A TRAFFIC SIGN DETECTOR FOR MOBILE ROBOT LOCALIZATION

Marco A. G. Moreira*, Henrique N. Machado*, Cristina F. C. Mendonça*,
Guilherme A. S. Pereira*

*Departamento de Engenharia Elétrica, Universidade Federal de Minas Gerais,
Av. Antônio Carlos, 6626, 31270-901, Belo Horizonte, MG, Brasil.

Emails: lelinhosgp@yahoo.com.br, henriquebhbr@gmail.com, crisfcm@yahoo.com.br,
gpereira@ufmg.br

Abstract— This paper describes a visual sensor that is able to localize a mobile robot relatively to planar marks. In our application, the planar marks are traffic signs, very common in urban scenarios. As an application for the sensor, we also propose an approach based on Particle Filters for combining the information from the visual sensor and the robot odometry. Experimental results with an actual mobile robot are presented.

Keywords— robot localization, landmark detector, perspective three-point problem.

1 Introduction

There are many applications for mobile robots in outdoor urban environments which demand precise robot localization. Since such areas are populated with traffic signs with standard dimensions, using them for localization is a natural approach.

Traffic sign detection is usually performed using a combination of color and shape information, as in (Bahlmann et al., 2005). Miura et al. (2000) present an active vision system, composed of two cameras. The system first detects candidates for traffic signs in a wide-angle image and then a telephotocamera is directed to each candidate position and capture it in a larger resolution. In this paper we present a traffic sign detector which provides absolute 3D localization information that can be combined with relative sensors in order to obtain precise localization. The main contribution of this work, when comparing with those above, is to use a single camera to perform planar mark detection and 3D localization.

The main idea of most methods for mobile robot outdoor localization is to combine relative sensors measures, such as the ones obtained from odometry and inertial sensors, and absolute information. Nowadays, most of these methods use GPS (Global Positioning System) sensors to provide absolute information. However, the GPS has some limitations for ground robots localization in urban scenarios. For example, it can be observed that weather condition, trees and buildings can obstruct the received signal from satellites, affecting its performance (Thrapp et al., 2001).

To improve GPS based localization, (Panzieri et al., 2002) and (Agrawal and Konolige, 2006) combine such information with the one provided by landmark detectors. Landmarks were also used in (Maeyama et al., 1994). In that paper, besides odometry, the authors use sonar and visual sensors in order to detect trees. Trees were also used as landmarks in (Adams et al., 2004), where a laser range finder was used as the main sensor.

To show the applicability of the traffic sign detector, we combine it with two types of odometry (traditional and visual based) to localize a mobile robot in an outdoor environment. A detailed description of the problem we are considering is given in the next section.

2 Problem Definition

Consider a wheeled mobile robot equipped with encoders and a visual sensor operating in an outdoor environment. The environment is populated with several traffic signs with previously known positions and orientations. The traffic signs, acting as planar visual marks (or beacons), are identical and cannot be uniquely identified. Our problem consists in estimating the robot's pose (position and orientation) in the environment by combining odometry information with information obtained from the camera and beacon positions.

A standard solution for this problem is to predict the robot position using odometry and correct the current estimate with absolute pose obtained from the visual mark positions. However, here, we have to deal with an extra problem: we use a single camera to estimate the relative pose between the robot and the planar marks. This results in up to four possible solutions to the problem, from which we have to choose the correct one during the localization process.

This paper describes the traffic sign based localization system, which uses standard computer vision techniques for beacon detection an relative pose estimation, and a particle filter for sensor integration. Although the complete robot localization methodology is presented here, this paper focuses on the computer vision part of it. A deeper discussion of the sensor fusion approach can be found in (Moreira et al., 2007)

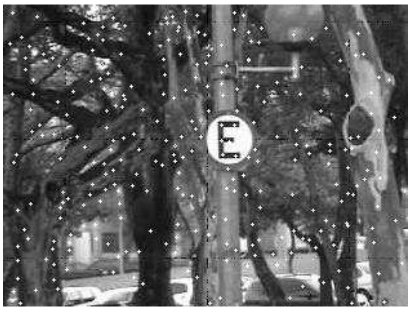

Figure 1: SIFT keypoints (small white marks) distribution over a typical outdoor image.

3 Planar Signs based Localization

This section describes the identification and localization of planar visual marks used in our localization algorithm. As visual marks we use traffic signs. More specifically, “Parking Allowed” traffic signs, which are very common in the streets of our university campus. They are also very simple: a letter E inside a red circle. The next subsections describe how we identify the marks in the image and how the pose between the mark and the camera is computed.

3.1 Beacon identification

To identify the E sign we use SIFT (Scale Invariant Feature Transform) (Lowe., 2004). This technique transforms stable keypoints in a grayscale image into feature vectors, each of which is invariant to image translation, scaling, and rotation, and partially invariant to affine projection and illumination changes.

The keypoints identification is the first step of the process. They are chosen based on their stability over scale and orientation of the image, i.e., over different views of the scene. It is implemented via a difference-of-Gaussian function. Typically, in an image of size 640x480 pixels, there are a few hundreds of potential interest points. Fig. 1 shows the distribution of keypoints on a sample image containing a traffic sign.

It could be noticed that the center of the E is almost always chosen as a stable point, regardless of scale and view point. So this point was chosen to be the feature that indicates the possibility of presence of a traffic sign in the image.

The goal of our system is to identify whether a given feature vector belongs to the center of the traffic sign or not. In order to do that, we use a simple perceptron node, which is trained with samples of feature vectors from points of the E center and from random positions other than that.

Once the center of the E has been identified — point p'_c in Fig. 2 —, it is easy to identify the corners of the letter. First, we get the grayscale level, g_L, of the image pixel at p'_c and binarize the image using a threshold value th proportional to that, $th = \alpha g_L$. From p'_c we use standard blob coloring methodologies to obtain the blob region which forms the E (see Fig. 3). The corners of the right side of the letter — p'_1 and p'_3 — will be the points from the blob that are farthest from the point p'_c. The corners of the left side — p'_0 and p'_2 —, the farthest ones from p'_1 and p'_3.

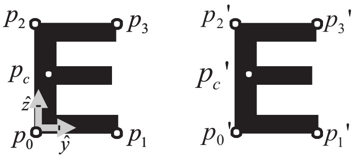

Figure 2: Definition of the corners of the traffic sign. Left: real corners and reference frame attached to the beacon. Right: projection of the corners onto the image plane.

To check if the sign has been correctly identified, we resample the region determined by points p'_0, p'_1, p'_2 and p'_3 in a 64x64 image. Then, we perform a normalized cross correlation operation between the obtained image and a standard image containing the traffic sign. If the result of the operation is above a certain threshold, we can assure that the identified points actually belong to the traffic sign.

The resampling task is performed on the following way. We define 64 equally spaced points between p'_1 and p'_3 — right side — and between p'_0 and p'_2 — left side. Then 64 horizontal line segments linking the corresponding points in the two sides are traced. The same operation is done with the upper and bottom sides of the examined traffic sign, obtaining 64 vertical lines. The 64x64 image is obtained by the intersection of the horizontal and vertical line segments. For reducing the discretization errors, bicubic interpolation is used on the resampling process.

It is not exactly true that equally spaced points in the image are related to equally spaced points in the original scene. However, this approximation can be made when the distance between object and camera is large when comparing with the object dimensions. This is very often true on the traffic sign identification process. Once the identification of the points from p'_0 to p'_3 does not depend on the object orientation, we may say that this resampling process is invariant to translation, scale and orientation (translation invariance is obtained when we identify the center of the E via SIFT). This pattern recognition methodology has advantages over standard ones found in the literature based on Fourier and Mellin transforms (Belo, 2006), which use the modulus invariance of the transforms under translation and scale, respectively, in combination with resampling methods, as the Log-Log and the Log-Polar

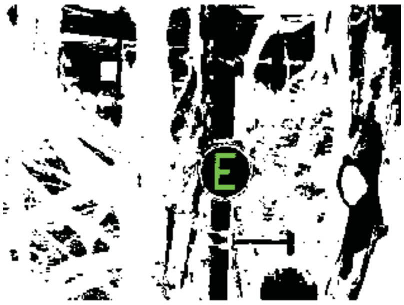

Figure 3: Identification of the main blob.

transform. It is known that most of the signal content is present on the phase of the transform. Thus, using only the modulus implies loosing information, which can lead to misrecognition.

Once the corners have been identified, it is possible to construct an equation system which lead us to the beacon localization. This is shown next.

3.2 Sign Localization

Using the identified corners, the localization of the landmark can be performed through the perspective three-point problem. This problem was solved in many ways and can be summarized as follows (Wolf et al., 1991): "from a single perspective view of the vertices of a known triangle, determine all the possible camera-triangle configurations". Figure 4 illustrates the geometry of the problem. Using the points p'_0, p'_1 and p'_2 and the camera intrinsic parameters, it is possible to find three unit vectors in the camera reference frame — $\hat{u}_0$, $\hat{u}_1$ and $\hat{u}_2$— pointing in the direction of the vertices of the triangle. From the cosine law, the unknown parameters — s_0, s_1 and s_2 — can be related with the sides of the triangle as follows:

$$\begin{aligned} s_0^2 + s_1^2 &= D_1^2 - 2s_0s_1\cos\theta_{01} \\ s_0^2 + s_2^2 &= D_2^2 - 2s_0s_2\cos\theta_{02} \\ s_1^2 + s_2^2 &= D_3^2 - 2s_1s_2\cos\theta_{12} \end{aligned}$$

where θ_{ij} is the angle between vectors $\hat{u}_i$ and $\hat{u}_j$.

This set of equations can produce up to four physically possible solutions. However, as was pointed out in (Wolf et al., 1991), the solution set usually consists of only two configurations. In this work, we use the approach suggested by (Fischler and Bolles, 1981) to compute the problem solutions via a quartic polynomial equation.

The use of the perspective three-point problem for mobile robot localization is also proposed in (Moreira and Pereira, 2006). However, there, only one of the solutions is picked, based on spatial restrictions on the robot localization. Here, as will be seen in the next section, all the solutions of the problem are used in a sensor fusion process. A particle filter naturally selects the correct one.

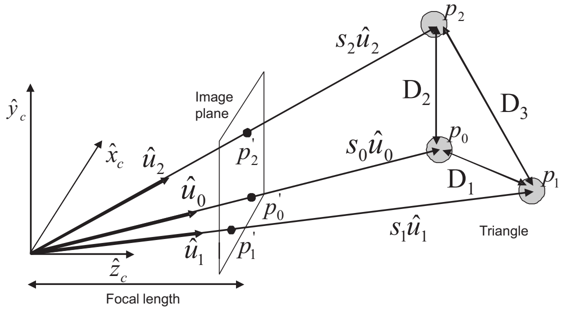

Figure 4: Perspective Three Point Problem.

4 Particle Filter

As mentioned before, in this work a Particle Filter (Thrun et al., 2005) is used to combine odometry and beacon localization. The main reason for using this filter instead of other techniques, such as the Kalman Filter, is its ability to deal with multimodal distributions. In our problem, multimodal distribution appears not only due to the fact that we have several identical beacons and the robot is not able to distinguish which one is in its field of view, but also because of the multiple solutions given by the planar beacon localization.

The Particle Filter was addressed in many works (Reckleitis, 2003) and, basically, it can be described as follows. Several particles, each one representing an instance of the robot pose, are distributed around the actual robot position. Once the robot starts moving, the new position of the particles are estimated using a model for the robot and the actual measurements of the robot velocities, which are used as inputs for the model. A random noise is added to the measurements to represent the sensor error. This is known as the prediction step. In the second step of the filter, known as correction step, the particles with low probability of being in the position given by an absolute sensor are eliminated, while the ones with high probabilities are replicated. The estimate of the robot pose is given by a function of the particles distribution. The approaches used in this paper for prediction, correction and final pose estimation are described next.

4.1 Prediction

In the prediction step, the robot pose, $p = (x, y, \theta)$, is computed using the kinematic model represented by Equations (1), where v_k e ω_k are the system inputs at instant k — linear velocity and angular velocity, respectively —, and Δt is the time interval between instants k and $k+1$.

$$\begin{aligned} x_{k+1} &= x_k + v_k.\Delta t.cos(\theta_k) \\ y_{k+1} &= y_k + v_k.\Delta t.sin(\theta_k) \\ \theta_{k+1} &= \theta_k + \omega_k.\Delta t \end{aligned} \quad (1)$$

The inputs are taken both from wheel odometry and also from the visual odometry sensor proposed in (Machado and Pereira, 2006). In one hand, since the wheel rotation is measured by wheel odometry in spite of the actual robot displacement, this sensor is responsible for large accumulation errors in the prediction step, mainly due to wheel slippage (Thrapp et al., 2001).

On the other hand, the visual odometry sensor, being a device connected to the robot, is immune to slippage but is contaminated by the noise caused by the vertical movement of the robot (see (Machado and Pereira, 2006)).

Considering the particularities of these two sensors, we decided to use most of time the linear and angular velocities measured by the wheel odometry sensors. However, if a slippage situation is detected, the measurements will be taken from the visual odometry sensor. Details on the combination of these two kinds of odometry are presented in (Moreira et al., 2007).

For each particle a random number is added to the variables which work as inputs to the model, v and ω. The variances of the random numbers are chosen so that they represent the precision of the sensor. Also, to each particle i is associated a weight w_i that represents the quality of the particle. In the first interaction of the filter, all particles start in the robot initial position and have the same weight, which will be updated in the correction step.

4.2 Correction

The correction step occurs every time a traffic sign is identified. The main problem is that it is not known which of the landmarks was actually identified and, even, which of the possible solutions associated with the beacon is the correct one. To take this uncertainty into account, we use the solution which better matches with the predicted pose of the considered particle, as proposed in (Adams et al., 2004). Then the weight w of each particle i is updated according to:

$$w^i_{k+1} = w^i_k \max_{j=1,\ldots,N} f(p^j_e, p^i_{k+1}), \qquad (2)$$

where f is a function of the predicted pose p^i_{k+1} of the i-th particle and of the estimated robot pose p^j_e supposing that the j-th solution is the correct one.

In our specific case, it was observed that the visual sensor may output accurate distance estimates. However, the same cannot be said about orientation. Thus, by using the polar coordinates r and θ and a heading value φ we will represent the uncertainty model of the sensor in a more faithful way than using cartesian coordinates. Thus, considering the pose in the correction step as shown in Fig. 5, function f in Equation 2 was defined as:

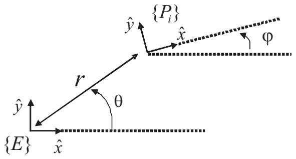

Figure 5: Estimation of particle's pose. $\{E\}$ is the reference frame of a beacon, as shown in Fig. 2, and $\{P_i\}$ is the reference frame of a particle i.

$$f(p^j_e, p^i) = \frac{e^{\frac{-(r^j_e - r_i)^2}{2\sigma_r}}}{\sqrt{2\pi}\sigma_r} \cdot \frac{e^{\frac{-(\theta^j_e - \theta_i)^2}{2\sigma_\theta}}}{\sqrt{2\pi}\sigma_\theta} \cdot \frac{e^{\frac{-(\varphi^j_e - \varphi_i)^2}{2\sigma_\varphi}}}{\sqrt{2\pi}\sigma_\varphi} .$$

As the robot moves, the set of particles are spread over the environment and some of them drift far enough so it do not represent the uncertainty associated with the robot movement anymore. Thus, it is necessary to resample in such a way that only the particles with considerable probability are preserved. As suggested in (Reckleitis, 2003), to monitor the representability of the particles we use the effective sample size. When the effective sample size drops below a threshold, the set of particles is resampled. On the resampling process, the simple technique of selecting with replacement is used. Details on this technique can be found in (Reckleitis, 2003), what by April of 2007 was available on-line.

5 EXPERIMENTS

This section shows experiments that validates the proposed methodology. We start by describing our experimental setup.

5.1 Experimental Setup

The experiments were realized with a P3-AT robot by ActivMedia Robotics. The robot was equipped with odometry and two IEEE1394 cameras. The robot was controlled using a joystick plugged in a laptop running Microsoft Windows XP (see Fig. 6). Each of the sensors were executed in a individual thread. The information acquired/computed by the threads were timestamped and asynchronously sent to a server, which was responsible for centralizing and storing the data to be processed off-line.

The experiments took place in an outdoor environment at the UFMG campus, in an concreted plane area of about $400m^2$. Twelve control points, distanced about 5.0 meters from each other, were marked and mapped. In order to validate the localization results, during the test execution, a joystick button was pressed every time the robot reached one of these points. Five "Parking Allowed" traffic signs were inserted on the environment and mapped using the visual system itself.

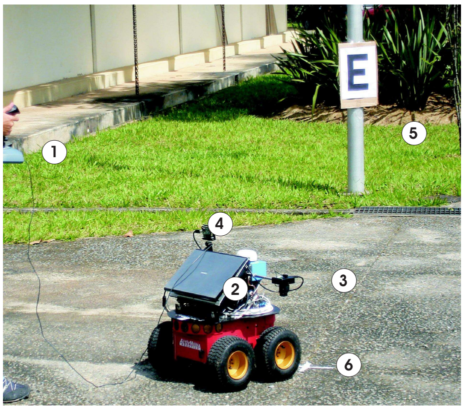

Figure 6: Experimental setup: 1 – joystick for robot control, 2 – laptop, 3 – camera for visual odometry, 4 – camera for beacon identification, 5 – traffic sign (beacon), 6 – a control point used for validation purposes only.

5.2 Results

In the description of our experiment, first consider the control points represented by the grid intersections in Fig. 7. They are labeled from A to L. During the experiment presented in this paper, the robot went through the control points on the following order: G, C, D, E, J, K, L, I, A, B, H, G. In this experiment the robot moved with linear velocities that varied from 0 to $0.6m/s$. Fig. 7 shows the path obtained using the combination of wheel and visual odometry and the complete filter (prediction and correction). The small circles and squares over the paths represent the exact moment the actual robot was over the control points, as perceived by the human operator. The beacons location are represented by the $\hat{x}$ and $\hat{y}$ axis of their reference frames $\{E_i\}$'s (the $\hat{z}$ axis points out of the page). The black marks on the particle filter path represent the moments the robot detects a beacon.

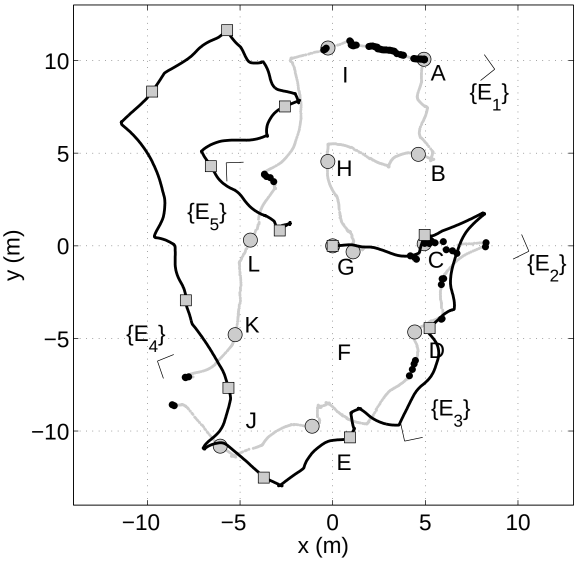

Figure 7: Robot path estimated by the odometry combination (black) and using the complete particle filter (gray). The black marks on the particle filter path represent the moment the robot has detected a traffic sign. The $\{E_i\}$'s are the reference frames of the signs. Letters from A to L label the control points used for validation. The gray circles and squares represent the moments the robot should be over a control point.

Observe in Fig.7 that the robot rarely detects a beacon. Therefore, most of the time the pose estimate only relies on odometry. In spite of this, it can be noticed that the absolute information from the traffic signs identification can drastically reduce the localization error. Notice in Fig. 7 that the particle filter allows the pose to "jump" to a better estimate as soon as it detects (by the correction step) that the current estimate is wrong. The reduction of the error after a beacon is detected is better observed in Fig. 8, which presents the localization error over the time. In this figure, the error value is computed by the distance between the control point and the robot pose in the instant it should be over the control point. The gray values represent the error obtained by the free prediction using the two odometry sensors and the black values represent the filter error. The small marks on the time axis represent the time the robot detected a beacon. Observe that right after a beacon detection, the pose estimate error is reduced.

6 CONCLUSIONS

This paper presented a visual sensor that can be used to improve robot localization in urban environments, which are populated by traffic signs. By using information from this sensor in a particle filter correction step, we were able to obtain good results, even in an outdoor environment, where the robot is subjected to wheel slippage and illumination changes.

In the next steps of our research we will work on simultaneous localization and mapping approaches in order to consider unmapped traffic signs, what is more likely to happen in real world scenarios.

ACKNOWLEDGMENTS

The authors would like to thank the financial support of FAPEMIG, CNPq, and CAPES.

References

Adams, M., Zhang, S. and Xie, L. (2004). Particle filter based outdoor robot localization using natural features extracted from laser

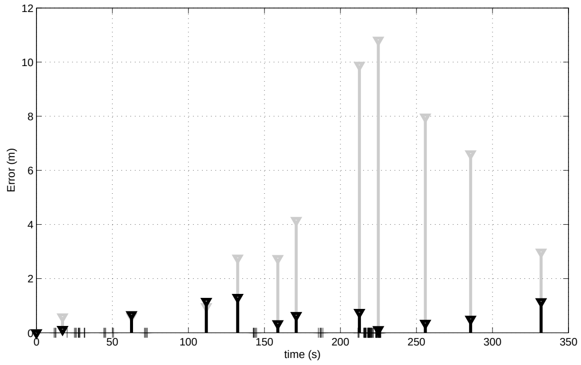

Figure 8: Evolution of the error. The black values represent the full particle filter. The gray values were obtained by using wheel and visual odometry.

scanners, *IEEE Int. Conf. on Robotics and Automation*, Vol. 2, pp. 1 493–1498.

Agrawal, M. and Konolige, K. (2006). Real-time localization in outdoor environments using stereo vision and inexpensive GPS, *IEEE Int. Conf. on Pattern Recognition*, Vol. 3, pp. 1063–1068.

Bahlmann, C., Zhu, Y., Ramesh, V., Pellkofer, M. and Koehler, T. (2005). A system for traffic sign detection, tracking, and recognition using color, shape, and motion information, *Proc. IEEE Intelligent Vehicles Symposium*, pp. 255–260.

Belo, F. A. W. (2006). *Desenvolvimento de algoritmos de exploração e mapeamento visual para robôs móveis de baixo custo*, Master's thesis, Pontifícia Universidade Católica do Rio de Janeiro.

Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, *Comm. ACM* **24**(6): 381–395.

Lowe., D. G. (2004). Distinctive image features from scale-invariant keypoints, *International Journal of Computer Vision* **60**(2): 91–110.

Machado, H. N. and Pereira, G. A. S. (2006). Medição das velocidades de um robô móvel utilizando seqüências de imagens de sua superfície de movimentação, *Anais do Congresso Brasileiro de Automática*, Salvador, Brazil, pp. 1025–1030.

Maeyama, S., Ohya, A. and Yuta, S. (1994). Positioning by tree detection sensor and dead reckoning for outdoor navigation of a mobile robot, *IEEE Int. Conf. on Multisensor Fusion and Integration for Inteligent Systems*, Las Vegas, NV, pp. 653–660.

Miura, J., Kanda, T. and Shirai, Y. (2000). An active vision system for real-time traffic sign recognition, *Proc. IEEE Intelligent Transportation Systems*, pp. 52–57.

Moreira, M. A. G., Machado, H. N., Mendonça, C. F. C. and Pereira, G. A. S. (2007). Mobile robot outdoor localization using planar beacons and visual improved odometry, *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems.*, San Diego, USA.

Moreira, M. A. G. and Pereira, G. A. S. (2006). Localização de robôs móveis a partir de marcos visuais mapeados em tempo real, *Anais do Congresso Brasileiro de Automática (CBA'06)*, pp. 399–404.

Panzieri, S., Pascucci, F. and Giovanni, U. (2002). An outdoor navigation system using GPS and inertial platform, *IEEE/ASME Transactions on Mechatronics* **7**(2): 134–142.

Reckleitis, I. M. (2003). *Cooperative Localization and Multi-Robot Exploration*, PhD thesis, School of Computer Science, McGill University, Montreal, Quebec, Canada.

Thrapp, R., Westbrook, C. and Subramanian, D. (2001). Robust localization algorithms for an autonomous campus tour guide, *IEEE Int. Conf. on Robotics and Automation*, Seoul, Korea, pp. 2065–2071.

Thrun, S., Burgard, W. and Fox, D. (2005). *Probabilistic Robotics*, The MIT Press.

Wolf, W. J., Mathis, D., Sklair, C. W. and Magee, M. (1991). The perspective view of three points, *IEEE Trans. Pattern Anal. Machine Intell.* **13**: 66–73.