

**Introdução a trajetórias de objetos móveis:
conceitos, armazenamento e análise de dados**

Editores

Vania Bogorny (Departamento de Informática e Estatística da
Universidade Federal de Santa Catarina, Florianópolis)

Fernando José Braz (Instituto Federal de Educação, Ciência e Tecnologia de Santa
Catarina e Universidade da Região de Joinville)

Autores:

Vania Bogorny

Departamento de Informática e Estatística da Universidade Federal de Santa Catarina,
Florianópolis

E-mail: vania@inf.ufsc.br

Fernando José Braz

Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina e
Universidade da Região de Joinville

E-mail: fernando.braz@ifc-videira.edu.br

Luis Otavio Campos Alvares

Departamento de Informática e Estatística da Universidade Federal de Santa Catarina,
Florianópolis

E-mail: alvares@inf.ufsc.br

Fernando Lucca Siqueira

Departamento de Automação e Sistemas da Universidade Federal de Santa Catarina,
Florianópolis

E-mail: fernandols@inf.ufsc.br

Alisson Moscato Loy

Instituto de Informatica da Universidade Federal do Rio Grande do Sul, Porto Alegre

E-mail: alisson.moscato@gmail.com

Bruno Neiva Moreno

Centro de Informática, Universidade Federal de Pernambuco, Recife

E-mail: brunoneiva@gmail.com

Valéria Cesário Times

Centro de Informática, Universidade Federal de Pernambuco, Recife

E-mail: vct@cin.ufpe.br

Agradecimentos

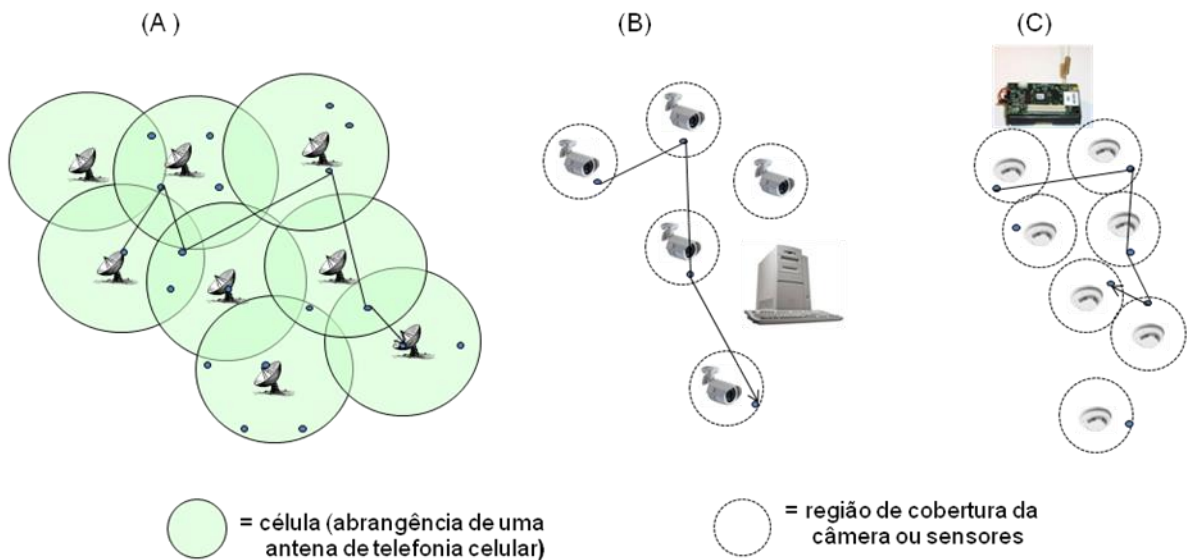
A preparação desta obra não seria possível sem a participação dos colaboradores, que escreveram e revisaram os capítulos. Além disso, o auxílio financeiro da Universidade da Região de Joinville (Univille) foi fundamental para a sua publicação em papel. O Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) também financiou parcialmente um dos editores.

Capítulo 1: Introdução

Vania Bogorny

Os recentes avanços na tecnologia de satélites e a redução no preço de dispositivos móveis, como aparelhos de GPS (sistema de posicionamento global, do inglês *global positioning system*), telefones celulares e redes de sensores, têm aumentado significativamente o uso desses mecanismos. Tais dispositivos registram a mobilidade dos indivíduos que os carregam. Por exemplo, um telefone celular emite um sinal toda vez que seu usuário se move dentro da cidade e passa da região de cobertura de uma antena para outra região, conforme demonstra a figura 1.1 (A). Um objeto móvel pode ser identificado por câmeras de segurança quando passar pela região de cobertura de uma câmera, conforme mostra a figura 1.1 (B). Em uma rede de sensores, cada sensor que registra a passagem de algum objeto pode registrar o lugar e o tempo em que determinado objeto passou por aquele sensor, como ilustrado na figura 1.1 (C).

Figura 1.1 – Exemplos de dispositivos móveis que geram trajetórias

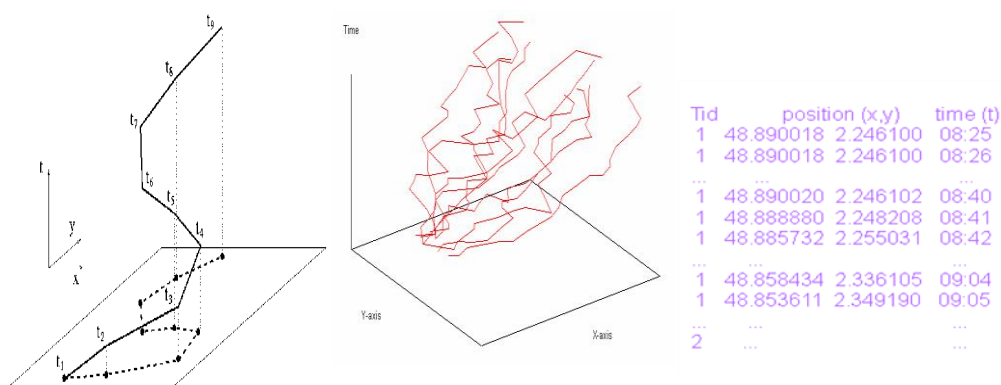


Um aparelho de GPS pode ser configurado para gravar novos dados e consegue, por exemplo, gravar 1 ponto a cada segundo que o objeto que o carrega se move, 1 ponto a cada 10 metros em que o objeto se move, gravar um ponto de 30 em 30 minutos etc. Esse ponto, em geral, representa o local por onde o objeto passou e o horário. Ao contrário dos exemplos da figura 1.1, um GPS coleta muito mais pontos na região por onde o indivíduo se move. Contudo, em todos os exemplos citados, o objeto que carrega o dispositivo deixa para trás a sua “pegada”. A pegada, na maioria dos dispositivos, corresponde a um lugar no espaço em um instante de tempo onde sua passagem foi registrada.

A sequência dos pontos registrados para cada indivíduo é chamada de trajetória do objeto móvel. A trajetória permite o rastreamento do indivíduo, bem como a análise e a interpretação da sua trajetória.

Os dados gerados por dispositivos móveis são dados espaçotemporais chamados de trajetórias brutas. Uma trajetória bruta é representada por meio de um conjunto de pontos (tid, x, y, t) , em que tid é o identificador da trajetória e x, y , uma coordenada geográfica que corresponde a um lugar no espaço num instante de tempo t . A figura 1.2 (esquerda) traz um exemplo de trajetória bruta, em que um enorme número de pontos está associado a uma única trajetória. A figura 1.2 (meio) ilustra um exemplo de um conjunto de trajetórias brutas, enquanto a figura 1.2 (direita) mostra um exemplo dos atributos de uma trajetória.

Figura 1.2 – Uma trajetória, um conjunto de trajetórias e uma tabela de trajetórias



Fonte: Adaptado de Giannotti e Pedreschi (2008)

Trajelórias podem ser geradas por qualquer objeto que carrega um dispositivo, seja um pedestre, um carro, um animal, um barco, um tornado. A consequência do crescente uso de dispositivos móveis nos mais diversos domínios de aplicação é o aumento na geração e na disponibilização de grandes volumes de dados espaçotemporais.

Cada trajetória tem um conjunto de propriedades próprias que descrevem o movimento de um objeto. Segundo Giannotti e Pedreschi (2008), tais propriedades podem ser divididas em características momentâneas e características gerais. As momentâneas referem-se ao estado do objeto em um determinado momento, como, por exemplo, o instante de tempo, a posição espacial, a direção do movimento, a velocidade do objeto, a aceleração etc. Já as características gerais dizem respeito à trajetória como um todo, como, por exemplo, a forma geométrica, o tamanho total da trajetória, a duração, a velocidade média, a distância viajada etc. Também existem propriedades de comparações entre trajetórias, como qual trajetória ocorreu primeiro, qual estava na frente, qual viajou o maior percurso e por mais tempo. As características de tempo e espaço permitem analisar onde e quando algum evento ocorreu.

Trajelórias também podem representar o movimento de diversos objetos diferentes, como um êxodo rural da população migrando para as grandes cidades, a migração de um grupo de pássaros em estações diferentes, o percurso de um tornado, o tráfego de informação na internet etc.

Dados de trajetórias, na forma como são originalmente gerados pelos dispositivos, não possuem muita utilidade, e necessita-se de novas técnicas, métodos e algoritmos para interpretar esses dados e extrair conhecimento útil e interessante em diversos domínios de aplicação.

As enormes quantidades de dados brutos geralmente escondem conhecimento novo e interessante, que pode ser muito útil em processos de tomada de decisão nas mais diversas áreas. Sistemas gerenciadores de bancos de dados convencionais (SGBD) e sistemas de informações tradicionais, em geral, não oferecem suporte para a análise e a mineração de tais dados.

Os dados espaçotemporais estão presentes em uma série de problemas da atualidade que estão despertando o interesse dos cientistas. A questão do aquecimento global é uma área de aplicação que analisa dados espaçotemporais sobre a evolução climática nos últimos anos. As empresas de transporte e planejamento urbano têm o grande desafio de resolver o problema de trânsito, principalmente nas grandes capitais, que, em virtude de recorrentes congestionamentos, resultantes do crescente aumento no número veículos, tende a agravar o problema nos próximos anos.

Outro ponto que vem recebendo a atenção dos cientistas em todo mundo são os desastres naturais, como terremotos, furacões, vulcões, *tsunamis* e desmatamentos; esses últimos acontecem principalmente na região da Amazônia. Tais problemas requerem soluções rápidas de análise e mineração de dados espaçotemporais capazes de prever fenômenos futuros e gerar simulações de rápida evacuação de regiões afetadas por desastres naturais.

O rastreamento de veículos por intermédio de aparelhos de GPS e o de criminosos por meio de telefone celular estão ficando cada vez mais recorrentes. Análise de dados censitários, agrupamento de regiões com determinados tipos de doenças, como câncer, estudos do DNA do corpo humano e muitas outras aplicações reais requerem algoritmos e técnicas de mineração de dados espaciais ou espaçotemporais.

Dados de trajetórias podem ser utilizados em aplicações como planejamento urbano e são úteis para analisar a mobilidade urbana, prever novos pontos de táxi, novas rotas de ônibus e planejar duplicações de rodovias, assim por diante. Na área de segurança, esses dados servem para prever acidentes e interpretar acidentes já ocorridos, bem como interpretar as trajetórias de indivíduos suspeitos de algum tipo de delito.

Os dados gerados por dispositivos móveis estão sendo cada vez mais empregados também na análise do comportamento de humanos e animais. A área de robótica analisa trajetórias de robôs, visando aprimorar o comportamento e o controle deles.

A importância dos problemas anteriormente mencionados tem despertado o interesse geral pela pesquisa na área de trajetórias, incluindo as áreas de bancos de dados (GÜTING; SCHNEIDER, 2005), visualização espaçotemporal (ANDRIENKO; ANDRIENKO; WROBEL, 2007), enriquecimento semântico (BOGORNY; KUIJPERS; ALVARES, 2009), mineração de grandes volumes de dados espaçotemporais (WANG; HSU; LEE, 2005) e *data warehouses* espaçotemporais (PELEKIS *et al.*, 2006; RAFFAETA *et al.*, 2011).

Na área de descoberta de padrões de comportamento em trajetórias existe ainda pouca pesquisa, principalmente no tocante à extração de padrões de comportamento que considerem trajetórias anteriormente enriquecidas com informações semânticas e de contexto. Os mais recentes trabalhos na literatura focam na descoberta de padrões de grupos de trajetórias, analisando basicamente o conjunto de pontos localizados no tempo e no espaço, sem se preocupar com o comportamento ou a interpretação do comportamento de um indivíduo.

Mediante a análise do comportamento de objetos móveis em relação a certos objetos ou a outras trajetórias, será possível, por exemplo:

- Interpretar as atitudes de um presidiário em regime semiaberto (monitorado por tornozeleira equipada por GPS) – se ele se encontra com outros presidiários em

regime semiaberto, se frequenta alguns lugares regularmente, se desvia de câmeras de segurança, se persegue outros indivíduos etc.;

- Interpretar as trajetórias de indivíduos suspeitos de crimes;
- Interpretar o comportamento de animais ameaçados de extinção que, por qualquer razão, sejam monitorados para acompanhar seu deslocamento;
- Interpretar o comportamento de motoristas (bons e maus motoristas), motoristas que tiveram acidente com cargas perigosas ou que transportam cargas valiosas, motoristas que transportam material perecível, como hortifrutigranjeiros;
- Interpretar o perfil de usuários de redes sociais, como Facebook, com base na sua trajetória;
- Descobrir padrões espaciais ou espaçotemporais em mensagens do Twitter, quando o indivíduo que envia a mensagem habilita o uso das coordenadas geográficas pelo celular;
- Analisar trajetórias de usuários da *web* (sequência de *sítes* visitados).

Pelo simples fato de termos cada vez mais dados coletados com informações de tempo e espaço, dezenas de outras aplicações podem fazer uso de trajetórias para análise de comportamento.

1.1 Análise de dados de trajetórias

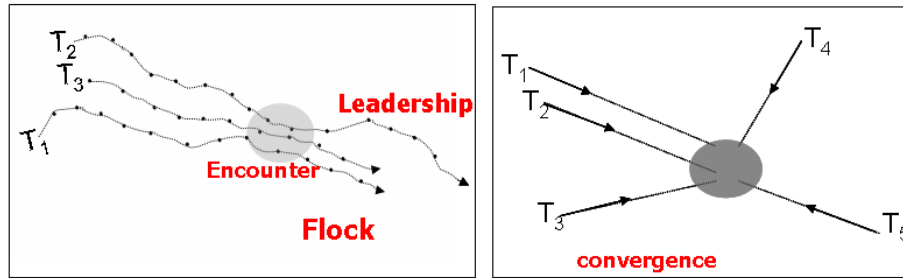
Existem diversos trabalhos na literatura que analisam dados brutos de trajetórias, buscando extrair padrões de comportamento de grupos de trajetórias. Mas o que é um padrão em trajetórias? Um padrão é um comportamento que se repete dentro da mesma trajetória ou entre trajetórias diferentes.

O trabalho de Laube (2002) foi um dos pioneiros na área de análise de comportamento de dados espaçotemporais gerados por dispositivos móveis e tem sido a base de muitas outras pesquisas na área. Nele define-se um padrão para trajetórias que possuem comportamento similar, analisando a mudança de direção. Um padrão deve conter um número mínimo de trajetórias que se movimentam na mesma direção. Nesse estudo os aspectos temporais não são considerados.

Em 2005, Laube (LAUBE; IMFELD; WEIBEL, 2005) propôs um novo tipo de padrão de trajetórias, analisando, além da direção do movimento, a região/localização onde o determinado movimento ocorreu. Definiram-se os quatro tipos mais conhecidos de padrões de trajetórias, os quais nós chamamos de padrões geométricos: *flock*, *leadership*, *convergence* e *encounter*. Na figura 1.3 há um exemplo de cada um desses padrões. O primeiro refere-se a um grupo de objetos que se movem na mesma direção e suas trajetórias estão próximas umas das outras (exemplo: um bando de aves). O padrão *leadership* caracteriza um conjunto de objetos que se movem na mesma direção, suas trajetórias estão próximas umas das outras e um dos objetos lidera o grupo, estando à frente dos demais. O padrão *convergence* (figura 1.3 direita) diz respeito a um grupo de objetos que se movem em direção ao mesmo local (exemplo: estudantes chegando ao *campus* universitário ou usuários acessando o mesmo *site*). O padrão *encounter* caracteriza um grupo de objetos móveis que se deslocam para o mesmo local e ao mesmo tempo.

Diversos outros padrões podem ser encontrados na literatura.

Figura 1.3 – Exemplos de padrões de comportamento em grupos de trajetórias



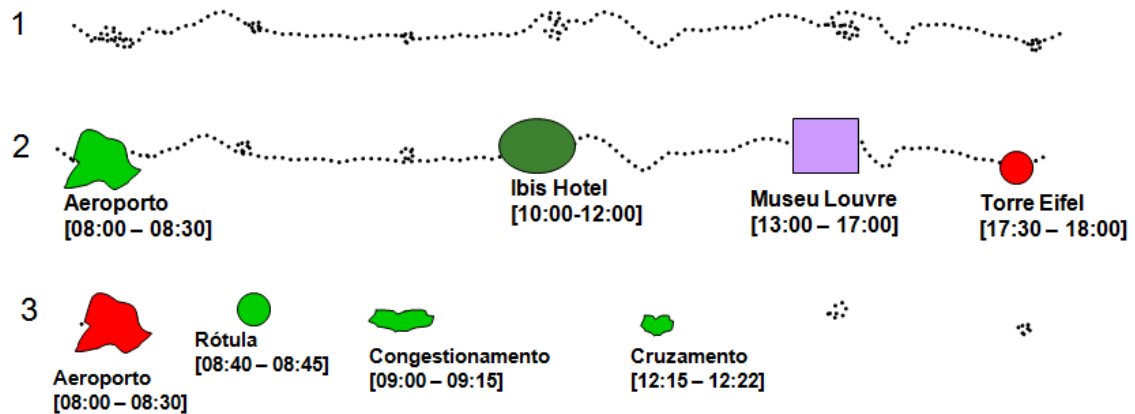
1.2 Informações de contexto / semântica em trajetórias

Por um lado, dados espaçotemporais coletados por meio de aerofotogrametria, sensoriamento remoto ou de imagens de satélite, como, por exemplo, os mapas disponíveis pelo Google Earth, por si só já contêm uma série de informações semânticas associadas que permitem fazer análise e mineração diretamente sobre esses dados. Tal fato já não acontece com os dados gerados por dispositivos móveis como GPS ou telefone celular. Dados gerados por dispositivos móveis são *brutos* e precisam ser semanticamente enriquecidos para que seja possível analisá-los e extrair conhecimento interessante.

A ideia de agregar semântica às trajetórias de objetos móveis para facilitar a análise e a mineração desse tipo de dado foi introduzida pela primeira vez por Alvares *et al.* (2007), com o algoritmo SMOT. Os autores basearam-se no modelo conceitual para trajetórias elaborado pelo grupo de pesquisa do professor Spaccapietra (SPACCAPIETRA *et al.*, 2008), chamado *stops* e *moves*.

Uma trajetória bruta pode ser enriquecida com diferentes informações semânticas, de acordo com o contexto da aplicação e o problema que o usuário pretende resolver. A figura 1.4 traz três exemplos de trajetórias: a primeira é uma trajetória bruta e a segunda e a terceira são semanticamente enriquecidas para diferentes domínios de aplicação. Na trajetória 2, a semântica insere-se no contexto de uma aplicação de turismo, em que os aspectos/lugares importantes são aeroporto, hotel e pontos turísticos (Museu do Louvre e Torre Eiffel). Na trajetória 3, a semântica está relacionada a uma aplicação de trânsito, e as partes interessantes correspondem a lugares movimentados como aeroporto, rotatórias, cruzamentos, semáforos, postos policiais, pedágios etc. Os exemplos citados mostram que, graças à semântica, os mesmos dados brutos podem ser utilizados em diferentes domínios de aplicação. Para isso, é necessário um modelo genérico e flexível para adicionar a semântica desejada (BOGORNÝ; WACHOWICZ, 2008).

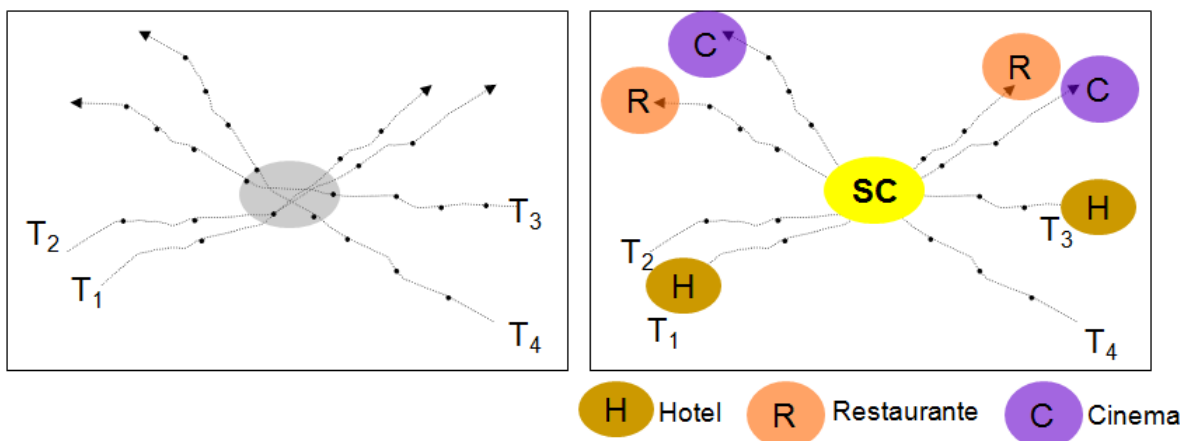
Figura 1.4 – Exemplos de trajetórias: (1) bruta e (2) e (3) semânticas



Ao tentarmos simplesmente analisar uma trajetória bruta, será muito difícil encontrar algum padrão novo e interessante. Consultas sobre dados brutos são custosas e complexas. A mineração de trajetórias brutas busca descobrir padrões geométricos, que podem não ser significativos em determinados domínios de aplicação. Além disso, abordagens que mineram trajetórias brutas são incapazes de descobrir padrões semânticos, pois foram elaboradas para trabalhar com conjuntos de pontos (tid, x, y, t).

Os padrões semânticos podem ser independentes de localização geográfica (podendo estar localizados em lugares esparsos, com poucas trajetórias ou poucos pontos, em vez de densos, lugares com muitos pontos ou trajetórias) e não apresentar nenhuma similaridade. Padrões semânticos dificilmente são extraídos de trajetórias brutas. A semântica também não pode ser agregada às trajetórias após a mineração. Isso está exposto no exemplo da figura 1.5, que apresenta uma analogia entre dados/padrões geométricos e semânticos.

Figura 1.5 – Exemplo de padrões semânticos e geométricos



Na figura 1.5 (esquerda), um padrão geométrico pode ser extraído, considerando a região densa B . Por exemplo, 100% das trajetórias cruzam a região B . Levando em conta a semântica das trajetórias no contexto de uma aplicação de turismo, conforme a figura 1.5 (direita), ao menos dois padrões semânticos podem ser observados: (i) um movimento de

hotel (H) para restaurante (R) passando pela região *B* e (ii) um movimento que vai ao cinema (C) passando pela região *B*.

No exemplo os trabalhos existentes baseados em densidade e similaridade de trajetórias conseguem, no máximo, descobrir que várias trajetórias convergem para a região *B* ou cruzam a região *B*. Sob o ponto de vista de uma aplicação qualquer, esse padrão somente será interessante se *B* for um lugar importante para o domínio do problema. Caso contrário, o padrão será desinteressante para o usuário.

No caso da figura 1.5 (direita), considerando os lugares por onde um ou mais objetos passaram, é possível gerar padrões que caracterizam a sequência de lugares visitados durante a(s) trajetória(s). Por exemplo, 20% das trajetórias que passaram em hotel (H) em seguida passaram pelo centro da cidade (região *B*) e após foram para um restaurante (R).

A área de pesquisa que considera contexto ou semântica em trajetórias ainda está na sua infância. O contexto na área de trajetórias pode estar (i) nas atividades que o objeto móvel realiza, (ii) nos relacionamentos espaçotemporais dessa trajetória com outras trajetórias ou outros objetos geográficos, (iii) na visualização dos dados, (iv) no comportamento, entre outros. O capítulo 3 apresenta alguns métodos bastante simples, desenvolvidos especialmente para o enriquecimento semântico de trajetórias. Tais métodos analisam basicamente as regiões geográficas que os objetos cruzam, as partes das trajetórias que contêm velocidade baixa e as regiões das trajetórias com certa variação na direção.

O restante deste livro está organizado da seguinte forma: O capítulo 2 apresenta os conceitos básicos necessários para entender os trabalhos de armazenamento, processamento e análise de trajetórias. O capítulo 3 traz trabalhos que focam no enriquecimento semântico de trajetórias, enquanto no capítulo 4 há alguns métodos que analisam o comportamento de objetos móveis a partir das suas trajetórias. O capítulo 5 descreve sucintamente a ferramenta Weka-STPM e, finalmente, o capítulo 6 expõe algumas tendências futuras para a área.

Capítulo 2: Processamento e armazenamento

Vania Bogorny, Valéria Cesário Times e Fernando José Braz

Para armazenar e analisar trajetórias são necessários alguns conceitos básicos, que serão apresentados neste capítulo, incluindo a representação de dados espaciais e espaçotemporais, os tipos de relacionamentos espaciais e espaçotemporais e especificidades dos *data warehouses* para esses tipos específicos de dados.

2.1 Bancos de dados espaciais e sistemas de informações geográficas

Um dado geográfico é um elemento do mundo real denominado objeto ou fenômeno geográfico, que tem uma localização com relação à superfície terrestre em um determinado período de tempo. Um dado geográfico também é chamado de feição geográfica.

Um dado espacial é um elemento localizado no espaço e que não necessariamente tem uma relação com a superfície terrestre.

Um objeto geográfico é qualquer elemento que tem uma localização e uma geometria bem definidas, como, por exemplo, um hospital, um bairro, uma rua. Um fenômeno geográfico é uma entidade do mundo real que não tem uma localização e uma geometria bem definidos, como temperatura, relevo, tipos de solo etc.

Objetos e fenômenos geográficos são representados de forma diferente em um banco de dados geográfico. Objetos são normalmente representados por um ponto, uma linha ou um polígono. Fenômenos são representados por grades de células, grades de pontos, isolinhas, redes triangulares ou redes complexas e pontos amostrados irregularmente.

Um objeto geográfico possui três características principais:

- Atributos não espaciais (o quê): descrevem um objeto geográfico qualitativa ou quantitativamente e equivalem aos atributos convencionais em bancos de dados não geográficos. Por exemplo: o nome, a população e o PIB (produto interno bruto) de uma cidade;
- Atributo espacial (onde): refere-se à localização e à representação do objeto geográfico, considerando a geometria (forma do objeto) e um sistema de coordenadas. Tal atributo requer um tipo de dado específico para ser representado, e esse tipo de dado não está disponível em bancos de dados convencionais. Por exemplo, a geometria do município, a geometria de uma rua;
- Relacionamentos espaciais (como): essa característica é especial dos dados geográficos, pois caracteriza relacionamentos de vizinhança espacial entre os objetos. Por exemplo, a distância entre dois objetos, o tipo de intersecção, ordem etc. Os relacionamentos espaciais demandam operadores específicos de bancos de dados geográficos para serem extraídos, visto que normalmente não são armazenados no banco de dados. Os relacionamentos espaciais estão detalhados na próxima seção.

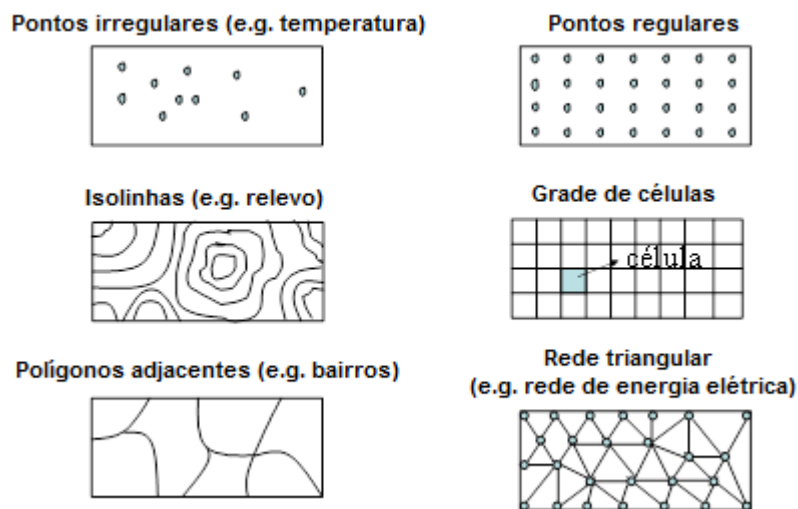
Caracteriza-se a representação espacial dos objetos e fenômenos geográficos de forma diferente. Os objetos geográficos podem ser representados da seguinte forma:

- Ponto: utilizado para representar objetos geográficos 0-dimensionais. Exemplos: hospitais, escolas, postos de combustível etc.;

- Linha: empregada para representar objetos geográficos com a forma espacial unidimensional, representados por linhas ou múltiplas linhas. Exemplos: rios, rodovias, linhas de ônibus etc.;
- Polígono: usado para representar objetos geográficos bidimensionais. Exemplos: lagoas, cidades, bairros, favelas, setores censitários, estados etc.

A figura 2.1 ilustra um exemplo da representação espacial de objetos geográficos do tipo ponto (postos de combustível) e linha (ruas) em um banco de dados geográficos (BDG). Os fenômenos geográficos podem ser representados conforme a figura 2.1.

Figura 2.1 – Exemplos de representação de fenômenos geográficos



Fonte: Adaptado de Câmara *et al.* (1996)

2.1.1 Relacionamentos espaciais

As entidades do mundo real, também conhecidas como objetos ou feições geográficas, estão fortemente relacionadas. Por exemplo, ferrovias podem cruzar rios, rios podem atravessar florestas, estados são agregados de municípios, postos de combustível estão próximos a rodovias, e assim por diante. Tais associações, denominadas relacionamentos espaciais, constituem a principal característica que diferencia dados geográficos de dados convencionais.

Alterações na geometria de qualquer objeto geográfico afetam a geometria ou o comportamento de objetos vizinhos. Por intermédio dessas associações é possível avaliar a influência que um objeto exerce sobre outro com o qual está geograficamente relacionado.

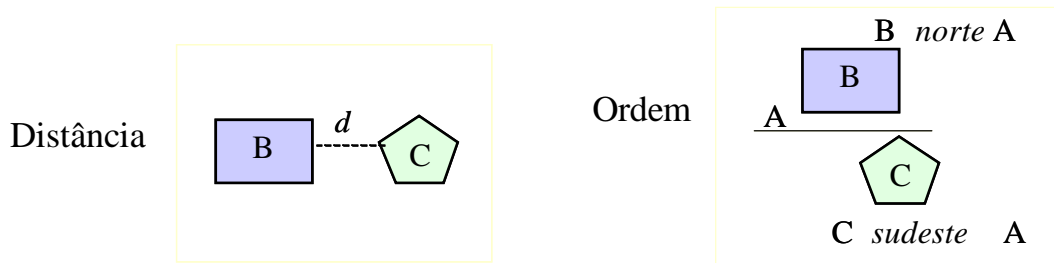
Os relacionamentos espaciais são classificados de várias formas: topológicos, orientados (ou de ordem), métricos e *fuzzy*.

Os relacionamentos orientados ou de ordem verificam se existe alguma relação de orientação entre dois objetos geométricos. Eles descrevem a ordem como os objetos estão posicionados uns em relação aos outros (CÂMARA *et al.*, 1996), partindo, normalmente, de um marco de referência que determina a direção na qual o objeto está localizado. Exemplos desse tipo de relacionamento são: acima, abaixo, ao norte, ao sul, a leste, a

oeste, à esquerda e à direita. Na figura 2.2 (direita) há um exemplo em que o objeto B está localizado a norte de A, e o objeto C a sudeste de A.

Métricos são os relacionamentos de distância que descrevem quão afastado um objeto está em relação ao outro ou a um marco de referência. Eles dependem de definições métricas no sentido de parametrizar o quanto é perto ou longe, o que dependerá das circunstâncias e das entidades geográficas relacionadas. Distância e comprimento são exemplos de relacionamentos métricos. A figura 2.2 (esquerda) esboça um exemplo de relacionamento de distância.

Figura 2.2 – Exemplo de relacionamentos de distância e ordem



Os relacionamentos *fuzzy* tratam de entidades geográficas que não possuem limites bem definidos. Um lago, por exemplo, pode ter sua área bastante modificada entre uma estação de seca e uma de chuvas. Muitos riachos chegam a desaparecer durante períodos de seca. Relacionamentos *fuzzy* em geral não são implementados por BDG.

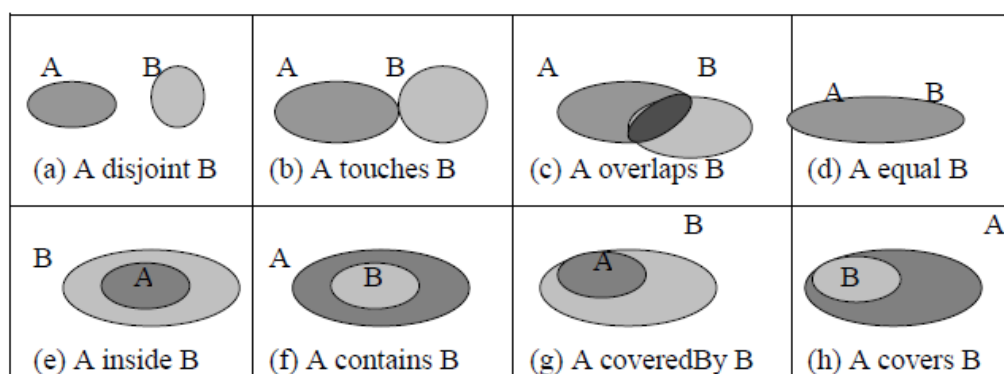
Relacionamentos topológicos referem-se à posição relativa dos objetos no espaço onde estão contidos. Eles descrevem se dois objetos interceptam-se ou não e qual a forma de interseção existente. Esse tipo de relacionamento espacial é preservado sob transformações como rotação, escala e translação. Na literatura encontraram-se vários métodos que buscam definir um conjunto significativo de relacionamentos topológicos, os quais resultam nos seguintes tipos de relação entre dois objetos espaciais (EGENHOFER, 1994): *disjoint*, *touches*, *overlaps*, *equal*, *inside*, *contains*, *covers* e *coveredBy*. Tais relacionamentos expressam ligações entre objetos espaciais do tipo ponto, linha e polígono. A figura 2.3 mostra um exemplo de relacionamento topológico de cruzamento entre os objetos geográficos do tipo rio e floresta.

Figura 2.3 – Relacionamento espacial do tipo rio cruza floresta



A figura 2.4 expõe, geograficamente, como os relacionamentos topológicos entre objetos geográficos do mundo real, representados espacialmente por polígonos, são traduzidos para um sistema computacional. No exemplo, dois objetos se tocam quando há intersecção apenas entre seus limites; sobrepõem-se quando há intersecção entre seu limite e interior; estão contidos ou contêm quando o interior e o limite de um dos objetos interceptam o interior do outro, sem interceptar seu limite; e um objeto cobre o outro quando há intersecção de interior dos dois objetos, mas o interior de um dos objetos não intercepta o limite do outro.

Figura 2.4 – Relacionamentos topológicos entre dois objetos do tipo área num espaço bidimensional



Em geral, os relacionamentos espaciais não estão armazenados na base de dados, pois são calculados por funções específicas, que manipulam dados geográficos. Como são muitos relacionamentos que precisam ser computados, essas operações são as mais custosas de um sistema gerenciador de bancos de dados geográficos (SGBDG). O uso de índices espaciais mostra-se fundamental para acelerar o processo de cálculo dos relacionamentos.

2.1.2 Bancos de dados geográficos

Um SGBDG consiste em um SGBD cujo modelo de dados e cuja linguagem de consulta foram estendidos para permitir a representação e a realização de consultas sobre objetos geográficos.

A implementação de um SGBDG consiste na extensão de um SGBD tradicional com estruturas de dados espaciais para permitir o armazenamento de geometrias de objetos, um conjunto de operações espaciais para executar computações geométricas (e.g., cálculo de área, de intersecção e de distância), técnicas de indexação espacial para otimizar o processamento de consultas baseadas no predicado espacial (e.g., contém, toca e intersecta) e extensões no otimizador de consultas para que as funções de custo e regras de tradução considerem geometrias e aproximações geométricas.

No passado, as aplicações envolvendo dados geográficos faziam uso limitado da tecnologia de BD porque apenas os dados convencionais eram mantidos nos SGBDs, enquanto as geometrias ficavam armazenadas em estruturas de arquivos mantidas separadamente. Contudo a tecnologia de BDG evoluiu, e hoje em dia a maioria dos fabricantes de SGBD (e.g., Oracle, IBM, DB2) oferece suporte ao gerenciamento de dados

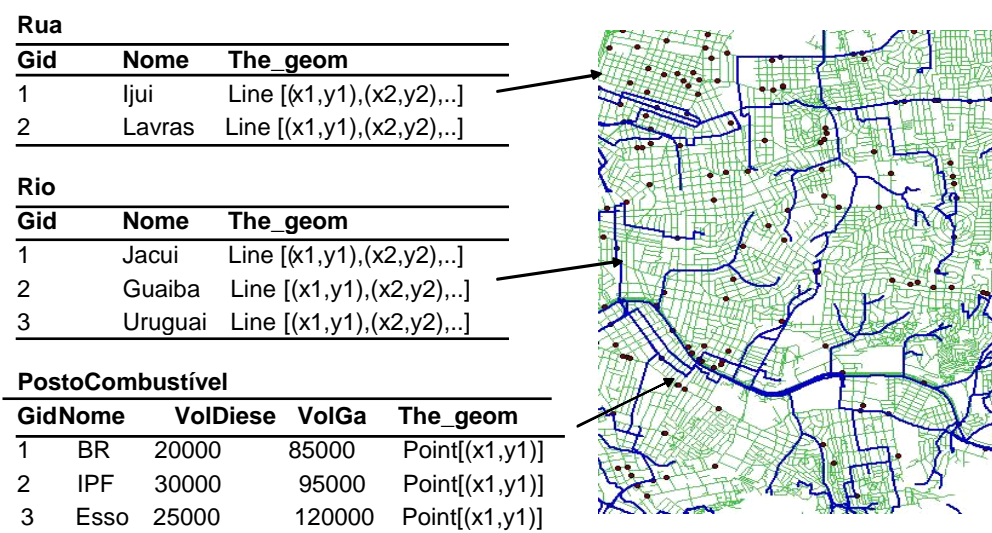
geográficos por meio de suas extensões. Entre os SGBDGs livres, o mais completo é a extensão do PostgreSQL, o PostGIS (PostGIS, 2012).

Na figura 2.5 há três tabelas com atributos geográficos armazenadas em um BDG. O atributo *gid* identifica o dado geográfico, e o atributo *the_geom* representa a geometria do objeto geográfico. É usual denominar o atributo que representa a geometria do objeto geográfico como *the_geom*, segundo o padrão OpenGIS (OGC, 2008).

A criação das tabelas em um BDG é similar à de BD convencionais, com a principal diferença que o atributo de localização exige um tipo de dado diferente (um tipo geométrico). Para criar as tabelas constantes da figura 2.5, pode-se usar o seguinte esquema:

```
Create table rua (gid integer, nome varchar(50), the_geom line);
Create table rio (gid integer, nome varchar(30), the_geom line);
Create table postoCombustivel(gid integer, nome varchar(30),
                             voldiesel integer, volgas integer, the_geom point);
```

Figura 2.5 – Exemplo de dados armazenados em um BDG



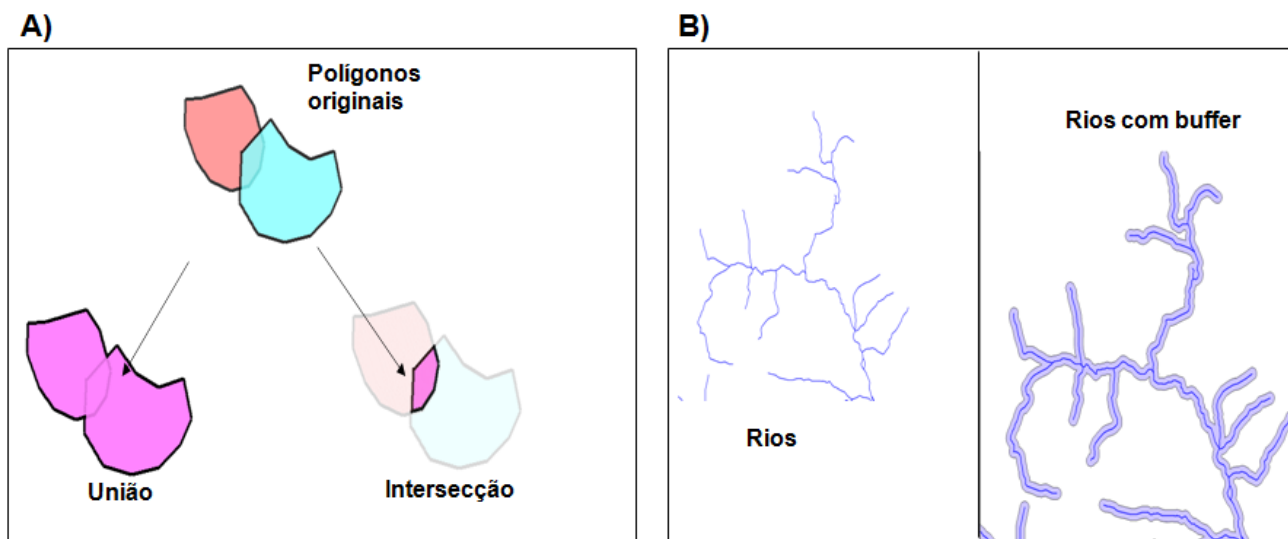
A imagem da direita da figura 2.5 possibilita visualizar os dados geográficos representados nas tabelas. Note que esses objetos têm relacionamentos espaciais visíveis, em que os postos de combustível (pontos) interceptam as ruas (linhas mais finas). Entretanto tal relacionamento não está armazenado no BDG e precisa ser calculado com operações espaciais. A próxima seção apresenta alguns exemplos de operações espaciais.

Uma distinção importante diz respeito a bancos de dados de imagens e bancos de dados espaciais. Embora o espaço geográfico possa ser representado por imagens geradas por satélites ou pela aquisição de fotografias aéreas, o foco principal de um SGBDG é a representação de objetos identificáveis que possuem uma localização. Porém técnicas de processamento de imagens podem ser usadas para identificar objetos espaciais de uma dada imagem para que, então, estes sejam armazenados em um BDG para processamento adicional.

2.1.3 Operações espaciais

Além dos relacionamentos espaciais, um BDG oferece uma série de operações para manipular dados geográficos, como, por exemplo: intersecção, união, *buffer*, *convex hull*, envelope, distância etc. A figura 2.6 (A) ilustra um exemplo das operações de união espacial e intersecção espacial. A operação de união retorna a partir de dois polígonos, um polígono unindo os objetos nas regiões que se interceptam. A operação de intersecção retorna a região ou polígono onde os dois objetos se sobrepõem.

Figura 2.6 – (A) Exemplo das operações de união e intersecção e (B) exemplo da operação *buffer*



Uma operação bastante utilizada em BDG é a *buffer* (figura 2.6 – B). A operação *buffer* expande o objeto geográfico de acordo com uma distância passada como parâmetro para a função. Por exemplo, para analisar as encostas ou margens de um rio, é possível fazer um *buffer* na geometria do rio para expandi-lo, por exemplo, 300 metros de cada lado, fazendo um *buffer* com raio 300. A próxima seção traz um exemplo de uso de um *buffer* em uma consulta espacial. A operação *buffer* pode ser aplicada sobre objetos representados por pontos, linhas ou polígonos e serve para expandir a geometria de objetos do tipo ponto, como, por exemplo, hospitais, escolas, postos de combustível etc.

A operação envelope é utilizada para agilizar uma consulta espacial e é também conhecida como menor retângulo que envolve um objeto (*minimum boundary rectangle* – MBR). Essa operação gera quatro pares de coordenadas no entorno de um dado objeto, gerando o menor retângulo que o envolve

A operação *convex hull* gera a envoltória convexa em um objeto que não tem sua geometria fechada. A operação distância retorna a distância euclidiana entre dois objetos.

2.1.4 Consultas espaciais

Aqui serão dados alguns exemplos de consultas espaciais em bancos de dados geográficos. Consultas espaciais envolvem quaisquer operadores espaciais implementados em um BDG e disponíveis na linguagem de consulta desse BD. Grande parte dos SGBDGs estende a linguagem SQL com tais operadores.

Um subconjunto padrão de operadores espaciais disponíveis no SQL espacial para consultas foi definido pelo Consórcio OpenGIS (OPENGIS 2008), um órgão que busca definir padrões para interoperabilidade em SGBDG. Com base nesses padrões, os SGBDGs comerciais e gratuitos implementam suas operações. Alguns exemplos de operações incluem: *intersects*, *crosses*, *equal*, *covers*, *covered By*, *touches*, *overlaps*, *contains*, *inside*, *buffer*, *envelope*, *convex Hull*, *closest Point*, *distance*, *azimuth* (retorna o ângulo de variação da direção), *length*, *area* etc.

Utilizando os exemplos das tabelas da figura 2.5, seguem alguns exemplos de consultas espaciais:

C1: Recupere os postos de combustível que estão localizados (interceptam) na rua Ijuí

```
Select p.*
From postocombustivel p, rua r
Where intersects (p.the_geom, r.the_geom) and r.nome = "Ijuí"
```

C2: Recupere o nome das ruas que cruzam o rio Jacuí

```
Select r.nome
From rua r, rio o
Where crosses (r.the_geom,o.the_geom) and o.nome = "Jacui"
```

C3: Recupere o nome dos postos de combustível que estão a uma distância inferior a 500 metros da margem do rio Guaíba

```
Select p.nome
From postocombustivel p, rio r
Where intersects (p.the_geom, buffer (r.the_geom,500)) and
r.nome = "Guaiba"
```

Observe nos exemplos das consultas espaciais que a junção das tabelas é espacial, pois a junção se faz por intermédio de uma operação espacial, e não por atributos descritivos como acontece em bancos de dados convencionais.

Para acelerar as consultas, aconselha-se o uso de índices espaciais. Um índice espacial é um tipo de índice estendido que permite indexar um atributo espacial. Existem vários tipos de índices diferentes, como Quad-Tree, Z-ordering-Tree etc.

2.2 Bancos de dados espaçotemporais

Um banco de dados espaçotemporal trata ao mesmo tempo as informações de tempo e espaço, em que o espaço está relacionado ao tempo. Um banco de dados espaçotemporal criado para trajetórias permite responder a perguntas (mediante consultas) sobre o presente, o passado e o futuro, bem como disparar *triggers*, de acordo com os exemplos a seguir:

- Consulta sobre o passado: durante o último ano quantas vezes o ônibus da linha 235 passou atrasado na estação Titri?

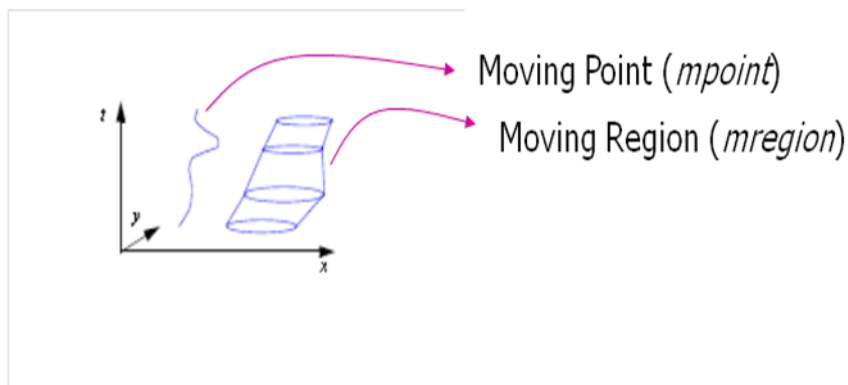
- *Trigger*: envie uma mensagem toda vez que uma aeronave entrar numa certa área geográfica
- Consulta sobre o futuro: quantos caminhões da empresa X chegarão ao seu destino nos próximos 30 minutos?
- Consulta sobre o presente: onde estão localizados os *shopping centers* a menos de 1 km de onde eu estou agora?

Até 1998 os sistemas de bancos de dados para objetos móveis eram desenvolvidos de maneira *ad hoc*, e não existiam SGBDs que permitissem a representação e a manipulação desses objetos. Wolfson *et al.* (1999; 2002) reportam o desenvolvimento do sistema DOMINO (*databases for moving objects*), que propunha que SGBDs para dados de objetos móveis fossem construídos como uma camada acima dos SGBDs tradicionais com suporte para dados geográficos, combinando as funcionalidades destes SGBDs com todas as especificidades do gerenciamento de dados espaçotemporais. Mais tarde, um grupo de pesquisa da Grécia (PELEKIS *et al.*, 2011), da Universidade de Piraeus (<http://infolab.cs.unipi.gr/hermes>), desenvolveu um protótipo de um BD de objetos móveis denominado HERMES, criado sobre a extensão espacial do Oracle. Esse banco de dados suporta apenas um tipo de dado de objetos móveis, aqueles representados por pontos móveis.

O primeiro protótipo de bancos de dados de objetos móveis que não foi construído sobre um banco de dados existente é o SECONDO, feito na Universidade de Hagen, na Alemanha (GÜTING; SCHNEIDER, 2005).

O SECONDO (GÜTING *et al.*, 2000) foi proposto para representar dois tipos de dados de objetos móveis: pontos móveis (*moving points*) e regiões móveis (*moving regions*). Na figura 2.7 há um exemplo dos dois tipos de dados. Exemplos de objetos móveis representados como *moving points* incluem trajetórias de pessoas (pedestres ou carros), trajetórias de animais, personagens em um jogo de computador etc. Regiões móveis são usadas para representar objetos móveis, como tornado, tempestade etc. E são representados a cada unidade de tempo como uma região (polígono).

Figura 2.7 – Exemplo de tipos de dados no SECONDO: pontos móveis e regiões móveis



Os dois tipos de dados permitem representar uma trajetória inteira em uma única tupla no banco de dados, enquanto um banco de dados geográfico faria uso de n tuplas para armazenar uma única trajetória, sendo necessários dois atributos (um para as

coordenadas geográficas e outro para o tempo), uma tupla para cada ponto e o respectivo tempo de coleta daquele ponto (ver exemplo da figura 5.2).

A definição de uma tabela em um banco de dados de objetos móveis como o SECONDO seria dada da seguinte maneira:

- ```
Create table voo (
 tid integer,
 origem: string,
 destino: string,
 rota: mpoint)
```
- ```
Create table clima (  
    tid integer,  
    tipo string,  
    area: mregion)
```

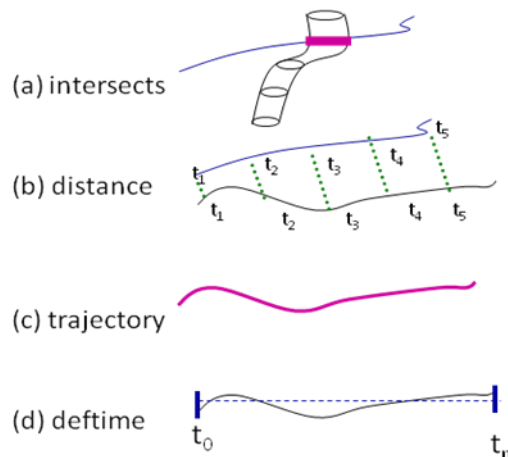
Nos exemplos são criadas as tabelas de voo, cujo atributo rota é do tipo *moving point*, caracterizando que a rota de um avião é considerada uma trajetória de pontos móveis. Já o segundo exemplo representa a criação de uma tabela de clima em que se representa a trajetória (do vento, da tempestade, do furacão) por meio de uma região móvel, definida no atributo área.

2.2.1 Operações espaçotemporais

A maior parte das operações espaçotemporais disponíveis no SECONDO analisa uma ou mais trajetórias. O SECONDO implementa uma série de operações espaçotemporais a serem aplicadas sobre as trajetórias. Observe que, enquanto as operações ou relações espaciais em bancos de dados geográficos empregam apenas a parte espacial dos objetos, verificando se eles se relacionam no espaço, as operações espaçotemporais devem levar em consideração o espaço e o tempo. Por exemplo: se dois objetos se interceptam no espaço, em qual instante de tempo a intersecção é válida? Se uma trajetória atravessa (cruza) um bairro, qual a duração do cruzamento? Em que instante de tempo a trajetória iniciou o cruzamento do bairro e em que instante de tempo ela saiu? Em um banco de dados de objetos móveis as operações precisam considerar as dimensões de tempo e espaço.

A figura 2.8 traz quatro exemplos de operações espaçotemporais que podem ser executadas sobre as trajetórias no banco de dados SECONDO.

Figura 2.8 – Exemplos de operações espaçotemporais



A operação *intersects* (figura 2.8 – a) retorna a parte da linha representada como *moving point* que intercepta espacialmente uma região móvel. Um exemplo de consulta com esse tipo de operação poderia ser: retorne a parte da trajetória de um ônibus (*moving point*) que intercepta uma tempestade (*moving region*).

A operação de distância, que em bancos de dados geográficos retorna a distância euclidiana entre dois objetos geográficos, em um banco de dados de objetos móveis retorna a distância espacial a cada instante de tempo correspondente às duas trajetórias. Conforme figura 2.8 (b), a distância espacial é calculada para o tempo t_1 das duas trajetórias, para o tempo t_2 das duas trajetórias, e assim por diante. A operação permite responder a perguntas do tipo: qual foi a menor distância entre duas aeronaves durante as trajetórias.

A operação *trajectory* (figura 2.8 – c) retorna a linha (a parte espacial) correspondente à trajetória passada como parâmetro para a função. Já a operação *deptime* (figura 2.8 – d) retorna o período de duração da trajetória (exemplo: uma hora, 24 horas etc).

Enquanto o SECONDO oferece dois tipos de dados para representar objetos móveis (*moving point* e *moving region*), o HERMES apresenta apenas o tipo *moving point*. Por um lado, utilizando os recursos do SGBD espacial do Oracle, o HERMES oferece muito mais operações para análise de trajetórias. As operações oferecidas pelo HERMES já permitem fazer diversas análises de similaridade entre trajetórias e também entre trajetórias e objetos geográficos. Por outro lado, o HERMES incorpora a complexidade do SGBD proprietário Oracle.

As operações de similaridade implementadas pelo HERMES são baseadas em distância e vizinho mais próximo. As operações de distância podem ser:

- Em relação a um objeto espacial de referência, como, por exemplo: Encontre as lojas a menos de 100 metros de onde eu estou e que tenham oferta de roupas esportivas;
- Em relação a um objeto móvel: Encontre as pessoas que passaram a menos de 100 metros de onde estou e que já compraram uma roupa esportiva.

A seguir, exemplos de operações de vizinho mais próximo:

- Em relação a um objeto espacial de referência, como, por exemplo: Encontre a loja mais próxima de onde estou agora e que tenha oferta de produtos esportivos;
- Em relação a um objeto móvel: Encontre as duas trajetórias mais próximas da minha rota atual.

As operações de distância e *buffer* e relacionamentos topológicos em bancos de dados de objetos móveis (nesse caso no HERMES) funcionam de forma diferente de bancos de dados espaciais, visto serem relacionadas a tempo e espaço, e não somente a espaço. A seguir são listadas algumas operações espaçotemporais disponíveis no HERMES e os tipos de consultas que podem ser realizados com tais operações:

- **f_buffer**: gera um polígono ao redor de um ponto em um tempo específico;
- **f_distance**: calcula a distância entre dois pontos móveis em um tempo específico;
- **f_within_distance**: determina se um ponto móvel está dentro de uma dada distância de outros objetos em um tempo específico;
- **f_union**: retorna a geometria do objeto que representa a união topológica entre dois objetos móveis em um determinado tempo;
- **get_enter_leave_points**: retorna o ponto em que um objeto móvel entrou em uma geometria e o ponto em que saiu da geometria (exemplo: retorna o ponto da trajetória em que uma pessoa entrou em um *shopping* e o ponto em que ela saiu do *shopping*).

2.2.2 Consultas espaçotemporais

Em seguida, damos um exemplo de consulta espaçotemporal utilizando a operação de distância entre trajetórias disponível no SECONDO e considerando a tabela de voos criada anteriormente:

C1: Recupere todos os pares de aviões que durante sua trajetória (seu voo) estiveram a menos de 500 metros de distância um do outro.

```
SELECT f.tid, g.tid
FROM voo f, voo g
WHERE f.tid <> g.tid AND min(distance(f.rota, g.rota)) < 0.5
```

C2: Encontre todos os voos que saem de Florianópolis e que percorrem mais de 500 quilômetros.

```
SELECT *
FROM voo
WHERE origem = 'Florianopolis' AND lenght(trajectory(rota)) > 500
```

A consulta C2 recupera todas as tuplas da tabela de voos cuja origem é Florianópolis, em seguida recupera a linha (com a operação *trajectory*) que representa a

trajetória (atributo rota) e por último recupera o comprimento da linha da trajetória, a qual deve ser maior do que 500 para entrar no resultado da consulta.

A consulta C3 trata de consulta espaçotemporal executada no banco de dados HERMES, que, em virtude da sua implementação em Oracle, requer consultas em PL/SQL, de forma mais complexa do que no SECONDO.

C3: Qual a distância percorrida pelo veículo X dentro de uma região?

```
DECLARE
distance double;
BEGIN
distance:= LENGTH (INTERSECTION (region, truckX.f_trajectory()));
END;
```

A consulta busca a intersecção da região com a trajetória. A consulta retorna a parte da trajetória (subtrajetória) que intercepta a região (com a operação *intersection*). A operação *length* retorna o comprimento dessa trajetória, que equivale à distância percorrida, e é uma operação espacial do Oracle.

O trabalho de Pelekis *et al.* (2011) traz outros exemplos de consulta no HERMES.

2.3 Data warehouses espaciais e espaçotemporais

Os *data warehouses* surgiram da necessidade de fazer análises sob várias dimensões de grandes volumes de dados armazenados em bancos de dados. De acordo com Inmon (1996), um *data warehouse* é uma “coleção de dados orientada por assunto, integrada, não-volátil, variante no tempo, que oferece apoio às decisões da administração”. Um *data warehouse* tem a capacidade de permitir acesso a dados coletados das mais diversas fontes, considerando um tema ou assunto específico para análise, e por conta da característica da não volatilidade possibilita manter o histórico dos dados relacionados com o assunto armazenado.

Os *data warehouses* utilizam ferramentas OLAP (*Online Analytical Processing*), que conseguem manipular e analisar um grande volume de dados sob múltiplas perspectivas (Codd; Codd; Salley, 1993). A tecnologia OLAP prevê a construção de resumos de dados considerando o armazenamento multidimensional dos dados. Realiza-se o armazenamento multidimensional por meio do uso de hierarquias, a serem explicadas mais adiante. Basicamente, o modelo multidimensional envolve dois grandes conceitos: fatos e dimensões. O primeiro representa o assunto a ser analisado e é representado no modelo multidimensional pela tabela de fatos. Já a dimensão possui diversas possibilidades de visualização sobre o fato, sendo representada pelas tabelas de dimensão. O exemplo a seguir pode auxiliar no entendimento dos conceitos relacionados com a modelagem multidimensional.

Exemplo: A área de *marketing* de uma rede de supermercados recebeu a tarefa de incrementar as vendas de um determinado produto em todas as suas filiais. Para isso, a primeira informação necessária é um resumo das vendas de cada produto por região.

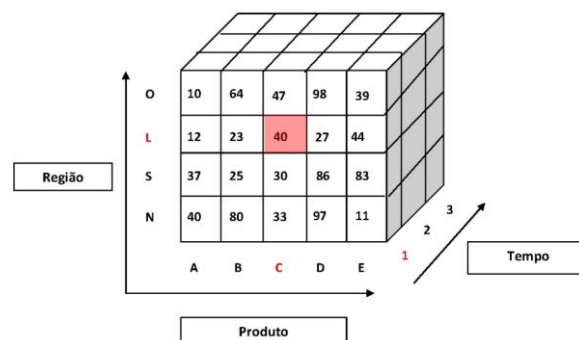
Para atingir a meta não se mostra necessário saber individualmente quais foram os compradores de cada produto, mas sim fazer os resumos dos dados relacionados com as vendas. O interesse está no resumo, e não no detalhe do registro de vendas.

No exemplo, o fato em análise é a venda. Em qualquer processo de análise de um fato é preciso utilizar medidas. As medidas apresentam um conhecimento por intermédio da indicação de um valor. No caso do fato vendas, o mesmo poderia ser mensurado por meio das seguintes medidas: quantidade de vendas, valores de vendas, quantidade de itens vendidos etc. Entretanto, para que a informação seja completa e que faça algum sentido, é imprescindível considerar algum eixo de visualização dessas medidas. Os eixos de visualização do fato vendas poderiam ser cliente, loja, produto e tempo, por exemplo. Os eixos representam as dimensões do modelo multidimensional.

Considerando uma estrutura multidimensional, consegue-se identificar a quantidade de vendas (medida do fato) por cliente, por loja, por produto e também pelo tempo de ocorrência do fato. As combinações entre fato e dimensões permitem diversas análises de dados.

O modelo multidimensional é representado como um cubo de dados. A figura 2.9 ilustra um exemplo de cubo com as vendas nas dimensões produto, região e tempo.

Figura 2.9 – Cubo de dados de vendas

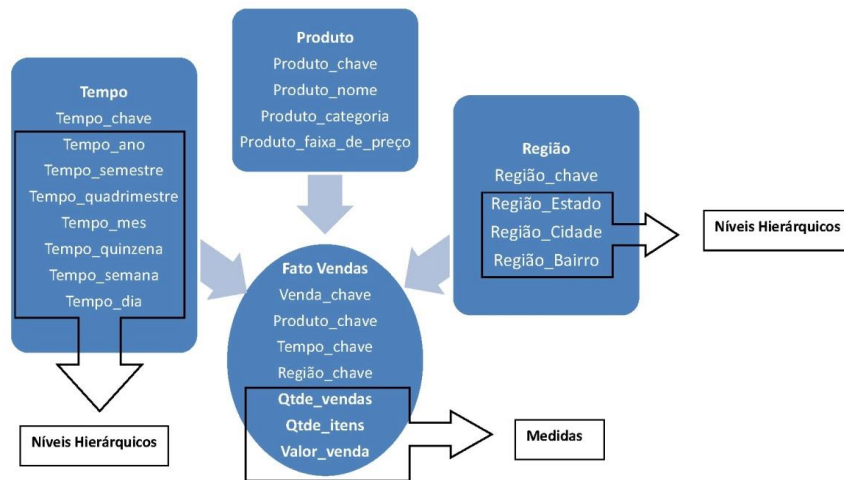


Cada célula do cubo na figura 2.9 representa a medida relacionada com as “coordenadas” das dimensões. A dimensão região é composta pelos registros N (norte), S (sul), L (leste) e O (oeste). Já a dimensão produto possui os registros dos produtos A, B, C, D e E. Finalmente, a dimensão tempo está estruturada na forma de quadrimestres (primeiro, segundo e terceiro). A célula grifada em vermelho representa o seguinte fato: 40 unidades do produto C foram comercializadas na região L (leste) no primeiro (1) quadrimestre do ano. Por uma questão de simplicidade de representação, a figura considera somente a medida de quantidade de itens vendidos.

Cada uma das dimensões do DW pode fazer uso do conceito das hierarquias. A estrutura hierárquica permite que os resumos de dados aconteçam também ao longo das dimensões. No exemplo em discussão a dimensão região poderia ser hierarquizada em **macrorregião** → **estado** → **cidade** → **bairro**.

Os *data warehouses* possuem dois modelos clássicos (INMON, 1996): estrela (*star*) e floco de neve (*snowflake*). O modelo estrela envolve, basicamente, uma tabela de fatos conectada a um conjunto de tabelas de dimensões. A figura 2.10 apresenta o cubo de dados utilizado de acordo com a arquitetura estrela.

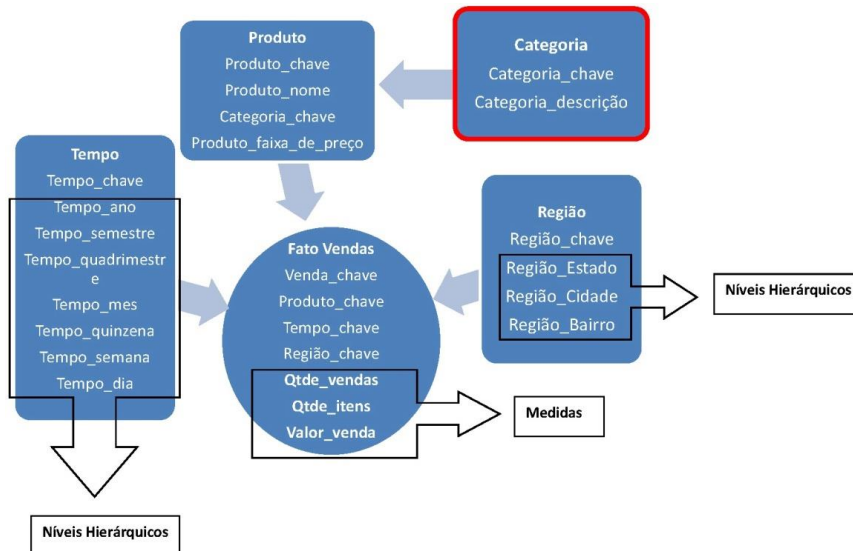
Figura 2.10 – Modelo estrela – fato vendas



Na figura 2.10 identifica-se um único nível de tabelas de dimensões (região, produto e tempo), todas conectadas com a tabela de fatos fato vendas. Trata-se da principal característica da arquitetura estrela. Na tabela de fatos percebem-se as três medidas comentadas anteriormente: quantidade de vendas, quantidade de itens e valor da venda.

O modelo floco de neve (figura 2.11) é uma especialização do estrela. Nele uma dimensão é normalizada em níveis ou hierarquias adicionais. Na figura, note que aconteceu um nível de normalização com base na dimensão produto. No modelo, a dimensão produto possui relacionamento com um nível adicional, o nível categoria. Esse relacionamento ocorre por meio do atributo **categoria_chave**.

Figura 2.11 – Modelo floco de neve – fato vendas



Independentemente de qual modelo seja empregado, existem algumas operações da tecnologia OLAP que permitem o processamento e a recuperação dos resumos de dados e fazem uso das medidas e dimensões relacionadas ao fato em análise. Algumas dessas operações são, segundo Inmon (1996): *roll up (drill up)*, *drill down* e *slice and dice*.

Elas utilizam o conceito de granularidade ou hierarquia. A granularidade de um cubo de dados tem relacionamento direto com o grau de detalhe que se espera das operações executadas em um cubo de dados. O nível de detalhe resulta da combinação entre as dimensões que compõem o cubo de dados. A granularidade pode ser comparada a uma hierarquia, em que o elemento mais superior é o mais genérico (exemplo, país) e o mais inferior é o mais detalhado (exemplo, bairro).

Na figura 2.11 observa-se que é possível fazer agregações nas dimensões tempo e região. O tempo pode ser agregado desde o dia da venda até ano e, caso da dimensão região, de bairro até estado. Não existe a possibilidade de, naquele exemplo, identificar resumos de dados por hora do dia e rua da região, por exemplo. Dessa forma, a definição da granularidade torna-se uma das decisões fundamentais na modelagem de qualquer ambiente multidimensional.

A operação de *roll up* oferece a possibilidade de agregação de dados ao longo dos níveis hierárquicos, partindo do menor nível de detalhe em direção ao de maior nível. A consulta a seguir ilustra um modelo.

1. Quantidade de vendas, por mês de venda e bairro da venda:

```
Select Tempo_Mes, Regiao_bairro, Sum(Qtde_vendas)
From Fato_vendas, Tempo, Região
Where Fato_vendas.Tempo_chave = Tempo.Tempo_chave AND
        Fato_vendas.Regiao_chave = Regiao.Regiao_chave
Group By Tempo_Mes, Regiao_bairro
```

2. Quantidade de vendas, por mês de venda e cidade da venda:

```
Select Tempo_Mes, Regiao_Cidade, Sum(Qtde_vendas)
From Fato_vendas, Tempo, Região
Where Fato_vendas.Tempo_chave = Tempo.Tempo_chave AND
        Fato_vendas.Regiao_chave = Regiao.Regiao
Group By Tempo_Mes, Regiao_Cidade
```

Veja que o nível de detalhamento dos resumos de dados migrou de **bairro** (na primeira consulta) para **cidade** (na segunda consulta).

A operação de *drill down* pode ser entendida como o inverso da operação de *roll up*, considerando o sentido da agregação dos dados. Enquanto na operação de *roll up* a agregação de dados acontece no sentido do maior nível de detalhe para o menor, na operação de *drill down* o processo de agregação ocorre no sentido do menor nível de detalhe para o maior.

Finalmente, a operação de *slice and dice* pode ser entendida como de “fatiamento” do cubo de dados. Aqui há o interesse por uma determinada porção do cubo de dados, independentemente do nível de agregação existente. O comando da linguagem SQL representa um exemplo de uma operação *slice and dice*, expondo os resumos de dados ao longo da dimensão região, considerando um bairro específico, de modo a identificar a “fatia” do cubo a ser visualizada.

```
Select Regiao_Bairro, Sum(Qtde_vendas)
From Fato_vendas, Região
Where Fato_vendas.Regiao_chave = Regiao.Regiao_chave AND
        Região.Regiao_bairro = 'Saguaçu'
Group By Regiao_bairro
```

2.3.1 *Data warehouse* espacial

Um *data warehouse* espacial combina as tradicionais funcionalidades de um *data warehouse* com as funcionalidades de sistemas de informações geográficas (SIG) (HAN; STEFANOVIC; KOPERSKI, 1998). A combinação se dá por meio da utilização de atributos espaciais. Os atributos espaciais podem estar tanto na tabela de fatos quanto nas tabelas de dimensões. Consultas do tipo “a quantidade de vendas de um determinado produto X em uma região geográfica cruzada por uma rodovia Y” ou “o volume de vendas em uma região que envolve dois bairros dentro de uma cidade” podem ser respondidas com um *data warehouse* espacial.

Entretanto a extensão das funcionalidades de um ambiente de *data warehouse* para contemplar medidas e dimensões espaciais não é tarefa tão simples. Uma série de propostas de modelagem e de desenvolvimento de operadores OLAP ainda está em estudo. Este livro não tem o foco de aprofundar o detalhamento dessas propostas. O objetivo é apresentar os principais conceitos de *data warehouses* espaciais para entender um pouco sobre DW de trajetórias.

Medidas espaciais

A tabela de fatos de um *data warehouse* espacial admite o armazenamento de medidas numéricas e também de medidas espaciais. Uma medida numérica poderia representar o volume de vendas, enquanto a medida espacial, a região onde aconteceu aquele volume de vendas. Uma dificuldade na utilização de medidas espaciais está no processo de agregação de dados. Em sistemas de *data warehouse* tradicionais, as funções de agregação são bem conhecidas e pesquisadas. Contudo, em *data warehouses* espaciais, as funções tradicionais não são suficientes. Geralmente, no caso de agregação de geometrias, a junção espacial ou união espacial tem sido usada para fornecer a resposta ao processo de agregação de dados espaciais.

A junção espacial consiste no agrupamento de geometrias que satisfazem uma determinada condição. Por exemplo, “recuperar o volume de vendas para todas as cidades cortadas pela BR-101”. A condição é a intersecção pela rodovia. Uma vez que as regiões foram identificadas e agrupadas, há o processo de junção entre todas as geometrias e, finalmente, a agregação do volume de vendas (soma) sobre aquela nova geometria incluindo as regiões.

Dimensões espaciais

Uma dimensão espacial emprega pelo menos um atributo espacial na sua estrutura, e a hierarquia considerada é espacial. Dois níveis espaciais de uma dada hierarquia possuem um relacionamento topológico entre si. Por exemplo, cidade e bairro são dois níveis, e o bairro está topologicamente ligado à cidade pelo relacionamento topológico contido. Esses relacionamentos espaciais foram detalhados no capítulo 2. Vários relacionamentos topológicos distintos podem ser considerados, embora o mais discutido na literatura seja o completamente contido, por ser mais facilmente definido e implementado (e.g. a geometria de uma cidade está completamente contida na geometria do estado correspondente).

De acordo com Malinowski e Zimányi (2004), dois são os principais tipos de hierarquias espaciais de um DW: estritas e não estritas. Em uma hierarquia estrita todos os relacionamentos entre os níveis da hierarquia possuem a cardinalidade 1:N. Instâncias desses relacionamentos podem ser representadas por meio de uma árvore. Por exemplo, **Endereço** \subset **Cidade** \subset **Estado**. As instâncias formam uma árvore cujos ramos possuem o

mesmo tamanho, e todo membro pai possui pelo menos um membro filho e todo membro filho está associado a um único membro pai, para satisfazer a cardinalidade de um para muitos do relacionamento.

Hierarquias não estritas possuem pelo menos um relacionamento com cardinalidade de N:M entre alguns dos seus níveis. A maioria das hierarquias de tal tipo ocorre quando existe um relacionamento parcial de estar contido entre níveis. Por exemplo, quando apenas parte de uma estrada pertence a um estado. Na prática, é difícil encontrar uma hierarquia espacial que seja não estrita com relação ao relacionamento espacial estar completamente contido.

Várias propostas de modelos de dimensões espaciais têm sido apresentadas na literatura com o objetivo de oferecer soluções aos problemas da representação espacial em *data warehouses*, no entanto estão fora do escopo deste livro.

2.3.2 *Data warehouses* espaçotemporais (trajetórias)

Data warehouses tradicionais podem ser definidos como uma coleção de dados orientados a assuntos originários de diversas fontes de dados distintas. Entretanto, em um ambiente de objetos móveis, um *data warehouse* deve considerar informações não espaciais, espaciais e temporais, tornando-se um problema de grande complexidade. Além disso, tratando-se de trajetórias, os dados não podem ser agrupados por assunto, pois, em geral, eles constituem um conjunto de pontos localizados no tempo e no espaço. Portanto, os modelos tradicionais de *data warehouse* não suportam análises de dados espaçotemporais de trajetórias.

Ainda não existe um modelo padrão para a utilização da tecnologia de DW no armazenamento e no análise de dados de trajetórias de objetos móveis, mas há algumas propostas que trabalham com o armazenamento das informações das trajetórias com e sem a utilização de estruturas espaciais. Neste livro serão apresentadas as abordagens de Braz e Orlando (2007), Pelekis *et al.* (2008) e Almeida (2010).

No trabalho de Braz *et al.* (2007) foi apresentado um modelo para *data warehouse* de trajetórias baseado em duas considerações básicas:

- A identificação dos objetos móveis e trajetórias foi abstraída, ou seja, a proposta tem como objetivo estudar o ambiente levando em conta as propriedades globais de um conjunto de objetos (por exemplo: número de trajetórias cruzando uma determinada área espaçotemporal), e não a individualização de trajetórias;
- Os registros do *data warehouse* são representados por células espaçotemporais sobre as quais uma série de medidas é representada.

Nesse modelo o conjunto de posições assumidas por um objeto ao longo do tempo representa a sua trajetória e é caracterizado individualmente por pontos (tid, x, y, t) , em que *tid* identifica a trajetória, o par *x, y* significa a posição espacial e *t* expressa o instante de tempo. No modelo multidimensional, cada célula (*cube*) representa as características de um conjunto de trajetórias que atravessa a célula. Por exemplo, considere que a referida célula poderia representar uma determinada região de uma grande metrópole composta por diversas avenidas, cada uma delas, naquele instante de tempo, comportando um volume de tráfego de diversos veículos (os objetos móveis). No exemplo, as medidas (velocidade, aceleração, distância, quantidade de objetos e tempo de permanência) seriam o conjunto de atributos das trajetórias dentro daquela região.

Para representar as trajetórias em cada célula, o modelo utiliza o tradicional esquema estrela com três dimensões: duas dimensões espaciais X e Y e uma dimensão temporal T. A figura 2.12 permite a visualização do modelo.

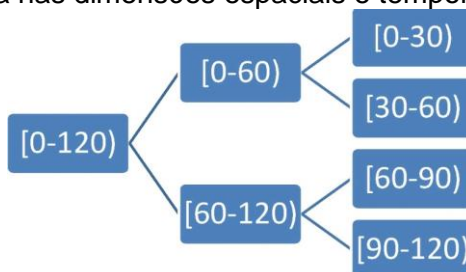
Figura 2.12 – Esquema estrela para DWT



Note que a tabela fatos possui como atributos algumas características das trajetórias: além do x,y e t, tem a velocidade da trajetória, aceleração, distância e presença, sendo esse último o número de trajetórias dentro da célula.

Em qualquer modelo de *data warehouse* o conceito de hierarquia é de fundamental importância para a análise de dados agregados. A hierarquia (ou granularidade) dentro das dimensões, como já dito anteriormente, está relacionada com a especificação de níveis de agregação dos dados. Nesse modelo de *data warehouse* para trajetórias o conceito de hierarquia é definido por meio da discretização dos valores das dimensões espaciais e temporais. A figura 2.13 apresenta um exemplo de agregação temporal.

Figura 2.13 – Hierarquia nas dimensões espaciais e temporais

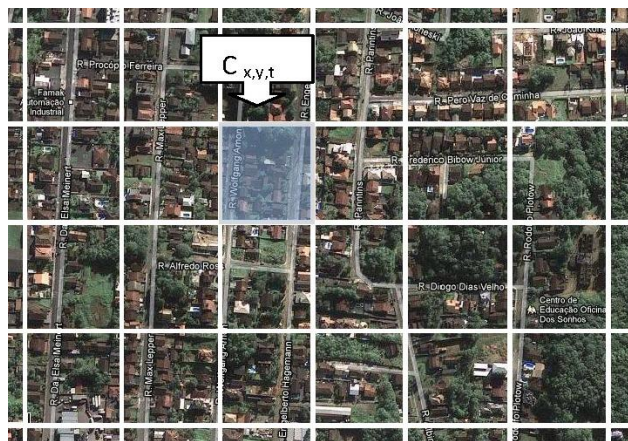


As dimensões foram discretizadas de acordo com intervalos de tempo definidos previamente no contexto da aplicação. Por exemplo, o nó [0-60) (figura 2.13) poderia representar o intervalo entre 0 e 60 minutos para a dimensão tempo, e o nó [60-120), o intervalo entre 60 e 120 minutos na mesma dimensão. O conceito repete-se ao longo dos diversos níveis hierárquicos.

Cada registro armazenado na tabela de fatos representa um resumo das medidas delimitadas pelas fronteiras da célula. As células são delimitadas pelos valores Xid , Yid e Tid . Cada um desses valores Xid , Yid e Tid representa também um intervalo definido previamente. Cada célula armazena medidas que apresentam características dos conjuntos de trajetórias envolvidas pelas fronteiras da célula. As células serão referenciadas pela notação $C_{x,y,t}$, representando uma célula definida pelas coordenadas x , y e t , ou seja, as coordenadas são o identificador da célula.

A figura 2.14 traz uma área espaçotemporal (grande região) em que cada célula, representada pela notação $C_{x,y,t}$, faz referência a uma sub-região (da grande região). Cada sub-região (célula) é representada no *data warehouse* de trajetórias por intermédio de um registro na tabela de fatos e suas respectivas dimensões.

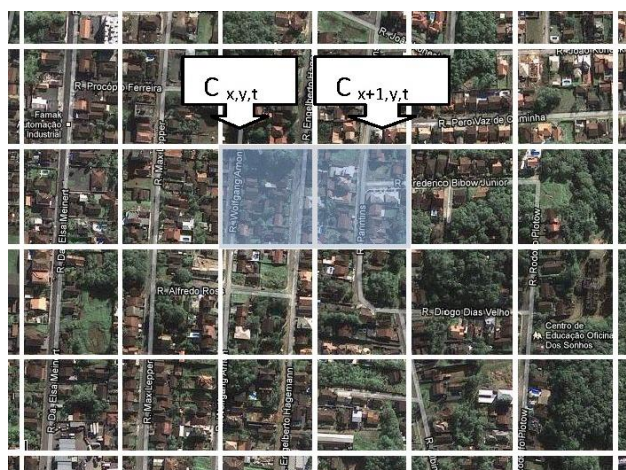
Figura 2.14 – Células do *data warehouse*



As medidas propostas são o número de observações (quantidade de registros), número de trajetórias iniciando na célula, número máximo de trajetórias na célula, distância percorrida pelos objetos, tempo gasto pelas trajetórias na célula e uma medida chamada presença. A medida presença tem como propósito permitir o cálculo da quantidade de objetos distintos que trafegam dentro de uma determinada célula.

Vejamos, por exemplo, uma consulta da medida presença ao longo da dimensão espacial X : determinar a quantidade de trajetórias considerando um incremento no nível hierárquico da dimensão X . Na figura 2.15 visualiza-se uma operação de *roll up* sobre a célula. Essa operação aumenta a área de abrangência da consulta ao longo da dimensão X . No exemplo a operação de *roll up* envolve as células $C_{x,y,t}$ e $C_{x+1,y,t}$ e, em vez de retornar a presença de trajetórias dentro de uma célula, retorna o total de trajetórias das duas células. A mesma operação pode ser realizada sobre qualquer das três dimensões – eixo x , y ou o tempo.

Figura 2.15 – Operação de *roll up* ao longo da dimensão X



Um protótipo considerando o modelo anterior foi desenvolvido e apresentado por Braz e Orlando (2007). Para o protótipo empregou-se um sistema gerenciador de banco de dados tradicional (MS SQL Server 2005) que permitiu o uso do modelo estrela. O protótipo tem basicamente dois componentes: o carregador de dados e o agregador de dados; a interface principal está mostrada na figura 2.16.

Uma vez definidos os níveis de granularidade, as operações serão executadas por meio da interface de visualização das medidas. Do lado esquerdo da interface fica a visualização do mapa da região de análise das trajetórias. O usuário pode escolher definir a célula base para a sua consulta mediante controles do tipo *combo-boxes* ou clicando diretamente sobre algum ponto do mapa. Na hipótese de clicar sobre o mapa, o aplicativo identifica as coordenadas daquele ponto e busca no *data warehouse* a célula que envolve aquele ponto específico. A partir daí todos os valores das medidas que envolvem as células relacionadas na consulta são processados e apresentados do lado direito da tela, na forma de gráficos. A operação do tipo *roll up*, quando envolve a medida presença, calcula o total de trajetórias presentes na célula selecionada. A utilização de gráficos permite acompanhar a evolução das medidas ao longo da dimensão temporal.

Dessa forma, a proposta oferece a possibilidade de responder a perguntas do tipo: Qual a velocidade média, a distância percorrida pelas trajetórias e o tempo de permanência delas dentro de uma determinada sub-região espaçotemporal?

A questão pode ser solucionada com a seguinte consulta:

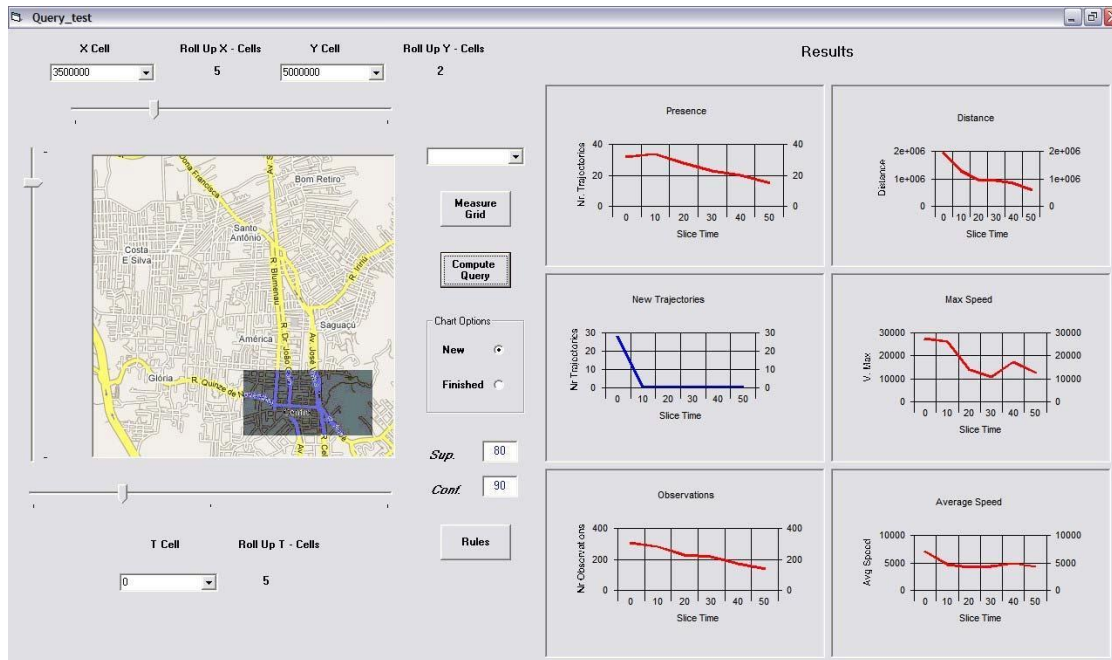
```

SELECT T.datelvl2, AVG(F.Velocidade), SUM(F.Distancia), SUM(F.Tempo)
FROM Fatos As F INNER JOIN
    Dim_Tempo As T ON (F.Tid = T.Tid) INNER JOIN
    Dim_X As X ON (F.Xid = X.Xid) INNER JOIN
    Dim_Y As Y ON (F.Yid = Y.Yid)
WHERE X.Xid BETWEEN 10 AND 20 AND
    Y.Yid BETWEEN 20 AND 30 AND
    T.Tid BETWEEN '10/01/2011' AND '15/01/2011'
GROUP BY T.datelvl2
    
```

Define-se a restrição espacial por meio da localização das células relacionadas com a consulta na tabela de fatos do modelo (operador *between* envolvendo as dimensões *Dim_X* e *Dim_Y*). A definição da restrição temporal permitirá acompanhar a evolução das

diversas medidas ao longo do tempo (operador *between* sobre a dimensão *Dim_Tempo*). Uma vez que as tuplas do *data warehouse* foram recuperadas pelas condições anteriores, os resumos das medidas velocidade, distância e tempo são agrupados pelo segundo nível hierárquico da dimensão temporal.

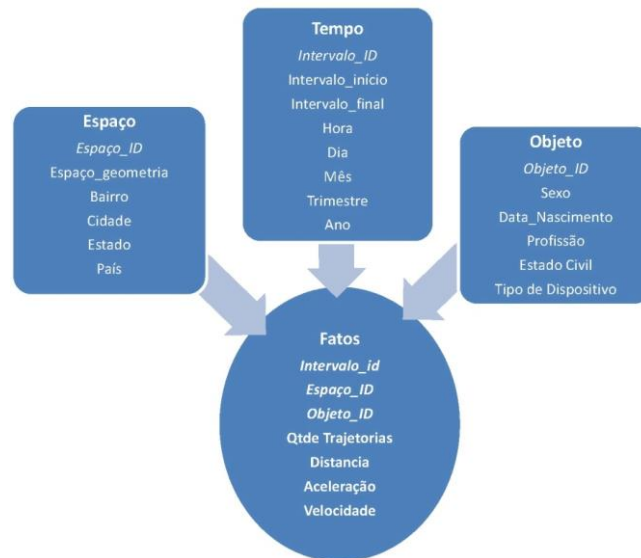
Figura 2.16 – Interface de emissão e apresentação de consulta



Braz e Orlando (2007) foram um dos pioneiros no estudo de *data warehouse* para trajetórias. Com base em tal pesquisa outros trabalhos surgiram na tentativa de atender às necessidades de um ambiente de trajetórias.

A proposta de Pelekis *et al.* (2008) é bastante similar ao trabalho descrito anteriormente, até mesmo recorre ao modelo estrela. A estrutura da tabela de fatos mantida pelo *data warehouse* é muito semelhante à proposta anterior. A principal diferença entre ambas reside nas dimensões utilizadas. A última abordagem também permite representar um conjunto de trajetórias como uma trajetória e reconstruir trajetórias com falhas. O modelo usa uma tabela de fatos e três tabelas de dimensões: espaço, tempo e perfil de objeto. Na figura 2.17 há um modelo dessa abordagem.

Figura 2.17 – Modelo estrela



Fonte: Adaptado de Pelekis *et al.* (2008)

A tabela de fatos, além das chaves para implantar o relacionamento com as dimensões, apresenta um conjunto de medidas representando informações agregadas. As medidas são as seguintes: quantidade de trajetórias distintas, distância média percorrida pelas trajetórias, duração média de permanência das trajetórias, velocidade e aceleração média das trajetórias. As medidas são aplicadas sobre um grupo de trajetórias, classificadas por um tipo de objeto (dimensão perfil do objeto) dentro de uma determinada região espaçotemporal (dimensões espaço e tempo).

A dimensão espaço possui uma hierarquia definida ao longo da célula: *bairro* → *cidade* → *estado* → *país*. A hierarquia permite o incremento na capacidade de análise dos dados, possibilitando o resumo de dados ao longo de cada um desses níveis hierárquicos. Já na dimensão temporal, os níveis hierárquicos cronológicos considerados são: *hora* → *dia* → *mês* → *trimestre* → *ano*.

Finalmente, a dimensão perfil do objeto apresenta como grande novidade a utilização de informações demográficas (sexo, data de nascimento, profissão, estado civil) em conjunto com informações do dispositivo usado na coleta do dado (tipo de dispositivo), permitindo a identificação do dispositivo que gera a trajetória. O emprego da dimensão perfil do objeto permite realizar consultas do tipo “quantos objetos de determinada faixa etária se movimentam a uma determinada velocidade média dentro de um intervalo temporal de dois dias”.

Almeida (2010) propôs um modelo semântico de *data warehouse* de trajetórias bem mais rico em termos semânticos se comparado com os anteriores. Na figura 2.18 é possível ver o esquema da proposta. Esse modelo de *data warehouse* é baseado no modelo conceitual de *stops* e *moves*, apresentado no início do livro, e por isso pode ser considerado semântico. O modelo permite responder a questões do tipo: (i) Quantos objetos fazem paradas e deslocamentos entre dois bairros; (ii) Qual a velocidade média dos objetos; (iii) Qual a distância percorrida por objetos de um determinado tipo, dentro de um dado bairro.

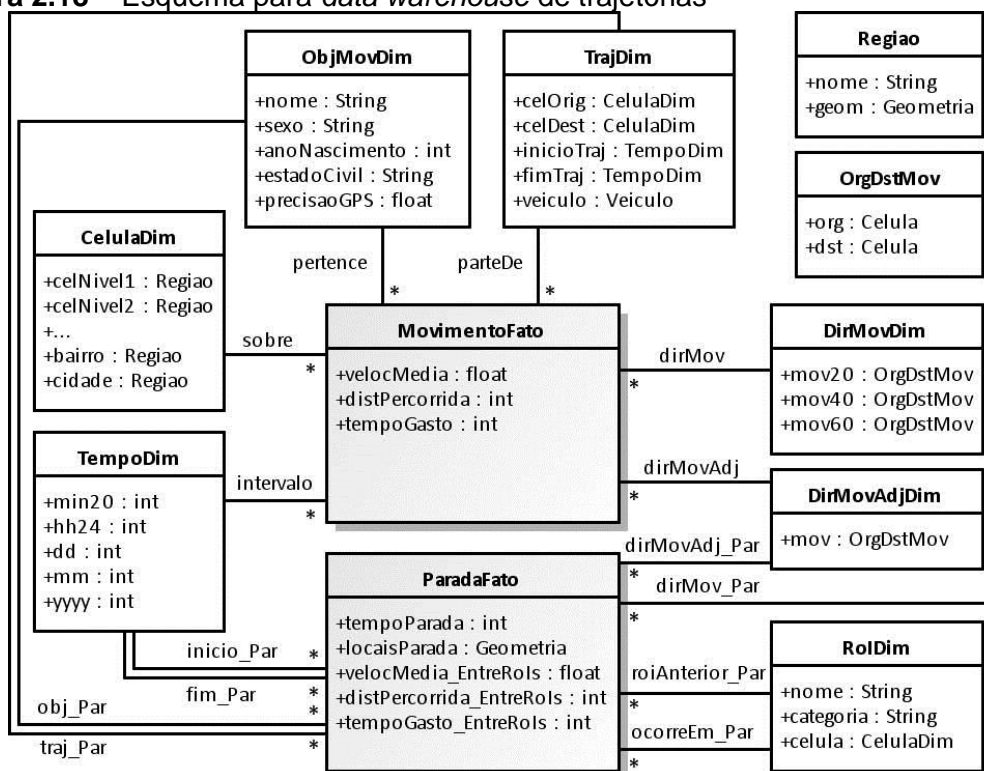
O modelo do *data warehouse* adotado na proposta está fortemente apoiado no modelo estrela. Entretanto nesse modelo existem duas tabelas de fatos, uma para representar medidas relacionadas aos *stops* (fato parada) e outra para representar medidas relacionadas aos *moves* (fato movimento). A tabela de fatos movimento armazena os dados sumarizados por célula espaçotemporal relacionados com a movimentação de

objetos dentro dos limites da respectiva célula. As medidas da tabela de fatos movimento são basicamente: velocidade, quantidade de objetos, espaço e tempo percorridos pelos objetos dentro de cada célula espaçotemporal. Assim, cada registro da tabela de fatos corresponde à representação de um conjunto de características de uma série de trajetórias envolvidas pelas fronteiras espaçotemporais da célula.

Já a tabela de fatos parada armazena os dados sumarizados também por célula espaçotemporal, porém as medidas principais são tempo de parada e local de parada. O processo de agregação dessa tabela de fatos leva em conta, para o conjunto de trajetórias, o tempo de permanência em uma região de parada. Agregações de dados considerando desde características dos objetos (dimensão *ObjMovDim*) até informações semânticas da região, armazenadas na dimensão *RoiDim*, podem ser realizadas. As regiões de interesse representam locais onde os objetos não se movem ou se movimentam em velocidade muito reduzida, permanecendo naquela região espacial por um considerável intervalo de tempo.

A hierarquia adotada para definir a região é *célula* → *bairro* → *cidade*.

Figura 2.18 – Esquema para *data warehouse* de trajetórias



Fonte: Almeida (2010)
[ver acentos na figura]

O modelo também armazena informações sobre a direção do movimento. Na realidade, utilizam-se duas dimensões para a representação desse conceito: as dimensões *DirMovDim* e *DirMovAdjDim*. A informação que representa a direção de um movimento (cidade A para cidade B, por exemplo) é armazenada na dimensão *DirMovDim*. A dimensão *DirMovrAdjDim* representa o sentido de movimentação entre regiões adjacentes.

O modelo permite a realização de consultas com alto poder de expressão dos fenômenos de movimentação de objetos dentro de uma determinada área espaçotemporal. Por exemplo:

Retorne a quantidade e a velocidade média de veículos em movimentação em uma região, em uma data, apresentado os resumos por hora e por região.

```
SELECT
cd.celulaNivel3_latlon AS latlon,
td.hh24 AS intervaloTempo,
COUNT(DISTINCT mf.objId) AS numObjetosMoveis,
AVG(mf.velocMedia) AS velocidadeMedia
FROM
MovimentoFato AS mf INNER JOIN
CelulaDim AS cd ON (mf.celulaId = cd.celulaId) INNER JOIN
TempoDim AS td ON (mf.tempoId = td.tempoId)
WHERE
td.tempo BETWEEN
to_timestamp('10-jan-2009 00:00', 'DD-mon-YYYY HH24:MI') AND
to_timestamp('10-jan-2009 23:40', 'DD-mon-YYYY HH24:MI')
GROUP BY latlon, intervaloTempo
```

A consulta anterior restringe os registros da tabela de fatos àqueles que tenham correspondência com o intervalo de data definido na cláusula *where*. A restrição é construída com base na dimensão de tempo (*TempoDim* no exemplo). Os resumos de dados são construídos com tal conjunto de dados, agrupados por célula espacial e intervalo de tempo.

O próximo capítulo expõe alguns métodos para análise semântica de trajetórias.

Capítulo 3

Análise semântica de trajetórias sob a ótica de *stops* e *moves*

Luis Otavio Alvares, Vania Bogorny e Bruno Neiva Moreno

O estudo de trajetórias sob o ponto de vista semântico é uma área de pesquisa ainda em sua infância. A primeira visão sobre trajetórias semânticas foi proposta em 2008 pelo grupo de pesquisa do professor Stefano Spaccapietra, da Escola Politécnica Federal de Lausana, na Suíça, no contexto do projeto europeu GeoPKDD, denominado *stops and moves*.

3.1 O modelo *stops* e *moves*

Até 2006 os estudos na área de trajetórias consideravam basicamente a geometria das trajetórias para fazer análise desses dados. No fim de 2006, publicado oficialmente em 2008, o trabalho de Spaccapietra *et al.* criou um modelo conceitual para trajetórias chamado de *stops* e *moves*.

O modelo de trajetórias proposto pelos autores define os *stops* como as partes importantes da trajetória do ponto de vista da aplicação, onde o objeto móvel permaneceu por um determinado intervalo de tempo. Um *move* é definido como a parte da trajetória (ou a subtrajetória) que liga (i) dois *stops* consecutivos, (ii) a parte da trajetória entre o início da trajetória e o primeiro *stop*, (iii) a parte da trajetória entre o último *stop* e o fim da trajetória ou (iv) entre o instante inicial e final da trajetória caso não exista nenhum *stop*. Nesse modelo, todas as partes da trajetória que não são *stops* representam *moves*.

Ao considerarmos, por exemplo, um museu como um *stop*, em que o objeto móvel (uma pessoa) ficou por duas horas, entende-se, do ponto de vista da aplicação, que o objeto ficou parado no *stop*, pois se mostra relevante a permanência da pessoa no museu por certo tempo. Provavelmente a pessoa visitou as várias partes do museu nesse período de tempo e, portanto, não ficou parada, mas isso não seria relevante do ponto de vista da aplicação. O que seria importante é que a pessoa ficou, digamos, das 10:05 horas até as 12:05 horas no museu.

Ao contrário de aplicações convencionais para fins operacionais, em que primeiramente se projeta o sistema para depois utilizá-lo e instanciá-lo, na área de análise de trajetórias, em geral, os dados existem antes de haver um sistema para manipulá-los ou extrair deles conhecimento e informações interessantes. Por isso, tal processo é similar à engenharia reversa de bancos de dados, em que a partir dos dados se constrói o esquema conceitual, ou seja, o modelo de *stops* e *moves* seria gerado com base nesses dados. Esse processo levou vários pesquisadores a elaborar vários métodos de instanciação do modelo de *stops* e *moves* levando em conta diferentes características das trajetórias para encontrar *stops* e *moves*. Alguns dos métodos serão detalhados a seguir.

3.2 Métodos para instanciação de *stops* e *moves*

Os métodos mais conhecidos para extração de *stops* e *moves* de um conjunto de trajetórias são: SMOT (*Intersection-Based Stops and Moves of Trajectories*), CB-SMOT (*Clustering-Based Stops and Moves of Trajectories*) e DB-SMOT (*Direction-Based Stops and Moves of Trajectories*).

3.2.1 O método SMOT

Os trabalhos sobre trajetórias de objetos móveis, até 2007, consideravam essencialmente o aspecto geométrico da trajetória. O uso de aspectos semânticos é proposto inicialmente com o artigo “A model for enriching trajectories with semantic geographical information” (ALVARES *et al.*, 2007). Esse texto mostra que consultas SQL diretamente sobre os dados das trajetórias podem ser complexas e exigir muito tempo de processamento. Outro aspecto é que, para a grande maioria das aplicações, as trajetórias devem ser consideradas no contexto em que ocorrem, e não isoladamente. Por exemplo, saber que uma trajetória passa por determinados hotéis, restaurantes e museus pode ser muito importante para uma aplicação na área de turismo, e isso exige que informações da região por onde a trajetória passa sejam levadas em conta. Alvares *et al.* (2007) propuseram um algoritmo para enriquecer as trajetórias com informação do contexto em que elas ocorrem por meio do modelo *stops* e *moves*.

O algoritmo SMoT instancia os *stops* e *moves* da seguinte forma:

- *Stops* são as partes de uma trajetória que interceptam um objeto geográfico de interesse da aplicação por um tempo mínimo;
- *Moves* são as demais partes da trajetória.

A ideia é adicionar informação geográfica relevante para a aplicação que está sendo considerada. Para isso, o usuário deve informar os tipos de objetos geográficos de interesse para a aplicação. Por exemplo, no caso de uma aplicação na área de turismo, tipos de objetos relevantes podem ser: hotéis, aeroportos, restaurantes, museus, parques etc. Em uma aplicação de tráfego urbano ou mobilidade urbana, é possível avaliar: ruas, semáforos, rotatórias, escolas, estádios, *shopping centers* etc. Esses objetos geográficos são denominados candidatos a *stops* (*candidate stops*, no original). O usuário também precisa informar o tempo mínimo de permanência da trajetória em cada tipo de objeto geográfico para que este seja considerado um *stop*. Por exemplo, no contexto de uma aplicação turística, uma parada de 30 minutos em um hotel pode ser entendida como irrelevante, enquanto uma parada de 30 minutos em um restaurante pode ser muito relevante para a aplicação.

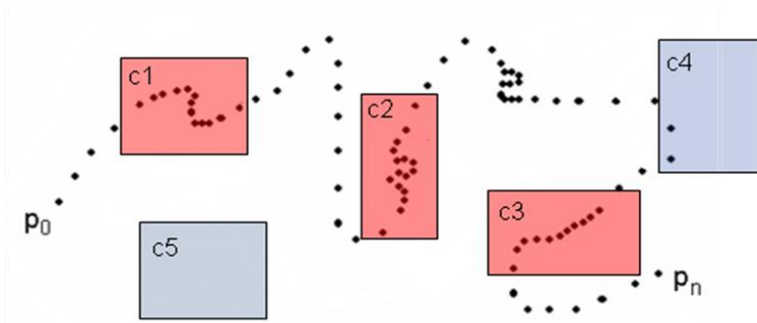
O algoritmo é relativamente simples; há um exemplo na figura 3.1. Para cada trajetória existente no conjunto de dados verifica-se a sua intersecção com os objetos relevantes para a aplicação (candidatos a *stop*). Os candidatos a *stop* (*c1*, *c2*, *c3* e *c4*) na figura 3.1 (B) têm intersecção com a trajetória, enquanto *c5* não tem. Para os candidatos a *stop* com intersecção com a trajetória, examina-se o período de tempo (a duração) dessa intersecção com a trajetória. Se a duração for maior que o tempo mínimo estabelecido para aquele tipo de objeto geográfico, o objeto será considerado um *stop*, o que ocorreu com os candidatos *c1*, *c2* e *c3*. Já em relação ao candidato *c4*, a intersecção foi por um tempo inferior ao tempo mínimo estabelecido para aquele tipo de objeto geográfico, de modo que não preenche os requisitos para ser um *stop* (veja figura 3.1 (C)). O passo seguinte é identificar cada *stop* com o nome do objeto geográfico interceptado e caracterizar o tempo de início do *stop* (tempo de início da intersecção da trajetória com o objeto geográfico) e fim do *stop* (tempo final da intersecção). Com isso, há três *stops* para a trajetória da figura 3.1 (identificados pelo nome) e quatro *moves* (as subtrajetórias representadas pelos pontos entre os *stops*): do ponto p_0 ao *stop* Museu do Louvre, do *stop* Museu do Louvre ao *stop* Hotel Ibis, do *stop* Hotel Ibis ao *stop* Museu d’Orsay e do *stop* Museu d’Orsay até o ponto p_n , que corresponde ao fim da trajetória.

Figura 3.1 – O método SMoT

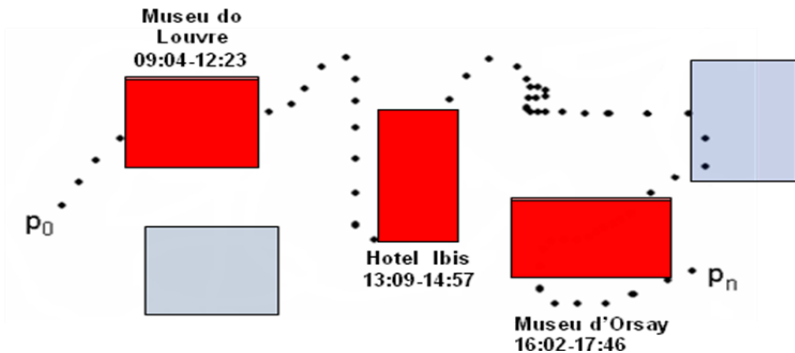
(A)



(B)



(C)



É possível deixar 9:04?

O algoritmo gera como resultado duas tabelas no banco de dados, uma de *stops* e outra de *moves*, com os seguintes atributos:

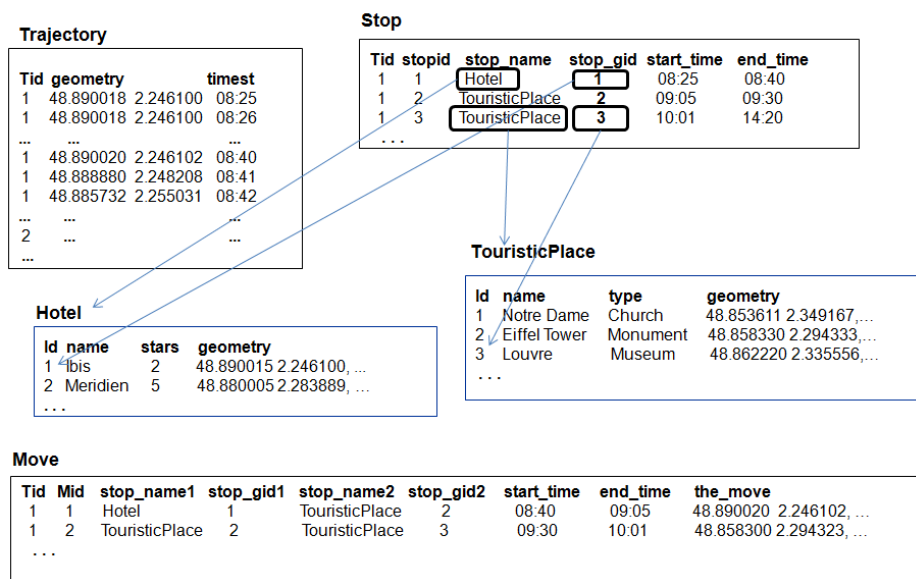
```
STOP (Tid integer, stopid integer, stop_name varchar, stop_gid integer, start_time timestamp, end_time timestamp, the_geom geometry)
```

```
MOVE (Tid integer, Mid integer, stop_name1 varchar, stop_gid1
integer, stop_name2 varchar, stop_gid2 integer, start_time
timestamp, end_time timestamp, the_move geometry)
```

Na tabela *Move*, cada *move* é caracterizado pelo início ou fim da trajetória e um *stop*, ou pelos dois *stops* que o delimitam.

A figura 3.2 expõe um exemplo de tais tabelas. A tabela original é a *Trajectory*. As tabelas *Hotel* e *TouristicPlace* foram definidas como objetos de interesse para a aplicação (candidatos a *stop*). As tabelas *Stop* e *Move* resultam da aplicação do algoritmo SMOt. Como se vê na figura, o atributo *stop_name* da tabela *Stop* contém o nome da tabela do candidato que originou o *stop*, e o atributo *stop_gid* corresponde ao identificador do objeto na tabela do candidato, isto é, esses dois atributos da tabela *Stop* formam uma chave estrangeira, o que permite conhecer todos os demais atributos do *stop*.

Figura 3.2 – Exemplo das tabelas do banco de dados



As consultas sobre as tabelas *Stop* e *Move* são muito mais simples e rápidas em relação à tabela *Trajectory*. Além disso, são muito mais ricas em informações que chamamos de semânticas. Por exemplo, se o usuário desejasse saber quantas trajetórias vão de um hotel para um lugar turístico, considerando a tabela *Trajectory*, seria preciso realizar operações espaciais como:

```
SELECT distinct count(t.Tid)
FROM trajectory t, trajectory u,
     hotel h, touristicPlace p
WHERE intersects (t.geometry, h.geometry) AND
     intersects (u.geometry, p.geometry) AND
     t.Tid=u.Tid AND u.timest>t.timest
```

Considerando a tabela de *stops*, não são necessárias consultas espaciais. Uma consulta convencional resolve o problema:

```
SELECT distinct count(a.Tid)
FROM stop a, stop b
WHERE a.stop_name='Hotel' AND
      b.stop_name='TouristicPlace' AND a.Tid=b.Tid
      AND a.stopid < b.stopid
```

No método SMOT (ALVARES *et al.*, 2007) a geometria dos *stops* corresponde à geometria do candidato a *stop* que a trajetória interceptou pelo tempo mínimo, ou seja, a geometria do *stop* é a geometria do objeto geográfico interceptado. Contudo a geometria do *stop* pode ser armazenada. O método CB-SMOT, descrito a seguir, armazena como geometria do *stop* a subtrajetória do *stop*.

Nesta seção foi apresentado o algoritmo mais simples de instanciação do modelo de *stops* e *moves*, que gera *stops* a partir da intersecção da trajetória com objetos geográficos relevantes para a aplicação.

3.2.2 O método CB-SMOT

A sigla CB-SMoT vem do inglês *clustering-based stops and moves of trajectories* (*stops* e *moves* de trajetórias baseados em clusterização). O algoritmo, inicialmente proposto por Palma *et al.* (2008), foi o segundo desenvolvido para a extração de *stops* e *moves* de trajetórias individuais após o algoritmo SMoT, desenvolvido pelo mesmo grupo.

Enquanto o algoritmo SMoT determinava os *stops* pela intersecção da trajetória com objetos de interesse da aplicação, o algoritmo CB-SMoT identifica partes da trajetória (*clusters*) em que a velocidade do objeto móvel é menor do que no restante da trajetória. Não é relevada a velocidade instantânea do objeto, que é muito suscetível à imprecisão do aparelho que coletou a trajetória, e sim a média da velocidade no trecho (subtrajetória) em questão, que é muito mais confiável.

Um exemplo claro para o uso desse algoritmo é na identificação de partes de tráfego lento (congestionamentos) em trajetórias urbanas.

Em sua versão original (PALMA *et al.*, 2008), o algoritmo utilizava dois parâmetros: *minTime* (duração mínima de uma subtrajetória com baixa velocidade para ser considerada um *stop – cluster*) e área, que indicava a proporção da trajetória que seria considerada como sendo com baixa velocidade. Na versão atual, descrita em Palma (2008) e publicamente disponível em Bogorny *et al.* (2011), existem três parâmetros: *minTime* (com o mesmo significado da versão original), *maxAvgSpeed* (velocidade média máxima de uma subtrajetória para que possa ser considerada um *cluster*) e *maxSpeed* (velocidade instantânea máxima aceitável dentro de um *cluster*).

O passo principal do algoritmo CB-SMoT é a geração dos *clusters* de baixa velocidade, realizada esquematicamente como descrito a seguir.

O procedimento é executado ponto a ponto para cada trajetória. Dado um ponto *p*, o algoritmo pega sucessivamente o seu vizinho mais lento (que é o ponto de menor velocidade imediatamente anterior ou posterior, levando em conta a ordem temporal) até obter um intervalo de tempo maior ou igual ao parâmetro *minTime*, ou que o vizinho mais lento tenha velocidade maior que o parâmetro *maxSpeed*. Se o vizinho mais lento tiver

velocidade maior que *maxSpeed*, ou se a velocidade média da subtrajetória obtida for maior que o parâmetro *maxAvgSpeed*, o algoritmo passa para a análise do próximo ponto. Caso contrário, é gerado um *cluster* de baixa velocidade a ser expandido pela adição de vizinhos mais lentos. Enquanto cada vizinho tiver velocidade menor que *maxSpeed* e a velocidade média do *cluster* não for superior a *maxAvgSpeed*, os vizinhos serão adicionados ao *cluster*. Em todo o processo só serão levados em conta pontos que não pertençam a outro *cluster*. Na figura 3.3 (A), considerando o ponto *p11*, o seu vizinho mais lento é o ponto *p10*. Com isso tem-se a subtrajetória $\langle p10, p11 \rangle$. O vizinho mais lento dessa subtrajetória, por sua vez, é o ponto *p12*, que também é adicionado à subtrajetória, e assim sucessivamente, até que a duração da subtrajetória seja maior ou igual ao parâmetro *minTime*, como pode ser observado pelos pontos pretos na figura 3.3 (B). Se a velocidade média da subtrajetória $\langle p9, p10, p11, p12, p13, p14 \rangle$ for menor que o parâmetro *maxAvgSpeed*, tal subtrajetória será a base de um *cluster*, que será expandido enquanto sua velocidade média for inferior a *maxAvgSpeed* e a velocidade do vizinho mais lento for menor que *maxSpeed*, chegando-se, no exemplo, ao *cluster* formado pelos pontos pretos na figura 3.3 (C), em que o fator limitante do *cluster* foi o parâmetro *maxSpeed*.

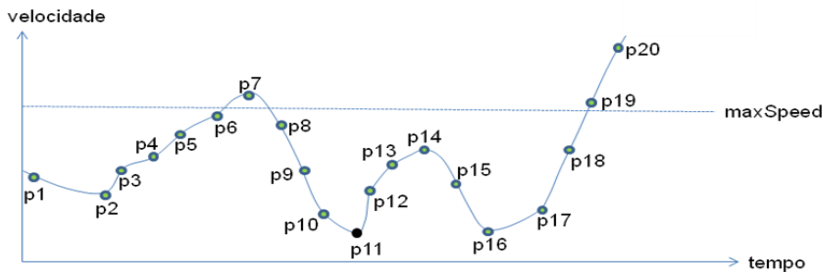
O algoritmo CB-SMoT tem dois passos principais. No primeiro definem-se os *clusters* de baixa velocidade, na segunda etapa é atribuída semântica aos *clusters* pelo teste de intersecção dos *clusters* com objetos geográficos conhecidos (especificados pelo usuário como relevantes para a aplicação).

A figura 3.4 exemplifica os passos do algoritmo CB-SMoT, que também considera as trajetórias individualmente. O primeiro passo é a determinação dos *clusters* de baixa velocidade, ilustrados na figura 3.4 (B) por quatro elipses identificadas por *g1*, *g2*, *g3* e *g4*, em que *g* se refere a uma geometria.

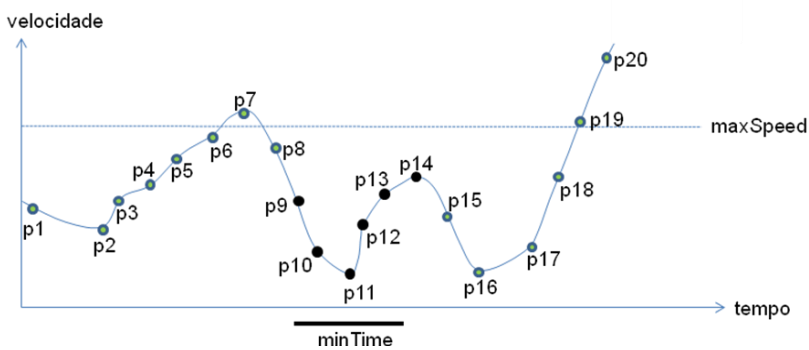
O passo seguinte consiste em adicionar semântica aos *clusters* encontrados, por meio do teste de sua intersecção com objetos geográficos de interesse da aplicação, os candidatos a *stop* (*c1*, *c2*, *c3*, *c4* e *c5*), de forma análoga ao método SMoT, como mostra a figura 3.4 (C). Como se nota na figura 3.4 (D), o *cluster g1* intercepta o objeto de interesse *c1*, que será o primeiro *stop*, identificado com o nome do objeto (Museu do Louvre, no exemplo) e caracterizado pelos instantes de início e fim da intersecção do *cluster* com o objeto geográfico (09:04 e 12:23). O mesmo acontece com os *clusters g2* e *g4*, que são identificados pelos objetos de interesse Hotel Ibis e Museu d'Orsay, respectivamente. O *cluster g3*, por sua vez, não foi identificado por nenhum objeto de interesse, caracterizando um *stop* desconhecido, pois se trata de trecho de baixa velocidade da trajetória e, portanto, de interesse para a aplicação, mas que não pode ser identificado por nenhum dos objetos definidos *a priori* como importantes.

Figura 3.3 – Identificação de *cluster* de baixa velocidade pelo método CB-SMoT

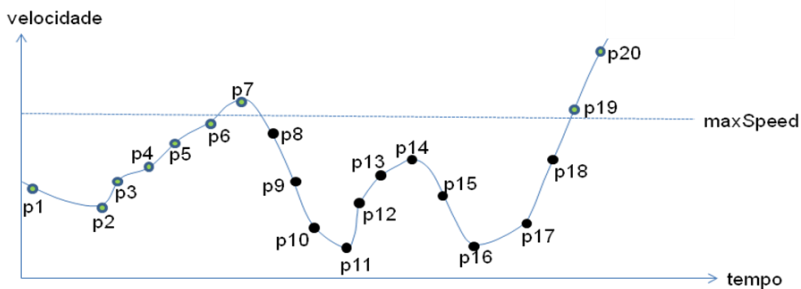
(A)



(B)



(C)

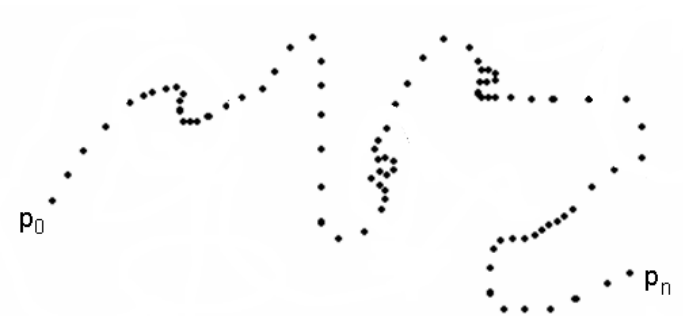


É importante mencionar que o *cluster* somente será rotulado pelo nome do objeto se a intersecção do *cluster* com o objeto atender uma duração mínima. O algoritmo CB-SMOT considera dois parâmetros de tempo mínimo, um deles para gerar o *cluster* e outro para verificar a duração da intersecção do *cluster* com os objetos geográficos. Cada objeto geográfico pode ter uma duração mínima associada. Por exemplo, a duração mínima de parada em um restaurante pode ser de uma hora, e a duração mínima de parada em um hotel pode ser seis horas, para que ambos sejam considerados *stops*.

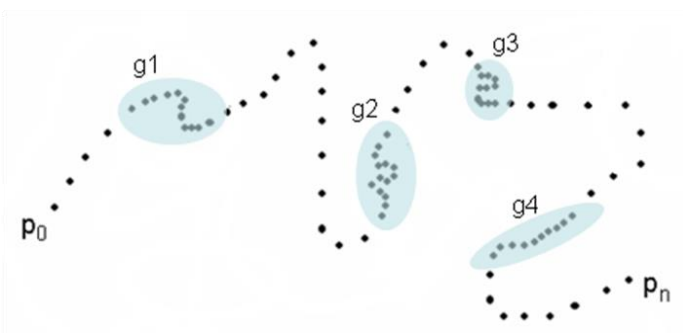
Quando os *clusters* não são rotulados com o nome de algum objeto geográfico, tornam-se *stops* desconhecidos e recebem o nome *unknown1*, *unknown2* etc. Quando dois ou mais *stops* desconhecidos se interceptam, eles recebem o mesmo nome, como na figura 3.5, em que a trajetória T_1 e a trajetória T_2 têm um *stop* desconhecido no mesmo local, por isso receberão a mesma identificação. Isso permite a descoberta de padrões de *stops* desconhecidos no caso de mineração desses dados.

Figura 3.4 – O método CB-SMoT

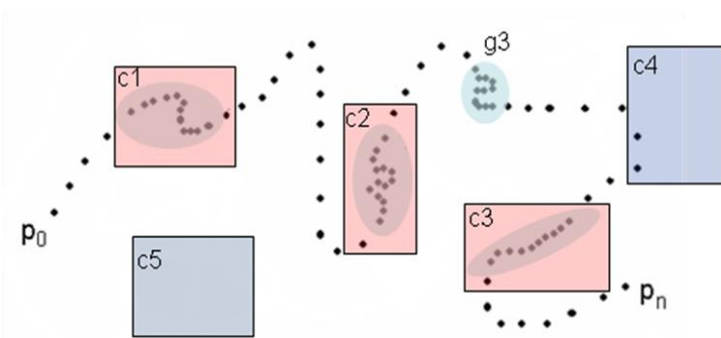
(A)



(B)



(C)



(D)

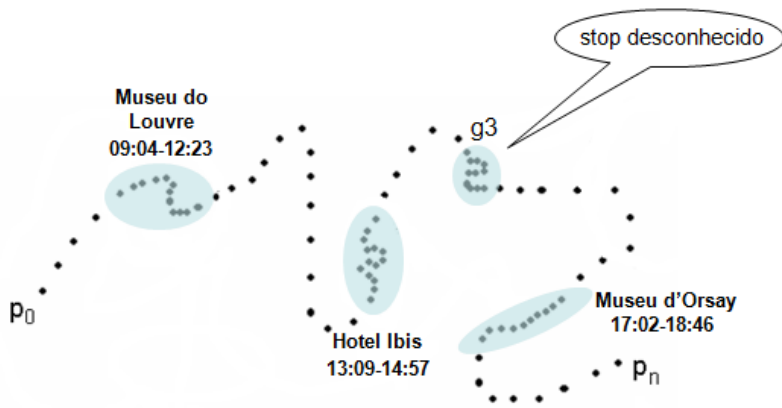
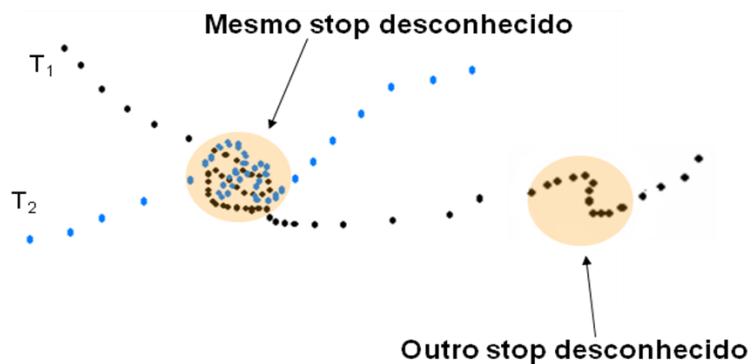


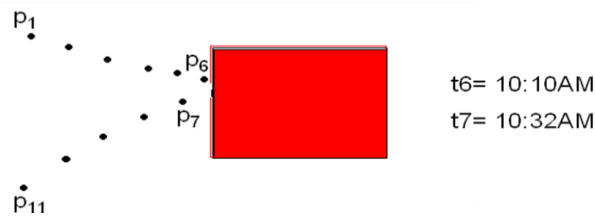
Figura 3.5 – Exemplo de *stops* desconhecidos que se interceptam



A saída do algoritmo CB-SMOT são duas tabelas, uma de *stops* e uma de *moves*, com formato similar ao das tabelas geradas pelo algoritmo SMOT. O CB-SMOT, contudo, armazena a geometria do *stop*, que representa a subtrajetória com velocidade baixa.

Um aspecto interessante é que o algoritmo CB-SMOT detecta *clusters* de baixa velocidade mesmo quando o objeto móvel entra em um prédio ou túnel e perde o sinal do GPS (figura 3.6). Vejamos a trajetória de uma pessoa que inicia no ponto p_1 : ela entra no prédio, representado pelo retângulo, passa pelo ponto p_6 e o sinal do GPS se perde. Após 22 minutos, ela sai do prédio e o ponto p_7 é gerado, continuando a trajetória até o ponto p_{11} . Se o tempo mínimo para um *cluster* (parâmetro *minTime* do algoritmo) for inferior a 22 minutos e a velocidade entre p_5 e p_6 e entre p_7 e p_8 for maior do que o parâmetro *maxSpeed*, seria detectado um *cluster* formado apenas pelos pontos p_6 e p_7 .

Figura 3.6 – Geração de *clusters* em regiões com perda de pontos ou perda de sinal



Esta seção apresentou um resumo do segundo método de instanciação de *stops* e *moves*, que identifica *stops* por intermédio da análise da variação da velocidade das trajetórias.

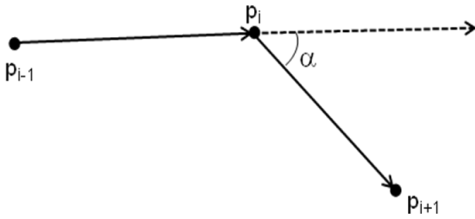
3.2.3 DB-SMOT

O terceiro método para instanciar *stops* e *moves*, denominado DB-SMOT (Direction-Based Stops and Moves of Trajectories) (Manso, Times, Oliveira, Alvares, & Bogorny, 2010), é um método de clusterização, em que os clusters são formados por partes da trajetória em que há variação de direção. A variação da direção deve ser maior do que um determinado valor, informado com parâmetro.

O método foi desenvolvido para uma aplicação de pesca, mas pode ser utilizado em outros domínios nos quais a variação da direção seja um aspecto importante. Barcos de pesca têm sido monitorados por GPS em muitos países, uma vez que a atividade está sob controle ambiental. Entretanto é difícil saber se a pesca está sendo realizada apenas em zonas autorizadas. A pesca em alto-mar, com espinhéis com dezenas de quilômetros de extensão e milhares de anzóis, pode ser decomposta em três partes: lançamento do espinhel, espera e recolhimento do espinhel. Um aspecto importante para caracterizar a atividade de pesca é a variação da direção do barco. Enquanto o barco está navegando em uma dada direção, por exemplo do porto para uma zona de pesca ou de uma zona de pesca de volta para o porto, a direção é praticamente constante, somente com pequenas correções de rumo. Já durante a pesca o barco tem variações maiores de direção, tanto no lançamento quanto na espera e no recolhimento, pois precisa cuidar para que o espinhel não seja atingido pela hélice do barco, em função da direção do vento e das ondas, bem como deve procurar se aproximar de cardumes.

O método é, portanto, baseado na variação da direção da trajetória. A figura 3.7 elucidada o conceito, tendo a subtrajetória $\langle p_{i-1}, p_i, p_{i+1} \rangle$. A trajetória tem uma direção entre os pontos p_{i-1} e p_i e muda de direção entre os pontos p_i e p_{i+1} . A variação da direção ocorrida corresponde ao ângulo α . Define-se então a variação da direção no ponto p_i como α .

Figura 3.7 – Variação de direção entre pontos de uma trajetória



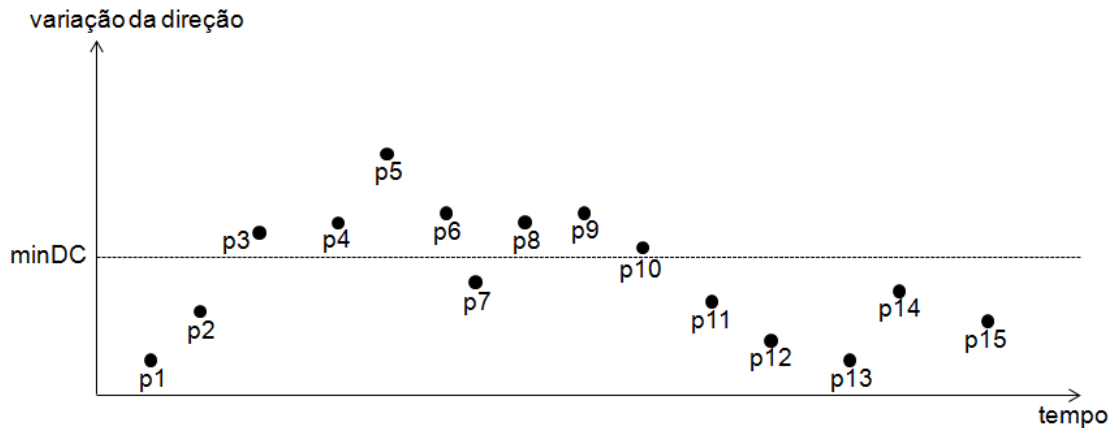
O método DB-SMoT, analogamente ao CB-SMOT, possui dois passos principais. Primeiramente são determinados os *clusters* com variação de direção acima do parâmetro de direção mínima, depois se atribui semântica aos *clusters*, pelo teste de intersecção dos *clusters* com objetos geográficos conhecidos (especificados pelo usuário como relevantes para a aplicação).

Para a geração dos *clusters* empregam-se três parâmetros: *minDC* (variação mínima da direção), *minTime* (duração mínima de uma subtrajetória com variação da direção para ser considerada um *stop – cluster*) e *maxTol* (tolerância, em número máximo de pontos contíguos sem grande variação da direção, aceitável em um *cluster*).

O passo principal do algoritmo DB-SMoT é a geração dos *clusters* de grande variação de direção, realizada esquematicamente como descrito a seguir. O algoritmo verifica a variação da direção de dois em dois pontos da trajetória. Quando a variação ultrapassar o limiar mínimo de mudança de direção (*minDC*), então os pontos são adicionados ao *cluster*. Se a mudança de direção for inferior a *minDC*, observa-se a tolerância máxima (*maxTol*), para ver se o ponto que não mudou significativamente de direção foi ruído ou se a mudança de direção acabou. Depois de adicionar os pontos que têm suficiente variação de direção ao *cluster*, confere-se se ele passa pela restrição do tempo mínimo de duração (*minTime*) de um *cluster*. Em caso positivo, considera-se a subtrajetória um *cluster* de grande variação de direção. Em uma aplicação de pesca, a duração de um *stop* de lançamento ou recolhimento pode durar, tipicamente, de duas a sete horas.

A figura 3.8 mostra o algoritmo de geração dos *clusters*. A análise começa pelo ponto p_1 , que não tem variação de direção maior que o parâmetro *minDC*. O mesmo ocorre com p_2 . Já o ponto p_3 tem variação de direção suficiente (maior que *minDC*) para dar início a um possível *cluster*, assim como os pontos p_4 , p_5 e p_6 . Já o ponto p_7 não apresenta variação de direção suficiente. Analisa-se então o parâmetro *maxTol* (que no exemplo da figura 3.7 tem o valor 1). Como p_7 é o primeiro ponto com variação de direção menor que *minDC* e *maxTol*=1, passa-se ao ponto seguinte, p_8 . Como p_8 tem variação de direção maior que *minDC*, a análise continua. Os pontos p_9 e p_{10} também demonstram grande variação de direção, mas o ponto p_{11} não. Como *maxTol*=1, parte-se para o ponto seguinte, ou seja, p_{12} , que também não tem variação de direção maior que *minDC*. Com isso, a subtrajetória que forma um possível *cluster* é $\langle p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10} \rangle$. Por último examina-se a duração dessa subtrajetória; se a duração for maior ou igual ao parâmetro *minTime*, a subtrajetória forma um *cluster*.

Figura 3.8 – Exemplo de formação de *cluster* com $maxTol=1$



O segundo passo é a atribuição de semântica aos clusters, que é realizado de forma semelhante ao método CB-SMoT. Se os clusters interceptam algum objeto geográfico de interesse (candidatos a *stop*) por um tempo mínimo, o *stop* é rotulado com o nome do objeto geográfico, senão será gerado um *unknown stop*. Na sequência, os *stops* são armazenados no banco de dados.

A seção 3.2 apresentou um conjunto de métodos diferentes para instanciar o modelo de *stops* e *moves*. Contudo a única semântica dada aos *stops* por tais métodos é o nome do objeto geográfico, quando os *stops* interceptam algum objeto geográfico de interesse da aplicação e por um tempo mínimo. O próximo tópico apresenta um método desenvolvido para fazer análise interna de *stops* e *moves*, analisando o comportamento da subtrajetória e utilizando uma base de conhecimento com informações sobre o domínio da aplicação.

3.3 Análise interna de *stops* e *moves*

Os métodos existentes para extração de *stops* (ALVARES *et al.*, 2007; MANSO *et al.*, 2010; PALMA *et al.*, 2008) só agregam semântica aos *stops* por meio da sobreposição dos *stops* da trajetória com as informações geográficas dos locais onde os *stops* ocorreram. Quando as informações geográficas não estão disponíveis, os métodos rotulam os *stops* como desconhecidos (*unknown stops*, em inglês), conforme dito anteriormente. Entretanto existem casos em que, mesmo se os *stops* forem conhecidos, se mostra interessante o estudo do comportamento dos objetos móveis dentro dos *stops*, para averiguar os padrões de movimento. Por exemplo, imagine que um *stop* foi gerado nas trajetórias de pessoas em um *shopping center*. Nessa situação pode ser interessante analisar o comportamento das pessoas dentro do local. Com base nas trajetórias, extraem-se padrões mediante uma análise sobre a variação da direção, da velocidade e do intervalo de tempo de seus *stops*.

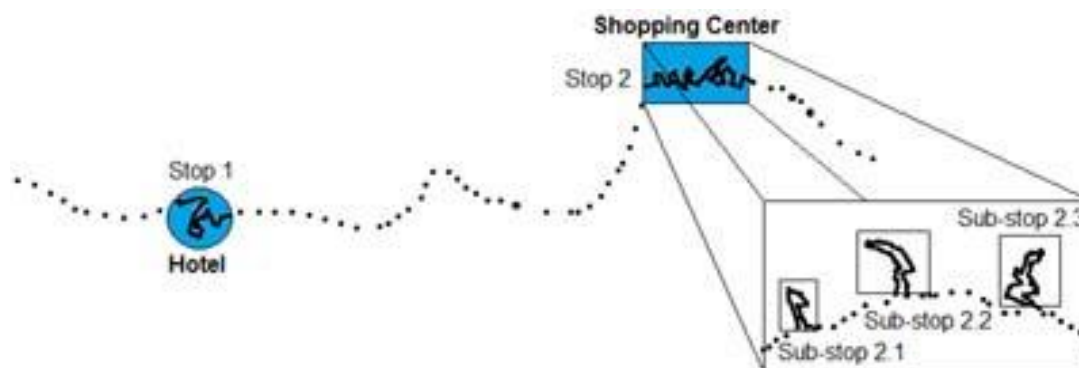
Um *stop* tem atributos espaciais e não espaciais, como visto na seção 3.2. O atributo espacial pode ser uma área (por exemplo, a área do candidato a *stop*) ou o conjunto de pontos da subtrajetória que forma o *stop* (linha). Um *move*, que é uma subtrajetória, caracteriza-se por uma linha.

Um *stop* e um *move* são caracterizados pelo atributo não espacial tempo: tempo inicial e tempo final, tanto do *stop* quanto do *move*. Os pontos da trajetória dentro desse intervalo representam uma subtrajetória. Por exemplo, se na trajetória de um veículo que trafega pela cidade for detectado um *stop*, este pode ter sido gerado em uma região de engarrafamento. Sabe-se, portanto, que naquela região o veículo continuou realizando seu movimento, mesmo que com velocidade inferior à velocidade média no restante da trajetória. Denomina-se a trajetória do veículo, dentro do *stop*, de subtrajetória do *stop*.

Tendo como base a subtrajetória, é possível fazer uma análise interna dos *stops* de modo a identificar novamente dentro desses *stops* as regiões com velocidade baixa ou significativa variação da direção por meio, por exemplo, dos métodos CB-SMOT e DB-SMOT, gerando novos *stops*, chamados de *substops*. Moreno *et al.* (2010) divulgaram um método para análise interna dos *stops*. No exemplo da figura 3.9, o segundo *stop*, localizado em um *shopping center*, pode ter três *substops*. Aplicando o método CB-SMOT sobre a subtrajetória do *stop* no *shopping center*, identificam-se *substops* dentro de um cinema e dentro de um restaurante, onde a velocidade do objeto chegaria próxima a zero. Com o método DB-SMOT, torna-se possível identificar um *substop* quando o indivíduo caminha olhando vitrines e entra em lojas, variando sua direção. A mesma análise pode ser aplicada sobre os *moves*, gerando *submoves*.

Uma vez identificados os *substops* ou *submoves*, utilizam-se informações do contexto da aplicação para dar semântica aos *substops*, podendo assim identificar as atividades que o objeto móvel realizou dentro do *stop* ou do *move*. Para usar as informações de contexto de uma aplicação na análise de trajetórias pode-se recorrer a uma base de conhecimento. Essa base de conhecimento deve conter informações sobre atividades específicas de contexto para *stops* e *moves* de trajetórias e subtrajetórias em um domínio específico. No exemplo do *shopping center* poderiam ser empregadas informações que caracterizam uma pessoa no cinema, fazendo compras, e em um restaurante. Para trajetórias geradas por pessoas em um parque florestal, por exemplo, é possível definir informações que se referem a movimentos específicos de atividades a serem realizadas em tal lugar. Por exemplo, a variação média de velocidade quando uma pessoa está caminhando é diferente da variação média da velocidade quando uma pessoa está andando de bicicleta.

Figura 3.9 – Exemplo de *stops* de uma trajetória com *substops*



Para interpretar as atividades de um objeto dentro de um *stop* ou um *move*, existe um conjunto de características que serve para qualquer domínio de aplicação: (i) duração; (ii) variação de direção; (iii) variação de velocidade; opcionalmente, (iv) comprimento da subtrajetória do *stop* ou do *move* e (v) semântica da atividade.

Em uma aplicação de um *shopping center*, um *substop* com duração de uma hora e meia, com pouca variação da direção e velocidade próxima de zero, poderia caracterizar

um *substop* em restaurante. Um *substop* com duração de duas horas ou mais, sem variação de direção e velocidade, poderia ser um *substop* no cinema. Um *substop* com duração de seis ou oito horas, com alguma variação de direção e velocidade, poderia caracterizar um grande *substop* de um indivíduo que trabalha no *shopping*.

Observa-se, então, que a análise interna dos *stops* e *moves* de uma trajetória é realizada por meio de suas subtrajetórias, processando ponto a ponto da subtrajetória com o objetivo de determinar se o comportamento do objeto móvel é semelhante ao de alguma regra definida na base de conhecimento. O algoritmo recebe como entrada um conjunto de informações de contexto representadas em uma base de conhecimento, um conjunto de *stops* e *moves* e retorna *stops* e *moves* contextualizados, ou seja, um conjunto de *stops* e *moves* com semântica.

Primeiro o algoritmo tenta inferir diretamente a semântica do *stop*, analisando a variação da direção e velocidade na base de conhecimento. Por exemplo, se um *stop* em um parque teve a duração de uma hora e com variação máxima de velocidade de 7 km/h, a semântica do *stop* é inferida como caminhada. Entretanto, se o *stop* não satisfaz os parâmetros da base de conhecimento, o algoritmo inicia a análise da subtrajetória dos *stops* e aplica os métodos CB-SMOT e DB-SMOT, descritos no início deste capítulo, para encontrar os *substops*. Cada um desses métodos utiliza os parâmetros definidos na base de conhecimento para gerar os *clusters*. O CB-SMOT usa a velocidade e a duração, e o DB-SMOT, a duração e a variação da direção. Uma vez gerados os *substops* com variação da direção e com velocidade baixa, infere-se automaticamente a semântica do *substop* por intermédio da base de conhecimento.

Quando um *stop*, *substop*, *move* ou *submove* analisado é contextualizado com mais de uma semântica, analisa-se se mais de uma trajetória tem o mesmo comportamento. Tal característica serve para diferenciar possíveis semânticas ambíguas. Por exemplo, se em um *shopping center* uma trajetória tem um *stop* interpretado como cinema e restaurante, verifica-se se um conjunto de trajetórias apresenta aproximadamente o mesmo tempo de início e fim do *stop*. Em caso positivo, o *stop* é interpretado como cinema, e não restaurante, pois uma sessão de cinema possui um tempo bem definido de início e fim, ao contrário de um restaurante, que está aberto em um determinado período e os clientes entram e saem espaçadamente ao longo desse período.

Este capítulo expôs um resumo do modelo conceitual de *stops* e *moves* em trajetórias proposto por Spaccapietra *et al.* (2008), além de três métodos para instanciar o modelo de *stops* e *moves*: SMOT, que gera *stops* com base na intersecção de trajetórias com objetos geográficos de interesse; CB-SMOT, que gera *stops* com base na variação da velocidade, e os *stops* são subtrajetórias com velocidade baixa; e DB-SMOT, que gera *stops* com base na variação da direção da trajetória, e os *stops* são subtrajetórias com significativa variação de direção. Trouxe também um método que analisa as subtrajetórias dos *stops* e *moves*, com base na variação da direção e velocidade, empregando também uma base de conhecimento para interpretar *substops* e *submoves*.

O próximo capítulo fala sobre alguns trabalhos que fazem análise de comportamento em trajetórias.

Capítulo 4

Análise de comportamento em trajetórias

Alisson Moscato Loy e Fernando Lucca Siqueira

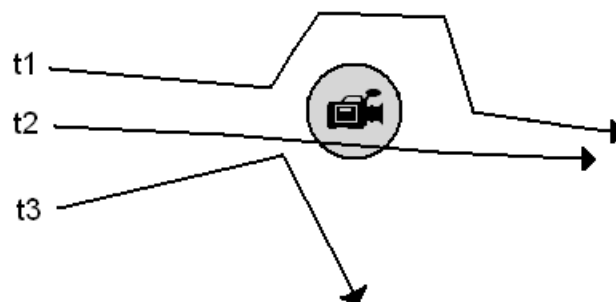
Atualmente existem poucos trabalhos na literatura que exploram o comportamento de trajetórias. A maioria tenta buscar padrões em grupos de trajetórias com base em regiões densas e similaridade de algum atributo, como, por exemplo, geometria/formato similar, velocidade similar, mesmo caminho percorrido etc. Neste capítulo serão discutidos dois trabalhos que focam na análise de comportamento em trajetórias, mais especificamente em desvios e perseguição.

4.1 O padrão comportamental *avoidance*

A maior parte dos estudos de padrões em trajetórias de objetos móveis tem seu foco na busca por comportamentos similares entre as trajetórias (e.g., objetos móveis que trafegam no mesmo caminho, no mesmo intervalo de tempo, realizam paradas nos mesmos locais). Diferentemente, o padrão comportamental *avoidance* ou desvio, proposto por Alvares *et al.* (2011), caracteriza um comportamento onde um objeto móvel evita determinados objetos ou regiões espaciais.

Tal padrão comportamental ocorre, por exemplo, quando pessoas evitam transitar próximo a alguma região enquanto caminham em um parque, veículos que mudam de rota ante situações de tráfego lento ou ainda indivíduos que se deslocam em um ambiente de forma suspeita, evitando câmeras de vigilância ou postos de segurança. A figura 4.1 ilustra o último exemplo. Supondo que a região circular representa a área de captura de imagem de uma câmera de vigilância, o comportamento das trajetórias $t1$ e $t3$ apresenta-se como suspeito, pois o comportamento esperado era que os objetos seguissem na sua direção original e interceptassem a região circular, tendo sua imagem registrada pela câmera, tal como a trajetória $t2$.

Figura 4.1 – Exemplo de trajetórias que transitam próximo a uma câmera de vigilância



Para entender as variáveis envolvidas na identificação do padrão comportamental *avoidance*, será considerada uma aplicação específica na área de segurança em que se examina o comportamento de pessoas se movendo em regiões monitoradas por câmeras de vigilância.

Uma câmera de vigilância tem um determinado ângulo e distância para captar as imagens. Algumas trajetórias vão atravessar a região de alcance da câmera, que é estática, enquanto outras, eventualmente, mudarão seu percurso de modo a evitá-la. O termo estático, aqui relacionado à região de alcance da câmera, designa regiões que não mudam de localização geográfica durante todo o processo de identificação das ocorrências de *avoidance*. As regiões estáticas são denominadas objeto-alvo.

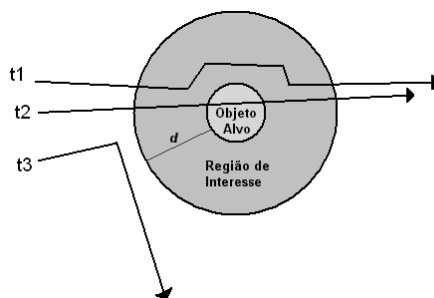
De acordo com o exemplo das câmeras de vigilância, o comportamento natural esperado por um indivíduo é que ele transite por uma região monitorada ignorando a presença delas, interceptando as regiões de captura da imagem, e, portanto, sendo filmado por onde passar. Com base nisso, o padrão comportamental *avoidance* ocorre quando um indivíduo se move em direção a uma determinada câmera e muda de direção de maneira a evitar cruzar pela área de captura de imagem. Com isso, tem-se a primeira condição para existência de um *avoidance*:

A trajetória de um objeto deve ir à direção do objeto-alvo e, num dado instante, mudar de direção de forma a evitá-lo, de modo que não ocorra nenhuma intersecção de tal trajetória com esse objeto-alvo.

É importante observar que deve haver um limite máximo onde essa mudança de direção para evitar a câmera seja considerada no intuito de evitar ser filmado. Por exemplo: Um pedestre que muda de direção a várias quadras de uma câmera de vigilância certamente não mudou seu trajeto com o intuito de evitar cruzar por aquela câmera, já que está muito distante dela. Com isso definiu-se uma região específica onde a procura pelo padrão comportamental será realizada. Tal região foi denominada região de interesse. A região de interesse envolve o objeto-alvo e é formada por um *buffer* de tamanho variável informado pelo usuário e simbolizado pela letra *d*.

Na figura 4.2 vê-se um exemplo de como a região de interesse se dispõe ao redor do objeto-alvo e da maneira com que as trajetórias se relacionam com essa região. O *buffer* que forma a região de interesse limita a análise das trajetórias até uma distância específica do objeto-alvo. Assim, as trajetórias ou partes de trajetórias que estiverem a uma distância maior que *d* serão descartadas da análise, uma vez que uma eventual mudança de direção fora dessa área não seria entendida como uma intenção de evitar o objeto-alvo (veja o exemplo da trajetória *t3* da figura 4.2, que não é tida como relevante no exemplo). A opção por um valor maior para a distância *d* pode incluir a trajetória *t3* na análise se a mesma interceptar a região de interesse, tornando-a então relevante.

Figura 4.2 – Exemplo de região de interesse, objeto-alvo e desvio de trajetórias



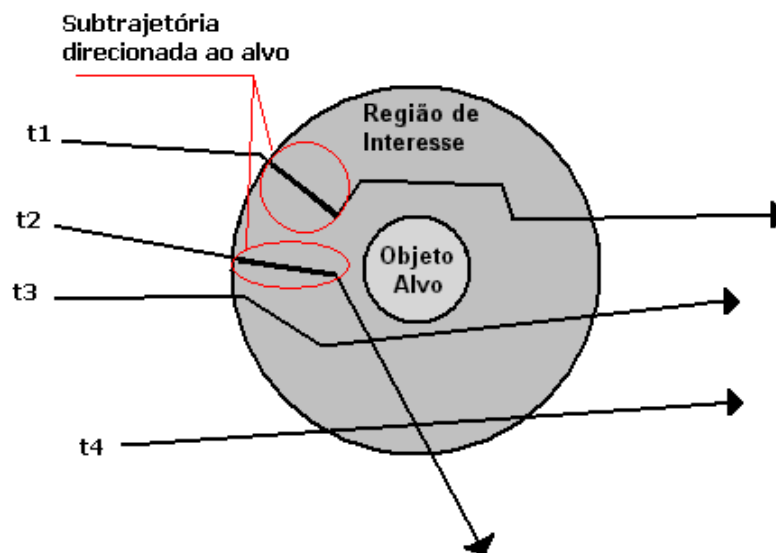
Com base nesse raciocínio, novas condições são necessárias para a correta identificação do padrão comportamental procurado:

Para ser um *avoidance*, a trajetória deve interceptar a região de interesse do objeto-alvo, e a mudança de direção da trajetória que caracteriza a intenção de evitar a interseção com o objeto-alvo precisa ocorrer dentro da região de interesse que envolve o objeto-alvo.

Para tornar mais robusta a identificação do padrão comportamental e menos susceptível a eventuais erros nos dados da trajetória, definiu-se que o objeto móvel deve se deslocar por um percurso mínimo dentro da região de interesse em direção ao objeto-alvo antes de mudar de direção, para só então se caracterizar um *avoidance*. O tamanho mínimo dessa porção de trajetória, denominada subtrajetória direcionada ao alvo, cujo símbolo é l , dependente do contexto da aplicação e do tipo de dado de trajetória a ser analisado, assim seu valor é dinâmico e deve ser fornecido pelo usuário especialista.

A figura 4.3 exemplifica o conceito. As trajetórias $t1$ e $t2$ apresentam subtrajetórias direcionadas ao alvo, pois dentro da região de interesse se movem na direção do objeto-alvo por uma distância $\geq l$. A trajetória $t3$ tem uma subtrajetória muito pequena ($< l$) que vai em direção ao objeto-alvo. Já a trajetória $t4$ em nenhum momento foi em direção ao alvo. Salienta-se que, para a porção da trajetória que se move em direção ao alvo dentro da região de interesse receber a denominação de subtrajetória direcionada ao alvo, ela deve ser maior ou igual ao valor de l , caso contrário a subtrajetória direcionada ao alvo não existirá.

Figura 4.3 – Exemplo de subtrajetória direcionada ao alvo



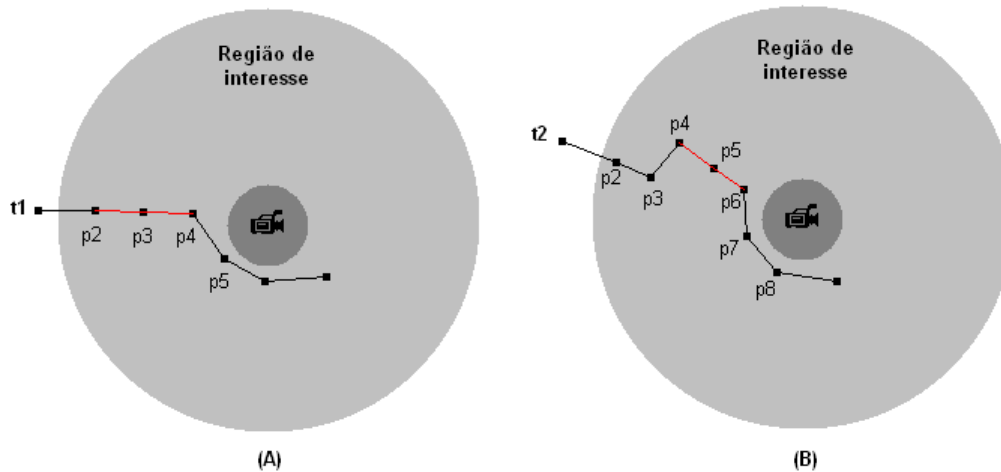
Com isso, acrescenta-se outra condição para existência do *avoidance*:

A trajetória deve apresentar uma subtrajetória direcionada ao alvo.

Se a região de interesse for grande o suficiente, podem ocorrer situações em que o objeto móvel se desloca mais de uma vez em direção ao objeto-alvo antes de evitá-lo definitivamente. A subtrajetória direcionada ao alvo deve ser a maior porção da trajetória, dentro da região de interesse, que vai na direção do objeto-alvo. Um exemplo está na figura 4.4. Enquanto na figura 4.4 (A) a trajetória $t1$ se mantém na direção do objeto-alvo desde o primeiro ponto dentro da região de interesse até o ponto em que é detectado o *avoidance* em $p4$, a trajetória $t2$ (figura 4.4 – B) apresenta duas subtrajetórias que se movem em direção ao objeto-alvo. A primeira é entre os pontos $p2$ e $p3$ e a segunda é

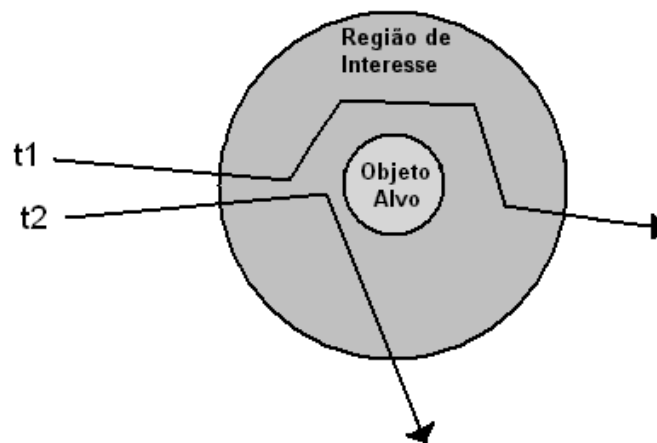
entre os pontos $p4$ e $p6$. No exemplo, a distância entre $p4$ e $p6$ será testada e, sendo maior ou igual a l , será considerada a subtrajetória direcionada ao alvo de $t2$ em relação ao objeto-alvo.

Figura 4.4 – Exemplo de subtrajetórias direcionadas ao alvo



Observando o comportamento das trajetórias de objetos móveis que evitam uma câmera de vigilância (figura 4.5), nota-se que algumas trajetórias retomam seu caminho original após evitar a câmera, como na trajetória $t1$, enquanto outras seguem pelo novo caminho, como a trajetória $t2$. Intuitivamente, é possível considerar que o comportamento *avoidance* é mais forte na trajetória $t1$, que, após evitar a câmera de vigilância, retorna ao seu trajeto original, como se a única ou a maior motivação para a mudança de direção fosse justamente não ser flagrada pela câmera, como ocorre com $t1$.

Figura 4.5 – Exemplo de comportamento da trajetória quando realiza o *avoidance*



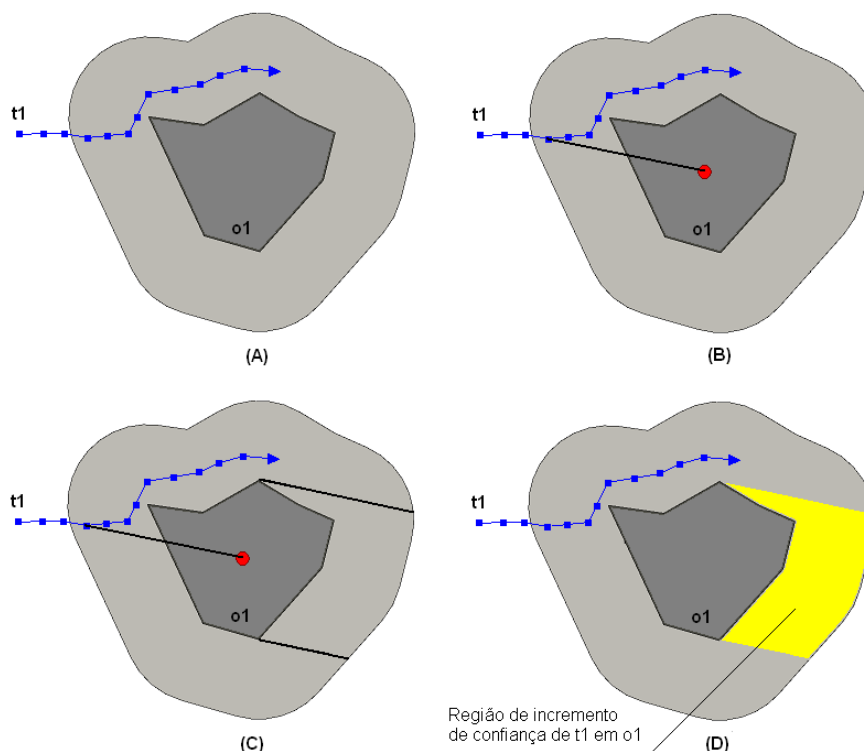
Com o objetivo de verificar se a trajetória retorna ao seu caminho original, foi criada uma região de teste de intersecção denominada região de incremento de confiança, utilizada para aumentar a confiança de que um desvio foi um *avoidance* intencional. A próxima seção descreve a confiança do *avoidance*.

4.1.1 Confiança do *avoidance*: região de incremento da confiança

Essa região está situada entre o objeto-alvo e a borda (limite) da região de interesse, no lado oposto ao primeiro ponto da subtrajetória direcionada ao alvo. A largura da região é dada pela área delimitada por duas retas paralelas e tangentes ao objeto-alvo, com uma inclinação do ângulo formado por uma linha que liga o centro geométrico do objeto-alvo ao primeiro ponto da subtrajetória direcionada ao alvo.

A figura 4.6 mostra como é calculada e posicionada a região de incremento de confiança em um objeto-alvo não circular. Na figura 4.6 (A), a trajetória $t1$ intercepta a região de interesse do objeto-alvo $o1$ correspondente a uma geometria não circular, realizando o *avoidance*. A figura 4.6 (B) ilustra a linha que conecta o primeiro ponto da subtrajetória direcionada ao alvo e o centro geométrico do objeto-alvo. Na figura 4.6 (C) visualizam-se as duas linhas paralelas e tangenciais ao objeto-alvo que delimitarão a região a ser criada. Tais linhas acompanham a inclinação da linha gerada em (B). Na figura 4.6 (D) vê-se a região de incremento de confiança gerada para $t1$ em relação ao objeto-alvo $o1$.

Figura 4.6 – Exemplo de região de incremento de confiança para objeto-alvo não circular

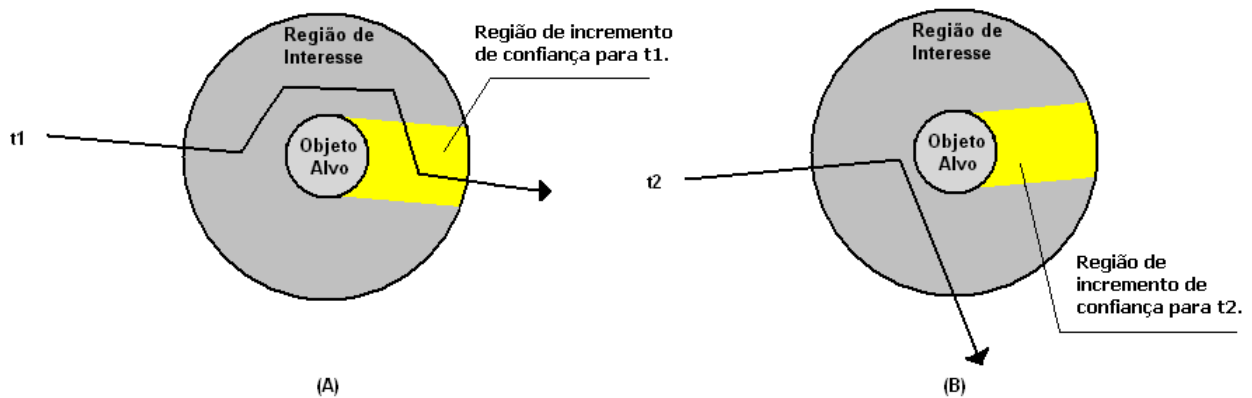


No exemplo da figura 4.7 notam-se a posição e o formato da região de incremento de confiança quando o objeto-alvo é uma região circular. Pela imagem, admite-se que a trajetória $t1$ apresenta um *avoidance* com maior grau de certeza, uma vez que intercepta a região de incremento de confiança na figura 4.7 (A), enquanto a trajetória $t2$ realiza o *avoidance* mas não intercepta a região de incremento de confiança, resultando em um menor grau de certeza (figura 4.7 – B).

Com base na figura 4.7, o teste de intersecção com a região de incremento de confiança permite caracterizar dois graus de confiança para a ocorrência do *avoidance*, denominados índice de confiança local. O valor atribuído a essa confiança local, denominada Av_{ik} , é computado a cada intersecção de uma trajetória com uma região de interesse, podendo ser: *weak* (valor 0,5) ou *strong* (valor 1,0), da seguinte forma:

- *Weak avoidance* (0,5): Quando uma trajetória intercepta a região de interesse, tem uma subtrajetória direcionada ao alvo, evita o objeto-alvo, mas não intercepta a região de incremento de confiança;
- *Strong avoidance* (1,0): Quando uma trajetória intercepta a região de interesse, tem uma subtrajetória direcionada ao alvo, evita o objeto-alvo e ainda intercepta a região de incremento de confiança.

Figura 4.7 – Exemplo da região de incremento de confiança para um objeto alvo circular

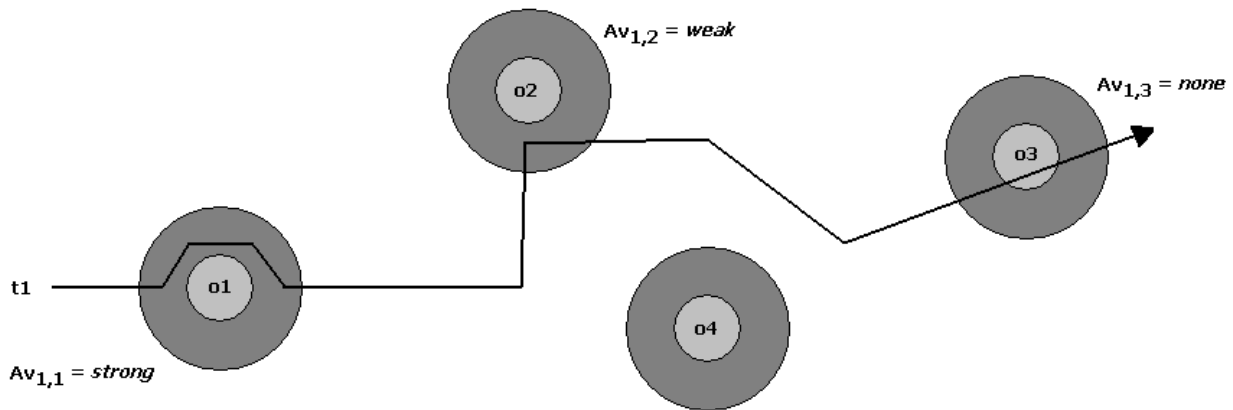


Quando uma trajetória intercepta a região de interesse, porém não evita o objeto-alvo, ou ainda, quando o evita e não apresenta uma subtrajetória direcionada ao alvo, Av_{ik} recebe *none* (valor 0,0), o que significa que não houve ocorrência de *avoidance*.

Até agora foi considerado somente um objeto-alvo. O que acontece se o objeto móvel passar por uma região com várias câmeras? Provavelmente o objeto móvel, ao se deslocar no espaço, intercepte mais de uma região de interesse, e em cada uma delas o objeto pode apresentar um comportamento distinto, realizando o *avoidance* com diferentes graus de certeza ou simplesmente interceptando as regiões dos objetos-alvo. Desse modo, consegue-se definir, além da confiança local de cada intersecção da trajetória com uma região de interesse, um índice de confiança global, cuja finalidade é avaliar o comportamento de toda a trajetória em relação aos objetos-alvo informados e indicar aquelas que têm, com maior grau de certeza, um comportamento suspeito ou de exceção, característico do padrão comportamental aqui descrito.

Na figura 4.8 a trajetória t_1 intercepta as regiões de interesse dos objetos-alvo o_1 , o_2 , e o_3 . No objeto-alvo o_1 , a trajetória realiza *avoidance* com confiança local *strong*. No objeto-alvo o_2 , a trajetória faz *avoidance* sem interceptar a região de incremento de confiança e recebe grau *weak* para confiança local. A intersecção de t_1 com o_3 não caracteriza comportamento *avoidance*, pois a trajetória intercepta o objeto-alvo, recebendo o valor *none* como confiança local. A trajetória t_1 não intercepta em nenhum momento a região de interesse do objeto-alvo o_4 , assim não há atribuição de confiança local para esse objeto-alvo.

Figura 4.8 – Exemplo de trajetória que intercepta múltiplas regiões de interesse



A identificação da confiança local da trajetória em cada objeto-alvo onde houve interação não basta para calcular a confiança global da trajetória. Antes é necessário avaliar mais profundamente as interações das demais trajetórias com os objetos-alvo, no intuito de melhorar o grau de certeza do índice. Dois fatores são calculados e influenciarão no grau de confiança global da trajetória: índice de confiança local por região de interesse (Pr_k) e fator de ponderação por objeto-alvo (w_k).

Assim como uma mesma trajetória pode interceptar diversas regiões de interesse, uma única região de interesse pode ser interceptada por diversas trajetórias distintas, como ocorre, por exemplo, em uma câmera de monitoramento posicionada em um cruzamento de avenidas movimentadas. Espera-se que num comportamento normal nem todas as trajetórias evitem tal região monitorada, com algumas interceptando a região de captura da imagem, ou seja, o objeto-alvo.

Partindo do pressuposto de que a ocorrência do padrão comportamental aqui delineado seja um comportamento de exceção, um alto índice de *avoidance* em um único objeto-alvo leva a crer que um evento desconhecido obstruiu o fluxo normal das trajetórias durante um período de tempo e fez os objetos móveis que transitaram por aquela região evitarem o objeto-alvo por motivo de força maior, como no caso de obras em vias públicas, acidentes, eventos sociais, enchentes, entre outros. Pela heurística até aqui descrita, o comportamento de evitar o objeto-alvo seria identificado como *avoidance*, resultando em um falso positivo.

4.1.2 Eliminando falsos positivos: confiança do *avoidance*

Com o objetivo de minimizar as ocorrências de falsos positivos na identificação do padrão comportamental *avoidance*, calcula-se um índice de confiança local por região de interesse Pr_k . O índice é calculado como sendo o percentual de trajetórias que realizam *avoidance* naquele objeto-alvo em relação ao total de trajetórias que interceptam a região de interesse. Esse percentual serve de referência na identificação de falsas ocorrências de *avoidance*, permitindo ao usuário uma validação das situações detectadas, pois, uma vez que o índice de confiança local por região de interesse seja muito elevado, tais ocorrências podem ser objeto de uma avaliação mais precisa e passar a ser tratadas como um evento

conhecido. A forma de tratar eventos conhecidos de modo que não gerem falsas ocorrências de *avoidance* é explicada mais adiante.

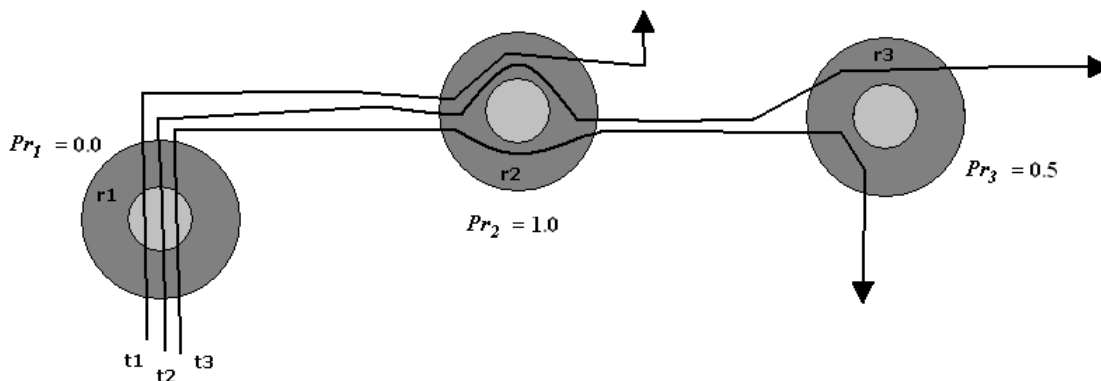
O percentual de *avoidance* por região de interesse Pr_k é calculado pela seguinte fórmula:

$$Pr_k = \frac{\sum_{i=1}^m Av_{ik}^0}{m}, \text{ para } Av_{ik} > 0$$

Em que m corresponde ao número de trajetórias que interceptam a região de interesse r_k , e Av_{ik} é a medida de *avoidance* da trajetória t_i em relação ao objeto-alvo o_k .

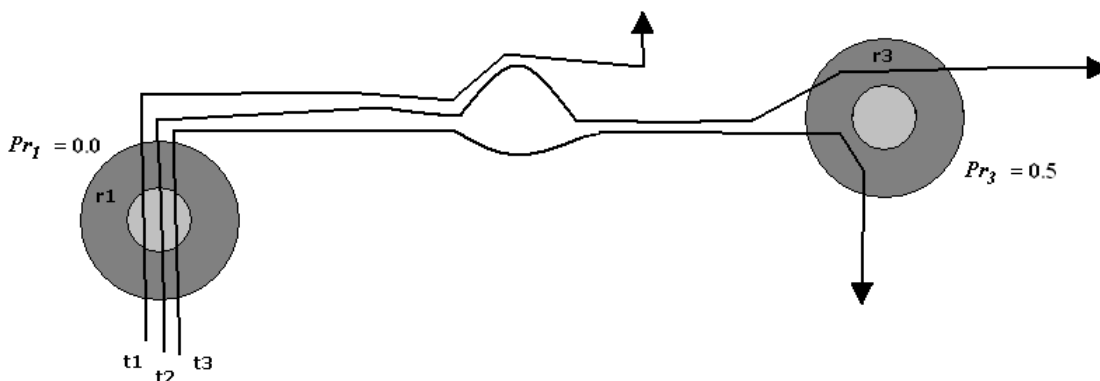
O expoente 0 (zero) garante que sejam computadas todas as ocorrências de *avoidance* identificadas na região de interesse, mesmo para o caso em que Av_{ik} seja *weak* (valor 0,5).

Figura 4.9 – Exemplo do cálculo do percentual de *avoidance* por região de interesse



A figura 4.9 traz um exemplo do cálculo de Pr_k para três regiões de interesse r_1 , r_2 e r_3 , e nota-se que todas as trajetórias que interceptam a região de interesse r_2 realizaram *avoidance*. Considerando que o *avoidance* é um comportamento de exceção, um valor de 100% sugere que havia um impedimento que bloqueou o fluxo normal das trajetórias que interceptariam o objeto-alvo, de modo que são falsas ocorrências de *avoidance*, ou seja, não foi um *avoidance* intencional, mas forçado por causas externas.

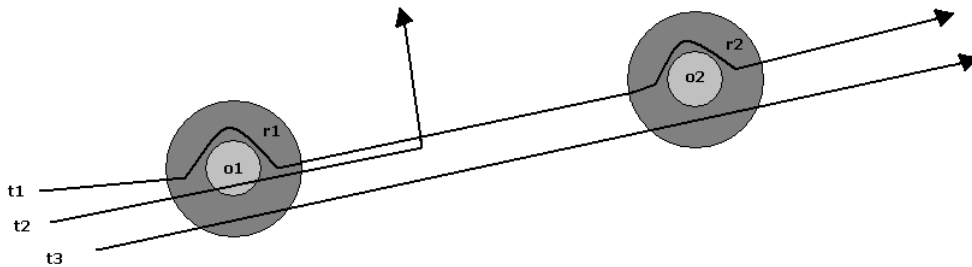
Figura 4.10 – Exemplo de exclusão do objeto-alvo o_2 por apresentar 100% de *avoidance*



Para evitar que ocorrências de *avoidance* forçado e não intencional afetem o cálculo dos índices de confiança, quando ocorrer 100% de *avoidance* para uma mesma região de interesse, elimina-se o objeto-alvo da análise. Assim, o exemplo da figura 4.9 passaria a ter dois objetos-alvo, conforme exemplo da figura 4.10, em que o objeto-alvo *o2* foi eliminado por apresentar $Pr_2 = 1.0$.

Ainda analisando o comportamento de múltiplas trajetórias em relação a múltiplos objetos-alvo, há casos em que nenhuma trajetória intercepta um determinado objeto-alvo onde ocorreu *avoidance* (veja o objeto alvo *o2* da figura 4.11). Não há nenhuma trajetória que intercepta o objeto-alvo *o2*. Dessa forma, não é possível afirmar se havia um caminho válido que permitisse a intersecção de *t1* com *o2*, dando margem a uma falsa ocorrência de *avoidance*. Já observando o objeto-alvo *o1* na figura 4.11, em que *t2* cruzou pela região do objeto-alvo *o1*, notamos que havia um caminho válido e desimpedido que permitiu tal intersecção. Isso quer dizer que, se *t1* realizou *avoidance* em *o1*, com maior grau de certeza o desvio foi intencional em relação ao alvo *o1*, o que no exemplo das câmeras significa evitar ter sua imagem registrada. No caso, o *avoidance* em relação ao alvo *o1* terá um peso maior do que o *avoidance* da mesma trajetória em relação ao alvo *o2*.

Figura 4.11 – Exemplo de objetos-alvo com e sem a intersecção de trajetórias



Considera-se que a presença ou a ausência de uma trajetória que intercepta um objeto-alvo reflete na confiança global das trajetórias que realizaram *avoidance* nesse objeto-alvo. Assim, atribui-se um maior grau de certeza às ocorrências de *avoidance* identificadas em objetos-alvo onde exista pelo menos uma trajetória que o intercepte. Para mensurar matematicamente tal efeito, atribuiu-se ao objeto-alvo um fator de ponderação w_k , o qual assume o seguinte valor para o cálculo da confiança global do *avoidance*:

- 1,0: quando existe pelo menos uma trajetória que intercepta o objeto-alvo;
- 0,5: quando nenhuma trajetória intercepta o objeto-alvo.

No exemplo da figura 4.11, o fator de ponderação para o objeto-alvo *o1* será 1,0, pois a trajetória *t2* o intercepta. Para o objeto-alvo *o2*, o fator de ponderação será 0,5, já que nenhuma trajetória o intercepta.

Ressalta-se que tanto o cálculo do Pr_k quanto a identificação da ponderação w_k são sensíveis ao volume de dados de trajetórias fornecidos, sendo mais precisos quanto maior for a quantidade de trajetórias informadas. A taxa de acerto da heurística é diretamente proporcional ao volume de dados submetidos à análise.

Para avaliar o comportamento de todo o percurso da trajetória em relação aos objetos-alvo informados, definiu-se um índice de confiança global. Esta é calculada com base nos valores da confiança local dessa trajetória, de acordo com o comportamento identificado em cada objeto-alvo e em relação ao fator de ponderação do objeto-alvo onde o *avoidance* foi identificado.

O cálculo da confiança global Av_t da trajetória pode ser feito pela seguinte fórmula:

$$Av_{ti} = \frac{\sum_{k=1}^n Av_{ik} w_k}{\sum_{k=1}^n w_k}$$

Em que n corresponde ao número de objetos-alvo cuja região de interesse foi interceptada pela trajetória t_i , Av_{ik} é a medida de *avoidance* da trajetória t_i em relação ao objeto-alvo o_k , e w_k é o fator de ponderação do objeto-alvo o_k .

O índice de confiança global da trajetória permite uma comparação entre duas ou mais trajetórias, ressaltando as que apresentam comportamento mais característico. No exemplo de aplicação na área de segurança, um alto índice global de *avoidance* em uma trajetória indica um comportamento suspeito, já que o objeto móvel transitou por uma região monitorada e evitou ser filmado.

4.1.3 Eliminado falsos positivos: utilizando conhecimento do domínio

Alguns fatores que podem levar a falsos *avoidance* são eventos previamente conhecidos como feiras de rua, passeatas, jogos de futebol, pavimentação de ruas, entre outros, pois impedem o fluxo normal de veículos ou pedestres em uma determinada região espacial em certo período de tempo. Uma forma de contornar tal situação é relacionar esse conjunto de eventos conhecidos, informando sua localização espacial e período de duração, conforme ilustrado na tabela 4.1. Assim, eventos conhecidos que impedem o fluxo de trajetórias em relação a um objeto-alvo são avaliados, e os falsos *avoidances* são desconsiderados para a região de interesse no período de duração do evento.

Para identificar e eliminar ocorrências de *avoidance* geradas indevidamente em decorrência da existência de um evento externo, é necessário que tal evento seja registrado previamente. O registro deve conter, no mínimo, um identificador, a geometria da região do evento e o período de ocorrência do evento, com data e horário de início e término.

Tabela 4.1 – Exemplo da tabela de eventos para reduzir falsas ocorrências de *avoidance*

gid	Evento	Início	Fim	the_geom
0	Manutenção na rede de esgoto	2011-01-17 10:00:00	2011-01-19 15:00:00	LINestring(...)
1	Feira de rua	2011-01-23 06:30:00	2011-01-23 15:00:00	LINestring(...)
...

Esta seção apresentou a descrição de um tipo inédito de padrão de comportamento em trajetórias denominado *avoidance*. A definição formal desse padrão e o algoritmo desenvolvido para identificá-lo podem ser encontrados no trabalho de Loy (2011).

4.2 O padrão comportamental de perseguição

Outro tipo de comportamento passível de existir entre trajetórias é o de perseguição, em que um indivíduo persegue outro indivíduo. Siqueira e Bogorny (2011) foram os primeiros a trazer definições formais e um algoritmo para extrair padrões de perseguição em trajetórias.

Uma perseguição é um padrão de comportamento que envolve duas trajetórias, a trajetória-alvo e a trajetória perseguidora; a segunda persegue a primeira. Em termos gerais dizemos que:

um padrão de perseguição entre trajetórias ocorre quando duas trajetórias andam próximas no espaço e no tempo, respeitando uma distância máxima, uma tolerância de tempo e uma duração mínima.

Uma perseguição pode ser intencional, quando um objeto tem a intenção de seguir outro objeto, como um jogador marcando outro jogador, ou natural, não intencional, quando dois ou mais objetos se movem juntos por coincidência, como carros andando em uma rodovia, ciclistas em um parque, pessoas em um *shopping* etc.

O comportamento de perseguição não vai ocorrer em duas trajetórias inteiras, haja vista a trajetória do perseguidor ser diferente da trajetória do alvo, pelo menos até que se inicie a perseguição ou quando ela acaba. Dificilmente existirá perseguição do início ao fim de duas trajetórias. Assim, o algoritmo chamado Tra-Chase, proposto por Siqueira e Bogorny (2011), analisa subtrajetórias, em vez de trajetórias inteiras. A figura 4.12 delinea um exemplo de uma perseguição entre as trajetórias $t1$ (alvo) e $t2$ (perseguidor). A perseguição acontece entre a subtrajetória dos pontos ($p7$ a $p18$) de $t1$ e a subtrajetória dos pontos ($q9$ a $q21$) da trajetória $t2$.

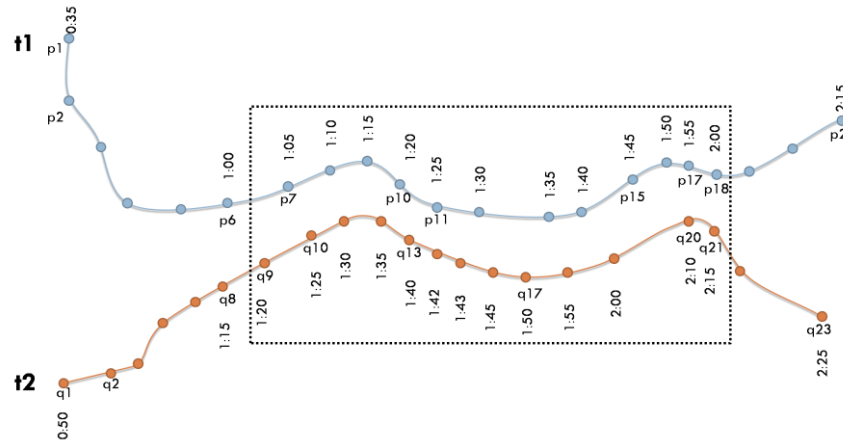
Colocar aqui figura 4.12

4.2.1 O algoritmo Tra-Chase

O algoritmo Tra-Chase é composto de três passos principais. Primeiramente identifica as subtrajetórias e verifica se estão próximas no tempo e no espaço, gerando uma subperseguição. Depois verifica a duração da soma do tempo das subperseguições; se a duração for maior do que a duração mínima necessária para caracterizar uma perseguição, gera-se então o padrão de perseguição. No terceiro passo o algoritmo identifica o tipo da perseguição. Com o tipo de perseguição serão reduzidas as perseguições naturais sem intenção, pois os tipos definidos têm características específicas, identificadas por meio do método CB-SMOT, descrito no capítulo anterior.

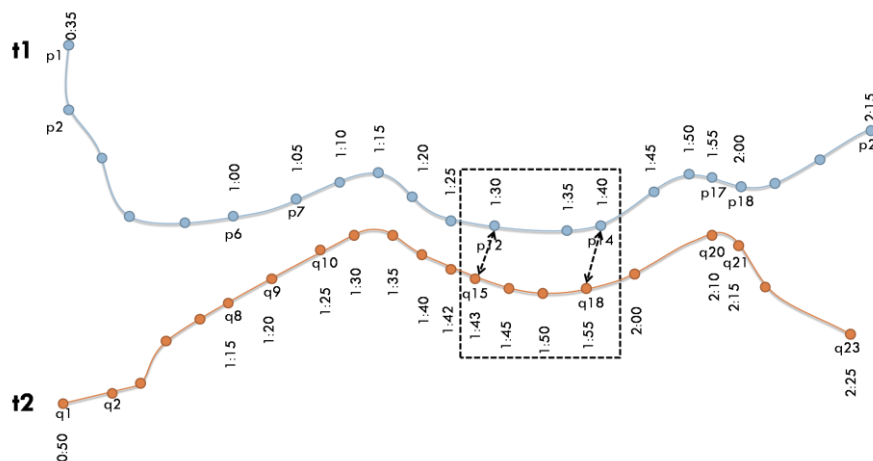
Uma perseguição não ocorre entre trajetórias com intervalo de tempo muito grande entre elas. Por exemplo, uma trajetória de dois anos atrás não vai perseguir uma trajetória de hoje, ou uma trajetória que aconteceu uma hora antes em relação à outra trajetória não caracterizará uma perseguição. Em uma perseguição imagina-se que a trajetória-alvo deve estar à frente da trajetória do perseguidor, e a diferença de tempo entre elas precisa permitir que o perseguidor não perca o alvo de vista. Assim, o primeiro parâmetro a ser considerado é a tolerância de tempo (Δt) aceitável que um alvo esteja na frente do perseguidor, ou seja, tempo máximo permitido entre um ponto p da trajetória do alvo e um ponto q da trajetória do perseguidor. No exemplo da figura 4.12, o início da perseguição acontece entre os pontos $p7$ da trajetória $t1$, que aconteceu 15 segundos antes do ponto $q9$ da trajetória $t2$. Na figura 4.12, o valor de Δt teria de ser no mínimo 15 segundos para que os pontos $p7$ e $q9$ estivessem próximos no tempo e entrassem na análise da perseguição.

Figura 4.12 – Exemplo de padrão de perseguição entre subtrajetórias



A tolerância de tempo Δt é o intervalo máximo de tempo entre duas subtrajetórias, para garantir que elas ocorreram em tempos similares. Quando duas subtrajetórias respeitam essa tolerância de tempo, formam juntas um candidato a perseguição. Por exemplo, na figura 4.13 a subtrajetória de t_1 (pontos p_{12} a p_{14}) tem intervalo de tempo 1:30 a 1:40, enquanto a subtrajetória de t_2 (pontos q_{15} a q_{18}) apresenta o intervalo de tempo 1:43 a 1:55. Para que ambas as trajetórias formem um candidato a perseguição, a tolerância de tempo Δt deve ser maior ou igual a 15 segundos, que é a maior diferença de tempo dentro da subtrajetória (tempo do ponto q_{18} – tempo 1:55 – menos o tempo do ponto p_{14} – tempo 1:40).

Figura 4.13 – Análise de tempo entre duas subtrajetórias (candidato a perseguição)

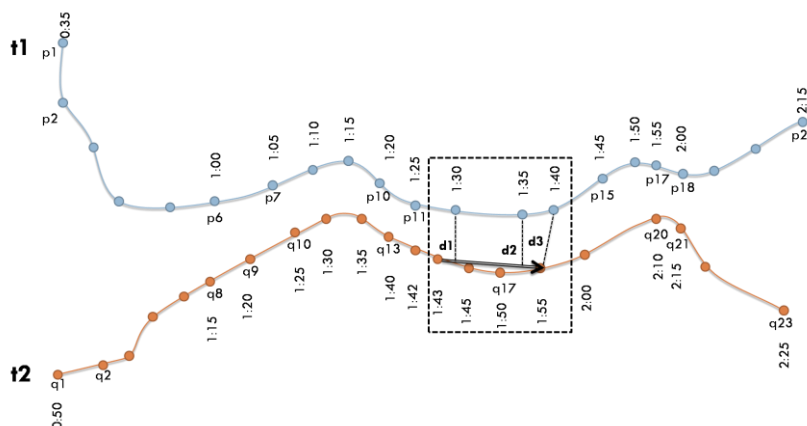


Com a definição de candidato a perseguição, o algoritmo consegue comparar subtrajetórias com intervalos de tempo diferente. Assim, o algoritmo não se prende a uma janela temporal fixa, em que cada subtrajetória precisa ter o mesmo instante de tempo inicial e final da janela temporal definida.

Para encontrar uma perseguição, duas trajetórias, além de estarem próximas no tempo, devem também estar próximas no espaço. Por exemplo, mesmo que suas subtrajetórias formem um candidato a perseguição, se a distância entre elas for de vários quilômetros, não se caracteriza uma perseguição, uma vez que a essa distância uma trajetória não está ciente da existência da outra. Para ocorrer uma perseguição, a distância entre as trajetórias tem de ser menor ou igual a uma distância mínima, chamada de Δd .

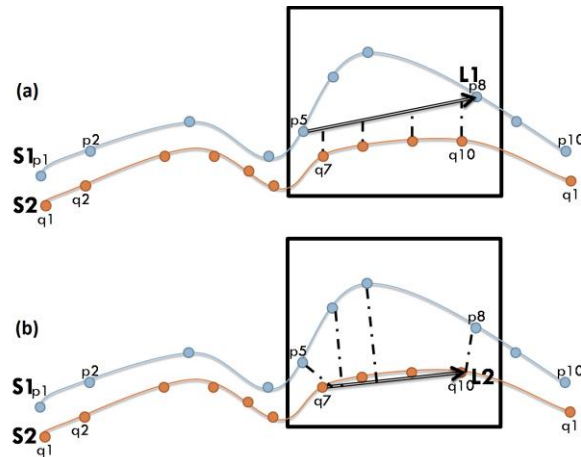
Na figura 4.14 há um exemplo do uso da distância. Para calcular a distância entre subtrajetórias, criou-se um novo conceito, denominado segmento representativo da linha, que é uma reta traçada entre o primeiro e o último ponto da subtrajetória. Um segmento representativo da linha contém duas vantagens principais. A primeira refere-se à otimização do processo, em geral custoso. Com um segmento de linha compara-se a distância de um conjunto de pontos com uma linha, processo de complexidade n , enquanto a comparação ponto a ponto entre as duas subtrajetórias tem complexidade $n \times m$. A segunda vantagem diz respeito à possibilidade de comparar a distância entre subtrajetórias com intervalos de tempo diferente. Como o método compara a distância de pontos com um segmento de linha, não é necessário que os pontos entre a reta tenham o mesmo instante de captura, uma vez que a restrição temporal já foi verificada no passo anterior. Trata-se de uma vantagem em relação a outros trabalhos, como de Cao, Mamoulis e Cheung (2006), que comparam somente pontos com exatamente o mesmo tempo de captura.

Figura 4.14 – Análise da distância entre duas subtrajetórias (subperseguição)



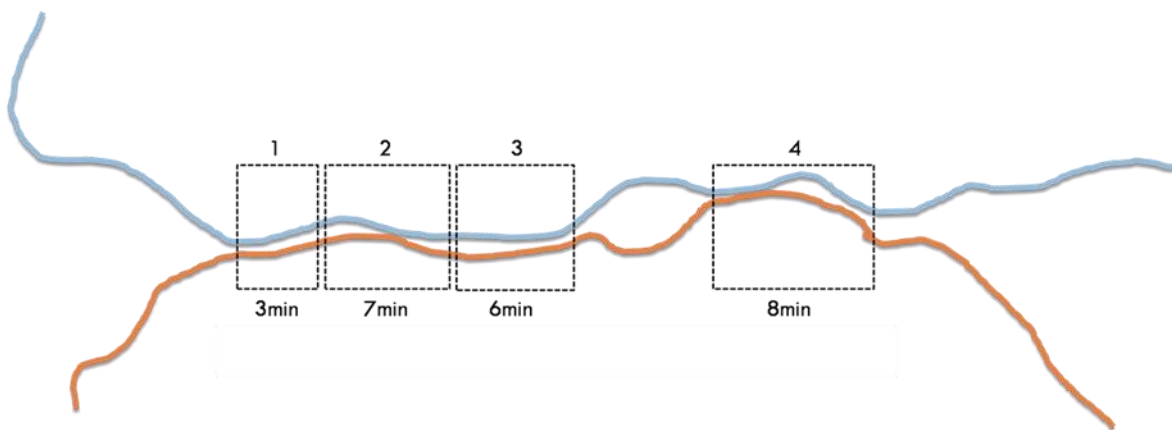
Nota-se que a verificação espacial com o segmento representativo da linha se faz dos dois lados. Isso evita casos como os da figura 4.15. Na figura 4.15 (a) a subtrajetória $\langle q7, q8, q9, q10 \rangle$ está próxima do segmento de linha $L1$, e na figura 4.15 (b) a subtrajetória $\langle p5, p6, p7, p8 \rangle$ está distante do segmento de linha $L2$. Caso o método só verificasse um lado, poderia gerar uma falsa subperseguição.

Figura 4.15 – Análise da distância entre duas subtrajetórias



Quando duas subtrajetórias satisfazem a condição temporal e espacial, ou seja, estão próximas no tempo e a uma distância menor que Δd , o candidato a perseguição transforma-se em uma subperseguição. Uma subperseguição ocorre quando duas subtrajetórias cumprem as restrições de proximidade no tempo e no espaço, porém ainda não é o suficiente para caracterizar uma perseguição, pois o mesmo comportamento pode ser observado em um curto intervalo de tempo, caracterizando um comportamento acidental ou por acaso. Para que seja uma perseguição de fato, esse comportamento deve ocorrer por um tempo mínimo, chamado Δc , conforme a figura 4.16. A soma da duração das subperseguições precisa ser maior que a duração mínima Δc . Considerando o exemplo da figura 4.16, se $\Delta c=20$ minutos, a soma da duração das quatro subperseguições (24 minutos) caracterizaria um padrão de perseguição, mesmo com as subperseguições sendo disjuntas. Se Δc fosse maior que 24 minutos, o exemplo não caracterizaria uma perseguição.

Figura 4.16 – Análise da duração das subperseguições para formar um padrão de perseguição



A duração da perseguição varia de uma aplicação para outra. Por exemplo, em trajetórias de carro a duração poderia ser 15 minutos, enquanto em trajetórias de pedestres poderia ser 30 minutos.

A definição de perseguição genérica é chamada de perseguição pura (PTC). Dependendo da aplicação, outros fatores são considerados interessantes para identificar uma perseguição, como a velocidade.

Usualmente, se a velocidade de duas trajetórias for muito distinta, elas vão se distanciar com o tempo. Mas há exceções, como, por exemplo, um perseguidor esperando o alvo se mover até certo ponto, para então segui-lo, movendo-se rapidamente para continuar vendo o alvo e alcançá-lo. Nesse caso, a velocidade média em cada subperseguição pode ser distinta, mas ainda seria uma perseguição.

Nesta seção abordaram-se definições básicas a fim de identificar um padrão de comportamento de perseguição entre trajetórias. A seguir, a última etapa do algoritmo, que identifica tipos de perseguição.

4.2.2 Tipos de perseguição

Um padrão de perseguição entre trajetórias apresenta diversos tipos de características. Apenas verificar se dois objetos estão próximos no tempo e no espaço por certo período de tempo não se mostra suficiente para assegurar com certeza uma perseguição. É difícil compreender o objetivo de uma perseguição, haja vista o padrão encontrado ser genérico. Uma perseguição genérica também pode resultar em falsos positivos, gerando uma perseguição natural, como veículos andando juntos em uma rodovia ou pedestres passeando em um *shopping*.

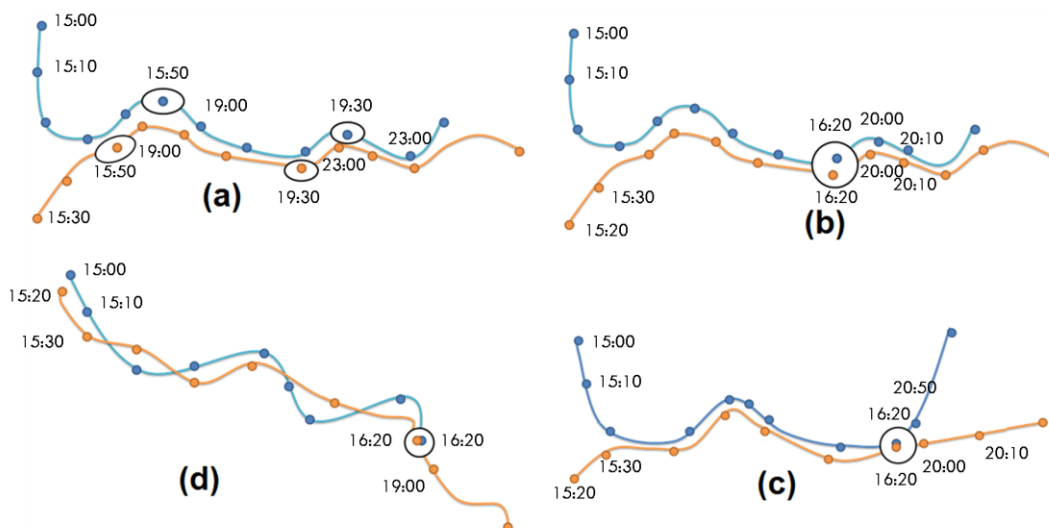
Com o objetivo de enriquecer o padrão de perseguição e de melhorar a confiança dele, introduziu-se o conceito de tipo de perseguição, cujo algoritmo é explicitado por Siqueira (2012). Os tipos definidos são espionagem, captura, assalto e caçada. Define-se o tipo de perseguição analisando os pontos de parada e movimento dentro das subtrajetórias que continham perseguição. Para isso, inicialmente são computados os *stops* das trajetórias, calculados pelo método CB-SMoT (PALMA *et al.*, 2008). Em seguida, avalia-se o comportamento das trajetórias após os *stops*.

O método CB-SMoT encontra regiões de baixa velocidade da trajetória. A figura 4.17 ilustra todos os quatro tipos de perseguição; as regiões destacadas são os *stops* das trajetórias.

Em uma perseguição, o perseguidor pode alcançar o alvo ou se manter sempre a distância. No segundo caso, o perseguidor pode agir como um espião ou detetive, seguindo o alvo para saber aonde ele está indo. Um espião vai estar sempre atrás do seu alvo, a certa distância. Quando o alvo para, o espião para também, todavia em lugar diferente do alvo, para evitar contato (figura 4.17 – a). No tipo de padrão de perseguição de espionagem, as trajetórias nunca se encontram em um mesmo local no espaço no mesmo tempo, ou seja, os *stops* de ambas as trajetórias estão próximos, porém disjuntos.

Quanto aos demais tipos de perseguição, as trajetórias vão se encontrar em algum momento. Um encontro acontece quando duas trajetórias permanecem paradas em um mesmo local e no mesmo tempo, isto é, os *stops* das duas trajetórias se interceptam no espaço e no tempo.

Figura 4.17 – Tipos de perseguição: (a) espionagem, (b) captura, (c) assalto, (d) caçada



Trocar a posição de c e d

Quando duas trajetórias se encontram durante uma perseguição, elas podem apresentar três comportamentos diferentes. O primeiro é o de captura (figura 4.17 – b). Uma captura acontece quando, a partir do primeiro encontro entre as trajetórias, qualquer *stop* consequente vai ter um encontro, pois ambas as trajetórias estarão andando juntas, como, por exemplo, a polícia perseguindo um suspeito. Assim que a polícia captura o indivíduo, ele entra no carro da polícia e os dois vão juntos até a delegacia. Como estão no mesmo veículo, mas cada um com a própria trajetória, se o carro parar, ambas as trajetórias vão gerar um *stop* naquele local e tempo.

Outro tipo de comportamento possível é o de assalto (figura 4.17 – c). Em um assalto, o criminoso persegue sua vítima e, após se encontrarem, as duas trajetórias se distanciam com o tempo. Depois do ato, o criminoso procura fugir e se afastar da vítima e do local do ocorrido o mais rápido possível. Como as trajetórias se afastam, não há mais perseguição, ou seja, o assalto ocorre no final do padrão de perseguição.

A principal característica do padrão de caça (figura 4.17 – d) é que a trajetória-alvo da perseguição termina juntamente com o padrão. Por exemplo, um leão persegue um cervo e, quando do encontro, mata (imobiliza) o alvo. A trajetória do leão continuará existindo, enquanto a do cervo terminará no local do encontro. Um assassinato ocorre de maneira similar: a trajetória do objeto-alvo acaba após o encontro.

Nesta seção definiram-se diferentes tipos de perseguição, aumentando a confiança da perseguição e adicionando semântica ao padrão de perseguição.

Este capítulo resumiu alguns métodos que analisam o comportamento individual de trajetórias. No próximo capítulo será feita uma breve descrição de ferramentas para análise de trajetórias semânticas.

Capítulo 5 Ferramentas para análise de trajetórias

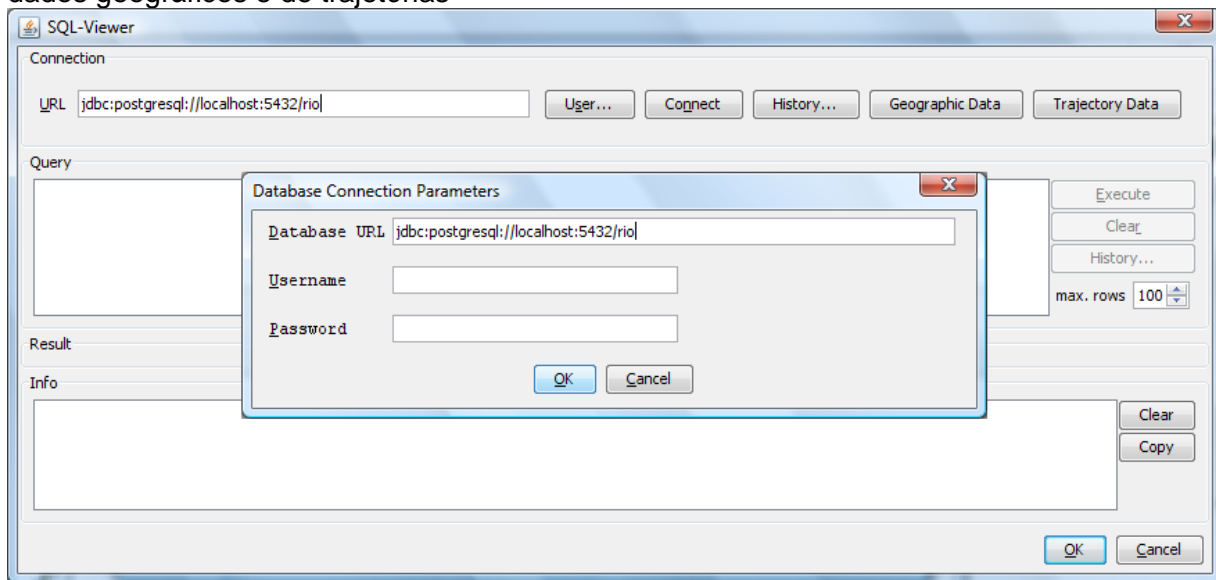
Vania Bogorny

Como as áreas de análise e mineração de trajetórias ainda são muito recentes, há poucas ferramentas que implementam os algoritmos para descoberta de conhecimento em trajetórias já elaborados até agora. O primeiro protótipo conhecido e desenvolvido especialmente para análise semântica de trajetórias é a ferramenta Weka-STPM (Weka-Spatio Temporal Pre-Processing Module), criada em 2011 em conjunto com alunos e professores da Universidade Federal do Rio Grande do Sul (UFRGS) e da Universidade Federal de Santa Catarina (UFSC) (BOGORNÝ *et al.*, 2011). Tal ferramenta constitui uma extensão do Weka, desenvolvido na Universidade de Waikato (Nova Zelândia) para mineração de dados convencionais. O Weka (IAN *et al.*, 2005) atualmente é a ferramenta de código aberto, desenvolvida em Java, mais utilizada para descoberta de conhecimento e mineração de dados em meios acadêmicos. Antes da criação do módulo Weka-STPM, criou-se o Weka-GDPM (Weka-Geographic Data Preprocessing Module) (BOGORNÝ *et al.*, 2006), uma ferramenta que pré-processa os dados geográficos para mineração de dados.

O Weka-GDPM conecta em um banco de dados geográficos via JDBC por meio de uma interface ativada pela *Geographic Data* e então calcula automaticamente os relacionamentos espaciais, escolhidos pelo usuário, entre os dados geográficos por ele selecionados do banco de dados. Os dados geográficos no Weka-GDPM são denominados de *relevant spatial features*, uma nomenclatura padrão do OpenGIS para referir-se a dados geográficos, sendo traduzidos como feições espaciais ou feições geográficas. Nessa ferramenta, se o usuário selecionar, por exemplo, as tabelas rios e bairros e a opção de relacionamentos topológicos, a ferramenta calculará o tipo de intersecção existente entre as instâncias da tabela de rios e as instâncias da tabela de bairros, criando uma nova tabela com os relacionamentos topológicos. O objetivo principal do Weka-GDPM é evitar que o usuário necessite especificar todas as junções espaciais possíveis entre as tabelas de dados geográficos para calcular os relacionamentos espaciais, já que estes são as características mais importantes dos dados geográficos a serem consideradas no processo de mineração de dados e que não estão armazenadas no banco de dados.

A figura 5.1 ilustra a interface estendida do Weka que permite o acesso ao banco de dados geográficos e de trajetórias.

Figura 5.1 – Interface do Weka para conexão no banco de dados e sua extensão para processar dados geográficos e de trajetórias



5.1 Pré-processamento de trajetórias por intermédio do Weka-STPM: adicionando semântica

Como ainda não existe um banco de dados de trajetórias comercial ou de código aberto que seja suficientemente consolidado, o Weka-STPM serve para usar o banco de dados geográficos PostGIS, a extensão do Postgres para manipular dados geográficos. Entretanto, em vez de representar um dado de trajetória com um atributo do tipo “*moving object*”, que não existe em BDG, criam-se um atributo geográfico para representar a localização (*the_geom*) e um atributo *time* para representar o tempo de cada ponto. A figura 5.2 traz um exemplo de como dados de trajetórias são armazenados em um banco de dados geográficos. O atributo *the_geom* contém o conjunto de coordenadas geográficas de cada ponto da trajetória (nesse caso é mostrada uma *string*, a forma como o SGBD PostGIS apresenta as coordenadas).

Como se observa na figura 5.2, cada ponto da trajetória fica armazenado em uma tupla (linha) da tabela, ou seja, se uma trajetória tem mil pontos, a trajetória será armazenada com mil tuplas na tabela. Note que isso prolonga o tempo de processamento de trajetórias e torna a análise dos dados muito mais lenta do que se usássemos um banco de dados espaçotemporal que armazenasse cada trajetória em uma única tupla no banco de dados.

Figura 5.2 – Exemplo de armazenamento de trajetória em BDG

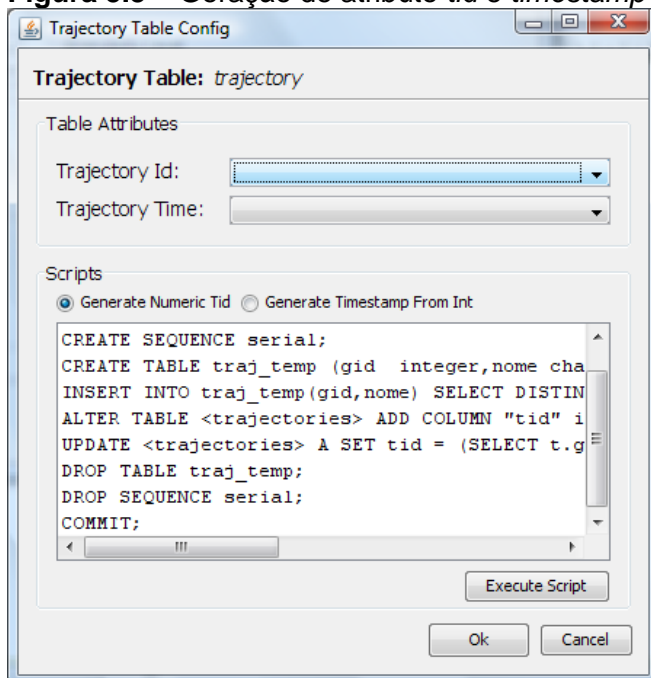
Create table trajetorias (gid integer, tid integer, time timestamp, the_geom point);

Gid	Tid	Time	The_Geom
12393	779	2007-03-05 16:29:09	“0101000000D675160591ED2441128A7F1A0F7B5C41”
12394	779	2007-03-05 16:29:10	“01010000006842F99A99ED2441644FA1750F7B5C41”
...
17298	779	2007-03-05 17:58:31	“010100000018818A584F602441137DA8A906715C41”
17299	779	2007-03-05 17:58:32	“01010000006A39A002366024417CD767F909715C41”

5.2 Exemplo de armazenamento de dados de trajetórias em bancos de dados geográficos

Para que o Weka-STPM consiga processar as trajetórias, faz-se necessário que a tabela de trajetórias tenha ao menos três atributos com estes nomes: *tid* (identificador da trajetória), que deve ser um inteiro; um atributo *the_geom*, que armazena cada ponto da trajetória; e um atributo *time*, que armazena o tempo de coleta de cada ponto. Como normalmente os dados não são gerados com todos os atributos com o respectivo nome, é preciso realizar um pré-processamento para editar o arquivo. O Weka-STPM já oferece uma interface que auxilia o usuário a fazer isso (figura 5.3). Basta escolher o atributo que representa o identificador da trajetória e o atributo que representa o tempo para que o Weka gere um *script* básico.

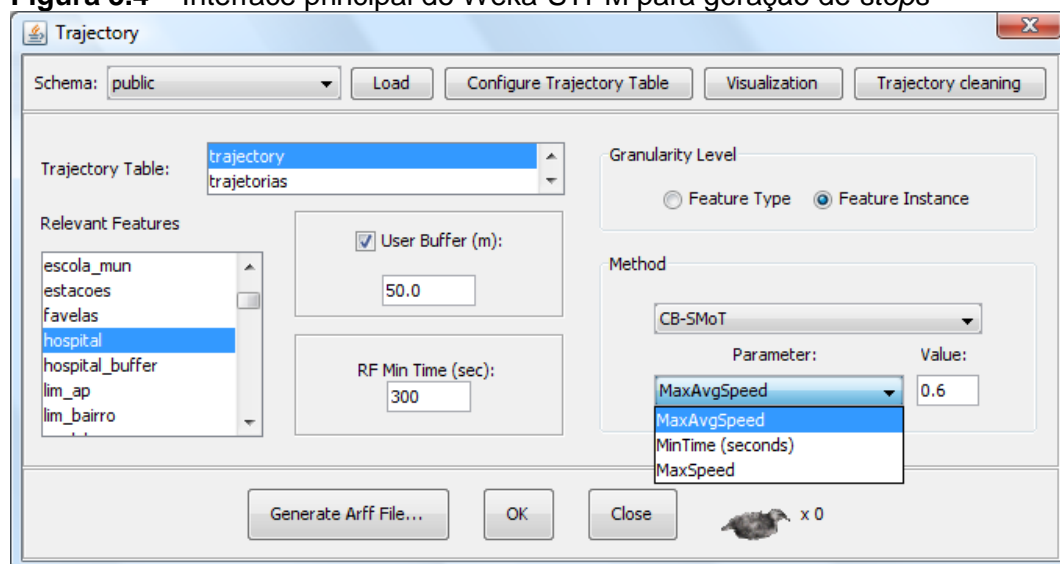
Figura 5.3 – Geração de atributo *tid* e *timestamp*



A interface principal do Weka-STPM onde o usuário pode gerar *stops* e *moves* de trajetórias consta da figura 5.4. O usuário deve, após carregar os dados do banco de dados (por meio do botão *load*), selecionar a tabela de trajetórias e as tabelas de dados geográficos de interesse da aplicação a serem utilizados na geração dos *stops*. Para cada *relevant feature* (tabela do banco de dados geográficos), o usuário precisa informar a duração mínima para que aquele lugar geográfico seja considerado um *stop* quando a trajetória interceptá-lo. A duração mínima é informada no atributo *RF Min Time* e deve ser definida para cada tabela selecionada na lista *relevant features*.

A operação de *buffer* aumenta a área dos *relevant features*, e a unidade de medida do *buffer* depende de como está configurado o banco de dados (metros ou quilômetros). Se a unidade de medida for metros, um *buffer* de tamanho 50 vai expandir as instâncias de cada *relevant feature* em 50 metros.

Figura 5.4 – Interface principal do Weka-STPM para geração de *stops*



A ferramenta Weka-STPM implementa atualmente dois métodos para adicionar semântica a trajetórias, o SMOT e o CB-SMOT, detalhados no capítulo 3. O método é escolhido no lado direito da interface da figura 5.4, e para cada método, diferentes parâmetros serão informados. No caso do CB-SmoT, o usuário deve informar a velocidade média máxima de cada trajetória para que seja considerado um *stop* (*MaxAvgSpeed*), a duração mínima de cada *cluster* gerado (*MinTime*) e a velocidade máxima aceitável para a trajetória em um *stop* (*MaxSpeed*). Esses parâmetros estão detalhados no capítulo 3.

A granularidade é um fator importante para a mineração de dados e, no caso de trajetórias, indica se os *stops* serão nomeados com o nome da tabela geográfica onde ocorreu o *stop* (exemplo: hospital, favela) ou com a instância da tabela onde ocorreu o *stop* (exemplo: Hotel Ibis, Favela da Rocinha). Para gerar *stops* do primeiro tipo a opção *feature type* deve ser selecionada, caso contrário, *feature instance*. Tal granularidade refere-se apenas à dimensão de espaço, ou seja, aos nomes ou às instâncias dos dados geográficos.

A granularidade dos dados permite descobrir diferentes tipos de padrões ou não descobrir padrões. Em geral, a granularidade é modificada diversas vezes e os dados pré-processados até que se chegue a um conjunto de padrões interessantes.

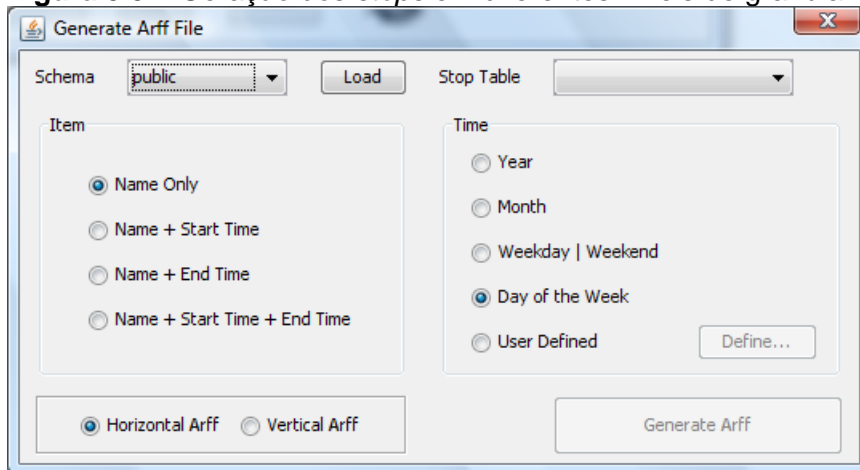
Definidos esses parâmetros, o usuário consegue gerar os *stops* clicando no botão OK, que vai gerar uma nova tabela de *stops* no banco de dados, com a estrutura apresentada no capítulo 3.

Uma vez gerados os *stops*, o usuário pode visualizar os *stops*, as trajetórias e os dados geográficos no botão *Visualization*. Também pode entrar no BD e fazer consultas na tabela de *stops*. Caso deseje modificar a granularidade do tempo ou gerar um arquivo para mineração no Weka, é necessário selecionar a opção *Generate Arff File* (figura 5.4). Nesse caso, inicia-se a transformação dos dados, detalhada na próxima seção.

5.3 Transformação dos dados para mineração

A transformação dos dados basicamente serve para definir a granularidade dos dados que serão incluídos no arquivo Arff. O arquivo texto com a extensão *arff* é o formato padrão do Weka. Nesse arquivo estão os nomes dos atributos e depois as instâncias dos dados. A geração do arquivo *arff* envolve a parte espacial e temporal dos *stops*, sendo composta de itens. A figura 5.5 ilustra como o usuário pode definir os itens.

Figura 5.5 – Geração dos *stops* em diferentes níveis de granularidade



Um item pode ser composto apenas pelo nome do *stop*, ou pelo nome do *stop* e o tempo de início do *stop*, ou pelo nome e o tempo final do *stop*, ou ainda pelo nome do *stop* e seu tempo inicial e final. Salienta-se que o nome do *stop* representa o lugar geográfico interceptado pela trajetória por um tempo mínimo, ou desconhecido (*unknown*), caso contrário.

Tendo escolhido o item, o usuário determina a granularidade de tempo, que pode ser ano, mês, dia da semana ou fim de semana, dia específico da semana, ou definida pelo usuário. No último caso a granularidade do tempo pode ser qualquer conjunto de intervalos em horas. Detalhes sobre a geração dos itens e os diferentes tipos de padrões que podem ser gerados estão expostos em Bogorny et al. (2009).

Gera-se o arquivo *arff* no formato horizontal ou vertical. A figura 5.6 traz um exemplo de arquivo *arff*. O formato clássico do Weka é o horizontal, em que cada atributo corresponde a uma coluna e cada registro corresponde a uma linha. Quando o valor do atributo é desconhecido, preenche-se com um ponto de interrogação.

Figura 5.6 – Exemplo de arquivo *arff*

```
@relation weather

@attribute outlook {sunny, overcast, rainy}
@attribute temperature real
@attribute humidity real
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}

@data
sunny, 85, 85, FALSE, no
sunny, 80, 90, TRUE, no
overcast, 83, 86, FALSE, yes
rainy, 70, 96, FALSE, yes
rainy, 68, 80, FALSE, yes
rainy, 65, 70, TRUE, no
overcast, 64, 65, TRUE, yes
sunny, 72, 95, FALSE, no
sunny, 69, 70, FALSE, yes
rainy, 75, 80, FALSE, yes
sunny, 75, 70, TRUE, yes
overcast, 72, 90, TRUE, yes
overcast, 81, 75, FALSE, yes
rainy, 71, 91, TRUE, no
```

Tendências futuras

Na área de banco de dados, o modelo relacional foi estabelecido e funciona como padrão para uma gama de aplicações. Quando se pensa em *data warehouse*, os tradicionais modelos Star e Snowflake conseguem atender a um razoável número de soluções. Em termos de bancos de dados de objetos móveis e *data warehouses* para esses dados, a pesquisa ainda está na sua infância. Na prática, ainda não existem ferramentas sólidas para armazenamento, processamento e análise de dados espaçotemporais gerados por dispositivos móveis.

Este livro não teve o objetivo de ser exaustivo, mas sim informativo sobre os trabalhos iniciais na área de análise de trajetórias, principalmente trajetórias semânticas. Há a necessidade cada vez mais de solução de problemas do mundo real inseridos no contexto espaçotemporal.

Os recentes avanços das tecnologias de satélites e o barateamento de dispositivos móveis, como aparelhos de GPS, telefone celular, redes de sensores, entre outros, têm facilitado a coleta de grandes volumes de dados espaciais e espaçotemporais. Tais dados estão presentes em uma série de problemas da atualidade que estão despertando o interesse de pesquisadores do mundo inteiro. O problema do aquecimento global envolve uma área que analisa dados espaçotemporais sobre a evolução climática nos últimos anos. Um grande desafio no tocante a dados de tempo e espaço está no problema do trânsito nas grandes capitais, poluição do ar e alastramento da violência.

Outro problema sério que vem recebendo a atenção dos cientistas em todo mundo são os desastres naturais, como terremotos, furacões e desmatamento. Eles requerem soluções inovadoras para armazenar, processar e analisar dados espaciais e espaçotemporais, bem como prever fenômenos futuros. O rastreamento de veículos mediante aparelhos de GPS e o de criminosos por meio de telefone celular estão cada vez mais populares. Análise de dados censitários, agrupamento de regiões com determinados tipos de doenças, como câncer, e muitas outras aplicações reais clamam por soluções de pesquisa envolvendo dados espaciais e espaçotemporais.

Os problemas supramencionados são apenas uma parte das dezenas de aplicações que podem ser beneficiadas pela ciência desenvolvida para análise de dados de trajetórias.

Uma área de pesquisa em aberto é o desenvolvimento de bancos de dados, *data warehouses* e algoritmos de mineração de dados que considerem informações semânticas e de contexto da aplicação. A área de análise de comportamentos específicos de trajetórias ou grupos de trajetórias também é pouca pesquisada até o momento.

Diversos trabalhos na área de semântica e contexto em trajetórias devem ser elaborados no projeto europeu SEEK (*Semantic Enrichment of Trajectory Knowledge Discovery*), envolvendo cooperação internacional entre Brasil, Itália e Grécia (<http://www.seek-project.eu/?q=node/4>), no período de 2012 a 2015.

Referências

- ALMEIDA, C. A. **Data warehouse de trajetórias**: um modelo semântico com suporte à agregação por direção dos movimentos. 2010. Dissertação (Mestrado)—Universidade Federal de Campina Grande, 2010.
- ALVARES, L. O. *et al.* A model for enriching trajectories with semantic geographical information. *In: INTERNATIONAL SYMPOSIUM ON ADVANCES IN GEOGRAPHIC INFORMATION SYSTEMS (ACM-GIS'07)*, 15., 2007, Seattle. **Anais...** Seattle, 2007. p. 162-169.
- ALVARES, L. O. *et al.* An algorithm to identify avoidance behavior in moving object. **Journal of the Brazilian Computer Society**, v. 11, n. 3, p. 193-203, 2011.
- ANDRIENKO, G. L.; ANDRIENKO, N. V.; WROBEL, S. Visual analytics tools for analysis of movement data. *SIGKDD Explorations*, v. 9, n. 2, p. 38-46, 2007.
- BOGORNY, V. *et al.* Weka-GDPM: integrating classical data mining toolkit to geographic information systems. *In: SBBB WORKSHOP ON DATA MINING ALGORITHMS AND APLICATIONS (WAAMD'06)*, 2006, Florianópolis. **Anais...** Florianópolis, 2006. p. 9-16.
- BOGORNY, V.; KUJIPERS, B.; ALVARES, L. O. ST-DMQL: a semantic trajectory data mining query language. **International Journal of Geographical Information Science**, v. 23, p. 1.245-1.276, 2009.
- BOGORNY, V.; WACHOWICZ, M. A framework for context-aware trajectory data mining. *In: CAO, L. et al. (Eds.). Data mining for business applications*. Nova York: Springer, 2008.
- BOGORNY, V. *et al.* Weka-STPM: a software architecture and prototype for semantic trajectory data mining and visualization. **Transactions in GIS**, v. 15, n. 2, p. 227-248, 2011.
- BRAZ, F. J. **Warehousing and mining aggregate measures concerning trajectories of moving objects**. 2009. Tese (Doutorado)—Università Ca' Foscari, 2009.
- BRAZ, F. J. *et al.* Approximate aggregations in trajectory data warehouse. *In: INTERNATIONAL CONFERENCE ON DATA ENGINEERING, WORKSHOP ON SPATIO-TEMPORAL DATA MINING*, 2007, Istambul. **Anais...** Istambul, 2007. p. 536-545.
- BRAZ, F.; ORLANDO, S. Trajectory data warehouses: proposal of design and application to exploit data. *In: BRAZILIAN SYMPOSIUM ON GEOINFORMATICS*, 9., 2007, Campos do Jordão. **Anais...** Campos do Jordão, 2007. p. 61-72.
- CÂMARA, G. *et al.* **Anatomia de sistemas de informação geográfica**. Campinas: Unicamp, 1996.
- CAO, H.; MAMOULIS, N.; CHEUNG, D. W. Discovery of collocation episodes in spatiotemporal data. *In: INTERNATIONAL CONFERENCE ON DATA MINING*, 2006, Hong Kong. **Anais...** Hong Kong, 2006. p. 823-827.
- CODD, E. F.; CODD, S. B.; SALLEY, C. T. Providing OLAP (on-line analytical processing) to user analyst: an IT mandate. **Codd and Date**, v. 32, p. 3-5, 1993.
- EGENHOFER, M. J. Deriving the composition of binary topological relations. *Journal of Visual Languages & Computing*, v. 5, n. 2, p. 133-149, 1994.
- GIANOTTI, F.; PEDRESCHI, D. **Mobility, data mining and privacy**. Nova York: Springer, 2008.

GÜTING, R. H. *et al.* A foundation for representing and querying moving objects in databases. **ACM Transactions on Database Systems**, v. 25, n. 1, p. 1-42, 2000.

GÜTING, R. H.; SCHNEIDER, M. **Moving objects databases**. San Francisco: Morgan Kaufmann, 2005.

HAN, J.; STEFANOVIC, N.; KOPERSKI, K. Selective materialization: an efficient method for spatial data cube construction. *In*: THE PACIFIC-ASIA CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING, 1998, Melbourne. **Anais...** Melbourne, 1998. p. 144-158.

INMON, W. H. **Building the data warehouse**. Nova York: Wiley, 1996.

KIMBALL, R. **The data warehouse toolkit: practical techniques for building**. John Wiley, 1996.

LAUBE, P.; IMFELD, S. Analyzing relative motion within groups of trackable moving point objects. *GIScience*. 2002. 132-144.

LAUBE, P.; IMFELD, S.; WEIBEL, R. Discovering relative motion patterns in groups of moving point object. **International Journal of Geographical Information Science**, v. 19, n. 6, p. 639-668, 2005.

LOY, A. M. **Um algoritmo para identificar padrões comportamentais do tipo avoidance em trajetórias de objetos móveis**. 2011. Dissertação (Mestrado em Ciência da Computação)–Universidade Federal do Rio Grande do Sul, 2011.

MALINOWSKI, E.; ZIMÁNYI, E. OLAP hierarchies: a conceptual perspective. *In*: INTERNATIONAL CONFERENCE ON ADVANCED INFORMATION SYSTEMS ENGINEERING, 16., 2004, Springer. **Anais...** Springer, 2004. p. 477-491.

MANSO, J. A. R. *et al.* DB-SMoT: a direction-based spatio-temporal clustering method. *In*: IEEE INTERNATIONAL CONFERENCE ON INTELLIGENT SYSTEMS, 5., 2010, Londres. **Anais...** Londres, 2010. p. 114-119.

MARKETOS, G. *et al.* Building real-world trajectory warehouses. *In*: INTERNATIONAL ACM SIGMOD WORKSHOP ON DATA ENGINEERING FOR WIRELESS AND MOBILE ACCESS (MobiDE'08), 7., 2008, Vancouver. **Anais...** Vancouver, 2008. p. 8-15.

MORENO, B. N. *et al.* Looking inside the stops of trajectories of moving objects. *In*: SIMPÓSIO BRASILEIRO DE GEOINFORMÁTICA (GeoInfo), 11., 2010, Campos do Jordão. **Anais...** Campos do Jordão, 2010. p. 9-20.

OGC – OPEN GEOSPATIAL CONSORTIUM. **Opengis standards and specifications: topic 5 – features**. 2008. Disponível em: <http://portal.opengeospatial.org/modules/admin/license_agreement.php?suppressHeaders=0&access_license>. Acesso em: maio 2012.

PALMA, A. T. **A clustering-based approach for discovering interesting places in trajectories**. 2008. Dissertação (Mestrado em Ciência da Computação)–Universidade Federal do Rio Grande do Sul, 2008.

PALMA, A. T. *et al.* A clustering-based approach for discovering interesting places in trajectories. *In*: ANNUAL SYMPOSIUM ON APPLIED COMPUTING (ACM-SAC'08), 23., 2008, Fortaleza. **Anais...** Fortaleza, 2008. p. 863-868.

PARENT, C.; SPACCAPIETRA, S.; ZIMÁNYI, E. Spatiotemporal conceptual models: data structures + space + time. *In*: ACM SYMPOSIUM ON ADVANCES IN GEOGRAPHIC INFORMATION, 17., 1999, Nova York. **Anais...** Nova York, 1999. p. 26-33.

- PELEKIS, N. *et al.* **Hands-on in Hermes MOD engine**. 2011. Disponível em: <http://infolab.cs.unipi.gr/hermes/Downloads/Hands_On_In_Hermes_MOD_Engine.pdf>. Acesso em: maio 2012.
- PELEKIS, N. *et al.* Hermes: aggregative LBS via a trajectory DB engine. *In: SIGMOD CONFERENCE*, 2008, Vancouver. **Anais...** Vancouver, 2008. p. 1.255-1.258.
- PostGIS. Disponível em: <<http://postgis.refrations.net>>. Acesso em: maio 2012.
- RAFFAETA, A. *et al.* visual mobility analysis using T-Warehouse. **International Journal of Data Warehousing and Mining**, v. 7, n. 1, p. 1-23, 2011.
- SIQUEIRA, F. L. **Identificação de comportamentos de perseguição em trajetórias de objetos móveis**. 2012. Dissertação (Mestrado em Ciências da Computação)—Universidade Federal de Santa Catarina, 2012.
- SIQUEIRA, F. L.; BOGORNY, V. Discovering chasing behavior in moving object trajectories. **Transactions in GIS**, v. 15, n. 5, p. 667-688, 2011.
- SPACCAPIETRA, S. *et al.* A conceptual view on trajectories. **Data & Knowledge Engineering**, v. 65, p. 126-145, 2008.
- WANG, J.; HSU, W.; LEE, M. L. A framework for mining topological patterns in spatio-temporal databases. *In: ACM INTERNATIONAL CONFERENCE ON INFORMATION AND KNOWLEDGE MANAGEMENT*, 2005, Bremen. **Anais...** Bremen, 2005. p. 429-436.
- WITTEN, I. H.; FRANK, E. **Data mining: practical machine learning tools and techniques**. 2. ed. San Francisco: Morgan Kaufmann, 2005.
- WOLFSON, O. *et al.* DOMINO: databases for moving objects tracking. *In: ACM SIGMOD CONFERENCE ON MANAGEMENT OF DATA*, 1999, Filadélfia. **Anais...** Filadélfia, 1999. p. 547-549.
- WOLFSON, O. *et al.* Management of dynamic location information in DOMINO. *In: INTERNATIONAL CONFERENCE ON EXTENDING DATABASE TECHNOLOGY (EDBT)*, 8., 2002, Praga. **Anais...** Praga, 2002. p. 769-771.