International Journal of Applied Mathematics

Volume 27 No. 6 2014, 549-566
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)
doi: http://dx.doi.org/10.12732/ijam.v27i6.3

MATHEMATICAL MORPHOLOGY APPLIED TO AUTOMATION OF INDOOR SHOOTING RANGES

Guilherme Pina Cardim[1], Maurício Araújo Dias[2 §]
Rodrigo Haruo Noguti[3], Ruben de Best[4], Erivaldo Antônio da Silva[5]
[1,2,3,4,5] Faculty of Science and Technology
Univ. Estadual Paulista - UNESP (São Paulo State University)
Roberto Simonsen Street, 305K
Presidente Prudente, SP, 19060-900, BRAZIL

Abstract: Target shooting is a military, sportive and for safety activity. As this activity rises, the automation of indoor shooting ranges becomes a need. Therefore, this paper proposes a program that, automatically, detects bullet holes in fixed targets and calculates score values. This purpose is achieved by a C++ code based on mathematical morphology and analysis of bullet holes locations in targets. The program is indicated to indoor shooting ranges and it works with images captured by a digital camera. Images of two types of targets were used for data collection. These targets were subjected to sessions of shots and then their images were processed. The achieved results are presented and they show the importance of this program to detect bullet holes and to calculate score values. The success rates for the bullet holes detections and scoring are approximately 97% and 100%, respectively. Thus, this work contributes to benefit the automation of indoor shooting ranges.

AMS Subject Classification: 03E99, 68U10, 68T45
Key Words: mathematical morphology, automation, computer vision, shot, target, range

---

Received: September 14, 2014

[§] Correspondence author

1. Introduction

Target shooting is practiced worldwide, ranging from sportive shooting, to private security professionals and military training. Whether for sport, or any other reason, normally the exercise is executed in indoor shooting ranges, where the participants skills are measured by the score they achieved. This score is calculated based on the position where the projectiles hit the targets, i.e. the bullet holes. Despite the fact that target shooting is related to several types of professional and sportive activities, as mentioned before, for all of them score counting is still, predominantly, done manually. In fact, the manual score counting is a time consuming and laborious task, and for this reason, recently some studies and techniques have appeared in order to automate this process.

One of those techniques currently applied uses acoustic sensors or sensors able to recognize shockwaves created by the impact of the projectiles on the targets [11]. Unfortunately, this method uses a technique that introduces a high cost for the shooting ranges. The research for less expensive alternatives originated several publications addressing computer vision.

Some of those publications, for example, have an indirect relation with target shooting and use some kind of scoring [2], [7], [15]. However, they consider only moving targets that appear in non-optical images dedicated almost always to military ends. There are also others, as the work by Yafei *et al.* [16], which detects the bullet holes, however, counts neither the holes nor the points.

The automatic score counting can be seen in a system for a kind of amateur sportive shooting, as in the project created by Fan [4]. He makes use of the Hough transform, Laplace transform, mathematical morphology and template matching. However, this system does not count partial scores, while this should occur during the intervals between the shots fired. Partial score counting is useful for following the shooters performance during a shooting session.

Partial score counting is found in literature in articles as [17]. This work presents an automatic score counting system destined to help military target shooting exercises. The system is based on the Papoulis theorem and on mathematical morphology. However, besides being complex, this system is directed only to target shooting as military practice.

Alternatively, we can find less complex methods for automatic score counting in target shooting exercises. One of them, proposed by Ali and Mansoor [1], is based on Hysteresis thresholding and mathematical morphology. However, this method works only on targets represented by concentric circumferences, while targets used in shooting ranges may vary in their geometric shape representation.

The last three studies mentioned before use mathematical morphology techniques for detecting bullet holes in targets. However, none of them executes this task by exclusively using mathematical morphology, which could bring some advantages to the system. For example, mathematical morphology is less susceptible to noise present in target images, that is, the image filtering does not need to be perfect. Moreover, mathematical morphology requires less computer resources when compared to other methods. Therefore, an algorithm based only on mathematical morphology reaches high precision applying less computational effort to detect bullet holes.

Perceiving the advantages of automatizing the detection of bullet holes and the score calculation, and circumventing the issues mentioned before, this work proposes a program for, automatically, detecting bullet holes in fixed targets and for score counting, useful for automatizing indoor shooting ranges. This proposal is satisfied by code developed in C++, based on mathematical morphology and the analysis of the bullet holes positioning on the targets.

Therefore, our work describes the development of a computer program that allows the automation of bullet holes detection and score counting. It processes images captured by a digital camera positioned approximately four meters from the target during the target shooting. This program overcomes the limitations previously mentioned by allowing the automatic detection of different types of targets, partial score counting, and for being a low complexity system with low computational costs. For that, we used mathematical morphology techniques for implementing a program that: provides routines for treating the images with the goal of improving the results; has functions for recognizing different kinds of targets; is capable of counting the amount of shots that hit the target, as well as the score achieved by the shooter.

With the intention of clarifying our work, this article is organized as follows. Section 2 introduces mathematical morphology concepts, necessary for understanding the remainder of the paper. Section 3 explains the program proposed in this work. The results achieved by the implementation of the proposed program are shown in Section 4. Section 5 brings discussions and conclusions about the work.

2. Theoretical Foundation

Mathematical morphology is considered a promising technique for images analysis. According to Soille [12], the definition of morphological images analysis refers to a theory for the analysis of spatial structures, and is so called because its focus on the analysis of objects shapes. The basis of morphology is the removal of pixels of a binary image, or the decrease of intensities in a gray-scale image, through the usage of a transformation that uses some known shape, which may be called a structuring element. In other words, morphology is based on the idea of extracting data related to the geometry and the topology of an image (an unknown set) through transformation using a structuring element (a defined set), and that, when applied to a digital object, can be called digital morphology.

Due to those peculiarities, mathematical morphology techniques are indicated for pattern and object recognition, and other applications, because they can simplify the images data, while preserving the features of the essential forms.

2.1. Basic Mathematical Morphology Operations

Mathematical morphology has as a characteristic using operations on images. Those operations consider images as being a set of pixels, and use to be decomposed in operations directed to two images classes: binary and gray scale. Considering this scope, mathematical morphology applied to images processing has two basic operations, erosion and dilation. Other operations, opening and closing, derive from those two operations and will also be explained.

2.1.1. Erosion

In an erosion operation, we consider an object A represented in an image and a structuring element B. A pixel x will belong to the resulting set only if each pixel of the structuring element, translated by x, fits in A. As all points of B, translated by x, need to be contained in A, we have the removal of pixels at the edges of the object, causing the object to be eroded [8]. In mathematical terms, the erosion of a set A (object represented in an image) by a set B (structuring element) is denoted by $A \ominus B$ and defined by:

$$A \ominus B = \{x : B_x \subset A\}, \tag{1}$$

where B_x denotes the translation of set B by x, or:

$$B_x = \{b + x : b \in B\}. \tag{2}$$

A similar idea of erosion can be extended to gray-scale images, but, in this case, the mathematical representation is:

$$(A \ominus B)(x) = \max\{y : B_x + y \leq A\}, \tag{3}$$

where B is a structuring element and A is an object represented in a gray-scale image [3].

2.1.2. Dilation

For dilation, we also consider an object A represented in an image and a structuring element B. The result of the transformation consists of, parting from a translation of x on the reflected set of B on its origin $((\breve{B})_x)$, we have all the x where the intersection of $(\breve{B})_x$ and A is different from empty. This operation adds pixels to the border of the object, causing the object to be dilated [8]. In mathematical terms, the dilation of a set A (object represented in an image) by a set B (structuring element) is denoted by $A \oplus B$ and defined by:

$$A \oplus B = \{x : (\breve{B})_x \cap A \neq 0\}, \tag{4}$$

where c denotes the complement, and:

$$\breve{B} = \{-b : b \in B\}. \tag{5}$$

A similar idea of dilation can be extended to gray-scale images, but, in this case, the mathematical representation is:

$$(A \oplus B)(x) = \min\{y : -\breve{B}_x + y \geq A\}, \tag{6}$$

where B is a structuring element and A is an object represented in a gray-scale image [3].

The dilation and the erosion are transformations dual among themselves. In other words, the complement of one erosion operation has the same result as a dilation on the complement of the image with the reflected structuring element [6].

2.1.3. Opening

The opening operating is the erosion followed by a dilation. In this operation, the dilation is used to minimize the effects caused by the erosion. However, seldom the opening, starting from the dilation step, perfectly recovers the eroded image [6], [5]. Overall, opening is used for smoothing outlines, breaking narrow isthmus in images and removing noise. In this work, opening is used to remove salt noise [3]. In mathematical terms, the opening of a set A (object represented in an image) by a set B (structuring element) is denoted by $A \circ B$ and defined by:

$$A \circ B = (A \ominus B) \oplus B. \tag{7}$$

2.1.4. Closing

The closing operation is a dilation followed by an erosion. In this operation the erosion works on recovering the original size of the dilated structures. In this work, closing is used for removing pepper noise [3]. In mathematical terms, the closing of a set A (object represented in an image) by a set B (structuring element) is denoted by $A \bullet B$, defined by:

$$A \bullet B = (A \oplus B) \ominus B. \tag{8}$$

3. Materials and Methods

The program proposed in this work was coded in the C++ programming language using the OpenCV images processing library. The option for the OpenCV library is due to the convenience provided regarding the mathematical morphology operations that were applied. This choice, then, fostered the use of C++, for being compatible with the library and also for being a proper language for the development of images processing programs.

We developed the program considering two types of fixed targets (Figure 1 and Figure 2), because they are the two types that are used the most in shooting ranges. The targets are different mainly by the way their pointing areas are defined. One of them displays the divisions between pointing areas using circumferences with the same center, and radius that grow according to an arithmetical progression (Figure 1). The other type of target has scoring areas that resemble an oval geometric shape (Figure 2).

In the images processing by the program, the sequence of steps dedicated to the detection of the holes is the same in both targets. The logic differs between both targets only for calculating the score, because this calculation is largely dependent on the type of target. It is also important to stress that the type of target is identified automatically.

The steps of the process in our program are: reading of the image of target; removal of white borders of the image; detection of bullet holes; localization of the coordinates of each detected bullet hole; identification of the type of target; localization of the target's central coordinates; calculation of the score; and exhibition of the score.

At the first step of the process, our program reads a gray-scale image of the target, for example Figure 3 or Figure 4 with dimensions, in pixels, 1944x2592 and 2592x1944, respectively.

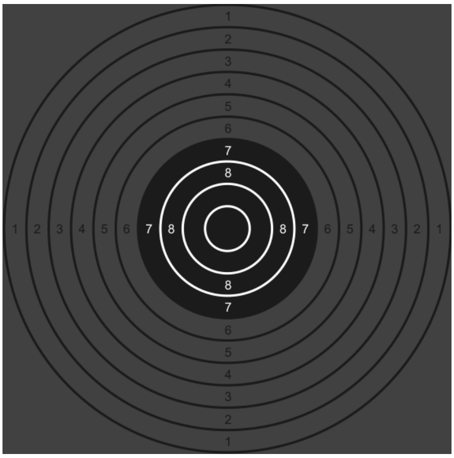

Figure 1: Target whose scoring areas are concentric circumferences.

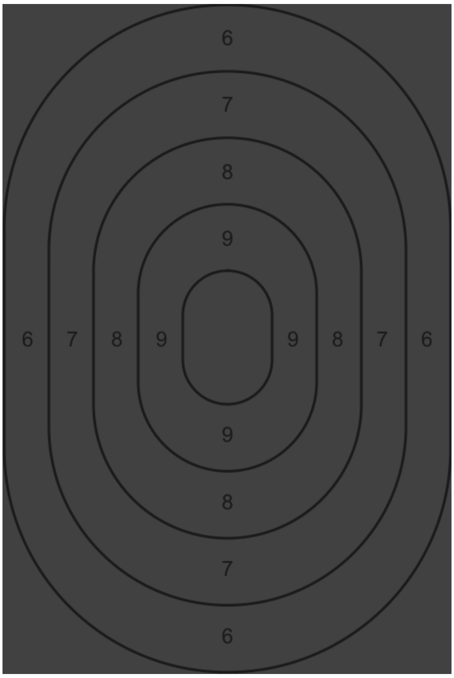

Figure 2: Target whose scoring areas are similar to concentric ovals.

The second step starts creating a new binary image from the gray-scale image, based on Otsu's Method. This conversion is needed to create two well-

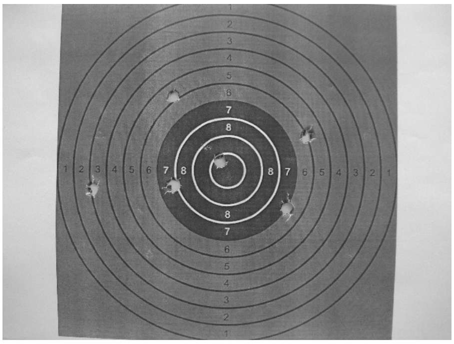

Figure 3: Bullet holes in targets whose scoring areas are concentric circumferences.

Figure 4: Bullet holes in targets whose scoring areas are similar to concentric ovals.

defined areas in the binary image: the first one represents the target area (central black area in the binary image); the second one represents the white borders (peripheral white areas in the binary image). Next, if present, upper, lower, left and right white borders of the binary image are eliminated. This task is important because our program uses the remaining central black area of the binary image as a mask to extract the exact target area from the gray-scale image.

So, the second step ends providing to the next steps a new gray-scale image containing the same original target of Figure 3 or Figure 4, but without any border. This new gray-scale image is needed mainly to facilitate the identification of the type of target, as will be explained at the fifth step.

From the third step on, we start using mathematical morphology. To remove the salt noise of the image, the program uses an opening with a disk structuring element of size 18. This way, the pixels of the salt noise are removed by the erosion, however, the holes of the shots are also changed. Aiming to make the bullet holes return to their size before the erosion, a dilation operation, regarding the second phase of the opening is applied. It should be emphasized that this step does not only remove the salt noise, but also does the holes segmentation. In this segmentation, the numbers and the central circumferences of the target are removed from the image. It is important to emphasize that the shape and size of the structuring element, in this and the next steps, are chosen based, respectively, on the type and caliber of the weapon.

Next, our program uses erosion with a disk structuring element of size 13 for finding the bullet holes. The step ends using dilation with a disk structuring element of size 12 for rounding the bullet holes, in order to highlight them.

In this work, the position of each bullet hole is given by the pair of coordinates (x, y) of the center of mass of each bullet hole found at the third step. However, to find all centers of mass, our program firstly identifies the contour and the moment of each bullet hole. The result of this fourth step is a vector containing the coordinates that show the position of each bullet hole found in the target.

As a fifth step, our program identifies the type of target, based on the shape of the gray-scale image previously provided by the third step. Being more thorough, on the one hand, the target whose scoring areas are concentric circumferences (Figure 1) has a square shape, which is very similar to the shape of the image of Figure 3 without the white borders removed by the third step. On the other hand, the target whose scoring areas are similar to concentric ovals (Figure 2) has a rectangle shape, which is very similar to the shape of the image of Figure 4 without the white borders removed by the third step. Therefore, our program explores these distinct features to easily identify the type of target. In other words, our program treats a square image as the target whose scoring areas are concentric circumferences, otherwise a rectangle image as the target whose scoring areas are similar to concentric ovals.

The main objective of the sixth step is to find the center of the target, because this information will be used by the seventh step as a reference to calculate the score. Unfortunately, to find the center of the target, our program needs to perform more tasks than only find the center of the image, since these centers infrequently have the same coordinates, although a center is always near to the other. This way, for both types of targets, the sixth step performs three main tasks: this step starts defining a mark in the center of an auxiliary image; from this mark, our program uses conditional dilation to reconstruct the most central scoring area in the same auxiliary image; this step finishes calculating the center of the reconstructed area, because the center of the most central scoring area is also the center of the target. However, the way used by our program to perform these three main tasks depends on the type of target, as will be explained in the next six paragraphs.

For the target whose scoring areas are concentric circumferences, the sixth step starts calculating the coordinates of the center of the image (Figure 5), considering its square shape. Next, our program creates an auxiliary binary image, equivalent in size, containing a mark on the center. This mark will be expanded in the auxiliary image by conditional dilation at the end of this step.

Our program also creates another binary image from the conversion of the gray-scale image, based on Otsu's Method. The main objective of this conversion is to separate the most central scoring area from the other scoring areas by a white circumference. Unfortunately, this conversion generates noises in the image. Therefore, our program uses closing, with a disk structuring element of size 3, to remove the pepper noise, followed by opening, with a disk structuring element of size 10, to remove the salt noise of the image. The opening also reconnects segments of the white circumference separated by bullet holes, guaranteeing the perfect isolation of the most central scoring area from the other scoring areas. This guarantee is important, because the most central scoring area of the target, in a complemented version of this image, will be used to define the limit of expansion in the conditional dilation.

Next, our program dilates the mark, defined in the center of the first binary image, until its expansion achieves the limits of the most central scoring area, defined in the complemented version of the second binary image. This conditional dilation uses a disk structuring element of size 7. So, the sixth step finishes finding the center of the target, by calculating the center of the most central scoring area, which is a disk in this case, previously reconstructed by conditional dilation.

For the target whose scoring areas are similar to concentric ovals, the sixth step starts calculating the coordinates of the center of the image (Figure 6), considering its rectangular shape. Then, our program generates a supplemen-

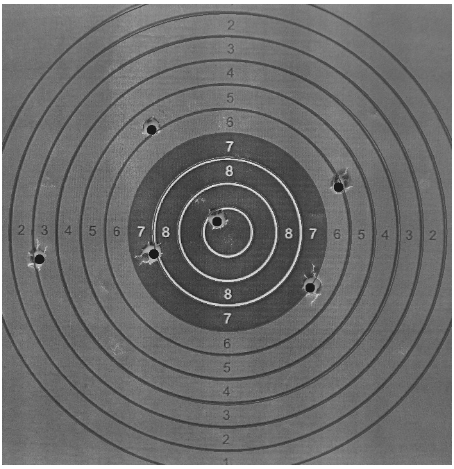

Figure 5: Position of the detected bullet holes, hihglighted in black, on targets which scoring areas are concentric circumferences.

tary binary image, with the same size, and with a mark in the center. At the end of this step, our program will use conditional dilation to expand the mark of the supplementary image, as will be explained later.

Our program uses bilateral filter to remove noises of the gray-scale image. Next, a morphological gradient, with a square structuring element of size 5, highlights the limits between the score areas (black oval lines in Figure 6). Then our program converts the gray-scale image to a binary image, since the next tasks are performed using binary morphology. After that, a closing, with a disk structuring element of size 40, is used to reconnect segments of the black oval lines separated by bullet holes, guaranteeing the perfect separation of the most central scoring area from the other scoring areas. This guarantee is important, because the most central scoring area of the target, in a complemented version of this image, will be used to define the limit of expansion in the conditional dilation.

Next, our program uses the most central scoring area, defined in the complemented version of the second binary image, as a reference limit of the conditional dilation to expand the mark, defined in the center of the first binary image. This conditional dilation uses a disk structuring element of size 5. At the end of the sixth step, our program calculates the center of the most central scoring area that seems an oval, and was previously reconstructed by conditional dilation. The center of the target is equal to the center calculated by this step.

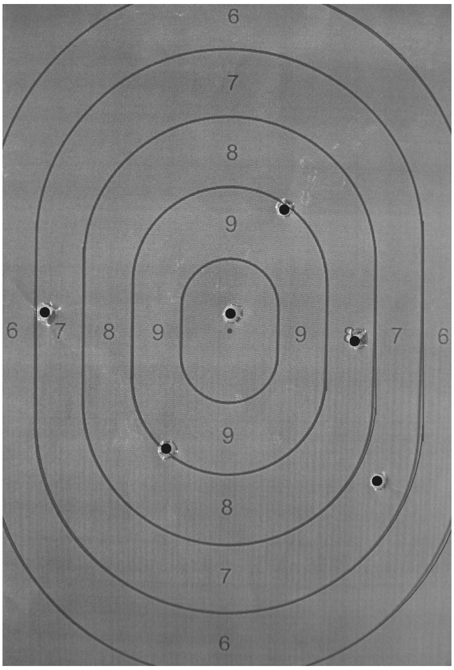

Figure 6: Position of the detected bullet holes, hihglighted in black, on targets which scoring areas are similar to concentric ovals.

At the seventh step, our program verifies the score of each shot, based on the previously-obtained positions, and conducts their sum. In this step, the process of our program follows one of two paths, depending on the image of what type of target is being processed. Therefore, we will first explain the portion of the process that refers to the target that has concentric circumferences (Figure 5), and thereafter, we will comment the logic applied for the target that has approximately oval shapes (Figure 6).

Once the central and the bullet holes coordinates are known, for the first type of target, the distances between the center of the target and the bullet holes are calculated. Based on this distance, each hit receives its score, which is added to the total score. When those steps are finished, the score achieved by the shooter is found.

Being more thorough, considering the Cartesian System of Coordinates and, without losing generality, presuming that the source of the system is at the center of the target, that is, the coordinates of the center of the target are $(0, 0)$. This way, if (x, y) are the coordinates of a bullet hole P on this target, in order to find its score we have to know in which region it is contained and what score is attributed to each region.

In the case of the regions of the target, we have two possibilities, or the shot P belongs to the smaller central disk $D_{r\text{-}1}(0,0) = \{(x,y) \in R^2; x^2+y^2 \leq (r_1)^2\}$, or the shot P belongs to some circumference, $A((0,0), r_i, r_{i+1}) = \{(x,y) \in R^2; x^2+y^2 \leq (r_{i+1})^2\}$, with $i = 1, 2, ..., 9$.

Regarding the scoring, despite of the fact that we can determine random values for each area, in this case we have that the score is linear with the radius of the circumference. Therefore, if $P \in D_{r\text{-}1}(0,0)$ then the score is 10, if $P \in A((0,0), r_i, r_{i+1})$ then the score is $(10-i)$. This defines a function $V(P)$ given by: $V(P) = 10$, if $P \in D_{r\text{-}1}(0,0)$ and $V(P) = (10-i)$, if $P \in A((0,0), r_i, r_{i+1})$, for $i = 1, 2, ..., 9$.

As $D_{r\text{-}1}(0,0) = A((0,0), 0, r_i)$, putting $r_0 = 0$, we rewrite the function V for $V(P) = (10-i)$ if $P \in A((0,0), r_i, r_{i+1})$, with $i = 0, 1, ..., 9$. Considering that $P = (x,y) \in A((0,0), r_i, r_{i+1})$ only, and only if, $(r_i)^2 < x^2+y^2 \leq (r_{i+1})^2$.

Regarding the second type of target, the score calculation is a little more complex and requires other methods. Each scoring area was decomposed in three parts: a central rectangle and two semi-circumferences at their rounded extremities. This way, with the intention of verifying the score of a bullet hole, it is only necessary to check whether it is contained in any of those parts. It is important to highlight that this process happens for every scoring area. The procedure is repeated from the center to the extremities of the target. Initially, we analyze the rectangular area making use of intervals between certain points $(x1, y1)$ and $(x2, y2)$. Concerning the semi-circumferences, it was needed to find the coordinates (x,y) that represent the centers of the circumferences that contained each of the semi-circumferences. Once the central positions and the radius of the semi-circumferences were known, it was possible to use the same concept of the first target.

Considering, also for this case, that the origin $(0,0)$ of the Cartesian Coordinates System is the center of the target. This way, if (x,y) are the coordinates of a bullet hole P in this target, in order to find its score we have to know in which region it is contained and what score is attributed to each region. For describing the region in this second type of target, we may consider the linear function $f(x) = kx$. This function basically represents a graphic where the line passes through the origin and has an inclination k. From this point forward, we will adopt $k > 1$. Thus, given a real positive number r_1, lets take the sets:

- $Qr_1 = \{(x,y) \in R^2; |x| \leq r_1 and |y| \leq kr_1\}$: rectangle with vertices at the points (r_1, kr_1), $(-r_1, kr_1)$, $(-r_1, -kr_1)$ and $(r_1, -kr_1)$, or, Qr_1 it is the rectangle with dimensions $2r_1$ (width on the x axis) e $2kr_1$ (height on the y axis) and which diagonals intersect exactly at the point (0, 0);

- $Sr_1(0, kr_1) = \{(x, y) \in R^2; x^2 + (y - kr_1)^2 \leq (r_1)^2$ and $y \geq kr_1\}$: north hemisphere of the disk with center $(0, kr_1)$ and radius r_1;
- $Sr_1(0, -kr_1) = \{(x, y) \in R^2; x^2 + (y + kr_1)^2 \leq (r_1)^2$ and $y \leq -kr_1\}$: south hemisphere of the disk with center $(0, -kr_1)$ and radius r_1;
- $Rr_1 = Qr_1 \cup Sr_1(0, kr_1) \cup Sr_1(0, -kr_1)$.

If $r_2 > r_1$, using the same construction above, we obtain the region Rr_2 and observe that $Rr_1 \subset Rr_2$. We can then define circumference $A(r_1, r_2) = Rr_2 - Rr_1$. In order to finish the description of this target, let's consider numbers $0 < r_1 < r_2 < ... < r_6$ and the circumferences $A(r_i, r_{i+1}) = Rr_{i+1} - Rr_i$, where $i = 1, 2, ..., 5$. Therefore, as seen at the first type of target, we can define the score as follows: $V(P) = 10$ if $P \in Rr_1$ and $V(P) = (10 - i)$ if $P \in A(r_i, r_{i+1})$, for $i = 1, 2, ..., 5$. We also emphasize that, in both cases, despite that fact that we know the numbers r_i, we made a more general presentation.

Images can show bullet holes equally distributed among two scoring areas. In other words, this is the case where half of the hole is located in one smaller scoring area, while another half is located in a larger scoring area. An example of this situation can be seen in Figure 3, where a bullet hole is between the area of two and three points. We bypass this problem considering the average of the scores for all the occurrences that have this characteristic.

At this point, as an optional step, the program can highlights the score areas and the positions of the bullet holes found in the images of the targets (respectively, the lines and the dark disks in Figures 5 and 6). At the eighth and last step, our program shows the score achieved by the shooter.

4. Results

For assessing the performance of the program in the detection of bullet holes and the score counting, target images subjected to shooting sessions were processed by the program, and, thereafter, analyzed. Because there are strict laws for the acquisition, possession, handling and shooting of firearms in Brazil, alternatively, cylinders with half a centimeter of diameter were used as projectiles. Those cylinders were launched manually on the targets, from approximately four meters, in order to simulate shots fired with firearms. After each shooting session, an image of the target hit was captured by a digital camera. For every type of target subjected to the shooting experiments, twenty five images were used to be submitted to the program for the detection of bullet holes and score counting. The quantitative validation of the detection of bullet holes was based on adapted version of the following metric [10], [9]:

$$M_T = (VP_T + VN)/(VP_T + VN + FN_T + FP_T). \tag{9}$$

In Equation 9: M_T is the percentage of detection of bullet holes T; VP_T is the number of true positives, meaning pixels correctly identified as belonging to the bullet holes, or the bullet holes correctly detected; VN is the number of true negatives, meaning pixels correctly identifies as being part of the target; FN_T is the number of false negatives, meaning pixels belonging to bullet holes incorrectly identified as belonging to the target, or the bullet holes undetected; FP_T as the number of false positives, meaning pixels belonging to the target incorrectly identified as belonging to holes, or not-existing bullet holes incorrectly detected as existent ones.

According to this metric, the detection rate of bullet holes is approximately 97%. Regarding the bullet holes correctly detected in the samples, the rate registered for the counting of the scoring is 100%. Equation 10 represents the success rate P_T for the counting of points, where PC is the amount of hits that received the correct score. In this case, VP_T is the hit rate, which is calculated considering only the bullet holes correctly identified,

$$P_T = PC/VP_T. \tag{10}$$

Note that, the program based on mathematical morphology presented in this work performs the detection of bullet holes in fixed targets and the score counting successfully.

5. Discussion

This work proposed a program for, automatically, detecting bullet holes in fixed targets and score counting, useful for automating shooting ranges, created in closed environments. This way, this proposal was attended by a code produced in C++, which automatically performs the detection of bullet holes in fixed targets and the score counting. This program makes intense use of morphological mathematics applied to images processing, and analysis of bullet holes locations in targets, with the goal of overcoming limitations found in other articles published in literature [11], [2], [7], [15], [16], [4], [17], [1], already commented in the section Introduction. It is important to note that our program has the convenience of having a low computational cost, provided by the characteristics of mathematical morphology.

Besides that, based on the metric shown in Equation 9, it is possible to demonstrate that the implementation of our program has a high success rate - close to 97% - at the detection of bullet holes. Also through the experiences made, it is noted that, regarding score counting, we obtained a success rate of 100%. Those rates outweigh the results shown by [17] and [1].

Therefore, the current work contributes: providing the automation of indoor shooting ranges; presenting low computational cost due to the use of mathematical morphology; facilitating the implementation; allowing partial score counting; performing the score counting for targets with different shapes; performing the target shooting score counting for multiple practices; automatically detecting the type of target that is being processed.

5.1. Conclusions

We conclude that the program presented in this work adequately meets the proposal of automatically performing the detection of bullet holes in fixed targets and the score counting.

Nevertheless, future works can be performed considering international standard targets, the use of countless types of arms and ammunition, a bigger interval for the minimal and maximal distance between the target and the camera, the modification of the dimensions of the images in the targets, and a more precise overlaid bullet holes detection, using, for example, *support vector machine* [14], [13].

References

[1] F. Ali and A.B. Mansoor, Computer vision based automatic scoring of shooting targets, In: *12th IEEE Intern. Multitopic Conf. (INMIC 2008)*, IEEE, Karachi (2008), 515-519, DOI: 10.1109/INMIC.2008.4777793.

[2] K. Augustyn, A new approach to automatic target recognition, *IEEE Transactions on Aerospace and Electronic Systems*, **28**, No 1 (1992), 105-114, DOI: 10.1109/7.135437.

[3] E.R. Dougherty and R.A. Lotufo, *Hands-on Morphological Image Processing*, SPIE Publications, Washington (2003).

[4] Xinnan Fan, Qianqian Cheng, Penghua Ding and Xuewu Zhang, Design of automatic target-scoring system of shooting game based on computer vision, In: *IEEE Intern. Conf. on Automation and Logistics (ICAL '09)*, IEEE, Shenyang (2009), 825-830, DOI: 10.1109/ICAL.2009.5262810.

[5] R.C. Gonzalez and R.E. Woods, *Digital Image Processing*, Prentice Hall, Upper Saddle River (2007).

[6] B. Jahnee and H. Haubecker, *Computer Vision and Applications: A Guide for Students and Practitioners*, Academic Press, San Diego (2000).

[7] D. Musicki, *Track Score and Target Existence*, Technical Report, University of Melbourne (2007).

[8] M.S. Nixon and A.S. Aguado, *Feature Extraction and Image Processing*, Elsevier, Oxford (2008).

[9] A. Prati, I. Mikic, M.M. Trivedi and R. Cucchiara, Detecting moving shadows: algorithms and evaluation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **25**, No 7 (2003), 918-923, DOI: 10.1109/TPAMI.2003.1206520.

[10] R.J. Radke, S. Andra, O. Al-Kofahi and B. Roysam, Image change detection algorithms: a systematic survey, *IEEE Transactions on Image Processing*, **14**, No 3 (2005), 294-307, DOI: 10.1109/TIP.2004.838698.

[11] C. Sanctuary, A. Sean and S.R. Hsieh, Remote strafe scoring system, *U. S. Patent 4813877* (1989).

[12] P. Soille, *Morphological Image Analysis*, Springer-Verlag, New York (1998).

[13] Qing Song, Wenjie Hu and Wenfang Xie, Robust support vector machine with bullet hole image classification, *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, **32**, No 4 (2002), 440-448, DOI: 10.1109/TSMCC.2002.807277.

[14] W.F. Xie, D.J. Hou and Q. Song, *Bullet-hole Image Classification with Support Vector Machines*, Technical Report, Nanyang Technological University (2000).

[15] S.K. Wong, High range resolution profiles as motion-invariant features for moving ground targets identification in SAR-based automatic target recognition, *IEEE Transactions on Aerospace and Electronic Systems*, **45**, No 3 (2009), 1017-1039, DOI: 10.1109/TAES.2009.5259180.

[16] Shao Yafei, Zhang Li and Guowei, Wu, The visual detection of bullet holes in a target, In: *5th Intern. Conf. on Signal Processing Proc. (WCCC-ICSP 2000)*, **2**, IEEE, Beijing (2000), 914-917, DOI: 10.1109/ICOSP.2000.891669.

[17] Cuiliu Ye and Hong Mi, The technology of image processing used in automatic target-scoring system, In: *2011 Fourth Intern. Joint Conf. on Computational Sciences and Optimization (CSO)*, IEEE, Kunming and Lijiang City (2011), 349-352, DOI: 10.1109/CSO.2011.287.