International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Automatic Camera Calibration Using a Single Image to extract Intrinsic and Extrinsic Parameters

Rui P. Duarte*[1], Carlos A. Cunha[2], José C. Cardoso[3]

Submitted: 28/01/2024 **Revised**: 06/03/2024 **Accepted**: 14/03/2024

Abstract: This article presents a methodology for accurately locating vanishing points in undistorted images, enabling the determination of a camera's intrinsic and extrinsic parameters as well as facilitating measurements within the image. Additionally, the development of a vanishing point filtering algorithm is introduced. The algorithm's effectiveness is validated by extracting real-world coordinates using only three points and their corresponding distances. Finally, the obtained vanishing points are compared with extrinsic parameters derived from multiple objects and with intrinsic parameters obtained from various shapes and images sourced from different test sites. Results show that through a single image, the intrinsic parameters are extracted accurately. Moreover, Using 3 points to determine the extrinsic parameters is an excellent alternative to the checkerboard, making the method more practical since it does not imply the manual positioning of the checkerboard to perform the camera calibration.

Keywords: *automatic detection, camera calibration, extrinsic parameters, intrinsic parameters, vanishing points*

1. Introduction

For a camera to acquire good-quality images, it must be well-calibrated. Thus it will be possible to carry out measurements with a high degree of accuracy between objects in an image scene. Calibrating a camera consists of obtaining matrices, representing the intrinsic and extrinsic parameters of a camera, that is, their internal and external parameters. This calibration can be done in various ways, each with advantages and disadvantages [1]. If we use a checkerboard to calibrate the camera, the matrices obtained will make it possible to extract data about the image distortion caused by the lens [2]–[5]. The problem with this method is that it is not automated, so we need to position the checkerboard in several places for each camera to acquire enough images to obtain good results.

To automate the process, one can use the vanishing points of an image [6], [7], which can be detected using a neural network (which increases the complexity of the detection), or with methods that can extract enough data from an image to determine the vanishing points. The problem with these methods is that they only allow the calculation of the intrinsic camera parameters, thus lacking the extrinsic parameters to measure the image.

For the extrinsic camera parameters, a checkerboard can be used, where it is possible to detect the points of the checkerboard automatically, or several points of an image can be selected manually, provided that their correspondences in the real world are known. The problem with these two methods, manual and automatic selection of points to use, is the need to take measurements in the real world before they can be used. Therefore, one key question arises: can a matrix for the extrinsic parameters be obtained effectively without knowledge of the real-world measurements?

This paper aims to determine the intrinsic and extrinsic parameters of a camera in a practical and automatic way, to make measurements on an image with the slightest possible error. This work shows that it is possible to extract all the necessary information using a single image to determine the required parameters as accurately as possible. To get the best results, the calibration results using the checkerboard will compare with the calibration accomplished using the manufacturer's data and vanishing points. The diverse manners in which the extrinsic parameters are determined will also be compared to find out which is the best method, among other types of tests, to analyze all the fields that can introduce errors in the final result.

The remainder of this paper organizes as follows. Section 2 summarizes the state of the art, and the methods discussed are deepened. Subsequently, Section 3 applies the methods studied and presents a detailed demonstration of each step. In Section 4, the results obtained will be presented, along with the analysis of the methods developed. Finally, in Section 5, the work concludes by presenting a critical analysis regarding the final state of the developed application and the results obtained. Also, it presents considerations for future work that may improve the results.

[1] *CISeD – Research Center in Digital Services, Viseu, Portugal*
ORCID ID : 0000-0002-6819-0985
[2] *CISeD – Research Center in Digital Services, Viseu, Portugal*
ORCID ID : 0000-0002-2754-5401
[3] *Polytechnic Institute of Viseu, Viseu, Portugal*
** Corresponding Author Email: pduarte@estgv.ipv.pt*

2. Method

This section describes fundamental concepts for calibrating a camera through image capture. First, the basic model of all studies, the pinhole camera, is described. Next is described one of the first implemented methods, using a checkerboard in the scenario. Moreover, a method that involves two static cameras, called stereo camera calibration, is capable of determining the depth of objects in the scene. Subsequently, it describes methods that will aid in detecting vanishing points using a single camera and image. Finally, it details a view of methods capable of detecting and identifying objects in an image, which can improve the accuracy of the measurements.

2.1. Pinhole Camera

The pinhole camera consists of a box of opaque walls with a small hole in one of the walls through which the rays of light pass, and the projection of the objects is inverted in both axes because the rays of light all cross in the hole. The pinhole camera is still used today [8]–[10], with the addition of one or more lenses at the hole location, which allows reaching more distant light rays (telephoto/long range lens) or even lateral light rays (wide angle lens). This change causes some distortion in the images, a distortion that can be of various types and different intensities: barrel, pincushion, and waveform [11].

The projected image is extracted from the following relations. Let C be the center of the camera, C_p is the projection of C in the image plane, A is any point in the world, A_p is the projection of A in the image plane, and f the focal length of the camera. It is possible to detect similarities between the triangle $\Delta C_p A_p C$ and the triangle ΔABC. Knowing that both triangles have equal angles, which makes them similar, therefore, knowing the value of c, for example, it is possible to know the value of b through the fundamental theorem of similarity of triangles given by $c/a = b/f$.

2.2. Manual Calibration

The manual calibration, performed with the aid of a chekcboard (cf. Fig. 1), is based on a set of images where the more images are used - continuously varying the position and rotation of the board concerning the camera - the better the result will be. Data related to the distortion caused by the lens in the image can also be acquired. This type of calibration also requires no pattern on the image, similar to the board used, so there are variations of the algorithm where circles are used instead of squares, for example.

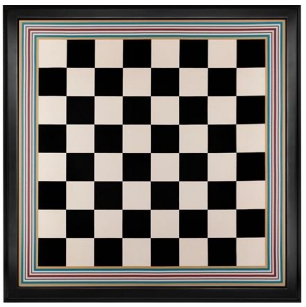

Fig. 1. An example of a checkboard.

One of the inputs in the algorithm is the size of the board to be used, whose value is obtained not based on the number of squares it has but on the number of intersections. These points are obtained based on where the black squares touch. The board in Fig. 1, for example, has a size of 7×7.

To obtain the extrinsic parameters, the checkerboard is used as a reference. This board defines the XY plane, where each side of the square corresponds to 1 unit in the real-world coordinate system or to the respective measure of the square, which allows obtaining the coordinates in centimeters, meters, or kilometers, depending on the unit of measure used to measure the side of the square of the board. By considering the side of the square to be 1 unit of measure, it is required to multiply the coordinates obtained by its value in the real world.

Once both camera parameters are determined (intrinsic and extrinsic), it is possible to use equation (1), where $(x, y, 1)$ are the homogeneous coordinates in the image plane and $(X, Y, Z, 1)$ the homogeneous coordinates in the real world, to determine the projection of any real-world point in the image plane knowing the values of X, Y, and Z in the real-world coordinate system. The inverse is also possible. However, it requires considering Z=0, since obtaining three values from two and being limited to measurements in one plane is impossible.

$$\begin{bmatrix} 1 \\ a \\ c \end{bmatrix} = K[Rt] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{1}$$

Equation (1) can be simplified into (2) by considering $P = K[Rt]$

$$\begin{bmatrix} 1 \\ a \\ c \end{bmatrix} = P \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2}$$

2.3. Stereo Camera Calibration

Calibrating with a stereo camera is commonly used in virtual reality games [12], [13]. Two cameras strategically positioned at a certain distance allow the stereo calibration, making it possible to determine how far away the objects are from the camera by detecting the differences between the images taken by the two cameras. The differences can be

concerning the position of the objects in both images [14], based on the fact that more distant objects vary their position less than closer objects, or the color difference [15].

Fig. 2 represents the basic idea for determining the depth at which an object lies concerning the cameras. Points O_l and O_r represent the center of the cameras used, and points p_l and p_r are the projections of point *P* on the image plane of camera *l* and *r*, respectively. In Fig. 2, *f* represents the focal length of both cameras, *Z* represents the depth at which point *P* locates, and *T* represents the distance of the two cameras.

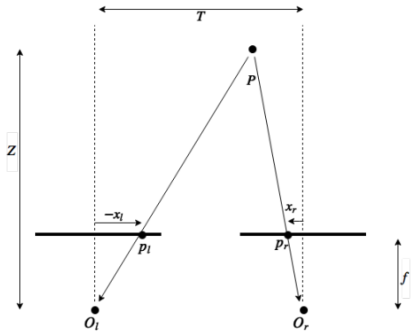

Fig. 2. Determining depth by triangulation.

To calibrate a stereo camera, the two cameras calibrate separately, so the exact real-world coordinates of an object must be used, using images captured simultaneously (without delay in any of the cameras). The formula used is the same as when using a checkerboard to calibrate the camera (given by (2)).

2.4. Vanishing Points

The vanishing points are formed by the intersection of two lines of the image that are parallel in the real world. In an image, each object will have its vanishing points unless they have some parallel side or edge, so there can be thousands of vanishing points if there are many variations between the various objects. The union of these points forms a line called the vanishing line.

One of the first techniques to emerge was photogrammetry [16], a method that uses several images to reconstruct an object based on its various perspectives. This technique is also used in the automotive area to build body kits or even to design and test the plotting of a vehicle. Photogrammetry is also used in archaeology [17] to study places where a person cannot enter, as was the case with the hidden chambers in the pyramids in Egypt.

Using vanishing points, measuring the height of something in an image is also possible based on some reference measurement, as presented in [18]. When using vanishing points, it is not necessary to calibrate the camera, but it is necessary to know where the vanishing lines are. Fig. 3 shows the blank horizon line (one of the vanishing lines), the reference measurement between t_r and b_r, the measurement to be determined using the vanishing points represented by the distance between *t* and *b*, and the intersection of the line segment $\overline{t_r b_r}$ and $\overline{tb}$ with the vanishing line, point i_r and *I*, respectively. The intersection points i_r and *i* are used as the basis for the scale calculation, which allows the distance between *t* and *b* to be found from the distance between t_r and b_r.

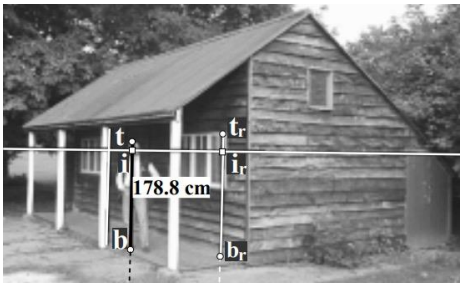

Fig. 3. Measurement example on an image with the vanishing points [18].

In order to calibrate a camera using vanishing points, several complementary methods are used. Canny's method [19] is a multi-stage algorithm that allows the detection of all possible edges in an image, and the Hough Transform [20] is a method that allows the detection of complex geometric shapes.

Canny's method focuses on three aspects defining a good edge detection algorithm. A good edge detector should be able to detect all possible edges in the image, detect the edges accurately, and should also detect the edges only once. The algorithm was broken down into several stages to fulfill the three fundamental aspects. In the first stage, the Gaussian blur is applied to remove any noise from the image. Next, the image's gradient intensity is detected, allowing an image with blur to detect horizontal, vertical, or even diagonal lines. The third stage eliminates parts of the image that might interfere with edge detection. To achieve this effect, a threshold applies to the image. Next, in the fourth stage, stage three is repeated to ensure that everything that might interfere with good edge detection has been removed. Finally, hysteresis [21] is used to detect edges more accurately (which consists of a physical property that indicates the tendency of a system to retain its properties in the absence of a stimulus.

2.5. Object Detection

There are currently two types of object detectors [22]: the traditional detector that uses a library of images with the location and respective identification of the objects, and the conventional detector that uses neural networks to learn to identify the objects, that is, instead of using a library of images generated, the algorithm will generate its own. An example of a neural network is Google's reCAPTCHA [23], where the algorithm developed learns to identify objects based on the answers given by users.

There are several ways to detect an object in an image.

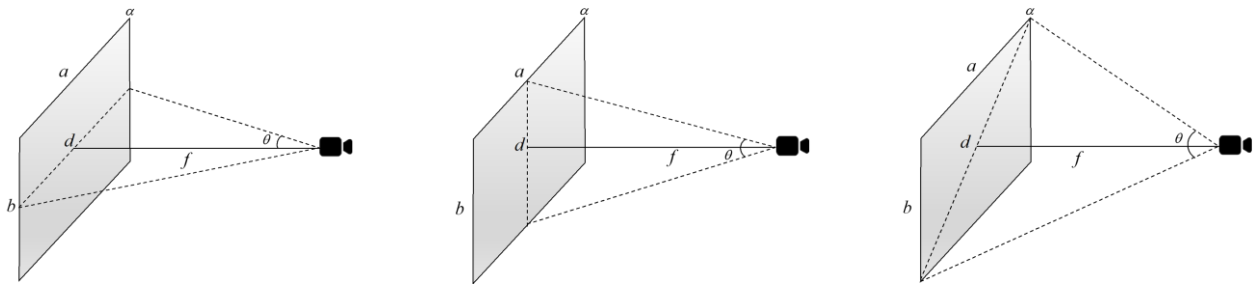

Fig. 5. Schematic for each type of field of view.

Initially, all possible patterns in an image were analyzed, which takes a long time and requires more machine resources. Later, as technology and methods developed, a grid division was used, where only the sections where an object is likely to present are analyzed. This method makes the algorithm much faster, although it can sometimes be less accurate [24]. YOLO [25], for example, uses this method, which makes it one of the fastest object identification algorithms nowadays. The architecture of YOLO divides into three stages. In the first stage of the algorithm, a grid is generated that is used by the algorithm in stage two to identify the areas where some object is more likely to exist. Finally, in stage three, the identification of the objects detected by the algorithm (in the areas selected in phase two).

3. Camera Calibration

The calibration of the camera is performed in two steps. First, it is necessary to obtain the intrinsic parameters and then the extrinsic parameters (in this order).

3.1. Manufacturer's Intrinsics

Using data provided by the camera manufacturer, it is possible to determine the camera's intrinsic parameters using some calculations. To this end, obtaining the field of view and the camera's resolution will be necessary. One of the first validations to consider in determining the offset, which can cause image distortion. In this work, we consider that the lens has no image distortion, and its center coincides with the center of the image, as shown in Fig. 4. Here, point *C* is the center of the camera, α the image plane, a the width of the image, b the height of the image, and f the focal length.

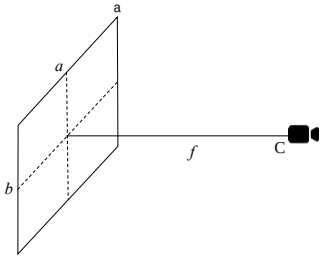

Fig. 4. Schematic showing the center of the camera and the picture plane.

The manufacturer can provide the field of view value of the camera in several ways, so, based on the scheme in Fig. 4 and the data that the manufacturer provides, it is possible to obtain the schemes in Fig. 5 where θ is the field of view, and d is the size of the line segment formed by the field of view in the image plane.

All schemes have the triangle formed by the field of view, and if the triangle is divided in half and the tangent formula is applied to half of the angle θ, (3) is obtained.

$$f = \frac{d}{2\times\tan\left(\frac{\theta}{2}\right)} \tag{3}$$

The camera used in the tests is the Logitec C930e, whose diagonal field of view is 90^0, which is equivalent to $\pi/2$ *radians*. The resolution used to capture the images is 1920×1080 *px*, so d is given by $d = \sqrt{1920^2 + 1080^2} = 2202.91$. Replacing the value of d and θ in (3), *f=1101.46.*

Once the focal length of the camera is determined one can obtain the matrix of intrinsic parameters of a camera, given by (4), where f is the focal length, C_x the x-coordinate of the projection of the center of the camera lens onto the image plane, and C_y the y-coordinate.

$$\begin{bmatrix} f & 0 & C_x \\ 0 & f & C_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1101.46 & 0 & 960 \\ 0 & 1101.46 & 540 \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

3.2. Intrinsics of the Vanishing Points

The intrinsic parameters can be obtained from the vanishing points by means of three approaches: using different methods to detect the lines, using different methods to extract the points from the detected lines, or taking a different approach, such as using neural networks.

A gray-scale image is used to obtain the intrinsic camera parameters through the vanishing points, which allows faster and simpler processing of the images so that there is only one color matrix instead of three if the image is RGB [26], [27]. First, applying some blur to the image - using the Gaussian Blur method - will eliminate some noise and lines that may arise. After applying the blur, the Canny algorithm detects the edges of all objects in the image. These edges

will be used in the next step by the Hough Transform method to detect the lines in them.

The Hough Transform method contains a threshold that makes the algorithm more sensitive. In order to make the detection of lines more dynamic, the algorithm is reapplied with a smaller and smaller threshold, starting with a high value, which makes the algorithm less sensitive, and ending in 1, the minimum value accepted by the method. The lines returned by the Hough Transform method will only be accepted if at least 100 lines are detected during the process. This procedure can compare with the progressive depth search, which is a search method that does a depth search limiting itself to expand the nodes only up to a certain level, increasing the limit with each pass of the algorithm if the goal has not been reached

To obtain the vanishing points from the detected lines, all the detected lines will be intersected, resulting in all the points that may become vanishing points. For this purpose, it is considered the line segment $\overline{AB}$ and the line segment $\overline{CD}$, where *a* represents the cross product of *A* with *B* and *b* the cross product of *C* with *D*. The value of the cross product of *a* with *b* defines the intersection of the line formed by points *A* and *B* with the line formed by points *C* and *D*, which represents one of the possible vanishing points, given by $V_p = (A \times B) \times (C \times D)$ in homogeneous coordinates (pass from (x, y) to $(x, y, 1)$). Once the homogeneous coordinates of the points are obtained, the homogeneous coordinates of the vanishing point can be determined. Finally, it will be necessary to return to the image coordinate system by dividing the *x*-coordinate and the *y*-coordinate by the *z*-coordinate, thus obtaining the coordinates of the vanishing point in the image plane.

Once all possible points have been obtained from the previously detected lines, it is necessary to filter the data due to the fact that there is not only a mixture of vanishing points from various objects but also due to the fact that many of the detected points are not vanishing points at all. Therefore, it is important to determine how the vanishing points can appear. In an image, they represent the intersection of the axes of the coordinate system of the object in question with the image plane, as shown in Fig. 6. The letter *C* represents the origin of the coordinate system, and the letters *A*, *B*, and *D* represent the intersection of the coordinate system with the image plane.

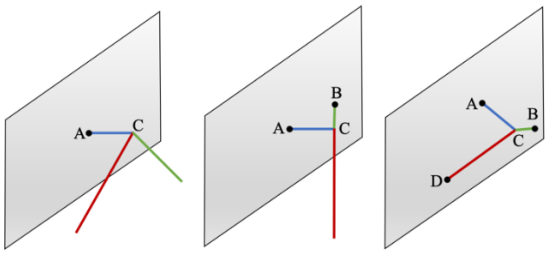

Fig. 6. Vanishing points detection.

To filter the vanishing points in order to obtain one of the previous situations, it is first necessary to find the most likely locations where they might be located. The method presented in [28] is used to find these locations, where the image is divided into several smaller sections. Then the section with the most points is detected, and this section is the place where the vanishing point may be located. The algorithm was adapted to detect the three most significant areas instead of the largest to find three vanishing points. Changes were also made to detect areas outside the camera's field of view so it is determined how much the image has to be expanded in all directions so that all points are "inside" the field of view. The coordinates of the points are also adjusted to remain in the exact location after the expansion. After the first pass of the algorithm, a few more will follow using the same algorithm, but instead of detecting the three most significant areas, the goal will be to detect the most prominent area. This process aims to find the exact location of the vanishing point within each of the previously detected cells since the point may not be in the center of the cell but in one of its corners.

After filtering the detected points, one should have, at best, a list with 3 points. To verify if these points can be considered vanishing points, the center of the triangle must coincide with the projection of the lens center on the image C_p, whose coordinates must coincide in turn with the center of the image *O*. Since the points considered may not have been detected correctly, an error margin was stipulated where the calculated center can be distant from the image center up to 5% of the diagonal. Next, if this is not the case, all possible pairs with the 3 points are formed. Since it is not possible with 2 points to calculate the projection of the lens center on the image C_p, the coordinates of the image center on the line formed by the 2 points *(A* and *B*) will be considered, as shown in Fig. 7. Next, it is checked if the coordinates are between the 2 points under test. Suppose it is verified that the projection of the image center belongs to the straight-line segment $\overline{AB}$. In that case, these will only be vanishing points if the distance between the obtained and image centers is within the stipulated error margin. If validation is not possible for each pair of points, it will be considered that only a single vanishing point exists whose coordinates coincide with the center of the image. It is checked to validate the points located closest to the image center and within the stipulated error margin.

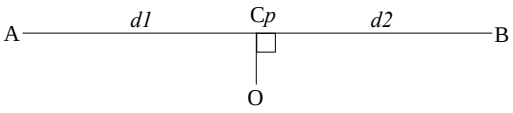

Fig. 7. Schema for two points.

After the filtering is complete, one of the situations referred in Fig. 6 should occur. It is also possible that none of the vanishing points are valid, so it will not be possible to

determine the intrinsic camera parameters. If only one vanishing point is detected, it will only be possible to extract the projection of the lens center onto the image plane. To make the matrix calculation possible, even if only one vanishing point is detected, the focal length is calculated based on the data provided by the manufacturer. This data cannot be validated, so there should be an option to disregard it. The calculation of the intrinsic parameters is only possible with two or three leak points. If three vanishing points are detected [29], it is possible to extract the projection of the lens center on the image plane through the center of the triangle formed by points *A*, *B*, and *D*. The point C_p coincides with the center of the triangle ΔABD because the angle $\angle ACB$, $\angle ACD$, and $\angle BCD$ are 90º. As for the focal length, it is possible to extract it considering $f = d_1 \times d_2 - d_3$ (see Fig. 8). Considering the triangle ΔACB rectangle at vertex *C*, if its base is the side $\overline{AB}$, the height is given by $h = \sqrt{d_1 \times d_2}$.

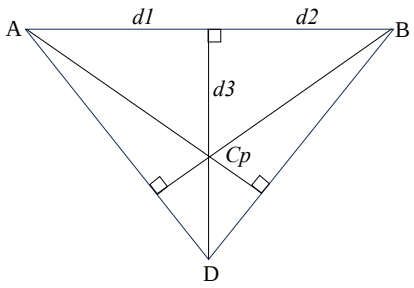

Fig. 8. Schema for three points.

When the object has only two vanishing points, it becomes impossible to know the point C_p since it can be at any point of the line segment $\overline{AB}$. To get around this problem, it will be considered that the point C_p will never be too far from the center of the image. Thus, the point *O* represents the center of the image in order to obtain the approximate point C_p and, therefore, the value of d_1 and d_2; d_3 will have the value zero because the distance between the point *M* and the point C_p, identified in the scheme of Fig. 8, is null when there are only two vanishing points. Once all the data is determined, the matrix of the intrinsic parameters of the camera is calculated.

3.3. Obtaining Extrinsic Parameters

To obtain the extrinsic parameters, the function *solvePNP* belonging to the *OpenCV* library is applied. This function uses (5) to obtain the extrinsic parameters matrix $[R \quad T]$, obtained by joining the rotation matrix *R*, with the translation vector $\vec{t}$. It uses intrinsic parameters matrix of the camera and associations between the homogeneous coordinates in the image plane $(x, y, 1)$ with those of the real world $(X, Y, Z, 1)$.

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & C_x \\ 0 & f & C_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (5)$$

To obtain the extrinsic parameters, we can use the known coordinates of the checkerboard that can be extracted automatically, or else, if possible, we can select the known points of some object that is always on the ground, such as a tile. This selection can be made manually or through some algorithm that recognizes the object in the image and makes the respective manual selection.

3.3.1. Extrinsic with only Three Points

When only three points are selected, a third variable, besides the distances between the points and their coordinates, can be passed to the program. This variable is the angle at each triangle vertex formed by the 3 points. To avoid the need to measure angles, only the three sides of the triangle will be measured, whose values are sufficient to determine the internal angles of the triangle, among other types of data relative to it.

To obtain the extrinsic parameters of the camera, it is necessary to create associations between the coordinates in the real world and those of the image. For that, consider one of the sides of the triangle as being the *X* axis of the coordinate system, being point *A* (leftmost vertex of the side selected to be the *X* axis), the origin of the coordinate system. To determine the coordinates in the real world through the sides of the triangle, the area of the triangle can be obtained in two ways, that is, using the base and the height of the triangle $(a = c \times h/2)$, or through the perimeter of the triangle $(a = \sqrt{s \times (s-a) \times (s-b) \times (s-c)})$, where the value of *s* can be obtained through $s = (a + b + c)/2$, which represents half the perimeter of the triangle. Fig. 9 presents the locations of the triangle from which the measurements are extracted, as well as the points that will be used as a reference for the calculation of the extrinsic parameters of the camera.

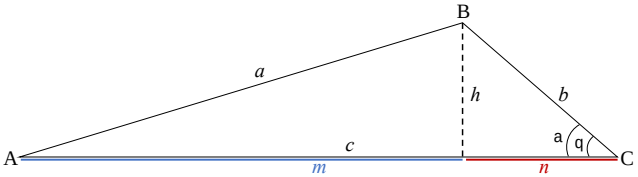

Fig. 9. Non-obtuse triangle for the extrinsic parameters.

Using the two formulas to determine the area *a*, the height of the triangle can be obtained using (6):

$$h = \frac{2}{c}\sqrt{s \times (s-a) \times (s-b) \times (s-c)}) \quad (6)$$

Once extracted the height of the triangle, it becomes possible to obtain the *Y*-coordinate of vertex *B*; however the

X-coordinate is still required. Its value can be determined by calculating the values of m and n; n is obtained through trigonometrical calculations. From the value of h and b, the value of θ is given by $\alpha = \sin^{-1}(h/b)$, and $n = b \times \cos(\alpha)$. The value of m is given by $m = |c - n|$, if $\theta \leq 90^0$; otherwise, $m = |c + n|$, as illustrated in Fig. 10.

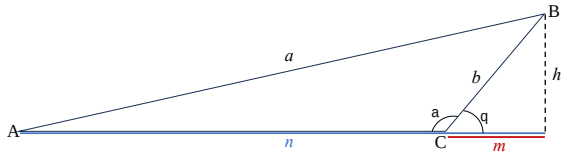

Fig. 10. Obtuse triangle in C for the extrinsic parameters.

From m and n, the X-coordinate of vertex$B$ is determined using $m = |c + n|$, if $\theta \geq 90^0$. For $\theta < 90^0$, $m = c - n$ to allow the possibility of obtaining negative coordinates, as shown in Fig. 11.

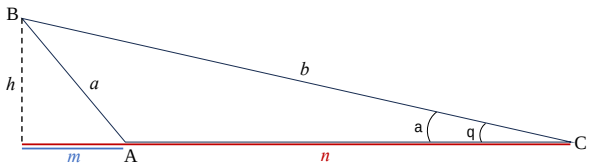

Fig. 11. Obtuse triangle in A for the extrinsic parameters.

By determining m, n, and h, one obtains the coordinates $A = (0,0,0)$, $B = (m,h,0)$, and $C = (c,0,0)$, which are used in the function *solvePNP* from *OpenCV* library to obtain the corresponding extrinsic parameter matrix.

3.4. Extrinsic with only four points

When four points are selected to determine the camera's extrinsic parameters, only two measurements are taken on a square object, i.e., that has all internal angles with 90^0. Once the measurements (width and height of the object) are extracted, the coordinates of the corners of the object are determined, leaving $A = (0,h,0)$, $B = (0,0,0)$, $C = (l,0,0)$ and $D = (l,h,0)$, whose data will be used in the solvePNP function belonging to the *OpenCV* library to obtain the respective extrinsic parameter matrix.

4. Results

To perform tests on the developed algorithm, an application was developed in Python, supported by the TKInter library. This application incorporates the intrinsic parameters with the checkerboard, the manufacturer's data, and the vanishing points. It also implements the extrinsic parameters, using the points detected from the checkerboard and the manual selection of three or four points from an image. The tool that allows the selection of points manually has the option to zoom in on the image to facilitate the selection of points, two adjustment parameters to sharpen the image (Beta and Gamma), a field that allows the selection of points by entering their coordinates, and a button that allows the acquisition of points and distances between them from a *.yaml* file. This allows the elimination of the human factor at the time of testing.

Finally, in order to be able to test the accuracy of the values, there is the option of drawing the axes of the world on the image and also the option of measuring distances in the chosen XY plane. When the intrinsic parameters are calculated, images are created similar to those used by the algorithm but with the final result represented in them. These images allow to validate the algorithm by correctly detecting and filtering the vanishing points. An image is also built when the extrinsic parameters are determined using the checkerboard, which lets one know if the detected points coincide with those on the checkerboard.

A Logitec C930e camera was used to perform the tests. Several measurements were also made on objects on the ground plane to determine if the values obtained through the calculated matrices vary much from the real measurement. A *.yaml* file was used to make the corresponding measurements on the images. This file lists pairs of coordinates corresponding to the desired points from the measurements. The coordinates of the points are registered, followed by the distance between them in the real world. In the case of obtaining the extrinsic parameters, a *.yaml* file is also used, but unlike the previous one, it contains two lists. One corresponds to the list of points to be used to calculate the extrinsic parameters, and the other for the distances between the points to be selected.

4.1. Intrinsic Parameters

The intrinsic parameters are obtained with the checkerboard. The first tests are carried out in a controlled test environment. First, a validation of the methods for obtaining the intrinsic parameters is the most adequate, along with the evaluation of the distance at which the board is positioned affects the final results. Next, the best intrinsic parameters will be used to determine which of the methods for the extrinsic parameters is best. To cover the largest area of view of the camera, the checkerboard is placed in several different positions, as shown in Fig. 12.

By using this technique to capture images, the algorithm is able to determine the image distortion correctly, thus reducing errors in the measurements.

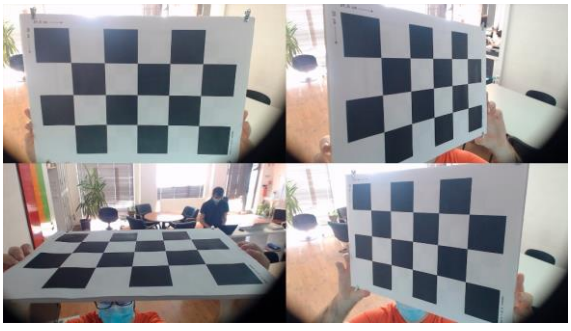

Fig. 12. Some of the images used to calibrate with the checkerboard.

To calculate the intrinsic parameters with the vanishing points, the image represented in Fig. 13 was used. In this image, it is possible to obtain the vanishing points with good accuracy, and to test the algorithm to validate and filter the points. Fig. 14 shows two (*A* and *B*) of the three points chosen to be vanishing points. The coordinates of *C*, the third vanishing point are outside the image boundaries, so it was decided to omit them. In this case, following the procedure mentioned in Sec. 3.2, the coordinates of point *B* are not considered valid for a vanishing point, resulting in its deletion.

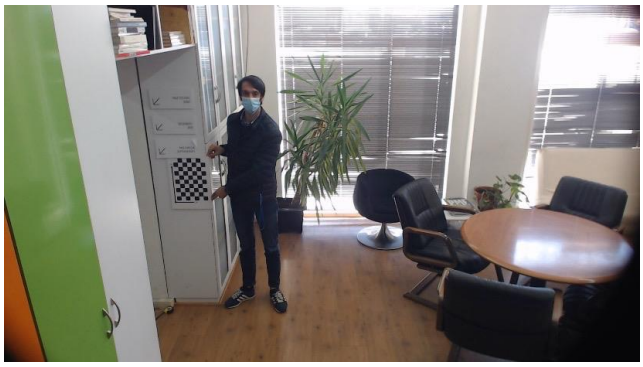

Fig. 13. Captured image for the calculation of the intrinsic parameters with the vanishing points in a controlled context.

Fig. 14. Application of filter to determine vanishing points.

4.2. Controlled Test Scenario

In the controlled test scenario, the board is placed within the camera field of view, at different distances, which are characterized as close (for extrinsic *A*), medium (*B*) or far (*C*). Fig. 15 presents the locations where the metrics were extracted for testing purposes, and Table 1 refers to the extrinsic parameters determined from the intrinsic parameters determined using the vanishing points.

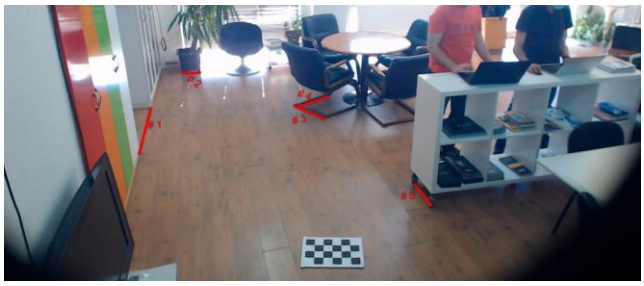

Fig. 15. Representation of the distances using the checkerboard in a near position.

From the points *A*, *B* and *C*, Table 1 calculates the extrinsic parameters. Next, five real-world distances are obtained from the points presented in this table, and the error is determined in Table 2. The percentage of error indicates how badly the distance is being calculated, and we repeated the same steps for the intrinsic parameters with the manufacturer's data and with the checkerboard, obtaining Table 3 and Table 4 respectively. The percentage error is calculated with (7), where the result represents the magnitude of the real difference from the actual distance.

$$\%error = \frac{|D_r - D_c|}{D_r} \quad (7)$$

After determining the percentage error for all the intrinsic parameters, Table 5 represents the average of all the results obtained in each of the previous tables. The results acquired with the vanishing points were able to overcome those obtained with the checkerboard, with a difference of 0.78%. Therefore, the best method to calculate the intrinsic parameters is using the vanishing points.

By analyzing Table 2 and Table 4, it is possible to verify that with the checkerboard, the farther away it is positioned, the better the results are. With the vanishing points, the opposite happened, obtaining better results when the checkerboard is positioned closer. Therefore, it is feasible to conclude that the checkerboard should be placed approximately mid-distance from the camera. This way, neither the worst nor the best results will occur.

4.3. Real Context Test Scenario

Based on the results in the previous section, the best intrinsic parameters are those obtained using the vanishing points. Using the known controlled scenario data of the prior section, the goal will be to determine which methods for calculating the extrinsic parameters produce the best results by considering which has the slightest percentage error in a

Table 1. Data collected for the calculation of the percentage error with the use of vanishing points (distances in *cm*).

Distance	P_1 *(x,y)*	P_2 *(x,y)*	Distance (P_x)	Real distance	Distance with *A*	Distance with *B*	Distance with *C*
1	(409,697)	(443,560)	141.16	119.5	117.35	119.09	117.54
2	(536, 452)	(591, 451)	55.01	26.0	26.27	26.6	26.59
3	(874, 551)	(965, 592)	99.81	55.25	55.54	56.17	55.71
4	(976, 524)	(883, 556)	98.35	55.25	53.38	52.84	51.78
5	(1240, 789)	(1289, 856)	83.01	32.0	33.89	33.79	33.51

real scenario. Two sites were considered for testing. The first, where square shapes are available, and the other where the square shapes are hard to identify.

4.3.1. Site 1

For the first scenario, the following contexts were used to determine the extrinsic parameters, using the data available in the site (Table 6 and Table 7): close (sidewalk, for extrinsic *A*, and road, for extrinsic *B*), and medium (sidewalk wainscot, for extrinsic *C*). The goal is to determine the method to determine the extrinsic parameters that produces best results (with minimum error). Results are presented in Table 8, Table 9, and Table 10 for distances represented in Fig. 16. These tables correspond to the percentage error using the intrinsic parameters obtained with the vanishing points and obtaining the extrinsic parameters with only three, four, or with the automatic detection of the checkerboard points.

Fig. 16. Representation of the distances in the real-context scenario, defined as site 1.

Based on the analysis presented in the tables, Table 11 shows the averages of the results obtained for each of the extrinsic parameters, using as a basis the intrinsic parameters obtained with the vanishing points. From Table 11, it is possible to conclude that the checkerboard is the best tool to calculate the extrinsic parameters.

In the tests using 3 points, a total of 6 triangles were used, of which only 3 are rectangle triangles. In order to check whether non-rectangle triangles are impacting a more significant error in the calculation of the extrinsic parameters, only rectangle triangles were considered in the tests with 3 points, with the average error lowering to 5.41%, which is significantly lower than the 14.35% presented in Table 11. Therefore, by using rectangle triangles, 3 points are the best approach to determine extrinsic parameters.

Table 2. Precentage of error in controlled environment with the use of vanishing points.

Calculated Extrinsic	% Error (#1)	% Error (#2)	% Error (#3)	% Error (#4)	% Error (#5)	% Average error
A	1.80%	1.04%	0.52%	3.38%	5.91%	2.53%
B	0.34%	2.31%	1.67%	4.36%	5.59%	2.85%
C	1.64%	2.27%	0.83%	6.28%	4.72%	3.15%

Table 3. Precentage of error in controlled environment using manufacturer data.

Calculated Extrinsic	% Error (#1)	% Error (#2)	% Error (#3)	% Error (#4)	% Error (#5)	% Average error
A	51.56%	58.92%	47.49%	36.47%	15.19%	41.93%
B	18.37%	28.54%	18.15%	10.79%	0.09%	15.19%
C	9.52%	3.62%	5.63%	7.96%	12.69%	7.88%

Table 4. Precentage of error in controlled environment using checkerboard.

Calculated Extrinsic	% Error (#1)	% Error (#2)	% Error (#3)	% Error (#4)	% Error (#5)	% Average error
A	1.06%	5.88%	1.85%	5.45%	4.97%	3.84%
B	1.14%	5.54%	1.70%	5.74%	4.50%	3.72%
C	0.29%	4.88%	0.58%	7.13%	3.53%	3.28%

Table 5. Average percentage of error in the controlled environment.

Extrinsic	Intrinsic		
	Vanishing points	Manufacturer data	Checkerboard
Checkerboard	0.29%	4.88%	0.58%

Table 6. Points for the calculation of extrinsic parameters using three points in site 1.

Extrinsic	$P_1(x,y)$	$P_2(x,y)$	$P_3(x,y)$	$Dist(P_1,P_2)$	$Dist(P_1,P_3)$	$Dist(P_2,P_3)$	Extrinsic calculated
1	(285, 884)	(881, 921)	(694, 955)	260.225	191.372	99.5	*A*
2	(752, 986)	(47, 1016)	(939, 947)	230.463	99.5	325.725	*B*
3	(285, 884)	(694, 955)	(47, 1016)	191.372	162.42	220.1	*C*
4	(752, 986)	(881, 921)	(939, 947)	107.05	99.5	39.5	*D*
5	(285, 884)	(694, 955)	(430, 1003)	191.37	159	106.5	*E*
6	(285, 884)	(881, 921)	(430, 1003)	260.23	159	206	*F*

Table 7. Points for the calculation of extrinsic parameters using four points in site 1.

Extrinsic	$P_1(x,y)$	$P_2(x,y)$	$P_3(x,y)$	$P_4(x,y)$	$dist(P_1,P_2)$	$dist(P_2,P_3)$	Extrinsic calculated
1	(381, 789)	(543, 775)	(939, 947)	(752, 986)	99.5	318.5	*A*
2	(285, 884)	(514, 857)	(694, 955)	(430, 1003)	106.5	159.0	*B*
3	(694, 955)	(881, 921)	(939, 947)	(752, 986)	99.5	39.5	*C*

Table 8. Percentage of error in site 1, with three points.

Calculated Extrinsic	% Error (#1)	% Error (#2)	% Error (#3)	% Error (#4)	% Error (#5)	% Average error
A	66.83%	28.57%	32.65%	21.51%	65.59%	43.93%
B	23.14%	7.81%	12.38%	9.23%	2.30%	10.97%
C	7.15%	18.30%	10.86%	11.97%	31.11%	15.88%
D	8.46%	9.19%	1.46%	0.91%	10.51%	6.11%
E	5.08%	6.03%	0.62%	3.21%	9.85%	4.96%
F	5.18%	6.59%	0.70%	3.31%	10.08%	5.17%

Table 9. Percentage of error in site 1, with four points.

Calculated Extrinsic	% Error (#1)	% Error (#2)	% Error (#3)	% Error (#4)	% Error (#5)	% Average error
A	15.49%	11.49%	7.27%	5.23%	9.06%	9.71%
B	6.34%	7.38%	0.11%	2.74%	9.72%	5.26%
C	6.57%	9.11%	0.35%	1.82%	12.43%	6.05%

Table 11. Average percentage of error in site 1.

Intrinsic	Extrinsic		
	3 points	4 points	Checkerboard
Vanishing Points	14.35%	7.01%	6.43%

4.3.2. Site 2

Site 2 is selected due to the difficulty to find square shapes to calculate the camera's extrinsic parameters using 4 points. Therefore, a checkerboard and 3 points are used to calculate the extrinsic parameters, characterized as *A* (close), *B* (medium - center of the walk), and *C* (medium - front of cabin). Table 12 presents the points to determine the extrinsic with 3 points, and Table 13 and Table 14 present the results percentage of error using the checkerboard and 3 points (distances represented in the image of Fig. 17).

Table 12. Points for the calculation of extrinsic parameters using three points in site 2.

Extrinsic	$P_1(x,y)$	$P_2(x,y)$	$P_3(x,y)$	$dist(P_1,P_2)$	$dist(P_1,P_3)$	$dist(P_2,P_3)$	Extrinsic calculated
1	(710, 561)	(770, 488)	(1336, 526)	116	298	284	*A*
2	(465, 549)	(770, 488)	(710, 561)	164.05	116	116	*B*
3	(1622, 594)	(1493, 523)	(1473, 523)	100.72	100	12	*C*

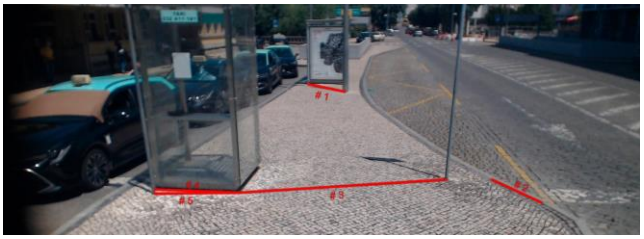

Fig. 17. Representation of the distances in the real-context scenario, defined as site 2.

By considering the average for results presented in Table 13 and Table 14, the 3 points method presents a high percentage error, affected by the extrinsic *A*. As referred in Sec. 4.3.1, this is due to the usage of non-rectangle triangles (illustrated in Fig. 18). By removing the result obtained with the extrinsic *A*, the 3 points method reduces the average percentage of error to 9%.

Table 13. Percentage of error in site 2, using the checkerboard.

Calculated Extrinsic	% Error (#1)	% Error (#2)	% Error (#3)	% Error (#4)	% Error (#5)	Average error
A	5.69%	11.87%	9.55%	3.23%	1.20%	6.31%
B	14.35%	1.19%	2.88%	2.91%	1.64%	4.59%
C	31.32%	8.51%	1.80%	0.01%	0.67%	8.46%

Table 14. Percentage of error in site 2, using three points.

Calculated Extrinsic	% Error (#1)	% Error (#2)	% Error (#3)	% Error (#4)	% Error (#5)	Average error
A	87.71%	203.34%	0.08%	9.16%	1.84%	60.43%
B	6.43%	5.55%	6.12%	0.00%	1.19%	3.86%
C	19.20%	0.00%	1.09%	27.33%	23.11%	14.14%

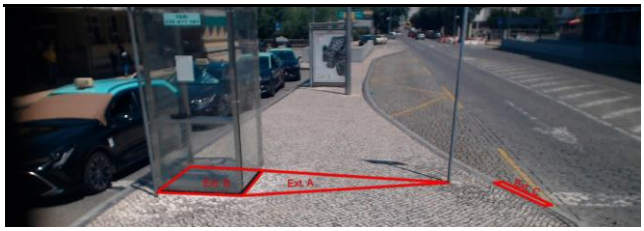

Fig. 18. Representation of the extrisinc used in site 2.

To improve the results, the average percentage error with extrinsic *B* and extrinsic *C* are considered, as well as the object ratio used for the calculation of the extrinsic parameters, given by the quotient between smallest and largest side.

Analyzing Table 16 it is possible to see that the ratio of the object *C* ended up to negatively affect the results. On the other hand, *B*'s ratio equal to 1 allows to conclude that the more square the object selected, the better the result with the use of 3 points.

Table 16. Percentage of error with the vanishing points using 3 points to calculate the extrinsic *B* and *C*, in site 2.

Calculated Extrinsic	%Average error	Object Ratio (0-1)
B	3.86%	1.00
C	14.14%	0.12

4.4. Issues Detected

During the tests, problems were detected with the developed methods, besides the limitation (ratio and format) applied to the objects to calculate the extrinsic parameters with 3 points. It was detected that if the intrinsic parameters are poorly determined, this will affect the calculation of the extrinsic parameters negatively, as was the case with the

manufacturer's data, where poor results were obtained, especially in the image captured on site 2. It was also detected that the checkboard is not always detected, so applying a filter that improves the sharpness of the image becomes essential for its correct detection.

Another problem detected is the objects' quantity or shape in the scenes. Due to the lack of geometric objects in some scenes, the algorithm cannot calculate the intrinsic parameters with the vanishing points or the extrinsic parameters using 4 points.

5. Conclusion

This paper aims to measure the impact of parameters in camera calibration with the slightest possible error. With this, an algorithm was developed that, through a single image, can extract the intrinsic parameters accurately. The algorithm is also able to calculate the extrinsic parameters in several ways, one of the developed ones (using 3 points) being very effective, provided that the object under analysis is a rectangle triangle and has a ratio as close as possible to 1. Using 3 points to determine the extrinsic parameters is an excellent alternative to the checkerboard, making the method more practical since it does not imply the manual positioning of the checkerboard to perform the camera calibration.

In summary, with the developed application, it is possible to obtain better results than the calibration with the checkerboard. However, it is still necessary to capture data from the site to calculate the extrinsic parameters using 3 points, whose coordinates have to be obtained manually with the help of the application. A stereo camera may solve this condition, so it will be possible to perform the measurements using it. Although it may increase the percentage error of the measurements, the stereo camera may make the algorithm more automatic.

Another possibility would be to use YOLO or a similar algorithm, which would be used to detect the objects, and subsequently their edges. This method would allow a filter to be applied to the lines detected by the Hough Transform method, making the developed point filter more effective.

Acknowledgements

This work is funded by National Funds through the FCT – Foundation for Science and Technology, I.P., within the scope of the project Ref. UIDB/05583/2020. Furthermore, we thank the Research Centre in Digital Services (CISeD) and the Instituto Polit\'ecnico de Viseu (IPV) for their support. Moreover, the authors greatly thank IPV, within the scope of the special support project "Don't Stand So Close To Me". Finally, the authors would like to thank Bruno Lopes, José Neves and Marco Bernardo for the data acquisition.

Author contributions

Rui Duarte: Conceptualization, Project Administration, Writing-Original draft preparation, **Carlos Cunha:** Data curation, Writing-Original draft preparation, Validation **José Cardoso:** Visualization, Investigation, Field study, Writing-Reviewing and Editing.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] Y.-J. Zhang, "Camera Calibration," in *3-D Computer Vision: Principles, Algorithms and Applications*, Singapore: Springer Nature Singapore, 2023, pp. 37–65. doi: 10.1007/978-981-19-7580-6_2.

[2] S. Sels, B. Ribbens, S. Vanlanduit, and R. Penne, "Camera Calibration Using Gray Code," *Sensors*, vol. 19, no. 2, 2019, doi: 10.3390/s19020246.

[3] S. Placht *et al.*, "ROCHADE: Robust Checkerboard Advanced Detection for Camera Calibration," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., Cham: Springer International Publishing, 2014, pp. 766–779.

[4] B. Chen, Y. Liu, and C. Xiong, "Automatic Checkerboard Detection for Robust Camera Calibration," in *2021 IEEE International Conference on Multimedia and Expo (ICME)*, 2021, pp. 1–6. doi: 10.1109/ICME51207.2021.9428389.

[5] I. V. Crombrugge, R. Penne, and S. Vanlanduit, "Extrinsic camera calibration for non-overlapping cameras with Gray code projection," *Optics and Lasers in Engineering*, vol. 134, p. 106305, 2020, doi: https://doi.org/10.1016/j.optlaseng.2020.106305.

[6] F. Yang, Y. Zhao, and X. Wang, "Camera Calibration Using Projective Invariants of Sphere Images," *IEEE Access*, vol. 8, pp. 28324–28336, 2020, doi: 10.1109/ACCESS.2020.2972029.

[7] S. J. Lee and S. S. Hwang, "Fast and Accurate Self-calibration Using Vanishing Point Detection in Manmade Environments," *International Journal of Control, Automation and Systems*, vol. 18, no. 10, pp. 2609–2620, Oct. 2020, doi: 10.1007/s12555-019-0284-1.

[8] R. Juarez-Salazar, J. Zheng, and V. H. Diaz-Ramirez, "Distorted pinhole camera modeling and calibration," *Appl. Opt.*, vol. 59, no. 36, pp. 11310–11318, Dec. 2020, doi: 10.1364/AO.412159.

[9] S. Inagaki, A. Sanpei, and H. Himura, "Multiple-pinhole camera for monitoring three-dimensional plasma shape," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators,*

Spectrometers, Detectors and Associated Equipment, vol. 1036, p. 166857, 2022, doi: https://doi.org/10.1016/j.nima.2022.166857.

[10] R. K. Megalingam, V. Shriram, B. Likhith, G. Rajesh, and S. Ghanta, "Monocular distance estimation using pinhole camera approximation to avoid vehicle crash and back-over accidents," in *2016 10th International Conference on Intelligent Systems and Control (ISCO)*, 2016, pp. 1–5. doi: 10.1109/ISCO.2016.7727017.

[11] J. Wang, F. Shi, J. Zhang, and Y. Liu, "A new calibration model of camera lens distortion," *Pattern Recognition*, vol. 41, no. 2, pp. 607–615, 2008, doi: https://doi.org/10.1016/j.patcog.2007.06.012.

[12] A. Langbein, D. A. Plecher, F. Pankratz, C. Eghtebas, F. Palmas, and G. Klinker, "Gamifying Stereo Camera Registration for Augmented Reality," in *2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, 2018, pp. 125–126. doi: 10.1109/ISMAR-Adjunct.2018.00049.

[13] Y. Chen, M. Zhang, P. Lu, X. Zeng, and Y. Wang, "Multi-dimensional Game Interface with Stereo Vision," in *Entertainment Computing - ICEC 2005*, F. Kishino, Y. Kitamura, H. Kato, and N. Nagata, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 368–376.

[14] A. Polanski, K. Wojciechowski, and A. Borek, "Stereo calibration by planar grid lines," in *Computer Analysis of Images and Patterns*, V. Hlaváč and R. Šára, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 1995, pp. 456–463.

[15] P. Rathnayaka, S.-H. Baek, and S.-Y. Park, "An Efficient Calibration Method for a Stereo Camera System with Heterogeneous Lenses Using an Embedded Checkerboard Pattern," *Journal of Sensors*, vol. 2017, p. 6742615, Sep. 2017, doi: 10.1155/2017/6742615.

[16] F. A. Van Den Heuvel, "Vanishing point detection for architectural photogrammetry," *International archives of photogrammetry and remote sensing*, vol. 32, pp. 652–659, 1998.

[17] A. G. Kucukkaya, "Photogrammetry and remote sensing in archeology," *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 88, no. 1, pp. 83–88, 2004, doi: https://doi.org/10.1016/j.jqsrt.2003.12.030.

[18] A. Criminisi, I. Reid, and A. Zisserman, "Single View Metrology," *International Journal of Computer Vision*, vol. 40, no. 2, pp. 123–148, Nov. 2000, doi: 10.1023/A:1026598000963.

[19] J. Canny, "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986, doi: 10.1109/TPAMI.1986.4767851.

[20] P. V. C. Hough, "METHOD AND MEANS FOR RECOGNIZING COMPLEX PATTERNS," Dec. 1962, [Online]. Available: https://www.osti.gov/biblio/4746348

[21] M. A. Krasnosel'skii and A. V. Pokrovskii, *Systems with hysteresis*. Springer Science & Business Media, 2012.

[22] Y. Amit, P. Felzenszwalb, and R. Girshick, "Object Detection," in *Computer Vision: A Reference Guide*, Cham: Springer International Publishing, 2020, pp. 1–9. doi: 10.1007/978-3-030-03243-2_660-1.

[23] D. Wang, M. Moh, and T.-S. Moh, "Using Deep Learning to Solve Google reCAPTCHA v2's Image Challenges," in *2020 14th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, 2020, pp. 1–5. doi: 10.1109/IMCOM48794.2020.9001774.

[24] R. Verschae and J. Ruiz-del-Solar, "Object Detection: Current and Future Directions," *Frontiers in Robotics and AI*, vol. 2, 2015, doi: 10.3389/frobt.2015.00029.

[25] N. Zheng, G. Loizou, X. Jiang, X. Lan, and X. Li, "Computer vision and pattern recognition," *International Journal of Computer Mathematics*, vol. 84, no. 9, pp. 1265–1266, 2007, doi: 10.1080/00207160701303912.

[26] P. Moghadam, J. A. Starzyk, and W. S. Wijesoma, "Fast Vanishing-Point Detection in Unstructured Environments," *IEEE Transactions on Image Processing*, vol. 21, no. 1, pp. 425–430, 2012, doi: 10.1109/TIP.2011.2162422.

[27] Y. Zhang, Y. Su, J. Yang, J. Ponce, and H. Kong, "When Dijkstra Meets Vanishing Point: A Stereo Vision Approach for Road Detection," *IEEE Transactions on Image Processing*, vol. 27, no. 5, pp. 2176–2188, 2018, doi: 10.1109/TIP.2018.2792910.

[28] "Sebastian Zanlongo, Matthew Turk, and Sanjay Parajuli, (2019). Vanishing Point Detection."

[29] B. Caprile and V. Torre, "Using vanishing points for camera calibration," *International Journal of Computer Vision*, vol. 4, no. 2, pp. 127–139, Mar. 1990, doi: 10.1007/BF00127813.