apenas um único dia de operação é capaz de gerar o mesmo resultado de quando são analisados 20 dias.

Número do Teste	Linha	Quantidade de Dias de Operação	β	r	Contida na Garagem Verdadeira
1 γ	422	1	5	50	SIM
2 γ				100	SIM
3 κ			10	50	SIM
4 κ				100	SIM
5		5	5	50	SIM
6				100	SIM
7			10	50	SIM
8				100	SIM
9		20	5	50	SIM
10				100	SIM
11			10	50	SIM
12				100	SIM
1	908	1	5	50	SIM
2				100	SIM
3			10	50	SIM
4				100	SIM
5		5	5	50	SIM
6				100	SIM
7			10	50	SIM
8				100	SIM
9		20	5	50	SIM
10				100	SIM
11			10	50	SIM
12				100	SIM
1 γ	864	1	5	50	SIM
2 γ				100	SIM
3 γ			10	50	SIM
4 γ				100	SIM
5 κ		5	5	50	SIM
6 κ				100	SIM
7 θ			10	50	SIM
8 θ				100	SIM
9		20	5	50	SIM
10				100	SIM
11 δ			10	50	SIM
12 δ				100	SIM

Tabela 6.1: Resultados parciais das garagens das linhas 422, 908 e 864, variando-se a quantidade de dias de operação analisados, raio da região de análise do comportamento e quantidade de instâncias que fazem parte do comportamento das primeiras e últimas posições de um ônibus em um dia.

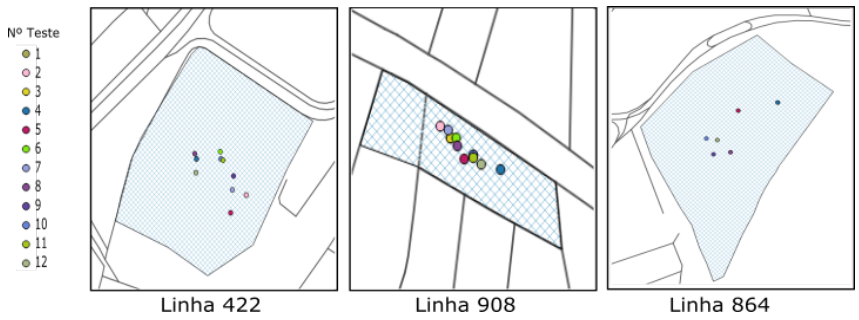


Figura 6.1: Resultado parcial da identificação dos 12 testes para identificação das 3 garagens, referentes as linhas 422, 864 e 908. Cada teste que identifica uma instância é identificado através de uma cor no mapa.

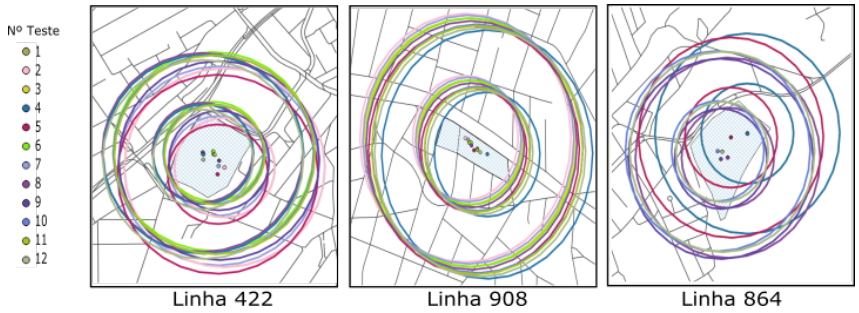


Figura 6.2: Resultado final do processo de identificação das regiões de garagens das linhas 422, 908 e 864.

6.1.2 Pontos Inicias e Finais

O algoritmo desta etapa possui o maior número de parâmetros de entrada em nossa metodologia. Mais especificamente, temos os seguintes parâmetros: (1) intervalo do primeiro horário de *rush*, (2) intervalo do último horário de *rush*, (3) raio (r) da região estudada ($R(.)$) e (4) α, que representa a condição dos valores mínimo e máximo de $SL^B(.)$.

Os valores para os parâmetros de início e término dos horários de *rush* foram identificados a partir de um estudo do comportamento dos ônibus sobre regiões de ponto inicial e final previamente conhecidas. Foram analisados os momentos em que os veículos permaneciam nestas regiões, gerando uma identificação no padrão dos resultados. Percebemos que durante dois intervalos distintos do dia, que ocorriam nas primeiras horas e nas últimas horas, a concentração de veículos nestas regiões era elevada. A Figura 6.3 apresenta um exemplo da ocorrência deste padrão. Neste sentido, para execução dos testes desta etapa serão adotados os intervalos de tempo de 6 às 8 horas e de 20 às 22 horas, respectivamente.

Outro importante parâmetro deste algoritmo é a condição α, que tem como objetivo selecionar para aplicação do algoritmo apenas os ônibus que fiquem parados por alguns

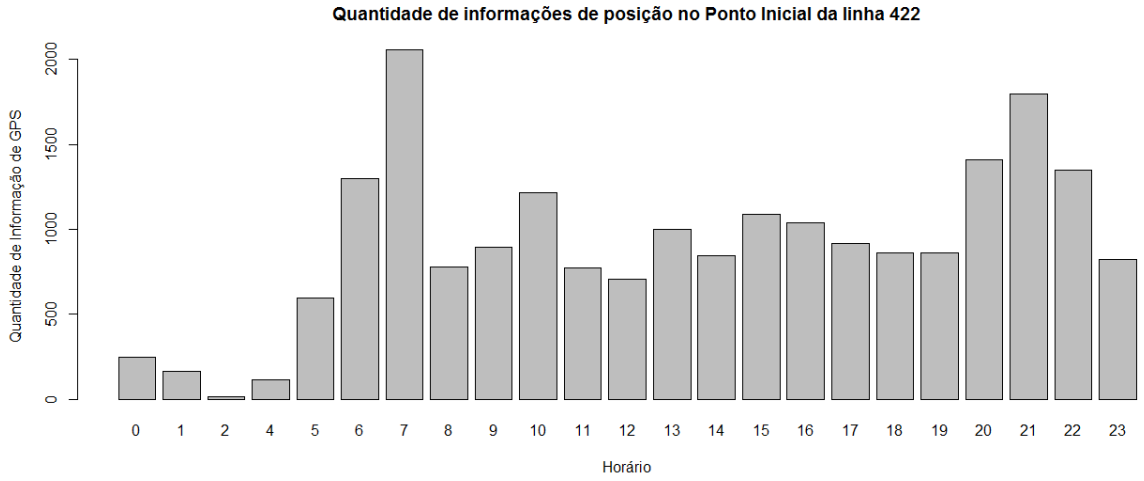


Figura 6.3: Quantidade de informações de posicionamento dos ônibus da linha 422 na região de Ponto Inicial durante às 24 horas de todos os dias de Fevereiro de 2016.

minutos em uma mesma região e eliminar aqueles que permanecem por longos períodos como, por exemplo, ônibus que tenham problemas mecânicos ou mal funcionamento de seu dispositivo de GPS. Para avaliar esta condição, foram realizados testes com valores mínimos de pontuação variando de 1 até 5 e valor máximo fixo em 10. Percebeu-se que os resultados melhoravam na mesma proporção que o valor mínimo era incrementado. Neste sentido, definiu-se que serão utilizados em todos os demais testes os valores de mínimo 5 e máximo 10 para a condição α.

De forma análoga aos testes realizados no algoritmo da garagem, os testes de ponto inicial e final serão divididos em duas etapas. A primeira delas visa identificar se o registro de PIF_d^B que minimiza a Equação 5.2 está de fato contido na região de ponto inicial verdadeira e se o registro de $\overline{PIF_d^B}$ que minimiza a Equação 5.3 está contido na região de ponto final. Em seguida, na segunda etapa é discutido o melhor valor para o raio que representa a região de ponto inicial e final inferida.

Para a primeira etapa são variados os parâmetros de quantidade de dias de operação em 1, 4 e 20 dias e o raio, r, da região estudada em 50 e 100 metros. Os resultados desta fase são apresentados na Tabela 6.2. Podemos observar que a quantidade de dias de operação analisados pelo algoritmo não produz diferença no resultado final. Além disso, o tamanho da região de análise do comportamento também não interfere nos demais resultado.

Para a segunda etapa são analisadas regiões circulares de raio 100, 200 e 400 metros centradas nas instâncias obtidas no teste anterior. Estes diferentes valores de raio foram

Número do Teste	Linha	Região Operacional	Quantidade de Dias de Operação	r	Contida na Verdadeira Região?
1	422	Ponto Inicial	1	50	SIM
2				100	SIM
3			4	50	SIM
4				100	SIM
5			20	50	SIM
6				100	SIM
7		Ponto Final	1	50	SIM
2				100	SIM
3			4	50	SIM
4				100	SIM
5			20	50	SIM
6				100	SIM
1	864	Ponto Inicial	1	50	SIM
2				100	SIM
3			4	50	SIM
4				100	SIM
5			20	50	SIM
6				100	SIM
7		Ponto Final	1	50	SIM
2				100	SIM
3			4	50	SIM
4				100	SIM
5			20	50	SIM
6				100	SIM
1	908	Ponto Inicial	1	50	SIM
2				100	SIM
3			4	50	SIM
4				100	SIM
5			20	50	SIM
6				100	SIM
7		Ponto Final	1	50	SIM
2				100	SIM
3			4	50	SIM
4				100	SIM
5			20	50	SIM
6				100	SIM

Tabela 6.2: Resultados parciais dos Pontos Inicias e Pontos Finais das linhas 422, 908 e 864, variando-se a quantidade de dias de operação analisados e raio da região de análise do comportamento.

testados devido à grande variedade na característica das regiões deste tipo, que podem ser localizadas sobre praças, terminais ou pontos de parada simples, que são similares aos utilizados para atender os usuários ao longo das rotas. A Figura 6.4 contém os resultados para estes valores de raios.

Identificamos que as regiões formadas por 200 metros de raio são capazes de abranger a grande maioria dos Pontos Inicias e Finais. Este valor pode parecer alto, mas regiões deste tipo devem ser capazes de comportar um número significativo de ônibus de diferentes linhas ao mesmo tempo. Além disso, vale ressaltar que este valor é menor do que o utilizado para representar as garagens dos ônibus.

6.1.3 Rota

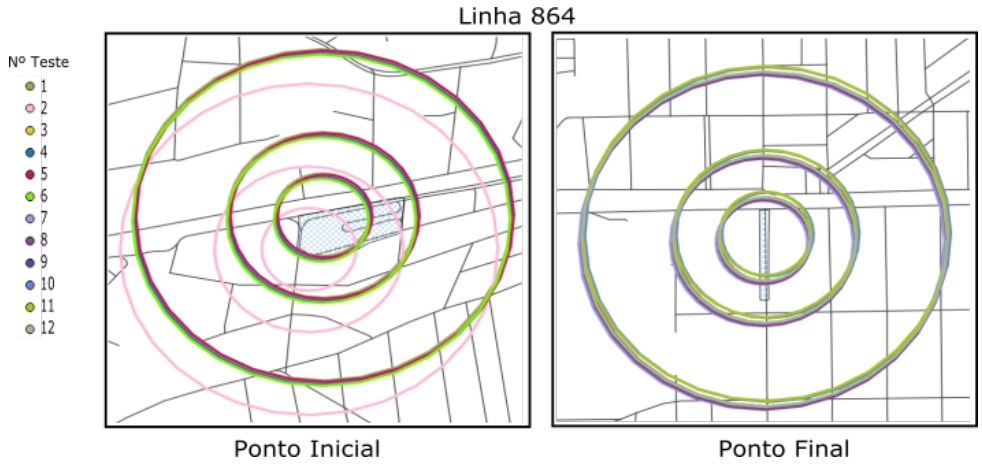
As estimativas das rotas foram avaliadas através de duas métricas. A primeira delas tem como objetivo analisar a efetividade da estratégia de divisão do espaço utilizado durante as viagens em células de um grid. Já a segunda visa avaliar o índice de acerto na escolha dos segmentos de rua que compõem os trajetos entre nós do grafo de transições descrito na Seção 5.

Além da quantidade de dias de operação analisados, temos as dimensões das células do grid como parâmetro de entrada deste algoritmo. São utilizados como valores deste parâmetro 50 e 100 metros quadrados. Os resultados destas duas métricas, com a variação deste parâmetro, são apresentadas a seguir.

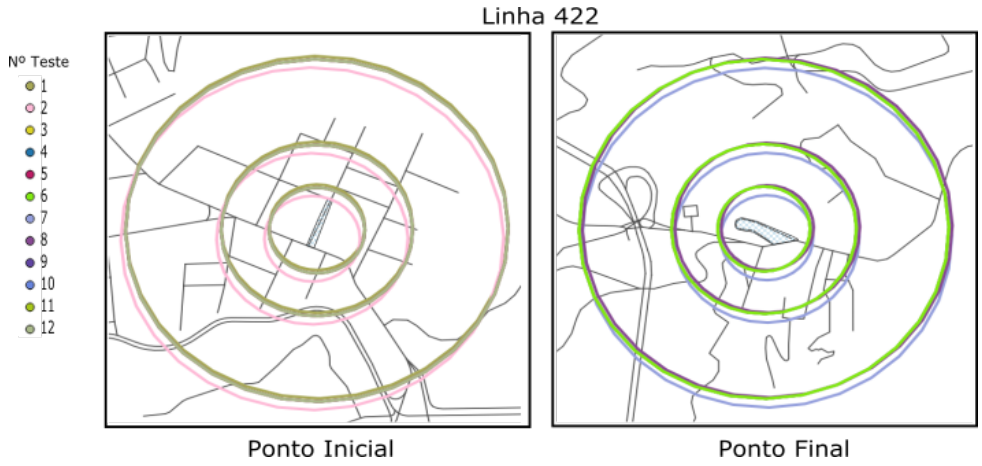
6.1.3.1 Divisão em células

Durante esta etapa é possível verificar se a estratégia de divisão das viagens dos ônibus em pequenas regiões similares de utilização (células de um grid) é de fato uma boa estratégia para resolução deste tipo de problema, pois são contabilizadas as células inferidas que são interceptadas pela rota verdadeira.

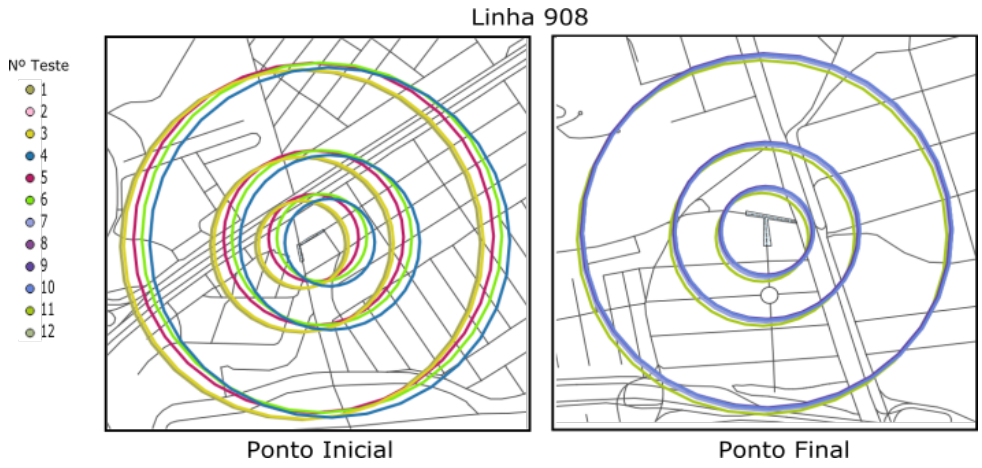
Os resultados obtidos, variando-se os parâmetros de quantidade de dias de operação e a dimensão das células do grid, são detalhados na Tabela 6.3. Podemos observar através dos resultados que, em média, 97% das células são interceptadas pelas rotas. As células que não são interceptadas, em sua grande maioria, estão localizadas fisicamente próximas da rota e somente não são interceptadas devido à imprecisão do GPS, como pode ser observado na Figura 6.6. Como exemplo de um resultado "visual" sobre um mapa, temos a Figura 6.5 que ilustra o resultado da linha 864, para a rota 1, analisando-se apenas 1



(a)



(b)



(c)

Figura 6.4: Resultado final das regiões de Ponto Inicial e Ponto Final das linhas 864, 422 e 908 contendo como raio os valores 100,200 e 400 metros

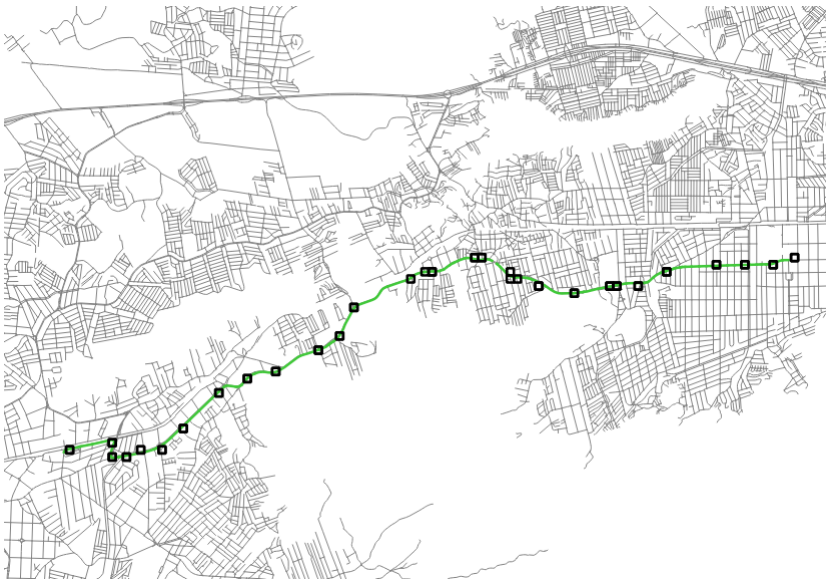
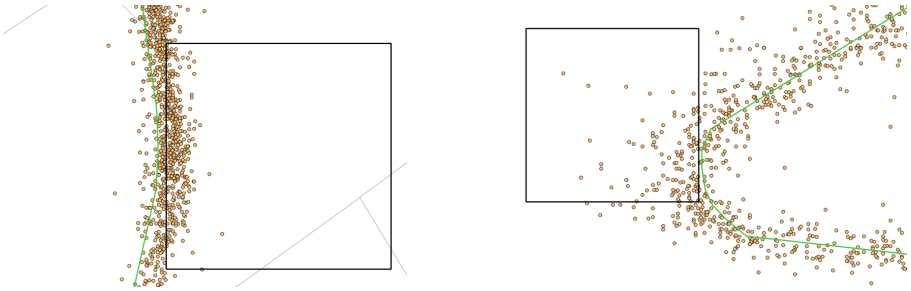


Figura 6.5: Análise das células do grid com dimensões de 100 metros quadrados pertencentes a rota 1 da linha 864 através da análise de 1 dia de operação, que são interceptadas pela rota original representada por um conjunto de segmentos de rua de cores verdes.

dia de histórico de posicionamento e células de dimensões de 100 metros quadrados.



(a) Célula da linha 422 (b) Célula da linha 908

Figura 6.6: Exemplos de células (representadas por quadrados de bordas pretas) contendo as informações de posicionamento (círculos amarelos), que não são interceptadas pela rota original, que é representada por um conjunto de segmentos de rua de cor verde, devido a baixa precisão dos dispositivos GPS.

O fato de uma célula não ser interceptada pela rota original **pode** influenciar diretamente no resultado final da extração da rota. Mais especificamente, o resultado final somente será influenciado de forma negativa quando existirem células que não foram interceptadas pela rota, mas que interceptam outros segmentos de ruas que não fazem parte da rota original. Sendo assim, a rota inferida terá em sua composição segmentos de rua que não fazem parte da rota original. Porém, caso estas células não interceptem nenhum segmento de rua, não haverá impactos sobre o resultado final, pois tais células não serão analisadas durante a próxima etapa do algoritmo.

Podemos ressaltar que a menor taxa de acerto dos resultados (95%) foi obtida devido

Linha	Rota	Quantidade de Dias	Dimensão Célula	Quantidade de Células	Quantidade Interceptada	Taxa de Acerto
864	RIF	1	50	28	28	100%
			100	29	29	100%
		4	50	28	28	100%
			100	28	28	100%
		20	50	29	29	100%
			100	28	28	100%
	RFI	1	50	24	24	100%
			100	24	24	100%
		4	50	24	24	100%
			100	23	23	100%
		20	50	23	23	100%
			100	23	23	100%
422	RIF	1	50	91	88	97%
			100	93	92	99%
		4	50	91	89	98%
			100	87	86	99%
		20	50	94	93	99%
			100	87	86	99%
	RFI	1	50	95	90	95%
			100	86	85	99%
		4	50	95	94	99%
			100	84	84	100%
		20	50	89	88	99%
			100	77	76	99%
908	RIF	1	50	75	74	99%
			100	70	70	100%
		4	50	75	74	99%
			100	65	65	100%
		20	50	67	66	99%
			100	64	64	100%
	RFI	1	50	67	66	99%
			100	62	62	100%
		4	50	64	64	100%
			100	62	62	100%
		20	50	62	61	99%
			100	57	57	100%

Tabela 6.3: Resultados da avaliação do número de células interceptadas pelas rotas originais das linhas 422, 908 e 864, variando-se a quantidade de dias de operação analisados e a dimensão das células do grid.

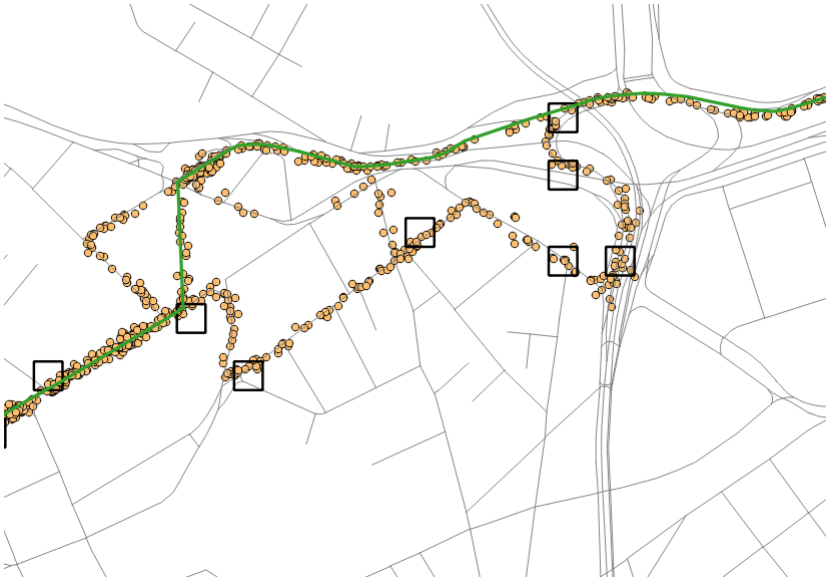


Figura 6.7: Resultado da criação das células do grid da linha 422 analisando-se 1 dia de operação. Podemos observar a existência de informações de posicionamento (círculos amarelos) que não seguem os segmentos da rota verdadeira, que são representados através da cor verde. Tais posições são frutos de desvios realizados por diferentes veículos ao longo do dia.

a um desvio de rota realizado por alguns ônibus da linha 422. Como apenas 1 dia de operação foi analisado, o impacto destes desvios sobre o resultado final foi significativo. A Figura 6.7 apresenta o momento em que os desvios desta rota ocorreram. Além disso, pode-se também destacar que, quanto menor a dimensão das células, maior é a quantidade das mesmas. Isto ocorre devido a precisão do GPS dos ônibus ser relativamente baixa, fazendo com que as células ditas "pequenas" não consigam comportar todo o erro gerado pela imprecisão do dispositivo, gerando assim uma maior quantidade de células pertencentes a rota dos ônibus. Esta relação também é impactada com o aumento da quantidade de dias de operação analisados.

Por fim, temos a comprovação de que a divisão em regiões similares de utilização dos ônibus é uma boa estratégia para resolução deste tipo de problema, uma vez que praticamente todas as células estudadas são interceptadas por suas respectivas rotas verdadeiras.

6.1.3.2 Trajetos entre nós do grafo de transições

Para que as rotas, rua a rua, das linhas sejam estimadas corretamente por nossa técnica, é preciso que os segmentos de ruas dos trajetos que representam as transições entre células do grid sejam estimadas corretamente. Para avaliar a qualidade da escolha destes trajetos, cada conjunto de segmentos que interligam as células do grid são comparados com os

seus correspondentes da rota verdadeira, a fim de identificar se estes são iguais em sua formação. O resultado final desta avaliação corresponde ao percentual de acerto desta métrica. Temos como exemplo de um resultado desta métrica o seguinte cenário: caso uma das rotas de uma linha contenha 50 células do grid para representar o grafo de sua trajetória, serão necessários 49 conjuntos de segmentos para interligarem todas estas células. Supondo que deste total 48 conjuntos de segmentos são ditos iguais, temos que esta rota apresenta 97% de taxa de acerto em suas ligações.

Ao longo dos testes, foi possível identificar que a incorreta inferência dos segmentos através desta metodologia pode ocorrer devido a quatro fatores, que são: (GPS) qualidade do dispositivo de GPS, (OSM) incorreto mapeamento das ruas pelo OSM, (Menor Caminho) existência de caminhos com menores custos entre as células e (Algoritmo) erro na escolha do segmento pela metodologia. Tais fatores são detalhados a seguir:

GPS As regiões geográficas em que a qualidade dos dispositivos de GPS são extremamente baixas interferem de forma significativa no processo de inferência de um segmento, pois a grande quantidade de informações imprecisas podem tornar a execução do algoritmo em um processo aleatório de escolha de segmentos (exemplo deste cenário na Figura 5.5.A).

OSM Os casos em que o erro é decorrente do incorreto mapeamento das ruas pelo OSM ocorrem em regiões estritamente residencias e em locais onde existam poucas sinalizações de placas informando os caminhos que podem ser utilizados pelos veículos. Esta ausência de informações dificulta o correto mapeamento das ruas pelos colaboradores do OSM.

Menor Caminho Nessa metodologia é escolhido o caminho com menor custo de deslocamento entre as células grid. Porém, em alguns casos, temos a existência de outros caminhos que possuem menores custos e que não são utilizados pelos ônibus, mas que, de forma incorreta, são escolhidos como os verdadeiros por esse método.

Algoritmo A escolha incorreta do segmento que intercepta as células do grid pode ocorrer devido à existência de outros segmentos, interceptados pela mesma célula, que ao serem prolongados no processo de projeção ortogonal dos pontos contidos na célula apresentam menores valores no erro médio quadrático.

A aplicação desta métrica sobre a variação dos parâmetros de dias de operações e dimensões das células do grid, em conjunto com a identificação dos erros de identificação

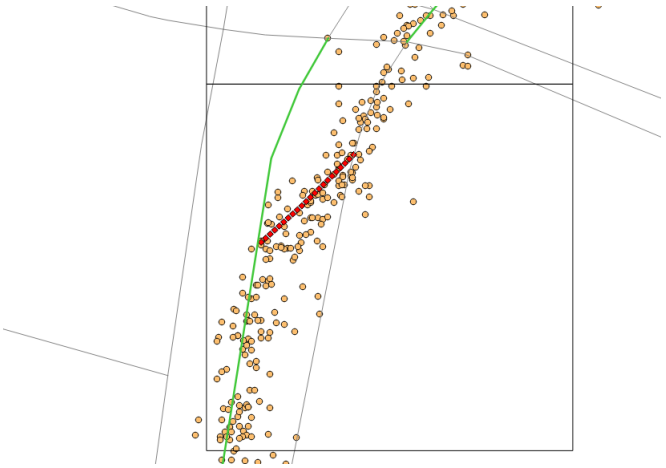


Figura 6.8: Exemplo da ausência do mapeamento de um segmento de rua no OpenStreetMaps. A reta em vermelho representa o segmento que deveria existir no mapa.

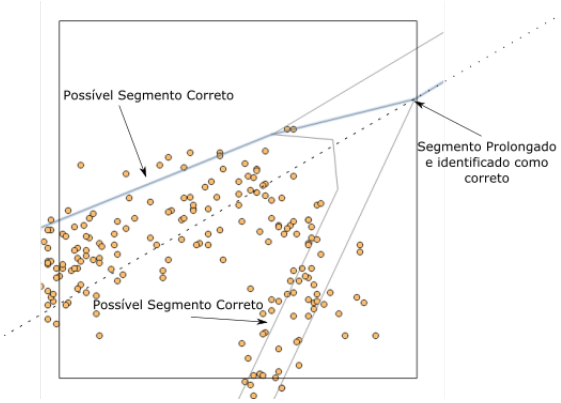


Figura 6.9: Exemplo da escolha do segmento incorreto devido à baixa precisão do GPS e da existência de um segmento que ao ser prolongado é identificado como o mais provável de ter sido utilizado.

dos segmentos, podem ser observados através da Tabela 6.4. Os erros produzidos pelo incorreto mapeamento do OSM representam baixos números totalizando 0,8% do total de interligações entre células analisadas e é responsável por 12,7% das ligações incorretas. Já os erros do tipo Algoritmo são responsáveis por 0,7% das ligações incorretas. A Figura 6.9 apresenta um resultado visual de um erro gerado por este fator.

Devemos destacar que o resultado sobre a rota 2 da linha 422 gerada a partir de apenas 1 dia de operação apresenta 6 ligações incorretas que não são produzidas por nenhum dos fatores de erros apresentados anterior, pois estas tiverem sua origem a partir de desvios da rota utilizadas pelos ônibus no dia estudado. Este resultado já havia sido mencionado na métrica anterior que visa contabilizar as células do grid interceptadas pela rota real.

Linha	Rota	Quantidade de Dias	Dimensão Célula	Ligações Corretas	Ligações Incorretas					Taxa de Acerto
					OSM	Menor Caminho	Algoritmo	GPS	TOTAL	
864	RIF	1	50	27	0	0	0	0	0	100%
			100	28	0	0	0	0	0	100%
		4	50	27	0	0	0	0	0	100%
			100	27	0	0	0	0	0	100%
		20	50	28	0	0	0	0	0	100%
			100	27	0	0	0	0	0	100%
	RFI	1	50	23	0	0	0	0	0	100%
			100	23	0	0	0	0	0	100%
		4	50	23	0	0	0	0	0	100%
			100	21	0	1	0	0	1	96%
		20	50	22	0	0	0	0	0	100%
			100	22	0	0	0	0	0	100%
422	RIF	1	50	88	1	0	1	0	2	98%
			100	88	2	0	2	0	4	96%
		4	50	87	1	0	1	1	3	97%
			100	82	2	0	1	1	4	95%
		20	50	90	1	0	1	1	3	97%
			100	82	2	0	1	1	4	95%
	RFI	1	50	88	0	0	0	0	6	94%
			100	83	1	0	1	0	2	98%
		4	50	94	0	0	0	0	0	100%
			100	80	1	1	1	0	3	96%
		20	50	87	0	0	0	1	1	99%
			100	74	1	1	0	0	2	97%
908	RIF	1	50	71	0	1	2	0	3	96%
			100	67	0	1	1	0	2	97%
		4	50	71	0	1	1	1	3	96%
			100	62	0	1	1	0	2	97%
		20	50	62	0	1	1	2	4	94%
			100	62	0	1	0	0	1	98%
	RFI	1	50	63	1	1	0	1	3	95%
			100	57	1	2	0	1	4	93%
		4	50	61	1	0	0	1	2	97%
			100	57	1	1	1	1	4	93%
		20	50	57	1	1	1	1	4	93%
			100	53	1	1	0	1	3	95%

Tabela 6.4: Resultados final de inferência das rotas das linha 422, 908 e 864, variando-se a quantidade de dias de operação analisados e a dimensão das células do grid.

6.2 Melhores valores dos parâmetros sobre grande amostra de dados

Os testes realizados neste grupo servem para analisar os resultados da aplicação da metodologia sobre uma significativa amostra de dados utilizando-se como parâmetro de entrada dos algoritmos os valores que obtiveram os melhores resultados no grupo anterior. Para isso, foram selecionados, aleatoriamente, dados de operações de 20 linhas de ônibus.

Destas informações, serão analisados apenas 4 dias de operações, pois os resultados do grupo anterior demonstraram que a quantidade de informações geradas durantes 4 dias é o mínimo suficiente para obtenção de bons resultados para a aplicação de todos os algoritmos.

6.2.1 Garagem

Seguindo a mesma avaliação do grupo anterior, os testes deste algoritmo foram divididos em duas etapas. Foram utilizados os valores 100 metros para o parâmetro r e para o parâmetro α foram utilizados 11 registros de posição.

A primeira etapa tem como objetivo identificar se as coordenadas geográficas dos centroides das regiões de garagem inferidas pelo algoritmo estão contidas nas garagens reais das empresas de ônibus. Para todas as 20 linhas testadas o algoritmo estimou corretamente as posições que estavam contidas nas garagens.

A segunda etapa de análise consiste em estudar se o melhor valor de raio para representar as regiões de garagens inferidas no grupo anterior é também capaz de conter todas as linhas da amostra deste grupo de testes. Para isso, foi utilizado o valor do raio como 300 metros. Com a aplicação deste valor para as 20 linhas da base de teste, todas as regiões foram capazes de abrangerem as reais regiões.

6.2.2 Pontos Inicias e Finais

Os testes realizados para avaliar a qualidade da estimativa dos pontos inicial e final das 20 linhas da base de teste foram análogos aos realizados para a estimativa das garagens. Como mencionado no grupo de testes anterior a estimativa dos pontos inicial e final de uma linha foi feita com base em regiões com pontuações entre 5 e 10. Estes parâmetros foram escolhidos experimentalmente com base na qualidade do resultado obtido. De fato, usando esses valores, todas os 40 pontos inicial/final das 20 linhas de teste foram estimadas

Tipo de Erro	GPS	OSM	Menor Caminho	Algoritmo
Número de ocorrências	20	8	23	35
Percentual	23%	9%	27%	41%

Tabela 6.5: Quantidade e percentual dos tipos de erros gerados pela inferência das conexões entre as células das 40 rotas.

corretamente e estão totalmente contidas em uma região com raio de 200 metros.

6.2.3 Rotas

Neste grupo de testes, os resultados das rotas são avaliadas sobre duas métricas. Para isso, utilizou-se a mesma métrica do grupo anterior que visa avaliar o índice de acerto na escolha dos segmentos de rua que compõem os trajetos entre nós do grafo de transições. Porém, temos também a métrica que é responsável por medir o erro entre a rota final inferida e a rota real. Esta última métrica foi adotada neste grupo pois sua avaliação pode ser melhor detalhada quando aplicada sobre uma grande quantidade de amostras de dados.

Como parâmetro da dimensão das células do grid será utilizada o melhor valor obtido nos resultados do grupo anterior que refere-se a células de 100 metros quadrados.

6.2.3.1 Trajetos entre nós do grafo de transições

As 40 rotas analisadas (cada linha é composta por duas rotas) geraram 2388 arestas no grafo de transições. Deste total, obtivemos 2302 (96%) conexões corretas e 86 incorretas (4%). A taxa média de acerto de conexões por rota também foi de aproximadamente 96%.

O gráfico de cima na Figura 6.10 apresenta os resultados obtidos para cada rota. Podemos observar que o pior índice de acerto foi de 80% e o maior de 100%. Identificamos que a rota responsável pela menor taxa de acerto apresenta também a menor quantidade de viagens disponíveis para análise. Neste sentido, concluímos que o algoritmo não foi capaz de obter um resultado satisfatório em virtude desta baixa quantidade de informações combinada com a baixa qualidade do GPS.

A análise quantitativa dos tipos de erros ocorridos nas inferências é vista na Tabela 6.5. O tipo de erro mais comum é gerado pelo seleção do segmento de rua com base no erro quadrático médio.

Além deste, devemos destacar a baixa taxa de erros provocados pelo incorreto mapeamento dos segmentos no OSM (taxa de 9% dos erros). Através desta taxa podemos concluir que a estratégia de adoção de uma base de dados cartográficos para guiar os estudos das rotas pode ser considerado uma boa alternativa.

6.2.3.2 Distância entre rota inferida e real

Através desta métrica, podemos avaliar o impacto na escolha dos segmentos que interligam as células na rota inferida. A distância entre as rotas é calculada a partir da distância euclidiana dos pontos da rota inferida, amostrada a cada 20 metros, para a curva que representa a rota real. Mais especificamente, analisamos o resultado da média e o desvio padrão de cada uma das rotas.

Para uma melhor visualização dos resultados sobre as 40 rotas geradas temos a Figura 6.10 que contém dois gráficos, sendo o de cima com os resultados da distância média de cada rota e o de baixo com o desvio padrão de cada um destas.

Podemos perceber que a média das distâncias entre as rotas apresentam resultados extremamente baixos. Isto deve-se ao fato da utilização da base de dados do OSM, pois todos os segmentos identificados corretamente terão o valor de distância igual a zero metros, e pelo alto índice de acerto das rotas entre nós do grafo de transições, apresentados anteriormente.

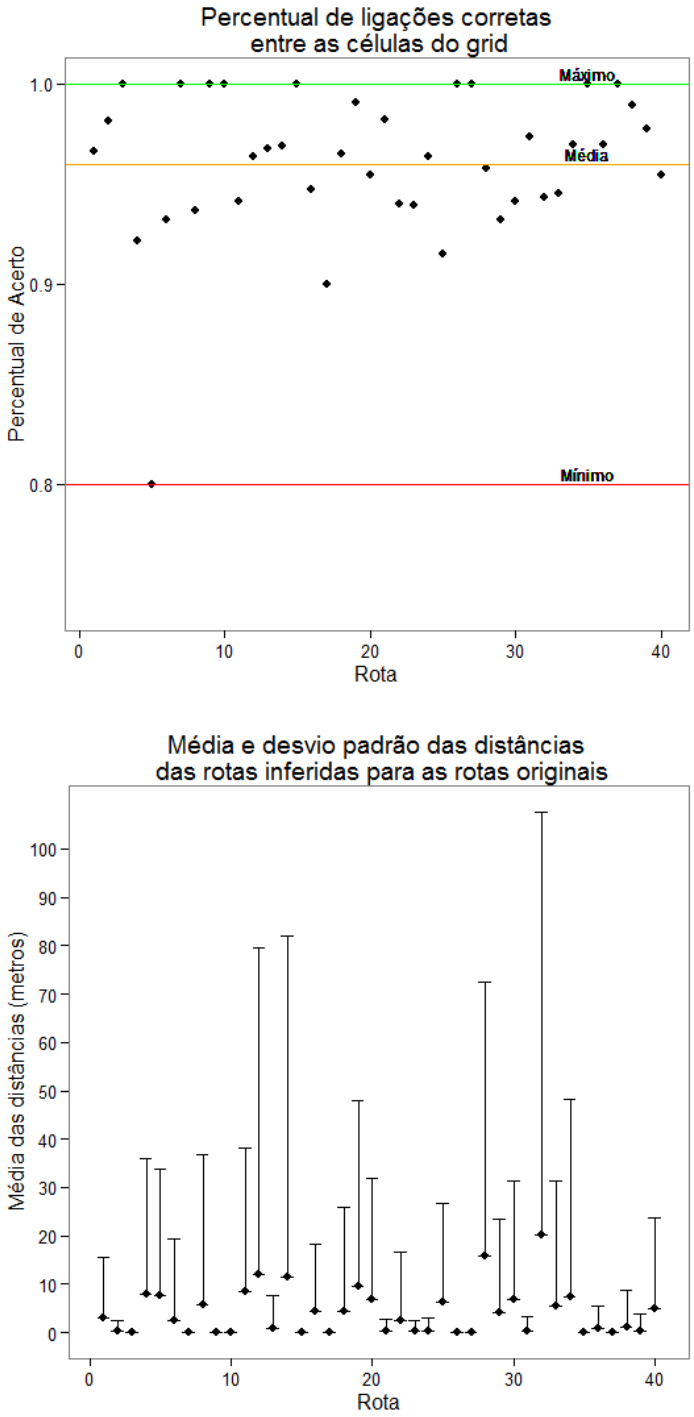


Figura 6.10: Percentual de acerto de trechos de ruas nos trajetos entre células do grid (cima). Erro e desvio padrão da distância entre a rota real e a inferida (baixo).

Capítulo 7

Conclusão

Estudos sobre a mobilidade urbana nas grandes cidades são cada vez mais objeto de pesquisas de diversas áreas de atuação. Como comprovação, temos o crescente número dos Sistemas Inteligentes de Transporte. Podemos destacar o crescimento de uma das categorias deste tipo de sistema denominada Advanced Public Transportations Systems, que analisa exclusivamente dados de transportes públicos de uma cidade. Na mesma proporção do crescimento de tais sistemas, temos a disponibilização, por parte das entidades detentoras, dos dados operacionais dos transportes públicos de uma cidade a população em geral. Aproveitando-se desta tendência, propomos uma metodologia capaz de estimar as regiões operacionais e a rota das linhas de ônibus públicos de uma cidade, a partir de análises sobre os dados históricos de posicionamento dos veículos, sem qualquer tipo de estado da operação, combinados com dados cartográficos da mesma cidade. Para isto, foram adotadas funções de otimização, hipóteses sobre a operação e um algoritmo de Map-Matching para identificação dos segmentos do mapa percorridos pelos ônibus.

Como contribuições deste trabalho temos a criação de algoritmos que são capazes de gerar de forma automática algumas das principais informações operacionais dos ônibus. O algoritmo de identificação da garagem de uma linha é capaz de identificar um dos registros de posição dos ônibus que esteja contido na região de garagem de sua linha partindo-se apenas da análise histórica dos registros de posição durante um único dia de operação. Outro algoritmo proposto por este trabalho refere-se à identificação dos pontos inicias e finais de uma linha. De forma análoga ao da identificação da garagem, este é capaz de identificar um dos registros de posição contido no ponto inicial e outro no ponto final de sua linha, a partir da análise histórica de um dia de operação. Neste sentido, quando houver modificações sobre as regiões de ponto inicial e final de uma linha, este algoritmo pode ser aplicado no dia imediatamente seguinte e gerar corretamente as novas regiões.

O último e mais complexo algoritmo produzido é responsável por obter as rotas, rua a rua, de uma linha. Diferentemente da grande maioria dos algoritmos deste tipo existentes na literatura, este é capaz de obter ótimos resultados mesmo que a taxa de divulgação da base de dados de posição dos veículos seja baixa (Ex.: Aproximadamente 1 minuto).

Os resultados da aplicação destes 3 algoritmos foram satisfatórios. Analisando-se apenas 1 dia de operação podemos obter em 100% dos casos as regiões de garagem, ponto inicial e final de todas as linha estudadas. Já para identificação da rota obtivemos 96% de acerto em uma de nossas métricas que mede a média da distância física da rota inferida para a rota original. Acreditamos que esta taxa poderia ser ainda melhor caso a base de dados de operação dos ônibus estudada não tivesse um significativo número de inconsistências. Dentre estes, podemos ressaltar a existência de ônibus (veículos) associados incorretamente a linhas e a baixa precisão dos dispositivos de GPSs em algumas regiões da cidade do Rio de Janeiro.

As regiões operacionais estimadas pela aplicação de nossa metodologia podem servir de insumo para novos trabalhos. Mais especificamente, uma pesquisa pode aproveitar a identificação das viagens e elaborar indicadores gerencias da linha, tais como: duração média da viagem, quantidade de viagens por dia e tempo médio de partida e chegada dos ônibus nos seus Pontos Inicias e Finais. Outros trabalhos podem verificar, até mesmo em tempo real, se os ônibus estão de fato operando sobre as regiões de sua linha. Por exemplo, os veículos podem ser monitorados afim de possibilitar a identificação de ocorrências de desvios sobre rotas ou não atendimento dos usuários nos pontos inicias ou finais durante algum momento do dia. Por fim, utilizando-se da malha viária, outra extensão deste trabalho pode buscar relações entre as regiões (Ex.: bairros ou zonas) da cidade com a quantidade de rotas existentes em cada uma.

Referências

[1] CleverDevices. `http://www.cleverdevices.com`. Accessed: 2016-01-01.

[2] Data.Rio portal de dados abertos da prefeitura do rio. `http://www.data.rio`. Accessed: 2016-01-01.

[3] Json. `http://www.json.org/`. Accessed: 2016-01-01.

[4] OSM open street maps stats. `http://wiki.openstreetmap.org/wiki/Stats`. Accessed: 2016-01-01.

[5] PostgreSQL. `http://www.postgresql.org/`. Accessed: 2016-01-01.

[6] Python. `https://www.python.org/`. Accessed: 2016-01-01.

[7] Bernstein, D.; Kornhauser, A. An introduction to map matching for personal navigation assistants. *Technical report, New Jersey TIDE Center Technical Report* (1998).

[8] Biagioni, J.; Gerlich, T.; Merrifield, T.; Eriksson, J. Easytracker: automatic transit tracking, mapping, and arrival time prediction using smartphones. In *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems* (2011), ACM, pp. 68–81.

[9] Borges, K. A.; Davis, C. A.; Laender, A. H. Omt-g: an object-oriented data model for geographic applications. *GeoInformatica 5*, 3 (2001), 221–260.

[10] Chen, F.; Shen, M.; Tang, Y. Local path searching based map matching algorithm for floating car data. *Procedia Environmental Sciences 10* (2011), 576–582.

[11] Davies, J. J.; Beresford, A. R.; Hopper, A. Scalable, distributed, real-time map generation. *IEEE Pervasive Computing 5*, 4 (2006), 47–54.

[12] Drane, C. R.; Rizos, C. *Positioning systems in intelligent transportation systems.* Artech House, Inc., 1998.

[13] Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X., et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd* (1996), vol. 96, pp. 226–231.

[14] Figueiredo, L.; Jesus, I.; Machado, J. T.; Ferreira, J.; De Carvalho, J. M. Towards the development of intelligent transportation systems. In *Intelligent Transportation Systems* (2001), vol. 88, pp. 1206–1211.

[15] Gasparini, L.; Bouillet, E.; Calabrese, F.; Verscheure, O.; O'Brien, B.; O'Donnell, M. System and analytics for continuously assessing transport systems from sparse and noisy observations: Case study in dublin. In *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on* (2011), IEEE, pp. 1827–1832.

[16] Ghosh, S.; Lee, T.; Lee, T. S. *Intelligent transportation systems: new principles and architectures*. CRC Press, 2002.

[17] Nwagboso, C. O. *Advanced Vehicle and Infrastructure Systems. Computer Application, Control and Automation. Chapter 4. Intelligent Vehicle Systems and Control*. 1997.

[18] Obradovic, D.; Lenz, H.; Schupfner, M. Fusion of map and sensor data in a modern car navigation system. *Journal of VLSI signal processing systems for signal, image and video technology 45*, 1-2 (2006), 111–122.

[19] Ochieng, W. Y.; Quddus, M. A.; Noland, R. B. Map-matching in complex urban road networks. *Brazilian Journal of Cartography 55*, 2 (2004), 1–18.

[20] Pinelli, F.; Calabrese, F.; Bouillet, E. A methodology for denoising and generating bus infrastructure data. *Intelligent Transportation Systems, IEEE Transactions on 16*, 2 (2015), 1042–1047.

[21] Pinelli, F.; Calabrese, F.; Bouillet, E. P. Robust bus-stop identification and denoising methodology. In *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on* (2013), IEEE, pp. 2298–2303.

[22] Quddus, M. A.; Ochieng, W. Y.; Noland, R. B. Current map-matching algorithms for transport applications: State-of-the art and future research directions. *Transportation Research Part C: Emerging Technologies 15*, 5 (2007), 312–328.

[23] Shibata, J.; French, R. L. A comparison of intelligent transportation systems progress in the united states, europe and japan. In *International Symposium on Automotive Technology & Automation (31st). Logistics management and environmental aspects... intelligent transportation systems and telemetrics... marketing, vehicle finance and leasing* (1998).

[24] Stenneth, L.; Philip, S. Y. Monitoring and mining gps traces in transit space. In *SDM* (2013), SIAM, pp. 359–368.

[25] Sun, D.; Luo, H.; Fu, L.; Liu, W.; Liao, X.; Zhao, M. Predicting bus arrival time on the basis of global positioning system data. *Transportation Research Record: Journal of the Transportation Research Board*, 2034 (2007), 62–72.

[26] United Nations. Un 2012 world urbanization prospects: The 2011 revision highlights. http://goo.gl/U5j9Dg, 2014.

[27] Wang, F.; Chen, W.; Wu, F.; Zhao, Y.; Hong, H.; Gu, T.; Wang, L.; Liang, R.; Bao, H. A visual reasoning approach for data-driven transport assessment on urban roads. In *Visual Analytics Science and Technology (VAST), 2014 IEEE Conference on* (2014), IEEE, pp. 103–112.

[28] White, C. E.; Bernstein, D.; Kornhauser, A. L. Some map matching algorithms for personal navigation assistants. *Transportation research part c: emerging technologies 8*, 1 (2000), 91–108.

[29] Yu, M. *Improved positioning of land vehicle in ITS using digital map and other accessory information.* Tese de Doutorado, The Hong Kong Polytechnic University, 2006.