Greice Martins de Freitas

Rastreamento de objetos em vídeos e separação em classes

Dissertação de Mestrado apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: Engenharia de Computação.

Orientador: Clésio Luis Tozzi

Banca Examinadora:
Prof. Dr. Clésio Luis Tozzi - UNICAMP
Prof. Dr. José Mario De Martino - UNICAMP
Prof. Dr. Maurício Galo - Unesp

Campinas, SP
2010

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE - UNICAMP

F884r

Freitas, Greice Martins de
Rastreamento de objetos em vídeos e separação em classes / Greice Martins de Freitas. – Campinas, SP: [s.n.], 2010.

Orientador: Clésio Luis Tozzi.
Tese (mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Rastreamento automático. 2. Subtração de fundo. 3. Kalman, Filtragem de. 4. Somas gaussianas. 5. Processamento de imagens. I. Tozzi, Clésio Luis. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Título em Inglês:	Tracking of objects in videos and separation in classes
Palavras-chave em Inglês:	Automatic tracking, Background subtraction, Kalman filtering, Gaussian sums, Image processing
Área de concentração:	Engenharia de Computação
Titulação:	Mestre em Engenharia Elétrica
Banca Examinadora:	José Mario De Martino, Maurício Galo
Data da defesa:	11/06/2010
Programa de Pós Graduação:	Engenharia Elétrica

COMISSÃO JULGADORA - TESE DE MESTRADO

Candidata: Greice Martins de Freitas

Data da Defesa: 11 de junho de 2010

Título da Tese: "Rastreamento de Objetos em Vídeos e Separação em Classes"

Prof. Dr. Clésio Luis Tozzi (Presidente): _____

Prof. Dr. Mauricio Galo: _____

Prof. Dr. José Mario De Martino: _____

Resumo

A crescente utilização de câmeras de vídeo para o monitoramento de ambientes, auxiliando no controle de entrada, saída e trânsito de indivíduos ou veículos tem aumentado a busca por sistemas visando a automatização do processo de monitoramento por vídeos. Como requisitos para estes sistemas identificam-se o tratamento da entrada e saída de objetos na cena, variações na forma e movimentação dos alvos seguidos, interações entre os alvos como encontros e separações, variações na iluminação da cena e o tratamento de ruídos presentes no vídeo.

O presente trabalho analisa e avalia as principais etapas de um sistema de rastreamento de múltiplos objetos através de uma câmera de vídeo fixa e propõe um sistema de rastreamento baseado em sistemas encontrados na literatura. O sistema proposto é composto de três fases: identificação do *foreground* através de técnicas de subtração de fundo; associação de objetos quadro a quadro através de métricas de cor, área e posição do centróide - com o auxílio da aplicação do filtro de Kalman - e, finalmente, classificação dos objetos a cada quadro segundo um sistema de gerenciamento de objetos.

Com o objetivo de verificar a eficiência do sistema de rastreamento proposto, testes foram realizados utilizando vídeos das bases de dados PETS e CAVIAR. A etapa de subtração de fundo foi avaliada através da comparação do modelo *Eigenbackground*, utilizado no presente sistema, com o modelo Mistura de Gaussianas, modelo de subtração de fundo mais utilizado em sistemas de rastreamento. O sistema de gerenciamento de objeto foi avaliado por meio da classificação e contagem manual dos objetos a cada quadro do vídeo. Estes resultados foram comparados à saída do sistema de gerenciamento de objetos.

Os resultados obtidos mostraram que o sistema de rastreamento proposto foi capaz de reconhecer e rastrear objetos em movimento em sequências de vídeos, lidando com oclusões e separações, mostrando adequabilidade para aplicação em sistemas de segurança em tempo real.

Palavras-chave: Rastreamento automático, Subtração de fundo, Filtragem de Kalman, Mistura de Gaussianas, Processamento de Imagens.

Abstract

There are immediate needs for the use of video cameras in environment monitoring, which can be verified by the task of assisting the entrance, exit and transit registering of people or vehicles in a area. In this context, automated surveillance systems based on video images are increasingly gaining interest. As requisites for these systems, it can be identified the treatment of entrances and exits of objects on a scene, shape variation and movement of followed targets, interactions between targets (such as meetings and splits), lighting variations and video noises.

This work analyses and evaluates the main steps of a multiple target tracking system through a fixed video camera and proposes a tracking system based on approaches found in the literature. The proposed system is composed of three steps: foreground identification through background subtraction techniques; object association through color, area and centroid position matching, by using the Kalman filter to estimate the object's position in the next frame, and, lastly, object classification according an object management system.

In order to assess the efficiency of the proposed tracking system, tests were performed by using videos from PETS and CAVIAR datasets. The background subtraction step was evaluated by means of a comparison between the Eigenbackground model, used in the proposed tracking system, and the Mixture of Gaussians model, one of the most used background subtraction models. The object management system was evaluated through manual classification and counting of objects on each video frame. These results were compared with the output of the object management system.

The obtained results showed that the proposed tracking system was able to recognize and track objects in movement on videos, as well as dealing with occlusions and separations, and, at the same time, encouraging future studies in order for its application on real time security systems.

Keywords: Automatic tracking, Background subtraction, Kalman filtering, Mixture of Gaussians, Image processing.

Agradecimentos

Ao Prof. Dr. Clésio Luis Tozzi - pelo apoio, confiança e oportunidade de desenvolver esta pesquisa possibilitando trilhar novos caminhos na minha vida profissional.

À Faculdade de Engenharia Elétrica da UNICAMP, pela oportunidade de realizar este trabalho.

À CAPES, pelo apoio financeiro.

Aos professores que tive ao longo desta jornada, pela formação acadêmica que me proporcionaram.

À Carmen da secretaria do DCA e aos demais funcionários da FEEC, pela atenção, presteza e dedicação.

Aos amigos e colegas do LCA pela convivência e auxílio.

Aos meus demais amigos, pela camaradagem e momentos de alegria, sem os quais seria muito difícil trilhar esse caminho.

Ao meu namorado André pelo amor, carinho, apoio, compreensão e paciência.

A meus pais e irmãos pela dedicação, amor e apoio nas horas difíceis.

Dedico aos meus pais
Fernando e Tania
irmãos Roberto e Rafael
e namorado André.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Símbolos

$\|x\|$	-	Módulo de x
A^T	-	Transposta da matriz A
A^{-1}	-	Inversa da matriz A
$A_{i,j}$	-	Elemento da matriz A na linha i e coluna j
$E[\cdot]$	-	Esperança de uma variável aleatória
$[AB]$	-	Concatenação das matrizes A e B
$\eta(x, \mu, \sigma^2)$	-	Função de densidade Gaussiana com média μ e variância σ^2
$X \sim N(\mu, \sigma^2)$	-	Variável aleatória X com distribuição normal de média μ e variância σ^2
$P(x)$	-	Probabilidade de x
$\bar{x}$	-	Vetor médio
$\Omega = (\bar{x}, U, \Lambda, N)$	-	Autoespaço com matriz de autovetores U, vetor de autovalores Λ e número de observações N
Δx	-	Variação da variável x de um quadro do vídeo para o seguinte

Trabalhos Publicados Pelo Autor

1. G.M. Freitas, C.L. Tozzi. "Rastreamento de Objetos por Eigenbackground e Separação em Classes". *Workshop de Visão Computacional* (WVC'2009), São Paulo, Brasil, Setembro 2009.

2. G.M. Freitas, C.L. Tozzi. "Multiple state-based video tracking for surveillance applications". *Brazilian Symposium on Computer Graphics and Image Processing* (Sibgrapi'2009), Rio de Janeiro, Brasil, Outubro 2009.

Capítulo 1

Introdução

A facilidade da obtenção de câmeras de vídeo e o barateamento da instalação das mesmas popularizaram este instrumento, que hoje pode ser facilmente encontrado em caixas eletrônicos, shoppings, supermercados, carros, ruas entre outros lugares, com a finalidade de detectar e rastrear movimento de pessoas, carros, animais, etc. Até mesmo câmeras mais simples como *webcams* podem ser utilizadas como instrumento de captura de movimento para alimentar aplicativos como jogos [1] ou programas que produzem efeitos e animações em vídeos através da detecção de faces [2]. Neste contexto, a busca por sistemas de rastreamento de movimento através de vídeos tem aumentado nos últimos anos, tanto na área acadêmica quanto na área comercial, buscando-se por sistemas robustos que atuem em tempo real.

Existem três principais aplicações para sistemas de rastreamento através de vídeos: *análise*, que abrange aplicações como avaliação de desempenho de atletas [3, 4], diagnóstico de pacientes com disfunções na marcha [5, 6] e aplicações na industria automobilística, para a verificação da atenção do motorista [7], previsão de colisão [8, 9] e sistemas de auxilio ao motorista na tarefa de estacionar o veículo [10]; *controle*, relacionado principalmente à indústria de entretenimento com interfaces para jogos [11] e interfaces Humano-Computador [12, 13]; e, finalmente, *segurança*, sendo este um problema clássico do rastreamento que inclui o monitoramento de ambientes, objetivo do presente estudo.

Sistemas de monitoramento de ambientes através de rastreamento são aplicados principalmente em locais de grande fluxo de carros e/ou pessoas, tais como plataformas de trens, faixas de pedestres, aeroportos, estacionamentos e estradas, com o principal objetivo de analisar as ações e atividades dos alvos seguidos, entrada e saída dos mesmos no ambiente monitorado ou detectar atividades indesejadas, auxiliando no trabalho e na tomada de decisões por parte dos profissionais de segurança, uma vez que a eficiência dos mesmos está condicionada a fatores como fadiga e distração.

Um sistema de rastreamento por vídeo é composto de três etapas principais: identificação e seg-

mentação de objetos em movimento, ou *foreground*, rastreamento do objeto ao longo do tempo e classificação dos objetos quanto à sua natureza ou ações executadas. A etapa de segmentação é geralmente realizada através da Subtração de Fundo que, por seu baixo custo computacional, permite sua aplicação em tempo real. Técnicas de Subtração de Fundo consistem na formação de um modelo do fundo da cena, ou *background*, que é "subtraído" quadro a quadro da sequência de vídeo. A diferença entre o quadro atual e o modelo de *background* resulta nas regiões que não pertencem ao fundo, que podem ser tanto um objeto em movimento quanto ruídos indesejáveis. Desta forma, realizar uma Subtração de Fundo robusta significa reduzir estes ruídos sem comprometer a identificação do *foreground*.

Dentre os principais fatores que dificultam a segmentação através da Subtração de Fundo pode-se citar a variação de iluminação do ambiente, seja ela repentina, como acender as luzes de uma sala, ou gradual, como o entardecer num ambiente externo, sombras, oscilações da câmera e objetos em movimento periódico, como uma árvore balançando ou ondas do mar. Também é desejável que o sistema adapte-se a variações no *background*, incorporando elementos do *foreground* caso este permaneça estático por um certo período de tempo.

Diversos modelos de subtração de fundo são propostos na literatura, dentre os quais pode-se citar a Mistura de Gaussianas [14] que, baseada em distribuições gaussianas, é um dos métodos mais utilizados em aplicações de rastreamento para uma câmera estática e o *Eigenbackground* [15], método inspirado na classificação de padrões através da Análise de Componentes Principais, que tem como objetivo diminuir a dimensionalidade do problema. Cada método possui suas vantagens e limitações que dependem do tipo do ambiente e da aplicação. Desta forma, é fundamental que na construção de um sistema de rastreamento seja selecionado o método que melhor atenda à aplicação.

A etapa de rastreamento consiste em gerar a trajetória dos alvos ao logo do tempo. Para tal, os objetos identificados a cada quadro são associados aos objetos identificados no quadro anterior. Esta associação permite identificar a posição e espaço ocupado pelo alvo em cada quadro do vídeo. Associar objetos quadro a quadro não é uma tarefa trivial, uma vez que o alvo pode mudar de direção, velocidade ou sofrer variações topológicas, deformações e variações de tamanho, ainda que tais mudanças sejam graduais.

Estimadores lineares, com destaque para o Filtro de Kalman [16], têm sido vastamente utilizados em sistemas de rastreamento para auxiliar na associação de objetos, sendo caracterizados por gerar uma previsão do estado do alvo baseado em seus estados anteriores. Sistemas de rastreamento que utilizam filtro de Kalman podem ser facilmente encontrados na literatura. Tais sistemas empregam o filtro para estimar a posição do alvo no quadro seguinte. As posições estimadas são comparadas às posições medidas no quadro seguinte e a associação é realizada se estimativa e medição forem suficientemente próximas, sendo a posição do objeto associado ao alvo utilizada para atualizar o

filtro.

Entretanto, a associação baseada apenas na posição dos objetos pode gerar ambiguidades: duas posições medidas podem ser relacionadas a uma estimativa ou uma estimativa pode não ser associada a alguma medida. Desta forma, é preciso estabelecer outros critérios adicionais de associação, baseados, por exemplo, na utilização de características do alvo como área, cor, forma, inclinação etc. Além disso, deve-se considerar que os alvos podem interagir uns com os outros e com o fundo, resultando em oclusões totais ou parciais, oclusões por elementos da cena, uniões, separações, entrada e saída de alvos. Assim, o sistema de rastreamento deve lidar com estas mudanças de estado, retomando o rastreamento dos alvos após estas interações.

Finalmente, a etapa de classificação diz respeito à aplicação do sistemas de rastreamento, identificando os alvos de interesse e analisando seus movimentos e suas interações, sendo os objetivos desta etapa dependentes da finalidade do sistema. Num sistema de monitoramento de atividades humanas, por exemplo, a etapa de classificação normalmente abrange a identificação de pessoas dentre os alvos seguidos e reconhecimento de suas atividades como andar, correr, deixar ou retirar objetos da cena.

O objetivo principal do presente trabalho é analisar e avaliar as principais etapas de um sistema de rastreamento de múltiplos objetos através de uma câmera de vídeo fixa e propor um sistema de rastreamento baseado em abordagens encontradas na literatura. Este sistema é composto de três fases principais: identificação do *foreground* através de técnicas de subtração de fundo; associação de objetos quadro a quadro através da posição do centróide com o auxílio da aplicação do filtro de Kalman, métricas de cor e área e, finalmente, classificação dos objetos de cada quadro segundo um sistema de gerenciamento de objetos, baseado no sistema desenvolvido por Lei *et al* [17].

O sistema de rastreamento proposto visa lidar com alvos que eventualmente podem sofrer interferências como oclusões por outros alvos ou por partes da cena, falhas na identificação do *foreground* e entrada e saída de outros alvos na cena. Neste sentido, o sistema de gerenciamento de objetos ganha destaque por manter informações relativas ao estado de cada objeto seguido durante e antes das interferências.

Foram considerados na avaliação dos resultados dois aspectos do sistema de rastreamento proposto: o modelo de subtração de fundo e o sistema de gerenciamento de objetos propostos. A avaliação do modelo de subtração de fundo foi realizada através da analise de sensibilidade de parâmetros dos modelos *Eigenbackground* e Mistura de Gaussianas. O desempenho de cada modelo foi avaliado segundo índices de precisão e retorno, que indicam a qualidade da identificação do *foreground*, além de resultados visuais. Com base nos resultados obtidos, o sistema de subtração de fundo *Eigenbackground* foi selecionado para compor a etapa de identificação do *foreground* do sistema de rastreamento. O desempenho do sistema de gerenciamento de objetos, por sua vez, foi avaliado através da contagem do número de objetos corretamente classificados em cada quadro dos vídeos testados.

Este trabalho está organizado em sete Capítulos: o Capítulo 2 apresenta uma breve revisão bibliográfica de modelos de subtração de fundo e uma descrição dos dois modelos estudados: *Eigenbackground* e Mistura de Gaussianas. O Capítulo 3 apresenta um estudo comparativo entre os dois modelos de subtração de fundo estudados, o Capítulo 4 apresenta uma breve revisão bibliográfica sobre sistemas de rastreamento que utilizam câmera fixa e modelos de subtração de fundo para identificar objetos em movimento. Neste capítulo também são descritos o filtro de Kalman e as métricas de associação de objetos através de cor, área e posição do centróide. O Capítulo 5 apresenta uma descrição do sistema de gerenciamento de objetos utilizado e o Capítulo 6 descreve a implementação do sistema de rastreamento proposto. Os experimentos e resultados utilizando o sistema de rastreamento proposto são apresentados no Capítulo 7. O Capítulo 8 apresenta conclusões e sugestões sobre futuras extensões do presente trabalho.

Capítulo 2

Subtração de Fundo

A subtração de fundo é uma etapa fundamental num sistema de rastreamento, definindo não apenas os alvos a serem rastreados como também sua forma e posição, assim, o modelo de subtração de fundo utilizado pode ser um fator determinante para o sucesso ou fracasso do sistema de rastreamento.

Pode-se definir a subtração de fundo como um conjunto de técnicas utilizadas para detecção de objetos num vídeo. Tais técnicas são baseadas na construção de um modelo que representa o fundo da cena, o chamado *background*. Este modelo é "subtraído" dos novos quadros, resultando nos objetos que diferem consideravelmente do modelo construído, o chamado *foreground*, o qual pode ser tanto um objeto em movimento quanto uma alteração na cena ou ainda um ruído resultante do próprio processo de aquisição da imagem. Neste contexto, o termo subtração refere-se tanto a operações matemáticas elementares quanto a abordagens probabilísticas mais complexas.

Construir um modelo e realizar a subtração do fundo não é uma tarefa trivial uma vez que, além dos ruídos, sombras podem ser detectadas como *foreground* e o ambiente, principalmente quando externo, está sujeito a variações na iluminação e modificações estruturais como incorporação ou saída de objetos, dificultando a separação de *foreground* e *background*, exigindo que o modelo se adapte a estes novos estados.

Métodos de subtração de fundo vêm sendo estudados desde a década de 70 com o trabalho de Jain e Nagel[18], onde cada novo quadro era subtraído do quadro anterior e, a esta subtração aplica-se um *threshold*, resultando no *foreground*. Tal abordagem é sensível ao valor do *threshold*, além de funcionar apenas em condições particulares de velocidade do objeto.

A popularização dos métodos de subtração de fundo ocorreu com a introdução das distribuições Gaussianas no modelamento do *background*, método proposto por Wren *et al* [19]. Os valores de cor assumidos por um *pixel* ao logo do tempo são modelados por uma única distribuição Gaussiana. A cada novo quadro, o novo valor deste *pixel* é avaliado nesta Gaussiana, se o valor pertence à distribuição, então o *pixel* é classificado como *background* e os parâmetros da distribuição são atualiza-

dos; caso contrário, o *pixel* é classificado como *foreground*. Apesar do sucesso, o modelo apresenta uma grande limitação na modelagem de *background* em ambientes externos em decorrência das variações de iluminação, além de não modelar *background* que apresenta movimento repetitivo, como ondas no mar, ou galhos de árvores balançando.

Com o objetivo de suprir as limitações do método de Wren *et al*, Stauffer e Grimson [14] propuseram o método conhecido como Mistura de Gaussianas (MoG - *Mixture of Gaussians*), sendo atualmente a abordagem mais utilizada em subtração de fundo. Os valores de cor assumidos por cada *pixel* da imagem são modelados por um número pré-definido de distribuições Gaussianas (normalmente de 3 a 5). Cada novo valor assumido pelo *pixel* é testado em cada distribuição e, caso pertença a alguma, é classificado como *background* e a distribuição é atualizada. A vantagem deste tipo de abordagem é permitir mais de uma camada de *background*, permitindo modelar variações de iluminação, movimentações repetitivas, além de incorporar novos elementos ao modelo de *background*. Entretanto é preciso definir o número de Gaussianas a serem utilizadas e a inicialização de seus parâmetros.

Elgammal *et al* [20] propuseram uma generalização da mistura de Gaussianas, chamado *Kernel Density Estimation* (KDE) com o objetivo de modelar o fundo de forma não-paramétrica, adaptando-se rapidamente a mudanças no *background* e identificando alvos com alta sensibilidade. Os valores assumidos por um *pixel* são modelados por uma função de probabilidade. A cada novo instante, o novo valor do *pixel* é avaliado se pertence à distribuição de densidade através de um estimador de *kernel*, no caso, uma distribuição Normal. Assim, o *pixel* é considerado pertencente ao *foreground* se a probabilidade do *pixel* pertencer à distribuição é menor que um *threshold*. Um segundo estágio de detecção de *background* é aplicado para remover falsos positivos, verificando se os *pixels* classificados como *foreground* pertencem a alguma distribuição de probabilidade numa vizinhança. Para eliminar efeitos de sombra, são utilizadas as normalizações dos canais R, G e B para o modelo. Este modelo tem como principal limitação o alto custo computacional, além de necessitar de espaço para armazenamento de imagens.

Oliver *et al* [15] propuseram uma modelagem de *background* baseada na Análise de Componentes Principais (PCA). A técnica chamada de *Eigenbackground* tem o objetivo reduzir a dimensionalidade do espaço da imagem, para tal, um autoespaço é constituído a partir de n quadros e cada novo quadro é levado a este autoespaço onde é subtraído da média das imagens. O resultado desta subtração é chamado de resíduo que, levado ao espaço da imagem, resulta nos objetos de *foreground*. Esta abordagem destaca-se em relação às abordagens clássicas por sua rapidez e diminuição do número de falsos positivos sem perder a qualidade do resultado.

Haritaoglu *et al* [21] utilizaram em seu sistema de rastreamento W^4 uma subtração de fundo baseada nos valores máximos e mínimos de uma sequência de vídeo em tons de cinza. É utilizado um

período de aproximadamente 30 segundos para o aprendizado do modelo e, deste período extrai-se o valor mínimo, máximo e a máxima diferença entre dois quadros que cada *pixel* assumiu. De acordo com inequações simples envolvendo os parâmetros adquiridos na fase de treinamento, o valor atual de cada *pixel* e a mediana da máxima diferença entre dois quadros, é estipulado se cada *pixel* pertence ao *foreground* ou *background*. Com ajuda de "mapas de suporte", o fundo é atualizado a cada n quadros, sendo este número estipulado pela taxa de atualização.

Jaraba *et al* [22] propôs um modelo de fundo chamado *Double-Background*. O propósito do modelo é identificar e distinguir objetos em movimento e estáticos (por exemplo uma pessoa parada, ou um carro estacionado). Para tal, computa-se a diferença entre o quadro atual e o modelo de fundo e a diferença entre o quadro atual e o anterior. O *threshold* aplicado a estas diferenças é automaticamente calculado como proposto no trabalho de Kim *et al* [23], que consideram variações na iluminação. Um objeto é considerado estático se for identificado nos dois tipos de subtração. Objetos que permanecem estáticos por um período de tempo são incorporados ao modelo de fundo.

O *Background-Weighted Histograms* é um modelo de subtração de fundo desenvolvido por Comaniciu *et al* [24]. Cada objeto seguido é representado por um histogramas de retropropagação desenvolvido por Swain e Ballard [25] de uma variável de interesse fotométrica, pode ser escala de cinza, cores ou textura. O objetivo é encontrar uma região na imagem com uma densidade que mais se aproxima à densidade da amostra do objeto seguido, para tal utiliza-se a distância de Bhattacharyya [26]. Quando esta distância se aproxima de 0 indica que a região se diferencia da amostra, indicando uma região de *background*, caso contrário, se a distância se aproxima de 1, indica combinação com a amostra, no caso o objeto seguido. Este método funciona apenas para casos em que objetos podem ser aproximados para uma elipse.

Han *et al* [27] utilizam um modelo de densidade não linear gerado pela soma ponderada de Gaussianas. O método chamado *Mean-Shift* detecta, através do deslocamento médio da largura das bandas, novos modelos, assim gerando novas Gaussianas e excluindo modelos antigos. Este modelo, assim como o KDE [20], também é limitado pelo uso da memória (armazenamento de n quadros) e também pelo alto custo computacional.

Lu *et al* [28] utilizam Transformada Wavelet (DWT - *Discrete Wavelet Transform*) para identificar objetos de *foreground*. O modelo de fundo é gerado a partir da decomposição em 3 camadas de uma imagem de fundo. Cada novo quadro é decomposto pela DWT e seu componente de alta freqüência é subtraídos do componente de alta freqüência do modelo de fundo. O resultado desta subtração é reconstruído formando assim uma imagem com elementos de *foreground*. Este tipo de subtração resulta na perda de informações da imagem, desta forma, há uma limitação referente ao tamanho do objeto que pretende-se identificar.

Técnica	**Autor(es)**	**Vantagens**	**Limitações**
Accumulative Difference Pictures	Jain e Nagel[18]	Rapidez de processamento	Sensível ao *threshold* e às variações de iluminação; depende da velocidade dos objetos no *foreground*
Modelo Gaussiano	Wren *et al* [19]	Compensa lentas variações de iluminação	*Pixel* pode ser modelado em mais de uma distribuição gaussiana; limitado a ambientes internos
Mistura de Gaussianas (MoG)	Stauffer e Grimson [14]	Modela sombras, movimentos repetitivos e refletância	Necessita de parâmetros iniciais para as gaussianas e a quantidade de gaussianas a serem utilizadas
Kernel Density Estimation (KDE)	Elgammal *et al* [20]	Modela fundos não estáticos; pouco sensível a sombras	Necessita armazenar um grande número de quadros; alto custo computacional
Eigenbackground	Oliver *et al* [15]	Baixo custo computacional;pouco sensível a variações na iluminação	Necessita de treinamento; é preciso definir valor de *threshold* e outros parâmetros
W^4	Haritaoglu *et al* [21]	Simples implementação	Aplicado apenas em imagens em tons de cinza; é sensível a mudanças de parâmetros
Double-Background	Jaraba *et al* [22]	Distingue objetos em movimento de objetos estáticos	Sensível a variações na iluminação
Background-Weighted Histograms	Comaniciu *et al* [24]	Método Simples	Alto custo computacional; objetos precisam ser aproximados para uma elipse
Mean-Shift	Han *et al* [27]	Incorpora novas distribuições	Intenso uso da memória para armazenamento de quadros ; alto custo computacional

Tab. 2.1: Técnicas de Subtração de Fundo: suas vantagens e limitações.

2.1 Mistura de Gaussianas

O modelo de subtração de fundo Mistura de Gaussianas (MoG - *Mixture of Gaussians*), desenvolvido por Stauffer e Grimson [29], é um dos métodos mais utilizados para identificação de *foreground* em vídeos. Este modelo é bastante conhecido por tratar de situações onde o *background* é dinâmico, isto é, os *pixels* do *background* podem assumir valores bastante diferentes, como no caso de uma árvore que balança, ou ondas no mar, além de tratar da variação do *background* com o tempo, situações que ocorrem principalmente em ambiente externos, em decorrência da grande variação de iluminação durante o dia.

Neste modelo, os valores assumidos pelos *pixels* são modelados por distribuições gaussianas, normalmente de 3 a 5. Tais distribuições são ordenadas de acordo com um peso, sendo que as gaussianas com maior peso são associadas ao modelo de *background*. A cada novo quadro, o valor assumido pelo *pixel* é testado nas gaussianas do modelo e, dependendo da distribuição a qual é associado, este *pixel* é classificado como *background* ou *foreground*.

O método baseia-se na suposição de que uma única distribuição gaussiana é suficiente para modelar os valores de um *pixel* se estes variam uniformemente dentro de um único intervalo de valores. Entretanto, se estes valores concentram-se em mais de um intervalo, duas ou mais distribuições Gaussianas devem ser utilizadas uma vez que a utilização de uma única distribuição resultaria numa alta variância.

Esta suposição é ilustrada pela figura 2.1, onde o gráfico mostra um dos quadros de um vídeo de 500 quadros e plotagem dos valores em R,G e B assumidos por um *pixel* na posição $(97, 196)$, que pertence a uma região onde uma corda balança ao longo do tempo. Neste caso, a utilização de uma única Gaussiana para modelar os valores assumidos pelo *pixel* não é ideal, uma vez que os mesmos estão concentrados em diferentes intervalos de valores. Este problema pode ser contornado utilizando-se duas ou mais Gaussianas para representar este conjunto de valores.

A atualização do modelo MoG permite que o mesmo seja aplicado em vídeos onde há variações na iluminação e onde elementos do *foreground* são incorporados ao *background*, como por exemplo no caso de rastreamento de objetos num estacionamento: os carros entram na cena e são rastreados, porém a partir do momento que estão estacionados, é desejável que estes sejam classificados como *background*, permitindo o rastreamento de outros objetos, sem lidar com grande número de oclusões. Neste sentido, o modelo MoG supõe que se um *pixel* classificado como *foreground* assume valores próximos durante um certo período de tempo, este *pixel* deve pertencer ao *background*, então a distribuição Gaussiana que modela tais valores passa a compor o modelo de fundo.

2.1.1 Modelo Gaussiano

Considerando os valores de um pixel no ponto x_0, y_0 até o instante de tempo t, seu histórico é definido por:

$$X_1, ..., X_t = I(x_0, y_0, i) : 1 \leq i \leq t, \tag{2.1}$$

onde I é a sequência de imagens e X_k é um vetor n-dimensional que contém os valores do *pixel* nos n canais de cores no instante k. A história recente de cada *pixel* é modelada como uma mistura de K Gaussianas, assim a probabilidade da intensidade do *pixel* observado é

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} * \eta(X_t, \mu_{i,t}, \Sigma_{i,t}), \tag{2.2}$$

onde K é o numero de distribuições, normalmente variando de 3 a 5, $\omega_{i,t}$ é o peso da i-ésima Gaussiana da mistura no instante t. $\mu_{i,t}$ e $\Sigma_{i,t}$ são a média e a matriz de covariância da i-ésima Gaussiana no instante t, e η é uma função de densidade Gaussiana definida por

$$\eta(X, \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{1}{n}} |\Sigma|^{\frac{1}{2}}} \exp^{-\frac{1}{2}(X-\mu)^T \Sigma^{-1}(X-\mu)}, \tag{2.3}$$

onde $|\Sigma|$ denota a determinante da matriz de covariância de dimensão $n \times n$. Para diminuir o custo computacional, assume-se que os canais de cores são independentes e possuem mesma variância, simplificando a inversão da matriz Σ na equação 2.3. Desta forma, a matriz de covariância Σ é definida por:

$$\Sigma_{k,t} = \sigma_k^2 \mathbf{I}, \tag{2.4}$$

onde σ_k denota a variância da k-ésima Gaussiana e $\mathbf{I}$ a matriz identidade de dimensão $n \times n$.

A cada novo instante de tempo t_0, cada novo valor de *pixel* X_{t_0} é testado nas K distribuições Gaussianas existentes com o objetivo de encontrar a distribuição a qual pertence. O valor de um *pixel* é dito pertencente à uma determinada distribuição Gaussiana, ou associado à distribuição, se seu valor distar da média até 2.5 vezes o desvio padrão, intervalo que abrange 95% dos dados da distribuição. Caso o *pixel* não seja associado às distribuições existentes, a distribuição com menor relação $\frac{\omega}{\sigma}$ é substituída por uma nova que representa a observação atual, sendo os seus parâmetros inicialmente ajustados para uma alta variância, um baixo peso e a média com o valor do pixel corrente.

2.1.2 Atualização dos Modelos e Estimação do Modelo de *Background*

A cada novo quadro, o peso de cada distribuição Gaussiana é atualizado por:

$$\omega_{k,t} = (1 - \alpha)\omega_{k,t-1} + \alpha(M_{k,t}), \tag{2.5}$$

onde α é a taxa de aprendizado, assim quanto mais próximo de 1, mais rápida é a incorporação de novos dados. $M_{k,t}$ assume valor igual a 1 se o valor do presente *pixel* pertence à k-ésima distribuição Gaussiana ou 0, caso contrário. O peso $\omega_{k,t}$ refere-se à esperança do *pixel* baseado em seus valores passados.

Os valores de μ e σ da k-ésima distribuição Gaussiana são atualizados quando o *pixel* pertence à sua distribuição. A atualização é realizada por:

$$\mu_t = (1 - \rho)\mu_{t-1} + \rho X_t \tag{2.6}$$

$$\sigma_t^2 = (1 - \rho)\sigma_{t-1}^2 + \rho(X_t - \mu_t)^T(X_t - \mu_t), \tag{2.7}$$

onde

$$\rho = \alpha\eta(X_t|\mu_k, \sigma_k), \tag{2.8}$$

onde $\eta(X_t|\mu_k, \sigma_k)$ é o valor da função de densidade Gaussiana, dado $\mu = \mu_k$ e $\Sigma = \Sigma_{k,t}$.

Os parâmetros de cada Gaussiana muda a cada quadro, assim deve-se determinar quais distribuições têm maior probabilidade de pertencer ao modelo de *background*. O modelo MoG assume que os valores assumidos pelos *pixels* pertencentes ao *background* variam de forma mais uniforme e em intervalos bem definidos de valores, já os valores assumidos pelos *pixels* pertencentes ao *foreground* tendem a variar mais e de forma indefinida, assim estes valores costumam aumentar a variância das distribuições às quais são associados ou não são associados às distribuições existentes, resultando na criação de novos modelos gaussianos. Desta forma, as distribuições com menor variância e maior peso são selecionadas para compor o modelo de *background* do *pixel* em questão.

A cada quadro, as distribuições gaussianas de um determinado *pixel* são ordenadas de forma decrescente de acordo com o valor de $\frac{\omega}{\sigma}$. Este valor mantém as gaussianas com maior peso e menor variância nas primeiras colocações. As primeiras B distribuições são escolhidas para compor o modelo de fundo, de acordo com a relação:

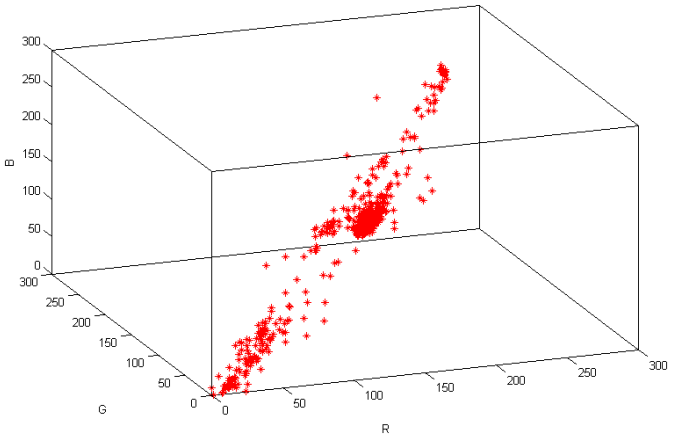
$$B = argmin_b(\sum_{k=1}^{b} \omega_k > T), \tag{2.9}$$

onde T é a fração do peso total dado ao modelos de *background*. Se o valor escolhido para T é pequeno, o *background* é geralmente unimodal, utilizando apenas a distribuição mais provável. Por outro lado, se T é grande, assume-se distribuição multimodal, causada por variações do *background*,

assim, duas ou mais cores separadas assumidas pelos mesmo *pixel* podem ser incorporadas ao modelo de *background*. Este resultado permite ao *background* aceitar duas ou mais cores separadas. A figura 2.1 mostra os valores (R, G, B) assumidos por um pixel localizado na região onde há uma corda balançando durante uma sequência de vídeo de 30 segundos. Pode-se notar que este pixel assume diferentes valores com o passar do tempo, assim pode ser modelado por mais de uma distribuição.

Se o valor do *pixel* analisado pertencer a alguma destas B distribuições gaussianas, então é classificado como *background*, caso contrário, será classificado como *foreground*.

(a) Imagem Original



(b) Distribuição RGB do *pixel* $(97, 196)$

Fig. 2.1: Distribuição RGB de um *pixel* na posição $(97, 196)$ ao longo do tempo. O círculo indica a região da posição do *pixel*.

2.2 *Eigenbackground*

A Análise de Componentes Principais (PCA - *Principal Component Analysis*) é uma técnica popular de parametrização da forma, aparência e movimento, com diversas aplicações em reconhecimento de padrões [30, 31, 32], funcionando como um método de extração de características não supervisionado, propício para dados com distribuição Gaussiana [33]. Oliver *et al.* [15] introduziram o uso do PCA numa técnica de subtração de fundo chamada *Eigenbackground*.

O modelo *Eigenbackground* é uma alternativa aos métodos tradicionais de subtração de fundo, destacando-se por considerar informações espaciais da imagem, sem necessidade de estimar funções de densidade a cada *pixel* da imagem, além de ser pouco sensível às variações de iluminação do ambiente.

A técnica é composta por duas etapas: treinamento e teste. A inicialização do modelo *Eigenbackground* ocorre na etapa de treinamento e consiste na formação de um autoespaço com o objetivo de se obter o menor número de características que represente o fundo com precisão. Para isso, um bloco de imagens é armazenado e dele computado a matriz de covariância. Sobre esta matriz é calculado o PCA a fim de eliminar ou reduzir redundâncias entre as características, e o conjunto de autovetores resultantes formam o autoespaço.

A fase de teste refere-se à identificação do *foreground*. As cenas da sequência de vídeo são projetadas no autoespaço formado na fase de treinamento, subtraídas da média e re-projetadas para o espaço da imagem, resultando numa imagem de resíduos, que após a aplicação de um *threshold* resulta no *foreground*.

Embora os bons resultados apresentado por Oliver *et al.* e por Xu *et al.* [34], o modelo *Eigenbackground* falha quando há variações do *background*, uma vez que a proposta original (Oliver *et al.* [15]) não inclui a atualização do modelo. Para suprir esta deficiência, um modelo de fusão de autoespaços é proposto por Han e Jain [27].

2.2.1 Formação do Autoespaço

Um autoespaço é formado por um conjunto de N imagens de dimensão $n \times m$ do *background*. Cada imagem é vetorizada e agrupada formando uma matriz X de tamanho $mn \times N$. Desta forma, dada uma imagem I_k, sua vetorização é feita através da leitura coluna a coluna da imagem, colocando o valor de cada pixel da imagem em um vetor coluna x de tamanho $nm \times 1$, desta forma:

$$X = [x_1 x_2 ... x_N]. \tag{2.10}$$

O vetor médio do *pixels* das imagens $\bar{x}$ é calculado por

$$\bar{x}(k) = \frac{1}{N}\sum_{i=1}^{N} X_{k,i}, \tag{2.11}$$

onde $1 \leq k \leq nm$.

A matriz de covariância, ou matriz de dispersão, é a matriz das covariâncias entre elementos de um vetor randômico, medindo o grau de linearidade entre dois elementos quaisquer e pode ser obtida por:

$$Cov(X) = E[(X - E[X])(X - E[X])^T], \tag{2.12}$$

onde $E[\cdot]$ é o valor da esperança. No caso de $E[X]$, o valor da esperança é equivalente à média $\bar{x}$. Assim pode-se aproximar o valor da matriz de covariância de X para

$$C = (X - \bar{x})(X - \bar{x})^T. \tag{2.13}$$

Nota-se que $C_{k,k}$, os elementos da diagonal da matriz C, denotam a variância da posição k das imagens vetorizadas, enquanto elementos fora da diagonal representam a covariância entre duas características quaisquer. Estatisticamente, quando a covariância é nula e as características têm distribuição Normal, as variáveis são independentes, eliminando redundâncias entre as mesmas.

Para que não exista covariância, e portanto redundâncias entre *pixels* diferentes, é necessário diagonalizar a matriz C. Duda e Hart [33] mostraram que devido ao processo de criação da matriz de covariância, esta pode ser diagonalizada através de uma mudança de base definida por autovetores (apêndice A). Desta forma deseja-se calcular

$$\Lambda U^T = U^T C, \tag{2.14}$$

onde U^T é a matriz de autovetores da matriz de covariância e Λ é a matriz diagonal de seus autovalores.

Cada autovalor da matriz de covariância representa a variância de uma das características transformada. Portanto, quanto maior o autovalor, maior a variância da característica na direção de seu autovetor correspondente, desta forma, pode-se reduzir a dimensionalidade do problema armazenando apenas os l autovetores correspondentes aos l maiores autovalores, desprezando características de menor relevância. Os l autovetores formam a matriz U de mudança de espaço (espaço da imagem $\rightarrow$ autoespaço). Assim, definimos o modelo de autoespaço Ω, como a média dos valores dos *pixels*, o conjunto (reduzido) de autovetores, seus autovalores e o número de observações:

$$\Omega = (\bar{x}, U, \Lambda, N). \tag{2.15}$$

2.2.2 Identificação do *Foreground*

Com o autoespaço formado, a identificação do *foreground* numa imagem I_i, já vetorizada, é realizada através da projeção da imagem subtraída da média $\bar{x}$ no autoespaço Ω, formado na etapa de treinamento, e re-projeção para o espaço da imagem.

Para facilitar os cálculos, utiliza-se autovetores unitários (de tamanho 1) na formação da matriz U, assim pode-se utilizar a relação $U^T = U^{-1}$, uma vez que os mesmos formam uma base ortonormal. A projeção de $I_i - \bar{x}$ em Ω é calculada por

$$I'_i = U^T(I_i - \bar{x}), \tag{2.16}$$

obtendo-se o vetor M-dimensional I'_i. Esta transformação chama-se transformada de Karhunen-Loève [35]. I'_i é projetado no espaço da imagem por

$$UI'_i. \tag{2.17}$$

A imagem reconstruída é dada por:

$$I_{rec} = UI'_i + \bar{x}. \tag{2.18}$$

O resíduo da imagem de entrada pelo modelo de fundo é calculado através da subtração da imagem de entrada pela imagem reconstruída, desta forma temos:

$$D_i = I_i - I_{rec}. \tag{2.19}$$

A figura 2.2 mostra um exemplo da aplicação do *Eigenbackground* para subtração de fundo: a imagem 2.2(a) é a imagem de entrada, 2.2(b) mostra a reconstrução da imagem e o resíduo é apresentado na imagem 2.2(c). Neste exemplo, o autoespaço foi formado através de 100 imagens do fundo e 10 autovetores.

Os pixels que compõe o foreground são identificados aplicando-se um threshold th ao módulo do vetor D_i, resultando numa imagem binária Res, onde 1 representa objetos de foreground e 0 o fundo, ou background:

$$\begin{cases} |D_i(k)| \geq th \Rightarrow Res(k) = 1 \\ |D_i(k)| < th \Rightarrow Res(k) = 0 \end{cases} \tag{2.20}$$

2.2.3 Atualização do modelo *Eigenbackground*

O modelo *Eigenbackground* apresentado por Oliver *et al* não prevê a atualização do autoespaço, o que limita o modelo a aplicações em cenas de pouca variação de iluminação, além de não tratar da incorporação ou saída de elementos do *background*. Para lidar com este tipo de problema, Han e Jain [27] propuseram a aplicação de uma fusão de autoespaço, chamada de PCA incremental (IPCA), desenvolvida por Hall *et al* [36]. A cada conjunto de M novas imagens, calcula-se seu autoespaço que será fundido com o autoespaço já existente, gerando um terceiro autoespaço que será utilizado como modelo de *background*.

Uma outra opção para a atualização do *Eigenbackground* é recalcular a média e autovetores a cada conjunto de M novas imagens, formando um novo autoespaço Ψ que substituirá o autoespaço calculado na fase de treinamento Ω. Este tipo de solução apresenta como principal dificuldade a rápida incorporação do *foreground* ao modelo, uma vez que as imagens utilizadas neste re-treinamento não são exclusivamente do *background*. Desta forma, a solução proposta é formar um conjunto de M novas imagens extraindo, com o auxílio do sistema de rastreamento, os alvos seguidos, evitando que estes contribuam para a formação do novo autoespaço.

Fusão de Autoespaços

A idéia principal do IPCA proposto por Hall *et al* é armazenar um bloco de imagens e atualizar o modelo de *background* a cada conjunto de M novas imagens, formando um novo autoespaço Ψ que, fundido com o autoespaço atual Ω, resulta num terceiro autoespaço Φ que substituirá Ω. A vantagem do IPCA é reduzir as operações necessárias para o calculo dos autovalores e autovetores do novo espaço formado, calculando-os através da rotação do espaço Ψ, uma vez que o cálculo direto da fusão é computacionalmente caro, inviabilizando sua aplicação num sistema de rastreamento.

Considerando uma coleção de M novos quadros, pela equação 2.10 obtemos $Y = [y_1, y_2, ..., y_M]$ e, assim como mostrado nas seções 2.2.2 e 2.2.1, pode-se obter o modelo de autoespaço do conjunto Y, definido por $\Psi = (\bar{y}, V, \Delta, M)$.

Ψ representa o autoespaço das novas observações, entretanto deseja-se calcular o autoespaço resultante da fusão entre o autoespaço antigo Ω e Ψ, a fim de formar um novo autoespaço $\Phi = (\bar{z}, W, \Pi, P)$.

Claramente, o número de novas observações é obtido por $P = N + M$ e a média da combinação por:

$$\bar{z} = \frac{1}{P}(N\bar{x} + M\bar{y}). \tag{2.21}$$

O cálculo direto da matriz de covariância por:

$$E = \frac{N}{P}(C) - \frac{M}{P}YY^T + \frac{NM}{P^2}(\bar{x} - \bar{y})(\bar{x} - \bar{y})^T. \tag{2.22}$$

Pode-se notar que o cálculo direto de E é computacionalmente custoso devido às dimensões dos vetores e matrizes envolvidos na operação, inviabilizando seu uso para aplicações em tempo real. Contudo, deseja-se encontrar autovetores e autovalores que satisfaçam

$$E = W\Pi W^T. \tag{2.23}$$

Desta maneira, Hall *et al* propõem derivar tais autovalores e autovetores através de três etapas:

1. construção de uma base ortonormal Υ que abrange os dois modelos de autoespaço Ω e Ψ e $\bar{x} - \bar{y}$;
2. através de Υ, encontrar os autovalores Π;
3. computar autovetores W.

Construção da Base Ortonormal

Para construir uma base ortonormal para a combinação dos autoespaços Ω e Ψ, deve-se escolher um conjunto de vetores ortonormais que abranjam os referidos autoespaços além de $(\bar{x} - \bar{y})$. Este conjunto difere dos autovetores W por uma rotação R, assim:

$$W = \Upsilon R, \tag{2.24}$$

onde Υ é expresso por:

$$\Upsilon = [Uv]. \tag{2.25}$$

v é um conjunto de bases ortonormais de Ψ que é ortogonal a Ω e mantém $(\bar{x} - \bar{y})$ ortogonal aos dois autoespaços. Definindo H por:

$$G = U^T V \tag{2.26}$$

$$H = V - UG, \tag{2.27}$$

e calculando-se o resíduo h de $(\bar{x} - \bar{y})$ com respeito ao autoespaço Ω, utilizando a equação 2.19, pode-se encontrar então o valor de v através da ortonormalização de Gramm-Schmidt [37]:

$$v = \text{Ortonormalização}(H, h). \tag{2.28}$$

Desta forma, voltando à equação B.4, encontramos o valor da base ortonormal Υ, concluindo o primeiro passo.

Formando um Novo Autoespaço

Um novo autoespaço pode ser obtido através da manipulação das equações já obtidas. O desenvolvimento pode ser encontrado no apêndice B, e resulta em:

$$R\Pi R^T = \frac{N}{P}\begin{bmatrix} \Lambda & 0 \\ 0 & 0 \end{bmatrix} + \frac{M}{P}\begin{bmatrix} G\Delta G^T & G\Delta\Gamma^T \\ \Gamma\Delta G^T & \Gamma\Delta\Gamma^T \end{bmatrix} + \frac{NM}{P^2}\begin{bmatrix} gg^T & g\gamma^T \\ \gamma g^T & \gamma\gamma^T \end{bmatrix}, \tag{2.29}$$

onde $g = U(\bar{x} - \bar{y})$, $\gamma = v^T(\bar{x} - \bar{y})$ e $\Gamma = v^T V$. Nota-se que os autovalores Π são os autovalores desejados, entretanto resta encontrar a matriz de autovetores W.

Calculando Autovetores

Com R e Υ calculados nas seções anteriores (seções 2.2.3 e 2.2.3), o calculo de autovetor Π é mostrado no Apêndice B , entretanto, como feito anteriormente, nem todos autovetores são utilizados para formar o autoespaço, desta forma alguns autovetores são descartados de acordo com o critério discutido na Seção 2.2.1.

O autoespaço formado trata de uma aproximação da fusão de dois autoespaços, uma vez que os cálculos estão sujeitos a arredondamentos e sofrem com a perda de informações pois os autovetores que compõe autoespaço anterior Ω estão truncados com o descarte de autovalores e autovetores.

Com o novo autoespaço Ψ calculado, este será utilizado para o reconhecimento do *foreground* nas próximas cenas, substituindo o autoespaço anterior Ω que pode ser descartado.

(a) Entrada

(b) Imagem reconstruída

(c) Diferença entre a entrada e a imagem reconstruída

Fig. 2.2: Processo de identificação do *foreground*.

Capítulo 3

Avaliação dos Modelos de Subtração de Fundo MoG e *Eigenbackground*

O objetivo deste capítulo é apresentar uma comparação entre os dois modelos de subtração de fundo estudados: Mistura de Gaussianas - MoG (Seção 2.1) e *Eigenbackground* (Seção 2.2). Esta comparação é realizada através de uma análise de sensibilidade de parâmetros, verificando a influência dos mesmos na identificação do *foreground* e definindo quais conjuntos de parâmetros produzem o resultado mais próximo do esperado.

Os testes para a análise de sensibilidade de parâmetros foram realizados com os vídeos das bases de dados PETS [38] e CAVIAR [39]. Os resultados obtidos foram avaliados de forma qualitativa e quantitativa, onde a primeira refere-se à análise visual dos resultados, enquanto a segunda visa analisar o desempenho de cada modelo através de taxas de precisão e revocação.

Para atualizar o modelo *Eigenbackground* foi empregada a técnica de fusão de autoespaços, descrita na Seção 2.2.3. Os parâmetros analisados para o referido modelo foram: numero de autovetores compõe o autoespaço, *threshold* e quantidade de novas imagens para atualização. Para o modelo MoG, foram analisados a variação da taxa de aprendizagem e do *threshold* do peso das gaussianas.

Existem outros parâmetros que também podem ser modificados em cada modelo, como mostra a Tabela 3.1, entretanto foram escolhidos os parâmetros que produziram diferenças mais significativas nos resultados através de uma análise visual prévia.

Para realizar a análise quantitativa, foram selecionadas aleatoriamente e segmentadas manualmente 10 imagens de cada vídeo como mostra a Figura 3.1. Para cada conjunto de parâmetros utilizados, são calculadas taxas de precisão ($Prec$) e revocação (Rev) baseadas no resultado obtido e no resultado desejado. As taxas são definidas por:

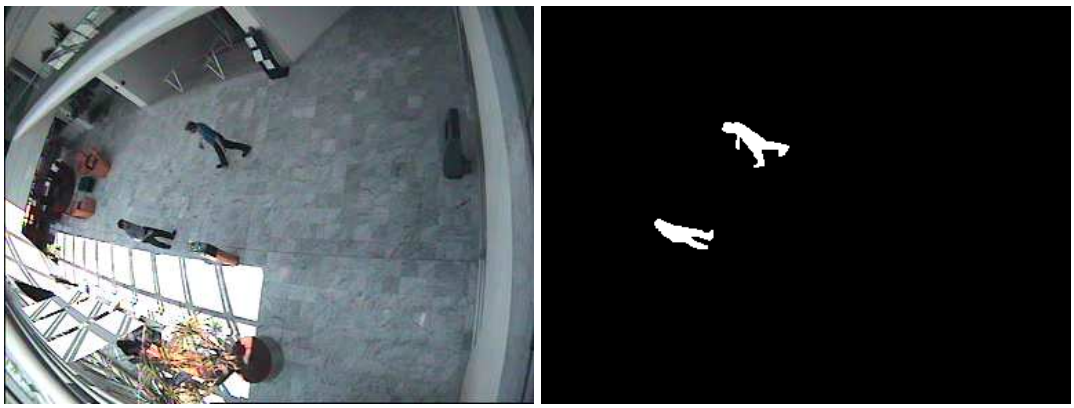
$$Prec = \frac{\text{número de } \textit{pixels} \text{ corretamente detectados como } \textit{foreground}}{\text{número total de } \textit{pixels} \text{ detectados como } \textit{foreground}} \tag{3.1}$$

Modelo	**Parâmetros Fixados**	**Parâmetros Testados**
Eigenbackground por fusão de autoespaços	Quantidade de imagens de treinamento $N = 100$	*Threshold* th Bloco de novas imagens M Quantidade de autovetores
Mistura de Gaussianas (MoG)	Número de componentes $K = 3$ Variância inicial $\sigma_0^2 = 9$ Peso inicial $\omega_0 = 0.1$	Taxa de aprendizagem α *Threshold* do peso T

Tab. 3.1: Parâmetros dos modelos de subtração de fundo.

$$Rev = \frac{\text{número de } \textit{pixels} \text{ corretamente detectados como } \textit{foreground}}{\text{número total de } \textit{pixels} \text{ de } \textit{foreground} \text{ na segmentação manual}}. \tag{3.2}$$

Estas taxas consideram apenas os *pixels* classificados como *foreground*, permitindo um melhor entendimento da qualidade da identificação dos objetos. A taxa de precisão expressa, dentre os *pixels* classificados como *foreground*, a quantidade de *pixels* corretamente classificados, enquanto a taxa de revocação expressa quanto dos *pixels* classificados como *foreground* realmente pertencem ao *foreground*. Tanto os valores de $Prec$ quanto os de Rev variam entre 0 e 1, sendo que um modelo ideal de subtração de fundo deve resultar em $Prec = Rev = 1$.

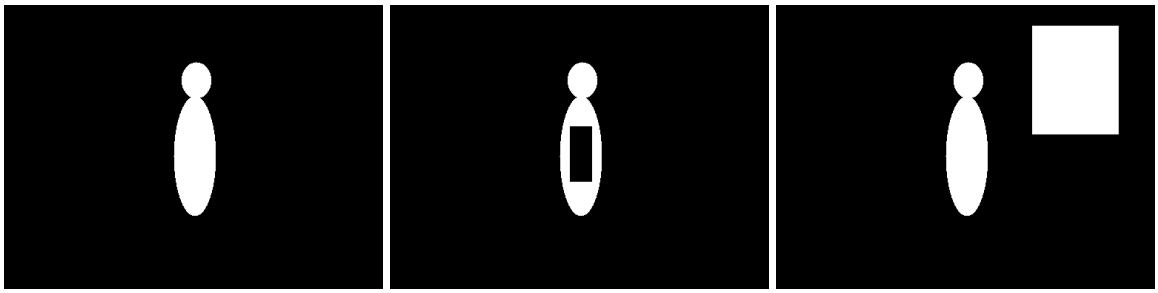


(a) Imagem original (b) Segmentação manual

Fig. 3.1: Imagem original e segmentação manual.

A Figura 3.2 mostra um exemplo de aplicação destas taxas, sendo que a Figura 3.2(a) mostra o resultado esperado e as Figuras 3.2(b) e 3.2(c), os resultados obtidos. No primeiro caso (Figura 3.2(b)), todos os *pixels* classificados como *foreground* pertencem a ele, portanto a taxa de precisão é igual ao 1, porém nem todos os pixels foram corretamente classificados, fazendo com que a taxa de revocação seja menor que 1. Já no segundo caso (Figura 3.2(b)), todos os pixels que pertencem ao *foreground* foram corretamente identificados, fazendo com que a taxa de revocação seja igual a 1,

porém nem todos *pixels* classificados pertencem ao *foreground*, baixando a taxa de precisão.



(a) Resultado desejado (b) Taxa de precisão = 1, Taxa de revocação = 0,74 (c) Taxa de precisão = 0,34, Taxa de revocação = 1

Fig. 3.2: Exemplo de comportamento das taxas de revocação e precisão.

Análise qualitativa é apresentada através de imagens do resultado da subtração de fundo, as quais são comparadas às imagens dos resultados esperados. Os resultados da análise quantitativa são apresentados na forma de gráficos, que representam a sensibilidade, em termos de taxa de revocação e precisão, de cada parâmetro variado.

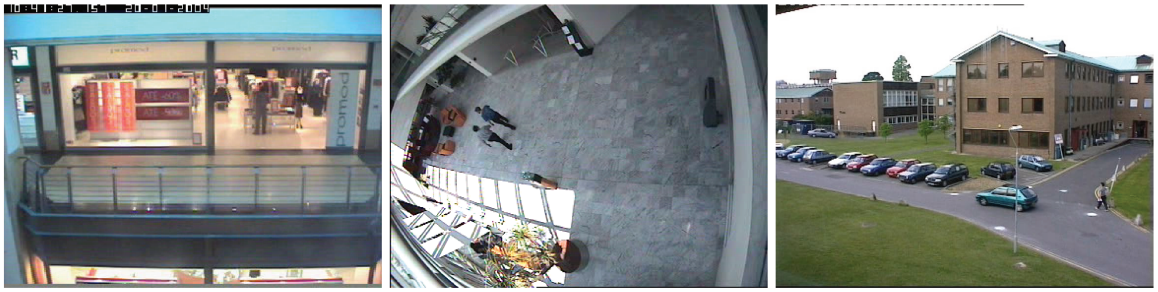
Ao final da análise dos dois modelos de subtração de fundo, resultados do modelo utilizado pelo sistema de rastreamento proposto são mostrados e comparados aos resultados obtidos pelo modelo *Eigenbackground* ressaltando as melhorias obtidas com o modelo proposto.

Outro aspecto a ser avaliado é o sistema de gerenciamento de objetos, o que definirá quais objetos são mais ou menos relevantes no sistema de rastreamento e os estados de cada objeto. Para testar o sistema de gerenciamento de objetos, foram avaliados quatro vídeos de segurança das bases de dados PETS e Caviar, escolhidos de acordo com seu grau de dificuldade:

Vídeo 1 Da base de dados CAVIAR, o vídeo mostrado na Figura 3.3(a) mostra o corredor de um shopping e tem como principal dificuldade ruídos nas imagens e elementos do *foreground* com cores similares às do *background*.

Vídeo 2 Da base de dados PETS, o vídeo mostrado na Figura 3.3(c) , além de apresentar variações de iluminação por apresentar cenas de um ambiente externo, ainda contém árvores que balançando com o vento na formação do *background*, e objetos de *foreground* que permanecem estáticos por um longo período de tempo.

Vídeo 3 Da base da dados CAVIAR, o vídeo mostrado na Figura 3.3(b) mostra o pátio interno de um shopping e apresenta como dificuldade a baixa qualidade das imagens e variações de iluminação uma vez que a iluminação externa influencia na iluminação de uma região através de uma janela encontrada na parte inferior da cena.



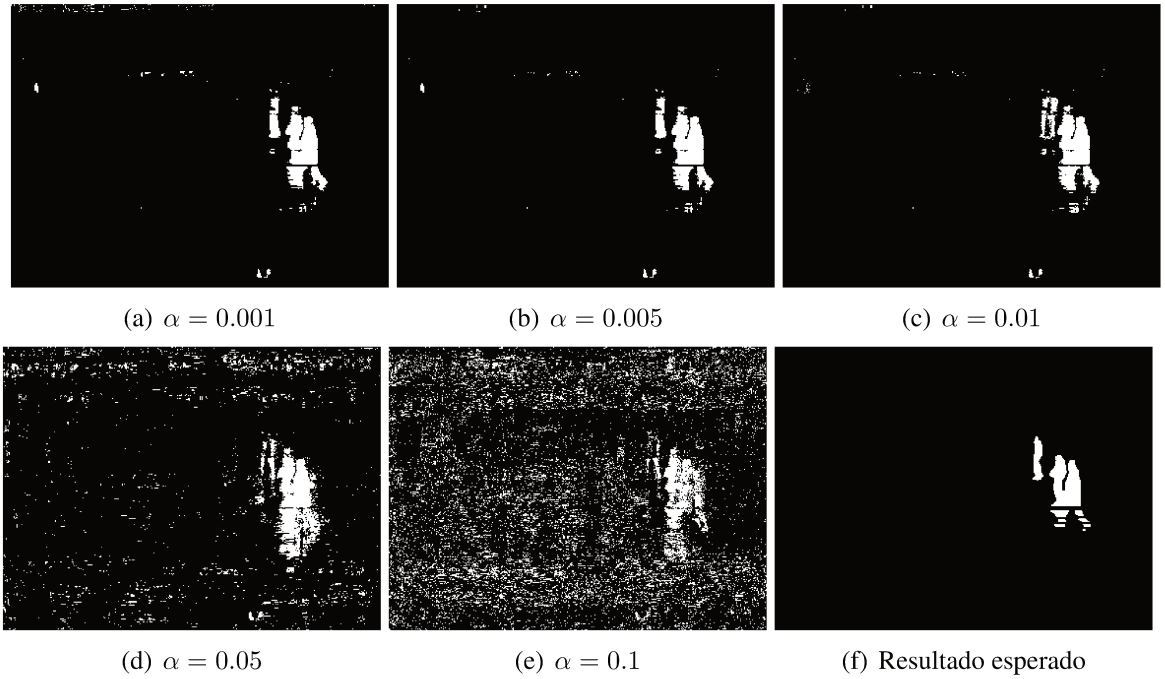
(a) Vídeo 1 (b) Vídeo 2 (c) Vídeo 3

Fig. 3.3: Videos utilizados para testes dos algoritmos de Subtração de Fundo.

3.1 Mistura de Gaussianas (MoG)

Como mostrado na Tabela 3.1, os testes realizados com o modelo de subtração de fundo MoG foram executados variando-se a taxa de aprendizagem α e o *threshold* T e fixando os demais parâmetros.

Os resultados obtidos mostraram que para um mesmo valor de T pode-se obter resultados bastante diferentes através da variação da taxa de atualização α, como mostra a Figura 3.5, referente ao vídeo 2 e na Figura 3.4, referente ao vídeo 1.



(a) $\alpha = 0.001$ (b) $\alpha = 0.005$ (c) $\alpha = 0.01$

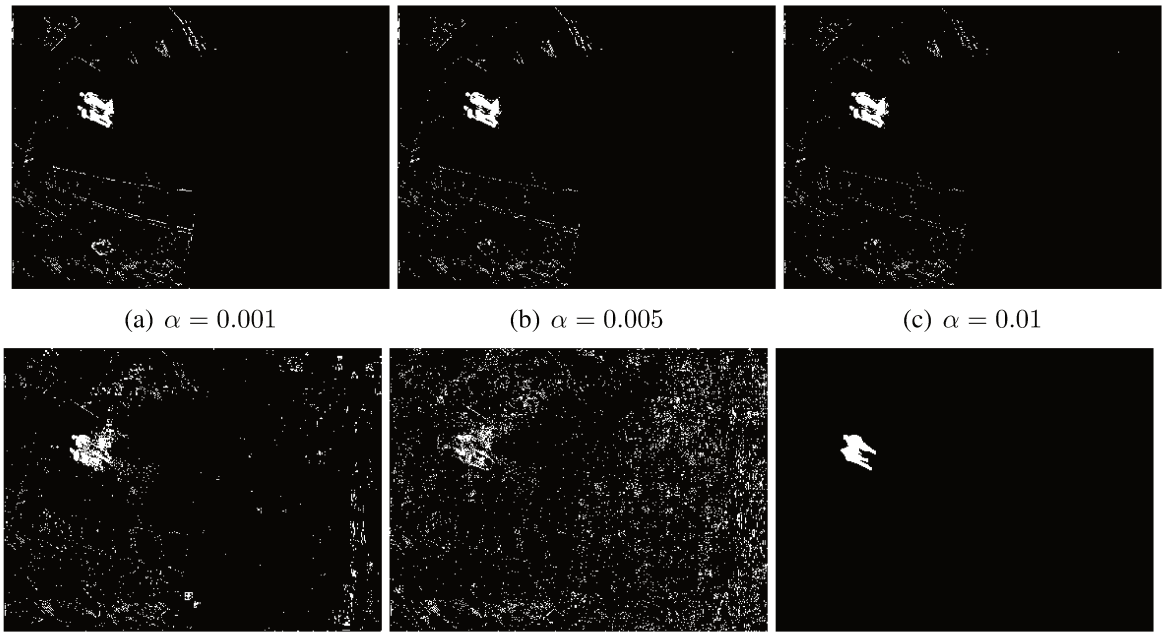
(d) $\alpha = 0.05$ (e) $\alpha = 0.1$ (f) Resultado esperado

Fig. 3.4: MoG - Imagens do vídeo 1 para $T = 0.2$.

Como já discutido na Seção 2.1, a variação do α resulta na variação de velocidade com que novos valores são incorporados ao modelo de fundo. Tal incorporação pode ser facilmente observada nos

resultados obtidos, uma vez que houve falhas na identificação do *foreground* nos casos em que a atualização é demasiadamente lenta e não incorpora mudanças rápidas na iluminação, o que pode ser observado nas Figuras 3.5(a) e 3.4(a).

Também houveram falhas quando a atualização do modelo é rápida sendo que neste caso, novas gaussianas são rapidamente incorporadas e descartadas, resultando numa identificação ruidosa e na rápida incorporação do *foreground* ao modelo de *background*, como pode-se observar nas Figuras 3.5(d), 3.5(e), 3.4(d) e 3.4(e), onde apesar de identificar *pixels* do *background* como *foreground*, pode-se notar que os pixels pertencentes ao *foreground* desejado (Figuras 3.5(f) e 3.4(f)) não são completamente identificados.



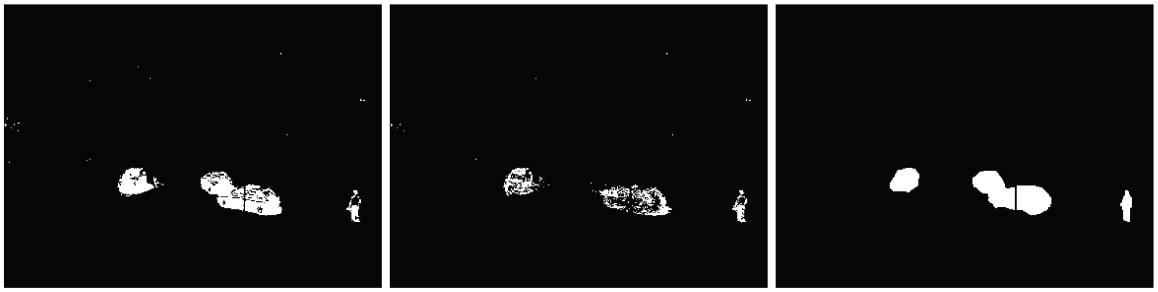
(a) $\alpha = 0.001$ (b) $\alpha = 0.005$ (c) $\alpha = 0.01$

(d) $\alpha = 0.05$ (e) $\alpha = 0.1$ (f) Resultado esperado

Fig. 3.5: MoG - Imagens do vídeo 2 para $T = 0.2$.

Nos resultados obtidos para o vídeo 3 pode-se notar de forma clara a incorporação de objetos do *foreground* ao modelo de *background*, como mostrado na seqüencia da Figura 3.6. O local deixado pelo carro que sai da vaga do estacionamento é incorporado ao modelo de *background* com mais ou menos velocidade, de acordo com o valor de α. Entretanto, observa-se também que partes do carro também são incorporadas, resultando numa imagem ruidosa.

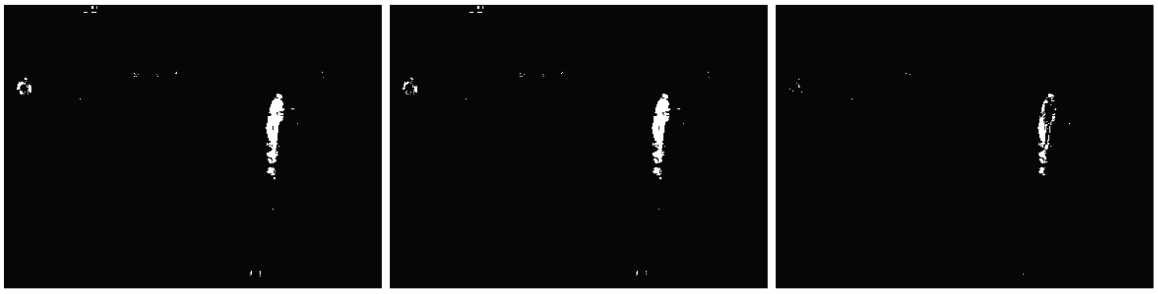
O efeito da variação do parâmetro T, como pode ser observado na Figura 3.7, é menos evidente que o efeito do parâmetro α, entretanto é notável notável, como mostra a figura. Como descrito na seção 2.1, para valores pequenos de T assume-se que o modelo do *background* pode ser aproximado para uma distribuição unimodal, enquanto que para T assume-se que mais de uma superfície pode surgir naquele *pixel*. A Figura 3.7, referente ao vídeo 1, mostra um momento em que o alvo está



(a) $\alpha = 0.001$ (b) $\alpha = 0.01$ (c) Resultado esperado

Fig. 3.6: MoG - Imagens do vídeo 3 para $T = 0.8$.

parado e começa a ser incorporado ao fundo, contudo, utilizando $T = 0.2$, Figura 3.7(a), o objeto não é incorporado uma vez que a nova superfície não é substituída pelo *background* já modelado. Entretanto, aumentando-se o valor de T, pode-se notar que o alvo passa a integrar o *background* como mostrado na Figura 3.7(c).

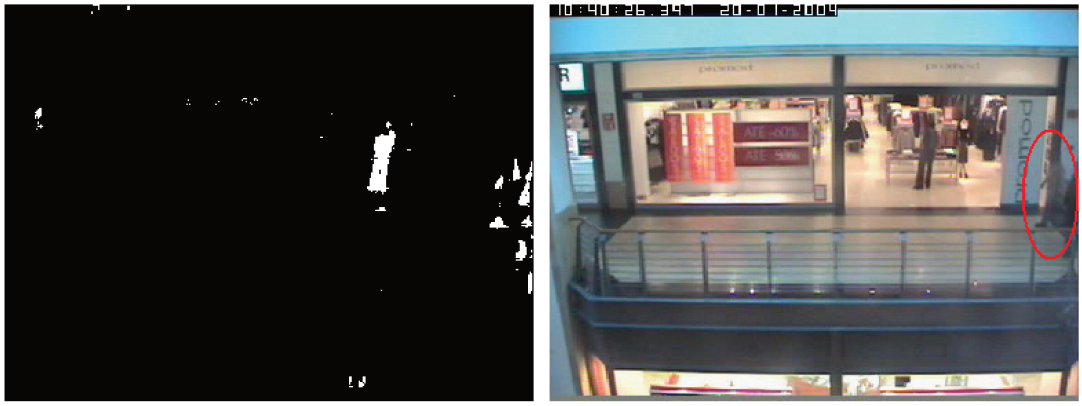


(a) $T = 0.2$ (b) $T = 0.5$ (c) $T = 0.8$

Fig. 3.7: MoG - Imagens do vídeo 1 para $\alpha = 0.01$.

A dificuldade em identificar a região de *foreground* no vídeo 1 está em lidar com os ruídos do vídeo, sombras e alvos que se confundem com elementos do *background* por possuírem cores semelhantes. A Figura 3.8 mostra o resultado da identificação do *foreground* para $T = 0.5$ e $\alpha = 0.005$ no quadro 432 do vídeo 1, onde um dos alvos passa pela cena confundindo-se com os ruídos do *foreground*.

Em relação ao vídeo 2, pode-se citar influência da iluminação externa na cena como principal causa de falsas identificações de *foreground*, especialmente na região inferior, próxima à janela. Variando-se o valor de T e α, pode-se reduzir o ruído, entretanto o *foreground* desejado é comprometido e o ruído não é totalmente eliminado. Sombras, embora pequenas, também são identificadas como *foreground*. No exemplo do quadro 208, mostrado na Figura 3.9, sendo a melhor segmentação obtida com $T = 0.5$ e $\alpha = 0.05$, o alvo confunde-se com os ruídos causados pela iluminação aparecendo "camuflado" entre o *foreground* identificado.



(a) $T = 0.5$ e $\alpha = 0.005$ (b) Quadro original

Fig. 3.8: MoG - Caso de falha na detecção do *foreground* no vídeo 1.

(a) $T = 0.2$ e $\alpha = 0.05$ (b) Quadro original

Fig. 3.9: MoG - Caso de falha na detecção do *foreground* no vídeo 2.

Para a análise quantitativa foram testadas diferentes combinações de T e α, onde $T = \{0.2, 0.5, 0.8\}$ e $\alpha = \{0.001, 0.005, 0.01, 0.05, 0.1\}$, combinando, para cada valor de T, valores de α. Os resultados desta análise são mostrados nos gráficos 3.10, 3.11 e 3.12, onde cada nó das linhas dos gráficos representa um valor de α, começando, da direita para a esquerda, em 0.1 e terminando em 0.001. O eixo x marca o valor da precisão e o eixo y o valor da taxa de revocação, desta forma, o resultado ideal é um ponto no valor $(1, 1)$, no canto superior direito do gráfico.

Através dos gráficos gerados, observa-se que a MoG apresentou comportamento parecido para os três vídeo analisado, apresentando taxas de precisão baixas quando a taxa de atualização é alta, taxas maiores de revocação para valores mais altos de T e comportamento não-linear, reduzindo a precisão quando α é demasiadamente pequeno, mantendo a revocação estável ou crescente. A não-linearidade foi notada na análise qualitativa onde observou-se que atualizando rapidamente o modelo de *background*, modelos gaussianos são gerados e descartados rapidamente, desta forma não

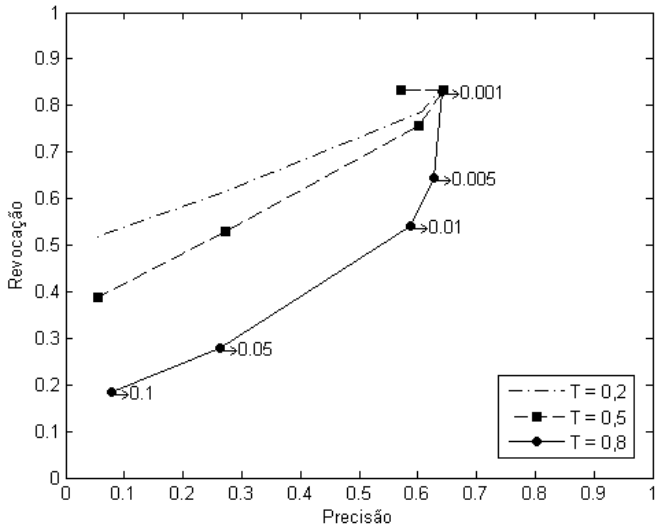


Fig. 3.10: MoG - Analise quantitativa do vídeo 1. Os valores nos gráficos mostram os valores de α utilizados.

conseguem modelar o fundo com precisão.

Para todos os vídeos obteve-se taxa de revocação relativamente alta (de 0.7 a 0.9), porém com taxas de precisão baixas, sendo que os melhores resultados foram obtidos com o vídeo 3, onde pode-se observar no gráfico 3.12, houve um equilíbrio próximo a 0.7 nas taxas de revocação e precisão, enquanto outros vídeos não obtiveram resultados superiores a 0.65 na taxa de precisão. A baixa taxa de precisão é justificada ao grande número de *pixels* erroneamente identificados como *foreground*, especialmente nos vídeos 1 e 2 por apresentaram bastante ruídos, como discutido anteriormente.

Os gráficos ainda mostram a tendência de obter-se resultados quantitativamente muito próximos com conjunto de parâmetros diferentes, como pode-se observar através da convergência das curvas e é confirmado pela analise qualitativa, como é mostrado na Figura 3.13 onde onde são apresentados resultados para a identificação do *foreground* da cena 1409 do vídeo 3 para três diferentes conjuntos de parâmetros.

Mesmo o resultado mostrado pela Figura 3.13, sendo o melhor obtido para o vídeo 3, apresenta algumas falhas como a identificação de *foreground* à esquerda da cena onde há uma árvore balançando.

3.2 *Eigenbackground* por fusão de autoespaços

Com o objetivo de verificar o efeito da variação de parâmetros no resultado do *Eigenbackground*, foram avaliados a influência dos valores do *threshold* th, da taxa de atualização M e da quantidade de autovetores utilizado na formação do autoespaço. Para os testes realizados, foram utilizados $M =$

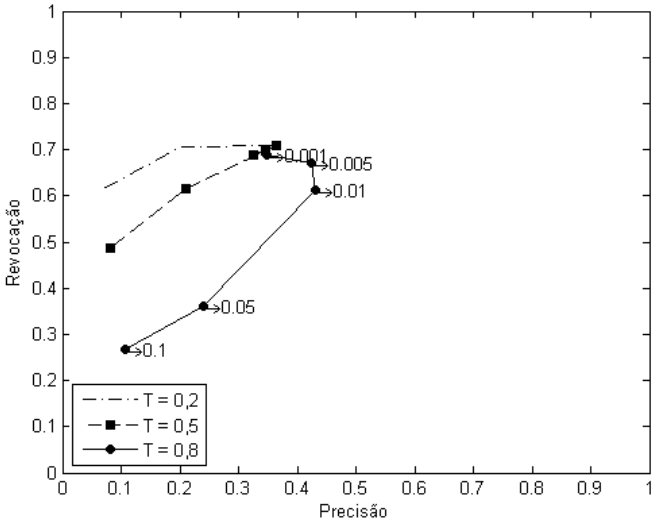


Fig. 3.11: MoG - Analise quantitativa do vídeo 2. Os valores nos gráficos mostram os valores de α utilizados.

$\{50, 100, 150, 200\}$, *threshold* $th = \{20, 25, 30, 35, 40\}$ e 2, 5, 10, 15, 20 e 25 autovalores, sendo que estes valores foram combinados entre si, gerando 120 conjuntos de parâmetros.

Os resultados obtidos com a variação do parâmetro M mostraram que o modelo é pouco sensível a este, gerando diferenças pouco significativas no resultado final. No caso de vídeo 2, mais sujeito a variações de iluminação, a variação de M mostrou resultado pouco mais visível, especialmente quando utilizado valores baixos de *threshold*, resultando em imagens mais ruidosas quando o número de quadros (valor de M) é menor. Um exemplo desta variação é mostrada na Figura 3.14, referente ao resultado obtido para um dos quadros do Vídeo 2 com $M = 50$(Figura 3.14(a)) e número de quadros, $M = 200$ (Figura 3.14(b)).

Quanto menor o número de quadros (M), mais rápida a incorporação dos elementos de *foreground* ao modelo de *background*, assim, se M é pequeno, mesmo objetos em movimento, em velocidade mais baixa, são incorporados, como mostra a região assinalada na Figura 3.14(a).

No caso do vídeo 3, onde a velocidade dos objetos é mais lenta, o valor do número de quadros (M) influenciou de forma significativa os resultados, como pode-se observar na Figura 3.15, onde um dos carros que está executando uma manobra no estacionamento é incorporado ao *background*.

A variação da quantidade de autovetores na formação do autoespaço não gerou diferenças significativas nos resultados obtidos. Fixando os demais parâmetros e variando apenas a quantidade de autovetores, utilizando 2, 5, 10, 15, 20 e 25 autovetores, o efeito notado foi uma pequena diminuição de ruídos classificados como *foreground*, como mostra a Figura 3.16.

Para ilustrar o efeito da variação do número de autovetores no resultado da subtração de fundo, um autoespaço foi gerado com uma sequência de 100 quadros contendo elementos de *foreground* de

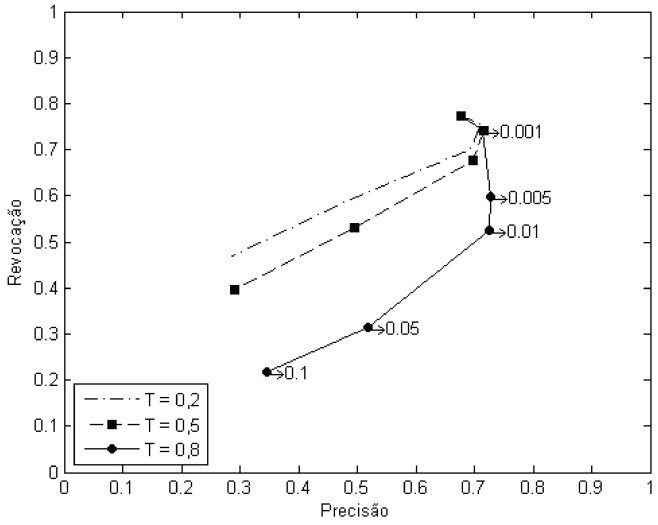
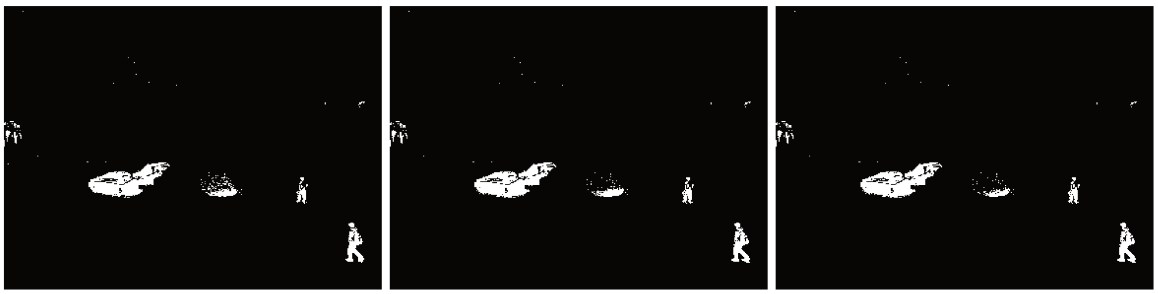


Fig. 3.12: MoG - Analise quantitativa do vídeo 3. Os valores nos gráficos mostram os valores de α utilizados.



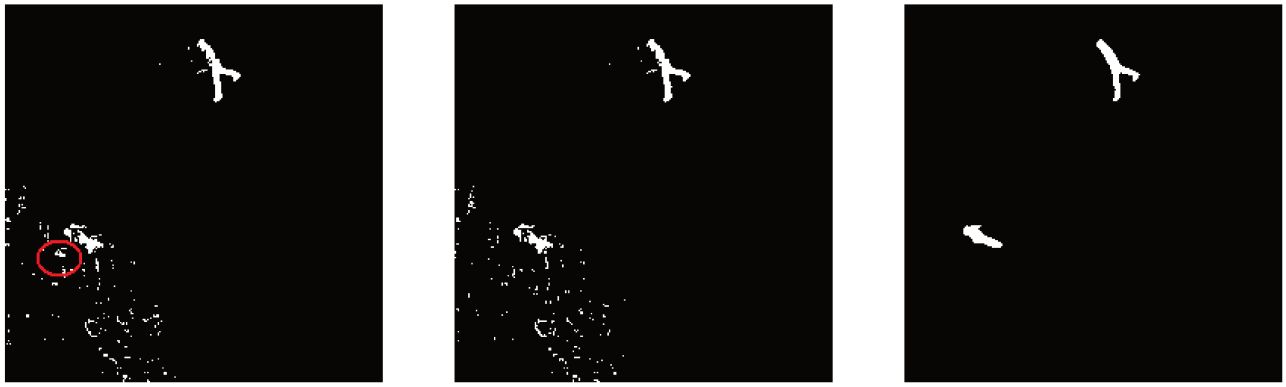
(a) $\alpha = 0.001$ e $T = 0.8$ (b) $\alpha = 0.005$ e $T = 0.2$ (c) $\alpha = 0.005$ e $T = 0.5$

Fig. 3.13: MoG - Imagens similares em conjuntos de parâmetros diferentes para o vídeo 3.

um vídeo da base de dados PETS de 2009, como mostra a Figura 3.17.

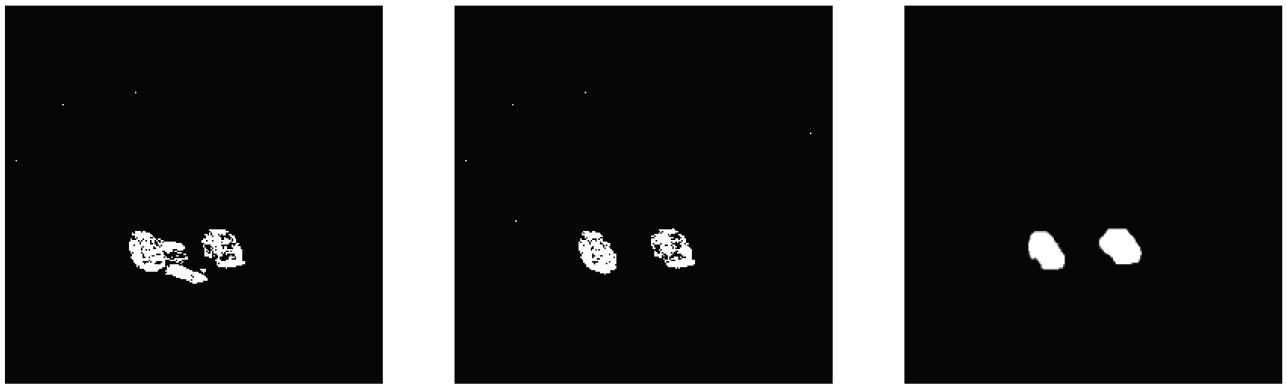
Durante o processo de formação do autoespaço, todo valor de *pixel* que se diferencia da imagem da média, ou ruídos, são estocados nos autovetores. Como são selecionados apenas os autovetores de maior magnitude, apenas as diferenças mais significativas, isto é, que mais se repetem, compõe o modelo de *background*. Se, durante a formação do autoespaço, algum objeto está se movendo na cena, como no caso da sequência de quadros mostrados na Figura 3.17, este também é estocado nos autovetores. Como resultado, estes objetos aparecem na reconstrução do autoespaço [1], como mostram as Figuras 3.18(c) e 3.18(d): quanto maior a quantidade de autovetores utilizados, mais destes objetos (ou ruídos) compõe o modelo de *background*.

[1] A reconstrução do autoespaço é realizada projetando uma matriz de valores 1 no autoespaço e re-projetando no espaço da imagem (equações 2.16 e 2.17). O valor 1 é escolhido por ser o elemento nulo da multiplicação, desta forma recuperando as informações dos autovetores.



(a) $M = 50$ (b) $M = 200$ (c) Resultado esperado

Fig. 3.14: *Eigenbackground* - Vídeo 2 e resultados para $th = 40$ e 15 autovetores.



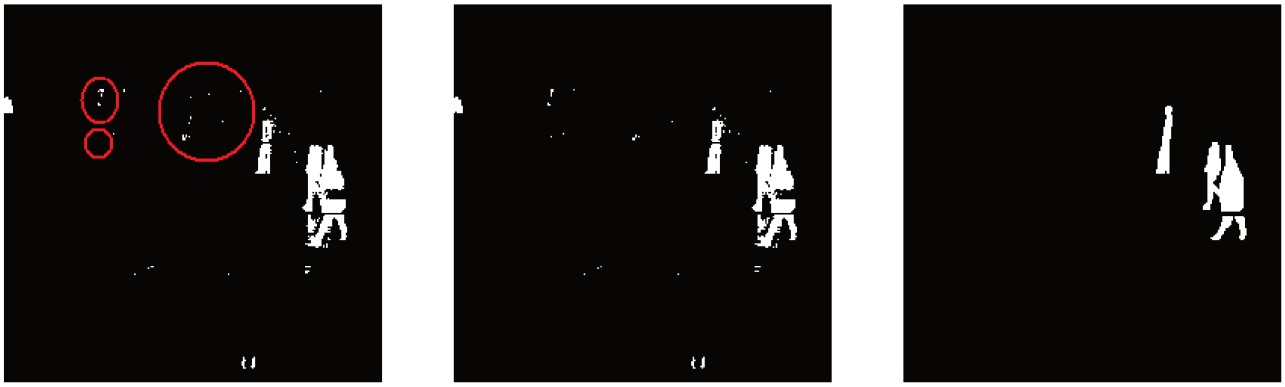
(a) $M = 50$ (b) $M = 200$ (c) Resultado esperado

Fig. 3.15: *Eigenbackground* - Vídeo 3 e resultados para $th = 30$ e 15 autovetores.

A Figura 3.19 mostra resultados da subtração de fundo para o quadro mostrado na Figura 3.18(a) e utilizando o modelo de *background* formado pela sequência mostrada na Figura 3.17. O quadro processado pertence ao conjunto de quadros utilizados no treinamento do *background*. Como resultado, observa-se na Figura 3.19 que, quanto mais autovetores são utilizados na identificação do *foreground*, menos os alvos existentes na Figura são identificados. Isso deve-se ao fato que tais alvos estavam presentes no treinamento do modelo de *background*, assim os valores dos *pixels* que os compõem são representados nos autovetores.

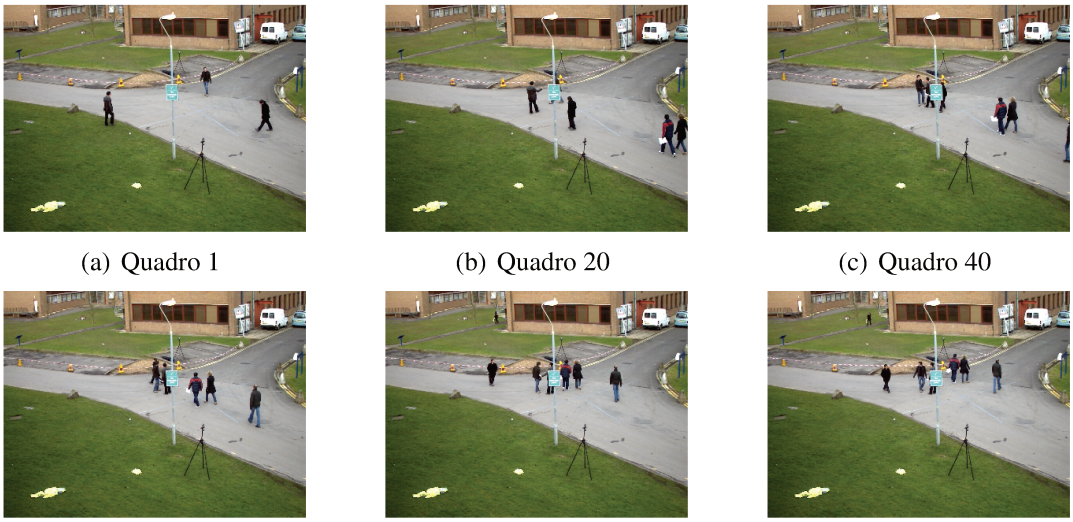
Por outro lado, nota-se que alguns alvos são incorporados ao modelo de *background* e, sua ausência no quadro processado (Figura 3.18(a)), resulta em falsas identificações de *foreground*, como pode ser observado nas Figuras 3.19(a) e 3.19(b).

A variação do valor do *threshold*, foi para todos os vídeos o fator que resultou em mudanças mais significativas no resultado final. Pode-se citar em particular o caso do vídeo 2, no qual um dos alvos que surge da parte inferior do vídeo, onde a cena é mais clara, é dificilmente identificado



(a) 1 autovetor (b) 25 autovetores (c) Resultado esperado

Fig. 3.16: *Eigenbackground* - Vídeo 2 com resultados para $th = 35$ e $M = 100$.



(a) Quadro 1 (b) Quadro 20 (c) Quadro 40

(d) Quadro 60 (e) Quadro 80 (f) Quadro 100

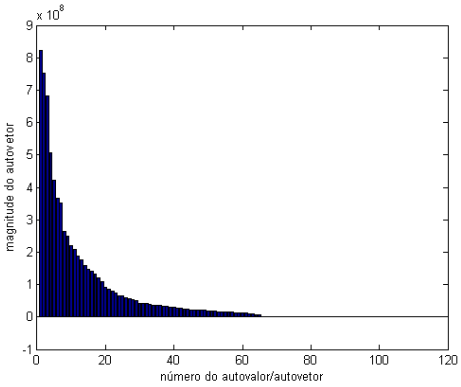
Fig. 3.17: *Eigenbackground* - Imagens do treinamento do autoespaço.

quando utilizados valores baixos de *threshold*, entretanto torna-se mais visível utilizando-se o maior *threshold* testado, como mostra a Figura 3.23(b).

Para a análise quantitativa, combinou-se para cada quantidade de autovetores, valores de *threshold*, fixando M em 100. Então os experimentos foram repetidos fixando 10 autovetores e variando-se o valor de M. Assim como na analise da MoG (seção 3.1), o comportamento dos gráficos foi similar para os três vídeos analisados, porém no caso do *Eigenbackground* os testes se comportam de maneira aproximadamente linear: quanto menor o *threshold*, menor a taxa de Precisão e maior a taxa de Revocação.

As Figuras 3.21, 3.22 e 3.23 mostram a avaliação quantitativa dos vídeos 1, 2 e 3, respectivamente. Pode-se observar que o video 2 obteve o pior desempenho referente ao valor Precisão, sendo que

(a) Imagem de entrada



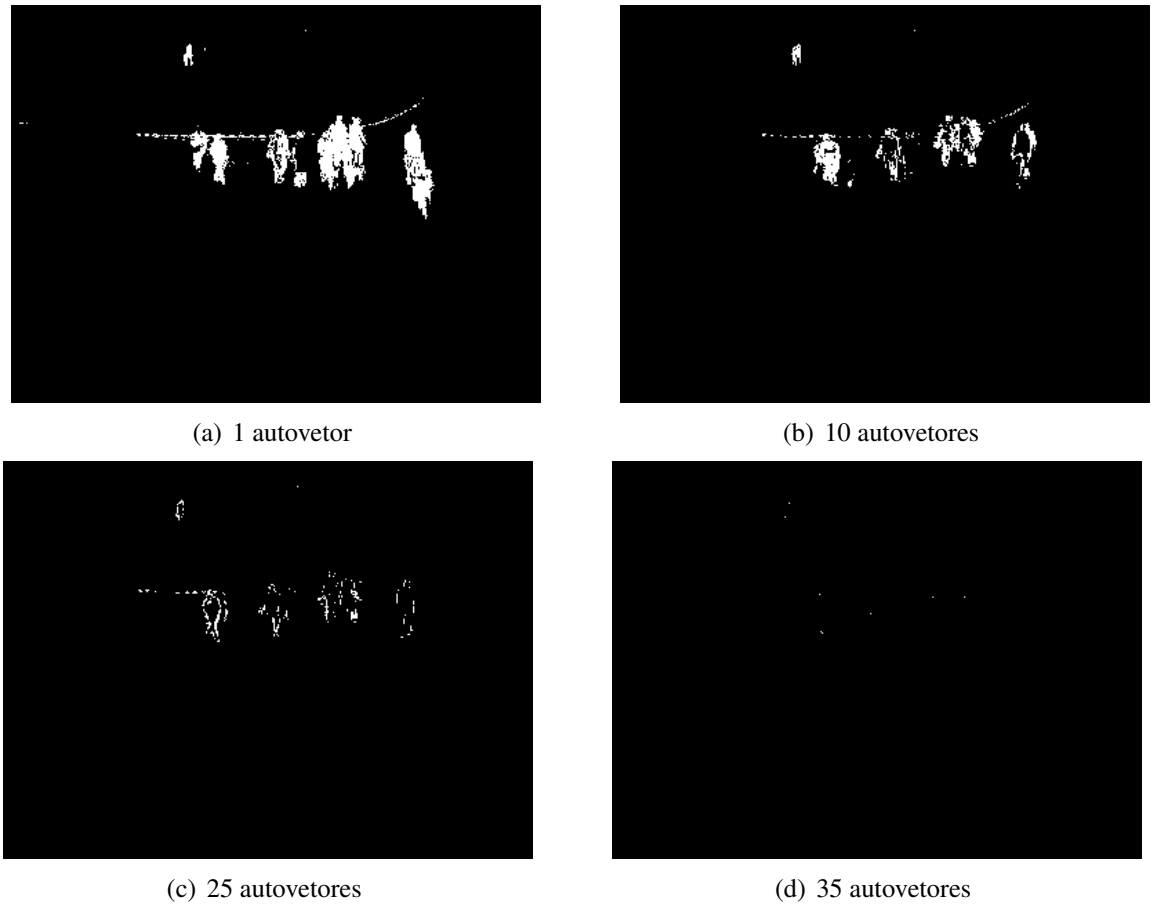
(b) Magnitude dos autovetores

(c) 1 autovetor

(d) 35 autovetores

Fig. 3.18: *Eigenbackground* - Reconstrução do autoespaço e subtração de fundo.

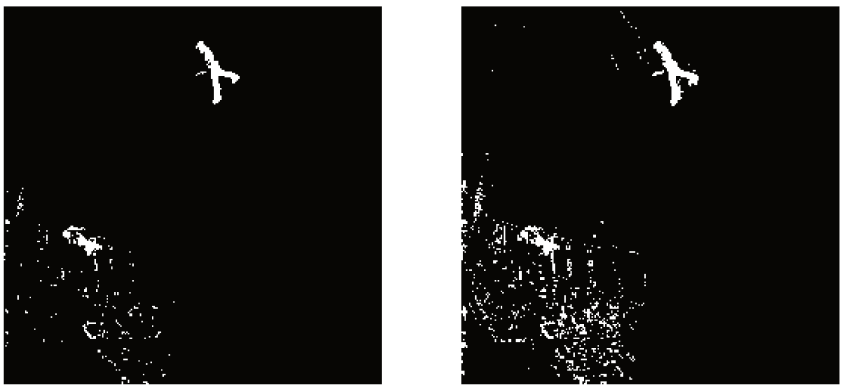
os resultados apresentaram muitos ruídos, além de conter sombras. Para este vídeo, os resultados obtidos pela MoG foram superiores, entretanto para os vídeos 1 e 3 pode-se observar a superioridade do *Eigenbackground* no sentido de conciliar as taxas de Revocação e Precisão, ainda que os resultados sejam próximos de 0.7.



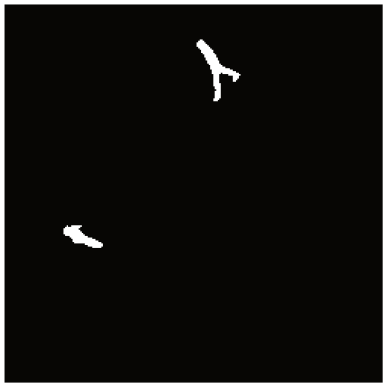
(a) 1 autovetor (b) 10 autovetores

(c) 25 autovetores (d) 35 autovetores

Fig. 3.19: *Eigenbackground* - Efeitos da quantidade de autovetores no resultado final.

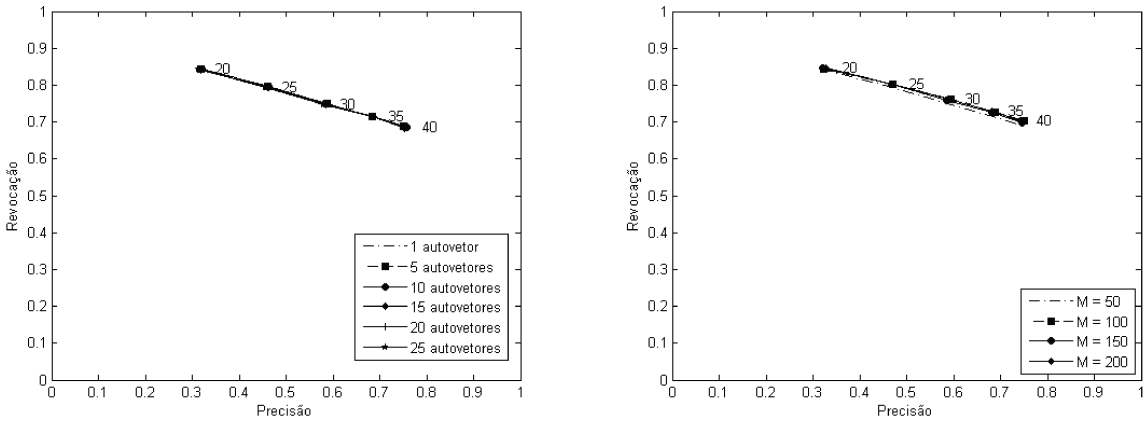


(a) $th = 40$ (b) $th = 25$



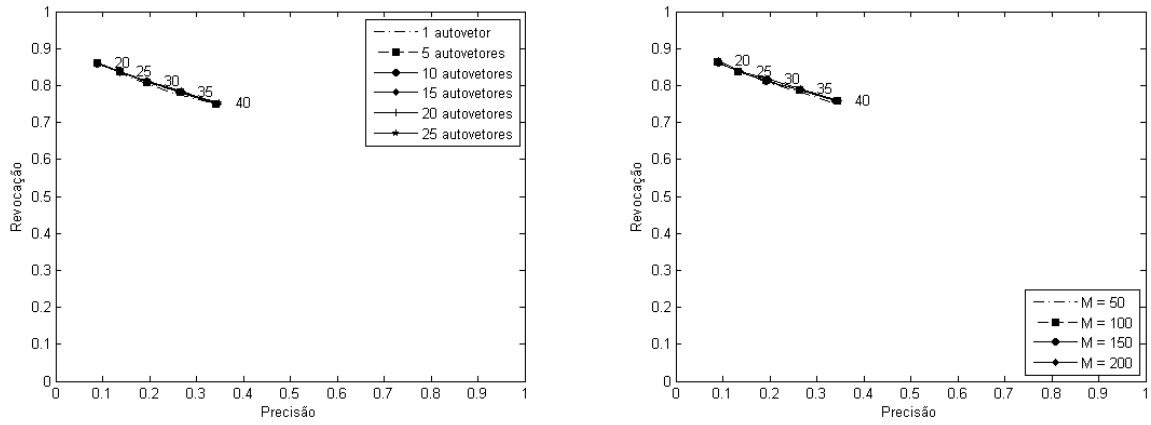
(c) Resultado esperado

Fig. 3.20: *Eigenbackground* - Vídeo 2 com resultados para 10 autovetores e $M = 100$.



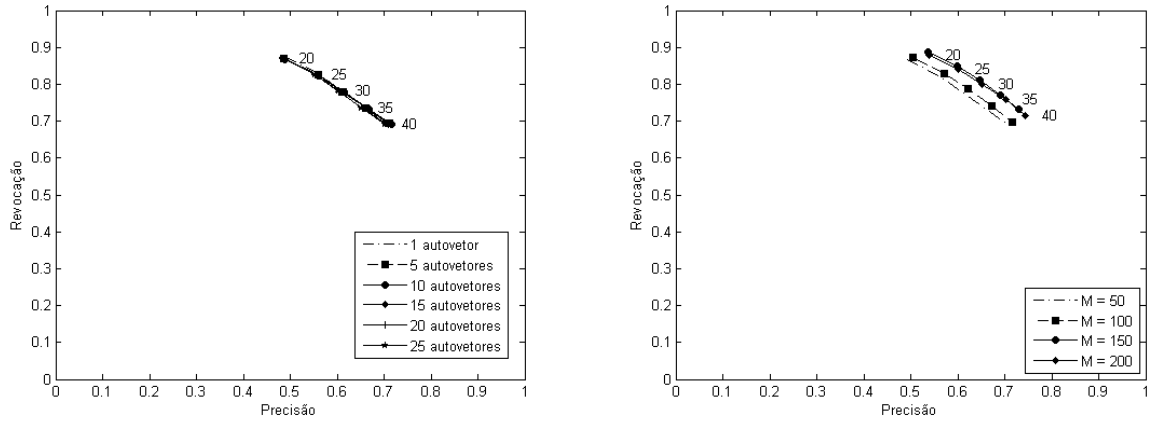
(a) $M = 100$ fixo (b) com 10 autovetores

Fig. 3.21: *Eigenbackground* - Analise quantitativa do vídeo 1. Os valores nos gráficos mostram os *thresholds* utilizados.



(a) $M = 100$ fixo (b) com 10 autovetores

Fig. 3.22: *Eigenbackground* - Analise quantitativa do vídeo 2. Os valores nos gráficos mostram os *thresholds* utilizados.



(a) $M = 100$ fixo (b) com 10 autovetores

Fig. 3.23: *Eigenbackground* - Analise quantitativa do vídeo 3. Os valores nos gráficos mostram os *thresholds* utilizados.

Capítulo 4

Sistemas de Rastreamento

O objetivo dos sistemas de rastreamento é seguir o alvo e gerar sua trajetória ao longo do tempo, localizando sua posição a cada quadro. Para isso, cada objeto é representado por um modelo de aparência que pode ser sua silhueta, centróide, *Bounding Box*, entre outros e, a partir deste modelo são definidos os tipos de movimentos ou deformações que o mesmo pode sofrer. Estes movimentos ou deformações são normalmente definidos por transformações de rotação, translação e/ou escala em conjunto com preditores lineares ou não lineares, com o objetivo de inferir sobre a posição do objeto no quadro seguinte, associando sua posição prevista com a posição estimada.

Um sistema de rastreamento está sujeito a diversas dificuldades em virtude da interação do alvo seguido com outros alvos e também com o ambiente, o que frequentemente torna a associação de objetos de um quadro ao outro uma tarefa ambígua. Desta forma, um rastreamento robusto significa lidar com fatores como oclusões, isto é, quando um objeto é totalmente ou parcialmente ocluído por outro ou por partes da cena como uma coluna, ou uma placa que se encontra entre o objeto e a câmera; desaparecimentos, que ocorrem quando um objeto sai da cena ou por algum motivo ele não é associado no quadro seguinte entretanto ele pode retornar, ou não, nas próximas cenas e entrada de novos objetos na cena. Além destas questões ainda é preciso considerar que podem existir falhas na etapa de subtração de fundo, que resulta em objetos de *foreground* que nem sempre são alvos a serem rastreados.

A abordagem mais comum de rastreamento trata da utilização de um estimador, linear ou não, que, baseado nas localizações anteriores dos alvos, gera uma estimativa de sua posição no quadro seguinte. Com base nesta estimativa, objetos da cena atual são relacionados a objetos da cena anterior através de uma função de custo, onde pretende-se minimizar a distância entre a posição estimada e a posição medida. Neste contexto, o Filtro de Kalman [16], consistindo na dicotomia previsão/correção, tem sido largamente empregado em sistemas de rastreamento. Entretanto, existe a dificuldade em relacionar as posições estimadas pelo filtro e as posições dos objetos medidas na imagem,

uma vez que duas posições medidas podem ser relacionadas a uma única estimativa ou então uma estimativa pode não se relacionar a nenhuma medida.

Com o objetivo de conciliar robustez e aplicabilidade em tempo real de sistemas de rastreamento, diversos trabalhos vêm sendo publicados, propondo diferentes abordagens para previsão da posição do objeto no quadro seguinte e gerenciamento dos alvos. A seguir serão apresentados alguns destes sistemas, buscando apresentar os trabalhos mais relevantes em termos de monitoramento de ambientes envolvendo uma única câmera estática e utilizando subtração de fundo para detecção do *foreground*. A tabela 4.1 mostra um resumo das técnicas apresentadas.

Koller *et al* [40] desenvolveram uma técnica de monitoramento de tráfego em rodovias. A imagem do *foreground* é obtida através da subtração de fundo abordada por Kilger [41] e, ao resultado desta subtração de fundo aplica-se o gradiente seguido de um *threshold*. Os objetos são rastreados a partir de *snakes* que definem seu contorno. Assumindo que os objetos sofrem apenas transformações afins de escala e translação. Dois filtros de Kalman são utilizados neste processo; um para estimar o contorno dos objetos no quadro seguinte e outro para estimar seu deslocamento. A posição de profundidade na cena de cada objeto é calculada a cada novo quadro, assim define-se se objetos podem ou não sofrer oclusões, que são identificadas por distorções no contorno que por sua vez levam a desvios na trajetória. Esta abordagem não considera oclusão total do objeto, uma vez que a câmera encontra-se acima dos alvos seguidos.

Wren *et al* [19] apresentaram um sistema em tempo real de rastreamento, “Pfinder", visando estabelecer uma interface humano-computador. A detecção do *foreground* é feita através do modelo Gaussiano proposto pelo mesmo autor, como mostrado no Capítulo 2. O mesmo modelo gaussiano é também utilizado para modelar os alvos seguidos. O rastreamento é realizado prevendo a aparência da pessoa rastreada no quadro seguinte utilizando informações de sua cor. Uma operação morfológica de crescimento de regiões é aplicada procurando o contorno do objeto e, baseado na coloração, procura-se verificar se as mãos e cabeça da pessoa rastreada são encontradas, caso contrário, o objeto é excluído da classe Pessoa. A cada quadro, o modelo estatísticos da pessoa encontrada é atualizado, bem como o modelo do fundo. O sistema funciona apenas para ambientes controlados e no rastreamento de uma única pessoa, falhando quando há oclusões das mãos ou cabeça da pessoa rastreada.

Intille *et al* [42] propuseram um sistema de rastreamento em tempo real utilizando o centróide dos objetos. O foreground é estimado através da modelagem por gaussianas apresentada em [19], seguido de operações morfológica de dilatação. Trata-se de uma versão modificada do sistema apresentado por Rangarajan e Shah [43], onde assume-se que objetos no mundo real percorrem caminhos suaves e percorrem uma distância pequena em um pequeno intervalo de tempo. Para associar objetos, um algoritmo polinomial é utilizado para reduzir a função custo de proximidade dos objetos entre dois quadros consecutivos, sendo que a correspondência inicial é feita através da aplicação de fluxo óptico

no gradiente dos dois primeiro quadros. Neste tipo de abordagem considera-se a entrada e saída de objetos analisando informações contextuais da cena, como portas, janelas etc, contudo não oferece tratamento para o caso de oclusão.

Haritaoglu *et al* [21] desenvolveram o sistema W4, visando sua aplicação em sistemas de segurança na identificação de pessoas. Para modelar o fundo é utilizada uma distribuição bimodal baseada nos valores dos pixels durante um período de treinamento. O rastreamento é feito através da silhueta dos objetos encontrados pela subtração de fundo. Para cada objeto detectado, é encontrado seu maior eixo e então, através de uma base de dados, estima-se a pose da pessoa seguida através de sua silhueta. Um algoritmo de previsão de movimento aplicado sobre a localização das mãos e cabeça junto à analise de textura da região estimada é utilizado para associar objetos quadro a quadro. O sistema ainda consegue rastrear pessoas que andam em grupo, identificando-as através da projeção em x e y do *foreground* (método chamado HYDRA[44]), além de verificar se as pessoas estão carregando algum objeto, ou até mesmo mochilas através da análise da simetria dos objetos encontrados e periodicidade do movimento.

Cupillard *et al* [45] propuseram uma abordagem para rastreamento de pessoas nas estações de metrô, onde o fluxo é intenso, o que dificulta rastrear pessoas individualmente, o que justifica a utilização de grupos para fazê-lo. A detecção de objetos em movimento é realizada aplicando-se um *threshold* à subtração da imagem atual de uma imagem do fundo. Cada objeto em movimento é computado como um nó de um grafo. Os objetos são associados quadro a quadro se a distância e similaridade entre o *Bounding Box* dos objetos forem mínimas. Uma estrutura chamada grupo, referente a grupo de pessoas é criada quando uma série de caminhos, definidos pela trajetória de regiões em movimento, pertence a um sub-grafo conexo. O trabalho não apresenta detalhes sobre outros tipos de tratamentos como oclusão por partes da cena ou além mesmo sobre a subtração de fundo.

Comaniciu *et al* [24] propuseram um novo conceito de sistema de rastreamento que influenciou trabalhos posteriores como os de Li *et al* [46] e Gallego *et al* [47]. Resumidamente, um espaço de características (cor, contorno ou textura) é escolhido para representar os objetos rastreados, assim cada alvo é representado neste espaço por uma função de densidade de probabilidade, sendo esta função centrada na origem do espaço. No espaço da imagem, o alvo é representado por um elipsóide que, para facilitar os cálculos, é normalizado para um circulo unitário. Para cada candidato, calcula-se a probabilidade de pertencer ao modelo do alvo, sendo que a similaridade é definida minimizando a distância de Bhattacharyya [26]. Os objetos são segmentados num modelo simples de subtração de fundo, o *Background-Weighted Histograms*, a fim de diminuir as regiões de busca de objetos. Entretanto este modelo de subtração . A pesar de apresentar resultados superiores a outros sistemas, este não lida com oclusões e há uma limitação quanto à forma do objeto, principalmente quando este

é deformável, uma vez que assume-se que sua forma é uma elipse.

Zhao e Nevatia [48] apresentaram um sistema de identificação de pessoas e reconhecimento de suas atividades, como andar, parar e correr para ambientes externos. Utilizando a subtração de fundo proposta por Wren [19] o *foreground* é identificado. Pessoas são identificadas através da identificação de picos na direção vertical que podem representar suas cabeças se satisfazer um limiar de quantidade de *pixels*. Este é um processo recursivo, iniciado através da analise dos picos mais próximos à câmera (identificados através de um modelo de câmera que permite identificar as coordenadas em 3D da imagem). As sombras também são identificadas no processo, assumindo que todas têm a mesma direção uma vez que a única fonte de luz é o Sol. Cada pessoa identificada é representada por uma elípse e suas características baseiam-se no centróide, eixos e rotação do mesmo, desta forma a melhor métrica de associação é aquela que minimiza a transformação de um elipsóide para seu equivalente no quadro seguinte. O estado de cada pessoa é previsto através da aplicação do filtro de Kalman nos centros e velocidade das elípses. O reconhecimento das atividades é realizado pela aplicação de fluxo óptico nos objetos reconhecidos, que oferece diversas vantagens sobre os métodos convencionais. A pesar dos bons resultado do sistema, oclusões severas não são tratadas, bem como casos de fragmentação dos alvos seguidos.

Lei e Xu [17] apresentaram um sistema de analise de vídeo em tempo real onde o *foreground* é obtido através da subtração de fundo MoG e objetos identificados são submetidos a um sistema de gerenciamento, sendo estes classificados entre sete categorias: Aparecimento, Maturidade, Oclusão, Temporariamente indisponível, Desaparecido, Reaparecido e Fora de cena. Tais categorias dão suporte a objetos em oclusão e desaparecidos dentro da cena, além de lidar com novos objetos e excluir objetos que saíram de cenas. A previsão de Filtros de Kalman de primeira e segunda ordem são utilizados para compor a métrica de associação de objetos, que além de levar em conta atributos como posição do centróide e sua velocidade, ainda utiliza atributos de área, cor, número de pixels e tamanho dos eixos da elipse contida no objeto. A pesar de ser um sistema robusto, existem falhas em situações onde há variação de luz, gerando falsos objetos de *foreground* e em cenas onde ocorrem muitas oclusões.

Leibe *et al* [49] visa detectar pedestres e estimar sua trajetória utilizando um sistema de otimização chamado Comprimento Mínimo de Descrição (MDL - *Minimum Description Length*) [50]. O trabalho é uma extensão da detecção de pedestres realizada por Leibe *et al* em 2005 [51], onde a segmentação dos quadros é realizada através da técnica de Mistura de Gaussianas, então um grupo de poses é testado na imagem segmentada através da métrica de Chamfer [52], com o objetivo de estimar onde encontram-se os pedestres. O sistema agrega informações ao longo do tempo revendo suas previsões quando novas evidências surgem ou comete algum erro.

Kong *et al* [53] propõem um sistema de rastreamento de pessoa para aplicações em lugares de

movimentação intensa como estações de trens. Objetos de *foreground* são detectados através do modelo Gaussiano de subtração de fundo. A associação dos objetos quadro a quadro é feita através de atributos de cor (um histograma modificado) da parte superior e de todo o objeto e área, quando nenhum objeto é associado, então um filtro de partículas descrito pelo mesmo autor num trabalho anterior [54] é utilizado para verificar se o objeto está sob oclusão ou sai de cena. A identificação de pessoas é feita através de modelos de silhuetas, desta forma, o modelo que tiver menor custo para se ajustar ao contorno de um objeto identificado é associado àquele objeto. O sistema mostrou bons resultados nos experimentos realizados, porém não é totalmente automatizado uma vez que a seleção da parte superior do objeto é feita manualmente.

Li *et al* [46] propuseram um sistema de rastreamento de pessoas onde o *foreground* é identificado através da técnica de mistura de Gaussianas. Cada alvo é representado por um *Bounding Box* e o rastreamento é feio através do algoritmo *Mean-Shift* aplicado sobre o centróide do *Bounding Box* de cada objeto. Tal método de associação é proposto por Comaniciu *et al* [24]. Durante oclusões, uma métrica de cor é aplicada para determinar o objeto que está sobrepondo o outro. Apenas as partes não oclusas dos objetos são utilizadas para atualizar seus atributos, baseados em suas cores. O algoritmo consegue rastrear objetos com até 70% de oclusão, entretanto não consegue identificar oclusões causadas por obstáculos na cena ou oclusão total do objeto.

Lu *et al* [28] desenvolveram um método para rastreamento de carros utilizando subtração de fundo baseada em transformada Wavelet. Considera-se neste método que os objetos são rígidos e não sofrem mudanças bruscas de direção de um quadro para o outro. O rastreamento é feito através da aplicação de um filtro de Kalman de primeira ordem sobre a posição do centróide e velocidade do *Bounding Box*, que delimita cada objeto seguido. Quando dois objetos entram em oclusão, a previsão do filtro de Kalman é utilizada para estimar a posição do centro do *Bounding Box*, que por sua vez é redimensionado de acordo com o *Bounding Box* da união dos dois objetos, estimando a posição dos objetos seguidos durante a oclusão. O método não trata de oclusão por elementos da cena e as falhas de rastreamento foram atribuídas às falhas na identificação do *foreground*.

Gallego *et al* [47] utilizaram dois métodos já conhecidos para rastrear objetos estáticos e em movimento em vídeos de segurança. Adotou-se como método de subtração de fundo o *Double-Background* e o modelo de rastreamento de objetos em movimento proposto por Comaniciu *et al* [24], baseado em *kernel*. No trabalho de Gallego *et al* ainda são tratadas as oclusões (quando dois componentes conexos são associados a um único componente conexo) e saída dos objetos de cena (quando nenhum objeto é associado ao alvo e este permanece sem se associado por um certo período de tempo). Para associar objetos estáticos verifica-se se há um objeto na área do alvo, caso contrário assume-se que o objeto se moveu ou que um objeto em movimento o está ocluindo.

Autores	**Finalidade do rastreamento**	**Modelo de subtração de fundo**	**Tratamento de oclusões**	**Forma de representação do objeto**
Koller *et al* [40]	Monitorar tráfego em rodovias	Kilger	Não trata oclusão total	Contorno
Wren *et al* [19]	*Smart-room*	Modelo Gaussiano	Trata, quando mãos e cabeça não estão em oclusão	Cor
Intille *et al* [42]	Não especificado	Modelo Gaussiano	Não	Centróide
Haritaoglu *et al* [21]	Segurança	W^4	Não trata oclusão total	Silhueta
Cupillard *et al* [45]	Segurança - metrô	Não há detalhes	Sim, mas não apresenta detalhes	*Bounding Box*
Comaniciu *et al* [24]	Não especificado	*Background-Weighted Histograms*	Não	Elípse
Zhao e Nevatia [48]	Segurança - ambientes externos	Modelo Gaussiano	Não trata oclusões severas	Elípse
Lei e Xu [17]	Não especificado	Mistura de Gaussianas	Sim	*Bounding Box*
Kong *et al* [53]	Segurança - trem, metrô	Modelo Gaussiano	Não especificado	Silhueta
Li *et al* [46]	Rastreamento de pessoas	Mistura de Gaussianas	Não trata quando ocorre por elementos da cena ou oclusão total	*Bounding Box*
Lu *et al* [28]	Rastreamento de carros	Wavelet	Não trata quando ocorre por elementos da cena	*Bounding Box*
Gallego *et al* [47]	Segurança	*Double-Background*	Sim	Elípse

Tab. 4.1: Sistemas de rastreamento.

4.1 Filtro de Kalman

Proposto por Kalman em 1960 [16], o filtro de Kalman trata de uma solução recursiva para a filtragem linear de dados e tem sido vastamente aplicado em sistemas de navegação, sistemas inerciais, controle de processos químicos, análise e processamento de imagens e sinais. Trata-se de um conjunto de equações matemáticas que permitem estimar o estado de um processo minimizando o erro quadrático médio[55].

A derivação do filtro de Kalman apoia-se no fato de que tanto os ruídos das equações de medição e transição como o vetor inicial de estado, são normalmente distribuídos. Desta forma, assume-se que o ruído do sistema tem parâmetros conhecidos, média nula, variância constante e não é correlacionado com o ruído das medições. Além disso, por se tratar de um sistema linear, espera-se que a transição de estados representada pelo modelo do sistema seja linear ou, pelo menos, linearizável em torno de um ponto [56].

Para sistemas de rastreamento, a utilização do Filtro de Kalman tem a vantagem de reduzir a área de pesquisa, uma vez que se dispõe de uma boa estimativa inicial do local em que cada elemento pode ser encontrado na imagem seguinte, assim reduzindo o custo computacional no método de correspondência, além de permitir o cálculo de estimativas de posição e velocidade, com as respectivas medidas de incerteza para cada quadro da sequência de vídeo e para todos os elementos característicos.

Para estimar o estado x_t de um processo no tempo através do filtro de Kalman, utiliza-se a equação estocástica linear:

$$x_t = Ax_{t-1} + Bu_{t-1} + w_{t-1}, \tag{4.1}$$

com a medição z_t dada por:

$$z_t = Hx_t + v_t. \tag{4.2}$$

As variáveis aleatórias w_t e v_t representam o ruído do processo e da medição, respectivamente. Assume-se que estes ruídos são independentes, branco e com distribuição normal:

$$p(w) \sim N(0, Q), \tag{4.3}$$

$$p(v) \sim N(0, R). \tag{4.4}$$

Na prática, a matriz de covariância do ruído do processo (Q) e a matriz de covariância do ruído da medida (R) mudam a cada iteração ou medição, mas assumimos que são constantes.

A matriz A, de tamanho $n \times n$, na equação 4.1, relaciona o estado no instante $t-1$ com o estado do processo no instante seguinte t na ausência de ruídos. A matriz B, de tamanho $n \times l$, relaciona a

entrada u ao estado x. A matriz H, de tamanho $m \times n$, na equação 4.2, relaciona o estado x_t com a medição z_t. Embora as matrizes A e H mudem a cada iteração ou nova medição, assume-se que estas matrizes são constantes.

A Figura 4.1 mostra a aplicação do filtro de Kalman na previsão da posição de um ponto que se desloca no plano. O círculo ao redor da posição estimada denota a área de pesquisa, a qual torna-se menor à medida que o filtro converge, isto é, a confiabilidade da previsão torna-se maior.

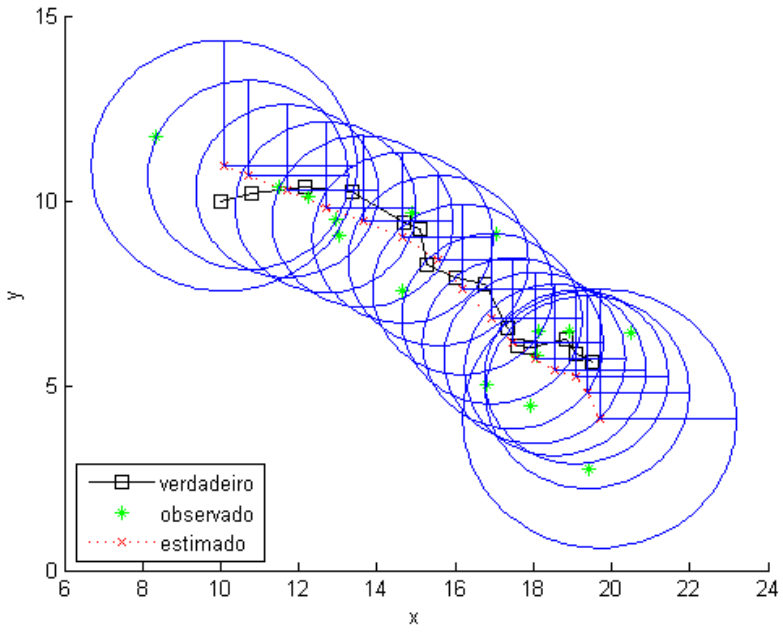


Fig. 4.1: Aplicação do Filtro de Kalman no rastreamento de um ponto.

4.1.1 Algoritmo do Filtro de Kalman Discreto

O filtro de Kalman estima um processo de forma recursiva, sendo constituído de duas fases complementares: previsão e atualização. As equações referentes à fase de previsão são responsáveis por incorporar novas medidas do processo à estimativa *a priori* (antes da medição), aperfeiçoando a estimativa *a posteriori* (após a medição). As equações de atualização são responsáveis em estimar o erro entre o estado do sistema estimado e o previsto, além de obter uma estimativa *a priori* do processo.

A previsão *a priori* do estado do processo é calculada através das equações:

$$\hat{x}_t^- = A\hat{x}_{t-1} + Bu_{t-1} \tag{4.5}$$

e

$$P_t^- = AP_{t-1}A^T + Q. \tag{4.6}$$

As matrizes A e B derivam da equação 4.1, enquanto a matriz de covariância Q deriva da equação 4.3. A matriz P_t trata-se da matriz de covariância do erro do sistema *a posteriori* no instante t. Na forma contínua, P_t é dada por:

$$P_t = E[e_t e_t^T], \tag{4.7}$$

onde

$$e_t \equiv x_t - \hat{x}_t. \tag{4.8}$$

Durante a fase de atualização, computa-se o ganho de Kalman (K_t, de tamanho $n \times n$). A matriz K_t deve minimizar a matriz da covariância do erro *a posteriori* (P_{t-1}):

$$K_t = P_t^- H^T (HP_t^- H^T + R)^{-1}. \tag{4.9}$$

O valor de K_t é inversamente proporcional ao valor da covariância do ruído da medição, desta forma, quanto maior o valor de K_t, menor o valor de R_t:

$$\lim_{R_t -> 0} K_t = H^{-1} \tag{4.10}$$

por outro lado, quando a covariância erro *a priori* P_t^- é reduzida, o ganho de Kalman terá seu valor reduzido:

$$\lim_{P_t^- -> 0} K_t = 0. \tag{4.11}$$

O próximo passo é medir o processo, incorporando esta medição (z_t) ao filtro para obter um estimativa *a posteriori* do estado do sistema:

$$\hat{x}_t^+ = \hat{x}_t^- + K_t(z_t - H\hat{x}_t^-). \tag{4.12}$$

A última equação de atualização fornece a estimação *a posteriori* da matriz de covariância do erro:

$$P_t^+ = [I - K_t H]P_t^-. \tag{4.13}$$

A Figura 4.2 mostra de forma resumida as etapas de predição e atualização do filtro para um

instante de tempo t.

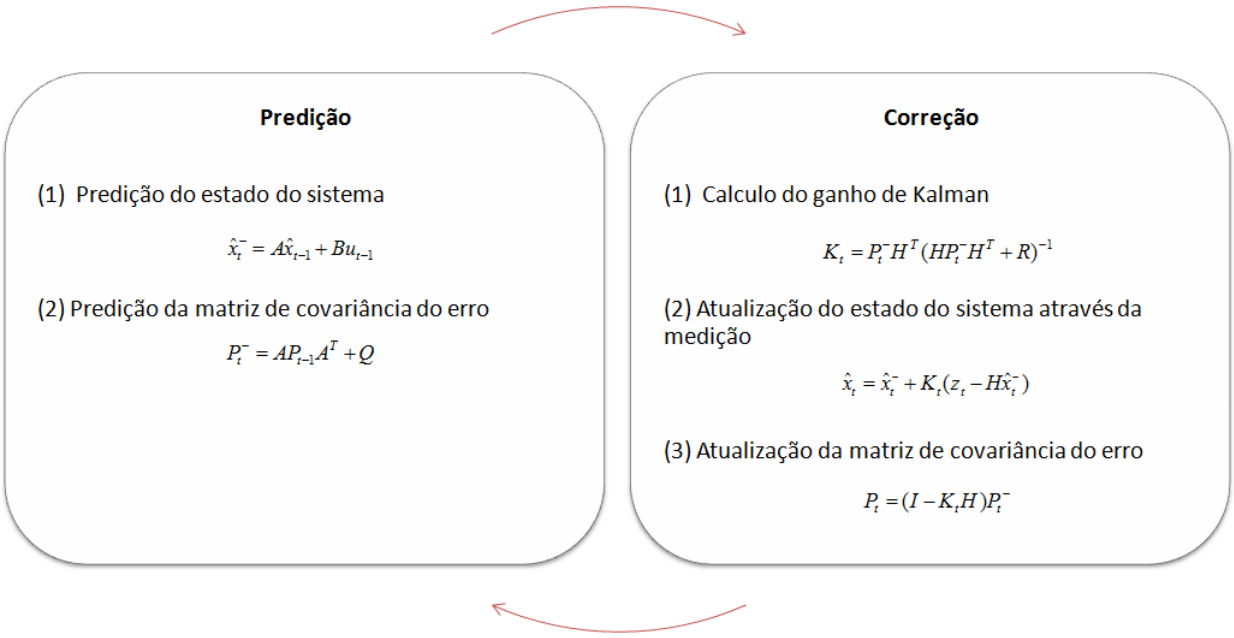


Fig. 4.2: Iteração do filtro de Kalman para o instante de tempo t.

4.1.2 Implementação do Filtro de Kalman no Sistema de Rastreamento Proposto

No sistema de rastreamento de objetos proposto, o filtro de Kalman é utilizado para estimar a posição dos alvos nos quadros seguintes. Desta forma, a cada novo quadro do vídeo, são fornecidas ao filtro a posição e velocidade instantânea de cada alvo, sendo que a posição de cada objeto é representada pela posição do centróide do mesmo na imagem, (c_x, c_y). Considera-se a velocidade e direção de deslocamento do objeto deve variar de acordo com o sistema:

$$\begin{cases} c_x = c_x^- + \Delta c_x \\ c_y = c_y^- + \Delta c_y \end{cases} \tag{4.14}$$

onde c_x^- (c_y^-) a posição x (y) do centróide do alvo no quadro anterior, e $\Delta c_x (\Delta c_y)$ a velocidade instantânea do alvo, calculada por:

$$\begin{cases} \Delta c_x = c_x - c_x^- \\ \Delta c_y = c_y - c_y^- \end{cases} \tag{4.15}$$

Desta forma, a matriz de mudança de estados A é dada por:

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.16}$$

e o vetor x, que representa o estado do alvo é definido como:

$$x = \begin{bmatrix} c_x \\ c_y \\ \Delta c_x \\ \Delta c_y \end{bmatrix}. \tag{4.17}$$

Considera-se também que não há erros de conversões de medidas, uma vez que a posição medida está no próprio plano da imagem, desta forma a matriz H é definida por:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \tag{4.18}$$

e a matriz R é iniciada como uma matriz identidade.

As equações do filtro de Kalman utilizadas no sistema de rastreamento proposto são as apresentadas na Seção 4.1.1, exceto pela equação 4.5. Nesta equação, a matriz B é considerada nula, uma vez que a posição do centróide do alvo na imagem é o estado do sistema, portanto não é preciso realizar transformações. Desta forma, a equação de previsão *a priori* é dada por:

$$\hat{x}_t^- = A\hat{x}_{t-1}. \tag{4.19}$$

4.2 Métricas de Associação

Como discutido no Capítulo 4, a etapa de rastreamento trata da associação de objetos quadro a quadro, que pode ser realizada através de atributos dos objetos como posição dos mesmos, área, cor, forma, textura etc, ou ainda combinações destes. Neste contexto, o filtro de Kalman, como descrito na Seção 4.1, é utilizado para auxiliar na associação de objetos através da posição dos mesmos, como exemplo os trabalhos [48, 17, 28]. A posição prevista pelo filtro de Kalman é associada a alguma posição medida numa dada imagem através de um cálculo de custo envolvendo a distância entre as duas posições.

Contudo, utilizar apenas o filtro de Kalman pode resultar em inconsistências no rastreamento, uma vez que os objetos rastreados podem alterar suas trajetórias, colidir, aparecer, partir, entre outras alterações que podem resultar em ambiguidades na associação de objetos, como um objeto não ser associado ou dois objetos serem associados a uma única previsão [56]. Para evitar este tipo de problema, outras características dos objetos, normalmente referindo-se à forma ou cor, são extraídas para auxiliar na tarefa de associação.

No presente trabalho, cada objeto é caracterizado por sua área cor e posição do centróide na imagem. Estes atributos foram escolhidos por não sofrerem mudanças consideráveis entre dois quadros consecutivos de um vídeo, além de serem características de fácil extração. Particularmente, o atributo de posição é utilizado no filtro de Kalman para gerar a previsão do objeto no quadro seguinte (veja a Seção 4.1), entretanto, para evitar ambiguidades na associação de objetos quadro a quadro, os atributos de área e cor também são utilizados na associação de objetos. Além disso, quando não é possível obter a posição do centróide do objeto (em caso de oclusão ou desaparecimento do objeto rastreado, por exemplo), os atributos de área e cor são mantidos, a fim de retomar o rastreamento quando a posição do centróide do objeto volta a ser medida.

A métrica utilizada para associar um objeto a outro no quadro seguinte varia de acordo com o estado do objeto, que será apresentado no Capítulo 5, podendo o objeto ser associado através da utilização simultânea de métricas de área cor e posição, ou apenas utilizando as métricas de área e cor.

Métrica de Posição

No presente trabalho, a posição de um objeto i é representada pela posição de seu centróide (x_i, y_i), no plano da imagem. Sejam (x_i^p, y_i^p) a posição prevista deste objeto no próximo quadro, gerada pelo filtro de Kalman, e (x_k^m, y_k^m) a posição medida de um objeto k qualquer no próximo quadro, então considera-se que k é uma candidato a ser associado ao objeto i se a distância $D_{i,k}$ for a menor entre as distâncias para todas as distâncias medidas, e pertencer a uma vizinhança fixa com

centro em (x_i^p, y_i^p).

Sejam j os objetos pertencente à cena seguinte, e (x_j^m, y_j^m) a posição de seu centróide, $D_{i,j}$ é definido por:

$$D_{i,j} = \sqrt{(x_j^m - x_i^p)^2 + (y_j^m - y_i^p)^2}. \tag{4.20}$$

Assim, o objeto i pe associado ao objeto k se a distância $D_{i,k}$ for a menor entre as distâncias calculadas e $D_{i,k}$ é menor que uma vizinhança v:

$$D_{i,k} = argmin_j(D_{i,j}), \tag{4.21}$$

$$D_{i,k} < v. \tag{4.22}$$

Métricas de Área e Cor

Além da métrica de posição, no presente trabalho são utilizadas métricas referentes a atributos de área e cor, compondo um critério de associação de objetos ao longo dos quadros do vídeo. A cada novo quadro, após a associação de objetos, tais atributos são atualizados de acordo com o novo estado do objeto.

A área de um objeto k, denotada por A_k, é definida pela quantidade de pixels que compõe o mesmo, assim, seja um objeto i e um objeto j pertencente ao quadro seguinte, a semelhança entre eles em relação à área é calculado por:

$$D_{i,j}^A = \frac{min(A_i, A_j)}{max(A_i, A_j)}. \tag{4.23}$$

Quando a área dos dois objetos forem iguais, então $D_{i,j}^A$ será igual a 1, por outro lado, quanto maior a diferença da área dos objetos, $D_{i,j}^A$ aproxima-se de 0, desta forma, deseja-se que o valor de $D_{i,j}^A$ mantenha-se próximo a 1, visto que normalmente um objeto sofre pouca mudança em sua área se considerar o curto período de transição de um quadro para outro do vídeo.

O atributo de cor de cada objeto é definido pela direção de maior variação dos canais R,G e B. Para tal, calcula-se o PCA dos valores dos pixels pertencentes ao objeto rastreado. Sejam I_i^r,I_i^g e I_i^b vetores coluna contendo, respectivamente, os valores R, G e B de cada pixel pertencente ao objeto i. O atributo de cor é obtido através de:

$$C_{cor} = [I_i^r I_i^g I_i^b][I_i^r I_i^g I_i^b]^T, \tag{4.24}$$

resultando numa matriz 3x3. Sobre a matriz C_{cor} são calculados autovalores e autovetores, sendo que

que o atributo de cor do objeto é definido pelo autovetor Φ_i referente ao maior autovalor.

Seja um objeto j no quadro seguinte, a semelhança entre este e o objeto do quadro anterior i é calculado através do valor do ângulo $\phi_{i,j}$ entre os respectivos autovetores, obtido pelo valor de seu cosseno:

$$D_{i,j}^{cor} = \cos \phi_{i,j} = \frac{\Phi_i \Phi_j^T}{||\Phi_i|| ||\Phi_j||}. \tag{4.25}$$

Como propriedade da função cosseno, quanto mais próximo de 1 for o valor de $D_{i,j}^{cor}$, menor o ângulo entre os autovetores, portanto mais parecida a distribuição de cor dos objetos comparados.

Capítulo 5

Sistema de Gerenciamento de Objetos

O sistema de gerenciamento de objetos utilizado no sistema de rastreamento de objetos proposto baseia-se no trabalho de Lei *et al*[17], no qual cada objeto identificado no vídeo é representado por uma das sete classes: Aparecido, Permanente, Ocluso, Temporariamente indisponível, Desaparecido, Reaparecido e Fora de cena. Cada classe representa o estado do objeto num determinado quadro do vídeo, desta forma, a cada novo quadro, o objeto pode transitar de uma classe para outra segundo regras de transição.

A classe Aparecido representa os objetos que acabam de entrar na cena. Os mesmos podem transitar para a classe Permanente se permanecerem na cena por um determinado número de quadros e obedecerem uma equação de movimento (equação 5.1). As classes Temporariamente Indisponível e Desaparecido tratam de objetos que não foram associados nos quadros seguintes. A classe Reaparecido trata de objetos que estavam nas classes Temporariamente Indisponível ou Desaparecido e voltaram a ser associados. A classe Ocluso trata da união de dois ou mais objetos. Finalmente, objetos pertencentes à classe Fora de Cena são aqueles que permaneceram na classe Desaparecido por um longo período de tempo, então os dados destes objetos são excluídos do sistema.

A vantagem da utilização de um sistema de gerenciamento de objetos, como o apresentando por Lei *et al*, é monitorar as interações dos objetos com a cena e com outros objetos, além de evitar o rastreamento de ruídos resultantes de falhas no processo de identificação do *foreground*. Além disso, a classificação de objetos é uma forma de tratar um dos principais problemas dos sistemas de rastreamento: a oclusão.

Entretanto, a proposta de Lei *et al* não classifica objetos que se separam como mostra, por exemplo, a Figura 5.1, onde duas pessoas que entram juntas na cena, sendo estas classificadas como um único objeto e, num determinado momento, se separam e cada pessoa segue em uma direção, gerando dois novos objetos. No sistema proposto por Lei *et al*, o objeto que contempla as duas pessoas juntas, após a separação seria classificado como Temporariamente Indisponível, depois Desaparecido e

finalmente, Fora de Cena. Por outro lado, os objetos que representam as pessoas que se separaram seriam classificados inicialmente como Aparecido, depois Permanente e este objeto não é associado ao objeto inicial (que representa as duas pessoas juntas), o qual é excluído do sistema. Além disso, existem casos em que o objeto se separa por ocorrer alguma falha no sistema de identificação do *foreground* ou por passar atrás de um obstáculo como um corrimão, como mostra a Figura 5.2. No sistema de gerenciamento de objetos proposto, o problema da separação é tratado através das classes Separado e Partes, permitindo rastrear um objeto dividido em partes, e também manter os dados do objeto antes da separação.

(a) Duas pessoas entram juntas na cena (b) Após a separação

Fig. 5.1: Caso de separação de objetos: duas pessoas entram juntas numa cena e se separam.

(a) Obstáculo separa a pessoa em 3 partes (b) Falha na identificação do *foreground* separa o grupo de pessoas

Fig. 5.2: Dois exemplos de separação: obstáculo e falha na identificação do *foreground*.

Uma outra diferença do sistema de gerenciamento proposto é a utilização de informações contextuais da cena para a classificação de objetos. Para cada vídeo, são definidas manualmente regiões

por onde os objetos podem entrar ou sair como portas, janelas, ou a lateral da cena, no caso de um ambiente aberto. Desta forma, é provável que objetos identificados fora destas regiões sejam apenas ruídos como por exemplo, árvores balançando. Para distinguir tais objetos, foram criadas as classes Iniciante e Fundo para classificar novos objetos que surgem na cena, dependendo da região onde este surge. Os dois objetos podem transitar para a classe Permanente, porém utilizando regras de transição diferentes.

Além das vantagens citadas anteriormente, o sistema de gerenciamento de objetos proposto participa na atualização do modelo de *background*, definindo quais objetos devem ser incorporado ao modelo, sendo estes classificados como Temporários.

O sistema de gerenciamento proposto possui algumas classes a mais que o sistema proposto por Lei *et al*, sejam elas para identificar o estado do objeto do objeto (como as classes Partes, Temporário, Iniciante e Fundo) ou apenas para manter informações dos objetos (como as classes Separado, Fictício e Ocluso). Ao todo, o sistema de gerenciamento de objetos proposto possui onze classes, como mostra a Tabela 5.1 que também apresenta uma breve descrição de cada classe.

Classe do objeto	**Significado**
Iniciante	Objeto aparece na cena e está numa região de entrada ou saída
Fundo	Objeto aparece na cena e NÃO está numa região de entrada ou saída
Fictício	Objeto criado para marcar a posição inicial do objeto da classe Fundo
Temporário	Objeto Fictício que será incorporado ao fundo
Permanente	Objeto Iniciante que é rastreado por um período de tempo e satisfaz equação de movimento ou objeto da classe Fundo que se desloca de sua posição inicial de forma que o objeto associado a ele, da classe Fictício, é associado a um novo objeto
Separado	Objeto da classe Permanente se divide em 2 ou mais partes
Partes	Objetos resultantes da classe Separado
Ocluso	Objeto está totalmente ou parcialmente sobreposto por outro no campo de visão da câmera
Indeterminado	Objeto resultante da união de dois ou mais objetos
Indisponível	Rastreamento temporariamente interrompido por ruídos ou por oclusão por partes da cena
Fora de Cena	Objeto encontra-se na classe Indisponível por um longo período de tempo ou sua última posição pertencia a uma região de entrada ou saída

Tab. 5.1: As classes do sistema de gerenciamento de objetos proposto.

O sistema de gerenciamento de objetos proposto organiza os objetos identificados numa lista, sendo que cada objeto possui um identificador, atributos de cor, área e posição (mostrados na Seção 4.2), classificação, um contador de número de quadros, e um identificador do objeto ao qual é rela-

cionado. Objetos pertencentes às classes Ocluso e Fictício, por exemplo, são relacionados a objetos Indefinido e Fundo, respectivamente. A Tabela 5.2 exemplifica esta organização.

A seguir serão apresentados as classificações adotadas em cada situação que o sistema de gerenciamento de objetos aborda: surgimento de novos objetos, separação de objetos, oclusão e desaparecimento, descrevendo brevemente o estado do objeto, destacando a classificação de cada um e as métricas utilizadas para associa-los no quadro seguinte.

5.1 A Classificação de Novos Objetos e Classe Permanente

5.1.1 Classificação de Novos Objetos: Classes Iniciante e Fundo

A classificação inicial de um objeto que aparece pela primeira vez na cena depende da localização do seu centróide e das regiões da cena marcadas como regiões de entrada ou saída, podendo ser classificado como Iniciante ou Fundo. As regiões de entrada ou saída são previamente e manualmente definidas pelo usuário do sistema, tratando-se de uma imagem binária onde regiões contendo *pixels* marcados com 1 definem as regiões de entradas ou saídas, que normalmente são portas, áreas livres, janelas, etc. A Figura 5.3 mostra dois exemplos de cenas e as regiões marcadas como regiões de entrada ou saída (em branco).

Se um objeto surge pela primeira vez na cena e a posição do seu centróide na imagem de entrada ou saída está na posição de um *pixel* com valor 1, então o objeto é classificado como Iniciante. Os objetos Iniciantes não são mostrados na saída do sistema de rastreamento pois pode se tratar de um alvo a ser rastreado ou apenas um ruído desta forma, um objeto pertencente a esta classe é rastreado e seus atributos são mantidos, porém só é destacado na saída do sistema de rastreamento se transita para a classe Permanente, que será apresentada mais tarde. A Figura 5.4(b) mostra objetos identificados pelo sistema de subtração de fundo (em branco). Os objetos à direita, destacados em vermelho, como pode-se verificar na Figura 5.4(a), compõe uma pessoa que acaba de entrar na cena. Estes objetos estão localizados numa região de entrada ou saída, como pode-se observar na Figura 5.3(b), que aponta estas regiões para o vídeo em questão, desta forma, estes objetos são classificados como Iniciantes.

Entretanto, se a posição inicial do centróide de um objeto está num *pixel* com valor 0, na imagem de entrada ou saída, o *pixel* é classificado como Fundo. Neste caso, um objeto auxiliar, classificado como Fictício é criado no sistema de gerenciamento de objetos. Um objeto pertencente à classe Fictício é sempre relacionado a um objeto da classe Fundo, herdando do mesmo seus atributos iniciais de cor, área e posição, porém, ao contrário dos atributos do objeto Fundo, estes atributos não são atualizados nos quadros seguintes. Assim como objetos pertencentes à classe Iniciante, objetos

(a) Quadro - Vídeo 1 (b) Regiões de entrada ou Saída - Vídeo 1

(c) Quadro - Vídeo 2 (d) Regiões de entrada ou Saída - Vídeo 2

Fig. 5.3: Cena de dois vídeos diferentes e regiões de entrada ou saída (em branco).

pertencentes à classe Fundo também não são mostrados na saída do sistema de rastreamento.

Na Figura 5.4(a), o objeto destacado em verde surge numa região marcada com 0, como pode se conferido na Figura 5.3(b), desta forma, o mesmo é classificado como Fundo e um objeto Fictício é criado no sistema de gerenciamento de objetos. A Tabela 5.2 mostra a situação do sistema de gerenciamento de gerenciamento de objetos para o quadro mostrado na Figura 5.4(a). Pode-se notar que o objeto de Id. 4, pertencente à classe Fictício, está associado ao objeto Fundo (Id. 1) e seus atributos de área cor e posição são idênticos.

Id.	Id. Objeto Relacionado	Classe	Área	Cor	Posição	Contador
1	1	Fundo	38	(0.90, 0.06, 0.42)	(23, 98)	2
2	2	Iniciante	454	(0.37, 0.68, 0.62)	(372, 155)	2
3	3	Iniciante	518	(0.53, 0.53, 0.64)	(377, 119)	2
4	1	Fictício	38	(0.90, 0.06, 0.42)	(23, 98)	2

Tab. 5.2: Sistema de gerenciamento de objeto para quadro mostrado na Figura 5.4(a).

(a) Objetos Iniciantes (b) Saída do sistema de rastreamento

Fig. 5.4: Objetos classificados como iniciante e saída do sistema de rastreamento. As regiões de entrada ou saída são mostradas na figura 5.3(b).

Os objetos pertencentes às classes Fundo e Iniciante são associados a objetos no quadro seguinte se métricas de posição, área e cor, apresentadas na seção 4.2 forem simultaneamente satisfeitas. Caso o objeto seja associado, seus atributos de área, cor e posição são atualizados no sistema de gerenciamento de objetos. Caso a associação não seja realizada, então o objeto é excluído do sistema de gerenciamento de objetos; se um objeto pertencente à classe Fundo é excluído, então o objeto Fictício relacionado a ele também é excluído.

5.1.2 Classe Permanente

A classe Permanente é a principal classe do sistema de gerenciamento de objetos, representando os objetos de interesse para o sistema de rastreamento. As uniões, separações e desaparecimentos que podem ocorrer na cena são definidas a partir das interações dos objetos Permanentes. Desta forma, objetos pertencentes à classe Permanente são destacados na saída do sistema de rastreamento, onde é mostrado o *Bounding Box* do objeto.

Tanto objetos pertencentes à classe Iniciante, quanto objetos pertencentes à classe Fundo podem transitar para a classe Permanente. Uma vez classificado como Permanente, um objeto pode transitar por outras classes e sempre voltar a esta, ao contrário dos objetos classificados como Iniciante e Fundo.

Para um objeto classificado como Iniciante transitar para a classe Permanente, o mesmo precisa satisfazer duas condições: tempo de permanência e fator de movimento, condições propostas no sistema de gerenciamento de objetos de Lei *et al*. A condição de tempo de permanência é satisfeita se o objeto Iniciante é rastreado continuamente por um número de quadros pré-definido pelo usuário do sistema de rastreamento. O número de quadros definido para satisfazer o tempo de permanência pode

depender da velocidade do vídeo (quantidade de quadros por segundo): quanto maior esta velocidade, maior o tempo de permanência do objeto. No sistema de gerenciamento de objetos o tempo de permanência é armazenado no atributo 'Contador', como pode ser observado na última coluna da Tab. 5.2, onde todos objetos estão com tempo de permanência igual a dois quadros.

O fator de movimento (denotado por mov) criado por Lei *et al*, visa identificar o tipo de movimento que o objeto está executando. Entende-se que ruídos costumam realizar movimentos oscilatórios na cena (como cordas balançando), desta forma, o fator de movimento atua como uma ferramenta para identificar estas oscilações, sendo definido por:

$$mov = \left(\frac{\sigma_{cx}^2}{\sigma_{vx}^2 + \xi} + \frac{\sigma_{cy}^2}{\sigma_{vy}^2 + \xi} \right) \frac{1}{2}, \tag{5.1}$$

onde $\sigma_{cx}^2(\sigma_{cy}^2)$ é a variância do centróide na direção $x(y)$, $\sigma_{vx}^2(\sigma_{vy}^2)$ é a variância da velocidade na direção $x(y)$ e ξ é uma pequena constante que evita divergência quando $\sigma_{cx}^2(\sigma_{cy}^2)$ e $\sigma_{vx}^2(\sigma_{vy}^2)$ são iguais a zero.

Ao fator mov é aplicado um limiar pré definido pelo usuário do sistema de rastreamento. Objetos com fator mov abaixo deste limiar são considerados ruídos e objetos acima do limiar são aqueles que satisfazem a condição do fator de movimento.

Objetos classificados como Fundo também podem transitar para a classe Permanente, sendo um caso particular de separação de objetos e será tratado com mais detalhes na seção 5.2.1.

5.2 A Separação de Objetos

Como citado anteriormente, uma das diferenças do sistema de gerenciamento de objetos proposto e o sistema proposto por Lei *et al* é o tratamento da separação de objetos. O objetivo deste tratamento é identificar quando o objeto sofre separação e monitorar o rastreamento das partes resultantes, a fim de voltar o objeto inicial (que se separou) à classe Permanente ou formar novos objetos classificados como Permanente.

O tratamento da separação é realizado apenas em objetos classificados como Permanente ou Fundo (ver Seção 5.2.1). Nos dois casos, considera-se que um objeto se partiu num determinado quadro se o mesmo não for associado aos objetos identificados e, na região que delimitada por seu *Bounding Box* transladado, surgem dois ou mais novos objetos. Desta forma, o objeto Permanente, que não foi associado, é classificado como Separado, os novos objetos são classificados como Partes e aparecem relacionados ao objeto Separado no sistema de gerenciamento de objetos (assim como objetos classificados como Fictício são relacionados a objetos classificados como Fundo).

A região onde os novos objetos podem surgir é delimitada pelo *Bounding Box* do objeto Perma-

nente que não foi associado, transladado pelo deslocamento previsto pelo filtro de Kalman. Desta forma, os novos objetos encontrados nesta região são classificados como Partes.

Os objetos pertencentes à classe Partes são associados no próximo quadro através das métricas de posição, área e cor, bem como objetos classificados como Separado, porém os atributos de área, cor e posição destes são calculados através da união dos objetos classificados como Partes, isto é, o atributo de cor, por exemplo, é extraído considerando-se as cores de todos os *pixels* pertencentes a cada objeto classificado como Parte associado ao objeto classificado como Separado, o mesmo ocorre com os atributos de área e posição. Assim, mesmo que o objeto classificado como Separado não seja associado a algum objeto, seus atributos ainda são atualizados, além disso, o objeto classificado como Separado é mostrado na saída do sistema de rastreamento, como os objetos classificados como Permanentes.

Se o objeto classificado como Separado é associado, então o mesmo transita para a classe Permanente e os objetos classificados como Partes, relacionados a ele, são excluídos do sistema caso não sejam associados a outros objetos ou, caso contrário, são classificados como Iniciantes, pois podem se tratar tanto de novos objetos, quanto de ruídos.

Pode ocorrer também do objeto Separado não voltar a ser associado e os objetos classificados como Partes relacionados a ele, se distanciarem a cada quadro, como por exemplo o caso das pessoas que entram unidas na cena e se separam (Figura 5.1). Neste caso a distância entre os objetos classificados como Partes torna-se maior a cada quadro. Para determinar se objetos estão se distanciando, as distâncias das partes em relação à posição do objeto Separado são armazenadas e, se estas distâncias foram crescentes durante um determinado número de quadros, então os objetos classificados como Partes transitam para a classe Permanente, formando novos objetos independentes e o objeto classificado como Separado, relacionado a eles, será excluído do sistema de gerenciamento de objetos.

5.2.1 Transição da Classe Fundo para a Classe Permanente

Devido à atualização do sistema de subtração de fundo, objetos que param na cena por um longo período de tempo, como um carro que estaciona, por exemplo, pode ser incorporado ao modelo de fundo. Nesta caso, quando o carro volta a andar, ele deve ser identificado como um novo objeto que está, provavelmente, fora das regiões de entrada ou saída, sendo portanto classificado como Fundo. Quando o objeto sai completamente do local onde estava parado, ou, como no exemplo do carro, da vaga onde estava estacionado, surge um espaço que é identificado como um novo objeto e precisa ser incorporado ao modelo de fundo. Além disso, o objeto classificado como Fundo deve ser classificado como Permanente e mostrado na saída do sistema de rastreamento.

A Figura 5.5 mostra um exemplo do carro estacionado num vídeo da base de dados PETS 2001 e a

transição de um objeto (o carro) que inicialmente é classificado como Fundo para a classe Permanente: o carro estacionado começa a se deslocar na vaga e o resultado é um novo objeto, destacado em verde, classificado como Fundo. A seguir,como pode se observar nas Figuras 5.5(c) e 5.5(d), o carro sai completamente da vaga, criando um novo objeto no local da vaga e o carro é classificado como Permanente. Após a separação, o espaço que o carro deixa na imagem é incorporado ao fundo, como mostra a Figura 5.5(f).

Desta forma, a transição da classe Fundo para a classe Permanente pode ser considerada um caso particular de separação de objetos, onde o objeto pertencente à classe Fundo se separa e as partes resultantes da separação são classificadas como Permanente e Temporário. Para que isso aconteça, uma das partes precisa ser associada ao objeto Fictício relacionado ao objeto classificado como Fundo que se separou através de métricas de área e posição.

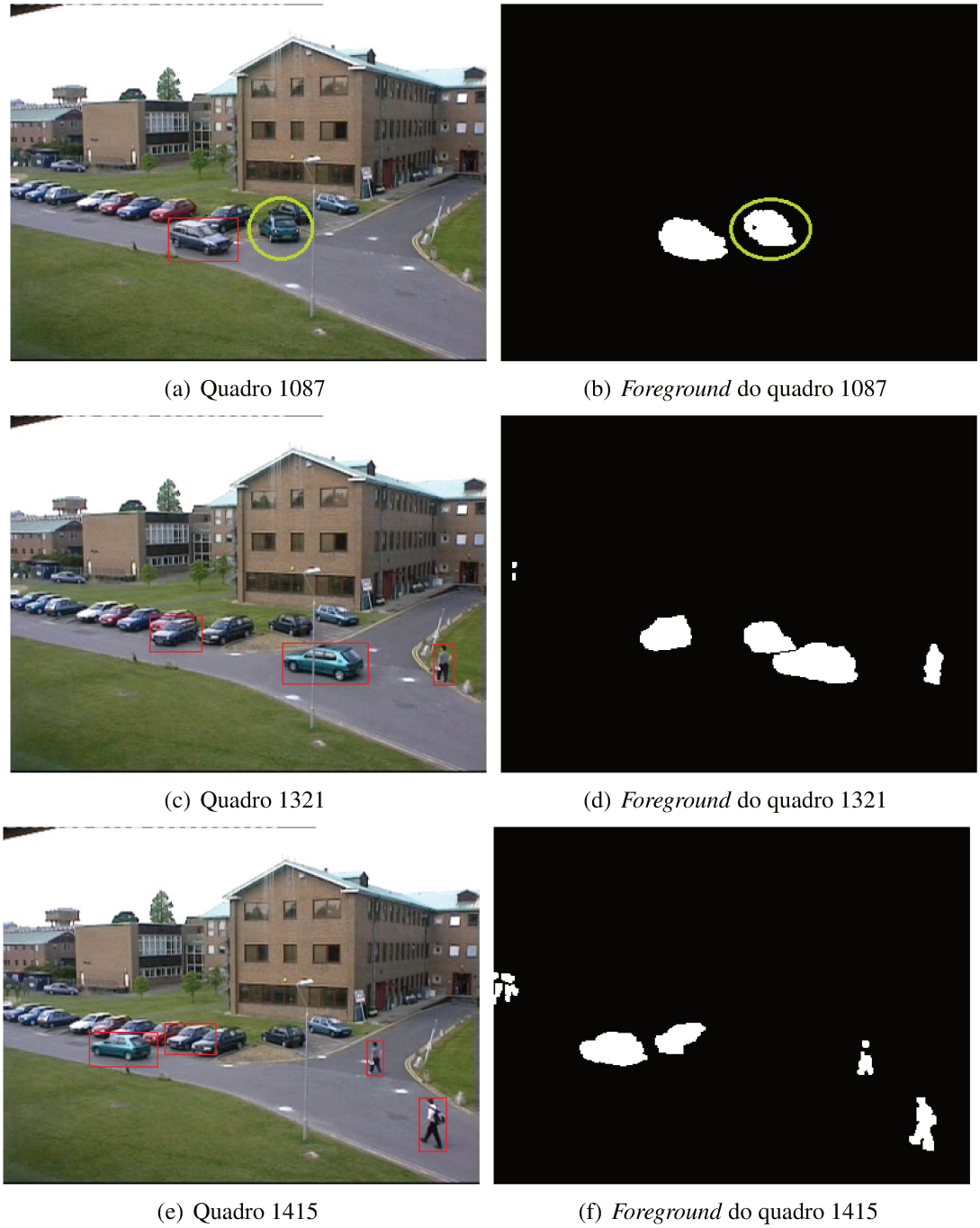
A Figura 5.6 mostra a separação do objeto classificado como Fundo, para o exemplo mostrado na Figura 5.5. Inicialmente, quando o carro começa a se mover, o objeto classificado como Fundo e o objeto Fictício relacionado a ele têm posições coincidentes, como mostra a Figura 5.6(a). À medida que o carro deixa sua posição inicial, a posição do objeto Fundo muda em relação à posição do objeto Fictício, como mostra a Figura 5.6(b). Quando o casso sai completamente da vaga, como mostra a Figura 5.6(c), o objeto Fundo se separa em duas partes e uma delas se associa ao objeto classificado como Fictício, então este transita para a classe Temporário. A outra parte não se associa, sendo então classificada como Permanente e o objeto classificado como Fundo é excluído do sistema de gerenciamento de objetos. O objeto classificado como Temporário será incorporado ao modelo de subtração de fundo, como mostra a Figura 5.5(f), e então é excluído do sistema de gerenciamento de objetos.

5.3 A Oclusão de Objetos por Outros Objetos

A oclusão ocorre quando o objeto rastreado não é visualizado (ou é parcialmente visualizado) devido à existência de um outro objeto, ou partes da cena, que bloqueiam sua visualização. A maior dificuldade de rastrear um objeto neste estado é a falta de informação para atualizar seus atributos (posição, área e cor), uma vez que os mesmos não podem ser diretamente medidos.

A oclusão por outros objetos é bastante comum em sistemas de rastreamento que utilizam uma única câmera devido à projeção dos objetos no plano da imagem, mesmo quando estes estão, na realidade, distantes. A Figura 5.7 mostra um exemplo de sobreposição de duas pessoas que andam separadas na cena de um vídeo da base de dados PETS 2007 porém, após alguns instantes, uma delas bloqueia parcialmente a visualização da outra (Figuras 5.7(b)).

Para o sistema de rastreamento proposto, dois ou mais objetos serão considerados em oclusão se



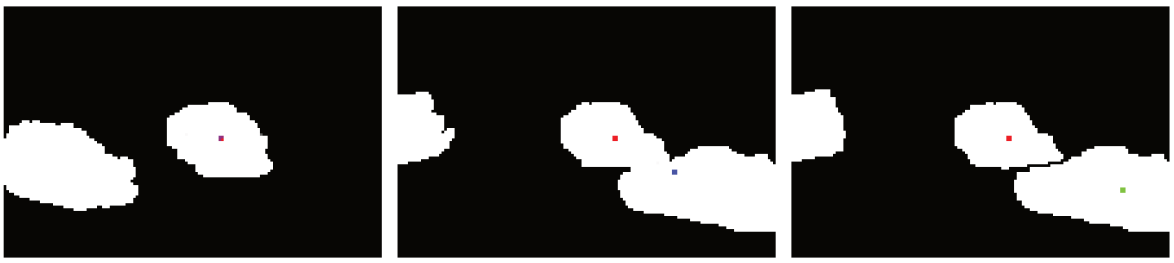
(a) Quadro 1087 (b) *Foreground* do quadro 1087

(c) Quadro 1321 (d) *Foreground* do quadro 1321

(e) Quadro 1415 (f) *Foreground* do quadro 1415

Fig. 5.5: Transição de um objeto classificado como Fundo (destacado em verde na figura 5.5(a)) para a classe Permanente.

for identificada a união dos mesmos através do resultado da subtração de fundo. Esta união pode ser causada pela sobreposição, como foi observado na Figura 5.7, ou pela união de dois objetos, como mostra a Figura 5.8, onde duas pessoas apertam as mãos, formando um único objeto. Oclusões parciais, causadas por obstáculos na cena serão tratadas como separações, como discutido na seção



(a) *Foreground* do quadro 1089 (b) *Foreground* do quadro 1317 (c) *Foreground* do quadro 1321

Fig. 5.6: Separação de objetos e transição da classe Fundo para a classe Permanente. O ponto vermelho indica a posição do objeto Fictício, em azul, o objeto Fundo e, em verde, um novo objeto que surge da separação.

(a) Detalhe do quadro 1214 (b) Detalhe do quadro 1230

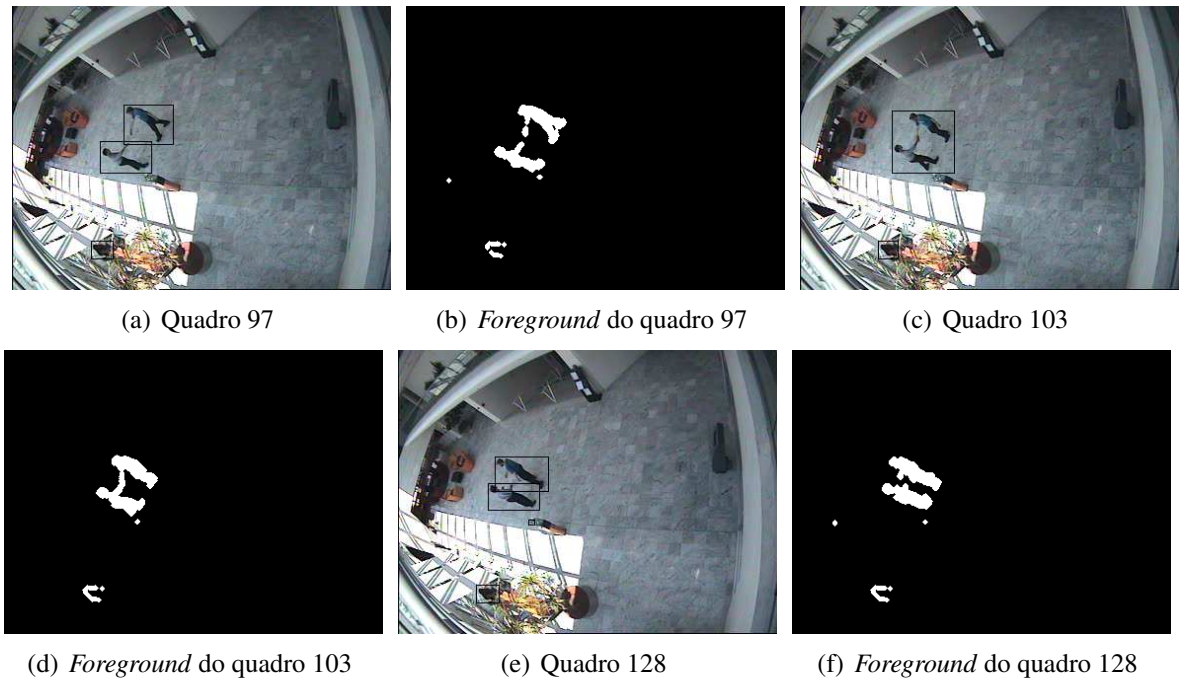
Fig. 5.7: Oclusão de alvo por sobreposição.

5.2. Já as oclusões totais, causadas por obstáculos na cena, serão tratadas como desaparecimento do objeto, como será discutido na seção 5.4.

Dois ou mais objetos, sendo que pelo menos um deles pertence à classe Permanente, são considerados em oclusão se não forem associados aos objetos identificados no quadro e suas posições previstas pelo filtro de Kalman estão contidas na região delimitada pelo *Bounding Box* de um novo objeto. A Figura 5.8 mostra um exemplo da base de dados CAVIAR 2001, onde duas pessoas que se encontram e apertam as mãos, formando um único objeto. Na Figura 5.8(f) pode-se observar que surge um novo objeto, maior, que poderia conter os dois objetos rastreados anteriormente, mostrados nas Figuras 5.8(a) e 5.8(b).

O objeto resultante da união é classificado como Indeterminado, podendo se tratar de uma oclusão, ou união de objetos. Os objetos que geraram o objeto Indeterminado e que não foram associados são mantidos no sistema de gerenciamento de objetos e classificados como Ocluso; seus atributos de área e cor são mantidos e estes serão relacionados ao objeto Indeterminado.

O objeto classificado como Indeterminado é associado quadro a quadro através das métricas de área, cor e posição e será mostrado na saída do sistema de rastreamento, como mostra a Figura 5.8(e), tendo seus atributos sempre atualizados, ao contrário dos objetos classificados como Ocluso



(a) Quadro 97 (b) *Foreground* do quadro 97 (c) Quadro 103

(d) *Foreground* do quadro 103 (e) Quadro 128 (f) *Foreground* do quadro 128

Fig. 5.8: Oclusão e separação de dois objetos.

relacionados a ele pois, como não pode-se determinar com exatidão a posição e os atributos de área e cor, tais atributos não são atualizados.

Os objetos classificados como Ocluso retornam à sua classe anterior num determinado quadro se o objeto classificado como Indeterminado não é associado e os objetos classificados como Ocluso relacionados a ele são associados a novos objetos através das métricas de área e cor. Caso dois ou mais objetos classificados não retornem à classe inicial e, na região delimitada pelo *Bounding Box* do objeto classificado como Indeterminado, transladado pelo deslocamento previsto pelo filtro de Kalman, surge um novo objeto, este novo objeto é classificado Indeterminado e os objetos não associados continuam classificados como Ocluso e são relacionados ao novo objeto Indeterminado. Quando o objeto classificado como Indeterminado não é associado num quadro, seus atributos são excluídos do sistema de gerenciamento de objetos.

A Figura 5.9 mostra uma sequência do oclusões retiradas do resultado do sistema de rastreamento proposto num vídeo da base de dados PETS 2001. Neste vídeo, dois carros, objetos classificados como Permanente, e um objeto classificado como Temporário entram em oclusão. Inicialmente, apenas um dos carros e o objeto Temporário estão em oclusão, como mostra a Figura 5.9(c), criando um objeto classificado como Indeterminado, destacado de verde na Figura 5.9(d). Os objetos classificados como Indeterminado e Permanente são classificados como Ocluso. Quando o carro classificado como Ocluso avança, como mostra a Figura 5.9(e), ele entra em oclusão com o outro carro classificado como Permanente, criando um novo objeto classificado como Indeterminado (Figura 5.9(f)), o

qual possui três objetos classificados como Ocluso relacionados a ele: dois carros e o espaço anteriormente classificado como Temporário. O objeto Indeterminado criado no quadro 1328 (Figura 5.9(d)) não é associado neste quadro, assim, este objeto é excluído do sistema. Nos quadros seguintes, um dos carros continua avançando (Figura 5.9(g)), desfazendo o objeto Indeterminado anterior, porém, o objeto classificado como Ocluso que anteriormente pertencia à classe Temporário é associado e volta à sua classe de origem e um novo objeto classificado como Indeterminado é criado (Figura 5.9(h)), tendo relacionado a ele os dois carros, classificados como Ocluso. Finalmente, no quadro 1412 (Figuras 5.9(i) e 5.9(j)), os carros voltam a se separar e retornam à classificação Permanente.

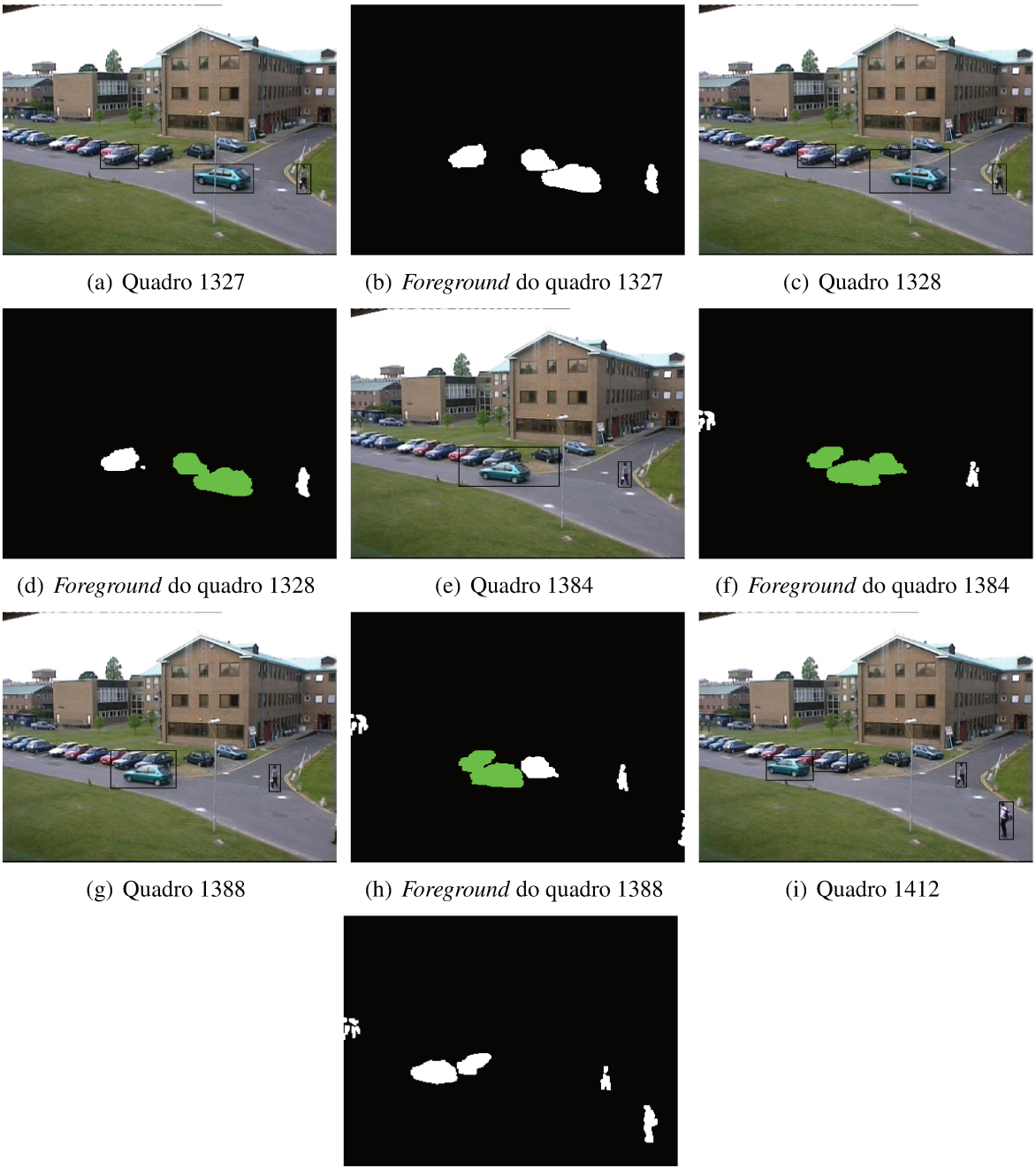
Pode-se notar que na saída do sistema de rastreamento para este exemplo, mostrada nas Figuras 5.9(a), 5.9(c), 5.9(e), 5.9(g) e 5.9(i), são mostrados apenas objetos classificados como Permanente e Indeterminado. Nota-se também na Figura 5.9(j) que o objeto classificado como Temporário, que sofreu oclusão foi incorporado ao modelo de fundo.

5.4 Desaparecimento de Objetos

Além da oclusão por outros objetos, um objeto pode sofrer oclusão por obstáculos da cena, como mostrado na Figura 5.10, onde uma pessoa, identificada como um objeto Permanente, passa atrás de uma placa, sofrendo oclusão (Figura 5.10(b)). Neste vídeo, da base de dados PETS 2009, o objeto não é associado aos objetos identificados na cena, sendo assim, sua classificação muda para Indisponível.

Os atributos do objeto classificado como Indisponível são mantidos para que possa ser associado novamente, caso ele reapareça na cena. Entretanto a associação é baseada nos atributos de área e cor, uma vez que não há como medir a posição exata do objeto na cena e, por consequência, o filtro de Kalman não pode ser atualizado, fazendo do atributo de posição uma métrica não confiável, uma vez que o objeto, enquanto estiver em oclusão, pode mudar a velocidade e direção de seu deslocamento. Quando o objeto classificado como Indisponível volta a ser associado, o mesmo retorna à classe Permanente.

Porém, se o objeto permanece nesta classe por um longo período de tempo a ser definido pelo usuário, esse é classificado como Fora de Cena e seus atributos são excluídos do sistema de gerenciamento de objetos. Contudo, o tempo pode ser menor se este objeto estiver localizado numa região de entrada ou saída, sendo que nestas regiões é mais provável que o objeto rastreado deixou a cena.



(a) Quadro 1327 (b) *Foreground* do quadro 1327 (c) Quadro 1328

(d) *Foreground* do quadro 1328 (e) Quadro 1384 (f) *Foreground* do quadro 1384

(g) Quadro 1388 (h) *Foreground* do quadro 1388 (i) Quadro 1412

(j) *Foreground* do quadro 1412

Fig. 5.9: Sequência de oclusões e separações. Em verde destacam-se os objetos classificados como Indeterminado.

(a) Três objetos classificados como Permanentes (b) Objeto Permanente transita para a classe Indisponível (destacado em verde)

Fig. 5.10: Oclusão de objeto e classificação como Indisponível.

Capítulo 6

Implementação do Sistema de Rastreamento Proposto

O sistema de rastreamento proposto neste trabalho é composto de três partes:

1. detecção do *foreground* através da subtração de fundo;
2. identificação de objetos e associação com os do quadro anterior;
3. classificação através do sistema de gerenciamento.

A primeira etapa trata da identificação do *foreground* através do modelo de detecção de fundo *Eigenbackground*. O modelo utilizado trata de uma modificação do modelo descrito na Seção 2.2, visando formar um modelo de subtração de fundo adaptado às condições de iluminação dos vídeos e integrado ao sistema de gerenciamento de objetos.

Com o *foreground* identificado, o próximo passo é identificar os *pixels* que compõem os objeto da cena, baseado em sua conectividade. Com os objetos identificados, suas posições, áreas e cores são extraídos e estocados para que no próximo quadro estes objetos sejam associados aos novos objetos identificados, realizando o rastreamento. A métrica de associação de objetos é realizada através da posição prevista pelo filtro de Kalman descrito na Seção 4.1, além da similaridade entre atributos de área e cor do objeto no quadro anterior e o novo objeto (Seção 4.2).

Finalmente, cada objeto rastreado é classificado no sistema de gerenciamento de objetos descrito no Capítulo 5. Este sistema permite ao usuário obter informações sobre o estado do objeto além de sua relação com outros objetos e com a cena. Este sistema também auxilia na continuidade do rastreamento do objeto ao longo dos quadros.

A seguir será descrito como cada etapa do sistema de rastreamento foi implementado.

6.1 Subtração de Fundo - *Eigenbackground* por Reinicialização de Autoespaço

O Capítulo 2 apresentou dois modelos de subtração de fundo: o MoG, vastamente utilizado em sistemas de rastreamento de objetos e o *Eigenbackground*, proposto por Oliver *et al*, baseado em PCA. Os resultados da análise de sensibilidade apresentados no Capítulo 3, mostraram que os dois modelos de subtração de fundo geraram resultados próximos quando testados em três vídeos com características diferentes.

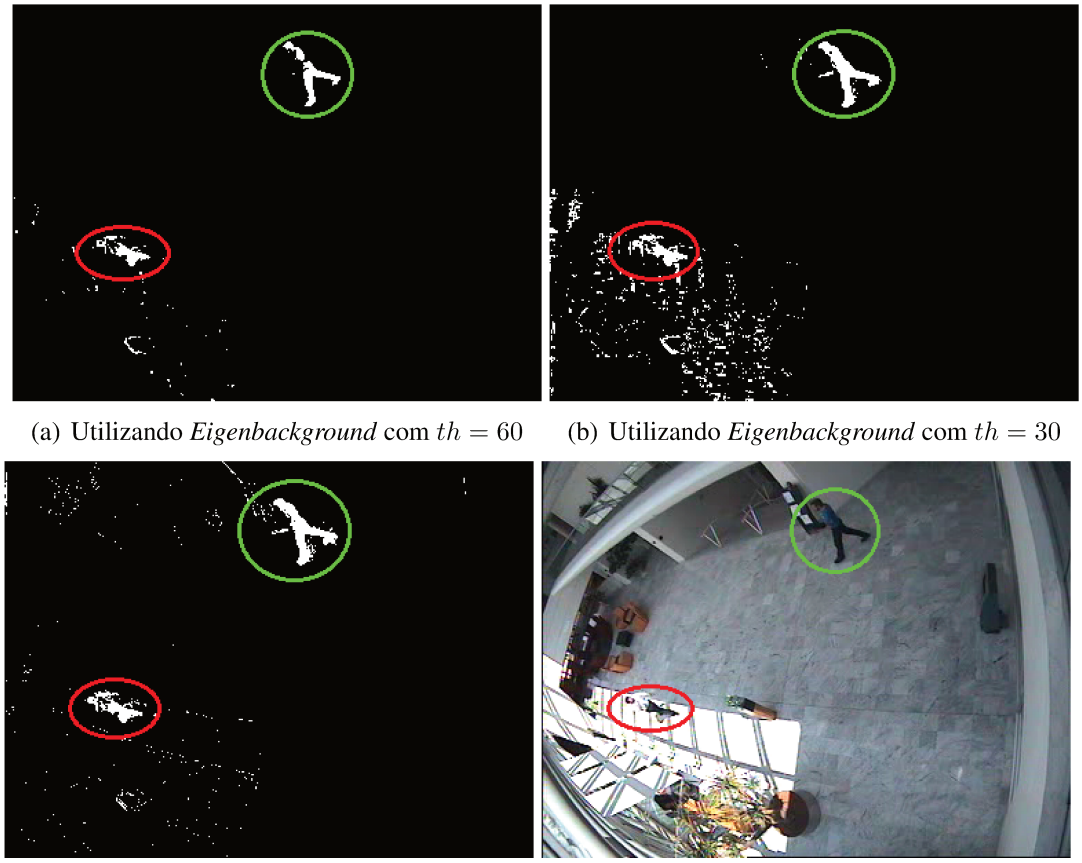
Xu *et al* [57] realizaram em seu trabalho uma comparação entre os modelos *Eigenbackground* e MoG, obtendo média de acerto de 0.669 e 0.509, respectivamente. No mesmo trabalho, o tempo de processamento obtido do *Eigenbackground*, sem o tempo para atualização do fundo, foi calculando, variando entre 21.1 e 42.2 quadros por segundo (variando o tamanho do quadro), fazendo deste um modelo aplicável em tempo real.

Além de apresentar resultados próximos ao modelo MoG, o modelo *Eigenbackground* também se destaca por sua fácil implementação e por ser menos sensível às variações de iluminação, desta forma o modelo é menos sensível à taxa de atualização, como pode ser observado nos testes realizados no Capítulo 3. Por estes motivos, o *Eigenbackground* foi escolhido como modelo de subtração de fundo para o sistema de rastreamento de objetos proposto.

O Autoespaço é formado como descrito na Seção 2.2.1. No caso do sistema proposto, a imagem vetorizada x_i é formada através da leitura coluna a coluna da imagem dos canais R,G e B, nesta ordem, assim, se a imagem colorida tem tamanho m x n, x_i tem tamanho $nm3$ x 1. O tratamento de imagens coloridas aumenta a dimensionalidade do problema, porém permite melhores resultados, como mostra o trabalho de Han e Jain [27].

Contudo, a forma com que o *threshold* é utilizado no modelo *Eigenbackground* gera, para alguns vídeos, resultados visualmente piores que o modelo MoG. Isso ocorre pois, ao contrário do modelo MoG onde existe um modelo para cada *pixel*, o *Eigenbackground* utiliza o mesmo autoespaço e *threshold* para toda a imagem. A Figura 6.1 mostra a influência da variação do *threshold* na identificação do *foreground* para vídeo da base de dados PETS 2001 utilizado nos testes realizados no Capítulo 3. Utilizando o modelo *Eigenbackground* e $th = 60$, como mostra a Figura 6.1(a), o objeto à esquerda do quadro, destacado em vermelho, aparece acompanhado de menos ruídos se comparado ao resultado obtido com $th = 30$, como mostra a Figura 6.1(b). Entretanto, o objeto à direita, destacado em verde, é melhor identificado quando se utiliza $th = 30$ e apresenta falhas quando utilizado $th = 60$. Para este quadro, o melhor resultado visual foi obtido através do modelo MoG, como mostra a Figura 6.1(c).

O problema ilustrado na Figura 6.1 ocorre pois a cena contém duas regiões com características



(a) Utilizando *Eigenbackground* com $th = 60$ (b) Utilizando *Eigenbackground* com $th = 30$

(c) MoG (d) Entrada

Fig. 6.1: Identificação do *foreground* através modelo *Eigenbackground* utilizando dois *thresholds* diferentes e através do modelo MoG.

diferentes como pode-se observar na Figura 6.1(d): a região próxima à janela, onde se encontra o objeto destacado em vermelho, é influenciada pela iluminação externa e portanto apresenta mais ruídos se comparada ao restante da cena, onde se encontra o objeto destacado de verde e a iluminação é praticamente constante, resultando em menos ruídos. Em geral, regiões mais claras da cena costumam apresentar mais ruídos se comparadas às regiões mais escuras em virtude da variação da iluminação.

Para contornar o problema das regiões com características diferentes, as imagens utilizadas no treinamento do modelo *Eigenbackground* foram particionadas manualmente em regiões baseando-se nas diferenças de iluminação da cena e, para cada uma destas regiões, foram formados diferentes autoespaços seguindo o formato apresentado na Seção 2.2.1.

A a subtração de fundo também é realizada particionando-se o quadro de entrada, permitindo aplicar em cada partição, que contém seu próprio autoespaço, um *threshold* diferente. O resultado da subtração de fundo em cada parte é unido, formando o resultado para o quadro de entrada, antes da partição. A Figura 6.2(a) mostra o particionamento realizado para um vídeo da base de dados PETS

2001 e a Figura 6.2(b) o resultado da subtração de fundo com o particionamento para o mesmo quadro mostrado na Figura 6.1(d), utilizando $th = 25$ e $th = 80$ (*thresholds* escolhidos através da análise visual dos resultados de suas variações) para as regiões 1 e 2, respectivamente. Pode-se perceber visualmente que o resultado obtido com o particionamento (Figura 6.2(b)) foi melhor que o resultado obtido anteriormente sem o particionamento, mostrado nas Figuras 6.1(a) e 6.1(b).

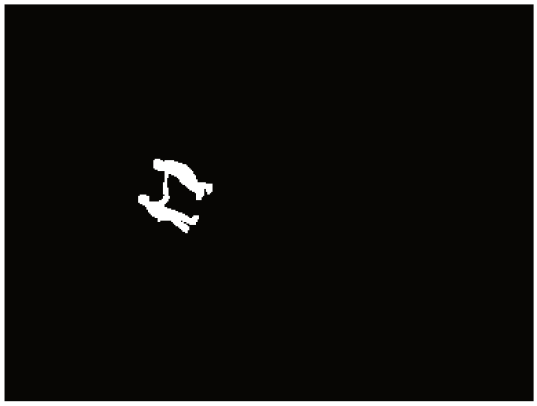
(a) Particionamento do quadro (b) Resultado da subtração de fundo

Fig. 6.2: Particionamento das imagens em duas regiões e o resultado obtido.

A Figura 6.3 mostra outro exemplo de resultado obtido com a partição do quadro para o mesmo vídeo mostrado na Figura 6.2. Com a utilização da partição obteve-se 0.6081 de taxa de Precisão e 0.7868 de taxa de Revocação, resultado superior aos resultados apresentados pelo modelo MoG para o mesmo vídeo, como mostrado no gráfico apresentado na Figura 3.11, e aos resultados apresentados pelo modelo *Eigenbackground* sem partição, como mostrado na Figura 3.22.

As avaliações do modelo *Eigenbackground* apresentadas no Capítulo 3 mostraram também que a taxa de atualização do modelo (valor de M nos gráficos mostrados nas Figuras 3.21, 3.22 e 3.23) pouco influenciou nos resultados da subtração de fundo, principalmente por se tratar de um modelo menos sensível às variações de iluminação se comparado ao modelo MoG. Assim, a atualização do modelo de *background* atuou principalmente na incorporação de elementos do *foreground* ao modelo de fundo (como ressalta a Figura 3.15, referente ao Vídeo 3 da avaliação dos sistemas de subtração de fundo). Desta forma, a atualização do *Eigenbackground* utilizada no sistema de rastreamento proposto foi através da reinicialização do autoespaço, auxiliada pelo sistema de gerenciamento de objetos, visando reduzir o esforço computacional de calcular a fusão de autoespaço.

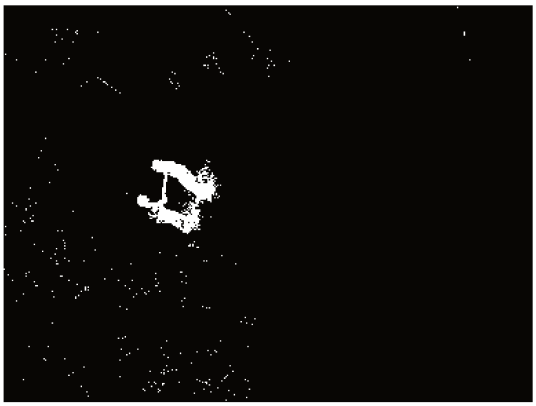
No caso da reinicialização do autoespaço, o sistema de gerenciamento de objetos define quais objetos devem ou não ser incorporados ao novo autoespaço que deve substituir o autoespaço atual. A cada quadro do vídeo, os objetos classificados como Iniciante, Permanente, Indefinido e Parte têm sua área removida do quadro processado e substituída pela mesma área na imagem média do autoespaço atual. A Figura 6.4 ilustra este processo para um quadro de um vídeo da base de dados PETS 2006.



(a) Resultado esperado

(b) *Eigenbackground* com particionamento $th = 25$ e $th = 80$, $M = 100$ e 10 autovetores

(c) *Eigenbackground* se particionamento, $th = 30$, $M = 100$ e 10 autovetores



(d) MoG com $T = 0.8$ e $\alpha = 0.01$

Fig. 6.3: Resultado da subtração de fundo através dos modelos *Eigenbackground* com e sem particionamento e MoG.

Os objetos classificados como Fundo e Temporário não são extraídos da imagem de entrada, desta forma, estes objetos devem ser incorporados ao novo autoespaço gerado.

Cada quadro do vídeo processado pelo sistema de rastreamento é armazenado e após um número de quadros pré definido pelo usuário, o novo autoespaço é re-calculado como descrito na Seção 2.2.1, substituindo o autoespaço atual.

6.2 Representação dos Objetos Rastreados

A imagem resultante da aplicação do *Eigenbackground* é uma imagem binária, onde os *pixel* marcado com 1 representa o *foreground* e os marcado com 0 representam o *background*. Os objetos são definidos analisando a 8-vizinhança de cada *pixel* marcado como *foreground*. Cada componente conexo é chamado de objeto e objetos contendo 1 *pixel* são considerados ruídos são descartados.

Como normalmente a subtração de fundo resulta em objetos disformes, duas operações morfológicas são empregadas com o objetivo de melhorar a forma dos objetos. A primeira operação, com o objetivo de preencher espaços vazios dentro de um contorno fechado é a operação de preenchimento e, em seguida uma operação de dilatação é realizada, uma vez que, para reduzir ruídos, na identificação do *foreground*, um valor de *threshold* (equação 2.20) mais alto é escolhido. A mesmo tempo que se reduz ruídos e sombras, parte da silhueta do objeto é danificada, assim a operação de dilatação tem por objetivo compensar este erro.

A dilatação é definida por:

$$I_{dil} = I_{binaria} \oplus B = \bigcup_{x \in I_{binaria}} B_x, \tag{6.1}$$

onde B_x denota o elemento B transladado por x, e I_{final} a imagem binária dos objetos. Neste trabalho, o elemento estruturante trata-se de uma cruz de tamanho 1:

$$B = \begin{matrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{matrix}. \tag{6.2}$$

A imagem resultante desta operação I_{dil} é rotulada de 1 ao número total de objetos e então as características de área e cor, como mostrado na Seção 4.2, são extraídas para cada um destes novos objetos, que são posicionados numa lista de acordo com seu rótulo.

A posição de cada objeto é definida por seu centróide e calculada por:

$$C_i = \frac{x_1 + x_2 + ... + x_k}{k}, \tag{6.3}$$

onde x_n representa as coordenada do n-ésimo pixel que compõe o objeto i.

Cada alvo identificado e rastreado tem seus atributos de área, cor, posição, velocidade instantânea, classificação (contendo a classe e a posição do objeto a qual é relacionado - veja no Capítulo 5) e tempo de permanência armazenados numa lista ordenada. Assim, para cada novo objeto identificado, seus atributos são identificados e estocados no final desta lista.

6.3 Saídas do Sistema de Rastreamento

O sistema de rastreamento proposto apresenta três resultados como saída para o usuário: a imagem dos objetos, sendo esta a imagem de *foreground* após as operações de preenchimento, dilatação e rotulação, como descritas na Seção 6.2, a imagem dos alvos no quadro avaliado e um tabela contendo

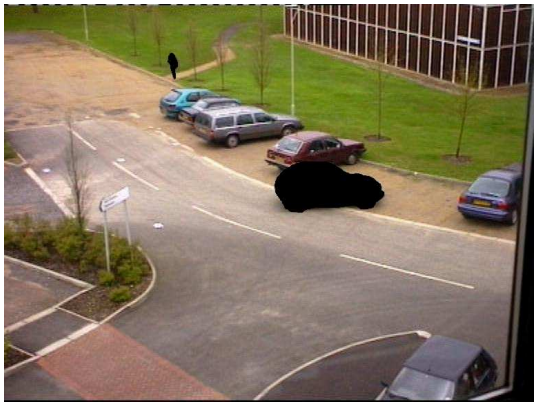
cada objeto e suas características.

A imagem dos alvos refere-se à marcação dos objetos de interesse do rastreamento, através de seu *Bounding Box*, no quadro processando. Considera-se como objeto de interesse os objetos pertencentes às classes Permanente, Zona de Oclusão e Separado. O *Bounding Box* dos objetos pertencentes às demais classes não são mostrados nesta imagem pois entende-se que são ruídos ou, no caso das classes Partes e Oclusão, o objeto é um fragmento de um objeto maior, representado pelo objeto relacionado, pertencente à classe Separado, ou não se tem informação exata de sua posição, respectivamente.

A tabela apresentada ao usuário contem, para cada objeto num quadro, a posição do seu centróide, classe a qual pertence, objeto a qual está relacionado, atributos de área e cor, tempo de permanência, isto é, a quantos quadros o objeto é rastreado e o valor de seu rótulo. Estas informações são dadas para todos os objetos identificados naquele quadro, mesmo os que não são mostrados na imagem dos alvos.

(a) Imagem da média

(b) Imagem de entrada



(c) Remoção dos objetos classificados como Permanente na imagem de entrada

(d) Área dos objetos removidos na imagem da média

(e) Composição final

Fig. 6.4: Exemplo de composição de imagem para atualização do autoespaço.

Capítulo 7

Avaliação do Sistema de Rastreamento de Objetos Proposto

O objetivo deste capítulo é apresentar uma avaliação do sistema de rastreamento proposto, buscando identificar se o sistema é capaz de rastrear os diferentes objetos que surgem nos vídeos, lidando com suas interações como oclusões, aparecimentos, desaparecimentos e separações, consequentemente classificando-os corretamente. Desta forma, a avaliação foi realizada através da saída do sistema de gerenciamento de objetos, verificando, em cada quadro dos vídeos testados, se os objetos identificados pelo sistema de subtração de fundo foram corretamente classificados.

Para esta avaliação, foram selecionados quatro vídeos das bases PETS e CAVIAR, mostrados na Figura 7.1, de nível de dificuldade variado:

Vídeo 1 - *Split* Vídeo da base de dados CAVIAR, trata-se de duas pessoas que entram juntas na cena e se separam, sendo esta a dificuldade do vídeo, além da baixa qualidade.

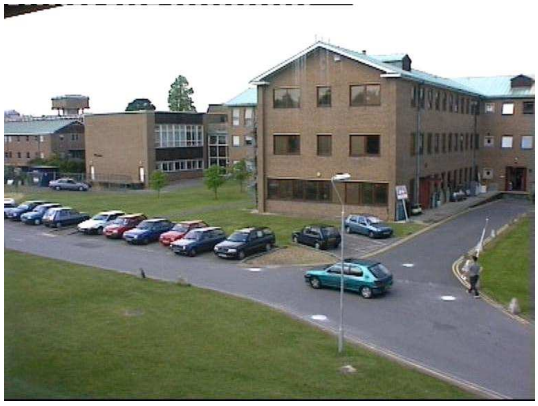
Vídeo 2 - *Meet Crowd* Também da base de dados CAVIAR, e no mesmo cenário apresentado no vídeo 1, este vídeo apresenta um grupos de pessoas que entram juntas na cena e saem juntas. A dificuldade do vídeo está em rastrear o grupo, além de lidar com a baixa qualidade do vídeo.

Vídeo 3 - PETS 2001 - DATASET 1 Pertencente à base de dados PETS do ano de 2001, trata-se de um vídeo de um ambiente externo e, portanto, sujeito a variações de iluminação, além de conter árvores que balançam. Os alvos passam por diversas classes durante o rastreamento, especialmente por ocorrências de oclusões.

Vídeo 4 - PETS 2001 - DATASET 2 Da mesma base de dados e ambiente do vídeo 3, este é o vídeo mais ruidoso da avaliação. Alguns dos alvos possuem cor próxima à cor do *background*, dificultando sua identificação.

(a) *Split*

(b) *Meet Crowd*



(c) PETS 2001 - DATASET 1

(d) PETS 2001 - DATASET 2

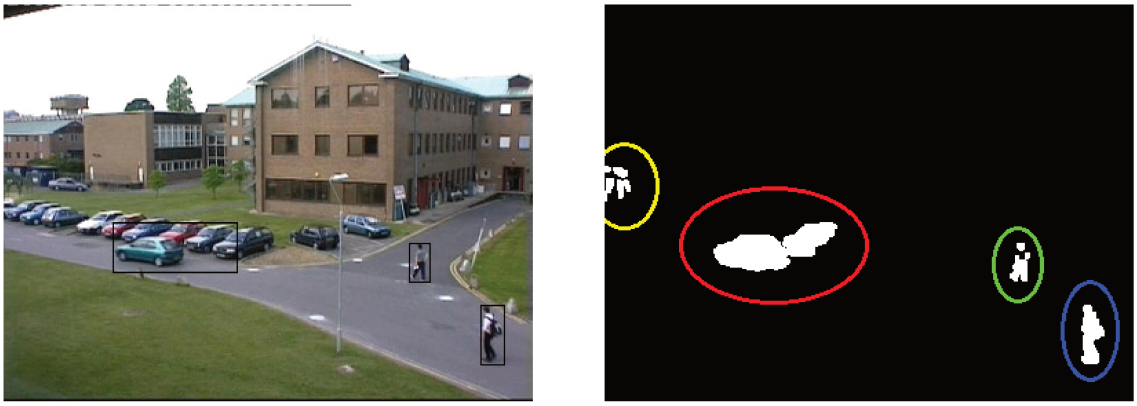
Fig. 7.1: Vídeos utilizados para avaliação do sistema de rastreamento proposto.

O nível de dificuldade está relacionado à quantidade de interações entre os alvos, oclusões e qualidade do vídeo. A avaliação foi iniciada em vídeos mais simples e curtos, avançando para cenas mais complexas, onde a detecção do *foreground* não é tão eficiente e a quantidade de alvos é maior.

O tamanho dos quadros dos vídeos foi fixado em 288 linhas por 384 colunas, sendo este o tamanho dos quadros dos vídeos da base de dados CAVIAR. Os videos da base de dados PETS, de tamanho original 576×768, foram redimensionados para facilitar o armazenamento das imagens no treinamento do modelo de subtração de fundo e atualização do mesmo.

Para avaliar o sistema de gerenciamento de objetos, os objetos de cada quadro dos vídeos foram manualmente classificados e esta classificação foi comparada à saída do sistema. A Figura 7.2 mostra um quadro do vídeos PETS - DATASET 1 e o resultado de sua subtração de fundo. O vídeo em questão apresenta um árvore no canto esquerdo que balança, gerando no resultado da subtração de fundo alguns objetos, destacado em amarelo na Figura 7.2(b). Estes objetos devem ser classificados como Fundo. Também neste quadro, dois carros aparecem sobrepostos, como destacado em vermelho na Figura 7.2(b). Neste caso, o sistema de gerenciamento de objetos deve apresentar dois objetos

classificados como Ocluso e um objeto classificado como Indefinido. Além disso, existe uma falha na identificação do *foreground*, fazendo com que uma das pessoas no vídeo fosse identificada como dois objetos, como destacado em verde na Figura 7.2(b). Estes dois objetos devem pertencer à classe Parte e um objeto pertencente à classe Separado deve constar na saída do sistema de gerenciamento de objetos. Por fim, um objeto deve ser classificado como Permanente, destacado em azul na Figura 7.2(b).



(a) Saída do sistema de rastreamento (b) Resultado da subtração de fundo

Fig. 7.2: Contagem e classificação dos objetos.

Pode-se observar que todos objetos foram identificados corretamente na saída do sistema de rastreamento, como mostra a Figura 7.2(a). Mas além disso, o sistema de gerenciamento de objetos também deve ser consultado, garantido que a classificação desejada foi satisfeita.

Os resultados serão apresentados em forma de tabela, que mostram a quantidade de objetos em cada classe e sua classificação desejada. Nesta tabela, as classes Fictício e Fora de cena não foram consideradas, pois objetos pertencentes à classe Fictício estão sempre relacionados à classe Fundo, repetindo sua contagem e, objetos pertencentes à classe Fora de cena são excluídos do sistema, não influenciando as demais classificações.

7.1 Vídeo *Split*

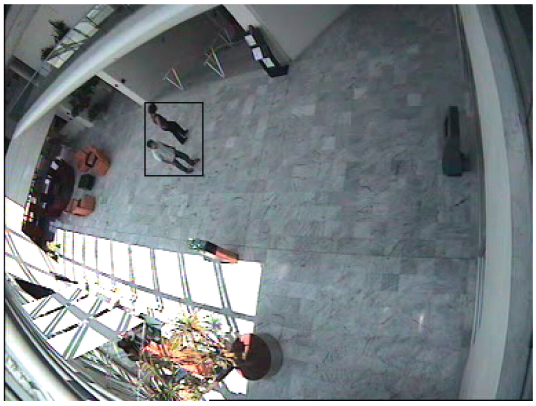
O vídeo *Split*, constituído de 414 frames, mostra duas pessoas entram juntas no saguão de um shopping e depois se separam, cada uma seguindo numa direção diferente, daí a dificuldade deste vídeo: lidar com a separação de um objeto inicial (composto pelas duas pessoas) em duas partes que devem formar dois objetos pertencentes à classe Permanente (cada pessoa seguindo numa direção).

A principal dificuldade deste vídeo é detectar a separação de dois objetos (pessoas que se separam), além de lidar com *background* ruidoso e com regiões com iluminação bastante diferenciada, o

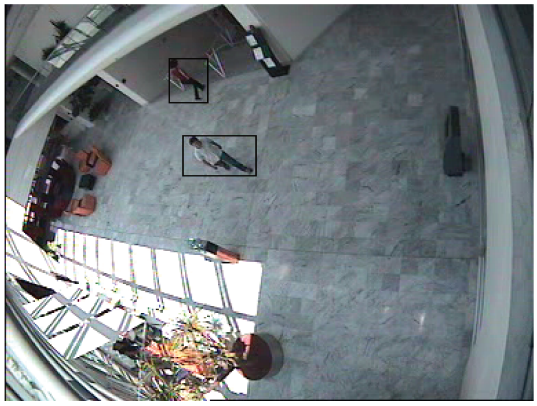
que afeta o modelo de subtração de fundo *Eigenbackground*.

O problema da subtração de fundo foi resolvido através da utilização de dois autoespaços para modelar o *background*, da forma que é mostrada na Seção 6.1, onde é mostrado um exemplo com um vídeo do mesmo ambiente. Assim, embora ainda existam ruídos e algumas falhas na detecção do *foreground*, estas não comprometeram de forma significativa o resultado obtido e, através do sistema de gerenciamento de objetos, os ruídos permaneceram, desconsiderando raras exceções, nas classes Fundo ou Iniciante, não transitando para a classe Permanente.

A classificação dos objetos após a separação foi de acordo com o esperado: um objeto da classe Permanente migrou para a classe Separado e dois novos objetos surgiram, classificados como Partes. Com o tempo, os objetos classificados como Partes se distanciam e migram para a classe Permanente e o objeto pertencente à classe Separado é excluído. A Figura 7.3 mostra a saída do sistema de rastreamento para esta transição.



(a) Saída do sistema de rastreamento, quadro 259



(b) Saída do sistema de rastreamento, quadro 293

Fig. 7.3: *Split* - Resultados.

A Tabela 7.1 mostra, na diagonal principal, a quantidade de objetos corretamente classificados pelo sistema de de gerenciamento de objetos, ao todo foram 830 objetos corretamente classificados de 895 objetos identificados.

Os principais erros encontrados foram objetos que deveriam pertencer à classe Fundo, sendo estes ruídos do vídeo, e foram classificados como Permanente e o objeto que deveria ser identificado como Separado e apenas uma parte sua foi classificada como Permanente, assim, apenas uma parte do alvo foi mostrada na saída do sistema de gerenciamento de objetos, a outra parte foi classificada como Fundo. A Figura 7.4 mostra a saída do sistema de rastreamento para estes erros: a Figura 7.4(a) um ruído, destacado em vermelho, é classificado como Permanente e aparece na saída do sistema de rastreamento, já na Figura 7.4(b), apenas a parte inferior de uma das pessoas rastreadas pode ser visualizada na saída do sistema de rastreamento.

		Classificação Obtida								
		Iniciante	Fundo	Permanente	Oclusо	Indefinido	Parte	Separado	Indisponível	Temporário
Classificação Desejada	Iniciante	102								
	Fundo		1618	21	4					
	Permanente			340						
	Oclusо				4					
	Indefinido					2				
	Parte		16	16			248			
	Separado							120		
	Indisponível								21	
	Temporário									1

Tab. 7.1: Erros referentes ao vídeo *Split*.

(a) Saída do sistema de rastreamento, quadro 358 (b) Saída do sistema de rastreamento, quadro 376

Fig. 7.4: *Split* - Falha do sistema de rastreamento.

7.2 Vídeo *Meet Crowd*

O vídeo *Meet Crowd*, composto de 498 quadros, apresenta um grupo de pessoas que entram caminhando juntas num saguão de um shopping, o atravessam e saem, completando o percurso do lado oposto de onde entraram. As pessoas entram na cena sobrepostas do ponto de vista da câmera, sendo identificadas como um único objeto como destacado em vermelho na Figura 7.5. Em alguns momentos da caminhada, algumas pessoas se distanciam do grupo mas não se separam o suficiente para serem identificadas como novos objetos.

O vídeo se passa no mesmo ambiente do vídeo *Split*, sofrendo as mesmas interferências de iluminação causadas pela existência de uma janela que permite a entrada de iluminação externa. Desta

(a) Saída do sistema de rastreamento, quadro 136

(b) Resultado da subtração de fundo

Fig. 7.5: *Meet Crowd* - Identificação do grupo como um único objeto.

forma, assim como o vídeo *Split*, o modelo de *background* é formado através da divisão dos quadros do vídeo em duas partes (as mesmas utilizadas no vídeo *Split*).

A Tabela 7.2 mostra a quantidade de objetos identificados pelo sistema de rastreamento proposto, totalizando 3269 objetos. Ao todo, 3267 objetos foram corretamente classificados, como mostra a diagonal principal da tabela. Os dois objetos classificados incorretamente trata-se de um ruído classificado como Permanente e, mais tarde, classificado como Indisponível.

		Classificação Obtida								
		Iniciante	Fundo	Permanente	Ocluso	Indefinido	Parte	Separado	Indisponível	Temporário
Classificação Desejada	Iniciante	930		1					1	
	Fundo		1261							
	Permanente			49						
	Ocluso									
	Indefinido									
	Parte						775			
	Separado							252		
	Indisponível									
	Temporário									

Tab. 7.2: Erros referentes ao vídeo *Meet Crowd*.

A Figura mostra 7.6 mostra o resultado do sistema de rastreamento em dois momentos: no primeiro quadro mostrado 7.6(a), os elementos do grupo estão sobrepostos, sendo classificados como um único objeto pertencente à classe Permanente. A Figura 7.6(b), mostra uma das pessoas do grupo

mais afastada, no entanto este afastamento não é suficiente para que esta pessoa seja considerada um objeto Permanente. Nesta figura, o grupo inicial é classificado como Separado e a pessoa é classificada como Parte. O novo grupo que surgiu, formado pelo restante das pessoas, também é classificado como Parte.

(a) Saída do sistema de rastreamento, quadro 207 (b) Saída do sistema de rastreamento, quadro 277

Fig. 7.6: *Meet Crowd* - Resultados.

7.3 Vídeo PETS 2001 - DATASET 1

O vídeo PETS 2001 - DATASET 1, constituído de 2945 quadros, tem como cenário o estacionamento de uma universidade, onde carros, pessoas e bicicletas passam, sofrendo oclusões e separações. Além disso, o vídeo se passa num ambiente externo, onde variações de iluminação, sombras e efeitos da ação do vento, como árvores balançando, são comuns.

Dentre as interações dos objetos, destaca-se a saída de um carro de uma das vagas do estacionamento, ilustrada pela Figura 7.7, visto que o modelo de *background* fora treinado com o carro estacionado. O objeto, inicialmente classificado como Fundo, transitou com sucesso para a classe Permanente e, o espaço anteriormente encoberto por este foi classificado como Temporário e logo absorvido pelo modelo de *background*.

Foram identificados e classificados 8316 objetos, como mostra a Tabela 7.3, destes, 8098 foram corretamente classificados. Dos erros encontrados, 107 objetos foram classificados como Permanente, quando deveriam ser classificados como Iniciante. Estes objetos representam, principalmente, uma árvore que balança, como mostra a Figura 7.9(a).

Outros 11 objetos foram classificados como indisponível, quando deveriam ser classificados como Permanente. Estes objetos, como mostrados na Figura 7.8, representam uma pessoa que estava em oclusão com um carro e, após o final da oclusão, não retorna imediatamente à classe Permanente,

		Classificação Obtida								
		Iniciante	Fundo	Permanente	Ocluso	Indefinido	Parte	Separado	Indisponível	Temporário
Classificação Desejada	Iniciante	191		107						
	Fundo		2001							
	Permanente			3219					11	
	Ocluso				811					
	Indefinido					407				
	Parte		59	41			744			
	Separado							347		
	Indisponível								304	
	Temporário									74

Tab. 7.3: Erros referentes ao vídeo PETS 2001 - DATASET 1.

permanecendo durante 11 quadros na classe Indisponível.

Outra falha encontrada neste vídeo são de objetos que deveriam ser classificados como Parte e são classificados como Permanente (41 objetos) ou Fundo (59 objetos). A Figura 7.9(b) mostra um exemplo deste tipo de falha: a pessoa que está sendo rastreada se divide em duas partes, porém apenas a parte de baixo é classificada como Permanente e rastreada, a outra é classificada como Fundo.

7.4 Vídeo PETS 2001 - DATASET 2

O vídeo PETS 2001 - DATASET 2, composto de 2822 quadros, se passa no mesmo ambiente do vídeo PETS 2001 - DATASET 1, apresentando as mesmas dificuldades deste vídeo adicionadas à má qualidade do vídeo PETS 2001 - DATASET 2 devido ao tipo de compressão do vídeo.

14477 objetos foram identificados e classificados e, destes, 14020 objetos foram classificados corretamente, como mostra a Tabela 7.4. A grande quantidade de objetos classificados como Fundo e Separado em relação ao vídeo PETS 2001 - DATASET 1, refletem a má qualidade do vídeo, o que influencia na identificação do *foreground*, ora identificando ruídos, ora deixando de identificar toda a área do alvo rastreado, partindo-o em duas ou mais partes.

A Figura 7.10 mostra um exemplo de interação entre objetos identificada com sucesso pelo sistema de gerenciamento de objeto. Uma bicicleta entra em oclusão duas vezes seguidas com pedestres, transitando da classe Permanente para a classe Ocluso, gerando também um objeto classificado como Indefinido, e depois novamente para a classe Permanente. O mesmo processo de transição de classes ocorre com cada pedestre.

(a) Saída do sistema de rastreamento, quadro 1208

(b) Saída do sistema de rastreamento, quadro 1320

(c) Saída do sistema de rastreamento, quadro 1404

Fig. 7.7: PETS 2001 - Resultados.

(a) Saída do sistema de rastreamento, quadro 1702

(b) Saída do sistema de rastreamento, quadro 1737

(c) Saída do sistema de rastreamento, quadro 1754

Fig. 7.8: PETS 2001 - DATASET 1: Falha na classificação de objeto que sai da oclusão.

		Classificação Obtida								
		Iniciante	Fundo	Permanente	Ocluso	Indefinido	Parte	Separado	Indisponível	Temporário
Classificação Desejada	Iniciante	723								
	Fundo		7865	6						
	Permanente		324	3742					93	
	Ocluso				295					
	Indefinido					147				
	Parte		17	17			834			
	Separado							406		
	Indisponível									
	Temporário									8

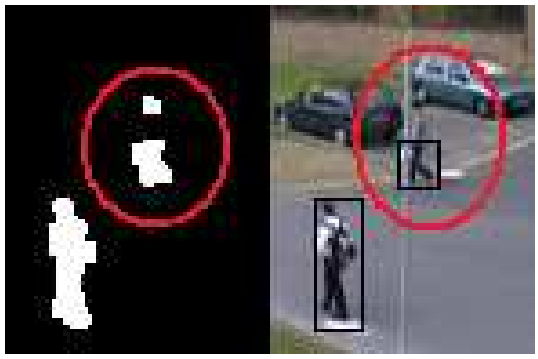
Tab. 7.4: Erros referentes ao vídeo PETS 2001 - DATASET 2.

Ao contrário dos vídeos anteriores, poucos ruídos foram classificados como Permanente, isto é, objetos pertencentes às classes Iniciante ou Fundo e classificados como Permanente. Porém, outros tipos de erros foram mais evidentes, como objetos que deveriam pertencer à classe Permanente sendo classificados como Indisponível ou Fundo.

No primeiro caso, mostrado pela Figura 7.11 onde um objeto pertencente à classe Permanente é classificado como Fundo ocorre devido a uma falha na identificação do *foreground*. A área pertencente ao objeto deixa de ser inteiramente identificada, fazendo com que a área pertencente ao alvo reduza de tamanho e seu atributo de cor seja alterado. Quando o alvo retorna ao seu tamanho inicial, ele deixa de ser associado ao seu correspondente no quadro anterior pois seus atributos de área e cor estão alterados, assim o mesmo é classificado como Indisponível.

Entretanto, o erro mais grave ocorre no segundo caso, mostrado na Figura 7.12 onde um objeto que deveria pertencer à classe Permanente é classificado como Fundo. Devido a falhas na identificação do *foreground*, o objeto que deveria surgir de uma separação, é classificado como Fundo e segue nesta classificação por 324 quadros. Como mostra a Figura 7.12(d), devido a falhas na identificação do *foreground*, o alvo se parte ao deixar o grupo, desta forma, uma de suas partes (a maior) é identificada como Fundo, como mostrado no resultado do sistema de rastreamento, na Figura 7.12(c). A partir daí, o objeto é sempre classificado como Fundo, como mostra a Figura 7.12(e).

(a) Saída do sistema de rastreamento, quadro 1469 - árvore balançando



(b) Detalhe do quadro 1518. À direita o resultado do sistema de rastreamento proposto e à esquerda a identificação do *foreground*.

Fig. 7.9: PETS 2001 - DATASET 1: Falha na classificação de objetos.

(a) Saída do sistema de rastreamento, quadro 856

(b) Saída do sistema de rastreamento, quadro 873

(c) Saída do sistema de rastreamento, quadro 890

(d) Saída do sistema de rastreamento, quadro 912

(e) Saída do sistema de rastreamento, quadro 923

Fig. 7.10: PETS 2001 - DATASET 2: Objetos da classe Permanente que transitam para a classe Ocluso e voltam para a classe Permanente.

(a) Detalhe do quadro 1060

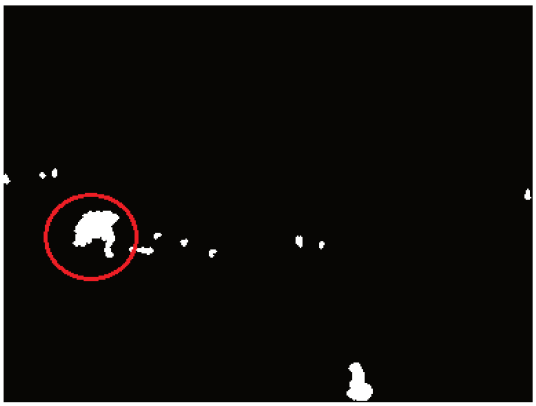
(b) Detalhe do quadro 1061

(c) Detalhe do quadro 1062

(d) Detalhe do quadro 1063

Fig. 7.11: PETS 2001 - DATASET 2: Falha na classificação de um alvo. À direita o resultado do sistema de rastreamento proposto e à esquerda o resultado da subtração de fundo.

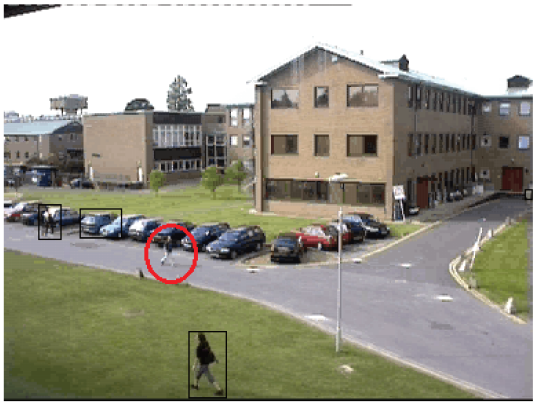
(a) Saída do sistema de rastreamento, quadro 2314



(b) Resultado da subtração de fundo para o quadro 2314

(c) Saída do sistema de rastreamento, quadro 2321

(d) Resultado da subtração de fundo para o quadro 2321



(e) Saída do sistema de rastreamento, quadro 2383

(f) Resultado da subtração de fundo para o quadro 2383

Fig. 7.12: PETS 2001 - DATASET 2: Falha na separação de dois alvos.

Capítulo 8

Conclusão

Este trabalho abordou o problema do rastreamento automático de múltiplos objetos em vídeos de câmeras estacionárias, com o objetivo principal de propor um sistema de rastreamento para vídeos de segurança baseado em sistemas já descritos na literatura. O sistema de rastreamento proposto é composto de três fases: identificação dos objetos a serem seguidos através do modelo de subtração de fundo *Eigenbackground*, associação destes objetos aos objetos do quadro anterior do vídeo e separação dos objetos em classes que definem seu estado.

Durante o processo de formulação do sistema proposto, diversos complicadores foram notados, tais como ruídos e má qualidade das imagens, interações dos alvos seguidos com o ambiente e entre si, além de variações na cor, tamanho e forma. Assim, o sistema foi modelado de forma a contornar tais dificuldades, utilizando e modificando alternativas sugeridas em outros trabalhos.

Durante o estudo, observou-se que a identificação do *foreground*, por fornecer dados como posição dos objetos na imagem, seus tamanhos e formas, é uma fase crucial para o sucesso ou fracasso do sistema de rastreamento. Desta forma, o modelo de subtração de fundo utilizado, *Eigenbackground*, foi testado e comparado ao tradicional modelo Mistura de Gaussianas (MoG). O estudo da sensibilidade de parâmetros mostrou, através das taxas de precisão e retorno, que tais modelos de subtração de fundo apresentaram resultados próximos quando tomado o melhor conjunto de parâmetros para cada vídeo, entretanto o modelo *Eigenbackground* destacou-se por ser mais tolerante a variações na iluminação além de apresentar melhores resultados na ocorrência de sombras.

O modelo *Eigenbackground* apresenta algumas limitações como a utilização de um único valor de *threshold* para todo o quadro processado, o que dificultou encontrar um valor que se adequasse à imagem toda, e falta de um sistema de atualização do modelo. Para resolver o problema do *threshold*, os quadros do vídeo foram particionados e, para cada parte, foi gerado um autoespaço para representar o modelo de *background* e estabelecido um *threshold* diferente. A partição dos quadros e adequação do *threshold* melhorou consideravelmente os resultados, especialmente em cenas nas quais haviam

variações bruscas de iluminação, como na presença de uma janela, corredor escuro, etc.

Para atualizar o modelo *Eigenbackground*, duas abordagens foram analisadas: a fusão de autoespaços e reinicialização de autoespaços. A fusão de autoespaços foi empregada para a comparação entre o modelo *Eigenbackground* ao MoG. Os quadros processados são armazenados e a cada conjunto de novos quadros, um novo autoespaço é calculado e fundido com o autoespaço anterior. Assim como a taxa de atualização do modelo MoG, esta taxa precisa ser cuidadosamente escolhida para que objetos não sejam incorporado rápido demais ao modelo de *background* ou para que a atualização demore demais e mudanças na cena sejam identificadas como *foreground*.

A atualização do modelo de *background* adotada pelo sistema de rastreamento proposto adota como modelo de *background* o autoespaço recalculado a cada conjunto de novos quadros, sem a realização da fusão. Entretanto, para esta abordagem, o sistema de gerenciamento de objetos têm papel fundamental para este tipo de atualização, definindo quais objetos serão ou não incorporados ao modelo de *background*.

A fase de associação de objetos quadro a quadro foi realizada com o auxílio do filtro de Kalman aplicado sobre o centróide dos objetos, além de métricas de cor e área. Entretanto, nos casos de separação, oclusão e desaparecimento de objetos, as métricas utilizadas falharam pois as informações de área, cor e centróide dos objetos não podem ser calculadas. Para manter o rastreamento nestas condições, um sistema de gerenciamento de objetos foi criado com o objetivo de definir o estado de cada objeto rastreado, definindo, por exemplo, se o objeto fundiu-se com outro objeto, caracterizando oclusão, se partiu em dois ou mais objetos, caracterizando separação, ou não encontrou um correspondente no quadro seguinte, caracterizando desaparecimento.

Este sistema foi projetado baseado num sistema de gerenciamento já existente [17], porém foi modelado de acordo com as dificuldades encontradas nos vídeos utilizados nos experimentos. Entre as modificações do sistema de gerenciamento original, pode-se citar a inclusão da classe Separado, que foi de grande valia para alguns casos de falha identificação do *foreground* ou em casos em que dois ou mais objetos entram juntos na cena.

A avaliação do sistema de gerenciamento de objetos mostrou que o mesmo obteve sucesso na identificação dos estados dos objetos rastreados e conseguiu recuperar o rastreamento após oclusões, separações e desaparecimentos. As falhas encontradas na classificação dos objetos foram geradas por falhas na identificação do *foreground*, principalmente a não identificação da área total do objeto rastreado.

Embora projetado visando a aplicação em vídeos de segurança, o sistema de rastreamento de objetos proposto pode ser adaptado a outras aplicações como por exemplo rastreamento de animais em experimentos em laboratório, movimentação de atletas em jogos de futebol ou basquete, monitoramento de tráfego, entre outras aplicações. Sua aplicação, no entanto, é limitada por utilizar uma única

câmera estacionária, o que torna o rastreamento ineficiente quando um grande número de objetos está presente no vídeo, pois nestes casos o número de oclusões é normalmente grande, impedindo que os objetos sejam seguidos individualmente.

Diversas extensões visando a melhoria deste trabalho podem ser consideradas, são elas:

- melhoria do método de identificação de *foreground* através do estudo de outras técnicas ou aperfeiçoamento do *Eigenbackground*;
- implementação do sistema para operar em tempo real;
- análise de sequências mais longas de vídeo de modo a verificar sua robustez às variações do ambiente;
- utilização de mais câmeras para obter informações multi-vistas da cena, reduzindo oclusões por sobreposição;
- extração de semântica nos vídeos, isto é, identificar, por exemplo, se o objeto seguido é uma pessoa ou um carro e identificar suas ações. Se o objeto rastreado for uma pessoa pode-se, por exemplo, identificar se está correndo ou andando, em qual direção e se encontra com outras pessoas no caminho.

Referências Bibliográficas

[1] Extended Reality Software. Extended reality. `http://www.extendedreality.com/webcam_games_info.html`. [Online; accessed 21-Outubro-2009].

[2] Mobinex Inc. Fix 8. `http://www.fix8.com/`. [Online; accessed 21-Outubro-2009].

[3] A. Agarwal and B. Triggs. Tracking articulated motion using a mixture of autoregressive models. In *European Conference on Computer Vision*, pages 54–65, 2004.

[4] G. Liu, X. L. Tang, H. D. Cheng, J. H. Huang, and J. F. Liu. A novel approach for tracking high speed skaters in sports using a panning camera. *Pattern Recognition*, 42(11):2922–2935, 2009.

[5] D. Meyer and J. Denzler. Model based extraction of articulated objects in image sequences for gait analysis. *International Conference on Image Processing*, 3:78, 1997.

[6] P. J. Figueroa, N. J. Leite, and R. M. L. Barros. A flexible software for tracking of markers used in human motion analysis. *Computer Methods and Programs in Biomedicine*, 72(2):155–165, 2003.

[7] P. Smith, M. Shah, and N. D. V. Lobo. Determining driver visual attention with one camera. *IEEE Transactions on Intelligent Transportation Systems*, 4:2003, 2003.

[8] P.E. An and C.J. Harris. An intelligent driver warning system for vehicle collision-avoidance. *IEEE Transactions System, Man and Cybernetics*, 26(2):254–261, March 1996.

[9] F. Heimes and H.H. Nagel. Towards active machine-vision-based driver assistance for urban areas. *International Journal of Computer Vision*, 50(1):5–34, October 2002.

[10] H.G. Jung, Y.H. Cho, P.J. Yoon, and J.H. Kim. Scanning laser radar-based target position designation for parking aid system. *IEEE Transactions on Intelligent Transportation Systems*, 9(3):406–424, September 2008.

[11] J. D. Smith and T. C. N. Graham. Use of eye movements for video game control. In *International Conference on Advances in Computer Entertainment Technology*, page 20, New York, NY, USA, 2006. ACM.

[12] G. R. Bradski. Computer vision face tracking for use in a perceptual user interface. *Intel Technology Journal*, (Q2), 1998.

[13] S. Wang, X. Xiong, Y. Xu, C. Wang, W. Zhang, X. Daiv, and D. Zhang. Face-tracking as an augmented input in video games: enhancing presence, role-playing and control. In *SIGCHI conference on Human Factors in computing systems*, pages 1097–1106, New York, NY, USA, 2006. ACM.

[14] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2, pages –252 Vol. 2, 1999.

[15] N.M. Oliver, B. Rosario, and A.P. Pentland. A bayesian computer vision system for modeling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):831–843, Aug 2000.

[16] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME, Journal of Basic Engineering*, (82 (Series D)):35–45, 1960.

[17] B. Lei and L.-Q. Xu. Real-time outdoor video surveillance with robust foreground extraction and object tracking via multi-state transition management. *Pattern Recognition Letters*, 27(15):1816 – 1825, 2006. Vision for Crime Detection and Prevention.

[18] R. Jain and H. H. Nagel. On the analysis of accumulative difference pictures from image sequences of real world scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(2):206–214, 1979.

[19] C.R. Wren, A. Azarbayejani, T. Darrell, and A.P. Pentland. Pfinder: real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, Jul 1997.

[20] A. Elgammal, D. Harwood, and L. Davis. Non-parametric model for background subtraction. In *6th European Conference on Computer Vision-Part II*, pages 751–767, London, UK, 2000. Springer-Verlag.

[21] I. Haritaoglu, D. Harwood, and L.S. Davis. W4: Real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):809–830, August 2000.

[22] E. Herrero-Jaraba, C. Orrite-Uruñuela, and J. Senar. Detected motion classification with a double-background and a neighborhood-based difference. *Pattern Recognition Letters*, 24:2079–2092, 2003.

[23] J. B. Kim, H. S. Park, M. H. Park, and H. J. Kim. A real-time region-based motion segmentation using adaptive thresholding and k-means clustering. In *Australian Joint Conference on Artificial Intelligence*, pages 213–224, London, UK, 2001. Springer-Verlag.

[24] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564–577, May 2003.

[25] M. Swain and D. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[26] A. Bhattacharyya. On a measure of divergence between two statistical populations defined by their probability distributions. *Bulletin of the Calcutta Mathematical Society*, 35:99 – 109, 1943.

[27] B.H. Han and R. Jain. Real-time subspace-based background modeling using multi-channel data. In *Advances in Visual Computing*, pages II: 162–172, 2007.

[28] H. Lu, S. Fei, J. Zheng, and T. Zhang. An occlusion tolerent method for multi-object tracking. In *World Congress on Intelligent Control and Automation*, pages 5105–5110, June 2008.

[29] C. Stauffer and W.E.L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–757, Aug 2000.

[30] T. Russ, C. Boehnen, and T. Peters. 3d face recognition using 3d alignment for pca. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1391–1398, Washington, DC, USA, 2006. IEEE Computer Society.

[31] R. Billon, A. Nédélec, and J. Tisseau. Gesture recognition in flow based on pca and using multiagent system. In *ACM Symposium on Virtual Reality Software and Technology*, pages 239–240, New York, NY, USA, 2008. ACM.

[32] A.F. Otoom, H. Gunes, and M. Piccardi. Feature extraction techniques for abandoned object classification in video surveillance. In *IEEE International Conference on Image Processing*, pages 1368–1371, 2008.

[33] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons Inc, 1973.

[34] P. Shi Z. Xu and I. Y. H. Gu. *An Eigenbackground Subtraction Method Using Recursive Error Compensation*. 2006.

[35] Y. Chien and F. King-Sun. On the generalized karhunen-loéve expansion. *IEEE Transactions on Information Theory*, 13(3):518–520, Jul 1967.

[36] P. Hall, D. Marshall, and R. Martin. Merging and splitting eigenspace models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(9):1042–1049, 2000.

[37] G. H. Golub and C. F. Van Loan. *Matrix Computations (Johns Hopkins Studies in Mathematical Sciences)*. The Johns Hopkins University Press, 1983.

[38] James Ferryman. Pets dataset. `http://www.cvg.rdg.ac.uk/VS/`, 2001.

[39] CAVIAR. Ec funded caviar project/ist 2001 37540. `http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/`, 2001.

[40] D. Koller, J. Weber, and J. Malik. Robust multiple car tracking with occlusion reasoning. In *European Conference on Computer Vision*, pages 189–196, 1994.

[41] M. Kilger. A shadow handler in a video-based real-time traffic monitoring system. In *Workshop on Applications of Computer Vision*, pages 11–18, 1992.

[42] S. S. Intille, J. W. Davis, and A. F. Bobick. Real-time closed-world tracking. In *Conference on Computer Vision and Pattern Recognition*, page 697, Washington, DC, USA, 1997. IEEE Computer Society.

[43] K. Rangarajan and M. Shah. Establishing motion correspondence. *Graphical Models and Image Processing*, 54(1):56–73, 1991.

[44] I. Haritaoglu, D. Harwood, and L. S. Davis. Hydra: Multiple people detection and tracking using silhouettes. *IEEE Workshop on Visual Surveillance*, 1, 1999.

[45] F. Cupillard, F. Brémond, M. Thonnat, I. S. Antipolis, and O. Group. Tracking groups of people for video surveillance. In *Video-Based Surveillance Systems: Computer Vision and Distributed Processing*, pages 89 – 101. Kluwer Academic Publishers, 2001.

[46] Z. Li, Q.L. Tang, and N. Sang. Improved mean shift algorithm for occlusion pedestrian tracking. *Electronics Letters*, 44(10):622–623, 2008.

[47] J. Gallego, M. Pardas, and J.L. Landabaso. Segmentation and tracking of static and moving objects in video surveillance scenarios. In *15th IEEE International Conference on Image Processing*, pages 2716–2719, 2008.

[48] T. Zhao and R. Nevatia. Tracking multiple humans in complex situations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:1208–1221, 2004.

[49] B. Leibe, K. Schindler, and L.J. Van Gool. Coupled detection and trajectory estimation for multi-object tracking. In *IEEE International Conference on Computer Vision*, pages 1–8. IEEE Computer Society, 2007.

[50] J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

[51] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1:878–885, 2005.

[52] H. G. Barrow, J. M. Tenenbaum, R. C. Bolles, and H. C. Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. Technical Report 153, AI Center, SRI International, 333 Ravenswood Ave, Menlo Park, CA 94025, 1977.

[53] S. Kong, M.K. Bhuyan, C. Sanderson, and B.C. Lovell. Tracking of persons for video surveillance of unattended environments. In *19th International Conference on Pattern Recognition*, pages 1–4, Dec. 2008.

[54] S. Kong, C. Sanderson, and B.C. Lovell. Classifying and tracking multiple persons for proactive surveillance of mass transport systems. In *IEEE Conference on Advanced Video and Signal Based Surveillance*. IEEE Computer Society.

[55] G. Welch and G. Bishop. An introduction to the kalman filter. Technical report, Chapel Hill, NC, USA, 1995.

[56] R. Pinho, J. Tavares, and M. Correia. Introdução à análise de movimento por visão computacional. *Relatório Interno, Faculdade de Engenharia, Universidade do Porto*, 2004.

[57] Z. Xu, I. Y.-H. Gu, and P. Shi. Recursive error-compensated dynamic eigenbackground learning and adaptive background subtraction in video. *Optical Engineering*, 47(5), 2008.

Apêndice A

Diagonalização Matriz de Covariância

Seja X uma matriz de características da forma

$$\begin{bmatrix} x_{11} & \cdots & x_{1m} \\ x_{21} & \cdots & x_{2m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nm} \end{bmatrix}, \tag{A.1}$$

onde x_{kl} é l-ésima observação da k-ésima característica, sua matriz de covariância é dada por

$$C_X = XX^T. \tag{A.2}$$

Pode-se fazer algumas afirmações sobre a matriz C_X:

- Os termos da diagonal de C_X são as variâncias das características
- Os termos fora da diagonal representa a covariância entre dois tipos de características
- C_X é uma matriz de dimensão $n \times n$ simétrica uma vez que

$$(XX^T)^T = (X^T)^T X^T = XX^T, \tag{A.3}$$

  garantindo que C_X seja diagonalizável.

Para diagonalizar C_X define-se uma base ortonormal desconhecida P tal que $Y = PX$, onde $C_Y \equiv YY^T$ é uma matriz diagonal, desta forma temos:

$$\begin{aligned} C_Y &= YY^T \\ &= (PX)(PX)^T \\ &= PXX^TP^T \\ &= P(XX^T)P^T \\ C_Y &= PC_XP^T, \end{aligned} \tag{A.4}$$

Onde P são autovetores ortonormais de C_X, segundo o seguinte teorema:

Theorem A.0.1 (Uma matriz simétrica é diagonalizada pela matriz de seus autovetores ortogonais) *Seja A uma matriz quadrada n x n simétrica com autovetores associados $\{e_1, e_2, \cdots, e_n\}$. Seja $E = [e_1e_2 \cdots e_n]$, onde a i-ésima coluna de E é o autovetor e_i. Este teorema garante que existe uma matriz diagonal D tal que $A = EDE^T$.*

Proof Seja A uma matriz qualquer e seja $E = [e_1e_2 \cdots e_n]$ a matriz de seus autovetores dispostos em colunas. Seja D uma matriz diagonal com seu i-ésimo autovalor na ii-ésima posição.

Sejam $AE = [Ae_1Ae_2 \cdots Ae_n]$ e $ED = [\lambda_1e_1\lambda_1e_2 \cdots \lambda_ne_n]$, se $AE = ED$, então $Ae_i = \lambda_ie_i$ para todo i é a definição da equação de autovalor, além disso pode-se escrever $A = EDE^{-1}$.

Para uma matriz simétrica, seja λ_1 e λ_2 autovalores distintos dos respectivos autovetores e_1 e e_2. Então,

$$\begin{aligned} \lambda_1e_1e_2 &= (\lambda_1e_1)^Te_2 \\ &= (Ae_1)^Te_2 \\ &= e_1^TA^Te_2 \\ &= e_1^TAe_2 \\ &= e_1^T(\lambda_2e_2) \\ \lambda_1e_1e_2 &= \lambda_2e_1e_2. \end{aligned} \tag{A.5}$$

Por esta relação podemos calcular que $(\lambda_1 - \lambda_2)e_1e_2 = 0$. Uma vez que autovalores são únicos, então $e_1e_2 = 0$. Assim, os autovetores de uma matriz simétrica são ortogonais, o que significa que E é uma matriz ortogonal, então $E^T = E^{-1}$ e podemos reescrever o resultado final como $A = EDE^T$. ∎

O teorema A.0.1 em conjunto com a equação A.4 garantem que dada uma matriz de covariância C_X, esta pode ser diagonalizada por seus autovetores T resultando na matriz diagonal C_Y.

Apêndice B

Formando um novo autoespaço

Sejam dois autoespaços $\Omega = (\bar{x}, U, \Lambda, N)$ e $\Psi = (\bar{y}, V, \Delta, M)$.

A matriz de covariância da fusão destes dois espaços é definida por

$$E = \frac{N}{P}(C) - \frac{M}{P}(D) + \frac{NM}{P^2}(\bar{x} - \bar{y})(\bar{x} - \bar{y})^T, \tag{B.1}$$

e o autoespaço resultante é dado por

$$E = W \Pi W^T, \tag{B.2}$$

onde W é uma matriz de autovetores, Π uma matriz diagonal de autovalores, C e D as matrizes de covariância dos autoespaços Ω e Ψ, respectivamente.

W também pode ser obtido por uma rotação R de uma base ortonormal Υ:

$$W = \Upsilon R. \tag{B.3}$$

Seja v uma base ortonormal a Ω, Ψ e $\bar{x} - \bar{y}$, Υ é calculado por:

$$\Upsilon = [Uv]. \tag{B.4}$$

Substituindo a equação B.4 em B.3 e juntando com B.1 em B.2, obtemos:

$$\frac{N}{P}(C) - \frac{M}{P}D + \frac{NM}{P}(\bar{x} - \bar{y})(\bar{x} - \bar{y})^T = [Uv] R \Pi R [Uv]^T. \tag{B.5}$$

Multiplicando o lado esquerdo da equação por $[Uv]^T$, e o lado direito por $[Uv]$, obtemos:

$$[Uv]^T (\frac{N}{P}(C) - \frac{M}{P}D + \frac{NM}{P}(\bar{x} - \bar{y})(\bar{x} - \bar{y})^T)[Uv] = R \Pi R, \tag{B.6}$$

o que forma um autoespaço cujo autovetores constituem a matriz de rotação R.

Como não se conhece *a priori* as matrizes de covariância C e D, estas precisam ser eliminada, utilizando-se o fato que $C \approx U\Lambda U^T$ [1] e $U^T v = 0$, uma vez que U e v são ortonormais. Assim obtemos:

$$[Uv]^T C[Uv] \approx \begin{bmatrix} \Lambda & 0 \\ 0 & 0 \end{bmatrix}. \tag{B.7}$$

O segundo termo de B.6 também pode ser reduzido uma vez que $D \approx V\Delta V^T$ [1] e $G = U^T V$, obtendo:

$$[Uv]^T D[Uv] \approx \begin{bmatrix} G\Delta G^T & G\Delta\Gamma^T \\ \Gamma\Delta G^T & \Gamma\Delta\Gamma^T \end{bmatrix}, \tag{B.8}$$

onde $\Gamma = v^T V$.

O ultimo termo em B.6 pode ser escrito como:

$$\begin{bmatrix} gg^T & g\gamma^T \\ \gamma g^T & \gamma\gamma^T \end{bmatrix}, \tag{B.9}$$

com $g = U(\bar{x} - \bar{y})$ e $\gamma = v^T(\bar{x} - \bar{y})$. Assim o novo autoespaço pode ser aproximado para:

$$R\Pi R^T = \frac{N}{P}\begin{bmatrix} \Lambda & 0 \\ 0 & 0 \end{bmatrix} + \frac{M}{P}\begin{bmatrix} G\Delta G^T & G\Delta\Gamma^T \\ \Gamma\Delta G^T & \Gamma\Delta\Gamma^T \end{bmatrix} + \frac{NM}{P^2}\begin{bmatrix} gg^T & g\gamma^T \\ \gamma g^T & \gamma\gamma^T \end{bmatrix}. \tag{B.10}$$

[1] Estas relações são aproximadas pois o número de autovalores e autovetores que compõe o autoespaço Ω e Ψ são truncados uma vez que alguns autovalores e autovetores são descartados de acordo com critério discutido na seção 2.2.1