

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA**

**DESCOBERTA DE PADRÕES DE ENCONTRO EM
TRAJETÓRIAS DE OBJETOS MÓVEIS**

Areli Andreia dos Santos

**Florianópolis - SC
2013/2**

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
CURSO DE SISTEMAS DE INFORMAÇÃO

DESCOBERTA DE PADRÕES DE ENCONTRO EM TRAJETÓRIAS DE OBJETOS MÓVEIS

Areli Andreia dos Santos

Trabalho de conclusão de curso
apresentado como parte dos requisitos
para obtenção do grau de Bacharel em
Sistemas de Informação pela
Universidade Federal de Santa Catarina.
Orientadora: Prof.^a Dr.^a Vania Bogorny

Florianópolis - SC
2013/2

Areli Andreia dos Santos

DESCOBERTA DE PADRÕES DE ENCONTRO EM TRAJETÓRIAS DE OBJETOS MÓVEIS

Trabalho de Conclusão de Curso apresentado
como parte dos requisitos para obtenção do
grau de Bacharel em Sistemas de Informação.

Prof.^a Dr.^a Vania Bogorny
Orientadora

Banca Examinadora:

Prof.^a Dr.^a Carina Friedrich Dorneles

Prof. Dr. Ronaldo dos Santos Mello

Dedico este trabalho aos meus pais
Janecir João dos Santos e Andreia
Edésia da Silva, que sempre me
instigaram a estudar e ir em busca dos
meus objetivos.

AGRADECIMENTOS

Agradeço primeiramente a Deus por guiar meus pensamentos e atitudes durante todo esse trabalho. Sou grata às orações da minha bisavó Nilda que me guiaram no início da faculdade, quando eu precisava viajar todo dia para chegar à UFSC e voltava sozinha à noite para casa de ônibus. Que ela continue intercedendo por mim em sua nova morada. Agradeço também a minha avó Edésia, que sempre me aconselhou e me ensinou o valor da vida e a lutar por ela.

Agradeço aos meus pais, que sempre me encorajaram a estudar e me incentivaram nos momentos mais difíceis durante toda minha faculdade. Meu pai que sempre me ofereceu apoio durante o meu curso e trabalhou para que eu tivesse materiais sempre que precisei, me deu meu primeiro computador (e o segundo também...) e a oportunidade de conhecer um mundo novo com ele. Minha mãe não precisou de palavras para me ensinar o valor da educação, boa professora que é, me educou com seu exemplo de dedicação e coragem.

Meus agradecimentos ao meu namorado André Salvaro Furtado pelo carinho, apoio, dedicação e toda ajuda durante a realização deste trabalho. Agradeço pelas dicas de implementação, pelas revisões e pelas palavras de incentivo. E a seus pais, que permitiram que sua casa se tornasse um segundo lar para mim.

Gostaria de agradecer especialmente a minha orientadora Vania Bogorny por toda sua atenção e dedicação. Sua energia me contagiou para que todas as partes desse trabalho fossem realizadas da melhor forma possível. Obrigada por me orientar e por me propor esse desafio. Agradeço também ao Professor Luis Otávio Alvares que contribuiu em discussões e definições desse trabalho. E os professores Ronaldo dos Santos Mello e Carina Friedrich Dorneles que aceitaram fazer parte da banca e por seus comentários e contribuições.

Aos meus colegas de faculdade que sobreviveram ao curso de Sistemas de Informação. Gostaria de expressar minha admiração a vocês que se dedicaram durante o curso trabalhando e estudando. Agradeço também aos meus colegas de trabalho que ouviram minhas reclamações, discutiram ideias e coletaram trajetórias que serviram para a validação desse trabalho. Obrigada pela disponibilidade e apoio.

Também Agradeço ao grupo de Bancos de Dados da UFSC que me ajudou a coletar trajetórias para a realização deste trabalho e aos alunos que contribuíram na discussão deste. Agradeço à Universidade Federal de Santa Catarina e ao Departamento de Informática e Estatística por oferecer estrutura e cursos de qualidade para os seus alunos.

De tudo ficaram três coisas:
A certeza de que estamos começando,
A certeza de que é preciso continuar e
A certeza de que podemos ser
interrompidos antes de terminar
Fazer da interrupção um caminho novo,
Fazer da queda um passo de dança,
Do medo uma escola,
Do sonho uma ponte,
Da procura um *encontro*,
E assim terá valido a pena existir!

Fernando Sabino, 1981

SUMÁRIO

1. INTRODUÇÃO	12
2. CONCEITOS BÁSICOS E TRABALHOS RELACIONADOS	17
2.1 Conceitos Básicos	17
2.1.1 Dados Geográficos e Trajetórias	17
2.1.2 Relacionamentos e operações espaciais	19
2.2 Trabalhos Relacionados	22
2.2.1 Comportamentos Individuais	22
2.2.2 Comportamentos de grupos	24
3. DESCOBERTA DE PADRÕES DE ENCONTRO EM TRAJETÓRIAS DE OBJETOS MÓVEIS	29
3.1 Encontro Espacial, Espaço-temporal e Frequente	29
3.2 Algoritmo para detecção de encontros	33
4. EXPERIMENTOS	45
4.1 Conjunto de dados - UFSC	45
4.1.1 Resultados	47
4.2 Conjunto de dados da Gincana em Amsterdam	50
4.2.1 Resultados	51
5. CONCLUSÕES E TRABALHOS FUTUROS	54
REFERÊNCIAS	56

LISTA DE FIGURAS

Figura 1 - Exemplo de Trajetória	13
Figura 2 - Monitoramento eletrônico.....	15
Figura 3 - Conjunto de Trajetórias de Objetos Móveis, visualizada no software QuantumGIS	19
Figura 4 - Relacionamentos Topológicos [Braz e Bogorny, 2012].....	20
Figura 5 - REMO [Laube, 2004]	26
Figura 6 - Padrão Convergence	28
Figura 7 - Encontro Espacial	29
Figura 8 - Área de Encontro	31
Figura 9 - Encontro espaçotemporal	31
Figura 10 - Trajetórias T1, T2 e T3 e seus encontros	32
Figura 11 - Encontro Frequente	33
Figura 12 – Candidatos a encontro entre t_1 , t_2 e t_3	42
Figura 13 - Trajetórias UFSC	46
Figura 14 - Principais encontros espaçotemporais UFSC	47
Figura 15 - Comparativo entre trajetórias originais e selecionadas de Amsterdam...51	
Figura 16 - Encontros espaciais e espaçotemporais em Amsterdam.....	53

LISTA DE TABELAS

Tabela 1 - Exemplo de armazenamento de trajetórias em bancos de dados.....	18
Tabela 2 - Padrões REMO [Laube, 2004]	27
Tabela 3 - Encontros Espaciais.....	34
Tabela 4 - Encontros Espaço-temporais	34
Tabela 5 - Encontros Frequentes.....	34
Tabela 6 –Tempos em um encontro espaço-temporal	41
Tabela 7- Tempos em um encontro espacial	41
Tabela 8 - Valores dos principais encontros espaço-temporais UFSC.....	47
Tabela 9 - Encontros no conjunto de dados da UFSC	48
Tabela 10 - Encontros em Amsterdam.....	51
Tabela 11 - Encontros frequentes em Amsterdam.....	52

LISTA DE REDUÇÕES

GPS: *Global Positioning System*

GIS: *Geographic Information System*

RFID: *Radio-frequency identification*

BDG: Bancos de dados Geográficos

SGBD: Sistema Gerenciador de Bancos de Dados

UFSC: Universidade Federal de Santa Catarina

RESUMO

Aparelhos com recursos de localização estão se tornando cada vez mais populares. Aparelhos de GPS para carros, celulares e *tablets* estão presentes no cotidiano de grande parte da população. Dispositivos como esses são capazes de gerar dados de localização associados ao tempo, chamados de Trajetórias de Objetos Móveis. A partir desses dados foram realizadas diversas pesquisas, mas a análise e a descoberta de padrões envolvendo dados espaçotemporais é um tema que ainda tem muito a ser explorado. Neste trabalho é abordado o padrão de encontro entre trajetórias. O estudo dos encontros pode ser muito útil para diversas análises. Um exemplo disso é a comparação dos trajetos de viaturas policiais com os movimentos de presos em regime semiaberto que fazem o uso de algum dispositivo GPS. Com os dados desses objetos é possível verificar se alguns desses elementos se encontram e com que frequência. Para identificar esses encontros foi desenvolvido um algoritmo capaz de detectar, classificar os encontros em espaciais ou espaçotemporais, determinar a duração dos encontros espaçotemporais, e ainda, identificar quais desses padrões são frequentes. Para validar o algoritmo proposto foram realizados experimentos com dados que simulavam encontros entre trajetórias coletadas no Campus da UFSC e dados reais de uma gincana em Amsterdam, onde haviam locais que os participantes deveriam visitar. Os experimentos demonstraram que o algoritmo foi capaz de detectar os diversos encontros, mas que a imprecisão dos aparelhos GPS ainda é um problema a ser resolvido.

Palavras-chave: Trajetórias de Objetos Móveis, Mineração de Dados, Encontros, Frequências de encontros.

1. INTRODUÇÃO

A redução no custo de dispositivos móveis, como aparelhos de GPS e celulares, bem como os avanços nas tecnologias de satélites e sensores sem fio permitiu um aumento significativo no uso desses mecanismos. Esses aparelhos permitem registrar o movimento do indivíduo que o carrega. Dessa forma, qualquer indivíduo que carrega um dispositivo móvel (e.g. pessoas, animais, carros ou barcos), ao se mover, gera um rastro, no qual cada instante de tempo corresponde a uma localização no espaço. Esse rastro é chamado na literatura como trajetória. Segundo Braz e Bogorny (2012) “uma trajetória é representada por 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 ”, chamado também de dado espaçotemporal.

A Figura 1 apresenta um exemplo de trajetória. Nela pode-se observar que os eixos X e Y representam a latitude e a longitude, respectivamente, e o tempo está representado no eixo Z. Uma trajetória pode ser formalmente definida como [SIQUEIRA e BOGORNÝ, 2012]:

Definição 1. Trajetória. Uma trajetória t é um conjunto (tid, p_0, \dots, p_n) , onde tid é o identificador da trajetória, $p_i = (x_i, y_i, t_i)$ é um ponto da trajetória onde (x_i, y_i) são as coordenadas geográficas que representam o espaço e t_i é o instante de tempo em que a coordenada foi capturada. O conjunto $(x_i, y_i, t_i) \in \mathbb{R}$ para cada $i = 0, \dots, n$ e $t_0 < \dots < t_n$.

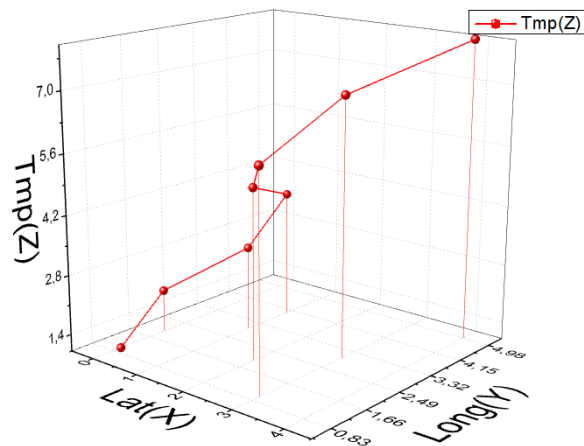


Figura 1 - Exemplo de Trajetória

Dados de trajetórias, da forma como são gerados pelos dispositivos são pouco úteis, mas podem ser analisados a partir de métodos e algoritmos que permitem a extração de conhecimento interessante. Já existem diversos trabalhos na literatura que tratam desses dados, como o que descreve desvios de trajetórias [Alvares et al., 2011] e o que permite detectar padrões que se repetem frequentemente nos mesmos lugares [Li et al., 2010]. Outros trabalhos detectam o comportamento de perseguição entre trajetórias [SIQUEIRA e BOGORNY, 2011], e ainda os diversos comportamentos definidos por [Laube, 2004], como *flock*, *leadership*, *convergence* e *encounter*.

Embora existam inúmeras publicações sobre padrões em trajetórias, poucas são as que abordam o padrão de encontro. Laube em [Laube, 2004] trata os encontros que ocorrem dentro de um raio, previamente determinado, ao mesmo tempo. [Gudsmundon, 2007] propõe melhorias ao algoritmo de [Laube, 2004] em termos de performance. Contudo, nenhum dos autores trata de encontros puramente espaciais ou da frequência com que esses encontros ocorrem.

Muito conhecimento pode ser extraído de encontros puramente espaciais, onde as trajetórias simplesmente se interceptam no espaço. Trajetórias que se interceptam no espaço, mas não ao mesmo tempo podem caracterizar um lugar

interessante, indicando o interesse de várias pessoas em um mesmo local. O padrão de encontro puramente espacial pode representar um ponto turístico que é visitado muitas vezes em vários momentos do dia, ou em dias diferentes, provocando assim um grande encontro espacial entre trajetórias. Pode ainda representar interesses específicos como uma biblioteca, que as pessoas procuram em diferentes horários para fazer pesquisas ou estudar. O simples encontro espacial das trajetórias pode ainda representar um local de trabalho, com diversos turnos diferentes, fazendo com que muitos objetos móveis se encontrem.

Aqueles encontros que acontecem ao mesmo tempo podem representar indivíduos que frequentaram um mesmo local no mesmo instante. Por exemplo, pessoas que estiveram em um mesmo bar no mesmo horário ou que assistiram a uma mesma sessão de cinema. Podem representar também indivíduos que se reuniram intencionalmente, como colegas de uma antiga turma de universidade, ou mesmo animais que se recolheram no mesmo local para descansar.

Já os encontros frequentes podem representar uma aula de dança que ocorre, por exemplo, todas as terças, às 20h. Nesse caso, várias trajetórias possuem um encontro ao mesmo tempo e repetidamente. Pode representar, ainda, um casal que se encontra em um mesmo dia no almoço e no jantar. O importante nos encontros frequentes é que o encontro entre os indivíduos seja espaçotemporal e ocorra por um número mínimo de vezes, entre os mesmos indivíduos, independentemente do local.

Identificar encontros em trajetórias pode ser especialmente importante em análise de trajetórias de presidiários em regime semiaberto, aplicação esta que motivou esse trabalho. Presidiários em regime semiaberto são indivíduos que saem da prisão durante o dia para trabalhar e voltam ao presídio a noite. Alguns desses presidiários são monitorados através de tornozeleiras com GPS. Nesse contexto, é

verificado apenas se os presidiários estão dentro das zonas permitidas, como no sistema SAC24 da Spacecom[Jornal Extra]. Este sistema encaminha alertas se o indivíduo monitorado sai do trajeto esperado ou rompe o laço da tornozeleira.

Para verificar o encontro destes indivíduos seria necessário que ao menos um agente monitorasse esses indivíduos a todo momento. Esta tarefa está sujeita a fadiga, e um encontro pode passar despercebido. A Figura 2 ilustra a tela de monitoramento do sistema SAC24. Na parte inferior da Figura 2 é apresentada uma lista com os presos que estão sendo monitorados, na parte superior, dentro do círculo menor, o mapa apresenta dois elementos próximos o UPR4 e o UPR3.

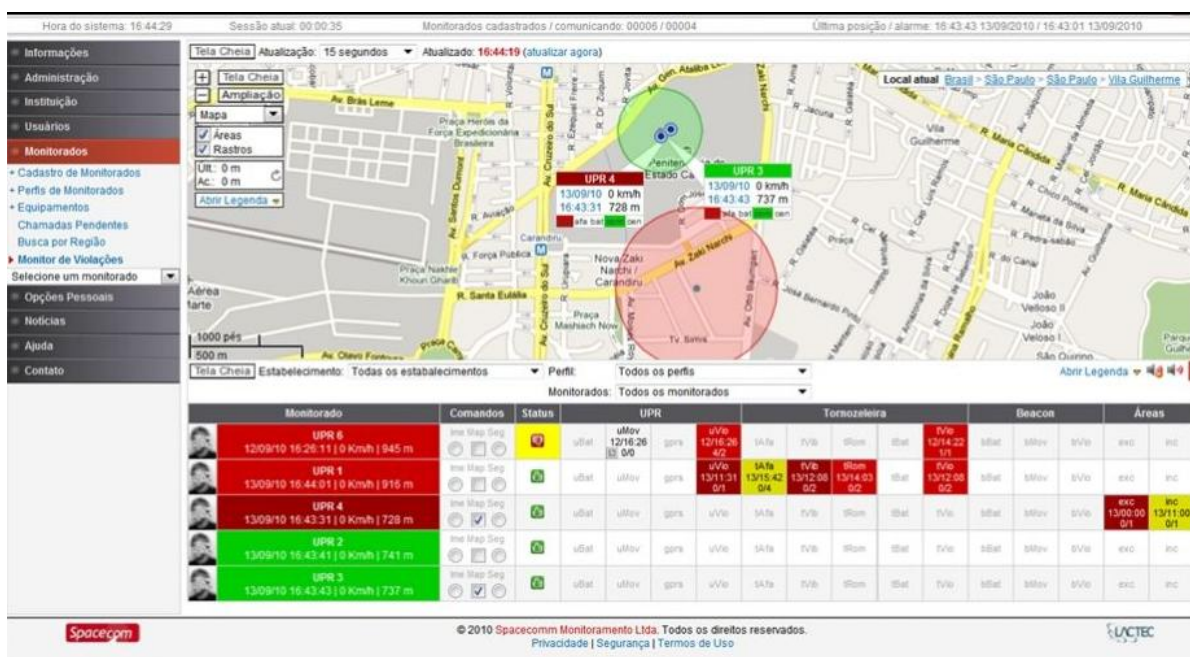


Figura 2 - Monitoramento eletrônico

Um encontro puramente espacial, para presos em regime semiaberto, pode identificar um local conhecido por vários criminosos, mas que ainda é desconhecido pela polícia. Esse local pode ser um ponto de venda de drogas ou um local aonde são vendidas armas de fogo ilegalmente. Neste contexto, o encontro espacial é caracterizado pelos objetos que se encontram em uma mesma área em

dias e horários diferentes. Se um encontro entre esses indivíduos acontecer no espaço e ao mesmo tempo, isso pode indicar uma reunião entre indivíduos criminosos. Ao conhecer os objetos que estão sendo monitorados, pode-se inferir que estão reunidos propositalmente.

Um encontro frequente se torna interessante neste contexto ao cruzar as informações dos presidiários com as viaturas policiais. Com isso seria possível identificar, por exemplo, policiais corruptos, que são aqueles que se encontram com criminosos em locais suspeitos com certa frequência.

Com base nos problemas anteriormente mencionados, o objetivo geral deste trabalho é definir diferentes tipos de comportamento de encontro e elaborar um algoritmo que detecte esses encontros em trajetórias de objetos móveis. Com a implementação de um algoritmo que detecte encontros pretende-se responder a perguntas do tipo: a partir desses dados quais são os elementos que se encontram? Esses objetos se encontram em que momento? Qual a frequência desses encontros? E a partir dessas informações o autor da busca pode elaborar estratégias para verificar os motivos que estão originando esses encontros.

O restante desse trabalho está organizado da seguinte forma: no Capítulo 2 são apresentados os conceitos básicos necessários para o entendimento desse trabalho, bem como os trabalhos relacionados, ou seja, que detectam padrões em trajetórias. No Capítulo 3 são definidos os diferentes tipos de encontro e é apresentado o algoritmo desenvolvido para detecção de encontros. O Capítulo 4 mostra os experimentos realizados para a detecção dos diferentes tipos de encontro e no Capítulo 5 são apresentadas as conclusões do trabalho.

2. CONCEITOS BÁSICOS E TRABALHOS RELACIONADOS

Um padrão em dados de trajetórias é um movimento que se repete várias vezes entre uma trajetória ou entre várias trajetórias [SIQUEIRA e BOGORNY, 2012]. Para facilitar a compreensão do trabalho aqui proposto, este capítulo apresenta conceitos e trabalhos existentes na área de análise de trajetórias de objetos móveis.

2.1 Conceitos Básicos

A seguir serão apresentados os conceitos necessários para o entendimento do trabalho desenvolvido. Serão abordados os assuntos: trajetórias, relacionamentos espaciais e os padrões descritos por trajetórias.

2.1.1 Dados Geográficos e Trajetórias

Um objeto geográfico é qualquer elemento que possua localização com relação à superfície terrestre e geometria bem definidos. Esses objetos são representados em Bancos de Dados Geográficos (BDGs) na forma de ponto, linha ou polígono. Possuem três características principais: atributos não espaciais, que descrevem o objeto de forma qualitativa ou quantitativa; atributos espaciais, que descrevem a localização e a representação do objeto; e os relacionamentos espaciais, que tratam dos relacionamentos de vizinhança entre objetos.

Trajétórias de objetos móveis podem ser armazenadas em um BDG. A Tabela 1 ilustra como esses dados podem ser armazenados. Para cada tupla da tabela existe:

- a) um identificador da trajetória, o *TID*;
- b) um atributo geométrico representado na forma de um ponto (*lat, lon*), a posição(*x,y*);
- c) o tempo em que o objeto estava na posição(*x,y*);

<i>TID</i>	<i>posição (x,y)</i>	<i>Tempo</i>
1	48.890018 2.246100	22/01/2013 08:25
1	48.890018 2.246100	22/01/2013 08:26
...
1	48.890020 2.246102	22/01/2013 08:40
1	48.888880 2.248208	22/01/2013 08:41
1	48.885732 2.255031	22/01/2013 08:42
...
2

Tabela 1 - Exemplo de armazenamento de trajetórias em bancos de dados

Embora a geometria de uma trajetória possa ser representada por uma linha, ela é armazenada como um conjunto de pontos, onde cada ponto é uma tupla da tabela. Dessa forma, cada ponto está associado a um instante de tempo. O conjunto formado pelo objeto geográfico ponto(*x,y*) e pelo tempo constitui um dado espaçotemporal.

Uma trajetória pode ser gerada por diferentes meios de transporte, como por exemplo, o objeto sai de um ônibus e vai a pé até seu destino final. Pode ainda parar em algum lugar durante o trajeto. Logo, uma trajetória pode ter subconjuntos de pontos com características diferentes, e por este motivo muitos dos padrões são detectados em apenas parte de uma trajetória, chamada sub-trajetória. A definição de *subtrajetória* [LEE *et al.*, 2008], apresentada a seguir é utilizada em diversos trabalhos da literatura.

Definição 2. Sub-Trajetória Uma sub-trajetória s de $t = (tid, p_0, \dots, p_n)$ é um conjunto de pontos $(p_i, p_{i+1}, \dots, p_{i+m})$, onde $p_i \in t$ e $i \geq 0$ e $(p_i + m) \leq n$.

Diversos objetos podem gerar trajetórias e essas trajetórias podem ser usadas em diversas aplicações. Ao utilizar trajetórias de barcos pesqueiros, pode-se detectar se a pesca foi realizada dentro da área permitida. Quando os dados são de animais, pode-se observar a migração desses indivíduos entre regiões distintas. O estudo de dados de carros, por exemplo, pode ser usado para análise e melhoria da mobilidade urbana.

A Figura 3 apresenta um conjunto de trajetórias de pedestres coletadas na cidade de Amsterdam [Bogorny e Wachowicz,(2008)].

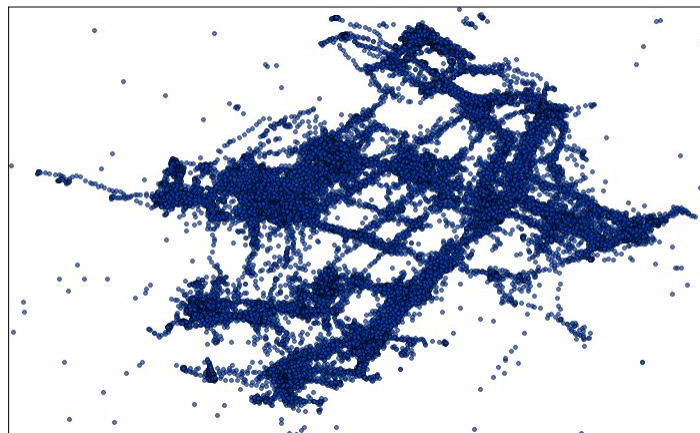


Figura 3 - Conjunto de Trajetórias de Objetos Móveis, visualizada no software QuantumGIS

Como pode ser observado na Figura 3, a visualização desses dados é complicada. Conseqüentemente, análises e detecção de padrões exigem algoritmos complexos para extrair padrões interessantes escondidos nestes dados.

2.1.2 Relacionamentos e operações espaciais

Para a realização do presente trabalho foram utilizadas funções que permitem descrever os relacionamentos espaciais entre objetos geográficos. Esses relacionamentos servem para descrever como os objetos se relacionam. Por exemplo, se duas trajetórias se cruzam, ou se estão próximas [Braz e Bogorny, 2012].

Existem muitas formas de classificar os relacionamentos espaciais. Uma delas é de acordo com a ordem dos elementos a partir de um referencial. Por exemplo, se os dados em questão forem os estados do Brasil, a relação de ordem entre os estados ao redor de Santa Catarina é que o Rio Grande do Sul está abaixo ou ao Sul, e o Paraná está ao Norte ou acima.

Os relacionamentos métricos são aqueles que descrevem o quão distantes dois objetos estão entre si, como a distância e o comprimento. Dependem de definições no sentido de parametrizar o quanto é perto ou longe.

Os relacionamentos topológicos são aqueles em que a posição do objeto é relativa ao espaço onde ele está contido. Descrevem relações entre objetos disjuntos, que se tocam, se sobrepõe, estão contidos ou contém, atravessam, são iguais, entre outros. A Figura 4 apresenta os relacionamentos descritos.

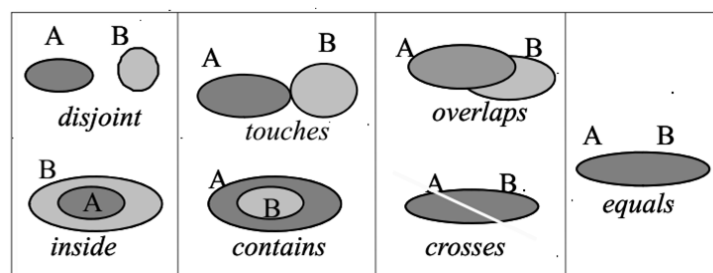


Figura 4 - Relacionamentos Topológicos [Braz e Bogorny, 2012]

Para calcular os relacionamentos, os bancos de dados geográficos utilizam funções específicas. O cálculo dessas funções é muito custoso para o SGBD e o uso de índices espaciais é um importante aliado para melhoria de desempenho.

As operações espaciais são recursos oferecidos por Bancos de Dados Geográficos (BDGs) que permitem fazer a união de objetos, expansão de objetos (*buffer*), calcular a distância, etc. A operação de *buffer* permite expandir um objeto de acordo com uma distância passada como parâmetro. Essa operação é muito importante para análise de trajetórias, visto que os dados oriundos de GPS não são muito precisos e às vezes é necessário aproximar objetos, expandindo-os para encontrar padrões.

Existem ainda operações que tratam dados espaçotemporais. Essas operações são implementadas em Bancos de Dados específicos como Hermes [Theodoridis et. al., 2006] e Secondo [Guting, 1999]. Essas operações, além de tratar do espaço, levam em consideração o tempo. São exemplos de operações *f_buffer*, semelhante ao *buffer*, mas específico para um ponto em um tempo específico e *f_distance*, que calcula a distância entre pontos móveis em um tempo específico. Dessa forma, um exemplo de consulta seria: recuperar todos os pares de indivíduos que estiveram a 5 metros de distância um do outro.

```
select c2.tid, c2.tid
from caminhada c1, caminhada c2
where c1.tid <> c2.tid
and min(distance(c1.rota, c2.rota)) < 5
```

2.2 Trabalhos Relacionados

Nesta seção serão apresentados trabalhos que abordam o tema de descoberta de padrões em trajetórias de objetos móveis. Esses trabalhos foram divididos entre os que reconhecem comportamentos em uma única trajetória e aqueles que tratam do comportamento de grupos de trajetórias.

2.2.1 Comportamentos Individuais

Dentre os algoritmos que identificam padrões em trajetórias, além daqueles que buscam padrões em grupos, existem aqueles que detectam comportamentos de trajetórias individuais. Um desses padrões é o *avoidance* [Alvares et al., 2011], que detecta as trajetórias que evitam um determinado local. Para identificar o desvio é necessário que os elementos de interesse estejam a uma distância mínima da trajetória que o evita. As principais características do padrão comportamental são que a trajetória deve estar seguindo na direção do objeto alvo, e esse movimento da trajetória que aponta para o alvo deve ter um comprimento mínimo antes do desvio acontecer.

O método para calcular o padrão *avoidance* não permite identificar quando uma trajetória desvia intencionalmente ou não. Para minimizar falsos positivos é utilizada uma métrica denominada fator de confiança global que é baseada no somatório das confianças de cada desvio. Quando uma trajetória segue a mesma a direção que ela apresentava antes de desviar do objeto sua confiança é maior. Assim, de acordo com o confiança global é determinado o grau de certeza do desvio.

Outros algoritmos procuram detectar a frequência com que determinados padrões ocorrem, sendo chamados de padrões periódicos ou frequentes. Comportamentos periódicos foram definidos como atividades repetitivas em certos locais com intervalos de tempo regulares [Li, 2010]. Esses comportamentos podem ser utilizados para predição de novos movimentos semelhantes, ou ainda, se o comportamento está fora do normal, dado um histórico. Para detectar esses comportamentos foi desenvolvido o algoritmo *Periodica* [Li, 2010], que detecta padrões de comportamento para indivíduos isoladamente.

Primeiramente os locais de interesse são descobertos, cada período é associado a pelo menos um local de interesse. Na segunda etapa, é criado um modelo estatístico para o comportamento periódico. Em seguida uma técnica de *clustering* é aplicada para determinar o número de comportamentos. Os locais de interesse são essenciais ao algoritmo e são obtidos para cada objeto, pela densidade dos pontos. Quanto mais pontos em um local maior a probabilidade de ser um local de interesse.

O método CB-SMOT, do inglês *clustering-based stops and moves of trajectories*, identifica partes da trajetória em que a velocidade do objeto móvel é menor do que o restante da trajetória. Esse algoritmo permite identificar partes de tráfego lento (e.g. congestionamento de carros). Foi proposto inicialmente por Palma et. al.(2008) e tem como parâmetros: *minTime* (duração mínima de uma subtrajetória com baixa velocidade), *maxAvgSpeed* (valor máximo da média de velocidade de uma trajetória para que possa ser considerado um cluster) e *maxSpeed* (velocidade instantânea máxima aceitável dentro de um cluster).

O CB-SMOT é composto por duas partes principais: primeiro gera clusters de baixa velocidade em cada trajetória; depois é verificado se esses clusters de baixa velocidade se interceptam com objetos geográficos de interesse da aplicação,

que pode ser, por exemplo, um ponto turístico. Essa intersecção deve respeitar um tempo mínimo. Quando os clusters não são interceptados por nenhum objeto conhecido, recebem a nomenclatura *unknown*, seguida por um identificador. Quando esses clusters de baixa velocidade se interceptam recebem o mesmo nome como, por exemplo: *unknown1*.

Diferentemente dos padrões abordados nessa seção, o trabalho aqui proposto trata da interação entre trajetórias. A frequência descrita por Li (2010) permite a detecção de padrões de comportamento para cada trajetória individualmente. A região onde um comportamento é frequente é encontrada no algoritmo e é determinante para a detecção do padrão. Isso não se aplica ao padrão de encontro, pois no algoritmo de Li a interação ocorre entre o objeto e o meio, e no padrão de encontro a interação ocorre entre dois ou mais objetos.

O CB-SMOT, por sua vez, pode ser usado para detectar encontros, uma vez que os *clusters* que se interceptam recebem o mesmo nome. Porém, o algoritmo não foi feito com esse propósito e precisaria de ajustes para detectar encontros espaçotemporais, bem como diferenciar os clusters para cada trajetória, já que pode gerar clusters que se interceptam na mesma trajetória.

2.2.2 Comportamentos de grupos

A partir das trajetórias é possível descobrir comportamentos entre duas ou mais trajetórias de um mesmo conjunto. O algoritmo TRA-CHASE [SIQUEIRA e BOGORNY, 2011] é capaz de detectar o padrão de perseguição entre trajetórias. O autor analisa subtrajetórias, distância mínima e tolerância de tempo. Distância mínima é a menor separação espacial entre os objetos para que possam

caracterizar uma perseguição. Tolerância de tempo é o intervalo de tempo máximo entre duas trajetórias para garantir que elas estão em um momento similar.

Duas trajetórias são candidatas ao padrão de perseguição quando estão em um mesmo intervalo de tempo e de proximidade. Uma sub-perseguição é definida quando uma trajetória está próxima de outra dentro de uma tolerância máxima de tempo. Uma perseguição acontece quando uma dessas sub-perseguições ocorrem por uma duração mínima de tempo.

Uma vez definida a perseguição, esta é classificada em tipos: caçada, assalto, captura e detetive. Nos casos em que a perseguição é do tipo caçada, assalto ou captura é caracterizado um encontro. Em todos os casos tratados como encontro, os objetos além de estarem próximos no espaço e tempo, também precisam caracterizar uma parada, detectada pelo algoritmo CB-SMOT. No presente trabalho não é necessário haver uma parada para caracterizar um encontro, uma vez que as pessoas podem se encontrar em uma caminhada.

Laube (2004), definiu uma série de padrões que consideram os aspectos geográficos e temporais. São eles: *track*, *flock*, *leadership*, *convergence* e *encounter*. Para identificar os três primeiros padrões (*track*, *flock*, *leadership*) o autor definiu a técnica REMO (RELative Motion). O conceito básico do REMO [Laube, 2004] é formar uma matriz cujas linhas representam cada trajetória (ou MPO - *Moving Point Object*) e as colunas descrevem um intervalo de tempo, representados pela letra *t*.

A Figura 5 (a) apresenta um conjunto de objetos, representados pela letra O. A Figura 5 (b) ilustra a matriz REMO utilizada para analisar o movimento dos objetos (representados em cada linha), e para cada instante de tempo (coluna) é apresentado o ângulo em relação ao Norte. A Figura 5 (c) mostra as direções traduzidas em ângulos que variam de 0° a 360°.

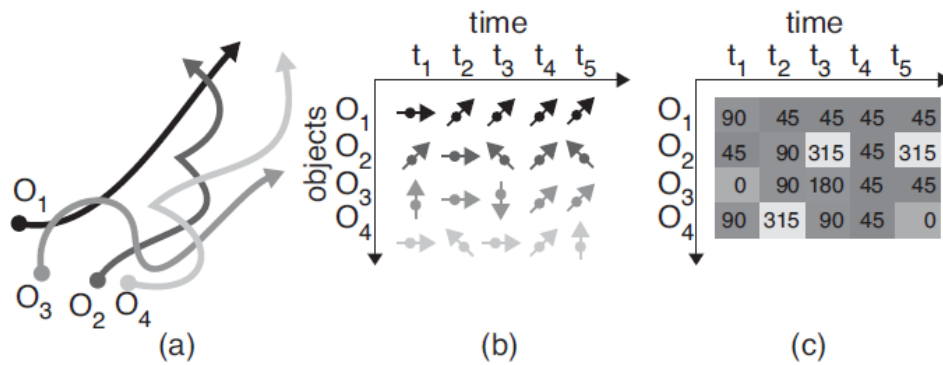


Figura 5 - REMO [Laube, 2004]

A partir da matriz REMO é possível perceber quando o mesmo objeto segue uma determinada direção por um certo período de tempo. Este é o padrão REMO denominado *Track*, que pode ser traduzido como trilha. Este padrão pode descrever, por exemplo, um carro ou um pássaro que se movem na mesma direção. Na Tabela 2, primeira coluna e primeira linha, pode-se observar uma matriz REMO em que o mesmo objeto permanece na mesma direção em quatro instantes de tempo seguidos. Na segunda coluna existem exemplos de trajetórias que descrevem o padrão, seguindo na mesma direção em todos os pontos coletados.

O padrão REMO *flock* é aquele que descreve o comportamento de um grupo ou rebanho, ou seja, em determinado momento todos os objetos dentro de uma área pré-determinada movem-se em uma mesma direção. Esse comportamento pode descrever animais fugindo de uma presa para um local seguro ou um grupo de turistas caminhando juntos para visitar outro lugar. Um exemplo de matriz para esse padrão pode ser observado na Tabela 2, segunda linha, primeira coluna. Nessa matriz, diferentes objetos movem-se na mesma direção em um mesmo instante de tempo. Porém, para que esses objetos configurem um *flock* é necessário que todos estejam em uma mesma área. Na Tabela 2, segunda linha e

segunda coluna, existem dois exemplos de objetos que se movem na mesma direção, em um mesmo instante de tempo, dentro uma região específica.

A última linha do Tabela 2 descreve o padrão Leadership. Como o nome sugere é aquele que propõe a ideia de liderança entre trajetórias. Esse padrão é uma extensão do padrão *flock*, com a condição adicional de que um dos objetos esteja avançado em relação aos outros, ou seja, move-se na frente dos outros objetos. Pode-se observar na Tabela 2, terceira linha e primeira coluna, que a matriz apresenta um objeto que se movimenta durante quatro instantes de tempo para uma mesma direção, e logo, mais dois objetos juntam-se a ele.

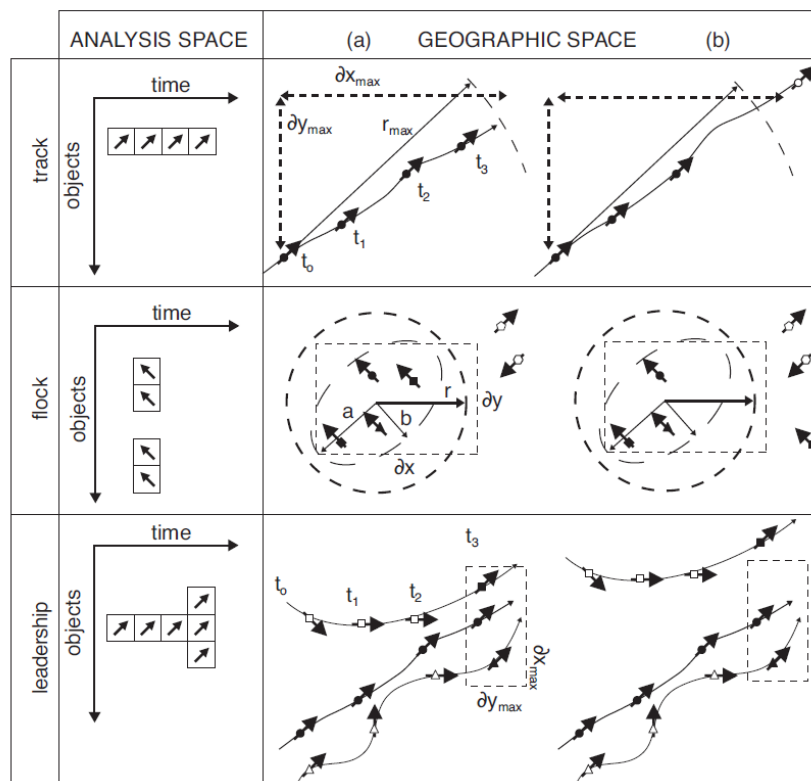


Tabela 2 - Padrões REMO [Laube, 2004]

Os padrões *convergence* e *encounter* não podem ser detectados através da matriz REMO, pois os objetos podem convergir ou se encontrar vindos de direções diferentes. Laube (2004) define o padrão *convergence* como aquele que descreve o comportamento de trajetórias que em um dado momento movem-se

para um mesmo ponto, mas que depois podem seguir outros caminhos, como ilustrado na Figura 6. Este padrão é similar ao encontro.

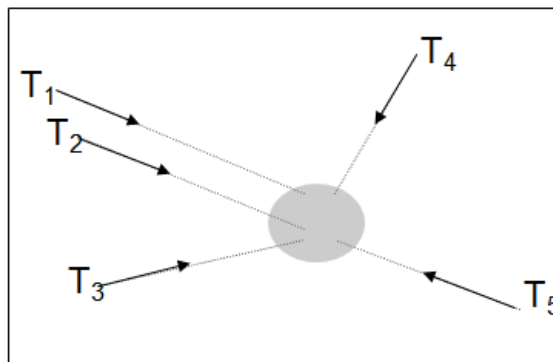


Figura 6 - Padrão Convergence

Para esse padrão o autor propõe um algoritmo que considera intervalos de tempo para que todos os objetos envolvidos na convergência, movam-se para a área determinada ao mesmo tempo. Cada trajetória é transformada em um vetor, ou semirreta direcionada, que são as retas que descrevem como as trajetórias chegariam ao centro, caso permanecessem naquela direção. Para determinar a convergência Laube (2004) divide a área de interesse (usando um diagrama de *Voronoi*) e acrescenta um *buffer* em cada reta. Caso mais de uma reta passe por uma mesma área é detectado um padrão de convergência.

Laube (2004) define o *encontro* quando as trajetórias se intersectam dentro de um raio definido em um tempo similar. Nesse caso, as trajetórias da Figura 6 precisariam estar dentro do círculo central para caracterizar um encontro. Além da localização dos elementos dentro do raio, o tempo em que esses elementos atravessam o mesmo local também é considerado.

O método para detecção de encontro proposto por Laube(2004) detecta apenas os casos em que o encontro ocorre no mesmo lugar ao e mesmo tempo. Além disso, o autor não se preocupa com os casos em que o encontro é frequente ou somente espacial.

3. DESCOBERTA DE PADRÕES DE ENCONTRO EM TRAJETÓRIAS DE OBJETOS MÓVEIS

Este capítulo descreve os três tipos de encontro definidos neste trabalho e o algoritmo que detecta esses encontros.

3.1 Encontro Espacial, Espaço-temporal e Frequente

Um encontro puramente espacial acontece quando duas ou mais trajetórias se aproximam a uma distância máxima, mas seus tempos não estão sincronizados. Analisando indivíduos suspeitos, esses *encontros* podem caracterizar uma área de interesse para vários criminosos, que visitam o mesmo lugar em horários diferentes. Essa área de interesse pode ser, por exemplo, um local que está sendo vigiado por criminosos para a prática de um delito.

Definição 3. Encontro Espacial. Sejam $s_1 = \langle p_0, \dots, p_n \rangle$ e $s_2 = \langle q_0, \dots, q_n \rangle$ duas subtrajetórias de t_1 e t_2 , respectivamente, e $maxDistance$ uma distância máxima. s_1 e s_2 apresentam um encontro espacial se e somente se $\forall p_i \in S_1 \mid distance(p_i, S_2) \leq maxDistance \wedge \forall q_i \in S_2 \mid distance(q_i, S_1) \leq maxDistance$.

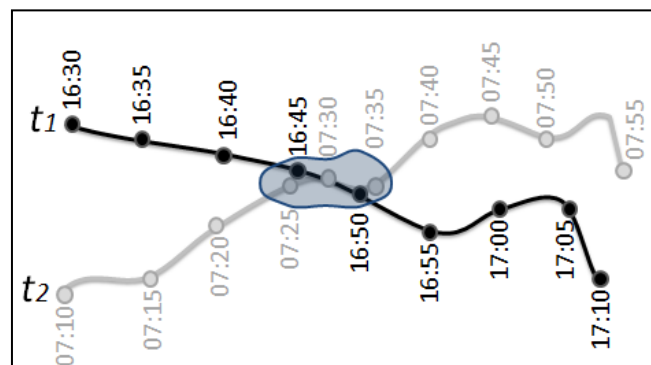


Figura 7 - Encontro Espacial

A Figura 7 apresenta um encontro puramente espacial, onde t_1 é a trajetória em preto e t_2 é a trajetória em cinza. É possível observar que as trajetórias se encontram no espaço, mas o tempo em que elas estavam na área de encontro é diferente. A trajetória t_1 passou pela área de encontro à tarde, por volta das 16:45, enquanto t_2 passou pela manhã, em torno das 07:30.

Todo encontro possui uma *área*, que é a porção do espaço que envolve a geometria das trajetórias que estão próximas. No presente trabalho não são utilizadas áreas pré-definidas, pois um dos objetivos é detectar a área de encontro. Para identificar a área do encontro são utilizadas duas funções bem conhecidas em BDGs: *makeLine* e *buffer*. A primeira função transforma a trajetória em uma linha e a segunda expande essa linha. Dessa forma, cada trajetória é transformada em um polígono e a área do encontro é dada pela geometria resultante da intersecção entre as trajetórias *bufferizadas*.

Na Figura 8, a área a , de cor mais escura, representa a intersecção entre as trajetórias t_1 e t_2 . A mesma área do encontro é utilizada para os encontros espaçotemporais. Nas próximas figuras são utilizadas as trajetórias originais, sem *buffer*, para facilitar o entendimento dos conceitos.

Definição 4. Área de Encontro. Dadas duas trajetórias t_1 e t_2 que se encontram espacialmente a uma distância máxima *maxDistance*, a área do encontro é dada por $a = \text{buffer}(\text{makeLine}(t_1), \text{maxDistance}) \cap \text{buffer}(\text{makeLine}(t_2), \text{maxDistance})$.

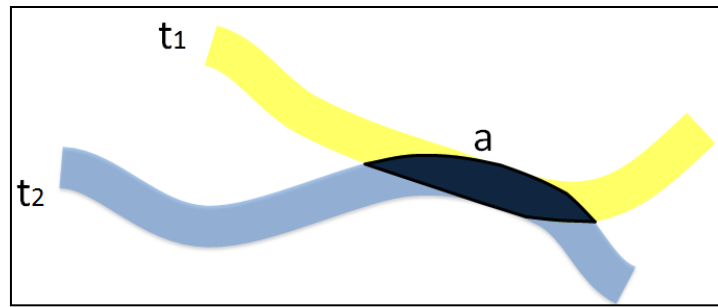


Figura 8 - Área de Encontro

Um encontro espaçotemporal é aquele em que duas ou mais trajetórias estão a uma distância mínima entre si, como no encontro espacial, mas também possuem tempo similar.

Definição 5. Encontro Espaçotemporal Seja $s_1 = \langle p_0, \dots, p_n \rangle$ e $s_2 = \langle q_0, \dots, q_m \rangle$ subtrajetórias de t_1 e t_2 , respectivamente. s_1 e s_2 caracterizam um encontro espaçotemporal se e somente se s_1 e s_2 possuem um encontro espacial e $[p_0.t, p_n.t] \cap [q_0.t, q_m.t] > \text{minTime}$.

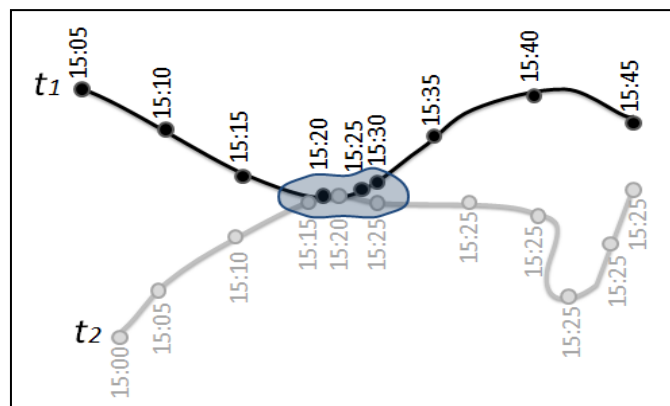


Figura 9 - Encontro espaçotemporal

Na Figura 9 é possível observar que além do encontro espacial, o tempo também é sincronizado. De fato, ele não precisa ser igual, mas deve respeitar uma tolerância mínima. Caso o encontro espaçotemporal ocorra entre mais de dois indivíduos, o tempo das trajetórias de todos os objetos deve estar sincronizado.

Os encontros espaciais e espaçotemporais podem ocorrer de forma evolutiva. Por exemplo, duas pessoas podem se encontrar em um determinado horário e algum tempo depois chegar uma terceira pessoa. Nesse caso são caracterizados dois encontros diferentes. Um exemplo disso pode ser observado na Figura 10. Esta figura (esquerda) apresenta três trajetórias, duas delas, t_1 e t_2 , se encontram por volta das 11:40 e chegam juntas ao encontro das 12:10, entre t_1 , t_2 e t_3 . Neste exemplo o encontro entre t_1 e t_2 (Figura 10 - centro) é diferente, pois tem duração e geometria diferentes do encontro e_2 entre t_1 , t_2 e t_3 (Figura 10 - direita).

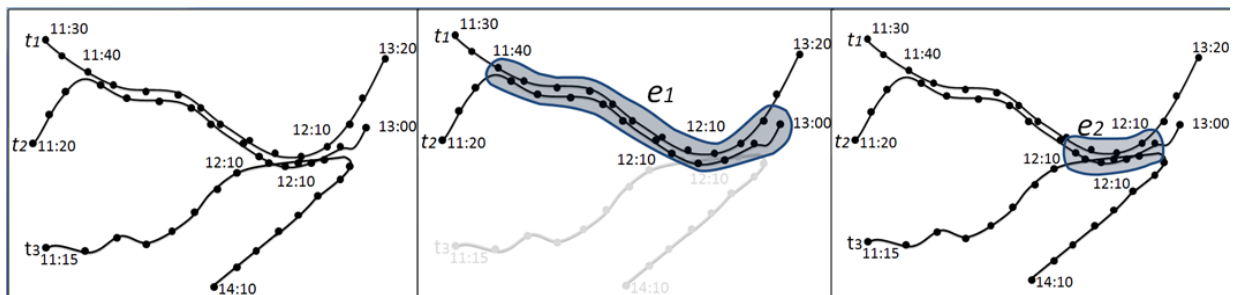


Figura 10 - Trajetórias T1, T2 e T3 e seus encontros

Encontros espaçotemporais podem ocorrer repetidamente entre os mesmos objetos. Quando isso ocorre, têm-se *encontros frequentes*. Os encontros frequentes podem ocorrer sempre no mesmo lugar ou em lugares diferentes. Considerando dois como o número mínimo de encontros para caracterizar um encontro frequente, a Figura 11 ilustra um exemplo onde duas trajetórias caracterizam um encontro frequente em lugares diferentes.

Definição 6. Encontro Frequente Seja E um conjunto de encontros espaçotemporais entre o mesmo conjunto de trajetórias e $minSup$ o número mínimo

de encontros para caracterizar um conjunto de encontros como frequente. E é um conjunto de encontros frequentes se e somente se $|E| \geq \text{minSup}$.

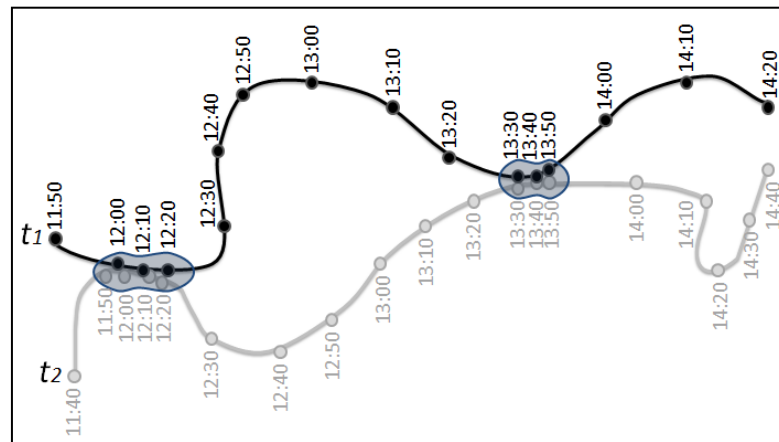


Figura 11 - Encontro Frequente

3.2 Algoritmo para detecção de encontros

O algoritmo elaborado neste trabalho tem por objetivo descobrir os *encontros* a partir de um conjunto de trajetórias, seja no mesmo local, mesmo local e período e com determinada frequência.

Nesta seção é apresentado o algoritmo geral (Algoritmo 1) que identifica os encontros (*trajectoryEncounter*) e dois algoritmos auxiliares que permitem verificar as diferentes características dos encontros. São eles:

1. *trajectoryEncounter*: é o algoritmo geral que identifica os encontros e controla as iterações. Este algoritmo também é responsável pelas chamadas aos algoritmos que auxiliam na descoberta dos encontros.
2. *encounterTimeVerification*: algoritmo que realiza a verificação do tempo dos candidatos a encontro.

3. *encounterMerge*: algoritmo utilizada para agrupar encontros semelhantes, sejam espaciais ou espaçotemporais. Este algoritmo determina quais candidatos são encontros.

A saída esperada do algoritmo é ilustrada nas tabelas 3, 4 e 5. A Tabela 3, apresenta os *TIDs* das trajetórias que visitaram a mesma área (*Áreas Visitadas*) em tempos diferentes.

TIDs	Áreas Visitadas:
3, 5, 9	POLYGON(53 39, 12 21, 13 22, 32 12, 53 39)
2, 5	POLYGON(55 56, 49 22, 39 34, 22 33, 55 56)
1, 8, 9, 10	POLYGON(12 33, 45 56, 34 56, 23 43, 12 33)

Tabela 3 - Encontros Espaciais

A Tabela 4 apresenta os encontros espaçotemporais. Esta tabela possui como informações as trajetórias envolvidas no encontro (coluna *TID*), as coordenadas do centroide da área de encontro (coluna *Área de Encontro(lon,lat)*), a data e hora do início do encontro (coluna *Início*) e do fim do encontro (coluna *Fim*).

TIDs	Área de encontro (lon, lat)	Início	Fim
99, 76, 75	(53.8532,39.3454)	01-Mar-2013/15:04	01-Mar-2013/15:30
89, 76	(26.8245,36.7457)	01-Mar-2011/14:20	01-Mar-2011/14:55

Tabela 4 - Encontros Espaçotemporais

A Tabela 5 representa os conjuntos de encontros frequentes. Esta tabela possui o conjunto de identificadores das trajetórias (coluna *TID*) que possuem encontros frequentes entre si, acompanhado da quantidade de vezes em que esses objetos se encontraram (coluna *Frequência*).

TIDs	Frequência
09589, 09976	3 vezes
09689, 09976, 09822	6 vezes

Tabela 5 - Encontros Frequentes

A função básica do algoritmo geral é verificar se existem encontros espaciais, espaçotemporais e frequentes dentro de um determinado conjunto de trajetórias. Para isso é verificado inicialmente se existem encontros dois a dois, ou seja, cada trajetória é comparada com todas as outras do conjunto. A cada iteração são gerados *candidatos a encontros*. Esses candidatos poderão se tornar encontros após serem verificados no algoritmo *encounterMerge* (Algoritmo 3). Esse algoritmo é chamado ao final de cada iteração.

Após a primeira verificação, têm-se um conjunto de encontros dois a dois. Estes encontros são então comparados às outras trajetórias do conjunto para a verificação dos encontros três a três. Ou seja, cada encontro é comparado com todas as trajetórias, gerando encontros entre três indivíduos. Essas comparações são chamadas *iterações*. Cada nova geração de encontros é feita a partir da *iteração* anterior e são repetidas até que não exista um novo encontro.

Para descobrir os encontros espaciais, espaçotemporais e frequentes foi proposto o Algoritmo 1 – *trajectoryEncounter*.

Algoritmo 1 – *trajectoryEncounter*

```

(1) Input:
(2) T: set of trajectories
(3) minTime: minimal time
(4) maxDistance: maximal distance
(5) minSup: minimal frequency
(6)
(7) Output:
(8) spatialEncounters, spatiotemporalEncounters, frequentEncounters
(9)
(10) Method:
(11) idIteration = 1 //first iteration, find two by two encounters
(12) TBuffer = bufferizeAll(makeLineAll(T), maxDistance) //transforms all trajs. in
                                     lines, then adds a buffer of maxDistance
(13) FOR EACH (t1 in TBuffer)//find pair-wise encounters
(14)   FOR EACH (t2 in TBuffer)
(15)     IF(t1.tid ≠ t2.tid)
(16)       IF (overlaps(t1.the_geom, t2.the_geom))//create a spatial candidate
(17)         candidateS.geom = intersection(t1.the_geom,t2.the_geom)
(18)         candidateS.tids.add(t1.tid)
(19)         candidateS.tids.add(t2.tid)
(20)         spatialCandidatesSet[1].add(candidateS)

```

```

(21)     END IF
(22)     END IF
(23)   END FOR
(24) END FOR
(25)
(26) spatialEncounters[1] = encounterMerge(spatialCandidatesSet[1], 0)
(27) spatioTempCandidatesSet[1] = encounterTimeVerification(minTime,
                                                             spatialCandidatesSet[1], T)
(28) spatiotemporalEncounters[1] = encounterMerge(spatioTempCandidatesSet[1], minTime)
(29)
(30) DO //loop to compare each encounter with each trajectory
(31)   idIteration = idIteration + 1;
(32)
(33)   FOR EACH (e in spatialEncounters[IdIteration-1])
(34)     FOR EACH (t in TBuffer)
(35)       IF(t.tid not in e.tids)
(36)         IF(overlaps(t.geom, e.geom)) //if the trajectory overlaps other encounter
(37)           candidateS.geom = intersection(t.the_geom, e.the_geom)
(38)           candidateS.tids.add(t.tid)
(39)           candidateS.tids.addAll(e.tids)
(40)           spatialCandidatesSet[IdIteration].add(candidateS)
(41)         END IF
(42)       END IF
(43)     END FOR
(44)   END FOR
(45)
(46)   spatialEncounters[idIteration] = encounterMerge(
                                                             spatialCandidatesSet[idIteration],0)
(47)   spatioTempCandidatesSet[idIteration] = encounterTimeVerification( minTime,
                                                             spatialCandidatesSet[idIteration],T)
(48)   spatiotemporalEncounters[idIteration] = encounterMerge(
                                                             spatioTempCandidatesSet[idIteration], minTime)
(49) WHILE spatiotemporalEncounters[idIteration].size() > 0
(50)
(51) frequentEncounters = apriori(spatiotemporalEncounters, minSup)
(52)
(53) RETURN spatialEncounters, spatiotemporalEncounters, frequentEncounters

```

O algoritmo possui como entradas: um conjunto de trajetórias T (linha 2); o tempo mínimo $minTime$, em minutos, que os indivíduos devem permanecer juntos para descrever um encontro espaçotemporal (linha 3); uma distância máxima $maxDistance$, em metros (linha 4); e o $minSup$ (linha 5). Os indivíduos não precisam estar sob a mesma coordenada para se encontrarem, mas devem estar a uma distância controlada. A saída (linha 8) é o conjunto de encontros espaciais, espaçotemporais e frequentes.

Primeiramente inicializa-se a variável $idIteration$ (linha 11). Essa variável é necessária para o controle das *iterações*. Em seguida é chamada a função *bufferizeAll* (linha 12) que transforma cada trajetória, que é inicialmente um conjunto

de pontos (tid, x, y, t), em um polígono. Para isso, primeiramente, cada trajetória é transformada em uma linha, através da função *makeLine*. Depois é adicionado um *buffer* de tamanho *maxDistance*, em metros. Essa transformação é feita para que seja possível verificar quais trajetórias estavam próximas umas das outras e que poderiam ter um encontro. Se fossem somente linhas, as trajetórias poderiam nunca atravessar o caminho uma da outra, permanecendo lado a lado durante todo o encontro, sem configurar uma intersecção. Quando as trajetórias estão a essa distância elas formam um *candidato* a encontro.

Com todas as trajetórias transformadas em polígonos, são então descobertos os candidatos a encontro espacial dois a dois (linhas 13 a 21). Para cada par de trajetórias (*bufferizadas*) diferentes (linhas 13 a 15), é verificado se existe alguma ligação espacial através da função *overlaps* (linha 16). Caso exista, é criado um novo candidato a encontro, e a *área do encontro* é calculada através da função *intersection*, que retorna a geometria da área resultante da intersecção entre os dois polígonos originais (linha 17). Em seguida são adicionados os *TIDs* das trajetórias envolvidas na lista de trajetórias do novo candidato (linhas 18 e 19) e o mesmo é adicionado à lista de candidatos a encontros espaciais (linha 20).

Na sequência, são verificados e agrupados os candidatos semelhantes, que são aqueles que ocorrem em áreas parecidas, através do algoritmo *encounterMerge* (linha 26). Logo é realizada a verificação temporal dos candidatos a encontro através do algoritmo *encounterTimeVerification* (linha 27). Nesse caso, utilizam-se os candidatos, pois os encontros em tempos diferentes podem ter sido agrupados. Em seguida, os candidatos a encontro espaçotemporal são agrupados quando possuem área e tempo similares (linha 28).

Uma vez finalizada a análise dos encontros dois a dois é iniciado o loop (linha 30) incrementando o identificador da iteração (linha 31). Em seguida os

encontros espaciais da iteração atual são identificados através da comparação das trajetórias com os encontros da iteração anterior (linhas 33 a 44). Em outras palavras, todo esse trecho do algoritmo analisa encontros três a três, quatro a quatro e assim sucessivamente.

Para cada encontro (linha 33), e para cada trajetória (linha 34), é verificado se esta trajetória já faz parte do encontro (linha 35). Essa verificação é feita para que apenas sejam gerados novos candidatos para trajetórias diferentes. Caso a trajetória atual não pertença ao encontro sendo testado é verificado se as geometrias da trajetória e do encontro se interceptam (linha 36), através da função *overlaps*. Se sim, um novo candidato a encontro espacial é criado (linha 37). Os *TIDs* das trajetórias que foram interceptadas e do encontro original são adicionados a lista de *TIDs* do novo candidato a encontro (linhas 38 e 39).

Ao final do reconhecimento dos candidatos a encontros espaciais de cada iteração é feito o agrupamento através do algoritmo *encounterMerge* para verificar se o candidato a encontro é único ou deve ser agrupado com outros, formando um encontro (linha 46). Em seguida o tempo dos candidatos é verificado através do algoritmo *encounterTimeVerification* (linha 47). E os encontros espaçotemporais são identificados através do agrupamento de seus candidatos (linha 48).

Antes do algoritmo *encounterMerge* não se pode dizer que já foram detectados os encontros, pois podem existir vários candidatos que representam um único encontro. Somente depois da unificação desses candidatos, ou de verificar que não há nenhum outro candidato que represente um mesmo encontro pode-se dizer que o padrão foi detectado. A verificação da continuidade do loop é feita em seguida (linha 49) e caso não exista nenhum novo encontro, ou seja, se o total de encontros espaçotemporais da iteração atual for igual a zero o laço é encerrado.

Caso contrário, o processo é repetido novamente para que seja possível detectar os encontros três a três, quatro a quatro e assim sucessivamente.

Assim que todos os encontros foram identificados é possível verificar quais deles são frequentes. Essa verificação é realizada através do método *Apriori* e retorna o conjunto de trajetórias que possuem ao menos *minSup* encontros (linha 51). No método *Apriori* foram utilizados os encontros espaçotemporais nas linhas da matriz e as trajetórias nas colunas. Nos casos em que a trajetória pertencia ao encontro, o valor da matriz foi preenchido com 1, e quando a trajetória não fazia parte do encontro foi preenchido com 0. Ao final são retornados os encontros espaciais, espaçotemporais e frequentes (linha 53).

Após identificar os encontros espaciais de cada iteração é verificada a condição temporal. Para isso utiliza-se o algoritmo *encounterTimeVerification*, ilustrada no Algoritmo 2. Esse algoritmo verifica se as trajetórias envolvidas no encontro estavam na mesma área e dentro da mesma janela de tempo. Em linhas gerais, esse algoritmo cria subtrajetórias das trajetórias envolvidas no encontro, para então comparar os tempos de entrada e saída de cada objeto envolvido, conforme detalhado no Algoritmo 2.

O algoritmo 2 tem como entradas o conjunto de trajetórias T (linha 2), o conjunto candidatos a encontro C (linha 3), e o tempo mínimo de duração do encontro *minTime* (linha 4). A saída um conjunto de candidatos a encontros espaçotemporais.

Para cada candidato a encontro (linha 10), é verificado se a trajetória t faz parte do candidato a encontro (linhas 11 e 12). Em caso positivo, é criada uma subtrajetória com todos os pontos de t que interceptam a área do candidato a encontro (linha 13).

Algoritmo 2 - *encounterTimeVerification*

```

(1) Input:
(2) T: set of trajectories
(3) spatialCandidatesSet: a set of encounters candidates
(4) minTime: minimum time
(5)
(6) Output:
(7) spatioTempCandidatesSet: a set of candidates of spatiotemporal encounters
(8)
(9) Method:
(10) FOR EACH (spatialCandidate in spatialCandidatesSet)
(11)   FOR EACH (t in T)
(12)     IF (t belongs spatialCandidate) //if the trajectory originated the candidate
(13)       s = makeSubtrajectory(t, spatialCandidate.the_geom) //makes a subtrajectory
           with all points within the geometry of the candidate
(14)       IF(s.numberOfPoints >= 2)
(15)         S.add(s)
(16)       END IF
(17)     END IF
(18)   END FOR
(19) IF(S.size >= 2) //if there are at least 2 trajectories
(20)   entrance = getLastEntrance(S)
(21)   exit      = getFirstExit(S)
(22)   IF (entrance < exit) //if the encounter is spatiotemporal
(23)     IF ((exit - entrance) >= minTime) //if the duration of the encounter lasted
           //at least minTime
(24)       spatioTempCandidate.the_geom = spatialCandidate.the_geom
(25)       spatioTempCandidate.tids     = spatialCandidate.tids
(24)       spatioTempCandidate.beginTime = entrance
(25)       spatioTempCandidate.endTime   = exit
(26)       spatioTempCandidatesSet.add(spatioTempCandidate);
(27)     END IF
(28)   END IF
(29) END IF
(30) END FOR
(31)
(32) RETURN spatioTempCandidatesSet

```

Para determinar um encontro, foi considerado que o mesmo deve envolver ao menos dois pontos de cada trajetória. Uma vez que dados de GPS são muito imprecisos, um encontro com somente um ponto de cada trajetória poderia caracterizar um ruído. Além disso, dois pontos são o mínimo para que se possa calcular o intervalo de tempo que cada trajetória permaneceu na área do encontro (linha 14).

Em seguida, os encontros passam por uma verificação de sincronia no tempo, para que possam ser detectados os encontros espaçotemporais. Para isso armazena-se na variável *entrance* (linha 20) o tempo da última trajetória a chegar ao

encontro. Armazena-se na variável *exit* (linha 21), o tempo da primeira trajetória que saiu do encontro. Se o valor de *exit* for maior que o valor de *entrance*, caracteriza-se um candidato a encontro espaçotemporal (linha 22). Em seguida é verificado se o tempo em que os candidatos permaneceram juntos satisfaz a condição do tempo mínimo (linha 34) para caracterizar um encontro espaçotemporal.

Caso o encontro seja espaçotemporal, são armazenados os tempos inicial e final do encontro (linhas 35 e 36). Para entender melhor o princípio utilizado para a definição dos encontros temporais são utilizadas as tabelas 6 e 7.

TID	entrance	exit
1	8:02	12:05
2	10:50	12:08
3	9:20	13:00

Tabela 6 – Tempos em um encontro espaçotemporal

TID	entrance	exit
1	8:02	10:00
2	10:50	12:08
3	12:40	13:00

Tabela 7- Tempos em um encontro espacial

Na tabela 6 temos um encontro espaçotemporal entre os objetos 1, 2 e 3. Nesse caso o último a chegar entrou na área de encontro às 10:50, segunda linha da Tabela 6. O primeiro a sair, deixou a área do encontro às 12:05, primeira linha da Tabela 6. Logo, os objetos envolvidos no encontro permaneceram juntos das 10:50 às 12:05. Considerando-se como tempo do encontro 1 hora, tem-se um encontro espaçotemporal.

Já para o caso da Tabela 7, pode-se observar que o encontro é somente espacial, pois os envolvidos entram e saem do local do encontro em horários não sincronizados. Nesse caso o último a chegar, entrou na área de encontro às 12:40, terceira linha da Tabela 7, e primeiro a sair, deixou a área às 10:00, primeira linha

da tabela 7. Como esses tempos não estão sincronizados, esse não é um encontro espaçotemporal.

Um encontro entre três trajetórias pode gerar diversos candidatos a encontro, como pode ser observado na Figura 12.

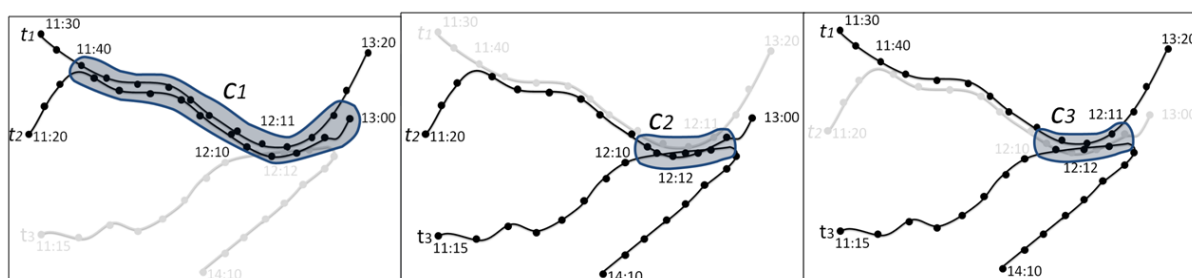


Figura 12 – Candidatos a encontro entre t_1 , t_2 e t_3

A Figura 12 apresenta os candidatos a encontro entre as trajetórias t_1 , t_2 e t_3 . Na Figura 12 - esquerda, o encontro entre as trajetórias t_1 e t_2 . No centro, o encontro entre t_2 e t_3 e na direita entre t_1 e t_3 . Ao observar os quadros do centro e da direita é possível perceber que eles possuem área e tempo semelhantes. Nesses casos, essas áreas devem ser unificadas. Para agrupar candidatos com área similar ou tempo similar utiliza-se o algoritmo *encounterMerge* ilustrada no Algoritmo 3.

O algoritmo tem como entradas um conjunto de candidatos *candidatesSet* (linha 2) e o tempo mínimo *minTime* (linha 3). A saída é o um conjunto de encontros E (linha 6). O algoritmo inicia copiando o conjunto de encontros para uma variável auxiliar para poder realizar a verificação entre dois candidatos do mesmo grupo (linha 9). Para cada candidato, verifica-se os outros elementos do conjunto, excluindo ele mesmo (linhas 10 a 12).

Logo, calcula-se a área de intersecção entre a geometria dos candidatos (linha 13). Para que os candidatos sejam agrupados a área de intersecção das geometrias deve corresponder a pelo menos 75% (linha 14). O valor de 75% da

área foi obtido a partir de experimentos com dados reais de encontros conhecidos. Ao diminuir este tamanho serão agrupados encontros com uma pequena sobreposição. Um valor mais alto exige uma sobreposição quase total das áreas de encontro, o que não é realístico em dados de GPS.

Em seguida são preenchidos os dados do encontro. A geometria é a união das áreas do encontro dos candidatos (linha 15). E as trajetórias envolvidas são todas aquelas que pertencem a pelo menos um dos candidatos que estão sendo agrupados (linhas 16 e 17). Logo é feita a verificação temporal, se o parâmetro *minTime* é igual a zero (linha 18), significa que os candidatos são de encontros espaciais. Nesse caso o encontro é adicionado à lista de encontros e os candidatos que o formaram são excluídos (linhas 29 a 31).

No caso dos candidatos a encontros espaçotemporais (*minTime* maior que zero) deve-se verificar se a duração do possível encontro corresponde ao mínimo esperado. Subtrai-se a data final do encontro que terminou mais cedo da data inicial do encontro que começou mais tarde (linha 21). Caso essa diferença seja maior que *minTime*, os encontros estão sincronizados no tempo. Assim, o tempo inicial do encontro é igual à data inicial do encontro que começou mais tarde (linha 22). E o tempo final do encontro é igual à data do encontro que terminou mais cedo (linha 23). O novo encontro é então adicionado à lista de encontros (linha 24) e os candidatos agrupados são excluídos (linhas 25 e 26).

Algoritmo 3 – *encounterMerge*

```

(1) Input:
(2) candidatesSet: a set of candidates
(3) minTime: minimum time
(4)
(5) Output:
(6) E: a set of encounters
(7)
(8) Method:
(9) auxCandidatesSet = candidatesSet
(10) FOR EACH (candidate in candidatesSet)
(11)   FOR EACH (candidateAux in auxCandidatesSet)
(12)     IF (candidate.id ≠ candidateAux.id)
(13)       totalArea = area(intersection(candidate.geom, candidateAux.geom))
(14)       IF (totalArea > 0 AND (totalArea ≥ (area(candidate.geom) * 0.75)
                OR totalArea ≥ (area(candidateAux.geom) * 0.75) ) )
(15)         encounter.geom = union(candidateAux.geom, candidate.geom)
(16)         encounter.tids.addAll(candidate.tids)
(17)         encounter.tids.addAll(candidateAux.tids)
(18)         IF (minTime > 0) //time should be verified
(19)           lastEntrance=getLastEntrance(candidate.beginTime,
                candidateAux.beginTime)
(20)           firstExit    = getFirstExit(candidate.endTime, candidateAux.endTime)
(21)           IF ((firstExit - lastEntrance) > minTime)//spatiotemporal encounter
(22)             encounter.beginTime = lastEntrance
(23)             encounter.endTime   = firstExit
(24)             E.add(encounter)
(25)             candidatesSet.remove(candidate.id)
(26)             candidatesSet.remove(candidateAux.id)
(27)           END IF
(28)         ELSE //is a spatial encounter
(29)           E.add(encounter)
(30)           candidatesSet.remove(candidate.id)
(31)           candidatesSet.remove(candidateAux.id)
(32)         END IF
(33)       END IF
(34)     END IF
(35)   END FOR
(36) END FOR
(37) E.addAll(candidatesSet) //add all Candidates where was not merged
(38) RETURN E

```

Depois que os todos os encontros similares foram agrupados, ainda podem existir candidatos que não se pareciam com nenhum outro. A Figura 12 ilustra um exemplo, onde os candidatos *c2* e *c3* (centro e direita) são agrupados, mas o candidato *c1* não possui outro candidato semelhante. Esses encontros são então incorporados à lista de encontros (linha 37).

Neste capítulo foram apresentados os algoritmos para detectar os encontros espaciais, espaçotemporais e frequentes. A seguir serão apresentados os experimentos realizados a partir desses algoritmos.

4. EXPERIMENTOS

Para validação do método foram realizados experimentos com dois conjuntos de dados distintos. O primeiro conjunto de dados foi gerado por um grupo de pessoas andando com diferentes aparelhos GPS, na UFSC, Campus Florianópolis - Brasil. O Segundo é referente a uma caça ao tesouro realizada na cidade de Amsterdam – Holanda [Bogorny e Wachowicz,(2008)]. Os conjuntos de dados são de pedestres, pois o algoritmo foi proposto para detectar encontros entre pessoas, não entre carros ou ônibus. Como o uso de SGBDs espaçotemporais ainda é muito restrito, já que a maioria são protótipos acadêmicos, foi utilizado nesse trabalho um BDG, bem como as funções existentes no mesmo. O SGBD escolhido foi o PostgreSQL 9.1, com a extensão PostGIS 2.0.

4.1 Conjunto de dados - UFSC

Os dados foram coletados no dia 6 de Setembro de 2013 por um grupo de estudantes da UFSC. O grupo caminhou pela Universidade, simulando encontros. Esse conjunto de dados possui 5 trajetórias, 3915 pontos e um total de 6 encontros espaçotemporais e 3 encontros espaciais. Ocorreu também um encontro frequente, que entre os 5 indivíduos aconteceu duas vezes. As trajetórias desse grupo, bem como os encontros espaçotemporais estão ilustrados na Figura 13.

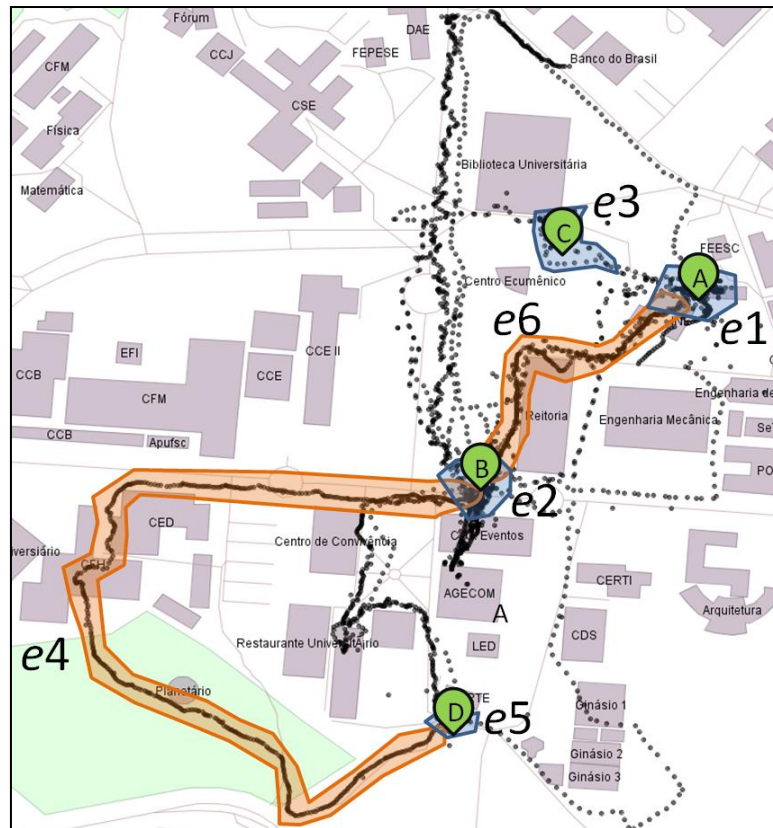


Figura 13 - Trajetórias UFSC

Na Figura 13 os prédios onde os encontros ocorreram estão representados por balões. As áreas esperadas para cada encontro estão representadas em azul e laranja. Na mesma figura são apresentados os seguintes encontros (ordenados cronologicamente):

- e1. Encontro entre 4 indivíduos em frente ao Prédio A;
- e2. Encontro entre 5 indivíduos em frente ao Prédio B;
- e3. Encontro entre 2 indivíduos em frente ao Prédio C;
- e4. Encontro entre 2 indivíduos caminhando juntos do Prédio B ao Prédio D;
- e5. Encontro entre 3 indivíduos em frente ao Prédio D;
- e6. Encontro entre 5 indivíduos caminhando juntos do Prédio B até o Prédio A.

4.1.1 Resultados

A partir desse conjunto de dados o algoritmo foi executado com valores de parâmetros diferentes. A Figura 14 e a Tabela 8 destacam os principais encontros espaçotemporais detectados com os parâmetros: $minTime = 5min$ e distância máxima $maxDistance = 5m$. Através desses parâmetros, todos os encontros gerados foram detectados corretamente.



Figura 14 - Principais encontros espaçotemporais UFSC

<i>idEncontro</i>	<i>TIDs</i>	<i>beginTime</i>	<i>endTime</i>
e1	1,2,3,5	2013-09-06 16:40:19	2013-09-06 16:45:35
e2	1,2,3,4,5	2013-09-06 17:23:25	2013-09-06 17:28:23
e3	1,3	2013-09-06 17:09:40	2013-09-06 17:15:36
e4	4,5	2013-09-06 17:17:06	2013-09-06 17:28:06
e5	2,4,5	2013-09-06 17:09:19	2013-09-06 17:17:01
e6	1,2,3,4,5	2013-09-06 17:21:43	2013-09-06 17:32:18

Tabela 8 - Valores dos principais encontros espaçotemporais UFSC

A Figura 14 apresenta os encontros 1, 2, 3, 4, 5 e 6. As áreas de cada encontro foram coloridas de azul e laranja para destacar encontros parcialmente sobrepostos. O primeiro encontro (e1), Figura 14 (canto superior direito), ocorreu na

reunião inicial entre todos os integrantes na frente do prédio A e corresponde a primeira linha da Tabela 9. Neste momento um dos aparelhos GPS não estava funcionando, logo, esse encontro ocorreu entre as outras quatro trajetórias, entre 16:40h e 16:45h. O segundo encontro (e2), segunda linha da Tabela 9, ocorreu entre todos os envolvidos, na frente do prédio B, das 17:23 às 17:28. O encontro 3 (e3) na Figura corresponde a terceira linha da Tabela 9. Esse encontro ocorreu entre as trajetórias 1 e 3 nas proximidades do Prédio C, e durou pouco mais de 5 minutos.

O encontro 4 (e4) foi entre as trajetórias 4 e 5, e corresponde a quarta linha da Tabela 9. Esse encontro começou às 17:17h próximo ao prédio D, onde os objetos caminharam juntos até as proximidades do prédio B, finalizando o encontro às 17:28h. O quinto encontro (e5) ocorreu entre as trajetórias 2, 4 e 5, nas proximidades do prédio D e durou cerca de oito minutos. O encontro 6 (e6), ilustrado na Figura 14, é a volta de todos os integrantes do grupo juntos do prédio B para o prédio A.

Para esse conjunto de dados o algoritmo foi executado com três conjuntos de valores distintos para os parâmetros, conforme ilustra a Tabela 9. Os valores dos parâmetros foram: duração mínima $minTime = [2min \text{ e } 5min]$, distância máxima $maxDistance = [5m \text{ e } 10m]$ e frequência mínima $minSup = 2$ encontros.

	Total de encontros	Encontros Espaciais	Encontros Espaço-temporais
$minTime = 2$ $maxDistance = 10$ $minSup = 2$	19	4	15
$minTime = 2$ $maxDistance = 5$ $minSup = 2$	32	14	18
$minTime = 5$ $maxDistance = 5$ $minSup = 2$	32	20	12

Tabela 9 - Encontros no conjunto de dados da UFSC

Pode-se observar na Tabela 9 que quanto maior a área do encontro, maior a probabilidade de mais trajetórias terem um encontro nessa mesma área. Quanto menor o *maxDistance*, menor será a área do encontro, diminuindo a probabilidade dessas áreas se interceptarem. Em casos em que o *maxDistance* é muito grande os objetos envolvidos no encontro não precisam estar muito próximos para que seja detectado um encontro, o que é mais realístico. Esse parâmetro deve ser o menor possível, mas existe para que seja possível detectar o padrão mesmo com dados imprecisos.

Com relação ao parâmetro temporal *minTime* a quantidade de encontros espaçotemporais fica mais restrita à medida que o valor do parâmetro aumenta. Os encontros com tempo menor englobam os encontros de maior duração, mas o contrário não acontece. Aumentando o valor do *minTime*, a quantidade total de encontros permanece igual, mas a quantidade de encontros espaçotemporais diminui e a de espaciais aumenta, conseqüentemente.

Para os encontros detectados com os parâmetros *minTime* = 2, *maxDistance* = 5 e *minSup* = 2 foram detectados encontros frequentes entre os 5 indivíduos. Os mesmos se encontraram, conforme ilustra a Figura 14, por duas vezes (*e1* e *e6*), caracterizando assim um encontro frequente.

Todos os encontros conhecidos foram detectados. O número de encontros total ficou acima do de encontros conhecidos porque um encontro com todos os objetos às vezes representa diversos encontros. Por exemplo, se dois indivíduos chegarem antes no local combinado, eles permanecerão juntos por um tempo maior que os demais que chegaram mais tarde. Assim este é caracterizado como dois encontros diferentes. Com esse experimento pôde-se demonstrar que o algoritmo detecta os padrões de encontro, sejam espaciais, espaçotemporais ou frequentes.

4.2 Conjunto de dados da Gincana em Amsterdam

O segundo conjunto de dados foi coletado entre os dias 7 e 9 de Fevereiro de 2005 em uma gincana móvel promovida pela *Waag Society* (SOCIETY, 2005). A gincana foi realizada na cidade de Amsterdam, na Holanda, onde estudantes de 12 a 14 anos receberam aparelhos celulares com GPS. Os estudantes receberam ainda um mapa com instruções para localizar pontos históricos. Nos locais destes pontos históricos devem ocorrer encontros entre alunos, sejam de um mesmo grupo ou grupos diferentes, gerando encontros espaciais e espaçotemporais. Neste conjunto de dados os encontros não são conhecidos mas o experimento foi realizado para ilustrar que o método é capaz de identificar encontros em qualquer conjunto de trajetórias.

Esse conjunto de dados possui 466 trajetórias e 62.759 pontos. Foi utilizado um subconjunto com 45 trajetórias e 15.258 pontos. Para formar o subconjunto foram eliminadas trajetórias com *outliers*, ou seja, que continham pontos com valores muito diferentes dos demais. Eliminou-se também trajetórias com menos de 100 pontos. A seleção de um subconjunto de trajetórias foi feita para que o resultado final pudesse ser melhor observado. Os encontros podem ocorrer de forma sobreposta e a presença de trajetórias com poucos pontos ou com pontos desviados poderia prejudicar o entendimento do resultado final.

A Figura 15 apresenta a comparação das trajetórias originais (esquerda) e as trajetórias selecionadas (direita). É possível observar na Figura 15, pelos pontos esparsos (ruído), que a quantidade de *outliers* é alta.



Figura 15 - Comparativo entre trajetórias originais e selecionadas de Amsterdam

4.2.1 Resultados

A partir das trajetórias selecionadas o algoritmo foi executado com dois conjuntos de valores distintos para os parâmetros, conforme ilustra a Tabela 10. Os valores dos parâmetros foram: duração mínima $minTime = 5min$, distância máxima $maxDistance = 2m$ e $10m$ e frequência mínima $minSup = 5$ encontros. Considerando a duração mínima do encontro como 10 minutos, distância máxima 10 metros e suporte mínimo = 5 foram detectados 147 encontros, dos quais 108 são espaciais e 39 espaçotemporais. Diminuindo a distância para 2 metros foram detectados 279 encontros, dos quais 255 são espaciais e 24 espaçotemporais.

	Total de encontros	Encontros Espaciais	Encontros Espaçotemporais
$minTime = 10$ $maxDistance = 10$ $minSup = 5$	147	108	39
$minTime = 10$ $maxDistance = 2$ $minSup = 5$	279	255	24

Tabela 10 - Encontros em Amsterdam

É possível observar que nesse conjunto de dados a quantidade de encontros espaciais é, ao menos, 50% maior que a de encontros espaçotemporais.

Isso ocorre porque os grupos participantes da gincana chegavam aos locais sinalizados em tempos diferentes. Mais uma vez, ao diminuir o parâmetro *maxDistance* o número total de encontros aumentou.

Como a quantidade trajetórias desse conjunto de dados é maior e conseqüentemente de encontros também, é possível fazer uma melhor análise dos encontros frequentes. A Tabela 11 ilustra os conjuntos de encontros frequentes com ao menos três trajetórias, em resultados obtidos com os parâmetros *minTime* = 10, *maxDistance* = 10, *minSup* = 5.

TIDs	Frequência	Região
[13, 15, 29]	5	A
[11, 15, 37]	5	A
[15, 29, 37]	5	A
[20, 41, 43]	5	B

Tabela 11 - Encontros frequentes em Amsterdam

Os diferentes encontros ocorreram em locais próximos, com tempos divergentes. Mesmo com uma distância mínima de 10 metros, alguns encontros espaçotemporais ficaram divididos devido à imprecisão dos aparelhos GPS. É possível observar que as três primeiras linhas da tabela 11 repetem certos elementos. Esses encontros ocorreram em regiões bem próximas, com tempos distintos, na região A, ilustrada na Figura 16. Já os encontros frequentes da última linha da tabela 11, se concentraram na região B da mesma figura. A Figura 16 ilustra todos os encontros espaciais (cor mais clara) e espaçotemporais (cor mais escura) obtidos com os parâmetros: *minTime* = 10, *maxDistance* = 10, *minSup* = 5.

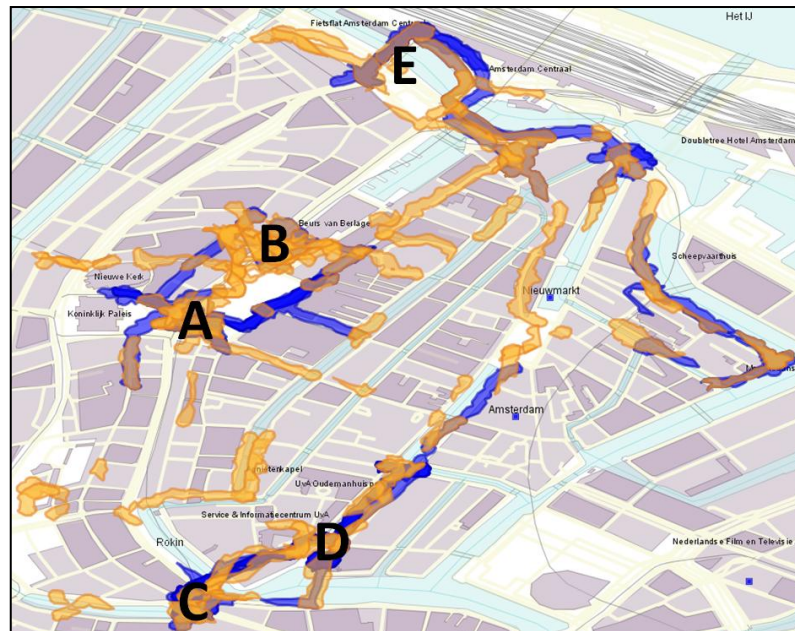


Figura 16 - Encontros espaciais e espaçotemporais em Amsterdam

É possível observar na Figura 16 cinco regiões com uma maior concentração de encontros. Verificando as coordenadas dessas regiões foi possível detectar locais históricos da cidade (A, B, C e D) e também um local para armazenamento de bicicletas (E). Os locais são listados a seguir:

- A. *National Monument* e Palácio Real de Amsterdam
- B. *Rembrandt* e sua ronda de noite em Amsterdam
- C. *Munttoren*
- D. *Casa de Rembrandt*
- E. *Fietsflat*

Não se sabe se os locais listados faziam parte da gincana realizada. Quanto ao estacionamento de bicicletas, este poderia ser um local de encontro das equipes, ou de reunião ao final do jogo. A cidade de Amsterdam é popularmente conhecida por utilizar bicicletas com meio de transporte. Dessa forma, muitos dos participantes da gincana devem ter se deslocado até a área da gincana de bicicleta.

5. CONCLUSÕES E TRABALHOS FUTUROS

Nas duas últimas décadas ocorreu a popularização de diversos tipos de dispositivos móveis que permitem gravar a localização de objetos, sejam pessoas, carros ou animais. Consequentemente, ocorreu também um aumento na quantidade de dados gerados a partir desses dispositivos, dados esses chamados de trajetórias de objetos móveis. A partir desses dados foram desenvolvidos diversos estudos para detectar padrões de comportamento desses objetos.

Dentre os estudos realizados, alguns focaram no comportamento de indivíduos isolados, como o *avoidance* [ALVARES, 2011], enquanto outros permitem detectar os comportamentos de interação entre indivíduos, como o *encontro*. Alguns autores propuseram métodos para a detecção de encontros. No entanto, esses métodos necessitam de uma referência do local pré-estabelecido. Nenhum desses métodos detecta encontros espaciais, nem avalia a frequência dos mesmos, funções que podem ser interessantes em uma aplicação investigativa, por exemplo.

O presente trabalho apresentou as definições de diferentes tipos de encontro: espacial, espaçotemporal e frequente. Os encontros espaciais permitem identificar locais visitados pelos objetos em tempos diferentes. Os espaçotemporais são os encontros entre indivíduos no mesmo local e tempo. Os frequentes são os que ocorrem entre os mesmos objetos repetidamente. Também foi definido o conceito de área de encontro e apresentado o conceito de encontros evolutivos. Foi apresentado o algoritmo para a detecção de encontros, bem como os algoritmos auxiliares para verificação temporal e agrupamento de encontros similares.

Para avaliar o método proposto foram realizados experimentos com dois conjuntos de dados diferentes. O primeiro foi coletado por alunos da UFSC, que simularam encontros no entorno do Campus Universitário. Esse primeiro experimento mostrou que o método proposto detecta os encontros corretamente, pois os mesmos eram conhecidos e foram detectados. O segundo experimento foi realizado com uma amostra de dados de uma gincana ocorrida em Amsterdam. O algoritmo detectou os encontros dos dois conjuntos de dados dentro do esperado.

Com relação aos trabalhos futuros pretende-se:

- Identificar se o encontro espaçotemporal ocorreu em movimento ou parado, caracterizado por *stopEncounter* e *movingEncounter*;
- Classificar encontros espaçotemporais de acordo com a duração dos mesmos, por exemplo *longEncounter*, quando a duração do encontro é longa e *shortEncounter*, quando um encontro tem curta duração;
- Classificar encontros espaçotemporais em marcados ou casuais a partir de uma agenda;
- A partir de uma base de dados, com vários objetos em dias diferentes, classificar os encontros frequentes de acordo com o dia da semana, ou mês em que ocorrem.
- Analisar os encontros de forma mais detalhada de acordo com a direção e velocidade dos mesmos;
- Melhorar a performance do algoritmo atual;
- Melhorar a forma como a área do encontro é detectada. Ao invés de usar a intersecção das trajetórias, usar somente a intersecção das subtrajetórias do encontro espaçotemporal.

REFERÊNCIAS

- ALVARES, L. O. ; Loy, A. M. ; Renso, C. ; BOGORNY, V.. **An algorithm to identify avoidance behavior in moving object trajectories.** Journal of the Brazilian Computer Society. v. 17, p. 193-203, 2011.
- BOGORNY, V. and Wachowicz, M. **A Framework for Context-Aware Trajectory Data Mining.** Longbing Cao, Philip S. Yu, Chengqi Ahang, Huaifeng Zhang. (Org.). Data Mining for Business Applications. Springer, 2008.
- BRAZ, F. J.; Bogorny, V. **Introdução a trajetórias de objetos móveis: Conceitos, armazenamento e análise de dados.** Editora Univille. 2012.
- CASTRO, J. **Nove meses após aprovação da lei de monitoramento eletrônico, menos da metade dos estados testou tornozeleiras.** Versão *On Line* do Jornal Extra. Disponível em <<http://extra.globo.com/noticias/brasil/nove-meses-apos-aprovacao-da-lei-de-monitoramento-eletronico-menos-da-metade-dos-estados-testou-tornozeleiras-1427200.html>>. Acesso em Março de 2013.
- Eclipse Foundation. Eclipse - The Eclipse Foundation open source community website. 2011. Disponível em: <<http://www.eclipse.org>>. Acesso em Março de 2013.
- Geofabrik. *OpenStreetMap Data Extracts.* Disponível em: <<http://download.geofabrik.de>>. Acesso em Outubro de 2013.
- GUDMUNDSSON, J.; van Kreveld M.; Speckmann B.. **Efficient Detection of Patterns in 2D Trajectories of Moving Points.** Geoinformatica vol. 11, no. 2. p. 195–215, 2007.
- GÜTING R.H., Dieker S., Freundorfer C., Becker L., Schenk H., **Secondo/QP: Implementation of a Generic Query Processor.** 10th Int. Conf. on Database and Expert System Applications (DEXA'99), LNCS 1677, Springer Verlag, 66-87, 1999.
- LAUBE P.; van Kreveld M.; Imfeld S. **Finding REMO-Detecting relative motion patterns in geo-spatial lifelines.** Developments in Spatial Data Handling: Proc. of the 11th International Symposium on Spatial Data Handling, Springer, Berlin Heidelberg New York, p. 201-214. 2004.
- LAUBE, P.; Imfeld, S.;Weibel, R. **Discovering relative motion patterns in groups of moving point objects.** International Journal of Geographical Information Science, v.19, n.6; p. 639-668, 2005.
- LI Z., Ding B., Han J., Kays R., Nye P. **Mining Periodic Behaviors for Moving Objects.** *In KDD*, pages 1099–1108, 2010.
- PALMA, A. T; Bogorny, V.; Kuijpers, B.; Alvares, L.O. **A Clustering-based Approach for Discovering Interesting Places in Trajectories.** 23rd Annual Symposium on Applied Computing, (ACM-SAC'08), Fortaleza, Ceará, 16-20 March (2008) Brazil. pp. 863-868.

PELEKIS N., Theodoridis Y., Vosinakis S. and Panayiotopoulos T. **Hermes - A Framework for Location-Based Data Management**. In the Proceedings of the 10th International Conference on Extending Database Technology (EDBT06) , LNCS 3896, pp. 1130-1134, Munich, Germany, 2006.

Quantum GIS Development Team. **Quantum GIS Geographic Information System**. [S.l.], 2009. Disponível em <<http://qgis.osgeo.org>>. Acesso em Março de 2013.

SIQUEIRA, F.L.; Bogorny, V. **Discovering Chasing Behavior Patterns in Moving Object Trajectories**. Transactions in GIS, 2011.

Refractions Research. **PostGIS**. 2011. Disponível em: <<http://postgis.refractions.net/>>. Acesso em Março de 2013.