Open Journal of Applied Sciences, **2013, 3, 75-78**
Published Online March 2013 (http://www.scirp.org/journal/ojapps)

Modeling Camera Image Formation Using a Feedforward Neural Network

Yongtae Do
Electronic Control Major, Division of Electronic & Electrical Engineering, Daegu University, 712-714, South Korea
Email: ytdo@daegu.ac.kr

Received 2012

ABSTRACT

One fundamental problem in computer vision and image processing is modeling the image formation of a camera, *i.e.*, mapping a point in three-dimensional space to its projected position on the camera's image plane. If the relationship between the space and the image plane is assumed to be linear, the relationship can be expressed in terms of a transformation matrix and the matrix is often identified by regression. In this paper, we show that the space-to-image relationship in a camera can be modeled by a simple neural network. Unlike most other cases employing neural networks, the structure of the network is optimized so as for each link between neurons to have a physical meaning. This makes it possible to effectively initialize link weights and quickly train the network.

Keywords: Camera Model; Camera Calibration; Image Formation; Neural Network

1. Introduction

A camera can be considered as a device that records objects in three-dimensional (3D) space in the form of their two-dimensional (2D) images. In some technical fields where the use of a camera is required, such as computer vision and image processing, accurate and efficient modeling of the camera's image formation process is a basic problem that must be solved.

For a camera installed in a certain task, the image formation in the camera is characterized with the internal and external parameters of the camera [1]. The internal parameters include focal length, optical image center, and lens distortion coefficients, whereas the external parameters are those for specifying the geometric position and orientation of the camera. The camera model parameter determination process is called camera calibration [2]. Once a camera is calibrated, it is possible to computationally relate objects in 3D world and their projections on the camera's image plane.

Camera modeling and calibration have received great attention in photogrammetry, computer vision, machine vision, and image processing communities particularly since 1980s as cameras and computers became smaller, cheaper, more powerful, and easier to use than before thanks to the rapid technical advances in electronics. The most widely used method is mathematically estimating the parameters of a camera model that best relate control points in 3D world and their corresponding 2D image points in the model [3-5]. To increase the accuracy of camera calibration, control points must be collected evenly from the space viewed by the camera. However, it is difficult to make accurate position measurements of the 3D points. Methods of automatic calibration [6, 7] and using planar points [7, 8] have been proposed to overcome this difficulty. Existing camera modeling and calibration techniques are well reviewed in [9, 10].

The relationship between the coordinates of a 3D point and the coordinates of its corresponding 2D image point is expressed in terms of a 3×4 matrix when the relationship in a camera is assumed linear. The elements of this transformation matrix can be determined by a regression technique using six or more control points and their image points.

In this paper, we show that the relationship between 3D points and their 2D images can be expressed by a neural network (NN). The model parameter can then be learned by training the NN. The proposed method is quite different from most existing NN-based methods for camera calibration, where NNs are usually used for identifying unknown parts which are not accommodated in a camera model. For example, in [11], an NN is used for learning camera's nonlinearity after linear parameter estimation. The nonlinearity is mostly due to lens distortion [12]. If the linear NN model of this paper is combined with an existing NN for learning nonlinearity, a complete camera model can be constructed with only NNs.

2. Image Formation Model

2.1. Pin-hole Model

Pin-hole camera model is widely used to relate the image coordinates of an object point visible by a camera and the coordinates of the point in the world coordinate system by distortion-free linear mapping [1, 2]. All rays of sight from 3D points in a scene are assumed to pass one particular spatial point, pin-hole, in the model. **Figure 1** shows the pin-hole camera model, where the following relationships are assumed

$$u = i + i_O, \quad v = j + j_O, \tag{1}$$

$$P_C = RP + T \tag{2}$$

for a 3D point $\boldsymbol{P} = [x\ y\ z]^{\mathrm{T}}$ in the world coordinate system {**W**}, its corresponding representation $\boldsymbol{P_C} = [x_C\ y_C\ z_C]^{\mathrm{T}}$ in the 3D camera coordinate system {**C**}, the projected point at $[u,\ v]^{\mathrm{T}}$ on the 2D image plane, and the optical image center at $[i_O\ j_O]^{\mathrm{T}}$ in the row-column image frame {**U**}. A 3D point in {**W**} can be transformed to the representation in {**C**} by a 3×3 rotation matrix $\boldsymbol{R}$ and a translation vector $\boldsymbol{T}$.

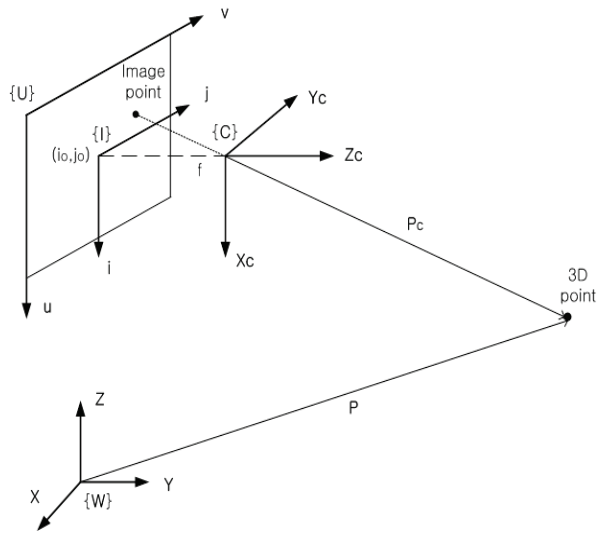

Figure 1. Pin-hole camera model.

The coordinates of an image point are computed in the model from the 3D coordinates in {**C**} by

$$i = -f\,x_C / z_C, \quad j = -f\,y_C / z_C, \tag{3}$$

where f is the focal length. Combining above equations leads us to the following equation

$$\begin{pmatrix} su \\ sv \\ s \end{pmatrix} = \begin{pmatrix} -f & 0 & i_o \\ 0 & -f & j_o \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} R & T \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \tag{4}$$

2.2. Neural Network Implementation

A feedforward neural network is capable of computing output values from given input values by propagating weighted values through links between neurons. We want to design an NN as shown in **Figure 2** that can represent the image formation process described in Section 2.1. However, it is not possible to build a network in this structure directly from Equation (4) due to the scale factor s, which is the coordinate z_C of a 3D point. Instead, Equation (4) leads us to a structure shown in **Figure 3**. **Figure 4** is a practical network implementation of **Figure 3**.

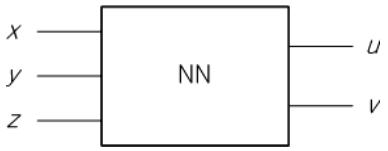

Figure 2. Image formation model by a neural network.

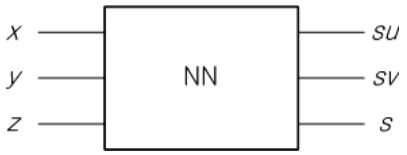

Figure 3. NN built from pinhole camera model.

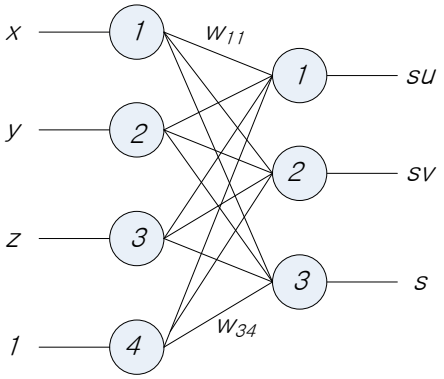

Figure 4. Implementation of the NN of Figure 3.

Like most other NNs and their applications, the key issue of the NN implementation presented in **Figure 4** is determining the weight of each link between neurons. From Equation (4), the physical meaning of w_{nm}, a link weight from neuron m to neuron n, can be specified as

$$\begin{aligned} w_{11} &= -fr_{11} + i_O r_{31}, \quad w_{12} = -fr_{12} + i_O r_{32}, \\ w_{13} &= -fr_{13} + i_O r_{33}, \quad w_{14} = -ft_1 + i_O t_3, \\ w_{21} &= -fr_{21} + j_O r_{31}, \quad w_{22} = -fr_{22} + j_O r_{32}, \\ w_{23} &= -fr_{23} + j_O r_{33}, \quad w_{24} = -ft_2 + j_O t_3, \\ w_{31} &= r_{31}, \quad w_{32} = r_{32}, \quad w_{33} = r_{33}, \quad w_{34} = t_3, \end{aligned} \tag{5}$$

where r_{pq} are elements of rotation matrix $\boldsymbol{R}$, and t_p are elements of translation vector $\boldsymbol{T}$; $1 \le p, q \le 3$.

The network shown in **Figure 4** has a quite simple structure. However, it is not simple to train the NN because we do not know the scale factor s for a given 3D point $\boldsymbol{P}$. We know only the projected image coordinates u and v for a control point $\boldsymbol{P}$. If the desired output is not available, it is not possible to train the network using a supervised learning algorithm, such as gradient descent optimization [13]. We thus need to develop a method to train the network in the structure of **Figure 4**.

An error function is defined as

$$E = (e_1^2 + e_2^2 + e_3^2)/2 \tag{6}$$

where $e_1 = o_1/o_3 - u$, $e_2 = o_2/o_3 - v$, $e_3 = ((o_1 o_2)/(uv))^{1/2} - o_3$ for three computed output neuron values, o_1, o_2, and o_3. Note that the error term of the 3rd output neuron, e_3, is derived from

$$o_1 o_2 = (su)(sv) = s^2 uv. \tag{7}$$

Then, the weights are trained by gradient descent. For a weight w_{nm}, n=1 or 2, as shown in **Figure 5**, a chain rule is applied to the given error E as

$$\frac{\partial E}{\partial w_{nm}} = \frac{\partial E}{\partial e_n}\frac{\partial e_n}{\partial o_n}\frac{\partial o_n}{\partial g_n}\frac{\partial g_n}{\partial w_{nm}} \tag{8}$$

where, assuming a linear activation function for output neurons, $\partial E / \partial e_n = e_n$, $\partial e_n / \partial o_n = 1/o_3$, $\partial o_n / \partial g_n = 1$, $\partial g_n / \partial w_{nm} = o_n$.

For the case of n=3, on the other hand, the following equation can be obtained by gradient descent,

$$\partial E / \partial w_{3m} = -e_3 o_m \tag{9}$$

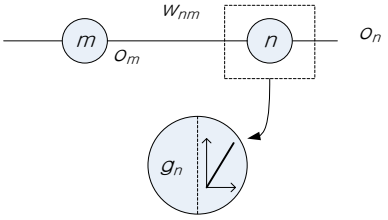

Figure 5. Connection between an input neuron *m* and an output neuron *n*.

3. Numerical Example

A camera is assumed to be located at x=–200, y=500, z=2000 and oriented by Z-Y-X Euler angles of θ_z=45°, θ_y=–30° and θ_x=120° in the world coordinate system {**W**}. It is also assumed that the focal length is f=25, the coordinates of the optical image center is (258, 204), and the dimension of a pixel is 0.023×0.023. This camera setup is drawn in **Figure 6**. An NN can then be built to express the image formation process of the camera as presented in **Figure 7**.

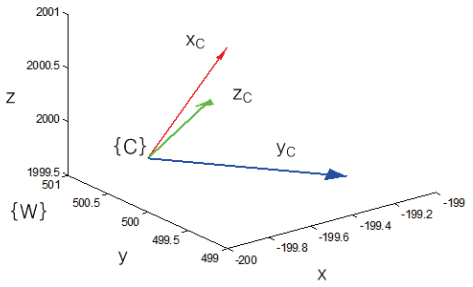

Figure 6. Camera setup assumed as an example.

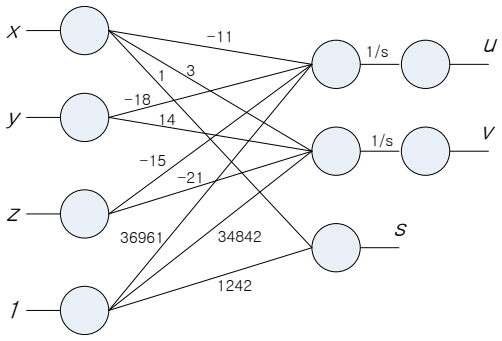

Figure 7. Neural network resulted from the camera setup.

4. Concluding Remarks

We have shown that a feedforward neural network can be

constructed to express the image formation process of a camera. The network constructed in this paper is in a quite simple structure with four input neurons and three output neurons of linear activation functions. Although most existing applications of NNs to camera modeling have focused on nonlinear lens distortion problem, the network of this paper models the linear perspective transformation. A method to learn the link weights between neurons of the proposed network is also described. The entire image formation of a camera may be modeled accurately if the proposed network is combined with an existing NN-based method developed for correcting lens distortion.

5. Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(2012R1A1A4A01010160).

REFERENCES

[1] R. Szeliski, “Computer Vision, Algorithms and Applications,” Springer-Verlag, London, 2011.

[2] R. Y. Tsai, "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*, Vol. 3, No. 4, 1987, pp. 323-344. doi:10.1109/JRA.1987.1087109

[3] K. Nakano, M. Okutomi and Y. Hasegawa, "Camera Calibration with Precise Extraction of Feature Points Using Projective Transformation," *Proceedings of IEEE International Conference on Robotics and Automation*, Vol. 3, 2002, pp. 2532-2538. doi:10.1109/ROBOT.2002.1013612

[4] J. Weng, P. Cohen, M. Herniou, "Camera Calibration with Distortion Models and Accuracy Evaluation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, Issue. 10, pp. 965-980, 1992. doi:10.1109/34.159901

[5] K. D. Gremban, C. E. Thorpe, T. Kanade, "Geometric Camera Calibration Using Systems of Linear Equations," *Proceedings of IEEE Conference on Robotics and Automation, 1988*, pp. 562-567. doi:10.1109/ROBOT.1988.12111

[6] R. Hartley and A. Zisserman, “Multiple View Geometry in Computer Vision”, Cambridge University Press, 2000.

[7] B. Triggs, “Autocalibration from Planar Scenes,” *Proceedings of European Conference on Computer Vision*, 1998, pp. 89-105.

[8] Z. Zhang, “A Flexible New Technique for Camera Calibration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, Issue. 11, 2000, pp. 1330-1334. doi:10.1109/34.888718

[9] Q. Wang, L. Fu and Z. Liu, “Review on Camera Calibration,” *Proceedings of Chinese Control and Decision Conference*, 2010, pp. 3354-3358.

[10] J. Salvi, X. Armangué and J. Batlle, “A Comparative Review of Camera Calibrating Methods with Accuracy Evaluation,” *Pattern Recognition*, Vol. 35, Issue 7, 2002, pp. 1617-1635. doi:10.1016/S0031-3203(01)00126-1

[11] X. Chen, H. Fang, Y. Yang and S. Qin, “The Research of Camera Distortion Correction Basing on Neural Network,” *Proceedings of Chinese Control and Decision Conference*, 2011, pp. 596-601.

[12] J. – P. Tardif, P. Sturm, M. Trudeau and S. Roy, “Calibration of Cameras with Radially Symmetric Distortion,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 31, Issue 9, 2009, pp. 1552-1566. doi:10.1109/TPAMI.2008.202

[13] C. M. Christopher, “Pattern Recognition and Machine Learning”, Springer Science + Business Media, New York, 2006.

