논문 2002-11-4-07

A Design of Multilayer Perceptron for Camera Calibration**

Yongtae Do*

Abstract

In this paper a new design of multi-layer perceptron(MLP) for camera calibration is proposed. Most existing techniques determine a transformation from 3D spatial points to their image points and camera parameters are tried to be estimated from the transformation. The technique proposed here, on the other hand, determines rays of sight uniquely from image points and parameters are estimated from the relationship using an MLP. By this approach projection and back-projection can be done more straightforwardly. Being based on a geometric model, a network design process becomes less ambiguous, which is a clear merit compared to other neural net based techniques. An MLP designed according to the technique proposed showed fast and stable learning in tests under various conditions.

1. Introduction

Vision is the most important and useful sense for both humans and machines. For successful visual sensing, an accurate mapping between the space viewed and corresponding image captured is important. Humans and animals learn or possess this mapping capability by nature. For machines, however, when they use cameras for visual perception, the intrinsic and extrinsic parameters of cameras should be computed first implicitly or explicitly to determine the mapping. This process called camera calibration is thus a key step for further processing in most 3D machine vision applications.

Although the problems of stereo and motion have been with the major research interests in the field of 3D vision, Faugueras[1] pointed out that camera calibration is even more important practically than these noble problems by two reasons like; (i) information obtainable by calibration is a prerequisite for all stereo algorithms, and (ii) calibration is basically the same as estimating the motion of a camera.

Many techniques have been proposed to calibrate cameras for the purpose of projection or back-projection. Existing camera calibration techniques can be classified by different criteria. For example, they can be grouped as linear or nonlinear[2], implicit or explicit[3], and analytic or iterative[4]. Since every technique has its own advantages and disadvantages[4,5], no one can be the absolute best in different conditions and applications.

Some researchers have tried to employ an artificial neural network for camera calibration based on its function approximation capability[6]. A major way of using neural networks is correcting an existing(non-neural) technique to reduce error. Wen and Schweitzer[7], for example, used a multi-layer perceptron(MLP) to identify the part that could not be described by an explicit camera model. Kume and Kanade[8] used an MLP to convert ideal image coordinates to real image coordinates for a camera. Choi and Oh[9] used the same approach but divided the image plane for higher accuracy and learning efficiency employing another network. On the other hand, Jun and Kim[10] used an MLP for

* School of Computer and Communication Engineering, Daegu University, 712-714, Korea

** This research was supported in part by the Daegu University Research Grant, 2000
<접수일자 : 2002년 4월 8일>

learning the whole projection mapping of a camera rather than using it partly. Neural networks were used also for back-projection. It was shown that both whole back-projection and the error of a linear technique could be learned by an MLP[11]. An MLP was applied for learning the back-projection of a stereo pan-tilt camera system[12].

In spite of positive results like the above, there are some disadvantages also that hinder its practical applications. First, it is rather ambiguous to determine a proper structure of the network used for calibration. Different numbers of hidden layers and nodes should be tested for a given data and it certainly is a very tedious work requiring considerable time and energy consumption. Second, when a camera calibrated is moved even slightly, the whole calibration process should be performed again for the same but moved camera. Third, the result of camera calibration for projection can not be utilized for back-projection, and vice versa. These problems are due mainly to the fact that most neural networks employed for calibration learn only the mapping between 3D world points and 2D image points implicitly. Thus, the weights of nets' synapses do not contain any physical meaning.

Recently Ahmed and his colleagues[13] presented an MLP structure by which a camera can be calibrated explicitly. Since it was designed based on a physical model, the connection weights of the network were related directly to the position, orientation and optical parameters of the camera calibrated. Therefore, no need for searching an optimal network structure is required unlike other techniques employing neural networks. Furthermore, the orthonormality of the rotation matrix between systems of world coordinate and camera coordinate could be kept without additional steps unlike existing analytical methods like [14]. However, the number of parameters to be learned in the Ahmed's network is always more than that of data given for calibration. This is because different scaling factors, that are required to map 3D points onto points on 2D plane, should be learned for different data. Although the constraint of the orthornormality they applied in determining rotation matrix might enable to reach an unique solution, the learning was slow in our test.

In this paper, we propose a new design of MLP for camera calibration. Like Ahmed's, it is designed based on a physical camera model and all advantages of Ahmed's approach can be found here. In addition, unlike almost all existing techniques including Ahmed's, where calibration was done to optimize the mapping from 3D points to 2D image points, the technique proposed in this paper learns the mapping from 2D image points to their rays of sight. Since this is unique and an one-to-one mapping, the projection and back-projection is very straightforward and easy to be done.

In the next section we review the Ahmed's method briefly, which has some similarities to our technique. The proposed method is then described in Section 3 in detail and tested in Section 4. Conclusions are given in Section 5.

2. An Explicit Camera Calibration by MLP

Most existing camera calibration techniques employ the pin-hole model due mainly to its simplicity. This model assumes the existence of a virtual point(pin-hole) through that all rays of sight pass. A 3D point in the world coordinate system {W}, $\underline{p}^W = (x, y, z)^T$, can then be related to the corresponding image point, $\underline{i} = (u, v)^T$, by the following equation

$$\begin{pmatrix} \gamma u \\ \gamma v \\ \gamma \end{pmatrix} = \boldsymbol{P}\ \boldsymbol{T} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \tag{1}$$

where $P=\begin{pmatrix} -f & 0 & u_0 & 0 \\ 0 & -f & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$ is the intrinsic parameter matrix,

$T=\begin{pmatrix} r_{11} & r_{12} & r_{13} & d_x \\ r_{21} & r_{22} & r_{23} & d_y \\ r_{31} & r_{32} & r_{33} & d_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$ is the extrinsic parameter matrix, γ is a scaling factor for a point, f is the focal length, $(u_0, v_0)^T$ is the optical center of the image plane, $\{r_{11}, r_{12}, \cdots, r_{33}\}$ and $\{d_x, d_y, d_z\}$ are the rotation and displacement elements of the pose of a camera calibrated. Except γ, that is varying dependent on $\underline{p}^W$, all other parameters are constants if the camera calibrated is stationary.

Ahmed[13] designed an MLP to learn the projection mapping represented in eq.(1) by minimizing the error function of the below with N number of data given

$$E=\sum_{t=1}^{N}\{(\gamma_t o_{1t}-u_t)^2+(\gamma_t o_{2t}-v_t)^2+(\gamma_t o_{3t}-1)^2\} \quad (2)$$

where o_{mt}, $m=1,\cdots,3$, $t=1,\cdots,N$, are the computed outputs of the net. The network was designed to learn the extrinsic parameter matrix **T** and the intrinsic parameter matrix **P** with connections of input-to-hidden and hidden-to-output layers respectively by an error gradient descent learning algorithm. When the scale factor γ is included in the learning scheme, the network can be represented like figure 1.

The number of parameters to be determined is not fixed and always more than that of training data given because each datum has its own scaling factor. Due to this reason, it is basically impossible to get an unique solution while using the orthornormality of the rotational matrix minimizing the following error function can provide some constraints in parameter searching

$$E_{orth}=\sum_{n=1}^{3}(r_{n1}^2+r_{n2}^2+r_{n3}^2-1)+(r_{11}r_{21}+r_{12}r_{22}+r_{13}r_{23})^2 +(r_{11}r_{31}+r_{12}r_{32}+r_{13}r_{33})^2+(r_{21}r_{31}+r_{22}r_{32}+r_{23}r_{33})^2 \quad (3)$$

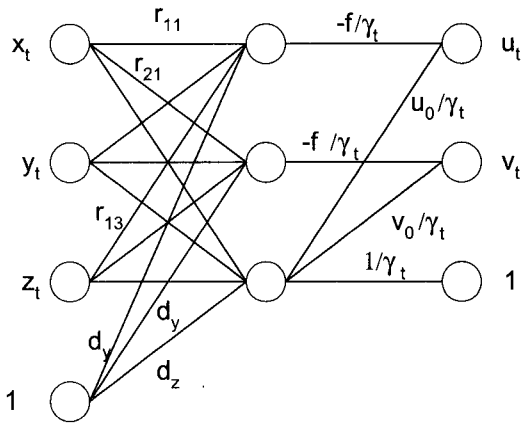

Figure 1. An MLP to learn the perspective projection of a camera.

3. Design of an MLP to Learn the Mapping from an Image Point to the Corresponding Ray of Sight

Almost all existing camera calibration techniques including that proposed by Ahmed find camera parameters with the projection transformation, that maps 3D points to the corresponding 2D image points. As a many-to-one mapping, it requires the learning of an additional parameter like γ of eq.(2) and the reserve mapping is impossible. We, in this paper, propose a new technique that finds the mapping from image points to their rays of sight rather than 3D points. Although infinite number of 3D points can be projected onto the same image point, the ray of sight on which all 3D points projected on the same image point lie is unique for an image point. Figure 2 shows the relationship between an image point and its corresponding ray of sight.

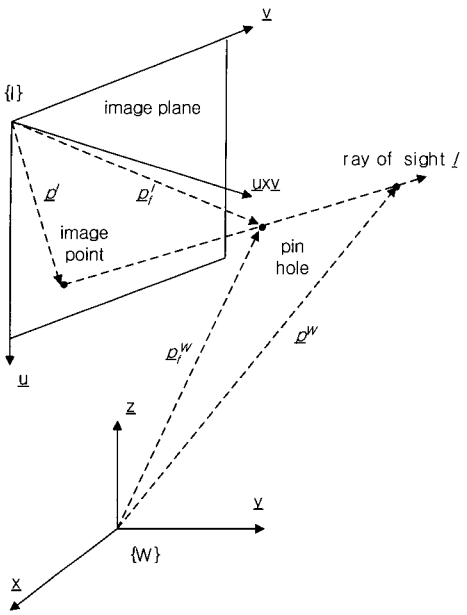

Figure 2. An image point and its ray of sight.

If we define a 3D frame {I} attached to the image plane as shown in the figure, an arbitrary image point can be represented as $\underline{p}^I=(u, v, 0)^T$ in {I}. When using the pin-hole camera model, a ray of sight from an image point can be uniquely determined as it passes the focal point $\underline{p}_f^I=(u_0, v_0, f)^T$ in {I}, which can also be represented as $\underline{p}_f^W=(p_{fx}, p_{fy}, p_{fz})^T$ in {W}. The aiming vector of the ray can then be defined as

$$\underline{a}^I=\underline{p}_f^I-\underline{p}^I \tag{4}$$

in {I}. This can be rewritten in {W} like

$$\underline{a}^W=\boldsymbol{R}_{WI}(\underline{p}_f^I-\underline{p}^I)=\boldsymbol{R}_{WI}\underline{a}^I \tag{5}$$

where $\boldsymbol{R}_{WI}=\begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}$ is a rotation matrix from {I} to {W}. As this ray should pass a 3D point $\underline{p}^W=(x, y, z)^T$ in {W}, that is projected onto the image point, the final equation defining a ray of sight from the t'th image point $\underline{p}_t^I$ becomes

$$\underline{p}_t^W=\underline{p}_f^W+s_t\boldsymbol{R}_{WI}(\underline{p}_f^I-\underline{p}_t^I) \tag{6}$$

where s_t is a scale factor representing the ratio between lengths of aiming vector and a vector to $\underline{p}_t^W$ from the pin-hole. This is not a constant like γ in eq.(4) but it needs not to be learned or exact in calibration process unlike γ as we try to find a ray rather than a point from a given image point. Actually it is just an arbitrary constant in a line equation specifying a 3D point on the line. The only condition we will imposed on s during the network learning is that it is a positive constant minimizing the distance between a 3D point and the ray from its image point.

Eq.(6) can be implemented by a neural network in the structure shown in figure 3. The outputs of the first and second hidden layers are the aiming vectors in {I} and {W}, $\underline{a}^I$ and $\underline{a}^W$, respectively. The output of the total network is the coordinate of a 3D point, that is on the ray of sight from the image point given.

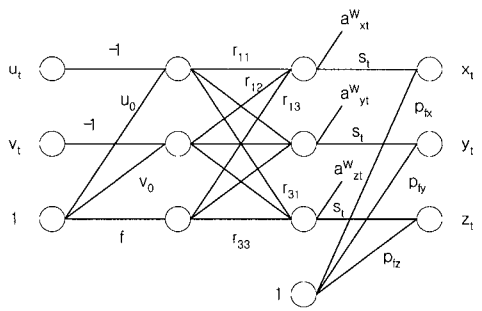

Figure 3. An MLP to learn a transformation from an image point to its ray of sight.

The network can be trained by the error back-propagation algorithm so that the error function of the following equation is minimized for N data given

$$E_t=\frac{1}{2}\sum_{n=1}^{3}(o_{nt}-p_{nt}^W)^2,\ t=1,\cdots,N \tag{7}$$

where o_{nt}, $n=1,\cdots,3$, are the outputs of the network for t'th data. Each parameters are modified iteratively to reduce the error function by

$$\frac{\partial E_t}{\partial p_{fn}^W} = o_{nt} - p_{nt}^W \tag{8.a}$$

$$\frac{\partial E_t}{\partial r_{mk}} = s_t(o_{mt} - p_{mt}^W)H_{kt}^{(1)}, \quad m=1,\cdots,3, \quad k=1,\cdots,3 \tag{8.b}$$

$$\frac{\partial E_t}{\partial w_h} = s_t \sum_{m=1}^{3}(o_{mt} - p_{mt}^W)r_{mh}, \quad h=1,\cdots,3 \tag{8.c}$$

where $H_{kt}^{(1)}$ is the *k'th* output of the first hidden layer for *t'th* datum and $w_1 = u_0$, $w_2 = v_0$, $w_3 = f$. The scaling factor s_t can be determined for the point by

$$s_t = (\underline{p_t^W} - \underline{p_f^W})^T \{(\underline{a_t^W})^T\}^\dagger \tag{9}$$

where $\dagger$ means pseudo inversion.

Since eq.(8.b) does not guarantee the orthornormality of the rotation matrix, we define the error terms, E_U and E_O, for the normality and orthogonality respectively like the below

$$E_{Uk} = \sum_{m=1}^{3} r_{mk}^2 - 1, \quad k=1,2 \tag{10.a}$$

$$E_O = r_{11}r_{12} + r_{21}r_{22} + r_{31}r_{32} \tag{10.b}$$

The first and second columns of the rotation matrix then can be adjusted to reduce the error terms like

$$\frac{\partial E_{orth}}{\partial r_{m1}} = 2E_{U1}r_{m1} + E_O r_{m2} \tag{11.a}$$

$$\frac{\partial E_{orth}}{\partial r_{m2}} = 2E_{U2}r_{m2} + E_O r_{m1} \tag{11.b}$$

where $E_{orth} = \frac{1}{2}(E_{U1}^2 + E_{U2}^2 + E_O^2)$. The third column, $\underline{r_3} = (r_{13}\ r_{23}\ r_{33})^T$, then can be determined from the two columns learned by

$$\underline{r_3} = \underline{r_1} \times \underline{r_2} \tag{12}$$

4. Results

The technique proposed was tested with synthetic and real data. The synthetic points were first generated randomly in space, whose positions were within the radius of 500[mm] from the center ray in approximately 1,000[mm] front of the camera assumed. The pin hole camera model was used then to computed corresponding image points. One hundred points were generated and half of them were used for calibration while those remained were used for testing the calibrated system. Zero mean random Gaussian noise with the variance of 0.5 pixel dimension was added onto the image coordinates of data used, which brought about errors of 0.64[pixel] in average.

After 10,000 epochs of learning using an MLP in the structure proposed, average errors of 0.49[pixel] and 0.67[pixel] were obtained for calibration and test data respectively by

$$projection\ error = (1/N)\sum_{t=1}^{N}\sqrt{(u_t - \hat{u}_t)^2 + (v_t - \hat{v}_t)^2} \tag{13}$$

where N is the number of points, $(\hat{u}_t, \hat{v}_t)$ are the estimated image coordinates by MLP for the real coordinates (u_t, v_t) of a t'th point. The errors resulted were quite small when comparing to 3.00[pixel] and 41.99[pixel] obtained by training the Ahmed's network for the same number of epochs. The real and estimated parameters were like table 1 below. To reach similar accuracy to that obtained by the MLP proposed, the Ahmed's network should be trained for more than 10 times of iterations in our test as shown in the table. The reason why testing error was considerably larger than calibration error when using Ahmed's network might be due to the fact that the number of parameters trained was larger than the data used for training as described previously. The learning rates used for both networks were experimentally set to the maxima which did not cause unstability in learning.

To check the sensitivity to the number of training data, the network structure proposed was trained with different number of data. Number of training data used were 20,40,60,80, and 100. After training with the data, the network was tested with 50 data that were not used for

training. The noise level of the data simulated was again zero mean and the variance of 0.5[pixel] to each dimension of the image point. The result obtained after 5,000 learning epochs was as shown in figure 4. We could get approximately the best accuracy if somewhat enough number of data were used as additional data did not significantly increase the accuracy. Similar result could be found in traditional calibration techniques as reported in [15].

Table 1. Example of network learning.

Parameters	Real values	Estimated by NN proposed	Estimated by Ahmed's network	
		10,000 epochs	10,000 epochs	100,000 epochs
d_x [mm]	-200.00	-197.54	-565.53	-159.15
d_y [mm]	500.00	498.47	661.60	481.56
d_z [mm]	2000.00	1998.80	2075.20	1990.60
r_{11}	0.612	0.617	0.978	0.727
r_{12}	0.047	0.052	-0.084	0.043
r_{13}	0.789	0.785	-0.002	0.685
r_{21}	0.612	0.609	-0.049	0.497
r_{22}	-0.660	-0.663	-0.603	-0.658
r_{23}	-0.436	-0.436	-0.794	-0.553
r_{31}	0.500	0.497	0.065	0.427
r_{32}	0.750	0.748	0.768	0.742
r_{33}	-0.433	-0.441	-0.587	-0.500
u_0 [pixel]	258.00	265.24	826.96	410.94
v_0 [pixel]	204.00	210.16	37.55	168.30
f [mm]	25.00	24.93	14.54	22.34

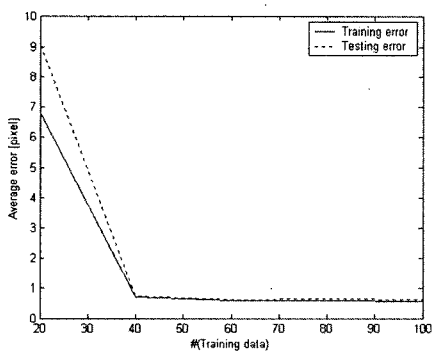

Figure 4. Effect of the number of training data to the calibration.

Figure 5 shows the result when different level of noise was added to the data used. The error of the proposed network was directly proportional to the error of data. This could be expected as the accuracy is limited by that of calibration data.

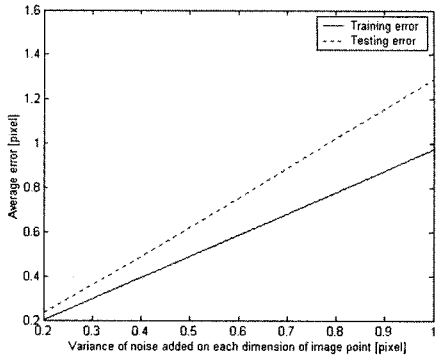

Figure 5. Effect of noise level to the calibration.

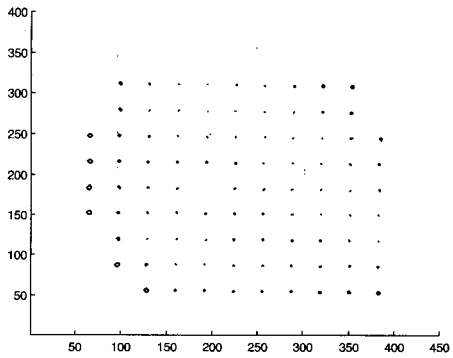

Figure 6. Error circles of projected points by training the MLP proposed using real data.

The MLP and its training algorithm proposed were tested with real data also. Three planes were placed at approximately 2,000[mm] in front of a camera and points on the planes were collected for calibration. Sixty points from the near and far planes each were used for training an MLP while ninety points from the center plane were used for testing. The interval between planes was about 242[mm]. We could obtain subpixel accuracy after 22,000 epochs. Figure 6 shows the error circles of the neural projection for the test points. The radius of a circle represents the distance between the real and

estimated image points. In the figure, the error circles of outer image points are larger than those of middle image points. This might be due to the lens distortion.

5. Conclusion

A new design of MLP is proposed for camera calibration. The proposed network finds camera parameters explicitly unlike most neural network based calibration techniques. The explicit calibration enables us to utilize the parameters obtained from back-projection in the application of projection or vice versa in a straightforward manner. Unlike Ahmed's network, which has some similarities to the network of this paper, the proposed network has a constant number of parameters and the learning is fast and accurate for moderate number of calibration data given as being proven in tests under various conditions. These advantages are mainly brought by relating image points to their rays of sight, that is an one-to-one mapping for each point while most existing techniques rely on a many-to-one projection mapping.

REFERENCES

[1] O. D. Faugueras and G. Toscani, "The calibration problem for stereoscopic vision," in *Sensor Devices and Systems for Robotics* (A.Casals, ed.), NATO ASI Series, Vol.F52, Springer-Verlag, Berlin, pp.195-213, 1989.

[2] Y. Do, *et al.*, "Calibrating stereoscopic 3D position measurement systems using artificial neural nets," *J. Korean Sensors Society*, Vol.7(6), pp.418-425, 1998.

[3] G-Q.Wei and S.D.Ma, "Implicit and explicit camera calibration: theory and experiments," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.16, No.5, pp.469-480, 1994.

[4] M. Ito, "Robot vision modelling - camera modelling and camera calibration," *Advanced Robotics*, Vol.5(3), pp.321-337, 1991.

[5] R. Y. Tsai, "A versatile camera calibration technique for high accuracy 3d machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE J.Robotics & Automation*, Vol. RA-3(4), pp.323-344, 1987.

[6] K. M. Hornik *et al.*, "Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks," *Neural Networks*, Vol.3, pp.551-560, 1990.

[7] J. Wen and G. Schweitzer, "Hybrid calibration of CCD cameras using artificial neural nets," in *Proc. Int. Joint Conf. Neural Networks*, pp.337-342, 1991.

[8] M. Kume and T.Kanade, "Camera system with neural network compensator for measuring 3-D position," *U.S. Patent*, No.5617490, 1997.

[9] D-H.Choi and S-Y.Oh, "Real-time neural network based camera localization and its extension to mobile robot control," *Int. J. Neural Systems*, Vol.8(3), pp.279-293, 1997.

[10] J. Jun and C.Kim, "Robust camera calibration using neural network," in *Proc. TENCON*, pp.694-697, 1999.

[11] Y. Do, "Application of neural networks for stereo-camera calibration," in *Proc. Int. Joint Conf. Neural Networks*, pp.2719-2722, 1999.

[12] J. Neubert, *et al.*, "Automatic training of a neural net for active stereo 3D reconstruction," in *Proc. IEEE Int. Conf. Robotics and Automation*, pp.2140-2146, 2001.

[13] M. Ahmed, *et. al.*, "A neural approach for single- and multi-image camera calibration," in *Proc. Int. Conf. Image Processing*, pp.925-929, 1999.

[14] S. Ganapathy, "Decomposition of transformation matrices for robot vision," in *Proc. IEEE Int. Conf. Robotics and Automation*, pp.130-139, 1984.

[15] H. Bacakoglu and M.Kamel, "A three-step camera calibration method," *IEEE Trans. Instrumentation and Measurement*, Vol.46 (5), pp.1165-1172, 1997.

著 者 紹 介

도 용 태
경북대학교 전자공학과(공학사)
서강대학교 전자공학과(공학석사)
Univ. of Hull(영) 전자공학과(공학박사)
CMU(미) Robotics Institute (객원부교수)
Univ. of Wisconsin(미), ECE (명예연구원)
현재 대구대학교 정보통신공학부 (부교수)
주관심 분야 : 자동화용 센서시스템 및 로봇시각