A dissertation for achieving the doctor degree from the faculty of

Mathematics, Informatics and Natural Sciences at the University of Hamburg

High-Precision Calibration Approaches to Robot Vision Systems

Fangwu Shu

Department of Informatics

University of Hamburg

Presented by:	Fangwu Shu Matr. No. 5676685
Reviewed by:	Prof. Dr. Jianwei Zhang Dr. Peer Stelldinger Prof. Dr. Alois Knoll

Date of the disputation: 11 November, 2009

I declare, that I have done all the work by myself and received no other kinds of help except for the knowledge referred in the bibliography.

Signature of the Author, Signing Date

Abstract

Motivated by the practical issues in applying the vision systems to the industrial robot vision applications, the dissertation has made great efforts on camera calibration, camera recalibration, vision systems calibration and pose estimation.

Firstly, the calibration methods from Tsai are analyzed and improved by solving the degenerate and extent situations. Since the image origin is not calibrated in Tsai methods, the direct linear method and the calibration with vanishing points are referred. They calibrate the image origin but neglect the lens distortion. The situation in practice is that the lens distortion is more sensitive than the image origin to pose estimation and it is difficult to give an initial guess to implement the distortion alignment. Therefore, a direct search algorithm for the image origin is introduced by use of the other camera parameters. Finally, the refinement with nonlinear minimization for all camera parameters comes into the discussing sight.

After the settle down of the mathematical issues in camera calibration, some approaches to online calibration are proposed according to the application environments. The calibration with a robot tool and with a calibration body are the alternative solutions for the robot vision applications. Taking further the application procedure into account, an approach to camera pose calibration with an external measurement system is introduced.

When the applications in industries are given more concerns, the camera recalibration needs to be considered. Since the camera is disturbed by its pose in most of cases, the recalibration is simplified to determine the changes happened to the camera pose. Three recalibration approaches are proposed for checking the changes and determining the corrections of the camera pose in real time.

Eventually, some contributions on vision systems calibration and pose estimation are made according to the applications. Although the application with a mono-camera system and the calibration to a stereo sensor are discussed in details, the dominating target is the multi-camera system. Some valuable approaches, including pattern weight,zero measurement, pattern compensation and security control, to improve the system performances in industrial applications are therefore brought forward.

All the methods and approaches referred in the dissertation aim at applying the vision systems accurately and efficiently to the robot vision applications. They relate fruitfully the techniques in laboratory to the industrial applications.

Acknowledgment

Firstly, I would like to thank my supervisor Prof. Dr. Jianwei Zhang, who gave me the opportunity to start my doctoral work and whose precise attitude to study and research affected me greatly and kind encouragement helped me much in completing this dissertation. I am thankful to Dr. Peer Stelldinger, who helped me much by giving me some pertinent advices to improve the dissertation. Thanks also go to Ms. Tatjana Tetsis, who helped me a lot as well.

I am indebted to Prof. Dr. Werner Neddermeyer, who guided me into the research field of the computer vision and supplied me many related projects so that I had opportunities to study, research and test my ideas in practice. I am also thankful to Prof. Dr. Wolfgang Winkler, who helped me much on both work and life during my stay in Gelsenkirchen.

I would like to take this opportunity to thank Dr. Liviu Toma and Dr. Angela Lilinthal. Dr. Toma is a good friend and partner, we had worked together for many years in many projects. Dr. Lilinthal was kindly and helpful when I met problems in mathematics.

Thanks also go to the Chinese colleagues, who have ever carried out their graduation projects in the laboratory of informatics faculty in Fachhochschule Gelsenkirchen, they are Wei Wei, Yuan Xia, Xiaojing Li, Lei Zhang and Fang Yuan.

In addition, I would like to extend my thanks to Ms. Shun Wang and Ms. Ying Zhu, whose careful examination makes this dissertation out of English style, grammar and spelling mistakes, and Ms. Lifang Xu, who has improved the quality of the figures in the dissertations.

Especially, I am full of heartfelt gratitude to my family and clearly remember all that they did for me during that period. Without their understanding and support, I have no possibility to complete the dissertation so well.

Contents

Chapter 1

Introduction

Vision systems are applied more and more often in industries, the industrialization process becomes more and more necessary and imperative. With many years of working experience in vision software developing and industrial projects implementing, some practical approaches to applying vision systems into robot vision applications as well as the improvements in estimation algorithms on camera calibration and measurement are discussed in this dissertation.

1.1 Practical issues

Oriented closely by the applications, our research work focuses on solving the practical issues from the robot vision applications in automotive industry. The issues are mainly from the following fields:

1. Accuracy of camera calibration
   The importance of the accuracy of camera calibration is obvious, since the uncertainty in calibration is inherited permanently by the vision system and always does its work in measurements. The uncertainty can be from the improper calibration methods as well as the inaccurate calibration data, such as the inaccuracy of *tsai* methods [7, 8] is caused by taking the image center as the image origin; the inaccuracy of the direct linear method [31] is from neglecting the lens distortion; the accuracy of *zhang* method [48] as well as other methods from the *vanishing points* is dependent too much on the accuracy of the image processing; the accuracy of the nonlinear minimization method may be affected by interactions between the camera parameters.

2. Efficiency and convenience on site
   Camera calibration on site is different from calibration in laboratory because of the different working environments. Efficiency indicates the quality of the calibration results and convenience means the complexity of the calibration procedure. An efficient and convenient calibration on site should make full use of the environment and need as few as possible additional equipments, but educe the accurate and stable calibration results.

3. Accuracy of measurement
   The vision systems referred in the dissertation are applied to robot vision applications, where the measurement task is to determine the pose of the work objects

and the accuracy is the issue, especially when the work object is relatively large. The inaccuracy in pose estimation is possible from camera calibration, estimation algorithm, pattern recognition or the non-rigidity of the work object. The error caused from the former two sources is usually called system error. In most of cases, a vision system is not accurate enough for the robot vision applications till the system error is removed away.

4. Stability and security of vision systems
When a vision system is applied to industrial applications, the stability and security have to be discussed. Much work must be done to prevent from the instability or mistakes caused from the errors in pattern recognition, disturb to camera poses, and so on.

1.2 Dissertation aims

Motivated by the above issues, the dissertation aims at the new ideas, better tools, proper designs, improvements or refined solutions on calibration, measurement or recalibration. They may not have the newest or best techniques, but they must satisfy well the application requirements; they may not be the simplest or easiest to implement, but they must be practical and economical in industries. They are outlined as follows:

1. Analyze respectively the calibration methods applied most frequently to practice, find out their advantages, disadvantages and applying situations, test their degenerate configurations and make improvements if possible.

2. Since some calibration methods calibrate only part of the camera parameters, develop some additional algorithms for estimating the uncalibrated parameters as a complementarity procedure in calibration.

3. Make clear of the interactions between camera parameters in estimation and propose some appropriate combination solutions for accurate and complete calibrations.

4. Develop some practical approaches for different types of applications by introducing some appropriate tools.

5. According to the specific working environments, some strategies are to be introduced to improve the performances of the vision systems in robot vision applications.

6. Develop some online approaches to check whether the camera pose is disturbed and estimate the correction to the camera pose if any change really happens.

7. Introduce some typical applications where some of the methods or approaches proposed in the dissertation are applied and tested.

8. All the methods or approaches proposed in this dissertation must be programmed, tested in laboratory and the stable and valuable ones are to be integrated into the vision systems for industrial applications.

Generally speaking, the aim of the dissertation is to do contributions for applying the vision systems more accurately and efficiently to the robot vision applications.

1.3 Notation description

In this dissertation, there are hundreds of symbols and equations for describing the constraints within many kinds of variables. For better understanding and quoting, some general rules are followed by the notations.

1.3.1 Camera parameters

Camera parameters include internal and external parameters and have settled symbols throughout this dissertation.

A. Internal parameters

To describe the camera projection, the following parameters are needed

1. f: the focal length of the camera lens;
2. S_x, S_y: the pixel size on camera chip;
3. C_x, C_y: the intersection of the optical axis and the camera chip plane;
4. k: the scale factor of the radial distortion of the camera lens.

In mathematics these six variables are condensed into five parameters

1. f_x, f_y: the scale factor for transferring millimeters into pixels;
2. C_x, C_y: the origin of the image frame;
3. K: the magnified distortion scale factor from k.

When the lens distortion is neglected, the internal camera parameters can be included into a matrix denoted as A

$$A = \begin{pmatrix} f_x & 0 & C_x \\ 0 & f_y & C_y \\ 0 & 0 & 1 \end{pmatrix}$$

B. External parameters

The external camera parameters are the six elements of a transformation between coordinate frames

1. x, y, z: the translation elements and sometimes denoted as t_x, t_y, t_z;
2. rx, ry, rz: the rotation elements and sometimes denoted as α, β, γ.

When expressing in equations of matrix forms, they are usually denoted as R and $\vec{t}$, or as a whole T for homogeneous coordinates

$$(R, \vec{t}) \Longleftrightarrow \begin{pmatrix} R & \vec{t} \\ 0 & 1 \end{pmatrix} = T$$

1.3.2 Point and pixel

Points in space and pixels in image and their coordinates are denoted with a subscript number

$$\vec{p_i} = (x_i, y_i, z_i)^T \qquad \vec{P_i} = (X_i, Y_i)^T$$

For representing the homogeneous coordinates, they are denoted as

$$\vec{p_i} = (x_i, y_i, z_i, 1)^T \qquad \vec{P_i} = (X_i, Y_i, 1)^T$$

A point or pixel in a certain coordinate frame, e.g. the camera frame C, is usually denoted with a superscript name

$$^c\vec{p_i} = \begin{pmatrix} ^cx_i \\ ^cy_i \\ ^cz_i \end{pmatrix} = \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}^c \qquad ^c\vec{P_i} = \begin{pmatrix} ^cX_i \\ ^cY_i \end{pmatrix} = \begin{pmatrix} X_i \\ Y_i \end{pmatrix}^c$$

1.3.3 Matrix, vector and coordinate frame

A matrix is denoted usually as a capital letter and a vector as a letter with an arrow on the top.

1. J: the coefficients matrix from an over-determined system and the element at row i and column j is denoted as J_{ij};

2. $\vec{x}$: the vector of unknowns from an over-determined system and the element at position i is denoted as x_i.

A matrix in this dissertation represents more often the transformation between coordinate frames.

1. ^AR_B: the rotation transformation from frame A to frame B, a 3 × 3 orthogonal matrix;

2. ^AT_B: the complete transformation from frame A to frame B, a 4 × 4 homography.

1.3.4 Others

1. $F(x, y, \cdots)$: a function with the unknowns $x, y, \cdots$;

2. Ω_∞: the absolute conic in projection space;

3. l_∞, π_∞: the line, plane at infinity in projection space;

4. (V_i, V_j): a pair of vanishing points from orthogonal directions.

1.4 Dissertation outline

This dissertation is structured by 5 chapters and here is the chapter 1 for a general introduction. Chapter 2 constructs a camera model, which is used throughout the dissertation, and the camera parameters from this model are well explained in both mathematics and real projection principle. Several calibration algorithms are described in chapter 3 for determining all or parts of the camera parameters. Most of time, these algorithms should be combined to carry out a complete and accurate calibration. For applying these calibration techniques into practices, some practical approaches are proposed in chapter 4. These approaches may use different tools or setups in calibration procedure according to the different applying environments and objectives. Finally, chapter 5 introduces some vision systems applied in robot vision applications, whose measuring tasks, measuring algorithms and some issues for applying themselves into applications are discussed in detail, and the techniques referred in foregoing chapters are tested in the researching or industrial applications.

Chapter 2

Camera Model

Camera is the basic element for computer vision. To model a camera is to describe in mathematics how the camera projects a visible object point into a corespondent image pixel on the camera chip.

2.1 Camera projection

To describe the projection procedure of the camera lens in mathematics, the following coordinate frames, as shown in the below figure, are defined

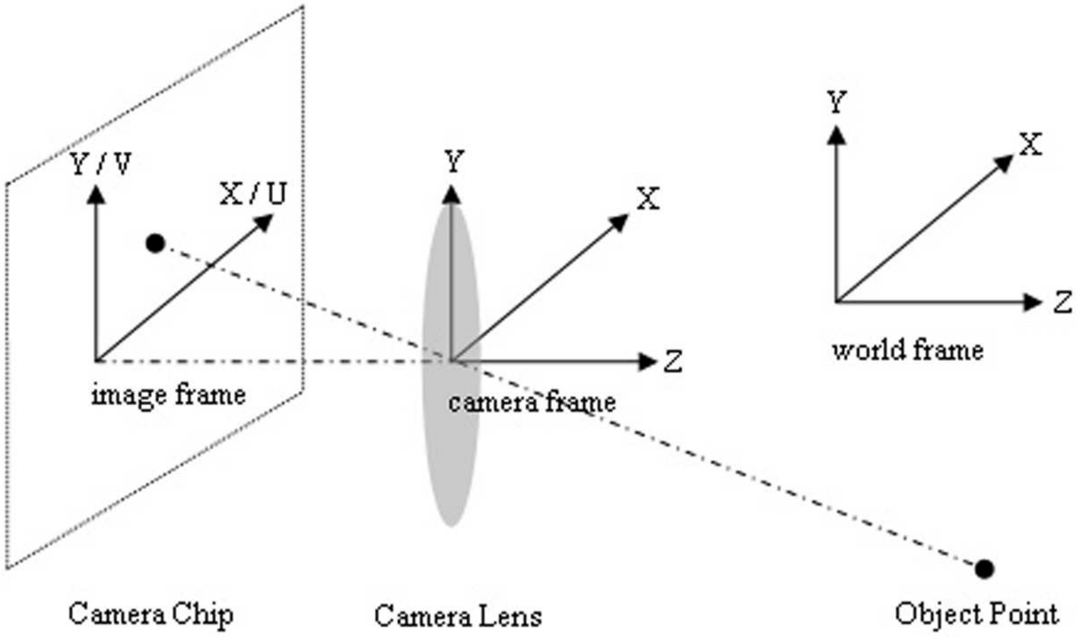


Figure 2.1: camera projection in mathematics

1. **the world frame**: the user defined unique reference frame;

2. **the image frame**: the 2D image coordinate frame centered the intersection of the optical axis and the camera chip plane.

3. **the camera frame**: the projection frame with the origin lying at the optical center point of the camera lens, z-axis pointing out against the camera chip and the other two axises are so defined that their directions are the same respectively as those of the image frame;

As seen from the above figure, the object point is projected into its corresponding image pixel along a ray passing through the optical center point of the camera lens. Therefore, here yield the following perspective equations

$$\frac{u}{f} = \frac{x_c}{z_c} \tag{2.1}$$

$$\frac{v}{f} = \frac{y_c}{z_c} \tag{2.2}$$

where f is the focal length of the camera lens, (x_c, y_c, z_c) are the coordinates of the object points in camera frame and (u, v) are the coordinates of the corresponding image pixel in image frame. Since (x_c, y_c, z_c) have the unit of millimeters, (u, v) must have the same unit. However, the image coordinates are usually denoted with unit of pixel. For converting pixels into millimeters, S_x, S_y are defined to denote the pixel size respectively in x- and y-axis of the pixels array on the camera chip. Let (X, Y) denote the image coordinates in pixels and yield

$$u = XS_x \tag{2.3}$$

$$v = YS_y \tag{2.4}$$

Substituting u, v with X, Y and S_x, S_y, one can find that there are only two independent parameters from S_x, S_y and f. One can simply verify as follows: if $\{S_x, S_y, f\}$ is a set of solution, $\{\lambda S_x, \lambda S_y, \lambda f\}$ with λ being an arbitrary non-zero factor must be another set of solution to satisfy the projection relations, namely

$$\frac{X \cdot \lambda S_x}{\lambda f} \equiv \frac{u}{f} = \frac{x_c}{z_c} \tag{2.5}$$

$$\frac{Y \cdot \lambda S_y}{\lambda f} \equiv \frac{v}{f} = \frac{y_c}{z_c} \tag{2.6}$$

In order to make the calibration procedure stable, the following two parameters are introduced

$$f_x \equiv f/S_x \tag{2.7}$$

$$f_y \equiv f/S_y \tag{2.8}$$

Then let's look into the image frame. As defined above, the image coordinates (X, Y) is with respect to image frame, whose origin is the intersection of the optical axis and the camera chip plane. However, an actual digital image for computer vision has its own image coordinates, which is not the same as defined above. Moreover, the intersection is dependent upon not only the camera and the lens, but also the mounting situation of the lens to the camera. Therefore, it is necessary to calibrate the image origin of the image frame. If the image origin is denoted as (C_x, C_y) and (X, Y) denote again the actual image coordinates, the projection procedure in camera frame can be described as

$$\frac{X - C_x}{f_x} = \frac{x_c}{z_c} \tag{2.9}$$

$$\frac{Y - C_y}{f_y} = \frac{y_c}{z_c} \tag{2.10}$$

2.2 Camera model

As *Faugeras* described in [52], an ordinary model for a pinhole camera can be written into matrix form as following

$$\delta \vec{P} = A\vec{p} \tag{2.11}$$

where δ is an arbitrary scale factor; $\vec{p}$ is an object point and $\vec{P}$ is the corresponding image projection; the projective matrix A, whose elements are called camera internal parameters, characterizes the properties of the camera optics and is given by

$$A = \begin{pmatrix} f_x & \gamma & C_x \\ 0 & f_y & C_y \\ 0 & 0 & 1 \end{pmatrix} \tag{2.12}$$

where (C_x, C_y) is the intersection of the optical axis with the camera chip, also named as image origin; f_x and f_y are the scale factors, which will transfer object millimeters into image pixels in x- and y-axis respectively; γ describes the skewness between the two directions of the pixels array on the chip and is determined only by the manufacturer of the camera. For a qualified camera used in industries, the skewness is often small enough to be neglected. Thus the camera model referred in the dissertation is with zero skewness.

The above camera model simply takes the camera frame as the world frame. In practice, the world frame is usually defined different from the camera frame, e.g. in a multi-camera vision system, which poses another task for camera calibration: determine the transformation $[R, \vec{t}]$ between the camera frame and the world frame. Since $[R, \vec{t}]$ describes the camera pose with respect to an external coordinate frame, whose elements are also named camera external parameters. With both the internal and the external parameters, the camera model is described as

$$\delta \vec{P} = A[R, \vec{t}]\vec{p} \tag{2.13}$$

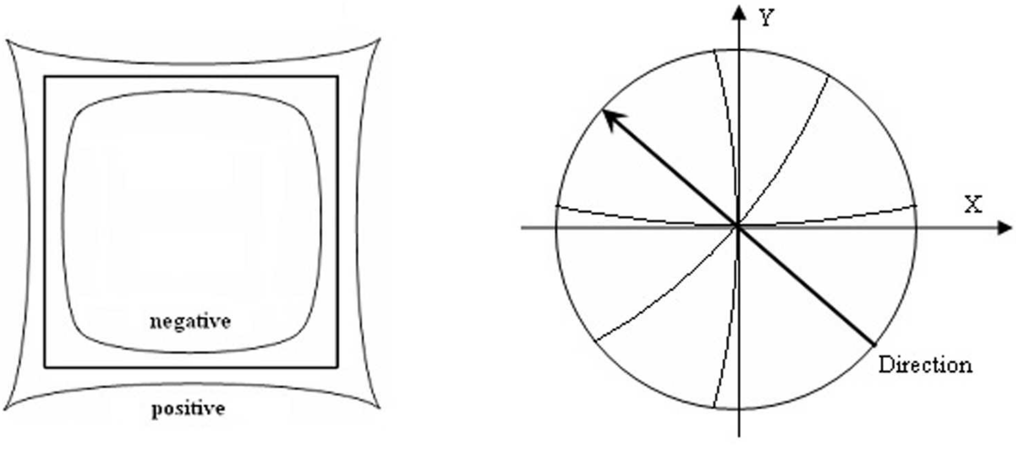
where $\vec{P} = (X, Y, 1)^T$ and $\vec{p} = (x, y, z, 1)^T$ for homogeneous coordinates, but $[R, \vec{t}]\vec{p}$ results a 3-vector of normal coordinates for the consistence of computation.

2.3 Lens distortion

It seems that the above camera model describes the camera projection well. However, the actual cameras do not follow the perfect model, since the lenses in practice have distortions. As *zhuang* concluded in [31], lens distortion can be classified traditionally into radial and tangential distortions. From figure 2.2, one sees that the tangential distortion is much more complex to model in mathematics. Fortunately, many camera calibration researchers have verified experimentally that the radial distortion always takes the dominant effect and the tangential distortion can be neglected in practice. The radial distortion in geometric associates with the position of the image point on the camera chip and is widely considered as following

$$\vec{P_r} = \vec{P_i} + \vec{P_i}(k_1 \cdot |\vec{P_i}|^2 + k_2 \cdot |\vec{P_i}|^4 + \cdots) \tag{2.14}$$

where $\vec{P_i}$ is the ideal pixel and $\vec{P_r}$ is the real pixel.



(a) radial distortion (b) tangential distortion

Figure 2.2: lens distortions in camera projection

If the higher order terms are dropped, one has

$$\vec{P_r} = \vec{P_i}(1 + k \cdot |\vec{P_i}|^2) \tag{2.15}$$

In order to integrate the distortion factor into the camera model expression for calculating, we need to express the ideal pixel $\vec{P_i}$ using $\vec{P_r}$. Considering the fact that $\parallel k \parallel \ll 1$, one can approximate

$$\vec{P_i} = \frac{\vec{P_r}}{1 + k \cdot |\vec{P_r}|^2} \tag{2.16}$$

Experiments show that most cameras used in practice have negative radial distortion as seen from figure 2.2, that is why the calibration results often satisfy $k \leq 0$.
Since the lens distortion is a nonlinear factor, the camera model with distortion is no longer linear. Combining the radial distortion and 2.13, one gets the following two projective equations

$$\frac{1}{1 + k\left((X - C_x)^2 S_x^2 + (Y - C_y)^2 S_y^2\right)} \frac{X - C_x}{f_x} = \frac{x_c}{z_c} \tag{2.17}$$

$$\frac{1}{1 + k\left((X - C_x)^2 S_x^2 + (Y - C_y)^2 S_y^2\right)} \frac{Y - C_y}{f_y} = \frac{y_c}{z_c} \tag{2.18}$$

As discussed before, S_x, S_y, f_x, f_y are not four independent parameters. For consistence reason, the distortion factor is defined as $K = k \cdot f^2$. Rearrange the above equations, the camera model with distortion yields

$$\frac{1}{1 + K\left(\frac{(X - C_x)^2}{{f_x}^2} + \frac{(Y - C_y)^2}{{f_y}^2}\right)} \frac{X - C_x}{f_x} = \frac{x_c}{z_c} \tag{2.19}$$

$$\frac{1}{1 + K\left(\frac{(X - C_x)^2}{{f_x}^2} + \frac{(Y - C_y)^2}{{f_y}^2}\right)} \frac{Y - C_y}{f_y} = \frac{y_c}{z_c} \tag{2.20}$$

where

$$\begin{pmatrix} x_c \\ y_c \\ z_c \\ 1 \end{pmatrix} = \left(R, \vec{t}\right) \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \tag{2.21}$$

Remark: With the new definition of distortion factor by $K = k \cdot f^2$, it satisfies no longer that $\parallel K \parallel \ll 1$. When f is very large, $\parallel K \parallel$ may be even larger than 1.

2.4 Camera calibration

Camera calibration is the procedure to determine all the camera parameters from the applied camera model. For the camera model discussed in the paper, the parameters are as follows

1. **Internal parameters:** f_x, f_y, C_x, C_y, K
2. **External parameters:** $x, y, z, \alpha, \beta, \gamma$

For extracting information from 2D images, the camera calibration is an important and necessary step for all vision systems, especially for the systems to be used in 3D measuring.

2.4.1 Classification of calibration methods

Many valuable calibration methods are reported by the researchers in the field of computer vision. Similar to *zhang*'s opinion in [48], the techniques for camera calibration can be roughly classified into three categories: photogrammetric calibration, calibration from view geometry and self-calibration.

Photogrammetric calibration: This is the traditional direction for camera calibration. The approaches are carried out with a calibration object whose geometry in space is known with good precision. These approaches are normally efficient: stable and accurate results can be expected. However, an expensive calibration object with 3D coordinates and an elaborate setup are usually needed, which makes its applying in practice to some degree difficult.

Calibration from view geometry: With further understanding in view geometry and camera projection, researchers have found some easy calibration methods from vanishing points or circular points, or from pure rotation. The calibration is done only with some orthogonality or parallelity properties in geometry of the calibration model, which makes the pattern recognition much more simple and accurate.

Self-calibration: The techniques need no special calibration object and are very flexible. Just by matching the camera images of a static scene from several positions, the camera is calibrated with fixed intrinsic parameters. Of course disadvantages are also obvious: too many parameters need to be estimated, and one cannot always expect to get reliable results.

In the next chapter, some calibration methods are to be described in detail. Since the aim of the dissertation is the industrial applications, where the most concerned issues are the accuracy and stability, most of the methods are of photogram-metric calibration, and none of them is of self-calibration techniques.

2.4.2 Calibration deviation

The normal way to check how well it is calibrated is to integrate the calibrated cameras into a vision system and do the measurements. However, it is usually an elaborate procedure to set up a vision system and do some test measurements. Actually, when the camera is calibrated, one can easily obtain the deviations in both pixels and millimeters for all calibration points to verify the calibration accuracy.

In the calibration procedure, the world points $\vec{p_i}$ and their corresponding images $\vec{P_i}$, or named ${}^{is}\vec{P_i}$ from pattern recognition, are used to determine the projective matrix A and the camera pose $(R, \vec{t})$. On the contrary way, when the camera is calibrated, the ideal image ${}^{should}\vec{P_i}$ for a calibration point $\vec{p_i}$ can be determined in mathematics. Firstly, the coordinates in the camera frame can be obtained by

$$\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}^{camera} = {}^{c}R_w \cdot \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}^{world} + \vec{t} \tag{2.22}$$

Replacing the above into 2.19 and 2.20 for camera model with distortion, we will get two equations about X_i, Y_i, A and $(R, \vec{t})$. Since the camera is calibrated, the projective matrix A and the camera pose $(R, \vec{t})$ are known and only X_i, Y_i are unnowns, and the equations can be rearranged into

$$F_x(X_i^2, Y_i^2) = 0 \tag{2.23}$$

$$F_y(X_i^2, Y_i^2) = 0 \tag{2.24}$$

Solving X_i and Y_i from the above equations for the ideal pixel ${}^{should}\vec{P_i}$, the deviations in pixels is obtained

$$\Delta\vec{P_i} = {}^{is}\vec{P_i} - {}^{should}\vec{P_i} = (X_i, Y_i)^{is} - (X_i, Y_i)^{should}, \quad i = 1, \cdots, N \tag{2.25}$$

where $\Delta\vec{P_i}$ show how well the the camera parameters match the coordinates of the calibration points, in other words, they show to some extent how well the camera is calibrated.

The above computed deviation is in pixel, which can be converted into millimeter. Looking into the above procedure, the point in the camera frame ${}^{camera}\vec{p_i}$ is obtained, consequently ${}^{camera}z_i$. With the projective matrix A known, the deviation in millimeter can be estimated as follows

$$\Delta\vec{P_i} =^{camera} z_i \cdot \begin{pmatrix} f_x^{-1} & 0 \\ 0 & f_y^{-1} \end{pmatrix} \cdot \Delta\vec{P_i}, \quad i = 1, \cdots, N \tag{2.26}$$

Of course, the distortion factor can also be used to obtain more accurate deviations by using the projective equations 2.19 and 2.20. Since the deviations in millimeter are independent to the focal length and the distance from the camera to the calibration

points, one can understand the calibration errors much easier.
Getting the deviations from all calibration points, one can easily determine how well the camera is calibrated. If a calibration point has large deviations, it must match its image pixel poorly and of course the point-pixel pair has done some negative *contributions* in the calibration procedure. The error source is mostly from the image processing: the pattern is not correctly recognized. In real situation, a reasonable threshold is usually set for checking the deviations. If some points have deviations larger than the threshold, the camera will be recalibrated without these *bad* calibration points. The procedure is repeated till all deviations are small enough.

Remark: Since the equations F_x, F_y have 2^{nd} order terms of X_i, Y_i, they may have two values for X_i and Y_i in solutions. The correct solutions for X_i and Y_i are the values with the same signs as x_c and y_c respectively.

2.5 Chapter review

Although the camera model adopted in this dissertation is referred by many researchers, the below points should be paid more attention to in order to understand the calibration and measurement approaches proposed later in the dissertation better.

1. Parameter sets f_x, f_y and f, S_x, S_y
   For this camera model only five internal parameters f_x, f_y, C_x, C_y and k are needed in mathematics but six parameters f, S_x, S_y, C_x, C_y and k are necessary to characterize a camera in practice. Our solution is to get the value for S_y from the camera specifications and take it as a constant, then

$$\begin{aligned} f &= S_y * f_y & (2.27) \\ S_x &= S_y * f_y / f_x & (2.28) \end{aligned}$$

   With the above equations, f_x, f_y and f, S_x are uniquely determined to each other.

2. Lens distortion
   After many times of test both in laboratory and at work site, a single coefficient k describes well the radial distortion of the camera lens. In order to describe simplifier and clearer in mathematics, a new coefficient K is defined as follows

$$K = k * f^2 \tag{2.29}$$

   With this definition, K may not satisfy $\parallel K \parallel \ll 1$.

3. Calibration deviation
   In most of articles on camera calibration, the stabilities of the calibration results from the same camera are observed to check the validity and reliability of the calibration method. It is very effective for testing in laboratory. However, a camera is usually calibrated once at working site and the stability checking can not be applied. Thus the calibration deviation is introduced for checking the compatibility of the calibration data and the camera parameters. Further more, the deviations can be transferred into millimeters for easier understanding. If all the calibration deviations from the valid calibration points are small enough, the estimated camera parameters match the calibration data well and the camera must be calibrated accurately.

Chapter 3

Analyze and Improvements of the Calibration Methods

In this chapter, some appropriate algorithms for camera calibration are analyzed and some improvements are done. The two calibration methods from Tsai [7, 8] are efficient, but the image origin is paid no attention to. To estimate the image origin, a direct linear method is proposed, which takes the image origin into consideration but neglects the lens distortion. Further in this direction, an algorithm from vanishing points in projection geometry is studied. Considering of the importance of the lens distortion to measurements, a direct searching technique for the image origin is then introduced. Finally, a properly designed minimization algorithm is discussed to refine all camera parameters in an integrated procedure.

3.1 Calibration with coplanar points

In most of practices, the calibration points are given in an array in a plane. In fact, this is the simplest way to supply more as possible known calibration points and the array of points are normally given in a calibration board. With the calibration board lying in one position in the camera sight, the camera can be calibrated if the image origin C_x, C_y are known and $f_x = f_y$.
Without any loss of generality, the calibration points are assumed to locate in the plane $z = 0$ of the world coordinate frame. Denote

$$\begin{aligned} X &= X - C_x && (3.1)\\ Y &= Y - C_y && (3.2)\\ f &= f_x = f_y && (3.3)\\ K &= K/f && (3.4) \end{aligned}$$

To replace the above expressions into equations 2.19 and 2.20 for a camera model with distortion,

$$\frac{X}{f + K\left(X^2 + Y^2\right)} = \frac{x_c}{z_c} \tag{3.5}$$

$$\frac{Y}{f + K\left(X^2 + Y^2\right)} = \frac{y_c}{z_c} \tag{3.6}$$

where

$$\begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = \left(R, \vec{t}\right) \begin{pmatrix} x \\ y \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} r_1 & r_2 & t_x \\ r_4 & r_5 & t_y \\ r_7 & r_8 & t_z \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{3.7}$$

3.1.1 Solving the calibration

According to the current situation, the task to calibrate a camera is to estimate the focal length f and radial distortion factor K as well as the camera external parameters $\left(R, \vec{t}\right)$. The procedure is below:

Calculation for an intermediate vector $\vec{v}$

Dividing both sides of equations 3.5 and 3.6, one gets

$$\frac{X}{Y} = \frac{x_c}{y_c} = \frac{r_1 x + r_2 y + t_x}{r_4 x + r_5 y + t_y} \tag{3.8}$$

If $t_y \neq 0$, we define

$$\vec{v} = \frac{(r_1, r_2, t_x, r_4, r_5)}{t_y} \tag{3.9}$$

Substituting the above definition into equation 3.8 and rearranging the resulting expression, yields to

$$Yxv_1 + Yyv_2 + Yv_3 - Xxv_4 - Xyv_5 = X \tag{3.10}$$

For every calibration point, one can obtain such an equation. With no less than 5 non-collinear calibration points, a set of solutions for the linear equations of $\vec{v}$ will be obtained. Practically we have much more calibration points to create an over-determined system, which can be solved by a linear least-squares algorithm.

Remark: When t_y is zero and t_x is non-zero, the vector $\vec{v}$ can be defined by t_x instead of t_y. If t_x and t_y are both zero, either the camera or the calibration plane should be rearranged; otherwise the calibration procedure fails. In practice, the projection (O_x, O_y) of the origin of the world coordinate frame on the image plane may be checked if it is near the image center. If $|O_x| > |O_y|$, t_x will be chosen, otherwise t_y will be selected.

Calculation for R, t_x and t_y

Now we have got the intermediate vector $\vec{v}$. However, what we want are R, t_x and t_y. If t_y is obtained, the others can be easy computed. Let's look into the rotation matrix R in RPY form

$$\begin{matrix} \cos\beta\cos\gamma & \sin\alpha\sin\beta\cos\gamma - \cos\alpha\sin\gamma & \cos\alpha\sin\beta\cos\gamma + \sin\alpha\sin\gamma \\ \cos\beta\sin\gamma & \sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma & \cos\alpha\sin\beta\sin\gamma - \sin\alpha\cos\gamma \\ -\sin\beta & \sin\alpha\cos\beta & \cos\alpha\cos\beta \end{matrix} \tag{3.11}$$

Looking into 3.11 and contrasting to 3.7, we have

$$\sqrt{(r_1+r_5)^2+(r_2-r_4)^2}+\sqrt{(r_1-r_5)^2+(r_2+r_4)^2} \equiv 2 \tag{3.12}$$

Substituting the intermediate vector $\vec{v}$, yields

$$|t_y| = \frac{2}{\sqrt{(v_1+v_5)^2+(v_2-v_4)^2}+\sqrt{(v_1-v_5)^2+(v_2+v_4)^2}} \tag{3.13}$$

At this moment, the sign of t_y cannot be decided. At first it is supposed to be plus, which will be verified later. Knowing t_y, one can calculate r_1, r_2, t_x, r_4, r_5 from vector $\vec{v}$ as follows

$$(r_1, r_2, t_x, r_4, r_5) = \vec{v} \cdot t_y \tag{3.14}$$

Since the z-axis from the camera frame is defined to be from the camera lens toward outside and no objects behind the camera can be seen, z_c must be positive. Looking back again into 3.5 and 3.6, the signs of X and x_c as well as Y and y_c should be consistent. At the same time, x_c and y_c can now be obtained by

$$x_c = r_1 x + r_2 y + t_x \tag{3.15}$$

$$y_c = r_4 x + r_5 y + t_y \tag{3.16}$$

Therefore, if both $sign(x_c) = sign(X)$ and $sign(y_c) = sign(Y)$, $t_y > 0$ is correct; otherwise $t_y < 0$, and r_1, r_2, t_x, r_4, r_5 should be reversed accordingly.
For a pure rotation homography, it must be an orthonormal unit 3×3 matrix, and the other elements can be computed as follows

$$\begin{aligned} r_3 &= \pm\sqrt{1-{r_1}^2-{r_2}^2} && (3.17) \\ r_6 &= -sign\left(r_3 \cdot (r_1 r_4 + r_2 r_5)\right)\sqrt{1-{r_4}^2-{r_5}^2} && (3.18) \\ (r_7, r_8, r_9) &= (r_1, r_2, r_3) \times (r_4, r_5, r_6) && (3.19) \end{aligned}$$

where the sign of r_3 is decided by f in the next step.

Remark: The resulting matrix R from above may be not an orthonormal one. It is strongly recommended to apply on R an orthonormalization procedure, which can be found in the appendixes.

Calculation for t_z, f and K

With R, t_x and t_y being known, it is possible to estimate the remaining parameters, t_z, f and K. Starting again from 3.5 and 3.6, substituting all known parameters and rearranging them, it yields

$$x_c \cdot f + x_c \left(X^2 + Y^2 \right) \cdot K - X \cdot t_z = (xr_7 + yr_8) \cdot X \tag{3.20}$$

$$y_c \cdot f + y_c \left(X^2 + Y^2 \right) \cdot K - Y \cdot t_z = (xr_7 + yr_8) \cdot Y \tag{3.21}$$

With more than 2 calibration points, the over-determined system of linear equations can be solved as before for the solution of f, K and t_z, then $K = K \cdot f$.

Remark: Do not forget to check the sign of f. If $f > 0$, things work well, otherwise the first assumption for the sign of r_3 is wrong. The following parameters must be reversed accordingly for above two equations

$$r_3 = -r_3 \quad \rightarrow \quad r_6 = -r_6 \quad \Longrightarrow \quad r_7 = -r_7, r_8 = -r_8 \tag{3.22}$$

$$f = -f, t_z = -t_z, K = -K \quad \Longrightarrow \quad balance \tag{3.23}$$

$$K = (-K) \cdot (-f) = K \cdot f = K \tag{3.24}$$

3.1.2 Extent configuration

The above procedure assumes that all calibration points are lying in the plane $z = 0$ of the world frame. Sometimes in practice it happens that the coplanar points are located in an arbitrary plane, as shown in figure 3.1. The calibration procedure referred from above can not be applied directly. Since the calibration points are coplanar, a virtual board can be defined to hold all the calibration points. A new reference frame, called board frame, is defined as in figure 3.1. If we denote the coordinates of the calibration points in the world frame as $\vec{p_i}$ and the transformation from the world to the board as $[R, \vec{t}]$, the board frame can be set up by using 3 non-collinear points $\vec{p}_o, \vec{p}_a, \vec{p}_b$ as follows

$$\vec{x} \;=\; \vec{v}_{oa}/|\vec{v}_{oa}| \tag{3.25}$$

$$\vec{z} \;=\; \vec{x} \times \vec{v}_{ob}/|\vec{v}_{ob}| \tag{3.26}$$

$$\vec{y} \;=\; \vec{z} \times \vec{x} \tag{3.27}$$

where $\vec{v}_{oa} = \vec{p}_a - \vec{p}_o$ and $\vec{v}_{ob} = \vec{p}_b - \vec{p}_o$. Then the transformation is

$$(R, \vec{t}) = (\vec{x}, \vec{y}, \vec{z}, \vec{p}_o) \tag{3.28}$$

Consequently, the coordinates of the calibration points in the board frame are obtained

$$\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}^{board} = (R, \vec{t})^{-1} \cdot \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}^{world} \tag{3.29}$$

With these coordinates, the calibration can be carried out in the board frame. The camera pose resulting from the calibration can be calculated back with respect to the world frame with $[R, \vec{t}]$.

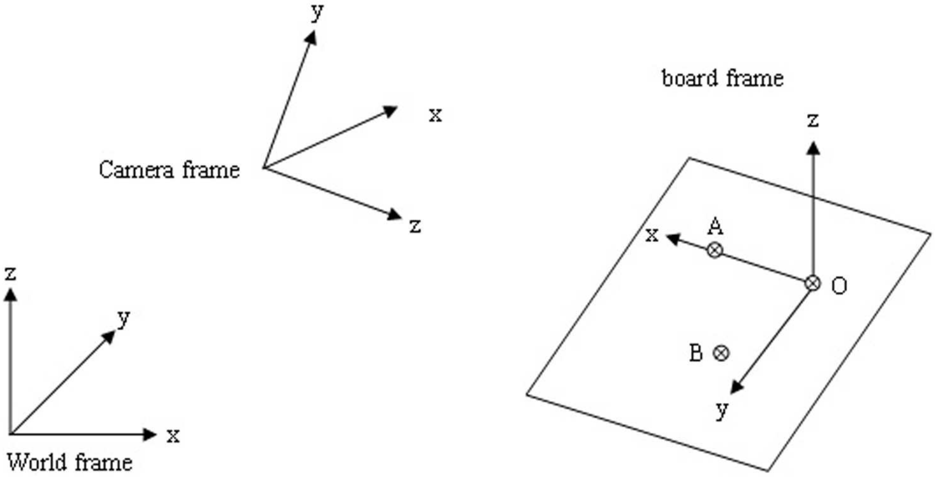


Figure 3.1: extent situation of the coplanar points

3.1.3 Degenerate configuration

If the calibration board is parallel or nearly parallel to the image plane, the calibration algorithm will fail: only the rotation and tx, ty are available, f, k and tz are unpredictable.

If the degenerate configuration occurs, the rotation from camera frame to the world frame will be actually a rotation around z-axis. That is

$$^{c}R_w = \begin{pmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{3.30}$$

where θ is the angle of the rotation. By contrasting the elements, one can easily find that $r_7 = r_8 = 0$, which result the equations 3.20 and 3.21 into

$$x_c \cdot f + x_c \left(X^2 + Y^2\right) \cdot K - X \cdot t_z = 0 \tag{3.31}$$

$$y_c \cdot f + y_c \left(X^2 + Y^2\right) \cdot K - Y \cdot t_z = 0 \tag{3.32}$$

The homogeneous equations result in an inability to solve f, K and t_z uniquely. But if we define $\check{f} = f/t_z$ and $\check{K} = K/t_z$, then

$$x_c \cdot \check{f} + x_c \left(X^2 + Y^2\right) \cdot \check{K} = -X \tag{3.33}$$

$$y_c \cdot \check{f} + y_c \left(X^2 + Y^2\right) \cdot \check{K} = -Y \tag{3.34}$$

where $\check{f}$ and $\check{K}$ can be uniquely solved, consequently

$$f = \check{f} \cdot t_z \tag{3.35}$$

$$k = K \cdot f = \check{K} \cdot t_z^2 \tag{3.36}$$

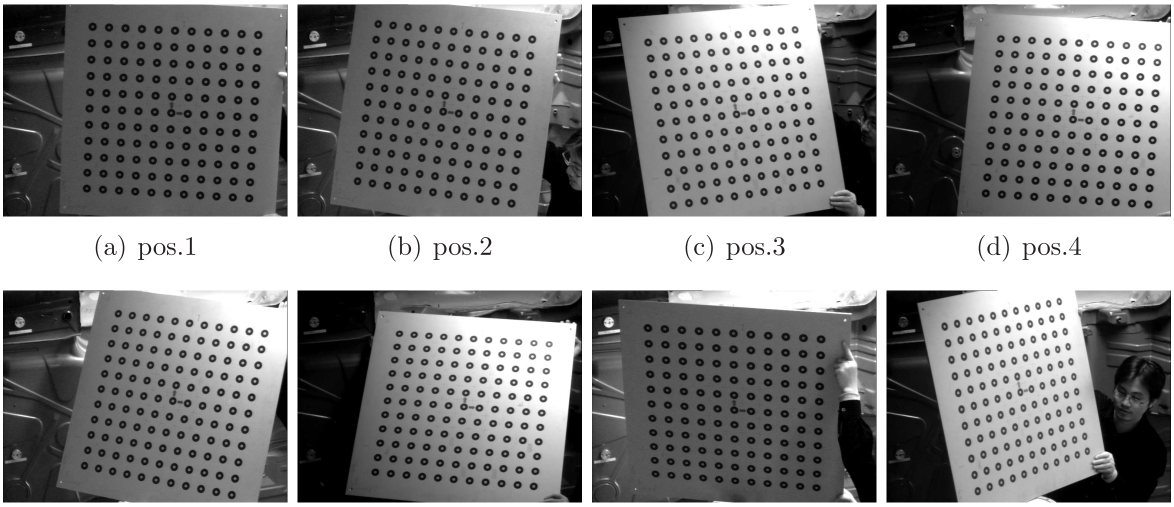
where t_z is an arbitrary positive value and represents the distance from the camera to the calibration board. If the translation in z-direction is given, the calibration is also completed. In practice, it is easy to check the degenerate configuration: select two parallel lines in the model plane. If their image lines are parallel, the direction of the lines is parallel to the image plane. If there are more different directions parallel to the image plane, the calibration board is parallel to the image plane. Otherwise, the calibration succeeds.

3.1.4 Experimental results

The cameras used in the experiments are of JAI M50 camera, which is well introduced in appendix, and the camera lenses are of 25mm focal length.

A. Calibration with the calibration board in normal poses

As shown in figure 3.2, a large calibration board with 11×11 patterns is used in the experiments. By moving the calibration board into such eight positions that the board plane is not nearly parallel to the camera chip plane, the camera is calibrated for eight times.



(a) pos.1 (b) pos.2 (c) pos.3 (d) pos.4

(e) pos.5 (f) pos.6 (g) pos.7 (h) pos.8

Figure 3.2: calibration with calibration board in normal poses

As shown in table 3.1, this method calibrates actually only two internal parameters f and K. The image origin C_x, C_y is simply supposed to be the image center of the image with the dimension of 768 × 572 pixels and f_x and f_y are taken for granted to be equal. For the camera pose, all the six parameters are estimated and the results are shown in table 3.2. Since too few internal parameters are considered and the estimated translation in z direction is too rough, this method is never applied to projects and thus the accuracy of the camera pose is not verified in this experiment.

From figure 3.3, we can see that the deviations of the calibration come to 4mm, which is a little far away from the industrial requirement. However, this method can be widely used for daily applications and satisfy well the general public.

	f_x	f_y	K		S_x=S_y	f	k
Pos.1	3163.97	3163.97	0.293969	→	0.008325	26.34	0.000424
Pos.2	3064.27	3064.27	0.300460	→	0.008325	25.51	0.000462
Pos.3	3173.88	3173.88	0.350217	→	0.008325	26.42	0.000502
Pos.4	2828.39	2828.39	0.275913	→	0.008325	23.55	0.000498
Pos.5	2972.56	2972.56	0.352071	→	0.008325	24.75	0.000575
Pos.6	2775.64	2775.64	0.387363	→	0.008325	23.11	0.000725
Pos.7	2955.27	2955.27	0.323586	→	0.008325	24.60	0.000535
Pos.8	3153.56	3153.56	0.374290	→	0.008325	26.25	0.000543
Aver	3010.94	3010.94	0.332234	→	0.008325	25.07	0.000533
MaxErr	162.94	162.94	0.055129	→	——	1.36	0.000193

Table 3.1: internal parameters with C_x=384 and C_y=286

	X	Y	Z	RX	RY	RZ
pos.1	279.24	641.10	4203.41	171.623	5.439	2.384
pos.2	1205.23	-83.04	4018.43	-176.633	16.580	7.650
pos.3	-1715.00	-461.07	3990.70	-169.950	-21.879	-9.883
pos.4	292.04	-910.97	3784.64	-165.672	5.678	5.007
pos.5	1565.25	-447.64	3755.30	-168.613	22.203	13.155
pos.6	515.75	-1645.52	3553.97	-153.921	6.074	7.873
pos.7	586.90	1923.73	3304.32	150.776	9.723	-0.603
pos.8	-3007.69	419.16	3136.14	-175.065	-44.271	-15.342

Table 3.2: external parameters of the calibrations

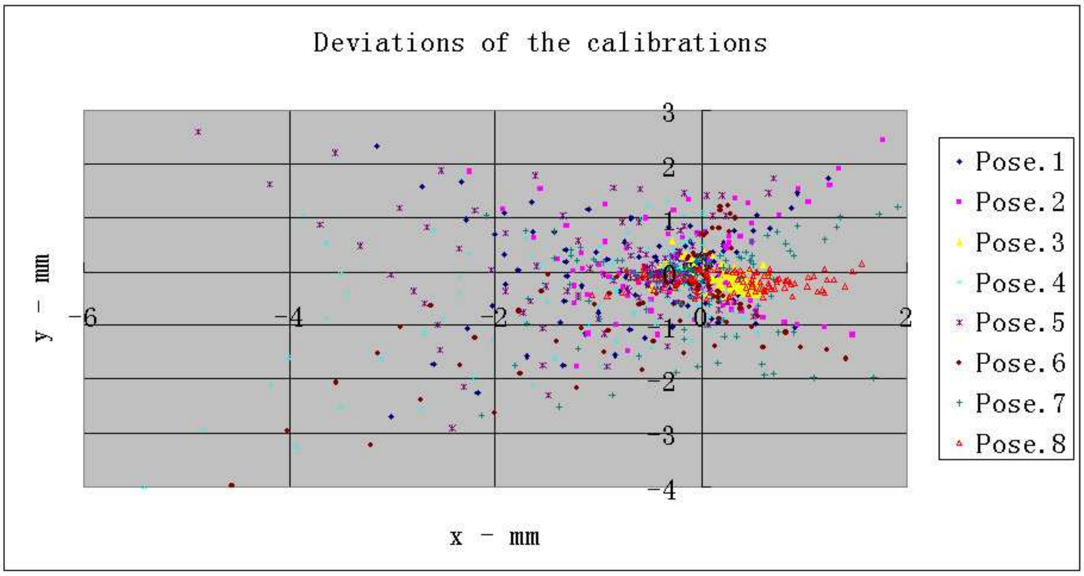


Figure 3.3: calibration deviations from coplanar points

B. Calibration with the board nearly parallel to the image plane

As described before, the calibration will fail or be unstable when the calibration board is nearly parallel to the plane of the image chip. If the calculation is completed, the rotation rx, ry, rz must be correct. Thus the images from figure 3.5 are used separately for calibration as follows

	$f_x = f_y$	K	x	y	z	rx	ry	rz
C1.Pos1	90.32	-0.000553	-5.60	1.54	91.38	177.797	-0.101	0.105
C1.Pos2	30.12	-0.000030	-14.27	-0.48	278.09	177.725	-0.090	0.106
C1.Pos3	1347.99	-0.059182	-42.79	42.34	1393.65	177.850	-1.009	0.122
C2.Pos1	4807.16	0.877158	10.24	226.52	4849.59	177.213	0.084	-0.027
C2.Pos2	3833.73	-0.387725	-3.11	174.08	3804.48	177.059	-0.123	-0.020
C2.Pos3	4375.80	0.607884	13.24	194.88	4197.90	176.866	0.111	-0.024
C3.Pos1	7878.31	2.309121	2.17	411.83	7931.23	176.959	0.044	0.201
C3.Pos2	6871.28	1.471520	7.53	328.73	6607.17	176.964	0.097	0.198
C3.Pos3	5777.78	-1.214248	-6.84	243.33	5373.35	177.054	-0.052	0.204
C4.Pos1	1571.03	0.107269	-17.51	64.62	1600.43	177.158	0.172	0.246
C4.Pos2	1320.65	0.011110	-13.60	48.43	1488.07	177.140	0.147	0.248
C4.Pos3	133.95	-0.003599	-10.04	-17.41	615.72	177.322	0.495	0.261

Table 3.3: calibration results with the board in parallel poses

The rotation values from figure 3.3 will be verified by contrasting to the results from the non-coplanar calibration to be described in the next section.

3.2 Calibration with non-coplanar points

In this section, the camera is calibrated with a set of non-coplanar calibration points, which are arbitrary, e.g. from more boards or the same board in different positions. Same as the procedure with coplanar calibration points, the calibration will be done only when the the image origin is known.
Denoting $f_y = f, f_x = sf$ and $X = X - C_x, Y = Y - C_y$, 2.19 and 2.20 become

$$\frac{1}{1+K\cdot r^2}\cdot\frac{X}{sf} = \frac{x_c}{z_c} \tag{3.37}$$

$$\frac{1}{1+K\cdot r^2}\cdot\frac{Y}{f} = \frac{y_c}{z_c} \tag{3.38}$$

where

$$r^2 = X^2/(s^2f^2) + Y^2/f^2 \tag{3.39}$$

$$\begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = \left(R, \vec{t}\right)\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} r_1 & r_2 & r_3 & t_x \\ r_4 & r_5 & r_6 & t_y \\ r_7 & r_8 & r_9 & t_z \end{pmatrix}\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \tag{3.40}$$

3.2.1 Solving the calibration

The calibration task here is to estimate the focal length f, s and radial distortion factor K as well as the camera external parameters $(R, \vec{t})$.

Calculation for a intermediate vector $\vec{v}$

Similarly dividing both sides of 3.37 and 3.38, one gets

$$\frac{X}{Y} = \frac{sx_c}{y_c} = \frac{s\cdot(r_1x + r_2y + r_3z + t_x)}{r_4x + r_5y + r_6z + t_y} \tag{3.41}$$

Defining a vector $\vec{v}$ in length of 7 as follows

$$\vec{v} = \left(\frac{sr_1, sr_2, sr_3}{t_y}, \frac{r_4, r_5, r_6}{t_y}, \frac{st_x}{t_y}\right) \tag{3.42}$$

Rearranging and rewriting into vector $\vec{v}$, then yields

$$Yxv_1 + Yyv_2 + Yzv_3 - Xxv_4 - Xyv_5 - Xzv_6 + Yv_7 = X \tag{3.43}$$

With no less than 7 of such calibration points, a set of solution for the linear equations of $\vec{v}$ will be obtained.

Calculation for R, t_x and t_y

Considering the orthonormal property of the rotation matrix R, one has

$$r_1^2 + r_2^2 + r_3^2 = 1 \tag{3.44}$$

$$r_4^2 + r_5^2 + r_6^2 = 1 \tag{3.45}$$

Substituting the intermediate vector $\vec{v}$, then yields

$$\begin{aligned} |t_y| &= \frac{1}{\sqrt{v_4^2+v_5^2+v_6^2}} && (3.46) \\ s &= |ty| \cdot \sqrt{v_1^2+v_2^2+v_3^2} && (3.47) \\ (r_1, r_2, r_3, t_x) &= (v_1, v_2, v_3, v_7) \cdot t_y/s && (3.48) \\ (r_4, r_5, r_6) &= (v_4, v_5, v_6) \cdot t_y && (3.49) \end{aligned}$$

The determination of the sign of t_y and remaining calculation for R and t_x are the same as in the coplanar case.

Calculation for t_z, f_x, f_y and K

Coming back to the equations 3.37 and 3.38 and noting $\check{K} = K/(s^2 f)$, it yields

$$\frac{X}{sf + (sX^2 + s^3Y^2)\check{K}} = \frac{x_c}{z_c} \qquad (3.50)$$

$$\frac{Y}{f + (X^2 + s^2Y^2)\check{K}} = \frac{y_c}{z_c} \qquad (3.51)$$

Subsituting all known parameters and rearranging them into below

$$\begin{aligned} sx_c \cdot f + x_c\left(sX^2 + s^3Y^2\right) \cdot \check{K} - X \cdot t_z &= (xr_7 + yr_8 + zr_9) \cdot X && (3.52) \\ y_c \cdot f + y_c\left(X^2 + s^2Y^2\right) \cdot \check{K} - Y \cdot t_z &= (xr_7 + yr_8 + zr_9) \cdot Y && (3.53) \end{aligned}$$

To solve the linear equations system, the solution for f, K and t_z are obtained, then $f_y = f, f_x = sf, K = s^2 f\check{K}$.

3.2.2 Degenerate configuration

There is also a degenerate configuration for calibration with non-coplanar calibration points: **if $n-1$ points from total n calibration points are coplanar, the calibration procedure will fail**.

Let us look into the linear system from 3.43, it will give a unique solution if and only if the coefficient matrix has full column ranks. The calibration procedure with coplanar calibration points shows that coplanar points result in a coefficient matrix with five ranks. When $n-1$ points from total n calibration points are coplanar, the n calibration points will result in the coefficient matrix maximum of six ranks, which will make the system from 3.43 with seven unknowns ill-conditioned. Consequently it results in no unique solution for the intermediate vector $\vec{v}$, and the procedure fails. In practice, the problem is solved by using much more stochastic distributed calibration points in the space where the camera can see.

Remark: when applying an over determined system of linear equations, it may happen that majority *eat* minority, e.g. 50 points are in a plane and only 2 points are out of the plane. When this situation really happens, the calibration with coplanar points may work better. A practical procedure may be as follows: find a best-fit plane for all calibration points; discard the farthest 5 percent points away; see if the left 95 percent points are coplanar: if yes, calibrate the camera with the 95 percent coplanar points; otherwise calibrate the camera with all non-coplanar points.

3.2.3 Experimental results

Similar to the last section, four cameras from JAI M50 with lenses of 25mm focal length and large calibration board are applied in the experiments.

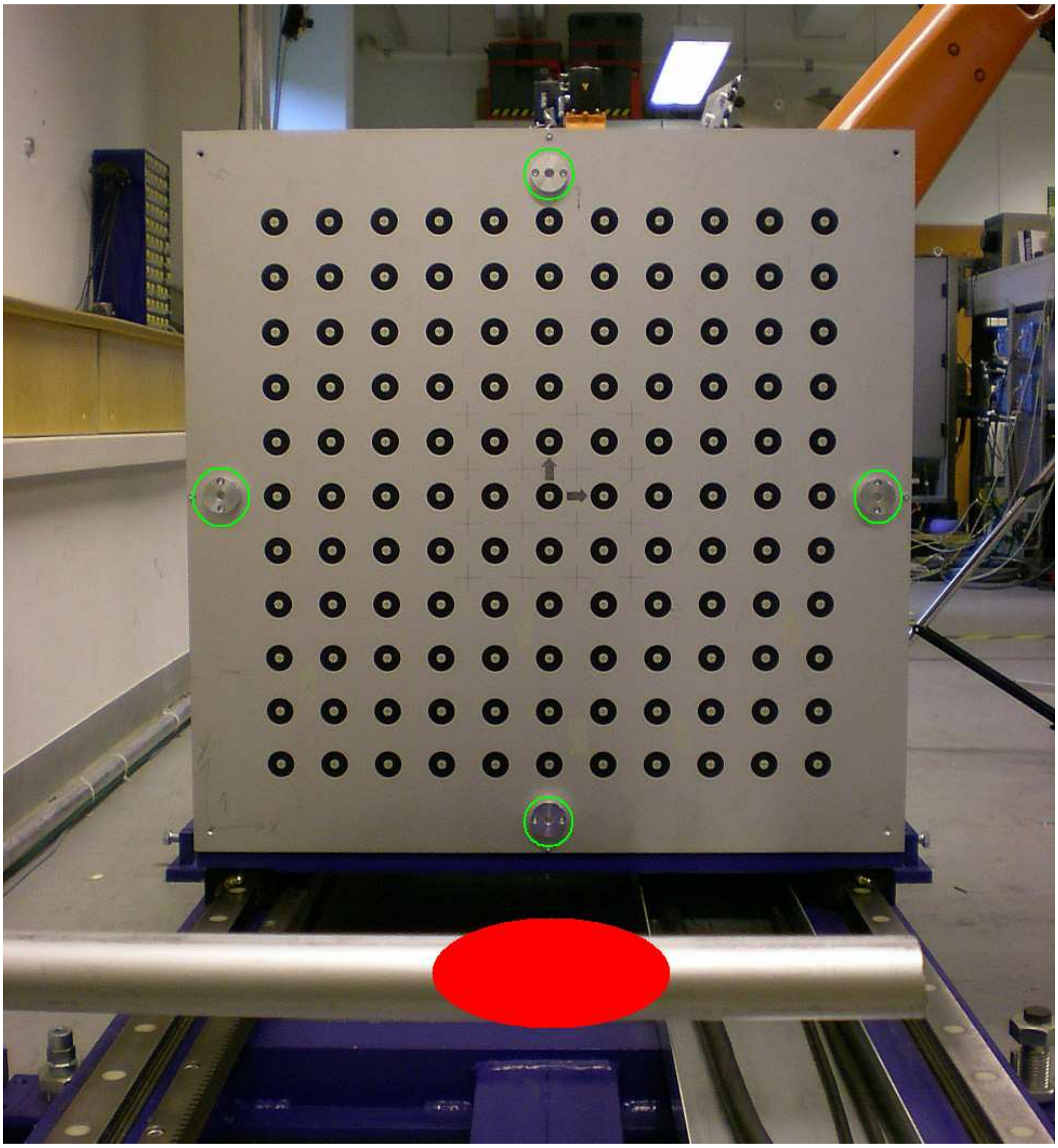
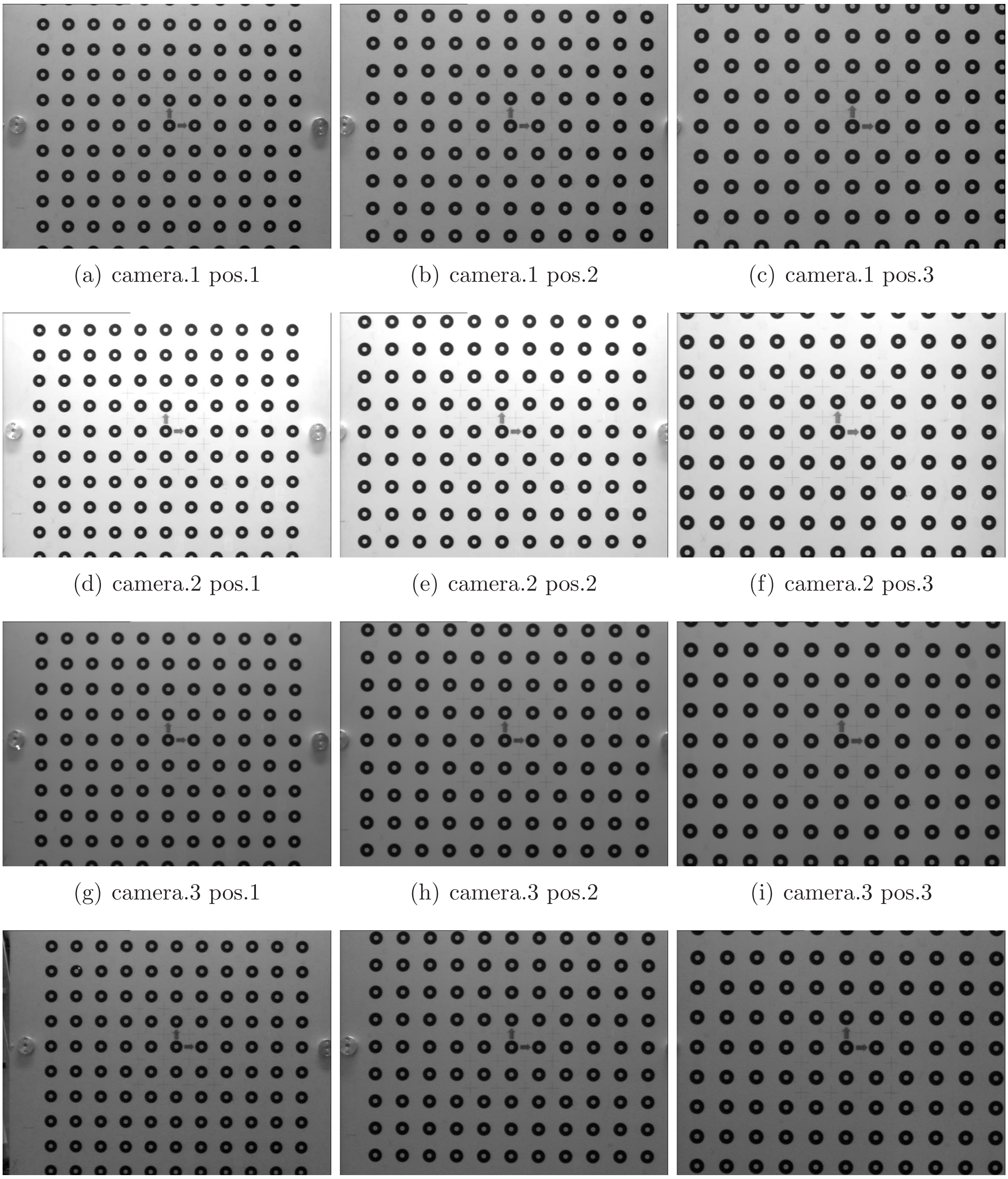


Figure 3.4: calibration setup

As shown in figure 3.4, the board is mounted on the robot base, the location marked with red ellipse is the position for camera mounting and the circles marked in green are the controlling marks to determine the current pose of the calibration board. We have repeated the calibration procedure for 4 times by mounting successively four cameras in front of the calibration board. For each calibration, the board is moved into 3 poses by driving the robot along to the robot linear axis to the 3 taught positions, where the camera *sees* at the board and 2D coordinates are obtained from the corresponding camera images as shown in figure 3.5. At the same time, in order to gather the 3D coordinates of the calibration points in a common frame, the board at each position is measured by a laser tracker system.



(a) camera.1 pos.1

(b) camera.1 pos.2

(c) camera.1 pos.3

(d) camera.2 pos.1

(e) camera.2 pos.2

(f) camera.2 pos.3

(g) camera.3 pos.1

(h) camera.3 pos.2

(i) camera.3 pos.3

(j) camera.4 pos.1

(k) camera.4 pos.2

(l) camera.4 pos.3

Figure 3.5: images of calibration board in vertical poses

				x	y	z	rx	ry	rz
Calibration board position.1				0.00	0.00	0.00	0.000	0.000	0.000
Calibration board position.2				-0.49	-0.67	250.03	-0.026	0.005	-0.001
Calibration board position.3				-1.00	-1.37	500.17	-0.039	0.008	-0.001
	f_x	f_y	K	x	y	z	rx	ry	rz
C1.Pos 12	3037.19	3033.02	-0.300553	-113.60	2.89	3075.72	179.906	-2.014	0.106
C1.Pos 13	3038.76	3034.85	-0.297157	-113.86	2.56	3077.50	179.912	-2.018	0.104
C1.Pos 23	3039.93	3035.91	-0.296909	-114.07	2.56	3078.46	179.912	-2.021	0.105
C1.Pos123	3038.79	3034.74	-0.298088	-113.84	2.63	3077.41	179.911	-2.017	0.105
Average	3038.67	3034.63	-0.298177	-113.84	2.66	3077.27	179.910	-2.018	0.105
MaxErr	1.48	1.61	0.002376	0.24	0.23	1.55	0.004	0.004	0.001
C2.Pos 12	3049.20	3045.41	-0.314686	21.03	2.81	3080.12	179.770	0.335	-0.025
C2.Pos 13	3049.90	3045.98	-0.308901	21.08	2.48	3080.93	179.776	0.335	-0.025
C2.Pos 23	3050.03	3045.86	-0.312536	21.19	2.24	3080.87	179.782	0.338	-0.024
C2.Pos123	3049.87	3045.96	-0.311599	21.07	2.52	3080.82	179.776	0.335	-0.025
Average	3049.75	3045.80	-0.311931	21.09	2.51	3080.69	179.776	0.336	-0.025
MaxErr	0.55	0.39	0.003030	0.10	0.30	0.56	0.006	0.002	0.001
C3.Pos.12	3054.97	3050.73	-0.330825	3.29	3.19	3079.99	179.764	0.108	0.202
C3.Pos 13	3055.96	3051.83	-0.328390	3.22	3.06	3081.10	179.767	0.107	0.202
C3.Pos 23	3056.44	3052.37	-0.332939	3.16	3.01	3081.38	179.768	0.106	0.201
C3.Pos123	3055.91	3051.77	-0.330677	3.23	3.06	3080.96	179.767	0.107	0.202
Average	3055.82	3051.68	-0.330708	3.23	3.08	3080.86	179.767	0.107	0.202
MaxErr	0.85	0.95	0.002318	0.06	0.11	0.87	0.002	0.001	0.001
C4.Pos 12	3022.68	3018.31	-0.312781	45.15	2.57	3082.31	179.682	1.249	0.248
C4.Pos 13	3021.46	3017.22	-0.307858	45.15	2.63	3081.24	179.681	1.249	0.246
C4.Pos 23	3019.66	3015.46	-0.303082	45.18	2.76	3079.82	179.678	1.250	0.247
C4.Pos123	3021.52	3017.23	-0.308850	45.14	2.66	3081.30	179.680	1.249	0.247
Average	3021.33	3017.06	-0.308143	45.16	2.66	3081.17	179.680	1.249	0.247
MaxErr	1.67	1.59	0.005061	0.02	0.11	1.35	0.002	0.001	0.001

Table 3.4: non-coplanar calibration with C_x=384 and C_y=286

Table 3.4 shows the calibration results with 2 or 3 images and we have found that the results are very consistent with each other. To the internal parameters, the maximum errors for f_x or f_y are about 0.05% and about 2% for K; to the camera pose, the maximum erros for x and y are about 0.2mm, about 0.005 grad for rx, ry and rz, but nearly 1.6mm for z. Actually, the maximum error for z is no more than 0.05%. When the camera looks the object too far away, from 2500mm to 3000mm, the deepth information becomes relatively little for the camera to estimate. Of course we can put the calibration board closer to the camera to get better results. However, this dissertation aims at the industrial applications, which have a situation similar to the above experiment. In fact, the accuracy for all parameters including z reaches the industrial requirement, which can be seen from the deviations of the calibrations with 3 images as shown in figure 3.6.

We have not forgot to verify the validity of the rotations from last section with coplanar points in a nearly parallel to the camera chip and the differences are shown in table 3.5.

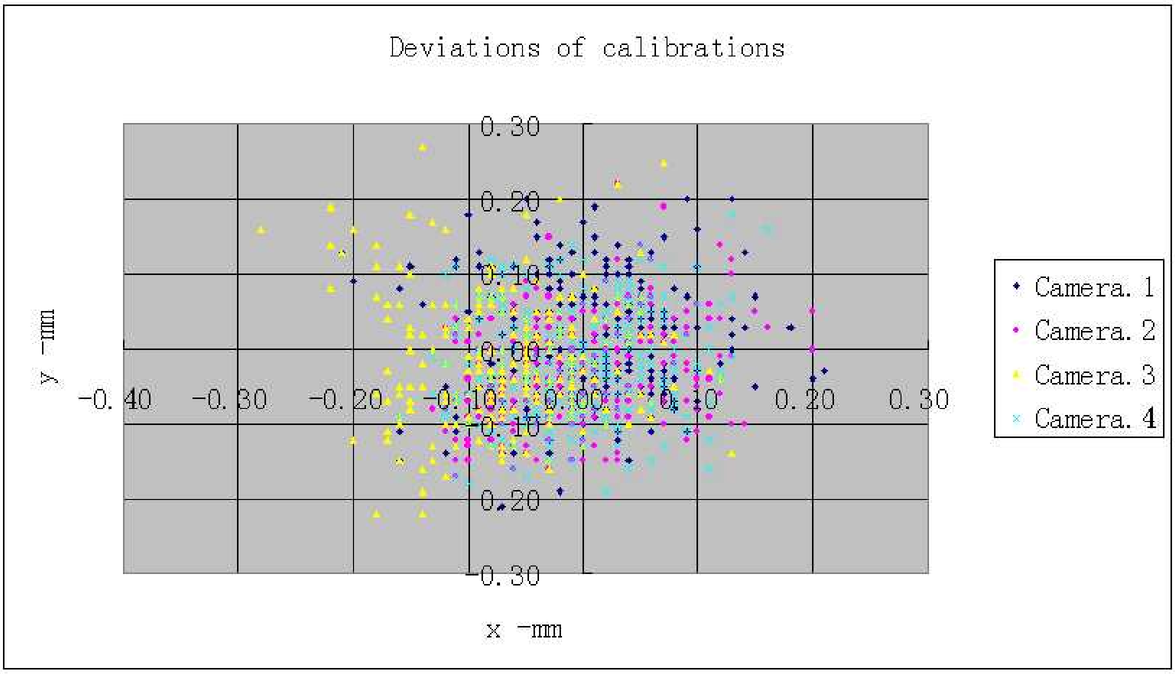


Figure 3.6: calibration deviations from non-coplanar points

	Camera.1			Camera.2		
Coplanar	177.791	-0.400	0.111	177.046	0.024	-0.024
Noncoplanar	179.910	-2.018	0.105	179.776	0.336	-0.025
Difference	2.120	1.618	0.006	2.730	0.312	0.001
	rx	ry	rz	rx	ry	rz
Coplanar	176.992	0.030	0.201	177.207	0.271	0.252
Noncoplanar	179.767	0.107	0.202	179.680	1.249	0.247
Difference	2.774	0.077	0.001	2.474	0.978	0.005
	Camera.3			Camera.4		

Table 3.5: differences between the resulted rotations

3.3 Calibration for a distortion-free model

In this section, a direct calibration method, where the image origin is estimated at the same time, is proposed at the price of radial distortion $k = 0$.

3.3.1 Solving the calibration

For a distortion-free camera model, $K = 0$ yields

$$(r_7 x + r_8 y + r_9 z + t_z) \cdot (X - C_x) - f_x \cdot (r_1 x + r_2 y + r_3 z + t_x) = 0 \tag{3.54}$$
$$(r_7 x + r_8 y + r_9 z + t_z) \cdot (Y - C_y) - f_y \cdot (r_4 x + r_5 y + r_6 z + t_y) = 0 \tag{3.55}$$

The above equations are nonlinear equations, which can not be solved directly. The usual solution as before is to define some mediate variables for linearizing them. Assume that $t_z! = 0$, we define an 11-vector $\vec{v}$ by

$$(v_1, v_2, v_3) = (r_7, r_8, r_9) / t_z \tag{3.56}$$
$$(v_4, v_5, v_6) = (r_7 C_x + r_1 f_x, r_8 C_x + r_2 f_x, r_9 C_x + r_3 f_x) / t_z \tag{3.57}$$
$$(v_7, v_8, v_9) = (r_7 C_y + r_4 f_y, r_8 C_y + r_5 f_y, r_9 C_y + r_6 f_y) / t_z \tag{3.58}$$
$$(v_{10}, v_{11}) = (C_x t_z + f_x t_x, C_y t_z + f_y t_y) / t_z \tag{3.59}$$

Then the above two equations can be rewritten into $\vec{v}$ as follows

$$Xx \cdot v_1 + Xy \cdot v_2 + Xz \cdot v_3 - x \cdot v_4 - y \cdot v_5 - z \cdot v_6 - v_{10} = -X \tag{3.60}$$
$$Yx \cdot v_1 + Yy \cdot v_2 + Yz \cdot v_3 - x \cdot v_7 - y \cdot v_8 - z \cdot v_9 - v_{11} = -Y \tag{3.61}$$

For every calibration point, there are such two equations as above. With more than 5 calibration points, the over-determined linear system can be solved with $\vec{v}$. Once $\vec{v}$ is known, all the wanted parameters can be solved from the vector. With the orthogonal properties of the rotation matrix, the parameters can be calculated as follows

$$t_z = \pm \frac{1}{\sqrt{v_1^2 + v_2^2 + v_3^2}} \tag{3.62}$$
$$C_x = (v_1, v_2, v_3) \cdot (v_4, v_5, v_6)^T \cdot t_z^2 \tag{3.63}$$
$$C_y = (v_1, v_2, v_3) \cdot (v_7, v_8, v_9)^T \cdot t_z^2 \tag{3.64}$$
$$f_x = |\,(v_4, v_5, v_6) - (v_1, v_2, v_3) \cdot C_x| \cdot |t_z| \tag{3.65}$$
$$f_y = |\,(v_7, v_8, v_9) - (v_1, v_2, v_3) \cdot C_y| \cdot |t_z| \tag{3.66}$$
$$t_x = (v_{10} - C_x) \cdot t_z / f_x \tag{3.67}$$
$$t_y = (v_{11} - C_y) \cdot t_z / f_y \tag{3.68}$$
$$(r_1, r_2, r_3) = ((v_4, v_5, v_6) - (v_1, v_2, v_3) \cdot C_x) \cdot t_z / f_x \tag{3.69}$$
$$(r_4, r_5, r_6) = ((v_7, v_8, v_9) - (v_1, v_2, v_3) \cdot C_y) \cdot t_z / f_y \tag{3.70}$$
$$(r_7, r_8, r_9) = (v_1, v_2, v_3) \cdot t_z \tag{3.71}$$

The sign of t_z can be determined as follows:
In camera projection procedure, only the object points lying in front of the camera lens are possible to be projected into corresponding image pixels on camera chip. On the other hand, the camera frame is defined with the origin lying on the optical center of the

lens and the positive direction of z-axis pointing from camera chip toward lens center. Thus all object points have positive z-coordinates with respect to the camera frame, that is

$$\begin{aligned} z &= r_7x + r_8y + r_9z + t_z \\ &= (v_1x + v_2y + v_3z) \cdot t_z + t_z \\ &= (v_1x + v_2y + v_3z + 1) \cdot t_z > 0 \end{aligned} \tag{3.72}$$

Firstly assume that t_z is positive, solve the above solutions and calculate the z-coordinates in the camera frame for all calibration points.
If $z_i > 0, \quad i = 1, \cdots, n$, then $t_z = |t_z|$;
If $z_i < 0, \quad i = 1, \cdots, n$, then $t_z = -|t_z|$;
If $z_i > 0, z_j < 0, \quad 1 < i, j < n$, the calibration procedure fails.

Remark: Since three rows of the rotation matrix are calculated separately, it may not be orthonormal. An orthonormalization procedure is necessary to be applied on the elements. After the orthonormalization, the parameters should be re-estimated for better accuracy.

3.3.2 Degenerate configuration

With the above calibration method, there are two degenerate configurations arising from the calculation procedure:

Configuration 1: if the world frame is defined in such a way that its origin locates exactly or nearly in the plane $z = 0$ of the camera frame, the calibration will fail or become unstable.

If the exact situation happens, the z-coordinate of the transformation from camera frame to world frame is zero $t_z = 0$. However, $t_z \neq 0$ is the precondition for linearization of the system of nonlinear equations and the calibration procedure fails of course with $t_z = 0$. If the origin locates nearly in the plane $z = 0$, that is $t_z \approx 0$, the mediate vector $\vec{v}$ will have very large elements, which will result in unstableness in continuous calculation and consequently the unpredictable solutions. In order to avoid such situations, t_z must be verified: if it is too small, either the calibration object or the camera should be rotated or moved into other positions.

Configuration 2: if all the calibration points are coplanar, the calibration will fail.

Let's look into the coefficient matrix for calculating the mediate vector $\vec{v}$,

$$\begin{pmatrix} X_ix_i & X_iy_i & X_iz_i & -x_i & -y_i & -z_i & 0 & 0 & 0 & -1 & 0 \\ Y_ix_i & Y_iy_i & Y_iz_i & 0 & 0 & 0 & -x_i & -y_i & -z_i & 0 & -1 \end{pmatrix} \tag{3.73}$$

The linear system can be solved with a unique solution for $\vec{v}$, if and only if the above matrix has a full column rank. If all the calibration points are coplanar, the column 4, 5 and 6 will give a maximum of rank 2, the same as column 7, 8 and 9. Thus the linear system is ill-conditioned, no unique solution for $\vec{v}$ and the calibration fails.

3.3.3 Experimental results

The images in figure 3.5 are used again in this experiment and the calibration results are listed in table 3.6, where **i.A** is from the calibration for a distortion-free model and **i.B** is with the non-coplanar points for the i^{th} camera.

	f_x	f_y	C_x	C_y	x	y	z	rx	ry	rz
1.A	3033.89	3031.27	390.68	273.45	-114.07	2.66	3081.55	-179.853	-2.145	0.104
1.B	3038.79	3034.74	384.00	286.00	-113.84	2.63	3077.41	179.911	-2.017	0.105
Δ	4.90	3.47	6.68	12.55	0.23	0.03	4.14	359.764	0.128	0.001
2.A	3044.80	3042.26	381.28	288.73	21.43	2.43	3084.89	179.726	0.393	-0.026
2.B	3049.87	3045.96	384.00	286.00	21.07	2.52	3080.82	179.776	0.335	-0.025
Δ	5.07	3.70	2.72	2.73	0.36	0.09	4.07	0.050	0.058	0.001
3.A	3056.34	3054.08	395.87	284.14	3.28	3.07	3091.20	179.802	-0.115	0.201
3.B	3055.91	3051.77	384.00	286.00	3.23	3.06	3080.96	179.767	0.107	0.202
Δ	0.43	2.31	11.87	1.86	0.05	0.01	10.24	0.035	0.222	0.001
4.A	3016.04	3013.47	392.09	289.72	44.92	2.72	3085.04	179.608	1.090	0.247
4.B	3021.52	3017.23	384.00	286.00	45.14	2.66	3081.30	179.680	1.249	0.247
Δ	5.48	3.76	8.09	3.72	0.22	0.06	3.74	0.072	0.159	0.000

Table 3.6: calibration result for a distortion-free model

From table 3.6, we have to say that the differences are bigger than expected. Since the lens distortion is more sensitive to the calibration than the image origin, the results marked as **i.B** are considered to be better than those marked as **i.A**. This conclusion is roughly demonstrated by the calibration deviations, which are shown in figure 3.6 and figure 3.7 respectively.

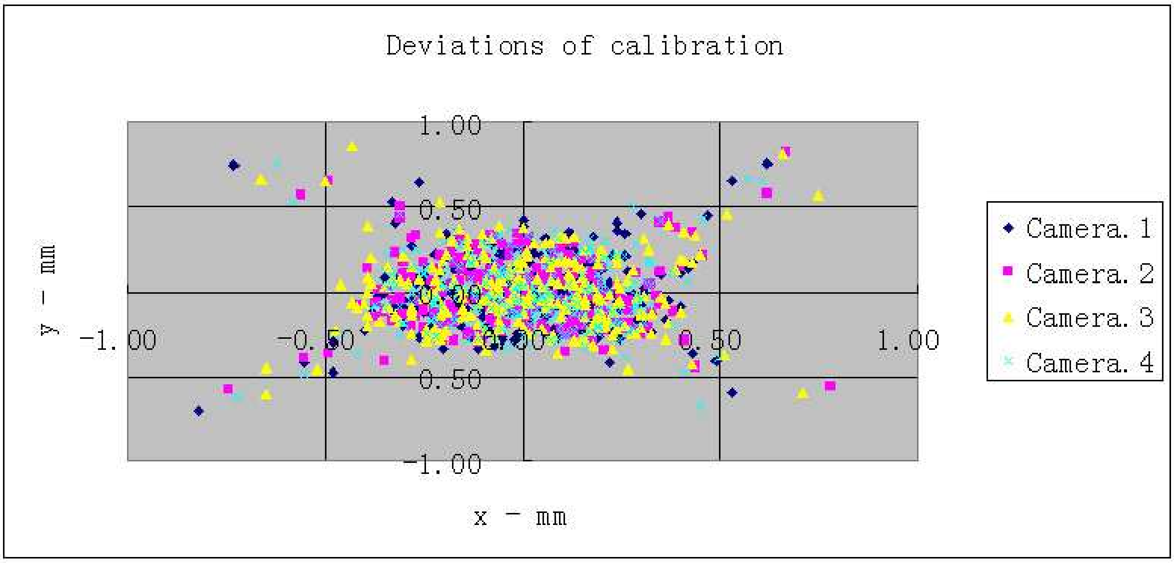


Figure 3.7: calibration deviations from a distortion-free model

3.4 Calibration with vanishing points

Similar to the last chapter, if the distortion of the camera lens is neglected or a distortion alignment procedure has been applied, there are some calibration methods in projection space for camera internal parameters including the image origin. In this section, an easy calibration method with vanishing points is discussed.

3.4.1 Projective ray and vanishing point

In the camera projection procedure, an image point $\vec{P}$ back projects to a ray defined by $\vec{P}$ and the optic center. Since the projective properties depend only on the camera projective matrix, it tells us the fact that the projection matrix relates the image point to the ray's direction.

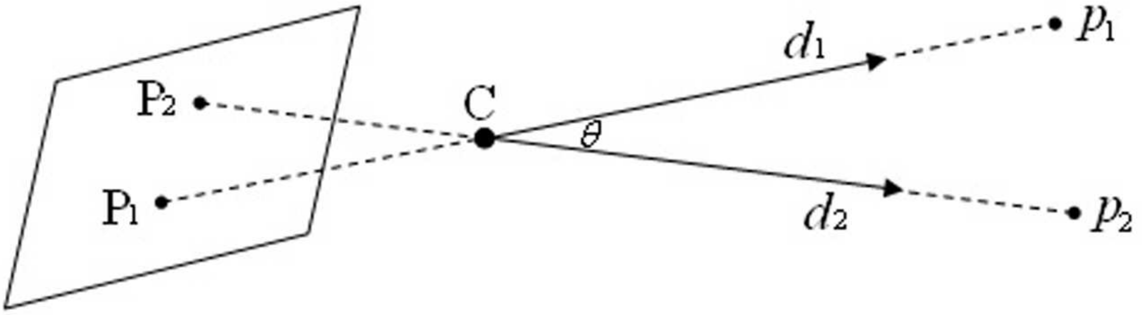


Figure 3.8: angle between projective rays

In figure 3.8, two arbitrary object points $\vec{p}_1, \vec{p}_2$ project into image point $\vec{P}_1, \vec{P}_2$ respectively. The angle θ between the two projective rays d_1, d_2 is given by

$$\cos\theta = \frac{d_1^T d_2}{\sqrt{d_1^T d_1}\sqrt{d_2^T d_2}} = \frac{\vec{p}_1^T \vec{p}_2}{\sqrt{\vec{p}_1^T \vec{p}_1}\sqrt{\vec{p}_2^T \vec{p}_2}} \tag{3.74}$$

Since $\vec{P}_1, \vec{P}_2$ are the corresponding image points from the object points $\vec{p}_1, \vec{p}_2$, both $\vec{p}_1, \vec{P}_1$ and $\vec{p}_2, \vec{P}_2$ must satisfy equation 2.11 in the camera frame,

$$\begin{aligned}\cos\theta &= \frac{(\delta_1 A^{-1}\vec{P}_1)^T(\delta_2 A^{-1}\vec{P}_2)}{\sqrt{(\delta_1 A^{-1}\vec{P}_1)^T(\delta_1 A^{-1}\vec{P}_1)}\sqrt{(\delta_2 A^{-1}\vec{P}_2)^T(\delta_2 A^{-1}\vec{P}_2)}} \\ &= \frac{\vec{P}_1^T A^{-T}A^{-1}\vec{P}_2}{\sqrt{\vec{P}_1^T A^{-T}A^{-1}\vec{P}_1}\sqrt{\vec{P}_2^T A^{-T}A^{-1}\vec{P}_2}} = \frac{\vec{P}_1^T W\vec{P}_2}{\sqrt{\vec{P}_1^T W\vec{P}_1}\sqrt{\vec{P}_2^T W\vec{P}_2}} \end{aligned} \tag{3.75}$$

where $W = A^{-T}A^{-1}$, which describes in fact the image of the absolute conic Ω_∞ after the camera projection, more details seen from [64] and [52].
The above equation supplies a constraint on the projective matrix, which is a surprising thing for camera calibration. However, the angle θ is in the camera frame and depends on the relative pose of the camera to the object. What we have in practice is usually the geometry of the real object, which is different from the projective rays.

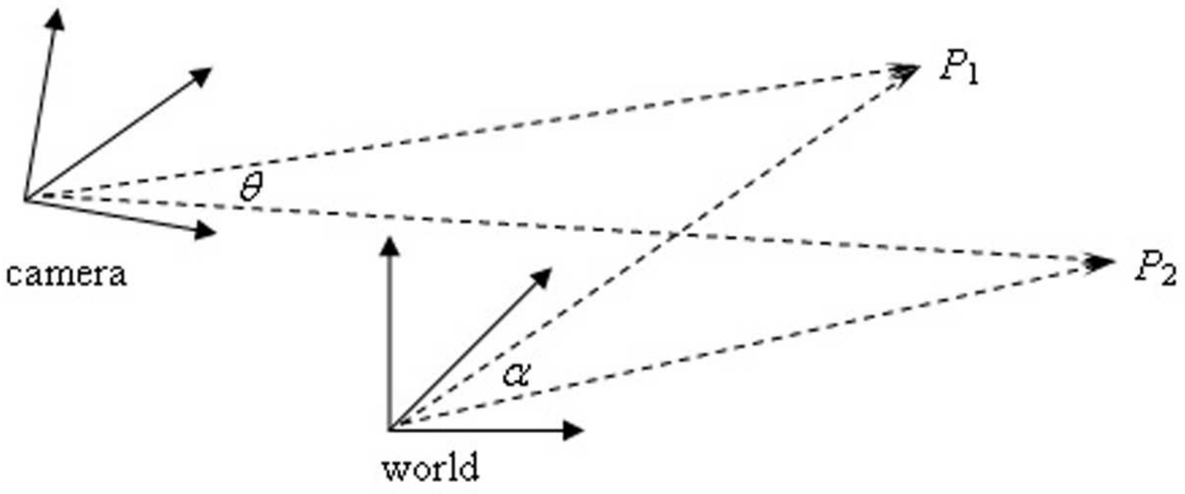


Figure 3.9: projective angles in different frames

In figure 3.9, p_1 and p_2 are two object points, α is the angle between $\vec{p_1}$ and $\vec{p_2}$ in world frame, then

$$\cos\alpha = \frac{\vec{p_1}^T\vec{p_2}}{\sqrt{\vec{p_1}^T\vec{p_1}}\sqrt{\vec{p_2}^T\vec{p_2}}} \tag{3.76}$$

If θ denotes the angle between the projective rays determined by $\vec{p_1}$ and $\vec{p_2}$ in camera frame and $T = [R, \vec{t}]$ the transformation from the camera frame to the world frame, one has

$$\cos\theta = \frac{(R\vec{p_1}+\vec{t})^T(R\vec{p_2}+\vec{t})}{\sqrt{(R\vec{p_1}+\vec{t})^T(R\vec{p_1}+\vec{t})}\sqrt{(R\vec{p_2}+\vec{t})^T(R\vec{p_2}+\vec{t})}} \tag{3.77}$$

In the above equation, the rotation R is an orthogonal matrix and the only *trouble* is $\vec{t}$. In practice, the world frame is usually defined with difference to the camera frame, which makes $\vec{t} \neq 0$. If the euclidean space extents to projective space and $\vec{p_i}$ denotes the homogeneous coordinate, the above equation can be rewritten into

$$\cos\theta = \frac{(T\vec{p_1})^T(T\vec{p_2})}{\sqrt{(T\vec{p_1})^T(T\vec{p_1})}\sqrt{(T\vec{p_2})^T(T\vec{p_2})}} = \frac{\vec{p_1}^T T^T T\vec{p_2}^T}{\sqrt{\vec{p_1}^T T^T T\vec{p_1}}\sqrt{\vec{p_2}^T T^T T\vec{p_2}}} \tag{3.78}$$

In order to eliminate $\vec{t}$, consequently T, $\vec{p_i}$ is forced to be with a form as $(x_i, y_i, z_i, 0)^T$ and yields

$$T\begin{pmatrix}\vec{p_i}\\0\end{pmatrix} = (R, \vec{t})\begin{pmatrix}\vec{p_i}\\0\end{pmatrix} = \begin{pmatrix}R\vec{p_i}\\0\end{pmatrix} \quad i = 1, 2 \tag{3.79}$$

Noticing that the pure rotation matrix R is an orthogonal one and satisfies $R^TR = R^{-1}R = I$, then

$$\cos\theta = \frac{\vec{p_1}^T\vec{p_2}^T}{\sqrt{\vec{p_1}^T\vec{p_1}}\sqrt{\vec{p_2}^T\vec{p_2}}} = \cos\alpha \tag{3.80}$$

$$0 < \theta, \alpha < 180 \quad \Longrightarrow \quad \theta = \alpha \tag{3.81}$$

An object point $\vec{p_i}$ with form of $(x_i, y_i, z_i, 0)^T$ means that it is a point at infinity. That is to say, the angle between points at infinity in both the world frame and the camera frame are the same.

Remark: in figure 3.9, one can understand as follows: when $\vec{p_i}$ is from the camera far away enough, till to infinity, the finite translation between the camera frame and the world frame can be ignored, or speaking strictly in mathematics, the projective ray is trending unlimitedly to be parallel to the direction of $\vec{p_i}$ in the world frame.

One of the distinguishing features of perspective projection is that the image of an object stretching off to infinity can have finite extent. Normally, the image projected from a point at infinity is called *vanishing point*. If the vanishing points from the object points at infinity $\vec{p_1}$ and $\vec{p_2}$ are denoted as $\vec{V_1}$ and $\vec{V_2}$ respectively, then

$$\cos\alpha = \cos\theta = \frac{\vec{V_1}^T A^{-T} A^{-1} \vec{V_2}}{\sqrt{\vec{V_1}^T A^{-T} A^{-1} \vec{V_1}}\sqrt{\vec{V_2}^T A^{-T} A^{-1} \vec{V_2}}} \tag{3.82}$$

Specially, if the two directions defined by $\vec{p_1}$ and $\vec{p_2}$ in world space are perpendicular,

$$\vec{V_1}^T A^{-T} A^{-1} \vec{V_2} = 0 \tag{3.83}$$

The equation is the basic constraint on camera projective matrix for the following calibration technique.

3.4.2 Calibration object

To complete the camera calibration, such constraints as 3.83 are needed, consequently some pairs of perpendicular directions and their vanishing points are necessary. In practice, the simplest calibration object must be a rectangle model as shown in figure 3.10

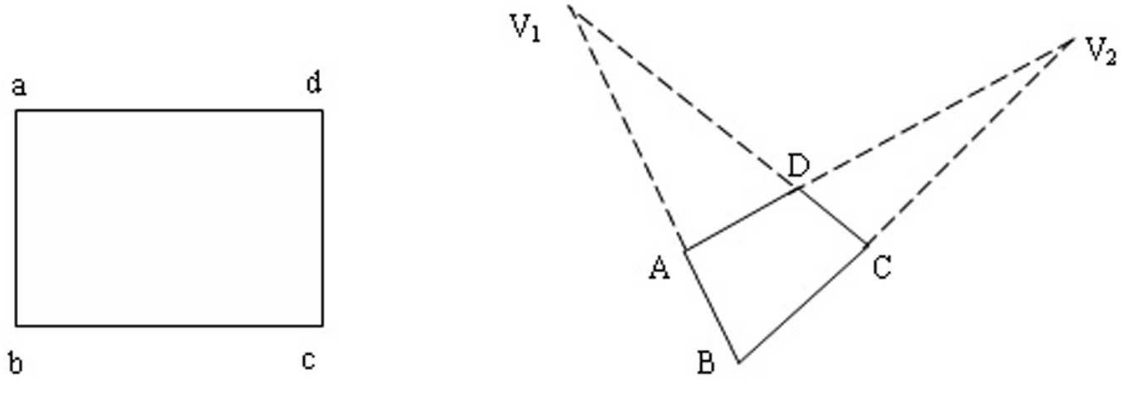


Figure 3.10: a rectangle model and its image

Since the parallel directions in real world are imaged into the same vanishing point, the intersections of lines AB, CD and AD, BC in image plane must be the responding vanishing points, which are denoted as $\vec{V_1}, \vec{V_2}$, namely

$$\vec{V_1} = l_{AB} \times l_{CD} \quad \vec{V_2} = l_{AD} \times l_{BC}$$

Considering the fact that the neighbor directions of a rectangle are perpendicular, the vanishing points $\vec{V}_1, \vec{V}_2$ satisfy

$$\vec{V}_1^T A^{-T} A^{-1} \vec{V}_2^T = 0 \tag{3.84}$$

Since the above constraint is independent to the camera pose, the calibration for camera projective matrix will be solved by moving the model or the camera into more than 4 different poses, and the movements need not to be known.

More pairs of perpendicular directions in the model plane?

From the above description, it is easy to arise such a question: why not set in the model plane more such pairs of perpendicular directions for giving more pairs of vanishing points instead of moving the camera or the model into different poses. The reason is that **all the pairs of vanishing points from the model plane will give the same constraint on the projective matrix**. The proof is as follows.
Denote $\vec{v}_1, \vec{v}_2$ two points at infinity, which are determined from two perpendicular directions in the model plane, and $\vec{V}_1, \vec{V}_2$ the responding vanishing points, then

$$\vec{v}_1^T \vec{v}_2 = \vec{v}_2^T \vec{v}_1 = 0 \quad \vec{V}_1^T A^{-T} A^{-1} \vec{V}_2^T = 0$$

Since the model plane intersects the plane π_∞ at infinity into a line l_∞, any point on the line at infinity must be a linear combination of $\vec{v}_1, \vec{v}_2$,

$$\vec{v}_3 = a\vec{v}_1 + b\vec{v}_2 \quad \vec{v}_4 = c\vec{v}_1 + d\vec{v}_2$$

If $\vec{v}_3$ and $\vec{v}_4$ are perpendicular to each other, one has

$$\begin{aligned} \vec{v}_3^T \vec{v}_4 &= (a\vec{v}_1 + b\vec{v}_2)^T (c\vec{v}_1 + d\vec{v}_2) \\ &= ac\vec{v}_1^T \vec{v}_1 + bd\vec{v}_2^T \vec{v}_2 = 0 \end{aligned} \tag{3.85}$$

Similarly denote the corresponding vanishing points as $\vec{V}_3, \vec{V}_4$, then

$$\begin{aligned} \vec{V}_3^T W \vec{V}_4 &= (\delta_3 A \vec{v}_3)^T A^{-T} A^{-1} (\delta_4 A \vec{v}_4)^T \\ &= \delta_3\delta_4(ac\vec{v}_1^T \vec{v}_1 + bd\vec{v}_2^T \vec{v}_2) + \delta_3\delta_4(ad\vec{V}_1^T W \vec{V}_2 + bc\vec{V}_2^T W \vec{V}_1) \\ &= \delta_3\delta_4(ad + bc)\vec{V}_1^T W \vec{V}_2 \end{aligned} \tag{3.86}$$

where $W = A^{-T}A^{-1}$. This means that it rises no other constraint on W, consequently on the projective matrix A, by forcing $\vec{V}_3^T W \vec{V}_4 = 0$.

3.4.3 Solving camera calibration

For a zero-distortion camera model, the projective matrix and its inverse are as follows

$$A = \begin{pmatrix} f_x & 0 & C_x \\ 0 & f_y & C_y \\ 0 & 0 & 1 \end{pmatrix} \Longrightarrow A^{-1} = \begin{pmatrix} \frac{1}{f_x} & 0 & -\frac{C_x}{f_x} \\ 0 & \frac{1}{f_y} & -\frac{C_y}{f_y} \\ 0 & 0 & 1 \end{pmatrix} \tag{3.87}$$

the image of the absolute conic $W = A^{-T} \cdot A^{-1}$ must be symmetric and is denoted as follows

$$W = \begin{pmatrix} \frac{1}{f_x^2} & 0 & -\frac{C_x}{f_x^2} \\ 0 & \frac{1}{f_y^2} & -\frac{C_y}{f_y^2} \\ -\frac{C_x}{f_x^2} & -\frac{C_y}{f_y^2} & 1 + \frac{C_x^2}{f_x^2} + \frac{C_y^2}{f_y^2} \end{pmatrix} = \begin{pmatrix} w_1 & 0 & w_3 \\ 0 & w_2 & w_4 \\ w_3 & w_4 & w_5 \end{pmatrix} \tag{3.88}$$

Noting that $w_5 = 1 + \frac{C_x^2}{f_x^2} + \frac{C_y^2}{f_y^2} \geq 1$, a 4-vector is defined

$$\vec{x} = (x_1, x_2, x_3, x_4)^T = (\frac{w_1}{w_5}, \frac{w_2}{w_5}, \frac{w_3}{w_5}, \frac{w_4}{w_5})^T \tag{3.89}$$

For every pair of vanishing points $\vec{V_i}$ and $\vec{V_j}$, we have

$$\vec{V_i}^T W \vec{V_j} = 0 \tag{3.90}$$

Extracting the above equation with the coordinates of the vanishing points by $\vec{V_n} = (X_n, Y_n, 1)^T, n = i, j$, one has

$$(X_i X_j, Y_i Y_j, X_i + X_j, Y_i + Y_j)^T \cdot \vec{x} = -1 \tag{3.91}$$

For every image of the model plane, there is an useful pair of vanishing points, which yield a constraint equation like above. If n different poses are observed and stack the n equations, one has

$$V\vec{x} = -1 \tag{3.92}$$

where V is $n \times 4$ matrix and the right side is a 4-vector with all elements -1. If $n \geq 4$, we can solve an unique solution for $\vec{x}$, consequently W including w_5, finally f_x, f_y, C_x and C_y as follows

$$\begin{aligned} C_x &= -x_3/x_1 \\ C_y &= -x_4/x_2 \\ w_5 &= 1 - C_x \cdot x_3 - C_y \cdot x_4 \\ f_x &= 1/\sqrt{w_5 x_1} \\ f_y &= 1/\sqrt{w_5 x_2} \end{aligned}$$

If the poses are too few for all parameters, the projective matrix can also partially estimated:
If n = 3, $f_x = f_y = f$ and C_x, C_y can be solved.
If n = 2, f_x, f_y or C_x, C_y can be solved with the others known;
If n = 1, one parameter can be solved, e.g. $f_x = f_y = f$ with C_x, C_y known.

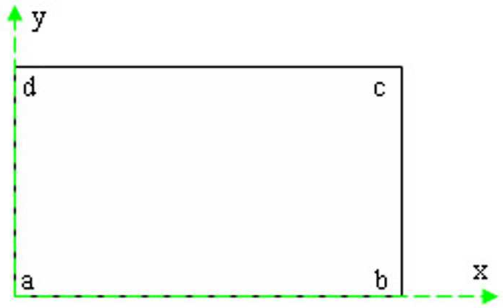


Figure 3.11: the model frame

Estimation of camera pose

Once the projective matrix A is obtained, the camera pose can be determined as follows: as shown in figure 3.11, the world frame is defined in the model plane and the size of the rectangle is given by $ab = cd = l > 0$ and $ad = bc = kl$, the coordinates of the points a, b, c, and d have the following values in the world frame

$$\begin{aligned}\vec{a} &= (0,0,0)^T \quad \vec{c} = (l,kl,0)^T \\ \vec{b} &= (l,0,0)^T \quad \vec{d} = (0,kl,0)^T\end{aligned}$$

As denoted before, the transformation from the camera frame to the reference frame is denoted as $[R|\vec{t}]$ and the four points are projected into A, B, C and D respectively. With the camera projective equation 2.13, the coordinates of the origin $\vec{a}$ of the world frame will define the translation

$$\vec{t} = [R|\vec{t}]\vec{a} = \delta_a \cdot A^{-1}\vec{A} \tag{3.93}$$

Since δ_a is unknown, the translation cannot be determined. Let's look into the other points, we have

$$\begin{aligned}\delta_b \cdot A^{-1}\vec{B} &= [R|\vec{t}]\vec{b} = l\vec{r}_1 + \vec{t} &(3.94)\\ \delta_d \cdot A^{-1}\vec{D} &= [R|\vec{t}]\vec{d} = kl\vec{r}_2 + \vec{t} &(3.95)\\ \delta_c \cdot A^{-1}\vec{C} &= [R|\vec{t}]\vec{c} = l\vec{r}_1 + kl\vec{r}_2 + \vec{t} &(3.96)\end{aligned}$$

where $\vec{r}_i$ is the i^{th} column of the rotation matrix R. From the above three equations, the first columns of the rotation matrix can be solved

$$\begin{aligned}\vec{r}_1 &= \lambda_c A^{-1}\vec{C} - \lambda_d A^{-1}\vec{D} &(3.97)\\ k\vec{r}_2 &= \lambda_c A^{-1}\vec{C} - \lambda_b A^{-1}\vec{B} &(3.98)\end{aligned}$$

where $\lambda_b = \delta_b/l, \lambda_c = \delta_c/l$ and $\lambda_d = \delta_d/l$. Thinking about the properties of a rotation matrix, one has

$$\begin{aligned}\vec{r}_1^T\vec{r}_2 = 0 &\quad (3.99)\\ |\vec{r}_1| = |\vec{r}_2| = 1 &\quad (3.100)\end{aligned}$$

If the length ration between the neighbor sides k is given, the problem becomes three equations for three unknowns $\lambda_b, \lambda_c, \lambda_d$. Since all unknowns are positive values, a unique solution for $\lambda_b, \lambda_c, \lambda_d$ will be obtained, consequently r_1, r_2, and then

$$\vec{r}_3 = \vec{r}_1 \times \vec{r}_2 \tag{3.101}$$

Once the rotation matrix is determined, the translation $\vec{t}$ is solved up to the scale factor l, namely

$$\vec{t} = (\lambda_b \cdot A^{-1}\vec{B} - \vec{r}_1) \cdot l \tag{3.102}$$

This solution for camera pose is also reasonable in geometry: the rotation is the relative pose, which is determined by the ratio of the lengths from both directions, but completely independent to the absolute size of the rectangle; the translation is up to the model size: with the same image size, a larger model object must be standing further away from the camera.

3.4.4 Degenerate configurations

A degenerate configuration is a situation that the camera and the model stand in such relative poses that the image of the model provides no useful constraint on the projective matrix.

Situation 1: *If the model plane at the second pose is parallel to the first pose, the second image will not provide any additional constraint.*

Denote π_i the plane at the i^{th} pose and a_i, b_i, c_i, d_i are the object points from the model,

$$\pi_1 \parallel \pi_2 \quad a_i, b_i, c_i, d_i \in \pi_i, \quad i = 1, 2$$

If the model $\{a_2, b_2, c_2, d_2\}$ from plane π_2 projects into the model $\{a_1^{'}, b_1^{'}, c_1^{'}, d_1^{'}\}$ in plane π_1, then

$$\overrightarrow{a_1^{'}b_1^{'}} \parallel \overrightarrow{a_2b_2} \quad b_1^{'}c_1^{'} \parallel b_2c_2 \quad c_1^{'}d_1^{'} \parallel c_2d_2 \quad d_1^{'}a_1^{'} \parallel d_2a_2$$

Since all parallel object lines will intersect to each other at the same point at infinity, the responding vanishing points $V_i^{'}$ from model $\{A_1^{'}B_1^{'}C_1^{'}D_1^{'}\}$ and V_i from $\{A_2B_2C_2D_2\}$ must be the same. At the same time, both $\{a_1b_1c_1d_1\}$ and $\{a_1^{'}b_1^{'}c_1^{'}d_1^{'}\}$ locate in the plane π_1, which guarantees that their vanishing points result in the same constraint. Consequently, the model $\{a_1b_1c_1d_1\}$ and $\{a_2b_2c_2d_2\}$ give the same constraint on the camera projective matrix.

Situation 2: *If the model plane is parallel to the image plane, or any side of the rectangle model is parallel to any coordinate axis of the image frame, there is no helpful constraint with this pose.*

With the camera projective procedure, a vanishing point V_i from an object point p_i at infinity is the intersection between the image plane and the projective ray passing through the optical center and parallel to the direction determined by p_i. When the model plane is parallel to the image plane, the vanishing points from all directions in the model plane are image points at infinity, which can not be detected in the image. For the same reason, when a side $l_{ab/cd}$ (or $l_{bc/ad}$) is parallel to a coordinate axis of the image frame, the corresponding lines in the image plane will satisfy

$$l_{AB} \parallel l_{CD} \Longrightarrow V_1 = l_{AB} \times l_{CD} = (X_i, Y_i, 0, 0)^T$$

A vanishing point at infinity will fail to satisfy the equation 3.83.

A square model: the improved solution

Although there is no solution for the problem that the model plane is parallel to the image plane, the left parallel problem can be avoided by introducing a square model instead of the rectangle model in practice. As shown in figure 3.12, a square model has two perpendicular diagonals. If $\vec{V}_1, \vec{V}_2, \vec{V}_3, \vec{V}_4$ denote the vanishing points imaged respectively from the directions $\vec{ac}, \vec{bd}, \vec{ab}(\vec{cd}), \vec{bc}(\vec{ad})$, one has

$$\vec{V}_1^T A^{-T} A^{-1} \vec{V}_2^T = 0 \quad \vec{V}_3^T A^{-T} A^{-1} \vec{V}_4^T = 0 \tag{3.103}$$

As proved before, the above 2 equations supply the same constraint on projective matrix A. However, when one pair of them does not work because of the parallel problem, the other pair maybe work well.

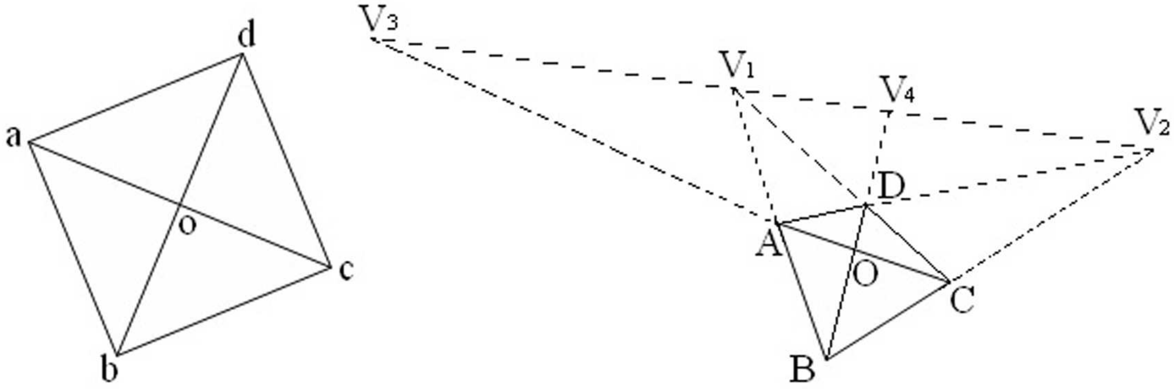


Figure 3.12: a square model and its image

Vanishing points from the diagonals

As shown in figure 3.12, $\vec{V_1}, \vec{V_2}$ denote the vanishing points from neighbor sides and $\vec{V_3}, \vec{V_4}$ the vanishing points from the diagonals. $\vec{V_1}, \vec{V_2}$ can be easily determined as described before, let us focus on $\vec{V_3}, \vec{V_4}$.
Denote v_3 as the intersection of line l_{ac} and the plane at infinity π_∞, v_3 is of course a point at infinity. Since o is the midpoint of the stretch ac and v_3 is a point at infinity, the four collinear points a, c, o, v_3 actually consist a harmonic points series, that is, the cross-ratio of them is a constant -1, namely

$$(a, c; o, v_3) = \frac{(a, c, o)}{(a, c, v_3)} = \frac{\overrightarrow{ao}}{\overrightarrow{oc}} \cdot \frac{\overrightarrow{v_3c}}{\overrightarrow{av_3}} = -1 \tag{3.104}$$

Suppose the harmonic points series project into A, C, O, V_3 respectively. Since collinearity and cross-ratio are invariant under camera projection, A, C, O, V_3 must be also a harmonic points series,

$$(\vec{A} \times \vec{C})^T \vec{V}_3 = 0 \tag{3.105}$$

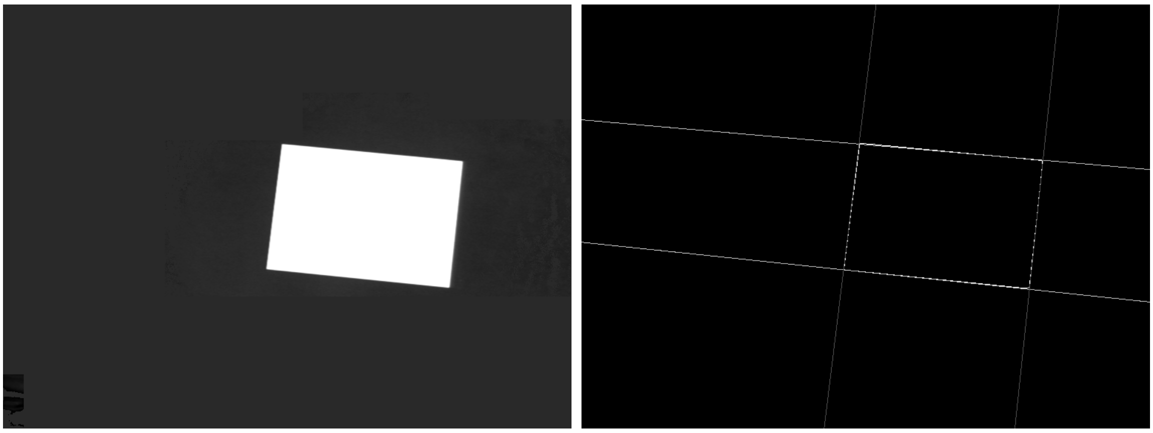
$$|\overrightarrow{AO}||\overrightarrow{V_3C}| + |\overrightarrow{OC}||\overrightarrow{AV_3}| = 0 \tag{3.106}$$

From above two equations, vanishing point $\vec{V_3}$ can be easily solved. In the same way, vanishing point $\vec{V_4}$ is obtained from B, D, and O.

Remark: In computing the vanishing point $\vec{V_3}(\vec{V_4})$, A, C and O are assumed to be collinear. However, they do not in practice hold this property due to noise and lens distortion. It is necessary to apply a maximum likelihood algorithm on them to get a best-fit line before solving $\vec{V_3}(\vec{V_4})$.

3.4.5 Experimental results

In experiments, a board with a white paper sticked to moves before the camera and 40 images are taken to calibrate a JAI CV-M50 camera with a lens of 8mm focal length.



(a) original image (b) vanishing points

Figure 3.13: procedure for searching the vanishing points

As seen in figure 3.13, a *hough transform* procedure is used for searching the vanishing points. Since a vanishing point is the cross from the opposite sides of the rectangle, it may have very large values in pixel coordinates because of noises. When all the 40 pairs of vanishing points are used, no reasonable result is obtained. Therefore, we have removed the pairs with too large coordinates and get 25 pairs listed in table 3.7.

	$V_i.X$	$V_i.Y$	$V_j.X$	$V_j.Y$		$V_i.X$	$V_i.Y$	$V_j.X$	$V_j.Y$
1	-835.42	1750.42	3855.88	2921.67	14	-2664.77	1575.62	1514.99	2802.02
2	-627.93	7084.14	-714.93	70.30	15	-3316.10	1612.66	1554.26	3159.30
3	-3746.61	310.44	485.93	2185.79	16	-2489.39	1669.08	2397.55	4325.27
4	-3602.38	1071.84	818.50	1862.84	17	-2931.65	2557.26	2017.78	2453.17
5	-2795.84	1534.05	926.14	1432.93	18	-3666.83	2847.24	1135.89	1426.38
6	-2779.55	-717.96	144.00	1329.41	19	-8773.22	8574.97	1027.73	976.67
7	-2913.01	90.67	371.34	1496.89	20	-3538.87	4639.55	1142.09	881.34
8	-3931.35	1920.56	893.66	1494.79	21	-702.16	133.64	1175.54	800.68
9	-5821.80	3467.00	1001.70	1532.92	22	-4107.74	3473.94	897.86	913.13
10	-2148.04	2581.75	1467.18	1308.23	23	-4186.94	4957.69	1047.40	973.62
11	-2724.90	3628.59	1479.61	1198.20	24	-1945.08	3924.50	1218.64	765.49
12	743.86	2504.87	-6912.47	1408.75	25	-2014.98	6407.42	1262.19	609.53
13	-3323.37	1239.05	1075.36	2576.89					

Table 3.7: vanishing points for the calibration

With the vanishing points pairs in above table, we have obtained the following values for the image origin and focal length as shown in table 3.8. Although this technique may be the new direction of camera calibration and some researchers, such as zhang in [48], has made some improvements in this field, we see no possibilities in the near future for this method to be improved so greatly that it may be applied to industrial applications.

	C_x	C_y	f_x	f_y	→	f	S_x	S_y
vanishing points 1-10	314.57	325.67	831.88	851.36	→	6.93	0.008135	0.008325
vanishing points 1-11	312.70	324.60	837.20	854.58	→	6.97	0.008156	0.008325
vanishing points 1-12	325.32	323.18	819.99	834.48	→	6.83	0.008180	0.008325
vanishing points 1-13	324.33	325.49	830.11	843.97	→	6.91	0.008188	0.008325
vanishing points 1-14	324.96	326.96	831.42	846.72	→	6.92	0.008175	0.008325
vanishing points 1-15	318.17	337.28	865.49	875.93	→	7.21	0.008226	0.008325
vanishing points 1-16	318.64	339.29	869.74	881.53	→	7.24	0.008214	0.008325
vanishing points 1-17	312.97	347.83	892.82	900.46	→	7.43	0.008254	0.008325
vanishing points 1-18	309.70	351.67	902.78	911.28	→	7.52	0.008247	0.008325
vanishing points 1-19	341.87	313.98	788.58	793.70	→	6.56	0.008271	0.008325
vanishing points 1-20	341.34	312.29	790.86	796.49	→	6.58	0.008266	0.008325
vanishing points 1-21	332.20	335.02	871.95	879.02	→	7.26	0.008258	0.008325
vanishing points 1-22	332.27	335.01	872.08	879.17	→	7.26	0.008258	0.008325
vanishing points 1-23	328.85	353.97	908.20	914.36	→	7.56	0.008269	0.008325
vanishing points 1-24	332.07	360.60	908.81	913.04	→	7.57	0.008286	0.008325
vanishing points 1-25	333.68	365.87	912.66	915.48	→	7.60	0.008299	0.008325
average	325.23	336.17	858.41	868.22	→	7.15	0.008230	0.008325

Table 3.8: calibration results from the calibration

3.5 Search of the image origin

From the calibration methods described in foregoing sections, we notice that none of them estimates both the image origin and the radial distortion as well as the other camera parameters. Since the radial distortion is very sensitive to measurements, a direct searching method for the image origin, known as the **polytope** method or the **simplex** method, is introduced in this section as a complementarity to the calibrations.

3.5.1 Response function

To estimate the image origin, an initial guess is needed. With the initial guess, the other camera parameters will be obtained from the calibration methods described in last sections for distortion camera model. Using the solved camera parameters, the best values for the image origin will be searched. The *best* means the searched values for the image origin are better than any other values by describing the camera projection with the other camera parameters together. Therefore, the following two residue functions are defined

$$R_x = \frac{X - C_x}{1 + k\left(\left(\frac{X-C_x}{f_x}\right)^2 + \left(\frac{Y-C_y}{f_y}\right)^2\right)} - \frac{f_x \cdot x_c}{z_c} \tag{3.107}$$

$$R_y = \frac{Y - C_y}{1 + k\left(\left(\frac{X-C_x}{f_x}\right)^2 + \left(\frac{Y-C_y}{f_y}\right)^2\right)} - \frac{f_y \cdot y_c}{z_c} \tag{3.108}$$

In an ideal situation, $R_x = R_y = 0$. But it in the actual situations will never happen, that is why they are called residues. The following response function F for all n calibration points is defined

$$F(\vec{P}) = \sum_{i=1}^{n} \left(R_{xi}^2 + R_{yi}^2\right) \tag{3.109}$$

where

$$\vec{P} \equiv (C_x, C_y)^T \tag{3.110}$$

The task now is to search a best pixel $\vec{P} = (C_x, C_y)^T$ to minimize the response function.

3.5.2 Searching algorithm

The searching algorithm is based on the theory that a pixel closer to the best pixel has a smaller response, which is supported by the continuity of the response function.
The algorithm starts with three appropriate non-collinear pixels $\vec{P}_b, \vec{P}_m, \vec{P}_w$, which are so ordered that $F(\vec{P}_b) < F(\vec{P}_m) < F(\vec{P}_w)$, as shown in figure 3.14. The three points composing of the so-called **polytope** are named respectively as the *best*, the *middle* and the *worst* pixel. In an iteration, a new polytope is generated by introducing a new pixel to replace the *worst* pixel till the polytope converge to a vanishing point. The procedure in detail consists of following steps:

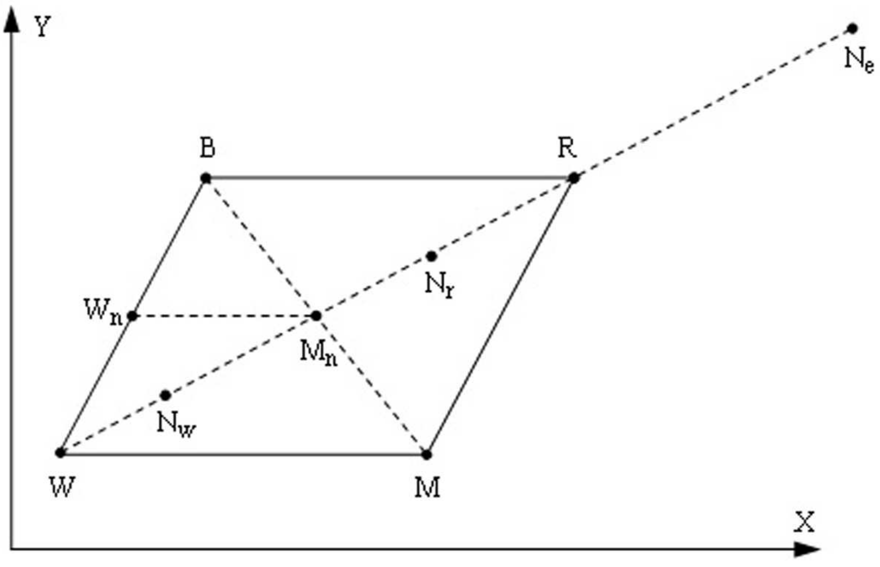


Figure 3.14: image origin searching by polytope method

1. Polytope initialization and controlling
   Order and name them into $\vec{P}_b \leq \vec{P}_m \leq \vec{P}_w$, and check if the polytope is becoming a vanishing point, which is determined by

$$\left(F(\vec{P}_w) - F(\vec{P}_b)\right)^2 < \varepsilon \tag{3.111}$$

   where ε is a sufficiently small value. If the required accuracy is reached, the searching procedure will end, otherwise go to the next step to continue searching.

2. Reflection pixel
   The *worst* pixel is optimized by moving it along a correct direction, which is found out by firstly generating the *reflection* pixel of $\vec{P}_w$,

$$\vec{P}_r = \vec{P}_b + \vec{P}_m - \vec{P}_w \tag{3.112}$$

   Its response $F(\vec{P}_r)$ is calculated and the direction is determined as follows:
   if $F(\vec{P}_r) < F(\vec{P}_b)$, go to the expansion direction in **step 3**;
   if $F(\vec{P}_b) \leq F(\vec{P}_r) < F(\vec{P}_m)$, maintain the polytope in **step 4**;
   if $F(\vec{P}_r) \geq F(\vec{P}_m)$, go to the contraction direction in **step 5**.

3. Polytope expansion
   Generate a new pixel at the extension of W-R by

$$\vec{P}_n = \vec{P}_r + \rho(\vec{P}_r - \vec{P}_w) \tag{3.113}$$

   where $\rho > 0$. Then the corresponding response is calculated and checked: if $F(\vec{P}_n) < F(\vec{P}_r)$, replace the *worst* pixel with the new pixel $\vec{P}_w = \vec{P}_n$; otherwise replace the *worst* pixel with the *reflection* pixel $\vec{P}_w = \vec{P}_r$; go back to **step 1** to continue optimizing.

4. Polytope maintain
   The polytope keeps its size, but the *worst* pixel is replaced by the *reflection* pixel $\vec{P}_w = \vec{P}_r$, and go back to **step 1**.

5. Polytope contraction
   Generate a new pixel on the portion of W-R, but the middle pixel $\vec{M}_n$ must be dropped out, because it must be also on the line determined by $\vec{P}_b$ and $\vec{P}_m$. If the *reflection* pixel is better than the *worst* pixel $F(\vec{P}_r) < F(\vec{P}_w)$, the new pixel should be closer to the *reflection* pixel, that is

$$\vec{P}_n = \vec{P}_r + \lambda(\vec{P}_w - \vec{P}_r) \tag{3.114}$$

   Otherwise the new pixel should be closer to the *worst* pixel, yields

$$\vec{P}_n = \vec{P}_w + \lambda(\vec{P}_r - \vec{P}_w) \tag{3.115}$$

   where $0 < \lambda < 0.5$. The corresponding response is calculated and checked: if $F(\vec{P}_n) < F(\vec{P}_w)$, replace the *worst* pixel with the new pixel $\vec{P}_w = \vec{P}_n$ and come back to **step 1**; otherwise there is no better pixel than $\vec{P}_w$ on the line WR, the optimizing area must be reduced by

$$\vec{P}_m = \left(\vec{P}_m + \vec{P}_b\right)/2 \tag{3.116}$$

$$\vec{P}_w = \left(\vec{P}_w + \vec{P}_b\right)/2 \tag{3.117}$$

   then go back to **step 1** to continue optimizing.

The above procedure is repeated till the polytope becomes a vanishing point, that is, the three positions are close enough to each other, and at the same time are also close enough to the best pixel.

Remark: The above optimizing algorithm is a general method in mathematics. How well it works in practice depends on how well the parameters are set. Fortunately, ρ and λ are not very sensitive in our case and $\rho \approx 0.5, \lambda \approx 0.25$ are good choices. If an initial pixel $\vec{P}_0$ for the image origin is obtained from other methods, e.g. calibration for a distortion-free model, or simply got from the image center, the three initial non-collinear positions can be set as follows

$$\vec{P}_b = \vec{P}_0 \tag{3.118}$$

$$\vec{P}_m = \vec{P}_0 + h \cdot \vec{E}_x \tag{3.119}$$

$$\vec{P}_w = \vec{P}_0 + h \cdot \vec{E}_y \tag{3.120}$$

where $\vec{E}_x = (1,0)^T, \vec{E}_y = (0,1)^T$ and h is the initial step length, which works well in image origin searching if $h \approx 5 - 10$ pixels.

3.5.3 Combination solution

The searching method for image origin consists in fact of two dependent stages: the first stage, any calibration method described in last sections, need a given image origin for calibrating the other parameters; the second stage, the polytope algorithm, need the solved parameters from the first stage for a better image origin. Given an inaccurate image origin, the inaccuracy will be inherited by dispatching into the other camera parameters, which will result in some inaccuracy to the new image origin from the polytope algorithm in the second stage. That is to say, the end result is dependent on

the initial guess of the image origin.
If the ideal pixel for the image origin is denoted as $\vec{P}_{ideal}$, the initial guess is denoted as $\vec{P}_{old}$ and the searched pixel is denoted as $\vec{P}_{new}$, it must be sure that $\vec{P}_{new}$ is closer to $\vec{P}_{ideal}$ than $\vec{P}_{old}$, that is

$$|\vec{P}_{new} - \vec{P}_{ideal}| < |\vec{P}_{old} - \vec{P}_{ideal}| \tag{3.121}$$

If $\vec{P}_{old}$ is replaced by $\vec{P}_{new}$ for calibration stage, a better pixel for the image origin can be expected from the following searching algorithm. Thus a practical combining solution is as follows:

1. calibrate the other camera parameters f_x, f_y, d and $(R, \vec{t})$ with a given image origin $\vec{P}_{old}$;

2. search a more accurate image origin $\vec{P}_{new}$ with the solved camera parameters from *step 1*;

3. replace the old image origin with the newly calculated one, $\vec{P}_{old} = \vec{P}_{new}$, then go back to *step 1* to continue searching.

The above procedure is repeated till the image origin changes no longer, which is determined by

$$\left(\vec{P}_{new} - \vec{P}_{old}\right)^2 < \varepsilon \tag{3.122}$$

where ε is a sufficiently small value.

Remark: Here an iterative procedure for calibration and searching is applied. The iteration can be integrated into the searching algorithm: every time a new pixel is created, the camera is calibrated with this pixel as the image origin, then the response is calculated with the newly solved camera parameters. When the searching algorithm is finished, the obtained pixel is the final solution for the image origin.

3.5.4 Experimental results

In this experiment, the images in figure 3.5 are applied for a non-coplanar calibration method followed by an image origin searching procedure, which brings better integrity of the calibration procedure.

From table 3.9 and 3.10, we find that the image origin has impact mainly on the camera pose, especially the rotation and almost no impact on other internal parameters. More concretely, deviations of C_x and C_y are compensated respectively by ry and rx.

Contrasting carefully the deviations from figure 3.6 and 3.15, such a conclusion can be generally made that the camera parameters estimated by the combination solution satisfy the calibration points a little better than those from non-coplanar calibration.

	f_x	f_y	C_x	C_y	K
Camera.1 Noncoplanar	3038.79	3034.74	384.00	286.00	-0.298088
Camera.1 Combination	3038.72	3034.65	383.71	279.92	-0.295396
Camera.1 Difference	0.07	0.09	0.29	6.08	0.002692
Camera.2 Noncoplanar	3049.87	3045.96	384.00	286.00	-0.311599
Camera.2 Combination	3049.80	3045.90	383.37	300.51	-0.313424
Camera.2 Difference	0.07	0.06	0.63	14.51	0.001825
Camera.3 Noncoplanar	3055.91	3051.77	384.00	286.00	-0.330677
Camera.3 Combination	3055.86	3051.71	416.11	296.46	-0.320664
Camera.3 Difference	0.05	0.06	32.11	10.46	0.010013
Camera.4 Noncoplanar	3021.52	3017.23	384.00	286.00	-0.308850
Camera.4 Combination	3021.51	3017.23	384.19	297.02	-0.307897
Camera.4 Difference	0.01	0.00	0.19	11.02	0.000953

Table 3.9: internal parameters affected by image origin

	x	y	z	rx	ry	rz
Camera.1 Noncoplanar	-113.84	2.63	3077.41	179.911	-2.017	0.105
Camera.1 Combination	-113.84	2.53	3077.41	-179.972	-2.012	0.105
Camera.1 Difference	0.00	0.10	0.00	0.117	0.005	0.000
Camera.2 Noncoplanar	21.07	2.52	3080.82	179.776	0.335	-0.025
Camera.2 Combination	21.06	2.49	3080.71	179.503	0.347	-0.025
Camera.2 Difference	0.01	0.03	0.11	0.273	0.012	0.000
Camera.3 Noncoplanar	3.23	3.06	3080.96	179.767	0.107	0.202
Camera.3 Combination	3.27	3.02	3081.21	179.571	-0.496	0.205
Camera.3 Difference	0.04	0.04	0.25	0.196	0.603	0.003
Camera.4 Noncoplanar	45.14	2.66	3081.30	179.680	1.249	0.247
Camera.4 Combination	45.15	2.63	3081.33	179.471	1.245	0.247
Camera.4 Difference	0.01	0.03	0.03	0.209	0.004	0.000

Table 3.10: external parameters affected by image origin

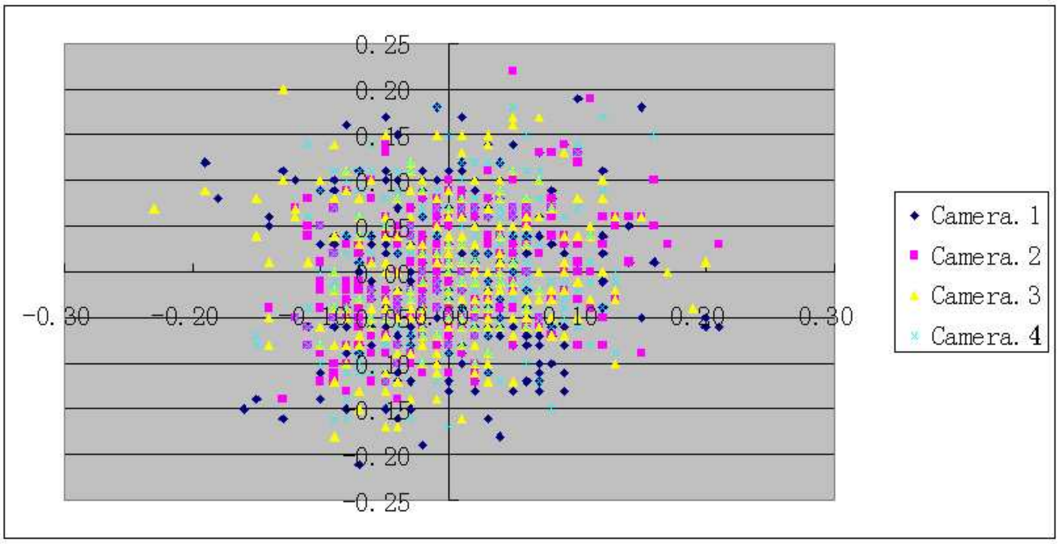


Figure 3.15: deviations from polytope searching

3.6 Refinement with nonlinear minimization

Some easy but effective calibration methods have been proposed in last sections. Since no iteration is required, the algorithms seem clear and easy and the procedures are implemented quickly. However, the actual constraints are applied mostly to restore the final camera parameters from the intermediate parameters, not directly to the intermediate parameters, which causes that the intermediate solutions satisfy the constraints not so well because of noises. Consequently, the accuracy of the final solution is not optimum for applying into measurement. In this section, an iterative algorithm with the objective of minimizing the the residual errors of nonlinear equations is applied.

3.6.1 Nonlinear minimization

In order to estimate all camera parameters accurately and simultaneously by an iterative algorithm, the following two functional relations from 2.19 and 2.20 for distortion model are introduced and to be zeroed

$$\begin{aligned} F_x(R, \vec{t}, f_x, f_y, C_x, C_y, K) &= z_c(X - C_x) - x_c f_x(1 + Kr) = 0 && (3.123) \\ F_y(R, \vec{t}, f_x, f_y, C_x, C_y, K) &= z_c(Y - C_y) - y_c f_y(1 + Kr) = 0 && (3.124) \end{aligned}$$

where

$$\begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = R \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \vec{t} \tag{3.125}$$

$$r = \left(\frac{X - C_x}{f_x}\right)^2 + \left(\frac{Y - C_y}{f_y}\right)^2 \tag{3.126}$$

where $(R, \vec{t})$ is the matrix form from $\alpha, \beta, \gamma, t_x, t_y, t_z$, which describe the transformation from the camera frame to the world frame. Ideally, F_x and F_y are both zero. Due to uncertainties or errors in the system modeling and patterns recognizing, they are in general nonzero. By stacking up such relations for all calibration points, a system of nonlinear equations is obtained. The solution from the system for both the internal and external parameters listed in F_x and F_y must have minimized the least squares error

$$F(R, \vec{t}, f_x, f_y, C_x, C_y, K) = \sum_{i=1}^{n} \left({F_{x,i}}^2 + {F_{y,i}}^2\right) \Longrightarrow min \tag{3.127}$$

To solve such systems of nonlinear equations, an iterative algorithm is usually applied as described with details in appendix.

3.6.2 Initial guess

To solve an over-determined system of nonlinear equations, a good set of initial guess values for f_x, f_y, C_x, C_y, K and $R, \vec{t}$ are necessary and there are two possibilities to get such a set of initial guess

1. The combination solution referred in section 3.5;
2. The calibration for distortion-free model with $K = 0$.

In fact, these two methods are complete procedure for camera calibration, the result of which can not be far away from the ideal values and must satisfy the following iteration algorithm well.

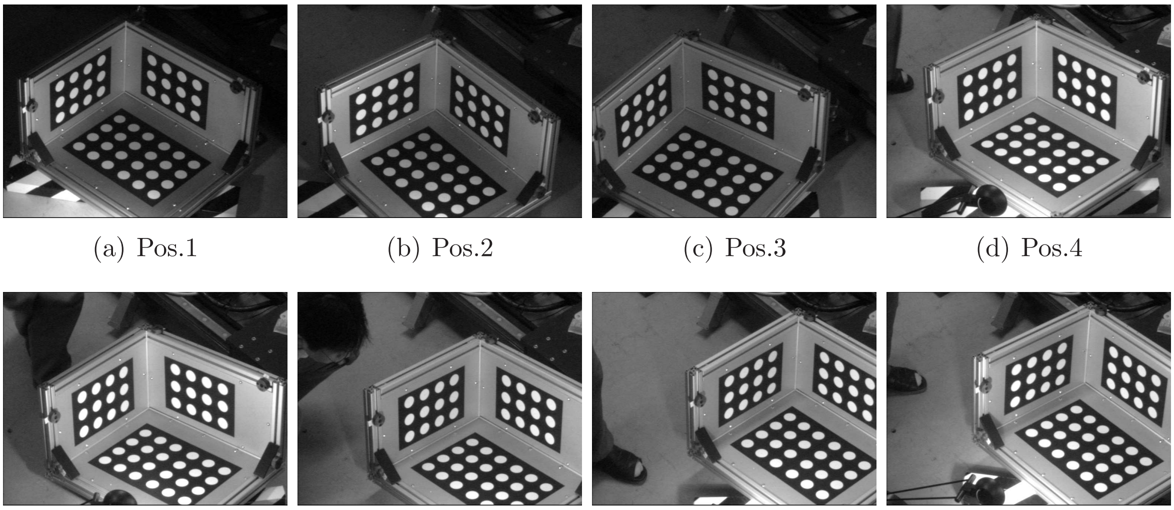
3.6.3 Convergence and stability

Whenever a nonlinear algorithm is employed, the convergence of the system and the stability of the final solution are always the issues. In the research field of the stability of the minimization algorithm in camera calibration, some valuable work has been done and a common opinion is arrived at: **if the iterative procedure is not properly designed, the minimization algorithm may be unstable or even diverges**.

Weng said in [15], if the distortion is included in the parameters space, the interaction between the distortion parameters and external parameters can lead to divergence, or even false solution. Therefore, he suggests to solve the distortion and the other parameters alternatively in each iteration.

However, zhuang has a different opinion in [31], if the image origin is assumed known, the calibration problem is reduced to a standard nonlinear square and any appropriate nonlinear least squares procedure can be applied to solve it with a good set of guess values. Thus he thinks that the iteration procedure should be so designed that the image origin and the remaining parameters can be estimated separately.

Although weng and zhuang have different opinions on designing the iteration, their common point is clear: *the camera parameters including internal and external parameters consist an over-dimensions space and the complex constraints between them makes it unsuitable to estimate them all together in an iterative procedure.* Therefore, we have done such an experiment: a calibration body, which is described in detail in section *calibration with a calibration body* from the next chapter, is placed successively into 8 positions, where a fixed camera can see it and the images are shown in figure 3.16. At each position, the camera is calibrated with the direct minimization method.


(a) Pos.1 (b) Pos.2 (c) Pos.3 (d) Pos.4

(e) Pos.5 (f) Pos.6 (g) Pos.7 (h) Pos.8

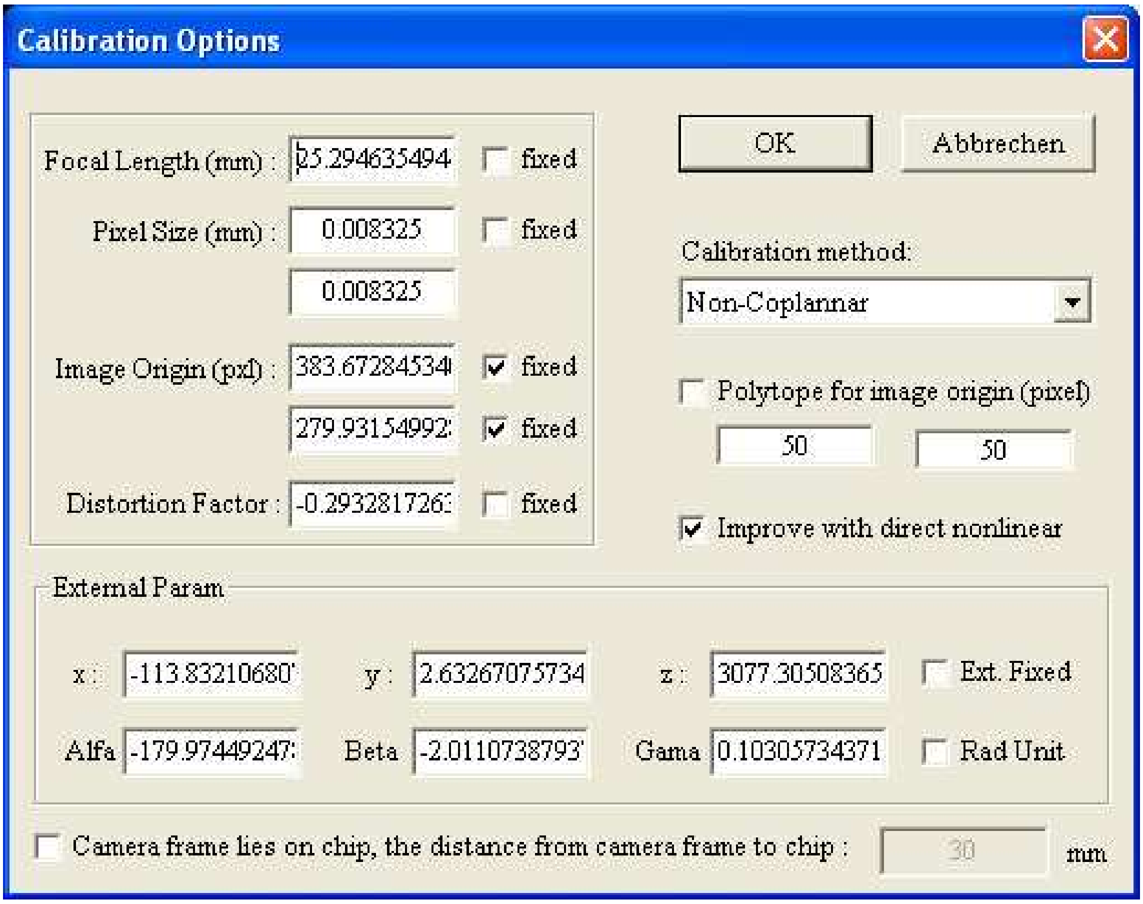
Figure 3.16: camera images for minimization algorithm

To analyze the interactions to each other, the internal parameters are much more valuable than the external parameters as a whole. That is why table 3.11 simply shows us the internal parameters.

	f_x	f_y	C_x	C_y	K		f	S_x	k
Pos.1	3057.25	3055.25	397.55	287.87	-0.256064	→	25.43	0.008320	-0.000396
Pos.2	3025.09	3024.16	397.26	263.91	-0.401871	→	25.18	0.008322	-0.000634
Pos.3	3043.98	3042.33	392.16	287.38	-0.341061	→	25.33	0.008321	-0.000532
Pos.4	3030.85	3028.16	297.97	315.31	-0.415928	→	25.21	0.008318	-0.000654
Pos.5	3029.79	3027.10	348.65	349.19	-0.410402	→	25.20	0.008318	-0.000646
Pos.6	3023.09	3020.97	396.39	342.70	-0.352578	→	25.15	0.008319	-0.000557
Pos.7	3049.92	3049.25	438.00	344.41	-0.255513	→	25.38	0.008323	-0.000397
Pos.8	3036.38	3034.24	389.44	336.34	-0.289593	→	25.26	0.008319	-0.000454
Average	3037.04	3035.18	382.18	315.89	-0.340376	→	25.27	0.008320	-0.000534
Max.Err	20.21	20.07	84.21	51.98	0.084863	→	0.17	0.000003	0.000138

Table 3.11: calibration testing for stability with S_y=0.008325

Since the calibrations are for the same camera, the internal parameters must be consistent or in a reasonable range. But what we see from table 3.11 are more differences than consistences: f_x and f_y, or clearer in forms of f and S_x, are quite consistent; C_x, C_y and k are distinct to each other to about 30%. Actually, we draw such a conclusion after experiments: **the camera parameters are profiled into two groups: C_x, C_y and $[R, \vec{t}]$, f and K. The interaction in groups is much greater than between groups, especially the image origin and the camera pose (mainly rx and ry), which correlate tightly**.



Calibration Options

Focal Length (mm) : 25.294635494 □ fixed

Pixel Size (mm) : 0.008325 □ fixed

0.008325

Image Origin (pxl) : 383.67284534 ☑ fixed

279.93154992 ☑ fixed

Distortion Factor : -0.293281726 □ fixed

OK Abbrechen

Calibration method: Non-Coplannar

□ Polytope for image origin (pixel) 50 50

☑ Improve with direct nonlinear

External Param

x : -113.83210680 y : 2.63267075734 z : 3077.30508365 □ Ext. Fixed

Alfa -179.97449247 Beta -2.0110738793 Gama 0.10305734371 □ Rad Unit

□ Camera frame lies on chip, the distance from camera frame to chip : 30 mm

Figure 3.17: calibration options

3.6.4 Iteration design

In order to prove the above conclusion, the iterative algorithm is so programmed that any parameter can be fixed in implement as shown in figure 3.17. With the flexible calibration options, we have tried some experiments by fixing successively the image origin and the distortion factor. For convenience, the first image from figure 3.16 is used and the experiments are carried out as follows

Distortion dominant: the image origin fixed

Thinking of the maximum errors of the image origin, the drifting ranges for C_x, C_y are set respectively to $(-30, +30)$ and $(-20, +20)$ from the base of $(397, 287)$ and the results are shown in table 3.12. As seen from the table 3.12, the image origin C_x, C_y differ in a range of (30, 20) pixels, and the influences to f_x, f_y are relatively slight, no more than 0.5% and its deviations are compensated mostly by the camera pose.

C_x	C_y	f_x	f_y	K	x	y	z	rx	ry	rz
367	267	3065.04	3062.80	-0.259716	2313.71	2093.14	2487.86	-129.362	-1.408	132.289
367	287	3064.12	3062.10	-0.224910	2313.56	2092.67	2487.25	-129.734	-1.412	132.287
367	307	3051.83	3049.91	-0.244054	2305.02	2084.87	2477.39	-130.106	-1.415	132.285
397	267	3064.53	3062.36	-0.261168	2313.06	2093.14	2487.83	-129.355	-1.761	131.862
397	287	3057.85	3055.84	-0.254797	2308.67	2088.94	2482.50	-129.726	-1.768	131.860
397	307	3047.91	3046.01	-0.266235	2301.84	2082.69	2474.46	-130.098	-1.774	131.858
427	267	3054.89	3052.47	-0.306805	2305.26	2087.38	2480.22	-129.344	-2.113	131.435
427	287	3048.92	3046.66	-0.302689	2301.47	2083.63	2475.32	-129.714	-2.122	131.433
427	307	3041.98	3039.87	-0.303425	2296.94	2079.30	2469.65	-130.085	-2.131	131.431
Average		3055.23	3053.11	-0.269311	2306.61	2087.31	2480.28	-129.725	-1.767	131.860
Max.Err		13.25	13.24	0.044401	9.68	8.01	10.63	0.382	0.363	0.429

Table 3.12: influences from image origin

Image origin dominant: the distortion factor fixed

Similarly the drifting range for distortion factor K is set to $(-0.08, +0.08)$ from the base of -0.25 and the results are shown in table 3.13.

K	f_x	f_y	C_x	C_y	x	y	z	rx	ry	rz
-0.1700	3075.17	3073.38	384.45	285.95	2321.76	2099.90	2496.65	-129.711	-1.620	132.038
-0.1900	3071.01	3069.18	386.88	286.21	2318.64	2097.26	2493.27	-129.715	-1.648	132.004
-0.2100	3066.84	3064.96	389.64	286.59	2315.51	2094.61	2489.86	-129.721	-1.681	131.965
-0.2300	3062.66	3060.73	392.78	287.08	2312.35	2091.97	2486.45	-129.729	-1.718	131.920
-0.2500	3058.51	3056.52	396.37	287.68	2309.20	2089.35	2483.05	-129.739	-1.760	131.869
-0.2700	3054.41	3052.35	400.42	288.31	2306.07	2086.79	2479.68	-129.749	-1.808	131.811
-0.2900	3050.46	3048.32	404.84	288.80	2303.01	2084.32	2476.41	-129.756	-1.860	131.748
-0.3100	3046.78	3044.53	409.27	288.72	2300.11	2082.03	2473.34	-129.752	-1.912	131.685
-0.3300	3043.49	3041.12	412.77	287.48	2297.45	2079.97	2470.57	-129.727	-1.953	131.636
Average	3058.81	3056.79	397.49	287.42	2309.34	2089.58	2483.25	-129.73	-1.77	131.85
Max.Err	16.36	16.59	15.28	1.47	12.41	10.32	13.40	0.02	0.18	0.22

Table 3.13: influences from radial distortion

As seen from the table 3.13, the distortion factor K affects the camera pose R very slightly. But it is more sensitive to other parameters than the image origin. Looking into both the table 3.12 and 3.13, we can imagine such a cycle procedure: the image origin is given with a deviations in (30, 20) and may result the deviations to the distortion about 0.04, which may cause to much smaller deviations to the image origin, no more than (8, 2) pixels. The situation is similar when we start with the image origin dominant procedure. Thus these are two appropriate design for the iterative procedure to implement the camera calibration. However, the **distortion dominant** principle is usually adopted in our applications for the following reasons

1. The cameras designed specially for industrial applications have good quality and the image origin is never be far away from the image center, which makes it possible for us to suppose the image origin to be the image center at the beginning.

2. Since the distortion K is sensitive to calibration, it is relatively difficult to give an appropriate start guess.

3. The image origin can be improved separately by searching after the minimization procedure. Furthermore, the cycle can be repeated till both the image origin C_x, C_y and the camera pose $R, \vec{t}$ converge.

4. In measurements, the distortion factor is much more sensitive than the image origin. If the image origin is off from the image center no more than 20 pixels, the measurement works still well, especially when the calibration procedure is online carried out and the calibration object is placed in the area where the object to be measured will stand in measuring procedure. More details can be found in [7].

3.6.5 Experimental results

For convenience of contrast, the images from figure 3.5 are used again and the final results are listed in table 3.14 and 3.15. In the minimization procedure, the image origin C_x, C_y are fixed and the initial guess for the other parameters are obtained from two methods: **i.A** denotes the values from the calibration for a distortion-free model and **i.B** denotes the values from the non-coplanar calibration followed by an image origin searching procedure.

	f_x	f_y	C_x	C_y	K
1.A	3038.39	3034.71	383.67	279.93	-0.293282
1.B	3038.42	3034.75	390.68	273.45	-0.289484
Δ	0.03	0.04	7.01	6.48	0.003798
2.A	3049.72	3045.86	383.59	300.90	-0.312715
2.B	3049.72	3045.86	381.28	288.73	-0.309871
Δ	0.00	0.00	2.31	12.17	0.002844
3.A	3055.59	3051.79	416.09	296.51	-0.318183
3.B	3055.61	3051.83	395.87	284.14	-0.320966
Δ	0.02	0.04	20.22	12.37	0.002783
4.A	3020.99	3017.11	384.18	297.17	-0.305812
4.B	3021.14	3017.26	392.09	289.72	-0.305936
Δ	0.15	0.15	7.91	7.45	0.000124

Table 3.14: internal parameters

For better understanding, table 3.14 and 3.15 should be observed together with table 3.6: although the initial guess for the camera parameters are quite different, they are optimized greatly by the minimization procedure and converge to the similar results. Since C_x, C_y are fixed, their differences are compensated mainly by rx and ry. The deviations from both calibrations are shown in figure 3.18.

	x	y	z	rx	ry	rz
1.A	-113.83	2.63	3077.31	-179.974	-2.011	0.103
1.B	-113.84	2.65	3077.44	-179.852	-2.143	0.103
Δ	0.01	0.02	0.13	0.122	0.132	0.000
2.A	21.20	2.43	3080.66	179.496	0.346	-0.026
2.B	21.20	2.46	3080.73	179.725	0.389	-0.027
Δ	0.00	0.03	0.07	0.229	0.043	0.001
3.A	3.32	3.03	3081.14	179.570	-0.494	0.203
3.B	3.32	3.08	3081.06	179.801	-0.115	0.200
Δ	0.00	0.05	0.08	0.231	0.379	0.003
4.A	45.18	2.76	3081.03	179.465	1.246	0.245
4.B	45.18	2.78	3081.17	179.606	1.096	0.246
Δ	0.00	0.02	0.14	0.141	0.150	0.001

Table 3.15: external parameters

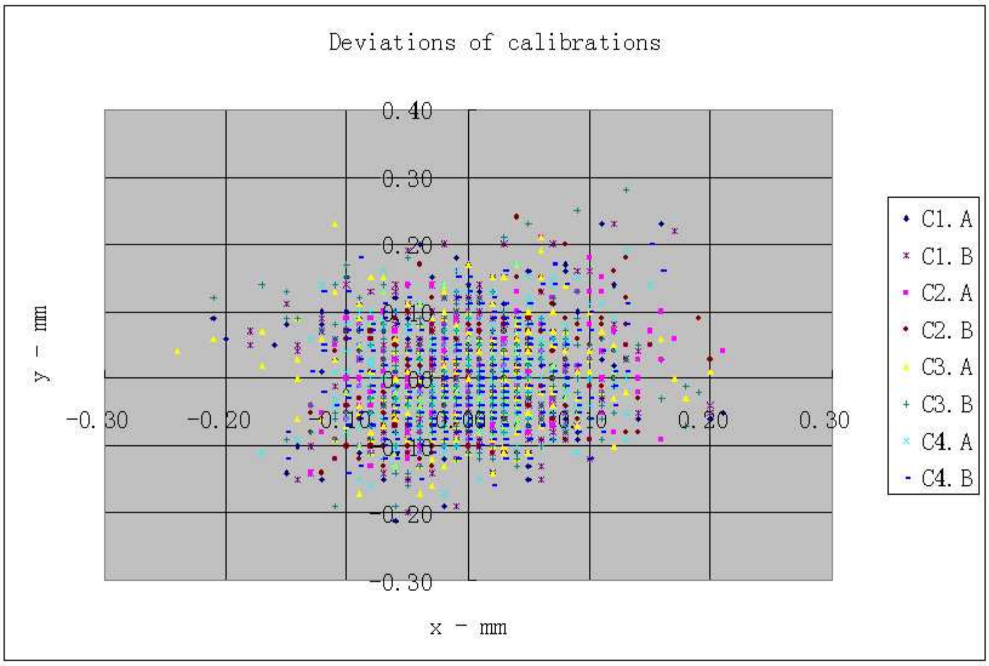


Figure 3.18: deviations from minimization procedures

3.7 Chapter review

In this chapter many algorithms about camera calibration are demonstrated in details. Some of them educe independent methods for camera calibration, and some of them work only for implement or improvement to some calibrations and is applied usually with other calibration methods. Here we make a conclusion for all of them.

3.7.1 Property overview

Since different methods need different preconditions and result in different outputs, the properties of the calibration methods are listed in table 3.16 for an overview and comparison.

	coplanar	non-coplanar	distortion free	vanish point	origin search	minimization
estimation	f_x, k, T	f_x, f_y, k, T	all but k	f_x, f_y, C_x, C_y	C_x, C_y	all
condition	$C_x, C_y, f_x = f_y$	C_x, C_y	k	k	f_x, f_y, k, T	—
object	coplanar	non-coplanar	non-coplanar	rectangle	—	non-coplanar
stability	good	good	ok	ok	good	very good
accuracy	bad	good	ok	bad	very good	best
robust	good	very good	very good	ok	ok	good
easiness	good	ok	ok	very good	best	ok

Table 3.16: properties overview of the calibration methods

where $T = (R, \vec{t})$ is the camera pose.

3.7.2 Applicable situations

According to the properties from above table and the experiences in practice, the applicable situations or rules of the calibration methods are roughly concluded as follows

1. Calibration with coplanar points
   When the calibration points with known coordinates are coplanar, we should use this method to calibrate a camera. Since only two $fx = f_y$ and k of the internal parameters are calibrated, the results are too rough for most practical applications. However, the easiness of the calibration condition in preparation makes it very popular to the general public for the desktop visions.

2. Calibration with non-coplanar points
   If plenty of calibration points are non-coplanar and well prepared with accurate coordinates, this calibration works well for the general applications. The disadvantages have two points: one is that the image origin must be given, and the other is that the accuracy is not enough for the accurate positioning applications, especially for the pose estimation in the robot vision applications.

3. Calibration for a distortion-free model
   Compared to the *calibration with non-coplanar points*, the only advantage of the calibration for a distortion free model is to estimate the image origin C_x, C_y, but the distortion is neglected, which causes the worse stability and accuracy. Thus this method is useful only after the distortion factor is known and a distortion alignment is made to the calibration points.

4. Calibration with vanishing points
   As same as the *calibration for a distortion free model*, the distortion is neglected. Since the calibration object moves to some arbitrary poses, this calibration method is only for determining the internal parameters in most of time. Thinking that the calibration object is very simple and the calibration technique is relatively new, we developed it mainly for testing, and comparison and the method has never applied to our applications.

5. Image origin searching
   Strictly saying, image origin searching is not a method for camera calibration since it estimates only two internal parameters with other parameters known. The method is used usually in an iterative procedure: given an initial guess to the image origin, the other parameters are estimated with any calibration method; with the obtained parameters, the image origin is improved by the image origin searching algorithm. This cycle is repeated till the image origin converges.

6. Refinement with nonlinear minimization
   This method works well only when the iterative procedure is appropriately designed and a relatively good set of initial guess of all camera parameters is given. To get the initial guess, one from the above calibration should be selected. Thus this procedure is only for improving the results from the single or any combinations of the above calibration methods.

7. Combination solutions
   As discussed above, no single calibration method works perfectly for all camera parameters. For the accurate positioning applications, we usually combine two or more methods above to complete a calibration procedure.

Generally speaking, the *calibration with coplanar points*, or the *calibration for a distortion free model*, or the *calibration with vanishing points* is acceptable usually to the general public. The *calibration with accurate non-coplanar points* may be applied to some practical applications. However, the industrial applications for accurate positioning usually need some appropriate *combination solutions* of these methods and the *image origin searching* or the *refinement with nonlinear minimization.*

3.7.3 Contributions

Although there is no completely new calibration method proposed in this chapter, we have put our own opinions boldly forward and made some improvements to these calibration methods.

1. The extent configuration to the *calibration with coplanar points* is proposed and the degenerate configurations of calibrations with both coplanar and non-coplanar are discussed in details.

2. In *calibration for a distortion free model*, the sign determination for t_z is strictly given in mathematics and the degenerate configurations are throughout analyzed.

3. Although the concept of the *calibration with vanishing points* is nothing new, the calibration procedure is built on our own understanding. The degenerate configuration is demonstrated and an improvement on calibration object is given.

4. A best-fit procedure between coordinates frames is proved in mathematics and solved in programming.
5. The *image origin searching* is improved by bringing the other calibration methods into the iteration procedure.
6. The *nonlinear minimization* is throughout analyzed and a new opinion about iterative design is brought forward. With specially designed codes and plenty of testing, an appropriate design is given for the camera calibrations to the industrial applications.

Chapter 4

Calibration Approaches Applied to Practice

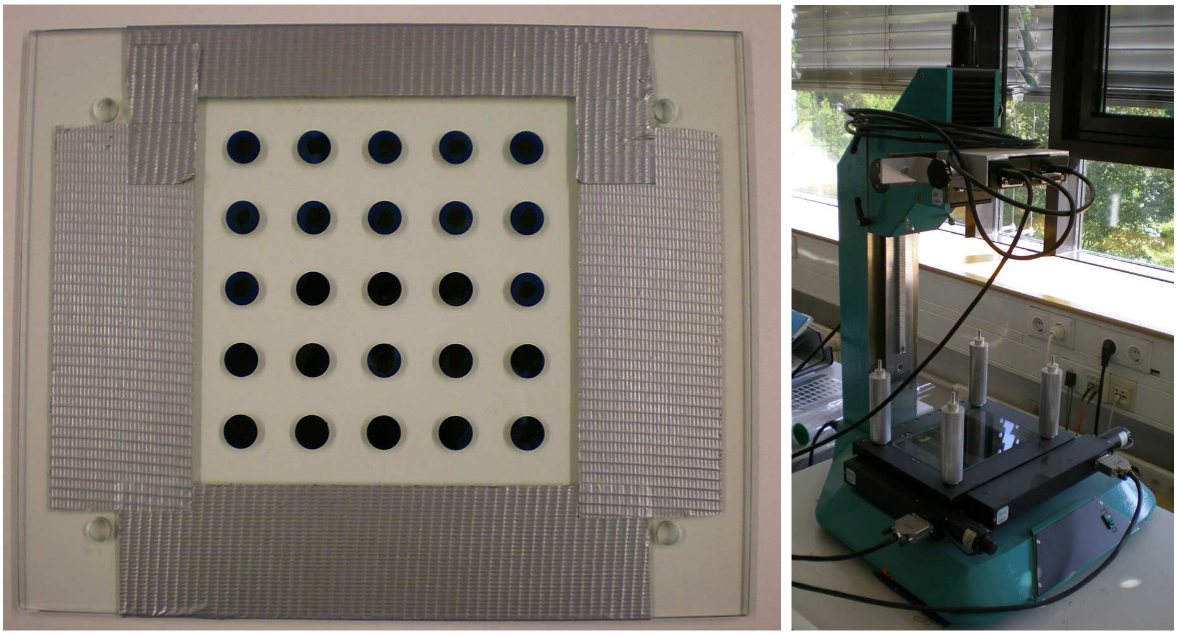
In the last chapter, a quite of algorithms or methods for camera calibration are well described in mathematics. In applying them to practice, they may have different forms, or some additional tools and setups are needed according to the specific application environments. In this chapter, some practical approaches to camera calibration in working area are proposed in details.

4.1 Calibration with calibration board

Calibration with a calibration board is the most popular and traditional calibration method. Although it is our first attempt to camera calibration in laboratory and little attention is given to its practical values, this approach is appropriate for accurate calibration to the stereo sensors.

4.1.1 Calibration board and setup

At the beginning, a calibration board is produced by printing an array of patterns on a paper sticked onto a metal board. But it is too inaccurate by printing and the deviation may be 1 mm or even more. As seen in figure 4.1, we finally have designed a board made of glass with circle patterns in an array. The coordinates of the patterns have deviations of no more than 0.02 mm and are constant against the change of temperature. The calibration setup equipment is called *microscope*, which can move independently in three orthogonal directions (X, Y, Z) with an accuracy of 0.01 mm.



(a) calibration board (b) calibration setup

Figure 4.1: camera calibration with a calibration board

When calibrating, the camera or stereo sensor is mounted on the top of the vertical part, the board is fixed on the horizontal plane of the microscope and the setup serves for accurate coordinates collecting. Although the microscope knows its movement and the current position and the arrange details of the patterns on the board is also known, the mounting mode of the board onto microscope should be so properly adjusted in mechanics that the axes directions of the board frame are parallel to the moving directions of microscope. This procedure is completed carefully once with a laser tracker.

4.1.2 Solving calibration

Since the camera is mounted on the microscope for calibration and remounted in the working environment for measurements, the camera pose should be determined again.

If we calibrate the cameras from a stereo vision system, where the cameras are relatively fixed and form a stereo sensor as a whole, the method works well since the stereo sensor is ready for measurements after calibration. Actually the setup is used by us only for calibrating the cameras from a stereo sensor. The calibration algorithm can be any or combination methods from the last chapter and the calibration procedure is as follows

1. Move the board into an appropriate position and define the current board frame as the reference frame for the cameras of the sensor, namely sensor frame.
2. The cameras are calibrated successively with any suitable algorithm in the sensor frame.
3. Demount the stereo sensor as a whole and it is ready for being applied to any measurement in the sensor frame.

4.1.3 Experimental results

The experiment is carried out for a stereo sensor with two cameras parallel mounted and the cameras are of JAI CV-M50 and equipped with a lens of 6 mm focal length.

	x	y	z	rx	ry	rz		
C1	-28.78	40.17	224.93	177.155	0.141	90.822		
C2	-28.81	-36.42	223.18	177.600	-0.706	90.309		
	f_x	f_y	C_x	C_y	K	f	S_x	k
C1	676.97	677.30	323.01	232.59	-0.225809	5.639	0.008329	-0.007102
C2	677.22	677.68	333.59	232.47	-0.222593	5.642	0.008331	-0.006993

Table 4.1: body calibration results with S_y=0.008325

From the values in table 4.1, we have to say that the calibration results, both the internal parameters and the cameras poses, are completely consistent with the real situation as seen from figure 4.1. The results are also approved by the deviations of the calibrations in figure 4.2, where the maximum deviation is under 0.08 mm.

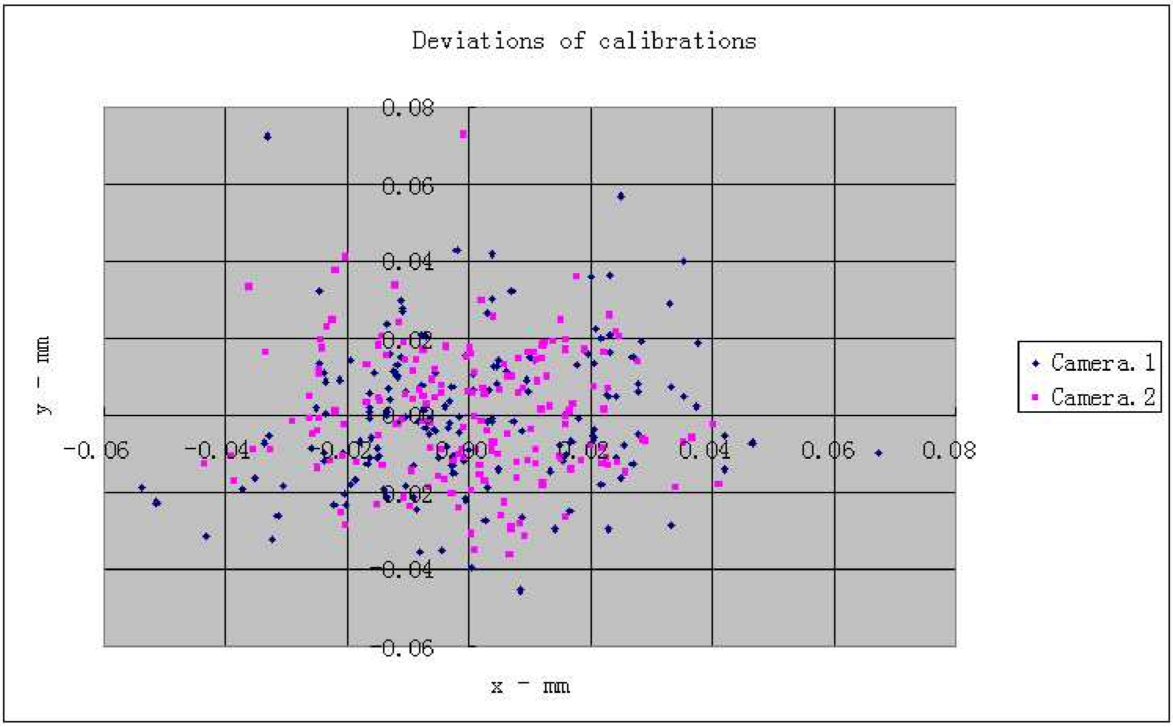


Figure 4.2: deviations from board calibrations

4.2 Calibration with robot tools

As the title mentioned, a practical approach to camera calibration with the aid of a set of specially designed robot tools is proposed in this section.

4.2.1 Motivation from applications

In car manufacturing industry, the first stage of process work is implemented in the press shop, where the steel sheets are pressed into all kinds of car body parts by the press machines. For safety reason, the work to feed the steel sheets into press machines is done more and more often by robots.

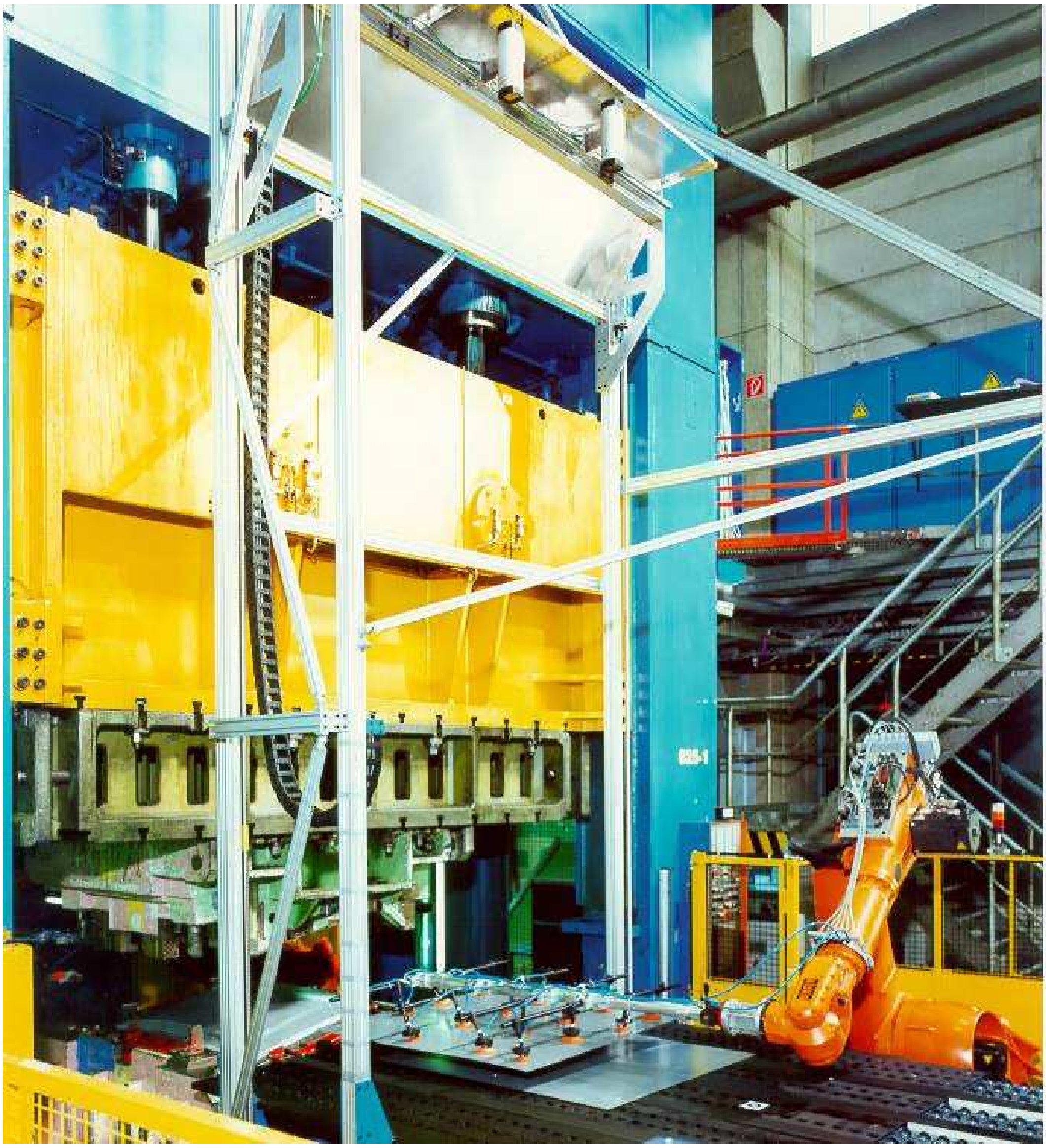
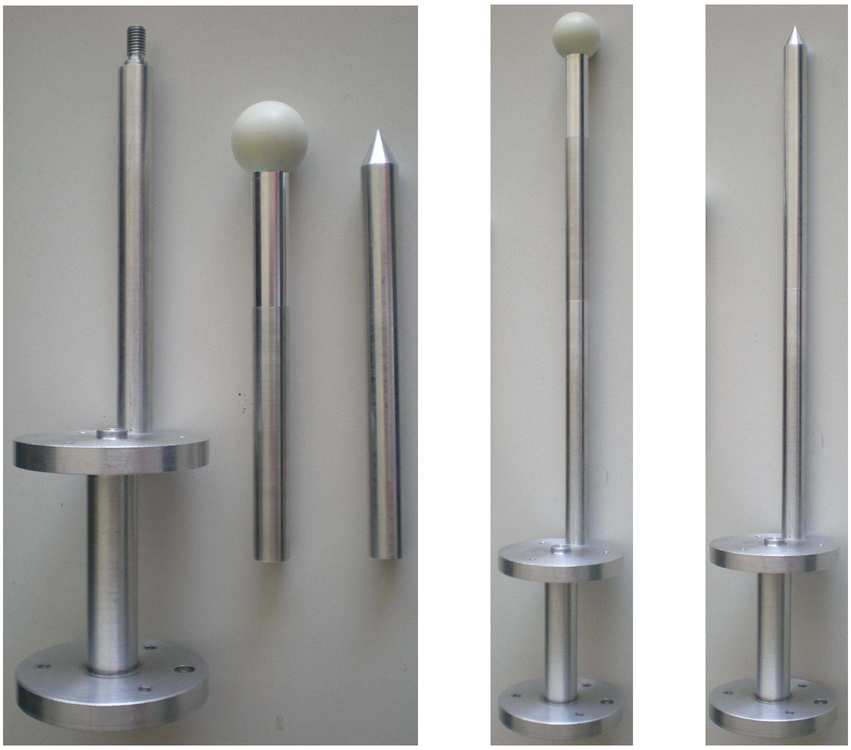


Figure 4.3: vision solution for steel sheets feeding in press shop

As demonstrated in figure 4.3, the steel sheet is delivered by a conveyor with high speed and stops before the press machine by braking the conveyor. However, the sheet is easy to glide in an arbitrary mode when brakes, especially when the sheet has just gone through an oil-washing machine. Therefore, one can never expect that the sheet always stops at the same position and waits for the robot to feed it correctly into the press machine. Our solution is a vision system with two cameras, which are fixed vertically and far away above the conveyor belt. When the sheet stops, the vision system is triggered to measure the current pose of the sheet and guide the robot to adjust its pose for grabbing the sheet at the part and lay it correctly into the press machine. Since this is an easy application and there is nothing else except the cameras to be calibrated, it is not worthwhile to get an external measurement system only for calibrating two cameras. It is natural to think of utilizing the robot to calibrate the cameras. If a pattern mounted on the robot hand moves in the camera sight-field by driving the robot and at the same time the robot remembers the positions of the pattern, the camera can be calibrated.

4.2.2 Robot tools

In the calibration procedure, the robot tools act as calibration objects for supplying the 3D coordinates.



(a) tool set (b) ball (c) tine

Figure 4.4: robot tools used for camera calibration

As seen in figure 4.4, the tools set is composed of two parts: a ball and a tine. Both the ball and the tine are at the end of a steel stick, whose another end has screw thread for mounting. Of course, in the figure there is still the base mechanics which is only for mounting the steel stick onto the robot hand and can have any shape or form.

Ball for camera calibration: in the camera calibration procedure, the ball is mounted on the robot hand and moves to several positions, where the camera sees the ball and searches it in the camera image; at the same time the robot controller gives the 3D coordinates of the ball in robot base frame.

Tine for tool calibration: in order to get the coordinates of the ball in robot base frame from the robot controller, the coordinates of the ball in robot hand frame and its mounting mode must be known. The tine is so designed in mechanics that the center point of the ball is exactly in the same position as the tine end when both of them are mounted alternatively on the robot hand. If the coordinates of the tine end in robot hand frame are obtained, the coordinates of the ball are also known to the robot. The procedure to estimate the coordinates of the tine, also the ball, is called tool calibration. The tine is introduced absolutely for solving this problem.

4.2.3 Robot tool calibration

The robot tool referred in the calibration procedure is the tine, the part C in figure 4.4, with which the calibration procedure is carried out as shown in figure 4.5.

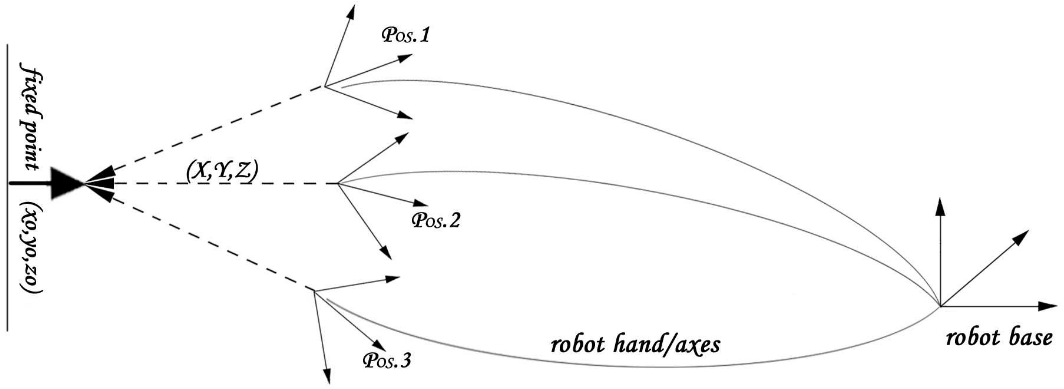


Figure 4.5: procedure of robot tool calibration

With this tool on the robot hand mounted, one can drive the robot for the tine to touch any point exactly within the working space. If a reasonable fixed point is selected and the robot is driven from some different directions to let the tine touch the fixed point, the calibration of the robot tool will be solved. The calibration procedure can be summarized into the following steps:

1. Select a fixed point which the robot with tine can reach, and its coordinates need not be known;

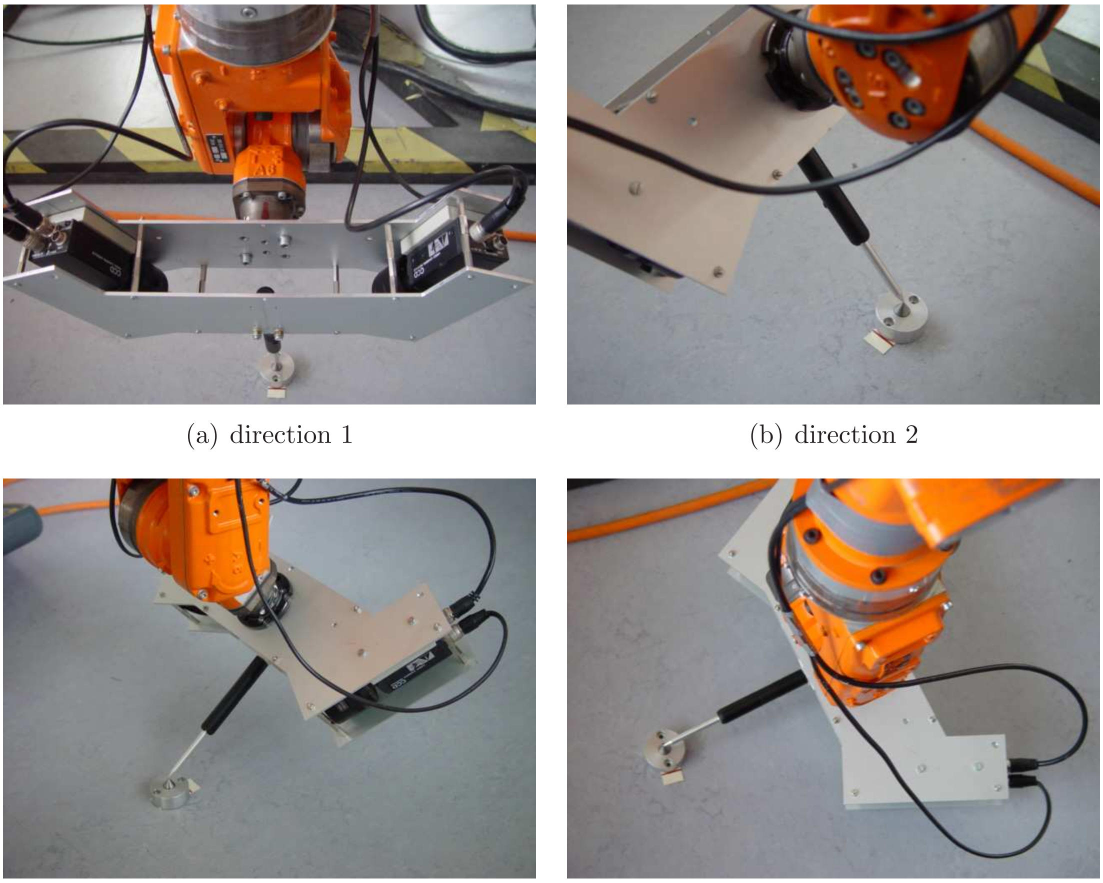
2. Drive the robot in different directions (at least two) to let the tine touch the fixed point exactly;

3. Calculate the coordinates of the tine in the robot hand frame;

4. Set the calculated values for the robot tool and activate the robot tool;

5. The robot tool is ready: the robot can give the correct coordinates of the center point of the ball with respect to the robot base.

Since both the fixed point with respect to the robot base frame and the tine with respect to the robot hand frame are kept unchanged whenever the tine touches the fixed point, the mathematics for calculation is obvious.



(a) direction 1

(b) direction 2

(c) direction 3

(d) direction 4

Figure 4.6: tool calibration from kuka robot

If the coordinates of the tine in the robot hand frame is denoted as $(x, y, z)^T$ and the fixed point as $(x_0, y_0, z_0)^T$, then

$$
{}^{B}T_{H}^{1} \bullet \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \cdots = {}^{B}T_{H}^{n} \bullet \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x_0 \\ y_0 \\ z_0 \\ 1 \end{pmatrix} \tag{4.1}
$$

Considering two directions and writing the transformation into rotation R and translation $\vec{t}$, one gets

$$
{}^{B}R_{H}^{1} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \vec{t^1} = {}^{B}R_{H}^{2} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \vec{t^2} \tag{4.2}
$$

and consequently the tool coordinates can be obtained

$$
\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \left({}^{B}R_{H}^{1} - {}^{B}R_{H}^{2} \right)^{-1} \cdot (\vec{t^2} - \vec{t^1}) \tag{4.3}
$$

If we have more than two directions, we will get an over-determined linear system, from which a set of best-fit results will be obtained by minimum errors. Actually such a module for robot tool calibration is already integrated into the robot controller software, e.g. KUKA robot needs four directions to complete the tool calibration, which is as shown in figure 4.6.

4.2.4 Camera calibration

Once the robot tool is calibrated, the robot can act as a flexible calibration setup for supplying accurate 3D coordinates of the calibration points, and the calibration algorithm can be a combination of the methods as described in the previous chapters. Therefore, the complete calibration procedure for the approach with the aid of a robot tool consists of the following steps:

1. Mount robot tool with a tine on the robot hand, and carry out the tool calibration procedure;
2. Activate the calibrated tool with correct values, and change to the other tool with a ball;
3. Drive the robot to some positions for gathering calibration information, which includes 3D coordinates of the ball in the world scene and 2D coordinates in the camera images;
4. Calibrate the camera by the method of non-coplanar calibration points with a set of initial parameters;
5. Refine the camera parameters by the direct nonlinear minimization, and combine the estimation of the image origin if necessary;
6. If the robot base frame is the reference frame, the calibration is completed; if not, compare the robot base frame with the reference frame, and consequently convert the camera pose into the reference frame.

Remark: In the real applications, the vision system from VMT has a communication module through a configurable protocol, such as profibus, serial port and so on, for controlling the robot. Thus the calibration procedure is usually carried out automatically in few minutes.

In fact, the calibration approach is also suitable for a robot hand camera if the following changes are made:

1. **Trade the positions with each other**: the camera is mounted on the robot hand and the tool with a ball is fixed in an unknown position.
2. **Get the coordinates of the ball in the robot hand frame**: replace firstly the tool with a ball by the tool with a tine and mount another calibrated tool with a tine on the robot hand. By driving the robot to touch the two tines exactly together, the coordinates of the tine in the robot base frame can be read from the robot controller, consequently the coordinates with respect to the robot hand frame is obtained.

3. **Collect the information for calibration**: drive the robot to some positions for the camera to search the ball in the images. At this moment, the camera is assumed to be still and the ball is conducting the opposite movements.

4. **The reference frame is the robot hand frame**: the estimation algorithm is the same as that of a stationary camera. However, the reference frame is not the robot base frame, but the robot hand frame.

Remark: In the step for information collecting, the robot can not move freely. If and only if the camera has no rotating movements, it will be a valid assumption that the camera is still and the ball is moving in the opposite directions. In fact, this kind of movements without rotation is one of the normal operation modes for all kinds of robots.

4.2.5 Experimental results

The example setup in our laboratory is shown in figure 4.7, where we can calibrate both the camera fixed on the floor and the camera mounted on the robot hand.

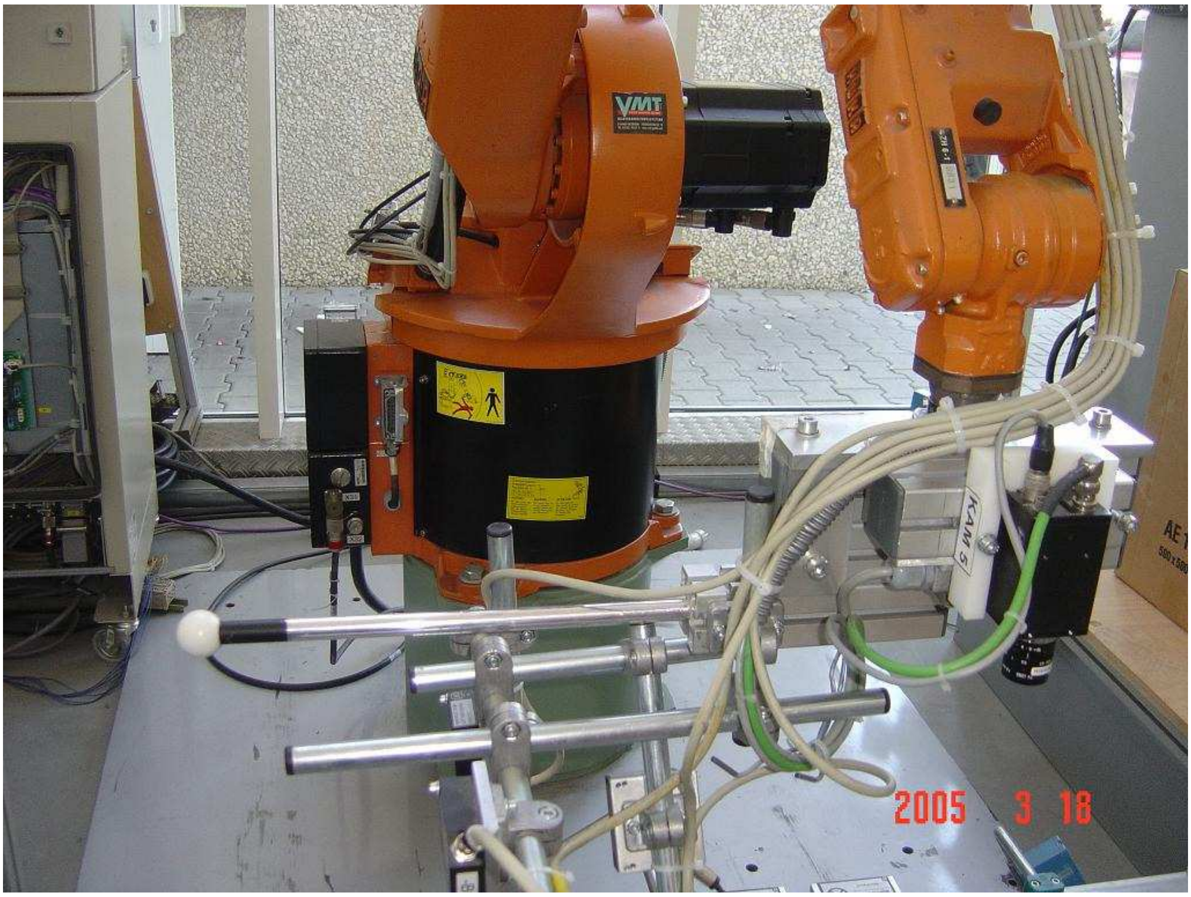


Figure 4.7: camera calibration with robot tools in laboratory

With this setup, a stationary camera fixed in the floor is calibrated with 27 positions, whose coordinates in both the image frame and the world frame are listed in table 4.2. For contrast, two methods are implemented in calibration: the first one is to calibrate all parameters with the nonlinear minimization procedure, and the second one is a combination of the image origin searching procedure and the nonlinear minimization procedure. The results are shown in table 4.3.

	X	Y	x	y	z		X	Y	x	y	z
1	31.51	14.91	643.5	-899	243.94	15	206.02	4.92	583.5	-899	243.94
2	-152.46	25.02	703.5	-899	278.94	16	199.74	-97.63	583.5	-934	243.94
3	-145.54	132.88	703.5	-864	278.94	17	25.10	-86.88	643.5	-934	243.94
4	39.15	122.34	643.5	-864	278.94	18	31.45	15.15	643.5	-899	243.94
5	223.77	111.44	583.5	-864	278.94	19	30.45	16.18	643.5	-899	208.94
6	216.86	2.57	583.5	-899	278.94	20	-136.44	26.30	703.5	-899	208.94
7	210.55	-105.57	583.5	-934	278.94	21	-130.41	123.53	703.5	-864	208.94
8	26.32	-93.90	643.5	-934	278.94	22	36.40	114.01	643.5	-864	208.94
9	-158.56	-83.02	703.5	-934	278.94	23	202.11	103.71	583.5	-864	208.94
10	-149.22	-76.31	703.5	-934	243.94	24	195.98	6.39	583.5	-899	208.94
11	-143.42	26.12	703.5	-899	243.94	25	190.41	-90.43	583.5	-934	208.94
12	-137.47	128.06	703.5	-864	243.94	26	24.10	-80.84	643.5	-934	208.94
13	37.68	117.84	643.5	-864	243.94	27	-141.40	-70.77	703.5	-934	208.94
14	212.19	107.45	583.5	-864	243.94						

Table 4.2: calibration points

	f_x	f_y	C_x	C_y	K	
direct min	1920.36	1926.98	-55.61	7.29	-0.280687	
combination	1921.42	1928.59	-17.90	-14.20	-0.210960	
	x	y	z	rx	ry	rz
direct min	651.76	-891.52	900.05	-179.067	1.915	176.606
combination	651.70	-891.55	900.87	-178.432	0.793	176.597

Table 4.3: calibration results with robot tools

Although the deviations from both calibrations are in the same level, as shown in figure 4.8, the results from the combination method are considered to be closer to the real values. In order to have a contrast to the current method from VMT GmbH, the eleven parameters in table 4.4 are converted into twelve parameters, which have clear physical meanings and can be directly used in a vision software.

	f	C_x	C_y	S_x	S_y	k
VMT	16.00	0.00	0.00	0.008325	0.008325	-0.00080
Here	16.06	-17.90	-14.20	0.008356	0.008325	-0.000818
Diff	0.06	17.90	14.20	0.000031	0.000000	0.000018
	x	y	z	rx	ry	rz
VMT	652.07	-891.18	930.98	-178.870	0.220	176.580
Here	651.70	-891.55	900.87	-178.432	0.793	176.597
Diff	0.37	0.37	30.11	0.438	0.573	0.017

Table 4.4: contrast of the the calibration results

In table 4.4, there are three main differences: C_x, C_y, z and rx, ry. Since VMT uses default internal parameters and calibrates only external parameters, but all the camera parameters are calibrated here.

1. **Difference of the image origin**: as referred before, the image origin is the intersection between the optical axis and the chip plane. The deviation caused from 17.9 pixels, or 0.149 mm on the camera chip, is possible for a camera with a c-mount lens.
2. **Difference of the shift in z-direction**: for easily applying some practical calibration approaches, the camera frame in VMT software is defined with regard to the camera shell, not in the camera lens.
3. **Difference of the rotation around x- and y-axis**: the difference may be the main reason for the differences of the image origin. C_x affects ry and C_y affects rx.

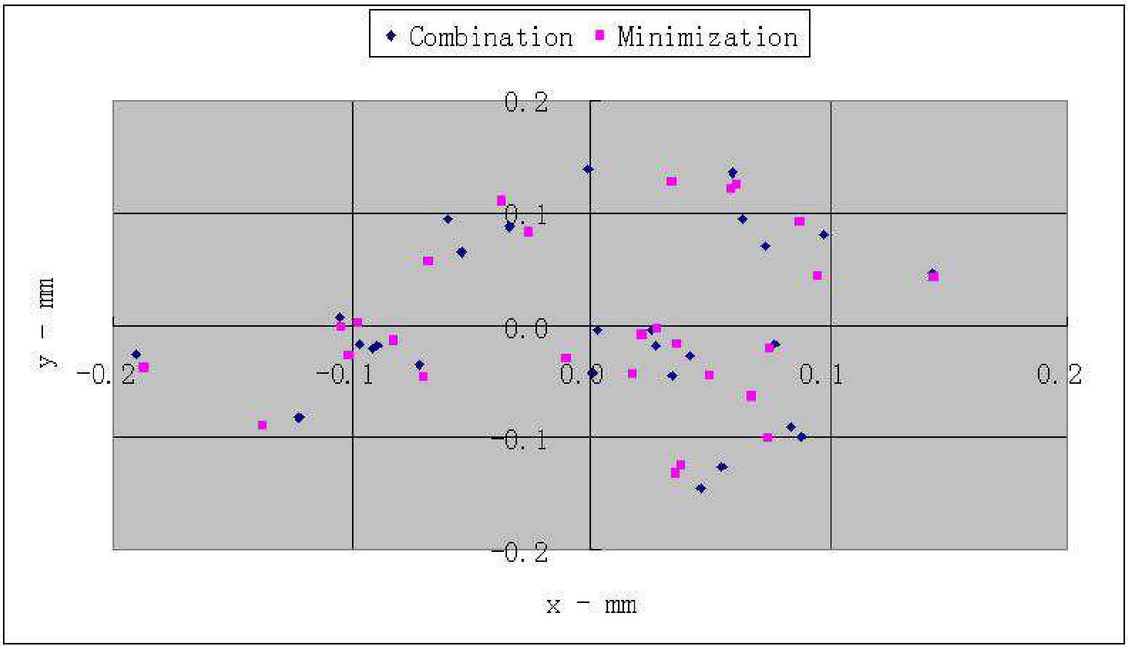


Figure 4.8: deviations from robot tools calibration

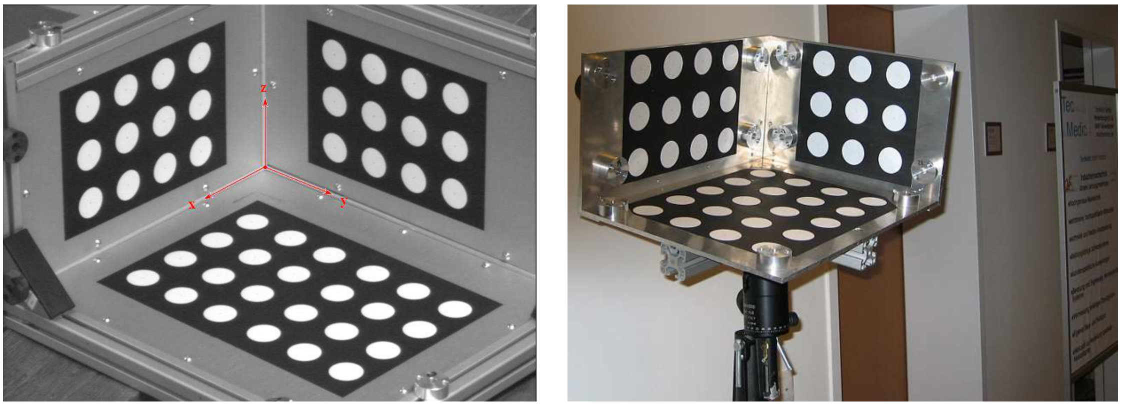
Remark: this approach for camera calibration is designed at the beginning only for applications in press shop, since there is always a robot and the conveyor usually has a dark color. Moreover, the required accuracy for the application is about 3 mm for the press machine. The applications from VMT systems with this approach to camera calibration can be found in press shops from Daimler-Chrysler Bremen, FAW-VW Changchun and Beijing Benz-Daimler-Chrysler.

4.3 Calibration with a calibration body

As we know, calibration boards are widely used in camera calibration. Since the calibration with a calibration board in automobile industry is protected by a patent in Germany, we have to think of other practical methods and thus an approach to camera calibration with a calibration body is introduced in applications.

4.3.1 Calibration body

The figure 4.9 shows two calibration bodies, the large one is for laboratory test and the small one for the practical applications. A calibration body is composed of three planes, where locate some calibration patterns that build up a calibration space. On the calibration body, there are some aluminum adapters, which serves as the control patterns for determining the pose of the calibration body.



(a) the large calibration body

(b) the small calibration body

Figure 4.9: calibration bodies for camera calibration

Before being used in applications, the calibration body must be prepared in laboratory as follows

1. **Object frame**: first of all, an object frame, or called body frame, on the calibration body must be set up. For example, the red lines marked in figure 4.9.
2. **Calibration patterns**: all calibration patterns must be numbered and their coordinates with respect to the object frame must be accurately measured. The patterns are usually numbered in such a way that the 2D-3D pairs matching in pattern recognition becomes as easy as possible in programming.
3. **Controlling patterns**: in the same way, the control patterns must also be numbered and in the object frame accurately measured.

To apply a ready calibration body in applications for camera calibration, the camera pose from the calibration results is with respect to the calibration body frame. Thus the pose of the body with respect to a reference frame should be determined and consequently the camera pose with respect to the reference frame is obtained. With a laser tracker, the controlling patterns can be measured in reference frame. Meanwhile, their coordinates in body frame are known and the transformation between the body frame and the reference frame is solved by a **best-fit** procedure.

4.3.2 Calibration procedure

Through the description above, the calibration procedure with a ready calibration body is obvious as follows

1. **Initialization for the image origin**: set up a default set of values for the image origin, e.g. the image center;

2. **Non-coplanar calibration**: move the calibration body into such a pose that the whole calibration body is in the camera sight field and the camera image is relatively clear, then calibrate the camera in calibration body frame with the linear method named *calibration with non-coplanar points*.

3. **Image origin searching**: with the results from *calibration with non-coplanar points*, new values of the image origin are obtained by an image origin searching procedure named **polytope** method.

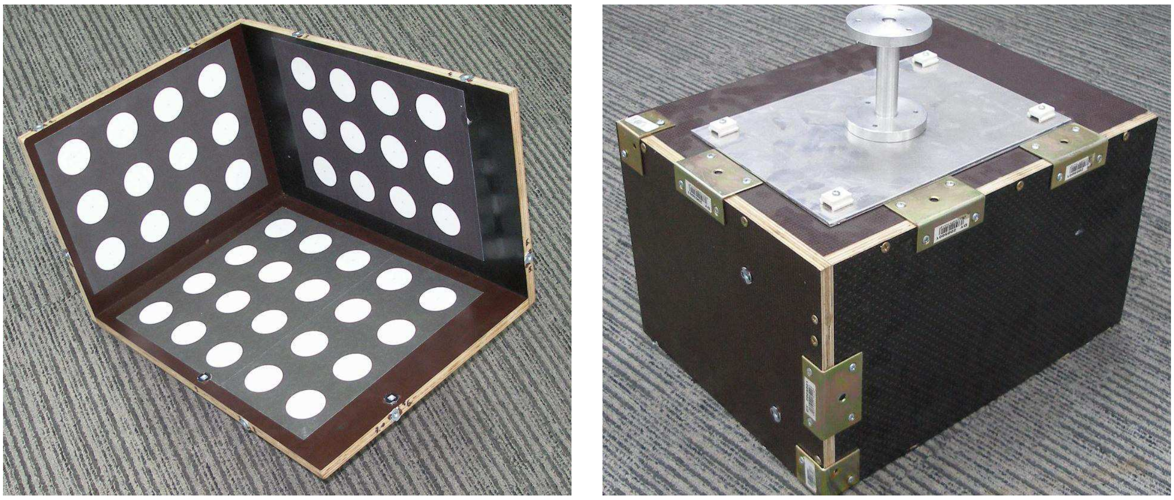
4. **Refinement with direct nonlinear minimization**: with the image origin from the last step being fixed, other camera parameters are refined by a direct nonlinear minimization.

5. **Camera pose with respect to the world frame**: determine the transformation between the calibration body frame and the world frame and the camera pose with respect to the world frame is obtained.

Remark: we notice that a laser tracker is involved in the above procedure and indeed this is a strict additional condition. Actually the approach is applied mainly to our multi-camera system to measure car bodies in painting or assembly shops, where there are many components, such as car bodies, robots, robot linear tracks and so on, need to be initialized and an external measurement system is in one way or another necessary in the startup procedure.

4.3.3 Experimental results

Since this approach has been applied to VMT system since 2006, the data sources for this experiment are from a practical project realized in 2007 in Volkswagen Shanghai, where a VMT system with four cameras is installed in painting shop for PVC fine sealing at the UBS station for the new car model *Lavida*.



(a) front view

(b) back view

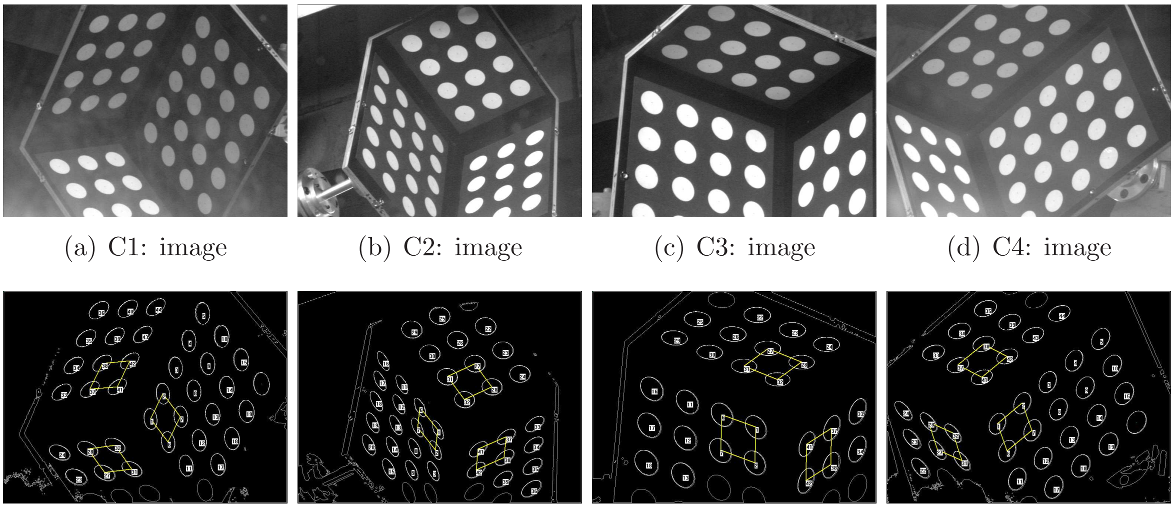
Figure 4.10: calibration body used in SVW project

For the use in China, we have produced a new calibration body for this project as shown in figure 4.10. With contrast to the old ones, this calibration body is smaller in size, lighter in weight and more flexible in mechanics for mounting on a robot hand, a framework or any other installations.

	x	y	z		x	y	z		x	y	z
01	55.00	61.00	0.00	16	249.56	60.46	0.32	31	120.71	0.48	60.20
02	55.03	125.53	0.42	17	249.80	125.40	0.68	32	55.97	0.91	59.57
03	55.10	190.42	0.52	18	249.76	190.31	0.68	33	-1.90	101.95	192.33
04	55.07	255.77	0.25	19	249.66	256.00	0.51	34	-1.61	166.65	191.67
05	54.72	320.79	-0.28	20	249.69	321.10	-0.09	35	-1.28	231.54	191.22
06	120.03	60.50	0.10	21	249.49	-1.11	191.82	36	-0.97	296.09	190.86
07	119.87	125.71	0.47	22	184.40	-0.22	190.82	37	-1.47	101.29	127.76
08	120.02	190.41	0.52	23	119.27	0.53	189.88	38	-0.98	166.24	127.04
09	119.79	256.21	0.31	24	54.71	1.15	189.30	39	-0.60	231.05	126.54
10	119.69	321.16	-0.15	25	249.92	-0.89	126.48	40	-0.21	295.83	126.01
11	185.07	60.29	0.26	26	185.11	-0.06	125.94	41	-0.96	100.43	62.68
12	184.91	125.25	0.57	27	120.47	0.62	125.21	42	-0.38	165.63	62.08
13	184.86	190.41	0.59	28	55.19	1.10	124.68	43	0.06	230.12	61.56
14	184.77	256.16	0.40	29	250.76	-0.87	61.17	44	0.46	295.14	60.75
15	184.77	320.86	-0.08	30	185.63	-0.18	60.76				

Table 4.5: coordinates of the calibration patterns in the body frame

The coordinates of the calibration patterns on the body are listed in table 4.5. With these coordinates, we can reckon approximately the origin and axes of the calibration body frame. Mounted on a robot hand, the calibration body moves successively to let the cameras see clearly and the following 4 images from the cameras are grabbed.



(a) C1: image (b) C2: image (c) C3: image (d) C4: image

(e) C1: pattern (f) C2: pattern (g) C3: pattern (h) C4: pattern

Figure 4.11: camera calibration in SVW project

At each position, where the camera grabs image, the pose of the calibration body is measured by a laser tracker in cabinet frame

	x	y	z	rx	ry	rz
Camera.1 / body	-318.02	-654.32	130.79	124.835	51.917	-60.713
Camera.2 / body	-174.43	528.07	236.38	-77.570	-27.438	89.099
Camera.3 / body	3052.53	-800.15	120.33	-84.533	-6.733	-117.363
Camera.4 / body	2889.96	410.93	252.00	88.648	69.975	110.471

Table 4.6: poses of the calibration body in cameras calibration

With the above described calibration procedure, the cameras are calibrated and the calibration results are listed in table 4.7. For applying the results to the vision system, a conversion as follows is necessary

1. f, S_x, S_y: f is the focal length of the camera lens and S_x, S_y represent the pixel size in millimeter on the camera chip. As discussed in the chapter **camera model**, two parameters f_x, f_y in mathematics describe well the functions of f, S_x, S_y in projection procedure. Simply taking $S_y = 0.008325$ from the camera data sheet, f, S_x are solved with equations 2.7 and 2.8.

2. C_x, C_y: the image coordinates in VMT software are with respect to the image center, the image origin from the calibration results must be corrected according to the current camera image.

3. k: the radial distortion factor can be recovered by $k = K/f^2$.

4. **Camera pose**: the camera poses from table 4.7 are with respect to the calibration body and should be converted into the world frame and the poses of the calibration body with respect to the world frame are listed in table 4.6.

After converted one by one as above described, the calibration results for a vision system are listed in table 4.8 and these results will be used in section *multi-camera system* from chapter *vision systems* for experiments.

	f_x	f_y	C_x	C_y	K	
Camera.1	4235.24	4227.56	416.04	332.77	-0.010727	
Camera.2	4417.89	4409.38	410.47	282.89	1.061292	
Camera.3	4323.44	4315.07	398.94	243.58	0.376498	
Camera.4	4233.38	4227.22	425.40	285.26	0.110802	
	x	y	z	rx	ry	rz
Camera.1	1664.88	1773.90	1865.12	171.226	50.766	35.348
Camera.2	1639.54	2632.79	1259.74	-132.715	-56.876	-172.961
Camera.3	1275.77	1341.93	1860.18	-149.759	-33.732	-177.617
Camera.4	1757.04	1255.77	2002.16	-156.141	39.317	67.727

Table 4.7: calibration results with a calibration body

	f	S_x	S_x	C_x	C_y	k
Camera.1	35.19	0.008310	0.008325	32.04	46.77	-0.000009
Camera.2	36.71	0.008309	0.008325	26.47	-3.11	0.000788
Camera.3	35.92	0.008309	0.008325	14.94	-42.42	0.000292
Camera.4	35.19	0.008313	0.008325	41.4	-0.74	0.000089
	x	y	z	rx	ry	rz
Camera.1	-1884.24	-3062.70	-938.71	-69.175	-2.986	-35.415
Camera.2	-1931.60	3070.88	-1049.34	-66.208	1.826	-144.400
Camera.3	4165.85	-2955.86	-880.70	-66.704	0.613	29.304
Camera.4	4098.95	2810.87	-952.76	-65.071	2.036	151.067

Table 4.8: values for applying in the vision system

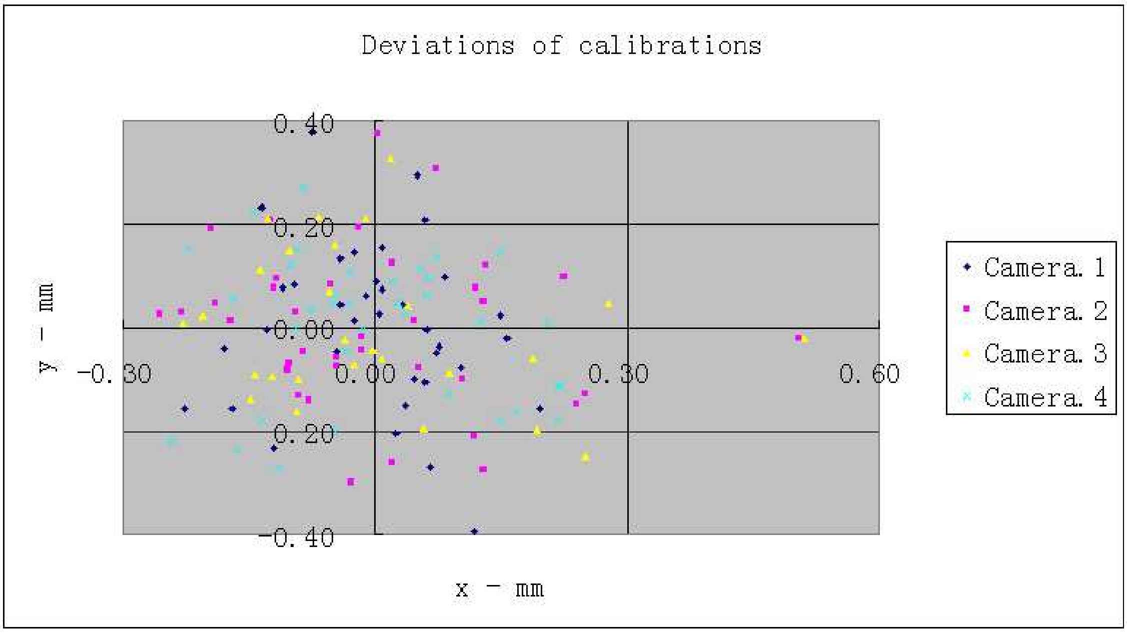


Figure 4.12: calibration deviations in SVW project

In the same way, the deviations of the calibrations are monitored and shown in figure 4.12. With contrast to the deviations in figure 3.18, we realize that the accuracy of calibration in working field is not as good as that in laboratory, and the final accuracy in measurement will be tested in the next chapter.

4.3.4 Extent of body calibration

Although the external measuring system in our projects is used passingly in camera calibration, we have tried to improve the calibration approach by dropping out the external measuring system. For instance, we have ever got such a query from a customer in China. Their application is simple, one object type, one robot without linear track and the measuring task is to guide the robot to take the object by sensing the object pose when it comes and stops before the robot. What is more, the application accuracy is 2 mm, not 1 mm, the standard accuracy for sealing applications in painting shop. Accordingly, a cheap quotation is claimed for the simple application, consequently, an economical approach for such kind of applications is necessary.

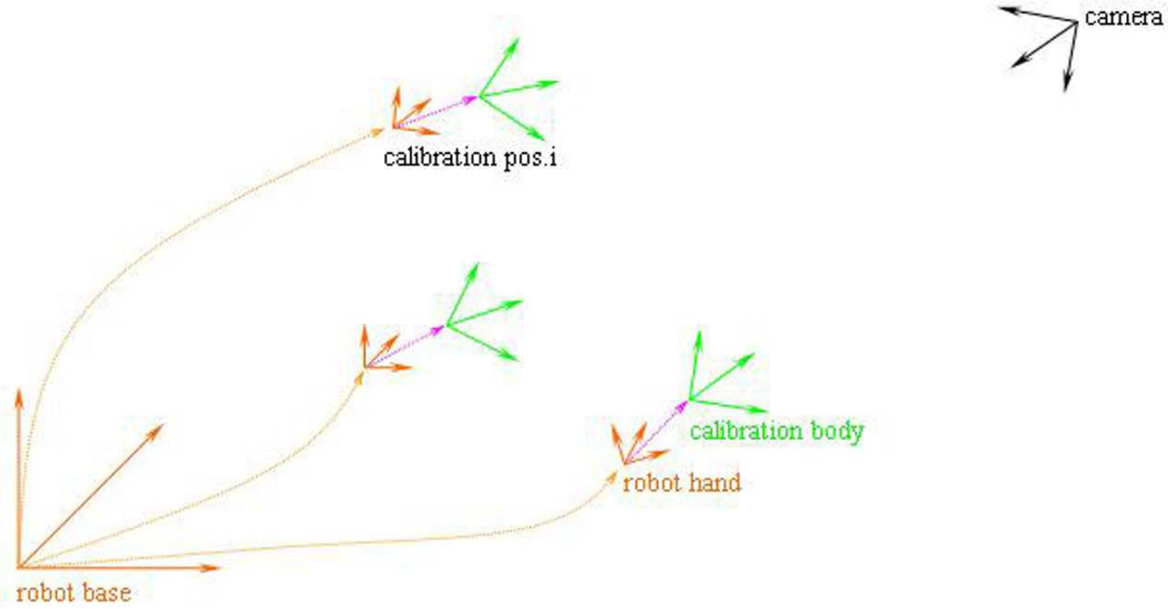


Figure 4.13: calibration with the calibration body on robot hand

For cost reduction, the startup procedure is simplified to use the robot in stead of the expensive laser tracker. The robot without linear track can be self-calibrated and the teaching pose of the object can be measured with the aid of the robot and corresponding tools. The camera calibration is solved again with the calibration body mounted on the robot hand, which is outlined in figure 4.13. Since the camera internal parameters are constant for a given camera, they can be estimated in a calibration with the calibration body in an arbitrary position. Therefore, the critical problem here is to determine the camera pose with respect to the robot base frame. Firstly, we make following notations in figure 4.13:

1. ${}^{r}T_{h}$: the transformation from the robot base frame to the robot hand frame is marked in orange in figure 4.13. It can be obtained in any time from the robot controller.

2. ${}^{h}T_{b}$: the transformation from the robot hand frame to the calibration body frame is marked in pink in figure 4.13. It is unknown but fixed once the calibration body mounted on the robot hand.

3. ${}^{b}T_{c}$: the transformation from the calibration body frame to the camera frame. It will be estimated by any camera calibration procedure discussed in the last chapter.

As seen from figure 4.13, the calibration procedure is completed by driving the robot into several positions, where the camera looks at the calibration body and gets calibrated in calibration body frame. With ${}^{r}T_{h}^{i}$ given from the robot controller and ${}^{b}T_{c}^{i}$ estimated in camera calibrations, the transformation ${}^{h}T_{b}$ from the robot hand frame to the calibration body frame is determined with the following equations

$$ {}^{r}T_{h}^{1} \bullet {}^{h}T_{b} \bullet {}^{b}T_{c}^{1} = \cdots = {}^{r}T_{h}^{i} \bullet {}^{h}T_{b} \bullet {}^{b}T_{c}^{i} = \cdots = {}^{r}T_{c} \tag{4.4} $$

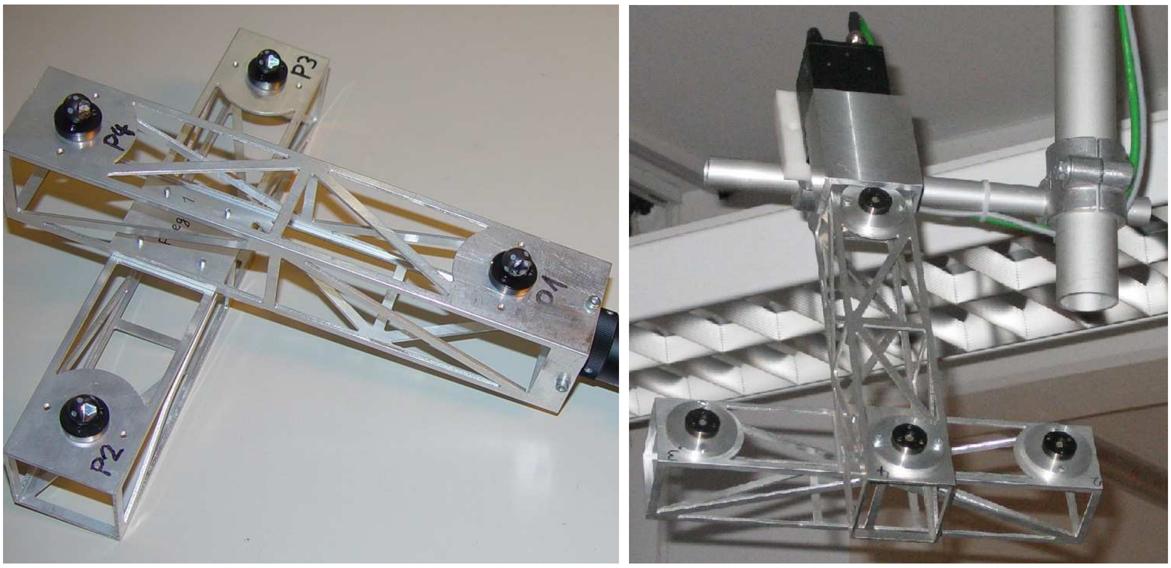
Once ${}^{h}T_{b}$ is obtained, the camera pose ${}^{r}T_{c}$ with respect to the robot base frame is also determined with above equations.

4.4 Calibration of camera pose

A camera can be calibrated easily and accurately in laboratory, but to calibrate a camera in the working area is usually an elaborate procedure. Thinking of the fact that the internal camera parameters are dependent only to the hardware, we have an idea for camera calibration in two steps: firstly calibrate the camera in laboratory and at the same time save some additional information in that procedure; in the working area, the camera pose is estimated with the determined internal parameters and the additional information. The calibration in laboratory is already well described, the main topic in this section is to determine the camera pose in the working area.

4.4.1 Pose calibration with a framework

Before the calibration approach with a calibration body is applied in the applications in 2006, VMT has used a framework to determine the camera pose in applications for almost five years. The framework and the mounting mode in calibration procedure is shown in figure 4.14.



(a) framework (b) mount mode

Figure 4.14: pose calibration with a framework

The method is only to calibrate the cameras of JAI CV-M50, on which the framework can be mounted with four different but fixed modes: upward, downward, P2 upward and P3 upward with relative to the camera. For each mode, the four reference points on the framework are accurately measured and their coordinates saved in camera frame. In working area, the framework is mounted again onto the camera with one of the four modes and the four reference points are measured in the world frame, the camera pose with respect to the world frame is determined by a *best-fit* procedure. Although this approach is easy to be carried out and has been applied to projects for many years, it has some obvious disadvantages. Firstly it is applicable only to the predefined types of cameras. The most critical disadvantage of the approach is the accuracy: the four defined mounting modes of the framework to the camera body are aligned by a metal adapter, which actually can not guarantee the alignment, especially the orientations.

4.4.2 Pose calibration with reference points

In mathematics, the calibration approach with a framework can be abstracted as follows: defining some features fixed with relative to the physical camera and determine their coordinates in camera frame in calibration procedure in laboratory, then the camera pose is estimated by measuring the features in the world frame in the working area. Going further from the approach, we can define some features directly on the camera body and the calibration for the camera pose will be completed without any tools. However, we should not forget the fact that the camera body is relatively small and the orientation of the estimated camera pose may have considerable errors. Finally, we have a new solution: *define a feature on the camera body, which is measured in camera frame in calibration procedure; in working area, two or more arbitrary reference points are observed by the camera and at the same time measured by an external measurement system, the camera pose is well estimated.*

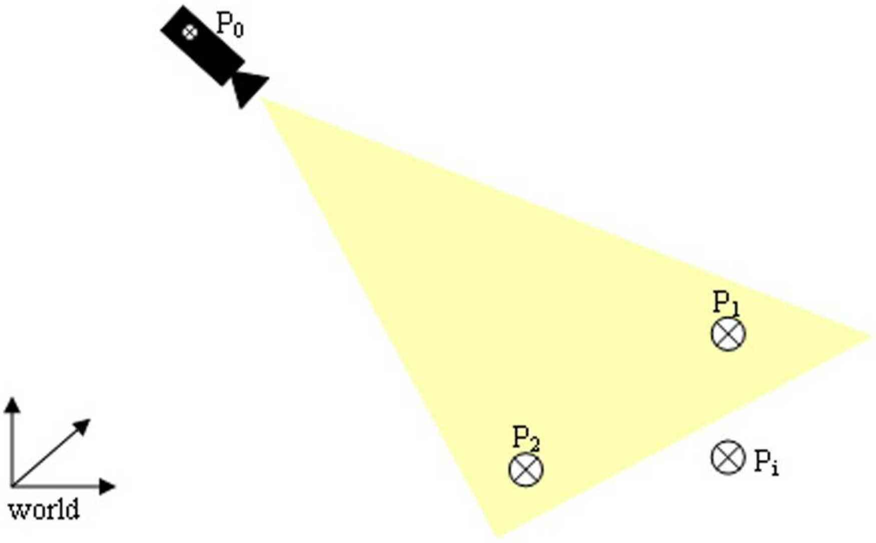


Figure 4.15: camera pose calibration in working area

As seen in figure 4.15, we denote ${}^c\vec{p}_0, {}^w\vec{p}_0$ the reference point on the camera body with respect to the camera frame and the world frame, ${}^w\vec{p}_i, \vec{P}_i$ the reference points in world frame and their corresponding images, and $R, \vec{t}$ the camera pose to be estimated, then

$$
\begin{aligned}
{}^w\vec{p}_0 &= R \cdot {}^c\vec{p}_0 + \vec{t} && (4.5)\\
{}^w\vec{p}_i &= R \cdot {}^c\vec{p}_i + \vec{t} \quad i = 1, 2, \cdots && (4.6)
\end{aligned}
$$

Since the internal camera parameters are calibrated, the coordinates of the reference points in camera frame can be represented by their image coordinates with the equation 5.1 and the above equations can be rewritten into

$$
\begin{pmatrix} {}^w x_i - {}^w x_0 \\ {}^w y_i - {}^w y_0 \\ {}^w z_i - {}^w z_0 \end{pmatrix} = {}^w\vec{p}_i - {}^w\vec{p}_0 = R \cdot ({}^c\vec{p}_i - {}^c\vec{p}_0) = R \cdot \begin{pmatrix} \delta_i X_i - {}^c x_0 \\ \delta_i Y_i - {}^c y_0 \\ \delta_i - {}^c z_0 \end{pmatrix} \qquad (4.7)
$$

Denoting the rotation matrix into RPY form, the $3 + i$ unknowns can be solved by $3i$ equations with no less than two reference points. In fact, R is an orthogonal matrix and there is a closed form solution for R. Once R is obtained, the translation is also determined by

$$\vec{t} = {}^{w}\vec{p}_0 - R \cdot {}^{c}\vec{p}_0 \tag{4.8}$$

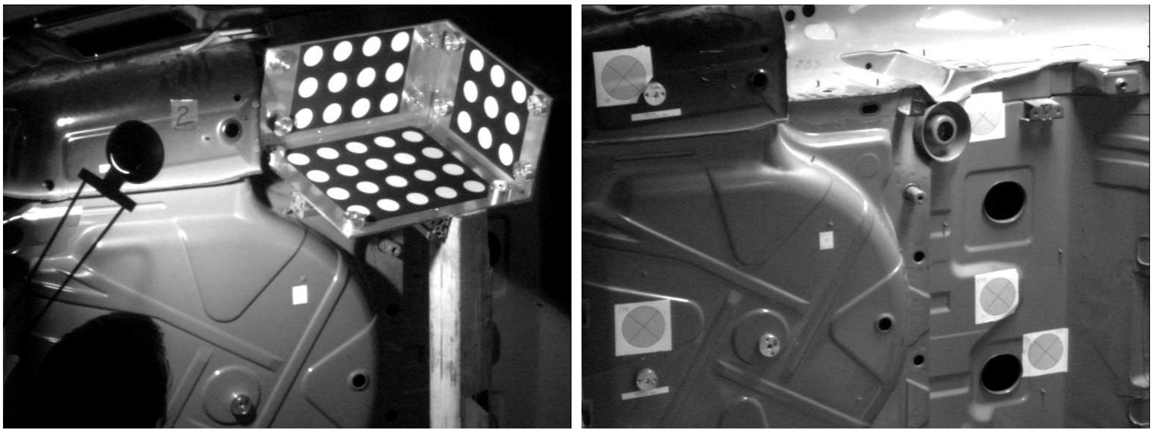
Remark: although only one reference point is needed, more reference points on the camera body are practically defined, which makes it easier for an external measurement system to see and measure one of them in the working area.

4.4.3 Experimental results

For laboratory test, a metal feature is sticked onto the body of JAI CV-M50 camera with the lens of 25 mm focal length. The calibration is carried out with a calibration body and the results together with the feature information are listed in table 4.9.

	f_x	f_y	C_x	C_y	K	
internal	3037.88	3034.23	383.67	279.93	-0.292873	
	x	y	z	rx	ry	rz
camera pose	4636.41	1632.12	-3815.94	-17.066	20.159	-179.861
feature	-30.60	21.76	-7.26	coordinates in camera frame		

Table 4.9: information for the calibration



(a) camera calibration

(b) pose estimation

Figure 4.16: simulation in laboratory

As shown in figure 4.16, four reference points are used for estimating the camera pose and their coordinates are searched manually from the camera image. The input information and the estimating results are listed in table 4.10.

Since sometimes VMT uses default values simply from camera manuals for internal camera parameters, the estimation procedure is tested with both sets of internal camera parameters. In table 4.10, **calib.A** denotes the camera pose calibrated with the internal parameters from the body calibration and **calib.B** denotes the camera pose calibrated with default internal parameters. Although this approach to camera pose calibration is still not applied in real applications, the experiment is carried out so strictly as in a working area since the setup in our laboratory is similar to a working area, which is shown in figure 4.17.

	X	Y	x	y	z
reference point.1	57	98	3688.20	678.19	39.88
reference point.2	541	150	2889.47	479.91	253.57
reference point.3	562	393	2857.98	93.34	235.85
reference point.4	84	434	3675.66	210.40	2.37

	f	S_x	S_y	C_x	C_y	k
intern.A	25.26	0.008315	0.008325	383.67	279.93	-0.000459
	x	y	z	rx	ry	rz
calib.A	4636.40	1632.10	-3815.99	-16.985	20.201	-179.798
camera pose	4636.41	1632.12	-3815.94	-17.066	20.159	-179.861
calib.B	4636.40	1632.10	-3815.95	-17.118	20.204	-179.845
	f	S_x	S_y	C_x	C_y	k
intern.B	25.00	0.008325	0.008325	384.00	286.00	0.000000

Table 4.10: Verification of the vision system

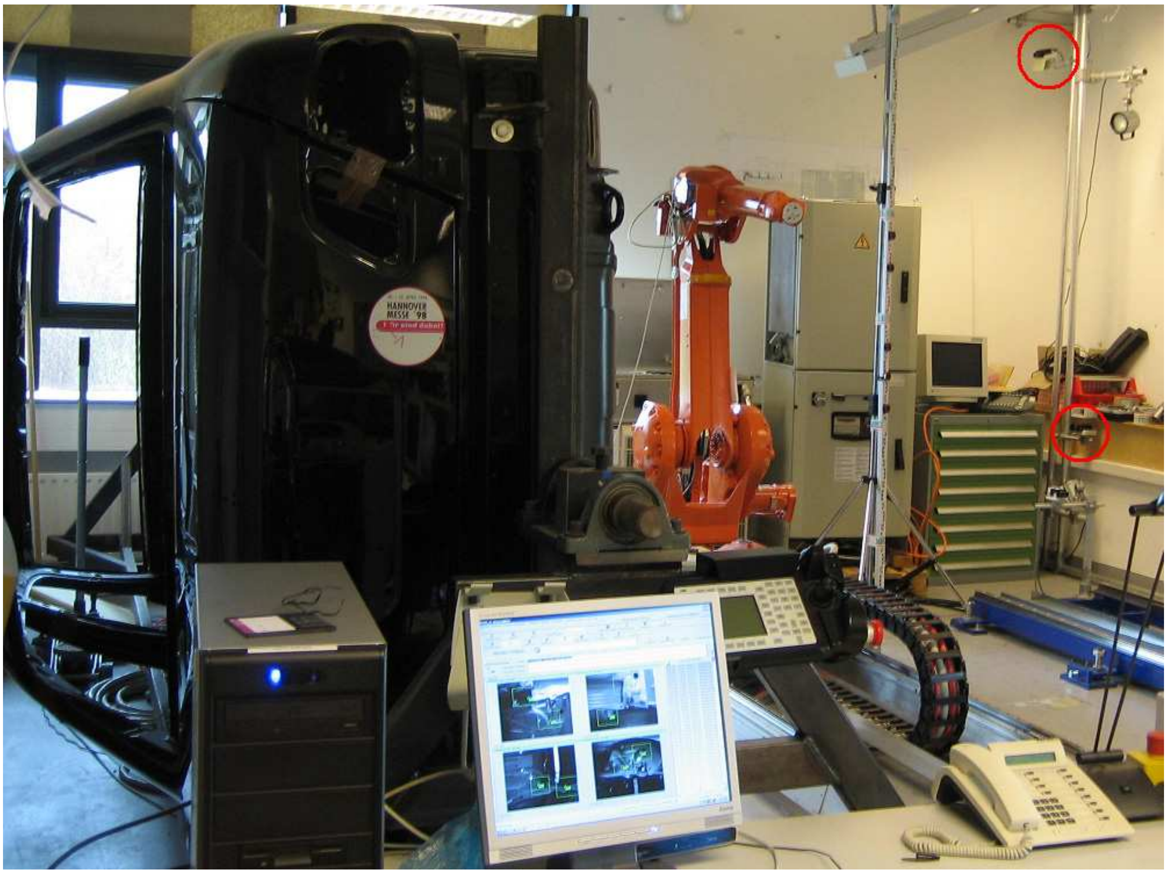


Figure 4.17: simulation environment in laboratory

4.5 Chapter review

Based on the algorithms for camera calibration as described in the last chapter, four practical approaches to camera calibration are proposed in this chapter according to the different applying situations.

4.5.1 Characters and applicable situations

The four approaches are designed for different purposes and their characters and applicable situations are as follows

1. Calibration with a 2D board
   A 2D board equipped with an accurate device which is able to move in three orthogonal directions is a very common and traditional setup for camera calibration. In our laboratory, the device is named as *microscope* with an accuracy about 0.01mm, the board is made of glass and the coordinates of the array patterns have an accuracy in 0.02mm. Thus the most important properties of the approach are the efficiency and the accuracy. Although it is designed mainly for experiments in the laboratory, our stereo sensors are calibrated with it. Firstly, the cameras of a stereo sensor are usually well calibrated in the laboratory. Secondly, a stereo sensor has to be calibrated very accurately since its uncertainty in calibration will be inherited in its measurements.

2. Calibration with robot tools
   Obviously, the applications, where this approach is applied, must have industrial robots. A communication module in the vision system that guides the robot with the robot tool to some specific positions and then the calibration procedure is done in a few minutes. This approach makes full use of the applying environment and is integrated perfectly into the vision system. However, the accuracy of the calibration results is not very good since the industrial robots have the absolute accuracies of 0.5 mm to 1 mm. Thus this approach is applied by us only for 2D pose estimation in the press sheets auto-feeding applications, where the largest sheets till to 4200 × 1900 mm may be measured and the accuracy of 2 mm is required.

3. Calibration with a 3D body
   With a 3D body, the cameras can be calibrated accurately and efficiently. The only problem is that the resulted camera pose is to the body frame, which is unknown to the world frame. This trouble may be solved by the extent mode of the body calibration. This approach is widely applied to the applications for a multi-camera system. When the cameras are located not far away from each other and can be calibrated with the 3D body mounted on the same robot, the robot base frame may be taken for the world frame and the extent mode is applied. Otherwise, an external measurement system, for example a laser tracker, is adopted to relate the cameras poses to the world frame.

4. Calibration of the camera pose
   The camera calibration is completed in two steps: calibrate the internal camera parameters in the laboratory and determine the camera pose on site. In the former procedure, the internal parameters are accurately calibrated and at the same time

some fixed points on the camera body are well defined. The latter procedure is done with a fixed point on the camera body and some arbitrary reference points in the camera sight. The advantage is that the whole procedure is easily carried out and the results are relatively accurate, and the disadvantage is that an external measurement system is necessary.

Remark: In fact, the calibration with a simpler 2D board is a popular and efficient approach for camera calibration on site. However, it is protected by the patent registered by a competitor of VMT GmbH. We cannot apply it to the vision applications and that is why we haven't referred in this dissertation.

4.5.2 Contributions

Since the approaches are designed for some specific purposes, they are able to be carried out on site and show us their practical values. The main contributions of this chapter can be concluded as follows

1. In the calibration with a 2D board in the laboratory, the high accuracy of the microscope and the coordinates on the 2D board reduces furthest the external errors in the calibration procedure. Thus it is a perfect workbench for testing all kinds of calibration algorithms.

2. The robot tool is introduced to determine the coordinates of the calibration ball and the calibration of the robot tool is solved both in mathematics and practice.

3. Through integrating the calibration module into the vision system, the approach with a robot tool is a complete procedure, which is able to be carried out quickly and automatically.

4. With a ready 3D body and an external measurement system, the cameras can be calibrated accurately and quickly on site.

5. The extent mode of the calibration with a 3D body makes it possible to calibrate a camera completely without any external measurement systems.

6. To the applications in automotive industry, the car bodies and robots must be initialized as well as the vision system and an external measurement system is needed in most of cases. Thus the approach to determining the camera pose on site has its practical values.

7. Although there are no smart setups or amazing ideas, the approaches are built on our thoughts and have gone through the trial in the industrial applications.

Chapter 5

Vision Systems Applied to Robot Vision Applications

A vision system with one or more cameras is usually designed for the specific vision applications, such as inspections, measurements, pose estimation and so on. The dissertation aims at the industrial applications in robot vision, the vision systems referred in this chapter are specially for pose estimation, which requires the cameras from the vision systems to be calibrated. The calibration issue is discussed in the last chapters, thus the cameras referred in this chapter are already calibrated.

5.1 Direction vector

For a calibrated camera, all the camera parameters, the projection matrix A and the camera pose $[R|\vec{t}]$, are known. Without loss of generality, the camera frame is assumed to be the world frame. Either from 2.13 for pinhole cameras or from 2.19 and 2.20 for a camera model with distortion, one can obtain the following equation

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \delta \begin{pmatrix} f(X, A, k) \\ g(Y, A, k) \\ 1 \end{pmatrix} = \delta \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} \tag{5.1}$$

where f and g are two functions defined by image coordinates and camera internal parameters; X and Y are used to denote the two functions f and g respectively by abuse of notation; δ is actually the distance from the object point to the lens plane. Since the distance δ cannot be obtained from any image, a single camera can not determine the 3D coordinates of any object point without other additional information. It is not difficult to understand geometrically: a 2D pixel can only define a **direction vector** in the camera frame, or determine a 3D point up to a certain scale factor. Generally speaking, equation 5.1 is the ground stone to all algorithms for coordinates determination or pose estimation.

5.2 Mono-camera system

Just as its name applied, a mono-camera system is such a setup that is configured with only one camera but it can do measurements. For convenience of contrasting to the following vision systems, it is called in this dissertation a mono-camera system. In this section, it will be described what a mono-camera system can measure and how it measures.

5.2.1 Measuring task

5.1 says clearly that a single camera can only measure an arbitrary single object point when its distance to the camera lens plane is known. Fortunately, the applications in automobile industry are usually not for measuring arbitrary object points, but for rigid work objects. A typical application of pose estimation is as follows

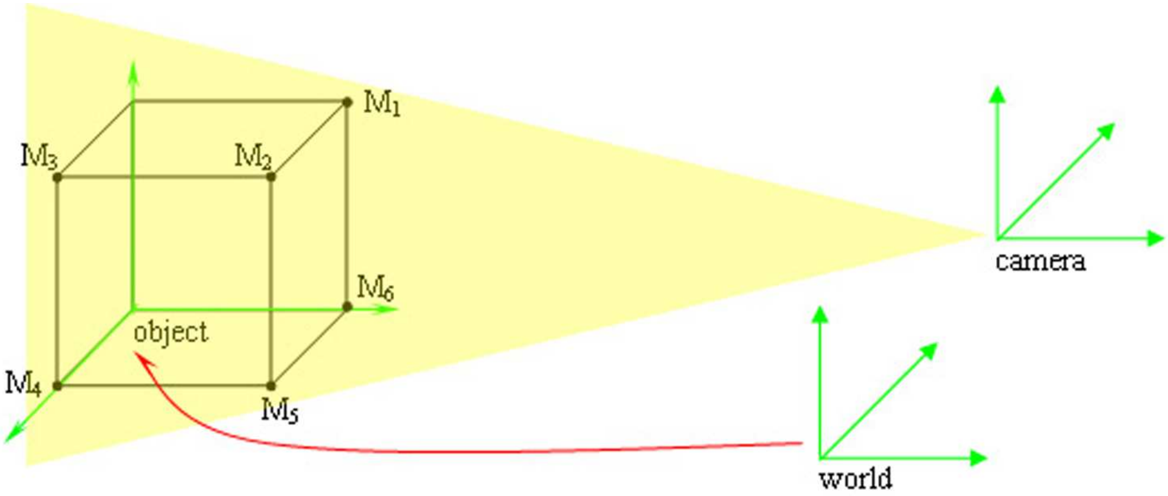


Figure 5.1: measuring with a single camera

As shown in figure 5.1, there are n ($n > 3$) marks on the object and their coordinates in the object frame are known. The object pose, the transformation from the world frame to the object frame, is what we want.

5.2.2 Coordinates estimation

Let's denote N nonlinear object points with known coordinates in the object frame as follows

$$^o\vec{p}_i = (^o x_i, ^o y_i, ^o z_i)^T \quad i = 1, 2, \cdots, N \tag{5.2}$$

Thinking of the fact that the distances between any two object points are independent to the reference frame, the distance between two arbitrary points $^o\vec{p}_i$ and $^o\vec{p}_j$ can be obtained by

$$d_{ij}^2 = (^o x_i - ^o x_i)^2 + (^o y_i - ^o y_i)^2 + (^o z_i - ^o z_i)^2 \tag{5.3}$$

As mentioned above, the coordinates of the object points in the camera frame can be denoted by the direction vectors with a calibrated camera

$$\begin{pmatrix} {}^c x_i \\ {}^c y_i \\ {}^c z_i \end{pmatrix} = \delta_i \begin{pmatrix} X_i \\ Y_i \\ 1 \end{pmatrix} \qquad (i = 1, 2, \cdots, N) \tag{5.4}$$

With the fact that the distances between the points are kept unchanged whichever frame the coordinates are with respect to, one has

$$(X_i\delta_i - X_j\delta_j)^2 + (Y_i\delta_i - Y_j\delta_j)^2 + (\delta_i - \delta_j)^2 = d_{ij}^2 \tag{5.5}$$

where X_i, Y_i and X_j, Y_j can be obtained from image coordinates X, Y and camera internal parameters, d_{ij} is also known with the point coordinates in the object frame, only δ_i, δ_j are variables. With N object points, one will get a nonlinear overdetermined system with N variables and $N(N-1)/2$ relations as above. The system can be solved with the Newton-Rapson method, which is explained in appendix, if the relations are no less than the variables, namely

$$\frac{N(N-1)}{2} \geq N \quad \Longrightarrow \quad N \geq 3 \tag{5.6}$$

It seems that with 3 or more known points, one will get a solvable overdetermined nonlinear system, which can be solved if a good set of initial values for δ_i are given. Once the δ_i is obtained, the coordinates of the points in the world frame can be easily determined

$${}^w\vec{p_i} = [R|\vec{t}] \cdot^c \vec{p_i} = \delta_i \cdot [R|\vec{t}] \cdot \begin{pmatrix} X_i \\ Y_i \\ 1 \end{pmatrix} \tag{5.7}$$

5.2.3 Initial guess for δ_i

In fact, the scale factor δ is the z-coordinate of the object point with respect to the camera frame, or can be approximately understood in geometry as the distance from the object point to the camera optical center. Normally for measuring with a mono-camera, the measuring distance is relatively small. That makes the mono-camera have a small scene deepth, only the objects within which can be seen clearly. Select two object points ${}^o\vec{p_i}$ and ${}^o\vec{p_i}$, which satisfy $d_{ij} \longrightarrow min$, and assume that $\delta_i = \delta_j$ and it yields

$$\delta_i = \delta_j = \frac{d_{ij}}{\sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2}} \tag{5.8}$$

Having Solved δ_i and δ_j, all scale factors can be obtained one after another by relating to δ_i or δ_j. With these initial guesses, the nonlinear system can be solved for all scale factors δ_i.

5.2.4 Pose estimation

With the above steps, the coordinates of the points in the camera frame are approximately estimated as $({}^c x_i, {}^c y_i, {}^c z_i)$. The coordinates of the points in the object frame are known, and yield

$$\begin{pmatrix} {}^{c}x_i \\ {}^{c}y_i \\ {}^{c}z_i \end{pmatrix} =^{c} T_o \cdot \begin{pmatrix} {}^{o}x_i \\ {}^{o}y_i \\ {}^{o}z_i \end{pmatrix} \tag{5.9}$$

With no less than three nonlinear points, the transformation ${}^{c}T_o$ can be estimated by a best-fit procedure, which is shown in the appendixes. Since the camera is calibrated, the camera pose with respect to the world frame must be known

$${}^{w}T_o =^{w} T_c \cdot^{c} T_o \tag{5.10}$$

5.2.5 Improvement of pose estimation

In the above calculation, we have thought only the distances between object points. However, what keeps unchanged to all coordinate frames is not only the distances, but also the relative positions of the points to each other. Thus the object pose can be directly estimated as follows:
Let's think in the camera frame and the transformation from camera to object is denoted as

$${}^{c}T_o = [R|\vec{t}] = \{\alpha, \beta, \gamma, x, y, z\} \tag{5.11}$$

then the coordinates of the points in the camera frame are as follows

$$\delta_i \begin{pmatrix} X_i \\ Y_i \\ 1 \end{pmatrix} =^{c} \vec{p}_i = [R|\vec{t}] \cdot^{o} \vec{p}_i \tag{5.12}$$

Removing the scale factor δ and writing the rotation into RPY, the above equation can be rearranged into the following two equations

$$F_x(\alpha, \beta, \gamma, x, y, z) = 0 \tag{5.13}$$

$$F_y(\alpha, \beta, \gamma, x, y, z) = 0 \tag{5.14}$$

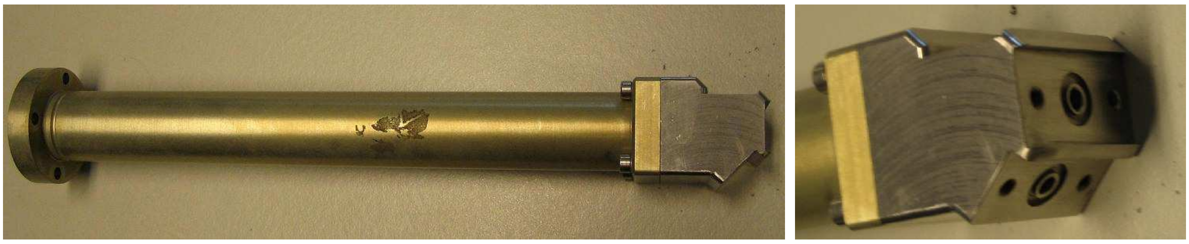
With $N \geq 3$ object points, one will get an overdetermined nonlinear system to be solved with a good set of initial guess, which can be obtained from the results of points measuring.

5.2.6 Application in automotive industry

Hereby a vision solution with a mono-camera is proposed for verifying the painting gun used in sealing applications.

Destination of the application

To let the robot do its applications at the right position, one of the preconditions is that the painting gun, which is shown in the below figure, is undamaged and its mounting situation is unchanged. As seen in figure 5.2, each nozzle has its pose and coordinates with respect to the robot hand. In fact, a coordinate frame on the nozzle is defined and the transformation between the nozzle frame and the robot hand frame will be determined in the tool calibration procedure. Once the painting gun is fixed on the robot hand and the robot application program is taught, the referred transformation must be kept absolutely unchanged. Otherwise, the robot application windage may be out of tolerance however accurate the work object is measured. In practice, some cases such as slight collision of the painting gun against other fixings, take-down and remounting the painting gun may happen. *The mono-camera system is applied to verify if the nozzle frames are changed with respect to the robot hand frame in necessary cases.*

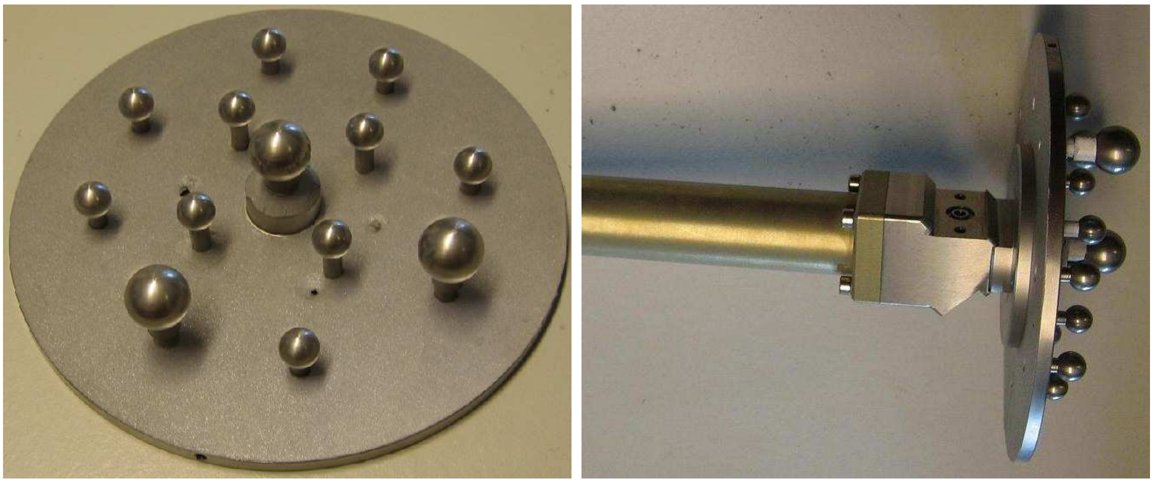


(a) overview (b) detail

Figure 5.2: a painting gun for sealing applications

Applied tool

Since the painting gun may be damaged, the mono-camera system of course can not measure it directly. Therefore, a specially designed tool is introduced as follows



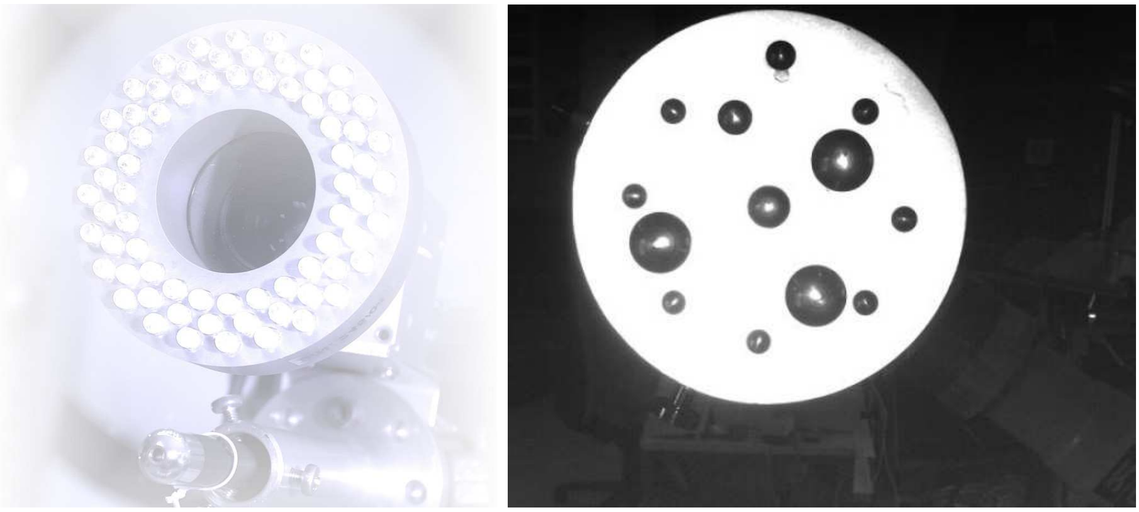
(a) verification tool (b) mounting mode

Figure 5.3: verification of a painting gun

As seen in figure 5.3, the balls with different sizes and heights can be freely screwed onto the calibration board in a free array. Once the balls or marks are fixed, a laser tracker system, as described in the appendix, will be used for setting up a tool frame on the board and determining the 3D coordinates of all marks with respect to the tool frame.

Infrared light for pattern recognition

For any vision system, the accuracy errors are mainly from two directions: camera calibration and pattern recognition. The first issue is discussed much more in detail in the last chapter. Hereby we focus on improving the accuracy of the pattern recognition. To get better accuracy in pattern recognition, the basic and critical factor is to get images with high quality. In this application, the infrared light is used as seen from figure 5.4. Many infrared light LEDs are fixed around the camera lens, the tool background is completely covered by a layer of infrared-light sensitive materials and the marks on the tool are naked steel balls. At the same time, the aperture of the camera is set so small that the camera can see almost nothing in normal environment. Once the infrared light is on, the tool will always have a perfect image: black rounds with clear contours on a totally white background, which makes it easy and accurate to search the circles and their center points.



(a) camera with infrared light

(b) tool under infrared light

Figure 5.4: verification setup with infrared light

Verification procedure in practice

The whole procedure of the application can be summarized as follows:

1. Prepare the camera in laboratory: calibrate the camera, only the internal camera parameters are neccessary, and equipe the infrared light around the camera lens;

2. Prepare the verification tool in laboratory: tightly screw the steel marks, define a tool frame and measure the 3D coordinates of all marks with respect to the tool frame; stick the infrared-light sensitive materials;

3. Prepare at the working area: mount the camera fixedly in a safety corner, and teach the application robot with the painting gun such a pose, which is normally called zero position, that the camera can see clearly the verification tool when it is mounted on the painting gun;

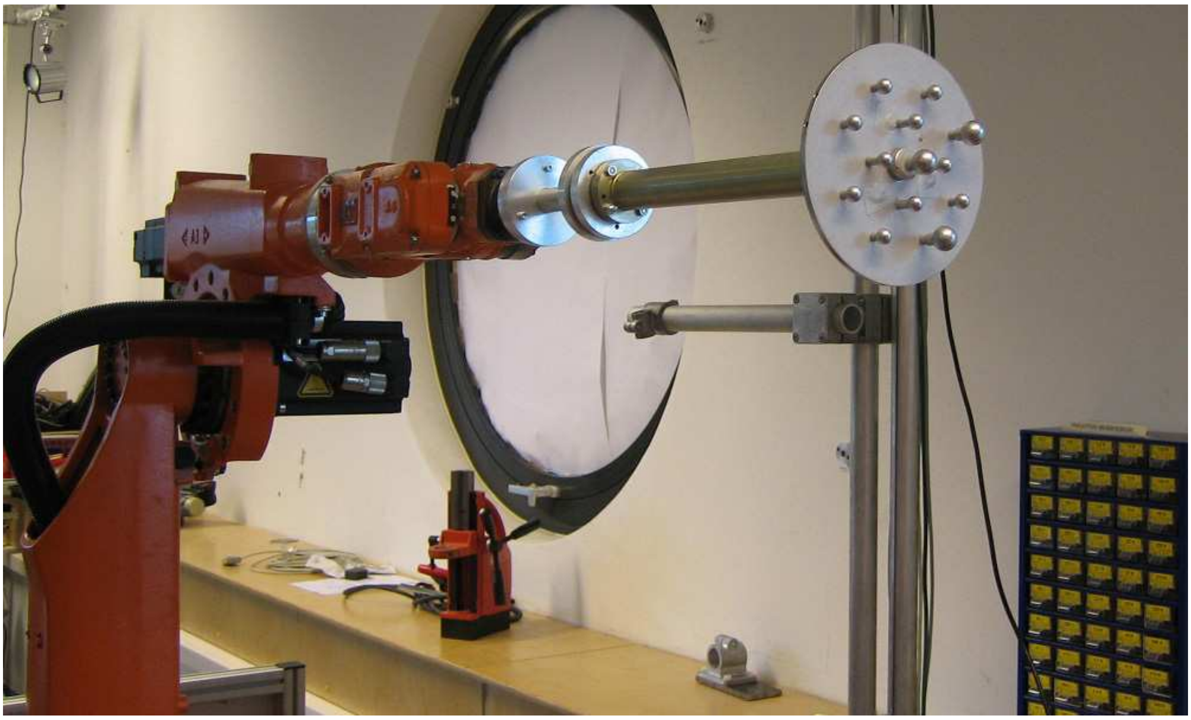


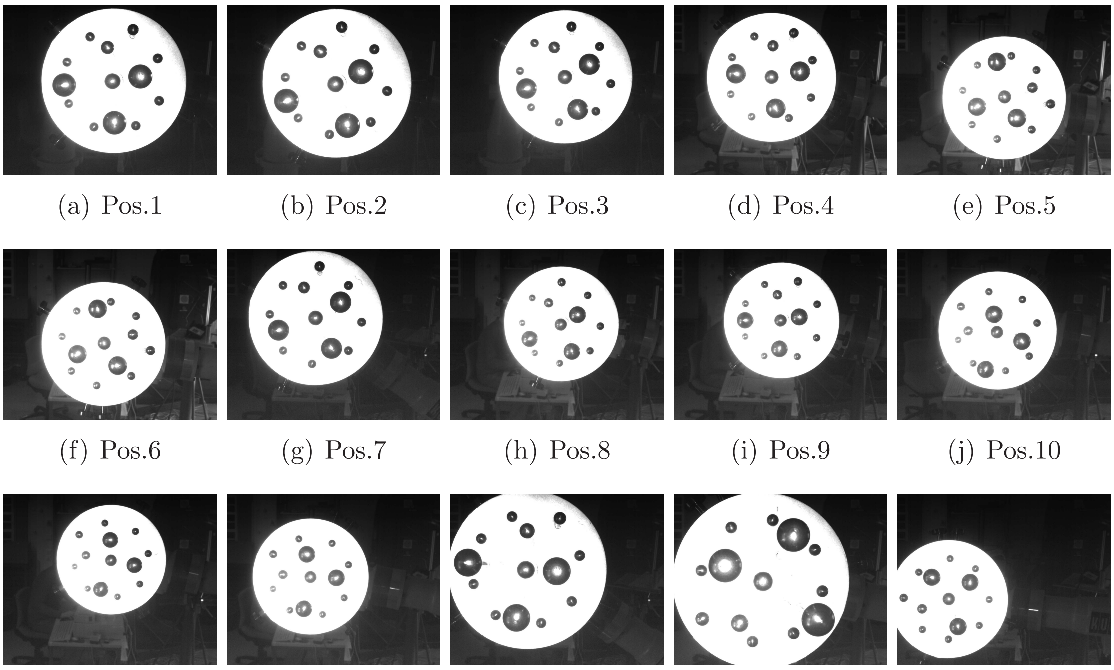
Figure 5.5: verification setup in laboratory

4. Initialization of the setup: mount the tool on the painting gun, drive the robot to the zero position, measure the tool frame with the mono-camera and save the zero pose.

5. Verification routine: mount the tool on the painting gun exactly as that in the initialization, measure the tool frame again with the mono-camera, calculate the relative transformation from the zero pose to the current pose, contrast the relative transformation to the permitted tolerance and execute the correct reactions.

Experimental results

In our experiments in laboratory, a fire-wire camera with a 25mm lens is used and the image resolution is 1280 × 1024 pixels. The measurement distance is from 150mm to 300mm. The camera is fixed and the tool is mounted on a KUKA robot hand. Through driving the robot, the tool is searched with 15 different poses, which are measured with the above described algorithm by the mono-camera. Meanwhile, all poses are measured accurately with a laser tracker system, which is called Leica system and is described in detail in the appendixes. To simulate the real application, the relative poses are calculated and the differences are shown in table 5.4.



(a) Pos.1 (b) Pos.2 (c) Pos.3 (d) Pos.4 (e) Pos.5

(f) Pos.6 (g) Pos.7 (h) Pos.8 (i) Pos.9 (j) Pos.10

(k) Pos.11 (l) Pos.12 (m) Pos.13 (n) Pos.14 (o) Pos.15

Figure 5.6: verification with the tool in 15 different poses

	x	y	z	rx	ry	rz
Pos.01	32.38	7.68	228.99	177.2596	3.6122	-61.2130
Pos.02	32.48	5.65	218.84	177.1109	3.5128	-76.2186
Pos.03	26.42	9.65	261.13	176.9395	3.3923	-75.8091
Pos.04	19.63	12.90	275.15	178.0261	-0.1283	-56.7000
Pos.05	23.44	5.69	294.22	-177.1840	0.9680	15.8600
Pos.06	19.78	8.01	292.76	-177.3882	-0.4669	15.8520
Pos.07	22.24	11.89	259.35	176.7931	-1.2952	-78.9048
Pos.08	36.27	3.79	327.39	176.2613	3.3615	-81.5008
Pos.09	34.61	0.12	323.68	177.0100	3.6773	-58.2138
Pos.10	30.31	-11.76	312.96	175.5897	1.0867	-35.9412
Pos.11	22.88	1.15	350.33	174.9833	1.8785	-133.8089
Pos.12	20.75	4.77	322.77	177.9787	-2.2794	-141.3489
Pos.13	33.09	6.85	196.92	177.5607	-1.1537	-49.0101
Pos.14	26.43	8.68	163.27	179.1989	-4.4035	155.7672
Pos.15	13.15	16.61	318.92	2.9885	-172.4454	7.1068

Table 5.1: poses measured by a mono-camera system

	x	y	z	rx	ry	rz
Pos.01	0	0	0	0	0	0
Pos.02	2.78	-5.57	10.07	0.1319	-0.1204	14.9962
Pos.03	10.03	-6.73	-33.70	0.2915	-0.2558	14.5759
Pos.04	-0.06	10.31	-45.24	2.7081	2.6925	-4.4261
Pos.05	-9.90	27.32	-62.52	-6.0618	1.0340	-76.9055
Pos.06	-13.72	30.15	-60.04	-6.2620	2.4625	-77.0527
Pos.07	5.95	10.09	-30.23	4.9055	0.4864	17.7031
Pos.08	3.96	-4.42	-98.97	0.3957	-0.9483	20.2237
Pos.09	-0.78	9.73	-94.84	0.0759	-0.2460	-3.0153
Pos.10	14.37	-1.93	-85.00	0.5592	-2.9748	74.6452
Pos.11	33.69	1.51	-122.04	-0.3207	-2.8412	72.4948
Pos.12	13.97	13.59	-93.90	4.2486	-4.1483	79.9903
Pos.13	-10.17	21.76	34.28	3.4040	3.3512	-12.0969
Pos.14	42.75	16.19	63.91	-1.5292	-8.1045	143.1147
Pos.15	-5.11	40.65	-86.71	7.1667	-10.3233	110.8338

Table 5.2: relative poses to the first position

	x	y	z	rx	ry	rz
Pos.01	0	0	0	0	0	0
Pos.02	2.87	-5.59	10.50	0.0895	-0.1174	15.1315
Pos.03	10.12	-6.80	-34.01	0.2958	-0.2207	14.7275
Pos.04	-0.02	10.31	-45.59	2.7163	2.6851	-4.5521
Pos.05	-9.89	27.33	-62.94	-6.0235	1.0679	-77.0109
Pos.06	-13.75	30.19	-60.27	-6.2239	2.4250	-76.9054
Pos.07	5.90	10.08	-30.56	4.9110	0.4728	17.8063
Pos.08	3.92	-4.44	-98.65	0.3914	-0.9965	20.3475
Pos.09	-0.75	9.76	-94.97	0.0519	-0.2920	-3.1306
Pos.10	14.34	-1.97	-84.70	0.5185	-2.9495	74.7904
Pos.11	33.68	1.52	-122.48	-0.3413	-2.8824	72.3896
Pos.12	13.94	13.57	-94.31	4.2973	-4.1966	79.8806
Pos.13	-10.14	21.72	33.85	3.4079	3.3023	-11.9938
Pos.14	42.79	16.18	64.22	-1.5583	-8.1090	143.2293
Pos.15	-5.11	40.70	-86.82	7.1334	-10.3466	110.9677

Table 5.3: relative poses measured by a laser tracker

	x	y	z	rx	ry	rz
pos.2	0.090	-0.019	0.430	-0.1024	0.1353	0.0030
pos.3	0.093	-0.067	-0.310	0.1043	0.1517	0.0352
pos.4	0.038	0.007	-0.340	0.1081	-0.1261	-0.0075
pos.5	0.005	0.017	-0.420	0.1032	-0.1055	0.0340
pos.6	-0.025	0.045	-0.240	0.1080	0.1473	-0.0375
pos.7	-0.049	-0.010	-0.330	0.1055	0.1031	-0.0136
pos.8	-0.040	-0.018	0.330	-0.1043	0.1237	-0.0483
pos.9	0.023	0.032	-0.140	-0.1241	-0.1153	-0.0460
pos.10	-0.030	-0.034	0.300	-0.0808	0.1452	0.0252
pos.11	-0.011	0.009	-0.440	-0.0907	-0.1052	-0.0412
pos.12	-0.029	-0.013	-0.400	0.0986	-0.1097	-0.0483
pos.13	0.032	-0.039	-0.420	0.1040	0.1031	-0.0488
pos.14	0.047	-0.010	0.310	-0.1292	0.1147	-0.0045
pos.15	-0.007	0.048	-0.110	-0.0830	0.1338	-0.0233
Aver	0.010	-0.004	-0.127	0.0012	0.0426	-0.0158
MaxErr	0.083	0.063	0.557	0.1304	0.1687	0.0510

Table 5.4: differences between the measurements

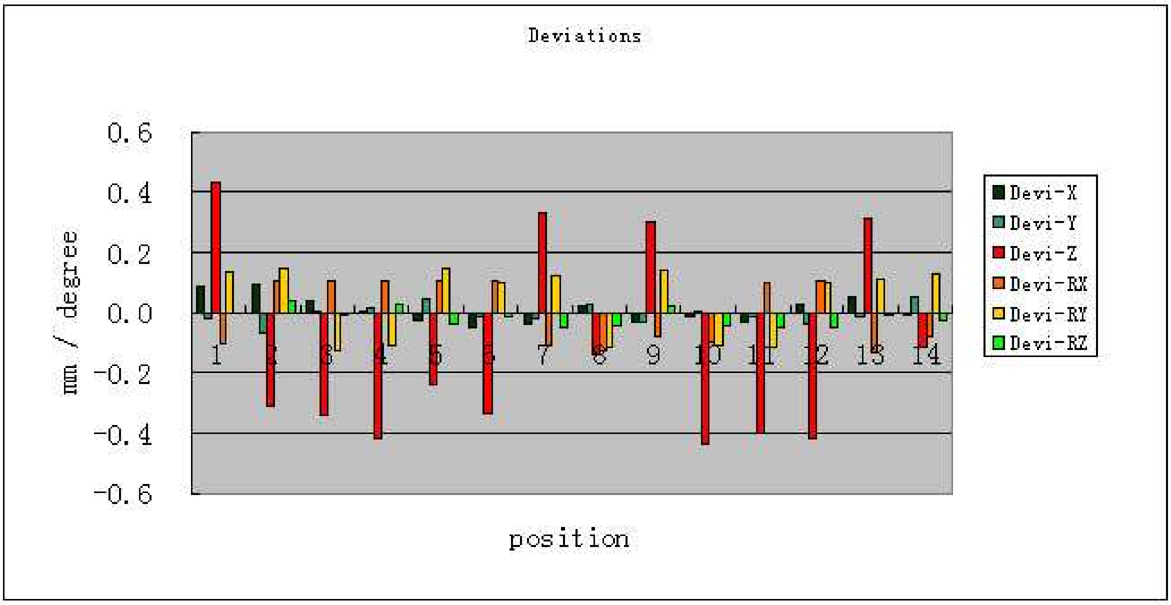
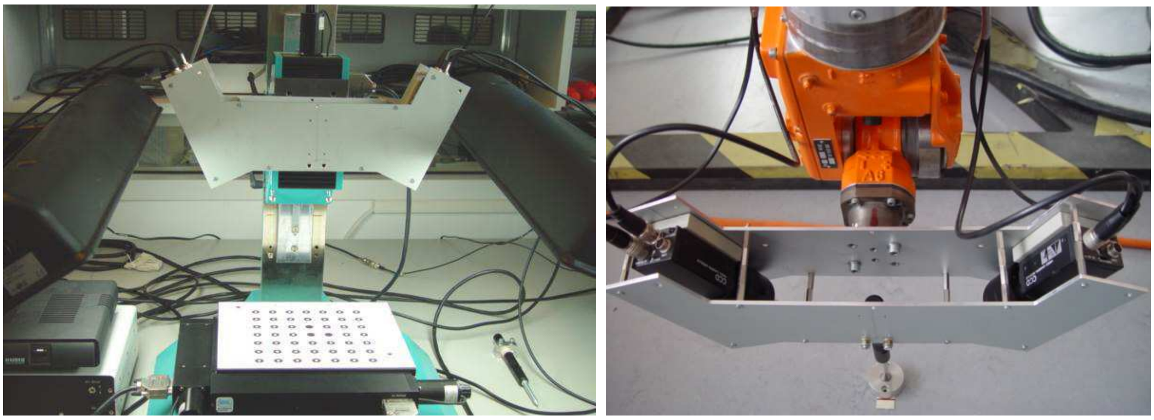


Figure 5.7: deviations of the verification

From table 5.4 or figure 5.7, careful readers may find that x, y and rz elements are very well estimated, but the results for z, rx and ry, especially z, are not so accurate. This situation is reasonable to a mono-camera system: a mono-camera can see the world only from one direction, the optical axis, which is almost parallel to the z-axis of the tool frame in our application. This fact results that any small movement in XOY plane is reflected directly and differences in z-axis shrink on the camera image. The movement or the difference is more parallel to the optical axis, the shrink is greater. Therefore, the mono-camera system gets the worst accuracy in z estimation.

5.3 Stereo vision system

A stereo vision system is composed of two cameras, whose relative poses are fixed to each other, and they have a common sight field. With the 2D information from both cameras, it is possible to rebuild the real world scene in the common sight field. The most common task of the stereo vision is to correlate the two camera images and find the corresponding pairs for obtaining the depth information of the scene. Since our focus is on pose estimation, a few special features are used for 3D coordinates estimating, or consequently the pose of the work object.



(a) calibration setup (b) sensor inside

Figure 5.8: configure of a stereo vision

5.3.1 Point coordinates estimation

Since the both calibrated cameras have different poses, the coordinates of any object point in the common sight field of them can be estimated. Let's denote as follows

$$\begin{array}{rll} \mathit{cameras\ poses}: & {}^{w}T_i = {}^{w}R_i + \vec{t}_i & i = 1,2 \\ \mathit{direction\ vectors}: & {}^{c}\vec{p}_i = \delta_i \begin{pmatrix} X_i \\ Y_i \\ 1 \end{pmatrix} & i = 1,2 \\ \mathit{wanted\ object\ point}: & {}^{w}\vec{p} = {}^{w}\vec{p}_1 = {}^{w}\vec{p}_2 & \end{array}$$

Relating the 2D - 3D coordinates, one has

$$ {}^{w}R_1 \cdot \delta_1 \begin{pmatrix} X_1 \\ Y_1 \\ 1 \end{pmatrix} + {}^{w}\vec{t}_1 = {}^{w}\vec{p} = {}^{w}R_2 \cdot \delta_2 \begin{pmatrix} X_2 \\ Y_2 \\ 1 \end{pmatrix} + {}^{w}\vec{t}_2 $$

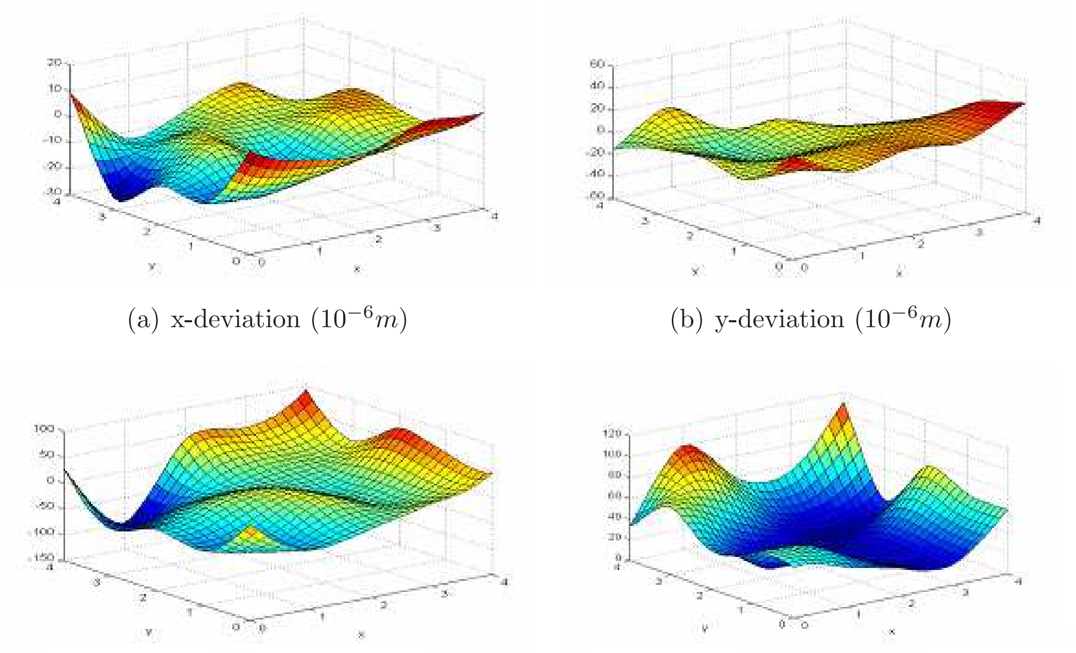
$$ \Longrightarrow \qquad \delta_1 \cdot {}^{w}R_1 \cdot \begin{pmatrix} X_1 \\ Y_1 \\ 1 \end{pmatrix} - \delta_2 \cdot {}^{w}R_2 \cdot \begin{pmatrix} X_2 \\ Y_2 \\ 1 \end{pmatrix} = \triangle\vec{t} $$

Since there are only two unknowns δ_1 and δ_2, the above equations can be easily solved and the coordinates of the object point are obtained

$$^{w}\vec{p} = {}^{w}\vec{p_i} = {}^{w}R_i \cdot \delta_i \begin{pmatrix} X_i \\ Y_i \\ 1 \end{pmatrix} + {}^{w}\vec{t_i} \qquad i = 1, 2$$

Experimental results

As described before, the calibration setup of the stereo vision can move accurately in three orthogonal directions. Thus the device frame is used as our world frame in experiments and one can also verify the accuracy of the stereo vision in coordinates estimation. The glass calibration board with 5 × 5 marks is used and one of the calibration position is selected for the estimation and the deviations are as follows



(a) x-deviation ($10^{-6}m$)

(b) y-deviation ($10^{-6}m$)

(c) z-deviation ($10^{-6}m$)

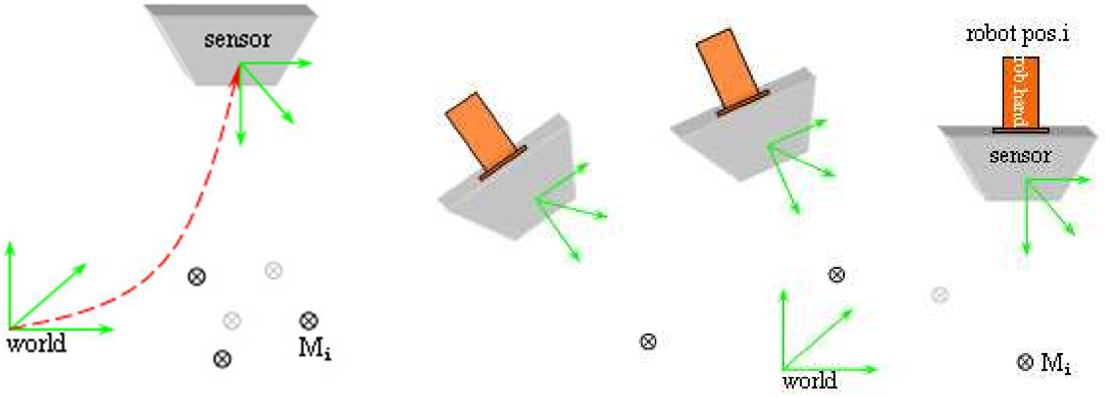
(d) vector-deviation ($10^{-6}m$)

Figure 5.9: measurement deviations from the stereo vision

Seen from the above figure, the deviations in x and y directions are very small, within 0.05mm and it is much worse in z direction, about 0.1mm. To get such results, the measure distance must be relatively small, from about 200mm to 300mm.

5.3.2 Stereo sensor calibration

As seen from figure 5.8, once the stereo vision system is calibrated, both cameras must be fixed relatively to each other and form a rigid body, which is also called stereo sensor. A stereo sensor has its own coordinate frame, in which its both cameras are calibrated. For a product of a stereo vision system, it is most of time delivered in a form of *black box* package. The user can simply do the measuring work in the sensor frame. However, every application has its own user defined world coordinate frame, which is usually different from the sensor frame. Only with the transformation between the sensor frame and the world frame, the user can get the expected results in the world frame.



(a) stationary sensor

(b) robot hand sensor

Figure 5.10: calibration of the stereo sensors

The procedure of determining the transformation ${}^wT_s$ is called sensor calibration. To carry out the calibration, some known object points ${}^w\vec{p}_i$ in the world frame are needed. Since the cameras are calibrated in the sensor frame, the coordinates ${}^s\vec{p}_i$ of the known object points with respect to the sensor frame can be estimated, then

$$ {}^w\vec{p}_i = {}^wT_s \cdot^s \vec{p}_i, \qquad i = 1, 2, 3, \cdots \tag{5.15}$$

For no less than three nonlinear known object points, the transformation ${}^wT_s$ can be determined. However, one of the most common applications of a stereo vision is to mount the sensor on a robot, as shown in figure 5.8. In this case, the sensor calibration is to determine the transformation ${}^hT_s$ between the sensor frame and the robot hand frame. But the calibration procedure is similar:

1. Select no less than three nonlinear known object points ${}^w\vec{p}_i$.

2. Drive the robot to such positions that the stereo sensor can see the known points one by one. For each position, the robot knows its hand frame with respect to the world frame and denoted as ${}^wT_h$. The coordinates of the known point with respect to the robot hand frame are obtained by ${}^h\vec{p}_i = {}^hT_w \cdot {}^w\vec{p}_i$.

3. At the same time, the known points are measured by the stereo sensor and denoted respectively as ${}^s\vec{p}_i$.

4. Relate the coordinates by ${}^hT_w \cdot {}^w\vec{p}_i = {}^h\vec{p}_i = {}^hT_s \cdot {}^s\vec{p}_i$ and solve the calibration.

5.3.3 Pose estimation of known object

For a stereo vision, the pose estimation is implemented through estimating the coordinates of individual object points. Selecting no less than three nonlinear known object points with respect to the object frame, their coordinates ${}^s\vec{p}_i$ with respect to the sensor frame can be determined by a stationary sensor or a mobile sensor. Since the sensor is calibrated, the coordinates of the object points with respect to the world frame are obtained

$$^{w}\vec{p}_i = {}^{w}T_s \cdot {}^{s}\vec{p}_i \qquad i = 1, 2, 3, \cdots \tag{5.16}$$

Since the object is known, the coordinates $^{o}\vec{p}_i$ of the object points with respect to the object frame are known. The object pose $^{w}T_o$ can be determined from the following relations

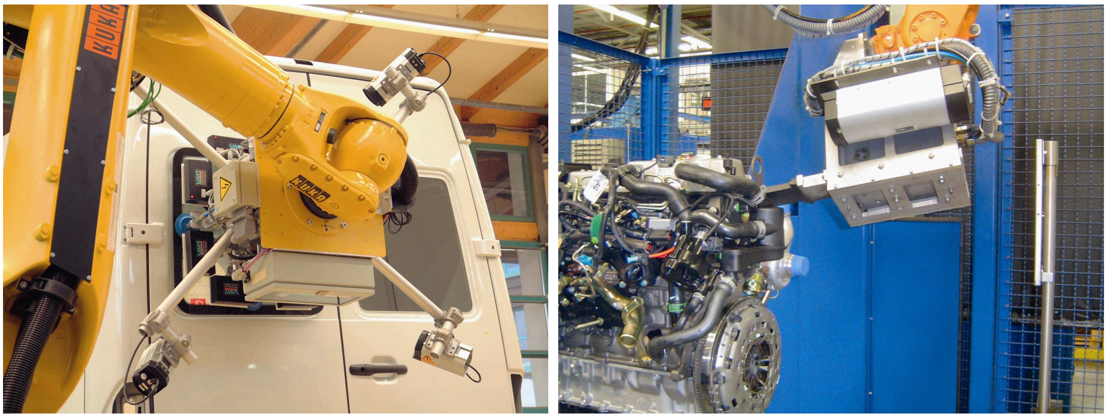
$$^{w}\vec{p}_i = {}^{w}T_o \cdot {}^{o}\vec{p}_i \qquad i = 1, 2, 3, \cdots \tag{5.17}$$

5.3.4 Application with a mobile stereo sensor

The familiar setup of stereo vision in most applications is to mount the stereo sensor on a robot hand, which can be seen from figure 5.8. The application we have tested in laboratory is for a robot to mount automatically a windowpane onto a car body.

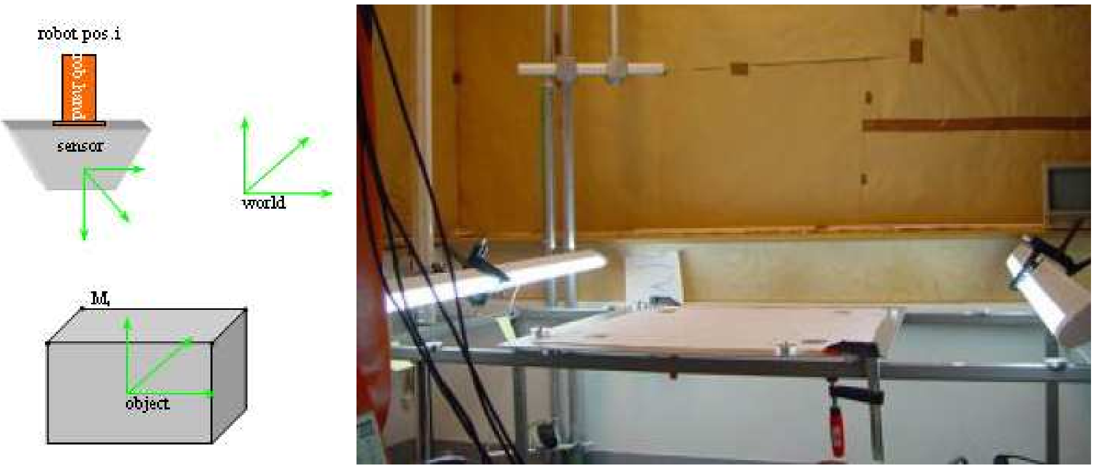
Solutions from multi-camera system

The current industrial solutions for such problems from VMT GmbH are to apply a multi-camera system, which is shown in the below figure. The left setup in the figure is using a single camera mounted on the robot hand. The robot goes to some measuring positions to let the camera search the selected features. Except that the setup needs some more seconds to complete the measure procedure, the following estimation algorithm is exactly the same as that there are many cameras. The right setup in the figure is especially designed for windowpane mounting. It does not estimate any pose of any object, what it measures is the gaps between the window rim and the pane in all directions and finally guide the robot to find a best-fit position for mounting.



(a) multi-camera setup (b) hand camera setup

Figure 5.11: current solution in applications



(a) stereo solution (b) experimental setup

Figure 5.12: stereo vision experiments in laboratory

Solution from stereo vision system

The greatest advantage of the stereo vision solution, as shown in figure 5.12, is that a new object can be automatically introduced.

New object definition

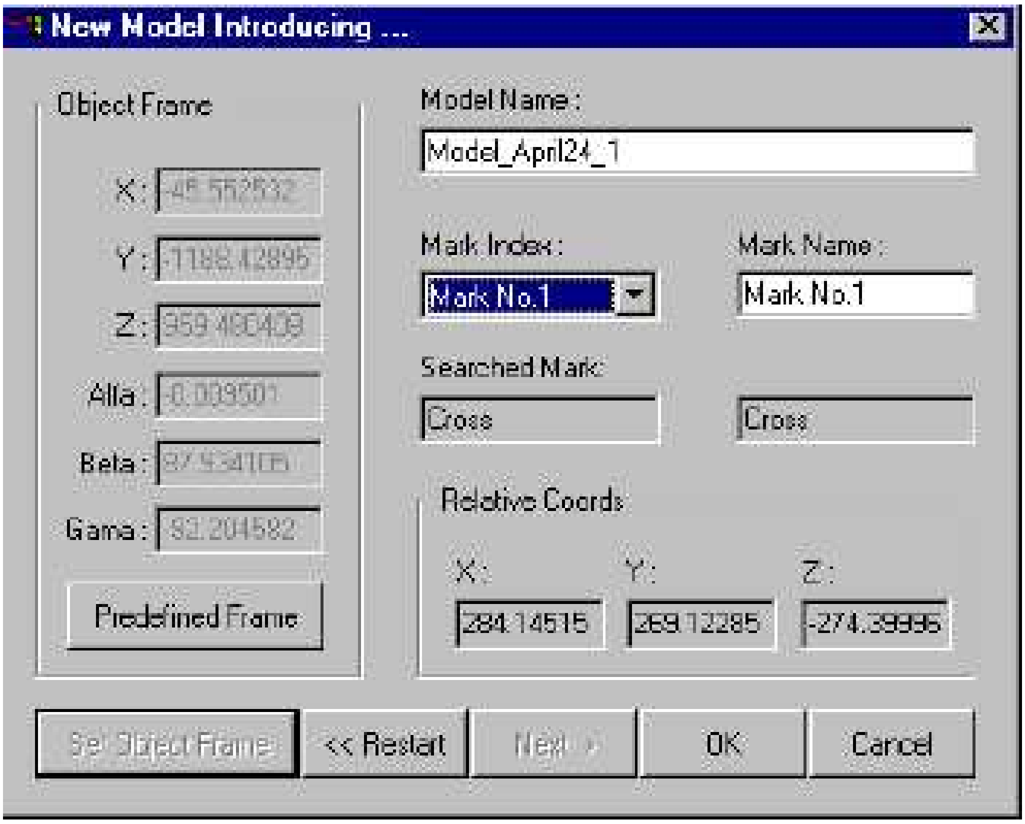
New object definition is to set up an object frame and get the coordinates of its marks with respect to the object frame. For convenience of depiction, the robot base frame is taken as the world frame and the procedure in our application is as follows

1. Drive the robot hand with the stereo sensor to a position ${}^{w}T_h$ near the object and define the sensor frame ${}^{w}T_s = {}^{w}T_h \cdot {}^{h}T_s = {}^{w}T_o$ as the object frame, which at this moment is also called zero frame ${}^{w}T_z = {}^{w}T_o$.

2. Drive the robot into such positions one by one that the sensor can see the selected marks on the object. If the robot hand positions are denoted as ${}^{w}T_{hi}$, the sensor positions must be ${}^{w}T_{si} = {}^{w}T_hi \cdot {}^{hi}T_{si}$.

3. Measure the mark with the stereo sensor and denote the coordinates in the sensor frame as ${}^{si}\vec{p}_i$ at each position.

4. Get the coordinates of the marks with respect to the object frame ${}^{o}\vec{p}_i = {}^{w}T_o{}^{-1} \cdot {}^{w}T_{si} \cdot {}^{si}\vec{p}_i$.


New Model Introducing ...

Object Frame

X: -48.552532

Y: -1188.42896

Z: 959.490409

Alfa: -0.009501

Beta: 87.934106

Gama: 93.204582

Model Name: Model_April24_1

Mark Index: Mark No.1

Mark Name: Mark No.1

Searched Mark: Cross Cross

Relative Coords

X: 284.14515 Y: 269.12285 Z: -274.39996

Figure 5.13: introducing a new object

Figure 5.13 is an example in the laboratory to introduce a new object with four marks. The left six values are to define the object frame and the right bottom three values are the calculated coordinates of the current mark with respect to the object frame.

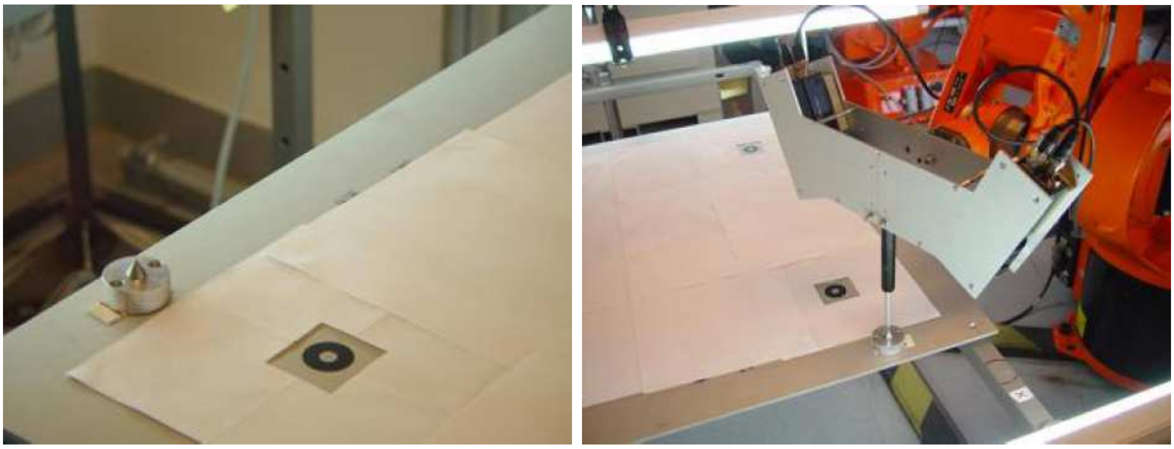
Experimental results and accuracy

Since the stereo sensor has been tested with high accuracy by using the calibration setup, there is no external measure system applied for checking the absolute accuracy. The values in table 5.5 are the results in the application.

	x	y	z	rx	ry	rz
at zero position	0.010	-0.004	0.011	-0.0011	0.0025	0.0006
an arbitrary pos	-0.166	-3.335	-6.013	0.3840	-0.0003	-0.0012

Table 5.5: Test measurements with mobile sensor

However, the absolute deviations of the stereo sensor can be seen in the first row of the table. Once a new object is introduced, a test measurement is carried out immediately with the object position unchanged. To check the application accuracy, the robot is taught to simulate the real application, some verification points touching as shown in figure 5.14. For this purpose, the zero frame must be remembered by the robot and the complete procedure will be described in detail in the section of *multi-camera system*. In laboratory, the application deviations are less than 1mm, which is estimated by human eyes and including the robot error in absolute positioning, which has a normal range of 0.5mm to 1mm for the applied KUKA robots.



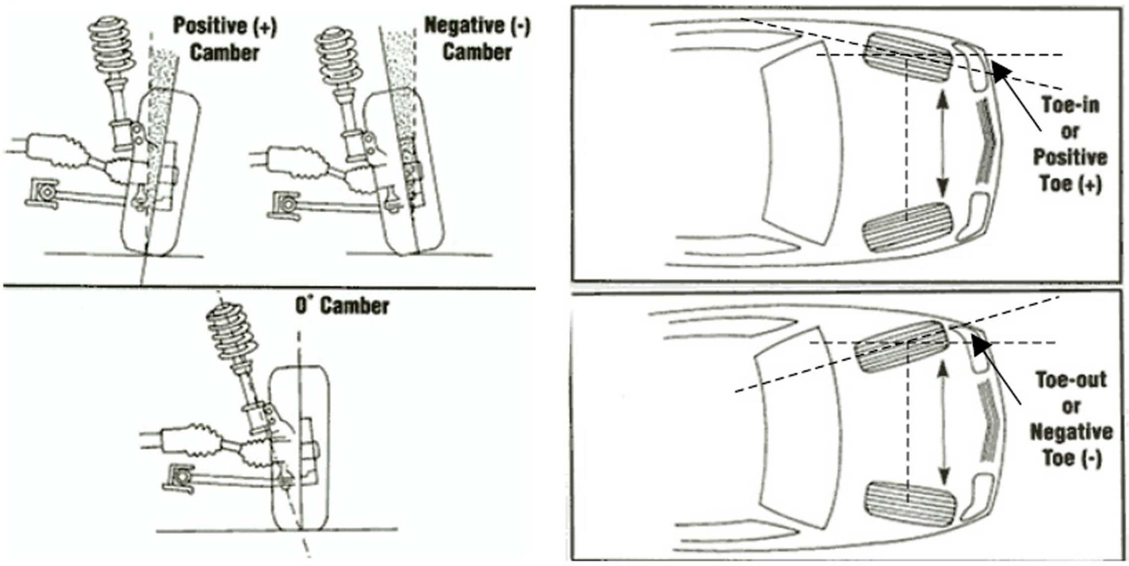
(a) verification points

(b) application simulation

Figure 5.14: accuracy testing of the stereo vision

5.3.5 Application with stationary stereo sensors

Currently the alignment of the wheels, or checking if the two front wheels of an automobile are parallel in mounting states, is more and more concerned by the automobile producers and handlers. As shown in figure 5.15, the measurement task is to determine the two angles, the camber and the toe.



(a) camber angle

(b) toe angle

Figure 5.15: installation verification of the front wheels

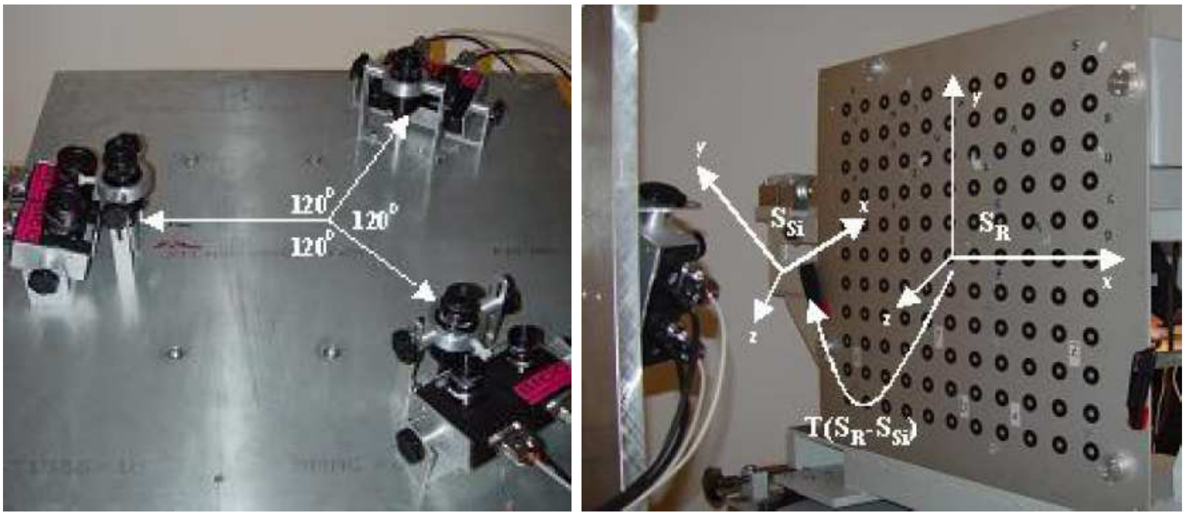
In our laboratory, a vision solution with three stereo sensors is proposed to determine the pose of the wheel plane. If poses for both front wheels are determined, the alignment problem is solved.

System setup and initial calibration

As seen from figure 5.16, the three sensors are fixed relatively on a metal plate with an angle of 120 degree to each other, which is a reasonable setup for the three sensors to determine the wheel plane. In order to constitute a vision system with the three sensors, their relative pose must be determined. This task is usually called initial calibration.
To initialize the system, a calibration plate with 121 points in an array is used to define a common reference frame for the three sensors. The reference is defined as shown in figure 5.16 and the coordinates of the points with respect to the reference frame are known. As described before, the sensors can be calibrated if they can see no less than three points. Then the sensors can measure any point with respect to the reference frame.

The wheel plane and the structured light

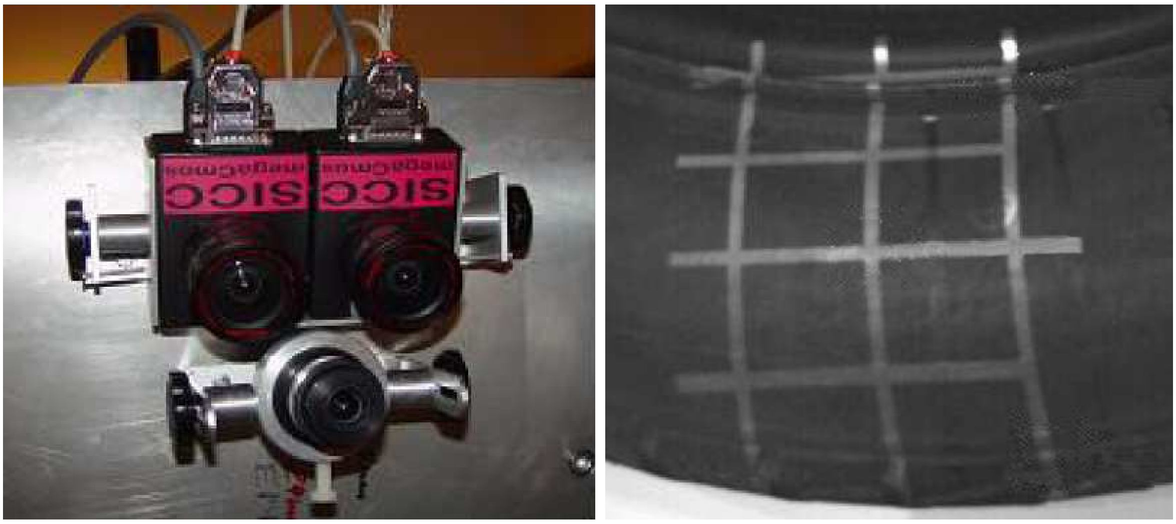
To measure the pose of the tire plane, a coordinate frame for the tire plane must be defined. As seen from figure 5.18, the rotation axis of the tire is defined as the z axis of the tire frame and the x direction is forward and y direction upward. Thus the camber and toe angles are determined by measuring the plane of the tire, or the rotations of the



(a) sensors layout (b) initial calibration

Figure 5.16: experimental setup

defined frame around x and y axes, namely RX and RY respectively. To measure the plane of the wheel, we should find on the wheel edge more than two fixed marks, which are coplanar and define the wheel plane. However, the solution will have no practical values if we must set some marks with hand onto the wheels for every automobile to be measured. In order to have the same marks for all types of wheels, a special designed structured light is applied in our experiments.



(a) cameras and light projector (b) wheel under structured light

Figure 5.17: stationary sensor with structured light

As seen from figure 5.17, the structured light draws three radial white lines and at least three white lines in the perpendicular direction on the dark wheel. The intersections of the white lines are to be recognized as pattern marks. With these marks, we can solve the two main issues in our experiments as follows

1. **Pair-matching of patterns**
   Pair-matching of patterns is an important and necessary procedure in coordinates

determining with a stereo vision system. Careful readers may notice that the structured light is so designed that the right radial line has only one cross intersection and others are trifurcate intersections, at least two of which must be seen. The three intersections marked with circles are serving as control points, which can clearly define a 2D coordinate frame in camera images. With the help of the frame, all marks in the image can be ordered into an array. Thus the pair-matching procedure is easily solved and the coordinate determination with the stereo sensors is possible.

2. **Determination of tire profiles**
   As we know, the tire surface is not at all a plane. The most difficult problem in the experiment is to get some marks which can define the wheel plane. The solution in laboratory is to recognize the profiles on the tires, which can be seen from figure 5.18. Coming back to the camera image of the structured light, the three radial white lines must have intersections with the tire profiles. Without loss of generality, let us look into only one line: with interpolating splines, the coordinates of the points on the line can be determined in reference frame and should form a smooth curve in space. However, the profiles on the tire may cause some heaves or concaves to the curve. Through analyzing the heaves or concaves from all three stereo sensors, we can get nine points on some profile, which is to be thought to define the tire plane and its pose can be determined.

Experimental results

To check how well the solution with a stereo vision system works, a specially designed workbench as shown in figure 5.18 is applied.

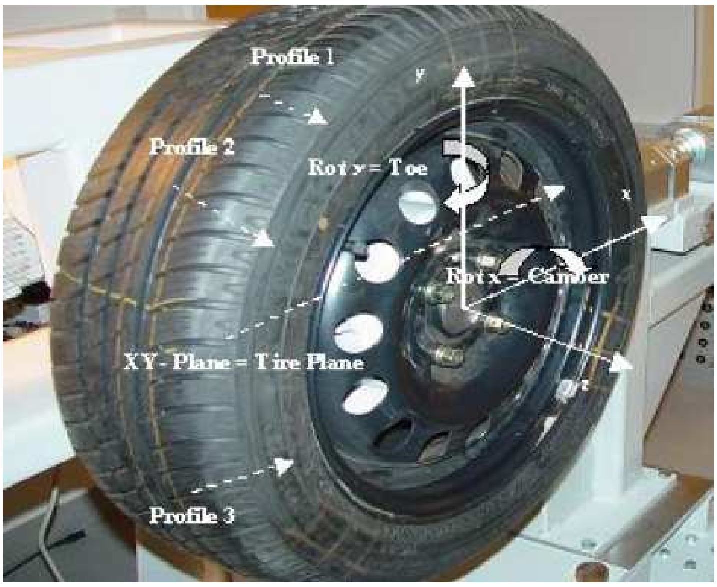


Figure 5.18: workbench for a wheel measuring

With this workbench, it is possible to adjust in both camber and toe angles and there are indicators for showing the current angles. Both of the adjusting ranges are from -3.0 to 2.0 degrees, which covers the most cases in practice. The experiment procedure is as follows

1. Set both camber and toe angles from the workbench to zero and define the tire frame, actual a plane but its vector direction is defined, as the zero frame.

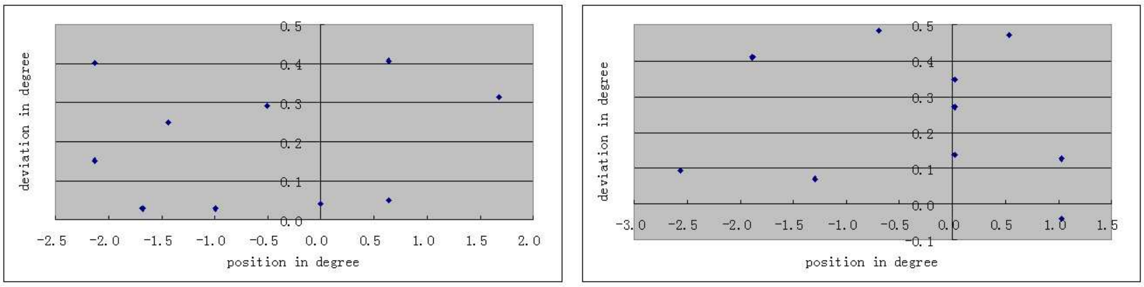
2. Measure with the three sensors to determine the zero frame ${}^{w}T_0$ and save the zero frame.

3. Adjust the tire to an arbitrary position by the camber and the toe angles $\{{}^{should}RX,$ ${}^{should}RY\}$, which can be read out from the workbench, and estimate the current tire frame with the sensors ${}^{w}T_i$.

4. Calculate the relative frame to the zero frame as ${}^{0}T_i = {}^{w}T_0{}^{-1} \cdot {}^{w}T_i \longrightarrow \{{}^{is}RX, {}^{is}RY\}$.

5. Get the deviations for the camber and the toe angles $\delta RX = {}^{is}RX - {}^{should}RX, \delta RY = {}^{is}RY - {}^{should}RY$.

Part of the results are shown in figure 5.19 and the deviation ranges are within 0.5 degree. The left figure shows the camber deviations according to the camber positions and the right figure shows the toe deviations according to the toe positions. In both figures, there are some cases that there are different deviations at the same position. That means, the camber deviation is influenced not only by the camber position, but also in some degrees by the toe position. For the same reason, the toe deviation arises also from both the camber and toe positions.



(a) camber contribution

(b) toe contribution

Figure 5.19: deviations in camber and toe measuring

In figure 5.19, the application accuracy seems not bad. In fact, the stability and applying conditions of the experiment have not reached the practical requirements. That is why the expensive laser solutions are widely used in this field. However, it is a good start for us to continue in the direction with the vision solutions. To get greater application values, the illumination schedule, consequently the pattern recognition and pair matching, especially the latter, must be improved.

5.4 Multi-camera system

A multi-camera system is of course a vision system with more than 2 cameras. Such a vision system is usually used in the automotive industry for measuring large working objects, such as a car body, which is often so large that a mono-camera or even two cameras can not *overview* it. In such a situation, the cameras are usually hidden in a corner far away from the measuring object, normally two or three meters away, sometimes even to five meters away, which makes it impossible for a mono-camera or stereo vision system to do the measurements. A typical layout of industrial applications with a multi-camera system is shown in figure 5.20.

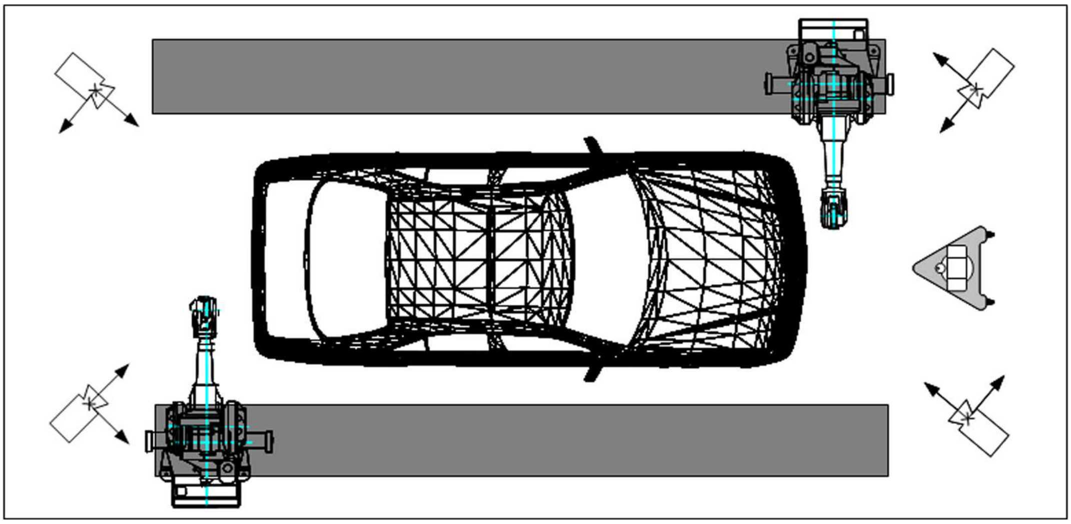


Figure 5.20: a typical layout for sealing applications

The application can be outlined as follows: an object to be processed, such as a type of a raw car, is transported and stops in the work cell. The robots try to carry out the routine applications, which are taught when a sample car stops in the ideal stop position. In production, the transportation equipments are usually not so accurate as expected and the car stops in such a position that is a little different from the ideal stop position. A multi-camera system is introduced for serving as an *eye* by measuring the current object pose, with which the robots adjust the working paths and do the applications at the correct positions with the correct poses.

Remark: The component denoted as a triangle in figure 5.20 is an additional measurement system, such as a laser tracker, which is only needed at the initializing procedure to define a world frame, relating the cameras poses and robots bases to the world frame, and adjusting the robots linear tracks.

5.4.1 Measurement task

In mathematics, the setup in figure 5.20 can be abstracted into figure 5.21. There are three types of coordinate frames: the world frame, the camera frame and the object frame. The world frame is defined by the user, and it is the reference to all the following work. The cameras have their own coordinate frames, which are marked with green color in the figure and determined with respect to the world frame in the calibration procedures. The object, here is the car body, is taken as a perfect rigid body and can be defined in mathematics by more than three features on the car body. The coordinates of the features marked with blue color in the figure with respect to the object frame must be given, usually can be obtained with the CAD drawings from the car body designer. The measuring task of the system hereby is to determine the transformation from the world frame to the object frame, which is usually called object pose and marked in the figure with red color.

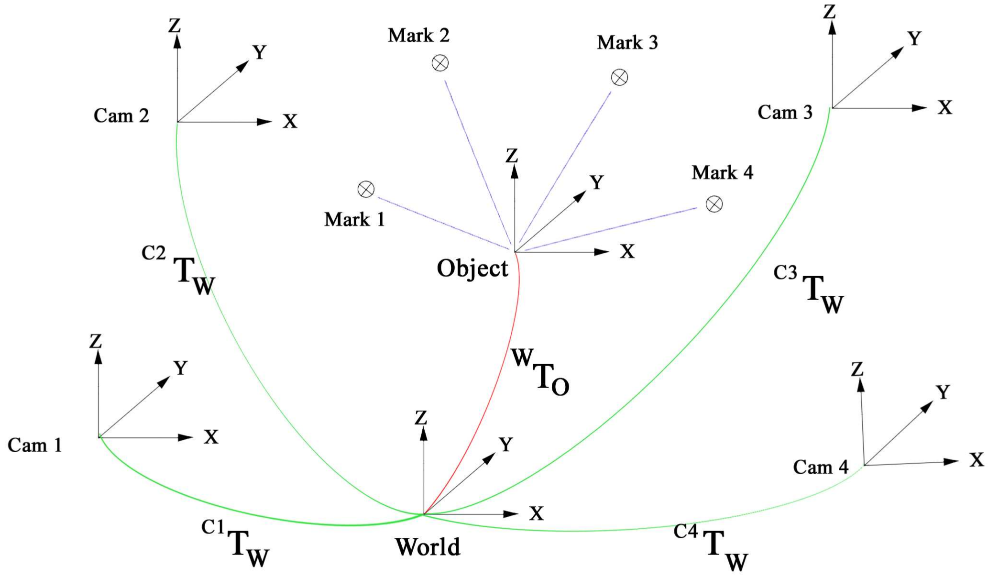


Figure 5.21: coordinate frames in a multi-camera system

5.4.2 Pose estimation

For convenience of calculation, we don't compute directly the transformation from the world frame to the object frame, but the inverse transformation which is denoted as follows

$$^{o}T_{w} = [R|\vec{t}] = \{\alpha, \beta, \gamma, x, y, z\} \tag{5.18}$$

Since there are many cameras, we cannot consider things any longer in the camera frame. Taking the above transformation for granted, the coordinates in the object frame can be related with the direction vector in the camera frame again

$$\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}^{o} = {}^{o}T_w \cdot {}^{w}\vec{p}_i = {}^{o}T_w \cdot {}^{w}T_c \cdot {}^{c}\vec{p}_i = {}^{o}T_w \cdot {}^{w}T_c \cdot \delta_i \cdot \begin{pmatrix} X_i \\ Y_i \\ 1 \end{pmatrix} \tag{5.19}$$

Removing the scale factor δ and representing the transformation ${}^{o}T_{w}$, the above equation can be rearranged into the following two equations

$$F_x(\alpha, \beta, \gamma, x, y, z) = 0 \tag{5.20}$$

$$F_y(\alpha, \beta, \gamma, x, y, z) = 0 \tag{5.21}$$

With $N \geq 3$ object points, one will get an overdetermined nonlinear system, which can be solved with a good set of initial guess.

Remark: In practice, the reference frame is usually defined near the working objects, and the translation and rotation are relatively small. With a set of all zeroed initial guess, one may expect to obtain the satisfied results from the overdetermined system.

To get a set of initial values for the elements of ${}^{o}T_{w}$, we denote N object points with known coordinates in the object frame as follows

$${}^{o}\vec{p}_i = ({}^{o}x_i, {}^{o}y_i, {}^{o}z_i)^T \quad i = 1, 2, \cdots, N \tag{5.22}$$

With the fact that the distances between the points are kept unchanged whichever frame the coordinates are with respect to, one has

$$\begin{aligned}(x_i^w - x_j^w)^2 + (y_i^w - y_j^w)^2 + (z_i^w - z_j^w)^2 &= \\ (x_i^o - x_j^o)^2 + (y_i^o - y_j^o)^2 + (z_i^o - z_j^o)^2 &= d_{ij}^2\end{aligned} \tag{5.23}$$

where x_i^o, y_i^o, z_i^o and x_j^o, y_j^o, z_j^o are known, the distance can be easily calculated. Since all the integrated cameras are calibrated, the pose of the camera corresponding to the object point ${}^{o}\vec{p}_i$ is denoted

$${}^{w}T_{ci} = {}^{w}R_{ci} + {}^{w}\vec{t}_{ci} \tag{5.24}$$

The coordinates of the object points in the camera frame can be denoted by the direction vectors

$$\begin{pmatrix} {}^{ci}x_i \\ {}^{ci}y_i \\ {}^{ci}z_i \end{pmatrix} = \delta_i \begin{pmatrix} X_i \\ Y_i \\ 1 \end{pmatrix} \qquad (i = 1, 2, \cdots, N) \tag{5.25}$$

Then the coordinates in the world frame can be represented as follows

$$\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}^w = {}^{w}R_{ci} \cdot \delta_i \begin{pmatrix} X_i \\ Y_i \\ 1 \end{pmatrix} + {}^{w}\vec{t}_{ci} = \delta_i \begin{pmatrix} L_i(X_i, Y_i) \\ M_i(X_i, Y_i) \\ N_i(X_i, Y_i) \end{pmatrix} + \begin{pmatrix} tx_i \\ ty_i \\ tz_i \end{pmatrix} \tag{5.26}$$

where L_i, M_i, N_i are three functions defined by the pose of related camera. Replacing the above equations into the distances calculating

$$F_{ij}({}^{w}R_{ci}, {}^{w}\vec{t}_{ci}, X_i, Y_i, {}^{w}R_{cj}, {}^{w}\vec{t}_{cj}, X_j, Y_j, \delta_i, \delta_j) = d_{ij}^2 \tag{5.27}$$

where only δ_i, δ_j are variables. With $N \geq 3$ object points, one will get a nonlinear overdetermined system with N variables and the initial values for δ_i are obtained with the following rules

1. Select a camera which searches more than one feature. If there is such a camera C_k searches more than one feature, denote two of them as i and j and go to **step 3**; otherwise go to **step 2**.

2. Select such two cameras c_i, c_j that the difference of their lens focuses is the minimum, that is $|f_i - f_j| \longrightarrow min$ and get two features m_i, m_j respectively from the two cameras, then go to **step 3**.

3. Similar to the mono-camera system, assume that $\delta_i = \delta_j$ and it yields

$$F_{ij}({}^wR_{ci}, {}^w\vec{t}_{ci}, X_i, Y_i, {}^wR_{cj}, {}^w\vec{t}_{cj}, X_j, Y_j, \delta) = d_{ij}^2$$

   Solving $\delta = \delta_i = \delta_j$, all other scale factors can be obtained one after another by relating to δ_i or δ_j.

With the initial values, a set of final values of δ_i will be obtained. Then the coordinates of the features with respect to the corresponding cameras are obtained, consequently the coordinates with respect to the world frame

$$\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}^w = {}^wT_{ci} \cdot \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}^{ci} = {}^wT_{ci} \cdot \delta_i \begin{pmatrix} X_i \\ Y_i \\ 1 \end{pmatrix} \qquad (i = 1, 2, \cdots, N) \tag{5.28}$$

Noticing the fact that the coordinates of the points with respect to the object frame are known, we have $N \geq 3$ points, whose coordinates with respect to both the world frame and the object frame are known. Therefore, the transformation ${}^cT_o$ can be estimated by a best-fit procedure. Finally the transformation is used as the initial guess in the nonlinear equations of 5.20 and 5.21 and then a better set of values for ${}^cT_o$ will be estimated.

5.4.3 Pattern compensation

In the above estimation procedure of the object pose, the 2D coordinates from patterns recognition and the 3D coordinates of the patterns in the object frame are simply matched through equation 5.19. The presupposition to match the 2D and 3D coordinates of a pattern is that the 2D pixel is exactly the projection from the 3D pattern point. Let's have a look how the presupposition is satisfied in practice.

1. **3D coordinates of a pattern**
   The pattern referred here is an object mark, which is normally a part of the object. When a pattern is selected for the vision system to 'search', its 3D coordinates are usually obtained from the design drawings of the car body. The values may be accurate, but the problem is the quality of the finished products. For instance, the common coordinate deviations of the patterns from a car body is about 2 mm, which makes it impossible for a vision system to determine the pose of the car body with the errors within 1 mm.

2. **2D coordinates of a pattern**
   In the camera image, a pattern is a block of pixels, of which a suitable pixel on the pattern is defined as the reference point usually through a mouse click. The coordinates of the reference point from the pattern recognition indicate the current position of the pattern and are used for continuous calculation.

As described above, the correspondence of the 2D pixel and 3D pattern point is guaranteed by human eyes. Thus the deviation may be to a few pixels and the error will be reflected on the result of the object pose determination. In fact, there is no difference whatever pixel is defined to be the reference point. The important factor is that the pixel indicated by the 2D coordinates is exactly projected from the object point by the 3D coordinates. A procedure called compensation of pattern coordinates is applied for solving the correspondence of 2D / 3D coordinates: according to the recognition results, the 3D coordinates of the patterns are adjusted slightly to reach a best-fit between all values. The adjustment of the pattern coordinates is called pattern compensation.

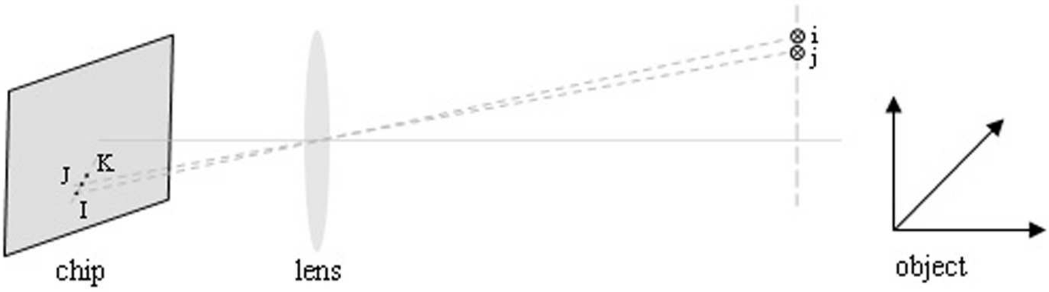


Figure 5.22: compensation of pattern coordinates

As shown in figure 5.22, an object point i is projected into the pixel I, but the coordinates from the pattern recognition indicate the pixel K, which is different from I. In the compensation procedure, we try to adjust the object point, actually its 3D coordinates, so that the new point j is projected exactly into the pixel K = J. If ${}^{o}\vec{p}_i, {}^{c}\vec{p}_i$ denote the the 3D coordinates of the object point in the object frame and the camera frame respectively and P_i the responding 2D coordinates, the compensation procedure is carried out in the following steps:

1. **Estimation of the object pose**
   With the camera parameters $A, {}^{w}T_{ci}$, the object pose can be estimated with the referred procedure and denoted as ${}^{w}T_o$.

2. **Calculation of 2D deviations of the patterns**
   Once the object pose is obtained, the coordinates ${}^{c}\vec{p}_i$ of patterns with respect to the related camera frame can be determined. As shown in figure 5.22, an object point i is taken for instance. Since the related camera is calibrated, its projection I can be easily calculated. In theory the pixel I should be the same pixel as the pattern recognition result K. In practice there have errors and the difference is denoted as

   $$\delta\vec{P}_i = |\vec{P}_i - \vec{P}_k| \qquad i = 1, 2, \cdots, n$$

   Supposing ε is an user defined sufficient small value. If $\delta\vec{P}_i < \varepsilon$ for all patterns, the procedure is completed; otherwise continue to the next step.

3. **Creation of a new 2D image pixel J**
   To reduce the deviation, a new pixel J is created by

   $$\vec{P}_j = \lambda\vec{P}_k + (1-\lambda)\vec{P}_i \qquad 0 < \lambda \leq 1 \tag{5.29}$$

If $\lambda = 0$, no compensation is applied; if $\lambda = 1$, the pattern recognition result is taken as the new pixel.

4. **Creation of the corresponding object point j**
   Since the new pixel J is closer to the pixel K than the image of the pattern i, the object point i should be compensated into j, which will project into J. However, an image pixel can not determine an object point. Thus the new object point j is considered to have the same distance as the point i to the plane of camera lens, that is

$$^{c}\vec{p}_i = (x_i, y_i, z_i)^T = \delta_i (X_i, Y_i, 1)^T = \delta_i \vec{P}_i$$
$$^{c}\vec{p}_j = (x_j, y_j, z_j)^T = \delta_j (X_j, Y_j, 1)^T = \delta_j \vec{P}_j$$

   where $\delta_i = \delta_j$ and $^{c}\vec{p}_i, \vec{P}_i$ and $\vec{P}_j$ are known, $\vec{p}_j$ is easily calculated, consequently the coordinates with respect to the object frame

$$^{o}\vec{p}_j = (^{w}T_o)^{-1} \cdot {}^{w}T_c \cdot {}^{c}\vec{p}_j.$$

5. **Update the 3D coordinates of the patterns**
   Taking the calculated coordinates $^{o}\vec{p}_i = {}^{o}\vec{p}_j$ for all patterns and go back to the first step to continue the procedure.

When the above procedure is completed, the projections of the 3D object points must be close enough respectively to their corresponding 2D pattern pixels. If the original coordinates and the current coordinates of the patterns are denoted respectively as $^{org}\vec{p}_i$ and $^{cur}\vec{p}_i$, the compensations are defined as follows

$$^{com}\vec{p}_i = {}^{cur}\vec{p}_i - {}^{org}\vec{p}_i \qquad i = 1, 2, \cdots, n \tag{5.30}$$

Remark: The value λ in creation of a new pixel seems to indicate the convergence speed of the compensation procedure. However, one must notice that the newly calculated coordinates of patterns may lead to a new object pose, which will cause new coordinates of patterns in the camera frames and finally the new image pixels. Sometimes $\lambda = 1$ may cause the repeating procedure to wiggle, instead of converge. In practice $\lambda = 0.5$ is applied and it converges actually with only several times of calculation.

5.4.4 Pattern weight

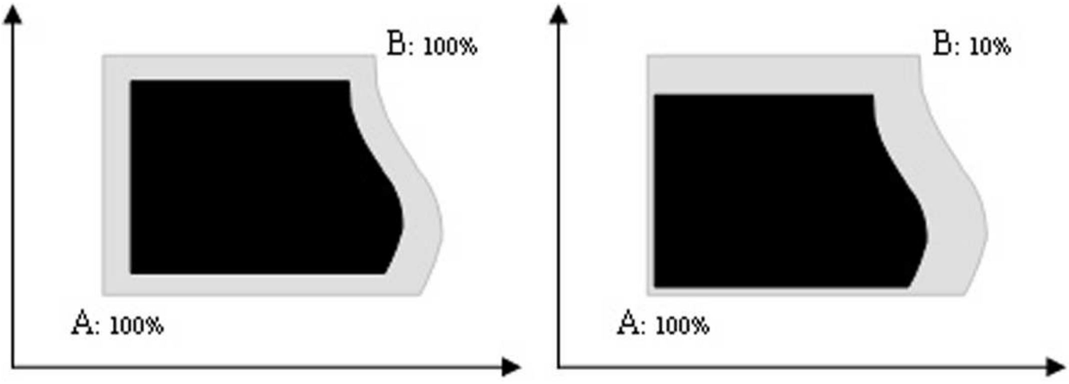
To estimate the object pose, the equations 5.20 and 5.21 are in ideal situation. In fact, F_x and F_y will never be exactly equal to zero. The algorithm to solve the over-determined system can be interpreted as minimizing the following function

$$\sum_{i=1}^{n} F_x^2 + F_y^2 \rightarrow min \qquad n \geq 3 \tag{5.31}$$

where n is the count of patterns. The above function takes all patterns without differences. Sometimes different patterns have different significances in determining the object pose. The function can be improved by introducing a weight factor for every pattern

$$\sum_{i=1}^{n} (F_x^2 + F_y^2) \cdot w_i^2 \rightarrow min \qquad 1 \geq w_i \geq 0 \tag{5.32}$$

where w_i is the user assigned weight factor. In real applications, the weight factors are in percentage forms, 100% is for normal patterns, 0% for the discarded patterns and the values from 1% to 99% for the less significant patterns.



(a) same weights (b) different weights

Figure 5.23: strategy of weights assignment

Generally speaking, it may be useful to assign different weights for different patterns in the following cases

1. The object to be measured is not a rigid body and its parts are subjected to distortion. The patterns from the distorted parts can be given a smaller weights to estimate the object pose more correctly.
2. The object to be measured may zoom within a certain degree. The normal strategy with the same weight for all patterns will lead to such a pose result that the errors are dispatched equally into the patterns as shown in figure 5.23a. Users can assign different weights for the patterns according to their significances in application to get a flexible result as shown in figure 5.23b.

Remark: The example from figure 5.23 is with 2 patterns to estimate the 2D pose (X, Y, RZ) of an object moving in a plane parallel to the camera chip. Pattern B does little work in estimating the shift X and Y, but is necessary in determining the rotation RZ.

5.4.5 Zero measurement

As shown in figure 5.21, all cameras are calibrated with respect to a world frame and the multi-camera system estimates the object pose of course in the world frame. In the best case, the object should come and stop exactly at the position where the robot routine is taught with a sample object. Normally the object comes and stops usually at a different position nearby. If we name the object pose with respect to the world frame as *absolute pose* and the pose with respect to the ideal position as *relative pose*, the robot needs the relative pose for doing its work correctly. Therefore, a *zero measurement* procedure is in practice introduced for the multi-camera system to measure the ideal pose and 'remember' it for later converting absolute poses into relative poses.

In practice, the ideal pose of the object must have been determined in initializing the system, which gives us a chance to verify the accuracy of the multi-camera system by contrasting the result vectors respectively from the additional measure system and the multi-camera system. What troubles us is the fact that the differences, about 2 mm in translation and 0.2 degree in rotation, are not satisfied to the industry required accuracy, maximum 1.0 mm in translation and 0.1 degree in rotation, which includes the robot error additionally. The possible error sources may be as follows

1. **Camera modeling**: whatever model is applied, it can not describe the camera behavior exactly correct and errors may arise in mathematical calculation.

2. **Camera calibration**: camera calibration is an elaborate procedure which may consist of many steps, such as 2D and 3D coordinates gathering. Each step may have errors and the errors are inherited into the camera parameters.

3. **Object distortion**: the multi-camera system referred here is mainly for estimating the pose of rigid objects. If the current object is not rigid enough, the measurement result, which is computed by taking the object an exactly rigid one, is of course not highly satisfied.

4. **Pattern recognition**: the result from pattern recognition can be accurate to pixel, even sub-pixel, but can not be 100% correct.

For a well-running vision system, the measuring errors arisen from object distortion and pattern recognition are stochastic and should be improved respectively. However, the errors from camera modeling and calibration are almost constant and totally called system error, which can be mostly eliminated through a zero measurement procedure.

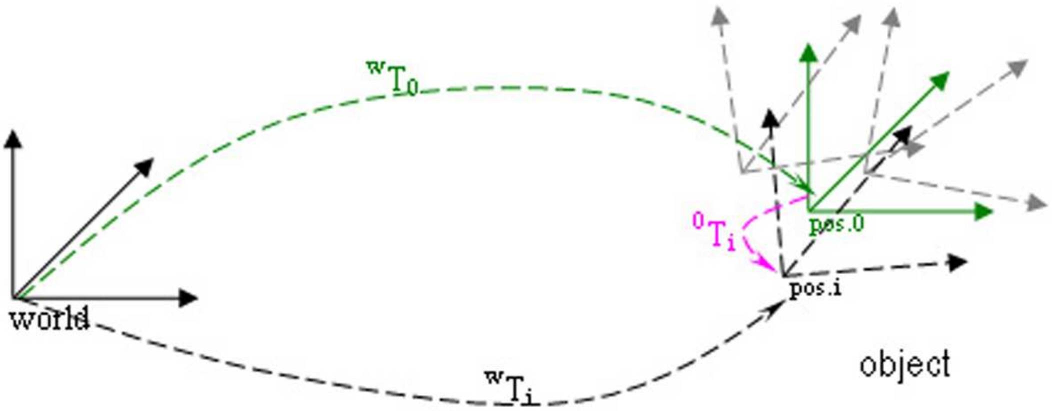


Figure 5.24: zero measurement

As denoted in figure 5.24, $^wT_0$ is for the absolute pose of the object in ideal position, ^wT_i for the object pose in an arbitrary position and consequently ^0T_i for the relative pose. In a qualified production line in industry, the arbitrary position is relative close to the ideal position. For example, in a PVC fine sealing station, the tolerances of the stop position for transporting a raw car are about 30 mm in translation and 0.2 degrees in rotation. Since the external factors are so similar, the system errors included respectively in $^wT_0$ and ^wT_i are supposed to be same and finally the relative pose ^0T_i should be much more accurate in correcting the robot path after a zero measurement procedure.

5.4.6 Security control

As known, security is the most important thing for applications in automobile industry. If the result vector from the vision system is not correct but is given directly to the related robots, collisions from robots against car body may happen and consequently an elaborate procedure for robot tool calibration is needed. For security controlling, the following approaches are applied.

A. Limit for correction vector

To guarantee the measurement to be correct, the most direct way is to check the validity of the values from the correction vector. As referred in section *zero measurement*, the tolerance for the object to move around the zero position is relatively finite, which makes it possible to limit the result vector in a scope. Most of the time, we can set rigorous ranges for X, Y, Z, RX, RY and RZ respectively according to some additional conditions in real applications.

B. Effective patterns and cameras

In a successful measurement, the pattern which has been in estimating the ultimate object pose is called an effective pattern; otherwise it is a noneffective pattern. If a camera has no effective patterns related, it is in this measurement a noneffective camera; otherwise it is an effective camera. In mathematics, we need minimum 3 nonlinear effective patterns for determining the object pose, but it is commonly considered that more effective patterns lead to better and stabler measurement, which is the reason that a redundancy strategy of patterns is usually adapted in practice.
Although there is no requirements in mathematics on cameras count for pose estimation, we can see clearly how important the cameras count is to the accuracy of the vision system from figure 5.20: the car body has a size about 4 meters long and 2.5 meters wide, which makes it impossible to measure it with only one camera. To measure with two effective cameras from the system, there are two possibilities

1. **two cameras from one side**: The two cameras are supposed without loss of generality to be at the front side. When the rear of the car body turns up or down but the fore side keeps almost unmoved, the measurement result can not be accurate, especially the rotation angles. Thus the guided robot works smoothly in the front and problems may come in the behind part.

2. **two cameras from one diagonal**: There may happen such a case that the car rotates a small angle around a car body diagonal, where the two effective cameras locate, the similar problems as above may arise when the robots do their application at the other two corner parts of the car.

Measuring with three or four cameras can greatly reduce such problems to happen. Therefore, it is a complementary and applicable strategy to assure the measurement by checking the counts of effective patterns and cameras.

C. Validity of pattern recognition

Among the factors which may affect the measurement result, the main and critical error arises from pattern recognition, for example, a pattern is wrongly recognized at an other position in the image. For instance, as shown in figure 5.25, an object is defined by four patterns A, B, C and D and denoted in green. When the corner E is recognized by mistake as the pattern D, the estimation procedure tries to find such an object pose by matching the coordinates searched for A, B, C and E to the reference object patterns A, B, C and D as well as possible. As a result, the error in searching pattern D is dispatched to some degree into other patterns and an object pose spoiled by rotating the object a little is obtained.

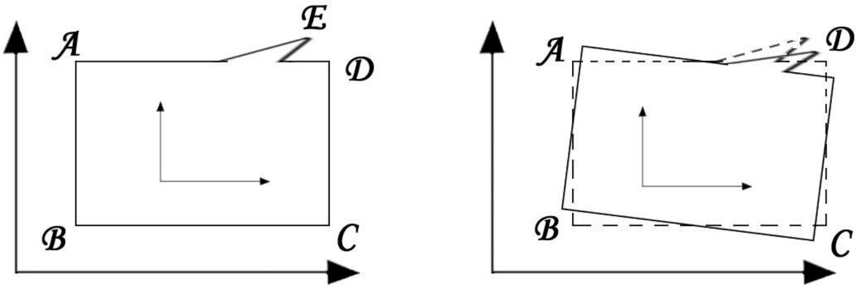


Figure 5.25: mistakes in pattern recognition

In order to *kick out* the wrongly recognized pattern, we have looked carefully into the estimation procedure and found that the deviation at the wrongly recognized pattern is larger than those at other patterns although the error is shared by all patterns. Therefore, the estimated pose ${}^{w}T_o$ is firstly taken for granted correctly and the coordinates of patterns in the camera frame is obtained by

$$^{c}\vec{p} = \begin{pmatrix} x_i \\ y_i \\ z_i \\ 1 \end{pmatrix}^{c} = {}^{c}T_w \cdot {}^{w}T_o \cdot \begin{pmatrix} x_i \\ y_i \\ z_i \\ 1 \end{pmatrix}^{o} = {}^{c}T_w \cdot {}^{w}T_o \cdot {}^{o}\vec{p} \tag{5.33}$$

With a point in the camera frame ${}^{c}\vec{p}$, its projection ${}^{should}\vec{P}_i$ in the corresponding calibrated camera can be uniquely determined and consequently the deviations $\delta\vec{P}_i$ is defined

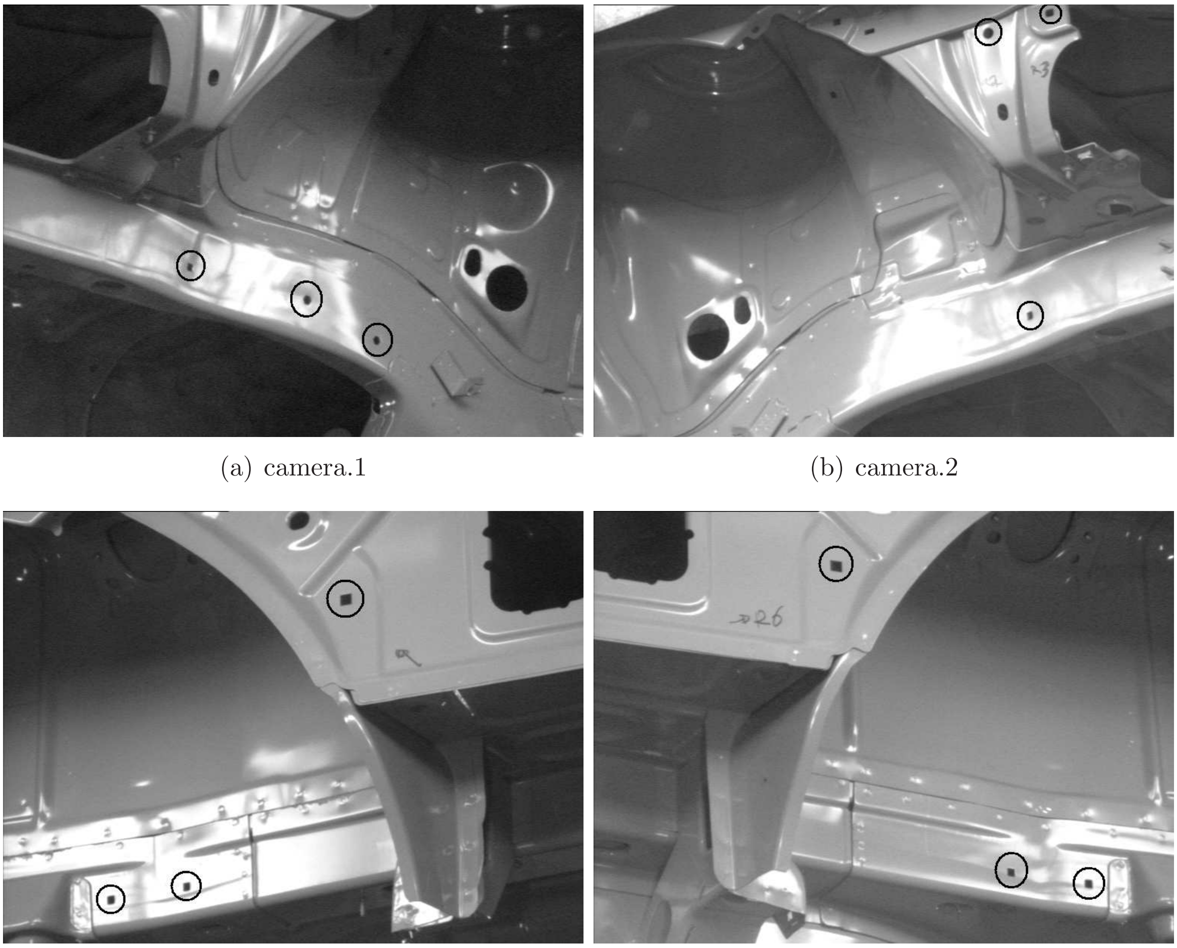
$$\delta\vec{P}_i = {}^{is}\vec{P}_i - {}^{should}\vec{P}_i, \quad i = 1, \cdots, N \tag{5.34}$$

where ${}^{is}\vec{P}_i$ is the image coordinates from pattern recognition. If tolerances are set separately for patterns, the security controlling can be completed by checking the deviations of the patterns after getting the object pose. When some patterns have deviations beyond their tolerances, they will be discarded and the system will try to estimate the object pose again with the left patterns.

Remark: The deviations can be checked in both pixels and millimeters as described in section *calibration deviations* from chapter *camera model*.

5.4.7 Experimental results

The data for this experiment are again from the practical application from VW Shanghai. The multi-camera system is composed of four JAI CV-M50 cameras, which are calibrated with a calibration body and the results are listed in table 4.8, and the lenses are of 35 millimeters focal length. In theory, it is enough for each camera to search a pattern. In practice, we usually introduce more patterns than needed for improving accuracy and stability of the measurement by a redundancy strategy. As seen in figure 5.26, the features marked in black circles are defined as our patterns for searching. Among these patterns, some of them are on the longeron of the car and have good rigidness property; some are on the shell of the car and are subjected to distortion. The patterns on the rigid part keep their coordinates well and others not. That is why we assign them different pattern weights: 100% and 50% respectively.



(a) camera.1

(b) camera.2

(c) camera.3

(d) camera.4

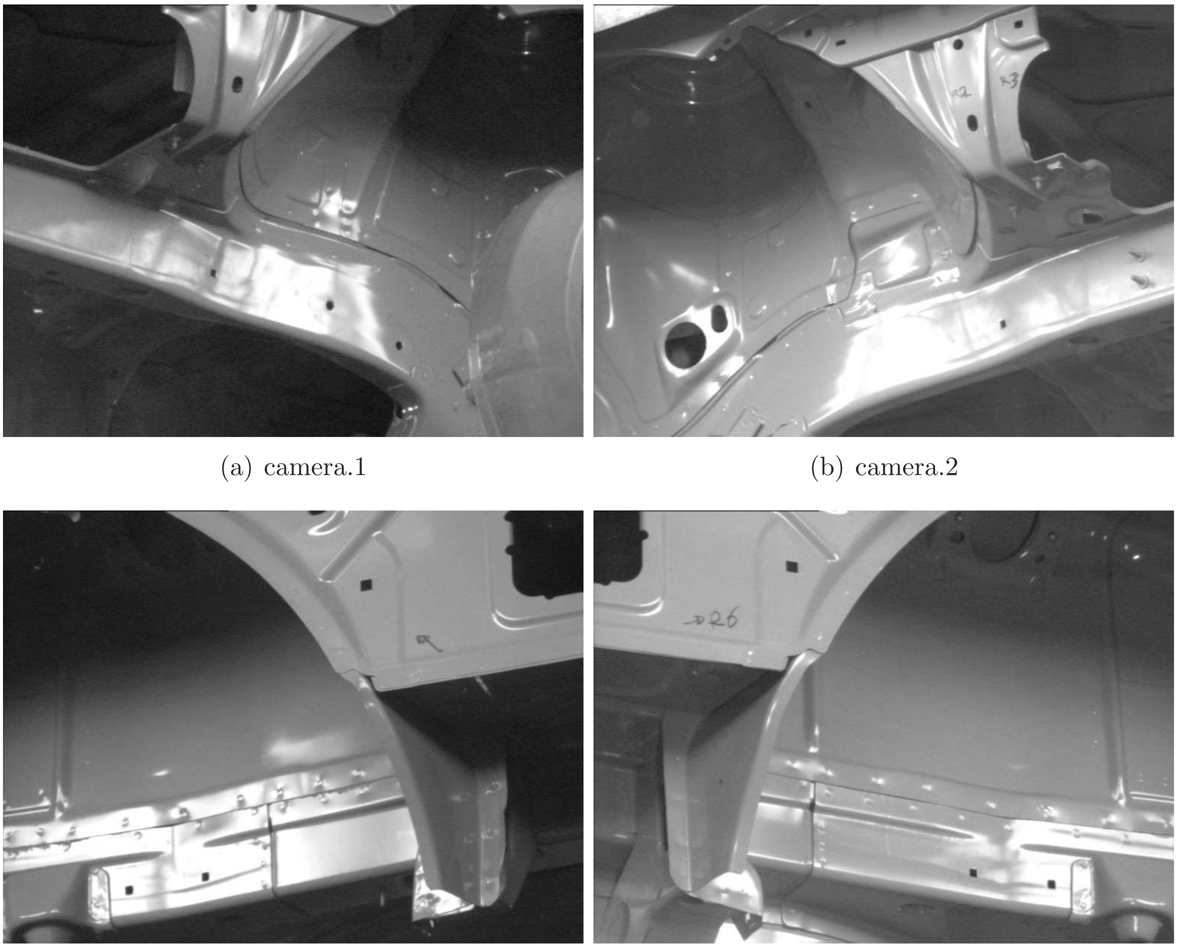
Figure 5.26: camera images at zero position

Actually the camera images in figure 5.26 are taken by the ideal stop position of the car body, where the robots routines should be taught. For our multi-camera system, the car frame at ideal position is indeed defined as the world frame,in which all cameras are calibrated. If we measure the car body with these images, the result pose of the car body should be a zero vector. In practice, VMT system has never got a zero vector and the difference this time is listed in table 5.6, which is usually called the **absolute error**.

	x	y	z	rx	ry	rz
absolute error	1.00	-0.78	1.52	-0.150	0.070	-0.020

Table 5.6: absolute measurement from the VMT system

Obviously, this error is too large for this type of applications. Especially the error of rx, 0.15 degree may cause about 4 mm deviation at a position of 2 m away. The zero measurement is introduced to reduce the errors by forcing the result to a zero vector but saving the absolute error in vision software.


(a) camera.1

(b) camera.2

(c) camera.3

(d) camera.4

Figure 5.27: camera images after a shift from zero position

To test how well the zero measurement works, a shift to the car body is made by hand and the measurements from both VMT system and also the external measure system. The camera images in figure 5.27 are taken after the shift and the measuring results are listed in table 5.7, where *absolute pose* denotes the shift without the zero measurement procedure, *relative pose* denotes the shift after the zero measurement and *laser tracker* denotes the shift measured by a laser tracker.

When the sample car stops at the ideal position, the zero measurement is usually followed by a procedure named *relaxation*, which compensates the coordinates of the patterns. After this procedure, the deviations should be relatively small in a good measure of a rigid object. The table 5.8 shows the compensations of the patterns and the current deviations in the shift measurement.

	x	y	z	rx	ry	rz
absolute pose	31.12	-2.59	1.59	-0.240	0.080	-0.270
relative pose	30.12	-1.80	0.11	-0.080	0.010	-0.240
laser tracker	30.23	-1.57	0.08	-0.086	-0.001	-0.259
absolute error	0.89	1.02	1.52	0.154	0.081	0.011
relative error	0.11	0.23	0.04	0.006	0.011	0.019

Table 5.7: relative measurement after the zero measurement

	x	y	z	x	y	z	d
Camera.1/Pattern.1	-0.78	0.22	0.61	0.06	-0.05	0.04	0.09
Camera.1/Pattern.2	0.04	-0.07	0.09	-0.02	0.10	-0.19	0.22
Camera.1/Pattern.3	1.05	-0.22	-1.22	-0.04	-0.03	-0.01	0.05
Camera.2/Pattern.1	-0.53	-0.43	-0.18	0.04	0.04	0.01	0.06
Camera.2/Pattern.2	-0.47	-0.06	0.42	-0.01	-0.01	0.01	0.02
Camera.2/Pattern.3	0.79	0.79	0.50	-0.01	-0.01	0.01	0.02
Camera.3/Pattern.1	0.12	-0.14	0.51	0.05	-0.04	-0.01	0.06
Camera.3/Pattern.2	0.19	-0.06	0.40	0.00	-0.01	-0.02	0.02
Camera.3/Pattern.3	0.07	-0.11	0.28	0.03	-0.03	0.02	0.05
Camera.4/Pattern.1	-0.78	-0.48	-2.06	0.05	0.04	0.11	0.13
Camera.4/Pattern.2	-0.41	-0.94	-2.51	-0.12	0.02	-0.06	0.14
Camera.4/Pattern.3	0.72	1.49	3.16	0.06	-0.02	0.09	0.11
	compensations			deviation in shift measure			

Table 5.8: patterns compensations and measurement deviations

In each measurement, the deviation of a pattern feedbacks the rigidness details or the recognition situation about the corresponding pattern. For instance, if the maximum distortion of features from a car is 2 mm, we can set the tolerances of deviations for all patterns to 2 mm. As a result, the nonrigid errors are permitted and errors from pattern recognition are prevented. If more details of the features distortion are known, a strategy of the different tolerances will work more efficiently.

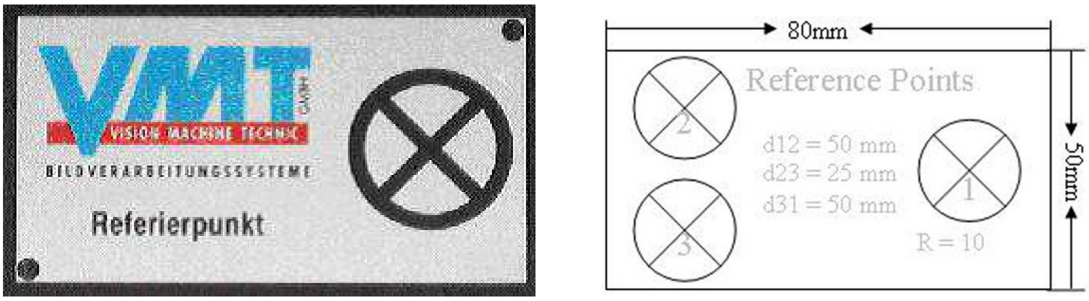
5.5 Camera pose recalibration

5.5.1 Motivations

As we know, camera calibration is important and necessary, but usually an elaborate procedure. An applied vision system can work smoothly and correctly only when all cameras are kept absolutely unchanged after the calibration. Unfortunately, in practical applications changes may happen to any camera from time to time, mostly to the camera pose. If the camera or its lens is changed, we have to carry out the elaborate calibration procedure again. In fact, the camera pose is changed much more often, e.g. changes by mistake during the daily cleaning work, changes by system re-design for a longer car body, and so on. If the camera is changed and the vision system is not informed, it may get wrong results, with which the robot maybe breaks the car body. Therefore, an approach to recalibrate the camera pose, which can be easily and quickly carried out in the working environment, has great values in industry.
Since a reference point is the easiest additional *condition* to have in an industrial production line, the proposed approaches only require the camera to observe some reference points, whose 3D coordinates in the world frame may be known or unknown. Due to the different prerequisites, we developed three online approaches to recalibrate the camera pose.

5.5.2 Reference point

A reference point, also called fixed point, in theory can be any fixed point in the working environment, e.g. a characteristic part of a fixed object, or a fixed artificial mark, and so on. In industrial applications, the reference points are usually artificial marks as seen in figure 5.28.



(a) reference point

(b) reference points group

Figure 5.28: designs of reference points

A reference points group is a collection of reference points, in which the points' positions to each other are relatively fixed. Actually, only the distances between the points are needed. The design of the reference points group in figure 5.28 is used in experiments. With this design, the ordering of the points is easy: the point farthest away from the other two points is P1, and in anti-clockwise order the other two points are P2 and P3. The reference points can be recognized well in the camera image with the camera at about 2m to 3m away. In practice, one can have any other design according to the actual situation.

5.5.3 Approach with known RPs

This approach estimates the complete camera pose, including both orientation and translation. As seen from figure 5.29, there are three reference points, whose 3D coordinates in the world frame have to be known.

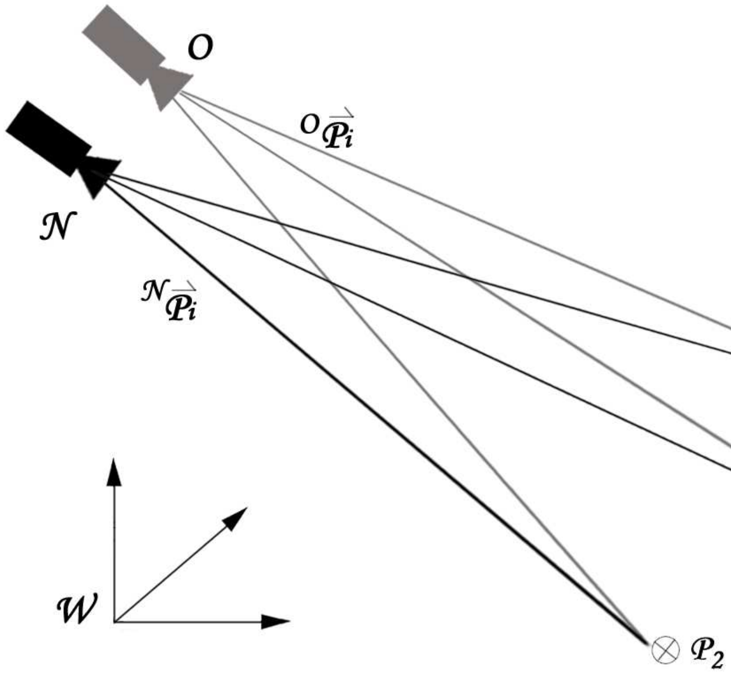


Figure 5.29: recalibration with 3 known reference points

With the coordinates of the reference points in the world frame and the transformation between the world frame and the old camera frame, the coordinates of the reference points in the old camera frame can be calculated as follows

$$\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}^o = {}^o\vec{P_i} = {}^oT_w \bullet {}^w\vec{P_i} = {}^oR_w \bullet \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}^o + \vec{t} \tag{5.35}$$

The three reference points with known coordinates will define a size-known object, which can be measured in the new camera frame by only one camera as described in section *mono-camera system*.

$$\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}^n = \delta_i \bullet \begin{pmatrix} X_i \\ Y_i \\ 1 \end{pmatrix}^n \qquad (i = 1, 2, 3) \tag{5.36}$$

With the 3D coordinates from both frames, one can get the following constraints over the transformation

$$\begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ z_1 & z_2 & z_3 \end{pmatrix}^o = {}^oR_n \bullet \begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ z_1 & z_2 & z_3 \end{pmatrix}^n + \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} \tag{5.37}$$

This is a *best-fit* problem between coordinate frames, which is solved well in the appendix.

5.5.4 Approach with arbitrary RPs

Mostly, the change to the camera pose happens by mistake or unconsciously, like a minor collision in cleaning, or little influence to working environment for installing a neighboring appliance, or people passing too close, and so on. The scale of the change is usually so small that the translation can be neglected. But the orientation, which is always the dominant influence, cannot be neglected. In this situation, a rough recalibration approach with 2 reference points is introduced to estimate the change in the orientation of the camera pose.

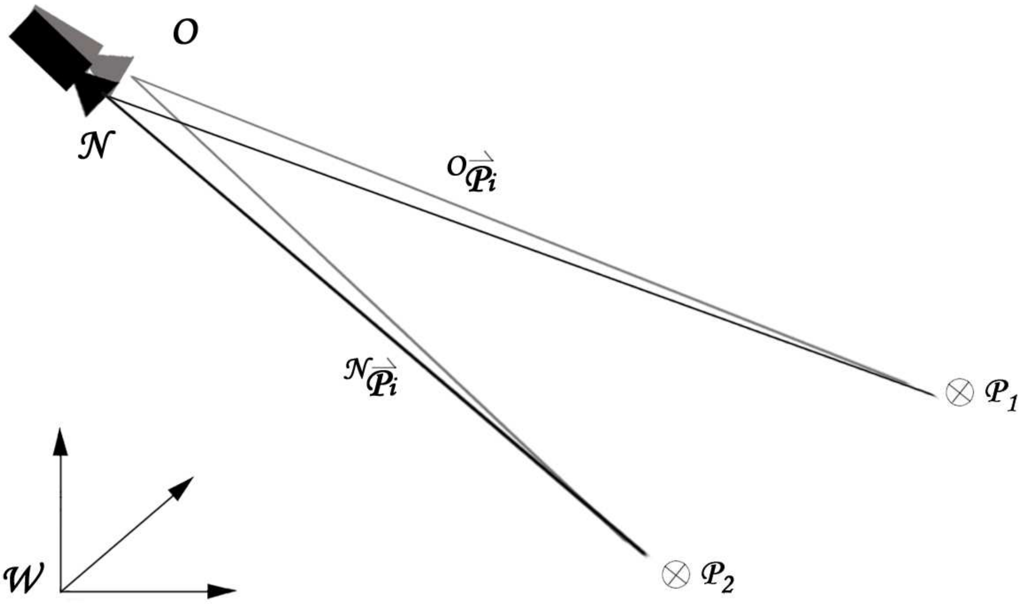


Figure 5.30: recalibration with 2 arbitrary reference points

As seen in figure 5.30, the two reference points are arbitrary points, whose coordinates in the world frame are fixed, but need not be known. What we have are the patterns coordinates in the camera images from the old and the new positions. From equation 5.1, we can construct the constraints between the two direction vectors in the old and new camera frames for each point

$$^{o}\delta \begin{pmatrix} ^{o}X_i \\ ^{o}Y_i \\ 1 \end{pmatrix} = \begin{pmatrix} ^{o}x_i \\ ^{o}y_i \\ ^{o}z_i \end{pmatrix} = (^{o}R_n, \vec{t}) \begin{pmatrix} ^{n}x_i \\ ^{n}y_i \\ ^{n}z_i \end{pmatrix} = {}^{o}R_n \bullet {}^{o}\delta_i \begin{pmatrix} ^{n}X_i \\ ^{n}Y_i \\ 1 \end{pmatrix} + \vec{t} \tag{5.38}$$

If the change in translation can be neglected, the origins of the two frames can be taken for granted that they are located in the same point. In other words, the two direction vectors have the same length. Making them into unit vectors, one gets

$$\frac{(^{o}X_i,\ ^{o}Y_i,\ 1)^T}{\sqrt{^{o}X_i^2 + {}^{o}Y_i^2 + 1}} = {}^{o}R_n \bullet \frac{(^{n}X_i,\ ^{n}Y_i,\ 1)^T}{\sqrt{^{n}X_i^2 + {}^{n}Y_i^2 + 1}} \tag{5.39}$$

If the orientation is written into RPY angles, the above relation will give two equations on α, β and γ. With two reference points, one can get an over-determined non-linear system for three unknowns. Since the change in the camera pose is very small, the system can be solved with an initial guess of all zero values.

5.5.5 Approach with groups of RPs

As seen from figure 5.31, there are two groups of reference points, whose coordinates in the world frame need not be known. Each group has at least three points, and the distances between points in the same group are already known. With these conditions, the complete correction for the camera pose can be estimated.

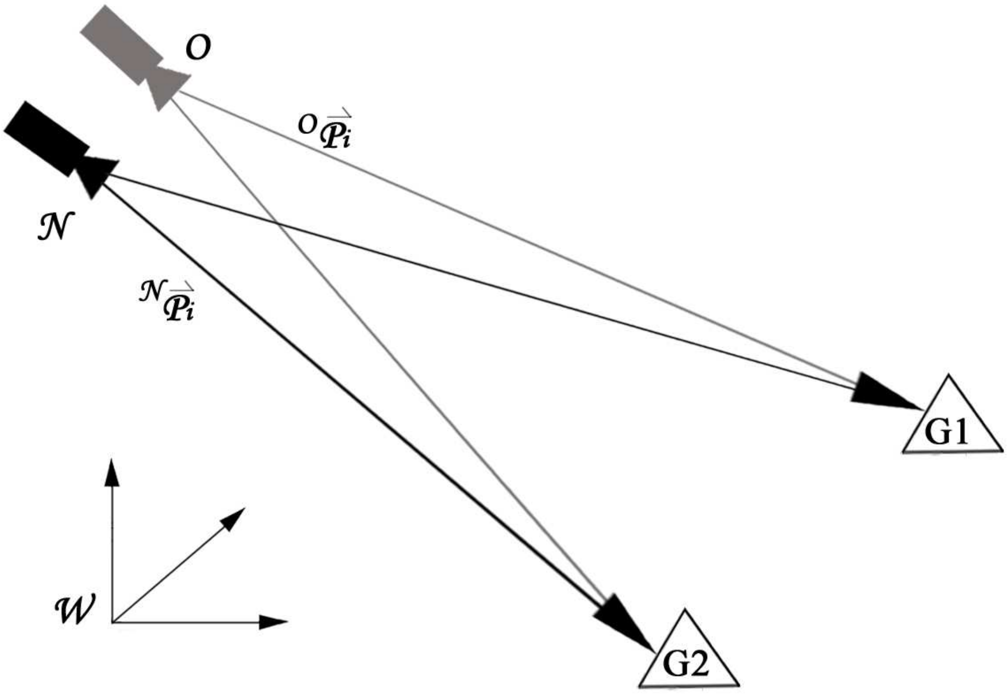

O
$^{O}\vec{\mathcal{P}_i}$
$\mathcal{N}$
$^{\mathcal{N}}\vec{\mathcal{P}_i}$
G1
G2
$\mathcal{W}$

Figure 5.31: recalibration with 2 groups of reference points

For every group, the reference points can be considered as the character points on an object. For the distances between the points are known, the object is a size-known object, which can be measured by only one camera with known fixed internal parameters

$$\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}^{(o,n)} = \delta_i \bullet \begin{pmatrix} X_i \\ Y_i \\ 1 \end{pmatrix}^{(o,n)} \tag{5.40}$$

Since the coordinates from the old frame and the new frame are known, the condition is suitable for recalibration with three known reference points. The scale factors as well as the orientation and translation are calculated and noted as follows

$$\left({}^{o}R_n,\ \vec{t},\ {}^{n}\vec{\delta} \right)_0 \tag{5.41}$$

However, there is an accuracy problem with the results from only one group. The group of reference points are located on a specially designed plate, which cannot be very large in practical applications. The fact that the reference points in the same group are too close to each other will cause an accuracy problem: small errors from recognition will lead large errors in the results. With two or more groups, which can be far away from each other, the result of the recalibration can be improved with the maxi-likelihood algorithm. Starting again from the wanted transformation, one can rewrite it into a functions form as follows

$$\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}^o = {}^oR_n \bullet {}^n\delta_i \begin{pmatrix} X_i \\ Y_i \\ 1 \end{pmatrix}^n + \vec{t} = \begin{pmatrix} M(R, \vec{t}, {}^nX_i, {}^nY_i, {}^n\delta_i) \\ N(R, \vec{t}, {}^nX_i, {}^nY_i, {}^n\delta_i) \\ L(R, \vec{t}, {}^nX_i, {}^nY_i, {}^n\delta_i) \end{pmatrix} \tag{5.42}$$

where M, N and L are three functions for translating the pattern coordinates from the new frame into the coordinates in the old frame. Replacing the coordinates in the old frame with the direction vector, one gets

$$^o\delta_i \begin{pmatrix} X_i \\ Y_i \\ 1 \end{pmatrix}^o = \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}^o = \begin{pmatrix} M(R, \vec{t}, {}^nX_i, {}^nY_i, {}^n\delta_i) \\ N(R, \vec{t}, {}^nX_i, {}^nY_i, {}^n\delta_i) \\ L(R, \vec{t}, {}^nX_i, {}^nY_i, {}^n\delta_i) \end{pmatrix} \tag{5.43}$$

By removing $^o\delta_i$, the above equations can be simplified into

$$\begin{cases} {}^oX_i \bullet L(R, \vec{t}, {}^nX_i, {}^nY_i, {}^n\delta_i) - M(R, \vec{t}, {}^nX_i, {}^nY_i, {}^n\delta_i) = 0 \\ {}^oY_i \bullet L(R, \vec{t}, {}^nX_i, {}^nY_i, {}^n\delta_i) - N(R, \vec{t}, {}^nX_i, {}^nY_i, {}^n\delta_i) = 0 \end{cases} \tag{5.44}$$

where the pattern coordinates X and Y are known, and stem from the pattern recognition in the procedure of image processing. Therefore, if we have six or more than six reference points, one will obtain an over-determined non-linear system, which will be solved with the Newton-Rapson algorithm, and the values in 5.41 can be taken as the initial guess.

5.5.6 Experimental results

The experiments are carried out with a JAI CV-M50 camera and a lens of 35 millimeters focal length. The camera is firstly calibrated with a calibration body and the reference points or groups are mounted. Then the camera pose is changed by hand 3 times, which are classified into *little change*, *small change* and *large change*. For each change, the camera is recalibrated with the calibration body and a laser tracker and the camera pose change is denoted as *laser tracker*. At the same time, the pose changes are also estimated with the approaches discussed in this section and the results are listed in table 5.9.

The values in table 5.9 show clearly that the approach with 2 reference points has a much worse accuracy on rotation estimation and the translation is neglected. However, at this moment only this method is integrated into the VMT system. The reasons are as follows

1. Firstly, the approach is easy to realize in practice, since it needs only 2 reference points, whose coordinates need not be known.

2. Secondly, online recalibration approaches are never used when the camera has a large change. When a small change is happened to the camera and the translation is really small, the accuracy for the approach to estimate the camera pose is enough for applications.

	x	y	z	rx	ry	rz
laser tracker	-2.84	1.28	0.75	0.092	0.02	0.006
2 reference points	—	—	—	0.051	0.048	-0.012
3 reference points	-1.98	2.06	1.21	0.109	0.029	-0.003
2 reference groups	-1.87	1.84	2.12	0.128	0.009	-0.007
laser tracker	9.86	1.32	-7.63	0.645	2.097	0.126
2 reference points	—	—	—	0.492	1.788	0.244
3 reference points	10.64	2.01	-7.08	0.597	2.021	0.187
2 reference groups	10.97	1.9	-7.16	0.603	2.015	0.191
laser tracker	42.04	388.92	-200.83	1.07	-1.654	8.047
2 reference points	—	—	—	2.142	-2.027	6.146
3 reference points	43.87	391.63	-198.76	1.128	-1.563	8.136
2 reference groups	43.64	390.75	-198.04	1.136	-1.547	8.132

Table 5.9: accuracy of the recalibration approaches

3. Finally, when the change to the camera pose is very small, the rotation deviations have the dominant influence in measurements and the influence from translation deviation can be neglected as shown in table 5.10, especially if we carry out a new zero measurement.

unit: mm / degree	x	y	z	rx	ry	rz
normal measure	-21.02	4.60	16.69	0.260	1.300	0.000
errors from x -0.5	-0.33	0.02	-0.54	-0.060	-0.020	0.010
errors from x+1.0	0.33	-0.08	0.62	0.110	0.010	-0.010
errors from x+2.0	0.68	-0.21	1.28	0.220	0.020	-0.020
errors from x+3.0	1.09	-0.31	1.98	0.330	0.040	-0.030
errors from rx-0.1	-2.83	2.44	1.77	-0.070	-0.450	0.050
errors from rx-0.2	-5.85	4.91	3.76	-0.160	-0.930	0.110

Table 5.10: different influences from translation and rotation

5.6 Chapter review

In this chapter, we discussed three types of vision systems: mono-camera system, stereo vision system and multi-camera system. Our research work on computer vision starts from the stereo vision system, the mono-camera system is referred because of a specific application, and our main target is on the multi-camera system.

5.6.1 Characters and applicable situations

The vision systems have different characters and are suitable for different types of applications.

1. Mono-camera system
   Since a single camera is not able to get the depth information from the camera image, a mono-camera system is usually used to determine the pose of the rigid objects with known patterns. The object to be measured must be small and the measurement distance must be short since the measurement accuracy is worse and worse as the measured object gets farther and farther away to the camera. A mono-camera system is used by us for verifying the mounting situation of the painting gun on the robot hand.

2. Stereo vision system
   A stereo vision system is composed of two cameras, which look the objects in parallel poses or with a fixed angle. The most dominant character is that the sights of the two cameras have a common part, where the measured objects locate. Then the stereo vision system can determine the 3D coordinates of any object point in the common sight. Since the common sight is relatively limited, the stereo vision system is sometimes mounted on a robot hand to extend the measurement range. Therefore, the stereo vision system is very flexible in measurements, such as absolute or relative coordinates of the specific object patterns, the pose of any rigid objects, and so on.

3. Multi-camera system
   A multi-camera system has more than two cameras, which may be installed far away from each other and have completely different sights. With the different camera looking different part of a large rigid object, the multi-camera system is usually designed for determining the pose of the object. The multi-camera system is often applied to robot vision applications, such as fine sealing in the painting shop, car doors or windows mounting in the assembly shop, and so on.

4. Recalibration of the camera pose
   Any vision system should keep its camera or cameras undisturbed so that it can do the measurements correctly. With some reference points in the camera sight, the proposed approaches to camera pose recalibration are able to check if the camera pose is disturbed and correct the camera pose if disturbed. These approaches are specially designed for a multi-camera system, that is why it is referred in this chapter.

Remark: In some situations, a single camera is mounted on a robot hand and the measurement is done by driving the robot into several positions for the camera to see the object. Although there is only one camera, each position can be taken as an independent *virtual* camera and we classify this configuration into a multi-camera system and not a mono-camera system.

5.6.2 Contributions

In this chapter, the following approaches or viewpoints have done valuable contributions in applying the vision systems efficiently and accurately to the industrial applications.

1. A mono-camera system is developed and its uncertainty in measurements are tested. Finally it is applied successfully to an industrial application.

2. Two types of configurations for a stereo vision system are set up and the accuracy in measurements is well verified.

3. The calibration concept for the stereo sensor is brought forward and solved in both mathematics and practice.

4. The applications of the stereo vision system are developed and tested in the laboratory.

5. The pattern compensation strategy has improved greatly the performances of the multi-camera system in practice: a. The mismatch possibility between 3D coordinates and 2D coordinates is decreased and the stability of the system is improved; b. The mistakes or deviations in pattern recognition may be informed; c. Through discarding the wrongly and badly recognized patterns, the result can be refined.

6. According to the working situations on site, a zero measurement procedure is introduced for producing the relative vector instead of the absolute vector. The zero measurement procedure improves the accuracy of the multi-camera system greatly in robot vision applications.

7. The pattern weight strategy enables the different kinds of patterns to contribute differently in resulting the final vector. It makes the multi-camera system work safer and more smoothly.

8. Some small points on security controlling are introduced to make the multi-camera system more suitable for the industrial applications.

9. To make the multi-camera system a more complete and qualified industrial product, the issue of the recalibration for the camera pose is also taken in consideration and three online approaches to the camera pose recalibration are proposed.

Chapter 6

Dissertation Review

As mentioned before, the aim of the dissertation is to apply the vision systems to the robot vision applications accurately and efficiently. After the camera model is determined, some camera calibration methods are thoroughly analyzed, and then some practical approaches to camera calibration are proposed, and finally some practical issues on applying the vision systems to the robot vision applications are brought forward and well solved. Actually, all approaches referred in this dissertation have roots in the practical projects and they outline briefly the research activities and the working experiences of the author during the past years.

6.1 Contributions of the dissertation

Although the exact contributory points have been referred in the chapter reviews, they are summarized here at a global viewpoint.

1. The deviations in pixel or millimeter for both the camera calibration and the pose measurement are introduced to verify the validity and accuracy of the calibration or measurement.

2. Some familiar but effective calibration methods for the stationary cameras are well discussed in the calibration chapter. After careful analysis and experiments in laboratory, their disadvantages are clearly pointed out as well as their valuable points. Finally, they are improved and completed by extending their applicable situations or determining their degenerate configurations.

3. The interactions between the camera parameters are sought hardly and an original opinion about that is brought boldly forward. Although it must be verified more in experiments and practices, it is a good attempt and start for the follow-up research on camera calibration.

4. To make full use of the advantages of the referred calibration methods and minimize their weaknesses, the combination solutions of the referred calibration methods are proposed and the appropriate iterative procedures are carefully used in practice.

5. Some practical approaches to camera calibration are proposed for the typical applications in robot vision, where the working environment is made the best of.

6. The proposed approaches to camera calibration are done at the beginning only for the real projects, but finally applied steadily in the industrial applications after many times of improvements in details.

7. To verify the mounting situation of a robot tool, a mono-camera system is introduced and its measurement task and applicable situations are defined. In fact, there are many such kinds of industrial applications may be settled down with a mono-camera system.

8. A set of accurate and elaborate calibration setup is used not only for calibrating the cameras but also for testing the accuracy of the system. For demonstrating how to apply the system to practice, a sensor calibration procedure is introduced and two applications with the stereo vision system is proposed.

9. The zero measurement and pattern compensation procedures have improved greatly the accuracy of the multi-camera system when it is applied to the robot vision applications.

10. By adopting the strategies of the pattern compensation, the pattern weight and the security controlling, the multi-camera system becomes a safe, stable and qualified vision system for the industrial applications.

11. The dissertation gets a separate sub-chapter for discussing the recalibration of the camera pose. Although only the partial recalibration approach is applied in the industrial applications, the attentions from more researchers may be raised on this issue in the near future.

12. Finally, the most important contribution of the dissertation is that the referred setups, opinions and attempts are caused from the practice and the proposed methods, approaches or procedures are to serve the industrial applications.

6.2 Open issues and future directions

In a view of vision systems in the industrial applications, the remaining open issues or the possible research directions in the future may be as follows

1. Complete recalibration of the camera pose
   Although three online approaches are proposed in this dissertation, one has gone through the trial of the applications but it corrects only the rotations change and neglects the shift changes. The other two approaches are of complete recalibration but have never been used in the applications. More attention and research work should be concentrated on this topic to get some stable and accurate solutions for the complete recalibration of the camera pose in practice.

2. Online calibration for the flexible cameras
   With the vision systems used more and more for the applications, some flexible cameras, such as rotating cameras and cameras with zoom lens or wide-angle lens, are needed sometimes in practice. Since the camera parameters are changing from time to time during the working period, the flexible cameras must be calibrated online, quickly and automatically.

3. Depth scenery rebuilding with a stereo vision system
   The stereo vision referred in this dissertation is only for determining the 3D coordinates of the object points, whose 2D pixels from both camera images must be paired exactly. In the real life, a stereo vision system is applied more often for rebuilding the depth scenery of the common sight, where the quickly and automatically pairs matching is a great challenge.

Generally speaking, all the possible future research work must be oriented by the practical applications.

Appendix A

System of Nonlinear Equations

Generally speaking, there are no very good methods to solve systems of more than one nonlinear equation. Unfortunately, problems about nonlinear systems are very common in practical computation. Therefore, the simplest multidimensional root finding method, the Newton-Rapson method, is here introduced. This method gives you a very efficient means of converging to a root, if you have a sufficiently good initial guess.

Let's think a typical problem with N functional relations to be zeroed as follows

$$F_i(\vec{x}) = 0 \quad where \quad \vec{x} = (x_1, x_2, \cdots, x_N) \tag{A.1}$$

Giving an intial guess $\vec{x}_0$, the functions F_i can be expanded in Taylor series in the neighborhood of $\vec{x}_0$

$$F_i(\vec{x}_0 + \delta\vec{x}) = F_i(\vec{x}_0) + \sum_{j=1}^{N} \frac{\partial F_i}{\partial x_j} \delta x_j + O(\delta\vec{x}^2) \tag{A.2}$$

The matrix of the partial derivations from above equation is usually called Jacobian matrix **J**:

$$J_{ij} = \frac{\partial F_i}{\partial x_j} \tag{A.3}$$

Then the Taylor series can be written into matrix notation

$$F(\vec{x}_0 + \delta\vec{x}) = F(\vec{x}_0) + J \cdot \delta\vec{x} + O(\delta\vec{x}^2) \tag{A.4}$$

Neglecting the terms of order $\vec{x}^2$ and higher and setting $F(\vec{x}_0 + \delta\vec{x}) = 0$, the above equations will become a set of linear equations for the correction $\delta\vec{x}$, which makes all the funtions closer to zero simutanoeusly, namely

$$F(\vec{x}_0) + J \cdot \delta\vec{x} = 0 \quad \Longrightarrow \quad \delta\vec{x} = -J^{-1} \cdot F(\vec{x}_0) \tag{A.5}$$

where J^{-1} can be obtained by a SVD or LU decomposition. Since $\delta\vec{x}$ makes the functions closer to zero, a better set of solution yields

$$\vec{x}_{new} = \vec{x}_{old} + \delta\vec{x} \tag{A.6}$$

The above process will be iterated till the system converges, which can be found by checking if both the function and the solution have converged.

However, the process may fail to converge, the reason in most of time is that the initial guess is not good enough if you are sure the root exists. A more sophisticated implementation of the New-Rapson method, which tries to improve the poor global convergence from Newton-Rapson, can also be found in [1].

Appendix B

R-Matrix Orthonormalization

A rotation matrix must be an orthonormal matrix. However, in many cases its elements are computed one after another independently, which makes the matrix satisfy this property not so well. An orthonormalization procedure can help us to get better accuracy.

The orthonormalization procedure is to approximate a best rotation matrix R from a given 3×3 matrix Q. In general case, the *best* means in the sense of the smallest Frobenius norm of the difference $R - Q$. Then the problem is as follows:

$$min(\parallel R - Q \parallel_F^2) \quad where \quad R^T R = I \tag{B.1}$$

Thinking the properties of the Frobenius norm of a matrix, one has

$$\begin{aligned} \parallel R - Q \parallel_F^2 &= trace((R-Q)^T(R-Q)) \\ &= 3 + trace(Q^T Q) - 2trace(R^T Q) \end{aligned} \tag{B.2}$$

Since $trace(Q^T Q)$ is fixed with the given matrix, the above minimum problem becomes the following maximum problem

$$min(\parallel R - Q \parallel_F^2) \quad \Longleftrightarrow \quad max(trace(R^T Q)) \tag{B.3}$$

Applying the singular value decomposition to Q: $Q = USV^T$, then

$$trace(R^T Q) = trace(R^T USV^T) = trace(V^T R^T US) \tag{B.4}$$

Defining $Z = V^T R^T U$, then Z is obviously an orthogonal matrix

$$trace(R^T Q) = trace(ZS) = \sum z_{ii} s_{ii} \leq \sum s_{ii} = trace(IS) \tag{B.5}$$

The equal case from above equation comes, or $trace(R^T Q)$ achieves maximum, if and only if

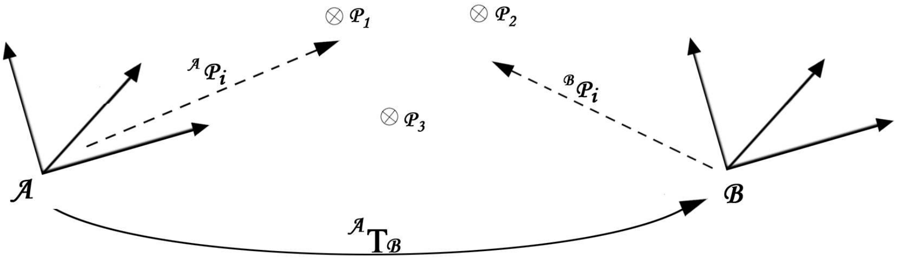
$$Z = V^T R^T U = I \quad \Longrightarrow \quad R = UV^T \tag{B.6}$$

Therefore, the above R is the solution for B.1, the destination of the orthonormalization procedure.

Appendix C

Best-Fit between Coordinate Frames

A common issue in practice is to determine the transformation between coordinate frames. In most of time, the coordinate frames are defined separately, which makes it difficult to solve the problem with any direct method. By measuring the coordinates of some object points with respect to both frames, the issue can be solved efficiently in mathematics. The calculation procedure with coordinates of object points to determine the transformation between coordinate frames is usually called *best-fit*.



$\otimes \mathcal{P}_1$
$\otimes \mathcal{P}_2$
$^{\mathcal{A}}\mathcal{P}_i$
$^{\mathcal{B}}\mathcal{P}_i$
$\otimes \mathcal{P}_3$
$\mathcal{A}$
$\mathcal{B}$
$^{\mathcal{A}}\mathbf{T}_{\mathcal{B}}$

Figure C.1: best-fit between coordinate frames

From figure C.1, the task of the best-fit is as follows: **with the coordinates of n ($n \geq 3$) non-collinear points $^{A}p_i$ and $^{B}p_i$ with respect to both frames, the transformation $^{A}T_B$ is needed to be determined**.

C.1 Solving the *best-fit*

Denote the $^{A}T_B$ into rotation R and translation $\vec{t} = (x, y, z)^T$, then

$$R \cdot \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}^B + \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}^A \tag{C.1}$$

If the rotation matrix R is denoted into RPY form as equation 3.11, the above equation can outspread into 3 nonlinear equations, which can be zeroed as follows

$$\left\{ \begin{array}{l} f_x(\alpha, \beta, \gamma, x, y, z) = 0 \\ f_y(\alpha, \beta, \gamma, x, y, z) = 0 \\ f_z(\alpha, \beta, \gamma, x, y, z) = 0 \end{array} \right. \tag{C.2}$$

n ($n \geq 3$) such points can form an overdetermined nonlinear system with 3n equations, which will be solved by the Newton-Rapson method with a set of start values. At this moment, it is not easy to get a good set of start values. Thus we try to prove that the system has an exclusive solution and always converges to the right solution if it converges.

C.2 Exclusive solution

Since the n points are non-collinear, there are only two situations: coplanar or non-coplanar. In both situations the *best-fit* has a unique solution.

C.2.1 Coplanar points

Without loss of generality, the following prove procedure is demonstrated with 3 non-collinear points. Firstly, we shift the frame A so that the origin of A locates in the plane π, which is determined by the different points $\vec{p}_1, \vec{p}_2, \vec{p}_3$. By abuse of notation, ${}^A\vec{p}_i$ denotes again the new coordinates of the points in A. Since the origin of frame A locates in π, there must have such 3 factors to satisfy

$$a \cdot {}^A\vec{p}_1 + b \cdot {}^A\vec{p}_2 + c \cdot {}^A\vec{p}_3 = 0 \tag{C.3}$$

The fact that $\vec{p}_1, \vec{p}_2$ and $\vec{p}_3$ are non-collinear makes $a + b + c \neq 0$. Otherwise, suppose $a + b + c = 0$, then

$$\begin{array}{ll} & c = -(a+b) \\ \Longrightarrow & a \cdot ({}^A\vec{p}_1 - {}^A\vec{p}_3) + b \cdot ({}^A\vec{p}_2 - {}^A\vec{p}_3) = 0 \\ \Longrightarrow & a \cdot \overrightarrow{p_3p_1} + b \cdot \overrightarrow{p_3p_2} = 0 \end{array} \tag{C.4}$$

The above equation states that p_1, p_2, p_3 are collinear, which conflicts with the precondition. Therefore, $a + b + c \neq 0$ satisfies.
Suppose that there are 2 frames B_1 and B_2 to satisfy the preconditions and denote $[R_1, \vec{t}_1]$ and $[R_2, \vec{t}_2]$ respectively the transformations from B_1 and B_2 to A, then

$$\begin{array}{ll} & R_1 \cdot {}^A\vec{p}_i + \vec{t}_1 = {}^B\vec{p}_i = R_2 \cdot {}^A\vec{p}_i + \vec{t}_2 \quad i = 1,2,3 \\ \Longrightarrow & (R_1 - R_2) \cdot {}^A\vec{p}_i + (\vec{t}_1 - \vec{t}_2) = 0 \qquad i = 1,2,3 \\ \Longrightarrow & (R_1 - R_2) \cdot (a \cdot {}^A\vec{p}_1 + b \cdot {}^A\vec{p}_2 + c \cdot {}^A\vec{p}_3) + (a+b+c)(\vec{t}_1 - \vec{t}_2) = 0 \\ \Longrightarrow & (a+b+c)(\vec{t}_1 - \vec{t}_2) = 0 \\ \Longrightarrow & \vec{t}_1 = \vec{t}_2 \end{array} \tag{C.5}$$

If denote $R_2^{-1} R_1 = R$, then

$$\begin{array}{ll} & R \cdot [{}^A\vec{p}_1, {}^A\vec{p}_2, {}^A\vec{p}_3] = [{}^A\vec{p}_1, {}^A\vec{p}_2, {}^A\vec{p}_3] \\ \Longrightarrow & R = R_2^{-1} R_1 = I \\ \Longrightarrow & R_1 = R_2 \end{array} \tag{C.6}$$

The above procedure indicates that the solution for the best-fit between coordinate frames is exclusive and the system C.2 can be solved with an arbitrary set of start values.

C.2.2 Non-coplanar points

Specially, if the points $\vec{p}_i$ are non-coplanar, the method described above works of course well, but there is a simpler method to determine the *best-fit* transformation.
Suppose there are n ($n \geq 4$) non-coplanar points and denote $\vec{p}_i$ their homogeneous coordinates, one has

$$^{A}T_{B} \cdot ({}^{B}\vec{p}_1, {}^{B}\vec{p}_1, \cdots, {}^{B}\vec{p}_n) = ({}^{A}\vec{p}_1, {}^{A}\vec{p}_1, \cdots, {}^{A}\vec{p}_n) \tag{C.7}$$

Since $\vec{p}_i$ are non-coplanar, the order of the matrix $({}^{B}\vec{p}_1, {}^{B}\vec{p}_1, \cdots, {}^{B}\vec{p}_n)$ must be 4 and its inverse matrix exists, thus

$$^{A}T_{B} = ({}^{A}\vec{p}_1, {}^{A}\vec{p}_1, \cdots, {}^{A}\vec{p}_n) \cdot ({}^{B}\vec{p}_1, {}^{B}\vec{p}_1, \cdots, {}^{B}\vec{p}_n)^{-1} \tag{C.8}$$

Appendix D

Distortion Alignment

Although distortion, only the radial distortion is referred here, is unavoidable for all camera lenses, we have proposed some approaches to camera calibration for a distortion-free model. If the distortion factor k is given, a distortion alignment can be applied and then the approaches can be applied more efficiently, or sometimes just for testing. The distortion alignment can be applied either to the camera images or the coordinates of the calibration points obtained in pattern recognition.

D.1 Alignment to camera images

Given a distorted image from the camera, an undistorted image is obtained by applying a distortion alignment to the distorted image. Denote $P_i(X_i, Y_i)$ the pixel gray at col X_i row Y_i from the undistorted image and $P_r(X_r, Y_r)$ the pixel gray at col X_r row Y_r from the distorted image. Thinking of the radial distortion with equation 2.15, one has

$$P_i(X_i, Y_i) = P_r(X_r, Y_r) \tag{D.1}$$

where

$$(X_r, Y_r) = (1 + k \cdot (X_i^2 + Y_i^2)) \cdot (X_i, Y_i) \tag{D.2}$$

Since the scale factor k is a float value, the calculated pixel index (X_r, Y_r) has also float values, which makes the pixel $P_r(X_r, Y_r)$ cannot be detected from the distorted image. Thus an interpolation method is needed and its gray value can be easily determined in practice from its four neighborhoods as follows

$$\begin{aligned} P_r(X_r, Y_r) \quad = \quad & P_r([X_r], [Y_r]) \cdot \nabla X_r \cdot \nabla Y_r + \\ & P_r([X_r] + 1, [Y_r]) \cdot \Delta X_r \cdot \nabla Y_r + \\ & P_r([X_r], [Y_r] + 1) \cdot \nabla X_r \cdot \Delta Y_r + \\ & P_r([X_r] + 1, [Y_r] + 1) \cdot \Delta X_r \cdot \Delta Y_r \end{aligned} \tag{D.3}$$

where

$$\begin{aligned} \Delta X_r = X_r - [X_r] \quad \Delta Y_r = Y_r - [Y_r] \\ \nabla X_r = 1 - \Delta X_r \quad \nabla Y_r = 1 - \Delta Y_r \end{aligned}$$

D.2 Alignment to pattern coordinates

In fact, the undistorted image is not necessary. What we need are the undistorted coordinates of the calibration points, which are the exact values for the estimation in the following calibration procedure.
With equation 2.16, the undistorted coordinates $P_i(X_i, Y_i)$ can be restored from the results $P_r(X_r, Y_r)$ of pattern recognition below

$$(X_i, Y_i) = \frac{(X_r, Y_r)}{1 + k \cdot (X_r^2 + Y_r^2)} \tag{D.4}$$

Since the calibration points are relatively fewer, it is much more efficient to apply the distortion alignment to the pattern coordinates.

Remark: the distortion factor k referred above is the original one, not the factor defined with $K = k \cdot f^2$. In practice it is easy to check it by verifying $k \ll 1$.

Appendix E

The Applied Camera

The cameras used in the experiments and applications referred in this dissertation are from a type of JAI CV-M50, which is a monochrome CCD camera designed for industrial applications and featured with high performance and unique functions and has a uniform and compact housing, which can be seen in figure E.1

Figure E.1: CV-M50 camera from JAI

The main features of CV-M50 cameras are as follows

- CCD sensor: monochrome 1/2" interline
- Sensing area: 6.6 (h) × 4.8 (v) mm
- Cell size: 0.0086 (h) × 0.0083 (v) mm
- Effective pixels: 752 (h) × 582 (v) pixels
- Scanning system: 625 lines and 25 frames/second
- Sensitivity on sensor: minimum 0.05lux illumination
- S/N ratio: better than 56 dB
- Gamma: 0.45 - 1.0
- Gain: manual - automatic. 0 to +15 dB by potentiometer or AGC
- Accumulation: field - frame
- Lens mount: C-mount
- Power: 12V DC ±10% and 2.5W
- Dimension: 40×50×80 mm (H/W/D)
- Mass: 230 g

$\star$ *the above features are for CCIR mode*

To digitize the video signals, we have introduced the frame grabber PX510/610, which can be connected to maximum 4 such cameras and outputs digitalized images with dimension of 768 (h) × 572 (v) pixels.

Although the cameras with better resolution, such as SONY XC-HR70, JAI CV-A1 and CV-A2, are applied to inspection applications in machine vision, VMT GmbH takes JAI CV-M50 cameras as the standard configuration for robot vision applications with multi-camera system during the past five years.

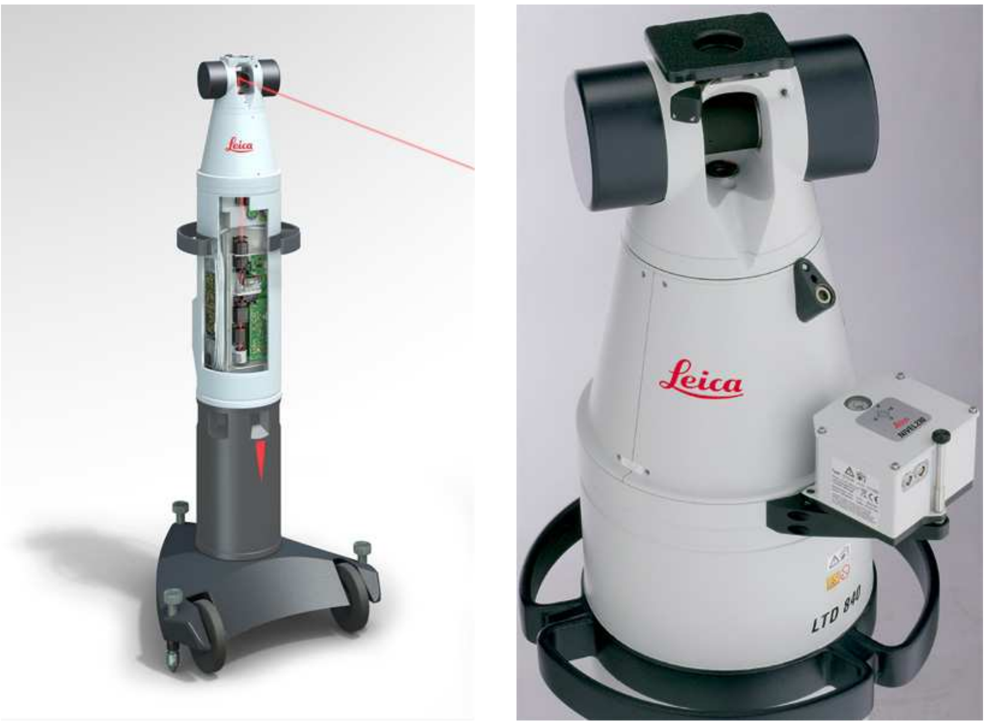
Appendix F

A Laser Tracker System

For applications with accurate measurement, a mobile laser tracker is widely applied to industries, especially in automobile industry.

F.1 Leica laser tracker

The laser tracker we have used by TecMedic GmbH is from Leica Geosystems AG, a company in Switzerland and is shown in figure F.1.

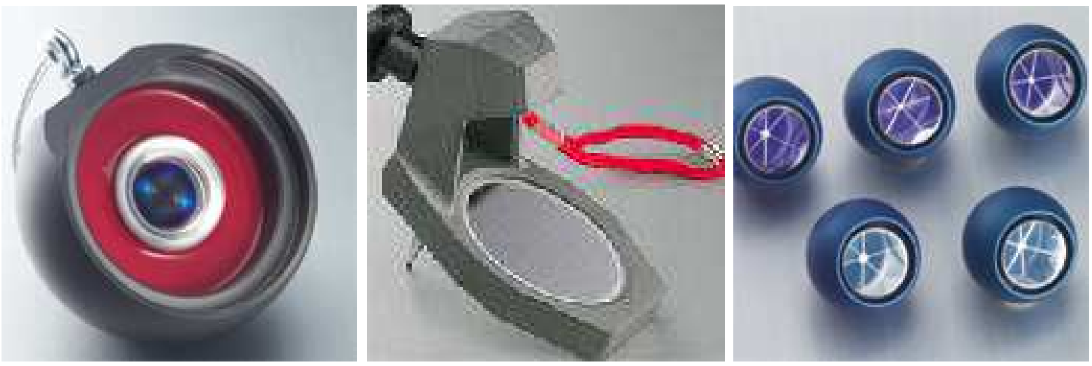


(a) overview

(b) detail

Figure F.1: Leica laser tracker LTD 840

A laser tracker uses of course the laser light to do the measuring tasks. The Leica laser tracker can measure the 3D coordinates of any point the laser light can reach with respect to the base coordinate frame, which is defined in the laser body. To do that, a laser reflector is needed. For different applications and environments, different reflectors can be utilized. The figure F.2 shows the most common reflectors for a Leica laser tracker.



(a) cateye (b) thumb (c) others

Figure F.2: laser reflectors for a laser tracker

In fact, the Leica measurement system is equipped with many other accessories, such as gradienter, meters for temperature and air pressure, kinds of adapters, and so on.

F.2 Technical parameters

The technical parameters of the laser tracker are profiled as follows

1. Measuring range
   Maximal distance: 40 meters; horizontal / vertical angles: $360^o / \pm 45^o$.

2. Measuring accuracy
   Distance resolution: 0.001 mm; distance repeatability: ± 0.012 mm; distance absolute accuracy: 0.025 mm.

3. Measuring rate
   Measuring rate: 3000 points per second; measuring rate output: 1000 points per second.

4. Tracking speed
   Lateral: $>$ 4 meters per second; radial: $>$ 6 meters per second.

More details can be found at www.leica-geosystems.com.

F.3 Frame determination

As referred in the last section, a laser tracker can measure accurately the 3D coordinates of any point it can *see*. With a *best-fit* procedure, the laser tracker can determine any object frame as follows

1. Features on the object
   The object to be measured must be a rigid body and the object frame is defined by its features, whose coordinates in the object frame must be given.

2. Features in the laser system frame
   The referring features can be measured with the laser tracker, that is to say, the coordinates of the features in the laser system frame are determined.

3. Object frame in the laser system frame
   If more than 3 features are so measured, a *best-fit* can be applied for determining the object frame with respect to the laser system frame.

4. Transformation between object frames
   If more objects are measured with the laser system, the objects' frames are determined with respect to the laser system frame and the transformation between object frames are known.

Remark: network orientation of the laser tracker

With the above determining procedure, all measuring work are completed with regard to the laser system, or the laser tracker body. In order to continue the measuring work when the laser tracker moves, a routine called *network orientation* is integrated in the laser software. When the laser tracker is set up in position A, some fix points are measured; when the laser tracker moves to position B, some of the fix points (at least 3) are measured again and the transformation between the laser tracker frames in both positions can be determined with a *best-fit* procedure. What is more, when the *network orientation* procedure is completed, the system will transfer the coordinate values from the following measuring automatically into values with respect to the first base frame, and the operator can continue the work just like nothing happened to the measurement system.

F.4 Applications with a laser tracker

With the help of the laser tracker, TecMedic GmbH has developed the following applications in automobile industry

1. Camera calibration
   In our practical approaches to camera calibration in working area, the cameras are mostly calibrated with a laser tracker to determine the camera pose with regard to our reference frame. The details have been described in the relevant chapters of the dissertation.

2. Determination of work objects
   To initialize a vision system, the work object must be measured initially at the ideal position. When the work object is too large, such as a car body, its translation and orientation can be determined by a laser tracker.

3. Determination of robot base
   All robots to be navigated by a vision system must be measured with respect to the world frame, which is known to the vision system. Only after that, the vision system can transform correctly the correction of the work object for the robots.

4. Verification of robot linear axis
   All robots must be calibrated after its installation in the working area. The arm

axes can be self-calibrated, and the linear axis needs an external measurement system to carry out the calibration.

5. Mounting and adjusting the working tools
   In *body in white* shop of automobile factory, a laser tracker is more and more often applied for adjusting or mounting the working tools, whose accurate relative poses will result qualified car bodies. TecMedic GmbH has done many such kinds of work in Opel Zaragoza, Spain through the integrator FFT GmbH.

Actually, all the calibration and measurement approaches proposed in the paper are tested in laboratory with a laser tracker.

Bibliography

[1] Numerical Recepies in C++ — Second Edition
W.H. Press, S.A. Tenkolsky, W.T., Vetterling, B.P. Flannary
Cambridge University Press, 2002.

[2] Geometry and Analysis of Projective Spaces
C. E. Springer
Freeman, 1964

[3] Close-range camera calibration
D. C. Brown
Photogrammetric Engineering, 37(8), 1971, pp. 855-866

[4] Algebraic Projective Geometry
J. Semple, G. Kneebone
Oxford University Press, 1979

[5] An improved adaptive stereo ranging method for 3D Measurements
L. A. Gerhard, W. I. Kwak
Proceedings CVPR'86, Miami Beach, Florida, June 1986, pp. 21-26

[6] Depth from three camera stereo
M. Pietikainen, D. Harwood
Proceedings CVPR'86, Miami Beach, Florida, June 1986, pp. 2-8

[7] An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision
R. Y. Tsai
Proceedings CVPR'86, Miami Beach, Florida, June 1986, pp. 364-374

[8] A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses
Roger Y. Tsai, IBM T.J. Watson Research Center
IEEE Journal of Robotics and Automation, VOL. RA-3, No.4, August, 1987

[9] A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf Cameras and Lenses
R. Y. Tsai
IEEE Journal of Robotics and Automation, vol. RA-3, No.4, August 1987, pp. 323-344

[10] Camera Calibration for 3D Computer Vision
O. Faugeras, G. Toscani
Proceedings of International Workshop on Industrial Applications of Machine Vision and Machine Intelligence, Tokyo, Japan, 1987, pp. 240-247

[11] Techniques for Calibration of the Scale Factor and Image Center for High Accuracy 3D Machine Vision Metrology
R. K. Lenz, R. Y. Tsai
IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 10, no. 5, September 1988, pp. 713-720

[12] Using vanishing points for camera calibratio
B. Caprile, V. Torre
International Journal of Computer Vision, vol. 4, no. 2, March 1990, pp. 127-140

[13] Robot vision modeling - camera modeling and camera calibration
M. Ito
dvanced Robotics, vol.5, no.3, 1991, pp. 321-335

[14] Camera Calibration: A quick and easy way to determine the scale factor
M. A. Penna
IEEE Transactions Pattern Analysis and Machine Intelligence, no. 13, 1991, pp. 1240-1245

[15] Camera Calibration with Distortion Models and Accuracy Evaluation
J. Weng, P. Cohen, M. Hernou
IEEE Transactions on Pattern Analysis and Machine Intelligence, VOL 14, No.10, October, 1992

[16] Estimation of relative camera position for uncalibrated cameras
R. Hartley
Proceedings of the 2nd European Conference on Computer Vision, Santa Margherita, Italy, May 1992, pp. 579-587

[17] Camera self-calibration: theory and experiments
O. Faugeras, T. -Q. Luong, S. Maybank
Proceedings of the 2nd European Conference on Computer Vision, Santa Margherita, Italy, May 1992, pp. 321-334

[18] Three dimensional computer vision
O. Faugeras
Massachusetts Institute of Technology, London, 1993

[19] Camera calibration Issues in Robot Calibration Eye-on-hand Configuration
H. Zhuang, Z. S. Roth, X. Xu, K. Wan
Robotics and Computer-Integrated Manufacturing, vol. 10, no. 6, 1993, pp 401-412

[20] A complete two-plane camera calibration method and experimental comparisons
G. -Q. Wei, S. D. Ma
Proceedings of 4th International Conference on Computer Vision, Berlin, May 1993, pp. 439-446

[21] Active Calibration: Alternative Strategy and Analysis
A. Basu
Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, New York, USA, June 1993, pp. 495-500

[22] On the Sensitivity of Camera Calibration
Jim Z. C. Lai
Image and Vision Computing Volume 11, Number 10, December 1993

[23] Self-Calibration from Multiple Views with a Rotating Camera
R. Hartley
Proceedings of 3rd European Conference on Computer Vision, Stockholm, vol. 1, 1994, pp. 471-478

[24] A stereo matching algorithm with an adaptive window: theory and experiments
T. Kanade, M. Okutomi
IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 16, 1994, pp. 920-932

[25] Implicit and explicit camera calibration: Theory and experiments
G. -Q. Wei, S. D. Ma
IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 16, October 1994, pp. 469-480

[26] Large accurate internal camera calibration using rotation with analysis of sources of error
G.P. Stein
5th International Conference on Computer Vision, 1995, pp. 230-236

[27] A self-calibration approach to extrinsic parameters estimation of stereo cameras
H. Zhuang
Robotics and Automation Systems, vol. 15, 1995, pp. 189-197

[28] Metric calibration of a stereo rig
A. Zissermann, P. Beardsley, I. Reid
IEEE Workshop on Representation of Visual Scenes, Boston 1995

[29] Stratification of three-dimensional vision: projective, affine, and metric representation
O. Faugeras
Journal of Optical Society of America, 1995, pp. 465-484

[30] The Fundamental Matrix: Theory, Algorithms, and Stability Analysis
Q. -T. Luong, O. Faugeras
International Journal of Computer Vision, vol. 17, no. 1, 1996, pp. 43-76

[31] Camera-Aided Robot Calibration
H. Zhuang, Zvi S. Roth
CRC Press, 1996.

[32] Self-Calibration from Image Sequences
M. N. Armstrong
PhD, University of Oxford, 1996

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

[34] Self-calibration of a moving camera from point correspondence and fundamental matrix
Q.-T. Luong, O. Faugeras
International Journal of Computer Vision 22(3), 1997, pp. 261-289

[35] Self-Calibration of Stationary Cameras
R. Hartley
International Journal of Computer Vision, vol. 22, no. 1, February 1997, pp. 5-23

[36] Lens distortion calibration using point correspondence
G. P. Stein
IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Juan, Puerto Rico, June 1997, pp. 602-608

[37] Robust Camera Calibration using 2D-to-3D feature correspondence
F. Dornaika, C. Garcia
Proceedings of Videometrics V - SPIE's Optical Science, Engineering and Instrumentation'97, August 1997, San Diego, pp. 123-133

[38] A Three-Step Camera Calibration Method
H. Bacakoglu, M. S. Kamel
IEEE Transactions on Instrumentation and Measurement, vol. 46, no. 5, October 1997, pp. 1165-1171

[39] Efficient parallel processing for depth calculation using stereo
J. Majumdar, Seethalakshmy
Robotics and Automation Systems, vol. 20, 1997, pp. 1-13

[40] Iterative Multi-Step Camera Calibration
J. Batista, H. Araujo, A. T. de Almeida
Proceedings of the 6th International Conference on Computer Vision, 1998, pp. 709-714

[41] A Multi-resolution Stereo Vision System for Mobile Robots
L. Iocchi, K. Konolige
Proceedings of the AIIA'98 Workshop on New Trends in Robotics Research, pp.317-321

[42] Camera calibration and the search of infinity
R. Hartley, E. Hayman, L. de Agapito, I. Reid
Proceedings of 7th International Conference on Computer Vision, Kerkyra, Greece, 1999, pp. 510-517

[43] A simple technique for self-calibration
P. Mendonca, R. Cipolla
Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Fort Collins, Colorado, USA, June 1999, pp. 112-116

[44] Camera calibration from vanishing points in images of architectural scenes
R. Cipolla, T. W. Drummond, D. Robertson
Proceedings of British Machine Vision Conference, vol. 2, September 1999, UK, pp. 382-391

[45] Flexible Calibration: Minimal Cases for Auto-Calibration
A. Heyden, K. Astro
Proceedings of 7th International Conference on Computer Vision, Kerkyra, Greece, 1999, pp. 350-355

[46] Stereo vision for industrial applications
A. Kosaka, A. C. Kak
Handbook of Industrial Robotics, Second Edition, John Wiley Sons, New York, 1999, pp.269-294.

[47] Camera self-calibration using the singular value decomposition of the fundamental matrix: From point correspondence to 3D measurements
M. I. A. Lourakis, R. Deriche
Technical Report 3748, INRIA Sophia-Antipolis, August 1999

[48] A Flexible New Technique for Camera Calibration
Z. Zhang
IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000

[49] Camera self-calibration using the singular value decomposition of the fundamental matrix
M. I. A. Lourakis, R. Deriche
Asian Conference on Computer Vision, Taiwan, January 2000, vol. 1, pp. 403-408

[50] A comparative review of camera calibrating methods with accuracy evaluation
X. Armangue, J. Salvi, J. Balle
Proceedings of 5th Ibero-American Symposium on Pattern Recognition, SIAPR 2000, Lisboa, Portugal, pp. 183-194

[51] Straight Lines Have to Be Straight
F. Devernay, O. Faugeras
Machine Vision and Applications, vol. 13, no. 1, 2001, pp. 14-24

[52] The Geometry of Multiple Images
O. Faugeras, T. -Q. Luong
Massachusetts Institute of Technology, London, 2001

[53] Analytical Geometry (chinese) — Second Edition
W. Qiu
Press Beijing University, 2001

[54] Multiview Geometry: Profiles and Self Calibration
P. Mendonca
PhD, University of Cambridge, 2001

[55] Estimating 3D Pose using Uncalibrated Cameras
R. Rosales, M. Siddiqui, J. Alon, S. Sclaroff
Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'01), vol. 1, 2001, pp. 821-827

[56] A Three-stage system for camera calibration
Q. Wu, G. Xu, L. Wang
SPIE Conference Proceedings: Multispectral Image Processing and Pattern Recognition, 2001, pp. 22-24

[57] Simple and Accurate Camera Calibration
A. Cumani
IEN Technical Report no. 631, July 2001

[58] Minimalist Camera Calibration
A. Cumani
IEN Technical Report no. 641, December 2001

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

[60] A simple camera calibration method
A. Cumani
Proceedings of the 2nd WSEAS Conf. Signal Processing, Computational Geometry and Vision, Rethymno 2002

[61] An world-independent approach for the calibration of mobile robotics active stereo heads
G. N. DeSousa, A. H. Jones, A. C. Kak
Proceedings of IEEE International Conference on Robotics and Automation (ICRA'02), vol. 4, 2002, pp. 3336-3341

[62] A New Easy Camera Calibration Technique Based on Circular Points
X. Meng, H. Li, Z. Hu
Journal of Software, vol.13, No.5, 2002

[63] Some Improvements on Two Auto-calibration Algorithms from the Fundamental Matrix
G. Roth, A. Whitehead
Proceedings of 16th International Conference on Pattern Recognition (ICPR'02), vol. 2, Quebec City, Canada, August 2002, pp. 312-315

[64] Multiple View Geometry in Computer Vision — Second Edition
R. Hartly, A. Zisserman
Cambridge University Press, 2003.

[65] Robust line-based calibration of lens distortion from single view
T. Thormaehlen, H. Broszio, I. Wassermann
Proceedings of Mirage 2003, INRIA Rocquencourt, France, March 2003, pp. 105-112

[66] Statistically Robust Approach to Lens Distortion Calibration with Model Selection
M. T. El-Melegy, A. A. Farag
Conference on Computer Vision and Pattern Recognition Workshop, June 2003, vol. 8, pp 91-99

[67] Robust correspondence methods for stereo vision
M. P. Eklund, A. A. Farag
International Journal of Pattern Recognition and Artificial Intelligence, 2003, vol. 17, no. 7, pp. 1059-1079

[68] A Linear Approach for for Determining Intrinsic Parameters and Pose of Cameras from Rectangles
F. Wu, G. Wang, Z. Hu
Journal of Software, vol.14, No.3, 2003

[69] Camera Calibration with One-Dimension Objects
Z. Zhang
IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 7, July 2004, pp. 892-899

[70] Calculating the 3D-Pose of Rigid Objects Using Active Appearance Models
P. Mittrapiyanuruk, G. N. DeSouza, A. C. Kak
Proceedings of International Conference in Robotics and Automation, 2004

[71] A Technique for Binocular Stereo Vision System Calibration by the Nonlinear Optimization and Calibration Points with Accurate Coordinates
H Chen, D Ye, R S Chen and G Chen
Journal of Physics: Conference Series 48 (2006) 806 - 810

[72] Camera calibrationless vision calibration for transfer robot system
Nobutaka Kimura, Toshio Moriya, and Kohsei Matsumoto
Proceedings of International Conference on Machine Vision Applications in Industrial Inspection XV, 2007.

[73] Self-calibration of environmental camera for mobile robot navigation
Huiying Chen, Kohsei Matsumoto, Jun Ota, Tamio Arai
Proceedings of International Conference on Robotics and Autonomous Systems archive, Volume 55, Issue 3, 177-190 (March 2007)

[74] Development of a stereo vision system for industrial robots
Jong-Kyu Oh; Chan-Ho Lee
International Conference on Control, Automation and Systems, 2007.

[75] Calibration of inertial and vision systems as a prelude to multi-sensor fusion
D.I.B. Randeniya, M. Gunaratne, S. Sarkar, A. Nazef
Transportation Research Part C16 (2008) 255 - 274

[76] Calibration Algorithm for Structured Light 3D Vision Measuring System
Haihua Cui, Ning Dai, Tianran Yuan, Xiaosheng Cheng, Wenhe Liao
Congress on Image and Signal Processing, Vol. 2, pp.324-328, 2008

[77] Integration of 3D Stereo Vision Measurements in Industrial Robot Applications
Frank Cheng and Xiaoting Chen
Proceedings of The 2008 IAJC-IJME International Conference, ISBN 978-1-60643-379-9

[78] Global Uncertainty-based Selection of Relative Poses for Multi-Camera Calibration
Bajramovic, F., Denzler, J.
Proceedings of the British Machine Vision Conference (BMVC). Volume 2. (2008) 745-754

[79] An Efficient Shortest Triangle Paths Algorithm for Uncertaintybased Multi-Camera Calibration
Bajramovic, F., Denzler, J.
OMNIVIS 2008, the Eighth Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras. (2008)

[80] Experimental Evaluation of Relative Pose Estimation Algorithms
rueckner, M., Bajramovic, F., Denzler, J.
Proceedings of the Third International Conference on Computer Vision Theory and Applications. Volume 2. (2008) 431-438

[81] The analysis of the errors for a stereo sensor built in two configurations
L. Toma Jr., F. Shu, A. Ignea, W. Neddermeyer
Buletinul Stiintific al Universitatii Politehnica din Timisoara, Tomul 47(61), 2002 Fascicola 1-2, Vol. 2, pp. 174-178

[82] Stereo Vision - A new solution for matching the left and the right image models with high accuracy
W. Neddermeyer, A. Ignea, L. Toma Jr., F. Shu
SCI2003 Orlando - Florida, USA, Proceedings, Volume X, pp. 145-149

[83] The development of a new system to measure Camber and Toe using stereo cameras
L. Toma Jr., F. Shu, A. Ignea, W. Neddermeyer
Buletinul Stiintific al Universitatii Politehnica din Timisoara, Tomul 49(63), 2004 Fascicola 1-2, Vol. x, pp. 236-239

[84] Stereo Vision Sensor for 3D Measurements - A complete solution to produce, calibrate and verify the accuracy of the measurements results
L. Toma Jr., F. Shu, W. Neddermeyer, A. Ignea
ICINCO2004 - Setubal, Portugal, Proceedings, Volume 2, pp. 410-416

[85] Precise online camera calibration in a robot navigating vision system
F. Shu, L. Toma, W. Neddermeyer, J. Zhang
vol.3 from IEEE International Conference on Mechatronics and Automation, Niagara Falls, Ontario, Canada, 2005

[86] Online approaches to camera pose recalibration
Fangwu Shu, Werner Neddermeyer, Jianwei Zhang
IEEE International Conference on Intelligent Robots and Systems, Beijing, China, 2006

[87] A multi-camera system for precise pose estimation in industrial applications
Fangwu Shu, Jianwei Zhang, Youfu Li
IEEE International Conference on Automation and Logistics, Shenyang, China, 2009

Index

About the author

Mr. Fangwu Shu was born in FengHuang, a small village in central China. He got his bachelor and master degrees in computer science in 1996 and 1999 respectively. After a short year of work in the field of image processing, he came to Germany in 2000 to work in the field of computer vision by TecMedic GmbH, an underling company from Fachhochschule Gelsenkirchen. With nearly one year of study on camera calibration and measurement techniques, he had been engaged in some research projects for applying the vision systems to the robot vision applications. Thanking to the related research work, he was matriculated as a doctor student by the department of informatics at the University of Hamburg in December, 2003.

Although he was a member of TecMedic GmbH, he had worked since 2003 more often for VMT GmbH, a business partner of TecMedic GmbH and a famous vision systems supplier in industries. During that period, many application issues arisen from VMT projects were worked over and solved in the laboratory of TecMedic GmbH. Due to the integration of VMT GmbH into Pepperl+Fuchs Group in Mannheim, he joined Pepperl+Fuchs Shanghai in 2007 to develop VMT business in China market.