

Innovating Sports Shooting With Computer Vision

Jana Gregorová*

Abstract

Bullseye shooting is a shooting sports discipline where the shooter is standing in one place and shooting at a static paper target. The shooter's goal is to land as many hits in the centre of the target as possible. The distance between the shooter and the target usually ranges from 10 m to 300 m. That way, while shooting, the shooter is unable to see where exactly their hits land unless some kind of auxiliary equipment is used. To calculate the score, all shooting needs to be stopped as the shooter walks right into the field to see their target up close. For a new round, the shooter either has to change the target for a new one or cover target hits up with cover up patches. The goal of this work is to find a solution to calculate shooter's score automatically. One of the possible approaches to this problem is taking a video of the target and processing it, so that new hits in the target can be detected and their score calculated. The resulting image will then be shown to the shooter on a screen near them. The result of this paper is a commercial distribution-ready solution both for individual shooters and commercial shooting ranges. The solution consists of a camera setup and an application which displays the video from camera and evaluates the shooter's score using computer vision. The solution brings significant improvements in terms of shooter's comfort and bullseye shooting training effectiveness. It is an innovative equipment not only for skilled shooters, but also for beginner shooters.

Keywords: Automatic Score Evaluation for Bullseye Shooting — Computer Vision in Sports Shooting — Background Subtraction — Template Matching — Image Stabilization — Cross-correlation of Two 2D Arrays

Supplementary Material: [Demonstration Video](#)

*, Faculty of Information Technology, Brno University of Technology

1. Introduction

Computer Vision is a key part of many sports today. It is commonly used as an auxiliary component in score evaluation and monitoring athletes' movements in order to increase training effectiveness [1]. One of the sports for which comprehensive solutions are not yet commonly available is sports shooting. This paper aims to integrate computer vision in sports shooting in order to innovate the training processes and to bring new audience into the sport.

Shooting sports is a very broad area consisting of many disciplines and many different scoring rule systems. This paper focuses on casual training of bullseye shooting [3] in preparation for not only slow precision fire but also rapid fire competitions. In these disciplines, the shooter stands still at their firing point

and shoots at a target which is firmly fastened to one spot. The target is a flat surface, usually paper or tag board and contains one or more bullseyes. A bullseye is an aiming point printed on a target. Hits in the bullseye are assigned the highest score value, typically ten points. A target usually contains other areas with different assigned score amounts. These amounts are always lower than the bullseyes'. A different number of points is assigned to the shooter depending on the shot hole location in the target. If a shot hole area either comes in contact with the outside of a scoring area or it crosses two areas, it is given the higher value. It is possible that the shooter hits the same area multiple times. According to NRA Precision Pistol Rules [3], only visible hits are scored. However, in the case of grouping three or more shots together so close that

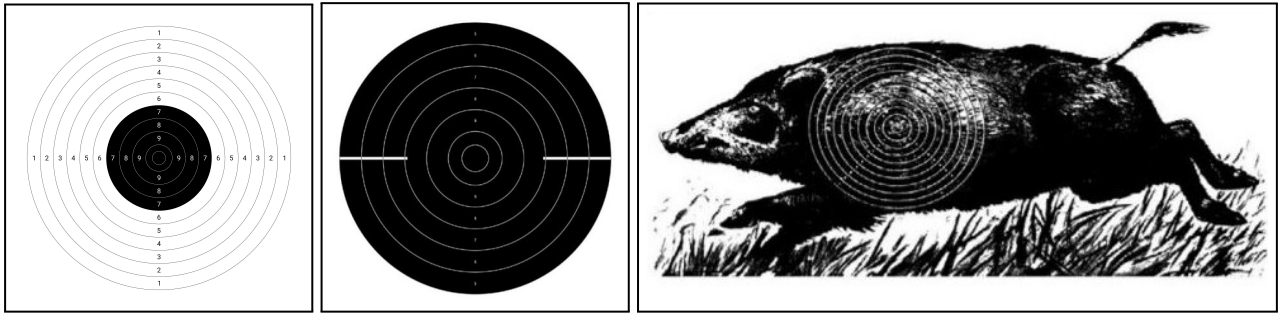


Figure 1. Different paper target types (images originated from [2]). Left to right: 25 m Precision and 50m Pistol Target, 25 m Rapid Fire Pistol Target, 50m Running Target for Paper Targets.

it is possible for another shot to go through the enlarged hole without leaving a mark, the shooter will be scored hits for the non visible hits as if they passed through this area. In such cases where this area crosses multiple scoring areas, these shots are assigned score of the highest traversed scoring area. All hits outside scored areas are scored as misses with zero points. The shooter's goal is to score as many points as possible.

During casual training, the distance between the target and the shooter can vary depending on shooter's skill, firearm used or preference. However, it is generally not shorter than 10m. In that manner, the shooter is unable to see where exactly their hits land unless using a telescope. In slow precision shooting, it is important especially for beginners (due to their generally irregular movement patterns and great variability in errors [4]) to see the place in the target they hit right after firing; this way, they could instantly see the impact their body position and breathwork has on their

shooting dexterity and could react to it right in the next shot by adjusting body position, breathwork and concentration levels.

In the course of one shooting round, varying number of shots in varying time limit can be fired. In any case, exact limits have to be determined without exception before the shooting round starts. The score is evaluated only after the shooting round is over. The person who calculates the score (during competitions, a non-competitor is selected for this position according to the hosting organization's rules; during training, it is usually a coach or the shooter himself who calculates score) has to take the necessary safety measures (all shooting in the area has to be stopped) as they walk into the shooting field to take a closer look at the target card they are evaluating. The score for each shot is evaluated manually (a scoring gauge can be used for doubtful hits [5]) and marked on the target. During competitions, these results are recorded in designated score cards as well.

When the round is over, in competitions, the fired targets are exchanged for new ones and the statistical office retains them until the expiration of the time allowed for challenges and protests. When training, the shooter is free to keep or discard the target, or to cover their target hits with cover up patches in order to reuse the same target.

The solution presented in this paper works by taking a video of the target at which the shooter is shooting at, transferring the stream into a computer application which shows the video of said target to the shooter and calculates the shooter's score at the same time. That way, the shooter is able to see their target and score in the course of a shooting round.



Figure 2. In the course of shooting, the shooter is hardly aware of their actual score and towards which directions they tend to pull the gun to. If they knew while shooting, they could try adjusting their position, breathing and aim accordingly.

2. Existing solutions

There are several solutions dedicated to shooting score evaluation available on the market. I analyzed these and grouped them depending on the technology they use to operate:



Figure 3. Left: A close (doubtful) hit situation. The upper left shot would be scored as a cut 8. Right: Grouped shots. It is impossible to tell how many shots hit the bullseye, provided that no other information is given.

1. Mobile Phone Solutions,
2. Solutions Using Designated Scanners,
3. Electronic Scoring Targets With Acoustic Location of New Target Hits,
4. Electronic Scoring Targets With Light Triangulation,
5. Electronic Scoring Targets Using Piezoelectric Sensors to Locate New Target Hits,
6. Open Source Solutions,
7. Other Solutions.

Mobile phone solutions are the most affordable and portable way to calculate and store shooting results. Existing applications (e.g. TargetScan ISSF Pistol & Rifle [6, 7], Target Scanner for Competition Shooters [8]) use computer vision to calculate score based on a photograph of a used target and save this score so that it can be viewed or compared at a later time. Prices range from \$8.49 (free for applications to only store results without evaluating these using photographs) to \$24.99. None of the existing solutions is able to calculate score during shooting and none of these solutions is hitherto used in shooting competitions.

Designated scanners (e.g. DISAG RM-IV [9], Orion scoring systems [10]) operate by scanning used shooting targets and evaluating score using a computer or other processing device. This way, shooting score is evaluated only after a shooting round is finished. Therefore, the shooter is not able to see their target up close in the course of shooting. Due to the generally large size of shooting targets for firearms (60×60 cm as opposed to 15×15 cm which is typical for air guns), these solutions are designed to work exclusively with air guns. Their price ranges from approximately \$290 to \$4,000 with some solutions

requiring a regular (monthly/yearly) maintenance fee to operate. The DISAG RM-IV solution has been certified by ISSF [11] as a valid tool for score evaluation in shooting competitions.

Electronic scoring targets with acoustic location of new target hits are distinctive targets with 3 or 4 built-in acoustic scanners. The location of each target hit is calculated using the data from these scanners by a PC (see Fig. 4 bottom). While shooting, the shooter is able to see their resulting score as well as the location of their target hits on a PC or a specific shooting monitor. Some of these electronic scoring targets (e.g. Megalink 4K187 [14], SIUS S310 [15]) have been certified by ISSF [11] for use in ISSF shooting competitions. All of these solutions are fit for air guns and are not suitable for firearms due to the large size or firearm shooting targets and due to firearm destructiveness. The price range is approximately between \$760 and \$6,000.

Electronic scoring targets with light triangulation use either lasers (SIUS LS25/50 [16]) or a high-speed infrared camera (DISAG OpticsScore [17]) to calculate shooter's score. The results are displayed on a monitor in the course of the shooting round. These solutions are designed for air guns only. Both examples are ISSF-certified [11] for use in shooting competitions. The cost of these systems is not provided publicly.

Electronic scoring targets using piezoelectric triangulation by Sport Quantum use two plates and four sensors to find position of the impact [18]. The impact plate has to be changed after some time because it gets slightly damaged with each impact. The image on the target can be changed interactively using an Android application. In this application, the shooter is able to



Figure 4. Example existing solutions. Top left: TargetScan ISSF Pistol & rifle (image originated from [6]). Top right: DISAG RM-IV designated target scanner (image originated from [12]). Bottom: PC-based system for personal training with the 4K187 electronic scoring target (displayed on the left of the image) by Megalink (image originated from [13]).

see their score, target hits, and also the location of their average hit. The solution for firearms is in development and not available as of today. According to Sport Quantum's website, it is going to function with firearms using any caliber and rifles up to 1000 joules for distances from 25 m to 50 m [19]. The solution for air guns is available for roughly \$2,600 [20] plus taxes.

Open source solutions are attempts to create affordable versions of Electronic Scoring Target. These vary in technology used. The FreETarget [21] uses acoustic triangulation to locate target hits. Electronic Target [22], for example, uses impact shockwave triangulation to locate target hits and then displays these hits in a user application. There is also a solution¹

using a web camera and processing the image using emguCV library. After that, a graphical representation of the result is displayed in a user application. The shooter is able to change the target from a distance by making the target device move the paper. All of these open source solutions are for air guns only.

Other solutions related to shooting sports – these solutions do not calculate shooter's score, however, I have researched these because they cover some of the issues which I have been dealing with in my own solution design. PitShooting [23] is a system which shows target during shooting using a camera attached above target stand and provides a solution for walking into the shooting field – targets are attached to a construction which moves the target to shooter's desired shooting distance using a provided tablet. The shooter is also able to move the target to the firing

¹<https://github.com/BlasterBB/BidaSius>

line in order to see the target up close and accurately calculate shooting score or to change targets. The solution is built-in in a shooting range. Other related solutions are shooting match managers(e.g. ABVisie Match Manager [24] and Rifle Target: Rifle Shooting Database [25]). There are many other solutions for mobile phone and computers which help the user to organize matches and register shooting results.

3. User requirements

Although bullseye shooting disciplines have worked without computer aid satisfactorily for many years, some areas could be made more effective, comfortable or appealing with little use of modern technology. In order to design and implement a valid solution, specific areas to be innovated have to be precisely determined first. So, what are these areas that the shooting training loses its efficiency in the most and could be changed with least effort?

The answers will probably vary person to person depending on the shooting skill, training approach and knowledge of technology. That is why I, as a future user of this particular system, decided to answer myself in the first place and run a survey to validate these answers subsequently.

As an intermediate shooter, seeing where my hits land during slow precision fire is very important. Usually I would have to identify my wrong movement tendencies after a shooting round is over. However, if I had an immediate feedback after a shot, I could determine certain movement tendencies and adjust my position right after seeing where the shot went through the target.

When it comes to rapid fire, it is not that important to see where the hit landed as there is very little time in preparation for the next shot. In these cases, I would greatly appreciate having a device to measure my shooting time, show it, and/or count only those shots that landed in the target during specific time frame.

As for score calculation, a simple application to do that for me could bring significant improvements in time efficiency. It would remove the necessity to wait for all shooters on the firing line to be done with their shooting rounds and the necessity to walk all the way to the target and to calculate the score. Reducing the amount of times a person has to cross the firing line would bring great improvements in safety and it would further improve time efficiency.

Furthermore, a system which could remember which hits belong to which round and highlight them in different colors would make changing the target or covering

past hits between each round obsolete. Again, this saves some extra time.

Finally, I would welcome an easy way to save my shooting statistics to see how I progress over time and to analyze my past statistics in order to be able to determine the factors which affect my performance, and, if such effects are desirable or not. Currently I never record my statistics due to high amount of work which is needed to do that.

The finalized solution should, in addition, be safe, practical, comfortable to use, accessible and easy to work with for shooters of all ages and skill levels.

4. Key Points to the Solution

In designing of the solution I took roles both as the user and as the developer. I specified the user requirements (as seen in Section 3) as the user and later planned my approach to assure these requirements are met.

First I had to record videos of targets being shot at in order to obtain development data. I used different methods which I was considering for the physical solution in order to see which work and which do not and to specify camera requirements. The outcome of this process is a physical solution design described more in detail in Section 5.

Secondly, I had to determine the process of score evaluation itself. This would be done prior to designing the user interface, as I had to confirm that the physical solution is valid by creating an application to evaluate shooting score using a pre-recorded video.

I designed a user interface and run a survey prior to implementing it in order to validate the whole solution design I have created thus far. The survey is still running, but I have made a provisional evaluation in order to see in which I will have to amend my existing design.

Finally, I will distribute the solution to the subjects which have signed up for testing in the survey I am currently running. I will collect user feedback and use it to further improve the solution.

5. Physical Solution Design

The physical solution consists of a camera and a computer. To define camera requirements, several facts need to be considered. First and foremost, the camera has to be placed in such a way that the shooter won't be able to hit it. In that manner, the camera could be placed on the firing line (next to or above the shooter). If this is the case, then a relatively powerful camera with lens is required in order to acquire data in sufficient quality. The advantage of this is that the image is not curved in any way, providing accurate results.

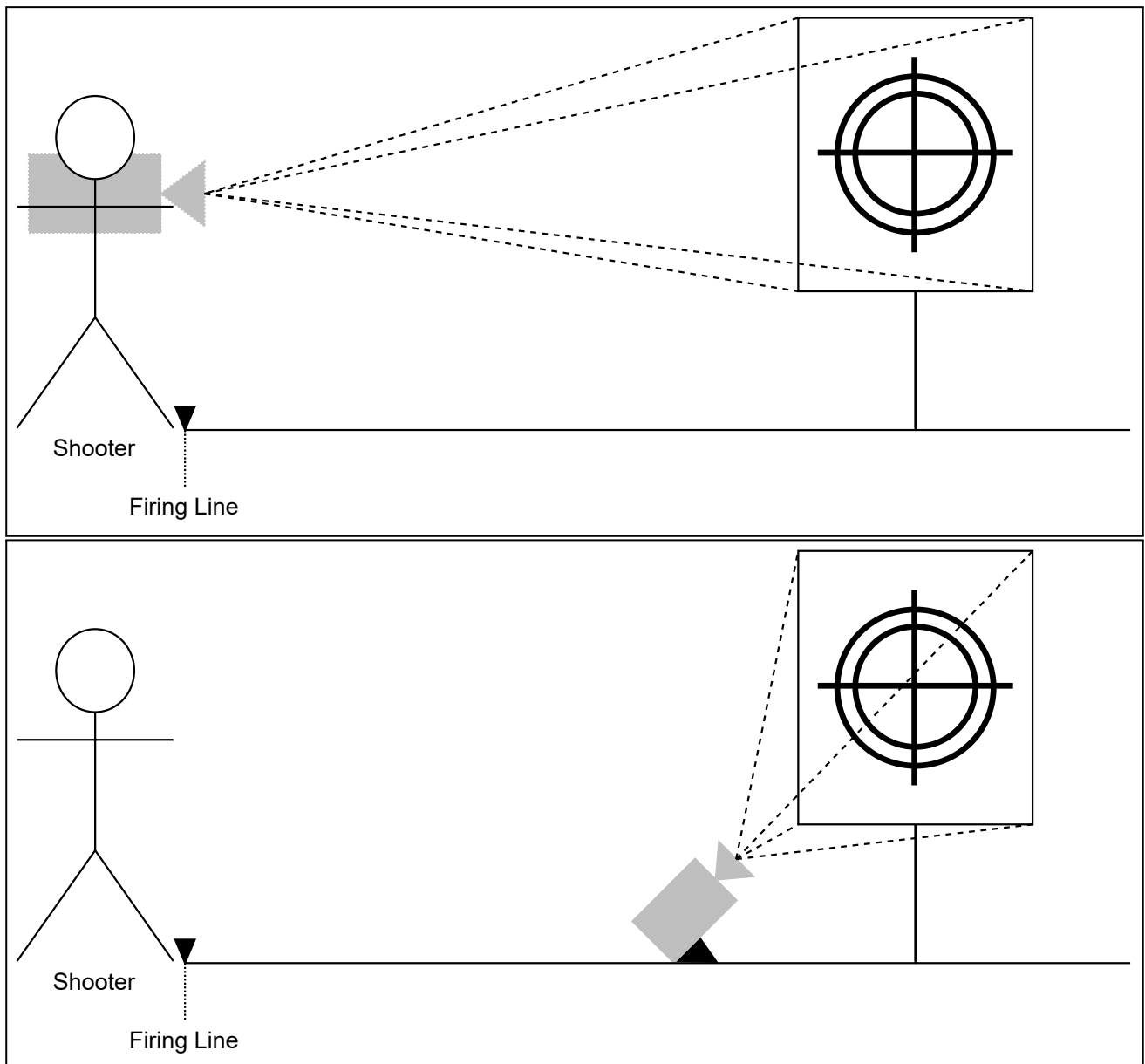


Figure 5. Camera placement options. Top: a camera with relatively powerful lens is placed on or right behind the firing line. Bottom: an inexpensive camera is placed near the target in the shooting field.

This solution is not very comfortable for individuals, as it requires the shooter taking additional equipment to the shooting range.

Another possible solution is using an IP camera placed right below (the shooter is least likely to hit below the target, most accidental shots hit above or sideways) the target. In this case, a common camera with no add-on lens will suffice. The camera needs to be an inexpensive one, because there is a possibility (although not very probable) that the shooter will accidentally hit the camera. The data from the camera could be sent to a processing device over WiFi. Due to the camera placement, the target is viewed at an angle which should be taken into account in processing of the camera output, as it could provide inaccurate results otherwise. The solution offers way more variability as

opposed to the previous one and is suited for individual shooters to take and use wherever they would like to as well as shooting ranges as a permanently integrated system.

At the opposite point, the user is going to be communicating with the system through a user interface, running on a PC a smartphone, or a tablet.

The target stand will remain unchanged.

6. Target Hit Detection and Score Calculation

Prior to the software implementation, I obtained a dataset using both methods described in the previous section. The dataset contains video recordings of targets being shot at from three different shooting ranges



Figure 6. Sample images taken by cameras at different placements. Left: image taken by a camera from the shooter's position. Right: image taken by a camera near the target in the shooting field. Note the placement's impact on deformation of target scoring areas.

and varying weather conditions. Two of said shooting ranges are outdoor ranges.

To detect target hits, simple background subtraction can be used. In theory, since the image is static, nothing is going to appear in the foreground unless the target is hit. In that moment, an object will appear in the foreground mask where the bullet made a new hole in the target. That way, new shapes in the foreground mask could be detected and then compared to their location in the original target to find the score.

I made tests using both Gaussian mixture model [26] (implemented as MOG2 Background Subtractor² in OpenCV [27]) and non-parametric model [28]. The non-parametric model showed better results for target hit identification (see Fig. 7), therefore it is used in the final program (implemented as K-nearest neighbours background subtraction³ algorithm in the OpenCV library). In order to obtain optimal results, the KNN Background Subtractor was tested on the full dataset with varying parameters.

However, background subtraction in and of itself proved insufficient. At this point, the program still detects noise in the background as the bullets are stopped in the backstop of the range.

I tried to address this by using image alignment [29]. The features in the template are going to be matched with respective features in each frame; this way, each frame is aligned with the target template before being passed on to the background subtractor. This method

works as needed for individual frames, for videos it is too computationally expensive and the results are distorted (see Fig. 8). Therefore, I have renounced from using this method in this project.

Instead, I made the program automatically locate the target and crop the video's background out. The program uses template matching⁴ to locate the target in the frame. A reference image of a target is used as the template. In order to make template matching scale invariant, the program loops over the template matching function multiple times and progressively makes the template image smaller. Since the template image is bigger than all the actual targets in the dataset, there is no need to execute previously mentioned loop with the template image scaled bigger than its original. To identify a successful match, I experimentally found a threshold value. Finding the closest match possible is not useful in this case, as it is possible that the target is not in the camera frame at all. Once the match value is greater than said threshold value, the program saves the rectangle with the supposed target location in the source image and then uses this rectangle to crop all frames the program later works with. That way, the target template matching loop is executed only once.

This approach has helped, nonetheless, the program was still detecting too many false positives. On that account, I set restrictions on which of all the detected areas are going to be detected as target hits. A hit stays in the image (and in foreground mask) for several frames. Therefore, comparing a detected ob-

²[cv::BackgroundSubtractorMOG2 Class Reference](#)

³[cv::BackgroundSubtractorKNN Class Reference](#)

⁴[Image Processing: Object Detection](#)

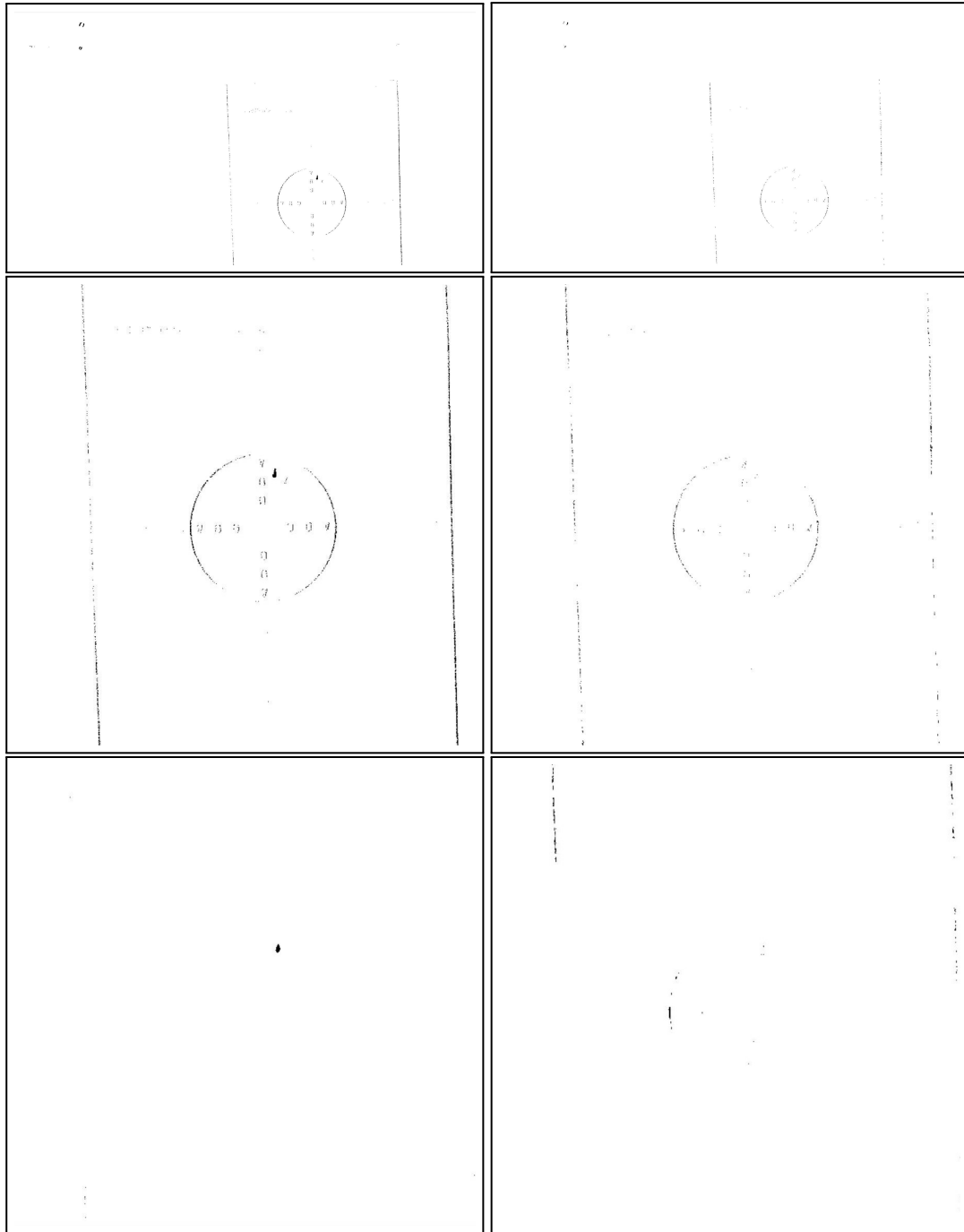


Figure 7. Foreground detection (left to right, top to bottom; each image is a side-to-side comparison of KNN (K-Nearest Neighbours Background Subtraction Algorithm) and MOG2 (Gaussian Mixture-based Background/Foreground Segmentation Algorithm)): without any image preprocessing, after cropping the image, after implementing image stabilization using cross correlation.

ject in the foreground mask with the same area in the previous frame's foreground mask and marking all the objects that were not present in the last frame irrelevant takes out objects that either appear in the video for a very short time or move very fast (such as flying bullet cases). Since both areas are binary images, each pixel is compared to the respective pixel in the same area of the past frame's mask. If the pixels do not match, the number of differences is incremented. I

experimentally evaluated the highest possible number of differences between two areas as mentioned above as 10. The disadvantage of this approach is that the hit is detected with a one frame delay. If the exact time of the hit needs to be recorded and shown to the user in the statistics, it could be done by recording the time of the past frame with each hit. However, the shooter will always see the hit being detected with one frame delay during shooting. Testing proved that the

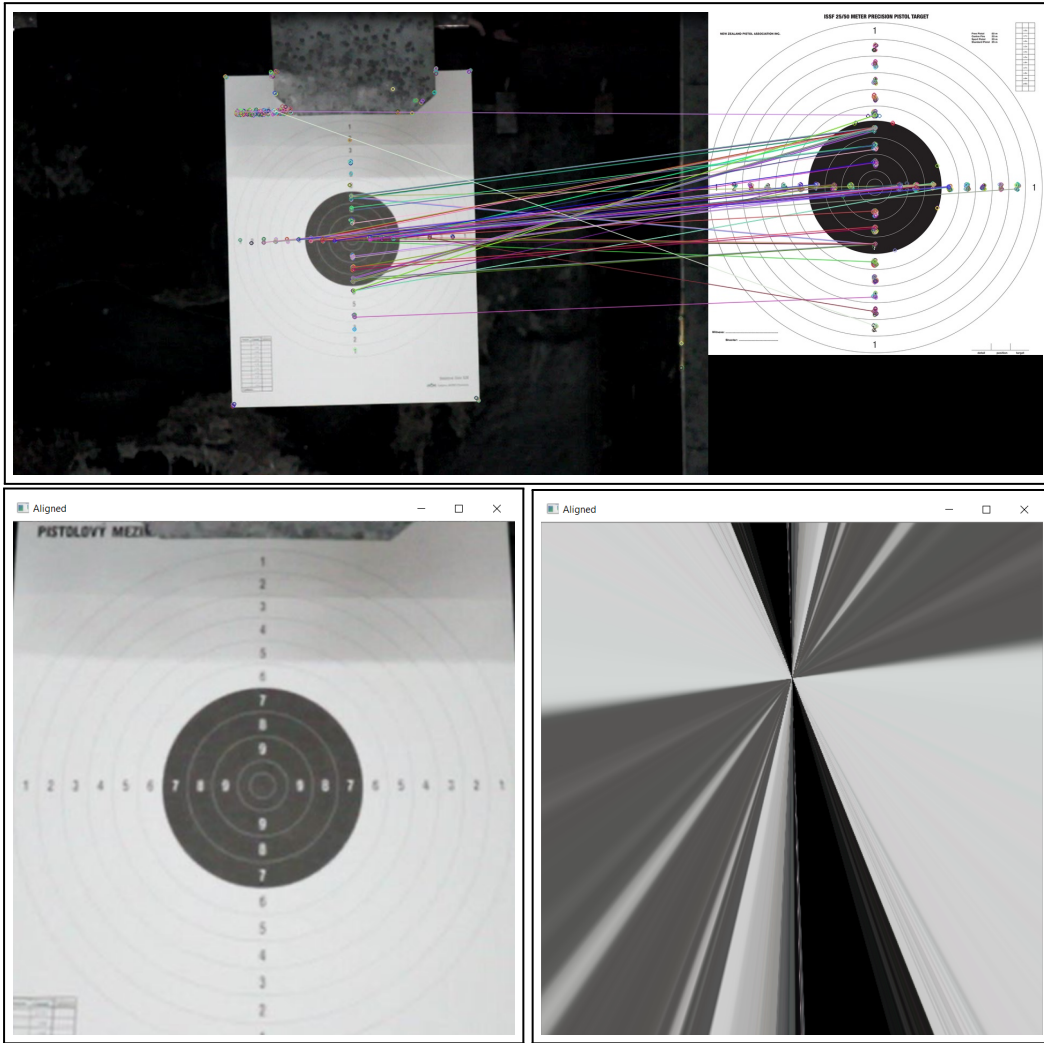


Figure 8. Top: image alignment – feature matches. Bottom, left to right: image alignment in a single picture, failed image alignment in a video.

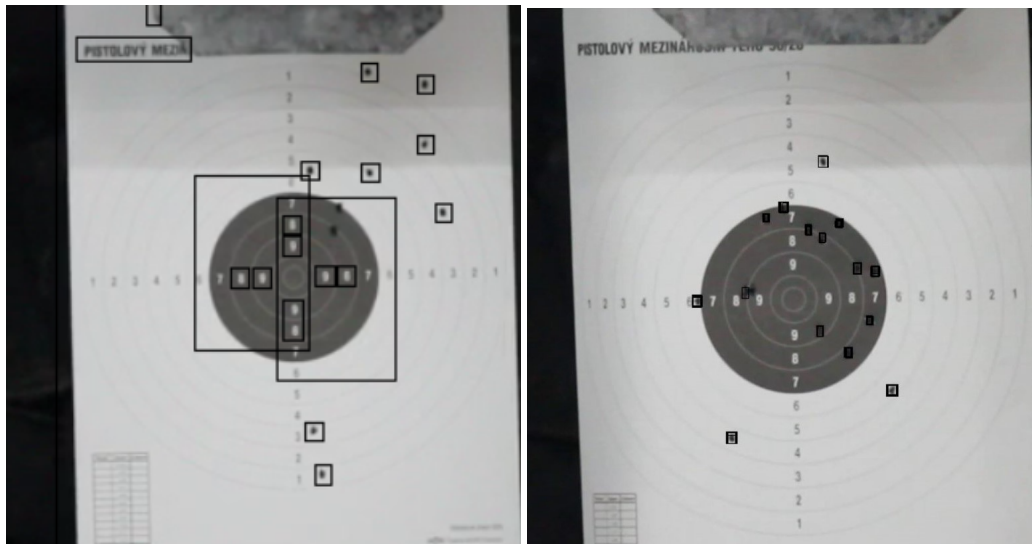


Figure 9. Left to right: Target hit detection without hit size restraints, target hit detection with hit size restraints and image stabilization.

advantages outweigh the disadvantages and much less false positives are being detected (see table 1 for exact numbers).

Furthermore, each target hit occupies a certain area of relatively the same size. I estimated that a hit could take up roughly between $\frac{1}{3000}$ and $\frac{1}{25000}$ of the total

Table 1. Comparison of Methods Used for Target Hit Detection. Each method is a combination of the listed method and all other methods above the method. Table data were retrieved by testing listed methods on the same sample video. Actual target hit count is the human-read amount of target hits in the input video. It deviates for some methods because in some cases the hits were cropped out of the image by the program itself.

Method Used	Actual Target Hit Count	Detected Hit Count	False Positives [%]	Undetected Target Hits [%]
Background Subtractor KNN (max. 1 hit per frame)	18	1 879	99.042	0
Restricted Hit Area	18	1 611	98.883	0
Locating the Target in the Frame and Cropping its Background Out	15	1099	98.635	0
Detected Object Must be Present in the Same Area in the Previous Frame	18	1 261	98.573	0
Image Stabilization	15	868	98.272	0
Intersection over Union	15	37	59.459	0
Detected Object Area Ratio Restrictions (x:y > 0.5)	17	18	5.556	0
Amount of Detected Pixels in Object Area > $\frac{1}{2}$ of Total Object Area	17	17	0	0
Using Gaussian Mixture Model in Place of the Parametric Model for Background Subtraction (Parameters Set to Provide Best Results); not used in the resulting solution	15	30	53.333	3.333

frame area after it is cropped to fit the target. Discarding all objects that won't fit into this gap successfully prevents small noise and target borders from being detected as target hits .

At this point, the program still falsely detects num-

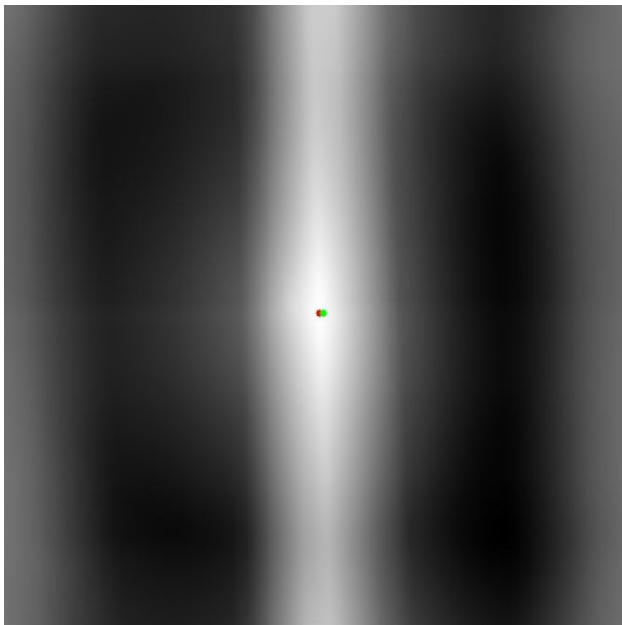


Figure 10. Cross correlation shift: the positions of the red and the green dot indicate the direction and the distance in which the target moved compared to the first frame processed.

bers as target hits. Moreover, the target slightly shakes after each target hit and that creates additional noise and even more false positives. The answer to that problem is image stabilization. I applied cross correlation to find the shift between the first processed frame of the video and each subsequent frame. OpenCV does not offer a ready-to-use cross correlating function for two 2D arrays, so I implemented my own one. The function flips one of the two images horizontally and vertically, converts both images into grayscale, blurs them using Gaussian Blur⁵, subtracts mean values from each of said images and calculates a DFT⁶-based convolution of these images. The result is then normalized and searched for its brightest spot. The difference between this spot and the brightest spot found when cross-correlating the reference frame with itself equals to the distance in pixels which the target traversed in the frame. Using the parameters of the shift (direction and distance), the current frame is then aligned with the first frame. This makes the hit detection sufficiently exact. However, the best results are obtained when the target is firmly attached both at its top and its bottom. Image stabilization helps if the whole target shakes, but in the case when the target is only attached at its top (in that manner, the bottom of the target could be

⁵Gaussian blur in the OpenCV library

⁶Discrete Fourier Transform of two 2D arrays in OpenCV

deformed in some way or move differently than the top of the target), the target hit detection could possibly become less exact.

As the reader can see in Fig. 9, the program detects each target hit several times. I implemented the Intersection over Union [30] evaluation metric to prevent the program from detecting the same hit more than once. If the ratio between the area of overlap of detected areas and the area of union of said areas is less than a threshold value I found through a series of experiments, then the detected object is evaluated as a new hit. Said threshold value is relatively small, therefore the program often detects grouped hits as only one target hit. This should not pose a problem for shooters of beginner through intermediate levels, as shooters at these levels are generally not able to intentionally hit the exact same spot multiple times. Expert shooters could be significantly disadvantaged, since most of their shots land right in the bull's eye. This could be solved at a later time through different evaluation techniques for expert shooters than for beginners and intermediates.

Prior to assessing respective score to a newly detected hit, the location of areas with different score values has to be determined. This is done only once for the whole shooting round when the first frame is processed, as the target stays in one place and the score areas do not move significantly. The frame is blurred using Gaussian Blur and converted to binary image through thresholding. The threshold is found using Otsu's method [31]. I used contour approximation⁷ to locate the prominent black circle of the target. This circle's radius is then divided by four to find the radius of the bull's eye. Each subsequent circle is then found by adding the bullseye's radius to the previous circle's radius. These areas are sufficiently exact when the target is viewed up straight by the camera. The more angled the camera view is, the less exact the scoring area detection is.

Should the camera be placed near the target such as in Fig. 5 bottom, detected score areas are not exact enough for a satisfactory score evaluation (see Fig. 12 middle – this would have been the final detected score areas as described in the previous paragraph). Instead of a circle, an ellipse is fitted on the located contour using OpenCV function `FitEllipse()`. Another ellipse is found by applying Canny edge detection, contour hierarchy and flood fill on the image of a target. Four points of each ellipse (spaced by 90 degrees) are used to find homography matrix. This matrix is

subsequently used to warp the score circles which are drawn as described in the previous paragraph. The result improves the score area location only slightly for when the camera is viewing the target up straight, but the improvement is significant when the target is viewed at an angle.

Finally, score of each individual hit is evaluated. Due to sports shooting rules, the whole area of the individual hit needs to be considered. Right after the particular hit is assessed as a new hit, its area is cropped out of the foreground mask and inserted into an empty mask image to isolate the hit from the foreground mask of the whole target which is likely to contain multiple other objects. The new mask is then compared (by applying bitwise `&`) with each scoring area's mask separately starting from the areas with the highest score. If any non-zero pixels are found in the result of the operation, then the respective score of the scoring area is assessed to the hit and the hit evaluation is finished.

7. User Interface Design

The user interface should be made as simple as possible both in appearance and controls, so that the user is not distracted from shooting. During shooting, besides their target with highlighted target hits, the user should be able to see only brief statistics (such as target hits, score count and time), just to have an overview about how they are doing. The detailed statistics can be viewed after the shooting round is over.

In order to create a quality user interface, the processes which the user is able to do have to be strictly defined first. I defined a use case for this reason.

The primary goal of this process is to receive score evaluation. The shooter starts by setting up their target and scoring system (unless it is provided by the shooting range). In the application, they can optionally select their shooting specifications (target distance, magazine capacity, time limit etc.) for a more exact score evaluation. Then they start the score calculation by clicking a start button and they can start shooting. After they're finished with the round, they stop the score calculation and view their statistics. From there, the shooter is able to repeat the whole act again.

The user interface is going to be divided in three sections:

1. Setup – the initial setup window which is going to show camera output and detected score areas,
2. Shooting – camera output with highlighted hits, score count and time,
3. Detailed Statistics – this window is going to display an overview of more detailed statistics of the finished shooting round.

⁷Implemented as the `ApproxPollyDP()` function in the OpenCV library

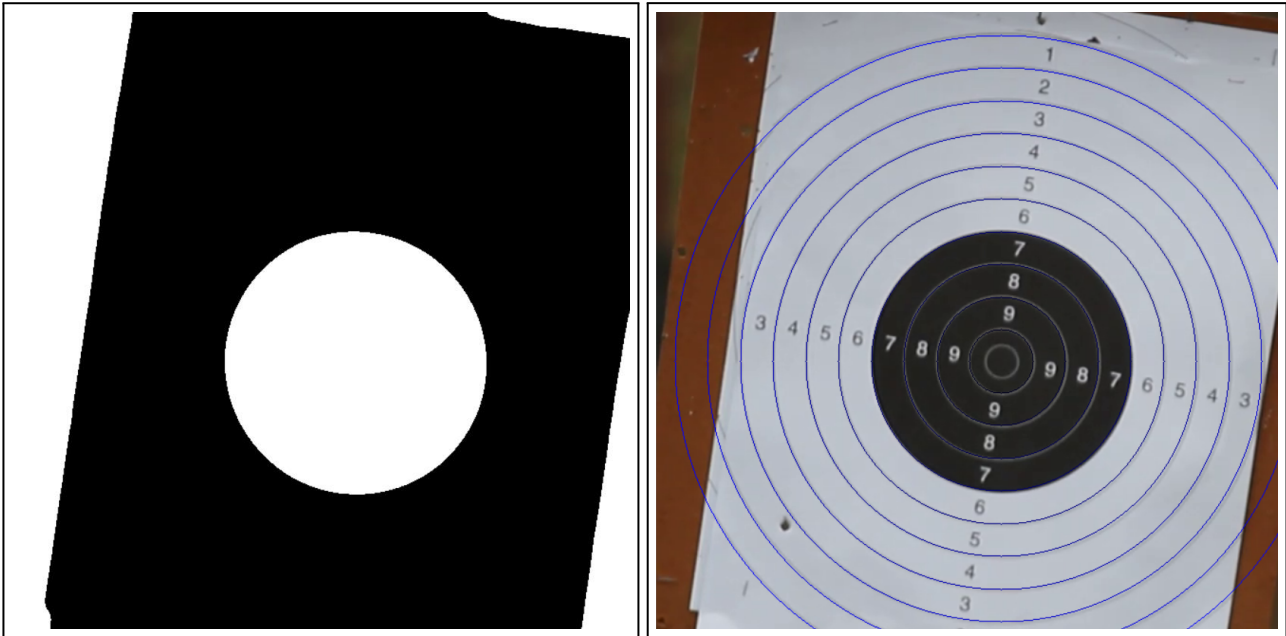


Figure 11. Locating score areas: binary image of the target which is used to find different score areas (left), detected score area borders (right).

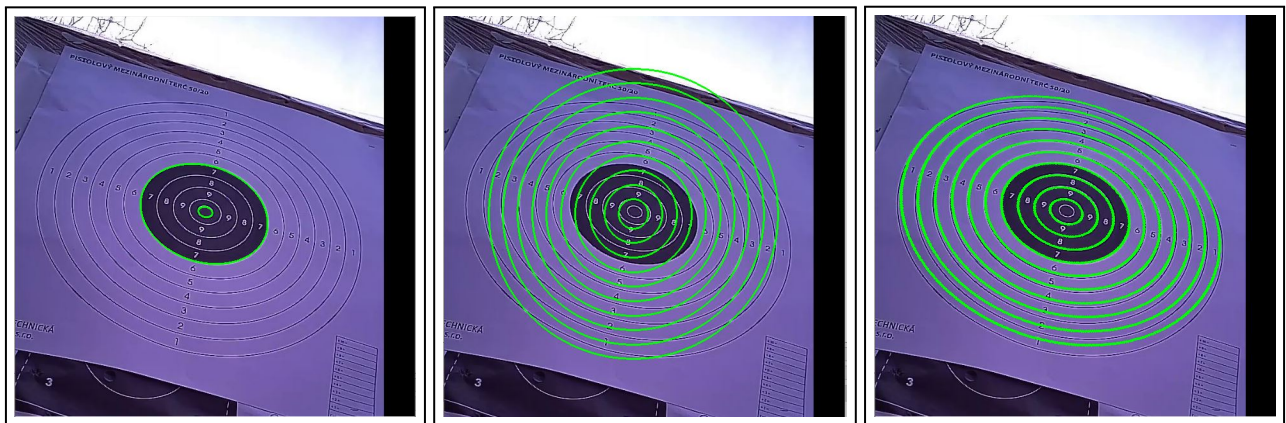


Figure 12. A more exact score area location. Left: one ellipse is fitted on the smallest score area border, another one is fitted in the borders of the black scoring area. Middle: Estimated scoring circles. Right: warped circles using homography.

8. Possible Distribution of the Solution

As previously mentioned, the system was designed to be suitable for individuals as well as shooting ranges. The application stays the same in both cases, and the physical solution depends entirely on either the user's or the shooting range's preferences.

A survey I run showed that 55.6% participants would consider the fact that the score evaluating system is integrated in a commercial shooting range as a positive aspect which might make them more likely to visit that particular range. 50% potential users would be interested in buying the system as individuals and 33.3% potential users would prefer the system built-in in a shooting range.

The system could be offered as proprietary or license. Considering that there would probably be a

relatively large amount of maintenance work needed to keep the system running, a license system is probably more suitable in this case. This also potentially makes the system more available to a larger amount of users, which is a desired aspect.

9. Conclusions

This paper presents a system which calculates shooting score from a video of the target in real time. This system helps primarily by increasing effectiveness of bullseye shooting training. As opposed to electronic shooting targets which are used in some ISSF competitions, this could potentially be used by a wider public and shooting hobbyists more than in competitions due to portability of the solution.

In a presently run survey, I validated my own re-



Figure 13. Successful score evaluation.

quirements on the system as a user. 18 other subjects of varying shooting skill and training approach have taken part in the survey so far. Most are interested in either using this system as individuals or built-in in a shooting range, with only 16.7% not interested in using this system at all. It is important to see where the shooter hit the target right after firing a shot for 66.7% participants and 66.7% thinks it is important to save results so that they can be viewed at a later time in order to see progress. None of the people asked currently use a similar system, because it is unavailable or they don't know about any such system. The survey results in confirmation of the design and purpose of the solution presented in this paper.

The system is currently still in development, however, it should be ready for user testing in the upcoming weeks. I already found 9 subjects ready for testing.

I plan to make a user application for mobile phones next in order to make the solution more convenient for individuals. As for functionality, I would like to implement score evaluation for other paper target options (silhouette, targets with multiple bullseyes, rapid fire targets etc.) as well as reactive targets and steel targets. Additionally, I would like to implement a unified scoring system between different bullseye shooting disciplines as an experiment. Furthermore, 66.6% survey participants stated that they would like to use the system to view their progress over time and past statistics, so I'd also like to implement that. Shooting ranges could provide scoring leader boards. Also, the system could possibly be able to analyze shooter's movement tendencies and provide with some training advice or suggested drills.

Acknowledgements

I would like to thank my supervisor prof. Ing. Adam Herout Ph.D. for his help.

References

- [1] Graham Thomas, Rikke Gade, Thomas B. Moeslund, Peter Carr, and Adrian Hilton. Computer vision for sports: Current applications and research topics. *Computer Vision and Image Understanding*, 159:3–18, 2017. Computer Vision in Sports.
- [2] International Shooting Sport Federation. 6. general technical rules, Feb 2020. <https://www.issf-sports.org/getfile.aspx?mod=docf&pane=1&inst=458&file=1.%20ISSF%20General%20Technical%20Rules.pdf>.
- [3] NATIONAL RIFLE ASSOCIATION OF AMERICA. Nra precision pistol rules, May 2018. <https://competitions.nra.org/documents/pdf/compete/RuleBooks/Pistol/pistol-book.pdf>.
- [4] Claudio Robazza. Issf coach (second level) sport science, Nov 2020. https://www.issf-sports.org/getfile.aspx?mod=docf&pane=1&inst=497&file=1.Sport_Science_material.pdf.
- [5] NATIONAL RIFLE ASSOCIATION OF AMERICA. Nra tournament operations guide: The ultimate guide to conducting nra sanctioned matches, Dec 2020. https://issuu.com/compshoot/docs/tourn_ops_guide/32.
- [6] Deep Scoring Ltd. Targetscan issf pistol & rifle (beta). <https://play.google.com/store/apps/details?id=com.gabrowski.targetscan&hl=en&gl=US>, Feb 2021.
- [7] Deep Scoring Ltd. Targetscan – pistol & rifle. <https://apps.apple.com/ca/app/targetscan-pistol-rifle/id448045769#?platform=iphone>, Sep 2020.
- [8] AC APPS. Target scanner for competition shooters. <https://play.google.com/store/apps/details?id=target.scanner&hl=en&gl=US>, Sep 2020.

- [9] DISAG GmbH & Co KG. Elektronische scheibenauswertung von disag, Jul 2018. <https://www.disag.de/produkte/rm-iv/>.
- [10] Shooter's Technology LLC. Athena: The world's first affordable electronic scoring target. <http://www.orionscoringsystem.com/orion/AboutOrion.aspx>.
- [11] International Shooting Sport Federation. Results of issf certification tests for electronic scoring targets, Nov 2018. https://www.issf-sports.org/getfile.aspx?mod=docf&pane=1&inst=31&iist=29&file=Results_of_ISSF_Certification_Tests_for_Electronic_Scoring_Targets_-_Version_November_2018.pdf.
- [12] Knestel Elektronik GmbH and DISAG GmbH & Co KG. Operating instructions disag rm iv ring and target measuring machine, Sep 2020. <https://www.disag.de/wp-content/uploads/Bedienungsanleitung-Englisch-14te-Ausgabe.pdf>.
- [13] Megalink AS Electronic Scoring System. Pc-based target 10m. <https://www.megalink.no/en/products/pc-based-personal-targets/pc-based-target-10m.html>.
- [14] Megalink AS. 4k187 target. <https://www.megalink.no/en/products/10m-15m-targets/4k187-target.html>.
- [15] SIUS AG. S310. SIUS AG. <https://sius.com/wp-content/uploads/2018/07/B-KB-S310-en-1.pdf>.
- [16] SIUS AG. Ls25/50 laserscore, Oct 2020. <https://sius.com/en/product/ls25-50-laserscore-11/>.
- [17] DISAG GmbH & Co KG. Der opticscore messrahmen, Feb 2020. <https://www.disag.de/produkte/opticscore/messrahmen/>.
- [18] SPORT QUANTUM. Technology, May 2019. <https://sportquantum.com/en/technology/>.
- [19] SPORT QUANTUM. Sq 50 – interactive target for sport shooting at 25 and 50 meters, Apr 2020. <https://sportquantum.com/en/targets-shooting-sport/sq50/>.
- [20] Ramon Suarez. Sport quantum air pistol & rifle electronic target review, Jul 2020. <https://www.olympicpistol.com/sport-quantum-air-pistol-rifle-electronic-target-review/>.
- [21] Allan Brown. Freetarget. <https://github.com/ten-point-nine/freETarget/releases/tag/1.6.1>, Jan 2021.
- [22] Matthew Waterman and Donato Salazar. *Electronic Target*, pages 4 – 16. Dec 2011.
- [23] pitshooting.eu. Progressive intelligent target system, Jun 2019. <https://www.pitshooting.eu/#catalog>.
- [24] ABVisie. Match manager, Jun 2014. <https://www.abvisie.nl/downloads/matchmanager-info-en.pdf>.
- [25] Enfilade. About rifle target: Rifle shooting database. <https://enfiladestudios.com/about-rifle-target-rifle-shooting-database.html>.
- [26] Zoran Zivkovic. Improved adaptive gaussian mixture model for background subtraction. volume 2, pages 28 – 31 Vol.2, 09 2004.
- [27] OpenCV. Opencv: Open source computer vision, Oct 2019. <https://docs.opencv.org/4.1.2/index.html>.
- [28] Zoran Zivkovic and Ferdinand van der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters*, 27(7):773–780, 2006.
- [29] Satya Mallick. Feature based image alignment using opencv (c++/python), Mar 2018. <https://learnopencv.com/feature-based-image-alignment-using-opencv-c-python/>.
- [30] Adrian Rosebrock. Intersection over union (iou) for object detection, Nov 2016. <https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/>.
- [31] Dr. Andrew Greensted. Otsu thresholding, Jun 2010. <http://www.labbookpages.co.uk/software/imgProc/otsuThreshold.html>.