Universidade da Beira Interior
Departamento de Informática

Nº 39 - 2014: *Biometric Recognition in Surveillance Scenarios: Subject Detection*

Elaborado por:

João Pais

Orientador:

Professor Doutor Hugo Proença

23 de Junho de 2014

Agradecimentos

A elaboração deste trabalho só foi possível com a ajuda de todas as pessoas que, ao longo da minha vida académica, contribuíram para que fosse possível chegar até aqui. Em primeiro lugar quero agradecer ao meu Orientador, Professor Doutor Hugo Proença, cuja a experiência e conhecimento contribuíram significativamente para o enriquecimento das minhas capacidades. Além do mais, quero agradecer a disponibilidade e auxilio prestados, pois foram muito valiosos. Quero também deixar um agradecimento a todos os colegas do Soft Computing and Image Analysis Laboratory (SOCIA LAB), e em especial ao João Neves pelo apoio e conhecimento prestado durante todo o desenvolvimento deste projeto. Um agradecimento muito especial para a minha namorada Maria pelo enorme apoio e compreensão nos momentos mais difíceis, assim como à minha família nesses mesmo momentos. Por ultimo um obrigado a todos os amigos e colegas que contribuíram diretamente ou indiretamente para o desenvolvimento deste trabalho.

Conteúdo

Conteúdo **iii**

Lista de Figuras **v**

Lista de Tabelas **vii**

Glossário **xi**

1 Introdução **1**
1.1 Enquadramento 1
1.2 Motivação 3
1.3 Objetivos 3
1.4 Dificuldades Inerentes à Deteção 4
1.5 Organização do Documento 6

2 Estado da Arte **7**
2.1 Introdução 7
2.2 Background Subtration 7
2.2.1 Self-Organizing Approach to Background Subtraction . . 9
2.3 Cascatas de Classificadores baseados em características *Haar* . . . 10
2.4 CamShift 11
2.5 Conclusões 12

3 Tecnologias e Ferramentas Utilizadas **13**
3.1 Introdução 13
3.2 Plataforma de Desenvolvimento 13
3.3 Linguagem C++ 13
3.4 Biblioteca OpenCV 14
3.5 Conclusões 14

4 Implementação **15**
4.1 Introdução 15
4.2 Background Subtraction 15

4.3 Conversão Binária 16
4.4 Pré-processamento 17
4.4.1 Erosão 17
4.4.2 Dilatação 18
4.4.3 *Closing* 18
4.4.4 Filtro *Gaussian Blur* 19
4.4.5 Detetor de Arestas 20
4.5 Deteção 23
4.5.1 Componentes Ligados 24
4.5.2 Extração de Características 25
4.5.3 Parametrização 28
4.6 Tracking 29
4.7 Conclusões 29

5 Resultados e Discussão **31**
5.1 Introdução 31
5.2 Cena de Teste 31
5.3 Características dos Vídeos de Teste 32
5.4 Testes e Discussão 33
5.4.1 Metodologia do Teste 33
5.4.2 Parâmetros Base 33
5.4.3 Resultados Base 34
5.4.4 Variação do Intervalo de Deteção 36
5.4.5 Variação do Nº. de Erosões e Dilatações 38
5.4.6 Variação da Área 40
5.4.7 Variação do Ângulo 45
5.4.8 Variação do Aspect ratio 47
5.4.9 Reflexão Crítica 49
5.5 Conclusões 49

6 Conclusões e Trabalho Futuro **51**
6.1 Conclusões Principais 51
6.2 Trabalho Futuro 52
6.2.1 Melhoramento do Background Subtration 52
6.2.2 Nova Característica 53
6.2.3 Algoritmo Genético 53
6.2.4 Maior Base de Dados de Vídeos de Teste 54
6.2.5 Implementar Outro Método de Tracking 54

Bibliografia **55**

Lista de Figuras

1.1 Esquema do projeto *(Biometric Recognition in Surveillance Scenarios)* 2
1.2 Ruído no Background Subtraction. 4
1.3 Sombras no Background Subtraction. 5

2.1 Esquema genérico de funcionamento do BS. 7
2.2 Representação esquemática do modelo da rede neuronal utilizado no SOBS. 9
2.3 Kernels utilizados para cálculo características *Haar* 10

4.1 Background Subtration. 16
4.2 Representação da função de *Threshold* binário. 16
4.3 Exemplo da operação erosão. 17
4.4 Exemplo da operação de dilatação. 18
4.5 Exemplo da operação Close. 18
4.6 Exemplo da operação *Gaussian Blur*. 19
4.7 Esquema de representação da Supressão. 21
4.8 Representação dos thresholds da função *Canny*. 22
4.9 Detetor de Arestas 22
4.10 Modelo do algoritmo 23
4.11 Deteção de Componentes Ligados. 24
4.12 Extração de características. 25
4.13 Frame final resultante da deteção. 26
4.14 Frame com sobreposição de duas pessoas. 27
4.15 Frame final com com o tracking implementado. 29

5.1 Exemplos de situações numa cena. 31
5.2 Gráficos com resultados utilizando a configuração base. 34
5.3 Exemplo de uma frame do vídeo *V12r*. 34
5.4 Exemplo de uma frame do vídeo *V17r*. 35
5.5 Gráfico linear com a variação do parâmetro FRAME_DETECT. . 36

5.6 Gráfico linear com a variação média do parâmetro FRAME_DETECT. 37
5.7 Gráfico linear com a variação dos parâmetros (PRE_N_ERODE , PRE_N_DILATE). 38
5.8 Gráfico linear com a variação média dos parâmetros (PRE_N_ERODE , PRE_N_DILATE). 39
5.9 Gráfico linear com a variação do parâmetro AREA_MIN. 40
5.10 Gráfico linear com a variação média dos parâmetros AREA_MIN. 41
5.11 Gráfico linear com a variação do parâmetro AREA_MAX. 42
5.12 Gráfico linear com a variação média dos parâmetros AREA_MAX. 43
5.13 Gráficos com resultados utilizando o melhor valor máximo e mínimo. 44
5.14 Gráfico linear com a variação dos parâmetros (ANGLE_MIN , ANGLE_MAX.) 45
5.15 Gráfico linear com a variação média dos parâmetros (ANGLE_MIN , ANGLE_MAX.) 46
5.16 Gráfico linear com a variação dos parâmetros (ASPEC_RACIO_MIN, ASPEC_RACIO_MAX.) 47
5.17 Gráfico linear com a variação média dos parâmetros (ASPEC_RACIO_MIN, ASPEC_RACIO_MAX.) 48

6.1 Exemplo de *output* do Self-Organizing Approach to Background Subtraction (SOBS). 52
6.2 Exemplo de nova característica. 53

Lista de Tabelas

4.1 Parâmetros 28

5.1 Detalhes dos vídeos utilizados. 32
5.2 Configuração base. 33
5.3 Tabela de resultados. 35

Acrónimos

OpenCV	Open Source Computer Vision
I/O	Input e Output
GUI	Graphical User Interface
2D	2 Dimensões
3D	3 Dimensões
IDE	Integrated Development Environment
SOBS	Self-Organizing Approach to Background Subtraction
ROI	Region of Interest
SOCIA LAB	Soft Computing and Image Analysis Laboratory
MoG	Mixture of Gaussians
BS	Background Subtration
SOM	Self-Organizing Maps
RGB	Red, Green, Blue

Glossário

aspect ratio Relação entre a largura e a altura de um retângulo.. 11, 25, 29

dataset Conjunto de dados para teste.. 8, 15

falso positivo Deteção errónea de um objeto.. 35

frame É um termo que define uma única imagem de um vídeo num determinado instante.. v, 3–5, 8, 9, 15–20, 22, 24–27, 29, 31, 33–37

framerate Número de frame por segundo.. 37

histograma Distribuição de frequências de um ou mais canais de cor numa imagem.. 11

kernel Em termos de processamento de imagem, um *kernel*, matriz de convolução ou máscara é uma pequena matriz utilizada em convoluções matemáticas.. v, 10

outlier Observações que apresentam um grande afastamento das restantes ou são inconsistentes com as mesmas.. 34

State of Art Significa o estado da arte, isto é, a situação atual em que se encontra determinada área.. 14

threshold É um termo aplicado para definir um limite, ou seja, por exemplo, uma determinada fronteira de valores. . v, 21, 22

tracking É uma técnica que consiste em seguir determinado objeto numa sequencia de imagens.. iv, v, 1, 11, 14, 15, 29

Capítulo 1

Introdução

1.1 Enquadramento

Este projeto está enquadrado no *Biometric Recognition in Surveillance Scenarios*1.1 do SOCIA LAB cujo o objetivo é o reconhecimento de pessoas através de um sistema de vídeo vigilância, pelo que este módulo vai tratar apenas da deteção de pessoas recorrendo apenas ao Background Subtration (BS) 2.2. No ramo dos sistemas biométricos, a deteção de pessoas é uma tarefa fundamental para etapas posteriores, tais como o tracking e identificação.

Obviamente, trata-se de uma área de elevada dificuldade, uma vez que vai ao encontro de tentar imitar a capacidade humana de processar imagens, algo que é simples para uma pessoa fazer, mas extremamente complexo no ponto de vista de um sistema de informação. Pelo que, neste trabalho são também apresentadas quais as dificuldades e circunstâncias que tornam a deteção uma tarefa tão difícil.

As potencialidades desta área de investigação são elevadíssimas, e suscitam grande interesse dos sectores da indústria, segurança e até mesmo o sector militar. Na imagem 1.1(sombreado a verde) é possível verificar precisamente onde se enquadra este trabalho relativamente ao *Biometric Recognition in Surveillance Scenarios*.

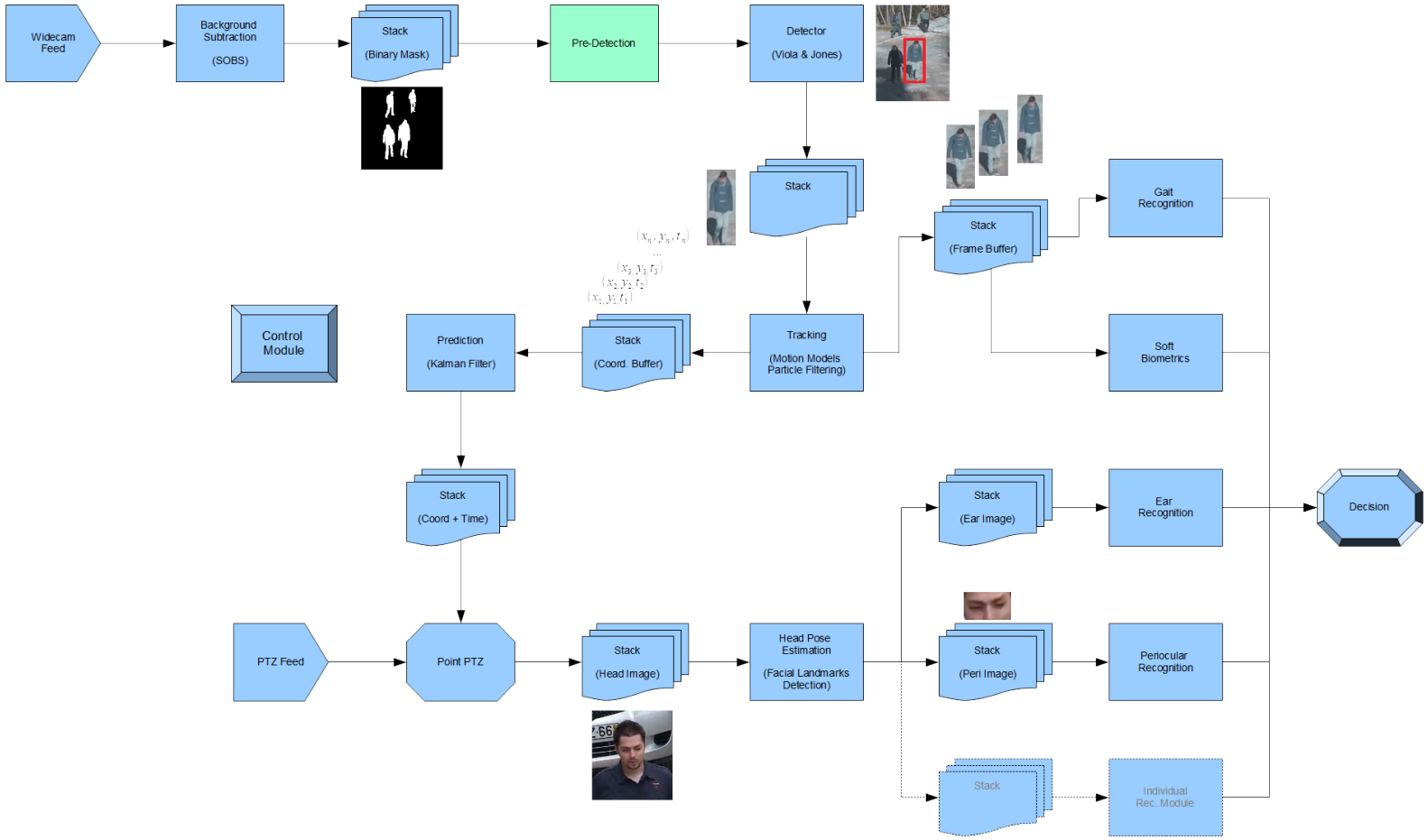


Figura 1.1: Esquema do projeto *(Biometric Recognition in Surveillance Scenarios)*

1.2 Motivação

O reconhecimento e identificação de pessoas através de sistema de vídeo vigilância é uma área de bastante interesse e elevada importância na sociedade atual, o facto de poder oferecer uma maior segurança recorrendo a tecnologia permitirá eliminar o erro humano da equação, é algo que ser Humano tenta desenvolver desde os primórdios da era digital. A possibilidade de detetar e identificar pessoas em imagens de vídeo acarreta um enorme potencial de aplicações, tais como a procura de pessoas desaparecidas ou até mesmo o controlo de ações suspeitas de indivíduos. Além do mais, as ciências da visão computacional exercem grande fascínio, não só na perceção do estímulo luminoso, mas também, no reconhecimento lógico de objetos circundantes. No caso do mundo natural, a visão é utilizada para inúmeras tarefas, que na maior parte dos caso está relacionada com a sobrevivência das espécies, como por exemplo, o reconhecimento de padrões. No que diz respeito à visão artificial, os processos e métodos automatizados de reconhecimento de padrões requerem eficácia, eficiência e adaptabilidade, algo que a crescente aumento da velocidade do poder de processamento dos computadores está a permitir. A crescente preocupação por parte da sociedade no campo da segurança ajuda a impulsionar esta área de investigação, deste modo, a comunidade científica tem publicado um grande número de trabalhos e artigos nesta área de investigação.

1.3 Objetivos

Neste trabalho é proposto um método preliminar de deteção de pessoas através de operações simples e eficientes, neste caso recorrendo apenas ao *output* gerado pelo método de BS para realizar a extração de características e consequente deteção, fornecendo deste modo uma área de interesse ou Region of Interest (ROI) para poupar tempo de processamento a métodos mais complexos que venham a ser utilizados numa fase posterior sobre essa mesma ROI, tendo em conta que o método visa ser utilizado em sistemas de tempo real. Este trabalho permite realizar um estudo das ciências de visão computacional, tendo como principal objetivo a deteção e localização de um objeto do tipo "pessoa"numa dada frame de um vídeo. A solução implementada terá que ser especialmente resistente a falsos positivos, isto é, evitar deteções erradas de objetos que não são de facto pessoas, além disso terá que apresentar resistência ao ruído gerado pelo cenário em si. Algo que é extremamente complicado e é abordado na secção seguinte.

1.4 Dificuldades Inerentes à Deteção

Devido à elevada diversidade de ambientes e à enorme quantidade de variáveis no mundo real a deteção é uma operação extremamente difícil, uma vez que é necessário ter em conta vários parâmetros que influenciam significativamente os resultados obtidos.

Um dos problemas, no que diz respeito á forma dos objetos, é o facto de a imagem apenas conter informação 2D de um ambiente 3D, como consequência é perdida informação, pelo que é preciso ter em conta que a forma do objeto apesar de não variar no mundo real vai variar bastante na imagem 2D, o facto de para todos os efeitos, uma pessoa ser um objeto deformável neste contexto, evita o uso de ferramentas que caso contrário seriam bastante úteis, como por exemplo, os momentos de Hu[6] que são extremamente eficazes na deteção de objetos não deformáveis.

No entanto, o cenário é que apresenta as maiores dificuldades, uma vez que, a variação de luminosidade e o pequenos movimentos de objetos estáticos como a influência do vento em árvores, influenciam bastante negativamente os resultados. As alterações de luminosidade provocam súbitas alterações no contraste devido a sombras, uma simples nuvem passageira pode provocar uma alteração de luminosidade enorme, como pode ser verificado na imagem 1.2.

(a) Frame original. (b) Output do método SOBS.

Figura 1.2: Ruído no Background Subtraction.

Um dos piores obstáculos é sem dúvida a presença de vários objetos no mesmo local, no que diz respeito ao BS, mesmo para o olho humano, é difícil distinguir a presença de várias pessoas num cenário quando estas se encontram próximas. Além disso, a presença de sombras das pessoas implica que o objeto presente no BS contem também essa mesma sombra anexada ao objeto, e como vai ser verificado mais à frente desde documento 2.2, o método selecionado contém alguma fraqueza neste campo e irá ter grande impacto nos resultados obtidos.

(a) Frame original. (b) Output do método SOBS.

Figura 1.3: Sombras no Background Subtraction.

Como se verifica na imagem 1.3b, o método de BS não foi capaz de tratar a sombra dos objetos, neste caso a sombras são bastante evidentes, assumindo a forma perfeita da pessoa. A sombra da pessoa à direita está sobreposta à pessoa da esquerda, tornando-se evidente, com este exemplo, das inúmeras dificuldades inerentes à deteção de pessoas.

O ideal seria desenvolver um método de BS em que todos estes fatores não influenciassem a deteção de qualquer objeto presente em qualquer cenário, sendo resistente a todas as variáveis inerentes ao mundo natural. Contudo, julgo que tal método ainda se encontra longe em termos de investigação, principal devido ao enorme número de variáveis num cenário e as suas quase infinitas configurações possíveis.

1.5 Organização do Documento

De modo a refletir o trabalho que foi feito, este documento encontra-se estruturado da seguinte forma:

1. O primeiro capítulo – **Introdução** – apresenta o projeto, a motivação para a sua escolha, o enquadramento para o mesmo, os seus objetivos e a respetiva organização do documento.

2. O segundo capítulo – **Estado da Arte** – contém trabalhos e técnicas existentes que se enquadram no domínio deste projeto.

3. O terceiro capítulo – **Tecnologias Utilizadas** – descreve as tecnologias utilizadas durante do desenvolvimento da aplicação.

4. O quarto capítulo – **Implementação** – descreve os conceitos mais importantes no âmbito deste projeto, bem como as tecnologias utilizadas durante do desenvolvimento da aplicação.

5. O quinto capítulo – **Resultados e Discussão** – contém os testes resultantes do método implementado assim como os resultados referentes ao mesmo.

6. O sexto capítulo – **Conclusões e Trabalho Futuro** – como o nome sugere, pretende-se inferir algumas conclusões gerais assim como possível trabalho que poderá melhorar os resultados.

Capítulo 2

Estado da Arte

2.1 Introdução

Neste capítulo é apresentado um estudo sobre as soluções existentes atualmente, assim como técnicas essenciais para desenvolver soluções semelhantes de deteção.

2.2 Background Subtration

O BS é a base do processamento e análise de vídeo e providencia suporte para todas as fases subsequentes. É uma técnica que consiste em separar o fundo de uma cena dos objetos em movimento na mesma (ver imagem 2.1), nomeadamente em vídeo, e é amplamente utilizado em sistemas em que a câmera está numa posição fixa. Existem inúmeros métodos, entre os quais o *Frame Difference*[7], que se trata de um dos métodos mais simples, até aos mais complexos como o Mixture of Gaussians (MoG)[16].

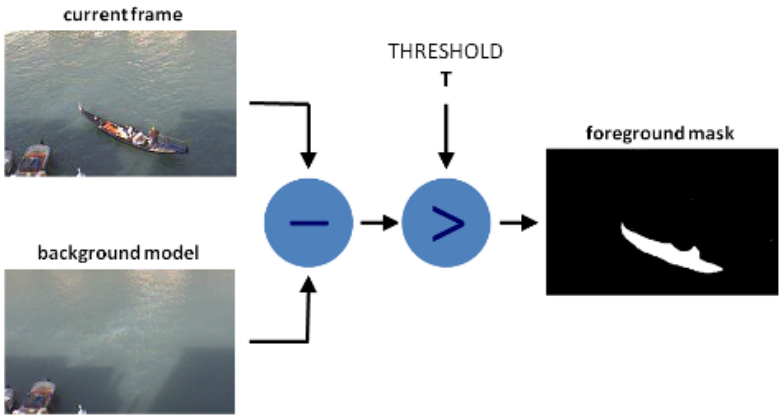


Figura 2.1: Esquema genérico de funcionamento do BS.

Este trabalho depende fortemente da qualidade do BS, pelo que este merece especial atenção. Um BS robusto tem que ser resistente a variações de luz, movimento cíclico e movimentos lentos de longa duração, este assenta na ideia de que cada frame pode ser comparada a um modelo de *background*, como pode ser verificado na imagem 2.1, a estratégia mais comum para construir um modelo de *background* é inferir o mesmo a partir das ultimas N frames. O modelo mais simples de BS é o *Frame Difference* [8], este consiste em usar frame imediatamente anterior como modelo de *background*. Por outro lado, os modelos *Adaptive Median*[14] e *Temporal Median* [5] obtêm o modelo de BS por análise estatística das ultimas N frames.

Avançando para modelos mais complexos, os métodos baseados em modelos gaussianos assumem que a crominância do *background* está normalmente distribuída. Este é o caso do método *Simple Gaussian* [22], que usa uma única distribuição de *Gauss* para modelar as intensidades de cada pixel, por outro lado, o método Mixture of Gaussians usa não uma, mas várias distribuições gaussianas, deste modo é possível tratar várias fontes de *backgrounds* permitindo que este método consiga tratar cenários complexos e dinâmicos. Contudo este modelo apresenta dificuldades quando confrontado com objetos que se movem lentamente. Uma tentativa de resolver este problema foi proposta por Zivkovic, Z.[23], que consistiu em adaptar dinamicamente o número de distribuições de Gauss por pixel. Outro modelo que apresenta bons resultados é o método *Eigenbakground* [15], muito resumidamente, este tira partido das relações entre os pixeis.

Todos apresentam vantagens e limitações, o processo de escolha de um método em especifico assenta sobre um estudo realizado pelo SOCIA LAB em que foi feita uma comparação dos métodos mais populares para o cenário em questão e datasets públicos, foi concluído que neste caso o mais fiável seria o SOBS [12], que é explicado na secção seguinte.

2.2.1 Self-Organizing Approach to Background Subtraction

O método SOBS foi proposto por Maddalena e Petrosino [12], a ideia é construir um modelo de *background* recorrendo a métodos de auto-organização. Com base no modelo construído, o algoritmo é capaz de atualizar seletivamente o modelo recorrendo a uma rede neuronal. A rede neuronal utilizada é organizada numa matriz de 2 Dimensões (2D) de neurónios, semelhante às redes Self-Organizing Maps (SOM) ou Kohonen Networks [9]. Estes mapas são constituídos por uma matriz de neurónios e permitem organizar topologicamente um conjunto de características em grupos, como também a representação de amostras de treino com menor dimensionalidade. Cada neurónio contem um vetor de características com o valor típico do grupo. No método SOBS, cada pixel é modelado por um SOM e treinado com os valores de Red, Green, Blue (RGB) das primeiras N frames. Deste modo, é possível adquirir os valores típicos das intensidades do background naquele ponto da cena, obtendo assim um modelo de background. A figura 2.2 exemplifica a estrutura utilizada pelo método SOBS numa região de 6 pixeis.

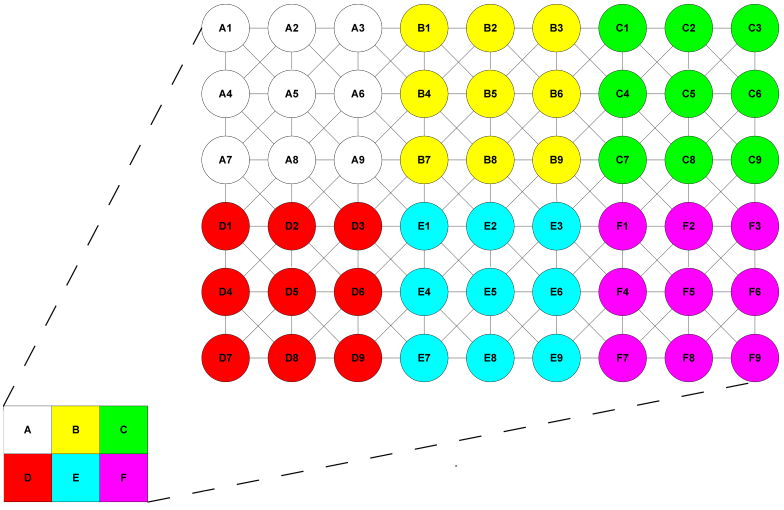


Figura 2.2: Representação esquemática do modelo da rede neuronal utilizado no SOBS.

Embora o método SOBS afirme que o mesmo tem em conta a remoção de sombras dos objetos, os resultados obtidos utilizando o nosso cenário não corroboram esta tese, de facto este só é eficaz em situações próximas de um cenário perfeito.

2.3 Cascatas de Classificadores baseados em características *Haar*

Em 2001, Paul Viola e Michael Jones desenvolveram um detetor de objetos baseado em características *Haar* e cascatas de classificadores [18]. Cada característica *Haar* mede a a diferença de intensidades entre duas ou mais regiões. A figura 2.3 ilustra um conjunto kernels usados para calcular várias características *Haar*. A ideia é fazer a convulsão deste kernel com todas as posições da imagem para assim determinar em que regiões esta característica está presente. No caso da deteção da face humana, um kernel horizontal, como representado na imagem 2.3 (1-b), seria útil para determinar se a região periocular está presente, uma vez que a região dos olhos é mais escura que a região inferior.

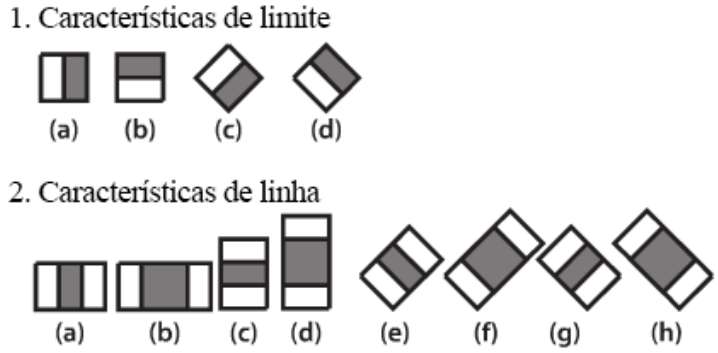


Figura 2.3: Kernels utilizados para cálculo características *Haar*

Para determinar quais as melhores características para detetar um determinado objeto, os autores propuseram usar um conjunto de treino, contendo um vasto número de exemplos do objeto de interesse e de exemplos onde o objeto não está presente. Desta forma, é possível determinar que características alcançam uma melhor separação entre objetos e não-objetos (classificadores fortes). Para além disso, a fase de treino é também usada para encontrar características que isoladas não apresentam uma discriminabilidade elevada (classificadores fracos), mas que são especializadas numa propriedade do objeto. De modo a aumentar eficiência temporal deste método, estes classificadores são dispostos em cascata, ou seja, a cada sub-região da imagem são aplicados, primeiramente, os classificadores fortes, sendo a utilização dos restantes dependente da resposta afirmativa dos classificadores fortes, deste modo, apenas as regiões mais prováveis serão examinadas exaustivamente. Dado a sua eficiência e rapidez, este método é tipicamente usado para deteção de faces frontais. Para além disso, a utilização desta abordagem é também popular para deteção de diferentes partes corporais.

2.4 CamShift

O *CamShift* é um método de tracking proposto por Bradski [2], em 2008, que é um melhoramento ao método original proposto por Comaniciu *et* al. [4], em 2003. Este método é relativamente simples e consiste em seguir objetos com base na cor, e é utilizado como base de outros métodos de tracking mais avançados. Na prática, o *CamShift* calcula uma projeção probabilística da distribuição dos gradientes do histograma de um dado objeto para encontrar o pico mais próximo dentro de uma determinada janela de procura. Deste modo, a localização média do objeto alvo do tracking é calculada utilizando os momentos de ordem 0, 1 e 2 da imagem:

$$M_{00} = \sum_x \sum_y P(x, y), \tag{2.1}$$

$$M_{10} = \sum_x \sum_y xP(x, y); M_{01} = \sum_x \sum_y yP(x, y), \tag{2.2}$$

$$M_{20} = \sum_x \sum_y x^2P(x, y); M_{02} = \sum_x \sum_y y^2P(x, y), \tag{2.3}$$

onde $P(x, y) = h(I(x, y))$ é a projeção da distribuição de probabilidade na posição x,y na janela de procura $I(x, y)$, que é computada a partir do histograma h de I. A posição média do objeto alvo pode então ser calculada com

$$x_c = \frac{M_{10}}{M_{00}}; y_c = \frac{M_{01}}{M_{00}}, \tag{2.4}$$

enquanto que o seu aspect ratio,

$$ratio = \frac{M_{20}}{x_c^2} \Big/ \frac{M_{02}}{y_c^2}\ , \tag{2.5}$$

é utilizado para atualizar a janela de procura com,

$$largura = \sqrt{2M_{00}}.ratio; altura = \sqrt{2M_{00}}/ratio. \tag{2.6}$$

A posição e as dimensões da janela de procura são atualizadas a cada iteração até convergirem. O principal problema do *CAMShift* é que não é capaz de distinguir objetos com cores semelhantes, isto porque a projeção da distribuição de probabilidade apenas considera a cor. Este também não é eficaz em alterações de luminosidade da cena e a oclusões parciais ou totais dos objetos.

2.5 Conclusões

É evidente que esta é uma área de investigação de bastante interesse no mundo académico e não só, muito trabalho já foi realizado mas é uma área sempre aberta a melhoramentos e até mesmo a novos métodos. No capítulo seguinte são apresentadas as ferramentas e tecnologias utilizadas assim como justificações para as escolhas realizadas.

Capítulo 3

Tecnologias e Ferramentas Utilizadas

3.1 Introdução

Este capitulo esclarece quais as ferramentas utilizadas, assim como justificações para o uso das mesmas em deterioramento de outras semelhantes.

3.2 Plataforma de Desenvolvimento

A escolha do Integrated Development Environment (IDE) *Mirosoft Visual Studio 2013* deve-se ao excelente modo de *debug* que possuí, permitindo um nível de detalhe da informação em tempo real de *debugging*, é extremamente útil devido à necessidade de utilizar uma biblioteca externa 3.4 referenciada mais à frente. Possui ainda um controlo de versões integrado no próprio IDE, o *Team Foundation Server*, que permite uma excelente gestão de *backups*.

3.3 Linguagem C++

A linguagem *C++* [19] é uma linguagem baseada em *C* [20], com a particularidade de ser orientada a objetos e possui recurso de baixo e de alto nível. A escolha da linguagem de programação *C++* deveu-se a dois fatores, o primeiro foi o facto de existir uma biblioteca 3.4 desenvolvida na mesma linguagem e que possui todas as ferramentas necessárias para o desenvolvimento do projeto. Por

outro lado, o segundo fator é o facto de o projeto ser uma componente de um projeto de maior escala atualmente a ser desenvolvido pelo SOCIA LAB, como tal esta componente terá que ser incorporada nesse mesmo projeto e o mesmo está a ser desenvolvido em *C++*.

3.4 Biblioteca OpenCV

Dado que o projeto envolve processamento de vídeo/imagem, foi necessário recorrer à utilização de uma biblioteca que possibilita o tratamento avançado de imagens. O Open Source Computer Vision (OpenCV) [1] é uma biblioteca multiplataforma desenvolvida em *C/C++* [19], que possui módulos de processamento de imagem e vídeo Input e Output (I/O), estruturas de dados, álgebra linear, Graphical User Interface (GUI), controlo de rato e teclado e mais de 2500 algoritmos otimizados relacionados com a visão computacional. Estes algoritmos permitem por exemplo, a deteção e reconhecimento de faces humanas, identificação de objetos, classificar ações humanas em vídeos,tracking, extração de modelos 3 Dimensões (3D) a partir de objetos e muitas outras funcionalidades, que incluem tanto o State of Art como a clássica visão computacional.

3.5 Conclusões

De um modo geral esta é a melhor combinação de ferramentas necessárias para o desenvolvimento do projeto, visto se tratarem de tecnologias bem documentadas, (isto no caso da biblioteca OpenCV), e devido a todo o suporte que o SOCIA LAB dispensou devido ao projeto de maior escala em que este próprio se encontra. Com base nestas ferramentas, o capítulo seguinte será relativo à implementação dos algoritmos desenvolvidos para tratar o problema da deteção.

Capítulo 4

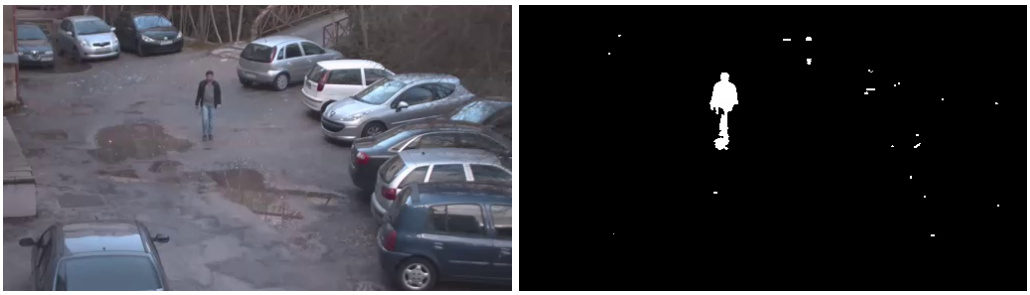
Implementação

4.1 Introdução

Neste capítulo é tratada a fase de implementação da solução apresentada, o mesmo está dividido em fases, desde o *Background Subtraction* até ao tracking passando pelas diversas fases do pré-processamento, de modo a conseguir apresentar uma vista linear e estruturada de todo o processo, e deste modo facilitar a terceiros seguir a linha de pensamento adotada para atingir os objetivos deste trabalho.

4.2 Background Subtraction

Conforme apresentado no capítulo 2.2, o método de *Background Subtraction* foi o SOBS, que segundo um estudo elaborado pelo SOCIA LAB, apresentou os melhores resultados no geral mediante o cenário utilizado e com os datasets públicos mais comuns da comunidade cientifica. O método SOBS requer 100 frames de aprendizagem da cena antes de produzir resultados, durante esse período de aprendizagem o algoritmo constrói um modelo de background que vai ser utilizado para a deteção de objetos nas restantes frames do vídeo. Na imagem 4.1b pode ser visto o *output* de uma determinada frame, esta apresenta ruído proveniente de reflexões e movimento de árvores devido ao vento, assim como algumas descontinuidades no objeto detetado. Uma tentativa de tratar estas imperfeições vai ser abordada na fase de Pré-Processamento na secção 4.4.



(a) Frame original. (b) Output do método SOBS.

Figura 4.1: Background Subtration.

4.3 Conversão Binária

Antes de qualquer outra operação, é necessário converter a frame resultante do *Background Subtraction* numa imagem binária, uma vez que, embora o vídeo resultante do método SOBS devolva uma máscara binária, é posteriormente convertido para formato *MPEG-4*, pelo que a informação binária é perdida devido à codificação vídeo do formato *MPEG-4*. A operação é realizada em 2 passos, inicialmente a frame é convertida de *RGB* para tons de cinza, de seguida é aplicada a função de threshold binário [17] dada por

$$destino(x,y) = \begin{cases} \text{valorMaximo} & \text{Se} fonte(x,y) > T(x,y) \\ 0 & \text{caso contrário,} \end{cases}$$

caso a intensidade do pixel fonte(x,y) for maior que o *threshold*, então a nova intensidade será o valorMaximo, caso contrário a intensidade será 0. Na imagem 4.2 pode ser visualizada uma representação gráfica da função de *threshold*.

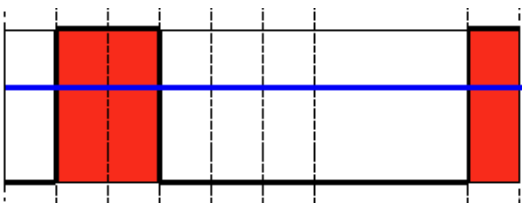


Figura 4.2: Representação da função de *Threshold* binário.

4.4 Pré-processamento

O pré-processamento é um conjunto de operações realizadas sobre uma imagem de modo a melhorar a mesma para atingir melhores resultados nas etapas seguintes, isto é, consiste em operações de remoção de ruído, aumento do contraste, entre outros... o pré-processamento é feito antes de qualquer operação de análise ou deteção.

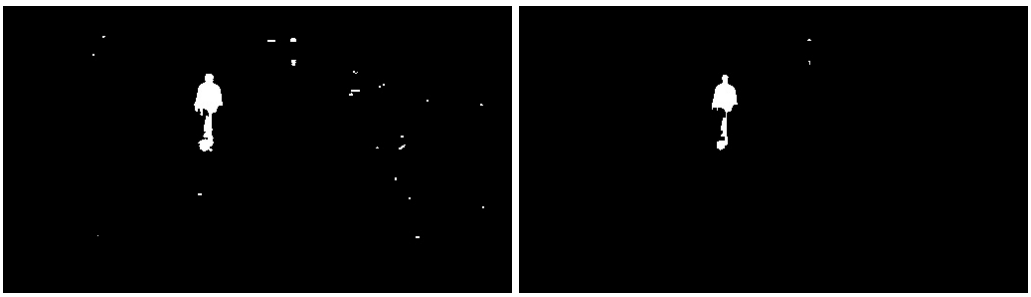
4.4.1 Erosão

A erosão binária, também conhecida como subtração de minkowski [13] no mundo da geometria, é uma operação morfológica de processamento de imagem, normalmente aplicada a imagens binárias. Consiste em analisar uma imagem com uma forma pré-definida e tirar conclusões acerca de como essa forma preenche ou não a imagem, isto é, analisa formas cujo o tamanho seja inferior à forma pré-definida, deste modo é possível retirar da frame detalhes que não são necessários, como ruído causado por reflexões ou um ligeiro movimento devido a vento. A operação de erosão é definida do seguinte modo, seja E um espaço Euclidiano ou uma matriz de inteiros, e A uma imagem binária em E. A erosão dessa mesma imagem A pelo elemento estrutural B é definida por

$$A \ominus B = \{z \in E | B_{\approx} \subseteq A\}, \tag{4.1}$$

onde $B_{\approx}$ é dado por $B_z = b + z | b \in B$.

Nas seguintes imagens é possível verificar o resultado da erosão4.3.



(a) Frame binária. (b) Frame após erosão.

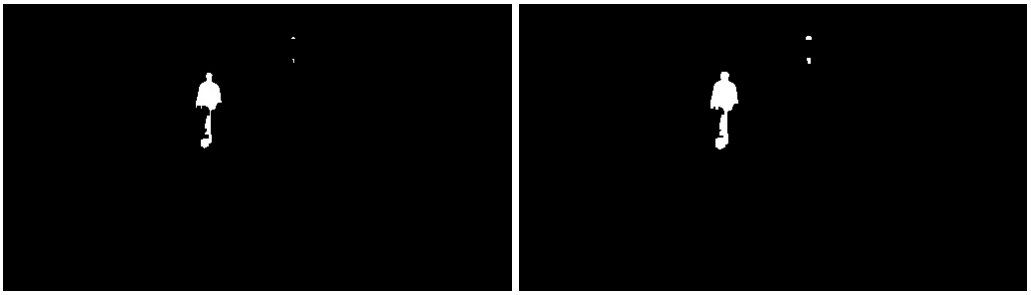
Figura 4.3: Exemplo da operação erosão.

4.4.2 Dilatação

A dilatação é a operação morfológica oposta da erosão 4.4.1, esta consiste essencialmente em analisar a imagem e expandir formas, deste modo e possível recuperar algum detalhe perdido pela operação da erosão, como pode ser verificado na imagem 4.4. A dilatação é definida por:

$$A \oplus B = \{z \in E | (B^s)_{\approx} \cap A \neq \emptyset\}, \tag{4.2}$$

onde B^s denota o simétrico de B, que é dado por $B_z = b + z | b \in B$



(a) Frame após erosão. (b) Frame após dilatação.

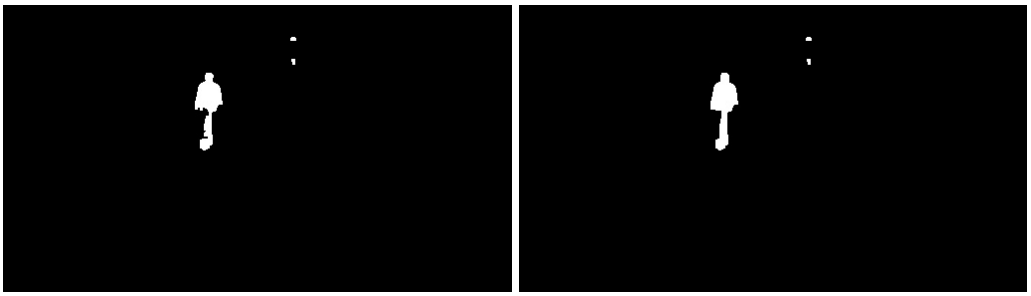
Figura 4.4: Exemplo da operação de dilatação.

4.4.3 *Closing*

Na morfologia de processamento de imagem o *closing* consiste em efetuar uma dilatação 4.4.2 seguida de uma erosão 4.4.1,

$$A \bullet B = (A \oplus B) \ominus B,$$

apesar de parecer contraditório relativamente à ordem das operações realizadas anteriormente, o *closing* permite retirar ruído extra e remove lacunas nas formas que foram perdidas previamente como pode ser visto na imagem 4.5.



(a) Frame após dilatação. (b) Frame após close.

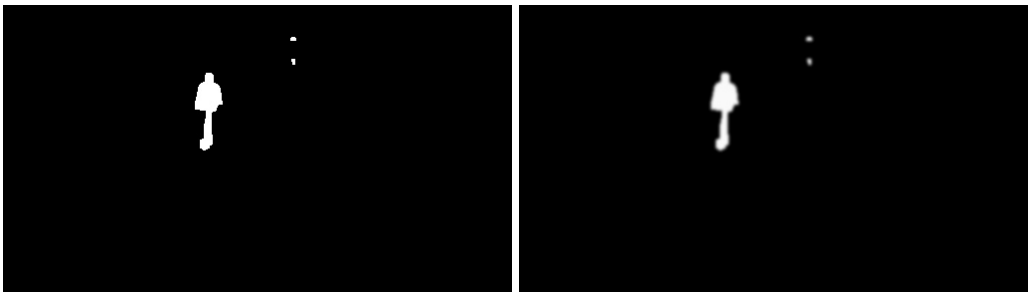
Figura 4.5: Exemplo da operação Close.

4.4.4 Filtro *Gaussian Blur*

O filtro *Gaussian Blur*, também conhecido como *Gaussian smoothing*, é o resultado de desfocar/suavizar da imagem através da função de Gauss, é amplamente utilizada em operações de processamento de imagem, tipicamente para reduzir ruído ou detalhes indesejados. (Em termos comparativos, o resultado da aplicação do filtro é semelhante a visualizar a imagem através de um vidro translucido ou o desfocar da imagem.) Matematicamente, em 2D a função de Gauss é definida pela equação

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{4.3}$$

onde x é a distância da origem no eixo horizontal, y é a distância da origem no eixo vertical e σ é o desvio padrão da distribuição de Gauss. Esta fórmula produz uma superfície cujos contornos são círculos concêntricos, pela distributivo de Gauss, a partir do ponto central, os valores desta distribuição são utilizados para construir uma matriz de convulsão que é aplicada à imagem original, isto é, cada novo pixel vai ter um valor médio relativamente aos seus pixeis vizinhos. Deste modo o filtro ajuda a remover ruído e prepara a frame para a fase seguinte de deteção de arestas 4.4.5.



(a) Frame após o *close*. (b) Frame após aplicação do *Gaussian Blur*.

Figura 4.6: Exemplo da operação *Gaussian Blur*.

4.4.5 Detetor de Arestas

Esta operação têm como objetivo detetar as arestas dos objetos presentes na frame, o algoritmo utilizado é o *Canny Edge Detector* [3]. Este algoritmo apresenta 4 etapas:

Redução de ruído

Esta etapa consiste na remoção de ruído, sendo um requisito para aplicar a função *Canny* disponibilizada pelo OpenCV, daí a aplicação prévia do filtro *Gaussian Blur* explicada no sub-capítulo anterior 4.4.4.

Intensidade do Gradiente

As arestas numa frame podem apontar em diversas direções, daí a utilização de algoritmos de deteção horizontal, vertical e em ambas as diagonais. O operador de deteção de arestas devolve as derivadas horizontal e vertical, podendo assim ser determinado o gradiente de arestas:

$$\mathbf{G} = \sqrt{\mathbf{G_x^2} + \mathbf{G_y^2}} \tag{4.4}$$

encontrar a direção das arestas é trivial uma vez calculados os gradientes segundo x e y

$$\mathbf{\Theta} = \arctan\left(\frac{\mathbf{G_y}}{\mathbf{G_x}}\right). \tag{4.5}$$

A direção do gradiente é sempre perpendicular às arestas, é arredondado a um de quatro ângulos representativos das várias direções possíveis (vertical, horizontal, e ambas as diagonais).

Supressão

A supressão consiste em "limar"as arestas resultantes até ficarem apenas linhas, isto é, são removidos pixeis que não são considerandos como pertencentes a determinada aresta isto faz com que o detalhe/definição das mesmas seja aumentado como se tratasse de uma erosão 4.4.1 das arestas. Para isso, em cada pixel, é verificado se o próprio é um máximo local na vizinhança da direção do gradiente, como pode ser verificado pela seguinte imagem 4.7.

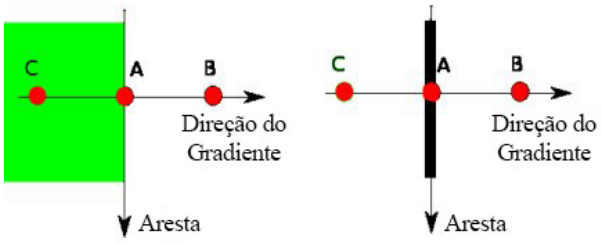


Figura 4.7: Esquema de representação da Supressão.

O ponto A está na aresta vertical, a direção do gradiente é normal à aresta, os pontos B e C encontram-se na direção do gradiente então é verificado se o ponto A é o máximo local entre B e C, em caso afirmativo o pixel é considerado, caso contrario é suprimido. Como foi dito anteriormente esta operação consiste em tornar as arestas mais finas.

Histerese

A função *Canny* usa dois thresholds, um superior e outro inferior, de modo a distinguir quais são as verdadeiras arestas, como se pode verificar em 4.8.

- Caso o gradiente de determinado pixel seja superior que o threshold superior, esse pixel é aceite como pertencendo a uma aresta, como no caso de A.

- Se o gradiente de determinado pixel for inferior ao threshold inferior, o pixel é rejeitado.

- Se o gradiente de determinado pixel estiver contido entre os dois thresholds, o pixel só é aceite se estiver conectado a um pixel cujo o gradiente seja maior que o threshold superior, neste caso C é aceite mas B é rejeitado.

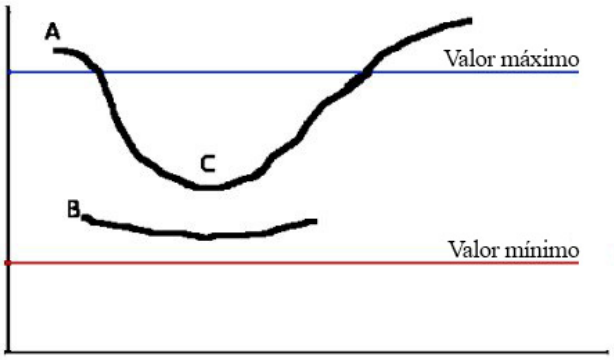
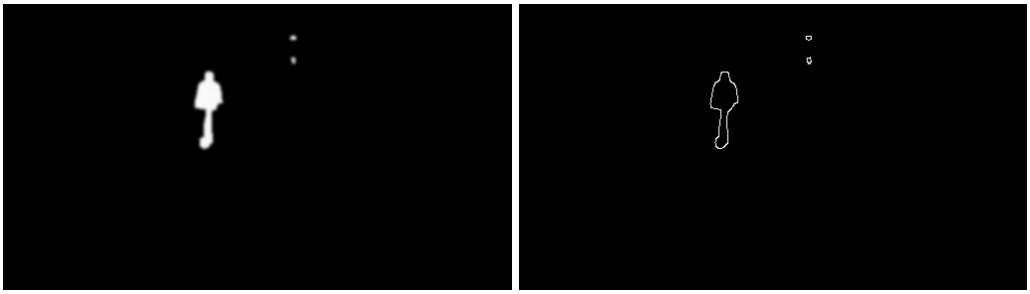


Figura 4.8: Representação dos thresholds da função *Canny*.

Na seguinte imagem 4.9 pode ser verificado o resultado da aplicação da função *Canny* do OpenCV.



(a) Frame após o *filtro gaussiano*.

(b) Frame após a detetor de arestas.

Figura 4.9: Detetor de Arestas

4.5 Deteção

Após a realização de todas as operações de pré-processamento, a frame resultante pode ser analisada para efetuar a deteção de objetos cujas características se enquadrem em classes de objetos do tipo "pessoa". Na seguinte imagem 4.10 pode ser vista a estrutura do algoritmo, esta evidencia todos os processos realizados na fase de pré-processamento assim como as diferentes etapas da deteção.

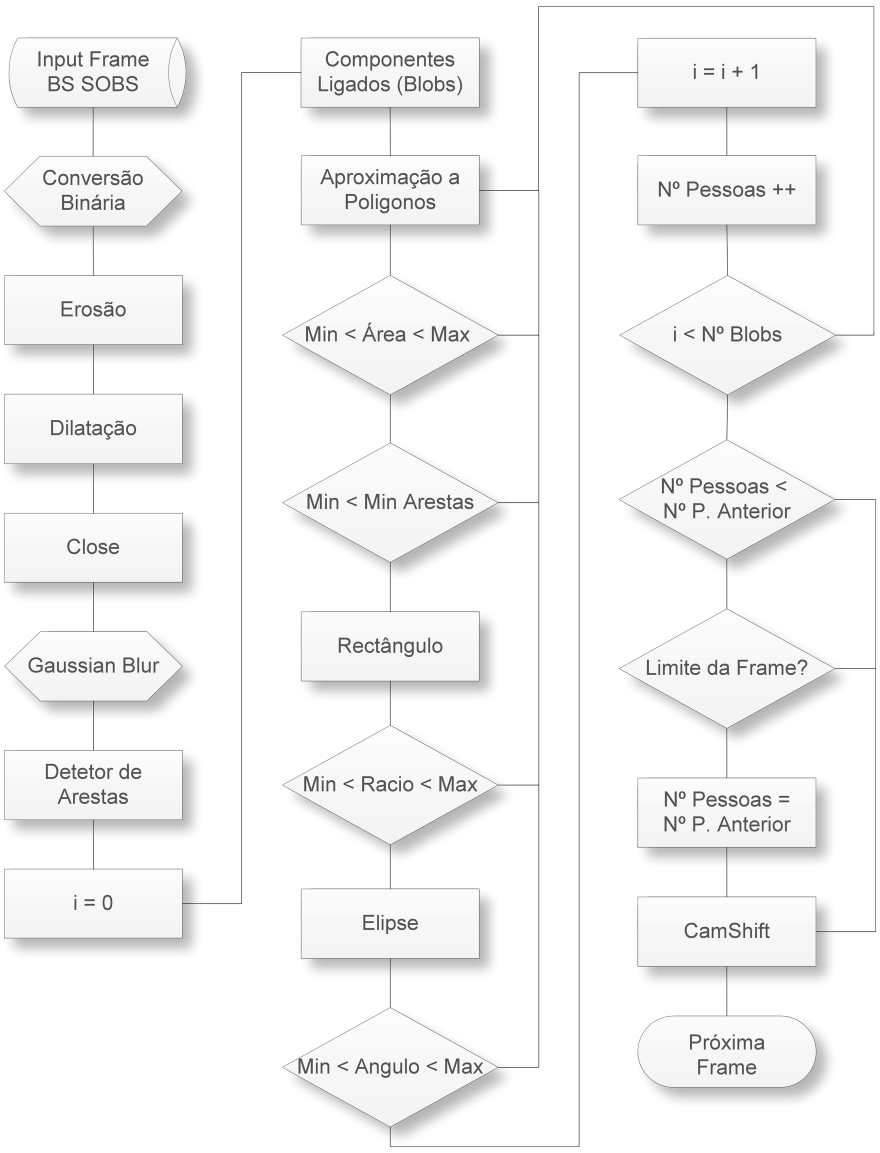
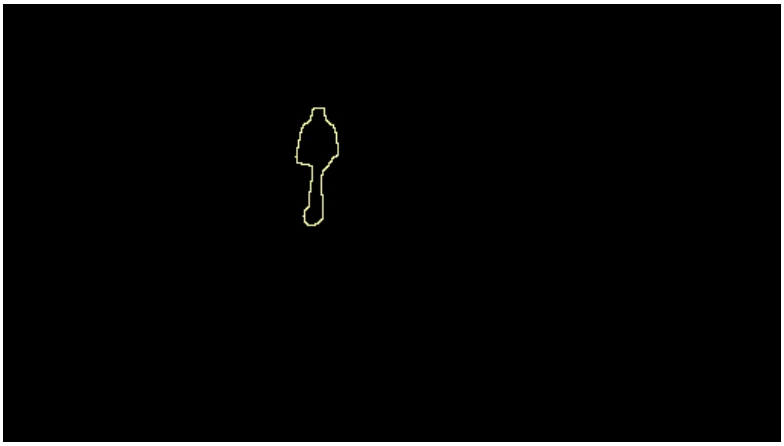


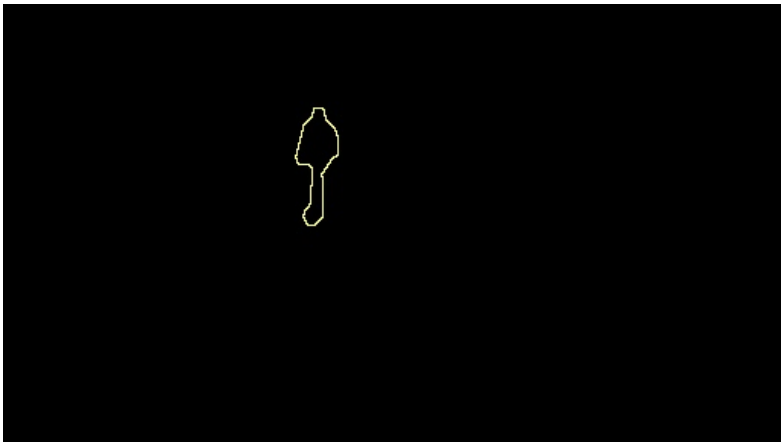
Figura 4.10: Modelo do algoritmo

4.5.1 Componentes Ligados

Esta operação consiste em analisar uma frame e agrupar os seus pixeis em componentes com base na conectividade dos mesmos, ou seja, todos os pixeis de um componente ligado apresentam valores semelhantes de intensidade e estão de alguma forma ligados uns aos outros. Uma vez que todos os grupos tenham sido determinados, são extraídos os contornos de cada um dos componentes para análise. Na imagem 4.11a pode ser visto um desses componentes ligados, posteriormente, na imagem 4.11b pode ser visto o resultado da operação da aproximação poligonal, esta operação tenta aproximar os contornos a poligonos deste modo o conjunto de pontos que formam o componente ligado é reduzido consideravelmente permitindo melhor eficiência nas etapas seguintes.



(a) Componente ligado.



(b) Componente ligado após aproximação poligonal.

Figura 4.11: Deteção de Componentes Ligados.

4.5.2 Extração de Características

Nesta fase estão reunidos todos os requisitos para a extração de características, usar apenas a frame resultante do *background subtraction* é uma grande limitação em relação a outros métodos já desenvolvidos que usam a informação da cor dos objetos, como por exemplo, o método descrito em 2.3. Além disso, mesmo para o olho humano é difícil dizer com exatidão que determinado objeto no *background subtraction* é ou não um objeto da classe pessoa, porque os objetos podem ser deformados devido a operações de pré-processamento.

Uma das formas de encontrar características foi recorrer a um grupo de pessoas às quais foram mostrados vários vídeos de *background subtraction*, como foi descrito pelos voluntários, uma das características evidentes é o tamanho do objeto, uma vez que uma pessoa vai apresentar uma área contida entre um máximo e mínimo, no capítulo 5 vão ser exibidos testes variando os limites da área e quais as suas consequências. Obviamente a área por si só não é relevante, mas por outro lado, uma característica que resulta da área é o aspect ratio, isto é, é evidente que um objeto do tipo "pessoa"vai ter uma altura superior à sua largura como pode ser visto na imagem 4.12, esta característica vai estar igualmente entre determinados limites. A área e o aspect ratio permitem à partida eliminar objetos que não interessam, como é o caso dos carros, uma vez que um carro nunca terá uma altura superior à sua largura.

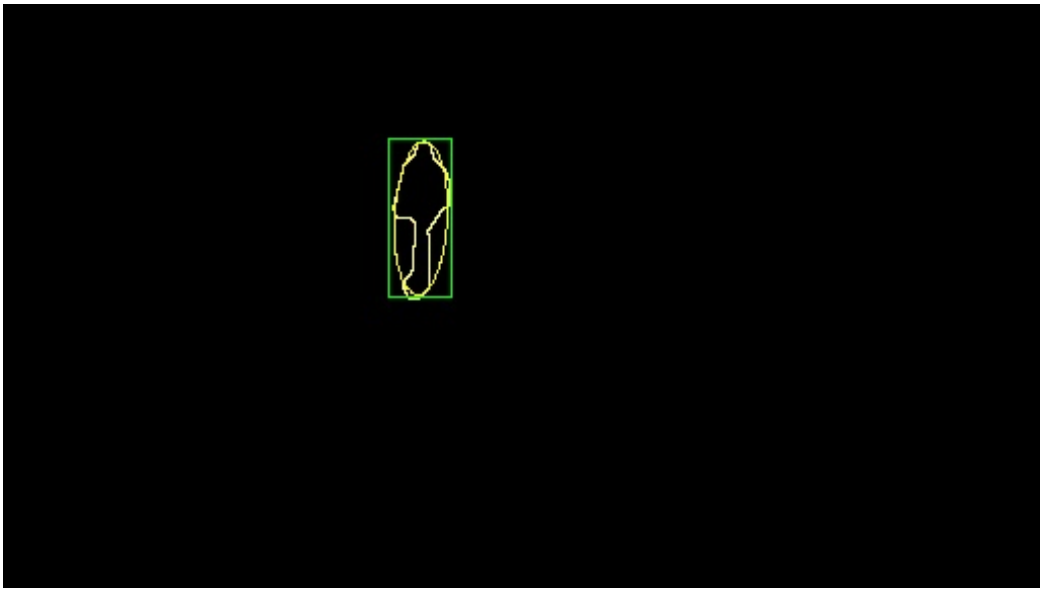


Figura 4.12: Extração de características.

Uma das características diferenciadoras é o ângulo de inclinação do objeto, isto é, um objeto do tipo "pessoa"vai assumir sempre uma posição vertical na cena, pelo que objetos resultantes de reflexões, na maior partes das vezes, vão ter um ângulo diferente de objetos reais na cena. Para calcular o ângulo é aplicada uma uma elipse ao objeto, a partir dessa elipse é calculado o ângulo recorrendo a uma função disponibilizada pela biblioteca OpenCV. Como pode ser verificado na imagem 4.12 o ângulo de um objeto do tipo "pessoa"é bastante reduzido, caso o objeto não apresente deformações resultantes do *background subtraction* ou do pré processamento esta simples característica permite obter bons resultados, mas determinadas situações, em que por exemplo a sombra não é tratada pelo algoritmo SOBS podem implicar um grande problema, pois a inclinação da elipse vai variar bastante.

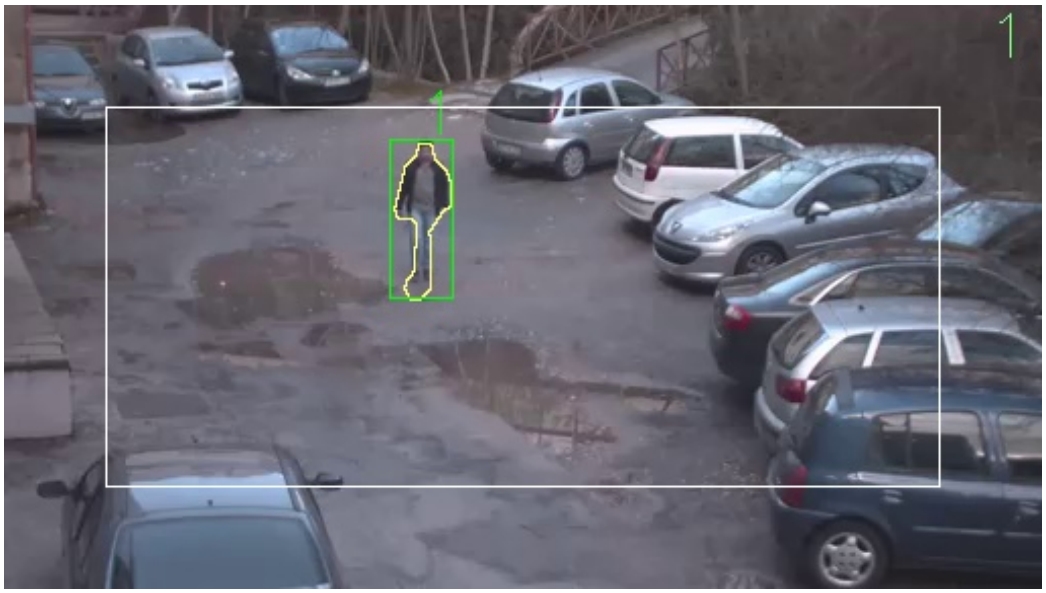


Figura 4.13: Frame final resultante da deteção.

Na imagem 4.13, é exibido o resultado da deteção na própria frame original de modo a ser possível ter uma perceção da exatidão da localização do objeto do tipo "pessoa"que foi detetado. No canto superior direito é exibido o número de pessoas atualmente na frame, esse mesmo número é atualizado sempre que é feita uma deteção. De modo a tentar resolver o problema de duas pessoas se cruzarem, foi implementado um mecanismo em que o detetor recebe a informação de quantas pessoas foram detetadas no instante de deteção imediatamente anterior, quer isto dizer que se num dado momento o detetor retornar um valor inferior àquele que recebeu da deteção anterior implica um de dois cenários: ou uma das pessoas saiu da cena, ou então houve cruzamento entre elas.

Para averiguar se determinada pessoa saiu ou não da cena é definida uma área limite que pode ser parametrizada e é observável na imagem 4.13, caso o centro de massa da pessoa se encontre fora desse limite é considerado que não houve cruzamento, e a pessoa deixa de ser considerada pela função de deteção. Caso contrário, o detetor corrige o valor de saída tendo em conta o número de pessoas na frame anterior.

(a) Componente ligado.

(b) Componente ligado após aproximação poligonal.

Figura 4.14: Frame com sobreposição de duas pessoas.

Na figura 4.14b é possível verificar o momento em que duas pessoas se cruzam na cena, mesmo para o olho humano é praticamente impossível encontrar duas formas humanas na dita frame. Aliás, até mesmo na frame original apresenta um desafio.

4.5.3 Parametrização

Como parte da solução implementada, é possível parametrizar diversos valores, os quais se encontram mencionados na tabela 4.1 e foram detalhados na secção 4.5.2. No capitulo 5 é demonstrado o impacto resultante da variação destes parâmetros na deteção.

PARÂMETRO	DESCRIÇÃO
FRAME_DETECT	Intervalo de deteção
PRE_N_ERODE	Nº. de erosões
PRE_N_DILATE	Nº. de dilatações
AREA_MIN	Área mínima do objeto
AREA_MAX	Área máxima do objeto
CONTOURS_MIN	Nº. mínimo de arestas
ANGLE_MIN	Ângulo mínimo do objeto
ANGLE_MAX	Ângulo máximo do objeto
ASPEC_RACIO_MIN	Rácio mínimo do objeto
ASPEC_RACIO_MAX	Rácio máximo do objeto
LIMIT_FRAME_X	Coordenada em X do limite da frame
LIMIT_FRAME_Y	Coordenada em Y do limite da frame

Tabela 4.1: Parâmetros

4.6 Tracking

Apesar de não ser um requisito deste trabalho, foi implementado um método de tracking relativamente simples, o *CAMShift* que foi referido com algum detalhe no capítulo 2 na secção 2.4. Após realizada uma deteção, o retângulo usado para verificar a área e o aspect ratio é dado à função de tracking, nomeadamente à função CamShift do OpenCV, isto na frame seguinte. Este retângulo é a ROI que indica ao algoritmo de tracking qual a posição inicial/anterior do objeto. A função devolve uma nova ROI com a localização mais provável do objeto na frame atual, a imagem 4.15 ilustra o resultado do tracking, é visível um novo retângulo (com cor diferente) que contem o objeto, neste caso corretamente identificado.

Figura 4.15: Frame final com com o tracking implementado.

4.7 Conclusões

Em jeito de conclusão deste capítulo, é de referir a clara organização da estrutura da implementação, a qual se reflete numa solução simples e fácil de perceber. No capítulo seguinte são apresentados os resultados desta implementação, e por consequência as consequências de determinadas opções tomadas.

Capítulo 5

Resultados e Discussão

5.1 Introdução

Neste capítulo são apresentados os resultados conseguidos com o trabalho realizado assim como uma análise e discussão dos mesmos. São expostas diversas situações assim como a variação dos parâmetros de deteção nas mesmas.

5.2 Cena de Teste

A cena de teste, apresenta inúmeras características que influenciam a performance do detetor. Neste caso concreto o cenário é de carácter complexo, uma vez que estão presentes muitos dos fatores que compõem um cenário considerado de difícil. Entre esses mesmos fatores, o número de objetos presentes na cena, o contraste entre os mesmos e o *background*, as condições de iluminação, as sombras tanto estáticas como dinâmicas (evidentes na imagem 5.1b), as reflexões (presentes na imagem 5.1a) e o complexo *background* devido à diversidade de objetos.

(a) Frame com reflexões. (b) Frame com sombra estática.

Figura 5.1: Exemplos de situações numa cena.

5.3 Características dos Vídeos de Teste

Tendo em conta que o sistema terá que ser aplicado a situações de tempo real, os vídeos capturados, inicialmente com uma resolução de 1280 por 720 pixeis foram redimensionados para uma resolução inferior de 512 por 288 pixeis. Deste modo é possível aumentar a eficiência da deteção, no entanto, com possíveis repercussões na eficácia, uma vez que estamos a reduzir o detalhe das imagens significativamente.

Na tabela 5.1, estão expostas diversas características dos vídeos utilizados. Entre elas, o numero efetivo de pessoas na cena e se estas se cruzam na cena a determinado momento, a existência de sombras evidentes nas pessoas alvo de deteção ou reflexões resultantes de poças de água na cena, assim como a presença de outros objetos, como por exemplo carros. Como pode ser verificado, estão disponíveis vários cenários com características e situações variadas que são testadas na secção seguinte.

VÍDEO	Nº. PESSOAS	CRUZAMENTO	SOMBRAS	REFLEXÕES	OUTROS OBJETOS
V6r	1	Não	Não	Não	Sim
V7r	1	Não	Não	Não	Não
V8r	1	Não	Não	Não	Não
V9r	2	Sim	Não	Não	Sim
V10r	2	Não	Não	Não	Sim
V12r	4	Sim	Sim	Não	Não
V15r	1	Não	Não	Sim	Não
V16r[a]	1	Não	Não	Não	Não
V17r	1	Não	Sim	Não	Sim
V18r	0	Nâo	Não	Não	Sim
V19r	2	Sim	Não	Sim	Não
V20r	0	Não	Não	Não	Sim
V21r	1	Não	Não	Sim	Não
V22r	0	Não	Não	Não	Não

[a]Este vídeo contém dois momentos de deteção distintos.

Tabela 5.1: Detalhes dos vídeos utilizados.

5.4 Testes e Discussão

Para gerar resultados partiu-se de uma configuração base, exibida na tabela 5.2, foram valores selecionados durante a fase de implementação, posteriormente são variados alguns parâmetros e apresentados os consequentes resultados.

5.4.1 Metodologia do Teste

Para efetuar os testes, em cada vídeo é selecionado um intervalo de frames, em que é atribuído o número de pessoas na cena entre esse mesmo intervalo com o número correto de pessoas na cena. Cada vídeo tem associado um ou mais ficheiro de configuração onde consta esse intervalo. A percentagem de acerto de uma determinada configuração e calculada segundo a expressão,

$$\%\ \text{Acerto} = \sum_{i=1}^{\text{Nº de deteções}} \frac{\text{Nº de pessoas detetadas}}{\text{Nº de pessoas no ficheiro de configuração}} \tag{5.1}$$

isto para cada ficheiro de configuração, e no final é calculada a média de acerto total.

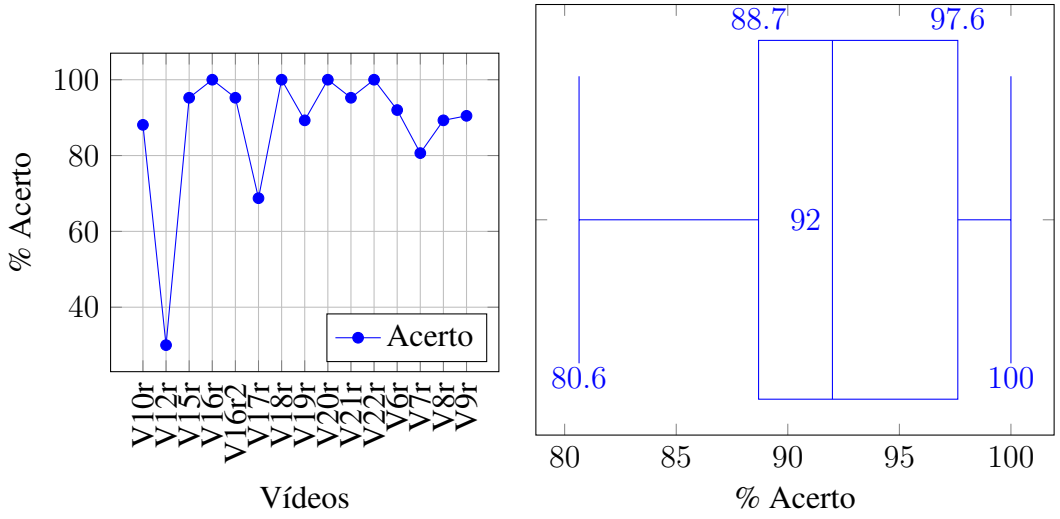
5.4.2 Parâmetros Base

PARÂMETRO	VALOR
FRAME_DETECT	10.000
PRE_N_ERODE	1.000
PRE_N_DILATE	1.000
AREA_MIN	250.000
AREA_MAX	2,250.000
CONTOURS_MIN	5.000
ANGLE_MIN	15.000
ANGLE_MAX	165.000
ASPEC_RACIO_MIN	1.618
ASPEC_RACIO_MAX	3.236
LIMIT_FRAME_X	50.000
LIMIT_FRAME_Y	50.000

Tabela 5.2: Configuração base.

5.4.3 Resultados Base

Com base na configuração referida na tabela 5.2, foi obtido uma taxa de acerto de 90.51%, que é um valor razoável. Como pode ser verificado no gráfico 5.2a o vídeo *V12r* foi onde foram obtidos os piores resultados, este foram claramente devido ao elevado número de pessoas presentes na cena, e aos momentos em que estes estavam muito próximos, como se verifica na imagem 5.3.

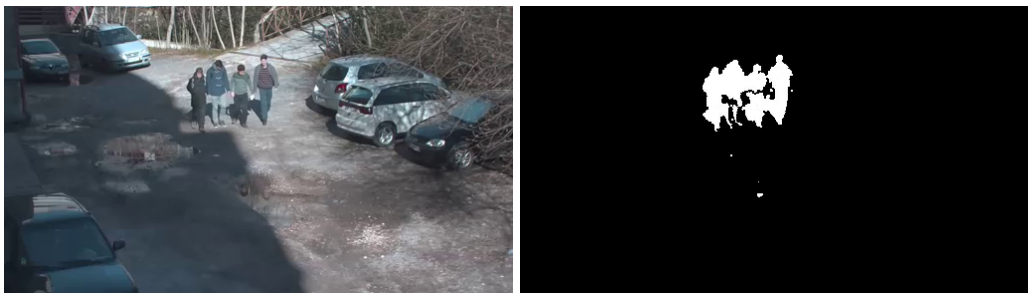


(a) Gráfico linear.

(b) Gráfico *Box-Plot*.

Figura 5.2: Gráficos com resultados utilizando a configuração base.

No gráfico 5.2b é possível verificar que a mediana, excluindo os outliers, é de 92%. Quer isto dizer que excluindo as cenas mais complexas, o detetor apresenta uma boa taxa de acerto, neste caso para uma deteção a cada 10 frames.



(a) Frame original.

(b) Frame resultante do SOBS.

Figura 5.3: Exemplo de uma frame do vídeo *V12r*.

Outra observação interessante é o facto de os vídeos que não têm pessoas (*V18r*, *V20r* e *V22r*) apresentam uma taxa de acerto de 100%, quer isto dizer que não há ocorrência de falsos positivos nestes vídeos incluindo aqueles em que aparecem outros objetos, como por exemplo veículos.

VÍDEO	% ACERTO
V10r	88.10
V12r	30.00
V15r	95.24
V16r	100.00
V16r2	95.24
V17r	68.75
V18r	100.00
V19r	89.29
V20r	100.00
V21r	95.24
V22r	100.00
V6r	92.00
V7r	80.65
V8r	89.29
V9r	90.48
Média	90.51

Tabela 5.3: Tabela de resultados.

A baixa taxa de acerto no vídeo *V17r* resulta do facto de ser causado ruído no BS devido à chegada de uma pessoa de veículo, como se verifica na imagem 5.4.

(a) Frame original. (b) Frame resultante do SOBS.

Figura 5.4: Exemplo de uma frame do vídeo *V17r*.

5.4.4 Variação do Intervalo de Deteção

Neste teste foi variado o parâmetro FRAME_DETECT, que corresponde ao intervalo de deteção, isto é, o intervalo de frames em que é realizada uma deteção, no caso base, é feita uma deteção a cada 10 frames. É interessante observar que a performance ao ser feita uma deteção por frame é muito reduzida, sendo apenas eficaz nos vídeos sem pessoas como se pode verificar no gráfico 5.5. O elevado erro geral é devido às anomalias provocadas por sombras e por haver situações em que as pessoas estão demasiado próximas. Como a deteção é feita com elevada frequência essas situações prejudicam os resultados porque o detetor vai apresentar um resultado errado durante essas situações com mais frequência.

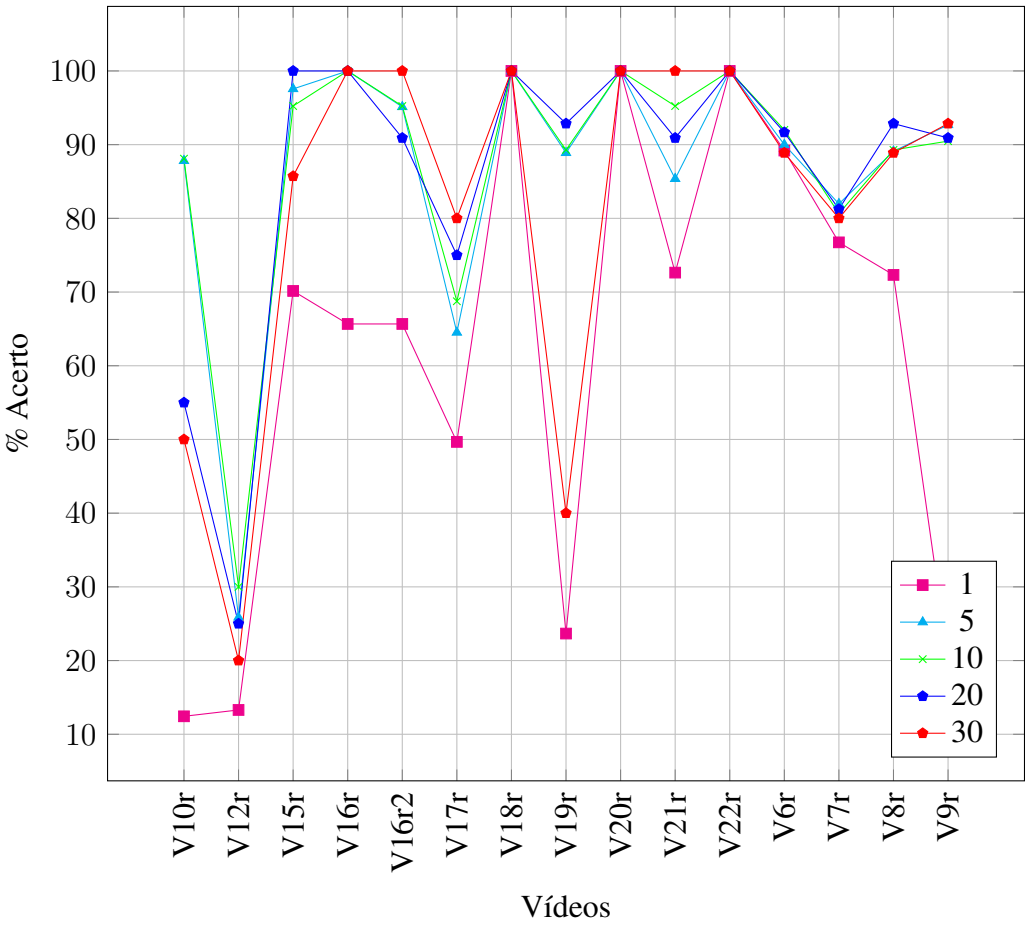


Figura 5.5: Gráfico linear com a variação do parâmetro FRAME_DETECT.

Nesta avaliação global dos resultados verifica-se que, em relação ao parâmetro FRAME_DETECT, o valor ideal situa-se na deteção a cada 10 frames, como demontra o gráfico 5.6, isto é devido aos vídeos apresentarem um framerate de 10 frames por segundo. Além disso quanto maior o intervalo de deteção, maior é a eficiência da deteção pois é gasto menos tempo de processamento na fase de deteção.

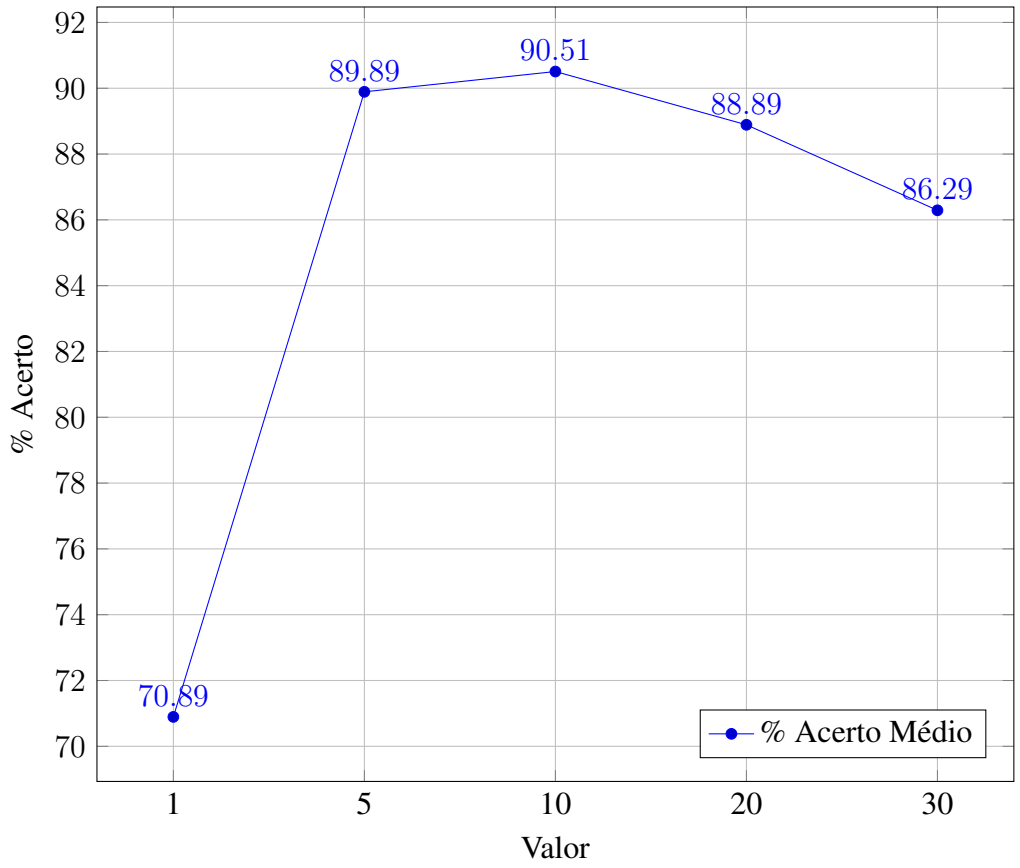


Figura 5.6: Gráfico linear com a variação média do parâmetro FRAME_DETECT.

5.4.5 Variação do Nº. de Erosões e Dilatações

Neste teste é variado o número de erosões e dilatações, este variação é representada como um par ordenado, em que por exemplo, (1,2) significa uma erosão e duas dilatações. Como foi referido na secção 4.4, o pré-processamento tem grande impacto nos resultados finais, uma vez que o detetor depende fortemente da qualidade do BS. No gráfico 5.7 é possível verificar que os resultados são interessantes no ponto de vista de vídeos individuais. Existem, de facto, vídeos em que a configuração base não apresenta a melhor performance, como por exemplo, o vídeo *V8r* recorrendo a um erosão e duas dilatações, este aumento deve-se ao facto de a dilatação aumentar os objetos, deste modo se a silhueta da pessoa ficar dividida devido a ruído, a dupla dilatação resolve o problema unindo novamente a forma.

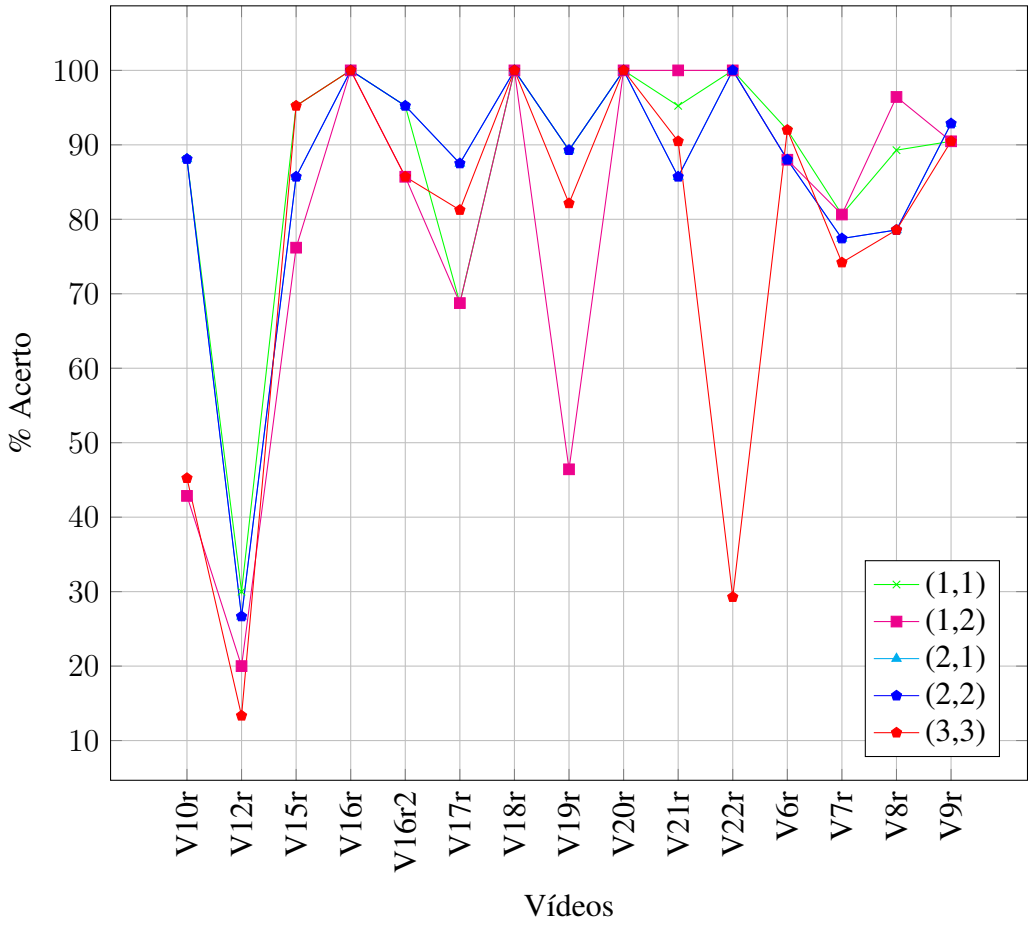


Figura 5.7: Gráfico linear com a variação dos parâmetros (PRE_N_ERODE , PRE_N_DILATE).

Embora no gráfico 5.7 existam situações em que a configuração base de apenas uma erosão e uma dilatação não seja a melhor, no gráfico 5.8 verifica-se que, no geral, a configuração base ainda apresenta a melhor média de resultados. Outra conclusão interessante que se pode retirar é o facto de segundo estes resultados a erosão ser mais relevante que a dilatação, isto é, ao aumentar a erosão obtém-se melhores resultados do que aumentando a dilação. Esta observação pode ser explicada devido à erosão eliminar bastante ruído, daí os resultados, a dilatação apenas tentar recuperar detalhe que a erosão retirado erroneamente.

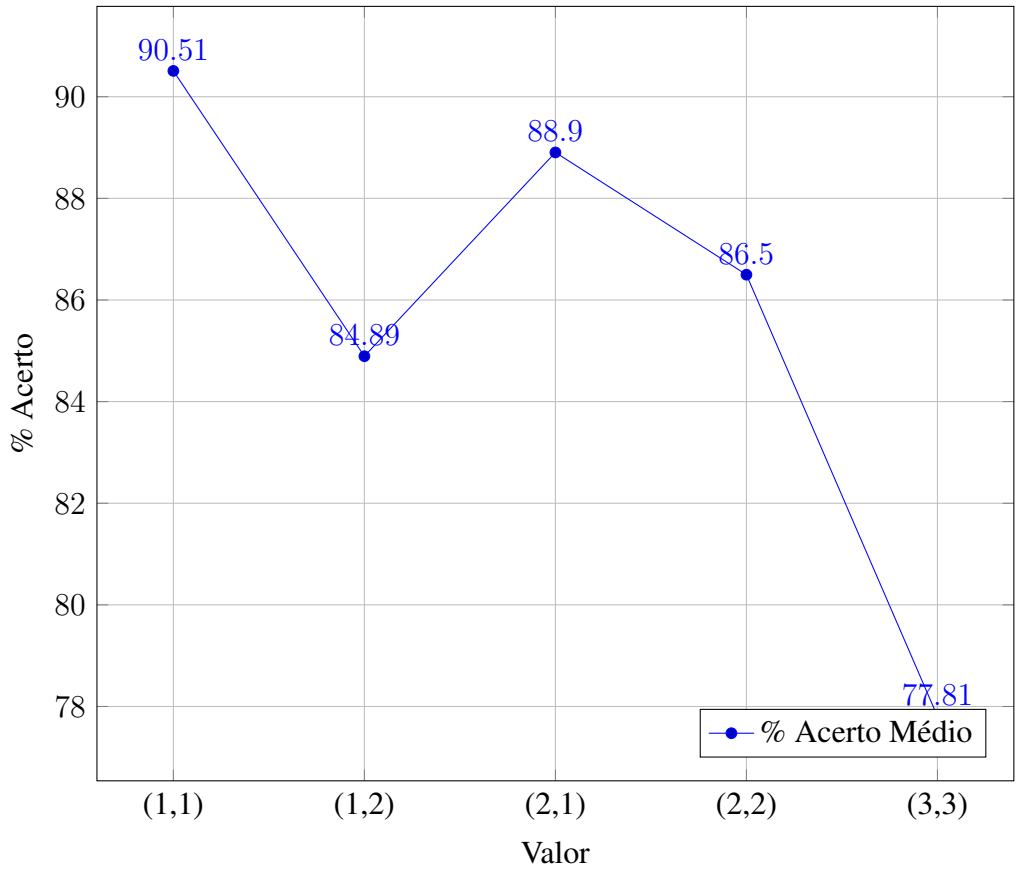


Figura 5.8: Gráfico linear com a variação média dos parâmetros (PRE_N_ERODE , PRE_N_DILATE).

5.4.6 Variação da Área

Área Mínima

O limite mínimo da área é bastante relevante uma vez que o mesmo é responsável por eliminar ruído que não foi retirado na fase de pré-processamento, como por exemplo reflexões. Como pode ser verificado no gráfico 5.9 no vídeo *V22r* que apenas contem ruído, quando o mínimo é de 50 pixeis, não é possível filtrar a enorme quantidade de ruído, na imagem 1.2b serve de exemplo para demonstrar o mesmo.

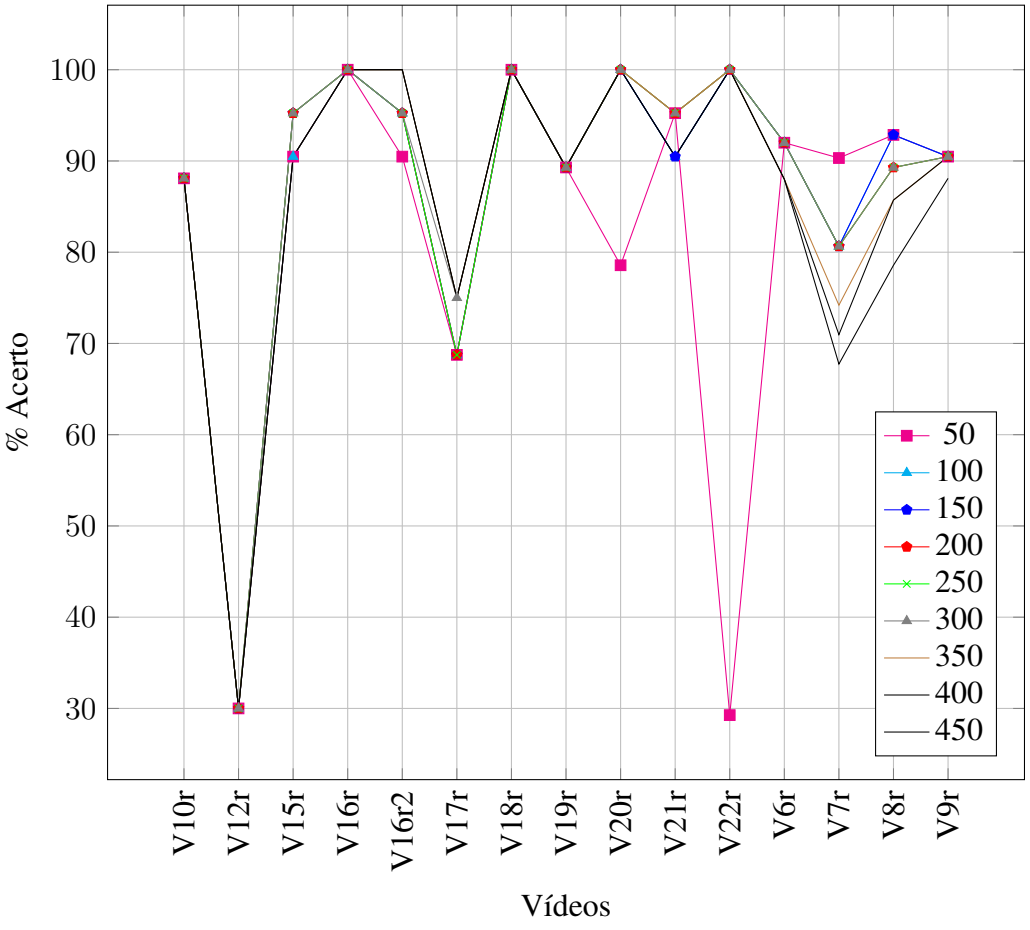


Figura 5.9: Gráfico linear com a variação do parâmetro AREA_MIN.

Um dado interessante é o facto de o valor mínimo utilizado na configuração base não é de facto o melhor valor, como pode ser verificado no gráfico 5.10, o valor de uma área mínima de 300 pixeis obtém uma taxa media de acerto ligeiramente superior, de 90.51% para 90.78%.

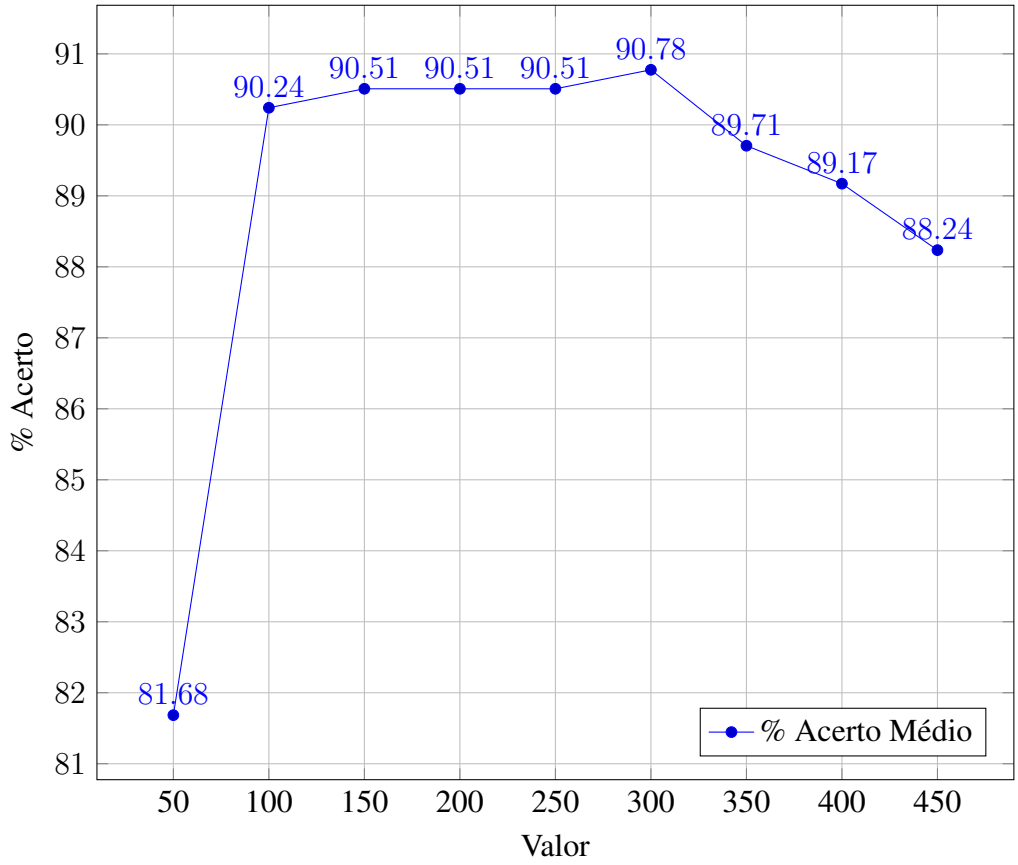


Figura 5.10: Gráfico linear com a variação média dos parâmetros AREA_MIN.

Área Máxima

O limite máximo da área permite excluir objetos de grandes dimensões, como por exemplo carros, que devido à sua área total ser bastante superior à área de uma pessoa são facilmente excluídos. Outro fator a ter em conta é que há situações em que uma pessoa se encontra muito próxima da câmara, e essa situação é ideal para uma fase posterior de reconhecimento, a área dessa pessoa vai ser razoavelmente grande e tem que ser garantido que é detetada. Por outro lado, a utilização de uma máximo maior permite contornar o problema das sombras, uma vez que a sombra conta para a área do objeto detetado. Como pode ser verificado no gráfico 5.11 as variações acontecem precisamente em situações desse tipo.

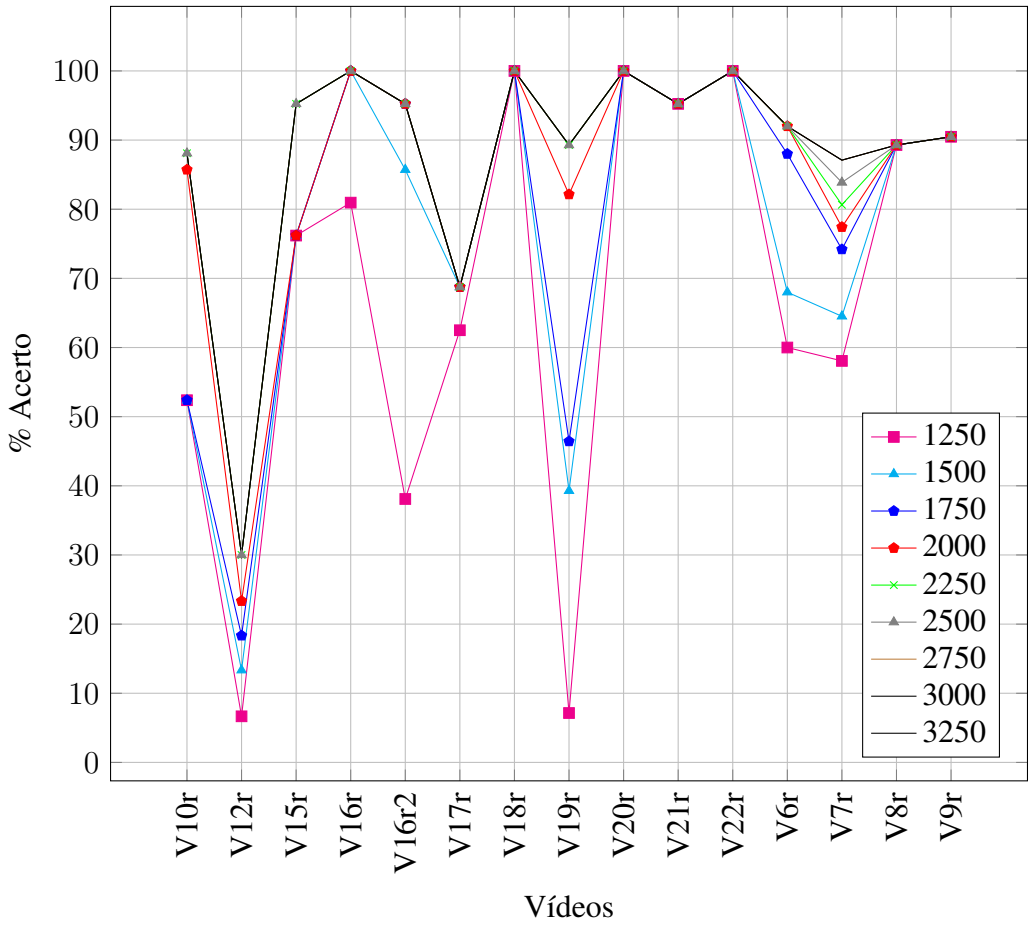


Figura 5.11: Gráfico linear com a variação do parâmetro AREA_MAX.

Em sintonia com os resultados obtidos no teste do valor mínimo 5.4.6, o valor máximo utilizado na configuração base também não é o que obtém melhores resultados. Aliás, pelo gráfico 5.12 é possível verificar que a percentagem média de acerto tende a estabilizar a partir de valores de 2750 pixeis, quer isto dizer que a exclusão de objetos grandes é maioritariamente conseguida através de outra característica.

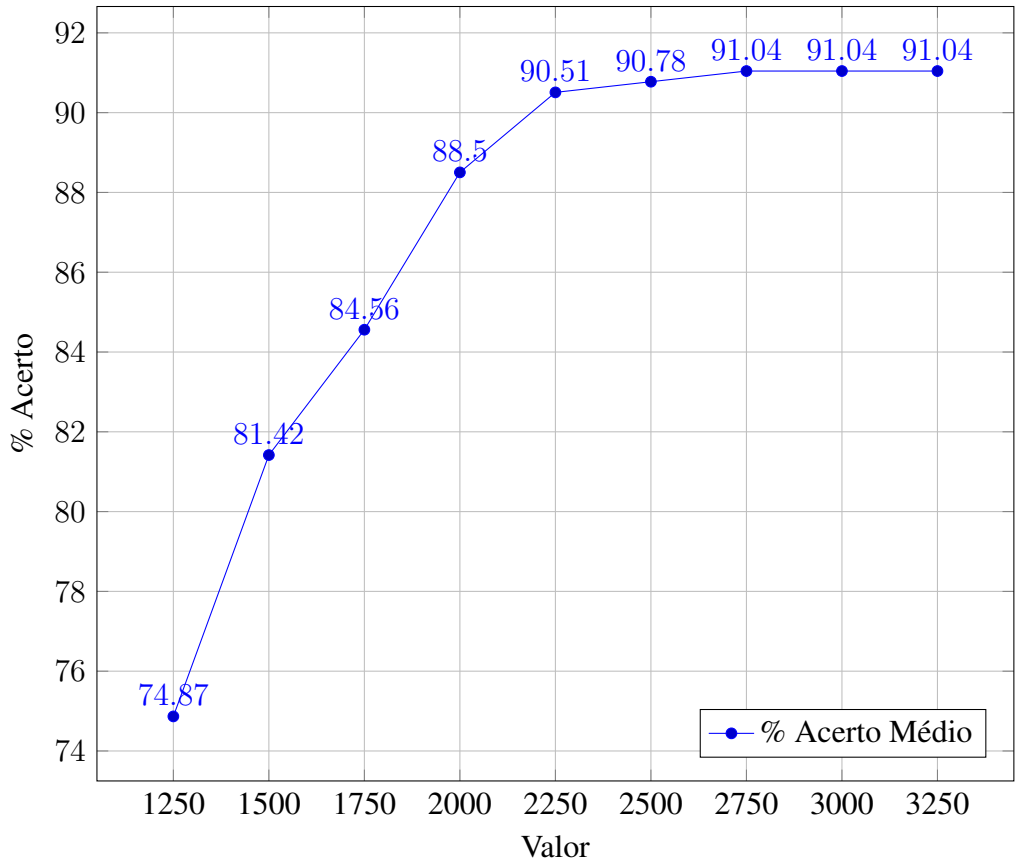


Figura 5.12: Gráfico linear com a variação média dos parâmetros AREA_MAX.

Teste com o melhor máximo e mínimo

Após a realização dos testes individuais aos limites mínimos e máximos, foi realizado um teste com o melhor valor de cada limite obtido, embora conscientemente tal resultado possa não ser de facto o melhor.

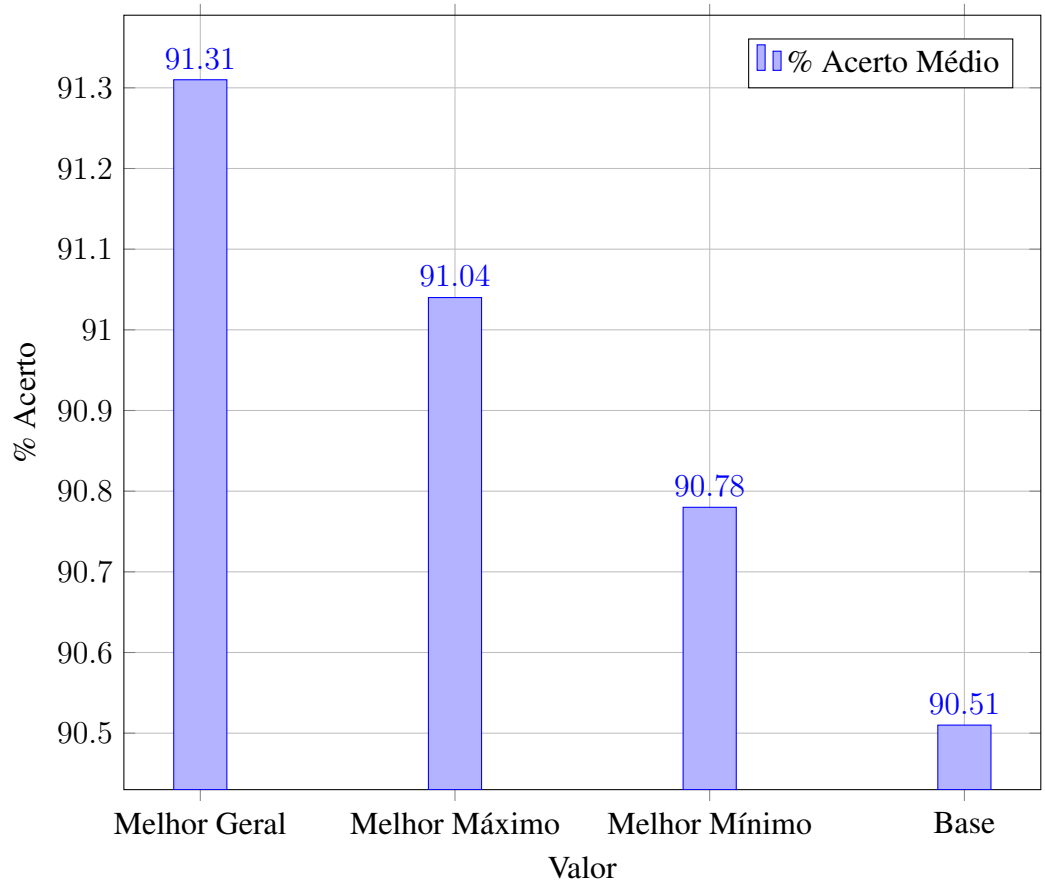


Figura 5.13: Gráficos com resultados utilizando o melhor valor máximo e mínimo.

Contudo, neste caso ao utilizar os melhores valores para o respetivo máximo e mínimo foi possível obter um aumento da taxa de acerto de 90.51% para 91.31%, demonstrado no gráfico 5.13, o que sugere que os parâmetros AREA_MAX e AREA_MIN são independentes. No entanto para ter a certeza de tal afirmação seriam precisos mais testes.

5.4.7 Variação do Ângulo

O ângulo é uma das características mais importante, porque em conjunto com a área providencia um método bastante eficaz de eliminar ruído provocado por reflexões, uma vez que estão dificilmente vão ter um alguém semelhante a um objeto na cena. No entanto as sombras dos objetos poderão causar problemas uma vez que as mesmas influenciam o calculo do ângulo do objeto. Na questão do ângulo máximo e mínimo do objeto, neste caso fará sentido variar os limites inferior e superior em conjunto, uma vez que os objetos apenas podem apresentar uma inclinação para um único lado. O gráfico 5.14 ilustra a variação do acerto nos diversos vídeos de acordo com o valor dos parâmetros ANGLE_MIN e ANGLE_MAX, neste caso, também sob a forma de um par ordenado.

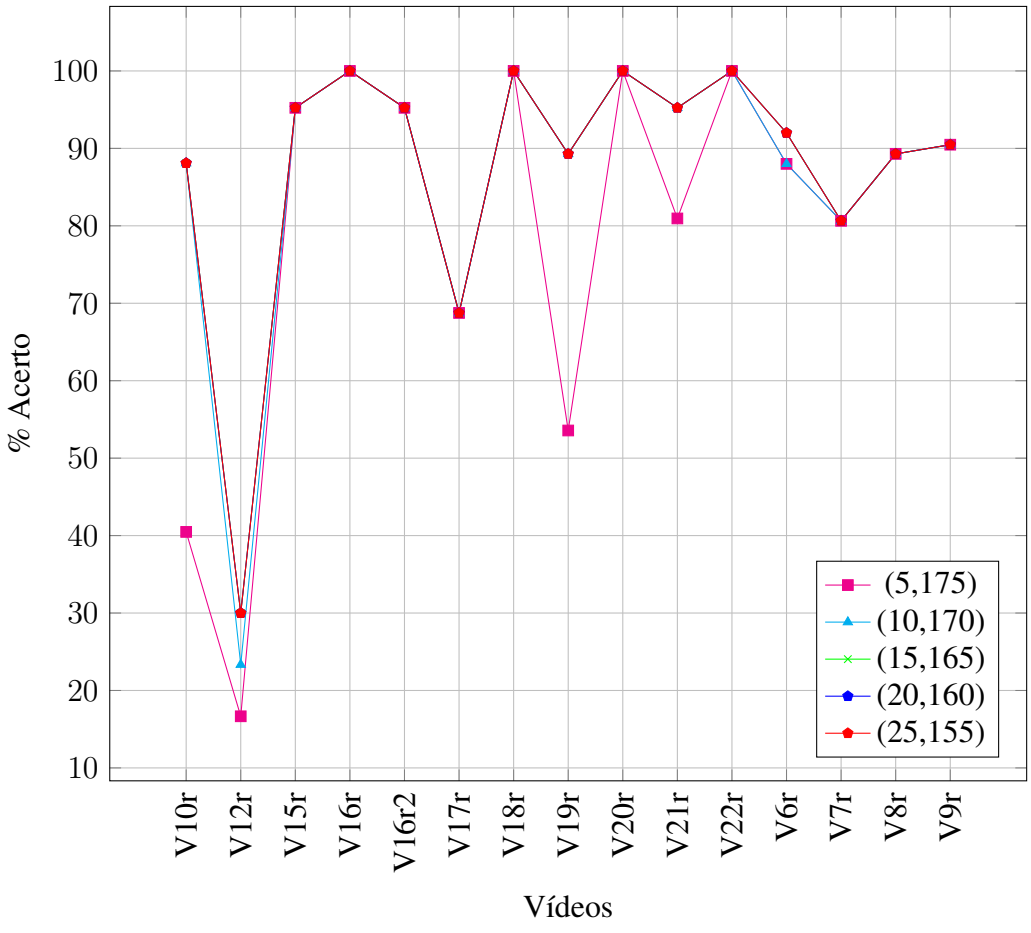


Figura 5.14: Gráfico linear com a variação dos parâmetros (ANGLE_MIN , ANGLE_MAX.)

Os resultados obtidos e ilustrados no gráfico 5.15 sugerem que á medida que o intervalo do ângulo diminui, começa a ser irrelevante, isto acontece porque não existem objetos nos vídeos de teste que cumpram simultaneamente aos condições da área e do limites do ângulo. Além disso o gráfico 5.15 mostra que ao utilizar limites reduzidos para os ângulos implica uma redução na taxa de acerto, isto porque como seria de esperar, o facto de o BS não ser perfeito resulta em deformidades no objeto que levam a que o ângulo seja alterado.

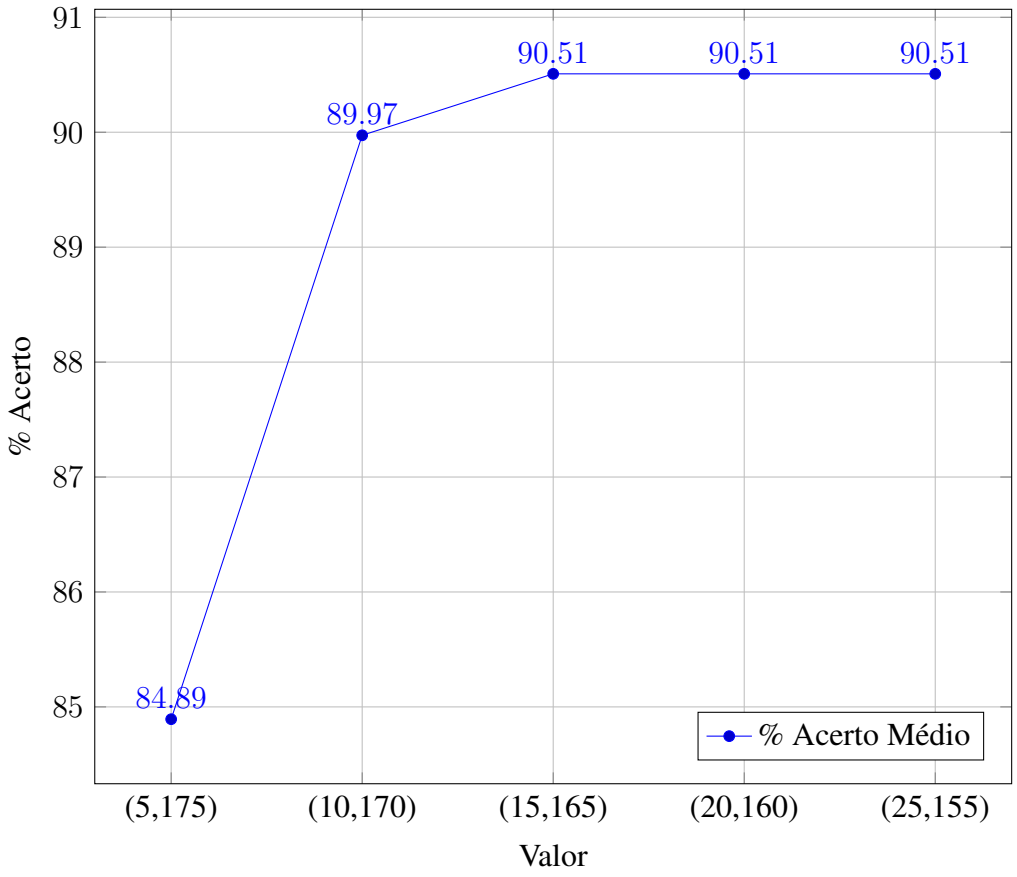


Figura 5.15: Gráfico linear com a variação média dos parâmetros (ANGLE_MIN , ANGLE_MAX.)

5.4.8 Variação do Aspect ratio

Neste teste é testada, de uma maneira muito rudimentar, a forma do objeto, isto é, um aspect ratio de 1 corresponde a uma forma quadrada, qualquer valor acima é um retângulo cuja a relação entre os seus lados é esse próprio valor. O valor na configuração base para o parâmetro ASPEC_RACIO_MIN não foi escolhido ao acaso, trata-se da proporção de ouro [21], assim como o valor de ASPEC_RACIO_MAX é duas vezes superior. Com base nos resultados expressos no gráfico 5.16 podemos afirmar que um aspect ratio próximo de 1 diminui significativamente a taxa de acerto, isto porque o único motivo pelo qual uma pessoa teria um aspect ratio de 1, seria a sombra da própria estar a um ângulo de 45º, algo que deve ser razoavelmente tratado pelo método SOBS.

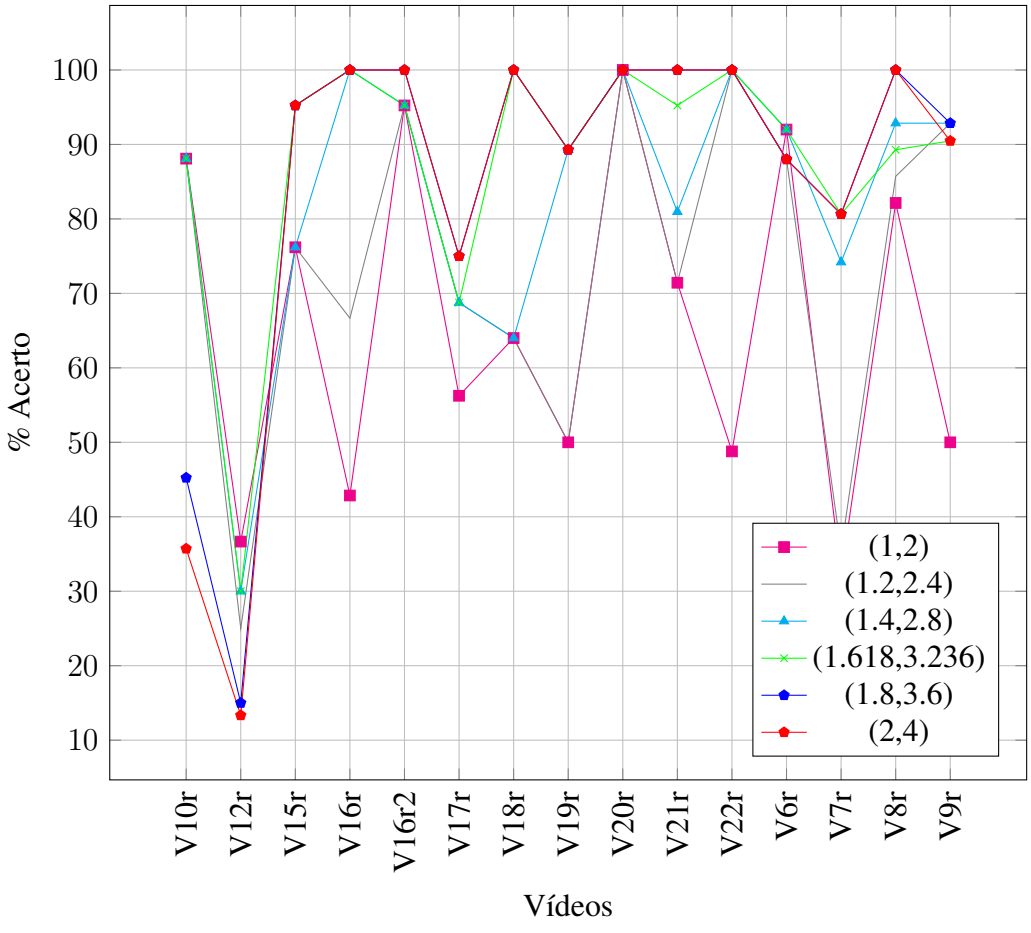


Figura 5.16: Gráfico linear com a variação dos parâmetros (ASPEC_RACIO_MIN, ASPEC_RACIO_MAX.)

Como pode ser verificado no gráfico 5.17 a decisão de utilizar a proporção de ouro (1.618) para o aspect ratio foi uma decisão acertada, pelo que é óbvio que uma pessoa vai sempre apresentar uma altura de cerca de 1.5 a sua largura.

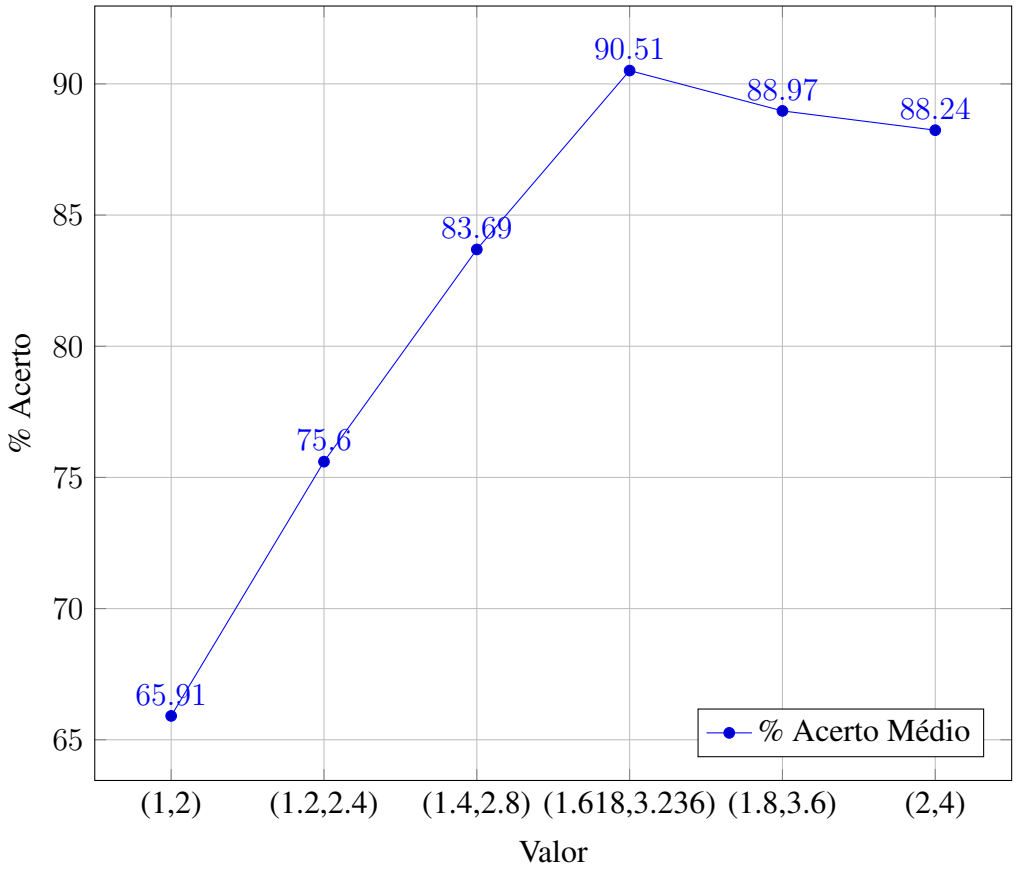


Figura 5.17: Gráfico linear com a variação média dos parâmetros (ASPEC_RACIO_MIN, ASPEC_RACIO_MAX.)

5.4.9 Reflexão Crítica

Após o estudo destes resultados, podemos concluir que a configuração base ficou muito próxima de obter os melhores resultados, embora para realmente confirmar tal afirmação seriam obviamente necessários mais testes de modo a cobrir todas as combinações de parametrizações. No entanto a melhoria na taxa de acerto de 90.51% para 91.31% é significativa, mas o modo como essa melhoria foi obtida (5.4.6) sugere que há independência de características.

É evidente que o pré-processamento é uma fase de elevada importância, como pode ser verificado na sub-secção 5.4.5, variando apenas o número de erosões e dilatações é possível observar alterações significativas nos resultados. O mesmo pode ser dito aspect ratio, em que claramente à um intervalo de valores que é ideal, e a mínima alteração causa uma grande taxa de erro.

No que diz respeito ao limite da área de deteção, é um parâmetro que não é relevante para a percentagem de acerto, uma vez que o modo como são realizados os testes implica que o intervalo de frames em que é realizada a deteção tem que se encontrar dentro do limite definido.

5.5 Conclusões

Este capitulo sumariza o trabalho realizado de uma forma satisfatória, o estudo dos testes da variação dos parâmetros de deteção permitiu observar o impacto desses mesmos parâmetros nos resultados obtidos, e deste modo, apurar o porquê de determinadas características serem mais eficazes que outras. No capítulo seguinte serão apuradas as conclusões finais referentes ao trabalho realizado e também serão abordadas diversas questões, como por exemplo, o que faltou fazer, trabalho extra que poderia ter sido realizado e também melhorias de modo a obter melhores resultados

Capítulo 6

Conclusões e Trabalho Futuro

6.1 Conclusões Principais

Os objetos propostos para este trabalho foram atingidos com sucesso, através das características extraídas do BS. É evidente que todo o trabalho realizado depende maioritariamente de um só fator, o BS. De uma forma crítica, ficou claro que o método SOBS, apesar de ser o melhor método no momento de acordo com estudo realizado pelo SOCIA LAB, está longe de obter bons resultados num ambiente de características difíceis. Este apresenta dificuldades em tratar objetos em situações de variação de luminosidade, e também o facto de o algoritmo não conseguir efetuar a remoção satisfatória de sombras faz com que, de um ponto de vista cientifico, a arte BS ainda necessita de muito trabalho, pois os métodos apenas são eficazes em situações de ambiente controlado. Alem disso, também é evidente que o pré-processamento possuí um elevada importância, pois nessa fase é possível colmatar as lacunas do BS. A dificuldade encontra-se em obter um conjunto de parâmetros que seja unânime para a variação das condições da cena, uma vez que a cena utilizada apresenta inúmeras dificuldades. Os resultados obtidos, tendo em conta todos estes fatores, são de uma forma geral satisfatórios, com uma taxa de acerto de 91.31%, com a consciência que estes resultados são de certo modo direcionados para a cena de teste em questão, isto é, o uso de características otimizadas para a cena em questão faz com que os parâmetros de deteção necessitem de adaptação a cada nova cena.

6.2 Trabalho Futuro

Com base nos resultado obtidos, numa perspetiva de trabalho futuro evidenciam-se as seguintes tarefas:

- Melhoramento do BS
- Nova Característica
- Algoritmo Genético
- Maior Base de Dados de Vídeos de Teste
- Implementar Outro Método de Tracking

6.2.1 Melhoramento do Background Subtration

Como foi referido neste trabalho, o BS requer uma forte atenção, pois os resultados dependem do mesmo. Pelo que um dos trabalhos a realizar seria o melhoramento o método SOBS, ou ate mesmo o desenvolvimento um novo método. De qualquer modo, o desenvolvimento de um método que trata-se as sombras dos objetos implicaria diretamente um melhoramento nos resultados, porque como se pode verificar na imagem 6.1, a sombra altera por completo a forma do objeto que por consequência se reflete na extração de características.

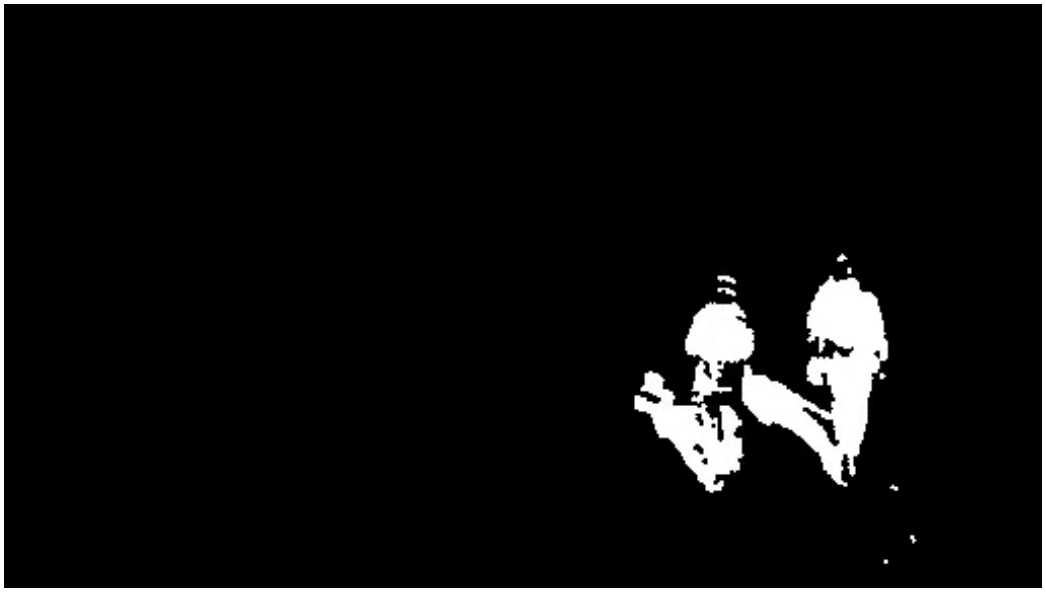


Figura 6.1: Exemplo de *output* do SOBS.

6.2.2 Nova Característica

De modo a melhorar a deteção em cenários com várias pessoas, um característica muito útil seria a análise de concavidades, isto é, procurar concavidades num objeto de modo a permitir a separação deste, isto no caso de um grupo de pessoas. Poderia ser utilizado o método proposto por Li *et* al. [10] em 2009. Como ilustra a imagem 6.2, a linha a vermelho representa as concavidades no objeto, que claramente se trata não de 1, mas 4 objetos distintos. Com base nessa análise seria possível à posteriori separar os objetos, aumentando assim a performance do detetor em cenários deste tipo.

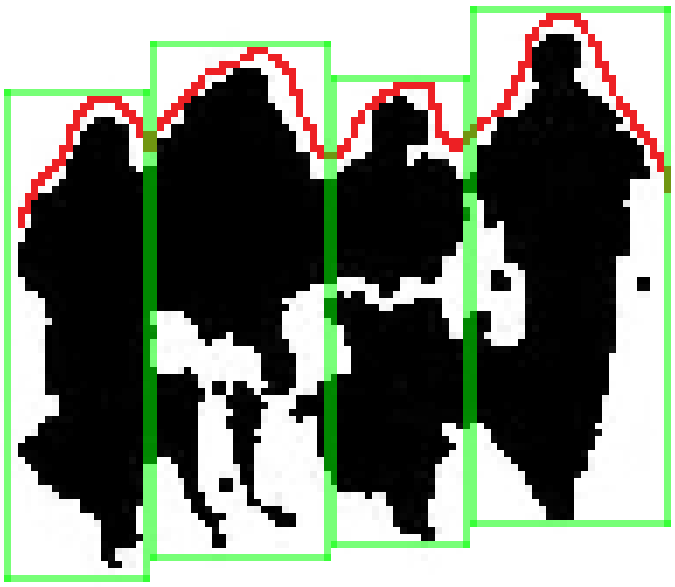


Figura 6.2: Exemplo de nova característica.

6.2.3 Algoritmo Genético

Outra situação clara que ficou por tratar, foi o facto de não terem sido testados exaustivamente todos os parâmetros de configuração possíveis, mesmo excluindo os parâmetros do detetor, temos uma enorme variedade só na parte de pré-processamento. Um modo de resolver este problema seria recorrer à implementação de um algoritmo genético, deste modo seria possível testar um grande conjunto de parâmetros e conseguir aprimorar a melhor configuração dos mesmos.

6.2.4 Maior Base de Dados de Vídeos de Teste

Na fase de resultados foi possível verificar que seria necessário outro conjunto de vídeos para efetuar mais testes e averiguar mais situações, de modo a aumentar a robustez do detetor, um cenário que deveria ser testado é a situação em que uma pessoa fica imóvel na cena, tal situação é por norma dificil de tratar pois os algoritmos de BS tendem a considerar o objeto parado como *background*. Tais situações deveriam ser aproveitadas pois nesse momento seria mais fácil focar um objeto para reconhecimento posterior.

6.2.5 Implementar Outro Método de Tracking

Como foi referido na secção 4.6, apesar de não ser um requisito foi implementado um método de tracking, o *CAMShift*. Este provou apenas ser eficaz numa situação em que apenas existe um objeto na cena, como pode ser verificado na imagem **??**, deste modo este método de tracking deveria ser substituído por outro, como por exemplo o *Optical Flow* [11].

Bibliografia

[1] G. Bradski. *Dr. Dobb's Journal of Software Tools*.

[2] G.R. Bradski. Real time face and object tracking as a component of a perceptual user interface. In *Applications of Computer Vision, 1998. WACV '98. Proceedings., Fourth IEEE Workshop on*, pages 214–219, Oct 1998.

[3] John Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-8(6):679–698, Nov 1986.

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

[5] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati. Detecting moving objects, ghosts, and shadows in video streams. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(10):1337–1342, Oct 2003.

[6] Ming-Kuei Hu. Visual pattern recognition by moment invariants. *Information Theory, IRE Transactions on*, 8(2):179–187, February 1962.

[7] R. Jain and H. Nagel. On the analysis of accumulative difference pictures from image sequences of real world scenes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1979.

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

[9] T. Kohonen. *Self-organization and Associative Memory: 3rd Edition*. Springer-Verlag New York, Inc., New York, NY, USA, 1989.

[10] Min Li, Zhaoxiang Zhang, Kaiqi Huang, and Tieniu Tan. Rapid and robust human detection and tracking based on omega-shape features. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pages 2545–2548, Nov 2009.

[11] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'81, pages 674–679, San Francisco, CA, USA, 1981. Morgan Kaufmann Publishers Inc.

[12] L. Maddalena and A. Petrosino. A self-organizing approach to background subtraction for visual surveillance applications. *Image Processing, IEEE Transactions on*, 17(7):1168–1177, July 2008.

[13] Juliette Mattioli. Minkowski operations and vector spaces. *Set-Valued Analysis*, 3(1):33–50, 1995.

[14] N.J.B. McFarlane and C.P. Schofield. Segmentation and tracking of piglets in images. *Machine Vision and Applications*, 8(3):187–193, 1995.

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

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

[17] George Stockman and Linda G. Shapiro. *Computer Vision*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1st edition, 2001.

[18] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511–I–518 vol.1, 2001.

[19] Wikipedia. C++ — wikipedia, the free encyclopedia, 2014. [Online; acedido a 3 de Maio de 2014].

[20] Wikipedia. C (programming language) — wikipedia, the free encyclopedia, 2014. [Online; acedido a 3 de Maio de 2014].

[21] Wikipedia. Golden ratio — wikipedia, the free encyclopedia, 2014. [Online; acedido a 8 de Maio de 2014].

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

[23] Z. Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 2, pages 28–31 Vol.2, Aug 2004.