UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

Ricardo Maureci Ferreira

PRÉ-PROCESSAMENTO DE DADOS DE TRAJETÓRIAS PARA MINERAÇÃO DE DADOS E ANÁLISE DE SIMILARIDADE

Florianópolis

2017

Ricardo Maureci Ferreira

PRÉ-PROCESSAMENTO DE DADOS DE TRAJETÓRIAS PARA MINERAÇÃO DE DADOS E ANÁLISE DE SIMILARIDADE

Trabalho de Conclusão de Curso submetido ao Curso de Sistemas de Informação para a obtenção do grau de Bacharel em Sistemas de Informação.
Orientador: Me. André Salvaro Furtado
Coorientadora: Profa. Dra. Vania Bogorny

Florianópolis

2017

Ricardo Maureci Ferreira

PRÉ-PROCESSAMENTO DE DADOS DE TRAJETÓRIAS PARA MINERAÇÃO DE DADOS E ANÁLISE DE SIMILARIDADE

Este Trabalho de Conclusão de Curso foi julgado aprovado para a obtenção do Título de "Bacharel em Sistemas de Informação", e aprovado em sua forma final pelo Curso de Sistemas de Informação.

Florianópolis, 11 de Dezembro de 2017.

Prof. Dr. Renato Cislaghi
Coordenador

Banca Examinadora:

Me. André Salvaro Furtado
Orientador

Profa. Dra. Vania Bogorny
Coorientadora

Prof. Dr. Luis Otavio Alvares

Prof. Dr. Renato Fileto

RESUMO

O crescimento do uso de dispositivos móveis com sensores GPS possibilitou o aumento da coleta e disponibilidade de dados do movimento de indivíduos como um novo tipo de dado, chamado de trajetória. Sobre esse tipo de dado foram propostas diversas técnicas de mineração para encontrar padrões no movimento de indivíduos. Entretanto, vários fatores durante a coleta podem afetar diretamente a estrutura e qualidade dos dados para tarefas de mineração e análise de similaridade, implicando diretamente em seu resultado. Por exemplo: i) dois indivíduos, realizando movimentos idênticos, geram trajetórias distintas quando seus aparelhos GPS estão configurados com diferentes intervalos de registro; ii) interferência no sinal do GPS causada por grandes obstáculos, como prédios e montanhas, geram ruídos e/ou *gaps* ao longo da trajetória; iii) trajetórias coletadas continuamente por um longo período não permitem identificar onde o indivíduo iniciou ou terminou determinado percurso. Por causa desses fatores, ocorridos ao longo da coleta, é necessário realizar um pré-processamento nos dados das trajetórias para garantir sua estrutura e qualidade através da aplicação de técnicas de organização e limpeza de trajetórias brutas, de acordo com critérios relevantes para o domínio de aplicação. Essas técnicas podem compactar a trajetória, reduzindo o número de pontos onde o indivíduo ficou parado, eliminar ruídos ou segmentar e selecionar trechos relevantes da trajetória. Como por exemplo, selecionar trechos de trajetórias entre duas regiões importantes da cidade em uma aplicação para análise de mobilidade. Desta forma este trabalho identifica e reúne métodos para organização, limpeza e pré-processamento de trajetórias brutas, de modo a melhorar a qualidade e estrutura dos dados para permitir a aplicação de diferentes técnicas de mineração e análise de similaridade em trajetórias. Tais métodos serão reunidos em um sistema que permitirá a manipulação de bases de trajetórias brutas. Esse sistema será utilizado para organizar bases públicas de trajetórias brutas, comumente utilizadas na literatura em um estudo de caso. Por fim, técnicas de análise de similaridade em trajetórias serão aplicadas para comparar as bases organizadas e pré-processadas em relação com as bases de trajetórias originais, avaliando a variação do resultado.

Palavras-chave: trajetória. mineração de dados. pré-processamento. análise de similaridade. dados espaço-temporais.

LISTA DE TABELAS

LISTA DE FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

LISTA DE SÍMBOLOS

SUMÁRIO

1 INTRODUÇÃO

O aumento no uso de dispositivos móveis dotados de sensores GPS permite que, cada vez mais, indivíduos tenham seus movimentos registrados. Esses dispositivos podem ser configurados para registrar a localização do indivíduo a cada determinado período de tempo. Por exemplo, um ponto pode ser registrado a cada 10 segundos com a coordenada geográfica da localização de um indivíduo, com a data e horário do momento do registro. Criando assim uma sequência de localizações ao longo do tempo na forma de uma trajetória bruta. Uma trajetória bruta é uma sequência de pontos espaço-temporais $((x_1, y_1), t_1), ((x_2, y_2), t_2), ..., ((x_n, y_n), t_n))$, onde (x_i, y_i) é a i-ésima coordenada espacial, t_i é o instante de tempo associado a essa coordenada e n é o número de pontos da trajetória (BOGORNY; BRAZ, 2012). Trajetórias podem ser geradas a partir de diferentes tipos de objetos móveis, como pessoas, veículos, animais ou até mesmo fenômenos naturais, como furacões. A coleta de trajetórias pode ser realizada de maneira passiva ou ativa (ZHENG; ZHOU, 2011). Uma coleta ativa requer intervenção humana para ligar e desligar o equipamento, como, por exemplo, um atleta que ativa seu monitor de atividades com GPS durante uma corrida. Já a coleta passiva é quando o equipamento com GPS é ativado e desligado automaticamente, por exemplo, um ônibus que ativa o GPS assim que o motor é ligado.

Ao longo da última década, diversos trabalhos foram realizados sobre trajetórias, como a descoberta de padrões de trajetórias de objetos móveis que desviam de objetos estáticos (ALVARES et al., 2011) e a identificação de motoristas perigosos (CARBONI; BOGORNY, 2015). Ainda a partir das trajetórias também é possível analisar a similaridade de dois objetos. Neste contexto, uma medida de similaridade é uma métrica que especifica se dois objetos são semelhantes um ao outro, de acordo com as características de suas trajetórias (ZHENG; ZHOU, 2011). Alguns trabalhos no âmbito da similaridade propõem medidas de similaridade para trajetórias (XIAO et al., 2014; FURTADO et al., 2015). Porém tanto a descoberta e a identificação de padrões, quanto a análise das trajetórias, estão sujeitas a problemas e situações recorrentes no processo de coleta que interferem diretamente na estrutura e na qualidade dos dados.

Uma série de fatores influenciam a coleta de trajetórias, tais como: i) condições climáticas, que podem causar variações na precisão do ponto coletado; ii) a interferência de prédios altos, túneis ou qualquer grande obstáculo pode interferir no sinal do GPS, criando um intervalo na trajetória; e iii) a intervenção deliberada do indivíduo na configuração e utilização do dispositivo usado na coleta.

O nível de precisão do GPS pode ser ajustado e deve ser considerado para cada situação, como, por exemplo, ao registrar o movimento de um navio ao atravessar o oceano, não é necessário coletar pontos a cada 1 segundo, e há uma maior tolerância quanto a precisão do posicionamento. Outro fator é como são gerenciadas as trajetórias durante a coleta: um objeto

pode não iniciar a coleta no momento correto, ou terminar um percurso e o aparelho GPS continuar registrando os dados, ou ainda ficar parado com o GPS registrando e seguir outra viagem. Nesse último caso o GPS possui todos os pontos coletados como única trajetória do objeto e não como duas trajetórias distintas. Também é possível perceber que, conforme os parâmetros estão configurados em diferentes dispositivos, o movimento idêntico de dois objetos pode gerar trajetórias distintas. Por exemplo, o intervalo de registro pode estar configurado em intervalos de tempo diferentes com os pontos da trajetória de um indivíduo sendo coletados a cada 5 segundos e o de outro a cada 10 segundos. Assim, a trajetória do primeiro indivíduo registraria mais pontos que a do segundo, com localizações que não correspondem de forma exata.

Em razão dos fatores externos aos quais a trajetória esteve sujeita durante a coleta, é necessário realizar um pré-processamento para que essa possa ser utilizada em uma análise (FURTADO, 2014). Esse pré-processamento garante a estrutura e a qualidade de dados necessária para o domínio de aplicação. Por esse motivo é importante que o formato dos dados coletados e as informações relativas ao processo de coleta sejam de conhecimento de quem irá realizar o processo de organização e limpeza dos dados. O formato do dado da trajetória geralmente possui, ao menos, a data e horário do momento do registro, a posição geográfica representada por algum sistema de referência geográfica e um identificador do usuário (GIANNOTTI; PEDRESCHI, 2008). Durante o processo de limpeza pode ser realizada a compressão da trajetória, que é a redução da quantidade de dados através da remoção de alguns pontos de modo a preservar o formato geral da trajetória, e a eliminação de pontos discrepantes ou ruídos, i.e., pontos que fogem do padrão da trajetória e são causados, geralmente, por uma interferência no sinal do GPS (ZHENG; ZHOU, 2011). Entretanto, quando o modelo de análise exigir, trechos das trajetórias relevantes para análise podem ser extraídos a fim de tornar mais eficiente o manuseio e processamento desses dados. Já algumas análises necessitam apenas de alguns trechos específicos das trajetórias ou de partes da trajetória entre duas regiões de interesse (FONTES; BOGORNY, 2013; FURTADO, 2014).

Trajetórias que iniciam em determinado local, como origem em uma universidade em direção aos bairros, poderiam prover dados para entender o comportamento ou a concentração dos alunos saindo da universidade. Dessa forma, trajetórias podem ser selecionadas de acordo com regiões pelas quais elas passam. Por exemplo, selecionar as partes das trajetórias que estão entre o aeroporto e o centro financeiro da cidade, ou trajetórias que vão do aeroporto até uma praia específica (FONTES; BOGORNY, 2013). Ou ainda em análises que usam o modelo *Stops and Moves* (SPACCAPIETRA et al., 2008), podem necessitar separar partes das trajetórias que sejam os *Stops* ou os *Moves*.

Outra situação comum ocorre no registro de uma trajetória, na qual geralmente há um identificador do usuário e o identificador de cada trajetória desse usuário. No início de cada nova trajetória, do mesmo usuário, é registrado o identificador deste usuário e um novo identificador para essa trajetória. Entretanto, pode ocorrer que o GPS seja desligado e quando ligado para uma nova coleta continue registrando com o identificador da trajetória anterior. Assim, temos uma trajetória que iniciou em um momento, ficou desligada por um período e continuou, do mesmo ponto em outro momento de tempo. Portanto, a trajetória pode ser quebrada de acordo com a informação do intervalo de tempo de cada registro, geralmente disponível através da documentação dos registros. Em uma trajetória, que foi coletada com um intervalo de registro de 1 segundo, um ponto seguinte com uma diferença do ponto anterior de poucos segundos pode ser devido alguma variação ou interferência no GPS, porém uma diferença de cinco minutos, de um ponto em relação ao ponto anterior, pode ter sido ocasionada pelo desligamento do GPS. Dessa forma, para cada necessidade de aplicar uma técnica de mineração de dados ou análise de similaridade, é então necessário a criação de um procedimento específico para cada base de dados escolhida. Esse procedimento irá extrair as trajetórias no formato desejado e pré-processá-las para permitir a aplicação dessas técnicas.

Neste cenário, este trabalho propõe um sistema do tipo *desktop*, com um conjunto de técnicas de pré-processamento, organização e limpeza de trajetórias disponíveis na literatura. Esse sistema suporta os formatos mais comuns para exportação de dados em aparelhos GPS, como CSV, GPX e JSON. Utilizando o banco de dados PostgreSQL com a extensão espacial PostGIS, que é o banco de dados geográfico gratuito mais completo para processamento de dados espaciais (ZHENG; ZHOU, 2011). Com esse sistema é possível, de maneira simples e eficaz, realizar a organização e limpeza de trajetórias brutas para aplicação de diferentes técnicas de mineração e análise de similaridade.

1.1 OBJETIVOS

Esse trabalho tem como objetivo geral o desenvolvimento de um sistema que reúna métodos para pré-processamento, organização e limpeza de trajetórias brutas, de modo a permitir a aplicação de diferentes técnicas de mineração e análise de similaridade em trajetórias.

1.1.1 OBJETIVOS ESPECÍFICOS

1. Identificar e implementar um conjunto de métodos para pré-processamento, organização e limpeza de trajetórias, de acordo com critérios já utilizados em trabalhos para mineração e análise de similaridade na literatura.

2. Organizar bases de dados de trajetórias comumente utilizadas na literatura, como Geolife (ZHENG et al., 2008) e Taxi San Francisco (PIORKOWSKI; SARAFIJANOVIC-DJUKIC; GROSSGLAUSER, 2009), para aplicação de técnicas de mineração de dados e/ou análise de similaridade.

3. Identificar critérios que garantam uma melhor estrutura e qualidade de dados para mineração e análise de similaridade em dados de trajetórias.

4. Avaliar os resultados das trajetórias pré-processadas usando métodos para a análise de similaridade e comparando com os resultados obtidos com as trajetórias brutas originais.

1.2 ESCOPO DO TRABALHO

O presente trabalho tem como escopo o pré-processamento, organização e limpeza de trajetórias brutas, disponibilizadas como bases de dados públicas. Foram utilizadas somente ferramentas livres e gratuitas. O sistema é do tipo *desktop*, construído em Java e suporta os formatos mais comuns para exportação de dados em aparelhos GPS, como CSV, GPX e JSON, e o banco de dados PostgreSQL, com a extensão espacial PostGIS. Técnicas de análise de similaridade, presentes na literatura, permitiram avaliar as trajetórias pré-processadas em relação as trajetórias brutas.

1.3 MÉTODO DE PESQUISA

Esse trabalho tem como método de desenvolvimento as seguintes etapas:

1. Realizar uma revisão na literatura de técnicas para pré-processamento, organização e limpeza de trajetórias.
2. Definir novas formas para organização e limpeza de trajetórias.
3. Implementar as técnicas levantadas e definidas para organização e limpeza de dados de trajetórias.
4. Projetar, implementar e testar um sistema do tipo *desktop* para pré-processamento de dados de trajetórias com as técnicas implementadas.
5. Organizar bases de dados públicas, comumente utilizadas na literatura, de acordo com as técnicas implementadas na etapa 3 para validar o sistema desenvolvido na etapa 4.
6. Identificar medidas de similaridade, utilizadas na literatura, para comparar trajetórias brutas com trajetórias pré-processadas.
7. Implementar as medidas de similaridade levantadas para avaliar os resultados sobre os dados pré-processados.
8. Fazer um estudo de caso comparando as bases de dados organizadas e pré-processadas, em relação as bases de dados de trajetórias brutas, utilizando as medidas de similaridade implementadas.

2 CONCEITOS BÁSICOS E TECNOLOGIAS

Este capítulo descreve o conceito de trajetória, como é o processo de coleta destas, as técnicas de pré-processamento e limpeza de trajetórias que podem ser aplicadas sobre esses dados para permitir a aplicação de técnicas de mineração de dados. No processo de coleta de trajetórias podem ser usados diversos modelos de aparelhos com sensores GPS, vários formatos de arquivos que permitem a transição dos dados entre os aparelhos e uma base de dados. Também são levantados os problemas comuns no processo de coleta de trajetórias e como esses problemas podem interferir na qualidade dos dados de trajetórias. Por fim são apresentadas algumas técnicas de pré-processamento e limpeza de trajetórias disponíveis na literatura.

2.1 TRAJETÓRIAS DE OBJETOS MÓVEIS

Alguns objetos possuem localização fixa no espaço, por exemplo, prédios e monumentos históricos. Entretanto determinados objetos não permanecem fixos no mesmo local, estes objetos se movem e não permitem que tenham uma localização fixa associada a eles. Por exemplo, um carro parado possui uma localização, mas quando em movimento, sua localização muda ao longo do tempo.

Dessa forma a sequência de localizações registrados para cada objeto móvel é chamada de trajetória. Uma trajetória é representada por um conjunto de pontos ao longo do tempo $((x_1, y_1), t_1), ((x_2, y_2), t_2), ..., ((x_n, y_n), t_n))$, onde (x_i, y_i) é a i-ésima coordenada espacial, t_i é o instante de tempo associado a essa coordenada e n é o número de pontos da trajetória (BOGORNY; BRAZ, 2012).

Essas trajetórias podem ser geradas por qualquer objeto que se mova no espaço e que possua um dispositivo capaz de registrar sua localização. Com a popularização de dispositivos móveis com sensores GPS, como *smartphones*, tornou-se possível a coleta de grandes volumes de dados de trajetórias. Dessa forma um aparelho com sensor GPS junto ao objeto é configurado para gravar dados e consegue, por exemplo, gravar a localização do indivíduo a cada segundo ou um ponto a cada 50 metros em que o objeto se move, assim registrando toda sua trajetória.

2.2 PROCESSO DE COLETA DE TRAJETÓRIAS

O processo de coleta está diretamente ligado a forma como o sensor GPS é gerenciado para obtenção dos dados de localização, podendo ser caracterizada como ativa ou passiva (ZHENG; ZHOU, 2011). Uma coleta ativa ocorre quando é necessária a intervenção humana para ligar e desligar o equipamento, como, por exemplo, um grupo de pessoas que ativam, individualmente, a coleta de suas trajetórias em seus aparelhos com sensores GPS, iniciam uma caminhada e

desativam quando desejarem (SANTOS, 2013; SOCIETY, 2005). Também se caracteriza como uma coleta ativa quando um pecuarista coloca sensores GPS em bovinos para monitorar seu comportamento em relação ao rebanho (FERREIRA, 2013). Por outro lado, a coleta passiva é quando o equipamento com GPS é ativado e desligado automaticamente. Por exemplo, um automóvel ativa o sensor GPS assim que o motor é ligado e coleta sua trajetória durante o período em que o motor estiver ligado (AVANCINI, 2010).

2.2.1 TECNOLOGIAS

O **Sistema de posicionamento global (GPS)** é um sistema que usa tecnologia baseada em satélites, cuja principal técnica é medir os intervalos de tempo entre receptor e alguns satélites simultaneamente observados (XU, 2007). O GPS é utilizado em aplicações de posicionamento e navegação. Através da posição conhecida dos satélites e a distância medida até o receptor, é possível então determinar a posição do receptor. O receptor GPS é disponibilizado como um sensor, que pode ser acoplado em dispositivos móveis, como por exemplo, computador de bordo veicular ou *smartphones*.

Com a redução do custo e melhor eficiência no consumo de bateria, celulares com sensores GPS têm sido utilizados para registro da localização do usuário. Esses dispositivos, através de aplicativos específicos, permitem o gerenciamento do processo de coleta das trajetórias (SANTOS, 2013; FURTADO, 2014). Esses aplicativos permitem iniciar, terminar e registrar a coleta de uma trajetória, bem como também visualizar informações dessas coletas, como número de pontos da trajetória, tempo de coleta, distância percorrida, média de velocidade e etc.
A Figura 1 apresenta a tela de 3 aplicativos gratuitos para registro de dados de trajetórias em celulares. Da esquerda para direita: *My Track*[1], *GPSLogger*[2] e *Cycle GPS Logger*[3]

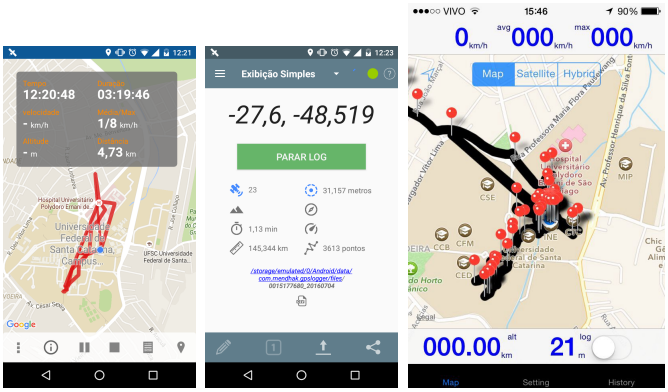

Figura 1 – Tela dos aplicativos *My Track* (esquerda) e *GPSLogger* (centro) para celulares e *tablets* Android; *Cycle GPS Logger* (direita) para celulares Iphone e *tablets* Ipad

[1] www.513gs.com/
[2] http://code.mendhak.com/gpslogger/
[3] www.karikino10.blogspot.com.br/

2.2.2 CONFIGURAÇÃO E ALTERNATIVAS DE COLETA

É durante a coleta que a trajetória é identificada como única. Na coleta ativa é comum a geração de um arquivo para cada trajetória, que geralmente é criado quando o usuário encerra a coleta de dados. A coleta passiva permite mais configurações, de acordo com seu uso. Por exemplo, uma trajetória inicia quando o motor do veículo é ligado e só termina quando este motor for desligado; O aparelho GPS pode ser configurado para gerar um arquivo com uma trajetória a cada dia ou várias trajetórias iniciadas após um determinado período sem movimento; Situação parecida ocorre com táxis e ônibus, que além disso permitem a identificação de acordo com algum estado do percurso, podendo registrar uma trajetória de ônibus quando, por exemplo, está com o itinerário no sentido centro-bairro ou um táxi quando está com passageiro.

2.2.3 FORMATOS DE ARMAZENAMENTO

Os aplicativos nos dispositivos móveis utilizam o sensor GPS do aparelho para coletar as coordenadas geográficas de sua localização. Através dessas coordenadas, e da sequência de coordenadas coletadas, com a variação de posições ao longo do tempo, é possível realizar cálculos para determinar velocidade, distância e outras informações da trajetória. Há duas alternativas para registrar esses dados, que são: i) o dado é enviado para uma base centralizada conforme é coletado, porém envolve o uso de algum tipo de rede, sujeito a falhas e largura de banda. e ii) armazenar os dados no próprio aparelho e realizar a exportação dos dados para uma base centralizada através de arquivos (GIANNOTTI; PEDRESCHI, 2008). A segunda alternativa é a mais utilizada em aplicações GPS e os formatos de dados mais comuns para exportação de dados são: DSV, JSON, KML e GPX.

Valores Separados por Delimitadores (DSV): É um arquivo de texto usado para armazenar dados, em que cada linha representa um registro, com seus valores separados por um delimitador comum. Qualquer caractere pode ser usado como delimitador, porém os mais comuns são: ponto, ponto e vírgula, dois pontos, barra vertical, espaço e vírgula. O cabeçalho desse arquivo, quando incluído, ocupa a primeira linha. Cada linha subsequente é uma linha com dados e as linhas são separadas pelo caractere de nova linha. O formato de arquivo delimitado mais popular é o CSV (*Comma Separated Value*), que utiliza delimitação por vírgula e está especificado na RFC 4180 (SHAFRANOVICH, 2005). Arquivos CSV são geralmente utilizados para transferir dados de um programa proprietário, como um GPS, para uma banco de dados que utiliza outro formato de dado. Assim é possível realizar a exportação dos dados de trajetória para formato CSV, pelo aplicativo que gerencia o GPS e realizar a importação destes dados no arquivo CSV para o banco de dados, através do sistema gerenciador do banco de dados. A Figura 2 ilustra um arquivo CSV com cinco pontos de uma trajetória de um caminhão na Grécia. Nesse

arquivo a coluna *obj-id* é o identificador do caminhão e *traj-id* é a identificação da trajetória do caminhão, nesse caso todos os cinco pontos pertencem a mesma trajetória. As colunas data e hora permitem saber o instante de tempo no qual foram registradas as localizações. As colunas *lon* e *lat* são a longitude e latitude, respectivamente, obtidas no sistema de referência geográfica WGS84[4] e as colunas x e y são a longitude e latitude, respectivamente, no sistema de referência geográfica GGRS87[5].

```
obj-id,traj-id,data,hora,lon,lat,x,y
0862,1,10/09/2002,09:15:59,23.845089,38.018470,486253.80,4207588.10
0862,1,10/09/2002,09:16:29,23.845179,38.018069,486261.60,4207543.60
0862,1,10/09/2002,09:16:59,23.845530,38.018241,486292.40,4207562.60
0862,1,10/09/2002,09:17:29,23.845499,38.017440,486289.60,4207473.80
0862,1,10/09/2002,09:17:59,23.844780,38.015609,486226.10,4207270.70
```

Figura 2 – Arquivo CSV com trajetórias de caminhões na Grécia.

Javascript Object Notation (JSON): É um formato para intercâmbio de dados especificado na RFC 7159. A estrutura deste arquivo segue a notação de objetos Javascript, mas seu uso não requer Javascript exclusivamente (BRAY, 2014). A simplicidade do JSON permite intercâmbio de dados entre diferentes aplicações, visto que JSON é suportado pelas principais linguagens de programação. Um arquivo JSON pode ser construído por dois tipos de estruturas: i) uma coleção pares nome/valor ou ii) uma lista ordenada de valores. Estas estruturas permitem fácil manuseio dos dados durante a manipulação destes, pois facilitam a criação de estruturas de dados, como vetores simples e associativos. A Figura 3 ilustra um trecho de um arquivo JSON exportado do serviço de histórico de localização do Google[6]. Nesse arquivo há uma coleção de localizações por onde o usuário passou. Para cada par de latitude e longitude há um *timestamp* associado. Também é possível perceber dados associados, como a precisão na localização do ponto coletado (*accuracy*), velocidade instantânea (*velocity*) e altitude.

```
{
  "locations" : [ {
    "timestampMs" : "1466978890766",
    "latitudeE7" : -277613766,
    "longitudeE7" : -485111577,
    "accuracy" : 90
  }, {
    "timestampMs" : "1466978828811",
    "latitudeE7" : -277616298,
    "longitudeE7" : -485111353,
    "accuracy" : 42,
  }, {
    "timestampMs" : "1466978691000",
    "latitudeE7" : -277609267,
    "longitudeE7" : -485109943,
    "accuracy" : 12,
    "velocity" : 0,
    "altitude" : 24
  }, {
    "timestampMs" : "1466978676000",
    "latitudeE7" : -277609379,
    "longitudeE7" : -485110047,
    "accuracy" : 12,
    "velocity" : 0,
    "altitude" : 26
  }
}
```

Figura 3 – Arquivo JSON com dados do histórico de localização de um usuário do Google.

[4] www.spatialreference.org/ref/epsg/wgs-84/
[5] www.spatialreference.org/ref/epsg/ggrs87-greek-grid/
[6] www.google.com/maps/timeline

Keyhole Markup Language (KML): É uma notação baseada em XML, criada inicialmente para a ferramenta Google Earth[7]. Esse formato permite expressar anotações geográficas na internet, bem como mapas bidimensionais e tridimensionais do planeta Terra. Um arquivo KML específica um conjunto de características (imagens, polígonos, modelos 3D, descrições textuais, etc.) para exibição em aplicações como Google Earth, Google Maps[8] e OpenStreetMap[9], ou qualquer outro software geoespacial que implemente o padrão KML. Nesse formato cada local no mapa possui uma longitude e latitude. Também é possível associar dados para tornar a informação mais completa, como altitude, inclinação e também o instante de tempo (*timestamp*). A Figura 4 ilustra dados de uma trajetória exportada em formato KML. Esse formato permite especificar uma trajetória com a *tag* <*gx:Track*> e possui *tags* específicas para denotar o par de coordenadas e *timestamp*.

```
<gx:Track>
<when>2016-07-04T20:10:01.730Z</when>
<when>2016-07-04T20:11:01.791Z</when>
<when>2016-07-04T20:12:01.834Z</when>
<gx:coord>-48.518776 -27.600426 -9999.0</gx:coord>
<gx:coord>-48.51884 -27.60039 -9999.0</gx:coord>
<gx:coord>-48.518795 -27.600449 -9999.0</gx:coord>
</gx:Track>
```

Figura 4 – Arquivo KML com uma trajetória exportado pelo aplicativo MyTracks.

Well-Known Text (WKT): É uma linguagem de marcação de texto utilizada para representação de formas geométricas. Esse é um formato padrão legível que pode ser usado para intercâmbio de dados espaciais entre aplicações. O formato foi originalmente definido pela *Open Geospatial Consortium* (OGC) e permite que aplicações possam trocar informações utilizando um modelo de referência geográfica independente da outra aplicação. Como por exemplo o banco de dados PostgreSQL pode trocar informações de pontos de um objeto com o banco de dados Maria DB[10]

```
1  156;2014-02-01 00:00:00.739166+01;POINT(41.8836718276551 12.4877775603346)
2  187;2014-02-01 00:00:01.148457+01;POINT(41.9285433333333 12.4690366666667)
3  297;2014-02-01 00:00:01.220066+01;POINT(41.8910686119733 12.4927045625339)
4  89;2014-02-01 00:00:01.470854+01;POINT(41.7931766914244 12.4321219603157)
5  79;2014-02-01 00:00:01.631136+01;POINT(41.90027472 12.46274618)
```

Figura 5 – Arquivo WKT com pontos de trajetórias de táxis em Roma.

GPS Exchange Format (GPX): É um esquema XML de padrão aberto concebido como formato comum de dados para aplicações GPS. Esse formato permite descrever pontos, trajetórias, rotas e dados opcionais, como elevação, instante de tempo (*timestamp*) e outras informações através de *tags* específicas que possibilitam o intercâmbio de dados entre dispositivos GPS e softwares. A Figura 6 apresenta uma trajetória no formato

[7] www.google.com/earth/
[8] www.google.com/maps/
[9] www.openstreetmap.org
[10] www.mariadb.org

GPX. Esse formato permite especificar uma trajetória através da *tag* <*trk*>, os pontos da trajetória através da *tag* <*trkpt*> e o instante de tempo para esse ponto na *tag* <*time*>.

```
<trk>
<name><![CDATA[Route de 2016-07-04 17:04]]></name>
<desc><![CDATA[]]></desc>
<type><![CDATA[]]></type>
<extensions><topografix:color>c0c0c0</topografix:color></extensions>
<trkseg>
<trkpt lat="-27.600426" lon="-48.518776">
<ele>-9999</ele>
<time>2016-07-04T20:10:01.730Z</time>
</trkpt>
<trkpt lat="-27.600389" lon="-48.518841">
<ele>-9999</ele>
<time>2016-07-04T20:11:01.791Z</time>
</trkpt>
```

Figura 6 – Arquivo GPX com uma trajetória exportado pelo aplicativo MyTracks.

Banco de dados geográficos são bancos de dados com capacidade de armazenar, manipular e indexar dados geográficos para uma área particular ou sujeito. Dado geográfico é qualquer dado que possua um atributo que o relacione a superfície terrestre. Uma das formas de representação desse tipo de dado é o formato vetorial (LONGLEY et al., 2005).

A representação vetorial utiliza formas geométricas como ponto, linha e polígono para representar os objetos existentes na superfície (ex., uma pessoa ou um poste podem ser representados, através de suas coordenadas, como um ponto). Dados vetoriais são representados em um banco de dados geográfico pelo tipo de dado *geometry*.

O banco de dados geográfico tem a capacidade de realizar operações sobre os dados geográficos adicionando novos tipos de índices, que aumentam a eficiência das pesquisas, e várias novas tabelas para gerenciar metadados referentes aos vários tipos de dados que podem ser necessário armazenar. Também é possível executar funções geométricas para manipular ou efetuar consultas sobre os dados armazenados (SHAW, 2013).

A Figura 7 ilustra uma tabela no banco de dados PostgreSQL[11] com dados de trajetórias de um caminhão na Grécia e a coluna *geom* apresenta o tipo *geometry* do objeto. Através da extensão espacial PostGIS[12] o banco de dados permite a criação de uma coluna do tipo *geometry*, que pode ser preenchida com um atributo do tipo ponto, criado a partir da latitude e longitude do objeto. Dessa forma é possível realizar operações espaciais utilizando este ponto, como por exemplo, verificar se um ponto está dentro de determinada região.

	gid bigint	truckid integer	tid integer	time timestamp without time z	lon numeric	lat numeric	geom geometry(Point)
1	19307	862	1	2002-09-10 09:15:59	23.845089	38.018470	0101000020 31BF0D0030D62B956B404441BD4851D5A47A5141
2	19308	862	1	2002-09-10 09:16:29	23.845179	38.018069	0101000020 31BF0D00945F92977040444616192F3AA967A5141
3	19309	862	1	2002-09-10 09:16:59	23.845530	38.018241	0101000020 31BF0D000211EF208440444 1BBD665BE9C7A5141
4	19310	862	1	2002-09-10 09:17:29	23.845499	38.017440	0101000020 31BF0D004ED6376782404441035BC072807A5141
5	19311	862	1	2002-09-10 09:17:59	23.844780	38.015609	0101000020 31BF0D00BD42436Z5A40444134C8D9C43F7A5141

Figura 7 – Tabela com trajetórias de caminhões com a coluna do tipo *geometry* em um banco de dados geográfico.

[11] www.postgresql.org
[12] www.postgis.net/

2.2.4 PROBLEMAS COMUNS

Ao longo do processo de coleta de trajetórias, uma série de problemas comuns podem afetar a qualidade e a estrutura dos dados, dificultando uma posterior tarefa de mineração. Entre esses problemas pode-se destacar três deles, que ocorrem de forma recorrente nas bases de dados de trajetórias geralmente utilizadas na literatura: i) erros causados pela imprecisão do sinal GPS; ii) a inexistência de dados em partes de uma trajetória, causada pela completa perda de comunicação entre o receptor e o sistema de localização (GPS); e iii) problemas estruturais gerados pela má configuração/estruturação dos dados no aplicativo de origem.

De forma simplificada, o sensor presente no dispositivo móvel (ex., *smartphone*) se comunica com uma série de satélites pertencentes a um sistema de localização (ex., GPS) para obter a posição atual do objeto. Quanto maior o número de satélites com os quais o sensor consegue se comunicar, mais precisa é essa localização. Entretanto, uma série de fatores como condições climáticas, a existência de obstáculos como prédios, entre outros, podem causar a perda de comunicação com um ou mais satélites, que resulta na perda de precisão da localização obtida. Geralmente em condições normais os erros de precisão são aleatórios e relativamente pequenos (variando de poucos metros até poucas dezenas de metros de acordo com a condição ambiental) porém uma restrição momentânea, que afeta a comunicação com muitos satélites, pode levar a um erro sistemático onde ocorre uma grande degradação na posição da localização obtida criando pontos discrepantes, a grandes distâncias da localização original (por vezes ultrapassando as centenas de metros) (YAN et al., 2010). Esse tipo de problema é referenciado na literatura como ruído e pode afetar de forma determinante o resultado de uma técnica de mineração por distorcer qualquer medida que considere a distância dos pontos obtidos (PARENT et al., 2013).

Outro tipo de problema é a falta de dados relativos a uma parte da trajetória. Por exemplo, uma trajetória é gerada com pontos a cada 5s, porém entre dois pontos dessa trajetória existe um "buraco" de 5 minutos onde não existe ponto algum, é outro problema que pode prejudicar o processo de mineração. Essa situação é causada pela completa perda de comunicação entre o dispositivo móvel e o sistema de localização de forma a não ser possível coletar qualquer ponto. Duas situações onde esse problema costuma ocorrer são: i) quando existe uma grande interferência ambiental (ex.,, quando um indivíduo atravessa um túnel ou entra em um elevador) ou uma falha no dispositivo (ex., termina a bateria) que leva a completa restrição do sinal; ou ii) quando o objeto atua de forma ativa, intencionalmente interrompendo a coleta ou desligando o aparelho. Esse tipo de problema é geralmente referenciado na literatura como *hole* (primeiro caso) ou *semantic gap* (segundo caso) (PARENT et al., 2013).

Por fim, a estruturação dos dados proveniente das configurações no processo de coleta pode também ser apontada como um problema recorrente. Entre outros fatores que compõe esse problema é possível citar: i) a falta de informações corretas acerca dos dados (ex., qual a projeção cartográfica ou qual a zona de fuso horário); e ii) a falta de uma clara identificação

única das trajetórias (*tid*), resultando em dúvidas de quando uma trajetória é iniciada ou finalizada e como essas trajetórias deveriam ser divididas.

O segundo e terceiro problemas apontados também podem afetar diretamente o resultado de algoritmos que tratem as trajetórias como únicas e contínuas. Por exemplo, no caso da coleta de dados de táxi como a realizada no projeto CRAWDAD que monitorou continuamente táxis na cidade de San Francisco (EUA) (PIORKOWSKI; SARAFIJANOVIC-DJUKIC; GROSSGLAUSER, 2009). Nesse conjunto de dados cada táxi possui um identificador para todas as suas trajetórias em um período de mais de 20 dias. Porém nessas trajetórias é possível identificar períodos com grandes *gaps* (possivelmente ocasionados pelo desligamento intencional do aparelho) e também diferentes situações onde a trajetória poderia ser dividida com a aplicação de um determinado critério (nesse caso, por exemplo, existe um atributo "*occupation*" que determina se o táxi estava livre ou ocupado e poderia ser utilizado para separar em várias trajetórias, por exemplo, onde o taxista procura por passageiro ou está ativamente em uma corrida). Portanto, uma série de métodos que trata trajetórias de forma única e contínua poderiam se beneficiar de uma melhor estruturação dessas trajetórias, segmentando-as em muitas diferentes trajetórias de acordo com diferentes critérios, como a distância entre dois pontos (ex., se existe mais de 10km entre dois pontos a trajetória é segmentada), temporais (ex., se existe um intervalo de tempo maior do que 10min entre dois pontos a trajetória é segmentada) ou semânticos (ex., se existe uma mudança no estado do atributo "*occupation*" a trajetória é segmentada). Em razão da existência recorrente desses três problemas uma etapa inicial de limpeza e pré-processamento é fundamental para garantir a qualidade de diversas técnicas de mineração de dados que podem ser aplicadas sobre dados de trajetória. Na Seção 2.3 algumas dessas técnicas são detalhadas.

2.3 PRÉ-PROCESSAMENTO E LIMPEZA DE BASE DE DADOS DE TRAJETÓRIAS

A etapa de pré-processamento consiste na aplicação de diferentes estratégias e técnicas para realizar as tarefas de carregamento, conversão, limpeza e organização dos dados (TAN; STEINBACH; KUMAR, 2005). Esse conjunto de tarefas geralmente segue uma ordem de execução, porém a execução de algumas delas é opcional, como por exemplo, realizar apenas a tarefa de limpeza em uma base previamente carregada.

Nesse processo a tarefa de carregamento é responsável por realizar a importação dos dados de fontes externas, como arquivos nos formatos citados na Seção 2.2.3, realizando as conversões de dados necessárias e/ou criação de dados faltantes para correções de possíveis problemas de estruturação dos dados. Por exemplo, no momento do carregamento pode ser necessário realizar a conversão das coordenadas do modelo de referência geográfica WGS84[13] para o modelo EPSG:3857[14] e/ou criar um identificador para cada arquivo que está sendo carre-

[13]http://spatialreference.org/ref/epsg/wgs-84/
[14]http://wiki.openstreetmap.org/wiki/EPSG:3857

gado. Desse modo para determinado conjunto de arquivos, em que cada arquivo é considerado uma trajetória única, será criada uma coluna adicional no banco de dados para o identificador de trajetória (*tid*), assim será registrado um valor *tid* único para os dados oriundos de cada arquivo (FURTADO, 2014).

A estruturação dos dados realizada na tarefa de carregamento pode facilitar a tarefa de limpeza de dados. Assim podemos, por exemplo, utilizar o *tid*, criado na tarefa de carregamento, para selecionar no banco de dados uma trajetória por vez e submeter esta a um método de remoção de ruídos. Desta forma um método de remoção de ruídos, de maneira geral, percorre a trajetória e verifica se algum ponto está em uma localização muito discrepante. A Figura 8 ilustra uma trajetória com ruído, apontado pela seta.

Figura 8 – Exemplo de ruído em uma trajetória da base de dados GeoLife

Assim os métodos para limpeza de trajetórias podem ser classificados como Suavização de Trajetórias e Remoção de Ruídos na Trajetória. Dos métodos disponíveis na literatura podemos destacar, para remoção de ruídos, a aplicação de regras simples restringindo a velocidade (ALVARES et al., 2009) e uma adaptação do método de agrupamento DBSCAN (ESTER et al., 1996). Para suavização de trajetórias os métodos mais conhecidos são *Mean Filter* e *Median Filter* (ZHENG; ZHOU, 2011), baseados na média e mediana do eixo x e y do ponto. Esses métodos são descritos a seguir:

Mean Filter: O filtro de médias é utilizado para suavizar a trajetória e consiste em considerar, dado um ponto $p_i = (x_i, y_i)$ de uma trajetória, os w pontos anteriores e posteriores (inclusive o ponto p_i) para calcular um novo valor $\hat{p}_i = (\hat{x}_i, \hat{y}_i)$, que é uma estimativa do valor verdadeiro de p_i (desconhecido). Esse valor é estimado pelo cálculo da média dos valores dos pontos anteriores e posteriores para as coordenadas x e y separadamente. Nas Equações 2.1 e 2.2 são definidos os cálculos para cada coordenada.

$$\hat{x}_i = \frac{1}{n} \sum_{j=i-w+1}^{i} x_j \tag{2.1}$$

$$\hat{y}_i = \frac{1}{n} \sum_{j=i-w+1}^{i} y_j \tag{2.2}$$

Para suavizar uma trajetória inteira essa função deve ser aplicada a cada ponto, o que resulta em uma nova trajetória com a estimativa dos pontos, dado por $(p_1, p_2, \ldots, p_{w-1}, \hat{p}_w, \ldots, \hat{p}_n)$. Observe que os primeiros $w-1$ pontos não são estimados, pois não possuem w pontos do passado para efetuar o cálculo da média.

Median Filter: O filtro de mediana, similar ao da média, também considera os w pontos anteriores e posteriores (inclusive o ponto p_i) para calcular a estimativa $\hat{p}_i = (\hat{x}_i, \hat{y}_i)$, mas utiliza a operação de mediana. A mediana consiste em ordenar o conjunto dos w valores e escolher o elemento central do conjunto ordenado. Quando w for par, tem-se dois elementos centrais, então pode se utilizar a média do par de elementos centrais. Sejam $(x'_1, x'_2, \ldots, x'_w)$ o conjunto de valores ordenados de $(x_{i-w+1}, \ldots, x_{i-1}, x_i)$ e $(y'_1, y'_2, \ldots, y'_w)$ o conjunto de valores ordenados de $(y_{i-w+1}, \ldots, y_{i-1}, y_i)$, as Equações 2.3 e 2.4 definem o cálculo da mediana para cada coordenada.

$$\hat{x}_i = \begin{cases} x'_{w/2} & \text{, se } w \text{ é ímpar;} \\ \frac{x'_{\lfloor w/2 \rfloor} + x'_{\lceil w/2 \rceil}}{2} & \text{, se } w \text{ for par.} \end{cases} \tag{2.3}$$

$$\hat{y}_i = \begin{cases} y'_{w/2} & \text{, se } w \text{ é ímpar;} \\ \frac{y'_{\lfloor w/2 \rfloor} + y'_{\lceil w/2 \rceil}}{2} & \text{, se } w \text{ for par.} \end{cases} \tag{2.4}$$

Para aplicar este filtro em uma trajetória inteira deve-se utilizar a função para cada ponto da trajetória, nas suas coordenadas x e y.

DBSCAN: É um algoritmo de agrupamento baseado em densidade. Dado um conjunto de pontos em algum espaço, esse agrupa os pontos que estão próximos e considera discrepantes os pontos que estão sozinhos em baixa densidade. Dessa forma, uma implementação[15] válida para encontrar ruídos em trajetórias é o agrupamento, para cada ponto, de uma quantidade de pontos específica dentro de um raio. Se houver algum ponto que não possua em sua vizinhança a quantidade especificada, então esse deve ser removido da trajetória.

Entretanto para algumas aplicações pode ser conveniente a remoção de ruídos considerando a aplicação de filtros simples de velocidade entre dois pontos (FURTADO, 2014). Essa é uma abordagem simples que toma uma velocidade como referência e verifica todos os pontos de uma trajetória da seguinte maneira: É selecionado o primeiro e segundo ponto de uma trajetória como pontos atual e seguinte, respectivamente. Assim é calculado a velocidade entre esses pontos, se a velocidade obtida for maior que a referência, então um dos pontos deve ser excluído de acordo com a abordagem utilizada, senão é tomado o ponto seguinte como atual, obtido o novo ponto seguinte e repetido, sucessivamente, o processo até a verificação completa de todos os pontos da trajetória. Neste modelo podem ser adotadas duas abordagens para exclusão dos pontos, sempre utilizando uma janela de dois pontos(atual e seguinte): i) Exclusão do ponto atual, atribuir o ponto seguinte como atual, obter novo ponto seguinte e repetir o processo até completar toda a trajetória. ii)Exclusão do ponto seguinte, continuar com o atual, obter novo ponto seguinte e repetir o processo até completar toda a trajetória. Estas abordagens podem gerar trajetórias diferentes conforme a posição do ruído. Por exemplo se o ruído for o primeiro ponto e for utilizada a segunda abordagem então provavelmente toda a trajetória estará comprometida. Entretanto se for optado pela primeira abordagem, o ruído será eliminado imediatamente.

[15] http://thechaoscomputing.blogspot.com.br/2015/10/removing-noise-from-gps-trajectory-with.html

Após essas etapas as trajetórias já estão carregadas no banco de dados e os pontos identificados como ruídos foram removidos ou suavizados. Assim a tarefa de organização realizará a segmentação das trajetórias, que consiste nos seguintes passos: É criado, no banco de dados, uma *sequence* numérica para o *tid*. Em seguida é realizada a seleção das trajetórias, ordenadas pelo tempo através do *timestamp*, no banco de dados e o algoritmo percorrerá cada trajetória realizando a segmentação conforme seu tipo. Dessa forma os tipos de segmentações são:

i) Baseado em Atributo de Estado: A segmentação ocorre de acordo com o estado de um atributo. por exemplo, através de um atributo do tipo "*occupation*", que indica o estado de ocupação de um táxi, o algoritmo percorrerá cada ponto da trajetória mantendo o mesmo *tid*, enquanto não houver mudança de estado. Quando ocorrer mudança no estado do objeto então será atribuído o próximo valor da *sequence* para o *tid*. Neste momento o ponto seguinte possui um *tid* diferente do ponto anterior, caracterizando uma nova trajetória.

ii)Intervalo de tempo máximo entre dois pontos: Semelhante a segmentação por distância, porém este algoritmo toma um valor de tempo como referência, geralmente em segundos. Dessa forma é realizado o cálculo da diferença de tempo, através do atributo do tipo *timestamp* de cada ponto, entre o ponto seguinte e o ponto atual. Se a diferença de tempo entre os dois pontos for maior que o valor referência, então é atribuído o próximo valor da *sequence* ao *tid*, senão continua mantendo o mesmo *tid*.

iii) Distância máxima entre dois pontos: Dado uma distância como referência, o algoritmo percorrerá a trajetória mantendo o *tid* atual e realizando o cálculo da distância entre o ponto atual e ponto seguinte, se a distância for maior que distância referência então será atribuído o próximo valor da *sequence* para o *tid* do ponto seguinte.

Em todos os três tipos de segmentações o algoritmo irá repetir o processo até o final do conjunto de pontos da trajetória inicial. Sempre passando do ponto atual para o ponto seguinte.

3 DESENVOLVIMENTO

Nesse capítulo é apresentado o sistema desenvolvido, bem como sua arquitetura, telas e técnicas utilizadas para processamento dos dados. O sistema foi desenvolvido como uma aplicação do tipo *desktop* na linguagem Java[1]. Em sua versão atual é compatível com o banco de dados PostgreSQL[2] versão 9.0 ou superior e sua extensão espacial PostGIS[3] versão 2.0 ou superior. Com o código fonte disponível através do link no Apêndice A.

O objetivo do sistema é fornecer ferramentas que facilitem a manipulação de bases de dados de trajetórias, com uma série de funcionalidades relativas às tarefas de carregamento, limpeza e organização desses dados. Dessa forma o sistema possui 6 módulos que implementam as seguintes funcionalidades:

1. Conectar ao banco de dados - Módulo de Conexão ao Banco de Dados (Seção 3.2)

   Permite gerenciar e realizar a conexão ao banco de dados do computador onde será realizado o pré-processamento dos dados de trajetórias.

2. Carregar arquivos de dados externos - Módulo de Carregamento de Dados (Seção 3.3)

   Responsável pela importação dos dados externos para o banco de dados. Esse módulo permite importar dados de arquivos do tipo DSV, GPX, JSON, KML e WKT.

3. Exportar dados de trajetórias - Módulo de Exportação de Dados (Seção 3.4)

   Responsável pela exportação de dados em uma tabela no banco de dados para um arquivo CSV.

4. Limpar ruídos em trajetórias - Módulo de Limpeza de Dados (Seção 3.5)

   Responsável por aplicar técnicas de remoção de ruídos e suavização de trajetórias. Entre as técnicas disponíveis estão a remoção de ruídos por velocidade ou densidade e suavização por média ou mediana.

5. Organizar e segmentar trajetórias - Módulo de Organização e Segmentação de Dados (Seção 3.6)

   Fornece recursos para segmentação de trajetórias por estado de ocupação, intervalo de tempo ou distância máxima entre os pontos.

[1] www.java.com
[2] www.postgresql.org
[3] http://postgis.net/

6. Selecionar trajetórias próximas a um ponto - Módulo de Seleção de Trajetórias Próximas a um Ponto (Seção 3.7)

   Possibilita a seleção de trajetórias que cruzam um circulo com determinado raio com centro em um ponto fornecido.

3.1 ARQUITETURA DO SISTEMA

Para construção do sistema foi adotada uma arquitetura baseada em *Model-View-Controller* (MVC). Essa arquitetura permite a separação do código em camadas, o que facilita a expansão e manutenção do sistema. Assim as classes de código foram agrupadas em pacotes responsáveis pela interface gráfica, lógica/controle do sistema e persistência ao disco.

A Figura 9 ilustra o diagrama de pacotes do sistema desenvolvido, bem como suas classes de código e relações entre pacotes.

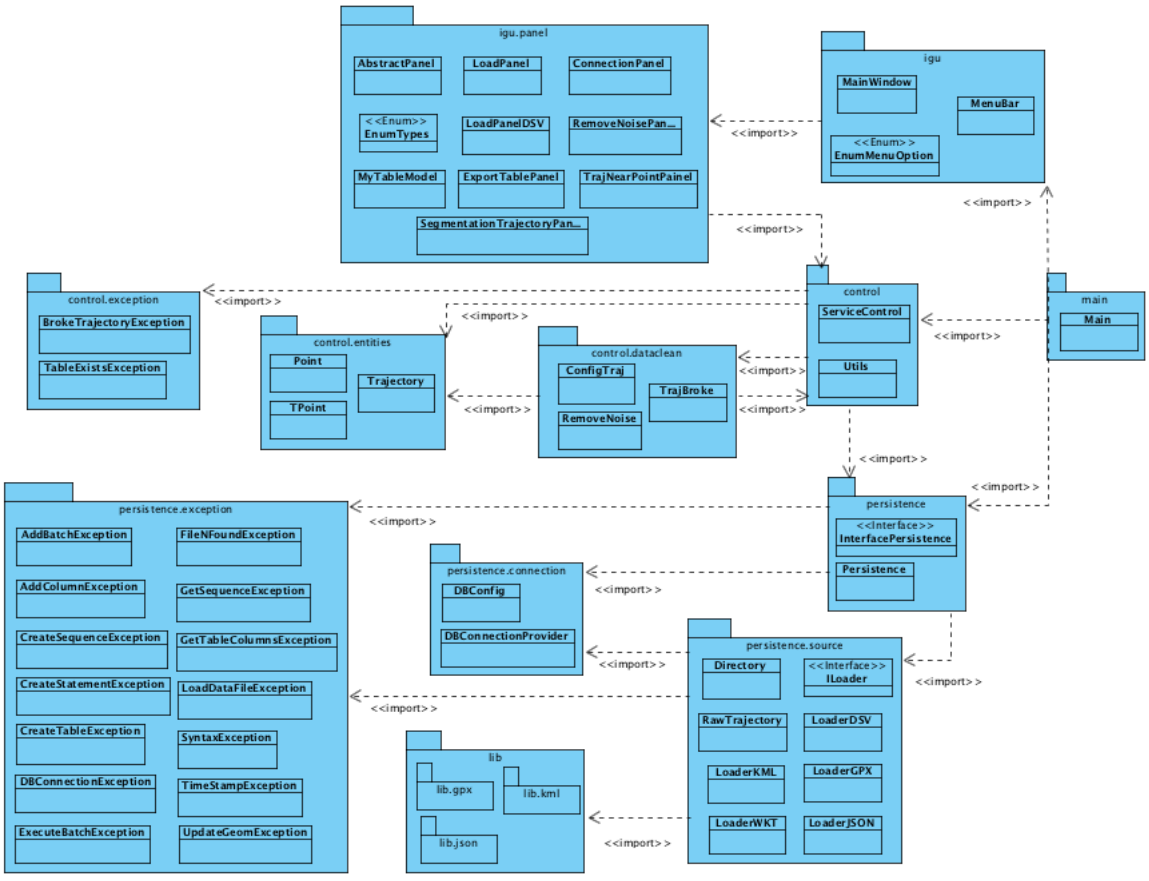

Figura 9 – Diagrama de Pacotes ilustrando a arquitetura utilizada para construção do sistema e os relacionamentos entre os pacotes de classes

1. Interface Gráfica do Usuário (pacote *igu*)

   Pacote *igu* possui as classes responsáveis pela estrutura da interface gráfica, bem como seus menus.

   Pacote *igu.panel* possui uma classe responsável para cada tela do sistema. Essa abordagem permite a criação de uma nova tela através da criação de uma nova classe que estenda a classe *AbstractPanel*. As classes desse pacote são responsáveis pela interação com a parte lógica do sistema (*control*).

2. Lógica do Sistema (pacote *control*)

   Pacote *control* é responsável pelo roteamento no sistema, comunicando as funcionalidades com as camadas de persistência e interface gráfica.

   Pacote *control.dataclean* possui classes responsáveis pela limpeza e organização das trajetórias. A classe *RemoveNoise* possui métodos para remoção e suavização de ruídos, enquanto a classe *TrajBroke* possui métodos para o processo de segmentação de trajetórias.

   Pacote *control.entities* reúne classes para abstração de pontos e trajetórias.

   Pacote *control.exception* agrupa as exceções lançadas pela parte lógica do sistema.

3. Persistência do Sistema (pacote *persistence*)

   Pacote *persistence* é responsável pela interação com dados em disco, seja arquivos ou banco de dados.

   Pacote *persistence.source* possui classes responsáveis pelo carregamento de arquivos de dados. Aqui cada classe é responsável por carregar apenas um formato de dados. Para implementar o carregamento de um novo formato de dados, deve-se então criar uma classe que implemente a interface *ILoader*.

   Pacote *persistence.connection* permite criar e gerenciar a conexão com o banco de dados. A classe *DBConnectionProvider* ainda fornece uma camada de abstração para operações no banco de dados.

   Pacote *persistence.exception* reúne as exceções lançadas pela parte de persistência do sistema.

   Pacote *persistence.lib* agrupa as bibliotecas de terceiros utilizadas pelos leitores de arquivos GPX, JSON e KML.

4. Inicialização do Sistema (pacote *main*)

   Responsável pela inicialização do sistema. A classe *Main* tem a função de instanciar as diferentes camadas do sistema e inicializar a interface gráfica.

3.2 MÓDULO DE CONEXÃO AO BANCO DE DADOS

Esse módulo é responsável por gerenciar e realizar a conexão ao banco de dados local do computador que está executando o sistema. Em sua interface é possível especificar o usuário, senha e o banco de dados a ser utilizado pelo sistema. Com essas informações é possível testar a conexão ou realizar a conexão em definitivo, conforme ilustrado pela Figura 10. Realizada a conexão, esta permanecerá ativa para os outros módulos do sistema até que o sistema seja encerrado.

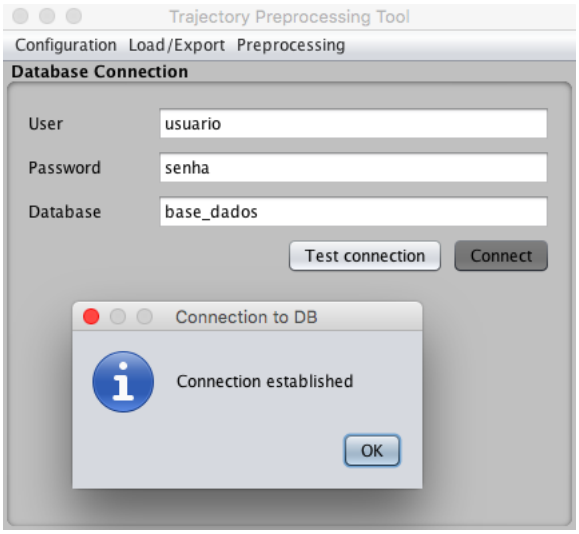

Figura 10 – Tela do módulo para conexão ao banco de dados

3.3 MÓDULO DE CARREGAMENTO DE DADOS

Esse módulo permite realizar a importação de dados externos em diferentes formatos (DSV, JSON, GPX, KML e WKT) para o banco de dados com o qual uma conexão foi previamente estabelecida. Dessa forma, o módulo recebe um arquivo ou diretório como entrada, processa de acordo com um conjunto de parâmetros fornecidos pelo usuário e realiza a inserção em determinada tabela do banco de dados. A Figura 11 apresenta a tela de carregamento de dados do tipo DSV e os parâmetros a serem fornecidos pelo usuário.

Quando um diretório é fornecido como entrada, ele e todos os seus subdiretórios são percorridos pelo sistema em busca de arquivos que contenham dados de trajetórias. Dessa forma, os campos (*Extensions*), (*Ignore directories*), e (*Ignore files*) fornecem as informações necessárias para percorrer os diretórios e selecionar os arquivos. Assim, é possível ignorar ou considerar determinados arquivos, diretórios ou extensões de arquivos durante o processo.

Os campos (*Start after line*) e (*Delimiter*) permitem determinar um número de linhas a serem ignoradas no início dos arquivos (ex., alguns aplicativos incluem linhas de cabeçalho antes dos dados relativos à trajetória) e saber qual seu delimitador padrão (ex., ponto, ponto e vírgula, etc.). Os campos de formato de data e hora, bem como o código SRID (código identificador do sistema de projeção geográfico) permitem a correta formatação e transformação dos dados durante o carregamento, garantindo a padronização no banco de dados.

O campo de seleção (*Save Metadata*) permite a criação de mais uma coluna na tabela no banco de dados chamada *path*, para então registrar o caminho completo do arquivo de origem do dado e uma coluna *folder_id*, onde é criado um identificador único para cada diretório lido. Já os campos de seleção (*Generate serial GID*) e (*Generate serial TID*) são para geração de identificadores únicos para os pontos e trajetórias. O campo (*Generate serial GID*) cria uma *sequence* e atualiza todas tuplas do banco de dados com os identificadores. Já o campo (*Generate serial TID*) cria uma *sequence* e insere o mesmo valor para todos os registros de cada arquivo lido, pegando o próximo valor da *sequence* na leitura do próximo arquivo, assim identificando todos os registros de um arquivo como uma única trajetória.

A tela do sistema permite flexibilidade quanto a criação da tabela, nela é possível especificar detalhes das colunas e o nome da tabela que será criada, bem como relacionar o nome da coluna no banco de dados com a posição da coluna no arquivo DSV de origem do dado.

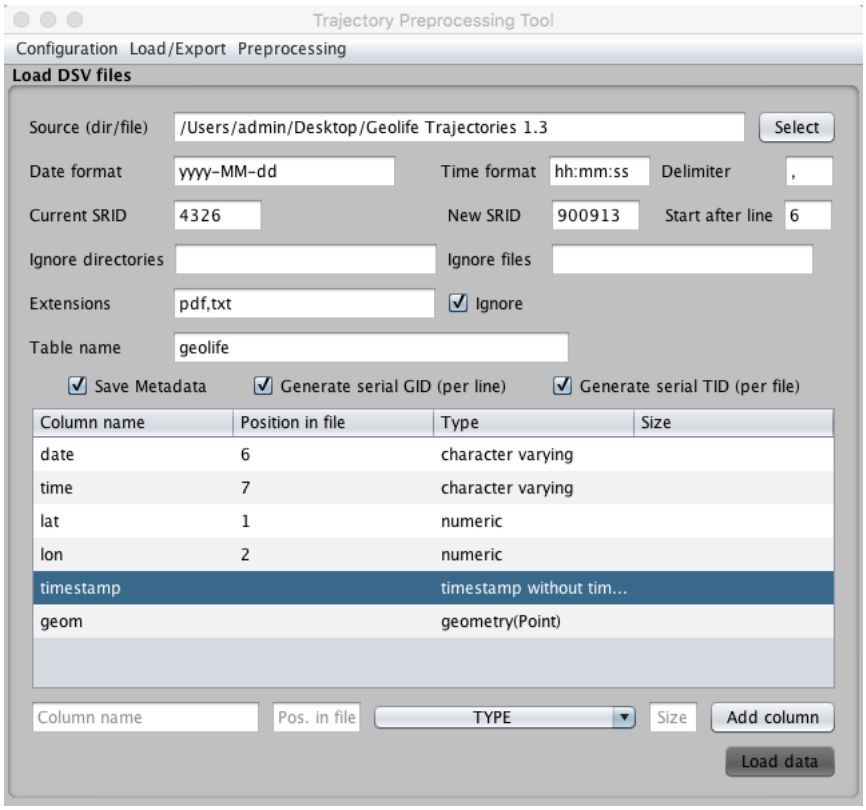

Figura 11 – Tela do módulo para carregamento de dados DSV

Os dados inseridos nos campos da tela de carregamento apresentados pela Figura 11 são para carregamento da base de dados Geolife (ZHENG et al., 2008). A seguir é demonstrado um exemplo de utilização do módulo com essa base de dados. A estrutura dos dados dessa base é apresentada pela Figura 12, onde é possível perceber muitos diretórios e arquivos, incluindo extensões do tipo *.pdf*, *.txt* e *.plt*, este último que segue o formato DSV. Conforme as opções na tela de configuração de carregamento, foi selecionado para ignorar arquivos com extensões *.pdf* e *.txt*, pois somente os arquivos com extensão *.plt* possuem dados de trajetórias. A Figura 13 ilustra o arquivo *.plt* com dados de uma trajetória, bem como o cabeçalho a ser ignorado pelo sistema durante o carregamento.

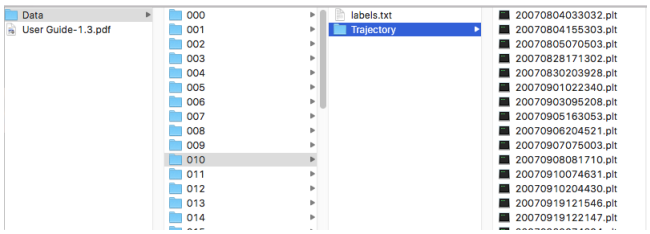

Figura 12 – Estrutura de pastas e arquivos da base de dados Geolife

```
1  Geolife trajectory
2  WGS 84
3  Altitude is in Feet
4  Reserved 3
5  0,2,255,My Track,0,0,2,8421376
6  0
7  39.984702,116.318417,0,492,39744.1201851852,2008-10-23,02:53:04
8  39.984683,116.31845,0,492,39744.1202546296,2008-10-23,02:53:10
9  39.984686,116.318417,0,492,39744.1203125,2008-10-23,02:53:15
10 39.984688,116.318385,0,492,39744.1203703704,2008-10-23,02:53:20
```

Figura 13 – Trecho de um arquivo de dados da base de dados GeoLife

Com essas informações na tela do sistema é então criada uma tabela com o nome fornecido no campo (*Table name*). A partir disso o sistema realiza a leitura de cada arquivo *.plt* e insere os dados de trajetórias na tabela criada no banco de dados.

A Figura 14 ilustra 5 registros da base de dados Geolife no banco de dados PostgreSQL após o processo de carregamento. É possível perceber as colunas *path* e *folder_id* criadas durante o processo de carregamento dos dados. A opção de gerar esses dados durante o carregamento permite fornecer mais opções de manipulação da base de dados durante o processo de mineração. Nessa base de dados, por exemplo, cada usuário que participou da coleta tem suas trajetórias em um subdiretório com seu número identificador, ou seja, o campo *folder_id* registra o identificador do usuário.

	gid integer	tid integer	date character var	time character v	lat numeric	lon numeric	timestamp timestamp without time z	geom geometry(Point)	path character varying(150)	folder_id integer
1	3238207	1926	2009-01-16	09:41:01	39.975228	116.36663	2009-01-16 09:41:01	010100002031BF0D0037D8E63F22B568413AE67DC8618C5241	/Users/rogerjames/Downloads/	14
2	3238208	1926	2009-01-16	09:41:03	39.97523	116.36663	2009-01-16 09:41:03	010100002031BF0D0037D8E63F22B56841F0ED15DB618C5241	/Users/rogerjames/Downloads/	14
3	3238209	1926	2009-01-16	09:41:08	39.975231	116.366649	2009-01-16 09:41:08	010100002031BF0D002D8095832B56841CFF161E4618C5241	/Users/rogerjames/Downloads/	14
4	3238210	1926	2009-01-16	09:41:11	39.97523	116.366663	2009-01-16 09:41:11	010100002031BF0D00AD8274B52B56841F0ED15DB618C5241	/Users/rogerjames/Downloads/	14
5	3238211	1926	2009-01-16	09:41:16	39.97523	116.366673	2009-01-16 09:41:16	010100002031BF0D009CCD13D92B56841F0ED15DB618C5241	/Users/rogerjames/Downloads/	14

Figura 14 – Dados da base Geolife inseridos no PostgreSQL

A Figura 15 apresenta a tela do Módulo de Carregamento responsável pelos arquivos JSON, GPX, KML e WKT. Nessa tela temos as mesmas funções que a tela de carregamento DSV, porém com a seguinte restrição: não é possível customizar a criação de colunas no banco de dados, devido a padronização dos arquivos aqui tratados. Esses arquivos se dão através de marcação por *tags* definidas para trajetória. A definição de *tags* para dados de trajetórias garante padronização dos dados e ao mesmo tempo limita os dados disponíveis somente ao que está especificado. Dessa forma o sistema ficou limitado a usar apenas o grupo de *tags* definidas. De maneira geral são carregadas as coordenadas e o instante de tempo do ponto em cada trajetória.

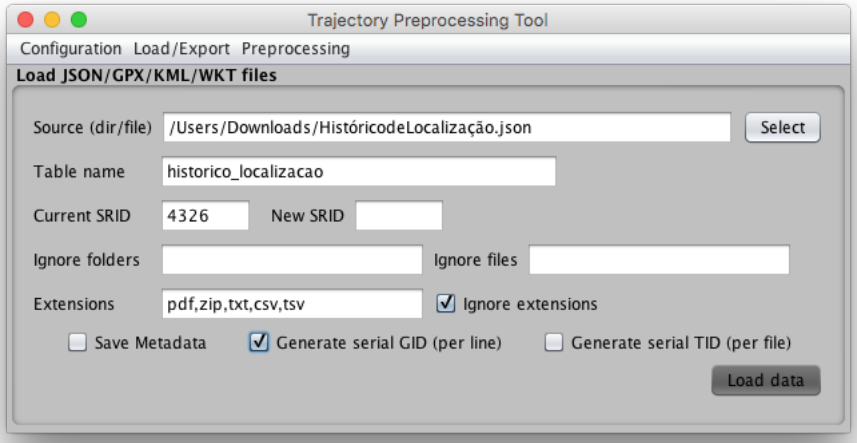

Figura 15 – Tela do módulo para carregamento de dados JSON/GPX/KML/WKT

O módulo de Carregamento de Dados aceita como entrada um único arquivo ou diretório. O sistema procura por arquivos nos formatos aceito sempre que um diretório é fornecido. Toda vez que um arquivo com formato aceito é detectado, então inicia-se o processo de leitura desse arquivo. O processo de leitura para carregamento se dá através de um padrão de projeto *Strategy*. Esse padrão de projeto permite instanciar uma classe específica para cada formato de dados. O uso desse padrão de projeto facilita a expansão do sistema a novos formatos de arquivos, sendo necessário apenas adicionar uma nova classe de código ao sistema. Já para o carregamento dos dados foram então utilizadas algumas *Application Programming Interface* (API) gratuitas. Essas APIs permitiram simplificar o processo de carregamento de dados e futuras manutenções do sistema, pois essas buscam abstrair o processo de leitura dos arquivos de dados. A seguir são apresentados os padrões de formatação de dados suportados pelo sistema nos diferentes tipos de arquivos aceitos, bem como as APIs utilizadas em cada um desses arquivos.

JSON possui seus dados no formato chave/valor e nesse trabalho foi utilizado o formato de trajetórias utilizado no Google Timeline[4]. O Google Timeline disponibiliza um arquivo JSON com uma coleção de *locations* do usuário. Uma *location* caracteriza uma trajetória pois é o conjunto de locais que o usuário passou. Para trabalhar com esse formato de dados foi utilizado a API JSON.org[5]. Essa API permitiu extrair as *locations* e seus registros(pontos) dos arquivos JSON. Cada registro possui as *tags timestampMs* para o instante de tempo do registro em microsegundos e *latitudeE7*/*longitudeE7* para latitude e longitude, respectivamente.

GPX é um formato XML para intercâmbio de dados entre aplicações GPS. Esse formato possui *tags* específicas que definem os dados. Entre essas *tags* devemos destacar as *tags* <*trk*> para trajetórias, <*trkpt*> para pontos e <*time*> para o intervalo de tempo, Conforme a documentação[6]. Para abranger as *tags* definidas na documentação foi utilizado a API MapThing[7] para leitura dos arquivos GPX. Essa API fornece uma lista de objetos Java com as trajetórias. Cada item dessa lista é uma trajetória formada por uma lista de objetos Java, que são os pontos da trajetória. Assim é necessário apenas acessar os atributos de cada um desses pontos e inseri-los no banco de dados.

KML também é uma notação baseada em XML e, de maneira muito similar ao GPX, possui as *tags* específicas <*gx:Track*>, <*gx:coord*> e <*when*>. Conforme a documentação[8] oficial essas *tags* definem trajetórias, par de coordenadas e instante de tempo, respectivamente. Para leitura dos arquivos KML foi utilizado a API Java API para KML - JAK[9]. Essa API abstrai a leitura dos arquivos KML, fornecendo uma lista com os pares de coordenadas da trajetória e uma lista com os instantes de tempo de cada par de coordenada. A partir dessas listas são realizados os registros dos pontos no banco de dados.

WKT é uma linguagem de marcação de texto utilizada para representação de formas geométricas e permite o intercâmbio de dados espaciais entre diferentes aplicações. Com o intuito de carregar apenas dados de trajetórias o leitor WKT desse trabalho considera que cada linha do arquivo seja um ponto geométrico. A OpenGIS [10] define que um ponto seja representado no formato *point(0,0)*. Assim o sistema espera que, além do ponto, cada registro do arquivo possua também um identificador de trajetória(tid) e o instante de tempo desse ponto. Ficando assim o formato de cada registro: *tid; yyyy-MM-dd HH:mm:ss; point(0,0)*. Assim utilizando o padrão de instante de tempo(*timestamp*) do banco de dados PostgreSQL.

[4] www.google.com/maps/timeline
[5] www.json.org
[6] www.topografix.com/gpx.asp
[7] www.reades.com/MapThing/
[8] http://developers.google.com/kml/
[9] https://labs.micromata.de/projects/jak/quickstart.html
[10] http://www.geoapi.org/3.0/javadoc/org/opengis/referencing/doc-files/WKT.html

3.4 MÓDULO DE EXPORTAÇÃO DE DADOS

A exportação de dados permite que uma tabela do banco de dados seja exportada para um arquivo CSV. Essa exportação de dados pode ser feita para compartilhamento de resultados após pré-processamento dos dados, ou até mesmo para intercâmbio de dados entre sistemas.

Através do botão *Select* é especificado o diretório destino onde será criado o arquivo CSV. O botão *Find* busca no banco de dados a tabela informada no campo texto e apresenta os nomes de suas colunas, para o usuário conferir se é esta a tabela a ser exportada. Ao exportar, então os dados serão gravados, no diretório informado, em um arquivo CSV com o nome da tabela origem e os nomes das colunas da tabela no cabeçalho do arquivo.

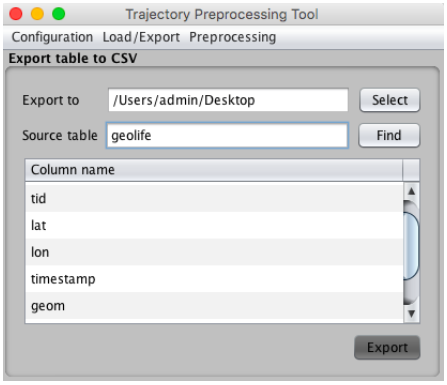

Figura 16 – Tela do módulo para exportação de tabela do banco de dados para arquivo CSV

A Figura 17 ilustra os registros da tabela de origem, utilizada pelo sistema para exportar para arquivo CSV. Assim o sistema criou um arquivo CSV com o nome da tabela e em seguida exportou os dados para esse arquivo CSV, como ilustra a Figura 18.

	gid integer	tid integer	date character var	time character var	lat numeric	lon numeric	timestamp timestamp without time zone	geom geometry(Poi	altitude numeric	path character v	folder_id integer
1	1	1	2008-10-23	02:53:04	39.984702	116.318417	2008-10-23 02:53:04	0101000020:	492	/Users/rc	1
2	2	1	2008-10-23	02:53:10	39.984683	116.31845	2008-10-23 02:53:10	0101000020:	492	/Users/rc	1
3	3	1	2008-10-23	02:53:15	39.984686	116.318417	2008-10-23 02:53:15	0101000020:	492	/Users/rc	1
4	4	1	2008-10-23	02:53:20	39.984688	116.318385	2008-10-23 02:53:20	0101000020:	492	/Users/rc	1
5	5	1	2008-10-23	02:53:25	39.984655	116.318263	2008-10-23 02:53:25	0101000020:	492	/Users/rc	1

Figura 17 – Registros da base de dados Geolife no banco de dados

```
1  gid,tid,date,time,lat,lon,timestamp,geom,altitude,path,folder_id
2  1,1,2008-10-23,02:53:04,39.984702,116.318417,2008-10-23 02:53:04,010100002031BF0D00C404695E83B26841CE41BFDDB98D5241,492,/Users/rogerjames/Desktop/Geolife Trajectories 1.
   3/Data/000/Trajectory/20081023025304.plt,1
3  2,1,2008-10-23,02:53:10,39.984683,116.31845,2008-10-23 02:53:10,010100002031BF0D003AAFF6D383B268413BB5142DB98D5241,492,/Users/rogerjames/Desktop/Geolife Trajectories 1.
   3/Data/000/Trajectory/20081023025304.plt,1
4  3,1,2008-10-23,02:53:15,39.984686,116.318417,2008-10-23 02:53:15,010100002031BF0D00C404695E83B26841BFBDF948B98D5241,492,/Users/rogerjames/Desktop/Geolife Trajectories 1.
   3/Data/000/Trajectory/20081023025304.plt,1
5  4,1,2008-10-23,02:53:20,39.984688,116.318385,2008-10-23 02:53:20,010100002031BF0D0034486BEC82B268411F6E925BB98D5241,492,/Users/rogerjames/Desktop/Geolife Trajectories 1.
   3/Data/000/Trajectory/20081023025304.plt,1
```

Figura 18 – Registros da base de dados Geolife em arquivo CSV exportado pelo sistema

3.5 MÓDULO DE LIMPEZA DE DADOS

Esse módulo é responsável pela aplicação das diferentes técnicas de remoção de ruídos. A Figura 19 apresenta a tela para remoção de ruídos com filtro por velocidade, filtro por densidade e suavização através da média e mediana, conforme descritos na Seção 2.3. Assim essa tela permite especificar uma tabela no banco de dados e identificar os tipos de colunas da tabela no banco de dados. Nesse processo as colunas dos tipos *geom* e *tid* são as mais importantes, pois é a partir delas que será realizada a identificação das trajetórias e a obtenção da coordenada geográfica para o cálculo da velocidade e distância entre os pontos.

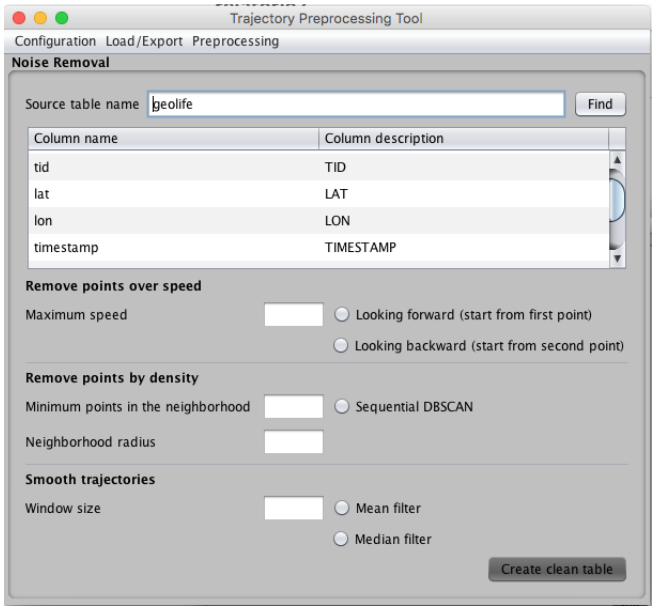

Figura 19 – Tela do módulo para remoção de ruídos em trajetórias

Todas as técnicas implementadas no sistema tratam uma trajetória por vez, sem considerar as demais trajetórias no banco de dados. Assim uma trajetória é fornecida como entrada do algoritmo e esse percorre seus pontos suavizando a trajetória ou identificando e removendo ruídos.

Filtro por velocidade: Esse filtro considera ruído um ponto que está acima da velocidade informada, em metros por segundo, no campo *Maximum speed*. O processo para identificar o ruído é através da verificação da velocidade entre o ponto atual e o ponto seguinte. A partir da constatação de uma velocidade acima da informada, então o algoritmo permite duas opções de exclusão do ruído. Pode-se optar por excluir o ponto atual, opção *Looking forward (start from first point)*, ou o ponto seguinte, opção *Looking backward (start from second point)*. Essas duas abordagens permitem escolhas de acordo com o padrão de ruídos na base de dados. Por exemplo, algumas trajetórias concentram ruídos nos primeiros pontos da trajetória, geralmente ocorridos ainda na sincronização do aparelho GPS, conforme ilustra a Figura 20 e assim pode-se optar pela remoção do primeiro ponto.

A Figura 21 ilustra o resultado da aplicação do Filtro de Velocidade de 150km/h nessas trajetórias.

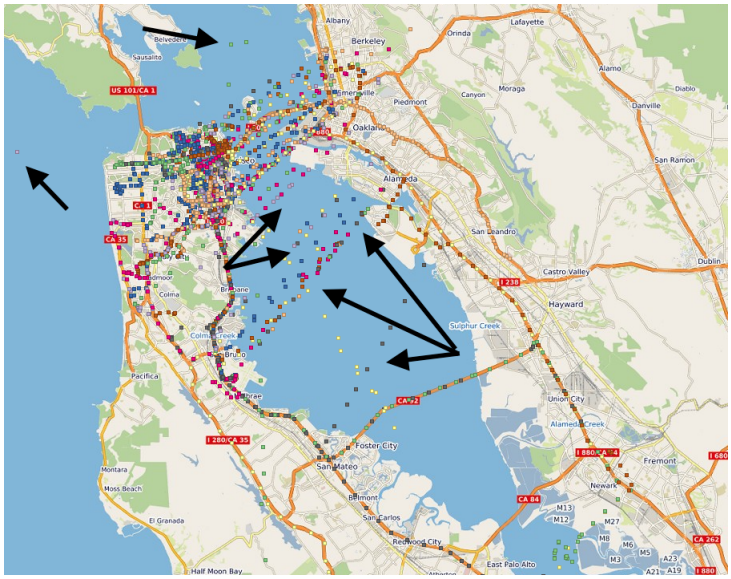

Figura 20 – Ruídos, identificados por setas, em trajetórias da base de dados Taxi San Francisco

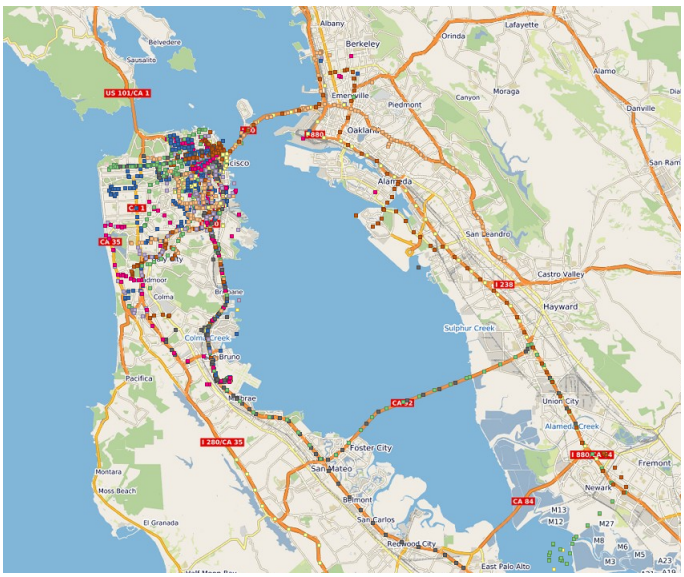

Figura 21 – Trajetórias com ruídos removidos por velocidade na base de dados Taxi San Francisco

Filtro por densidade: Essa abordagem é uma variação do DBSCAN (ESTER et al., 1996) para encontrar ruídos em uma trajetória. O algoritmo verifica, para cada ponto, a existência de uma quantidade mínima de pontos em sua vizinhança, informada no campo *Minimum points in the neighborhood*, dentro de um raio, informado em metros no campo *Neighborhood radius*. Então é calculada a distância Euclidiana do ponto atual até seus pontos seguintes e anteriores, dentro do raio, a fim de verificar a quantidade miníma de pontos em sua vizinhança. Quando ocorrer de um ponto não atingir a quantidade miníma de pontos em sua vizinhança, então esse é considerado ruído e é removido da trajetória.

A Figura 22 ilustra ruídos, apontado por seta, em uma trajetória da base de dados Geolife. Já a Figura 23 ilustra a trajetória após aplicação do Filtro de Densidade de 5 vizinhos em um raio de 100 metros.

Figura 22 – Ruídos apontados por seta em trajetória da base de dados Geolife

Figura 23 – Trajetória da base de dados Geolife após remoção de ruídos por densidade

Suavização pela média e mediana: Esses filtros permitem suavizar a trajetória no sentido de manter os pontos mais alinhados. Para suavizar uma trajetória é necessário calcular novos valores para os eixos x e y de cada ponto da trajetória. Esse cálculo é através da média ou mediana de uma janela de pontos na proximidade do ponto alvo. Dessa forma o algoritmo utiliza um tamanho de janela, informado pelo campo *Window size*, e obtém os pontos na proximidade, anteriores e seguintes, a fim de formar o tamanho da janela. O cálculo da média ou mediana é realizado através dos valores x e y de todos os elementos que compõem a janela. Por fim, o resultado desse cálculo é atribuído ao x e y do ponto alvo e repetido todo processo para cada ponto até o fim da trajetória. A Figura 24 ilustra o processo de suavização por mediana de uma trajetória com ruído. A trajetória original tem seus pontos representados por triângulos e apresenta um ruído distante dos demais pontos da trajetória. Já os pontos da trajetória suavizada estão representados por círculos. A trajetória foi suavizada por mediana utilizando uma janela de 30 pontos. Dessa forma foi calculado a mediana para o ruído e esse foi alocado junto aos demais pontos da trajetória, conforme apontado pela seta na Figura 24.

Figura 24 – Trajetória com ruído em triângulos e a mesma trajetória suavizada por mediana em círculos, com o ruído suavizado apontado por seta

Para cada aplicação de um desses filtros é então criada uma nova tabela no banco de dados, com nome baseado na tabela origem, na técnica e parâmetros utilizados. Por exemplo, *nomeTabelaOrigem_removednoise_median_30*, onde *nomeTabelaOrigem* é o nome da tabela de origem das trajetórias, *median* é a técnica de suavização por mediana e 30 é o parâmetro fornecido para o tamanho da janela.

3.6 MÓDULO DE ORGANIZAÇÃO E SEGMENTAÇÃO DE DADOS

O módulo de organização de trajetórias, ilustrado pela Figura 25, permite realizar a segmentação das trajetórias conforme descrito na Seção 2.3. Para essa segmentação é criada uma nova coluna, de nome *old_tid*, no banco de dados para onde serão copiados todos os valores *tid* originais das trajetórias, preservando assim os identificadores originais. A partir disso o sistema poderá realizar as três segmentações possíveis, porém seguirá a seguinte ordem: primeiro a segmentação por estado do objeto, se o campo *Segment by status change* estiver selecionado e possuir um atributo de valor 0 ou 1 no banco de dados, de forma a identificar se está ocupado ou não. Em seguida será segmentada por tempo, se um valor em segundos for especificado em *Maximum sampling time gap*, onde é verificado o intervalo de tempo entre os pontos. E por último a segmentação por distância entre os pontos, se um valor em metros for especificado em *Maximum distance gap*.

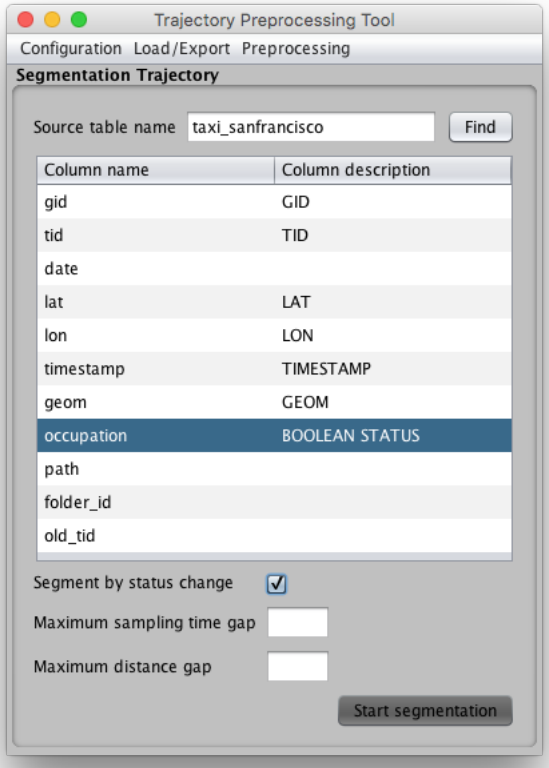

Figura 25 – Tela do módulo para segmentação de trajetórias

A segmentações por intervalos de tempo e por distância visam eliminar grandes intervalos de tempo ou espaço durante uma trajetória. A Figura 26 ilustra uma trajetória com grandes intervalos.

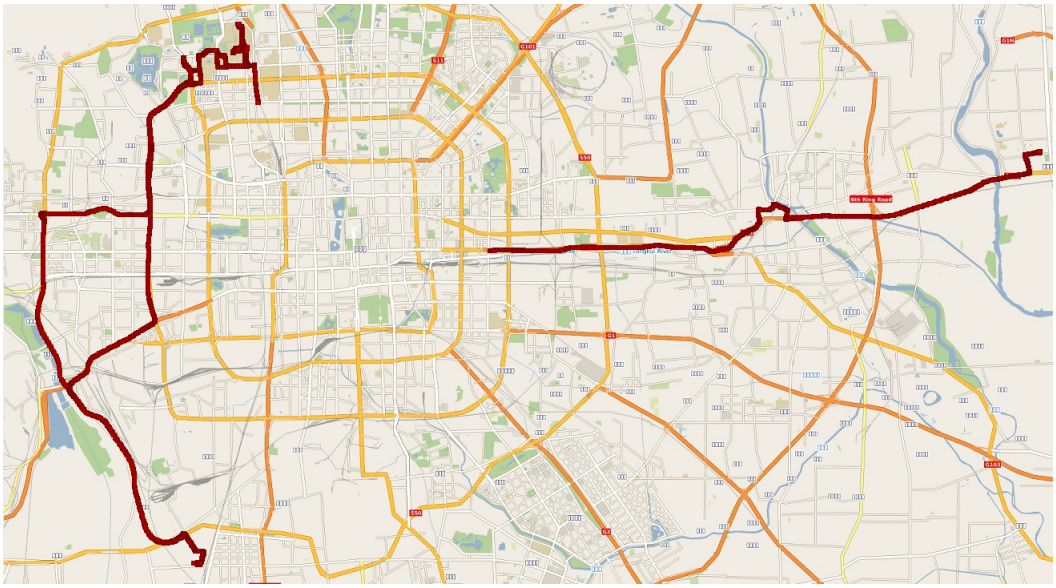

Figura 26 – Uma trajetória com grandes intervalos

Dessa forma a trajetória é segmentada e assim cada segmento é considerado uma trajetória, com início e fim. A Figura 27 ilustra a trajetória segmentada por intervalos de tempo de 5 minutos. Essas trajetórias estão identificadas com cores distintas na Figura 27.

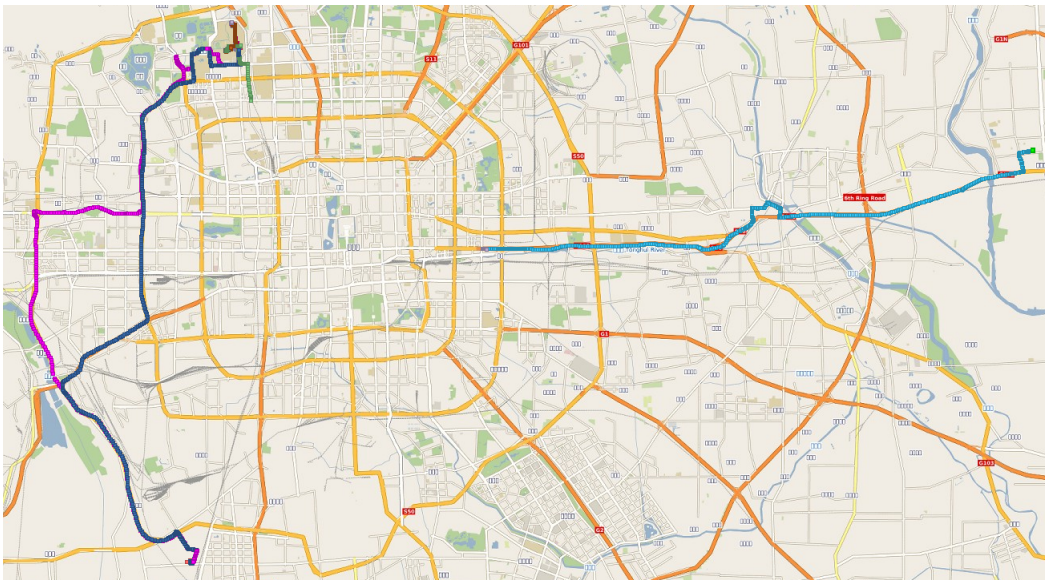

Figura 27 – Segmentos de uma trajetória após processo de segmentação por tempo de 5 minutos

Nesse processo cada técnica de segmentação criará temporariamente uma coluna própria no banco de dados. Para segmentação por estado será criada a coluna *status_tid*, para segmentação por tempo *sample_tid* e *distance_tid* para segmentação por distância. A segmentação por estado utiliza os identificadores de trajetórias da coluna *tid* e salva os novos identificadores da segmentação na coluna *status_tid*. As outras duas técnicas utilizam a coluna da técnica anterior, se essa ocorreu, como identificador da trajetória. Caso não ocorra nenhuma técnica anteriormente, então é utilizado a coluna *tid* como referência para o identificador da trajetória. Ao término das segmentações os valores da coluna da última segmentação serão copiados para a coluna *tid* e as colunas temporárias serão excluídas. Dessa forma as trajetórias recebem novos identificadores e preservam os identificadores originais da base de dados na coluna *old_tid*.

A Figura 28 ilustra uma consulta no banco de dados dos registros de uma trajetória de táxi segmentada por estado (*status*). Nessa Figura é possível verificar que a trajetória possuía um identificador de número 14, preservado no campo *old_tid* e o campo *occupation* apresenta a mudança de estado do táxi. Perceba que o *tid* muda conforme alterna o valor de *occupation*. Esse padrão é o resultado da segmentação por estado, onde temos trajetórias únicas de acordo com o estado de ocupação do táxi.

	gid integer	tid integ	date character var	lat numeric	lon numeric	timestamp timestamp without time	geom geometry(Point)	occupation numeric	path characte	folder_id integer	old_tid numeric
32	275104	2	1211673513	37.77457	-122.4276	2008-05-24 20:58:33	0101000020318	1	/Users/	1	14
33	275103	2	1211673582	37.77376	-122.42499	2008-05-24 20:59:42	0101000020318	1	/Users/	1	14
34	275102	2	1211673688	37.76924	-122.41057	2008-05-24 21:01:28	0101000020318	1	/Users/	1	14
35	275101	2	1211673748	37.766	-122.40532	2008-05-24 21:02:28	0101000020318	1	/Users/	1	14
36	275085	3	1211674421	37.61729	-122.38465	2008-05-24 21:13:41	0101000020318	0	/Users/	1	14
37	275084	3	1211674425	37.61786	-122.38562	2008-05-24 21:13:45	0101000020318	0	/Users/	1	14
38	275083	3	1211674591	37.6178	-122.38722	2008-05-24 21:16:31	0101000020318	0	/Users/	1	14

Figura 28 – Trajetórias segmentadas pelo atributo de estado de ocupação no banco de dados PostgreSQL

3.7 MÓDULO DE SELEÇÃO DE TRAJETÓRIAS PRÓXIMAS A UM PONTO

Esse módulo permite encontrar e selecionar trajetórias que cruzam um círculo com determinado raio e centro em um ponto geográfico especificado. A partir de uma tabela com trajetórias identificadas (*tid*) e com o atributo geométrico do tipo *point*, o sistema verifica as trajetórias que cruzam o círculo traçado. O ponto de referência é fornecido através dos campos de *longitude* e *latitude*, bem como também o raio em metros, através do campo *Distance up to*. Dessa forma o sistema cria uma nova tabela no banco de dados, no formato *nomeTabelaOrigem_ trajsnearpoint*, copiando somente as trajetórias que cruzam o raio do ponto referência.

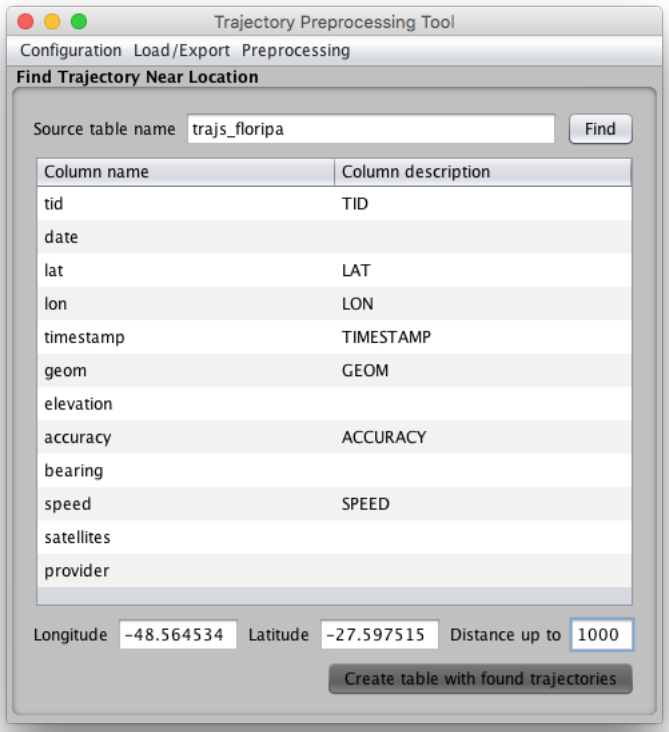

Figura 29 – Tela do módulo para seleção de trajetórias próximas a determinado ponto

A Figura 30 ilustra 11 trajetórias na cidade de Florianópolis que serão utilizadas como base para selecionar trajetórias que passam próxima a ponte que liga à região continental da cidade. Na tela do sistema, ilustrado pela Figura 29, uma região junto as pontes da cidade foi definido como ponto geográfico referência, através das longitude, latitude e distância do raio a ser considerada. Assim o sistema separou, em uma nova tabela, as trajetórias que cruzaram esse ponto. A Figura 31 ilustra duas trajetórias selecionadas por cruzarem o raio de 1km da ponte de Florianópolis.

Figura 30 – Trajetórias na cidade de Florianópolis

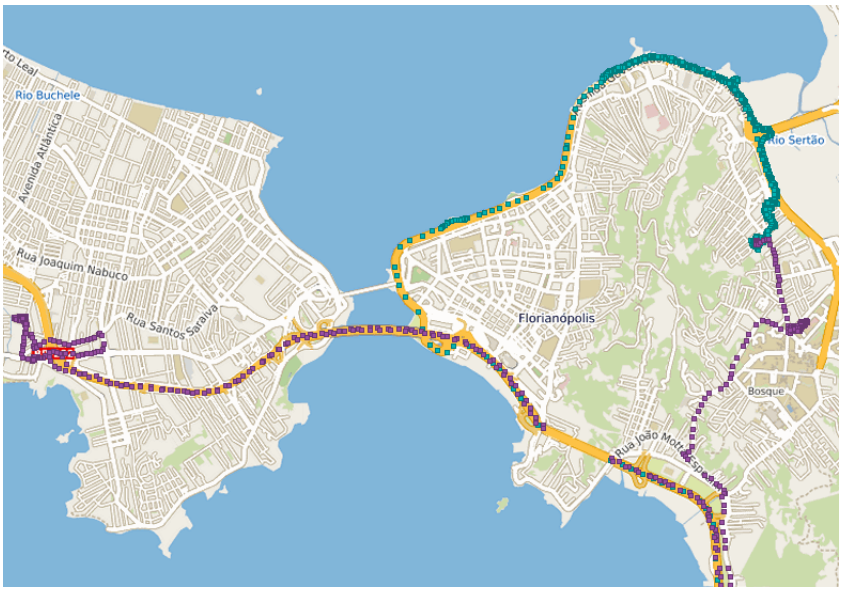

Figura 31 – Trajetórias que cruzam o raio de 1km da ponte de Florianópolis

4 PRÉ-PROCESSAMENTO DE BASES DE DADOS DE TRAJETÓRIAS

Atualmente, diversas bases de dados de trajetórias estão disponíveis gratuitamente na internet. Essas bases são utilizadas por pesquisadores em todo o mundo e entre elas podemos destacar, por serem amplamente utilizadas na literatura, Geolife (ZHENG et al., 2008) com trajetórias de pessoas na China, Taxi San Francisco (PIORKOWSKI; SARAFIJANOVIC-DJUKIC; GROSSGLAUSER, 2009) com trajetórias de táxis no EUA e T-Drive (YUAN et al., 2011) com trajetórias de táxis na China. Todas essas bases de dados estão disponíveis em formatos distintos, conforme disponibilizadas por seus mantenedores e não mantendo padronização entre elas. Assim, toda vez que alguém deseja manipular os dados dessas bases, deve então realizar um esforço para compreender a estrutura dos dados e então realizar a importação para um banco de dados.

Por esse motivo, através de uma busca, foram reunidas 15 bases de dados de trajetórias para serem pré-processadas pelo sistema desenvolvido de forma a validá-lo. Os detalhes de cada uma dessas bases de dados são apresentados na Tabela 1, onde é possível perceber a quantidade de pontos registrados, a quantidade de trajetórias identificadas e a quantidade de objetos que realizaram a coleta dos dados em cada base. Ainda na Tabela 1 há a distância média em metros e o tempo médio em segundos entre os pontos das trajetórias. Esses dois últimos foram levantados após o carregamento das bases de dados.

Tabela 1 – Bases de dados de trajetórias

Base de Dados	Pontos	Trajetórias	Objetos	Distância Média (m)	Tempo Médio (s)
AIS Brest	5 756 438	824	824	615.58	57 096,31
Athens School Bus	66 096	145	2	266.33	1859.85
Cruz dataset	18 107	163	28	63.28	12,47
Dublin Bus	83 436 279	20 908	961	178.08	86.36
Floripa dataset	1 804 499	232	13	49.91	37.63
GeoLife	24 876 978	18 670	182	76.76	18.19
Greek Trucks	112 203	50	50	224.79	255.60
Greek Trucks Rev	94 098	1 100	50	273.72	100.48
NYC buses	840 940	30	30	106.83	2684.29
Taxi Roma	21 817 851	316	316	66.12	78.39
Taxi San Francisco	11 219 955	536	536	512.17	125.75
T-Drive	17 723 420	10 357	10 357	3783.46	3268.67
Uber San Francisco	1 128 663	24 999	N.I.	69.27	17.88
W4M Oldenburg	4 780 954	100 000	N.I.	133.89	600
W4M Milano	1 806 293	45 000	17 087	1417.47	4998.91

Já a Tabela 2 apresenta as bases de dados após o processo de segmentação das trajetórias em 5 minutos, com excessão da base de dados Oldenburg, pois essa é uma base de dados sintéticos e assim não há necessidade de segmentação ou remoção de ruídos. Nessa tabela podemos perceber o aumento no número de trajetórias e uma grande redução do tempo médio entre os pontos das trajetórias, devido a eliminação dos intervalos de tempo iguais ou superiores a 5 minutos nas trajetórias.

Tabela 2 – Bases de dados de trajetórias segmentadas em 5 minutos

Base de Dados	**Pontos**	**Trajetórias**	**Objetos**	**Distância Média (m)**	**Tempo Médio (s)**
AIS Brest	5 756 438	74 614	824	131,29	125,07
Athens School Bus	66 096	828	2	218,32	33,70
Cruz dataset	18 107	178	28	56,99	11,42
Dublin Bus	83 436 279	106 831	961	189,01	20,38
Floripa dataset	1 804 499	1 610	13	23,40	63,96
GeoLife	24 876 978	58 482	182	93,53	8,74
Greek Trucks	112 203	4 353	50	141,23	34,68
Greek Trucks Rev	94 098	3 961	50	219,86	42,52
NYC buses	840 940	1 648	30	111,40	30,67
Taxi Roma	21 817 851	14 060	316	67,87	8,67
Taxi San Francisco	11 219 955	98 525	536	638,72	64,58
T-Drive	17 723 420	3 154 960	10 357	3 166,99	165,68
Uber San Francisco	1 128 663	25 087	N.I.	69,31	9,03
W4M Milano	1 806 293	454 682	17 087	1 441,67	149,45

Quando uma base de dados é disponibilizada na internet, é comum ser acompanhada de uma documentação. Nessa documentação geralmente há informações sobre a quantidade de objetos que registraram suas trajetórias, o modelo de referência geográfico utilizado e o formato do arquivo de origem dos dados. É notável que a documentação é a maneira mais rápida e simples de obter informações sobre uma base de dados. Entretanto quando a documentação é inexistente ou pouco clara sobre os detalhes da base, é então necessário análise dos dados para compreender detalhes simples, como por exemplo quais dados são referentes a latitude e longitude, qual o sistema de referência geográfico utilizado, qual o tempo médio comum entre os pontos ou ainda se as trajetórias possuem identificadores.

Nesse trabalho, foi realizada uma análise dos dados em todas as 15 bases de dados utilizadas, mesmo quando a documentação estava disponível. Essa análise se faz necessária devido a um equívoco comum das documentações, que é a troca da latitude pela longitude. A Figura 32 ilustra a documentação e os dados da base de dados Greek Trucks. Na parte superior da Figura está a documentação que descreve, entre outros detalhes, que as trajetórias são originárias de caminhões entregando concreto na Grécia e a ordem dos dados no arquivo. A ordem dos dados diz que a latitude (lat) é o dado anterior a longitude (lon). Sendo assim, de acordo com a documentação, a primeira linha dos dados no arquivo seria um ponto no meio do mar vermelho, conforme ilustra a Figura 33, o que seria muito distante para um caminhão carregando concreto na Grécia.

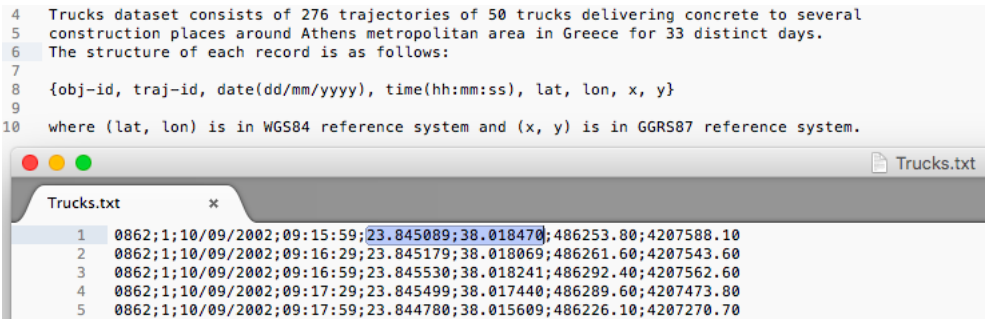

Figura 32 – Documentação e trecho de dados da base de dados Greek Trucks, com latitude e longitude invertidos

Figura 33 – Ponto em local não esperado devido a inversão da latitude e longitude na base de dados Greek Trucks

Por isso os dados devem ser conferidos através de uma amostra da base de dados. Também é necessário compreender como os dados estão estruturados, como se dá a identificação dos objetos e trajetórias. Assim esse trabalho fez um levantamento das características de todas as bases de dados processadas. O resultado desse levantamento é mostrado na Tabela 3. Nessa tabela podemos ver, entre outros detalhes, o tipo de arquivo original dos dados, formato da data e hora, a quantidade de linhas no cabeçalho dos arquivos, o modelo de referência geográfico utilizado e como está organizado o identificador do objeto, se este está registrado junto aos dados ou por pasta.

Tabela 3 – Características dos dados em diferentes bases de dados de trajetórias

Base de dados	Formato data	Formato hora	Delimitador	SRID	Pular linhas	Extensão	Estrutura	ID objeto	TID
AIS Brest	2009-02-05 09:16:25		;	4326	1	.csv	arquivo único	no arquivo	N.I.
Athens buses	18/10/2000	10:15:20	;	4326	0	.txt	arquivo único	no arquivo	no arquivo
Cruz dataset	2014-09-13 07:24:32		,	4326	1	.csv	com arquivos	no arquivo	no arquivo
Dublin Bus	1356998403000000		,	4326	0	.csv	com pastas	no arquivo	no arquivo
Floripa dataset	2016-06-10T12:12:40Z		,	4326	1	.txt	com pastas	por pasta	por arquivo
GeoLife	2008-10-23	02:53:04	,	4326	6	.plt	com pastas	por pasta	por arquivo
Greek Trucks	2002-08-27 09:15:59		,	4326	1	.txt	arquivo único	N.I	no arquivo
Greek Trucks Rev	2002-08-27 09:15:59		,	4121	1	.txt	arquivo único	N.I	no arquivo
NYC buses	2011-04-16 01:02:41		,	4326	1	.csv	arquivo único	no arquivo	N.I.
Taxi Roma	2014-02-01 00:00:03+01		;	4326	0	.wkt	arquivo único	no arquivo	no arquivo
Taxi San Francisco	1213084687		espaço	4326	0	.txt	com arquivos	por arquivo	por arquivo
T-Drive	2008-02-02 15:36:08		,	4326	0	.txt	com pastas	no arquivo	por arquivo
Uber San Francisco	2007-01-07T10:54:50+00:00		tab	4326	0	.tsv	arquivo único	N.I.	no arquivo
W4M Oldenburg	0042		tab	N.I.	0	.txt	arquivo único	N.I.	no arquivo
W4M Milano	03-04-2007 18:48:15		,	4326	0	.csv	com arquivos	no arquivo	no arquivo

Dessa forma a documentação da base de dados e uma breve análise dos dados permitiram maior compreensão dos detalhes que foram úteis para o carregamento dos dados e também para um provável uso futuro desses dados. Em todas as bases de dados carregadas por esse trabalho foi empregado maior esforço em detectar os identificadores das trajetórias e objetos, pois muitas vezes a documentação não foi clara sobre esses detalhes. Por exemplo, a documentação da base de dados W4M Milano[1] informa que os dados foram coletados por 17 000 veículos, porém não é possível identificar esses veículos nos dados, viabilizando somente o uso das trajetórias individualmente.

A seguir é apresentada uma descrição de cada base de dados carregada, bem como as abordagens utilizadas para compreensão dos dados.

AIS Brest[2] é uma base de dados coletados pelo Sistema de Identificação Automático(AIS) de barcos, na cidade de Brest na França, durante o ano de 2009. Esses dados foram coletados de maneira autônoma pelo sistema do barco e transmitidos para a central AIS. A base de dados está disponível em arquivo único do tipo CSV. Nos dados estão as coordenadas geográficas, instante de tempo, velocidade, identificador MMSI do barco e o tipo de barco. Nessa base deve-se considerar todos os registros de um barco como trajetória única, pois não há identificador para as trajetórias.

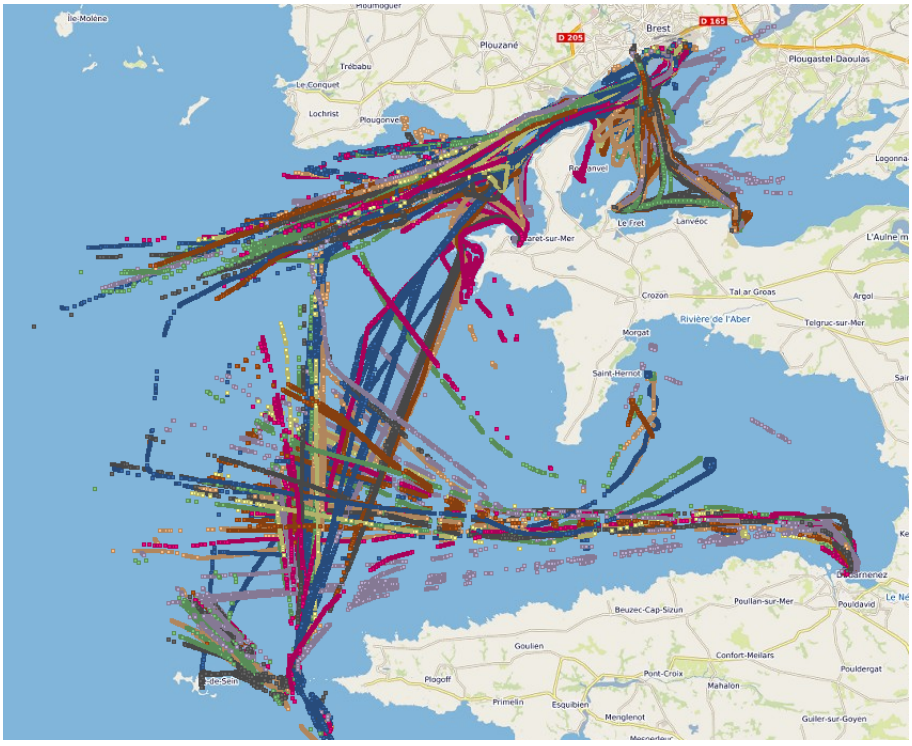

Figura 34 – Trajetórias da Base de Dados AIS Brest

[1] http://kdd.isti.cnr.it/W4M/
[2] https://sites.google.com/site/movingobjectsatsea/data-challenge

Athens School Bus[3] são dados de dois ônibus escolares na região metropolitana de Atenas, Grécia. Esses ônibus registraram um total de 145 trajetórias em um período de 108 dias. A base de dados está disponível em arquivo único de formato CSV. Os dados possuem identificador do objeto, identificador da trajetória, instante de tempo e coordenadas geográficas no padrão WGS84, conforme a documentação disponível.

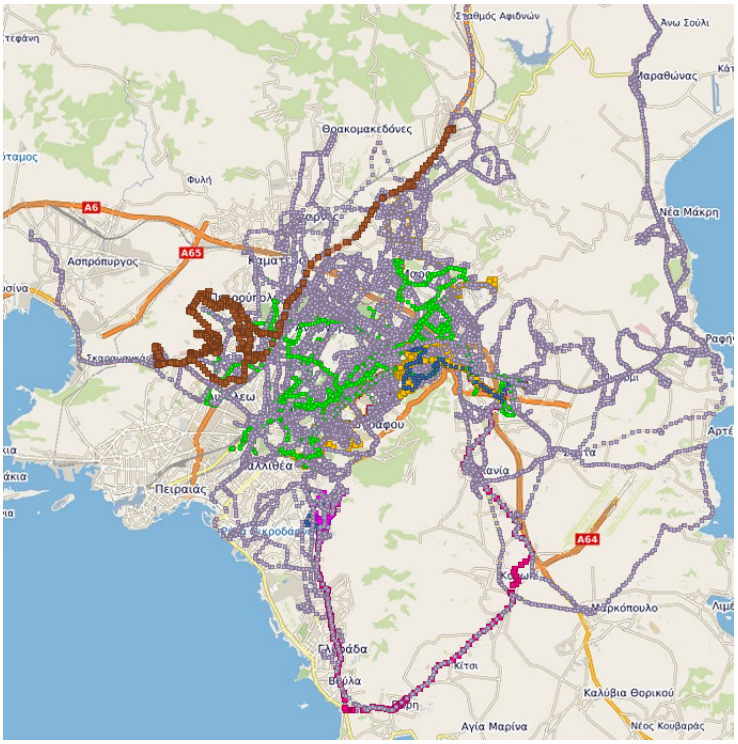

Figura 35 – Trajetórias da Base de Dados Athens School Bus

[3] http://chorochronos.datastories.org/?q=node/10

Cruz dataset[4] são 163 trajetórias coletadas por 28 celulares na cidade de Aracaju, Sergipe, no período entre Setembro de 2014 e Janeiro de 2016. Dessas trajetórias, 87 são de carros e 76 de ônibus. A base de dados está disponibilizada em dois arquivos no formato CSV. Em um arquivo, cada registro contém o identificador da trajetória, identificador do objeto, média de velocidade, tempo em relação ao ponto anterior em segundos, meio de transporte, distância percorrida e o nome da linha de ônibus quando utilizada. No outro arquivo estão os dados das trajetórias, com o identificador da trajetória, instante de tempo e coordenadas. A junção dos dados dos dois arquivos se dá pelo identificador da trajetória disponível em ambos arquivos.

Figura 36 – Trajetórias da Base de Dados Cruz dataset

[4]https://archive.ics.uci.edu/ml/datasets/GPS+Trajectories

Dublin Bus[5] possui mais de 83 milhões de pontos e é a maior base de dados utilizada nesse trabalho. Essa base é formada por dados coletados por 961 ônibus na cidade de Dublin, na Irlanda, entre Novembro de 2012 e Janeiro de 2013. A base de dados está dísponível em pastas com arquivos no formato CSV. Entre os dados estão o instante de tempo em microssegundo, identificador da linha do ônibus, identificador da trajetória do ônibus, direção do ônibus, identificador do operador, trânsito congestionado ou não, tempo que está atrasado, identificador do ônibus, se está em uma parada, identificador da parada e as coordenadas geográficas.

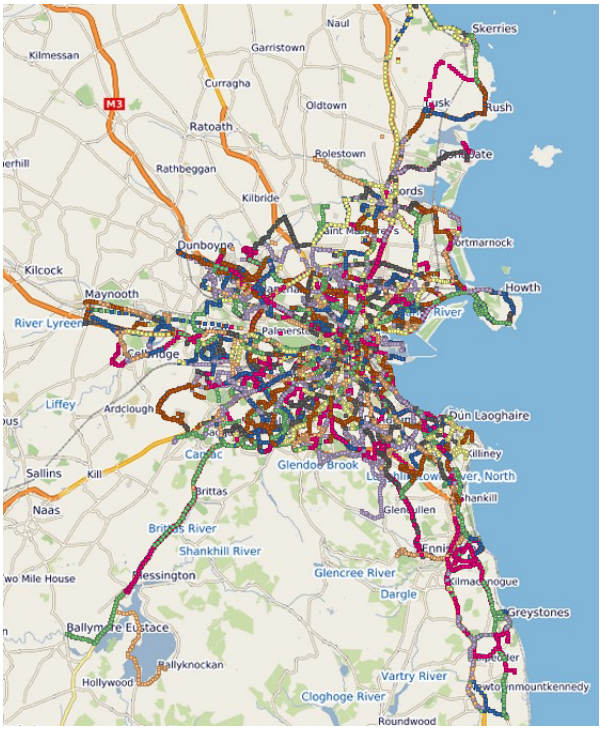

Figura 37 – Trajetórias da Base de Dados Dublin Bus

[5]https://data.gov.ie/dataset/dublin-bus-gps-sample-data-from-dublin-city-council-insight-project

Floripa dataset são dados de trajetórias de 13 pessoas pela cidade de Florianópolis. Os dados foram coletados pelos celulares dessas pessoas entre Março e Julho de 2016, totalizando 232 trajetórias. A base de dados é formada por 13 pastas com arquivos CSV das trajetórias de cada pessoa. Essa base de dados não está disponível na internet, mas foi processada por esse trabalho pois os autores fazem parte da autoria dos dados.

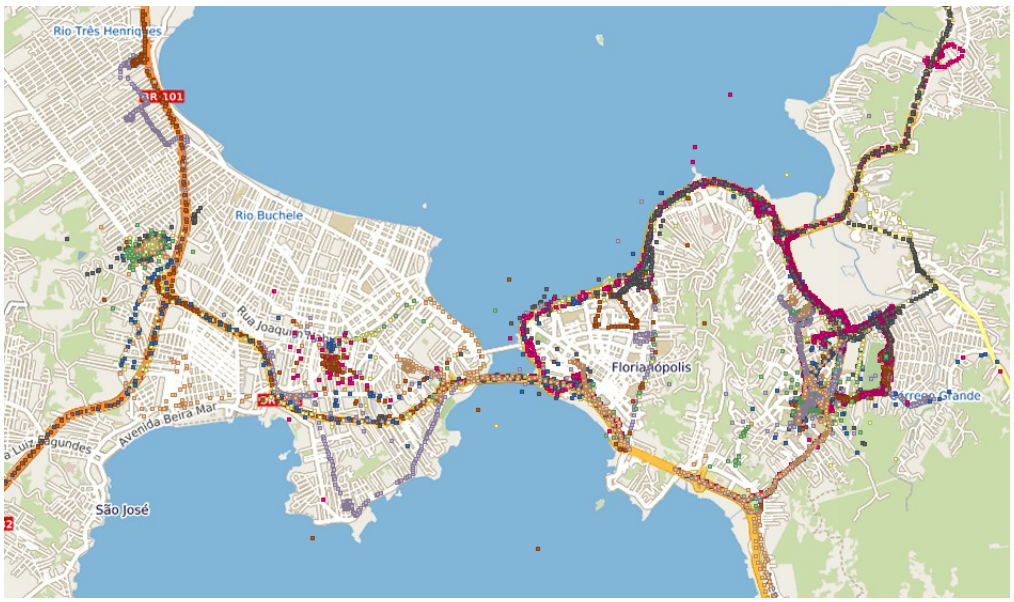

Figura 38 – Trajetórias da Base de Dados Floripa dataset

Microsoft GeoLife[6] é o resultado de um projeto de pesquisa da *Microsoft* na Ásia. Os dados foram coletados por 182 pessoas entre Abril de 2007 e Agosto de 2012. Esses dados cobrem mais de 30 cidades pela China, mas a maior parte se concentra na cidade de Pequim. Essa base de dados está disponível em pastas por usuário e nessas pastas estão arquivos *.plt* com trajetórias, onde cada arquivo corresponde uma trajetória. Nesses arquivos estão dados do par de coordenadas geográficas, data e instante de tempo. Ainda, em algumas dessas pastas, há um arquivo identificando o meio de transporte utilizado pela pessoa. Nesse arquivo há o registro da data, o instante de tempo e o meio de transporte utilizado pela pessoa. Com esse arquivo é possível identificar, em algumas trajetórias, os meios de transporte utilizado e assim agregar mais dados às trajetórias. Entretanto o sistema não suporta carregar esse tipo de arquivo, pois não é algo comum nas bases de dados de trajetórias. Para isso foi desenvolvido um código exclusivo para carregar somente os arquivos de meios de transporte.

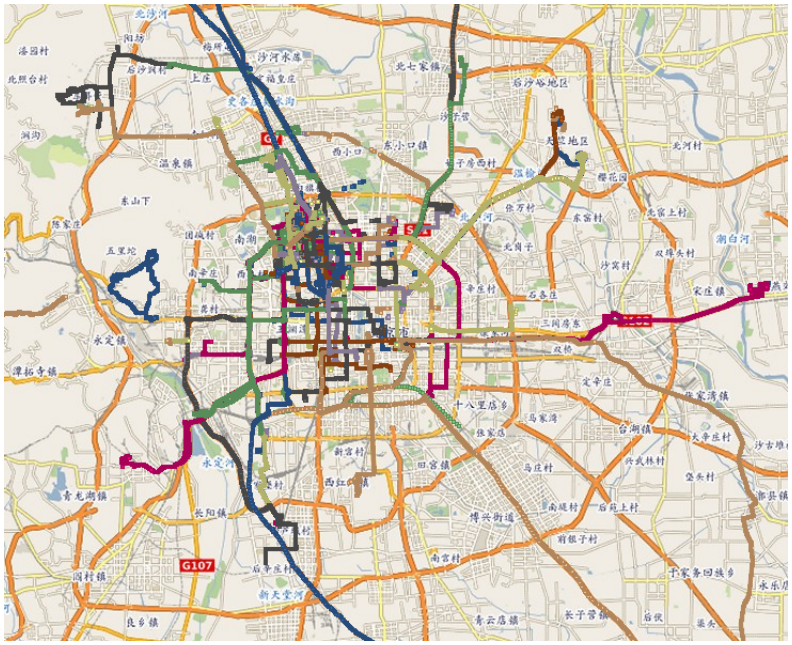

Figura 39 – Trajetórias da Base de Dados Microsoft Geolife

[6] https://www.microsoft.com/en-us/download/details.aspx?id=52367

Greek Trucks[7] são dados de trajetórias de 50 caminhões entregando concreto na cidade de Atenas, na Grécia. Essa é uma base de dados de tamanho reduzido que foi disponibilizada somente com a primeira trajetória de cada caminhão. A base de dados está disponível em arquivo único identificando cada caminhão e sua trajetória de identificador número 1, além é claro dos dados das coordenadas geográficas e instante de tempo.

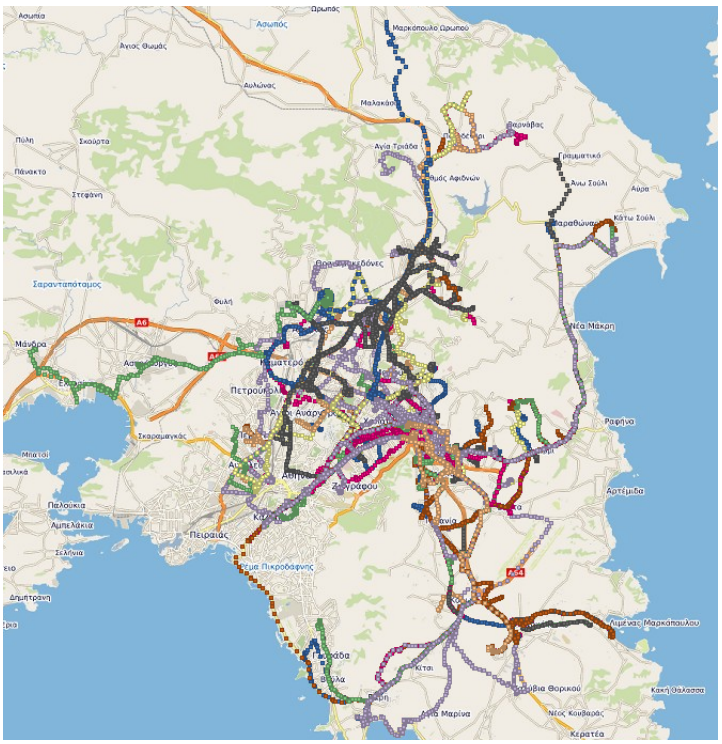

Figura 40 – Trajetórias da Base de Dados Greek Trucks

[7] http://www.chorochronos.org/?q=node/5

Greek Trucks rev[8] é uma versão completa da base de dados Greek Trucks. Nessa versão estão disponíveis todas as trajetórias dos 50 caminhões. Entretanto não é possível identificar os caminhões e sim somente as trajetórias. Nessa base de dados de arquivo único há o identificador da trajetória, par de coordenadas geográficas e instante de tempo.

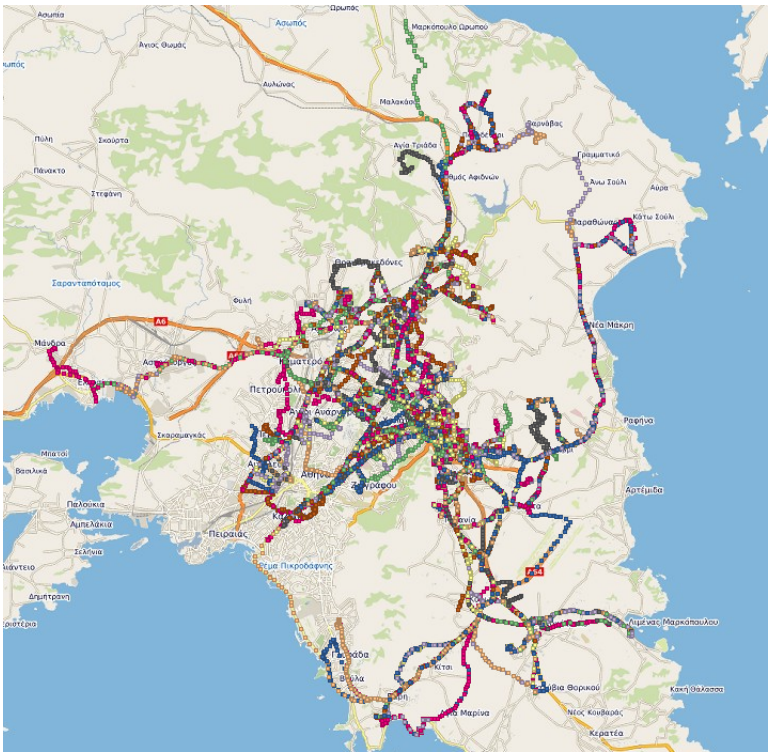

Figura 41 – Trajetórias da Base de Dados Greek Trucks rev

[8] http://www.chorochronos.org/?q=node/10

NYC buses[9] são dados da linha linha B63 de Brooklyn, na cidade de Nova Iorque. Os dados foram coletados pelo sistema de monitoramento de ônibus da cidade e está disponível na internet somente o período entre Abril e Maio de 2011. Os dados estão em arquivo único, identificando 30 diferentes ônibus e não há identificação de suas trajetórias. Para cada registro nessa base está o identificador do ônibus, as coordenadas geográficas, instante de tempo do registro, se o ônibus está em trânsito, distância já percorrida, distância para próxima parada, identificador da próxima parada e identificador da linha.

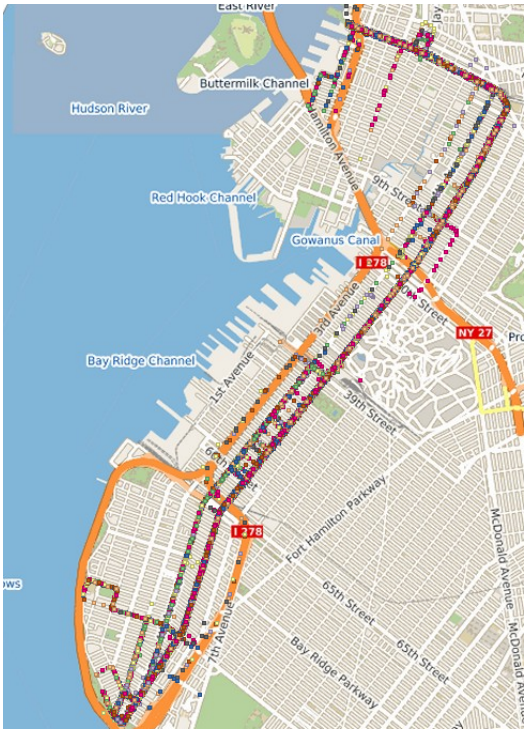

Figura 42 – Trajetórias da Base de Dados NYC buses

[9] http://bustime.mta.info/wiki/Developers/ArchiveData

Taxi Roma[10] são dados de trajetórias de 316 táxis na cidade de Roma, na Itália. Os dados foram coletados por *tablets* dentro dos táxis por um período de 30 dias. O intervalo dos registros por esses *tablets* foi em média de 7 segundos, conforme a documentação. Cada táxi está identificado por um identificador numérico e esse identifica a trajetória de um táxi. Os dados estão disponíveis através de um arquivo WKT com o identificador do táxi, instante de tempo e o ponto com as coordenadas geográficas.

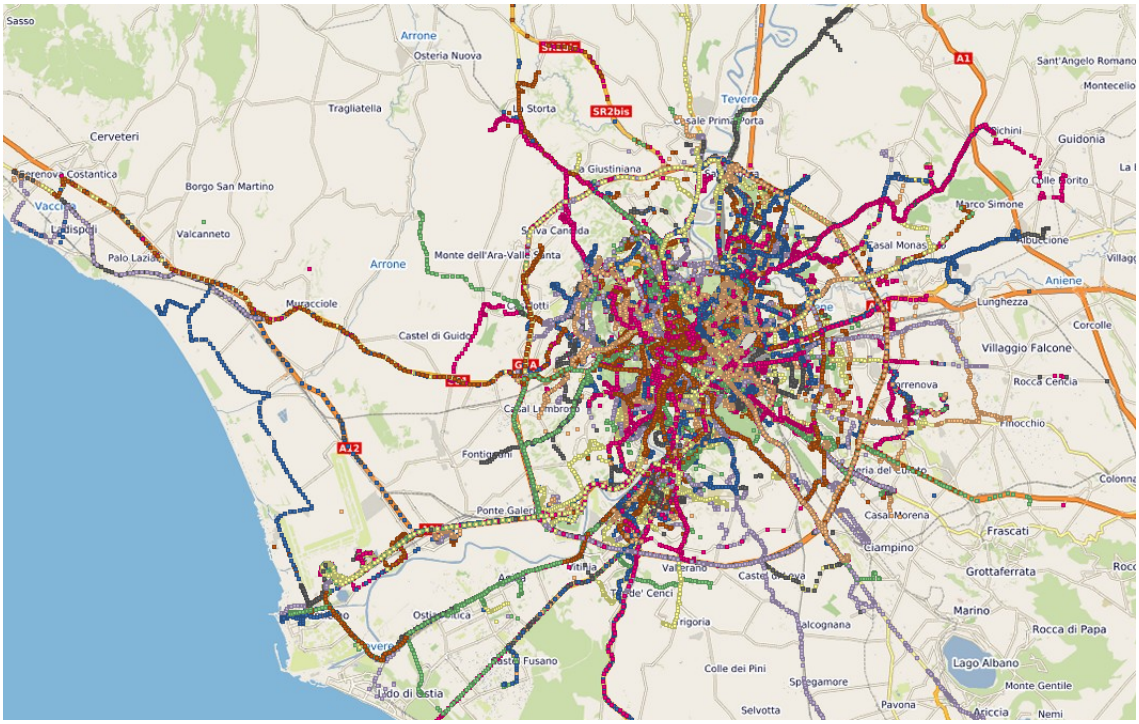

Figura 43 – Trajetórias da Base de Dados Taxi Roma

[10] http://crawdad.org/roma/taxi/20140717/

Taxi San Francisco[11] são trajetórias de 536 táxis na cidade de São Francisco na Califórnia, EUA. A base de dados é formada por 536 arquivos, onde cada arquivo corresponde a trajetória de um único táxi durante os 30 dias de coleta. Em cada arquivo há o par de coordenadas geográficas, instante de tempo e um marcador se o táxi está ocupado ou não. Para esse marcador é utilizado 1 para ocupado e 0 para não ocupado.

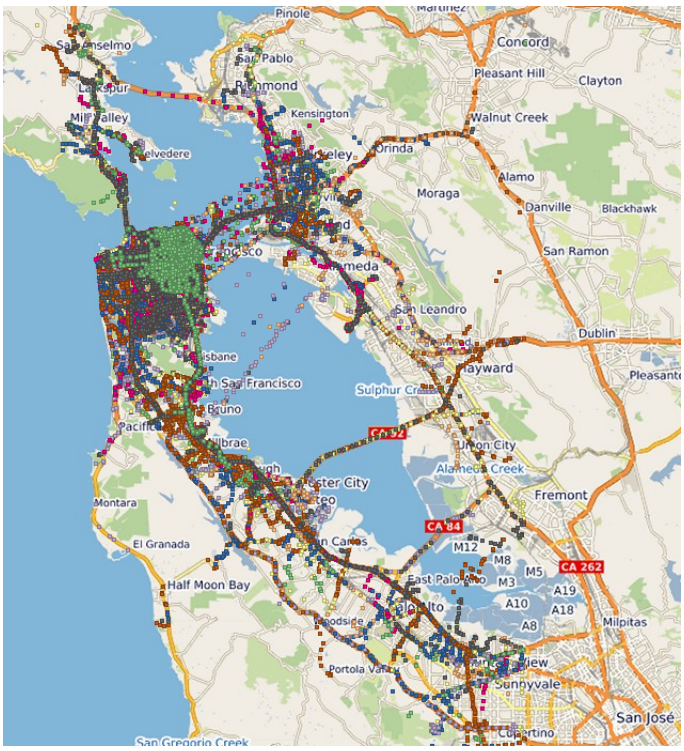

Figura 44 – Trajetórias da Base de Dados Taxi San Francisco

[11] http://www.crawdad.org/epfl/mobility/20090224/

Microsoft T-Drive[12] são dados de trajetórias de táxis na cidade de Pequim, na China. Esses dados foram coletados por 10 357 táxis em um período de 6 dias. Os dados estão disponíveis em arquivos, onde cada arquivo representa a trajetória de um táxi no período coletado. Nesses arquivos há o identificador do táxi, o par de coordenadas geográficas e o instante de tempo. De acordo com a documentação o tempo médio do registro de cada ponto é de 177 segundos. Porém em nossa análise foi obtido 3268.67 segundos, pois considera todos os dias de um táxi como única trajetória.

Figura 45 – Trajetórias da Base de Dados Microsoft T-Drive

[12]https://www.microsoft.com/en-us/research/publication/t-drive-trajectory-data-sample/

Uber San Francisco[13] são dados de trajetórias do serviço de transporte Uber[14] na cidade de São Francisco na Califórnia, EUA. Os dados possuem identificador da trajetória, instante de tempo e o par de coordenadas geográficas. Porém essa base de dados não apresenta o identificador do objeto. Assim há 24 999 trajetórias com um intervalo de tempo de 4 segundos entre seus pontos e, conforme a documentação, não há pontos redundantes nos momentos de parada do veículo.

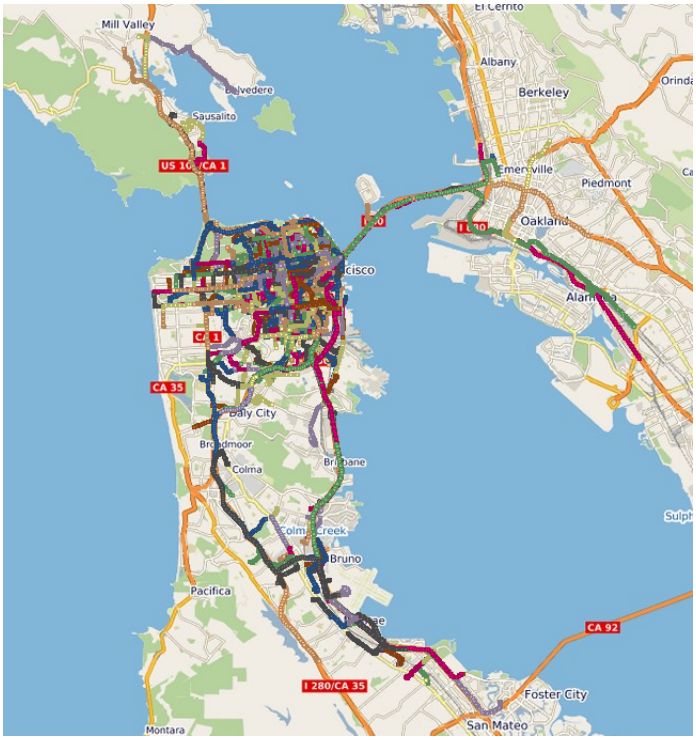

Figura 46 – Trajetórias da Base de Dados Uber San Francisco

[13] http://www.infochimps.com/datasets/uber-anonymized-gps-log
[14] http://www.uber.com

W4M Oldenburg[15] são dados sintéticos gerados através do gerador Thomas-Brinkhoff[16].

A base de dados possuí 100 000 trajetórias em um período de 24 horas na cidade de Oldemburgo, na Alemanha e possui intervalo de tempo entre os pontos de 10 minutos. Os dados dessa base possuem identificador da trajetória, instante de tempo e par de coordenadas geográficas. Entretanto nessa base de dados há uma característica diferente de todas as outras utilizadas nesse trabalho. É utilizado um número inteiro para definir o instante de tempo, ou seja, o primeiro instante de tempo é zero 0, o intervalo seguinte é 1 e assim por diante.

Figura 47 – Trajetórias da Base de Dados W4M Oldenburg

[15] http://kdd.isti.cnr.it/W4M/
[16] www.fh-oow.de/institute/iapg/personen/brinkhoff/generator/

W4M Milano[17] é uma base de dados com 45 000 trajetórias de aproximadamente 17 000 veículos na cidade de Milão na Itália. Os dados são da primeira semana de Abril de 2007 e possuem identificador da trajetória, instante de tempo e par de coordenadas geográficas. Um detalhe dessa base dados é que não possui identificador dos veículos, impossibilitando assim a identificação de trajetórias de um mesmo veículo.

Figura 48 – Trajetórias da Base de Dados W4M Milano

[17] http://kdd.isti.cnr.it/W4M/

Para uma discussão das técnicas de pré-processamento empregadas pelos sistema foram então selecionadas duas bases de dados como estudos de caso. A escolha pelas bases Geolife (ZHENG et al., 2008) e Taxi San Francisco (PIORKOWSKI; SARAFIJANOVIC-DJUKIC; GROSSGLAUSER, 2009) ocorreu devido serem amplamente utilizadas na literatura (ZHENG et al., 2009; AMICI et al., 2014; SHEN et al., 2017; KIERMEIER; WERNER, 2017), possuírem grande volume de registros, estarem disponíveis publicamente e possuírem características distintas, pois foram coletadas por diferentes objetos móveis. Ainda entre as características próprias de cada base, vale ressaltar que GeoLife possui maior densidade de pontos, por ter o tempo médio de coleta dos pontos em 18 segundos. Essa base de dados ainda abrange uma quantidade maior de meios de transportes, como caminhadas, ônibus, carros e aviões, pois são registros dos movimentos das pessoas em seu cotidiano. Enquanto a base Taxi San Francisco é uma base de dados inteiramente formada por trajetórias de táxis. Essa base de dados ainda possui o estado de ocupação do táxi, que identifica se o táxi está em um trajeto com o cliente ou não e também possui os pontos mais esparsos, pois o intervalo de coleta médio entre os pontos é de 125 segundos.

Assim essas duas bases de dados serão utilizadas em estudos de caso nas cinco seções seguintes:

1. Carregamento de Dados de Trajetórias (Seção 4.1)
2. Segmentação de Trajetórias (Seção 4.2)
3. Limpeza de Trajetórias (Seção 4.3)
4. Seleção de Trajetórias Próximas a um Ponto (Seção 4.4)
5. Exportação de Dados do Banco de Dados (Seção 4.5)

4.1 CARREGAMENTO DE DADOS DE TRAJETÓRIAS

Nessa seção serão apresentadas as etapas para carregamento das bases de dados Geolife (ZHENG et al., 2008) e Taxi San Francisco (PIORKOWSKI; SARAFIJANOVIC-DJUKIC; GROSSGLAUSER, 2009). Junto com essas bases de dados foi disponibilizado uma documentação onde estão descritos os dados e como esses foram coletados. Ainda assim uma rápida análise através dos diretórios e arquivos dessas bases de dados confirmou as informações da documentação. Dessa forma obteve-se maior compreensão da estrutura dos dados, bem como detalhes de delimitadores do arquivo, formato de data e todo conjunto de arquivos da base de dados. Com o devido conhecimento sobre o formato e estrutura dos arquivos foi então iniciado o processo de carregamento das base de dados para o banco de dados.

4.1.1 ESTUDO DE CASO 1 - GEOLIFE

A base de dados GeoLife foi coletada por 182 pessoas entre Abril de 2007 e Agosto de 2012. As suas trajetórias foram registradas e disponibilizadas como um conjunto de 182 diretórios. Onde cada diretório representa uma pessoa e possuí arquivos de suas trajetórias. Cada arquivo de trajetória corresponde a uma única trajetória da pessoa. Ainda em alguns diretórios é possível encontrar um arquivo de anotação de meios de transportes, onde está registrado o meio de transporte utilizado, com data e horário. A Figura 49 apresenta a estrutura de arquivos de trajetórias, onde cada pasta numerada é referente a pessoa e cada arquivo de extensão *.plt* é uma trajetória dessa pessoa. Com base nessa estrutura de arquivos é então iniciado o processo de carregamento dos dados.

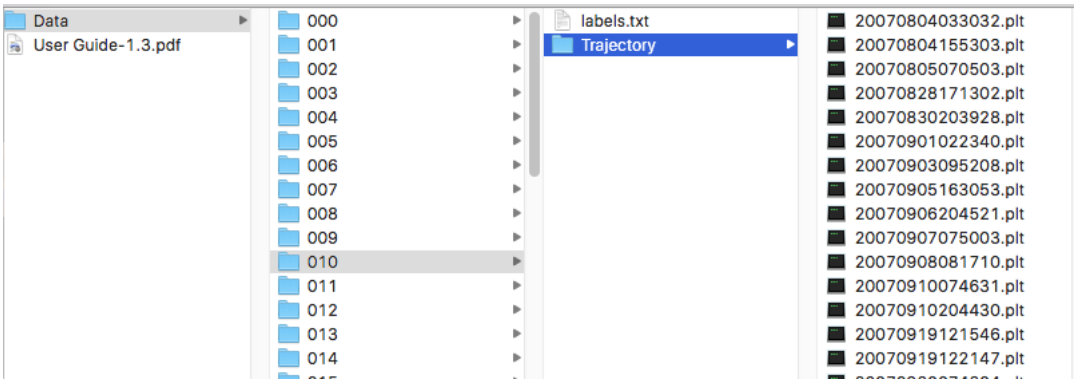

Figura 49 – Estrutura de pastas e arquivos da base de dados GeoLife

```
1  Geolife trajectory
2  WGS 84
3  Altitude is in Feet
4  Reserved 3
5  0,2,255,My Track,0,0,2,8421376
6  0
7  39.984702,116.318417,0,492,39744.1201851852,2008-10-23,02:53:04
8  39.984683,116.31845,0,492,39744.1202546296,2008-10-23,02:53:10
9  39.984686,116.318417,0,492,39744.1203125,2008-10-23,02:53:15
10 39.984688,116.318385,0,492,39744.1203703704,2008-10-23,02:53:20
```

Figura 50 – Trecho de dados de um arquivo da base de dados GeoLife

A Figura 50 ilustra a estrutura dos dados nos arquivos de trajetórias. Podemos perceber que o formato de data e hora é *yyyy-MM-dd HH:mm:ss*, o delimitador das colunas nos arquivos é por vírgula, todos os arquivos possuem um cabeçalho com 6 linhas e o modelo de referência geográfico é o WGS84(4326). A documentação ainda descreve que a data está na Coluna 6, a hora na Coluna 7, a latitude e longitude nas Colunas 1 e 2 respectivamente. As outras colunas do arquivo não são relevantes para esse carregamento, pois são dados de altitude e data em outro formato.

A partir do levantamento dessas informações é então utilizada a tela ilustrada pela Figura 51 para o carregamento dos dados. Nessa tela é estabelecido o diretório raiz da base de dados, descrito o formato de data e hora utilizado. Ainda é definido o delimitador de coluna do arquivo e especificado a conversão do modelo de referência geográfico de WGS84(4326) para WGS84 *Web Mercator*(900913), pois esse é um modelo compatível com mapas *online*. Ainda é declarado para ignorar 6 linhas no início dos arquivos e detalhado para ignorar os arquivos com extensões *.pdf* e *.txt*, pois os arquivos de trajetórias nessa base somente possuem extensão *.plt*. Também é apontado para criar um identificador(GID) para cada registro no banco de dados e um identificador(TID) para cada arquivo(trajetória). A opção por salvar metadados cria um identificador único para cada diretório e permite a identificação de cada pessoa. Também foi definido os nomes de colunas e associadas com suas respectivas colunas nos arquivos de origem das trajetórias. Por fim esses dados foram gravados no banco de dados na tabela criada de nome *geolife*, conforme declarado na tela.

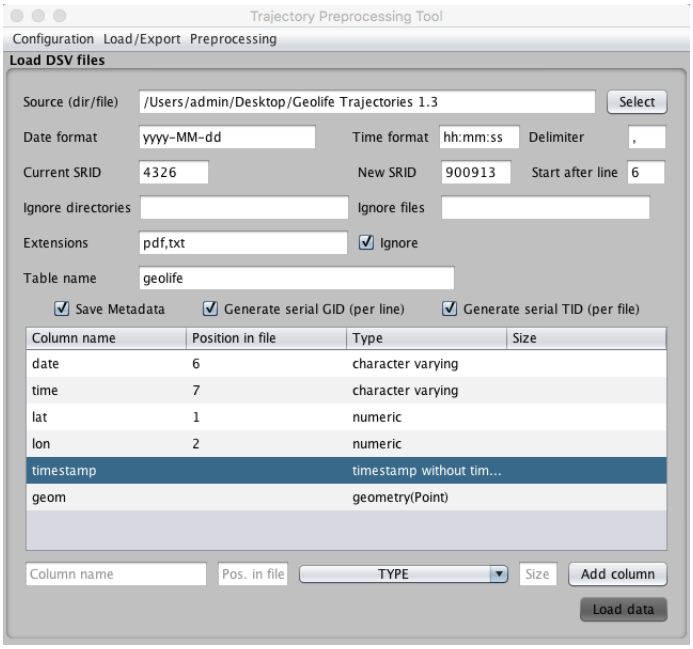

Figura 51 – Tela do sistema com informações para carregamento dos dados da base de dados GeoLife

A Figura 52 ilustra os dados da base de dados GeoLife inseridos na tabela *geolife* no banco de dados. Detalhe para a coluna *gid* e *tid* que foram geradas pelo sistema, e ainda a coluna *folder_id*, que identifica cada diretório da base de dados representando uma pessoa.

	gid integer	tid integer	date character var	time character v	lat numeric	lon numeric	timestamp timestamp without time z	geom geometry(Point)	path character varying(150)	folder_id integer
1	3238207	1926	2009-01-16	09:41:01	39.975228	116.36663	2009-01-16 09:41:01	0101000020 31BF0D0037D8E63F22B568413AE67DC8618C5241	/Users/rogerjames/Downloads/	14
2	3238208	1926	2009-01-16	09:41:03	39.97523	116.36663	2009-01-16 09:41:03	0101000020 31BF0D0037D8E63F22B56841F0ED15DB618C5241	/Users/rogerjames/Downloads/	14
3	3238209	1926	2009-01-16	09:41:08	39.975231	116.366649	2009-01-16 09:41:08	0101000020 31BF0D002D8095832ZB56841CFF161E4618C5241	/Users/rogerjames/Downloads/	14
4	3238210	1926	2009-01-16	09:41:11	39.97523	116.366663	2009-01-16 09:41:11	0101000020 31BF0D00AD8274B52ZB56841F0ED15DB618C5241	/Users/rogerjames/Downloads/	14
5	3238211	1926	2009-01-16	09:41:16	39.97523	116.366673	2009-01-16 09:41:16	0101000020 31BF0D009CCD13D92ZB56841F0ED15DB618C5241	/Users/rogerjames/Downloads/	14

Figura 52 – Registros da base de dados GeoLife no banco de dados PostgreSQL

Ao final do carregamento foi realizado uma consulta no banco de dados, ilustrado pela Figura 53, para confirmar se os dados carregados estão de acordo com a documentação da base de dados. Assim os números na consulta ao banco de dados conferem com a documentação da base de dados, com 24 876 978 registros de pontos em 18 670 trajetórias.

```
select count(gid) as Quantidade_gid, count(distinct(tid)) as Quantidade_tid from geolife
```

Output pane

Data Output | Explain | Messages | History

	quantidade_gid bigint	quantidade_tid bigint
1	24876978	18670

Figura 53 – Consulta no banco de dados PostgresSQL apresentando a quantidade de registros e trajetórias da base de dados GeoLife

4.1.2 ESTUDO DE CASO 2 - TAXI SAN FRANCISCO

A base de dados Taxi San Francisco é formada por trajetórias de 536 táxis pelo período de 30 dias em Maio de 2008. A base está disponível em forma de um conjunto de arquivos, onde cada arquivo corresponde a trajetória de um táxi durante o período. Assim são 536 arquivos referentes as trajetórias, mais o arquivo de documentação da base de dados e um arquivo com a data de atualização de cada trajetória, conforme ilustrado pela Figura 54.

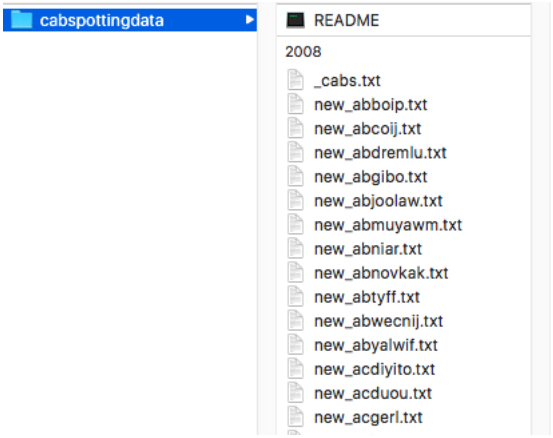

Figura 54 – Estrutura de arquivos da base de dados Taxi San Francisco

```
The format of each mobility trace file is the following -
each line contains [latitude, longitude, occupancy, time], e.g.: [37.75134 -122.39488 0 1213084687],
where latitude and longitude are in decimal degrees,
 occupancy shows if a cab has a fare (1 = occupied, 0 = free) and time is in UNIX epoch format.
```

Figura 55 – Documentação da base de dados Taxi San Francisco

A documentação da base de dados Taxi San Francisco, ilustrada pela Figura 55, informa a existência de quatro dados em cada registro. Esses dados são a latitude, longitude, ocupação do táxi e o instante de tempo do registro, conforme ilustra a Figura 56. Nessa documentação ainda é descrito que o par de coordenadas geográficas estão em graus decimais, a data e horário estão no padrão *unix timestamp* e, por fim, o dado de ocupação do táxi informando se o táxi está ocupado ou não. Para informar a ocupação do táxi é utilizado o número 1 para ocupado e 0 para livre. A Figura 56 ilustra um trecho dos dados de um dos arquivos de trajetória. Nesse trecho é possível perceber o dado que representa a ocupação do táxi e também o delimitador desses dados, que nesse caso é o caractere de espaço em branco.

```
37.77514 -122.43646 0 1213082663
37.77275 -122.43732 0 1213082602
37.77209 -122.43716 0 1213082601
37.77208 -122.43719 1 1213082570
37.77134 -122.43661 1 1213082522
37.77086 -122.43018 1 1213082409
```

Figura 56 – Trecho de dados em um arquivo da base de dados Taxi San Francisco

A tela do sistema utilizada para o carregamento da base de dados Taxi San Francisco está ilustrado pela Figura 57. Nessa tela foi necessário aplicar um espaço em branco, com o teclado, para fornecer ao sistema o delimitador dos dados. Foi também informado o modelo de referência geográfica 4326 e os arquivos a serem ignorados, *_cabs.txt* e *README*, esses que são os arquivos com as datas de atualização e documentação respectivamente. A estrutura dos dados é caracterizada por cada táxi possuir uma trajetória no período e essa está em arquivo único, dessa forma a opção de gerar o identificador(TID) por arquivo permite identificação de cada táxi e toda a sua trajetória. Em seguida, na tela, são associadas as colunas da tabela a serem criadas com as colunas do arquivo a ser carregado. A latitude e longitude estão na coluna 1 e 2, a data na coluna 4 do arquivo origem está no formato *unix timestamp*, então será identificada pelo sistema e convertida para o padrão *yyyy-MM-dd HH:mm:ss* do banco de dados PostgreSQL. Ainda foi adicionado uma coluna de ocupação para registrar a ocupação do táxi, essa dado é relacionado com a coluna 3 do arquivo origem. Por fim os dados serão carregados na tabela de nome *taxi_sanfrancisco* criada no banco de dados.

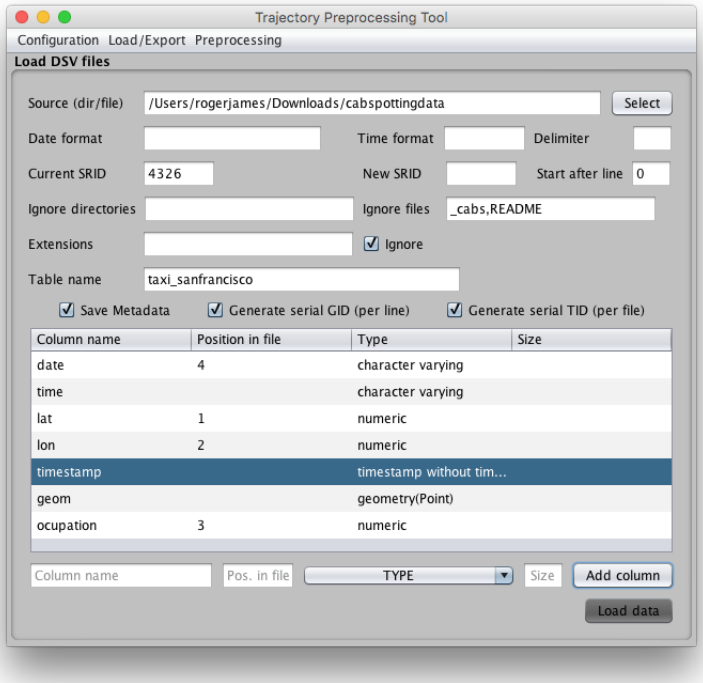

Figura 57 – Tela do sistema com informações para carregamento da base de dados Taxi San Francisco

A Figura 58 ilustra os dados inseridos na tabela de nome *taxi_ sanfrancisco* criada no banco de dados. Além disso podemos perceber na Figura a data convertida a partir do *unix timestamp* e a troca da ocupação do táxi, de 0 para 1.

10	23486	1	1211035610	37.7514	-122.39496	2008-05-17 11:46:50	0101000020 31B	0	/Users/r	1
11	23485	1	1211035670	37.75141	-122.39497	2008-05-17 11:47:50	0101000020 31B	0	/Users/r	1
12	23484	1	1211035732	37.7514	-122.39496	2008-05-17 11:48:52	0101000020 31B	0	/Users/r	1
13	23483	1	1211035796	37.75067	-122.39533	2008-05-17 11:49:56	0101000020 31B	0	/Users/r	1
14	23482	1	1211035844	37.74978	-122.39709	2008-05-17 11:50:44	0101000020 31B	0	/Users/r	1
15	23481	1	1211035870	37.74977	-122.39724	2008-05-17 11:51:10	0101000020 31B	1	/Users/r	1
16	23480	1	1211035931	37.74896	-122.40619	2008-05-17 11:52:11	0101000020 31B	1	/Users/r	1
17	23479	1	1211036022	37.74831	-122.41335	2008-05-17 11:53:42	0101000020 31B	1	/Users/r	1
18	23478	1	1211036082	37.75157	-122.414	2008-05-17 11:54:42	0101000020 31B	1	/Users/r	1

Figura 58 – Registros da base de dados Taxi de San Francisco no banco de dados PostgresSQL

Encerrado o processo de carregamento, foi então executada uma consulta no banco de dados, ilustrada pela Figura 59, para conferência dos registros realizados. Essa consulta permitiu comparar a quantidade de registros com a documentação da base de dados. Assim os números dessa base de dados são 11 219 955 registros de 536 táxis.

```
select count(gid) as Quantidade_gid, count(distinct(tid)) as Quantidade_tid from taxi_sanfrancisco
```

Output pane

Data Output | Explain | Messages | History

	quantidade_gid bigint	quantidade_tid bigint
1	11219955	536

Figura 59 – Consulta no banco de dados PostgresSQL apresentando a quantidade de registros e trajetórias da base de dados Taxi San Francisco

As segmentações das bases de dados GeoLife e Taxi San Francisco foram feitas através do sistema desenvolvido por esse trabalho, conforme a Seção 3.6. Essas segmentações ocorreram a partir de tabelas com trajetórias em um banco de dados PostgreSQL. Para a segmentação por intervalo de tempo foi utilizado um intervalo de 5 minutos em ambas as bases. Esse intervalo foi escolhido por ser um tempo superior ao tempo médio encontrado em ambas as bases, também por ser um tempo aceitável para a troca de um meio de transporte por uma pessoa e inclusive um táxi trocar seu estado de ocupado para não ocupado. Sendo assim a base de dados Taxi San Francisco também foi segmentada pelo estado de ocupação do táxi, pois possuí tal atributo que permite identificar quando o táxi está ocupado ou não.

4.2.1 ESTUDO DE CASO 1 - GEOLIFE

A base de dados GeoLife possui originalmente 18 670 trajetórias de 182 pessoas. Essas trajetórias foram submetidas à segmentação por intervalo de tempo em 5 minutos. Ou seja, sempre que houver um tempo igual ou superior a 5 minutos entre um ponto e o ponto seguinte, então o sistema irá considerar a partir do ponto seguinte como uma outra trajetória. Dessa forma é esperado que aumente a quantidade de trajetórias da base de dados e reduza grandes espaços temporais entre os pontos dessa trajetória.

A Figura 60 ilustra uma trajetória da base de dados Geolife, onde é possível perceber grande espaçamento entre diferentes partes da trajetória.

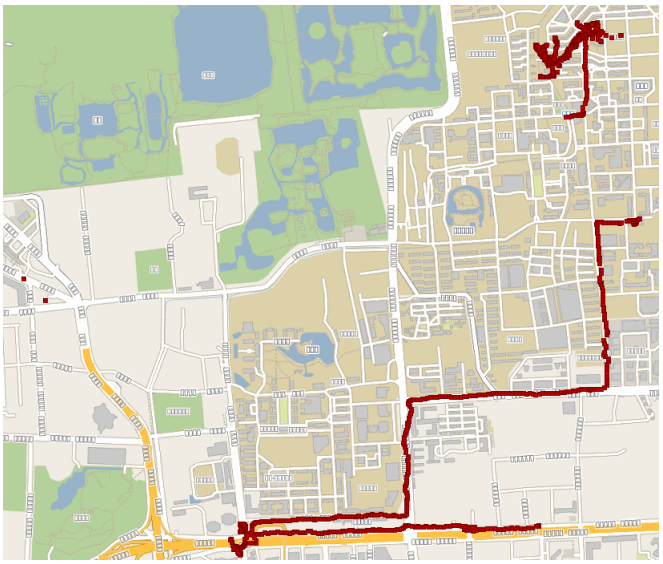

Figura 60 – Uma trajetória com grandes intervalos na base de dados GeoLife

Entretanto os espaços na trajetória são eliminados através do processo de segmentação, criando assim trajetórias menores, conforme ilustrado pela Figura 61. Dessa forma a segmentação na base de dados Geolife aumentou o número de trajetórias da base de dados para 58 482. Com a segmentação em intervalos de 5 minutos na base de dados Geolife foi então possível eliminar grandes intervalos de tempo no meio de suas trajetórias e assim reduzir o tempo médio entre seus pontos de 18.19 segundos para 8.74 segundos. Em relação a trajetória ilustrada pela Figura 60, essa trajetória foi então segmentada em 7 segmentos com novos identificadores, caracterizando assim as 7 trajetórias em cores distintas ilustradas pela Figura 61.

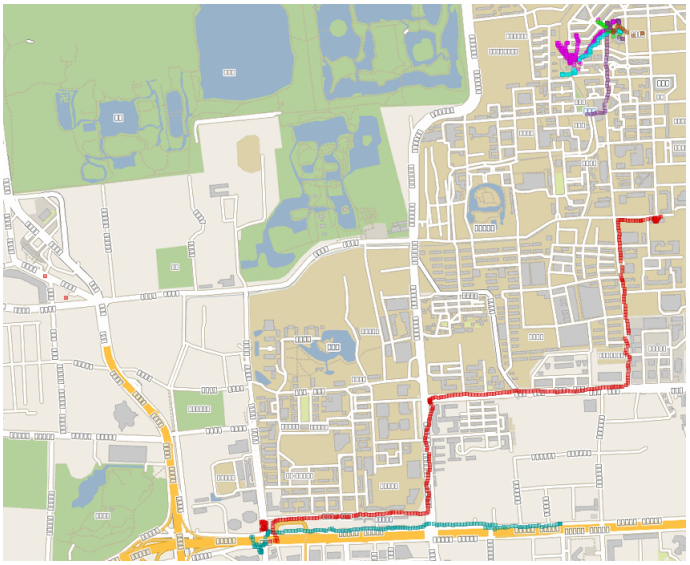

Figura 61 – Segmentos de uma trajetória da base de dados Geolife após processo de segmentação por tempo de 5 minutos

4.2.2 ESTUDO DE CASO 2 - TAXI SAN FRANCISCO

A base de dados Taxi San Francisco possui registros de trajetórias de 536 táxis em um período de 30 dias, onde cada trajetória é todo o percurso do táxi nesse período. Sendo assim são esperadoz intervalos de tempo durante a trajetória, porque os motoristas realizam intervalos e encerram suas jornadas de trabalho. Essa base de dados ainda conta com um atributo de ocupação do táxi, que identifica se o táxi está ocupado ou não. Dessa forma é então possível segmentar as trajetórias de duas maneiras: de acordo com os intervalos de tempo nas trajetórias ou através do estado de ocupação do táxi, obtendo assim trajetórias onde o táxi está ocupado e trajetórias que o táxi está livre.

As 536 trajetórias originais foram então segmentadas pelos dois critérios, intervalo de tempo de 5 minutos e ocupação do táxi. Com a segmentação por intervalos de tempo de 5 minutos foi possível obter 98 525 trajetórias e reduzir o tempo médio entre os pontos de 125.75 para 64.58 segundos. Já a segmentação pelo estado de ocupação do táxi atingiu 928 301 trajetórias e obteve tempo médio entre os pontos de 76.45 segundos

A Figura 62 ilustra uma das 536 trajetórias da base de dados. Essa trajetória possui 23 495 pontos e são referentes ao percurso de um táxi nos 30 dias de coleta dos dados.

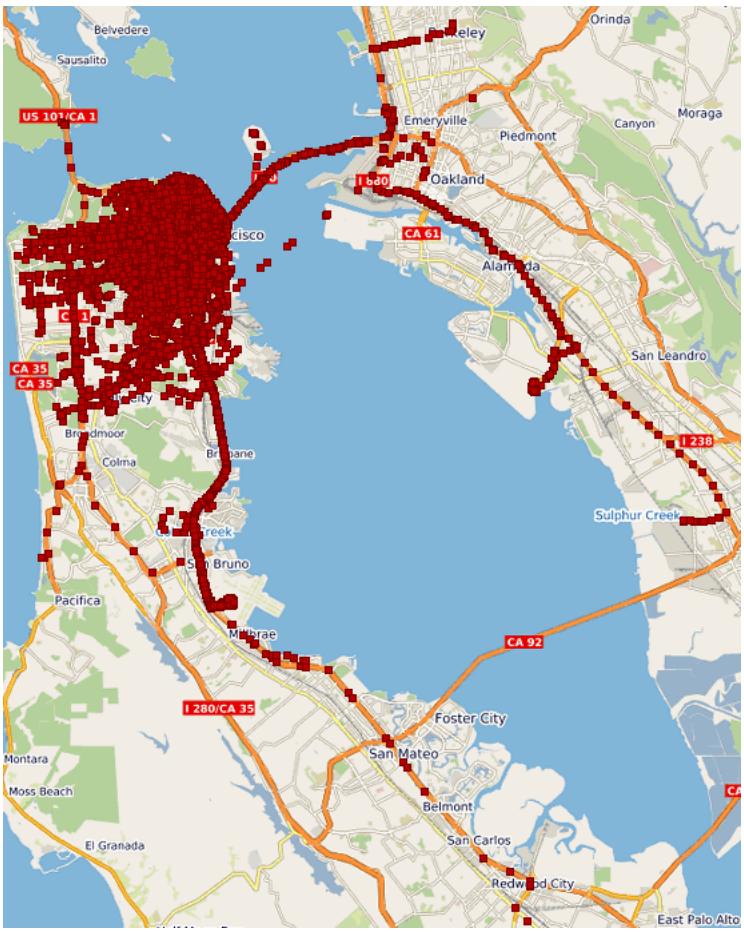

Figura 62 – Uma trajetória da base de dados Taxi San Francisco

A trajetória ilustrada pela Figura 62 foi então segmentada por intervalos de tempo de 5 minutos. Nessa segmentação foi possível eliminar intervalos de tempo igual ou superior a 5 minutos e assim formando 135 trajetórias a partir dos segmentos da trajetória original, conforme ilustrado pela Figura 63, onde há uma cor diferente para cada trajetória.

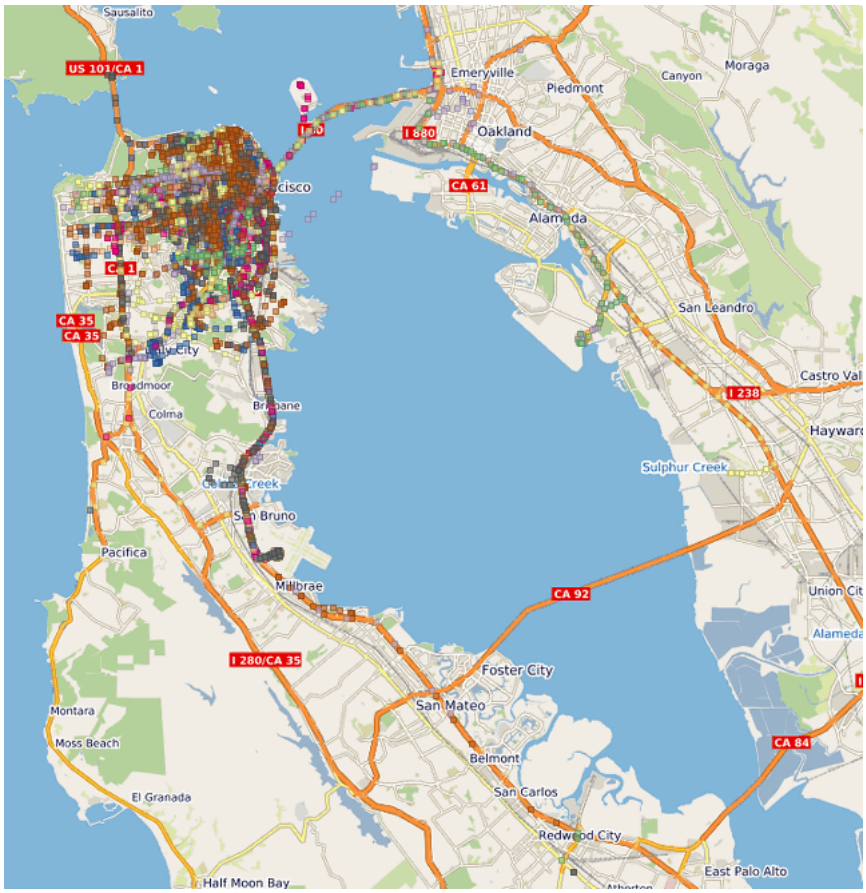

Figura 63 – Segmentos de uma trajetória da base de dados Taxi San Francisco após processo de segmentação por tempo de 5 minutos

Ainda assim a mesma trajetória original ilustrada pela Figura 62 foi segmentada pelo atributo de ocupação, criando então 1 915 trajetórias, atingindo uma média de 63 trajetórias por dia. O resultado dessa segmentação é ilustrada pela Figura 64, onde foi aplicada uma cor diferente para cada trajetória.

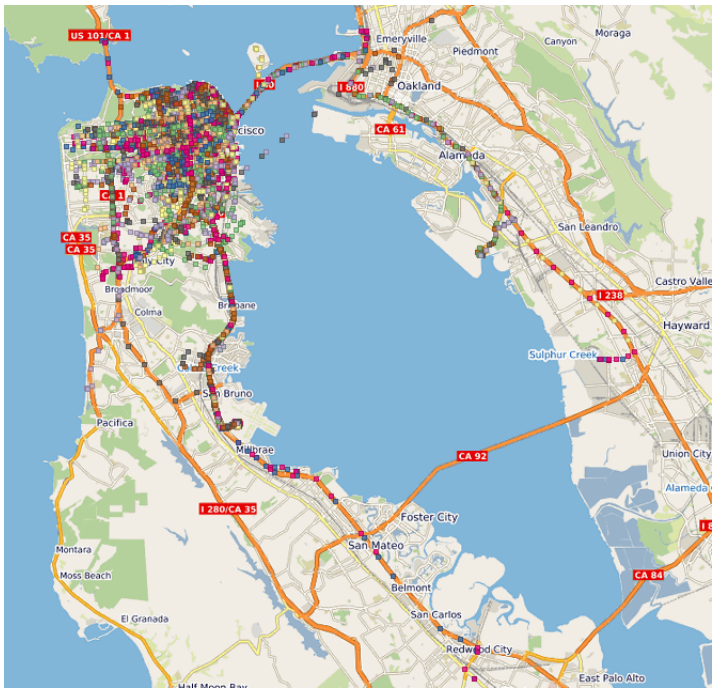

Figura 64 – Segmentos de uma trajetória da base de dados Taxi San Francisco após processo de segmentação por estado de ocupação do táxi

4.2.3 TEMPO MÉDIO ENTRE PONTOS DA TRAJETÓRIA

A definição da necessidade em segmentar uma base de dados, bem como decidir qual intervalo de tempo utilizar, exige uma breve análise dos dados. Essa análise pode obter o intervalo de tempo médio entre os pontos e assim ajudar na decisão em segmentar a base de dados, bem como servir de referência para definição do intervalo de tempo a ser utilizado pela segmentação. Intervalos de poucos segundos ou minutos são os mais utilizados em trajetórias. Já uma média de tempo alta entre os pontos em uma base de dados pode indicar grandes espaçamentos nas trajetórias. Para verificar esse efeito foi então aplicado a segmentação em todas as bases de dados utilizadas nesse trabalho. Das bases de dados utilizadas nesse trabalho, as bases AIS Brest e W4M Milano são as que possuíam os maiores intervalos de tempo médio entre os pontos. Sendo AIS Brest com o maior tempo médio de 57 096,31 segundos e seguido por W4M Milano, com tempo médio de 4 998,91 segundos. Dessa forma essas bases foram segmentadas utilizando intervalo de tempo de 5 minutos. A base de dados AIS Brest reduziu o tempo médio entre os pontos de 57 096,31 segundos para 125,07 e a base de dados W4M Milano reduziu de 4 998,91 segundos para 149,45 segundos.

4.3.1 ESTUDOS DE CASO 1 E 2 - GEOLIFE E TAXI SAN FRANCISCO

Essa seção apresenta o processo de limpeza das trajetórias com ruídos das bases de dados GeoLife e Taxi San Francisco. Para esse processo foi utilizado o módulo de limpeza de trajetórias do sistema desenvolvido, conforme descrito na Seção 3.5. Ainda foram selecionadas manualmente 5 trajetórias com nítidos ruídos em cada base de dados, totalizando assim 10 trajetórias com ruídos. As trajetórias selecionadas da base de dados Geolife estão ilustradas pela Figura 65 e as trajetórias da base Taxi San Francisco pela Figura 68.

Nesse intuito foram aplicadas nas trajetórias as técnicas de limpeza disponíveis no sistema desenvolvido. Entretanto, devido as características das trajetórias, nem todas as técnicas disponíveis no sistema fazem sentido para todas as bases de dados. Por exemplo, a base de dados GeoLife possui maior densidade de pontos em suas trajetórias, devido ao intervalo médio entre os pontos de 18,19 segundos. Já Taxi San Francisco possui os pontos mais esparsos devido ao intervalo médio de 125,75 segundos. Assim faz mais sentido o uso da técnica DBSCAN, descrita na Seção 2.3, na base de dados GeoLife do que na base de dados Taxi San Francisco. Visto que a técnica DBSCAN busca uma densidade de pontos dentro de determinado raio.

A Figura 65 ilustra, em diferentes cores, as 5 trajetórias com ruídos da base de dados GeoLife. Os ruídos nessas trajetórias estão apontados por setas.

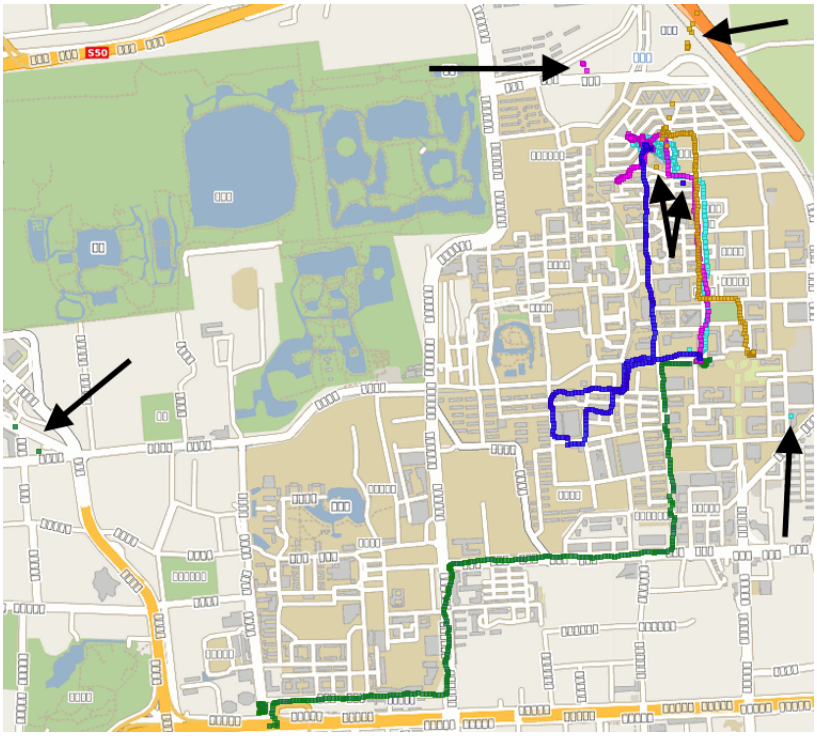

Figura 65 – Ruídos apontados por setas em trajetórias da base de dados Geolife

As trajetórias foram limpas pelas diferentes técnicas fornecidas pelo sistema, utilizando 4 parâmetros diferentes. O resultado desse processo é ainda discutido nessa seção. Entretanto foram selecionadas as duas técnicas que alcançaram melhores resultados na base de dados GeoLife para serem apresentadas visualmente. A Figura 66 ilustra as cinco trajetórias limpas através da técnica DBSCAN com parâmetro de 5 vizinhos em uma vizinhança com raio de 100 metros. Podemos notar que nem todos os ruídos foram removidos, sendo os ruídos remanescentes ainda apontados por setas, conforme ilustra a Figura 66.

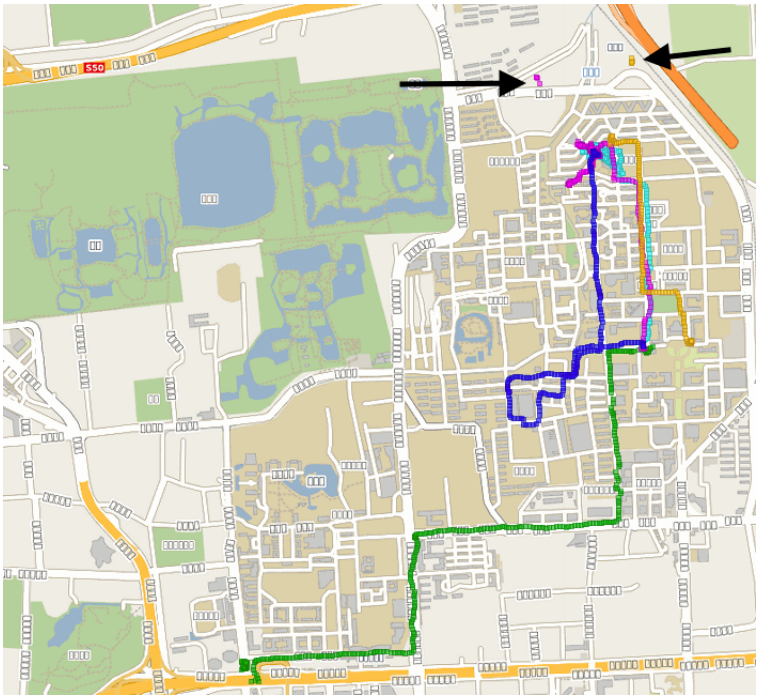

Figura 66 – Trajetórias da base de dados Geolife com ruídos remanescente, apontados por setas, após remoção de ruídos por densidade com DBSCAN

Já a Figura 67 ilustra as trajetórias que foram limpas através da técnica de remoção de pontos acima de determinada velocidade. Os parâmetros utilizados nesse processo foi deletar o primeiro ponto quando a velocidade, entre esse e o ponto seguinte, for maior ou igual a 150km/h. Ainda na Figura 67 podemos perceber alguns ruídos que não foram removidos. Os ruídos excedentes não foram removidos pois possuem distância e velocidade, em relação ao restante da trajetória, dentro dos parâmetros utilizados.

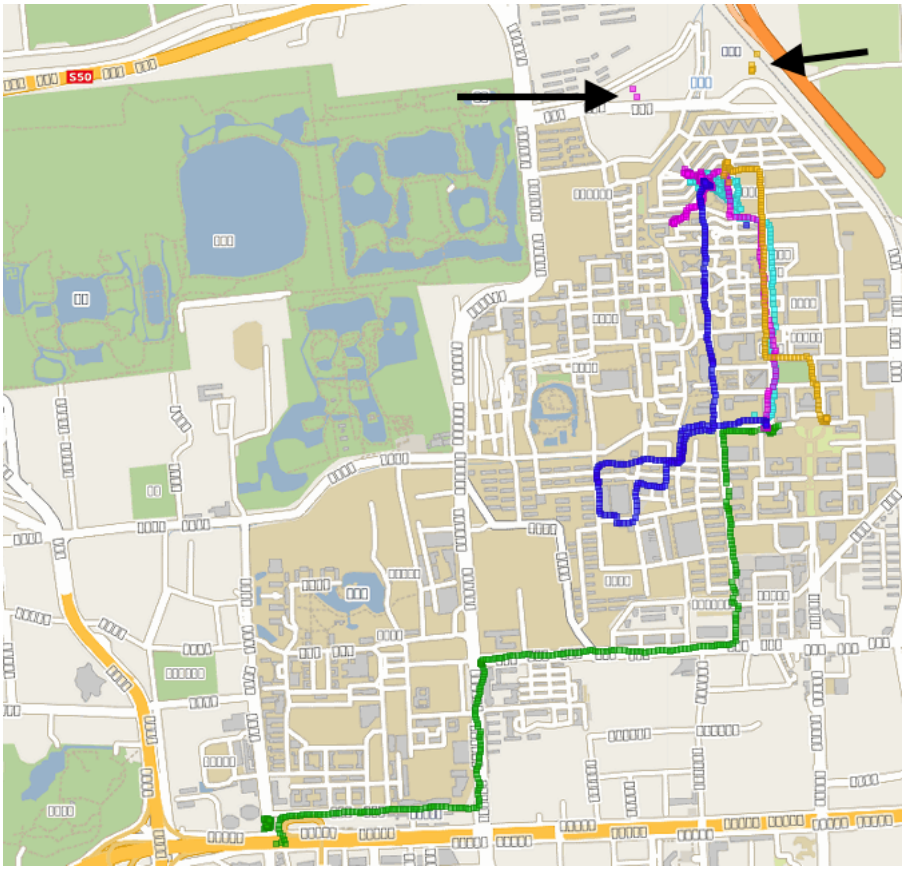

Figura 67 – Trajetórias da base de dados Geolife com ruídos remanescente, apontados por setas, após remoção de ruídos por velocidade superior a 150km/h

As trajetórias com ruídos da base de dados Taxi San Francisco estão ilustradas pela Figura 68. Podemos perceber, em cores diferentes, as 5 trajetórias com os ruídos devidamente apontados pelas setas. Nessa base de dados é notável um padrão dos ruídos nas trajetórias. Esse padrão de ruídos alinhados geralmente é formado no inicio da coleta quando o sensor GPS está realizando a sincronia com os satélites.

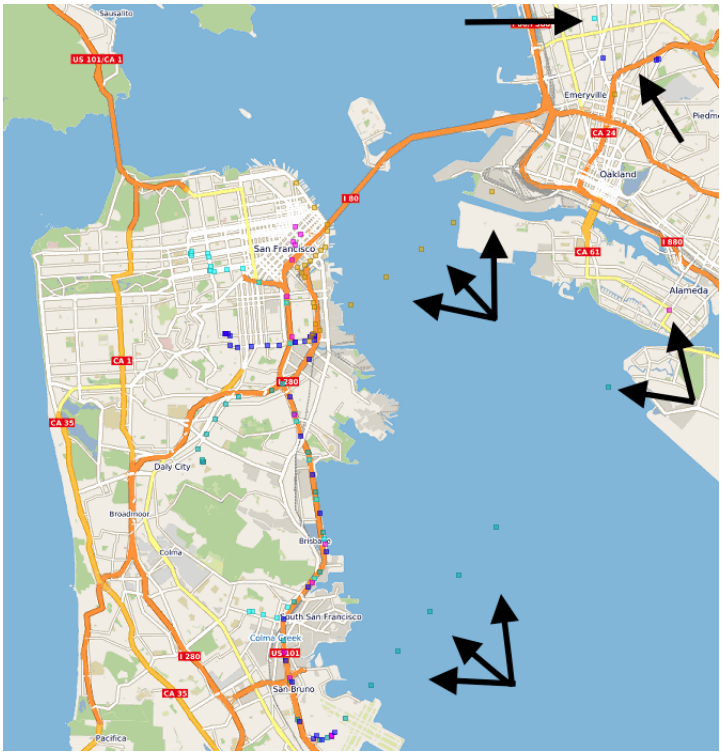

Figura 68 – Ruídos apontados por setas em trajetórias da base de dados Taxi San Francisco

Para a limpeza das trajetórias da base de dados Taxi San Francisco foi utilizado somente a técnica de remoção por velocidade. A escolha por essa técnica foi devido as trajetórias possuírem baixa densidade de pontos, já que o tempo médio entre os pontos é de 125,75 segundos. A Figura 69 ilustra as trajetórias selecionadas para limpeza da base de dados Taxi San Francisco com seus ruídos já removidos. A técnica utilizada foi remover o primeiro ponto quando a velocidade entre esse e o ponto seguinte for igual ou superior a 150km/h. Essa técnica obteve ótimo resultado nessa base de dados, removendo todos os ruídos.

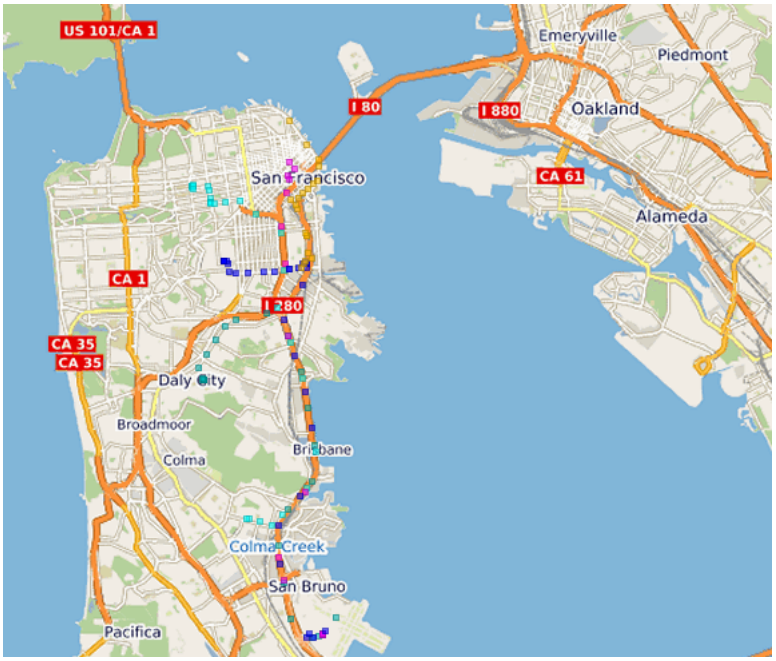

Figura 69 – Trajetórias da base de dados Taxi San Francisco após processo de remoção de ruídos por velocidade superior a 150km/h

A seguir é apresentado em tabelas o resultado das limpezas das trajetórias em ambas as bases de dados. Nessas tabelas é apresentado o resultado da limpeza das trajetórias pelo sistema em comparação com a limpeza manual das mesmas trajetórias. Essa comparação verifica quantos pontos a técnica removeu e quantos pontos ela acertou em comparação com a remoção manual. A comparação foi realizada em somente 5 trajetórias de cada base de dados, pois há um custo elevado em realizar tal tarefa para toda a base de dados.

Para compreensão das tabelas vamos definir que **TID** é o identificador da trajetória, **RM** é a quantidade de pontos removidos manualmente, **CR** é a quantidade de pontos corretamente removidos pela técnica e **TR** é a quantidade total de pontos removidos pela técnica. Ainda temos o prefixo **GL** no identificador da trajetória se referindo à uma trajetória da base de dados GeoLife e **SF** à base de dados Taxi San Francisco.

A Tabela 4 apresenta as quantidades de pontos removidos pela técnica DBSCAN na base de dados GeoLife. Para essa técnica foram utilizado 4 diferentes parâmetros. Foi utilizado uma variação do raio da vizinhança para uma determinada quantidade de vizinhos do ponto. Assim, para uma quantidade miníma de 2 pontos em uma vizinhança foi verificado o raio de 50 e 100 metros. Para a quantidade miníma de 5 pontos na vizinhança foi verificado os raios de 100 e 200 metros. Esses valores para o raio foram definidos baseados na distância média dos pontos na base GeoLife, que estão em média 76 metros distantes entre si.

Tabela 4 – Quantidade de pontos identificados como ruído pelo DBSCAN na base de dados GeoLife. Onde RM é a referência dos ruídos removidos manualmente, CR são os ruídos removidos corretamente pelo algoritmo e TR é o total de ruídos removidos na trajetória

pontos mínimo		2 pontos				5 pontos			
distância		50m		100m		100m		200m	
TID	**RM**	**CR**	**TR**	**CR**	**TR**	**CR**	**TR**	**CR**	**TR**
GL 2	2	**2**	**2**	**2**	**2**	**2**	**2**	**2**	**2**
GL 32	2	**2**	**2**	1	1	1	1	1	1
GL 86	4	0	0	0	0	**4**	**4**	**4**	**4**
GL 93	2	**2**	**2**	0	0	**2**	**2**	**2**	**2**
GL 105	11	9	9	1	1	9	9	2	2

A tabela ainda permite fazer uma análise, através de uma amostra, para descobrir qual o melhor conjunto de parâmetros a ser utilizado em toda a base de dados. Os números em negrito destacam quais conjuntos de parâmetros acertaram a quantidade de removidos(TR) e corretos removidos(CR) em comparação com os removidos manualmente(RM). Podemos perceber na Tabela 4 que a técnica que agrupa 5 pontos em uma vizinhança com raio de 100 metros, entre os conjuntos de parâmetros testados, foi então a que obteve melhores resultados. Esse conjunto de parâmetros permitiu remover ruídos em todas as trajetórias com alto índice de acerto. Já o pior conjunto de parâmetros foi agrupar uma vizinhança com 2 pontos em um raio de 100 metros. O sucesso e falha do conjunto de parâmetros se deve ao padrão de densidade dos pontos nas trajetórias. Por exemplo, agrupar somente 2 pontos em um raio de 100 metros permite que um ruído seja considerado parte da trajetória e não seja removido. Dessa forma, considerar um número maior de vizinhos garantirá a exclusão de ruídos isolados.

Para a remoção de ruídos por velocidade foi utilizado as bases de dados GeoLife e Taxi San Francisco. Para essa técnica foi utilizado um conjunto de 4 parâmetros. Sendo a remoção do primeiro ponto e a remoção do segundo, conforme descritos na Seção 2.3. Para cada um desses foram utilizadas as velocidades de 150km/h e 200km/h como parâmetro. A Tabela 5 apresenta os números das remoções de ruídos nessas bases. Na Tabela estão destacados em negrito o conjunto de parâmetros que melhor removeu ruídos nas trajetórias. Ainda na Tabela podemos perceber que nenhum conjunto de parâmetros nessa técnica foi relevante para a base de dados Geolife, certamente por grande parte dos dados serem de pedestres e assim não atingirem altas velocidades. Entretanto podemos perceber que para as trajetórias da base de dados Taxi San Francisco o melhor conjunto de parâmetros foi remover o segundo ponto quando a velocidade for igual ou superior a 200km/h. Já o pior conjunto de parâmetros foi remover o segundo ponto com velocidade igual ou superior a 150km/h, esse que não conseguiu remover pontos corretamente em nenhuma das trajetórias.

Tabela 5 – Quantidade de pontos identificados como ruído por velocidade nas bases de dados GeoLife e Taxi San Francisco. Onde RM é a referência dos ruídos removidos manualmente, CR são os ruídos removidos corretamente pelo algoritmo e TR é o total de ruídos removidos na trajetória

Iniciando no		1º ponto				2º ponto			
Velocidade acima de		150km/h		200km/h		150km/h		200km/h	
TID	**RM**	**CR**	**TR**	**CR**	**TR**	**CR**	**TR**	**CR**	**TR**
GL 2	2	**2**	**2**	1	5	**2**	**2**	0	3
GL 32	2	1	1	1	1	1	3	1	2
GL 86	4	2	2	1	1	0	2	0	1
GL 93	2	1	1	1	1	0	1	0	1
GL 105	11	6	6	5	5	3	6	2	4
SF 24545	7	6	7	6	7	6	6	6	6
SF 160337	6	**6**	**6**	**6**	**6**	5	13	5	10
SF 49211	1	1	3	1	2	1	2	**1**	**1**
SF 49266	1	1	3	1	2	1	2	**1**	**1**
SF 49495	3	3	9	3	6	3	7	**3**	**3**

Além das técnicas de remoção de ruídos para limpeza de trajetória, há ainda disponível no sistema as técnicas para suaviazação de trajetórias. Diferente das técnicas de remoção de ruídos, uma técnica por filtro de suavização não remove nenhum ponto da trajetória. O intuito de um filtro de suavização, como descrito na Seção 2.3, é trazer o ponto para mais próximo aos demais pontos da trajetória e reduzindo assim sua distância de fora da trajetória. Esses filtros necessitam de alta densidade de pontos na trajetória para realizar as suavizações por média e mediana. Sendo assim são melhor aplicados em bases de dados com alta densidade, como é o caso da base de dados GeoLife.

A Figura 70 ilustra as 5 trajetórias com ruídos da base de dados GeoLife. Porém uma aproximação das trajetórias foi realizada na Figura para melhor visualização da suavização dessas trajetórias. As trajetórias em vermelho são as originais com ruídos, já as trajetórias em azul são o resultado do filtro de suavização por média utilizando uma janela de 3 pontos. Ainda na Figura 70 podemos perceber dois recursos interessantes dos filtros de suavização. Primeiro podemos perceber a suavização da trajetória nas curvas, canto direito inferior destacado pelo elipse laranja. E o segundo recurso é a criação de pontos em espaços vazios, como podemos perceber na região central esquerda da figura destacado pela elipse azul. Originalmente nessa região não haviam pontos, os pontos eram ruídos e estavam distantes do restante da trajetória. Dessa forma o filtro de suavização, através da média dos pontos, trouxe os ruídos para mais próximo da trajetória. O filtro de suavização se torna um ótimo recurso quando a remoção de pontos não é uma opção, visto que essa técnica não irá remover nenhum ponto. Por outro lado ela acaba alterando a localização original dos pontos, devido aos ajustes que faz em relação aos demais pontos.

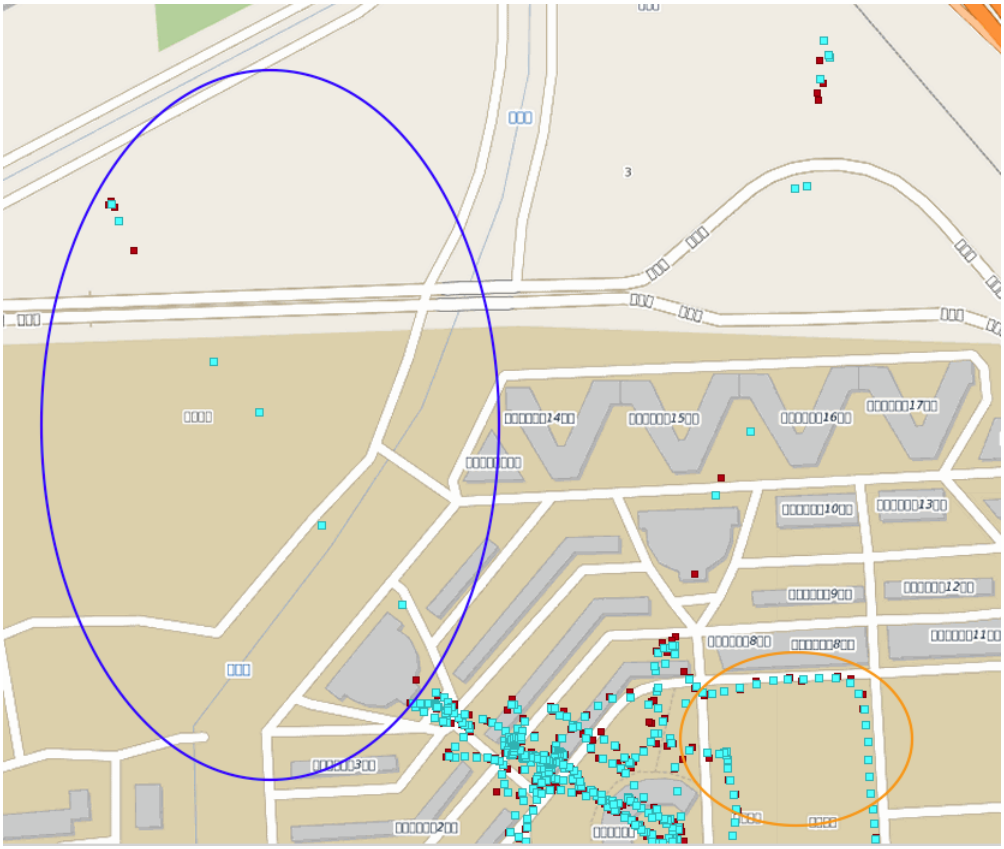

Figura 70 – Trajetórias da base de dados GeoLife após processo de suavização por média destacado pelas elipses

4.4 SELEÇÃO DE TRAJETÓRIAS PRÓXIMAS A UM PONTO

Nessa seção foram selecionadas trajetórias próximas a um ponto de interesse nas bases de dados Geolife e Taxi San Francisco. Essa seleção permite obter trajetórias que cruzaram determinado ponto, reduzindo assim a quantidade de dados a ser utilizada em uma análise de determinada região. Dessa forma foram selecionadas trajetórias que cruzaram um raio de 250 metros de um aeroporto internacional em cada base de dados.

4.4.1 ESTUDO DE CASO 1 - GEOLIFE

O aeroporto internacional de Pequim foi escolhido como ponto de referência e foi definido com a longitude 40.076641 e latitude 116.583966. A partir do raio de 250 metros desse ponto foram então selecionadas 78 das 18 670 trajetórias da base de dados Geolife. A Figura 71 ilustra essas trajetórias, com a região do aeroporto destacada por um círculo. Ainda na Figura 71, além das trajetórias por vias terrestres, também é possível perceber trajetórias de aviões passando pela área de referência do aeroporto, nas partes superior e esquerda da Figura.

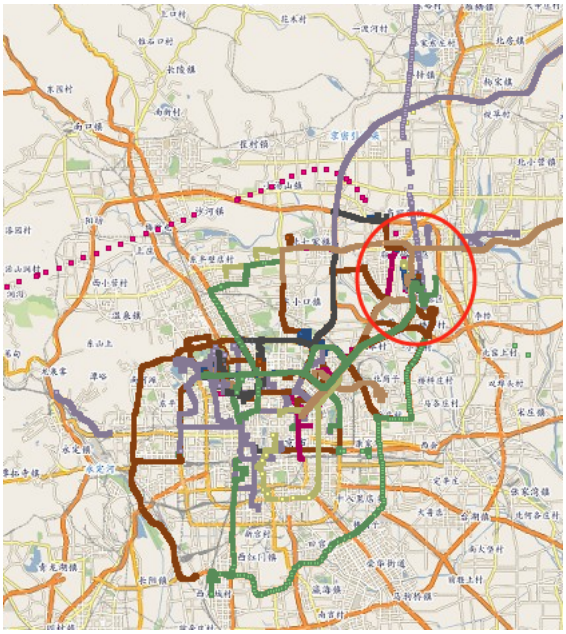

Figura 71 – Trajetórias que cruzam raio de 250 metros do aeroporto internacional de Pequim, destacado por um círculo, na base de dados GeoLife.

4.4.2 ESTUDO DE CASO 2 - TAXI SAN FRANCISCO

Para esse estudo de caso foi utilizado a base de dados Taxi San Francisco segmentada por estado de ocupação do táxi, pois assim as viagens de táxi pela cidade são melhores representadas. A partir dessa base de dados foram selecionadas trajetórias próximas ao aeroporto internacional de São Francisco, definido pelo ponto com as coordenadas de longitude -122.38656 e latitude 37.61701. A Figura 72 ilustra as trajetórias selecionadas da base de dados e a região do aeroporto está destacada por um círculo. Com esse processo foi possível selecionar 45 595 trajetórias entre as 928 301 trajetórias da base de dados Taxi San Francisco segmentada conforme a ocupação do táxi.

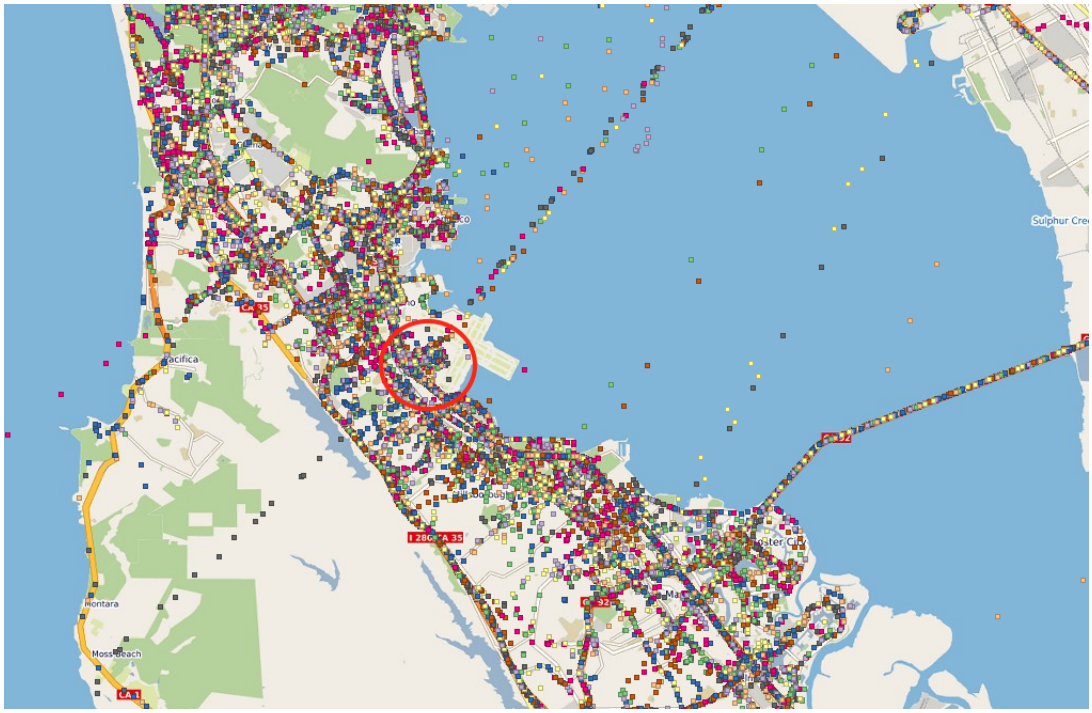

Figura 72 – Trajetórias que cruzam raio de 250 metros do aeroporto internacional de São Francisco, destacado por um círculo, na base de dados Taxi San Francisco.

4.5 EXPORTAÇÃO DE DADOS DO BANCO DE DADOS

A Exportação de Dados é uma funcionalidade do sistema que permite exportar uma tabela do banco de dados para um arquivo externo de formato CSV. Através dessa funcionalidade é possível exportar bases de dados processadas ou até mesmo partes dessa base, como algumas trajetórias selecionadas pelo sistema que cruzam determinado ponto de uma região. Assim esses arquivos externos permitem o intercâmbio dos dados processados, seja para outros sistemas ou pesquisadores.

Assim foi utilizado o sistema em 2 estudos de casos para exportar as bases de dados Geolife e Taxi San Francisco. O sistema permitiu exportar para arquivo CSV as tabelas contendo as bases de dados completas, carregadas no banco de dados e também tabelas com as trajetórias selecionadas em cada base.

4.5.1 ESTUDO DE CASO 1 - GEOLIFE

A tabela no banco de dados com a base de dados Geolife possui 24 876 978 registros e a tabela com as trajetórias selecionadas próximas ao aeroporto internacional de Pequim possui 300 264 registros. Essas tabelas foram então exportadas pelo sistema e a Figura 71 ilustra os arquivos gerados e seus tamanhos em disco. O arquivo com a base de dados Geolife inteira totalizou 5.6GB e o arquivo com as trajetórias selecionadas com 67.2MB. A Figura 74 ilustra um trecho dos dados do arquivo CSV exportado, referente a tabela com as trajetórias selecionadas. Podemos perceber ainda na Figura 74 o cabeçalho do arquivo, com o nome das colunas da tabela de origem, que permitem identificar cada tipo de dado no arquivo.

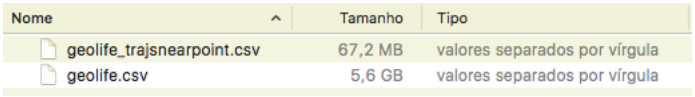

Figura 73 – Arquivos CSV com trajetórias da base de dados Geolife exportados pelo sistema

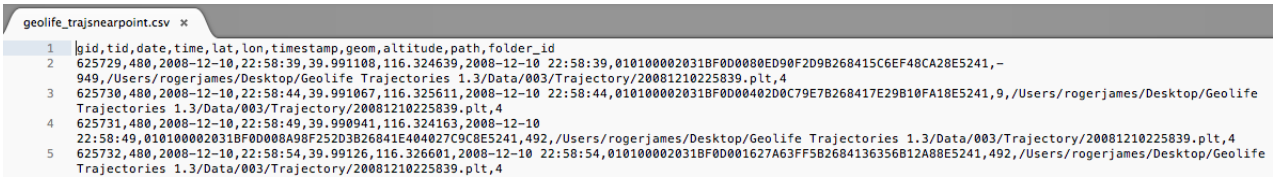

Figura 74 – Trecho do arquivo de dados exportado pelo sistema da tabela com trajetórias selecionadas da base de dados Geolife

4.5.2 ESTUDO DE CASO 2 - TAXI SAN FRANCISCO

A base de dados Taxi San Francisco após o carregamento totalizou 11 219 955 registros em uma tabela no banco de dados e a tabela com as trajetórias selecionadas próximas ao aeroporto internacional de São Francisco com 1 298 290 registros. A exportação dessas tabelas pelo sistema está ilustrado pela Figura 75, onde podemos perceber os arquivos gerados e o tamanho que ocupam em disco. O arquivo com todos os registros da base de dados Taxi San Francisco totaliza 2.06GB em disco, enquanto o arquivo com as trajetórias selecionadas ocupam apenas 239.5MB. Já a Figura 76 ilustra um trecho dos dados do arquivo CSV exportado a partir da tabela com trajetórias selecionadas da base de dados Taxi San Francisco.

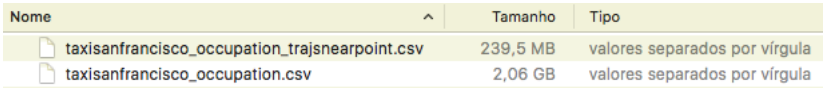

Figura 75 – Arquivos CSV com trajetórias da base de dados Geolife exportados pelo sistema

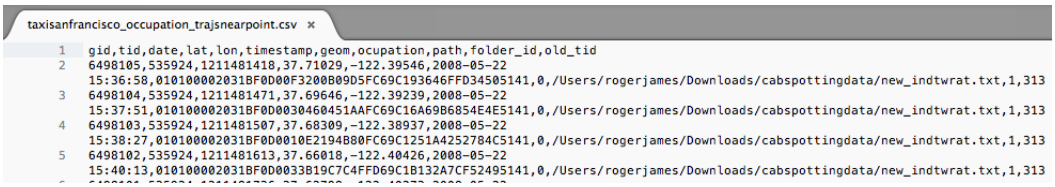

Figura 76 – Trecho do arquivo de dados exportado pelo sistema da tabela com trajetórias selecionadas da base de dados Taxi San Francisco

5 AVALIAÇÃO DE SIMILARIDADE DE TRAJETÓRIAS

Nesse capítulo é apresentada uma avaliação do impacto do uso de técnicas de limpeza e pré-processamento no cálculo da similaridade entre trajetórias. Essa avaliação é feita através de uma análise da variação do grau de similaridade entre as trajetórias, em função das transformações realizadas, nas bases de dados originais e segmentadas, por fim também é verificado a influência da existência de ruídos no cálculo da similaridade dessas trajetórias.

Para avaliar a similaridade entre trajetórias é preciso utilizar uma medida de similaridade que quantifica o quão similares ou distantes duas trajetórias são (KEOGH; RATANAMAHATANA, 2005). Após a quantificação da similaridade o resultado da medida pode ser utilizado para agrupar trajetórias para uso em determinadas aplicações, pois em muitos casos as trajetórias possuem muitas informações associadas e é preciso agrupar trajetórias segundo alguma similaridade entre elas (HWANG et al., 2005). Ainda assim o uso de similaridade entre trajetórias é importante em diversas outras aplicações. Como por exemplo determinar o movimento de um grupo de objetos que compartilham o mesmo padrão de movimento e assim inferir as futuras localizações de um desses objetos.

Para duas trajetórias serem consideradas semelhantes entre si, elas devem satisfazer alguns requisitos particulares. Essas devem estar suficientemente próximas umas das outras no espaço euclidiano e, além disso, devem ter direções semelhantes (LIU; SCHNEIDER, 2012). De encontro a isso surge o grande desafio: Como medir a similaridade entre essas trajetórias?

Para isso existem na literatura uma série de medidas de distância e similaridade para trajetórias (ZHENG; ZHOU, 2011). Essas medidas são baseadas no cálculo de distância entre os pares de objetos do conjunto de dados. Nesse processo, é aplicada a função de distância nos valores dos atributos de interesse sobre os objetos. Com o uso de uma determinada medida de distância, é possível descobrir a similaridade entre dois objetos, ou seja, podem-se identificar objetos que mais se aproximam, de acordo com suas características (ESLING; AGON, 2012). Entre essas medidas podemos citar *Sum-of-Pairs Distance* (AGRAWAL; FALOUTSOS; SWAMI, 1993), *Dynamic Time Warping* (DTW) (BERNDT; CLIFFORD, 1994), *Longest Common Subsequence* (LCSS) (ZHENG et al., 2008), *Edit Distance with Real Penalty* (ERP) (CHEN et al., 2010) e *Edit Distance on Real Sequences* (EDR) (CHEN; ÖZSU; ORIA, 2005).

Esse trabalho utiliza duas medidas de distância disponíveis na literatura para analisar a similaridade entre trajetórias. As medidas DTW e EDR foram escolhidas pois são amplamente utilizadas na literatura e possuem abordagens distintas para o cálculo da distância. A medida de distância DTW realiza comparações para todos os pontos de uma trajetória buscando o melhor alinhamento entre as duas trajetórias comparadas, o que a torna uma medida sensível a ruídos. Por outro lado, o EDR calcula a distância das edições feitas em uma trajetória para se aproximar da outra trajetória comparada. O EDR ainda possui um limiar de aceitação, onde se a distância de um ponto não estiver dentro do limite aceito, então esse ponto ficará fora do cálculo. Assim o efeito de ruídos na medida de distância EDR é inferior do que na medida de distância DTW. A seguir são apresentadas em detalhes as medidas de distância DTW e EDR.

Dynamic Time Warping (DTW) permite calcular a distância entre trajetórias de comprimentos distintos. O DTW resolve uma limitação existente na distância euclidiana. Na distância euclidiana é exigido que as duas trajetórias sejam do mesmo comprimento, o que é muito improvável para aplicações na vida real. Medidas de similaridade mais ideais devem ter certa flexibilidade no comprimento das duas trajetórias. A ideia básica do DTW é permitir "repetir" alguns pontos quantas vezes for necessário para obter o melhor alinhamento entre as duas trajetórias, conforme ilustrado pela Figura 77.

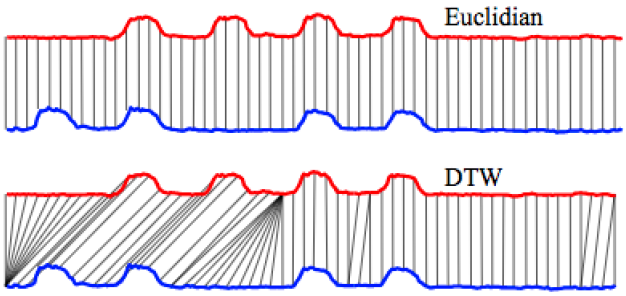

Figura 77 – Alinhamento dos pontos de duas trajetórias pela distância euclidiana e alinhamento dos pontos de duas trajetórias pela medida de distância DTW (KEOGH, 2002)

Dada uma trajetória $A =< a_1, ..., a_n >$, o início de A é denotado por a_1 e a sobra de A é $< a_2, ..., a_n >$. A distância de distorção temporal (*Dynamic Time Warping*) entre duas trajetórias A e B com comprimentos de n e m é definida como (BERNDT; CLIFFORD, 1994):

$$DTW(A,B) = \begin{cases} 0, \text{ se } n = 0 \text{ e } m = 0 \\ \infty, \text{se } n = 0 \text{ ou } m = 0 \\ \text{d(início(A), início(B))} + \text{mínimo} \begin{cases} \text{DTW(A, sobra(B))} \\ \text{DTW(sobra(A), B)} \\ \text{DTW(sobra(A), sobra(B))} \end{cases} \end{cases} \tag{5.1}$$

Onde d(,) pode ser qualquer uma das funções de distância definidas em pontos.

$$DTW\ normalizada = \frac{DTW(A,B)}{maior(|A|,|B|)} \tag{5.2}$$

A normalização da medida de distância DTW se deve ao fato do tamanho das trajetórias ser variável. Essa variação no tamanho das trajetórias afeta os resultados da medida de distância, devido a medida somar as distâncias entre pares de pontos da trajetória. Assim a normalização é feita através do tamanho da maior trajetória comparada, de forma a tornar comparáveis os resultados de trajetórias com diferentes tamanhos.

Edit Distance on Real Sequence (EDR) calcula a distância entre duas trajetórias contando o número de operações de edição (inserir, excluir e substituir) que são necessárias para transformar uma trajetória em outra. A medida de distância EDR utiliza um valor ε como limiar de aceitação de distância para decidir se dois pontos são similares. Portanto se a distância entre dois pontos for menor que ε, então esses são considerados similares e se estiverem mais afastados, serão considerados diferentes. Dessa forma, dadas duas trajetórias A e B com comprimentos de n e m, respectivamente, com limiar de aceitação ε. Então a distância EDR entre A e B é o número de inserção, exclusão ou substituição de operações necessárias para alterar A em B (CHEN; ÖZSU; ORIA, 2005).

$$EDR(A,B) = \begin{cases} n & \text{se } m = 0 \\ m & \text{se } n = 0 \\ \text{mínimo\{EDR(A), sobra(B) + } custo, & \text{senão} \\ \text{EDR(sobra(A), B) + 1, EDR(A, sobra(B)) + 1\}} & \end{cases} \tag{5.3}$$

onde

$$custo = \begin{cases} 0, \text{se d(início(A), início(B))} \leq \varepsilon \\ 1, \text{senão} \end{cases} \tag{5.4}$$

$$EDR\ normalizado = \frac{EDR(A,B)}{maior(|A|,|B|)} \tag{5.5}$$

A distância EDR foi normalizada pelo tamanho da maior trajetória, devido as trajetórias possuirem tamanhos distintos.

$$Similaridade\ EDR = 1 - EDR\ normalizado \tag{5.6}$$

Já para transformar a distância EDR em uma medida de similaridade, foi então subtraído o valor da distância EDR normalizada de 1. Obtendo-se assim um grau de similaridade EDR entre 0 e 1. Nesse sentido a maior similaridade é 1 e a menor similaridade é 0.

5.1 AVALIAÇÃO DA VARIAÇÃO DO GRAU DE SIMILARIDADE DE TRAJETÓRIAS EM DIFERENTES BASES DE DADOS

Para avaliação da variação do grau de similaridade nas bases de dados foram então utilizadas 3 variações de cada base de dados carregadas por esse trabalho. A versão original, uma versão segmentada por intervalos de 5 minutos e uma outra versão segmentada em 10 minutos.

Em cada uma dessas 3 variações de bases de dados foi calculado, para cada trajetória, a média do grau de similaridade das 10 trajetórias mais similares e a média das 10 trajetórias menos similares. Com isso foi formado uma lista com as médias das 10 trajetórias mais similares e menos similares de cada trajetória. A partir dessa lista, com as médias, foi então calculada a média para todas as trajetórias da base de dados. Formando assim a média das médias das 10 trajetórias mais similares e a média das médias das 10 trajetórias menos similares para cada base de dados.

O intuito da análise de similaridade nessas bases de dados é observar a variação do grau de similaridade das trajetórias através das diferentes técnicas de pré-processamento disponíveis no sistema criado por esse trabalho. O esperado é que se obtenha, após segmentação ou limpeza dos dados de trajetórias, um grau de similaridade maior entre as trajetórias mais similares e que o grau de similaridade entre as menos similares diminua. Assim aumentamos a discernibilidade entre as trajetórias, permitindo discernir de forma mais fácil as trajetórias mais similares das menos similares.

A Tabela 6 apresenta o comparativo dos resultados da medida de similaridade EDR nas bases de dados originais e segmentadas em 5 minutos. Já a Tabela 7 apresenta o comparativo dos resultados da medida de similaridade EDR nas bases de dados originais e segmentadas em 10 minutos. Para cada base de dados são apresentadas as média das 10 trajetórias mais similares e a média das 10 trajetórias menos similares. Ainda nas tabelas são apresentadas as variações das médias das 10 trajetórias mais similares entre as bases de dados original e segmentada. Para uso do EDR foi utilizado um limiar de aceitação de 100 metros, ou seja, se a distância entre dois pontos for maior que 100 metros, então esses não serão considerados similares pelo EDR. Ainda foi definido que a medida de similaridade EDR mede o quão uma trajetória é dissimilar em relação a outra trajetória, variando entre 0 e 1, ou seja, 1 para a maior similaridade e 0 para a menor similaridade entre duas trajetórias.

Tabela 6 – Comparativo das médias das 10 trajetórias mais e menos similares utilizando a medida de similaridade EDR nas bases de dados originais e nas mesmas bases de dados segmentadas em 5 minutos

Base de dados	Original		Segmentada		**Variação da média 10 mais similares**
	Média 10 mais similares	Média 10 menos similares	Média 10 mais similares	Média 10 menos similares	
AIS Brest	0.0336	0.0000	0.8506	0.0000	**0.817**
Athens School Bus	0.0196	0.0196	0.2919	0.0000	**0.2723**
Cruz dataset	0.2781	0.0000	0.2655	0.0000	-0.0126
Dublin Bus	0.2290	0.0000	0.2746	0.0000	**0.0456**
Floripa dataset	0.5861	0.0000	0.7488	0.0000	**0.1627**
GeoLife	0.2343	0.0000	0.2459	0.0000	**0.0116**
Greek Trucks	0.1061	0.0009	0.5877	0.0000	**0.4816**
Greek Trucks rev	0.3595	0.0000	0.5564	0.0000	**0.1969**
NYC buses	0.4555	0.3273	0.5743	0.0000	**0.1188**
Taxi Roma	0.0888	0.0003	0.1321	0.0000	**0.0433**
Taxi San Francisco	0.0555	0.0020	0.1972	0.0000	**0.1417**
T-Drive	0.1044	0.0000	0.3493	0.0000	**0.2449**
Uber San Francisco	0.3360	0.0000	0.3353	0.0000	-0.0007
W4M Milano	0.1120	0.0000	0.4386	0.0000	**0.3266**

Tabela 7 – Comparativo das médias das 10 trajetórias mais e menos similares utilizando a medida de similaridade EDR nas bases de dados originais e nas mesmas bases de dados segmentadas em 10 minutos

Base de dados	Original		Segmentada		**Variação da média 10 mais similares**
	Média 10 mais similares	Média 10 menos similares	Média 10 mais similares	Média 10 menos similares	
AIS Brest	0.0336	0.0000	0.6604	0.0000	**0.6268**
Athens School Bus	0.0196	0.0196	0.2765	0.0000	**0.2511**
Cruz dataset	0.2781	0.0000	0.2707	0.0000	-0.0074
Dublin Bus	0.2290	0.0000	0.2601	0.0000	**0.0311**
Floripa dataset	0.5861	0.0000	0.6014	0.0000	**0.0153**
GeoLife	0.2343	0.0000	0.2464	0.0000	**0.0121**
Greek Trucks	0.1061	0.0009	0.4968	0.0000	**0.3907**
Greek Trucks rev	0.3595	0.0000	0.4903	0.0000	**0.1308**
NYC buses	0.4555	0.3273	0.5740	0.0000	**0.1185**
Taxi Roma	0.0888	0.0003	0.1230	0.0000	**0.0342**
Taxi San Francisco	0.0555	0.0020	0.1361	0.0000	**0.0806**
T-Drive	0.1044	0.0000	0.1254	0.0000	**0.021**
Uber San Francisco	0.3360	0.0000	0.3359	0.0000	-0.0001
W4M Milano	0.1120	0.0000	0.2786	0.0000	**0.1666**

Dos resultados obtidos com EDR, entre as 14 bases de dados, apenas 2 não obtiveram aumento na média do grau de similaridade das 10 trajetórias mais similares após a segmentação. As bases de dados *Cruz dataset* e *Uber San Francisco* não obtiveram aumento na média do grau de similaridade pois já foram disponibilizadas pré-processadas, ou seja, o mantenedor da base de dados já havia realizado tratamento e organização dos dados antes da disponibilização pública. Por exemplo, os mantenedores da base de dados Uber San Francisco já haviam retirado pontos redundantes nas trajetórias. Sendo assim tentativas adicionais de pré-processamento, nessas duas bases, podem prejudicar a qualidade dos dados. Já as bases de dados que tiveram maiores variações na média do grau de similaridade das 10 trajetórias mais similares foi devido a eliminação de grandes intervalos de tempo no meio das trajetórias pelo processo de segmentação. Dessa forma *AIS Brest, Athens School Bus, Greek Trucks* e *W4M Milano* estão entre as bases de dados com maiores variações na média do grau de similaridade das 10 trajetórias mais similares após o processo de segmentação.

Também é notável que os valores de similaridades nas bases de dados segmentadas em 5 minutos é superior as bases segmentadas em 10 minutos. Esse efeito se dá pelo fato da segmentação por 5 minutos obter trajetórias mais curtas, aumentando assim a probabilidade de encontrar trajetórias mais similares.

A Tabela 8 apresenta o comparativo dos resultados da medida de distância DTW nas bases de dados segmentadas em 5 minutos e a Tabela 9 apresenta o comparativo dos resultados da medida de distância DTW nas bases de dados segmentadas em 10 minutos. O resultado do algoritmo DTW é a distância entre as duas trajetórias. Dessa forma é esperado que valores próximos a zero sejam de trajetórias muito próximas e valores maiores sejam trajetórias mais distantes. Assim a coluna de variação da média das 10 trajetórias mais similares representa a distância reduzida em relação a base de dados original. Ou seja, valores mais altos nessa coluna representam maiores reduções da média de distância na base de dados segmentada.

Tabela 8 – Comparativo das médias das 10 trajetórias mais e menos similares utilizando a medida de distância DTW nas bases de dados originais e nas mesmas bases de dados segmentadas em 5 minutos

Base de dados	Original		Segmentada		**Variação da média 10 mais similares**
	Média 10 mais similares	Média 10 menos similares	Média 10 mais similares	Média 10 menos similares	
AIS Brest	5593.8281	126332.9776	217.7267	165022.2955	**5376.1014**
Athens School Bus	9620.5167	9620.5167	1860.7674	143494.5089	**7759.7493**
Cruz dataset	889.0211	8117.8113	853.5970	8619.3020	**35.4241**
Dublin Bus	2708.9523	35374.5971	1424.2897	45781.0800	**1284.6626**
Floripa dataset	7016.5408	249450.2344	399.5205	618784.2483	**6617.0203**
GeoLife	24279.2474	2566490.8597	9529.3183	2884578.8921	**14746.6817**
Greek Trucks	3892.3143	16835.7402	359.6540	38639.0693	**3532.6603**
Greek Trucks rev	784.5961	29535.1097	389.8328	38345.0859	**394.7633**
NYC buses	639.6675	1700.4447	110.3643	8077.2820	**529.3032**
Taxi Roma	2809.9455	15480.2882	2411.7852	48551.0418	**398.1603**
Taxi San Francisco	2474.6594	9554.5683	1163.0920	71141.9398	**1311.5674**
T-Drive	2268.1517	892945.5415	963.8534	2134521.1497	**1304.2983**
Uber San Francisco	313.9430	28760.2141	313.6620	28766.6383	**0.281**
W4M Milano	1516.2719	26178.7800	408.8328	26041.1445	**1107.4391**

Tabela 9 – Comparativo das médias das 10 trajetórias mais e menos similares utilizando a medida de distância DTW nas bases de dados originais e nas mesmas bases de dados segmentadas em 10 minutos

Base de dados	Original		Segmentada		**Variação da média 10 mais similares**
	Média 10 mais similares	Média 10 menos similares	Média 10 mais similares	Média 10 menos similares	
AIS Brest	5593.8281	126332.9776	615.1107	160461.1138	**4978.7174**
Athens School Bus	9620.5167	9620.5167	2061.3335	89686.3983	**7559.1832**
Cruz dataset	889.0211	8117.8113	885.7523	8372.4076	**3.2688**
Dublin Bus	2708.9523	35374.5971	1636.7873	42088.6397	**1072.165**
Floripa dataset	7016.5408	249450.2344	1279.3684	617623.2895	**5737.1724**
GeoLife	24279.2474	2566490.8597	13886.9016	2777423.9313	**10392.3458**
Greek Trucks	3892.3143	16835.7402	538.3048	38123.1654	**3354.0095**
Greek Trucks rev	784.5961	29535.1097	501.2246	37802.0752	**283.3715**
NYC buses	639.6675	1700.4447	116.3028	7875.2147	**523.3647**
Taxi Roma	2809.9455	15480.2882	2591.9574	46758.3706	**217.9881**
Taxi San Francisco	2474.6594	9554.5683	1598.1456	44786.1355	**876.5138**
T-Drive	2268.1517	892945.5415	1854.9604	1188952.4067	**413.1913**
Uber San Francisco	313.9430	28760.2141	314.0121	28761.9895	-0.0691
W4M Milano	1516.2719	26178.7800	722.2912	26652.3645	**793.9807**

Com a segmentação das trajetórias por intervalos de tempo de 5 minutos foi então possível reduzir a distância DTW da média das 10 trajetórias mais similares em todas as bases de dados. A segmentação de trajetórias por intervalos de tempo de 5 minutos permite melhor comparação das trajetórias da base de dados, devido ao tamanho reduzido das trajetórias em relação a base de dados original ou até mesmo com a segmentação por maiores intervalos de tempo. Já a distância DTW média das 10 trajetórias menos similares aumentaram nas bases de dados segmentadas em relação as bases de dados originais. Assim a segmentação das trajetórias permitiu maior discernimento entre essas trajetórias, tornando mais fácil diferenciar trajetórias mais similares das menos similares.

5.2 AVALIAÇÃO DA VARIAÇÃO DO GRAU DE SIMILARIDADE DE TRAJETÓRIAS EM DIFERENTES VARIAÇÕES NA BASE DE DADOS TAXI SAN FRANCISCO

Algumas bases de dados permitem diferentes segmentações de suas trajetórias. A base de dados Taxi San Francisco, por exemplo, permite; além da segmentação por intervalos de tempo, a segmentação pelo estado de ocupação do táxi, possibilitando assim obter trajetórias do táxi ocupado por passageiros e trajetórias sem passageiros. As Figuras 78 e 79 apresentam um comparativo da similaridade das trajetórias nessas diferentes segmentações da base de dados Taxi San Francisco. Dessa forma foram calculadas as similaridades entre as trajetórias da base de dados original, as similaridades entre as trajetórias da base dados original segmentada por intervalo de tempo em 5 minutos e as similaridades entre as trajetórias da base de dados original segmentada pelo estado de ocupação do táxi. Essa análise se deu pelo cálculo da média das 10 trajetórias mais similares para cada trajetória e por fim obtido a média das 10 trajetórias mais similares para cada base de dados. Além disso, foi verificado o comportamento da similaridade para as médias das 50, 100, 200 e 500 trajetórias mais similares em cada versão da base de dados. Somente não foi calculado a média das 500 trajetórias mais similares na base de dados original, pois a amostra utilizada só possuía 492 trajetórias. A Figura 78 ilustra o comparativo das variações das médias de similaridade EDR das trajetórias mais similares para a base de dados original, segmentada por tempo e segmentada pelo estado de ocupação do táxi da base de dados Taxi San Francisco. Com a medida de similaridade EDR existe a tendência da média das trajetórias mais similares se aproximarem de 0 conforme aumenta o número de trajetórias, visto que para a medida de similaridade EDR os valores mais próximos de 0 são trajetórias menos similares.

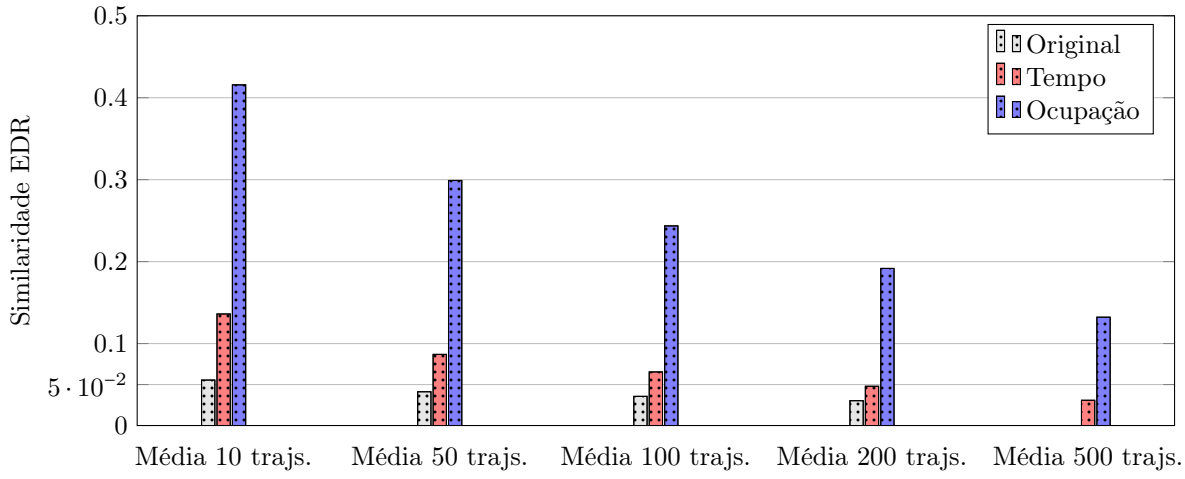

Figura 78 – Comparativo da similaridade EDR média entre as 10, 50, 100, 200 e 500 trajetórias mais similares nas versões original, segmentada por tempo em 5 minutos e segmentada por ocupação do táxi da base de dados Taxi San Francisco

A Figura 79 ilustra o comparativo da variação das médias da medida de distância DTW das trajetórias mais similares para a base de dados original, segmentada por tempo e segmentada pelo estado de ocupação do táxi da base de dados Taxi San Francisco. Com a medida DTW é notado um crescimento nas médias de distância entre as trajetórias conforme o número de trajetórias aumenta. Esse crescimento é devido a medida de distância DTW medir a distância entre as trajetórias, tornando trajetórias mais distante menos similares.

A segmentação pelo estado de ocupação do táxi atingiu, nas medidas de similaridade EDR e DTW, valores bem distantes das versões original e segmentada por tempo da base de dados. O uso da segmentação pelo estado de ocupação do táxi permitiu que as trajetórias ocupadas por passageiros fossem separadas das trajetórias não ocupadas por passageiros, tornando mais fácil o discernimento entre essas trajetórias. Dessa forma é mais fácil encontrar as trajetórias comuns aos trajetos dos táxis quando ocupado ou não.

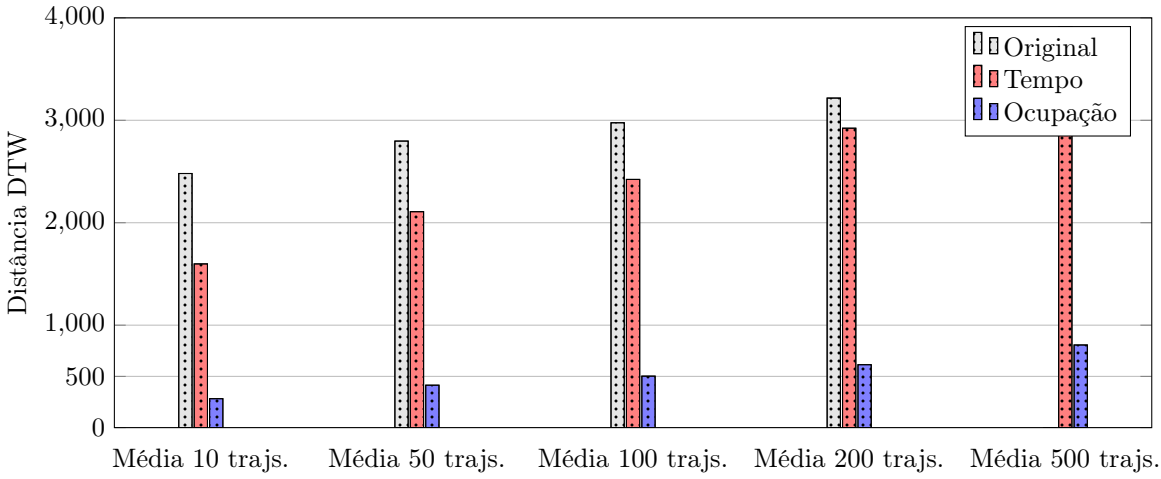

Figura 79 – Comparativo da distância DTW média entre as 10, 50, 100, 200 e 500 trajetórias mais similares nas versões original, segmentada por tempo em 5 minutos e segmentada por ocupação do táxi da base de dados Taxi San Francisco

5.3 AVALIAÇÃO DA VARIAÇÃO DO GRAU DE SIMILARIDADE DE TRAJETÓRIAS COM RUÍDOS NA BASE DE DADOS TAXI SAN FRANCISCO

De forma geral a presença de ruídos nas trajetórias é um problema que dificulta o encontro de trajetórias mais similares entre si. Já o processo de remoção desses ruídos visa tornar a trajetória mais próxima do formato real do movimento que ela representa.

Para avaliar o impacto dos ruídos nas trajetórias, foram selecionadas 100 trajetórias com ruídos na base de dados Taxi San Francisco segmentada pelo estado de ocupação do táxi, pois representam melhor os movimentos dos táxis, ilustradas pela Figura 80. Essas trajetórias permitiram analisar a variação do grau de similaridade das trajetórias antes e após a remoção de seus ruídos.

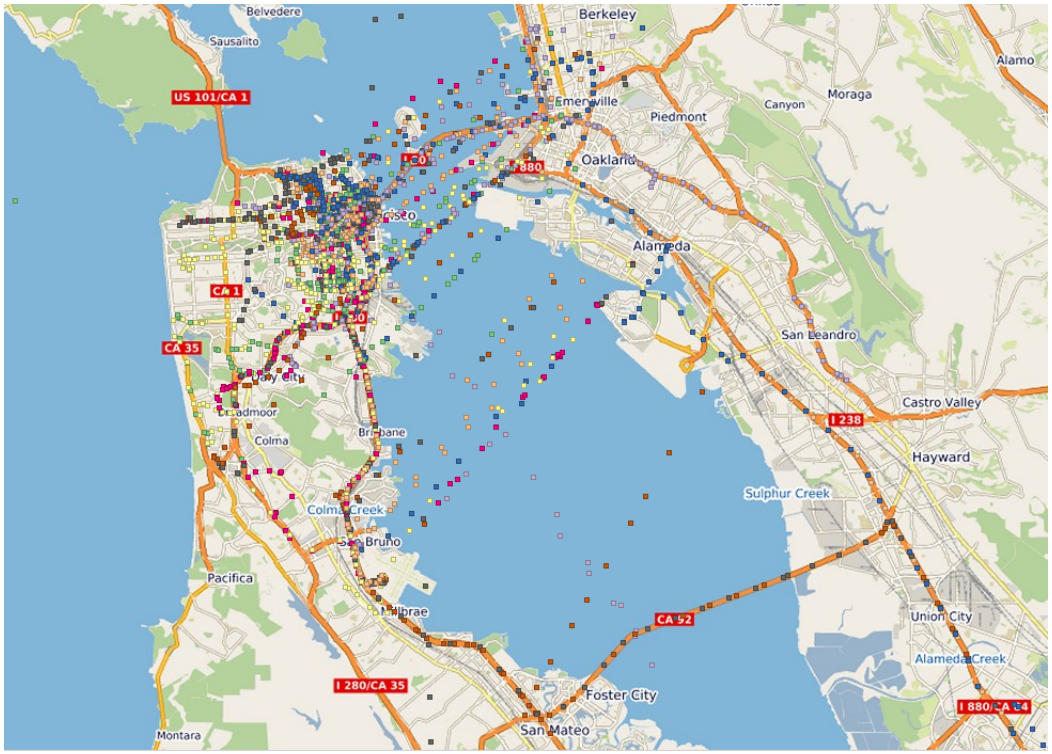

Figura 80 – Trajetórias com ruídos na base de dados Taxi San Francisco

A Figura 81 ilustra as mesmas 100 trajetórias da base de dados Taxi San Francisco, ilustradas na Figura 80, após remoção dos ruídos com velocidade acima de 200km/h. A escolha por esse parâmetro para remoção dos ruídos foi obtido através do Estudo de Caso na Seção 4.3.

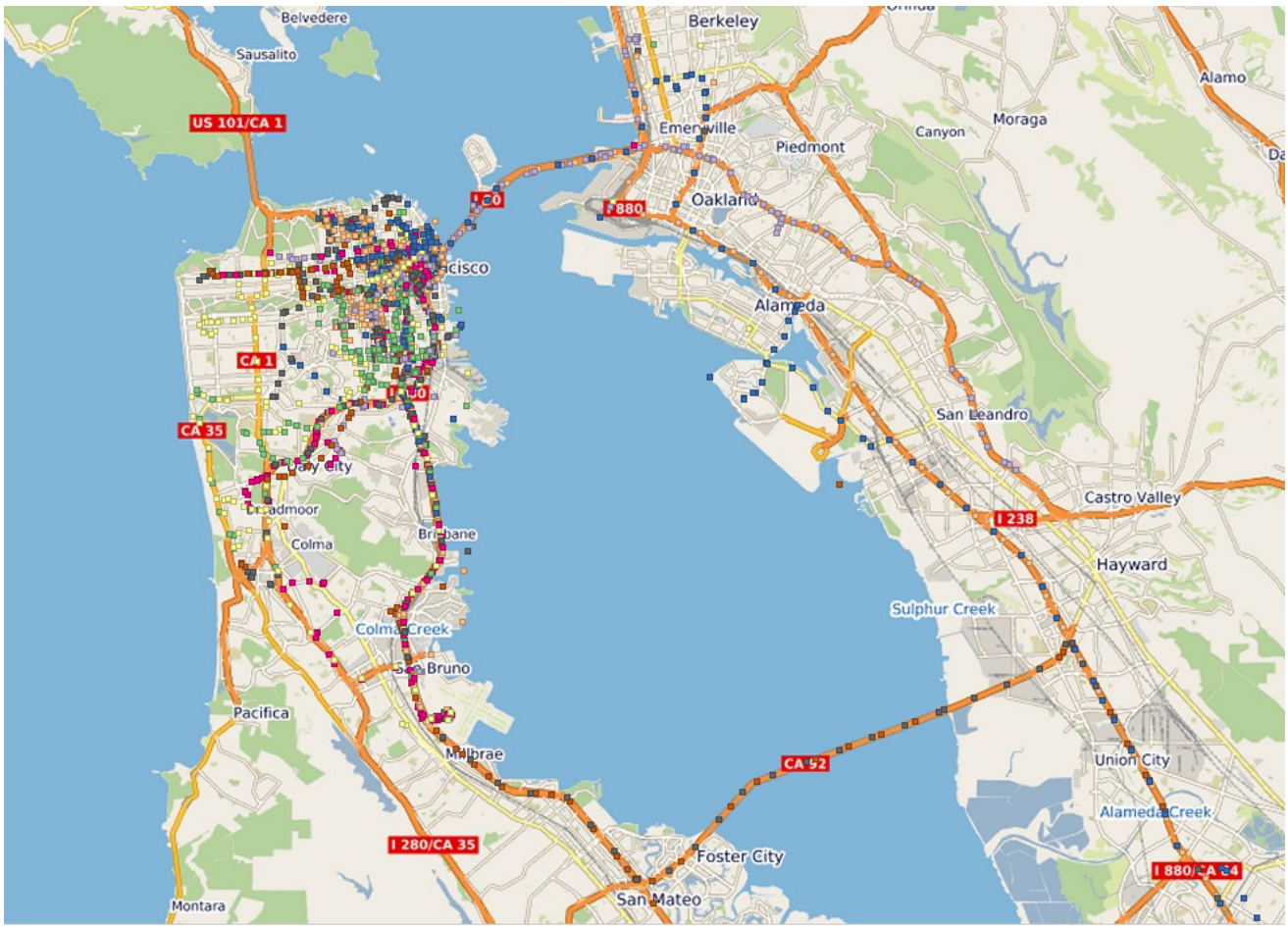

Figura 81 – Trajetórias da base de dados Taxi San Francisco após remoção dos ruídos com velocidade acima de 200km/h

Nesse processo foi obtido a média do grau de similaridade das 10 trajetórias mais similares para cada trajetória selecionada da base de dados original e comparado com o grau de similaridade médio das 10 trajetórias mais similares da mesma trajetória na base de dados com ruídos removidos. Dessa forma podemos verificar se houve maior discernibilidade entre as trajetórias após as remoções de seus ruídos, tornando mais fácil encontrar trajetórias mais similares entre si. O gráfico ilustrado pela Figura 82 apresenta a concentração das trajetórias com a medida de similaridade EDR. A linha tracejada transversalmente representa o encontro do grau de similaridade das trajetórias originais com as trajetórias limpas após remoção dos ruídos. Na parte superior do gráfico estão concentradas as trajetórias que obtiveram aumento no seu grau de similaridade médio após a remoção de seus ruídos. Já a parte inferior estão as trajetórias que reduziram seu grau de similaridade médio após a remoção de ruídos.

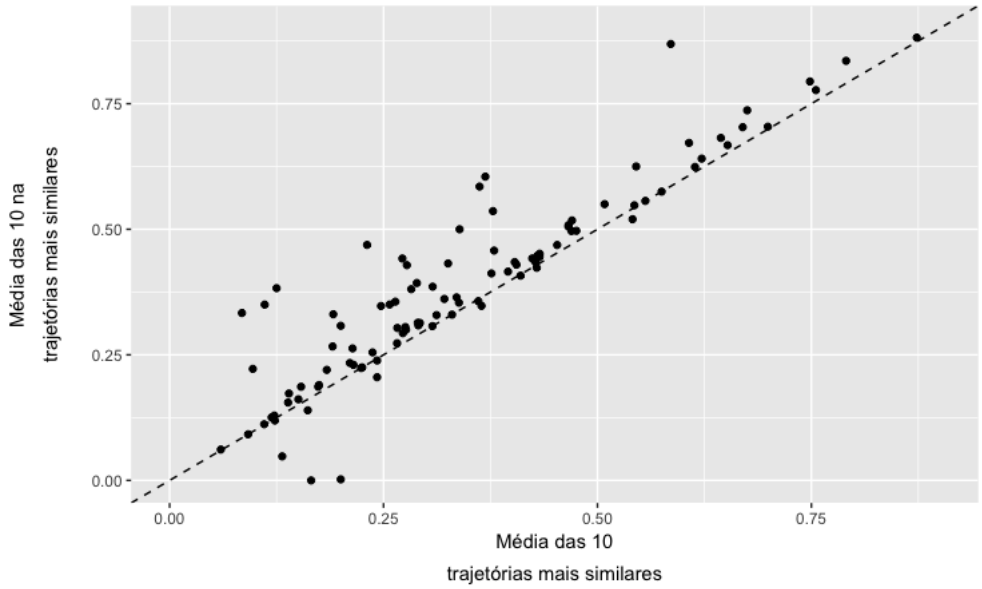

Figura 82 – Comparação do grau de similaridade EDR entre trajetórias com e sem ruídos da base de dados Taxi San Francisco. Na parte superior da linha tracejada estão as trajetórias que aumentaram seu grau médio de similaridade EDR após remoção dos ruídos e, de forma oposta, na parte inferior as trajetórias que diminuíram seu grau médio de similaridade EDR

A Figura 83 apresenta o gráfico com a concentração das trajetórias com a medida de distância DTW. A linha tracejada transversalmente representa o encontro da distância DTW entre as trajetórias originais e trajetórias limpas após remoção dos ruídos. Na parte inferior do gráfico estão concentradas as trajetórias que reduziram suas distâncias DTW após a remoção de seus ruídos. Já a parte superior estão as trajetórias que aumentaram suas distâncias DTW após a remoção de ruídos.

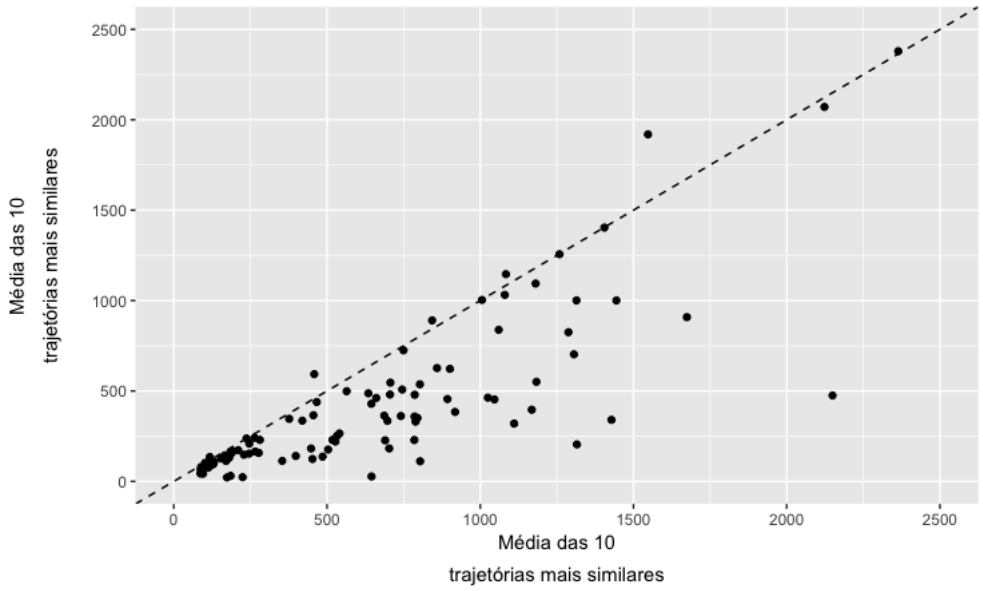

Figura 83 – Comparação da distância DTW entre trajetórias com e sem ruídos da base de dados Taxi San Francisco. Na parte inferior da linha tracejada estão as trajetórias que reduziram sua distância média DTW após remoção dos ruídos e, de forma oposta, na parte superior as trajetórias que aumentaram sua distância média DTW

As trajetórias apresentaram aumento na média da similaridade EDR e DTW após a remoção de seus ruídos. A remoção dos ruídos permite então encontrar trajetórias mais similares entre si, devido ao fato que a remoção desses ruídos busca trazer a trajetória para uma representação mais próxima do movimento real, possibilitando assim encontrar outras trajetórias que realizaram trajetos similares.

6 CONCLUSÃO E TRABALHOS FUTUROS

O crescente uso de dispositivos móveis dotados com sensores GPS tem permitido, cada vez mais, que indivíduos tenham seus movimentos registrados. A sequência desses registros formam a trajetória bruta do indivíduo. Porém o uso em larga escala dos dispositivos com sensores GPS, principalmente por grandes grupos de pessoas ou frotas de veículos, têm formado numerosas bases de dados dessas trajetórias. Algumas dessas bases de dados estão disponíveis gratuitamente na internet e são utilizadas por pesquisadores em todo o mundo. Entretanto essas bases de dados não possuem padronização e são disponibilizadas conforme o desejo de seus mantenedores. Por esse motivo, sempre que é necessário utilizar alguma dessas bases de dados, é necessário um esforço para compreensão de sua estrutura e para organização em um formato que permita sua utilização.

Com a intenção de facilitar esse processo, o presente trabalho reuniu técnicas de pré-processamento de dados de trajetórias disponíveis na literatura. Essas técnicas englobam carregamento, organização, limpeza de bases de dados de trajetórias e foram reunidas em um sistema do tipo *desktop*, desenvolvido em Java e que utiliza o banco de dados PostgreSQL com a extensão espacial PostGIS.

Para validar o sistema foram utilizadas 15 bases de dados de trajetórias de diferentes fontes e regiões do planeta. Cada uma dessas bases passou por uma análise para compreensão dos detalhes de sua estrutura e organização dos dados, a fim de fornecer entradas para o sistema carregá-las. Também durante essa análise foi calculado o tempo e distância médio entre os pontos nas trajetórias de cada base de dados. Para carregar essas bases de dados pelo sistema, foi então desenvolvido um módulo que é responsável pelo carregamento de dados de trajetórias em diferentes formatos de arquivos. Esse módulo permitiu então definir e criar uma tabela, bem como suas colunas, no banco de dados onde foram carregadas as bases de dados. Através do módulo do sistema foram realizados diversos procedimentos sobre os dados. Foram criados os identificadores únicos para cada registro, foram identificadas as trajetórias de acordo com cada base de dados, as datas foram convertidas para se adequarem ao padrão do PostgreSQL e foram definidos no banco de dados o modelo de referência geográfico utilizado na base de dados durante a coleta dos dados.

Um módulo de Segmentação de Trajetórias permitiu que as bases de dados carregadas passassem por um processo de segmentação de suas trajetórias. A segmentação de trajetórias busca eliminar grandes intervalos de tempo no meio dessas ou ainda segmentar as trajetórias conforme o táxi está ocupado ou não, no caso da base de dados Taxi San Francisco. Assim o módulo do sistema permitiu segmentar as bases de dados já carregadas anteriormente. As trajetórias de cada base de dados foram segmentadas quando havia um intervalo igual ou superior a 5 minutos entre dois pontos, separando assim em trajetórias distintas. Esse processo

também foi repetido para intervalos de tempo de 10 minutos. Além disso a base de dados Taxi San Francisco foi segmentada pelo estado de ocupação do táxi, separando trajetórias ocupadas por passageiros das trajetórias não ocupadas. Após a segmentação das trajetórias foi possível analisar o impacto dos intervalos de tempo na distância e tempo médio entre os pontos das trajetórias de cada base de dados.

Devido algumas bases de dados possuírem ruídos em suas trajetórias, foi então desenvolvido um módulo para remoção de ruídos através de técnicas que filtram esses ruídos. O módulo de Limpeza de Dados suporta os filtros de ruídos por velocidade e densidade de pontos. Já o filtro de suavização da trajetória é fornecido pelo módulo na intenção de não remover nenhum ponto, mas sim de trazer esse ruído para próximo dos demais pontos da trajetória. Para validar os filtros utilizados por esse módulo foram selecionadas as bases de dados Geolife e Taxi San Francisco em um estudo de caso para remoção dos ruídos por velocidade e densidade de pontos. Ainda na base de dados Geolife foi analisado o comportamento dos ruídos após a suavização das trajetórias por média. Nesse estudo de caso foram selecionadas, manualmente, trajetórias com ruídos em ambas as bases de dados, assim os ruídos foram identificados e removidos manualmente. Após esse processo foi possível verificar qual técnica de remoção de ruídos, utilizada pelo sistema, removia mais ruídos e quais desses removidos eram realmente um ruído, conforme havia sido identificado. Após as remoções dos ruídos pelo sistema, e comparado com as remoções manualmente, foi então possível constatar que a técnica de remoção deve ser escolhida de acordo o padrão dos pontos em sua trajetória. Trajetórias mais densas podem fazer bom uso dos filtros de densidade para remoção dos ruídos e para trajetórias menos densas o filtro de velocidade deve ser aplicado.

Ao final foi realizada uma avaliação do impacto do uso das técnicas de pré-processamento no cálculo da similaridade entre trajetórias. Nessa avaliação foram utilizadas duas medidas de similaridade amplamente utilizadas na literatura. Foi utilizada a medida de similaridade DTW, que é sensível a ruídos e a medida de similaridade EDR, que não é sensível a ruídos. Assim, essa avaliação foi feita através da variação do grau de similaridade entre as trajetórias da base de dados original, a base de dados original segmentada por intervalos de tempo de 5 e 10 minutos. Já na base de dados Taxi San Francisco foi avaliado a variação do grau de similaridade entre as bases de dados original, segmentada por tempo e segmentada conforme a ocupação do táxi. Também nessa base de dados foi analisado o impacto da remoção de ruídos no grau de similaridade das trajetórias.

Através da análise de similaridade foi possível perceber um aumento na discernibilidade entre as trajetórias. No sentido que, após a aplicação das técnicas de pré-processamento disponíveis no sistema, fica mais fácil diferenciar as trajetórias mais similares das menos similares entre si. Assim os resultados mostraram que, após o processo de segmentação por tempo e ocupação do táxi, bem como a remoção dos ruídos, as trajetórias aumentaram seu grau de similaridade em relação as mais similares, enquanto o grau de similaridade diminuiu entre as

menos similares.

Como trabalhos futuro podem ser incluídas no sistema outras técnicas de pré-processamento ou melhorias nas existentes, como:

- A funcionalidade de reajustar o intervalo de tempo médio entre os pontos.
  No sentido que duas trajetórias coletadas com o intervalo de registro configurado para diferentes intervalos possam ficar com a mesma quantidade de pontos, (ex., com a mesma taxa de amostragem através da inserção ou remoção de pontos)

- Compressão de dados de trajetórias.
  Permitir reduzir a quantidade de pontos em uma trajetória, mas mantendo a representação mais próxima de seu movimento real (AVANCINI, 2010; ZHENG; ZHOU, 2011).

- Permitir que a seleção de trajetórias utilize mais de um ponto como referência.
  Assim será possível selecionar trajetórias que cruzaram mais de um ponto em comum.

- Filtros de ruídos.
  Na literatura existem outros filtros mais sofisticados que a média e mediana, como os filtros de Kalman e de partículas, que consideram a dinâmica da trajetória (ZHENG; ZHOU, 2011).

- Melhorias na usabilidade do sistema.
  Uso de ícones ou guias de ajuda podem facilitar o uso do sistema.

- Funcionalidade de Log do sistema.
  Um recurso de log pode fornecer informações importantes durante e após o processamento dos dados, (ex., quantidade de registros carregados, tempo decorrido, uso de recursos do sistema, etc.)

REFERÊNCIAS

AGRAWAL, R.; FALOUTSOS, C.; SWAMI, A. N. Efficient similarity search in sequence databases. In: *Proceedings of the 4th International Conference on Foundations of Data Organization and Algorithms*. London, UK, UK: Springer-Verlag, 1993. (FODO '93), p. 69–84. ISBN 3-540-57301-1. <http://dl.acm.org/citation.cfm?id=645415.652239>.

ALVARES, L. et al. An algorithm to identify avoidance behavior in moving object trajectories. *Journal of the Brazilian Computer Society*, Springer-Verlag, v. 17, n. 3, p. 193–203, 2011. ISSN 0104-6500. <http://dx.doi.org/10.1007/s13173-011-0037-3>.

ALVARES, L. O. et al. A framework for trajectory data preprocessing for data mining. In: *SEKE*. Boston,Massachusetts, USA: Knowledge Systems Institute Graduate School, 2009. p. 698–702. ISBN 1-891706-24-1.

AMICI, R. et al. Performance assessment of an epidemic protocol in vanet using real traces. *Procedia Computer Science*, v. 40, n. Supplement C, p. 92 – 99, 2014. ISSN 1877-0509. Fourth International Conference on Selected Topics in Mobile Wireless Networking (MoWNet'2014). <http://www.sciencedirect.com/science/article/pii/S1877050914014021>.

AVANCINI, H. M. *Análise da Redução de Dados em Trajetórias de Objetos Móveis*. 127 p. — Universidade Federal de Santa Catarina, Florianópolis, Brasil, 2010.

BERNDT, D. J.; CLIFFORD, J. Using dynamic time warping to find patterns in time series. In: FAYYAD, U. M.; UTHURUSAMY, R. (Ed.). *KDD Workshop*. Seattle, WA, EUA: AAAI Press, 1994. p. 359–370. ISBN 0-929280-73-3.

BOGORNY, V.; BRAZ, F. J. *Introdução a trajetórias de Objetos Móveis: conceitos, armazenamento e análise de dados.* Joinville, SC, Brasil: Ed. Univille, 2012. 116 p.

BRAY, T. *The JavaScript Object Notation (JSON) Data Interchange Format*. EUA, March 2014. 1-16 p. <https://tools.ietf.org/html/rfc7159>.

CARBONI, E.; BOGORNY, V. Inferring drivers behavior through trajectory analysis. In: ANGELOV, P. et al. (Ed.). *Intelligent Systems'2014*. Springer International Publishing, 2015, (Advances in Intelligent Systems and Computing, v. 322). p. 837–848. ISBN 978-3-319-11312-8. <http://dx.doi.org/10.1007/978-3-319-11313-5_73>.

CHEN, L.; ÖZSU, M. T.; ORIA, V. Robust and fast similarity search for moving object trajectories. In: *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*. New York, NY, USA: ACM, 2005. (SIGMOD '05), p. 491–502. ISBN 1-59593-060-4.

CHEN, Z. et al. Searching trajectories by locations: An efficiency study. In: *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*. New York, NY, USA: ACM, 2010. (SIGMOD '10), p. 255–266. ISBN 978-1-4503-0032-2. <http://doi.acm.org/10.1145/1807167.1807197>.

ESLING, P.; AGON, C. Time-series data mining. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 45, n. 1, p. 12:1–12:34, dez. 2012. ISSN 0360-0300. <http://doi.acm.org/10.1145/2379776.2379788>.

ESTER, M. et al. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. Portland, Oregon, EUA: AAAI Press, 1996. (KDD'96), p. 226–231. <http://dl.acm.org/citation.cfm?id=3001460.3001507>.

FERREIRA, E. Z. *Sistema de Monitoramento e Análise de Comportamento de Bovinos*. 41 p. — Universidade Federal de Santa Catarina, Florianópolis, Brasil, 2013.

FONTES, V. C.; BOGORNY, V. Discovering semantic spatial and spatio-temporal outliers from moving object trajectories. Florianópolis, Brasil, p. 28, 2013.

FURTADO, A. S. *Análise de Mobilidade a Partir de Trajetórias GPS de Ônibus*. 79 p. — Universidade do Estado de Santa Catarina, Florianópolis, Brasil, 2014.

FURTADO, A. S. et al. Multidimensional similarity measuring for semantic trajectories. *Transactions in GIS*, 2015. ISSN 1467-9671. <http://dx.doi.org/10.1111/tgis.12156>.

GIANNOTTI, F.; PEDRESCHI, D. *Mobility, Data Mining and Privacy*. Berlin, Alemanha: Ed. Springer, 2008. 412 p.

HWANG, J.-R. et al. Spatio-temporal similarity analysis between trajectories on road networks. In: _____. *Perspectives in Conceptual Modeling: ER 2005 Workshops AOIS, BP-UML, CoMoGIS, eCOMO, and QoIS, Klagenfurt, Austria, October 24-28, 2005. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005. p. 280–289. ISBN 978-3-540-32239-9.

KEOGH, E. Exact indexing of dynamic time warping. In: *Proceedings of the 28th International Conference on Very Large Data Bases*. VLDB Endowment, 2002. (VLDB '02), p. 406–417. <http://dl.acm.org/citation.cfm?id=1287369.1287405>.

KEOGH, E.; RATANAMAHATANA, C. A. Exact indexing of dynamic time warping. *Knowledge and Information Systems*, v. 7, n. 3, p. 358–386, 2005. ISSN 0219-3116. <http://dx.doi.org/10.1007/s10115-004-0154-9>.

KIERMEIER, M.; WERNER, M. Similarity Search for Spatial Trajectories Using Online Lower Bounding DTW and Presorting Strategies. In: SCHEWE, S.; SCHNEIDER, T.; WIJSEN, J. (Ed.). *24th International Symposium on Temporal Representation and Reasoning (TIME 2017)*. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017. (Leibniz International Proceedings in Informatics (LIPIcs), v. 90), p. 18:1–18:15. ISBN 978-3-95977-052-1. ISSN 1868-8969. <http://drops.dagstuhl.de/opus/volltexte/2017/7919>.

LIU, H.; SCHNEIDER, M. Similarity measurement of moving object trajectories. In: *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on GeoStreaming*. New York, NY, USA: ACM, 2012. (IWGS '12), p. 19–22. ISBN 978-1-4503-1695-8. <http://doi.acm.org/10.1145/2442968.2442971>.

LONGLEY, P. A. et al. *Geographic Information Systems and Science*. Chichester, England: Wiley, 2005. ISBN 0470870001.

PARENT, C. et al. Semantic trajectories modeling and analysis. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 45, n. 4, p. 42:1–42:32, ago. 2013. ISSN 0360-0300. <http://doi.acm.org/10.1145/2501654.2501656>.

PIORKOWSKI, M.; SARAFIJANOVIC-DJUKIC, N.; GROSSGLAUSER, M. *CRAWDAD dataset epfl/mobility (v. 2009-02-24)*. feb 2009. Downloaded from http://crawdad.org/epfl/mobility/20090224.

SANTOS, A. A. d. *Descoberta de Padrões de Encontro em Trajetórias de Objetos Móveis*. 57 p. — Universidade Federal de Santa Catarina, Florianópolis, Brasil, 2013.

SHAFRANOVICH, Y. *Common Format and MIME Type for Comma-Separated Values (CSV) Files*. EUA, October 2005. 1-8 p. <https://tools.ietf.org/html/rfc4180.txt>.

SHAW, P. *GIS Succinctly*. Syncfusion, 2013. <https://www.syncfusion.com/resources/techportal/details/ebooks/gis>.

SHEN, H. et al. Protecting trajectory privacy: A user-centric analysis. *Journal of Network and Computer Applications*, v. 82, n. Supplement C, p. 128 – 139, 2017. ISSN 1084-8045. <http://www.sciencedirect.com/science/article/pii/S1084804517300413>.

SOCIETY, W. *FREQUENCY 1550 - Mobile Learning Game*. 2005. [Online; acessado 26-Outubro-2017]. <http://freq1550.waag.org/preview.html>.

SPACCAPIETRA, S. et al. A conceptual view on trajectories. *Data Knowl. Eng.*, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 65, n. 1, p. 126–146, abr. 2008. ISSN 0169-023X. <http://dx.doi.org/10.1016/j.datak.2007.10.008>.

TAN, P.-N.; STEINBACH, M.; KUMAR, V. *Introduction to Data Mining*. Boston, MA, USA: Pearson, 2005. ISBN 0321321367.

XIAO, X. et al. Inferring social ties between users with human location history. *Journal of Ambient Intelligence and Humanized Computing*, Springer Berlin Heidelberg, v. 5, n. 1, p. 3–19, 2014. ISSN 1868-5137. <http://dx.doi.org/10.1007/s12652-012-0117-z>.

XU, G. *GPS: Theory, Algorithms and Applications*. Berlin, Germany: Springer, 2007. ISBN 978-3-540-72714-9.

YAN, Z. et al. A hybrid model and computing platform for spatio-semantic trajectories. In: *Proceedings of the 7th International Conference on The Semantic Web: Research and Applications - Volume Part I*. Berlin, Heidelberg: Springer-Verlag, 2010. (ESWC'10), p. 60–75. ISBN 3-642-13485-8, 978-3-642-13485-2. <http://dx.doi.org/10.1007/978-3-642-13486-9_5>.

YUAN, J. et al. Driving with knowledge from the physical world. In: *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2011. (KDD '11), p. 316–324. ISBN 978-1-4503-0813-7. <http://doi.acm.org/10.1145/2020408.2020462>.

ZHENG, Y. et al. Geolife: Managing and understanding your past life over maps. In: *Proceedings of the The Ninth International Conference on Mobile Data Management*. Washington, DC, USA: IEEE Computer Society, 2008. (MDM '08), p. 211–212. ISBN 978-0-7695-3154-0. <https://doi.org/10.1109/MDM.2008.20>.

ZHENG, Y. et al. Mining interesting locations and travel sequences from gps trajectories. In: *Proceedings of the 18th International Conference on World Wide Web*. New York, NY, USA: ACM, 2009. (WWW '09), p. 791–800. ISBN 978-1-60558-487-4. <http://doi.acm.org/10.1145/1526709.1526816>.

ZHENG, Y.; ZHOU, X. *Computing with Spatial Trajectories*. 1st. ed. New York, NY, EUA: Springer Publishing Company, Incorporated, 2011. 328 p. ISBN 1461416280, 9781461416289.

APÊNDICE A – Código Fonte do Sistema Desenvolvido

O código do sistema desenvolvido está disponível na internet através do repositório GitHub no link: https://github.com/RicardoMaurici/trajectory-preprocessing-tool

APÊNDICE B – Artigo do Trabalho Desenvolvido

Pré-Processamento de Dados de Trajetórias para Mineração de Dados e Análise de Similaridade

Ricardo M. Ferreira[1]

[1]Departamento de Informática e Estatística – Universidade Federal de Santa Catarina (UFSC)
Caixa Postal 476 – 88.040-900 – Florianópolis – SC – Brazil

Abstract. *This paper proposes a system that combines preprocessing techniques for moving object trajectory data. We implemented several preprocessing techinques to load, to organize and to clean trajectory data. The system supports trajectory segmentation, which aims to eliminate large time intervals across the trajectory and noise removal, which find and remove outliers points. To validate our proposal, we use several public databases available on internet to load and organize them. and then we evaluate the preprocessing techinques by using a similarity analysis before and after applying preprocessing techinques. The results show that by using preprocessing techiniques we significantly improve the simiarity analysis.*

Resumo. *Este artigo apresenta a proposta de um sistema que reúna técnicas para pré-processamento de dados de trajetórias de objetos móveis. Entre essas técnicas, temos o carregamento e organização de dados de trajetórias. A segmentação de trajetórias, que visa eliminar grandes intervalos de tempo durante a trajetória. A remoção de ruídos, que busca eliminar ou suavizar pontos distantes do restante da trajetória. Para validação do sistema, foram então carregadas e organizadas bases de dados públicas disponíveis na internet. Por fim uma análise de similaridade sobre esses dados permitiu observar a variação do grau de similaridade das trajetórias através das técnicas de pré-processamento disponíveis no sistema.*

1. Introdução

O aumento no uso de dispositivos móveis dotados de sensores GPS permite que, cada vez mais, indivíduos tenham seus movimentos registrados. Esses dispositivos podem ser configurados para registrar a localização do indivíduo a cada determinado período de tempo. Criando assim uma sequência de localizações ao longo do tempo na forma de uma trajetória.

Ao longo da última década, diversos trabalhos foram realizados sobre trajetórias, como a descoberta de padrões de trajetórias de objetos móveis que desviam de objetos estáticos [Alvares et al. 2011] e a identificação de motoristas perigosos [Carboni and Bogorny 2015]. Ainda a partir das trajetórias também é possível analisar a similaridade de dois objetos. Neste contexto, uma medida de similaridade é uma métrica que específica se dois objetos são semelhantes um ao outro, de acordo com as características de suas trajetórias [Zheng and Zhou 2011]. Alguns trabalhos no âmbito da similaridade propõem medidas de similaridade para trajetórias [Xiao et al. 2014, Furtado et al. 2015]. Porém tanto a descoberta e a identificação de padrões, quanto a

análise das trajetórias, estão sujeitas a problemas e situações recorrentes no processo de coleta que interferem diretamente na estrutura e na qualidade dos dados.

Esses problemas podem ser influenciados por uma série de fatores durante a coleta de trajetórias, como: i) condições climáticas, que podem causar variações na precisão do ponto coletado; ii) a interferência de prédios altos, túneis ou qualquer grande obstáculo pode interferir no sinal do GPS, criando um intervalo na trajetória; e iii) a intervenção deliberada do indivíduo na configuração e utilização do dispositivo usado na coleta.

Em razão dos fatores externos aos quais a trajetória esteve sujeita durante a coleta, é necessário realizar um pré-processamento para que essa possa ser utilizada em uma análise [Furtado 2014]. Esse pré-processamento garante a estrutura e a qualidade de dados necessária para o domínio de aplicação.

Nesse cenário esse trabalho propõe um sistema que reúna técnicas de pré-processamento de dados de trajetórias disponíveis na literatura, contemplando o carregamento, limpeza e organização dessas trajetórias.

O restante desse artigo está estruturado da seguinte forma: a Seção 2 apresenta alguns conceitos de trajetórias de objetos móveis e o processo de coleta desses dados. A Seção 3 apresenta o sistema desenvolvido, bem como as técnicas de pré-processamento adotadas. A Seção 4 apresenta a validação, através do carregamento de bases de dados e da análise de similaridade dos dados pré-processados pelo sistema. Por fim, a Seção 5 conclui o documento e sugere orientações de trabalhos futuros.

2. Conceitos Básicos

Nessa Seção é apresentado o conceito de Trajetórias de Objetos Móveis, o processo e recursos utilizados para coleta desses dados.

2.1. Trajetórias de Objetos Móveis

Alguns objetos possuem localização fixa no espaço, por exemplo, prédios e monumentos históricos. Entretanto determinados objetos não permanecem fixos no mesmo local, estes objetos se movem e não permitem que tenham uma localização fixa associada a eles. Por exemplo, um carro parado possui uma localização, mas quando em movimento, sua localização muda ao longo do tempo.

Dessa forma a sequência de localizações registradas para cada objeto móvel é chamada de trajetória. Uma trajetória é representada por um conjunto de pontos ao longo do tempo $((x_1, y_1), t_1), ((x_2, y_2), t_2), ..., ((x_n, y_n), t_n))$, onde (x_i, y_i) é a i-ésima coordenada espacial, t_i é o instante de tempo associado a essa coordenada e n é o número de pontos da trajetória [Bogorny and Braz 2012].

2.2. Processo de Coleta de Trajetórias

O processo de coleta está diretamente ligado a forma como o sensor GPS é gerenciado para obtenção dos dados de localização, podendo ser caracterizada como ativa ou passiva [Zheng and Zhou 2011]. Uma coleta ativa ocorre quando é necessária a intervenção humana para ligar e desligar o equipamento, como, por exemplo, um grupo de pessoas que ativam, individualmente, a coleta de suas trajetórias em seus aparelhos com sensores GPS, iniciam uma caminhada e desativam quando desejarem [Santos 2013, Society 2005]. Por

outro lado, a coleta passiva é quando o equipamento com GPS é ativado e desligado automaticamente. Por exemplo, um automóvel ativa o sensor GPS assim que o motor é ligado e coleta sua trajetória durante o período em que o motor estiver ligado [Avancini 2010].

2.3. Recursos para Coleta de Dados de Trajetórias

Aplicativos nos dispositivos móveis utilizam o sensor GPS do aparelho para coletar as coordenadas geográficas de sua localização. Através destas coordenadas, e da sequência de coordenadas coletadas, com a variação de posições ao longo do tempo, é possível realizar cálculos para determinar velocidade, distância e outras informações da trajetória. Há duas alternativas para registrar estes dados, que são: i) o dado é enviado para uma base centralizada conforme é coletado, porém envolve o uso de algum tipo de rede, sujeito a falhas e largura de banda. e ii) armazenar os dados no próprio aparelho e realizar a exportação dos dados para uma base centralizada através de arquivos [Giannotti and Pedreschi 2008]. A segunda alternativa é a mais utilizada em aplicações GPS e os formatos de dados mais comuns para exportação de dados são: DSV, JSON, KML e GPX.

3. Proposta e Desenvolvimento

O pré-processamento dos dados de trajetórias consiste na aplicação de diferentes estratégias e técnicas para realizar as tarefas de carregamento, conversão, limpeza e organização desses dados [Tan et al. 2005]. Esse conjunto de tarefas geralmente segue uma ordem de execução, porém a execução de algumas delas é opcional, como por exemplo, realizar apenas a tarefa de limpeza em uma base previamente carregada.

Nesse processo a tarefa de carregamento é responsável por realizar a importação dos dados de fontes externas, realizando as conversões de dados necessárias e/ou criação de dados faltantes para correções de possíveis problemas de estruturação dos dados. Por exemplo, no momento do carregamento pode ser necessário realizar a conversão das coordenadas do modelo de referência geográfica WGS84[1] para o modelo EPSG:3857[2] e/ou criar um identificador para cada arquivo que está sendo carregado. Desse modo para determinado conjunto de arquivos, em que cada arquivo é considerado uma trajetória única, será criada uma coluna adicional no banco de dados para o identificador de trajetória (*tid*), registrando então um valor *tid* único para os dados oriundos de cada arquivo [Furtado 2014].

Assim esse trabalho propõe um sistema do tipo *desktop*, desenvolvido em Java, para pré-processamento dos dados de trajetórias. Esse sistema utiliza o banco de dados PostgreSQL, com a sua extensão espacial PostGIS e possui 3 principais módulos. O Módulo de Carregamento de dados, Módulo de Limpeza de Dados e o Módulo de Organização e Segmentação de Trajetórias.

3.1. Módulo de Carregamento de Dados

Esse módulo permite realizar a importação de dados externos em diferentes formatos (DSV, JSON, GPX, KML e WKT) para o banco de dados com o qual uma conexão foi previamente estabelecida. Dessa forma, o módulo recebe um arquivo ou diretório como entrada, processa de acordo com um conjunto de parâmetros fornecidos pelo usuário e

[1]http://spatialreference.org/ref/epsg/wgs-84/

[2]http://wiki.openstreetmap.org/wiki/EPSG:3857

realiza a inserção em determinada tabela do banco de dados. A Figura 1 apresenta a tela de carregamento de dados do tipo DSV e os parâmetros a serem fornecidos pelo usuário.

Quando um diretório é fornecido como entrada, ele e todos os seus subdiretórios são percorridos pelo sistema em busca de arquivos que contenham dados de trajetórias. Dessa forma, os campos (*Extensions*), (*Ignore directories*), e (*Ignore files*) fornecem as informações necessárias para percorrer os diretórios e selecionar os arquivos. Assim, é possível ignorar ou considerar determinados arquivos, diretórios ou extensões de arquivos durante o processo.

Os campos (*Start after line*) e (*Delimiter*) permitem determinar um número de linhas a ser ignorado no início dos arquivos (ex., alguns aplicativos incluem linhas de cabeçalho antes dos dados relativos à trajetória) e saber qual seu delimitador padrão (ex., ponto, ponto e vírgula, etc.). Os campos de formato de data e hora, bem como o código identificador do sistema de projeção geográfico (SRID) permitem a correta formatação e transformação dos dados durante o carregamento, garantindo a padronização no banco de dados.

O campo de seleção (*Save Metadata*) permite a criação de mais uma coluna na tabela no banco de dados chamada *path*, para então registrar o caminho completo do arquivo de origem do dado e uma coluna *folder_id*, onde é criado um identificador único para cada diretório lido. Já os campos de seleção (*Generate serial GID*) e (*Generate serial TID*) são para geração de identificadores únicos para os pontos e trajetórias. O campo (*Generate serial GID*) cria uma *sequence* e atualiza todas tuplas do banco de dados com os identificadores. Já o campo (*Generate serial TID*) cria uma *sequence* e insere o mesmo valor para todos os registros de cada arquivo lido, pegando o próximo valor da *sequence* na leitura do próximo arquivo, identificando assim todos os registros de um arquivo como uma única trajetória.

A tela do sistema permite flexibilidade quanto a criação da tabela, nela é possível especificar detalhes das colunas e o nome da tabela que será criada, bem como relacionar o nome da coluna no banco de dados com a posição da coluna no arquivo DSV de origem do dado.

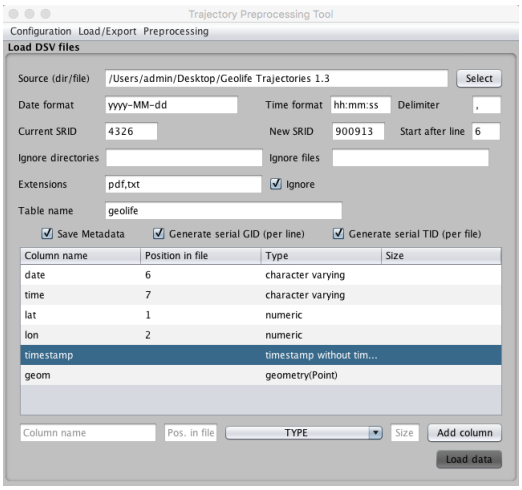

Figura 1. Tela do módulo para carregamento de dados DSV

3.2. Módulo de Limpeza de Dados

Esse módulo é responsável pela aplicação das diferentes técnicas de remoção de ruídos. A Figura 2 apresenta a tela para remoção de ruídos com filtro por velocidade, filtro por densidade e suavização através da média e mediana. Assim essa tela permite especificar uma tabela no banco de dados e identificar os tipos de colunas da tabela no banco de dados. Nesse processo as colunas dos tipos *geom* e *tid* são as mais importantes, pois é a partir delas que será realizada a identificação das trajetórias e a obtenção da coordenada geográfica para o cálculo da velocidade e distância entre os pontos.

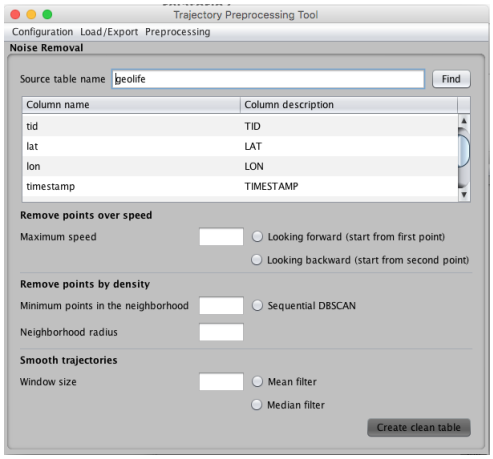

Figura 2. Tela do módulo para remoção de ruídos em trajetórias

Todas as técnicas implementadas no sistema tratam uma trajetória por vez, sem considerar as demais trajetórias no banco de dados. Assim uma trajetória é fornecida como entrada do algoritmo e esse percorre seus pontos suavizando a trajetória ou identificando e removendo ruídos.

Filtro por velocidade: Esse filtro considera ruído um ponto que está acima da velocidade informada, em metros por segundo, no campo *Maximum speed*. O processo para identificar o ruído é através da verificação da velocidade entre o ponto atual e o ponto seguinte. A partir da constatação de uma velocidade acima da informada, então o algoritmo permite duas opções de exclusão do ruído. Pode-se optar por excluir o ponto atual, opção *Looking forward (start from first point)*, ou o ponto seguinte, opção *Looking backward (start from second point)*. Essas duas abordagens permitem escolhas de acordo com o padrão de ruídos na base de dados. Por exemplo, algumas trajetórias concentram ruídos nos primeiros pontos da trajetória, geralmente ocorridos ainda na sincronização do aparelho GPS e assim pode-se optar pela remoção do primeiro ponto.

Filtro por densidade: Essa abordagem é uma variação do DBSCAN [Ester et al. 1996] para encontrar ruídos em uma trajetória. O algoritmo verifica, para cada ponto, a existência de uma quantidade minima de pontos em sua vizinhança, informada no campo *Minimum points in the neighborhood*, dentro de um raio, informado em metros no campo *Neighborhood radius*. Então é calculada a distância Euclidiana do ponto atual até seus pontos seguintes e anteriores, dentro do raio, a fim de verificar a quantidade miníma de pontos em sua vizinhança. Quando ocorrer de

um ponto não atingir a quantidade miníma de pontos em sua vizinhança, então este é considerado ruído e é removido da trajetória.

Suavização pela média e mediana: Esses filtros permitem suavizar a trajetória no sentido de manter os pontos mais alinhados. Para suavizar uma trajetória é necessário calcular novos valores para os eixos *x* e *y* de cada ponto da trajetória. Esse cálculo é através da média ou mediana de uma janela de pontos na proximidade do ponto alvo. Dessa forma o algoritmo utiliza um tamanho de janela, informado pelo campo *Window size*, e obtém os pontos na proximidade, anteriores e seguintes, a fim de formar o tamanho da janela. O cálculo da média ou mediana é realizado através dos valores *x* e *y* de todos os elementos que compõem a janela. Por fim, o resultado desse cálculo é atribuído ao *x* e *y* do ponto alvo e repetido todo processo para cada ponto até o fim da trajetória.

Para cada aplicação de um desses filtros é então criada uma nova tabela no banco de dados, com nome baseado na tabela origem, na técnica e parâmetros utilizados. Por exemplo, *nomeTabelaOrigem_removednoise_median_30*, onde *nomeTabelaOrigem* é o nome da tabela de origem das trajetórias, *median* é a técnica de suavização por mediana e 30 é o parâmetro fornecido para o tamanho da janela.

3.3. Módulo de Organização e Segmentação de Dados

O módulo de organização e segmentação de trajetórias, ilustrado pela Figura 3, permite realizar a segmentação das trajetórias. Para essa segmentação é criada uma nova coluna, de nome *old_tid*, no banco de dados para onde serão copiados todos os valores *tid* originais das trajetórias, preservando assim os identificadores originais. A partir disso o sistema poderá realizar as três segmentações possíveis, porém seguirá a seguinte ordem: primeiro a segmentação por estado do objeto, se o campo *Segment by status change* estiver selecionado e possuir um atributo de valor 0 ou 1 no banco de dados, de forma a identificar se está ocupado ou não. Em seguida será segmentada por tempo, se um valor em segundos for especificado em *Maximum sampling time gap*, onde é verificado o intervalo de tempo entre os pontos. E por último a segmentação por distância entre os pontos, se um valor em metros for especificado em *Maximum distance gap*. Dessa forma as segmentações por intervalos de tempo e por distância visam eliminar grandes intervalos de tempo ou espaço durante uma trajetória.

Nesse processo cada técnica de segmentação criará temporariamente uma coluna própria no banco de dados. Para segmentação por estado será criada a coluna *status_tid*, para segmentação por tempo *sample_tid* e *distance_tid* para segmentação por distância. A segmentação por estado utiliza os identificadores de trajetórias da coluna *tid* e salva os novos identificadores da segmentação na coluna *status_tid*. As outras duas técnicas utilizam a coluna da técnica anterior, se esta ocorreu, como identificador da trajetória. Caso não ocorra nenhuma técnica anteriormente, então é utilizado a coluna *tid* como referência para o identificador da trajetória. Ao término das segmentações os valores da coluna da última segmentação serão copiados para a coluna *tid* e as colunas temporárias serão excluídas. Dessa forma as trajetórias recebem novos identificadores e preservam os identificadores originais da base de dados na coluna *old_tid*.

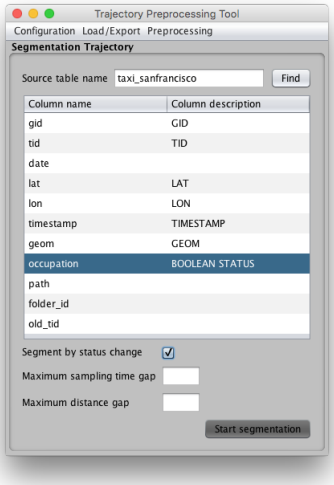

Figura 3. Tela do módulo para segmentação de trajetórias

4. Validação do Sistema Desenvolvido

Atualmente, diversas bases de dados de trajetórias estão disponíveis gratuitamente na internet. Essas bases são utilizadas por pesquisadores em todo o mundo e têm seus dados disponíveis em formatos distintos, conforme disponibilizadas por seus mantenedores e não mantendo padronização entre elas. Assim toda vez que alguém deseja manipular os dados dessas bases, deve-se então realizar um esforço para compreender a estrutura dos dados e então realizar a importação para um banco de dados.

Por esse motivo, através de uma busca, foram reunidas 14 bases de dados de trajetórias para serem pré-processadas pelo sistema desenvolvido de forma a validá-lo. A partir dessas bases de dados carregadas foi então realizada uma avaliação do impacto do uso de técnicas de pré-processamento no cálculo da similaridade entre trajetórias. Essa avaliação é feita através de uma análise da variação do grau de similaridade entre as trajetórias, em função das transformações realizadas, nas bases de dados originais e segmentadas.

Para avaliar a similaridade entre trajetórias é preciso utilizar uma medida de similaridade que quantifica o quão similares ou distantes duas trajetórias são [Keogh and Ratanamahatana 2005]. Após a quantificação da similaridade o resultado da medida pode ser utilizado para agrupar trajetórias para uso em determinadas aplicações, pois em muitos casos as trajetórias possuem muitas informações associadas e é preciso agrupar trajetórias segundo alguma similaridade entre elas [Hwang et al. 2005].

Para isso existem na literatura uma série de medidas de distância e similaridade para trajetórias [Zheng and Zhou 2011]. Essas medidas são baseadas no cálculo de distância entre os pares de objetos do conjunto de dados. Nesse processo, é aplicada a função de distância nos valores dos atributos de interesse sobre os objetos. Com o uso de uma determinada medida de distância, é possível descobrir a similaridade entre dois objetos, ou seja, podem-se identificar objetos que mais se aproximam, de acordo com suas características [Esling and Agon 2012]. Entre essas medidas podemos citar *Sum-of-Pairs Distance* [Agrawal et al. 1993], *Dyna-*

mic Time Warping (DTW) [Berndt and Clifford 1994], *Longest Common Subsequence* (LCSS) [Zheng et al. 2008], *Edit Distance with Real Penalty* (ERP) [Chen et al. 2010] e *Edit Distance on Real Sequences* (EDR) [Chen et al. 2005].

Esse trabalho utiliza a medida de similaridade EDR, pois é uma medida amplamente utilizada na literatura para similariadde entre trajetórias. A medida *Edit Distance on Real Sequence* (EDR) calcula a distância entre duas trajetórias contando o número de operações de edição (inserir, excluir e substituir) que são necessárias para transformar uma trajetória em outra. A medida de distância EDR utiliza um valor ε como limiar de aceitação de distância para decidir se dois pontos são similares. Portanto se a distância entre dois pontos for menor que ε, então esses são considerados similares e se estiverem mais afastados, serão considerados diferentes. Dessa forma, dadas duas trajetórias A e B com comprimentos de n e m, respectivamente, com limiar de aceitação ε. Então a distância EDR entre A e B é o número de inserção, exclusão ou substituição de operações necessárias para alterar A em B [Chen et al. 2005].

$$EDR(A,B) = \begin{cases} n & \text{se } m = 0 \\ m & \text{se } n = 0 \\ \text{mínimo}\{\text{EDR(A), sobra(B)} + custo, & \text{senão} \\ \text{EDR(sobra(A), B)} + 1, \text{EDR(A, sobra(B))} + 1\} & \end{cases} \quad (1)$$

onde

$$custo = \begin{cases} 0, \text{se d(início(A), início(B))} \leq \varepsilon \\ 1, \text{senão} \end{cases} \quad (2)$$

$$EDR\ normalizado = \frac{EDR(A,B)}{maior(|A|,|B|)} \quad (3)$$

A distância EDR foi normalizada pelo tamanho da maior trajetória, devido as trajetórias possuirem tamanhos distintos.

$$Similaridade\ EDR = 1 - EDR\ normalizado \quad (4)$$

Já para transformar a distância EDR em uma medida de similaridade, foi então subtraído o valor da distância EDR normalizada de 1. Obtendo-se assim um grau de similaridade EDR entre 0 e 1. Nesse sentido a maior similaridade é 1 e a menor similaridade é 0.

4.1. Avaliação da Variação do Grau de Similaridade de Trajetórias em Diferentes Bases de Dados

Para avaliação da variação do grau de similaridade nas bases de dados foram então utilizadas a versão original e uma versão segmentada por intervalos de 5 minutos da mesma base de dados.

Em cada uma dessas versões de bases de dados foi calculada, para cada trajetória, a média do grau de similaridade das 10 trajetórias mais similares e a média das 10 trajetórias menos similares. Com isso foi formado uma lista com as médias das 10 trajetórias mais similares e menos similares de cada trajetória. A partir dessa lista, com as médias, foi então calculada a média para todas as trajetórias da base de dados. Formando assim a

média das médias das 10 trajetórias mais similares e a média das médias das 10 trajetórias menos similares para cada base de dados.

O intuito da análise de similaridade nessas bases de dados é observar a variação do grau de similaridade das trajetórias através da técnica de Segmentação de Trajetórias disponível no sistema criado por esse trabalho. O esperado é que se obtenha, após a segmentação de trajetórias, um grau de similaridade maior entre as trajetórias mais similares e que o grau de similaridade entre as menos similares diminua. Assim aumentamos a discernibilidade entre as trajetórias, permitindo discernir de forma mais fácil as trajetórias mais similares das menos similares.

A Tabela 1 apresenta o comparativo dos resultados da medida de similaridade EDR nas bases de dados originais e segmentadas em 5 minutos. Para cada base de dados são apresentadas as média das 10 trajetórias mais similares e a média das 10 trajetórias menos similares. Ainda na tabela são apresentadas as variações das médias das 10 trajetórias mais similares entre as bases de dados original e segmentada. Para uso do EDR foi utilizado um limiar de aceitação de 100 metros, ou seja, se a distância entre dois pontos for maior que 100 metros, então esses não serão considerados similares pelo EDR.

Tabela 1. Comparativo das médias das 10 trajetórias mais e menos similares utilizando a medida de similaridade EDR nas bases de dados originais e nas mesmas bases de dados segmentadas em 5 minutos

Base de dados	Original		Segmentada		**Variação da média 10 mais similares**
	Média 10 mais similares	Média 10 menos similares	Média 10 mais similares	Média 10 menos similares	
AIS Brest	0.0336	0.0000	0.8506	0.0000	**0.817**
Athens School Bus	0.0196	0.0196	0.2919	0.0000	**0.2723**
Cruz dataset	0.2781	0.0000	0.2655	0.0000	-0.0126
Dublin Bus	0.2290	0.0000	0.2746	0.0000	**0.0456**
Floripa dataset	0.5861	0.0000	0.7488	0.0000	**0.1627**
GeoLife	0.2343	0.0000	0.2459	0.0000	**0.0116**
Greek Trucks	0.1061	0.0009	0.5877	0.0000	**0.4816**
Greek Trucks rev	0.3595	0.0000	0.5564	0.0000	**0.1969**
NYC buses	0.4555	0.3273	0.5743	0.0000	**0.1188**
Taxi Roma	0.0888	0.0003	0.1321	0.0000	**0.0433**
Taxi San Francisco	0.0555	0.0020	0.1972	0.0000	**0.1417**
T-Drive	0.1044	0.0000	0.3493	0.0000	**0.2449**
Uber San Francisco	0.3360	0.0000	0.3353	0.0000	-0.0007
W4M Milano	0.1120	0.0000	0.4386	0.0000	**0.3266**

Dos resultados obtidos com EDR, entre as 14 bases de dados, apenas 2 não obtiveram aumento na média do grau de similaridade das 10 trajetórias mais similares após a segmentação. As bases de dados *Cruz dataset* e *Uber San Francisco* não obtiveram aumento na média do grau de similaridade pois já foram disponibilizadas pré-processadas, ou seja, o mantenedor da base de dados já havia realizado tratamento e organização dos dados antes da disponibilização pública. Por exemplo, os mantenedores da base de dados Uber San Francisco já haviam retirado pontos redundantes nas trajetórias. Sendo assim tentativas adicionais de pré-processamento, nessas duas bases, podem prejudicar a qualidade dos dados. Já as bases de dados que tiveram maiores variações na média do grau de similaridade das 10 trajetórias mais similares foi devido a eliminação de grandes intervalos de tempo no meio das trajetórias pelo processo de segmentação. Dessa forma *AIS Brest, Athens School Bus, Greek Trucks* e *W4M Milano* estão entre as bases de dados com maiores variações na média do grau de similaridade das 10 trajetórias mais similares após o processo de segmentação.

5. Conclusão e Trabalhos Futuros

O crescente uso de dispositivos móveis dotados com sensores GPS tem permitido, cada vez mais, que indivíduos tenham seus movimentos registrados. A sequência desses registros formam a trajetória bruta do indivíduo. Porém o uso em larga escala dos dispositivos com sensores GPS, principalmente por grandes grupos de pessoas ou frotas de veículos, têm formado numerosas bases de dados dessas trajetórias. Algumas dessas bases de dados estão disponíveis gratuitamente na internet e são utilizadas por pesquisadores em todo o mundo. Entretanto essas bases de dados não possuem padronização e são disponibilizadas conforme o desejo de seus mantenedores. Por esse motivo, sempre que é necessário utilizar alguma dessas bases de dados, é necessário um esforço para compreensão de sua estrutura e para organização em um formato que permita sua utilização.

Na intenção de facilitar esse processo, esse trabalho reuniu técnicas de pré-processamento de dados de trajetórias disponíveis na literatura. Essas técnicas englobam carregamento, organização, segmentação de trajetórias, limpeza de dados e estão reunidas em um sistema do tipo *desktop*, desenvolvido em Java e que utiliza o banco de dados PostgreSQL, com a extensão espacial PostGIS.

Para validar o sistema foram utilizadas 14 bases de dados de trajetórias de diferentes fontes e regiões do planeta. Cada uma dessas bases passou por uma análise para compreensão dos detalhes de sua estrutura e organização dos dados, a fim de fornecer entradas para o sistema carregá-las. Para carregar essas bases de dados pelo sistema, foi então desenvolvido um módulo que é responsável pelo carregamento de dados de trajetórias em diferentes formatos de arquivos. Esse módulo então permitiu definir e criar uma tabela, bem como suas colunas, no banco de dados onde foram carregadas as bases de dados. Através do módulo de Carregamento foram realizados diversos procedimentos sobre os dados. Foram criados os identificadores únicos para cada registro, foram identificadas as trajetórias de acordo com cada base de dados, as datas foram convertidas para se adequarem ao padrão do PostgreSQL e foram definidos no banco de dados o modelo de referência geográfico utilizado na base de dados durante a coleta dos dados.

Um módulo de Segmentação de Trajetórias permitiu que as bases de dados carregadas passassem por um processo de segmentação de suas trajetórias. A segmentação de

trajetórias busca eliminar grandes intervalos de tempo no meio dessas. Assim o módulo de Segmentação permitiu segmentar as trajetórias das bases de dados já carregadas anteriormente. As trajetórias de cada base de dados foram segmentadas quando havia um intervalo igual ou superior a 5 minutos entre dois pontos, separando assim em trajetórias distintas.

Devido algumas bases de dados possuírem ruídos em suas trajetórias, foi então desenvolvido um módulo para remoção de ruídos através de técnicas que filtram esses ruídos. O módulo de Limpeza de Dados suporta os filtros de ruídos por velocidade e densidade de pontos. Já o filtro de suavização da trajetória é fornecido pelo módulo na intenção de não remover nenhum ponto, mas sim de trazer esse ruído para próximo dos demais pontos da trajetória.

Ao final foi realizada uma avaliação do impacto do uso das técnicas de pré-processamento no cálculo da similaridade entre trajetórias. Nessa avaliação foi utilizada a medida de similaridade EDR, que é uma medida de similaridade amplamente utilizada na literatura. Assim, essa avaliação foi feita através da variação do grau de similaridade entre as trajetórias da base de dados original e a mesma segmentada por intervalos de tempo de 5 minutos.

Através da análise de similaridade foi possível perceber um aumento na discernibilidade entre as trajetórias. No sentido que, após a aplicação das técnicas de pré-processamento disponíveis no sistema, fica mais fácil diferenciar as trajetórias mais similares das menos similares entre si. Assim os resultados mostraram que, após o processo de segmentação por tempo, as trajetórias aumentaram seu grau de similaridade em relação as mais similares, enquanto o grau de similaridade diminuiu entre as menos similares.

Como trabalhos futuro podem ser incluídas no sistema outras técnicas de pré-processamento ou melhorias nas existentes, como:

- A funcionalidade de reajustar o intervalo de tempo médio entre os pontos. No sentido que duas trajetórias coletadas com o intervalo de registro configurado para diferentes intervalos possam ficar com a mesma quantidade de pontos, (ex., com a mesma taxa de amostragem através da inserção ou remoção de pontos)
- Compressão de dados de trajetórias. Permitir reduzir a quantidade de pontos em uma trajetória, mas mantendo a sua representação próxima de seu movimento real [Avancini 2010, Zheng and Zhou 2011].
- Remoção de ruídos. Na literatura existem outros filtros de ruídos mais sofisticados que a média e mediana, como os filtros de Kalman e de partículas, que consideram a dinâmica da trajetória [Zheng and Zhou 2011].
- Melhorias na usabilidade do sistema. Uso de ícones ou guias de ajuda podem facilitar o uso do sistema.
- Funcionalidade de log do sistema. Um recurso de log pode fornecer informações importantes durante e após o processamento dos dados, (ex., quantidade de registros carregados, tempo decorrido, uso de recursos do sistema, etc.)

Referências

Agrawal, R., Faloutsos, C., and Swami, A. N. (1993). Efficient similarity search in sequence databases. In *Proceedings of the 4th International Conference on Foundations of Data Organization and Algorithms*, FODO '93, pages 69–84, London, UK, UK. Springer-Verlag.

Alvares, L., Loy, A., Renso, C., and Bogorny, V. (2011). An algorithm to identify avoidance behavior in moving object trajectories. *Journal of the Brazilian Computer Society*, 17(3):193–203.

Avancini, H. M. (2010). Análise da redução de dados em trajetórias de objetos móveis.

Berndt, D. J. and Clifford, J. (1994). Using dynamic time warping to find patterns in time series. In Fayyad, U. M. and Uthurusamy, R., editors, *KDD Workshop*, pages 359–370, Seattle, WA, EUA. AAAI Press.

Bogorny, V. and Braz, F. J. (2012). *Introdução a trajetórias de Objetos Móveis: conceitos, armazenamento e análise de dados.* Ed. Univille, Joinville, SC, Brasil.

Carboni, E. and Bogorny, V. (2015). Inferring drivers behavior through trajectory analysis. In Angelov, P., Atanassov, K., Doukovska, L., Hadjiski, M., Jotsov, V., Kacprzyk, J., Kasabov, N., Sotirov, S., Szmidt, E., and Zadrożny, S., editors, *Intelligent Systems'2014*, volume 322 of *Advances in Intelligent Systems and Computing*, pages 837–848. Springer International Publishing.

Chen, L., Özsu, M. T., and Oria, V. (2005). Robust and fast similarity search for moving object trajectories. In *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*, SIGMOD '05, pages 491–502, New York, NY, USA. ACM.

Chen, Z., Shen, H. T., Zhou, X., Zheng, Y., and Xie, X. (2010). Searching trajectories by locations: An efficiency study. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, SIGMOD '10, pages 255–266, New York, NY, USA. ACM.

Esling, P. and Agon, C. (2012). Time-series data mining. *ACM Comput. Surv.*, 45(1):12:1–12:34.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, pages 226–231, Portland, Oregon, EUA. AAAI Press.

Furtado, A. S. (2014). Análise de mobilidade a partir de trajetórias gps de Ônibus.

Furtado, A. S., Kopanaki, D., Alvares, L. O., and Bogorny, V. (2015). Multidimensional similarity measuring for semantic trajectories. *Transactions in GIS*.

Giannotti, F. and Pedreschi, D. (2008). *Mobility, Data Mining and Privacy*. Ed. Springer, Berlin, Alemanha.

Hwang, J.-R., Kang, H.-Y., Li, Ki-Joune", e.-J., Liddle, S. W., Song, I.-Y., Bertolotto, M., Comyn-Wattiau, I., van den Heuvel, W.-J., Kolp, M., Trujillo, J., Kop, C., and

Mayr, H. C. (2005). *Spatio-temporal Similarity Analysis Between Trajectories on Road Networks*, pages 280–289. Springer Berlin Heidelberg, Berlin, Heidelberg.

Keogh, E. and Ratanamahatana, C. A. (2005). Exact indexing of dynamic time warping. *Knowledge and Information Systems*, 7(3):358–386.

Santos, A. A. d. (2013). Descoberta de padrões de encontro em trajetórias de objetos móveis.

Society, W. (2005). Frequency 1550 - mobile learning game. [Online; acessado 26-Outubro-2017].

Tan, P.-N., Steinbach, M., and Kumar, V. (2005). *Introduction to Data Mining*. Pearson, Boston, MA, USA.

Xiao, X., Zheng, Y., Luo, Q., and Xie, X. (2014). Inferring social ties between users with human location history. *Journal of Ambient Intelligence and Humanized Computing*, 5(1):3–19.

Zheng, Y., Wang, L., Zhang, R., Xie, X., and Ma, W.-Y. (2008). Geolife: Managing and understanding your past life over maps. In *Proceedings of the The Ninth International Conference on Mobile Data Management*, MDM '08, pages 211–212, Washington, DC, USA. IEEE Computer Society.

Zheng, Y. and Zhou, X. (2011). *Computing with Spatial Trajectories*. Springer Publishing Company, Incorporated, New York, NY, EUA, 1st edition.