

André Gustavo Hochuli

**Detecção de Eventos Não Convencionais
em Cenas de Vídeo Utilizando Vetores de
Características**

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Mestre em Informática.

Curitiba
2007

André Gustavo Hochuli

Detecção de Eventos Não
Convencionais em Cenas de Vídeo
Utilizando Vetores de
Características

Dissertação apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná como requisito parcial para obtenção do título de Mestre em Informática.

Área de Concentração: Ciência da Computação

Orientador: Alessandro Lameiras Koerich
Co-orientador: Alceu de Souza Britto Jr

Curitiba
2007

Hochuli, André Gustavo

Detecção de Eventos Não Convencionais em Cenas de Vídeo Utilizando Vetores de Características. Curitiba, 2007. 50p.

Dissertação - Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Informática.

1. Rastreamento 2. Classificação de Eventos 3. Detecção de Eventos não-convencionais I. Pontifícia Universidade Católica do Paraná. Centro de Ciências Exatas e Tecnologia. Programa de Pós-Graduação em Informática II - t

A minha família, e a todos os colegas que
contribuíram para o sucesso deste trabalho.

Agradecimentos

Agradeço primeiramente a meus pais por terem me apoiado de forma incondicional, em todos os sentidos, para que eu realiza-se este trabalho. À meus irmãos que de alguma forma também tiveram sua parcela de apoio nessa jornada. À meus colegas de trabalho e laboratório pela ajuda das mais variadas possíveis que forneceram, desde dúvidas solucionadas até participação em construção de bases de vídeos.

Agradeço também aos meus familiares, em especial a minha madrinha *Prof^a. Dra.* Celina Imaculada Girardi e ao meu padrinho Fernando Cesar Girardi, por todo o apoio fornecido de forma direta ou indireta.

Agradeço aos meus orientadores *Prof. Dr.* Alessandro L. Koerich e *Prof. Dr.* Alceu de Souza Britto Jr., pelo profissionalismo, dedicação e amizade durante esses anos. Ao *Prof. Dr.* Luiz Eduardo S. de Oliveira, pelos auxílios diversos.

Agradeço também a todos que de forma direta ou indireta contribuíram para o resultado final deste trabalho.

E por fim, mas não menos importante, agradeço a Deus por ter me dado força, inteligência e capacidade de realizar este trabalho.

“O homem, como qualquer outro animal, é por natureza indolente. Se nada o estimula, mal se dedica a pensar e se comporta guiado apenas pelo hábito, como um autômato.”

Albert Einstein

Sumário

Agradecimentos	ii
Sumário	iii
Lista de Figuras	v
Lista de Tabelas	vii
Lista de Símbolos	viii
Lista de Abreviações	x
Resumo	xi
Abstract	xii
Capítulo 1	
Introdução	1
1.1 Objetivos	2
1.2 Desafios	4
1.3 Contribuições	5
1.4 Organização do Documento	6
Capítulo 2	
Estado da Arte	7
2.1 Detecção e Rastreamento do Movimento	7
2.2 Análise de Comportamento e Classificação de Eventos	11
2.3 Discussão	16
Capítulo 3	
Metodologia	18
3.1 Detecção de Movimento	19
3.2 Rastreamento	23
3.3 Extração de Características	26

3.4	Classificação do Movimento	29
3.5	Normalização dos vetores	32
3.6	Conclusão	33
Capítulo 4		
Resultados Experimentais		34
4.1	Base de Vídeos	35
4.1.1	Base Estacionamento	35
4.1.2	Base Caviar	36
4.2	Ajuste de Parâmetros	36
4.3	Resultados e Análise	37
4.3.1	Base Estacionamento	37
4.3.2	Base Caviar	38
4.3.3	Análise de Erros	40
4.4	Conclusão	41
Capítulo 5		
Conclusão		42
5.1	Contribuições	43
5.2	Trabalhos Futuros	44
Referências Bibliográficas		45
Apêndice A		
Taxas de Acerto		48

Lista de Figuras

Figura 1.1	Sistemas de monitoramento	2
Figura 1.2	Elementos do método proposto	4
Figura 2.1	Segmentação do movimento	8
Figura 2.2	Representação dos objetos	11
Figura 3.1	Visão geral do sistema.	19
Figura 3.2	Comparativo entre a técnica utilizada e a subtração para um fundo fixo. (a) Quadro atual, (b) Quadro de fundo. (c) Imagem resultante da subtração do quadro atual com o quadro de fundo. (d) Imagem resultante da aplicação do método proposto por Stauffer and Grimson (2000), (e) Imagem resultante da aplicação do algoritmo de realce	22
Figura 3.3	Exemplo de detecção e segmentação do movimento: (a) quadro ori- ginal do vídeo com objetos em movimento, (b) segmentação do movimento por gaussianas, (c) <i>blobs</i> resultantes após aplicação de filtros, subtração de fundo e detecção de contornos.	23
Figura 3.4	Objeto aceito: (a) objeto aceito com a <i>Bounding Box</i> . (b) Seg- mentação do movimento. (c) Contornos do objeto	24
Figura 3.5	Objeto rejeitado: (a) Objeto rejeitado sem <i>Bounding Box</i> . (b) Segmentação do movimento. (c) Contornos do objeto	24
Figura 3.6	A. Objetos rastreados no quadro anterior, B. Objetos desconhecidos no quadro atual	24
Figura 3.7	Posição do objeto na cena	26
Figura 3.8	Velocidade entre dois quadros consecutivos	27
Figura 3.9	Característica deslocamento	28
Figura 3.10	Característica variação da <i>Bounding Box</i>	28
Figura 3.11	Característica direção	29

Figura 3.12 Ilustração do processo de extração e geração dos vetores de características a partir de objetos de interesse em movimento no quadros do vídeo	29
Figura 3.13 Exemplo do processo de classificação dos vetores de características relativos ao objeto de interesse.	31
Figura 4.1 Sequência de quadros do movimento “Andar” da base Estacionamento	38
Figura 4.2 Sequência de quadros do movimento “Correr” da base Estacionamento	39
Figura 4.3 Sequência de quadros do movimento “ZigZag” da base Estacionamento	39
Figura 4.4 Sequência de quadros do movimento “Andar” da base Caviar	40
Figura 4.5 Sequência de quadros do movimento “Briga” da base Caviar	40

Lista de Tabelas

Tabela 4.1	Detalhes dos vídeos da base estacionamento	36
Tabela 4.2	Detalhes dos vídeos da base caviar	36
Tabela 4.3	Matriz de confusão para o conjunto de características <i>Vel</i> e <i>Dir</i> com 10 Vizinhos.	38
Tabela 4.4	Matriz de confusão para o conjunto de características <i>Vel</i> e <i>Pos</i> com 5 Vizinhos.	39
Tabela A.1	Taxas de Acerto para os conjuntos da Base Estacionamento. O valor “1” representa quais características estavam sendo utilizadas no conjunto .	49
Tabela A.2	Taxas de Acerto para os conjuntos da Base Caviar. O valor “1”re- presenta quais características estavam sendo utilizadas no conjunto	50

Lista de Símbolos

Alt	Altura de um objeto
Alt_{Min}	Limiar de altura mínima
Alt_{Max}	Limiar de altura máxima
$Bbox$	Característica tamanho (Bounding Box)
C	Movimento convencional
dP	Distância euclidiana entre os centros
dS	Diferença de tamanho
dD	Diferença da direção
dT	Diferença de persistência
$Desl$	Característica deslocamento
Dir	Característica direção
F^n	Quadro atual de uma sequência de quadros
J	Tamanho da janela temporal
$Larg$	Largura de um objeto
$Larg_{Min}$	Limiar de largura mínima
$Larg_{Max}$	Limiar de largura máxima
MAX	Valor máximo de uma característica
MIN	Valor mínimo de uma característica
NC	Movimento não-convencional
O^i	Objeto com índice i
O_c^i	Posição central do objeto i
O_r^i	Tamanho do objeto i
O_s^i	Posição estimada do objeto i
$O_{t l}^i$	Custo de persistência do objeto i
Pos	Característica posição
qps	Quadros por segundo

Q	Intervalo de quadros
R^j	Região de Movimento com índice j
TTL	Persistência de um objeto
TTL_{Max}	Persistência Máxima
V	Conjunto de vetores de característica
V_i	Vetor de característica i
VAL	Valor de uma característica
Vel	Característica velocidade
$VNorm$	Valor normalizado de uma característica
wP	Peso relativo a posição do objeto
wS	Peso relativo ao tamanho do objeto
wD	Peso relativo a distância do objeto
x	Posição relativa em x do plano cartesiano
y	Posição relativa em y do plano cartesiano

Lista de Abreviações

DFT	Transformação discreta de fourier (<i>Discrete fourier transform</i>)
FPS	Quadros por segundo (<i>Frames per second</i>)
QPS	Quadros por segundo
HMM	Modelos escondidos de markov (<i>Hidden markov model</i>)
HHMM	Modelos escondidos hierárquicos de markov (<i>Hierarchical hidden markov model</i>)
k NN	k Vizinhos mais próximos (<i>k Nearest neighbors</i>)
PCA	Análise de componente principal (<i>Principal component analysis</i>)
SVM	Máquina de vetores de suporte (<i>Support vector machine</i>)

Resumo

Este trabalho apresenta uma proposta para a detecção em tempo-real de eventos não convencionais em cenas de vídeo. O método consiste em analisar um vídeo proveniente de uma câmera de segurança para detectar, segmentar e rastrear objetos em movimento na cena e classificar o movimento destes objetos como sendo convencional ou não convencional. De cada objeto rastreado em uma cena, são extraídas características como deslocamento, coordenadas, velocidade, variação da direção e do tamanho da *bounding box* ao longo de sua trajetória. Estas características são agregadas em vetores de características. Na fase de treinamento, o sistema por meio de intervenção humana, rotula cada um desses vetores como sendo convencional ou não convencional e armazena os mesmos em uma base de dados. Durante a classificação os vetores gerados em tempo real são comparados com vetores previamente rotulados durante a etapa de treinamento e um algoritmo de aprendizagem baseado em instâncias é utilizado para classificar o movimento como sendo convencional ou não convencional. Resultados experimentais sobre vídeos reais de duas bases, CAVIAR e Estacionamento, mostram que a solução proposta é capaz de identificar em tempo real movimentos não convencionais com uma precisão entre 78% e 82%. Mostramos também que a partir de um conjunto de características previamente definido, o método é capaz de identificar quais as características mais discriminantes para determinados ambientes e tipos de eventos, tornando assim um método adaptável.

Palavras-chave: Rastreamento, Classificação de Eventos, Detecção de Eventos não-convencionais, Monitoramento de Vídeo

Abstract

This work presents an approach for real-time event detection of non-conventional events in video scenes. This approach consists of the real-time analysis of video from a surveillance camera to detect, segment and tracking of moving objects in scene, and classify the movement of these object as conventional or non-conventional. For each tracked object in the scene, features such as displacement, speed, position, changes in the direction and in the size of the bounding box enclosing the object are extracted along the trajectory of the objects in scene. These features are aggregated in a feature vector. During training, each feature vector is labelled by an human supervisor as representing a conventional or non-conventional movement and they are further stored in a database. During classification, feature vector extracted from the moving objects are matched against these labelled feature vectors stored in the database and an instance-based learning algorithm is employed to classify the object movement as conventional or non-conventional. Experimental results on real videos, shows that the proposed solution is able to identify non-conventional movements in real-time with accuracies between 78% and 82%. We also show that our approach is able to identify which characteristics are more discriminative, according to the enviroment and types of events, making the system adaptive.

Keywords: Tracking, Event Classification, Non-Conventional Event Detection, Video Surveillance

Capítulo 1

Introdução

Nos dias de hoje, crimes, assaltos, acidentes e ataques terroristas são alguns dos grandes problemas da sociedade. Para evitar estes incidentes é necessário desenvolver sistemas de vigilância inovadores, capazes de monitorar vários tipos de ambientes, como prédios, estacionamentos, lojas, shoppings centers, ruas, parques públicos, estádios, etc (Mecocci et al., 2003). O objetivo destes sistemas seria alertar instantaneamente quando uma ação suspeita está acontecendo ou quando há indícios de que possa acontecer. Atualmente, o uso de sistemas de monitoramento é bem difundido, no entanto, na maioria das vezes é utilizado após a ocorrência de um evento para recuperar as imagens armazenadas e identificar possíveis suspeitos ou as causas do evento, ou seja, não é usado como uma ferramenta de prevenção que possa identificar de forma instantânea esses eventos.

Os primeiros avanços na área de monitoramento computadorizado, nos anos 70, dizem respeito ao desenvolvimento de sistemas capazes de interagir com sensores (câmeras) espalhados pelo ambiente, onde as imagens captadas eram exibidas em um monitor, ampliando assim o campo de visão do operador humano (Figura 1.1). Desde então, avanços significativos têm sido feitos no desenvolvimento de novos sensores, na forma de armazenamento de dados, comunicação e transmissão. Além disso, algoritmos de inteligência artificial, de processamento de imagens e de visão computacional têm sido integrados a estes sistemas para aumentar o poder e a qualidade da informação passada para o operador, permitindo assim uma interpretação melhor e mais rápida da cena e também diminuindo o risco de falha humana.

Com todos esses avanços, a segunda geração destes sistemas, além de ampliar o campo de visão, é também capaz de identificar automaticamente o momento em que houve alguma alteração na cena e alertar operadores ou armazenar somente esses períodos. Alguns sistemas, como o proposto por Marchesotti et al. (2002), também diferenciam os tipos de objetos presentes em uma cena pelas suas características, o que é bastante útil



Figura 1.1: Sistemas de monitoramento

para identificar algum objeto estranho no ambiente, como por exemplo, diferenciar pessoas de carros em um estacionamento, em ruas, etc.

Para melhorar a eficiência dos sistemas de monitoramento e criar a terceira geração, técnicas de classificação de eventos vêm sendo amplamente estudadas (Hu et al., 2004; Gavril, 1999; Yilmaz et al., 2006). Assim, a terceira geração não se limitaria a somente alertar quando da ocorrência de uma alteração na cena, mas somente se esta alteração fosse relevante, ou seja, quando da ocorrência de um movimento não convencional ou da eminência de ocorrer um evento. Isso permitiria o uso dessa ferramenta como uma forma de prevenção.

A classificação de eventos em vídeos de segurança é uma área relativamente nova na computação e que cresce a cada dia em virtude da seu vasto campo de aplicação. Além da necessidade de criar uma ferramenta eficaz contra o crime, a identificação de eventos não convencionais em cenas de vídeo pode ser utilizada em outros contextos, como por exemplo: identificar quando uma pessoa sofre algum tipo de acidente dentro de casa (Lühr et al., 2003; Hara et al., 2002); atividades fora do comum em um escritório de trabalho ou infrações de trânsito em uma rua ou rodovia movimentada (Brand and Kettner, 2000), também são considerados não convencionais em seus respectivos contextos. Enfim, um evento não convencional, pode ser definido como uma ação que não pertence ao contexto.

1.1 Objetivos

Com base no exposto anteriormente, o objetivo deste trabalho é desenvolver um método capaz de identificar automaticamente e em tempo real, eventos não usuais em cenas de vídeo em diferentes ambientes. Através de uma câmera posicionada em um ponto estratégico do ambiente, as imagens são capturadas e processadas em tempo real.

Caso seja identificado uma situação fora do comum, um alarme ou uma mensagem no monitor deverá ser exibida para o operador.

Evitando o uso de abordagens complexas como as que utilizam os Modelos Escondidos de Markov (HMM), Modelos Escondidos Hierárquicos de Markov (HHMM), Cadeias de Markov, ou de características específicas de um objeto como postura, cor ou forma, a proposta consiste em extrair características que sejam independentes do tipo de objeto e do ambiente, como características do movimento ao longo do tempo e utilizar um classificador menos complexo para identificar o tipo de evento. O objetivo de utilizar um classificador de baixa complexidade, além de propiciar uma resposta em tempo-real, é que essa solução possa ser aplicada a determinados tipos de eventos, sem grandes esforços de adaptação ao ambiente.

Devido ao fato de levarmos em consideração apenas as informações relativas ao movimento dos objetos, evitamos que o sistema fique restrito a tipos específicos de objetos pelo uso de características específicas como contornos, cor, postura e etc, ou então características de ambiente como regiões de interesse, sensores espalhados no ambiente, seqüência de ações no ambiente, etc.

Para validar a idéia de adaptabilidade, em princípio aplicamos a solução em dois ambientes distintos. Inicialmente a solução estará restrita a ambientes onde haja baixo fluxo de objetos no campo de visão da câmera, pois focamos nossa pesquisa em classificação de eventos, e portanto não levaremos em conta casos de sobreposição, oclusão, etc, embora que para estes casos utilizamos o algoritmo proposto por Latecki and Miezianko (2006).

A Figura 1.2 apresenta uma visão geral dos principais elementos do método proposto. Basicamente, quando um objeto entra no campo de visão de uma câmera, são extraídas informações sobre o seu padrão de movimento. Essas informações são então processadas e é gerado um vetor de características. Este vetor de características, por sua vez, é passado a um algoritmo que classifica o tipo de movimento como sendo convencional ou não convencional, de acordo com os padrões estabelecidos para o ambiente. O método também conta com uma fase de treinamento, onde o algoritmo aprende o que são movimentos convencionais para o ambiente escolhido, e uma fase de monitoramento (classificação de novos eventos), onde os dados são passados para o classificador pré-treinado e este será o encarregado de distinguir automaticamente as ações. Dessa forma, acredita-se ter construído uma solução relativamente robusta e de fácil adaptação em ambientes distintos.

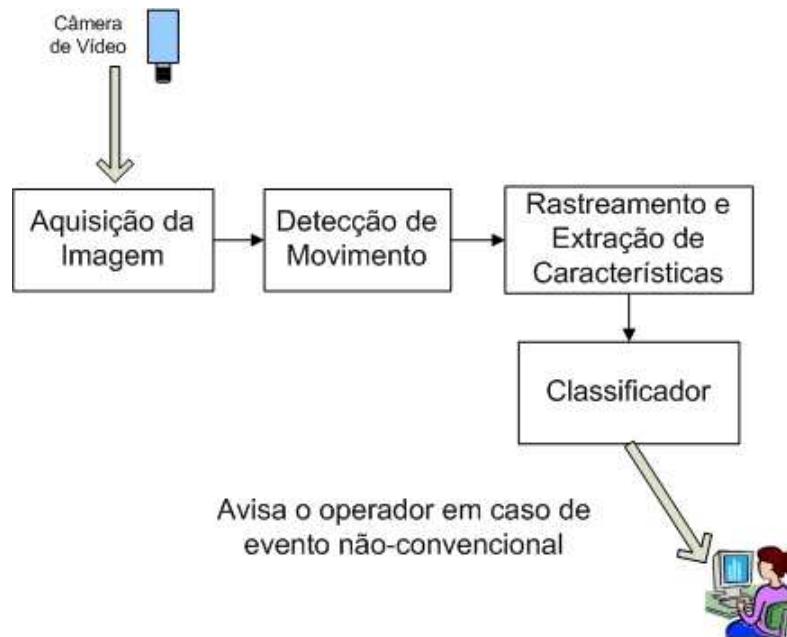


Figura 1.2: Elementos do método proposto

1.2 Desafios

Duda et al. (2001) revelam os principais problemas relacionados à área de reconhecimento de padrões. Segundo eles, o reconhecimento de padrões envolve várias etapas, dentre as quais segmentação, extração de características e classificação são as que mais se relacionam com nosso projeto. Resumidamente, a etapa de segmentação visa reduzir o campo de avaliação; a de extração de características tem por objetivo obter informações relevantes e discriminantes a respeito dos objetos a serem analisados; por fim, a etapa de classificação é responsável por decidir a classe a qual o padrão pertence, ou seja, se é convencional ou não em nosso contexto. Na etapa de segmentação o problema principal consiste em evitar que objetos que não interessam sejam erroneamente analisados como objetos de interesse. Já para a etapa de extração de características, o problema consiste em definir as principais características que representem de forma eficiente o padrão. Na classificação, o problema é saber em quantas classes iremos separar os padrões, como as classes serão formadas, bem como qual a fronteira de decisão entre elas.

Atualmente os sistemas capazes de classificar automaticamente eventos em cenas de vídeo são muito restritos. Basicamente, estes funcionam para um ambiente e evento específico, sob condições de iluminação controladas (Lühr et al., 2003)(Cucchiara et al., 2005a). Outra limitação é a adaptabilidade dos sistemas devido ao fato deles serem desenvolvidos para uma determinada finalidade, ou seja, a solução de um caso pode não servir para outro. Também, alguns sistemas necessitam que o ambiente seja adaptado

com sensores de movimento para que o funcionamento seja adequado, deixando a solução ainda mais restrita (Hara et al., 2002). Desta forma, o esforço necessário para aplicar uma mesma solução para demais ambientes seria quase que inviável.

Para que um algoritmo seja eficiente, deve lidar com três variáveis do ambiente que influenciam diretamente no resultado: luz, posição dos objetos e qualidade da imagem. Diferentemente de outros sistemas de visão computacional, onde podemos ter um ambiente controlado quanto à iluminação, posicionamento dos objetos, etc, a grande dificuldade encontrada nos sistemas de monitoramento visual é justamente a variabilidade destes fatores citados. Geralmente esses sistemas operam com câmeras que captam imagens coloridas, utilizando luz natural ou então luz artificial sem qualquer controle sobre a luminosidade ou outra característica da cena. E como se trata de um ambiente onde circulam pessoas ou objetos se movimentam, a variação da distância do objeto e sua movimentação no ambiente é muito grande.

Para que um sistema deste tipo seja adaptativo a qualquer ambiente, é necessário que seja capaz de lidar com estas variabilidades, ou pelo menos, minimizar o efeito sobre o resultado final. Outra questão que devemos levar em consideração nesse tipo de sistema, é o tempo de resposta ou a complexidade computacional para o processamento da informação. Quanto mais variáveis se tem para processar no ambiente, mais tempo se leva. A questão é relacionada com o número de variáveis que iremos trabalhar e a forma como essas variáveis interferem no resultado do sistema. Se houver um número excessivo da variáveis, o método poderá ficar lento, do ponto de vista de um operador humano. Um sistema que opere em tempo quase real, ou seja, tempo de resposta quase instantâneo, é de suma importância neste contexto, pois quanto mais rápido o operador for alertado, mais rápido uma ação preventiva ou corretiva poderá ser tomada.

1.3 Contribuições

Uma das principais contribuições do nosso projeto é a criação de uma solução de fácil adaptabilidade para determinados ambientes, sem a necessidade de processos complexos para integrar com o ambiente, e com taxa de acerto razoável, incluindo as três etapas necessárias em sistemas deste tipo, não limitando a solução apenas na etapa de classificação. Acreditamos este ser um ponto forte do nosso trabalho, pois poucas propostas apresentam como foco a questão da adaptabilidade. Em vários trabalhos, técnicas interessantes e eficazes foram desenvolvidas, porém a solução tornava-se restrita a um específico ambiente, movimento ou objeto. Com isto, levamos em consideração que uma boa solução seria desenvolver um método que utiliza características simples para classi-

ficar o movimento, tornando assim uma solução de baixa complexidade e por sua vez de fácil adaptabilidade em variados ambientes, com limitações mínimas.

Também como contribuição, foi feita uma seleção de características para definir quais são as características mais significativas no processo de detecção e classificação de eventos não convencionais. Através de um levantamento das características mais utilizadas nos trabalhos relacionados, construímos o conjunto de características e avaliamos quais apresentam um maior poder discriminante. Assim, esperamos contribuir para que em pesquisas futuras, as características possam ser escolhidas de acordo com os resultados que apresentaremos neste documento.

Em nível social, acreditamos que a nossa solução possa contribuir, no presente ou como motivação para pesquisas futuras, no combate a violência, prevenção de acidentes e demais situações que podem ser detectados eventos não-convencionais.

Os resultados da nossa pesquisa, foram publicados nas conferências “2007 IEEE International Conference on Systems, Man, and Cybernetics (SMC 2007)” (Hochuli et al., 2007b) e “2007 IEEE Pacific Rim Symposium on Image Video and Technology (PSIVT 2007)” (Hochuli et al., 2007a).

1.4 Organização do Documento

Nos próximos capítulos abordaremos em detalhes as técnicas disponíveis, bem como faremos um aprofundamento da nossa proposta. O documento está organizado da seguinte forma: No Capítulo 2 apresentamos uma revisão bibliográfica dos trabalhos relacionados. Na seqüência apresentaremos a solução proposta, detalhando as etapas do processo como detecção de movimento, rastreamento e classificação. Por fim, no Capítulo 4 apresentamos os resultados experimentais, enquanto o Capítulo 5 apresenta uma conclusão geral da proposta e trabalhos futuros.

Capítulo 2

Estado da Arte

O objetivo principal das pesquisas na área de segurança e monitoramento baseado em vídeos é desenvolver algoritmos inteligentes que substituam a forma tradicional como esse tipo de monitoramento é feito. Hoje, inúmeras câmeras são espalhadas pelo ambiente e um operador controla todo o sistema. Se for possível criar um sistema inteligente, as falhas humanas seriam reduzidas e assim aumentaria o nível de segurança.

Neste capítulo apresentamos uma revisão bibliográfica dos principais trabalhos que estão relacionados ao nosso projeto, de maneira particular às áreas: detecção e rastreamento do movimento, análise de comportamento e classificação de eventos.

2.1 Detecção e Rastreamento do Movimento

O primeiro passo nos sistemas de monitoramento é a detecção dos objetos em movimento em cenas de vídeo. A detecção ou segmentação do movimento é importante a fim de delimitarmos as regiões de interesse a serem analisadas. Existem diversas estratégias para encontrar as áreas onde houve movimento na cena de um vídeo. Segundo Hu et al. (2004) as três estratégias mais comuns são: subtração do fundo, diferença temporal e fluxo óptico.

A subtração do fundo, como o próprio nome diz, consiste em subtrair o quadro atual de uma imagem de fundo. O resultado dessa subtração contém os objetos que não pertenciam ao fundo ou que tiveram suas posições alteradas. O método possui baixa complexidade computacional, porém é sensível a pequenas variações do ambiente. Não é uma boa estratégia para ambientes com muita movimentação e que não tenham controle de iluminação.

A estratégia de diferença temporal consiste em se calcular a diferença da imagem atual com uma imagem anterior. Geralmente é usada uma imagem de 2 ou 3 quadros

anteriores. É uma estratégia interessante para ambientes com muita movimentação como shopping centers ou ambientes externos, porém apenas as bordas dos objetos são ressaltados, pois a diferença do interior dos objetos para eles mesmos é nula.

Na Figura 2.1, vemos dois exemplos das técnicas citadas acima. A Figura 2.1B apresenta a técnica baseada em subtração do fundo, enquanto a Figura 2.1C ilustra a técnica de diferença temporal.

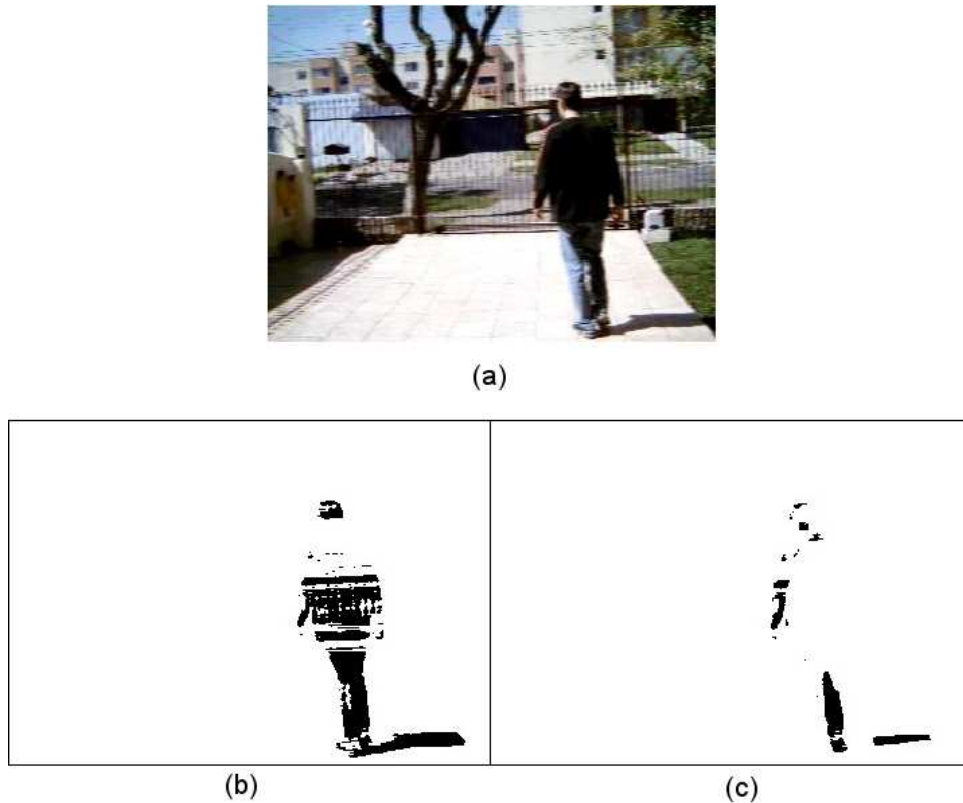


Figura 2.1: Segmentação do movimento
 (a) Quadro Atual. (b) Técnica baseada em Subtração da imagem de fundo. (c) Diferença Temporal entre 2 quadros consecutivos.

A análise do movimento através do fluxo óptico é feita por meio dos vetores de fluxo de cada objeto, obtido durante um certo período. A vantagem desta técnica é que ela é invariante a movimentação da câmera.

Para refinar a segmentação do movimento, fazendo com que o método se torne mais robusto, alguns autores propõem outros métodos aplicado a uma simples subtração de pixels. Em Stauffer and Grimson (2000), são utilizadas diversas Gaussianas para fazer a segmentação do movimento na imagem. A cada quadro os pixels em movimento são assimilados a uma gaussiana conforme sua variância, desvio padrão e peso. Todos os pixels que possam ser assimilados por uma gaussiana, fazem parte do fundo, caso não

exista uma gaussiana que o absorva, o pixel é então considerado como movimento. As gaussianas neste caso absorvem movimentos contínuos, o que é a grande vantagem desta técnica. Caso exista algum objeto que apresente um padrão de movimento uniforme, como por exemplo as pás de um ventilador, movimentos de ondas em uma lagoa ou a própria variação da luz, depois de um certo período este movimento é assimilado a uma gaussiana e classificado como fundo.

Lei and Xu (2005) complementaram o trabalho de Stauffer and Grimson (2000) utilizando mais algumas restrições para refinar a identificação de movimento. Além de absorver movimentos constantes, o método é capaz de minimizar o efeito das sombras dos objetos e também a variação da luz. Para eliminar sombras, a técnica consiste em verificar se o pixel classificado como movimento está mais escuro que o resto da região em movimento, este método é denominado *Brightness Distortion* ou Distorção do Brilho. Já para variações de luz repentinas, onde ocorre a detecção errônea de movimento, o problema pode ser suavizado adicionando variáveis de tamanho máximo de objetos e número máximo de objetos na cena.

Já em Gao and Zhou (2001) a estimativa do fundo é feita através de uma janela de tamanho pré-definido (sugerido, 100x100 pixels) que percorre a imagem. A área que a janela ocupa é denominada “*local-region*” (em português, região-local). É considerado que a variação da luz e a distribuição do ruído é uniforme dentro de cada janela. Para cada *local-region* da imagem de fundo, é calculada uma gaussiana. Posteriormente, em cada novo quadro, é computado o histograma da diferença entre a respectiva *local-region* do quadro atual e da imagem de fundo. Se a curva do histograma desta região aproximar-se da curva da respectiva gaussiana da imagem de fundo, então é classificado com fundo, senão é considerado movimento.

Ridder et al. (1995) utilizam uma abordagem baseada no filtro de Kalman. A filtragem por Kalman consiste em prever um novo estado do sistema comparando o valor previsto com o valor atual e obtém uma nova estimativa por meio de um peso obtido através da diferença entre o valor previsto e o valor atual. Para aplicações de estimação de fundo em imagens, basicamente o filtro de Kalman compara a diferença entre o fundo estimado e o valor do pixel na imagem atual. Se o peso obtido nesta diferença for maior que o limiar configurado, então o pixel é considerado como movimento.

O próximo passo depois de identificadas as regiões de movimento, é identificar o objeto e acompanhá-lo durante a cena. Esta etapa é denominada *Rastreamento*. Nesta etapa somente os objetos de interesse devem ser rastreados. Para isso devemos extrair informações a respeito do objeto de modo que seja possível identifica-lo.

Hu et al. (2004) também abordam as principais técnicas disponíveis para rastrear

o movimento. O rastreamento baseado em **regiões** consiste em analisar as regiões da imagem que houve movimento. Para isso, uma imagem é mantida como fundo e subtraída do quadro atual. No método baseado em **contornos**, os objetos são rastreados por linhas externas chamadas de *bounding boxes* que são atualizadas quadro a quadro. O objetivo destes algoritmos é prover diretamente a forma dos objetos e suas descrições de maneira mais eficiente que o método baseado em regiões. Já os baseados em **características**, extraem informações dos objetos a serem rastreados como:

- **Globais:** Perímetros, áreas, cores, etc;
- **Locais:** Linhas, cantos e curvas;
- **Gráficas:** Distâncias entre pontos do objeto, geometria das características;

E por fim temos o rastreamento baseado em **modelos**, onde os objetos a serem rastreados são comparados com modelos previamente projetados. A grande desvantagem deste método é o alto custo computacional, porém são métodos bastante eficientes.

Dentro deste contexto de rastreamento, a técnica utilizada em Lei and Xu (2005) consiste em extrair características do objeto em movimento como cor, forma e posição e então aplicá-las a uma função de custo para identificar se o objeto é pertinente ou não. Lok and Chan (2005) utilizaram os pontos rígidos do corpo humano para construir um modelo. São considerados como pontos rígidos os joelhos, tornozelos, quadril, ombros e cotovelo. A classificação do movimento de uma pessoa, é feita utilizando uma medida de semelhança entre os modelos.

Em Yilmaz et al. (2006), os autores apresentam algumas técnicas para representar um objeto, tais como:

- **Pontos:** O objeto é representado por um único ponto ou um conjunto de pontos. Seu uso é recomendado quando o objeto apresenta uma pequena área na imagem;
- **Formas Geométricas:** O objeto é representado por elipses ou retângulos. Embora o uso seja mais adequado em objetos rígidos, também pode ser usado em objetos não-rígidos;
- **Formas Articuladas:** O objeto é dividido em partes. É indicado para objetos não-rígidos;
- **Esqueleto:** Extrai-se um esqueleto do objeto. Uma espécie de “Contorno Interno”. Pode ser usado para objetos rígidos e não-rígidos;

- **Contornos e Silhuetas:** Como o próprio nome diz, exalta o contorno e a silhueta do objeto. É indicado para objetos altamente não-rígidos, por exemplo, uma pessoa muito próxima da câmera;

Na Figura 2.2, estão ilustrados as técnicas descritas acima.

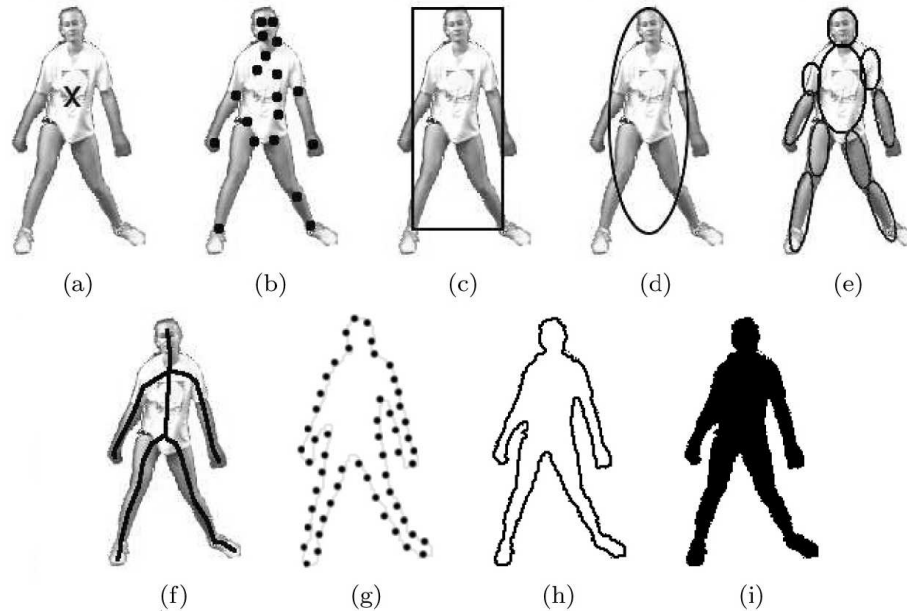


Figura 2.2: Representação dos objetos

(a) Centro de Massa. (b) Múltiplos Pontos. (c)(d) Retângulo e Elipse. (e) Baseado em partes do objeto. (f) Esqueleto. (g) Pontos do Contorno. (h) Contorno. (i) Silhueta.

Figura retirada de Yilmaz et al. (2006).

2.2 Análise de Comportamento e Classificação de Eventos

Na área de análise de comportamento e classificação de eventos, as pesquisas se concentram em técnicas baseadas em comparações de modelos (*Template Matching*) ou espaço de estados (*State-Space*) (Aggarwal and Cai, 1999). As técnicas baseadas em comparações de modelos consistem em extrair características ao longo do tempo do objeto em movimento e comparar, por meio de um classificador, com as características previamente armazenadas. Já o espaço de estados consiste em dividir o movimento em estados, e a transição entre esses estados é feita por meio de uma função de probabilidade. Um movimento, neste caso, é então identificado pela sua seqüência de estados. Por exemplo, o movimento “andar” tem uma seqüência de estados diferente do movimento “correr”.

A grande vantagem dos métodos baseados em espaço de estados é a robustez com relação a duração do movimento, porém a complexidade computacional dos sistemas baseados em comparações de modelos é inferior. A questão relacionada com a duração do tempo é determinar a exata fração de tempo que caracteriza um movimento, por exemplo, saber de quanto em quanto tempo as características precisam ser extraídas para se ter uma boa avaliação. Para solucionar este problema, Polana and Nelson (1994) normalizaram a duração do movimento aplicando ciclos, ou seja, para avaliar o movimento, mesmo que este seja longo demais, a cada intervalo de tempo o movimento é avaliado.

O uso do modelo de processo de Markov e uma função de tempo de transição de estado foi o método proposto por Hara et al. (2002). A idéia é identificar os hábitos de uma pessoa dentro de uma casa. O modelo de Markov é construído com base nos ambientes da casa (estados) e as transições com base nos hábitos da pessoa, ou seja, com base no uso dos ambientes. A transição entre os estados se dá por meio de uma função probabilística calculada de acordo com o uso dos sensores de presença. Juntamente é utilizada uma função de tempo de transição de estados para gerar o que autor denominou de “Diário de Atividades”. Neste “diário” estão anotadas a seqüência de ações e o tempo de cada uma. Uma ação não-usual nesse caso é determinada por uma seqüência diferente de estados do modelo de Markov ou por um valor de tempo fora da média. A desvantagem desta proposta é o uso de sensores de presença, ou seja, é necessário adequar o ambiente ao sistema, o que torna a solução muito restritiva e específica. Outro ponto negativo, é que é muito voltado aos hábitos da pessoa, então se outra pessoa ocupar a casa e tiver hábitos diferentes o sistema precisa ser recalibrado.

Em Mecocci et al. (2003), foi proposto um método baseado em protótipos de trajetórias de movimento. De cada objeto em movimento é extraído o seu “blob”. “Blob” é um componente conexo formado por uma seqüência de pixels *foregrounds* conectados, que juntos formam um objeto qualquer. De cada “blob” são extraídas informações sobre seu centróide (posição central), perímetro, etc, para futuramente computar uma trajetória. Na fase de treinamento, o sistema rastreia os objetos em movimento, computa as trajetórias normalizadas e gera protótipos dessas trajetórias. Depois de calibrado o sistema, na fase monitoramento, cada nova trajetória é comparada com os protótipos de trajetórias obtidos no treinamento. Se uma nova trajetória distanciar-se dos protótipos, um evento não-usual é detectado. Como este método computa as trajetórias dos objetos em movimento, pode ser aplicado em vários ambientes sem grandes modificações, ou seja, não leva em consideração características do objeto. Uma proposta para aumentar a eficiência, talvez seria a criação de protótipos de trajetórias conforme o período do dia, por exemplo, as trajetórias do período diurno seriam diferentes do período noturno.

Identificar ações dentro de uma casa de forma com que os objetos da casa interajam com a pessoa foi proposto em Lühr et al. (2003). O método consiste em modelar cada ambiente de uma casa e suas respectivas atividades utilizando o conceito de HHMM. O modelo foi organizado hierarquicamente da seguinte forma: ambiente da casa→possíveis atividades no ambiente→seqüência de estados de cada atividade. Nos experimentos foram selecionados os ambientes cozinha e sala de estar. Cada ambiente foi dividido conforme as regiões de interesse, em uma cozinha por exemplo as regiões de interesse seriam o fogão, geladeira, mesa de preparo e etc. Cada região de interesse representa um estado da seqüência de estados da atividade. Para a atividade cozinhar do ambiente cozinha, foram propostos quatro modelos (seqüência de estados) diferentes desta atividade para que seja possível identificar várias maneiras diferentes, ordem de estados diferentes, de executar a mesma tarefa. Exemplificando, a seqüência de estados da atividade cozinhar contém, não necessariamente nesta ordem, os estados “ir a geladeira”, “preparo”, “em frente ao fogão”, etc. Da mesma forma foram construídos cinco modelos para o ambiente da sala de estar. Um evento não usual ocorre quando uma seqüência de estados difere das existentes nos modelos.

O método é interessante embora tenha sido o primeiro teste, no entanto é necessário modelar cada possível atividade de cada ambiente e para cada atividade prever várias formas de executá-las. Isto torna a solução restrita a um determinado ambiente e pouco robusto, pois necessita-se remodelar a solução para cada novo ambiente ou quando o ambiente sofrer modificações. Também pode haver certa dificuldade em aplicar este método em ambientes externos com vários objetos ao mesmo tempo na cena. Brand and Kettner (2000) identificaram eventos em um escritório e em um cruzamento de trânsito, empregando um conceito similar ao de Lühr et al. (2003), porém utilizando HMM. Ao invés de organizar o modelo de forma hierárquica, os eventos são identificados por um conjunto de estados. Por exemplo, o evento “ao computador” é representado por um conjunto de estados diferente de outro evento, porém a rotulação de quais estados pertencem a um evento foi feita de forma manual, analisando visualmente os resultados obtidos. A detecção de anomalia é determinada quando a função de probabilidade do HMM assume um valor muito baixo para esta transição.

O uso das informações de movimento do objeto, também foi utilizado em Niu et al. (2004). Após a detecção de movimento baseada em uma diferença entre quadros, o objeto é então rastreado durante todo o restante da cena. A cada espaço de tempo determinado são extraídas informações sobre sua posição e velocidade. Com essas informações é projetada uma curva do novo movimento e comparada com curvas de movimentos conhecidos previamente calculados na fase de treinamento. Foi concluído que cada tipo de evento/a-

tividade possui um tipo de curva característica. Assim, a atividade pode ser classificada através da comparação entre a curva do movimento atual com a curva dos movimentos conhecidos. Não há restrições com relação ao tamanho do objeto ou distância da câmera uma vez que as informações são obtidas do movimento. Também devido à baixa complexidade computacional o uso é indicado para aplicações que necessitem de processamento em tempo real. Contudo, o uso em ambientes com muita movimentação em todos os sentidos como ruas ou shoppings, pode atrapalhar na projeção de curvas e conseqüentemente a eficiência do sistema.

O uso da postura da pessoa para classificar uma situação não-usual dentro de uma casa foi apresentado em Cucchiara et al. (2005a,b). Como se trata de um ambiente onde oclusões do objeto em movimento são consideradas normais, devido a existência de objetos fixos como mesas, cadeiras ou estantes, foi proposto um método probabilístico para prever a próxima posição depois de uma oclusão. Para a classificação são extraídos da imagem somente os pixels referentes ao corpo da pessoa (silhueta), e com base nisso é calculado o histograma. É conhecido previamente os histogramas de quatro posturas: em pé, sentado, agachado e deitado. Através de uma diferença de histogramas, entre o histograma da silhueta atual e os histogramas previamente calculados, é possível identificar a postura da pessoa. Como o próprio autor comenta, a grande falha deste artigo é a distância da câmera em relação ao objeto, pois quanto mais longe, menor será diferença de histograma entre as posições. Também há erros de classificação entre as posturas agachado e sentado por apresentarem histogramas parecidos.

Wang (2006) implementa três formas diferentes de avaliar a postura de um humano. O primeiro método, denominado *Inner Distance shape context (ISDC)*, é baseado nas distancias entre os pontos do contorno e o ponto de origem do “blob”. No treinamento é gerado um contorno médio de todos os tipos de postura. Na classificação é computada a distância entre os pontos do contorno do blob com o contorno médio de cada postura. No segundo método, *Fitted Ellipse (FE)*, para cada blob é extraída uma elipse que o contorna. No treinamento é feita uma distribuição da orientação da elipse e da proporção entre o maior e o menor eixo de cada postura. Na classificação é usada a distância de Mahalanobis para comparar a orientação e proporção dos eixos da respectiva elipse com as demais elipses da base. O terceiro método, *Projeção de Histograma*, é feita uma projeção de histograma horizontal e vertical do blob. No treinamento é feito uma projeção média das posturas. Na classificação a soma absoluta das diferenças das projeções verticais e horizontais são utilizadas para comparação. O autor comenta que cada uma das técnicas pode ser usada separadamente para classificação de postura, utilizando a classe com o menor valor de similaridade. No entanto para aumentar a eficiência ele propõe utilizar a

regra de votação para escolher qual a classe escolhida para a postura.

Niu and Abdel-Mottaleb (2004) apresentam uma solução que combina as características do movimento e da postura para classificar o tipo de evento. Para avaliar o movimento, primeiramente a região de interesse é dividida em 64 regiões horizontais. Para cada região é computado o fluxo ótico médio de cada região, formando assim um vetor de 128 posições, considerando os fluxos x e y . Para extrair a informação sobre a postura a região de interesse é normalizada em 64×48 pixels, o que forma um vetor de 3072 posições. Para diminuir a dimensão, é aplicado o algoritmo PCA sobre o vetor. A ideia é que cada postura apresente uma forma diferente na imagem, sendo assim o vetor de cada postura seria projetado pelo PCA em um espaço distinto. Para modelar os eventos, um HMM é construído com base em uma seqüência de vetores extraídos ao longo do tempo na cena. Para que o sistema seja invariante a posição, foram capturados os movimentos em oito ângulos diferentes. Para cada ângulo e cada evento foi criado um modelo. Nos testes foram realizados experimentos utilizando apenas fluxo ótico, apenas postura e as duas juntas. Apesar dos testes terem mostrado bons resultados, o ponto fraco dessa solução é similar ao anterior, ou seja distância da camera para classificar postura. Além disso o uso de vários HMMs para modelar os movimentos de vários ângulos, tornar a solução um pouco pesada.

Classificar a ação do jogador em uma partida de tênis, proposta em Zhu et al. (2006), consiste em extrair as magnitudes do fluxo ótico dentro da região de movimento (Blob). Dessa forma obteríamos para qual sentido (x,y) as partes do blob se movimentaram. A região do blob é dividida em 3 regiões verticais. Para a primeira e última região são computados histogramas de fluxo ótico. Esses histogramas são passados como características para um classificador do tipo máquina de vetores de suporte. Em resumo a ideia consiste que um movimento de rebatida para a esquerda apresente na região esquerda do blob, um maior numero de fluxo ótico que o da direita. A mesma analogia é feita para o movimento de rebatida para a direita. Embora este artigo esteja um pouco fora do escopo, mostra que utilizando apenas informações do movimento, pode-se classificar eventos distintos com uma certa precisão.

Wu et al. (2005) propõem duas soluções de baixo custo computacional. Uma para ambientes com um menor fluxo de pessoas e outra para ambientes com uma maior fluxo, como ruas, shoppings e etc. No primeiro método são computadas as distâncias entre as bordas do objeto para o centro médio do objeto. Essas distâncias são transformadas em coeficientes da transformada discreta de Fourier (DFT). Sobre esses coeficientes é aplicado um algoritmo de análise de componentes principais (PCA) para reduzir e normalizar o número de características. Por fim é aplicado um SVM para classificar um movimento com

base nas características geradas pelo PCA. A idéia de extrair as distâncias é que conforme o movimento da pessoa, essas distâncias variam indicando um movimento anormal, como agachado, carregando objetos ou correndo. No entanto, o autor comenta que este método é válido em ambientes que tenham pouco fluxo, pois a extração do blob se torna fácil. Para ambientes com maior fluxo de pessoas, onde é muito complexo definirmos um fundo da cena fixo (constantes), são computadas as velocidades dos pixels por meio de fluxo ótico. Através de um limiar de velocidade pré-definido, os pixels são rotulados como convencionais ou não convencionais conforme a sua velocidade. O artigo não mostrou resultados em números a respeito segundo método, porém comentou que apresentou-se bem em situação de correndo ou balançando a mão em meio a multidão. Na primeira técnica, a taxa de acerto variou entre 82% e 86% dependendo do tipo de evento classificado. No entanto recaímos no grande problema da postura que é a distância da câmera e a posição lateral para obtermos a postura.

2.3 Discussão

Foram apresentadas nesta seção algumas das propostas existentes para identificar eventos em vídeos, aplicado em vários contextos diferentes. Porém o que foi levado em questão é a forma como as informações são obtidas e como a classificação do evento é feita, independente do contexto aplicado. O que podemos perceber é que a grande maioria dos artigos apresenta soluções que apesar de os resultados mostrarem uma boa taxa de acerto, ainda são um pouco restritas a determinados objetos ou ambientes. Os modelos que se baseiam na postura por exemplo, são dependentes da distância da câmera em relação ao objeto, pois em uma distância grande os objetos apresentam-se pequenos e dificulta a classificação da postura, sem contar que para classificação da postura precisamos de uma visão lateral do objeto. Além disso, em objetos rígidos como carros, o uso da postura, contorno ou forma como única característica seria pouco discriminativa por não haver variação da mesma. Já para os sistemas baseados em modelos (HMM, HHMM), muitos estão restritos aos ambientes. Essas soluções necessitam de um modelamento das ações no ambiente, ou seja, o modelo construído para um ambiente dificilmente serve para outro ambiente. Outro ponto é que a modelagem do ambiente ou dos eventos, é uma tarefa um pouco complexa. Algumas soluções foram propostas com base nas informações do movimento, no entanto nada foi comentado na questão da adaptabilidade do algoritmo. Além disso alguns usam a informação do movimento em conjunto com alguma característica específica do ambiente ou um classificador de alta complexidade, o que acaba “amarrando” a solução.

Com base no exposto anteriormente, nossa proposta pretende focar na questão da adaptabilidade e da escolha do melhor conjunto de características para a classificação de eventos. Utilizando características que não estejam relacionadas com o ambiente e nem com o tipo de objeto, apresentamos uma solução de fácil adaptabilidade em determinados ambientes. Para isto, utilizamos características relacionadas ao movimento, como velocidade, posição, direção e deslocamento. Desta forma, as características ficam independentes do tipo do objeto, cor, forma, etc. A única exceção com relação ao movimento, é a característica de variação da *Bounding Box*, que descreve a variação do tamanho do objeto na trajetória. No entanto não é uma característica que descreve um único objeto ou um movimento singular, e sim uma característica que descreve a variação do tamanho de qualquer objeto ao longo da cena.

Capítulo 3

Metodologia

Em nossa abordagem procuramos desenvolver um método que leva em consideração apenas informações relativas ao movimento, ignorando informações únicas dos objetos como tamanho, cor, geometria. Dessa forma não restringimos nossa proposta apenas para alguns tipos de objetos, facilitando assim a adaptação do sistema em ambientes com tipos de objetos diferentes. A metodologia consiste em através de uma câmera posicionada estrategicamente no ambiente, capturar e processar em tempo real os movimentos ocorridos. A idéia é que o algoritmo seja capaz de identificar um movimento que não seja comum ao ambiente, ou seja, um movimento não-convencional.

A solução proposta é dividida em três etapas conforme ilustra a Figura 3.1. Na primeira, denominada detecção de movimento, processamos um quadro do vídeo e detectamos regiões onde houve movimento. A segunda etapa consiste em extrair, se houverem, objetos em movimento (dentro das regiões de movimento), e rastreá-los enquanto eles permanecerem no campo de visão da câmera, essa etapa é denominada rastreamento. Com base em informações do movimento do objeto, acompanha-se o objeto durante sua trajetória e paralelamente extrai-se características do movimento. Na terceira e última etapa, as características extraídas na etapa anterior são repassadas a um classificador que, com base nos vetores de referência armazenados durante a fase de treinamento, classifica o movimento como convencional ou não-convencional.

O foco da nossa pesquisa é a identificação de eventos não-convencionais e avaliação de características para classificação de eventos. Por conta disto, não aprofundamos muito nas linhas de pesquisa de detecção de movimento e rastreamento (*tracking*), o que implica em restrições do algoritmo. Para um bom funcionamento devemos ter pouco fluxo de objetos simultaneamente na cena, evitando assim casos de sobreposição e oclusão de objetos, que serão detalhados posteriormente. No desenvolvimento deste projeto foi utilizado

a biblioteca de processamento de imagens e visão computacional *OpenCV*¹. Operações básicas sobre imagens foram realizadas utilizando as funções dessa biblioteca. Os algoritmos restantes, com por exemplo, os algoritmos de extração de características, algoritmos de tracking, detecção de movimento, enfim, funções mais complexas, foram implementadas usando linguagem de programação C.

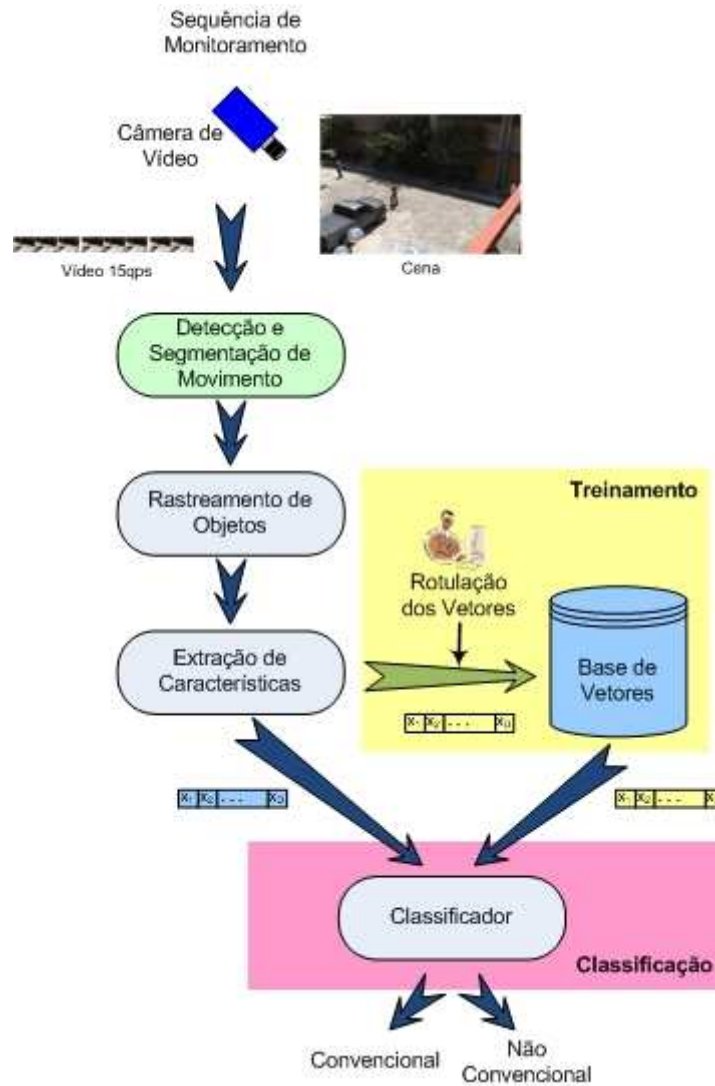


Figura 3.1: Visão geral do sistema.

Nas seções seguintes apresentamos cada etapa do método proposto em detalhes.

3.1 Detecção de Movimento

A detecção de movimento consiste em segmentar as regiões da imagem nas quais ocorreu algum movimento. O processo tradicional consiste em dada uma imagem de fundo

¹<http://sourceforge.net/projects/opencvlibrary/>

da cena, subtrair os quadros subsequentes do vídeo deste fundo. Os pixels resultantes dessa subtração, aplicado a um limiar, são considerados movimento. No entanto uma simples subtração ocasiona o que chamamos de falsos movimentos. Falsos movimentos são ocasionados por diferença de iluminação, sombras de objetos e até mesmo ruídos gerados pela compactação do vídeo.

Como exposto anteriormente, diversas técnicas foram desenvolvidas para detectar movimento eliminando ou minimizando o efeito desses falsos movimentos. Em nossa abordagem escolhemos a técnica proposta por Stauffer and Grimson (2000) aplicada a uma subtração para um fundo fixo.

Segundo Stauffer e Grimson, uma variação constante dos pixels (variações de luz, movimentos repetitivos, etc) em uma determinada região, pode ser assimilada por uma gaussiana conforme sua variância, desvio padrão e peso. A cada quadro, os pixels que são absorvidos por uma gaussiana são considerados parte do fundo. Os pixels que não são absorvidos, são então considerados movimento. A grande vantagem desta técnica é a absorção de movimentos contínuos que não interessam para o rastreamento, como por exemplo, ondas do mar ou as pás de um ventilador. Depois de um certo período, devido ao seu padrão constante, este movimento é assimilado por uma gaussiana e considerado como sendo parte do fundo da cena.

No entanto, pequenos ruídos ou bruscas variações de iluminação são ainda detectados como movimento. Como uma contribuição a técnica proposta por Stauffer e Grimson, adicionamos uma subtração para um fundo fixo para confirmar se o pixel é realmente um movimento. Embora essa subtração aparenta ser um processo repetitivo, em nossos testes percebemos que alguns ruídos remanescentes foram eliminados por essa subtração. Esse processo é ilustrado pelo algoritmo “Movimento”.

MOVIMENTO (*imagem*[0...*largura*][0...*altura*], *background*[0...*largura*][0...*altura*], *imagemBinaria*[0...*largura*][0...*altura*])

```

1
2 // Entrada: entrada: imagem[], background[]
3 // Saída: imagemBinaria[]
4 for i ← i + 1 until i ≤ largura do
5   for j ← j + 1 until j ≤ altura do
6     pixel ← imagem[i][j];
7     absorveu ← 0;
8     for k ← k + 1 until k ≤ nGaussianas do
9       // pertence(): Função que verifica se o pixel pertence há uma gaussiana ou não.
10      if pertence(Gaussiana[k], pixel) then
```

```

11     absorveu ← 1;
12     endif
13     endfor
14     if absorveu = 1 then
15         imagemBinaria[i][j] ← 0;
16     else if ||pixel − background[i][j]|| ≤ limiar then
17         imagemBinaria[i][j] ← 0;
18     else
19         imagemBinaria[i][j] ← 255;
20     endif
21 endfor
22 endfor

```

No entanto, para objetos que se movem lentamente, apenas as bordas são ressaltadas sendo que o interior do objeto é absorvido rapidamente. Assim os pontos do objeto ficam esparsos, conforme ilustra a Figura 3.3b. Para conectarmos os pontos das bordas do objeto, sem alterar seu tamanho através de uma dilatação, aplicamos uma subtração com um fundo fixo nestas regiões. Então, para cada pixel não absorvido por uma Gaussiana, é aplicado uma janela de dimensão 3×3 , e dentro desta janela é feita a subtração com um fundo fixo. Assim, se o pixel pertencer a um objeto em movimento, os pixels vizinhos deste objeto que foram absorvidos serão ressaltados pela janela, pois eles não pertenciam ao fundo. A Figura 3.2 mostra um comparativo entre uma simples subtração de fundo e a técnica utilizada. Essa etapa possui um resultado similar ao processo morfológico de fechamento, no entanto não deforma o contorno e apenas os pixels do interior do objeto são ressaltados. O algoritmo “Realce” abaixo ilustra esta etapa.

```

REALCE (imagemBinaria[0...largura][0...altura], imagem[0...largura][0...altura],
background[0...largura][0...altura], imagemRealce[0...largura][0...altura])

```

```

1
2 // Entrada: entrada: imagemBinaria[], imagem[], background[]
3 // Saída: imagemRealce[]
4 for i ← 3 until i ≤ largura − 3 do
5     for j ← 3 until j ≤ altura − 3 do
6         pixel ← imagemBinaria[i][j];
7         if pixel == 255 then
8             for iJan ← i − 3 until iJan ≤ i + 3 do
9                 for jJan ← j − 3 until jJan ≤ j + 3 do
10                    pixelBg ← imagem[iJan][jJan];
11                    if ||pixelBg − background[iJan][jJan]|| ≥ limiar then

```

```

12     imagemRealce[iJan][jJan] ← 255;
13     else
14     imagemRealce[iJan][jJan] ← 0;
15     endif
16   endfor
17 endfor
18 endif
19 endfor
20 endfor

```

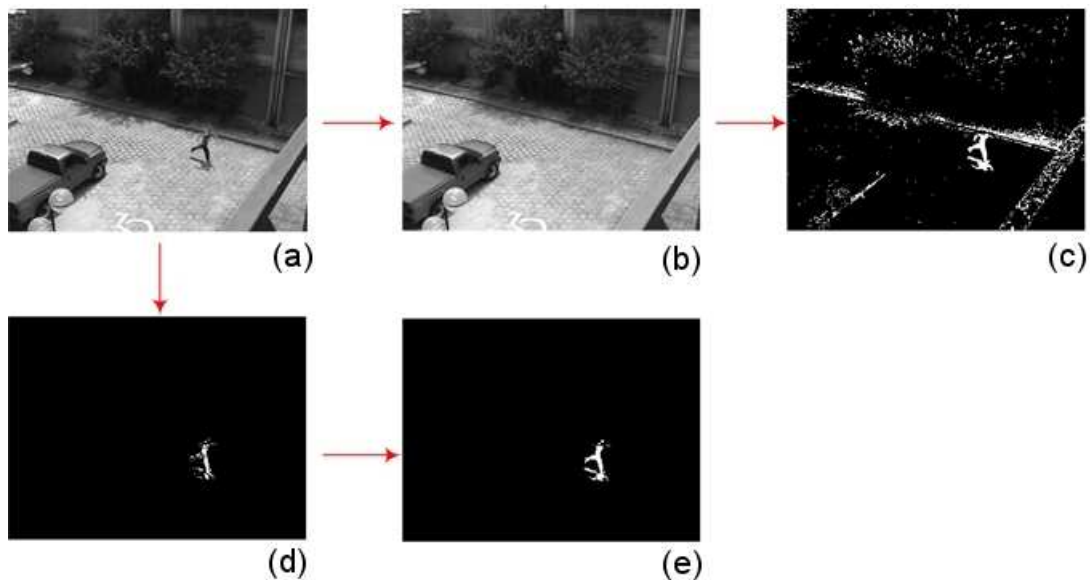


Figura 3.2: Comparativo entre a técnica utilizada e a subtração para um fundo fixo. (a) Quadro atual, (b) Quadro de fundo. (c) Imagem resultante da subtração do quadro atual com o quadro de fundo. (d) Imagem resultante da aplicação do método proposto por Stauffer and Grimson (2000), (e) Imagem resultante da aplicação do algoritmo de realce

Para eliminar os ruídos remanescentes é aplicado um filtro da mediana com janela de dimensão 3×3 . O resultado parcial é um conjunto de pixels do objeto em movimento, podendo haver, porém, pixels desconectados. Para que estes pixels formem um único objeto é utilizado um algoritmo de detecção de contornos baseado em aproximação poligonal, implementado na biblioteca *OpenCV*. Desta forma, o que antes era um conjunto de pixels do objeto torna-se um único objeto com todos os pixels conectados chamado de *blob*. A Figura 3.3 ilustra de maneira simplificada a detecção e segmentação do movimento. Uma vez identificado um *blob*, o mesmo deve ser rastreado enquanto permanecer no campo visão da câmera.

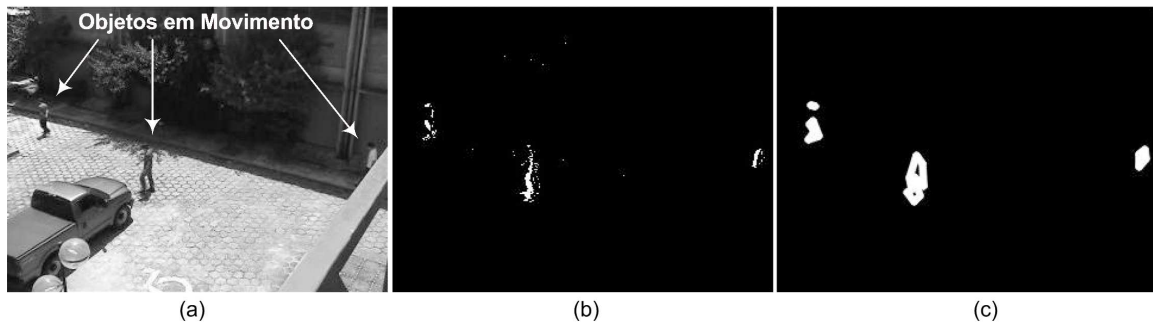


Figura 3.3: Exemplo de detecção e segmentação do movimento: (a) quadro original do vídeo com objetos em movimento, (b) segmentação do movimento por gaussianas, (c) *blobs* resultantes após aplicação de filtros, subtração de fundo e detecção de contornos.

3.2 Rastreamento

O rastreamento consiste em perseguir um objeto em movimento durante sua permanência em cena. Para eliminarmos objetos que não interessam ao rastreamento é aplicado um filtro de tamanho baseado na largura e altura esperada do objeto de interesse, nas Figuras 3.4 e 3.5 podemos ver exemplos de um objeto aceito e outro rejeitado. A idéia de se aplicar filtros para eliminar regiões de movimento indesejáveis foi proposta por Lei and Xu (2005), no qual a filtragem é feita de acordo com a velocidade e direção do movimento aplicadas a uma função de custo. Então um objeto é considerado de interesse se:

$$Alt_{min} < Alt < Alt_{max} \quad (3.1)$$

$$Larg_{min} < Larg < Larg_{max} \quad (3.2)$$

na qual Alt e $Larg$ são as respectivas altura e largura do objeto, Alt_{min} , Alt_{max} , $Larg_{min}$, $Larg_{max}$ são os limiares máximos e mínimos de altura e largura. Esses valores foram definidos através de um processo empírico, no qual um mesmo vídeo de validação era processado e os valores eram ajustados até que os objetos de interesse fossem aceitos pelo algoritmo e os demais recusados.

Para rastrear o objeto e atualizá-lo durante sua permanência em cena utilizamos um método similar ao proposto por Latecki and Mieziako (2006), porém, com algumas modificações. Suponha um objeto O^i no quadro F^n , onde O^i é um objeto sendo rastreado. No quadro seguinte, F^{n+1} , dadas j regiões de movimento (R^j), temos que saber qual R^j representa o objeto O^i do quadro anterior, este caso está ilustrado na Figura 3.6. Para isto utilizamos a seguinte função de custo:

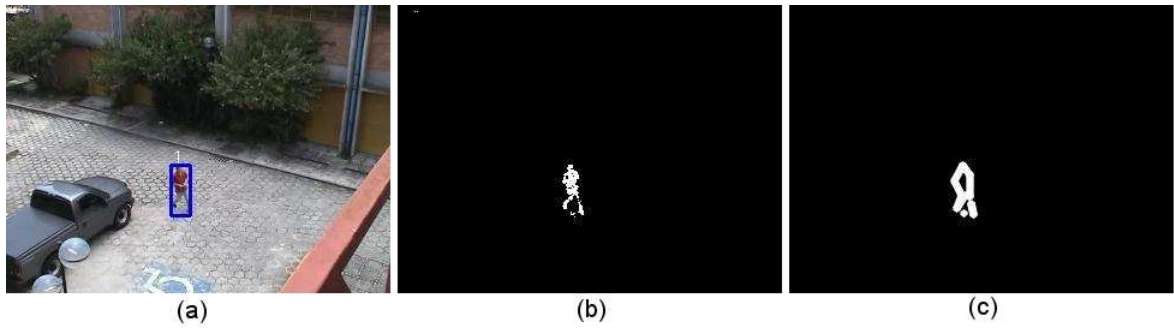


Figura 3.4: Objeto aceito: (a) objeto aceito com a *Bounding Box*. (b) Segmentação do movimento. (c) Contornos do objeto

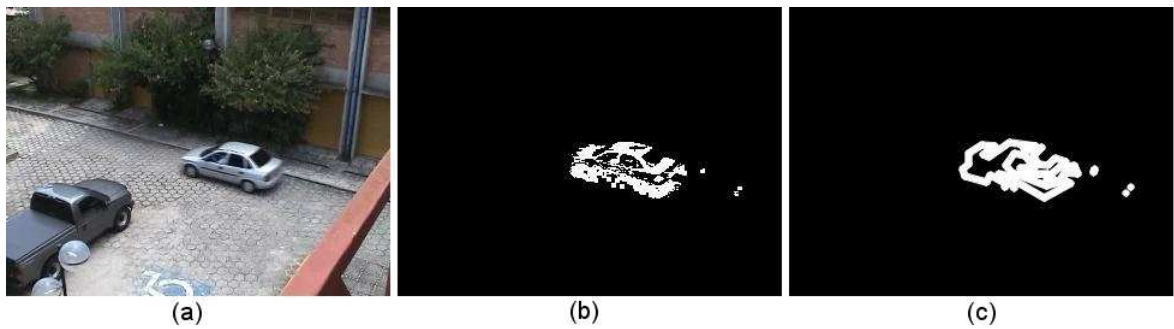


Figura 3.5: Objeto rejeitado: (a) Objeto rejeitado sem *Bounding Box*. (b) Segmentação do movimento. (c) Contornos do objeto

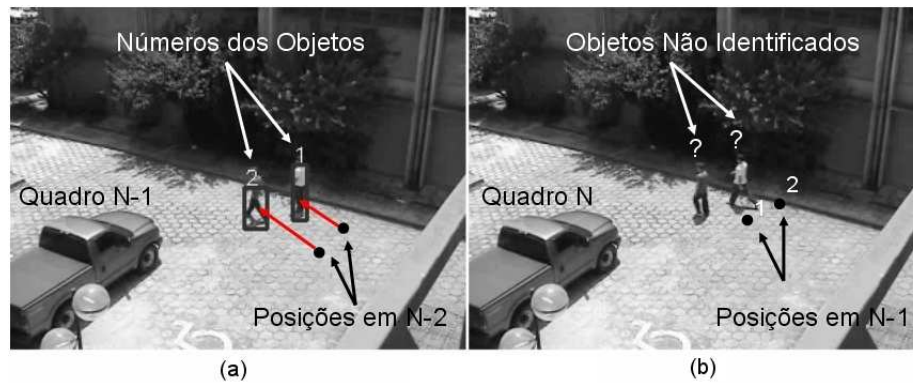


Figura 3.6: A. Objetos rastreados no quadro anterior, B. Objetos desconhecidos no quadro atual

$$Custo = (wP \times dP) + (wS \times dS) + (wD \times dD) + dT \quad (3.3)$$

na qual $wP, wS,$ e wD são pesos cuja soma deve ser menor ou igual a 1 e dP é a distância euclidiana em pixels entre os centros da região de movimento e do objeto, dS é a diferença de tamanho entre as *bounding boxes* da região de movimento e do objeto, dD é a diferença

de direção entre a posição do objeto estimada pelo algoritmo Lucas and Kanade (1981) no quadro atual, e da diferença entre os centros da região de movimento e do objeto e dT é a diferença do custo de persistência (TTL, *time to live*) do objeto. Estes parâmetros são definidos a seguir:

$$dP = |R_c^j - O_c^i| \quad (3.4)$$

na qual R_c^j é o centro da região de movimento e O_c^i é o último centro conhecido do objeto. O valor de dP não deve ser superior a um limiar de proximidade medido em pixels. Este limiar de proximidade varia de acordo com os objetos que se deseja rastrear, principalmente pela velocidade de deslocamento desses objetos na cena.

$$dS = \frac{|R_r^j - O_r^i|}{(R_r^j - O_r^i)} \quad (3.5)$$

na qual R_r^j e O_r^i representam o tamanho da *bounding box* da região de movimento e do objeto respectivamente.

$$dD = |\arctan(O_s^i - O_c^i) - \arctan(R_c^j - O_c^i)| \quad (3.6)$$

na qual O_s^i é a posição estimada do objeto no quadro atual. O valor do ângulo deve ser entre 0 e 2π .

$$dT = (TTL_{MAX} - O_{TTL}^i) \quad (3.7)$$

na qual TTL_{MAX} é a persistência máxima em quadros e O_{TTL}^i é a persistência do objeto. Se o objeto for encontrado no quadro atual, o valor de O_{TTL}^i é ajustado em TTL_{MAX} , caso contrário é decrementado por um até O_{TTL}^i ser igual 0, onde o objeto deverá ser eliminado do rastreamento. O valor sugerido para TTL_{MAX} é três vezes a taxa de quadros por segundo do vídeo (qps), no entanto este valor pode ser adaptativo de acordo com a velocidade dos objetos na cena.

Cada objeto do quadro anterior deverá ser assimilado pela região de movimento do quadro atual que obtiver o menor custo. Os valores dos centros e da *bounding box* dos objetos passam a ser os valores das regiões de movimento. Caso haja uma região de movimento que não foi atribuída a nenhum objeto, então um novo objeto deverá ser criado com os valores desta região de movimento. Caso exista um objeto que não foi assimilado por nenhuma região de movimento, este objeto poderá estar ocluso, e neste caso o algoritmo *Lucas-Kanade* irá falhar na estimação do movimento. Neste caso, é feita a predição de movimento deste objeto como:

$$O_s^i = S \times O_s^i + (1 - S) \times (R_c^j - O_c^i) \quad (3.8)$$

na qual S é o valor de velocidade fixo. A região de movimento R_c^j , deverá ser a região mais próxima ao objeto respeitando o limiar de proximidade. Então a posição estimada do objeto e de sua respectiva *bounding box* será computada respectivamente por:

$$O_c^i = O_c^i + O_s^i \quad (3.9)$$

$$O_r^i = O_r^i + O_s^i \quad (3.10)$$

3.3 Extração de Características

Dado um instante t da trajetória de um objeto de interesse, são extraídas características do movimento que formarão o vetor de característica indicado por V_i . O vetor é composto por cinco características:

$$V_i = [Vel, Pos, Desl, Bbox, Dir] \quad (3.11)$$

na qual Vel representa a velocidade do objeto, Pos indica a coordenada x, y do objeto na cena (ver Figura 3.7), $Desl$ indica o deslocamento do objeto em x e y , $Bbox$ indica a variação do tamanho da *bounding box* em x e y e Dir indica a variação na direção do objeto. As características são computadas como se segue.

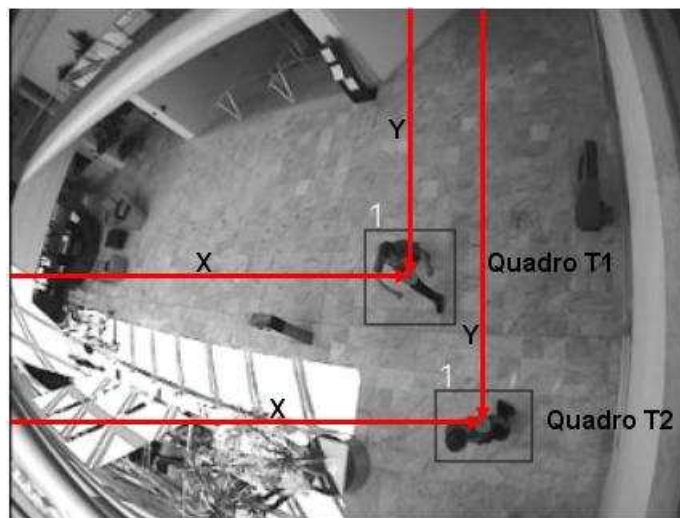


Figura 3.7: Posição do objeto na cena

A característica velocidade é obtida pela distância percorrida em pixels entre dois centros do objeto (ver Figura 3.8), O_c^i , em quadros consecutivos, $t-1$ e t , em um intervalo de tempo Q como segue:

$$Vel = \sqrt{\frac{(O_{c_{t-1}}^i - O_{c_t}^i)^2}{Q}} \quad (3.12)$$

A característica velocidade foi escolhida, por ser uma característica que permite distinguir objetos rápidos dos objetos lentos.

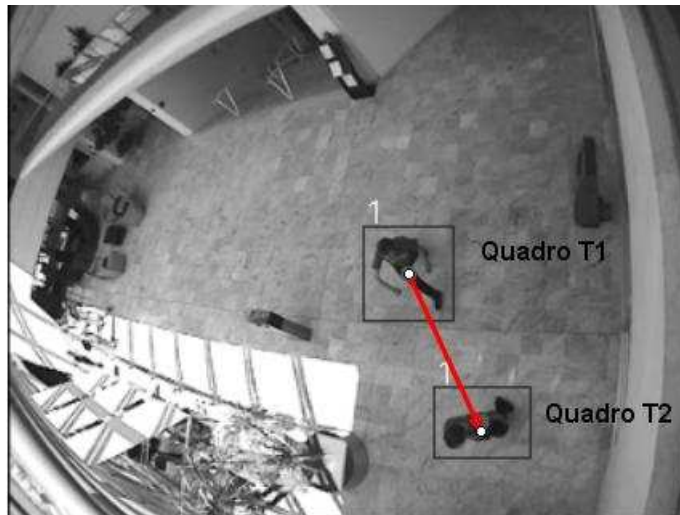


Figura 3.8: Velocidade entre dois quadros consecutivos

O deslocamento (em x e y , ver Figura 3.9) é medido utilizando a diferença entre dois centros do objeto, em quadros consecutivos $t-1$ e t , conforme a fórmula abaixo:

$$Desl = O_{c_{t-1}}^i - O_{c_t}^i \quad (3.13)$$

O deslocamento permite distinguir eventos pelo seu sentido, por exemplo, andar para cima ou para baixo.

A variação do tamanho da *Bounding Box*, ilustrado na Figura 3.10 é medido utilizando a diferença do tamanho entre duas *Bounding Boxes* do objeto, como segue:

$$Bbox = |O_{r_{t-1}}^i - O_{r_t}^i| \quad (3.14)$$

O uso da variação da *Bounding Box* é similar ao uso da postura, pois uma variação no corpo de um objeto não-rígido altera o tamanho de sua *bounding box*.

A variação da direção, ilustrado na Figura 3.11, permite identificar variações na trajetória do objeto. É calculada com base na diferença das direções de duas posições

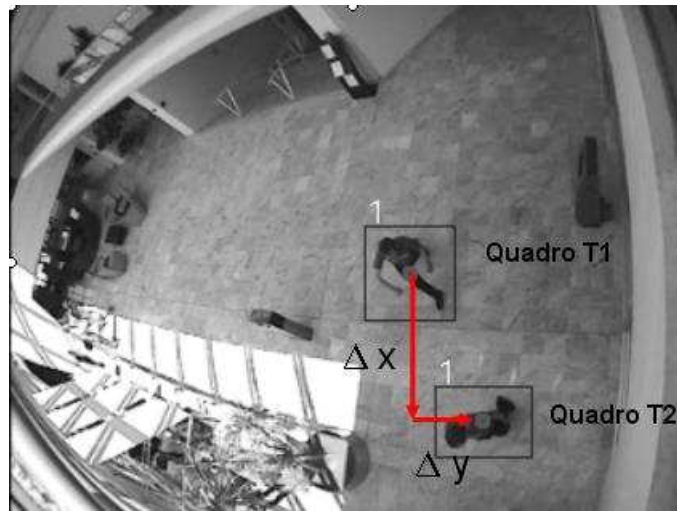


Figura 3.9: Característica deslocamento

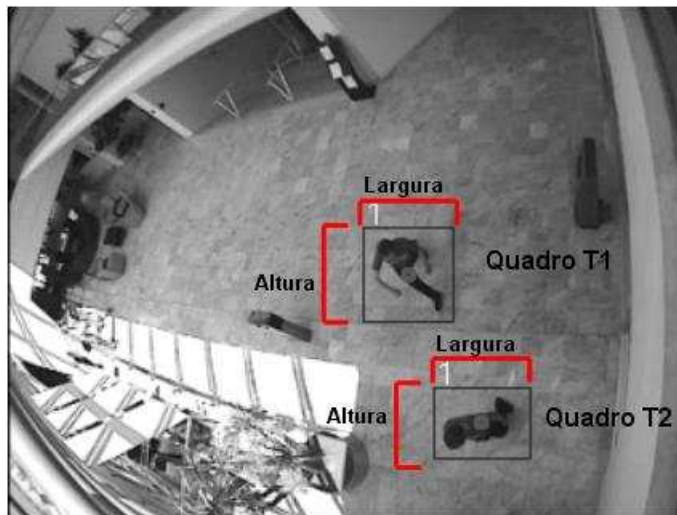


Figura 3.10: Característica variação da *Bounding Box*

consecutivas, em quadros consecutivos $t - 1$ e t , conforme a fórmula abaixo:

$$Dir = \arctan(O_{ct-2}^i - O_{ct-1}^i) - \arctan(O_{ct-1}^i - O_{ct}^i) \quad (3.15)$$

O intervalo de tempo para extrair as características deve ser definido de forma que não seja muito curto, pois as características ficariam repetidas, ou seja, sem grandes variações. Por outro lado se o intervalo for longo demais perderemos informações de modo que as variações das características ficariam “desconexas” e não fariam muito sentido. Em nosso caso, a extração de características é feita em um intervalo de Q quadros. Este valor é definido empiricamente e em nossos experimentos foi utilizado $Q = 3$. O valor de Q depende também do tamanho do objeto e da taxa de quadros do vídeo sendo capturado.

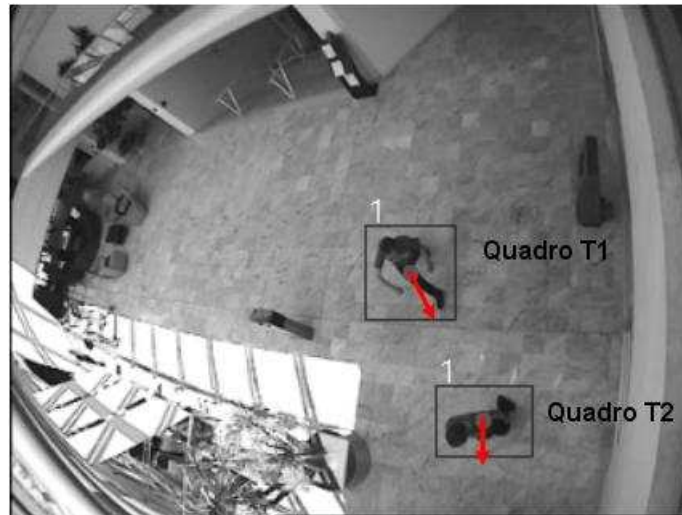


Figura 3.11: Característica direção

Supondo dois tipos de cenários: uma rodovia e um shopping center, o tempo de avaliação de uma rodovia deve ser mais curto que o de um shopping, pois os objetos em uma rodovia possuem um deslocamento mais rápido e ficam menos tempo no campo de visão. A Figura 3.12 ilustra o processo de extração de características de um vídeo e geração dos vetores de características. Dada uma sequência de vídeo que contenha um objeto em cena, a cada Q quadros são extraídos os vetores de características desse objeto, que agregados formam o conjunto V de vetores.

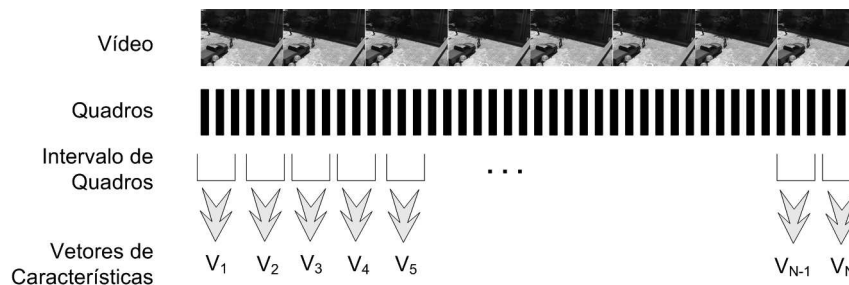


Figura 3.12: Ilustração do processo de extração e geração dos vetores de características a partir de objetos de interesse em movimento no quadros do vídeo

3.4 Classificação do Movimento

Para a utilização de algoritmos de aprendizagem baseados em instâncias na classificação de eventos, inicialmente é necessária a construção de uma base de vetores de características de referência gerados a partir da análise do movimentos de objetos de in-

teresse nos quadros do vídeo. A cada vetor deve ser atribuído um rótulo, “C” ou “NC”, indicando que o objeto de interesse representado pelo vetor de características está realizando um movimento convencional ou não-convencional em cena respectivamente. A base de vetores de referência é composta por Z vetores de características rotulados, sendo idealmente 50

A etapa de classificação consiste em rotular automaticamente um novo vetor, baseado nos vetores de referência previamente rotulados na fase de treinamento. Cada novo vetor rotulado, representa o tipo de movimento realizado pelo objeto naquele instante da trajetória. No entanto, tomar como base o rótulo de um único vetor para classificar a trajetória do objeto pode ser uma estratégia falha devido a este único vetor não representar uma sequência de ações, e sim um dado instante. Para ter-se um poder de decisão mais eficiente e descritivo da trajetória, a idéia será construir um conjunto de vetores, no qual, aplicando-se o método de votação, decidirmos qual o tipo de movimento que representa essa trajetória, convencional ou não-convencional.

Com base nisso, na etapa de classificação é utilizada uma janela temporal com o objetivo de extrair e rotular esses vetores, formar o conjunto citado e classificar trechos do movimento em tempo real. A classificação então consiste em, dado um objeto em movimento, extrair-se um conjunto de vetores \mathcal{V} que corresponde a um trecho do movimento. O número de vetores neste conjunto varia de acordo com o tamanho da janela temporal, ou seja, se J quadros é o tamanho da janela, o conjunto \mathcal{V} será composto de J/Q vetores. O processo de classificação é constituído de dois estágios: classificar individualmente cada vetor $V_i \in \mathcal{V}$ (Fig 3.13), utilizando um método baseado em instâncias, mais especificamente o classificador k vizinhos mais próximos (k -NN) (Aha et al., 1991) e posteriormente fazer uma votação entre os vetores do conjunto para classificar o movimento. O algoritmo a seguir ilustra o processo de extração de características:

MONTAconjunto (*InfoObjeto*, $Q, i, V[0 \dots J/Q][0 \dots 7]$)

```

1 // Entrada: InfoObjeto,  $V[]$ 
2 // InfoObjeto é uma estrutura contendo informações do objeto.
3 //  $Q$  é o intervalo de extração de características.
4 //  $i$  é a posição ocupada no conjunto de vetores.
5 //  $V[]$  é o conjunto de vetores de características do objeto.
6 if intervalo ==  $Q$  then
7    $V[i][0] \leftarrow \text{InfoObjeto.Posicao}.x;$ 
8    $V[i][1] \leftarrow \text{InfoObjeto.Posicao}.y;$ 
9    $V[i][2] \leftarrow \text{InfoObjeto.Velocidade};$ 
10   $V[i][3] \leftarrow \text{InfoObjeto.Deslocamento}.x;$ 

```

```

11   $V[i][4] \leftarrow InfoObjeto.Deslocamento.y;$ 
12   $V[i][5] \leftarrow InfoObjeto.VarBBox.x;$  //Variacao da Bounding Box em x
13   $V[i][6] \leftarrow InfoObjeto.VarBBox.y;$  //Variacao da Bounding Box em y
14   $V[i][7] \leftarrow InfoObjeto.Direcao;$ 
15  if  $i == J/Q$  then
16     $Classifica(v[i]);$  //Será detalhada mais a frente
17  else
18     $i \leftarrow i + 1;$ 
19  endif
20 endif

```

No primeiro estágio é computada a distância Euclidiana de cada vetor $V_i \in \mathcal{V}$ para os Z vetores de referência da base de vetores. Os k vetores de referência da base mais próximos irão rotular cada vetor $V_i \in \mathcal{V}$. A distância Euclidiana entre um vetor D -dimensional de referência V_z e um vetor V_i é definida como:

$$d(V_z, V_i) = \sqrt{\sum_{d=1}^D (V_{zd} - V_{id})^2} \quad (3.16)$$

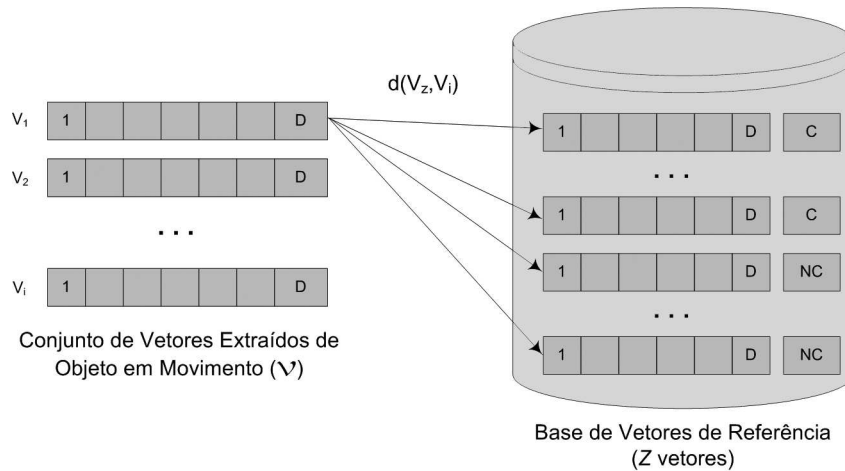


Figura 3.13: Exemplo do processo de classificação dos vetores de características relativos ao objeto de interesse.

Uma vez classificado cada vetor $V_i \in \mathcal{V}$, a decisão final sobre o movimento do objeto será dada pela votação entre as “instâncias” do conjunto, sendo atribuído a classificação convencional ou não convencional de acordo com a maioria (*majority vote*). Por exemplo, se existirem em \mathcal{V} sete vetores classificados pelo k -NN como não-convencional (NC) e dois como convencional (C), a decisão será atribuir a classificação “não convencional” ao movimento do objeto de interesse. Abaixo é ilustrado o algoritmo de classificação do movimento:

```

CLASSIFICA (Vetores[0...J/Q][0...7], K Vizinhos, Base Vetores)
1 // Entrada: V[], K Vizinhos, Base Vetores
2 // Saida: Movimento "C" ou "NC"
3 // InfoObjeto é uma estrutura contendo informações do objeto.
4 // V[] é o conjunto de vetores de características do objeto.
5 for i ← 0 until i ≤ J/Q do
6   rotulo ← knn(v[i], base, k)
7   if rotulo == 'C' then
8     Convencional ++;
9   else
10    NConvencional ++;
11  endif
12 endfor
13 if 'C' ≥ 'NC' then
14  return 'C'
15 else
16  return 'NC'
17 endif

```

3.5 Normalização dos vetores

A normalização é utilizada para restringirmos a variação (*range*) dos valores das características, e por consequência obtermos a mesma unidade de avaliação entre os parâmetros. Em nossa proposta utilizamos um método de normalização linear, na qual a variação dos valores ficam entre -1 e 1 . A fórmula de normalização é dada como segue:

$$VNorm = 1 - \frac{MAX - VAL}{\frac{MAX - MIN}{2}} \quad (3.17)$$

na qual $VNorm$ é o valor normalizado, MAX e MIN são respectivamente os valores máximos e mínimos de uma característica e VAL é o valor não normalizado da característica.

O processo de normalização foi realizado individualmente para cada característica, ou seja, uma normalização dos valores de *Vel*, outra para *Dir* e assim por diante. No processo de normalização dos vetores de testes, além de fazermos uma normalização individual, os valores MAX e MIN das características foram utilizados da base de treinamento, uma vez que os vetores de testes devem ser normalizados com base nos vetores de treinamento.

3.6 Conclusão

Neste capítulo apresentamos o método proposto. Podemos ver que foi desenvolvido soluções para as demais etapas, embora que o foco deste trabalho seja a classificação do movimento, apresentando assim uma solução completa. Focando na questão da adaptabilidade, o que é um diferencial do método proposto, as características utilizadas tornam a proposta independente do tipo de objeto e do ambiente, pois são obtidas analisando o movimento. No próximo capítulo apresentaremos os resultados obtidos.

Capítulo 4

Resultados Experimentais

O objetivo de nossos experimentos é validar as hipóteses propostas anteriormente. A primeira hipótese diz respeito a questão da adaptabilidade do método. Para isso, escolhemos duas bases distintas com movimentos distintos. Em uma das bases temos um ambiente aberto, onde foram simulados os eventos andar, correr e zig-zag. Na outra base, temos um ambiente um pouco mais controlado, pois trata-se de um *hall* de entrada, onde a variabilidade de objetos é menor e a variabilidade de movimentação também. A idéia é avaliar o desempenho do método proposto nos dois ambientes para testarmos a adaptabilidade. Não existe um índice que permita mensurar o quão adaptável é a solução, no entanto, será considerado se a proposta é adaptável ou não, se apresentar uma taxa de acerto razoável em ambos os ambientes.

A segunda hipótese a ser testada é se existe um único conjunto de características que seja mais adequado para a tarefa de classificação de eventos. Por tratar-se de um espaço de busca relativamente pequeno, utilizamos o método de força bruta para avaliar o melhor conjunto de características. Foram formados conjuntos baseados nas combinações das 5 características entre si, o que resulta em trinta e duas combinações, ou conjuntos de características, diferentes. Em ambas as bases serão testados cada um dos conjuntos de características. O conjunto de características que melhor taxa de acerto apresentar será o escolhido como o conjunto ideal. Também testamos cinco valores de vizinhos para o k : 3,5,10,15,20, com cada uma das combinações de características.

Para reduzirmos o processamento, a resolução utilizada foi de 320x240 pixels em níveis de cinza, em todos os vídeos. A taxa de quadros por segundo foi mantida conforme a versão original do vídeo. Os testes foram realizados em um máquina AMD Athlon (tm) 64 Bits 3000+, com 1 GB de memória ram, e disco de 120 GB 7200 rpm sata. Todos os testes apresentaram uma resposta em tempo real, o que era uma das propostas do nosso método.

Nas seções a seguir, detalharemos as bases utilizadas e faremos uma análise dos resultados.

4.1 Base de Vídeos

4.1.1 Base Estacionamento

Nesta base¹ os vídeos foram capturados em um estacionamento através de uma câmera de segurança instalada em um prédio vizinho sem qualquer controle de iluminação. Essa base foi criada pelos autores devido a escassez de vídeos com o intuito de detecção de movimentos não convencionais. A resolução original é de 720 x 480 pixels à uma taxa de 30 quadros por segundo, comprimidos utilizando MPEG2. Foram capturados três tipos de movimento: Andar, Correr e Zig-Zag. Os eventos foram simulados pelos participantes do vídeo, portanto não foi um movimento espontâneo.

Para cada tipo de movimento foram colhidos dois vídeos, dos quais um foi usado para teste e outro para treinamento. Apesar de três tipos de movimentos estarem representados nos vídeos, consideramos um problema de duas classes onde “andar” é considerado como um movimento convencional e os movimentos “zig-zag” e “correr” são considerados como eventos não convencionais. Um maior detalhamento dos vídeos é apresentado na Tabela 4.1.

O processo de rotulação foi feito por evento. Para cada evento rotulado, os vetores gerados desse evento eram rotulados com a mesma classe. No entanto foi necessária uma rotulação temporal, pois em certos períodos da trajetória do movimento, o mesmo pode ser considerado convencional ou não-convencional. Por exemplo, dado um objeto em movimento, inicialmente este objeto pode apresentar um movimento de andar (convencional) e depois de um certo período um movimento de corrida (não-convencional), ou vice-versa, ao longo de sua trajetória. Logo, para cada tipo de evento apresentado na trajetória do objeto, a rotulação deve ser feita com a respectiva classe.

Para facilitar a rotulação, por ser esta base de nossa autoria, dividimos os vídeos pelos eventos. Dessa forma sabemos que em um determinado vídeo apenas uma classe de evento está representada.

¹<http://www.ppgia.pucpr.br/pesquisa/imagem/basePPGIA/>

Tabela 4.1: Detalhes dos vídeos da base estacionamento

<i>Video</i>	<i>Etapa</i>	<i>Duracao Total(Seg)</i>	<i>Eventos</i>	<i>Classe</i>
Andar1.mpg	Treinamento	100	94	C
Correndo1.mpg	Treinamento	83	77	NC
ZigZag1.mpg	Treinamento	104	62	NC
Andar2.mpg	Teste	100	112	C
Correndo2.mpg	Teste	83	50	NC
ZigZag2.mpg	Teste	104	31	NC

4.1.2 Base Caviar

A base CAVIAR² é uma base disponível na internet, mantida e fornecida pelo instituto INRIA³. Os vídeos foram filmados com uma câmera que utiliza lentes de “ampla abertura”, com uma resolução de 384x288 pixels a uma taxa de 25 quadros por segundo e compressão MPEG2. Os detalhes dos vídeos utilizados na etapa de treinamento e na etapa de testes, podem ser visualizados na Tabela 4.2. Baseado em nosso contexto de trabalho, os vídeos contêm dois tipos de movimento: Andar (Convencional) e Briga (Não-Convencional). Ambos os eventos foram simulados, analogamente a base estacionamento.

O processo de rotulação seguiu o mesmo padrão da base Estacionamento. No entanto, nesses vídeos haviam eventos misturados, ou seja, classes distintas em uma mesma cena, o que nos rendeu um pouco mais de trabalho pois o processo manual de verificação é mais trabalhoso.

Tabela 4.2: Detalhes dos vídeos da base caviar

Vídeo	Etapa	Duração (Seg)	Eventos	
			C	NC
Walk2	Treinamento	45,4	26	00
FightOneManDown	Treinamento	38,5	06	30
FightRunAway2	Treinamento	22,2	25	11
Walk1	Teste	24	21	00
FightChase	Teste	17,4	00	06
FighrRunAway1	Teste	22,2	02	10

4.2 Ajuste de Parâmetros

Antes de iniciarmos os testes, foi necessário ajustar parâmetros do método proposto, citados na Seção 3.2. Os valores foram definidos de forma empírica, utilizando

²<http://homepages.inf.ed.ac.uk/rbf/CAVIAR/>

³<http://www.inria.fr/index.en.html>

um vídeo de exemplo de cada ambiente, os quais não foram utilizados nos experimentos. Em cada testes os valores eram modificados e anotados os resultados até obtermos uma configuração ideal. Uma configuração é dita ideal quando não se tem perda do rastro dos objetos no decorrer da cena, ou uma perda mínima. Verificamos que para ambos os ambientes, os valores dos parâmetros poderiam ser mantidos os mesmos. O limiar de distância máxima, utilizado por dP , foi setado em 40. Isto significa que um mesmo objeto não se movimenta mais que 40 pixels entre um quadro e outro. Para TTL_{MAX} utilizamos o valor 45. Para os demais foram definidos valores conforme sugere o autor do artigo (Latecki and Mieziako, 2006). S foi mantido em 0.9 e os valores dos pesos wP , wS , wD em 0.4, 0.1 e 0.5 respectivamente.

4.3 Resultados e Análise

Nesta seção, analisaremos os resultados obtidos nos testes. Em razão de cada base utilizada conter tipos diferentes de movimentos, preferimos fazer uma análise em separado das duas bases. Todos os resultados obtidos com os conjuntos de características para ambas as bases, podem ser encontrados no Apêndice A ao final deste documento.

4.3.1 Base Estacionamento

Os vídeos de treinamento citados na Tabela 4.1 foram utilizados para construir a base de vetores de referência. Os vídeos de testes foram utilizados para fazer a validação. Cada vídeo foi processado separadamente para extraír os vetores de características e seus respectivos conjuntos de vetores de características. Cada conjunto representa um tipo de evento. O método proposto de extração de características foi aplicado em ambos os vídeos, apenas a rotulação dos mesmos foi de forma manual. Com base na rotulação manual dos eventos e comparando com a posterior rotulação automática obtemos a acuracidade do método.

Neste teste obtemos uma taxa de acerto de 78.76% com as características Vel e Dir utilizando 10 vizinhos. Por outro lado, a pior taxa de acerto foi de 55.96% com 3 vizinhos e característica $Bbox$ sendo escolhida como a característica menos discriminativa. A matriz de confusão para a melhor combinação de características pode ser vista na Tabela 4.3.

Conforme detalhado acima, esta base contém os movimentos de Andar, Correr e Zig-Zag, conforme ilustra as Figuras 4.1, 4.2 e 4.3. Analisando as características dos movimentos, os três apresentam de um certo modo características bem similares como postura, deslocamento e velocidade constantes. No entanto, o que diferencia estes três

Tabela 4.3: Matriz de confusão para o conjunto de características *Vel* e *Dir* com 10 Vizinhos.

Movimento	Convencional Andando	Não-Convencional	
		Correndo	Zig-Zag
Andando	96	06	10
Correndo	03	28	00
Zig-Zag	22	13	15

movimentos, são características únicas de cada movimento. O movimento correr apresenta um valor nominal de velocidade e deslocamento superior que os movimentos andar e zig-zag. Já o movimento zig-zag apresenta uma variação de direção que foge dos padrões do movimento andar e correr, o qual apresentam direções quase constantes. Utilizando a características velocidade e variação de direção, o classificador foi capaz de distinguir os três movimentos na cena, ignorando os valores de postura (tamanho da *Bounding Box*), deslocamento e posição. Nota-se uma elevada confusão entre os movimentos Zig-Zag e Andar, a discussão desse problema encontra-se na análise de erros, ao final deste capítulo.

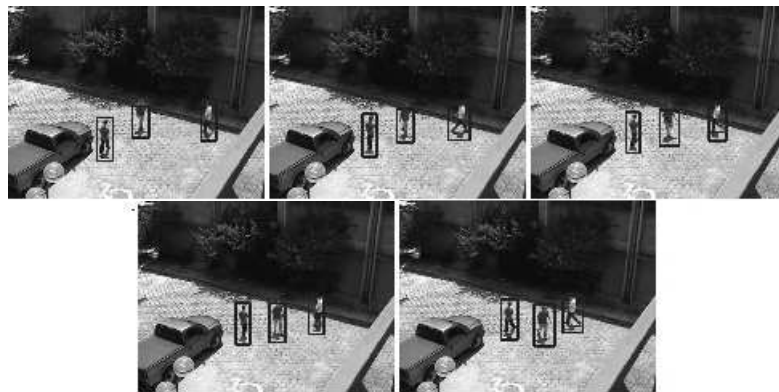


Figura 4.1: Sequência de quadros do movimento “Andar” da base Estacionamento

4.3.2 Base Caviar

Analogamente a base Estacionamento, os eventos apresentados nos vídeos de treinamento (ver Tabela 4.2) compõem a base de vetores de referência. Os vetores e os conjuntos de vetores dos eventos dos vídeos de testes foram utilizados na validação. Para ambos os vídeos, a extração dos vetores e dos conjuntos de vetores foram feitos através do método proposto, apenas a rotulação foi feita de modo manual. Através da rotulação manual dos eventos e da rotulação automática obtida posteriormente pelo método proposto, obtemos a acuracidade do método.

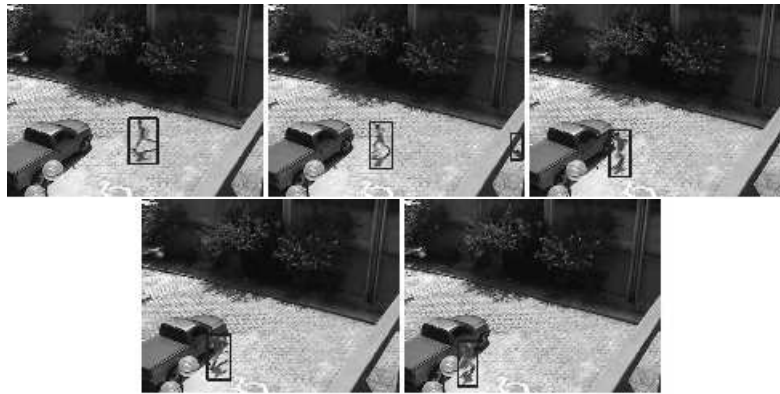


Figura 4.2: Sequência de quadros do movimento “Correr” da base Estacionamento

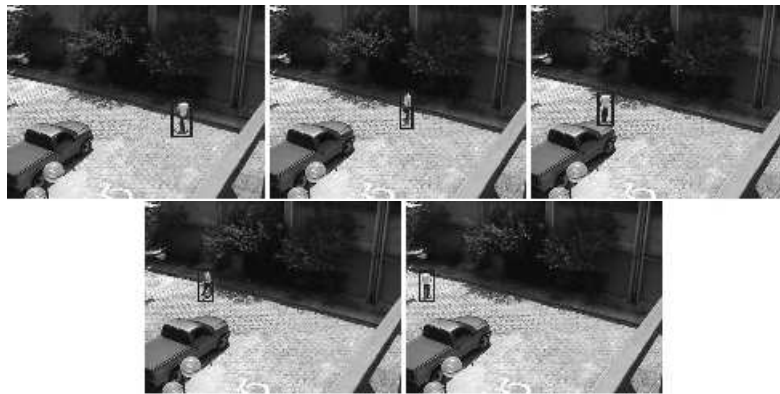


Figura 4.3: Sequência de quadros do movimento “ZigZag” da base Estacionamento

Na Base Caviar obtivemos um acerto de 82.05% utilizando *Vel* e *Pos* como características com 5 vizinhos. No pior caso obtemos 51.28% com *BBox* e *Dir* utilizando 5 vizinhos. A matriz de confusão pode ser vista na tabela 4.4.

Tabela 4.4: Matriz de confusão para o conjunto de características *Vel* e *Pos* com 5 Vizinhos.

Movimento	Convencional Andando	Não-Convencional Brigando
Andando	20	03
Brigando	04	12

Nesta base estão representados os movimentos Andar (convencional) e briga (não-convencional). Analogamente ao processo utilizado na Base Estacionamento, podemos analisar as características dos movimentos desta base. O movimento andar, é caracterizado por velocidade praticamente constantes. Já o movimento briga, é constituído por uma velocidade quase zero. Verificando os resultados, as características escolhidas pelo classificador foram posição e velocidade .

Podemos verificar que com a características velocidade, é possível distinguir o evento andar e o movimento briga. No entanto foi escolhido também a característica posição. A característica posição, foi levada em consideração devido ao fato do lugar onde os eventos ocorreram. Os exemplos de briga, foram todos encenados em regiões similares da cena, enquanto o movimento andar, foi encenado em regiões distintas do movimento briga. Uma breve sequência de quadros de cada evento pode ser visto nas Figuras 4.4 e 4.5.



Figura 4.4: Sequência de quadros do movimento “Andar” da base Caviar

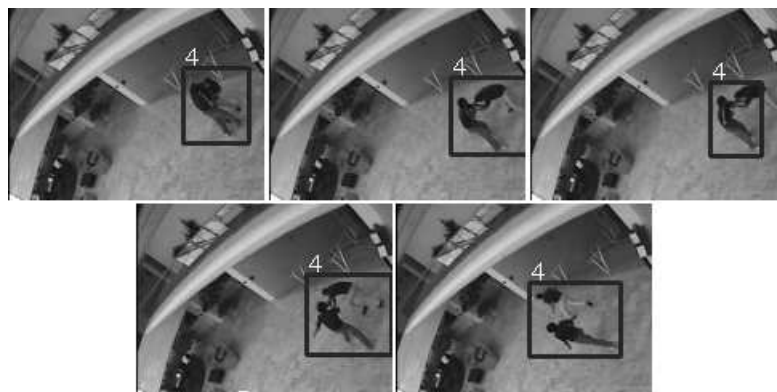


Figura 4.5: Sequência de quadros do movimento “Briga” da base Caviar

4.3.3 Análise de Erros

Na base estacionamento, notamos uma confusão entre os eventos Zig-Zag e Andar. Isso ocorreu devido em alguns momentos as características destes movimentos apresentar uma grande similaridade. Certos momentos da trajetória do zig-zag apresentam características do movimento andar, ou seja, pouca variação de direção, velocidade, deslocamento e *bounding box*. Como utilizamos os vetores de características como uma informação

isolada, alguns vetores do zig-zag são muito similares ao andar. Se, futuramente, utilizarmos alguma informação sobre o que antecedeu o movimento alguns quadros atrás, ou vetores anteriores, poderíamos usar esta informação para diferenciar estes movimentos.

Na base CAVIAR, a matriz de confusão mostrou-se mais estável entre os dois eventos. O problema é que devido o pouco número de eventos gerados, um erro representa um percentual elevado, o que ocasiona uma taxa de erro similar a base estacionamento. Também devido ao pouco número de exemplos, a característica posição foi escolhida como uma característica discriminativa, quando na verdade apenas reflete que naquela posição da cena ocorreram os eventos e por isso seria interessante avaliar a posição. Não vemos isso como um erro do classificador, pois na verdade, para esses exemplos realmente a posição interferiu. Se em um ambiente tivemos movimentos não-convencionais que ocorrem sempre no mesmo lugar, a característica posição será, como neste caso, discriminativo. No entanto uma maior variedade de simulações de briga em lugares distintos da cena, levará ao uso de outras características, uma vez que variando a posição ela não será mais discriminativa, e sim características como deslocamento ou variação de *bounding box*.

4.4 Conclusão

Neste capítulo apresentamos os resultados dos nossos testes. Acreditamos que para uma primeira avaliação do algoritmo os resultados foram promissores. Discutiremos mais sobre os testes no próximo capítulo.

Capítulo 5

Conclusão

Neste documento apresentamos uma solução baseada em instâncias para detecção de eventos não-convencionais. Diferentemente de abordagens que tratam apenas a detecção de eventos como um caso isolado, desenvolvemos um aplicação completa, envolvendo as três principais etapas: detecção e segmentação de movimento, rastreamento e classificação. Utilizando ambientes reais como cenário de aplicação e testes, sem qualquer tipo de controle de iluminação, variação do fundo ou demais fatores que significativamente influenciam no desempenho de um sistema deste tipo, através de um classificador k -NN, obtivemos taxas de acerto entre 78% e 82% conforme o caso. As taxas de correta classificação do movimento obtidas foram inferiores as encontradas em trabalhos similares (Niu et al., 2004; Wu et al., 2005; Niu and Abdel-Mottaleb, 2004; Cucchiara et al., 2005b), no entanto essa comparação torna-se um pouco relativa, pois alguns trabalhos foram testados em ambientes controlados, com segmentação do fundo controlada (objeto branco em um fundo escuro, por exemplo), movimentos específicos ou o método foi desenvolvido e testado para um ambientes específico. Também muitas vezes somente a etapa de classificação do movimento foi desenvolvida. No entanto, para uma primeira avaliação do nosso método, concluímos que a metodologia proposta é válida, e que os resultados obtidos são altamente satisfatórios. Comparando nossos resultados com trabalhos relacionados (Niu et al., 2004; Wu et al., 2005; Niu and Abdel-Mottaleb, 2004; Cucchiara et al., 2005b), vemos algumas vantagens e desvantagens do nosso método. Uma vantagem é que a solução não está restrita a um ambiente específico, posição da câmera, tamanho do objeto, etc. Já uma desvantagem, é que um método baseado em instâncias pode se tornar computacionalmente pesado se houver um grande número de vetores de referência na base de treinamento. Outro fator que deve ser levado em conta é a necessidade de exemplos negativos, ou seja, exemplos de movimentos não convencionais.

Em nossos testes, um grande problema encontrado foi a disparidade entre o número

de exemplos encontrados de movimentos convencionais e de não-convencionais. O reflexo desta disparidade acarreta em um maior aprendizado do algoritmo em eventos convencionais do que não-convencionais, quando a proposta do sistema é inversa. No entanto, essa é a realidade que temos. Dificilmente encontraremos um ambiente onde o número de eventos não-convencionais seja no mínimo igual ao de eventos convencionais. No entanto, mesmo com esta disparidade avaliamos como positivo o resultado da proposta, indicando que através de simples características pode-se obter boas taxas de acerto e dentro do contexto de tempo real de processamento. Esse será um ponto a ser avaliado em nossos futuros trabalhos.

Com relação ao cumprimento dos objetivos propostos, na questão da adaptabilidade, conseguimos provar que aplicando a mesma solução em ambientes distintos, obtemos uma taxa de acerto similar, sem grandes esforços. Já na questão de avaliar o melhor conjunto de características para detecção de eventos, concluímos que não existe um conjunto que seja comum aos dois ambientes testados. Devido a variabilidade entre os eventos existentes em cada ambiente, com características diferenciadas para cada evento, não é possível definir um único conjunto de características comum a todos os ambientes. No entanto concluímos que a seleção das características feita pelo classificador foi correta, pois foram escolhidas as características mais discriminativas baseadas nos tipos de eventos que estavam sendo analisadas. Também, a classificação do movimento apresentou-se em tempo real, cumprindo um terceiro objetivo do método

5.1 Contribuições

Nosso trabalho contribuiu no sentido de desenvolvermos um método adaptativo a ambientes reais, enfatizando que este deve ser um ponto a ser levado em consideração nos trabalhos de detecção de eventos. Não encontramos na literatura, nenhum trabalho que avalia o desempenho do método na questão da adaptabilidade, muitos levam em consideração apenas a taxa de acerto e alguns consideram também o tempo de resposta como forma de avaliação. Avaliamos também o desempenho de várias características na tarefa de classificação de eventos. Este é um pontos importante, pois em trabalhos futuros já poderemos pré-definir quais características são mais discriminativas que outras em determinado tipo de evento.

Em âmbito social, nossa contribuição é no desenvolvimento de uma ferramenta de auxílio no combate a violência, prevenção de acidentes e demais situações de risco. Através de câmeras espalhadas por regiões de uma cidade, em praças, cruzamentos de trânsitos ou rodovias, as imagens seriam capturadas e processadas. Caso movimentos não-

convencionais tais como brigas ou conversões proibidas de automóveis fossem detectadas, haveria a emissão de um alerta ao operador que tomaria providências mais rapidamente. Também acreditamos que o nosso trabalho seja uma motivação para pesquisas futuras nesta área, ajudando no desenvolvimento dessa tecnologia em nosso país.

5.2 Trabalhos Futuros

Embora os resultados os resultados obtidos sejam considerados satisfatórios, a primeira versão da proposta oferece duas frentes que podem ser melhoradas: o rastreamento e a forma de classificação. Um aprofundamento nessas etapas implicará numa versão mais estável e eficiente.

A solução de rastreamento se mostrou razoável, porém a oclusão de objetos ainda é um campo não resolvido. Nossa solução é restrita a ambientes com pouco fluxo de objetos. Deste modo, uma solução que apresente robustez em um ambiente movimentado seria de grande importância para o aprimoramento do método.

Tomando como base a escassez de eventos não-convencionais, utilizaremos um paradigma baseado em *one-class SVMs*. Basicamente o sistema seria treinado apenas com eventos convencionais, devido a facilidade de exemplos, e um evento não-convencional seria detectado quando não fosse encontrado similaridade com eventos convencionais pré-treinados. Também existe a possibilidade de implementarmos um método de aprendizagem incremental, o qual o sistema vai aprimorando no decorrer do funcionamento.

Nos testes, como proposta, gerar mais vetores de referência dos movimentos e uma maior variabilidade dos movimentos no campo da cena. Até o momento a proposta foi testada apenas em dois ambientes distintos para validar a idéia de adaptabilidade. Nos próximos testes, pretendemos incluir não somente mais ambientes, como também diferentes objetos. Detectar eventos em uma rodovia, onde um movimento não convencional seria um carro dando a ré, cruzando a rodovia ou estacionado no acostamento, poderá ser bastante interessante.

Referências Bibliográficas

- Aggarwal, J. K., Cai, Q., 1999. Human motion analysis: A review. *Computer Vision and Image Understanding: CVIU* 73 (3), 428–440.
- Aha, D. W., Kibler, D., Albert, M. K., 1991. Instance-based learning algorithms. *Machine Learning* 6 (1), 37–66.
- Brand, M., Kettner, V., 2000. Discovery and segmentation of activities in video. *IEEE Trans. Pattern Anal. Mach. Intell.* 22 (8), 844–851.
- Cucchiara, R., Grana, C., Prati, A., Vezzani, R., 2005a. Computer vision system for in-house video surveillance. *IEEE Proc.-Vis. Image Signal Process.* 152 (2), 242–249.
- Cucchiara, R., Grana, C., Prati, A., Vezzani, R., 2005b. Probabilistic posture classification for human-behavior analysis. *IEEE Transactions on Systems, Man, and Cybernetics, Part A* 35 (1), 42–54.
- Duda, R., Hart, P., Stork, D., 2001. *Pattern Classification*. Wiley.
- Gao, D., Zhou, J., 2001. Adaptive background estimation for real-time traffic monitoring. In: *IEEE Intelligent Transportation System Conference Proceedings*. Oakland, USA, pp. 330–333.
- Gavrila, D. M., Jan. 1999. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding* 73 (1), 82–98.
- Hara, K., Omori, T., Ueno, R., 2002. Detection of unusual human behavior in intelligent house. In: *IEEE Workshop on Neural Networks for Signal Processing*. Martigny, Switzerland, pp. 697–706.
- Hochuli, A. G., de Oliveira, L. E. S., de Souza Britto, A., Koerich, A. L., 2007a. Detection and classification of human movements in video scenes. In: *PSIVT2007: 2007 IEEE Pacific Rim Symposium on Image Video and Technology*. Santiago, Chile, pp. 678–691.

- Hochuli, A. G., Koerich, A. L., de Souza Britto, A., 2007b. Detection of non-conventional events on video scene. In: SMC2007: 2007 IEEE International Conference on Systems, Man, and Cybernetics. IEEE Computer Society, Montreal, Canada, pp. 302–307.
- Hu, W., Tan, T., Wang, L., Maybank, S. J., Aug. 2004. A survey on visual surveillance of object motion and behaviors. SMC-C 34 (3), 334–352.
- Latecki, L. J., Miezianko, R., 2006. Object tracking with dynamic template update and occlusion detection. In: 18th Intl Conf on Pattern Recognition. Washington, USA, pp. 556–560.
- Lei, B., Xu, L.-Q., 2005. From pixels to objects and trajectories: A generic real-time outdoor video surveillance system. In: IEEE International Symposium on Imaging for Crime Detection and Prevention. London, UK, pp. 117–122.
- Lok, W., Chan, K. L., 2005. Model-based human motion analysis in monocular video. In: ICASSP2005: 2005 IEEE International Conference on Acoustics, Speech, and Signal Processing. Vol. 2. Philadelphia, USA, pp. 697–700.
- Lucas, B. D., Kanade, T., 1981. An iterative image registration technique with an application to stereo vision. In: 7th Intl Joint Conf Artificial Intelligence. Vancouver, Canada, pp. 674–679.
- Lühr, S., Bui, H. H., Venkatesh, S., West, G. A. W., 2003. Recognition of human activity through hierarchical stochastic learning. In: IEEE Annual Conference on Pervasive Computing and Communications. Fort Worth, USA, pp. 416–421.
- Marchesotti, L., Marcenaro, L., Regazzoni, C. S., 2002. A video surveillance architecture for alarm generation and video sequences retrieval. In: IEEE International Conference on Image Processing. New York, USA, pp. 892–895.
- Mecocci, A., Pannozzo, M., Fumarola, A., 2003. Automatic detection of anomalous behavioral events for advanced real-time video surveillance. In: IEEE International Symposium on Computational Intelligence for Measurement Systems and Applications. Lugano, Switzerland, pp. 187–192.
- Niu, F., Abdel-Mottaleb, M., 2004. View-invariant human activity recognition based on shape and motion features. In: ISMSE '04: Proceedings of the IEEE Sixth International Symposium on Multimedia Software Engineering (ISMSE'04). IEEE Computer Society, Washington, DC, USA, pp. 546–556.

- Niu, W., Long, J., Han, D., Wang, Y.-F., 2004. Human activity detection and recognition for video surveillance. In: IEEE International Conference on Multimedia and Expo. Taipei, Taiwan, pp. 719–722.
- Polana, R., Nelson, R., 1994. Low level recognition of human motion (or how to get your man without finding his body parts. In: Proceedings of the IEEE Workshop on Motion of Non-Rigid and Articulated Objects. Austin, TX, USA, pp. 77–82.
- Ridder, C., Munkelt, O., Kirchner, H., 1995. Adaptive background estimation and foreground detection using kalman filtering. ICAM: IEEE International Conference on Agile Manufacturing, 193–199.
- Stauffer, C., Grimson, W. E. L., 2000. Learning patterns of activity using real-time tracking. IEEE Trans. Pattern Anal. Mach. Intell. 22 (8), 747–757.
- Wang, L., 2006. From blob metrics to posture classification to activity profiling. In: ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition. IEEE Computer Society, Washington, DC, USA, pp. 736–739.
- Wu, X., Ou, Y., Qian, H., Xu, Y., 2005. A detection system for human abnormal behavior. In: IROS 2005: 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE Computer Society, Washington, DC, USA, pp. 1204–1208.
- Yilmaz, A., Javed, O., Shah, M., 2006. Object tracking: A survey. ACM Comput. Surv. 38 (4), 13.
- Zhu, G., Xu, C., Huang, Q., Gao, W., 2006. Action recognition in broadcast tennis video. In: ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition. Hong Kong, China, pp. 251–254.

Apêndice A

Taxas de Acerto

A seguir apresentaremos as tabelas contendo as taxas de acerto para todos os conjuntos de características testados. A tabela A.1 apresenta os resultados dos testes realizados na base Estacionamento. Os testes da base Caviar são apresentados na tabela A.2.

Tabela A.1: Taxas de Acerto para os conjuntos da Base Estacionamento. O valor “1” representa quais características estavam sendo utilizadas no conjunto

Características					k-Vizinhos				
<i>Vel</i>	<i>Pos</i>	<i>Desl</i>	<i>BBox</i>	<i>Dir</i>	3	5	10	15	20
0	0	0	0	1	64.77	63.21	72.02	62.69	65.80
0	0	0	1	0	55.96	58.55	63.21	61.66	64.25
0	0	0	1	1	63.21	63.21	61.66	61.66	62.18
0	0	1	0	0	67.36	71.50	74.09	69.95	70.98
0	0	1	0	1	65.28	64.77	63.73	64.25	66.84
0	0	1	1	0	64.25	66.32	58.55	61.14	59.59
0	0	1	1	1	60.62	63.73	60.10	62.18	56.99
0	1	0	0	0	63.21	63.21	64.25	63.21	65.28
0	1	0	0	1	59.07	61.66	58.55	60.62	60.62
0	1	0	1	0	64.77	69.95	68.39	67.36	73.58
0	1	0	1	1	63.21	62.18	63.21	61.66	65.80
0	1	1	0	0	64.25	61.14	63.21	63.21	65.28
0	1	1	0	1	62.18	65.28	61.14	59.59	62.69
0	1	1	1	0	67.88	64.77	63.21	67.36	66.32
0	1	1	1	1	58.55	61.66	62.69	60.10	63.73
1	0	0	0	0	63.21	63.73	68.39	65.80	70.98
1	0	0	0	1	72.54	70.98	78.76	76.68	70.98
1	0	0	1	0	67.36	66.84	69.95	71.50	69.95
1	0	0	1	1	66.84	66.32	65.80	67.88	64.25
1	0	1	0	0	66.32	67.88	72.02	72.02	72.54
1	0	1	0	1	72.54	66.32	63.21	62.69	62.69
1	0	1	1	0	65.80	64.77	62.69	60.10	57.51
1	0	1	1	1	63.73	61.14	60.62	61.66	61.14
1	1	0	0	0	65.28	66.32	69.95	71.50	69.43
1	1	0	0	1	62.18	60.10	67.36	67.36	70.98
1	1	0	1	0	69.43	73.58	71.50	73.58	76.17
1	1	0	1	1	69.95	72.54	71.50	69.43	67.88
1	1	1	0	0	59.59	65.80	66.32	63.73	64.25
1	1	1	0	1	58.55	59.07	57.51	61.66	62.69
1	1	1	1	0	64.77	68.91	64.77	69.43	70.47
1	1	1	1	1	64.77	67.36	64.77	66.84	66.32

Tabela A.2: Taxas de Acerto para os conjuntos da Base Caviar. O valor “1” representa quais características estavam sendo utilizadas no conjunto

Características					k-Vizinhos				
<i>Vel</i>	<i>Pos</i>	<i>Desl</i>	<i>BBox</i>	<i>Dir</i>	3	5	10	15	20
0	0	0	0	1	64.10	61.54	61.54	56.41	58.97
0	0	0	1	0	53.85	56.41	58.97	61.54	58.97
0	0	0	1	1	51.28	51.28	56.41	56.41	58.97
0	0	1	0	0	66.67	66.67	69.23	66.67	66.67
0	0	1	0	1	74.36	64.10	58.97	64.10	61.54
0	0	1	1	0	64.10	64.10	61.54	58.97	58.97
0	0	1	1	1	64.10	71.79	58.97	61.54	58.97
0	1	0	0	0	74.36	76.92	76.92	76.92	71.79
0	1	0	0	1	71.79	74.36	66.67	61.54	61.54
0	1	0	1	0	69.23	69.23	69.23	71.79	66.67
0	1	0	1	1	66.67	58.97	61.54	58.97	53.85
0	1	1	0	0	64.10	69.23	66.67	71.79	66.67
0	1	1	0	1	69.23	74.36	69.23	58.97	53.85
0	1	1	1	0	64.10	74.36	69.23	71.79	66.67
0	1	1	1	1	66.67	71.79	64.10	61.54	61.54
1	0	0	0	0	61.54	64.10	56.41	64.10	58.97
1	0	0	0	1	66.67	66.67	58.97	58.97	58.97
1	0	0	1	0	58.97	58.97	58.97	58.97	61.54
1	0	0	1	1	61.54	58.97	53.85	53.85	58.97
1	0	1	0	0	64.10	66.67	69.23	69.23	69.23
1	0	1	0	1	64.10	58.97	58.97	61.54	56.41
1	0	1	1	0	61.54	66.67	64.10	64.10	58.97
1	0	1	1	1	56.41	64.10	58.97	64.10	56.41
1	1	0	0	0	82.05	71.79	71.79	69.23	66.67
1	1	0	0	1	76.92	71.79	66.67	61.54	58.97
1	1	0	1	0	66.67	69.23	69.23	76.92	71.79
1	1	0	1	1	66.67	66.67	66.67	66.67	64.10
1	1	1	0	0	71.79	69.23	69.23	71.79	64.10
1	1	1	0	1	66.67	66.67	64.10	64.10	56.41
1	1	1	1	0	66.67	71.79	66.67	69.23	64.10
1	1	1	1	1	64.10	66.67	61.54	76.92	64.10