APPLYING NEURAL NETWORKS IN 3-D VISION

MÁRCIO MENDONÇA[(1)], IVAN N. DA SILVA[(2)], JOSÉ E. C. CASTANHO[(2)]

[(1)]Department of Electrical Technology
Federal Center of Technology and Education
86300-000, Cornelio Procópio, PR
Brazil

[(2)]Department of Electrical Engineering
School of Engineering, São Paulo State University, CP. 473
17015-970, Bauru, SP
Brazil

Abstract: - This work presents a proposal for camera calibration using neural networks of the type multilayer perceptron. Camera calibration is employed in 3-D computer vision for pose determination and it requires a solution of non-linear system of equations. By employing neural network, it becomes unnecessary to know the parameters of the cameras, such as focus, distortions and aspects referent to the geometry of the system. Camera simulations and real experiments are used to demonstrate and evaluate the proposed.

Key-Words: - Computer vision, camera calibration, neural networks, stereo vision, 3-D vision, 3-D image acquisition.

1 Introduction

The goal of camera calibration is to establish the relationship between global 3-D coordinates of a point and 2-D coordinates of the projected image [3]. The process of camera calibration is a pre-requirement for most applications in computer vision to determine the position and orientation of an object in relation to a global coordinate system. Most methods of camera calibration usually require a careful and difficult procedure and include a complex mathematical model.

In this work, it is presented a method to establish the relationship between 3-D coordinates and the image coordinates of a point through neural networks. The method is advantageous when compared to traditional ones since it does not need the information of the camera geometry and nor a complex experimental procedure to be settled. Therefore, the procedure can be more useful since it is easy to employ in common situations.

There are many techniques to obtain 3-D information [13], [8] [2] including stereo vision [1], [15], and the use of structured light [12], [6]. In stereo vision, two or more cameras are used to visualize the same point in the space inside the vision field.

So, each camera obtains a different coordinate for this point in its respective images, that is, each point in the space can generate only a pair of coordinates, (x_1, y_1) and (x_2, y_2), respectively in the each camera. In that way, using triangulation it is possible to recover the point position in the space starting from the coordinates of the point projected in both cameras. However, this implies in solving a non-linear system of equations, which is usually done through optimization techniques.

With the proposed method, a neural network is used to learn the relationship among global coordinates and camera coordinates. Moreover, it is straight up to recover the 3-D information without knowing camera parameters explicitly if there are two images of the same scene.

The method is also robust enough for dealing with different cameras, which have different focal lengths, because it is capable of recognizing the actual focus of the cameras once the neural networks have been trained.

The neural network training is accomplished using coordinates of points in the space and their respective projections in a group of cameras. The inputs of the neural network are the image points in

the cameras and the outputs are the corresponding 3-D space coordinates of each point.

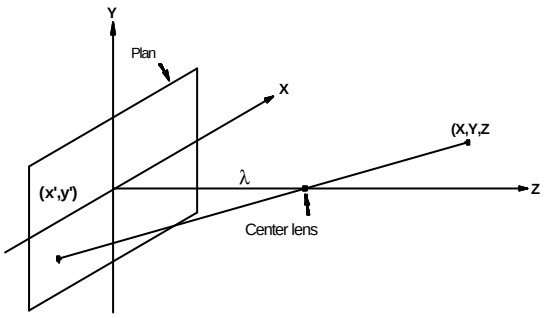

Fig. 1 - The pinhole camera model for the perspective transformation.

In the following sections, it is presented a revision of the theory underlying the problem of camera calibration and after that, the proposed method is described. In section 3 the test results are presented and analyzed.

2 The camera model

Each point in the space can have only one corresponding projection in images - (x_1, y_1) *and* (x_2, y_2) - for each camera. Using the camera model (see Fig.1), it is possible to determine the coordinate x and y of a given point in the field of vision for both images. This model [5], that is used to simulate the camera, includes the rotation, translation, perspective, and location transformations that are needed to get the image projection of a point in space. Usually, the coordinates of cameras are expressed in homogeneous form to simplify the processing of matrices. The model used in this work considers rotation in only two axes. The complete transformation from real world coordinates to image coordinates is given by the composition of all the transformations as shown in Eq. 1, [4].

Where the perspective transformation, is given by:

$$C_h = P \times C \times R \times G \times W_h \qquad (1)$$

The components of Eq. 1 are described below:

The matrix ***P***, for perspective transformation, is given by:

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{-1}{\lambda} & 1 \end{bmatrix} \qquad (2)$$

The matrix ***G***, for translation of the camera in relation to the origin of the global coordinates, is given by:

$$G = \begin{bmatrix} 1 & 0 & 0 & -X_0 \\ 0 & 1 & 0 & -Y_0 \\ 0 & 0 & 1 & -Z_0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \qquad (3)$$

Matrix rotation ***R***, with two degrees of freedom, is given by:

$$R = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta.\cos\alpha & \cos\theta.\cos\alpha & \sin\alpha & 0 \\ \sin\theta.\sin\alpha & -\cos\theta.\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (4)$$

The displacement of the structure to fix the camera in a particular position is given by:

$$C = \begin{bmatrix} 1 & 0 & 0 & -r_1 \\ 0 & 1 & 0 & -r_2 \\ 0 & 0 & 1 & -r_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (5)$$

Wh contains the global coordinates expressed in homogeneous form, and ***Ch*** the homogeneous image coordinates after transformations. Image coordinates in both cameras are obtained from points (x_1, y_1), and (x_2, y_2) by Eq. 1.

Then, for camera 1 the matrix ***Ch*** is the following:

$$C_h^1 = P_1 \times C_1 \times R_1 \times G_1 \times W_h \qquad (6)$$

$$C_h^2 = P_2 \times C_2 \times R_2 \times G_2 \times W_h \qquad (7)$$

The matrix ***Ch*** has dimension 4x1 and the camera coordinates of the projected points can be obtained by:

$$X_1 = \frac{C_{h1}^1}{C_{h4}^1} \qquad (8)$$

$$Y_1 = \frac{C_{h2}^1}{C_{h4}^1} \quad (9)$$

$$X_2 = \frac{C_{h1}^2}{C_{h4}^2} \quad (10)$$

$$Y_2 = \frac{C_{h2}^2}{C_{h4}^2} \quad (11)$$

This model was used to simulate the experiments under controlled conditions. This model is a quite simple representation of a real camera, since it does not consider the lens distortions. However, it is good enough for the first steps of this work. In several works lens distortions are considered and suitable mathematical models are developed [8]. In this work, the above problem is not considered since it is supposed that using neural networks the distortions problems will be automatically accounted for.

In the next section, the implemented approach for the camera calibration is discussed.

3 Implementation

In this work, the solved of the calibration camera using neural networks. Neural networks are suitable for solving non linear problems, and are capable of learning relationships between the camera coordinate and global coordinate [9], [14], [11], [10], without needing to determine a complex mathematical model. As presented in Fig. 2, the layers of a neural network are usually classified in three groups. Input layer, where the patterns are presented to the network, in this case as cameras coordinates (x_{1i}, y_{1i}) and (x_{2i}, y_{2i}). Hidden layers, where most of processing is carried out through weighted connections, can be considered as characteristics extractors. Out put layer presents the results, which in this work are the global coordinates ***Xg, Yg*** e ***Zg***.

To implement the neural networks it was used the Neural Network Matlab Toolbox and the Levemberg [7]. method for training. The hardware used was a PC with a CPU AMD K6-2, 500 Mhz with 128MB. Two different cameras were used in the experiments: a WebCam from Creative Labs and a Digital Camera from Sony. The experiments have been comprised comparisons among the results from simulations of camera model and those using real images. In addition, to have a reference for comparison the problem was solved using least square estimation method. Both results are presented.

For purposes of experimentation, it were generated several grids of points in random space inside a defined area or volume. The first experiments were conducted with a square grid with around 300 points with the ***Zg*** coordinate fixed in 0.2m, and an area delimited by coordinates ***Xg*** and ***Yg*** ranging from 0.65m to 1.35m. The experiments with volumes used the same area with points spread out in a volume with 1.0m of height. The experiments with real images were conducted with a square grid and volumes similar of those used in simulated images.

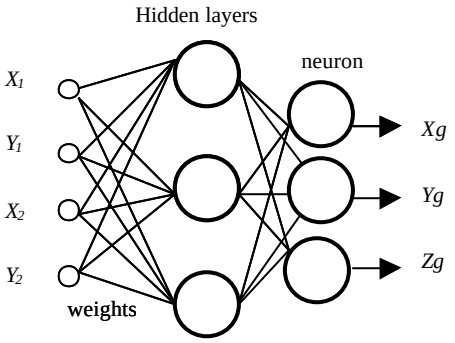

Fig. 2 - Neural Network

When using simulated images two types of data were generated. One data set was generated without noise, and in the other, a Gaussian noise was added to represent the acquisition error for image coordinates. Experiments included several different configurations to evaluate the influence of number of neurons and layers in the results, as well as the accuracy of training. Experiments are arranged in six groups:

- Training in which the network recognizes camera intrinsic parameters as the focus with synthetic data set;
- Training in a surface (constant ***z*** coordinates) using synthetic data set;
- Training in a volume using synthetic data.
- Training with a surface using real images and points.
- Training with a volume with real images planes and real points.
- Training in which the net recognizes camera position in space.

The results of theses experiments are presented and discussed in the next section.

4 Results and discussion

The generated and simulated data sets were used as inputs for the camera calibration problem for both neural networks and least square solution. The absolute and relative error between the actual simulated coordinates and those solutions inferred using neural network is show Table 1, as well as, the least square solution in Table 2. Table 3 shows several configurations parameters used in network experiments, such as topology, number of epochs, neurons, and input. In experiments 1 and 2, it was used neural networks with a simulated input data set. The very small error obtained demonstrates the suitability of the approach.

Table 1 - Results using neural networks.

Exp.	Points	Absolute Error	Relative Error
01	300	0.00000	0%
02	300	0.00001	0%
03	250[E]	0.00213	0.2%
04	250[E]	0.00091	0.1%
06	200	-0.00011	0%
07	250	-0.33330	0%
08	250	0.00010	0%
09	250	0.00025	0%
13	300[A]	0.00117	0.1%
14	100	-0.00024	0%
15	300	0.00000	0%
18 [R]	130	0.20560	5%
20 [R]	130	-0.28720	-2%
21 [R]	130	-0.01500	-4%
22 [R]	520	0.01895	1%
23 [R]	130[P]	0.02560	1.5%

Table 2 - Results using Least Square Method.

Exp.	Points	Absolute Error	Relative error
05	150[E]	-0.00678	-0.8%
10	100 [A]	0.01169	1.2%
11	200	-0.00267	-0.3%
12	300	0.00076	0.1%
17 [R]	130	0.23590	4%
19 [R]	130	-0.03695	1%

A Random points
R Experiments with real points.
E Simulations of the Gaussian error
F Simulations of the focus recognize
P Experience of the position recognize

In experiments 3, 4 and 5 it was added a Gaussian error of 5% in input data to get a closer simulation of the actual conditions of image acquisition. Experiments 3 and 4 were conducted by neural networks with good results. Experiment 5 was run using least square estimation with comparable results.

Experiment 6 was carried out using cameras with different focus lengths, after a previous training with two camera models. This showed that the neural networks were capable of identifying which model was used. Experiments 7, 8, and 9 were accomplished with neural network and another set of synthetic data in a plane surface as input. In experiments 10, 11, and 12 the input data was generated in a volume delimited by a parallelepiped and the solution was got using least square method. The experiments 13, 14, and 15 were conducted with test data generated in a volume delimited by a parallelepiped and the solution was obtained using neural networks.

Table 3 - Characteristics Neural Network.

Num.	Error	Epochs	Topology	Input
01	10^{-7}	7	[20-30-3]	300
02	10^{-7}	9	[20-30-3]	300
03	10^{-6}	252	[20-30-3]	250 [E]
04	10^{-7}	989	[20-30-3]	250 [E]
06	10^{-8}	43	[20-30-5] [F]	200
07	10^{-6}	7	[20-30-3]	250
08	10^{-7}	9	[20-30-3]	250
09	10^{-8}	12	[20-30-3]	250
13	10^{-7}	10	[20-30-3]	300 [A]
14	10^{-8}	8	[20-30-3]	100
15	10^{-8}	13	[20-30-3]	300
18[R]	10^{-8}	130	[20-30-3]	130
20[R]	10^{-6}	358	[50-3]	130
21[R]	10^{-8}	1359	[50-3]	130
22[R]	10^{-5}	5000	[55-3]	520
23[R]	10^{-6}	1256	[55-4]	130

A Random points
R Experiments with real image points.
E Simulations of the Gaussian error
F Simulations of the focus recognize
P Experience of the position recognize

Experiments 17 and 18 were carried out with acquisition of real points in images of planes using the WebCam, which present a high degree of lens distortion. These experiments made possible to evaluate the behavior of the method using low cost off the shelf cameras. The experiments 19, 20, and 21 were similar to 16 and 17 with the difference that they were got using a Sony camera. Both methods

least square and neural network presented similar results.

The experiment 22 was done using a neural network and real images with the training points inside a 3-D volume. In this experiment, a total of 520 points spread out in four planes were chosen in the digitized test image.

In the experiment 23, results were obtained using a neural network with one additional neuron in the output layer. This was done to enable the neural network to produce the distance of the camera from a reference in space [16]. This is useful in applications where the camera is not fixed and its position information is of interest.

All the results were obtained using neural networks with one or two hidden layers. Observing the results it is possible to conclude that they are similar in both cases, confirming that the network was able to generalize and solve the problem. At least one hidden layer is necessary to solve any non linear problem [7]. The need to use more than two hidden layers will occur only when the problem is in a discontinuous domain, which is not the present case. That is, a calibration function exists in the continuous domain. The experiments in which the neural network was trained with error 10^{-6} have produced good results. Increasing the training error to 10^{-7} and 10^{-8} does not produce significant improvement in the accuracy, and increases the processing time. In real images the neural network were trained with error 10^{-5} for volumes and 10^{-7}, 10^{-8} for surface when the processing time was not so long.

Using less than 100 patterns for tests the obtained results showed inaccurate. Increasing to 200 patterns or more the error has decreased to an insignificant value. In cases in which Gaussian error was added in the input, the neural network still presented good accuracy demonstrating its robustness. The neural network used in the work in the work have reached generalization in all cases, presenting better precision than least square methods in simulated data. However, in experiments with real images the two methods are equivalent with a small advantage for least square. However, the accuracy in both methods is not too significant. Thus, using neural networks is simpler since the implementation does not requires a complex mathematical model for the camera, and avoids much of the practical calibration details, mainly the geometric displacement of the system and the lens distortion.

5 Conclusions

In this work a camera calibration method using neural networks in the context of 3-D information recovery was presented, and the results were compared with those provided by the traditional least square estimation. The main advantages of the proposed method are:

One does not need to know complex mathematical models, that is, using the method does not require deep technical knowledge, and so it is useful for an inexpert user;

An initial estimation of calibration parameters are not needed. The method can be applied to several different cameras.

The neural network can sense and recognize when different cameras are used, and it will still produce the correct outputs, once it was previously trained, without the need of an explicitly input parameter that warns the system of the change.

The neural network can also recognize some specific positions of camera, since it has been previously trained. This feature is useful in dynamic systems. Therefore, the presented approach is very flexible and significantly more easy of being employed than those previously presented in the literature.

References:

[1] Aguilar, J. , F. Torres and M. Lopes *Stereo Vision for 3-D measurement: accuracy analysis, calibration and industrial applications.* Elsevier Science. vol. 18, n.4, 1996,pp. 193-200.

[2] Andreff N., R. Horaud, and B Espiau. *On-line Hand-Eye Calibration.* in Second International Conference on 3-D Digital Imaging and Modeling (3DIM'99), 1999,pp. 430-436.

[3] Echigo, T. *A Camera Calibration Technique using Three Sets of Parallel Lines*. Machine Vision and Applications. vol.3, 1990, pp.159-167.

[4] Ganapathy, S. *Decomposition of matrices for robot vision.* Pattern Recognition Letters, vol. 2, 1984, pp. 401-412.

[5] Gonzalez, R. C., Richard E. Woods. *Digital Image Processing.* Addison-Wesley Publishing Company, Inc.1992.

[6] Guisser, L., R. Payrissat, S. Castan. *An accurate 3-D vision system using a projected grid for surface descriptions.* Image and Vision Computing, vol. 18, 2000, pp. 463-491.

[7] Haykin, S. Neural Networks, a Comprehensive Foundation. Prentice Hall Inc., 2ed. 1999

[8] Heikkilä, J.. *Geometric Camera Calibration Using Circular Control Points.* IEEE Transactions on Pattern Analysis and Machine Intelligence. vol. 22, n.10, 2000, pp. 1066-1076.

[9] Lynch, M. B.,R. Cihan H. Dagli , Mahesh Vallenki. *The use of fedforward neural networks for*

machine vision calibration. Int. Journal of Production Economics, 60-61, 1999, pp. 479-489.
[10] Oing, G. Wei and G Hirzinger *Multisensory Visual Servoing by a Neural Network. IEEE Transactions on Systems.* Man and Cybernetics - Part B, vol. 29, n.2, 1999.
[11] Tien, F.-C., Chang, C.A. *Using neural networks for 3-D measurement in stereo vision inspection system.* International Journal of Production Research, vol. 37, issue 9, 1999, pp. 1935-1948.
[12] Trucco, E., R., B. Fischer, A. W. Fitzgibbon and D. K. Naidu. *Calibration, data consistency and model acquisition with laser stripers. Computer Integrated Manufacturing*, vol.11, n. 4, 1998, pp. 293-310.
[13] Wang, Ling-Ling, Wen-H. Tsai *Computing Camera Parameters using Vanishing-Line Information from a Rectangular Parallelepiped.* Machine Vision and Applications. vol. 3, 1990, pp. 129-141.
[14] Wells, Gordon, Christophe Venaille, Carme Torras. Promising Research, *Vision-based robot positioning using neural networks . Image and Vision Computing*, vol. 14, 1996, pp. 715-732,.
[15] Weng, Juyang, Paul Cohen, and Marc Herniou. *Camera Calibration with Distortion Models and Accuracy Evaluation*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, n.10, 1992.
[16] Yuncai, Liu, Thomas S. Huang and Olivier D. Faugeras. *Determination of Camera Location form 2-D to 3-D Line and Point Correspondence.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, n.1, 1990.