UNIVERSIDADE FEDERAL DO PARANÁ

MARCIO ROBERTO MIRANDA ASSIS

IDENTIFICADOR *MULTITHREAD* DE FLUXO DE OBJETOS: ABORDAGEM POR AGRUPAMENTO DE VETORES DE MOVIMENTO E MODELAGEM DE *BACKGROUND*.

CURITIBA

2009

MARCIO ROBERTO MIRANDA ASSIS

IDENTIFICADOR *MULTITHREAD* DE FLUXO DE OBJETOS: ABORDAGEM POR AGRUPAMENTO DE VETORES DE MOVIMENTO E MODELAGEM DE *BACKGROUND*.

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, Universidade Federal do Paraná.

Orientador: Prof. Dr. Daniel Weingaertner

Co-Orientador: Prof. Dr. Bruno Muller

CURITIBA

2009

In memoriam da minha querida avó

Emiriam Correia de Miranda

RESUMO

A automatização da identificação de atividades humanas nos mais diversos ambientes, em especial na análise do comportamento humano (AH), tem-se mostrado um dos principais desafios da visão computacional. Por vários anos, pouco resultado significativo surgiu para solucionar os problemas relacionados à AH, conseqüência de uma série de fatores que se apresentaram como limitações, tanto de caráter teórico como prático. Nos últimos anos, várias destas limitações foram resolvidas, e como conseqüência retomaram-se as pesquisas em atividades humanas. Este documento propõe um estudo sobre os elementos envolvidos no processo de reconhecimento de atividades humanas, como também um método com base em cálculo do fluxo óptico, modelagem de *background* e paralelização, para identificar e classificar fluxo de objetos em ambientes internos.

Palavras-chave: atividades humanas, descritores de características, fluxo óptico, modelagem de *background*, paralelização.

ABSTRACT

The automation of identification of human activities in many different environments, particularly in analysis of human behavior, has proved a major challenge in computational vision. For several years, little significant results emerged to address the problems related to human activities, result of a number of factors that were presented as limitations, both theoretical and practical in nature. In the past few years several of these limitations were solved, thus the research in human activities was restored. This document proposes a study about elements involved in the recognition of human activities, but also a method based on calculation of optical flow, modeling of background and parallelization, to identify and classify stream of objects in indoor environments.

Keywords: features descriptors, background model, human activities, optical flow, parallelization.

LISTA DE FIGURAS

2.1 *Arquitetura de um sistema bottom-up padrão: i) são extraídas as características básicas, ii) as características são representadas por descritores, iii) os descritores são agrupados, aumentado assim seus níveis semânticos, e por fim a AH é identificada.* 6

2.2 *Arquitetura de um sistema top-down padrão: na fase de treino, a partir de um modelo (i) parâmetros são coletados de um conjunto de quadros (ii) para refinar o próprio modelo. Na execução, é realizado um reconhecimento de padrão (iii) para identificar o modelo previamente treinado e conseqüentemente reconhecer a AH de interesse.* 7

2.3 *Representação gráfica de um vetor de movimento. A partir de um pixel em um quadro n, tenta-se encontrar um pixel com as mesmas características em uma determinada região r no quadro $n+m$, neste caso $m = 1$ e $r = 6 \times 6$, o deslocamento de posição caracteriza o vetor de movimento.* 9

2.4 *Interpretação da direção de um vetor de movimento. Através do ângulo α, formado entre o vetor de movimento e o eixo x do quadro obtém-se a direção do deslocamento do pixel.* 10

2.5 *Agrupamento de dados e seus respectivos centróides no espaço R^2. Os dados são naturalmente agrupados pelo critérios de distância, dados próximos podem corresponder ao mesmo objeto.* 11

2.6 *AFD que reconhece a atividade humana "sentar em uma cadeira", onde "longe da cadeira" é o estado inicial, e "sentado na cadeira" representa o estado final do AFD ("sentado na cadeira" $\subseteq F$).* 12

2.7 *Classificação de paralelização respeitando o critério de componente.* 15

2.8 *Problema da abertura.* 16

4.1 *Arquitetura geral do sistema de identificação de fluxo oposto. Com exceções das etapas 5, 6 e 7, que são executadas focando a paralelização, as demais etapas são todas executadas em série.* 21

4.2 *Conversão do espaço de cor utilizado na implementação. Em (a) o quadro original no espaço de cor RGB, na ordem BGR, em (b) o quadro convertido para tons de cinza.* . . 25

4.3 *Modelo de background. É realizada uma segmentação dos elementos de foreground (áreas em branco) e background do quadro analisado (áreas em preto).* 26

4.4 *Possíveis cantos. Para cada canto também são válidas todas suas rotações.* 28

4.5 *Mapeamento de características em quadros adjacentes. Na figura apenas são destacados alguns pontos em um região do quadro principal.* 28

4.6 *Vetores de movimentos originados de um objeto em movimento (uma mulher) e de interferencias (região central do quadro).* 29

4.7 *Filtragem dos ruídos contidos em um quadro através de MB.* 30

4.8 *Vetores de movimento pertencentes ao mesmo objeto podem possuir direções divergentes. Na figura, dois vetores tem direção divergentes das demais por o braço fazer um movimento pendular.* 31

4.9 *Representação gráfica do vetor de movimento resultante. A média dos vetores movimento que compõem o agrupamento determina o vetor resultante.* 32

4.10 *Intervalo de valores de ângulos aceitos pela aproximação. Os objetos, representados pelos vetores de movimento resultantes, dentro desta faixa são caracterizados como fluxo normal. O fluxo normal é definido através do conhecimento prévio do contexto onde está sendo aplicada a técnica.* 33

4.11 *Identificação da AH de interesse. Os passos envolvidos na iteração são: 1) captura de um quadro, 2) conversão do espaço de cor do quadro original, 3) atualização do MB, 4) cálculo do fluxo óptico, 5-6) filtragem do fluxo óptico por MB, 7) obtenção dos vetores de movimento resultante, e 8) identificação da atividade humana por análise angular.* . 34

4.12 *Divisão do quadro analisado em m regiões, nesta implementação* $m = 4$. 36

4.13 *Fusão de regiões. Para melhorar a ilustração são considerados somente alguns vetores de movimento em uma sub-região do quadro completo.* 37

4.14 *Refinamento de vetores de movimento resultante.* 37

4.15 *As cinco câmeras de vigilância do aeroporto Grawitch.* 39

4.16 *Diagrama do módulo da fusão. Para a prosseguir com a fusão é necessário esperar a execução de todas as threads.* 44

LISTA DE TABELAS

4.1 *Lista de hardware utilizado no desenvolvimento, execução e testes do sistema de identificação de fluxo oposto proposto.* 22

4.2 *Lista de software utilizados no desenvolvimento, execução e testes do sistema de identificação de fluxo oposto proposto.* 23

4.3 *Descrição das configurações de testes utilizados para os testes do sistema AH proposto.* . . 40

4.4 *Resultados obtidos pela submissão de todos as configurações de testes à base de vídeo disponibilizada pela TRECVid 2009.* 41

4.5 *Porcentagem média de uso dos processadores e seus respectivos tempo de execução.* . . 43

LISTA DE ALGORITMOS

4.1 *Fusão de contexto das áreas particionadas.* 38

LISTA DE ABREVIATURAS

ACH: Análise de Comportamento Humano;

AH: Atividade Humana;

AFD: Automato Finito Deterministico;

RB: Redes de *Bayes*;

EC: Evento Composto;

FLN: *Functional Link Network*;

FN: Falso Negativo;

FNN: *FeedForward Neural Network*;

HMM: Modelo Oculto de *Markov*;

FP: Falso Positivo;

MB Modelo de *Background*;

VM Vetor de Movimento;

OWN: *Optimal Weighting Network*;

EP: Evento Primitivo;

TREC: *Text Retrivel Conference*;

TRECVid: *TREC Video Retrieval Evaluation*;

VN: Verdadeiro Negativo;

VP: Verdadeiro Positivo.

SUMÁRIO

1 INTRODUÇÃO **1**

1.1 Motivação 2

1.2 Objetivo e Contribuições 2

1.3 Organização da dissertação 3

2 REVISÃO BIBLIOGRÁFICA **4**

2.1 Abordagens de reconhecimento de atividades humanas 6

2.1.1 Abordagem *bottom-up* 6

2.1.2 Abordagem *top-down* 7

2.2 Descritores de características espaciais e temporais 8

2.2.1 Baixo nível 8

2.2.2 Nível intermediário 9

2.2.2.1 Vetor de movimento 9

2.2.2.2 Centróide 10

2.2.3 Alto nível 12

2.2.3.1 Máquinas de estados 12

2.3 Métodos Estatísticos de Apoio à Decisão 13

2.4 Paralelização 14

2.5 Fluxo Óptico 15

2.5.1 Métodos diferenciais 16

2.5.2 Métodos de correlação 17

2.5.3 Métodos de freqüência de energia 17

3 *TREC VIDEO RETRIEVAL EVALUATION* **18**

3.1 Identificação de eventos de vigilância 19

3.2 Eventos de vigilância propostos 19

3.2.1 E20 – Fluxo oposto 19

4 SISTEMA DE IDENTIFICAÇÃO DE AH PROPOSTO 20

4.1 Arquitetura do Sistema 21

4.2 Plataforma de *Hardware* e *Software* Utilizada 22

4.3 Implementação 23

4.3.1 Abertura de Vídeo e Captura de Quadros 24

4.3.2 Conversão de Espaço de Cor 24

4.3.3 Remoção de *Background* Através de Regra de *Bayes* 25

4.3.4 Identificação de Fluxo Oposto 27

4.3.4.1 Fluxo óptico 27

4.3.4.2 Obtenção dos vetores de movimento resultantes 30

4.3.4.3 Análise angular dos vetores resultantes 32

4.3.5 Paralelização 35

4.3.5.1 Particionamento da área do quadro 35

4.3.5.2 Fusão de regiões 36

4.4 Base de Testes Utilizada 39

4.4.1 Aeroporto de *Grawitch* 39

4.5 Testes e Resultados Obtidos 40

4.5.1 Precisão 40

4.5.2 Desempenho 43

5 CONCLUSÃO 45

REFERENCIAS 51

A PROJETOS RELACIONADOS 52

A.1 Ayers, D. – 2001 – *Computer Vision Lab School of Eletrical Enginieering and Computer Science* [3]. 53

A.2 Cutler, R. – 1999 – *University of Maryland, College Park e SRI International* [7]. 54

A.3 Kan, A. H. – 2004 – *Institute for Infoconm Research* [2]. 55

A.4 Yamato, J. – 1992 – *NTT Human Interface Laboratories* [41]. 56

A.5 Psarrou, A. – 2002 – *Harrow Scholl Computer Science of University of Westminster e Department of Computer Science of University of London* [26]. 57

A.6 Hongeng, S. – 2003 – *Institute for Robotics and Intelligent System, University of Southern California* [11]. 58

A.7 Lee, S. C. – 2008 – *University of Southern California* [6]. 59

A.8 Stergiou, A. – 2008 – *Athens Information Technology* [35] 60

A.9 Taj, M. – 2008 – *Queen Mary, University of London* [37] 61

A.10 Orhan, O. B. – 2008 – *University of Central Florida* [24] 62

B EVENTOS DE VIGILÂNCIA 63

B.1 E01 – Porta abrindo e fechando 63

B.2 E04 – Uso do caixa eletrônico 63

B.3 E05 – Pessoa correndo 64

B.4 E06 – Colocando o celular próximo à orelha 64

B.5 E08 – Largando um objeto 64

B.6 E09 – Pegando um objeto 65

B.7 E10 – Identificação de vestimentas 65

B.8 E11 – Sentando 65

B.9 E12 – Levantando 66

B.10 E14 – Reunião 66

B.11 E15 – Separação 66

B.12 E16 – Abraçando 67

B.13 E17 – Transferência de objeto 67

B.14 E18 – Pessoa apontando 67

B.15 E19 – Permanência à frente do elevador 68

B.16 E21 – Tirando uma foto 68

CAPÍTULO 1

INTRODUÇÃO

A automatização da identificação de atividades humanas (AH) nos mais diversos ambientes, em especial na análise do comportamento humano (ACH), tem-se mostrado um dos principais desafios da visão computacional. Vários pesquisadores [2, 3, 7, 11, 25, 26, 30, 41] dedicaram suas pesquisas para solucionar esta classe de problemas, porém tais soluções contemplam somente problemas específicos e são focados na identificação de atividades.

Por vários anos, pouco avanço significativo surgiu para solucionar os problemas relacionados à ACH, conseqüência de uma série de fatores que se apresentaram como limitações, tanto de caráter teórico como prático. Dentre elas destacaram-se a falta de conhecimento para a implementação de estruturas complexas (descritores) [10] destinadas a representar atividades humanas, a deficiência na extração e representação de informações de baixo nível das imagens ou quadros [43], e também a limitação computacional do período.

Os pontos acima conduziram o caminho das pesquisas para a extração e representação de características de baixo nível e não para a ACH [30]. Através das pesquisas, novos avanços na extração de características foram alcançados, permitindo a obtenção de informação de maior qualidade e conseqüentemente incentivando as pesquisas de características de alto nível com a capacidade de representar conteúdos mais significativos.

Estes avanços, somados à crescente capacidade computacional, foram responsáveis pela retomada das pesquisas em ACH nos últimos anos. Contudo, poucos trabalhos a tratam em sua plenitude e sim somente o reconhecimento de AH específicas. O comportamento humano pode ser interpretado como um conjunto de atividades humanas (AH) em sinergia, sendo este o foco deste estudo, mais especificamente o controle de fluxo.

1.1 Motivação

Uma das motivações para a realização deste trabalho é a quantidade vasta de possibilidades de atuação de sistemas de reconhecimento de AH, em especial a análise de fluxo de movimento, nos diversos segmentos da sociedade. Dentre os nichos de atuação estão: monitoramento e vigilância [7], logística [3], tráfego de veículos [8], etc.

Outras duas motivações são respectivamente: o custo computacional envolvido na obtenção do fluxo óptico [1, 13, 40] e do agrupamento de padrões [31], e a participação na conferência anual *TREC Video Retrieval Evaluation* (TRECVid) [33] edição 2009. Esta conferência tem como um dos objetivos destacar novas abordagens para solucionar um conjunto pré-definido de AH (especificamente eventos de vigilância).

1.2 Objetivo e Contribuições

Este trabalho tem como objetivo gerar um documento científico contendo a descrição da implementação de um sistema capaz de identificar através de uma câmera fixa, um ou mais objetos em direção contraditória ao fluxo de movimentos do ambiente analisado.

O sistema proposto pretende explorar a paralelização (através do uso de *threads*[1]) oferecida pelos processadores e sistemas operacionais atuais. Em especial os utilizados por esta aproximação[2], *Linux*[3] executado sobre plataforma *I386*.

Como contribuições este trabalho propõem: i) um método de remoção de ruídos do fluxo óptico através da remoção do *background* (seção 4.3.3), ii) o aprimoramento do algoritmo de agrupamento de padrões *k-means* (seção 4.3.4), iii) e a paralelização do processo de reconhecimento da AH de interesse, o fluxo oposto (seção 4.3.5).

[1] *Thread* é uma técnica de divisão do processo em duas ou mais tarefas que podem ser executadas paralelamente dependendo do processador utilizado.

[2] Neste trabalho o termo aproximação é usado para indicar técnica ou método.

[3] *Linux* é o termo designado aos sistemas operacionais que utilizam o *kernel* (núcleo) *Linux*.

1.3 Organização da dissertação

O capítulo 2 apresenta a revisão bibliográfica do assunto ACH, assim como conceitos fundamentais para este estudo. O que é a TRECVid, tal como seus objetivos, eventos humanos, tal como o evento de interesse deste trabalho são descritos no capítulo 3. No capítulo 4 é descrita a implementação e os testes do sistema de identificação de atividade humana proposto por este trabalho. E por fim, no capítulo 5 o trabalho é concluído.

CAPÍTULO 2

REVISÃO BIBLIOGRÁFICA

Uma imagem estática é caracterizada como uma matriz (altura e base), onde cada posição é denominada *pixel* [10]. Cada *pixel* esta associado a uma cor, pertencente a um espaço de cor [10], que descreve a informação visual de mais baixo nível contida na imagem. Existem vários espaços de cores, cada qual com suas peculiaridades e características, dentre eles estão: RGB, $Y'C_bC_r$, escala de tons de cinza, etc.

Um vídeo pode ser representado por um conjunto de imagens estáticas (quadros), ordenadas uma após a outra em um intervalo regular de tempo (segundos) para dar a ilusão de movimento. Assim, enquanto imagens estáticas contêm duas dimensões (altura e largura), vídeos contêm três dimensões (altura, largura e quadros por segundos).

Através da dimensão temporal provida pela exibição de consecutivos quadros é possível extrair conteúdos com mais informações. O rastreamento [36], por exemplo, tem como objetivo determinar a localização ou a trajetória de um objeto de interesse em um ambiente qualquer, analisando quadro a quadro. Também devido à dimensão tempo é possível interpretar algumas atividades humanas mais complexas, como por exemplo um banhista se afogando [2].

Além da dimensão temporal, a identificação de atividades humanas também é dependente do contexto ao qual está ocorrendo. O termo contexto é aqui utilizado para indicar o conhecimento prévio do ambiente onde a atividade está acontecendo, tal como a localização espacial de alguns objetos na cena. A desconsideração do contexto pode gerar interpretações ambíguas dos eventos [23, 39] devido ao fato de algumas atividades serem caracterizadas de várias formas diferentes quando submetidas a contextos diferentes.

Ayers e Shah [3] afirmam que o conhecimento prévio do ambiente possibilita o uso do contexto por parte da aproximação e também que o conhecimento prévio do ambiente influencia no custo computacional desprendido. Bobick *et al.* [15] definem contexto como sendo os limites em um espaço de conhecimento, isto é, tudo que esteja fora deste limite não é útil para resolução do problema proposto.

O principal desafio para reconhecimento de atividades complexas é traduzir as ações de interesse em descritores computacionais que modelam seu significado efetivamente [17]. Segundo Shah [30], o processo de análise de atividades humanas envolve: a extração de características, a representação e a interpretação das informações visuais. Um bom modelo de análise é um sistema que tenha um conjunto pré-definido de atividades a reconhecer e que aprenda novas atividades com o tempo.

Diversas aproximações para identificação de atividades humanas em ambientes controlados foram propostas no últimos anos: Ayers e Shah [3] propõem um sistema para detecção automática de um conjunto de atividades humanas pré-estabelecidas ocorridas em um escritório, Kan *et al.* [2] descrevem um *framework*[1] para reconhecimento de atividades efetivando-o através da identificação de crises aquáticas. Yamato *et al.* [41] realizam a identificação de uma série de movimentos de tenistas.

Mais recentemente, na conferência *TRECVid* 2008 (capítulo 3), os autores de [35] descrevem um método para identificar fluxo oposto de objetos dentro das dependências de um aeroporto da Inglaterra, tal método é apresentado no apêndice A.8. Em [6] e [37] são propostos métodos para identificar aglomerações de pessoas e entrada/saída de um elevador respectivamente. Também são apresentadas soluções de reconhecimento de diversas outras atividades [24, 37].

[1]Estrutura de suporte onde projetos podem ser organizados e desenvolvidos.

A seguir são descritos alguns conceitos fundamentais para a apresentação do processo de AH no decorrer deste trabalho. Na seção 2.1, são relatadas as abordagens de reconhecimento de atividades humanas. Os descritores de características espaciais e temporais, que representam as informações contidas nas imagens estáticas ou nos quadros, e que são imprescindíveis para realização da AH estão evidenciados na seção 2.2.

2.1 Abordagens de reconhecimento de atividades humanas

Atividades humanas são reconhecidas através da interpretação dos descritores de características espaciais e temporais (seção 2.2) extraídos de uma imagem estática ou de uma seqüência de quadros[2]. O processo de identificação das AH pode ser dividido em dois conjuntos caracterizados pelas abordagens empregadas: *bottom-up* e *top-down* [41].

2.1.1 Abordagem *bottom-up*

Nesta abordagem, inicialmente são extraídos dos quadros os descritores de baixo nível, como por exemplo o centróide [10]. Em seguida, estes são agrupados em camadas para formar descritores de mais alto nível com o objetivo de representar atividades humanas de interesse, por fim é realizada a interpretação dos descritores resultantes (figura 2.1.1).

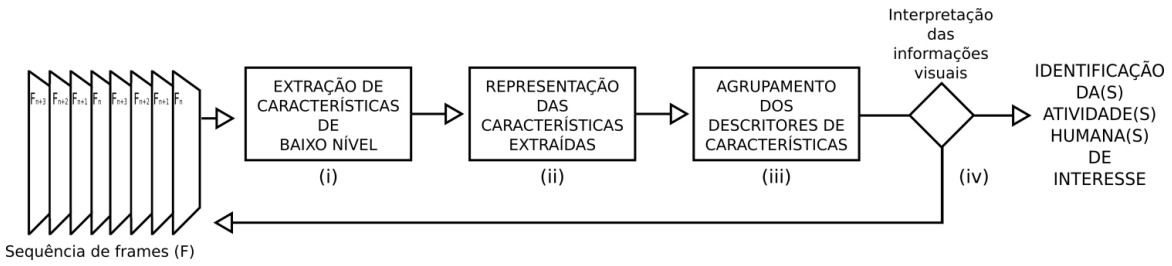

Figura 2.1: *Arquitetura de um sistema bottom-up padrão: i) são extraídas as características básicas, ii) as características são representadas por descritores, iii) os descritores são agrupados, aumentado assim seus níveis semânticos, e por fim a AH é identificada.*

[2]No restante deste documento serão considerados somente quadros e não uma imagem estática.

Por seu caráter hierárquico, uma vantagem desta abordagem é a independência entre as camadas de descritores [41], com isso descritores de camadas inferiores podem ser substituídos sem afetar as camadas superiores (seção 2.2). O caminho inverso também é valido, isto é, descritores de mais alto nível também podem ser substituídos sem afetar as camadas inferiores.

Uma conseqüência direta desta abordagem, é que a quantidade de camadas (níveis de descritores utilizados) variam de acordo com a complexidade envolvida na atividade humana que se pretende reconhecer.

2.1.2 Abordagem *top-down*

A abordagem *top-down* é caracterizada pela reconstrução da geometria do corpo através de modelos [14] (cones, caixas, cilindros, etc.). Os parâmetros dos modelos são extraídos de uma seqüência de quadros e a identificação realizada por reconhecimento de padrão [31].

Ao contrário da abordagem anterior, esta não possui camadas e são sensíveis a fatores relacionados à qualidade dos quadros, tais como: ruídos, aspecto/resolução das imagens ou quadros, entre outros [41]. Na figura 2.1.1 é exibido o diagrama de um típico sistema de reconhecimento de atividade humana *top-down*.

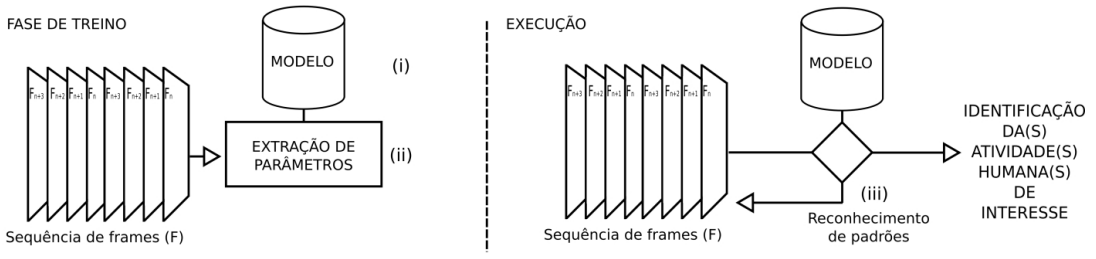

Figura 2.2: *Arquitetura de um sistema top-down padrão: na fase de treino, a partir de um modelo (i) parâmetros são coletados de um conjunto de quadros (ii) para refinar o próprio modelo. Na execução, é realizado um reconhecimento de padrão (iii) para identificar o modelo previamente treinado e conseqüentemente reconhecer a AH de interesse.*

2.2 Descritores de características espaciais e temporais

Atividades humanas podem ser representadas computacionalmente por estruturas modeladas através do tempo [26], denominadas descritores de alto nível (seção 2.2.1). Estas estruturas utilizam-se de outras estruturas denominadas (descritores de baixo nível detalhados na seção 2.2.2) como fonte de dados para sua modelagem e reconhecimento.

Os descritores de características são responsáveis por representar as informações espaciais de um quadro ou as informações temporais de um conjunto (janela) de quadros. Podemos agrupar os descritores de características em três classes disjuntas, caracterizados por seus níveis de complexidades[3] e conseqüentemente pelas informações que representam.

Este tópico está organizado como segue: na seção 2.2.1 são apresentadas com mais detalhes as características de baixo nível. A seção 2.2.2 descreve os descritores de nível intermediário que são responsáveis por fazer a integração com os descritores de baixo nível e os descritores de alto nível que são evidenciados na seção 2.2.3.

2.2.1 Baixo nível

Os descritores de baixo nível representam as informações mais primitivas, do ponto de vista de valores, e as mais relacionadas ao espaço de cor utilizado na imagem ou quadro explorado, como por exemplo bordas [10, 44], texturas [10, 21, 31] e regiões homogêneas [10, 44].

Estes descritores são extremamente sensíveis aos ruídos, provenientes dos dispositivos captadores de vídeo ou de fontes relacionadas à iluminação. A qualidade das informações representadas por estes descritores é diretamente associada ao espaço de cor utilizado, assim os descritores de baixo nível extraídos de um quadro em escala de cinza terá provavelmente menos qualidade que se o espaço de cor fosse o RGB, por exemplo.

[3]Neste trabalho complexidade está associada ao nível semântico e temporal que o descritor representa.

2.2.2 Nível intermediário

Descritores de nível intermediário são responsáveis por realizar a integração (comunicação) e caracterizar um nível semântico entre as informações representadas pelos descritores de baixo e alto nível [2, 31]. Os vetores de movimentos (seção 2.2.2.1), os centróides (seção 2.2.2.2) e as Redes de *Bayes* [22] são exemplos de descritores de nível intermediário.

2.2.2.1 Vetor de movimento

Um vetor de movimento [31] pode ser definido como a representação do deslocamento de um *pixel* através de uma seqüência de quadros, sendo que usualmente esta janela é composta por dois quadros adjacentes (figura 2.3). O processo consiste basicamente em, dado um *pixel* inicial, realizar uma busca por um *pixel* correspondente, com características similares (ex. valores do espaço de cor utilizado) em uma região r dos quadros término da janela.

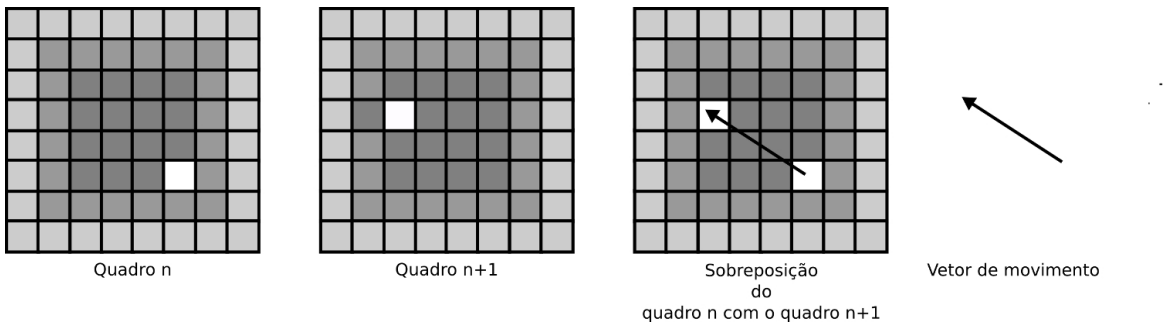

Figura 2.3: *Representação gráfica de um vetor de movimento. A partir de um pixel em um quadro n, tenta-se encontrar um pixel com as mesmas características em uma determinada região r no quadro $n + m$, neste caso $m = 1$ e $r = 6 \times 6$, o deslocamento de posição caracteriza o vetor de movimento.*

A magnitude do vetor de movimento pode ser obtida através do cálculo da distância euclidiana [10], ou através de outras distâncias [10], das coordenadas do *pixel* no quadro n, que representa o ponto inicial, e as coordenadas no quadro $n + 1$, o ponto final. A formulação matemática que expressa a distância euclidiana bidimensional entre os pontos P e Q é evidenciada na equação 2.1.

$$DE_{P_{x,y},Q_{x,y}} = \sqrt{(Q_x - P_x)^2 + (Q_y - P_y)^2} \tag{2.1}$$

É possível obter a direção do deslocamento *pixel* através da interpretação do ângulo α (figura 2.4), formado pelo vetor de movimento e o eixo x do quadro sobreposto[4] (figura 2.3). Contudo, o sistema de coordenadas considerado na obtenção do α pode coincidir ou não com o sistema de coordenadas usado pelo dispositivo de captura de vídeo.

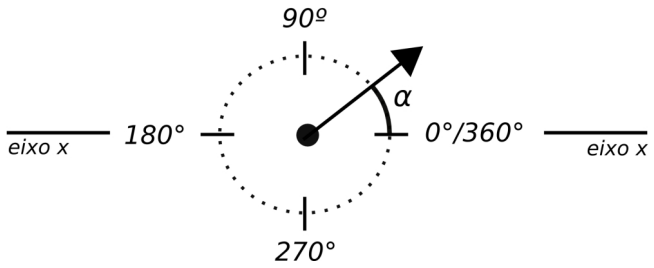

Figura 2.4: *Interpretação da direção de um vetor de movimento. Através do ângulo α, formado entre o vetor de movimento e o eixo x do quadro obtém-se a direção do deslocamento do pixel.*

Um problema inerente à obtenção do vetor de movimento é a possibilidade de encontrar dois ou mais *pixels* com características semelhantes ao *pixel* base, e com isso obter uma direção e uma magnitude que não correspondem com seus valores corretos. Outro problema é que um ruído [10] pode levar o processo a encontrar um falso deslocamento.

2.2.2.2 Centróide

O centróide pode ser definido como o centro geométrico de um corpo. Estendendo esta definição, é possível interpretar o centróide como sendo a média de um conjunto de dados aleatórios[5] plotados em um ou mais espaços (ex. $R^1, R^2, ..., R^n$). O centróide é útil para diversas finalidades na visão computacional, em especial para a realização de rastreamento de um objeto em um conjunto de quadros.

[4]Neste contexto, um quadro sobreposto significa a composição de dois quadros adjacentes.
[5]Neste trabalho dados aleatórios faz referência a dados pseudo aleatórios.

Este tipo de descritor pode ser usado como fonte de informações posicionais, dentro de uma seqüência de quadros, para descritores de mais alto nível (ex. o rastreamento citado anteriormente), ou pode ser usado diretamente quando a atividade a qual o centróide está sendo empregado não tenha um grau de complexidade elevado.

Dentre os métodos para a obtenção de centróides, a classificação por agrupamento de padrões [31] é a mais utilizada, onde se destaca o algoritmo *k-means* [18]. O funcionamento do *k-means* consiste no agrupamento de similaridade dos dados disponível em k conjuntos distintos (podendo k ser conhecido ou não), destes são obtidos k centróides (figura 2.5). Cada iteração do algoritmo *k-means* realiza uma comparação entre os conjuntos obtidos na iteração anterior e os conjuntos obtidos na iteração atual; caso os conjuntos sejam iguais o algoritmo termina, caso contrário uma nova iteração é realizada. Um problema inerente desta abordagem é que o número de comparações aumenta com o crescimento da quantidade de grupos.

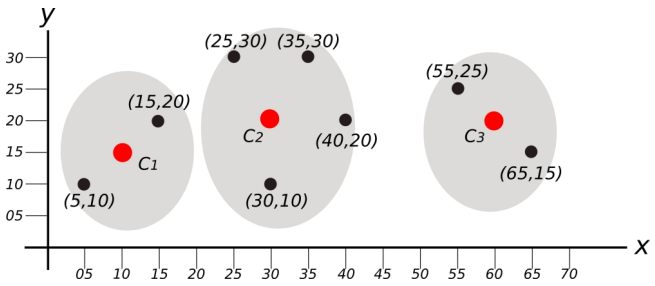

Figura 2.5: *Agrupamento de dados e seus respectivos centróides no espaço R^2. Os dados são naturalmente agrupados pelo critérios de distância, dados próximos podem corresponder ao mesmo objeto.*

Outro problema é o fato do *k-means* poder convergir para mínimos locais ou até mesmo nunca convergir. Isto pode ocorrer por existir a possibilidade de determinados dados (os quais estão sendo classificados) mudarem de conjunto consecutivamente a cada iteração. Para contornar esta deficiência pode ser suficiente definir um número máximo de iterações para garantir o término do processo, porém conseqüentemente a precisão no resultado pode ser afetada.

2.2.3 Alto nível

Os descritores de alto nível expressam conteúdos mais complexos com grande caráter semântico. Dentre eles, podem ser evidenciadas as máquinas de estados e os modelos ocultos de *Markov* [27,28]. Destes descritores somente máquinas de estados (seção 2.2.3.1) serão apresentados, uma vez que o demais não são abordados neste trabalho.

2.2.3.1 Máquinas de estados

Máquina de estado pode ser definida como um autômato finito determinístico (AFD) [12]. Os AFD têm como característica, depois de ler uma seqüencia de símbolos de entrada (*string*) estar somente em um estado. Um AFD é matematicamente descrito como uma tupla de cinco elementos $A = (Q, \sum, \delta, q_0, F)$, onde:

- Q - conjunto finito de estados;
- $\sum$ - conjunto finito de símbolos de entrada que são aceitos (alfabeto do AFD);
- δ - função de transição de um estado para o outro, dado um elemento de $\sum$;
- q_0 - estado inicial do AFD;
- F - conjunto de estados de aceitação do AFD ($F \subseteq Q$).

Os AFD's são úteis para modelar problemas simples ou até mesmo complexos que não necessitem a armazenagem de informações Um AFD pode ser representado graficamente como um grafo [29]. Na figura 2.6, é exposto um exemplo de AFD que identifica a atividade humana "sentar em uma cadeira", a qual é composta pelos estados: *longe da cadeira*, *próximo à cadeira* e *sentado na cadeira*. ($\sum = \{andar, parar, sentar\}$).

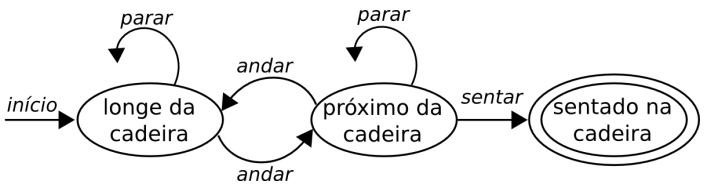

Figura 2.6: *AFD que reconhece a atividade humana "sentar em uma cadeira", onde "longe da cadeira" é o estado inicial, e "sentado na cadeira" representa o estado final do AFD ("sentado na cadeira" $\subseteq F$).*

Como exemplo de aplicações utilizando máquinas de estados finitas destacam-se trabalhos como o desenvolvido por Ayers e Shah [3], onde são modeladas atividades humanas pré-definidas em máquinas de estados finitas e aplicadas em um ambiente de escritório para identificação de tais atividades. Cutler *et al.* [7] utilizam estas estruturas para representar atividades de veículos e humanas, através de imagens aéreas.

2.3 Métodos Estatísticos de Apoio à Decisão

Neste trabalho, para classificar os *pixels* pertencentes a um quadro em *pixel* de *background* ou em *pixel* de *foreground* (seção 4.3.3) é utilizado um método estatístico de apoio à decisão. Esta seção propõe-se a fazer uma breve contextualização deste tema, assim como a descrição de dois destes métodos: o critério de *Bayes* usado na seção 4.3.3 e a média ponderada usada nas seções 4.3.4 e 4.3.5.2.

Métodos estatísticos se apóiam em observações para indicar a probabilidade de um determinado evento[6] ocorrer ou não, ao contrário dos métodos determinísticos que afirmam se o evento ocorre ou não. Como exemplo, podemos citar uma situação do mundo real: que cor é uma determinada laranja? Através de métodos estatísticos a resposta seria $n\%$ amarela, contudo para métodos determinísticos teríamos apenas que ela é amarela.

Na visão computacional, métodos determinísticos não são apropriados, isto devido a uma série de fatores relacionados à interpretação dos dados obtidos. Quando se analisa um *pixel* individualmente, este pode gerar uma interpretação divergente da obtida ao se realizar a análise em seus *pixels* vizinhos. Já os métodos estatísticos utilizam todos os dados disponíveis para aumentar ou diminuir a probabilidade do *pixel* ser algo ou não.

[6] Um evento nessa seção aborda uma situação do mundo real

Dos métodos estatísticos de apoio à decisão, o mais simples utilizado, não só na visão computacional como em outras áreas de aplicação, é a média ponderada. Esta, através de pesos obtidos por observações retorna o valor médio de uma ou mais observações. A média aritmética é uma caso de média ponderada onde os pesos dos elementos envolvidos são iguais a 1. Outros tipos de média são: médias desarmônica, média geométrica, etc.

A classificação de *Bayes* [42] é fundamentada no teorema de probabilidade de *Bayes*. Este tem como objetivo, a partir de uma amostra[7] desconhecida e um conjunto de classes, calcular a probabilidade da referida amostra pertencer a cada uma destas classes. Uma particularidade desta classificação é que ela considera que a característica de uma determinada classe é independete das características das demais classes, o que simplifica os cálculos envolvidos. A classificação de *Bayes* obtém melhores resultados quando os valores dos atributos são discretos [42].

2.4 Paralelização

Na computação, a paralelização surgiu como uma alternativa de abordagem a problemas que demandam muito tempo para ser solucionados. De uma forma mais abrangente, particiona-se o problema principal em n segmentos que serão processados individualmente por processadores diferentes. Com isso o contexto do problema e conseqüentemente o custo de processamento é dividido pelo fator de n. Após o processamento, os resultados locais dos n segmentos são fundidos para a obtenção de um único resultado final.

Quando se pretende aplicar a paralelização em um programa, dois fatores são relevantes: a granularidade da paralelização e o nível da paralelização. No contexto deste trabalho, a granularidade é definida como a quantidade de instruções que serão paralelizadas. Com isso, pode-se classificar em uma ordem crescente (considerando a quantidade de instruções) a granularidade em três conjuntos: *grão fino*, *grão médio* e *grão grosso*.

[7] Um subconjunto de elementos pertencentes a uma população

O nível da paralelização descreve em mais alto nível quais componentes do programa que se espera paralelizar. Almeida e Árabe [1] propuseram agrupar a paralelização em cinco classes focadas nos componentes do programa o qual se deseja paralelizar (figura 2.7): i) instruções, ii) laços de iterações, iii) rotinas, iv) processos ou v) programas por completo.

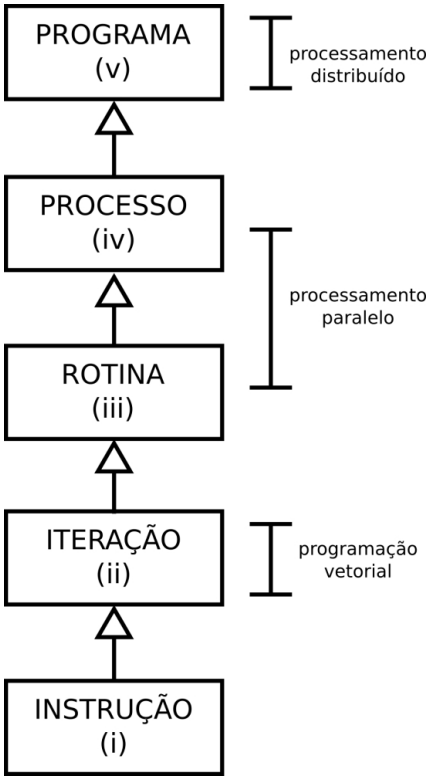

Figura 2.7: *Classificação de paralelização respeitando o critério de componente.*

2.5 Fluxo Óptico

O fluxo óptico é definido como sendo o mapeamento dos *pixels* de um quadro no momento t no quadro no momento $t+n$, sendo n uma janela de tempo (número de quadros adjacentes) e tem como objetivo principal a obtenção do deslocamento de objetos ocorrido dentro desta janela de tempo. Os métodos utilizados para detecção do fluxo óptico podem ser agrupados em três conjuntos definidos por suas abordagens: técnicas diferenciais, técnicas de correlação e técnicas baseadas em freqüência de energia.

Antes das abordagens é conveniente enunciar e descrever o principal problema do cálculo do fluxo óptico, o problema da abertura. Este problema está relacionado com a robustez e a precisão do processo e se resume em como determinar o tamanho da janela de procura da característica do quadro n em $n+1$.

A figura 2.8 demonstra o problema da abertura: em (a), o tamanho da janela não é suficiente para capturar o deslocamento do *pixel* n em destaque com a característica a, porém em (b), ao se aumentar o tamanho da janela, outros *pixels* com características similares podem ser associados com o *pixel* do quadro n e com isso gerar vetores de movimento que não correspondem ao real deslocamento do *pixel*.

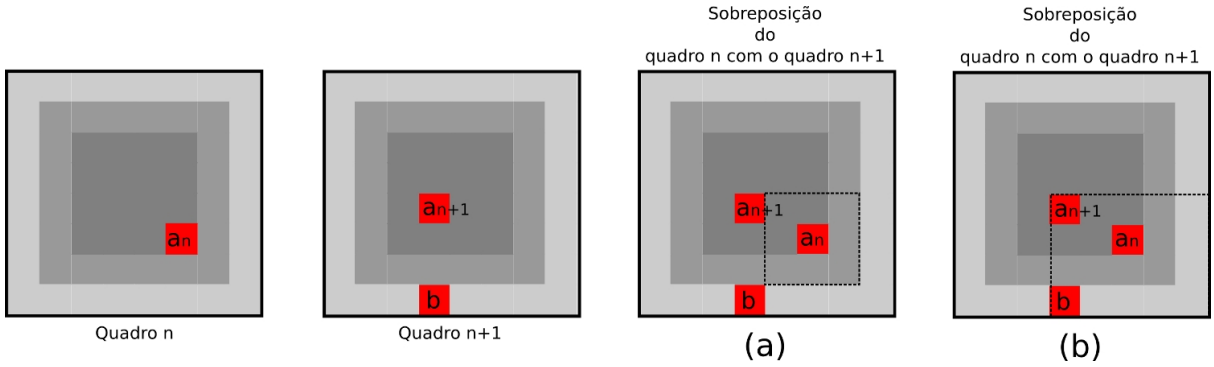

Figura 2.8: *Problema da abertura.*

2.5.1 Métodos diferenciais

Estes métodos para a obtenção do fluxo óptico se baseiam nas derivadas espaço-temporais das intensidades dos *pixels* dos quadros analisado. Dentre os métodos pertencentes a este conjunto estão os descritos por Lucas e Kanade [20] o qual tem uma abordagem local e o de Horn e Schunck [13] que possui um caráter mais global em relação ao quadro analisado. Tais métodos são sensíveis à iluminação não uniforme e instáveis em relação a novos objetos entrando no quadro, isso torna o método menos preciso e mais susceptível a erros, por outro lado há uma diminuição no tempo de processamento, tornando-o satisfatório para o processamento em tempo real [4].

2.5.2 Métodos de correlação

Este conjunto de métodos, também rotulado na literatura como "casamento de regiões", mostra-se como uma alternativa às limitações aos métodos diferenciais (seção 2.5.1). O fluxo óptico é obtido através do deslocamento de regiões, as quais tenham a melhor correspondência (correlação) entre si. A obtenção do fluxo óptico pelos métodos pertencentes a este conjunto, também se mostra eficaz e com baixo custo computacional, porém são instáveis em relação a condições de movimento linear e de baixa velocidade [4].

2.5.3 Métodos de freqüência de energia

Os métodos pertencentes a este conjunto utilizam filtros de freqüência [10] com sensibilidade direcional sobre o quadro no domínio de *Fourier* [10]. Eles se destacam devido a sua capacidade de estimar movimentos com padrões aleatórios, sendo esse um fator de sensibilidade dos outros métodos na obtenção do resultado esperado [4]. Contudo, é acrescentado ao processo o custo computacional da transformada para o domínio da freqüência dos dados obtidos no quadro. Um exemplo de aproximação pertencente a este conjunto é a abordagem proposta por Fleet e Jepson [9].

CAPÍTULO 3

TREC VIDEO RETRIEVAL EVALUATION

A *TREC Video Retrieval Evaluation* (TRECVid) surgiu em 2001 com o objetivo de oferecer a pesquisadores de todo o mundo um canal para apresentar e comparar seus trabalhos relacionados à recuperação de informações em vídeo. Realizada anualmente, é patrocinada principalmente pelo NIST[1] e outros órgãos do governo dos Estados Unidos da América.

A TRECVid está em sua 8ª edição, a cada ano novos desafios são propostos de acordo com as tendências das pesquisas no âmbito da visão computacional e do processamento de vídeo. Cada desafio é constituído por um conjunto de tarefas (obrigatórias e opcionais), que são oferecidas aos seus participantes. Os desafios são tratados de forma individual, sendo que um eventual pesquisador pode participar de uma ou mais categorias (desafios).

Nas edições anteriores da conferência TRECVid, que compreendem de 2001 a 2007, o foco principal foi a estruturação de vídeos através de seu particionamento semântico, isto é, a divisão do vídeo em unidades mais significativas do ponto de vista humano (extração de *shots* [43] e segmentação de estórias [38]), para prover entre outros benefícios uma melhor indexação e gerência de tais vídeos. Outro ponto ainda muito explorado nas conferências TRECVid é a extração de características de baixo e alto nível (seção 2.2).

Em 2008 (8ª edição) foram propostos dois novos desafios, dentre eles a *identificação de eventos de vigilância* o qual é o foco de atuação deste trabalho. A organização deste capítulo está feita da seguinte forma: na seção 3.1 é descrito o desafio de interesse *identificação de eventos de vigilância* e na seção 3.2 os eventos para serem detectados propostos aos participantes da conferência, ambos oferecidos pela TRECVid 2008.

[1] http://www.nist.gov/

3.1 Identificação de eventos de vigilância

Na conferência TRECVid 2008, foi proposta pela primeira vez a identificação de eventos relacionados à segurança. O objetivo deste desafio foi avaliar sistemas que fossem aptos a reconhecer instâncias de um conjunto pré-definidos de eventos de interesse, tendo como ambiente um movimentado aeroporto da Inglaterra (aeroporto internacional de *Gatwick*).

A identificação de eventos de vigilância foi dividida em duas categorias: a *detecção de eventos retrospectivos* e o *estilo livre*. Na primeira categoria, o foco principal foi a avaliação dos métodos propostos para detectar observações de eventos pré-definidos. Em contrapartida, no estilo livre, o objetivo foi obter inovações, explorando caminhos não abordados na categoria anterior, oferecendo para isso uma maior liberdade aos participantes.

3.2 Eventos de vigilância propostos

A definição formal de evento que foi utilizada na conferência TRECVid 2008 para o desafio de detecção de evento de vigilância é como segue: *um evento é uma ação ou mudança de estado no vídeo que pode ser importante para o gerenciamento de segurança do aeroporto.* Foram definidos e disponibilizados 17 (dezessete) eventos para detecção [2]. Para cada um deles foi indicado, além de sua descrição, o quadro inicial e o quadro final. Na próxima seção (3.2.1) será descrito o evento que é foco de atuação deste estudo, fluxo oposto. A descrição dos demais eventos podem serem encontradas no apêndice B.

3.2.1 E20 – Fluxo oposto

Neste evento alguém se move através de uma porta na direção oposta ao fluxo normal do tráfego. Começa no momento em que a pessoa está iniciando o movimento através da porta e termina no momento em que a pessoa tem passado totalmente através porta. Um requisito para este evento é definição prévia do fluxo normal do tráfego. Uma peculiaridade é que a TRECVid 2008 restringe somente a câmera 1 para detectar este evento.

[2] http://www-nlpir.nist.gov/projects/tv2009/tv2009.html

CAPÍTULO 4

SISTEMA DE IDENTIFICAÇÃO DE AH PROPOSTO

O sistema de identificação de atividades humanas proposto neste estudo foi baseado no trabalho de Fernando *et al.* [8], e teve como objetivo principal a identificação do fluxo oposto de objetos em um ambiente com fluxo controlado. Para o treinamento e testes do sistema foi utilizado como ambiente a área de desembarque do aeroporto internacional de *Gatwick*. Este ambiente foi oferecido pela base de vídeos da *TRECVid 2008/2009* (seção 4.3.1).

O fluxo oposto é caracterizado quando primeiramente temos um fluxo uniforme de objetos (considerando sua orientação) e é detectado um objeto em orientação não uniforme, isto é, a direção do deslocamento do objeto não condiz com a direção da maioria dos elementos observados. Para obter tanto a orientação dos objetos contidos no quadro quanto à desuniformidade da direção, foram usados respectivamente: vetores de movimento (seção 2.2.2.1) e o ângulo do vetor de movimento em relação ao eixo x do quadro analisado.

Este sistema de identificação de AH também explora a paralelização de grão médio, através de *threads*, oferecida pelas plataformas *hardware* e *software* disponíveis atualmente. Outras características e métodos utilizados como suporte para a realização tanto da paralelização quanto do fluxo óptico também são descritos, porém com menos ênfase.

O restante deste capítulo está organizado como segue: na seção 4.1 é evidenciada a arquitetura geral do sistema, na seção 4.2 são descritas a plataforma de *hardware* e de *software* utilizadas. A implementação de cada etapa do sistema são evidenciadas na seção 4.3. A base de vídeos, os testes e os resultados são detalhados nas seções 4.4 e 4.5.

4.1 Arquitetura do Sistema

Fernando *et al.* em [8] utilizaram o cálculo do fluxo óptico entre dois quadros adjacentes, o agrupamento dos vetores de movimento e a análise da direção dos grupos formados para determinar se um objeto condiz ou não com a direção de robôs. Esta dissertação propôs a mudança do contexto do sistema original para identificação de atividades humanas que envolvessem movimento, mais especificamente a identificação de fluxo oposto de objetos, além da contribuição para tornar o sistema mais robusto a ruídos pela filtragem do fluxo óptico através da remoção de *background*, e da diminuição da complexidade computacional envolvida no processo através da exploração da paralelização. A figura 4.1 exibe a arquitetura do sistema de identificação do fluxo oposto proposto por este trabalho.

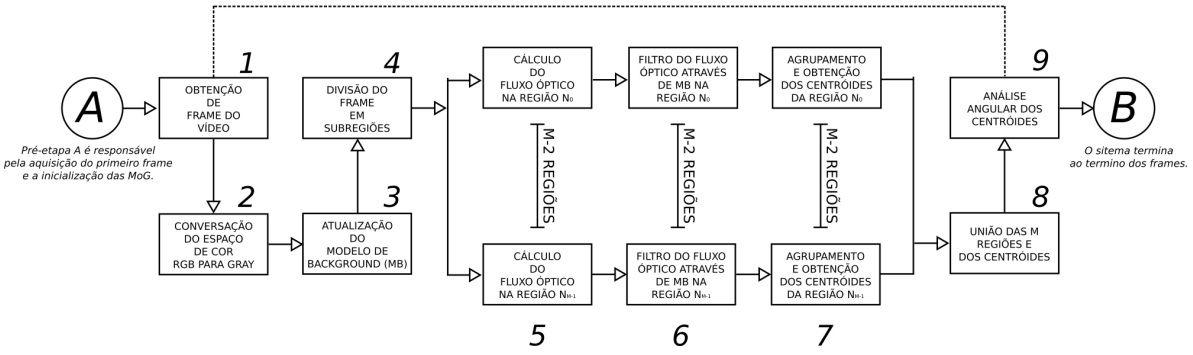

Figura 4.1: *Arquitetura geral do sistema de identificação de fluxo oposto. Com exceções das etapas 5, 6 e 7, que são executadas focando a paralelização, as demais etapas são todas executadas em série.*

Ao total o sistema é composto por nove etapas, as quais são enunciadas a seguir e descritas com mais detalhes na seção 4.3: 1) captura de quadros do vídeo recebido como parâmetro, 2) conversão de espaço de cor RGB para a escala de tons de cinza, 3) atualização do modelo de *background* (MB) através dos dados extraídos do quadro recebido, 4) divisão do quadro em sub-regiões processadas em *multithread*, 5) cálculo do fluxo óptico de cada sub-região, 6) filtro do fluxo óptico através do MB, 7) agrupamento dos vetores de movimento, obtidos pelo fluxo ópticos, e obtenção de centróides, 8) união das regiões e dos centróides (fusão de resultados), e por último, 9) avaliação de resultado através da análise da angulação dos vetores de movimento.

4.2 Plataforma de *Hardware* e *Software* Utilizada

Nesta seção serão descritos os elementos de *hardware* e *software* utilizados para a implementação do sistema proposto, tal como o ambiente utilizado para sua execução e para seus testes. A tabela 4.1 exibe o *hardware* e está organizada em quatro colunas as quais indicam respectivamente: i) o modelo do processador utilizado, ii) o *clock*[1] máximo indicado pelo fabricante que o processador pode atingir, iii) o número de núcleos que cada processador contém, e iv) a quantidade de memória principal que o *hardware* possui.

Ainda considerando o *hardware*, foram utilizados dois computadores com tecnologias distintas e quantidades de núcleos diversificadas. Esta estratégia foi adotada com o objetivo de verificar e validar a efetividade do sistema quanto à obtenção do paralelismo, sendo que ele de fato somente é atingido quando o *hardware* utilizado tem mais de um núcleo, podendo ser multiprocessado ou com um processador com vários núcleos.

Modelo do processador	*Clock* máximo	Qtde núcleos	Qtde memória
Mobile Sempron 3500+	1.8Ghz	1	1 GBytes
AMD Athlon 64 X2 Dual Core Processor 5000+	2.6Ghz	2	2 GBytes

Tabela 4.1: *Lista de hardware utilizado no desenvolvimento, execução e testes do sistema de identificação de fluxo oposto proposto.*

O conjunto de *software* é exibido na tabela 4.2, onde cada linha descreve a categoria, o nome e a versão de um *software*. Para simplificar a leitura, foram tabulados somente os elementos mais críticos para implementação sendo eles o sistema operacional, o compilador e as bibliotecas envolvidas na implementação e nos testes do sistema proposto.

Um destaque no conjunto de *software* utilizado é a biblioteca *OpenCV*[2] desenvolvida pela *Intel*[3]. Esta se destaca por sua eficiência e por ter um conjunto diversificado de funções que vão desde a manipulação de imagens (conversões de escala, de espaço de cor, etc), como tarefas mais avançadas como classificação de padrões e filtros estatísticos.

[1] Velocidade através da qual o processador executa as instruções.
[2] http://opencv.willowgarage.com/wiki/
[3] http://www.intel.com/

Categoria do *software*	*Software* Utilizado	Versão
Sistema operacional	*Ubuntu Hardy Heron*	8.04 com *kernel 2.6.24-23-Generic*
Compilador	GNU *Cross Compiler* (GCC)	4.2.4
Biblioteca	libc	2.6.1
Biblioteca	openCV	1.0.0-4

Tabela 4.2: *Lista de software utilizados no desenvolvimento, execução e testes do sistema de identificação de fluxo oposto proposto.*

4.3 Implementação

A partir deste ponto as etapas do sistema de identificação de fluxo oposto serão denominadas de módulos que serão descritos individualmente nas seções subseqüentes. Estas descrições serão focadas na implementação, heurísticas utilizadas e resultados esperados. Os parâmetros dos módulos e limiares de ajuste também serão evidenciados. Estes parâmetros são responsáveis pelo controle da eficiência e do desempenho do sistema, considerando fatores como iluminação, complexidade computacional, ruídos, dentre outros.

Como já dito anteriormente as contribuições do sistema proposto nesta dissertação ao sistema descrito por Fernando *et al.* em [8] foram respectivamente: i) o método de remoção de ruídos do fluxo óptico através da remoção do *background*, ii) a modificação do algoritmo de agrupamento, e iii) a paralelização. Com a exceção dos itens i), ii) e iii), todos os demais algoritmos foram implementados usando funções disponíveis na *openCV*.

As demais seções descrevem a arquitetura da figura 4.1. As seções 4.3.1 e 4.3.2 descrevem o processo de captura de quadros e a conversão de seus espaços de cores. Na seção 4.3.1 é apresentado o método de remoção de *background* utilizado neste trabalho. A abordagem de identificação de fluxo oposto é detalhada na seção 4.3.4. A paralelização, uns dos focos principais deste trabalho, é descrita na seções 4.3.5. Na seção 4.4 e 4.5 são apresentadas respectivamente: as bases de vídeos utilizadas para validar os sistema proposto, e os testes e resultados obtidos.

4.3.1 Abertura de Vídeo e Captura de Quadros

Este módulo é responsável pela leitura e extração dos quadros de interesse do arquivo de vídeo recebido como entrada. Os quadros extraídos são de fundamental importância para o sistema, porque neles que estão contidas todas as informações espaciais que serão analisadas pelos módulos subseqüentes.

É válida nesta seção uma breve explicação sobre os conceitos de arquivo, formato e *codecs* de vídeo. Um arquivo pode ser caracterizado como um "contêiner", ou casca, o qual é composto por vídeo, som e outros itens em seu estado bruto e conseqüentemente com um tamanho em *bytes* elevado. O *codec* é responsável justamente por codificar os itens citados anteriormente, com isso se busca a diminuição do espaço utilizado no estado bruto, a regulagem da qualidade de exibição e controlar a compatibilidade entre os visualizadores deste tipo de mídia. O conjunto arquivo mais *codec* define um formato de vídeo.

4.3.2 Conversão de Espaço de Cor

O quadro extraído através do módulo de captura está no espaço de cor RGB, este espaço de cor é composto por três bandas de dados que expressão a matiz vermelha (R), verde (G) e azul (B). Empiricamente concluímos que, para o contexto deste trabalho, as informações contidas nas três bandas não são totalmente necessárias, sendo que a diferença de intensidade da iluminação entre os *pixels* de quadros adjacentes já é suficiente para a execução dos demais algoritmos contidos no sistema proposto.

Com isso, o espaço de cor utilizado foi substituído pelos tons de cinza os quais caracterizam a intensidade de iluminação dos *pixels* do quadro e que possui somente uma banda de dados. Com a utilização de somente uma banda de dados, o esforço computacional empregado é reduzido pelo fator de três, e conseqüentemente os resultados são obtidos mais rapidamente. Seus pontos negativos são o ruído gerado pela perda de informação no processo de transformação e a sensibilidade deste espaço de cor à iluminação não uniforme.

Para a conversão do espaço RGB para tons de cinza foi utilizada a equação (4.1) descrita por Gonzales e Woods [10] e implementada pela biblioteca *openCV*. A equação é uma média ponderada das bandas pertencentes ao espaço de cor original, sendo que os pesos foram determinados considerando a capacidade de observação do olho humano à luminosidade. Pode ser visto na figura 4.2 o quadro original (à esquerda) e o quadro após a conversão (à direita) em tons de cinza.

$$Cinza = R \times 0,299 + G \times 0,587 + B \times 0,114 \tag{4.1}$$

(a) (b)

Figura 4.2: *Conversão do espaço de cor utilizado na implementação. Em (a) o quadro original no espaço de cor RGB, na ordem BGR, em (b) o quadro convertido para tons de cinza.*

4.3.3 Remoção de *Background* Através de Regra de *Bayes*

O sistema proposto neste trabalho utiliza o resultado do cálculo do fluxo óptico (descrito na próxima seção) para a identificação da AH de interesse. Para refinar o resultado obtido é realizada a filtragem dos vetores de movimento através da remoção do *background* com atualização do quadro de referência.

A idéia da filtragem por remoção de *background*, que podemos chamar de modelagem de *background*, é eliminar os vetores de movimentos que não foram originados dos objetos em movimento, como por exemplo os resultantes da iluminação não uniforme. O método de filtragem se resume em identificar os elementos de *foreground* do quadro e descartar todos os vetores de movimento que não estejam relacionados com estes elementos.

Nesta dissertação, a modelagem do *background* é realizada pelo algoritmo descrito por Li *et al.* em [19]. A atualização do quadro de referência utilizada por Li é realizada através do método estatístico de apoio à decisão de *Bayes* (seção 2.3), que classifica os *pixels* extraídos dos quadros em um determinado momento t, em *pixels* pertencentes ao *foreground* ou pertencentes ao *background*.

O algoritmo de Li *et al.* para modelagem de *background* utilizada no sistema de AH proposto é implementado pela biblioteca *openCV*. Li *et al.* também em [19] realizam uma comparação de seu método com outras duas técnicas de segmentação: as misturas gaussianas [16] e a subtração de fundo a partir de uma quadro de referência sem atualização. No primeiro caso, a modelagem de *background* se mostrou mais eficiente na geração dos resultados e sem a complexidade computacional de manter as gaussianas. No segundo caso, como o quadro referência não é atualizado, alguns elementos que passam a pertencer ao *background* continuam sendo classificados como objetos de *foreground*.

O MB é inicializado no começo do sistema, logo após a obtenção e a conversão do espaço de cor do primeiro quadro capturado (na primeira iteração o MB é composto por todos os elementos do primeiro quadro). A cada nova iteração é realizada a atualização do MB com os dados provenientes do novo quadro capturado. Na figura 4.3 (b) é exibido o MB resultante da subtração do quadro (a) com o quadro de referência gerado pelo algoritmo de Li, o resultado é um quadro binário contendo os elementos de *foreground* (áreas em branco) e de *background* (áreas em preto).

(a) (b)

Figura 4.3: *Modelo de background. É realizada uma segmentação dos elementos de foreground (áreas em branco) e background do quadro analisado (áreas em preto).*

4.3.4 Identificação de Fluxo Oposto

A identificação do fluxo oposto está diretamente associada com a direção dos objetos contidos e considerados no quadro analisado. Sendo assim, foram utilizados vetores de movimento, obtidos através do fluxo óptico (seção 2.5), para a identificação da AH de interesse. Dois problemas foram identificados quando foi calculado o fluxo óptico: divergência na direção do movimento entre partes de um mesmo objeto, e o custo elevado para realizar o mapeamento de todos os *pixels* do quadro no momento t em $t + 1$. Para resolver estes problemas foram utilizados respectivamente, a obtenção de vetores resultantes através de agrupamento, e a paralelização de regiões do quadro analisado. As subseções seguintes têm como objetivo descrever as abordagens e os métodos utilizados na obtenção dos itens acima.

4.3.4.1 Fluxo óptico

No sistema proposto por este trabalho para calcular o fluxo óptico, foi utilizado o método diferencial descrito por Bouguet em [5], o qual se mostra como um melhoramento da abordagem de Lucas e Kanade [20], por se propor a compensar o problema da *abertura*, discutido na seção 2.5. Para superar este problema, e conseqüentemente detectar movimentos com grande deslocamento e manter a precisão, a imagem foi representada em uma pirâmide o que tornou o mapeamento de características mais eficiente. O processo e o algoritmo estão descritos em [5]. A escolha desta abordagem diferencial (seção 2.5.1) foi devido ao seu bom desempenho na utilização em tempo real e sua baixa complexidade em relação aos métodos baseados em freqüência (seção 2.5.3).

O método diferencial [5] realiza o mapeamento da característica de interesse entre quadros adjacentes. A característica de interesse utilizada por este trabalho foram os cantos, que são regiões que contêm um grande nível de curvatura na sua imagem gradiente espacial[4]. Neste trabalho, os cantos são obtidos através do algoritmo desenvolvido por Shi e Tomasi [32]. Na figura 4.4 são exibidos os cantos mais comumente detectados.

[4]http://en.wikipedia.org/wiki/Feature_detection_(computer_vision)

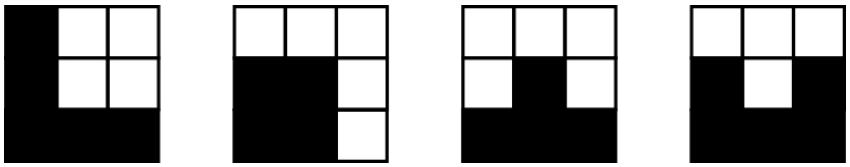

Figura 4.4: *Possíveis cantos. Para cada canto também são válidas todas suas rotações.*

A escolha dos cantos se deu ao fato de esta característica já ter sido utilizada pelos autores do método de obtenção de fluxo óptico utilizado [5], além de oferecer um resultado satisfatório para a realização do mapeamento no contexto deste trabalho. Como exemplo, a figura 4.5 (a) exibe os cantos encontrados em uma região do quadro no momento t e o mapeamento dos cantos na região no momento $t + 1$ (b), assim como os vetores de movimentos originados deste mapeamento (c). É possível que alguns cantos não possam ser mapeados devido a fatores como: erro no mapeamento, não houve mudança nas regiões consideradas, etc.

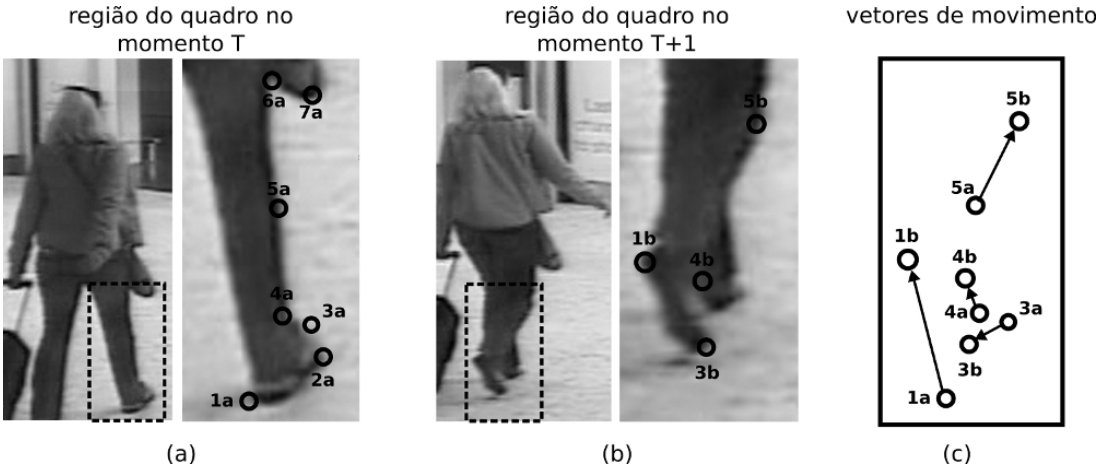

Figura 4.5: *Mapeamento de características em quadros adjacentes. Na figura apenas são destacados alguns pontos em um região do quadro principal.*

O algoritmo de Shi e Tomasi [32] recebe dois argumentos que controlam a qualidade e o número de cantos detectados, e que conseqüentemente determinam a quantidade de vetores de movimento, são eles: a distância e a qualidade dos cantos cantos detectados. A distância determina o quão próximos os cantos detectados estarão um dos outros, sendo que é utilizada a distância Euclidiana [10] para determinar a proximidade. A qualidade dos cantos determina o número de cantos detectados.

Uma abordagem assumida neste trabalho para se obter mais vetores de movimento e com isso mais deslocamentos, foi diminuir a qualidade e a distância dos cantos do algoritmo de detecção de cantos. Como conseqüência foram detectados cantos que não correspondiam com objetos em movimento, interpretados como interferências[5] que são destacados na figura 4.6.

Figura 4.6: *Vetores de movimentos originados de um objeto em movimento (uma mulher) e de interferencias (região central do quadro).*

Para suprimir as interferências geradas pela diminuição de qualidade dos cantos detectados, foi realizada a filtragem de vetores de movimento através do MB. A filtragem foi realizada comparando as coordenadas dos vetores de movimento com o MB. Se o ponto de origem ou de destino do vetor de movimento estivesse em uma região que pertencera ao *background*, o vetor era considerado ruído, e com isso eliminado do conjunto de vetores aceitos.

Tal técnica se mostrou satisfatória para remover interferências de qualidade e também ruídos de outras fontes, como por exemplo os provenientes dos dispositivos de captura de vídeo. Na figura 4.7 é exibido o resultado da aplicação do filtro por MB, onde os vetores de movimento contidos no quadro são comparados com o MB, se a coordenada de origem ou de destino do vetor não condizer com um elemento de *foreground* o vetor é considerado ruído. Neste exemplo somente algumas características reais foram consideras.

[5]Nesta dissertação interferência é definida como elementos que geram entropia ao sistema.

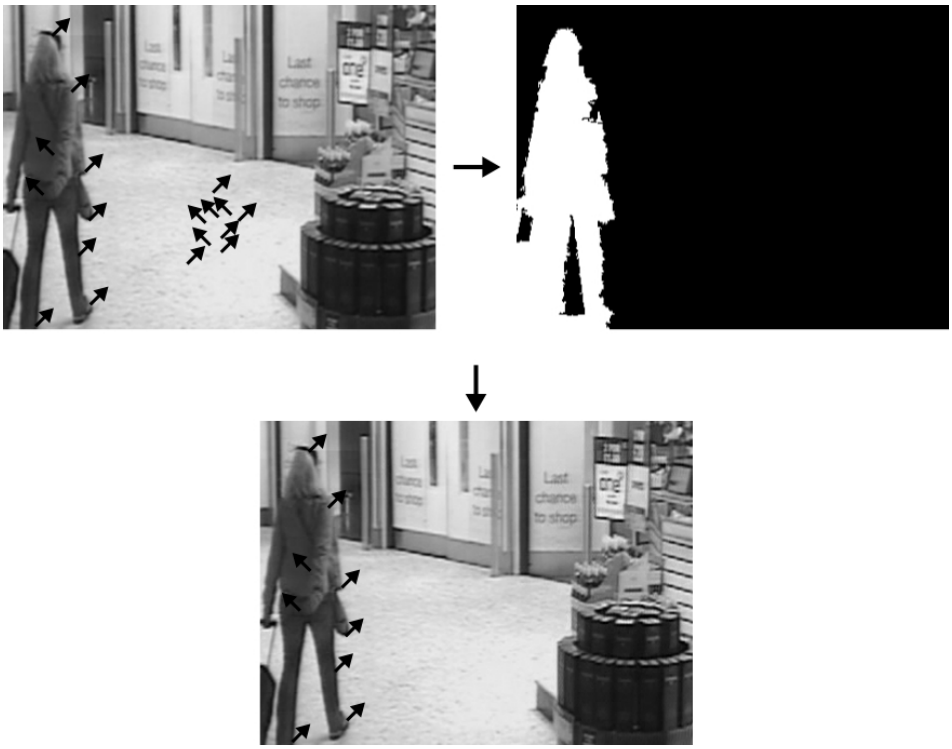

Figura 4.7: *Filtragem dos ruídos contidos em um quadro através de MB.*

4.3.4.2 Obtenção dos vetores de movimento resultantes

O resultado do módulo responsável pelo cálculo do fluxo óptico e pelo filtro das possíveis interferências e ruídos é um campo de vetores de movimento que pretendem indicar o real deslocamento de *pixels* dentro da janela de quadros. Porém manusear todos os vetores elevam o custo de processamento, assim pode-se relacionar o tempo de análise dos vetores de movimento ao tempo gasto para se obter os resultados.

Outro problema inerente da análise individual dos vetores pertencentes a um objeto, é que o vetor de movimento pode não condizer com a real direção do objeto, isto devido a alguns objetos (pessoas, animais, plantas, etc) não apresentarem simetria em seus movimentos. Em um exemplo exibido na figura 4.8 temos uma mulher se deslocando, alguns membros de seu corpo (braços) vão em contraposição à grande maioria, com isso se analisarmos os vetores de movimento individualmente teremos o fluxo esperado e oposto no mesmo objeto, gerando com isso identificação errônea da AH.

Figura 4.8: *Vetores de movimento pertencentes ao mesmo objeto podem possuir direções divergentes. Na figura, dois vetores tem direção divergentes das demais por o braço fazer um movimento pendular.*

A alternativa adotada por esta implementação para diminuir o custo de processamento e amenizar a detecção errônea de movimento foi a obtenção dos vetores de movimento resultantes. Um vetor de movimento resultante associado à um objeto é uma representação de todos os VM originados deste objeto e próximos entre si. A coordenada origem de um vetor de movimento resultante é a média das coordenadas origens de todos os VM relacionados com ele, o que acaba sendo a definição de centróide (seção 2.2.2.2).

Os centróides podem ser obtidos por agrupamentos de padrões (seção 2.2.2.2). Das técnicas de agrupamentos, a utilizada por este trabalho foi a obtenção dos centróides através do algoritmo *k-means* modificado (com $k = 8$ obtido empíricamente em testes das sequências de vídeos análisado), escolhido por ser um método iterativo sem a complexidade computacional presente nos métodos estatísticos e pelos resultados descritos em [34], onde se mostrou melhor que outros métodos de agrupamento.

O critério de agrupamento dos vetores de movimento utilizado foi a distância Euclidiana [10] entre eles. A escolha deste critério partiu do pressuposto que vetores de movimentos próximos pertencem ao mesmo objeto. A modificação realizada foi a remoção das comparações dos grupos entre iterações adjacentes, substituindo-as por um número fixo de iterações obtido empiricamente diminuindo assim o custo computacional empregado na comparações dos grupos, sendo que seus resultados se mostraram satisfatório no contexto deste trabalho e comprovado nos testes realizados na seção 4.5.

O resultado do agrupamento realizado são os centróides, que representam os grupos de vetores de movimento obtidos. Após a obtenção dos centróides, todos os vetores de movimento são deslocados aos seus respectivos centróides, coincidindo suas origens. Após o deslocamento, para cada centróide é realizado o cálculo da média aritmética das coordenadas finais dos vetores associados a ele.

O conjunto centróide e média caracterizam o vetor de movimento resultante, que contém mais qualidade e precisão nas informações relacionadas. Isto se deve ao fato de que o vetor de movimento resultante representa a média dos vetores associados a ele, com isso a direção é deslocada para onde a maioria dos vetores pertencentes ao centróide aponta (figura 4.9), eliminando com isso interferências de direções não uniforme.

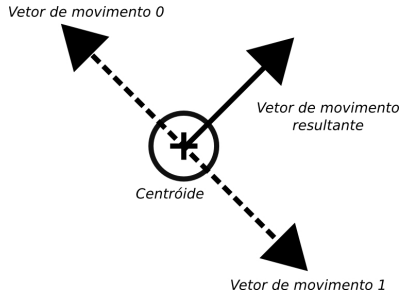

Figura 4.9: *Representação gráfica do vetor de movimento resultante. A média dos vetores movimento que compõem o agrupamento determina o vetor resultante.*

4.3.4.3 Análise angular dos vetores resultantes

A análise angular dos centróides se apresenta como a última etapa do sistema de reconhecimento de ações humanas proposto. Sua escolha como abordagem para identificação de AH foi porque os vetores de movimento resultantes ofereceram todos os dados necessários para a análise angular, simplificando a tarefa.

Nesta etapa temos todos os vetores de movimentos extraídos do quadro, cada qual representando um objeto em movimento. Em cada um dos vetores de movimento resultante é realizada uma análise angular para identificar se o objeto caracterizado pelo vetor esta em fluxo oposto ou não. A análise é descrita com mais detalhes no decorrer desta seção.

Para cada vetor de movimento resultante $\overrightarrow{x}$ é calculado seu ângulo θ de orientação, este ângulo se forma entre o $\overrightarrow{x}$ e o eixo x do quadro, considerando o quadro representado em duas dimensões (x, y). Para cálculo do ângulo de orientação é utilizado a função trigonométrica citada em (4.2) que retorna diretamente o valor esperado em grau.

$$\theta = \arccos \frac{(x_1 - x_0)}{\sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2}} \tag{4.2}$$

Após a obtenção dos ângulos de orientação θ, é realizada a comparação destes com dois ângulos limiares α_1 e α_2, que definem a faixa de aceitação do fluxo normal. Os limiares definem o fluxo oposto e o fluxo normal do ambiente, eles foram obtidos através do conhecimento prévio do contexto analisado: a área de desembarque do aéroporto *Grawitch*, a qual tem restrições quanto a circulação. Assim tudo que estiver dentro desta faixa será caracterizado como fluxo oposto. Na figura 4.10 a área em cinza representa a faixa de ângulos considerados fluxo oposto.

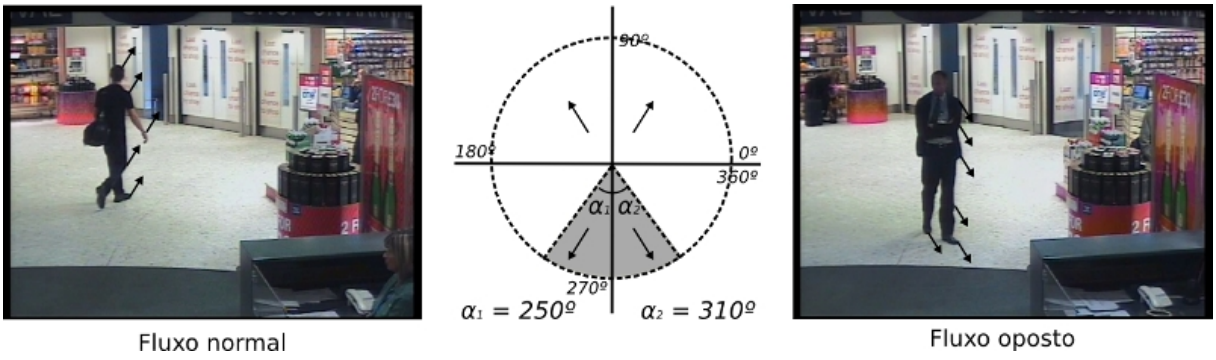

Figura 4.10: *Intervalo de valores de ângulos aceitos pela aproximação. Os objetos, representados pelos vetores de movimento resultantes, dentro desta faixa são caracterizados como fluxo normal. O fluxo normal é definido através do conhecimento prévio do contexto onde está sendo aplicada a técnica.*

Um exemplo do funcionamento do sistema de identificação de AH é exibido na figura 4.11, onde é evidenciado o funcionamento de uma iteração do sistema.

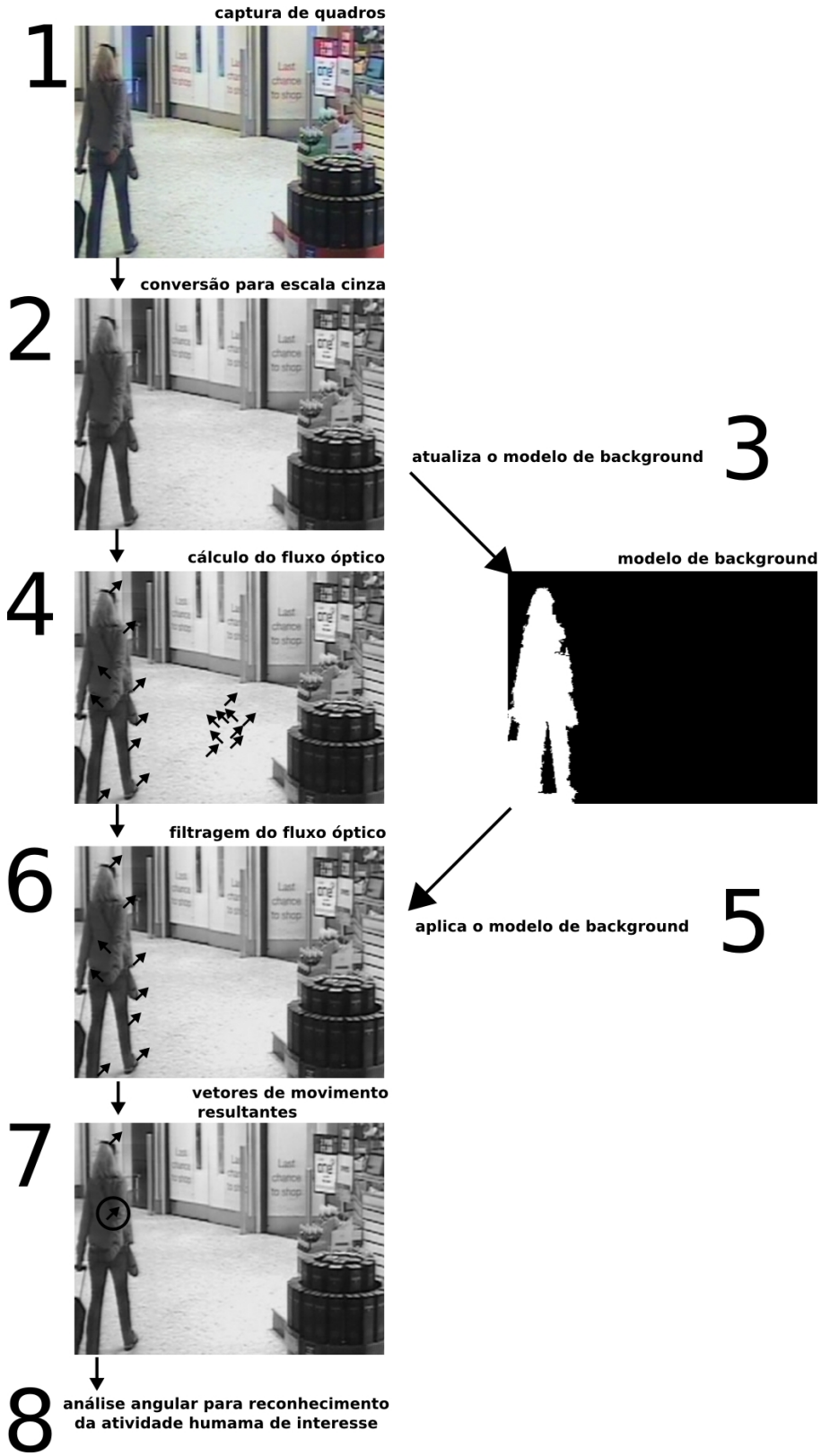

Figura 4.11: *Identificação da AH de interesse. Os passos envolvidos na iteração são: 1) captura de um quadro, 2) conversão do espaço de cor do quadro original, 3) atualização do MB, 4) cálculo do fluxo óptico, 5-6) filtragem do fluxo óptico por MB, 7) obtenção dos vetores de movimento resultante, e 8) identificação da atividade humana por análise angular.*

4.3.5 Paralelização

O processo que compreende desde o cálculo do fluxo óptico até a a análise angular dos vetores resultantes, quando realizada em um quadro completo demanda um esforço computacional relativamente alto, o qual é diretamente proporcional à dimensão do quadro analisado.

Como já mencionado anteriormente, este trabalho explora a paralelização oferecida pelos elementos de *hardware* e *software* atuais. Nesta implementação, foi utilizada a paralelização de grão médio em nível de rotinas (seção 2.4), utilizando *threads*. Esta escolha foi assumida devido à paralelização de grão médio, no contexto deste trabalho, oferecer os recursos suficientes para a obtenção do fluxo oposto e também pelo fato de que a maioria dos sistemas operacionais atuais oferecem suporte nativo a *threads*. Na seção 4.3.5.1 e 4.3.5.2 são descritas as abordagens utilizadas para realizar a divisão do quadro e a fusão das áreas envolvidas na paralelização respectivamente.

4.3.5.1 Particionamento da área do quadro

O objetivo quanto ao particionamento do quadro em regiões foi segmentar o esforço computacional do problema principal explorando a paralelização oferecida pelos *hardware* e *software* atuais. O tipo de paralelização utilizada nesta implementação é a de grão médio, utilizando *threads*. Cada região é executada em uma *thread*, sendo assim o número de regiões particionadas está diretamente relacionada ao número de *threads* que serão executadas.

Neste trabalho o quadro foi dividido em 4 regiões (figura 4.12), executadas em *threads* diferentes. A escolha de quatro regiões é conseqüência de dois fatores: a estratégia de fusão descrita na próxima seção e a arquitetura atual dos processadores. Atualmente, pode-se encontrar facilmente processadores de quatro núcleos com preços acessíveis; as *threads* neste ambiente são executadas em paralelo, porém em ambientes que só tem um núcleo o processo não atinge o desempenho esperado porque os fluxos são serializados.

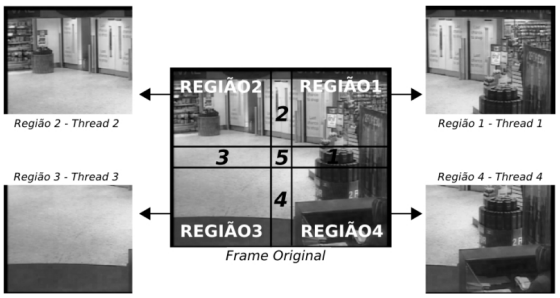

Figura 4.12: *Divisão do quadro analisado em m regiões, nesta implementação* $m = 4$.

4.3.5.2 Fusão de regiões

A estratégia de fusão assumida por este trabalho consiste na sobreposição de áreas, onde todas as regiões particionadas possuem áreas de intersecção com suas regiões adjacentes (1, 2, 3, 4 e 5 na figura 4.12). A fusão se faz necessária devido a cada região particionada e analisada individualmente possuir um contexto reduzido em relação ao quadro principal. Esta redução no contexto pode gerar uma interpretação diferente daquela que se obteria se fosse analisado o quadro por completo.

A importância da fusão fica evidente quando a análise é realizada nos *pixels* próximos às bordas das regiões obtidas no processo de particionamento. Os centróides são obtidos através do agrupamento de vetores de movimentos, este por sua vez podem ter problemas de direção, pois como já mencionado anteriormente, alguns corpos não possuem simetria em seus movimentos. Como exemplo, considere uma pessoa em movimento, seu tronco vai para frente mas seu braço vai para trás. Se analisados separadamente, gerarão fluxos contraditórios. Nestes casos, a fusão permitirá a análise correta.

Este problema se acentua nos *pixels* próximos às bordas, pois ao realizar o agrupamento para geração dos vetores de movimento resultantes nestas regiões não são levados em consideração os vetores que estão em regiões analisadas em outras *threads*. Isso pode ocasionar a perda de precisão na identificação da AH de interesse devido ao fato de possíveis surgimentos de falsos vetores de movimentos resultantes, o quais não existiriam se fosse considerado o quadro completo.

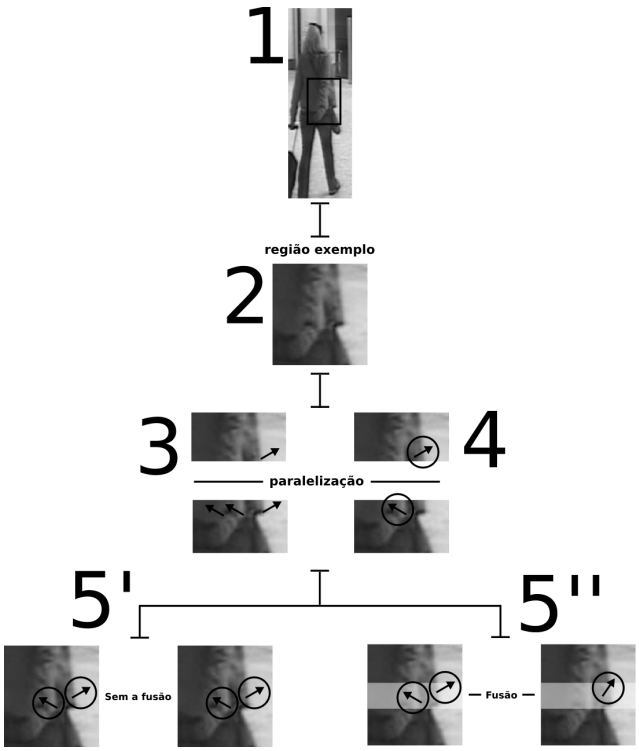

Figura 4.13: *Fusão de regiões. Para melhorar a ilustração são considerados somente alguns vetores de movimento em uma sub-região do quadro completo.*

A situação descrita no parágrafo anterior é demonstrada na figura 4.3. Inicialmente é realizada o particionamento em duas regiões (2) de uma área extraída do quadro principal (1). Em seguida, o fluxo óptico é calculado (3) em cada região, assim como seus respectivos VM resultantes (4). Caso não seja utilizada a fusão todos os VM resultantes serão considerados (5'), prejudicando com isso a precisão na identificao do fluxo oposto, porém aplicando a fusão (5") é obtido o real deslocamento do objeto contido na área extraída do quadro principal. A figura 4.14 exibe o processo da fusão.

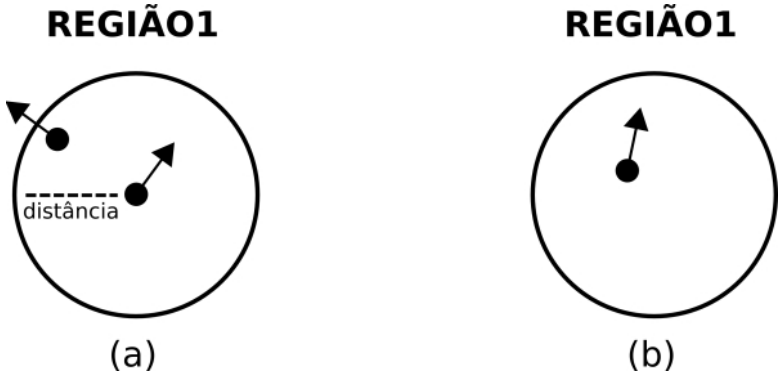

Figura 4.14: *Refinamento de vetores de movimento resultante.*

O processo de fusão consiste na busca e análise dos centróides mais próximos entre si. Os centróides obtidos pelo agrupamento de VM nas áreas particionadas são inseridos em um *array* de oito posições, onde cada duas posições representa uma área particionada: 1 e 2 representam a primeira área particionada, 3 e 4 a segunda área, e assim sucessivamente. A cada iteração a posição a_n é comparada com a posição a_{n+1} caso o centróide esteja na faixa de intersecção definida na figura 4.12, e dentro de uma distância r (figura 4.14), é realizada a média entre eles e um é suprimido. O processo é descrito no algoritmo abaixo.

Entrada: *array* com 8 posições contendo os centróides e as regiões de fronteiras.

Saída: *array* com o resultado da união dos contextos das áreas particionadas.

```
function calculaDistancia(vetorA, vetorB):
    distancia = sqrt((vetorA.x − vetorB.x)^2 + (vetorA.y − vetorB.y)^2) ;
    return distancia;

function uneVetores(vetorA, vetorB):
    vetorResultado = (vetorA + vetorB)/2;
    return vetorResultado;

begin
    for i = 0; i < 7; i++ do
        if array[i] dentro de alguma área de fronteira then
            for j = i; i < 7; j++ do
                distancia = calculaDistancia(array[i], array[j]);
                if distancia <= r then
                    array[i] = uneVetores(array[i], array[j]);
                end
            end
        end
    end
end
```

Algoritmo 4.1: *Fusão de contexto das áreas particionadas.*

4.4 Base de Testes Utilizada

Para testar a efetividade desta implementação foi utilizada uma base de vídeo de larga utilização na comunidade de desenvolvedores de aproximações para identificação atividades humanas. Esta base foi e ainda é utilizada, dentre outras conferências, na *TRECVid* 2008/2009. A base é composta por vídeos abrangendo cenários *indoor* de um aeroporto da Inglaterra. A seguir, na seção 4.4.1 a base de vídeo é descrita.

4.4.1 Aeroporto de *Grawitch*

Como base de avaliação foram disponibilizadas gentilmente pela *UK Home Office*[6] aproximadamente 100 horas de vídeo geradas por cinco das câmeras de vigilância do aeroporto *Grawitch*. Estas 100 horas foram obtidas ao longo de 10 dias, através da gravação de 2 horas/dia em cada uma das 5 câmeras distribuídas no aeroporto. Esta estratégia de aquisição de vídeos foi traçada com o objetivo de tornar a base mais diversificada. Os vídeos disponibilizados foram divididos em dois conjuntos: base de testes e base de avaliação, sendo 5 dias de vídeo para os testes e 5 dias para os vídeos de avaliação. Na figura 4.15 são espostos quadros de cada uma das cinco câmeras da base de vídeo *TRECVid*.

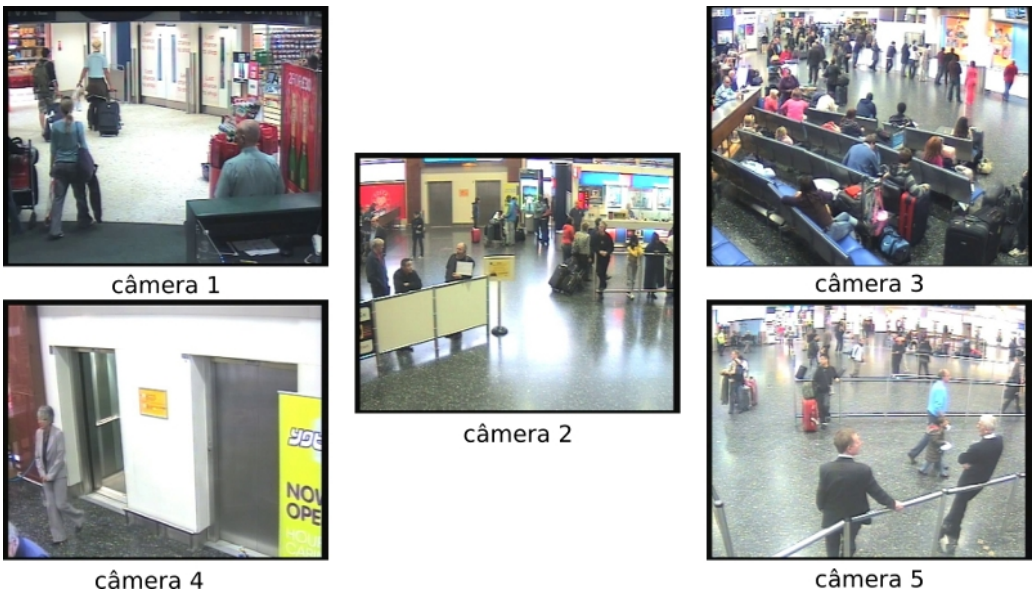

Figura 4.15: *As cinco câmeras de vigilância do aeroporto Grawitch.*

[6]http://www.homeoffice.gov.uk/

4.5 Testes e Resultados Obtidos

O sistema de identificação de fluxo oposto foi submetido a seis configurações de testes compostas pelas variações das contribuições propostas neste trabalho. Os objetivos da elaboração destas configurações de testes foram verificar a precisão e o desempenho do filtro por MB do fluxo óptico (seção 4.3.3), e da paralelização e fusão de regiões (seção 4.3.5). As configurações de testes são enunciados na tabela 4.3.

Configurações de testes	Componentes do sistema de identificação AH
01	sem paralelização e sem filtro por MB;
02	sem paralelização e com filtro por MB;
03	com paralelização, sem fusão e sem filtro por MB;
04	com paralelização, com fusão e sem filtro por MB;
05	com paralelização, sem fusão e com filtro por MB;
06	com paralelização, com fusão e com filtro por MB;

Tabela 4.3: *Descrição das configurações de testes utilizados para os testes do sistema AH proposto.*

Todas as configurações de testes foram submetidas à 6000 quadros da base de vídeos oferecida pela *TRECVid* 2009 (câmera 1 - área de desembarque do aeroporto *Grawitch*). Destes vídeos, foram analisados 6000 quadros, esta quantidade foi definida através de julgamento humano. O julgamento considerou a quantidade e direção de objetos em movimento no quadro, tal como a diversidade dos objetos (pessoas, carros, etc). Nas seções subseqüentes são apresentados os testes e resultados considerando a precisão (4.5.1) e o desempenho (4.5.2) do sistema proposto neste trabalho respectivamente.

4.5.1 Precisão

O objetivo desta subseção foi avaliar a efetividade das contribuições implementadas no sistema de identificação do fluxo oposto proposto neste trabalho, em relação aos demais casos de testes. Para a obtenção da efetividade, neste trabalho, considerou-se a precisão na identificação do fluxo oposto de cada caso de teste quando submetidos a um ambiente do mundo real (aeroporto de *Grawitch*).

O fator de precisão utilizado foi o calculado pela fórmula (4.3)[7], que mede a porcentagem de precisão do sistema com base em observações. Ao total são quatro os tipos de observações consideradas pela fórmula: falsos positivos (FP)[8], ii) falsos negativos (FN)[9], iii) negativos verdadeiros (NV) e iv) positivos verdadeiros (PV). Em (4.3), n sumariza todas as observações realizadas durante a execução de um caso de teste, sendo $n = VP + VN + FP + FN$.

$$\frac{VP + VN}{n} \times 100\% \tag{4.3}$$

Para uma alta precisão procurou-se minimizar falsos positivos, falsos negativos, assim como maximizar positivos e negativos verdadeiros, através da supressão das interferências e ruídos, pelo uso do filtro por MB, obtenção dos VM resultantes e da fusão de regiões.

A tabela 4.5 exibe o fator de precisão obtido em cada uma das configurações de testes. As colunas da tabela significam respectivamente: i) ias configurações de testes (descritos na tabela 4.3), ii) o número de falsos positivos encontrados, iii) o número de falsos negativos encontrados, iv) o número de verdadeiros positivos encontrados, v) o número de verdadeiros negativos encontrados, e vi) a precisão atingida em cada caso de teste.

Configurações de testes	FP	FN	VP	VN	Precisão
01	1624	22	330	4024	72,53%
02	291	35	317	5357	94,56%
03	1722	30	322	3926	70,80%
04	1332	37	315	4316	77,18%
05	418	44	308	5233	92,35%
06	370	41	311	5275	93,10%

Tabela 4.4: *Resultados obtidos pela submissão de todos as configurações de testes à base de vídeo disponibilizada pela TRECVid 2009.*

[7]Métrica estatística para avaliar a precisão de um modelo.
[8]Eventos não existentes detectados como existentes.
[9]Eventos não detectados porém existentes.

Nas configurações 1, 3 e 4 (tabela 4.4) ao remover a filtragem do fluxo óptico por MB, uma grande quantidade de interferências e ruídos (iluminação não uniforme, movimentos oscilatórios, etc) são identificados como vetores de movimento. Com isso, o número de FP aumentou aproximadamente 80% em relação às configurações de testes 2, 5 e 6 que não utilizam o filtro por MB. A passagem de ruídos também teve como conseqüência a diminuição do número de FN, devido a algumas áreas em volta dos objetos em movimento que foram suprimidas pelo processo de filtragem por MB, serem novamente consideradas.

Analisando as configurações de teste que utilizaram a filtragem por MB (2, 5 e 6), os fatores que influenciaram o surgimento de FP e FN, e a diminuição na precisão foram:

- *movimentos oscilatórios:* a atualização do MB não foi capaz de suprimir alguns movimentos oscilatórios que não correspondiam aos objetos de interesse (pessoas). Dentre estes movimentos estão as portas fechando e abrindo;
- *texturas:* regiões contidas em objetos em movimentos que possuíam grande diversidade de texturas (como por exemplo roupas e cabelos) geraram vetores de movimento que não representavam o deslocamento real. Devido às texturas estarem contidas nos objetos em movimento, a filtragem por MB não eliminou estes vetores.

Dentre todias as configurações de testes definidas neste trabalho, as que tiveram melhor resultados considerando a precisão na identificação de objetos em fluxo oposto, foram as configuraçoes 2, 5 e 6 com índice de precisão acima de 90% (tabela 4.4). Todas elas utilizaram a filtragem do fluxo óptico através da MB, o que demonstrou que o uso do filtro por MB melhora o processo de filtragem do fluxo óptico.

Quanto as configurações que utilizaram a paralelização (5 e 6), a redução no número de FP da configuração 6 em relação à 5 foi devido ao uso da fusão de regiões (seção 4.3.5.2) que ajustou os VM resultantes como direções divergentes. Porém, a fusão não foi robusta o suficiente quando comparada com a configuração 2 que obteve a maior precisão, sendo que esta utilizou o contexto do quadro completo para a identificação da AH.

4.5.2 Desempenho

Esta subseção avaliou o custo computacional desprendido na execução das configurações da tabela 4.4, focando na técnica de paralelização utilizada neste trabalho. A métrica utilizada para medir o custo computacional, foi o cálculo médio da porcentagem de utilização dos núcleos presentes no conjunto de *hardware* (seção 4.2) utilizados nos testes.

A formulação matemática apresentada em (4.4) mostra como é realizado o cálculo médio da porcentagem de utilização dos núcleos dos processadores, sendo P_i a porcentagem de uso do núcleo do processador (para cada núcleo) no exato momento da $i$ésima identificação e N o número de AH identificadas. Uma observação pertinente é que todas as medidas obtidas foram arredondadas usando o critério de maior valor, por exemplo se foi obtido $60,6\%$ a porcentagem foi arrendondada para 61%.

$$\frac{\sum_{i=0}^{N} P_i}{N} \tag{4.4}$$

A tabela 4.5 exibe os dados coletados durante a submissão das configurações à base de vídeo da *TRECVid*. A tabela está disposta em 5 colunas: i) configurações, ii) porcentagem de utilização do núcleo do processador do primeiro *hardware*, iii) tempo de execução do sistema no primeiro *hardware*, iv) porcentagens de utilização dos núcleos do processador do segundo *hardware*, e v) o tempo de execução do sistema no segundo *hardware*.

Configurações de testes	% primeiro *hardware*	T. de execução	*hardware*	T. de execução
01	97%	1537.201s	97%/23%	1543.459s
02	97%	1624.365s	73%/70%	1638.244s
03	98%	1569.721s	63%/55%	1353.921s
04	96%	1580.220s	66%/67%	1409.369s
05	96%	1714.332s	72%/72%	1501.931s
06	97%	1802.467s	78%/80%	1606.367s

Tabela 4.5: *Porcentagem média de uso dos processadores e seus respectivos tempo de execução.*

Todas as configurações de testes executados no primeiro conjunto de *hardware* obtiveram valores próximos ao 100% de utilização do processador, isto porque a execução das quatro *threads* foram serializadas devido à existência de somente um núcleo no *hardware*.

Para o segundo conjunto de *hardware*, as configurações de testes que não utilizaram a paralelização (1 e 2) tiveram seu processamento escalonado alternadamente entre os dois núcleos, sendo o sistema operacional (seção 4.2) presente no conjunto de *hardware* o responsável por este escalonamento. Isto fica evidente na primeira linha da tabela 4.5 que mostra a utilização de 97% do primeiro núcleo quando o segundo não passou de 23% de utilização.

As configurações 3, 4, 5 e 6 quando executados no segundo conjunto de *hardware* tiveram comportamentos próximos aos esperados: divisão do processamento entre os dois núcleos de forma igual. Porém isso na prática não ocorreu devido à influência de fatores oriundos do *hardware* e *software* utilizados, tais como: concorrência de recursos, serialização, escalonamento de processos, etc, sendo que a análise destes fatores não foi abordada neste trabalho. Como exemplo, temos na figura 4.16 o módulo de fusão de regiões deste trabalho: como a etapa de fusão é serializada é necessário a espera de todas as *threads*, limitando o tempo de processamento e o custo à última *thread* executada.

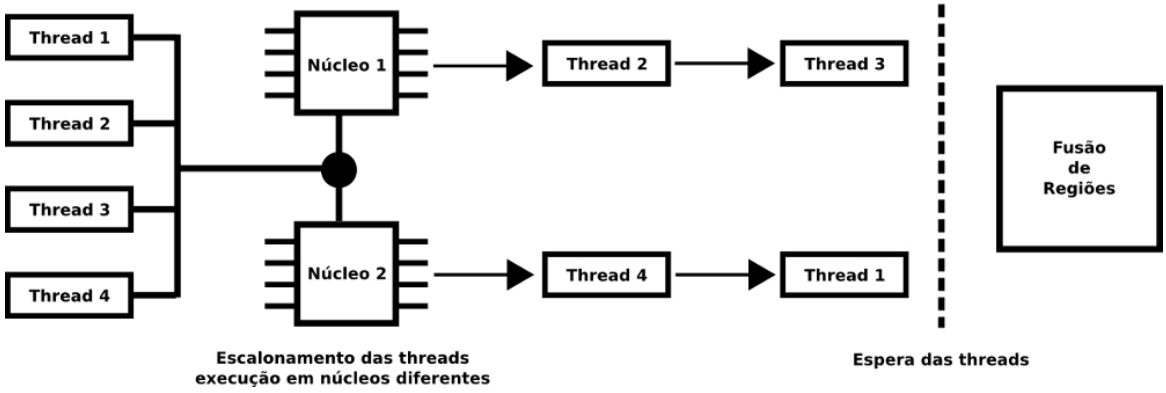

Figura 4.16: *Diagrama do módulo da fusão. Para a prosseguir com a fusão é necessário esperar a execução de todas as threads.*

CAPÍTULO 5

CONCLUSÃO

Métodos capazes de estimar a plenitude do comportamento humano ainda não são uma realidade, porém vários estudos estão sendo realizados para a detecção de atividade humana. Dentre os vários nichos de atuação dos métodos de identificação de AH está a vigilância e o monitoramento que cada vez mais ganha destaque pela crescente complexidade dos ambientes e do aumento do nível de segurança aplicado pelas instituições.

Esta dissertação apresentou, além de uma revisão sobre o problema da identificação da atividade humana destacando os elementos envolvidos no processo (características, abordagens, decisões estatísticas, etc), um método para identificação de uma atividade humana: detecção de fluxo oposto. A abordagem escolhida para a identificação foi a análise angular dos vetores de movimentos através do cálculo *multithread* do fluxo óptico. Esta escolha foi devido ao fluxo óptico retornar a distância e a direção do deslocamento, ideal para o contexto da atividade humana de interesse.

Durante a revisão bibliográfica e o desenvolvimento do sistema foram identificados problemas no cálculo do fluxo óptico, o que gerava resultados não esperados, tal como interferências e ruídos em relação à iluminação e identificação de movimento em direção que não condizia com o deslocamento real dos objetos. Com o objetivo de resolver os problemas encontrados, este trabalho propôs como contribuição um filtro de ruídos através da modelagem de *background* (seção 4.3.3) que se mostrou eficaz para eliminação dos VM que não pertenciam aos objetos em movimento, como demonstrado nos testes de precisão realizados na seção 4.5.1). Também foi proposta a obtenção de vetores de movimento resultantes obtidos pelo agrupamento de característica através do algoritmo *k-means* modificado (seção 4.3.4.2).

Outra contribuição foi a inserção da paralelização (seção 4.3.5) através de *threads* que atualmente é disponibilizada pela maioria dos sistemas operacionais, o qual se apresentaram como alternativa para a divisão do esforço computacional desprendido no cálculo do fluxo óptico e na obtenção dos vetores de movimento resultantes, o que se tornou evidente nos testes de desempenho realizados na seção 4.5.2. Também como contribuição deste trabalho, foi proposta a fusão de regiões para resolver o problema de união dos resultados obtidos em cada região (seção 4.5.1). Contudo, a inserção da fusão acarretou um acréscimo no custo de processamento do sistema (seção 4.5.2).

Como trabalho futuro pretende-se avaliar outras abordagens de particionamento e fusão de áreas, dentre elas a divisão de áreas sobrepostas, formando assim um mosaico, colocando possivelmente pesos em cada área do mosaico, além de avaliar a efetividade do sistema proposto em outros ambientes. Também como trabalho futuro modificar a arquitetura atual do sistema proposto nesta dissertação, para que ao invés calcular o fluxo óptico e após aplicar o filtro MB, calcular o fluxo óptico somente nas regiões de *foreground* extraídas pelo modelagem de *background*. Por fim, ainda como trabalho futuro, desenvolver um sistema de análise de comportamento humano, usando como fonte de informação o método proposto neste trabalho assim como métodos desenvolvidos para identificação de outras atividades humanas.

BIBLIOGRAFIA

[1] V. A. F. Almeida e J. N. C. Árabe. *Introdução a Supercomputação*. Livros técnicos e científicos editora, 1991.

[2] T. K. Ann, A. H. Kan, E. H. Lung, Y. Wei, e W. Junxian. Automated recognition of highly complex human behavior. *ICPR '04: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04)*, volume 4, páginas 327–330, Washington, DC, USA, 2004. IEEE Computer Society.

[3] D. Ayers e M. Shah. Monitoring human behavior from video taken in an office environment. *Image Vision Computer*, 19(12):833–846, 2001.

[4] J. L. Barron, D. J. Fleet, S. S. Beauchemin, e T. A. Burkitt. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994.

[5] Jean Y. Bouguet. Pyramidal implementation of the lucas kanade feature tracker: Description of the algorithm. Jean-Yves Bouguet, 2002.

[6] Lee S. C., C. Huang, e R. Nevatia. Definition, detection and evaluation of meetings events in airport survellance videos. *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, 2008.

[7] R. Cutler, C. Shekhar, J. B. Burns, R. Chellappa, R. Bolles, e L. Davis. Monitoring human and vehicle activities using airborne video. *28th Applied Imagery Pattern Recognition Workshop (AIPR)*, Washington, D.C, 1999.

[8] W.S.P. Fernando, L. Udawatta, e P. Pathirana. Identification of moving obstacles with pyramidal lucas kanade optical flow and k means clustering. *ICIAFS*, páginas 111–117, 2007.

[9] D. J. Fleet e A. D. Jepson. Computation of component image velocity from local phase information. *Int. J. Comput. Vision*, 5(1):77–104, 1990.

[10] R. C. Gonzalez e R. E. Woods. *Digital Image Processing*. Addison-Wesley Longman Publishing Co, Inc, 2001.

[11] S. Hongeng e R. Nevatia. Large-scale event detection using semi-hidden markov models. *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, páginas 1455, Washington, DC, USA, 2003. IEEE Computer Society.

[12] J. E. Hopcroft, R. Motwani, e J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation (3rd Edition)*. Addison-Wesley Longman Publishing Co. Inc., 2006.

[13] B. K. P. Horn e B. G. Schunck. Determining optical flow: a retrospective. *AI*, 59:81–87, 1993.

[14] Y. Huang e T. S. Huang. Model-based human body tracking. *ICPR '02: Proceedings of the 16 th International Conference on Pattern Recognition (ICPR'02)*, volume 1, páginas 10552, Washington, DC, USA, 2002. IEEE Computer Society.

[15] S. S. Intille, J. W. Davis, e Bobick. A. F. Real-time closed-world tracking. *CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, Washington, DC, USA, 1997. IEEE Computer Society.

[16] P. Kaewtrakulpong e R. Bowden. An improved adaptive background mixture model for realtime tracking with shadow detection. *In Proc. 2nd European Workshop on Advanced Video Based Surveillance Systems, AVBS01, VIDEO BASED SURVEILLANCE SYSTEMS: Computer Vision and Distributed Processing*. Kluwer Academic Publishers, September de 2001.

[17] R. Kasturi e R. C. Jain. Dynamic vision. *Computer Vision: Principles*, páginas 469–480. IEEE Computer Society Press, 1991.

[18] L.I. Kuncheva e D. P. Vetrov. Evaluation of stability of k-means cluster ensembles with respect to random initialization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1798–1808, 2006.

[19] L. Li, W. Huang, I. Y. H. Gu, e Q. Tian. Foreground object detection from videos containing complex background. *MULTIMEDIA '03: Proceedings of the eleventh ACM international conference on Multimedia*, páginas 2–10, New York, NY, USA, 2003. ACM.

[20] B. D. Lucas e T. Kanade. An iterative image registration technique with an application to stereo vision. *IJCAI81*, páginas 674–679, 1981.

[21] S. G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989.

[22] R. L. Marques e I. Dutra. Redes bayesianas: o que são, para que servem, algoritmos e exemplos de aplicações. Relatório técnico, Coppe Sistemas - UFRJ, Cidade Universitária, Centro de Tecnologia, Bloco H, Sala 319, Caixa Postal: 68511 CEP: 21941-972, march de 2003.

[23] Y. Nakamura, Y. Kimura, Y. Yu, e Y. Ohta. Mmid: Multimodal multi-view integrated database for human behavior understanding. *FG '98: Proceedings of the 3rd. International Conference on Face & Gesture Recognition*, páginas 540, Washington, DC, USA, 1998. IEEE Computer Society.

[24] O. B. Orhan, J. Hochreiter, J. Poock, Q. Chen, A. Chabra, e M. Shah. University of central florida at trecvid 2008 content based copy detection and surveillance event detection. *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, 2008.

[25] M. Pantic, A. Pentland, A. Nijholt, e T. Huang. Human computing and machine understanding of human behavior: a survey. *ICMI '06: Proceedings of the 8th international conference on Multimodal interfaces*, páginas 239–248, New York, NY, USA, 2006. ACM.

[26] A. Psarrou, M. Walter, e S. Gong. Recognition of human gestures and behaviour based on motion trajectories. *Image and Vision Computing*, 20(5–6):349–358, 2002.

[27] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Readings in speech recognition*, páginas 267–296, 1990.

[28] A. V. Santos, G. P. Dimuro, L. V. Barboza, A. C. R. Costa, R. H. S. Reiser, e M. A. Campos. Probabilidades intervalares em modelos ocultos de markov. *TEMA - Tendências em Matemática Aplicada e Computacional*, 2:361–370, 2006.

[29] J. P. O. Santos, M. P. Mello, e I. T. C. Murari. *Introdução à Análise Combinatória (3^a edição)*. Editora Unicamp, 2002.

[30] M. Shah. Understanding human behavior from motion imagery. *Mach. Vision Appl.*, 14(4):210–214, 2003.

[31] L. G. Shapiro e G. C. Stockman. *Computer vision*. Prentice-Hall, Inc, 2001.

[32] Jianbo Shi e C. Tomasi. Good features to track. *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on*, páginas 593–600, 1994.

[33] Alan F. Smeaton, Paul Over, e Wessel Kraaij. Evaluation campaigns and trecvid. *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, páginas 321–330, New York, NY, USA, 2006. ACM Press.

[34] M. Steinbach, G. Karypis, e V. Kumar. A comparison of document clustering techniques. *KDD Workshop on Text Mining*, 2000.

[35] A. Stergiou, A. Pnevmatikakis, L. Polymenakos, e N. Katasarakis. Detecting single-actor events in video streams. *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, 2008.

[36] A. Strehl e J. Ghosh. Cluster ensembles – a knowledge reuse framework for combining multiple partitions. *Journal on Machine Learning Research*, 3:583–617, 2002.

[37] M. Taj, F. Daniyal, e A. Cavallaro. Event analysis on trecvid 2008 london gatwick dataset. *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, 2008.

[38] W. Tavanapong e J. Zhou. Shot clustering techniques for story browsing. *IEEE Transactions on Multimedia*, 6(4):517–527, 2004.

[39] M. Walter, A. Psarrou, e S. Gong. Learning prior and observation augmented density models for behaviour recognition. *British Machine Vision Conference*, páginas 23–32, 1999.

[40] W. Xiong, J. C. Lee, e D. M. Ip. Net comparison: a fast and effective method for classifying image sequences. *Proceedings of the storage and retrieval for image and video databases III*, páginas 318–328, 1995.

[41] J. Yamato, J. Ohya, e K. Ishii. Recognizing human action in time-sequential images using hidden markov model. *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR '92., 1992 IEEE Computer Society Conference on*, páginas 379–385, 1992.

[42] H. Zhang. Valerie Barr e Zdravko Markov, editors, *FLAIRS Conference*. AAAI Press, 2004.

[43] H. Zhang, A. Kankanhalli, e S. W. Smoliar. Automatic partitioning of full-motion video. *Multimedia Systems*, 1(1):10–28, 1993.

[44] D. Ziou e S. Tabbone. Edge detection techniques – an overview. *Pattern Recognition and Image Analysis*, 8:537–559, 1998.

APÊNDICE A

PROJETOS RELACIONADOS

Neste apêndice são apresentados os trabalhos relacionados à identificação de AH presentes na literatura científica e apresentados na *TRECVid* 2008. O apêndice esta dividido em seções onde os trabalhos estão referenciadas pelo nome do primeiro autor, ano e instituição onde o foi desenvolvido.

Para cada trabalho, uma breve explicação sobre os objetivos e escopo dos trabalhos são descritos. Os métodos são apresentados em uma estrutura padrão (formulários) de modo a permitir uma visão comparativa dos procedimentos adotados em cada trabalho. Abaixo segue um descrição de cada item apresentado no formulário:

- Dados técnicos: exibe o tipo de descritores (seção 2.2) de baixo, médio e alto nível que o método implementa, assim como o uso de contexto e os resultados descritos pelos autores dos respectivos métodos;
- Contribuições: relata as contribuições geradas pelos métodos não somente para a área de AH mas possivelmente para as demais áreas da visão computacional;
- Limitações: descreve as fraquezas dos métodos que foram expostas pelos próprios autores em suas publicações.

A.1 Ayers, D. – 2001 – *Computer Vision Lab School of Eletrical Enginieering and Computer Science* [3].

Sistema de detecção automática de um conjunto de atividades humanas pré-estabelecidas ocorridas em um ambiente controlado (um escritório). A saída deste sistema é composta de uma descrição textual da atividade identificada e os respectivos *keyframes* [43].

- Dados técnicos:
  - descritores de baixo nível: *skin*, *tracking* e detecção de mudança de cenas em imagens coloridas;
  - descritores de nível médio: não possui;
  - descritores de alto nível: máquina de estados finitos modificada para armazenar informações;
  - utiliza contexto: sim;
  - resultados: os autores afirmam que todas as atividades pré-definidas foram identificadas nos testes. A extração de *keyframes* reduz significativamente o número de quadros extraídos durante o reconhecimento de uma atividade.
- Contribuições:
  - sistema de identificação de atividades humanas em ambientes restritos;
  - heurísticas de seleção de *keyframes* em segmentos de vídeos.
- Limitações:
  - a efetividade do sistema depende dos descritores de baixo nível utilizados;
  - a qualidade do conhecimento prévio inserido influência o sistema;
  - o sistema não permite a modelagem de novas atividades automaticamente.

A.2 Cutler, R. – 1999 – *University of Maryland, College Park e SRI International* [7].

Este sistema de vídeo vigilância foi proposto com o objetivo de reconhecer atividades de interesse envolvendo humanos e veículos através de imagens aéreas (em escalas cinza a 20 FPS[1]) obtidas de uma aeronave *Twin Otter*.

- Dados técnicos:
  - descritores de baixo nível: tamanho do objeto, *ground speed*, periodicidade do movimento, área, centróide, *bounding box*, velocidade e quantidade de quadros que o objeto esta presente;
  - descritores de nível médio: objetos seguidos geoalocados e *site-model*;
  - descritores de alto nível: máquinas de estados de finita;
  - utiliza contexto: sim;
  - resultados: os autores não descrevem os resultados.
- Contribuições:
  - um robusto sistema de vigilância aérea submetido à condições realista de operação.
- Limitações:
  - interpretações alternativas tem que ser explicitamente combinadas na máquina de estados.

[1] Abreviação de quadros por segundo.

A.3 Kan, A. H. – 2004 – *Institute for Infoconm Research* [2].

Framework para o reconhecimento automático de atividades humanas complexas. Para efetivar o sistema é utilizando, como ambiente de teste, a identificação de crises aquáticas em uma piscina, em especial é estudado o comportamento de um afogamento.

- Dados técnicos:
  - descritores de baixo nível: centróide, ângulo de orientação, parâmetros de elipse que melhor se encaixa na área, área acumulativa de *pixels*, média de saturação de cor e o nível de energia das bordas;
  - descritores de nível médio: índice de submersão, índice de atividade, índice de respingo, velocidade transicional e a postura;
  - descritores de alto nível: *Generalized Reduced Multivariate Polynomial Network*, uma modificação do *Functional Link Network* (FLN);
  - utiliza contexto: sim;
  - resultados: três tipos de comportamento foram considerados, afogamento, nado normal e ondulações da água, sendo que os dois últimos foram classificados corretamente como não afogamento. Também é demonstrado através do gráfico da curva ROC que o *Generalized Reduced Multivariate Polynomial Network* é mais eficientemente que outras aproximações: *Optimal Weighting Method* (OWN) e a *Feedforward Neural Network* (FNN).
- Contribuições:
  - *framework* para automatizar a detecção de atividades humanas;
  - um método para fusão de características com o objetivo de inferir atividade de afogamento;
  - uma série de descritores foram propostos para reter mais conteúdo semântico das atividades.
- Limitações:
  - a sensibilidade em relação a obtenção de descritores de baixo nível podem comprometer as camadas superiores;
  - a introdução de uma camada intermediária ao processo pode gerar um acumulo de imprecisão;
  - devido os descritores de médio nível serem determinados de acordo com aplicação, tornasse necessário a definições de novos descritores no caso de uma possível troca de contexto.

A.4 Yamato, J. – 1992 – *NTT Human Interface Laboratories* [41].

É proposto um método para reconhecimento de atividades humanas onde é usado como descritores de alto nível os HMM. Como "caso de uso" foi realizada a identificação de um conjunto de movimentos de um tenista: *forehand stroke*, *backhand stroke*, *forehand volley*, *backhand volley*, *smash* e *service*.

- Dados técnicos:
  - descritores de baixo nível: vetor de característica *mesh*;
  - descritores de nível médio: um símbolo, criado pela quantização do vetor de características;
  - descritores de alto nível: HMM;
  - utiliza contexto: não;
  - resultados: é constatado pelos autores que o desempenho da aproximação é diretamente proporcional aos dados de treino. Quanto maior a diversidade e quantidade de dados para o treino melhor será a precisão. Quando os dados usados no treino e no reconhecimento são os mesmo o desempenho fica acima de 90% e quando são todos diferentes o desempenho cai significativamente.
- Contribuições:
  - extractor de área humana (eliminador de fundos complexos);
  - *tracking* a fim de identificar o tenista dentro da cena;
  - aproximação baseada em HMM para reconhecer atividades humanas de interesse.
- Limitações:
  - sensibilidade da característica *mesh* influência a fase de treino e a de reconhecimento.

A.5 Psarrou, A. – 2002 – *Harrow Scholl Computer Science of University of Westminster e Department of Computer Science of University of London* [26].

Framework com o objetivo de reconhecer atividades humanas. Este reconhecimento é baseado no aprendizado prévio usando HMM e modelos de densidade aguçados pela observação visual. Para validação são usados como seqüencia de testes o reconhecimento: de caminhadas entre diferentes áreas de um escritório, de gestos comunicativos e o de símbolos realizados por gestos.

- Dados técnicos:
  - descritores de baixo nível: centróide, deslocamento do centróide em dois quadros consecutivos e *skin*;
  - descritores de nível médio: não tem;
  - descritores de alto nível: HMM e modelos de densidade aguçados pela observação visual;
  - utiliza contexto: sim;
  - resultados: nos testes os autores mostram que os algoritmos de *observação aguçada* e *não-aguçada* tiveram uma taxa de reconhecimento superior ao algoritmo de *condensação* e também obtiveram o custo computacional inferior ao obtido pelo algoritmo de *condensação*.
- Contribuições:
  - *framework* para reconhecer atividades e símbolos e gestos;
  - melhoramento no algoritmo de condensação (observação aguçada e não-aguçada) que diminui o custo computacional e aumenta a taxa de reconhecimento.
- Limitações:
  - não são descritas limitações pelo autores.

A.6 Hongeng, S. – 2003 – *Institute for Robotics and Intelligent System, University of Southern California* [11].

É proposta uma aproximação para reconhecer atividades em vídeos. Essas atividades são compostas de eventos são descritos como *eventos primitivos* (EP) e *eventos compostos* (EC) (formados por um conjunto de PE). Para modelar os PE e os CE são usados respectivamente as redes de *Bayesian* e semi-HMM modelada em um AFD modificado. Esta modificação além de amenizar a interferência de ruído nas cadeias de *Markov* padrão também oferece uma menor complexidade ($O(NT)$) que os semi-HMM originais ($O(NT^2)$).

- Dados técnicos:
  - descritores de baixo nível: forma e trajetória obtidos por *tracking*;
  - descritores de nível médio: redes de *Bayes*;
  - descritores de alto nível: semi-HMM;
  - utiliza contexto: sim;
  - resultados: no referido artigo, os teste realizados com atividades complexas alcançaram a taxa de 96.7%, porém os autores também afirmam que a avaliação completa do sistema necessita de testes com uma base maior de vídeos.
- Contribuições:
  - modelador de eventos e um método de reconhecimento usando uma modificação dos semi-HMM integradas com redes de *Bayes*;
  - algoritmo eficiente para realizar inferências com os modelos.
- Limitações:
  - ruídos gaussiano afetam diretamente os EP e podem comprometer o sistema;
  - número de objetos em movimento afetam o tempo de computação;
  - contexto da cena afeta o tempo de computação;
  - eventos na biblioteca que são de interesse afetam o tempo de computação.

A.7 Lee, S. C. – 2008 – *University of Southern California* [6].

Este trabalho foi apresentado na *TRECVid* 2008. Tem como foco a identificação de pessoas se reunindo (*meeting*), definido no apêndice B.10, em um ambiente complexo e com uma grande variedade de pessoas. Para isso é realizado o *track* de indivíduos e baseado em suas trajetórias é avaliado se *meeting* esta ocorrendo ou não. Esta técnica pode ser caracterizada como *bottom-up* (seção 2.1.1) devido utilizar o empilhamento de características, *track* usada como característica base.

- Dados técnicos:
  - descritores de baixo nível: não são descritos pelos autores;
  - descritores de nível médio: trajetória de indivíduos obtidas por *tracking*;
  - descritores de alto nível: decisão tomada com auxílio de regras para definir *meeting*;
  - utiliza contexto: sim;
  - resultados: de 17 trechos de vídeos pertencentes a base *TRECVid* 2008, os autores determinaram 22 *meeting* deste total o método foi capaz de identificar 20.
- Contribuições:
  - método de identificação de eventos de reunião de pessoas (apêndice B.10) através de *tracking*.
- Limitações:
  - baixa resolução dos indivíduos devido a distância no campo de visão da câmera pode gerar degradação no *tracking* com isso gerar inconsistência na detecção;
  - indivíduos passando uns pelos outros podem gerar a falsa interpretação de *meeting*.

A.8 Stergiou, A. – 2008 – *Athens Information Technology* [35]

Nesta trabalho que também foi apresentada na *TRECVid* 2008, teve como proposta a identificação de eventos fluxo oposto (seção 3.2.1), pessoas correndo (apêndice B.3) e a permanência à frente do elevadores (apêndice B.15), sendo que o ambiente de ação é o aeroporto de *Gatwick* na Inglaterra. Para identificar os dois primeiros eventos foram usados vetores de movimento (seção 2.2.2.1) como característica para identificação, para o terceiro caso é utilizado *blobs* [35] e segmentação adaptativa de *foreground* [35].

- Dados técnicos:
  - descritores de baixo nível: utiliza textura e valores do espaço de uma região de *pixels*;
  - descritores de nível médio: direção e magnitude do vetores de movimento, e *tracking*;
  - descritores de alto nível: segmentação adaptativa de *foreground*, e *blobs*;
  - utiliza contexto: sim;
  - resultados: não são relatados pelos autores.
- Contribuições:
  - método para reconhecer fluxo oposto e pessoas correndo a partir de vetores de movimento e regras de identificação;
  - método para reconhecer pessoas entrando em elevadores baseado em *blobs*, segmentação adaptativa de *foreground* e contagem de indivíduos.
- Limitações:
  - dificuldade para identificar os eventos quando o individuo esta sozinho e não tem sua localização prévia estabelecida;
  - dificuldade para identificar os eventos quando o ambiente é com um número excessivo de indivíduos e objetos em movimento.

A.9 Taj, M. – 2008 – *Queen Mary, University of London* [37]

Este estudo apresentado na *TRECVid* 2008, foi desenvolvido para análise de eventos voltados a vigilância em ambientes reais. Foram analisados características de baixo nível e de alto nível, dentre elas destaque para vetores de movimento (seção 2.2.2.1), detecção de mudanças entre quadros e detecção de pedestres para o reconhecimento dos seguintes eventos: pessoa correndo (apêndice B.3), permanência à frente do elevador (apêndice B.15) e fluxo oposto (seção 3.2.1).

- Dados técnicos:
  - descritores de baixo nível: valores do espaço de cor de cada *pixel*;
  - descritores de nível médio: magnitude e direção normalizada do vetor de movimento, suavização temporal, detecção de mudanças, e *bound box*;
  - descritores de alto nível: não tem;
  - utiliza contexto: sim;
  - resultados: os resultados obtidos foram satisfatórios segundo descrito pelos autores.
- Contribuições:
  - método capaz de detectar e identificar ações humanas (pessoa correndo, pessoas entrada em um elevador, e fluxo oposto).
- Limitações:
  - pessoas sobre veículos são detectadas como pessoas correndo;
  - movimentos randômicos de pessoas em frente aos elevadores podem gerar falsos positivos;
  - dificuldade para identificar os eventos quando o ambiente é com um número excessivo de indivíduos e objetos em movimento.

A.10 Orhan, O. B. – 2008 – *University of Central Florida* [24]

O trabalho realizado por Orhan, O. B *et al.* [24] teve como abrangência mais de um tópico da *TRECVid* 2008, dentre eles a detecção de eventos de vigilância. As atividades humanas de interesse abordadas pelos autores para a detecção de eventos de vigilância foram: pessoas correndo (apêndice B.3), largando objetos (apêndice B.5), fluxo oposto (seção 3.2.1) e tirando uma foto (apêndice B.16).

- Dados técnicos:
  - descritores de baixo nível: valores do espaço de cor de cada *pixel*;
  - descritores de nível médio: magnitude e direção do vetor de movimento, *tracking*, agrupamento de características, e diferença de intensidade;
  - descritores de alto nível: não tem;
  - utiliza contexto: sim;
  - resultados: para pessoas correndo somente 45% dos eventos foram detectados com baixa precisão. Para a atividade de largar objetos obteve uma precisão de aproximadamente 30%. Para o fluxo oposto, o sistema obteve precisão extremamente baixa (menos que 1%). Por fim, para a atividade humana de tirar foto, segundo a *TRECVid* tem baixa precisão.
- Contribuições:
  - método robusto para identificação de pessoas correndo em um ambiente fechado, que trata movimento cíclicos e oclusões;
  - método para identificação em tempo real de pessoas largando objetos em um ambiente fechado;
  - método para identificação de fluxo oposto em um ambiente fechado;
  - método para identificação de pessoas tirando foto em um ambiente fechado;
- Limitações:
  - pessoas sentando pode ser interpretado pelo método como uma pessoa largando um objeto;
  - picos de intensidades podem ser interpretados pelo método como um *flash* proveniente de uma máquina fotográfica.

APÊNDICE B

EVENTOS DE VIGILÂNCIA

Neste apêndice serão descritos todos os eventos de vigilância utilizados como nicho para os trabalhos submetidos na TRECVid 2008 e que também serão usados na edição 2009. São no total de dezessete eventos, contudo somente 16 serão evidenciados devido ao fato que o evento E.20 já ter sido detalhado na seção 3.2.1.

B.1 E01 – Porta abrindo e fechando

Este evento é indicado como opcional pela TRECVid 2008, e é constituído basicamente por uma porta abrindo e após um período indeterminado de tempo a mesma porta fechando. O evento começa no momento mais próximo do inicio da abertura da porta, e termina quando a porta esta completamente fechada.

B.2 E04 – Uso do caixa eletrônico

Também indicado como opcional, este evento consiste da utilização de um caixa eletrônico. Inicia quando uma pessoa se aproxima do caixa eletrônico e fica na eminência de inserir um cartão, e termina quando a mesma pessoa começa a se afastar da máquina (possivelmente finalizando o uso do caixa). O fator de precisão para este evento é o momento exato da inserção do cartão (para o inicio), e o momento exato de afastamento do caixa por parte do usuário. Este evento não pretende identificar se usuário terminou corretamente o uso do caixa eletrônico.

B.3 E05 – Pessoa correndo

Evento composto por uma pessoa correndo nas dependências do aeroporto de *Gatwick*. O evento inicia imediatamente após a primeira visualização duma pessoa correndo e finaliza na ultima vez que é possível ver o mesmo individuo correndo. Não abrange pessoas que já entram correndo no campo de visão das câmeras, e problemas de oclusão oriundos de pessoas, objetos e obstáculos.

B.4 E06 – Colocando o celular próximo à orelha

É caracterizado pelo ato de uma pessoa colocar o celular próximo a uma de suas duas orelhas. Começa no momento exato em que uma pessoa qualquer, localizada nas dependências do aeroporto, inicia o movimento de deslocamento do celular em direção à sua própria cabeça, e o termina quando o deslocamento é finalizado. Este evento não pretende abranger o caso em que o sujeito já entra em cena com o celular junto a sua orelha, e sim somente o ato do deslocamento do celular.

B.5 E08 – Largando um objeto

Alguma pessoa derruba, coloca no chão ou em algum outro lugar (não são considerados partes do seu corpo ou de outra pessoa) um objeto qualquer. O delimitador inicial do evento é o último momento em que uma pessoa qualquer esta com um objeto em sua posse, e o delimitador final é exatamente após a mesma pessoas se encontrar sem o mesmo objeto. Neste evento não são considerados como objetos seres humanos, exclui-se então, por exemplo, os casos como o ato de colocar um bebê em um carrinho.

B.6 E09 – Pegando um objeto

Este evento é o caso oposto do descrito na subseção B.5, sendo que aqui ao invés de largar o objeto o mesmo é pego. O inicio do evento é determinado pela ultima vez em que uma pessoa qualquer não esta de posse de um objeto, e o término do evento é o imediatamente após a mesma pessoa esta com um objeto qualquer em sua posse. Um observação descrita nas definições da TRECVid 2008 é que este evento não se aplica quando os objetos identificados são comidas ou utensílios utilizados em refeições (talheres, guardanapos, etc). Também neste caso seres humanos não são considerados objetos. Existe um caso especial em que após um momento de oclusão da pessoa em questão, a mesma aparece com um objeto. Tal situação pode ser caracterizada como o caso de transferência de objetos (seção B.13) porém continua sendo o evento E09.

B.7 E10 – Identificação de vestimentas

Mais um evento opcional de simples detecção que compreende reconhecimento de cores e formas, e que é usado como suporte para outros eventos. Consiste em detectar quando alguém relacionado à segurança (neste cenário são uniformizados com vestimentas nas cores amarela ou verde) aparece no campo de visão das câmeras. O delimitador inicial é exatamente após a aparição de uma pessoa com as características de interesse, e o delimitador final é o último momento em que a pessoa é visível pelas câmeras do aeroporto.

B.8 E11 – Sentando

Opcional, este evento é caracterizado por uma pessoa sentando em algum lugar ou objeto dentro das áreas monitoradas pelas câmeras do aeroporto. Começa no exato momento em que a pessoa inicia o movimento para sentar, e termina logo após a mesma pessoa finalizar o movimento e encontrar-se totalmente sentada. Alguns deslocamentos sobre a área do local de acento são permitidos, desde que não hajam movimentos verticais de postura.

B.9 E12 – Levantando

Este evento pode ser interpretado como um processo complementar do evento descrito na seção B.8, sendo este também um caso opcional. Inicia no momento em que uma pessoa sentada em uma área qualquer inicia o movimento para levantar (não são considerados deslocamentos laterais, os quais são abordados no evento E11), e finaliza no momento em que a mesma pessoa esta em uma posição ereta.

B.10 E14 – Reunião

É composto por um grupo de pessoas andando em direção a outro grupo, parando em frente a ele e então realizando algum tipo de comunicação (seja ela verbal ou gestual). O evento começa quando acontece a primeira comunicação entre um membro de um grupo com um membro de outro grupo, sendo que cada grupo pode ter uma ou mais pessoas, e termina quando os dois grupos se afastam após a comunicação ter sido realizada.

B.11 E15 – Separação

A partir de um grupo em movimento ou parado (sentado, em pé ou ambos) com duas ou mais pessoas comunicando-se, uma ou mais pessoas se separam e deixam o grupo e o campo de visão das câmeras. O momento inicial é quando os membros do grupo de pessoas estão mais próximo um dos outros, e o final é o momento que ao menos um membro deixa o grupo e o campo de visão da câmera (quadro). Pode haver o caso de ocorrer simultaneamente o evento de *separação* e *reunião* (B.10), como por exemplo, um pessoa se afasta do grupo enquanto outra se aproxima.

B.12 E16 – Abraçando

Evento que é definido pelo ato de uma pessoa colocar um ou ambos os braços em volta de alguma parte do corpo de outra pessoa. Começa no momento em que duas pessoas qualquer não tem contato físico antes do abraço, e termina quando as mesmas não tem mais contato físico após o abraço. Este evento não pretende contemplar o caso quando as pessoas já entram abraçados no campo de visão da câmera, e também não pretende tratar casos de oclusão.

B.13 E17 – Transferência de objeto

Este evento opcional é descrito quando uma pessoa passa/transfere um objeto de sua posse para uma outra pessoa qualquer. Começa no exato momento do inicio da transferência do objeto, e termina imediatamente após o objeto ser totalmente transferido. Este evento não é caracterizado quando o objeto é transferido para outra pessoa indiretamente, isto é, o objeto é colocado em outra localização e a outra pessoa o pega. Neste caso temos a composição de dois eventos: *largar um objeto* (B.5) e *pegar um objeto* (B.6). Outra observação é que neste caso, seres humanos também não são considerados objetos.

B.14 E18 – Pessoa apontando

Diretamente ligado a gestos e posturas, é caracterizado por uma pessoa apontando com seus dedos ou braços para algum lugar. Este evento é delimitado inicialmente no exato momento que uma pessoa desloca seus dedos ou braços para uma posição de apontamento, e tem seu delimitador final imediatamente após a mudança de posição dos dedos ou braço.

B.15 E19 – Permanência à frente do elevador

O seguinte cenário compõem este evento: uma ou mais pessoas esperam na frente de um elevador, este chega abre e fecha sua porta e o número de pessoas continua inalterado no campo de visão da câmera. O momento inicial para detecção do evento é quando a porta do elevador abre com uma ou mais pessoas esperando a sua frente, e o final é quando o elevador fecha sua porta e as pessoas continuam presente no ambiente. O evento opcional *porta abrindo e fechando* (B.1) pode ser utilizando como auxílio para identificar quando o elevador chega.

B.16 E21 – Tirando uma foto

Este evento é caracterizado por uma pessoa qualquer tirando uma foto nas dependências do aeroporto. O evento inicia no momento em que uma pessoa mantém uma câmera em uma posição fixa para dispará-la, e termina no exato momento que a mesma pessoa afasta a câmera da posição após a fotografia. Uma observação feita pela TRECVid 2008 é que este evento não distingue o tipo de câmeras, podendo esta ser uma máquina digital, um celular, etc.