11th IFAC Workshop on Intelligent Manufacturing Systems
The International Federation of Automatic Control
May 22-24, 2013. São Paulo, Brazil

Camera Calibration Using Detection and Neural Networks

André Vitor Bosisio Moura Pedra*. Marcio Mendonça*. Marco Antonio Ferreira Finocchio*. Lúcia Valéria Ramos de Arruda. José Eduardo Cogo Castanho*****

**Electrical Engineering Departament, Federal Techonological University of Paraná, Cornélio Procópio, PR 86300-000 BRAZIL (Tel: +55 (43) 3520-4000; e-mail: andrebosisio@hotmail.com; mendonca@utfpr.edu.br; mafinocchio@utfpr.edu.br).*
*** Electrical Engineering Departament, Federal Techonological University of Paraná, Curitiba, PR 80230-901 BRAZIL (Tel: +55 (41) 3310-4545; e-mail: lvrarruda@utfpr.edu.br).*
**** Electrical Engineering Departament, State University Júlio de Mesquita Filho, Bauru, SP 17015-970 BRAZIL (Tel: +55 (14) 3103-6115; e-mail: castanho@feb.unesp.br)}*

Abstract: Several applications use robotic vision, such as a robot navigating through an unknown surrounding, can use vision as main navigate sensor. This paper focuses on studying camera calibration via stereo vision by means of neural network. A neurocalibration method is proposed based on the neural networks ability to learn nonlinear relationship among a two and three dimension coordinate systems and also its information generalization skill. The data used to train neural network mapping are generated from a calibration grid point obtained through the use of Harris edge detection algorithm. The experimental results indicated that the neurocalibration method is feasible and efficient.

Keywords: Camera calibration; Computer vision; Neural networks; Harris corner extraction.

1. INTRODUCTION

Computer vision is related to ways of a computer or any machine can see through the environment. Therefore, computer vision science approaches image processing techniques to extract and process important characteristics from objects and their environments. For example, an usual information that can easily be extracted is, the position of a given object expressed as global coordinates in a reference system centered in the camera (Gonzalez and Woods, 2000).

An essential task in autonomous robots is to move safely in an unknown environment (Gini and Marchi, 2002). Visually guided system uses computer vision to detect and recognize obstacles and/or targets allowing safety navigation. This can be reached through extracting metric information from scene and objects displayed. In this context, stereo vision approaches the problem of reconstructing depth metric information with a high precision (Chen Shanshan et al., 2009).

On the other hand, camera calibration is the process of choosing camera parameters such as internal geometry and optical characteristics (intrinsic parameters) and/or 3D position and camera orientation related to general coordinates axis (extrinsic parameters) (Lenz and Tsai, 1988).

There are different techniques for camera calibration. The most known are, linear calibration method proposed by Faugeras and Toscani (apud Salvi et al., 2002), calibration nonlinear optimization (Weng et al., 1992), the two steps method developed by Tsai (1987) and the double plane method (Martins et al., 1981). This last one develops a camera model that establishes a geometrical relationship between image points in 3D coordinates and image pixels in 2D coordinates and vice versa (Ahmed et al., 1999).

Besides the above mentioned parameters, there are also others inherent factors in images, among them we can cite the radial distortion, centrifugal distortion, thin prism distortion and barrel distortion, which are generated by imaging process. These measurement errors are caused by non-uniforms factors of charge-coupled device (CCD), resulting in a nonlinear mapping between images coordinate system and the global coordinate system (Li Guo Jin and Li Guang Rui, 2011).

According to some authors (Li Guo Jin and Li Guang Rui, 2011) the intrinsic and extrinsic parameters of a camera need not be determined or directly obtained to generate a good camera calibration. This study looks for an alternative method to camera calibration, by using neural networks and edge detection algorithms. Neural networks are able to learn the nonlinear relationship between the camera coordinate system and the world coordinate system (Lynch et al., 1999).

This is accomplished by adjusting neural network synaptic weights during the training phase. The mapping among the input data image data and output data (target) are previously known. These image data for training are optimally extracted by using the well-known Harris method to image edge detection (Gao Qingji et al., 2011).

2. CALIBRATION DATA ACQUISITION

A calibration grid with known dimensions is first designed with photogrammetry characteristics and, able to represent the relationship between the scene 3D coordinates and the 2D

978-3-902823-33-5/13/$20.00 © 2013 IFAC

coordinates of the images seen by camera lens. The scheme is show in Fig. 1.

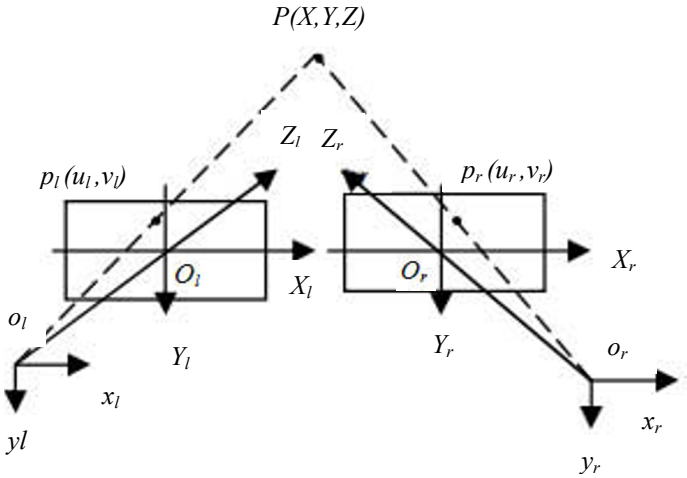

Fig. 1. 3D coordinates system.

The 3D coordinates system is shown at Fig. 1, where $x_l\, y_l\, z_l$ and $x_r\, y_r\, z_r$ are respectively the 3D reference coordinates of left and right cameras, o_l and o_r are optical centers of two cameras; $O_{il}\, X_l\, Y_l$ and $O_{ir}\, X_r\, Y_r$ are the imaging plane or 2D reference frames to left and right cameras. In this system, o_{il} and o_{ir} are respectively the optical axis intersection point of left and right cameras. This point intersect with their own imaging plane, $X_l\, , Y_l$ that is parallel to x_l, y_l, and X_r, Y_r plan that is parallel to x_r, y_r. P is a random point on the surface of the object measured. As both cameras are the point p_l on the left camera image and the point p_r from right camera image are the image points of the same space point P. Because the point P is on both lines $o_l\, p_l$ and $o_r\, p_r$, P point is the intersection point of both lines, namely the 3D position of point P is uniquely determined (apud. Yingjie Xing et al., 2007).

2.1 Case 1 – Two dimensions

In the case of two camera are calibrated to a two dimensions (z=0) system, we obtained two photos of the calibration grid, showing the real experimental settings. Both images have the same incidence angle, among a small horizontal shift between them, as specified in Table 1.

Table 1. 2D shooting distance and angle

	Distance (cm)	Angle (degrees)
Photo 1	23.5	180
Photo 2	23.5	180

The 2D calibration grid point is shown in Fig 2.

Thus we use Harris algorithm in order to extract the edges coordinates of the image in Fig. 2, obtaining the corner display image in Fig. 3, where "*" represents the edge and its respective coordinate (u,v).

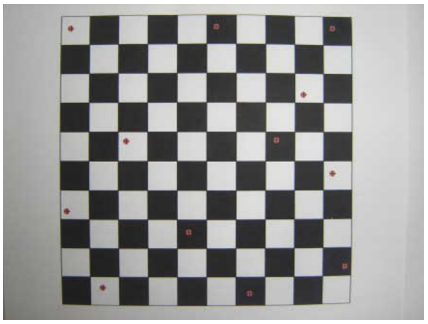

Fig. 2. Calibration grid point - 2D.

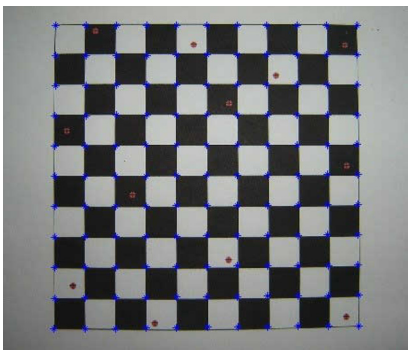

Fig. 3. Corner display.

The several points with (u,v) 2D coordinates of the image composes the input vector to neural network. Since there are 242 measured edges for training, a set of them (12 points) is used to test neural model accuracy. These validation points are circular red color point observed in Fig. 2. The point X and Y coordinates point are referenced to the image bottom line, and they are measured by the Autocad software.

2.2 Case 2 – Three dimensions

A second calibration grid, now referencing to axis Z, which indicates depth, is carried out. Two images are used to recover scene three-dimensional information. Table 2 shows captured images characteristics, when real experimental settings used.

Table 2. 3D shooting distance and angle

	Distance (cm)	Angle (degrees)
Photo 1	25	70
Photo 2	25	70

The 3D calibration grid is shown in Fig. 4. The experiment goal is to correctly identify and measure screw heads position.

The software Autocad is used to determine X and Y coordinate of the screws. The height of each element is measured with a rule with an accuracy of ± 0.02 mm.

In this situation each one of three axes contains 121 points. Thereby, for two images, the input training vector has 726 corners (2D coordinates of the image). A subset containing

12 points randomly distributed between the screw head and targets located in the ZX and ZY planes are used to validation step.

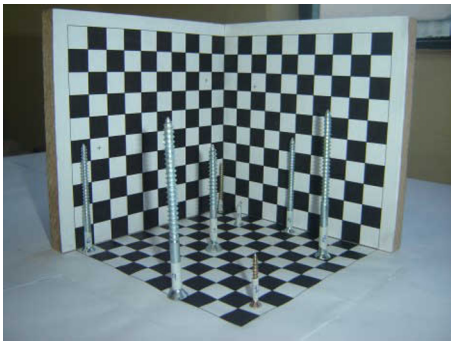

Fig. 4. Calibration grid point - 3D.

3. CALIBRATION WITH NEURAL NETWORK

After acquiring the corners of the images, the neural network must be designed.

3.1 Neural network topology

Artificial multilayers neural networks (ANN's) can approximate any arbitrary continuous function with a priori specified accuracy (Ken-Ichi Funahashi, 1898). This paper uses feedfoward multilayer neural network, with one hidden layer and trained with backpropagation algorithm to calibrate the camera (Haykin, 2001).

The topology of the neural network is depended on the calibration mode (2D or 3D). For two dimensions case, the neural network architecture is shown in Fig. 5. The input vector is composed by right (u_r, v_r) and left (u_l, v_l) camera coordinates and the output vector is the target coordinates (X_w,Y_w).

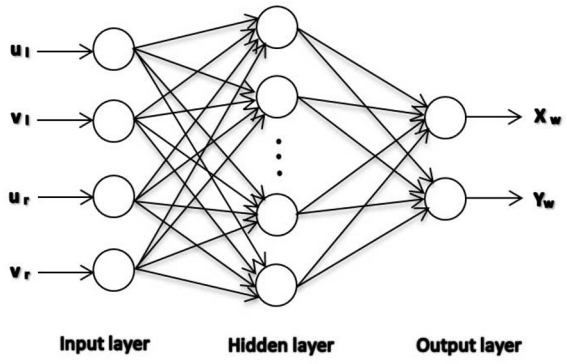

Fig. 5. Topology of the neural network – 2D.

Thus, the input layer has 4 neurons corresponding to images edges while the output layer consists of two neurons representing the world coordinates (X_w, Y_w).

For three dimensions case, output layer is augmented with information about image depth. Thus, the output layer has three neurons, representing the world coordinates X_w, Y_w and Z_w.

All neural models are developed with Neural Network Toolbox (NNT) of Matlab software.

Regarding the neural network hidden layer, the j^{th} input (net_j) and output (Y_j) hidden neurons are respectively:

$$net_j = \sum_{j}^{n} W_{ji} \cdot x_i , \qquad (1)$$

$$Y_j = f_1(net_j), \quad (i = 1,2,3;\ j = 1,2,3,...,\ m). \qquad (2)$$

Where the symbol n is the number of neuron nodes in the input layer, m is the number of hidden layer neuron nodes, W_{ji} is the synaptic weight between the input layer nodes i and the hidden node j, the function f_1 is the transfer function of the hidden layer. In this paper, *tansig* function is adopted, whose mathematical expression is as follow:

$$f(x) = \frac{2}{1 + e^{-2 \cdot x}} - 1 . \qquad (3)$$

The k^{th} network output layer node is:

$$y_k = f_2(\sum_{j}^{m} w_{jk} Y_j), \quad (j = 1,2,3,...,\ m; k = 1,2,3). \qquad (4)$$

Where the symbol w_{jk} is the synaptic weight between the hidden node j and the output layer node k, f_2 is the transfer function of the output layer. In this paper, linear transfer function *purelin* is adopted, whose mathematical formula is:

$$f(x) = kx . \qquad (5)$$

The network performance was evaluated according to the Mean Square Error (MSE) function which measures the error between the network responses and desired values. The mathematical expression of the MSE is given by:

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (\varepsilon_i)^2 . \qquad (6)$$

Where ε_i is the difference between neural network responses and desired values.

The adopted training function is *trainlm*, which updates the weights and bias by Levenberg-Marquardt (LM) optimization method. This supervised learning method presents high success rate and speed. It is known that for certain problems LM algorithm is 10 to 1000 times faster than the basic backpropagation algorithm (Hagan, 1994).

In this algorithm, Jacobian jx of the objective function E with respect to the weight and bias variables x is computed. The

978-3-902823-33-5/13/$20.00 © 2013 IFAC

objective function corresponds to MSE function in equation 6, during training step. Dummy variables *jj* and *je* are computed and used to adjusted the increment of weight and bias variables *x,* according to Levenberg-Marquardt method. The mathematical computations are shown below.

$$jj = jx \cdot jx,$$
$$je = jx \cdot E, \qquad (7)$$
$$dx = \frac{-(jj + I \cdot mu)}{je}.$$

Where *E* is all errors, *I* is the identity matrix and *mu* is a regularized parameter that can vary during the training.

3.2 Neural network training step

The most important property of neural networks is the ability to learn from its environment and thereby improve its performance. This is done through an interactive process of weights up-date. This process is named training and learning occurs when the ANN reaches a generalized solution for a class of problems (Haykin, 2001).

The ANN learning capability is directly related to the number of neurons in the hidden layer. Thus, it is necessary to define an optimal architecture (size of hidden layer) in order to obtain a lower error in the training phase and to prevent the network only memorize initial information (overfitting).

For this, we randomly varied the number of neurons in the hidden layer and checked the achieved performance as a selection criterion. A summary of results is shown is Table 3 and Table 4 for 2D and 3D cases. In these tables, goal column is the desired mse error value, actual iteration steps column correspond to the number of iteration to attain the goal, the number of nodes is in the hidden nodes column and performance is the attained mse error.

Table 3. Neural models with different hidden layer – 2D

Goal	Maximum iteration steps	Actual iteration steps	Hidden nodes	Performance (MSE)
0.000001	300	276	3	3.3301e-6
		165	5	2.3301e-6
		116	10	3.2057e-6
		117	15	5.4309e-6
		300	20	2.6301e-6
		175	25	2.7042e-6
		36	30	3.3850e-6
		63	35	3.4528e-6
		300	40	2.7695e-6
		167	45	4.5348e-6

Generally, data set used during training is divided into three subsets. The first subset is the training set, which is used to compute and update network weights and biases. The second subset is the validation set. The validation set is used to monitor error decaying during the training process. Validation error normally decreases during the initial phase of training, as does the training set and it is used to stop training step.

Table 4 Neural models with different hidden layer - 3D

Goal	Maximum iteration steps	Actual iteration steps	Hidden nodes	Performance (MSE)
0.00001	80	67	5	0.00017407
		80	8	9.1289e-5
		53	10	0.00012156
		60	15	0.00013298
		64	20	0.00010026
		80	25	0.00010000
		19	30	0.00013781
		41	35	0.00016037
		73	40	0.00011224
		80	45	0.00016815
		14	50	0.00010149

However, if the network over fits data then validation errors typically begins to rise. Thus network weight and biases are saved at the minimum of validation error.

The last subset is the test set that is not used during training. These data are used to compare different models. If the error on the test set reaches a minimum at a significantly different iteration number than the validation set, this might indicate a poor division of the data set.

The function used in this paper to divide data set was *dividerand*, which compose training subset with 70% of data, 15% for validation set and 15% for testing set.

Fig. 6 shows the training process for the case of tree dimensions, with above mentioned configurations.

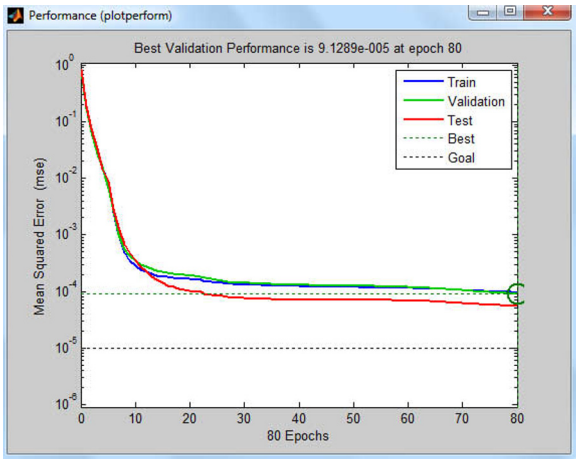

Fig. 6. Training process – 3D.

3. CAMERA CALIBRATION TEST

After the training process, validation data were presented to the network. For 2D calibration method, Table 5 world and RNA coordinates and obtained errors in this validation step. Table 6, we note that the average error in the X coordinate is

0.0398 cm and the Y coordinate error is 0.0393 cm. The 2D real and measured points are shown in the picture at Fig. 7.

Table 5. Camera calibration test result – 2D

World coordinates		RNA coordinates		RMS error
X (cm)	*Y (cm)*	*X (cm)*	*Y (cm)*	*(cm)*
0.5082	14.3564	0.5189	14.4274	0.0718
7.9683	14.391	7.9838	14.4494	0.0604
13.9786	14.391	14.0484	14.4326	0.0813
12.4473	10.9087	12.4870	10.9707	0.0736
3.4158	8.5206	3.4679	8.5773	0.0770
11.0389	8.5851	11.0712	8.5957	0.0340
13.9786	6.8551	14.0295	6.8967	0.0657
0.3418	4.9119	0.3619	4.9413	0.0356
6.5764	3.8541	6.6172	3.8398	0.0432
14.6616	1.9965	14.7278	2.0042	0.0666
			Average RMS error	0.001667

Table 6. Average absolute error - 2D

X (cm)	Y (cm)
0.0398	0.0393

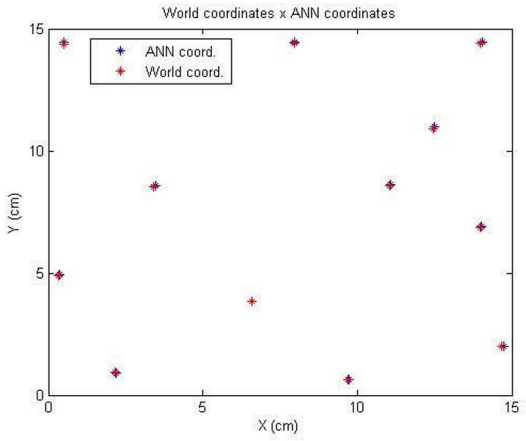

Fig. 7. Comparative world coordinates and ANN coordinates.

Analogously to the previous case, the same results can be shown for three dimensions case. Table 7 shows world and RNA coordinates and obtained errors in this validation step.

Table 7. Camera calibration test result - 3D

World coordinates			RNA coordinates			RMS error
X (cm)	*Y (cm)*	*Z cm)*	*X (cm)*	*Y (cm)*	*Z (cm)*	*(cm)*
0.5082	14.3564	7.494	0.5225	14.5730	7.2794	0.3052
13.9786	14.391	4.486	14.1745	14.4124	4.2573	0.3019
12.4473	10.9087	1.902	12.6175	10.9645	1.6858	0.2808
6.8061	8.5206	7.5	6.7013	8.6214	7.5417	0.1513
13.9786	6.8551	7.412	14.2924	7.2013	7.2665	0.4894
0.3418	4.9119	9.992	0.3299	5.5120	9.9045	0.6066
11.193	2.2382	10.024	11.5106	2.5960	9.9602	0.4827
2.1492	0.9292	2.918	2.5535	1.7033	2.7484	0.8896
12.8508	15	11.2886	13.0477	15.1610	11.2876	0.2543
2.2044	15	6.7977	2.1418	15.0465	6.7807	0.0798
15	11.6141	10.9087	15.0838	11.7013	10.8628	0.1294
15	12.8508	0.9292	15.0248	12.8340	1.0008	0.0776
					Average RMS error	0.3374

Table 8, we note that the average error to coordinates X, Y and Z are respectively 0.1584 cm, 0.2320 cm and 0.1086 cm. The 3D real and measured points are shown in the picture at Fig. 8.

Table 8. Average absolute error - 3D

X (cm)	Y (cm)	Z (cm)
0.1584	0.2320	0.1086

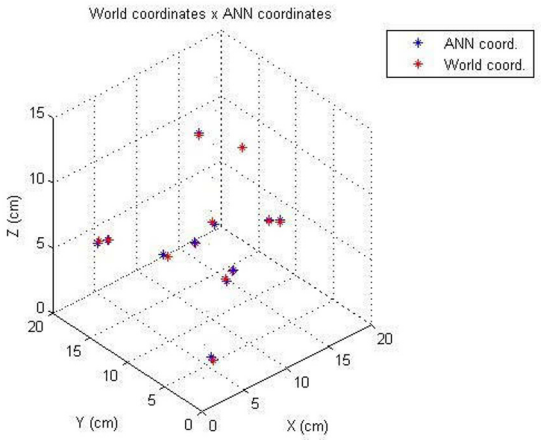

Fig. 8. Comparative world coordinates and ANN coordinates.

4. CONCLUSION

In this paper, we develop a neural network model for camera calibration. The neural model is a mapping function that can replace the camera model used in several computer vision methods. The neural model is trained with real images of a 3D grid. In these images, the corner edge coordinates are extracted with Harris algorithm, limiting the 3D references frames.

The presence of noise in these types of applications is commonly due to in distortion errors and/or acquisition errors to obtain of image coordinates. To circumvent this drawback, several images with different poses and distortions are acquired and used to train the neural network model. The final results are shown the proposed method is robust and accurate.

In fact, the neural network found the nonlinear relationship between image coordinates and world coordinates, with small errors. Thus the proposed method is recommended to applications where there is difficulty in obtaining camera geometric models.

As future work, we intend to apply this method in autonomous robotic systems, allowing the robot to recognize geometric objects (targets and obstacles, for example) within its vision field.

5. REFERENCES

Armed, M. T., Hemayed, E. E. and Farag, A. A. (1999). Neurocalibration: A Neural Networks That Can Tell

Camera Calibration Parameters. *IEEE International Conference on Computer Vision,* Vol. 1, pp. 463-468.

Chen Shanshan, Zuo Wuheng and Zheng Lijun (2009). Camera Calibration via Stereo Vision Using Tsai's Method. *Education Technology and Computer Science, ETCS'09, First International Workshop on*, Vol. 3, pp. 273-277.

Gao Qingji, Xu Ping and Wang Man (2011). Breakage detection for grid images based on improved Harris corner. *Transportation, Mechanical and Electrical Engineering (TMEE), 2011 International Conference on.* pp. 2339-2342.

Ge Dong-Yuan, Yao Xi-Fan (2009). Camera Calibration and Precision Analysus Based on BP Neural Network. *Image and Signal Processing, CISP'09. 2nd International Congress on*, pp. 1-5.

Gini, G. and Marchi, A. (2002). Indoor Robot Navigation with Single Camera Vision. *PRIS'02*, pp 67-76.

Gonzalez, R. C. and Woods, R. E (2000). *Processamento de Imagens Digitais.* Ed. Edgard Blücher, São Paulo, SP.

Hagan, M. T. and Menhaj, M. B. (1994). Training Feedforward Networks with the Marquardt Algorithm. *IEEE Transaction on Neural Networks,* 5(6), pp 989-993.

Haykin, S. (2001*). Redes Neurais – Princípios e prática*, 2nd ed., Bookman, Porto Alegre, RS.

Ken-Ichi Funahashi (1989). On the Aproximate Realization of Continuous Mappings by Neural Networks. *Neural Networks*, 2(3), pp. 183-192.

Lenz, R. and Tsai, R. (1988). Techniques for Calibration of the Scale Factor and Image for High Accuracy 3-D Machine Vision Metrology. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(5), pp. 68-75.

Li Guo Jin and Li Guang Rui (2011). Camera Calibration for Monocular Vision System Based on Harris Corner Extraction and Neural Network. *Consumer Electronics, Communication and Networks (CECNet)*, *2011 International Conference on,* pp. 1-4.

Lynch, M. B., H. Dagli, C. H. and Vallenki, M. (1999). The Use of Feedforward Neural Networks for Machine Vision Calibration*. International Journal Production Economics,* 60-61(1), pp. 479-489.

Martins, H. A., Birk, J. R. and Kelley, R. B. (1981). Camera Models Based on Data from Two Calibration Planes. *Computer Graphics Image Processing*, 17(2), pp. 173-180.

Mendonça. M. (2003). *Redes Neurais Artificiais Aplicadas à Visão Computacional. Dissertação de mestrado,* Universidade Estadual Paulista, Bauru, SP.

Salvi, J., Armangué, X. and Batlle, J. (2002). A comparative review of camera calibrating methods with accuracy evaluation. *Pattern Recognition,* 35(10), pp. 1617-1635.

Tsai, R. Y. (1987). A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses. *IEEE Journal of Robotics and Automation*, 3(4), pp.323-344.

Volosyak, I., Kouzmitcheva, O., Ristié, D. and Gräser A. (2005). Improvement of Visual Perceptual Capabilities by Feedback Structures for Robotic System FRIEND. *IEEE Transaction on System*, 35(1), pp. 66-74.

Weng, J., Cohen, P. and Herniou, M. (1992). Camera Calibration with Distortion Models And Accuracy Evaluation. *IEEE Transaction on Pattern Analysis and Machine Intelligence,* 14(10), pp. 965-980.

Yingjie Xing, Jing Sun and Zhentong Chen (2007). Analyzing and Improving of Neural Networks used in Stereo Calibration*. Third International Natural Computation. ICNC 2007*, Vol.1, pp. 745-749.