FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Real time, image based, target shooting scoring system

Guilherme Pinto Ferreira Novo



Mestrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Armando Luís Sousa Araújo

July 29, 2024

© Guilherme Pinto Ferreira Novo, 2024

Resumo

Esta dissertação propõe um sistema de pontuação automática de tiro de baixo custo e de alta precisão. Utiliza técnicas de processamento de imagem para analisar alvos em tempo real. No tiro desportivo, os sistemas de pontuação tradicionais exigem muito tempo e recursos e, como tal, a automação é necessária para eficiência e precisão. O sistema proposto utiliza uma câmara posicionada de modo a não interferir na linha de tiro do atirador, que captura cada frame. Em seguida, deteta e pontua com precisão os impactos usando algoritmos de processamento de imagens, como processamento morfológico de imagem, transformação de perspectiva e deteção de anéis.

Este sistema destina-se ao uso prático em ambientes internos oficiais onde uma moldura e fita preta podem ser usadas como alvo para uma solução mais acessível, ao contrário dos sistemas existentes que são de custo elevado. É apropriado para atiradores amadores que procuram precisão de nível profissional sem a necessidade de equipamento adicional. O algoritmo é desenvolvido em duas fases: imagens frontais padrão são usadas para testar o método e depois é aplicado um ângulo às mesmas para testar a sua robustez. É desenvolvido em Python usando bibliotecas de visão computacional e uma configuração considerando todas as compensações necessárias para atingir o objetivo deste projeto. Resultados experimentais demonstram que este sistema automático demora um segundo entre deteção de tiro e apresentação da pontuação com uma eficácia de 98,3%.

Palavras-Chave: ISSF Air Rifle, Pontuação Automática de Alvos, Visão de Computador, Processamento de Imagem, Melhoramento de Imagem, Reconhecimento de Objetos, Deteção de Contornos, Sistema de Pontuação

Abstract

This thesis proposes a comprehensive, low-cost and high accuracy automatic shooting scoring system. It employs image processing techniques to analyse targets in real-time during live shooting sessions. In shooting sports, traditional scoring systems require a lot of time and resources, therefore, automation is necessary for efficiency and accuracy. The proposed system uses a camera positioned at a slight angle, as to not interfere with the shooter's line of shot, that captures every frame. Then, it precisely detects and scores pellet strikes by using image processing algorithms such as morphological image processing, perspective transform and target ring detection.

This system is intended for practical use in official indoor environments where a frame and black tape can be used as targets for a more affordable solution, unlike existing systems that are expensive. It is appropriate for amateur shooters looking for professional-grade precision without the need for additional equipment.

The algorithm is developed in two phases: standard frontal images are used to test the method and then an angle is applied to them in order to test its robustness. It is developed in Python using computer vision libraries and an optimisation setup was created considering all the trade-offs needed to prioritise affordability and achieve the purpose of this project.

Experimental results demonstrate that this automatic system has a one second delay between shot detection and score display with an efficiency of 98.3%.

Keywords: ISSF Air Rifle, Target Automatic Scoring, Computer Vision, Image Processing, Image Enhancement, Object Recognition, Contour Detection, Scoring System

Agradecimentos

A escrita e desenvolvimento desta dissertação marcam o fim do meu percurso académico. Um percurso com altos e baixos, mas um do qual me posso orgulhar. De pequena criança a começar a aprender o que são letras e números na pequena escola da Ponte em Santiago de Riba-Ul a Mestre de Engenharia Eletrotécnica e de Computadores formado na FEUP. Uma longa caminhada de sucessos, fracassos, aprendizagens e, acima de tudo, de muito crescimento. Nada disto teria sido possível sozinho e, como tal, estou eternamente grato a todos os professores, familiares e amigos que de alguma forma me ajudaram ao longo destes anos.

Em primeiro lugar, quero expressar a minha gratidão ao meu orientador Prof. Dr. Armando Araújo, que me aceitou desde início e me deu a oportunidade de realizar esta dissertação com total liberdade, mas estando sempre disponível para discutir ideias, criticar resultados e guiar-me na direção certa. Obrigado pela oportunidade de desenvolver este projeto ao seu lado.

A seguir, quero agradecer a todos os meus amigos que fizeram parte deste percurso, principalmente na faculdade. Um obrigado aos meus colegas de casa, Kamilot, Rodri, Pedrito e Matos, e a todas as pessoas que passaram por Aval de Cima. Um muito obrigado aos Primaços do Terraço e um grande obrigado ao Sengo por todos os momentos passados lado a lado.

Por fim, um obrigado ao meu pai, aos meus irmãos, ao meu padrinho, às minhas avós, à Marta e especialmente à minha mãe por tudo o que fez e sacrificou para que eu pudesse alcançar este objetivo. A vossa presença diária na minha vida é e sempre foi indispensável. Sem vocês não estaria aqui.

Guilherme Novo

"Yet be man low, or be he great,
no one controls a strong man's fate"

Bumpy Johnson

Contents

1 Introduction **1**
- 1.1 Problem Statement . . . 2
- 1.2 Objectives . . . 2
- 1.3 Dissertation Structure . . . 2

2 Background **5**
- 2.1 Computer Vision in Sports . . . 5
- 2.2 Rules . . . 5
- 2.3 Scoring Systems . . . 6

3 Fundamental Aspects **9**
- 3.1 Target Frame . . . 9
- 3.2 Digital Images . . . 10
- 3.3 Computer Vision . . . 10
  - 3.3.1 Images in CV . . . 11
  - 3.3.2 Image Processing . . . 13

4 State of the Art **17**
- 4.1 Electronic Scoring Targets . . . 17
  - 4.1.1 Calculate Position using TDOA . . . 18
- 4.2 Virtual Image Scoring . . . 19
  - 4.2.1 Important Breakthroughs . . . 20
  - 4.2.2 System Breakdown . . . 22

5 Methodology **25**
- 5.1 Image acquisition . . . 25
  - 5.1.1 Image Equipment . . . 26
  - 5.1.2 Camera Configuration and Setup . . . 26
  - 5.1.3 Image Capture Process . . . 27
- 5.2 Image Processing and Hole Detection . . . 28
  - 5.2.1 Preprocessing of Images . . . 29
  - 5.2.2 Hole Detection . . . 35
- 5.3 Scoring and Displaying Results . . . 36
  - 5.3.1 Target and Hole Centre . . . 36
  - 5.3.2 Pixels per Millimetre . . . 36
  - 5.3.3 Scoring Algorithm . . . 36
  - 5.3.4 Displaying Results . . . 38

6 Results and Discussion **41**
6.1 Automatic Scoring Algorithm Accuracy . . . 41
6.1.1 Official Scores . . . 41
6.1.2 Generated Scores and Comparison . . . 43
6.2 Image Processing for Target Detection . . . 45
6.2.1 Image Correction . . . 45
6.2.2 Image Segmentation . . . 46
6.3 Extra Features . . . 47
6.3.1 Image Differencing . . . 47
6.3.2 Overlapping Holes Recognition . . . 48
6.3.3 Serial ID Identification . . . 49
6.4 Final Discussion . . . 49

7 Conclusion and Future Work **51**
7.1 Conclusion . . . 51
7.2 Future Work . . . 52

A Appendix **53**
A.1 Complementary Results . . . 54
A.1.1 Frontal Targets 95421-95480 . . . 54
A.1.2 Angled Targets 95421-95480 . . . 56
A.1.3 Frontal Targets 62061-62120 . . . 59
A.1.4 Angled Targets 62061-62120 . . . 61
A.2 Software and Libraries used . . . 63

References **65**

List of Figures

1.1 Target used for scoring[20] . . . 1

3.1 Target and Frame . . . 9
(a) Target . . . 9
(b) Frame[6] . . . 9
3.2 Grayscale Pixel Grid[31] . . . 12
3.3 RGB Visualizations[1] . . . 12
(a) Plane . . . 12
(b) Channels . . . 12
3.4 HSV Visualization[1] . . . 13
3.5 Colour Spaces . . . 13
(a) RGB . . . 13
(b) HSV . . . 13
(c) Grayscale . . . 13
3.6 Blurred Images . . . 14
(a) Original . . . 14
(b) Size 3 . . . 14
(c) Size 9 . . . 14
3.7 Edge Detection . . . 15
(a) Original . . . 15
(b) Edges . . . 15
3.8 Perspective Transformation [39] . . . 15

4.1 Expanded view of an EST [2] . . . 18
4.2 VIS Components . . . 19
4.3 Automatic Scoring System Placement [51] . . . 22
4.4 Score Processing Architecture . . . 23

5.1 Logitech Brio 4k Stream Edition [3] . . . 26
5.2 Camera's Position . . . 27
5.3 Video Capture Flowchart . . . 28
5.4 Reference Targets . . . 29
(a) Frontal Target 95431 . . . 29
(b) Angled Target 95431 . . . 29
5.5 Blurred Targets . . . 29
(a) Frontal Target 95431 . . . 29
(b) Angled Target 95431 . . . 29
5.6 Grayscale Targets . . . 30
(a) Frontal Target 95431 . . . 30

(b) Angled Target 95431 . . . 30
5.7 Binary Targets . . . 31
(a) Frontal Target 95431 . . . 31
(b) Frontal Target 95431 . . . 31
5.8 Contour and Bounding Rectangle . . . 32
(a) Frontal Target 95431 . . . 32
(b) Angled Target 95431 . . . 32
5.9 Zoomed in Targets . . . 33
(a) Frontal Target 95431 . . . 33
(b) Angled Target 95431 . . . 33
5.10 Perspective Transformation . . . 34
(a) Angled Target 95431 . . . 34
(b) Top-Down View . . . 34
5.11 Warped Circle . . . 35
5.12 Value Channel . . . 35
(a) V . . . 35
(b) Binary . . . 35
5.13 Scoring Zones [40] . . . 37
5.14 Virtual Target . . . 38
5.15 Original and Digital Targets . . . 39
(a) Frontal Target 95431 . . . 39
(b) Angled Target 95431 . . . 39
5.16 Methodology Flowchart . . . 40

6.1 Official Scoring Cards . . . 42
(a) Card A . . . 42
(b) Card B . . . 42
6.2 Generated Scores Card A . . . 43
(a) Frontal Targets . . . 43
(b) Angled Targets . . . 43
6.3 Generated Scores Card B . . . 44
(a) Frontal Targets . . . 44
(b) Angled Targets . . . 44
6.4 Miss Classified Targets . . . 44
(a) 95461 . . . 44
(b) 95472 . . . 44
6.5 Image Correction . . . 45
(a) Before . . . 45
(b) After . . . 45
6.6 Final Scores . . . 46
(a) Without Correction . . . 46
(b) With Correction . . . 46
6.7 Binarization process . . . 46
(a) Original . . . 46
(b) HSV . . . 46
(c) V . . . 46
(d) Binary . . . 46
6.8 Contours . . . 47
(a) Original . . . 47

(b) Binary 47
6.9 Subtraction Process [51] 48
6.10 ID Number Placement [51] 49

List of Tables

2.1 Comparison between types of scoring 7

Abbreviations

ISSF	International Shooting Sport Federation
VIS	Virtual Image Scoring
EST	Electronic Scoring Targets
PPI	Pixels Per Inch
CV	Computer Vision
OpenCV	Open Source Computer Vision Library
TDOA	Time Difference Of Arrival
CHT	Circular Hough Transform
ROI	Region Of Interest

Chapter 1

Introduction

Shooting competitions have been around for centuries, with records as early as the 16^{th} century. They have always been a part of the modern Olympic Family and more recently became necessary to create a federation capable of laying down the rules and regulations for this sport and its competitions and so, in 1907, the International Shooting Sport Federation (ISSF) was officially founded. Since then, all kinds of shooting events have been hoisted all over the world, including European, World and Olympic tournaments.

There are 19 different types of events each one using one of the following firearms: rifle, pistol or shotgun. They vary in distance from the shooter to the target ranging from 10 meters all the way up to 300 meters but the goal is to always try and get as many hits as possible in the target's centre [24]. For each event there are specific targets with different dimensions. In this dissertation, the focus will be on 10m Air Rifle competitions. Each bullet is scored from 0.0 to 10.9 with shots further from the centre being worth less. The closest point to the bullet hole is the one taken into consideration when rating the shot [20], except when the shot enters the 10 zone in which case the point taken into consideration is the centre of the pellet hole, but that will be covered in more detail later. Figure 1.1 shows the targets used in these type of events and its rings dimensions.



1 Ring	45.5mm	6 Ring	20.5mm
2 Ring	40.5mm	7 Ring	15.5mm
3 Ring	35.5mm	8 Ring	10.5mm
4 Ring	30.5mm	9 Ring	5.5mm
5 Ring	25.5mm	10 Ring	0.5mm

Figure 1.1: Target used for scoring[20]

Initially, all targets were manually scored with the use of basic scoring tools including templates and scoring gauges in addition to human vision. In manual scoring, either participants score each other's targets using scorecards or all targets are scored by a group of official scorers [9]. Human scoring is cheap and simple, but on the other hand it can be slow and inconsistent and that

cannot happen when World and Olympic titles are at stake. So, with that, came the need to have accurate and secure scoring methods that assured a fair and reliable rating of each shot.

In the late 1980s, the ISSF officially changed its scoring system to electronic instead of the traditional manual scoring. Although it was already superior to manual scoring, there were moments when new equipment needed to be built in order to increase its effectiveness and reliability even more. However, this came at a cost. Money.

The main goal of this project is to create a dependable, consistent and precise system capable of analysing, scoring and displaying shots taken on 10m Air Rifle targets while also having a low-cost build and implementation.

1.1 Problem Statement

Compared to manual scoring, electronic methods are faster and more precise while requiring less personnel and time to score and display the final result. They resort to sensor based software or computer vision programs. Either way, both methods' implementation and maintenance costs are high and not every shooting club can afford them. This dissertation aims to develop an algorithm capable of outputting accurate and fast scoring results by using computer vision techniques to process images captured in real-time via web camera, while making it affordable.

1.2 Objectives

In order to be able to achieve this project's final goal, there are some crucial steps that need to be performed as to reach the final product. Therefore, this dissertation has the following assignments:

1. Study the state of the art in computer vision systems and target detection algorithms.
2. Development and implementation of a detection algorithm.
3. Compare different approaches used to score target images.
4. Validation of the quality and performance of the developed methods.

1.3 Dissertation Structure

This document is composed of 7 chapters. The first one is the current introduction, which contains a contextualisation and definition of the problem while also presenting the main objectives of this work.

In Chapter 2 some background related to the rules and regulations of ISSF official scoring competitions is given to the reader as well as an introduction to scoring systems.

In Chapter 3 fundamental aspects and concepts are presented in order to give some insights about the scope of this project and familiarise the reader with important concepts that will be frequently used during this thesis.

Chapter 4 contains an analysis of some state-of-the-art systems that are currently in use and that provided valuable insights and methods to problems that appeared during the development and testing of this system.

In Chapter 5, the methodology and techniques developed and employed throughout this work are described in detail.

Next, in Chapter 6 a comprehensive review and discussion of the final results is presented, as well as some extra features that could be implemented to improve the robustness of the system.

Finally, Chapter 7 contains the final conclusions about the whole study together with future work to be done and possible improvements to the chosen methodologies.

Chapter 2

Background

This chapter covers the importance of computer vision in sports while also delving into some of the ISSF official rules and regulations that need to be considered when building this project. Additionally, it introduces different scoring systems that have been used throughout the sport's evolution.

2.1 Computer Vision in Sports

Computer Vision (CV) has become one of the key elements in the realm of sports by transforming how players train, how coaches analyse and how the audience interacts with the games. By using advanced algorithms and image processing techniques, systems are capable of real-time data analysis and offer insightful information on every aspect of the game. [49]. Its methods can be used to recognise actions or activities as well as monitor athletes' performances and compare them to other previous acquired information, thus helping improve their game [25]. Furthermore, computer vision improves the overall viewing experience by providing interactive replays, augmented reality overlays and personalised content. Last but not least, by tracking player movements and object trajectories, it facilitates precise officiating or, in this case, precise scoring of shooting targets.

2.2 Rules

To be able to score properly there are a set of rules that need to be taken into consideration when creating the program and they can vary according to which target is being used. Hence, these are the rules that apply to the target used in this project.

To start with, the targets used are 20 centimetre squares with a black circle in the middle measuring 30.5 millimetres diameter wise which contains six smaller rings inside with distances between them as small as 5 millimetres. The centre ring has a diameter of only 0.5 millimetres. The scoring is done in tenths, meaning each ring is divided into ten parts, with every micrometre contributing to the score. The scoring takes into consideration the closest point of impact relative to the centre of the target, i.e., after finding the pellet hole the program then must find the closest

point to the target's centre and score based on that distance not the distance from the hole's centre, except when the pellet touches the 10 ring. For a shot to be scored a higher value it must at least touch the outside edge of the ring. With those rules in mind, the specific dimensions that the target has are presented below.

The 10m Air Rifle Target is divided into 10 circles. The scoring ring values 1-8 are printed on the target in vertical and horizontal lines at 90º to each other while the ring 9 is not marked with a number. The 10-point zone is a white dot. The diameter of the outer ring is 45.5 millimetres, and the diameter of the smaller rings are represented in Figure 1.1. The diameter of the black circle ranging from the centre to the fourth ring is 30.5mm [20].

2.3 Scoring Systems

There are two official electronic scoring systems, Virtual Image Scoring (VIS) and Electronic Scoring Targets (EST).

VIS uses an algorithm based on computer vision that can analyse and process photographs or video frames to obtain information to score digital or virtual targets [9]. A camera is mounted in front of the shooting frame in order to capture every important image of the target [51]. Several image processing algorithms are then applied in order to extract the pellet's position relatively to the target's centre. Although it is still expensive, it is the most affordable electronic system and produces fairly fast results. Since it's a computer software it can have additional features such as a database and communication with online services.

EST or acoustics based scoring systems rely on sensors to compute the precise location that a projectile passes through a target [9]. Some use pairs of vector sensors to calculate the hit or miss by using the difference between the shock wave's time of arrival and amplitude of the sensors [41]. In addition to involving complex circuits and having sensors placed in front of the target, they are quite expensive [8]. Others, instead of a paper target use an electronic one with an acoustical, laser or light beam system built within that gathers information about the shot so it can be calculated.

In both systems, scores are displayed within milliseconds which contributes to a better viewer experience [9]. They also have a similar scoring accuracy but with EST being slightly more expensive. Table 2.1 shows a comparison between VIS, EST and manual scoring when scoring 120 targets.

SYSTEM	SCORING ACCURACY	SCORING SPEED	PERSONNEL REQUIRED	PRESENTATION	ONLINE RESULTS	COST
Manual Scoring	0.100mm - 0.200mm	30 min - 2hours	One or two scorers per 10 firing points	Scores shown after scoring	Possible with hand entry after scoring	75-100€
VIS	<0.100mm	1-2 minutes	One or two scorers familiar with software	Scores shown and uploaded after scoring	Online posting integrated into software	1000-2000€
EST	<0.100mm	Milliseconds	One EST trained officer and one target officer	Scores shown after scoring	Limited online results posting	3000-6000€

Table 2.1: Comparison between types of scoring

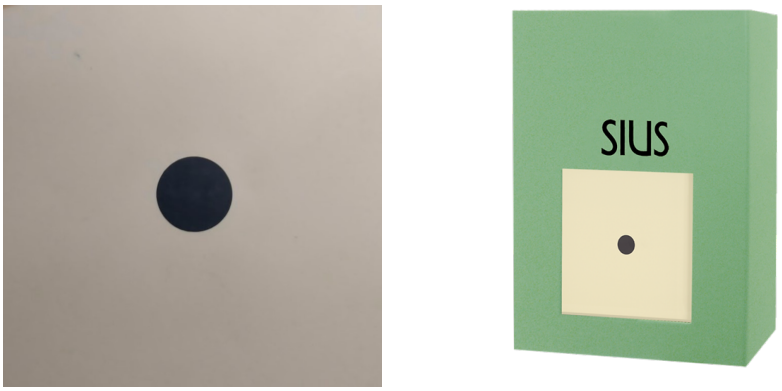
In this project an approach similar to the VIS method will be implemented using Python programming language and computer vision libraries while trying to minimise even more the costs of implementing and running this scoring process.

Chapter 3

Fundamental Aspects

3.1 Target Frame

The target frame is an essential part of the system as it will hold the target in place and will have a major role in isolating the target from the rest of the image. As will be mentioned in Section 3.3.2.4 a computer can use a change in colour to detect the edges of an object and isolate it. This process will be explained in more detail when its implementation is covered but for now is important to know that having different colours between objects helps in image processing and analysis. With that in mind, Figure 3.1b shows that official target systems use a frame that differs in colour from the target (Figure 3.1a). Another important aspect to be noticed is that within the target there is a hole the size of the black circle and it plays a very important role as it allows the use of ringless black tape behind it.



(a) Target (b) Frame[6]

Figure 3.1: Target and Frame

For the earlier testing stages of this project, and with a low-cost build in mind, a frame was 3D printed and painted in green around the target zone in order to replicate the official ones from SIUS.

3.2 Digital Images

Pixels are the building blocks of a digital image and in computer vision the computer analyses an image as a big grid with each space containing one pixel[18]. A camera resolution defines the dimension of that same grid which contains more pixels the bigger it is. Having more pixels means having more spatial resolution and more pixels per unit area which helps measuring fine margins as the ones encountered along this project. Therefore, the number of pixels captured by the camera will massively influence the software's ability to score the targets accurately. In this project a 4k resolution camera will be used meaning the pixels in the captured images will be in a 3840x2160 grid, i.e, there will be 8294400 pixels in one image.

A few factors must be considered to maximise the results because, even though a good camera will be used, it is not the ideal one given the low-cost build that this work is aiming for. A good environment lighting is crucial for the camera to be able to capture the image with sharpness and detail. The reference points must be stable so that the measurements are uniform across different shots and the distance between camera and target must be ideal so that it can capture the image without losing too much detail. In other words, the pixels in the captured image must be mostly within the target so that the measurements can be done with more precision since there will be more pixels per millimetre and all the conversions can be done more accurately.

In the physical world, image resolution can also be measured in pixels per inch (PPI), which describes the number of pixels displayed in an image per inch. Higher resolutions have greater PPI, which produces more pixel information and a higher quality image for the same image size. [32]. To calculate the number of pixels per inch of an image the width of the image in pixels W, the height of the image in pixels H and the diagonal length of the image in inches D must be known and the following equation used:

$$PPI = \frac{\sqrt{W^2 + H^2}}{D} \tag{3.1}$$

Experiments show that enlarging an image beyond its original size results in blurriness and a loss of detail. This occurs because the size of an image and the number of pixels per inch (PPI) are inversely proportional.

Pixels per inch (PPI) is primarily useful for printing images and holds less significance in computer vision. However, understanding PPI helps highlighting the importance of having a high number of pixels to effectively analyse and distinguish small details in images.

3.3 Computer Vision

Artificial intelligence's field of Computer Vision allows systems to analyse digital photos, movies, and other visual inputs to extract relevant data [12]. Based on this information, they are able to take actions, make recommendations or simply display data. While humans rely on retinas, optic nerves, a visual cortex and life experiences to perceive the world around them, computers must use cameras, data and algorithms to achieve the same result. Nonetheless, with proper software,

computers can surpass human capabilities and notice fine margins that are imperceptible to the human eye [26].

CV is being used in various realms of the real-world. From enabling self-driving cars by recognising its surroundings to diagnosing diseases through the analysis of medical scans, it has become indispensable to the ongoing development and evolution of the world [48].

The fundamental processes start with image acquisition, where visual information from the environment is captured using devices such as cameras, sensors or scanners. This same data is then preprocessed using various techniques in order to prepare it for future analysis. It is the foundation for the process of feature extraction which will be crucial in this work [17].

In this project, the goal is to use a system's ability to process and analyse a target image in order to make an accurate calculation of the score.

3.3.1 Images in CV

To understand how a computer perceives an image we must become familiar with two concepts. Pixels and colour spaces.

Pixels were already introduced in Section 3.2 as being the building blocks of digital images. The simplest possible definition is to consider them little squares [11].They are the smallest items of information in an image and are arranged in a 2-dimensional grid. Each pixel can be represented by one or more values depending on what colour space the image is being analysed. They can also be defined as a function, where x, y are spatial coordinates and I is the intensity of the pixel at that point:

$$I\ (x,y) \in P^i \text{ and } x,y \in N \tag{3.2}$$

A mathematical concept known as a colour space explains how colours can be represented as tuples of numbers. These values are used to define a specific colour within the colour space. There are various colour spaces but in this project the focus is on three of them. Grayscale, RGB and HSV.

3.3.1.1 Grayscale

Grayscale is a one channel colour space where each pixel is represented by a single 8-bit number and does not contain information about colour, instead, the value represents the brightness of each pixel. 0 represents black, 255 represents white and every value in between represents varying shades of grey [7]. Each pixel can be defined as:

$$I(x,y) = \text{value}, \quad \text{value} \in [0,255] \tag{3.3}$$

In Figure 3.2 it is possible to see how a computer reads a pixel grid and transposes each value to its corresponding shade of grey.

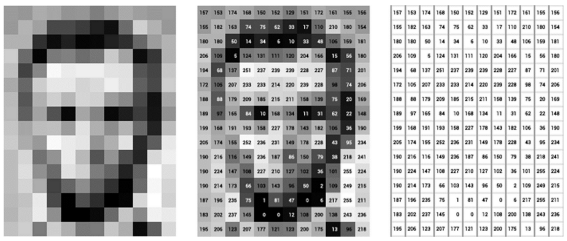
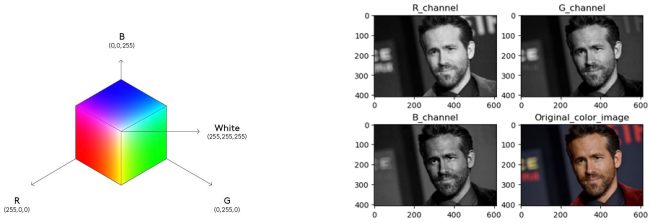


Figure 3.2: Grayscale Pixel Grid[31]

3.3.1.2 RGB

RGB is the most commonly used colour space. It represents colours as combinations of three channels corresponding to three main colours: Red (R), Green (G) and Blue (B). Each pixel contains an 8-bit value from each channel and, based on the additive colour theory, can produce a wide spectrum of colours. They can be seen as points in a three dimensional coordinate plane in which the three perpendicular axis correspond to the three colours [1]. Figure 3.3a helps visualise that idea. The three different colour channels and the merged result can be seen in Figure 3.3b. Note that the colour channels don't actually have colour, their concept is similar to the grayscale channel in which the pixel value represents the intensity of the colour and not the actual colour. They can be defined by a triplet where each value ranges from 0 to 255:

$$I(x,y) = (R,G,B) \tag{3.4}$$



(a) Plane (b) Channels

Figure 3.3: RGB Visualizations[1]

3.3.1.3 HSV

HSV colour space represents colours in three components: Hue (H), Saturation (S) and Value (V). Provides a more intuitive representation of colour perception. Hue represents the type of colour without any white or black added. Its value corresponds to an angle, ranging from 0 to 360 degrees, that is associated to a colour in a colour wheel. Red is at 0 degrees on the colour wheel, followed by green at 120 degrees, blue at 240 degrees, and red once more at 360 degrees. [47]. Saturation represents the depth or purity of the colour. It can be represented as a percentage where 0% is grayscale and 100% is fully saturated, i.e, very intense colour. Value determines the brightness of the colour. It is also represented as a percentage where 0% is no brightness or black and 100% is

full brightness or white. Similar to the RGB its pixels can also be defined as triplet. In the three dimensional space this colour model can be represented as a cone [45].

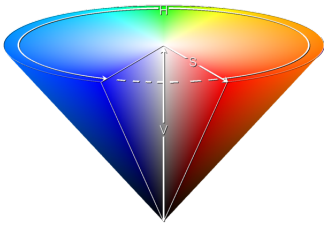


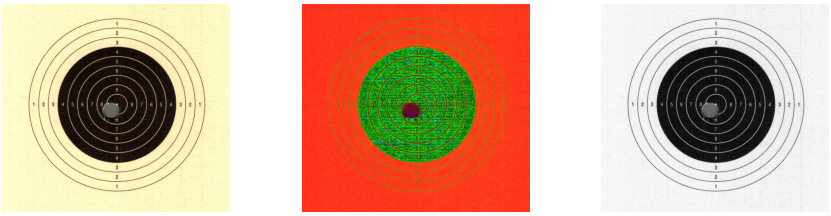
Figure 3.4: HSV Visualization[1]

3.3.2 Image Processing

In order to be able to interpret and understand visual data as close as possible to humans, computers must resort to the use of various techniques and algorithms to manipulate and analyse digital images. This includes enhancing, filtering and segmenting images as a way to make sense of visual input and retrieve information. These processes enable the development of efficient systems capable of performing visual tasks with high accuracy and reliability. Amongst all processes available in image analysis there are a few that are crucial to this project so the next subsections contain a brief explanation of each one.

3.3.2.1 Changing Colour Spaces

This process changes the representation of colours from one colour space to another. They rely on mathematical algorithms to convert each pixel in the image. Involves recalculating colour values based on the rules of the target colour model and, depending on the conversion, making additional adjustments to match the different value ranges between colour spaces. The result is an image in the desired colour model which can be used for further analysis and processing.



(a) RGB (b) HSV (c) Grayscale

Figure 3.5: Colour Spaces

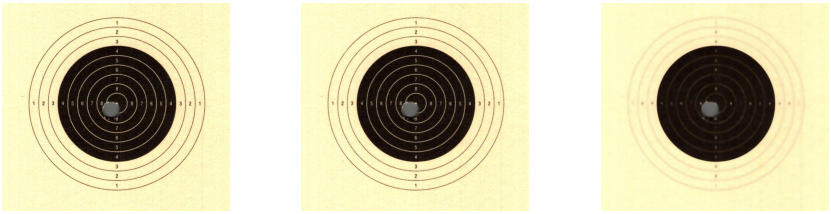
3.3.2.2 Thresholding

Thresholding is a simple yet powerful processing technique that involves converting a single-channel model image into a binary format where pixels are classified as foreground or background

depending on their intensity value. The first step is to convert the image to to a single-channel model where pixel values represent intensity. The next step is to compare each value to a threshold value previously chosen that serves as dividing point between foreground and background pixels. If it's higher then set the new pixel value to 1, representing foreground, if not then set it to 0, representing background. It's a very useful technique for segmentation and region isolation.

3.3.2.3 Smoothing

Also known as blurring or filtering, smoothing is a technique aimed at reducing noise and creating a more visually simplified representation of an image [21]. Involves applying convolution operations with a smoothing kernel (matrix of weights) to an image. The kernel or filter determines how neighbouring pixel values influence the smoothing operation. It is applied to every pixel in the image starting on the top left and sliding over the image. At each position the weighted sum of pixels in the kernel's neighbourhood is computed and replaces the original pixel value [21], resulting in a smoothed or blurred version of the original image. The degree of smoothing depends on the size and shape of the kernel as well as the values within it. Figure 3.6 shows how different sized kernels produce different results.

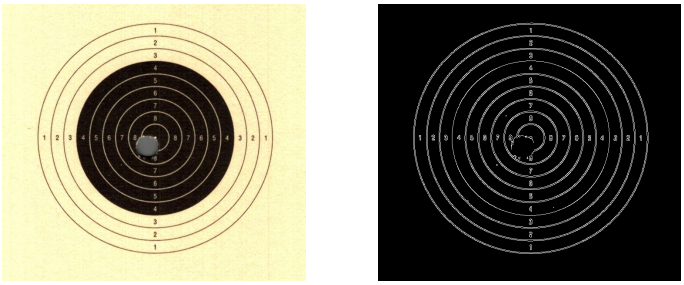


(a) Original (b) Size 3 (c) Size 9

Figure 3.6: Blurred Images

3.3.2.4 Edge Detection

A basic method for determining boundaries within an image is edge detection, which involves examining each pixel value and looking for noticeable variations in colour or intensity. In an image, edges frequently line up with the perimeters of important structures. It is crucial in image segmentation and object recognition. Similar to thresholding, it starts by converting the image to a single-channel model and then may or may not apply a smoothing filter. From there the gradient of the image is calculated in order to highlight areas with rapid changes in intensity. After that, the edge strength at each pixel is computed by combining the gradient information in both horizontal and vertical directions and threshold value is defined as to distinguish between pixels that are part of the edge and those that are not. In the end, additional steps can be performed to try and complete the detected edges. The final result is a binary image where edge pixels are highlighted representing the location of significant intensity changes in the original image.



(a) Original (b) Edges

Figure 3.7: Edge Detection

3.3.2.5 Perspective Transformation

Perspective transformation, as the name suggests, is a technique used to change the perspective from which an image is viewed. It maps the points in one plane to another using projective transformation, which simulates the change in viewpoint and corrects distortions. The mathematical foundation for perspective transformation is a 3x3 transformation matrix that allows the transformation of the coordinates, even though the work is performed over two-dimensional images [21]. Straight lines remain straight in the output image, but parallel lines may converge or diverge. It can be mathematically represented as:

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \tag{3.5}$$

where H is the 3x3 matrix, (x,y) and (x',y') are the points in the input and output image respectively and w' is a scaling factor.



Figure 3.8: Perspective Transformation [39]

Chapter 4

State of the Art

This chapter provides a comprehensive review of the current state of the art in automatic scoring shooting systems, describing the methods, applications and technologies that are used in this field. In addition, their ability to provide real-time feedback, which is essential for training and competitive scenarios, will be examined as offering instant results can facilitate performance analysis and improve the overall effectiveness of the system. Furthermore, the challenges and limitations currently faced by the existing systems are also analysed, including issues related to environmental conditions and sensor calibration. Lastly, this chapter includes an analysis of the systems currently in use and insights into emerging ones that are likely to have influence in this field.

4.1 Electronic Scoring Targets

As mentioned in Section 2.3 EST are the most used systems in the professional realm of shooting and the only ones being commercially sold. These are automated scoring systems that use electronics to determine the shot placement and score [14].

Unlike conventional paper targets, which consist of a flat plane, these targets are three-dimensional boxes with an interior that serves as an acoustic (or "echo") chamber. A circular cutout the size of the black scoring part of the target can be found on the foam face of the box. A black rubber mat hanging within the box creates the dark area that the shooter sees, and a second mat and foam surface close off the acoustic chamber at the back. There are no discernible scoring rings, just a continuous white ring encircling a black centre. [2].

Four sensors, one in each of the chamber's four corners, send a signal to a processor either wirelessly or through a cable when a projectile enters the acoustic chamber. The sensor array also transmits temperature data for computations as the shock wave's (the sound's) speed varies with air temperature. The shock wave from the projectile hits the nearest sensor first, and it takes milliseconds for the sensors farther distant to register it. The position of impact is then determined by the computer by measuring the time difference of arrival (TDOA) of each individual sensor [2].

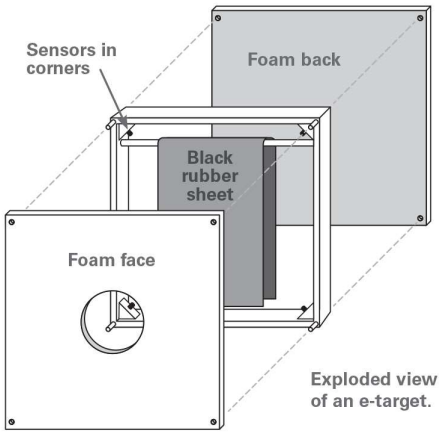


Figure 4.1: Expanded view of an EST [2]

4.1.1 Calculate Position using TDOA

The spatially separated sensors receive signals from the shock wave at different times, which are then processed by the processor. The time difference between the signal received by a pair of sensors and the source is known as the TDOA. [15]. The TDOA between the sensor i and j when the source s is emitting a wave is given by:

$$TDOA_{ij} = \frac{(||m_i - s|| - ||m_j - s||)}{c} \tag{4.1}$$

where, m_i and m_j are the spatial positions of the sensors, s is the position of the source and c is the speed of sound.

When the TDOA from at least 3 sensors is known, the position of the acoustic source can be calculated. First, a sensor will be at the origin of the coordinate system and all distance differences will be defined to reference that same sensor, r_1, using the following equation:

$$r_{i1} = r_i - r_1 = (t_i - t_0)c - (t_1 - t_0)c = (t_i - t_1)c \tag{4.2}$$

Next, to locate the source, three unknowns need to be discovered, the coordinates of the source, (x_s,y_s) and the distance to the reference sensor or the origin of the coordinate system, r_s, which is easily calculated using the coordinates as $r_s = x_s^2 + y_s^2$. To discover the coordinates the following system is used:

$$\begin{bmatrix} x_2 & y_2 & r_{21} \\ x_3 & y_3 & r_{31} \\ & \cdots & \\ x_n & y_n & r_{n1} \end{bmatrix} \cdot \begin{bmatrix} x_s & y_s & r_s \end{bmatrix} = \frac{1}{2} \begin{bmatrix} K_2^2 - r_{21}^2 \\ K_3^2 - r_{31}^2 \\ \cdots \\ K_n^2 - r_{n1}^2 \end{bmatrix} \tag{4.3}$$

where n is the number of used sensors and $K_i^2 = x_i^2 + y_i^2$.

4.1.1.1 Temperature Variance

As stated above, the temperature can have an impact on the measurements as the speed of sound in air varies with the temperature [5]. Differences of 10 degrees Celsius, which are normal indoors, can have a meaningful impact on the results [15], therefore the sensors also pass this data to the computer so it can be taken into consideration. The speed of sound is related to the temperature as follows:

$$c = 331.4 + 0,6T \tag{4.4}$$

where c is the speed of sound and T is the temperature in Celsius.

4.2 Virtual Image Scoring

VIS are automatic shooting scoring systems based on image processing using data captured by a single camera [51]. A complete image processing system is composed of four components [29, 36]:

1. An image acquisition equipment. It can be a camera or a scanner, preferably with connection to the processing unit.
2. A processing unit capable of analysing and processing images. It can be a computer or another type of processor.
3. A storage unit to export and import data. It can be within the processor, an online database or an external device such as pen drives.
4. A display unit capable of showing the final results. It can be a monitor or a projector.

Figure 4.2 shows its composition.

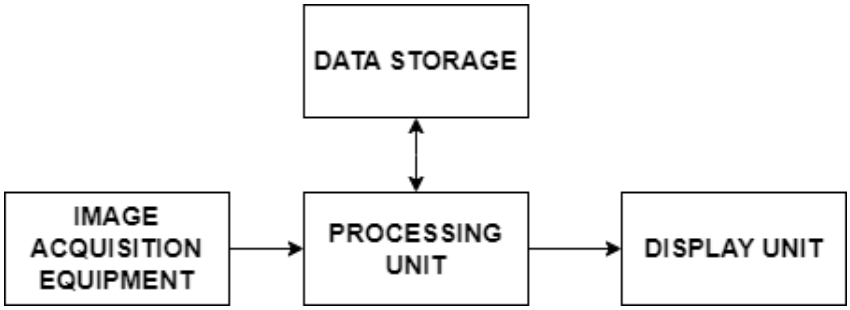


Figure 4.2: VIS Components

Along the years there has been a constant evolution in development regarding systems based on image processing. In this next sections a few important landmarks and dates will be presented in greater detail, as well as a breakdown of the generic structure used in these type of systems.

4.2.1 Important Breakthroughs

4.2.1.1 F. Ali and A.B. Mansoor

In 2008, F. Ali and A.B. Mansoor proposed an automated scoring system for target shooting that was quick and inexpensive, employing computer vision techniques. [8]. They start by thickening the projectile hit boundaries with morphological operations on the target image and then segment the target area by hysteresis thresholding. By using adjustable thresholds they can also manage the effects of illumination variations.To calculate the score inside the bullseye, a distance transform is used to segment it enabling the algorithm to score both inside and outside the bullseye. Following the target region's segmentation, the projectile strikes are identified and any overlapping holes are also scored by designating a threshold pixel area. They tested their program on 100 target photos with different number of projectile hits and achieved 98.3% accuracy [8]. Although the system proved to be efficient it was built to analyse and score targets from 1 to 10 only and did not follow ISSF rules, therefore it can only serve as a basis for this work as it cannot be used in the same environment as this project's system.

4.2.1.2 J. Rudzinski and M. Luckner

In 2012, J. Rudzinski and M. Luckner suggested a low-cost computer vision-based automated shooting target scoring system. The suggested framework only uses direct image handling techniques, such as the Hough transformation and edge recognition, and requires no extra hardware [40]. Their aim was to create a professional system that could be used by amateur shooters without the need of special hardware. Any consumer grade camera or mobile device capable of capturing images with at least a 0.5 mega pixels resolution could be used. The system consists of three major steps. Target detection, hole detection and score calculation. The first two rely solely on existing image processing techniques while the latter requires mathematics. It is estimated to correctly score 92% of the holes [40]. It was designed to score ISSF official targets as the one shown in Figure 1.1 and serves as a strong foundation for this work but it relies on detecting the target's rings during image processing and this project uses ringless targets, therefore it cannot be used in the same setup as this project's system.

4.2.1.3 T. Thin Zin

In 2013, T. Thi Zin et al suggested an autonomous scoring system for archery targets. The framework uses morphological processing on the objective image to increase the bolt hit limits' thickness before using a division cycle to break the target region using highlights in form and colour . As it is a system used on archery targets and built following official archery rules and dimensions it cannot be used to solve this project's problem but it took a big step in image processing scoring techniques as it uses the geometrical properties of circles and ellipses in order to segment the target's concentric circles and calculate hit scores [55].

4.2.1.4 Y. Chi Lin

A programmed rendition was proposed by Y. Chi Lin et al in 2015. The region of a projectile opening and the sequential ID number on the target sheet are identified separately during image preparation processes in order to determine the score that the shooter with that ID number obtained, therefore contributing to an even more automated scoring system by taking away the need to manually assign each score to its shooter [34].

4.2.1.5 P. Widayaka

In 2019, P. Widayaka et al presented a paper describing an image processing based automatic shooting scoring system for live shooting sessions. To record each and every shot, a camera is positioned in front of the target frame. Afterwards, a number of image processing techniques are employed, including morphological image processing, perspective transformation, target ring identification, and image removal. By extracting the detected contour and projectile hole position and using the bounding box function, the contour detection method is employed to execute a perspective transform and obtain the circle diameter and centre circle position. Experimental findings demonstrate that this method correctly scored 91% of the targets [51]. Similar to F. Ali and A.B. Mansoor's system, it was not built following ISSF rules and relies on audio sensors to detect when a pellet was shot and for those reasons it cannot be applied in the same environment as this project's system.

4.2.1.6 Z. Su and W. Chen

In 2019, Z. Su and W. Chen proposed a system where edge information of input images was used to extract the region of interest where the key features were located, allowing the creation of a mask that can isolate those components from the rest of the image, reducing computational requirements. By only examining that area, they achieved an effective target extraction and, subsequently, applied all the processing techniques required to produce clear results [46]. Although this system was not developed following official rules, it provides valuable techniques that can be implemented in all types of image based scoring systems.

4.2.1.7 Machine Learning

Machine Learning is a field of artificial intelligence focused on the development of algorithms and statistical models that enable computers to carry out tasks without direct human input. Instead, these systems rely on patterns and deductions attained from data. The primary goal of machine learning is to build models that can evolve from training data into making predictions or decisions based on new, unseen data [10]. One sub field of machine learning is deep learning.

Deep Learning uses neural networks with many layers to model complex patterns in data. They are particularly effective in dealing with unstructured data such as images. Neural networks are composed of layers of nodes where each connection represents a weighted link. These weights

are adjusted based on the error of the output relative to the expected result. By feeding training data into the system, it progressively achieves more accurate results [23].

In 2024, M. Butt et al built an automatic image based scoring system using deep learning techniques. Their study examined the feasibility of the system using object detection models. The results showed that with precise training and a meticulous validation dataset, accurate detection is possible. Their best model resulted in a mean average precision of 96.7% and was achieved by the YOLOv8m model, which is a single-shot detector. In order to improve the scoring speed, the system is supposed to eliminate image processing techniques, but results show that applying those same techniques can improve the overall efficiency of the detection model. In addition, to improve its robustness and achieve the best outcome, the training dataset must include images from all different possible scenarios to have equally annotated instances of each one [13].

In short, deep learning models are starting to emerge in the realm of automatic image based shooting scoring systems, but due to its complex implementation, training requirements and overall not so great precision, it is still not a viable option to pursue when building these type of systems, therefore this work does not contain deep learning models.

4.2.2 System Breakdown

These systems use a camera installed slightly above the target frame as to not interfere with the shooter's line of sight [43]. A general implementation of the scoring system can be seen in Figure 4.3.

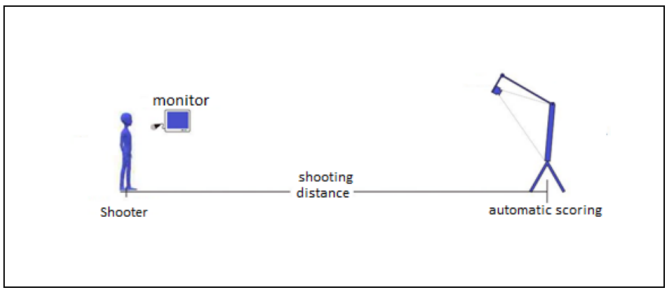


Figure 4.3: Automatic Scoring System Placement [51]

They start by capturing an image of the target sheet which can be done using different techniques. One way is to implement a microphone sensor into the frame that sends an analogue value to the computer when it detects the presence of the pellet's impact sound which then can be used to trigger the camera into taking a picture of the target sheet so it can be further analysed and scored [43]. Another way, and the one inspiring this work, is by having a constant live recording of the target and as soon as a pellet hole is detected by the program it isolates the frame and analyses it.

After acquiring the image it can then start to perform preprocessing operations in order to eliminate background interference and end up with a clear image for further procedures [44]. The operations include perspective transformations and binarisation as well as other techniques that will be later presented in great detail. Following the preprocessing, the program proceeds to detect

the target's ring and its centre as well as the pellet's hole. When all this information is gathered, the scoring algorithm takes the data and calculates the score for each shot through the centre position of the pellet and the target. Figure 4.4 shows the proposed architecture for these systems.

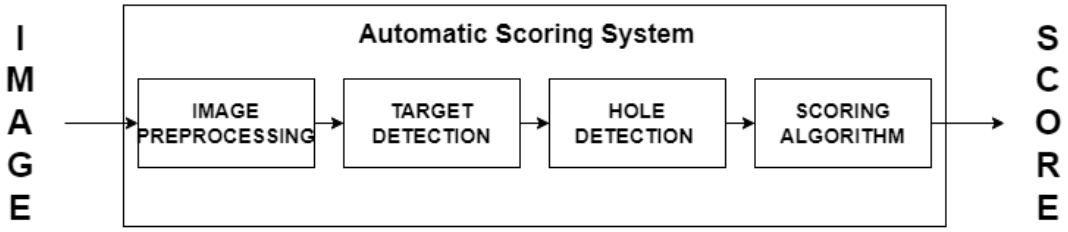


Figure 4.4: Score Processing Architecture

Chapter 5

Methodology

This chapter describes the approach used in this dissertation to conduct the analysis and obtain the desired results. The chapter is broken up into three sections, each of which focuses on a different stage of the research process.

Firstly, it provides an overview of the techniques and equipment used for acquiring images in automatic shooting target systems. This includes the use of a high-resolution camera and the specific settings and configurations required to capture clear and accurate images of the targets. Understanding the principles and capabilities of these imaging systems is essential for comprehending the subsequent methodology employed in this study.

Secondly, the methodology for processing the acquired images and detecting holes is presented. This process involves several key steps. Initially, the preprocessing of images to enhance quality and clarity is shown and explained. Subsequently, the techniques for detecting holes are discussed, including various algorithms and approaches to accurately identify and characterise the holes. The variations in these methods allow for a comprehensive analysis of their advantages and trade-offs.

Finally, the chapter delves into the method used for scoring the detected holes and displaying the results. Accurate scoring is crucial for providing reliable feedback to the shooter. This section presents the scoring algorithms in detail, outlining the necessary steps to calculate scores based on the detected holes. Additionally, the methods for displaying the results in an intuitive and user-friendly manner are discussed. By employing these techniques, the study ensures that the resulting scores are accurate and that the feedback is clear and useful for the shooter.

By presenting these methodologies, this chapter provides a clear and detailed framework for the analysis conducted in this dissertation, highlighting the importance of each step in achieving accurate and reliable results in automatic shooting target systems.

5.1 Image acquisition

This section establishes the foundation for the study by providing an overview of the techniques and equipment used for acquiring images in automatic shooting target systems. Accurate image

acquisition is critical to ensure that subsequent analysis and scoring are based on clear and precise visual data.

5.1.1 Image Equipment

The primary equipment used for image acquisition in this study includes a high-resolution camera capable of capturing detailed images of the shooting targets. This camera was selected based on its resolution, frame rate and ability to capture highly detailed images. The chosen camera was a Logitech Brio 4k Stream Edition [3]. After conducting research, it was concluded that this camera, with its state-of-the-art build, USB connectivity, and affordable price, is the ideal choice for this work.



Figure 5.1: Logitech Brio 4k Stream Edition [3]

5.1.2 Camera Configuration and Setup

To achieve optimal image quality, the camera is configured with specific settings tailored to the shooting environment. Its high 4K resolution is particularly crucial, as it provides the necessary pixel density and clarity for capturing images, enabling precise and accurate measurement of small margins. With the scoring zones being as small as 0.025 millimetres and with the need of having a minimum of one pixel per scoring zone, it needs to have at least 40 pixels per millimetre. Using the maximum resolution, this ratio is obtained when capturing images up to widths of 96 millimetres which is enough to capture the whole scoring area. Its exposure and ISO settings are adjusted to ensure clear images under 1500 lumen, which is the default luminous flux in the shooting environment.

For the second part of this work the camera is strategically positioned to provide an unobstructed view of the target. The placement ensures that the entire target is within the camera's field of view, with a focus on minimising angles that could introduce a lot of perspective distortion. Figure 5.2 helps visualise the concept.



Figure 5.2: Camera's Position

During dataset acquisition the ideal placement for the camera was found to be 15 centimetres above and 20 centimetres in front of the target, which resulted in an angle around 37º. This proved to be the best placement as it could fully capture the target without inducing much distortion and without obstructing the shooter's view of the target.

5.1.3 Image Capture Process

The image capture process is automated to synchronise with the shooting events. The hole detection algorithm will be explained in great detail in the next section, for now it is important to shed some light on the video capture process and flow of the system.

Using real-time video capture the program can isolate frames as soon as it detects a new hole in the target. After isolating the images and applying the image processing and hole detection algorithms, the system then displays the shot on a virtual target and stores the score on a file that can be accessed later. Next, as the system is built to only analyse one shot at a time, it waits until it no longer detects a hole meaning the target has been replaced and it can repeat the whole process again. This ensures that each image corresponds accurately to a single shot, facilitating precise analysis. Figure 5.3 shows a flowchart for better understanding.

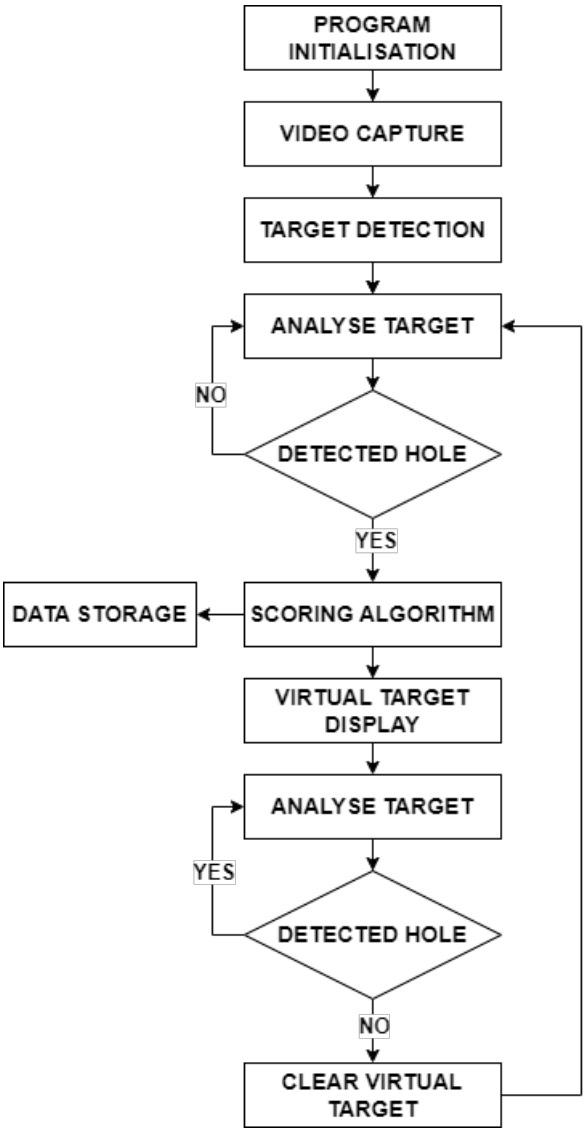


Figure 5.3: Video Capture Flowchart

For the testing of this algorithm, the video capture feature was not necessary as the dataset was obtained by capturing images of previously marked targets as to minimise resources and maximise samples.

5.2 Image Processing and Hole Detection

This section presents the methodology for processing the acquired images and detecting holes. Accurate image processing and hole detection are crucial for reliable scoring and feedback in automatic shooting target systems. Along this section two examples will be shown as the system was tested for frontal images and images taken from an upper angle inducing some perspective distortion. The same target will be used in both examples, but rotated in order for the reader to be able to distinguish them in the final result. Both targets are presented in Figure 5.4.



(a) Frontal Target 95431 (b) Angled Target 95431

Figure 5.4: Reference Targets

5.2.1 Preprocessing of Images

The initial step in the image processing workflow involves preprocessing the acquired images to enhance their quality and clarity. This includes several task that are presented in detail below.

5.2.1.1 Smoothing

Smoothing or blurring is a technique used to reduce noise in an image, facilitating following tasks such as edge detection and image segmentation. It is a fundamental preprocessing procedure that reduces the impact of noise and irrelevant details, improving the efficiency of higher-level algorithms [53]. The blurring method used is the median filter, implemented in Python's OpenCV by the function '*cv.medianBlur()*'. With the use of a square kernel of a specified size, this function replaces each pixel's value with the median value of the intensities in its neighbourhood. After sorting the pixel value within the kernel, the middle value is used to compute the median. When it comes to eliminating salt-and-pepper noise and maintaining edge sharpness, this method works better than Mean or Gaussian blurring [22, 28].

The input to this function is the original image (Figure 5.4) and the kernel size, which must be a positive odd integer, in this case is 3. The output is a smoothed image that preserves image structure while lowering noise. Each pixel's value is based on the median of its immediate neighbourhood [4]. Results shown in Figure 5.5.



(a) Frontal Target 95431 (b) Angled Target 95431

Figure 5.5: Blurred Targets

At first glance there is almost no difference but it helps cleaning background noise and some unwanted pixels that might meddle with the next processes.

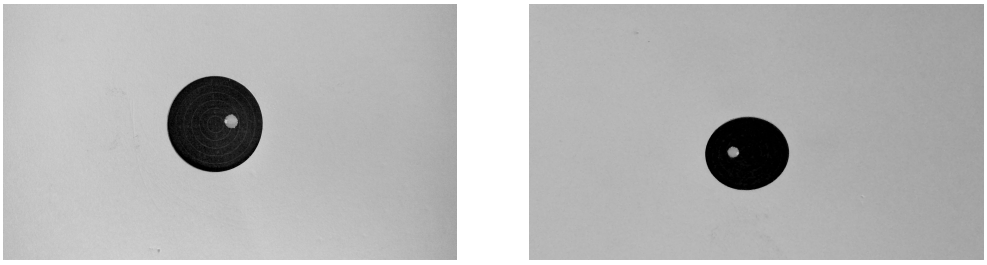
5.2.1.2 Grayscale

The next step is changing the colour model to grayscale in order to reduce the amount of colour channels. The camera produces colour space distortion in its captured images due the differences between ambient conditions and the type of input conversion equipment [52]. Therefore, the image must be uniformly turned into a grayscale image before moving on to the binarisation phase. Grayscale conversion in computer vision is a technique that converts a colour image into shades of grey, representing intensity variations. It is an essential process in edge detection and object recognition [22]. In order to convert an image from RGB to grayscale in OpenCV, the function '*cv.cvtColor()*' is commonly used. This function combines red, green and blue using a weighted sum that reflects human perception:

$$Y = 0.299R + 0.587G + 0.114B \tag{5.1}$$

More weight is given to the green channel as the human eye is more sensitive to that colour [37].

The inputs to this function are the blurred coloured image (Figure 5.5) and a conversion code '*cv.COLOR_BGR2GRAY*', specifying the desired colour space transformation. The output is a single-channel image where each pixel's intensity represents the corresponding grayscale value [4]. Results shown in Figure 5.6.



(a) Frontal Target 95431 (b) Angled Target 95431

Figure 5.6: Grayscale Targets

5.2.1.3 Thresholding

After those steps, the binarisation of the image in order to fully be able to distinguish the target's black circle from the rest of the frame and background is ready to be made. Thresholding is a technique used to simplify images by converting them from grayscale into binary, allowing rapid segmentation of target and background, thus facilitating feature extraction [54]. Every pixel's intensity value is compared to a predetermined threshold to determine how each one should be set: pixels with intensities above the threshold are set to a maximum value, usually white, while those below are set to a minimum value, usually black. This process effectively highlights regions

of interest within an image [22, 45]. In OpenCV, the function '*cv.threshold()*' implements this technique.

The inputs are the grayscale image (Figure 5.6), the threshold value 100, the maximum value 255 (white) and the thresholding type flag '*cv.TRESH_BINARY*'. By using this flag, the function assigns the maximum value to pixels greater than the threshold and zero to the rest. The output is a binary image where objects of interest are distinctly separated from the background [4]. Results shown in Figure 5.7.



(a) Frontal Target 95431 (b) Frontal Target 95431

Figure 5.7: Binary Targets

5.2.1.4 Contour Detection

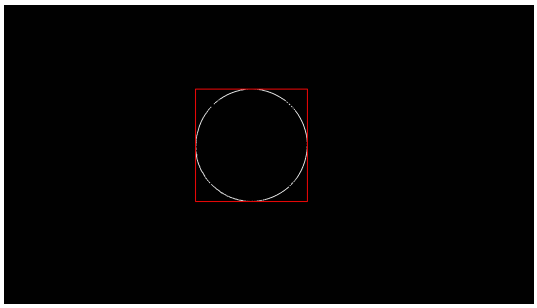
The circle or ellipse are achieved by contour following [16]. With the process of binarisation done, is only a matter of applying contour detection techniques such as edge detection in order to retrieve the target's bounding rectangle and apply further processing techniques such as digital zoom and perspective transforms. Edge Detection is a critical technique used to identify the boundaries of objects within an image and one effective approach involves finding contours, which are lines joining all the continuous points that have the same intensity along a boundary [22, 48]. In OpenCV, the function '*cv.findCountours()*' allows the detection of these contours in a binary image.

This function's inputs are the binary image (Figure 5.7), the retrieval mode '*cv.RETR_LIST*' that retrieves all the contours, and the approximation method '*cv.CHAIN_APPROX_SIMPLE*', which compresses segments into their endpoints. The output is a list of contours where each contour is represented as a list of points outlining the detect boundary [4].

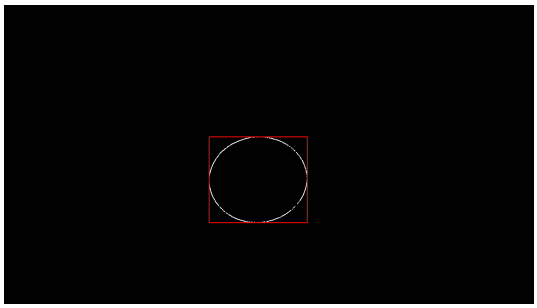
By looking at Figure 5.7 it is clear that there are only three contours, the black zone, the hole and the contour of the whole image. Using a function called '*cv.contourArea()*' it is possible to isolate each one by area. The function takes a contour, represented by a list of points such as the ones retrieve by '*cv.findCountours()*', and uses Green's theorem to integrate the contour points and compute the enclosed area. The output is a scalar value representing the area of the contour in pixel units. The second biggest contour is the one taken into account when isolating the black zone as the biggest one corresponds to the whole image. After retrieving the necessary contour, the function '*cv.boundingRect()*' is applied. This function takes a list of array points outlining the contour and returns the bounding rectangle as a tuple (x,y,w,h) where (x,y) represent the coordinates of the top-left corner of the rectangle and (w,h) represent its width and height,

respectively. This rectangle will be used to extract the region of interest from the image, in this case, the scoring zone of the target.

By using functions such as '*cv.drawContours()*' and '*cv.rectangle()*' it is possible to draw the previous retrieve shapes in a blank image. Results shown in Figure 5.8.



(a) Frontal Target 95431



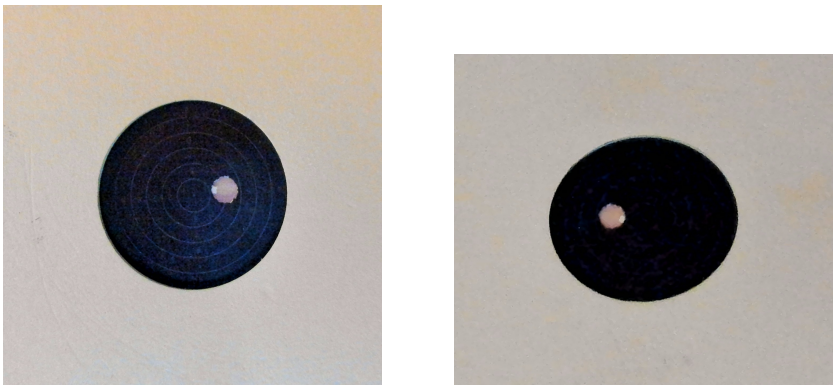
(b) Angled Target 95431

Figure 5.8: Contour and Bounding Rectangle

Note that these were only drawn for visualisation purposes, the program only needs to know the four corner points of the bounding rectangle. At this point, it is visible that the image on top is a circle as its bounding rectangle corresponds to a square while the image on the bottom is an ellipse due to the angle that the photo was taken.

5.2.1.5 Digital Zoom

After discovering the bounding rectangles, the next step is to zoom in on the original image (Figure 5.4) so it can have more pixels inside the black target in order to increase the scoring accuracy. Digital zoom is a technique in computer vision that enhances the perceived resolution of a zoomed object by increasing the number of pixels representing it. Unlike optical zoom, which magnifies an image through lens adjustments, digital zoom employs software algorithms to interpolate additional pixels. This interpolation process involves estimating and generating new pixel values based on the existing pixel data, effectively increasing the image's resolution [19]. To do that, the program takes the previous rectangle, enlarges it to make sure it captures the full target, and crops the original image. Following that, it interpolates the additional pixels and resizes the image to its original 3840 x 2160 pixels.



(a) Frontal Target 95431 (b) Angled Target 95431

Figure 5.9: Zoomed in Targets

After the digital zoom, the previous processes are repeated on these new images in order to get them ready for the next procedure, which is hole detection for the one on the left and conversion from ellipse to circle for the one on the right.

5.2.1.6 Perspective Transformation

The first phase of this transformation is getting a top-down view of the target, i.e, warp the image to make it seem like it is being looked at from a frontal perspective. Four points are needed to execute the transformation which will be the points from the bounding rectangle previously retrieved. The formula for perspective transformation is:

$$[\mathrm{x^i\,y^i\,w^i}] = [u\,v\,1] \begin{matrix} \mathrm{a_{11}} & \mathrm{a_{12}} & \mathrm{a_{13}} \\ \mathrm{a_{21}} & \mathrm{a_{22}} & \mathrm{a_{23}} \\ \mathrm{a_{31}} & \mathrm{a_{32}} & \mathrm{a_{33}} \end{matrix} \tag{5.2}$$

where x^i , y^i and w^i are the three coordinates following perspective transformation, and u , v are the two-dimensional coordinates of the original image. Following the perspective transformation, the following equations display the pixel positions in the new image [33]:

$$x = \left(\frac{x^i}{w^i}\right) = \frac{a_{11}u + a_{21}v + a_{31}\cdot 1}{a_{13}u + a_{23}v + 1\cdot 1} \tag{5.3}$$

$$y = \left(\frac{x^i}{w^i}\right) = \frac{a_{12}u + a_{22}v + a_{32}\cdot 1}{a_{13}u + a_{23}v + 1\cdot 1} \tag{5.4}$$

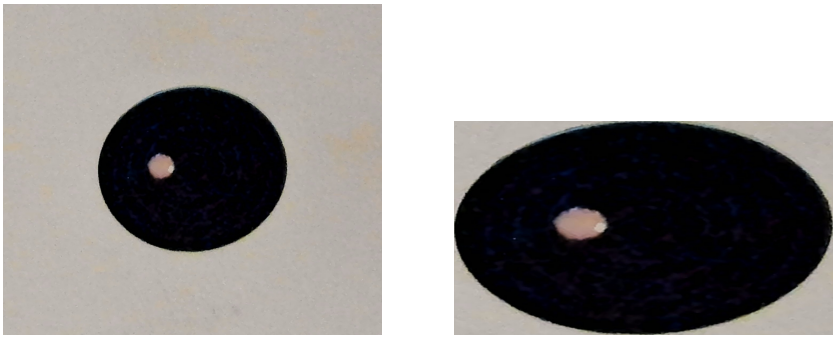
To recreate this in Python, the first step is to create an array containing the four points of the bounding rectangle. By using the value from the previously obtained tuple (x,y,w,h) where (x,y) represent the coordinates of the top-left corner of the rectangle and (w,h) represent its width and height, respectively, the array is:

$$[(x,y)(x+w,y)(x+w,y+h)(x,y+h)]$$

The next step is to calculate the width and height of the new, transformed image, which are equivalent to the width and height of the bounding rectangle previously obtained, as that is the region that is going to be warped. The width and height correspond to the (w,h) from the tuple, respectively. After that, a new array must be created containing the four points of the warped image. Those points are:

$$[(0,0)(w,0)(w,h)(0,h)]$$

Following that, a perspective transformation matrix must be created using those two arrays that can later be used to transpose all the pixels. In order to do that, OpenCV contains a function called '*cv.getPerspectiveTransform()*' that is specifically designed to compute this matrix given two sets of four points: the source and the destination. The source is the array containing the four points of the rectangle while the destination is the array containing the four points of the warped image. The output is a 3x3 matrix that contains the mapping from the source to the destination. Finally, the matrix can be used with the function '*cv.warpPerspective()*' to apply the transformation to the image. Its inputs are the original image and the 3x3 transformation matrix. The output is a warped image with the matrix applied to all the pixels[39, 4]. Results shown in Figure 5.10.



(a) Angled Target 95431 (b) Top-Down View

Figure 5.10: Perspective Transformation

The second phase is to transform the ellipse into a circle. To do that the bounding rectangle can be converted into a square by scaling the y-axis to match the x-axis thus ending up with a circle and preserving the largest amount possible of pixels within the black circle. The program does this by applying an affine transformation which is a linear mapping method that preserves points, straight lines and planes. A 2x3 matrix must be created in order to execute the scaling transformation and can be represented as:

$$[(1,0,0)(0,scaley,0)]$$

where $scaley$ is the scaling factor in the y-axis calculated by the bounding rectangle's tuple:

$$scaley = \frac{x+w}{y+h} \tag{5.5}$$

After obtaining the matrix, the program uses the function '*cv.warpAffine()*' to apply the matrix to all the pixels in the image. Similar to the '*cv.warpPerspective()*' function, it takes a source image and the matrix and returns the transformed image where all pixels are scaled [4]. Results shown in Figure 5.11.

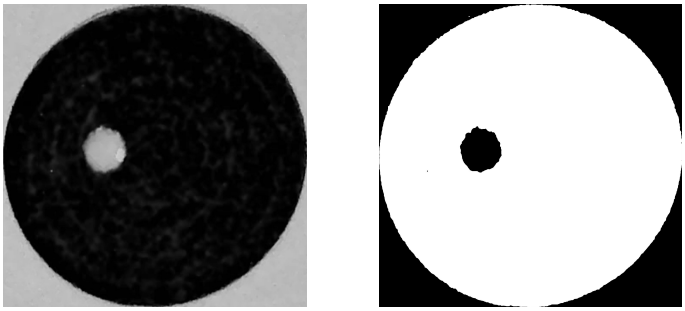


Figure 5.11: Warped Circle

The image is ready for the process of hole detection.

5.2.2 Hole Detection

The first step is to binarise the image again. Using a different approach this time, the image is converted to HSV using the function '*cv.cvtColor()*' with the colour code '*cv.COLOR_BGR2HSV*' and the Value channel is isolated using '*cv.split()*', a function that allows the separation of the different channels that compose a colour space. To complete the binarisation, the function '*cv.inRange()* is used. By giving it a range from 0 to 100, it isolates pixels that are within the specified range by turning them white and the rest black. Figure 5.12 shows the results from the isolation of the Value channel and consequent binarisation.



(a) V (b) Binary

Figure 5.12: Value Channel

In the end, both the hole and the scoring zone are highlighted and the image is ready for another contour detection. By repeating the process from Section 5.2.1.4 and retrieving both the

second and third biggest contour by area, it is possible to get the list of points that represent both circles. With this data the program is ready to proceed to the scoring algorithm.

5.3 Scoring and Displaying Results

This section outlines the methodology for scoring the detected holes and displaying the results. Accurate scoring and clear presentation are essential for providing reliable feedback to the shooter.

5.3.1 Target and Hole Centre

The most important and obvious step is finding both the target's and hole's centre and radius. There are numerous ways to retrieve the centre of a contour but, after various tries, the method with best results was finding the centre of the bounding rectangle. By using the function '*cv.boundingRect()*' and inputting the list of points previously acquired, the tuple (x, y, w, h) from both circles is obtained and can be used to find the centre of the rectangles by doing:

$$\left(\frac{x+w}{2}, \frac{y+h}{2}\right) \tag{5.6}$$

where (x, y) is the top left corner of the rectangle and w, h are the width and height of the rectangle, respectively.

The radius of the scoring zone and the hole can be retrieved using '*cv.minEnclosingCircle()*, which is a function used to find the smallest enclosing circle for a set of points or contour. By inputting the known contours, the respective radii in pixels is acquired.

5.3.2 Pixels per Millimetre

Figure 1.1 shows that the smallest ring has a diameter of 0.5 millimetres, equating to a radius of 0.25mm. For accurate scoring, this radius must be divided into ten parts. As the measurements are done in pixels there as to be a minimum of one pixel per tenth of a score in order for the program to find and score that same region, therefore the minimum resolution required is 1 pixel per 0.025 millimetres, or 40 pixels per millimetre. To determine the conversion rate from millimetres to pixels, measurements known in both units can be used. The radius of the black scoring zone, known in both pixels and millimetres, serves as a basis for this conversion. To accurately calculate the score, it is essential to ensure that the captured image predominantly comprises the scoring area and is captured at the highest possible resolution. This approach maximises the utility of the pixels, ensuring sufficient pixel density within the target. In this work, the average conversion rate was found to be 63 pixels per millimetre, which is sufficient for achieving accurate results.

5.3.3 Scoring Algorithm

After locating both centres and ensuring there are sufficient pixels within the target to accurately calculate the score, the algorithm can be applied. The process begins by checking if the hole hits

the 10 zone. This is done by comparing the distance between the two centres to the radius of the hole. If the distance is shorter, it touches the zone, if it is not, the hole does not touch. The distance between the two points can be calculated using a straightforward formula for the distance between two points:

$$d = \sqrt{(x_b - x_t)^2 + (y_b - y_t)^2} \tag{5.7}$$

where (x_b, y_b) is the centre of the hole and (x_t, y_t) is the centre of the target.

5.3.3.1 Outside 10 Zone

The first step involves subtracting the pellet radius, previously calculated, from the distance between the centres, as outside the 10 zone, the relevant point is the one closest to the target's centre. After that, determine the distance between the closest point and the outermost ring. Lastly, the distance between scoring zones must be calculated. Figure 5.13 shows a close up between scoring rings in official targets. Each grey line correspond to a limit between scoring zones and the difference between scoring zones can be calculated as being a tenth of the distance between two rings, which is 0.25mm.

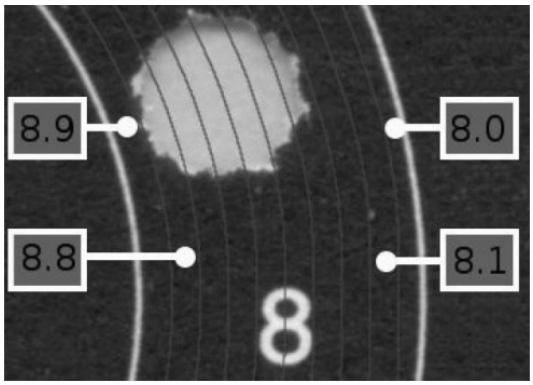


Figure 5.13: Scoring Zones [40]

After obtaining those values, the score is increased by 0.1 based on how many scoring zones fit in the distance between the closest point and the outermost ring, rounded down.

5.3.3.2 Inside the 10 Zone

Here, the calculation becomes more challenging due to minimal margins. When the hole touches the centre, the score is assumed to be 10, leaving only the tenths to be calculated. The distance from the centre of the hole to the 10 zone ring is used, with the scoring zones now being 0.025mm, as the radius is 0.25mm. Using the same principle as before, the score is increased by 0.1 based on how many scoring zones fit in the distance between the centre of the hole and the 10 zone ring, rounded down.

5.3.4 Displaying Results

The final step involves displaying the results in an intuitive and user-friendly manner. This includes immediate feedback and analysis report.

A real-time virtual target is created to show the shooter's score and the location of each hole on a digital representation of the target. To do this, the program must first create a blank image using the function '*np.ones()*. By inputting the desired image dimensions (height and width), the number of channels (1 for grayscale, 3 for RGB) and the standard image data type '*np.uint8*', the function returns an image with the specified dimensions, channels and all the pixels set to 1, which can be multiplied by any number from 0 to 255 in order to become a shade of grey thus ending up with a fully coloured background [4, 30]. This virtual target has dimensions of 1920x1080, 3 channels and a initial pixel value of 126. The next step is to draw the black scoring zones and the rings within it in order to replicate the official targets. To execute this task, the algorithm uses OpenCV's function '*cv.circle()*. By passing the blank image, the centre, the radius, the colour of the circle and the thickness of the outline, the function returns an image with the specified circle drawn [4, 30]. The centre is the centre of the image and is the same for all circles. The radius of each one corresponds to the original dimensions converted to pixels using the ratio previously obtained. The black scoring circle has black colour and a thickness of -1, meaning the circle will be completely filled. The rest of the rings have white colour and a thickness of 1. The final result is shown in Figure 5.14

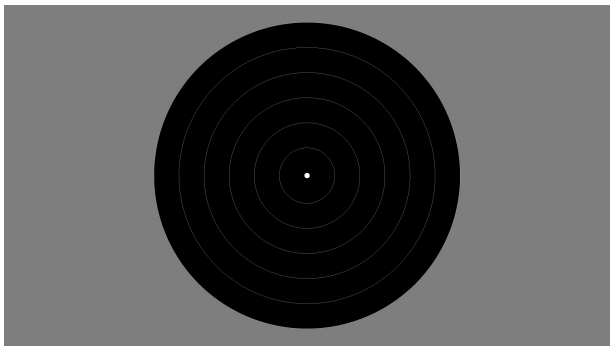


Figure 5.14: Virtual Target

The final step is to draw the holes in the newly created virtual target. To do this, translation vectors must be calculated in order to accurately depict the original shot. It can be done by translating the centre of the target to the centre of the new image, therefore creating the vectors that can later be used to translate the holes. The translation vector from the target's centre (x_1, y_1) to the image's centre (x_2, y_2) is given by:

$$\text{Translation vector} = \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$

where:

$$\Delta x = x_2 - x_1$$

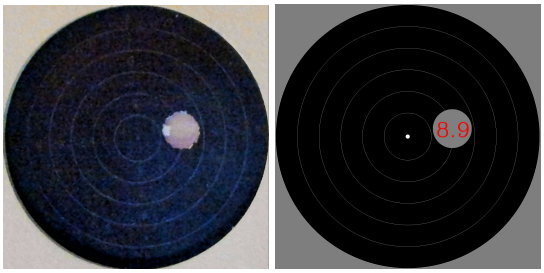
$$\Delta y = y_2 - y_1$$

So, the translation vector **T** from point 1 to point 2 is:

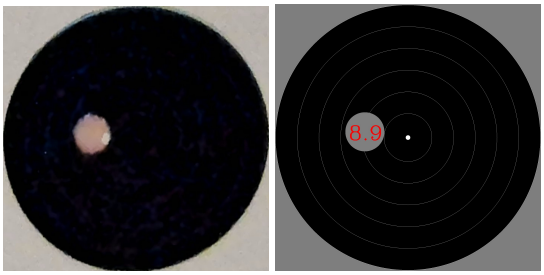
$$\mathbf{T} = \begin{pmatrix} x_2 - x_1 \\ y_2 - y_1 \end{pmatrix}$$

By applying this vector to the hole's centre, its position on the virtual target can be discovered. Using the same function '*cv.circle()*' and inputting the translated centre, the converted radius, a colour and a thickness of -1, the pellet hole can be accurately represented in the virtual target. In addition, to facilitate the shooter's experience, the score can also be displayed by using the function '*cv.putText()* and passing the score previously calculated. A few other variables such as image, coordinates of the text in the image, font type and scale, colour and thickness must also be inputted so the software knows where and how to place the text/score. Figure 5.15 shows the original target and its virtual representation.

The average time between the detection of the shot and the display of results is less than one second.



(a) Frontal Target 95431



(b) Angled Target 95431

Figure 5.15: Original and Digital Targets

5.3.4.1 Data Storage

To facilitate long-term analysis and comparison, all scoring data and results are stored in a file that can be accessed later for offline analysis or sharing.

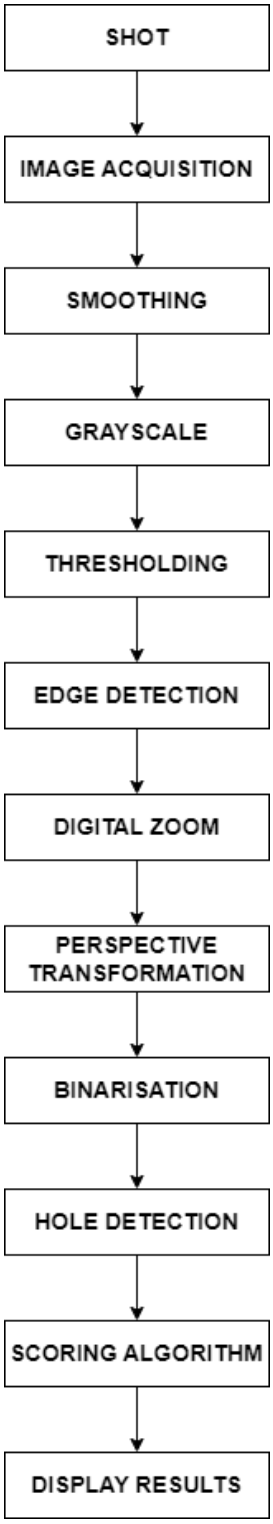


Figure 5.16: Methodology Flowchart

Chapter 6

Results and Discussion

The results obtained from the methods and techniques used in this dissertation are presented in this chapter, covering two important aspects of this study: Automatic Scoring Algorithm Accuracy and Image Processing for Target Detection. This chapter's primary goal is to provide the techniques' results and discuss their consequences within the context of the dissertation. It's essential to use both subjective assessments and quantifiable measures to evaluate the effectiveness and quality of the scoring system. The determination of the final scoring efficiency is the main topic of this chapter, along with two important metrics: the statistical analysis of the scoring precision and the subjective evaluation.

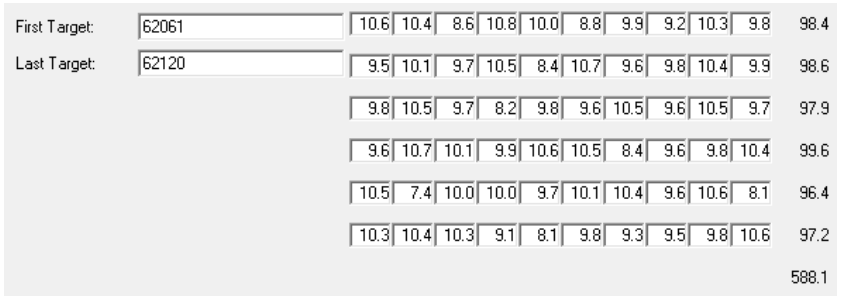
6.1 Automatic Scoring Algorithm Accuracy

This section presents the results obtained from the automatic scoring algorithm when applied to the dataset, including both frontal and angled images. The final digital targets will be provided along with the official scoring cards. The main objective is to evaluate the accuracy and reliability of the system by comparing its scoring outputs to the official scores recorded during the shooting events. This comparison will provide valuable insights into the algorithm's performance, highlighting areas of high accuracy as well as any limitations that may need further investigation. Through statistical evaluations and visual comparisons, the robustness and potential improvements of the scoring system will be explored.

6.1.1 Official Scores

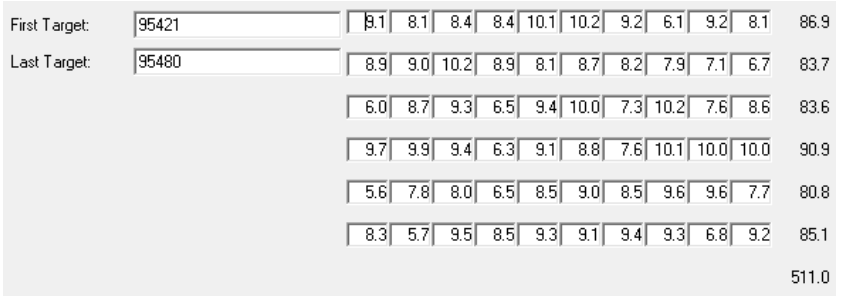
The official scoring cards will be presented in this subsection, with the results displayed in a systematic manner. The digital targets will later be presented following an order starting at the top and going from left to right regarding the official cards. This ensures that readers can compare and analyse the scoring between the two.

Figure 6.1a shows Card A which contains the scores from targets 62061 to 62120 which will be the first batch analysed while Figure 6.1b shows Card B which contains the scores from targets 95421 to 95480.



First Target:	62061										
Last Target:	62120										
		10.6	10.4	8.6	10.8	10.0	8.8	9.9	9.2	10.3	9.8	98.4
		9.5	10.1	9.7	10.5	8.4	10.7	9.6	9.8	10.4	9.9	98.6
		9.8	10.5	9.7	8.2	9.8	9.6	10.5	9.6	10.5	9.7	97.9
		9.6	10.7	10.1	9.9	10.6	10.5	8.4	9.6	9.8	10.4	99.6
		10.5	7.4	10.0	10.0	9.7	10.1	10.4	9.6	10.6	8.1	96.4
		10.3	10.4	10.3	9.1	8.1	9.8	9.3	9.5	9.8	10.6	97.2
												588.1

(a) Card A



First Target:	95421										
Last Target:	95480										
		8.1	8.1	8.4	8.4	10.1	10.2	9.2	6.1	9.2	8.1	86.9
		8.9	9.0	10.2	8.9	8.1	8.7	8.2	7.9	7.1	6.7	83.7
		6.0	8.7	9.3	6.5	9.4	10.0	7.3	10.2	7.6	8.6	83.6
		9.7	9.9	9.4	6.3	9.1	8.8	7.6	10.1	10.0	10.0	90.9
		5.6	7.8	8.0	6.5	8.5	9.0	8.5	9.6	9.6	7.7	80.8
		8.3	5.7	9.5	8.5	9.3	9.1	9.4	9.3	6.8	9.2	85.1
												511.0

(b) Card B

Figure 6.1: Official Scoring Cards

6.1.2 Generated Scores and Comparison

This subsection will present the scores generated by the algorithm and a comparison to the official scores. It will be divided into two batches corresponding to the two official scoring cards with an aim to evaluate the accuracy and provide a clear understanding of the system's strengths and areas for improvement. To provide a clear comprehension to the readers, only the scores will be shown here and every digital target will be displayed in Appendix A.

Using the previously described methods, both frontal and angled images produced very efficient scoring results. Apart from two specific targets that will be discussed later, every other score corresponded precisely to the official one, demonstrating the reliability of this image-based system. The consistency across different photographic angles validates the effectiveness of the algorithm, confirming that it can interpret and score images regardless of the perspective from which they are taken. In the end, after applying the algorithm to 120 targets, using two different perspectives, the system only failed twice.

Figure 6.2 and Figure 6.3 show generated (coloured background) and official (white background) scores side by side for each scoring card.

10.6	10.6	10.4	10.4	8.6	8.6	10.8	10.8	10.0	10.0	8.8	8.8	9.9	9.9	9.2	9.2	10.3	10.3	9.8	9.8
9.5	9.5	10.1	10.1	9.7	9.7	10.5	10.5	8.4	8.4	10.7	10.7	9.6	9.6	9.8	9.8	10.4	10.4	9.9	9.9
9.8	9.8	10.5	10.5	9.7	9.7	8.2	8.2	9.8	9.8	9.6	9.6	10.5	10.5	9.6	9.6	10.5	10.5	9.7	9.7
9.6	9.6	10.7	10.7	10.1	10.1	9.9	9.9	10.6	10.6	10.5	10.5	8.4	8.4	9.6	9.6	9.8	9.8	10.4	10.4
10.5	10.5	7.4	7.4	10.0	10.0	10.0	10.0	9.7	9.7	10.1	10.1	10.4	10.4	9.6	9.6	10.6	10.6	8.1	8.1
10.3	10.3	10.4	10.4	10.3	10.3	9.1	9.1	8.1	8.1	9.8	9.8	9.3	9.3	9.5	9.5	9.8	9.8	10.6	10.6

(a) Frontal Targets

10.6	10.6	10.4	10.4	8.6	8.6	10.8	10.8	10.0	10.0	8.8	8.8	9.9	9.9	9.2	9.2	10.3	10.3	9.8	9.8
9.5	9.5	10.1	10.1	9.7	9.7	10.5	10.5	8.4	8.4	10.7	10.7	9.6	9.6	9.8	9.8	10.4	10.4	9.9	9.9
9.8	9.8	10.5	10.5	9.7	9.7	8.2	8.2	9.8	9.8	9.6	9.6	10.5	10.5	9.6	9.6	10.5	10.5	9.7	9.7
9.6	9.6	10.7	10.7	10.1	10.1	9.9	9.9	10.6	10.6	10.5	10.5	8.4	8.4	9.6	9.6	9.8	9.8	10.4	10.4
10.5	10.5	7.4	7.4	10.0	10.0	10.0	10.0	9.7	9.7	10.1	10.1	10.4	10.4	9.6	9.6	10.6	10.6	8.1	8.1
10.3	10.3	10.4	10.4	10.3	10.3	9.1	9.1	8.1	8.1	9.8	9.8	9.3	9.3	9.5	9.5	9.8	9.8	10.6	10.6

(b) Angled Targets

Figure 6.2: Generated Scores Card A

9.1	9.1	8.1	8.1	8.4	8.4	8.4	8.4	10.1	10.1	10.2	10.2	9.2	9.2	6.1	6.1	9.2	9.2	8.1	8.1
8.9	8.9	9.0	9.0	10.2	10.2	8.9	8.9	8.1	8.1	8.7	8.7	8.2	8.2	7.9	7.9	7.1	7.1	6.7	6.7
6.0	6.0	8.7	8.7	9.3	9.3	6.5	6.5	9.4	9.4	10.0	10.0	7.3	7.3	10.2	10.2	7.6	7.6	8.6	8.6
9.7	9.7	9.9	9.9	9.4	9.4	6.3	6.3	9.1	9.1	8.8	8.8	7.6	7.6	10.1	10.1	10.0	10.0	10.0	10.0
5.8	5.6	7.8	7.8	8.0	8.0	6.5	6.5	8.5	8.5	9.0	9.0	8.5	8.5	9.6	9.6	9.6	9.6	7.7	7.7
8.3	8.3	5.8	5.7	9.5	9.5	8.5	8.5	9.3	9.3	9.1	9.1	9.4	9.4	9.3	9.3	6.8	6.8	9.2	9.2

(a) Frontal Targets

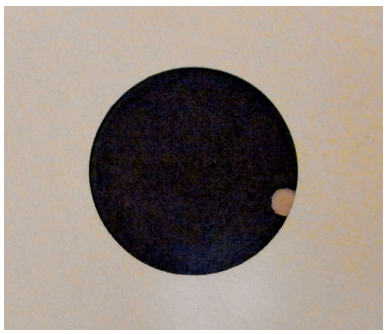
9.1	9.1	8.1	8.1	8.4	8.4	8.4	8.4	10.1	10.1	10.2	10.2	9.2	9.2	6.1	6.1	9.2	9.2	8.1	8.1
8.9	8.9	9.0	9.0	10.2	10.2	8.9	8.9	8.1	8.1	8.7	8.7	8.2	8.2	7.9	7.9	7.1	7.1	6.7	6.7
6.0	6.0	8.7	8.7	9.3	9.3	6.5	6.5	9.4	9.4	10.0	10.0	7.3	7.3	10.2	10.2	7.6	7.6	8.6	8.6
9.7	9.7	9.9	9.9	9.4	9.4	6.3	6.3	9.1	9.1	8.8	8.8	7.6	7.6	10.1	10.1	10.0	10.0	10.0	10.0
5.8	5.6	7.8	7.8	8.0	8.0	6.5	6.5	8.5	8.5	9.0	9.0	8.5	8.5	9.6	9.6	9.6	9.6	7.7	7.7
8.3	8.3	5.8	5.7	9.5	9.5	8.5	8.5	9.3	9.3	9.1	9.1	9.4	9.4	9.3	9.3	6.8	6.8	9.2	9.2

(b) Angled Targets

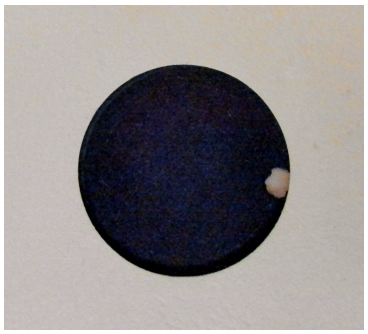
Figure 6.3: Generated Scores Card B

6.1.2.1 Wrong Scores

Figure 6.3 shows that targets 95461 and 95472 are miss classified in both images. To explain why, a few details must be mentioned. In Section 3.1 the current setup of the system was explained as being a frame with a hole the size of the black ring of the official target in order to be used repeatedly by simply replacing the background, in this case, black tape. Later on, in subsection 5.1.3, it was explained that, instead of black tape, previously marked targets were being used as background as a means to maximise dataset acquisition. With that in mind, Figure 6.4 shows the frontal images taken of these two targets.



(a) 95461



(b) 95472

Figure 6.4: Miss Classified Targets

By closely inspecting both images, it becomes evident that the frame cuts a piece of the hole, leaving the circle incomplete. When the algorithm analyses the images, it cannot capture the whole circle leaving it with an incomplete contour that, when applied the centre detection methods, does not give an accurate representation of the centre, therefore, miss calculates the score as well.

There are a few workarounds that can be implemented in order to try and improve the system when it comes to these situations. The most used and worked on method is Hough Transform [35]. It can overcome the former shortages and retrieve the centre from incomplete circles such as the ones encountered on these specific cases. Its implementation on this system is a more difficult and still being worked on as the current methods for retrieving and detection centre are more accurate than Hough Transform when used on full circles.

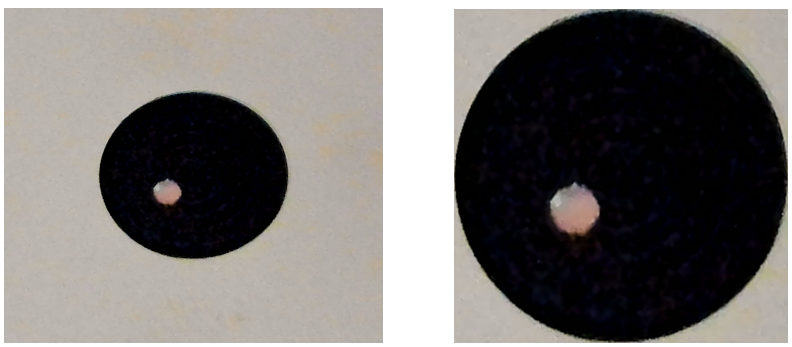
6.2 Image Processing for Target Detection

In this section, the goal is to visually analyse the processes each image goes through and inspect if there are any inconsistencies or improvements that could be done in order to improve the current state of the system. By comparing it to existing non-official ones, it is possible to understand what additions can be implemented to expand its use beyond what was originally planed. The example used in this section is target 95434.

6.2.1 Image Correction

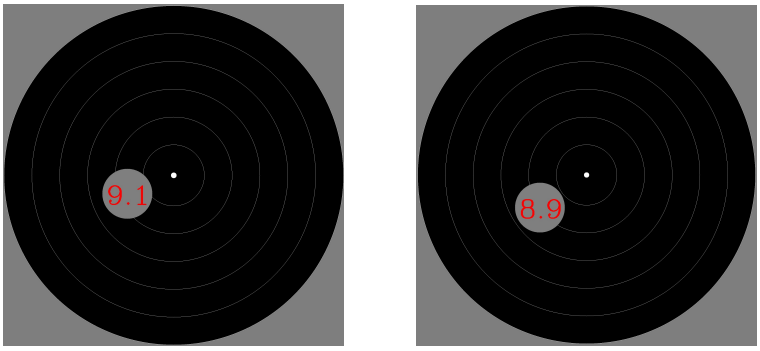
Perspective transformation and image correction are crucial for mitigating distortions caused by the camera's angle and ensuring accurate scoring. These are the first processes applied to the captured images and, by demonstrating the effectiveness of these techniques, their contribution to the overall precision of the system is highlighted. These methods are used by most existing systems that analyse angled photos such as [51, 40, 33].

Figure 6.5 shows the before and after using image correction while Figure 6.6 shows the difference in the final score with and without these techniques.



(a) Before (b) After

Figure 6.5: Image Correction



(a) Without Correction (b) With Correction

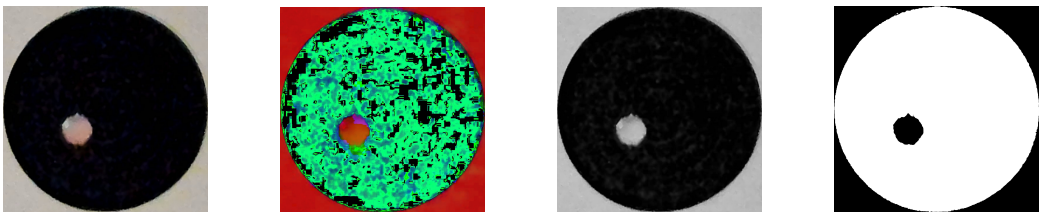
Figure 6.6: Final Scores

After observing the previous figures, it becomes obvious that these are necessary preprocessing methods that guarantee a more accurate and effective scoring.

6.2.2 Image Segmentation

Following a correctly done perspective transformation, a image segmentation is necessary to divide the image into its constituent objects, in this case separate the target from the frame and the bullet hole from the target. The fundamental approach to image segmentation is thresholding which converts the image into a binary image. Most existing systems use grayscaling as the middle step when thresholding an image as the original image must first be represented by a single colour channel. This system uses the Value (V) channel from the HSV colour model. After various experiments it was the most successful method and a way to apply as many computer vision techniques as possible.

Figure 6.7 shows the different stages that a image goes through before being completely binarised.



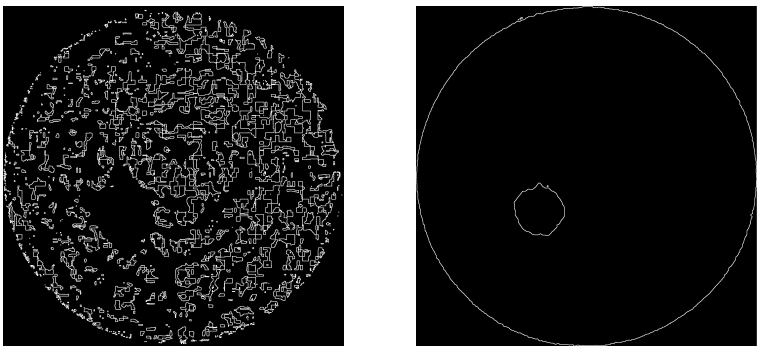
(a) Original (b) HSV (c) V (d) Binary

Figure 6.7: Binarization process

In the end, it is visible that the process of binarisation was successfully completed as there is a complete distinction between the target and the hole.

The next step in image segmentation is contour detection which can be obtained with any edge detection method. It allows to fully isolate the target and the bullet hole and can be used to obtain useful information such as the centre and the diameter.

Figure 6.8 shows the difference between applying edge detection on the original image and on the binary image.



(a) Original (b) Binary

Figure 6.8: Contours

As is evident, even the smallest noises and features on an image are captured by the edge detection methods, therefore it becomes clear that the previous methods are crucial for hole detection and later scoring.

Results shown along this section prove that the techniques used were successful in attaining the objectives of the particular setup that was built for this project. In the following subsection, some features will be presented that could potentially be added to this system as a way to expand its use.

6.3 Extra Features

Finally, this section explores features from other advanced systems that could potentially be integrated into this one if there is a need to expand the system's capabilities beyond what was originally planned. The analysis includes a review of methodologies used for better hole isolation and recognition. By examining these features, enhancements that could further increase the precision and functionality of this system can be identified. The integration of these advanced features can create a more robust and versatile scoring solution that meets the evolving needs of shooting sports and training applications.

6.3.1 Image Differencing

Image Differencing is a method used in real-time processing where two images are compared, typically consecutive frames in a video, to detect changes between them [50]. The objective is to locate areas in the updated image where modifications have taken place, isolating newly added objects. In automatic shooting scoring systems, by subtracting the pixels in the same position in both images, the algorithm is able to detect new holes and obtain the position of each one [51].

By comparing each consecutive frame it can always identify new holes. It's a simple method and gives good results in real-time analysis [42].

This particular method is very useful in any type of image-based scoring algorithm, but it needs to have consecutive images without any interference besides the newly induced holes. The dataset used in this system was not obtained in a stable setup and relied on previously marked targets, therefore there were numerous differences between the original target and the captured ones, making this method a non-viable solution for the current state of the system.

Figure 6.9 shows this method's results in P. D. Widayaka *et al* system. The top two images are the consecutive frames, the bottom left is after applying the method and the bottom right is the final result after applying some morphological operations.

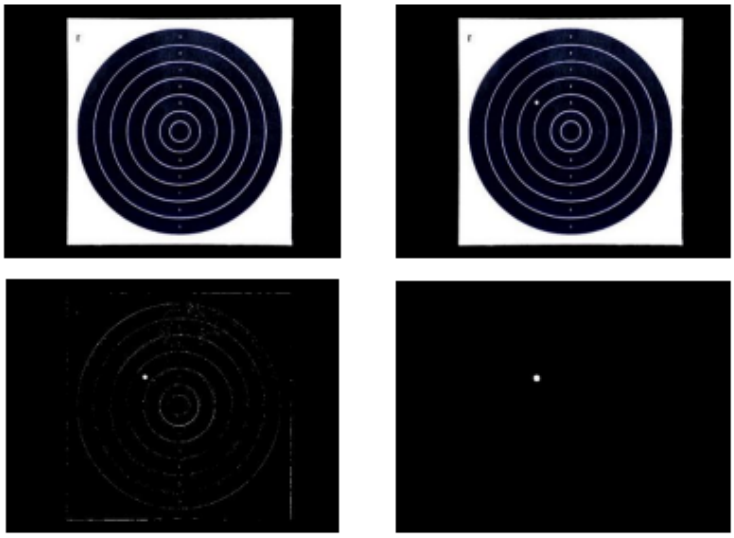


Figure 6.9: Subtraction Process [51]

6.3.2 Overlapping Holes Recognition

In some scenarios, automatic shooting scoring systems are required to detect and score multiple holes within one target. Chances are that one or more holes will overlap creating weird shapes that normal algorithms can't recognise as being two or more different circles. The current state of this system can still accurately score multiple holes in the same target but only if they are not overlapping. The most common solution to this problem is using the Circular Hough Transform (CHT). It is a specialisation of the Hough Transform that extracts circular shapes from an image [38]. By giving it specific values, such as the radius, it can identify all the other parameters and break down shapes in circles that fit that criteria [27]. By employing this technique in scoring systems, the algorithm can detect and score overlapping circles.

As its implementation is complex and not essential for this work, it ended up not being implemented in this version of the system.

6.3.3 Serial ID Identification

ISSF official targets contain an unique ID number that allows them to be directly linked to a shooter or batch. By making a system capable of identifying the unique cards, the need for human interaction is greatly reduced as it becomes capable of automatically assign each score to its shooter.

Due to the way official scanners are built, there are only four spots where the ID number can be placed in the target [51], therefore there are only four regions of interest (ROI) that need to be analysed in order to retrieve it. Figure 6.10 shows where they are.

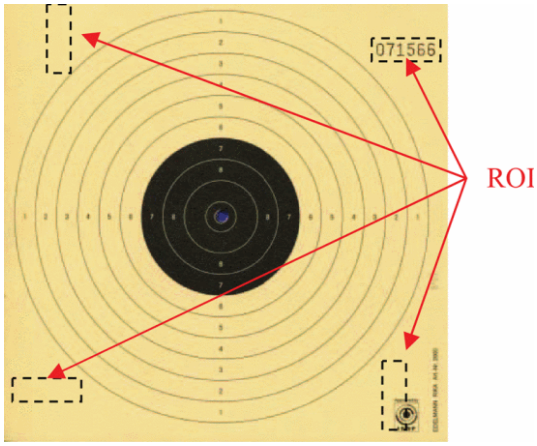


Figure 6.10: ID Number Placement [51]

The easiest way to check if the ROI contains the serial ID is to check if there is a minimal number of black pixels within it, as the number is always printed in black. If it does not contain the ID then check the next ROI until the right one is discovered. After finding it, contour detection is applied to segment the digits of the serial number. Lastly, a template matching technique is used to identify each digit [51].

The current algorithm is implemented to analyse a target frame and not the official targets and, therefore, this feature was not included as there are no ID numbers to be identified.

6.4 Final Discussion

The statistical analysis of the automatic shooting scoring system implemented in this dissertation reveals impressive performance across various methodologies, with the algorithm demonstrating significant scoring accuracy, achieving near-perfect efficiency. The hole recognition methods also displayed good effectiveness, although there was a slight decrease in performance for deformed hole. Notably, the target detection algorithm consistently achieved accurate outcomes, allowing the perspective transformation method to also yield great results. All methodologies used produced satisfactory and consistent results overall.

In addition to the analysis of current methodologies, potential future enhancements to this automatic shooting scoring system were also discussed. Extra features, such as image subtraction, overlapping holes differentiation and serial number identification, could further enhance the system's accuracy and usability in different and less optimal conditions. The implementation of machine learning techniques to adaptively improve detection and scoring under varying conditions is another promising avenue for future research.

In conclusion, the automatic shooting scoring system proposed in this dissertation produced excellent results in most scenarios, achieving almost perfect accuracy. However, it is important to note that for holes with distorted shapes, the results were less reliable. This highlights the need for further refinement and optimisation of the algorithms to handle challenging conditions more effectively. The potential for future enhancements presents a promising direction for further improving the system's performance and robustness.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

The focus of this dissertation is to explore the application of computer vision techniques to enhance the accuracy and reliability of automatic image-based shooting scoring systems. Specifically, the objective is to developed a system that accurately scores shooting targets by mitigating errors caused by various environmental and technical factors while keeping a low-cost build and implementation. Throughout the research, a number of important objectives were pursued in order to accomplish this goal.

Firstly, the preceding analysis of shooting systems conducted in the case study proved to be crucial in the context of scoring system optimisation. Understanding the shooting conditions enables a better selection of appropriate detection and scoring algorithms. This information is vital for maximising scoring accuracy while ensuring the reliability and consistency of the system.

Furthermore, the system's setup played a critical role in accurately detecting the impact points under different shooting scenarios. Through the incorporation of perspective transformations, the algorithm achieves a more straightforward analysis of the targets, leading to a more reliable and effective scoring system.

Two different circle detection processes, namely Circular Hough Transform and Contour Detection, were investigated to determine which method would yield the highest accuracy in the given scenario. Firstly, the Circular Hough Transform proves to be advantageous for the overall accuracy of the system when faced with multiple shots, as it allows for a more flexible and comprehensive identification of circles within the same image dimensions. Secondly, the Contour Detection promotes superior accuracy for detecting individual circles, as it operates within a more precise detection framework. The increased accuracy in single holes results in higher total detection accuracy for this system as a whole, as its main purpose is to detect individual shots.

In addition to optimising the accuracy of the scoring system, this study also addressed the process of creating a visual display of scores to enhance its feasibility in practical applications. The approach used relied on translation vectors calculated through the centres of the original and digital targets to mitigate the visual deviation issue during the display phase. Although there was a

minor deviation in the overall visual accuracy, this approach yields significant benefits in terms of visual results, as the variations do not induce error to the viewers. As a result, the scoring display system exhibits a constant and predictable accuracy, which greatly facilitates viewer experience and the transfer of information to the shooters.

Finally, the proposed methods were rigorously evaluated in terms of their accuracy, robustness and performance. The efficiency of the proposed strategies was evaluated through an in-depth review, offering valuable insights into their advantages and disadvantages.

These contributions promote the advancement of automatic image-based shooting scoring technology by providing insights into the optimisation between accuracy and consistency. By adopting approaches for scoring accuracy maximisation and visual display, shooting scoring systems can become more efficient, reliable, and seamlessly integrated into competitive or training shooting environments. This research paves the way for future studies and practical implementations of automatic shooting scoring systems, fostering the widespread adoption of advanced scoring technologies.

7.2 Future Work

While not all of the ideas were put into practice or tested, many were carefully thought through for the automatic image based shooting scoring system during the course of this paper. The first area of focus was Image Differencing. This method emphasises the significance of accurately identifying and aligning the differences between sequential shooting frames. To achieve this, a preliminary step of building a steady setup and acquiring a reference target image is indispensable. Obtaining a more precise acquisition of these frames will lead to improved shot detection and scoring accuracy, consequently resulting in more robust outcomes.

Another future development is the recognition and handling of overlapping holes. When multiple shots land close to each other or overlap, it is crucial to identify and separate each hole accurately. One promising approach involves using the Circular Hough Transform to detect and map the boundaries of each shot hole. By applying this technique, the system can analyse shot patterns and identify overlapping areas. This method leverages advanced image processing techniques and warrants further investigation to optimise its effectiveness.

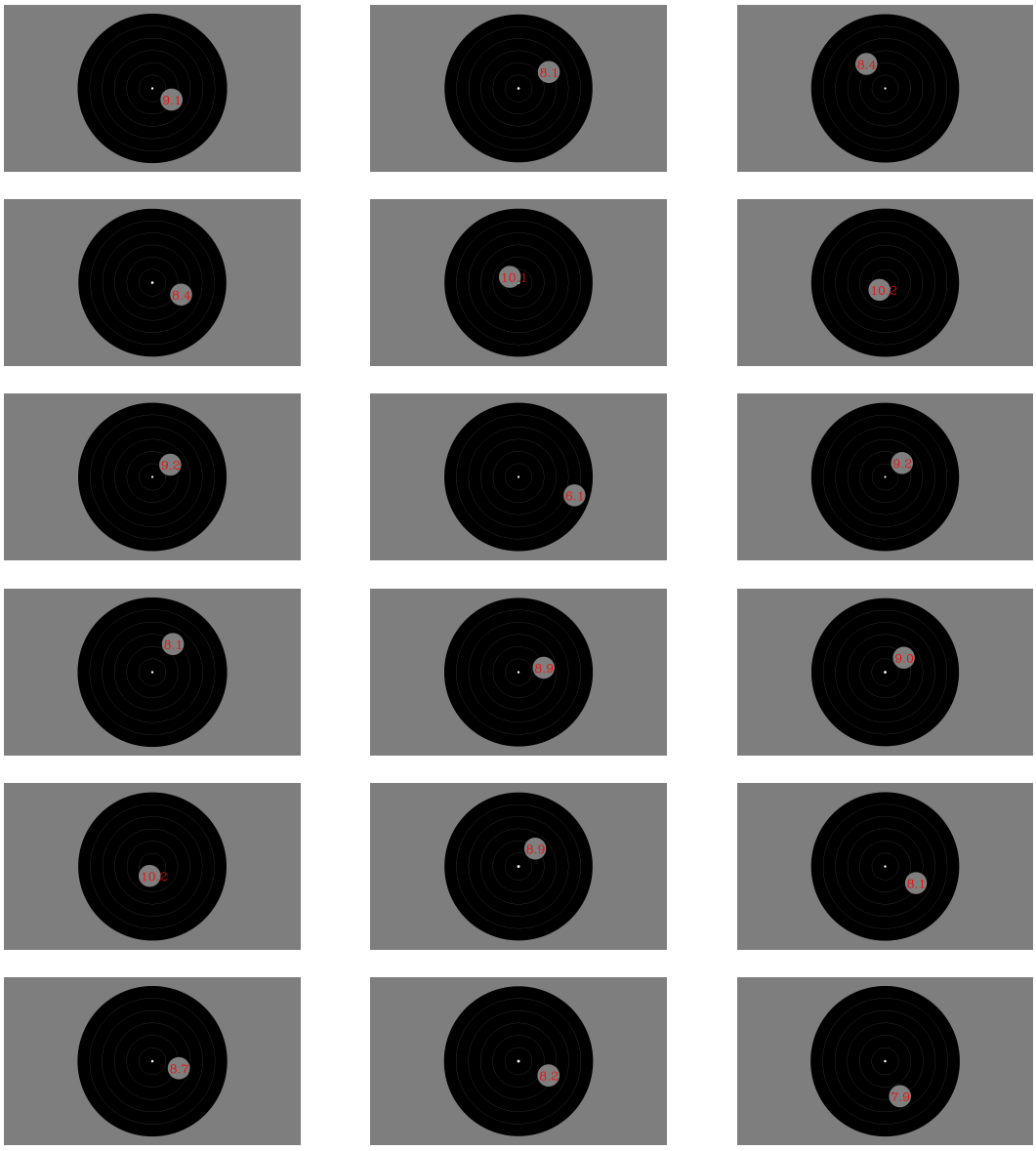
To conclude, in the event that there's a need to identify each target, the field of computer vision presents viable alternatives. For instance, techniques can be employed for Serial ID Identification, where each shot is tagged and tracked across multiple frames. This approach proves advantageous due to the ability of the system to automatically identify and assign each target to its shooter or category. By employing advanced algorithms, it becomes possible to improve the detection and scoring of each shot, even in complex scenarios with multiple shooters or targets. The automatic image based shot scoring system's overall accuracy, dependability, and efficiency might all be greatly improved by employing these methodologies.

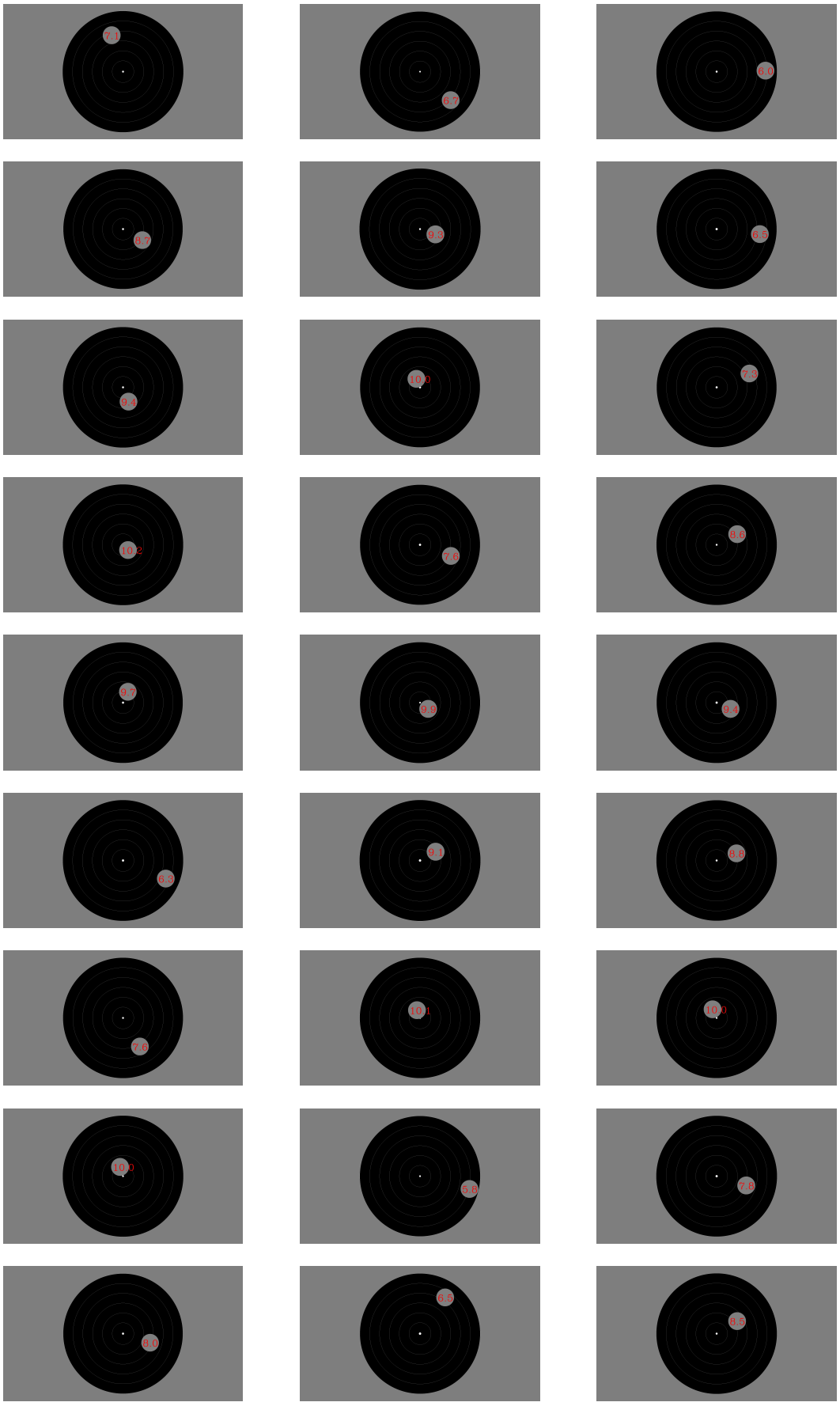
Appendix A

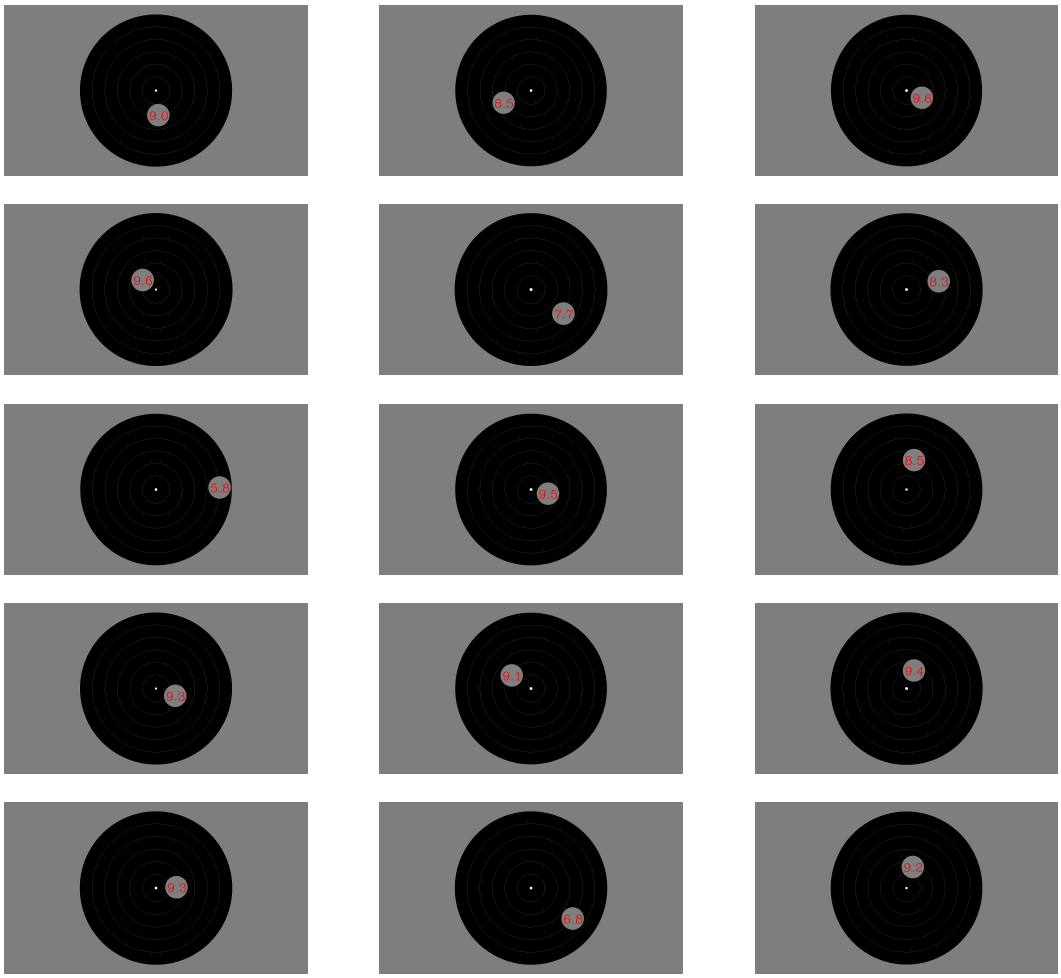
Appendix

A.1 Complementary Results

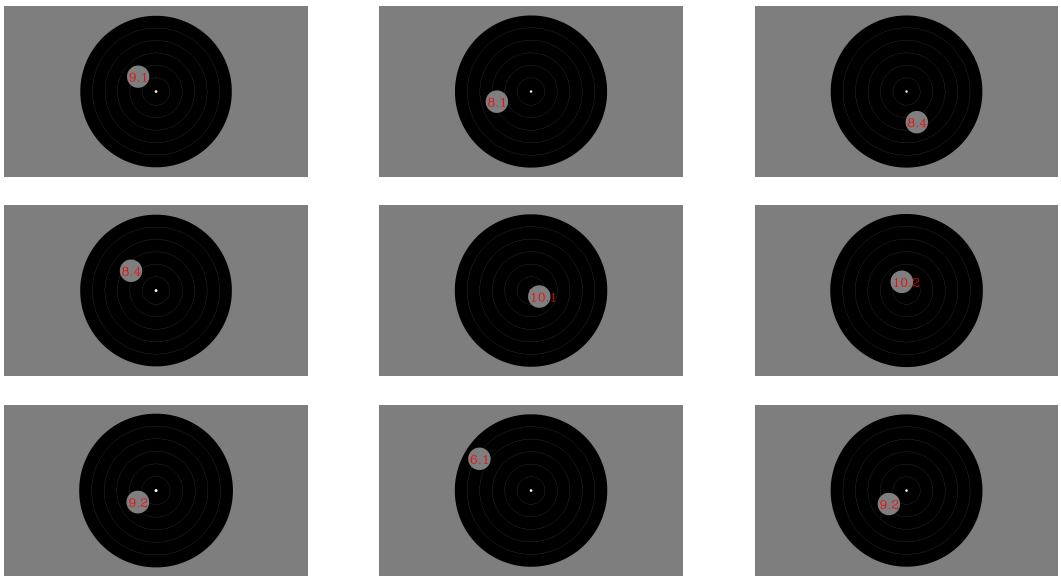
A.1.1 Frontal Targets 95421-95480

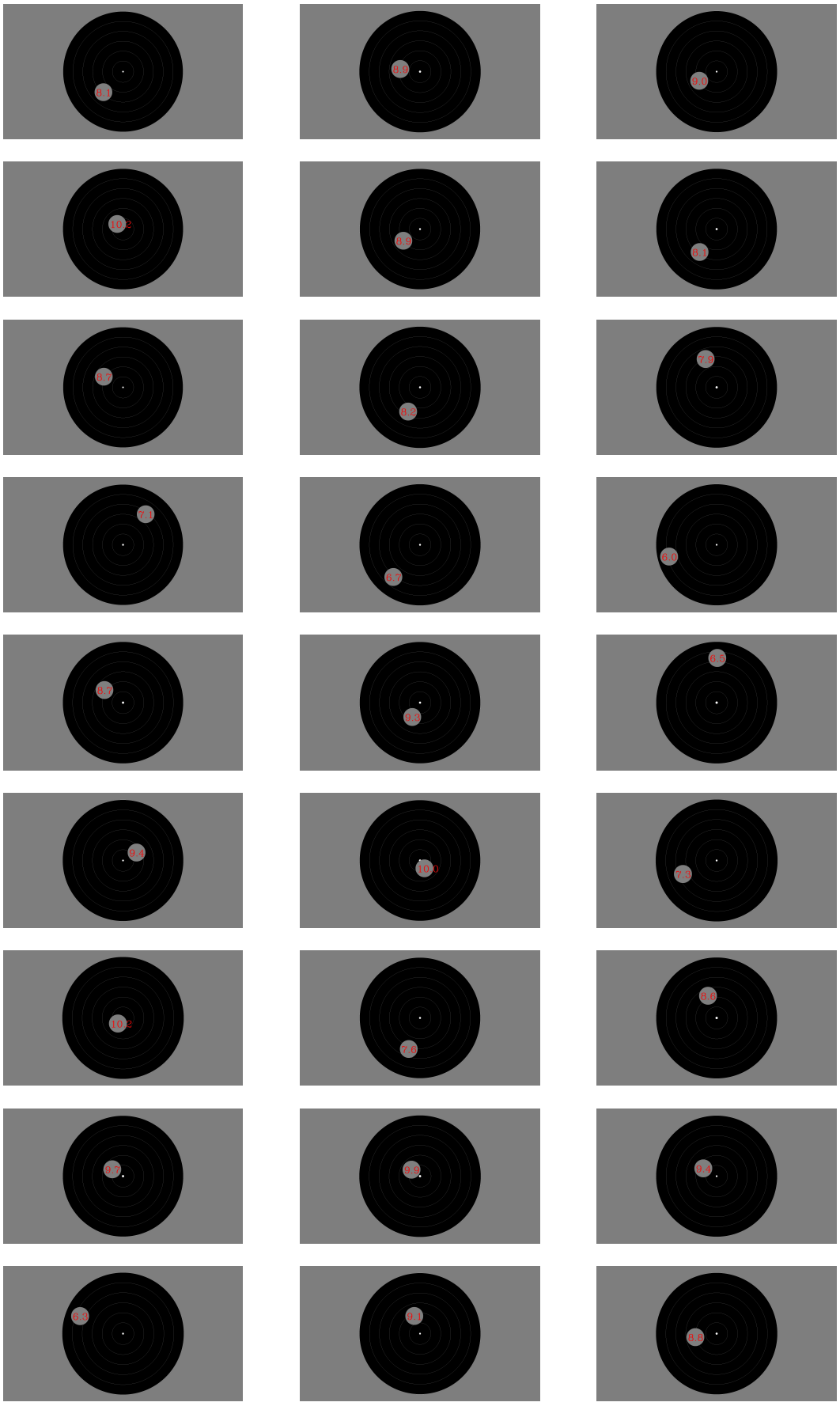


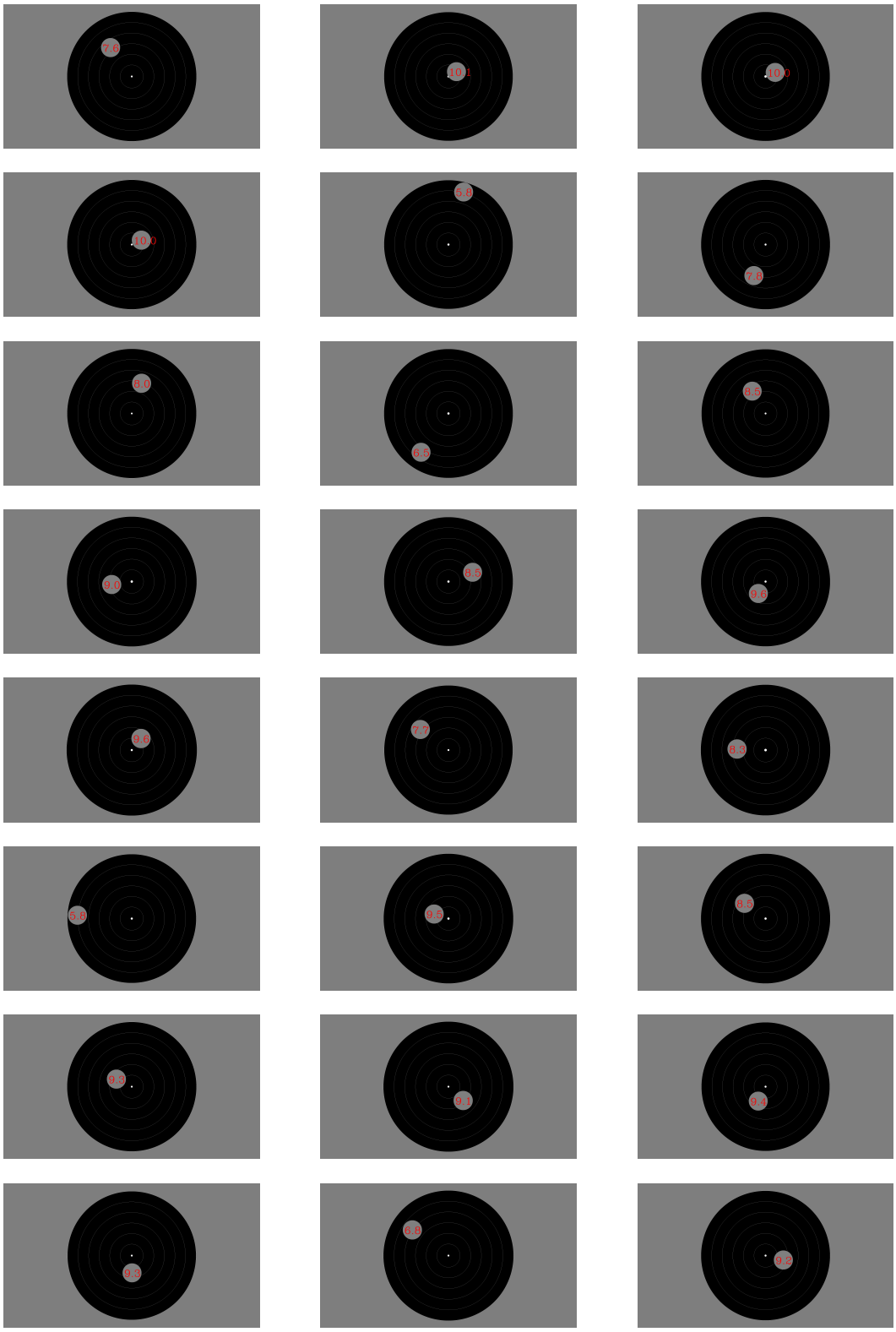

7.1
6.7
6.0
8.7
9.3
8.5
9.4
10.0
7.3
10.2
7.6
8.6
9.7
9.4
9.4
6.9
9.1
9.8
7.6
10.1
10.0
10.0
6.8
7.8
8.0
6.5
8.5



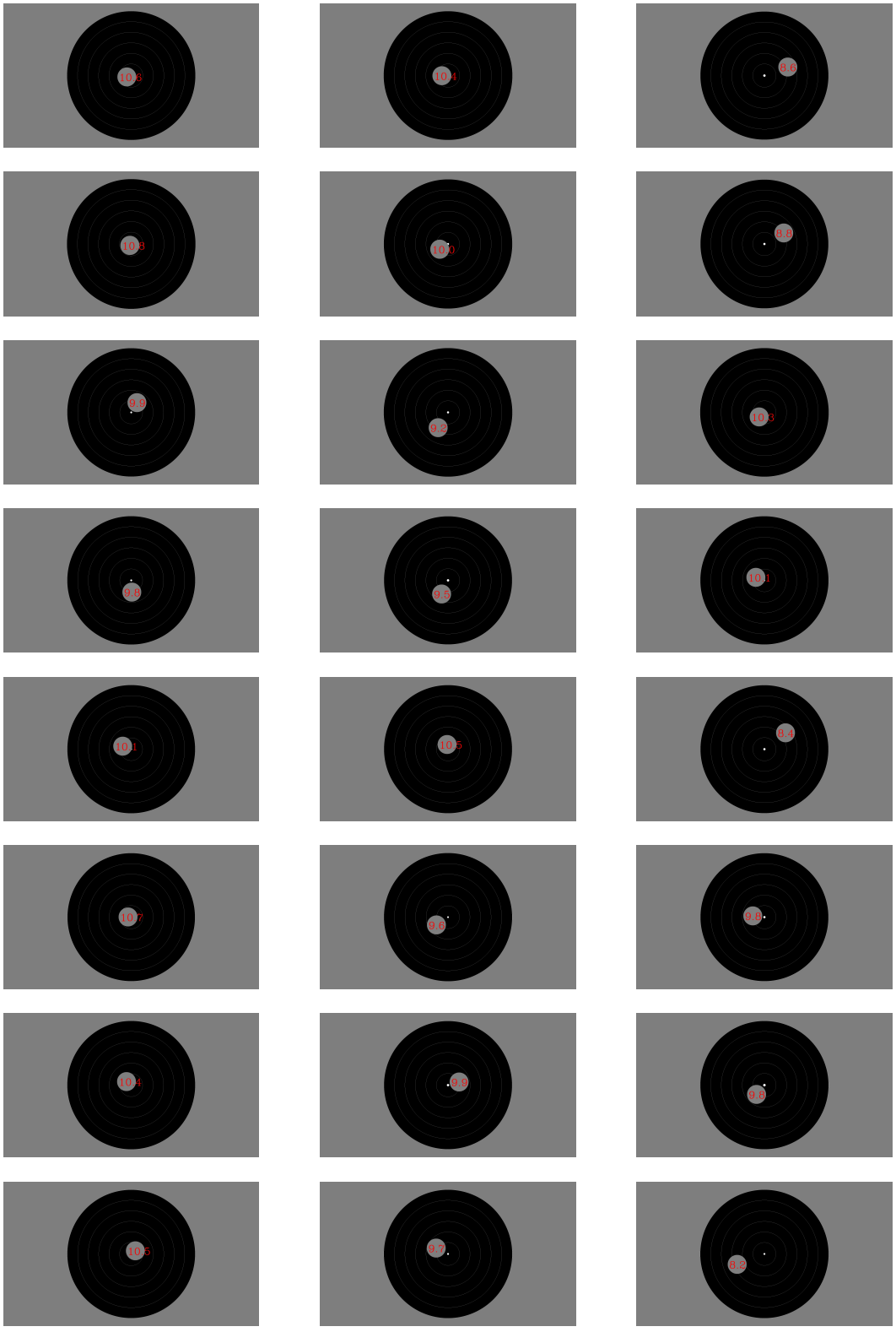
A.1.2 Angled Targets 95421-95480

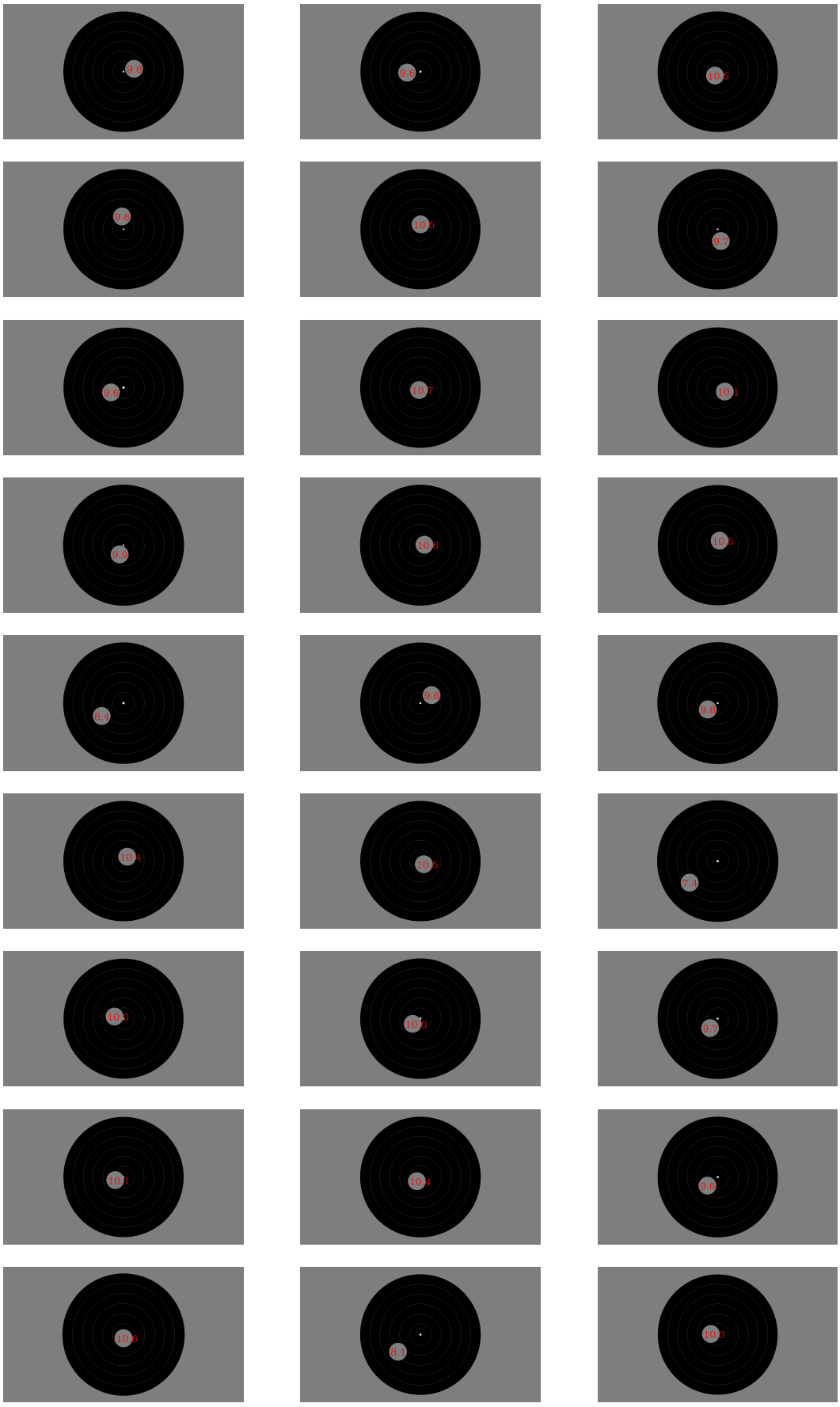


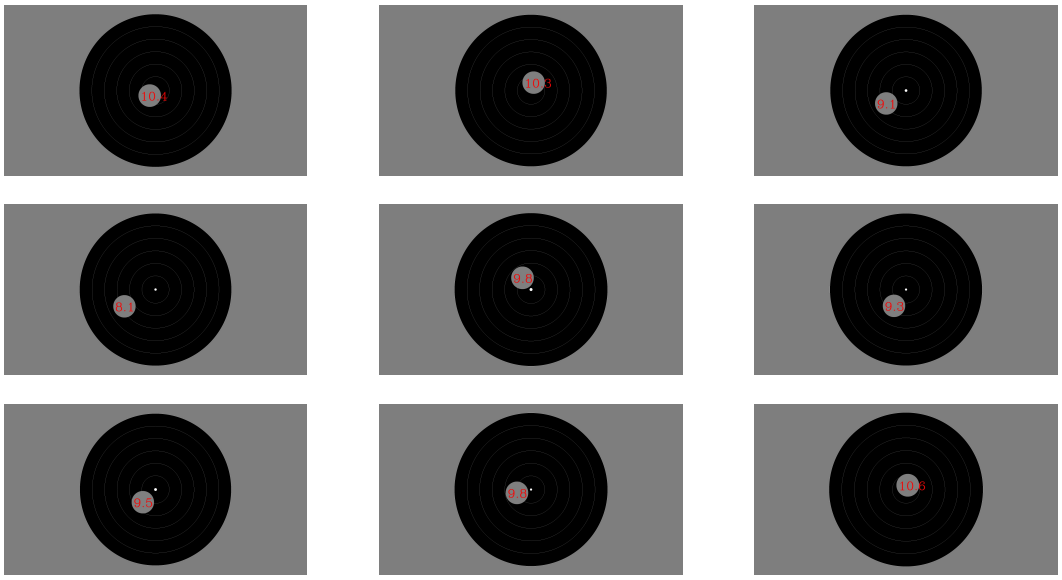

8.1
8.9
9.0
10.2
9.0
8.1
9.7
8.2
7.9
7.1
8.7
8.0
9.7
9.3
8.5
9.4
10.0
7.3
10.2
7.6
9.6
9.7
9.9
9.4
8.3
9.1
8.8



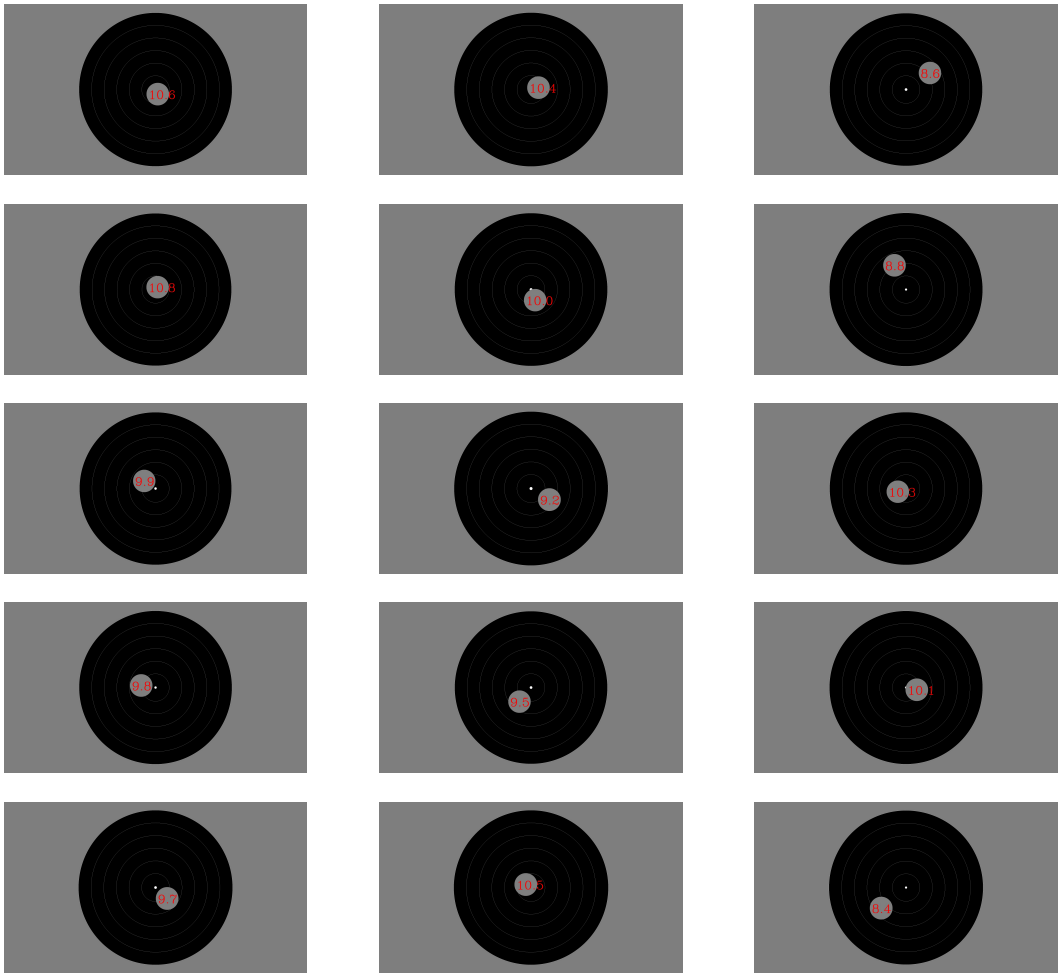
A.1.3 Frontal Targets 62061-62120

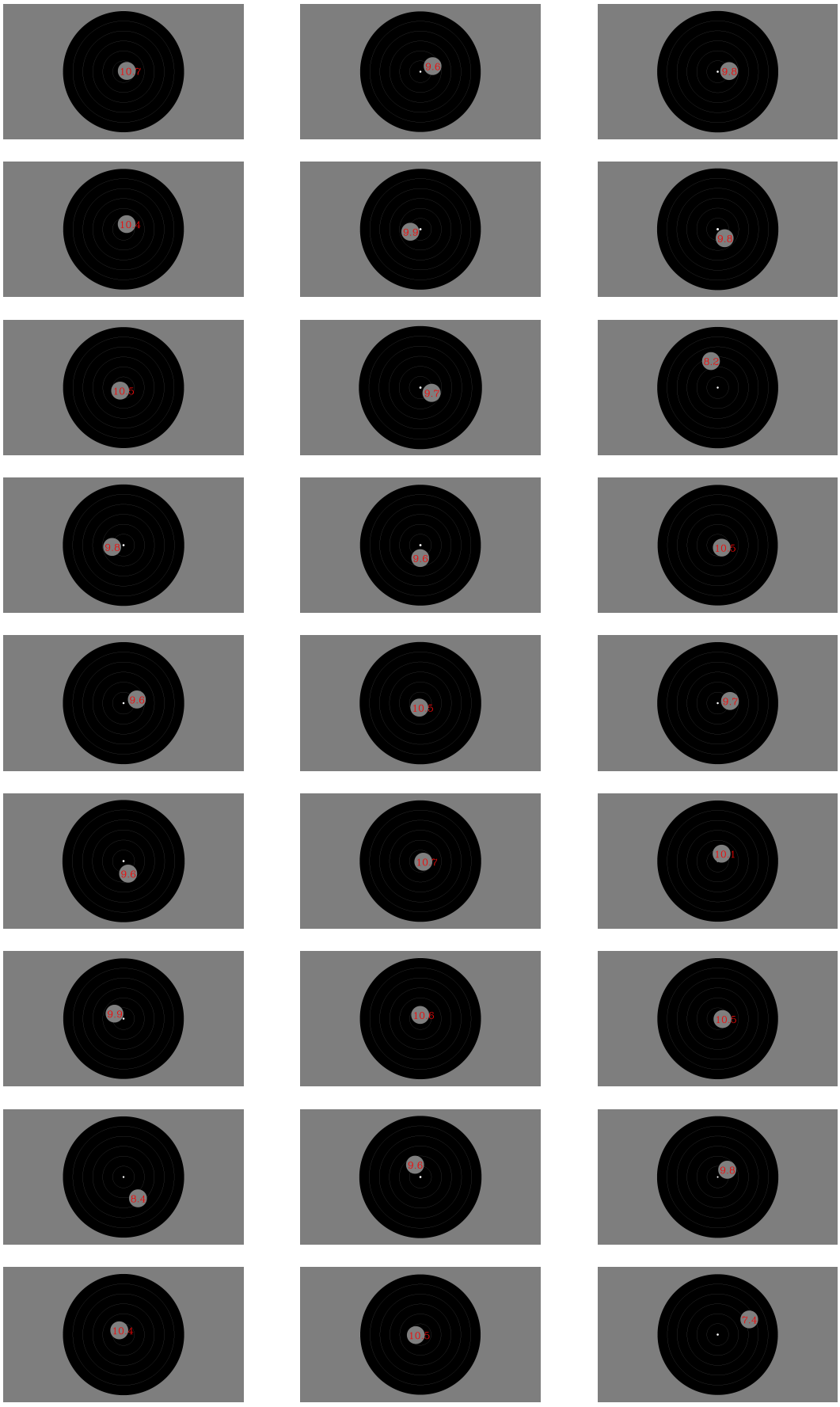


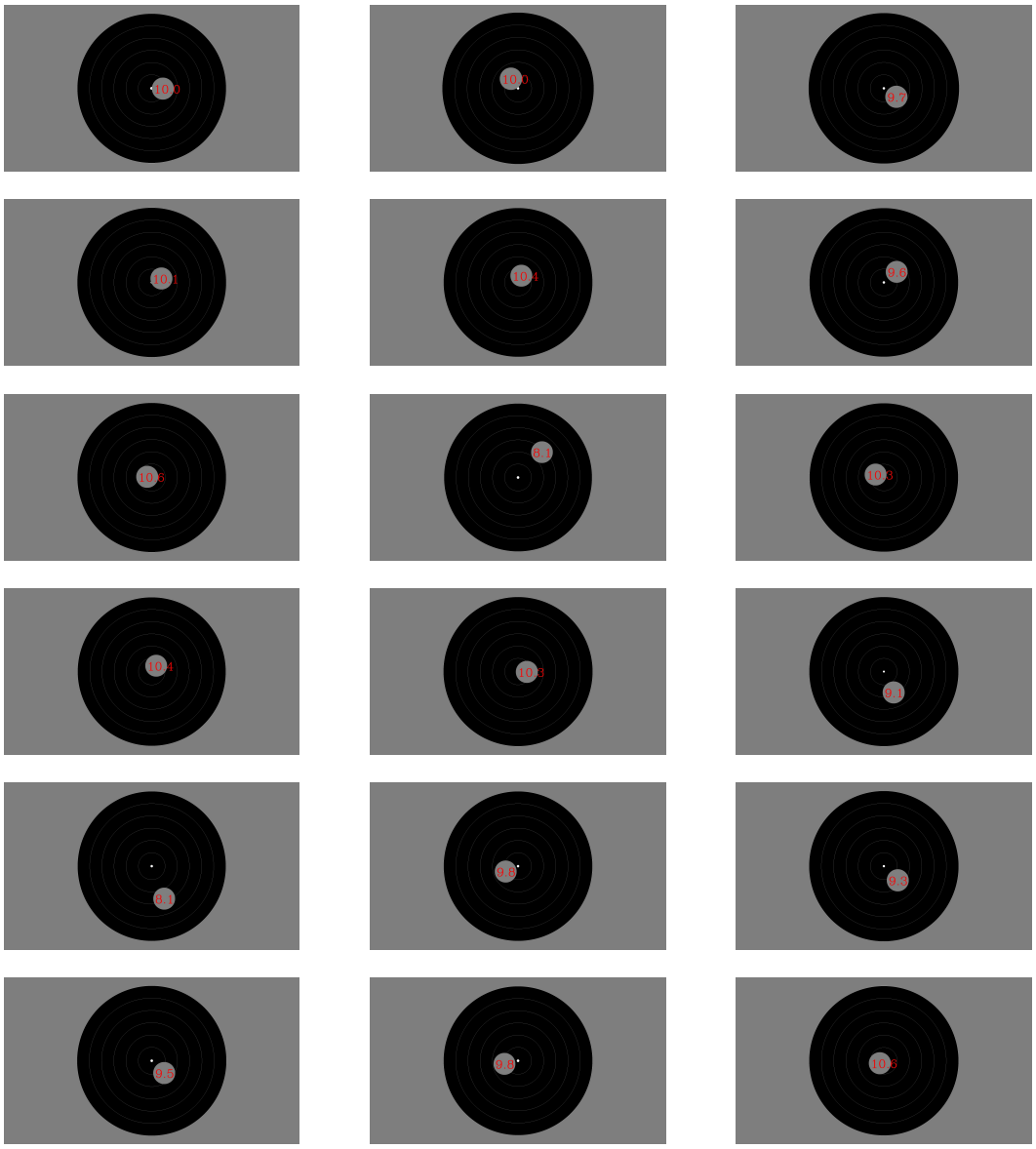




A.1.4 Angled Targets 62061-62120




10.7
9.8
9.8
10.4
9.8
9.8
10.5
9.7
9.2
9.8
9.8
10.5
9.6
10.5
9.7
9.0
10.7
10.1
9.9
10.6
10.5
8.4
9.6
9.8
10.4
10.5
7.4



A.2 Software and Libraries used

- **Visual Studio Code** : Visual Studio Code is a free, open-source code editor developed by Microsoft. It was used to implement all the proposed algorithms.

- **Overleaf** : Overleaf is an online Latex platform for writing and editing documents. It was used for every document writing task.

- **Outlook** : Outlook is a personal information manager and email application developed by Microsoft. All emails and files exchanged between student and supervisor were through this application.

- **NumPy** : NumPy is a powerful library for numerical and mathematical computations in Python.
- **OpenCV** : OpenCV is a popular open-source computer vision and image processing library.
- **os**: A module that provides a way to interact with the operating system, allowing you to perform tasks such as file and directory manipulation.
- **glob** : A module that provides functionality to match files and directories, allowing you to find files and directories whose names match a specified pattern.

References

[1] Color models and color picker - ant design. https://ant-design.antgroup.com/docs/blog/color-picker.

[2] Electronic targets—coming to a range near you? https://www.ssusa.org/articles/2017/6/2/electronic-targets-coming-to-a-range-near-you.

[3] Logitech brio webcam with 4k ultra hd video. https://www.logitech.com/en-gb/products/webcams/brio-4k-hdr-webcam.960-001106.html.

[4] Opencv: Image processing. https://docs.opencv.org/4.x/d2/d96/tutorial_py_table_of_contents_imgproc.html.

[5] Acoustics/sound speed. 2013. https://en.wikibooks.org/wiki/Acoustics/SoundSpeed.

[6] SIUS AG. Hs10 hybridscore | electronic target system. https://www.sius.com/en/product-page/hs10-hybridscore.

[7] D. Alexander. Algorithmics: Data structures and data types. http://www0.cs.ucl.ac.uk/staff/Daniel.Alexander/D0b/d0b2002.pdf.

[8] F. Ali and A. B. Mansoor. Computer vision based automatic scoring of shooting targets. *IEEE INMIC 2008: 12th IEEE International Multitopic Conference - Conference Proceedings*, pages 515–519, 12 2008.

[9] G. Anderson and D. Emeritus. Modern scoring systems modern scoring systems and deciding scoring issues.

[10] C. M. Bishop. Pattern recoginiton and machine learning. *Information Science and Statistics*, page 738, 2006.

[11] J. F. Blinn. What is a pixel? *IEEE Computer Graphics and Applications*, 25:82–87, 9 2005.

[12] B.Marr. 7 amazing examples of computer and machine vision in practice. https://www.forbes.com/sites/bernardmarr/2019/04/08/7-amazing-examples-of-computer-and-machine-vision-in-practice/#3dbb3f751018.

[13] M. Butt, N. Glas, J. Monsuur, R. Stoop, and A. Keijzer. Application of yolov8 and detectron2 for bullet hole detection and score calculation from shooting cards. *AI (Switzerland)*, 5:72–90, 3 2024.

[14] L. Carlos, G. Cerda, Y. Daniel, and O. Ceda. Design of an electronic scoring system using the hard target method for detection and location of the impacts on a shooting target.

[15] J. Chakraborty, G. Ottoy, M. Gelaude, J. P. Goemaere, and L. De Strycker. Acoustic localization of unknown sources with wireless sensor nodes. *2014 17th International Conference on Computer and Information Technology, ICCIT 2014*, pages 488–493, 3 2003.

[16] M. Y. Cheng and Ch. C. Lee. On real-time contour error estimation for contour following tasks. *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*, 2:1047–1052, 2005.

[17] P. Corke. Robotics, vision and control. 2017.

[18] K. Demaagd, A. Oliver, N. Oostendorp, and K. Scott. Practical computer vision with simplecv. July 2012. https://www.oreilly.com/library/view/practical-computer-vision/9781449337865/ch04.html.

[19] H. Farid. Digital image forensics. *Proceedings of the National Academy of Sciences*, 2001.

[20] International Shooting Federation. Issf technical rules 2023. 2023. https://www.issf-sports.org/getfile.aspx?mod=docf&pane=1&inst=458&file=Technical_Rules.pdf.

[21] G.Bueno, G. Oscar, D. Suarez, J. Luis, E. Aranda, J. S. Tercero, I. Serrano, G. Noelia, and V. Enano. Learning image processing with opencv. 2015.

[22] R. C. Gonzalez and R. E. Woods. Digital image processing (3rd edition). *Prentice-Hall, Inc. Upper Saddle River, NJ, USA ©2006*, page 976, 2007.

[23] I. Goodfellow, Y. Bengio, and A. Courville. Deep learning. page 775, 2016.

[24] J. Gregorová. Innovating sports shooting with computer vision. 2021. https://excel.fit.vutbr.cz/submissions/2021/044/44.pdf.

[25] K. Host and M. Iva Si C-Kos. An overview of human action recognition in sports based on computer vision. *Heliyon*, 2017.

[26] IBM. What is computer vision? https://www.ibm.com/topics/computer-vision.

[27] J. Illingworth and J. Kittler. The adaptive hough transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9:690–698, 1987.

[28] A. K. Jain. Fundamentals of digital image processing. *Vision*, page 569, 1989.

[29] B. Jähne. Digital image processing. *Digital Image Processing: 6th Revised and Extended Edition*, pages 1–607, 2005.

[30] A. Kaehler and G. R. Bradski. Learning opencv 3 : computer vision in c++ with the opencv library.

[31] K. Korlakunta. Gray scale and color images using opencv. https://korlakuntasaikamal10.medium.com/gray-scale-and-color-images-using-opencv-eda0c13c292a.

[32] The Great American Photo Lab. Digital explained. https://northcoastphoto.com/digital-explained/.

[33] Y. Li, C. Zhang, R. Cheng, Y. Xu, P. Li, and H. Ma. Automatic target-scoring model based on image processing. *Proceedings - 2022 9th International Conference on Digital Home, ICDH 2022*, pages 25–30, 2022.

[34] Y. C. Lin, S. G. Miaou, Y. C. Lin, and S. L. Chen. An automatic scoring system for air pistol shooting competition based on image recognition of target sheets. *2015 IEEE International Conference on Consumer Electronics - Taiwan, ICCE-TW 2015*, pages 140–141, 8 2015.

[35] M. Nakanishi and T. Ogura. Real-time line extraction using a highly parallel hough transform board. *IEEE International Conference on Image Processing*, 2:582–585, 1997.

[36] M. Nixon and A. Aguado. Feature extraction and image processing. *Feature Extraction and Image Processing for Computer Vision, Second Edition*, pages 399–434, 2012.

[37] W. K. Pratt. Digital image processing : Piks scientific inside. page 782, 2007.

[38] M. Rizon, H. Yazid, P. Saad, A. Yeon, M. Shakaff, A. R. Saad, M. Sugisaka, S. Yaacob, M. R. Mamat, and M. Karthigayan. Object detection using circular hough transform. *American Journal of Applied Sciences*, 2:1606–1609, 2005.

[39] A. Rosebrock. 4 point opencv getperspective transform example. https://pyimagesearch.com/2014/08/25/4-point-opencv-getperspective-transform-example/.

[40] J. Rudzinski and M. Luckner. Low-cost computer vision based automatic scoring of shooting targets. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7828 LNAI:185–195, 2013.

[41] C. Sanctuary, L. Conner, S. Amour, S. Diego, S. R. Hsieh, and C. Mesa. Remote strafe scoring system. 7 1987.

[42] R. D. Sharma, S. L. Agrwal, S. K. Gupta, and A. Prajapati. Optimized dynamic background subtraction technique for moving object detection and tracking. *2nd International Conference on Telecommunication and Networks, TEL-NET 2017*, 2018-January:1–3, 7 2017.

[43] A. Soetedjo, M. I. Ashari, A. Mahmudi, and Y. I. Nakhoda. Implementation of sensor on the gun system using embedded camera for shooting training. *Proceedings of 2014 2nd International Conference on Technology, Informatics, Management, Engineering and Environment, TIME-E 2014*, pages 69–74, 1 2015.

[44] E. Stenhager and N. Lavesson. Hit detection in sports pistol shooting. *33rd Workshop of the Swedish Artificial Intelligence Society, SAIS 2021*, 6 2021.

[45] G. Stockman and L. Shapiro. Computer vision. 2001.

[46] Z. Su and W. Chen. Effective target extraction of automatic target-scoring system. *Proceedings of 2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference, IAEAC 2019*, pages 1402–1406, 12 2019.

[47] S. Sural, G. Qian, and S. Pramanik. Segmentation and histogram generation using the hsv color space for image retrieval. *IEEE International Conference on Image Processing*, 2, 2002.

[48] R. Szeliski. Computer vision. 2022.

[49] G. Thomas, R. Gade, T. B. Moeslund, P. Carr, and A. Hilton. Computer vision for sports: Current applications and research topics. *Computer Vision and Image Understanding*, 159:3–18, 6 2017.

[50] Z. Wang, Y. Xu, J. Ford, F. S. Makedon, Z. Zhuang, L. Gao, and J. D. Pearlman. An adaptive approach for image subtraction. *Proceedings of the 26th Annual Conference of the IEEE Engineering in Medicine and Biology Society*, pages 1818–1820, 2004.

[51] P. D. Widayaka, H. Kusuma, and M. Attamimi. Automatic shooting scoring system based on image processing. *Journal of Physics: Conference Series*, 1201:012047, 5 2019.

[52] J. Xian, Z. Li, and S. Wu. A laser simulated shooting system based on image recognition. *10th International Conference on Computer Science and Education, ICCSE 2015*, pages 735–739, 9 2015.

[53] S. Yafei, Z. Li, and W. Guowei. The visual detection of bullet holes in a target. pages 914–917, 11 2002.

[54] M. Ye, D. Zhou, and X. Zhu. Design of ball firing system of ball bore in self cover bore based on image recognition.

[55] T. T. Zin, I. Oka, T. Sasayama, S. Ata, H. Watanabe, and H. Sasano. Image processing approach to automatic scoring system for archery targets. *Proceedings - 2013 9th International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IIH-MSP 2013*, pages 25–262, 2013.