2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

Monocular Vision for 3D Distance Computation in Augmented Reality Applications

Saúl Martínez-Díaz
División de Estudios de Posgrado e Investigación
Tecnológico Nacional de México-Instituto Tecnológico de La Paz
La Paz, México
ORCID: 0000-0003-4962-5995

***Abstract*—Augmented reality is a growing technology with potential applications in education, medicine, entertainment, and tourism, among others. Basically, what this technology seeks is to combine information from the real world with virtual information, without the user perceiving the difference between the two. To achieve this, the augmented reality system must be able to dimension the real-world objects in real time, to generate realistic virtual scenarios. To carry out this dimensioning, a good alternative is to use an artificial vision system that provides a good compromise between cost and performance. In this work a method is presented to calculate the distances among known reference objects in real world and the camera, using a monocular artificial vision system.**

Keywords— monocular vision, augmented reality, visual odometry

I. INTRODUCTION

The term Augmented Reality (AR) refers to a variety of technologies, capable of combining a user's view of the real world with alphanumeric, symbolic, or graphic information. While virtual reality places the user within a completely computer-generated environment, AR focuses on presenting information from the physical world and computer-generated information, so that both appear to be part of the same physical world. In this way, AR must combine the physical with the virtual objects in real time, keeping a record in three dimensions (3D). A fundamental part of an AR system is the generation of digital maps of the user's physical environment. For this, it is necessary to know pose (rotation and translation) of the camera respect to real world objects, in real-world coordinates. Thus, the system will be able to dimension real world objects and distances among them, to insert feasible virtual objects.

Also, in the last few years, computers processing power has increased, and computers cost has dropped. For all the above, it is now possible to process high-definition digital images in real time. Therefore, interest in developing technologies based on image processing and artificial vision has grown. These technologies can be used for navigation of robots [1,2], localization and mapping of unknown places (SLAM) [3,4] virtual (VR), and augmented reality (AR) [5], among others.

Global Positioning Systems (GPS) can be used to determine the location of objects and, therefore, estimate distance among them in coordinates of the real world. Some drawbacks of this technology are: signal interference indoors and large error in distance measurement (several decimeters). Also, equipment provided with visible light and infrared cameras (RGBD) can be used. In this case, drawbacks are the high cost and possible interference on infrared sensors due to sunlight, limiting its use outdoors, with natural light.

It is also possible to use only visible light cameras, which can be used indoor and outdoor, with some advantages such as low cost and low error rate in short distances. The vision system can be configurated to operate with a single camera (monocular vision) or two cameras (stereoscopic vision). In both cases, generally, it is necessary to calibrate each camera to obtain its intrinsic and extrinsic parameters.

In stereo vision one of the cameras is used as reference coordinate system. Pose of the other camera respect to reference camera is obtained by means of a stereoscopic calibration method. Location of three-dimensional objects can be obtained by triangulation of each 3D point matched with the two cameras. However, stereo vision also has some disadvantages.

Some of them are:

- Each camera has a different response to the same light signal making difficult to match points captured with both cameras.
- With two cameras, more physical space, energy, and computational cost is required than with a single camera.
- If stereo calibration is lost due to vibration, triangulation will no longer be reliable.
- The difference between two distant points becomes minimal (they can even be confused with just one).

For the reasons stated above, monocular vision could be a good alternative. Unfortunately, since only one camera is used, it is not possible to use the same triangulation method to compute camera pose. Instead, the same camera is moved to take images in different positions. Overlapping points in different images can be used to estimate the change of pose of the camera. The new pose can only be determined up to an unknown scale factor. The determination of this scale factor is fundamental for this type of applications.

In this paper, we propose a method to compute the distance of a calibrated camera to objects of known dimensions, in real three-dimensional world coordinates, for augmented reality applications. This distance calculation method is less

restrictive than other methods proposed in the literature. For this, distance of at least three points of the reference object must be known. Unlike other approaches, the proposed method does not require additional sensors or the use of heuristic techniques to determine the distance. By determining the distance of the camera to known objects, it is possible to insert new virtual objects of the appropriate size. If none of the known objects are detected in a frame, the location is estimated using a keyframe-based algorithm. The accumulated error of said estimate can be corrected when one of the reference objects is found again.

II. RELATED WORK

In the literature, two approaches have been shown to be successful for pose estimation with monocular systems: filtering methods [6,7] and keyframe-based methods [8,9]. In the paper of Strasdat et al [10] is showed that keyframe-based techniques can be more accurate than filtering methods, with a similar computational cost.

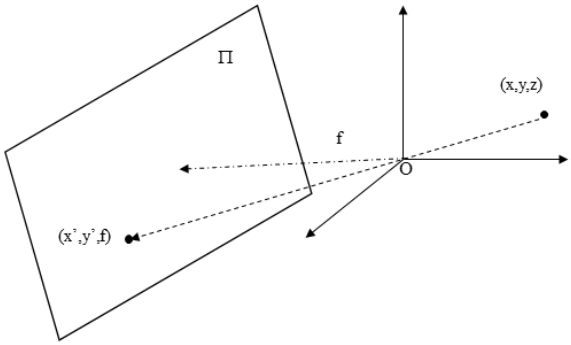
In reference [3], the authors proposed a SLAM algorithm with monocular techniques that exploits the stiffness constraints of objects, to find their true scale. They used a binary descriptor based on the Oriented FAST and Rotated BRIEF (ORB) functions. Once an object is observed at various camera positions, the scene features obtained from consecutive images are triangulated, allowing an estimation of the object's location in three dimensions. Furthermore, in reference [11], the authors proposed a support vector machine (SVM) combined with a bag of binary words and a Robust Accelerated Feature (SURF) descriptor, for the recognition stage. However, the last two approaches require a large amount of memory to maintain previous location maps and the dictionary for binary word bags; Unfortunately, memory and other hardware resources are limited on many AR systems.

Another major issue of monocular vision is determination of the known scale factor. This factor should be estimated using some initial fit technique. Several methods have been proposed to compute it: in [12], an inertial measurement unit (IMU) is used as an auxiliar sensor to determine the scale. In [13], authors use a convolutional neural network to estimate depth at each frame; however, many frames are required to finally reduce the estimation error. In reference [14] authors suppose that the field of view of the camera is always perpendicular to the ground; under such supposition, the camera is positioned at a known distance from a person. Then, the number of pixels detected on the face of the person is computed; Finally, a ratio of pixels and distance is established; such ratio allows to estimate the distance of the same person in other images, even if they are taken from different distances from the camera.

III. BASICS

A. Camera Model

the simplest way to represent the operation of a camera is using the pinhole model. This model supposes that a beam of light enters the by the pinhole and is projected onto the camera formation image plane. Figure 1 shows this principle. In the figure, a point P (x, y, z) in real world is projected to the point P' (x', y ', f) of the camera plane (Π) in the chamber. Here, f represents the perpendicular distance from the pinhole to the camera's plane (focal distance). From figure 1, we can obtain the following equalities [15]:



$$\frac{x'}{x} = \frac{y'}{y} = \frac{f}{z} = \lambda \quad (1)$$

In this case, λ is the scale factor. When f and λ are known, we can compute the real-world coordinates of P, from the pixels coordinates of the point projected on the image plane. Usually, the focal length and other intrinsic and extrinsic parameters can be obtained in the process of camera calibration.

Fig. 1. Representation of camera (pinhole model).

B. Camera Calibration

Calibration provides a model of camera's geometry. This information is used to define its intrinsic and extrinsic parameters. Let P' be the projection on the plane of the camera of the three-dimensional point P. Using homogeneous coordinates, we can define $P = [X\ Y\ Z\ 1]^T$ and $P' = [x\ y\ 1]^T$. In matrix notation, the mapping from P to P' can be expressed as

$$\lambda P' = A[R\ t]P \quad (2)$$

Where [R t] is the augmented matrix of extrinsic parameters. This augmented matrix includes rotation matrix R and translation vector t. Besides, A is the matrix of intrinsic parameters, defined by:

$$A = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

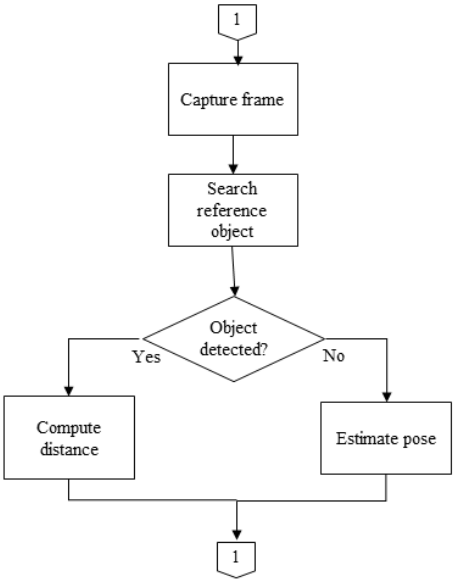


Fig. 2. General proposed method.

Here, f_x, f_y give information (depending on the pixel size) of the focal length in the x and y direction, respectively; c_x and c_y are the coordinates of the principal point of the image; s is known as skew and represents the angle of inclination of the pixel.

Zhang [16] proposed a calibration technique based on the observation, from various positions, of a flat, checkerboard-like pattern. The advantage of this calibration method is that it allows the camera parameters to be easily obtained from the reference system, solving a system of equations, without the need to know the position of the reference points or the camera.

IV. METODOLOGY

Figure 2 shows the general operation of the proposed method. The system captures each image and searches for the reference object. Reference object detection is based on corner finding. Due to corners are invariant to translation, rotation, and illumination, they are robust features to detect the objects of interest. In this work the method proposed by Harris [17] was used. The main idea of Harris algorithm is to search for strong derivatives in two orthogonal directions of the image. If the reference object is found, its distance is calculated using the method described in the next section; otherwise, the distance is estimated with a keyframe-based algorithm. The basic keyframe-based algorithm used is:

- Extract features of interest from each image using the SURF algorithm [18].
- Match features between consecutive images.
- Triangulate between the coincident points applying the calibration parameters and estimate the distance.
- Optimize estimation with algorithm proposed in reference [19].

A. Distance Calculation

From equation (1) the following relationships can be set:

$$x = \frac{x'}{f} z \qquad y = \frac{y'}{f} z \tag{4}$$

Now, suppose that in 3D space there are three points $P_1(x_1, y_1, z_1)$, $P_2(x_2, y_2, z_2)$, and $P_3(x_3, y_3, z_3)$. Also assume that d_1, d_2, and d_3 are the distances between $P_1 - P_2$, $P_1 - P_3$ and $P_3 - P_2$, respectively. Using the equalities in (4), computation of the quadratic Euclidean distance between pairs of points can be stated as:

$$d_1 = \left(\frac{x'_1}{f} z_1 - \frac{x'_2}{f} z_2\right)^2 + \left(\frac{y'_1}{f} z_1 - \frac{y'_2}{f} z_2\right)^2 + (z_1 - z_2)^2$$
$$d_2 = \left(\frac{x'_1}{f} z_1 - \frac{x'_3}{f} z_3\right)^2 + \left(\frac{y'_1}{f} z_1 - \frac{y'_3}{f} z_3\right)^2 + (z_1 - z_3)^2$$
$$d_3 = \left(\frac{x'_3}{f} z_3 - \frac{x'_2}{f} z_2\right)^2 + \left(\frac{y'_3}{f} z_3 - \frac{y'_2}{f} z_2\right)^2 + (z_3 - z_2)^2 \tag{5}$$

This system of equations (5) can be solved with any iterative method to obtain the unknown coordinates z_i, in real-world units. With these coordinates it is possible to calculate the Euclidean distance from the camera to each of these points, in three-dimensional space. Furthermore, if two consecutive images are taken with the same camera in different positions, applying the proposed method, it is possible to estimate the relative displacement (rotation and translation) between both positions.

Fig. 3. Example of images used to compute distance.

V. RESULTS

A. Initial Setup

In this section, we illustrate the performance of the proposed method (depth calculation and frame-based estimation) by means of computer simulations.

The distance calculation was tested using our own images acquired with a low-cost webcam. The results were compared with a Kinect device. Kinect is an RGBD system that provides the 3D coordinates of points in the real world. It contains a visible light camera with a resolution of 1920x1080x3 and an infrared light system to provide depth information. To test the proposed algorithm, a set of 150 images were taken from different distances. Each image containing a chessboard as reference object, as shown in figure 3. Each square of the chessboard is 27x27 millimeters long. RGB images were acquired with low-cost Microsoft USB webcam at a resolution of 640x480x3 pixels. The light source provided inhomogeneous illumination from a set of alternating current lamps.

To test the frame-based distance estimation of the system, 600 images from the 00 sequence of the KITTI dataset [20] were used. This data set contains images acquired with PointGray Flea2 grayscale cameras. Each grayscale image has a resolution of 376x1241 pixels. The data set includes real references obtained with a high precision GPS/IMU inertial navigation system. Figure 4 is an example of images used, which were acquired in a residential setting.

Fig. 4. Example of a image of KITTI dataser.

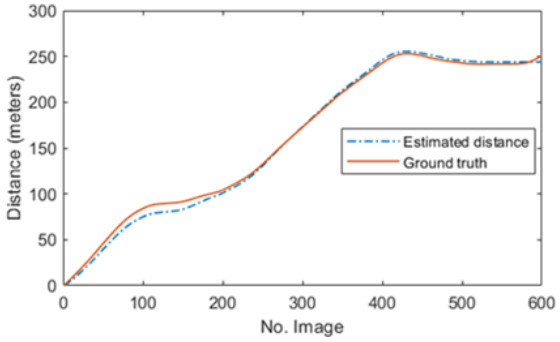


Fig. 5. Comparison of keyframe-based estimated distance and groundtruth.

B. Distance Computation

At this stage, the camera was first placed at different distances from the pattern. Then, the distance was calculated with the proposed algorithm and compared with the readings obtained from the Kinect sensor. Table I shows a part of the results obtained. From the data collected, a correlation of 98.4163% is obtained between the Kinect distances and those measured with our method, with a 9% average percentage error. When objects are three meters (or more) apart from the camera, results are less reliable.

TABLE I. COMPARISON OF RESULTS

Distance (cm) with proposed method	Distance (cm) with Kinect
97.93	97.5
105.08	105
112.29	112.5
120.14	120
127.9	127.5
135.45	135
141.8	142.5
148.94	150

C. Distance Estimation

Because the frame-based algorithm requires initial scaling, such scaling was performed using the first two rows of the ground-truth file provided with the dataset. From each image we detected and matched the SURF features. Subsequently, triangulation and distance estimation were carried out. These last results were compared with the reference file included in the data set. Figure (5) shows the results of the comparison between real and estimated data. A correlation of 0.9996 was obtained between both graphs with a 5.88% average percentage error.

When the reference object is not detected for a long time (more than 600 frames), the frame-based estimation error increases considerably. Said error due to drift can be corrected when the reference object is detected again. Another alternative is to use various reference objects that can be found in the scene; this involves using a robust multi-object detection technique, such as a convolutional neural network.

VI. CONCLUSIONS AND FUTURE WORK

In this article, a method to calculate the three-dimensional distance between a known object and the camera was presented. The system uses monocular vision for application in augmented reality. However, this method can also be used for applications in industrial control systems, SLAM or autonomous robot navigation, using a single camera. Points can be obtained from any known reference object. If the

reference object is not detected at any time, the distance can still be estimated using a keyframe-based algorithm.

The experimental results were promising, even using low-quality images taken with a webcam, under uncontrolled illumination conditions.

Future work includes the design of robust multiclass classifiers, such as convolutional neural networks, to recognize common objects expected to be found in the environment. Besides, the development of specialized parallel processing hardware, to reduce the time of some algorithms such as SURF, matching, and pose estimation.

ACKNOWLEDGMENT

This work was supported by Tecnológico Nacional de México-Instituto Tecnológico de La Paz.

REFERENCES

[1] S. Feng, Z. Wu, Y. Zhao and P. A. Vela, "Image-based trajectory tracking through unknown environments without absolute positioning," in IEEE/ASME Transactions on Mechatronics, vol. 27, no. 4, pp. 2098-2106, Aug. 2022, doi: 10.1109/TMECH.2022.3175819.

[2] T. Alves, T. Hormigo and R. Ventura, "Vision-based navigation solution for autonomous underwater vehicles," 2022 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), 2022, pp. 226-231, doi: 10.1109/ICARSC55462.2022.9784778.

[3] R. Mur-Artal and J. D. Tardós, "Visual-inertial monocular SLAM with map reuse," in IEEE Robotics and Automation Letters, vol. 2, no. 2, pp. 796-803, April 2017, doi: 10.1109/LRA.2017.2653359.

[4] H. Bavle, P. De La Puente, J. P. How and P. Campoy, "VPS-SLAM: visual planar semantic SLAM for aerial robotic systems," in IEEE Access, vol. 8, pp. 60704-60718, 2020, doi: 10.1109/ACCESS.2020.2983121.

[5] D. Chatzopoulos, C. Bermejo, Z. Huang and P. Hui, "Mobile augmented reality survey: from where we are to where we go," in IEEE Access, vol. 5, pp. 6917-6950, 2017, doi: 10.1109/ACCESS.2017.2698164.

[6] J. Civera, A. J. Davison and J. M. M. Montiel, "Inverse depth parametrization for monocular SLAM," in IEEE Transactions on Robotics, vol. 24, no. 5, pp. 932-945, Oct. 2008, doi: 10.1109/TRO.2008.2003276.

[7] J. Zhang, G. Zeng and H. Zha, "Scalable monocular SLAM by fusing and connecting line segments with inverse depth filter," 2018 24th International Conference on Pattern Recognition (ICPR), 2018, pp. 2283-2288, doi: 10.1109/ICPR.2018.8546049.

[8] R. Mur-Artal, J. M. M. Montiel and J. D. Tardós, "ORB-SLAM: a versatile and accurate monocular SLAM system," in IEEE Transactions on Robotics, vol. 31, no. 5, pp. 1147-1163, Oct. 2015, doi: 10.1109/TRO.2015.2463671.

[9] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," in Int. J. Robot. Res. Vol. 34, no. 3, pp. 314–334, 2015, doi: 10.1177/0278364914554813.

[10] H. Strasdat, J.M.M. Montiel and A.J. Davison, "Visual SLAM: why filter?," in Image and Vision Computing, vol. 30 no. 2, pp. 65–77, 2012, doi:10.1016/j.imavis.2012.02.009.

[11] A. J. Siddiqui, A. Mammeri and A. Boukerche, "Real-time vehicle make and model recognition based on a bag of SURF features," in IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 11, pp. 3205-3219, Nov. 2016, doi: 10.1109/TITS.2016.2545640.

[12] J. J. Tarrio and S. Pedre, "Realtime edge based visual inertial odometry for MAV teleoperation in indoor environments," in J. Intell. Robot Syst. Vol. 90, pp. 235-252, 2018, doi:10.1007/s10846-017-0670-y.

[13] X. Yin, X. Wang, X. Du and Q. Chen, "Scale recovery for monocular visual odometry using depth estimated with deep convolutional neural fields," 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 5871-5879, doi: 10.1109/ICCV.2017.625.

[14] N. Yao, E. Anaya, Q. Tao, S. Cho, H. Zheng and F. Zhang, "Monocular vision-based human following on miniature robotic blimp," 2017 IEEE International Conference on Robotics and Automation (ICRA), 2017, pp. 3244-3249, doi: 10.1109/ICRA.2017.7989369.

[15] A. I. Barranco-Gutiérrez, S. Martínez-Díaz and J. L. Gómez-Torres, "Visión estereoscópica con MATLAB y OPENCV," Pearson, 2018.

[16] Z. Zhang, "A flexible new technique for camera calibration," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 11, pp. 1330-1334, Nov. 2000, doi: 10.1109/34.888718.

[17] C. Harris, and M. Stephens, "A combined corner and edge detector," Proceedings of the 4th Alvey Vision Conference, 1988, pp. 147–151, doi: 10.5244/C.2.23.

[18] H. Bay, A. Ess, T. Tuytelaars and L. Van Gool, "SURF:Speeded up robust features," in Computer Vision and Image Understanding vol. 110 no. 3, pp. 346–359, 2008, doi:10.1016/j.cviu.2007.09.014.

[19] B. Triggs, P. F. McLauchlan, R.I. Hartley and A. W. Fitzgibbon, "Bundle adjustment a modern synthesis," International Workshop on Vision Algorithms, 2000, vol.1883, pp. 298–372, doi: 10.1007/3-540-44480-7_21.

[20] A. Geiger, P. Lenz, C. Stiller and R. Urtasun, "Vision meets robotics: the KITTI dataset," in The International Journal of Robotics, vol. 32 no. 11, pp. 1231–1237, 2013, doi: 10.1177/0278364913491297.