This paper appears in: International Conference on Robotics and Automation, 2002

An World-Independent Approach for the Calibration of Mobile Robotics Active Stereo Heads

Guilherme Nelson DeSouza Andrew H. Jones Avinash C. Kak
Robot Vision Lab.
Purdue University
{gdesouza, ajones, kak}@ecn.purdue.edu

Abstract

This paper describes an approach for calibrating active stereo heads for which no prior knowledge of the world coordinate system is required. During an automated procedure derived from this approach, the mobile robot can be loosely positioned in front of a calibration pattern. Through our procedure, both the camera calibration and the neck-eye calibration are obtained. With this set of calibration matrices, the XYZ coordinates of a landmark can be determined with respect to the robot coordinate frame. The so calculated XYZ coordinates are accurate regardless of the orientation of the stereo head, even for orientations that were not used for calibration. This paper validates our approach by presenting results as accurate as a few millimeters in the calculation of the XYZ coordinates of objects at 1.5m from the stereo head.

1 Introduction

In any robotic vision application, calibration is a task that consumes a great deal of time and effort. Whether it is an application for visual servoing, bin-picking, object modeling, or mobile robot navigation, sensing devices such as cameras, encoders, sonars and lasers must be calibrated. The result from calibration is nothing but a mapping function from sensory units such as pixels, revolutions, and waveform cycles into *world* units such as meters, centimeters, seconds, etc.

In order to obtain such a mapping, it is not sufficient to only calibrate the sensor – as in camera calibration. In fact, one must also determine the pose (position and orientation) of the sensor with respect to some world reference. In visual servoing for example, cameras are frequently attached to a robotic manipulator's end-effector. In this case, the calibration of the camera pose with respect to the end-effector is referred to as the *hand-eye* problem [3, 4, 5, 11]. Here, since the base of the robot is fixed, the hand-eye calibration can be obtained simultaneously [3, 4] or separately [6, 9] with the world-base calibration – the transformation between the world coordinate system and the robot-base coordinate system.

In mobile robotic applications using active stereo heads (with pan, tilt, and vergence), a variation of the hand-eye problem is presented and is commonly referred to as the *neck-eye* problem or *head-eye* problem. The neck-eye problem is very similar to the hand-eye problem and it has been solved in the past using the same kind of approaches used for hand-eye calibration [7, 8][1]. However, the calibration of mobile robot stereo heads differ from the calibration of cameras on robotic manipulators in two important aspects. First, as it follows in sec. 2.3, there are two pairs of neck-eye matrices to be calibrated for active stereo heads, while there is only one pair for robotic manipulators. Second, while the latter has its base fixed, mobile robots have moving bases. That means that the robot base can assume any pose with respect to the world coordinate system – which is usually defined by a calibration target. This problem has forced researchers to carefully position the mobile robot in a known place, with a specific orientation, before any calibration can be accomplished. The burden of performing this tasks adds to the normal complexity of the calibration procedure, and most importantly, introduces errors that can be smaller or larger depending on the human ability or skill to perform it.

In this paper we present a formalism that shows that the complete calibration of an active stereo head (camera and neck-eye calibration) can be performed without any prior knowledge of the pose of the robot with respect to the world-coordinate frame. In essence, our calibration procedure allows us to *loosely* move the robot in front of the calibration pattern, run a few scripts and have its sensors calibrated automatically with respect to the robot coordinate frame. It is important to mention here that most mobile robot navigation schemes require the calculation of world coordinates of landmarks for self-localization [2]. However,

[1] In [8] only the intrinsic parameter of the cameras are calibrated, while in [7] a look-up table with the intrinsic parameters for each calibrated position of the stereo head was employed.

even in self-localization the robot must first locate the landmarks with respect to its own coordinate frame. Once the relative position is known, then the world position of the landmarks can be determined. In that sense, our approach does not introduce any shortcomings compared to other approaches, but instead removes the burden of an exact positioning of the robot during calibration.

Finally, every formalization of the hand-eye problem, or neck-eye problem, can be reduced to the problem of finding a solution for the equation (or a set of equations) AX=YB [4], or AX=XB [10]. In [4], we presented an iterative approach to solve this problem. In this paper a slight variation of that algorithm is employed. As it will be shown in the following sections, the results approached pixel accuracy, even for postions of the stereo head that were not used for calibration.

2 Problem Formulation

2.1 Notation

In this paper, we followed the Denavit-Hartenberg (D-H) notation [1]. In this notation, a homogeneous transformation matrix (HTM) H is a 4x4 matrix that relates a homogeneous vector represented in one coordinate frame with its representation in terms of another coordinate frame. The superscript before the letter H stands for the new frame the vector is to be converted into, and the subscript after the H indicates the frame in which the vector was originally represented. For example, ${}^{B}H_{w}$ is the HTM that transforms a 4x1 homogeneous vector from the *world* coordinate frame into the *Base* coordinate frame. The inverse of this matrix is denoted by ${}^{w}H_{B}$, and it transforms a vector in the *Base* coordinate frame into the *world* coordinate frame. The same notation is applied for the rotational part of the HTMs. For example, ${}^{B}R_{w}$ is the rotation of a vector from the *world* frame into the *B*ase coordinate frame. The inverse of this matrix is denoted by ${}^{w}R_{B}$.

2.2 Geometry of the System

Before formulating the problem, it is important to explain the geometry of our stereo head and the choices of all necessary coordinate frames. On the left in Fig. 1 are depicted the camera and stereo head settings. As the reader can observe, the stereo head can move with three degrees of freedom: pan, tilt and vergence. At the neck base, a coordinate frame denoted B is defined. It is around the B coordinate frame that both pan and tilt are performed. Next, at the top-left and top-right of B, two additional coordinate frames, denoted E_l and E_r, are established for the left and right camera mounting points respectively. It is around these two

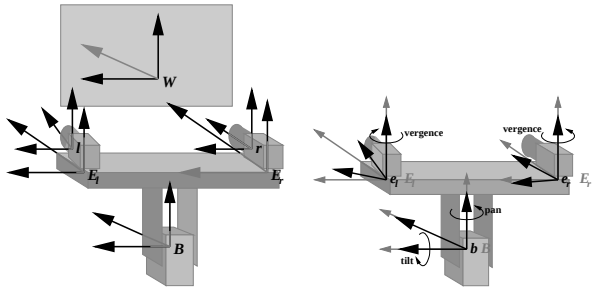

Figure 1: Camera setting and coordinate frames

coordinate frames that the vergence is perfromed – a positive value makes the right and left cameras turn towards each other. Additionally, near the two camera mounting points we defined the cameras coordinate frames l and r. Finally, the calibration pattern, which defines the world coordinate system W, is placed in front of the stereo head.

Our problem formulation starts by creating three additional frames: b, e_l, and e_r, as shown on the right in Fig. 1. These arbitrary frames are necessary to determine the neck-eye calibration independently of the pan, tilt and vergence angles. In this instance, the coordinate frame b is a coordinate frame with the same origin as B. However, while B is fixed, the coordinate frame b rotates according to pan or tilt angles, following the motion of the stereo head. In that way, the HTMs ${}^{E_l}H_b$ and ${}^{E_r}H_b$ are always constant regardless of the pan and tilt angles and are the first pair of neck-eye matrices that need to be calibrated (Sec.2.3.1). Similarly, the frames e_l and e_r have the same origin as E_l and E_r respectively and they rotate with the cameras as they are turned by a certain vergence angle. As before, ${}^{E_l}H_{e_l}$ and ${}^{E_r}H_{e_r}$ are constant regardless of the vergence angle and are the second pair of neck-eye matrices to be calibrated (Sec. 2.3.3).

2.3 Calibration Equations

Without loss of generality, we will formalize our calibration problem for one single camera. Therefore, we will drop some of the subscripts above and use the symbols E and c for the eye coordinate frame and the camera coordinate frame respectively. In summary, our coordinate frames are now: B, b, E, e, c, and w.

2.3.1 Calibration of ${}^{c}H_{e}$

The calibration of the camera with respect to its mounting point, ${}^{c}H_{e}$, is obtained by taking the following steps:

1. fix pan and tilt (usually equal to zero);
2. choose a value for vergence, v_0, and calculate ${}^{e_0}H_{E}$. Since e_0 and E have the same origin and

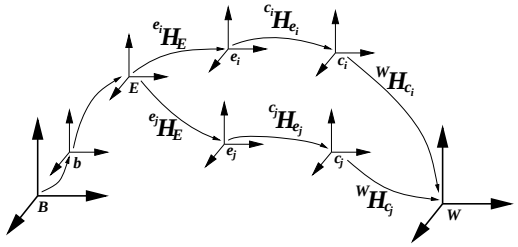

Figure 2: Chain of HTMs used for the calibration of ${}^{c}H_{e}$ for two values of vergence v_i and v_j.

are rotated only by the chosen vergence angle, the HTM ${}^{e_0}H_E$ is trivially calculated.

3. place the calibration pattern in an arbitrary position and perform camera calibration[2] , obtaining the extrinsic parameters: ${}^{c_0}H_w$

4. repeat steps 2 and 3 for the same postion of the pattern but different vergence angles: v_1, v_2, v_3, ...; obtaining the corresponding extrinsic parameters: ${}^{c_1}H_w$, ${}^{c_2}H_w$, ${}^{c_3}H_w$, ...

5. Apply a modified version of hand-eye calibration algorithm in [4] as explained in the next section.

2.3.2 Iterative algorithm

As illustrated in Fig. 2, for each pair of values (v_i, v_j) we can write:

$$ {}^{w}H_{c_j} * {}^{c_j}H_{e_j} * {}^{e_j}H_E = {}^{w}H_{c_i} * {}^{c_i}H_{e_i} * {}^{e_i}H_E $$

which can be re-written as:

$$ {}^{c_i}H_w * {}^{w}H_{c_j} * {}^{c_j}H_{e_j} = {}^{c_i}H_{e_i} * {}^{e_i}H_E * {}^{E}H_{e_j} $$

and re-combined as

$$ {}^{c_i}H_{c_j} * {}^{c_j}H_{e_j} = {}^{c_i}H_{e_i} * {}^{e_i}H_{e_j} $$

or

$$ A_{ij} * {}^{c_j}H_{e_j} = {}^{c_i}H_{e_i} * B_{ij} $$

This last equation is in the expected form $AX = YB$ mentioned earlier. To solve this equation, we use an iterative algorithm similar to [4]. Since the camera frame c_i is fixed with respect to the mounting point frame e_i for any value of i, the HTMs X and Y represent in fact the same transformation ${}^{c}H_{e}$. Therefore, the proposed modified method uses every iteration of the algorithm to refine only the value of the matrix ${}^{c}H_{e}$. The process begins by writing n equations in the form $Y = A_{ij}XB_{ji}$, and another n equations in the form $X = A_{ji}YB_{ij}$, where A_{ji} and B_{ji} are the inverses of A_{ij} and B_{ij} respectively. Each set of equations represents the paring of two HTMs from two different choices of vergence v_i, v_j as explained above. That results in the following set of equations:

$$ \begin{aligned} Y_1 &= A_{ij} * X * B_{ji} && (1)\\ Y_2 &= A_{jk} * X * B_{kj} && (2)\\ &\ldots \\ Y_n &= A_{ni} * X * B_{in} && (3) \end{aligned} $$

and

$$ \begin{aligned} X_1 &= A_{ji} * Y * B_{ij} && (4)\\ X_2 &= A_{kj} * Y * B_{jk} && (5)\\ &\ldots \\ X_n &= A_{in} * Y * B_{ni} && (6) \end{aligned} $$

Next, given an initial guess for X, the algorithm computes the values for Y_1, Y_2, ... Y_n that satisfy equations (1) to (3) above. But since the camera is stationary with respect to its mounting point, as we mentioned previously, the matrices Y_1, Y_2, ... Y_n are in fact estimates of the same transformation matrix ${}^{c}H_{e}$. For this reason, we "average" these n HTMs to find an updated matrix that will represent Y. This averaging is done via the quaternion representation of the rotation parts of the HTMs above and it is an improved version of [4] presented here in Appendix A. Next, this updated Y is used in equations (4)-(6) to obtain n values for the matrices X_1, X_2, ... X_n. Again, X_1, X_2, ... X_n are all estimates of ${}^{c}H_{e}$. So, we average them to obtain an updated estimate for X. This process continues until all estimates have converged to the correct values or at least to within the specified tolerance. The proof of this convergence is given in [4].

In order to make the process converge faster, a rough approximation for ${}^{c}H_{e}$ can be used, but this is not necessary for convergence. For our experiments, we used the identity matrix as the initial guess for ${}^{c}H_{e}$.

2.3.3 Calibration of ${}^{E}H_b$

The calibration of the camera mounting point with respect to the base, ${}^{E}H_b$, was obtained using a procedure very similar to that in sections 2.3.1 and 2.3.2. The only difference here is that we now fix the vergence angle, and we perform camera calibrations (intrinsic and extrinsic parameters) for different values of pan and tilt. Like before, since the two neck base coordinate frames B and b have the same origin and are rotated only by the pan and tilt angles, the HTMs ${}^{b_i}H_B$ corresponding to different pairs of angles (p_i, t_i) can be trivially calculated. Despite that difference, the two new sets of n equations in the form $AX = YB$ are obtained in a similar way as was done for eq. (1) through

[2] The intrinsic and extrinsic parameters of the cameras were obtained using the camera model with radial distortion presented in [13].

Depth	Left	Right	Top	Bottom
1,601.5	14.74	14.47	1.12	1.03
7,483.6	337.0	309.3	48.7	21.6
15,029.8	1,409.3	1,187.2	243.7	26.4

Table 1: Errors in XYZ coordinates due to pixel accuracy for different directions. All numbers expressed in millimeters

(6). For these new sets, X and Y represent ${}^{E}H_{b}$. The actual derivation of these equations is straighforward and therefore is omitted due to the limitation in the length of the paper.

3 Results

3.1 Pixel accuracy

We define pixel accuracy as a measurement of the goodness in 3D reconstruction. It basically addresses the question of how accurate 3D reconstruction can be due to pixel quantization. During 3D reconstruction, the same landmark must be located in both left and right images. If this match is off by a pixel or a fraction of a pixel, the resulting XYZ coordinates will reflect this mismatch. For our tests, we selected landmarks either manually or automatically. In either case, the best result that can be obtained is limited by this pixel accuracy. In other words, whenever a perfect match cannot be chosen due to the pixel quantization, the XYZ reconstruction will carry an error, even if the calibration introduced no errors at all.

In order to gain a perspective on the XYZ error associated with the selection of the UV pixel, we selected a point in the left image, and then selected the correct match and its 4 neighboring points in the right image (top, bottom, left, and right). After the 3D reconstruction of this data, the XYZ values for each direction were compared with respect to the center point. This procedure was performed for a set of points corresponding to landmarks at different depths. The errors associated with each direction are shown in Table 1. Note that the errors for the left and right directions are greater than for the top and bottom. This difference was due to the camera setting (baseline), where the stereo disparity is mostly along the horizontal line.

3.2 Test Methods

For testing our calibration procedure, the robot was placed in a fixed position and the XYZ coordinates of various landmarks were calculated. If our calibration procedure is correct, it should be able to determine the XYZ coordinates of all these landmarks regardless of the current pan and tilt angles. While it is important to compare the calculated absolute XYZ coordinates, the focus here is to prove that these world coordinates are constant. The intent of this section is to validate this claim using two different methods.

Figure 3: Image of the hallway with landmark selection used in Test Method 1

Method 1 - Obtain testing data rich in depth:

The purpose of this first testing method was to obtain data rich in depth information. For that, we placed the mobile robot in a hallway (16 meters long) and from the stereo images of this hallway, a user selects UV pixels in both the left and right images that correspond to the same landmark. To assure richness in depth information, these landmarks are chosen at distances varying from 1.5m to 15m. Fig. 3 shows the left image of the hallway and the selected points. The stereo head is then moved to various pan and tilt configurations and the same landmarks are selected by the user. Using the selected UV pixel points, 3D reconstruction, the neck-eye calibration, and the kinematics of the stereo head, we determined the XYZ coordinates of the landmarks. We calculated 6 values of XYZ coordinates for each landmark corresponding to 6 different positions of the stereo head. One of these positions – pan and tilt equal to zero – was regarded as the reference or ground truth. For all other positions of the head, we calculated the distance beween the XYZ coordinates of the landmarks obtained for that position and the reference postion. All these distances were then averaged and the results are shown in Table 2. The table presents the above mentioned average distances projected onto the X, Y, and Z directions (columns 3 to 5). Also, the table shows the actual average distance in space (column 6).

While the natural environment provided ample number of landmarks to be selected, it required a user to click on all landmarks consistently and correctly in both images. Furthermore, that selection had to be done for each pan/tilt position of the stereo head.

Actual Depth	Calculated Depth	dX	dY	dZ	Dist.
1,602	1,604	1.47	11.42	-2.12	12.75
7,484	7,467	-6.40	28.23	1.88	147.68
15,030	15,467	-22.33	-518.9	10.84	520.79

Table 2: Table showing the average distances over 6 positions of the stereo head obtained from Test Method 1. All values are expressed in millimeters

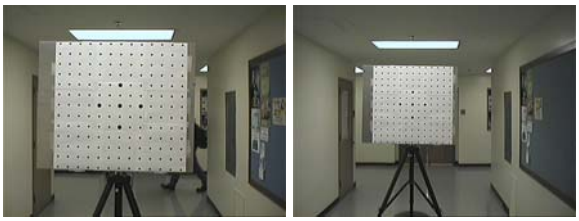

Figure 4: Images of the calibration pattern for automatic detection of landmarks used in Test Method 2

This difficulty led us to devise an automated system that selects landmarks – method 2, which is described below.

Method 2 - Automated landmark selection:

The automated selection of points in both left and right images was performed by employing the same pattern and software that was used for calibration. That allowed us to achieve the purpose of this method which was the extraction of a much larger number of matching points than in the previous method. Moreover, the advantage of using this software and pattern is to eliminate human-introduced errors in the selection and matching of points. The pattern was initially mounted on a tripod and arbitrarily placed in front of the stereo head (Fig. 4). Stereo images at different pan and tilt angles were captured. From all these images, the software mentioned above detected matching points on the calibration pattern for each pan and tilt configuration. The software then calculated the XYZ coordinates of these points using 3D reconstruction, the neck-eye calibration, and the kinematics of the stereo head. The different XYZ coordinates for corresponding points (from the different pan and tilt angles) were then compared with respect to the center view (pan and tilt equal to zero) by means of their distances. Tables 3 shows the average distances over all the points in the pattern. The third column lists these distances project in the X direction. Similarly, the fourth and fifth columns show the distances in the Y and Z directions. The use of projections is important to show how much of the error is along a particular direction, namely the Y direction which encompasses the depth.

Pan	Tilt	dX	dY	dZ	Distance
-10^o	10^o	-1.859	-5.569	0.714	6.020
-10^o	0^o	-1.668	1.479	4.002	5.092
-10^o	-10^o	0.738	1.997	4.001	4.921
0^o	10^o	0.520	-2.549	3.549	4.878
0^o	-10^o	0.296	2.504	2.709	4.169
10^o	10^o	2.551	1.835	3.781	5.419
10^o	0^o	3.021	3.531	-0.221	5.014
10^o	-10^o	2.541	2.971	4.194	6.488
Average		0.4530	0.4051	2.8411	5.2489
Std Deviation		1.8849	3.9256	1.6090	1.5473

Table 3: Table showing the average distances in millimeters for the pattern at 1.2m from the stereo head

Pan	Tilt	dX	dY	dZ	Distance
-15^o	15^o	2.420	-19.411	-4.827	20.405
-15^o	0^o	4.145	-5.952	5.315	10.990
-10^o	10^o	5.190	-13.101	-3.415	15.439
-10^o	0^o	6.856	-5.210	7.208	12.832
-10^o	-10^o	6.062	3.995	3.817	10.689
0^o	-15^o	9.680	-9.668	-3.258	14.710
0^o	10^o	9.807	-6.284	-2.730	13.829
0^o	-10^o	11.481	3.749	3.491	14.469
10^o	10^o	13.796	6.250	1.593	16.806
10^o	0^o	4.203	4.589	1.718	8.913
10^o	-10^o	14.980	5.512	6.331	18.521
15^o	-15^o	14.226	11.974	-3.344	19.919
15^o	0^o	18.766	11.457	-6.399	23.852
Average		9.3548	-0.9332	0.1588	15.4897
Std Deviation		4.5064	11.2934	3.9168	6.9080

Table 4: Table showing the average distances in millimeters for the pattern at 2.1m from the stereo head

Next, the tripod was moved to a different depth with respect to the stereo head and additional images were taken. The same procedure above was repeated on these images. Table 4 shows the similar averages for the new depth.

4 Conclusions

This paper presents an approach for automatically calibrating a stereo head without prior knowledge of the world coordinate frame. The approach was validated using actual landmarks from an indoor environment. Also, artificial landmarks were employed to increase the number of data points used for testing. The upshot of a large number of artificial landmarks is the reliability of the results. These results, shown in the tables in section 3, prove that the approach provides a mean to estimate the XYZ coordinates with respect to the

robot base of landmarks to within a few millimeters. The estimates were equally accurate regardless of the configuration of the stereo head. The configurations used for testing were not necessarily the same used for calibration. Finally, this approach proved sufficiently accurate to be useful for applications of mobile robotic navigation such as self-localization.

Appendix A

Quaternions are 4-dimensional unit vectors used to represent rotations of rigid objects. A quaternion can be thought of as a pair (s, v), where s is a real number representing an *amount* of rotation, and v represents the axis around which this rotation is performed. Since quaternions are unit vectors, it is reasonable to consider the true measure of the size of a quaternion to be its s-component. If this component is one, then, regardless the value of the v-component, the equivalent rotation matrix is the identity matrix. Also, if the amount of rotation is given by the angle θ with respect to the e axis (the unit vector along the axis of rotation), then the quaternion can be expressed as $q = (s, v)$, where:

$$\begin{aligned} s &= cos(\frac{\theta}{2}) \\ v &= sin(\frac{\theta}{2}) * e \end{aligned}$$

In our current implementation, the "average" of two quaternions representing the rotation parts of two HTMs, ${}^aR_b$ and ${}^cR_b$, is the result from taking half of the amount of rotation given by ${}^aR_c$. In other words, if we take the quaternion representation of ${}^aR_c$ and determine half of its amount of rotation, $\frac{\theta}{2}$, with respect to the same e axis, we can calculate the new average quaternion, which is now given by:

$$\begin{aligned} s &= cos(\frac{\theta}{4}) \\ v &= sin(\frac{\theta}{4}) * e \end{aligned}$$

If we call this rotation ${}^mR_c$, the final average will be given by: ${}^mR_b = {}^mR_c * {}^cR_b$

References

[1] Denavit, J., and Hartenberg, R., S., "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices", Journal of Applied Mechanics, vol. 77, pp. 215-221, 1955.

[2] DeSouza, G., and Kak, A., "Vision for Mobile Robot Navigation: A Survey," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 2, pp. 237-267, 2002.

[3] Dornaika, F., and Horaud, R., "Simultaneous robot-world and hand-eye calibration," IEEE Transactions on Robotics & Automation, vol.14, no.4, pp.617-622, Aug. 1998.

[4] Hirsh, R., DeSouza, G., Kak, A., "An Iterative Approach to the Hand-Eye and Base-World Calibration Problem," IEEE International Conference on Robotics and Automation, May 2001., pp. 2171-76. USA.

[5] Horaud, R., and Dornaika, F., "Hand-eye calibration," International Journal of Robotics Research, vol.14, no.3, pp.195-210, June 1995.

[6] Motai, Y., and Kosaka, A., "SmartView: Hand-Eye Robotic Calibration for Active Viewpoint Generation and Object Grasping," IEEE International Conference on Robotics and Automation, vol. 1, May 2001, pp. 2185-90. USA.

[7] Li, M., "Kinematic calibration of an active head-eye system," IEEE Transactions on Robotics & Automation, vol.15, no.1, Feb. 1998, pp. 153-158, USA.

[8] Ma, S. D., "A self-calibration technique for active vision systems," IEEE Transactions on Robotics & Automation, vol.12, no.1, Feb. 1996, pp. 114-120, USA.

[9] Rahardja, Krisnawan, "Vision-Based Bin-Picking: Recognition and Localization of Multiple Complex Objects Using Simple Visual Cues", Master Thesis, Purdue University, May 1996.

[10] Shiu, Y. C., and Ahmad, S., "Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form AX=XB", IEEE Transactions on Robotics & Automation, vol.5, no.1, Feb. 1989, pp.16-29. USA.

[11] Tsai, R. Y., and Lenz, R. K., "A new technique for fully autonomous and efficient 3-D robotics hand/eye calibration", IEEE Transactions on Robotics & Automation, vol.5, no.3, June 1989, pp.345-358. USA.

[12] Wang C-C "Extrinsic calibration of a vision sensor mounted on a robot", IEEE Transactions on Robotics & Automation, vol.8, no.2, April 1992, pp.161-75. USA.

[13] Weng J, Cohen P, and Herniou M. "Camera calibration with distortion models and accuracy evaluation", IEEE Transactions on Pattern Analysis & Machine Intelligence, vol.14, no.10, Oct. 1992, pp.965-80. USA.