

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

Fernando de Lucca Siqueira

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

Florianópolis

2012

Fernando de Lucca Siqueira

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

Dissertação submetida ao Programa
de Pós-Graduação em Ciência da Com-
putação para a obtenção do Grau de
Mestre em Ciência da Computação.
Orientadora: Prof. Dra. Vania Bo-
gorny

Florianópolis

2012

Catálogo na fonte elaborada pela biblioteca da
Universidade Federal de Santa Catarina

A ficha catalográfica é confeccionada pela Biblioteca Central.

Tamanho: 7cm x 12 cm

Fonte: Times New Roman 9,5

Maiores informações em:

<http://www.bu.ufsc.br/design/Catalogacao.html>

Fernando de Lucca Siqueira

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

Esta Dissertação foi julgada aprovada para a obtenção do Título de “Mestre em Ciência da Computação”, e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

Florianópolis, 28 de fevereiro 2012.

Prof. Dr. Ronaldo dos Santos Mello
Coordenador do Curso

Prof. Dra. Vania Bogorny
Orientadora

Banca Examinadora:

Prof. Dra. Vania Bogorny
Presidente

Prof. Dr. Ronaldo dos Santos Mello

Prof. Dr. José Leomar Todesco

Prof. Dr. Marco Antônio Casanova

Dedico este trabalho aos meus pais Amaro das Flores Siqueira Neto e Marilene de Lucca Siqueira, que sempre acreditaram em meus sonhos.

AGRADECIMENTOS

Agradecimento especial para minha orientadora, Vania Bogorny, que tornou tudo isso possível. Sempre disposta a me guiar pelo caminho certo, enquanto me deixava caminhar com meus próprios pés, seus conselhos e críticas fizeram deste trabalho o que ele é hoje. Obrigado pela paciência, ensinamentos e por me orientar neste mestrado.

Meus agradecimentos aos meus pais, Amaro das Flores Siqueira Neto e Marilene de Lucca Siqueira, que confiaram e me apoiaram durante todos estes anos que estive estudando fora de casa. O que eu sou hoje devo tudo a eles.

Também agradeço a minha irmã Marina de Lucca Siqueira e ao seu marido Alessandro Tavares dos Santos quer tornaram minha estadia em Florianópolis mais familiar.

Palavras não podem expressar meus agradecimentos a todos os meus amigos espalhados por o todo país. Meus amigos de infância de Minas Gerais, que ainda mantenho contato, meus amigos da faculdade de Florianópolis, que posso contar sempre e meus amigos de vários lugares que conheci pela internet, provando que não é preciso carne e osso para se ter uma amizade verdadeira.

Gostaria de agradecer a Universidade Federal de Santa Catarina, especialmente ao Departamento de Informática e Estatística, por ter sido meu lar de trabalho e estudo durante estes anos. Também agradeço ao Grupo de Banco de Dados da UFSC que me ajudou fazendo críticas e sugestões ao trabalho. Agradeço ao CNPQ por apoiar minha pesquisa durante meu mestrado.

*Prefiro ser odiado por quem eu sou, do
que ser amado pelo que não sou.*

Kurt Cobain

RESUMO

Tecnologias como celulares, GPS e redes de sensores estão ficando cada vez mais populares. Estes dispositivos geram uma grande quantidade de dados chamados de Trajetórias de Objetos Móveis. Uma trajetória é um conjunto de pontos localizados no espaço e no tempo. Estes dados são normalmente volumosos e confusos, sendo necessário criar métodos e algoritmos para extrair informações interessantes destes dados. Vários estudos tem focado na descoberta de padrões em trajetórias como flocks, desvios, recorrência, liderança, etc. Neste trabalho é proposto um novo tipo de padrão: comportamento de perseguição em trajetórias. Mais especificamente, são apresentadas definições formais do comportamento e são definidos diferentes tipos de perseguição, bem como um algoritmo para identificar o padrão. As principais características consideradas são o tempo, a distância e a velocidade, que são utilizadas de forma diferente em relação a trabalhos existentes. O trabalho é validado com experimentos sobre dados sintéticos e dados reais, demonstrando que o método encontra padrões não identificados por outras abordagens.

Palavras-chave: Trajetórias de Objetos Móveis. Mineração de Dados. Padrões de Comportamento. Perseguição em Trajetórias.

ABSTRACT

Modern tracking technology like GPS, cell phones and sensor networks are becoming very popular. These devices generate a large amount of data, called trajectories of moving objects. A trajectory is a list of space-time points. These data are typically large and confuse. Therefore, there is a need for creating new methods and algorithms to extract interesting information from these data. Several studies have focused on the discovery of behavior patterns like flocks, avoidance, recurrence, leadership, etc. In this work we propose a new pattern: chasing behavior in trajectories. More specifically, we present a formal definition of the behavior and different types of chasing, as well an algorithm to compute the pattern. The main thresholds considered are time, distance, and speed, which are used in a different way, if compared with related work. Our work is validated with experiments on both synthetic data and real data, showing that the method finds patterns that are not discovered by other approaches.

Keywords: Moving Object Trajectories. Data Mining. Behaviour Patterns. Trajectory Chasing.

LISTA DE FIGURAS

Figura 1 (a) Representação de uma trajetória num espaço 3D, (b) várias trajetórias num espaço 3D e (c) formato de dados de trajetórias.....	25
Figura 2 Representação da trajetória das tropas de Napoleão (Encyclopædia Britannica Online, 2012).....	26
Figura 3 Exemplo de stops em uma trajetória (PALMA et al., 2008)	32
Figura 4 Exemplo de stops gerados com base na variação da direção (ROCHA et al., 2010).....	33
Figura 5 Exemplo de classificação de trajetórias em (LEE et al., 2008).....	34
Figura 6 Exemplo de comportamento avoidance de (ALVARES et al., 2011).....	36
Figura 7 Exemplo de flock e leadership de (LAUBE; IMFELD; WEIBEL, 2005).....	37
Figura 8 Padrão de Perseguição.....	41
Figura 9 Candidato a Perseguição.....	42
Figura 10 Segmento de Linha Representativa (q15,q18).....	44
Figura 11 (a)Sub-trajetória (q7, ..., q10) está próxima de L1 (b)Sub-trajetória (p5, ..., p8) está distante de L2.....	45
Figura 12 Sub-Perseguições 1, 2, 3 e 4 foram uma Perseguição de 24 minutos.....	46
Figura 13 Tipo de perseguição de detetive.....	47
Figura 14 Tipo de perseguição de captura.....	48
Figura 15 Tipo de perseguição de assalto.....	49
Figura 16 Tipo de perseguição de caça.....	50
Figura 17 Visualização do quarto conjunto de dados no Google Earth (Google Inc., 2009) em Outubro, 2011.....	60
Figura 18 Trajetórias do primeiro conjunto de dados.....	61
Figura 19 Padrão encontrado pelos algoritmos PTC, STC e CDA para $\Delta d=15m$ $\Delta t= 1$ minuto.....	64
Figura 20 Padrão encontrado pelos algoritmos PTC e STC para $\Delta d=15m$ $\Delta t= 1$ minuto.....	65
Figura 21 Comparação entre a perseguição real, o CDA e o STC para a perseguição C1 e visualização no Google Earth (Google Inc.,	

2009) em Março, 2011.....	67
Figura 22 (a)Pontos p8,p9 de S1 e pontos q10,q11 de S2 no mesmo local X. (b) Distância d1 analisada por MFF e CDA para o ponto p5 e a distância d2 analisada pelo TRA-CHASE no ponto p5.	68
Figura 23 Stops encontrados na perseguição C3 e sua localização nas trajetórias da UFSC.....	70
Figura 24 Stops encontrados na perseguição C2 e sua localização nas trajetórias.....	71
Figura 25 Resultados para a Perseguição C3 com parâmetros $\Delta d = (3m, 10m, 20m)$ e $\Delta t = 30s$	74
Figura 26 Resultados para a Perseguição C2 com parâmetros $\Delta t = (10s, 30s, 150s)$ e $\Delta d = 10m$	75

LISTA DE TABELAS

Tabela 1	Resumo dos Tipos de Padrão de Perseguição	50
Tabela 2	Padrões encontrados pelos algoritmos com distância $\Delta d = 80.0m$, tolerância de tempo $\Delta t = 5min$ e duração mínima $\Delta c = 10min$	62
Tabela 3	Número de padrões encontrados pelos algoritmos PTC, STC, CDA e MFF com os dados originais	63
Tabela 4	Número de padrões encontrados pelos algoritmos PTC, STC, CDA e MFF com os dados sincronizados	63
Tabela 5	Comparação dos padrões encontrados pelos diferentes algoritmos	66
Tabela 6	Tipos de perseguição por grupo	69
Tabela 7	Padrões encontrados para o conjunto de dados da UFSC	70
Tabela 8	Stops encontrados pelo método CB-SMoT nas trajetórias T1, T2, T3 e T4	70
Tabela 9	Tempo de processamento em milissegundos para cada conjunto de dados	72

LISTA DE ABREVIATURAS E SIGLAS

CB-SMoT	Clustering Based Stops and Moves of Trajectory
CDA	Collocation Discovery Algorithm
CLASS-CHASE	Class Chasing
DBSCAN	Density-Based Spatial Clustering of Applications With Noise
GPS	Global Positioning System
KDDLab	Knowledge Discovery and Data Mining Laboratory
MFF	Moving Flock Finder
PTC	Pure Trajectory Chasing
STC	Speed Trajectory Chasing
SUB-CHASE	Sub-trajectory Chasing
TRA-CHASE	Trajectory Chasing

SUMÁRIO

1 INTRODUÇÃO	25
1.1 DEFINIÇÃO DO PROBLEMA E JUSTIFICATIVA	27
1.2 OBJETIVOS	28
1.3 METODOLOGIA E ESTRUTURA DA DISSERTAÇÃO	28
2 CONCEITOS BÁSICOS E ESTADO DA ARTE	31
2.1 CONCEITOS BÁSICOS	31
2.2 ESTADO DA ARTE	34
3 IDENTIFICANDO PADRÕES DE PERSEGUIÇÃO ENTRE TRAJETÓRIAS DE OBJETOS MÓVEIS ...	41
3.1 DEFINIÇÃO DO PADRÃO DE PERSEGUIÇÃO	41
3.2 TIPOS DE PERSEGUIÇÃO	46
3.3 UM ALGORITMO PARA IDENTIFICAR PADRÕES DE PERSEGUIÇÃO	50
4 EXPERIMENTOS	59
4.1 DESCRIÇÃO DOS CONJUNTOS DE DADOS	59
4.2 EXPERIMENTOS PARA IDENTIFICAR PADRÕES GE- NÉRICOS DE PERSEGUIÇÃO	61
4.2.1 Conjunto de Dados Sintéticos	61
4.2.2 Conjunto de Dados da Gincana de Amsterdã	62
4.2.3 Conjunto de Dados de Jurerê	65
4.3 EXPERIMENTOS PARA IDENTIFICAR TIPOS DE PER- SEGUIÇÃO	68
4.3.1 Conjunto de Dados da Gincana de Amsterdã	68
4.3.2 Conjunto de Dados da UFSC	69
4.4 ANÁLISE DA PERFORMANCE	72
4.5 ANÁLISE DOS PARÂMETROS	73
5 CONCLUSÃO E TRABALHOS FUTUROS	77
REFERÊNCIAS	79

1 INTRODUÇÃO

O barateamento de tecnologias como GPS, celulares e redes de sensores difundiu o uso destes dispositivos na sociedade. Tais dispositivos geram uma grande quantidade de um novo tipo de dado espaço-temporal, chamado de *trajetórias de objetos móveis*. Uma trajetória, como ilustra a figura 1(a), é um conjunto de pontos localizados no espaço e no tempo e que caracteriza o movimento de um objeto. Existem diversas maneiras de gerar uma trajetória. Por exemplo, um dispositivo pode capturar a localização geográfica a cada segundo, a cada 10 segundos, a cada 5 metros, ou quando o objeto atinge certo local. Por isso, o volume dos dados gerados é muito grande, tornando difícil a sua análise, conforme pode ser visto na figura 1(b).

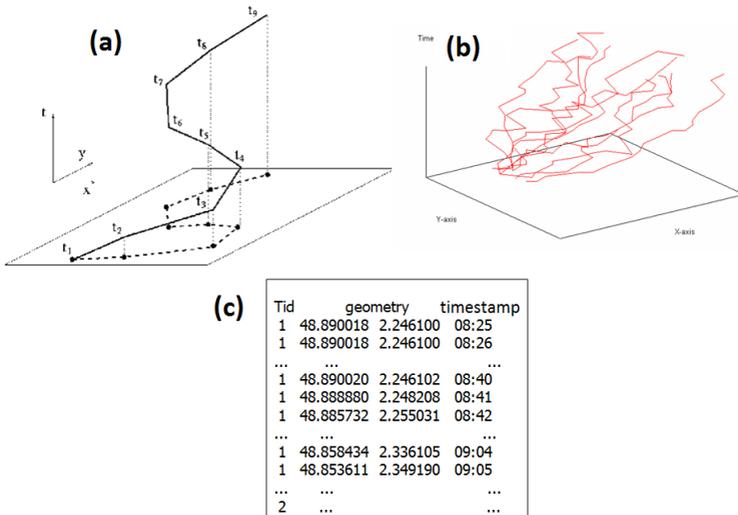


Figura 1 – (a) Representação de uma trajetória num espaço 3D, (b) várias trajetórias num espaço 3D e (c) formato de dados de trajetórias.

Cada trajetória tem um conjunto de propriedades próprias que descrevem o movimento de um objeto. Segundo Giannotti (GIANNOTTI; PEDRESCHI, 2008), tais propriedades podem ser divididas em características momentâneas e características gerais. As características momentâneas se referem ao estado do objeto em um determinado momento como, por exemplo, o instante de tempo, a posição espa-

cial, a direção do movimento, a velocidade do objeto, a aceleração, o tempo e a distância acumulados. Já as características gerais referem-se a trajetória como um todo como, por exemplo, a forma geométrica, o tamanho total da trajetória, a duração, a velocidade média, 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.

Trajетórias também podem representar o movimento de diversos objetos diferentes. 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 e até mesmo redes sociais podem ser expressas através de uma trajetória. Um dos exemplos mais antigos desta representação é a campanha de Napoleão contra a Rússia em 1812, conforme ilustra figura 2. A figura ilustra a quantidade de soldados da tropa de Napoleão na ida e na volta de Moscou, bem como os locais por onde os soldados passaram. É possível notar que a trajetória do grupo de soldados na ida à Moscou era bem mais larga do que no retorno. A morte de vários soldados reduziu significativamente a largura da trajetória.

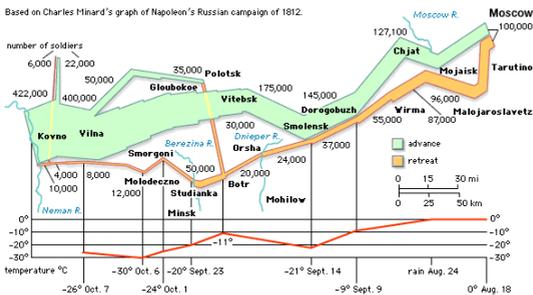


Figura 2 – Representação da trajetória das tropas de Napoleão (Encyclopædia Britannica Online, 2012)

Além das características de movimento, uma trajetória pode apresentar diversos comportamentos distintos como, por exemplo, andar lentamente ou rapidamente, seguir a mesma rota diariamente, fazer várias curvas, parar, evitar lugares específicos, etc. Quando um comportamento específico é observado em várias trajetórias distintas, aquele comportamento é considerado um padrão seguido por diversas trajetórias, seja o padrão accidental ou proposital.

O grande desafio na análise de trajetórias é que os dados são volumosos e confusos, tornando difícil a extração de informações interessantes para o processo de tomada de decisão, conforme pode ser notado na figura 1(b). Tal desafio é observado em diversos estudos de mineração de dados, não sendo um problema exclusivo de trajetórias. Diversos estudos focam na área de análise de trajetórias de objetos móveis tentando extrair padrões de comportamento destes grandes conjuntos de dados como, por exemplo, *sequências de eventos periódicos* (CAO; MAMOULIS; CHEUNG, 2006), *desvios* (ALVARES et al., 2011), *flocks* (WACHOWICZ et al., 2011), entre outros.

Um padrão de comportamento pouco explorado na literatura é o de *perseguição* em trajetórias, que é o foco deste trabalho.

1.1 DEFINIÇÃO DO PROBLEMA E JUSTIFICATIVA

Diversos trabalhos tem sido propostos para análise de comportamento, mas poucos trabalhos abordam ou transmitem a ideia de perseguição em trajetórias. Os trabalhos (DODGE; WEIBEL; LAUTENSCHÜTZ, 2008), (LEGENDRE et al., 2006) e (REYNOLDS, 1999) abordam o comportamento de perseguição, mas nenhum deles define formalmente o padrão e um algoritmo para computá-lo.

Vários objetos podem apresentar o comportamento de perseguição, tornando o padrão útil em várias aplicações. Por exemplo, monitoramento de veículos de pessoas importantes que podem ser perseguidos pela mídia ou terroristas, barcos perseguidos por piratas, animais perseguindo presas ou até mesmo em jogos eletrônicos, onde um inimigo pode perseguir um jogador. A identificação automática do comportamento de perseguição pode ser útil na identificação de suspeitos de crimes como assassinos, ou indivíduos que perseguem suas vítimas antes de cometer o crime. Em ambientes fechados, como lojas ou shopping centers, câmeras de vigilância podem gerar trajetórias de diversos indivíduos, sendo possível analisar o comportamento de perseguição entre eles. Até mesmo em jogos de futebol (KANG; HWANG; LI, 2006), uma perseguição pode identificar os jogadores mais marcados e ajudar a criar novas táticas de ataque ou defesa. Pode-se identificar perseguições também em frotas de ônibus, onde um motorista pode tentar perseguir e eventualmente ultrapassar outro ônibus da mesma rota.

Um padrão de perseguição pode ser encontrado em diversos lugares, com várias aplicações em potencial que justificam este trabalho.

1.2 OBJETIVOS

O objetivo principal deste trabalho é identificar o padrão de perseguição em trajetórias de objetos móveis. Os objetivos específicos são:

- definir formalmente o padrão de perseguição em trajetórias de objetos móveis;
- definir formalmente tipos de perseguição;
- definir um algoritmo para computar o padrão de perseguição em um conjunto de trajetórias.

1.3 METODOLOGIA E ESTRUTURA DA DISSERTAÇÃO

A metodologia utilizada nesta dissertação compreende os seguintes passos:

- definir formalmente os conceitos básicos do comportamento de perseguição em trajetórias de objetos móveis e diferentes tipos de padrões de perseguição;
- definir e implementar um algoritmo para identificar o padrão de perseguição em trajetórias;
- coletar dados de trajetórias para experimentos;
- realizar experimentos, comparando o algoritmo com outras abordagens utilizando:
 - dados sintéticos
 - dados reais
 - dados gerados para simular perseguições
- validar os padrões de perseguição com dados gerados;
- validar a identificação de tipos de padrões de perseguição;
- analisar o algoritmo e os resultados;
- analisar e discutir os parâmetros de entrada.

O restante desta dissertação está organizada da seguinte forma: O capítulo 2 descreve os conceitos básicos e apresenta os trabalhos relacionados existentes na literatura.

O capítulo 3 apresenta a contribuição deste trabalho, com as definições formais e o algoritmo TRA-CHASE (Trajectory Chasing).

O capítulo 4 apresenta os experimentos realizados para validar o algoritmo TRA-CHASE, uma discussão sobre o mesmo e uma análise dos parâmetros de entrada.

O capítulo 5 apresenta as conclusões finais do trabalho e sugestões de próximos passos.

2 CONCEITOS BÁSICOS E ESTADO DA ARTE

Um padrão em dados de trajetórias é um movimento que se repete várias vezes dentro de uma trajetória ou entre várias trajetórias. Para entender o trabalho proposto, este capítulo apresenta alguns conceitos, definições, e trabalhos existentes na área de análise de trajetórias de objetos móveis.

2.1 CONCEITOS BÁSICOS

Dispositivos móveis geram dados espaço-temporais chamados de trajetórias de objetos móveis. A trajetória de um objeto móvel é um conjunto de pontos localizados no tempo e no espaço, com um identificador único do objeto, conforme a definição 1.

Definição 1. Trajetória Uma *trajetória* T é um conjunto $(tid, p_0, p_1, \dots, p_n)$, 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 $(x_i, y_i, t_i) \in R$ para cada $i = 0, \dots, n$ e $t_0 < t_1 < \dots < t_n$.

Uma trajetória, conforme ilustra a figura 1, é gerada capturando a coordenada geográfica de um objeto em um instante de tempo ou após certa distância de deslocamento. A coleta do ponto varia de acordo com a configuração do dispositivo móvel, tornando possível a geração de pontos a cada 1 segundo ou 5 minutos, 3 metros ou 20 metros, etc. Por consequência, uma trajetória tem uma enorme quantidade de pontos, sendo difícil analisar a trajetória como um todo. Além disso uma trajetória dificilmente irá se comportar de maneira constante durante toda sua vida. Por exemplo, um pedestre pode sair de casa e andar até uma loja. Durante seu trajeto, ele pode parar, andar mais rápido ou mais devagar, pegar um táxi ou um ônibus, etc. Sendo assim, a maioria dos padrões de trajetórias são identificados em sub-trajetórias, e não na trajetória completa. Para contornar estes problemas, surgiu a definição 2 de *sub-trajetória* (LEE et al., 2008), utilizada em diversos trabalhos da literatura.

Definição 2. Sub-Trajeto Uma *sub-trajetória* S de $T = (tid, p_0, p_1, p_2, \dots, p_n)$ é um conjunto $\langle p_i, p_{i+1}, \dots, p_{i+m}, \rangle$, onde $p_i \in T$ e $0 \leq i \leq (i + m) \leq n$.

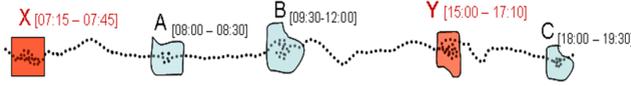


Figura 3 – Exemplo de stops em uma trajetória (PALMA et al., 2008)

Existem estudos que definem visões diferentes de trajetórias e sub-trajetórias. Spaccapietra (SPACCAPIETRA et al., 2008), por exemplo, apresenta uma visão conceitual de trajetórias, definindo a mesma como sendo um conjunto de *stops* e *moves*. De acordo com Spaccapietra, *stops* (definição 4) são regiões importantes onde uma trajetória permanece por um certo período de tempo. Por exemplo, na figura 3, a trajetória realizou 5 stops, onde no stop A, permaneceu na mesma região por 30 minutos (08:00 a 08:30). Uma região pode ser definida como a *geometria* (definição 3) de uma área de interesse. Por exemplo, uma região que representa um stop pode ser a geometria de um bairro, uma cidade, uma universidade, um shopping, etc.

Definição 3. Geometria Uma *geometria* G é um conjunto de pontos espaciais $\langle gid, g_0, g_1, \dots, g_n \rangle$, onde gid é o identificador da geometria, $g_i = (x_i, y_i)$ e $x_i, y_i \in R$ para cada $i = 0, \dots, n$.

Spaccapietra define que um *stop* é representado por uma geometria e uma duração.

Definição 4. Stop Um *stop* o de trajetória T é uma tupla (G, B) , onde G é uma geometria que contém uma sub-trajetória máxima S de T e B é um par (b_1, b_2) , onde b_1 é o tempo inicial de S e b_2 é o tempo final de S .

Spaccapietra define um modelo conceitual para trajetórias, porém não define como calcular ou identificar os stops e as regiões que representam os stops. Alguns trabalhos tem focado no cálculo/instanciação dos stops. Alvares (ALVARES et al., 2007) definiu que um stop ocorre quando uma trajetória intersecta uma região geográfica (geometria) por um certo período de tempo. Estas regiões são pré-definidas e chamadas de *candidato a stop*, onde um *candidato a stop* tem uma geometria e uma duração mínima. Por exemplo, para uma aplicação de turismo, um ponto turístico poderia ser a geometria de um *candidato a stop* e, se uma trajetória permanecer por um período de tempo mínimo nesta região, então seria gerado um stop para aquela região.

Palma (PALMA et al., 2008) definiu que stops são sub-trajetórias onde a trajetória teve velocidade baixa. A velocidade média da *sub-*



Figura 4 – Exemplo de stops gerados com base na variação da direção (ROCHA et al., 2010)

trajetória do stop não pode ser maior que a velocidade média da trajetória, e a velocidade baixa deve ser observada por um certo período de tempo. Por exemplo, em uma aplicação de trânsito, os stops poderiam ser interpretados como áreas de congestionamento, desde que o congestionamento tenha tido uma duração mínima. Palma definiu o algoritmo *CB-SMoT* (Clustering Based Stops and Moves of Trajectory) para identificar stops em áreas de velocidade baixa em trajetórias, definindo um stop como tendo uma geometria G , um tempo inicial $b1$ e final $b2$ e a velocidade baixa. O algoritmo *CB-SMoT* será utilizado neste trabalho.

Rocha (ROCHA et al., 2010) define um stop com base na variação da direção da trajetória, conforme ilustra a figura 4, onde as regiões escuras são as áreas da trajetória com maior variação da direção. Um stop é gerado se uma sub-trajetória varia sua direção acima de um determinado limite e por um certo período de tempo. Este método é bastante útil em aplicações onde a variação da direção é importante como, por exemplo, para identificar regiões de pesca. Quando o barco está pescando, o motor é desligado e o movimento do barco é controlado pelo próprio mar, fazendo a direção variar mais do que durante o movimento do barco na direção da área de pesca.

Esta seção apresentou alguns conceitos básicos sobre trajetórias e stops, que serão utilizados mais adiante neste trabalho. A seguir são apresentados os trabalhos relacionados.

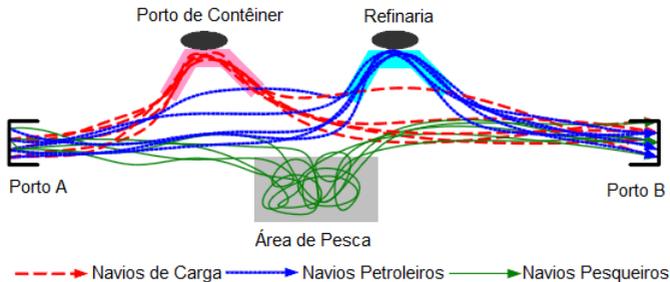


Figura 5 – Exemplo de classificação de trajetórias em (LEE et al., 2008)

2.2 ESTADO DA ARTE

Diversos trabalhos tem sido propostos para identificar padrões em trajetórias de objetos móveis. Eles podem ser divididos em três grupos principais.

O primeiro grupo tenta identificar padrões em grupos de trajetórias, onde uma trajetória não está ciente de outra e os padrões são extraídos entre objetos que não tem interação entre si. Como exemplos destes trabalhos podemos citar (GIANNOTTI et al., 2007), (LEE et al., 2008), (HORNSBY; COLE, 2007) e (CAO; MAMOULIS; CHEUNG, 2005). O objetivo destes trabalhos é extrair padrões entre objetos com movimentos similares. Giannotti (GIANNOTTI et al., 2007) propõe um algoritmo para extrair sequências de regiões frequentemente visitadas em uma ordem específica e com tempos de transição similares. Um padrão é gerado quando existir uma sequência de regiões visitadas por um número mínimo de trajetórias. Por exemplo, trajetórias de estudantes que saem de um metrô, demoram 10 minutos para chegar em uma ponte e depois levam mais 15 minutos para chegar em uma universidade.

Lee (LEE et al., 2008) propõe um método para classificar sub-trajetórias com diferentes comportamentos e diferentes objetivos. Trajetórias com o mesmo objetivo são adicionadas ao mesmo grupo. Por exemplo, na figura 5, trajetórias de navios que param em um porto de contêiner são classificadas como trajetórias de navios de carga. Trajetórias de navios em áreas de pesca são classificadas como trajetórias de navios pesqueiros, enquanto trajetórias de navios que param em refinarias são classificadas como trajetórias de navios petroleiros.

O trabalho de Hornsby (HORNSBY; COLE, 2007) define um modelo para representar movimentos de trajetórias através de eventos

como, por exemplo, *partida*, *chegada*, *repetição*, etc. Diferentes trajetórias podem apresentar o mesmo evento no mesmo local. Por exemplo, barcos pesqueiros que voltam para o porto podem ser modelados como trajetórias que tem uma partida do mar e uma chegada no porto. Se existir um número mínimo de trajetórias que podem ser modeladas com estes eventos, então existe um padrão de partida do mar e chegada no porto.

Cao (CAO; MAMOULIS; CHEUNG, 2005) (CAO; MAMOULIS; CHEUNG, 2007) busca identificar *padrões periódicos*, onde um objeto segue a mesma rota em tempos regulares. Por exemplo, uma pessoa pode todos os dias sair de casa às 7 horas, ir para o seu local de trabalho, ficar até as 18 horas e retornar ao lugar inicial. Cao definiu que um *padrão periódico* é uma sequência de regiões intersectadas pela trajetória, sendo as regiões informadas pelo usuário ou calculadas pelo método. Quando calculadas, as regiões são identificadas com o algoritmo de clusterização *DBSCAN* (ESTER et al., 1996), que identifica regiões densas. O método utiliza a direção, o comprimento e a distância para encontrar similaridade entre sub-trajetórias.

O trabalho de Kanda (KANDA et al., 2008) propõe um algoritmo que analisa as trajetórias de clientes de um shopping e classifica qual o tipo do cliente: cliente parado, andando rápido, olhando vitrines, etc. O resultado é analisado por um robô social que ajuda as pessoas de acordo com o seu tipo. Para isso, o robô deve antecipar as ações dos objetos (caminhando, andando rápido ou parado). O robô infere a ação do objeto observando onde ele se localiza no espaço e como se movimenta. Se o robô quer convidar pessoas para conhecer uma loja, ele deve buscar pessoas que andam perto de vitrines em velocidade baixa, e ignorar pessoas que andam no meio do corredor em alta velocidade. Já se for um robô de entretenimento, deve buscar pessoas que estão paradas em algum lugar e ignorar as outras em movimento.

O segundo grupo de trabalhos foca na identificação de padrões em trajetórias individuais, analisando o comportamento individual dos objetos. Os trabalhos anteriormente citados como Alvares (ALVARES et al., 2007), Rocha (ROCHA et al., 2010) e Palma (PALMA et al., 2008) são exemplos de trabalhos deste grupo. Outros trabalhos como Alvares (ALVARES et al., 2011), abordam um novo tipo de padrão em trajetórias, o *avoidance*, ilustrado na figura 6. O objetivo deste trabalho é verificar se uma trajetória evita uma região como, por exemplo, um ladrão evitando uma câmera de segurança. O trabalho também avalia a confiança do padrão, para assegurar se foi um desvio intencional ou acidental.

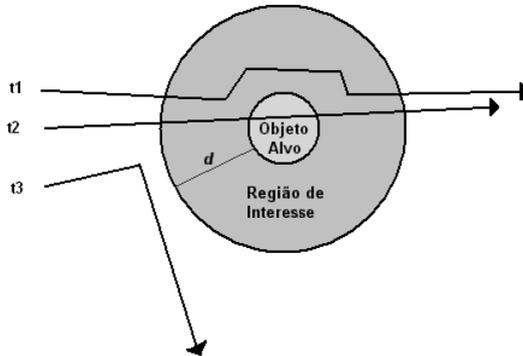


Figura 6 – Exemplo de comportamento avoidance de (ALVARES et al., 2011)

Baglioni (BAGLIONI et al., 2009) enriquece trajetórias individuais com informações semânticas obtidas de ontologias, buscando inferir o objetivo do objeto com base na trajetória. Por exemplo, uma trajetória que sempre começa em um mesmo local (ex. casa) e todo dia para no mesmo local por um certo período de tempo (ex. trabalho) é a trajetória de um trabalhador. Já uma trajetória que para várias vezes em pontos turísticos é uma trajetória de turista.

Os trabalhos dos dois grupos anteriores definem formalmente os padrões e propõem algoritmos para extrair os padrões de trajetórias. Existe um terceiro grupo de trabalhos que analisam padrões em grupos de trajetórias, onde o comportamento de uma trajetória afeta outras trajetórias. O trabalho aqui proposto se insere neste grupo.

Laube (LAUBE; IMFELD; WEIBEL, 2005), por exemplo, define cinco tipos de padrões de comportamento de trajetórias: *convergência*, *encontro*, *recorrência*, *flock* e *liderança*. Dentre estes, dois padrões são mais próximos a ideia de perseguição: *flock* e *liderança*, ilustrados na figura 7. O padrão de *flock* refere-se a um grupo de objetos que se move na mesma direção e no mesmo período de tempo. O método traça um círculo em torno de um ponto e procura por outros objetos dentro deste círculo que se movem na mesma direção e no mesmo instante de tempo. Por exemplo, um bando de pássaros migrando. O padrão de *liderança* faz uma pequena modificação ao anterior: o grupo deve ter um líder que se move em uma direção e, após um certo período de tempo, objetos próximos passam a se mover na mesma direção do líder. Ambos os padrões utilizam tempo, localização, direção e distância para

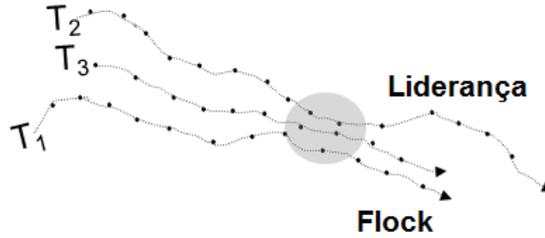


Figura 7 – Exemplo de flock e leadership de (LAUBE; IMFELD; WEIBEL, 2005)

identificar estes comportamentos, mas a velocidade e o comprimento do padrão não são considerados.

Dodge (DODGE; WEIBEL; LAUTENSCHÜTZ, 2008) propôs um framework conceitual de movimento e uma taxonomia de padrões de movimento. Este trabalho utiliza abordagens existentes, como (BLYTHE; MILLER; TODD, 1996), (GUDMUNDSSON; LAUBE; WOLLE, 2008), (ANDRIENKO; ANDRIENKO, 2007) e (LAUBE; IMFELD, 2002) para definir um conjunto de medidas para identificação de padrões de movimento (ex: distância, velocidade, direção, etc). Dodge divide os padrões de movimento em dois grupos: *padrões genéricos* e *padrões comportamentais*. A diferença principal destes grupos é que os padrões comportamentais são sensíveis ao contexto. Por exemplo, o movimento de uma certa espécie de animal é considerado um padrão comportamental. Um tipo de padrão comportamental definido foi de caça/evasão. Uma caça pode ser interpretada como um tipo de perseguição. Este padrão é descrito como um animal tentando escapar de um predador que o persegue, onde ambas as trajetórias tem movimentos rápidos e com várias curvas. Dodge apenas descreveu o padrão e suas medidas associadas, sem apresentar uma definição formal ou um algoritmo específico para computá-lo.

Legendre (LEGENDRE et al., 2006) propôs uma nova abordagem para modelagem de trajetórias. O trabalho define movimentos de objetos com regras de comportamento, onde um objeto tem certo comportamento dependendo do contexto. Por exemplo, a caminhada de um objeto em um prédio deve seguir algumas regras como evitar paredes, obstáculos e outros objetos. Este trabalho define um comportamento de perseguição, onde um objeto deve se mover na direção de

outro objeto ou de um ponto estático, mas nem o tempo nem a distância são considerados para identificar se um objeto persegue outro por um período de tempo.

Em Hornsby (HORNSBY; KING, 2008) é apresentado um conjunto de relações de movimento de veículos em rodovias. Estas relações, como *estáAtrás*, *naFrenteDe*, *dirigeAoLado* e *ultrapassa*, descrevem a posição relativa entre dois veículos em um tempo específico. Por exemplo, se um veículo está na frente ou atrás de outro veículo. Uma perseguição poderia ser interpretada como um objeto atrás de um alvo por um certo período de tempo. Entretanto, a relação só é definida para um instante de tempo específico e não para um período de tempo, sendo considerado apenas quando um objeto está exatamente atrás de outro no mesmo instante de tempo. Estas restrições descaracterizam uma perseguição.

Reynolds (REYNOLDS, 1999) abordou o problema de personagens autônomos em um mundo virtual. Ele definiu uma série de ações e movimentos como *procurar*, *caçar*, *fugir* e *passar*, que juntos modelam o que ele chama de *comportamentos de direção*. O objetivo principal do trabalho é definir um caminho a ser seguido pelo personagem a partir de um objetivo como, por exemplo, seguir um corredor evitando obstáculos. O comportamento de *caça* poderia caracterizar uma perseguição, porém a proposta é definir o caminho a ser seguido por um objeto em tempo real, enquanto a proposta desta dissertação é identificar padrões de perseguição que já ocorreram, analisando trajetórias passadas, e não definir a rota de um objeto. Assim, características diferentes dos dados precisam ser consideradas.

Recentemente, Wachowicz (WACHOWICZ et al., 2011) propôs um algoritmo que identifica flocks entre objetos que estão andando juntos por um certo período de tempo. A grande diferença para o trabalho de (LAUBE; IMFELD; WEIBEL, 2005) é que as trajetórias não podem estar imóveis, identificando somente flocks em movimento. O maior problema desta abordagem é que os objetos precisam estar andando juntos no mesmo instante de tempo, sendo as vezes necessário que o conjunto de dados seja sincronizado com uma ferramenta externa, para que todas as trajetórias tenham intervalo de captura iguais. Isto é um sério problema quando se trata de dados reais, que podem ser gerados de várias formas diferentes. Apesar de uma perseguição poder ser interpretada como um flock de dois objetos, os objetos nem sempre estarão tão próximos no mesmo instante de tempo, como Wachowicz sugere.

Em (CAO; MAMOULIS; CHEUNG, 2006), Cao explora *Collocation Episodes* em trajetórias. O objetivo principal do trabalho é encontrar relações entre objetos que resultam em relações com um terceiro objeto.

Por exemplo, uma onça caçando um cervo. Após caçar o cervo, ou seja, ocorrer uma relação entre os dois objetos, um corvo irá se relacionar com o cervo morto. Cao identifica episódios co-relacionados entre objetos usando o conceito de *janela temporal*, onde trajetórias são divididas em intervalos fixos de tempo, e cada intervalo é analisado para encontrar relações entre os objetos. A relação entre os objetos dentro de cada janela é identificada pela distância entre os pontos das trajetórias em cada *janela temporal*. O tempo é utilizado para assegurar uma duração mínima do padrão, mas é necessário que as trajetórias tenham o mesmo instante de captura dentro de uma *janela temporal*. Esta restrição limita o método quando trajetórias são geradas com intervalo de captura diferentes (ex: uma trajetória é coletada a cada segundo e outra a cada 5 segundos). O objetivo do trabalho de Cao não é identificar padrões de perseguição e, apesar de considerar a distância e o tempo entre trajetórias, o modo como estes atributos são analisados não é suficiente para caracterizar uma perseguição.

Este capítulo apresentou alguns conceitos básicos e alguns trabalhos relacionados. Trabalhos anteriores focaram em diversos tipos de comportamento em trajetórias. Apesar de alguns citarem o padrão de perseguição, nenhum apresentou definições, características ou algoritmos suficientes para extrair este comportamento em dados de trajetórias. Enquanto outras propostas analisam grupos de trajetórias ou trajetórias individuais, o trabalho aqui proposto visa analisar e classificar trajetórias aos pares, buscando identificar o padrão de perseguição entre elas.

3 IDENTIFICANDO PADRÕES DE PERSEGUIÇÃO ENTRE TRAJETÓRIAS DE OBJETOS MÓVEIS

Uma *perseguição* é um padrão de comportamento que envolve pelo menos duas trajetórias, a trajetória *alvo* e a trajetória *perseguidora*, onde a trajetória perseguidora persegue a trajetória alvo, conforme ilustra a figura 8. Em termos gerais:

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

Para entender melhor o padrão de perseguição, é definida uma série de conceitos que, juntos, formam a definição de perseguição em trajetórias de objetos móveis. Estas definições são apresentadas na seção 3.1.

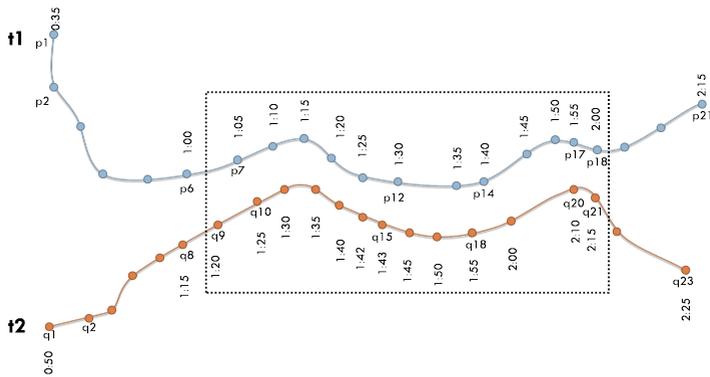


Figura 8 – Padrão de Perseguição

3.1 DEFINIÇÃO DO PADRÃO DE PERSEGUIÇÃO

Uma perseguição não ocorre entre trajetórias com intervalo de tempo muito grande entre elas. Por exemplo, uma trajetória que aconteceu ontem não irá perseguir uma trajetória gerada hoje. Para analisar o tempo, Cao (CAO; MAMOULIS; CHEUNG, 2006) utiliza o conceito de *janela temporal*, dividindo as trajetórias em sub-trajetórias de mesma duração. Por exemplo, para uma janela temporal de 1 minuto, uma

trajetória de 10 minutos será dividida em 10 sub-trajetórias de 1 minuto.

Neste trabalho, para restringir o intervalo de tempo entre duas trajetórias foi introduzido o conceito de tolerância de tempo Δt :

Uma tolerância de tempo é o intervalo máximo de tempo permitido entre duas sub-trajetórias para que possa haver uma perseguição, ou seja, as trajetórias ocorrerem em tempos similares.

A figura 9 ilustra um exemplo onde a sub-trajetória $S1$, formada por (p12, p13, p14) tem intervalo de tempo 1:30 a 1:40, enquanto a sub-trajetória $S2$, formada por (q15, q16, q17, q18) tem o intervalo de tempo 1:43 a 1:55. Para que ambas as sub-trajetórias respeitem a restrição temporal, a tolerância de tempo Δt deve ser pelo menos 15 segundos (maior diferença entre o tempo inicial de $S1$ com $S2$ e o tempo final de $S1$ com $S2$). Quando duas sub-trajetórias respeitam esta tolerância de tempo, elas formam juntas um *candidato a perseguição*.

Definição 5. Candidato a Perseguição Seja $S1 = \langle p_0, p_1, \dots, p_n \rangle$ e $S2 = \langle q_0, q_1, \dots, q_m \rangle$ sub-trajetórias de T1 e T2, respectivamente e Δt uma tolerância de tempo. S1 e S2 foram um *candidato a perseguição* se e somente se $|t_{p_0} - t_{q_0}| \leq \Delta t$ e $|t_{p_n} - t_{q_m}| \leq \Delta t$ e $t_{q_m} > t_{p_n}$.

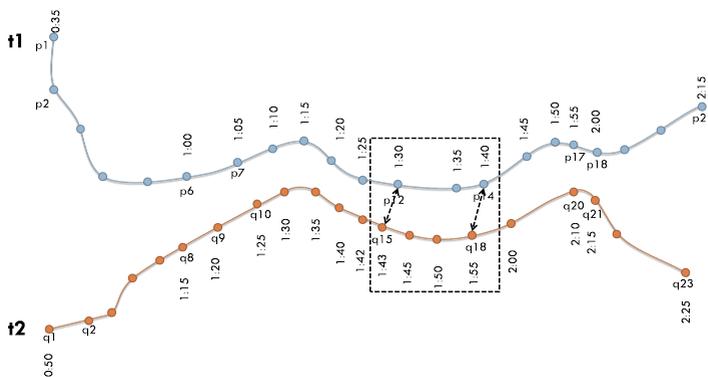


Figura 9 – Candidato a Perseguição

A definição de *candidato a perseguição* permite comparar sub-trajetórias com intervalos de tempo diferente, não se limitando a um mesmo intervalo de tempo.

Além de próximas no tempo, as trajetórias também devem estar próximas no espaço para ocorrer uma perseguição.

Mesmo que duas sub-trajetórias formem um candidato a perseguição, se a distância espacial entre elas for muito grande, não será

caracterizada uma perseguição, uma vez que em grandes distâncias uma trajetória não vê e nem está ciente da existência da outra. Sobre a distância, define-se que:

Para ocorrer uma perseguição, a distância entre as trajetórias deve respeitar uma distância máxima, onde uma trajetória consiga ver ou deduzir para onde a outra trajetória está se dirigindo.

Para verificar a distância foi criado um novo conceito chamado de *segmento de linha representativa*, conforme definição 6. Um *segmento de linha representativa* é um segmento de linha que conecta o ponto inicial ao ponto final de uma sub-trajetória, conforme ilustra a figura 10. Utilizar um Segmento de Linha Representativa tem duas vantagens principais. A primeira refere-se a otimização do algoritmo. Estudos anteriores, como (LAUBE; IMFELD; WEIBEL, 2005), (CAO; MAMOULIS; CHEUNG, 2006) e (WACHOWICZ et al., 2011), comparam a distância entre duas sub-trajetórias analisando ponto a ponto. Com um segmento de linha, é possível comparar a distância de um conjunto de pontos com uma linha. A segunda vantagem refere-se a possibilidade de comparar a distância entre trajetórias com intervalos de tempo diferente. Além de analisar ponto a ponto, estudos anteriores comparam a distância entre pontos de mesmo instante de captura. No exemplo da figura 10, o ponto p_{12} de tempo 1:30 só poderia ser comparado com o ponto q_{11} , de tempo 1:30. Como o método aqui proposto compara a distância de pontos com um segmento de linha, não é necessário que tenham o mesmo instante de captura, uma vez que a restrição temporal já foi verificada no passo anterior, podendo assim comparar a distância do ponto p_{12} ao segmento de linha (q_{15} , q_{18}), conforme ilustra a figura 10.

Definição 6. Segmento de Linha Representativa Seja uma sub-trajetória $S = \langle p_0, p_1, \dots, p_n \rangle$, o *Segmento de Linha Representativa* L de S é o segmento de linha (p_0, p_n) .

Uma vez que duas sub-trajetórias formam um *candidato a perseguição*, o próximo passo é verificar a restrição espacial para analisar se formam uma *sub-perseguição*. Uma sub-perseguição ocorre quando a média das distâncias dos pontos de uma sub-trajetória para o segmento de linha representativa de outra sub-trajetória é menor do que uma distância máxima Δd . No exemplo da figura 10, o candidato a perseguição da sub-trajetória S_1 (p_{12} , p_{13} , p_{14}) e a sub-trajetória S_2 (q_{15} , q_{16} , q_{17} , q_{18}) será uma sub-perseguição se e somente se a média da distância de todos pontos de S_1 para o segmento de Linha Representativa (q_{15} , q_{18}) for menor que Δd e a média da distância de todos os pontos de S_2 para o *segmento de linha representativa* (p_{12} , p_{14}) for menor que

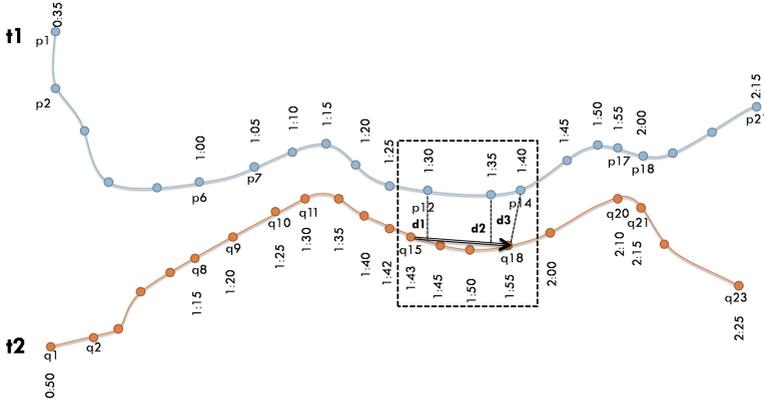


Figura 10 – Segmento de Linha Representativa (q15,q18)

Δd . Uma sub-perseguição é formalizada conforme a definição 7.

Definição 7. Sub-Perseguição Seja $S1 = \langle p0, p1, \dots, pn \rangle$ e $S2 = \langle q0, q1, \dots, qm \rangle$ um candidato a perseguição, $L1$ um segmento de linha representativa de $S1$ e $L2$ um segmento de linha representativa de $S2$. $S1$ e $S2$ formam uma sub-perseguição em relação a distância máxima Δd se e somente se $(\sum \text{distância}(p_i, L2)/n) \leq \Delta d$, onde $0 \leq i \leq n$ e $(\sum \text{distância}(q_j, L1)/m) \leq \Delta d$, onde $0 \leq j \leq m$.

É importante notar que a restrição espacial com o segmento de linha representativa é verificada nos dois lados. Isto evita casos como aquele demonstrado na figura 11, onde na figura 11(a) a sub-trajetória (q7, q8, q9, q10) está próxima do segmento de linha L1 e na figura 11(b) a sub-trajetória (p5, p6, p7, p8) está distante do segmento de linha L2. Se o método só verificasse um lado, este exemplo seria considerado uma sub-perseguição.

Uma *sub-perseguição* ocorre quando duas sub-trajetórias cumprem as restrições temporal e espacial. Contudo, isto ainda não é o suficiente para caracterizar uma *perseguição*, uma vez que este comportamento pode ser observado em um curto intervalo de tempo, caracterizando um padrão acidental ou por acaso. Para que seja observada uma perseguição de fato, este comportamento deve ocorrer por um período de tempo mínimo Δc , formando um padrão de perseguição, conforme a definição 8. Em termos gerais:

Uma perseguição é um conjunto de sub-perseguições, onde a soma da duração de todas sub-perseguições é maior que uma duração mínima.

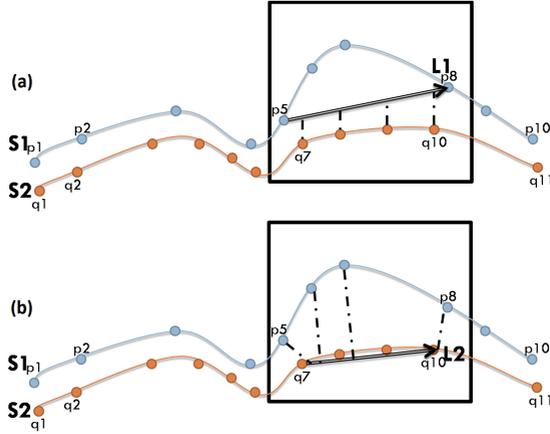


Figura 11 – (a) Sub-trajetória $(q7, \dots, q10)$ está próxima de $L1$ (b) Sub-trajetória $(p5, \dots, p8)$ está distante de $L2$

No exemplo da figura 12, as sub-perseguições $sub1$, $sub2$, $sub3$ e $sub4$ tem duração de 3 minutos, 7 minutos, 6 minutos e 8 minutos respectivamente. Para um Δc de 20 minutos, o conjunto destas sub-perseguições formará um padrão de perseguição de 24 minutos.

Assim, um padrão de perseguição pode ser definido conforme a definição 8.

Definição 8. Perseguição (PTC) Um padrão de perseguição C é um conjunto (cid, A, B) onde cid (chasing identifier) é um par (tid_{T1}, tid_{T2}) que identifica a perseguição entre uma trajetória $T1$ chamada *alvo* que está sendo perseguida por uma trajetória $T2$, chamada *perseguidora*, A é um conjunto de sub-perseguições entre $T1$ e $T2$ e B é um par $(b1, b2)$ onde $b1$ é o tempo inicial da primeira sub-perseguição em A , $b2$ o tempo final da última sub-perseguição em A e $(b2 - b1) \geq \Delta c$, onde Δc é uma duração mínima.

Esta definição de perseguição é chamada de Perseguição Pura (Pure Trajectory Chasing - PTC). Dependendo da aplicação, outros fatores podem ser interessantes para identificar uma perseguição como, por exemplo, se a velocidade média das trajetórias é similar, de acordo com a definição 9.

Definição 9. Velocidade Média Similar Seja $S1$ e $S2$ um candidato a perseguição, $\Delta a1$ e $\Delta a2$ a velocidade média de $S1$ e $S2$, respectivamente, e α percentual máximo da diferença entre as ve-

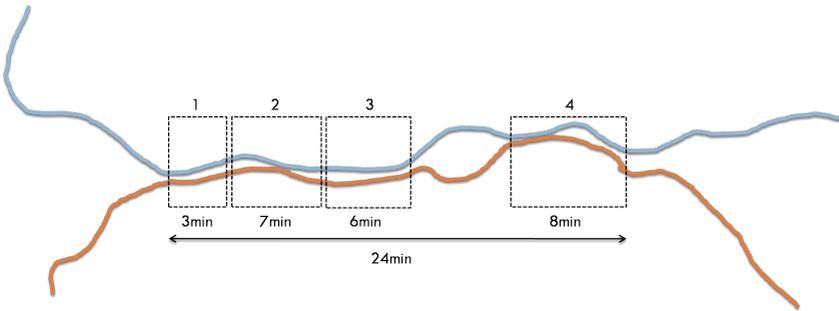


Figura 12 – Sub-Perseguições 1, 2, 3 e 4 foram uma Perseguição de 24 minutos

locidades, ambas as sub-trajetórias terão *velocidade média* similar se $(1 - \alpha) \geq (\Delta a1/\Delta a2) \geq (1 + \alpha)$ com $\alpha \in [0,1]$.

Se a velocidade de duas trajetórias é muito distinta, elas irão se distanciar com o tempo. Há exceções como, por exemplo, um perseguidor esperando o alvo mover até um certo ponto, e então se mover rapidamente para continuar vendo o alvo. Neste caso, a velocidade média das trajetórias pode ser distinta, mas ainda caracteriza uma perseguição.

Uma Perseguição com Velocidade (Speed Trajectory Chasing - STC) é definida conforme a definição 10.

Definição 10. Perseguição com Velocidade (STC) Um padrão de perseguição com velocidade é um padrão de perseguição C onde a velocidade média de $T1$ é similar à velocidade média de $T2$ durante a perseguição C .

Nesta seção foram apresentadas as definições básicas para identificar um padrão de perseguição entre trajetórias de objetos móveis. A seguir são apresentados conceitos para caracterizar diferentes tipos de perseguição.

3.2 TIPOS DE PERSEGUIÇÃO

Um padrão de perseguição entre trajetórias pode apresentar características diferentes. Com as definições até aqui apresentadas, é difícil compreender o objetivo de uma perseguição, uma vez que o padrão encontrado é genérico. Uma perseguição genérica também pode

resultar em falsos positivos, onde um padrão que ocorreu por acaso é identificado como perseguição.

Com o objetivo de enriquecer semanticamente o padrão de perseguição e melhorar a confiança do mesmo, é introduzido o conceito de tipo de perseguição, que é baseado na análise do comportamento das trajetórias durante e após a perseguição. Os tipos definidos são: *detetive*, *captura*, *assalto* e *caçada*. O tipo de perseguição é definido analisando os stops das trajetórias, que podem ser calculados pelo método *CB-SMoT* (PALMA et al., 2008). O método *CB-SMoT* encontra como stops as sub-trajetórias de baixa velocidade em trajetórias individuais. Para identificar os tipos de perseguição, são analisadas as partes das trajetórias onde elas se encontram praticamente paradas. Sendo assim, o método *CB-SMoT* é o mais apropriado entre os métodos que encontram stops em trajetórias para o trabalho aqui proposto.

Em uma perseguição, o perseguidor pode alcançar o alvo ou se manter a uma certa distância. No segundo caso, o perseguidor pode agir como um espião ou detetive, perseguindo o alvo para saber onde ele está se dirigindo. Um espião vai estar sempre atrás do seu alvo, seguindo seus passos com uma certa distância, evitando ser visto. Quando o alvo para, o espião deve parar também, mas em lugar diferente do alvo, para evitar contato, conforme demonstra a figura 13. Na figura 13, a trajetória *t2* está perseguindo a trajetória *t1* e, quando *t1* parou no instante de tempo 15:50, *t2* também parou em 15:50, mas em lugar diferente. No tipo de padrão de perseguição de *detetive*, as trajetórias *não* se encontram no mesmo local no mesmo tempo, ou seja, os stops de ambas as trajetórias ocorrem em regiões (locais) diferentes. A definição 11 formaliza este comportamento.

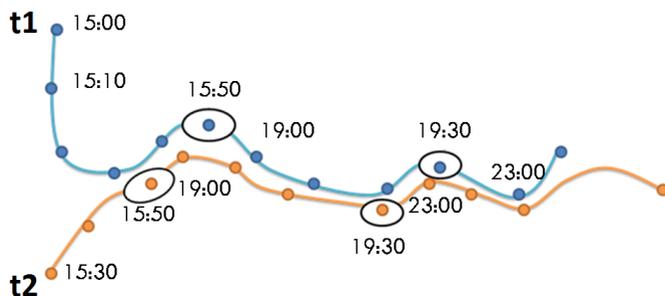


Figura 13 – Tipo de perseguição de detetive

Definição 11. Detetive Seja C um padrão de perseguição (cid, A, B_C) entre as trajetórias $T1$ e $T2$, $O1$ e $O2$ o conjunto de stops de $T1$ e $T2$, respectivamente, uma perseguição é do tipo *detetive* se para cada $o_i = (G_{oi}, B_{oi}) \in O1$ onde $B_{oi} \cap B_C \neq \emptyset$, exista um $o_j = (G_{oj}, B_{oj}) \in O2$ onde $B_{oj} \cap B_{oi} \neq \emptyset$ e $B_{oj} \cap B_C \neq \emptyset$ e $(G_{oi} \cap G_{oj}) = \emptyset$.

Para os demais tipos de perseguição, as trajetórias irão se encontrar no mesmo espaço em algum momento. Para facilitar a análise de encontro entre trajetórias, foi definido formalmente um *encontro* na definição 12. Um encontro acontece quando duas trajetórias permanecem paradas em um mesmo local no espaço e no mesmo tempo, ou seja, os stops das duas trajetórias se intersectam no espaço e no tempo.

Definição 12. Encontro Seja C um padrão de perseguição (cid, A, B_C) entre as trajetórias $T1$ e $T2$, $o_i = (G_{oi}, B_{oi})$ e $o_j = (G_{oj}, B_{oj})$ stops de $T1$ e $T2$, respectivamente, um *encontro* entre $T1$ e $T2$ ocorre quando $B_{oi} \cap B_C \neq \emptyset$ e $B_{oj} \cap B_C \neq \emptyset$ e $B_{oi} \cap B_{oj} \neq \emptyset$ e $G_{oi} \cap G_{oj} \neq \emptyset$.

Quando duas trajetórias se encontram durante uma perseguição elas podem apresentar três comportamentos diferentes. O primeiro comportamento é o de *captura*, conforme definição 13. Uma captura acontece quando, a partir do primeiro encontro entre as trajetórias, qualquer outro stop consequente terá um encontro, pois ambas as trajetórias estarão andando juntas, como ilustra a figura 14. Por exemplo, a polícia perseguindo um fugitivo. Após a captura, ambas as trajetórias terão stops iguais, no mesmo tempo e espaço.

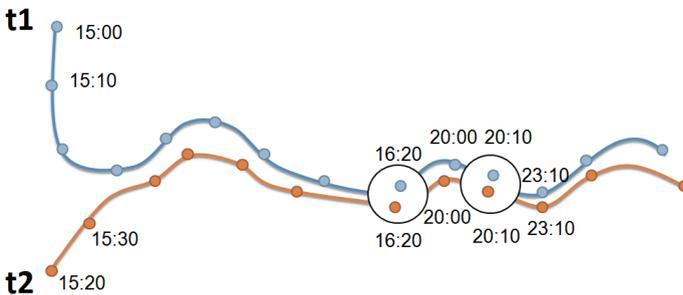


Figura 14 – Tipo de perseguição de captura

Definição 13. Captura Seja C um padrão de perseguição entre as trajetórias $T1$ e $T2$, $O1$ e $O2$ o conjunto de stops de $T1$ e $T2$, respectivamente, uma perseguição é do tipo *captura* se para o primeiro

$o_i = (G_{oi}, B_{oi}) \in O1$ que tenha um *encontro* com um $o_j = (G_{oj}, B_{oj}) \in O2$, para cada $o_{i+1}, o_{i+2}, \dots, o_{i+k}$ existir um encontro respectivo com $o_{j+1}, o_{j+2}, \dots, o_{j+k}$.

Um outro tipo de comportamento possível é o de *assalto*, segundo a definição 14. Em um *assalto*, o criminoso persegue sua vítima e, após se encontrarem, as duas trajetórias irão se distanciar no tempo, como demonstra a figura 15. Depois de um *assalto* o criminoso procura fugir e se afastar da vítima e do local do assalto o mais rápido possível. Como as trajetórias irão se afastar, não há mais perseguição, ou seja, o assalto ocorre no final do padrão de perseguição. Foi considerado um valor arbitrário de 5 vezes o valor de Δd , para que as trajetórias estejam distantes e se afastando. O valor de 5 vezes Δd foi julgado adequado para considerar que as trajetórias se afastaram.

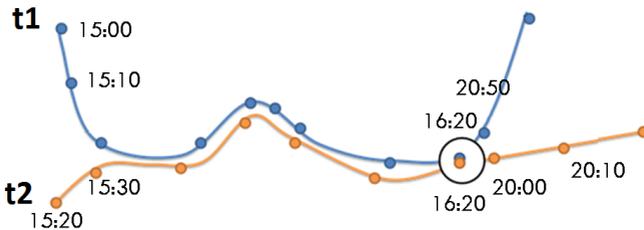


Figura 15 – Tipo de perseguição de assalto

Definição 14. Assalto Seja C um padrão de perseguição entre as trajetórias $T1$ e $T2$, $O1$ e $O2$ o conjunto de stops (G,B) de $T1$ e $T2$, respectivamente, uma perseguição é do tipo *assalto* se e somente se existir um $o_i = (G_{oi}, B_{oi}) \in O1$ e um $o_j = (G_{oj}, B_{oj}) \in O2$, onde o_i encontra o_j e C termina em $b2_{oi}$ ou $b2_{oj}$ e $T1$ e $T2$ se afastem até uma distância maior que $5 * \Delta d$, onde Δd é a distância mínima e 5 é um valor arbitrário.

O último tipo de perseguição é o de *caça*, segundo a definição 15. A principal característica do padrão de caça é que a trajetória alvo da perseguição termina juntamente com o padrão. Por exemplo, um leão persegue um cervo e, quando acontece um encontro, o leão mata o alvo. A trajetória do leão continuará, enquanto a do cervo terminará no local do *encontro*. Um assassinato também ocorre de maneira similar. Este comportamento está ilustrado na figura 16.

Definição 15. Caça Seja C um padrão de perseguição entre as trajetórias $T1$ e $T2$, $O1$ e $O2$ o conjunto de stops (G,B) de $T1$ e $T2$, res-

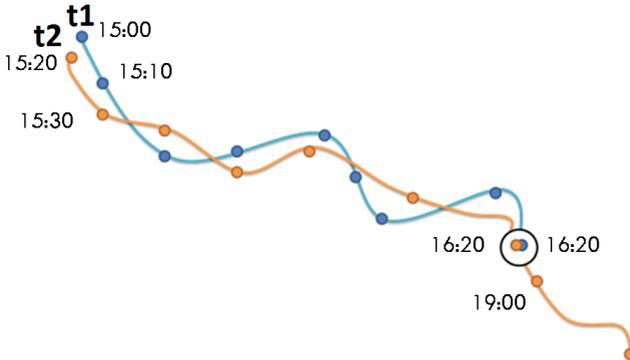


Figura 16 – Tipo de perseguição de caça

pectivamente, uma perseguição é do tipo *caça* se e somente se existir um $o_i = (G_{oi}, B_{oi}) \in O1$ e um $o_j = (G_{oj}, B_{oj}) \in O2$ onde o_i encontra o_j e a trajetória $T1$ termina em $b2_{oi}$.

Tabela 1 – Resumo dos Tipos de Padrão de Perseguição

Tipo	Encontro	Comportamento das trajetórias
Detetive	Não	Perseguidor para ao mesmo tempo que o alvo e tenta não ser visto
Captura	Sim	Trajetoárias andam juntas após encontro
Assalto	Sim	Trajetoárias se distanciam após encontro
Caça	Sim	Trajetoária do alvo termina após encontro

Nesta seção foram apresentados e definidos formalmente diferentes tipos de perseguição. A tabela 1 resume as principais características dos tipos de perseguição apresentados. Tipos de perseguição aumentam a confiança e adicionam semântica ao padrão de perseguição.

3.3 UM ALGORITMO PARA IDENTIFICAR PADRÕES DE PERSEGUIÇÃO

As definições apresentadas na seção anterior são implementadas no algoritmo TRA-CHASE (Trajectory Chasing - algoritmo 3.1). Em resumo, o algoritmo procura por padrões de perseguição em cada par de trajetórias e, em seguida, define o tipo daquela perseguição. A função *GETsubtrajectory* (ilustrada no algoritmo 3.2) agrupa pontos

em sub-trajetórias, o método SUB-CHASE (ilustrado no algoritmo 3.3), identifica perseguições em sub-trajetórias e o método CLASS-CHASE (detalhado no algoritmo 3.4) identifica o tipo da perseguição.

Algoritmo 3.1 – TRA-CHASE

```

1  Entrada:
2  T: grupo de trajetórias
3   $\Delta c$ : duração mínima
4   $\Delta t$ : tempo de tolerância
5   $\Delta d$ : distância máxima
6  velocidade: com ou sem velocidade
7
8  Saída:
9  chasingSet: um grupo de padrões de perseguição
10
11 Método:
12 PARA CADA(t1 em T)
13 PARA CADA(t2 em T)
14 SE(t1.tid  $\neq$  t2.tid)
15     C = novo CHASINGPATTERN;
16     ENQUANTO(t1 tem próximo ponto)
17     P1 = (t1.pontoAtual, t1.próximoPonto);
18     ENQUANTO(t2 tem próximo ponto)
19     P2 = (t2.pontoAtual, t2.próximoPonto);
20     S1 = GETsub-trajectory(P1,t1, $\Delta t$ );
21     S2 = GETsub-trajectory(P2,t2, $\Delta t$ );
22     chase = SUB-CHASE(S1, S2,  $\Delta t$ ,  $\Delta d$ , speed);
23     SE (chase)
24         C.adiciona(S1,S2);
25         t1.moveParaTempo(S1.tempoFinal);
26         t2.moveParaTempo(S2.tempoFinal);
27     SENÃO
28         chase = SUB-CHASE(P1, P2,  $\Delta t$ ,  $\Delta d$ , speed);
29         SE (chase)
30             C.adiciona(P1,P2);
31             t1.movaParaPróximoPonto;
32             t2.movaParaPróximoPonto;
33         SENÃO
34             t2.movaParaPróximoPonto;
35     FIM ENQUANTO
36     t1.movaParaPróximoPonto;
37 FIM ENQUANTO
38 SE C.Duração  $\geq$   $\Delta c$ 
39     C.tipo = CLASS-CHASE(C);
40     chasingSet.adiciona(C);
41 FIM SE
42 FIM SE
43 FIM PARA CADA
44 FIM PARA CADA
45 PARA CADA(C em chasingSet)
46     SE (C.countChasing > 2)
47         chasingSet.remove(C)
48     FIM SE
49 FIM PARA CADA
50 RETORNE chasingSet;
51 FIM MÉTODO

```

O algoritmo TRA-CHASE tem como entrada o conjunto de trajetórias T de objetos diferentes, a duração mínima do padrão de perse-

guição Δc , o tempo de tolerância máximo Δt , a distância máxima Δd e um booleano *velocidade*, que indica se o algoritmo vai considerar a velocidade média entre as trajetórias ou não.

Conforme ilustra o algoritmo 3.1, para cada par de trajetórias diferentes (linhas 12, 13, 14), o algoritmo analisa as sub-trajetórias (linhas 22 e 28) para verificar se apresentam um padrão de perseguição. O algoritmo então move para as próximas sub-trajetórias até cobrir toda a trajetória.

O primeiro passo é criar as primeiras sub-trajetórias $P1$ e $P2$ (linhas 17 e 19), com os dois pontos iniciais das trajetórias $t1$ e $t2$, respectivamente. Em seguida, são geradas as sub-trajetórias $S1$ e $S2$ (linhas 20 e 21) com o método *GETsub-trajectory* (algoritmo 3.2) que a partir deste momento gera sub-trajetórias considerando a tolerância de tempo Δt .

Uma vez geradas as sub-trajetórias, o próximo passo é identificar o comportamento de perseguição entre as duas sub-trajetórias $S1$ e $S2$ com o método SUB-CHASE (linha 22). Se uma sub-perseguição é encontrada entre as sub-trajetórias, ambas $S1$ e $S2$ são adicionadas a lista de sub-perseguições C (linha 24), e o algoritmo passa para o último ponto das sub-trajetórias $S1$ e $S2$ (linhas 25 e 26).

Caso não exista uma sub-perseguição entre $S1$ e $S2$, então o algoritmo procura pelo padrão, analisando ponto a ponto das sub-trajetórias $P1$ e $P2$ (linha 28). Se existe um padrão entre $P1$ e $P2$ (linhas 30), ambas as sub-trajetórias são adicionadas a C (linha 30), e o algoritmo passa para o próximo ponto de $t1$ e $t2$ (linhas 31 e 32) e recomeça o processo. Caso nenhum padrão de perseguição seja encontrado entre $P1$ e $P2$, o algoritmo passa para o próximo ponto de $t2$ (linha 34), procurando por uma sub-trajetória de $t2$ que possa estar perseguindo a sub-trajetória atual de $t1$. Por fim, se nenhuma sub-trajetória de $t2$ tem um comportamento de perseguição com a atual sub-trajetória de $t1$, o algoritmo passa para o próximo ponto de $t1$ (linha 36) e recomeça o processo.

Finalmente, se a duração do conjunto de sub-perseguições C for maior que a duração mínima Δc (linha 38), o algoritmo define o tipo de padrão de perseguição C (linha 39) e adiciona C ao conjunto de padrões de perseguição *chasingSet* (linha 40). Um padrão de perseguição é composto por duas listas de pontos, a primeira sendo os pontos do alvo que estão sendo seguidos e a segunda os pontos do perseguidor que está perseguindo um alvo.

Uma perseguição identificada pode não significar uma perseguição real, mas uma coincidência. Por exemplo, vários carros em uma

rodovia poderiam gerar uma perseguição. Normalmente, uma perseguição é um padrão de comportamento entre apenas dois objetos, logo, dificilmente irão existir situações em que vários objetos perseguem um alvo ao mesmo tempo. Então, após identificar todas as perseguições e seus tipos, o algoritmo considera uma perseguição de fato apenas quando um alvo é perseguido em um mesmo período de tempo por até dois perseguidores (linha 46). Múltiplos perseguidores descaracterizam o padrão de perseguição, sendo mais similar a um padrão de flock, por isso, quando mais de dois perseguidores seguem um alvo, estas perseguições são removidas do conjunto.

Algoritmo 3.2 – GETsub-trajectory

```

1
2  Entrada:
3  P: sub-trajetória
4  t: trajetória
5   $\Delta t$ : tolerância de tempo
6
7  Saída:
8  S: sub-trajetória
9
10 Método:
11 q = P.retorneÚltimoPonto;
12 S.adicione(P);
13 t.movaParaPonto(q+1);
14 PARA CADA (Ponto p de t)
15     SE (p.tempo < q.tempo + ( $\Delta t/2$ ))
16         S.adicione(p)
17     SENÃO
18         BREAK;
19     FIM SE
20 FIM PARA CADA
21 RETORNE S;
22 FIM MÉTODO

```

O algoritmo 3.2 apresenta o pseudo-código do método de geração de sub-trajetórias. Este método tem como entrada um par de pontos P de uma trajetória t e o tempo de tolerância Δt . A saída é uma sub-trajetória S . Primeiramente, o método localiza o último ponto de P (linha 11) denominado q . Então o método parte do ponto seguinte de t com o mesmo instante de tempo de q (linha 13) e verifica se o tempo de captura de cada ponto consecutivo é menor do que o tempo de captura de q mais $\Delta t/2$ (linha 15). Por exemplo, se q tem tempo 5:00, então o método agrupa pontos de t a partir do instante de tempo de $q + 1$. É utilizado $\Delta t/2$ para evitar mais parâmetros para o método, sendo que metade da tolerância de tempo gerou boas sub-trajetórias. O método agrupa os pontos por tempo para solucionar o problema de trajetórias diferentes serem geradas com diferentes intervalos de captura (1 em 1 segundo, 5 em 5 segundos, etc). Gerar sub-trajetórias

agrupando por número de pontos, ao invés de agrupar por tempo de captura do ponto, pode gerar sub-trajetórias com tempos que não caracterizam perseguição. Dessa forma, o uso do tempo para geração de sub-trajetórias otimiza o algoritmo.

Algoritmo 3.3 – SUB-CHASE

```

1  Entrada:
2  L1: sub-trajetória DO alvo
3  L2: sub-trajetória DO perseguidor
4   $\Delta t$ : tempo de tolerância
5   $\Delta d$ : distância máxima
6  velocidade: com ou sem velocidade
7
8  Saída:
9  chase: se L2 está perseguindo L1
10
11 Método:
12 chase = falso;
13 SE CANDIDATOAPERSEGUIÇÃO(L1, L2,  $\Delta t$ )
14   SE ESTÁPERTO(L1, L2,  $\Delta d$ )
15   SE ESTÁNAFRENTE(L1, L2)
16     SE velocidade
17     SE MESMAVELOCIDADEMEDIA(L1, L2)
18       chase = verdadeiro
19     FIM SE
20   SENÃO
21     chase = verdadeiro
22   FIM SENÃO
23 FIM SE
24 RETORNE chase
25 FIM MÉTODO

```

Uma sub-perseguição é calculada com o método SUB-CHASE, detalhado no algoritmo 3.3. Primeiramente, é verificada a restrição temporal de acordo com a definição 5, para certificar que as duas sub-trajetórias ocorreram no mesmo período de tempo. O método testa se ambas as sub-trajetórias formam um candidato a perseguição (linha 13). Neste passo é verificado se a diferença entre o tempo inicial da sub-trajetória $S2$ e o tempo inicial da sub-trajetória $S1$ e a diferença entre o tempo final de $S2$ e o tempo final de $S1$ é menor que o tempo de tolerância Δt e o tempo final de $S2$ é maior que o tempo final de $S1$. Isto significa que a sub-trajetória do perseguidor deve ter ocorrido em um tempo posterior ao da sub-trajetória do alvo. Por exemplo, se o alvo passa por um local X no tempo $time1$, o perseguidor deve passar perto do local X em um tempo $time2$ onde $time2 \geq time1$.

Apenas se $S1$ e $S2$ formarem um candidato a perseguição é que o método segue para o próximo passo, onde analisa a distância entre as sub-trajetórias de acordo com a definição 7 (linha 14), verificando se o candidato a perseguição é uma sub-perseguição. O perseguidor não precisa agir como o alvo, mas ele sempre tenta ficar próximo, podendo

mudar seu comportamento durante a sub-trajetória, aproximando-se ou afastando-se do alvo. Por isso, o método utiliza a distância média para avaliar se os objetos permanecem próximos usando a definição de segmento de linha representativa (definição 6). Se a distância média é menor do que Δd , então as duas sub-trajetórias estão próximas, formando uma sub-perseguição.

Outro fator importante é a ordem dos objetos no espaço. Normalmente, em uma perseguição, o alvo estará a frente do perseguidor, pois o perseguidor deve ver e saber para onde o alvo está se movendo. Este passo é verificado na linha 15 do método SUB-CHASE (algoritmo 3.3).

Algoritmo 3.4 – CLASS-CHASE

```

1  Entrada :
2  C: perseguição entre trajetórias
3
4  Saída:
5  tipo: tipo da perseguição
6
7  Método:
8  t1 = C.alvo;
9  t2 = C.perseguidor;
10 O1= CBSMoT(t1);
11 O2= CBSMoT(t2);
12 PARA CADA o1 ∈ O1
13   PARA CADA o2 ∈ O2
14     SE (verificaIntervaloTempo(o1,o2))
15       SE (o1.nome = o2.nome)
16         SE(andamJuntas(t1.depoisDeo1 , t2.depoisDeo2))
17           tipo = captura;
18         SENÃO SE(distanciam(t1.depoisDeo1 , t2.depoisDeo2))
19           tipo = assalto;
20         SENÃO SE(terminaDepois(t1 ,o1))
21           tipo = caça;
22         SENÃO SE(stopDurantePerseguiacao(o1 , C) e
23           stopDurantePerseguiacao(o2 , C) )
24           tipo = detetive;
25   FIM PARA CADA
26 FIM PARA CADA
27 RETORNE tipo;
28 FIM MÉTODO

```

A última propriedade a ser analisada é a velocidade (linha 17). Em alguns casos, mesmo que duas trajetórias movam juntas por um certo período de tempo, elas devem ter a mesma velocidade média. Se o alvo estiver movendo mais rápido que o perseguidor e o alvo estiver se afastando, quer dizer que o perseguidor não tinha interesse em perseguir o alvo, pois ele não aumentou sua velocidade. Por outro lado, se o perseguidor estiver movendo mais rápido que o alvo ele irá ultrapassar o alvo. O método então verifica se as duas sub-trajetórias tem a mesma velocidade média segundo a definição 9. Foi definido um limite máximo

de 20% de diferença entre a velocidade das trajetórias. Um valor de 20% foi considerado razoável para uma trajetória estar tentando ajustar sua velocidade com a velocidade de outra trajetória.

Um fator importante é que, caso durante uma perseguição os papéis se invertam, ou seja, o alvo se torna o perseguidor e o perseguidor se torna o alvo, o algoritmo considera este comportamento como uma nova perseguição, diferente da anterior.

Após encontrar o padrão de perseguição, o passo seguinte é classificar o tipo de perseguição com o método *CLASS-CHASE*, ilustrado no algoritmo 3.4. O método *CLASS-CHASE* tem como entrada o padrão de perseguição que deve ser classificado, e tem como saída o tipo da perseguição encontrada. Primeiramente é aplicado o método *CB-SMoT* (PALMA et al., 2008) em ambas as trajetórias, do alvo e do perseguidor. Este algoritmo foi utilizado para identificar as sub-trajetórias em que os objetos permanecem imóveis ou se encontram. O algoritmo calcula os stops das trajetórias (linhas 10 e 11) e, para cada stop encontrado, é verificado se o tempo dos stops se intercepta (linha 14), ou seja, se a intersecção dos intervalos de tempo de ambos os stops é diferente de vazio. Este passo serve para ter certeza de que ambos os stops aconteceram em intervalo de tempo semelhante, evitando assim comparar stops com tempos diferentes. Importante notar que o método não compara o tempo exato dos stops e sim se eles ocorreram em *intervalos similares*.

No próximo passo, o algoritmo verifica se as trajetórias se encontraram, ou seja, se ambos os stops tem o mesmo nome (linha 15). Stops de mesmo nome significa que ocorreram na mesma região. Se após o encontro, todos os próximos stops o_i em $O1$ tiverem um encontro com os stops o_j em $O2$ respectivamente, ou seja, as trajetórias passam a andar juntas (linha 16), o padrão é definido como um padrão de perseguição de *captura*. Caso esse comportamento não tenha sido observado é feito o teste do padrão de perseguição de *assalto*, onde é verificado se as trajetórias se distanciam após o stop (linha 18). Duas trajetórias se distanciam quando andam para lados opostos ou em velocidades diferentes, mas o importante é que a distância entre as trajetórias irá aumentar.

Para o tipo *caça*, a trajetória alvo $t1$ deve ter seu fim após seu stop (linha 20), pois uma caça se caracteriza pela morte do alvo, que permanece imóvel ou desaparece depois da perseguição. Quando o par de stops analisados não tem o mesmo nome, ou seja, não estão no mesmo espaço, o algoritmo verifica se ambos os stops ocorreram durante a perseguição (linha 23), caracterizando assim um padrão de *detetive*.

Os stops estarão espacialmente próximos por terem ocorrido durante a perseguição, mas não tão próximos a ponto de ser um encontro.

Por fim, o método retorna o tipo de perseguição encontrada.

Neste capítulo foram apresentadas as definições necessárias para caracterizar o padrão de perseguição e o algoritmo TRA-CHASE, que identifica e classifica padrões de comportamento de perseguição entre trajetórias. O próximo capítulo apresenta os experimentos.

4 EXPERIMENTOS

Para avaliar o método proposto foi realizada uma série de experimentos, com quatro conjuntos de dados distintos. Dos quatro conjuntos, dois foram gerados simulando perseguições, ou seja, nestes conjuntos sabe-se exatamente onde e quando cada padrão de perseguição ocorreu. A seguir são descritos os conjuntos de dados e os experimentos realizados.

4.1 DESCRIÇÃO DOS CONJUNTOS DE DADOS

O primeiro conjunto de dados foi gerado por um grupo de pessoas andando com um aparelho de GPS na praia de Jurerê Internacional, localizada em Florianópolis, Brasil. Neste conjunto de dados, um alvo caminhou por cerca de 1 hora e um grupo de 5 possíveis perseguidores andaram em tempos diferentes, simulando o comportamento de perseguição. Este conjunto de dados possui 6 trajetórias e 6.255 pontos e são trajetórias de pedestres. Os pontos foram capturados a cada 2 segundos para o alvo e a cada 1 segundo para os perseguidores, a fim de comprovar que o algoritmo funciona em trajetórias com intervalos de captura diferentes.

O segundo conjunto de dados foi gerado para simular tipos de perseguição. Estes dados foram gerados na Universidade Federal de Santa Catarina (UFSC), conforme ilustra a figura 17. Foram gerados padrões de perseguição de vários tipos diferentes. Neste conjunto, 6 indivíduos receberam aparelhos de GPS e instruções de como deveriam se comportar como, por exemplo, caminhar, parar em certos locais, perseguir algum indivíduo, desligar o aparelho de GPS, etc. Este conjunto de dados possui 6 trajetórias e 5.881 pontos. Foram gerados três padrões de perseguição de tipos diferentes, sendo um de detetive, um de captura e um de caça.

Também foram utilizados dois conjuntos de dados (sendo um de dados reais) para comparar os padrões encontrados pelo método aqui proposto com dois outros métodos existentes na literatura. O primeiro conjunto de dados, ilustrado na figura 18, são dados gerados sinteticamente pelo KDDLab (WACHOWICZ et al., 2010) para simular padrões do tipo flock. Neste conjunto de dados vários objetos movem-se juntos por um certo período de tempo. Foi utilizado um sub-conjunto de 17 trajetórias com 1.627 pontos. O objetivo de utilizar estes dados é

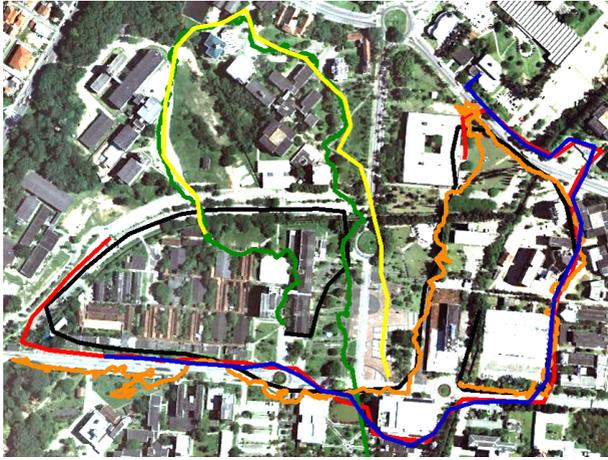


Figura 17 – Visualização do quarto conjunto de dados no Google Earth (Google Inc., 2009) em Outubro, 2011.

demonstrar que o algoritmo TRA-CHASE encontra padrões de perseguição que não são descobertos pelo algoritmo do flock, ou seja, flocks não necessariamente representam uma perseguição.

O último conjunto de dados é de uma gincana móvel desenvolvida pela Waag Society (SOCIETY, 2005) em Amsterdã. Na gincana, estudantes de 12 a 14 anos receberam um aparelho de GPS e celular, usados para responder perguntas referentes a pontos históricos georeferenciados no dispositivo. Cada estudante recebeu um mapa histórico com locais marcados e instruções para encontrá-los na cidade. Os estudantes foram divididos em seis grupos diferentes e o jogo ocorreu de 7 a 9 de fevereiro de 2005. Cada grupo era identificado por uma cor (vermelho, azul, amarelo, laranja, verde e roxo). No total, o conjunto de dados tem 466 trajetórias e 62.759 pontos. Este conjunto de dados foi utilizado para tentar encontrar padrões de perseguição entre os estudantes. Se algum estudante não conseguir responder uma pergunta, ele pode tentar seguir outro que conseguiu decifrar o problema, caracterizando assim uma perseguição.

Os experimentos foram divididos em duas seções: Experimentos de Perseguição, onde o trabalho proposto foi comparado com outras abordagens sem identificar tipos de perseguição; e experimentos de tipos de perseguição, onde foi analisado o tipo dos padrões de perseguição encontrados. O código do algoritmo foi desenvolvido em *java* na IDE

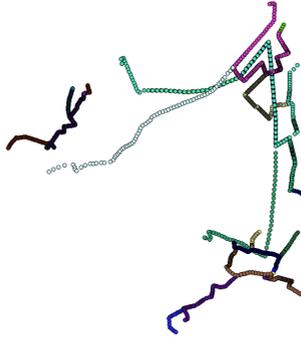


Figura 18 – Trajetórias do primeiro conjunto de dados

Eclipse (Eclipse Foundation, 2011), os dados foram armazenados no banco de dados *Postgres* com a extensão *PostGIS* (Refractions Research, 2011) e os resultados são apresentados usando os softwares abertos *Quantum GIS* (Quantum GIS Development Team, 2009) e *Google Earth* (Google Inc., 2009).

4.2 EXPERIMENTOS PARA IDENTIFICAR PADRÕES GENÉRICOS DE PERSEGUIÇÃO

Nesta seção, o algoritmo TRA-CHASE (Pure TRA-CHASE PTC e Speed TRA-CHASE STC) é comparado com dois outros algoritmos: o Moving Flock Finder (MFF) (WACHOWICZ et al., 2011), (WACHOWICZ et al., 2010) e o Collocation Discovery Algorithm (CDA) (CAO; MAMOULIS; CHEUNG, 2006). Ambos os algoritmos, MFF e CDA, encontram padrões que poderiam ser interpretados como perseguições. O algoritmo STC é similar ao PTC, mas o primeiro considera que a velocidade média entre as trajetórias deve ser similar durante a perseguição, enquanto o algoritmo PTC não considera a velocidade. O objetivo destes experimentos é comprovar que o algoritmo TRA-CHASE encontra padrões diferentes destas duas abordagens.

4.2.1 Conjunto de Dados Sintéticos

Neste primeiro experimento foi utilizado o conjunto de dados sintéticos do KDDLab (WACHOWICZ et al., 2010). Todos os quatro

Tabela 2 – Padrões encontrados pelos algoritmos com distância $\Delta d = 80.0m$, tolerância de tempo $\Delta t = 5min$ e duração mínima $\Delta c = 10min$

Padrão	PTC	STC	CDA	MFF
C1	18:23 - 19:50	18:25 - 19:50	18:27 - 19:47	18:26 - 19:45
C2	18:16 - 18:55	18:16 - 18:49	-	-
C3	18:18 - 18:35	18:18 - 18:34	-	-
C4	20:11 - 20:21	-	-	-
C5	18:18 - 18:32	18:18 - 18:28	-	-
C6	18:23 - 19:50	18:23 - 19:50	-	-

algoritmos foram executados com os mesmos parâmetros: distância $\Delta d = 80.0m$, duração mínima $\Delta c = 10min$ e tolerância de tempo $\Delta t = 5min$. Foi definido 80 metros como a distância mínima, pois a distância média entre os pontos de uma trajetória nestes dados sintéticos variava de 40 a 160 metros em um intervalo de captura de um minuto. Foi definido que o parâmetro *janela temporal* do CDA deve ter o mesmo valor que o *time tolerance* usado pelo método proposto nesta dissertação.

Como resultados, o algoritmo do flock (MFF) encontrou um padrão, o collocation (CDA) encontrou dois, PTC encontrou seis e STC encontrou cinco padrões, conforme ilustra a tabela 2. Ambos o MFF e o CDA encontraram o mesmo padrão. Uma vez que o CDA não verifica qual objeto está na frente do outro, os dois padrões encontrados pelo CDA foram idênticos. O algoritmo de perseguição PTC (sem considerar velocidade) encontrou mais padrões que os outros, uma vez que não considera a diferença de velocidade entre as trajetórias. Os padrões encontrados pelo STC também foram encontrados pelo PTC, mas eram mais curtos, pois as sub-trajetórias não tiveram a mesma velocidade média.

Com este experimento foi possível demonstrar que o algoritmo TRA-CHASE encontra padrões não identificados pelo MFF ou CDA, comprovando que as abordagens são diferentes.

4.2.2 Conjunto de Dados da Gincana de Amsterdã

Neste experimento foi utilizado o conjunto de dados reais da Waag Society (SOCIETY, 2005). O objetivo foi encontrar padrões de perseguição entre indivíduos de *grupos distintos*, assumindo que caso um aluno não saiba para onde ir, ele poderia perseguir outro estudante de outro grupo até o local a ser visitado. Neste experimento não foram

Tabela 3 – Número de padrões encontrados pelos algoritmos PTC, STC, CDA e MFF com os dados originais

Parâmetros	PTC	STC	CDA	MFF
$\Delta d=15m \Delta t= 1$ minuto	93	90	11	0
$\Delta d=15m \Delta t= 3$ minutos	139	136	4	0
$\Delta d=30m \Delta t= 1$ minuto	169	167	36	0
$\Delta d=30m \Delta t= 3$ minutos	217	215	27	0

Tabela 4 – Número de padrões encontrados pelos algoritmos PTC, STC, CDA e MFF com os dados sincronizados

Parâmetros	PTC	STC	CDA	MFF
$\Delta d=15m \Delta t= 1$ minuto	105	101	64	0
$\Delta d=15m \Delta t= 3$ minutos	152	148	33	0
$\Delta d=30m \Delta t= 1$ minuto	244	240	180	0
$\Delta d=30m \Delta t= 3$ minutos	297	297	72	0

consideradas perseguições entre alunos do mesmo grupo, uma vez que o objetivo era identificar o padrão dentre alunos de grupos distintos. Entre todos os conjuntos de dados, este era o maior e o mais complexo. Cada trajetória possui vários pontos com tempos de captura distintos. Por exemplo, a trajetória de um mesmo objeto poder ter dois pontos consecutivos de 1 até 60 segundos de diferença.

Já que o algoritmo MFF não trabalha com dados de tempo de captura diferente, foi utilizado um software de sincronização de (WACHOWICZ et al., 2011) para tentar encontrar os padrões de flock. Experimentos foram realizados com ambos os conjuntos de dados, o sincronizado e o original, com os parâmetros $\Delta t = [1$ minuto e 3 minutos], $\Delta d = [15$ metros e 30 metros] e $\Delta c = 10$ minutos. Os resultados estão ilustrados nas tabelas 3 e 4. Como o intervalo de tempo entre dois pontos consecutivos da trajetória de um objeto poderia variar em até 1 minuto, foram definidos valores na grandeza de minutos para a tolerância de tempo. No contexto de trajetórias de pedestres em uma cidade, uma distância de até 30 metros é uma distância aceitável para que um objeto não perca o outro de vista. Uma duração de 10 minutos é um tempo razoável para considerar que o comportamento de perseguição é real e não somente ao acaso.

Mesmo sincronizando os dados e considerando dois valores diferentes para a distância, o MFF não encontrou nenhum padrão. Como o MFF só encontra flocks em movimento, uma possível causa para não ter encontrado padrões são as paradas dos estudantes nos pontos históricos. Como as trajetórias não se moveram nestes pontos, o algoritmo

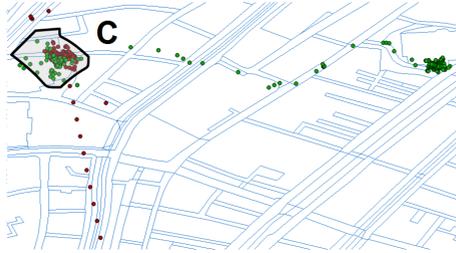


Figura 19 – Padrão encontrado pelos algoritmos PTC, STC e CDA para $\Delta d=15m$ $\Delta t= 1$ minuto

não considera o padrão de movimento como um flock. O CDA encontrou poucos padrões com os dados originais e muitos com os dados sincronizados, além de encontrar mais padrões com uma janela temporal menor. O CDA divide as trajetórias em sub-trajetórias de mesma duração. Logo, dados sincronizados irão gerar sub-trajetórias de mesmo número de pontos, fazendo com que o algoritmo identifique melhor o padrão. Ambos PTC e STC encontraram quase os mesmos padrões, mostrando que a velocidade entre as trajetórias foi similar. Como esperado e ilustrado nas tabelas 3 e 4, todos os algoritmos encontraram mais padrões com os dados sincronizados, uma vez que são uniformes e mais comportados.

Uma diferença entre os algoritmos TRA-CHASE e CDA pode ser notada analisando o parâmetro do tempo. Quanto maior o tempo, mais padrões são encontrados pelo TRA-CHASE e menos padrões são encontrados pelo CDA. Para o TRA-CHASE, um tempo maior permite comparar pontos com intervalo de captura maior entre eles. Já no CDA, um tempo maior significa uma janela temporal maior, ou seja, as trajetórias serão divididas em fatias maiores, que podem não estar próximas no espaço.

Os algoritmos PTC e STC encontraram vários padrões de perseguição. Isto ocorreu porque a maioria dos padrões encontrados ocorreram em regiões em que os estudantes pararam de se mover ou se moviam lentamente. Estas regiões representam ou os locais históricos (onde os estudantes respondiam as perguntas) ou onde os estudantes descansavam, já que a gincana durou três dias. Como os pontos nestas regiões estavam bem próximos, os algoritmos CDA, PTC e STC encontraram os mesmos padrões. Este tipo de padrão, localizado em regiões de grande concentração de pontos (regiões densas), conforme ilustra a figura 19, foi o único tipo de padrão encontrado pelo CDA.

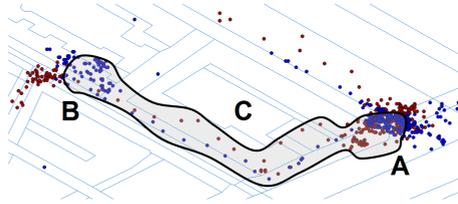


Figura 20 – Padrão encontrado pelos algoritmos PTC e STC para $\Delta d=15m$ $\Delta t= 1$ minuto

A figura 20 demonstra um tipo de padrão mais interessante que foi encontrado apenas pelos algoritmos PTC e STC, propostos nesta dissertação. Duas trajetórias moveram da região A para a região B. O TRA-CHASE identificou isto como uma perseguição C. O algoritmo CDA não encontrou padrões deste tipo.

Este experimento encontrou padrões de perseguição entre trajetórias de estudantes. Tal comportamento pode ser interpretado como um estudante que, por não saber o próximo local da pergunta, perseguiu algum estudante que sabia o próximo local. Também podem ser casos onde estudantes se reuniram para responder as perguntas em conjunto.

4.2.3 Conjunto de Dados de Jurerê

Neste experimento, foi utilizado o conjunto de dados gerado em uma caminhada na praia de Jurerê, Florianópolis. Como neste conjunto de dados sabe-se onde e quando cada perseguição acontece (inicia e termina), os resultados dos algoritmos podem ser comparados com os padrões reais. Os quatro algoritmos foram executados com os parâmetros de distância $\Delta d = 10m$, duração $\Delta c = 3min$ e tempo de tolerância $\Delta t = 30$ segundos. Em uma perseguição de pedestre, 10 metros é uma distância aceitável para um objeto manter o outro por perto, podendo visualizar o alvo enquanto mantém uma certa distância. Como o intervalo de tempo entre os pontos coletados foi de 1 e 2 segundos, com os pedestres andando em baixa velocidade, um tempo de tolerância de 30 segundos é uma boa medida para um objeto estar 10 metros atrás do outro.

Este conjunto de dados teve 6 perseguições, onde o objeto 0 foi perseguido uma vez pelos objetos 1 e 2 e duas vezes pelos objetos 3 e 5. Neste experimento, o PTC encontrou 7 padrões e o STC os 6 padrões

originais, conforme ilustra a tabela 5. O PTC encontrou um padrão a mais que o STC. O padrão adicional foi um teste realizado, onde o possível perseguidor ultrapassou o alvo correndo, com uma velocidade maior que o alvo. Este teste comprova a diferença entre o PTC e o STC quando a velocidade média é considerada. Como as trajetórias são curtas, cada padrão de perseguição também ocorreu por pouco tempo.

Ambos os algoritmos flock (MFF) e collocation (CDA) não encontraram nenhum padrão. O experimento foi então realizado, aumentando a distância de $10m$ para $\Delta d = 25m$, conforme ilustra a tabela 5. O CDA então encontrou 3 padrões e o MFF mesmo assim não encontrou nenhum padrão. Com o aumento da distância o CDA encontrou mais padrões, porém o MFF continuou não encontrando, uma vez que os dados tem intervalo de captura diferentes (1 segundo e 2 segundos).

Tabela 5 – Comparação dos padrões encontrados pelos diferentes algoritmos

Padrão	Real	PTC $\Delta d=10m$	STC $\Delta d=10m$	CDA $\Delta d=25m$	MFF $\Delta d=25m$
C1	17:12:00 17:25:40	17:09:12 17:23:27	17:10:41 17:23:19	17:07:33 17:10:14	-
C2	17:27:52 17:33:45	17:27:30 17:33:45	17:27:44 17:33:50	-	-
C3a	17:37:00 17:40:40	17:36:48 17:40:45	17:36:48 17:40:45	-	-
C3b	17:44:30 17:50:50	17:43:43 17:50:49	17:44:16 17:51:04	-	-
-	-	17:51:56 17:55:10	-	-	-
C4a	17:56:45 18:00:40	17:57:21 18:00:48	17:57:55 18:00:50	17:55:39 17:58:14	-
C4b	18:02:50 18:09:40	18:03:22 18:09:56	18:03:42 18:10:00	18:04:46 18:09:59	-

A figura 21 ilustra uma comparação do padrão real C1, o padrão gerado pelo CDA e pelo STC. Importante notar que o padrão encontrado pelo CDA foi anterior a perseguição real, enquanto o TRA-CHASE encontrou um padrão bem semelhante ao real. Para todas as outras perseguições reais, o TRA-CHASE encontrou padrões bem similares ao original, enquanto o CDA encontrou apenas dois similares (padrões C4a e C4b).

Este experimento comprova que os algoritmos CDA e MFF não são suficientes para identificar padrões de perseguição entre trajetórias, o que justifica este trabalho de pesquisa.

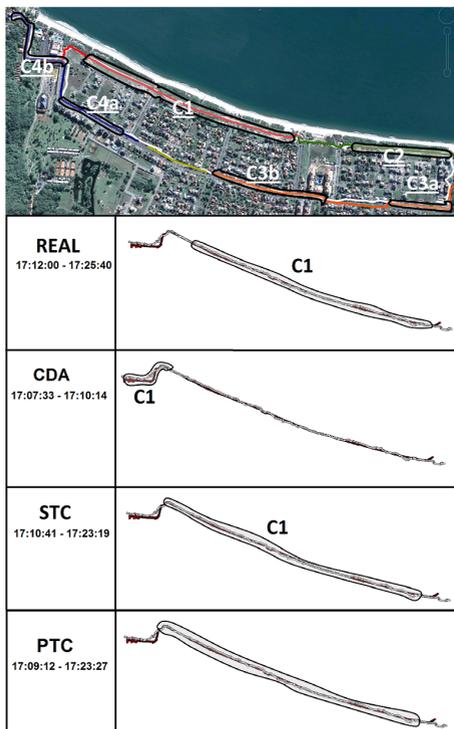


Figura 21 – Comparação entre a perseguição real, o CDA e o STC para a perseguição C1 e visualização no Google Earth (Google Inc., 2009) em Março, 2011

Para explicar a diferença entre os algoritmos, a figura 22 ilustra duas sub-trajetórias $S1$ e $S2$, exemplificando porque o CDA e o MFF não encontram padrões iguais ao TRA-CHASE. O CDA e o MFF, entre outras abordagens, comparam pontos com o mesmo instante de tempo. Assim, os pontos $p5$ e $p6$ da sub-trajetória $S1$ com o respectivo tempo $1:20$ e $1:25$ só poderiam ser comparados com os pontos $q2$ e $q3$ da sub-trajetória $S2$, com tempo $1:20$ e $1:25$ respectivamente, como ilustra a figura 22(b). Estes pontos, de distância $d1$, estão distantes entre si, não respeitando a distância máxima permitida. Por outro lado, o algoritmo TRA-CHASE, para um $\Delta t = 0:15$, compara todos os pontos entre $q2$ e $q6$ da sub-trajetória $S2$ com os pontos $p5$ e $p6$ da sub-trajetória $S1$, pois compara pontos de instante de tempo de $1:25 + \Delta t$, ou seja, $1:40$, como demonstra a figura 22(b). Verificando os pontos futuros de $S2$,

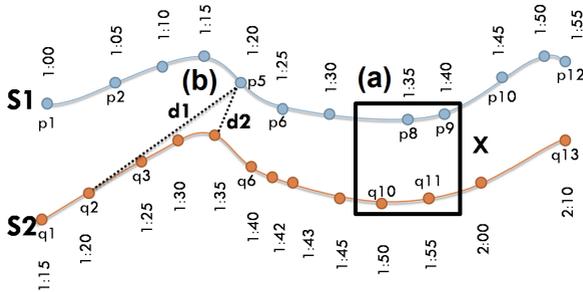


Figura 22 – (a) Pontos p8,p9 de S1 e pontos q10,q11 de S2 no mesmo local X. (b) Distância d1 analisada por MFF e CDA para o ponto p5 e a distância d2 analisada pelo TRA-CHASE no ponto p5.

as sub-trajetórias ficam mais próximas (distância $d2$).

Nesta seção o algoritmo TRA-CHASE foi comparado com outras abordagens, comprovando que ele encontra padrões ainda não descobertos por outros estudos. A seção seguinte avalia a classificação dos tipos de perseguição. Os resultados obtidos até aqui foram publicados na revista Transactions in GIS (SIQUEIRA; BOGORNY, 2011c).

4.3 EXPERIMENTOS PARA IDENTIFICAR TIPOS DE PERSEGUIÇÃO

Esta seção apresenta experimentos para validar a classificação dos tipos de perseguição usando o conjunto de dados de Amsterdã e um conjunto de dados gerados na UFSC. No segundo conjunto de dados os tipos de perseguição são conhecidos.

4.3.1 Conjunto de Dados da Gincana de Amsterdã

O objetivo deste experimento é identificar o tipo de algumas perseguições entre alunos de grupos diferentes e entre alunos do mesmo grupo. Quando um aluno perseguir um aluno de outro grupo, ele tenta não ser visto, caracterizando uma perseguição de *detetive*. Já alunos do mesmo grupo podem apresentar perseguições do tipo *captura*, onde um aluno persegue seu colega e ambos passam a andar juntos como um time.

Tabela 6 – Tipos de perseguição por grupo

Perseguição	Mesmo Grupo	Grupo Diferente	Total
Detetive	33	94	127
Captura	3	1	4
Caçada	1	0	1
Total	37	95	132

Experimentos foram realizados considerando a distância $\Delta d=15$ metros, a tolerância de tempo $\Delta t=1$ minuto e a duração da perseguição $\Delta c=10$ minutos para o algoritmo PTC. Diferentemente do experimento anterior, que foi realizado apenas com alunos de grupos diferentes, desta vez foram analisadas perseguições entre todos os alunos. O resultado é apresentado na tabela 6.

Foram encontradas 132 perseguições no total (37 dentro do mesmo grupo e 95 entre grupos diferentes) onde todas foram classificadas. Como era de se esperar, a grande parte das perseguições foi de detetive, sendo a maioria entre grupos diferentes (94). Em grupos diferentes um aluno teria perseguido outro com a intenção de saber para onde se dirigir após uma charada. Já em alunos do mesmo grupo, ocorreram mais perseguições do tipo captura (3), onde dois alunos poderiam ter se unido para desvendar as charadas e andarem juntos depois do encontro. O padrão de perseguição de caçada encontrado se refere a um aluno que participou somente do segundo dia de gincana e por um curto período de tempo (por volta de 2 horas). Após ser perseguido por um outro aluno do mesmo time, sua trajetória acabou, podendo significar que ele desistiu da gincana.

Não há como validar cada tipo de perseguição, porém os resultados obtidos refletem o esperado, onde as trajetórias de alunos de grupos distintos caracterizam perseguição de *detetive*, enquanto trajetórias de alunos do mesmo grupo apresentam padrões de perseguição de *captura* (se encontram e passam a andar juntos).

4.3.2 Conjunto de Dados da UFSC

Este experimento foi realizado com os dados gerados na UFSC, buscando simular tipos específicos de perseguição, sabendo-se onde e quando a perseguição inicia e termina, bem como o seu tipo. Neste conjunto de dados é possível comprovar se o algoritmo identifica as perseguições e as classifica corretamente. Os resultados do experimento estão expressos na tabela 7.

Tabela 7 – Padrões encontrados para o conjunto de dados da UFSC

ID	Alvo	Perseguidor	Tempo Real	Tempo Encontrado	Tipo
C1	T6	T5	10:45:50-11:11:10	10:45:32-11:11:38	Captura
C2	T4	T3	11:19:20-11:30:20	11:18:56-11:31:08	Caçada
C3	T2	T1	19:26:50-19:48:21	19:27:03-19:48:09	Detetive

Foram encontradas 3 perseguições, sendo uma do tipo *detetive*, uma do tipo *captura* e uma do tipo *caçada*, todas identificadas corretamente de acordo com o padrão real, conforme ilustra a tabela 7.

A figura 23 ilustra o processo de descoberta e classificação do padrão C3. Primeiramente, o padrão de perseguição C3 foi identificado, onde a trajetória T1 persegue a trajetória alvo T2.

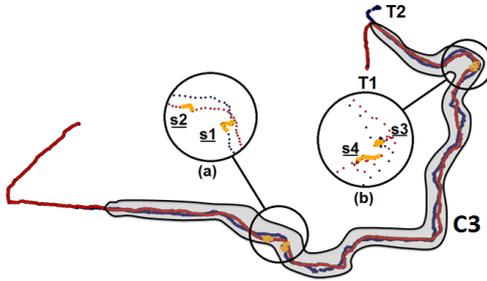


Figura 23 – Stops encontrados na perseguição C3 e sua localização nas trajetórias da UFSC.

Tabela 8 – Stops encontrados pelo método CB-SMoT nas trajetórias T1, T2, T3 e T4

ID Stop	ID Trajetória	Nome	Início	Final
s1	T2	0unknown	19:30:07	19:34:09
s2	T1	1unknown	19:30:21	19:34:23
s3	T2	2unknown	19:42:01	19:44:01
s4	T1	3unknown	19:42:33	19:44:33
s5	T4	4unknown	11:28:03	11:30:05
s6	T3	4unknown	11:27:50	11:31:36

O segundo passo foi classificar o tipo de perseguição. O método CB-SMoT, utilizado para identificar encontros (stops), gerou 6 stops, 2 na trajetória T1, 2 na trajetória T2, 1 na trajetória T3 e 1 na trajetória T4, como ilustra a tabela 8. Pelo nome dos stops pode-se verificar

que os mesmos aconteceram em locais distintos. A figura 23(a) e (b) ilustra espacialmente onde ocorreram os stops. De acordo com a tabela 8, a trajetória $T2$ teve um stop $s1$ durante o tempo 19:30:07 e 19:34:09. A trajetória $T1$ teve um stop $s2$ durante o tempo 19:30:21 e 19:34:23. Os stops $s1$ e $s2$ tem *nomes diferentes* na tabela, indicando que aconteceram em locais distintos, mas em intervalos de tempo semelhantes. Essa mesma característica foi observada entre os stops $s3$ e $s4$, caracterizando uma perseguição do tipo detetive.

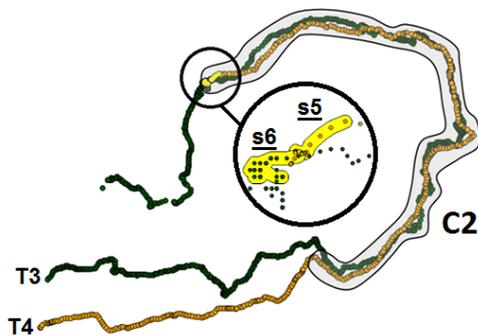


Figura 24 – Stops encontrados na perseguição C2 e sua localização nas trajetórias.

A perseguição $C2$ foi classificada como tipo *caçada*. A figura 24 ilustra o padrão de perseguição $C2$, encontrado entre a trajetória alvo $T4$ e a perseguidora $T3$, e seus respectivos stops $s5$ e $s6$. Como os stops se intersectam, o método CB-SMoT identifica ambos com o mesmo nome, como demonstra a tabela 8. De acordo com a definição 15, uma perseguição é do tipo *caçada* quando a trajetória alvo tem seu término logo após o encontro com seu perseguidor. Segundo a tabela 8, o stop do alvo $T4$ terminou em 11:30:05 e a trajetória $T4$ tem seu último ponto no tempo 11:30:53. Como o final do padrão pode não terminar exatamente no final da trajetória, é considerada uma tolerância mínima de 60 segundos para que a trajetória alvo termine depois de seu stop. Sendo assim, a trajetória $T4$ ainda está dentro deste limite, caracterizando uma perseguição do tipo *caçada*. Neste conjunto de dados, onde o intervalo de captura dos pontos é a cada 1 ou 2 segundos, 60 segundos é um tempo suficiente de tolerância para eventuais erros de precisão e ruídos.

4.4 ANÁLISE DA PERFORMANCE

Uma preocupação com algoritmos de mineração de dados é o tempo de processamento. Por processar grandes quantidades de dados, é necessário analisar o impacto de diversos fatores no tempo de processamento do algoritmo. Para analisar a performance foi calculado o tempo médio gasto pelo algoritmo TRA-CHASE no conjunto de dados de Jurerê, Amsterdã e caminhada na UFSC para os mesmos parâmetros de distância, tolerância de tempo e duração. Os resultados estão ilustrados na tabela 9.

Tabela 9 – Tempo de processamento em milissegundos para cada conjunto de dados

Conjunto de Dados	Trajetórias	Pontos	Tempo
UFSC	6	5881	13472
Jurerê	6	6255	10113
Amsterdã	466	62759	4917668

O tempo de processamento foi maior para os dados de Amsterdã devido ao grande número de pontos. O número de trajetórias também impactam no tempo, uma vez que o algoritmo processa pares de trajetórias. Uma trajetória a mais significa $n-1$ interações a mais, onde n é o número de trajetórias.

O parâmetro de tolerância de tempo também influencia no tempo de processamento. Uma tolerância de tempo maior irá criar sub-trajetórias maiores, podendo otimizar o algoritmo comparando vários pontos de uma só vez. Já uma distância maior irá permitir que trajetórias mais distantes possam apresentar comportamento de perseguição, aumentando o tempo de processamento.

Uma otimização implementada no algoritmo foi em relação ao intervalo de tempo das trajetórias. Duas trajetórias que não se intersectam no tempo não podem apresentar comportamento de perseguição entre elas. A definição 5 de candidato a perseguição se aplica apenas para sub-trajetórias, e não para a trajetória completa. Para evitar desperdício de tempo, uma vez selecionadas duas trajetórias, o algoritmo verifica se seus intervalos de tempo se cruzam, ou seja, se uma trajetória começa antes da outra terminar. Caso não se cruzem, não é possível existir uma perseguição e o algoritmo passa para o próximo par de trajetórias, evitando comparações desnecessárias. Caso se cruzem, o algoritmo executa normalmente.

Existem ainda alguns problemas em relação ao tempo de proces-

samento que não são possíveis de serem resolvidos por otimizações no algoritmo como, por exemplo, o banco de dados. Além de problemas com consultas SQLs, não existe suporte a indexação de dados espaço-temporais no banco de dados geográficos. Sendo assim, consultas com dados espaço-temporais consomem mais tempo de processamento do que consultas convencionais. Outro problema é em relação ao método CB-SMoT, utilizado pelo TRA-CHASE. O método CB-SMoT tem sua própria complexidade que não pode ser controlada ou alterada pelo TRA-CHASE, porém aumenta o tempo de processamento final.

4.5 ANÁLISE DOS PARÂMETROS

Um problema muito comum em algoritmos de mineração de dados é como definir os valores dos parâmetros. Testar vários valores até encontrar os melhores parâmetros é muitas vezes custoso, além de variável, dependendo da aplicação e dos dados. Para tentar minimizar o número de testes e indicar bons valores iniciais para os parâmetros, foi realizado um estudo sobre cada um dos parâmetros para avaliar o melhor valor.

A contribuição principal deste trabalho e o que difere dos demais é o modo como a dimensão do tempo (tolerância de tempo) é utilizada. Porém, justamente por ser diferente, é difícil definir um valor inicial sem entender o seu significado. O valor da tolerância de tempo deve ser baseado em quanto tempo o perseguidor leva para alcançar o alvo. Por exemplo, na figura 22(a), o alvo $S1$ entrou no local X no tempo 1:35 e o perseguidor $S2$ entrou no mesmo local no tempo 1:50. Então, a tolerância de tempo deve ser pelo menos 0:15 (a diferença de tempo de $S1$ e $S2$ no mesmo local X), permitindo que uma sub-trajetória $S1$ no tempo 1:35 possa ser comparada com uma sub-trajetória $S2$ no tempo 1:50.

A distância também está ligada à tolerância de tempo. Os métodos de collocation e flock definem a proximidade analisando a distância de pontos no mesmo tempo de captura, ou seja, comparam pontos com os mesmos instantes de tempo. Já o TRA-CHASE analisa a proximidade de duas sub-trajetórias, fixando o tempo do alvo e movendo o tempo do perseguidor para o futuro, encontrando assim, proximidade entre trajetórias de tempo de captura diferentes, o que caracteriza uma perseguição. A figura 22(b) ilustra como o TRA-CHASE e os outros métodos analisam a distância. Para o ponto $p5$ de $S1$, as outras abordagens comparam ao ponto $q2$ de $S2$ (distância $d1$), enquanto o TRA-

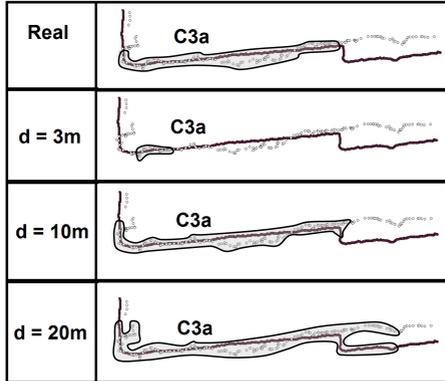


Figura 25 – Resultados para a Perseguição C3 com parâmetros $\Delta d = (3m, 10m, 20m)$ e $\Delta t = 30s$.

CHASE, compara o ponto p_5 com o ponto q_5 de S_2 , com a distância d_2 para um $\Delta t = 0:15$.

Aumentar a distância utilizada nos métodos de collocation e flock poderia ser uma opção para encontrar perseguições, mas isto geraria outro problema. Uma distância grande descaracteriza uma perseguição (o perseguidor perde o alvo de vista).

Para avaliar os parâmetros foi utilizado o conjunto de dados de Jurerê, onde se sabe como os padrões ocorreram. Neste conjunto de dados, a distância média entre as trajetórias no mesmo instante de tempo variou de 15 a 40 metros, enquanto a distância média entre trajetórias de diferentes instantes de tempo variou de 3 a 15 metros, com uma diferença de tempo de captura de 15 a 45 segundos.

O primeiro passo foi analisar o impacto da variação da distância no algoritmo, como ilustra a figura 25. Como tolerância de tempo foi considerado o tempo médio da diferença de tempo de captura de dois pontos próximos no espaço (15 a 45 segundos), correspondendo a 30 segundos. Para a distância foram considerados 3 valores: 3, 10 e 20 metros. Os valores 3 e 10 metros são coerentes com a variação da distância em instantes de tempo diferentes (3 e 15 segundos). O valor de 20 metros é um valor entre a variação de distância de mesmo instante de tempo (15 a 40 metros).

Como pode ser observado na figura 25, uma distância muito curta ($\Delta d = 3$ metros) gera padrões curtos, uma vez que em uma perseguição, dependendo do seu tipo, duas trajetórias podem não permanecer

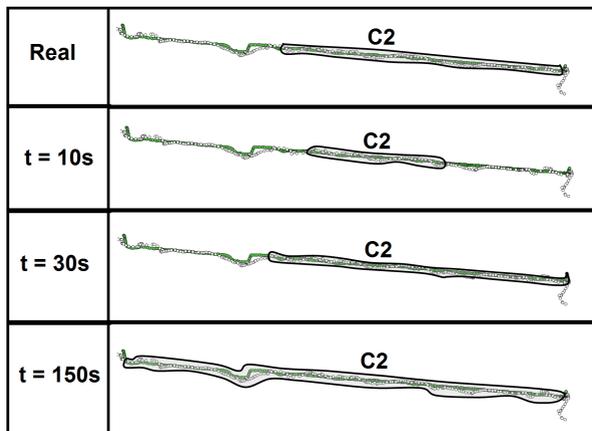


Figura 26 – Resultados para a Perseguição C2 com parâmetros $\Delta t =$ (10s, 30s, 150s) e $\Delta d = 10m$.

tão próximas. Por outro lado, considerando distância grande ($\Delta d = 20$ metros), pode gerar padrões que não caracterizam um comportamento de perseguição, pois as trajetórias estão distantes uma da outra. Utilizando um valor médio ($\Delta d = 10$ metros) os resultados foram mais condizentes com a realidade.

O segundo experimento analisou o impacto da variação da tolerância de tempo. Foram definidos 3 valores para a tolerância de tempo: 10, 30 e 150 segundos. Os valores de 10 segundos e 30 segundos são valores próximos do intervalo de tempo entre pontos de trajetórias próximas (15 a 45 segundos) e 150 segundos é um valor acima do esperado. A figura 26 ilustra o resultado. Com um tempo de tolerância muito curto ($\Delta t = 10$ segundos), o algoritmo encontra um padrão de perseguição verdadeiro, mas menor do que o real. Com uma tolerância de tempo muito alta ($\Delta t = 150$ segundos), o algoritmo encontrou perseguição durante toda a trajetória, demonstrando que um valor irreal de tolerância de tempo descaracteriza uma perseguição. O valor médio ($\Delta t = 30$ segundos) encontrou um padrão bem similar a perseguição real, comprovando ser o melhor valor.

Os valores dos parâmetros podem variar de acordo com o contexto da aplicação onde se quer identificar um comportamento de perseguição. Diferentes aplicações e diferentes dados requerem valores de parâmetros diferentes. Fatores como uma região com muitas ou poucas trajetórias podem influenciar nos parâmetros. Por exemplo, em uma

praia bem movimentada, 20 metros pode ser uma distância grande para enxergar um objeto, enquanto em uma rua deserta, é uma distância aceitável. Em termos gerais, para a tolerância de tempo, o valor mínimo deve ser maior que o intervalo de tempo de captura entre dois pontos, pois caso os pontos tenham sido capturados a cada 1 minuto, uma tolerância de tempo de 30 segundos não irá gerar nenhum padrão. O valor máximo para a tolerância de tempo deve ser baseado em quanto tempo o perseguidor levaria para ultrapassar o alvo em um certo instante de tempo, caso o alvo esteja parado.

Para a distância, o valor mínimo deve ser maior que zero, uma vez que no valor zero os objetos estão juntos, e não perseguindo. O valor máximo deve ser menor que um valor adequado para que o perseguidor ainda consiga visualizar o alvo. Por exemplo, 100 metros pode parecer um valor muito grande para que o perseguidor tenha contato visual com o alvo, mas no caso de aviões ou rastreadores pode ser um valor aceitável.

A distância e a tolerância de tempo são parâmetros inversamente proporcionais. Quanto maior a tolerância de tempo, menor a distância a ser utilizada para identificar uma perseguição, uma vez que a tolerância de tempo permite que o perseguidor avance no tempo para ficar mais próximo do alvo, como demonstra a figura 22.

5 CONCLUSÃO E TRABALHOS FUTUROS

Dispositivos móveis geram uma enorme quantidade de dados chamados de trajetórias de objetos móveis. Uma trajetória pode apresentar diversos tipos de comportamento diferentes como andar juntos, desviar de objetos, andar lentamente, etc. Na literatura existem dezenas de abordagens para fazer análise de trajetórias, porém, apesar de existirem outros trabalhos que mencionam o comportamento de perseguição, nenhum o definiu formalmente ou propôs um algoritmo para computá-lo. As contribuições principais deste trabalho foram a definição formal do comportamento de perseguição em trajetórias, o algoritmo TRA-CHASE que identifica o padrão, experimentos para validar o método proposto e um estudo sobre a performance e os parâmetros de entrada. A identificação do padrão de perseguição pode ser útil em diversos domínios de aplicação como animais caçando, policiais perseguindo bandidos, um espião investigando uma pessoa, ou um indivíduo perseguindo uma vítima para cometer um crime.

O trabalho apresentou diversas definições que juntas caracterizam um padrão de perseguição. Para aumentar a confiança e dar mais semântica ao padrão, também foram definidos tipos de padrões de perseguição, baseado no comportamento das trajetórias durante e após a perseguição. O modo como os parâmetros de tempo e distância foram utilizados diferem de outros trabalhos da literatura, permitindo a comparação de pontos de diferentes instantes de tempo.

Para avaliar o método proposto, foram realizadas duas série de experimentos. A primeira série comparou o TRA-CHASE com duas outras abordagens, demonstrando que o método encontra padrões não identificados pelos outros métodos. Um dos experimentos realizados foi com dados onde se conhecia a região das perseguições. O método identificou todas corretamente. A segunda série de experimentos validou os tipos de perseguição. Foi gerado um conjunto de dados simulando alguns tipos de perseguição, que foram classificados corretamente pelo método. Os experimentos realizados comprovaram que o método proposto encontra padrões de perseguição em trajetórias e seu tipo.

As definições de perseguição e a primeira série de experimentos foram publicados na revista *Transactions in GIS* (SIQUEIRA; BOGORNY, 2011c) e apresentados no *Workshop de Teses e Dissertações em Bancos de Dados no XXVI Simpósio Brasileiro de Banco de Dados* (SIQUEIRA; BOGORNY, 2011b). O algoritmo com os tipos de perseguição e a segunda série de experimentos foram publicados no *Simpósio Brasileiro*

de Geoinformática - GeoInfo (SIQUEIRA; BOGORNY, 2011a) e recebeu o prêmio de segundo melhor artigo do evento.

Como passos futuros pretende-se:

- Definir novos tipos de perseguição, avaliando outras características. Existem outros fatores, além dos stops, que podem caracterizar um tipo de perseguição. Por exemplo, o tipo do veículo ou objeto e a velocidade em que se movimentam podem auxiliar na classificação.
- Criar novas formas de validação dos resultados, avaliando fatores externos. Identificar o número de perseguidores pode não ser o suficiente para garantir que foi uma perseguição intencional. Fatores como o local onde ocorreu a perseguição ou o período do dia podem auxiliar na geração de padrões com mais certeza.
- Investigar a descoberta de padrões de perseguição em aplicações de tempo real
- Realizar experimentos para testar a resistência de falsos-positivos do método

REFERÊNCIAS

- ALVARES, L. O. et al. A model for enriching trajectories with semantic geographical information. In: *ACM-GIS*. New York, NY, USA: ACM Press, 2007. p. 162–169.
- ALVARES, L. O. et al. An algorithm to identify avoidance behavior in moving object trajectories. *J. Braz. Comp. Soc.*, v. 17, n. 3, p. 193–203, 2011.
- ANDRIENKO, N. V.; ANDRIENKO, G. L. Designing visual analytics methods for massive collections of movement data. *Cartographica*, v. 42, n. 2, p. 117–138, 2007.
- BAGLIONI, M. et al. Towards semantic interpretation of movement behavior. In: SESTER, M.; BERNARD, L.; PAELKE, V. (Ed.). *AGILE Conf.* [S.l.]: Springer, 2009. (Lecture Notes in Geoinformation and Cartography), p. 271–288. ISBN 978-3-642-00317-2.
- BLYTHE, P. W.; MILLER, G. F.; TODD, P. M. Human simulation of adaptive behavior: Interactive studies of pursuit, evasion, courtship, fighting and play. In: *Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*. [S.l.: s.n.], 1996. p. 13–22.
- CAO, H.; MAMOULIS, N.; CHEUNG, D. W. Mining frequent spatio-temporal sequential patterns. In: *ICDM '05: Proceedings of the Fifth IEEE International Conference on Data Mining*. Washington, DC, USA: IEEE Computer Society, 2005. p. 82–89. ISBN 0-7695-2278-5.
- CAO, H.; MAMOULIS, N.; CHEUNG, D. W. Discovery of collocation episodes in spatiotemporal data. In: *ICDM*. [S.l.]: IEEE Computer Society, 2006. p. 823–827.
- CAO, H.; MAMOULIS, N.; CHEUNG, D. W. Discovery of periodic patterns in spatiotemporal sequences. *IEEE Trans. Knowl. Data Eng.*, v. 19, n. 4, p. 453–467, 2007.
- DODGE, S.; WEIBEL, R.; LAUTENSCHÜTZ, A.-K. Towards a taxonomy of movement patterns. *Information Visualization*, v. 7, n. 3–4, p. 240–252, 2008.

Eclipse Foundation. *Eclipse - The Eclipse Foundation open source community website*. 2011. Disponível em: <http://www.eclipse.org/>. Acessado em December (2011).

Encyclopædia Britannica Online. *Napoleon I: statistical map of Russian campaign*. 2012. Disponível em: <http://www.britannica.com/EBchecked/media/70821/Statistical-map-of-Napoleons-Russian-campaign-of-1812-The-size>. Acessado em Março (2012).

ESTER, M. et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In: SIMOUDIS, E.; HAN, J.; FAYYAD, U. M. (Ed.). *Second International Conference on Knowledge Discovery and Data Mining*. [S.l.]: AAAI Press, 1996. p. 226–231.

GIANNOTTI, F. et al. Trajectory pattern mining. In: BERKHIN, P.; CARUANA, R.; WU, X. (Ed.). *KDD*. [S.l.]: ACM Press, 2007. p. 330–339. ISBN 978-1-59593-609-7.

GIANNOTTI, F.; PEDRESCHI, D. *Mobility, Data Mining and Privacy*. [S.l.]: Springer, 2008.

Google Inc. *Google Earth*. 2009. Disponível em: <http://www.google.com.br/intl/pt-BR/earth/index.html>. Acessado em Dezembro (2011).

GUDMUNDSSON, J.; LAUBE, P.; WOLLE, T. Movement patterns in spatio-temporal data. In: SHEKHAR, S.; XIONG, H. (Ed.). *Encyclopedia of GIS*. [S.l.]: Springer, 2008. p. 726–732. ISBN 978-0-387-30858-6.

HORNSBY, K. S.; COLE, S. J. Modeling moving geospatial objects from an event-based perspective. *T. GIS*, v. 11, n. 4, p. 555–573, 2007.

HORNSBY, K. S.; KING, K. Modeling motion relations for moving objects on road networks. *GeoInformatica*, v. 12, n. 4, p. 477–495, 2008.

KANDA, T. et al. Who will be the customer?: a social robot that anticipates people’s behavior from their trajectories. In: *UbiComp*. [S.l.: s.n.], 2008. p. 380–389.

KANG, C.-H.; HWANG, J.-R.; LI, K.-J. Trajectory analysis for soccer players. In: *ICDM Workshops*. [S.l.]: IEEE Computer Society, 2006. p. 377–381. ISBN 0-7695-2702-7.

LAUBE, P.; IMFELD, S. Analyzing relative motion within groups of trackable moving point objects. In: EGENHOFER, M. J.; MARK, D. M. (Ed.). *GIScience*. [S.l.]: Springer, 2002. (Lecture Notes in Computer Science, v. 2478), p. 132–144. ISBN 3-540-44253-7.

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.

LEE, J.-G. et al. TraClass: trajectory classification using hierarchical region-based and trajectory-based clustering. *PVLDB*, v. 1, n. 1, p. 1081–1094, 2008.

LEGENDRE, F. et al. Modeling mobility with behavioral rules: The case of incident and emergency situations. In: CHO, K.; JACQUET, P. (Ed.). *AINTEC*. [S.l.]: Springer, 2006. (Lecture Notes in Computer Science, v. 4311), p. 186–205. ISBN 3-540-49364-6.

PALMA, A. T. et al. A clustering-based approach for discovering interesting places in trajectories. In: WAINWRIGHT, R. L.; HADDAD, H. (Ed.). *SAC*. [S.l.]: ACM, 2008. p. 863–868. ISBN 978-1-59593-753-7.

Quantum GIS Development Team. *Quantum GIS Geographic Information System*. [S.l.], 2009. <<http://qgis.osgeo.org>>.

Refractions Research. *PostGIS*. 2011. Disponível em: <http://postgis.refractions.net/>. Acessado em December (2011).

REYNOLDS, C. W. Steering behaviors for autonomous characters. *Game Developers Conference*, 1999.

ROCHA, J. A. M. R. et al. Db-smot: A direction-based spatio-temporal clustering method. In: *IEEE Conf. of Intelligent Systems*. [S.l.]: IEEE, 2010. p. 114–119. ISBN 978-1-4244-5164-7.

SIQUEIRA, F. de L.; BOGORNY, V. Class-chase: Um algoritmo para classificação de tipos de padrões de perseguição em trajetórias de objetos móveis. In: *GeoInfo*. [S.l.: s.n.], 2011. p. 135–140. ISSN 2179-4847.

SIQUEIRA, F. de L.; BOGORNY, V. Descoberta de padrões de perseguição em bancos de dados espaço-temporais. In: *Workshop de Teses e Dissertações em Bancos de Dados no XXVI Simpósio Brasileiro de Banco de Dados*. [S.l.: s.n.], 2011.

SIQUEIRA, F. de L.; BOGORNY, V. Discovering chasing behavior in moving object trajectories. *T. GIS*, v. 15, n. 5, p. 667–688, 2011.

SOCIETY, W. *Frequency 1550*. 2005. Disponível em: <http://www.waag.org/project/frequentie>. Acessado em September (2007).

SPACCAPIETRA, S. et al. A conceptual view on trajectories. *Data and Knowledge Engineering*, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 65, n. 1, p. 126–146, 2008. ISSN 0169-023X.

WACHOWICZ, M. et al. *Moving Flock Finder, Knowledge Discovery and Data Mining Laboratory (KDDLAB-Pisa)*. 2010. Disponível em: <http://www-kdd.isti.cnr.it/moving-flock>. Acessado em Dezembro (2010).

WACHOWICZ, M. et al. Finding moving flock patterns among pedestrian through collective coherence. *Accepted in International Journal of Geographical Information Science*, 2011.