

Binocular vision calibration and 3D re-construction with an orthogonal learning neural network

Dong-yuan Ge^{1,2}  · Xi-fan Yao³ · Zhao-tong Lian²

Received: 18 April 2015 / Revised: 28 June 2015 / Accepted: 26 July 2015 /

Published online: 16 August 2015

© Springer Science+Business Media New York 2015

Abstract A new approach for binocular vision system calibration and 3D re-construction is proposed. While the system is calibrated, the sum of square distances between the vector coordinates recombined with the coordinates of feature points in the world frame and those in image frame to the fitted hyperplane is taken as an objective function. An orthogonal learning neural network is designed, where a self-adaptive minor component extracting method is adopted. When the network comes to equilibrium, the projective matrixes for the two cameras are obtained from the eigen-vectors of the autocorrelation matrix corresponding to the minimum eigen values, so the calibration of the binocular vision system is achieved. As for 3D re-construction, an autocorrelation matrix is obtained from feature point coordinates in image planes and calibration data, and an orthogonal learning network is designed. After the network is trained, the autocorrelation matrix's eigen-vector corresponding to the minimum eigen-values is obtained, from which the 3D coordinates are obtained also. The proposed approach is a novel application of minor component analysis and orthogonal learning network in binocular vision system and 3D re-construction.

Keywords Binocular vision · System calibration · Eigen-value · Eigen-vector · Orthogonal learning · State transition matrix

✉ Dong-yuan Ge
gordon399@163.com

Xi-fan Yao
mexfyao@scut.edu.cn

Zhao-tong Lian
lianzt@umac.mo

¹ Department of Mechanical and Electrical Engineering, Tan Kah Kee College, Xiamen University, Zhangzhou, Fujian Province 363105, People's Republic of China

² Faculty of Business Administration, University of Macau, Macau, SAR, People's Republic of China

³ School of Mechanical and Automotive Engineering, South China University of Technology, Guangzhou, Guangdong Province 510640, People's Republic of China

1 Introduction

Camera calibration is a process of determining the internal camera geometric and optical characteristics (intrinsic parameters) and/or the 3D position and orientation of the camera frame relative to a certain world frame (extrinsic parameters) [6, 12]. The applications of vision system include 3D sensing and measurement, precision manufacturing, automated assembly, monitoring and tracking etc. While a binocular vision system is calibrated, the scene's 3D geometric information can be reconstructed by obtaining two digital images from different angles. There are many research reports on the camera calibration. For example, MA obtained a intrinsic parameters of the camera by designing two sets of three pure orthogonal translation movements, and the orientations of the camera with respect to the hand frame with a set pairwise orthogonal translation movements [11]. ZHANG obtained the intrinsic and extrinsic parameters via the homography matrix in the light of orthogonality of the rotational matrix with the homography obtained from the 3D feature coordinates in a target block and its 2D coordinates. In general, the calculated rotational matrix did not satisfy the orthogonal properties well [19]. Rahman and Krouglicof proposed the quaternion representation of spatial orientation, which resulted in a system of equations that was minimally redundant and free of singularities, and applied a technique to minimize the error between the reconstructed image points and their experimentally determined counterparts in the “distortion free” space, so the technique facilitated the incorporation of the exact lens distortion model as opposed to that relying on an approximation one [15]. Recently, we have proposed a method for camera calibration with an adaptive principal component extraction network in which the sum of square distances from the vector coordinates of feature points to those in hyperplane is taken as objective function, and the eigen-vector of autocorrelation matrix corresponding to minimal eigen-values as a projective matrix. But the calibration of binocular vision system and 3D reconstruction has not been done [5]. Chen presented a novel method to analyze the blur distribution in an image and found the optimal focusing distance so that additional constraints could be used to generate absolute measurement of the models [4]. ZHENG proposed a minimum calibration condition that consisted of two vanishing points and a vanishing line so as to estimate camera intrinsic parameters (including the principal point coordinates) and rotation angles, which adopted least squares optimization instead of closed-form computation. The proposed method was practical and suitable for more traffic scenes while roadside camera is calibrated [21]. YIN presented a semi-automatic scene calibration method that combined tracked blobs with user-selected line scene features to recover the homographies between camera views, so the system could deal with mapping a network of cameras with overlapped fields of view into a single ground plane view, even when the overlap was not substantial [17]. There is no similar work done on the binocular vision system calibration and 3D reconstruction by means of minor components analysis and the adaptive orthogonal learning network. Therefore, based on our previous work on neuro-calibration techniques, in this study we put forward a novel method where a self-adaptive orthogonal learning network is used to achieve calibration for the binocular vision system and 3D measurement.

2 Model of the binocular vision system

In the binocular vision system as shown in Fig. 1, camera frames are C_1 and C_2 , $o_1u_1v_1$ and $o_2u_2v_2$ are the image coordinate systems measured in pixels, $O_wX_wY_wZ_w$ is the world frame

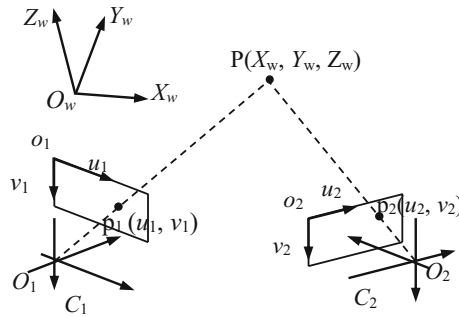


Fig. 1 Binocular vision system

measured in mm, the homogeneous coordinates of feature point P in the world frame are $(X_{wi}, Y_{wi}, Z_{wi}, 1)$, which is projected into the image planes so p_1 and p_2 are obtained, and their homogeneous coordinates are $(u_{1i}, v_{1i}, 1)$ and $(u_{2i}, v_{2i}, 1)$ respectively. The projective matrixes of the left and right cameras are M_1 and M_2 respectively, and the transformation relations of $o_1u_1v_1$ or $o_2u_2v_2$ and $O_wX_wY_wZ_w$ can be described as follows:

$$Z_{ci}^{(1)} \begin{bmatrix} u_{1i} \\ v_{1i} \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11}^{(1)} & m_{12}^{(1)} & m_{13}^{(1)} & m_{14}^{(1)} \\ m_{21}^{(1)} & m_{22}^{(1)} & m_{23}^{(1)} & m_{24}^{(1)} \\ m_{31}^{(1)} & m_{32}^{(1)} & m_{33}^{(1)} & m_{34}^{(1)} \end{bmatrix} \begin{bmatrix} X_{wi} \\ Y_{wi} \\ Z_{wi} \\ 1 \end{bmatrix} \tag{1}$$

$$Z_{ci}^{(2)} \begin{bmatrix} u_{2i} \\ v_{2i} \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11}^{(2)} & m_{12}^{(2)} & m_{13}^{(2)} & m_{14}^{(2)} \\ m_{21}^{(2)} & m_{22}^{(2)} & m_{23}^{(2)} & m_{24}^{(2)} \\ m_{31}^{(2)} & m_{32}^{(2)} & m_{33}^{(2)} & m_{34}^{(2)} \end{bmatrix} \begin{bmatrix} X_{wi} \\ Y_{wi} \\ Z_{wi} \\ 1 \end{bmatrix} \tag{2}$$

where $m_{11}^{(1)}, \dots, m_{34}^{(1)}, m_{11}^{(2)}, \dots, m_{34}^{(2)}$ are elements of projections matrixes of the left and right cameras.

If $Z_{ci}^{(1)}$ and $Z_{ci}^{(2)}$ in Eqs. (1) and (2) are cancelled respectively, then we can obtain

$$X_{wi}m_{11}^{(1)} + Y_{wi}m_{12}^{(1)} + Z_{wi}m_{13}^{(1)} + m_{14}^{(1)} - u_{1i}X_{wi}m_{31}^{(1)} - u_{1i}Y_{wi}m_{32}^{(1)} - u_{1i}Z_{wi}m_{33}^{(1)} - u_{1i}m_{34}^{(1)} = 0 \tag{3}$$

$$X_{wi}m_{21}^{(1)} + Y_{wi}m_{22}^{(1)} + Z_{wi}m_{23}^{(1)} + m_{24}^{(1)} - v_{1i}X_{wi}m_{31}^{(1)} - v_{1i}Y_{wi}m_{32}^{(1)} - v_{1i}Z_{wi}m_{33}^{(1)} - v_{1i}m_{34}^{(1)} = 0 \tag{4}$$

$$X_{wi}m_{11}^{(2)} + Y_{wi}m_{12}^{(2)} + Z_{wi}m_{13}^{(2)} + m_{14}^{(2)} - u_{2i}X_{wi}m_{31}^{(2)} - u_{2i}Y_{wi}m_{32}^{(2)} - u_{2i}Z_{wi}m_{33}^{(2)} - u_{2i}m_{34}^{(2)} = 0 \tag{5}$$

$$X_{wi}m_{21}^{(2)} + Y_{wi}m_{22}^{(2)} + Z_{wi}m_{23}^{(2)} + m_{24}^{(2)} - v_{2i}X_{wi}m_{31}^{(2)} - v_{2i}Y_{wi}m_{32}^{(2)} - v_{2i}Z_{wi}m_{33}^{(2)} - v_{2i}m_{34}^{(2)} = 0 \tag{6}$$

At the same time, Eqs. (3), (4), (5) and (6) can be divided by $-u_{1i}, -v_{1i}, -u_{2i}$ and $-v_{2i}$ respectively, which don't change the transformation relation of two sides in Eqs. (3)–(6). While the binocular vision system is calibrated, if n feature points' coordinates in world frame

and in image frames are obtained, a linear equation can be obtained according to Eqs. (3) and (4), that is

$$\mathbf{A}_1 \mathbf{n}_1 = \mathbf{0}^{(1)} \tag{7}$$

where

$$\mathbf{A}_1 = \begin{bmatrix} -X_{w1}/u_{11} & -Y_{w1}/u_{11} & -Z_{w1}/u_{11} & -1/u_{11} & 0 & 0 & 0 & 0 & X_{w1} & Y_{w1} & Z_{w1} & 1 \\ 0 & 0 & 0 & 0 & -X_{w1}/v_{11} & -Y_{w1}/v_{11} & -Z_{w1}/v_{11} & -1/v_{11} & X_{w1} & Y_{w1} & Z_{w1} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -X_{wn}/u_{1n} & -Y_{wn}/u_{1n} & -Z_{wn}/u_{1n} & -1/u_{1n} & 0 & 0 & 0 & 0 & X_{wn} & Y_{wn} & Z_{wn} & 1 \\ 0 & 0 & 0 & 0 & -X_{wn}/v_{1n} & -Y_{wn}/v_{1n} & -Z_{wn}/v_{1n} & -1/v_{1n} & X_{wn} & Y_{wn} & Z_{wn} & 1 \end{bmatrix},$$

which is a $2n \times 12$ matrix; \mathbf{n}_1 is a column vector, consisting of $m_{11}^{(1)}, \dots, m_{14}^{(1)}, m_{21}^{(1)}, \dots, m_{24}^{(1)}, m_{31}^{(1)}, \dots, m_{34}^{(1)}$; and $\mathbf{0}^{(1)}$ is a vector consisting of 12 constants of 0.

As for the right camera, a similar equation like Eq. (7) can be obtained from Eqs. (5) and (6), that is

$$\mathbf{A}_2 \mathbf{n}_2 = \mathbf{0}^{(2)} \tag{8}$$

According to Eqs. (7) and (8), we can get an overdetermined equation, that is

$$\mathbf{A} \mathbf{n} = \mathbf{0} \tag{9}$$

where $\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{0}_1 \\ \mathbf{0}_2 & \mathbf{A}_2 \end{bmatrix}$, which is a $4n \times 24$ matrix; $\mathbf{n} = [\mathbf{n}_1, \mathbf{n}_2]^T$, which is a column vector consisting of elements of \mathbf{n}_1 and \mathbf{n}_2 , and $\mathbf{0}_1$ and $\mathbf{0}_2$ are zero matrixes with 12×12 .

3 Minor component analysis and solving algorithm

While the binocular vision system is calibrated, the 22 elements of the fitting projective matrixes except $m_{34}^{(1)}$ and $m_{34}^{(2)}$ in Eq. (9) are taken as the coefficients of the hyper-plane, which constitute the fitting vector $\mathbf{m} = [m_{11}^{(1)}, \dots, m_{33}^{(1)}, m_{11}^{(2)}, \dots, m_{33}^{(2)}]^T$. The coordinates of sampled points in the world frame and in the image frame are transformed to vector points \mathbf{x}_i , so the algorithm we adopted is to minimize the sum of the squared distances between all the vector points (i.e. combination coordinates) \mathbf{x}_i and fitting hyper-plane, thus the objective function is

$$\min_{\mathbf{m}} E(\mathbf{m}) = \sum_{i=1}^N \sum_{j=4i-3}^{4i} e_j^2 \tag{10}$$

where $e_j = \mathbf{m}^T \mathbf{x}_j / \|\mathbf{m}\|_2$.

Let $\mathbf{x}_{4i-3} = [-X_{wi}/u_{1i} \quad -Y_{wi}/u_{1i} \quad -Z_{wi}/u_{1i} \quad -1/u_{1i} \quad \mathbf{0}^{(1)} \quad X_{wi} \quad Y_{wi} \quad Z_{wi} \quad \mathbf{0}]^T$, $\mathbf{x}_{4i-2} = [\mathbf{0}^{(1)} \quad -X_{wi}/v_{1i} \quad -Y_{wi}/v_{1i} \quad -Z_{wi}/v_{1i} \quad -1/v_{1i} \quad X_{wi} \quad Y_{wi} \quad Z_{wi} \quad \mathbf{0}]^T$, $\mathbf{x}_{4i-1} = [\mathbf{0} \quad -X_{wi}/u_{2i} \quad -Y_{wi}/u_{2i} \quad -Z_{wi}/u_{2i} \quad -1/u_{2i} \quad \mathbf{0}^{(1)} \quad X_{wi} \quad Y_{wi} \quad Z_{wi}]^T$, $\mathbf{x}_{4i} = [\mathbf{0} \quad \mathbf{0}^{(1)} \quad -X_{wi}/v_{2i} \quad -Y_{wi}/v_{2i} \quad -Z_{wi}/v_{2i} \quad -1/v_{2i} \quad X_{wi} \quad Y_{wi} \quad Z_{wi}]^T$. where $\mathbf{0}$ is a 1×11 row vector, and $\mathbf{0}^{(1)}$ is a 1×4 row vector. Thus

$$E(\mathbf{m}) = \frac{\sum_{i=1}^N \sum_{j=4i-3}^{4i-2} (\mathbf{m}^T \mathbf{x}_j + m_{34}^{(1)})^2 + \sum_{j=4i-1}^{4i} (\mathbf{m}^T \mathbf{x}_j + m_{34}^{(2)})^2}{\|\mathbf{m}\|_2^2} = \frac{\mathbf{m}^T \mathbf{R} \mathbf{m} + 2m_{34}^{(1)} \mathbf{m}^T \mathbf{b}_1 + 2m_{34}^{(2)} \mathbf{m}^T \mathbf{b}_2 + 2N \left((m_{34}^{(1)})^2 + (m_{34}^{(2)})^2 \right)}{\|\mathbf{m}\|_2^2} \tag{11}$$

where $\mathbf{R} = \sum_{i=1}^N \sum_{j=4i-3}^{4i-2} \mathbf{x}_j \mathbf{x}_j^T + \sum_{i=1}^N \sum_{j=4i-1}^{4i} \mathbf{x}_j \mathbf{x}_j^T$, which can be written as $[\mathbf{R}_1 + \mathbf{R}_2]$; \mathbf{R}_1 and \mathbf{R}_2 are 22×22 real symmetric matrixes, i.e., $\mathbf{R}_1 = \sum_{i=1}^N \sum_{j=4i-3}^{4i-2} \mathbf{x}_j \mathbf{x}_j^T$, and $\mathbf{R}_2 = \sum_{i=1}^N \sum_{j=4i-1}^{4i} \mathbf{x}_j \mathbf{x}_j^T$; $\mathbf{b}_1 = \sum_{i=1}^N \sum_{j=4i-3}^{4i-2} \mathbf{x}_j$; and $\mathbf{b}_2 = \sum_{i=1}^N \sum_{j=4i-1}^{4i} \mathbf{x}_j$.

In order to get the minimum, the critical point can be obtained from $dE/d\mathbf{m} = \mathbf{0}$, that is

$$\left(\mathbf{R} + m_{34}^{(1)} \mathbf{b}_1 + m_{34}^{(2)} \mathbf{b}_2 \right) \mathbf{m} - \frac{\mathbf{m}^T \mathbf{R} \mathbf{m} + 2m_{34}^{(1)} \mathbf{m}^T \mathbf{b}_1 + 2m_{34}^{(2)} \mathbf{m}^T \mathbf{b}_2 + 2N \left(m_{34}^{(1)} + m_{34}^{(2)} \right)^2}{\|\mathbf{m}\|_2^2} \mathbf{m} = \mathbf{0} \quad (12)$$

Let $\lambda = \frac{\mathbf{m}^T \mathbf{R} \mathbf{m} + 2m_{34}^{(1)} \mathbf{m}^T \mathbf{b}_1 + 2m_{34}^{(2)} \mathbf{m}^T \mathbf{b}_2 + 2N \left(m_{34}^{(1)} + m_{34}^{(2)} \right)^2}{\|\mathbf{m}\|_2^2}$. According to the expected values of $m_{34}^{(1)}$ and $m_{34}^{(2)}$ from Eqs. (7) and (8). And assuming $m_{34}^{(1)} = -\mathbf{m}^T \mathbf{b}_1 / 2N$ and $m_{34}^{(2)} = -\mathbf{m}^T \mathbf{b}_2 / 2N$, so we have

$$\mathbf{M} \mathbf{m} - \lambda \mathbf{m} = \mathbf{0} \quad (13)$$

where $\mathbf{M} = \mathbf{R} - \mathbf{B}$, $\lambda = \frac{\mathbf{m}^T \mathbf{M} \mathbf{m}}{\mathbf{m}^T \mathbf{m}}$, $\mathbf{B} = [\mathbf{B}_1 \ \mathbf{B}_2]$, $\mathbf{B}_1 = \mathbf{b}_1 \mathbf{b}_1^T / 2N$, $\mathbf{B}_2 = \mathbf{b}_2 \mathbf{b}_2^T / 2N$, and $\mathbf{M} = \begin{bmatrix} \mathbf{M}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_2 \end{bmatrix}$. Thus λ is the eigen-value of \mathbf{M} , and \mathbf{m} is its corresponding eigen-vector [1].

At the same time, assuming eigen values of \mathbf{M}_1 and \mathbf{M}_2 to be $\lambda_1, \lambda_2, \dots$, and $\lambda_{11}, \mu_1, \mu_2, \dots, \mu_{11}$, respectively, so there are orthogonal matrixes \mathbf{P} and \mathbf{Q} , which meet $\mathbf{M}_1 = \mathbf{P} \mathbf{\Lambda}_1 \mathbf{P}^{-1}$, and $\mathbf{M}_2 = \mathbf{Q} \mathbf{\Lambda}_2 \mathbf{Q}^{-1}$, where $\mathbf{\Lambda}_1 = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_{11})$, $\mathbf{\Lambda}_2 = \text{diag}(\mu_1, \mu_2, \dots, \mu_{11})$. Thus

$$\mathbf{M} = \begin{bmatrix} \mathbf{P} \mathbf{\Lambda}_1 \mathbf{P}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q} \mathbf{\Lambda}_2 \mathbf{Q}^{-1} \end{bmatrix} = \begin{bmatrix} \mathbf{P} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q} \end{bmatrix} \mathbf{\Lambda} \begin{bmatrix} \mathbf{P} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q} \end{bmatrix}^{-1} \quad (14)$$

As for the eigen values of \mathbf{M} , we have sorted them in order from large to small. For example $\mathbf{\Lambda} = \text{diag}(\mu_1 \ \lambda_1 \ \dots \ \lambda_{10} \ \lambda_{11} \ \mu_{11})$, or $\mathbf{\Lambda} = \text{diag}(\lambda_1 \ \mu_1 \ \dots \ \mu_{11} \ \lambda_{10} \ \lambda_{11})$, and so on. Thus \mathbf{M} can be described as $\mathbf{M} = \mathbf{B} \mathbf{\Lambda} \mathbf{B}^{-1}$, where \mathbf{B} consists of 22 column vectors such as $[\mathbf{m}_1, \mathbf{0}]^T, [\mathbf{0}, \mathbf{m}_2]^T, [\mathbf{m}_3, \mathbf{0}]^T, \dots, [\mathbf{0}, \mathbf{m}_j]^T, \dots, [\mathbf{m}_{22}, \mathbf{0}]^T$, which is an orthogonal eigen vector of \mathbf{M} corresponding to the eigen values respectively, and the projective matrixes of binocular vision are obtained from the normalization eigenvectors of the \mathbf{M} corresponding to the minimum eigen values.

Due to the projective matrices of the left and right cameras being different in binocular vision system, if the left camera's projective matrix is obtained from the eigen vector $\mathbf{m} = [\mathbf{m}_{22}, \mathbf{0}]^T$ of the autocorrelation matrices corresponding to the minimal eigen value λ_{11} , so the right camera's projective matrix can be obtained from the eigen vector $\mathbf{m} = [\mathbf{0}, \mathbf{m}_{21}]^T$ corresponding to the minimal eigen value μ_{11} , where $\mathbf{0}$ is a 1×11 matrix. And $m_{34}^{(1)} = -\sum_{i=1}^N \sum_{j=4i-3}^{4i-2} \mathbf{m}_{22}^T \mathbf{x}_j / 2N$, $m_{34}^{(2)} = -\sum_{i=1}^N \sum_{j=4i-1}^{4i} \mathbf{m}_{21}^T \mathbf{x}_j / 2N$, where $i = 1, 2, \dots, N$. Thus the projective matrix of the left camera is $\mathbf{m}_L = [\mathbf{m}_{22}, m_{34}^{(1)}]^T$, and that for the right camera is $\mathbf{m}_R = [\mathbf{m}_{21}, m_{34}^{(2)}]^T$.

4 Design of the self-adaptive orthogonal learning neural network

An orthogonal learning neural network with a lateral connection proposed by KUNG [10] was adopted in the experiments. The structure is shown in Fig. 2, and its input data are the row vectors of the auto-correlation matrix \mathbf{M} . There are 22 neurons in the output layer, and the 1st neuron connects with input neurons with $\mathbf{m}_1=[m_1^{(1)}, m_1^{(2)}, \dots, m_1^{(22)}]^T$, but without a lateral connection while trained. The j^{th} neuron connects with both the input neurons as $\mathbf{m}_j=[m_j^{(1)}, m_j^{(2)}, \dots, m_j^{(22)}]^T$, and the front $(j-1)$ outputs as the lateral weight vector $\mathbf{W}_j=[w_j^{(1)}, w_j^{(2)}, \dots, w_j^{(j-1)}]^T$. The network is trained out step by step. While the j^{th} neuron is training, the 1st, 2nd, ..., and $(j-1)^{\text{th}}$ neurons have already been trained, i.e. stable values $\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_{j-1}$ are obtained, all of which are orthogonal each other. When the j^{th} neuron is trained completely, the lateral connection weights approximate 0, and \mathbf{m}_j is perpendicular to $\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_{j-1}$ respectively [13].

The output of the 1st neuron is

$$O_1 = \mathbf{m}_1^T \mathbf{M}_i \tag{15}$$

Due to the fact that there is no lateral connection, the learning algorithm for the 1st neuron is as following,

$$\Delta \mathbf{m}_1 = \beta \left(O_1 \mathbf{M}_i - \frac{O_1^2}{\mathbf{m}_1^T \mathbf{m}_1} \mathbf{m}_1 \right) \tag{16}$$

The 1st term of Eq. (16) is that for Hebbian learning rule, which represents a self-strengthening function. When the network comes to the stable state, i.e. $\Delta \mathbf{m}_1 \rightarrow \mathbf{0}$, we have $O_1 \mathbf{M}_i - \frac{O_1^2}{\mathbf{m}_1^T \mathbf{m}_1} \mathbf{m}_1 = \mathbf{0}$. According to Eqs. (15) and (16), $\mathbf{M} \mathbf{m}_i - \lambda_1 \mathbf{m}_1 = \mathbf{0}$, and $\lambda_1 = \frac{\mathbf{m}_1^T \mathbf{M} \mathbf{m}_1}{\mathbf{m}_1^T \mathbf{m}_1}$, where \mathbf{m}_1 is the eigen-vector of the self-correlation matrix \mathbf{M} , which is corresponding to the max eigen value λ_1 .

The training for the j^{th} neuron is similar to the above method, i.e.

$$\mathbf{O}_j = \mathbf{V}_j \mathbf{M}_i \tag{17}$$

$$O_j = \mathbf{m}_j^T \mathbf{M}_i + \mathbf{W}_j^T \mathbf{O}_j \tag{18}$$

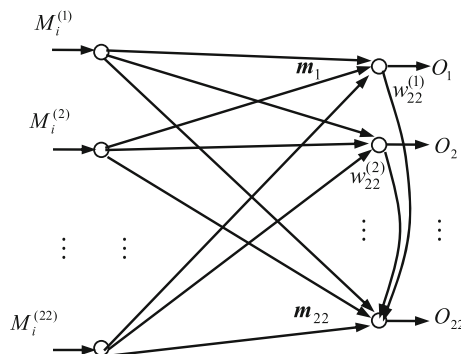


Fig. 2 Structure of an orthogonal learning network

where $\mathbf{O}_j=[O_1, O_2, \dots, O_{j-1}]^T$, which is the output of the front $(j-1)$ neurons; $\mathbf{V}_j = [\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_{j-1}]^T$, a weight matrix; and $\mathbf{W}_j=[w_j^{(1)}, w_j^{(2)}, \dots, w_j^{(j-1)}]^T$, the vector of the lateral connection weights for the j^{th} neuron.

After normalized, the learning rule for the j^{th} neuron is as follows [8, 18],

$$\Delta \mathbf{m}_j = \beta \left(O_j \mathbf{M}_j - \frac{O_j^2}{\mathbf{m}_j^T \mathbf{m}_j} \mathbf{m}_j \right) \tag{19}$$

$$\Delta \mathbf{W}_j = -\gamma \left(O_j \mathbf{O}_j + O_j^2 \mathbf{W}_j \right) \tag{20}$$

where β and γ are positive parameters which determine the learning rates, and their values are set according to the corresponding autocorrelation-matrix for fast training speed with no oscillations. The 1st term of Eq. (19) is that for Hebbian learning rule, which represents a self-strengthening function; the 2nd terms in Eqs. (19) and (20) play a stable role for system, and the 1st term in Eq. (20) stands for the anti-Hebbian learning rule, which causes an inhibition function, and makes the outputs of the network non-correlative even if the input signals are correlative. That is the weight \mathbf{W}_j plays the role of “subtracting” the first $(j-1)$ components from the j^{th} neuron, i.e. the first principal component \mathbf{m}_1 of \mathbf{M} , the second principal component \mathbf{m}_2 of \mathbf{M} , ..., and the $(j-1)^{\text{th}}$ principal component \mathbf{m}_{j-1} of \mathbf{M} are subtracted. Thus the j^{th} vector \mathbf{m}_j tends to become orthogonal to all the previous components, i.e. $\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_{j-1}$, when the train of j^{th} neuron is over. Hence the orthogonal learning rule constitutes an anti-Hebbian rule.

And the iteration algorithms for $\mathbf{m}_j(t+1)$ and $\mathbf{W}_j(t+1)$ are $\mathbf{m}_j(t+1)=\mathbf{m}_j(t)+\Delta \mathbf{m}_j(t)$, and $\mathbf{W}_j(t+1)=\mathbf{W}_j(t)+\Delta \mathbf{W}_j(t)$, respectively. Assuming β and γ are sufficiently small values so that the values of $\mathbf{m}_j(t+1)$ and $\mathbf{W}_j(t+1)$ remain approximately constant during that period of time while an average of the variable in an equation is taken over one sweep of the training data (one sweep means one round of training process involving all the given sample input patterns). To facilitate the proof, we assume that $\beta=\gamma$. Therefore, according to the Eqs. (19) and (20) the weight iteration in one sweep for the orthogonal learning network can be rewritten in the form of state transition matrix as follows,

$$\begin{bmatrix} \mathbf{m}_j(t+1) \\ \mathbf{W}_j(t+1) \end{bmatrix} = \begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{m}_j(t) \\ \mathbf{W}_j(t) \end{bmatrix} \tag{21}$$

where $\mathbf{M}_{11} = \mathbf{E}_{22} + \gamma \left(\left(\sum_{i=1}^{22} \mathbf{M}_i \mathbf{M}_i^T \right) - \frac{\sigma(t)}{\mathbf{m}_j^T \mathbf{m}_j} \mathbf{E}_{22} \right)$, $\mathbf{M}_{12} = \gamma \left(\sum_{i=1}^{22} \mathbf{M}_i \mathbf{M}_i^T \right) \mathbf{V}_j^T$, $\mathbf{M}_{21} = -\gamma \mathbf{V}_j \left(\sum_{i=1}^{22} \mathbf{M}_i \mathbf{M}_i^T \right)$, $\mathbf{M}_{22} = \mathbf{E}_{j-1} - \gamma \left(\mathbf{V}_j \left(\sum_{i=1}^{22} \mathbf{M}_i \mathbf{M}_i^T \right) \mathbf{V}_j^T + \sigma(t) \mathbf{E}_{j-1} \right)$, and $\sigma(t) = E\{O_j^2(t)\}$, where \mathbf{E}_{22} is a unit matrix with 22×22 , and \mathbf{E}_{j-1} is a unit matrix with $(j-1) \times (j-1)$.

In the experiment, first the \mathbf{m}_j is initialed at random; then let them be normalized to form a unitary vector; and at every iteration, while the tuned vector \mathbf{m}_j is obtained, which is normalized to obtained a unitary vector again. Thus $\mathbf{m}_j^T \mathbf{m}_j = 1$, i.e. \mathbf{m}_j is a unitary vector during the iteration processing. On the other hand, while $\mathbf{m}_j(t+1)$ in Eq. (21) is left multiplied by \mathbf{V}_j , then $\mathbf{W}_j(t+1)$ is added, so we have

$$\mathbf{V}_j \mathbf{m}_j(t+1) + \mathbf{W}_j(t+1) = (1-\gamma\sigma(t))(\mathbf{V}_j \mathbf{m}_j(t) + \mathbf{W}_j(t)) \tag{22}$$

As $1 - \gamma\sigma(t) < 1$, when $t \rightarrow \infty$, we have

$$\mathbf{V}_j \mathbf{m}_j(t+1) + \mathbf{W}_j(t+1) \rightarrow \mathbf{0} \tag{23}$$

At the same time, according to Eq. (19), when the system comes to the steady state, that is $\Delta \mathbf{m}_j(t+1) \rightarrow \mathbf{0}$, we have $\left(\left(\sum_{i=1}^{22} \mathbf{M}_i \mathbf{M}_i^T \right) - \lambda_j \mathbf{E}_{22} \right) \mathbf{m}_j + \left(\sum_{i=1}^{22} \mathbf{M}_i \mathbf{M}_i^T \right) \mathbf{V}_j^T \mathbf{W}_j \rightarrow \mathbf{0}$. Due to $\left(\left(\sum_{i=1}^{22} \mathbf{M}_i \mathbf{M}_i^T \right) - \lambda_j \mathbf{E}_{22} \right) \mathbf{m}_j \rightarrow \mathbf{0}$, thus we have $\left(\sum_{i=1}^{22} \mathbf{M}_i \mathbf{M}_i^T \right) \mathbf{V}_j^T \mathbf{W}_j \rightarrow \mathbf{0}$, and $\mathbf{W}_j \rightarrow \mathbf{0}$ (with probability 1). From Eq. (23) we know $\mathbf{V}_j \mathbf{m}_j(t+1) \rightarrow \mathbf{0}$, that is $\mathbf{m}_j(t+1)$ is orthogonal to the vector elements of \mathbf{V}_j (i.e. $\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_{j-1}$), if the iteration times are sufficiently large.

Assuming the learning rates β and γ decrease to zero at the proper speed (for example, let $\beta = \Delta t$), thus the Eq. (19) can be described differentially, that is,

$$\frac{d\mathbf{m}_j}{dt} = O_j \mathbf{M}_i - \frac{O_j^2}{\mathbf{m}_j^T \mathbf{m}_j} \mathbf{m}_j \tag{24}$$

When the system comes to a stable equilibrium, i.e. $d\mathbf{m}_j/dt \rightarrow 0$, the lateral connection approximates to zero. Thus, $O_j = \mathbf{m}_i^T \mathbf{M}_i$. According to Eq. (24), the asymptotic stable solution is obtained as follows,

$$\mathbf{M} \mathbf{m}_j = \frac{\mathbf{m}_j^T \mathbf{M} \mathbf{m}_j}{\mathbf{m}_j^T \mathbf{m}_j} \mathbf{m}_j \tag{25}$$

where $\mathbf{M} = \sum_{i=1}^{22} \mathbf{M}_i^T \mathbf{M}_i$, which is an auto-correlation and symmetrical matrix.

The flow of the solving program is shown in Fig. 3, where ε_1 and ε_2 are jump conditions for program iteration.

When the network comes to the stable state, $\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_{22}$ will converge to the eigenvectors of an auto-correlation matrix \mathbf{M} , that is $\lim_{n \rightarrow \infty} \lambda_j = \left(\mathbf{m}_j^T \mathbf{M} \mathbf{m}_j \right) / \left(\mathbf{m}_j^T \mathbf{m}_j \right)$, which is equivalent to the Lagrange multiplier in Eq. (13). Thus the eigenvectors of \mathbf{M} corresponding to a minimum eigen value λ_{11} and μ_{11} , i.e. $[\mathbf{m}_1 \ \mathbf{0}]$ and $[\mathbf{0} \ \mathbf{m}_2]$ are obtained respectively, elements of which can be taken as the fitting coefficients of the projective matrixes of the cameras in the binocular vision system.

5 Results system calibration and 3D re-construction

5.1 Binocular vision system calibration experiment

As been shown in Fig. 4, a precise robot, which consists of servomechanism, motion controllers, mechanical body, binocular vision system and so on. In the vision system, two cameras are mounted at the ends of a manipulator, which move with the end-effector together (eye-in-hand), so the transformation relation of end-effector and cameras is constant while the manipulator moves. While the vision system is calibrated, first the 3D coordinates of feature points are measured with a 3 dimension coordinate measuring machine [16, 22]. And in order to obtain the variations in Z axis direction of feature points coordinates, the target block's images are sampled with the cameras at different positions by moving the manipulator

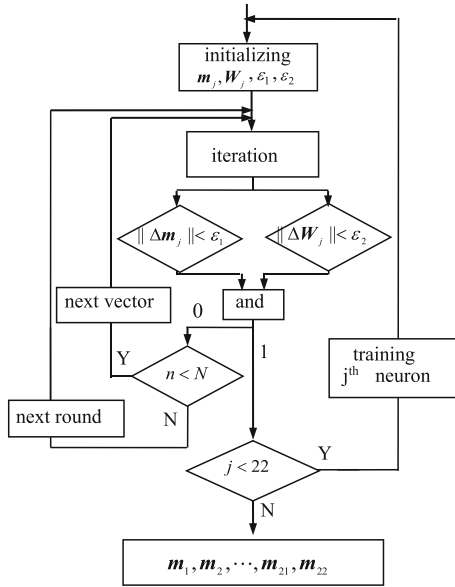


Fig. 3 Flow chart of the program

vertically. Then the corresponding 2D coordinates are estimated with sub-pixel accuracy in the light of the improved Canny’s edge detector algorithm [2, 9].

In the program, the forward and lateral connection weights are initialized at random, and let $\varepsilon_1 = 0.05$, $\varepsilon_2 = 0.005$. After the 22nd neuron trained, the eigen-vector of an auto-correlation matrix of input signals can be obtained, which corresponds to the minimal eigen values, and if $v_{22} = [m^{(1)} \ 0]^T$, else $v_{21} = [0 \ m^{(2)}]^T$. So parameters $m_{34}^{(1)}$ and $m_{34}^{(2)}$ can be obtained, i.e.



Fig. 4 Binocular vision system and manipulators

$m_{34}^{(1)} = -\sum_{i=1}^N \sum_{j=4i-3}^{4i-2} \mathbf{m}_{22}^T \mathbf{x}_j / 2N$, and $m_{34}^{(2)} = -\sum_{i=1}^N \sum_{j=4i-1}^{4i} \mathbf{m}_{21}^T \mathbf{x}_j / 2N$, where $i=1,2,\dots,N$, \mathbf{m}_{21} and \mathbf{m}_{22} are corresponding to eigen values λ_{11} and μ_{11} respectively. So the projective matrixes of the left and right cameras in the binocular vision system can be obtained from the eigenvectors corresponding to minimum eigen values $m_{34}^{(1)}$ and $m_{34}^{(2)}$ respectively, which are shown in Table 1.

The elements in Table 1 constitute the projective matrixes of left and right cameras, so the transformation relations between the image frames and the world frame in binocular vision system are estimated, which are obtained at once.

5.2 3D re-construction by the self-adaptive orthogonal learning network

In the system of binocular vision, the 3D re-construction can be carried out in the light of the self-adaptive orthogonal learning network with lateral restraint. Assume homogeneous coordinates of a feature point in the world frame to be $(X, Y, Z, 1)$, whose projective coordinates in the left and right camera image planes are $(u_{1i}, v_{1i}, 1)$ and $(u_{2i}, v_{2i}, 1)$, according to the camera’s mathematic model, 4 equations can be obtained from Eqs. (3) – (6) as follows:

$$\left(u_1 m_{31}^{(1)} - m_{11}^{(1)}\right) X + \left(u_1 m_{32}^{(1)} - m_{12}^{(1)}\right) Y + \left(u_1 m_{33}^{(1)} - m_{13}^{(1)}\right) Z = m_{14}^{(1)} - u_1 m_{34}^{(1)} \tag{26}$$

$$\left(v_1 m_{31}^{(1)} - m_{21}^{(1)}\right) X + \left(v_1 m_{32}^{(1)} - m_{22}^{(1)}\right) Y + \left(v_1 m_{33}^{(1)} - m_{23}^{(1)}\right) Z = m_{24}^{(1)} - v_1 m_{34}^{(1)} \tag{27}$$

$$\left(u_2 m_{31}^{(2)} - m_{11}^{(2)}\right) X + \left(u_2 m_{32}^{(2)} - m_{12}^{(2)}\right) Y + \left(u_2 m_{33}^{(2)} - m_{13}^{(2)}\right) Z = m_{14}^{(2)} - u_2 m_{34}^{(2)} \tag{28}$$

$$\left(v_2 m_{31}^{(2)} - m_{21}^{(2)}\right) X + \left(v_2 m_{32}^{(2)} - m_{22}^{(2)}\right) Y + \left(v_2 m_{33}^{(2)} - m_{23}^{(2)}\right) Z = m_{24}^{(2)} - v_2 m_{34}^{(2)} \tag{29}$$

According to analytic geometry, the physical meaning of Eqs. (26) and (27) (or Eqs. (28) and (29)) denotes the line for O_1P_1 (or O_2P_2) as shown in Fig 1. In order to obtain the 3D information, the coordinates of P can be obtained from the cross point of O_1P_1 and O_2P_2 [20]. In the solving algorithm, from Eqs. (26)–(29), we let

Table 1 Projective matrixes of binocular vision system obtained with the proposed approach

	Projective matrixes			
Left camera	3.8260×10^{-3}	-1.5979×10^{-4}	2.0241×10^{-3}	7.9791×10^{-1}
	6.1174×10^{-4}	3.9304×10^{-3}	1.1073×10^{-3}	6.0275×10^{-1}
	-4.5228×10^{-7}	-2.4757×10^{-7}	4.6707×10^{-6}	1.9841×10^{-3}
Right camera	-4.1328×10^{-3}	-1.5096×10^{-4}	-1.5225×10^{-3}	-7.6599×10^{-1}
	-9.1105×10^{-5}	-4.1629×10^{-3}	-1.6724×10^{-3}	-6.4283×10^{-1}
	-1.7054×10^{-7}	-4.0999×10^{-7}	-4.7566×10^{-6}	-1.9938×10^{-3}

$$\mathbf{N} = \begin{bmatrix} \frac{u_1 m_{31}^{(1)} - m_{11}^{(1)}}{m_{14}^{(1)} - u_1 m_{34}^{(1)}} & \frac{u_1 m_{32}^{(1)} - m_{12}^{(1)}}{m_{14}^{(1)} - u_1 m_{34}^{(1)}} & \frac{u_1 m_{33}^{(1)} - m_{13}^{(1)}}{m_{14}^{(1)} - u_1 m_{34}^{(1)}} \\ \frac{v_1 m_{31}^{(1)} - m_{21}^{(1)}}{m_{24}^{(1)} - v_1 m_{34}^{(1)}} & \frac{v_1 m_{32}^{(1)} - m_{22}^{(1)}}{m_{24}^{(1)} - v_1 m_{34}^{(1)}} & \frac{v_1 m_{33}^{(1)} - m_{23}^{(1)}}{m_{24}^{(1)} - v_1 m_{34}^{(1)}} \\ \frac{u_2 m_{31}^{(2)} - m_{11}^{(2)}}{m_{14}^{(2)} - u_2 m_{34}^{(2)}} & \frac{u_2 m_{32}^{(2)} - m_{12}^{(2)}}{m_{14}^{(2)} - u_2 m_{34}^{(2)}} & \frac{u_2 m_{33}^{(2)} - m_{13}^{(2)}}{m_{14}^{(2)} - u_2 m_{34}^{(2)}} \\ \frac{v_2 m_{31}^{(2)} - m_{21}^{(2)}}{m_{24}^{(2)} - v_2 m_{34}^{(2)}} & \frac{v_2 m_{32}^{(2)} - m_{22}^{(2)}}{m_{24}^{(2)} - v_2 m_{34}^{(2)}} & \frac{v_2 m_{33}^{(2)} - m_{23}^{(2)}}{m_{24}^{(2)} - v_2 m_{34}^{(2)}} \end{bmatrix}, \mathbf{d} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \text{ and } \mathbf{c} = [-1 \quad -1 \quad -1 \quad -1]^T,$$

so we have

$$\mathbf{N}\mathbf{d} + \mathbf{c} = \mathbf{0} \tag{30}$$

According to the above methods, let the objective function be as follows,

$$\min_{\mathbf{d}} L(\mathbf{d}) = \sum_{i=1}^4 r_i^2 \tag{31}$$

where $r_i = \frac{|\mathbf{N}_i \mathbf{d} + c_i|}{\|\mathbf{d}\|_2}$, and \mathbf{N}_i is the i^{th} row vector of \mathbf{N} . In the algorithm, let $c_i = c$ ($i=1,2,3,4$), where c is the constant proportional to 1, which has no influence on the solving iteration of 3D re-construction. For example, if a fitting coefficient vector \mathbf{d} can make the sum of the squares of the distances from all the vector points' coordinates \mathbf{N}_i to the fitting hyperplane minimizing Eq. (31) with constant 1, it is the same for Eq. (31) with arbitrary constant c , and there is only an offset value for Eq. (31). Thus Eq. (31) can be re-written as follows,

$$L = \frac{\sum_{i=1}^4 \mathbf{d}^T \mathbf{N}_i^T \mathbf{N}_i \mathbf{d} + 2c \sum_{i=1}^4 \mathbf{N}_i \mathbf{d} + 4c^2}{\|\mathbf{d}\|_2^2} = \frac{\mathbf{d}^T \mathbf{s} \mathbf{d} + 2c \mathbf{t} \mathbf{d} + 4c^2}{\|\mathbf{d}\|_2^2} \tag{32}$$

where $\mathbf{s} = \sum_{i=1}^4 \mathbf{N}_i^T \mathbf{N}_i$, and $\mathbf{t} = \sum_{i=1}^4 \mathbf{N}_i$, where \mathbf{N}_i is the i^{th} row vector of \mathbf{N} .

In order to minimize L , the critical points of Eq. (32) can be gotten by letting $dL/d\mathbf{d} = 0$, that is

$$\mathbf{s} \mathbf{d} + c \mathbf{t}^T - \frac{\mathbf{d}^T \mathbf{s} \mathbf{d} + 2c \mathbf{t} \mathbf{d} + 4c^2}{\|\mathbf{d}\|_2^2} \mathbf{d} = 0 \tag{33}$$

According to Eq. (30), we know the expected value of $c = -\mathbf{N}_i \mathbf{d} = -\mathbf{t} \mathbf{d} / 4$, thus

$$\mathbf{T} \mathbf{d} - \lambda \mathbf{d} = 0 \tag{34}$$

where $\mathbf{T} = \mathbf{s} - \mathbf{t}^T \mathbf{t} / 4$, and $\lambda = \frac{\mathbf{d}^T \mathbf{T} \mathbf{d}}{\|\mathbf{d}\|_2^2}$.

Thus \mathbf{d} is the eigen-vector of \mathbf{T} corresponding to the eigen-value λ , and the 3D coordinates of the feature points in the world frame can be obtained from the eigen-vector of \mathbf{T} corresponding to the minimum eigen-value.

For the 3D re-construction, the experiment is carried out in high precision robot, as can be seen from Fig. 4. Firstly move the manipulator in vertical direction through controller of servosystem, and the target block's images are sampled by the two cameras in the stereo vision

Table 2 Coordinates of the feature points projected in the left and right image planes (/Pixel)

	Left camera planes	Right camera planes
1	(483.1786, 325.9986)	(415.7194, 381.6402)
2	(482.5360, 385.5565)	(415.4020, 438.8772)
3	(504.5829, 372.8164)	(437.6854, 421.7096)
4	(526.6801, 316.3231)	(459.5774, 362.9787)
5	(463.9776, 345.7225)	(398.7574, 402.3754)
6	(464.4011, 390.5767)	(400.6617, 444.0929)

system. Then 6 points are chosen at random for precision analysis at several positions, and the 2D coordinates of the feature points projected in the left and right camera image planes are obtained and shown in Table 2.

In the 3D re-construction programming, the adaptive orthogonal learning network is designed similarly as Fig. 2; and the flow chart of program is similar to Fig. 3, whose input neurons are three and the output neurons are three too. And the input signals are the i^{th} row vector of \mathbf{T} . After the 3rd neuron trained, the system came to an equilibrium state, and the scale was obtained according to Eq. (35), so the world coordinates of the feature points could be obtained from the weight vector which connected the input neurons according to Eq. (36), and the 3D re-construction was achieved.

$$s = \mathbf{td} / 4 \tag{35}$$

$$\mathbf{D} = \mathbf{d}^{(3)} / s \tag{36}$$

where $\mathbf{d}^{(3)}$ is the eigen-vector of \mathbf{T} corresponding to the minimal eigen-value, and \mathbf{D} is the solved 3D coordinates of the feature point in the world frame.

On the other hand, if the least square method (LSM) is adopted, the projective matrixes of the left and right camera are obtained as shown in Table 3.

While precision analysis experiment is carried out, the actual coordinates (AC) of feature points in the world frame are measured and shown in the 1st line of Table 4. When the 3D re-construction is achieved, if their coordinates are obtained in the light of the adaptive orthogonal learning algorithm (abbreviated as CwAOL), which are shown in the 2nd line in Table 4. For the system calibration with the least square method, the 3D coordinates of the feature points in the world coordinate system (abbreviated as CwLSM) are obtained as shown in the 3rd line in Table 4.

Table 3 Projective matrixes obtained with LSM

	Projective matrixes			
Left camera	1.929473	-8.0778×10^{-2}	1.0222826	4.0213×10^2
	3.0807×10^{-1}	1.98270185	5.5941×10^{-1}	3.0380×10^2
	-2.2964×10^{-4}	-1.2526×10^{-4}	2.3587×10^{-3}	1
Right camera	2.0725555950	7.5858×10^{-2}	0.764422705	3.8425×10^2
	4.4564×10^{-2}	2.0889978	0.839632498	3.2244×10^2
	8.2712×10^{-5}	2.0608×10^{-4}	2.3880×10^{-3}	1

Table 4 3D coordinates and precision performance indexes (/mm)

No.	1	2
AC	(66.0076, 45.0060, 450.0000)	(66.0047, 105.0050, 450.0000)
CwAOL	(66.0914, 45.0819, 450.1075)	(66.0331, 104.9067, 449.8954)
CwLSM	(66.0776, 45.0737, 450.1294)	(66.0092, 104.8910, 449.8587)
PIwAOL	0.1560	0.1463
PIwLSM	0.1619	0.1816
No.	3	4
AC	(86.0086, 85.0002, 430.0000)	(106.0013, 25.0011, 420.0000)
CwAOL	(86.1524, 85.1208, 430.1142)	(106.0368, 25.0341, 420.1894)
CwLSM	(86.1420, 85.1188, 430.1399)	(106.0318, 25.0243, 420.2013)
PIwAOL	0.2197	0.1955
PIwLSM	0.2268	0.2050
No.	5	6
AC	(46.0048, 65.0066, 420.0000)	(46.0026, 105.0088, 400.0000)
CwAOL	(45.9932, 64.9682, 419.8164)	(46.1314, 104.9285, 399.8992)
CwLSM	(45.9707, 64.9566, 419.8124)	(46.1135, 104.9248, 399.9403)
PIwAOL	0.1879	0.1822
PIwLSM	0.1972	0.1514

The difference between the actual coordinates in the world frame and the solved coordinates is taken as the precision performance index [3, 7, 14]. That is

$$d_i = \sqrt{\left(X_i^{(a)} - X_i^{(s)}\right)^2 + \left(Y_i^{(a)} - Y_i^{(s)}\right)^2 + \left(Z_i^{(a)} - Z_i^{(s)}\right)^2} \quad (37)$$

where $(X_i^{(a)}, Y_i^{(a)}, Z_i^{(a)})$ are the actual coordinates in the world frame, and $(X_i^{(s)}, Y_i^{(s)}, Z_i^{(s)})$ are the solved 3D coordinates according to the proposed technique, or the other data processing methods such as LSM.

The precision performance index of the two algorithms can be obtained according to Eq. (37), as shown in the 4th and 5th line in Table 4. PIwAOL in Table 4 denotes the precision index with the proposed technique, i.e. the self-adaptive orthogonal learning network. And PIwLSM denotes the precision index with the least square method. From Table 4, it is demonstrated that the proposed approach has higher precision, and can meet the precision requirements for engineering practice. From the above calculation, we found that the presented results are interesting: with patterns of lateral inhibition, the orthogonal learning neural network can achieve fast self-organization for relatively loose structural constraints according to simple anti-Hebbian rule, or a slight modification. The proposed technique is referential and helpful for precision manufacture and measurement, such as precision processing of micro-drill, gear, and other work-pieces.

6 Conclusions and future works

The proposed approach in the paper has the following key features as opposed to other techniques [4–6, 11, 12, 15, 17, 19, 21]: 1) The fitting projective matrixes for the left and right cameras in the binocular vision system are obtained from the eigen-vectors of an auto-correlation matrix corresponding to minimal eigen-values, which minimize the sum of the

square of distances from the combined vector coordinates of the feature points to the fitting hyperplane; 2) A self-adaptive orthogonal learning neural network is designed to obtain the eigenvectors of the autocorrelation matrix corresponding to the minimal eigen-values, where the j^{th} neuron is trained, its corresponding vector is perpendicular to the front $(j-1)$ vectors already obtained; 3) 3D re-construction is carried out according to the technique proposed above with the advantages such as programming easily and high precision. Such a study provides a new and applicable technique in data processing for calibrating binocular vision systems and 3D re-construction. In the future we will carry out research on how to set the learning rate of the orthogonal learning neural network to enable the training speed as fast as possible.

Acknowledgments Foundation item: The work described in this paper is partially supported by the National Natural Science Foundation of China (51175187), the Hunan Provincial Natural Science Foundation of China (09JJ6092), and the Science & Technology Foundation of Guangdong Province under grant Nos. 2013B021300023 and 2013B090600112. The authors also gratefully acknowledge the suggestions of the reviewers, which helped to improve the presentation.

References

1. Božić B, Ristić K, Pejić M (2014) Parameter estimation and accuracy analysis of the free geodetic network adjustment using singular value decomposition. *Technical Gazette* 21(2):451–456
2. Brandusa PA, Catalin B (2011) Development of an embedded artificial vision system for an autonomous robot. *Int J Innov Comput Inf Control* 7(2):745–762
3. Chatterjee C, Roychowdhury VP, Chong EKP (1997) A nonlinear gauss-seidel algorithm for noncoplanar and coplanar camera calibration with convergence analysis. *Comp Vision Image Underst (CVIU)* 67(1):58–80
4. Chen SY, Li YF (2013) Finding optimal focusing distance and edge blur distribution for weakly calibrated 3-D vision. *IEEE Trans Ind Inf* 9(3):1680–1687
5. Ge DY, Yao XF (2010) Application of the Sanger operator with lateral connection in camera calibration. *J Optoelectron Laser* 21(11):1720–1724
6. Guillemaut JY, Aguado AS, Illingworth J (2005) Using points at infinity for parameter decoupling in camera calibration. *IEEE Trans Pattern Anal Mach Intell* 27(2):265–370
7. Kang JH, Kang SJ, Kim S (2015) Line recognition algorithm for 3D polygonal model using a parallel computing platform. *Multimed Tools Appl* 74(1):259–270
8. Khan MAU, Khan TM, Khan RB, Kiyani A, Khan MA (2012) Noise characterization in web cameras using independent component analysis. *Int J Innov Comput Inf Control* 7(2):302–311
9. Khan SA, Usman M, Riaz N (2015) Face recognition via optimized features fusion. *J Intell Fuzzy Syst* 28(4): 1819–1828
10. Kung SY, Diamantaras KI (1990) A neural network learning algorithm for adaptive principle components extraction (APEX). *Proc Int Conf Acoust Speech Signal Process* 2:861–864
11. Ma SD (1996) A self-calibration technique for active vision system. *IEEE Trans Robot Autom* 12(1):114–120
12. Miyagawa I, Arai H, Koike H (2010) Simple camera calibration from a single image using five points on two orthogonal 1-d objects. *IEEE Trans Image Process* 19(6):1528–1538
13. Palmieri F, Zhu J, Chang CH (1993) Anti-Hebbian learning in topologically constrained linear networks a tutorial. *IEEE Trans Neural Netw* 4(5):748–761
14. Perez U, Cho SH, Asfour S (2009) Volumetric calibration of stereo camera in visual servo based robot control. *Int J Adv Robot Syst* 6(1):35–42
15. Rahman T, Krouglicof N (2012) An efficient camera calibration technique offering robustness and accuracy over a wide range of lens distortion. *IEEE Trans Image Process* 21(2):626–637
16. Wang B, Liu W, Jia Z et al (2011) Dimensional measurement of hot, large forgings with stereo vision structured light system. *Proc IMechE B J Eng Manuf* 225(6):901–908
17. Yin F, Makris D, Velastin SA, Ellis T (2015) Calibration and object correspondence in camera networks with widely separated overlapping views. *IET Comput Vis* 9(3):354–367

18. Yuan YM, Fang YH, Cui FX, Li DC (2011) Research on preprocessing algorithm for differential polarization spectrum of oil spills on water. *Acta Opt Sin* 31(11):1128001-1–1128001-7
19. Zhang Z (2000) A flexible new technique for camera calibration. *IEEE Trans Pattern Anal Mach Intell* 22(11):1330–1334
20. Zhao ZX, Wen GJ, Zhang X, Li DR (2012) Model-based estimation for pose, velocity of projectile from stereo linear array image. *Meas Sci Rev* 12(3):104–110
21. Zheng Y, Peng SL (2014) A practical roadside camera calibration method based on least squares optimization. *IEEE Trans Intell Transp Syst* 15(2):831–843
22. Zhou JM, Zhou QX, Shu LL, Xu DD (2011) Research on thermal image enhancement for detecting early mechanical damage in apple. *Sens Lett* 9(3):1031–1036



Dong-yuan Ge (b. April 14, 1970) received his M. Sc. in Mechanical Engineering (2004) from Wuhan University of Technology, and received his Ph. D. in Mechanical Engineering (2013) from South China University of Technology. He is an associate professor at Department of Mechanical and Electrical Engineering, Tan Kah Kee College, Xiamen University. His current research interests include Machine Vision and Computational Intelligence. He has (co-)authored more than 2 books and 56 papers.



Xi-fan Yao (b. October 22, 1964) received his M. Sc. in Mechanical Engineering (1988) and Ph. D. in Mechanical Engineering (1999) from South China University of Technology. Now he is full professor of School of Mechanical and Automotive Engineering, South China University of Technology. His current research interests include Intelligent Manufacture, Cloud Manufacturing and Computational Intelligence and so on. He has (co-)authored more than 2 books and over 255 papers.



Zhao-tong Lian (b. March 05, 1964) received his M.Sc. in Statistics (1991) from Zhongshan University, and Ph. D. in Industrial Engineering and Operations Research (1997) from Hong Kong University of Science and Technology. Now he is full professor of Electronic Business at Business Administration, University of Macau, Macau SAR, P. R. China. He has (co-)authored more than 5 books and 70 papers.