

Faculdade de Engenharia da Universidade do Porto



Sistema para pontuação de um alvo de tiro desportivo

Eduardo Miguel Ribeiro Pereira Bártolo

**Dissertação realizada no âmbito do
Mestrado Integrado em Engenharia Eletrotécnica e de
Computadores
Major Automação**

Orientador: Professor Doutor Armando Luís Sousa Araújo

16 de Julho de 2018

© Eduardo Miguel Ribeiro Pereira Bártolo, 2018



Resumo

Foi no século XIX que se começaram a organizar os primeiros concursos de tiro desportivo. Nestes, a contagem da pontuação, para classificar os concorrentes, era efetuada manualmente. Com a evolução tecnológica, já muitos locais de treino possuem sistemas automáticos de classificação. Porém, o preço destes sistemas é demasiado elevado para poderem ser generalizadamente usados, ao nível dos clubes de tiro desportivo. O método automático de classificação mais utilizado é baseado na triangulação do sinal sonoro do impacto do projétil no alvo. Todavia, recentemente, o mais moderno sistema de classificação de alvos é o sistema de visão por computador. Este calcula o centro do impacto dos projéteis no alvo.

Nesta tese pretendeu-se utilizar uma câmara fotográfica para registar a imagem do alvo. O sistema a implementar deveria ser acessível a qualquer clube de tiro e de fácil utilização. Foram utilizadas imagens com a menor resolução possível. Estas foram sujeitas ao algoritmo desenvolvido, de forma a calcular o centro do alvo e o centro do disparo e, assim, atribuir uma classificação.

Através da análise dos resultados obtidos com 120 alvos (60 de pistola e 60 de carabina), pode concluir-se que o sistema é capaz de classificar alvos de tiro desportivo com boa precisão em relação aos valores de referência. Os desvios obtidos são, no máximo de duas décimas. Assim, embora não cumpra os requisitos de um sistema de classificação oficial do ISSF, o sistema pode ser usado para classificação em treinos e provas de clube.

Palavras Chave: Processamento de Imagem, Deteção de Impactos, Raspberry Pi, OpenCV, Python

Abstract

Since the nineteenth century, the first competitions of shooting sports began to be organized. In these competitions, targets were manually classified.

With the technological evolution, many training sites already have automatic classification systems, but the price of these systems is too high to be widely used in the shooting clubs.

The most commonly used automatic classification method is based on the triangulation of the sound signal of the impact of the projectile on the target. However, in recent times, the potential of computer vision systems have been investigated as a cheap way of classifying targets, calculating the center of the impact of projectiles.

The system to be implemented in this dissertation should be available to any shooting club and easy to use. Due to problems in acquisition of camera images with sufficient sharpness of the target, we choose to use images with lower resolution, which were subjected to the algorithm developed, in order to calculate the center of the target and the center of the shoot, to assign the shooting rating.

Analyzing the results obtained with 120 targets, we were able to conclude the system classified correctly shooting targets. 50% of the targets has same results as the reference values. The remaining had a deviation of one tenth. Although it wasn't able to comply with the requirements of an ISSF official classification system, the system can be used for training and club competitions.

Keywords: Image Processing, Impact Detection, Raspberry Pi, OpenCV, Python.

Agradecimentos

Quero aproveitar este espaço para dirigir algumas palavras sinceras de gratidão a todos aqueles que possibilitaram a realização deste trabalho e que me acompanharam durante a minha vida académica.

Gostaria de agradecer, em primeiro lugar, ao Professor Armando Araújo, pelo apoio e disponibilidade demonstrados, não só durante este projeto, mas também ao longo do meu percurso académico.

Em segundo lugar, quero agradecer ao Professor Mário Sousa, pela amizade e disponibilidade, com que sempre me orientou, nas dúvidas que lhe fui colocando, ao longo do meu percurso académico.

Finalmente, gostaria de dar um especial obrigado à minha família que, ao longo dos anos, me apoiou, nos bons e maus momentos, e me deu a oportunidade de obter este grau de formação.

Muito obrigado a todos!

Eduardo Miguel Ribeiro Pereira Bártolo

“At its heart, engineering is about using science to find creative, practical solutions. It is a noble profession.”

Queen Elizabeth II

Índice

RESUMO.....	2
ABSTRACT.....	4
AGRADECIMENTOS	6
ABREVIATURAS E SÍMBOLOS.....	14
1.INTRODUÇÃO	15
1.1 Enquadramento.....	15
1.3 Objetivos	17
1.4. Metodologia	17
1.5. Estrutura da dissertação	18
2 REVISÃO BIBLIOGRÁFICA	21
2.1 - Sistemas eletrônicos de classificação de alvos baseados em visão.....	21
2.1.1 - Sistemas de medição baseados em visão	21
2.1.2 Artigos científicos	24
3. HARDWARE	26
3.1 Computador de Placa Única – SBC.....	26
3.2. Câmara	27
3.3 Alvos a classificar	28
3.3.1. Constituição do alvo	28
3.4 - Foco luz LED.....	31
3.5 Protótipo.....	32
4. SOFTWARE	36
4.1 – Open CV	36
4.1.1 - Threshold	36
4.1.2 Transformada de Hough	37
4.1.3 - Procura de Contornos	38
4.1.4- Blur	39

4.1.5 Findblob	42
4.1.6 CvtColor	42
4.2 Processamento de imagem	43
4.2.1 Determinação do Centro do Alvo	46
4.2.2 Classificação do disparo	48
5. RESULTADOS	53
5.1 Comparação e análise	53
5.1.1 Carabina de 10 Metros	53
5.1.2 Pistola 10 Metros	55
6 CONCLUSÃO E TRABALHO FUTURO	58
6.1 Conclusões	58
6.2 Trabalho Futuro	59
7 BIBLIOGRAFIA	60

Lista de Ilustrações

Ilustração 1-Sistema de classificação Orion	22
Ilustração 2- Sistema da Elite Score 10 M	23
Ilustração 3- Aplicativo TargetScan	23
Ilustração 4- <i>Raspberry Pi 3 model B</i>	26
Ilustração 5- Módulo PI NoIR Camera V2	27
Ilustração 6: Alvo de Carabina a 10 metros	28
Ilustração 7-Alvo de Pistola a 10 metros	30
Ilustração 8- Foco LED utilizado na iluminação dos alvos.	32
Ilustração 9- Protótipo do Sistema.	32
Ilustração 10 - Protótipo do Sistema – Pormenores.	33
Ilustração 11- Aplicação da técnica de Image Thresholding	37
Ilustração 12- Detecção de círculos usando a transformada de Hough	38
Ilustração 13- Exemplo do uso da função Findcountours	39
Ilustração 14 - Figura ilustrativa da função BLUR.	39
Ilustração 15- Figura ilustrativa da função GAUSSIAN BLUR.	40
Ilustração 16- Figura ilustrativa da função MEDIA BLUR	41
Ilustração 17-Exemplo da aplicação da função FindBlob	42
Ilustração 18- Exemplo da aplicação da função CvtColor	43
Ilustração 19- Fases do Processamento da Imagem	43
Ilustração 20-Fase da aplicação da função Imread	44
Ilustração 21.	Erro! Marcador não definido.
Ilustração 22- Aquisição da imagem através da função Imread.	44
Ilustração 23-Código da função Imread.	45
Ilustração 24- Fase da aplicação da função CvColor.	45
Ilustração 25- Aplicação da função <i>CvColor</i>	46
Ilustração 26-Código da função CvColor	46
Ilustração 27- Fase da aplicação da deteção do centro do alvo	47
Ilustração 28: Transformada de Hough aplicada a um alvo	47
Ilustração 29-Código da função utilizada para detectar o centro do alvo	48
Ilustração 30- Fase da aplicação da função Threshold.	49
Ilustração 31: Aplicação da função Threshold inversa	49
Ilustração 32-Código da função Threshold	49
Ilustração 33- Fase da aplicação da função FindBlob	50
Ilustração 34-Findblob aplicado ao alvo	51
Ilustração 35-Código da função FindBlob	51
Ilustração 36-Comparação dos resultados obtidos com a referência.	55
Ilustração 37 -Comparação dos resultados obtidos com a referência.	57

Lista de Tabelas

Tabela 1: Dimensões do alvo de precisão categoria C10	29
Tabela 2: Dimensões do alvo de precisão categoria P10	30
Tabela 3– Comparação da Pontuação – Protótipo VS Referência	54
Tabela 5 – Comparação da Pontuação – Protótipo VS Referência	56

Abreviaturas e Símbolos

ISSF- International Shooting Sport Federation
C10- Carabina 10 Metros
P10- Pistola 10 Metros

1.Introdução

1.1 Enquadramento.

Inicialmente o tiro desportivo teve raízes militares, como é visível no seguinte exemplo: as linhas de tiro utilizadas nos combates serviram como modelo para as primeiras competições. Em 1867, no Campo de Instrução de Châlons, em França, foi realizada uma prova de tiro ao alvo com fuzis [1].

Desde a primeira edição dos Jogos Olímpicos da Era Moderna, em 1896, o tiro desportivo faz parte das modalidades olímpicas. Em 1907 oito nações formaram a ISSF [2], organização responsável pela criação de regulamentos de tiro desportivo e pelo seu desenvolvimento.

Nos últimos trinta anos, a revolução digital acelerou o desenvolvimento dos sistemas de pontuação de tiro desportivo tendo levado à diminuição do tempo para a obtenção de pontuações, tornando o sistema de pontuação mais facilmente compreensível pelos espectadores.

1.2. Motivação

Quando foram realizadas as primeiras provas de tiro desportivo oficiais, nos jogos olímpicos, o sistema de classificação de alvos de tiro desportivo, manual, precisava de bastantes recursos humanos e logísticos para classificar, lentamente, os alvos. Devido a esse facto, a sua presença enquanto modalidade olímpica estava em risco, pois não

captava a atenção dos espectadores, já que existia um grande atraso entre o final das provas e a proclamação dos vencedores.

Nos últimos trinta anos, porém, a revolução digital permitiu o desenvolvimento de sistemas eletrônicos de pontuação, que tiveram a sua primeira implementação nos jogos olímpicos em 1988 [3].

Desde então, o tiro desportivo tornou-se mais apelativo para o público e de mais fácil divulgação pelos *media*, devido ao facto de as pontuações serem obtidas quase em tempo real.

Atualmente, existem diversos sistemas de pontuação automática, que vão desde sistemas acústicos a óticos e híbridos. Todavia a maioria desses sistemas são proprietários, logo, demasiado caros para serem aplicados nos clubes de tiro, tornando difícil a divulgação do tiro desportivo como modalidade acessível ao grande público.

Por isso, a implementação de uma solução de baixo custo, fiável e de fácil compreensão e manejo, que consiga ter aceitação pelos clubes locais de tiro, poderá ser fundamental para tornar o tiro desportivo um desporto mais espetacular e, assim, facilmente aceite pelos espectadores

Dessas premissas surgiu o desafio de desenvolver e implementar um protótipo baseado em visão, que satisfizesse todos os requisitos acima mencionados.

1.3 Objetivos

O sistema tem como seu objetivo principal a pontuação de alvos de tiro desportivo Carabina 10 Metros e Pistola 10 metros, de acordo com as regras da ISSF, através de um sistema baseado em visão por computador.

O sistema deveria fornecer ao utilizador a pontuação de cada disparo e a pontuação total.

O sistema a desenvolver deveria ser um sistema económico, comparativamente aos atualmente existentes no mercado, e ser facilmente implementado por qualquer membro num clube de tiro.

1.4. Metodologia

De maneira a pontuar os alvos no sistema baseado em visão por computador, no início, pensou-se captar imagens fotográficas do alvo e, através do desenvolvimento de um algoritmo que fosse adequado, classificar os alvos com precisão, de acordo com as regras do ISSF,

Porém as imagens obtidas não tinham nitidez suficiente, devido ao facto de a câmara NoIR Camera V2 não possuir Zoom, fotografando uma superfície maior do que o alvo; e, também, por não se obter imagens com uma iluminação uniforme, mesmo recorrendo a um foco de LEDs.

Para ultrapassar os problemas encontrados na obtenção das imagens, decidiu-se recorrer a imagens digitalizadas com menor resolução.

Nas imagens digitalizadas, com o algoritmo desenvolvido, detetou-se o centro de cada alvo, em seguida, detetou-se o centro dos disparos efetuados em cada alvo e calculou-se a distância do centro do disparo ao centro do alvo e, assim, atribuiu-se a pontuação de acordo com as regras da ISSF.

Finalmente, analisaram-se os resultados obtidos, comparando-os com os valores de referência da ISSF.

1.5. Estrutura da dissertação

O presente documento é constituído por seis capítulos.

No primeiro capítulo, *Introdução*, é descrito o enquadramento dos sistemas de tiro desportivo, a motivação que esteve por detrás do desenvolvimento do projeto, os objetivos e as metodologias utilizadas.

No segundo capítulo, *Revisão Bibliográfica*, são descritos os sistemas de pontuação de tiro desportivo presentes no mercado e a mais recente investigação feita sobre o tema.

No terceiro capítulo, *Hardware*, é analisado o sistema implementado para o desenvolvimento do projeto.

No quarto capítulo, *Software* são descritos os algoritmos relevantes utilizados para a execução do projeto.

No quinto capítulo, *Resultados*, são apresentados e analisados os resultados obtidos com o sistema implementado.

Finalmente, no sexto capítulo, *Conclusões e Trabalho Futuro*, reflete-se sobre o projeto desenvolvido e sobre os posteriores desenvolvimentos, a implementar em projetos futuros.

2 Revisão Bibliográfica

A revisão bibliográfica permite-nos a obtenção do conhecimento necessário para a realização da dissertação, dando-nos uma visão integrada sobre os conceitos, tecnologias e ferramentas relacionados com o tema.

Vão ser abordados conceitos teóricos inerentes a esta dissertação, começando por falar das principais tecnologias envolvidas na mesma.

2.1 - Sistemas eletrónicos de classificação de alvos baseados em visão

2.1.1 - Sistemas de medição baseados em visão

A maioria dos sistemas de medição baseados em visão não classificam os alvos em tempo real, sendo utilizados alvos de papel para registar a pontuação, alvos estes que, numa primeira fase, são registados em foto e, posteriormente, sujeitos a processamento de algoritmos proprietários, de maneira a obter a pontuação. Todavia, a empresa Elite Scorer comercializa sistemas baseados em visão com classificação em tempo real.

O Orion Scoring System [4] é um produto offline de software de computador que classifica as imagens digitalizadas de alvos de Carabina 10 metros muito mais rapidamente e com maior precisão do que o sistema de classificação manual.

Orion usa uma técnica de visão computacional conhecida como Visual Image Scoring (VIS). O VIS funciona interpretando imagens digitalizadas de alvos de papel, a que aplica uma série complexa de algoritmos para localizar os centros do alvo e os centros dos disparos, calculando as distâncias entre os centros do alvo e os centros dos disparos, para depois determinar as respectivas pontuações.

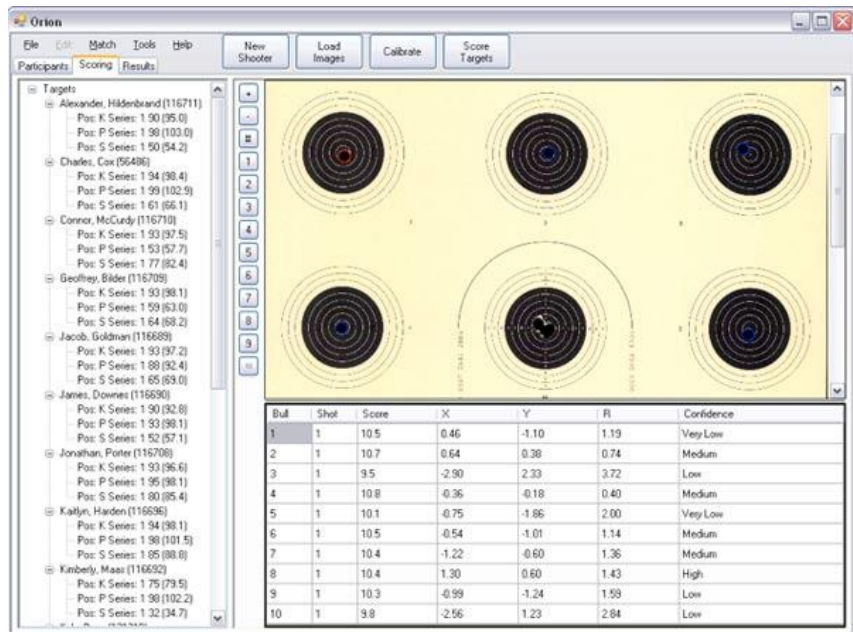


Ilustração 1-Sistema de classificação Orion [5].

O Elite Scorer 10 M [6] é o primeiro sistema eletrônico de pontuação wireless local sem necessidade de ligação à internet, construído para tiro de 10 metros.

O Elite Scorer 10 M utiliza um rolo de papel com 160 mm de largura para registrar os disparos, sendo que cada rolo de papel pode ser utilizado em 1200 tiros, assim minimizando o custo de operação.

Este sistema utiliza algoritmos avançados de processamento de imagem digital para pontuar o alvo (estando a patente pendente), sendo a

precisão do sistema de pontuação Elite Scorer 10 M de 0.1 mm.



Ilustração 2- Sistema da Elite Score 10 M [7].

O TargetScan [8] é um aplicativo offline de smartphone, que utiliza a câmara do mesmo para captar a imagem do alvo, sendo depois processada pelo seu algoritmo, que pontua o alvo e fornece a análise estatística dos disparos, permitindo ao atirador melhorar a eficácia no disparo.



Ilustração 3- Aplicativo TargetScan [9].

2.1.2 Artigos científicos

Dos vários artigos publicados sobre sistemas de classificação de alvos usando visão por computador, destacam-se os seguintes:

- Em 2008 Ali Faizan mostra as vantagens dos sistemas de classificação baseados em visão, mas, embora consiga uma excelente precisão, de 98,3% de alvos classificados corretamente, não efetua a classificação de acordo com as regras do ISSF (pontuação às décimas) [11].
- Em 2009 Xin Nan Fan apresenta um sistema de classificação baseado em visão de baixo custo, que é capaz de classificar tiros sobrepostos, conseguindo uma elevada precisão de classificação de alvos, pontuados corretamente, em tempo real [10].
- Em 2012 Rudzinski Jacek demonstra a facilidade de implementação dos sistemas baseados em visão, conseguindo obter uma excelente precisão, de 92% de alvos classificados corretamente, em tempo real [12].
- Em 2014 Cardim Guilherme demonstra a aplicação com sucesso de um sistema de classificação de alvos baseados em visão por computador para carreiras de tiro internas (não ao ar livre) com excelente percentagem de alvos classificados corretamente, de 97%. Todavia o software desenvolvido não efetua a classificação de acordo com as regras do ISSF (pontuação às décimas) [13].

3. Hardware

Neste capítulo são explicitadas as razões subjacentes à escolha do hardware utilizado no projeto e no desenvolvimento do protótipo

3.1 Computador de Placa Única – SBC

O computador escolhido para a integração neste sistema foi o *Raspberry Pi 3 model B* [14] [15] devido à sua velocidade de processamento permitir obter resultados em menos de 30 segundos porque possui um CPU quad-core ARM Cortex-A7 de 1200MHz e interface para o módulo PI NoIR Camera V2.



Ilustração 4- *Raspberry Pi 3 model B* [16]

3.2. Câmara

Como já foi anteriormente referido, a escolha recaiu no computador de placa única *Raspberry Pi*, compatível com módulo PI NoIR Camera V2 [17] para a captação da imagem.

O módulo NoIR Camera V2 apresenta uma câmara de 8MP, com possibilidade de tirar fotografias, com resolução de 3200 x 2400 pixels, pois consegue obter imagens com resolução suficiente para resolver a pontuação às décimas: 600x 600 pixels.

Contudo, no decorrer do projeto chegou-se à conclusão que as imagens obtidas pela câmara não tinham nitidez suficiente e, por isso, recorreu-se a imagens previamente digitalizadas, com menor resolução, para se classificar os alvos de tiro desportivo.

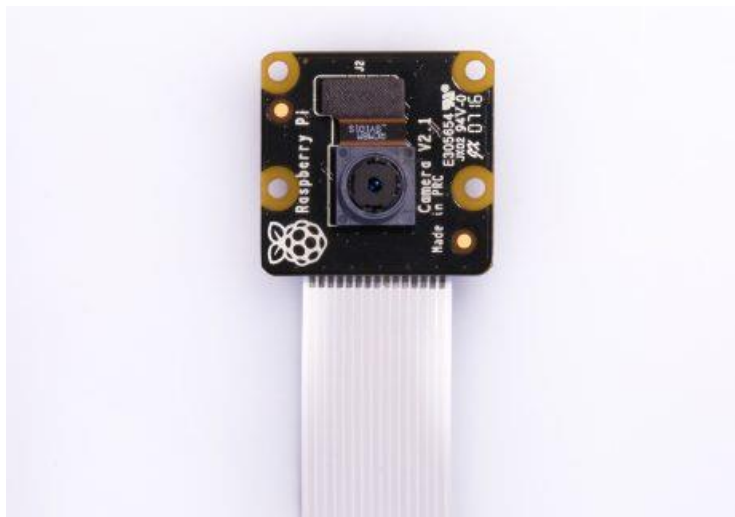


Ilustração 5- Módulo PI NoIR Camera V2 [18]

3.3 Alvos a classificar

3.3.1. Constituição do alvo

O alvo de precisão de carabina é constituído por 10 anéis, numerados de 1 a 10, com as regiões pontuáveis de 1 a 10, sendo que, nas finais, as pontuações vão até às décimas de 1.0 a 10.9. A ilustração 6 representa o alvo da categoria selecionada para análise.

Foi escolhido o alvo de precisão de Carabina 10 Metros para a realização do projeto.

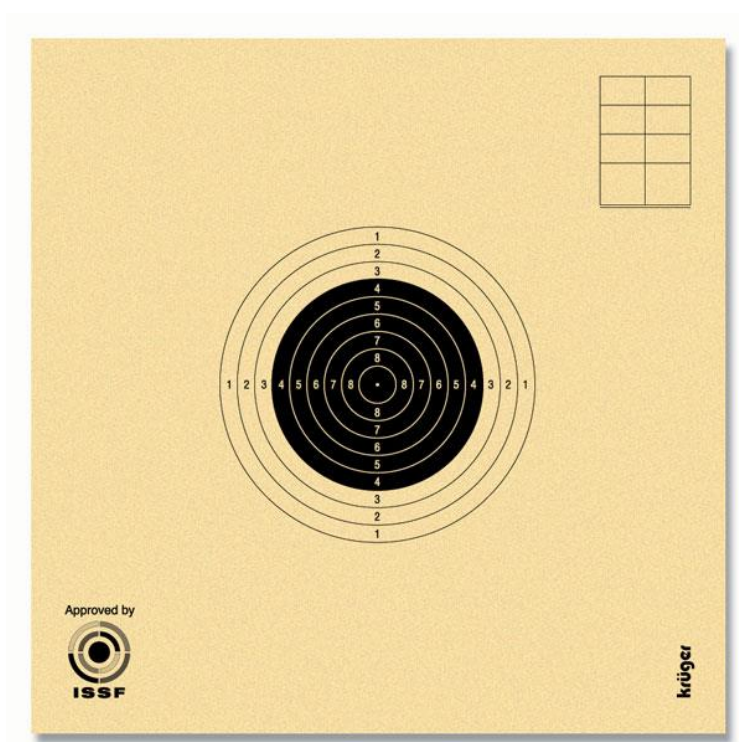


Ilustração 6: Alvo de Carabina a 10 metros [19]

A tabela 1 representa as dimensões associadas ao alvo utilizado nas categorias C10.

Tabela 1: Dimensões do alvo de precisão categoria C10

Anel do 10	0.5 mm	(±0.1 mm)	Anel do 5	25.5 mm	(±0.1 mm)
Anel do 9	5.5 mm	(±0.1 mm)	Anel do 4	30.5 mm	(±0.1 mm)
Anel do 8	10.5 mm	(±0.1 mm)	Anel do 3	35.5 mm	(±0.1 mm)
Anel do 7	15.5 mm	(±0.1 mm)	Anel do 2	40.5 mm	(±0.1 mm)
Anel do 6	20.5 mm	(±0.1 mm)	Anel do 1	45.5 mm	(±0.1 mm)

Tamanho do alvo: 80 mm x 80 mm (LxA)

Diâmetro do anel do 1: 45.5 mm

Resolução necessária: 45.5/200 mm

O alvo de precisão de pistola é constituído por 10 anéis, numerados de 1 a 10, com as regiões pontuáveis de 1 a 10, sendo que, nas finais, as pontuações vão até às décimas de 1.0 a 10.9. A ilustração 7 representa o alvo da categoria selecionada para análise.

Foi escolhido o alvo de precisão de Pistola 10 metros para a realização do projeto.



Ilustração 7-Alvo de Pistola a 10 metros [20]

A tabela 2 representa as dimensões associadas ao alvo utilizado nas categorias P10.

Tabela 2: Dimensões do alvo de precisão categoria P10

Anel do 10	11.5 mm	(±0.1 mm)	Anel do 5	91.5 mm	(±0.1 mm)
Anel do 9	27.5 mm	(±0.1 mm)	Anel do 4	107.5 mm	(±0.1 mm)
Anel do 8	43.5 mm	(±0.1 mm)	Anel do 3	123.5 mm	(±0.1 mm)
Anel do 7	59.5 mm	(±0.1 mm)	Anel do 2	139.5 mm	(±0.1 mm)
Anel do 6	75.5 mm	(±0.1 mm)	Anel do 1	155.5 mm	(±0.1 mm)

De forma a calcular-se a resolução mínima para cada um dos alvos, seguiu-se a seguinte linha de pensamento: no alvo de Pistola 10 metros, a distância entre anéis é de 8 mm. Temos de resolver em 10 partes, logo 0.8 mm. Daí resulta que para o alvo de Pistola 10 metros necessitamos de 1 pixel por cada .8 mm (valor teórico).

Para o alvo de Carabina 10 metros, seguindo a linha de pensamento anterior, temos a resolução mínima de .25 mm, logo 1 pixel por cada .25 mm.

Adicionalmente, através do teorema da amostragem [21] sabemos que são necessários, pelo menos, o dobro dos pixels anteriormente referidos. Deste modo, a câmara a usar deveria ter, no mínimo, uma resolução de pixels 8 pixels/mm (valor teórico).

3.4 - Foco luz LED

Neste sistema de aquisição de imagem o controlo da iluminação é fundamental. Uma boa qualidade de iluminação permite reduzir a complexidade dos processos de análise de imagem necessários à extração da informação pertinente [22] e [23].

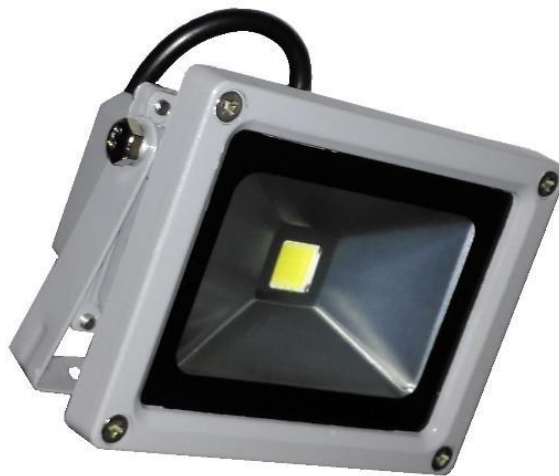


Ilustração 8- Foco LED utilizado na iluminação dos alvos.

Como tal foi utilizado um projetor *Light Emitting Diode* (LED), de 10 W, de maneira a obter iluminação uniforme no alvo. Porém, mesmo assim, continuaram a existir problemas de iluminação. Assim, desistiu-se de utilizar o projetor de LED e optou-se por recorrer a imagens previamente digitalizadas.

3.6 Protótipo



Ilustração 9- Protótipo do Sistema



Ilustração 10 - Protótipo do Sistema – Pormenores

A - Módulo PI NoIR Camera V2 , B - *Raspberry Pi 3 model B*, C - Tripé

As imagens, que se apresentaram, ilustram a materialização do protótipo utilizado para este projeto, testado *indoors*.

No início, o protótipo localizou-se na dianteira do alvo, abaixo deste, distando cerca de 30 cm do alvo. Todavia, com esta configuração as imagens obtidas não tinham nitidez suficiente e havia muitos pixels desperdiçados, contendo informação inútil para o projeto.

Com o intuito de melhorar a nitidez das imagens, aproximou-se o protótipo do alvo, distando o protótipo cerca de 5 cm do alvo, mas, mesmo assim, as imagens não tinham nitidez suficiente para classificar o alvo.

Por isso, decidiu-se recorrer a imagens previamente digitalizadas, de baixa resolução, para classificar os alvos de tiro desportivo

4. Software

4.1 – Open CV

Para o desenvolvimento do algoritmo do projeto recorreu-se ao *OpenCV* [24]

A *OpenCV* é uma biblioteca open source, que possui módulos de processamento de imagens, estrutura de dados e álgebra linear. Esta biblioteca foi desenvolvida nas linguagens de programação C/C++, dando suporte a programadores que utilizem Java, Python.

Dentro das linguagens de programação compatíveis com o Open CV a escolha recaiu sobre *Python* [25], devido à sua simplicidade e portabilidade e por possuir uma variedade de bibliotecas de visão computacional e inteligência artificial.

Todavia, no Python a execução do algoritmo é mais demorada, comparativamente com C++, porque sempre é efetuado o processamento da imagem, todo o código que invoca *OpenCV* é processado na sua linguagem nativa C++.[26].

4.1.1 - Threshold

O Threshold [27] é o método mais simples de segmentação de imagem.

A segmentação de imagem é uma tarefa essencial na área do processamento de imagem e visão computacional,

utilizada com o intuito de dividir, ou segmentar, uma imagem num determinado número de regiões, facilitando a sua análise e/ou tratamento.

O resultado desta técnica de processamento é uma imagem binária cujos pixels apresentam o valor de zero (preto) ou um (branco). A ilustração 11 pretende ilustrar a aplicação deste método.

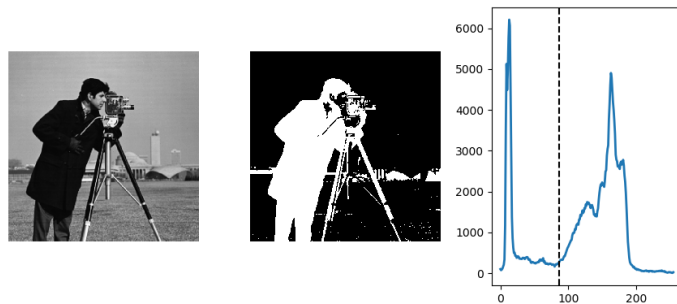


Ilustração 11- Aplicação da técnica de Image Thresholding [28]

4.1.2 Transformada de Hough

A Transformada de Hough é uma técnica matemática que realiza a deteção de formas geométricas em imagens digitais. A sua primeira conceção estava baseada na localização de retas. Posteriormente, a transformada de Hough foi estendida para possibilitar a localização de outras formas geométricas que possam ser parametrizadas, tais como círculos e elipses.

Tem como princípio a representação de um conjunto de pontos definidos inicialmente num espaço euclidiano num outro espaço, permitindo a deteção de pontos que constituem uma determinada figura geométrica [29].



Ilustração 12- Detecção de círculos usando a transformada de Hough [30]

Na Ilustração 12 pode ser visualizada a deteção do contorno circular presente na imagem, representado a verde.

4.1.3 - Procura de Contornos

Os contornos são uma ferramenta útil para análise de formas e deteção e reconhecimento de objetos [31].

Um contorno é uma curva contínua situada na parte exterior de um objeto. Este objeto distingue-se do fundo da imagem e, normalmente, tem a mesma cor ou intensidade por ele todo.

Na Ilustração 13 é possível visualizar o resultado da execução deste algoritmo

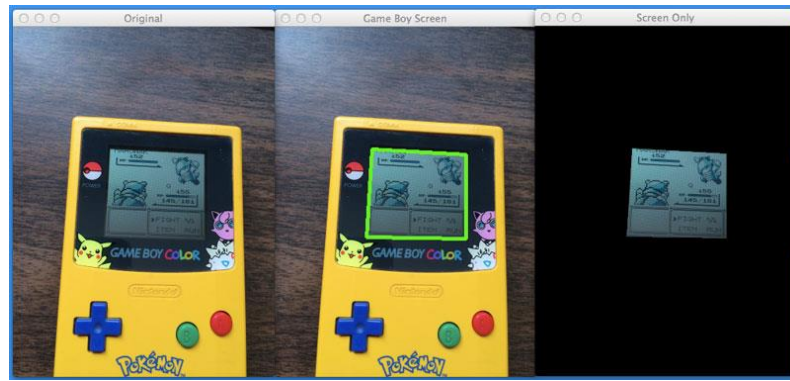


Ilustração 13- Exemplo do uso da função Findcountours [32].

4.1.4- Blur

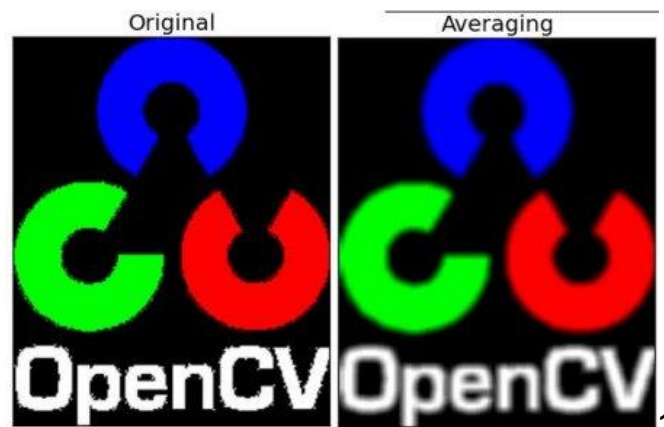
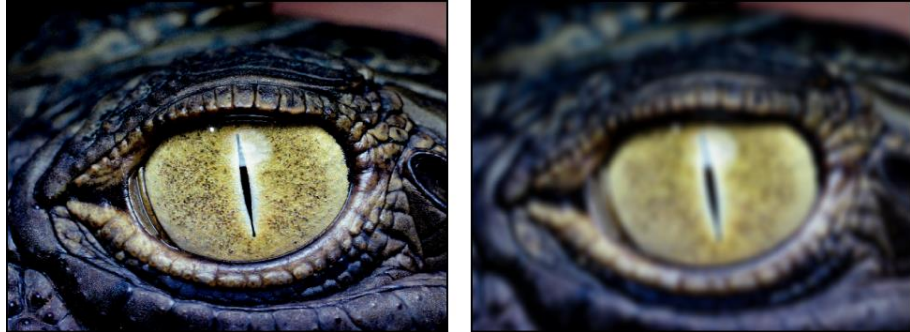


Ilustração 14 - Figura ilustrativa da função BLUR

.A função “Blur” [33] utiliza filtros passa baixo “LPF”, pois permite a passagem de baixas frequências e atenua a amplitude das frequências mais altas do que a frequência definida para corte.

Ao remover a frequência mais alta, estes filtros removem o ruído das imagens, obtendo-se, assim, imagens com os contornos mais esbatidos.

4.1.4.1 Gaussian blur



Original image

Gaussian Blur filter applied

Ilustração 15- Figura ilustrativa da função GAUSSIAN BLUR[34]

No processamento de imagens, a função “GAUSSIAN BLUR” [35] (também conhecida como suavização gaussiana) é o resultado do desfoque de uma imagem por uma função gaussiana

É um efeito amplamente utilizado no software gráfico, normalmente para reduzir o ruído da imagem e reduzir os detalhes.

O efeito visual dessa técnica de desfoque é um desfoque suave que lembra a visualização da imagem através de uma tela translúcida, distintamente diferente do efeito bokeh produzido por uma lente fora de foco ou a sombra de um objeto sob iluminação normal.

O alisamento gaussiano também é usado como um estágio de pré-processamento em algoritmos de visão computacional para melhorar as estruturas de imagem em diferentes escalas

4.1.4.2 - Filtro mediano

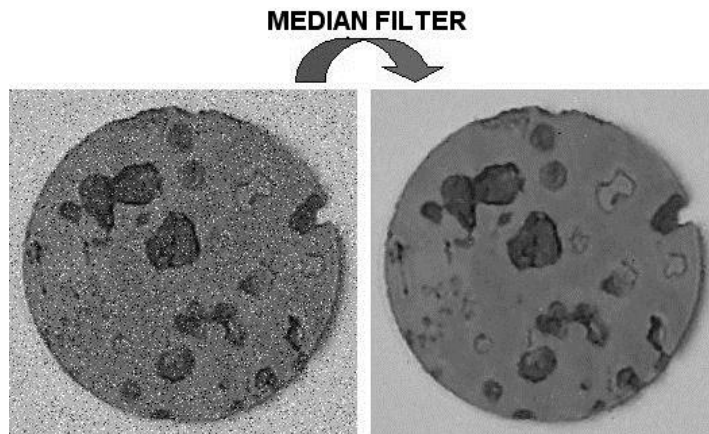


Ilustração 16- Figura ilustrativa da função MEDIA BLUR [36]

O filtro mediano [37] é uma técnica de filtragem digital não linear, usada frequentemente para remover o ruído de uma imagem ou sinal.

Essa redução de ruído é uma etapa típica de pré-processamento para melhorar os resultados do processamento posterior (por exemplo, detecção de borda numa imagem). A filtragem mediana é muito usada no processamento de imagens digitais porque, sob certas condições, preserva as bordas enquanto remove o ruído.

4.1.5 Findblob



Ilustração 17-Exemplo da aplicação da função FindBlob [38]

FindBlob é uma função que detecta blobs, grupos de pixels conectados numa imagem, que compartilham algumas propriedades comuns (por exemplo, o valor da escala de cinza). Na imagem acima, as regiões escuras conectadas são blobs, e o objetivo da detecção de blobs é identificar e marcar essas regiões [39].

4.1.6 CvtColor

A função CvtColor [40] converte uma imagem de entrada de um espaço de cores para outro. No caso de uma transformação para o espaço de cores RGB, a ordem dos canais deve ser especificada explicitamente

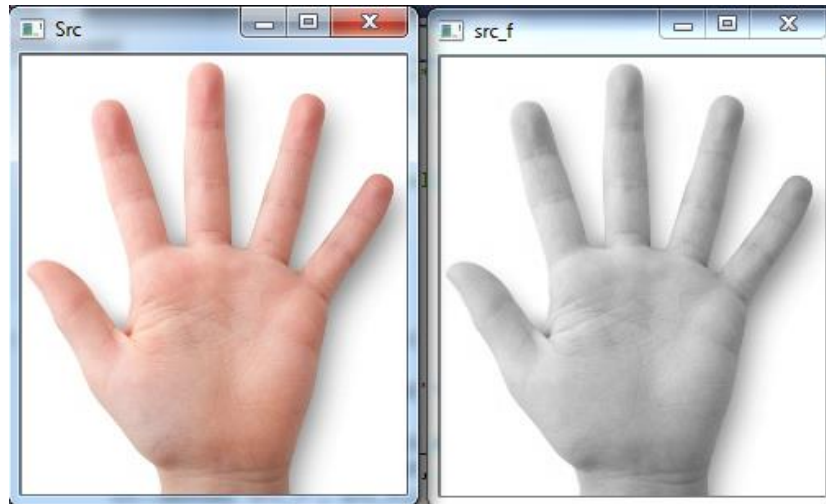


Ilustração 18- Exemplo da aplicação da função CvtColor [41]

4.2 Processamento de imagem

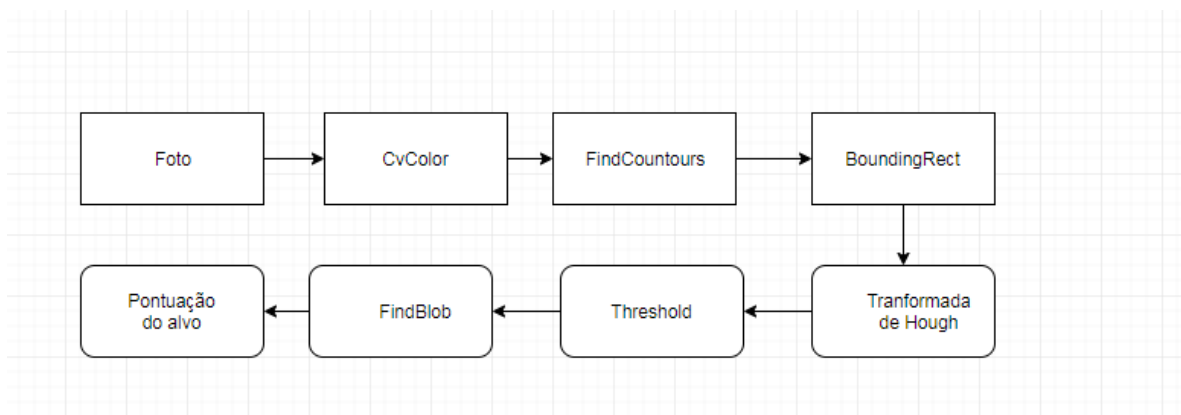


Ilustração 19- Fases do Processamento da Imagem

O processamento de imagem visa localizar o centro do alvo, localizar os centros dos disparos e calcular a distância entre o centro do disparo e o centro do alvo através de uma sequência de funções aplicados à imagem.

- **Imread**

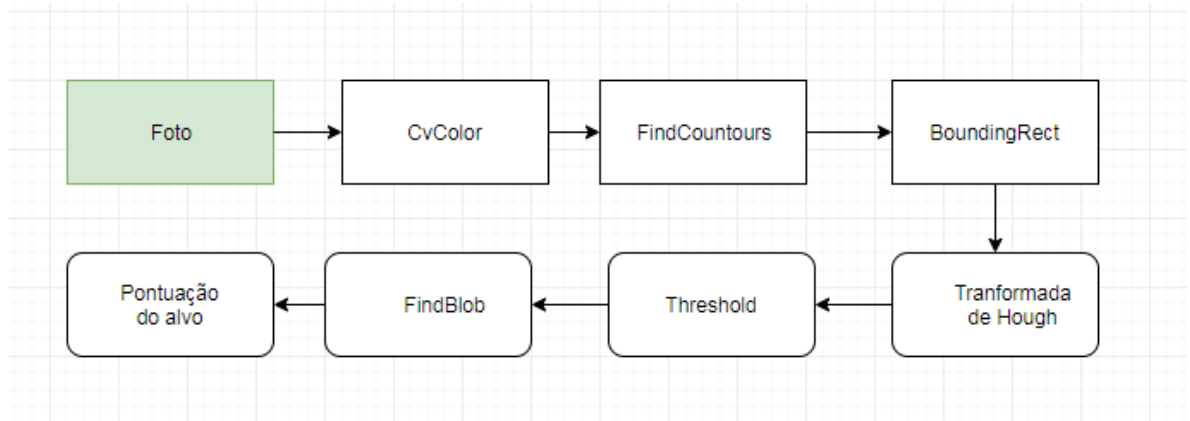


Ilustração 20-Fase da aplicação da função imread

Realizou-se a leitura de imagens através da função imread, permitindo o seu posterior processamento no decorrer do programa.

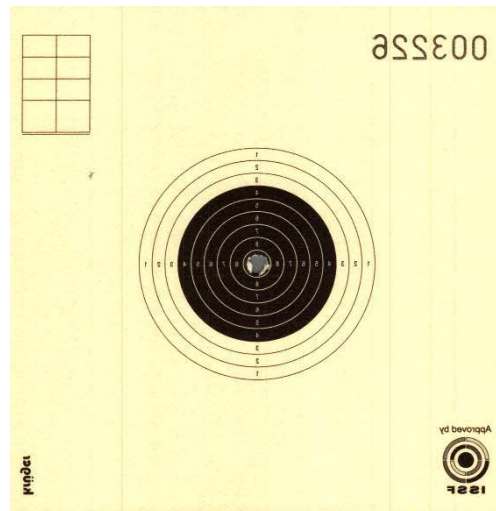


Ilustração 21- Aquisição da imagem através da função imread

Seguidamente, será feita a apresentação do código da função Imread:

```
# import the necessary packages
import sys
sys.path.append('/usr/local/lib/python3.4/site-packages')

# import the necessary packages
import argparse
import numpy as np
import imutils
import cv2

image = cv2.imread('/home/pi/Desktop/Tese/CCF_003076_p10.jpg')
```

Ilustração 22-Código da função Imread

• CvtColor

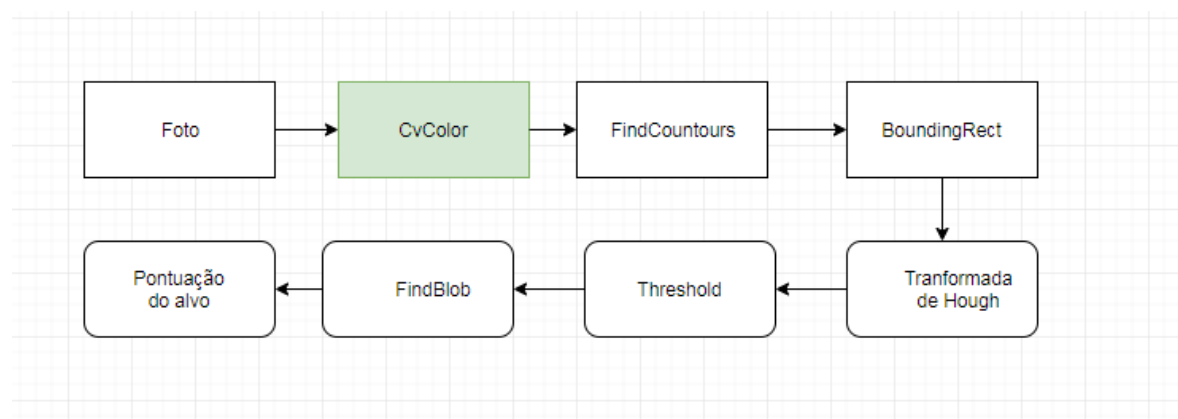


Ilustração 23- Fase da aplicação da função CvtColor

Efetuiu-se a conversão da imagem do modelo de cores RGB para grayscale com a função CvtColor, resultando numa imagem em tons de cinza.

Esta transformação irá facilitar a aplicação das funções findcountours, boundingRect e threshold.



Ilustração 24- Aplicação da função *CvtColor*

O código da função *CvtColor* é o seguinte:

```
gray = cv2.cvtColor(orig, cv2.COLOR_BGR2GRAY)
```

Ilustração 25-Código da função *CvtColor*

4.2.1 Determinação do Centro do Alvo

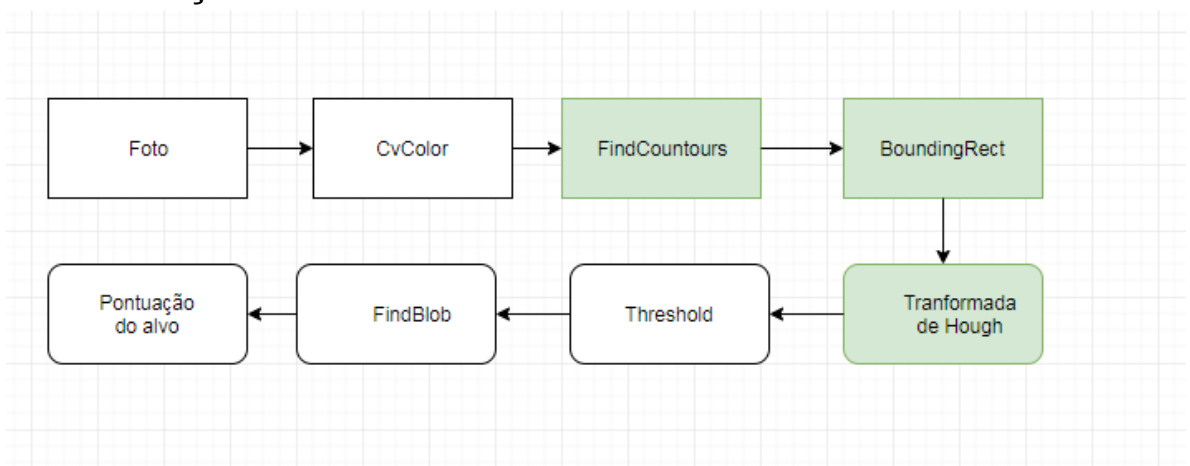


Ilustração 26- Fase da aplicação da deteção do centro do alvo

Os pixels que constituem os contornos detetados são guardados num vetor de pontos, onde cada ponto corresponde à posição de um pixel na matriz da imagem.

Após a obtenção dos contornos visíveis na imagem com a função `findcountours`, é utilizada a função `boundingRect`, de forma a determinar os pontos importantes para o cálculo do centro do alvo. A função `boundingRect` define os contornos através de uma caixa.

A implementação conjunta das duas funções retornou valores próximos dos que seriam esperados.



Ilustração 27: Transformada de Hough aplicada a um alvo

O código aplicado para detetar o centro do alvo é assim representado:


```

im2, contours, hierarchy = cv2.findContours(th1, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
cv2.drawContours(th1, contours, -1, (0,255,0), 3)

cnt = contours[0]
hull = cv2.convexHull(cnt)

x,y,w,h = cv2.boundingRect(cnt)
cv2.rectangle(th1,(x,y),(x+w,y+h),(0,255,0),2)
icwit=x+w/2
ichei=y+h/2
centrox= int(round(icwit))
centroy= int(round(ichei))

print('centro', icwit, ichei)

circles = cv2.HoughCircles(th1,cv2.HOUGH_GRADIENT,1.2,100)
centrox= int(round(icwit))
centroy= int(round(ichei))

circles = np.uint16(np.around(circles))
for i in circles[0,:]:

    # draw the outer circle
    cv2.circle(orig,(centrox,centroy), i[2],[0,255,0],2)
    # draw the center of the circle
    cv2.circle(orig,(centrox,centroy),1,[0,0,255],3)
print('centro', icwit, ichei)
print('raio', i[2]*0.25)
cv2.imwrite('/home/pi/Desktop/circles.jpg',orig)

```

Ilustração 28-Código da função utilizada para detectar o centro do alvo

4.2.2 Classificação do disparo

• Threshold inverso

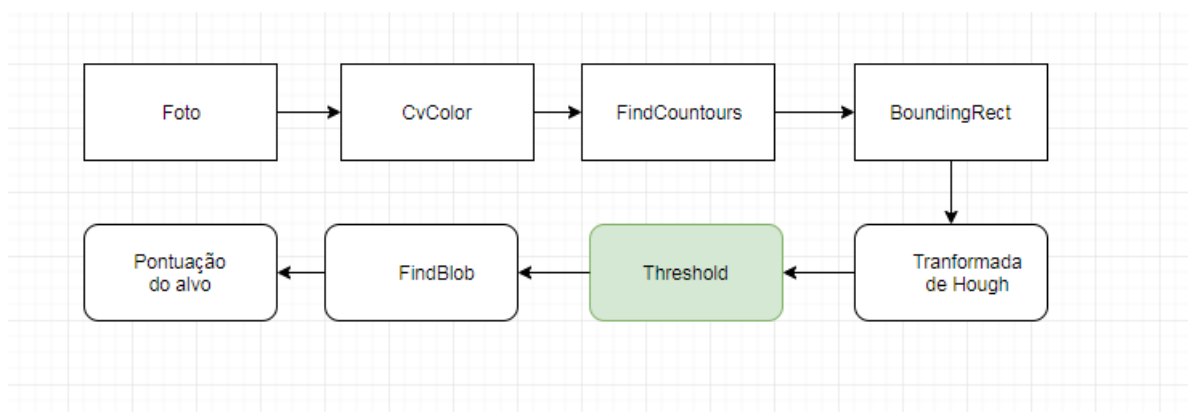


Ilustração 29- Fase da aplicação da função Threshold

Após a deteção do centro do alvo, segmentou-se a imagem com recurso à função threshold, com o objetivo de isolar o disparo, evidenciado a branco na ilustração 30.



Ilustração 30: Aplicação da função Threshold inversa

Código da função Threshold:

```
#threshold mais agressivo para apenas detetar os tiros
gray1 = cv2.cvtColor(orig, cv2.COLOR_BGR2GRAY)
cv2.imwrite('/home/pi/Desktop/gray1.jpg', gray1)
ret, th2 = cv2.threshold(gray, 85, 255, cv2.THRESH_BINARY)
cv2.imwrite('/home/pi/Desktop/th2.jpg', th2)
```

Ilustração 31-Código da função Threshold

- Findblob

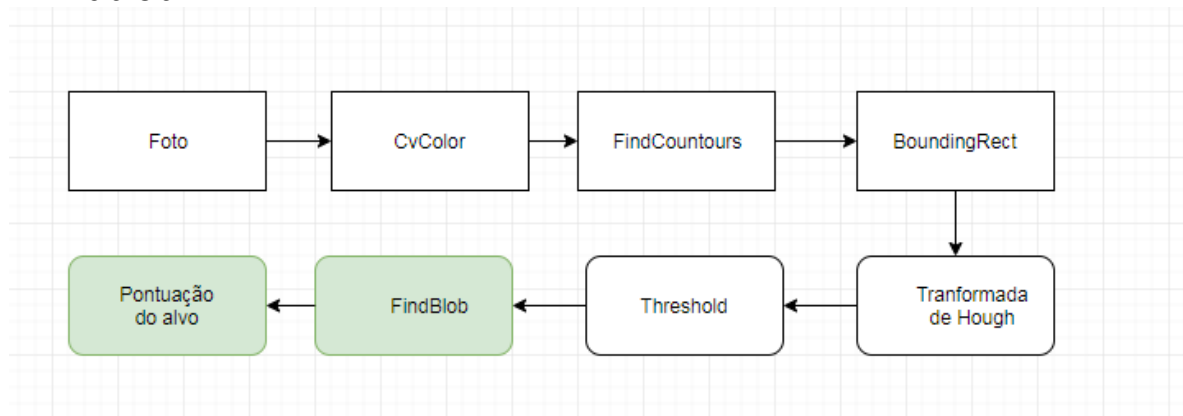


Ilustração 32- Fase da aplicação da função FindBlob

Findblob foi a função utilizada para detetar a marca do disparo.

Foram ajustados os parâmetros do algoritmo, tais como a área mínima e máxima e a cor do blob, de maneira a identificar as marcas causadas pelos disparos.

Após a deteção do centro do alvo e do centro dos disparos, calculou-se a distância (em mm) entre o centro do alvo e os centros dos disparos e através da implementação de uma função matemática linear calculou-se a pontuação dos disparos, de acordo com os regulamentos da ISSF:

$$\text{Pontuação_C10} = 11 - 0.4 * \text{Distância} \text{ e } \text{Pontuação_P10} = 11 - 0.125 * \text{Distância}$$

5. Resultados

Neste capítulo serão discutidos os resultados obtidos com as imagens de baixa resolução, previamente digitalizadas, a que foram aplicadas as funções anteriormente referidas. Esclarece-se que, já na fase final do trabalho, aplicamos as mesmas funções a imagens digitalizadas com melhor resolução, tendo os resultados obtidos sido muito semelhantes aos anteriores.

5.1 Comparação e análise

5.1.1 Carabina 10 Metros

As referências a seguir indicadas são as obtidas com o sistema de pontuação eletrónico, de forma a poder estabelecer uma comparação e análise entre os resultados a que chegamos com o sistema implementado e os do sistema de pontuação eletrónico.

Tabela 3 – Comparação da Pontuação – Protótipo VS Referência

Disparo	P1	P2	P3	P4	P5	P6	P7	P8
Protótipo	10.7	10.4	10.9	9.4	10,3	8.5	9,9	9.1
Referência	10.7	10.5	10.8	9.4	10.4	8.5	9.8	9.1
Diferença	0	-0.1	0.1	0	-0.1	0	0.1	0
Disparo	P9	P10	P11	P12	P13	P14	P15	P16
Protótipo	10.5	9.7	9.0	10.4	7.9	7.5	9.1	9.7
Referência	10.5	9.8	9.0	10.5	7.9	7.6	9.1	9.6
Diferença	0	-0.1	0	-0.1	0	-0.1	0	0.1
Disparo	P17	P18	P19	P20	P21	P22	P23	P24
Protótipo	10.3	10.3	10	9.8	9.9	9.5	8.9	9.7
Referência	10.3	10.2	10	9.9	9.9	9.4	8.8	9.7
Diferença	0	0.1	0	-0.1	0	0.1	0.1	0
Disparo	P25	P26	P27	P28	P29	P30	P31	P32
Protótipo	10.1	10.2	10	10.5	9.4	9.2	10.5	7.6
Referência	10.1	10	10	10.6	9.4	9.1	10.5	7.5
Diferença	0	0.2	0	-0.1	0	0.1	0	0.1
Disparo	P33	P34	P35	P36	P37	P38	P39	P40
Protótipo	9.5	9.1	9.1	10.3	10.5	9.7	10	10
Referência	9.5	9	9.1	10.4	10.4	9.8	10	10
Diferença	0	0.1	0	-0.1	0.1	-0.1	0	0
Disparo	P41	P42	P43	P44	P45	P46	P37	P48
Protótipo	10.4	9.5	9.3	9	9.8	9.7	8.8	9.3
Referência	10.4	9.3	9.4	9	9.6	9.7	8.7	9.3
Diferença	0	0.2	0.1	0	0.2	0	0.1	0
Disparo	P49	P50	P51	P52	P53	P54	P55	P56
Protótipo	10.3	10.4	8.7	9.2	9.7	10.4	9.5	10.8
Referência	10.3	10.3	8.7	9.3	9.7	10.5	9.5	10.7
Diferença	0	0.1	0	-0.1	0	-0.1	0	0.1
Disparo	P57	P58	P59	P60				
Protótipo	7.7	9.4	9.6	9.5				
Referência	7.8	9.4	9.6	9.4				
Diferença	-0.1	0	0	0.1				

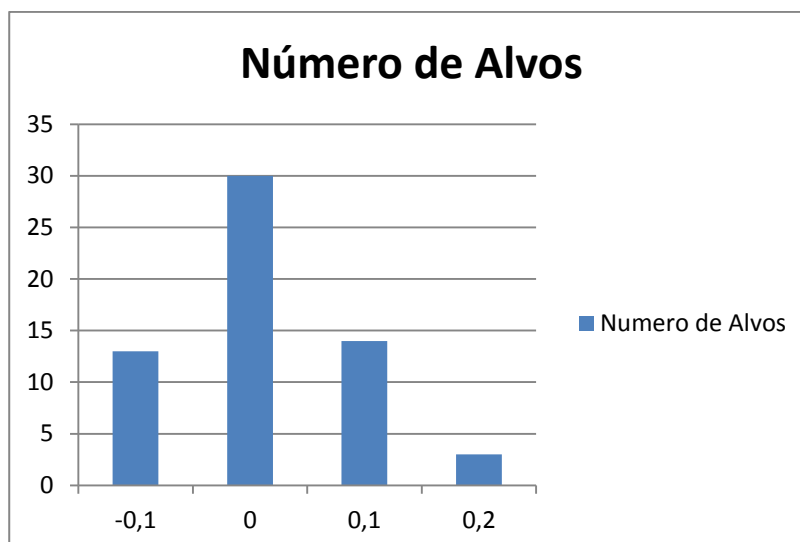


Ilustração 35-Comparação dos resultados obtidos com a referência

É possível constatar que, em 30 dos 60 alvos avaliados, a pontuação atribuída pelo programa foi idêntica à da referência. Em 27 ocorreu um desvio de uma décima (quer acima ou abaixo da referência). Nos restantes 3 o desvio correspondeu a duas décimas.

5.1.2 Pistola 10 Metros

As referências a seguir indicadas são as obtidas com o sistema de pontuação eletrónico, de forma a poder estabelecer uma comparação e análise entre os resultados a que chegamos com o sistema implementado e os do sistema de pontuação eletrónico.

Tabela 4 – Comparação da Pontuação – Protótipo VS Referência

Disparo	P1	P2	P3	P4	P5	P6	P7	P8
Protótipo	10.5	10.9	10.9	10.8	10,8	9.8	10.9	10.5
Referência	10.6	10.9	10.8	10.8	10.7	9.8	10.8	10.5
Diferença	-0.1	0	0.1	0	0.1	0	0.1	0
Disparo	P9	P10	P11	P12	P13	P14	P15	P16
Protótipo	6.2	10.7	10.9	6.4	10.3	8.8	6.8	6.8
Referência	6.2	10.6	10.9	6.5	10.3	8.9	6.8	6.8
Diferença	0	0.1	0	-0.1	0	-0.1	0	0
Disparo	P17	P18	P19	P20	P21	P22	P23	P24
Protótipo	8.5	9.7	10.2	9.5	10.1	8.0	10.7	9.6
Referência	8.6	9.6	10.2	9.4	10.1	8.1	10.7	9.7
Diferença	-0.1	0.1	0	0.1	0	-0.1	0	-0.1
Disparo	P25	P26	P27	P28	P29	P30	P31	P32
Protótipo	10.0	9.9	10.3	6.5	9.2	8.8	8.6	8.7
Referência	10.0	9.8	10.3	6.4	9.2	8.7	8.6	8.8
Diferença	0	0.1	0	0.1	0	0.1	0	-0.1
Disparo	P33	P34	P35	P36	P37	P38	P39	P40
Protótipo	9.7	7.7	9.3	10.8	8.6	9.4	9.4	9.7
Referência	9.7	7,8	9.3	10.9	8.6	9.5	9.4	9.8
Diferença	0	0.1	0	-0.1	0	-0.1	0	-0.1
Disparo	P41	P42	P43	P44	P45	P46	P47	P48
Protótipo	8.1	6.6	9	10.1	10.2	9.4	7.4	10.5
Referência	8.1	6.4	9	10.2	10.2	9.3	7.4	10.4
Diferença	0	0.2	0	-0.1	0	0.1	0	0.1
Disparo	P49	P50	P51	P52	P53	P54	P55	P56
Protótipo	9.8	7.6	4.4	9.6	8.6	9	10.3	6.7
Referência	9.8	7.5	4.4	9.5	8.6	8.9	10.3	6.8
Diferença	0	0.1	0	0.1	0	0.1	0	-0.1
Disparo	P57	P58	P59	P60				
Protótipo	8	8.2	8.5	7.9				
Referência	8.1	8.2	8.3	7.9				
Diferença	-0.1	0	0.2	0				

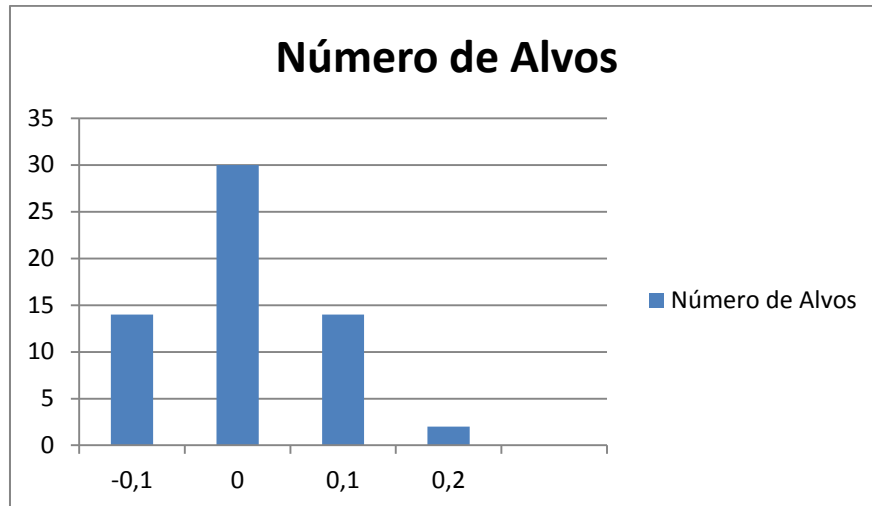


Ilustração 36 -Comparação dos resultados obtidos com a referência

Também neste caso foi possível constatar que, em 30 dos 60 alvos avaliados, a pontuação atribuída pelo programa foi idêntica à da referência. Em 28 ocorreu um desvio de uma décima (quer acima ou abaixo da referência). Nos restantes 2 o desvio correspondeu a duas décimas.

6 Conclusões e Trabalho Futuro

6.1 Conclusões

Esta dissertação cumpriu os objetivos: o desenvolvimento de um sistema de pontuação de um alvo de tiro desportivo, com visão de baixo custo, acessível a qualquer clube de tiro

Todavia o sistema não cumpre os regulamentos exigidos pela ISSF (International Shooting Sport Federation), por não conseguir a classificação decimal. Logo, o sistema implementado não pode ser utilizado em competições, embora seja muito útil para o treino dos atiradores e provas de clube.

No início do projeto, foi realizada uma pesquisa dos sistemas de pontuação baseados em visão disponíveis no mercado e dos artigos publicados sobre o tema.

De seguida, descreveu-se o hardware e o software necessários para o desenvolvimento dum protótipo, que cumprisse os objetivos traçados.

Na fase seguinte, implementaram-se e testaram-se o hardware e os algoritmos, que posteriormente iriam classificar os alvos, que tinham sido pontuados por um sistema automático de classificação.

Todavia no decorrer do teste do hardware chegou-se à conclusão que as imagens obtidas pela câmara utilizada não tinham nitidez suficiente para classificar os alvos e, por isso, prescindimos da câmara e recorreremos a imagens previamente

digitalizadas de baixa resolução para se classificar os alvos de tiro desportivo

Embora os resultados tenham sido satisfatórios, o sistema poderá ser sujeito a melhorias, que serão discutidas na secção seguinte.

6.2 Trabalho Futuro

Como referido, o sistema implementado poderá ser melhorado em alguns pontos que passo a referenciar:

- Desenvolvimento de um aplicativo interface *online*: onde estejam disponíveis as pontuações de cada atirador e as estatísticas dos disparos efetuados. Assim poderia tornar-se o tiro desportivo mais apelativo para ao público;
- Sistema de deteção de impactos em tempo real: tornar o sistema de classificação mais realista e, assim, não existir um grande atraso entre o momento do disparo e o momento que a classificação é mostrada ao público
- Implementação de uma base de dados: agregar toda a informação num único sítio, num servidor acessível a cada utilizador, caso possua uma conta válida. Permitiria, também, realizar estatísticas sobre os disparos efetuados.

7 Bibliografia

[1] Tiro esportivo

Disponível em :<http://www.brasil2016.gov.br/pt-br/megaeventos/olimpiadas/modalidades/tiro-esportivo>

[Acedido em Março 2018]

[2] ISSF, s.d. *ISSF - International Shooting Sport Federation*.

Disponível em: www.issf-sports.org

[Acedido em Março 2018]

[3] How Olympic Shooting Competitions are Organized and What This Means for Youth Shooting

Disponível em:: http://thecmp.org/wp-content/uploads/Olympic-Organization_web.pdf

[Acedido em Março 2018]

[4] Orion Scoring System

Disponível em:: <http://www.orionscoringsystem.com/orion/Home.aspx>

[Acedido em Março 2018]

[5] Orion Scoring System

Disponível em:: http://www.odcmp.org/0110/images/OrionImg/Verify_s.jpg

[Acedido em Março 2018]

[6] Elite Socorrer.

Disponível em: <http://www.elitescorer.com/>.

[Acedido em Março 2018]

[7] Elite Scorer 10M

Disponível em:

<https://www.facebook.com/EliteScorer/photos/a.509346189122208.1073741830.457847044272123/611222458934580/?type=3&theater>

[Acedido em Março 2018]

[8] TargetScan

Disponível em:

https://targetshootingapp.com/?utm_source=Kickstarter&utm_medium=www&utm_campaign=Link

[Acedido em Março 2018]

[9] TargetScan

Disponível em: https://ksr-ugc.imgix.net/assets/011/495/243/2160353c0d0b3343a744f027a08b8a80_original.jpg?crop=faces&w=1552&h=873&fit=crop&v=1463683575&auto=format&q=92&s=bdaa4f031bbde132767df92af46b101f

[Acedido em Março 2018]

[10] Xin Nan Fan “Design of Automatic Target-Scoring System of Shooting Game Based On Computer Vision ”

Disponível em:

https://www.researchgate.net/publication/251898706_Design_of_Automatic_Target-Scoring_System_of_Shooting_Game_Based_On_Computer_Vision

[Acedido em Março 2018]

[11] A. Faizan, Computer Vision based Automatic Scoring of shooting targets, Multitopic Conference, 2008. INMIC 2008. IEEE International.

Disponível em :

https://www.researchgate.net/publication/238513937_Computer_Vision_based_Automatic_Scoring_of_shooting_targets

[Acedido em Março 2018]

[12] J. Rudzinski, " Low–Cost Computer Vision Based Automatic Scoring of Shooting Targets " Paper presented at the meeting of the KES, 2012. Disponível em

[https://www.researchgate.net/publication/262272062_Low-](https://www.researchgate.net/publication/262272062_Low-Cost_Computer_Vision_Based_Automatic_Scoring_of_Shooting_Targets)

[Cost_Computer_Vision_Based_Automatic_Scoring_of_Shooting_Targets](https://www.researchgate.net/publication/262272062_Low-Cost_Computer_Vision_Based_Automatic_Scoring_of_Shooting_Targets)

[Acedido em Março 2018]

[13] G. Cardim, M. Dias, R. Noguti, R. Best, E. Silva, “Mathematical Morphology Applied to Automation of Indoor Shooting Ranges”. Disponível em

<http://www.diogenes.bg/ijam/contents/2014-27-6/3/3.pdf>.

[Acedido em Março 2018]

[14] Raspberry-Pi-3-Model-b

<https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>
[Acedido em Março 2018]

[15] Anon., s.d. *Raspberry Pi - Teach, Learn, and Make with Raspberry Pi*.
Disponível em : <https://www.raspberrypi.org/products/pi-noir-camera-v2/>
[Acedido em Março 2018]

[16] Raspberry-Pi-3-Model-b
Disponível em :https://www.modmypi.com/image/cache/data/rpi-products/raspberry-pi-3-model-b/DSC_0303-800x609.jpg
[Acedido em Março 2018]

[17] Módulo PI NoIR Camera V2
Disponível em :<https://www.raspberrypi.org/products/pi-noir-camera-v2/>
[Acedido em Março 2018]

[18] Módulo PI NoIR Camera V2
Disponível em : <https://www.raspberrypi.org/products/pi-noir-camera-v2/>
[Acedido em Março 2018]

[19] Alvo Carabina 10 metros
Disponível em: [https://targets.krueger-shops.eu//sch_en_articles.php?VID=RELdyq2asYiwObXb&saSearch\[category\]=Air+ifle+10+m](https://targets.krueger-shops.eu//sch_en_articles.php?VID=RELdyq2asYiwObXb&saSearch[category]=Air+ifle+10+m)
[Acedido em Março 2018]

[20] Alvo Pistola 10 metros
Disponível em: [https://targets.krueger-shops.eu//sch_en_articles.php?VID=RELdyq2asYiwObXb&saSearch\[category\]=Air+pistol+10+m](https://targets.krueger-shops.eu//sch_en_articles.php?VID=RELdyq2asYiwObXb&saSearch[category]=Air+pistol+10+m)
[Acedido em Março 2018]

[21] Teorema da amostragem:
Disponível em:https://pt.wikipedia.org/wiki/Teorema_da_amostragem_de_Nyquist%E2%80%933_Shannon
[Acedido em Março 2018]

[22] G.W. Awcock, R. Thomas: Applied Image Processing, McGraw-Hill International Editions, 1996, ISBN: 0-07-001470-1

[Acedido em Março 2018]

[23] C. Steger, M. Ulrich, C. Wiedemann: Machine Vision Algorithms and Applications (Wiley-VCH), 2008. ISBN: 978-3-527-40734-7

[Acedido em Março 2018]

[24] Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library.

Disponível em

:www.bogotobogo.com/cplusplus/files/OReilly%20Learning%20OpenCV.pdf.

[Acedido em Março 2018]

[25] Langtangen, H. P., 2008. *Python Scripting for Computational Science*. Noruega: Springer.

[Acedido em Março 2018]

[26] Anon., s.d. *Learn OpenCV (C++ / Python)*. [Online] Available at:

<https://www.learnopencv.com/blob-detection-using-opencv-python-c/>

[Acedido em Maio 2018].

[27] Threshholding

Disponível em:

<https://www.wavemetrics.com/products/igorpro/imageprocessing/threshholding.htm>

[Acedido em Março 2018]

[28] Image Thresholing

Disponível em: [http://www.scipy-](http://www.scipy-lectures.org/images/sphx_qlr_plot_threshold_001.png)

[lectures.org/images/sphx_qlr_plot_threshold_001.png](http://www.scipy-lectures.org/images/sphx_qlr_plot_threshold_001.png)

[Acedido em Março 2018]

[29] The Hough Transform Disponível em: <http://aishack.in/tutorials/hough-transform-basics/>

[Acedido em Março 2018]

[30] Detecting Circles Using OpenCV

Disponível em: [http://www.pyimagesearch.com/2014/](http://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/)

[07/21/detecting-circles-images-using-opencv-hough-circles/](http://www.pyimagesearch.com/2014/07/21/detecting-circles-images-using-opencv-hough-circles/).

[Acedido em Março 2018]

[31] Contours : Getting Started

Disponível em:

https://docs.opencv.org/3.3.1/d4/d73/tutorial_py_contours_begin.html

[Acedido em Março 2018]

[32] Findcountours

Disponível em: <https://www.pyimagesearch.com/wp-content/uploads/2014/03/finding-the-gameboy-screen.jpg~>

[Acedido em Março 2018]

[33] *OpenCV*, s.d. *OpenCV: Smoothing Images*.

Disponível em: https://docs.opencv.org/3.1.0/d4/d13/tutorial_py_filtering.html

[Acedido em Março 2018]

[34] Gaussian_blur

Disponível em:

https://support.apple.com/library/APPLE/APPLECARE_ALLGEOS/Product_Help/en_US/PUBLIC_USERS/133281/S0447_Gaussian.png

[Acedido em Março 2018]

[35] Gaussian_blur

Disponível em:

<https://docs.opencv.org/2.4/modules/imgproc/doc/filtering.html?highlight=gaussianblur#gaussianblur>

[Acedido em Março 2018]

[36] Median_filter

Disponível em:

<http://tracer.lcc.uma.es/problems/mfp/MFExample.jpg>

[Acedido em Março 2018]

[37] Median_filter

Disponível em:

<https://docs.opencv.org/2.4/modules/imgproc/doc/filtering.html?highlight=medianblur#medianblur>

[Acedido em Março 2018]

[38] Findblob

Disponível em: <https://stackoverflow.com/questions/21217506/robust-tracking-of-blobs>

[Acedido em Março 2018]

[39] Anon., s.d. *Learn OpenCV (C++ / Python)*.

Disponível em: <https://www.learnopencv.com/blob-detection-using-opencv-python-c/>

[Acedido em Março 2018]

[40] CvtColor

Disponível em:

https://docs.opencv.org/2.4/modules/imgproc/doc/miscellaneous_transformations.html?

[Acedido em Março 2018]

[41] Exemplo da aplicação da função CvtColor

Disponível em: <http://answers.opencv.org/question/11529/displaying-mat-image-in-android-screen/>

[Acedido em Março 2018]