

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



FEUP

Análise da variação dinâmica da complexidade de um algoritmo de tracking

Jorge Rodrigues

Mestrado Integrado em Engenharia Eletrotécnica e Computadores

Orientador: Professor Luís Corte-Real (PhD)

Co-orientador: Pedro Carvalho (PhD)

24 de Outubro de 2012

Resumo

A visão computacional é uma área em rápida expansão, estando já presente em inúmeros aspectos da vida quotidiana. Os métodos de deteção e seguimento de pessoas em tempo real assumem, neste contexto, um papel de extrema importância.

Esta dissertação procura analisar a possibilidade de adaptação da complexidade do algoritmo de seguimento, de acordo com informações recolhidas relativamente à qualidade em cada fase do processamento.

A partir de um algoritmo de seguimento de pessoas, foi preparada uma plataforma de testes com o intuito de avaliar o desempenho do algoritmo e foram implementadas medidas de avaliação, sem recurso a informação de referência, da qualidade dos resultados nas principais componentes do algoritmo. As medidas são combinadas de forma a dotar o algoritmo de capacidade de decisão relativamente ao melhor caminho a seguir. A decisão pode ser tomada quer em termos de melhoria da qualidade dos resultados do seguimento ou diminuição do tempo de processamento.

A melhoria da qualidade dos resultados do seguimento passou pela obtenção de melhores resultados ao nível da segmentação e da deteção de pessoas. Para um determinado método de segmentação, o algoritmo avalia a qualidade dos resultados e, com base nesta avaliação, decide se o método está a produzir bons resultados no seguimento. Foi também usada uma implementação do método HOG para deteção de pessoas, baseado em *support vector machine*, e avaliado como alternativa à deteção baseada em segmentação.

Relativamente à diminuição dos tempos de processamento foram implementados métodos que, partindo de informações relativas à complexidade da cena (número de pessoas, distância entre elas ou consistência dos seus movimentos), ignoram determinadas fases do processamento.

Os testes realizados permitiram concluir que é possível uma alteração do método de segmentação, caso se saiba à partida a qualidade dos resultados de cada método. Apenas recorrendo a medidas sem informação de referência não foi possível essa adaptação. Concluiu-se também que, ao nível da deteção de pessoas, o detetor HOG pode ser considerado como uma alternativa ao detetor baseado em segmentação testado (procura de cabeças). Relativamente à diminuição do tempo de processamento verificou-se tal ser possível, sem uma perda significativa da qualidade do seguimento, em determinadas situações onde a complexidade da cena é reduzida (poucas pessoas na cena, elevada distância entre as pessoas e movimentações suaves).

Abstract

Computer vision is a rapidly expanding area, being already present in many aspects of everyday life. Real time methods for people detection and tracking assume, in this context, a very important role.

This dissertation seeks to examine the possibility of adapting algorithm complexity, according to information gathered at each processing stage.

Starting from an algorithm that perform human detection and tracking, a test platform was assembled in order to assess the performance of the algorithm. Quality evaluation measures, without *ground-truth* information, were implemented in the major modules of the algorithm. The measures are combined to give the algorithm the ability to decide on the best way forward. A decision may be made in terms of improving the quality of tracking results or by decreasing processing time.

For the improvement of the tracking results we try to adapt/modify the segmentation and people detection method. The algorithm evaluates the segmentation quality and, based on such review, decides whether the current method is to produce good tracking results. It has also used an implementation of the method HOG, for people detection, based on *support vector machine*, and evaluated as an alternative to segmentation-based detection.

Regarding the reduction of processing time were implemented methods that based on scene information (number of persons, the distance between them and consistency of their movements), attempt to skip processing phases.

The tests showed that it is possible to change segmentation or people detection method, but only if the the quality of the results of each method is known. Only by comparing the results of each method is possible to decide whether or not certain method leads to good results. Using only measures without *ground truth* information such adaptation was not possible. Regarding the decrease in processing time was found to be just as possible without significant loss of tracking quality, in certain situations where the scene complexity is reduced (few people in the scene, high distance between people and smooth movements).

Agradecimentos

Sirvo-me deste espaço para agradecer a todos quantos contribuíram para a realização desta dissertação.

Em primeiro lugar gostaria de agradecer ao INESC Porto pela oportunidade de desenvolver esta dissertação nas suas instalações, servindo-me dos recursos disponibilizados pela instituição.

Agradeço ao orientador, o professor Luís Corte-Real, e ao co-orientador, o Eng.º Pedro Carvalho, pelo acompanhamento, orientação e disponibilidade demonstrada durante a realização desta dissertação.

Ao Eng.º Lucian Ciobanu e ao Eng.º João Santos por toda a ajuda que me proporcionaram.

Por fim, gostava também de agradecer à minha família e à minha namorada pelo apoio, força e motivação que sempre me transmitiram.

A todos um muito obrigado!

Jorge Rodrigues

Conteúdo

1	Introdução	1
1.1	Motivação e contextualização	1
1.2	Objetivos	2
1.3	Estrutura da dissertação	4
2	Estado da Arte	5
2.1	Deteção de objetos	6
2.1.1	Deteção baseada em segmentação	6
2.1.2	Deteção baseada em aprendizagem	10
2.2	Descritores	11
2.3	Seguimento	13
2.4	Métodos de avaliação do desempenho	17
2.4.1	Com informação de referência	18
2.4.2	Sem informação de referência	20
3	Plataforma de testes	25
3.1	Descrição da plataforma	25
3.2	Algoritmo de seguimento	26
3.3	Métodos de segmentação	28
3.4	Deteção de pessoas	31
3.4.1	Baseada em segmentação	31
3.4.2	Histograma de gradientes orientados	32
3.5	Medidas de avaliação individuais	35
3.5.1	Complexidade da cena	35
3.5.2	Qualidade da segmentação	36
3.5.3	Qualidade da deteção e seguimento	37
3.6	Medidas de avaliação baseadas em informação de referência	38
4	Metodologia para avaliação de desempenho	43
4.1	Descrição das sequências de teste	43
4.2	Framework de avaliação	45
4.2.1	Validação das medidas	45
4.2.2	Método de decisão	51
4.3	Experiências a realizar	51
4.3.1	Medição do tempo de processamento	52
4.3.2	Alteração do método de segmentação/deteção de pessoas	52
4.3.3	Modificação do processo de definição da correspondência	53

5	Resultados	57
5.1	Segmentação	57
5.2	Deteção de pessoas	61
5.3	Seguimento	65
5.4	Alteração da segmentação/deteção	68
5.4.1	Alteração manual	68
5.4.2	Alteração automática	71
5.5	Redução do tempo de processamento	75
6	Conclusões e trabalho futuro	85
6.1	Conclusões	85
6.2	Trabalho futuro	87
	Referências	89

Lista de Figuras

2.1	Abordagem geral para sistemas de análise de movimento em humanos (Imagem retirada de [1]).	5
2.2	Deteção baseada em segmentação.	7
2.3	Aplicação do método de subtração do fundo (Imagem retirada de [2]).	7
2.4	Processo de segmentação. a) Possíveis cabeças. b) Cabeças encontradas. c) Pessoas segmentadas representadas por uma elipse. Imagem retirada de [2].	7
2.5	Arquitetura do processo de treino <i>AdaBoost</i> [3], usado por Viola em [4].	10
2.6	Deteção de pessoas com recurso a detetor HOG (Imagem retirada de [5]).	11
2.7	Deteção de pessoas em cenas densamente povoadas (Imagem retirada de [6]).	11
2.8	a) Proximidade. b) Velocidade máxima. c) Variação da velocidade. d) Velocidade dos pontos na vizinhança. e) Restrição relativa a constantes de rigidez dos objetos. Imagem retirada de [7].	13
2.9	Hierarquia dos modelos de seguimento PGM. Imagem retirada de [7].	14
2.10	Equações do filtro de <i>Kalman</i> estendido (EKF) [8].	16
2.11	Seguimento baseado na cor da camisola. Imagem retirada de [9].	17
2.12	<i>Ground truth</i> usada na abordagem híbrida. Imagem retirada de [10].	19
2.13	Aplicação do método de <i>Optical flow</i> (Imagem retirada de [11]).	20
2.14	Comparação de resultados de diferentes métodos de cálculo do <i>optical flow</i> para uma imagem de teste: a) Imagem de teste. b) Método Proposto por Horn & Schunck. c) Método Proposto por Lucas & Kanade. d) Método Proposto por Nagel. Imagens retiradas de [12].	21
2.15	Avaliação da segmentação. Imagem retirada de [13] e editada posteriormente.	22
2.16	Figura que exemplifica o cálculo do contraste no contorno do objeto. a) Objeto segmentado. b) Linhas perpendiculares ao contorno do objeto. c) Detalhe num pixel do contorno. (Imagem retirada de [14])	22
3.1	Diagrama de blocos da plataforma de testes.	25
3.2	Diagrama de blocos do algoritmo de seguimento (Extraída de [2] e editada posteriormente).	27
3.3	Diagrama de blocos do método de segmentação proposto por <i>Liyuan Li</i> (Imagem retirada de [15]).	30
3.4	Exemplo de criação e atualização de <i>codewords</i> (Imagem retirada de [16]).	31
3.5	Processo de segmentação (Imagem retirada de [2]). a) Possíveis cabeças. b) Cabeças encontradas. c) Pessoas segmentadas representadas por uma elipse. d) Imagem após retiradas as pessoas encontradas. e) Máscara de segmentação após retirados os objetos correspondentes às pessoas encontradas (note-se que mais uma pessoa foi detetada). f) Todas as pessoas foram detetadas.	32
3.6	Processo de treino do detetor HOG (Imagem retirada de [17]).	33

3.7	a) Imagem da média do gradiente para uma sequência de imagens de treino. b) Imagem original. c) Imagem do descritor HOG.(Imagem retirada de [17]). . . .	33
3.8	Exemplo de detecção de pessoas pelo método HOG e criação das respectivas elipses. a) <i>frame</i> 268 e 269 da sequência OSOW1. b) máscaras de segmentação criadas depois da detecção pelo método HOG.	34
3.9	Gráfico da função usada para pesar as medidas individuais de modo a obter a medida global. (Imagem retirada de [14]).	38
3.10	Método de <i>symetric partition-distance</i> . Na imagem da esquerda e da direita estão representadas duas partições diferentes da mesma imagem. Na imagem central estão selecionados os pontos onde as duas partições não coincidem. (Imagem retirada de [18]).	39
3.11	Imagens temporárias criadas para aplicação da medida <i>symetric particion-distance</i> ao seguimento.	39
3.12	Método de cálculo de TP,TN,FP,FN (Imagem retirada de [19]).	40
4.1	Imagens ilustrativas das sequências (frame 250): A) sequência OSOW1. B)sequência OSOW2.	44
4.2	Imagem ilustrativa da sequência PETS2006. (Frame 1050)	45
4.3	Módulo do <i>optical flow</i> ao longo da sequência OSOW2(cima). Número de objetos ao longo da sequência, obtido pela informação de referência (meio). Amplitude das movimentações(<i>GT Motion</i>) ao longo da sequência, obtido pela informação de referência (fundo).	46
4.4	Avaliação do método de segmentação <i>codebook</i> , para a sequência OSOW1, recorrendo às medidas <i>shape regularity</i> e <i>partition-distance</i>	48
4.5	Avaliação do método de segmentação AVG, para a sequência PETS2006, recorrendo às medidas <i>shape regularity</i> e <i>partition-distance</i>	48
4.6	Evolução da medida <i>Fuzzy</i> para a sequência OSOW1, calculada para todos os métodos de segmentação definidos.	50
4.7	Evolução da medida <i>symetrics partition-distance</i> para a sequência OSOW1, calculada para todos os métodos de segmentação definidos.	50
4.8	Diagrama ilustrativo da avaliação da detecção.	53
4.9	Evolução da tempo de match em comparação com o tempo de processamento total, ao longo da sequência OSOW1	54
4.10	Evolução da tempo de match em comparação com o tempo de processamento total, ao longo da sequência OSOW2	54
5.1	Segmentação de uma imagem pelos métodos de segmentação propostos.	57
5.2	Evolução da medida <i>symetric partition-distance</i> para os métodos de segmentação.	59
5.3	Evolução do tempo de segmentação, para cada método de segmentação definido.	60
5.4	Evolução da medida <i>symetric partition-distance</i> na avaliação da qualidade da detecção, pelo método de detecção de cabeças (segmentação AVG) e o pelo método de detecção HOG.	63
5.5	Comparação da evolução da tempo de detecção entre o método de procura de cabeças (segmentação AVG) e o método HOG.	64
5.6	Evolução da medida <i>symetric partition-distance</i> para os métodos de segmentação definidos ao longo da sequência OSOW1.	69
5.7	Evolução da medida <i>fuzzy</i> , na avaliação da qualidade do seguimento, ao longo das sequências de teste, partindo de cada um dos métodos de segmentação definidos e partindo do detetor HOG.	72

5.8	<i>bounding-box</i> do resultado do seguimento no <i>frame</i> 1100 da sequência OSOW1, partindo da segmentação MOG e da segmentação AVG, aplicado na imagem original.	74
5.9	<i>bounding-box</i> do resultado do seguimento, partindo da segmentação AVG, aplicado na imagem original, nos <i>frames</i> 440, 530, 670 e 750 da sequência OSOW1.	75
5.10	Evolução do tempo de processamento por <i>frame</i> , em segundos, no algoritmo base e no modo USkip.	77
5.11	Comparação entre a medida <i>symetric partition-distance</i> e o número de objetos por <i>frame</i> .	78
5.12	Valores para a covariância do erro, dados pelo filtro de <i>Kalman</i> , para o algoritmo base, na sequência OSOW1.	81
5.13	<i>Optical flow</i> calculado na vizinhança de cada objeto para o algoritmo base, na sequência OSOW1.	82
6.1	Diagrama ilustrativo do processo de avaliação dos resultados obtidos pelos diferentes módulos do algoritmo de seguimento.	85

Lista de Tabelas

4.1	Tempo de cálculo médio, máximo e mínimo (por <i>frame</i>) do módulo do <i>optical flow</i> para cada sequência de teste.	47
4.2	Coefficiente de correlação entre o número de objetos e o valor do módulo do <i>optical flow</i> , e distância percorrida pelos mesmos e o módulo do <i>optical flow</i>	47
4.3	Coefficientes de correlação entre a medida <i>shape regularity</i> e a medida <i>partition-distance</i> para as diversas sequências.	48
4.4	Coefficientes de correlação entre a medida <i>color contrast</i> e a medida <i>partition-distance</i> para as diversas sequências.	48
4.5	Coefficientes de correlação entre as medidas <i>symetric partition-distance</i> e <i>fuzzy</i> , calculados para cada sequência de imagens e para cada método de segmentação.	50
5.1	Tempo médio, máximo e mínimo (em segundos) para a segmentação por <i>frame</i> , usando os métodos de segmentação definidos.	61
5.2	Resultados da detecção pelo método de procura de cabeças.	62
5.3	Comparação entre os resultados da detecção obtidos pelo método de procura de cabeças e pelo método HOG.	62
5.4	Comparação do tempo de detecção e do tempo total de processamento, em segundos, entre o detetor de cabeças e o detetor HOG.	65
5.5	Resultados do seguimento, partindo da segmentação obtida por cada um dos métodos definidos.	66
5.6	Tempos necessário, em segundos, para que o algoritmo proceda ao seguimento, usando as segmentações obtidas pelos diferentes métodos de segmentação definidos.	66
5.7	Comparação dos resultados do seguimento entre o detetor pela procura de cabeças e o detetor HOG, para as sequências de teste.	67
5.8	COMparação do tempo de processamento (em segundos) para o seguimento entre o detetor HOG e o detetor baseado na procura de cabeças.	67
5.9	Valores obtidos pela medida <i>F-score</i> quando são usadas as segmentações obtidas por cada um dos métodos de segmentação, para as sequências de teste.	69
5.10	Comparação dos resultados do seguimento obtidos pela combinação entre segmentações e os resultados obtidos partindo da segmentação AVG.	70
5.11	Comparação do tempo total de processamento, em segundos, do algoritmo de seguimento usando a segmentação AVG e as segmentações combinadas.	71
5.12	Comparação dos resultados do seguimento para as sequências de teste entre os resultados da segmentação AVG, da segmentação combinada pela medida <i>symetric partition-distance</i> e da segmentação combinada pela medida <i>fuzzy</i>	73
5.13	Comparação entre algoritmo base e modo USkip, para a sequência OSOW1.	76
5.14	Variação do tempo de <i>match</i> e do tempo total, em segundos, para as sequências de teste, no modo USkip.	76

5.15	Comparação dos resultados do seguimento entre algoritmo base (segmentação pelo método AVG) e modo ASkip, para as sequências de teste.	79
5.16	Resultados ao nível do tempo de processamento para o algoritmo de seguimento base (segmentação AVG) e para o modo ASkip.	79
5.17	Comparação entre os resultados do algoritmo base (segmentação CB) e o resultados no modo ASkip (segmentação CB).	80
5.18	Resultados ao nível do tempo de processamento para o algoritmo de seguimento base(segmentação CB) e para o modo ASkip.	80
5.19	Comparação dos resultados do seguimento entre algoritmo base (segmentação pelo método AVG) e modo ASkip, para as sequências de teste.	82
5.20	Resultados ao nível do tempo de processamento, em segundos, para o algoritmo de seguimento base (segmentação AVG) e para o modo CSkip.	83

Abreviaturas e Símbolos

APD	Asymmetric Partition Distance
ASkip	Assisted Skip
AVG	Running Average
BN	Bayesian Network
DBN	Dynamic Bayesian Network
BLOB	Binary Large Object
CB	Codebook
CLEAR	Classification of Events, Activities, and Relationships
CSkip	Covariance Skip
DoG	Difference of Gaussians
FGD	Foreground Detection
FPS	Frames Per Second
GPL	General Public License
HMM	Hidden Markov Model
HOG	Histogram of Oriented Gradients
JPDAF	Joint Probabilistic Data Association Filter
KFM	Kalman Filter Model
LoG	Laplacian of Gaussian
MHT	Multiple Hypothesis Tracking
MODA	Multiple Object Detection Accuracy
MODP	Multiple Object Detection Precision
MOG	Mixture of Gaussians
MOTA	Multiple Object Tracking Accuracy
MOTP	Multiple Object Tracking Precision
MRF	Markov Random Field
OFM	Optical Flow Modulus
OpenCV	Open Source Computer Vision
PD	Partition Distance
PETS	Performance Evaluation of Tracking and Surveillance
PGM	Probabilistic Graphical Model
SIFT	Scale Invariant Feature Transform
SPD	Symmetric Partition Distance
SURF	Speeded Up Robust Feature
USkip	Unassisted Skip
VACE	Video Analysis and Content Extraction

Capítulo 1

Introdução

1.1 Motivação e contextualização

Nas últimas décadas tem-se assistido a um rápido crescimento e desenvolvimento na área das tecnologias de informação. O aumento da capacidade de armazenamento de dados e a elevada qualidade das câmaras fotográficas e de vídeo a baixo preço aliados ao desenvolvimento das redes de comunicação e transferência de dados levaram a uma procura crescente de sistemas automáticos de monitorização e controlo [20].

Os sistemas de recolha e de análise de imagens de vídeo tornaram-se acessíveis e estão hoje presentes nas mais variadas áreas da ciência. Na medicina, sistemas de análise de imagem são muitas vezes importantes como complemento ao diagnóstico. Imagens obtidas através das mais variadas técnicas (raios-X, ecografia), precisam ser processadas de forma a reduzir ruído e facilitar a sua análise. Malformações nos ossos, nos tecidos, quistos e infeções são muitas vezes detetados de forma automática usando sistemas de análise de imagem.

Na indústria, cada vez mais se substituem operários por robôs ou manipuladores no transporte de materiais perigosos. Sistemas de visão computacional incorporados em sistemas de condução autónoma possibilitam aos robôs uma locomoção mais segura. A inspeção de produtos em linhas de montagem é por vezes feita de forma automática, recorrendo a sistemas baseados em visão computacional.

No entretenimento têm vindo a desenvolver-se aplicações baseadas na captura e análise de imagem. Tem-se verificado um rápido crescimento em aplicações, quer em número quer na qualidade das mesmas, disponíveis para telemóveis, *smartphones* e consolas de jogos, baseadas em análise de imagem. Aplicações deste tipo dão às respetivas plataformas uma maior interatividade com o utilizador, traduzindo-se num aumento do número de utilizadores.

A videovigilância é outra das áreas em permanente desenvolvimento, motivando variados estudos. Para além dos fatores técnicos descritos anteriormente (capacidade de processamento e qualidade das imagens), fatores sociais como violência e ações criminosas fazem com que cada vez mais sejam utilizados sistemas videovigilância com objetivo de evitar agressões, roubos ou atos de vandalismo, bem como obter provas dessas mesmas ações. Detetar e seguir pessoas são

as principais tarefas pedidas a estes sistemas, sendo de grande importância na detecção de intrusões, monitorização ou contagem de pessoas. Uma vez que a gravação contínua de imagens leva à obtenção de uma grande quantidade de dados, grande parte dos quais sem qualquer informação relevante, é necessário dotar os sistemas de capacidade de análise. O sistema, de forma automática deverá ser capaz de analisar a situação e decidir em conformidade. A decisão poderá passar pelo tratamento automático da situação ou pela chamada de um operador. Assim sendo, estes sistemas dispensam a presença constante de um operador, sendo estes chamados apenas quando necessários (situações de perigo, erro...). Sistemas com estas capacidades traduzem-se em aumento de segurança e conforto por parte de quem os usa, assim como uma redução de custos.

1.2 Objetivos

Esta dissertação enquadra-se na área da monitorização humana, recorrendo a uma câmara fixa, situada num ponto elevado, em ambientes interiores. As imagens recolhidas pela câmara serão processadas de forma a detetar as pessoas presentes e respetivas movimentações.

Em algoritmos de seguimento em tempo real, a *frame rate* utilizada é um fator importante no sucesso do algoritmo. Uma *frame rate* elevada leva a uma maior quantidade de imagens recolhidas por intervalo de tempo, reduzindo a distância percorrida pelas pessoas entre cada *frame*, facilitando o seguimento das mesmas, mas reduzindo o tempo disponível para processamento, visto que há mais imagens a processar no mesmo intervalo de tempo. Uma redução da *frame rate* corresponde a redução do número de imagens capturadas por intervalo e, conseqüentemente, um aumento do tempo disponível para processamento. Contudo, existe um limite abaixo do qual o algoritmo pode ser incapaz de operar corretamente. Uma *frame rate* demasiado baixa corresponde a um intervalo de tempo elevado entre capturas consecutivas, podendo levar a movimentações na imagens demasiado grandes para serem tratadas pelo algoritmo, introduzindo um elevado grau de incerteza e perdendo-se o seguimento. Por outro lado, uma *frame rate* demasiado elevada poderá conduzir a um aumento do peso computacional do algoritmo acima das capacidades de processamento do sistema, gerando bloqueios e perdendo-se o seguimento. A complexidade do cenário também é outro aspeto a ter em conta. Fatores como o número de pessoas, a distância entre elas ou a velocidade com que se movem influenciam a eficácia do algoritmo. Estes fatores podem introduzir alguma incerteza no algoritmo, comprometendo os resultados.

Tendo em conta tudo isto, pretende-se analisar a possibilidade de adaptar/modificar a complexidade do algoritmo em função do contexto. A diminuição da complexidade do algoritmo pode passar por uma diminuição do tempo de processamento (dentro de limites aceitáveis) ou pela implementação de métodos mais rápidos de deteção e seguimento, ainda que mais sensíveis ao ruído. Foram implementados e testados diferentes métodos de deteção de pessoas, com complexidades diferentes, assim como medidas de avaliação da qualidade dos mesmos. As medidas serão obtidas durante a execução do algoritmo de seguimento e usadas de forma a possibilitar uma adaptação do mesmo. Pretende-se portanto estudar formas de reduzir a complexidade do algoritmo, e conseqüentemente o tempo de processamento, em situações em que não seja necessário

um processamento muito elevado, sem comprometer de forma significativa os resultados.

1.3 Estrutura da dissertação

No primeiro capítulo é feita uma contextualização do tema, detecção e seguimento de pessoas em cenários complexos. É também apresentada a forma como o contexto sociocultural, económico e tecnológico incentivam a utilização do vídeo e o processamento automático. Foram ainda identificados os problemas genéricos, os principais objetivos a atingir bem como o cenário de aplicação.

No segundo capítulo é feito o levantamento do estado da arte, de forma a conhecer as técnicas e métodos mais comuns relativamente a este tema. Foram estudadas publicações de diferentes autores referentes a métodos de detecção e seguimento de pessoas assim como medidas avaliação da qualidade dos mesmos.

No terceiro capítulo é apresentada a plataforma de testes criada. São descritos os blocos principais que a constituem e explicada a sua função na plataforma. É descrito o algoritmo de seguimento e os métodos de segmentação implementados. São também apresentadas as medidas a usar na avaliação do desempenho da detecção e do seguimento.

No capítulo 4 é dada a conhecer a metodologia para avaliação do desempenho da plataforma de testes. As medidas apresentadas no capítulo 3 serão validadas usando medidas baseadas em informação de referência. Baseando-se nos resultados obtidos pelas medidas de avaliação, será criado um método de tomada de decisão relativamente à adaptação da complexidade do algoritmo. São também identificadas as sequências de imagens a usar assim como os testes a efetuar.

Todos os resultados obtidos estão apresentados no capítulo 5, sendo tiradas conclusões sobre os mesmos e apresentadas sugestões relativamente a futuros desenvolvimentos no capítulo 6.

Capítulo 2

Estado da Arte

Foram analisados diversos artigos relativos ao tema da detecção de seguimento de pessoas em seqüências de vídeo. Apesar de diferentes métodos propostos pelos autores, foram identificadas quatro fases principais, detecção, seguimento, estimação da postura corporal e reconhecimento [1]. Na figura 2.1 é apresentado um diagrama que descreve a abordagem geral dos algoritmos de detecção e seguimento de pessoas em seqüências de vídeo. Dependendo do objetivo específico, algumas destas etapas podem no entanto ser ignoradas. No contexto desta dissertação, apenas se pretende obter um registo da posição de cada objeto durante o seu percurso na cena, pelo que as etapas de estimação da postura corporal e reconhecimento podem ser ignoradas.

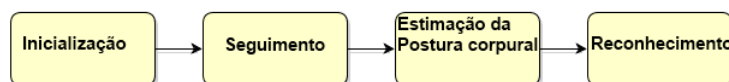


Figura 2.1: Abordagem geral para sistemas de análise de movimento em humanos (Imagem retirada de [1]).

A fase de inicialização (*Initialisation*) é responsável pela definição do modelo humano, aproximando a forma ou aparência, que irão ser usados como forma de representar pessoas. As representações da forma humana são variadas e podem ser formas básicas como elipses ou retângulos, ou superfícies. É também nesta fase que é definido o modelo da câmara responsável pelo mapeamento entre a imagem e o mundo real. Conhecendo as dimensões e a posição de determinado objeto é possível, recorrendo a este modelo, obter as suas reais dimensões. Deste modo, é possível adicionar ao algoritmo de seguimento algumas restrições e forma a melhorar o seu desempenho. Existem vários métodos para este efeito mas não serão tratados neste estudo [21, 22].

A fase do seguimento (*Tracking*) é composta essencialmente por duas etapas, detecção e correspondência temporal. A detecção pretende, partindo das imagens recolhidas, retirar informação sobre a posição de objetos de interesse numa imagem. Dependendo dos objetivos, a detecção poderá fornecer outras informações sobre os objetos. Aos objetos detetados é atribuída uma identidade única. A correspondência temporal, como o nome indica, pretende efetuar uma correspondência entre os objetos nas diferentes imagens que fazem parte da seqüência de vídeo, atribuindo a

cada objeto, identidade atribuída na imagem anterior. Desta forma consegue-se um registo das movimentações dos objetos ao longo da cena.

Após o seguimento, terá início a fase denominada *Pose estimation*. Sabendo a localização de cada pessoa presente na imagem, poderá ser necessário identificar a respetiva postura corporal. Dependendo dos objetivos e dos resultados pedidos ao algoritmo de seguimento, esta fase pode ser entendida como uma fase de pós-processamento ou como uma fase essencial do mesmo. Vários níveis de precisão podem ser pedidos nesta fase. Poderá apenas ser pedida informação relativa ao centro de massa da pessoa a ser seguida ou, em situações extremas a posição da cabeça, dos braços ou das mãos relativamente ao corpo.

A fase *Recognition* poderá ser entendida como uma fase de pós-processamento no algoritmo de seguimento. O reconhecimento consiste em classificar o movimento de uma pessoa, reconhecer o que ação está associada a essa pessoa. Geralmente pede-se a identificação de ações simples como caminhar ou correr, mas ações mais complexas podem também ser pedidas.

Uma vez que não se pretende efetuar um reconhecimento da postura corporal, apenas será dada atenção às duas primeiras fases, *Initialization* e *Tracking*. Interessa-nos em particular a posição do centro de massa ou o retângulo envolvente de todas as pessoas presentes na imagem. Assim, a fase de *Pose estimation* e a fase de *Recognition* não têm interesse no contexto desta dissertação.

2.1 Detecção de objetos

Como referido anteriormente, a deteção consiste em localizar objetos de interesse na imagem. Neste caso, consiste em retirar da imagem informação sobre o número de pessoas presentes, bem como as suas posições e respetivas dimensões. Existem abordagens diferentes para este efeito, das quais destacamos a deteção baseada em segmentação de imagem e a deteção baseada em *support vector machines*(SVM).

2.1.1 Deteção baseada em segmentação

No contexto da deteção de pessoas, as técnicas baseadas em segmentação são muito usadas [23, 24, 2]. A segmentação é o processo pelo qual uma imagem digital é dividida em regiões, agrupando os pixels segundo um determinado critério. Sendo a segmentação um processo muito sensível a ruído (frequentemente são considerados pixels da imagem como sendo pertencentes à pessoa a segmentar quando na verdade são parte do fundo, ou vice-versa), é necessário detetar na imagem segmentada o que realmente é uma pessoa ou apenas ruído/erro de segmentação. Neste contexto, o método de procura de cabeças é amplamente usado. Trata-se de um método de deteção de pessoas baseado em segmentação, uma vez que é necessário obter em primeiro lugar a imagem segmentada para depois se proceder à deteção. Como se percebe pela análise da imagem 2.2, o primeiro passo consiste na criação de um modelo de fundo.

Inspirados em Haritaoglu et al [23], Siebel et al [24] e Tao et al [2] aplicam este método nos seus algoritmos de seguimento, como detetor de pessoas. Assumindo um valor médio para

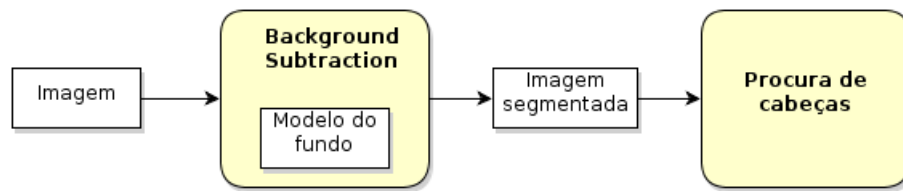


Figura 2.2: Detecção baseada em segmentação.

a cabeça humana, são procurados na vertical áreas que tenham a área pedida. Na figura 2.3, é exemplificada a aplicação do método de subtração do fundo, usando como modelo de fundo a técnica *Mixture of Gaussians*.



Figura 2.3: Aplicação do método de subtração do fundo (Imagem retirada de [2]).

Após a aplicação do método de subtração do fundo foi aplicado o método de procura de cabeças, com o objetivo de detetar cada pessoa presente na imagem de forma individual, partindo da imagem segmentada. A figura 2.4 mostra a aplicação deste método.

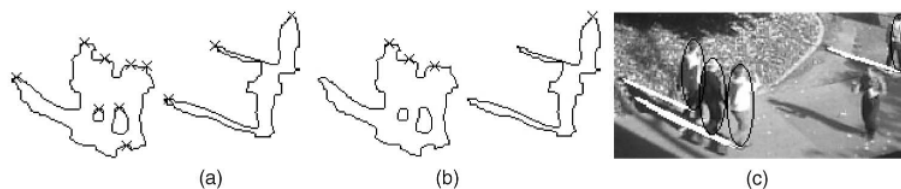


Figura 2.4: Processo de segmentação. a) Possíveis cabeças. b) Cabeças encontradas. c) Pessoas segmentadas representadas por uma elipse. Imagem retirada de [2].

Background Subtraction

A subtração do fundo é uma abordagem muito utilizada na deteção de objetos móveis na imagem [2]. Deve ser o mais robusta possível de forma a tratar alterações de iluminação ou objetos móveis na cena que não sejam de interesse (chuva, sombras, movimentos devido ao vento). Este método consiste na subtração da imagem a um modelo do fundo. O resultado desta segmentação é convertido numa imagem binária, onde os pixels a preto corresponderão a pixels móveis na imagem. O modelo de fundo vai sendo continuamente atualizado com base nas novas imagens, de forma a poder adaptar-se a mudanças subtis na iluminação ou pequenos movimentos na imagem. Um primeiro passo consiste na criação de um modelo de fundo, podendo ser utilizadas diversas técnicas. Como modelo de fundo mais simples, podemos considerar a utilização de uma imagem estática do ambiente onde vai ser efetuado o seguimento, num momento onde nenhum objeto esteja presente.

Trata-se de um modelo de fundo não adaptativo, uma vez que o mesmo não se adapta a possíveis alterações no fundo, nomeadamente na iluminação. Este tipo de modelo praticamente não é usado para efeitos de seguimento em ambientes não controlados, sendo aplicado por alguns autores apenas em ambientes controlados [25]. Relativamente a modelos de fundo adaptativos, são referidos na bibliografia variadas opções:

- *Running average*

O modelo de fundo criado pelo método *Running average* é geralmente de criação e atualização rápida mas exige grande quantidade de memória [26, 27]. Cada pixel do modelo de fundo é baseado no seu histórico recente. Geralmente, o histórico recente compreende apenas um número fixo de *frames* e é dada maior importância aos mais recentes. Este modelo na incorpora nenhuma relação com as intensidades dos pixels na vizinhança. A equação 2.1 apresenta a fórmula de cálculo do modelo de fundo. O parâmetro α é a taxa de aprendizagem e, normalmente toma o valor 0,05. Isto significa que é considerada a influência dos últimos $1 \div \alpha$ *frames*. A grande desvantagem deste método é a não existência de um método automático de escolha do valor do *threshold* a usar. Um determinado valor de *threshold* pode resultar muito bem em algumas situações e não ser o ideal em outras.

$$fundo(t) = (1 - \alpha) \times fundo(t - 1) + \alpha \times imagem(t) \quad (2.1)$$

- *Mixture of Gaussians*

Em [28] foi proposto que a cor de cada pixel fosse modelizada usando uma distribuição gaussiana. A média e covariância do modelo são calculados através da observação de imagens consecutivas. Depois de construído um modelo de fundo, cada pixel da cena é classificado como pertencente ao fundo ou não, de acordo com a sua distância ao pixel correspondente no modelo. Verificou-se contudo, que este modelo não produzia bons resultados em ambientes exteriores [29]. Um método alternativo foi usado em [30] utilizando uma mistura de Gaussianos para criar o modelo do fundo. Cada pixel na imagem é comparado com todos os Gaussianos presentes no modelo. Se for encontrado o Gaussiano correspondente, a média e a variância são atualizadas, caso contrário é adicionado um novo Gaussiano com média e variância igual à média e variância do pixel.

- *Kernel Density Estimators*

Outro método com bons resultados foi apresentado em [31] e usa não só a cor do pixel correspondente mas também informação dos seus vizinhos mais próximos. Esta abordagem corrige problemas resultantes de pequenos movimentos da câmara, mas apresenta um consumo de memória e tempo de cálculo elevados.

- *Mean shift*

Em [32], o modelo de fundo foi construído pelo método de *Mean shift*. Como se trata de um método iterativo, tanto o tempo de construção do modelo como o consumo de memória podem ser bastante elevados.

- *SKDA (Sequential KD Approximation)*

Com o intuito de ultrapassar as limitações apresentadas pelo método Mean shift, foi usado em [33] o método de criação do modelo de fundo denominado *Sequential KD Approximation* (SKDA). É um método combinado de estimação e propagação através do qual se obtém uma aproximação da função densidade.

- *Eigenbackgrounds*

O método de *Eigenbackgrounds* [34] baseia o seu funcionamento na técnica de decomposição em vetores próprios. Esta técnica de decomposição é uma forma de reduzir a dimensionalidade de um espaço. A decomposição em vetores próprios é aplicada a uma sequência de imagens obtendo uma imagem vetorizada correspondente aos objetos móveis da imagem. Segundo os autores, esta abordagem é mais rápida que o método de *Mixture of Gaussians*.

O método de *Background subtraction*, como foi dito anteriormente, é um método muito usado para deteção de objetos móveis numa imagem. É necessário um período de treino para o modelo de fundo, o que significa que não pode ser aplicado numa imagem isolada. Para segmentação de imagens isoladas, uma vez que não temos qualquer informação relativa ao movimento, são apresentados de seguida alguns métodos de segmentação, que, apesar de não serem frequentemente usados em algoritmos de seguimento em tempo real, são muito referenciados na bibliografia consultada.

O método de *region growing*, ou crescimento de regiões, é um método muito usado em análise de imagem. Aplicado em [35], o método consiste em encontrar, usando um critério bastante apertado, pixels que correspondam aos objetos a segmentar. Estes pixels são chamados de sementes (*seeds*). Depois disto, são agregados outros pixels na vizinhança da semente e que satisfaçam a condição de agregação. Esta condição deve ser mais permissiva que a condição anterior. O método termina quando mais nenhum pixel for adicionado à semente [36].

Outro método de segmentação identificado foi o *active contour*. É usado em [37] com o objetivo de encontrar o contorno do objeto. Em cada iteração do algoritmo, o contorno encontrado vai-se aproximando do real contorno do objeto. Este método tem alguns inconvenientes a ter em conta. Em primeiro lugar, o peso computacional. Apesar de alguns melhoramentos, este método requer grande processamento, uma vez que em cada iteração o método deve minimizar a função de energia. Outro problema reside no facto deste método não obter bons resultados nas partes côncavas do contorno do objeto. Por fim, o facto de necessitar de uma estimativa inicial do contorno do objeto a segmentar [38].

Em [39] foi aplicada um dos mais simples métodos de *clustering*, *K-means*. O *Clustering* é uma técnica de agrupamento de pixels segundo as suas características formando *clusters*. Para um número previamente definido de *clusters*, define-se para cada um o seu centroide. De seguida associa-se a cada pixel da imagem o *cluster* cujo centroide se encontra mais próximo do pixel em questão. Por fim volta a calcular-se o centroide de cada *cluster* e repete-se a associação de cada pixel ao cluster mais próximo. Termina-se quando nenhum pixel mudar de *cluster*. É um método

bastante simples e rápido mas tem como inconveniente a necessidade de conhecimento prévio do número de *clusters* [40].

2.1.2 Detecção baseada em aprendizagem

Como alternativa aos métodos baseados em segmentação apresentam-se os métodos baseados em *support vector machines* (SVM). Os SVM são conceitos estatísticos que, após uma aprendizagem controlada, são capazes de analisar e reconhecer padrões num conjunto de dados. O SVM é treinado recorrendo a dois conjuntos de dados, cada qual correspondendo a uma classe. Um conjunto referente a amostras de pessoas e outro referente a amostras de fundo. Depois disto, o SVM é capaz de classificar os pixels da imagem como pertencente a uma ou outra classe, neste caso, se pertencente a uma pessoa ou ao fundo.

Papageorgiou e Poggio [41] apresentam o método designado por *Haar wavelets*. Como descrito pelos autores, as imagens foram representadas recorrendo a *Haar wavelets* e SVM para classificar os pixels como objeto de interesse ou não. Os autores apresentam os resultados da aplicação do método à deteção de faces, pessoas e carros. Uma vez que esta abordagem não inclui qualquer informação relativa ao movimento dos objetos, Viola [4] apresenta uma solução na qual esta informação é tida em conta. O algoritmo de deteção de pessoas apresentado compara os resultados entre duas imagens consecutivas, obtendo assim informação relativamente ao movimento dos objetos. O treino do detetor é feito utilizando o método *AdaBoost* [3]. Trata-se de um método para a construção de um classificador mais robusto, utilizando uma combinação linear de classificadores menos robustos. Desta forma, o algoritmo proposto por Viola, funde os dois tipos de informação (aparência e movimento) no mesmo detetor de pessoas, figura 2.5. Segundo o autor, o sistema proposto pelo mesmo apresenta resultados bastante satisfatórios na deteção de pessoas nas mais variadas situações. A implementação descrita foi testada a uma *frame rate* de 4 *frames/segundo*, sendo capaz de detetar pessoas relativamente afastadas da câmara, mantendo um baixo número de falsas deteções.

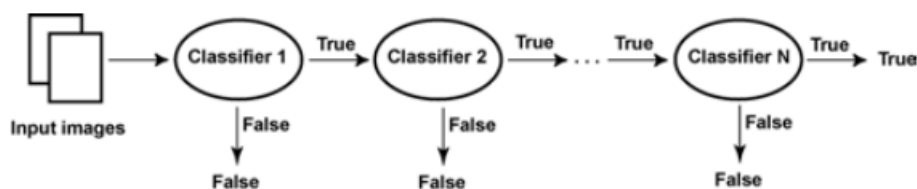


Figura 2.5: Arquitetura do processo de treino *AdaBoost* [3], usado por Viola em [4].

Mais recentemente, Dalal e Triggs [17] apresentam o detetor *Histogram of Oriented Gradients* (HOG), com bons resultados. Este método baseia-se no pressuposto que a forma e aparência de determinado objeto podem ser bem caracterizados pela intensidade do gradiente e orientação dos contornos do mesmo, sem o prévio conhecimento da sua localização. A imagem 2.6 mostra um exemplo de deteção usando o detetor HOG.

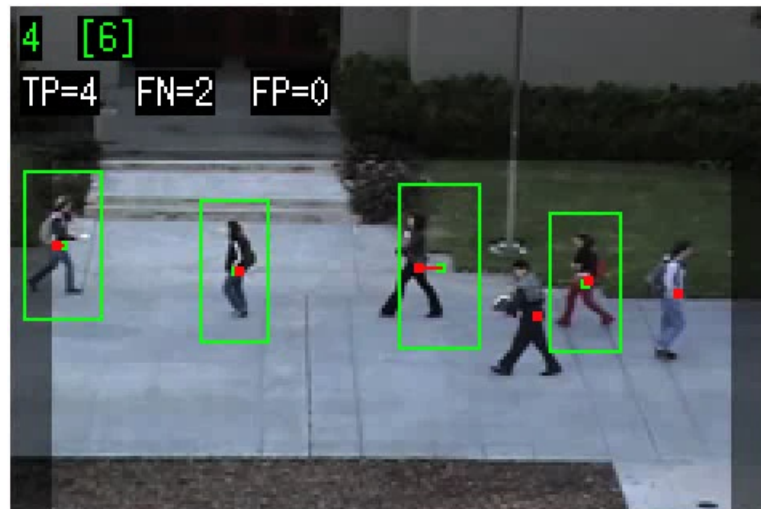


Figura 2.6: Detecção de pessoas com recurso a detetor HOG (Imagem retirada de [5]).

Leibe [6], apresenta ainda um método de deteção de pessoas em cenas densamente povoadas que combina uma representação da forma do objeto, obtida através de um processo de aprendizagem, com a transformada de *Hough* generalizada. Segundo os autores, este método permite uma deteção dos objetos de interesse em novas imagens, mas também criar uma segmentação estatística relativamente aos objetos detetados. A imagem 2.7 mostra um exemplo da deteção baseada neste método.

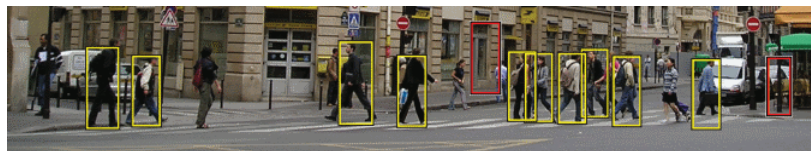


Figura 2.7: Detecção de pessoas em cenas densamente povoadas (Imagem retirada de [6]).

2.2 Descritores

Após a segmentação da imagem é necessário proceder a uma descrição de todos os objetos. Interessa nesta fase distinguir os objetos de interesse de outros que podem ser descartados. É necessário também distinguir diferentes instâncias do mesmo objeto. Numa cena em que várias pessoas estão presentes é fundamental detetar o número de pessoas correspondente, mas também verificar que são entidades independentes, apesar de terem algumas características semelhantes. Esta operação é bastante complexa e, caso se usem descritores que não sejam os ideais, podem comprometer todo o processo. Assumindo que todos os objetos têm características que os tornam únicos, o desafio é encontra-las.

Os histogramas de cor, usados em [42, 43, 44], apresentam-se como um dos mais simples descritores de objetos. Comparando os histogramas de cor de cada objeto, pode facilmente obter-se uma descrição de cada um deles. Este descritor possui uma boa relação entre simplicidade e

eficácia, pelo que é muito usado pelos autores aqui estudados, principalmente quando combinado com outras características. Os histogramas de cor são invariantes à translação e à rotação em torno do eixo de visão mas são sensíveis a situações de ocultação e alterações de escala. Com o intuito de ultrapassar as limitações dos descritores sensíveis à escala, é apresentado o descritor *Scale Invariant Feature Transform*(SIFT) [45]. Este descritor tem como principal vantagem, além da já referida invariância a alterações de escala, o facto de ser também invariante a variação da orientação e, até certo ponto, na iluminação da imagem a analisar. Em primeiro lugar, são calculados pontos chave na imagem recorrendo aos valores máximos e mínimos da função diferença de Gaussianos (DoG). A DoG é uma função muito usada em tratamento de imagem com o objetivo de realçar os contornos dos objetos presentes numa imagem digital. Ao contrário de alguns métodos de realce de contornos que passam pelo realce dos detalhes de alta frequência, realçando também ruído eventualmente presente na imagem, a DoG remove as altas frequências da imagem, removendo também o ruído, à custa de uma diminuição do contraste, o que a torna bastante útil no processamento de imagens com elevado ruído. Deve-se realçar o facto de os descritores SIFT terem um elevado peso computacional, pelo que não devem ser usados em situações de tempo real [45].

O descritor SURF [46] é usado como alternativa ao SIFT. Apesar de menos discriminativo, este descritor é mais eficiente que o descritor SIFT. É baseado na matriz Hessiana, mas, com o objetivo de reduzir o tempo de computação, recorre a um detetor baseado em *DoH*. Este descritor deve a sua rapidez e eficiência ao uso da imagem integral. Para além disso apenas usa vetores de 64 posições (enquanto que o SIFT usa vetores de 128 posições) o que reduz o tempo de computação e *matching* aumentando o robustez. Em [47] foram feitas algumas simplificações ao descritor com o intuito de o mesmo ser usado em tempo real. Segundo os autores os objetivos foram atingidos, mantendo uma performance semelhante ao descritor SURF base.

O *Histogram of Oriented Gradients* [17], referenciado anteriormente para deteção de pessoas, é outro descritor sugerido na bibliografia. É calculado dividindo cada região da imagem em pequenas células e, para cada pixel em cada célula, é calculada a direção do gradiente. As células são agrupadas em blocos e, com o objetivo de aumentar a invariância à iluminação, é feita uma normalização dos blocos. Os histogramas normalizados de todos os blocos produzem um vetor que descreve o objeto. Uma vez que este descritor opera em pequenas células, é invariante à geometria do objeto.

Os descritores apresentados, quando usados individualmente, podem ser insuficientes para descrever um objeto inequivocamente. A forma pode ser descrever uma pessoa isolada mas não é capaz de distinguir inequivocamente uma pessoa numa multidão. Por outro lado a cor pode descrever dois objetos de cores diferentes mas caso tenham a mesma cor isso já não é possível. Deste modo pode ser necessário usar vários descritores em simultâneo. Por exemplo, usando a cor e a forma podemos discriminar corretamente um objeto entre outros de variadas formas e cores. Em [48] é feita uma comparação dos resultados de várias combinações de características baseadas em cor e textura. Em [49] os objetos foram descritos e caracterizados usando a cor e gradiente.

2.3 Seguimento

Em análise de vídeo, o seguimento é usado sempre que se pretende identificar as movimentações de determinado objeto numa sequência de vídeo. Depois de detetados os objetos de interesse, ao qual foi atribuída uma identidade e modelizados usando um modelo de forma, pretende-se gerar uma trajetória do mesmo através da sua localização em cada *frame* do vídeo.

A principal dificuldade consiste na atribuição da mesma identidade a objetos detetados em instantes de tempo diferentes. É um processo complexo, principalmente quando os objetos estão parcialmente ou totalmente ocultos, para o qual foram propostos variados métodos.

Os métodos de seguimento estudados foram divididos em duas categorias: determinísticos e probabilísticos. Os métodos determinísticos baseiam-se em *matching*. A correspondência entre objetos em *frames* consecutivos é conseguida através da minimização de uma função de custo, combinada com algumas restrições ao movimento. Nos métodos probabilísticos é feita uma estimação do estado do objeto no *frame* seguinte. Abordagens baseadas em métodos probabilísticos lidam melhor com o ruído presente nas imagens ou perturbações não previstas nas imagens [7].

Métodos Determinísticos

Como referido anteriormente, os métodos determinísticos de correspondência definem uma função custo para a associação entre objetos entre *frames* consecutivos. Para cada objeto no *frame* pretende-se obter apenas uma correspondência no *frame* anterior, minimizando a função custo. No fundo, trata-se de um problema de otimização. A função custo é definida usando uma combinação de restrições de velocidade e rigidez do objeto que limitam o número de associações possíveis. Assumindo que, entre *frames* consecutivos, a direção do movimento, a velocidade e a distância entre os objetos não sofrem variações bruscas e que os objetos são corpos rígidos é possível obter correspondência entre os mesmos. A figura 2.8, apresenta uma interpretação gráfica das restrições acima descritas.

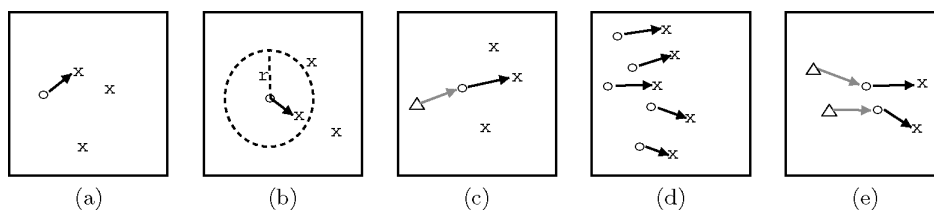


Figura 2.8: a) Proximidade. b) Velocidade máxima. c) Variação da velocidade. d) Velocidade dos pontos na vizinhança. e) Restrição relativa a constantes de rigidez dos objetos. Imagem retirada de [7].

Apresentamos agora algumas soluções para o problema da correspondência de objetos entre *frames* consecutivos. Sethi and Jain [50] resolveram este problema usando uma combinação de constantes relativas à distância entre os objetos e à sua rigidez. O seu algoritmo inicia-se definindo a correspondência entre os objetos em imagens consecutivas usando a distância mínima entre eles. As correspondências são depois atualizadas iterativamente de forma a minimizar uma função de custo, previamente definida, e que, como dito anteriormente, leva em consideração a distância mínima e constantes relativas à rigidez dos objetos. Este método é incapaz de lidar com oclusões

ou entradas e saídas de cena. De forma a resolver este problema Salari and Sethi [51] efetuaram em primeiro lugar a correspondências para os objetos detetados e adicionaram pontos hipotéticos para a correspondência dos objetos restantes. Veenman et al [52] expandiram o trabalho apresentado por Sethi and Jain [50] introduzindo uma constante relativa à velocidade dos objetos. A correspondência é feita considerando que pontos vizinhos que constituem o objeto não apresentam grandes variações de velocidade. Ou seja, pontos próximos movem-se com velocidades semelhantes. A função custo é minimizada pelo método húngaro [53]. Esta abordagem permite efetuar a correspondência em situações de oclusão, mas não considera situações de entrada e saída de objetos da cena.

O *Optical flow* consiste num campo de vetores que definem a translação de pixeis pertencentes a um objeto. No fundo, representa/mede a quantidade de movimento que cada pixel possui. Em [54] é feito o seguimento em tempo real de várias pessoas. Os objetos foram detetados usando *optical flow* e representados pelo retângulo envolvente. Em [55] é apresentado um método de seguimento que combina *optical flow* com filtro de partículas e explora situações onde a aplicação de cada método individualmente falha.

Métodos Probabilísticos

O seguimento baseado em métodos probabilísticos constitui uma alternativa ao seguimento baseado em métodos determinísticos. Na figura 2.9, é apresentado a hierarquia dos métodos probabilísticos de seguimento, *Probabilistic Graphical Models*(PGM).

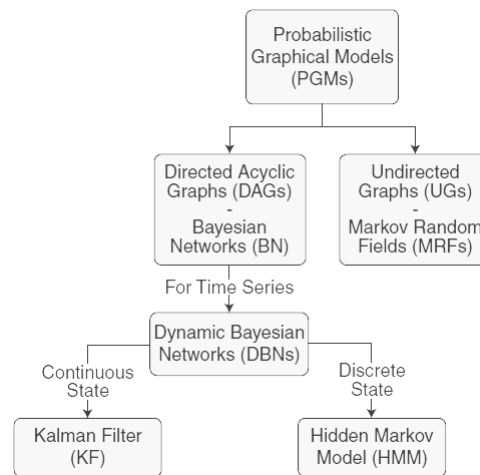


Figura 2.9: Hierarquia dos modelos de seguimento PGM. Imagem retirada de [7].

Sabendo que as imagens recolhidas são muitas vezes perturbadas por ruído e que por vezes os movimentos dos objetos apresentam alguma aleatoriedade, estes modelos, recorrendo ao espaço de estados, fazem uma estimação do movimento com base em observações anteriores. Partindo da posição do objeto, obtida por um método de deteção, é estimada a velocidade e aceleração de forma a prever a posição do objeto no *frame* seguinte. Métodos probabilísticos de seguimento têm como objetivo estimar o estado atual do objeto, partindo do conhecimento do estado anterior. A solução ótima é dada por um filtro Bayesiano recursivo. Este filtro é composto por duas fases,

predição e correção. Através de um equação dinâmica é calculado o estado atual, partindo do estado anterior. O estado é depois corrigido usando a informação da medida atual. Caso apenas se pretenda a previsão para um objeto, os dois passos acima são suficiente. Por outro lado, se forem vários os objetos presentes, é necessário um método de associação dos objetos da cena atual à cena anterior.

Para o seguimento de objetos individualmente, o filtro de *Kalman* e o filtro de partículas encontram-se amplamente referenciados. O filtro de *Kalman* é usado para estimar o estado de um sistema linear, assumindo ruído como sendo Gaussiano. É composto por duas partes, estimação e correção. A fase da estimação é responsável pela previsão do estado seguinte do objeto enquanto que a fase de correção é responsável pela correção e atualização da previsão, recorrendo a observações atuais. O filtro de Kalman é definido por um conjuntos de equações matemáticas que permitem uma implementação computacional que estime o estado de um processo de forma a minimizar o erro associado à fase da estimação. No nosso caso, seguimento de pessoas, o processo diz respeito à dinâmica das pessoas enquanto que as características a estimar serão a posição e a velocidade. O filtro de Kalman é caracterizado como ótimo uma vez que é capaz de incorporar todas as informações disponíveis sobre o processo. Além das variáveis medidas pelo sistema são fornecidas também ao filtro informações relativas ao erro na obtenção das mesmas.

Para lidar com sistemas não-lineares foi desenvolvido o filtro de *Kalman* estendido(EKF), que lineariza a estimativa da média e covariância [56]. O filtro de partículas (PF) [57] apresenta-se como uma alternativa ao EKF, com a vantagem de, para um número suficientemente grande de amostras, a sua estimativa se aproximar da estimativa ótima Bayesiana. É utilizado como previsor em sistemas discretos [58, 59] e, ao contrário do EKF, não assume o ruído como Gaussiano. Yang et al [58] apresentam algumas propostas para a obtenção de melhores resultados no seguimento, utilizando o filtro de partículas. Segundo os autores, foram conseguidas melhorias significativas, apesar do aumento computacional do mesmo. Uma comparação entre o filtro de *Kalman* e o filtro de partículas, aplicados ao seguimento, foi feito em [60]. Os autores chegaram à conclusão que o PF poderá ser mais indicado para o seguimento de múltiplos objetos em situações complexas e que o KF deverá ser usado em situações mais simples, tais como vídeo vigilância em áreas com baixa densidade de pessoas. A figura 2.10 apresenta a sequência de operações para a aplicação do filtro EKF.

Em [2] é usado um filtro de *Kalman* para estimar a posição do objeto. Depois de segmentados os objetos, é efetuado o seguimento usando este tipo de filtro. Este consiste num processo cíclico composto por três fases: previsão da posição, baseada na posição na imagem anterior; procura da melhor correspondência; atualização da representação do objeto. Como descrito anteriormente, os objetos são representados usando uma máscara elíptica. É usado um descritor baseado em cor. Uma vez que nem todos os pontos na máscara pertencem ao objeto, é guardado também um modelo com a probabilidade de cada pixel pertencente à máscara fazer parte do objeto. A previsão é feita usando um filtro de *Kalman*, resultando um valor previsto para a posição e uma matriz de covariância do erro. Estes dados combinados com o conhecimento da máxima velocidade de deslocamento do objeto, e assumindo que dois objetos não podem ocupar o mesmo espaço no

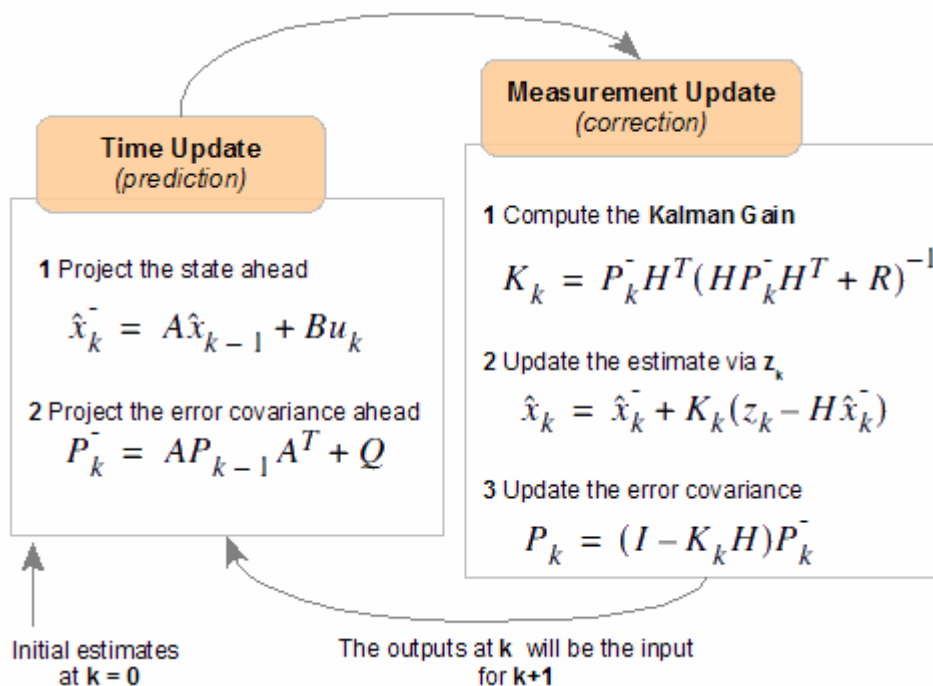


Figura 2.10: Equações do filtro de *Kalman* estendido (EKF) [8].

mesmo momento, é criada uma área de pesquisa onde possivelmente se encontra o objeto. Esta área deve ser o mais pequena possível, de forma a minimizar o esforço computacional bem como as falsas deteções. Depois de obtida uma correspondência positiva o modelo do objeto e o modelo de probabilidades são atualizados.

O algoritmo *Mean shift* [61] é um método estatístico que procura, numa distribuição probabilística, o seu valor máximo. O método cria uma janela de pesquisa que é posicionado numa secção da distribuição. Dentro da janela o valor máximo pode ser calculado fazendo uma simples média. De seguida a janela é movida para o valor máximo e o processo repete-se até a sua convergência para um ponto. Para a aplicação do método no seguimento é necessário representar as imagens de vídeo como uma distribuição probabilística. Cada pixel na imagem é descrito por uma função de probabilidade, dependendo da sua cor. O valor da função é a probabilidade do pixel correspondente pertencer ao objeto a ser seguido. Representando cada *frame* como uma distribuição de probabilidades é possível a aplicação do método *mean shift*. Em [62] é feita uma comparação entre alguns métodos de seguimento baseados em *mean shift*, nomeadamente entre os métodos propostos por Bradski [63], Allen et al [9] e Cominiu et al [61]. Segundo Artner, o método proposto por Bradski, *CAMShift*, apresenta-se como o mais simples dos métodos analisados. Apresenta bons resultados se a cor do objeto é significativamente diferente da cor do fundo mas falha em situações de variações na iluminação ou na aparência do objeto. Estas limitações não se verificam no método apresentado por Allen. O objeto a ser seguido é modelizado usando um histograma de cor, sendo que cada pixel do objeto tem um peso diferente no histograma, de acordo com a sua distância ao centro do objeto. O método falha quando se pretende, por exemplo, efetuar o seguimento de uma face quando várias estão presentes na imagem. Segundo Artner, o

método proposto por Cominiu et al apresenta-se como o mais robusto mas também com um mais elevado peso computacional [62].

No que diz respeito ao seguimento de vários objetos em simultâneo, os métodos JPDAF (*Joint Probabilistic Data Association Filter*) e MHT (*Multiple Hypothesis Tracking*) são os mais usados [7]. O método JPDAF efetua a associação de um número fixo de objetos entre duas imagens, pelo que a entrada ou saída de objetos provoca erros de associação. Este método faz uma estimação conjunta do estado de todos os objetos. Tem como inconveniente o fato de se tornar muito pesado a nível computacional com o aumento do número de objetos. Por sua vez o MHT é capaz de lidar com entradas e saídas de objetos da cena, bem como situações de ocultação [64]. Este método calcula e guarda todas as hipóteses de associação, pelo que o seu peso computacional é bastante elevado. Para ultrapassar este problema foi usado em [65] um método que limita o número de hipóteses a considerar.



Figura 2.11: Seguimento baseado na cor da camisola. Imagem retirada de [9].

2.4 Métodos de avaliação do desempenho

Para que os objetivos propostos sejam atingidos é necessário estudar e implementar métodos que retirem da cena informação sobre a sua complexidade. É essencial também proceder a uma avaliação do desempenho dos algoritmos de deteção e seguimento. Desta forma, podemos adaptar os passos seguintes da execução do algoritmo. Obtendo informação sobre a complexidade da cena, poderemos escolher o método de segmentação que se espera que tenha melhores resultados, tendo em conta uma análise prévia. Do mesmo modo, tendo uma medida objetiva da qualidade da segmentação e da qualidade do seguimento, pode-se variar o método de deteção e/ou variar parâmetros no algoritmo de seguimento.

Os métodos apresentados encontram-se divididos relativamente ao uso ou não de informação de referência. A informação de referência é informação que se toma como ideal, relativamente quer à segmentação quer à dimensão e posição de cada objeto na cena. Esta informação é de grande utilidade quando se pretende comparar objetivamente os resultados obtidos por diferentes métodos, avaliando assim o seu desempenho.

2.4.1 Com informação de referência

Para avaliação do desempenho, recorrendo a informação de referência, existem várias propostas na literatura. Aqui, apresentaremos apenas as que consideramos mais interessantes no contexto do trabalho a desenvolver. Estas medidas serão importantes para avaliação do desempenho global do algoritmo mas também para validação das medidas sem informação de referência. Desta forma fará sentido apresentar medidas de avaliação da qualidade da segmentação, da qualidade da deteção de pessoas e da qualidade do seguimento.

Avaliação da qualidade da segmentação

Para efeitos de avaliação da segmentação Carvalho et al [18] propõem a aplicação do método de *partition-distance*(PD). Este método mede a semelhança entre a segmentação efetuada pelo método em análise e a segmentação ótima, ou de referência. O método foi ainda estendido pelos autores de forma a avaliar também a qualidade do seguimento, comparando os resultados do seguimento entre *frames* consecutivos e informação de referência.

Polak et al [66] propõem uma medida (OCE) para o erro para avaliação da segmentação, calculada ao nível do objeto. Segundo os autores, esta medida avalia corretamente a qualidade da segmentação em situações de sub e sobre segmentação, mas falha em imagens segmentadas onde foram aplicadas operações pós processamento. Esta medida é indicada quando o objetivo da segmentação é delinear múltiplos objetos de geometria similar mas escalas diferentes.

Em [67] são apresentadas três medidas de avaliação da qualidade da segmentação baseadas em informação de referência. As medidas não avaliam apenas a precisão a nível espacial mas também a consistência temporal da máscara de segmentação. As medidas são pesadas de forma a levarem em conta a relevância visual dos erros de segmentação. A mais valia destas medidas reside na sua simplicidade, quando comparadas com outras para o mesmo efeito. Segundo os autores, estas medidas são ideais para a elaboração de rankings entre métodos de segmentação.

Cavallaro et al [68], apresentam uma métrica para a avaliação baseada no desvio entre a máscara de segmentação e a segmentação ideal. A métrica avalia tanto a precisão espacial como a coerência temporal através de uma comparação direta dos resultados. A métrica é pesada pela importância visual dos erros de segmentação. A métrica foi usada para avaliar a qualidade de algoritmos de deteção de movimento.

Avaliação da qualidade da deteção e seguimento

Bashir et al [19] introduzem as métricas *Tracker Detection Rate*(TRDR) e *False Alarm Rate*(FAR), *Detection Rate*(DR), *Specificity*(SP), *Accuracy*(AC), *Positive Prediction*(PP), *Negative Prediction*(NP), *False Negative Rate*(FNR), *False Positive Rate*(FPR). São métricas baseadas no cálculo, em cada *frame*, dos indicadores de *True Positive*(TP), *True Negative*(TN), *False Positive*(FP), *False Negative*(FN). Também para avaliação da qualidade do seguimento, o autor

apresenta algumas medidas. São sugeridas as medidas a *Object Tracking Error*(OTE) e a *Average Object Tracking Error*(AOTE).

Popoola et al sugerem em [69] as medidas *Object Detection Rate*(OBR), *Average Size Detection Rate*(ASDR), *Label Tracking Detection Rate*(LTDR), *Non-Label Tracking Detection Rate*(NTDR) para avaliação da qualidade do seguimento.

São apresentadas em [70] algumas medidas de avaliação de seguimento. São medidas calculadas comparando a trajetória real com a trajetória dada pelo algoritmo de seguimento, retirando assim informação sobre a qualidade do mesmo.

Em [71] é implementado um algoritmo de detecção e seguimento que retira informação da cena, produzindo um melhor desempenho no seguimento. É criado um modelo da cena baseado na probabilidade de, em cada região do espaço, terem lugar eventos significativos (aparecimento de novos objetos, saída de cena de objetos existentes ou ocultação e reaparecimento após ocultação). A avaliação é feita seguindo o protocolo VACE-CLEAR (*Video Analysis and Content Extraction-CLassification of Events, Activities, and Relationships*). Este protocolo define quatro escalas, MODA (*multiple object detection accuracy*), MODP (*multiple object detection precision*), MOTA (*multiple object tracking accuracy*) e MOTP (*multiple object tracking precision*). Também seguindo o protocolo VACE-CLEAR, foram apresentadas em [72] os resultados do *workshop PETS2010* (*Performance Evaluation of Tracking and Surveillance*). Em [73] foi estudada a eficácia das escalas MOTA e MOTP num algoritmo de seguimento.

Carvalho et al [10] propõem uma abordagem híbrida, combinando várias medidas de desempenho com diversos requisitos em termos de segmentação de referência, ultrapassando assim algumas das limitações das abordagens individuais. É proposta uma abordagem baseada na métrica *partition-distance* capaz de fornecer maior quantidade de informação relativamente às métricas baseadas em *bounding box*, enquanto minimiza a necessidade de segmentação de referência. Na figura 2.12 está representada a combinação usada pelos autores para a criação de *ground truth*.

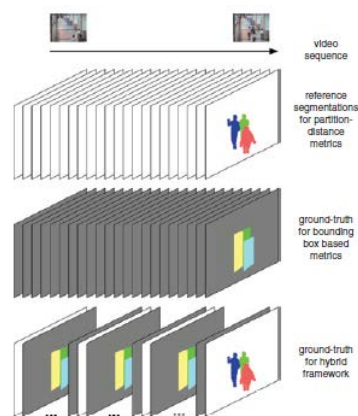


Figura 2.12: *Ground truth* usada na abordagem híbrida. Imagem retirada de [10].

2.4.2 Sem informação de referência

As medidas de avaliação de desempenho sem o recurso a informação de referência são de grande importância neste trabalho. São medidas de avaliação de desempenho conceptualmente diferentes das medidas baseadas em informação de referência. Estas medidas, usando apenas os resultados dos métodos onde são implementadas, tentam avaliar a qualidade desses mesmos métodos. No fundo, trata-se de dotar o algoritmo de seguimento de capacidade de autoavaliação.

Foram encontradas algumas medidas na literatura [74], mas em menor número que as baseadas em informação de referência. Além disso, estas medidas não são medidas absolutas, apenas estimam a qualidade dos resultados. Desta forma, é necessária uma validação das medidas, comparando-as com as medidas baseadas em informação de referência, mais robustas e menos sujeitas a ruído.

Avaliação da complexidade da cena

A complexidade da cena pode ser afetada pelo número de objetos presentes e distância entre eles, número de oclusões ou condições de iluminação. Quanto maior o número de objetos ou o número de oclusões maior será tipicamente o tempo necessário para o processamento. A complexidade excessiva poderá levar a atrasos no processamento, comprometendo o funcionamento em tempo real. Desta forma é desejável adaptar o algoritmo de seguimento de forma a que o processamento se torne mais rápido, mesmo que eventualmente se perca eficácia. O conhecimento da complexidade da cena combinado com as medidas de desempenho permitirá definir métodos de adaptação da complexidade do algoritmo, mantendo uma relação aceitável entre rapidez de processamento e resultados.

Em [11], a deteção de movimento na imagem é feita usando a técnica de *Optical Flow*. Trata-se de uma técnica bastante usada na deteção e seguimento baseado em deteção de movimento. Partindo do princípio que tanto o valor da intensidade dos pixels pertencentes a uma região como o valor do gradiente se mantêm constantes, apesar da variação da sua posição, são criados vetores de deslocamento que definem a translação de cada pixel pertencente a uma região [7]. Na figura 2.13, retirada de [11], é visível a aplicação desta técnica na deteção de movimento entre duas imagens consecutivas.



Figura 2.13: Aplicação do método de *Optical flow* (Imagem retirada de [11]).

Em [12] é feita uma comparação empírica entre as várias formas de cálculo de *optical flow*. Foram testadas as implementações de 9 métodos diferentes de cálculo de *Optical Flow* para diferentes conjuntos de imagens de teste. Segundo os autores, o trabalho desenvolvido mostra uma grande variação de desempenho entre os métodos estudados. A figura 2.14 mostra uma comparação da aplicação de 3 diferentes métodos de *optical flow*, nomeadamente o método proposto por Horn & Schunck [75], Lucas & Kanade [76] e Nagel [77].

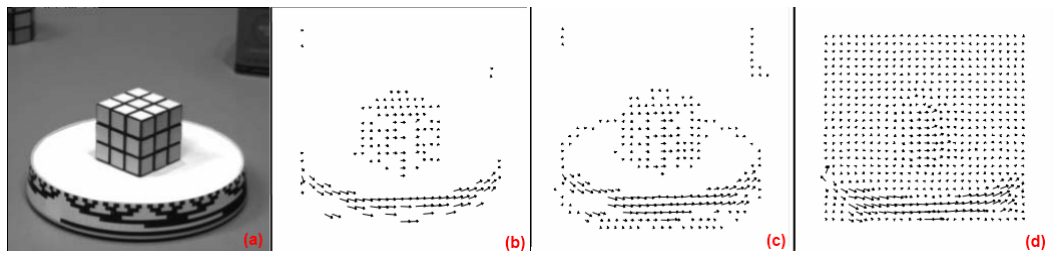


Figura 2.14: Comparação de resultados de diferentes métodos de cálculo do *optical flow* para uma imagem de teste: a) Imagem de teste. b) Método Proposto por Horn & Schunck. c) Método Proposto por Lucas & Kanade. d) Método Proposto por Nagel. Imagens retiradas de [12].

Alguns autores estudaram formas de retirar informação da cena quanto à sua complexidade. Em [78] é feita a implementação de um método de estimação da complexidade da cena de forma a adaptar o algoritmo de seguimento. Wang et al [79] e Nguyen et al [80] fundem nos seus algoritmos de deteção e seguimento informação sobre o contexto onde os objetos estão inseridos. Os autores referem que a inclusão deste tipo de informação reduz a probabilidade de aparecimento de erros no seguimento. A informação relativa ao número de objetos também é considerada importante. Assim, em [72] são apresentados alguns resultados de diferentes métodos de contagem de objetos.

Avaliação da qualidade da segmentação

A avaliação da qualidade da segmentação dos objetos de interesse, e posterior deteção, é importante na medida em que os resultados desta avaliação irão definir o método de segmentação a usar. Desta forma, cada método de segmentação/deteção é avaliado através de um conjunto de medidas que sejam representativas da sua qualidade. Foram selecionadas na bibliografia as medidas descritas de seguida.

Em [13] são apresentadas diversas medidas de avaliação da qualidade da segmentação baseadas na uniformidade da forma, cor e movimento dos objetos segmentados. São medidas de avaliação individual dos objetos e medidas de avaliação global da cena. São apresentadas medidas como a *shape regularity*, a *spatial uniformity*, o *local contrast to neighbors* ou a *criticality*. Segundo os autores, são úteis como indicadores da qualidade da segmentação. Na figura 2.15 é apresentado o sumário da avaliação efetuada em [13]. Segundo os autores, estas medidas estimam a qualidade da segmentação. De realçar que a metodologia proposta também permite a avaliação recorrendo a informação de referência.

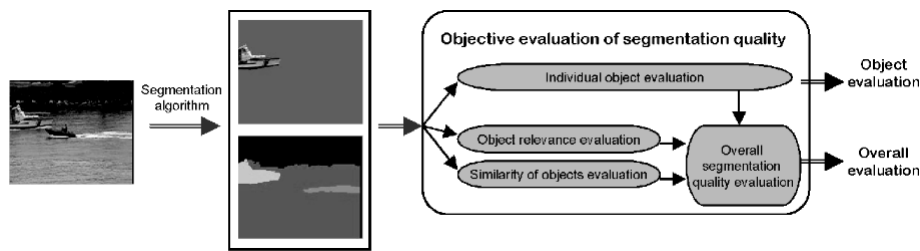


Figura 2.15: Avaliação da segmentação. Imagem retirada de [13] e editada posteriormente.

O *Boundary Spatial Color Contrast* foi proposto em [81] e considera que existe uma diferença de contraste entre o objeto móvel na imagem e o fundo. A medida proposta apresenta semelhanças com a medida *local contrast to neighbors* proposta por Correia [13]. Esta medida (*local contrast to neighbors*) avalia a qualidade da segmentação analisando o contraste na periferia do contorno do objeto. Uma segmentação incorreta poderá corresponder a valores de contraste mais baixos, relativamente a valores obtidos com uma segmentação ideal. São definidas linhas perpendiculares, de comprimento $2L + 1$, em cada ponto do contorno e, nas extremidades das linhas, criada uma janela de dimensão $M \times M$. A média da intensidade dos pixels em cada janela é então comparada no interior e exterior do objeto. Procedendo da mesma forma para todos os pontos do contorno, e calculando a respetiva média, obtemos uma medida que estima a qualidade da segmentação. Na figura 2.16 está exemplificada a aplicação deste método.

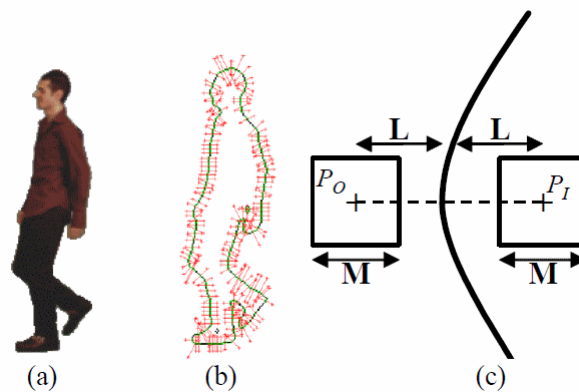


Figura 2.16: Figura que exemplifica o cálculo do contraste no contorno do objeto. a) Objeto segmentado. b) Linhas perpendiculares ao contorno do objeto. c) Detalhe num pixel do contorno. (Imagem retirada de [14])

Avaliação da qualidade da deteção e do seguimento

Para a avaliação da qualidade do seguimento, sem o recurso a informação de referência, é apresentado em [14] uma combinação de medidas relativas à cor, movimento e contornos do objeto. Partindo dos pressupostos que, para um bom seguimento, as fronteiras dos objetos coincidem com as fronteiras da cor, o histograma de cor calculado no interior do objeto é constante entre *frames* e o histograma de cor do fundo é diferente do histograma de cor do objeto, é possível inferir sobre a

qualidade da detecção e seguimento. São ainda assumidas duas condições relativas ao movimento na imagem. São propostas três medidas que, quando combinadas permitem inferir sobre a qualidade da detecção e do seguimento. Assumindo o objeto como móvel, a medida d_{motion} avalia se as fronteiras do objeto coincidem com as fronteiras do movimento. A medida d_{hist} é o resultado da diferença entre o histograma do objeto em cada *frame* da sequência. A medida d_{color} avalia a diferença de intensidades entre os pixels na periferia do contorno do objeto, no interior e no exterior. A medida global apresenta-se como uma combinação das medidas individuais, recorrendo a uma função *fuzzy*. O autor mostra também que a medida apresenta uma boa correlação com um conjunto de três medidas baseadas em informação de referência.

Capítulo 3

Plataforma de testes

Neste capítulo é apresentada e descrita de forma pormenorizada a plataforma usada para efetuar todos os testes necessários. Serão descritos todos os métodos e ferramentas usadas, o modo de funcionamento e a sua função na plataforma.

3.1 Descrição da plataforma

De forma a possibilitar a avaliação dos resultados nas principais fases do processamento, foi definida uma plataforma de testes. Fazem parte da plataforma o algoritmo de seguimento assim como métodos de avaliação e decisão, sendo composta pelo blocos principais apresentados na figura 3.1. Os blocos identificados podem ser divididos em 5 grupos distintos: a) Avaliação (blocos 1, 2 e 3); b) Decisão (blocos 4 e 5); c) Segmentação (blocos 6, 7, 8 e 9); d) Detecção (blocos 10 e 11) e) Seguimento (bloco 12).

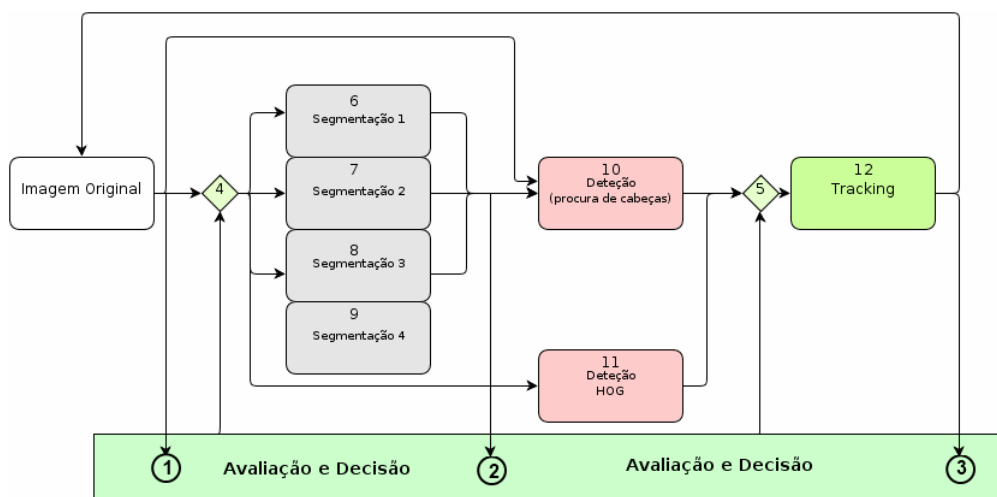


Figura 3.1: Diagrama de blocos da plataforma de testes.

Os blocos de avaliação têm como objetivo recolher informações que sejam úteis sobre um determinado processo. Neste caso, o bloco 1 tem como objetivo retirar informações sobre a imagem

original, que sejam úteis e que possam influenciar o processo seguinte. Corresponde à implementação de uma medida de complexidade da cena que dará indicação da quantidade de movimento presente, usando para o efeito um método de cálculo de *optical flow*. O bloco 2 será responsável pela implementação de medidas que avaliem a qualidade da segmentação. O bloco 3 está localizado no final do ciclo de processamento e pretende avaliar os resultados da detecção e seguimento. Individualmente ou combinadas de forma a serem úteis, estas medidas serão responsáveis pela obtenção de medidas de decisão, a serem usadas nos blocos 5 e 6.

Resumindo, a plataforma de testes é um sistema automático que usa sondas ao longo do processamento com o objetivo de realimentarem o sistema.

Sequência de operações

O processamento terá início com o carregamento das imagens originais. Após o carregamento é avaliada a complexidade da cena (Figura 3.1, bloco 1). Esta avaliação é feita recorrendo a medidas descritas na secção 3.5.

O próximo passo é o carregamento das imagens segmentadas. Estas, foram previamente segmentadas, e estão disponíveis imagens obtidas através de quatro métodos de segmentação distintos. Uma vez que todos requerem um período de treino do fundo, optou-se por segmentar previamente todas as imagens (Figura 3.1, blocos 6,7,8 e 9). Na secção 3.3 serão descritos os métodos usados. A decisão sobre o método a usar é tomada tendo em conta a avaliação relativa à complexidade da cena, mas também a avaliação da qualidade da própria segmentação (Figura 3.1, bloco 2). Assim, se a qualidade da segmentação for aquém do esperado, a próxima imagem a usar será a obtida por um método diferente. Além disso, o resultado da detecção e do seguimento também terá influência nesta escolha (Figura 3.1, bloco 3).

De seguida é feita a detecção de pessoas na imagem e, na plataforma implementada, poderá ser conseguido de duas formas: a) procura de cabeças; b) detetor HOG. Mais uma vez, caso os resultados não sejam os ideais, poderá ser alterado o método de detecção de pessoas. Mais informações sobre os métodos na secção 3.4.

Após a detecção, o seguimento é feito de forma semelhante ao descrito no algoritmo base.

3.2 Algoritmo de seguimento

O algoritmo de seguimento implementado na plataforma (bloco 12 na figura 3.1) é baseado no algoritmo usado por Tao [2], tendo sido estudado no capítulo 2. A figura 3.2 mostra os blocos principais do algoritmo de seguimento usado. Como referido anteriormente, o algoritmo de seguimento incorporado na plataforma preenche os requisitos pedidos (efetua detecção e seguimento de pessoas em ambientes interiores). Além disso, a existência do código fonte e documentação detalhada, tornaram-no na escolha mais indicada. Será de seguida apresentado em detalhe as várias etapas que o constituem.

O algoritmo dá início ao processamento após a receção da sequência de imagens e as respetivas máscaras de segmentação. Depois de carregadas as imagens, o algoritmo inicia o processamento

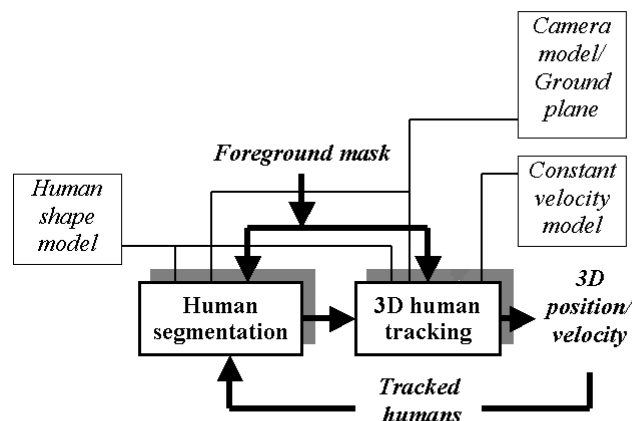


Figura 3.2: Diagrama de blocos do algoritmo de seguimento (Extraída de [2] e editada posteriormente).

com a primeira imagem da sequência, verificando se já existem *tracks* presentes. Uma *track* corresponde a uma pessoa detetada e contém informações sobre a posição, velocidade, modelo de aparência e taxa de ocultação. Cada *track* é identificada com um ID diferente. Na primeira imagem nenhuma *track* estará presente, pelo que o algoritmo ignora a atualização da *track*. Segue-se então a procura, nas máscaras de segmentação, de pessoas presentes na imagem. Esta procura é feita através de um método de procura de cabeças. Uma vez que a cabeça é o ponto mais elevado de uma pessoa (partindo do princípio que a pessoa se encontra de pé) e estará menos sujeita a ocultações, este método é considerado pelo autor como robusto e de implementação simples. Um ponto pode ser considerado como uma possível cabeça se o mesmo é o ponto mais elevado do objeto. Em cada ponto encontrado é criado um modelo da cabeça humana, assumindo dimensões médias. Todos os modelos criados cuja respetiva cabeça não possua um número suficiente de pixels serão imediatamente descartados e o respetivo objeto eliminado da máscara de segmentação. Para todas as cabeças encontradas é calculada a altura do objeto correspondente e, caso se encontre fora de uma gama de valores previamente definidos, o candidato é excluído. Os candidatos restantes serão então considerados como pessoas. Este método estará explicado em maior detalhe na secção 3.4. As pessoas são de seguida processadas uma a uma, desde a mais próxima até à mais afastada da câmara. O processamento é efetuado por ordem, relativamente à distância do objeto à câmara, de forma a tratar corretamente situações de ocultação. Do ponto de vista da câmara, as pessoas que está mais próxima oculta, parcialmente ou totalmente, a pessoa que se encontra atrás (mais afastada da câmara). Após cada processamento, a máscara de segmentação é atualizada de forma a apenas incluir as pessoas não processadas.

A partir do momento em que existe uma *track*, o algoritmo tenta em primeiro lugar efetuar uma correspondência entre a *track* existente e os objetos existentes na nova imagem. Partindo da posição e da velocidade da *track* presente é estimada a posição da mesma *track* na nova imagem

e definida uma janela de pesquisa. A estimação da posição é feita recorrendo a um filtro de *Kalman*. O filtro de *Kalman* é responsável por modelizar de forma linear a posição e velocidade da pessoa, calculando assim a posição mais provável na nova imagem. É então procurada, no interior da janela de pesquisa, uma correspondência para o modelo de aparência da *track* (*matching*). Caso seja feita uma correspondência positiva, a *track* (ID é mantido), os parâmetros do filtro de *Kalman* e o modelo de aparência são atualizados. Caso nenhuma correspondência seja encontrada, o algoritmo marca a *track* existente com inativa e continua o processamento com a procura de cabeças.

3.3 Métodos de segmentação

Os blocos 6,7,8 e 9 da imagem 3.1 correspondem a métodos de segmentação distintos. No fundo correspondem a diferentes modelos de fundo a usar no método de *Background Subtraction*. Foram escolhidos métodos com diferentes complexidades e peso computacional. Assim, foram selecionados do estado da arte os métodos *Running Average*, *Mixture of Gaussians*, coocorrência de cor e *Codebooks*. Uma breve explicação do funcionamento dos mesmos é apresentado de seguida, bem como o resultado da segmentação para uma imagem de teste (figura 5.1).

Running Average

O método *Running Average* (AVG) é o mais simples dos métodos testados. O modelo do fundo é construído à custa da média das imagens que compõem a sequência. À imagem mais recente é dada maior importância neste cálculo. A atualização do modelo de fundo é baseada na fórmula 3.1. O parâmetro α é a velocidade de atualização do modelo de fundo. Assim, um α elevado significa que é dada uma grande importância à imagem atual e, conseqüentemente, mais rapidamente "esquecidas" as anteriores. Um valor elevado de α leva a uma deteção de pequenos movimentos na imagem enquanto que um valor pequeno levará a que esses pequenos movimentos não sejam detetados. Na prática, este valor deve ser ajustado de acordo com os objetivos. Depois de construído o modelo, é subtraída à imagem atual o fundo, resultando uma imagem em que apenas as partes móveis da imagem estarão presentes.

$$fundo(t) = (1 - \alpha) \times fundo(t - 1) + \alpha \times imagem(t) \quad (3.1)$$

Mixture of Gaussians

O método de segmentação de imagem denominado *Mixture of Gaussians* (MOG), é muito referenciado na bibliografia com o objetivo de detetar objetos móveis em imagens [82, 83]. Cada pixel da imagem é modelizado por uma mistura de K gaussianos (5, na implementação usada), onde se assume que, cada gaussiano representa uma cor diferente. Cada pixel da nova imagem é comparado com o respetivo pixel do modelo. Caso seja encontrada uma correspondência, o pixel do modelo é atualizado, caso contrário é adicionado um novo gaussiano ao modelo. Os objetos de

interesse são então extraídos recorrendo ao método de *Background subtraction*.

Coocorrência de cor

O método de deteção e segmentação de objetos baseado em *Coocorrência de cor*, é proposto por *Liyuan Li* em [15], por simplicidade será designado por "FGD" (*Foreground Detetion*). O algoritmo é composto por quatro etapas principais: deteção de movimento, classificação do movimento, segmentação dos objetos de interesse e aprendizagem e atualização do fundo. Na figura 3.3 está representado o diagrama de blocos do algoritmo proposto.

Como a maioria dos métodos de segmentação, este método requer um período de treino para o fundo. Desta forma é criada uma imagem que será constantemente atualizada e que representa o fundo. Na primeira etapa é feita a deteção do movimento usando métodos de subtração de imagens. À imagem atual é subtraída a imagem do fundo, obtendo-se assim os pixels móveis. Nesta fase serão também filtrados os pixels que apresentem movimento considerado insignificante.

A segunda etapa é responsável pela classificação dos pixels resultantes da deteção do movimento. Os pixels são classificados como pertencentes a objetos móveis ou como pertencentes a objetos estáticos na imagem. Para cada pixel estático é gerado um vetor $c_t = [r_t \ g_t \ b_t]^T$, contendo a respetiva intensidade de cor. Para cada pixel móvel é gerado um vetor $cc_t = [r_{t-1} \ g_{t-1} \ b_{t-1} \ r_t \ g_t \ b_t]^T$ com a intensidade de cor na imagem atual e com a intensidade de cor na imagem anterior. Para o fundo é criada uma tabela (tabela de cor) contendo os N_1 valores (c_t) mais comuns para a intensidade de cor dos pixels que o constituem. De forma análoga, para os objetos móveis é criada também uma tabela (tabela de coocorrência de cor) contendo os N_2 valores (cc_t) mais comuns para a intensidade de cor dos pixels que constituem os objetos. A classificação é feita através da comparação entre os vetores obtidos para os pixels móveis e a tabela que os N_2 valores mais comuns para a intensidade dos pixels do fundo.

Segundo os autores, depois da classificação verifica-se que apenas uma pequena percentagem dos pixels são incorretamente classificados e que estes são pontos isolados na imagem. Assim, apenas algumas operações morfológicas simples (abertura e fecho) serão suficientes para remover os pontos incorretamente classificados e tornar compactos os objetos detetados.

A última etapa, atualização do fundo, pode ser dividida em duas fases: atualização das tabelas com os valores mais comuns para a intensidade dos pixels (pixels do fundo, e pixels dos objetos móveis) e a atualização de uma imagem correspondente ao fundo. A atualização das tabelas (cor e coocorrência de cor) é feita de forma a dotar o algoritmo de capacidade para lidar quer com alterações graduais ou pontuais no fundo. A atualização da imagem correspondente ao fundo é essencial de forma a tornar a deteção de movimento (primeira etapa) o mais precisa possível.

Codebooks

O método *Codebooks* (CB) [84, 16] é um método de segmentação baseado em *clustering*. *Clustering* é uma técnica de agrupamento automático de dados segundo um determinado critério de semelhança. O critério de semelhança é definido de acordo com o objetivo pretendido (cor, dimensão, forma...). O método CB cria um modelo de fundo, chamado *codebook* usando um

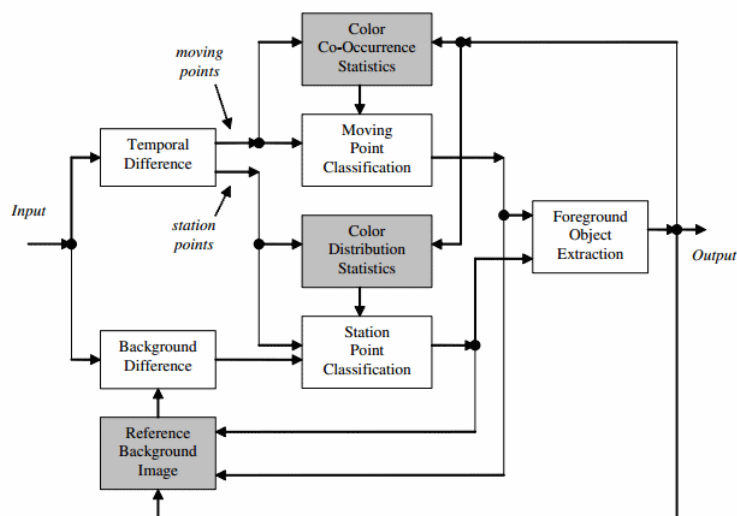


Figura 3.3: Diagrama de blocos do método de segmentação proposto por *Liyuan Li* (Imagem retirada de [15]).

período inicial de treino. Um *codebook* é constituído por um determinado número de *codewords*. Um *codeword* pode ser entendido como um vetor que contém diversas informações relativas ao valor de intensidade dos pixels presentes numa imagem. Um *codeword* descreve uma gama de intensidades da qual fazem parte os pixels que possuam valores dentro dessa mesma gama. Para além do valor da intensidade, o *codeword* contém também informação relativa ao valor máximo e mínimo de brilho, número de vezes que o próprio *codeword* foi observado, o máximo número de *frames* desde a última observação e o número do *frame* relativo à primeira e última observação.

O método CB inicia-se pela criação de um modelo de fundo (*codebook*) vazio, sem qualquer *codeword*. Depois disto, é necessário criar o modelo de fundo, recorrendo a um conjunto de imagens de treino. Os pixels de cada imagem de treino são agrupados em *codewords*, usando como critério de agregação a intensidade dos mesmos. Começando pelo primeiro pixel da primeira imagem de treino, verifica-se que não existe nenhum *codeword*. Assim, é criado um novo com a intensidade deste pixel. Para os pixels seguintes, a intensidade dos mesmos é comparada com o valor de intensidade dos *codewords* existentes. Se a intensidade do pixel está dentro da gama definida para um determinado *codeword* esse pixel é agregado ao *codeword*, sendo o mesmo atualizado. Se a intensidade do pixel estiver fora da gama definida é criado um novo *codeword*. No final do período de treino os *codewords* existentes agregam todos os pixels pertencentes ao fundo. Para melhor perceber este processo, é apresentada a figura 3.4 onde é mostrado o processo de criação de um *codebook* composto por 3 *codewords*.

Depois da criação do modelo de fundo, cada pixel de cada imagem, é classificado como pertencente ao fundo ou pertencente ao objeto móvel caso a intensidade do mesmo esteja ou não presente em algum dos *codewords* existentes.

Segundo Bradski [16], o método codebook produz bons resultados nas mais variadas situações, sendo que tanto o processo de treino como de segmentação de rápida execução, quando

comparado com o método *Mixture of Gaussians* [84]. Contudo, não é capaz de lidar muito bem com variações de iluminação da cena (manhã, tarde ou ligar e desligar de iluminação artificial).

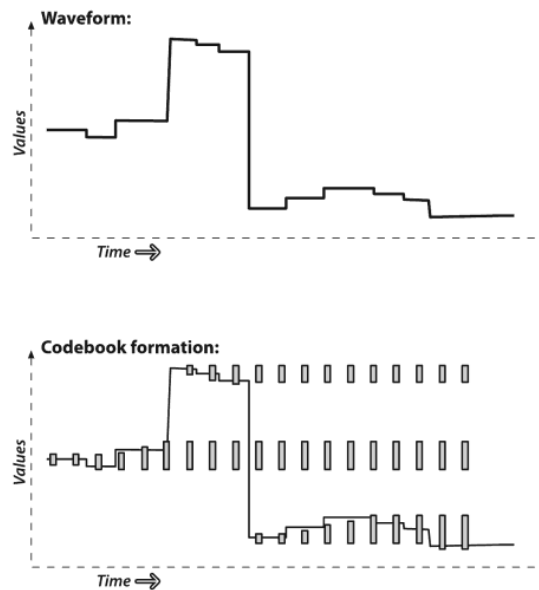


Figura 3.4: Exemplo de criação e atualização de *codewords* (Imagem retirada de [16]).

De referir que ao resultado da segmentação de cada método foi necessário aplicar algum pós-processamento, de forma a tornar mais compactos os objetos encontrados. O pós-processamento consistiu apenas em simples operações morfológicas de "abertura" e "fecho", específicas para imagens binárias.

3.4 Detecção de pessoas

A plataforma de testes apresenta dois métodos de deteção de pessoas distintos. Um método de deteção baseado em segmentação e procura de cabeças e o método HOG, que não se baseia em segmentação. O método de deteção baseado em segmentação, como dito anteriormente, faz parte do algoritmo base. O método HOG foi implementado como alternativa.

3.4.1 Baseada em segmentação

À semelhança de [23], Siebel et al [24] e Tao et al [2], o algoritmo de seguimento faz a deteção de pessoas recorrendo ao método de procura de cabeças. A procura é feita na imagem segmentada e não na imagem original. Como já foi dito anteriormente, a cabeça é o parte do corpo menos sujeito a ocultações e corresponde, na maior parte da situações, ao ponto mais elevado num objeto, nas situações em que a câmara se encontra num ponto elevado. O método faz uma procura (na vertical) na imagem segmentada por pontos que sejam máximos locais. Estes pontos são avaliados juntamente com os seus vizinhos e, caso verifiquem algumas condições, são marcados como possíveis cabeças. As possíveis cabeças são filtrados com base no número de pixels na

vertical. Depois disto, são filtradas de novo recorrendo ao modelo da câmara. O modelo da câmara é responsável pela conversão de áreas e distâncias entre a imagem 2D e o mundo real a 3 dimensões. Depois de calculada a altura média e criado o modelo de forma é criada uma nova *track*, correspondendo à pessoa detetada. Por fim é removido o objeto correspondente na máscara de segmentação, seguindo-se o processo de seguimento. Este processo é repetido até que nenhuma cabeça seja detetada na imagem.

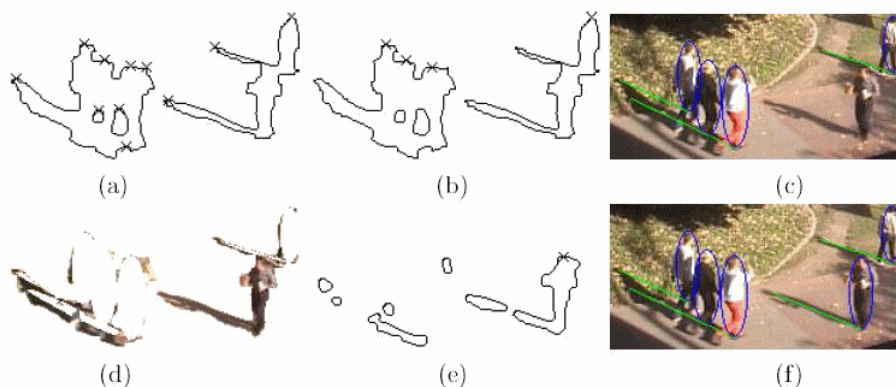


Figura 3.5: Processo de segmentação (Imagem retirada de [2]). a) Possíveis cabeças. b) Cabeças encontradas. c) Pessoas segmentadas representadas por uma elipse. d) Imagem após retirar as pessoas encontradas. e) Máscara de segmentação após retirados os objetos correspondentes às pessoas encontradas (note-se que mais uma pessoa foi detetada). f) Todas as pessoas foram detetadas.

3.4.2 Histograma de gradientes orientados

Como alternativa ao método de deteção baseado em segmentação (a procura de cabeças é feita na imagem segmentada), foi implementado outro método de deteção que não necessita da imagem segmentada. A deteção é feita recorrendo ao método denominado *Histogram of oriented gradients* [17](HOG).

O método HOG foi já estudado no estado da arte. Trata-se de um método muito comum na literatura e, como já foi dito, é um método baseado em SVM (*Support Vector Machine*). Depois de treinado o método com duas sequências de imagens, uma contendo pessoas e outra apenas fundo, o método HOG é capaz de detetar pessoas presentes numa imagem. Tem como inconveniente o facto de necessitar de treino específico para diferentes sequências.

Para treinar o detetor HOG é necessário, em primeiro lugar, compilar dois conjuntos de imagens com a mesma resolução. O primeiro conjunto diz respeito a imagem onde estão presentes pessoas enquanto que o segundo conjunto possui imagens onde apenas é mostrado o fundo. As imagens são então codificadas num *feature space*. *Feature space* é um espaço abstrato onde cada amostra é representada por um ponto num espaço n-dimensional. A dimensão do espaço é dada pelo número de características usadas para descrever os padrões. Amostras semelhantes são agrupadas permitindo a localização de padrões. Este processo é feita quer para as imagens contendo pessoas quer para as que contêm só fundo. Deste modo, o detetor é capaz, para novas imagens,

de fazer a distinção entre pessoas e fundo. O processo de treino está identificado na figura 3.6 enquanto que na figura 3.7 são mostradas as imagens do detetor HOG resultante.

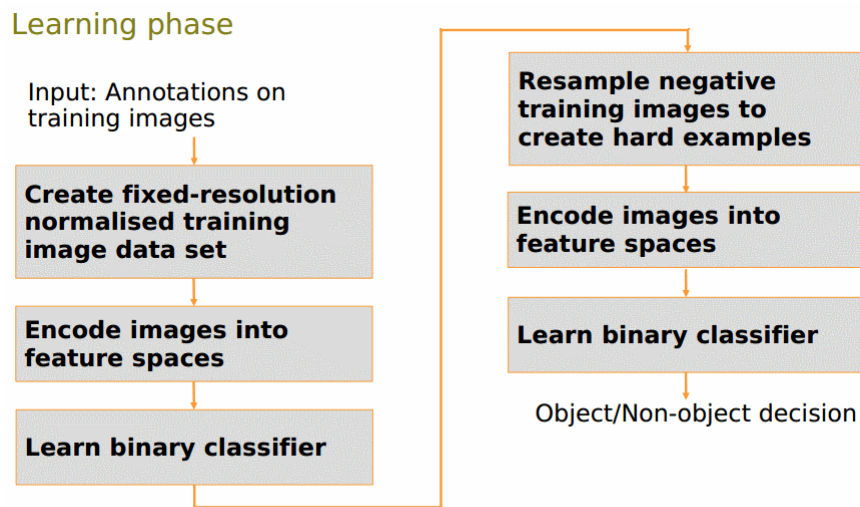


Figura 3.6: Processo de treino do detetor HOG (Imagem retirada de [17]).

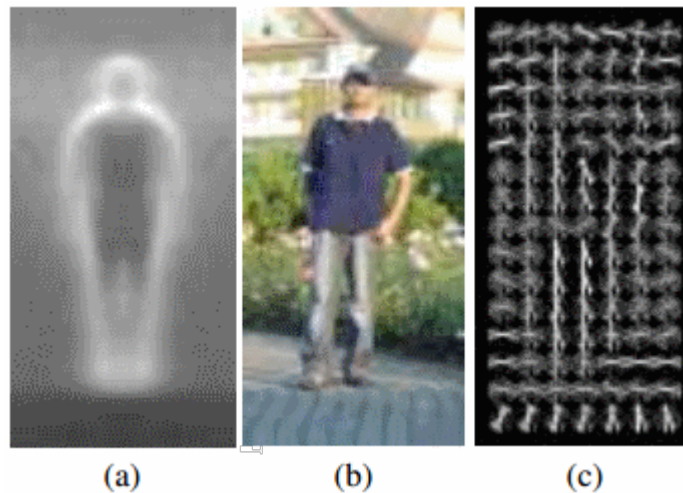


Figura 3.7: a) Imagem da média do gradiente para uma sequência de imagens de treino. b) Imagem original. c) Imagem do descritor HOG.(Imagem retirada de [17]).

A figura 3.8 mostra os resultados da deteção pelo método HOG, numa imagem retirada das sequências de teste apresentadas.

No capítulo 6 será feita a comparação entre os resultados destes dois métodos de deteção, bem como a comparação a nível requisitos computacionais.

O algoritmo de seguimento procede, em primeiro lugar, à atualização das *tracks*, usando a informação relativa à máscara de segmentação, e só depois são procuradas novas pessoas. Uma vez que, usando o detetor HOG, não estão disponíveis máscaras de segmentação, o processo de atualização não é possível. Assim, após a deteção pelo método HOG, foi necessário criar máscaras de

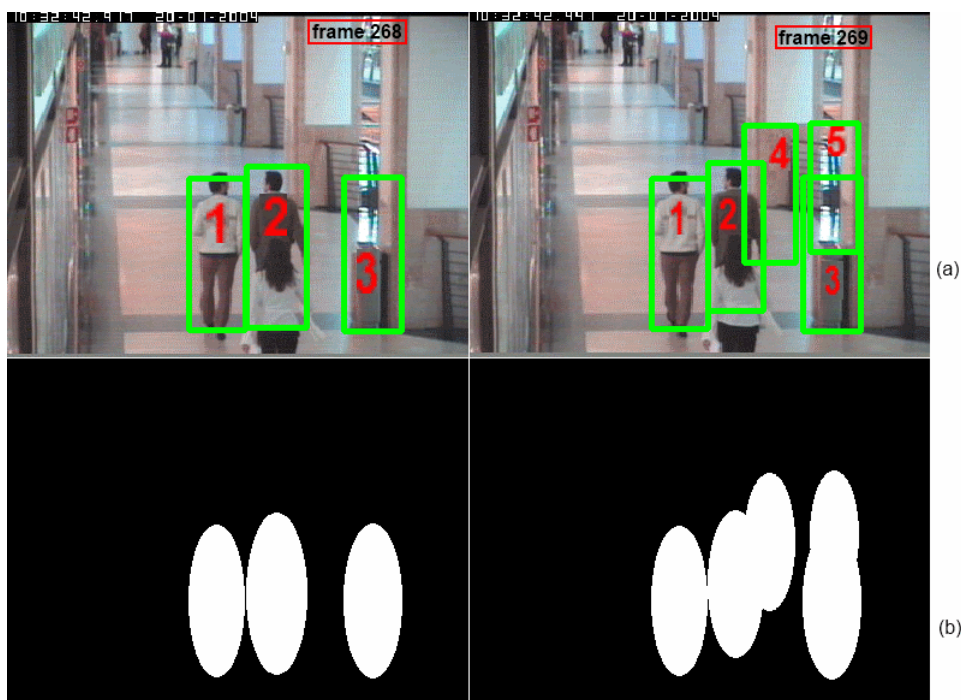


Figura 3.8: Exemplo de detecção de pessoas pelo método HOG e criação das respectivas elipses. a) *frame* 268 e 269 da sequência OSOW1. b) máscaras de segmentação criadas depois da detecção pelo método HOG.

segmentação, contendo informação relativa às pessoas detetadas. Além disto é necessário verificar se as pessoas detetadas em cada *frame* não estão já a ser seguidas (quando utilizado o detetor baseado em segmentação, eram eliminadas da imagem segmentada as máscaras correspondentes aos objetos que iam sendo atualizados, pelo que apenas eram criadas novas *tracks* para as pessoas que ainda não estavam a ser seguidas). Isto foi conseguido comparando a *bounding box* obtida para cada pessoa detetada com a *bounding box* de todas as *tracks* existentes. Se o centroide de cada pessoa se encontrar dentro de alguma das *bounding box* relativas a *tracks* criadas anteriormente, considera-se que a pessoa detetada já possui uma *track atribuída*. Para as pessoas onde isto não aconteça é criada uma nova *track* assim como respetivo modelo de aparência. Caso dois objetos verifiquem estes requisitos, é feita a correspondência com o objeto que se encontre mais próximo. Desta forma, apenas se criam *tracks* novas, se realmente essa pessoa ainda não tiver sido detetada antes. Apesar de não ser a solução ideal, o ideal seria adaptar o algoritmo de seguimento de forma a tratar esta forma de detecção, conseguia-se desta forma ultrapassar este problema. Este procedimento está ilustrado na figura 3.8, e mostra as pessoas detetadas. Para o *frame* 269 apenas serão criadas duas *tracks* para os objetos 4 e 5, uma vez que os objetos 1,2 e 3 obtêm correspondência na imagem anterior. Esta imagem binária será o correspondente à imagem segmentada do detetor de cabeças.

3.5 Medidas de avaliação individuais

As medidas descritas nesta secção são obtidas sem recurso a informação de referência. São apresentadas medidas de avaliação da complexidade da cena, da qualidade da segmentação e da qualidade da deteção e seguimento. As medidas apresentadas apenas usam informação relativa ao método que visam avaliar. A medida de complexidade da cena apenas usa informação obtida na cena assim como as medidas de avaliação da segmentação apenas utilizam informação relativa ao resultado da própria segmentação. Desta forma é possível obter uma estimação da qualidade dos métodos a avaliar. As medidas apresentadas em pormenor de seguida foram escolhidas, em detrimento de outras referidas no estado da arte, tendo em conta principalmente dois fatores: a) avaliam, segundo os autores, a qualidade dos resultados dos quais é necessária avaliação; b) baixo peso computacional na execução da sua implementação.

3.5.1 Complexidade da cena

Optical flow

A quantidade de movimento presente numa cena é um indicador da velocidade das movimentações dos objetos presentes numa imagem, quando as imagens são obtidas usando uma câmara fixa. É um fator importante na classificação da cena quanto à sua complexidade. Uma cena com pouco movimento poderá significar que estão presentes na cena poucos objetos ou que os mesmos se deslocam a velocidade reduzida. Isto poderá ter como consequência um menor tempo de processamento por parte do algoritmo de seguimento, devido à redução do nível de incerteza quanto à posição dos objetos na cena. Uma vez que os objetos serão processados individualmente, um menor número poderá corresponder também a um menor processamento. Assim, usando o método de *optical flow* proposto em [76], é possível obter os vetores de movimento associados aos objetos móveis numa imagem. Desta forma é possível obter informação quanto à velocidade e direção de deslocamento dos objetos da imagem. O método faz uma pesquisa na imagem por pontos ou características locais que sejam interessantes para seguir. De seguida, as características ou pontos localizados anteriormente são procurados na imagem seguinte. Quando encontrados, é criado um vetor que une os dois pontos ou características.

Os vetores de movimento contêm informação relativa à amplitude e direção do movimento. Calculando a soma dos módulos de todos os vetores de movimento, obtemos uma medida quantitativa do nível de movimento presente na cena. Relativamente à direção dos vetores, a informação a retirar não é tão relevante. Uma cena em que vários objetos se desloquem em direções opostas, a soma das direções dos vetores vão ser anuladas, tornando esta informação irrelevante.

O *optical flow* é calculado em cada *frame* antes de qualquer processamento. Como existe uma grande variação do módulo entre *frames* consecutivos, o valor do *optical flow* no respetivo *frame* é a média dos valores obtidos nos últimos 5 *frames*.

Foi usada a implementação em pirâmide do método de Lucas & Kanade [76].

3.5.2 Qualidade da segmentação

Para avaliação da qualidade da segmentação foram escolhidas do estado da arte duas medidas concretas: *shape regularity* e *boundary color contrast*. São medidas que estão bem documentadas e segundo os autores estimam bem a qualidade da segmentação. De seguida apresenta-se o método de cálculo de cada uma delas, assim como a comparação com medidas baseadas em informação de referência, neste caso, *partion distance*. É também calculado o coeficiente de correlação entre as mesmas de forma a melhor avaliar a sua eficácia na estimação da qualidade da segmentação.

shape regularity

A medida *shape regularity*, proposta em [85], usa uma combinação de características dos objetos criados na segmentação, obtendo assim uma medida quanto à sua qualidade. Relações entre área, perímetro e *thickness* são combinados usando as equações 3.3, 3.4, 3.5, 3.6 e 3.7 de modo a obter a equação 3.2, responsável pelo cálculo desta medida. Segundo o autor, a característica *thickness* é o número de erosões que é necessário aplicar ao objeto até que o mesmo desapareça por completo da imagem. Para imagens com vários objetos, é criada uma medida global em que cada medida individual é pesada pela respetiva área. A medida global é por fim normalizada usando a largura da imagem.

$$shape_regularity(E) = 0.5 \times compact(E) + 0.5 \times circ_elong(E) \quad (3.2)$$

$$compact(E) = \max\left(\frac{compactness(E)}{75}, 1\right) \quad (3.3)$$

$$circ_elong(E) = \max\left(circ(E), \max\left(\frac{elong(E)}{5}, 1\right)\right) \quad (3.4)$$

$$compactness(E) = \frac{perimeter^2(E)}{area(E)} \quad (3.5)$$

$$circ(E) = \frac{4 \times \Pi \times area(E)}{perimeter^2(E)} \quad (3.6)$$

$$elong(E) = \frac{area(E)}{(2 \times thickness(E))^2} \quad (3.7)$$

Boundary Spatial Color Contrast

A medida *Boundary Spatial Color Contrast* [14] proposta por Erdem usa a cor e o contorno de forma a avaliar a qualidade da segmentação. É calculada a diferença de intensidade dos pixels na periferia do contorno de forma a verificar se o contorno do objeto coincide com o contorno dado

pela cor. Esta medida é calculada pelas equações 3.8 e 3.9, onde $C_0^i(t)$ e $C_1^i(t)$ são as intensidades dos pixels dentro e fora do contorno. De forma a tornar mais rápido o cálculo desta medida, o contorno do objeto foi aproximado por retas. No ponto central de cada reta foi traçada uma linha perpendicular de comprimento fixo e, nas duas extremidades definida uma janela. É calculada a diferença de intensidade entre os pixels que ficam na janela situado no interior e exterior do contorno. Este procedimento é repetido para todos os K_t segmentos de reta.

$$\delta_{color}(t, i) = \frac{\|C_0^i(t) - C_1^i(t)\|}{\sqrt{3 * 255^2}} \quad (3.8)$$

$$d_{color}(t) = 1 - \frac{1}{K_t} \sum_{i=1}^{K_t} \delta_{color}(t, i) \quad (3.9)$$

3.5.3 Qualidade da detecção e seguimento

Para a avaliação da qualidade da detecção e do seguimento, foi usado o método *ngtMetrics* por Erdem em [14]. A medida *fuzzy* é apresentada como uma combinação de três medidas individuais: d_{motion} (*motion differences*), d_{hist} (*inter-frame color histogram differencing*), d_{color} (*intra-frame color along the boundary*).

Para o cálculo de d_{color} assume-se que as fronteiras do objeto coincidem com as fronteiras de cor. Desta forma, devem existir diferenças entre a cor dentro e fora do objeto. São definidas linhas, de comprimento L perpendiculares ao contorno do objeto e, nas extremidades criadas janelas de dimensão $M \times M$. A média das diferença de cor entre as janelas, para todo o contorno, dá-nos a medida pedida. A medida é normalizada entre 0 e 1, sendo que valores mais elevados correspondem a uma pior segmentação. O cálculo de d_{hist} baseia-se no facto de que, para o mesmo objeto, o seu histograma de cor não deve sofrer alterações significativas de *frame* para *frame*. Desta forma, o facto de ocorrerem alterações grandes no seu histograma de cor pode significar que partes do fundo estarão a ser incluídas no objeto, significando uma segmentação errada. A medida d_{motion} é calculada de forma análoga a d_{color} , exceto que nesta medida não são calculadas diferenças entre a cor mas sim diferenças entre vetores de movimento. A medida d_{motion} visa avaliar se as fronteiras do objeto mantêm o mesmo nível de movimento entre imagens consecutivas. Se parte do fundo for incluída no objeto na nova imagem, não haverá movimento nessa secção do objeto.

Estas três medidas individuais são então combinadas numa única medida global 3.10, onde $\mu(\cdot)$ é a uma função *fuzzy* dada pela equação 3.11. É também sugerido o cálculo da medida global pela média das três medidas individuais. Segundo Erdem, a combinação das medidas usando a equação 3.10 dá uma maior importância a erros maiores. A imagem 3.9 é um exemplo apresentado pelos autores, onde $c_1 = 0, 2$, $c_2 = 0, 5$, $c_3 = 0, 8$.

$$D = \frac{\mu(D_{color})D_{color} + \mu(D_{hist})D_{hist} + \mu(D_{motion})D_{motion}}{\mu(D_{color}) + \mu(D_{hist}) + \mu(D_{motion})} \quad (3.10)$$

$$S(D) = \begin{cases} 0 & \text{if } D \leq c_1 \\ \frac{(D-c_1)^2}{(c_2-c_1)(c_3-c_2)} & \text{if } c_1 \leq D \leq c_2 \\ 1 - \frac{(D-c_3)^2}{(c_3-c_2)(c_3-c_1)} & \text{if } c_2 \leq D \leq c_3 \\ 1 & \text{if } D \geq c_3 \end{cases} \quad (3.11)$$

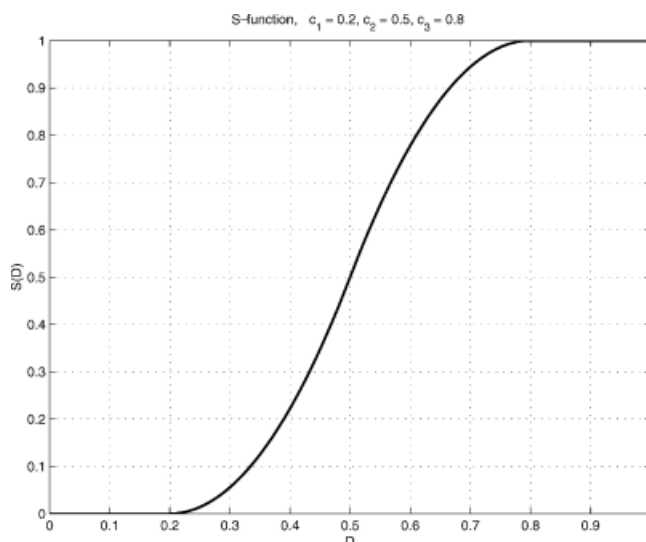


Figura 3.9: Gráfico da função usada para pesar as medidas individuais de modo a obter a medida global. (Imagem retirada de [14]).

3.6 Medidas de avaliação baseadas em informação de referência

Partition-Distance

O método *partition-distance*(PD) [18] é um método de cálculo de medidas de avaliação, baseado em informação de referência. É usada para avaliar a qualidade da segmentação, comparando-a com a segmentação ideal. Este método calcula várias medidas diferentes, das quais destacamos a *symetric particion-distance* (SPD).

Dadas duas partições, a medida *symetric partition-distance* apresenta-se como o número mínimo de elementos que devem ser eliminados da primeira partição de forma a que as duas sejam idênticas. Segundo a medida descrita, e tomando como exemplo duas partições de dimensão 8x8 apresentadas na figura 5.2, as duas partições encontram-se a 10 pixels de distância entre elas. A imagem central pode ser considerada como uma máscara do erro de segmentação, possibilitando também uma melhor avaliação do método de segmentação usado. De forma a melhor avaliar a segmentação, cada pixel da máscara do erro de segmentação é pesado pela respetiva relevância visual. A relevância visual é dada pela distância do pixel ao contorno do objeto. Desta forma, as discrepâncias ao longo do contorno do objeto serão menos penalizadas do que discrepâncias no seu interior.

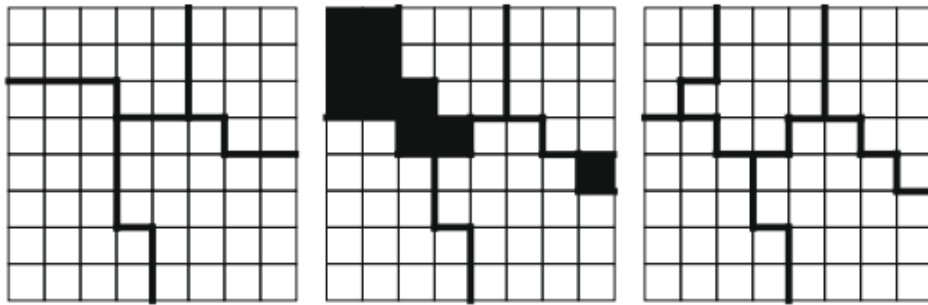


Figura 3.10: Método de *symetric partition-distance*. Na imagem da esquerda e da direita estão representadas duas partições diferentes da mesma imagem. Na imagem central estão selecionados os pontos onde as duas partições não coincidem. (Imagem retirada de [18]).

A medida *symetric partition-distance*, além de usada para avaliar a qualidade da segmentação, será também usada de forma a avaliar a qualidade do seguimento. Para este efeito, usando os resultados do seguimento, é criada uma imagem contendo as *bounding-box* dos objetos a serem seguidos, em dois *frames* consecutivos. De forma a ser possível fazer uma comparação, é criada, da mesma forma, uma imagem relativa à informação de referência. As duas imagens obtidas são depois comparadas e avaliada a sua discrepância, de forma similar à usada na avaliação da segmentação. A figura 3.11 mostra um exemplo das imagens a usar no cálculo desta medida. A imagem da esquerda diz respeito aos resultados do seguimento enquanto que a imagem da direita diz respeito à informação de referência. Tratando-se de uma medida de erro, consideram-se bons resultados valores pequenos de *symetric partition-distance*.

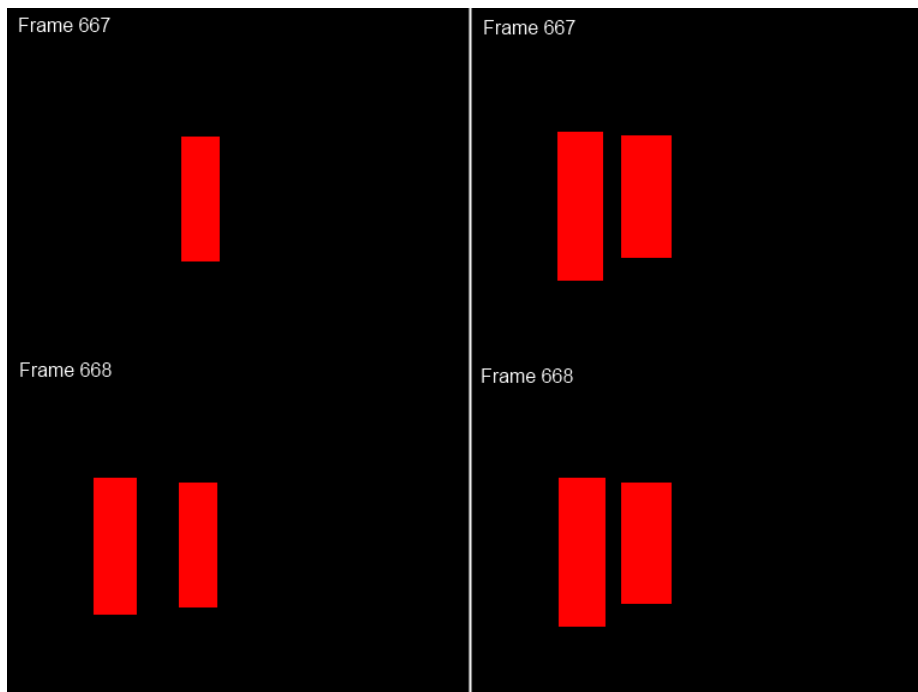


Figura 3.11: Imagens temporárias criadas para aplicação da medida *symetric particion-distance* ao seguimento.

Medidas baseadas em Bounding-Box

As medidas baseadas em *Bounding-Box* são medidas de avaliação baseadas na comparação dos resultados obtidos com os resultados considerados ideais. Serão avaliados separadamente os resultados da detecção e do seguimento, usando para o efeito as medidas apresentadas em seguida.

São usadas as medidas propostas por Bashir et al [19]. Estas medidas estão divididas em dois grupos: medidas calculadas por *frame* (*Frame-based Metrics*) e medidas calculadas por objeto (*Object-based Metrics*). As medidas calculadas por *frame* serão usadas para avaliar a qualidade da detecção enquanto que as medidas calculadas por objeto serão usadas para avaliar a qualidade do seguimento.

Medidas por *frame*

As medidas calculadas por *frame* são calculadas em cada *frame* individualmente, sendo portanto indicadoras da qualidade da detecção de objetos. Para o cálculo desta medidas é necessário obter em primeiro lugar quatro medidas básicas: TP (*True Positive*), TN (*True Negative*), FP (*False Positive*) e FN (*False Negative*). Para todos os *frames* da sequência, são calculadas estas quantidades (Figura 3.12).

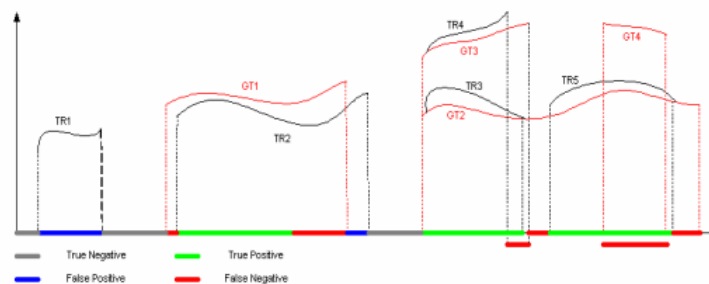


Figura 3.12: Método de cálculo de TP,TN,FP,FN (Imagem retirada de [19]).

- *True Positive*(TP): Número de *frames* onde tanto o sistema como a *Ground Truth* concordam quanto à presença de um ou mais objetos, e a *bounding box* de pelo menos um objeto coincide com a *bounding box* do *Ground Truth*.
- *True Negative*(TN): Número de *frames* onde tanto o sistema como a *Ground Truth* concordam quanto à ausência de qualquer objeto.
- *False Negative*(FN): Número de *frames* onde a *Ground Truth* indica a presença de pelo menos um objeto, e o sistema ou não apresenta qualquer objeto ou a *bounding box* do objeto detetado não coincide com a *bounding box* do objeto indicado na *Ground Truth*.
- *False Positive*(FP): Número de *frames* onde o sistema indica a presença de pelo menos um objeto, e a *Ground Truth* ou não apresenta qualquer objeto ou a *bounding box* do objeto detetado não coincide com a *bounding box* do objeto indicado na *Ground Truth*.

- *Total Ground Truth Frames (TG)*: Número de *frames* contendo objetos *Ground Truth*.
- *Total Frames (TF)*: Número de *frames* da sequência.

Tracker Detection Rate

$$TRDR = \frac{TP}{TG} \quad (3.12)$$

Recall

$$R = \frac{TP}{TP + FN} \quad (3.13)$$

Specificity

$$SP = \frac{TN}{FP + TN} \quad (3.14)$$

Accuracy

$$AC = \frac{TP + TN}{TF} \quad (3.15)$$

Precision

$$P = \frac{TP}{TP + FP} \quad (3.16)$$

Negative Prediction

$$NP = \frac{TN}{FN + TN} \quad (3.17)$$

False Positive Rate

$$FPR = \frac{FP}{FP + TN} \quad (3.18)$$

False Negative Rate

$$FNR = \frac{FN}{TP + FN} \quad (3.19)$$

False Alarm Rate

$$FAR = \frac{FP}{TP + FP} \quad (3.20)$$

Os resultados obtidos recorrendo a estas medidas variam entre 0 e 1, sendo que uma boa deteção produz valores reduzidos de *False Alarm Rate*, *False Negative Rate* e *False Positive Rate* e valores elevados nas restantes medidas.

Medidas por objeto

As medidas *Object-based Metrics* são calculadas com base nas trajetórias completas dos objetos entre o *Ground Truth* e o sistema. Uma vez que uma *track* dada pelo *Ground Truth* pode corresponder a várias *tracks* do sistema, é necessário criar em primeiro lugar uma mapa com as respetivas correspondências. Após este mapeamento, são calculadas a TP, TN, P e FN que servem de base ao cálculo das medidas seguintes:

Tracker Detection Rate

$$TRDR = \frac{TP}{TG} \quad (3.21)$$

Recall

$$R = \frac{TP}{TP + FN} \quad (3.22)$$

Precision

$$P = \frac{TP}{TP + FP} \quad (3.23)$$

F-Score

$$FS = 2 \cdot \frac{P \cdot R}{P + R} \quad (3.24)$$

Tracker Success Rate

$$TSR = \frac{\text{tracks nao fragmentadas}^1}{\text{total de tracks}} \quad (3.25)$$

False Alarm Rate

$$FAR = \frac{FP}{TP + FP} \quad (3.26)$$

False Negative Rate

$$FNR = \frac{FN}{FN + TP} \quad (3.27)$$

Fragmentation Error Rate

$$FER = \frac{\text{nmero de fragmentos}^1}{\text{nmero total de tracks}} \quad (3.28)$$

Avg. Object Tracking Error

$$OTE = \frac{1}{N_{obj}} \times \left(\frac{1}{N_{rg}} \sum_{i \in g(t_i) \wedge r(t_i)} \sqrt{(x_i^g - x_i^r)^2 + (y_i^g - y_i^r)^2} \right) \quad (3.29)$$

Destas medidas realçamos o *Precision* (P), *Recall* (R) e *Avg. Obj. Tracking Error* (AOTE). A medida de R representa a taxa de detecção enquanto que a medida P representa a taxa entre os objetos detetados e os objetos que realmente interessam. O AOTE é uma medida da discrepância entre as *Bounding Box* dadas pelo *Ground Truth* e pelo sistema. A medida *F-Score* é a média harmónica das medidas P e R. Os resultados obtidos recorrendo a estas medidas variam entre 0 e 1, exceção feita às medidas AOTE e FER. Bons resultados no seguimento produzirão valores baixos nestas duas medidas assim como nas medidas FAR e FNR. Para as restantes medidas, valores próximos de 1 correspondem a bons resultados.

As medidas apresentadas (baseadas em *bounding-box* e *symetric partition-distance*) estimam, segundo os autores, estimarem corretamente a qualidade do seguimento, fato que levou a que as mesmas fossem escolhidas.

¹Entende-se por *track* fragmentada um *track* da informação de referência que corresponda a várias *tracks* no sistema.

Capítulo 4

Metodologia para avaliação de desempenho

Neste capítulo, além de apresentadas as sequências de imagens a usar, será descrita a metodologia a adotar para a avaliação do desempenho dos vários métodos que compõem a plataforma de testes. Os resultados das medidas apresentadas anteriormente (*Optical Flow*, *Shape Regularity*, *Zonal Contrast*, d_{color} , d_{motion} e d_{hist}) serão comparados com resultados obtidos através de medidas baseadas em informação de referência, procurando assim uma validação das mesmas.

4.1 Descrição das sequências de teste

Como forma de avaliar o desempenho dos métodos propostos, são usados dois *datasets*: CAVIAR [86] e PETS2006 [87]. Estes dois *datasets* são amplamente usados pelos autores estudados, pelo que é possível efetuar uma comparação entre os resultados obtidos. Cada um destes *datasets* disponibiliza diversas sequências, das quais algumas foram selecionadas as que se indicam de seguida. A seleção foi feita de modo a obtermos sequências de complexidades diferentes, tanto em relação ao número de pessoas simultaneamente na cena como em relação às suas movimentações. Além disso, a existência de alguma informação de referência (relativamente à segmentação ideal e aos resultados do seguimento) possibilita uma avaliação mais fiável. Estes fatores foram determinantes na sua escolha.

Caviar:

Este *dataset* disponibiliza diversas sequências distintas, das quais serão usadas duas: OneShopOneWait1 e OneShopOneWait2. Estão disponíveis em formato vídeo MPEG e imagens em formato JPEG de resolução PAL *standard* (384 x 288 pixels, 25 frames por segundo).

- OneShopOneWait1 (OSOW1)
Composta por 1377 imagens ou 55 segundos de vídeo MPEG. São fornecidas imagens uniformemente espaçadas no tempo relativas à segmentação ideal e informação de referência relativa à posição e respetivas dimensões de todos os objetos ao longo de toda a sequência.
- OneShopOneWait2 (OSOW2)
Composta por 1462 imagens ou 58 segundos de vídeo MPEG. À semelhança da sequência OSOW1, são fornecidas imagens uniformemente espaçadas no tempo relativas à segmentação ideal e informação de referência relativa à posição e respetivas dimensões de todas as pessoas ao longo de toda a sequência.

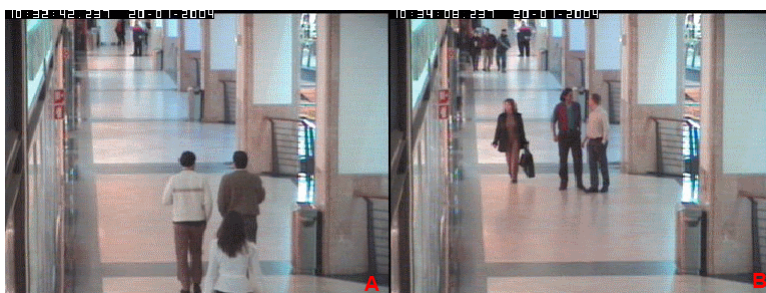


Figura 4.1: Imagens ilustrativas das sequências (frame 250): A) sequência OSOW1. B) sequência OSOW2.

PETS2006:

Do *dataset* PETS2006 foi selecionada a sequência *S1 (Take 1-C) - cam3*. Esta sequência é composta por 3020 imagens gravadas a 25 *frames* por segundo, comprimidas em JPEG e resolução PAL *standard* (768 x 576). É disponibilizada informação de referência relativa à segmentação em determinados *frames* da sequência. Relativamente ao seguimento, não é apresentada qualquer informação sobre os objetos (posição e dimensões).

Estas sequências possuem complexidades diferentes, principalmente no que diz respeito ao número de pessoas presentes na cena e à quantidade de movimento. Relativamente às sequências do *dataset* Caviar, as duas sequências escolhidas diferem bastante no número de pessoas em simultâneo na cena. Enquanto que para a sequência OSOW1 o número de pessoas varia entre 0 e 5, na sequência OSOW2 o número de pessoas varia entre 4 e 8. A sequência escolhida do *dataset* PETS2006 difere das sequências escolhidas do *dataset* Caviar principalmente na direção do movimento. As movimentações das pessoas presentes na cena dão-se maioritariamente numa direção perpendicular ao eixo da câmara, enquanto que para as sequências Caviar esse movimento é numa direção paralela ao eixo da câmara. Estas informações foram retiradas da informação de referência fornecida para cada uma das sequências.

As diferentes complexidades apresentadas pelas sequências escolhidas permitirão uma análise mais correta aos resultados obtidos.



Figura 4.2: Imagem ilustrativa da sequência PETS2006. (Frame 1050)

Descrição do sistema de testes

Todos os testes foram realizados numa máquina com processador Intel Pentium Dual T2310 @1.46GHZ, 2GB de memória RAM em ambiente Windows 7 Ultimate (32 bits).

4.2 Framework de avaliação

4.2.1 Validação das medidas

Nesta secção será feita a comparação dos resultados obtidos pelas diversas medidas (que não se baseiam em informação de referência) introduzidas na plataforma de testes com os resultados de medidas baseadas em informação de referência, com o objetivo de validar as mesmas. A validação das medidas é essencial de modo a perceber qual o grau de aproximação, relativamente às medidas baseadas em informação de referência. Uma vez que as medidas baseadas em informação de referência são menos sujeitas a ruído e a erros, é importante obter uma comparação entre os resultados obtidos por estas e os resultados obtidos através de medidas sem informação de referência. Ao conseguirmos uma elevada correlação entre elas poderemos concluir que temos uma medida com uma boa aproximação da medida com informação de referência.

Medidas de avaliação da complexidade da cena

Para a avaliação da complexidade da cena, será usada, como referido no capítulo anterior, a medida descrita como o valor absoluto da soma do módulo dos vetores de *optical flow* calculados em cada imagem. A medida será designada por módulo do *optical flow*, (OFM) *Optical Flow*

Modulus. O intervalo de tempo necessário ao cálculo destes valores será medido durante o processamento. Contudo a tabela 4.1 mostra o tempo de cálculo médio, mínimo e máximo para as seqüências de teste a usar. De notar que o tempo de cálculo é praticamente uniforme para as seqüências OSOW1 e OSOW2 sendo claramente superior na seqüência PETS. Estas diferenças devem-se à resolução das imagens. As imagens da seqüência PETS possuem uma resolução de 720x576 enquanto as imagens das restantes seqüências têm resolução 384x288. A quantidade de movimento presente tem uma pequena influência no tempo de cálculo, mas devido ao facto de o número de vetores ser limitado, essa influência é fortemente atenuada.

De forma a validar a medida módulo do *optical flow* como estimadora da quantidade de movimento na imagem, os valores obtidos foram comparados com informação de referência. A informação disponível permitiu retirar dados sobre o número de pessoas em cada frame assim como a amplitude das respetivas movimentações (*GT Motion*). A tabela 4.2 apresenta as respetivas correlações entre o módulo *optical flow* e as informações de referência. Para uma melhor análise, é apresentada na figura 4.3 a evolução das três medidas ao longo da seqüência OSOW2.

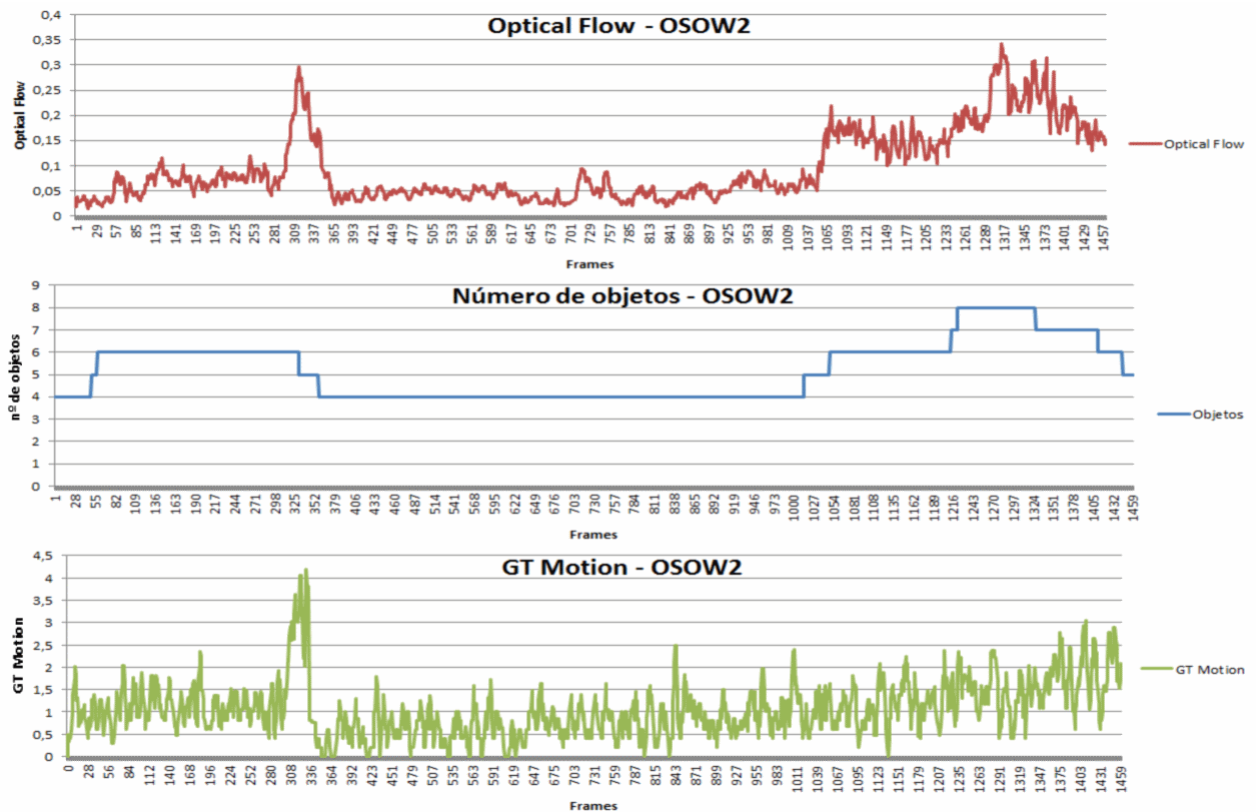


Figura 4.3: Módulo do *optical flow* ao longo da seqüência OSOW2(cima). Número de objetos ao longo da seqüência, obtido pela informação de referência (meio). Amplitude das movimentações(*GT Motion*) ao longo da seqüência, obtido pela informação de referência (fundo).

Tabela 4.1: Tempo de cálculo médio, máximo e mínimo (por *frame*) do módulo do *optical flow* para cada sequência de teste.

	OSOW1	OSOW2	PETS
Tempo de cálculo médio (s)	0,096	0,104	0,185
Tempo de cálculo máximo (s)	0,125	0,141	0,212
Tempo de cálculo mínimo (s)	0,078	0,076	0,167

Tabela 4.2: Coeficiente de correlação entre o número de objetos e o valor do módulo do *optical flow*, e distância percorrida pelos mesmos e o módulo do *optical flow*.

	OSOW1	OSOW2	PETS
Número de Objetos	0,27	0,79	N/A ²
Distancia percorrida	0,45	0,52	N/A ²

As duas medidas retiradas da informação de referência possuem algumas limitações. O número de objetos pode não ser um bom indicador da quantidade de movimento, uma vez que numa cena com vários objetos parados iremos obter um valor de módulo do *optical flow* baixo. A informação relativa ao movimento dos objetos na cena apresenta problemas quando os objetos se encontram parcialmente ocultos. Os objetos movem-se, mas essa informação não é mostrada no cálculo do valor do módulo do *optical flow*.

Através dos resultados obtidos pelo cálculo do módulo do *optical flow* e os resultados obtidos pelas medidas de informação de referência (número de objetos por *frame* e distância percorrida por *frame*) foi calculado o coeficiente que relaciona os valores obtidos, estando apresentados na tabela 4.2. Como se percebe pela análise da tabela os valores obtidos não mostram uma boa correlação quer entre o módulo do *optical flow* e o número de objetos quer entre o módulo do *optical flow* e a distância percorrida pelos objetos, confirmando as limitações identificadas. Relativamente à sequência PETS2006, uma vez que não foi fornecida informação de referência relativa ao seguimento, não foi possível o cálculo do coeficiente de correlação entre as medidas apresentadas.

Medidas de avaliação da qualidade da segmentação

Para validação desta medida, os resultados foram comparados com os obtidos através da medida *symetric partition-distance*, calculada pelo método *partition-distance* proposta por Carvalho et al em [18]. Se for encontrada uma correlação forte entre as métricas, significará que a medida avaliada será uma boa aproximação da qualidade da segmentação. Foram traçados gráficos com a evolução das medidas *shape regularity* e *partition-distance* e obtido o valor do coeficiente de correlação (Figuras 4.4, 4.5). A tabela 4.3 apresenta os valores para a correlação entre as duas métricas, para cada sequência testada e para os diversos métodos de segmentação implementados.

²N/A - Não disponível. Informação de referência relativa ao seguimento não fornecida para a sequência PETS.

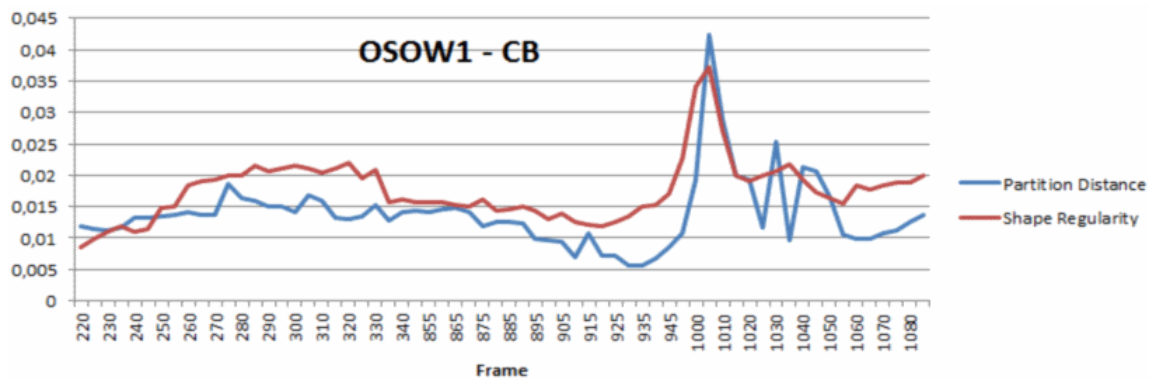


Figura 4.4: Avaliação do método de segmentação *codebook*, para a sequência OSOW1, recorrendo às medidas *shape regularity* e *partition-distance*.

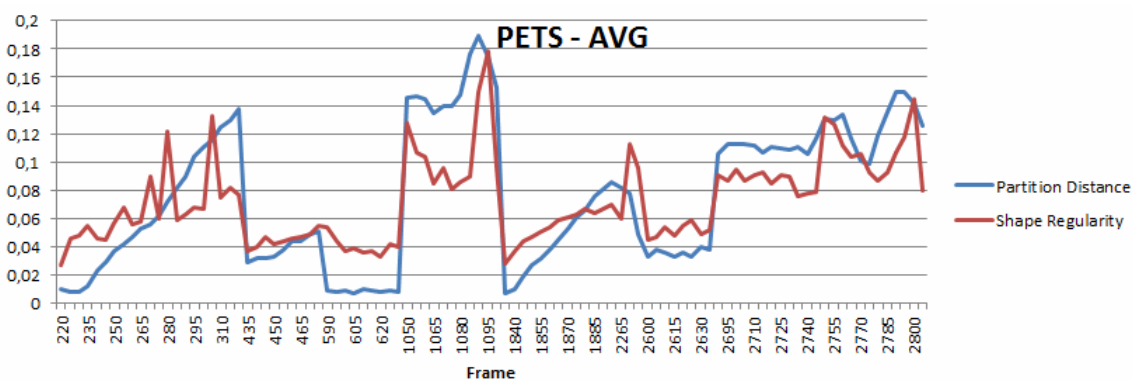


Figura 4.5: Avaliação do método de segmentação AVG, para a sequência PETS2006, recorrendo às medidas *shape regularity* e *partition-distance*.

Tabela 4.3: Coeficientes de correlação entre a medida *shape regularity* e a medida *partition-distance* para as diversas sequências.

	CB	AVG	FGD	MOG
OSOW1	0,74	0,64	0,58	0,59
OSOW2	0,92	0,73	0,56	0,74
PETS	0,82	0,84	0,52	0,81

Tabela 4.4: Coeficientes de correlação entre a medida *color contrast* e a medida *partition-distance* para as diversas sequências.

	AVG	MOG	FGD	CB
OSOW1	0,65	0,40	0,13	0,32
OSOW2	0,61	0,09	0,38	-0,13
PETS	0,28	0,21	-0,15	0,15

Relativamente ao intervalo de tempo necessário para o cálculos das medidas *shape-regularity* e *color contrast* (as duas medidas são calculadas simultaneamente), verificou-se que, para as sequências testadas (OSOW1 e OSOW2), nunca ultrapassou os 0,015 segundos.

Analisando os valores de correlação apresentados na tabela 4.3 verifica-se que existem valores interessantes de correlação, especialmente para os métodos de segmentação *codebooks* (CB) e *mixtures of gaussians* (MOG). Para o método *codebook*, verifica-se mesmo que os valores mínimos de correlação são superiores a 0,74. Desta forma, podemos considerar esta medida como uma razoável estimativa da qualidade da segmentação.

Relativamente à medida *color contrast* será necessário um ajuste cuidadoso de todos os parâmetros, uma vez que a mesma não apresenta uma grande correlação com o *partiton-distance*, como se comprova pela análise da tabela 4.4, onde são apresentados os valores da correlação para esta medida.

Medidas de avaliação da qualidade da deteção e do seguimento

Para avaliação da qualidade da deteção e do seguimento, foram implementadas as medidas *ngtMetrics*, propostas por Erdem [14]. Trata-se de um conjunto de três medidas individuais que, quando combinadas, formam a medida global de avaliação da qualidade. A medida global toma o nome de *fuzzy*. Trata-se de uma medida de erro, pelo que melhores resultados ao nível do seguimento corresponderão a menores valores nesta medida.

Para validação desta medida será calculada a correlação entre os valores obtidos com os valores obtidos pela medida baseada em informação de referência *symetric partition-distance*. O modo de cálculo desta medida é semelhante ao modo de cálculo da medida com o mesmo nome para avaliação da qualidade da segmentação. Contudo, diferem nas imagens a ser usadas para o respetivo cálculo. Quando se pretende uma avaliação da qualidade da segmentação, o cálculo da medida é feito pela comparação da imagem segmentada com a segmentação ideal. Quando o pretendido é a avaliação da qualidade do seguimento, são criadas imagens temporárias contendo duas imagens consecutivas, quer para os resultados do seguimento quer para a informação de referência. Este processo está exemplificado na secção 3.6, figura 3.11. A figura 4.6 mostra a evolução dos valores obtidos pela medida 4.6, quando são fornecidas ao algoritmo de seguimento as imagens segmentadas através de cada um dos métodos de segmentação. O método de segmentação FGD não foi incluído nesta análise uma vez que não apresentou qualquer resultado ao nível do seguimento, para as sequências OSOW1 e OSOW2, devido, principalmente, à segmentação muito ruidosa que produz.

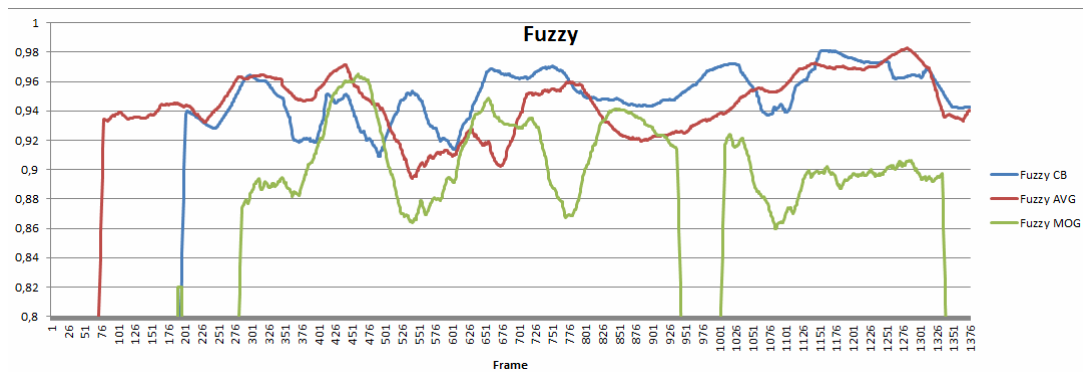


Figura 4.6: Evolução da medida *Fuzzy* para a sequência OSOW1, calculada para todos os métodos de segmentação definidos.

Por sua vez, a figura 4.7 mostra a evolução da medida *symetric partition-distance*, calculada nas mesmas condições da medida *fuzzy*.

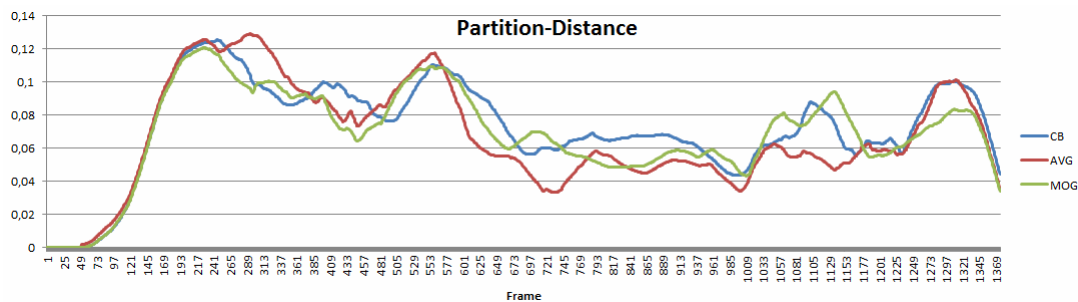


Figura 4.7: Evolução da medida *symetrics partition-distance* para a sequência OSOW1, calculada para todos os métodos de segmentação definidos.

Tabela 4.5: Coeficientes de correlação entre as medidas *symetric partition-distance* e *fuzzy*, calculados para cada sequência de imagens e para cada método de segmentação.

	CB	AVG	MOG
OSOW1	0,562	0,523	0,472
OSOW2	0,427	0,257	0,344
PETS	0,473	0,498	0,399

Analisando os valores de correlação entre as duas medidas, apresentados na tabela 4.5, pode concluir-se que não existe uma correlação muito forte entre as medidas. Os baixos valores de correlação podem significar que a medida *fuzzy* implementada não consegue estima corretamente a qualidade da detecção e do seguimento. Além disso, pela comparação entre os gráficos das figuras 4.6 e 4.7 não é possível obter um consenso relativamente ao método de segmentação que melhores resultados produz ao nível do seguimento. Por fim, interessa referir que, apenas com esta medida (sem usar qualquer tipo de informação de referência), não nos foi possível dizer se determinado método de segmentação está ou não, em cada momento, a obter bons resultados. É possível obter

bons resultados nesta medida estando apenas a seguir um objeto quando estão vários na cena, ou, pelo contrário, obter resultados menos bons estando a seguir corretamente mais objetos.

4.2.2 Método de decisão

A camada de decisão implementada tem um papel preponderante na plataforma. É responsável pela recolha dos resultados obtidos pelos diferentes métodos de avaliação e, com base nesses dados, tomar uma decisão quanto ao caminho a seguir. São recolhidos dados relativos à complexidade da cena, obtidos pelo método de *Optical Flow*, relativos à qualidade da segmentação, medidas *shape regularity* e *color contrast*, e relativos à qualidade da deteção e seguimento, medidas d_{color} , d_{motion} , d_{hist} e *fuzzy*. Além disso, será também recolhida informação sobre o número de objetos a serem seguidos, assim como as suas posições. Depois de recolhidas estas medidas são avaliadas e será tomada uma decisão relativamente ao método de segmentação/deteção de pessoas e relativamente à alteração da complexidade do algoritmo de seguimento. Será definido um conjunto de regras que serão responsáveis pela tomada de decisão.

Em primeiro lugar a camada de decisão faz uma avaliação da qualidade dos resultados do seguimento e, caso os mesmos sejam considerados maus, será alterado o método de segmentação/deteção. Depois, serão analisados os resultados relativamente ao tempo de processamento. Caso o tempo de processamento ultrapasse um determinado limite, será feita uma tentativa de redução da complexidade do algoritmo de seguimento. A camada de decisão poderá provocar apenas a alteração do método de segmentação/deteção, apenas a alteração da complexidade do algoritmo ou provocar a alteração quer do método de segmentação/deteção quer da complexidade do algoritmo de seguimento.

Relativamente ao impacto da camada de decisão no algoritmo de seguimento, a avaliação será faseada. Numa primeira fase será avaliado o impacto, quer em termos de qualidade do seguimento quer em termos de tempo de processamento, de cada decisão tomada individualmente no desempenho do algoritmo de seguimento. Serão avaliados individualmente os resultados obtidos pela alteração do método de segmentação/deteção e os resultados obtidos pela alteração da complexidade do algoritmo de seguimento. Numa segunda fase será feita uma avaliação do impacto que todas as decisões tomadas em conjunto têm nos resultados finais. De forma a avaliar o impacto que a camada de decisão tem nos resultados do algoritmo de seguimento foram definidas algumas experiências a realizar, que são descritas na secção 4.3.

4.3 Experiências a realizar

Foi definido um conjunto de experiências a realizar, de forma a avaliar o desempenho de todos os módulos que constituem a plataforma. Além da avaliação da qualidade dos resultados, os métodos serão também avaliados relativamente ao tempo de processamento.

4.3.1 Medição do tempo de processamento

Todos os métodos aqui apresentados serão avaliados quanto ao tempo necessário ao seu processamento. O desempenho do sistema de testes terá impacto direto nesta avaliação, mas uma vez que se pretende uma comparação entre os diferentes métodos, este impacto será eliminado. Para além dos tempos de cálculo apresentados de seguida, serão registados outros que se considerem relevantes durante a fase de testes. Todos os resultados serão apresentados em ms (milissegundos) e, caso se justifique, em FPS (frames por segundo).

-Tempo médio de *optical flow* (TMOF)

Intervalo de tempo médio necessário o cálculo da soma do módulo de todos os vetores de movimento, calculados pelo método de *optical flow*, para uma imagem.

-Tempo médio de segmentação(TMS)

Intervalo de tempo médio necessário para efetuar a segmentação de uma imagem, recorrendo a determinado método de segmentação.

-Tempo de cálculo da qualidade da segmentação (TQS)

Intervalo de tempo necessário ao cálculo das medidas de avaliação da qualidade da segmentação.

-Tempo médio de deteção por objeto (TDF)

Intervalo de tempo médio necessário à deteção de cada objeto numa imagem, usando o método de procura de cabeças.

-Tempo médio de deteção pelo método HOG (TDH)

Intervalo de tempo necessário à deteção de pessoas presentes numa imagem, quando é usado o método HOG.

-Tempo médio de avaliação da qualidade da deteção e seguimento(TMADS)

Tempo necessário ao cálculo das medidas de avaliação da qualidade do seguimento.

-Tempo médio de *Match* Intervalo de tempo médio necessário para a procura na imagem segmentada da melhor correspondência. Quando usado o detetor HOG o cálculo deste tempo não se justifica, uma vez que não é necessário este passo.

-Tempo de processamento total (TPT)

Intervalo de tempo medido desde o momento em que as imagens são carregadas até ao final do processamento. Estará incluído neste valor o tempo de cálculo de cada medida que seja necessário em cada experiência efetuada.

4.3.2 Alteração do método de segmentação/deteção de pessoas

Além da variação do método de segmentação, foi adicionado à plataforma um método de variação do algoritmo de deteção. O algoritmo de seguimento dispõe de um detetor de pessoas baseado na procura de cabeças, em imagens previamente segmentadas. Como alternativa a este método de deteção, foi implementado o detetor HOG que não necessita de obter uma segmentação prévia das

imagens, reduzindo assim erros inerentes a este processo. De forma a avaliar o desempenho de cada um dos métodos de detecção, os resultados de ambos os detetores (procura de cabeças e HOG) foram comparados com a informação de referência disponível para cada sequência. O algoritmo de seguimento foi modificado de forma a que a detecção de pessoas seja feita para todos os *frames*, sem qualquer informação relativa aos *frames* anteriores. A imagem 4.8 mostra a forma como será feita a avaliação dos resultados da detecção. Serão avaliados os resultados relativos à detecção, partindo das imagens segmentadas pelos métodos de segmentação implementados, para cada uma das sequências de teste. Os resultados foram avaliados recorrendo às medidas *frame-based metrics*, baseadas na informação relativa à *bounding-box* dos objetos detetados e calculadas por *frame*. Esta avaliação apenas será feita para as sequências OSOW1 e OSOW2, uma vez que não estava disponível a informação de referência relativa à detecção e ao seguimento para a sequência PETS. Além da avaliação dos resultados pela comparação com informação de referência, os métodos de detecção propostos serão também avaliados relativamente ao tempo necessário para que o método efetua a detecção em cada *frame*.

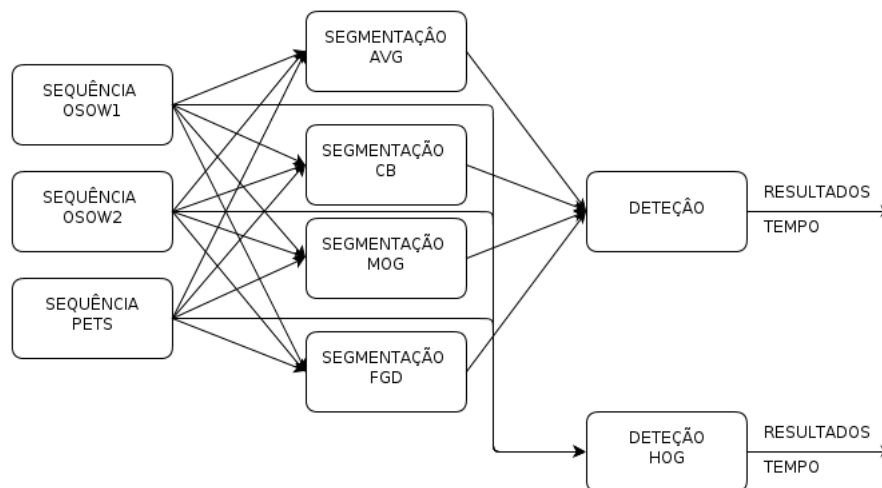


Figura 4.8: Diagrama ilustrativo da avaliação da detecção.

Os resultados da detecção obtidos pelo detetor HOG serão depois comparados com os resultados obtidos pelo detetor de cabeças, usando a segmentação que melhores resultados obteve.

Será ainda comparada a qualidade do seguimento, quando se usa o detetor HOG, com a qualidade do seguimento quando se usa o método de procura de cabeças. Mais uma vez, o método de procura de cabeças usará a segmentação que permite obter os melhores resultados.

4.3.3 Modificação do processo de definição da correspondência

Como se percebe pela análise das figuras 4.9 e 4.10, a procura na imagem segmentada pela melhor correspondência é um processo muito dispendioso a nível de tempo de processamento. Esta fase é responsável por 50%-75% do tempo total de processamento. Quando não são incluídos no algoritmo os métodos de cálculo das medidas implementadas (*Optical Flow*, *Shape Regularity*, *Color Contrast* e *ngtMetrics*), a fase de procura pela melhor correspondência tem um peso de

70%-90% no tempo de processamento total. Tendo em conta estes valores, esta fase é uma forte candidata a uma redução da complexidade e consequentemente a uma redução do seu peso no tempo de processamento total.

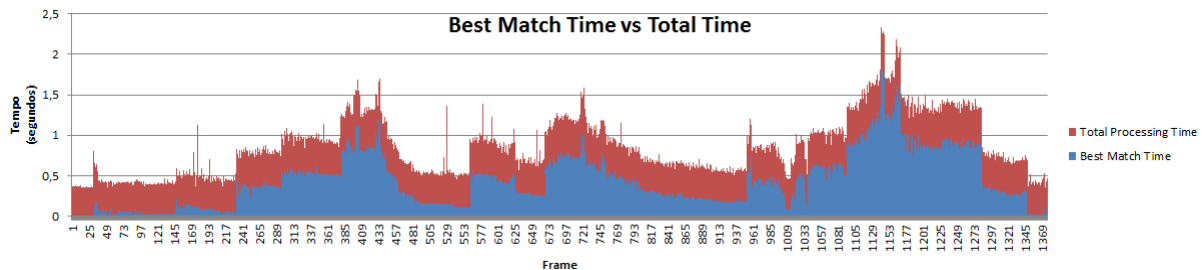


Figura 4.9: Evolução da tempo de match em comparação com o tempo de processamento total, ao longo da sequência OSOW1 .

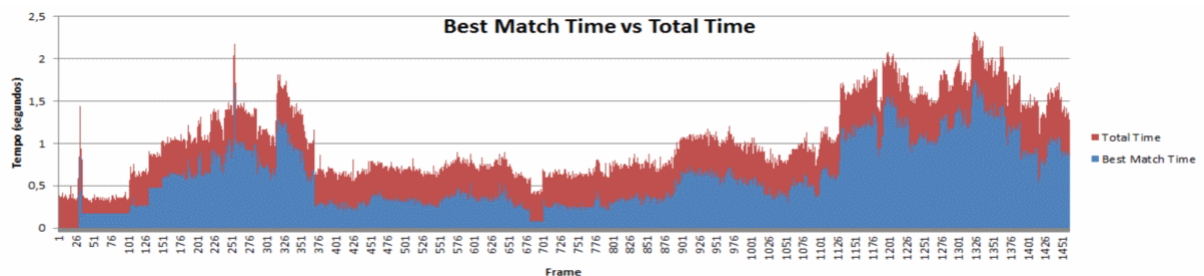


Figura 4.10: Evolução da tempo de match em comparação com o tempo de processamento total, ao longo da sequência OSOW2 .

Foram definidos modos de execução que levam em consideração diversos fatores na decisão relativa à procura ou não da melhor correspondência. De forma a avaliar os resultados de cada um dos modos definidos, são calculados numa primeira fase os resultados obtidos pelo algoritmo base. Os resultados serão obtidos pelo processamento integral de todos os *frames*, usando as imagens segmentadas pelo método que melhores resultados obteve, no que diz respeito ao seguimento. Serão obtidos resultados para as sequências de teste definidas, na qual esteja disponível informação de referência relativa ao seguimento.

Modo USkip(*Unassisted skip*): A procura da melhor correspondência na imagem original não é feita, em todos os objetos detetados, em *frames* alternados. Quando é ignorada a procura pela melhor correspondência e considerada a estimativa obtida como posição correta do objeto. Este modo é importante como medida de comparação para os modos definidos de seguida. É um modo que não leva em conta qualquer informação relativa à cena.

Modo ASkip(*Assisted skip*): A decisão relativa à procura ou não da melhor correspondência é feita por objeto e não em todos os objetos do mesmo *frame*. A procura pela melhor correspondência não é realizada nos objetos que tenham na sua vizinhança outros objetos. É considerada vizinhança os objetos cujo seu centro se encontre a uma distância inferior à diagonal da *bounding-box* do objeto em causa. Além disso definiu-se que, para cada objeto, em cada três *frames* a procura pela melhor correspondência tem de ser feita pelo menos uma vez. Esta é uma forma de obter um controlo maior sobre a procura pela melhor correspondência.

Modo CSkip(*Covariance skip*): Trata-se de uma extensão ao modo ASkip. Além das duas restrições referidas no modo anterior é também adicionada uma restrição relativa ao módulo do *optical flow* na vizinhança do objeto e outra relativa à covariância do erro, obtido pelo filtro de *Kalman*, para o objeto em questão. Como já foi referido anteriormente, o algoritmo usa o filtro de *Kalman* no processo de seguimento. Este filtro pode ser dividido em duas fases: 1) Previsão; 2) Correção. Na fase de previsão é calculada uma estimativa da posição do objeto na *frame* seguinte, baseando-se na posição do objeto nos *frames* anteriores, e calculada a covariância do erro da estimativa (*a priori*). Na fase da correção a posição estimado do objeto é atualizada, pela incorporação de informação relativa à posição observada do objeto, e atualizada covariância do erro (*measurement error*). A fusão entre a posição estimada e a posição observada é feita com base no ganho K. O ganho K pode ser entendido como uma medida da confiança que o filtro atribui à estimativa. Isto é, quando a covariância do *measurement error* se aproxima de zero, a observação é considerada como fiável, sendo dada maior importância à observação e menor importância à estimativa. Por outro lado, se a covariância do erro da estimativa *a priori* se aproxima de zero a estimativa é considerada com fiável, sendo atribuída maior importância à estimativa e menor importância à observação [8]. O erro da observação (*measurement error*) tende para zero quando a posição estimada se aproxima da posição observada e tende a afastar-se de zero quando a posição estimada se afasta da posição observada. Assim, considera-se que o objeto tem um comportamento bem definido (a estimativa relativa à posição do objeto tem sido, nos instantes anteriores, dada com grande confiança pelo filtro) quando a covariância do erro da observação (*measurement error*) se aproxima de zero. Com base nisto, foi definido um valor limiar a partir do qual se considera que o erro da observação é baixo. Considera-se então que o objeto está a ser seguido corretamente e que a estimativa dada é boa, pelo que, caso se verifiquem as restantes condições definidas no modo CSkip, será possível ignorar a procura pela melhor correspondência nos *frames* seguintes, aceitando a estimativa como posição real do objeto.

Os modos definidos visam avaliar os ganhos obtidos ao nível do tempo de processamento, mas também garantir que a qualidade dos resultados do seguimento não seja reduzida substancialmente.

Capítulo 5

Resultados

Neste capítulo serão apresentados e discutidos os resultados obtidos nos diversos testes efetuados. Os resultados serão avaliados relativamente à qualidade e relativamente ao esforço computacional necessário à obtenção desses mesmos resultados.

5.1 Segmentação

Os resultados da segmentação obtidos pelos quatro métodos de segmentação implementados (AVG,CB,MOG e FGD), serão avaliados comparativamente com a segmentação de referência, recorrendo à medida *symetric partition-distance*. A figura 5.1 mostra a segmentação usando cada um dos métodos propostos, numa imagem de teste retirada da sequência PETS2006.

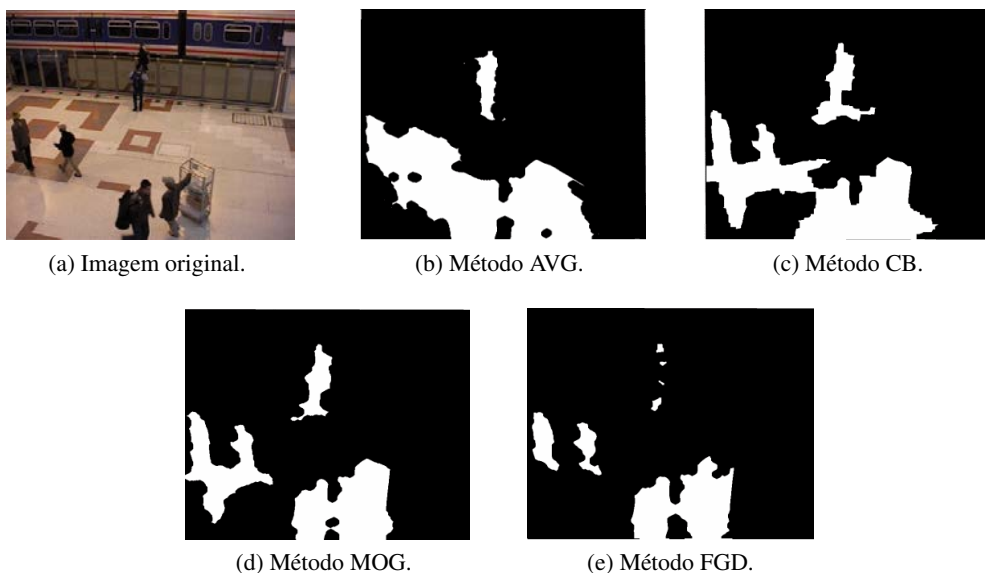


Figura 5.1: Segmentação de uma imagem pelos métodos de segmentação propostos.

De forma a ser possível esta comparação, são fornecidas algumas imagens, correspondendo a *frames* regularmente espaçados no tempo, contendo a segmentação ideal desses mesmos *frames*.

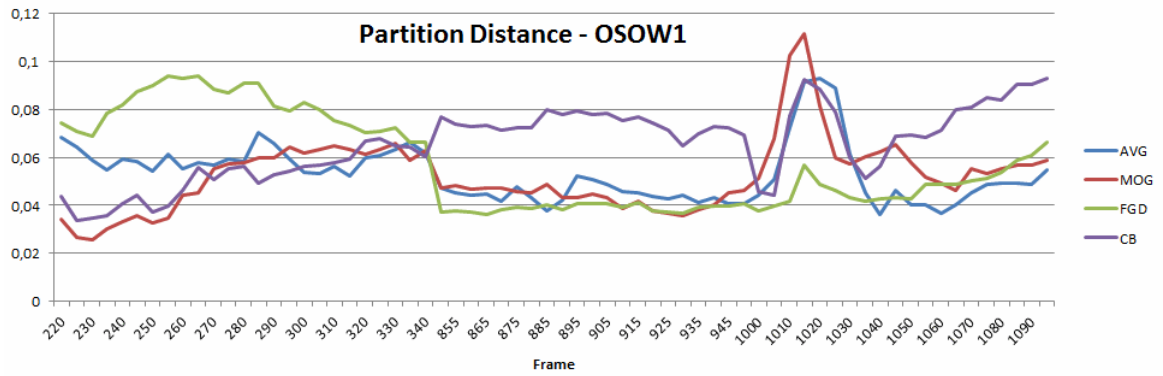
Desta forma foi possível traçar os gráficos da figura 5.2 onde são apresentados os resultados obtidos pelos métodos de segmentação implementados. Pela análise dos gráficos é possível fazer uma avaliação do método que produz melhores resultados em cada *frame*. A medida *symetric partition-distance* é uma medida de erro, pelo que melhores resultados da segmentação correspondem a menores valores nesta medida.

Como se percebe pela análise da figura 5.2a, nenhum método de segmentação se destaca dos restantes, relativamente à qualidade dos seus resultados. Não é possível afirmar que determinado método de segmentação obtém melhores resultados ao longo de toda a sequência OSOW1. Apesar disto é possível verificar que ao longo de certas secções da sequência existem métodos com melhores resultados. Por exemplo, entre os *frames* 850 e 1040 verifica-se que o método FGD produz um erro de segmentação inferior aos restantes. Do mesmo modo verifica-se que, para esta mesma secção, o método CB apresenta os erros mais elevados.

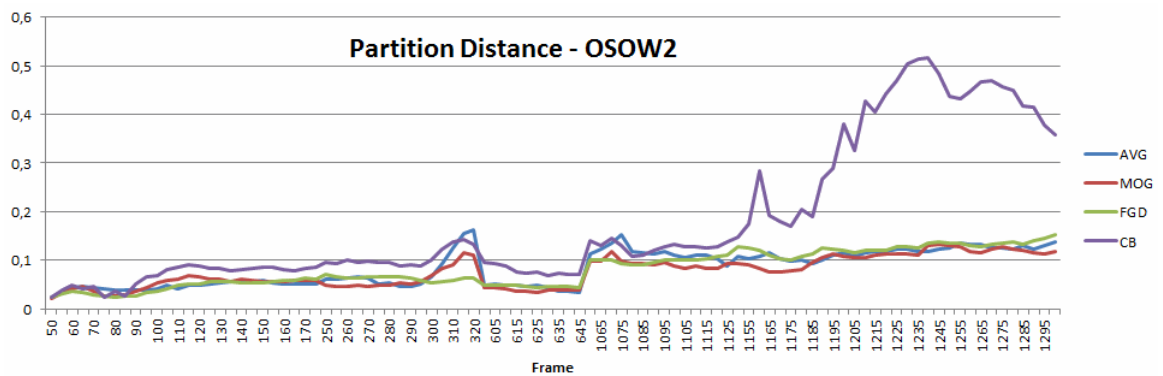
Relativamente aos resultados obtidos para a sequência OSOW2, figura 5.2b, verifica-se que o método CB produz os piores resultados ao longo de toda a sequência, em especial a partir do *frame* 1090. Não se pode retirar uma conclusão quanto ao melhor método de segmentação para a totalidade desta sequência, uma vez que os métodos FGD, MOG e AVG vão alternando na obtenção dos melhores resultados em cada *frame*. À semelhança do verificado para a sequência OSOW1, também para esta sequência se pode verificar que, para determinadas secções da sequência, existem métodos que produzem um menor erro de segmentação que os restantes.

Para a sequência PETS, figura 5.2c, conclui-se que o método de segmentação FGD produz valores de erro inferiores aos obtidos para os restantes métodos, ao longo da quase totalidade da sequência. De referir também que, no global, o método AVG apresenta os piores resultados. O fato de ao longo da totalidade das sequências não haver um método que se destaque não significa que os métodos produzam resultados semelhantes ao nível da deteção de pessoas, como se comprova pela análise da tabela 5.2. Os resultados da deteção obtidos pelo método de segmentação FGD, quer para a sequência OSOW1 quer para a sequência OSOW2, são muito inferiores aos resultados obtidos pelos outros métodos de segmentação apesar de o mesmo não se verificar nos resultados obtidos pela medida *symetric partition-distance*. De notar também que o fato de a apenas termos disponível a segmentação de referência de alguns *frames* espaçados no tempo pode comprometer os resultados obtidos pela medida *symetric partition-distance*.

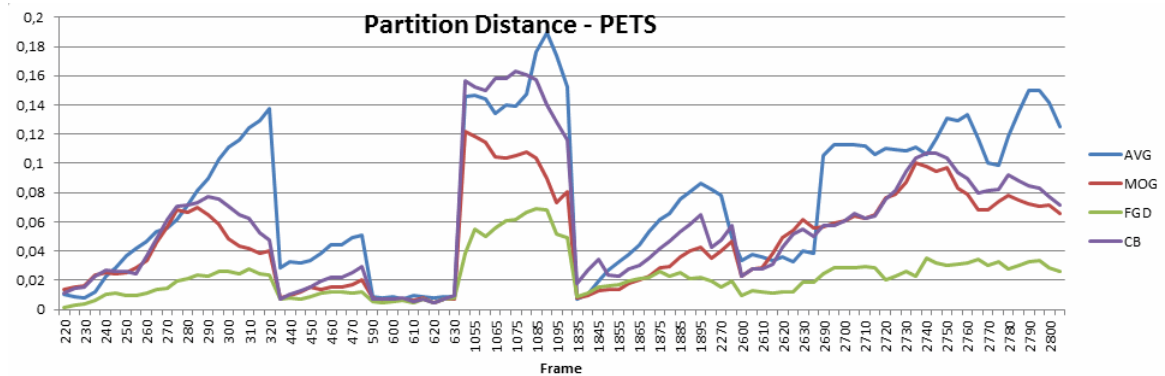
Uma importante parte da análise dos resultados da segmentação diz respeito ao tempo necessário para que os *frames* de cada sequência sejam segmentados. Em algoritmos de seguimento, poderá ser necessário proceder a reduções do tempo de processamento, de forma a manter o funcionamento em tempo real, à custa de uma possível diminuição ao nível da qualidade dos resultados. Desta forma é essencial uma análise cuidada ao tempo necessário para a segmentação de cada *frame* (tempo de segmentação). O tempo necessário à segmentação, quando a mesma é obtida pelos métodos definidos, não varia de forma significativa ao longo de cada sequência, como se pode verificar pela análise da figura 5.3, onde são traçados os gráficos da evolução do tempo de segmentação por *frame*, para a sequência OSOW1, OSOW2 e PETS. Isto significa que o tempo de segmentação não varia de forma significativa com o número, ou com as dimensões dos objetos a



(a) ao longo da sequência OSOW1.



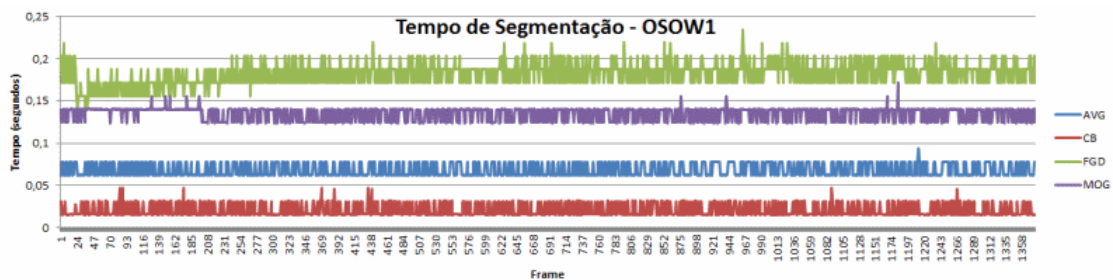
(b) ao longo da sequência OSOW2.



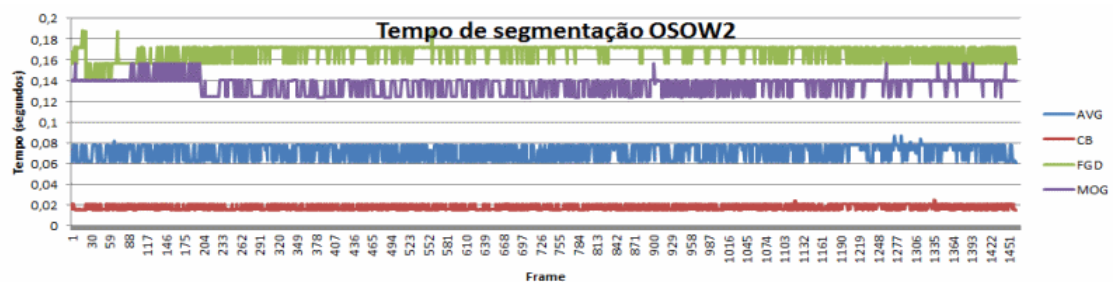
(c) ao longo da sequência PETS.

Figura 5.2: Evolução da medida *symetric partition-distance* para os métodos de segmentação.

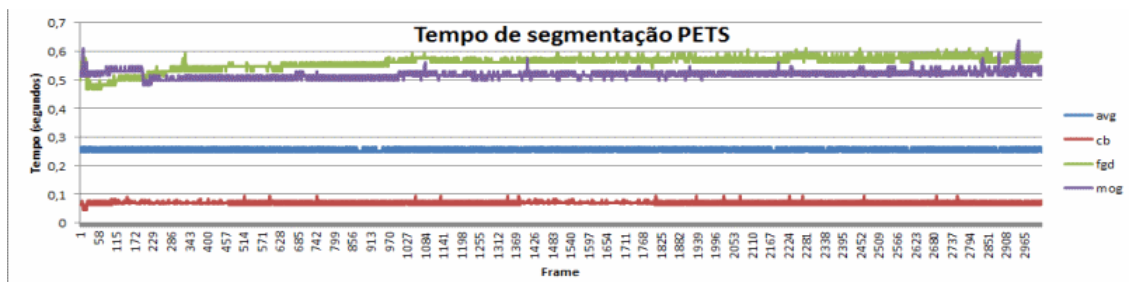
segmentar. Pequenas variações no tempo dizem principalmente respeito a variações no desempenho da máquina de testes. Verificou-se apenas uma variação relativamente à dimensão da imagem a segmentar. Assim, uma vez que os *frames* das sequências OSOW1 e OSOW2, pertencentes ao *dataset* Caviar, são de iguais dimensões, os resultados foram agrupados. A tabela 5.1a apresenta o tempo de segmentação para as sequências do *dataset* Caviar enquanto que a tabela 5.1b apresenta o tempo de segmentação para a sequência PETS. Pela análise dos valores das tabela 5.1, verifica-se que o método de segmentação mais rápido, para todas as sequências de teste, é o CB, enquanto que o menos rápido é o FGD.



(a) ao longo da sequência OSOW1.



(b) ao longo da sequência OSOW2.



(c) ao longo da sequência PETS.

Figura 5.3: Evolução do tempo de segmentação, para cada método de segmentação definido.

Não faz parte dos objetivos a maximização dos resultados da segmentação. Apenas se pretende obter segmentações diferentes, ao nível da qualidade e peso computacional, das mesmas sequências de forma a ser possível obter uma comparação. Contudo, para as várias segmentações obtidas, verificou-se que não existe uma relação direta entre qualidade e tempo necessário à segmentação.

Tabela 5.1: Tempo médio, máximo e mínimo (em segundos) para a segmentação por *frame*, usando os métodos de segmentação definidos.

(a) Sequências do <i>dataset</i> Caviar (OSOW1 e OSOW2).				
	AVG	CB	FGD	MOG
Médio (s)	0,068	0,020	0,182	0,135
Máximo (s)	0,094	0,047	0,234	0,171
Mínimo (s)	0,062	0,015	0,140	0,124
Variância (s)	0,00005	0,00005	0,00057	0,00016
(b) Sequência do <i>dataset</i> PETS2006.				
	AVG	CB	FGD	MOG
Médio (s)	0,255	0,070	0,562	0,519
Máximo (s)	0,266	0,094	0,609	0,640
Mínimo (s)	0,249	0,046	0,468	0,483
Variância (s)	0,00006	0,00005	0,00016	0,00006

5.2 Detecção de pessoas

Depois da avaliação das várias segmentações produzidas é necessário avaliar a qualidade dos resultados do método de deteção de pessoas, procura de cabeças, existente no algoritmo de seguimento. Desta forma pode verificar-se qual o método de segmentação que produz melhores resultados ao nível da deteção de pessoas. Para além desta avaliação, será também avaliado o desempenho do detetor HOG implementado.

Em primeiro lugar é avaliado o desempenho do detetor de cabeças partindo da segmentação obtida pelos diferentes métodos de segmentação. A segmentação que obtiver melhores resultados será usada como comparação com o detetor HOG. Será considerada o melhor método de segmentação, o método que obtiver valores mais elevados na medida *F-Score*, definida na secção 3.6 como uma combinação das medidas *Recall* e *Precision*. A medida *Recall* corresponde à percentagem dos objetos relevantes que foram detetados enquanto que a medida *Precision* representa a percentagem dos objetos detetados que são relevantes. Considerou-se que estas duas medidas, das várias apresentadas, são as que melhor avaliam a qualidade da deteção, no contexto desta dissertação.

Da análise da tabela 5.2, onde são apresentados os resultados da deteção pelo método de procura de cabeças, 5.2a para a sequência OSOW1 e 5.2b para a sequência OSOW2, chega-se à conclusão que o detetor baseado na procura de cabeças obtém melhores resultados quando a segmentação é obtida pelo método AVG, para as duas sequências testadas. De referir que o método FGD apresenta valores superiores aos obtidos pelo método AVG, na medida *Precision*, mas apresenta resultados muito inferiores na medida *Recall*. Isto significa que, partindo da segmentação obtida pelo método FGD, o detetor apenas foi capaz de detetar na imagem uma percentagem muito reduzida de pessoas, ainda que as mesmas tenham sido detetadas corretamente. Pode-se também verificar que o método CB é o que produz um valor mais reduzido na medida *Precision*, o que

Tabela 5.2: Resultados da detecção pelo método de procura de cabeças.

(a) sequência OSOW1.				
	AVG	CB	MOG	FGD
Recall (R)	0,615	0,590	0,426	0,001
Precision (P)	0,934	0,779	0,917	1,000
F-Score (FS)	0,742	0,671	0,582	0,002
(b) sequência OSOW2.				
	AVG	CB	MOG	FGD
Recall (R)	0,751	0,644	0,399	0,030
Precision (P)	0,890	0,681	0,873	1,000
F-Score (FS)	0,815	0,662	0,548	0,058

significa que uma elevada percentagem das detecções são erradas.

Comparação da detecção pelo método de procura de cabeças e pelo método HOG.

De forma a podermos considerar ou não o detetor HOG como uma alternativa ao detetor baseado na procura de cabeças, foi feita uma comparação dos resultados obtidos pelos dois métodos. Além de avaliado o desempenho em termos de qualidade dos resultados foi também avaliado o desempenho em termos temporais. A comparação é feita entre o detetor de cabeças, usando a segmentação que melhores resultados produz (AVG, valor mais elevado na medida *F-Score*, tabela 5.2), e o detetor HOG.

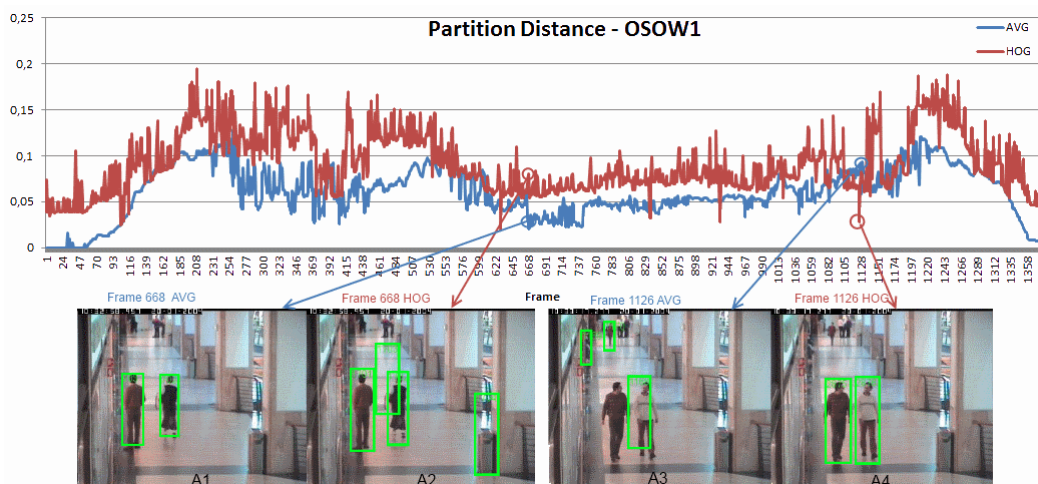
Tabela 5.3: Comparação entre os resultados da detecção obtidos pelo método de procura de cabeças e pelo método HOG.

	OSOW1			OSOW2		
	AVG	HOG	Δ_1	AVG	HOG	Δ_2
Recall (R)	0,615	0,740	20%	0,751	0,877	17%
Precision (P)	0,934	0,735	-21%	0,890	0,899	1%
F-Score (FS)	0,742	0,737	-1%	0,815	0,883	8%

Como se percebe pela análise da tabela 5.3, para a sequência OSOW1, o detetor HOG produz melhores resultados na medida *Recall*(+20%), mas piores resultados na medida *Precision*(-21%). Isto significa que o detetor HOG consegue um número superior de detecções, ainda que uma grande parte sejam detecções erradas. Na combinação das duas medidas obtêm-se uma diminuição de 1%, medida *F-Score*. Para a sequência OSOW2 o detetor HOG produz resultados melhores nestas duas medidas (+17% e +1% respetivamente), quando comparado com a procura por cabeças.

O desempenho dos dois detetores de pessoas foi ainda comparado recorrendo à medida *partition-distance*. A figura 5.4 mostra a evolução da medida calculada ao longo da sequência de teste, para os dois métodos de detecção em análise. Como se percebe pela análise da figura 5.4a, relativa à

sequência OSOW1, também esta medida mostra um desempenho superior do detetor baseado na procura de cabeças, relativamente ao detetor HOG, confirmando os resultados obtidos pelas medidas baseadas em *bounding-box*. No entanto também se verifica que em certos *frames* isolados o detetor HOG consegue resultados superiores. Na figura são também mostrados os resultados da deteção, obtidos pelo detetor de cabeças e pelo detetor HOG, em determinados *frames* das sequências.



(a) Ao longo da sequência OSOW1.



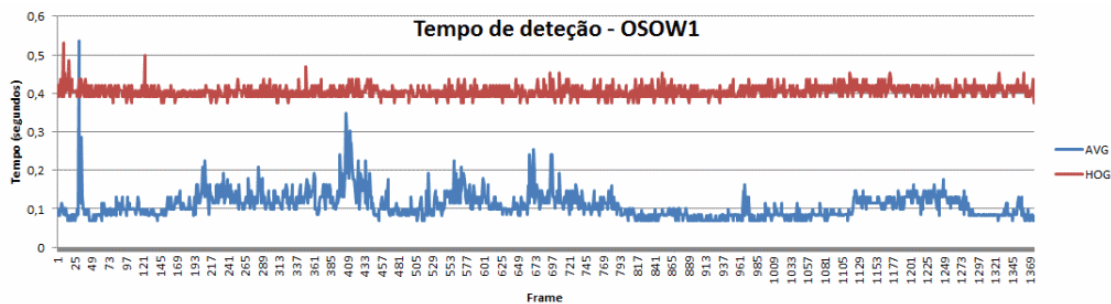
(b) Ao longo da sequência OSOW2.

Figura 5.4: Evolução da medida *symetric partition-distance* na avaliação da qualidade da deteção, pelo método de deteção de cabeças (segmentação AVG) e o pelo método de deteção HOG.

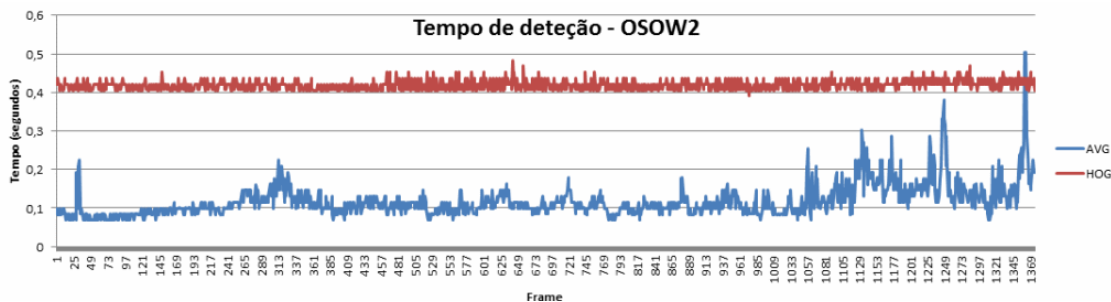
Analisando os resultados obtidos para a sequência OSOW2, figura 5.4b, verifica-se que, apesar de os melhores resultados ao longo de grande parte da sequência serem obtidos pelo detetor baseado na procura de cabeças, no último terço da sequência o detetor HOG produz resultados superiores. Isto fica a dever-se ao fato de a sequência OSOW2 apresentar, na parte final, um elevado número de objetos. A segmentação de imagens com grande densidade de objetos pode ser problemática uma vez que as operações de pós-processamento tendem a fundir os objetos mais próximos

num só, dificultando depois a tarefa do detetor. Desta forma justifica-se a obtenção de melhores resultados do detetor HOG, relativamente ao detetor baseado em segmentação, em sequências com grande densidade de objetos. As medidas baseadas em *bounding-box* confirmam isto mesmo, tabela 5.3, apresentando resultados superiores nas medidas *Recall* e *Precision*.

Relativamente ao tempo necessário à deteção, este é, na quase totalidade dos *frames*, inferior quando a deteção é baseada na segmentação, como se percebe pela análise dos gráficos da figura 5.5. Apenas em *frames* esporádicos, nas sequências testadas, isto não acontece. É ainda possível verificar que o detetor HOG apresenta um tempo de deteção que se mantém praticamente constante ao longo de toda a sequência, sendo invariável ao número de objetos presentes. É ainda importante verificar que, para um número elevado de deteção por *frame*, o tempo de deteção pela procura de cabeças de aproxima do tempo obtido pelo detetor HOG. É importante referir que o tempo de deteção, quando calculado pela procura de cabeças, já leva em consideração o tempo necessário à segmentação das imagens.



(a) ao longo da sequência OSOW1.



(b) ao longo da sequência OSOW2.

Figura 5.5: Comparação da evolução da tempo de deteção entre o método de procura de cabeças (segmentação AVG) e o método HOG.

Na tabela 5.4 são apresentados os valores totais do tempo de deteção (em segundos), para cada sequência. Mais uma vez se comprova os resultados apresentados anteriormente, para a sequência OSOW1. O detetor HOG apresenta um número mais elevado de deteções assim como um maior tempo de deteção.

Relativamente à sequência PETS, verificou-se que o detetor HOG não apresenta resultados satisfatórios. O tipo de imagens que compõem a sequência PETS têm características muito diferentes das imagens das sequências Caviar (OSOW1 e OSOW2), principalmente ao nível das dimensões

Tabela 5.4: Comparação do tempo de deteção e do tempo total de processamento, em segundos, entre o detetor de cabeças e o detetor HOG.

	OSOW1		OSOW2	
	AVG	HOG	AVG	HOG
Tempo de deteção (s)	55,130	554,645	77,620	610,000
Tempo de segmentação (s)	93,568	0,000	99,348	0,000
Tempo total (s)	148,698	554,645	176,968	610,000
Número de deteções	1383	3311	2542	4107

e da qualidade da mesma. Assim, é necessário treinar o detetor para este tipo de sequência, de forma a melhorar os resultados obtidos.

5.3 Seguimento

Após a avaliação dos resultados obtidos para a deteção de pessoas é necessário avaliar os resultados na fase do seguimento. Serão em primeiro lugar avaliados os resultados obtidos para cada segmentação definida e depois é feita a comparação com os resultados obtidos quando o detetor de pessoas é o detetor HOG.

O algoritmo foi testado usando o detetor baseado na procura de cabeças, partindo das segmentações obtidas por cada método de segmentação. Os resultados foram avaliados recorrendo às medidas baseadas em *bounding-box* e são apresentados na tabela 5.5. Ao contrário da avaliação da deteção, as medidas eram calculadas por *frame*, na avaliação do seguimento as medidas serão calculadas por objeto. Destas medidas destacam-se a *Recall*, a *Precision*, a *F-Score* e a *False Alarm Rate*, que consideramos serem as mais indicadas (no contexto desta dissertação) para avaliar a qualidade do seguimento. Foi ainda registado o tempo dispendido pelo algoritmo de seguimento, para cada segmentação usada. Estes valores obtidos servirão como resultados base do algoritmo de seguimento, na comparação com resultados obtidos depois da implementação dos métodos de melhoria de desempenho ou diminuição do tempo de processamento.

Da análise da tabela 5.5, onde estão apresentados os resultados obtidos ao nível seguimento, conclui-se que a segmentação obtida pelo método AVG obtém os melhores resultados tanto nas medidas *Recall* e *Precision* como na medida *False Alarm Rate*, para as duas sequências testadas. De realçar é também o facto de o método de segmentação FGD obter resultados muito inferiores aos obtidos pelos restantes métodos. Apesar deste método de segmentação levar a bons resultados nas medidas *Precision* e *False Alarm Rate*, verifica-se que apenas é efetuado o seguimento de uma pequena percentagem dos objetos presentes na sequência (baixo valor de *Recall*). Verifica-se portanto que o método de segmentação AVG conduz a melhores resultados ao nível do seguimento, quando comparado com os restantes métodos.

Relativamente ao tempo de processamento, tabela 5.6, verifica-se que a segmentação FGD leva a um menor tempo de processamento (ainda que partindo de um tempo de segmentação mais elevado) devido principalmente ao número reduzido de objetos a processar. Ainda que o método

Tabela 5.5: Resultados do seguimento, partindo da segmentação obtida por cada um dos métodos definidos.

(a) Para a sequência OSOW1.

	AVG	CB	MOG	FGD
Recall (R)	0,497	0,350	0,217	0,010
Precision (P)	0,647	0,525	0,536	1,000
F-Score (FS)	0,562	0,420	0,309	0,019
False Alarme Rate (FAR)	0,353	0,475	0,464	0,000

(b) Para a sequência OSOW2.

	AVG	CB	MOG	FGD
Recall (R)	0,516	0,505	0,281	0,051
Precision (P)	0,582	0,217	0,734	0,965
F-Score (FS)	0,547	0,304	0,407	0,098
False Alarme Rate (FAR)	0,418	0,783	0,266	0,035

Tabela 5.6: Tempos necessário, em segundos, para que o algoritmo proceda ao seguimento, usando as segmentações obtidas pelos diferentes métodos de segmentação definidos.

(a) Para a sequência OSOW1.

	AVG	CB	MOG	FGD
Segmentation Time (s)	93,568	27,52	185,76	250,432
Processing Time (s)	659,647	1017,587	560,54	25,015
Total Time (s)	753,215	1045,107	746,3	275,447

(b) Para a sequência OSOW2.

	AVG	CB	MOG	FGD
Segmentation Time (s)	99,348	29,22	197,235	265,902
Processing Time (s)	984,426	2291,301	527,609	111,959
Total Time (s)	1083,774	2320,521	724,844	377,861

de segmentação MOG leve a um menor tempo total de processamento, conclui-se assim que a melhor relação entre resultados obtidos e tempo de processamento é obtida quando a segmentação é dada pelo método AVG.

Comparação dos resultados do seguimento, entre o método de procura de cabeças e o detetor HOG

A comparação dos resultados do seguimento tem importância de forma a avaliar as alterações introduzidas no algoritmo de seguimento de forma a suportar o detetor HOG. Serão comparados os resultados obtidos quando a deteção é feita pelo método HOG com os resultados obtidos pela procura de cabeças, partindo da segmentação AVG (melhores resultados entre os métodos de segmentação). Os resultados desta comparação estão apresentados nas tabelas 5.7 e 5.8.

Tabela 5.7: Comparação dos resultados do seguimento entre o detetor pela procura de cabeças e o detetor HOG, para as sequências de teste.

	OSOW1			OSOW2		
	AVG	HOG	Δ_1	AVG	HOG	Δ_2
Recall (R)	0,497	0,442	-11%	0,516	0,265	-49%
Precision (P)	0,647	0,626	-3%	0,582	0,55	-6%
F-Score (FS)	0,562	0,518	-8%	0,547	0,358	-35%
False Alarm Rate (FAR)	0,353	0,374	6%	0,418	0,45	8%

Pela análise da tabela 5.7 rapidamente se pode concluir que os resultados ao nível do seguimento, partindo da deteção pelo método HOG, foram prejudicados. Pode verificar-se uma quebra generalizada em todas as medidas apresentadas, nas duas sequências de teste. Destaca-se a diminuição de 49% no número de objetos relevantes detetados na sequência OSOW2, assim como o aumento, nas duas sequências, do número de falsas deteções. Quando comparados os resultados obtidos para a sequência OSOW2 com os resultados obtidos para a sequência OSOW1, verifica-se que a diminuição da medida *Recall* é muito superior para esta sequência. Como já foi dito anteriormente, a sequência OSOW2 é caracterizada por uma elevada densidade de pessoas, o que influenciou estes resultados. As alterações introduzidas no algoritmo de seguimento, de modo a suportar a deteção de pessoas pelo detetor HOG, não obtêm bons resultados em situações como esta, de grande densidade de pessoas, degradando os resultados do seguimento.

Tabela 5.8: Comparação do tempo de processamento (em segundos) para o seguimento entre o detetor HOG e o detetor baseado na procura de cabeças.

	OSOW1			OSOW2		
	AVG	HOG	Δ_1	AVG	HOG	Δ_2
Segmentation Time (s)	93,568	0	—	99,348	0	—
Processing Time (s)	659,647	1132,782	72%	984,426	1296,621	32%
Total Time (s)	753,215	1132,782	50%	1083,774	1296,621	20%

No que diz respeito ao tempo de processamento a tabela 5.8 apresenta a comparação entre os resultados obtidos. Verificou-se um aumento de 50% para a sequência OSOW1 e um aumento de 20% para a sequência OSOW2 no tempo total, relativamente ao tempo obtido pela segmentação AVG. Apesar de o detetor HOG não necessitar de segmentação, o tempo total é superior ao obtido quando se usa a procura de cabeças.

Apesar dos bons resultados obtidos pelo detetor HOG relativamente à deteção de pessoas, tabela 5.3, os resultados obtidos no seguimento, principalmente na sequência OSOW2, foram comprometidos relativamente ao detetor baseado na procura de cabeças, tabela 5.7. A explicação reside no fato de terem sido feitas alterações ao algoritmo de forma a efetuar o seguimento quando a deteção é feita pelo detetor HOG. As alterações efetuadas estão explicadas na secção 3.4.2.

Tendo em conta estes resultados conclui-se que a solução encontrada para adaptar o algoritmo de seguimento ao detetor HOG requer melhorias de forma a poder considerar este detetor como uma alternativa ao detetor baseado na procura de cabeças, quando se pretende efetuar o seguimento.

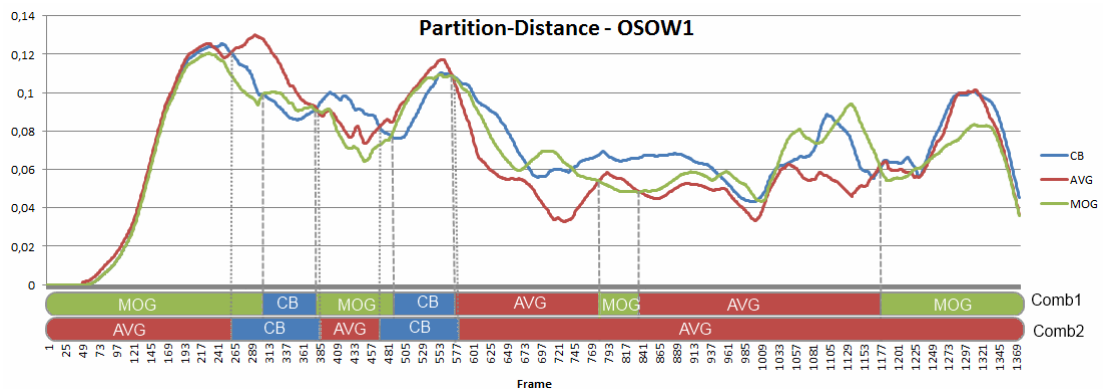
5.4 Alteração da segmentação/deteção

Como já foi verificado, os resultados do algoritmo de seguimento são sensíveis à qualidade da segmentação. Tendo isto em conta, foi implementado um método que, recolhendo os resultados obtidos pelas medidas de avaliação (que não se baseiam em informação de referência) da complexidade da cena e da qualidade da segmentação, da deteção e do seguimento, altera o método de segmentação na tentativa de obtenção de melhores resultados no seguimento.

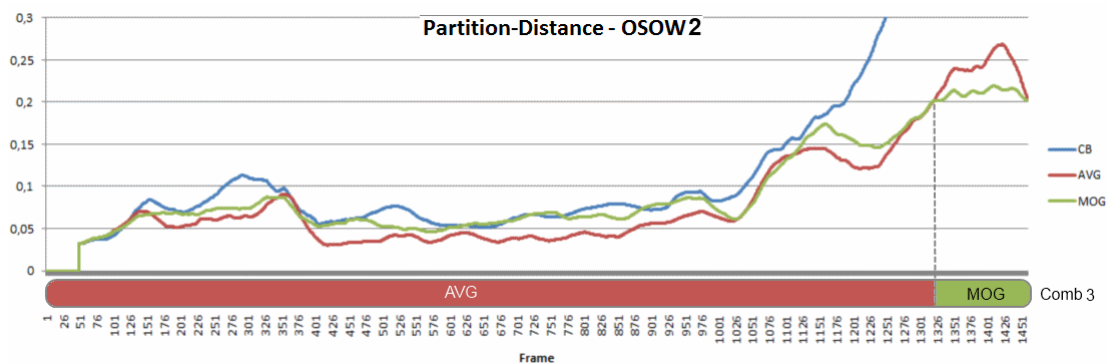
5.4.1 Alteração manual

De forma a analisar a possibilidade de alteração do método de segmentação durante a execução do algoritmo de seguimento, foi usada uma combinação da melhor segmentação em cada *frame* da sequência. Este teste será importante na medida em analisa a possibilidade de alteração do método de segmentação ou de deteção, recorrendo a medidas de avaliação baseadas em informação de referência, obtendo assim valores comparativos para a adaptação baseada em medidas sem informação de referência. A escolha da melhor segmentação foi feita de forma manual, baseada nos resultados obtidos pela medida *symetric partition-distance*. Esta medida foi calculada para os resultados obtidos ao nível do seguimento, partindo das segmentações obtidas por cada um dos métodos de segmentação. Desta forma foram traçados os gráficos, apresentado na figura 5.6, que mostram a evolução da medida ao longo das sequências de teste. Os resultados foram avaliados e foi selecionado o método de segmentação que apresenta menor erro em cada *frame*. Depois de testado o algoritmo de seguimento, usando esta combinação de segmentações, foram comparados os resultados com os resultados obtidos pelo método AVG. O método AVG obteve os melhores resultados, quando comparado com os restantes métodos de segmentação, ver tabela 5.9.

Numa primeira fase, a escolha é feita entre os três melhores métodos de segmentação, criando a denominada COMB1 (AVG, CB, MOG). Por melhor método entende-se o método de



(a) Ao longo da sequência OSOW1.



(b) Ao longo da sequência OSOW2.

Figura 5.6: Evolução da medida *symetric partition-distance* para os métodos de segmentação definidos ao longo da sequência OSOW1.

segmentação que melhores resultados apresenta ao nível do seguimento, recorrendo à medida *F-Score* (tabela 5.9). Na segunda fase, de forma a diminuir o número de alterações do método de segmentação, essa escolha é limitada apenas aos dois melhores métodos (AVG, CB), COMB2. As barras coloridas da figura 5.6 mostram a composição, em termos de método de segmentação, das combinações escolhidas. Para a sequência OSOW2 apenas foi definida uma combinação de segmentações devido ao fato de, para esta sequência, o método de segmentação CB apresentar resultados sempre inferiores aos apresentados pelos restantes métodos. Foi excluído deste teste a segmentação dada pelo método FGD uma vez que o mesmo produz resultados muito inferiores aos restantes. De forma a limitar a frequência com que é feita a alteração da segmentação, aos valores obtidos pela medida *symetric partition-distance* foi aplicado um filtro de média de dimensão 50

Tabela 5.9: Valores obtidos pela medida *F-score* quando são usadas as segmentações obtidas por cada um dos métodos de segmentação, para as sequências de teste.

	AVG	CB	MOG	FGD
OSOW1	0,562	0,420	0,309	0,019
OSOW2	0,547	0,304	0,407	0,098

frames e definiu-se um valor mínimo de 20 *frames* entre alterações consecutivas.

Os resultados obtidos, quando é são usadas as combinações entre as várias segmentações, estão apresentados na tabela 5.10. São ainda apresentadas as variações percentuais entre os resultados obtidos por cada uma das combinações de segmentação e os resultados da segmentação AVG. A coluna Δ_1 apresenta a variação percentual entre os resultados da segmentação AVG e os resultados da segmentação COMB1, enquanto que a coluna Δ_2 apresenta a variação percentual entre os resultados da segmentação AVG e os resultados da segmentação combinada COMB2. Da mesma forma, a coluna Δ_3 apresenta a variação percentual entre os resultados da segmentação AVG e os resultados da segmentação COMB3.

Tabela 5.10: Comparação dos resultados do seguimento obtidos pela combinação entre segmentações e os resultados obtidos partindo da segmentação AVG.

(a) Para a sequência OSOW1.

	AVG	COMB 1	Δ_1	COMB 2	Δ_2
Recall (R)	0,497	0,491	-1%	0,544	10%
Precision (P)	0,647	0,698	8%	0,686	6%
F-Score (FS)	0,562	0,576	3%	0,607	8%
False Alarme Rate (FAR)	0,353	0,302	-14%	0,314	-11%

(b) Para a sequência OSOW2.

	AVG	COMB3	Δ_3
Recall (R)	0,516	0,519	0%
Precision (P)	0,582	0,611	5%
F-Score (FS)	0,547	0,561	3%
False Alarme Rate (FAR)	0,418	0,389	-7%

Como se percebe pela análise da tabela 5.10, houve uma melhoria generalizada na grande parte das medidas apresentadas. Destaca-se a redução na medida *False Alarm Rate* na ordem dos 14% para a segmentação combinada COMB1, de 11% para a segmentação combinada COMB2 e de 7% para a segmentação combinada COMB3, esta referente à sequência OSOW2. Além disso, nota-se também uma melhoria significativa na medida *Precision*, 8% para a COMB1, 6% para a COMB2 e 5% para a COMB3. Interessante é também o facto de a segmentação combinada COMB2 obter melhores resultados do que a segmentação combinada COMB1 nas maioria das medidas. Este facto é explicado pelo menor número de alterações entre os métodos de segmentação que leva a uma maior consistência na segmentação de cada objeto. Quanto maior o número de alterações maior será possibilidade de perda de *tracks* devido a alterações bruscas na segmentação. Conclui-se portanto que a alteração da segmentação é possível e obtém melhorias nos resultados, quando a escolha da mesma é feita através de medidas baseadas em informação de referência, neste caso *symetric partition-distance*.

Relativamente ao tempo de processamento, os resultados estão apresentados na tabela 5.11. Destaca-se o aumento do tempo de processamento total, apesar da diminuição de 12% no tempo de

Tabela 5.11: Comparação do tempo total de processamento, em segundos, do algoritmo de seguimento usando a segmentação AVG e as segmentações combinadas.

(a) Para a sequência OSOW1.

	AVG	COMB 1	Δ_1	COMB 2	Δ_2
Tempo de segmentação total (s)	93,57	127,64	36%	82,53	-12%
Tempo de processamento total (s)	656,06	674,53	3%	738,35	13%
Tempo total (s)	749,63	802,17	7%	820,87	10%
FPS	1,84	1,72	-7%	1,68	-9%

(b) Para a sequência OSOW2.

	AVG	COMB3	Δ_3
Tempo de segmentação total (s)	99,348	125,947	27%
Tempo de processamento total (s)	984,426	990,308	1%
Tempo total (s)	1083,774	1116,255	3%
FPS	1,35	1,31	-3%

segmentação, na ordem dos 13%, quando se usa a segmentação combinada COMB2. Isto deve-se ao aumento do número de objetos a seguir, facto que pode ser verificado na aumento das medidas de desempenho *Precision* e *Recall*. No que diz respeito à combinação COMB1, o aumento de 7% no tempo de processamento total é explicado pelo aumento de 36% no tempo necessário para a segmentação das imagens que constituem a referida combinação de segmentações. Relativamente à sequência OSOW2, combinação COMB3, o tempo total de processamento aumentou em 3% devido ao aumento de 27% do tempo de segmentação.

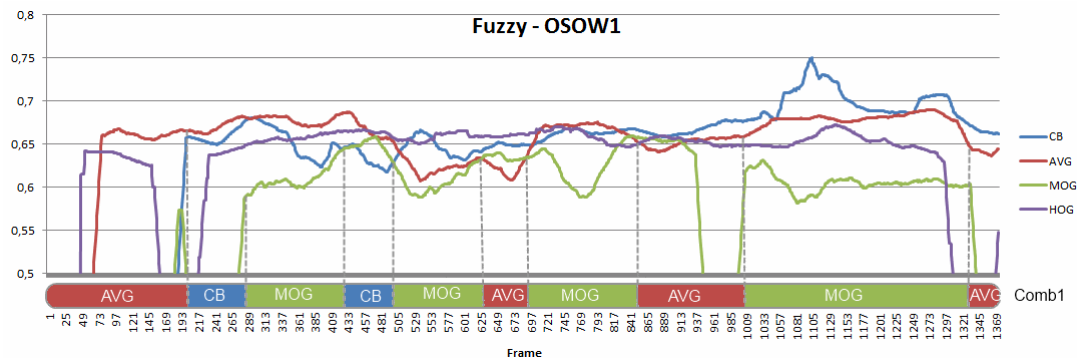
Esta análise não pode ser feita para a sequência PETS, uma vez que não fornecida informação relativa aos resultados ideais do seguimento. Assim, além de não ser possível o cálculo das medidas baseadas em *bounding-box* também não é possível o cálculo da medida *symetric partition-distance* para a totalidade da sequência.

5.4.2 Alteração automática

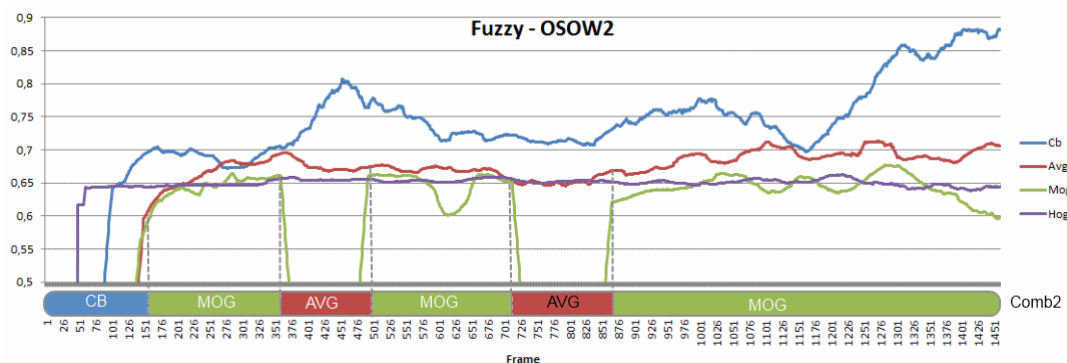
A análise feita na secção anterior permitiu concluir que, caso se disponha informação de referência relativamente a uma sequência de imagens, é possível obter uma melhoria dos resultados do seguimento através da escolha, em cada *frame*, do melhor método de segmentação. Nesta secção pretende-se verificar se é possível obter melhorias usando apenas informação recolhida pelo algoritmo de seguimento, usando medidas de avaliação sem recurso a informação de referência.

Em primeiro lugar é necessário avaliar a qualidade do seguimento e, baseados nesta informação decidir se deve ou não ser alterado o método de segmentação ou de deteção. A medida implementada para o efeito é a apresentada por Erdem em [14], *fuzzy*. Calculando o valor desta medida para as sequências de teste, partindo de cada método de segmentação definido, é possível

fazer uma comparação entre os resultados obtidos. O gráfico com a evolução da medida *fuzzy*, para as sequências OSOW1 e OSOW2, é apresentado na figura 5.7.



(a) para a sequência OSOW1.



(b) para a sequência OSOW2.

Figura 5.7: Evolução da medida *fuzzy*, na avaliação da qualidade do seguimento, ao longo das sequências de teste, partindo de cada um dos métodos de segmentação definidos e partindo do detetor HOG.

De forma a evitar constantes alterações no método de segmentação, os valores obtidos pela medida *fuzzy* foram filtrados usando para cada *frame* o valor médio dos últimos 50 *frames*. Tratando-se esta medida de uma medida de erro, os valores mais reduzidos corresponderão a um melhor seguimento sendo que a medida toma o valor 0 (zero) quando nenhum objeto está a ser seguido. Ou seja, a medida poderá apresentar um valor 0 (não está a ser seguido qualquer objeto), quando na verdade estão vários objetos na cena. Para cada frame da sequência foi escolhido o método de segmentação que obteve o menor valor de erro, excluindo os métodos de segmentação que apresentem o valor zero nesta medida. Analisando o gráfico traçado na figura 5.7b, consegue-se perceber claramente que o método de segmentação que provoca maior erro nesta medida é o CB. Esta conclusão está de acordo com a retirada na secção anterior relativamente à medida *partition-distance*, apresentada na figura 5.6b. Apesar disto, as conclusões retiradas relativamente aos restantes métodos de segmentação não é consensual entre a medida *fuzzy*, menor erro para o método MOG, e *partition-distance*, menor erro para o método AVG.

A comparação dos resultados está apresentada na tabela 5.12. A coluna AVG corresponde aos resultados obtidos pelo que foi considerada a melhor segmentação para a sequência OSOW1, pela medida F-Score, enquanto que a coluna COMB2 diz respeito aos melhores resultados obtidos

quando a escolha foi feita pela medida *symetric partition-distance*. Δ_{avg} e Δ_{comb2} é a variação percentual entre os resultados da coluna fuzzy e a coluna AVG e entre a coluna fuzzy e a coluna COMB2, respetivamente. Desta forma é também feita a comparação entre os resultados obtidos pela combinação de segmentações dada através da medida baseada em informação de referência e os resultados obtidos pela combinação de segmentações dada pela medida que não se baseia em informação de referência.

Tabela 5.12: Comparação dos resultados do seguimento para as sequências de teste entre os resultados da segmentação AVG, da segmentação combinada pela medida *symetric partition-distance* e da segmentação combinada pela medida *fuzzy*

(a) Para a sequência OSOW1.						
	AVG	fuzzy	Δ_{avg}	COMB2	fuzzy	Δ_{comb2}
Recall (R)	0,497	0,351	-29%	0,544	0,351	-35%
Precision (P)	0,647	0,529	-18%	0,686	0,529	-23%
F-Score (FS)	0,562	0,422	-25%	0,607	0,422	-30%
False Alarme Rate (FAR)	0,353	0,471	34%	0,314	0,471	50%

(b) Para a sequência OSOW2.						
	AVG	fuzzy	Δ_{avg}	COMB3	fuzzy	Δ_{comb3}
Recall (R)	0,516	0,371	-28%	0,519	0,371	-29%
Precision (P)	0,582	0,695	19%	0,611	0,695	14%
F-Score (FS)	0,547	0,483	-12%	0,561	0,483	-14%
False Alarme Rate (FAR)	0,418	0,305	-27%	0,389	0,305	-22%

Analisando os valores obtidos para a sequência OSOW1, tabela 5.12, verifica-se que não houve melhoria nos resultados do seguimento quando a decisão sobre o melhor método de segmentação é baseada nos resultados da medida *fuzzy*. Pelo contrário, houve uma quebra acentuada dos resultados relativamente aos resultados obtidos pelo melhor método AVG. De realçar a diminuição de 29% e 18% nas medidas *Recall* e *Precision*, respetivamente, além do aumento de 34% nas falsas deteções. Quando a comparação é feita com a combinação de segmentações dada pela medida *symetric partition-distance*, a variações nos resultados é ainda maior uma vez que os resultados obtidos pela combinação COMB2 são superiores aos obtidos pela segmentação dada pelo método AVG.

Também na sequência OSOW2 se verifica uma quebra nos resultados do seguimento. Relativamente à medida *Recall* houve uma diminuição de 28% e um aumento de 30% nas falsas deteções, quando comparados com os resultados obtidos pelo de segmentação AVG. Apesar destes valores nota-se uma melhoria na medida *Precision* de 19% e uma diminuição de 27% no número de falsas deteções. Estes valores são explicados pelo elevado número de *frames* onde é usada a segmentação dada pelo método de segmentação MOG e, como se pode constatar pela análise da tabela 5.5b, este método de segmentação é o que leva a valores mais elevados na medida *Precision* e a valores mais reduzidos na medida *Recall*.

Concluí-se assim que uma tomada de decisão baseada na comparação dos resultados obtidos

pela medida *fuzzy* não leva a melhorias nos resultados do seguimento, para as sequências testadas, ao contrário do que acontecia com a medida baseada em informação de referência, *symetric partition-distance*.

Esta análise foi importante de forma a avaliar a relação existente entre a medida *fuzzy* e os resultados finais do algoritmo de seguimento. Contudo, esta análise baseou-se na comparação manual dos resultados obtidos usando esta medida, para os vários métodos de segmentação disponíveis. Quando se pretende que seja o algoritmo de seguimento a avaliar os resultados da medida *fuzzy*, os resultados obtidos pelos outros métodos de segmentação não estão disponíveis. Terá de ser tomada uma decisão com base apenas nos resultados obtidos para o método de segmentação que está a ser usado. Assim, terá de ser definido um valor limite abaixo do qual se consideraria que o seguimento está a obter bons resultados. A definição deste valor limite não é possível uma vez que não conseguimos afirmar que determinado valor na medida *fuzzy* corresponde a um bom seguimento ou a um mau seguimento. Um valor de 0 (zero) nesta medida poderá corresponder a um bom seguimento quando nenhum objeto está presente na imagem ou significar um mau seguimento quando estão objetos na imagem e nenhum está a ser seguido. Um exemplo disto mesmo pode ser visível na figura 5.8.

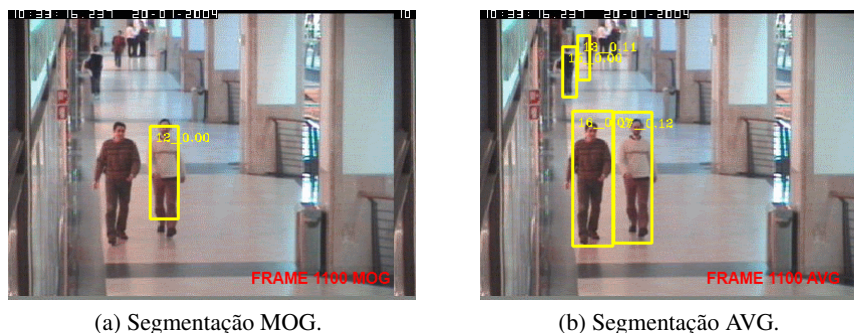


Figura 5.8: *bounding-box* do resultado do seguimento no *frame* 1100 da sequência OSOW1, partindo da segmentação MOG e da segmentação AVG, aplicado na imagem original.

A figura 5.8 mostra o resultado do seguimento para o *frame* 1100, quando a segmentação é obtida pelo método MOG (5.8a) e quando a segmentação é dada pelo método AVG (5.8b). Na figura 5.8a apenas um objeto está a ser seguido enquanto que na 5.8b três objetos estão a ser seguidos corretamente. Analisando agora a figura 5.7, referente aos resultados da medida *fuzzy*, verifica-se que, para o *frame* 1100, o método MOG apresenta um menor valor de erro comparativamente com o método AVG. Conclui-se então que, apesar de a medida *fuzzy* apresentar um menor valor de erro para o método de segmentação MOG, isso não significa necessariamente que esse mesmo método de segmentação esteja a obter resultados superiores aos que seriam obtidos pelo método AVG. Um outro exemplo é apresentado na figura 5.9. Esta imagem mostra que, para o mesmo método de segmentação, neste caso AVG, o mesmo valor de erro na medida *fuzzy* pode corresponder a diferentes resultados no seguimento. Os *frames* 440 e 750 e os *frames* 530 e 670 apresentam valores semelhantes na medida *fuzzy*, mas produzem resultados distintos no seguimento. Os melhores

resultados no seguimento são obtidos nos *frames* 440 e 670 enquanto que os piores resultados são obtidos nos *frames* 530 e 750.

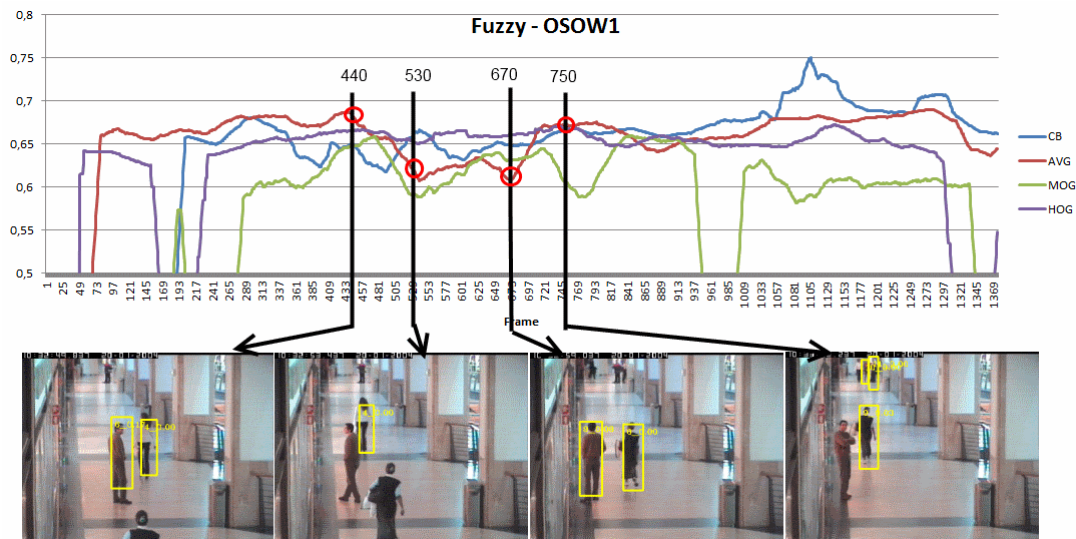


Figura 5.9: *bounding-box* do resultado do seguimento, partindo da segmentação AVG, aplicado na imagem original, nos *frames* 440, 530, 670 e 750 da sequência OSOW1.

Depois de analisados todos os resultados obtidos, é possível retirar várias conclusões importantes relativamente à alteração automática do método de segmentação. Conclui-se que a decisão, baseada nos resultados obtidos pela medida *fuzzy*, relativa ao melhor método de segmentação pode ser tomada quando se conhecem os resultados da mesma medida para todos os métodos de segmentação. Desta forma é possível elaborar um *ranking* relativamente ao método de segmentação que produz um erro inferior na medida *fuzzy*. Apesar disto, a decisão não leva à obtenção de melhorias nos resultados do seguimento, nas sequências de teste usadas. Em segundo lugar verificou-se também não ser possível estabelecer uma valor limiar, acima do qual o algoritmo de seguimento considera que os resultados obtidos ao nível do seguimento não são satisfatórios. A definição deste limiar é essencial uma vez que, durante o processamento, o algoritmo de seguimento não tem informação relativa aos restantes métodos de segmentação, não sendo possível efetuar uma comparação de resultados. Verificou-se que um determinado valor na medida *fuzzy* pode corresponder a bons ou maus resultados ao nível do seguimento. Conclui-se portanto não ser possível a implementação de um mecanismo automático de alteração do método de segmentação baseado na medida sem informação de referência *fuzzy*.

5.5 Redução do tempo de processamento

Como já foi explicado anteriormente, a redução do tempo de processamento será tentada recorrendo a um método referido em 4.3.3 que, de acordo com algumas condições previamente definidas, dá ao algoritmo de seguimento a possibilidade de não efetuar a procura pela melhor correspondência nas imagens segmentadas. Ao não ser feita esta procura, o algoritmo assume a

posição estimada pelo filtro de *Kalman* como a posição correta do objeto. As condições definidas para que se possa ignorar a procura pela melhor correspondência são relativas à quantidade de movimento, número de objetos, distância entre eles e grau de confiança dado pelo filtro de *Kalman* às observações.

De forma a podermos analisar os resultados obtidos será feita uma comparação com os resultados do algoritmo base. O algoritmo base corresponde ao algoritmo de seguimento inicial, sem qualquer medida implementada, tendo como segmentação a que melhores resultados produz no seguimento, relativamente à medida *F-Score*, método AVG. Os resultados serão então comparados ao nível do tempo de processamento e ao nível da qualidade do seguimento.

modo:USkip

Foram definidos vários modos de operação que apresentam combinações de algumas das condições referidas. Todos os modos definidos partem da mesma segmentação usada no algoritmo base. O modo mais simples USkip (*unassisted skip*) ignora a procura pela melhor correspondência em todos os objetos *frames* alternados. Isto significa que em cada dois *frames* não é feita a procura da melhor correspondência, para todos os objetos presentes, num deles.

Para a sequência OSOW1 os melhores resultados no seguimento foram obtidos pelo método de segmentação AVG, sendo também usada esta segmentação neste modo. Os resultados obtidos no seguimento por este modo estão apresentados na tabela 5.13. Como se trata de um método que visa, em primeiro lugar, a diminuição do tempo de processamento, serão os resultados temporais os mais importantes. Contudo, também se pretende que os resultados do seguimento não se deteriorem demasiado. A nível temporal, os resultados estão apresentados na tabela 5.14.

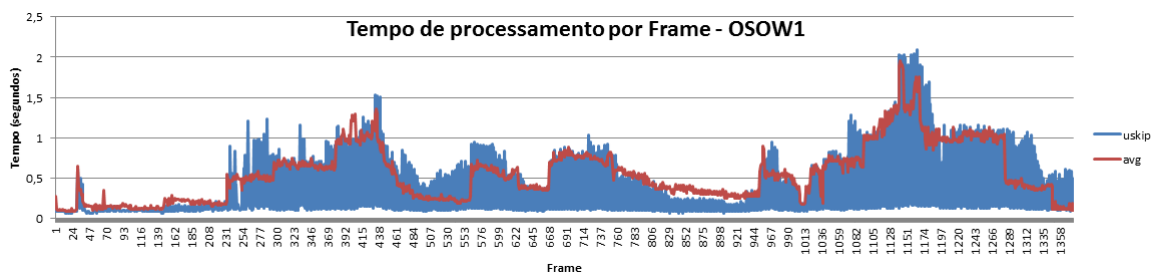
Tabela 5.13: Comparação entre algoritmo base e modo USkip, para a sequência OSOW1.

	OSOW1			OSOW2		
	AVG	USkip	Δ_{USkip}	AVG	USkip	Δ_{USkip}
Recall (R)	0,497	0,446	-10%	0,516	0,403	-22%
Precision (P)	0,647	0,568	-12%	0,582	0,509	-12%
F-Score (FS)	0,562	0,500	-11%	0,547	0,450	-18%
False Alarme Rate (FAR)	0,353	0,432	22%	0,418	0,491	17%

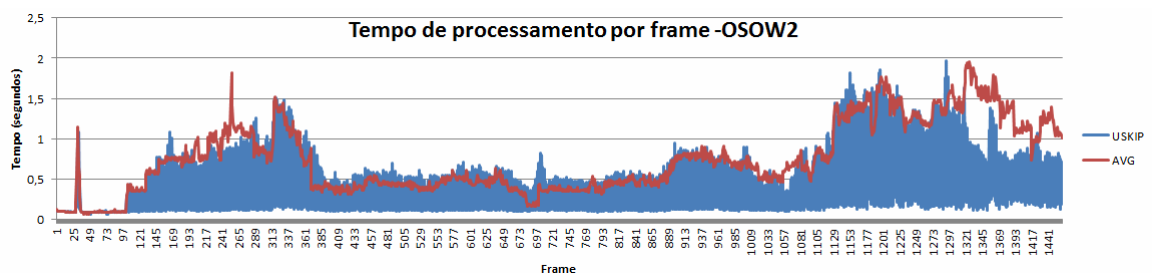
Tabela 5.14: Variação do tempo de *match* e do tempo total, em segundos, para as sequências de teste, no modo USkip.

	OSOW1			OSOW2		
	AVG	USkip	Δ_{USkip}	AVG	USkip	Δ_{USkip}
Match Time (s)	592	369	-38%	907	467	-49%
Total Time (s)	753	518	-31%	1084	631	-42%

Analisando os resultados obtidos relativamente ao tempo de processamento, tabela 5.14, verifica-se, como esperado, uma redução quer o tempo de procura pela melhor correspondência quer no tempo total de processamento do algoritmo de seguimento, relativamente ao algoritmo base. A redução foi de 38% no tempo dispendido pelo algoritmo na procura da melhor correspondência na imagem e de 31% no tempo total de processamento da sequência OSOW1. Para a sequência OSOW2 as melhorias foram de 49% no tempo de procura pela melhor correspondência e de 42% no tempo total de processamento. No que diz respeito à qualidade do seguimento, é de notar uma redução na medida *Recall* (-10%) e *Precision* (-12%) além de um aumento na medida *False Negative Rate* de 10%. Para a sequência OSOW2, os resultados foram ainda mais penalizados com uma redução de 22% na medida *Recall*, de 12% na medida *Precision* e de um aumento no número de falsas deteções de 17%.



(a) Ao longo da sequência OSOW1.



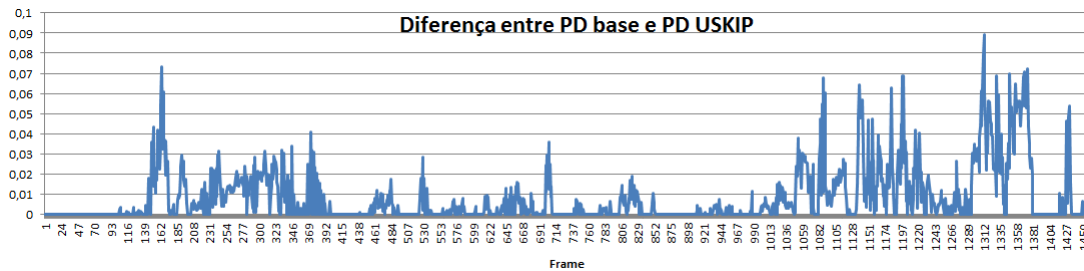
(b) Ao longo da sequência OSOW1.

Figura 5.10: Evolução do tempo de processamento por *frame*, em segundos, no algoritmo base e no modo USkip.

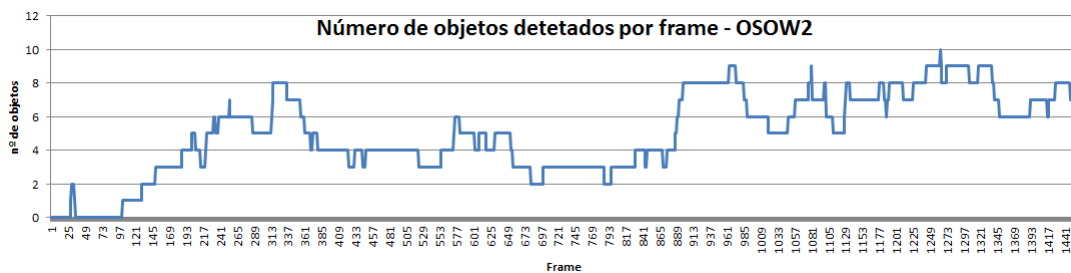
Os gráficos da figura 5.10 mostram a evolução do tempo de processamento total ao longo da sequência OSOW1 e OSOW2 para o modo USkip, em comparação com o algoritmo base. Como se percebe pela sua análise, os ganhos verificados no tempo de processamento são distribuídos ao longo de toda a sequência. Este modo de processamento não reduz significativamente o tempo de processamento em cada *frame*.

De forma a verificar em que *frames* os resultados foram mais prejudicados, foi traçado o gráfico da figura 5.11a que mostra a diferença entre os valores obtidos pela medida *symetric partition distance*, calculados para o resultado do seguimento do algoritmo base, e os valores obtidos pela mesma medida para os resultados do seguimento, obtidos no modo USkip. Comparando os gráficos da figura 5.11 verifica-se que os erros ocorrem principalmente nos *frames* onde o número de objetos é elevado. Assim, de forma a melhorar os resultados do seguimento, fará sentido incluir

a informação relativa ao número de objetos ou a distância entre os mesmos como uma restrição à não procura pela melhor correspondência.



(a) Diferença dos valores obtidos pela medida *symetric partition distance* entre os resultados obtidos pelo algoritmo base e os resultados obtidos pelo modo USkip, na sequência OSOW2.



(b) Número de objetos detetados por *frame* na sequência OSOW2.

Figura 5.11: Comparação entre a medida *symetric partition-distance* e o número de objetos por *frame*.

modo:ASkip

De modo a melhorar os resultados obtido pelo modo USkip, foi definido um novo modo, ASkip, onde a informação relativa ao número de objetos presentes e a distância ao mais próximo é tida em conta. Além disso a decisão de efetuar ou não a procura pela melhor correspondência é tomada objeto a objeto e não *frame a frame* como no modo anterior. Este modo está explicado com maior detalhe na secção 4.3.3.

Os resultados obtidos por este modo estão apresentados na tabela 5.15 e na tabela 5.16. Relativamente ao tempo de procura pela melhor correspondência e ao tempo de processamento total, os resultados apresentados nas tabelas 5.16a e 5.16b pioram relativamente ao modo USkip. Verificou-se um aumento de 43% e de 77% no tempo de *match* e de 29% e 58% no tempo total de processamento, para as sequências OSOW1 e OSOW2, respetivamente. Relativamente ao algoritmo base estes resultados melhoraram em 11% e 9% para o tempo de *match* e em 11% e 8% no tempo total de processamento, para as sequências OSOW1 e OSOW2, respetivamente. Como seria de esperar, os resultados ao nível do tempo de processamento foram prejudicados em relação ao modo USkip, uma vez que este é um modo mais restritivo relativamente à não procura pela melhor correspondência. Apesar disso, os resultados foram melhorados relativamente ao algoritmo base, uma vez que existem objetos onde a procura pela melhor correspondência foi ignorada.

Tabela 5.15: Comparação dos resultados do seguimento entre algoritmo base (segmentação pelo método AVG) e modo ASkip, para as sequências de teste.

(a) Para a sequência OSOW1.

	AVG	ASkip	Δ_{AVG}	USkip	ASkip	Δ_{USkip}
Recall (R)	0,497	0,445	-10%	0,446	0,445	0%
Precision (P)	0,647	0,620	-4%	0,568	0,620	9%
F-Score (FS)	0,562	0,518	-8%	0,500	0,518	4%
False Alarme Rate (FAR)	0,353	0,380	8%	0,432	0,380	-12%

(b) Para a sequência OSOW2.

	AVG	ASkip	Δ_{AVG}	USkip	ASkip	Δ_{USkip}
Recall (R)	0,516	0,494	-4%	0,403	0,494	23%
Precision (P)	0,582	0,560	-4%	0,509	0,560	10%
F-Score (FS)	0,547	0,525	-4%	0,450	0,525	17%
False Alarme Rate (FAR)	0,418	0,440	5%	0,491	0,440	-10%

Relativamente à qualidade dos resultados do seguimento, na sequência OSOW1 verifica-se um aumento de 9% na medida *Precision* e um aumento de 12% na medida *False Alarm Rate*, relativamente aos resultados obtidos no modo USkip, tabela 5.15a. Quando comparados com o algoritmo base (segmentação AVG) os resultados do seguimento saem prejudicados em todas as medidas calculadas. Na sequência OSOW2 seguem a mesma tendência dos resultados da sequência OSOW1. Verifica-se que os resultados apresentados na tabela 5.15b piorarem relativamente aos resultados obtidos pelo algoritmo base, mas melhoram quando comparados com os resultados obtidos no modo USkip.

Tabela 5.16: Resultados ao nível do tempo de processamento para o algoritmo de seguimento base (segmentação AVG) e para o modo ASkip.

(a) Para a sequência OSOW1.

	AVG	ASkip	Δ_{AVG}	USkip	ASkip	Δ_{USkip}
Match Time (s)	592	526	-11%	369	526	43%
Total Time (s)	753	668	-11%	518	668	29%

(b) Para a sequência OSOW2.

	AVG	ASkip	Δ_{AVG}	USkip	ASkip	Δ_{USkip}
Match Time (s)	907	828	-9%	467	828	77%
Total Time (s)	1084	997	-8%	631	997	58%

Tabela 5.17: Comparação entre os resultados do algoritmo base (segmentação CB) e o resultados no modo ASkip (segmentação CB).

	OSOW1			OSOW2		
	CB	ASkip	Δ_{ASkip1}	CB	ASkip	Δ_{ASkip2}
Recall (R)	0,350	0,482	38%	0,505	0,606	20%
Precision (P)	0,525	0,546	4%	0,217	0,274	26%
F-Score (FS)	0,420	0,512	22%	0,304	0,377	24%
False Alarme Rate (FAR)	0,475	0,454	-4%	0,783	0,726	-7%

Tabela 5.18: Resultados ao nível do tempo de processamento para o algoritmo de seguimento base(segmentação CB) e para o modo ASkip.

	OSOW1			OSOW2		
	CB	ASkip	Δ_{CB}	CB	ASkip	Δ_{CB}
Match Time (s)	928	662	-29%	2133	1874	-12%
Total Time (s)	1045	755	-28%	2320	2087	-10%

Verificou-se também que, para segmentações menos corretas, como por exemplo a segmentação dada pelo método CB, este modo obteve uma melhoria significativa na qualidade do seguimento aliado a uma melhoria no tempo de processamento. Na tabela 5.17, está apresentada a comparação dos resultados do seguimento quando, tanto o algoritmo base como o modo ASkip, é usada a segmentação dada pelo método CB. Este modo obteve uma melhoria nas medidas *Recall* e *Precision* de 38% e 4%, respetivamente, para a sequência OSOW1 e de 20% e 26%, respetivamente, para a sequência OSOW2. Além disso verificou-se também uma diminuição do número de falsas deteções, 4% para a sequência OSOW1 e 7% para a sequência OSOW2. Estes valores podem significar que, para determinados *frames* onde a segmentação apresenta demasiado ruído, a estimação da posição pelo filtro de *Kalman* é mais correta do que a procura na imagem segmentada pela melhor correspondência.

Relativamente ao tempo de processamento verifica-se uma diminuição do tempo de procura pela melhor correspondência e do tempo total de processamento, relativamente ao algoritmo base. Neste caso o algoritmo base usa a segmentação dada pelo método CB. Apesar da melhoria no tempo de processamento e da melhoria dos resultados do seguimento, este método de segmentação leva a resultados inferiores aos registados pelo método de segmentação AVG. Este teste pretendeu demonstrar que é possível obter uma melhoria nos resultados do seguimento obtendo também uma melhoria do tempo de processamento.

Conclui-se que o modo ASkip obteve algumas melhorias nos resultados do seguimento, à custa de alguma diminuição no ganho em termos de tempo de processamento, relativamente ao modo ASkip. Relativamente aos resultados obtidos pelo o algoritmo base os resultados do seguimento pioraram ligeiramente (excepção feita aos resultados obtidos pelo método de segmentação

CB). Assim, com vista a uma melhoria dos resultados do seguimento comparativamente com os resultados do algoritmo base, foi implementado e testado o modo CSkip cujos resultados são apresentados em seguida.

modo:CSkip

Apesar de alguma melhoria na qualidade do seguimento obtido no modo ASkip, relativamente ao modo USkip, o facto é que relativamente ao algoritmo base os resultados pioraram. Assim, foram adicionados ao modo anterior mais algumas limitações de forma a tentar minimizar essa perda nos resultados. Neste modo foram adicionadas informações relativas aos valores da covariância do erro da observação, dada pelo filtro de *Kalman*. Quando a covariância do erro se afasta de 0 (zero) significa que o filtro de *Kalman* tem vindo a considerar pouco fiáveis as observações feitas anteriormente. Isto pode significar que o objeto apresenta um movimento pouco regular, pelo que nestes casos será sempre feita uma procura pela melhor correspondência e não apenas confiar na estimação. Quando os valores da covariância são muito próximo de 0 (zero) significa que o filtro de *Kalman* considera as observações como fiáveis, o que pode significar que o objeto, num passado recente, teve um comportamento bem definido. A informação relativa ao *optical flow* na vizinhança de cada objeto é outro fator que foi analisado na decisão relativa à procura ou não pela melhor correspondência.

Foram definidos valores limite tanto para a distância ao objeto mais próximo como para o valor do módulo *optical flow* na vizinhança. O valor a partir do qual se considera que existe um objeto na vizinhança de um outro é o valor referente à diagonal da *bounding-box* do respetivo objeto. No que diz respeito ao valor do módulo do *optical flow*, o limiar foi escolhido depois de uma análise aos valores obtidos nesta medida, figura 5.13. Quanto à covariância do erro o valor limite escolhido foi um valor relativamente próximo de 0 (zero), figura 5.12. Analisando o gráfico da figura 5.12, verificou-se que sempre que os picos existentes no traçado correspondem a objetos ocultos ou na iminência do seu seguimento ser perdido. Assim, foram escolhido para valores limiar os valores indicados pelo linha preta na imagem, 0.2 para a covariância do erro e 2 para o módulo do *optical flow* na vizinhança do objeto. Sempre que os valores ultrapassem os limites definidos considera-se que não estão reunidas as condições para ser ignorada a pesquisa pela melhor correspondência.

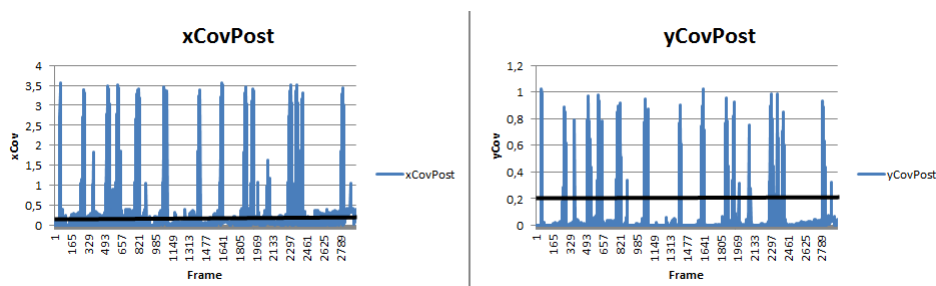


Figura 5.12: Valores para a covariância do erro, dados pelo filtro de *Kalman*, para o algoritmo base, na sequência OSOW1.

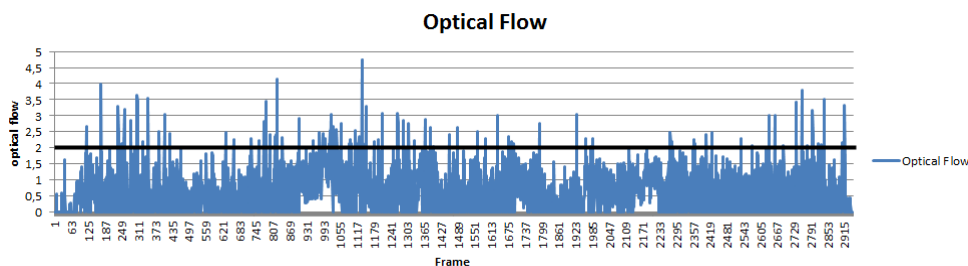


Figura 5.13: *Optical flow* calculado na vizinhança de cada objeto para o algoritmo base, na sequência OSOW1.

Tabela 5.19: Comparação dos resultados do seguimento entre algoritmo base (segmentação pelo método AVG) e modo ASkip, para as sequências de teste.

(a) Para a sequência OSOW1.						
	AVG	CSkip	Δ_{AVG}	ASkip	CSkip	Δ_{ASkip}
Recall (R)	0,497	0,493	-1%	0,445	0,493	11%
Precision (P)	0,647	0,682	5%	0,620	0,682	10%
F-Score (FS)	0,562	0,572	2%	0,518	0,572	10%
False Alarme Rate (FAR)	0,353	0,318	-10%	0,380	0,318	-16%
(b) Para a sequência OSOW2.						
	AVG	CSkip	Δ_{AVG}	ASkip	CSkip	Δ_{ASkip}
Recall (R)	0,516	0,450	-13%	0,494	0,450	-9%
Precision (P)	0,582	0,513	-12%	0,560	0,513	-8%
F-Score (FS)	0,547	0,480	-12%	0,525	0,480	-9%
False Alarme Rate (FAR)	0,418	0,487	16%	0,440	0,487	11%

Os resultados obtido neste modo, para a sequência OSOW1, estão apresentados nas tabelas 5.19 e 5.20.

Como se comprova pela análise da tabela 5.19a, os resultados do seguimento para a sequência OSOW1 foram melhorados, relativamente ao modo ASkip. A medida *Recall* obteve uma melhoria de 11% e a medida *Precision* obteve uma melhoria de 10%. Além disso, o número de falsas deteções foi reduzido em 16%. Relativamente ao algoritmo base, também foi verificada uma melhoria dos resultados do seguimento. A medida *Precision* teve um aumento de 5% tendo a medida *Recall* sofrido uma redução de 1%. No que diz respeito à sequência OSOW2, os resultados estão apresentados na tabela 5.19b. Para esta sequência, os resultados não acompanharam as melhorias obtidas para a sequência OSOW1. Pelo contrário, os resultados foram prejudicados, relativamente ao modo ASkip. De realçar uma diminuição de 9% na medida *Recall*, de 8% na medida *Precision*, relativamente ao modo ASkip.

Analisando os resultados obtidos para o tempo de processamento (tabela 5.20a), foram registadas perdas, relativamente ao modo ASkip, na sequência OSOW1. Este era um resultado já

Tabela 5.20: Resultados ao nível do tempo de processamento, em segundos, para o algoritmo de seguimento base (segmentação AVG) e para o modo CSkip.

(a) Para a sequência OSOW1.						
	AVG	CSkip	Δ_{AVG}	ASkip	CSkip	Δ_{ASkip}
Match Time (s)	592	575	-3%	526	575	9%
Total Time (s)	753	812	8%	668	812	22%

(b) Para a sequência OSOW2.						
	AVG	CSkip	Δ_{AVG}	ASkip	CSkip	Δ_{ASkip}
Match Time (s)	907	823	-9%	828	823	-1%
Total Time (s)	1084	997	-9%	997	997	0%

esperado, uma vez que este modo é mais rígido no que diz respeito às condições para que seja ignorada a pesquisa pela melhor correspondência. Foi registado um aumento de 9% no tempo necessário para efetuar a procura pela melhor correspondência e uma aumento de 22% no tempo de processamento total. Comparativamente com o algoritmo base o tempo de *match* foi reduzido em 3% mas registou-se um aumento no tempo de processamento total de 8% devido à necessidade de cálculo do *optical flow*.

Para a sequência OSOW2, tabela 5.20b, os resultados são semelhantes aos obtidos no modo ASkip. Apesar de o número de objetos processados ter sido menor, facto comprovado pela diminuição do valor da medida *Recall*, o facto de este modo requerer o cálculo do *optical flow* faz com que o tempo total se mantenha praticamente inalterado, quando comparado com o modo ASkip. Comparativamente como os resultados obtidos pelo algoritmo base tanto o tempo necessário à procura pela melhor correspondência como o tempo total de processamento foram reduzidos em 9%.

Os resultados obtidos mostram que a modo CSkip precisa de ser ajustado, nomeadamente no que diz respeito aos valores limiar para a covariância do erro e para o nível de *optical flow*, de forma a maximizar o número de objetos aos quais não é feita a procura pela melhor correspondência minimizando a diminuição da qualidade do seguimento. Concluí-se também que o facto de serem adicionadas mais restrições à não procura pela melhor correspondência reduz os ganhos em termos temporais. Uma vez que estes testes visam, em primeiro lugar a melhoria dos tempos obtidos para o seguimento, não foram feitos testes exaustivos relativamente aos melhores valores dos limiares a escolher. Apesar disto, pretende-se verificar se é possível obter uma melhoria na qualidade do seguimento.

Analisando os resultados obtido no diferentes modos, tanto em termos temporais como em termos da qualidade do seguimento, e comparando-os com os resultados obtidos pelo algoritmo base, verificou-se ser possível reduzir o tempo de processamento por *frame*, reduzindo o número de pesquisas pela melhor correspondência, sem prejudicar, em larga medida, os resultados do seguimento.

Capítulo 6

Conclusões e trabalho futuro

6.1 Conclusões

Esta dissertação focou-se na análise da variação dinâmica da complexidade de um algoritmo de seguimento. Em particular, foram estudados métodos com o objetivo de reduzir o tempo de processamento por *frame* e métodos para permitir obter uma melhoria na qualidade do seguimento. Estes métodos usam informações relativas ao processo de seguimento e informações relativas à cena de forma a poder adaptar a complexidade do algoritmo. O impacto da variação da complexidade do algoritmo de seguimento nos resultados finais foi depois avaliado, sendo neste capítulo apresentadas as conclusões retiradas. A figura 6.1 mostra o processo de avaliação dos resultados obtidos pelo algoritmo de seguimento.

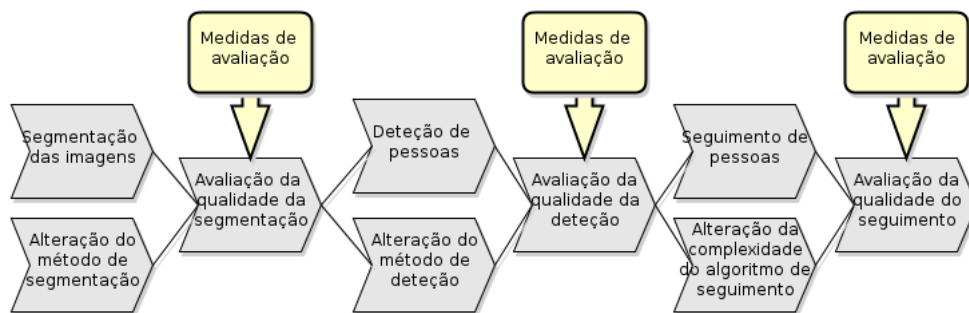


Figura 6.1: Diagrama ilustrativo do processo de avaliação dos resultados obtidos pelos diferentes módulos do algoritmo de seguimento.

Com as experiências realizadas verificou-se a baixa correlação existente entre as medidas implementadas que não se baseiam em informação de referência com as medidas baseadas neste tipo de informação. As medidas não baseadas em informação de referência produzem elevado ruído, pelo que a avaliação que fazem é por vezes incorreta. Além disso, não foi possível fazer uma avaliação quantitativa dos métodos que as medidas visam avaliar sem o recurso à comparação com algum tipo de informação. Apenas podemos dizer se determinado método produz ou não bons resultados através da comparação com os resultados de diferentes métodos.

Relativamente aos resultados obtidos pelos métodos de segmentação implementados, verificou-se não existir uma relação direta entre qualidade da segmentação e peso computacional. Nas sequências de teste analisadas nem sempre o método de segmentação mais dispendioso, em termos de tempo de processamento, produziu a melhor segmentação.

No que diz respeito à deteção de pessoas verificou-se que a implementação do detetor HOG utilizada produz bons resultados nas sequências do *dataset* Caviar, mas maus resultados nas sequências do *dataset* PETS. Será necessário treinar o detetor com imagens retiradas deste *dataset* para que o mesmo produza resultados aceitáveis. Verificou-se também que, para as sequências do *dataset* Caviar, o detetor HOG apresenta um tempo de deteção que é sempre superior ao tempo obtido pelo detetor baseado em segmentação (incluindo o tempo necessário à segmentação), apesar de se aproximar deste em situações de grande densidade de pessoas. Verificou-se também que os resultados obtidos pelo detetor de pessoas HOG ao nível do seguimento não refletem os resultados obtidos ao nível da deteção, sendo por isso necessário melhorar o método implementado no algoritmo que permite que a deteção seja feita pelo detetor HOG.

Verificou-se não ser possível a alteração do método de segmentação/deteção de pessoas durante o processamento das sequências de teste, tendo por base a avaliação feita apenas com medidas não baseadas em informação de referência (*fuzzy*). Não foi possível decidir se determinado valor obtido por cada medida implementada corresponde a bons ou maus resultados. Conclui-se também que, a alteração da segmentação durante o processamento é possível e obtém melhorias, quando a decisão é tomada com base em medidas baseadas em informação de referência, neste caso a medida *symetric partition distance*.

Relativamente ao método implementado que visa reduzir o tempo de processamento chegou-se à conclusão que a maior parte do esforço computacional exigido durante a execução do algoritmo de seguimento deve-se à procura nas imagens pela melhor correspondência. Assim, as experiências realizadas focaram-se na tentativa de redução do número de pesquisas. Foi implementado um método que, partindo de informações relativas aos objetos a serem seguidos, tomava uma decisão de forma autónoma quanto à procura ou não pela melhor correspondência. A análise dos resultados das experiências realizadas permite concluir que é possível reduzir o tempo de processamento das sequências de teste, à custa de alguma diminuição da qualidade dos resultados. Em alguns dos testes (tabela 5.17) verificou-se mesmo uma melhoria dos resultados do seguimento, acompanhados por uma diminuição do tempo de processamento, quando é ignorada a pesquisa pela melhor correspondência. A melhoria dos resultados deve-se ao fato da segmentação, quando a mesma é má, introduzir erros que serão minimizados pela não procura pela melhor correspondência.

Apesar de se ter verificado a impossibilidade de alteração do método de segmentação, usando apenas as medidas e métodos apresentados anteriormente, este trabalho permitiu concluir que é possível obter melhorias nos resultados do algoritmo de seguimento testado, principalmente ao nível do tempo de processamento, usando apenas medidas que não se baseiam em *Ground Truth*.

6.2 Trabalho futuro

O estudo feito nesta dissertação, apesar de obter alguns resultados conclusivos, pode ser continuado de forma a melhorar o desempenho dos métodos implementados.

A implementação de novas medidas, não baseadas em informação de referência, que estimem com maior precisão a qualidade dos métodos que visam avaliar, será um passo importante a dar na continuação deste trabalho. Uma vez que não foi possível a variação de forma automática do método de segmentação, será importante estudar métodos alternativos que consigam atingir este objetivo.

Apesar do detetor de pessoas HOG ter obtido bons resultados no que diz respeito à deteção, é necessário uma melhoria na fase do seguimento, de forma a tornar este detetor uma alternativa à deteção através de métodos baseados em segmentação.

Relativamente à diminuição do tempo de processamento, será importante um estudo mais aprofundado no que diz respeito às condições exigidas para que se possa ignorar a procura pela melhor correspondência. Será interessante também estudar a possibilidade de limitar o tempo de processamento por *frame*, de forma a manter, em situações mais complexas, o funcionamento em tempo real por parte do algoritmo de seguimento. O estudo e implementação de métodos que permitam ao algoritmo efetuar *buffering* das imagens, em situações onde o processamento é mais exigente em termos de tempo, será outro trabalho que poderá ser realizado no futuro.

Referências

- [1] Thomas B. Moeslund e Erik Granum. A survey of computer vision-based human motion capture. *Comput. Vis. Image Underst.*, 81:231–268, March 2001.
- [2] Zhao Tao e R Nevatia. Tracking multiple humans in complex situations. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(9):1208–1221, 2004.
- [3] Robert E. Schapire e Yoram Singer. Improved boosting algorithms using confidence-rated predictions. Em *Machine Learning*, páginas 80–91, 1999.
- [4] Paul Viola, Michael J. Jones, e Daniel Snow. Detecting pedestrians using patterns of motion and appearance. Em *Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2, ICCV '03*, páginas 734–, Washington, DC, USA, 2003. IEEE Computer Society.
- [5] Antoni Bert Chan. *Beyond dynamic textures: a family of stochastic dynamical models for video with applications to computer vision*. Tese de doutoramento, La Jolla, CA, USA, 2008. AAI3331461.
- [6] Bastian Leibe, Aleš Leonardis, e Bernt Schiele. Robust object detection with interleaved categorization and segmentation. *Int. J. Comput. Vision*, 77(1-3):259–289, Maio 2008.
- [7] Alper Yilmaz, Omar Javed, e Mubarak Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38(4):13, 2006.
- [8] Greg Welch e Gary Bishop. An introduction to the kalman filter, 1995.
- [9] John G Allen, Richard Y D Xu, e Jesse S Jin. Object tracking using camshift algorithm and multiple quantized feature spaces. *Reproduction*, 36, 2006.
- [10] Pedro Carvalho, Jaime S. Cardoso, e Luis Corte-Real. Hybrid framework for evaluating video object tracking algorithms. *Electronics Letters*, 46(6):411–412, 2010.
- [11] Nils Papenberg, Andrés Bruhn, Thomas Brox, Stephan Didas, e Joachim Weickert. Highly accurate optic flow computation with theoretically justified warping. *International Journal of Computer Vision*, 67(2):141–158, 2006.
- [12] J.L. Barron, D.J. Fleet, S.S. Beauchemin, e T.A. Burkitt. Performance of optical flow techniques. Em *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR '92., 1992 IEEE Computer Society Conference on*, páginas 236 –242, jun 1992.
- [13] P L Correia e F Pereira. Objective evaluation of video segmentation quality. *Image Processing, IEEE Transactions on*, 12(2):186–200, 2003.

- [14] C.E. Erdem, B. Sankur, e A.M. Tekalp. Performance measures for video object segmentation and tracking. volume 13, páginas 937–951, july 2004.
- [15] Liyuan Li, Weimin Huang, Irene Y. H. Gu, e Qi Tian. Foreground object detection from videos containing complex background. Em *In MULTIMEDIA '03: Proceedings of the eleventh ACM international conference on Multimedia*, páginas 2–10. ACM Press, 2003.
- [16] Gary Bradski e Adrian Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly, Cambridge, MA, 2008.
- [17] N. Dalal e B. Triggs. Histograms of oriented gradients for human detection. Em *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, páginas 886–893 vol. 1, june 2005.
- [18] P. Carvalho, J.S. Cardoso, L. Corte-Real, e L.F. Teixeira. Partition-distance methods for assessing spatial segmentations of images and videos. *Computer Vision and Image Understanding*, 1(103), 2 2009.
- [19] Faisal Bashir e Fatih Porikli. Performance evaluation of object detection and tracking systems. *Performance Evaluation*, 27(1):18–24, 2006.
- [20] A. Utsumi, H. Mori, J. Ohya, e M. Yachida. Multiple-view-based tracking of multiple humans. Em *Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on*, volume 1, páginas 597–601 vol.1, aug 1998.
- [21] R. Cipolla, T. Drummond, e D. Robertson. Camera calibration from vanishing points in images of architectural scenes, 1999.
- [22] Fengjun Lv, Tao Zhao, e Ram Nevatia. Self-calibration of a camera from video of a walking human. Em *In ICPR*, páginas 639–644, 2002.
- [23] Ismail Haritaoglu, David Harwood, e Larry S. Davis. W4: Real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:809–830, 2000.
- [24] Nils T. Siebel e Stephen J. Maybank. Fusion of multiple tracking algorithms for robust people tracking. Em *Proceedings of the 7th European Conference on Computer Vision-Part IV, ECCV '02*, páginas 373–387, London, UK, UK, 2002. Springer-Verlag.
- [25] W.N. Goncalves, J.B.O. Monteiro, J. de Andrade Silva, B.B. Machado, H. Pistori, e V. Oda-kura. Multiple mice tracking using a combination of particle filter and k-means. Em *Computer Graphics and Image Processing, 2007. SIBGRAPI 2007. XX Brazilian Symposium on*, páginas 173–178, oct. 2007.
- [26] B.P.L. Lo e S.A. Velastin. Automatic congestion detection system for underground platforms. Em *Intelligent Multimedia, Video and Speech Processing, 2001. Proceedings of 2001 International Symposium on*, páginas 158–161, 2001.
- [27] R. Cucchiara, C. Grana, M. Piccardi, e A. Prati. Detecting moving objects, ghosts, and shadows in video streams. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(10):1337–1342, oct. 2003.

- [28] C.R. Wren, A. Azarbayejani, T. Darrell, e A.P. Pentland. Pfinder: real-time tracking of the human body. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):780–785, jul 1997.
- [29] Xiang Gao, T.E. Boult, F. Coetzee, e V. Ramesh. Error analysis of background adaption. Em *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 1, páginas 503–510 vol.1, 2000.
- [30] C. Stauffer e W.E.L. Grimson. Adaptive background mixture models for real-time tracking. Em *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2, páginas 2 vol. (xxiii+637+663), 1999.
- [31] A. Elgammal, R. Duraiswami, D. Harwood, e L.S. Davis. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. *Proceedings of the IEEE*, 90(7):1151–1163, jul 2002.
- [32] M. Piccardi e T. Jan. Mean-shift background image modelling. Em *Image Processing, 2004. ICIP '04. 2004 International Conference on*, volume 5, páginas 3399–3402 Vol. 5, oct. 2004.
- [33] Bohyung Han e et al. Sequential kernel density approximation through mode propagation: Applications to background modeling. Em *IN PROC. ACCV 2004*, 2004.
- [34] N.M. Oliver, B. Rosario, e A.P. Pentland. A bayesian computer vision system for modeling human interactions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):831–843, aug 2000.
- [35] M. A. Ali, S. Indupalli, e B. Boufama. Tracking multiple people for video surveillance.
- [36] Jun Tang. A color image segmentation algorithm based on region growing. Em *Computer Engineering and Technology (ICCET), 2010 2nd International Conference on*, volume 6, páginas V6–634–V6–637, april 2010.
- [37] Dong-Xian Lai, Yuan-Hsiang Chang, e Zhi-He Zhong. Active contour tracking of moving objects using edge flows and ant colony optimization in video sequences. Em *Proceedings of the 3rd Pacific Rim Symposium on Advances in Image and Video Technology, PSIVT '09*, páginas 1104–1116, Berlin, Heidelberg, 2008. Springer-Verlag.
- [38] Cheng-Hung Chuang e Wen-Nung Lie. Fast and accurate active contours for object boundary segmentation. Em *Circuits and Systems, 2000. IEEE APCCAS 2000. The 2000 IEEE Asia-Pacific Conference on*, páginas 473–476, 2000.
- [39] S. Indupalli, M.A. Ali, e B. Boufama. A novel clustering-based method for adaptive background segmentation. Em *Computer and Robot Vision, 2006. The 3rd Canadian Conference on*, página 37, june 2006.
- [40] A.-I. Sarpe. Image segmentation with clustering k-means and watershed transform. Em *Advances in Multimedia (MMEDIA), 2010 Second International Conferences on*, páginas 13–17, june 2010.
- [41] Constantine Papageorgiou e Tomaso Poggio. A trainable system for object detection. *Int. J. Comput. Vision*, 38(1):15–33, Junho 2000.

- [42] Che-Hung Lin, Sheng-Luen Chung, e Jing-Ming Guo. People tracking in a building using color histogram classifiers and gaussian weighted individual separation approaches. Em *Proceedings of the 17th international conference on Advances in multimedia modeling - Volume Part II*, MMM'11, páginas 177–186, Berlin, Heidelberg, 2011. Springer-Verlag.
- [43] A. Müfit Ferman, A. Murat Tekalp, e Rajiv Mehrotra. Robust color histogram descriptors for video segment retrieval and identification, 2002.
- [44] Michael J. Swain e Dana H. Ballard. Color indexing. *International Journal of Computer Vision*, 7:11–32, 1991.
- [45] D G Lowe. Object recognition from local scale-invariant features. Em *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, páginas 1150–1157 vol.2, 1999.
- [46] Herbert Bay, Tinne Tuytelaars, e Luc Van Gool. Surf: Speeded up robust features. Em *In ECCV*, páginas 404–417, 2006.
- [47] Marek Kraft e Adam Schmidt. Simplifying surf feature descriptor to achieve real-time performance. Em Robert Burduk, Marek Kurzyński, Michał Woźniak, e Andrzej Żołnierek, editores, *Computer Recognition Systems 4*, volume 95 de *Advances in Intelligent and Soft Computing*, páginas 431–440. Springer Berlin Heidelberg, 2011.
- [48] A. Yavlinsky, M.J. Pickering, D. Heesch, e S. Ruger. A comparative study of evidence combination strategies. Em *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, volume 3, páginas iii – 1040–3 vol.3, may 2004.
- [49] Cheng Chang, R. Ansari, e A. Khokhar. Multiple object tracking with kernel particle filter. Em *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, páginas 566 – 573 vol. 1, june 2005.
- [50] Ishwar K. Sethi e Ramesh Jain. Finding trajectories of feature points in a monocular image sequence. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-9(1):56–73, jan. 1987.
- [51] V. Salari e I.K. Sethi. Feature point correspondence in the presence of occlusion. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(1):87–91, jan 1990.
- [52] C. J. Veenman, M. J. T. Reinders, e E. Backer. Resolving motion correspondence for densely moving points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:54–72, 2001.
- [53] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistic Quarterly*, 2:83–97, 1955.
- [54] S. Yamamoto, Y. Mae, Y. Shirai, e J. Miura. Realtime multiple object tracking based on optical flows. Em *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on*, volume 3, páginas 2328 –2333 vol.3, may 1995.
- [55] T. Kodama, T. Yamaguchi, e H. Harada. A method of object tracking based on particle filter and optical flow. Em *ICCAS-SICE, 2009*, páginas 2685 –2690, aug. 2009.
- [56] Concept Derivation, , e Maria Isabel Ribeiro. Kalman and extended kalman filters:.

- [57] J. Gonzalez. Particle filters.
- [58] Bo Yang, Xinting Pan, Aidong Men, e Xiaobo Chen. A robust particle filter for people tracking. Em *Future Networks, 2010. ICFN '10. Second International Conference on*, páginas 20–23, jan. 2010.
- [59] Paul Brasnett, Lyudmila Mihaylova, David Bull, e Nishan Canagarajah. Sequential monte carlo tracking by fusing multiple cues in video sequences. *Image Vision Comput.*, 25(8):1217–1227, Agosto 2007.
- [60] M. Marron, J.C. Garcia, M.A. Sotelo, M. Cabello, D. Pizarro, F. Huerta, e J. Cerro. Comparing a kalman filter and a particle filter in a multiple objects tracking application. Em *Intelligent Signal Processing, 2007. WISP 2007. IEEE International Symposium on*, páginas 1–6, oct. 2007.
- [61] D. Comaniciu e P. Meer. Mean shift: a robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):603–619, may 2002.
- [62] Nicole M. Artner. A comparison of mean shift tracking methods. Em *12th Central European Seminar on Computer Graphics*, páginas 197–204, April 2008.
- [63] Gary R. Bradski. Computer vision face tracking for use in a perceptual user interface, 1998.
- [64] S.S. Blackman. Multiple hypothesis tracking for multiple target tracking. *Aerospace and Electronic Systems Magazine, IEEE*, 19(1):5–18, jan. 2004.
- [65] I.J. Cox e S.L. Hingorani. An efficient implementation of reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(2):138–150, feb 1996.
- [66] Mark Polak, Hong Zhang, e Minghong Pi. An evaluation metric for image segmentation of multiple objects. *Image and Vision Computing*, 27(8):1223–1227, Julho 2009.
- [67] P. Villegas e X. Marichal. Perceptually-weighted evaluation criteria for segmentation masks in video sequences. *Image Processing, IEEE Transactions on*, 13(8):1092–1103, aug. 2004.
- [68] Andrea Cavallaro, Elisa Drelie Gelasca, e Touradj Ebrahimi. Objective evaluation of segmentation quality using spatio-temporal context. Em *In IEEE International Conference on Image Processing*, páginas 301–304, 2002.
- [69] J. Popoola e A. Amer. Performance evaluation for tracking algorithms using object labels. Em *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, páginas 733–736, 31 2008-april 4 2008.
- [70] Chris J. Needham e Roger D. Boyle. Performance evaluation metrics and statistics for positional tracker evaluation. Em *Proceedings of the 3rd international conference on Computer vision systems, ICVS'03*, páginas 278–289, Berlin, Heidelberg, 2003. Springer-Verlag.
- [71] E. Maggio e A. Cavallaro. Learning scene context for multiple object tracking. *Image Processing, IEEE Transactions on*, 18(8):1873–1884, aug 2009.
- [72] A. Ellis e J. Ferryman. Pets2010 and pets2009 evaluation of results using individual ground truthed single views. Em *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*, páginas 135–142, 29 2010-sept. 1 2010.

- [73] Keni Bernardin, Er Elbs, e Rainer Stiefelhagen. Multiple object tracking performance metrics and evaluation in a smart room environment.
- [74] Hui Zhang, Jason E. Fritts, e Sally A. Goldman. Image segmentation evaluation: A survey of unsupervised methods. *Computer Vision and Image Understanding*, 110(2):260 – 280, 2008.
- [75] Berthold K. P. Horn e Brian G. Schunck. Determining optical flow. *ARTIFICIAL INTELLIGENCE*, 17:185–203, 1981.
- [76] Bruce D. Lucas e Takeo Kanade. An iterative image registration technique with an application to stereo vision (ijcai). Em *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81)*, páginas 674–679, April 1981.
- [77] Hans-Hellmut Nagel. On the estimation of optical flow: Relations between different approaches and some new results. *Artificial Intelligence*, 33(3):299 – 324, 1987.
- [78] Minu Ayromlou, Michael Zillich, Wolfgang Ponweiser, e Markus Vincze. Measuring scene complexity to adapt feature selection of model-based object tracking. Em James Crowley, Justus Piater, Markus Vincze, e Lucas Paletta, editores, *Computer Vision Systems*, volume 2626 de *Lecture Notes in Computer Science*, páginas 448–459. Springer Berlin / Heidelberg, 2003.
- [79] Jigang Wang, Predrag Neskovic, e Leon N Cooper. Context-based tracking of object features.
- [80] H.T. Nguyen, Qiang Ji, e A.W.M. Smeulders. Spatio-temporal context for robust multitarget tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(1):52–64, jan 2007.
- [81] J.C. SanMiguel e J.M. Marti andnez. On the evaluation of background subtraction algorithms without ground-truth. Em *Advanced Video and Signal Based Surveillance (AVSS), 2010 Seventh IEEE International Conference on*, páginas 180 –187, 29 2010-sept. 1 2010.
- [82] P. Kaewtrakulpong e R. Bowden. An improved adaptive background mixture model for real-time tracking with shadow detection. Em *Proceedings of 2nd European Workshop on Advanced Video Based Surveillance Systems*, volume 5308, 2001.
- [83] A.J. Lipton, H. Fujiyoshi, e R.S. Patil. Moving target classification and tracking from real-time video. Em *Applications of Computer Vision, 1998. WACV '98. Proceedings., Fourth IEEE Workshop on*, páginas 8 –14, oct 1998.
- [84] A. Ilyas, M. Scuturici, e S. Miguet. Real time foreground-background segmentation using a modified codebook model. Em *Advanced Video and Signal Based Surveillance, 2009. AVSS '09. Sixth IEEE International Conference on*, páginas 454 –459, sept. 2009.
- [85] P L Correia e F Pereira. Classification of video segmentation application scenarios. *Circuits and Systems for Video Technology, IEEE Transactions on*, 14(5):735–741, 2004.
- [86] Caviar: Context aware vision using image-based active recognition. <http://homepages.inf.ed.ac.uk/rbf/CAVIAR/>, 2001.
- [87] Pets2006: Ninth iee international workshop on performance evaluation of tracking and surveillance. <http://www.cvg.rdg.ac.uk/PETS2006/>, 2006.