Faculdade de Engenharia da Universidade do Porto

Seguimento de grupos de pessoas utilizando vídeo

Alexandra Maria Pereira Familiar

Mestrado Integrado em Engenharia Eletrotécnica e Computadores

Orientador: Luís Corte Real

Co-Orientador: Pedro Carvalho

27 de Julho de 2015

U.PORTO
FEUP FACULDADE DE ENGENHARIA UNIVERSIDADE DO PORTO

MIEEC - MESTRADO INTEGRADO EM ENGENHARIA ELETROTÉCNICA E DE COMPUTADORES | **2014/2015**

A Dissertação intitulada

"Seguimento de Grupos de Pessoas Utilizando Vídeo"

foi aprovada em provas realizadas em 20-07-2015

o júri

Presidente Professor Doutor Aníbal João de Sousa Ferreira
Professor Associado do Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto

Professor Doutor José Manuel de Castro Torres
Professor Associado da Faculdade de Ciências e Tecnologia da Universidade Fernando Pessoa

Professor Doutor Luís António Pereira de Meneses Corte-Real
Professor Associado do Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto

O autor declara que a presente dissertação (ou relatório de projeto) é da sua exclusiva autoria e foi escrita sem qualquer apoio externo não explicitamente autorizado. Os resultados, ideias, parágrafos, ou outros extratos tomados de ou inspirados em trabalhos de outros autores, e demais referências bibliográficas usadas, são corretamente citados.

Autor - Alexandra Maria Pereira Familiar

Faculdade de Engenharia da Universidade do Porto

Resumo

Nos últimos anos tem-se verificado um aumento da monitorização de espaços. A sua dimensão e o grande número de pessoas que aí se podem reunir compromete a segurança do espaço e das pessoas. Assim, criou-se a necessidade de dar capacidade aos sistemas de segurança de perceberem o que captam para assim reconhecer comportamentos anómalos e antever situações de perigo.

Quando um conjunto de pessoas se junta, é comum a formação de grupo obrigando a que, para além de seguimento de pessoas, se invista na análise e gestão de grupos. Embora a análise de grupo traga vantagens, como por exemplo a previsão das posições das pessoas do grupo quando estas estão ocultas, também acarreta desafios, que incluem a própria definição de grupo. Definir o grupo corretamente é um passo importante e difícil para a sua gestão: os grupos poderão ter diferentes dimensões de pessoas dispostas nas mais variadas posições. O grande número de ocultações das pessoas é outro desafio, pois quando há junções de pessoas são comuns as ocultações e a imprevisibilidade das trajetórias o que torna o processo de seguimento muito mais exigente. A estes desafios acrescentam-se ainda as alterações nas dinâmicas dentro dos grupos (por exemplo, quando alguém se junta ou separa de um grupo) e a interação entre grupos (junção e desintegração desses) juntar-se ou separar-se de grupos, e esses podem desintegrar-se ou juntar-se a outros. Estes são os desafios principais que o algoritmo de gestão de grupos têm de lidar. Depois de um estudo do estado da arte, a presente dissertação apresenta uma proposta de um algoritmo para lidar com grupos.

O algoritmo de gestão de grupos, recebendo como entrada a informação das pessoas em cena, deteta os grupos e armazena quais as pessoas a que eles pertencem ao longo do tempo. A informação de entrada, resultado de um algoritmo de seguimento de pessoas, pode estar sujeita a erros e assim o algoritmo de gestão tem de ter a capacidade de os detetar e saber lidar com eles por forma a conseguir um bom desempenho.

Abstract

With time, the need to monitor spaces where people gather has increased. The space dimension and the number of persons that can meet in those locations could potentially compromise the security of the space and people. Therefore, there was the need to give security systems the ability to understand what is being monitored to recognize strange behaviours and predict potentially dangerous situations.

When a group of people gets together, it's common that a group is formed, which must be handled differently when compared to people as group analysis and management become additional tasks of the system. Even though the group analysis brings advantages, such as predicting the position of the persons in the group even under heavy occlusion, it also adds several challenges, which include the group definition itself. Correctly defining a group is an important and challenging step for group analysis as groups may have a varying number of people positioned in a multitude of ways. The number of occlusions also adds another challenge as when people join a group, they may be positioned behind other people and may take very different trajectories even within a group. An additional challenge comes from the dynamics within groups and between groups: a person may join or leave a group, groups may disband or join another group. These are some of the main challenges that a group management algorithm will have to handle. After a State-of-the-Art analysis, this dissertation presents an algorithm proposal that can handle groups.

The group management algorithm receives the information of the persons in the scene, detects the existing groups and stores the persons that belong to it through time. The input information comes from a people tracking algorithm, which means that errors may be part of this input. Therefore, the group management should be prepared to detect and handle them without a significant loss in performance.

Agradecimentos

O meu primeiro e mais sentido agradecimento é para os meus pais, cujo esforço e força tornou possível a minha formação superior. Agradeço também à minha irmã e à minha sobrinha por mostrarem interesse e orgulho em mim e no meu trabalho e por me darem força quando foi preciso.

Ao Bruno Moreira pelo incansável interesse em ajudar, pelas criticas construtivas que ajudaram a elevar a tese a um nível superior, pela paciência e pela força e confiança que mostrou ter no trabalho produzido.

Quero também agradecer ao meu avô, que sempre me mostrou que a sabedoria e a força de vontade são as melhores armas que uma pessoa pode ter.

Ao Gil Coelho e ao Américo Pereira a minha gratidão pelos conselhos dados ao longo do execução da tese que muito vieram acrescentar.

Quero também prestar agradecimento ao meu Orientador Professor Doutor Luís Corte-Real e ao meu Supervisor Externo Doutor Pedro Carvalho pela orientação, discussão e pelo interesse que mostraram e pela vontade de tornar o trabalho melhor.

De uma forma geral, agradeço a todos os que de alguma forma ajudaram nesta fase académica.

Alexandra Maria Pereira Familiar

A dissertação foi desenvolvida também no contexto e em colaboração com os projetos: Media Arts and Technologies (MAT), NORTE-07-0124-FEDER-000061, financiado pelo Programa Operacional Regional do Norte (ON.2 Ű O Novo Norte), sob o Quadro de Referência Estratégica Nacional (QREN), através do Fundo Europeu de Desenvolvimento Regional (FEDER), e por fundos nacionais através da agência de financiamento Portuguesa, Fundação para a Ciência e a Tecnologia (FCT); Project QREN 23277 RETAIL PRO, um projeto de I&D em co-promoção financiado pelo Fundo Europeu de Desenvolvi-? mento Regional (FEDER) através do ON2 como parte do Quadro de Referência Estratégica Nacional (QREN), e gerido pela Agência de Inovação (ADI); QREN 33910 ARENA, um projeto de I&D financiado pelo Fundo Europeu de Desenvolvimento Regional (FEDER) através do Programa Operacional Regional do Norte (ON.2 Ű O Novo Norte) como parte do Quadro de Referência Estratégica Nacional (QREN), e gerido pelo IAPMEI - Agência para a Competitividade e Inovação, I.P.

"Look, a distraction!"

Desconhecido

Conteúdo

Lista de Figuras

Abreviaturas e Símbolos

CAVIAR	*Context Aware Vision using Image-based Active Recognition*
DPF	*Decentralized Particle Filter*
EOH	*Edge Orientation Histogram*
FN	Falsos Negativos
FP	Falsos Positivos
GAB	*Gentle Adaboost*
GI	*Group Incoherence*
GT	*Ground Truth*
HOG	*Histogram of Oriented Gradients*
HSV	*Hue Saturation Value*
KAT	*Kalman Appearance Tracker*
KLT	*Kanade Lucas Tomasi*
LAC	*Linear Asymmetric Classifier*
LBP	*Local Binary Patterns*
LOT	*Locally Orderless Tracker*
MHT	*Multiple Hypothesis Tracking*
MO	*Multiple Object*
MT	*Multiple Track*
MTT	*Multiple Traget TRacking*
MOG	*Mixture Of Gaussians*
NCC	*Normalized Cross-Correlation*
PDF	*Probability Density Function*
RGB	*Red, Green, Blue*
SVM	*Support Vector Machine*
TCM	*Texton Cooccurrence Matrix*
TF	*Track Fragmentation*
TP	*True Positives*
ScRek	*Scenario recognition based on knowledge*

Capítulo 1

Introdução

1.1 Contextualização

Nos últimos anos tem-se verificado um aumento de monitorização dos espaços. A frequência destes encontros, nem sempre pacíficos, veio criar a necessidade de garantir a segurança das pessoas. Para isso, é necessário monitorizar e vigiar espaços tais como aeroportos, bancos, centros comerciais e até estádios de futebol. A sua dimensão, bem como o grande número de pessoas que aí se reúne, inviabiliza a vigilância desses locais sem tecnologia adequada. Assim, tornou-se importante a instalação de câmaras de segurança em locais chave. Com o aumento da dimensão destes espaços e do número de pessoas nele reunidas, a instalação de câmaras tornou-se por si só insuficiente: num só edifício, poderá haver a necessidade de vigiar mais do que uma área ao mesmo tempo, e na mesma área pode haver a necessidade de monitorizar mais do que uma pessoa. Isto acarreta uma maior dificuldade de monitorização, podendo tornar impossível a garantia de segurança no espaço. De forma a tornar os sistemas de vigilância mais eficientes, tem-se investido em torná-los mais autónomos, de forma a que possam lidar com mais do que uma área e com um número crescente de pessoas. Assim, a necessidade de investir na captação de imagem pelas câmaras evoluiu para a necessidade de dar capacidade aos sistemas de vigilância de perceber o que captam. Fazer um investimento neste campo possibilita reconhecer comportamentos anómalos e antever situações de perigo, dando tempo às entidades competentes de intervirem.

Quando várias pessoas se juntam no mesmo sitio, é comum formarem-se grupos. Com o aumento da densidade, há perda de eficiência dos algoritmos de deteção e de seguimento, não sendo adequado o uso de algoritmos típicos de deteção e de seguimento de pessoas. Assim, cria-se a necessidade de desenvolver algoritmos de grupos que possam fazer parte de um algoritmo maior que seja também capaz de seguir indivíduos. Embora seja vantajoso seguir um grupo em vez de seguir todas as pessoas que o compõem, surgem novos desafios, muitos ainda sem solução. Conseguir identificar e definir um grupo, segui-lo, saber lidar com as interações entre pessoas e grupos de pessoas são questões que é necessário enfrentar

para se conseguir produzir um bom sistema de seguimento. Quando se lidam com cenas onde há formação de grupos de pessoas, o aumento do número de ocultações e a imprevisibilidade das trajetórias são problemas a ter em conta ao projetar o sistema. Tendo em conta a necessidade de desenvolver algoritmos de seguimento que saibam lidar com qualquer ambiente, com a presente dissertação pretende-se estudar e desenvolver métodos para lidar com o seguimento pessoas em grupos.

1.2 Desafios

Um algoritmo de gestão de grupos, para se revelar eficiente e robusto, terá de superar diversos obstáculos. Um dos maiores desafios é a própria definição de grupo, que é fulcral para se conseguir detetar e seguir um conjunto de indivíduos como sendo um grupo, pois este não é apenas um conjunto de pessoas próximas umas das outras, mas um conjunto de pessoas unidas por objetivos comuns. Um grupo poderá ser um conjunto de pessoas que se junta para falar ou até uma fila de pessoas. Sendo composto por pessoas, a sua dinâmica interna pode facilmente modificar-se: pode entrar ou sair uma pessoa do grupo, haver alterações no comportamento de um dos seus elementos (uma pessoa afasta-se ligeiramente do grupo e depois aproxima-se), por exemplo. Os grupos também poderão interagir uns com os outros: dois grupos juntarem-se, um grupo passar pelo meio do outro ou até trocarem elementos. O algoritmo terá de avaliar cada situação, processá-la e agir em conformidade. O resultado poderá ser juntar ou retirar alguém ao grupo, formar/eliminar um grupo ou decidir manter a sua estrutura. Estas situações, sendo comuns no dia a dia, terão de ser corretamente identificadas e tratadas pelo algoritmo.

Um algoritmo de gestão de grupos recebe o resultado de um algoritmo de seguimento de pessoas individuais e, como tal terá de lidar com informação com erros. Esta poderá traduzir-se em falsas deteções (o algoritmo deteta um alvo que não existe), ausência de deteções de pessoas na cena (o algoritmo não consegue detetar todos os alvos em cena) ou erros na localização e dimensão das pessoas. Para conseguir um algoritmo de gestão de grupos robusto, este terá de ser capaz de detetar e adaptar-se a estas situações de erro. Um tratamento deficiente de erros provoca perda de eficiência visto que ocorrerá validação de informação incorreta (inclusão de uma falsa pessoa num grupo, por exemplo) ou rejeição de informação válida (considerar que a informação de uma pessoa válida é incorreta e excluí-la da cena).

Por último, é necessário avaliar os resultados obtidos. A Literatura carece de métricas uniformizadas para avaliação de algoritmos que lidem com grupos obrigando à criação ou adaptação de métricas já existentes.

1.3 Objetivos

O primeiro objetivo da presente tese será compreender os conceitos subjacentes à identificação, seguimento e gestão de grupos. Um segundo objetivo será estudar e analisar diferentes técnicas que contribuam para uma melhor gestão de grupos.

A informação anterior será depois usada de forma a desenvolver um algoritmo de gestão de grupos em tempo real. Tendo a informação das pessoas em cena, o algoritmo terá de ser capaz de detetar os diferentes grupos bem como os seus elementos. Ao longo do tempo, terá de conseguir seguir os grupos, gerir as suas identidades, detetar junções e separações de pessoas-grupos, bem como detetar mudanças nas suas dinâmicas internas. Tudo isto terá de ser feito tendo em conta as interferências (erros) devendo lidar com este de forma a conseguir detetar e seguir os grupos sem decréscimos significativos de eficiência.

Um outro objetivo é a reconstrução das trajetórias das pessoas em cena mesmo quando a informação destas se vê corrompida por erros. O algoritmo terá de conseguir estimar a informação em falta, por forma a conseguir reconstruir uma possível trajetória da pessoa.

1.4 Estrutura do documento

O presente documento está dividido em 5 capítulos. O segundo capítulo descreve alguns dos algoritmos mais relevantes no contexto da criação de um algoritmo de gestão de grupos. Serão depois detalhados no terceiro capítulo algumas das sequências de dados e métricas disponíveis, sendo feita uma discussão daquelas que serão utilizadas. O quarto capítulo apresenta os detalhes do algoritmo escolhido como base, assim como as melhorias propostas e respetivos resultados avaliados em sequências de dados relevantes. O último capítulo apresenta uma discussão do trabalho desenvolvido e algumas sugestões de trabalho futuro.

Capítulo 2

Revisão da literatura

2.1 Conceitos Gerais

Quando se pretende fazer seguir uma entidade, é necessário saber quais os objetos de interesse a seguir e as características que os distinguem dos restantes. Isto é fulcral, pois o mesmo método pode não ser eficaz para seguir todos os objetos na cena. Assim importa destacar três entidades: pessoa, grupo e multidão. Um grupo é definido como um conjunto estruturado de pessoas que interagem umas com as outras e cujo comportamento é influenciado pelos restantes [1]: quando uma pessoa se encontra num grupo, ela adotará o comportamento do grupo a que pertence, comportamento esse que seria diferente se a pessoa se encontrasse sozinha. Na Figura 2.1 está representado um exemplo de um grupo com a sua respetiva caixa delimitadora (*bounding box*).

Figura 2.1: Exemplo de um grupo de pessoas (extraído de [2])

A estrutura do grupo é definida como um conjunto de regiões em movimento com as seguintes particularidades [3] :

- Coerência de tamanho: Cada região terá de ter o tamanho de uma pessoa ou maior se se tratar de várias pessoas juntas;

- Coerência espacial: Todas as regiões em movimento dentro de um grupo têm de estar próximas umas das outras
- Coerência temporal: A velocidade das regiões em movimento dentro de um grupo não poderá ser superior à velocidade de uma só pessoa.
- Coerência estrutural: O número e o tamanho das regiões em movimento dentro de um grupo terá de ser constante.

Esta forma de classificar um grupo pressupõe, ao contrário do que é mencionado em [1], que a estrutura de um grupo pode ser constituído apenas por uma pessoa, isto é, é considerado grupo toda a entidade que se pretende seguir. Quando o conjunto de pessoas se torna suficientemente denso de forma a não se conseguir distinguir indivíduos ou grupos no cenário, estaremos então na presença de uma multidão [4]. Na figura 2.2 temos o exemplo de uma multidão.

Figura 2.2: Exemplo de um grupo de pessoas (extraído de [5])

Uma multidão [1], ao contrário de um grupo, não tem uma estrutura estável e definida. Uma quantidade de pessoas que esteja próxima constituirá uma multidão apenas quando o seu interesse está polarizado, por exemplo, um conjunto de pessoas a tentar sair por uma porta de emergência. Os mesmo autores consideram que as multidões são pouco estruturadas e as interações entre pessoas que a ela pertencem são rápidas e difusas.

Na literatura são mencionadas duas formas principais de lidar com grupos e multidões [6]: através de uma abordagem holística ou através de uma abordagem baseada no objeto. A abordagem holística trata o grupo uma única entidade [7], não requerendo uma segmentação de cada pessoa. Por este motivo pode ser usada, por exemplo, para detetar quando uma ou mais pessoas se movem contra uma multidão. Na abordagem baseada no objeto [8], o grupo é visto como um conjunto de entidades e para proceder à sua analise é necessária a deteção dos objetos a que ele pertencem e por esse mesmo facto, usa-se em ambientes com pouca densidade.

2.2 Subtração de fundo

Uma das formas de detetar objetos de interesse a partir de uma sequência de vídeo, é através da subtração de fundo. De um modo geral, os métodos de subtração de fundo passam por quatro fases: pré-processamento, modelização de fundo, subtração de fundo (*background subtraction*) e validação. Como resultado, obtém-se uma máscara, tipicamente binária, onde se distinguem os pixeis que pertencem ao fundo (*background*) dos que pertencem ao primeiro plano (*foreground*), tal como se pode ver na Figura 2.3

Figura 2.3: Exemplo de subtração de fundo: à esquerda a imagem original e à direita a sua respetiva máscara (extraído de [2])

A fase mais importante será o *background modeling* e a eficiência do processo de subtração de fundo depende dele. As técnicas usadas para esta fase podem ser classificadas em dois tipos [9]: não recursivas e recursivas. As técnicas não recursivas caracterizam-se por usarem uma janela que percorre a imagem de forma a poder estimar o modelo de *background*. A previsão é feita através da informação das últimas imagens (*frames*) do vídeo. As técnicas recursivas não mantêm um vetor com as estimações do *background* mas atualizam recursivamente um modelo de *background* em cada *frame* recebida.

A forma mais simples de se proceder à subtração de fundo é através do método de *frame differencing* [10] onde o *background* é estimado através da *frame* anterior. É feita uma subtração dos píxeis da *frame* atual pelos da anterior. Se a diferença for superior a um dado limiar, então o píxel é considerado *foreground*. Este método, embora seja muito simples e acarrete menor custo computacional, tem uma eficácia muito dependente do ruído e da taxa de atualização da imagem e tem baixo desempenho quando a diferença entre *frames* não representa o *foreground*.

O Filtro da Mediana (*Median Filter*) [11], é uma das abordagens mais usadas para subtração de fundo. Assume-se que os píxeis permanecem no *background* em mais de metade das *frames* e a estimação do *background* é o valor da mediana para cada píxel nas n últimas *frames*.

O método *Running Average Gaussian* [12] tenta encontrar uma função de probabilidade gaussiana que se enquadre nos últimos n píxeis. Como apenas requer o armazenamento de dois valores (média e desvio de padrão) este método tem a vantagem de não ter grandes custos de memória. Uma outra possibilidade é o uso do filtro linear preditivo [13]. Este

filtro é aplicado tendo por base as informações dos píxeis de *frames* anteriores, estimando o valor para o modelo de *background* atual. Os coeficientes deste filtro são estimados para cada *frame* e estão relacionados com a o valor da co-variância. Este facto torna o método desadequado para aplicações em tempo real.

Um outro método não recursivo é o *optical flow* [14], que considera *foreground* todos os píxeis em que se deu uma alteração da luminosidade.

Ao contrário dos métodos não recursivos, os métodos recursivos atualizam o modelo de *background* em cada *frame* recebida. Assim, não são ignoradas as *frames* mais distantes aquando a estimação do *background*. Embora estas abordagens requeiram menor armazenamento, têm o inconveniente de poder haver propagação de erros, visto que um erro pode manter-se por um longo período de tempo.

Um método recursivo é o filtro de mediana aproximada [13], em que a mediana é estimada somando 1 quando o valor do píxel de entrada é maior que a mediana, e decrementando 1 se for inferior. Desta forma, o valor irá eventualmente convergir para um valor acima de metade dos valores de píxel e abaixo da outra metade dos valores, ou seja, a mediana.

Um método muito usado para modelizar *background* é o *Mixture of Gaussians* (MOG) [13] que assume que a intensidade de cada píxel pode ser modelada através de um conjunto de distribuições gaussianas. Assim, esta técnica determina quais os valores de intensidade que têm uma maior probabilidade de pertencerem ao *background*.

Para além destas abordagens, continua a investir-se no desenvolvimento de novas técnicas de subtração de fundo. Em [15] é proposto um método que recorre às propriedades da *texton co-occurrence matrix* (TCM) para detetar objetos em movimento. A TCM é uma matriz que calcula distribuição dos valores das co-ocorrências num dado desvio. Este método, embora proposto para recuperação de imagens, *image retrieval* [16], é agora aplicado em subtração de fundo, pois é capaz de integrar características de cor, textura e forma. O método é usado para descrever relações espaciais sendo aplicado na vizinhança dos píxeis. O uso do TCM permite resolver problemas que estejam associados a fundos não estacionários, variação de luminosidade e camuflagem.

Um técnica recente, intitulada ViBe [17], calcula o modelo de fundo tendo por base amostras extraídas de *background* . Este modelo é feito através de vinte amostras de *background* para cada píxel, escolhidas de forma aleatória. Ao ser escolhido um píxel para ser inserido no modelo, é usado um mecanismo de propagação espacial, pelo que também os píxeis da vizinhança são inseridos. Uma particularidade desta abordagem é o facto de se ignorar a informação temporal quando se procede à substituição dos píxeis, isto é, píxeis de *frames* mais recentes ou de $frames$ mais antigas têm igual probabilidade de serem substituídos. Entretanto foram propostas modificações [18] ao método original. Uma destas alterações foi a distinção de máscara de segmentação e da máscara de atualização, isto é, não se permite que píxeis pertencentes ao *foreground* sirvam para atualizar o modelo. Também foram aplicados métodos de abertura de áreas, *area openings* [19], às

máscaras e foram adicionados mecanismos para evitar a propagação espacial. Por fim, foram adicionadas medidas adaptadas de distância [20] e *thresholding*.

2.3 Deteção de pessoas

A deteção de pessoas consiste em, a partir de uma imagem com múltiplas regiões, conseguir distinguir e extrair aquelas que contêm uma pessoa. Este processo é importante por ser frequentemente a base do seguimento de pessoas. Na figura 2.4 podemos ver a deteção de cinco pessoas.

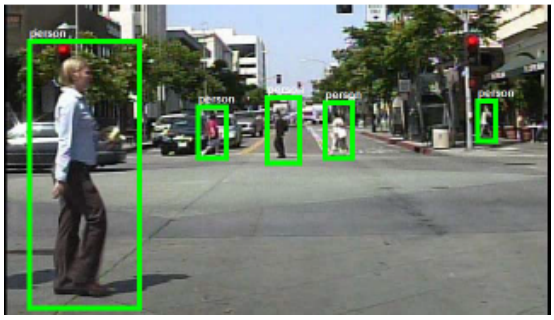


Figura 2.4: Exemplo de deteção de pessoas com as suas respetivas *bounding boxes* (extraído de [21])

O Histograma de Orientação de Bordas (EOH) [22], embora usado inicialemente para deteção facial, tem vindo a ser usado para deteção de pessoas. A extração da informação de silhuetas através deste método tem a vantagem de ser invariante às mudanças de iluminação e a facilidade de extração de propriedades geométricas tem vindo a tornar comum o uso deste método. Em [23], é usado este método em combinação com o características de Haar-Like [24] e com o classificador Real Adaboost [25] para deteção de pessoas. Esta combinação consegue resultados que se aproximam do método de Histogramas de Orientação de Gradientes (HOG) [26]. O HOG é um dos métodos mais usados para deteção de pessoas e consiste na extração de informação usando gradientes segundo diversas orientações, que são posteriormente armazenados em histogramas. Este método mostra-se eficaz a lidar com mudanças de luminosidade e com pequenas variações de pose. Estas características são avaliadas segundo uma *support vector machine* (SVM) linear [27]. Uma forma proposta para aumentar a eficácia do HOG é usá-lo juntamente com outros métodos.

Em [28] são usados Histogramas de Orientação de Gradientes juntamente com descritores espaço-temporais ou de fluxo ótico para analisar uma pessoa. Uma janela de [64x128] píxeis percorre a imagem em diferentes escalas, sendo usado um classificador SVM [27] para detetar a presença de uma pessoa em cada janela resultante, através de um conjunto de dados pré-definidos para treino. Este classificador é usado por mostrar melhores performances quando comparado com outros classificadores lineares. O método foca-se em

detetar pessoas em pé, ou numa posição quase vertical e que estejam sempre ou quase sempre visíveis. A maior desvantagem destas técnicas é o elevado custo computacional.

De forma idêntica, em [29] associam-se as características do HOG com as características Haar-Like [30]. As características de Haar-Like são usadas para detetar bordas ou mudanças de textura através do calculo da intensidade média de regiões vizinhas ao longo de diferentes orientações. Verifica-se que, para um mesmo classificador, a junção das características apresenta melhores resultados que a utilização isolada do HOG.

Outras características normalmente usadas para deteção de pessoas são as *local binary patterns* (LBP) [31]. Este método extrai características relacionadas com a informação de textura, sendo adaptado de [32]. Nesta adaptação, o classificador usado é o *Linear Asymmetric Classifier* (LAC) [33], que apresenta vantagens pelo uso de uso de classificação por cascata.

Em [34], o HOG e o LBP [35] são usados paralelamente e o resultado combinado é usado posteriormente num classificador SVM para assim detetar pessoas [27]. Houve a necessidade de associar os dois porque o HOG, em condições em que o o *background* tem bordas com ruído, mostra-se pouco eficiente, podendo ser complementado com o LBP visto que este tem a capacidade de filtrar o ruído por usar o conceito de padrão uniformizado. Esta abordagem mostra-se ineficaz a lidar com deformações articuladas de pessoas.

De forma a aumentar a velocidade, há métodos que recorrem ao classificador de cascata, como em [36]. Este classificador usa uma estrutura em árvore para minimizar o tempo e custo de classificação. Nesta abordagem, inicialmente são identificados os píxeis de *foreground* através da média da diferença do fundo, eliminando-se possíveis interferências no *background*. Seguidamente, é feito um treino de um classificador fraco usando *Weighted Linear Regression Model* embebido num Classificador *Cascade Gentle Adaboost* (GAB) [37]. Seguidamente, é feita uma fusão de classificadores *Cascade* em que o primeiro usa as características *Haar-Like* da zona ombros-cabeça e o segundo usa características *Shapelet* [38]. O primeiro classificador é usado para eliminar as regiões que não foram classificadas como pessoas e o segundo para reforçar a decisão do primeiro e excluir potenciais zonas que poderiam ter sido classificadas como pessoas, mesmo não o sendo.

Shape Context é um método inicialmente proposto por [39] usado para reconhecimento e correspondência de objetos. Baseia-se na extração da informação espacial de pontos extraídos da forma dos objetos, obtendo assim uma descrição sumária da forma existente na imagem. Posteriormente, em [40], foi usado na deteção de pessoas. De uma forma geral, este método escolhe n pontos dos contornos do objeto. Depois, para cada um desses pontos terá n-1 vetores obtidos através da ligação desse ponto com os restantes. Esses vetores servirão então como descritores do contorno do objeto e essa informação é posteriormente organizada num histograma espacial. Esta abordagem tem a vantagem de não ser dispendiosa computacionalmente. O classificador segue um sistema de votação para determinar se a forma corresponde a uma pessoa.

Uma abordagem alternativa será usar Descritores da Co-Variância da Região [41] em

que após extração de informação de cada píxel, tal como coordenadas, valores de cor e segundas derivadas de acordo com o eixos horizontal e vertical e do preenchimento de vetores com estes valores, o método calcula a co-variância para cada tipo de informação e reúne esses cálculos numa matriz que servirá como descritor da região.

Em [42], a segmentação e o seguimento de pessoas é feito através de uma busca prévia, procurando o melhor caminho no domínio espaço-temporal. Este algoritmo torna-se pouco prático devido ao seu elevado custo computacional na procura de correspondência das formas de silhueta, embora apresente bons resultados.

Quando se lida com cenários onde há presença de grupos, é importante usar métodos que se preocupem em lidar com ocultações.

Um destes métodos foi proposto por [43], onde cada pessoa é representada por um conjunto de manchas (*blobs*), correspondentes às principais partes do corpo, que por sua vez são representados pela sua distribuição de cor e pela sua localização relativa ao corpo total. Usando uma estimação de máxima verosimilhança, é determinado a que pessoa pertence um determinado píxel de *foreground*. Este método falhará quando, por exemplo, as pessoas passam objetos entre si ou se as pessoas adotarem posturas menos convencionais.

Métodos que lidem especificamente com o problema de deteção de pessoas em cenários com alguma densidade de pessoas, como em [44], combinam características retiradas a nível local e global para detetar pessoas mesmo sobre ocultações parciais. Tal permite complementar as deteções a nível local, que normalmente não distinguem pessoas que estão muito próximas, com a distinção possível com características globais.

2.4 Seguimento de pessoas

O seguimento de pessoas permite acompanhar a sua trajetória ao longo de uma sequência temporal, mantendo a correspondência dos objetos detetados ao longo das *frames*.

O método *Normalized Cross-Correlation* (NCC) caracteriza-se por usar os valores de intensidade da *bounding box* relativa ao objeto de interesse como referência para proceder à posterior correspondência. Em [45] é abordado um método baseado em NCC em que para cada *frame* são geradas janelas candidatas à volta da posição do objeto da *frame* anterior e é feita uma comparação com a referência usando correlação cruzada normalizada e a posição da janela que obtém melhor classificação é considerada a nova posição do objeto. Esta versão tem o inconveniente de não atualizar o modelo do objeto que se segue, podendo ficar desatualizado à medida que a sequência de vídeo avança e consequentemente comprometer a eficiência do seguimento.

No método Lucas-Kanade [46] é calculada uma transformada de afinidade, *affine-transformed match* entre a *bounding box* do objeto de interesse e as janelas candidatas. Posteriormente, é determinada qual a janela que tem maior afinidade com a do objeto sendo essa nova posição a nova posição do objeto. Um outro método para seguimento de pessoas é o Filtro de Kalman [47] e distingue-se por prever o estado de intensidade

em cada *frame*, através de um modelo de ruído gaussiano. Este método usa um modelo preditivo, atualizado com novas observações, permitindo obter uma janela de pesquisa onde o objeto provavelmente estará. O filtro de Kalman tem vindo a ser aplicado no seguimento de pessoas, como é o caso do *Kalman Appearance Tracker* (KAT) [48], em que é usado o filtro de Kalman para se prever a aparência do objeto a seguir de forma a resolver problemas relacionados com ocultações. O objeto é representado por um modelo de aparência, o movimento é modelado por uma translação a duas dimensões sem alteração da escala e a procura do objeto é feita em torno da sua posição da *frame* anterior. As janelas candidatas são comparadas com a janela de referência sendo escolhida a mais provável.

O método Mean-Shift [49] aplica-se a uma distribuição (por exemplo, histogramas). Para cada *frame*, os histogramas das janelas candidatas são comparados com os histogramas da janela do objeto através da métrica Bhattacharyya [50], que permitirá calcular a janela que, numa dada iteração, melhor se adapta à procura. A partir dessa nova janela, o método repete-se iterativamente até terminar o número máximo de iterações ou o deslocamento da janela ser inferior a um dado limiar.

O método de seguimento *Locally Orderless* (LOT) [51] estima automaticamente a quantidade de (des)ordem no objeto. Inicialmente, a *bounding box* do objeto a seguir é segmentada em super píxeis tal como é possível ver na Figura 2.5. Por sua vez, cada super píxel é representado através do seu centro de massa e dos valores HSV. Posteriormente, recorre-se ao filtro de partículas [52] com pesos gaussianos, onde cada partícula representa a janela onde os super píxeis são formados. A probabilidade de cada janela candidata é determinada através da Distância *Earth Mover* [53] entre a própria e os super píxeis. Esta função tem parâmetros que correspondem ao nível de flexibilidade. O estado seguinte é determinado através da média ponderada de todas as janelas de acordo com as probabilidades ponderadas. Segundo [54], o método, embora se mostre eficiente, revela-se computacionalmente pesado.

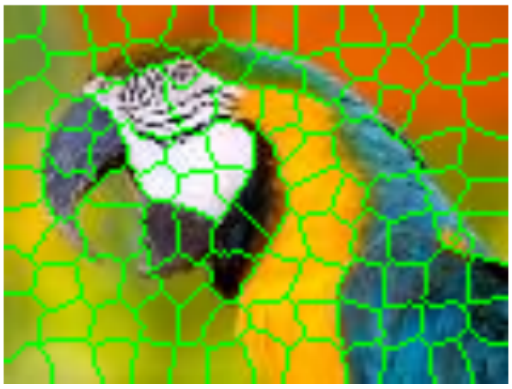


Figura 2.5: *Locally Orderless Method*: Imagem com limites dos super-píxeis (extraído de [51])

Na literatura existem outros métodos de seguimento de pessoas, tais como os referidos em [54].

2.5 Contagem de pessoas e estimação de densidade

Em [4], os modelos para estimação de densidade e contagem de pessoas são categorizados em três classes: análise do píxel, análise da textura e análise de objetos. Os métodos de contagem e estimação de densidade baseados na análise do píxel procedem a uma análise individual dos píxeis, sendo mais usados para a estimação de densidade de multidões e menos para a contagem de pessoas. É o caso da abordagem explicada em [7]. Inicialmente é usado o algoritmo *Mixture Of Gaussians* (MOG), baseado num método de modelização de fundo para gerar a máscara de *background* para cada imagem. É também aplicado um algoritmo de deteção de bordas às imagens originais. É usado o operador lógico AND entre as imagens obtidas após subtração de fundo e as imagem após deteção de bordas obtendo-se vários conjuntos de píxeis pertences ao contorno. Seguidamente, é feito um histograma de orientação de bordas em vez de se proceder à sua soma, o que permite que as bordas que dizem respeito às pessoas, geralmente verticais, sejam distinguidas de ruído, sombras e carros. Para estimação de densidade assume-se que todas as pessoas têm tamanho igual e que estão todas contidas num plano horizontal. Nesse plano 2D são atribuídos pesos diferentes a localizações diferentes. É feito uma calibração de câmara e um treino para se obter a relação entre as características assimiladas e o número de pessoas. Este treino é feito de forma controlada e o modelo de treino usado é *feed-forward-neural-networks* [55], um modelo não linear. Esta abordagem tem a vantagem de não requerer novo treino quando a câmara é reposicionada.

Nos métodos de análise de textura, em comparação com os de análise ao píxel, procede-se a uma análise mais geral, isto é, não se dá tanta relevância ao píxel individualmente mas sim aos conjuntos de píxeis que formam manchas. Estes métodos são geralmente usados para estimar o número de pessoas presentes numa região. Um exemplo de um método deste tipo é abordado [56] com o objetivo detetar densidades anormais de multidões através de aprendizagem baseada na análise da textura. É usado um modelo de projeção de perspetiva para gerar um conjunto de células multiresolução. Para tal, assumem-se quatro condições: as pessoas têm todas o mesmo tamanho, as pessoas estão todas contidas num plano horizontal, a câmara só se mexe no plano vertical e o centro da imagem é o centro ótico da câmara. A informação de textura é extraída de cada célula através do método *gray-level dependence matrix* [57]. Seguidamente, é construído um vetor de textura multi-escala e usa-se um algoritmo de classificação - SVM - para fazer corresponder as características de textura à densidade da cena. Como trabalho futuro, os autores propõem o uso de uma técnica de remoção de fundo na extração das características.

No artigo [58] é proposto um método que tem como objetivo estimar o número de peões que se movem em cada direção. É usado um algoritmo que utiliza uma técnica de

segmentação de movimento baseado em textura e uma regressão do processo Gaussiano. Inicialmente, esta técnica recorre a *mixture of dynamic texture* [59] para segmentar a multidão de acordo com as várias direções de movimento, obtendo-se vários *motion clusters*, um para cada direção; para cada um são extraídas algumas características de segmentação, características internas dos contornos e características de texturas. Estas são normalizadas para compensar a perspetiva da câmara e é usada uma regressão do processo Gaussiano para relacionar com o número de pessoas por segmento.

A análise do objeto baseia-se na procura de objetos na cena e apresenta melhores resultados do que as abordagens baseadas em análise de pixel e análise de textura quando se lida com densidades moderadas. Em ambientes de densidade elevada, com ocultação e desordem, o uso deste tipo de abordagem pode tornar-se ineficaz visto que a contagem de pessoas se torna impraticável perante estas condições.

No artigo [60] é proposto um algoritmo para fazer segmentação de pessoas em vídeo baseado no algoritmo de *expectation-maximization* [61]. São criadas múltiplas hipóteses para as posições de objeto para haver um segmentação correta tanto de crianças como de adultos. É usado o método *joint image likelihood* [62] de múltiplos objetos para lidar com as ocultações. As características das imagens são baseadas em contornos dos objetos do fundo. A *joint image likelihood* é obtida através do algoritmo *expectation-maximization*. Os autores sugerem para trabalho futuro a implementação de extração de características mais complexas. O algoritmo proposto tem a vantagem de funcionar com vários tipos de câmaras. A maior desvantagem a ele associada é a dependência da extração dos objetos segmentados.

Em [63] é apresentado um método para segmentar indivíduos numa multidão. É usado um algoritmo de seguimento de características chamado *Kanade-Lucas-Tomasi* (KLT) [64] que permite detetar objetos em movimento numa cena de forma simples e eficaz não requerendo subtração de fundo. Como este algoritmo obtém resultados com trajetórias pouco homogéneas, é usado posteriormente um filtro temporal e espacial e um algoritmo de *clustering* para agrupar características similares numa trajetória. Como trabalho futuro, é proposto uma combinação deste método com métodos de contagem de objetos estáticos, calibração automática de câmara e separação dos objetos do fundo.

Em [65] é proposta uma abordagem para detetar movimentos independentes numa multidão através do método *Bayesian clustering* [66] não supervisionado. Esta abordagem usa o pressuposto que dois pontos que se movem juntos pertencem à mesma entidade. Para a deteção são usadas as características de *Rosten-Drummond* [67] e *Tomasi-Kanade* [64]. Depois é usado um algoritmo de fluxo ótico juntamente com uma pesquisa exaustiva baseada na correlação cruzada normalizada para seguir as características entre imagens. Um algoritmo de *Bayesian Clustering* sem supervisão é usado para agrupar as características de forma a identificar cada indivíduo a mover-se na multidão embora o método não faça o seguimento das entidades. Esta proposta tem a vantagem de não precisar de treino e de não requerer o uso de modelos de aparência para fazer o seguimento de entidades.

Com a aplicação desta abordagem, dão-se falsos positivos quando as pessoas transportam objetos e falsos negativos quando duas pessoas se deslocam perto uma da outra. Quando as pessoas na cena mexem os seus braços de forma ampla, o algoritmo tende a falhar, mostrando-se eficiente apenas a lidar com movimentos rígidos.

Uma outra proposta de contagem de pessoas sem recorrer a treino é proposta em [68]. É feita uma calibração de câmara através da inserção de algumas amostras de alvos humanos. O modelo de câmara apenas requer três parâmetros: altura da câmara, ângulo *tilt-up* da câmara e a distância focal. A estimação da densidade da multidão é feita em dois passos: deteção e localização de pessoas na multidão e mapeamento dos alvos para um plano físico. No primeiro passo, inicialmente é criada uma máscara para detetar o bordo da multidão. Depois é feita uma comparação com o modelo de um alvo humano. No caso de ser muito maior do que esse, é identificado como uma multidão e procede-se então à estimação da multidão usando para tal uma lista de critérios definidos de acordo com o tipo de cena. Como o método consegue segmentar as pessoas individualmente, permite fazer a contagem e portanto estimar a sua densidade.

2.6 Seguimento de grupos

No artigo [69] é proposta uma abordagem de seguimento de grupos na presença de ocultações bem como deteção de eventos, tais como entrar em grupo, deixar o grupo, juntar e separar. Os autores dividem o algoritmo em duas partes: segmentação do *foreground* e depois seguimento. Na primeira parte, é considerada uma câmara estacionária e assume-se que o modelo de fundo é estático, sendo feita a subtração de fundo com base num modelo de fundo ideal. Nesta proposta, a correspondência de uma pessoa ou de um grupo de pessoas é feita de imagem em imagem através de dois critérios de similaridade: histograma de cor e localização. A correspondência usando a localização torna-se ineficaz quando há junção ou separação de entidades, sendo aí importante o uso do histograma de cor RGB (*Red*, *Green*, *Blue*). São usadas matrizes para auxiliar a deteção de eventos, que os representam matematicamente. São também guardados histogramas de cor das entidades presentes na cena, atualizando-se o histograma com o mais recentemente gerado da entidade respetiva. Os autores consideram que o algoritmo é eficiente a detetar separações e junções dos grupos embora não refiram em que circunstancias é que tal acontece.

Em [3], os autores propõem um algoritmo para fazer seguimento de grupos de pessoas num metro para reconhecimento de comportamentos anómalos de vandalismo ou violência. O algoritmo está dividido em duas partes: deteção de movimento e seguimento de grupo. A deteção de movimento está dividida em três partes. Numa primeira parte é usado um detetor de regiões móveis, sendo usada a subtração por uma imagem de referência. À diferença é aplicado um limiar (*threshold*), sendo depois filtrada e analisados os componentes ligados da imagem. Na segunda parte, é feita a extração de características: centro de gravidade, posição, altura e largura em 2D e 3D. Por último, temos a classificação das regiões móveis em classes relacionadas com pessoas (pessoa, pessoa parcialmente oculta, grupo e multidão) e outros objetos (metro, objeto de cena e ruído). No seguimento de grupo, é inicialmente definido o conceito de grupo através de quatro fatores: coerência de tamanho, coerência no espaço, coerência temporal e coerência de estrutura. No segundo passo, o seguimento de *frame* em *frame* fazer corresponder as regiões móveis de uma imagem às da seguinte. O sistema, segundo os autores, sabe lidar com eventos de separação e junção de grupos recorrendo para tal a regiões moveis (*moving regions*). O algoritmo tenta ainda calcular as trajetórias dessas regiões permitindo a criação, atualização e remoção das trajetórias. No terceiro passo, a estrutura do grupo irá corresponder a um conjunto de trajetórias ligadas temporalmente que correspondem a um grupo de pessoas. O artigo não detalha os critérios usados para a classificação nem o método usado para seguimento.

Um outro algoritmo de seguimento que lida com grupos é proposto em [70]. Este tem como objetivo final detetar movimentos anómalos de multidões. Os passos são: encontrar manchas de *foreground*, extração de características, modelação Bayesiana, deteção do evento de fuga. Quando não estão em fuga, as pessoas deslocam-se para os seus destinos mas quando ocorre um evento inesperado, desviam-se das suas trajetórias. São calculadas

funções de Densidade de Probabilidade (PDF- *Probability Density Function*) das posições e das direções dos fluxos. Conseguem-se modelizar os casos de fuga e não fuga e comparar os valores obtidos com esses casos. Os fluxos são caracterizados pelas PDFs. A deteção de eventos pode ser realizada com uma formulação Bayesiana. Eventos de fuga podem ser caracterizados por pessoas a afastarem-se de locais onde normalmente estão paradas onde se verifica o aumento da sua velocidade face ao normal e a mudança de direção face ao estado anterior. O algoritmo foi desenvolvido para lidar com multidões de baixa densidade, sendo proposto como trabalho futuro adaptar o algoritmo de forma a lidar com multidões mais densas.

Um algoritmo que analisa a interação entre grupos e procede ao seu seguimento é proposto em [71]. O método é dividido em quatro fases : deteção de pessoas; seguimento $frame$ a $frame$; deteção de grupos e seu seguimento; deteção de eventos. Os autores caracterizam os grupos através de três parâmetros: média das distâncias entre os elementos que compõem o grupo, desvios padrão da velocidade e desvio padrão da direção. Analisam os grupos nesses parâmetros dentro de uma janela temporal, por forma a detetar formações e dissociações de grupos. Dentro dessa janela, é feita uma análise da coerência do grupo ao longo das $frames$ que se encontram *à posteriori* da $frame$ a ser processada para assim se conseguir decidir melhor se o conjunto de pessoas que se está a testar é ou não grupo. A avaliação da coerência nas $frames$ *à posteriori* tem pesos que vão diminuindo de acordo com o seu espaçamento à $frame$ processada. Para reconhecimento de eventos, os autores propõem a ferramenta ScReK, *Scenario Recognition based on Knolwledge* que recebe como entrada os objetos de interesse, os modelos de cenário e operadores especiais se necessário e tem como vantagem a redução do tempo computacional.

Baseado no anterior, o algoritmo proposto em [72] tem como objetivo a deteção e seguimento de grupos baseados, mas não recorre à avaliação do grupo nas $frames$ que seguem a $frame$ processada. O método tem como entrada um conjunto de dados trajetórias dos objetos físicos . Essas trajetórias são obtidas através do algoritmo *Mean-Shift Clustering*. Para a deteção de eventos é usada a linguagem de reconhecimento de eventos (ScReK) tal como em [71]. Segundo os autores, o algoritmo funciona em tempo real e em vídeos longos. Para trabalho futuro é proposta a adição de probabilidades ao reconhecimento de eventos.

No artigo [73] é proposto um algoritmo de deteção e reconhecimento comportamento de grupos também baseado em [71]. Segundo os autores, um dos problemas associados ao seguimento de grupos está relacionado com grupos que saem e voltam a entrar no plano de captura da imagem. Então propõem um processo de re-aquisição que tenta usar identificadores de grupos anteriores para resolver este problema e descritores baseado nas matrizes de co-variância [74].

Uma solução apresentada em [75] usa uma *framework* que contém dois sub-espaços, um para indivíduos e outro para grupos, que partilham informação entre si. Para tal, é adaptado o *Decentralized Particle Filter* (DPF) [76] para o contexto do problema, onde

se definem variáveis para estado dos indivíduos, definidos pela sua posição e velocidade, e para grupos, associando uma identificação de grupo a cada indivíduo. Permite ainda prever de forma probabilística o estado de um indivíduo tendo a informação anterior ou o estado do grupo perante as novas observações. Os resultados obtidos usa informação conjunta de grupo e indivíduo e apresenta em geral melhores resultados que usando apenas uma das duas componentes.

O *Multiple Hypothesis Tracking* (MHT) [77] apresenta uma adaptação de um algoritmo criado para resolução de problemas de associação de dados e que tem atualmente sido usado no âmbito de seguimento de múltiplos objectos *Multiple Target Tracking*, MTT. Em MHT, caso a associação entre a observação atual e as trajetórias anteriores não seja ideal, isto é, quando o algoritmo não consegue, com alta fiabilidade, associar um elemento do passado à observação atual, são criadas múltiplas hipóteses, propagadas ao longo do tempo que, usando novos dados, irão apoiar a decisão do momento passado. Embora não seja referido no artigo, o MHT tem como principais desvantagem o custo computacional elevado e a complexidade. No artigo [78] é proposto um algoritmo para deteção de grupos e de eventos de separação e junção de grupos baseado no MHT. Segundo os autores, é usado o MHT devido às suas vantagens, tais como a capacidade de conseguir processar dados em ambientes densos . Inicialmente, com o conjunto de trajetórias de entrada, são procurados grupos candidatos, isto é, um conjunto de trajetórias próximas umas das outras. Esses grupos candidatos são avaliados ao longo do tempo, e se convergirem para um único *cluster*, são considerados como grupo. O seguimento do grupo é feito baseando-se em [79]: as diferentes trajetórias pertencem a um estrutura e as pessoas que pertencem a um grupo são todas seguidas como se de apenas uma entidade se tratassem. O seguimento do grupo é feito através do filtro de Kalman. Como vantagens, os autores referem o facto da inicialização dos grupos ser mais estável e preciso, devido ao uso do MHT, a diminuição da complexidade computacional e a habilidade de lidar com áreas com grande densidade de alvos.

Um algoritmo de seguimento de grupo baseado no modelo tradicional de seguimento de grupo é proposto em [80]. O grupo é seguido como uma única entidade possibilitando a obtenção da distribuição dos objetos que pertencem ao grupo através da tendência do grupo. A Figura 2.6 descreve por meio de um diagrama de blocos o algoritmo. São recolhidas as informações das deteções na inicialização, e são agrupadas em grupos no módulo *Agrupamento dentro do Grupo*. É calculado o centro do agrupamento e o seu peso e essa informação é passada ao modulo seguinte que irá calcular o centro do grupo. Com essa informação, o módulo *Seguimento e previsão do centroide* procederá à previsão do centro do agrupamento na *frame* seguinte. Por fim, no último bloco é feita uma estimativa da posição do grupo no tempo futuro através da velocidade do conjunto. Essa previsão servirá como semente para o próximo movimento.

No artigo [6] é proposto um método de deteção e localização de comportamentos anómalos em multidões usando para tal *Social Force Model*. Inicialmente é posta uma grelha

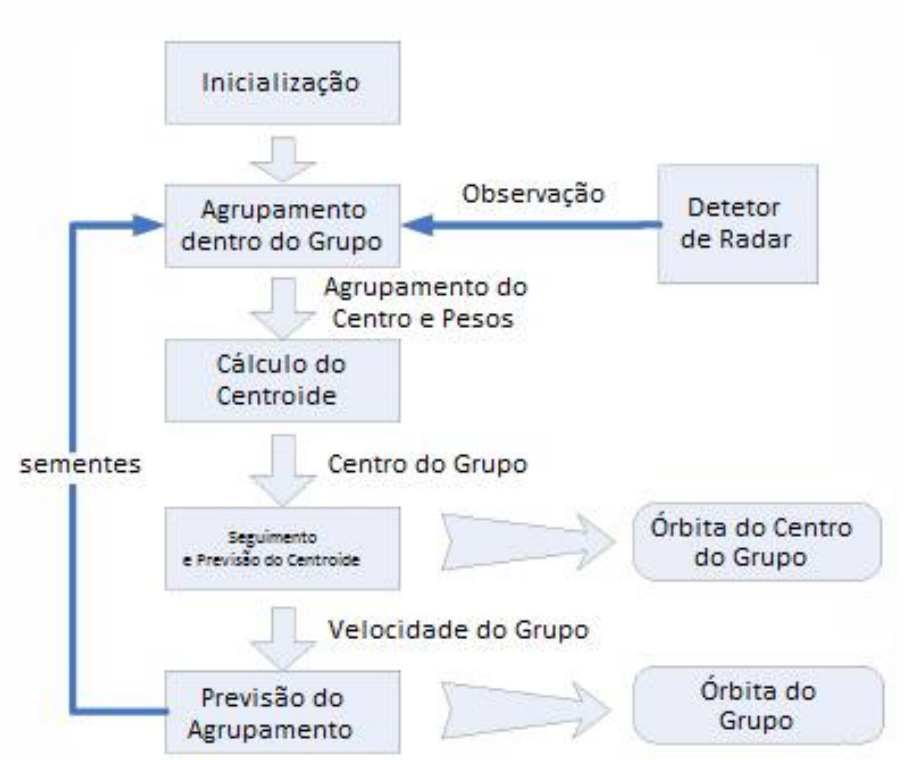


Figura 2.6: Diagrama de blocos correspondente ao algoritmo proposto em [80] (adaptado de [80])

de partículas em cima de cada imagem. Essa grelha sofrerá alterações com a média espaço-tempo do *optical flow*, tal como mostra a Figura 2.7. As partículas em movimento são tratadas individualmente e as suas forças de interação calculadas através de um modelo de força. Depois de calculadas todas as forças, é feito um mapa dentro do plano de imagem. Com este mapa, consegue-se obter o *Force Flow* para cada pixel em cada *frame*. Por fim, à uma seleção de conjuntos de espaço-temporais para serem usados como modelos de comportamento anormal de uma multidão. As diferentes *frames* são então classificadas como comportamento normal ou anormal através do método de classificação *bag of words*.

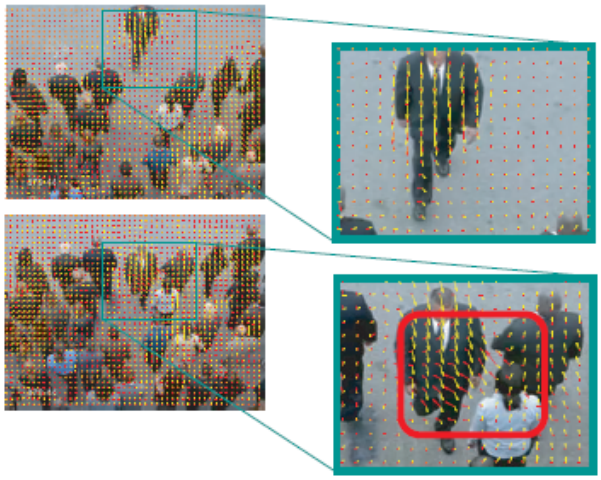


Figura 2.7: O *optical flow*, a amarelo, e o cálculo dos vetores de interação das forças, a vermelho, de duas *frames* (extraído de [6])

2.7 Algoritmos de fusão

Embora se continue a investir no desenvolvimento de algoritmos de seguimento de pessoas, sabe-se que é difícil, senão impossível, encontrar um algoritmo que se adapte com eficiência às mudanças que se dão na sequência e que mantenha uma boa performance em qualquer circunstância. Uma forma de contornar este problema passa por apoiar as decisões do seguimento em vários algoritmos, desenvolvendo *frameworks* que usem múltiplos métodos de seguimento, por forma a melhorar o desempenho global. Uma abordagem comum nos algoritmos de fusão é o sistema de votação tal como em [81]. Este algoritmo usa como entrada a informação das caixas delimitadoras dos objetos de interesse, *bounding boxes*, podendo assim lidar com algoritmos de seguimento de qualquer tipo. Inicialmente, é feita uma votação onde são calculadas as distâncias das posições entre as *bounding boxes*. As distâncias que forem menores do que um dado *threshold*, votarão para a mesma posição. Se pelo menos 50% das *bounding boxes* votarem para a mesma posição, o resultado é classificado como viável. Esta abordagem mostra-se demasiado dependente da sequência por ser difícil encontrar um valor de *threshold* eficiente para qualquer situação. Com o intuito de não requerer o uso de um *threshold*, em [82] adota-se uma abordagem baseada na atração dos campos. Dessa forma, a posição resultante do algoritmo de fusão será mais atraída pelas *bounding boxes* dos algoritmos de seguimento mais próximos. Seguidamente, é calculada uma função de energia que será a soma dos valores das atrações do resultados dos seguimentos. O resultado de fusão, será aquele que maximiza essa energia. Nesta abordagem, são dados diferentes pesos aos algoritmos de seguimento com base numa análise de eficiência. Como, por vezes, há algoritmos de seguimento que prejudicam o processo de fusão sem acrescentarem vantagem, esta abordagem permite também excluir algoritmos. A exclusão pode ser local, isto é, são retirados diferentes algoritmos de seguimento para

cada sequência de vídeo ou então de forma global, excluindo-se o algoritmo para todas as sequências. De forma semelhante, em [83], para calcular a *bounding box* resultante da fusão dos algoritmos, é feita uma média ponderada em que os pesos são calculados através de um coeficiente de credibilidade. Este coeficiente tem em conta a consistência de cada algoritmo de seguimento entre *frames* consecutivas e entre pares de algoritmos.

Em vez de se fazer seguimento apoiado no resultado de diferentes algoritmos, é possível para cada *frame* escolher o melhor algoritmo que segue os objetos de interesse tal como proposto em [84]. O algoritmo de seguimento abordado é constituído por quatro componentes: um modelo de aparência, um modelo de movimento, uma representação de estado e um modelo de observação. Seguidamente, há uma divisão de cada componente em sub-componentes. O estado do objeto contém o centro, a escala e a informação espacial que contém a projeção das bordas verticais. O modelo de movimento é composto por múltiplas distribuições de Gauss. O modelo consiste na observação das respostas do filtro de Gauss às características de intensidade. Os algoritmos de seguimento básicos são formados a partir da combinação das quatro componentes. Em cada *frame*, o algoritmo que tiver melhor estado do objeto é o escolhido para fazer o seguimento.

Quando é necessário seguir um objeto a longo prazo pode ser útil ter um modelo de aparência ou de comportamento que tenha em atenção alterações para assim tornar o seguimento mais robusto. Assim, o algoritmo proposto em [85] lida com seguimento de objetos que sofrem de mudanças significativas de aparência. O algoritmo trabalha a dois níveis diferentes: local e global. A nível local o objeto é dividido em manchas (*patches*) caracterizados através de um histograma de cinzas e da sua localização espacial. Em cada nova *frame* é usado o filtro de Kalman e uma transformação afim, isto é, transformada linear seguida de uma translação, para prever a localização das *patches*. Posteriormente é feita uma comparação dessa com o objeto e quando se mostra muito distante essa *patch* é removida. A nível global, são usados histogramas de cor HSV (Hue, $Saturation$, $Value$) para descrever o objeto e para o fundo. É mantido um modelo probabilístico com descrição de cor, forma e movimento que é atualizado ao longo do processo de seguimento do objeto. Para definir o movimento é usado do *optical flow* do conjunto de pontos salientes do KLT. Na camada local é usado o modelo probabilístico da camada global para dar pesos a cada *patch*. Recorrendo à informação dos dois níveis, é calculada a probabilidade do píxel pertencer ao objeto.

Uma abordagem alternativa de seguimento de objetos recorre a algoritmos com aprendizagem. No artigo [86] é proposto uma possível método de seguimento através da combinação dos resultados da deteção e do seguimento através de *optical flow*. Para fazer a aprendizagem do detetor é usado o método *Random Ferns* [87]. O método *optical flow* aplica um KLT à região do objeto a ser seguido e estima a posição da nova janela na *frame* atual. A cada janela candidata é feita uma normalização cruzada correlacionada, e é selecionada aquele que tiver o maior nível de similaridade com o modelo do objeto. Quando o objeto é localizado procede-se então à aprendizagem: as amostras positivas são

aqueles em volta do objeto e como amostras negativas usam-se pontos mais afastados. Esta abordagem mostra-se eficiente a lidar com ocultações de pequena duração.

Capítulo 3

Sequências de dados e Métricas

3.1 Sequências Disponíveis

As sequências de dados a serem utilizadas deverão ser o mais realistas possíveis, pois devem representar de forma fiel uma aplicação prática do sistema a desenvolver. Assim, terão de seguir certas diretrizes de forma a garantir uma boa simulação de eventos reais. A cena contemplará várias pessoas em grupo, sozinhas, em movimento e paradas. Deverão surgir interações entre grupo-pessoa tais como: pessoa a juntar-se ou separar-se de um grupo, dois grupos a fundirem-se ou um grupo a separar-se. Na cena terão de surgir situações de ocultação parcial e total de pessoas de forma a testar a robustez do sistema podendo estas ser provocadas por outras pessoas ou por objetos presentes na cena. Se tal não for possível, deverão ser simuladas ocultações (por exemplo, por meio de adição de erros).

As sequências a ser usadas poderão ser fruto de uma captura de vídeo propositadamente feita para o efeito ou então ser extraídas de conjunto de dados como CAVIAR [88] visto cumprir uma parte considerável dos requisitos necessários, tal como se pode ser na Figura 3.1 em que à esquerda vemos vários grupos de pessoas e à direita uma pessoa que não pertence a nenhum grupo.

Figura 3.1: Exemplos de *frames* extraídas do conjunto de dados CAVIAR ([88])

No conjunto de dados BIWI [89] são capturadas sequências num local com constantes movimentações de pessoas. Estas sequências, usadas em [90], estão manualmente anotadas tanto com informação de indivíduos como de grupos. Na Figura 3.2 podemos ver dois exemplos de imagens deste conjunto de dados.

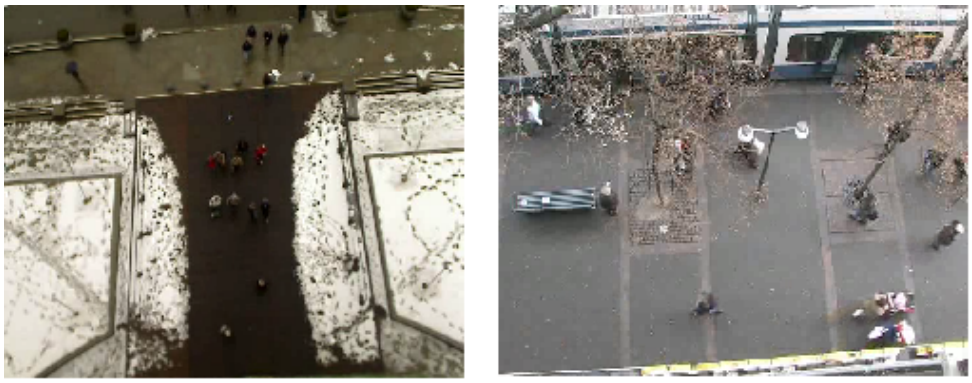


Figura 3.2: Exemplos de *frames* extraídas do conjunto de dados BIWI (extraído de [89])

Friends meet é um conjunto de dados usado em [75], e detalhado em [91], que tem anotações disponíveis de sequências específicas para algoritmos de grupos. As anotações incluem informação de deteção e seguimento tanto de indivíduos como de grupos tal como podemos ver na Figura 3.3. Nas sequências incluem-se as interações sociais mais predominantes.

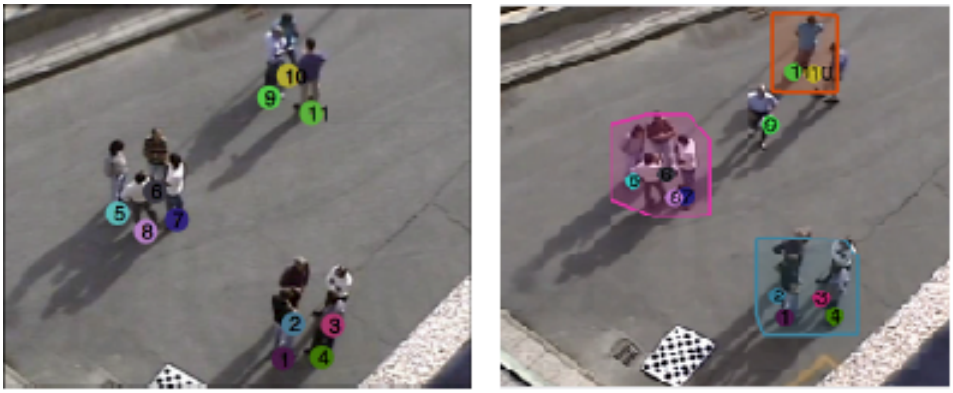


Figura 3.3: Exemplos de *frames* extraídas do conjunto de dados *Friends meet* anotado (extraído de [91])

3.1.1 Descrição das sequências usadas

As sequências usadas pertencem ao *Dataset*, *Friends meet* [91]. Estas simulam situações de uma festa de *cocktail* onde se dão frequentemente eventos de junção e separação, seja de grupos ou de pessoas. Na Figura 3.4 podem-se ver cenários típicos das sequências usadas.

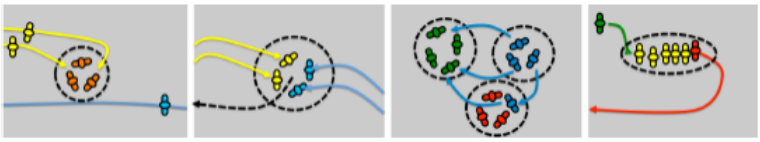


Figura 3.4: Cenários típicos no *Friends Meet*: separação e junção de pessoas e grupos e filas (Extraído de [75])

O facto de simular a interação entre pessoas e grupos e de as sequências terem sido cedidas com as devidas anotações são o motivo de escolha destas em detrimento de outras possíveis.

O conjunto de sequências retratam um ambiente exterior, *outdoor*. Essas têm duração variável entre 30 segundos e 1.5 minutos e contêm entre três e onze indivíduos sozinhos e em grupos.

Na Figura 3.5 é mostrado o fundo, comum em todas as sequências, sem a presença de atores. Encontra-se devidamente marcada com as zonas de entrada e/ou saída de pessoas com as letras A, B, C, D e E para facilitar a descrição de todas as sequências usadas.

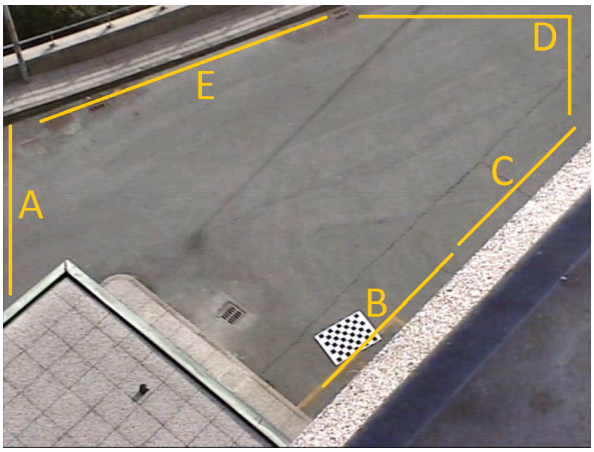


Figura 3.5: Cena sem atores

Todas as sequências têm uma resolução de [960 X 720] e uma *frame rate* de 29 *frames* por segundo. A Tabela 3.1 apresenta um quadro-resumo de cada sequência onde se inclui a duração, o número total de pessoas, se ocorre a junção de uma pessoa a um grupo, se ocorre junção de grupos (onde se inclui a junção de dois grupos num), se uma pessoa se separa do grupo ou se um grupo se divide em vários grupos.

A sequência I inicia-se com um grupo composto por quatro pessoas parado na zona B. Depois, aparecem dois grupos em cena, cada um composto por duas pessoas, um pela zona A movimentando-se para a zona D e o outro pela zona D movimentando-se em direção à zona A. Estes dois grupos encontram-se a meio da cena formando um novo grupo que aí permanece por uns segundos. Depois dirigem-se para a zona de saída B onde desaparecem de cena.

#	**Duração (mm:ss)**	**Número de Pessoas**	**Junção de Pessoa a Grupo?**	**Junção de Grupos?**	**Separação de Pessoa de Grupo?**	**Separação de Subgrupo de Grupo?**
I	00:22	8	Não	Sim	Não	Não
II	00:26	3	Não	Sim	Não	Não
III	00:18	4	Não	Não	Não	Sim
IV	00:12	8	Não	Não	Não	Sim
V	00:22	4	Não	Não	Sim	Não
VI	00:37	8	Sim	Sim	Sim	Não
VII	00:13	4	Não	Sim	Não	Não
VIII	00:17	8	Não	Sim	Não	Não
IX	00:11	4	Não	Sim	Não	Não
X	00:20	6	Não	Sim	Não	Não
XI	00:17	6	Não	Sim	Sim	Não
XII	01:10	9	Sim	Sim	Sim	Não
XIII	00:29	11	Sim	Sim	Sim	Sim

Tabela 3.1: Tabela resumo das sequências usadas para análise

Na sequência II entram duas pessoas em cena uma pela zona A e outra pela D, movimentando-se uma em direção à outra. Quando estão próximas da zona B, uma pessoa entra por essa zona e aí as três pessoas em cena formam um grupo. Este grupo mantém-se imóvel durante uns momentos, deslocando posteriormente por forma a abandonar a cena pela zona B.

A sequência III inicia-se com um grupo em cena composto por quatro elementos. Esse grupo mantém-se na mesma posição durante uns momentos, separando-se depois em dois grupos. Um sai de cena pela zona A e outro pela zona D.

No caso da sequência IV, estão dois grupos presentes na cena, cada um com quatro elementos: um perto da zona B e outro a meio da cena. Após alguns segundos sem mudar de posição, este último grupo divide-se em dois grupos e cada um desses grupos formados recentemente dirigem-se para a zona A e D de onde desaparecem de cena. O grupo que se encontrava perto da zona B, permanece na mesma posição durante toda a sequência.

Na sequência V encontra-se em cena um grupo composto por quatro pessoas. A dado momento, uma das pessoas que compõe o grupo, abandona-o, dirigindo-se para a zona de saída B, de onde desaparece.

A sequência VI começa com um grupo composto por três pessoas que entra em cena entre a zona B e C. Passado uns segundos, entra uma pessoa pela zona B seguida de um grupo composto por quatro elementos. A pessoa e o grupo que a seguia, acabam por se juntar num novo grupo. Após alguns momentos, uma pessoa do grupo recém formado, abandona o grupo juntando-se ao outro que já se encontrava em cena. O grupo que perdeu um elemento, desloca-se para a zona B onde desaparece de cena.

Na sequência VII aparecem dois grupos em cena, cada um com dois elementos. Um grupo aparece pela zona de entrada A e outro pela D. O grupo que entrou pela entrada A

movimenta-se para a zona D, e o outro grupo para a zona A. A meio da cena cruzam-se.

A sequência VIII inclui inicialmente um grupo em cena composto por quatro pessoas perto da zona B. Passado uns segundos, entram dois grupos compostos por duas pessoas cada, um pela zona A e outro pela zona D. O grupo que entrou pela entrada A movimenta-se para a zona D, e o outro grupo para a zona C por forma a desaparecerem da cena. Entretanto uma pessoa entra pela zona B, movimentando-se para a zona E.

Na sequência IX entram dois grupos, um pela zona A e outro pela zona D, compostos por duas pessoas cada. O grupo que entrou pela entrada A movimenta-se para a zona D e o outro grupo para a zona A por forma a desaparecerem da cena. A meio da cena há um cruzamento entre os grupos.

Na sequência X encontra-se um grupo composto por seis pessoas em fila indiana orientada em paralelo com a zona B como mostra a Figura 3.6.

Figura 3.6: Sequência X do *dataset Friends Meet*: simulação de fila indiana

A sequência XI tem em cena um grupo em fila indiana orientada em paralelo com a zona B. Depois a pessoa que está no topo da fila, abandona o grupo em direção à zona D, onde sai de cena, e uma pessoa vinda da zona D junta-se ao grupo.

Na sequência XII encontra-se, inicialmente, um grupo composto por quatro elementos perto da zona B. Depois aparece um grupo pela zona A constituído por quatro elementos dispostos em linha. Esse grupo movimenta-se para o centro da cena onde se dispõem em circulo. Entretanto, o grupo que se encontrava na zona B é abandonado por um elemento que sai de cena pela zona B e um elemento do grupo que se encontra no meio da cena abandona o seu grupo juntando-se ao outro. Esse elemento, passado uns momentos, abandona esse grupo e junta-se ao seu grupo inicial. Os dois grupos depois de permanecerem uns segundos parados, retiram-se de cena pela zona B e uma pessoa entra em cena pela mesma zona, movimentando-se para a zona E.

No caso da sequência XIII encontram-se três grupos em cena: dois com quatro elementos e um com três. O grupo composto por três pessoas a certo momento desintegra-se: um

elemento do grupo junta-se a um dos grupos de quatro elementos, os dois outros elementos permanecem uns momentos sem pertencerem a nenhum grupo e depois juntam-se cada um a um grupo. Por fim, um dos elementos que se juntou ao grupo separa-se desse e junta-se ao outro grupo.

3.2 Métricas Disponíveis

O uso de métricas é importante para demonstrar a robustez e a eficiência dos algoritmos, permitir comparar o desempenho de algoritmos e avaliar a melhoria quando se proceder a um aperfeiçoamento de um algoritmo [92].

Em [93] é apresentado um procedimento para a avaliação sistemática e objetiva das características dos algoritmos de seguimento. Propõem assim para cada *frame* t se proceda aos seguintes passos:

- estabelecer a melhor correspondência entre a hipótese h_j e o objeto o_i;
- para cada correspondência, calcular o erro;
- acumular todos os erros ao longo das *frames*.

Segundo os mesmo autores, uma correspondência entre o objeto o_i e hipótese h_j não deve ser feita quando a distância $dist_{ij}$ excede um dado *threshold*, T. Para algoritmos de seguimento que estimam também o tamanho dos objetos, os autores propõem que a distância seja calculada usado para tal a sobreposição das *bounding boxes* do objeto estimado com a *bounding box* do objeto de *ground truth*, tal como proposto em [94] e quando apenas estimam a posição do centroide do objeto, propõem a distância euclidiana.

Em [95], é proposta outra forma de fazer as correspondências: Dado o *ground truth* GT_i e a estimação de trajetória ε_i, definem *recall* conforme a Equação 3.1, onde $\rho_{i,j}$ avalia quanto do GT_j é coberto por ε_i, tomando valores entre 0 (sem sobreposição) e 1 (total sobreposição). É também definida uma medida de precisão, $\nu_{i,j}$, na Equação 3.2, que avalia quanto de ε_i é coberto por GT_i.

$$\rho_{i,j} = \frac{|\varepsilon_i \bigcap GT_j|}{|GT_j|} \tag{3.1}$$

$$\nu_{i,j} = \frac{|\varepsilon_i \bigcap GT_j|}{|\varepsilon_i|} \tag{3.2}$$

Na Figura 3.7, é ilustrada a relação precisão/*recall*. Como se pode ver, é possível ter valores de precisão altos e *recall* baixos num mesmo caso ou vice-versa.

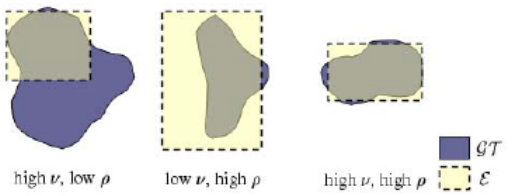


Figura 3.7: Relação precisão(ε_i) e *recall* (ρ_j). À esquerda podemos ver a relação entre uma precisão alta e um *recall* baixo. Ao meio temos baixa precisão e alto *recall* e por último valores altos das duas medidas. (Extraído de [95])

Usando estas medidas de precisão e de *recall*, os autores propõem uma correspondência $GT_i \longrightarrow \xi_i$ feita segundo o seguinte pseudo-algoritmo:

---

para cada estimação, ξ_i

- calcular F com cada objeto de *ground truth* da seguinte forma $F_{i,j} = \frac{2\nu_{i,j}\rho_{i_j}}{\nu_{i,j}+\rho_{i_j}}$
- $GT_i \longrightarrow \xi_i$ se $F_{i,j} > t_c$

Em [96] é proposta uma outra forma de proceder às correspondências baseada na sobreposição das *bounding boxes*: Sendo GT_{ik}, a entidade i de *ground truth* e ST_{jk}, a entidade j gerada pelo algoritmo, e A(GT_{ik}, ST_{jk}), a área de sobreposição, temos a Equação 3.3 representado na Figura 3.8:

$$A(GT_{ik}, ST_{ik}) = \frac{Area(GT_{ik} \cap ST_{jk})}{Area(GT_{ik} \cup ST_{jk})} \tag{3.3}$$

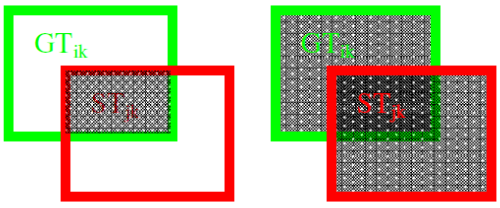


Figura 3.8: Área($GT_{ik} \cap ST_{jk}$) e Área($GT_{ik} \cup ST_{jk}$) (Extraído de [96])

Depois de concluídas as correspondências entre os objetos e o *ground truth*, é necessário encontrar métricas que procedam a uma avaliação correta. Na literatura encontram-se vários tipos de métrica de avaliação de deteção de movimento e o seguimento dos objetos.

Em [97] são propostas algumas métricas para seguimento. Os autores propõem uma correspondência espacial entre os objetos definidos como hipótese e os objetos da cena determinada pela máxima distância entre o centro das coordenadas e os objetos, para pontos

suficientemente afastados. Baseando-se em verdadeiros e falsos positivos, são propostas quatro métricas espaciais para medir o desempenho:

$$\text{taxa de acerto} = \frac{\text{Número total de objetos detetados em toda a sequência}}{\text{Número de objetos do } \textit{ground truth}} \tag{3.4}$$

$$\text{Taxa de erro} = \frac{\text{Número total de objetos não detetados em toda a sequência}}{\text{Número de objetos do } \textit{ground truth}} \tag{3.5}$$

$$\text{Tentativas falhadas} = \frac{\text{Número total de hipóteses que o algoritmo detetou erradamente como objeto}}{\text{Número total de objetos}} \tag{3.6}$$

$$\text{Mudanças de correspondências} = \frac{\text{Número de mudanças de correspondência}}{\text{número correto de mudanças}} \tag{3.7}$$

A métrica que diz respeito à equação 3.7 foi criada pela necessidade de avaliar os algoritmos relativamente às correspondências que fazem entre *frames*, visto que é possível ao longo da sequencia, o algoritmo fazer corresponder erradamente os objetos (por exemplo, seguir um objeto A como sendo um B).

Em [97] são ainda propostas outras métricas:

$$\text{Taxa de sucesso de ocultações} = \frac{\text{Número total de ocultações dinâmicas bem sucedidas}}{\text{número total de ocultações}} \tag{3.8}$$

$$\text{Taxa de sucesso do seguimento} = \frac{\text{Número total de objetos não fragmentados}}{\text{número de objetos do } \textit{ground truth}} \tag{3.9}$$

Em [92] são propostas outras métricas de precisão e *recall* tais como definidas nas Equações 3.10 e 3.11, onde Positivos Verdadeiros (*True Positives*, TP) são o número de entidades estimadas que têm correspondência com as entidades de *ground truth* e Falsos Positivos (*False Positives*, FP), o número de entidades que não correspondem ao *ground truth* e por fim Falsos Negativos (*False Negatives*, FN), o número de entidades não detetadas.

$$Precision = \frac{TP}{TP + FP} \tag{3.10}$$

$$Recall = \frac{TP}{TP + FN} \tag{3.11}$$

Visto não terem sido encontradas métricas especializadas em avaliar a performance de algoritmos de seguimento de grupos, em [75] é usada uma métrica baseada em [95]. Este último propõe:

- *False Positive* (FP): objeto que não corresponde a nenhum objeto de *ground truth*;
- *False Negative* (FN): objeto do *ground truth* não associado a nenhum objeto estimado;
- *Multiple Trackers* (MT): duas ou mais estimações que se associam a um mesmo objeto de *ground truth*;
- *Multiple Objects* (MO): dois ou mais objetos de *ground truth* que associam ao mesmo objeto estimado.

Em [96] é também proposta a métrica *Track Fragmentation* (TF), fragmentação da trajetória, que tem como objetivo contabilizar o número de falhas na continuidade de uma trajetória do *ground truth*. Idealmente, o seu valor seria zero, significando assim que o algoritmo produziu uma trajetória continua da entidade. Na Figura 3.9 podemos ver um exemplo a avaliação da fragmentação para a trajetória de um objeto. Visto que a trajetória gerada pelo sistema (linha tracejada) está fragmentada duas vezes quando comparada com a trajetória do *ground truth* (linha contínua), *Track Fragmentation* tem valor dois.

Figura 3.9: O número de fragmentações da trajetória é 2 (FN=2): quando comparada a trajetória do objeto de *ground truth* (linha contínua) e a trajetória do objeto estimado pelo sistema de seguimento (linha tracejada), vemos que esta ultima foi descontinuada duas vezes (Extraído de [96])

Por fim, e em relação à deteção de grupos, em [75] é proposta a métrica *Group Detection Success Rate* (GDSR). Esta representa a taxa de deteção correta de grupos, sendo que é estipulado que um grupo é corretamente detetado se contiver pelo menos 60% dos membros do grupo do *ground truth* [98].

3.2.1 Métricas Usadas

O uso de métricas é fulcral para se poder avaliar quantitativamente a performance do algoritmo e para quantificar a melhoria das alterações neste implementadas. Assim, numa primeira fase é necessário avaliar as melhorias implementadas no algoritmo e numa segunda fase comparar as avaliações com as do artigo [75].

Respeitando as métricas usadas em [75], por forma a tornar os algoritmos comparáveis, a correspondência entre objetos estimados e objetos de *ground truth* será feita de acordo com o artigo [96]. Esta correspondência feita através do cálculo de $F_{i,j}$, baseada numa medida de precisão e outra de *recall*, tal como já explicado na Secção 3.2. Visto que nem em [96] nem em [75] foram propostos valores de t_c, para efeitos de avaliação, estipulou-se que $t_c = 75\%$.

Serão usadas as métricas Verdadeiros Positivos (TP), Falsos Positivos (FP) e Falsos negativos(FN) baseadas em [96]. Na Figura 3.10, vê-se o mapa de configuração em que as entradas das tabelas dizem respeito às associações feitas entre *ground truth*, GT, e os objetos estimados, ε. Este mapa tem em consideração quatro erros: Falsos positivos (FP), Falsos Negativos (FN), Objetos Múltiplos, *Multiple Object* (MO) e Múltiplas Trajetórias, *Multiple Track* (MT). Visto não se pretender contabilizar estes últimos dois erros, foram feitas algumas alterações. Quando encontrada uma correspondência $GT \longrightarrow \varepsilon$ de $1 \longrightarrow 1$ essa será contabilizada como TP. Se for uma correspondência do mesmo tipo mas de $0 \longrightarrow 1$, considerar-se à um FP. Se, pelo contrário, houver uma correspondência $\varepsilon \longrightarrow GT$, será contabilizada como um FN. No caso em que é encontrado o erro MO, isto é, sempre que houver uma correspondência $\varepsilon \longrightarrow GT$ de $1 \longrightarrow n$, apenas se considera como correspondência válida (TP) aquela que tiver maior valor de sobreposição. Os restantes serão contabilizados como sendo FN. De forma semelhante, sempre que detetado um erro MT, isto é, se encontrada uma correspondência $GT \longrightarrow \varepsilon$ de $1 \longrightarrow n$, a entidade de *ground truth* à qual esteja associada a maior sobreposição com a entidade estimada será considerada TP e as restantes serão contabilizadas como FN.

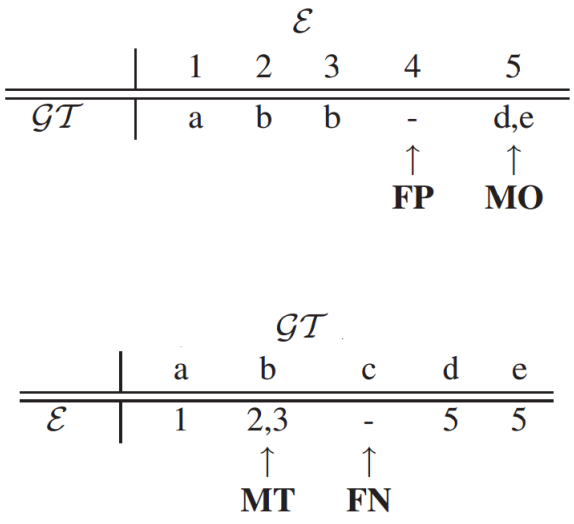


Figura 3.10: Mapa de configuração em que cada entrada é uma correspondência válida, isto é, a sobreposição entre as entidades foi superior a t_c. Na tabela superior temos as correspondências $GT \longrightarrow \varepsilon$ com a deteção dos erros FP e MO (quinta e sexta coluna da tabela, respetivamente). Na tabela inferior tem-se as correspondências $\varepsilon \longrightarrow GT$ e a deteção dos erros MT e FN (terceira e quarta coluna respetivamente) [Figura baseada de [96]]

Feitas todas as correspondências e averiguados todos os erros, serão calculadas as percentagens para cada sequência usada (x) segundo as Equações 3.12, 3.13 e 3.14.

$$TP_{seq_x} = \frac{1}{nFrames} \sum_{n=1}^{n=nFrames} \frac{TP_n}{\sum_{j=1}^{j=nEntidadesGT} GT_j} \tag{3.12}$$

$$FP_{seq_x} = \frac{1}{nFrames} \sum_{n=1}^{n=nFrames} \frac{FP_n}{\sum_{i=1}^{i=nEntidades\varepsilon} \varepsilon_i} \tag{3.13}$$

$$FN_{seq_x} = \frac{1}{nFrames} \sum_{n=1}^{n=nFrames} \frac{FN_n}{\sum_{j=1}^{j=nEntidadesGT} GT_j} \tag{3.14}$$

Estas métricas serviram para calcular duas outras, precisão e *recall* respeitando o proposto em [97]. Estas métricas encontram-se já explicadas na Secção 3.2.

A métrica *Track Fragmentation* [96] também será usada para avaliar o algoritmo. Como explicado na Secção 3.2, a fragmentação é dada para cada trajetória. Visto que o somatório de todas as fragmentações não é um bom cálculo para comparação entre sequências (note-se que cada sequência tem um número diferente de entidades, logo de trajetória), serão calculadas as fragmentações médias por entidade para as sequências. Por exemplo, se a *Track Fragmentation* média para um dada sequência for 2, concluir-se-á que em média o algoritmo implementado fragmenta duas vezes cada trajetória. Por forma a facilitar a visualização da evolução desta métrica, é feita uma transformação que coloque estes valores numa escala de 0 a 1, com qualidade crescente (onde 1 representa uma situação ideal), conforme definida na Equação 3.15.

$$TF^* = 1 - \frac{TF}{1+TF} \tag{3.15}$$

Por fim, para se poder comparar algoritmo proposto com o do artigo que introduz o *dataset* [75], será calculada a métrica *Group Detection Success Rate* (GDSR). Respeitando os mesmo autores, considerar-se-á que grupo é corretamente detetado se contiver pelo menos 60% dos membros correspondentes do *ground truth*.

3.3 Preparação do *ground truth*

No desenvolver da presente tese, foram usados dois tipos de *ground truth* com tipos de informação e objetivos diferentes.

Primeiro foi usado um *ground truth* com informação das pessoas presentes nas diferentes cenas. Esta informação é usada como entrada do algoritmo de deteção de grupos. Este *ground truth*, fornecido com as sequências, encontrava-se em formato .Mat, ficheiro próprio

da ferramenta *Matlab*. O *OpenCv*, ferramenta usada na elaboração do algoritmo, não se encontra preparado para a leitura de ficheiros .Mat, pelo que foi necessário converter num outro legível. Assim, usando o *Matlab*, foi criado um script para conversão de .Mat para .csv , *comma separate values*.

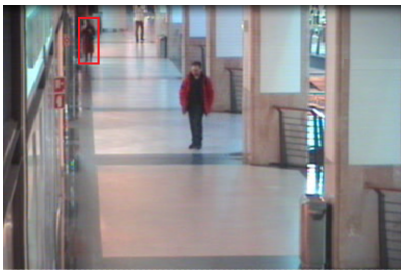
A informação de *ground truth* disponível continha a posição vertical e horizontal do centroide da *bounding box* referente a cada pessoa presente em cena em certas *frames*, bem como a velocidade instantânea nessas mesmas, que não foi usada. Como o *ground truth* disponível só continha informação das *bounding boxes* de 30 em 30 *frames*, foi necessário usar um método de estimação de posições nas *frames* em que essa informação não estava disponível, sendo escolhido o método de interpolação linear.

O tamanho das *bounding boxes*, representantes de cada objeto, não foi cedido. Ao analisar as diferenças de largura e altura das diferentes pessoas nas diferentes posições em cena, concluiu-se que as diferenças não eram significativas não se justificando uma calibração ou método semelhante para calcular as suas dimensões. Na Figura 3.11 podemos observar as dimensões das *bounding boxes* de duas pessoas diferentes: uma pessoa no local A e D (ver Figura 3.5), (primeira e segunda imagem respetivamente) e outra nos locais B e E (terceira e quarta imagem respetivamente).

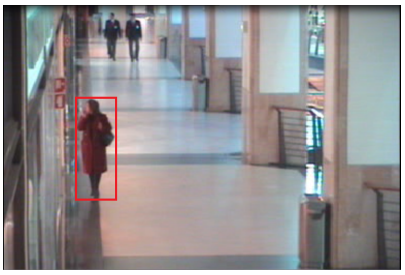
Figura 3.11: Comparação das dimensões das *bounding boxes* nas sequências *Friends Meet*: as primeiras duas dizem respeito a uma pessoa na posição A e D respetivamente, e as duas ultimas a outra nas posições B e E, respetivamente.

Não se justificando o cálculo das dimensões para o desenvolver da dissertação, considerou-se que todas as pessoas teriam exatamente o mesmo tamanho, independente da sua identidade e da sua posição em cena. Notando-se que esta consideração foi feita para as sequências usadas, podendo noutras sequências as dimensões das caixas variar significativamente, necessitando-se assim de as ter em conta. É o caso das sequências do CAVIAR, em que como podemos observar na Figura 3.12, as dimensões das *bounding boxes* relativas às pessoas variam devido à posição das pessoas em cena. Observando as figuras, nota-se uma clara diferença de dimensão da *bounding boxes*.

Na Figura 3.12a, onde a pessoa se encontra ao fundo do corredor, a *bounding box* é significativamente mais pequena do que na Figura 3.12b, em que a mesma pessoa já se encontra mais próxima da câmara de captura. Nos casos em que seja necessário fazer correções nas dimensões das *bounding boxes* poder-se-á optar por fazer uma calibração, ou de uma forma menos exigente, uma interpolação a fim de descobrir como variam as dimensões das pessoas em relação aos diversos pontos de cena.



(a) Pessoa com a respetiva *bounding box* (a vermelho), ao fundo do corredor



(b) Pessoa com a respetiva *bounding box* (a vermelho), a meio do corredor

Figura 3.12: Sequências CAVIAR [88]: A mesma pessoa em dois locais diferentes: na primeira imagem, a *bounding box* é significativamente mais pequena do que quando a pessoa se encontra mais próxima da câmara.

Outra informação de *ground truth* usada diz respeito aos grupos e tem como objetivo servir de referência aquando a avaliação do algoritmo de seguimento. Para a obtenção desta informação, foi necessário extrair do código de geração do *ground truth* a informação necessária. Ao contrário da informação da posição das pessoas nas diferentes *frames*, a informação do grupo apenas continha a sua constituição, isto é, eram dados os identificadores virtuais das pessoas que constituem cada grupo em cada *frame*. Havendo a necessidade de saber o comprimento, a largura e a posição de cada *bounding box* respeitante a cada grupo, foi aplicada a fórmula da Equação 3.16 onde os seus parâmetros são definidos na Equação 3.17 e $BoundingBox_{i,k}$ representa o parâmetro k da pessoa i.

$$BoundingBox_{Grupo} = \{x, y, largura, altura\} \tag{3.16}$$

$$\begin{cases} x = min(BoundingBox_{i,x}) \\ y = min(BoundingBox_{i,y}) \\ largura = max(BoundingBox_{i,x} + BoundingBox_{i,largura} - BoundingBox_{j,x}) \\ altura = max(BoundingBox_{i,y} + BoundingBox_{i,altura} - BoundingBox_{j,y}) \end{cases} \tag{3.17}$$

Para facilitar a avaliação do algoritmo, foi criado um ficheiro .csv com a compilação tratada da informação dos grupos e das pessoas em cena contendo o número da *frame* a que diz respeito a informação, o identificador da entidade (seja grupo ou pessoa), posição, largura e altura da *bounding box* e se é grupo ou não (se 1 é grupo, se 0 é pessoa), como se pode ver na tabela 3.2.

Número da Frame	Identificador da entidade	Posição x (centróide)	Posição y (centróide)	Largura	Altura	Grupo ?
627	1	599	459	50	90	0
627	4	597	411	113	103	1

Tabela 3.2: Exemplo da informação de grupos e pessoas compilada extraído do .csv relativo à sequência I

3.4 Adição de erro

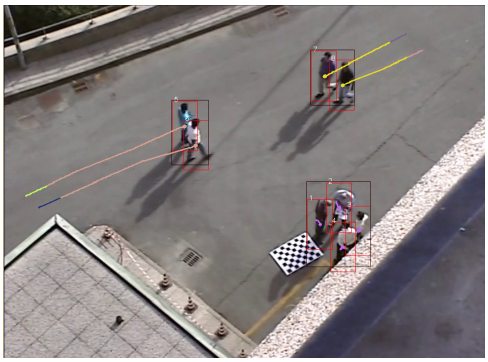
Tal como referido anteriormente, o algoritmo de gestão recebe como entrada informação espacial das pessoas presentes na *frame*, isto é, posição em x e em y e a altura e largura das *bounding boxes*. Esta informação simula o resultado de um algoritmo de seguimento de pessoas que em condições ideais não apresenta erros. Sabendo que dificilmente o resultado de um algoritmo é o ideal e pretendendo-se criar um algoritmo de gestão de grupos robusto a perturbações que geralmente afetam os algoritmos de seguimento, surge a necessidade de adicionar ruído na informação usada como entrada.

Segundo [54], há três erros típicos dos algoritmos de seguimento:

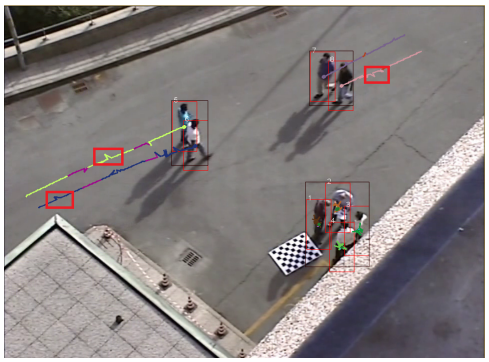
- Desvio na localização da entidade em relação ao *ground truth*
- Falsos positivos, isto é, o algoritmo identifica erradamente algo como sendo um alvo
- Falsos negativos, isto é, o algoritmo falha na deteção do alvo em algumas *frames*

Visto não ter sido encontrada informação sobre o desvio médio da localização do alvo em relação ao *ground truth* de nenhum algoritmo de seguimento, bem como a probabilidade de haver desvio na localização do alvo, foi necessário estipular probabilidades no sentido de conseguir reproduzir o mais fielmente possível estes erros. Assim sendo, fixou-se o valor do desvio do centroide das *bounding boxes* num máximo de 5 pixies para cada sentido do eixo (x, y) e com probabilidade p(*track*). Visualmente, a adição deste tipo de erro resultará no que se pode observar em 3.13. Se se comparar a linha que representa a trajetória das pessoas, conclui-se que na imagem da direita esta linha apresenta oscilações fruto da adição de um desvio nos pontos da trajetórias das pessoas em relação aos pontos de *ground truth*.

Por forma a simular os falsos negativos, o erro adicionado terá um probabilidade $p(FN_n)$ do tipo uniforme. Por vezes, o seguidor falha a deteção do alvo em mais do que uma $frame$ seguida. Com objetivo de simular o intervalo de falsos positivos, adicionou-se uma probabilidade de falha sabendo que falhou na anterior, $p(FN\ FN_{n-1})$, definido com o valor de 90%. Podemos observar o resultado da adição deste erro na Figura 3.14. Na imagem do lado direito, podemos observar que a linha que representa a trajetória da pessoa não é continua. O espaço onde a linha não se encontra desenhada mostra a ocorrência de falsos negativos no intervalo de tempo correspondente a esses pontos de trajetória.



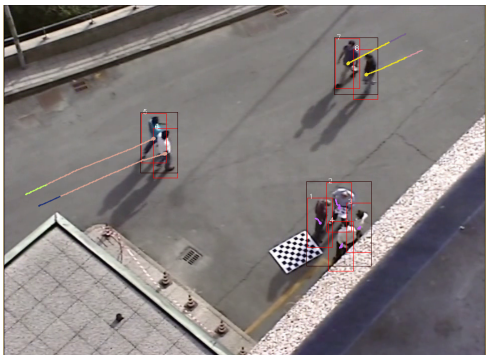
(a) *Frame* original



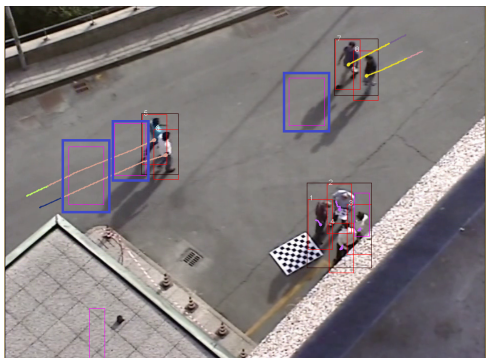
(b) *Frame* com desvios das trajetórias e re-dimensionamentos das *bounding boxes*

Figura 3.13: Erros: Desvio das trajetórias e re-dimensionamento das *bounding boxes*. O desvio das trajetórias encontra-se destacado pelos retângulos vermelhos.

Em relação aos falsos positivos, estes foram adicionados com probabilidade $p(FP)$, pelo que para cada pessoa na *frame*, aparecerá um falso positivo com probabilidade $p(FP)$. Visto que os falsos positivos dizem respeito a entidades falsas, quando um é gerado, este terá de ter um identificador, posição x e y do seu centroide e as dimensões da *bounding boxes*. Cada uma destas informações é gerada aleatoriamente respeitando algumas regras. A posição do centroide terá de estar respeitar a resolução dos vídeos. Visto que esta é de [960 X 720] pixeis, a posição x do centroide terá um valor entre 20 e 900 e a posição y entre 20 e 680 pixeis. A largura e a altura das *bounding boxes* têm valores entre 25, 75 e 75, 125 respetivamente. A escolha destes valores deve-se ao facto de as *bounding boxes* das pessoas terem dimensão de [50x100] pixeis e de os valores gerados não poderem discrepar muito destes. Podemos observar o resultado visual da adição deste tipo de erro na Figura 3.15 em vários locais da imagem, *bounding boxes* que não correspondem a nenhuma entidade presente na *frame*.



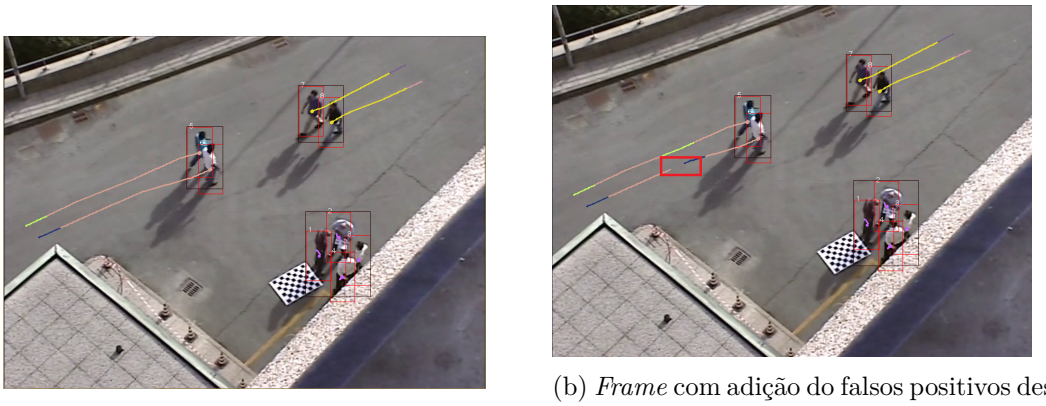
(a) *Frame* original



(b) *Frame* com adição de falsos positivos destacados com retângulos azuis

Figura 3.15: Falsos positivos destacados com retangulos azuis

Por forma a avaliar a robustez do algoritmo a erros, serão usados diferentes tipos de



(a) *Frame* original

(b) *Frame* com adição do falsos positivos destacados com um retângulo vermelho

Figura 3.14: Falsos Negativos destacados com retângulo vermelho

erros, definidos Tabela 3.3, onde todas as probabilidades de erro foram sujeitas a uma validação visual.

	P(track) %	P(FN) %	P(FP) %
Tipo 0	0	0	0
Tipo I	2.5	2.5	1.125
Tipo II	5	5	2.5
Tipo III	10	10	5

Tabela 3.3: Tabela de probabilidades de erro em função dos níveis de ruído.

Capítulo 4

Algoritmo de Gestão de Grupos

O presente capítulo tem como objetivo abordar um algoritmo de gestão de grupos. Na primeira secção são definidos os conceitos necessários para perceber o algoritmo de gestão. Na segunda secção é dada uma explicação do algoritmo proposto em [72], incluindo as suas fases, e alguns pormenores de implementação. Por fim, na última secção são apresentadas propostas de evolução do algoritmo base por forma a aumentar o seu desempenho.

4.1 Conceitos

4.1.1 Definição de grupo

Socialmente, um grupo corresponde a um conjunto de pessoas que se encontram espacialmente próximas e que interagem umas com as outras e que partilham um objetivo em comum [1]. Na prática, os conceitos de proximidade espacial e interação, para além de difíceis de transpor para a prática, são subjetivos e insuficientes para uma adequada deteção de grupos. Não sendo a coerência espacial um critério para a definição de grupo por si só suficiente, é necessário acrescentar outros como tamanho, duração, velocidade e estrutura.

Sendo um grupo um conjunto de duas ou mais pessoas, podemos concluir que o seu tamanho é sempre igual ou superior ao tamanho de uma pessoa [72]. O tamanho do grupo poderá ser igual ao tamanho de uma pessoa nos casos onde haja ocultação total de uma pessoa por parte de outra. Embora a pessoa não esteja visível, poderá cumprir os restantes requisitos, sendo incluída no grupo da pessoa que a oculta. A velocidade e a direção das pessoas também são critérios a ter em conta aquando da deteção de grupos [72] [73]. Enquanto parte de um grupo, as pessoas que os compõem terão todas velocidades e direções idênticas. A adição deste critério evita considerar-se elemento do grupo, por exemplo, uma pessoa que se desloca em sentido contrário a este, ou então, uma pessoa que passa relativamente perto do grupo mas mais rápido do que este.

Por fim, o grupo terá de ser estruturalmente coerente ao longo do tempo, isto é, o grupo terá de ser composto pelas mesmas pessoas para manter a mesma identidade. Assim, a ocorrência de eventos como a separação ou junção de elementos irá gerar novos grupos.

4.1.2 *Group Incoherence*: medida de coerência do grupo

O *Group (In)Coherence* [72] (GI) é um critério que permite inferir quanto à possibilidade de um certo número de pessoas poder ou não compor um grupo: quanto maior for o seu valor, menor é a possibilidade de o conjunto de pessoas ser um grupo. Este é definido através da média das distancias, desvio de velocidade e desvio de direção. Dando diferentes pesos a cada uma destas parcelas, por elas mostrarem diferentes importâncias aquando a deteção do grupo, tem-se que Group(In)coherence é definido pela Equação 4.1:

$$Group(In)Coherence = w_1 \cdot distanceAvg + w_2 \cdot speedStdDev + w_3 \cdot directionStdDev \tag{4.1}$$

Por questão de lógica e coerência, $group(In)Coherence$ será referido como $groupIncoherence$ (GI), visto tratar-se de uma medida que aumenta com a incoerência do grupo.

A média das distâncias das diferentes entidades, *distanceAvg*, é calculada através da Equação 4.2, sendo $nFrames$ o número de *frames* total em que as entidades parte do grupo estão presentes, $nParess_k$ o número de pares de entidades na *frame* k, n_k o número de pessoas na *frame* k, com $i, j \in [1, n_k]$ a corresponderem às entidades na *frame* k e $dist(i,j)$ a distância entre os centroides das *bounding boxes* das entidades i e j.

$$distanceAvg = \frac{1}{nFrames} \sum_{k=1}^{nFrames} \frac{1}{nParess_k} \sum_{i=1}^{n_k} \sum_{j=i+1}^{n_k} dist(i,j) \tag{4.2}$$

A média do desvio das velocidades, *speedStdDev*, é calculada respeitando a Equação 4.3, onde $speedSD_k$ é definido de acordo com a Equação 4.4, sendo s_i a média do desvio das velocidades instantâneas da entidade i e m_k a média da velocidade das entidades em teste na *frame* k.

$$speedStdDev = \frac{1}{nFrames} \sum_{k=1}^{nFrames} speedSD_k \tag{4.3}$$

$$speedSD_k = \sqrt{\frac{1}{n_k} \sum_{i=1}^{n_k} (s_i - m_k)^2} \tag{4.4}$$

A média do desvio da direção das entidades, *directionStdDev*, é calculada através da Equação 4.5, onde m_i é o vetor de movimento da entidade i e d_k é a média dos desvios do

vetor direção da entidade na k.

$$directionStdDev = \frac{1}{nFrames} \sum_{k=1}^{nFrames} \sqrt{\frac{1}{n_k} \sum_{i=1}^{n_k} \frac{1}{2}((m_{i_x} - d_{k_x})^2 + (m_{i_y} - d_{k_y})^2)} \tag{4.5}$$

É de notar que todas estas medidas são calculadas tendo em consideração uma janela temporal, T, isto é, a decisão tomada numa dada $frame$ (t_c) tem em conta a informação guardada ao longo das frames correspondentes à janela temporal. Assim, se se está a processar e a decidir em (t_c), está-se a ter em conta as $frames$ de $(t_c$-T) a (t_c), como se pode ver na Figura 4.1, respeitando que $T = 20$ $frames$, tal como proposto em [72].

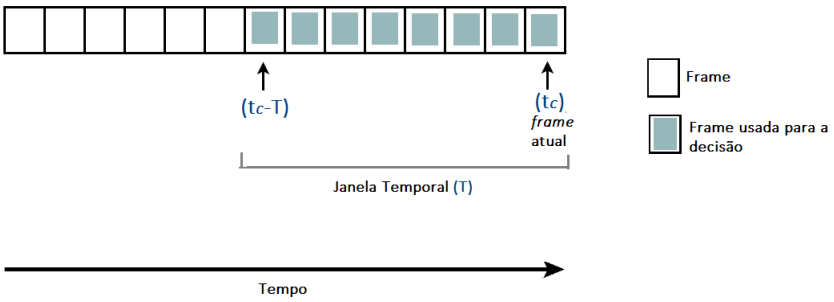


Figura 4.1: Relação $frame$ processada e janela temporal: ao processar e decidir na $frame$ atual (t_c), essa decisão tem em conta todas as $frames$ desde $t_c - T$ a T.

Embora o artigo refira a necessidade de normalização dos pesos $w1$, $w2$, $w3$, não é explicado de que forma o faz, tendo-se estipulado que $w_1 + w_2 + w_3 = 1$.

Os parâmetros $distanceAvg$, $speedStdDev$, $directionStdDev$ também foram submetidos a uma normalização (Equações 4.6, 4.7, 4.7, respetivamente), dividindo o seu valor pelo seu máximo encontrado em todas sequências.

$$distanceAvg_{Norm} = \frac{distanceAvg}{max(distanceAvg)}\Big|_{max=900pixeis} \tag{4.6}$$

$$speedStdDev_{Norm} = \frac{speedStdDev}{max(speedStdDev)}\Big|_{max=2pixeis} \tag{4.7}$$

$$directionStdDev_{Norm} = \frac{directionStdDev}{max(directionStdDev)}\Big|_{max=2.30pixeis} \tag{4.8}$$

As normalizações feitas têm como objetivo facilitar a comparação dos diferentes parâmetros para assim se poder alterar facilmente os seus pesos e poder adicionar novos parâmetros sem serem necessários cálculos adicionais.

4.2 Algoritmo base

O algoritmo implementado tem por base o trabalho de [72], tendo como objetivo a deteção e a gestão de grupos. Na Figura 4.2, é mostrado um diagrama de blocos que representa, de forma simplificada, o algoritmo implementado. Como entrada, recebe um conjunto de pessoas que simula o resultado de um algoritmo de seguimento de pessoas. Ao receber as entidades na *frame* atual, as pessoas seguem para o módulo de tratamento de erros. Este módulo tem como objetivo detetar e tratar erros por forma a não prejudicar a performance do algoritmo. Seguidamente, as novas entidades seguem para o módulo de criação de entidades e as restantes para o de atualização de entidades. Do módulo de atualização de pessoas resultam as pessoas com a informação de interesse atualizada. Esta informação será a entrada do módulo separação.

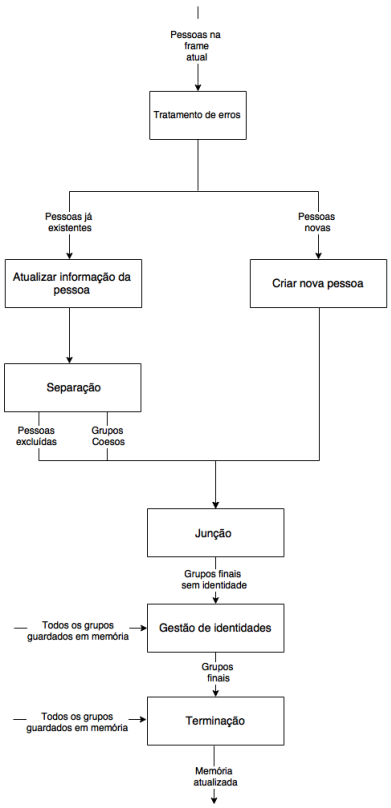


Figura 4.2: Algoritmo em diagrama de blocos

O módulo separação tem como objetivo detetar a ocorrência de eventos de separação nos grupos já existentes (tendo em conta a informação recém chegada). Note-se que as entidades recém formadas não são consideradas entrada para este módulo. Isto deve-se ao facto de que uma entidade que se forma numa dada *frame* nunca poderá sofrer um evento de desintegração. Deste módulo resultam os grupos sem as pessoas já excluídas e as entidades que se separaram dos grupos, que seguirão para o módulo seguinte, módulo junção.

O módulo junção tem como função detetar e proceder à junção de entidades, recebendo para tal as entidades novas, os grupos que não sofreram separação e as entidades que se separaram de outras. Visto que neste módulo não é feita a gestão de identidades dos grupos, o módulo de gestão de identidades irá atribuir um identificador a cada grupo de acordo com a informação existente, isto é, se houver registo de um grupo com os mesmo constituintes, é-lhe dado o seu identificador, se não, o novo grupo terá um identificador diferente de todos os existentes, para assim o poder identificar univocamente.

Por fim, temos o módulo de terminação, que recebendo como entrada os grupos atuais e os grupos guardados em memória, elimina os grupos que já não se encontrem em cena há um tempo superior a $t_{memória}$, definido como 5 vezes a janela temporal ($5 \times T$).

4.2.1 Tratamento de erros

Este módulo tem como objetivo fazer o tratamento de dois tipos de erros frequentes como consequência de algoritmos de seguimento: Falsos Positivos (FP) e Falsos Negativos (FN).

O primeiro tipo de erro dá-se quando o algoritmo julga ter encontrado um alvo de seguimento, neste caso pessoas, enquanto na verdade não o fez. Assim, para evitar processá-los como se de entidades se tratassem, é necessário proceder à sua deteção e posterior eliminação. Considera-se que uma pessoa é potencialmente um falso positivo se esta esteve ausente em número superior a $\frac{T}{5}$ *frames* entre $t_c - T$ e t_c. Estes casos não são avaliados relativamente à sua inclusão num grupo, mas a sua informação é armazenada. Se a entrada da informação desta pessoa continuar a surgir de forma a que haja informação válida em pelo menos $\frac{T}{5}$ *frames* entre $t_c - T$ e t_c, a pessoa será tida como real e proceder-se-á ao seu normal processamento. Em termos práticos, imagine-se que existe informação da pessoa P_1 entre $t_c - T$ e t_c em menos de $\frac{T}{5}$ *frames*. Inicialmente, o algoritmo considera a entidade como um potencial FP não procedendo ao seu processamento. Se se começar a receber informação desse alvo de tal forma que a informação tida entre $t_c - T$ e t_c seja superior a $\frac{T}{5}$, essa pessoa é classificada como entidade real. Se pelo contrário, a informação dessa continuar escassa, o alvo acabará por ser eliminado naturalmente no processo de terminação.

Este módulo também lida com Falsos Negativos. Este erro resulta numa fragmentação da informação da trajetória das pessoas e é corrigido procedendo-se à estimação das posições das pessoas num processo de previsão linear do movimento, ou seja, considera-se que a pessoa mantém o movimento linear das últimas $frames$. Tal é feito quando a pessoa não foi considerada Falso Positivo na *frame* atual e não foi eliminada no processo de terminação. Apenas se procede à estimação das posições, se houver informação real (não estimada) de $t_c - \frac{5T}{4}$ a t_c. Isto evita que se esteja a estimar a trajetória de uma pessoa, quando esta já saiu de cena.

Se se receber informação de uma pessoa, outrora considerada falso positivo, havendo entre $t_c - T$ e t_c informação em pelo menos $\frac{T}{5}$ *frames* é feita uma interpolação entre $t_c - T$ e t_c das posições, para assim se conseguir obter uma estimação da trajetória.

4.2.2 Criar e atualizar a entidade pessoa

O módulo *criar entidade* tem como objetivo criar a entidade *pessoa* (Ep_i) para cada pessoa que apresente um identificador diferente daqueles que já estão armazenados. Cada pessoa é então representada de acordo com a Equação 4.9, onde Id_i é um número inteiro que tem como objetivo identificar inequivocamente uma entidade. Duas entidades diferentes têm identificadores diferentes, e a mesma entidade terá sempre o mesmo identificador, salvo erros de gestão de identidades por parte dos algoritmos de seguimento que fornecem as entidades à camada de gestão de grupos; Bb_i é o conjunto de caixas delimitadoras, *Bounding Boxes* da pessoa ao longo das *frames*; Gp_i é o conjunto de identificadores de grupos a que a pessoa vai pertencendo ao longos das *frames*; Fr_i é o conjunto de *frames* onde há informação sobre a pessoas P_i.

$$P_i = \{Id_i, Bb_i, Gp_i, Fr_i\} \tag{4.9}$$

Na fase de atualização de entidades são consideradas todas as pessoas presentes na *frame* atual cuja presença já foi detetada anteriormente, isto é, pessoas que já estiveram presentes em frames anteriores. Esta fase destina-se à atualização do *vetor de frames* e *vetor de bounding boxes* com a informação recebida na *frame* atual.

4.2.3 Separação

Este módulo recebe como entrada todas as entidades que se encontram na *frame*. Depois de atualizadas as entidades, os grupos que estavam presentes na *frame* anterior são avaliados de forma a averiguar se houve alguma separação. A separação dá-se quando uma entidade se tem vindo a afastar do grupo a que pertencia ao longo de um dado número de *frames*. Sabe-se que este evento ocorreu quando a dinâmica do grupo se alterou de tal forma que o GI deste se torna superior a um dado *threshold* (cujo valor é definido futuramente). Note-se que o GI avalia o quão incoerente é o grupo, logo se alguém se tem vindo a afastar dele, esse valor tem vindo a subir. Quando isto acontece é necessário perceber quais as entidades que dele se separaram. A Figura 4.3 corresponde a um diagrama de blocos referente ao algoritmo de deteção de separação. Como se pode perceber pelo diagrama, para cada grupo, vão sendo extraídas as pessoas que estariam a aumentar o GI do grupo até se obter um valor de GI inferior a de um *threshold*. Este *threshold* é o limite que define se o conjunto de pessoas é ou não grupo. Assim, à medida que se vão extraindo as pessoas que mais sobem o GI, vai-se obtendo um grupo mais coeso. Quando, ao se avaliar o grupo, já se obtém GI com um valor inferior ao do *threhsold*, sabe-se que

esse grupo é coeso. As pessoas extraídas deixam de pertencer ao grupo e, juntamente com este, prosseguem para o módulo de junção.

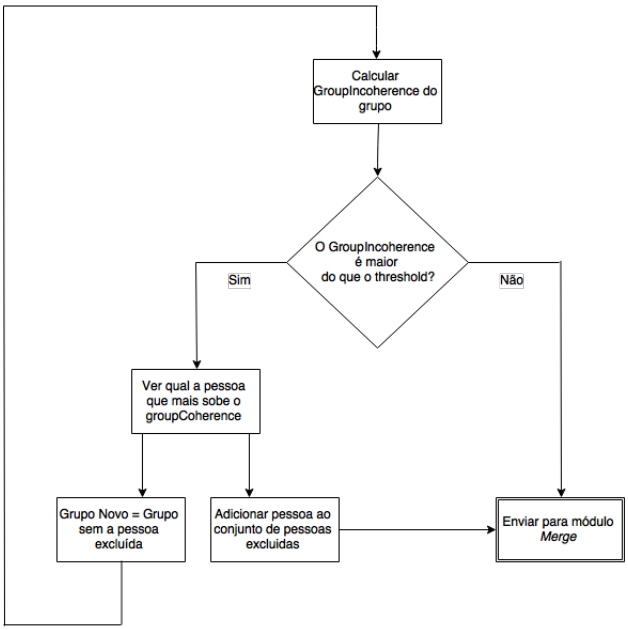


Figura 4.3: Diagrama de blocos representativo do módulo de separação

4.2.4 Junção

Este módulo tem como objetivo avaliar se ocorreram eventos de junção entre os grupos e pessoas resultantes do módulo de separação e as pessoas novas. O diagrama de blocos que representa o método de junção encontra-se na Figura 4.4. O método de junção de entidades assenta na premissa de que duas entidades juntas com um GI inferior ao *threshold* pertencem ao mesmo grupo. Assim, são encontrados os pares de entidades cujo seu GI seja o mais baixo de todas os pares possíveis. Se o seu valor for inferior a um dado *threshold*, essas duas entidades são agregadas numa só. O processo de junção de entidades termina quando nenhum par de entidades têm um GI mais baixo que o valor de *threshold*.

4.2.5 Gestão de Identidades e Terminação

O módulo de gestão de identidades tem como objetivo atribuir identidades aos grupos detetados em *frame*. Assim, a primeira fase é examinar todos os grupos guardados em memória. A identidade do grupo nessa *frame* será atribuída de acordo com as identidades já existentes. Se em memória existir um grupo que seja composto pelos mesmos elementos que o novo grupo, ser-lhe-á atribuído o mesmo identificador. Se pelo contrário, o grupo não tiver aparecido até então, é-lhe atribuído um identificador diferente de todos os outros. Quando há junção de dois grupos, em [72] é atribuído o identificador do grupo mais antigo ao grupo recém formado. Esta definição não foi mantida pois a definição de grupo

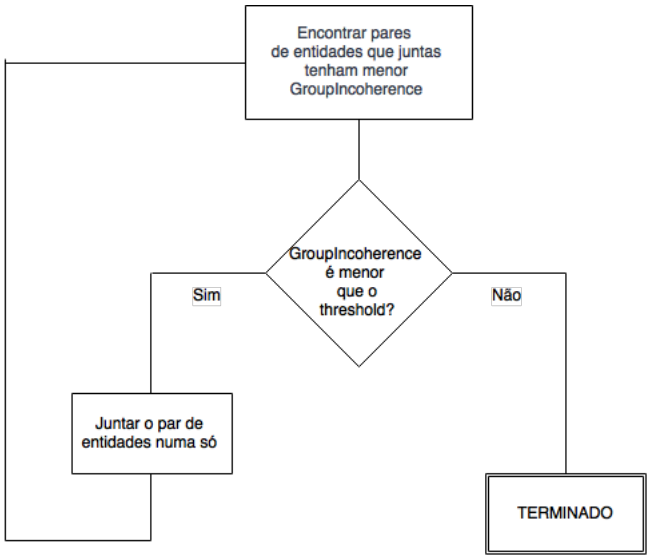


Figura 4.4: Diagrama de blocos representativo do módulo de junção

usada indica que dois grupos só serão iguais se e só se forem constituídos pelos mesmos elementos. A título de exemplo, dizer-se que o grupo 4 se juntou ao grupo 5 e que este conjunto manteve o identificador 4 não faz sentido na presente definição. Desta forma, se já não tiver aparecido um grupo com os elementos do grupo 4 e do grupo 5 como parte de um outro grupo, quererá dizer que se formou um novo grupo ao qual é atribuída uma nova identificação. Esta definição não afeta a avaliação do desempenho pois as métricas usadas não usam a correspondência de elementos do grupo para que se considere este como correto.

A fase de terminação é a fase onde os grupos e as pessoas mais antigos são eliminadas da memória. Respeitando o mencionado no artigo, são eliminados todos os grupos que não foram detetados desde $t_c - 5T$. Como o artigo não menciona eliminação das pessoas, de forma idêntica aos grupos, estas são retiradas de memória quando não há informação real (não estimada) destas desde $t_c - 5T$.

4.3 Melhorias e Resultados

Com a presente secção pretende-se propor algumas melhorias ao algoritmo proposto em [72] e abordado na Secção 4.2. Com estas melhorias pretende-se atingir um aumento do desempenho do algoritmo em termos gerais (em todas as sequências) ou em casos específicos quando assim se justificar.

Em cada subsecção será feita uma comparação do desempenho do algoritmo com e sem alteração, por forma a conseguir-se avaliar o progresso obtido. Será feita, sempre que necessário, uma análise das sequências onde a alteração trouxe melhor proveito e onde prejudicou o desempenho.

As avaliações serão feitas para as sequências descritas na Secção 3.1.1 e as métricas usadas serão as descritas ea Secção 3.2.1.

Cada subsecção, à exceção da primeira que apresenta os resultados e a discussão do algoritmo original, abordará uma proposta de melhoria. De uma subsecção para a seguinte as melhorias são mantidas, sendo portanto um processo incremental.

No fim da secção, é feito um resumo dos resultados para avaliar de forma geral as melhorias provocadas pelas alterações propostas.

4.3.1 Algoritmo com Pesos Originais

O algoritmo descrito na Secção 4.2, proposto em [72], tem como objetivo a deteção de grupos usando para tal uma medida de decisão intitulada pelos autores de *GroupIncoherence* (GI). Esta medida é construída através de uma média aritmética ponderada entre a distância média do conjunto de pessoas, desvio de velocidade e desvio de direção do mesmo sendo w_1, w_2 e w_3 os seus pesos respetivos. O facto de terem optado por uma média aritmética ponderada em detrimento de média aritmética simples, revela a necessidade de atribuir importâncias relativas diferentes, sendo que o maior valor de w diz respeito ao parâmetro com maior relevância. Os autores propõem $w_1 = 0.7$, $w_2 = 0.15$, $w_3 = 0.15$, mostrando assim que a distância média entre pessoas é a medida mais importante e que o desvio das velocidades e das direções têm igual relevância. Inicialmente testou-se o algoritmo com valores de *threshold* entre 0 e 1, com incrementos de 0.1. Avaliado o desempenho para cada um desses valores, observou-se que havia uma degradação do comportamento do algoritmo com valores de *threshold* superiores a 0.3. Assim procurou-se um novo valor de *threshold* iterativamente, entre 0 e 0.3 com incrementos de 0.005 obtendo-se o valor de $threshold = 0.125$, que corresponde ao melhor desempenho para os pesos propostos.

Tal como se pode verificar na Figura 4.5, antes da adição de erros o algoritmo apresenta resultados de precisão, *recall* e GDSR superiores a 90%. Relativamente à fragmentação de trajetória, esta procede a uma fragmentação média de 0.45 vezes por pessoa. Isto deve-se ao facto de os valores de GI variarem em torno do valor de *threshold* e assim o algoritmo decidir separar e juntar conjuntos de pessoas em grupos ao longo do tempo, fragmentando a trajetória dos grupos.

Tais valores de desempenho do algoritmo indicam que a formulação do algoritmo consegue proceder à deteção de grupos numa situação sem erros. No entanto, ao adicionar erros de Tipo II (detalhados na Secção 3.4), verifica-se uma perda significativa de desempenho, obtendo valores de precisão de aproximadamente 70%. O valor de *recall* é alto (aproximadamente 80%), o que significa que embora a maioria dos grupos existentes (de *ground truth*) sejam detetados, são criadas entidades adicionais. Observa-se também um aumento das fragmentações por entidade de 0.45 para 2.23 visto que os erros fazem variar o GI em torno do *threshold* fazendo o algoritmo separar e juntar entidades incorretamente, fragmentando as suas trajetórias.

Tendo em atenção as alterações dos valores de precisão por cada sequência, na Figura 4.6, verifica-se que as sequências 10 e 11 apresentam os resultados mais baixos. Estas sequências correspondem a casos onde se verifica a existência de filas de pessoas e onde a adição de erros, mais precisamente na posição das *bounding boxes* das pessoas, leva a uma maior desestabilização do GI. Note-se que este tipo de erros faz alterar os três parâmetros usados para calcular o GI (distancia, velocidade e direção) o que pode provocar a rutura do grupo.

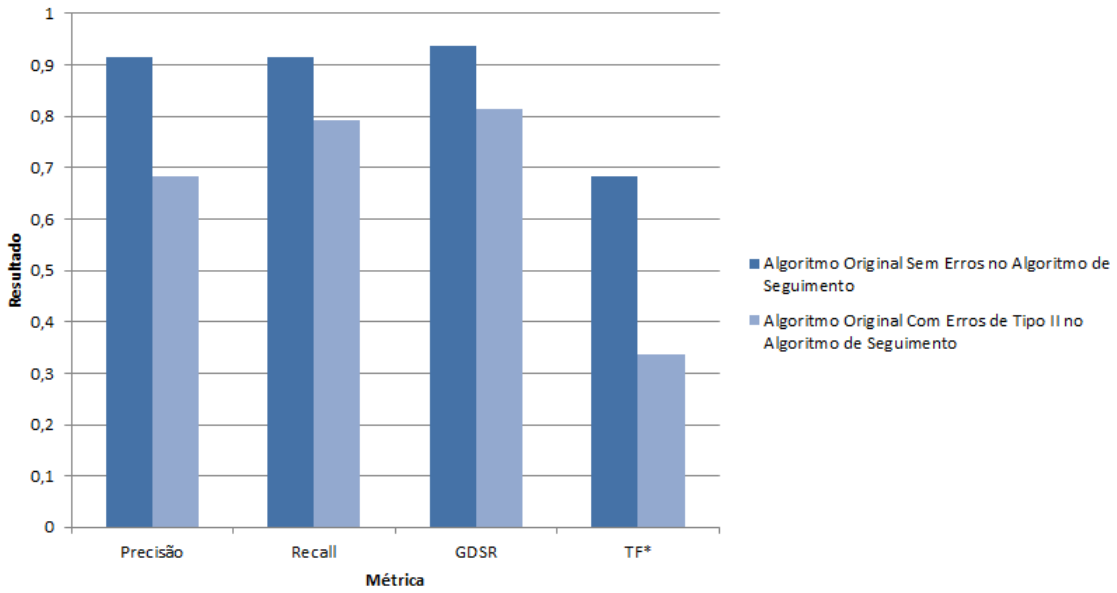


Figura 4.5: Comparação do desempenho do algoritmo original sem e com erros na entrada usando as métricas Precisão, *Recall*, GDSR, TF*

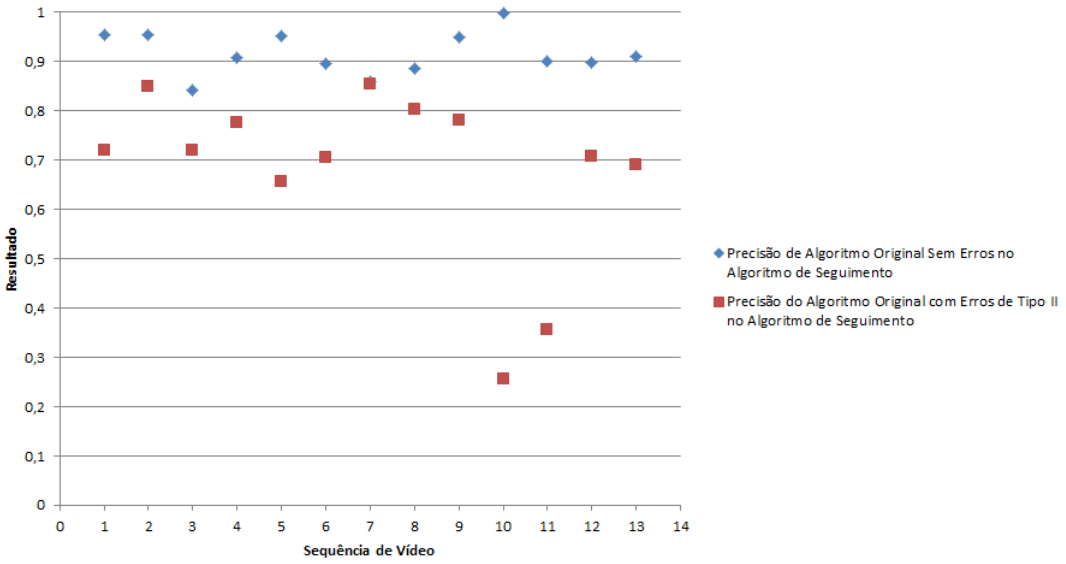


Figura 4.6: Comparação da precisão do algoritmo original, para cada sequência, com e sem erros na entrada

Devido à diminuição do desempenho do algoritmo quando a sua entrada é sujeita a erros, as melhorias propostas de seguida serão sempre avaliadas para entradas não livres de erros, mais precisamente sob erros de Tipo II.

4.3.2 Determinação de Novos Pesos e *threshold*

Por forma a averiguar se um conjunto de pesos e *threshold* diferentes melhorava a performance, procurou-se de forma iterativa um novo conjunto destes valores. Para tal, os parâmetros w_1, w_2, w_3 e $threshold$ foram testados de 0 a 1 em incrementos de 0.01, verificando-se qual a combinação de valores que levou a uma maximização das métricas para as sequências de teste. A melhor combinação de parâmetros é aquela apresentada na Equação 4.10, o que significa aumentar o peso da média das distância em 5% e transferir 5% do peso do desvio das direções para o desvio das velocidades.

$$\begin{cases} w_1 = 0.75 \\ w_2 = 0.15 \\ w_3 = 0.10 \\ threshold = 0.14 \end{cases} \tag{4.10}$$

Este novo conjunto de pesos e *threshold* permite um aumento de 2% da precisão, aumentando também o *recall* (1%), GDSR (1%) e melhorando a fragmentação de trajetória (de 2.23 fragmentações por entidade para 1.94), tal como verificado na Figura 4.7. Analisando a Figura 4.8, verifica-se que as melhorias ocorrem em 8 das 13 sequências (piorando apenas cerca de 1% em 3 sequências). Tal significa que a nova formulação é mais indicada

para estas sequências do que a original e por conseguinte permite gerir de forma mais eficiente os grupos. A sequência onde se observa a maior melhoria com a implementação destes novos pesos e *threshold* é a sequência 5, cuja melhoria é aproximadamente de 10%. Na Figura 4.9 é feita uma ilustração do desempenho do algoritmo com os pesos e *threshold* originais, do algoritmo com os valores alterados bem como do desempenho ideal (dado pelo *groundtruth*). Nos quadros superiores podemos observar a deteção do grupo respeitando o *groundtruth*. Como se pode observar, no segundo quadro, a pessoa do canto superior esquerdo começa a deslocar-se por forma a poder abandonar o grupo. Segundo a definição de grupo exposta na Secção 4.1.1, o grupo é um conjunto de pessoas com um objetivo comum. Visto que a pessoa iniciou a sua rota por forma a abandoná-lo, o seu objetivo tornou-se diferente dos restantes elementos, resultando num aumento da distancia média entre as pessoas, desvio da velocidade e direção e consequentemente do GI. Assim sendo, o algoritmo deveria detetar que o grupo foi reduzido a três elementos e o GI deveria tomar um valor superior ao *threshold* por forma a ser detetada a respetiva incoerência no grupo. Com os pesos e *threshold* originais, a deteção da incoerência do grupo é feita mais tardiamente, tal como se pode observar na segunda fila de quadros da Figura 4.9. Com a alteração dos pesos e do *threshold*, consegue-se uma melhoria, como pode ser observado na fila inferior da mesma imagem. A maior rapidez na deteção da alteração do grupo quando comparado com o algoritmo com pesos e *threshold* originais, originou, neste caso, um aumento de precisão de 10%.

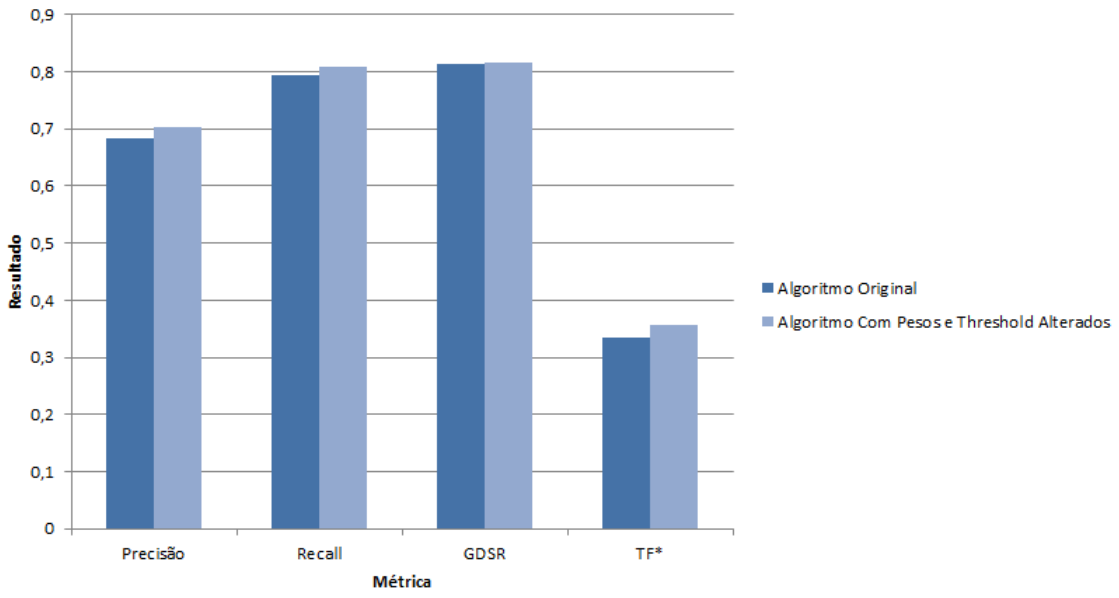


Figura 4.7: Comparação algoritmo com pesos originais e com pesos e *threshold* melhorado

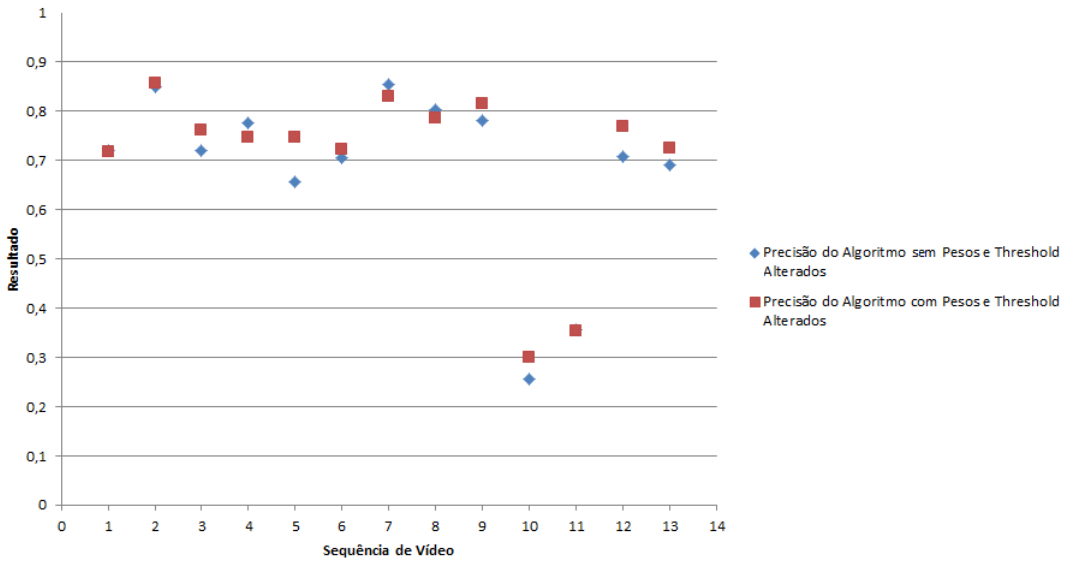


Figura 4.8: Comparação da precisão do algoritmo com pesos originais e com pesos e *threshold* melhorado nas diversas sequências

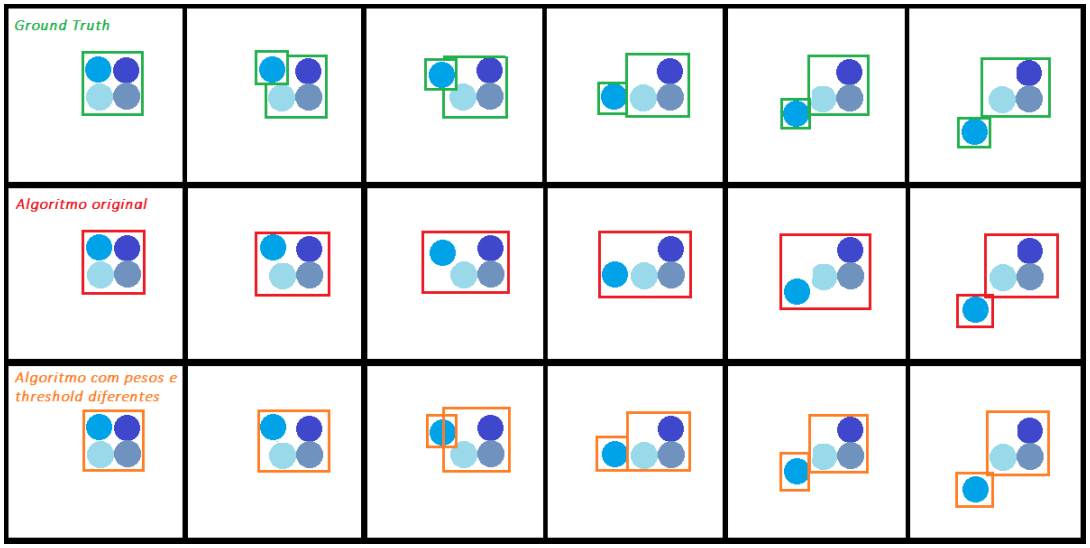


Figura 4.9: Exemplo da deteção de um grupo: na fila de quadros superiores tem-se a deteção do grupo segundo o *ground truth*, na fila do meio tem-se a deteção com o algoritmo com pesos originais e na última fila tem-se a deteção do grupo usando o algoritmo com pesos e *thresold* melhorados

4.3.3 *Threshold* em Histerese

Como já referido anteriormente, o algoritmo original revela uma diminuição de prestação quando na entrada há ocorrência de erros. Após observação do desempenho do algoritmo e dos valores de GI, constatou-se que o uso de apenas um *threshold* como decisor torna a resposta do algoritmo mais sensível a erros na entrada.

É portanto proposto um *threshold* em histerese, isto é, o uso de dois limites em vez de um, conforme indicado na Equação 4.11. Estes valores foram obtidos variando o th_{min} e th_{max} entre 0 e 0.3 (visto já ter sido concluído anteriormente que quando o *threshold* é superior a 0.3, há uma diminuição de qualidade do algoritmo) em incrementos de 0.01 de forma a maximizar a precisão nas sequências de teste.

$$\text{conjunto de pessoas} = \begin{cases} \text{grupo coerente} & GI < th_{min} \\ \text{grupo não coerente} & GI > th_{max} \\ \text{mantém caso da } \textit{frame} \text{ anterior} & \text{restantes casos} \end{cases} \tag{4.11}$$

Tal como pode ser concluído pela interpretação da Equação 4.11, quando o GI de um conjunto de pessoas estiver abaixo de um *threshold* mínimo, th_{min}, é avaliado como grupo coerente, se estiver acima do *threshold* máximo, th_{max} o conjunto de pessoas não é considerado grupo ou se o valor se encontrar entre os dois limiares, o algoritmo decide manter o estado anterior, isto é, o algoritmo replica a decisão da *frame* anterior: se eram grupo, assim continuam, se não, continuam sem o ser.

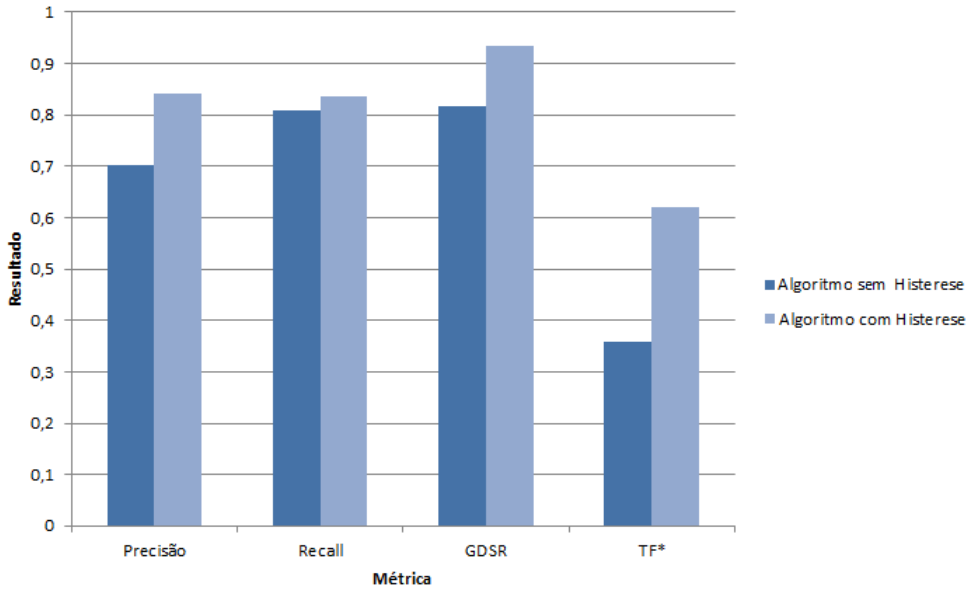


Figura 4.10: Comparação do desempenho geral do algoritmo com e sem histerese em termos de precisão, *recall*, GDSR e TF^*

Como podemos ver na Figura 4.10, a substituição do *threshold* único por um em histerese histerese resultou numa melhoria significativa da precisão (14%), de *recall* (3%), GDSR (12%) e uma melhoria do resultado da fragmentação de 1.94 fragmentações por trajetoria para 0.64. Tal deve-se à capacidade que o algoritmo adquire em manter a formação de grupos quando sujeito a erros. Neste ponto verifica-se já que a precisão se

encontra a cerca de 84%, o *recall* a 83% e o GDSR perto de 93%, o que significa que o algoritmo, mesmo em situações com erro, já é capaz de detetar corretamente quase todos os grupos e entidades, assim como ter uma taxa de erros baixa (precisão) relativamente à criação de entidades não existentes no *ground truth*.

Avaliando a Figura 4.11, vemos que com a substituição do *threshold* único pela histerese, houve um melhoramento na precisão do algoritmo em quase todas as sequências. Como exceção temos a sequência 13, que contém um maior número de pessoas quando comparado com as restantes sequências e um número substancialmente maior de separações e junções de pessoas e grupos, que ocorrem aqui de forma mais lenta. O afastamento do limiar na histerese torna o algoritmo mais lento na deteção deste tipo de eventos (note-se que o algoritmo só decide juntar e separar pessoas quando o GI se encontra abaixo do th_{min} ou acima de th_{max}). Também na sequência 7 se verifica um decréscimo da precisão havendo neste caso dois grupos de duas pessoas que se cruzam e, apesar de ocorrer um aumento do desvio da direção, como esta surge agora com um peso reduzido, não é suficiente alta para evitar a junção do grupo (neste caso a distância média entre elementos é muito baixa, reduzindo muito esse valor e levando à ocorrência da junção das pessoas em grupo). Devido ao uso da histerese, como o *threshold* de separação, th_{min} é superior, será necessário algum afastamento (e, obrigatoriamente, mais tempo) para que as pessoas deixem de ser consideradas grupo. Devido à curta duração desta sequência, o algoritmo não se consegue adaptar rápido o suficiente para evitar a diminuição da precisão.

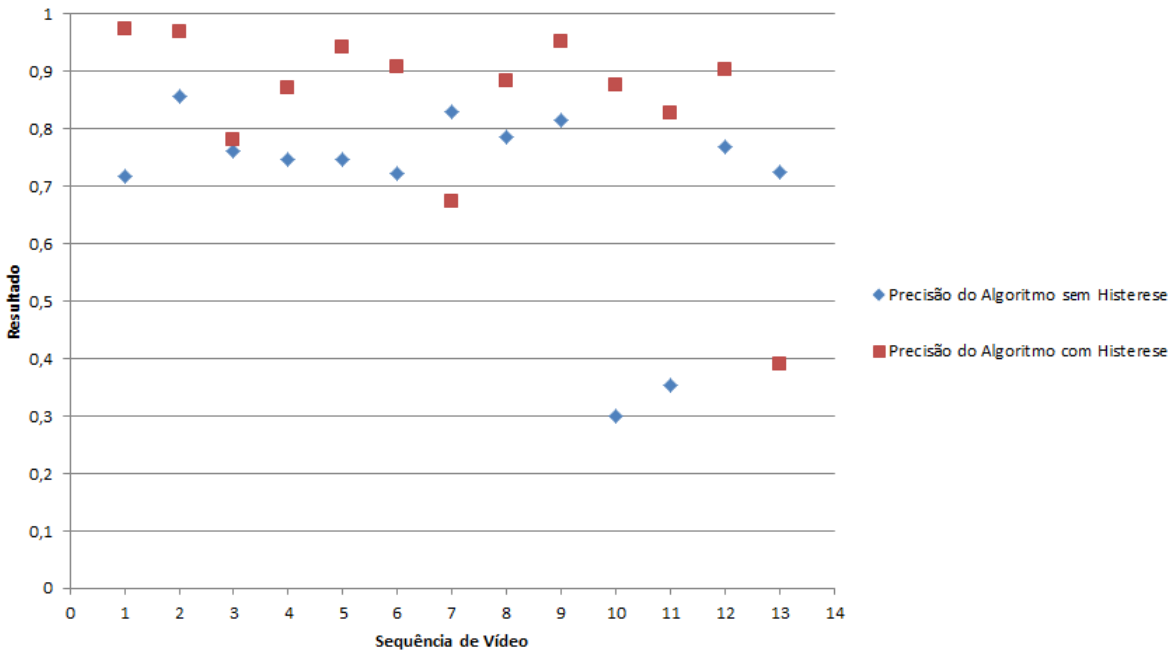


Figura 4.11: Comparação da precisão do algoritmo com e sem histerese nas diferentes sequências

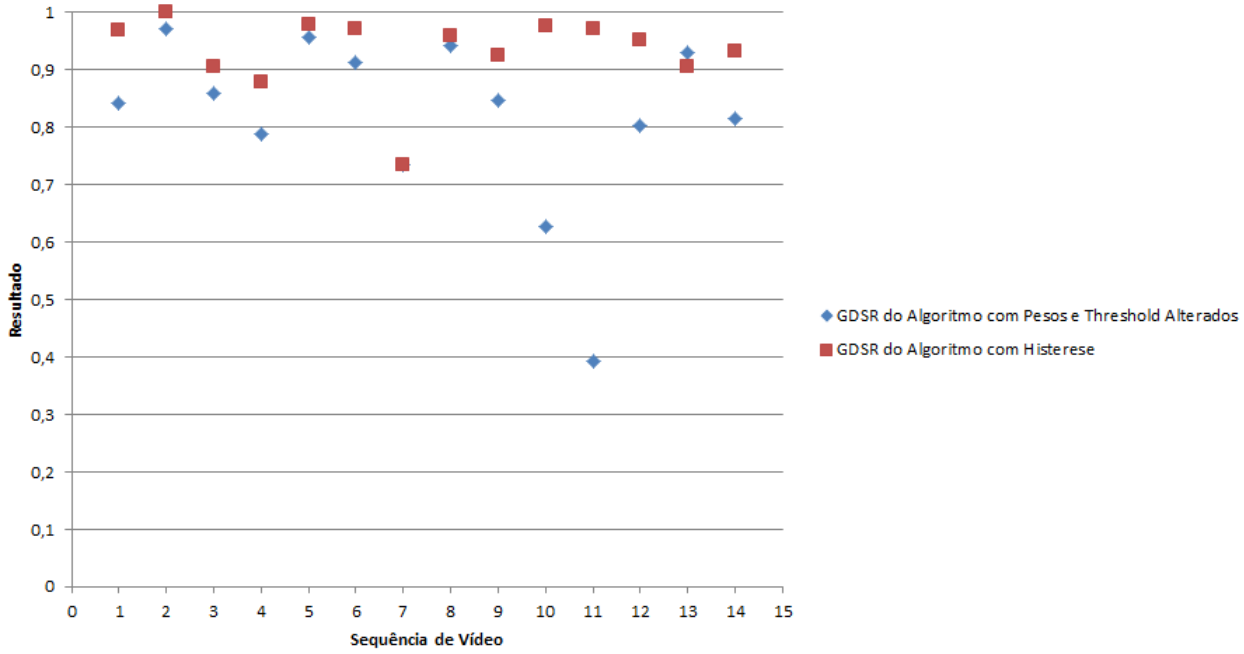


Figura 4.12: Comparação do GDSR do algoritmo com e sem histerese nas diferentes sequências

Na Figura 4.12 verifica-se que o GDSR das sequências é, com a introdução da histerese, superior a 90% em 11 das 13 sequências em teste. Tal significa que mais de 90% dos grupos de cada $frame$ estão a ser corretamente identificados (sendo que um grupo corretamente detetado é um grupo que contém pelos menos 60% dos constituintes do grupo do *ground truth*). Neste caso, a sequência 7, por detetar erradamente dois grupos como um só durante uma parte da sequência, apresenta um valor de GDSR baixo (79%). No caso da sequência 13 a diferença entre valores de GDSR e precisão é significativa visto que o GDSR considera que um grupo se encontra bem identificado quando são incluídos pelos menos 60% das pessoas. Na Figura 4.13, podemos ver a comparação da deteção feita pelo algoritmo com histerese (quadro do lado direito) com a informação dada pelo *ground truth* (quadro do lado esquerdo) para um cenário idêntico ao da sequência 13. Se se avaliar a deteção do grupo quanto à precisão, ter-se-á de de considerar a deteção errada, pois nenhum dos dois grupos detetados (a laranja) terá um valor de sobreposição superior a 75% com o grupo de *ground truth*. Mas quando avaliada usando o GDSR, como um dos grupos detetados contém 60% dos elementos constituintes propostos pelo *ground truth* a deteção é contabilizada como válida. Esta diferença justifica o facto de nesta sequência o algoritmo ter uma precisão baixa (40%) mas um GDSR alto (superior a 90%).

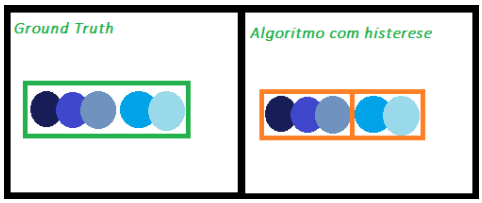


Figura 4.13: Deteção do grupo em fila segundo o *ground truth* (quadro do lado esquerdo) e segundo o algoritmo com *threshold* em histerese (quadro do lado direito)

4.3.4 Velocidade e Direção Médias

A velocidade e a direção são medidas muito ruidosas, tal como referido em [72]. O facto de no algoritmo original ([72]) se usar a velocidade e a direção instantâneas vem acentuar ainda mais esse problema. Numa tentativa de minimizar essa debilidade é proposto que o cálculo do parâmetro *speedStdDev* siga a formulação da Equação 4.3, onde s_i (velocidade instantânea) é substituído por s_i* (velocidade média das 5 últimas *frames*). Neste caso, $P_{x,i,k}$ é a coordenada x na *frame* k da pessoa i, $P_{x,i,k-5}$ é a coordenada x na *frame* $k-5$ da pessoa i, e de forma análoga $P_{y,i,k}$ é a coordenada y na *frame* k da pessoa i e $P_{yi,k-5}$ é a coordenada y na *frame* $k-5$ da pessoa i.

$$s_i* = \frac{1}{5}\sqrt{(P_{x,i,k} - P_{x,i,k-5})^2 + (P_{y,i,k} - P_{y,i,k-5})^2} \tag{4.12}$$

A substituição da forma de cálculo da velocidade e da direção traduz-se numa melhoria da precisão de cerca de 1%, de *recall* em 2% de GDSR de 0.3% tal como pode ser observado na Figura 4.14.

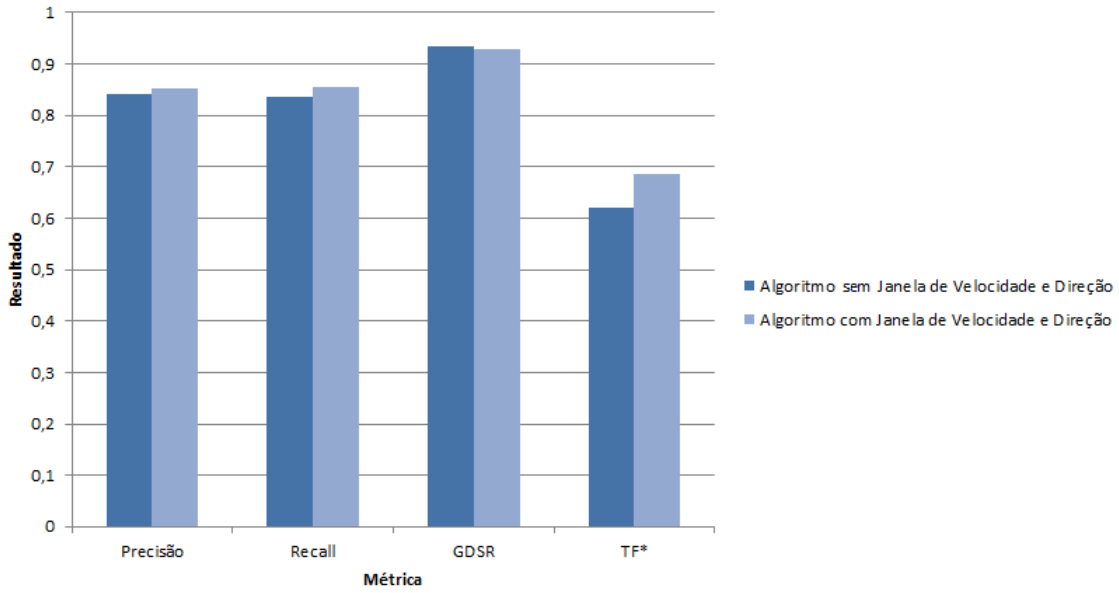


Figura 4.14: Comparação da desempenho do algoritmo com velocidade e direção instantâneas e velocidade e direção médias em precisão, *recall*, GDSR e TF*

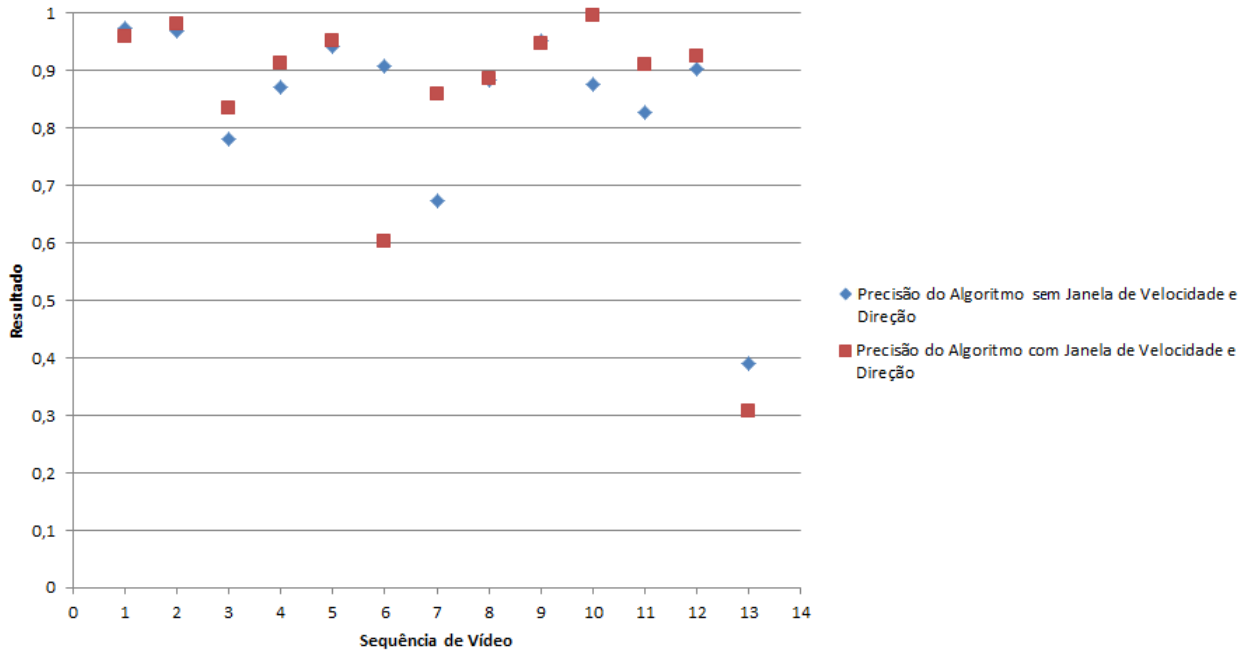


Figura 4.15: Comparação da precisão do algoritmo com velocidade e direção instantâneas e velocidade e direção médias nas diferentes sequências

Na Figura 4.15 pode verificar-se que ocorre uma melhoria em 7 das 13 sequências em termos de precisão, provando que o uso da velocidade e da direção média contribui para um melhoramento do desempenho geral do algoritmo .

4.3.5 Outras Formulações de Distâncias

4.3.5.1 Distância ao Centroide

Um dos parâmetros necessários para calcular o *GI* é a distância média entre as pessoas. A avaliação desta pode induzir em erro quando se pretende detetar grupos na formação do caso da Figura 4.16. Comparando a distância média entre as pessoas do quadro do lado esquerdo com as distâncias médias das pessoas do quadro do lado direito, observa-se que estas são idênticas, embora no quadro do lado esquerdo esteja apenas um grupo e no do lado direito esteja um grupo e uma pessoa. Se o aumento da distância média for muito pequeno, o algoritmo poderá não ser suficientemente sensível ao ponto de detetar a diferença e assim proceder a um junção incorreta das pessoas num só grupo (no caso do quadro do lado direito).

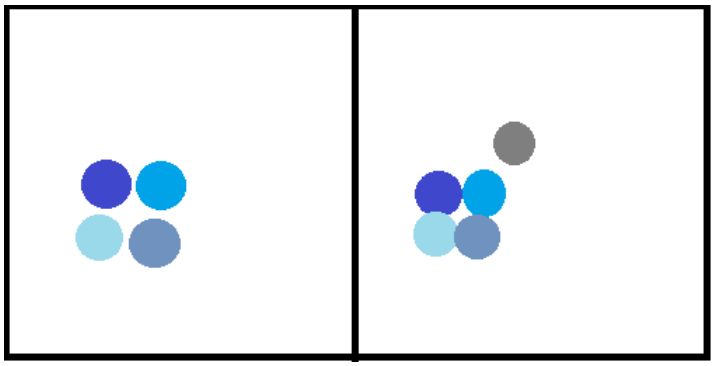


Figura 4.16: Comparação de dois conjuntos de pessoas distintos com média de distâncias semelhantes

Isto é o que acontece no cenário da sequência 6, cuja ilustração se encontra na Figura 4.17. Nesta, há dois grupos, cada um composto por quatro pessoas, em que com o uso da distância média, o algoritmo tende a juntar todas as pessoas num só grupo. Note-se que como temos dois grupos de quatro pessoas muito juntos, a distância média não é suficientemente elevada para que o GI suba ao ponto do algoritmo se decidir pela separação. Como tal, é proposto o uso da distância média das pessoas ao centroide visto que este representa o ponto central do grupo. Visto que os valores médios ao centroide são mais pequenos do que as distância médias entre pessoas, é necessário proceder a nova normalização. Respeitando o método de normalização discrito na Equação 4.6, max adquire o valor de 400 pixeis em vez de 900 pixeis.

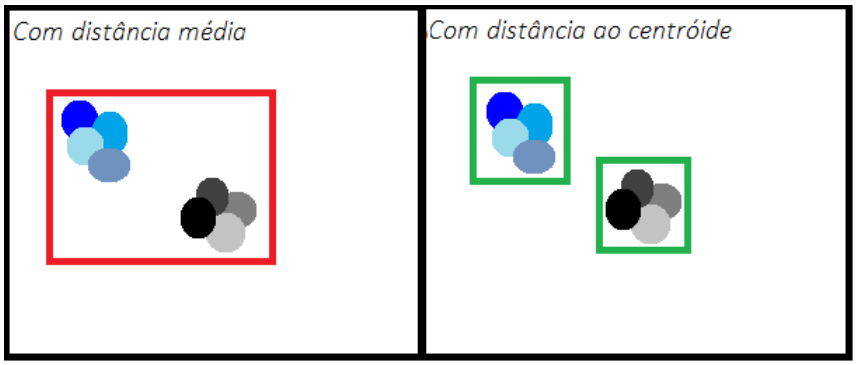


Figura 4.17: Comparação da deteção de grupos do algoritmo com distância média e com distância média ao centroide

Esta substituição apresenta uma melhoria na precisão (3.5%), *recall* (cerca de 3%), de GDSR (0.3%) e fragmentação das trajetórias (diminui de 0.45 fragmentações por trajetória para 0.43). Tais resultados demonstram que o uso do da distancia média ao centroide é mais adequada do que a distância média entre as pessoas.

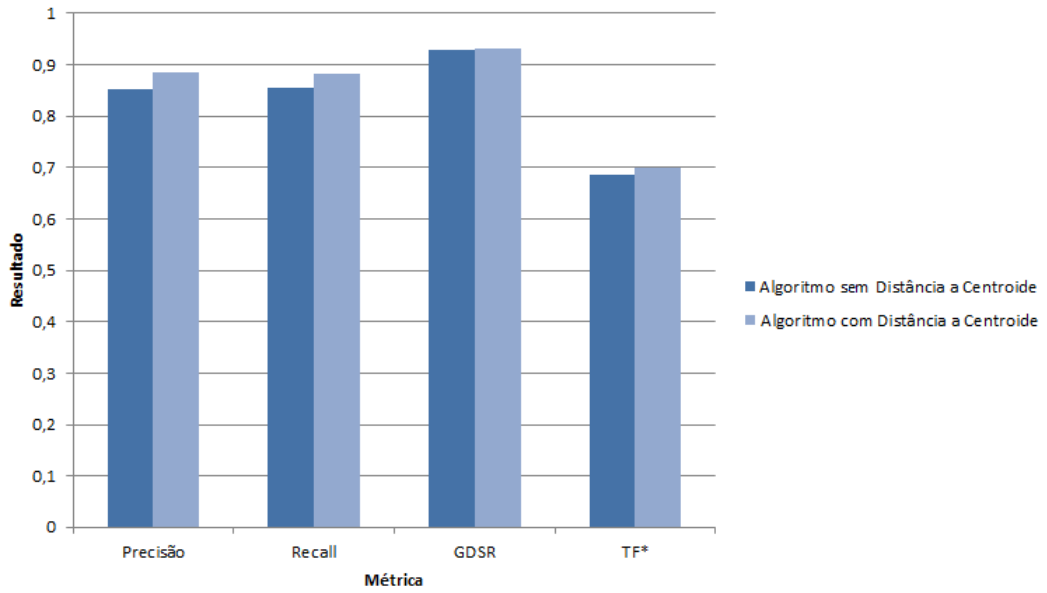


Figura 4.18: Comparação do desempenho geral do algoritmo com e sem histerese nas métricas precisão, *recall*, GDSR e TF*

Observando a Figura 4.19, conclui-se que a distância ao centroide provoca um significativo melhoramento nas sequências 6 e 13. As perdas de desempenho na sequência 7 deve-se ao facto de quando os grupos se cruzam, a sua distância ao centroide torna-se ainda menor. Na Figura 4.20 temos uma ilustração do cenário nesta sequência: dois grupos movimentam-se em direções opostas, ficando os seus elementos muito próximos durante um intervalo de tempo. O uso da distância média ao centroide nesses momentos desce significativamente e o peso dado à direção não é suficiente para evitar que o algoritmo junte todos os elementos num só grupo. Assim, tal como visto na Figura 4.20, em vez do algoritmo detetar dois grupos distintos (quadros superiores), o algoritmo junta-os (quadros inferiores) baixando assim o desempenho nesta sequência.

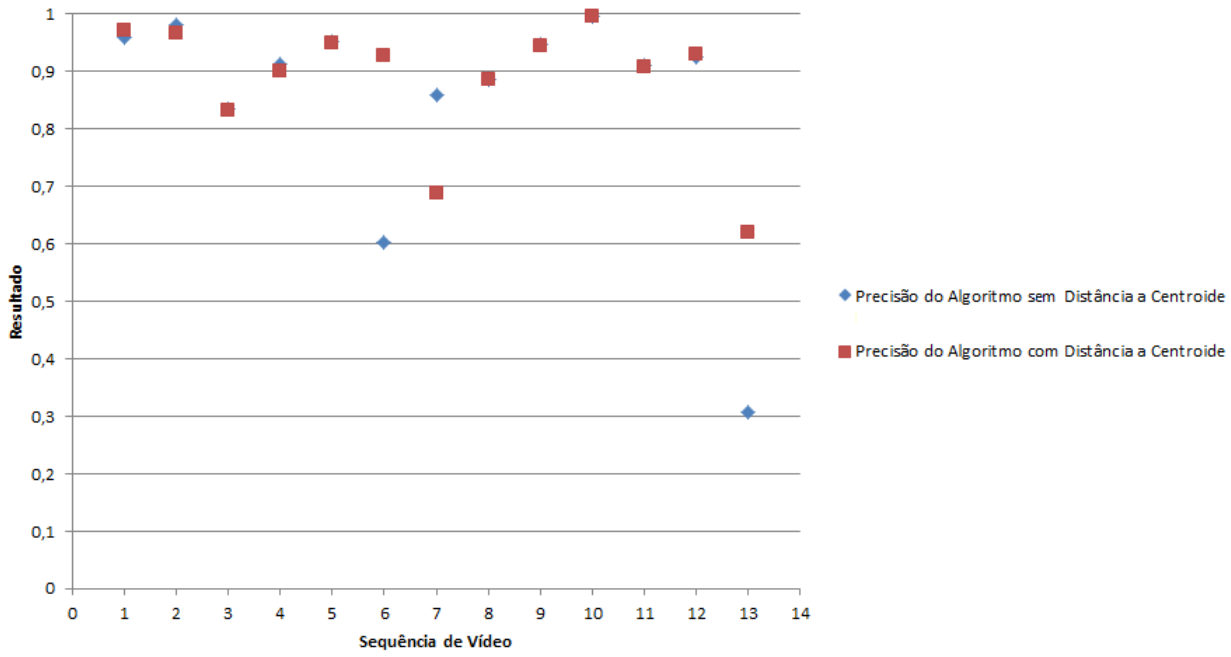


Figura 4.19: Comparação da precisão ao longo das sequências do algoritmo com distância média entre as pessoas com o algoritmo com distância média ao centroide nas diversas sequências

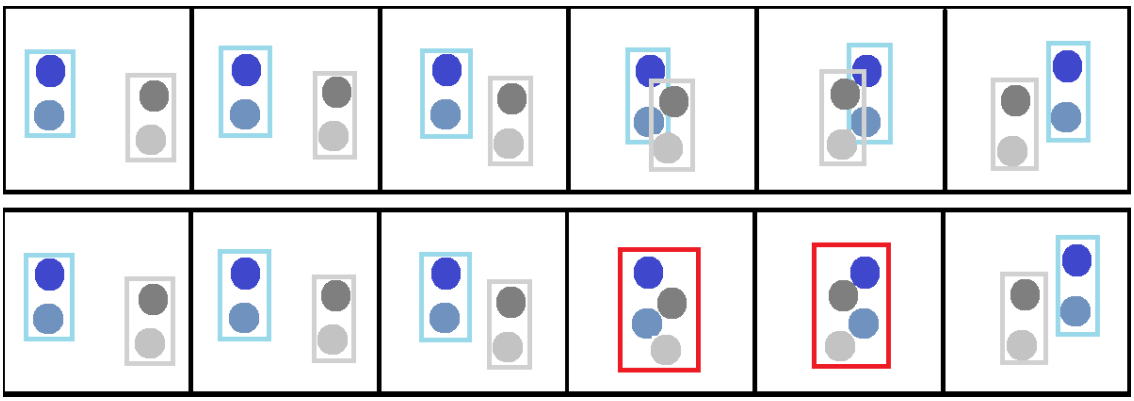


Figura 4.20: Comparação da resposta do algoritmo com o uso da distância média ao centroide com o *ground truth*. Nos quadros superiores é ilustrada a resposta correta e nos quadros inferiores a deteção incorreta dos grupos (a vermelho)

4.3.5.2 Distância ao Centroide e às duas pessoas mais próximas

A distância ao centroide, embora melhor que a distância média entre pessoas, poderá induzir em erro o algoritmo, pois as pessoas, embora possam formar um grupo, poderão estar longe do centroide, tal como demonstra a Figura 4.23. Nesta figura estão presentes três grupos distintos em que a distância média ao centroide e a distância média aos dois mais próximos tomam importâncias diferentes. O grupo do lado esquerdo (escala de cinzas) teria distância média ao centroide e distância média aos dois mais próximos idênticas. Mas

se comparamos com o grupo do meio (escala de azuis), vemos que embora se esteja na presença de um grupo, a distância média ao centroide já é maior, e se avaliarmos a distância média ao aos dois mais próximos esta já é mais pequena. O grupo em fila (lado esquerdo em escala de vermelhos) embora tenha uma distância média aos dois mais próximos pequena, tem uma distância ao centroide elevada. Um grupo poderá tomar qualquer uma destas três formas, revelando assim a importância da distância a ser calculada tendo também em conta a distância às duas pessoas mais próximas. A normalização da distância aos dois elementos mais próximos foi feita determinando o valor mais elevado dessa encontrado em todas as sequências e dividindo a distância por esse valor.

Na tentativa de incluir a distância média aos dois elementos mais próximos no cálculo da distância média, procedeu-se a um determinação de pesos tendo-se chegado à Equação 4.13, onde $distancia'$ é a nova distância proposta, $distanciaMediaCentroide$ é a distância média das pessoas ao centroide e $distance2Melhores$ é a distância média aos dois elementos mais próximos.

$$distância' = 0.75 \times distânciaMediaCentroide + 0.25 \times distância2MaisPróximos \quad (4.13)$$

A escolha de pesos foi determinada pela maximização da precisão do algoritmo nas sequências em estudo fazendo-se variar os pesos entre 0 e 1, com incrementos de 0.01.

A adição da distância aos dois elementos mais próximos, em conjunto com a distância média ao centroide, tal como se pode verificar na Figura 4.21, traduz-se num aumento de precisão (2%), de *recall* (cerca de 3%), de GDSR (0.3%) e do desempenho na fragmentação (a fragmentação diminui de 0.43 para 0.41 fragmentações por trajetória). A introdução desta melhoria leva a que o algoritmo apresente em média uma precisão e *recall* superiores a 90%, continuando ainda a aproximar o GDSR do seu valor ideal (100%).

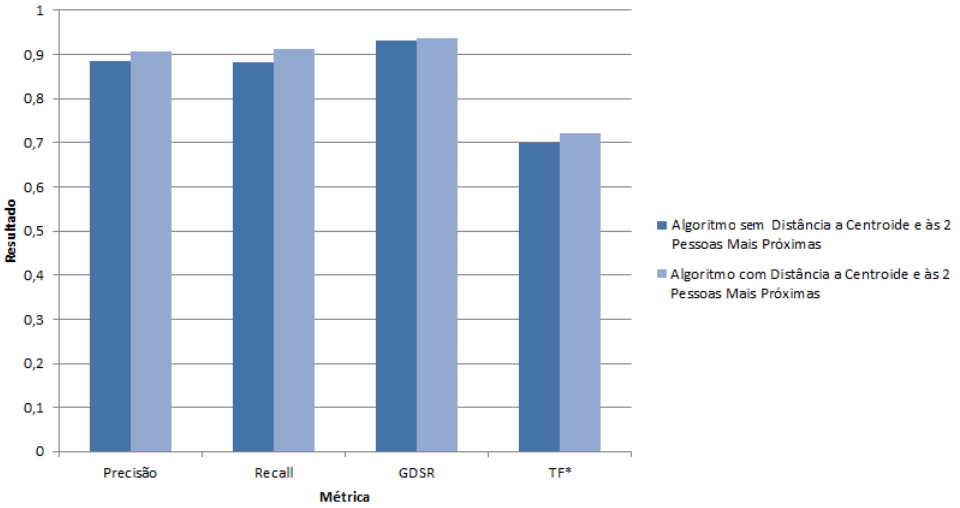


Figura 4.21: Comparação do desempenho geral do algoritmo com distância ao centroide com o algoritmo com distância média ao centroide e distância às duas pessoas mais próximas nas métricas precisão, *recall*, GDSR e TF*

Em relação aos valores de precisão das diversas sequências, na Figura 4.22, verifica-se um aumento significativo nas sequências com piores resultados (7 e 13) sem no entanto ocorrer uma perda significativa de desempenho em qualquer outra sequência. A sequência 7 apresenta uma melhoria de precisão de 15%. Nesta sequência há a interseção de dois grupos o que faz com que os elementos fiquem momentaneamente em linha (tal como se pode observar na ilustração 4.20). Quando tal acontece, a distância ao centroide é baixa mas a distância aos dois mais próximos aumenta (note-se que os extremos apresentam uma distancia elevada na média dos dois mais próximos) fazendo com que a média ponderada entre elas (Equação 4.13) obtenha valores elevados e o algoritmo não decida juntar todas as pessoas num grupo. Isto, juntamente com o facto de a sequência ser curta, faz com que se tenha um aumento significativo de precisão. Na sequência 15 é observada uma melhoria de precisão de 10%. Esta é uma das sequência em que a distância toma um papel muito importante visto que nesta há vários grupos parados. Note-se que as pessoas estando paradas, a velocidade e a direção tomam valores idênticos e consequentemente a distância vai ser o parâmetro que vai permitir ao algoritmo detetar os grupos.

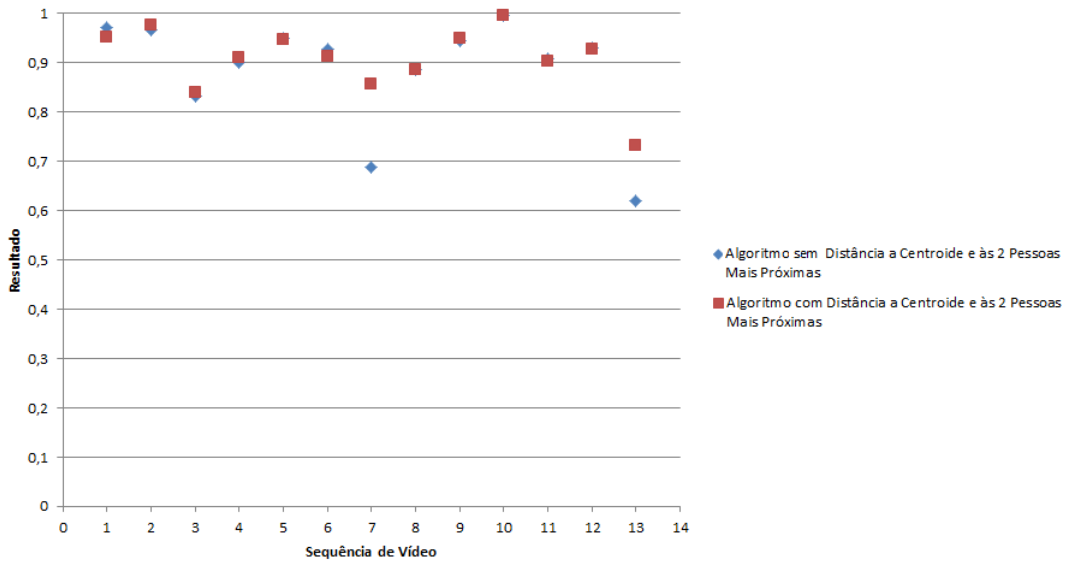


Figura 4.22: Comparação da precisão ao longo das sequências do algoritmo com distância ao centroide com o algoritmo com distância média ao centroide e distância às duas pessoas mais próximas

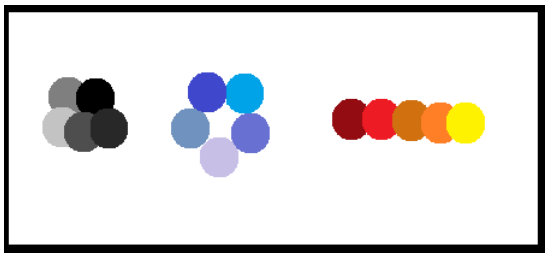


Figura 4.23: Ilustração de três grupos com disposições e distâncias ao centroide diferentes

4.3.5.3 Fator Tamanho do Grupo

O grupos podem ter vários tamanhos e, consequentemente, diferentes valores de distância (neste caso, de distância ao centroide e distância às duas pessoas mais próximas). Na Figura 4.24 temos três grupos distintos com um número de pessoas diferente. Analisando as distâncias médias ao centroide vemos que estas aumentam com o número de pessoas do grupo. O grupo do meio, que contém cinco pessoas, terá uma distância média ao centroide superior ao grupo que tem três (grupo do lado esquerdo) e menor do que o grupo que contém sete (grupo do lado direito). Por este motivo é proposto que a média das distâncias contenha um fator multiplicativo que varie com o número de pessoas do grupo de acordo com a Equação 4.14, onde $distancia''$ corresponde à nova distância proposta e $distancia'$ à distância proposta na subsecção 4.3.5.2 (média ponderada entre a distância média ao centroide e a distância média às duas pessoas mais próximas).

$$distancia'' = distancia' \times (1.5 - \frac{n_{pessoas}}{20}) \tag{4.14}$$

O peso a adicionar à formulação foi escolhido através de experimentação e otimização de valores de modo a melhorar a precisão geral do algoritmo.

Conforme apresentado na Figura 4.25, a melhoria mais significativa ocorre no *recall* (2%) embora a precisão também tenha melhorado (0.5%). O GDSR e fragmentação da trajétoria não sofrem uma alteração significativa (menor que 0.5%).

Analisando os resultados da precisão para as várias sequências, na Figura 4.26, verifica-se que os níveis de precisão se encontram todos acima de 80%, significando que o algoritmo com as atuais alterações já procede eficientemente à deteção dos grupos.

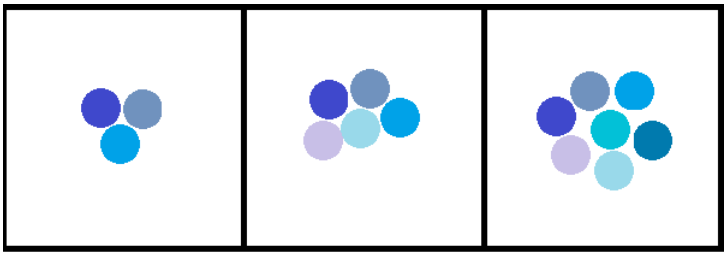


Figura 4.24: Ilustração do aumento da distância ao centroide com o aumento do número de pessoas no grupo

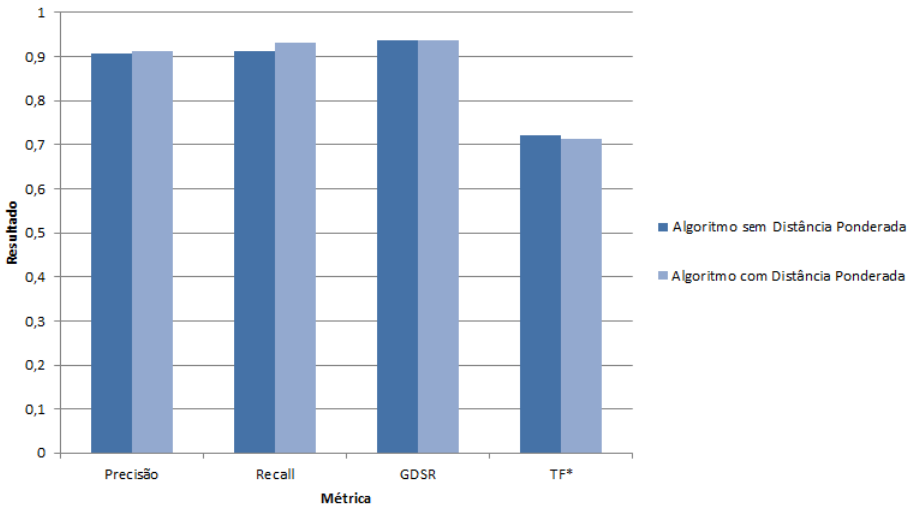


Figura 4.25: Comparação do desempenho geral do algoritmo sem e com o fator tamanho do grupo nas métricas precisão, *recall*, GDSR e TF*

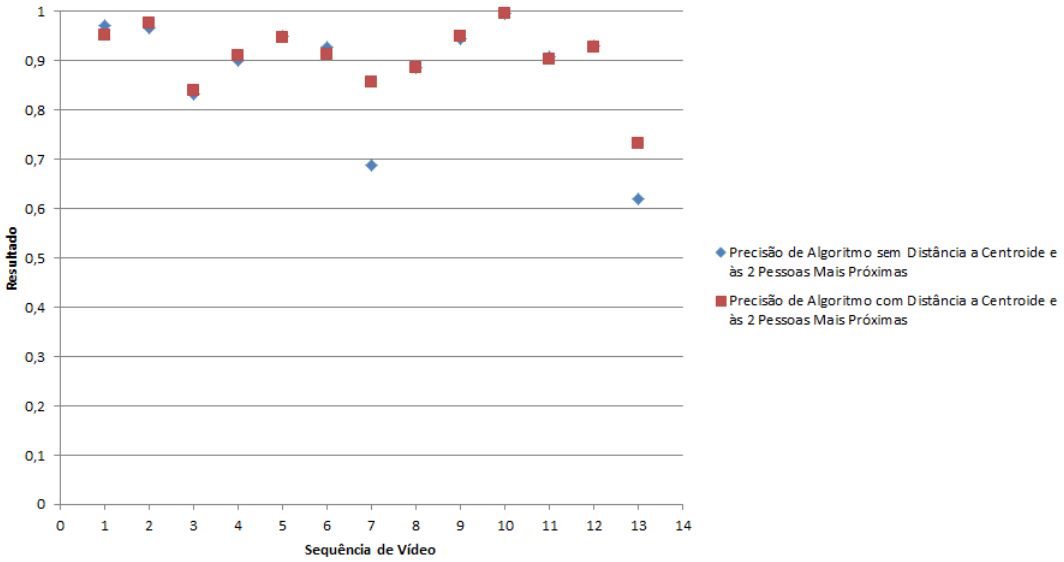


Figura 4.26: Comparação da precisão ao longo das sequências do algoritmo sem e com o fator tamanho do grupo

4.3.6 Função Não Linear

A classificação de um conjunto de pessoas depende de vários fatores: distância, velocidade e direção. Quando numa cena se encontram pessoas paradas, o fator que toma maior importância é a distância, mas se as pessoas estiverem em movimento, a velocidade e a direção ganharão relevância na decisão.

Tendo isto em conta, é proposto um GI que utilize pesos distintos de acordo com a situação, conforme a Equação 4.15, respeitando o diagrama de blocos da Figura 4.27.

$$\begin{cases} GI_1 = 0.85 * distanceAvg + 0.10 * speedStdDev + 0.05 * directionStdDev \\ GI_2 = 0.6 * distanceAvg + 0.15 * speedStdDev + 0.25 * directionStdDev \\ GI_3 = 0.75 * distanceAvg + 0.15 * speedStdDev + 0.10 * directionStdDev \end{cases} \tag{4.15}$$

Para as sequências usadas, a velocidade média é considerada baixa se for inferior a 0.2 e alta se tomar valores superiores a 0.75. Os valores dos pesos e dos limiares de decisão da velocidade média (0.2 e 0.75) foram obtidos experimentalmente de modo a melhorar os casos relevantes sem no entanto diminuir o desempenho nos restantes.

Atendendo aos resultados da Figura 4.28, verifica-se que embora as alterações de precisão sejam reduzidas, há uma melhoria na sequência 6 e na sequência 11 (Figura 4.29). Na sequência 6 há a junção de pessoas a grupos que se encontram parados. Como a velocidade média é baixa nestes casos, é usado o GI_1 que dá maior importância à distância. Assim, quando a pessoa começa a afastar-se do grupo, o algoritmo deteta mais rapidamente a mudança na estrutura do grupo. O mesmo acontece quando uma pessoa se junta. Na sequência 11 há uma fila parada onde vão entrando e saindo pessoas. O algoritmo, tal

como na sequência 6, deteta mais rapidamente estas alterações, resultando num aumento da precisão.

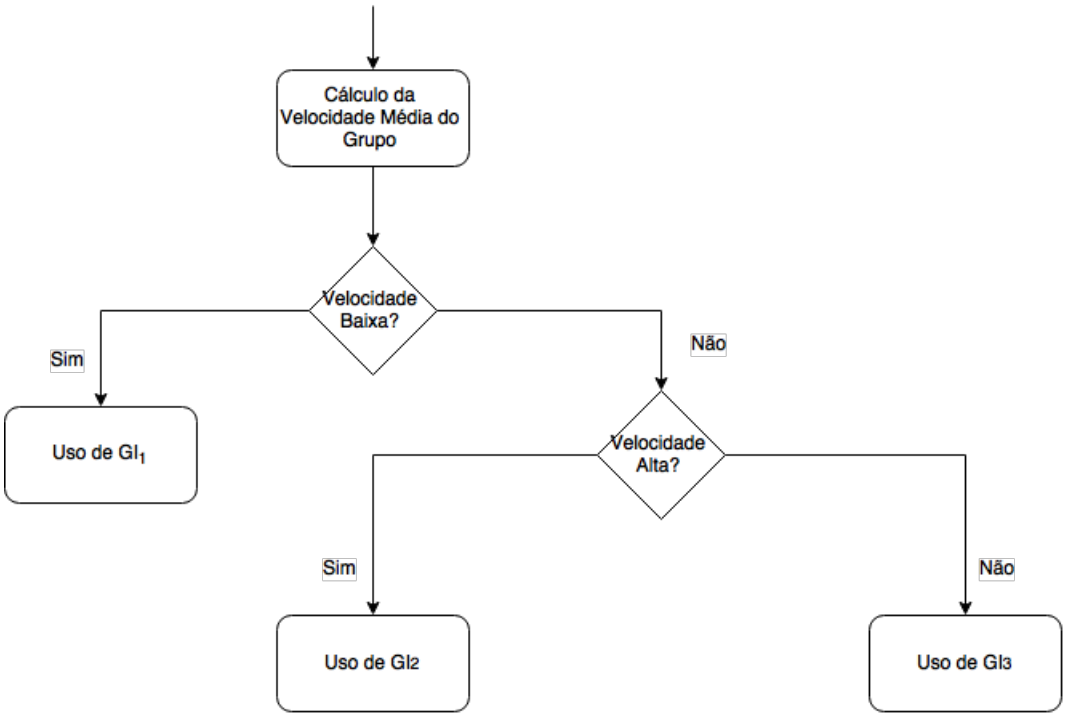


Figura 4.27: Esquema da decisão para o cálculo do GI. O cálculo da velocidade média será usado como fator decisivo para a escolha dos pesos a atribuir às componentes do GI

.

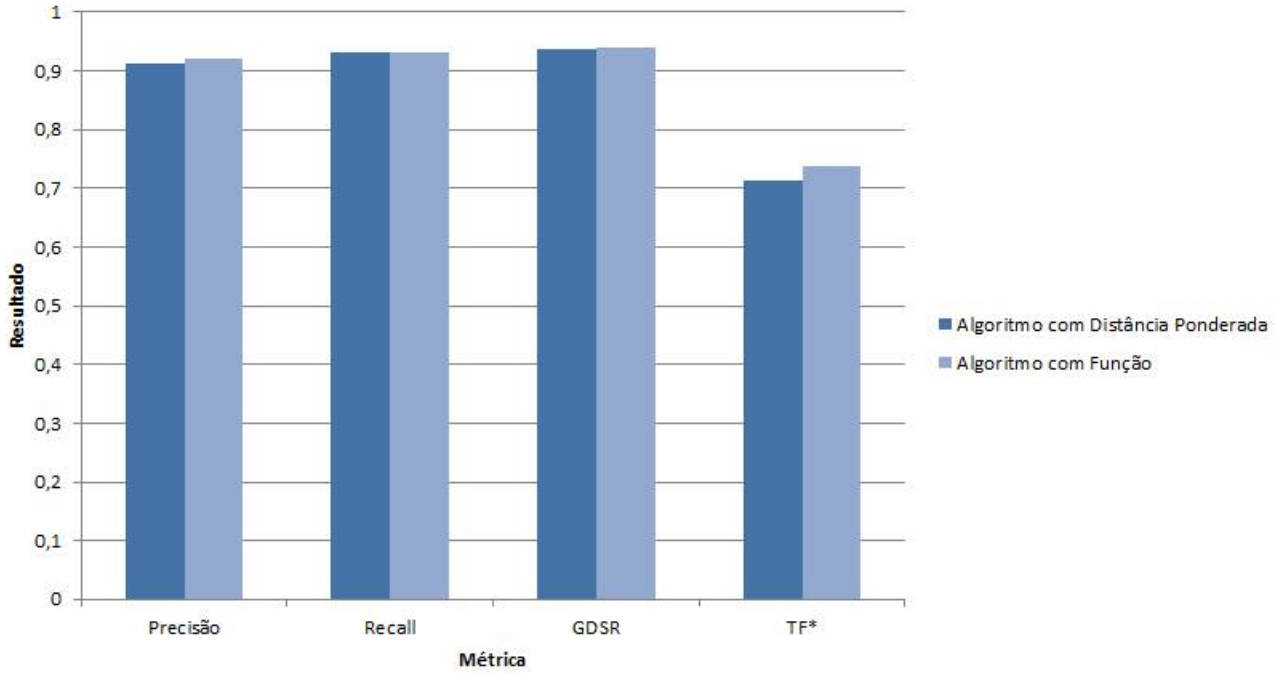


Figura 4.28: Comparação do desempenho geral do algoritmo sem e com o uso da função não linear nas métricas precisão, *recall*, GDSR e TF*

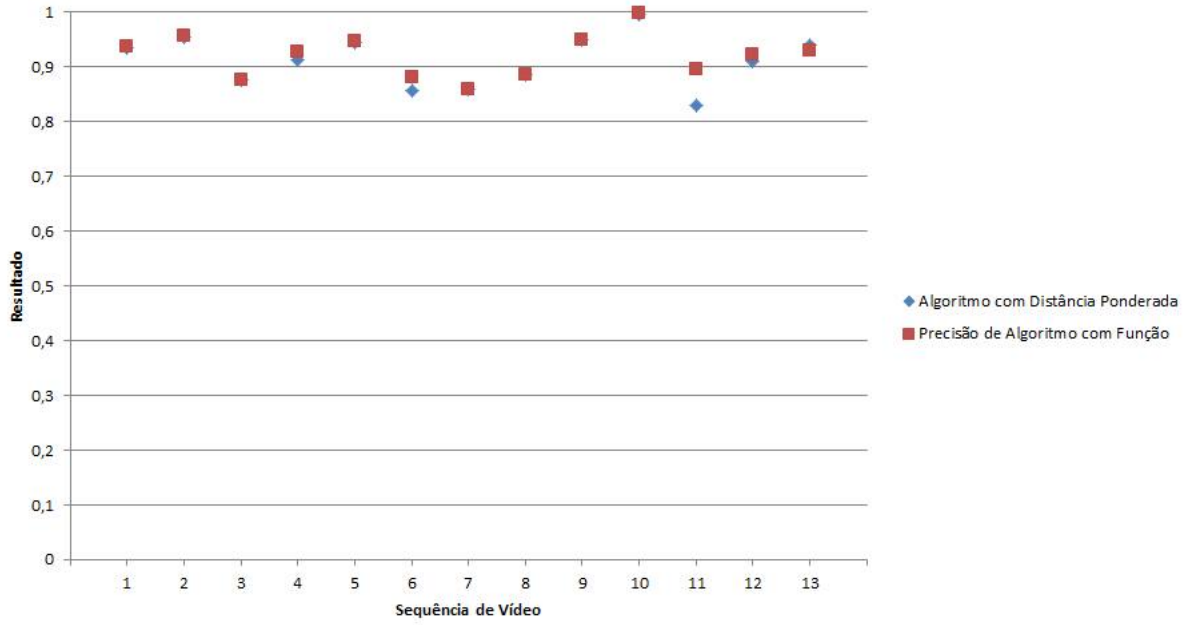


Figura 4.29: Comparação da precisão ao longo das sequências do algoritmo sem e com o uso da função não linear

4.3.7 Ângulo e Direção Média

Embora o uso da direção apresente bons resultados para as sequências, esta poderá não ser a formulação mais correta noutras situações. Observando a Figura 4.30, verifica-se que embora as pessoas estejam a tomar direções opostas, apenas a direção em y é diferente. Tal poderá não ser suficiente para a medida de direção do GI se alterar significativamente, pois a alteração apenas ocorre num único eixo.

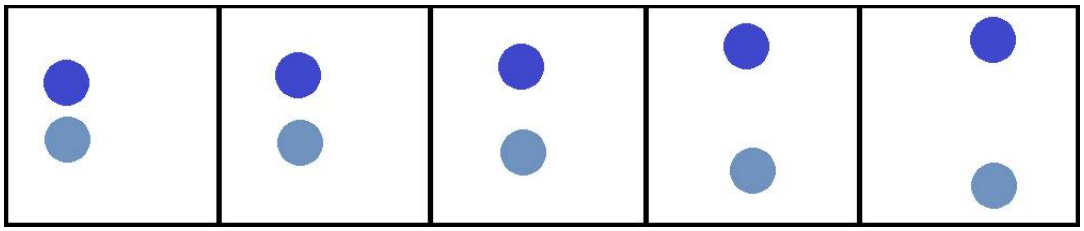


Figura 4.30: Movimento simulado de duas pessoas onde a métrica de direção poderá não ser significativa. O eixo horizontal é o eixo dos XX e vertical dos YY. Verifica-se que, apesar de seguirem direções distintas, esta diferença apenas ocorre face a um eixo (vertical), pelo que a definição da distância poderá não ter um valor suficientemente alto para separar os grupos.

De forma a que estas situações sejam corretamente contempladas, foi alterada a medida da direção do GI de modo a incluir a informação do ângulo. Atendendo à posição e perspetiva da câmara, o ângulo de uma pessoa pode ser estimado através da Equação 4.16, onde $\triangle y$ corresponde à diferença da coordenada y da pessoa numa janela, que foi considerada com valor 6, e $\triangle x$ o equivalente na coordenada x (foi tomado o eixo horizontal como o eixo dos XX e vertical como eixo dos YY). Tal permite calcular o ângulo médio de um conjunto de pessoas, que define o ângulo para o qual o grupo se desloca. É depois calculado o desvio padrão dos ângulos, conforme indica a Equação 4.17, tomando o ângulo mais curto de acordo com a Equação 4.18.

$$\hat{a}ngulo_{estimado} = atan_2(\frac{\triangle y}{\triangle x}) \tag{4.16}$$

$$desvio_{\hat{a}ngulo} = \sqrt{\frac{1}{N}\sum_{i=1}^{n_{pessoas}} (\hat{a}ngulo_{mínimo})^2} \tag{4.17}$$

$$\begin{aligned} \hat{a}ngulo_{mínimo} = min(|&max(\hat{a}ngulo_{pessoa}, \hat{a}ngulo_{médio}) - \\ &min(\hat{a}ngulo_{pessoa}, \hat{a}ngulo_{médio}) + 180|, \\ &|\hat{a}ngulo_{pessoa} - \hat{a}ngulo_{médio}|) \end{aligned} \tag{4.18}$$

Assim, a direção usada no GI é substituída pela definição da Equação 4.19, onde $desvio_{direção}$ corresponde à definição original apresentada no artigo [72] e $desvio_{angulo}$ é o

definido na Equação 4.17 normalizado a 360º. A escolha dos pesos foi feita de modo a que casos semelhantes ao da Figura 4.30 não sejam detetados como grupo mas sem no entanto tornar o algoritmo demasiado sensível a pequenas variações de ângulo.

$$direcao' = 0.75 \times desvio_{direção} + 0.25 \times desvio_{angulo} \tag{4.19}$$

Na Figura 4.31, não se verificam à primeira vista alterações significativas, mas atendendo ao apresentado na Figura 4.32, a sequência 2 apresenta uma melhoria na sua precisão. Assim, embora para estas sequências a introdução do ângulo não traduza num aumento muito significativo de desempenho (apenas na sequência 2 o aumento é bem visível), trata-se de uma adição que ajuda o algoritmo a melhor decidir, em casos particulares, se um conjunto de pessoas é de facto um grupo. Verifica-se que nos restantes casos há um ligeiro aumento da precisão não havendo nenhum caso onde esta adição prejudique a avaliação.

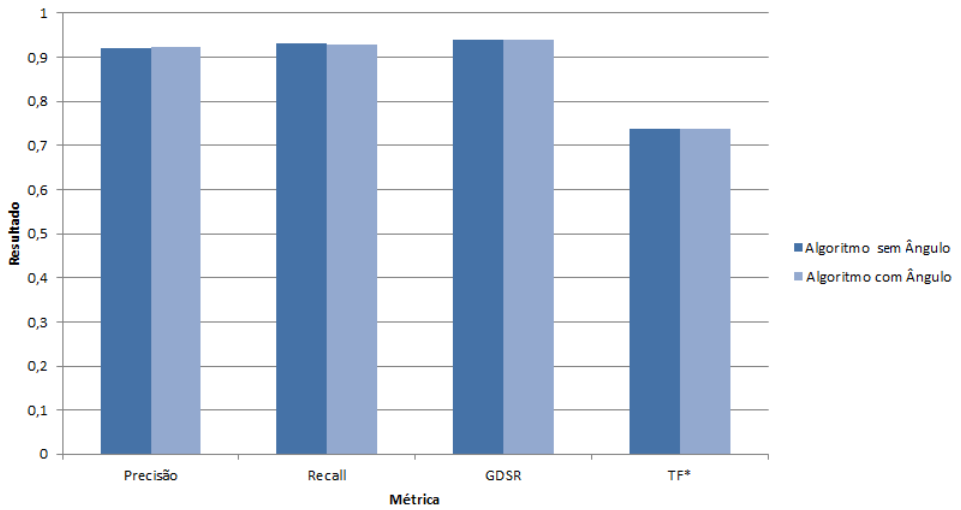


Figura 4.31: Comparação do desempenho geral do algoritmo sem e com o uso o ângulo nas métricas precisção, *recall*, GDSR e TF*

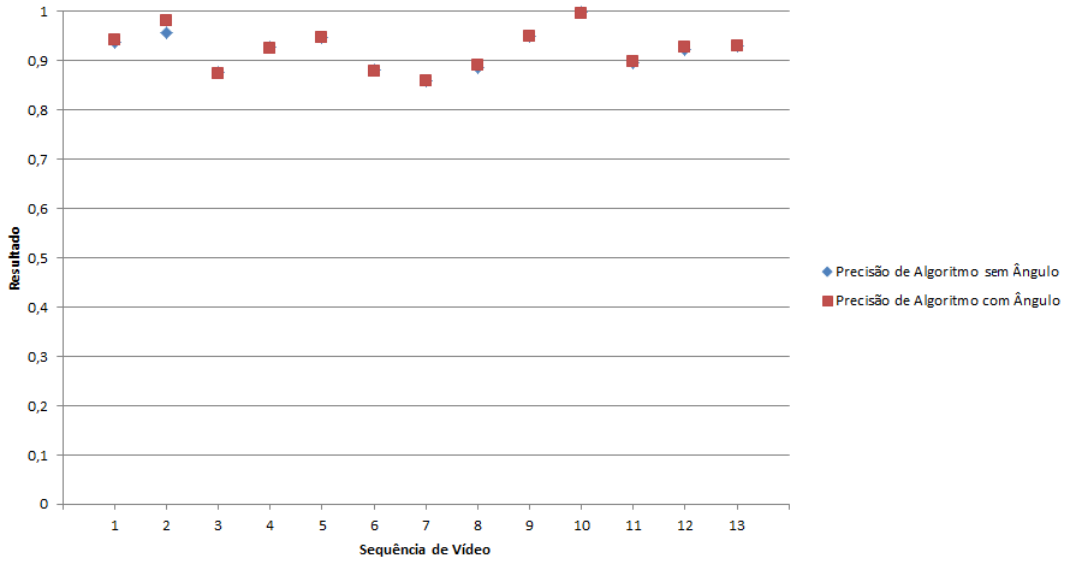


Figura 4.32: Comparação da precisão do algoritmo sem e com o uso do ângulo nas várias sequências

4.3.8 Resultados

O algoritmo de deteção de grupos proposto em [72] e abordado na secção 4.2 apresenta um bom desempenho quando a sua entrada não contém erros. Contudo, com o aumento do nível de erros, o seu desempenho decresce significativamente, tal como se pode ver na Figura 4.33. Comparando o resultado do algoritmo com entrada sem erros (erros tipo 0) e com entrada sujeita a erros mais agressivos (erros tipo III), observa-se um decréscimo de 40% na precisão, 25% em *recall*, 30% em GDSR e um aumento da fragmentação média por entidade (TF) de 0.45 fragmentações por trajetória para 6.69.

Assim, pode-se concluir que o algoritmo, embora apto a lidar com entradas livre de erros, não consegue manter o seu bom desempenho quando surgem erros, mostrando resultados piores à medida que os erros aumentam na entrada.

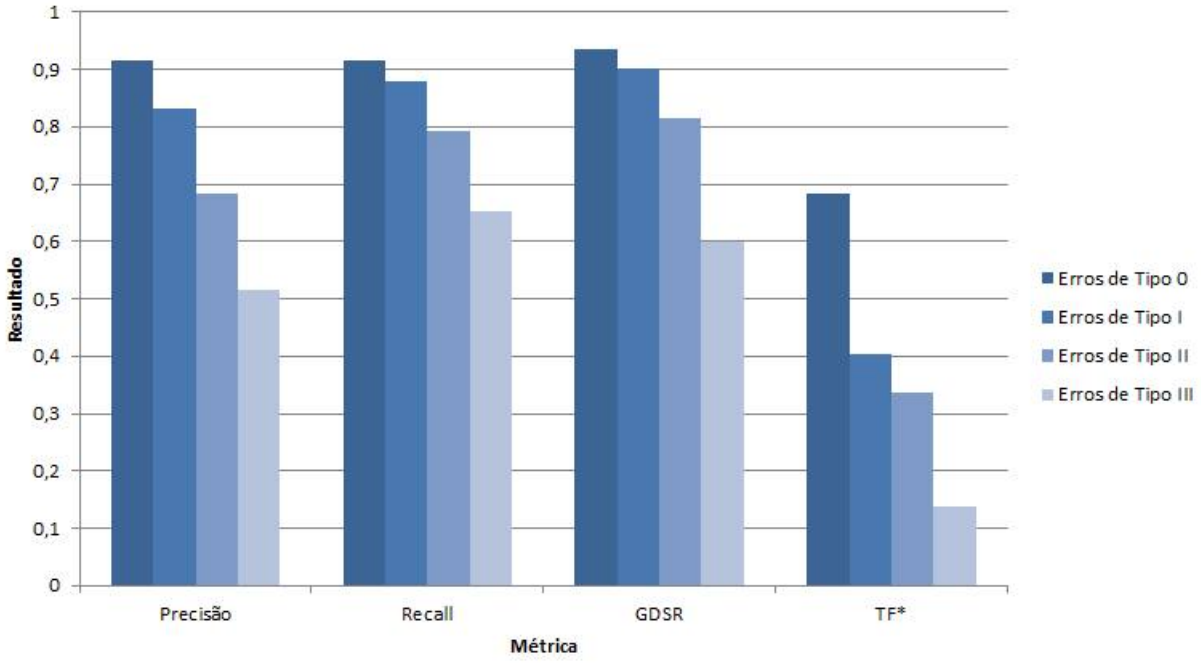


Figura 4.33: Evolução do desempenho do algoritmo base perante os diversos tipos de erro

Ao contrário do algoritmo original, o algoritmo com as melhorias propostas nesta secção consegue manter o seu bom desempenho com e sem a presença de erros na entrada. Observa-se na Figura 4.34 que o algoritmo melhorado consegue manter níveis de precisão, *recall* e GDSR superiores a 90% quando sujeito a entradas com e sem erros de tipo I e II. A resposta ao ruído do tipo III revela uma diminuição de precisão de cerca de 10%, 7% de *recall* e 6% de GDSR. Em relação à fragmentação das trajetórias (TF), o algoritmo com entrada perfeita fragmenta em média 0.37 vezes cada trajetória, e com a entrada sujeita aos erros mais agressivos (erro tipo III) fragmenta 0.54, o que não é um decréscimo de desempenho muito significativo.

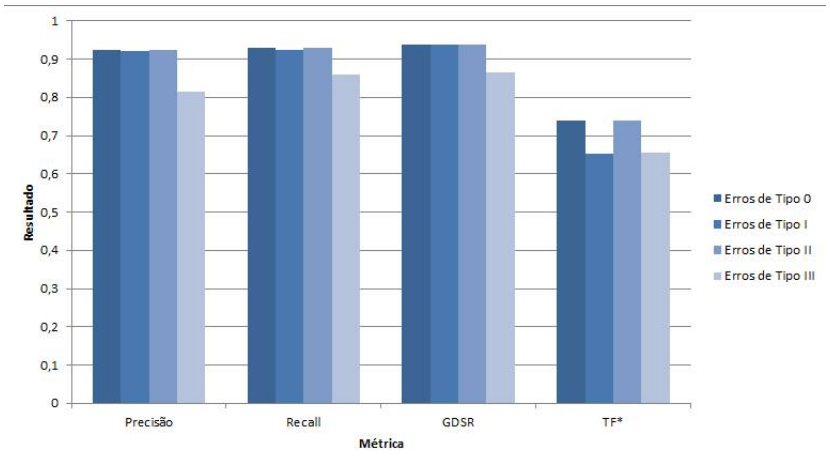


Figura 4.34: Evolução do desempenho do algoritmo modificado perante os diversos tipos de erro

Comparando o desempenho do algoritmo original com o algoritmo que inclui as melhorias propostas neste capítulo quando submetidos a entradas com ruído do nível II (Figura 4.36), observa-se um aumento de precisão (30%), de *recall* (10%), de GDSR (11%) e de fragmentação das trajetórias (de 2.08 fragmentações por entidade para 0.39).

Como pode ser concluído pela observação da Figura 4.35, o algoritmo com as melhorias propostas apresenta um melhor desempenho em todas as sequências independentemente da métricas usada para avaliar o algoritmo (precisão, *recall*, GDSR e TF*).

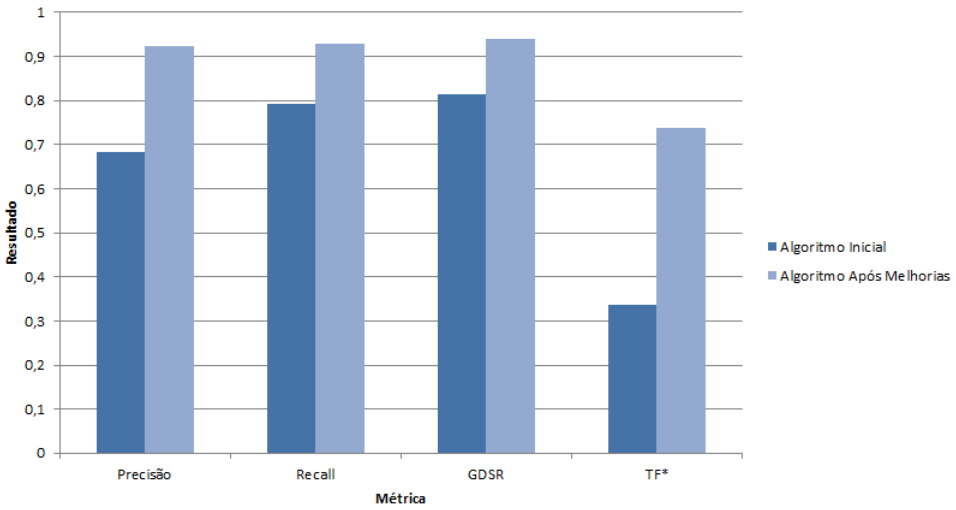
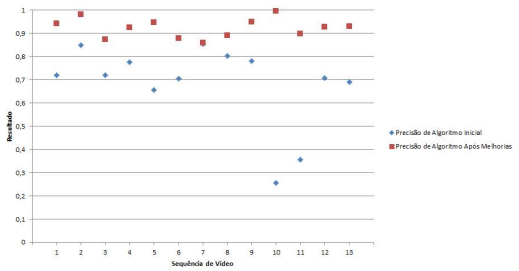
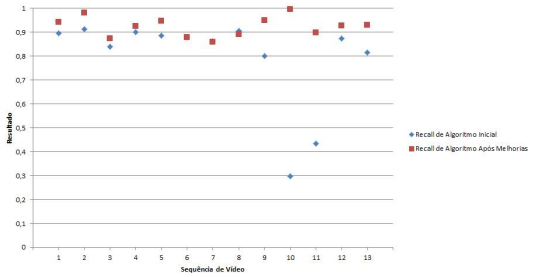


Figura 4.36: Comparação do desempenho do algoritmo base e algoritmo modificado no caso de erros de tipo II

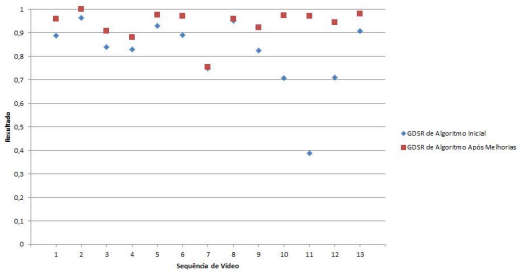
Por forma a comparar o algoritmo base, $GTBR$ ([72]), algoritmo modificado ($GTBR_m$) e o algoritmo de deteção do artigo que propõe as sequências usadas, $DEEPER-JIGT$ ([75]), é apresentada a Tabela 4.1 com os valores do GDSR para os respetivos algoritmos. Como se pode ver, o algoritmo $GTBR$ não apresentava valores de GDSR superiores ao



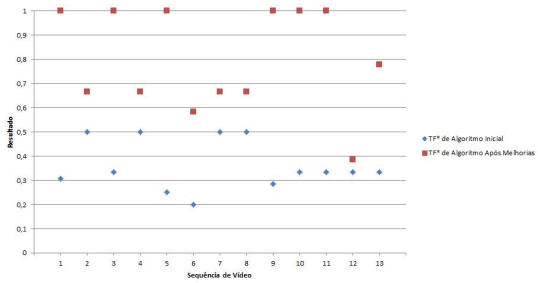
(a) Comparação da precisão do algoritmo original com o algoritmo melhorado



(b) Comparação do *recall* do algoritmo original com o algoritmo melhorado



(c) Comparação do GDSR do algoritmo original com o algoritmo melhorado



(d) Comparação da TF* do algoritmo original com o algoritmo melhorado

Figura 4.35: Comparação do desempenho do algoritmo original com o algoritmo melhorado em todas as sequências e nas métricas precisão, *recall*, GDSR e TF*

algoritmo $DEEPER-JIGT$. No entanto, após a integração das melhorias propostas ($GTBR_m$), o valor desta métrica atinge os 93%, apresentando uma melhoria de 5% face ao $DEEPER-JIGT$ e uma melhoria de 12% face ao algoritmo $GTBR$.

	GDSR
DEEPER-JIGT	88.46%
GTBR	81%
$GTBR_m$	**93%**

Tabela 4.1: Comparação do GDSR para três algoritmos distintos

Capítulo 5

Conclusões e Trabalho Futuro

5.1 Conclusões

O presente trabalho teve como objetivo a implementação de um algoritmo de deteção de grupos e eventos testado em sequências relevantes da Literatura. O algoritmo recebe como entrada o resultado de um possível algoritmo de seguimento. Quando a entrada não está corrompida com erros, o algoritmo base mostra um bom desempenho, com valores de precisão, *recall* e GDSR superiores a 90%. Verifica-se, no entanto, que com a adição de erros ocorre um decréscimo significativo na qualidade dos resultados, com vários valores inferiores a 75%. Assim, são dadas a conhecer possíveis modificações ao algoritmo original com vista a aumentar o seu desempenho quando as entradas estão sujeitas a erros.

Em primeiro lugar, o algoritmo foi testado com diferentes valores de pesos para a média das distâncias, desvios de velocidade e desvio de direção e verificou-se que os pesos originais não eram os que apresentavam melhores resultados. Verificou-se que com a adaptação dos pesos houve uma melhoria no desempenho do algoritmo.

Com a análise da resposta do algoritmo de deteção de grupos em ambiente com erros, conclui-se que o uso de um *threshold* torna o algoritmo muito sensível a perturbações, pelo que foi proposto o uso de um *threshold* em histerese, que melhorou significativamente a resposta do algoritmo, aumentando 10% a precisão.

Como as velocidades e as direções instantâneas se mostram muito sensíveis a pequenas oscilações nas posições das pessoas e tamanho das caixas, procedeu-se a uma substituição dessas por velocidades e direções médias, resultando numa melhoria em termos gerais, tendo no caso da precisão resultado num aumento de 3%.

Embora o algoritmo com as soluções propostas obtenha já melhores resultados que o original, foram identificadas algumas situações comuns em ambientes realistas onde o algoritmo não procedia a uma deteção correta dos grupos. Visto a distância ter um peso muito significativo na decisão (75%), foram analisados alguns casos específicos onde esta poderia piorar o desempenho e, como tal, foram analisadas alternativas ao cálculo da distância média. O objetivo destas alterações foi aperfeiçoar a resposta do algoritmo a

algumas situações particulares mas que ocorrem frequentemente em ambientes realistas. Com as alterações propostas, a distância média foi substituída por uma média ponderada à média das distância aos centroides e aos elementos mais próximos ajustada ao número de elementos e à distância máxima de um elemento ao centroide, obtendo-se melhorias em todas as métricas.

De forma a adaptar os pesos dos parâmetros quando os grupos se encontram parados ou em movimento, foi adicionada uma função não linear que, dependendo da velocidade média do grupo, ajusta os pesos dados à distância média modificada, desvio de velocidade e desvio de direção. Esta alteração permitiu um aumento de precisão nas situações onde algoritmo tinha pior desempenho, sem no entanto degradar o desempenho nas restantes.

Por último, a direção foi analisada e concluiu-se que a sua formulação poderia ser complementada com a determinação do valor do ângulo entre as pessoas e o grupo.

As melhorias introduzidas deram a capacidade ao algoritmo de lidar de forma mais robusta com entradas com erro (visto haver aumento dos valores de precisão, *recall* e GDSR) e de reconstruir mais fielmente as trajetórias das pessoas (TF* aumentou com as melhorias). Comparando com o algoritmo original verifica-se uma melhoria de 12% em GDSR. Face ao algoritmo que propõe o *dataset* na literatura, o presente algoritmo traz um aumento de desempenho de 5% em GDSR.

5.2 Trabalho Futuro

O algoritmo proposto foi avaliado recebendo como entrada a informação correta de seguimentos das pessoas (*ground truth* das pessoas) em cena, com introdução de erros comuns aos algoritmos de seguimento. Um primeiro ponto de trabalho futuro será a integração de um algoritmo de deteção e seguimento real de forma a testar a robustez do algoritmo. Um outro aspeto do uso de um algoritmo de seguimento real será a introdução de outros tipos de erros, tal como troca de identidade entre pessoas. Este tipo de erros, embora não tido em conta no algoritmo de gestão, são possíveis de ocorrer num ambiente de seguimento de pessoas, havendo interesse em tratá-lo.

A integração do algoritmo num outro de seguimento de pessoas por forma a haver troca de informação poderá ser interessante e obter bons valores de desempenho. Em vez de o algoritmo de gestão receber informação das pessoas proveniente do algoritmo de seguimento, estes dois poderão trocar informações apoiando-se mutuamente por forma a conseguir fazer as decisões com maior certeza.

A introdução de modelos de aparência para caracterizar os grupos é também uma possível proposta de trabalho futuro. Com isto adicionar-se-ia ao algoritmo de gestão a capacidade de lidar com o aparecimento e re-aparecimento de grupos, o que poderia ser particularmente útil quando se lidasse com sequências mais longas. Esta informação também poderia ser utilizada em situações onde o algoritmo de seguimento não consegue detetar as pessoas e é necessário re-identificar o grupo novamente como uma entidade. A

integração de métodos de re-identificação de pessoas no algoritmo de gestão de grupos permitiria reconhecer as pessoas que entram e saem dos grupos, auxiliando e complementando o algoritmo de seguimento de pessoas.

O uso de informação de calibração será também importante para conseguir obter o tamanho e a posição das pessoas em cena. Embora nas sequências usadas não houvesse a necessidade de o fazer, esta informação poderá ser útil para melhorar a performance noutras situações.

Testar o algoritmo com outros *datasets* criaria novos desafios à deteção e seguimento de grupos e seria interessante verificar se o algoritmo continua a ser capaz de lidar com essas alterações. Em particular, verificou-se que as sequências usadas eram curtas, pelo que poderá ser interessante estender os testes a sequências mais longas e avaliar se o desempenho se mantém.

Por fim, no contexto de segurança, terá interesse reconhecer o comportamento dos grupos por forma a prever situações anómalas. Com a informação dos grupos gerada pelo algoritmo de gestão, é possível a integração de métodos para prever situações que podem comprometer a segurança das pessoas.

Referências

[1] Michel e Francoise Gauquelin. Dicionario de psicologia. 1980.

[2] Stephen J McKenna, Sumer Jabri, Zoran Duric, Azriel Rosenfeld, e Harry Wechsler. Tracking groups of people. *Computer Vision and Image Understanding*, 80(1):42–56, 2000.

[3] Frédéric Cupillard, François Brémond, e Thonnat Monique. Tracking groups of people for video surveillance. Em *Video-Based Surveillance Systems*, páginas 89–100. Springer, 2002.

[4] Silveira Jacques Junior et al. Crowd analysis using computer vision techniques. *IEEE Signal Processing Magazine*, 27(5):66–77, 2010.

[5] Teng Li, Huan Chang, Meng Wang, Bingbing Ni, Richang Hong, e Shuicheng Yan. Crowded scene analysis: A survey. 2014.

[6] Ramin Mehran, Akira Oyama, e Mubarak Shah. Abnormal crowd behavior detection using social force model. Em *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, páginas 935–942. IEEE, 2009.

[7] Dan Kong, Douglas Gray, e Hai Tao. A viewpoint invariant approach for crowd counting. Em *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 3, páginas 1187–1190. IEEE, 2006.

[8] Saad Ali e Mubarak Shah. A lagrangian particle dynamics approach for crowd flow segmentation and stability analysis. Em *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, páginas 1–6. IEEE, 2007.

[9] S Cheung Sen-Ching e Chandrika Kamath. Robust techniques for background subtraction in urban traffic video. Em *Electronic Imaging 2004*, páginas 881–892. International Society for Optics and Photonics, 2004.

[10] Jaswant Jain e Anil Jain. Displacement measurement and its application in interframe image coding. *Communications, IEEE Transactions on*, 29(12):1799–1808, 1981.

[11] BPL Lo e SA 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. IEEE, 2001.

[12] Christopher Richard Wren, Ali Azarbayejani, Trevor Darrell, e Alex Paul Pentland. Pfinder: Real-time tracking of the human body. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):780–785, 1997.

[13] Paolo Remagnino, Adam Baumberg, Tom Grove, David Hogg, TN Tan, Anthony D Worrall, Keith D Baker, et al. An integrated traffic and pedestrian model-based vision system. Em *BMVC*, 1997.

[14] Berthold K Horn e Brian G Schunck. Determining optical flow. Em *1981 Technical Symposium East*, páginas 319–331. International Society for Optics and Photonics, 1981.

[15] Deepak Kumar Panda e Sukadev Meher. Dynamic background subtraction using texton co-occurrence matrix. 2014.

[16] Guang-Hai Liu e Jing-Yu Yang. Image retrieval based on the texton co-occurrence matrix. *Pattern Recognition*, 41(12):3521–3527, 2008.

[17] Olivier Barnich e Marc Van Droogenbroeck. Vibe: A universal background subtraction algorithm for video sequences. *Image Processing, IEEE Transactions on*, 20(6):1709–1724, 2011.

[18] Marc Van Droogenbroeck e Olivier Paquot. Background subtraction: Experiments and improvements for vibe. Em *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, páginas 32–37. IEEE, 2012.

[19] Luc Vincent. Morphological area openings and closings for grey-scale images. Em *Shape in Picture*, páginas 197–208. Springer, 1994.

[20] Kyungnam Kim, Thanarat H Chalidabhongse, David Harwood, e Larry Davis. Real-time foreground–background segmentation using codebook model. *Real-time imaging*, 11(3):172–185, 2005.

[21] Piotr Dollar, Christian Wojek, Bernt Schiele, e Pietro Perona. Pedestrian detection: An evaluation of the state of the art. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(4):743–761, 2012.

[22] Kobi Levi e Yair Weiss. Learning object detection from a small number of examples: the importance of good features. Em *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, páginas II–53. IEEE, 2004.

[23] David Gerónimo, Antonio López, Daniel Ponsa, e Angel D Sappa. Haar wavelets and edge orientation histograms for on–board pedestrian detection. Em *Pattern Recognition and Image Analysis*, páginas 418–425. Springer, 2007.

[24] Rainer Lienhart e Jochen Maydt. An extended set of haar-like features for rapid object detection. Em *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 1, páginas I–900. IEEE, 2002.

[25] Robert E Schapire e Yoram Singer. Improved boosting algorithms using confidence-rated predictions. *Machine learning*, 37(3):297–336, 1999.

[26] Navneet Dalal e Bill 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. IEEE, 2005.

[27] Corinna Cortes e Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[28] Navneet Dalal, Bill Triggs, e Cordelia Schmid. Human detection using oriented histograms of flow and appearance. Em *Computer Vision–ECCV 2006*, páginas 428–441. Springer, 2006.

[29] Christian Wojek e Bernt Schiele. A performance evaluation of single and multi-feature people detection. Em *Pattern Recognition*, páginas 82–91. Springer, 2008.

[30] Constantine Papageorgiou e Tomaso Poggio. A trainable system for object detection. *International Journal of Computer Vision*, 38(1):15–33, 2000.

[31] Yadong Mu, Shuicheng Yan, Yi Liu, Thomas Huang, e Bingfeng Zhou. Discriminative local binary patterns for human detection in personal album. Em *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, páginas 1–8. IEEE, 2008.

[32] Timo Ojala, Matti Pietikainen, e David Harwood. Performance evaluation of texture measures with classification based on kullback discrimination of distributions. Em *Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on*, número 1, páginas 582–585, 1994.

[33] Jianxin Wu, Matthew D Mullin, e James M Rehg. Linear asymmetric classifier for cascade detectors. Em *Proceedings of the 22nd international conference on Machine learning*, páginas 988–995. ACM, 2005.

[34] Xiaoyu Wang, Tony X Han, e Shuicheng Yan. An hog-lbp human detector with partial occlusion handling. Em *Computer Vision, 2009 IEEE 12th International Conference on*, páginas 32–39. IEEE, 2009.

[35] Timo Ahonen, Abdenour Hadid, e Matti Pietikäinen. Face recognition with local binary patterns. Em *Computer vision-eccv 2004*, páginas 469–481. Springer, 2004.

[36] Tian Hong, Wang Lixia, e Ding Xiaoqing. Pedestrian detection based on merged cascade classifier. Em *Soft Computing and Pattern Recognition (SoCPaR), 2011 International Conference of*, páginas 237–241. IEEE, 2011.

[37] Yoav Freund, Robert E Schapire, et al. Experiments with a new boosting algorithm. Em *ICML*, volume 96, páginas 148–156, 1996.

[38] Payam Sabzmeydani e Greg Mori. Detecting pedestrians by learning shapelet features. Em *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, páginas 1–8. IEEE, 2007.

[39] Serge Belongie, Jitendra Malik, e Jan Puzicha. Shape context: A new descriptor for shape matching and object recognition. Em *NIPS*, volume 2, página 3, 2000.

[40] Edgar Seemann, Bastian Leibe, Krystian Mikolajczyk, e Bernt Schiele. An evaluation of local shape-based features for pedestrian detection. Em *BMVC*, volume 5, página 10. Citeseer, 2005.

[41] Oncel Tuzel, Fatih Porikli, e Peter Meer. Region covariance: A fast descriptor for detection and classification. Em *Computer Vision–ECCV 2006*, páginas 589–600. Springer, 2006.

[42] Junqiu Wang, Yasushi Makihara, e Yasushi Yagi. Human tracking and segmentation supported by silhouette-based gait recognition. Em *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, páginas 1698–1703. IEEE, 2008.

[43] Ahmed M Elgammal e Larry S Davis. Probabilistic framework for segmenting people under occlusion. Em *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, páginas 145–152. IEEE, 2001.

[44] Bastian Leibe, Edgar Seemann, e Bernt Schiele. Pedestrian detection in crowded scenes. Em *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, páginas 878–885. IEEE, 2005.

[45] Kai Briechle e Uwe D Hanebeck. Template matching using fast normalized cross correlation. Em *Aerospace/Defense Sensing, Simulation, and Controls*, páginas 95–102. International Society for Optics and Photonics, 2001.

[46] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. Em *IJCAI*, volume 81, páginas 674–679, 1981.

[47] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, 82(1):35–45, 1960.

[48] Hieu Tat Nguyen e Arnold WM Smeulders. Fast occluded object tracking by a robust appearance filter. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(8):1099–1104, 2004.

[49] Yizong Cheng. Mean shift, mode seeking, and clustering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(8):790–799, 1995.

[50] Frank J Aherne, Neil A Thacker, e Peter I Rockett. The bhattacharyya metric as an absolute similarity measure for frequency coded data. *Kybernetika*, 34(4):363–368, 1998.

[51] Shaul Oron, Aharon Bar-Hillel, Dan Levi, e Shai Avidan. Locally orderless tracking. Em *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, páginas 1940–1947. IEEE, 2012.

[52] Jun S Liu e Rong Chen. Sequential monte carlo methods for dynamic systems. *Journal of the American statistical association*, 93(443):1032–1044, 1998.

[53] Yossi Rubner, Carlo Tomasi, e Leonidas J Guibas. The earth mover's distance as a metric for image retrieval. *International journal of computer vision*, 40(2):99–121, 2000.

[54] Arnold WM Smeulders, Dung M Chu, Rita Cucchiara, Simone Calderara, Afshin Dehghan, e Mubarak Shah. Visual tracking: An experimental survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(7):1442–1468, 2014.

[55] George Bebis e Michael Georgiopoulos. Feed-forward neural networks. *Potentials, IEEE*, 13(4):27–31, 1994.

[56] Xinyu Wu, Guoyuan Liang, Ka Keung Lee, e Yangsheng Xu. Crowd density estimation using texture analysis and learning. Em *Robotics and Biomimetics, 2006. ROBIO'06. IEEE International Conference on*, páginas 214–219. IEEE, 2006.

[57] Robert M Haralick, Karthikeyan Shanmugam, e Its' Hak Dinstein. Textural features for image classification. *Systems, Man and Cybernetics, IEEE Transactions on*, (6):610–621, 1973.

[58] Antoni B Chan, Z-SJ Liang, e Nuno Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. Em *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, páginas 1–7. IEEE, 2008.

[59] Antoni B Chan e Nuno Vasconcelos. Modeling, clustering, and segmenting video with mixtures of dynamic textures. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(5):909–926, 2008.

[60] Jens Rittscher, Peter H Tu, e Nils Krahnstoever. Simultaneous estimation of segmentation and shape. Em *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, páginas 486–493. IEEE, 2005.

[61] Todd K Moon. The expectation-maximization algorithm. *Signal processing magazine, IEEE*, 13(6):47–60, 1996.

[62] Christopher Rasmussen e Gregory D. Hager. Probabilistic data association methods for tracking complex visual objects. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(6):560–576, 2001.

[63] Vincent Rabaud e Serge Belongie. Counting crowded moving objects. Em *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, páginas 705–711. IEEE, 2006.

[64] Carlo Tomasi e Takeo Kanade. *Detection and tracking of point features*. School of Computer Science, Carnegie Mellon Univ. Pittsburgh, 1991.

[65] Gabriel J Brostow e Roberto Cipolla. Unsupervised bayesian detection of independent motion in crowds. Em *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, páginas 594–601. IEEE, 2006.

[66] David A Binder. Bayesian cluster analysis. *Biometrika*, 65(1):31–38, 1978.

[67] Edward Rosten e Tom Drummond. Machine learning for high-speed corner detection. Em *Computer Vision–ECCV 2006*, páginas 430–443. Springer, 2006.

[68] Zhong Zhang, Weihong Yin, e Péter L Venetianer. Fast crowd density estimation in surveillance videos without training. Em *Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on*, páginas 452–457. IEEE, 2012.

[69] Hamed Kiani Galoogahi. Tracking groups of people in presence of occlusion. Em *Image and Video Technology (PSIVT), 2010 Fourth Pacific-Rim Symposium on*, páginas 438–443. IEEE, 2010.

[70] Si Wu, Hau-San Wong, e Zhiwen Yu. A bayesian model for crowd escape behavior detection. 2014.

[71] Sofia Zaidenberg, Bernard Boulay, Carolina Garate, Duc Phu Chau, Etienne Corvée, e Francois Bremond. Group interaction and group tracking for video-surveillance in underground railway stations. Em *International Workshop on Behaviour Analysis and Video Understanding (ICVS 2011)*, página 10, 2011.

[72] Carolina Gárate, Sofia Zaidenberg, Julien Badie, Francois Brémond, et al. Group tracking and behavior recognition in long video surveillance sequences. 2014.

[73] Carolina Garate, Sofia Zaidenberg, Julien Badie, e François Bremond. Group tracking and behavior recognition in long video surveillance sequences. 2014.

[74] Slawomir Bak, Etienne Corvee, François Bremond, e Monique Thonnat. Multiple-shot human re-identification by mean riemannian covariance grid. Em *Advanced Video and Signal-Based Surveillance (AVSS), 2011 8th IEEE International Conference on*, páginas 179–184. IEEE, 2011.

[75] Loris Bazzani, Marco Cristani, e Vittorio Murino. Decentralized particle filter for joint individual-group tracking. Em *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, páginas 1886–1893. IEEE, 2012.

[76] Tianshi Chen, Thomas B Schon, Henrik Ohlsson, e Lennart Ljung. Decentralized particle filter with arbitrary state decomposition. *Signal Processing, IEEE Transactions on*, 59(2):465–478, 2011.

[77] Samuel S Blackman. Multiple hypothesis tracking for multiple target tracking. *Aerospace and Electronic Systems Magazine, IEEE*, 19(1):5–18, 2004.

[78] Huaili Wang, Desheng Wang, Lisheng Tian, e Pengfei Ding. A new algorithm for group tracking. Em *Radar, 2001 CIE International Conference on, Proceedings*, páginas 1164–1168. IEEE, 2001.

[79] Erwin Taenzer. Tracking multiple targets simultaneously with a phased array radar. *Aerospace and Electronic Systems, IEEE Transactions on*, (5):604–614, 1980.

[80] Zhang Zi-xu, Zhang Wei, e Chen Ming-yan. A new method of tracking group space object. Em *Microwave and Millimeter Wave Circuits and System Technology (MMWCST), 2013 International Workshop on*, páginas 403–406. IEEE, 2013.

[81] Christian Bailer, Alain Pagani, e Didier Stricker. **A user supported tracking framework for interactive video production**. Em *Proceedings of the 10th European Conference on Visual Media Production*, página 2. ACM, 2013.

[82] Christian Bailer, Alain Pagani, e Didier Stricker. A superior tracking approach: Building a strong tracker through fusion. Em *Computer Vision–ECCV 2014*, páginas 170–185. Springer, 2014.

[83] Yue Gao, Rongrong Ji, Longfei Zhang, e Alexander Hauptmann. Symbiotic tracker ensemble towards a unified tracking framework. 2014.

[84] Junseok Kwon e Kyoung Mu Lee. Tracking by sampling trackers. Em *Computer Vision (ICCV), 2011 IEEE International Conference on*, páginas 1195–1202. IEEE, 2011.

[85] Luka Cehovin, Matej Kristan, e Ales Leonardis. An adaptive coupled-layer visual model for robust visual tracking. Em *Computer Vision (ICCV), 2011 IEEE International Conference on*, páginas 1363–1370. IEEE, 2011.

[86] Zdenek Kalal, Jiri Matas, e Krystian Mikolajczyk. Pn learning: Bootstrapping binary classifiers by structural constraints. Em *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, páginas 49–56. IEEE, 2010.

[87] Mustafa Ozuysal, Pascal Fua, e Vincent Lepetit. Fast keypoint recognition in ten lines of code. Em *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, páginas 1–8. Ieee, 2007.

[88] Caviar dataset. http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/. Accessed: 2015-02-08.

[89] Biwi walking pedestrians dataset. http://www.vision.ee.ethz.ch/datasets/index.en.html. Accessed: 2015-02-09.

[90] Stefano Pellegrini, Andreas Ess, Konrad Schindler, e Luc Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. Em *Computer Vision, 2009 IEEE 12th International Conference on*, páginas 261–268. IEEE, 2009.

[91] Friends meet dataset. http://www.iit.it/en/datasets-and-code/datasets/fmdataset.html. Accessed: 2015-02-09.

[92] Tim Ellis. Performance metrics and methods for tracking in surveillance. Em *Proceedings of the 3rd IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETSâĂŹ02)*, páginas 26–31, 2002.

[93] Bernardin Keni e Stiefelhagen Rainer. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008, 2008.

[94] Rania Y Khalaf Stephen S Intille. Improving multiple people tracking using temporal consistency. *Massachusetts Institute of Technology, Cambridge, MA, MIT Dept, of Architecture House_n Project Technical Report*, 2001.

[95] Kevin Smith, Daniel Gatica-Perez, Jean-Marc Odobez, e Sileye Ba. Evaluating multi-object tracking. Em *Computer Vision and Pattern Recognition-Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on*, páginas 36–36. IEEE, 2005.

[96] Fei Yin, Dimitrios Makris, e Sergio A Velastin. Performance evaluation of object tracking algorithms. Em *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, Rio De Janeiro, Brazil*, 2007.

[97] Thomas Schlögl, Csaba Beleznai, Martin Winter, e Horst Bischof. Performance evaluation metrics for motion detection and tracking. Em *ICPR (4)*, páginas 519–522, 2004.

[98] Marco Cristani, Loris Bazzani, Giulia Paggetti, Andrea Fossati, Diego Tosato, Alessio Del Bue, Gloria Menegaz, e Vittorio Murino. Social interaction discovery by statistical analysis of f-formations. Em *BMVC*, volume 2, página 4, 2011.