

UNIVERSIDADE DO VALE DO RIO DOS SINOS
CIÊNCIAS EXATAS E TECNOLÓGICAS.
PROGRAMA INTERDISCIPLINAR DE PÓS-GRADUAÇÃO EM
COMPUTAÇÃO APLICADA

Luciano Hennemann

**Detecção de Movimentos
Não Usuais no Espaço-Fase**

São Leopoldo

2008

Luciano Hennemann

**Detecção de Movimentos
Não Usuais no Espaço-Fase**

Dissertação de Mestrado apresentada à
Universidade do Vale do Rio dos Sinos
como requisito parcial para obtenção
do título de Mestre em Computação
Aplicada.

Orientador: Prof. Dr. Cláudio Rosito Jung

São Leopoldo

2008

CIP — CATALOGAÇÃO NA PUBLICAÇÃO

Hennemann, Luciano

Detecção de Movimentos Não Usuais no Espaço-Fase. / por Luciano Hennemann — São Leopoldo: Ciências Exatas e Tecnológicas da Unisinos, 2008.

74 f.: il.

Dissertação (mestrado) — Universidade do Vale do Rio dos Sinos. Ciências Exatas e Tecnológicas. Programa Interdisciplinar de Pós-Graduação em Computação Aplicada, São Leopoldo, BR-RS, 2008. Orientador: Jung, Cláudio Rosito.

1. Reconhecimento de Padrões. 2. Vigilância Eletrônica. 3. Ocupação Espacial. 4. Trajetória. 5. Movimento não usual. 6. Histograma *4D*. 7. Classificação. 8. Mistura de Gaussianas. I. Jung, Cláudio Rosito. II. Título.

UNIVERSIDADE DO VALE DO RIO DOS SINOS

Reitor: Dr. Marcelo Fernandes de Aquino

Diretora da Unidade de Pós-Graduação e Pesquisa: Prof^a. Dr^a. Ione Bentz

Coordenador do PIPCA: Prof. Dr. Arthur Tórgo Gómez

Dedico este estudo:
ao professor e orientador Dr. Cláudio Jung;
à esposa Edinéia que sempre esteve ao meu lado;
aos meus pais Waldemar e Maria,
que sempre acreditaram na minha capacidade;
e a minha irmã Eliane.

Agradecimentos

Gostaria de agradecer, primeiramente, ao professor Dr. Cláudio Rosito Jung pelo incentivo, ânimo, amizade, apoio e orientação em todas as etapas desta dissertação, bem como, pela sua enorme paciência em esclarecer minhas dúvidas.

Também à professora Dra. Soraia Musse por suas valiosas contribuições e aconselhamentos durante o curso, dando origem a este trabalho.

Agradeço a todos os demais Mestrandos e colegas do Curso de Física que incentivaram nos momentos difíceis e assim influenciaram diretamente nos resultados desta dissertação.

A todos os professores deste Curso que proporcionaram minha capacitação para chegar até este momento e a todos os participantes desta banca que receberam meu trabalho.

Por fim, ao professor Dr. Ney Lemke pelo grande incentivo para a continuidade de meus estudos neste Mestrado e em especial ao colega e amigo Dr. Adriano Werhli, que mesmo distante sempre esteve a disposição para sugerir e aconselhar.

Resumo

Este trabalho tem como finalidade apresentar um modelo para detecção de comportamentos não usuais baseados em trajetórias. O modelo segue uma linha de pesquisa bastante atual na área de câmeras inteligentes ou vigilância eletrônica que tende a concorrer com a enorme variedade de dispositivos baseadas em *hardware* hoje disponíveis no mercado. O fundamento está no processamento das trajetórias de pedestres ou objetos adquiridas por meio de filmagens feitas de ambientes com tráfego. O modelo é fundamentado em um período de treinamento, onde irá aprender o perfil das trajetórias selecionando, agrupando e posteriormente, guardando-as em um banco de dados. Depois irá compará-las com trajetórias novas que vão sendo adquiridas continuamente no período de teste ou operação propriamente dito. No período de teste, uma trajetória será classificada como usual se compatível com as trajetórias adquiridas durante o treinamento, ou não usual, caso contrário. Neste trabalho, portanto, serão apresentados algoritmos que detectam padrões de similaridade entre um conjunto de trajetórias no período de treinamento com cada nova trajetória adquirida no período de teste.

Palavras-chave: Reconhecimento de Padrões, Vigilância Eletrônica, Ocupação Espacial, Trajetória, Movimento não usual, Histograma 4D, Classificação, Mistura de Gaussianas.

TITLE: “DETECTION OF UNUSUAL MOTION IN SPACE-PHASE”

Abstract

This work aims to present a model for detection of unusual motion based on trajectories. This model relates to the research field on intelligent cameras and surveillance systems, that tends to compete nowadays with the enormous range of devices based in hardware available on the market. The main idea of the proposed approach is to analyze trajectories acquired from film scenes. The first step of the algorithm consists of a training period, that learns the profile of trajectories, selecting, grouping and later, keeping them in a database. After that, the algorithm compares new trajectories that are being acquired continuously in the test period. In test period, one given trajectory will be classified as usual if it is compatible with the trajectories acquired during the training period, or unusual otherwise. This work, therefore, will present algorithms that detect patterns of similarity between a set of trajectories in the training period with each new trajectory acquired in the test period.

Keywords: Electronic Surveillance, Spatial Occupation, Trajectory, Unusual Motion, Histograms $4D$, Classification, Mixture of Gaussians.

Sumário

Resumo	5
Abstract	6
Lista de Abreviaturas	9
Lista de Figuras	10
1 Introdução	12
1.1 Fundamentação do Problema	13
1.2 O Objetivo	14
2 Trabalhos Relacionados	15
2.1 Análise Comportamental Baseado em Trajetórias	15
2.2 Uso de <i>Clusters</i> no Tratamento de Trajetórias	30
3 O Método Proposto	37
3.1 Introdução	37
3.2 O Histograma $4D$ - Espaço-Fase	38
3.2.1 Quantização do Histograma	39
3.2.2 Preenchimento do Histograma	40
3.3 Agrupamento de Trajetórias	44
3.3.1 Remoção de <i>Outliers</i>	48
3.4 Período de Teste ou Operação.	50
4 Resultados Obtidos	54
5 Considerações Finais	67
5.1 Conclusões	67

5.2	Trabalhos Futuros	69
	Bibliografia	71

Lista de Abreviaturas

SOM	Mapa de Ocupação Espacial - <i>Spatial Occupancy Map</i>
PDF	Função de Densidade de Probabilidade - <i>Probability Density Function</i>
CCTV	Circuito Fechado de Televisão - <i>Closed Circuit Television</i>
HMM	<i>Hidden Markov Models</i>
HTTP	Protocolo de Transferência de Hipertexto - <i>HyperText Transfer Protocol</i>
JPEG	Formato de Compressão de Imagens com Perda de Dados - <i>Joint Photographic Experts Group</i>
CHMM	<i>Coupled Hidden Markov Model</i>
GMM	Modelo de Mistura de Gaussianas - <i>Gaussian Mixture Model</i>

Lista de Figuras

2.1	Sistema de segurança descrito por Valera e Velastin [1].	16
2.2	Detecção de objetos conforme Valera e Velastin [1].	16
2.3	Subtração de <i>background</i> de Valera e Velastin [1].	17
2.4	O <i>Hidden Markov Model (HMM)</i> utilizado em Nair e Clark [2].	19
2.5	Arquitetura do <i>framework</i> de Mahajan et al. [3].	21
2.6	Modelo de Makris e Ellis [4].	22
2.7	Caminho reconhecido durante o treinamento em Makris e Ellis [4].	22
2.8	Envelope reconhecido pelo treinamento em Makris e Ellis [4].	23
2.9	Envelope de caminhos no método de Junejo et al. [5].	25
2.10	Exemplo de envelopes no método de Junejo et al. [5].	26
2.11	Emprego de <i>Clusters</i> em Lou et al. [6].	30
2.12	Captura de trajetórias em Johnson e Hogg [7].	31
2.13	Reconhecimento de trajetórias reais em Johnson e Hogg [7].	31
2.14	Classificação de trajetórias de Maybank et al. [8].	32
2.15	O modelo proposto por Hu et al. [9].	33
2.16	Clusterização hierárquica de trajetórias por Hu et al. [9].	34
2.17	Estágios do treinamento na técnica de Zhou e Kimber [10].	35
2.18	Extração das imagens na técnica de Zhou e Kimber [10].	35
3.1	Exemplo de trajetórias normais.	38
3.2	Evolução do núcleo gaussiano.	42
3.3	Setores circulares.	43
3.4	Efeito da terceira dimensão (θ) no histograma $4D$	44
3.5	Trajetoária em preto incoerente com o treinamento.	45
3.6	Exemplo de formação de <i>clusters</i>	47
3.7	A presença de <i>outliers</i>	48

3.8	Remoção de <i>outliers</i>	50
3.9	Exemplo de treinamento.	52
3.10	Exemplo de trajetória coerente com o treinamento.	52
4.1	Classes de trajetórias coerentes no período de treinamento.	55
4.2	Teste utilizando uma trajetória.	55
4.3	Trajetoária não usual.	56
4.4	Transição entre <i>clusters</i>	56
4.5	Teste com trajetória dobrando a direita.	58
4.6	Trajetoária de teste com contorno complexo.	59
4.7	Trajetoária de teste com variação de velocidade.	60
4.8	Treinamento conforme o <i>tracking</i> de cenas de vídeo reais.	61
4.9	Trajetoária no sentido sul-norte de teste empregada.	63
4.10	Trajetoária no sentido oeste-leste de teste empregada.	64
4.11	Trajetoária curva de teste.	65
4.12	Trajetoária de teste não usual.	66

Capítulo 1

Introdução

Com o surgimento da informática, onde inúmeros algoritmos podem compor *softwares* poderosos de uso amplo e suportados por *hardwares* com grande capacidade, existe um campo de pesquisa que aponta para um mercado crescente na área de visão computacional: segurança eletrônica e câmeras inteligentes. Algoritmos de visão computacional podem ser utilizados na detecção de rotas de colisão de trânsito, espaço aéreo e reconhecimento facial, entre outras. Em particular, aplicações em vigilância eletrônica também tornaram-se indispensáveis atualmente, para proteção de bens materiais em bancos, monumentos, empresas, comércio e pessoas, tanto a nível estatal ou privado, contra vândalos, furtos e seqüestros. Em geral, os sistemas convencionais de vigilância eletrônica consistem na utilização de sensores eletrônicos ou circuitos fechados de TV e de operadores, que analisam continuamente as imagens obtidas por estas câmeras. Além de gastos dispendiosos com estes tipos de equipamentos, é preciso contar com pessoal de segurança qualificado. O trabalho deste pessoal torna-se cansativo quando poucos agentes de segurança observam vários monitores, com áreas que apresentam intenso tráfego. Nestas circunstâncias, poderá passar despercebido algum ato ou comportamento não usual, podendo resultar em algum tipo de ocorrência criminal.

Dessa forma, o desenvolvimento de algoritmos que possam detectar eventos não usuais de forma automática pode auxiliar o trabalho dos operadores humanos. Deve-se salientar que a detecção automática de eventos não usuais é um problema complexo, pois há vários aspectos a serem analisados: movimento de pessoas, postura, interações, etc. O julgamento final é uma interpretação puramente humana

e, em particular, este trabalho foca a detecção de eventos suspeitos com relação ao movimento de objetos ou pessoas como será visto a seguir.

1.1 Fundamentação do Problema

Para que possamos avaliar um ambiente e observar o comportamento das pessoas utilizando cenas filmadas, o algoritmo proposto deve identificar padrões de comportamento usuais ou não usuais, tais como:

- Uma primeira condição é monitorar regiões onde ocorre a movimentação das pessoas na cena filmada. Normalmente, ao percebermos que existem certas regiões onde as pessoas caminham mais freqüentemente, logo verificamos uma movimentação que não condiz ao que foi observado, ou seja, um deslocamento não usual, como por exemplo, caminhar fora da calçada. Este tipo de observação é feita com o auxílio de um mapa de ocupação espacial, ou seja, a área que será filmada definirá a superfície de um plano bidimensional onde as pessoas serão monitoradas individualmente por suas posições. Qualquer pessoa transitando fora de um caminho pré-estabelecido naturalmente nos chamará atenção e será considerado suspeito, pois geralmente as pessoas percorrem áreas bem definidas.
- O sentido das pessoas quando filmadas neste mapa de ocupação espacial também pode ser considerado como uma condição importante a ser monitorada, podendo-se extrair algumas condições não usuais. Se o fluxo global, em uma cena, for predominantemente em um único sentido, uma pessoa se deslocando em sentido contrário apresenta um comportamento não usual.
- A velocidade de deslocamento (em módulo) de cada componente na cena deve ser monitorada. Quando obtemos um fluxo de pessoas numa direção e observamos uma ou mais pessoas caminhando devagar, paradas ou correndo em relação a outras é necessária uma atenção redobrada. Tal situação pode ser perfeitamente uma fuga ou um comportamento de observação para um conseqüente furto.

O algoritmo a ser desenvolvido neste trabalho focará a detecção de movimentos não usuais, com base nos três aspectos acima citados. Além disso, o processo de

armazenagem e classificação das trajetórias no banco de dados será analisada, assim como, serão exploradas informações referentes a posição, velocidade e o espaço-fase.

1.2 O Objetivo

O objetivo principal deste trabalho é desenvolver um algoritmo baseado na análise de trajetórias para detecção de movimentos não usuais. Deve haver um sistema de treinamento prévio com trajetórias capturadas neste período de treinamento, que posteriormente seja capaz de identificar em trajetórias diferentes daquelas treinadas, ou seja, identificar comportamentos não usuais. A capacidade de diferenciação de comportamento usual e não usual, será realizada comparando a posição e a velocidade local de cada ponto de uma trajetória a ser analisada com o padrão de movimentação, obtida durante a fase de treinamento.

O restante desta monografia está estruturada da seguinte forma: no segundo Capítulo, alguns trabalhos relacionados com esse tópico de pesquisa são analisados, e o terceiro Capítulo descreve o método proposto. No Capítulo quarto mostraremos alguns resultados experimentais, e no quinto, as conclusões e algumas idéias para a continuidade do trabalho.

Capítulo 2

Trabalhos Relacionados

Atualmente, o campo de pesquisas que envolve detecção de comportamentos não usuais em trajetórias de pessoas, baseadas cenas de vídeo, vem aumentando significativamente com trabalhos inovadores. A tentativa de construir um sistema que seja capaz de modelar ou aprender comportamentos tipicamente humanos, e depois confrontá-lo com a realidade tem demonstrado potencial no auxílio de operadores humanos que operam em monitoramento de vigilância. Este Capítulo descreverá, brevemente, alguns métodos para detecção de movimentos não usuais baseados em trajetórias. Devemos salientar que existe uma grande variedade de algoritmos que utilizam outros tipos de informação, como por exemplo, avaliação de posturas, relações interpessoais, como descrito em Soldera e Hosie et al. [11, 12] e outros, mas esta breve revisão está direcionada em trabalhos relacionados à análise de trajetórias. Em particular, a Seção 2.2 aborda algoritmos que agrupam trajetórias em classes. Essa estratégia também será adotada neste trabalho.

2.1 Análise Comportamental Baseado em Trajetórias

Um panorama geral sobre sistemas de segurança pode ser obtido em Valera e Velastin [1], resumindo que sistemas de detecção de objetos móveis devem incluir a detecção e reconhecimento do que necessitamos monitorar, seu acompanhamento, análise de seu comportamento e processos de busca de dados relevantes (ver Figura 2.1).



Figura 2.1 – Processos envolvidos em um sistema de segurança conforme Valera e Velastin [1].

A demanda em pesquisa e sofisticação desses sistemas têm sido intensificadas, e seu emprego também tem sido crescente, principalmente em cenários militares, públicos e comerciais. A evolução dos sistemas de segurança inteligentes começou com a entrada do sistema CCTV (*Closed-Circuit Television*), que consiste em um número de câmeras espalhadas em um ambiente, conectadas a uma quantia de monitores inicialmente selecionáveis por comandos individuais (uma matriz de monitores). Nestes sistemas, usavam-se técnicas analógicas de armazenamento e visualização de imagens. Hoje, isso é feito de maneira digital e as mesmas informações circulam pelos diversos monitores, ou podendo passar por sistemas de seleção individual CCTV que são interligados por cabos coaxiais sem grande degradação das imagens. A detecção dos objetos pode basicamente ser feita de duas maneiras assumindo que a câmera está parada:

- Pela diferença temporal; que consiste em subtrair informações entre dois quadros consecutivos, conforme Figura 2.2. Trata-se de uma boa opção em ambientes dinâmicos devido a sua fácil adaptação, mas se torna pobre no caso de uma extração maior de informações a nível de pixels.



Figura 2.2 – Detecção de objetos por diferença temporal, conforme Valera e Velastin [1].

- Pela subtração de *background*: que subtrai o fundo da imagem isolando o que se quer monitorar (ver Figura 2.3). Este processo tem um desempenho

superior no caso de extração de informações de objetos, mas é muito sensível em mudanças dinâmicas no ambiente.



Figura 2.3 – Detecção de objetos pela subtração de *background* extraído de Valera e Velastin [1].

O acompanhamento de objetos (*tracking*) pode ser feito utilizando modelos geométricos em $2D$ ou $3D$, dependendo da quantidade e complexidade da aplicação. Normalmente, são tidos como alvos: carros, pessoas ou ambas. Há diversas abordagens comumente utilizadas para realizar o acompanhamento de objetos como filtros de *Kalman*, emprego de modelos de probabilidades como distribuições Gaussianas, filtros *Bayesianos*, modelos de *Markov*, entre outros. As demais citações no decorrer deste Capítulo comentarão algumas delas.

A próxima etapa dos sistemas inteligentes de vigilância é a análise do comportamento. Corresponde em classificar o problema de uma determinada característica ou postura não usual através do tempo. Isto presume que o sistema deve possuir algum tipo de treinamento prévio, para comparar uma situação corrente com outra guardada em sua base de dados. Aqui entram técnicas poderosas de reconhecimento, como redes *Bayesianas* e modelos de *Markov*.

Como estágio final temos que obter um banco de dados, onde todas as informações extraídas de imagens que servirão tanto para referência, bem como, os dados adquiridos durante uma supervisão, devem estar disponíveis para futuras consultas e manutenção de históricos.

Atualmente, estamos em um estágio que chamamos de terceira geração de

sistemas inteligentes [1]. Este conceito define que temos sistemas concebidos com um número muito grande de câmeras operando simultaneamente. Monitoram uma área geográfica considerável, com o objetivo de analisar comportamentos de objetos ou humanos previamente modelados. Estes comportamentos ainda são exaustivamente estudados e aprimorados, a fim de termos premissas e regras firmemente estabelecidas de comportamentos usuais e não usuais, devendo ser o mais semelhante possível com a realidade. A seguir, serão descritos alguns artigos de detecção de movimentos não usuais com foco semelhante ao deste trabalho.

Nair e Clark [2] descreve um sistema automático de vigilância que detecta atividades não usuais em uma cena. É composto por um sistema de detecção e acompanhamento das pessoas na cena, reconhecimentos de atividades usuais e detecção de atividades que apresentam amplas diferenças das características normais. Modelos de *Markov HMM (Hidden Markov Models)*, que são *frameworks* de reconhecimento estocástico, formam a base do reconhecimento e detecção anormal. Foi utilizado um computador *Pentium III* com *Windows 2000* para implementar o sistema de monitoração em um corredor de um escritório em tempo real. A arquitetura do sistema utiliza uma *netcam* para monitorar o corredor do escritório configurado como servidor HTTP (*HyperText Transfer Protocol*) por onde podemos fazer o *download* do vídeo e os quadros podem ser comprimidos em extensão JPEG (*Joint Photographic Experts Group*) via *internet*. Várias câmeras podem ser adicionadas ao sistema. Para detecção e acompanhamento das pessoas na cena, é utilizada a técnica de subtração de *background*.

Para a atividade de reconhecimento, utilizando HMM, é necessário que os módulos de detecção e acompanhamento de objetos forneçam dados das seqüências de tempo dos *blobs* da cena. Essas informações são necessárias para que as atividades de cada *blob* sejam classificadas por um HMM discreto. Um vetor característico do *blob* é obtido a partir das coordenadas do centro de massa do respectivo *blob*, e também a direção e altura. Sendo HMM discreto, uma representação simbólica é dada ao vetor de dados característico. Cada vetor característico é convertido a um símbolo discreto usando um vetor de quantização. Uma vez treinado, mapas com vetores quantizados são indexados a vetores de códigos (levando em conta a métrica Euclidiana). Veja a Figura 2.4.

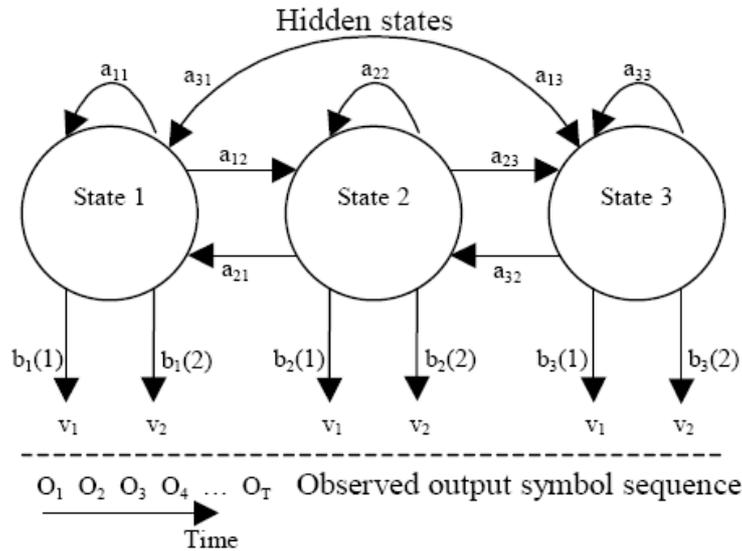


Figura 2.4 – O *Hidden Markov Model* (HMM) utilizado em Nair e Clark [2].

Cada atividade no corredor do escritório é reconhecida usando um conjunto de HMM's que reconhece uma atividade. A probabilidade da seqüência observada é computada para cada HMM, e a atividade é reconhecida por uma representação com o mais alto *log* de semelhança. Assim, se o *log* de semelhança é mínimo para todos os HMM's esta atividade é considerada não usual.

No artigo de Gavrilá [13] é salientada a importância do campo de pesquisas, que engloba o uso da visão computacional na pesquisa do comportamento humano. Os sistemas de vigilância que chamamos de inteligentes são aqueles que filtram comportamentos não usuais e notificam o agente de segurança para acontecimentos realmente importantes. Isso evita que o agente fique exposto à rotina entediante de observar inúmeros monitores continuamente, o que acaba provocando fadiga no trabalho operando sistemas de vigilância tradicionais. Mas envolve um cuidadoso estudo de caso para que certos comportamentos realmente possam ser definidos, e incorporados, para evitarmos os alarmes indevidos (falsos). Esta aplicação se torna importante em ambientes como estacionamentos, acessos a prédios, lojas, bancos, etc. Os gestos e trajetórias das pessoas podem ser captadas e classificadas em três grupos:

- Uma aproximação $2D$ sem levar em conta a forma e postura das pessoas se restringindo apenas a mapear uma pessoa sobre um plano bidimensional.

- Uma aproximação $2D$ que leva em consideração a forma com que o corpo humano aparece numa cena de vídeo. Alguns movimentos de certas partes do corpo humano, são captados conforme modelos bidimensionais simulados a partir de objetos que se assemelhem a alguma parte do corpo humano.
- Uma aproximação $3D$ utiliza uma referência com base em articulações com graus de liberdade, centrado em um plano de coordenadas. Apesar da técnica ser mais difícil de ser tratada, pois também leva em consideração o tempo, a mesma apresenta um modelo com maior precisão.

O *framework* de Mahajan et al. [3] executa atividades de reconhecimento baseado em um modelo de máquina multicamadas de estados finitos, implementada usando como base um módulo de processamento de imagens mais simples para detecção espaço-temporal e identificação de limites em objetos. Esta máquina de estados finitos (autômato) passa por um período de treinamento em modo não supervisionado com padrões usuais por um período longo de tempo. Já no período de reconhecimento, atividades usuais são aceitas e os padrões que desviarem são reconhecidas e classificadas como anormais. A camada de processamento de imagens de baixo nível opera da seguinte forma:

- Com segmentação usando multicamadas adaptativas de *background* e subtração.
- Identifica pontos mais e menos intensos nos segmentos das imagens.
- Os objetos a serem identificados são carros, pessoas e sacolas.
- No acompanhamento de objetos foram utilizados filtros de *Kalman* e igualase os quadros internos das seqüências de imagens usando a intersecção dos volumes utilizando histogramas de saturação da cor dos objetos detectados.
- Detecção do objeto no plano XY . Humanos são computados pela posição dos pés em uma superfície plana, pelo ponto mais baixo da região segmentada ao longo da direção vertical e aplicado neste, a homografia da superfície plana. Para a visão superior (topo) dos carros é computado o centróide da região segmentada e aplicado a homografia. Na visão em perspectiva dos carros é utilizado o ponto mais baixo.

Durante o período de treinamento, os dados são processados através de tarefas que devem ser concluídas passo a passo pelo algoritmo. É, então, construído um autômato genérico com inúmeras camadas de estados com as diferentes situações encontradas. O algoritmo faz, continuamente, neste período de treinamento varreduras para saber se surgiu algum novo estado lógico. Se positivo, ele é incorporado ou memorizado. Assim, no período de reconhecimento cada evento é comparado com aqueles memorizados pelo autômato, e ocorrendo alguma situação com uma característica diferente é sinalizado um comportamento não usual. A arquitetura do *framework* da máquina multicamadas de estados finitos pode ser vista na Figura 2.5.

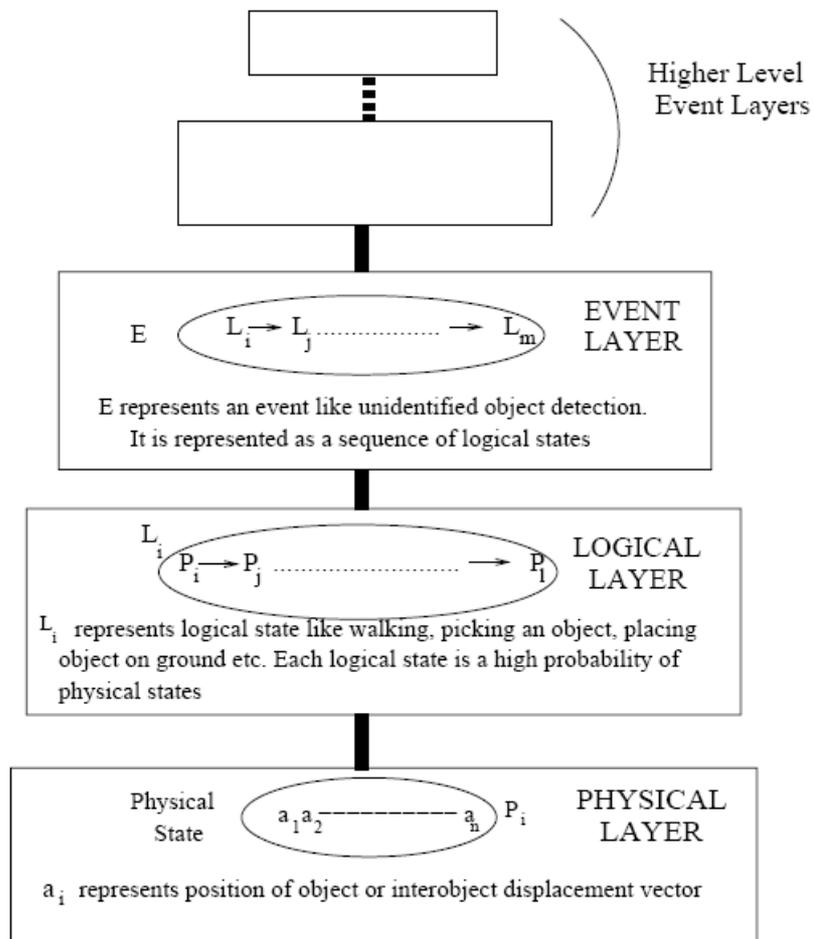


Figura 2.5 – Arquitetura do *framework* de Mahajan et al. [3].

O modelo físico de Makris e Ellis [4] pode ser ensinado pelo usuário ou aprendido pelo modelo durante um período de treinamento. Em cada parte da cena o movimento é descrito semanticamente, e assim é classificado como entradas,

caminhos, rotas ou pontos de parada criando os envelopes de caminhos (ver Figura 2.6).

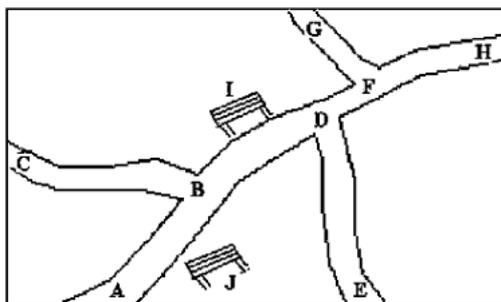


Figura 2.6 – Modelo de Makris e Ellis [4].

O modelo acima possui entradas A,C,E,G,H com junções B,D,F. Os caminhos possíveis são AB, CB, DE, FH, as rotas possíveis são ABDFH, EDFG, CBDE com zonas de parada I e J. A estrutura da cena afeta o comportamento das pessoas em suas trajetórias. Quando temos tráfego de veículos obtemos um deslocamento de carros em um envelope com caminhos fixo, mas nas áreas de deslocamento de pedestres obtemos o mesmo padrão mesmo não havendo limites precisos para determinar estes envelopes de caminhos. A Figura 2.7 mostra um envelope de caminhos reconhecido durante o treinamento.

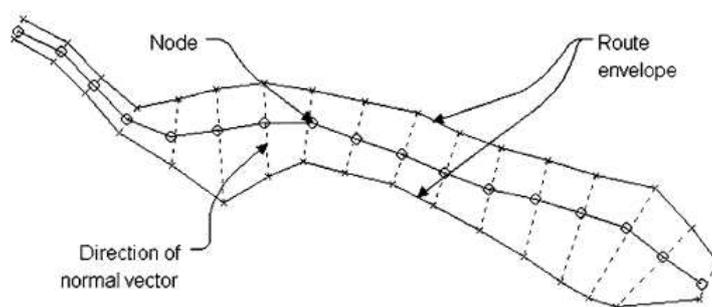


Figura 2.7 – Caminho reconhecido durante o treinamento em Makris e Ellis [4].

Dois parâmetros se tornam importantes no treinamento: o fator de amostragem, que define a distância entre dois nodos consecutivos definindo a resolução do envelope de caminhos e a distância mínima entre duas rotas, que determina quando dois envelopes de caminhos distintos devem ser combinados para formar um somente por estar mais próximo. Com o treinamento concluído,

as próximas trajetórias são analisadas utilizando cadeias de *Markov*, que permite estipular a probabilidade de uma trajetória percorrer por cada caminho disponível. Assim sendo, esse modelo permite que novas trajetórias sejam classificadas como usuais ou não usuais e a Figura 2.8 nos mostra envelopes conseguidos pelo período de treinamento.

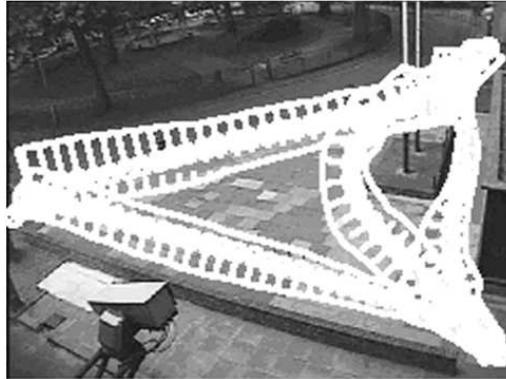


Figura 2.8 – Envelope reconhecido pelo treinamento em Makris e Ellis [4].

Já no trabalho de Niu et al. [14] é utilizado um processo que emprega várias câmeras para uma mesma área filmada, instaladas em locais diferentes. Assim, é formado um vetor com a posição da mesma pessoa em cada câmera e em cada instante, durante sua trajetória. O acompanhamento das pessoas é feito com auxílio de algoritmos vinculados ao paradigma de hipótese e verificação como: filtros de *Monte Carlo*, filtro de partículas, condensação baseado na importância, propagação condicional de densidade e algoritmos genéticos. A aplicação destes algoritmos gera a possibilidade de não necessariamente empregar distribuições Gaussianas, que serviriam para remover o ruído. Durante a filmagem, cada quadro pode ser interpretado como um instante ou estado de uma pessoa (*blob*) se locomovendo na cena. Em cada estado do *blob* é considerada a posição, velocidade e aceleração instantâneas, e para cada pessoa em movimento é prevista sua posição de acordo com sua velocidade e aceleração anteriores. O problema da oclusão, com essa tratativa, passa a ser reduzido pois se pode prever as trajetórias oclusas baseando-se nas informações dos estados anteriores. Também nas informações de outras câmeras quando o objeto está ocluído desta, e também das informações provenientes do modelo de *Markov*. Em se tratando de comportamento individual ou grupo, são utilizadas as propriedades estatísticas das cadeias de *Markov*, que distinguem tipos

de comportamento como:

- Comportamento de perseguição, onde a posição relativa é quase constante e velocidade relativa nula.
- Comportamento de colisão, onde ocorre a variação linear da distância entre dois obstáculos ou objetos com velocidade constante e não nula.

O modelo descrito acima nos mostra que o comportamento é determinado pela análise da diferenciação e correspondência dos pixels entre as trajetórias das pessoas que estão no modelo. Baseado no treinamento obteve-se 100% de sucesso na detecção de padrões como o de perseguição e 80% para os comportamentos de colisão, obtendo-se assim um bom desempenho na classificação de comportamentos.

O próximo comentário que faremos será sobre o trabalho de Cupillard et al. [15], que utiliza várias câmeras monitorando uma mesma cena, rastreando os acontecimentos ocorridos em uma estação de trem. É criado um modelo tridimensional de uma cena baseada nas áreas de interesse, para que o sistema determine o comportamento, juntando a visão de cada câmera num único modelo para esta cena. Nesse modelo também é estimada a forma tridimensional de cada objeto. Existe uma gama de objetos gravados na sua base de dados, o que torna esse modelo capaz de classificar novos objetos pela forma dos mesmos. Os objetos procurados são pessoas, grupos, metrô, pessoa oclusa, multidão, ruídos e objetos não conhecidos. A classificação é feita quadro a quadro e o sistema mantém um grafo de objetos móveis da cena, obtidos pela união do grafo de objetos de cada câmera. É utilizado um *tracker* para cada tipo diferente de personagens da cena. Um *tracker* de pessoas sozinhas e em grupos, como por exemplo, trabalham juntos monitorando cada integrante e contabilizando as pessoas de cada grupo, até que a cena fique lotada a 2/3 da área visível sobreposta por objetos em movimento. Atingido esse patamar, o *tracker* de multidões é acionado obtendo-se assim uma maior precisão do contexto, pois é muito difícil separar pessoas entre si devido ao excesso de oclusão.

A grande finalidade do trabalho é determinar comportamentos diferentes dos padrões normais no complexo do metrô. Foi definido um contexto que escreve e reutiliza todos os métodos necessários para a caracterização e o reconhecimento dos comportamentos. Esse contexto é flexível e declarativo permitindo dinamizar

as conclusões sobre trajetórias usuais e não usuais. O modelo proporciona aos personagens da cena (pessoa, grupo ou multidão), variação de velocidade, forma e trajetória, obtendo-se um estado de agitação, por exemplo, demonstrando um tipo de comportamento. Com isso podemos avaliar um cenário que está ocorrendo num determinado instante no trem que pode ser de luta, trapaça, obstrução de caminho, vandalismo, assalto, etc. O reconhecimento dos comportamentos procurados é feito comparando se o cenário está ativo ou não no sistema. E estando ativo, a busca passa para o cenário seguinte, definido no formalismo do comportamento na seqüência até se encaixar num evento¹, que seja uma sentença no contexto pré estabelecido no formalismo. Apesar do sistema empregar o mesmo formalismo humano que define uma determinada situação como usual ou não, este ainda perde em eficiência se a observação fosse realizada diretamente pela percepção humana, como esperado.

Um período de treinamento é pré-requisito em muitos trabalhos citados e serve como base na discretização futura das trajetórias. Cada modelo delimita como essas trajetórias serão captadas e tratadas posteriormente, gerando sua base de dados particular baseado no que se quer fazer. Conforme descrito no trabalho de Junejo et al. [5], durante o treinamento de uma cena, inúmeras trajetórias são capturadas e gravadas, sendo que as similares são agrupadas, formando as arestas de um grafo. Estas arestas significam os caminhos mais comuns obtidos, onde nas suas redondezas obtemos uma área conhecida como envelope de caminhos, que nos mostra o quanto abrange essa área em torno deste caminho propriamente dito, conforme ilustrada na Figura 2.9.

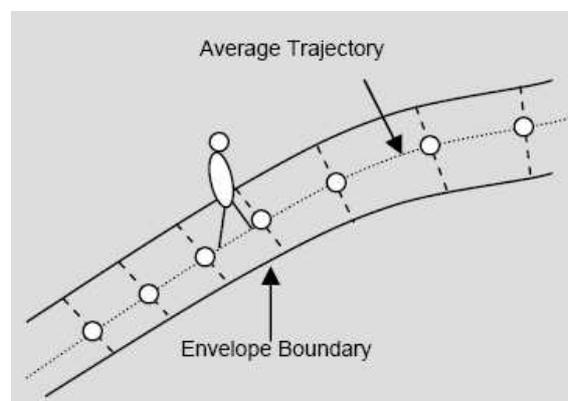


Figura 2.9 – Envelope de caminhos no método de Junejo et al. [5].

¹É uma situação momentânea em que o personagem se encontra.

Segundo esse método, as trajetórias pós treinamento são verificadas conforme alguns critérios como: características espaciais, velocidade e curvatura, nesta ordem. As características espaciais de uma ou mais trajetória(s) tidas como inconforme(s), ou seja, são aquelas em que 90% dos pontos estão fora do envelope de caminhos correspondentes avaliados pelo teste da distância de *Hausdorff*. O segundo critério é utilizado se a trajetória não for considerada inconforme, sendo avaliada a velocidade da trajetória pelo teste de *Mahanobilis*. O último critério é utilizado para avaliar se a curvatura da trajetória é reta ou irregular. Como exemplo prático podemos verificar na Figura 2.10, onde a cor vermelha mostra uma ou mais trajetórias não usuais e as demais com a mesma cor mostram trajetórias com velocidades semelhantes.

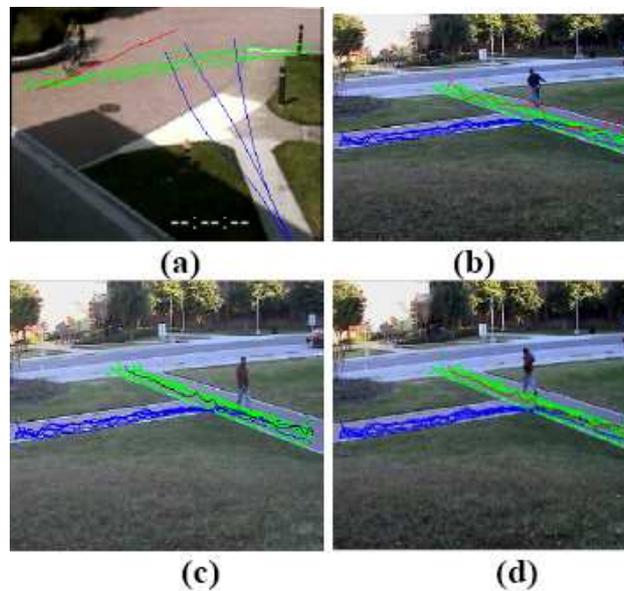


Figura 2.10 – Exemplo de envelopes no método de Junejo et al. [5].

A trajetória em vermelho, mostrada na Figura 2.10(a) caracteriza um ciclista que está trafegando fora do envelope de caminhos caracterizado pelas trajetórias verdes. Na Figura 2.10(b) temos uma trajetória não usual de uma pessoa caminhando em zigue-zague. Uma pessoa com trajetória normal está caminhando na calçada conforme Figura 2.10(c) e na Figura 2.10(d) uma pessoa corre em uma área com velocidade que não corresponde ao treinamento gravado no envelope de caminhos.

Em Fung e Jerrat [16], a diferença entre quadros consecutivos é obtida pelo cálculo em blocos durante a separação do *background* do *foreground*. Assim, se a

média das diferenças entre dois blocos de quadros consecutivos for maior que o limiar é considerado que há atividade no bloco corrente. Blocos ativos e próximos entre si são juntados formando *blobs* correspondendo a objetos em movimento na cena. Blocos com formas bastante irregulares como árvores são reconhecidos e eliminados podendo gerar confusão caso o intruso não seja detectado se caminhar nesta região. Após a remoção do *background* irregular, são utilizadas várias técnicas para detectar um comportamento não usual. A primeira providência é avaliar se a área do *blob* varia muito, ou seja, se a quantidade de pixels varia muito no decorrer da trajetória. Cada pixel da câmera pode ter um fator de peso que corresponde à distância da câmera à área filmada por esse pixel. Esse sistema de pesos é eficiente para o problema das trajetórias que se afastam da câmera, causando a diminuição gradual do tamanho do *blob*. É verificado, também, se a cor dos pixels varia muito. Se a média das diferenças de cor de cada pixel de *blobs* entre quadros consecutivos for maior que um limiar é considerado não usual. Procura-se também, detectar variação na forma do objeto utilizando o teste de *Freer*, que utiliza uma relação entre volume e perímetro. Caso o valor obtido for maior que um limiar especificado é considerado não usual, em função das anormalidades nas características físicas da pessoa projetada na câmera.

Aqui, também, se utilizou uma rede neural para avaliar as trajetórias quanto a sucessão de zonas, pelas quais, cada pessoa se deslocou. A rede neural precisa de um período de treinamento no início do processo, para adquirir capacidade de detectar tipos de comportamentos pré-definidos classificados como suspeitos.

A proposta de Hung e Gong [17] mostra um *framework* que extrai e representa características não usuais em cenas de vídeo no decorrer do tempo. A detecção dos movimentos e suas interações são traduzidas em quantização das ocorrências, ou seja, o método quantiza quantas vezes um mesmo evento ocorre e o organiza de maneira hierárquica, através das seqüências de cenas. Os resultados mostraram que é possível reduzir matrizes multidimensionais altamente populadas, representando correlações entre regiões com movimentos não usuais para um vetor com atividades não usuais claramente definida. Esta facilidade detecta movimento não usual causado por um evento individual ou um fenômeno de causa e efeito mais complexo pela separação espacial, mas temporariamente correlacionada pela localização da cena.

A extração das características salientes está fundamentada em entropia, ou

seja, quanto mais improvável, mais saliência ou não usual obtemos o movimento e vice versa. A entropia, neste caso, é uma medida estatística de imprevisibilidade. E esta entropia somente, não diferencia características usuais de não usuais. Como exemplo, podemos citar o caso do ruído que deverá ter alta entropia, se sua distribuição não se estender de forma plana. Com isso, a entropia é estendida a múltiplas escalas e atributos, para que não somente seu pico seja avaliado e sim sua distribuição pelas vizinhanças. Portanto, o método acumula eventos de movimentos salientes, baseados na interação espaço-temporal com relação às vizinhanças. Acumulando situações de reincidência de eventos bastante característicos através das seqüências de vídeo, gera-se então, um modelo de correspondências. Assim é permitindo quantizar o quanto não usual foi aquele movimento naquele local espaço-temporal, baseando-se em uma realidade de um evento usual daquele respectivo movimento.

O trabalho de Hosie et al. [12] está relacionado a classificação e análise de comportamentos a partir de dados de câmeras de segurança. As pesquisas relacionadas a análise de comportamentos (seja individual ou de grupos) normalmente envolvem domínios que estabelecem regras, ou seja, sabemos que times de futebol são caracterizados pelas cores de cada time e isso é uma regra. Para um sistema de vigilância ser eficaz, o sistema de consulta a eventos incorporados deve, seguir a lógica de forma a obter de maneira rápida um histórico de um determinado acontecimento que se deseje analisar. Os pontos básicos a serem atendidos por um histórico são: o número de pessoas envolvidas, o espaço ou local e o tempo. A segmentação e categorização dos comportamentos foram feitos como segue:

- Pares de primitivas são definidas como um conjunto de movimentos (classes de pares), que podem ocorrer entre dois alvos em uma cena por um intervalo de tempo para amostragem. Em um período de treinamento, este par de primitivas, pode desenvolver uma série de movimentos a serem monitorados. A quantidade de quadros estabelecida para cada movimento é fixada com o auxílio de um algoritmo de acompanhamento de objetos, para distinção de cada classe de pares e assim estabelecer uma árvore de regras. Aqui nesse caso, a velocidade é invariante e não influencia na medição.
- Cadeias de pares são usadas para obter combinações de pares de primitivas

mais complexas de movimentos entre si, por um período temporal mais longo. Podemos obter séries de movimentos estacionários ou convergentes, por exemplo. As cadeias de pares são especificadas para serem usadas em grafos diretos, mas é esperado sua incorporação em sistemas de aprendizado de máquina no futuro, para automaticamente, derivar os grafos provenientes de amostras positivas de comportamentos complexos.

- Os grupos de primitivas são utilizados em grupos, cuja a detecção de um determinado comportamento ou tarefa é mais complexa, requerendo uma segmentação mais direcionada a dados específicos (alvos). A finalidade é encontrar comportamentos específicos entre estes alvos em uma cena.

O trabalho de Soldera [11] está fundamentado no tratamento de trajetórias de pessoas, extraídas com base em cenas de vídeo. Estas trajetórias servirão para um treinamento gerando um histórico de ocupação na cena. Esse histórico é conhecido por SOM [18] (*Spatial Occupancy Map*) e fornece as regiões que obtivemos alta ou baixa ocupação espacial. Em um período posterior de teste, cada nova trajetória é comparada com esses mapas e trajetórias não usuais são detectadas, quando a trajetória apresentar baixa ocupação espacial. O conceito de SOM está fundamentado em uma imagem de fundo, normalmente é a própria imagem da área em que se deseja monitorar, onde uma matriz inicialmente com zeros é incrementado continuamente durante o período de treinamento. Cada pixel da matriz é incrementada em uma unidade à medida que uma trajetória é percorrida sobre ela. Com isso obtemos vários trilhos ou áreas de ocupação espacial decorrentes desta movimentação. Assim, concluído o treinamento, obteremos áreas de ocupação densamente incrementadas e áreas nulas ou quase nulas. Isto diferencia um comportamento usual de outro não usual. Aqui utiliza-se câmeras posicionadas no topo de cada cena a ser monitorada, de maneira a minimizar os efeitos da deformação da perspectiva. Neste sistema, também, obtemos um processo de alargamento das trajetórias criando uma máscara (convolução), impedindo que ocorra a formação de espaços vazios entre as mesmas no período de treinamento. Isto poderia gerar uma detecção errônea de trajetórias não usuais no período de teste, em função desta percorrer uma região pouco incrementada.

O trabalho de Soldera [11] servirá de base para esta proposta, onde o modelo

passa por modificações e aperfeiçoamentos no método de detecção de trajetórias que descreveremos no próximo Capítulo.

2.2 Uso de *Clusters* no Tratamento de Trajetórias

O trabalho de Lou et. al. [6] propõe que trajetórias semelhantes em velocidade, forma e localização física possam ser agrupadas de maneira a formar *clusters*². As trajetórias tidas como modelo padrão em cada *cluster* são mantidas numa rede de *Bayes*, para determinar a quais *clusters* as novas trajetórias filmadas se enquadrarão. A Figura 2.11 ilustra esse processo.

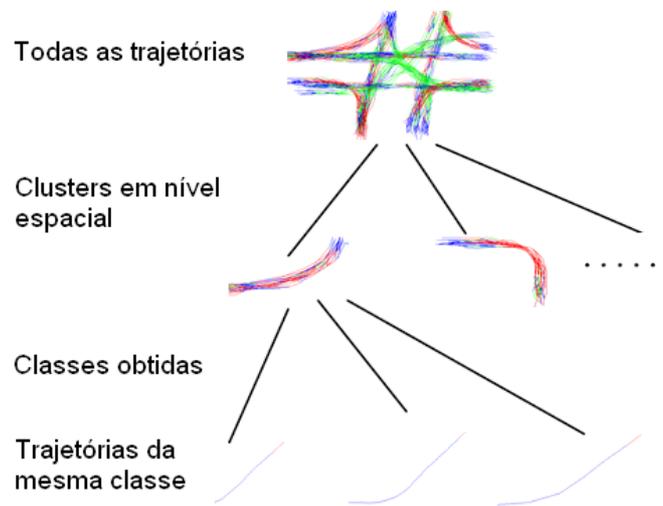


Figura 2.11 – Emprego de *Clusters* em Lou et al. [6].

A primeira parte da Figura mostra todas as trajetórias misturadas. A segunda nos mostra agrupamentos por velocidade, forma e localização física, e a terceira ilustra diferentes trajetórias de uma mesma classe. A medição da similaridade espacial nas trajetórias é obtida pelo emprego da distância de *Hausdorff*. Se uma nova trajetória tiver uma probabilidade baixa de pertencer a um *cluster* qualquer, será considerada não usual, e caso contrário será aceita em um dos *clusters* que apresentar uma característica semelhante. São usados, também, contextos para descrever as trajetórias em linguagem natural como: a trajetória *y* está entrando numa nova região.

²São agrupamentos em classes de trajetórias que apresentam características semelhantes.

Outro trabalho a ser comentado é o trabalho de Johnson e Hogg [7], que captura a trajetória de pedestres em movimento no mundo real usando seqüências longas de imagens, captadas por uma câmera de vídeo estática (veja Figura 2.12).

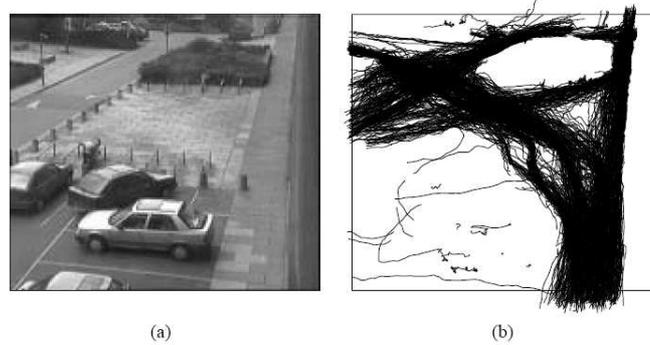


Figura 2.12 – Captura de trajetórias em Johnson e Hogg [7].

As trajetórias são amostradas a uma taxa de dois quadros por segundo. Em vez de utilizar seqüências de posições para descrever o movimento dos objetos, são utilizadas trajetórias, que são descritas em termos de fluxo de vetores. Portanto, o fluxo de um vetor significa a posição e a velocidade instantânea do objeto. Esse fluxo de vetores é calculado à partir dos dados crus, obtidos das cenas de vídeo, considerando a mudança das coordenadas ao longo da trajetória capturada pela câmera. Assumindo que cada ponto da trajetória é obtido em intervalos de tempo constantes, tal diferença fornece a velocidade instantânea.

Alguns resultados na aquisição de trajetórias podem ser observados na Figura 2.13. Vale lembrar que essa aquisição foi feita utilizando métodos estatísticos, e assim pode efetivamente ser usado como uma ferramenta de comparação para obter características de uma ou mais trajetórias.

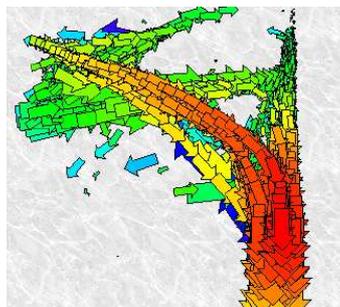


Figura 2.13 – Reconhecimento de trajetórias reais em Johnson e Hogg [7].

Também, pode ser usado quando se tem apenas partes de trajetórias e com base nas porções reais podemos estimar e simular com protótipo de vetores, as trajetórias completas.

Maybank et al. [8] desenvolveu um *framework* de um sistema de vigilância que recupera dados conforme a necessidade do usuário. Sequências de vídeo obtidas por câmeras digitais servem de dados de entrada do sistema. É obtido o acompanhamento dos objetos móveis da cena e suas trajetórias são extraídas. Um algoritmo espectral é usado para formar *clusters*, agrupando trajetórias semelhantes e assim servir de referência de treinamento para o modelo das atividades. Descrições semânticas são adicionadas ao modelo de atividade de maneira a formar modelos de atividades semânticos, e índices semânticos são construídos para servirem de base de dados de vídeo.

Portanto, o processo de vigilância e a recuperação dos dados são combinados explorando a semântica na recuperação dos dados obtidos por vídeo, através do aprendizado obtido pelos modelos criados pelas atividades dos objetos. Este aprendizado é obtido pelo movimento característico das trajetórias que são agrupadas hierarquicamente (ver Figura 2.14).

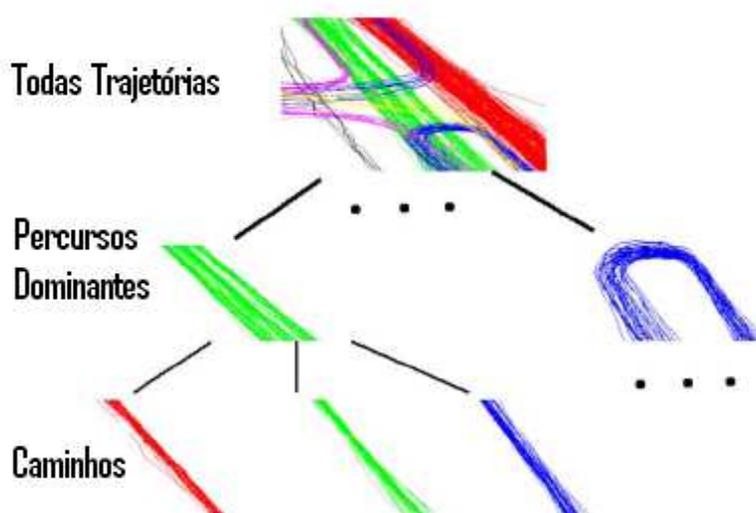


Figura 2.14 – Classificação de trajetórias de Maybank et al. [8].

A informação espacial e temporal é utilizada para formar os *clusters*, dividindo as trajetórias em categorias e em subcategorias com mesmo perfil característico. Cada *cluster* de trajetórias corresponde a um modelo de atividade, e cada modelo

de atividade representa um conteúdo semântico de uma categoria correspondente de atividades. Foi proposto uma estrutura hierárquica de índices semânticos e recuperação das atividades dos objetos. Os modelos de atividade são usados como chaves nos índices para acessar os objetos individualmente. Cada atividade individual, herda automaticamente todas as descrições semânticas dos modelos de atividade a que pertence. Somente um item de descrição semântica para um modelo de atividade é obtido manualmente e os demais itens do modelo de atividade são obtidos automaticamente. Baseado na estrutura hierárquica dos índices semânticos e recuperação de dados, as consultas com palavras-chaves, consultas a múltiplos objetos e breves descrições, podem ser obtidos também, utilizando-se as restrições temporais de sucessão, simultaneidade e algoritmos específicos para desenhar trajetórias conforme queira o usuário.

Hu et al. [9] apresenta um sistema de aprendizagem de padrões de movimentos de objetos que pode ser usado para detectar e prever comportamentos anômalos. O sistema projetado está baseado no uso de *clusters*, semelhante ao descrito por Maybank et al. [8]. São armazenados as informações contidas nos padrões de contorno no deslocamento dos objetos usando o algoritmo *fuzzy K-means*. Também, é proposto um algoritmo para fazer o acompanhamento de múltiplos objetos (ver Figura 2.15).

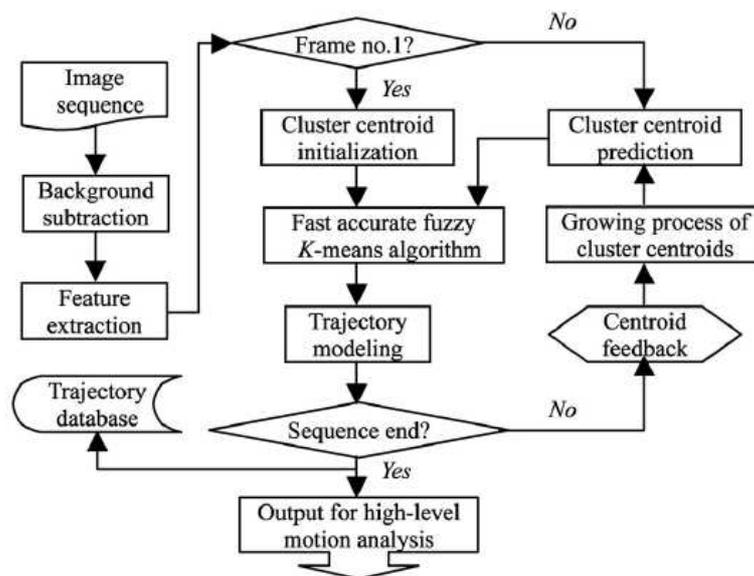


Figura 2.15 – O modelo proposto por Hu et al. [9].

Durante o treinamento, trajetórias são agrupadas hierarquicamente usando informação temporal e espacial, onde cada perfil ou padrão de contorno é representado por uma cadeia de distribuições Gaussianas. Baseado no aprendizado estatístico destes padrões, outros métodos estatísticos são usados para detectar e prever anomalias e comportamentos.

Obtendo-se uma quantidade significativa de trajetórias, este algoritmo constrói automaticamente um padrão estatístico dos perfis de contorno obtidos. Este sistema é composto por um módulo de acompanhamento de objetos, um módulo de treinamento dos perfis ou padrões obtidos estatisticamente, detecção de anomalias e predição de comportamentos futuros (ver Figura 2.16).

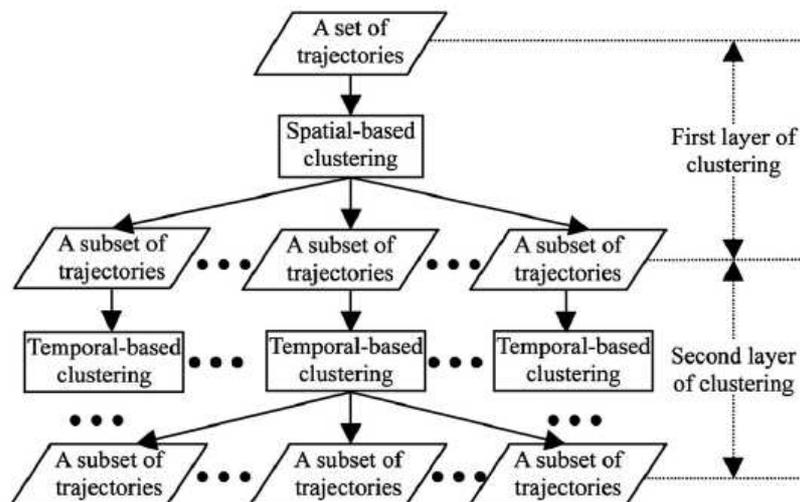


Figura 2.16 – Clusterização hierárquica de trajetórias por Hu et al. [9].

O módulo de acompanhamento de múltiplos objetos é implementado formando e agrupando em *clusters* os pixels de cada plano de imagem, e ao mesmo tempo comparando com as distribuições de pixels entre sucessivas imagens. A saída deste módulo fornece características extraídas destas trajetórias como cores, tamanho etc. Isso forma uma amostragem de dados para o treinamento deste perfil. Posteriormente, esse perfil é treinado utilizando o processo de agrupamento hierárquico ou *clusters*, baseado em um algoritmo do tipo *fuzzy k-means*. Cada padrão ou perfil de contorno obtido é representado por uma cadeia de distribuições de Gaussianas e distribuições estatísticas deste perfil de contorno típico é então formado. No módulo de detecção anômalo é calculado as probabilidades de um

certo comportamento corresponder a um perfil de contorno obtido no treinamento, e também, é calculado estatisticamente o seu grau de anormalidade. No módulo de predição, trajetórias parciais são comparadas com os perfis de contornos obtidos do treinamento para que comportamentos futuros sejam estocásticamente inferidos.

O trabalho de Zhou e Kimber [10] mostra um *framework* que combina múltiplas direções de captura de cenas de vídeo em um nível de inferência, utilizando também Modelos de *Markov* (CHMM). São utilizados dois estágios de treinamento para obter um conjunto de eventos usuais e assim treinar estes Modelos de *Markov*. Estabelecendo-se um limiar de semelhança para um segmento de teste, que foi gerado pelo modelo, é possível construir um detector de eventos não usuais. Este é um modelo semi-supervisionado que depende do número de interações. Assim, o treinamento é feito em dois estágios ilustrado na Figura 2.17.

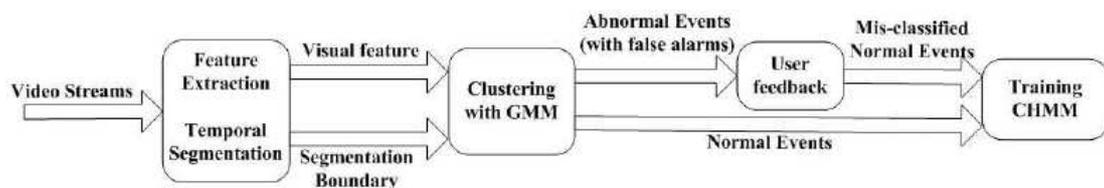


Figura 2.17 – Estágios do treinamento na técnica de Zhou e Kimber [10].

No primeiro estágio é gerado um conjunto de eventos usuais combinando *clusters*, que são supervisionados através de uma realimentação (*feedback*) do usuário, ou seja, uma pré aceitação por parte do usuário que pode reavaliar alguns dos comportamentos capturados pelos vídeos. Em um segundo estágio, o modelo é treinado precisamente com um conjunto definido de eventos usuais sem intervenção do usuário. O diagrama da Figura 2.18, bem como a anterior, demonstram como o processo é feito em todas as suas etapas.

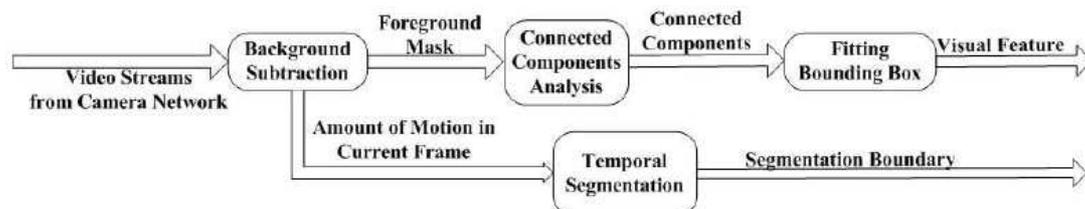


Figura 2.18 – Extração das imagens na técnica de Zhou e Kimber [10].

O processamento do algoritmo de acompanhamento de objetos é feito de maneira semelhante ao citado anteriormente por Nair [2], usando técnicas de subtração de *background*. Os eventos são organizados em *clusters* utilizando o modelo de Misturas de Gaussianas GMM (*Gaussian Mixture Model*). Estes *clusters* organizam eventos que possuem características similares, mantendo-os agrupados por classes. Uma vez estabelecida as classes, que normalmente possuem um conjunto grande de eventos de mesma característica, ainda poderão ocorrer classes com números pequenos de eventos que não puderam serem agrupados. Assim, estas serão re-agrupadas novamente, reajustando-as com os existentes ou criando novos. É uma maneira de obter um ajuste fino na coleta destes eventos e impedir que posteriormente estes venham a ser consideradas não usuais. Em um segundo estágio, é treinado o Modelo de *Markov* com um conjunto de eventos usuais diretamente. Dispondo-se de uma seqüência de vídeo para teste, é possível avaliar a vizinhança em que o segmento esta sendo gerado pelo Modelo de *Markov*. Se a semelhança for baixa, se comparado a um limiar estabelecido, o segmento é classificado como não usual.

O trabalho de Jung et al. [19] apresenta uma forma de detecção de comportamentos estranhos baseado em trajetórias. A detecção de movimento não usual é baseada em dois aspectos: ocupação espacial e coerência das trajetórias. A ocupação é a mesma descrita em Soldera [11]. A coerência da trajetória é explorada pela classificação das mesmas em um período de treinamento utilizando GMMs, e avaliando a pertinência de cada nova trajetória no período de teste com as classes obtidas.

Neste trabalho, pretende-se ampliar o conceito do SOM implantado por Soldera [11], incorporando a informação de velocidade instantânea e coerência de trajetórias. Mais especificamente, pretende-se agrupar trajetórias coerentes em *clusters* e gerar histogramas $4D$ considerando novas dimensões como posição, orientação do deslocamento e velocidade, que descrevam o movimento em cada *cluster*. Para uma nova trajetória a ser analisada, seu comportamento é comparado ao histograma $4D$ de cada *cluster* obtido primeiramente, na fase de treinamento, verificando sua coerência. Em seguida será submetida ao período de teste ou operação onde as informações do histograma $4D$ serão confrontadas com as novas trajetórias. Tal idéia será descrita no próximo Capítulo.

Capítulo 3

O Método Proposto

3.1 Introdução

O método que começaremos a descrever está baseado, fundamentalmente, na interpretação de comportamentos de trajetórias extraídas de cenas de vídeo. A idéia geral do método é utilizar um conjunto de trajetórias em um período de treinamento, que o algoritmo irá utilizar como exemplos usuais. Na fase de teste, cada nova trajetória será confrontada com aquelas do treinamento: se for coerente, será considerada usual; caso contrário, não usual. Normalmente, trajetórias usuais ocorrem em caminhos pré-determinados, como passagens de pedestres (conforme ilustrado na Figura 3.1).

Para a aplicação na prática do algoritmo proposto, seria ideal a captura automática das trajetórias, usando algoritmos de visão computacional [18, 19, 14, 4, 16, 20, 21, 22, 13]. Entretanto, o foco deste trabalho não é a captura das trajetórias, mas sim a fase posterior: a análise das mesmas. Dessa forma, assume-se neste trabalho que as trajetórias já são conhecidas.

Este trabalho estende o algoritmo proposto por Soldera [11], incluindo diversos pontos de aperfeiçoamento. Um dos problemas que o método de Soldera [11] apresenta é considerar em sua análise das trajetórias somente a ocupação espacial (bidimensional), ou seja, a coerência da posição de cada ponto da trajetória sendo analisada com um histórico de ocupação espacial obtido no período de treinamento, desconsiderando a velocidade instantânea. Assim, pessoas se movimentando no sentido contrário do fluxo normal de pessoas seriam consideradas usuais.

O método aqui proposto considera quatro dimensões, ou seja, utiliza a

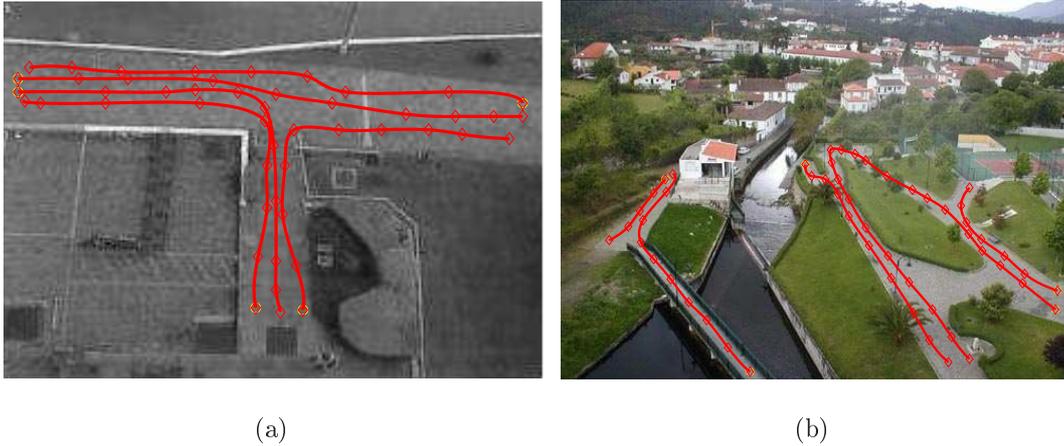


Figura 3.1 – Exemplo de trajetórias normais.

ocupação espacial bidimensional, de Soldara, acrescida da direção e módulo da velocidade de cada pessoa (ou seja, o **espaço-fase** será analisado). Assim, será obtido um histograma $4D$ do histórico de movimentação das pessoas no período de treinamento, que será utilizado para validar trajetórias no período de teste. Além da inclusão da informação de velocidade, este trabalho agrupa trajetórias em classes coerentes, conforme será descrito ao longo deste Capítulo.

3.2 O Histograma $4D$ - Espaço-Fase

O modelo de percepção de trajetórias suspeitas está fundamentado na coerência da trajetória de teste com as de treinamento conforme quatro critérios (correspondendo à 4 dimensões no histograma). A cada instante, são extraídas as coordenadas espaciais (x, y) da pessoa no plano, além de sua velocidade instantânea (v_x, v_y) ¹. Assim, cada ponto de uma trajetória corresponde a um ponto no espaço $4D$ (x, y, θ, v) . Com a utilização de diversas trajetórias no período de treinamento, se obtém um histograma de ocupação $4D$, que será utilizado na avaliação das trajetórias.

Na implementação do algoritmo, o histograma $4D$ é um arranjo com quatro dimensões, inicializado com zeros. Na medida que o tempo vai transcorrendo, as pessoas são filmadas percorrendo esta área monitorada, e o histograma é atualizado. A idéia básica é incrementar em uma unidade a posição (x, y, θ, v) do arranjo, cada

¹O vetor velocidade instantânea também pode ser caracterizado pelo par (θ, v) , onde θ denota a orientação do vetor, e v seu módulo.

vez que alguma pessoa passar na posição (x, y) com velocidade (em módulo) v e orientação θ . Entretanto, tal abordagem pode ser muito sensível a variações na obtenção das coordenadas (x, y, θ, v) das pessoas filmadas (pequenas diferenças nesses parâmetros acarretam no incremento de posições diferentes do histograma), além de requerer uma grande quantidade de trajetórias para uma estimativa confiável do histograma.

Portanto obter um histograma $4D$ no **espaço-fase** equivale a dizermos que, obteremos as coordenadas espaciais (x, y) da pessoa no plano, o módulo e orientação da velocidade no decorrer do tempo em cada ponto do arranjo sendo dispostas em um vetor. A abordagem proposta para a construção do histograma $4D$ é dada a seguir.

3.2.1 Quantização do Histograma

A primeira questão a ser analisada é a quantização das coordenadas (x, y, θ, v) no cálculo do histograma. Para os parâmetros espaciais (x, y) , a discretização é inerentemente dada pelo tamanho da imagem digital representando a imagem analisada. Sendo $N_x \times N_y$ a resolução da imagem (em pixels), então x é quantizado em N_x níveis e y em N_y níveis, ou seja, os valores discretizados x^d e y^d podem assumir os valores:

$$x^d \in \{0, 1, \dots, N_x - 1\}, \quad y^d \in \{0, 1, \dots, N_y - 1\}. \quad (3.1)$$

A orientação θ do vetor velocidade será quantizado em N_θ níveis (neste trabalho utilizaremos $N_\theta = 8$), sendo que cada um compreende um setor circular com ângulo interno $2\pi/N_\theta$. Assim, os ângulos discretizados θ^d podem assumir os valores

$$\theta^d \in \{\theta_0, \theta_1, \dots, \theta_{N_\theta-1}\}, \quad (3.2)$$

onde os valores θ_j são as bissetrizes de cada setor circular, dados por

$$\theta_j = \frac{j}{N_\theta} 2\pi, \quad \text{para } j = 0, \dots, N_\theta - 1. \quad (3.3)$$

Analogamente, o módulo da velocidade é quantizado em N_v níveis, e então os módulos de velocidade discretizados v^d podem assumir os valores

$$v^d \in \{0, 1, \dots, N_v - 1\}. \quad (3.4)$$

Em particular, pretende-se utilizar um valor baixo para N_v (mais precisamente, $N_v = 3$), quantizando o módulo da velocidade nos níveis baixo, médio e alto, de acordo com:

$$v^d = \begin{cases} 0 \text{ (velocidade baixa),} & \text{se } 0 \leq v \leq v_b \\ 1 \text{ (velocidade média),} & \text{se } v_b \leq v \leq v_a \\ 2 \text{ (velocidade alta),} & \text{se } v \geq v_a \end{cases}, \quad (3.5)$$

onde v_b é um limiar de velocidade baixa, e v_a é um limiar de velocidade alta. Tais valores devem ser determinados com base na aplicação desejada, mas este trabalho propõe uma seleção automática com base nas trajetórias do período de treinamento.

Para determinar v_a e v_b automaticamente, assumimos que a maioria das trajetórias possui velocidade (em módulo) na faixa média na maior parte do tempo, com algumas pessoas andando mais rápido ou mais devagar. Assim, é gerado um vetor contendo as velocidades de todas as trajetórias em todos instantes de tempo, sendo calculado a média μ_v e o desvio padrão σ_v . Os limiares v_a e v_b são determinados com base no desvio da média, ponderados pelo desvio padrão:

$$v_a = \mu_v + k\sigma_v \text{ e } v_b = \mu_v - k\sigma_v, \quad (3.6)$$

onde k controla a extensão da faixa de velocidade média nas trajetórias. Neste trabalho foi utilizado $k = 2$.

Em resumo, o histograma $4D$ será representado por um arranjo de dimensões $N_x \times N_y \times N_\theta \times N_v$. Cada vez que uma pessoa i é detectada no tempo t , ela é caracterizada através de suas coordenadas espaciais $(x_i(t), y_i(t))$ e coordenadas de velocidade $(\theta_i(t), v_i(t))$. Tais coordenadas correspondem a 4 posições $(x_i^d(t), y_i^d(t), \theta_i^d(t), v_i^d(t))$ no histograma, onde $x_i^d(t) = x_i(t)$, $y_i^d(t) = y_i(t)$, e $\theta_i^d(t)$, $v_i^d(t)$ são valores discretizados obtidos através das relações (3.3) e (3.5), respectivamente. Ao invés de incrementar apenas a coordenada $(x_i^d(t), y_i^d(t), \theta_i^d(t), v_i^d(t))$ do histograma, será preenchida uma região em torno de $(x_i^d(t), y_i^d(t), \theta_i^d(t), v_i^d(t))$. Com isso, a imprecisão no cálculo de (x, y, θ, v) é diluída. Tal abordagem é discutida a seguir.

3.2.2 Preenchimento do Histograma

A idéia no preenchimento do histograma é usar um núcleo de preenchimento, que afeta uma região em torno da coordenada quantizada $(x_i^d(t), y_i^d(t), \theta_i^d(t), v_i^d(t))$.

Tal preenchimento é dado por:

$$H(x^d, y^d, \theta^d, v^d) = \sum_{i=1}^N \sum_{t=1}^{N_f(i)} g(x^d - x_i^d(t), y^d - y_i^d(t), \theta^d - \theta_i(t), v^d - v_i^d(t)), \quad (3.7)$$

onde N é o número de trajetórias avaliadas, $N_f(i)$ é a duração da i -ésima trajetória, e a função $g(x, y, \theta, v)$ é o núcleo que gerará o espalhamento em torno do ponto $(x_i^d(t), y_i^d(t), \theta_i^d(t), v_i^d(t))$. Tal função $g(x, y, \theta, v)$ será descrita com um produto separável, que desacopla o espalhamento nas dimensões espaciais (x e y), na dimensão da orientação (θ) e no módulo da velocidade (v). Ou seja,

$$g(x, y, \theta, v) = g_1(x, y)g_2(\theta)g_3(v). \quad (3.8)$$

A função $g_1(x, y)$ escolhida é um núcleo Gaussiano, implementado através uma função Gaussiana bidimensional discreta truncada descrita por

$$g_1(x, y) = g_\sigma(x, y) = \begin{cases} \frac{1}{c} e^{-\frac{x^2+y^2}{2\sigma^2}}, & \text{se } -2\sigma \leq x, y \leq 2\sigma \\ 0 & \text{caso contrário} \end{cases}, \quad (3.9)$$

sendo que c é a constante de normalização segundo a condição

$$\sum_{(x,y)} g_\sigma(x, y) = 1. \quad (3.10)$$

De fato, tal Gaussiana foi utilizada em [11, 19], com o objetivo de atenuar as imprecisões na obtenção das trajetórias, minimizando principalmente, o efeito da deformação da perspectiva quando forem utilizados trajetórias capturadas por câmeras não situadas no topo das cenas monitoradas. Além disto levaremos em consideração o tamanho de uma pessoa (vários algoritmos de acompanhamento de pessoas retornam apenas o ponto central do objeto acompanhado, e a Gaussiana estende a influência espacial para compreender o espaço ocupado pela pessoa). O valor do desvio padrão σ deve ser escolhido com base no tamanho médio esperado de uma pessoa ou objeto nas imagens capturadas, sendo sua unidade o pixel. A Figura 3.2 mostra exemplos de trajetórias utilizadas no período de treinamento e o resultado do histograma com o núcleo Gaussiano utilizando diferentes valores de σ a partir da Figura original (a); $\sigma = 5$ em (b), $\sigma = 15$ em (c) e $\sigma = 30$ em (d). Vale lembrar que somente são mostradas as duas primeiras dimensões do histograma $4D$.

A função $g_2(\theta)$, responsável pelo espalhamento da dimensão da orientação, leva em consideração o setor circular ao qual $\theta_i(t)$ pertence (cujo centro é $\theta_i^d(t)$)

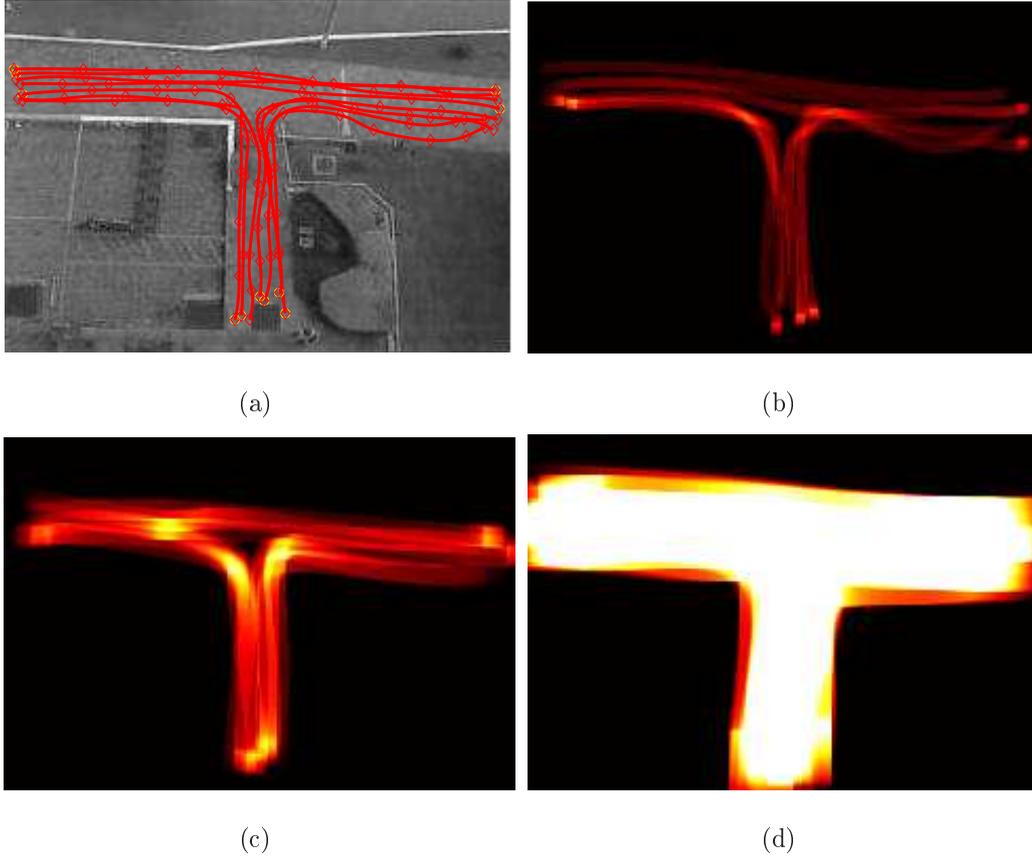


Figura 3.2 – (a) $\sigma = 1$ sem espalhamento. (b) $\sigma = 5$. (c) $\sigma = 15$. (d) $\sigma = 30$.

e os setores vizinhos. A idéia é ponderar a influência de cada setor circular com base na distância de $\theta_i(t)$ ao centro do setor circular $\theta_i^d(t)$: se $\theta_i(t)$ estiver no centro de um setor circular, tal setor carregará um peso grande no cálculo do histograma. Entretanto, se $\theta_i(t)$ estiver perto da fronteira entre dois setores, cada um destes terá um peso semelhante no cômputo do histograma.

Mais precisamente, a influência em cada nível θ será dada por

$$g_2(\theta) = \max \left\{ 0, 1 - \frac{|\theta|_{\text{ang}}}{\Delta\theta_{N_\theta}} \right\}, \quad \text{para } \theta \in \{\theta_0, \dots, \theta_{N_\theta-1}\}, \quad (3.11)$$

onde θ_j denotam os ângulos discretizados conforme a Equação (3.3), $\Delta\theta_{N_\theta} = \frac{2\pi}{N_\theta}$ é o ângulo interno do setor circular utilizado na discretização de θ , e

$$|\theta|_{\text{ang}} = \min \{|\theta|, 2\pi - |\theta|\} \quad (3.12)$$

representa a distância angular. Pode-se notar que $g_2(\theta^d - \theta_i(t))$ assume o valor máximo ($= 1$) quando $\theta_i(t)$ coincide com o centro dos setores circulares quantizados (ou seja, $\theta_i(t) \in \{\theta_0, \theta_1, \dots, \theta_{N_\theta-1}\}$) e $\theta^d = \theta_i(t)$.

A Figura 3.3 ilustra as variáveis utilizadas na Equação (3.11). Esta regra vale para qualquer número de setores circulares que necessitemos implementar, utilizando a mesma lógica de separação e sobreposição de setores circulares. Quanto maior a quantidade de sobreposições criadas, maior a capacidade de localização do vetor velocidade em um respectivo setor circular e maior a certeza (sensibilidade) do seu sentido. Por outro lado, maior é a quantidade de memória utilizada. Testamos o algoritmo com diversos N_θ e concluimos que, com $N_\theta = 8$ obtemos um bom comprometimento entre localização (precisão em um determinado setor circular) e memória.

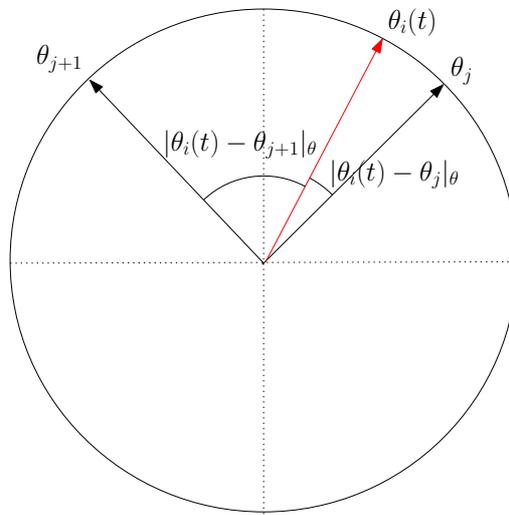


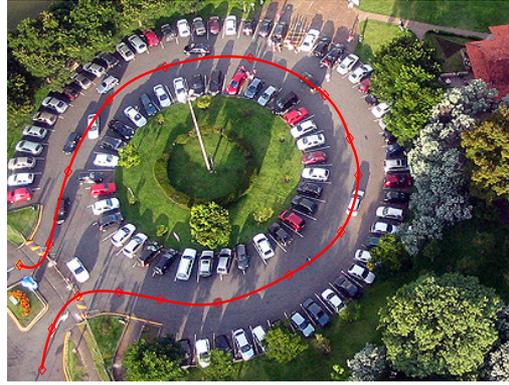
Figura 3.3 – Setores circulares.

A Figura 3.4 ilustra o efeito da terceira dimensão (θ) do histograma para capturar informações de orientação local de trajetórias. Um exemplo de trajetória circular é mostrado na Figura 3.4(a), e “fatias” do histograma ao longo da terceira dimensão θ (para diferentes orientações) são ilustradas na Figura 3.4(b). Percebe-se que cada pedaço da trajetória é capturado pela respectiva orientação local.

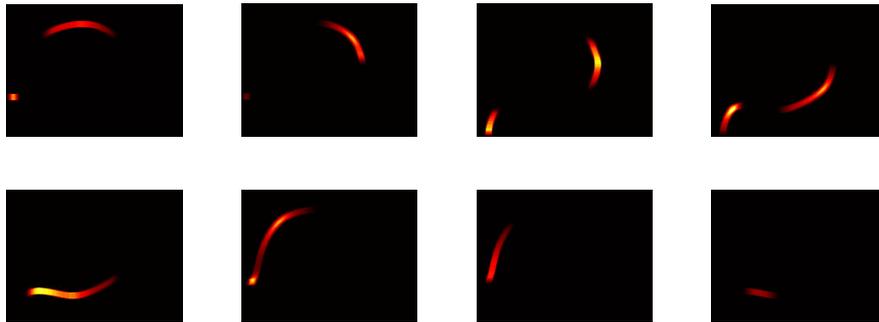
Com relação a quarta dimensão v , deve-se mencionar que a idéia é utilizar uma quantização N_v em poucos níveis ($N_v = 3$ é a sugestão deste trabalho). Assim, não faz muito sentido espalhar a influência de v nos vizinhos, e então o núcleo $g_3(v)$ sugerido é dado por

$$g_3(v) = \delta(v), \quad (3.13)$$

onde $\delta(v)$ é a função Delta de Dirac discreta.



(a)



(b)

Figura 3.4 – Pedacos da trajetória orientadas nas “fatias” ao longo da terceira dimensão θ do histograma.

3.3 Agrupamento de Trajetórias

O histograma $4D$, descrito até então, permite que se detecte a coerência local de uma nova trajetória em relação a um conjunto de trajetórias utilizadas em uma fase de treinamento. Entretanto, a coerência temporal da trajetória é perdida quando se calcula o histograma: duas trajetórias com comportamento global bastante distintos podem ser coerentes localmente em alguns pontos. Por exemplo, considere as trajetórias da Figura 3.5. Nesse caso, há um grupo de trajetórias coerentes da esquerda para a direita, e outro grupo de trajetórias coerentes subindo ao centro e depois dobrando à esquerda. Uma nova trajetória subindo ao centro e dobrando à direita é localmente coerente com o histograma (na primeira parte é coerente com o primeiro grupo de trajetórias, e na segunda parte com o segundo grupo) ao longo de toda trajetória. Entretanto, percebe-se que ela apresenta um comportamento global distinto de todas as outras trajetórias utilizadas no treinamento.

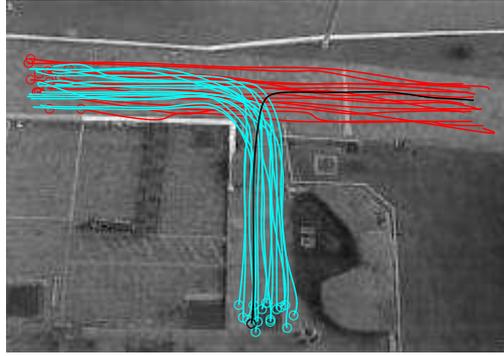


Figura 3.5 – Trajetória em preto incoerente com o treinamento.

Para evitar esse tipo de situação, utiliza-se neste trabalho um algoritmo que agrupa trajetórias em classes semelhantes (*clustering*), preservando a coerência temporal, mantendo o agrupamento de trajetórias com as mesmas características de contorno (veja Figura 3.6). Métodos de agrupamento de trajetórias são abordados brevemente nos trabalhos de Johnson e Hogg e Lou et al. [7, 6]. Já o trabalho de Jung et al. [19] implementa um algoritmo semelhante ao que trataremos neste item.

A idéia básica é extrair, para cada trajetória, um vetor de feições que a caracteriza. Assim sendo, trajetórias semelhantes gerariam vetores similares. Dessa forma, cada conjunto de vetores de feições associados às trajetórias coerentes formariam um aglomerado de pontos, que podem ser detectados utilizando algoritmos de *clustering* [23, 24, 25, 26, 27, 28, 29, 30]. Independente do algoritmo de *clustering* adotado, o primeiro passo é a extração de vetores de feição para cada trajetória. Pretende-se adotar uma estratégia similar a usada em Jung et al. [19], descrita a seguir.

Seja $(x^n(s), y^n(s))$ para $0 \leq s \leq 1$ a parametrização de uma trajetória $(x(t), y(t))$ com relação ao comprimento de arco s , de modo que $(x^n(0), y^n(0))$ represente o ponto inicial de uma trajetória e $(x^n(1), y^n(1))$ o ponto final da mesma. Um modo de obter o vetor de feições \mathbf{x} é concatenar N deslocamentos, mantendo $(\Delta x_i, \Delta y_i)$ contabilizados em comprimentos de arcos equidistantes como

$$(\Delta x_i, \Delta y_i) = (x(s_{i+1}) - x(s_i), y(s_{i+1}) - y(s_i)), \quad (3.14)$$

onde $s_i = \frac{i}{N}, i = 0, \dots, N - 1$. Portanto, quando tivermos uma trajetória j em um plano bidimensional obteremos um vetor \mathbf{x}_j , através da concatenação dos N vetores

de deslocamento associados com a trajetória:

$$\mathbf{x}_j = (\Delta x_0, \Delta y_0, \Delta x_1, \Delta y_1, \dots, \Delta x_{N-1}, \Delta y_{N-1}). \quad (3.15)$$

Assim, trajetórias com mesma característica de contorno produzirão vetores com características similares, gerando um *cluster* $2N$ -dimensional que será modelado por uma distribuição de probabilidade Gaussiana. Sendo k uma classe representando trajetórias coerentes, esta é caracterizada pela média $\boldsymbol{\mu}_k$, matriz de covariância \mathbf{C}_k e probabilidade *a priori* w_k . Cada *cluster* terá uma função Gaussiana diferente e a distribuição genérica $p(\mathbf{x})$ que considera a existência de todos os vetores, pode ser obtida pela mistura de Gaussianas

$$p(\mathbf{x}) = \sum_{k=1}^{k_{\max}} w_k p_k(\mathbf{x}), \quad (3.16)$$

sendo que k_{\max} é o número de classes e

$$p_k(\mathbf{x}) = \frac{(2\pi)^{-N}}{\sqrt{|\mathbf{C}_k|}} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_k)^T \mathbf{C}_k^{-1} (\mathbf{x} - \boldsymbol{\mu}_k) \right\} \quad (3.17)$$

é a distribuição normal associada com a k -ésima classe. Esse número de Gaussianas corresponde ao número de *clusters* e cada vetor com sua característica adquirida durante o período de treinamento atribuída a uma classe k é dada por

$$k = \operatorname{argmax}_{j \in \{1, \dots, k_{\max}\}} w_j p_j(\mathbf{x}), \quad (3.18)$$

o que equivale a selecionar a classe com a maior probabilidade *a posteriori* de acordo com a regra de *Bayes*. Dado o conjunto de vetores de feição \mathbf{x}_j , obtidos a partir das trajetórias no período de treinamento, o número de classes k_{\max} , assim como os vetores $\boldsymbol{\mu}_k$, matrizes \mathbf{C}_k e probabilidades w_k podem ser obtidas automaticamente usando o algoritmo descrito em Figueiredo e Jain [25].

O número de vetores de deslocamento N , usado para criar \mathbf{x}_j , deve ser escolhido com base na complexidade das trajetórias. Para trajetórias mais simples, pode-se reproduzir o comportamento global das mesmas, com valores baixos de N . Por outro lado, se as trajetórias forem mais complexas, elas serão melhor representadas utilizando-se valores maiores de N . Também, deve-se notar que quanto maior o valor de N , maior a dimensionalidade dos vetores de feição \mathbf{x} . Assim, é necessária uma maior quantidade de trajetórias (amostras) para obter a distribuição das misturas de Gaussianas de forma confiável.

A Figura 3.6(a) mostra um exemplo bastante simples de formação de *clusters* utilizando o ambiente em T, onde obtemos um conjunto de trajetórias no período de treinamento (formando diferentes classes de trajetória coerentes). Os vetores de feição \mathbf{x}_j , referentes a essas trajetórias (usando $N = 1$), são mostrados nesta figura em cores diferentes. Dispostos de trajetórias partindo do caminho central dobrando a direita e vice-versa e também trajetórias partindo da esquerda para direita. A Figura 3.6(b) mostra as classes obtidas (foram detectadas 3 classes).

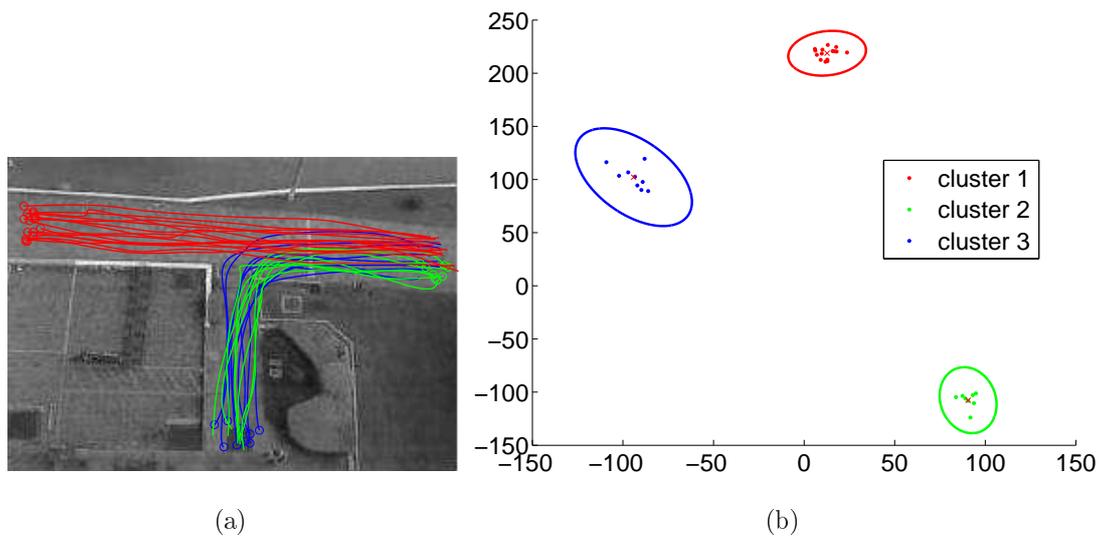


Figura 3.6 – Exemplo de formação de *clusters*.

Após a aplicação do algoritmo de *clustering* às trajetórias no período de treinamento, são obtidas k_{\max} classes de trajetórias coerentes. Assim sendo, as trajetórias de cada classe são utilizadas na construção de um histograma $4D$ distinto (conforme descrito nas seções anteriores). Dada uma nova trajetória no período de teste, a idéia é comparar sua coerência com cada um dos k_{\max} histogramas $4D$ calculados no período de treinamento.

Embora o emprego de *clusters* forneça resultados satisfatórios, é preciso lembrar que quando dispusermos um volume grande de fluxo de pessoas ou objetos em uma cena durante um período de treinamento, poderemos ter uma variedade bastante significativa de variações de trajetórias. Imaginemos o ambiente da Figura 3.7(a) onde foram identificadas apenas duas trajetórias com características distintas das demais no período de treinamento. Após aplicado o processo de *clustering* podem ocorrer situações em que os vetores de feições de algumas trajetórias se encontrem a uma distância consideravelmente grande dos demais *clusters* como

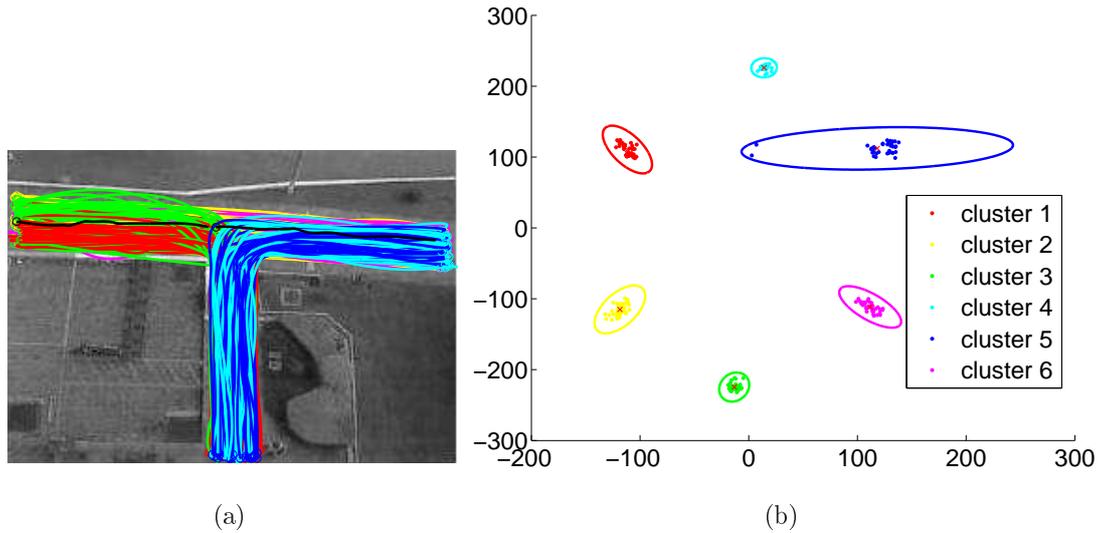


Figura 3.7 – A presença de *outliers*.

mostradas na Figura 3.7(b). O algoritmo que calcula automaticamente a mistura de Gaussianas tenta incorporá-las a algum *cluster*, proporcionando um agrupamento deficiente. Isso motivou a utilização de uma técnica para remoção de *outliers*, que será descrito a seguir.

3.3.1 Remoção de *Outliers*.

O estudo e a remoção dos *outliers* tem sido objeto de pesquisas, principalmente quando necessitamos empregar estruturas de dados multivariáveis [31]. É tido como *outlier* em uma amostragem com dados brutos (contaminada), qualquer amostra que seja significativamente diferente das demais. No contexto deste trabalho, *outliers* são gerados por trajetórias que não são coerentes com nenhum *cluster*.

O trabalho de Peña e Prieto [31] apresenta um algoritmo de remoção de *outliers*, baseado na análise de projeções a partir de uma amostragem de pontos em uma determinada quantidade de direções. Estes pontos serão analisados num plano bidimensional e as direções são obtidas maximizando e minimizando o coeficiente da curtose² das mesmas.

O uso de projeções para identificar *outliers* é a base de muitas aplicações para

²O coeficiente da curtose é um indicador de “achatamento” de uma curva da função de densidade, definindo valores mais altos, gerando curvas afuniladas e mais baixos gerando curvas achatadas. As curvas, também, são conhecidas como: leptocúrtica é uma curtose positiva, mesocúrtica uma curtose nula e a laticúrtica é conhecida como uma curtose negativa.

este tipo de detecção. Estas aplicações se baseiam no fato de que, em amostragens multivariadas contaminadas, cada *outlier* deve ser um ponto extremo ao longo de uma direção, afastando-se da média de dados não contaminados. Outros métodos desenvolvidos ao longo do tempo requerem projetar dados para randomicamente, gerar direções e precisam de um número muito grande destas direções para obter sucesso na detecção. O sucesso destes métodos melhorou consideravelmente sob ponto de vista computacional, mesmo que, ainda seja usado um número limitado de direções. Na abordagem de Peña e Prieto [31], as direções são geradas pela variação nos valores atribuídos ao coeficiente de curtose. Tal método é brevemente descrito a seguir, e maiores detalhes do algoritmo estão na referência bibliográfica.

Inicialmente, uma seleção inicial de possíveis *outliers* é feita com base em projeções sucessivas. As amostras que não foram marcadas como possíveis *outliers* formam o conjunto U , cuja média $\boldsymbol{\mu}_n$ e uma matriz de covariância \mathbf{C}_n são calculados. Para todas as observações $\mathbf{x}_i \notin U$ será calculada a distância de *Mahalanobis*

$$d_i = (\mathbf{x}_i - \boldsymbol{\mu}_n)^T \mathbf{C}_n^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_n). \quad (3.19)$$

Se essa distância satisfizer a relação $d_i < \chi_{2N,\alpha}^2$, a observação \mathbf{x}_i não será considerada *outlier*, sendo excluído da lista criada preliminarmente e incluída no conjunto U . É importante observar que o limiar $\chi_{2N,\alpha}^2$ representa o percentil α de uma distribuição Chi-quadrada com $2N$ graus de liberdade. No algoritmo proposto, é utilizado um valor $\alpha = 0,9999$, de modo que apenas observações significativamente distantes do centro de um *cluster* serão consideradas *outliers*, com probabilidade de erro de 0,01%. A média e covariância de U são recalculados e o processo é repetido até que todos os *outliers* de cada *cluster* sejam identificados e marcados, sendo posteriormente excluídos.

O tratamento dos *outliers* utilizado no pós processamento à classificação inicial gerou resultados bastante satisfatórios. À partir da base de dados obtidas no treinamento mostrado na Figura 3.7(a) foi utilizado o algoritmo de remoção de *outliers*, e desta forma, as duas trajetórias foram identificadas e removidas. O resultado final da classificação é mostrado na Figura 3.8.

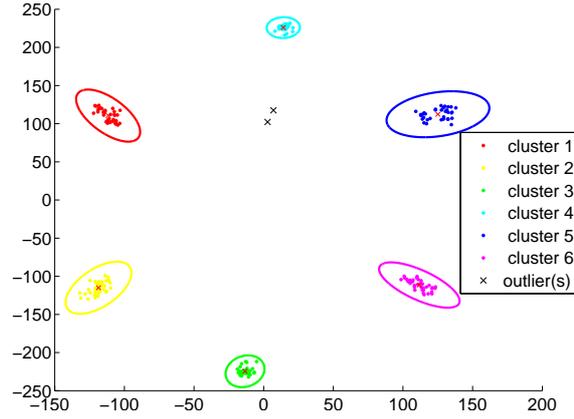


Figura 3.8 – Remoção de *outliers*.

3.4 Período de Teste ou Operação.

Quando concluimos o treinamento, o modelo terá executado o levantamento e armazenamento no seu banco de dados e o perfil de todas trajetórias aprendidas. Estas estarão acumuladas e divididas em k_{\max} *clusters*. O histograma $4D$ é calculado utilizando as trajetórias de cada cluster, ou seja, o resultado da fase de treinamento é um conjunto de histogramas $H_k(x^d, y^d, \theta^d, v^d)$, para $k = 1, \dots, k_{\max}$.

A nova etapa é monitorar cada nova trajetória alimentada para o sistema, verificando sua coerência com cada um dos histogramas computados na fase de treinamento. Mais precisamente, para cada ponto $(x(t), y(t))$ de uma nova trajetória a ser analisada se obtém as coordenadas de espaço-fase discretizadas $(x^d(t), y^d(t), \theta^d(t), v^d(t))$, onde $t = 1, \dots, N_f$ denota o tempo, e N_f é a duração da trajetória. Assim, pode-se verificar a cada instante de tempo t se a nova trajetória está coerente com o k -ésimo histograma H_k , analisando a função

$$d_k(t) = H_k(x^d(t), y^d(t), \theta^d(t), v^d(t)). \quad (3.20)$$

Se a trajetória for localmente consistente com o k -ésimo histograma, os valores $d_k(t)$ serão altos; caso contrário, os valores de $d_k(t)$ serão mais próximos de zero. Dessa forma, pode-se avaliar (ao longo do tempo t) se uma nova trajetória está consistente com algum dos histogramas computados no treinamento, permitindo a detecção de porções da trajetória como usuais ou não usuais.

A definição do que será considerado não usual deverá ser referenciado ao período de treinamento inicial. Vale lembrar que, neste trabalho, as trajetórias

são agrupadas e guardadas no banco de dados, constituindo-se a base para uma posterior comparação e avaliação para novas trajetórias no período de teste ou operação. Com base no treinamento é que será possível um observador humano (agente de segurança), decidir se uma trajetória que caracteriza um comportamento oferece algum risco pessoal ou material. Uma vez estabelecido o número k_{\max} de *clusters*, atribuímos limiares de coerência para cada *cluster* k . Assim, podemos definir se uma nova trajetória pode ser considerada usual ou não usual no tempo t se

$$d_k(t) < T_k, \quad (3.21)$$

ou seja, se sua coerência local com o *cluster* k é menor do que um certo limiar T_k . Tal limiar deve ser selecionado individualmente para cada *cluster*, para que um diferente número de trajetórias possam ser detectadas em cada *cluster*. Vamos considerar $(x_i(t), y_i(t))$, para $i = 1, \dots, N_k$ o número de trajetórias pertencentes ao *cluster* k obtido em um período de treinamento. Para cada trajetória, é calculada a coerência $d_k^i(t)$ em cada instante de tempo t de acordo com a Equação (3.20) gerando um conjunto

$$P_k = \bigcup_{i=1}^{N_k} \bigcup_{t=1}^{N_f(i)} \{d_k^i(t)\}, \quad (3.22)$$

que acumula a coerência individual de todas as trajetórias em um *cluster* k . Uma possibilidade para selecionar T_k seria para recuperar um valor mínimo em P_k , que define a mínima coerência observada no período de treinamento. Entretanto, algumas trajetórias neste período de treinamento podem não ser totalmente coerentes com o histograma $4D$ em todos os pontos, o que pode gerar valores muito baixos. Conseqüentemente, selecionamos T_k como sendo o valor que remove uma fração r dos menores valores no conjunto P_k , ou seja, T_k é r -percentil da distribuição de $d_k^i(t)$. Neste trabalho, foi utilizado $r = 0.1$, de modo que T_k separa os 10% menores valores de P_k .

Tomamos como exemplo um conjunto de trajetórias simples, que se agrupam em dois *clusters* somente conforme Figura 3.9. Possuímos trajetórias partindo do caminho central e dobrando à direita e outras que partem da direita para a esquerda.

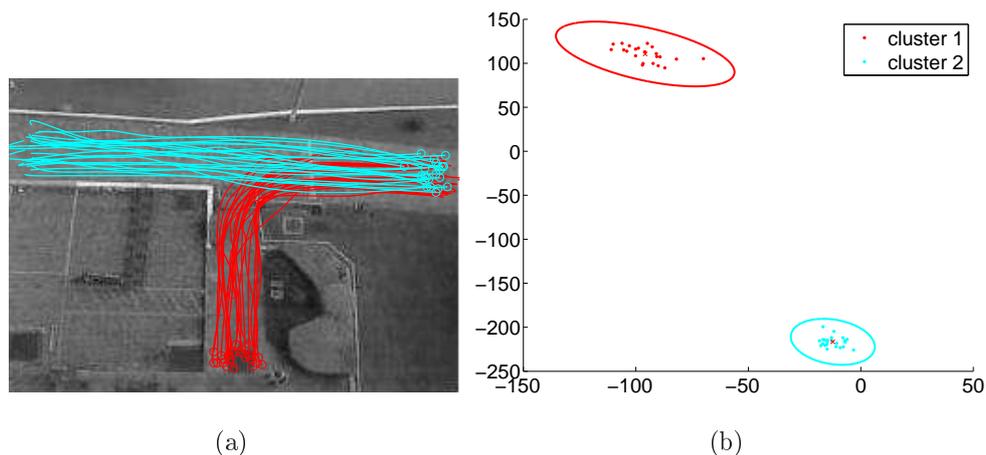


Figura 3.9 – (a) Trajetórias usadas para treinamento. (b) Os *clusters* criados.

Observamos na Figura 3.10(a) o percurso de uma trajetória somente, que será analisada no período de teste. A Figura 3.10(c) mostra a coerência dessa trajetória com os histogramas das duas classes mostradas na Figura 3.9(b). Percebe-se que os valores de $d_2(t)$ (referentes ao *cluster 2*) são praticamente nulos, indicando que a trajetória de teste não é coerente com esse *cluster* em nenhum instante de tempo.

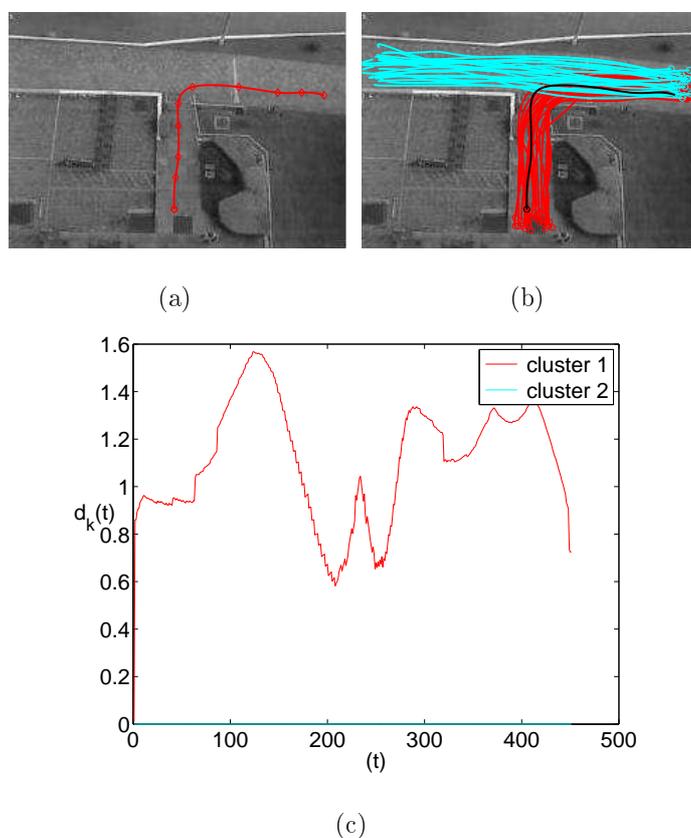


Figura 3.10 – (a) Trajetória coerente com o treinamento usada para teste. (b) Sobreposição desta no treinamento. (c) Gráfico de $d_k(t)$.

Já o gráfico de $d_1(t)$ apresenta valores sempre maiores do que o limiar T_1 , indicando coerência com o primeiro *cluster* durante toda trajetória. Assim, a trajetória é considerada usual.

Capítulo 4

Resultados Obtidos

Neste Capítulo mostraremos alguns resultados obtidos com o método proposto. Como treinamento inicial (utilizado para a obtenção das simulações e resultados), foram inicialmente utilizadas trajetórias inseridas manualmente, através do auxílio de cliques no *mouse*. Numa segunda etapa, as simulações transcorreram a partir de dados obtidos de algoritmos de acompanhamento de objetos extraídos de cenas de vídeo reais. Nestas simulações, foram utilizados 8 níveis de quantização em θ (i.e. $N_\theta = 8$) e 3 níveis em v (i.e. $N_v = 3$), sendo que os limiares v_a e v_b foram extraídos automaticamente de acordo com o procedimento descrito no Capítulo anterior. O valor do desvio padrão utilizado para suavizar as dimensões espaciais do histograma $4D$, conforme a Equação (3.9), foi definido como $\sigma = 15$ em todas as simulações, exceto quando explicitado outro valor.

Vejamos como exemplo a base de treinamento apresentada na Figura 4.1. A Figura 4.1(a) mostra as trajetórias utilizadas no período de treinamento, enquanto que a Figura 4.1(b) mostra as classes de trajetórias obtidas automaticamente.

A Figura 4.2(a) mostra uma trajetória simulada onde uma pessoa caminha pela calçada (região treinada), depois caminha pelo gramado (região não treinada), novamente por uma região treinada e concluindo seu percurso sobre o gramado. A Figura 4.2(b) mostra um gráfico com a coerência temporal $d_k(t)$ entre a nova trajetória e os histogramas $4D$ referentes às 6 classes obtidas no treinamento.

As linhas coloridas, que constam no gráfico, mostram os *clusters* em que a trajetória de teste se enquadrou ao treinamento e obteve as partes do percurso aprovados pelo histograma $4D$. Estas partes são consideradas usuais e as demais,

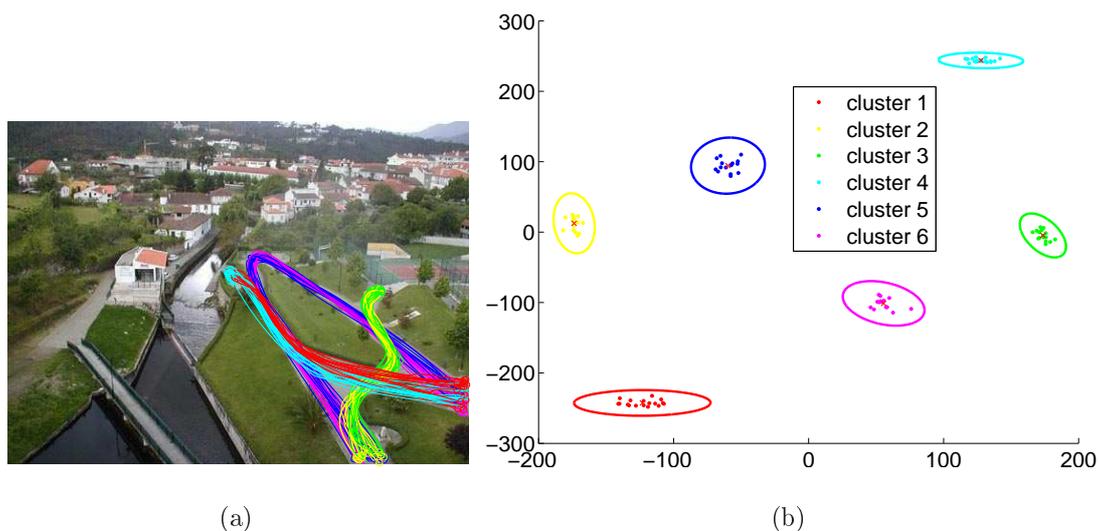


Figura 4.1 – Classes de trajetórias coerentes no período de treinamento.

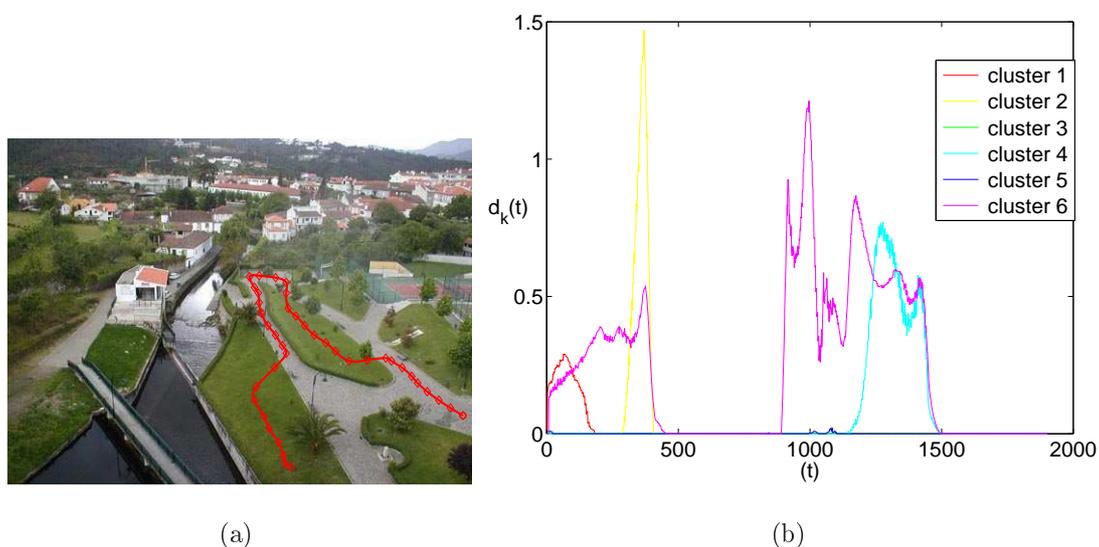


Figura 4.2 – (a) Teste utilizando uma trajetória. (b) Gráfico de $d_k(t)$.

que se apresentam com valores muito baixos, são segmentos que são considerados como não usuais (ou seja, não coerentes com nenhuma classe de trajetórias). Vale novamente lembrar que, quanto mais trajetórias tivermos no período de treinamento, mais estáveis se tornarão os gráficos e as análises e menos suscetível o modelo estará a alarmes falsos.

A Figura 4.3 se refere ao treinamento ilustrado na Figura 3.9(a). Na Figura 4.3(a) observamos uma trajetória atípica, onde uma pessoa percorreu regiões treinadas e não treinadas. A Figura 4.3(b) mostra um gráfico com as regiões (a nível de pixels) que foram contempladas. Regiões do gráfico que indicam valores

baixos, são justamente as mesmas em que esta pessoa percorreu sobre o canteiro, sendo considerado como comportamento não usual. Algumas porções da trajetória foram coerentes com o *cluster 1*, e nenhuma parte foi coerente com o *cluster 2*. Tais resultados estão em consonância com a inspeção visual das trajetórias.

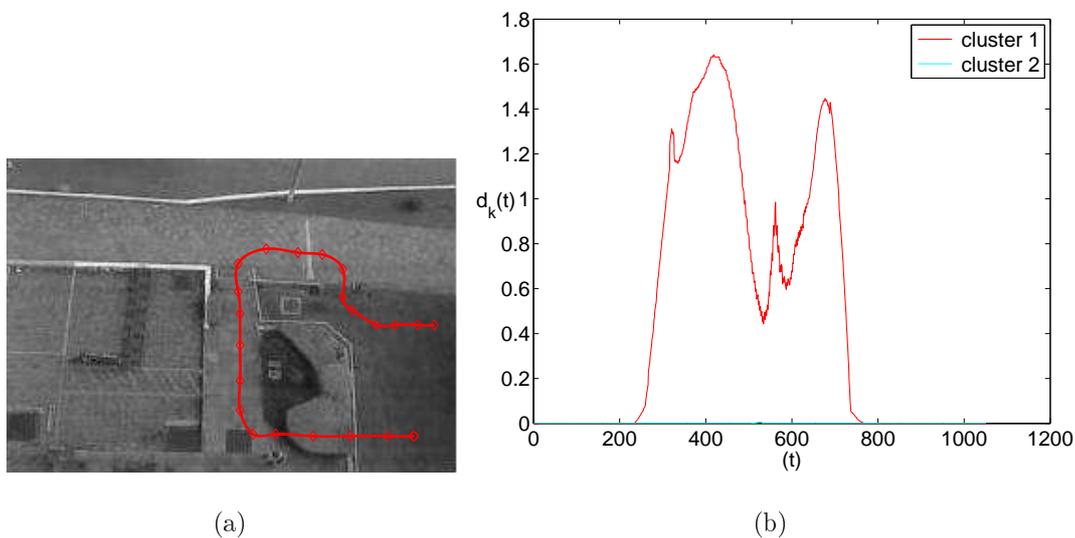


Figura 4.3 – (a) Trajetória não usual. (b) Gráfico de $d_k(t)$.

A Figura 4.4 ilustra outro exemplo de análise de trajetória. A Figura 4.4(a) caracteriza uma transição entre *clusters*, ou seja, a pessoa percorre todo o percurso em regiões treinadas previamente, partindo da região central e dobrando à esquerda.

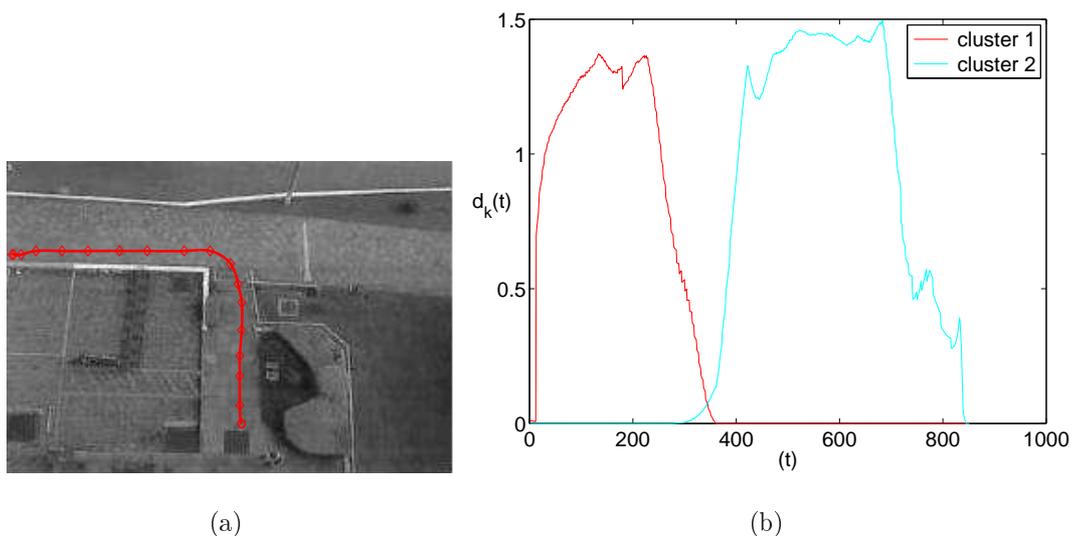


Figura 4.4 – (a) Transição entre *clusters*. (b) Gráfico $d_k(t)$.

O gráfico da Figura 4.4(b) indica que a primeira parcela da trajetória é coerente apenas com *cluster 1*, e a segunda parcela coerente apenas com o *cluster 2*. Assim,

esta trajetória como um todo não é considerada usual. Salienta-se que esta é uma melhoria ao algoritmo de Soldera [11], no qual a mesma trajetória ilustrada na Figura 4.4 seria considerada completamente usual (pois não há o conceito de *clusters*).

Nos exemplos ilustrados nas Figuras 4.5, 4.6, 4.7, o período de treinamento foi alimentado com 191 trajetórias (inseridas manualmente), sendo geradas 6 classes. As variações são: trajetórias no sentido leste-oeste, leste-sul, oeste-sul e vice-versa. A trajetória de teste está em preto na Figura 4.5(a). A Figura 4.5(b) mostra o gráfico em que podem serem vistos os *clusters*, em que seu perfil de contorno foi coerente e em que amplitude. Já na Figura 4.5(c) obtemos os quadros com cada *cluster* de trajetórias discretizados e a coerência da trajetória de teste com estes. Nessas ilustrações, partes usuais da trajetória são marcadas em branco, e não usuais em preto. Também, podemos analisar visualmente as partes do contorno da trajetória de teste que foram compatíveis e, portanto, verificar se houve compatibilidade parcial ou total a um ou mais *clusters*. Aqui, o observador humano (agente de segurança) pode visualizar a coerência da trajetória e atribuir, conforme o contexto próprio, se a trajetória desta pessoa ou objeto é usual ou não usual. A Figura 4.6 mostra uma variação no contorno da trajetória de teste e nos quadros segue o mesmo tipo de análise visual.

A Figura 4.7 mostra novas simulações, usando novas trajetórias de teste (relativamente complexas), avaliadas com base no mesmo treinamento. Na Figura 4.7(a), o objeto ou pessoa se desloca com velocidade usual da direita para a esquerda até o meio do percurso e depois acelera. Note, que o módulo da velocidade fica fora da média obtida no treinamento e se encaixa no limiar de velocidade alta v_a . A figura apresenta parte em branco (usual) e o restante em preto (não usual). As demais Figuras 4.7(b) e 4.7(c) exemplificam outros casos.

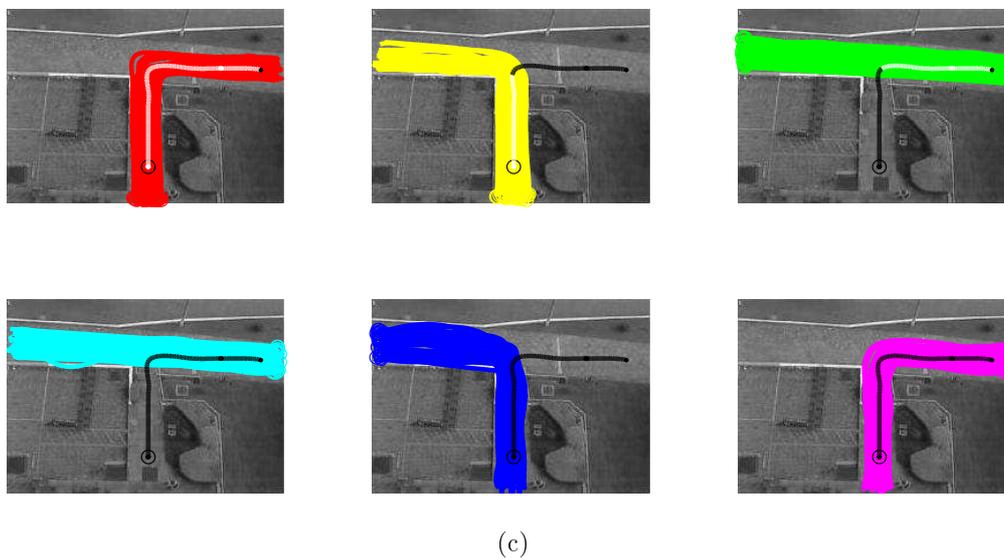
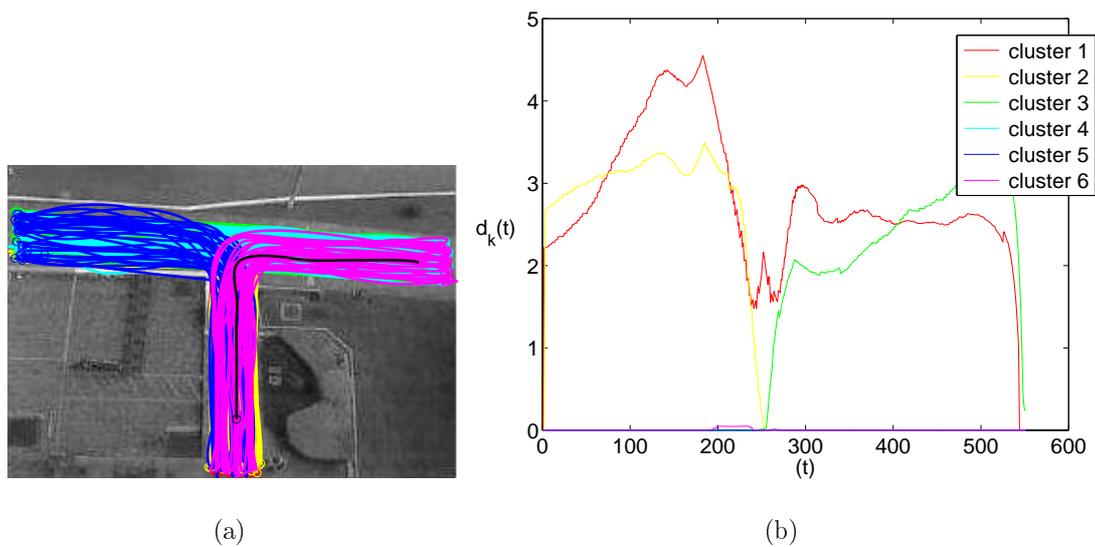


Figura 4.5 – (a) Trajetória de teste dobrando a direita sobreposta no treinamento. (b) Gráfico $d_k(t)$. (c) Em branco estão as porções da trajetória coerentes com o respectivo *cluster*.

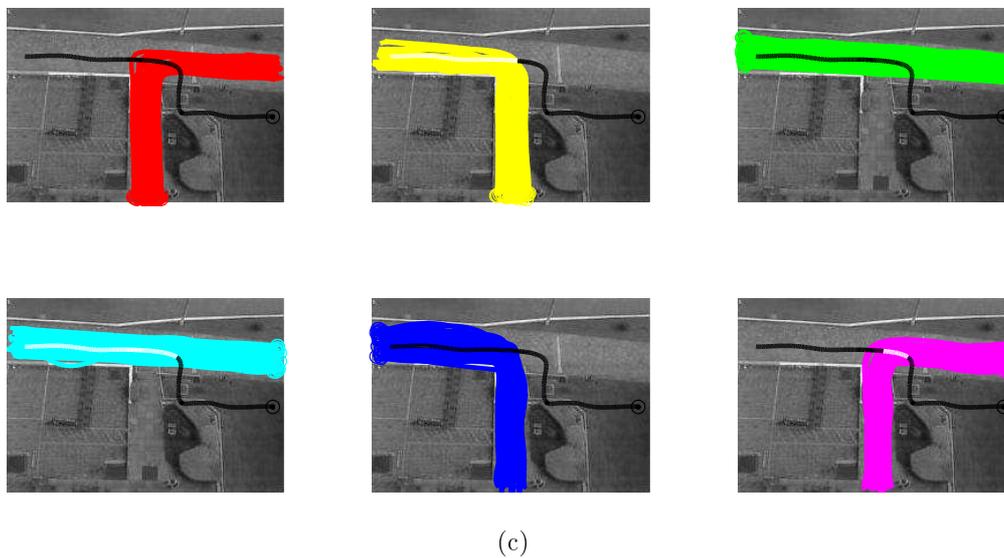
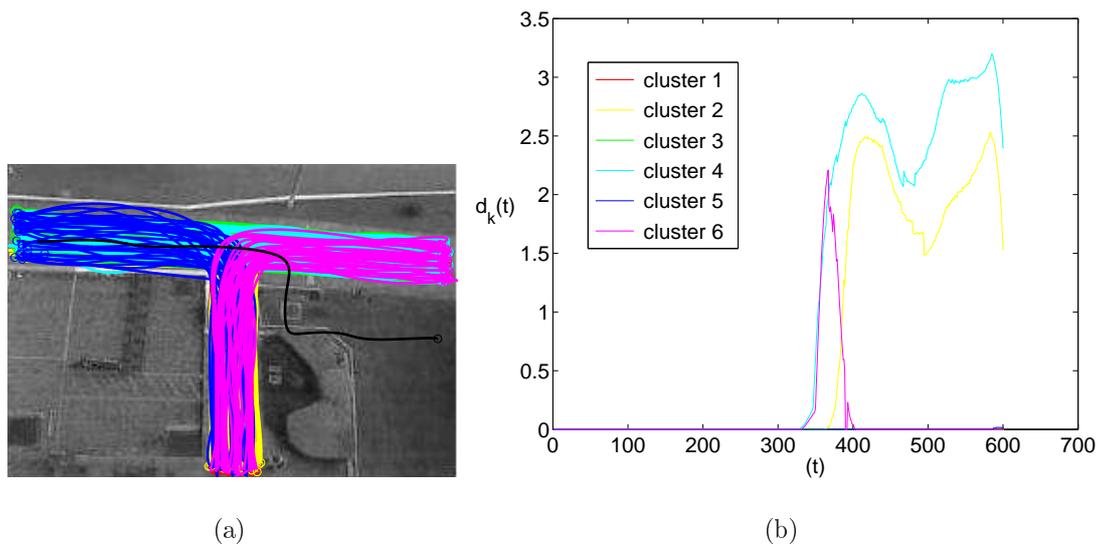


Figura 4.6 – (a) Trajetória de teste complexa sobreposta no treinamento. (b) Gráfico $d_k(t)$. (c) *Clusters* em que houve coerência.

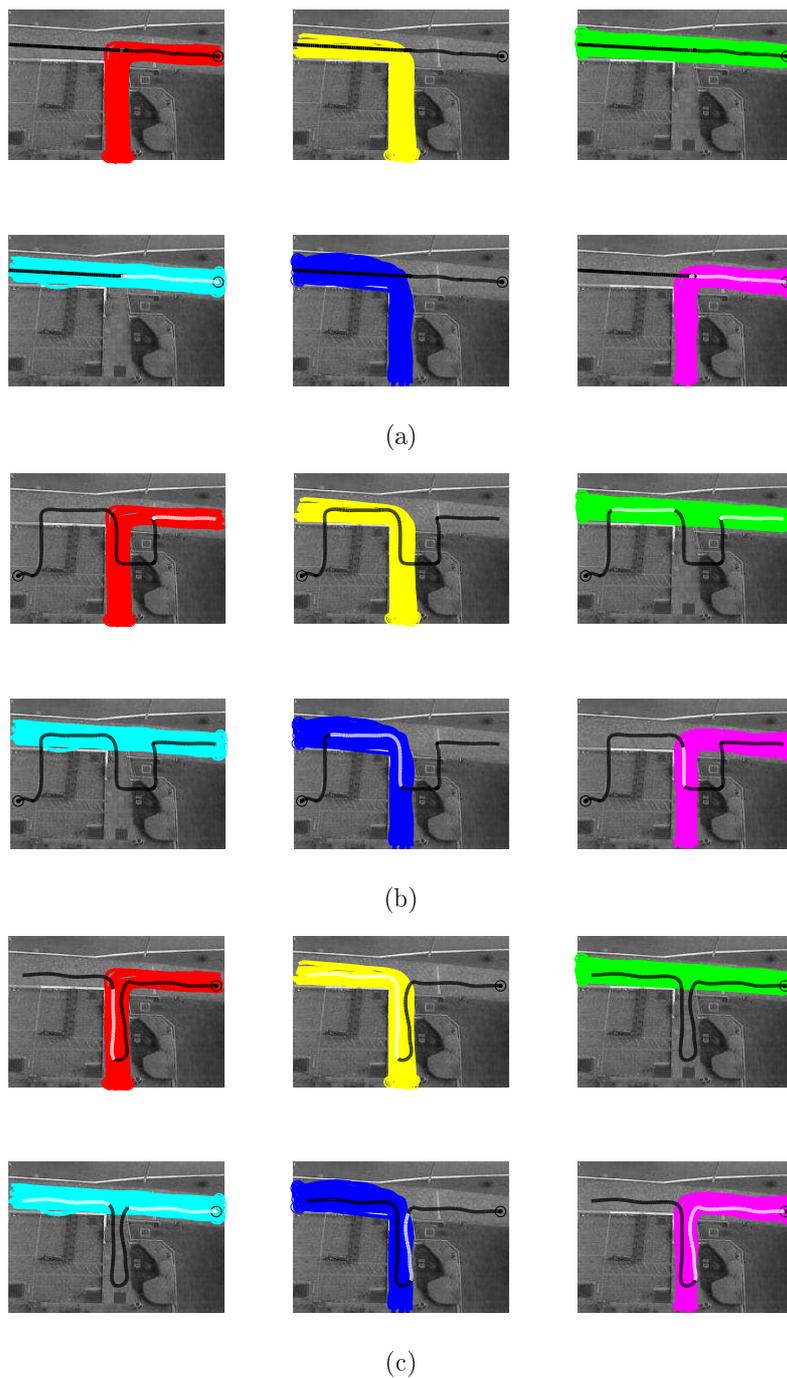


Figura 4.7 – (a) Trajetória de teste com variação de velocidade. (b) Trajetória que alterna com o canteiro. (c) Trajetória com desvio de direção.

Na simulação que apresentaremos a seguir, o treinamento foi feito utilizando-se um algoritmo de acompanhamento de objetos (*tracking*) com cenas reais de vídeo com veículos em um viaduto, cedidas por Hu et al. [9]. Foram fornecidas 1248 trajetórias de diversos padrões de contorno, e aqui foram aplicados os recursos de *clustering* e também remoção de *outliers*. Vale lembrar que possuímos uma área

de monitoramento com circulação comum de vários objetos formando sobreposições (camadas) com inúmeras trajetórias. O processo de avaliação no período de teste ou operação é visual e segue os mesmos moldes dos exemplos anteriores, onde o observador humano (agente de segurança) determina a aceitação do comportamento usual ou não usual, baseado nos critérios estabelecidos no histograma 4D. O valor do desvio padrão utilizado para veículos em todas as simulações foi aumentado para $\sigma = 25$.

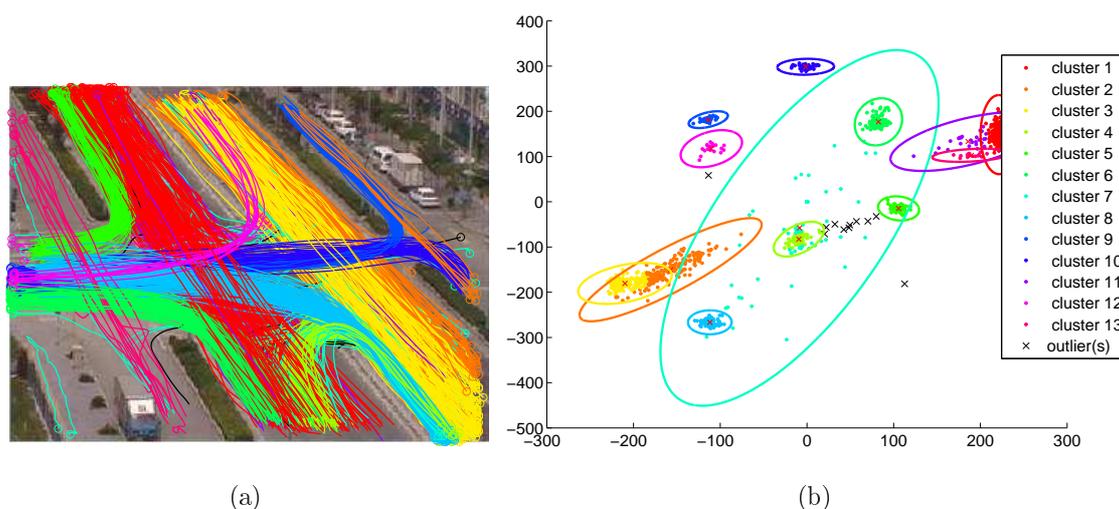


Figura 4.8 – (a) Treinamento conforme o *tracking* de cenas de vídeo reais. (b) *Clusters* e *outliers* obtidos.

Deste banco de dados foram extraídas 90% das trajetórias aleatoriamente como base de dados para treinamento, conforme Figura 4.8. Os *outliers* removidos estão representados por um “x” conforme a Figura 4.8(b). Os 10% restantes, ou seja, 126 trajetórias foram usadas como teste. A aleatoriedade foi utilizada de maneira a não tornar a aquisição das mesmas tendenciosas a um ou outro *cluster* ou perfil de contorno. As Figuras 4.9-4.12 ilustram a coerência de algumas destas trajetórias de teste com os histogramas obtidos no período de treinamento. Nesta simulação com dados reais foi possível detectar alguns problemas e verificar o quanto os algoritmos de *clustering* e remoção de *outliers* podem ser melhorados. Na Figura 4.8(b) podemos observar que existe uma imperfeição no processamento de um dos *clusters*, que acabou incorporando outros no seu interior (círculo maior). O reflexo desta imperfeição pode ser percebido diretamente no *cluster* apresentado no quadro 7 (da esquerda para a direita, em verde claro) mostrado nas Figuras de teste 4.9(c), 4.10(c),

4.11(c) e 4.12(c). Isto ocorre pela característica do algoritmo de agrupamento (*clustering*) utilizado, que tenta gerar classes muito grandes para acomodar amostras espalhadas no espaço de feições. Conforme comentamos no Capítulo 3 e também neste, os problemas gerados durante o emprego destas técnicas (*clustering* e remoção de *outliers*) ainda não estão devidamente implementados e solucionados.

No conjunto de Figuras 4.12 podemos observar uma trajetória de um veículo que foi identificado como não usual. Podemos observar que esta não é coerente integralmente com o histograma $4D$ em nenhum dos *clusters* mostrados na Figura 4.12(c). Como dito anteriormente, esta é uma trajetória típica para alarme e notificação do usuário, ficando a seu critério a classificação como usual ou não usual.

Em relação ao custo computacional, o algoritmo e as simulações foram implementadas e processadas usando a ferramenta *MATLAB*, versão 7.2.0.232(*R2006A*), em plataforma *Windows XP* em um *notebook* com processador AMD Turion 2.0 GHz utilizando memória de 1,5 GB RAM. O tempo de processamento, durante um treinamento, varia em função do número e a característica de contorno (grau de complexidade) das trajetórias. Nas simulações mais complexas e numerosas, testadas neste trabalho, onde utilizamos o bloco de trajetórias cedido pelo Dr. Weiming Hu, processamos 1248 trajetórias obtendo-se os resultados com um tempo de 45 segundos. O período de teste (comparação com os histogramas de cada *cluster*) ocorre em tempo-real.

Este trabalho possui uma versão compacta, com o título “*Detection of Unusual Motion Using Trajectory Clustering and 4D Histograms*”, que foi submetido ao periódico *Pattern Recognition*.

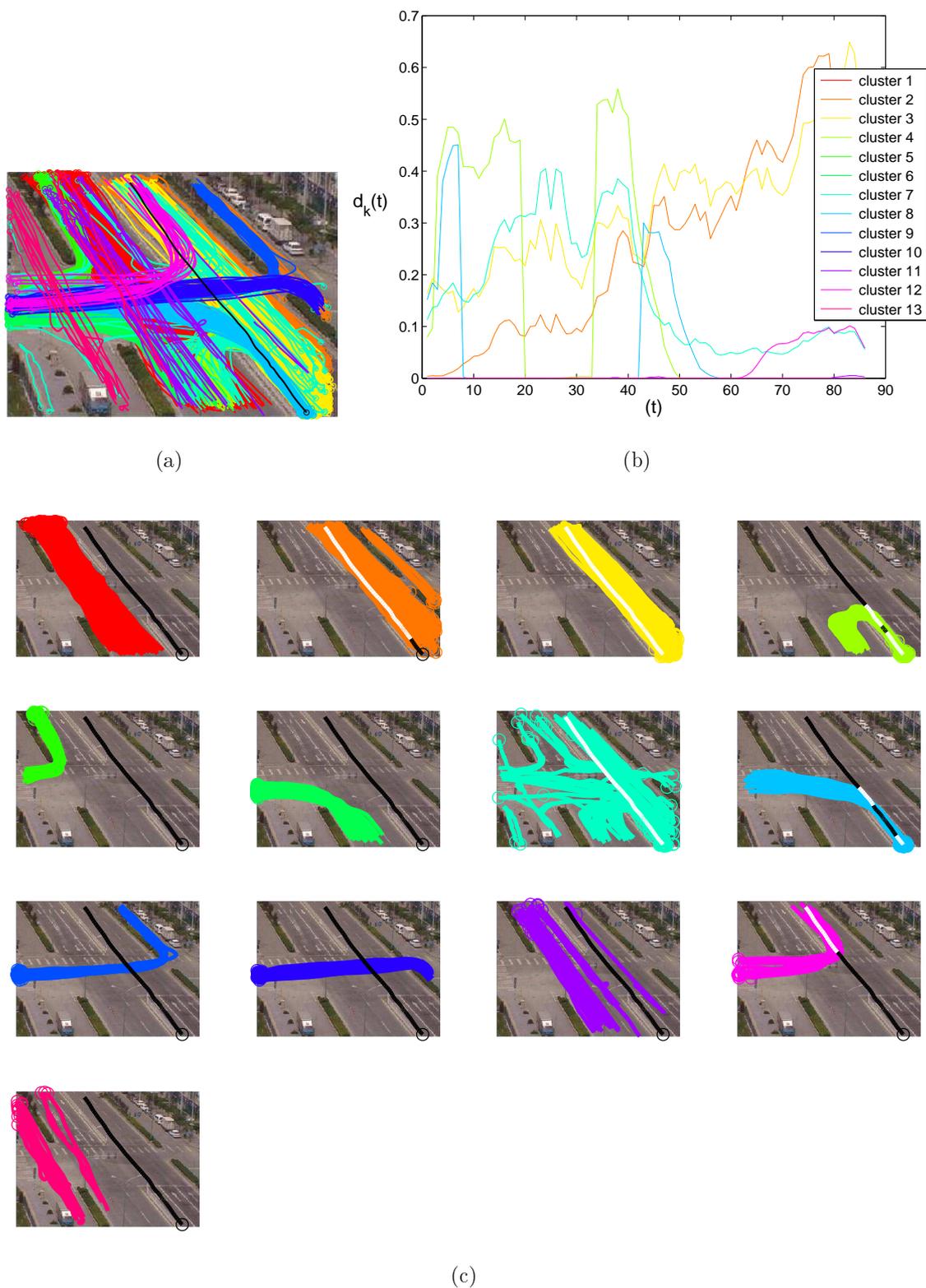
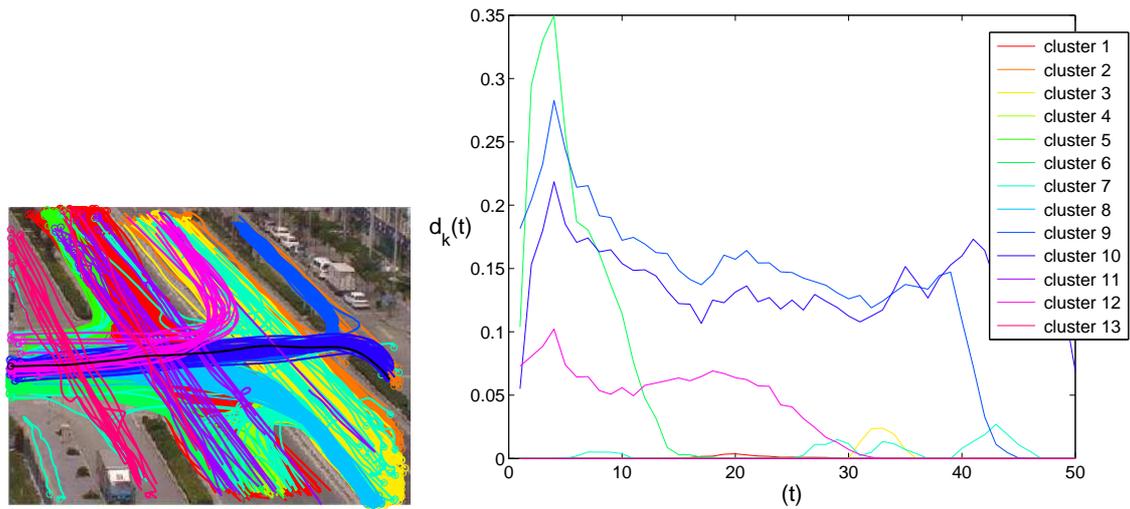
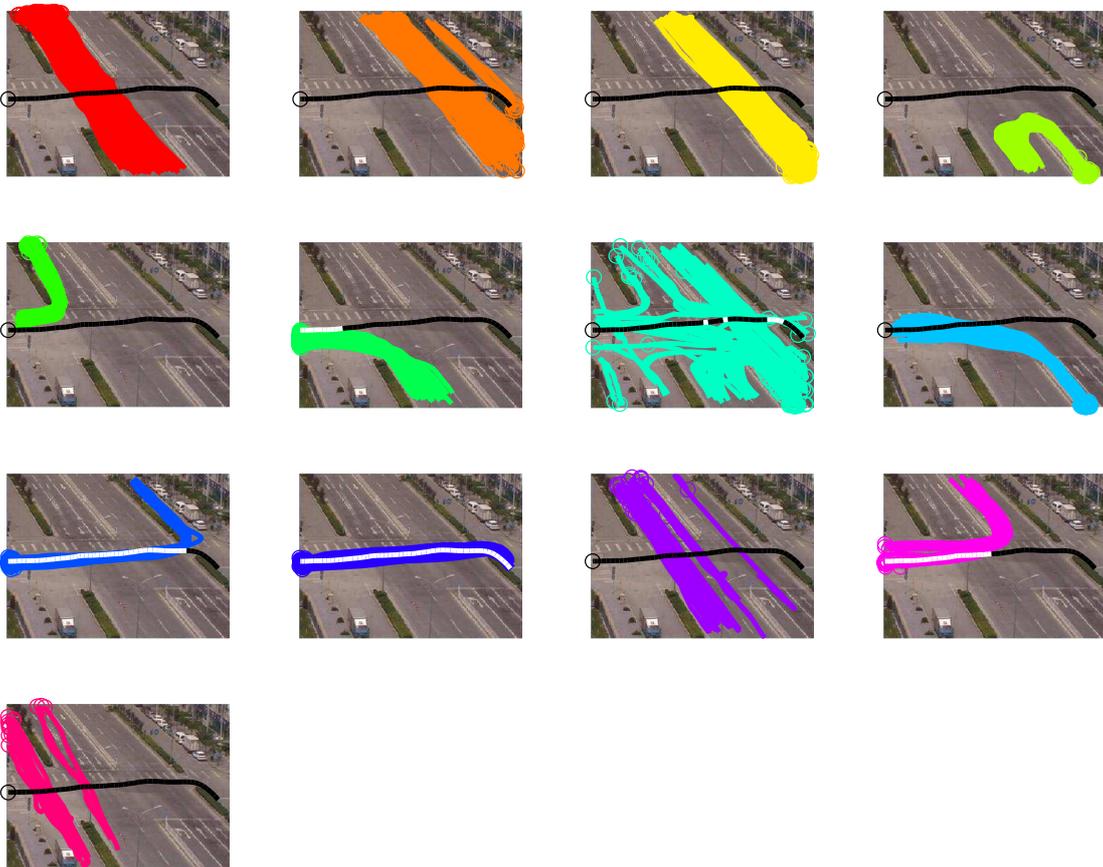


Figura 4.9 – (a) Trajetória no sentido sul-norte de teste empregada. (b) Gráfico $d_k(t)$. (c) Em branco estão as porções da trajetória coerentes com o respectivo *cluster*.



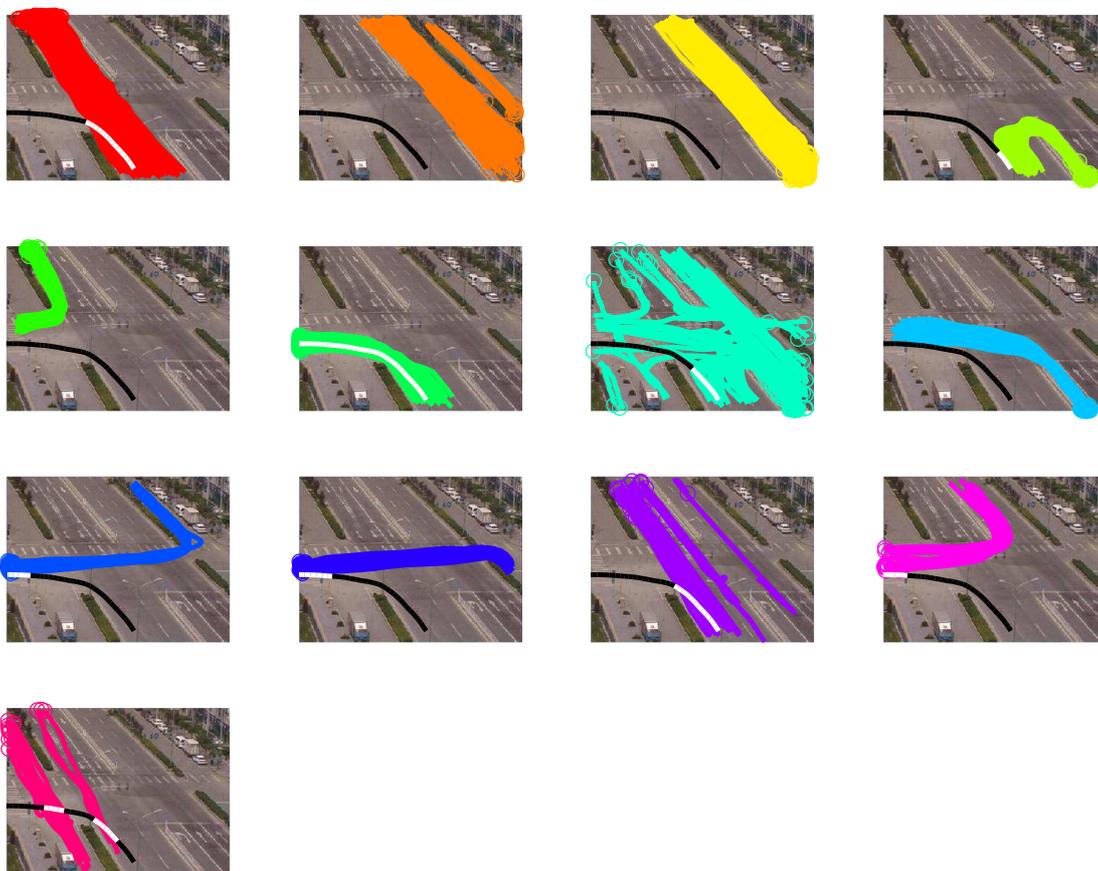
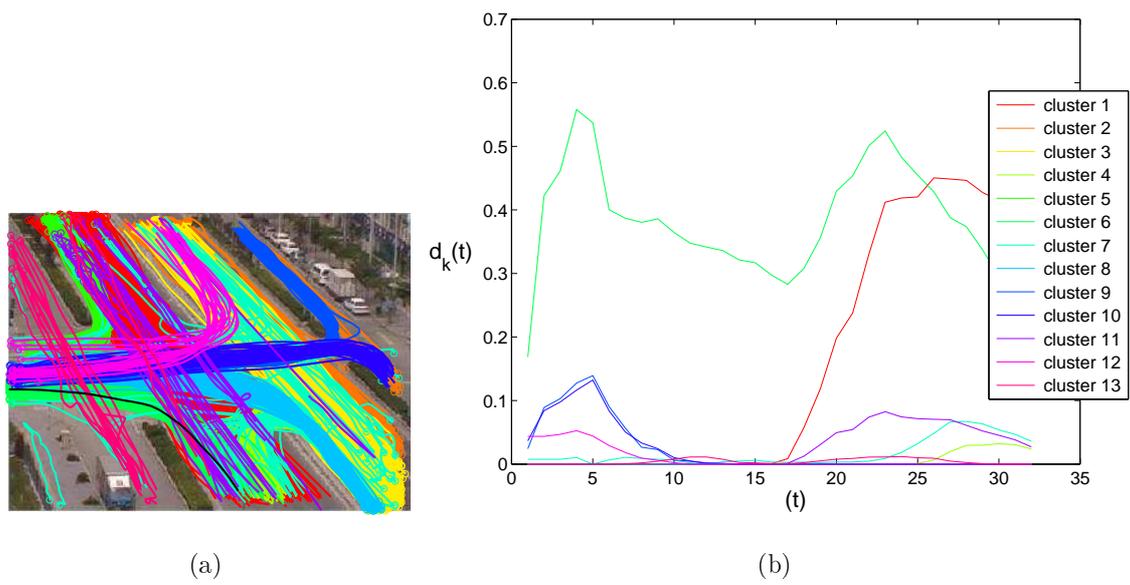
(a)

(b)



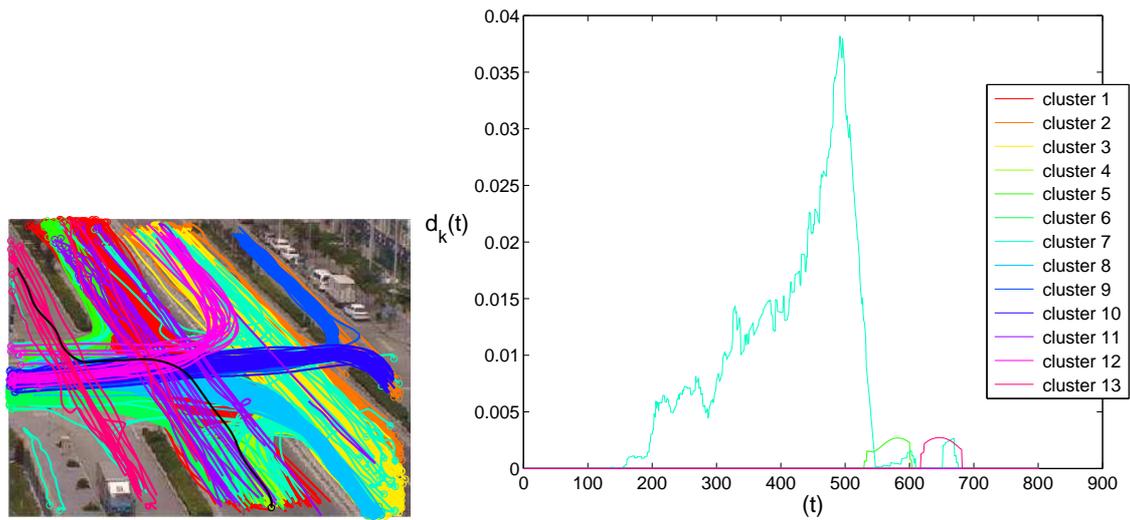
(c)

Figura 4.10 – (a) Trajetória no sentido oeste-leste de teste empregada. (b) Gráfico $d_k(t)$. (c) Em branco estão as porções da trajetória coerentes com o respectivo *cluster*.



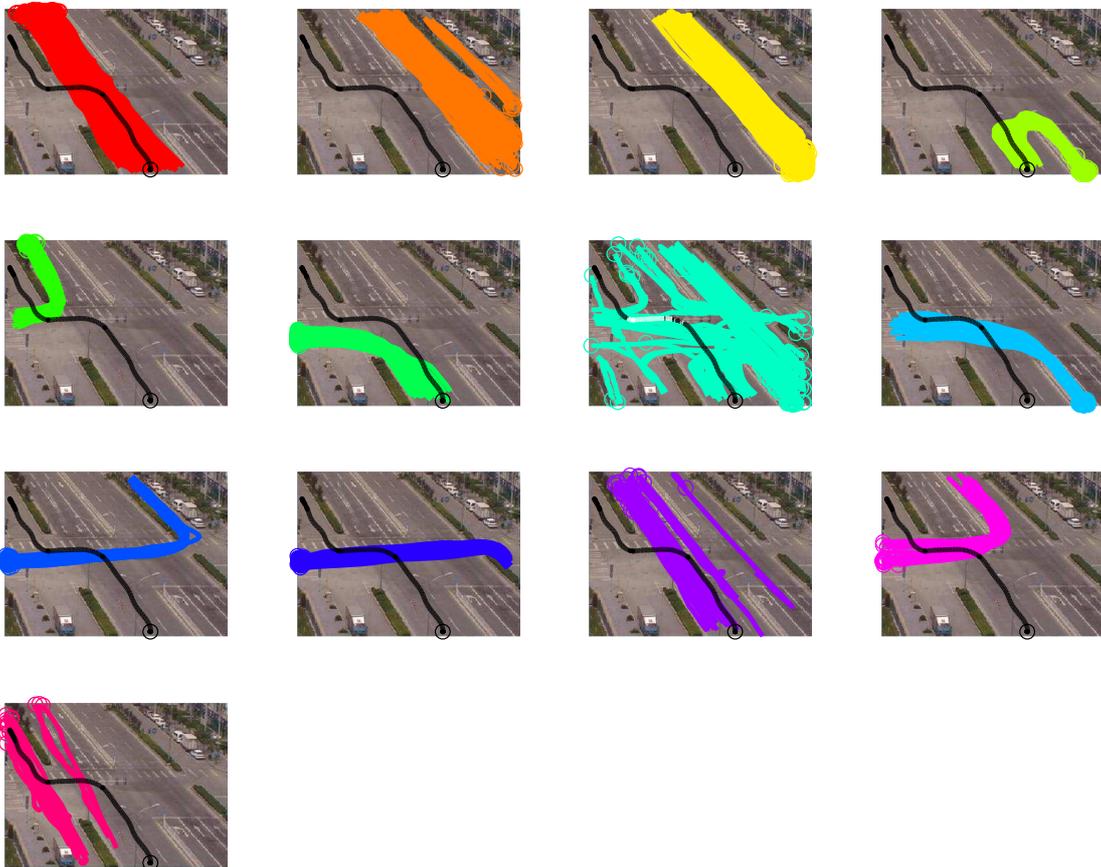
(c)

Figura 4.11 – (a) Trajetória curva de teste. (b) Gráfico $d_k(t)$. (c) Em branco estão as porções da trajetória coerentes com o respectivo *cluster*.



(a)

(b)



(c)

Figura 4.12 – (a) Trajetória não usual. (b) Gráfico $d_k(t)$. (c) Não houve coerência integral desta trajetória em nenhum dos *Clusters*.

Capítulo 5

Considerações Finais

Neste Capítulo serão revisados os conceitos que motivaram o desenvolvimento deste trabalho, as contribuições que resultaram do desenvolvimento do projeto e trabalhos futuros que foram vislumbrados no decorrer desta pesquisa.

5.1 Conclusões

O trabalho proposto está fundamentado na idéia de classificação de trajetórias (idealmente obtidas automaticamente através de algoritmos de visão computacional) como usuais ou não usuais. Toda a base de dados que o algoritmo utiliza para fazer esta classificação é proveniente de um período de treinamento obrigatório, onde o movimento das pessoas ou objetos é aprendido e utilizado como padrão de normalidade. Quanto mais trajetórias forem adquiridas nesta etapa, mais detalhadas serão estas informações no seu banco de dados e menores as chances de se obterem alarmes falsos, nas futuras comparações com trajetórias no período de teste ou operação. Também, é importante salientar que o algoritmo deve ser aplicado em condições similares àquelas encontradas no período de treinamento, evitando-se alarmes incoerentes.

Durante o período de treinamento, as trajetórias com padrões de contorno semelhantes passam por um processo de agrupamento em classes de trajetórias similares. O método se resume em extrair, para cada trajetória, um vetor de feições que a caracteriza. Assim sendo, trajetórias semelhantes gerariam vetores similares. Dessa forma, cada conjunto de vetores de feições associados a trajetórias coerentes formariam um aglomerado de pontos, que podem ser detectados utilizando

o algoritmo de *clustering*, mencionado no Capítulo 3. Assim, trajetórias com mesma característica de contorno produzirão vetores com características similares, gerando um *cluster* 2N-dimensional, que é modelado por uma distribuição de probabilidade Gaussiana.

Também, incluímos um processo de remoção de *outliers*. Este processo, torna-se interessante na medida que excluimos trajetórias que possuem características significativamente diferentes dos *clusters* detectados durante o treinamento.

Obtendo-se os grupos de trajetórias, um histograma 4D é computado para cada *cluster* detectado no treinamento. Neste histograma estão definidos alguns critérios considerados para este trabalho, como importantes para a supervisão de um determinado ambiente. Primeiramente, foi estabelecido o preenchimento de um mapa de ocupação espacial bidimensional (SOM). Porções não usuais da trajetória são detectadas se a pessoa ou objeto trafegar em uma região pouco ocupada no espaço, ou seja, a cada instante são extraídas as coordenadas espaciais (x, y) da pessoa ou objeto no plano. Também, são obtidos a direção e o módulo do vetor velocidade analisando-se o **espaço-fase**. A orientação do vetor velocidade foi quantizado em oito setores circulares. Já o módulo foi quantizado em três níveis (baixo, normal e alto), sendo que esses limiares são obtidos automaticamente em função do deslocamento médio das pessoas na cena. Assim, é obtido um histograma 4D do histórico de movimentação obtido no treinamento, que será utilizado para validar cada trajetória no período de teste ou operação. No último estágio, o período de teste ou operação, cada nova trajetória obtida será comparada com o histograma de cada *cluster* obtido no treinamento. Conforme os critérios estabelecidos pelo histograma, esta trajetória (ou parte desta) será considerada usual ou não usual, sendo que nesta última condição o operador humano (agente de segurança) deve ser notificado por um alarme.

Vale lembrar que o observador humano (agente de segurança), com base no alarme gerado, deve avaliar e decidir o que será efetivamente considerado usual ou não usual. A definição de comportamento não usual é subjetiva e dependente de contexto, e todos os resultados apresentados neste trabalho se baseiam na noção de “normalidade” apresentada ao longo do texto.

Os resultados experimentais mostradas ao longo do trabalho indicam que o algoritmo desenvolvido pode ser efetivamente utilizado para alertar agentes de

segurança sobre a presença de movimentos não usuais. Deve-se salientar que muitos dos testes foram realizados com trajetórias inseridas manualmente, e que um estudo mais aprofundado com trajetórias reais (obtidos por algoritmos de acompanhamento de objetos) deve ser realizado.

5.2 Trabalhos Futuros

Embora o algoritmo proposto tenha apresentados resultados promissores, há alguns aspectos no modelo que podem ser aprimorados. O uso de trajetórias reais (obtidas com algoritmos de acompanhamento de objetos), que contêm imprecisões e falhas nas trajetórias, gerou alguns erros no algoritmo de *clustering* que utilizamos. Assim, um estudo mais aprofundado sobre o problema de *clustering* de trajetórias é um ponto que pode ser melhorado.

Outro ponto que pode ser aperfeiçoado se refere ao período de treinamento deste modelo. Há diversos ambientes onde os fluxos de pessoas se alteram dinamicamente, gerando uma grande quantidade de falsos alarmes. Por exemplo, imaginemos um treinamento feito na entrada de uma escola (fluxo predominante em direção à escola). Na saída da escola, a maioria das trajetórias sendo analisadas estarão em direção oposta àquela treinada, embora a situação seja usual. Assim, uma melhoria significativa no modelo seria a utilização de diferentes conjuntos de treinamento para diferentes horas do dia, ou de uma estratégia para treinar novamente o modelo adaptativamente.

Todas as técnicas empregadas neste trabalho, inclusive no que se refere ao histograma $4D$, armazenam uma quantidade muito grande de informações ainda não utilizadas, estando guardadas, no seu banco de dados e que podem certamente serem exploradas. Um reaproveitamento otimizado destas informações ainda ocultas, poderia fornecer ao usuário uma ferramenta poderosa de alto nível. Estas informações poderiam, por exemplo, serem usadas para uma análise individual de uma ou várias trajetórias simultaneamente comparando suas características de contorno em um determinado momento. Em um outro exemplo, seria possível reutilizar algumas informações para melhorar o *tracking* utilizado para treinamento, ou seja, obtendo-se uma ou mais trajetórias incompletas sendo possível prever suas características de contorno, as mesmas poderiam ser simuladas e completadas.

Finalmente, uma validação mais profunda do desempenho do algoritmo desenvolvido junto a profissionais da área de segurança é necessária, para avaliar os pontos fortes e as questões em que o método pode ser efetivamente melhorado. Além disso, uma consulta a esses profissionais permitiria um levantamento de novos eventos possivelmente suspeitos que poderiam ser incorporados ao modelo.

Bibliografía

- [1] VALERA, M.; VELASTIN, S. Intelligent distributed surveillance systems: a review. *IEEE Proceedings, Vision Image and Signal Processing, (VISP)*, v. 152, p. 192–204, April 2005.
- [2] NAIR, V.; CLARK, J. Automated visual surveillance using hidden markov models. *International Conference on Vision Interface*, Calgary, Canadá, p. 88–93, May 2002.
- [3] MAHAJAN, D. et al. A framework for activity recognition and detection of unusual activities. *Indian Conference on Computer Vision, Graphics and Image Processing, (ICVGIP)*, p. 15–21, December 2004.
- [4] MAKRIS, D.; ELLIS, T. Learning semantic scene models from observing activity in visual surveillance. *IEEE Transactions on Systems, Man, and Cybernetics, Part B 35*, v. 3, p. 397–408, June 2005.
- [5] JUNEJO, I. N.; JAVED, O.; M.SHAH. Multi feature path modeling for video surveillance. *International Conference on Pattern Recognition, (ICPR 04)*, v. 2, p. 716–719, August 2004.
- [6] LOU, J. et al. Semantic interpretation of object activities in a surveillance system. *International Conference on Pattern Recognition, (ICPR'02)*, v. 3, p. 777–780, August 2002.
- [7] JOHNSON, N.; HOGG, D. Learning the distribution of object trajectories for event recognition. *Image and Vision Computing - Elsevier*, v. 14, n. 8, p. 609–615, August 1996.

- [8] WEIMING, H. et al. Semantic-based surveillance video retrieval. *IEEE Transactions on Image Processing*, v. 16, n. 4, p. 1168–1181, April 2007.
- [9] HU, W. et al. A system for learning statistical motion patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE Computer Society, Washington, DC, USA, v. 28, n. 9, p. 1450–1464, September 2006. ISSN 0162-8828.
- [10] ZHOU, H.; KIMBER, D. Unusual event detection via multi-camera video mining. In: *Proceedings of the 18th International Conference on Pattern Recognition, (ICPR'06)*. Washington, DC, USA: IEEE Computer Society, 2006. p. 1161–1166. ISBN 0-7695-2521-0.
- [11] SOLDERA, J. *Detecção de Comportamentos Estranhos em Seqüências de Vídeo*. Dissertação (Dissertação de mestrado) — Universidade do Vale do Rio dos Sinos - UNISINOS, São Leopoldo, RS, 2007.
- [12] HOSIE, R.; VENKATESH, S.; WEST, G. Classifying and detecting group behaviour from visual surveillance data. In: *Proceedings of the 14th International Conference on Pattern Recognition, (ICPR'98)*. Washington, DC, USA: IEEE Computer Society, 1998. v. 1, p. 602–604. ISBN 0-8186-8512-3.
- [13] GAVRILA, D. M. The analysis of human motion and its application for visual surveillance. In: *Proceedings of the Second IEEE Workshop on Visual Surveillance, (VS'99)*. Washington, DC, USA: IEEE Computer Society, 1999. p. 3–5. ISBN 0-7695-0037-4.
- [14] NIU, W. et al. Human activity detection and recognition for video surveillance. In *Proceedings of the IEEE International Conference on Multimedia and Expo, (ICME'04)*, Taipei, Taiwan, v. 1, p. 719–722, June 2004.
- [15] CUPILLARD, F. et al. Video understanding for metro surveillance. *Spatial Session on Intelligent Transportation Systems - IEEE International Conference On Networking, Sensing and Control, (ICNSC 2004)*, Taiwan, v. 1, p. 186–191, March 2004.

- [16] FUNG, C. C.; JERRAT, N. A neural network based intelligent intruders detection and tracking system using cctv images. *Proceedings of the IEEE Region 10 Conference on Intelligent Systems and Technologies for teh next Millenium (Tencon'2000)*, Kaula Lumpur, Malaysia, v. 2, p. 409–414, September 2000.
- [17] HUNG, H.; GONG, S. Detecting and quantifying unusual interactions by correlating salient motion. In: *IEEE International Conference on Advanced Video and Signal based Surveillance*. [S.l.: s.n.], 2005. p. 46–51.
- [18] JUNIOR, J. C. S. J. *Utilizando Visão Computacional para Simular e Validar Comportamentos de Multidões de Humanos Virtuais*. Dissertação (Dissertação de mestrado) — Universidade do Vale do Rio dos Sinos - UNISINOS, São Leopoldo, RS, 2006.
- [19] JUNG, C. R. et al. Detection of unusual motion using computer vision. *XIX Brazilian Symposium on Computer Graphics and Image Processing, (SIBGRAPI'06)*, p. 349–356, October 2006.
- [20] FERRIER, N. J.; ROWE, S. M.; BLAKE, A. Real-time traffic monitoring. In: *Workshop on Applications of Computer Vision, (WACV94)*. [S.l.: s.n.], 1994. p. 81–88.
- [21] PEURSUM, P.; VENKATESH, S.; WEST, G. Observation-switching linear dynamic systems for tracking humans through unexpected partial occlusions by scene objects. In: *Proceedings of the 18th International Conference on Pattern Recognition, (ICPR'06)*. [S.l.: s.n.], 2006. p. 929–934. ISBN 0-7695-2521-0.
- [22] ADAM, A.; RIVLIN, E.; SHIMSHONI, I. Robust fragments-based tracking using the integral histogram. In: *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (CVPR'06)*. [S.l.: s.n.], 2006. p. 798–805. ISBN 0-7695-2597-0.
- [23] ASHARAF, S.; SHEVADE, S.; MURTY, M. N. Rapid and brief communication rough support vector clustering. *Computer Science and Automation, Indian Institute of Science - Elsevier*, Bangalore 560012, India, p. 1779–1783, 2004.

- [24] SCHÖLKOPF, B.; SMOLA, A.; MÜLLER, K.-R. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, MIT Press, Cambridge, MA, USA, v. 10, n. 5, p. 1299–1319, 1998. ISSN 0899-7667.
- [25] FIGUEIREDO, M. A. T.; JAIN, A. K. Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 24, n. 3, p. 381–396, March 2002.
- [26] KUMAR, M.; ORLIN, J. B. *Scale-invariant Clustering with Minimum Volume Ellipsoids*. Rutgers Business School RUTCOR and Sloan School of Management and Operations Research Center, New Jersey EUA, September 2005.
- [27] LAW, M. H. C.; FIGUEIREDO, M. A. T.; JAIN, A. K. Simultaneous feature selection and clustering using mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE Computer Society, Washington, DC, USA, v. 26, n. 9, p. 1154–1166, 2004.
- [28] FÖRSTNER, W.; MOONEN, B. *A Metric for Covariance Matrices*. Stuttgart University, 1999. Disponível em: <wf@ipb.uni-bonn.de>.
- [29] ZIVKOVIC, Z.; HEIJDEN, F. van der. Recursive unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 26, n. 5, p. 651–656, May 2004.
- [30] GRAHAM, M. W.; MILLER, D. J. Unsupervised learning of parsimonious mixtures on large spaces with integrated feature and component selection. *IEEE Transactions on Signal Processing*, v. 54, n. 4, April 2006.
- [31] PEÑA, D.; PRIETO, F. J. Multivariate outlier detection and robust covariance matrix estimation. *American Statistical Association and the American Society for Quality Technometrics*, Universidad Carlos III, Madrid, Spain, v. 43, n. 3, p. 286–310, August 2001.