THEORY AND DEVELOPMENT OF A CAMERA-BASED NONCONTACT VIBRATION MEASUREMENT SYSTEM

A Thesis Presented to the Faculty of Graduate School

University of Missouri

In Partial Fulfillment
Of the Requirements for the Degree

Master of Science

By

DAN FENG

Dr. P. Frank Pai, Thesis Supervisor

DECEMBER 2012

The undersigned, appointed by the Dean of the Graduate School, have examined the thesis entitled

THEORY AND DEVELOPMENT OF A CAMERA-BASED NONCONTACT

VIBRATION MEASUREMENT SYSTEM

Presented by Dan Feng,

A candidate for the degree of Master of Science

And hereby certify that, in their opinion, it is worthy of acceptance.

_____

Dr. P. Frank Pai

_____

Dr. Ye Duan

_____

Dr. Yuyi Lin

ACKNOWLEDGEMENTS

First of all, I want to express my sincere appreciation to my advisor Dr. P. Frank Pai for his guidance, training and support throughout my study at the University of Missouri. His rigorous attitude in research influences me very much. Also, I want to express my appreciation to Dr. Ye Duan and Dr. Yuyi Lin for their serving on my thesis committee.

I want to thank my fellows in Dr. Pai's Structural Mechanics and Controls Laboratory, Dr. Jun Liu, Hao Peng, Bao Anh Nguyen and Cory Kaufman, for their friendship and help during my tenure at Mizzou. Moreover, I thank my parents for their endless love to me and support of all my decisions and their financial support throughout my college and graduate studies.

This research has been supported by the NSF under Grants CMS-0120798 and CMMI-1039433. The support is gratefully acknowledged.

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

THEORY AND DEVELOPMENT OF A CAMERA-BASED NONCONTACT VIBRATION MEASUREMENT SYSTEM

Dan Feng

Dr. P. Frank Pai, Thesis Advisor

ABSTRACT

Dramatic advancement in technologies for high-speed high-resolution digital cameras in recent years enables the development of camera-based full-field noncontact measurement systems for vibration testing of flexible multibody systems undergoing large rigid-body motion and elastic/plastic deformations. A few of such systems exist in today's metrology market, but they are inconvenient for use and prohibitively expensive. Most seriously, they are not really appropriate for structural vibration testing because their measurement accuracy is low due to several technical reasons, including inappropriate setting of cameras and experimental setup because of user's innocence of video-grammetry, non-precise corner detection and other problems of image processing techniques, and inaccurate modeling and calibration of cameras. This thesis develops and puts together a complete set of necessary techniques for the development of a camera-based noncontact full-field vibration measurement system using inexpensive off-the-shelf digital cameras. An optimal combination of appropriate methods for corner detection, camera calibration, lens distortion modeling, and measurement applications is proposed and numerically and experimentally verified. Moreover, we derive/improve some image processing methods and 3D reconstruction algorithms to improve vibration measurement accuracy.

The proposed methods include: 1) a corner detection method for processing 2D images with sub-pixel resolutions, 2) an improved flexible camera calibration method for easy and fast calibration with high accuracy, 3) a lens distortion model for correcting radial, decentering, and thin prism distortions, 4) a set of guidelines for setting up cameras and experiments for measurement, and 5) algorithms for measurement applications. The proposed corner detection method improves Foerstner's corner detector, which improved Moravec's and Harris's corner detectors. The proposed camera calibration method improves Zhang's flexible technique, which works without knowing the object's 3D geometry or computer vision. The method only requires the camera to observe a planar pattern (e.g., a checker board) shown at two or more independent orientations by arbitrarily moving the planar pattern (or the camera). Estimation of the camera's intrinsic parameters (i.e., focal length, principal point, the skewness parameter and aspect ratios of the two image axes, and lens distortion parameters) and extrinsic parameters (i.e., camera's location and orientation with respect to the referential world coordinate system) consists of an approximate initial guess based on linear closed-form solutions and then nonlinear optimization for refinement. This approach is between the photogrammetric calibration and the self-calibration. Compared with photogrammetric calibration techniques that use expensive calibration objects of two or three orthogonal planes, the proposed technique is easy to use and flexible.

To examine the proposed methods and their combined effects against high measurement accuracy, two Canon EOS-7D DSLR cameras are used for theoretical studies and experimental verifications. Numerical and experimental results show that the recommended methods together with our improved image processing techniques is

feasible for the development of a camera-based noncontact full-field vibration measurement system with high precision and low cost. This camera-based measurement instrument has the potential for developing new structural testing techniques and can open new possibilities for research and development in mechanical and aerospace engineering, computer science, animal science, and many other fields.

CHAPTER 1

INTRODUCTION

1.1 Background

Accurate noncontact full-field measurement of deformable physical systems is needed in many today's engineering fields. Two significant examples are the static and dynamic testing of high-speed flexible multibody systems used in mechanical engineering and highly flexible deployable/inflatable structures systems used in outer-space engineering [1,2]. Such flexible mechanical systems also serve in many aerospace, architecture, automobile manufacturing, shipbuilding, nuclear, and wind-power industries; examples include helicopter rotors, aircraft wings, robot manipulators, wind turbine blades, antennas, and so on [1-7]. A flexible multibody system consists of interconnected elastic and rigid components undergoing large displacements and rotations caused by flexibility-induced geometric nonlinearities and dynamic coupling between elastic and rigid components. Accurate measurement of static/dynamic deformations of such multibody systems is important for model verification and design optimization, but it is a very challenging task by using traditional metrology methods.

For example, electronic strain gages can only measure local relative straining displacements between two points, and other contact sensors are often too heavy and/or bulky. Moreover, gauging sensors using small springs cannot give accurate readings because the spring force causes extra deformation of the measured flexible system. Although noncontact laser sensors like scanning laser vibrometers can provide accurate measurement of small deformations, they cannot track and measure displacements of

points fixed on a multibody system undergoing large rigid-elastic deformation [2]. Other noncontact sensors often can only measure small displacements [8]. Moreover, large flexible multibody systems often require full-field measurements in order to understand/verify relative deformations between different rigid and elastic components.

Fortunately, recent fast development in high-speed high-resolution digital cameras enables the design of camera-based full-field noncontact measurement systems [9-11], and there are more and more commercial video-grammetry metrology systems available in today's metrology market [12-15]. Unfortunately, they are mostly designed for measuring multibody systems undergoing large slow-speed motions with low requirements of measurement precision. For engineering applications, especially for structural vibration testing, high measurement accuracy and speed are necessary because post-processing of measured dynamic deformations/displacements is needed in order to estimate velocities, accelerations, and strains. Hence, challenging issues exist for video-grammetry to grow and become a popular measurement method for such engineering applications.

To reveal the challenging issues in photo- and video-grammetry for structural testing, we consider an eight-camera Eagle-500 motion analysis system in the Structural Mechanics and Controls (SMC) Laboratory of the University of Missouri. Figure 1.1 shows an Eagle-500 motion analysis system prepared for static/dynamic testing of helicopter rotor blades. The system uses 8 *complimentary metal-oxide-semiconductor* (CMOS) cameras to capture pictures of a structure when 8 visible red digital **light-emitting diode** (LED) strobes light up the retro-reflective markers adhered on the structure. The cameras and strobes are synchronized to work at a speed between 0.1 to

2000 frames per second (FPS). For a frame rate between 0.1 and 500 FPS, a full resolution of 1280×1024 pixels is used. For a frame rate within 500~1000 (1000~2000) FPS, a resolution of 1280×512 (1280×256) pixels is used. Using triangulation techniques and the known focal lengths (after calibrations using an L-frame with 4 precisely located spherical markers and a T-wand with 3 spherical markers) of the cameras and the known image coordinates of the bright points (caused by the retro-reflective markers) on the CMOS sensors inside the cameras, the Eagle-500 signal processing software EVaRT4.6 computes and records the instant 3D object coordinates of the center of each retro-reflective marker that is seen by at least two cameras. Hence, 3D time traces of all markers are available for performing dynamic animation using stick figures. Because the 3D coordinates of each marker are estimated by optimization using *bundle adjustment* [9-13] when two or more cameras see the marker, the measurement accuracy of the system is high enough for vibration testing of highly flexible structures that undergo large rigid-elastic deformations [2,3]. Because on-earth structures are often designed to have vibration amplitudes less than their thickness (i.e., linear vibrations) under normal operation conditions, the measurement accuracy of the Eagle-500 system is lower than necessary for valid vibration testing of such structures [2].

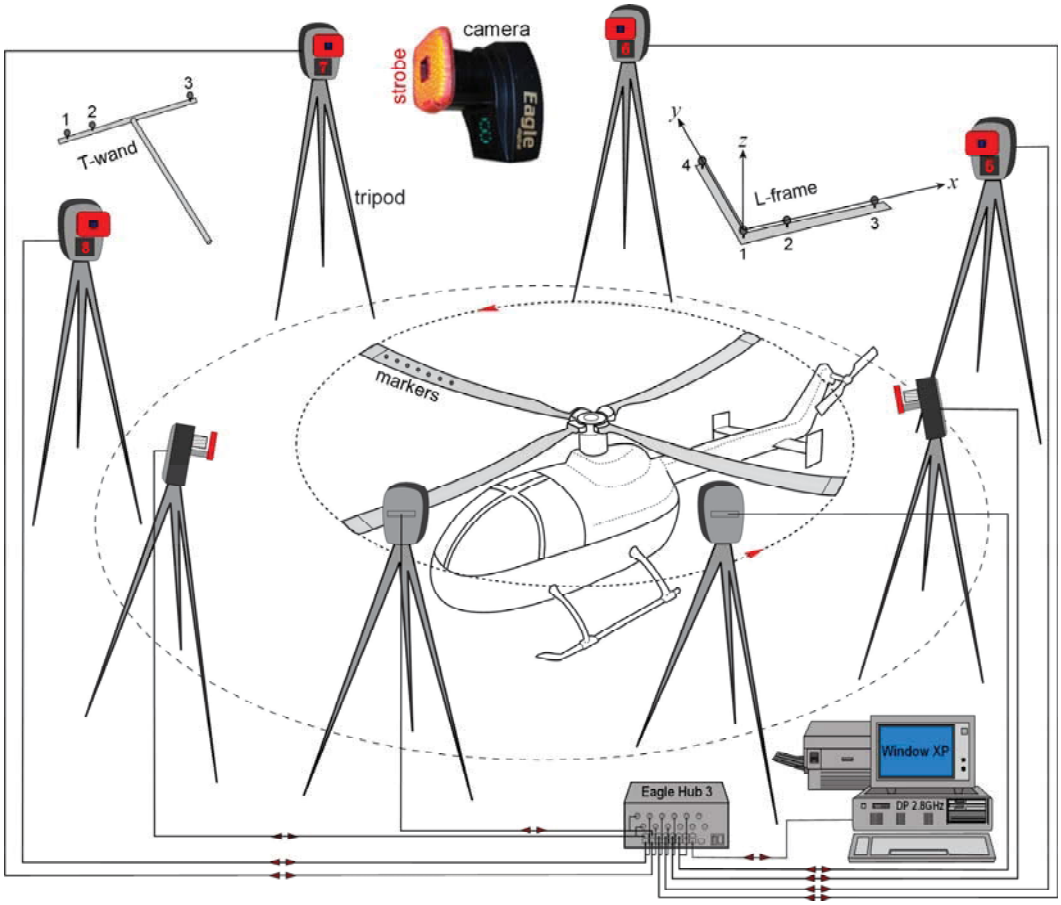


Figure 1.1 An Eagle-500 motion analysis system set up for measuring large static/dynamic deformations of flexible rotor blades.

Before such a camera-based system can be used for measurement, a fixed coordinate system needs to be defined for reference, each camera's actual focal length, location, and orientation with respect to the fixed coordinate system need to be determined, and a lens distortion model for correcting radial, decentering, and thin prism distortions needs to be developed [8-11]. The measurement accuracy highly depends on accurate estimations of these parameters. Unfortunately, because all these system parameters are nonlinearly coupled in the estimation process by optimization and there

are measurement errors caused by hardware, software, and the user, the estimations are difficult to be accurate. More seriously, these systems are often not designed for vibration testing of structures, and the measurement theories and signal processing methods used in such systems are unknown and not available to structural testing engineers. Experimental results [2,8] reveal that the measurement accuracy (≡ (RMS of measurement errors)/(maximum object length)) ranges from 1/200 to 1/3000, depending on experimental setup, camera calibration, frame rates, and user's experience. For example, the best accuracy obtained in the MSC Lab using this Eagle-500 system was about *0.25mm* when the maximum length of the object was *479mm*, the measurement volume was set at $2m \times 2m \times 2m$, and the frame rate was 500 FPS. With such a low and unstable measurement accuracy, it is difficult to further process the measured data to derive velocities, accelerations, and strains for dynamics characterization, system identification, damage inspection, health monitoring, and many other engineering applications.

The measurement accuracy of such camera-based systems is limited by several factors. (1) The image resolution and frame rate of such systems are low. (2) Each camera's parameters change with setup conditions, time and environments, and hence fast and accurate on-site calibration is needed but difficult. (3) Photo-grammetry software of such systems is not developed for structural vibration testing. (4) Users without certain understanding of photo-grammetry cannot appropriately setup such a system for accurate measurement. Because the large spherical markers used in such camera-based motion analysis systems may cause serious unknown aerodynamic effects; sticking markers onto the object under measurement is a burdensome work, especially for some large space structures; the image resolution and frame rate of such systems are normally low; and

such systems are too complicated for users with little or no knowledge of photo-grammetry to calibrate cameras and set up hardware appropriately [2,3], there are rooms for improvement but new technology development is necessary.

From the above discussions and our experience with the Eagle-500 system we summarize the *challenging issues* in developing camera-based vibration measurement systems into three areas: 1) experimental setup, 2) camera modeling and calibration, and 3) image processing and 3D reconstruction. Researchers and software and hardware industries have been trying to solve the challenging issues in order to improve the accuracy and efficiency of camera-based noncontact measurement systems.

1.2 Literature Review and Discussions

Challenging problems in experimental setup of a camera-based measurement system can be solved or released only if the user has good understanding and experience of how photo-/video-grammetry works [9-12]. For example, to record clear and sharp images of a vibrating structure, each camera's focus needs to be fixed at the center of movement range, and the depth of field needs to be enough to cover the movement range. The depth of field can be increased by decreasing the aperture diameter. If the structure's vibration frequency is high, the shutter speed needs to be high in order to record clear images. In order for the light sensor (or film) to have enough exposure when a high shutter speed is used, the aperture diameter needs to be large, a high ISO number for the sensor needs to be used, and/or the structure needs to be brightly laminated. Unfortunately, a large aperture diameter will result in a short depth of field and blur images, and a high ISO number will results in grainy images for colorful parts. The

influences of all these factors together can be appropriately adjusted only if the user understands camera optics [11].

To reduce the aerodynamic and size effects caused by spherical markers, small, flat, circular retro-reflective markers (see Figure 1.1) have been used in vibration testing of highly flexible structures [2,8]. If a flat circular marker on the structure is not relatively deformed, its projected image on the sensor plane should be an ellipse if lens distortion is negligible. However, its size and the directions of two axes vary with the orientation of the deformed structure with respect to the image coordinate system. The measured deformed structural geometry at the previous time step (or the current time step during an iteration process) can be used to estimate the structure's orientation. Moreover, a sub-pixel target marking method [13] that uses the gray value of each pixel for weighting can be used in the ellipse fitting process to precisely locate the marker center. Of course, markers of other shapes can be also used. For example, small cylindrical retro-reflective markers wrapped on a very flexible spinning shaft have been used to measure and investigate the forward and backward whirling dynamics of spinning shafts [16]. Accounting for the projected marker shape in the calculation of each marker's center should be able to improve the measurement accuracy.

To reduce the time for setting up markers on a structure, some researchers used a slide projector as an easy and quick way to create temporary dots or patterned shapes on a structure [17,18]. Recently available holographic laser projectors can be a better choice for projecting bright dots and/or patterned shapes on a structure. The projected dots and/or shapes can enhance distinctness of corners and other interest points and ease the corner and feature detection process during image processing. Unfortunately, because

such projected dots and/or patterned shapes do not move with the structure, they can only be used for measuring the dynamically deformed overall geometries without the possibility of tracking specified points on the structure. However, with the use of few markers fixed on the structure and a structural theory [2], the deformed overall structural geometry still can be used to estimate strain distribution.

A pinhole model is a simplified, ideal, lensless model of cameras. An actual camera consists of lenses, an aperture-control diaphragm, a shutter, and a light sensor. The relative positions among these components may change with time, usage, and environment, especially temperature variation. Moreover, a lens has different thicknesses and curvatures varying from the lens center to its outside radius, and hence image distortion always exists. Furthermore, a far object's image can be focused on the lens' focal point, but a near object's image cannot. Considering all these factors, camera calibration right before measurement is really necessary for using camera-based measurement systems [9-11]. Unfortunately, camera calibration is a very challenging nonlinear inverse engineering problem [11,19]. It requires the use of a well-calibrated object's 2D images taken by the camera before its intrinsic and extrinsic parameters are known. Because numerical corner and feature detection on the images is needed before nonlinear optimization iterations with a rough initial guess of these parameter values are performed, camera calibration is difficult to be accurate.

Image processing actually plays the key role in a camera-based noncontact vibration measurement system because its accuracy directly affects the measurement accuracy of the system. The most important part of image processing is corner detection because removing corner points from an image impedes human recognition but removing

straight edges does not. Moreover, corner points are translation and rotation invariant, and they are stable under varying lighting conditions.

Many researchers investigated corner detection methods. One of the earliest methods was the Moravec corner detection algorithm [20]. This algorithm compares the similarity between a pixel's immediate surrounding patch and another neighboring patch on its horizontal, vertical, or diagonal line. If the summation of squared differences between the grey values of these two patches is maximal, this pixel is a corner point. Unfortunately, this operator is not isotropic, and it may miss corner points not on the examined directions [20]. Harris and Stephens algorithm [21] is an improved version of the Moravec algorithm; its operator is isotropic with a circularly weighted window. The Tomas corner detector directly computes the smallest eigenvalue of the Harris matrix (shown later in Section 2.3.2) to accelerate and stabilize computation [22,23]. The Foerstner corner detection method can provide corner locations with sub-pixel accuracy [24]. According to the fact that the intersection of all tangent lines of a corner should be the correct corner location, Foerstner algorithm estimates the corner location by using all the tangent lines in a given window by least-square fitting [24]. Some other corner detectors were developed in the last few years. Wang and Brady [25] considered a corner point as a point with a large curvature along the image surface. SUSAN (Smallest Univalue Segment Assimilating Nucleus) corner detector [26, 27], Trajkovic and Hedley corner detector [28], and AST (Accelerated Segment Test) based feature detectors [29] use a circular mask to process low level images. However, most corner detectors are not very robust and require significant manual modification works in order to reduce detection errors. In this thesis, we adopt and improve the Foerstner corner detection

method in order to ensure sub-pixel accuracy with good repeatability and information content [30,31].

After finding the locations of five (or more) corner points on each of three (or more) 2D images of a well-calibrated planar pattern, these corner locations can be used for camera calibration to extract intrinsic and extrinsic parameters of a camera [19]. A camera's intrinsic parameters include the focal length, principal point location on the sensor plane, the skewness parameter and aspect ratios of the two image axes, and lens distortion parameters. These intrinsic parameters may change with time, environment, and usage. Extrinsic parameters are the 3D location and orientation of a camera with respected to a defined referential coordinate system.

Many calibration methods have been invented. There are two groups of methods for calibration of the intrinsic parameters. *Photogrammetric calibration* is performed by observing a calibration object of high precision, which usually consists of two or three planes orthogonal to each other [8]. Tsai used a plane undergoing a precisely known translation [32]. These approaches require an expensive calibration apparatus and an elaborate setup. On the other hand, *self-calibration* is a linear method and does not use any calibration object, and it depends on the rigidity of the scene to provide two or more constraints on the camera's intrinsic parameters by using image information alone. Maybankand Faugeras proposed a self-calibration method by moving a camera with respect to a static scene [33]. If a camera's internal parameters are fixed, one can extract all its internal and external parameters from three different image views [34]. This approach is very flexible because it simultaneously estimates many parameters, but the result may not be reliable. There are some techniques between photogrammetric

calibration and self-calibration and are called co-linear calibration methods, including the geometric approach [35], the neural nets method [36], statistical methods [37], and the vanishing points method [38-40]. Several free camera calibration toolboxes are available from the internet, including the Microsoft Easy Camera Calibration Tool [41], Caltech Camera Calibration Toolbox for MATLAB [42], the toolbox for MATLAB by Heikkila [43], and Tsai Camera Calibration Software [44]. We adopt a co-linear method that can observe a planer pattern at two or more independent orientations by moving the planar pattern or camera arbitrarily [19,45].

After camera calibration, the intrinsic and extrinsic parameters of cameras are known and hence they can be used for vibration measurements. The captured 2D images of a structure can be used with the known intrinsic and extrinsic parameters of all cameras and triangulation techniques to calculate the physical coordinates of the marker centers on the structure. These identified physical coordinates of marker centers can be used to reconstruct a 3D model of the measured structure. Tracking of markers through the measured time period is challenging even up to now [9-15]. During the post-measurement data processing, it is very often that markers at different times cannot be smoothly connected and user's subjective justification and manipulation are needed. With the use of cameras at high frame rates, this problem can be partially released because the incremental displacement of each marker may be smaller than the distance between any two markers. If the incremental displacements of markers are greater than the distance between markers or even there are missing points, smooth and continuous motions are assumed and a numerical template of few degrees of freedom (DOFs) is often used to track markers [12-15]. For a structure, its linear mode shapes from modal analysis are the

best candidates for building a physical template. To build a physical template for a structure, only a few low-frequency mode shapes are needed, and the corresponding modal coordinates can be determined from the time-varying operational deflection shapes of the previous time steps.

1.3 Motivation and Goals

For camera-based vibration measurements, flexibility, robustness and low cost are important. Current commercially available camera-based photo- and video-grammetry measurement systems are either not accurate enough for structural vibration testing or too expensive (e.g., more than $100,000 per system). Our motivation is to develop such a system with an affordable cost, high measurement accuracy, and easy for calibration and usage by using off-the-shelf digital cameras, an appropriate combination of necessary techniques, and some new development/improvement of software and hardware.

Although several methods and algorithms for corner and edge detection, thinning, thresholding, and morphology are available from today's internet, an appropriate combination of them for structural vibration testing does not exist. With our experience with the Eagle-500 system and others, trying markers of different sizes and shapes, experimenting different ways of hardware setup and image processing techniques, and knowing what cause inaccurate measurements, we want to put together in this thesis a complete set of necessary techniques for the development of a camera-based noncontact full-field vibration measurement system using off-the-shelf digital cameras, including methods for corner detection, camera calibration, lens distortion modeling, and

measurement applications. We also want to derive/improve necessary image processing methods and 3D reconstruction codes for vibration measurements with better accuracy.

1.4 Thesis Overview

This thesis presents a high-precision easy-to-use noncontact vibration measurement system based on the use of high-speed high-resolution digital cameras and advanced image processing techniques. For theoretical studies and experimental verifications, we use two Canon EOS-7D DSLR (digital single lens reflex) cameras. For comparison of numerical and experimental results, we use these two Canon cameras and an eight-camera Eagle-500 motion analysis system.

In Chapter One we present the background for developing camera-based noncontact measurement systems, provide a literature review, present the motivation and research goals, and then give an overview of this thesis. We point out the need for and feasibility of camera-based vibration measurement systems and the remaining challenging tasks to be researched and solved in order to mature this technology.

In Chapter Two we present fundamental concepts and formulas needed for understanding camera optics and image recording and processing. The sections on Light and Its Properties and Basic Camera Optics explain how to determine the three major camera parameters (i.e., aperture diameter, shutter speed, and ISO number) in order to capture clear images of a vibrating structure. Because corner detection is to extract interest points from a 2D image in order to infer its contents, corner detection is the important first step in camera calibration, we present a corner detection method in detail and describe how to refine a corner point’s position with sub-pixel accuracy.

In Chapter Three we present a flexible technique for easy calibration of cameras without knowing the object's 3D geometry or computer vision. Different experimental conditions are compared in order to determine suitable image/video formats and recording qualities. A four-point calibration method is also derived and illustrated in Section 3.3, which allows the use of any single rectangular pattern to perform quick camera calibration. It reduces the minimum number of required corner points from 5 to 4 by adding an orthogonal constraint. Although the calibration result may not be as accurate as those from the use of a checker board with many squares, this method is simple and quick and is very useful for outdoor field applications.

In order to prepare the reader for following chapters, the coordinate systems to be used in this thesis is defined and explained here. Figure 1.2 shows three coordinate systems with a pinhole camera model. The $\xi\eta\zeta$ is a Cartesian coordinate system with the origin point in the perspective (or lens) center of a pinhole camera model and is called the *image coordinate system*. The *xyz* is a Cartesian coordinate system fixed on the solid object plane and is called the *world* (or *object*) *coordinate system*, whose origin point is (ξ_0, η_0, ζ_0) with respect to the $\xi\eta\zeta$ coordinate system. Moreover, The uv is the image-plane coordinate system, (u,v) are the ideal, undistorted image-plane coordinates of the object point's image on the sensor plane, $(\tilde{u},\tilde{v})$ are the measured, distorted image coordinates, and (u_0, v_0) are the image plane coordinates of the *principal point* (i.e., *point of symmetry*, or *center of radial distortion*) that may not coincide with the *fiducial center* (i.e., the image center) [9-11]. The undistorted coordinates (u,v) are essentially unknown, but one can use a backward lens-distortion model from previous calibration to estimate

them. Note that the image (sensor) plane may not be perpendicular to the *optic axis* ζ due to misalignment.

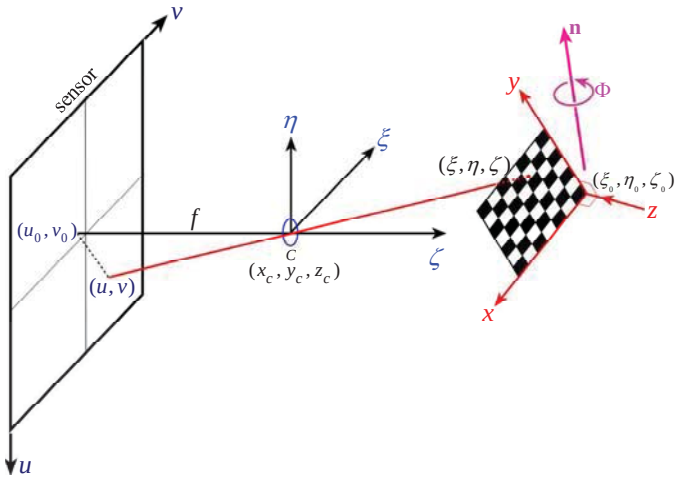


Figure 1.2 Three coordinate systems with a pinhole camera model.

As shown later in Chapter Three, the relations among these coordinate systems are listed below:

$$\begin{Bmatrix} u \\ v \\ 1 \end{Bmatrix} = [A] \begin{Bmatrix} \xi \\ \eta \\ \zeta \end{Bmatrix} \frac{1}{\zeta}, \begin{Bmatrix} \xi \\ \eta \\ \zeta \end{Bmatrix} = [T] \begin{Bmatrix} x \\ y \\ z \end{Bmatrix} + \begin{Bmatrix} \xi_0 \\ \eta_0 \\ \zeta_0 \end{Bmatrix} = [[T]\{\tau\}] \begin{Bmatrix} x \\ y \\ z \\ 1 \end{Bmatrix} \tag{1.1.1}$$

$$[A] \equiv \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \{\tau\} \equiv \begin{Bmatrix} \xi_0 \\ \eta_0 \\ \zeta_0 \end{Bmatrix}$$

where $[A]$ is the camera's linear intrinsic parameter affine matrix having 5 unknown parameters, and $[[T]\{\tau\}]$ is the extrinsic parameter matrix having 6 unknown parameters,

including ξ_0, η_0, ζ_0 in $\{\tau\}$ and 3 Euler angles or modified Rodrigues parameters in the 3×3 matrix $[T]$ [46,47].

In Chapter Four we present a measurement theory and steps and setups for experimental verification of the proposed camera-based noncontact system for static and dynamic measurements. It shows how to find and arrange the marker centers. For static measurement, retro-reflective markers and holographic laser points are used to create dots or patterned shapes on a structure. For dynamic measurement, two Canon EOS-7D cameras and a wireless remote controller are used to setup the system. One hinged-free cantilevered flexible beam and one hinged-free flapping L-shaped flexible beam are utilized to examine the availability and accuracy of our system. Experimental results show that, although the maximum frame rate of a Canon EOS-7D camera is 60 FPS, the measurement accuracy is higher than Eagle-500 motion analysis system.

In Chapter Five we summarize the steps for development of a camera-based vibration measurement system, provide experimental guidelines, and list recommendations for future research. Although the measurement result shows high measurement accuracy, some further processing and improvement are recommended to obtain a higher precision. Such recommendations include both equipment improvements and theoretical enhancements. These recommendations are expected in helping achieve a whole automatic system applied for both retro-reflective makers and holographic laser points.

CHAPTER 2

IMAGE RECORDING AND PROCESSING

In this chapter, some fundamentals for the development of a camera-based noncontact vibration measurement system are presented. It is important for a user of camera-based measurement systems to have good understanding of these fundamentals in order to appropriately set up such measurement systems for system calibration and then for obtaining accurate and repeatable measurements of time-varying coordinates of dynamic object points.

2.1 Light and Its Properties

Light is an electromagnetic wave, and it also shows particle properties when it is emitted or absorbed. Light is emitted by accelerated electric charges, and the speed of light in vacuum, *c*, is a fundamental physical constant. It is more convenient to represent the propagation direction of a light wave by rays (called *geometric optics*) instead of wave fronts (called *physical optics*).

The propagation direction of a light wave can be changed due to *reflection* and *refraction*, which can be explained by *Huygens' principle*. Huygens' principle states that every point of a wave front may be considered as the source of secondary wavelets that spread out in all directions with a speed equal to the speed of propagation of the wave [48]. At a smooth interface between two optical materials, the *law of reflection* states that the reflection angle of a light ray is equal to the incidence angle for light rays of any wavelengths and for any pair of materials. The refraction angle of a light ray is

determined by the *index of refraction* (also called the *refractive index*) n (≥ 1), which is defined as

$$n \equiv \frac{c}{v} = \frac{\lambda_0}{\lambda} \tag{2.1.1}$$

where v is the propagation speed of a light wave in a material, λ is the wavelength of the light, and λ_0 is the wavelength of the same light in vacuum. When a light ray enters a dense material, its frequency does not change, but its wavelength and speed decrease. For a monochromatic light ray propagating through two in-contact materials, the *law of refraction* (or *Snell's law*) states that the incidence angle θ_1 (with respect to the contacting surface's normal) in the first material and the refraction angle θ_2 in the second material are related as

$$n_1 \sin\theta_1 = n_2 \sin\theta_2 \tag{2.1.2}$$

2.2 Basic Camera Optics

Before image recording and processing, one needs to understand some basics of camera optics in order to obtain pictures/videos of high quality. Precision of pictures/videos will affect the accuracy of other tasks needed for the development of a camera-based non-contact measurement system, including corner detection, marker finding for calibration, and motion tracking of moving objects.

A *pinhole camera* is a simple lensless camera with a single small aperture, and is effectively a light-proof box with a small hole in one side. Unfortunately, a pinhole camera has no much control over the light intensity received by the light sensor. Hence,

like a human eye, a practical camera consists of a lens, an aperture-control diaphragm (i.e., an iris), and a light sensor (i.e., a retina).

2.2.1 Principle of Imaging

Figure 2.1 shows how an image of an object is formed on the sensor plane in a camera. A lens is an optical system with two refracting surfaces. After the light rays emit from the object, they go through and refracted by the lens and then converge at the light sensor plane. A real camera often uses a combination of several lenses in order to correct various *aberrations*, and the lenses may be assembled with an aperture-control diaphragm.

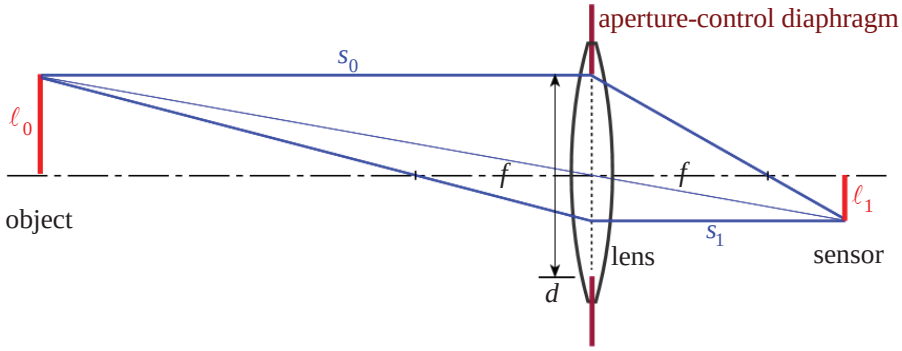


Figure 2.1 The imaging principle of cameras.

For a thin lens, the left and right focal lengths are essentially equal even if the two sides have different curvatures [49]. For the ray passing through the lens center, there is refraction when the ray enters and leaves the lens but the direction remains the same. Although the ray is parallelly shifted, the amount is negligibly small because the lens thickness is thin. Hence, it can be shown by using the similarity between right triangles in Figure 2.1 that

$$\frac{1}{f} = \frac{1}{s_0} + \frac{1}{s_1} \tag{2.2.1}$$

$$m \equiv \frac{\ell_1}{\ell_0} = \frac{s_1}{s_0} \tag{2.2.2}$$

where m is the image magnification, s_0 is the object distance, and s_1 is the image distance. If $s_0 / s_1 \to \infty$, we have $f \approx s_1$. This enables the use of a pinhole camera model for the derivation of equations for camera calibration, as shown later in Chapter 3.

Using the concept that the image formed by the first refracting surface (with a radius of curvature $R_1 > 0$) can serve as an object for the second reflecting surface (with a radius of curvature $R_2 < 0$), one can prove that [49]

$$\frac{1}{f} = (n-1)\left(\frac{1}{R_1} - \frac{1}{R_2}\right) \tag{2.2.3}$$

This is called the lens maker's equation for a thin lens.

2.2.2 Photographic Triangle

The quality of a recorded image is determined by three factors: aperture, shutter speed, and ISO speed. They are referred as the photographic triangle. However, the light meter is the heart of the photographic triangle. Whether you shoot in Program mode, Shutter Priority mode, Aperture Priority mode, or even Manual mode, you may think that as long as the light meter indicates that everything is okay and hence it must be okay to shoot. Unfortunately, this is a shotgun approach because each picture-taking situation has at least six possible combinations of f-stops and shutter speeds that will all result in a

correct exposure. How to consistently record creatively correct exposures every time requires certain understanding of the relations between these three factors.

In camera optics, the ISO (International Organization for Standardization) speed (or sensitivity) represents the image sensor's sensitivity to light. If the speed is high, the sensor needs less light to make a complete exposure. In simple words, we need a high ISO speed if we take a picture in a dark environment, and a low ISO speed can be used in a bright environment.

An aperture is a hole or an opening through which light travels, and it determines the cone angle of a bundle of rays comes to a focus. The *f*-number (or *f*-stop number) N is the ratio of the focal length f of the lens and the diameter d of the aperture (see Figure 2.1), i.e.,

$$N \equiv \frac{f}{d} \tag{2.2.4}$$

If $f/d=5.6$, the *f*-number is often denoted as *f/5.6*. So, a large *f*-number means a narrow aperture. The full stops are *f/4, f/5.6, f/8, f/11, f/16, f/22* and *f/32*. Increase of the *f*-number from one value to the next one means the volume of light entering the lens is cut in half. However, many cameras today offer not only full stops but also the ability to set the aperture to one-third stops, i.e., *f/4, f/4.5, f/5, f/5.6, f/6.3, f/7.1, f/8, f/9, f/10, f/11,* and so on. A lens is an elliptically shaped glass element with a magnet at the central area to gather a specific amount of light and funnel it through to the light sensor. This magnet is called the *sweet spot* and has a diameter equivalent to f/8 to f/11. Hence, apertures from f/8 to f/11 are often the sharpest and offer the greatest contrast in exposure.

The shutter speed indicates the time interval that a camera's shutter opens for sensor exposure. A shutter speed of 2" means the shutter is opened for 2 seconds, and a shutter speed of 2 means the shutter is opened for 1/2 second.

To capture a sharp image of a moving object, the sensor exposure time needs to be short (by using a high shutter speed) but the aperture diameter needs to be large in order to collect enough light. However, a large *d* results in a short *depth of field*. For the sensor to record an image properly, the total light energy per unit area on the sensor (the "exposure") must fall within certain limits. The intensity of light reaching the sensor is proportional to the area viewed by the camera lens and to the effective area of the lens. The area viewed by the camera lens is proportional to the square of the angle of view of the lens and hence is roughly proportional to $1/f^2$. The effective area of the lenses controlled by the lens aperture and is proportional to d^2. Hence, the energy *E* collected by a sensor is proportional to the brightness *B* of the object and the area of the aperture and is inversely proportional to the square of the focal length as

$$E \propto \frac{B\pi d^2}{4f^2} = \frac{B\pi}{4N^2} \tag{2.2.5}$$

The depth of field is also affected by the focal length and the distance between the camera and the object.

Aberrations are mainly caused by the change of refractive index with light's wavelength and the paraxial approximation used in calculating the focal point, and they can be separated into two classes: monochromatic and chromatic. *Monochromatic aberrations* are caused by the geometry of the lens and occur both in reflection and refraction. They appear even when using monochromatic light, hence the name.

Chromatic aberrations are due to dispersion caused by the variation of a lens's refractive index with light's wavelength. They do not appear when monochromatic light is used.

2.2.3 Depth of Field

As shown in Figure 2.2, the distance between the nearest and farthest objects that can have sharp projections on the light sensor is called the *depth of field* (DOF), and the distance between the focuses of these nearest and farthest objects is called the *depth of focus*. Each point within the depth of field can have a sharp projection on the light sensor with a width less than the diameter of the *circle of confusion*.

When the cone of rays from the lens does not converge to a perfect focus, it will appear as a circle on the sensor. The *circle of confusion* is the one less than the resolution of human eyes, and an area smaller than the circle of confusion is perceived as one point. After setting the focal length, one can change the definition by trimming the focusing ring. Because all interest points need to be clear on the sensor in order to have high precision image processing, all interest points need to be inside of the depth of field. For dynamic problems, one needs to make sure that the object's range of movement does not exceed the depth of field, or a larger depth of field is needed.

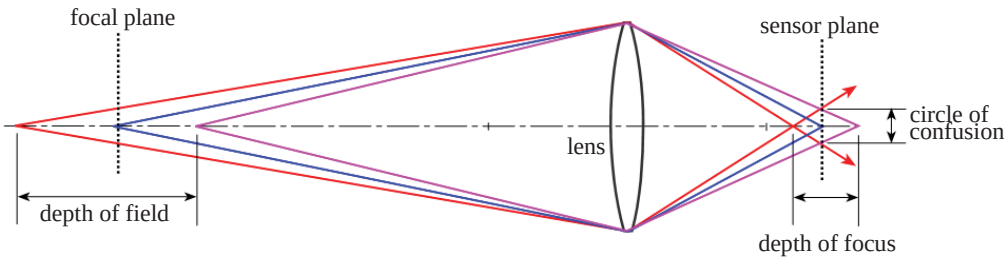


Figure 2.2 The depth of field, the depth of focus, and the circle of confusion.

To have a more precise definition of clear images, the circle of confusion is often defined to be the size of smallest detail that can be distinguished from a viewing distance of 250 mm. In other words, it is the size of spot that cannot be distinguished from a true point at a viewing distance of 250 mm, and it is usually taken to be a spot or circle with a diameter 0.25mm. Hence, any image detail smaller than 0.25mm in a well illuminated subject cannot be distinguished by a normal unaided eye at a distance of 250 mm, and hence details finer than this are not required in the final print. This definition is often used as the basis in lens designs, calculation of depth of field tables, and determination of screen sizes for printing. The circle of confusion is also approximately equal to *f/1000*.

The depth of field (DOF) can be approximated as [50-52]

$$DOF = \begin{cases} \dfrac{2N\hat{c}f^2 s_0^2}{f^4 - N^2\hat{c}^2 s_0^2} & \text{if } s_0 < \dfrac{f^2}{N\hat{c}} \\ 2N\hat{c}\dfrac{m+1}{m^2} & \text{if } s_0 \geq \dfrac{f^2}{N\hat{c}} \end{cases} \tag{2.2.6}$$

where $\hat{c}$ is the diameter of the circle of confusion, s_0 is the subject distance, and m is the image magnification.

For a camera-based non-contact vibration measurement system, the focal length of each camera after calibration needs to be fixed in order to use the calibrated, known focal length with triangulation techniques for finding the physical coordinates of measured points. Hence, the only parameter can change the depth of field is the aperture diameter *d*. As shown in Figure 2.3 and (2.2.6), a larger *f*-number means a smaller diameter of effective aperture, and it results in a larger depth of field.

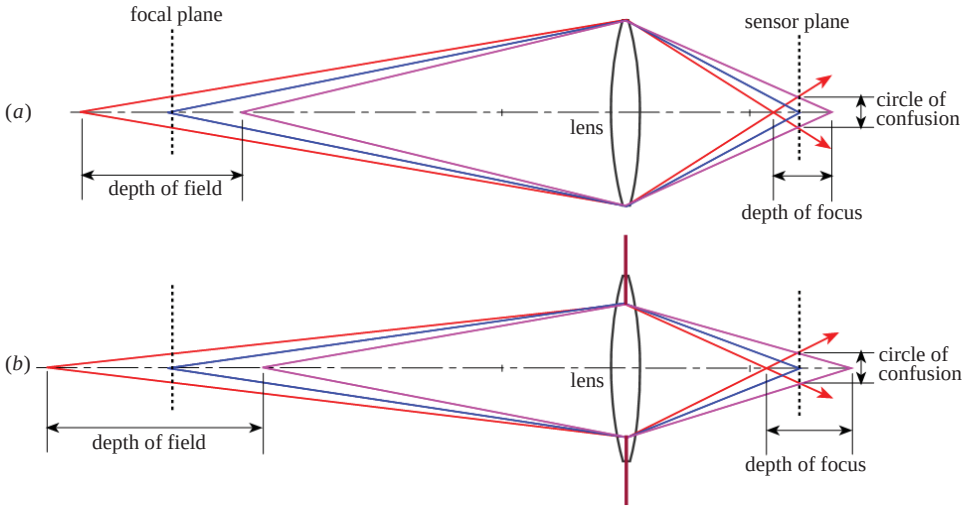


Figure 2.3 (a) a wide diameter of effective aperture, (b) a narrow diameter of effective aperture.

In Figure 2.3, the diameter of the circle of confusion is fixed. Hence, when the aperture diameter decreases, the depth of focus and the depth of field increase.

2.2.4 Discussions

It follows from the discussions presented in Section 2.2.2 that a clear image without paths of movement can be captured only by using a high shutter speed. And, Section 2.2.3 reveals that a large depth of field is needed for dynamic problems in order to have sharp projections for a wide range of movement, but it requires the use of a small aperture diameter. Unfortunately, a high shutter speed with a small aperture diameter results in low exposure of the sensor to light. Hence, for a camera-based non-contact vibration measurement system one needs to use a high ISO speed in order to have clear images of a vibrating object under measurement.

An image stabilizer is to prevent digital photos from being blurred. It reduces camera shake caused by hand movement when a slow shutter speed is used or a long telephoto lens is used without a tripod. Almost all cameras today have some form of image stabilization, including DSLR cameras. DSLR image stabilization is either built into the camera body or interchangeable lens. Since the image stabilizer is difficult to be modeled in a camera-based measurement system, it is better to be turned off during dynamic measurement.

2.3 Corner Detection

For the calibration of a camera-based measurement system and for using such a system for measurement, the first step is to perform *corner detection* to extract the image coordinates of interest corner points from a 2D image. To extract the image coordinates of an image point from a 2D image, a corner detection method is needed. Corner detection is used in computer vision to extract certain kinds of features and infer the contents of an image. Corner detection is frequently used in motion detection, image matching, video tracking, image mosaicing, panorama stitching, 3D modeling, and object recognition. Corner detection overlaps with the topic of interest point detection. An interest point can be a corner, an isolated point of local intensity maximum or minimum, a line end, or a point on a curve where the curvature is locally maximal. The quality of a corner detector is determined by its ability to detect the same corner in multiple similar images under different conditions of lighting, translation, rotation, scale, and affine.

Removing corner points from an image impedes human recognition, but removing straight edges does not. Moreover, corner points are translation and rotation invariant,

and they are stable under varying lighting conditions. Hence, corner detection plays the key role in image processing. In a comparative study of different interest point detectors [30,31], the Harris corner detector [21] was proven to be the most repeatable and informative. As an improvement of the Harris corner detector, the Foerstener corner detector [24] is capable of providing sub-pixel accuracy in corner detection.

An image from today's ordinary digital cameras consists of three values at each pixel to indicate the amounts of red, green, and blue colors (i.e., the so-called RGB color space). For the purpose of extracting corner points without too much computational time, it is better to transform the three RGB values at each pixel into just one grey value.

2.3.1 Calculation of Grey Value Distribution

For obtaining the grey-value distribution of an RGB color image, we need to transform the RGB color space into another space that contains the grey value component, such as the Y component of the YIQ and YUV color spaces.

YIQ is a color space used by the NTSC (National Television Standards Committee) color TV system, employed mainly in North and Central America and Japan. In this space, Y component represents luminance (i.e., brightness or grey value), and it means achromatic image without any color. Moreover, I stands for in-phase, and Q stands for quadrature; they represent the chrominance information with color information. According to the U.S. Federal Communication Commission Rules and Regulations, I-channel bandwidth is at 1.3 MHz less than 2 dB down, and at 3.6 MHz at least 20 dB down; Q-channel bandwidth is at 400 kHz less than 2 dB down, at 500 kHz less than 6 dB down, and at 600 kHz at least 6 dB down [53]. For human eye's sensitivity, I-channel

bandwidth is in the color range from orange to blue; and Q-channel bandwidth is in the range from purple to green.

According to Rec. ITU-R BT.709 [54], the formula used to calculate luminance in analog video systems like high-definition television (HDTV) is

$$Y = 0.2126R + 0.7152G + 0.0722B \tag{2.3.1}$$

However, transitional 1035i HDTV formats may use the SMPTE 240M coefficients as

$$Y = 0.212R + 0.701G + 0.087B \tag{2.3.2}$$

According to Rec. ITU-R BT.601 [55], the formula used to calculate luminance in digital video signals is

$$Y = 0.299R + 0.587G + 0.114B \tag{2.3.3}$$

Since we will use digital images, we follow Rec. ITU-R BT.601 [55] and adopt the following matrix for transforming RGB to YIQ:

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.595716 & -0.274453 & -0.321263 \\ 0.211456 & -0.522591 & 0.311135 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{2.3.4}$$

where $R, G, B, Y \in [0,1]$, $I \in [-0.5957, 0.5957]$, and $Q \in [-0.5226, 0.5226]$.

YUV is another popular color space and is used in the PAL (Phase Alternating Line) color TV system (wildly used by European and many other countries), where U represents blue–luminance difference and V represents red–luminance difference. Its transform matrix is given by

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.14713 & -0.28886 & 0.436 \\ 0.615 & -0.51499 & -0.10001 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{2.3.5}$$

where $R, G, B, Y \in [0,1]$, $U \in [-0.436, 0.436]$, and $V \in [-0.615, 0.615]$.

There are some other color spaces, including YDbDr (used in the SECAM color TV system), YPbPr (used in video electronics), and YCbCr (used as a part of the color image pipeline in video and digital photography systems). Although these different color spaces have different second and third components, they all have the same Y component [55], as shown in (2.3.3).

Because we will use gradients of grey values to detect corner points, the absolute grey value at each pixel is not important. Hence, one can redefine Y as $\tilde{Y} = 0.5094R + G + 0.1942B$ by dividing (2.3.3) by 0.587 in order to reduce one multiplication operation in calculating the grey value at each pixel. For example, the calculation of all Y values for an image of 3456×2304 pixels takes 0.3704 CPU seconds by using (2.3.3), but it takes 0.3172 CPU seconds for all $\tilde{Y}$ values. Since the difference between the two computational times is not much, we still recommend the use of (2.3.3).

2.3.2 Corner Detection Using Gradients of Grey Values

Moravec [20] defined a corner to be a point with low self-similarity, and similarity was defined as the sum of squared differences S_{SD} between a patch centered on the pixel and another nearby patch. If the pixel is in a region of uniform intensity, then the nearby patches will look similar and S_{SD} will be low. If the pixel is on an edge, nearby patches in a direction perpendicular to the edge will look quite different and S_{SD} will be high, but nearby patches along the edge will look similar and S_{SD} will be low. If the pixel has large S_{SD} values along all directions, it is a corner. Unfortunately, the major problem

of this corner detector is that too many nearby patches along different directions need to be examined in order to have robust detection results.

Harris and Stephens [21] improved upon Moravec's corner detector [56] and proposed to use local gradients of grey-value distribution for corner detection. For a grayscale 2D image with a grey-value distribution I, an image patch centered at (u,v) and another patch centered at $(u+x,v+y)$ are considered and both patches are over an area $2m\Delta_1 \times 2n\Delta_2$ (e.g., $m=n=5$), where Δ_1 and Δ_2 are the known horizontal and vertical distances between two adjacent pixels. The weighted *sum of squared differences* S_{SD} between these two patches is given by

$$\begin{aligned} \mathrm{S}_{SD}(x,y) &= \sum_{i=-m}^{m}\sum_{j=-n}^{n} w_{ij}[I(u+i\Delta_1+x, v+j\Delta_2+y) - I(u+i\Delta_1, v+j\Delta_2)]^2 \\ &\approx \sum_{i=-m}^{m}\sum_{j=-n}^{n} w_{ij}[I_x(u+i\Delta_1, v+j\Delta_2)x + I_y(u+i\Delta_1, v+j\Delta_2)y]^2 \qquad (2.3.6) \\ &= \{x\ y\}[G]\begin{Bmatrix} x \\ y \end{Bmatrix} \end{aligned}$$

$$[G] \equiv \sum_{i=-m}^{m}\sum_{j=-n}^{n} w_{ij}\begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix} \equiv \begin{bmatrix} \langle I_x^2 \rangle & \langle I_xI_y \rangle \\ \langle I_xI_y \rangle & \langle I_y^2 \rangle \end{bmatrix} \qquad (2.3.7)$$

where $I_x \equiv \partial I / \partial x$, $[G]$ is called the Harris matrix, and angle brackets denote weighted averaging. Because the pixels in a 2D image are discrete, one may consider the right hand side of (2.3.6) as a shape weighted by the coordinate variables of known points. If a circular window is used, then the response will be isotropic.

A corner (or, in general, an interest point) is characterized by a large variation of S_{SD} in all directions of the vector $\{x,y\}$. The two eigenvalues λ_i of $[G]$ can determine the character of the pixel at (u,v), as shown in Appendix A. If both λ_i are close to zero,

the pixel is in a region of uniform intensity and is not a corner point. If $|\lambda_1| >> 0$ and $\lambda_2 \approx 0$, there is an edge along the direction of the eigenvector $\{x, y\}_2$. If $|\lambda_1| >> 0$ and $|\lambda_2| >> 0$, there is a corner at the pixel.

2.3.3 Weighting Function

In order to reduce noise and change the operator to be isotropic, it is better to locally smooth the grey value distribution. One can use the Laplacian of Gaussian (LoG) filter as the weighting operator w_{ij} [56,57]. When an input image $f(x, y)$ is given, it can be convolved with a Gaussian Kernel $G_\sigma(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$, where σ is a scale parameter. To extract a fine edge, a small value for σ needs to be used.

According to the definition of convolution, the derivative of a single-variable function $f(t)$ convoluted with a weighting function $h(t)$ is given by

$$\begin{aligned} \frac{d}{dt}\left[h(t) * f(t)\right] &= \frac{d}{dt}\int f(\tau)h(t-\tau)d\tau = \int f(\tau)\frac{d}{dt}h(t-\tau)d\tau \\ &= f(t) * \frac{d}{dt}h(t) \end{aligned} \tag{2.3.8}$$

Similarly, one can show that

$$\nabla^2\left[G_\sigma(x, y) * f(x, y)\right] = LoG * f(x, y), \quad LoG \equiv \nabla^2 G_\sigma(x, y) \tag{2.3.9}$$

where

$$LoG = \nabla^2 G_\sigma(x, y) = -\frac{1}{\pi\sigma^4}\left[1 - \frac{x^2 + y^2}{2\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2.3.10}$$

Hence, one can replace the use of weighting functions in (2.3.7) as

$$
\begin{aligned}
w_{ij} I_x^2 &\rightarrow LoG * I_x^2 \\
w_{ij} I_y^2 &\rightarrow LoG * I_y^2 \\
w_{ij} I_x I_y &\rightarrow LoG * (I_x I_y)
\end{aligned}
\tag{2.3.11}
$$

LoG can be approximated by simple convolution kernels like mask matrices. In order to make sure the convolution result of a homogeneous region is always zero, the summation or average of all elements of the kernel has to be zero. For corner detection, because we only compare grey values or gradients of grey values, it is not necessary to force the summation (or average) of all elements of the kernel to be zero. For convenience and computational efficiency, one can set all elements to be positive, with the central element being one and others being less than or equal to one. Hence, one can replace the use of weighting functions in (2.3.7) as

$$
\begin{aligned}
w_{ij} I_x^2 &\rightarrow mask(i\Delta_1, j\Delta_2) \times I_x^2 \\
w_{ij} I_y^2 &\rightarrow mask(i\Delta_1, j\Delta_2) \times I_y^2 \\
w_{ij} I_x I_y &\rightarrow mask(i\Delta_1, j\Delta_2) \times (I_x I_y)
\end{aligned}
\tag{2.3.12}
$$

where $(i\Delta_1, j\Delta_2)$ is a local location with respect to (u, v) in the pixel patch. In other words, we only need to smooth a small area around the initial guess corner to save computational time. The following are two recommended masks from [42]:

$$w_{ij} = e^{-i^2/25} e^{-j^2/25}, \quad i, j = -5,...,5$$

$$[w_{ij}] = \begin{bmatrix} 0.1353 & 0.1940 & 0.2567 & 0.3135 & 0.3535 & 0.3679 & 0.3535 & 0.3135 & 0.2567 & 0.1940 & 0.1353 \\ 0.1940 & 0.2780 & 0.3679 & 0.4493 & 0.5066 & 0.5273 & 0.5066 & 0.4493 & 0.3679 & 0.2780 & 0.1940 \\ 0.2567 & 0.3679 & 0.4868 & 0.5945 & 0.6703 & 0.6977 & 0.6703 & 0.5945 & 0.4868 & 0.3679 & 0.2567 \\ 0.3135 & 0.4493 & 0.5945 & 0.7261 & 0.8187 & 0.8521 & 0.8187 & 0.7261 & 0.5945 & 0.4493 & 0.3135 \\ 0.3535 & 0.5066 & 0.6703 & 0.8187 & 0.9231 & 0.9608 & 0.9231 & 0.8187 & 0.6703 & 0.5066 & 0.3535 \\ 0.3679 & 0.5273 & 0.6977 & 0.8521 & 0.9608 & 1.0000 & 0.9608 & 0.8521 & 0.6977 & 0.5273 & 0.3679 \\ 0.3535 & 0.5066 & 0.6703 & 0.8187 & 0.9231 & 0.9608 & 0.9231 & 0.8187 & 0.6703 & 0.5066 & 0.3535 \\ 0.3135 & 0.4493 & 0.5945 & 0.7261 & 0.8187 & 0.8521 & 0.8187 & 0.7261 & 0.5945 & 0.4493 & 0.3135 \\ 0.2567 & 0.3679 & 0.4868 & 0.5945 & 0.6703 & 0.6977 & 0.6703 & 0.5945 & 0.4868 & 0.3679 & 0.2567 \\ 0.1940 & 0.2780 & 0.3679 & 0.4493 & 0.5066 & 0.5273 & 0.5066 & 0.4493 & 0.3679 & 0.2780 & 0.1940 \\ 0.1353 & 0.1940 & 0.2567 & 0.3135 & 0.3535 & 0.3679 & 0.3535 & 0.3135 & 0.2567 & 0.1940 & 0.1353 \end{bmatrix} \tag{2.3.13}$$

$$w_{ij} = \frac{1}{X_{ij}^2 + X_{ji}^2}, \quad i, j = -5,...,5$$

$$[w_{ij}] = \begin{bmatrix} 0.0200 & 0.0244 & 0.0294 & 0.0345 & 0.0385 & 0.0400 & 0.0385 & 0.0345 & 0.0294 & 0.0244 & 0.0200 \\ 0.0244 & 0.0313 & 0.0400 & 0.0500 & 0.0588 & 0.0625 & 0.0588 & 0.0500 & 0.0400 & 0.0313 & 0.0244 \\ 0.0294 & 0.0400 & 0.0556 & 0.0769 & 0.1000 & 0.1111 & 0.1000 & 0.0769 & 0.0556 & 0.0400 & 0.0294 \\ 0.0345 & 0.0500 & 0.0769 & 0.1250 & 0.2000 & 0.2500 & 0.2000 & 0.1250 & 0.0769 & 0.0500 & 0.0345 \\ 0.0385 & 0.0588 & 0.1000 & 0.2000 & 0.5000 & 1.0000 & 0.5000 & 0.2000 & 0.1000 & 0.0588 & 0.0385 \\ 0.0400 & 0.0625 & 0.1111 & 0.2500 & 1.0000 & 1.0000 & 1.0000 & 0.2500 & 0.1111 & 0.0625 & 0.0400 \\ 0.0385 & 0.0588 & 0.1000 & 0.2000 & 0.5000 & 1.0000 & 0.5000 & 0.2000 & 0.1000 & 0.0588 & 0.0385 \\ 0.0345 & 0.0500 & 0.0769 & 0.1250 & 0.2000 & 0.2500 & 0.2000 & 0.1250 & 0.0769 & 0.0500 & 0.0345 \\ 0.0294 & 0.0400 & 0.0556 & 0.0769 & 0.1000 & 0.1111 & 0.1000 & 0.0769 & 0.0556 & 0.0400 & 0.0294 \\ 0.0244 & 0.0313 & 0.0400 & 0.0500 & 0.0588 & 0.0625 & 0.0588 & 0.0500 & 0.0400 & 0.0313 & 0.0244 \\ 0.0200 & 0.0244 & 0.0294 & 0.0345 & 0.0385 & 0.0400 & 0.0385 & 0.0345 & 0.0294 & 0.0244 & 0.0200 \end{bmatrix} \tag{2.3.14}$$

where

$$[X_{ij}] = \begin{bmatrix} -5 & -4 & -3 & -2 & -1 & 1 & 1 & 2 & 3 & 4 & 5 \\ -5 & -4 & -3 & -2 & -1 & 1 & 1 & 2 & 3 & 4 & 5 \\ -5 & -4 & -3 & -2 & -1 & 1 & 1 & 2 & 3 & 4 & 5 \\ -5 & -4 & -3 & -2 & -1 & 1 & 1 & 2 & 3 & 4 & 5 \\ -5 & -4 & -3 & -2 & -1 & 1 & 1 & 2 & 3 & 4 & 5 \\ -5 & -4 & -3 & -2 & -1 & 1 & 1 & 2 & 3 & 4 & 5 \\ -5 & -4 & -3 & -2 & -1 & 1 & 1 & 2 & 3 & 4 & 5 \\ -5 & -4 & -3 & -2 & -1 & 1 & 1 & 2 & 3 & 4 & 5 \\ -5 & -4 & -3 & -2 & -1 & 1 & 1 & 2 & 3 & 4 & 5 \\ -5 & -4 & -3 & -2 & -1 & 1 & 1 & 2 & 3 & 4 & 5 \\ -5 & -4 & -3 & -2 & -1 & 1 & 1 & 2 & 3 & 4 & 5 \\ -5 & -4 & -3 & -2 & -1 & 1 & 1 & 2 & 3 & 4 & 5 \end{bmatrix} \quad (2.3.15)$$

We choose to use the mask matrix shown in (2.3.13).

2.3.4 Sub-pixel Corner Detection

A picture under corner detection processing always contains some noisy points that confuse the corner detector in finding interest points. There are two ways to release this problem. One way is to use a piece of paper or fabric to build a background without noisy points, and the other way is to manually choose a small local area on the 2D image plane for detecting each corner point. We choose to use the second method to roughly locate a corner point, and then use the Foerstner algorithm [24] to refine the corner location with sub-pixel accuracy, as shown next.

Since we use pictures of a rectangular checker board with black and white squares for camera calibration, for a 2D image of the rectangular checker board, one can provide initial location estimations of the four outer corners by clicking the computer mouse (with the use of the "*ginput*" command of MATLAB in the program) roughly at the four outer corners on the image plane. If a white square is at one of the four corners, the

corner point may not appear in the image. Then, we consider the second outline for a corner point. The initial location estimations of the many other inner corners can be calculated using the method presented later in Section 2.3.5.

Because each corner location picked by the mouse through the "*ginput*" command is described by two decimal numbers but each pixel location on an image is described by two integers, the one picked by the mouse may not coincide with a pixel. One way is to use the summation of weighted grey values of the pixels close to the clicked corner location to represent its grey value, which is computationally expensive. Another way is to replace the picked corner location with a nearest pixel location. This way is computationally efficient and is highly recommended.After a corner point on a 2D image is roughly located, one can use the following iteration method to refine the corner location estimation.

After a pixel is identified using the eigenvalues of [G] to be possibly a corner point, one can use the following iteration method to find a nearby, actual corner point. It follows from (2.3.6) that

$$\frac{\partial S_{SD}}{2\partial x} \approx \sum_{i=-m}^{m} \sum_{j=-n}^{n} w_{ij}(I_x^2 x + I_x I_y y) = \left\langle I_x^2 \right\rangle x + \left\langle I_x I_y \right\rangle y \tag{2.3.16}$$

Similarly, we have

$$\frac{\partial S_{SD}}{2\partial y} \approx \sum_{i=-m}^{m} \sum_{j=-n}^{n} w_{ij}(I_x I_y x + I_y^2 y) = \left\langle I_x I_y \right\rangle x + \left\langle I_y^2 \right\rangle y \tag{2.3.17}$$

Equations (2.3.16) and (2.3.17) can be rewritten as

$$[G]\begin{Bmatrix} x \\ y \end{Bmatrix} = \{B\},\ \{B\} \equiv \begin{Bmatrix} \sum_{i=-m}^{m}\sum_{j=-n}^{n} w_{ij}(I_x^2 x + I_x I_y y) \\ \sum_{i=-m}^{m}\sum_{j=-n}^{n} w_{ij}(I_x I_y x + I_y^2 y) \end{Bmatrix} \tag{2.3.18}$$

With an initial guess $\{x, y\} = \{x_0, y_0\}$, one can estimate and update the nearby, actual corner point to be at

$$\begin{Bmatrix} x \\ y \end{Bmatrix} = [G]^{-1}\{B\} = \frac{1}{|G|}\begin{bmatrix} \langle I_y^2 \rangle & -\langle I_x I_y \rangle \\ -\langle I_x I_y \rangle & \langle I_x^2 \rangle \end{bmatrix} \begin{Bmatrix} \sum_{i=-m}^{m}\sum_{j=-n}^{n} w_{ij}(I_x^2 x_0 + I_x I_y y_0) \\ \sum_{i=-m}^{m}\sum_{j=-n}^{n} w_{ij}(I_x I_y x_0 + I_y^2 y_0) \end{Bmatrix} \tag{2.3.19}$$

One may consider the right hand side of (2.3.19) as an intersection of two direction lines. Because the grey value of a corner point is locally extreme, the answer $\{x, y\}$ will converge by iterations to the exact corner position where S_{SD} has an extreme value and $S_{SDx} = S_{SDy} = \{x, y\}[G]\{x, y\}^T = 0$. As shown in Appendix A, if $\lambda_1 >> \lambda_2$ (e.g., $\lambda_1 > \lambda_2 / 50$) in the *i*th iteration step, the pixel is considered to be on an edge. In order to have a faster convergence in the next step, one can search for the next solution along the first eigenvector of the current step. This is the so-called Foerstner algorithm [24], and it is based on the least-squares fitting.

For a set of consecutive images of a dynamic object, with the interest points identified by a corner detector from the first image, each interest point's next location can be predicted using the frame rate and current velocity in order to narrow down the area for searching the point in the next image. To improve the accuracy and efficiency one can even use a structural model of the system to predict each interest point's next location.

2.3.5 Finding Inner Corners

As shown in Section 2.3.4, locations of the four outer corners of an image of the checker board can be roughly picked by the mouse and then refined by the Foerstner algorithm [24]. For camera calibration, unfortunately, because at least five corner points are needed (shown later in Chapter 3), at least one inner point is necessary, and more inner points are needed for accurate and robust camera calibration. Because the 2D image of a rectangular checker board is distorted from a rectangle by projection and lens distortion, it is difficult even just to roughly estimate the locations of inner corners before the intrinsic and extrinsic parameters of a camera are unknown.

The number of squares on the checker board needs to be known in order to roughly estimate the locations of all inner corners on the image plane. The first step is to find the number of squares along each of its two edge directions. After finding the exact locations of the four outer corners, the four straight lines between these four outer corners and the number of squares along each of these lines can be determined, as shown next.

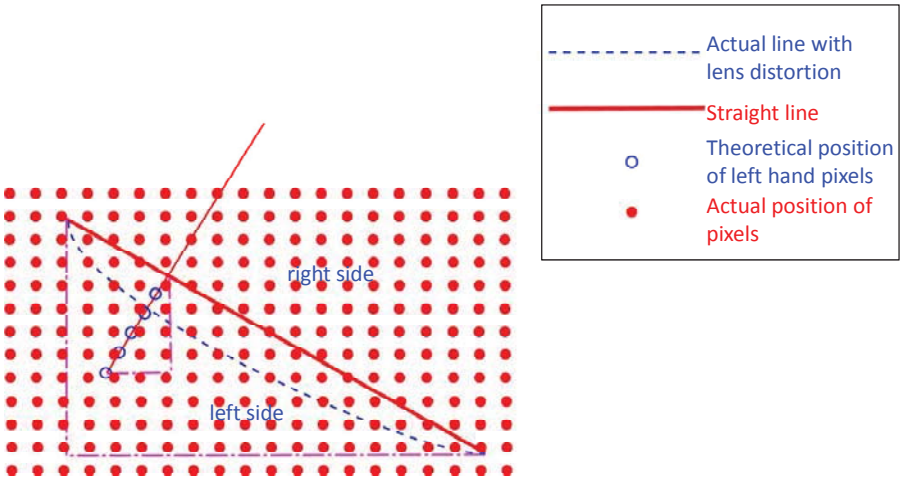


Figure 2.4 Theory for finding the number of squares.

Figure 2.4 shows a method to determine whether a pixel on (or near to) an outer corner line between two outer corners is on the boundary of two adjacent black and white squares. On (or near to) the line normal to the outer corner line at the pixel, five (or more) pixels on the left side and five on the right side of the outer corner line are chosen for calculation. The grey values of the five left pixels are multiplied by -1 and those of five right pixels are multiplied by +1, and then the sum $\tilde{S}$ of these weighted grey values is used to determine whether the pixel is on the boundary of two adjacent black and white squares. If the sign of $\tilde{S}_i$ of the *i*th pixel is different from the sign of $\tilde{S}_{i+1}$ of the (i+1)th pixel, there is a boundary of two adjacent black and white squares. Because a white pixel has a grey value larger than that of a black pixel, the left side of the *i*th pixel is a white square if $\tilde{S}_i < 0$. Similarly, the left side of the (i+1)th pixel is a black square if $\tilde{S}_{i+1} > 0$. Hence, if $\tilde{S}_i < 0$ and $\tilde{S}_{i+1} > 0$, the left side of the outer corner line changes from a white square to a black square. Numerical results show that this method is pretty robust even with image distortion.

Without processing pixels around the two ends of the outer corner line, the number of squares is equal to the number of boundaries plus one. If the number of squares on one outer corner line is different from that on the other parallel outer corner line, it is better to find out by visual inspection and then input the number of squares manually. Although the number of squares can be numerically counted using the above method, the actual physical size of squares needs to be input manually as a required data for camera calibration.

After the numbers of squares along two orthogonal outer corner lines of the checker board are obtained, locations of inner corners need to be roughly estimated and then refined to have sub-pixel accuracy. Because the *xy* plane in Figure 1.2 is assumed to be on the checker board, we have z=0 and it follows from (1.1.1) that

$$\zeta\{m\} = [H]\{M\}, \{m\} \equiv \begin{Bmatrix} u \\ v \\ 1 \end{Bmatrix}, \{M\} \equiv \begin{Bmatrix} x \\ y \\ 1 \end{Bmatrix}, [H] \equiv \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} T_{11} & T_{12} & \xi_0 \\ T_{21} & T_{22} & \eta_0 \\ T_{31} & T_{32} & \zeta_0 \end{bmatrix} \quad (2.3.20)$$

Because the homography matrix $[H]$ has 9 unknowns, the minimum number of object points needed for a unique solution of $[H]$for each image is 5. However, because the third row of $[H]$ do not affect the values of the image plane coordinates *u* and *v*, one can set $\zeta_0 = 1$ to reduce the number of unknowns and then find a homography matrix for estimating the initial guesses of inner corners. Then one can use the method presented in Section 2.3.4 to refine the locations of inner corners with sub-pixel accuracy.

2.3.6 Image Noise

Image noise is often related to electronic noise and is undesirable random information in images. It can be produced by the sensor and circuitry of a digital camera. Image noises can be signal-independent or signal-dependent. Signal-independent noise is a random set of image data added to the image pixels. This kind of noise occurs when an image is transmitted electronically from one place to another. While signal-dependent noise has a noise level at each pixel being a function of the pixel's grey value [58].

The following are some examples of signal-independent noise. *Amplifier noise* (also known as the Gaussian noise) often happens in dark areas of the image [59]. The

main part of amplifier noise is *Johnson-Nyquist noise* [60-62]. It is not related to the applied voltage, but is generated by the thermal agitation of the charge carriers inside an electrical conductor at equilibrium. In an idealistic resistor, the thermal noise is approximately white, and the amplitude of the signal almost follows the Gaussian probability density function [63]. For a CMOS sensor, Johnson-Nyquist noise includes the noise from resetting capacitors, called the *kTC noise* [64]. Because typical digital camera sensors (i.e., charge-coupled device (CCD) or CMOS) normally are less sensitive to blue light, the blue color channel is amplified more than green and red ones. It results in more significant noise in the blue channel for color cameras [65]. Figure 2.5 is an image captured under dim light, which has an obvious blue channel noise problem. *Salt-and-pepper noise* is displayed as dark pixels in bright regions and bright pixels in dark regions. It results from transmission and analog to digital conversion [66,67]. Because of photodiode leakage currents, a slower shutter speed will increase this kind of noise. *Shot noise* is another signal-independent noise. It can be caused by statistical quantum fluctuations and the dark leakage current in the image sensor. It has a root-mean-square value proportional to the square root of image intensity, and follows a Poisson distribution [65]. A low ISO speed will reduce the shot noise due to less amplification of signal.

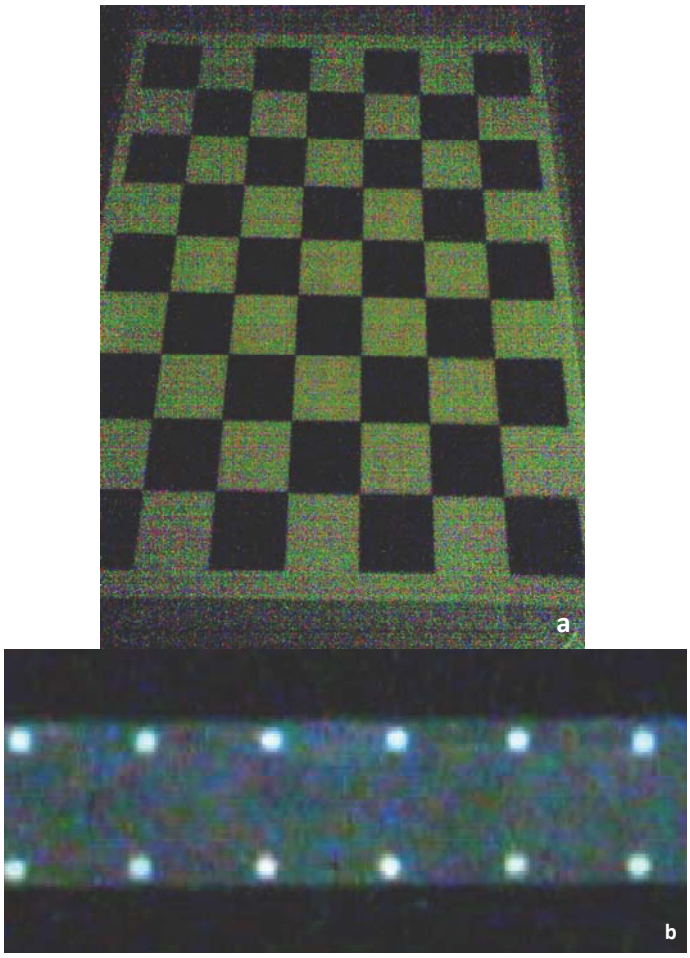


Figure 2.5 Images with blue channel noise: a) an image taken by a Nokia phone camera, and b) a screenshot of a video frame taken by a Canon EOS-7D camera.

Signal-dependent noise includes quantization noise and film grain. *Quantization noise* (uniform noise) is generated by quantizing pixels from a sensed image. It is uniformly distributed unless having big dithering caused by other noise sources. *Film grain* is caused by small particles developed from silver halides that have received enough photons. Although digital photography does not exhibit film grain, the image noise sometimes appears as grainy points. For digital SLR cameras it is better to work in

a low temperature environment, because the higher temperature can increase the noise caused by leakage [68].

2.3.7 Comparison of Numerical Results

This section shows and compares some results from corner detection. One thing needs to be mentioned is that the pixel coordinates extracted using the command "*imread*" of MATLAB are different from the point coordinates extracted using the command "*ginput*" of MATLAB and software like *paint*. The coordinate system for pixel coordinates and RGB values from "*imread*" has the first axis pointing downward and the second axis pointing to the right. But the coordinate system for point coordinates from "*ginput*" has the first axis pointing to the right and the second one pointing downward.

In Figure 2.6, all corner points are detected using grey values calculated from (2.3.3). The original image is a medium-size picture captured using a Canon EOS-7D digital SLR camera. But because of the limited computer memory, we reduce the resolution from 3456×2304 pixels to 1153×768 pixels. Figure 2.6a shows the original image with a white background. Figure 2.6b shows the possible corner points detected by the Harris corner detector presented in Section 2.3.2. It shows that several pixel locations are identified as possible corner points around each real corner, and Figure 2.7b clearly shows the reason. Figure 2.7b shows the distribution of grey values. White squares have larger grey values then black ones. However, gaps between two adjacent white squares exist, and the grey value significantly changes from a tip of one white square to the neighboring tip of another white square. That is why we cannot find a unique point around each corner using the Harris corner detector. Figure 2.6c shows the center of each

group of possible pixel locations around each real corner, and the method will be further discussed in Chapter 4. Figure 2.6d shows the corner points detected using the sub-pixel corner detection method presented in Sections 2.3.4 and 2.3.5. Note that the outer corners shown in Figure 2.6d are on the second outlines of the image shown in Figure 2.6a.

Figure 2.6b shows that the results from the Harris corner detector without initial guesses have some features of edges and/or noise. Figure 2.6c shows that the center finding method can converge to the actual corners, but some corner points can be missed. Figure 2.6d shows that the sub-pixel corner detection method with initial guess can find a unique location for each of all corner points.

Although just five arbitrary corner points are theoretically needed for camera calibration (see Chapter 3), the physical distance between two arbitrary corner points needs to be physically measured. Moreover, far more than 5 corner points are needed for deriving an accurate lens distortion model. Hence, we highly recommend the use of a checker board like that shown in Figure 2.6a with many equally separated squares with known sizes and the sub-pixel corner detection method with initial guess to find many corner points in order to have accurate camera calibration.

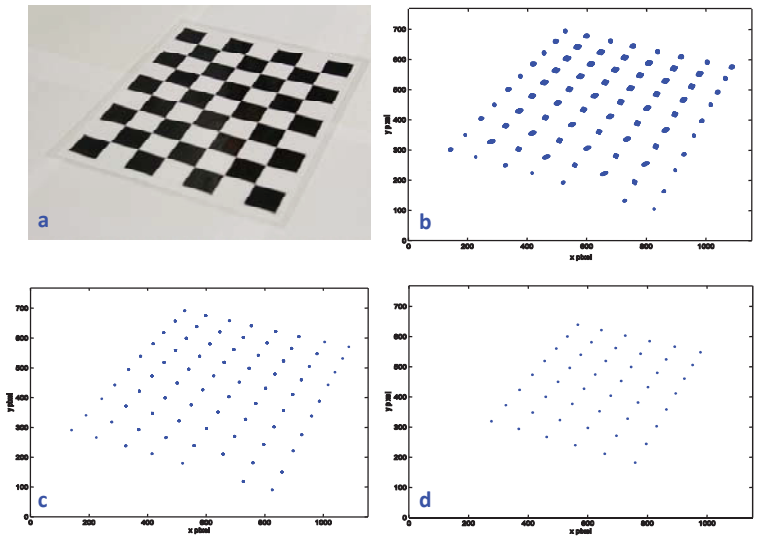


Figure 2.6 Extraction of corner points of a checker board image: a) original image, b) possible corner points detected by the Harris corner detector, c) the center of each group of corners in b), and d) corner points detected by the sub-pixel method.

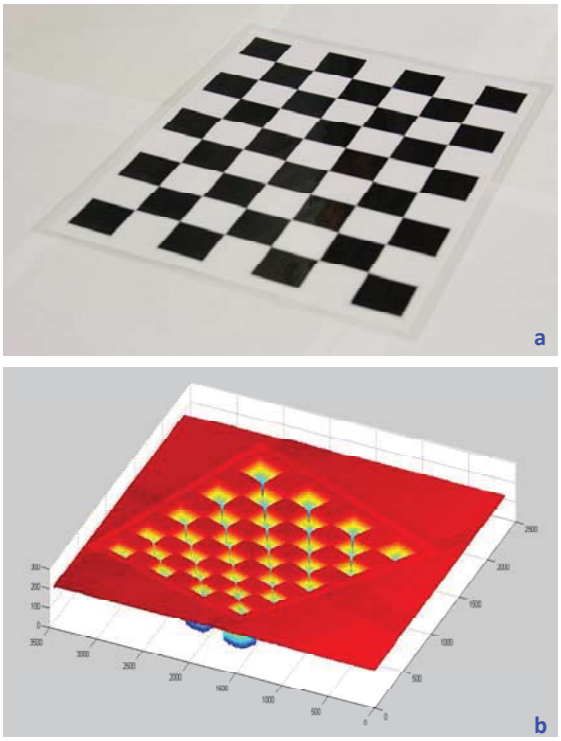


Figure 2.7 Distribution of grey values: a) original 3456×2304 image of the checker board, and b) magnitudes of grey values.

Figure 2.8 compares the distribution of grey values of Figure 2.7a calculated using $Y' = R + G + B$ with that using (2.3.3). Figures 2.8b and 2.8d show that, although $Y' = R + G + B$ is computationally efficient because of no multiplication operations, its distribution is not as smooth and sharp as that from using (2.3.3). Hence, we recommend the use of (2.3.3) for calculating grey values.

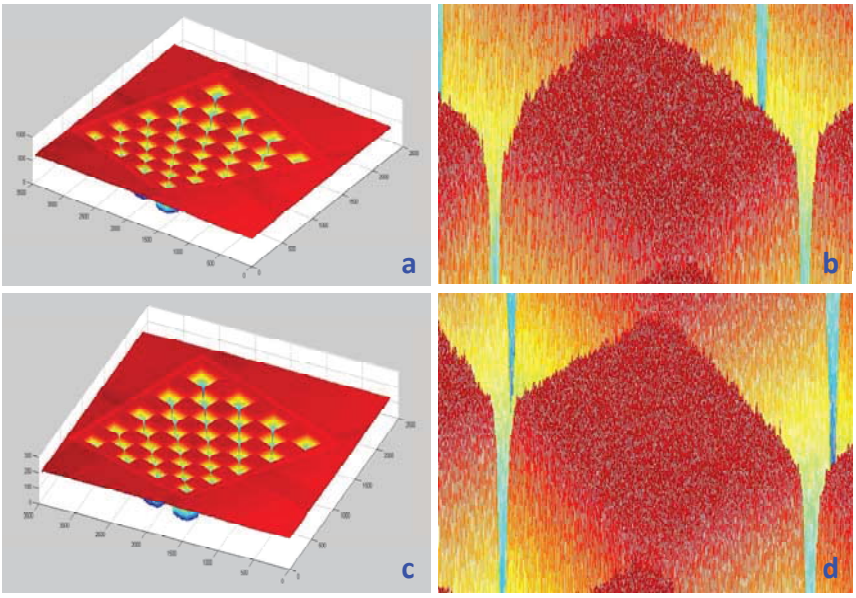


Figure 2.8 Distribution of grey values: a,b) $Y' = R + G + B$, and c,d) $Y = 0.299R + 0.587G + 0.114B$.

Figure 2.9 shows two pictures captured by the same camera using the same parameters but different ISO speeds. Figure 2.9b shows less grainy points and sharper lines. Hence, the use of a low ISO speed can increase the clarity for color images. However, for white areas (no color), our experiments reveal opposite results, as shown next.

The two pictures in Figures 2.10a and 2.10b are captured using the same 3456×2304 resolution, *f*-number 5, exposure time 1/320 s, and a focal length 28 mm. Figure 2.10 shows that the use of a higher ISO speed results in a smoother distribution of grey values over a white area.

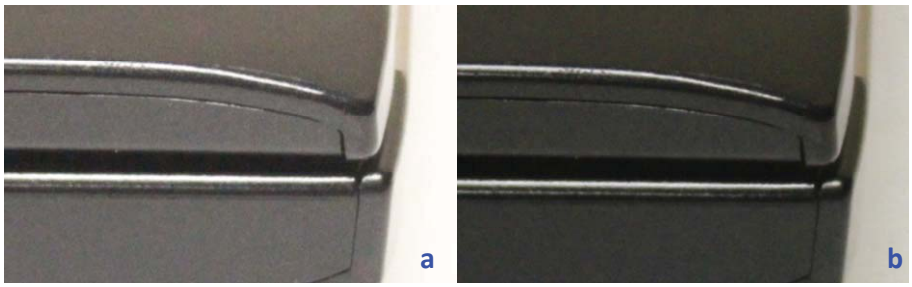


Figure 2.9 The image of a laptop's corner using an ISO speed of: a) 4000, and b) 1000.

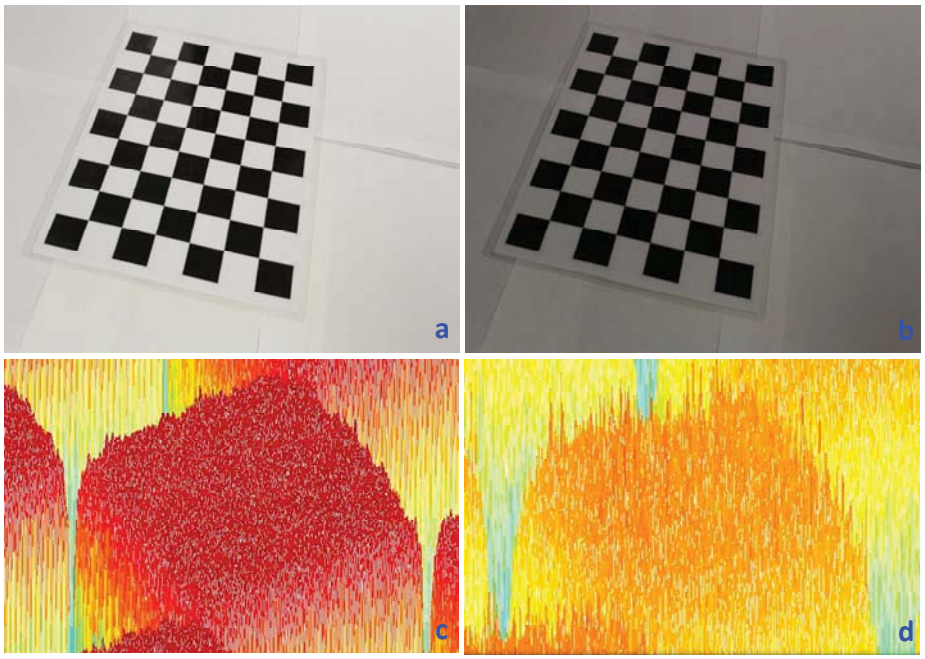
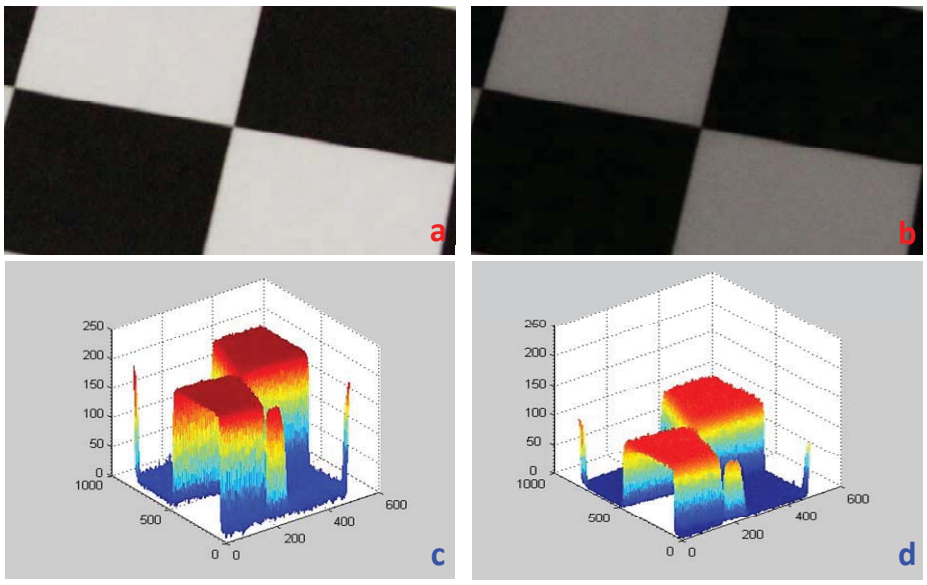


Figure 2.10 Distribution of grey values over white squares when the used ISO speed is: a,c) 4000, and b,d) 1000.

Figure 2.11 shows that a high ISO speed results in a smooth distribution of grey values over an area of high grey values, and a low ISO speed results in a smooth distribution of grey values over an area of low grey values. Note that grainy points appear in black areas of Figure 2.11a, and black points appear in white areas of Figure 2.11b.

For the design of our camera-based measurement system, markers with high image grey values will be the target points for motion tracking. Hence, we recommend the use of a high shutter speed, a large f-number (a small aperture diameter), and a high ISO speed in order to have high measurement accuracy.



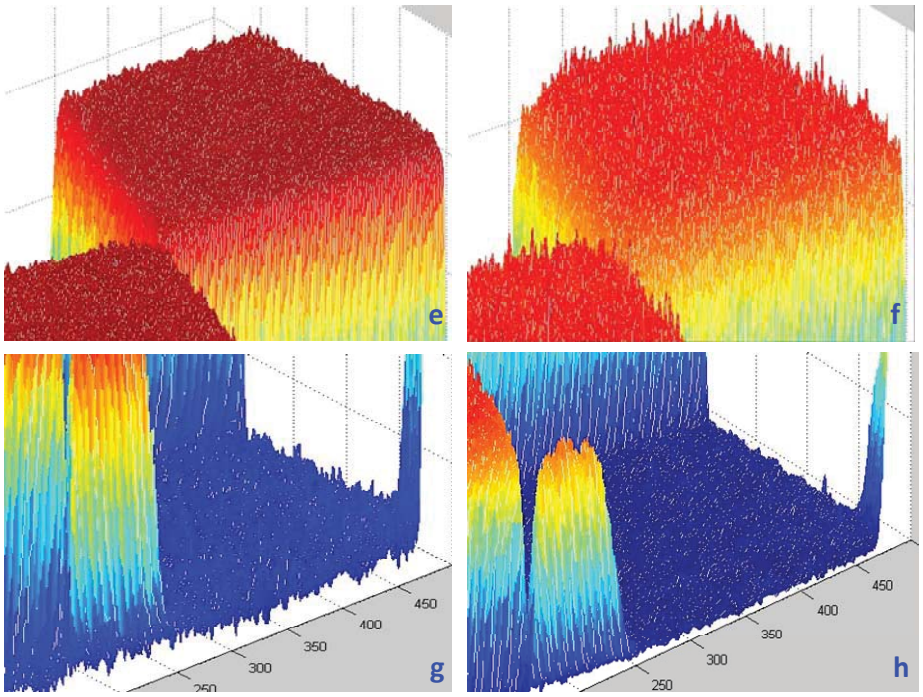


Figure 2.11 Distribution of grey values using an ISO speed of 4000 (a,c,e,g) and 1000 (b,d,f,h): a,b) images, c,d) grey values of a,b), e,f) grey values of white squares of c,d), and g,h) grey values of black squares of c,d).

CHAPTER 3

CAMERA CALIBRATION

In this chapter, a new flexible technique [19] for easy calibration of cameras without knowing the object's 3D geometry or computer vision is adopted and improved. This technique only requires observing a planer pattern at two or more independent orientations by moving the planar pattern or camera arbitrarily. Moreover, a lens distortion model for correcting radial and tangential distortions is also developed. The proposed procedure consists of a linear approximate estimation by closed-form solutions and then a nonlinear refinement based on the maximum likelihood estimation. Compared with classical techniques that use expensive calibration objects of two or three orthogonal planes, the proposed technique is easy to use and flexible.

This chapter also compares the accuracy of calibration results obtained from the use of different image formats and different image-recording qualities. It also introduces some utility software that can process images from a Canon EOS-7D DSLR camera, including one for extracting static images from a movie file.

A four-point fast calibration method applicable to any single rectangular pattern is also presented in this chapter. This method makes camera calibration more convenient and much easier. People can carry a small rectangular pattern to perform camera calibration and then noncontact measurements at anywhere and anytime.

3.1 Calibration Algorithm

3.1.1 Intrinsic Parameters

According to the coordinate systems shown in Figure 1.2 and (1.1.1), the intrinsic parameters of a camera include the focal length, aspect ratios, principal point, and skew factor as the linear part, and lens distortion parameters as the nonlinear part. In order to avoid unknown automatic change of the focal length, the camera's focus mode should be switched to the manual mode in calibration and in measurement. Moreover, each camera's focal length, ISO speed, aperture diameter, and shutter speed during the whole process of calibration and measurement need to be fixed using the manual mode.

The affine matrix $[A]$ shown in (1.1.1) contains five intrinsic parameters. α and β represent the image distance s_1 (see Figure 2.1) in terms of horizontal and vertical pixels, but many people consider them as the focal length in terms of pixels [69]. In the manual focus mode, the focal length can be changed manually by the zoom ring for coarse adjustment and the focusing ring for trimming. As shown in Section 2.2.1, because in measurement of large buildings and other civil structures $s_0 / s_1 \to \infty$, we have $f \approx s_1$ and hence $\alpha = s_1 / \Delta_1 \approx f / \Delta_1$ and $\beta = s_1 / \Delta_2 \approx f / \Delta_2$, where Δ_1 and Δ_2 are the horizontal and vertical distances between two adjacent pixels on the sensor plane, respectively. Moreover, (u_0, v_0) indicates the principal point on the sensor plane and is the image center, but it may be different from the fiducial center due to misalignment and/or distortion. Furthermore, γ is a skewness parameter of the two image axes, which is often zero for today's sensors. Without misalignment and distortion, the intrinsic parameters should be $u_0 = v_0 = \gamma = 0$, $\alpha = s_1 / \Delta_1$, and $\beta = s_1 / \Delta_2$. Figure 3.1 shows some possible

influences of erroneous intrinsic parameters. Figure 3.1a shows the effect of an aspect ratio of 2.0. For cameras with radially symmetric lenses, the aspect ratio should be 1.0, but recording and digitizing processes may change it.

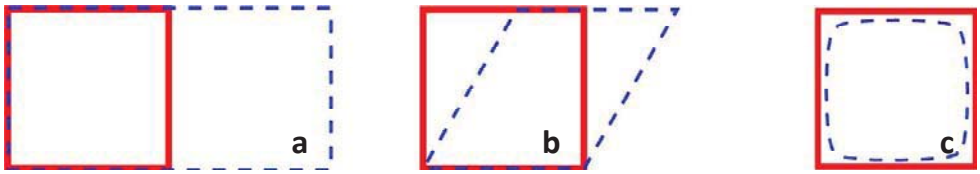


Figure 3.1 Solid lines show the correct images of a square and dotted lines show the images under incorrect intrinsic parameters: a) extension, b) skew, and c) distortion.

Optical aberration is performance departure of an optical system from the prediction of paraxial optics [70]. Optical aberration occurs when the light from one point of an object does not converge into a single point after going through an imaging system or one point in the image is not from a single point. The problem is not from flaws in optical elements but is due to the not completely simple paraxial theory [71]. For example, the nonlinear lens distortion shown in Figure 3.1c is a form of optical aberration. There are three major types of distortion: *radial distortion*, *decentering distortion*, and *thin prism distortion*.

Radial distortion is caused by constraints in lens manufacturing. Because a light beam passing through a lens' edges experiences severe bending due to imperfect periphery, a straight physical line often has a curved image on the sensor plane [72]. *Radial distortion* causes the inward or outward shift of image points from their perspective projections. If a camera's lenses are radially symmetric and on the same

optical axis, radial distortion is symmetric with respect to the principal point. There are two fundamental types of radial distortion: barrel and pincushion distortions. As shown in Figure 3.2a, barrel distortion causes an image bulging toward the edge, but it can be used to map an infinitely wide object plan into a finite image area [73]. Pincushion distortion causes an image shrinking toward the center (see Figure 3.2b), but it can be used to eliminate the globe effect [74]. A combination of both barrel and pincushion distortions results in the so-called mustache or complex distortion. Mustache distortion has barrel distortion close to the image center and pincushion distortion near periphery, as shown in Figure 3.2c. Mathematically speaking, the order of nonlinearity of barrel and pincushion distortions is quadratic and that of mustache distortion is quartic.

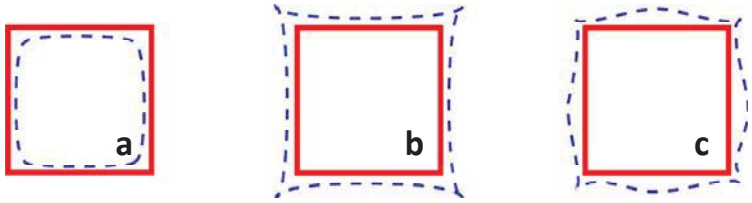


Figure 3.2 Radial distortions: a) barrel distortion, b) pincushion distortion, and c) mustache distortion.

Decentering distortion is caused by misalignment of the optical centers of assembled lenses and has both radial and tangential components [75]. Flaws in design, manufacturing, and assembly of lenses lead to errors in lens parallelism to the image plan and result in the so-called *thin prism distortion*. These distortions can be modeled as additional radial and tangential distortions to the optical system [75]. How to estimate image distortion and develop a nonlinear lens distortion model will be presented later in Section 3.1.5.

3.1.2 Extrinsic Parameters

There are 6 unknown extrinsic parameters in the matrix $[[T]\{\tau\}]$ in (1.1.1). As shown in Figure 1.2, $[T]$ represents the rotational coordinate transformation from the coordinate system *xyz* to the coordinate system $\xi\eta\zeta$, and $\{\tau\}(=\{\xi_0, \eta_0, \zeta_0\}^T)$ represents the translation from $\xi\eta\zeta$ to xyz.

The rotational matrix $[T]$ relates the coordinate systems xyz and $\xi\eta\zeta$ in the following form:

$$\begin{Bmatrix} \mathbf{i}_1 \\ \mathbf{i}_2 \\ \mathbf{i}_3 \end{Bmatrix} = [T] \begin{Bmatrix} \mathbf{i}_x \\ \mathbf{i}_y \\ \mathbf{i}_z \end{Bmatrix} \tag{3.1.1}$$

where $\mathbf{i}_1, \mathbf{i}_2$ and $\mathbf{i}_3$ are unit vectors along the axes ξ, η and ζ, and $\mathbf{i}_x, \mathbf{i}_y$ and $\mathbf{i}_z$ are unit vectors along the axes x, y and z, respectively. Because $[T]$ is a coordinate rotational matrix for two orthogonal coordinate systems, $[T]$ is a unitary matrix and

$$[T]^{-1} = [T]^T \tag{3.1.2}$$

The matrix $[T]$ can be presented in terms of three Euler angles [76,77]. For example, Figure 3.3 shows that the $\xi\eta\zeta$ system can be rotated to the *xyz* system by three sequential rotation angles θ_1, θ_2 and θ_3, where the angle θ_1 is rotated with respect to (w.r.t.) the ζ axis, θ_2 is w.r.t. the new ξ axis, and θ_3 is w.r.t. the new ζ axis (i.e., the *z* axis). In the reversed order, the *xyz* system can be rotated back to the $\xi\eta\zeta$ system by the same three angles $-\theta_3, -\theta_2$ and $-\theta_1$.

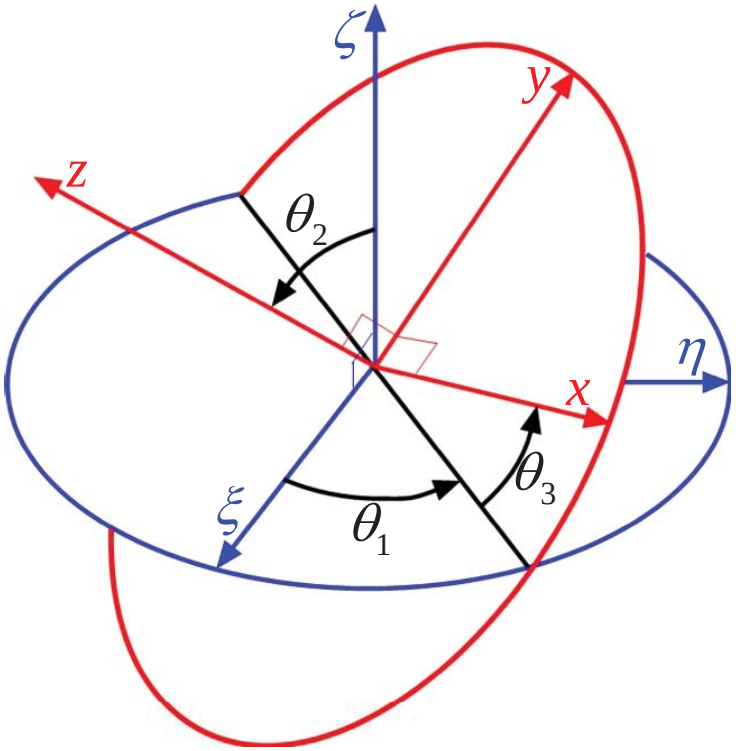


Figure 3.3 Three sequential Euler angles.

Hence, we have

$$[T]=\begin{bmatrix}\cos\theta_1 & -\sin\theta_1 & 0\\ \sin\theta_1 & \cos\theta_1 & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\ 0 & \cos\theta_2 & -\sin\theta_2\\ 0 & \sin\theta_2 & \cos\theta_2\end{bmatrix}\begin{bmatrix}\cos\theta_3 & -\sin\theta_3 & 0\\ \sin\theta_3 & \cos\theta_3 & 0\\ 0 & 0 & 1\end{bmatrix} \tag{3.1.3}$$

Alternatively, (3.1.3) can be written as

$$[T]\equiv\begin{bmatrix}T_{11} & T_{12} & T_{13}\\ T_{21} & T_{22} & T_{23}\\ T_{31} & T_{32} & T_{33}\end{bmatrix}=\begin{bmatrix}c_1c_3-s_1c_2s_3 & -c_1s_3-s_1c_2c_3 & s_1s_2\\ s_1c_3+c_1c_2s_3 & -s_1s_3+c_1c_2c_3 & -c_1s_2\\ s_2s_3 & s_2c_3 & c_2\end{bmatrix} \tag{3.1.4}$$

where $s_i\equiv\sin\theta_i$ and $c_i\equiv\cos\theta_i$ (i=1,2,3). It follows from (3.1.4) that

$$\tan\theta_1 = \frac{T_{13}}{-T_{23}},\ \cos\theta_2 = T_{33},\ \tan\theta_3 = \frac{T_{31}}{T_{32}} \qquad (3.1.5a,b,c)$$

$$\sin\theta_3 = -T_{12}\cos\theta_1 - T_{22}\sin\theta_1,\ \ \cos\theta_3 = T_{11}\cos\theta_1 + T_{21}\sin\theta_1 \qquad (3.1.5d,e)$$

$$\sin\theta_1\sin\theta_3 = \frac{T_{33}T_{11} - T_{22}}{1 - T_{33}^2} \qquad (3.1.5f)$$

$$\sin\theta_2 = T_{13}\sin\theta_1 - T_{23}\cos\theta_1 \qquad (3.1.5g)$$

Eqs. (3.1.5a) and (3.1.5c) reveal the singularity problems of the use of Euler angles when T_{23}=0 and $T_{32} = 0$.

If $[T]$ is known, (3.1.5a) gives two answers $\theta_1 = \theta_{11}$ and $\theta_1 = \theta_{12}$ for θ_1. Substituting $\theta_1 = \theta_{11}$ into (3.1.5d) and (3.1.5e) yields a unique answer $\theta_3 = \theta_{31}$. If θ_{31} agrees well with (3.1.5c) (and (3.1.5f)), $\theta_1 = \theta_{11}$ and $\theta_3 = \theta_{31}$ are the answers for θ_1 and θ_3, respectively. If θ_{31} doesn't agree well with (3.1.5c), $\theta_1 = \theta_{12}$ and $\theta_3 = \theta_{32}$ (from (3.1.5d) and (3.1.5e)) should be the answers. Then, one can obtain a unique answer for θ_2 from (3.1.5b) and (3.1.5g). Note that (3.1.5a) gives $\theta_1 = \pm\pi/2$ if $T_{23} = 0$.

Alternatively, one can obtain $\theta_3 = \theta_{31}$ and $\theta_3 = \theta_{32}$ from (3.1.5c). Substituting $\theta_3 = \theta_{31}$ into (3.1.5d) and (3.1.5e) yields a unique answer $\theta_1 = \theta_{11}$. If θ_{11} agrees well with (3.1.5a) (and (3.1.5f)), $\theta_3 = \theta_{31}$ and $\theta_1 = \theta_{11}$ are the answers for θ_3 and θ_1, respectively. If θ_{11} doesn't agree well with (3.1.5a), $\theta_3 = \theta_{32}$ and $\theta_1 = \theta_{12}$ (from (3.1.5d) and (3.1.5e)) should be the answers. Then, one can obtain a unique answer for θ_2 from (3.1.5b) and (3.1.5g). Note that (3.1.5c) gives $\theta_3 = \pm\pi/2$ if $T_{32} = 0$.

To avoid the singularity problems, one can present $[T]$ in terms of four *Euler parameters* (the so-called quaternion representation), three *Rodrigues parameters*, or three *modified Rodrigues parameters* [46,47,78,79]. According to Euler's principal

rotation theorem [46,47,77,79], the coordinate transformation matrix $[T]$ can be presented in terms of a unique *principal rotation angle* Φ w.r.t. a unit vector **n** (see Figure 3.4) as

$$[T]=\{n\}\{n\}^T(1-\cos\Phi)+[n]\sin\Phi+[I]\cos\Phi$$
$$=\begin{bmatrix} n_1^2(1-\cos\Phi)+\cos\Phi & n_1n_2(1-\cos\Phi)+n_3\sin\Phi & n_1n_3(1-\cos\Phi)-n_2\sin\Phi \\ n_1n_2(1-\cos\Phi)-n_3\sin\Phi & n_2^2(1-\cos\Phi)+\cos\Phi & n_2n_3(1-\cos\Phi)+n_1\sin\Phi \\ n_1n_3(1-\cos\Phi)+n_2\sin\Phi & n_2n_3(1-\cos\Phi)-n_1\sin\Phi & n_3^2(1-\cos\Phi)+\cos\Phi \end{bmatrix} \quad (3.1.6)$$

where $[I]$ is a 3×3 identity matrix and

$$\mathbf{n}=n_1\mathbf{i}_x+n_2\mathbf{i}_y+n_3\mathbf{i}_z=n_1\mathbf{i}_1+n_2\mathbf{i}_2+n_3\mathbf{i}_3,\ n_1^2+n_2^2+n_3^2=1$$
$$\{n\}\equiv\begin{Bmatrix} n_1 \\ n_2 \\ n_3 \end{Bmatrix},\ [n]\equiv\begin{bmatrix} 0 & n_3 & -n_2 \\ -n_3 & 0 & n_1 \\ n_2 & -n_1 & 0 \end{bmatrix} \quad (3.1.7)$$

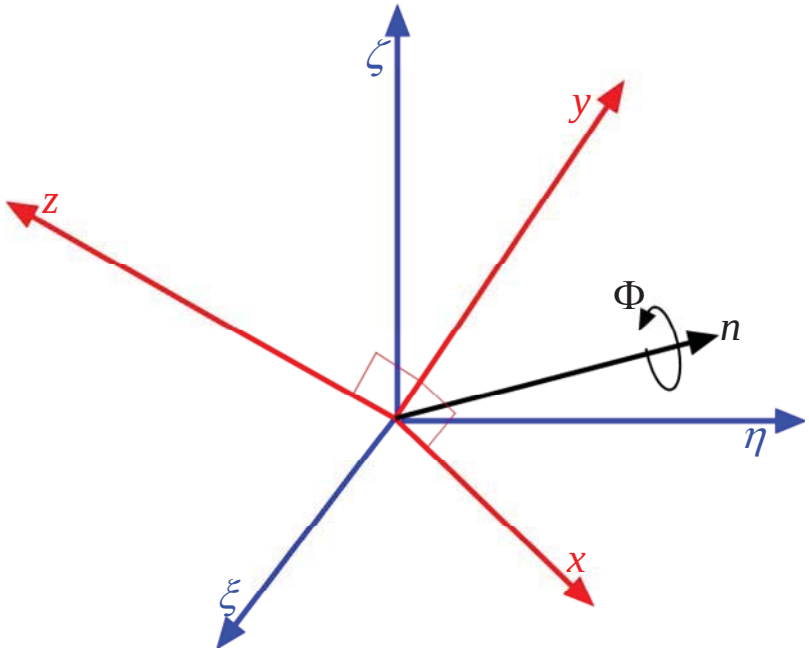


Figure 3.4 The principal rotation angle Φ and the rotation axis n.

It follows from (3.1.6) that

$$\cos\Phi = \frac{T_{11} + T_{22} + T_{33} - 1}{2} \tag{3.1.8}$$

$$n_i = \pm\sqrt{\frac{T_{ii} - \cos\Phi}{1 - \cos\Phi}},\ n_i n_j = \frac{T_{ij} + T_{ji}}{2(1 - \cos\Phi)} \tag{3.1.9}$$

$$\sin\Phi = \frac{n_1(T_{23} - T_{32}) + n_2(T_{31} - T_{13}) + n_3(T_{12} - T_{21})}{2} \tag{3.1.10}$$

One can use (3.1.8) to obtain $\cos\Phi$, (3.1.9) to obtain two sets of answers $\pm(n_1, n_2, n_3)$, and then use (3.1.10) to obtain $\sin\Phi$. The sign of $\pm(n_1, n_2, n_3)$ can be determined by assuming $\Phi \geq 0$ and choosing the smaller one from the two obtained values. Hence, a unique set of values for Φ and n_i $(i = 1,2,3)$ can be obtained. Because $\cos\Phi = 1$ corresponds to $\Phi = 2\pi n$ $(n = 0,1,...)$ and hence $[T] = [I]$. Hence, no singular points need to be treated in (3.1.8)-(3.1.10). Because of (3.1.7), the $[T]$ has three eigenvalues λ_i $(= 1$ and $\cos\Phi \pm j\sin\Phi,\ j \equiv \sqrt{-1})$, and the eigenvector corresponding to $\lambda = 1$ is actually the vector **n** [77,78]. This property can be used to double check the results obtained from (3.1.8)-(3.1.10).

Based on the obtained Φ and n_i, $[T]$ can be presented in terms of four *Euler parameters* $(\beta_0 \equiv \cos\frac{\Phi}{2},\ \beta_i \equiv n_i \sin\frac{\Phi}{2})$ without singular points, three *Rodrigues parameters* $(\gamma_i \equiv n_i \tan\frac{\Phi}{2})$ with a singular point at $\Phi = 180^\circ$, or three *modified Rodrigues parameters* $(\nu_i \equiv n_i \tan\frac{\Phi}{4})$ with a singular point at $\Phi = 360^\circ$ [78]. Alternatively, one can define

$$\mathbf{r} \equiv \mathbf{n}\Phi,\ \ r_i \equiv n_i\Phi\ \ (i = 1,2,3) \tag{3.1.11}$$

In the nonlinear optimization process shown later in Section 3.1.3 to estimate the intrinsic and extrinsic parameters, because $\Phi \geq 0$ is assumed, the direction of **n** may need to change discontinuously. When Φ changes from 0^+ to 0 then to 0^+ and n_i changes sign, r_i changes from $\varepsilon(\approx 0)$ to 0 and then to $-\varepsilon$. When Φ changes from $\Phi < 2k\pi$ to $\Phi > 2k\pi$ (k being a positive integer), both r_i and n_i change continuously. Hence, although n_i may be discontinuous at $\Phi = 0$, r_i, Φ, and $[T]$ are always continuous.

3.1.3 Estimation of Extrinsic and Intrinsic Parameters

The 3×3 homography matrix $[H]$ in (2.3.20) from (1.1.1) can be rewritten as

$$[H] \equiv [\{h_1\},\{h_2\},\{h_3\}] \equiv [\{\bar{h}_1\},\{\bar{h}_2\},\{\bar{h}_3\}]^T = [A][\{t_1\},\{t_2\},\{\tau\}] \tag{3.1.12}$$

where $\{h_k\}$ is the kth column and $\{\bar{h}_k\}^T$ is the kth row of $[H]$, and $\{t_k\}$ is the kth column of $[T]$. According to the maximum likelihood criterion, $[H]$ can be estimated by minimizing the following nonlinear objective function F_{oh}:

$$F_{oh} \equiv \sum_{i=1}^{I\geq 5} \left\| \{m_i\} - \{\hat{m}_i([A],\{M\}_i,[T],\{\tau\})\} \right\|^2, \{\hat{m}_i\} \equiv \begin{Bmatrix} \{\bar{h}_1\}^T\{M_i\}/\{\bar{h}_3\}^T\{M_i\} \\ \{\bar{h}_2\}^T\{M_i\}/\{\bar{h}_3\}^T\{M_i\} \\ 1 \end{Bmatrix} \tag{3.1.13}$$

where $\{m_i\} \equiv \{u_i, v_i, 1\}^T$. The image coordinates (u_i, v_i) of the ith image point can be determined from the image by using a corner detector (see Sec. 2.3), and $\{\hat{m}_i\}$ is obtained from (2.3.20) with $\{M\}_i \equiv (x_i, y_i)$ of the ith object point on the checker board being known. $[H]$ contains 3×3 unknowns (but only 8 degrees of freedom) that are functions of 11 nonlinearly coupled unknown intrinsic and extrinsic parameters. In order to obtain a

unique numerical solution of $[H]$ without nonlinear minimization processing of (3.1.13), we rewrite (2.3.20) for the *i*th point as

$$\begin{bmatrix} \{M_i\}^T & \{0\}_{1\times 3} & -u_i\{M_i\}^T \\ \{0\}_{1\times 3} & \{M_i\}^T & -v_i\{M_i\}^T \end{bmatrix} \{H\} = \{0\}, \{H\} \equiv \{\{\bar{h}_1\}^T, \{\bar{h}_2\}^T, \{\bar{h}_3\}^T\}^T \qquad (3.1.14)$$

Because there are *I* object points on the model plane, we have

$$[L]\{H\} = \{0\} \qquad (3.1.15)$$

where $[L]$ is a $2I \times 9$ matrix. The solution of $\{H\}$ corresponds to the column of the right singular matrix $[V]$ of $[L]$ ($=[U][D][V]^T$ by singular value decomposition) with the smallest singular value, and also the eigenvector of $[L]^T[L]$ with the smallest eigenvalue, as shown in Appendix B. Because $\{H\}$ contains 9 unknowns, at least 5 ($\leq I$) object points are needed for a unique solution. Because $[L]$ is often an ill conditioned matrix, better results can be obtained by performing a simple data normalization [80].

After the $[H]$ for each of J (≥ 3) images is obtained, it can be used to estimate the five linear intrinsic parameters (i.e. $\alpha, \beta, \gamma, u_0, v_0$) and six extrinsic parameters (three for translation and three for rotation). Because there are 11 unknown intrinsic and extrinsic parameters but only 9 components in $[H]$, two extra constraints on these 11 parameters are needed. Since $\{t_1\}$ and $\{t_2\}$ are unit vectors orthogonal to each other, $\{t_1\}^T\{t_2\} = 0$ and $|\{t_1\}| = |\{t_2\}|$. Hence, it follows from (3.1.12) that

$$\{h_1\}^T[B]\{h_2\} = 0 \qquad (3.1.16)$$

$$\{h_1\}^T[B]\{h_1\} - \{h_2\}^T[B]\{h_2\} = 0 \qquad (3.1.17)$$

where

$$[B] \equiv \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{13} & B_{23} & B_{33} \end{bmatrix} \equiv [A]^{-T}[A]^{-1} \tag{3.1.18}$$

If we define

$$\{h_i\}^T [B] \{h_j\} \equiv \{\tilde{h}_{ij}\}^T \{B\} \tag{3.1.19}$$

where

$$\begin{aligned} \{B\} &\equiv \{B_{11}, B_{12}, B_{22}, B_{13}, B_{23}, B_{33}\}^T \\ &= \left\{ \frac{1}{\alpha^2}, \frac{-\gamma}{\alpha^2\beta}, \frac{\alpha^2+\gamma^2}{\alpha^2\beta^2}, \frac{v_0\gamma - u_0\beta}{\alpha^2\beta}, \frac{u_0\beta\gamma - v_0(\alpha^2+\gamma^2)}{\alpha^2\beta^2}, \frac{(u_0\beta - v_0\gamma)^2 + v_0^2\alpha^2}{\alpha^2\beta^2} + 1 \right\}^T \end{aligned} \tag{3.1.20}$$

$$\{\tilde{h}_{ij}\} \equiv \{h_{i1}h_{j1}, h_{i1}h_{j2} + h_{i2}h_{j1}, h_{i2}h_{j2}, h_{i3}h_{j1} + h_{i1}h_{j3}, h_{i3}h_{j2} + h_{i2}h_{j3}, h_{i3}h_{j3}\}^T \tag{3.1.21}$$

(3.1.16) and (3.1.17) can be combined as

$$\begin{bmatrix} \{\tilde{h}_{12}\}^T \\ \{\tilde{h}_{11}\}^T - \{\tilde{h}_{22}\}^T \end{bmatrix} \{B\} = \{0\} \tag{3.1.22}$$

Because there are 6 unknowns in $\{B\}$, at least 3 ($\leq J$) images are required. Note that since these images need to have the same focal length in order to calibrate the camera, both the zoom and focusing rings should be fixed during capturing pictures of the checker board. For J (≥ 3) images,

$$[\tilde{H}]\{B\} = \{0\} \tag{3.1.23}$$

where $[\tilde{H}]$ is a $2J \times 6$ matrix. Because the right-hand side of (3.1.23) is $\{0\}$, solving (3.1.23) for $\{B\}$ yields a solution that is up to a scale factor λ as $[B] = \lambda[A]^{-T}[A]^{-1}$. After $[B]$ is obtained, the five intrinsic parameters in $[A]$ in (1.1.1) and the scale factor λ can be obtained by using the B_{ij} defined in (3.1.20) as

$$
\begin{aligned}
v_0 &= (B_{12}B_{13} - B_{11}B_{23})/(B_{11}B_{22} - B_{12}^2) \\
\lambda &= B_{33} - (B_{13}^2 + v_0(B_{12}B_{13} - B_{11}B_{23}))/B_{11} \\
\alpha &= \sqrt{\lambda / B_{11}} \\
\beta &= \sqrt{\lambda B_{11}/(B_{11}B_{22} - B_{12}^2)} \\
\gamma &= -B_{12}\alpha^2\beta/\lambda \\
u_0 &= \gamma v_0/\beta - B_{13}\alpha^2/\lambda
\end{aligned}
\tag{3.1.24}
$$

After $[A]$ is obtained, the extrinsic matrix $[[T]\{\tau\}]$ for each image can be obtained from (3.1.12) as

$$
\begin{aligned}
\{t_1\} &= [A]^{-1}\{h_1\}/\tilde{\lambda} \\
\{t_2\} &= [A]^{-1}\{h_2\}/\tilde{\lambda} \\
\{t_3\} &= \{t_1\} \times \{t_2\} \\
\{\tau\} &= [A]^{-1}\{h_3\}/\tilde{\lambda}
\end{aligned}
\tag{3.1.25}
$$

where

$$
\tilde{\lambda} \equiv \left\| [A]^{-1}\{h_1\} \right\| = \left\| [A]^{-1}\{h_2\} \right\|, [T] = [\{t_1\}, \{t_2\}, \{t_3\}] \tag{3.1.26}
$$

To insure $[T]$ is a unitary matrix with $\left|[T]\right| = 1$, one needs to renormalize it or replace $[T]$ ($=[U][D][V]^T$ from singular-value decomposition) with $[U][V]^T$ [19].

The above analytical solution of intrinsic and extrinsic parameters is based on the $\{H\}$ obtained from (3.1.15), which is to minimize a physically meaningless algebraic distance. For a more physically meaningful and accurate solution, one can refine it through the following nonlinear optimization process based on the maximum likelihood inference. Because there are *J* images of the model plane and *I* object points on each model plane, the solution can be refined by nonlinear minimization of the following objective function F_o:

$$F_0 \equiv \sum_{i=1}^{I}\sum_{j=1}^{J}\left\|\{m_{ij}\}-\{\hat{m}_{ij}([A],\{M\}_i,[T]_j,\{\tau\}_j)\}\right\|^2 \qquad (3.1.27)$$

There are $5+6J$ parameters in (3.1.27) to be determined by nonlinear optimization. However, because the lens distortion effect is not removed from the image coordinates, this process cannot provide accurate solutions and it often converges slowly.

3.1.4 Nonlinear Regression

In order to find the minimum (or maximum) value of the nonlinear objective function F_0, one needs to use nonlinear regression to approximate the $5+6J$ independent parameters. There are three most common algorithms for nonlinear regression: Gauss-Newton algorithm, gradient descent, and Levenberg-Marquardt algorithm.

The Gauss-Newton algorithm [81] is a modification of Newton's method, and hence its rate of convergence is at most quadratic. Although second derivatives are not required, it can only be used to minimize a sum-of-squares objective function. In the Gauss-Newton algorithm, one can add a small increment Δ_i to each variable to renew the sum of squares and locally approximate the objective function $F_0(\{p\})=\frac{1}{2}\sum_{k=1}^{K} r_k^2(\{p\})$ through a second-order Taylor series expansion as

$$F_0(\{p\}+\{\Delta\}) \approx F_0(\{p\})+\left\{\frac{\partial F_0}{\partial p_i}\right\}^T\{\Delta\}+\frac{1}{2}\{\Delta\}^T\left[\frac{\partial^2 F_0}{\partial p_i \partial p_j}\right]\{\Delta\} \qquad (3.1.28)$$

where r_k^2 are the distance squares, $\{p\}$ is the parameter vector, and $\{\Delta\}$ is the increment vector.

For a sum-of-squares objective function, we have

$$\left\{\frac{\partial F_0}{\partial p_i}\right\} \equiv \{\nabla F_0\} = \left\{\frac{\partial r_k}{\partial p_i}\frac{\partial F_0}{\partial r_k}\right\} = [J_r]^T\{r\},\ \left[\frac{\partial^2 F_0}{\partial p_i \partial p_j}\right] = [J_r]^T[J_r]$$
$$\{r\} \equiv \begin{Bmatrix} r_1 \\ \vdots \\ r_K \end{Bmatrix}, [J_r] \equiv \left[\frac{\partial r_i}{\partial p_j}\right] \tag{3.1.29a}$$

Hence, (3.1.28) can be rewritten as

$$F_0(\{p\}+\{\Delta\}) \approx F_0(\{p\}) + \{r\}^T[J_r]\{\Delta\} + \frac{1}{2}\{\Delta\}^T[J_r]^T[J_r]\{\Delta\} \tag{3.1.29b}$$

Because $\{\partial F_0(\{p\}+\{\Delta\})/\partial \Delta_i\} = \{0\}$ at a minimum (or maximum) point of F_0, taking derivatives on (3.1.29b) yields

$$[J_r]^T\{r\} + [J_r]^T[J_r]\{\Delta\} = \{0\} \tag{3.1.30}$$

The $\{\Delta\}$ needed for the next iteration can be obtained from (3.1.30).

The gradient descent method [82] (also known as the steepest descent method) updates parameter values in the direction opposite to the gradient of the object function as

$$\{p_{n+1}\} = \{p_n\} - \gamma_n\{\nabla F_0\} \tag{3.1.31}$$

where $\{p_n\}$ is the *n*th step parameter vector and γ_n is the *n*th step size (γ can be changed at every step).

Because the Gauss-Newton method observes changes of parameter values, if the initial guess of solution is close to a local minimum, it converges faster than the gradient descent method. However, if the initial guess is poor, it does not guarantee convergence and may not converge to the solution and it is computationally expensive. On the other hand, because the gradient descent method surveys the gradient of the object function, it provides a steady and convergent progress toward the solution even if the initial guess is far from the solution. However, its convergence speed is low except the first few steps

and it may get stuck at a local minimum. Hence, it is only recommended for problems with a large number of parameters.

The Levenberg-Marquardt algorithm [83,84] (also named as the damped least-squares method) is a hybrid of the Gauss-Newton algorithm and the gradient descent method. It adaptively varies the parameter update process by using the gradient descent method when the initial guesses is far from a local minimum and using the Gauss-Newton method when the initial guesses is closed to a local minimum. Hence, it also can be considered as a trust-region method, where the trust region is used to represent the subset of the region of the objective function. It often uses a quadratic model function to approximate the optimization solution. If the approximated model is adequate, the region is expanded; otherwise, it is contacted [85].

Levenberg [83] added a damping factor λ to change (3.1.30) into

$$([J_r]^T[J_r] + \lambda[I])\{\Delta\} = -[J_r]^T\{r\} \tag{3.1.32}$$

where λ needs to be adjusted at each iteration. If the reduction of the object function value is fast, a smaller value can be chosen to make a step more like the Gauss-Newton method. If the reduction is small, λ will be increased to make the step more like the gradient descent method. Marquardt [84] improved (3.1.32) by scaling each component of the gradient according to the curvature to avoid slow convergence in the direction of small gradient as

$$([J_r]^T[J_r] + \lambda\ \mathrm{diag}([J_r]^T[J_r]))\{\Delta\} = -[J_r]^T\{r\} \tag{3.1.33}$$

Because the Levenberg-Marquardt algorithm can start with an initial guess far from the final minimum, it is more robust than the Gauss-Newton method, but its convergence

speed is slower. Nevertheless, because it can only find a local minimum but not the global one, a good initial guess is still needed.

Because there are $5+6J$ ($J \geq 3$) parameters need to be optimized in (3.1.27), we recommend to use the Levenberg-Marquardt algorithm. For a camera, its 5 linear intrinsic parameters are unchanged in the J images, and they do not change too much from the initial estimation obtained from (3.1.24) during the nonlinear regression. Hence, one can reduce the number of parameters in (3.1.27) to 11 by refining each image respectively to increase the robustness of nonlinear regression using

$$F_{0j} \equiv \sum_{i=1}^{I} \left\| \{m_{ij}\} - \{\hat{m}_{ij}([A], \{M\}_i, [T]_j, \{\tau\}_j)\} \right\|^2 \tag{3.1.34}$$

To guarantee correct convergence, it is better to check the condition number of $[J_r]^T[J_r]$ for each image. If it is ill-conditioned (i.e., condition number $> 10^6$), the image is not valid for use.

3.1.5 Lens Distortion Model

After the first-time estimation of intrinsic and extrinsic parameters using (3.1.24) and (3.1.25) or even (3.1.27), lens distortion needs to be estimated and accounted for in order to obtain a more accurate set of parameter values. A classical lens distortion model was proposed by Brown and Conrady [86,87]. The Brown-Conrady model (or "plumb bob" model) adds a tangential component to the radial distortion, and the radial distortion is expressed as [88,89]

$$\delta_{rr} = \sum_{n=1}^{\infty} K_n r^{2n+1} \tag{3.1.35}$$

where r is the radial distance from the principal point on the sensor plane, and K_n is the nth-order radial distortion coefficient. $K_1 > 0$ corresponds to barrel distortion, and $K_1 < 0$ corresponds to pincushion distortion. The radial distortion δ_{rr} is along the radial direction of the polar coordinate system $r\varphi$, and can be decomposed into Cartesian coordinates as

$$\begin{aligned} &U_r = \hat{u}\sum_{n=1}^{\infty} K_n r^{2n}, \quad V_r = \hat{v}\sum_{n=1}^{\infty} K_n r^{2n} \\ &\hat{u} \equiv (\tilde{u} - u_0)\Delta_1 = r\cos\varphi, \ \hat{v} \equiv (\tilde{v} - v_0)\Delta_2 = r\sin\varphi, \ r^2 \equiv \hat{u}^2 + \hat{v}^2 \end{aligned} \tag{3.1.36}$$

where $(\tilde{u}, \tilde{v})$ are the measured, distorted pixel image coordinates, and (u_0, v_0) is the distortion center. Moreover, Δ_1 and Δ_2 are the horizontal and vertical distances between two adjacent pixels, $\Delta_1 = s_1 / \alpha$, and $\Delta_2 = s_1 / \beta$.

The decentering distortion consists of a radial component δ_{rd} and a tangential component δ_{td} given by [87,89]

$$\begin{aligned} \delta_{rd} &= 3\sin(\varphi - \varphi_0)\sum_{n=1}^{\infty} J_n r^{2n} \\ \delta_{td} &= \cos(\varphi - \varphi_0)\sum_{n=1}^{\infty} J_n r^{2n} \end{aligned} \tag{3.1.37}$$

where J_n is the coefficient of the nth-order decentering distortion, and φ_0 is the angle between the u axis and the axis of maximum tangential distortion due to decentering distortion. In terms of Cartesian coordinates, the decentering distortion consists of U_d and V_d given by

$$\begin{Bmatrix} U_d \\ V_d \end{Bmatrix} = \begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix} \begin{Bmatrix} \delta_{rd} \\ \delta_{td} \end{Bmatrix} \tag{3.1.38}$$

Since $\cos\varphi = \hat{u}/r$ and $\sin\varphi = \hat{v}/r$, if $P_1 \equiv -J_1 \sin\phi_0$, $P_2 \equiv J_1 \cos\phi_0$, and $P_n \equiv J_{n-1}/J_1$ ($n \geq 3$), (3.1.38) can be rewritten as

$$\begin{aligned} U_d &= \left(P_1(r^2 + 2\hat{u}^2) + 2P_2\hat{u}\hat{v}\right)\left(1 + \sum_{n=3}^{\infty} P_n r^{2(n-2)}\right) \\ V_d &= \left(P_2(r^2 + 2\hat{v}^2) + 2P_1\hat{u}\hat{v}\right)\left(1 + \sum_{n=3}^{\infty} P_n r^{2(n-2)}\right) \end{aligned} \tag{3.1.39}$$

Thin prism distortion consists of a radial component δ_{rp} and a tangential component δ_{tp} given by [87,88]

$$\begin{aligned} \delta_{rp} &= \sin(\varphi - \varphi_1)\sum_{n=1}^{\infty} I_n r^{2n} \\ \delta_{tp} &= \cos(\varphi - \varphi_1)\sum_{n=1}^{\infty} I_n r^{2n} \end{aligned} \tag{3.1.40}$$

where I_n is the coefficient of the nth-order thin prism distortion, and φ_1 is the angle between the u axis and the axis of maximum tangential distortion caused by thin prism distortion. In terms of Cartesian coordinates and definitions $S_1 \equiv -I_1 \sin\phi_1$, $S_2 \equiv I_1 \cos\phi_1$, and $S_n \equiv I_{n-1}/I_1$ ($n \geq 3$), the thin prism distortion consists of U_p and V_p given by

$$\begin{aligned} U_p &= S_1 r^2\left(1 + \sum_{n=3}^{\infty} S_n r^{2(n-2)}\right) \\ V_p &= S_2 r^2\left(1 + \sum_{n=3}^{\infty} S_n r^{2(n-2)}\right) \end{aligned} \tag{3.1.41}$$

Although the decentering and thin-prism distortions have similar terms, their directions of maximum tangential distortion are different. Following (3.1.36), (3.1.39), and (3.1.41), a general lens distortion model can be expressed as

$$
\begin{aligned}
& u = \tilde{u} + U/\Delta_1, \quad v = \tilde{v} + V/\Delta_2 \\
& U \equiv \hat{u}\sum_{n=1}^{\infty} K_n r^{2n} + (P_1(r^2 + 2\hat{u}^2) + 2P_2\hat{u}\hat{v})(1 + \sum_{n=3}^{\infty} P_n r^{2(n-2)}) + S_1 r^2 (1 + \sum_{n=3}^{\infty} S_n r^{2(n-2)}) \\
& V \equiv \hat{v}\sum_{n=1}^{\infty} K_n r^{2n} + (P_2(r^2 + 2\hat{v}^2) + 2P_1\hat{u}\hat{v})(1 + \sum_{n=3}^{\infty} P_n r^{2(n-2)}) + S_2 r^2 (1 + \sum_{n=3}^{\infty} S_n r^{2(n-2)}) \\
& E_{rror} \equiv \sum_{j=1}^{J}\sum_{i=1}^{I}\left[(u\Delta_1 - \tilde{u}\Delta_1 - U)^2 + (v\Delta_2 - \tilde{v}\Delta_2 - V)^2 \right]
\end{aligned}
\tag{3.1.42}
$$

The measured, distorted pixel coordinates $(\tilde{u}, \tilde{v})$ can be obtained from the 2D image by corner detection. In our model, 5 intrinsic parameters and 6 lens distortion parameters K_1, K_2, P_1, P_2, S_1, and S_2 are used. However, most of today's cameras do not have imperfection in centering (i.e., $P_1 = P_2 = 0$), thin-prism distortion is negligible ($S_1 = S_2 = 0$), and rectangular pixels ($\gamma = 0$) or even square pixels ($\alpha = \beta$) are used. Moreover, when only a few (less than 4) images are used for calibration, it is well known to be very difficult to have reliable estimation of the principal point (u_0, v_0). (However, numerical simulations and experimental results by many researchers show that the estimation error in (u_0, v_0) has only minor effects in 3D reconstruction.) Hence, one can assume that the principal point (u_0, v_0) coincides with the fiducial point (i.e., the image center). Furthermore, it was pointed out in the literature that a distortion model with more parameters may not provide better corrections (negligible when compared with sensor quantization) and sometimes it causes numerical instability [32,90]. Hence, for the first estimation of intrinsic and lens distortion parameters, one can use these assumptions to reduce the camera model to have an easy and reasonable estimation. After the intrinsic and extrinsic parameters are estimated and refined using the methods presented in Section 3.1.3, one can use (1.1.1) with the known object coordinates of the *I* object points on the

checker board to estimate the undistorted image coordinates (u,v) of the $I \times J$ image points. Then, the unknown parameters in this backward lens distortion model can be determined by linear least squares fitting to minimize the error function E_{rror} defined in (3.1.42).

Unfortunately, this backward model is slow in calculating (u,v). In order to increase the efficiency for general everyday applications, researchers try to find a forward distortion model from the backward one [91-94], but it still a high-order formula with polynomials in both numerator and denominator. So that some researchers try to develop a forward lens distortion model that can directly provide the measured, distorted pixel image coordinates from the ideal, undistorted pixel image coordinates in the following form:

$$
\begin{aligned}
&\tilde{u} = u + \hat{U}/\Delta_1, \quad \tilde{v} = v + \hat{V}/\Delta_2, \quad \hat{u} \equiv (u-u_0)\Delta_1, \hat{v} \equiv (v-v_0)\Delta_2, \hat{r}^2 \equiv \hat{u}^2 + \hat{v}^2 \\
&\hat{U} \equiv \hat{u}\sum_{n=1}^{\infty} \hat{K}_n \hat{r}^{2n} + (\hat{P}_1(\hat{r}^2 + 2\hat{u}^2) + 2\hat{P}_2\hat{u}\hat{v})(1 + \sum_{n=3}^{\infty} \hat{P}_n \hat{r}^{2(n-2)}) + \hat{S}_1\hat{r}^2(1 + \sum_{n=3}^{\infty} \hat{S}_n \hat{r}^{2(n-2)}) \\
&\hat{V} \equiv \hat{v}\sum_{n=1}^{\infty} \hat{K}_n \hat{r}^{2n} + (\hat{P}_2(\hat{r}^2 + 2\hat{v}^2) + 2\hat{P}_1\hat{u}\hat{v})(1 + \sum_{n=3}^{\infty} \hat{P}_n \hat{r}^{2(n-2)}) + \hat{S}_2\hat{r}^2(1 + \sum_{n=3}^{\infty} \hat{S}_n \hat{r}^{2(n-2)}) \\
&\hat{E}_{rror} \equiv \sum_{j=1}^{J}\sum_{i=1}^{I}\left[(\tilde{u}\Delta_1 - u\Delta_1 - \hat{U})^2 + (\tilde{v}\Delta_2 - v\Delta_2 - \hat{V})^2\right]
\end{aligned}
\tag{3.1.43}
$$

It follows from (3.1.42) and (3.1.43) that $\hat{U} = -U$ and $\hat{V} = -V$, but the coefficients $\hat{K}_i$, $\hat{P}_i$ and $\hat{S}_i$ are different from K_i, P_i and S_i. While $\hat{K}_1 > 0$ corresponds to pincushion distortion, and $\hat{K}_1 < 0$ corresponds to barrel distortion.

Since (u,v) of the $I \times J$ image points are unknown and can only be estimated, development of an optical distortion model is an iterative process. One can obtain a better estimation but it can never be exact, especially when noise is significant. With the initial

estimations of $[A]$ from (3.1.24), $[[T]\{\tau\}]$ from (3.1.25), and K_i, P_i and S_i from linear least squares fitting of (3.1.42), estimations of these parameters can be simultaneously refined by nonlinear minimization of the following objective function F_0:

$$F_0 \equiv \sum_{i=1}^{I}\sum_{j=1}^{J}\left\|\{m_{ij}\}-\{\hat{m}_{ij}([A],\{M\}_i,[T]_j,\{\tau\}_j,\hat{K}_1,\hat{K}_2,\hat{P}_1,\hat{P}_2,\hat{S}_1,\hat{S}_2)\}\right\|^2 \qquad (3.1.44)$$

where $\{m_{ij}\}$ contains the measured, distorted image coordinates $(\tilde{u},\tilde{v})$.

3.1.6 Numerical Simulation of Camera Calibration

To verify the accuracy of the proposed camera calibration algorithm, we consider a camera with the following parameters [19]:

$$\begin{aligned}&\alpha = 1250,\ \beta = 900,\ \gamma = 1.09083\ (\text{i.e.}, \cos^{-1}(\gamma/\alpha) = 89.95^\circ)\\ &u_0 = 255,\ v_0 = 255\end{aligned} \qquad (3.1.45a)$$

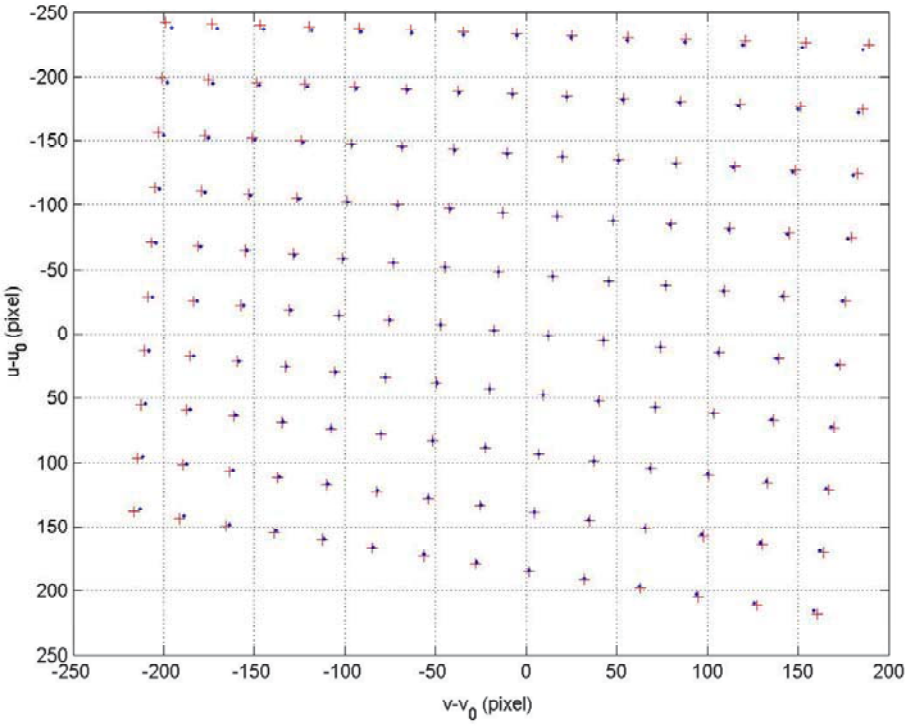
The sensor plane has 512×512 pixels. A checker board with 10×14 corner points uniformly distributed over an area of $180mm\times250mm$ is used as the model plane. The images of the following 3 positions of the model plane are used for simulations:

$$\{\tau\} \equiv \begin{Bmatrix}\xi_0\\ \eta_0\\ \zeta_0\end{Bmatrix} = \begin{Bmatrix}-90\\ 105\\ 500\end{Bmatrix}, \begin{Bmatrix}-90\\ 105\\ 510\end{Bmatrix}, \begin{Bmatrix}-105\\ 105\\ 525\end{Bmatrix}; \quad \begin{Bmatrix}\theta_1\\ \theta_2\\ \theta_3\end{Bmatrix} = \begin{Bmatrix}5^\circ\\ 160^\circ\\ 10^\circ\end{Bmatrix}, \begin{Bmatrix}5^\circ\\ 185^\circ\\ 5^\circ\end{Bmatrix}, \begin{Bmatrix}45^\circ\\ 200^\circ\\ 30^\circ\end{Bmatrix} \qquad (3.1.45b)$$

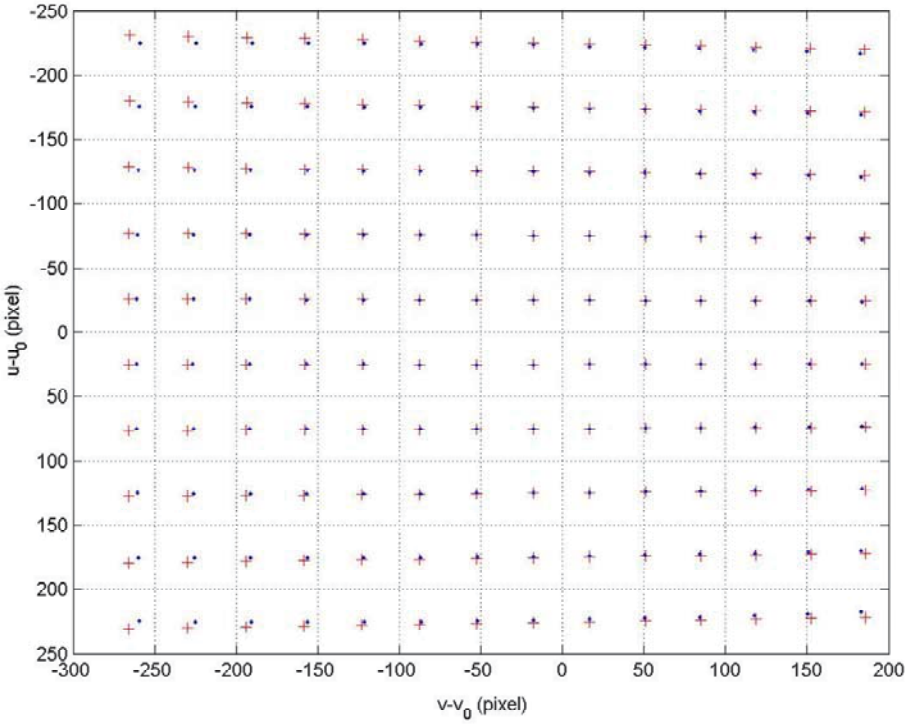
where the values of $\{\tau\}$ are in *mm*.

By using the above information, one can get the ideal, undistorted image coordinates (u,v) of all corner points. Because the image distance s_1 shown in Figure 2.1 is unknown here, one can set $s_1 = 1$ and hence $\Delta_1 = 1/\alpha$ and $\Delta_2 = 1/\beta$. The forward lens

distortion model is assumed to have $\hat{K}_1 = -0.23$, $\hat{K}_2 = 0.2$ and the other four parameters being zero, which corresponds to a barrel distortion. By using this forward lens distortion model, a group of distorted image coordinates $(\tilde{u}, \tilde{v})$ can be obtained using (1.1.1) and (3.1.43). Moreover, Gaussian noise with a zero mean and a standard deviation of σ pixels is added to the pixel image coordinates $(\tilde{u}, \tilde{v})$. Without added noise (i.e., $\sigma = 0$), Figure 3.5 shows the three images on the sensor plane. Applying the nonlinear optimization process shown in (3.1.44), intrinsic and extrinsic parameters and a forward lens distortion model similar to the assumed one can be found. Using the so-obtained intrinsic and extrinsic parameters and the forward lens distortion model, one can obtain a backward distortion model from (3.1.42) by linear least-squares fitting. Without added noise, Table 3.1 compares the obtained forward and backward lens distortion models. It shows that the estimated forward model agrees well with the assumed one. Table 3.2 shows that the estimated intrinsic parameter values are exactly the same as the assumed ones. Table 3.3 shows that the estimated translation vectors of the three locations of the model plane are exactly the same as the assumed ones. Tables 3.4-3.6 show that the estimated [T] of the three orientations of the model plane agree very well with the assumed ones. With the first-time optimization, the value of the objective function is 1.3372×10^{-5}, and it converges to 2.6353×10^{-7} at the end of optimization.


u-u0 (pixel)
v-v0 (pixel)
-250
-200
-150
-100
-50
0
50
100
150
200
250

a


u-u0 (pixel)
v-v0 (pixel)
-300
-250
-200
-150
-100
-50
0
50
100
150
200
250

b

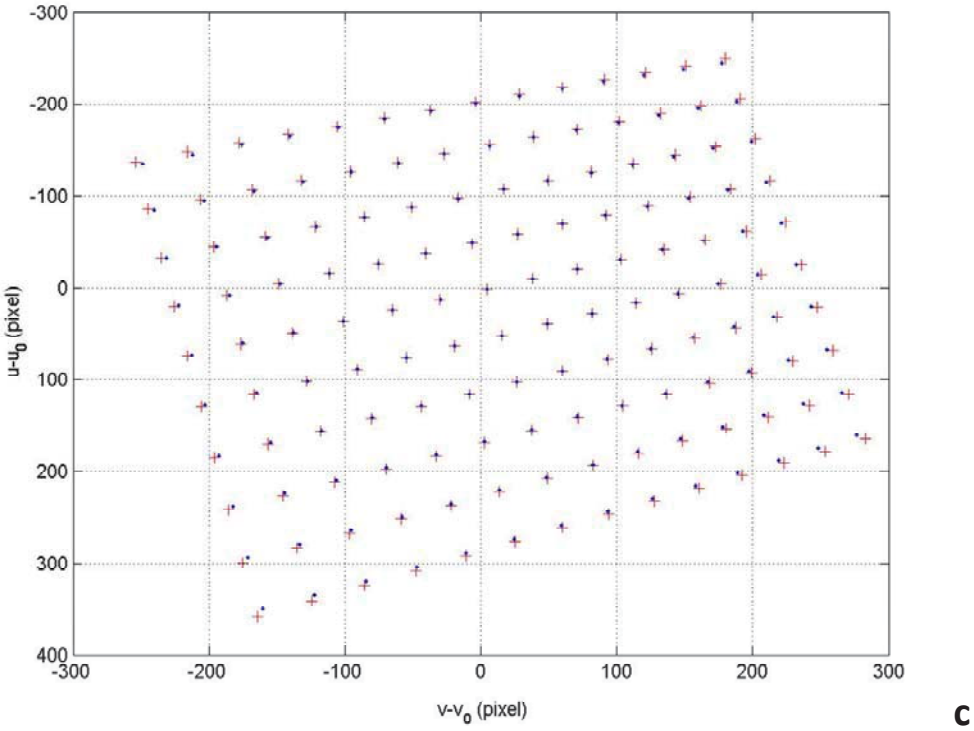


Figure 3.5 The 140 corner points on each of three images, where the blue points are distorted into the red crosses by the lens distortion.

Table 3.1 The estimated forward and backward lens distortion models.

Forward model	Assumed	Estimated		Backward model	Estimated
$\hat{K}_1$	-0.23	-0.23		K_1	0.23124
$\hat{K}_2$	0.2	0.2		K_2	-0.077803
$\hat{P}_1$	0.0	-4.7909×10^{-7}		P_1	-5.3734×10^{-7}
$\hat{P}_2$	0.0	-1.9462×10^{-8}		P_2	-5.292×10^{-7}
$\hat{S}_1$	0.0	5.41×10^{-7}		S_1	6.6182×10^{-7}
$\hat{S}_2$	0.0	-4.9183×10^{-7}		S_2	9.5093×10^{-7}

Table 3.2 The assumed and estimated intrinsic parameters.

Parameters (pixel)	α	β	γ	u_0	v_0
Assumed	1250	900	1.0908	255	255
Estimated	1250	900	1.0908	255	255

Table 3.3 The assumed and estimated translation vectors.

Translation vectors (mm)	$\{\tau\}_1^T$	$\{\tau\}_2^T$	$\{\tau\}_3^T$
Assumed	{-90 105 500}	{-90 105 510}	{-105 105 525}
Estimated	{-90 105 500}	{-90 105 510}	{-105 105 525}

Table 3.4 The assumed and estimated [*T*] of the first image.

Rotational matrix	$[T]_1$
Assumed	$\begin{bmatrix} 0.99528 & -0.092332 & 0.029809 \\ -0.076723 & -0.93703 & -0.34072 \\ 0.059391 & 0.33682 & -0.93969 \end{bmatrix}$
Estimated	$\begin{bmatrix} 0.99528 & -0.092332 & 0.029809 \\ -0.076723 & -0.93703 & -0.34072 \\ 0.059391 & 0.33682 & -0.93969 \end{bmatrix}$

Table 3.5 The assumed and estimated [*T*] of the second image.

Rotational matrix	$[T]_2$
Assumed	$\begin{bmatrix} 0.99997 & -3.3039\times10^{-4} & -7.5961\times10^{-3} \\ 3.3039\times10^{-4} & -0.99622 & 0.086824 \\ -7.5961\times10^{-3} & -0.086824 & -0.99619 \end{bmatrix}$
Estimated	$\begin{bmatrix} 0.99997 & -3.3034\times10^{-4} & -7.5967\times10^{-3} \\ 3.3049\times10^{-4} & -0.99622 & 0.086824 \\ -7.5967\times10^{-3} & -0.086824 & -0.99619 \end{bmatrix}$

Table 3.6 The assumed and estimated [*T*] of the third image.

Rotational matrix	$[T]_3$
Assumed	$\begin{bmatrix} 0.9446 & 0.22189 & -0.24184 \\ 0.28014 & -0.929 & 0.24184 \\ -0.17101 & -0.2962 & -0.93969 \end{bmatrix}$
Estimated	$\begin{bmatrix} 0.9446 & 0.22189 & -0.24184 \\ 0.28014 & -0.929 & 0.24184 \\ -0.17101 & -0.2962 & -0.93969 \end{bmatrix}$

Using (3.1.5a)-(3.1.5g) and the three estimated $[T]$s shown in Tables 3.4-3.6, the three Euler angles are calculated to be

$$\begin{Bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{Bmatrix} = \begin{Bmatrix} 5.0000^\circ \\ 159.9996^\circ \\ 10.0001^\circ \end{Bmatrix}, \begin{Bmatrix} 5.0004^\circ \\ 185.0031^\circ \\ 5.0004^\circ \end{Bmatrix}, \begin{Bmatrix} 45^\circ \\ 200.0004^\circ \\ 29.9998^\circ \end{Bmatrix}$$

They agree well with the assumed ones. Figure 3.6 shows the three reverse-calculated locations of the simulated camera.

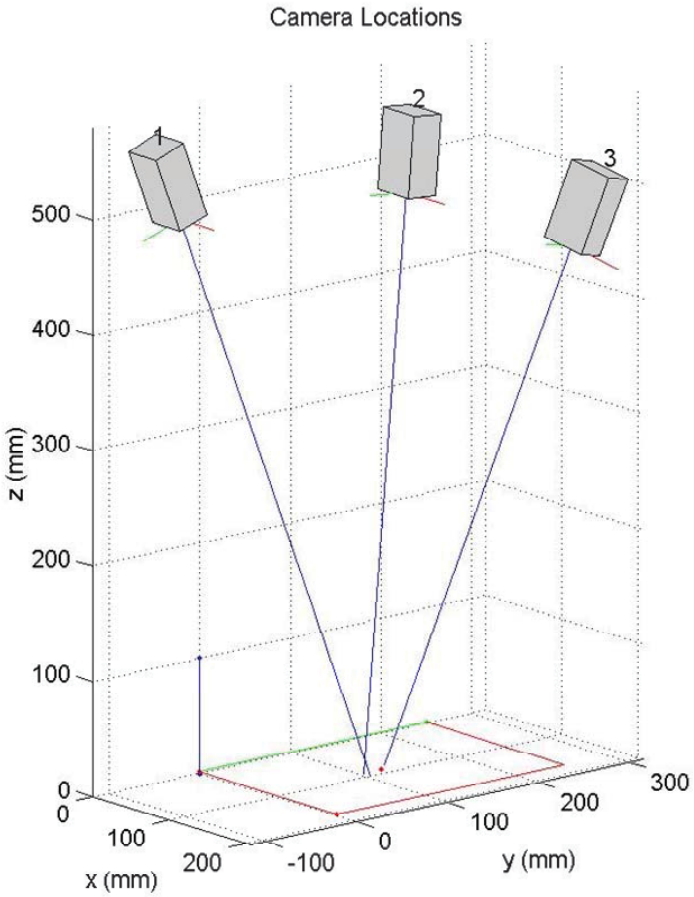


Figure 3.6 The three locations of the simulated camera ($\sigma = 0$).

Next we examine the accuracy of the camera calibration algorithm when random noise is added to the distorted image coordinates $(\tilde{u}, \tilde{v})$. This noise can be due to inaccurate extraction of image coordinates of corner points using a corner detector. Because the noise can be considered as a part of lens distortion, we use the same assumed intrinsic and extrinsic parameters and the same random noise with $\sigma = 0.1$ but update the forward lens distortion model at the beginning of each optimization process using

(3.1.44). Table 3.7 shows that, after five runs of the optimization process, the objective function converges to a fixed value, which is non-zero due to the added noise.

Table 3.7 Minimization of the objective function F_0 during five runs of optimization process.

Iteration times	***1***	***2***	***3***	***4***	***5***
$F_{0\min}$	7.3706	7.3727	7.3706	7.3722	7.3706

Tables 3.8 and 3.9 show the convergences of the forward and backward lens distortion models. Since noise is added, the converged forward distortion model is different from the assumed one. Table 3.8 shows that the first iteration already converges to a solution very close to the final one, and the absolute values of the last four parameters keep reducing during iterations. Table 3.9 shows that the backward model has similar convergence behavior. Tables 3.8 and 3.9 show that $\hat{K}_1 \approx -K_1$ and $\hat{K}_2 \approx -K_2$.

Table 3.8 The estimated forward lens distortion model.

Iteration times	$\hat{K}_1$	$\hat{K}_2$	$\hat{P}_1$	$\hat{P}_2$	$\hat{S}_1$	$\hat{S}_2$
0 **(Assumed)**	-0.23	0.2	0.0	0.0	0.0	0.0
1	-0.23144	0.21168	7.0611×10^{-5}	5.5258×10^{-5}	-2.4598×10^{-4}	-1.406×10^{-5}
2	-0.2315	0.2119	1.0585×10^{-4}	7.1062×10^{-5}	-2.8241×10^{-4}	-3.3442×10^{-5}
3	-0.23144	0.21167	9.3499×10^{-5}	4.7655×10^{-5}	-2.8445×10^{-4}	1.7045×10^{-7}
4	-0.23131	0.21021	1.1112×10^{-4}	7.1805×10^{-5}	-2.9355×10^{-4}	-4.0137×10^{-5}
5	-0.23145	0.21174	9.325×10^{-5}	4.9422×10^{-5}	-2.8716×10^{-4}	-4.1325×10^{-6}

Table 3.9 The estimated backward lens distortion model.

Iteration times	K_1	K_2	P_1	P_2	S_1	S_2
0 **(Assumed)**	–	–	–	–	–	–
1	0.2331	-0.095551	-5.3067×10^{-5}	-3.3349×10^{-5}	2.2464×10^{-4}	-5.2765×10^{-5}
2	0.23315	-0.095648	-8.6969×10^{-5}	-5.2255×10^{-5}	2.5727×10^{-4}	-2.7337×10^{-5}
3	0.2331	-0.095555	-7.7251×10^{-5}	-2.5294×10^{-5}	2.6507×10^{-4}	-6.7687×10^{-5}
4	0.23299	-0.094383	-9.245×10^{-5}	-5.2522×10^{-5}	2.6918×10^{-4}	-2.2685×10^{-5}
5	0.23311	-0.095626	-7.7347×10^{-5}	-2.7027×10^{-5}	2.6849×10^{-4}	-6.3627×10^{-5}

Table 3.10 shows the five estimated intrinsic parameters from the five runs of the optimization process. They approach constants after the third run. Except γ, the converged values of the other four parameters are very close to the assumed ones.

Table 3.10 The estimated intrinsic parameters from five runs of optimization.

Iteration times (pixel)	α	β	γ	u_0	v_0
***0* (Assumed)**	1250	900	1.0908	255	255
1	1251.2	900.94	0.95257	254.91	254.81
2	1251.9	901.44	0.93197	254.89	254.83
3	1251.3	900.96	0.95159	255.01	254.78
4	1251.8	901.38	0.93489	254.93	254.83
5	1251.2	900.92	0.95314	255.03	254.79

Table 3.11 shows the estimated translation vectors of the three simulated camera locations from the five runs of the optimization process. The errors decrease significantly from the first to the second run. Tables 3.12-3.14 show the convergence of the rotational matrix of each simulated camera location. After five runs, the results are almost the same as the assumed ones. The results shown in Tables 3.7- 3.14 indicate that the proposed camera calibration algorithm is accurate, fast, and robust.

Table 3.11 The estimated translation vectors of the three camera locations from five runs of optimization.

Iteration times (mm)	$\{\tau\}_1^T$	$\{\tau\}_2^T$	$\{\tau\}_3^T$
***0* (Assumed)**	{-90 105 500}	{-90 105 510}	{-105 105 525}
1	{-89.956 105.11 500.51}	{-89.958 105.11 510.47}	{-104.96 105.09 525.47}
2	{-89.943 105.09 500.78}	{-89.945 105.1 510.75}	{-104.94 105.07 525.74}
3	{-89.993 105.12 500.51}	{-89.996 105.12 510.47}	{-104.99 105.1 525.47}
4	{-89.961 105.09 500.75}	{-89.963 105.1 510.71}	{-104.96 105.07 525.7}
5	{-90.001 105.12 500.49}	{-90.004 105.12 510.45}	{-105 105.1 525.45}

Table 3.12 The estimated rotation matrix of the first camera location from five runs of optimization.

Iteration times	$[T]_1$
***0* (Assumed)**	$\begin{bmatrix} 0.99528 & -0.092332 & 0.029809 \\ -0.076723 & -0.93703 & -0.34072 \\ 0.059391 & 0.33682 & -0.93969 \end{bmatrix}$
1	$\begin{bmatrix} 0.99528 & -0.092389 & 0.029574 \\ -0.076843 & -0.93694 & -0.34093 \\ 0.059207 & 0.33705 & -0.93962 \end{bmatrix}$
2	$\begin{bmatrix} 0.99528 & -0.092396 & 0.0296 \\ -0.076836 & -0.93692 & -0.341 \\ 0.05924 & 0.33712 & -0.9396 \end{bmatrix}$
⋮	⋮
5	$\begin{bmatrix} 0.99528 & -0.092419 & 0.029656 \\ -0.076842 & -0.93694 & -0.34095 \\ 0.059296 & 0.33706 & -0.93961 \end{bmatrix}$

Table 3.13 The estimated rotation matrix of the second camera location from five runs of optimization.

Iteration times	$[T]_2$
0 **(Assumed)**	$\begin{bmatrix} 0.99997 & -3.3039\times10^{-4} & -7.5961\times10^{-3} \\ 3.3039\times10^{-4} & -0.99622 & 0.086824 \\ -7.5961\times10^{-3} & -0.086824 & -0.99619 \end{bmatrix}$
1	$\begin{bmatrix} 0.99997 & -4.0162\times10^{-4} & -7.6775\times10^{-3} \\ 2.6487\times10^{-4} & -0.99624 & 0.086613 \\ -7.6835\times10^{-3} & -0.086613 & -0.99621 \end{bmatrix}$
2	$\begin{bmatrix} 0.99997 & -4.1496\times10^{-4} & -7.6352\times10^{-3} \\ 2.487\times10^{-3} & -0.99623 & 0.086715 \\ -7.6424\times10^{-3} & -0.086715 & -0.9962 \end{bmatrix}$
⋮	⋮
5	$\begin{bmatrix} 0.99997 & -3.9354\times10^{-4} & -7.5915\times10^{-3} \\ 2.6529\times10^{-4} & -0.99624 & 0.086589 \\ -7.5971\times10^{-3} & -0.086589 & -0.99622 \end{bmatrix}$

Table 3.14 The estimated rotation matrix of the third camera location from five runs of optimization.

Iteration times	$[T]_3$
0 **(Assumed)**	$\begin{bmatrix} 0.9446 & 0.22189 & -0.24184 \\ 0.28014 & -0.929 & 0.24184 \\ -0.17101 & -0.2962 & -0.93969 \end{bmatrix}$
1	$\begin{bmatrix} 0.94462 & 0.22182 & -0.24184 \\ 0.28013 & -0.92892 & 0.24216 \\ -0.17093 & -0.2965 & -0.93961 \end{bmatrix}$
2	$\begin{bmatrix} 0.94462 & 0.2218 & -0.24186 \\ 0.28014 & -0.92888 & 0.24229 \\ -0.17092 & -0.29663 & -0.93957 \end{bmatrix}$
⋮	⋮
5	$\begin{bmatrix} 0.94464 & 0.22185 & -0.24175 \\ 0.28013 & -0.92892 & 0.24214 \\ -0.17085 & -0.29645 & -0.93964 \end{bmatrix}$

After the fifth run of optimization, the three Euler angles of each camera location are

$$\begin{Bmatrix}\theta_1\\ \theta_2\\ \theta_3\end{Bmatrix}=\begin{Bmatrix}4.9711^\circ\\ 159.9862^\circ\\ 9.9775^\circ\end{Bmatrix},\begin{Bmatrix}5.0105^\circ\\ 184.9833^\circ\\ 5.0141^\circ\end{Bmatrix},\begin{Bmatrix}44.9538^\circ\\ 200.0088^\circ\\ 29.9557^\circ\end{Bmatrix}$$

Since the noise is not very big, they are almost the same with the assumed ones. Figure 3.7 shows the locations of three simulated camera locations.

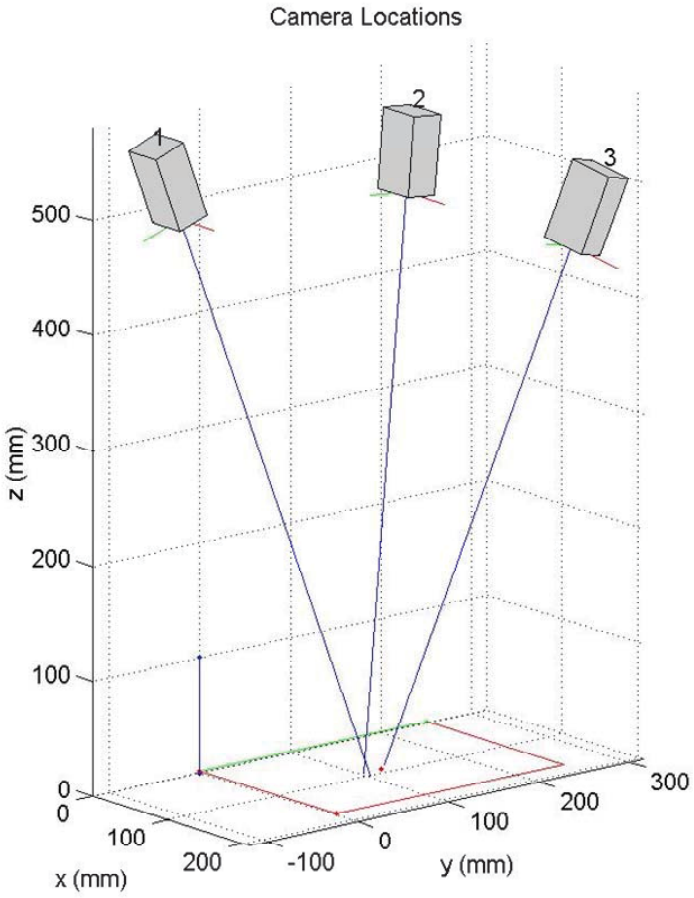


Figure 3.7 The three locations of the simulated camera ($\sigma = 0.1$).

If the random noise added to the distorted image coordinates $(\tilde{u},\tilde{v})$ has $\sigma = 1$, Table 3.15 shows that the converged objective function has a value larger than that corresponding to $\sigma = 0.1$ (Table 3.7).

Table 3.15 Minimization of the objective function F_0 during five runs of optimization.

Iteration times	***1***	***2***	***3***	***4***	***5***
$F_{0\min}$	736.93	736.91	736.89	736.88	736.88

Tables 3.16 and 3.17 show the convergence of the forward and backward lens distortion models. Table 3.18 shows the changes of the five intrinsic parameters. Table 3.19 shows the convergence of translation vectors of the three simulated camera locations. Tables 3.20-3.22 show the convergence of the rotational matrix of each simulated camera location. Since more noise is added, the differences between these parameters and the assumed ones increase.

Table 3.16 The estimated forward lens distortion model.

Iteration times	$\hat{K}_1$	$\hat{K}_2$	$\hat{P}_1$	$\hat{P}_2$	$\hat{S}_1$	$\hat{S}_2$
***0* (Assumed)**	-0.23	0.2	0.0	0.0	0.0	0.0
1	-0.24832	0.3506	-3.6855×10^{-4}	-4.5658×10^{-4}	-4.4533×10^{-4}	1.702×10^{-3}
2	-0.249	0.35609	-8.1169×10^{-4}	-6.8673×10^{-4}	3.6803×10^{-4}	2.1192×10^{-3}
3	-0.24928	0.35824	-1.0226×10^{-3}	-7.8167×10^{-4}	7.4295×10^{-4}	2.2871×10^{-3}
4	-0.2494	0.35916	-1.1234×10^{-3}	-8.2139×10^{-4}	9.2212×10^{-4}	2.3571×10^{-3}
5	-0.24945	0.35954	-1.1713×10^{-3}	-8.3753×10^{-4}	1.0071×10^{-3}	2.3857×10^{-3}

Table 3.17 The estimated backward lens distortion model.

Iteration times	K_1	K_2	P_1	P_2	S_1	S_2
***0* (Assumed)**	–	–	–	–	–	–
1	0.25331	-0.29825	6.8775×10^{-4}	7.362×10^{-4}	3.7479×10^{-5}	-2.4773×10^{-3}
2	0.25394	-0.30303	1.1701×10^{-3}	9.8416×10^{-3}	-8.3551×10^{-4}	-2.9171×10^{-3}
3	0.25419	-0.30492	1.3976×10^{-3}	1.0873×10^{-3}	-1.2347×10^{-3}	-3.0963×10^{-3}
4	0.2543	-0.30569	1.5061×10^{-3}	1.1305×10^{-3}	-1.4253×10^{-3}	-3.1709×10^{-3}
5	0.25435	-0.306	1.5578×10^{-3}	1.148×10^{-3}	-1.5158×10^{-3}	-3.2014×10^{-3}

Table 3.18 The estimated intrinsic parameters from the five runs of optimization.

Iteration times (pixel)	α	β	γ	u_0	v_0
***0* (Assumed)**	1250	900	1.0908	255	255
1	1265.2	911.55	-0.37343	248.24	249.53
2	1265.7	911.92	-0.38321	245.96	248.72
3	1265.6	911.86	-0.37794	244.97	248.41
4	1265.6	911.84	-0.37531	244.5	248.27
5	1265.6	911.82	-0.37427	244.27	248.22

Table 3.19 The estimated translation vectors of the three camera locations from five runs of optimization.

Iteration times (mm)	$\{\tau\}_1^T$	$\{\tau\}_2^T$	$\{\tau\}_3^T$
***0* (Assumed)**	{-90 105 500}	{-90 105 510}	{-105 105 525}
1	{-87.187 108.05 506.19}	{-87.161 108.11 515.82}	{-102.07 107.95 530.83}
2	{-86.274 108.49 506.44}	{-86.231 108.57 516.07}	{-101.11 108.42 531.1}
3	{-85.878 108.67 506.43}	{-85.827 108.75 516.06}	{-100.69 108.6 531.11}
4	{-85.689 108.74 506.43}	{-85.634 108.82 516.06}	{-100.5 108.68 531.11}
5	{-85.599 108.77 506.43}	{-85.542 108.85 516.06}	{-100.4 108.71 531.12}

Table 3.20 The estimated rotation matrix of the first camera location from five runs of optimization.

Iteration times	$[T]_1$
***0* (Assumed)**	$\begin{bmatrix} 0.99528 & -0.092332 & 0.029809 \\ -0.076723 & -0.93703 & -0.34072 \\ 0.059391 & 0.33682 & -0.93969 \end{bmatrix}$
1	$\begin{bmatrix} 0.99528 & -0.092332 & 0.029809 \\ -0.076723 & -0.93703 & -0.34072 \\ 0.059391 & 0.33682 & -0.93969 \end{bmatrix}$
2	$\begin{bmatrix} 0.99564 & -0.090756 & 0.021548 \\ -0.077642 & -0.93435 & -0.34781 \\ 0.051699 & 0.34462 & -0.93732 \end{bmatrix}$
⋮	⋮
5	$\begin{bmatrix} 0.99571 & -0.090292 & 0.020277 \\ -0.077617 & -0.93416 & -0.34831 \\ 0.050391 & 0.34524 & -0.93716 \end{bmatrix}$

Table 3.21 The estimated rotation matrix of the second camera location from five runs of optimization.

Iteration times	$[T]_2$
0 **(Assumed)**	$\begin{bmatrix} 0.99997 & -3.3039\times10^{-4} & -7.5961\times10^{-3} \\ 3.3039\times10^{-4} & -0.99622 & 0.086824 \\ -7.5961\times10^{-3} & -0.086824 & -0.99619 \end{bmatrix}$
1	$\begin{bmatrix} 0.99992 & -1.4816\times10^{-3} & -0.012817 \\ -4.3752\times10^{-4} & -0.99671 & 0.081081 \\ -0.012895 & -0.081069 & -0.99663 \end{bmatrix}$
2	$\begin{bmatrix} 0.99989 & -1.6357\times10^{-3} & -0.014555 \\ -4.6241\times10^{-4} & -0.99677 & 0.080252 \\ -0.014639 & -0.080237 & -0.99667 \end{bmatrix}$
⋮	⋮
5	$\begin{bmatrix} 0.99987 & -1.7402\times10^{-3} & -0.01589 \\ -4.6849\times10^{-4} & -0.99682 & 0.079688 \\ -0.015978 & -0.07967 & -0.99669 \end{bmatrix}$

Table 3.22 The estimated rotation matrix of the third camera location from five runs of optimization.

Iteration times	$[T]_3$
0 **(Assumed)**	$\begin{bmatrix} 0.9446 & 0.22189 & -0.24184 \\ 0.28014 & -0.929 & 0.24184 \\ -0.17101 & -0.2962 & -0.93969 \end{bmatrix}$
1	$\begin{bmatrix} 0.944 & 0.21976 & -0.24609 \\ 0.27937 & -0.92923 & 0.24183 \\ -0.17554 & -0.29704 & -0.93859 \end{bmatrix}$
2	$\begin{bmatrix} 0.94369 & 0.21922 & -0.24777 \\ 0.27921 & -0.92948 & 0.24108 \\ -0.17745 & -0.29668 & -0.93835 \end{bmatrix}$
⋮	⋮
5	$\begin{bmatrix} 0.94346 & 0.21883 & -0.249 \\ 0.27911 & -0.92965 & 0.24054 \\ -0.17885 & -0.29643 & -0.93816 \end{bmatrix}$

After the fifth run of optimization, the three Euler angles are

$$\begin{Bmatrix}\theta_1\\ \theta_2\\ \theta_3\end{Bmatrix}=\begin{Bmatrix}3.3317^\circ\\ 159.5800^\circ\\ 8.3042^\circ\end{Bmatrix},\begin{Bmatrix}11.2770^\circ\\ 184.6631^\circ\\ 11.3404^\circ\end{Bmatrix},\begin{Bmatrix}45.9901^\circ\\ 200.2552^\circ\\ 31.1045^\circ\end{Bmatrix}$$

Because of the big noise, the differences between these angles and assumed ones increase. Figure 3.8 shows the locations of three simulated camera locations.

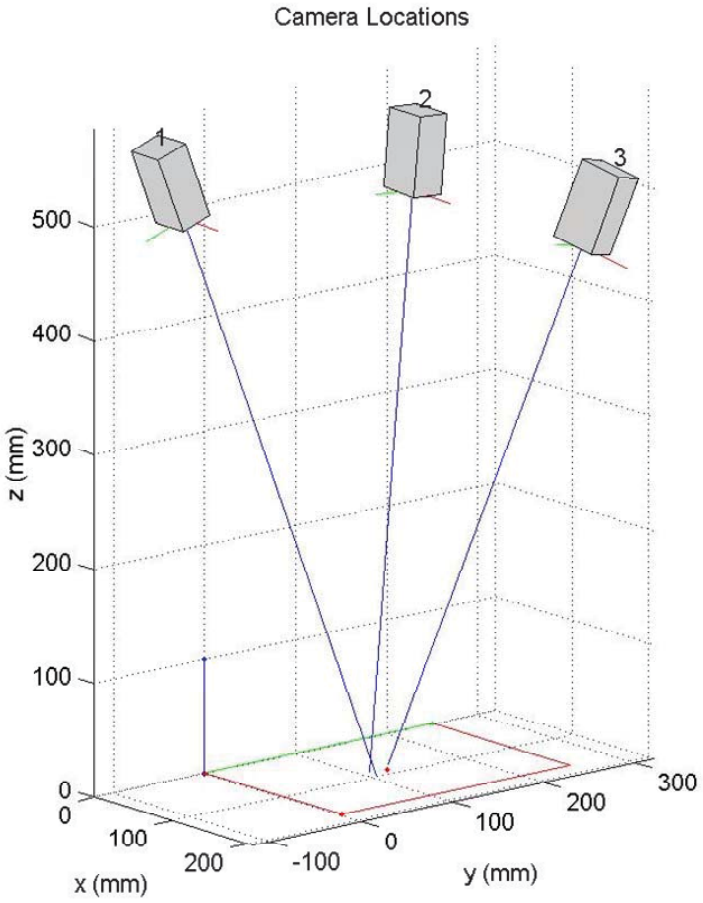


Figure 3.8 The three locations of the simulated camera ($\sigma = 1$).

When one of the three Euler angles equals to 180° or 0°, singularity problems exist. It means that, when calculating Euler angles from the rotation matrix, the other two Euler angles either cannot be obtained or have no definite values. In this case, one needs

to use the *principal rotation angle* method presented in (3.1.6)-(3.1.10). However, because $\theta_i = 180^\circ$ or 0° is rare to actually happen in real numbers, the use of Euler angles and (3.1.5a)-(3.1.5g) often works.

3.2 Selection of Experiment Conditions

Camera manufacturers normally provide with each purchased camera some utility software. For example, a Canon EOS-7D DSLR camera comes with the Digital Photo Professional for editing images and converting RAW images to other formats, the EOS Utility for remote camera setting and shooting using a computer, the Picture Style Editor for changing image styles and editing image characteristics, and the Zoom Browser EX for organizing and editing images and movies. As shown later in Sections 3.2.1 and 3.2.5, an image edited/converted by the Digital Photo Professional may not have a better accuracy in corner detection and calibration, and hence it is bootless for us. Since the EOS Utility can only control one camera at one time, it cannot be used for simultaneous control of multiple cameras. Because we need vivid, sharp and crisp images for camera calibration and static/dynamic measurements, the standard picture style will be used and hence the Picture Style Editor is also useless for us. However, because the Zoom Browser EX can extract static frames from a movie file, it is very useful for the purpose of dynamic measurement.

3.2.1 Image File Formats

Here we show that the image file format used for image processing can affect the accuracy of corner detection. A commercial DSLR camera often provides RAW, TIFF, and JPEG images.

A RAW image is directly available from the photo sensor without processing but is not ready for printing or editing. It is also called a digital negative image and is similar to a negative film in photography. Although the standard RAW image format is named ISO 12234-2 [95], different camera manufacturers use different format names, and CR2 is used by Canon. A RAW image can be converted to other formats by accessory utility software from the manufacturer (e.g., the Digital Photo Professional of Canon).

A TIFF (Tagged Image File Format) image has lossless or no compression [96], and it can be edited without losing image quality. A JPEG (Joint Photographic Experts Group) image is obtained using a lossy compression method, and it is usually stored in the JPEG File Interchange Format with the extension JPG or JPEG [97-101].

In order to reduce or avert the effect of shaking caused by touching the shutter button, we recommend using a self-timer or a remote controller. Figure 3.9 shows a Canon RC-6 wireless remote controller. For a Canon EOS-7D camera, since its wireless remote control sensor is on the front side (see Figure 3.10), the wireless remote control doesn't work well if the controller is fired behind the camera. Hence, we recommend the use of both the wireless remote controller and the self-timer to allow the experimenter have time to move out of the scene after shooting the remote controller.

Figure 3.9 A Canon RC-6 wireless remote controller.

Figure 3.10 The location of a Canon EOS-7D camera's remote control sensor.

3.2.2 Image Recording Quality

Each commercial digital camera provides different image recording qualities for customers to choose. Table 3.23 shows the different recording qualities and their dimensions of a Canon EOS-7D DSLR camera. Notice that the Medium RAW and the Medium JPEG have different dimensions. The photo sensor of a Canon EOS-7D camera

has a size of $22.3\times14.9\ \text{mm}^2$ with 5184×3456 pixels, and hence its sensor density is $r = 5.4\ \text{M pixels/cm}^2$. Hence, the horizontal and vertical distances between two adjacent pixels can be calculated to be $\Delta_1 = 22.3\text{mm}/5184 = 4.3\mu\text{m}$ and $\Delta_2 = 14.9\text{mm}/3456 = 4.3\mu\text{m}$.

For an image with less recording quality, some pixels on the photo sensor are jumped over and are not used. The Sensor Pixels Coefficient *e* is defined as the ratio between the dimensions of the sensor and the image, and it is listed in the last column of Table 3.23 for different recording qualities. If a JPEG image's length *L* along one direction is presented in pixel, it can be changed into L' in millimeter by using the following formula:

$$L' = \frac{L}{\sqrt{r}} \times e \times 10 \tag{3.2.1}$$

Alternatively, (3.2.1) can be rewritten as

$$L_1' = L_1 \times e \times \Delta_1, \quad L_2' = L_2 \times e \times \Delta_2 \tag{3.2.2}$$

where L_1 and L_2 represent the lengths in pixel along the two different directions.

Figure 3.11 shows the difference between Large Fine and Large Normal JPEG images. A Fine JPEG image has more grey levels than a Normal JPEG image, and hence it requires more storage space.

Table 3.23 Different image recording qualities of a Canon EOS-7D DSLR camera.

Image-recording Quality	Dimensions (*pixel* × *pixel*)	Sensor Pixels Coefficient
Large Fine JPEG	5184×3456	**1**
Large Normal JPEG	5184×3456	**1**
8bit TIFF from RAW	5184×3456	**1**
Medium Fine JPEG	3456×2304	**1.5**
Medium Normal JPEG	3456×2304	**1.5**
8bit TIFF from Medium RAW	3888×2592	**4/3**
Small Fine JPEG	2592×1728	**2**
Small Normal JPEG	2592×1728	**2**
8bit TIFF from Small RAW	2592×1728	**2**

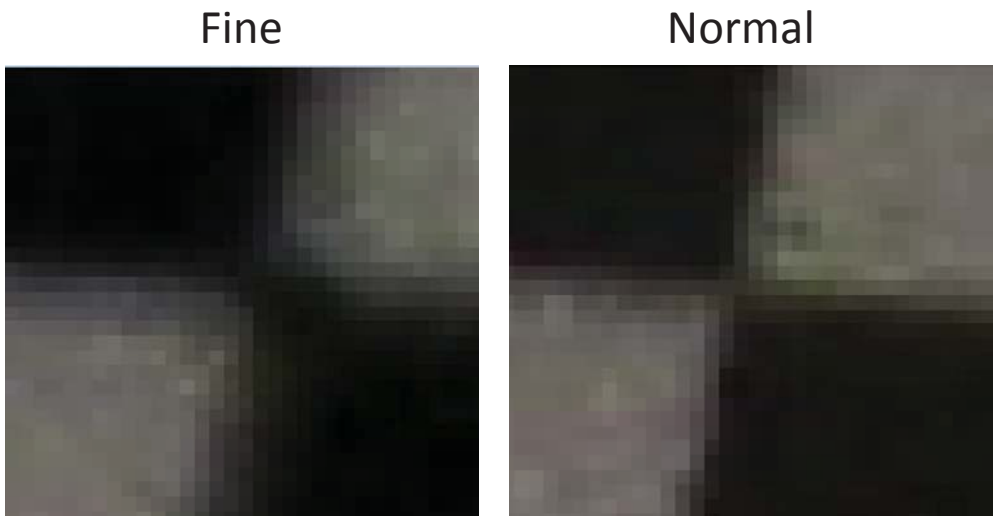


Figure 3.11 Difference between a Large Fine JPEG image and a Large Normal JPEG image of the checker board's squares.

3.2.3 Comparison of Image File Formats

To compare camera calibration results using images in different formats we choose the three images shown in Figure 3.12, where the side length of the squares on the checker board is 30mm. The three images are taken from different view angles using a Canon EOS-7D camera. The images are captured with ISO speed at 4000, *f*-number at *f*/8.0, shutter speed at 1/80 second, and a focal length of 80mm.

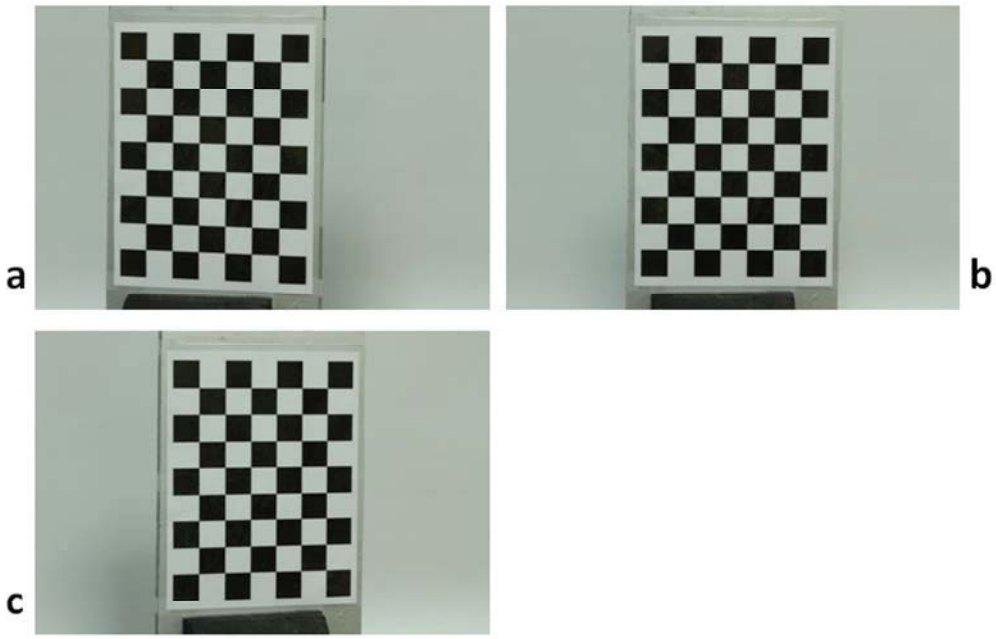


Figure 3.12 Three images of the checker board taken from different view angles

Table 3.24 shows the 6 forward and 6 backward lens distortion parameters extracted from the three images in the small fine JPEG format, and the 5 forward lens distortion parameters by using the Caltech Camera Calibration Toolbox for MATLAB [42]. The corresponding parameter values of the two models have opposite signs and about the same absolute values. Table 3.25 compares the root-mean-square (RMS) errors of the estimated image coordinates of the corner points when the a, b and c images of

Figure 3.12 in three different formats are used. The computed world coordinates (x, y, z) of the corner points on the checker board can be converted to the undistorted coordinates (u, v) on the image plane by using (1.1.1). Then, the distorted image plane coordinates $(\tilde{u}, \tilde{v})$ can be computed by using the obtained forward lens distortion model. After that the RMS errors between the measured $(\tilde{u}, \tilde{v})$ and the calculated $(\tilde{u}, \tilde{v})$ of all corner points can be calculated.

Table 3.26 compares the RMS errors of the estimated world coordinates of the corner points when the a, b and c images of Figure 3.12 in three different formats are used. One can use the obtained backward lens distortion model to compute the undistorted image coordinates (u, v) and then use (1.1.1) to compute the world coordinates (x, y, z). After that the RMS errors between the known world coordinates (x, y, z) of corner points on the checker board and the computed world coordinates (x, y, z) of all corner points can be calculated.

Table 3.24 Forward and backward lens distortion models from three small fine JPEG images.

Forward model		**Backward model**		**Forward model (Caltech Toolbox)**	
$\hat{K}_1$	0.29432	K_1	-0.29343	$\hat{K}_1$	0.23704
$\hat{K}_2$	-2.1083	K_2	2.1084	$\hat{K}_2$	3.42114
$\hat{P}_1$	5.6489×10^{-3}	P_1	-5.6208×10^{-3}	$\hat{P}_1$	0.01362
$\hat{P}_2$	1.3392×10^{-4}	P_2	-1.4071×10^{-4}	$\hat{P}_2$	5.29×10^{-3}
$\hat{S}_1$	4.8705×10^{-3}	S_1	-4.8659×10^{-3}	$\hat{K}_3$	0.00000
$\hat{S}_2$	3.6251×10^{-3}	S_2	-3.6059×10^{-3}		

Table 3.25 The RMS errors of the estimated image coordinates of the corner points in the a, b and c images of Figure 3.12 when the small fine JPEG, small normal JPEG, and 8bit TIFF converted from small RAW of these images are used.

RMS (pixel)	Small Fine JPEG		Small Normal JPEG		8bit TIFF from Small RAW	
	u axis	v axis	u axis	v axis	u axis	v axis
Image a	0.7382	0.3898	0.7070	0.3642	0.8034	0.4009
Image b	0.5752	0.3213	0.5637	0.3210	0.5980	0.3191
Image c	0.5141	0.3283	0.5082	0.3291	0.5427	0.3336

Table 3.26 The RMS errors of the estimated world coordinates of the corner points in the a, b and c images of Figure 3.12 when the small fine JPEG, small normal JPEG, and 8bit TIFF converted from small RAW of these images are used.

RMS (mm)	Small Fine JPEG		Small Normal JPEG		8bit TIFF from Small RAW	
	x axis	y axis	x axis	y axis	x axis	y axis
Image a	0.7781	2.2171	0.7713	2.2112	0.7775	2.2166
Image b	1.2919	0.6602	1.2882	0.6623	1.3060	0.6719
Image c	2.2289	6.9538	2.2382	6.9385	2.2335	6.9868

Tables 3.25 and 3.26 show that, although TIFF images use a lossless compression method and JPEG images use a lossy one, TIFF images may not be better than JPEG images for accurate corner detection and camera calibration. In other words, image quality (e.g., pixel density) is not a major factor for accurate corner detection.

The obtained intrinsic and extrinsic parameters are shown below:

$$\alpha = 73.073\text{mm},\ \beta = 73.295\text{mm},\ \gamma = -0.038393\text{mm},\ u_0 = 11.439\text{mm},\ v_0 = 7.8453\text{mm}$$

$$\{\tau\} \equiv \begin{Bmatrix} \xi_0 \\ \eta_0 \\ \zeta_0 \end{Bmatrix} = \begin{Bmatrix} 15.467 \\ -350.5 \\ 1585.1 \end{Bmatrix}, \begin{Bmatrix} 72.481 \\ -353.1 \\ 1654.1 \end{Bmatrix}, \begin{Bmatrix} 68.348 \\ -356.39 \\ 1693.4 \end{Bmatrix} \text{mm};\quad \begin{Bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{Bmatrix} = \begin{Bmatrix} 66.2530^\circ \\ 192.4936^\circ \\ 65.1252^\circ \end{Bmatrix}, \begin{Bmatrix} 329.2583^\circ \\ 185.4012^\circ \\ 328.8721^\circ \end{Bmatrix}, \begin{Bmatrix} 287.4028^\circ \\ 194.9769^\circ \\ 287.7076^\circ \end{Bmatrix}$$

Figure 3.13 shows the three locations of the camera.

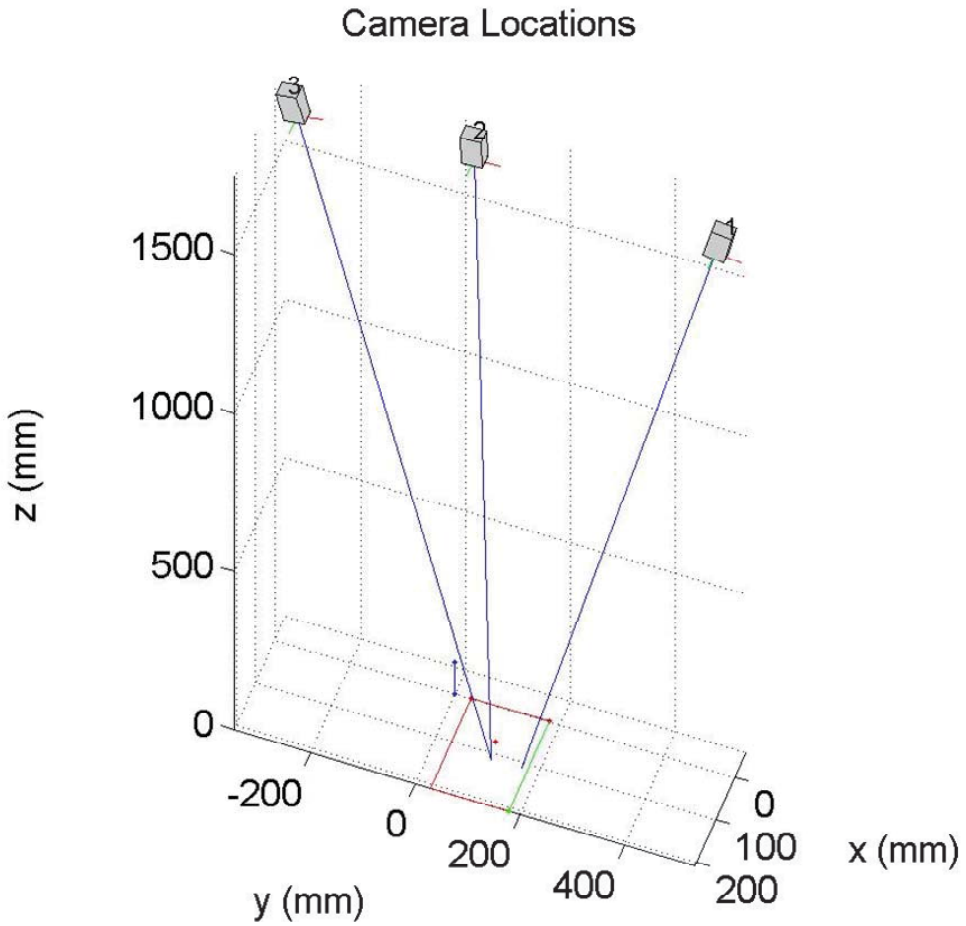


Figure 3.13 The three locations of the camera.

Figure 3.14 shows another group of three images. The only difference between this group and the group in Figure 3.12 is the view angle. Angles between cameras are larger in this group.

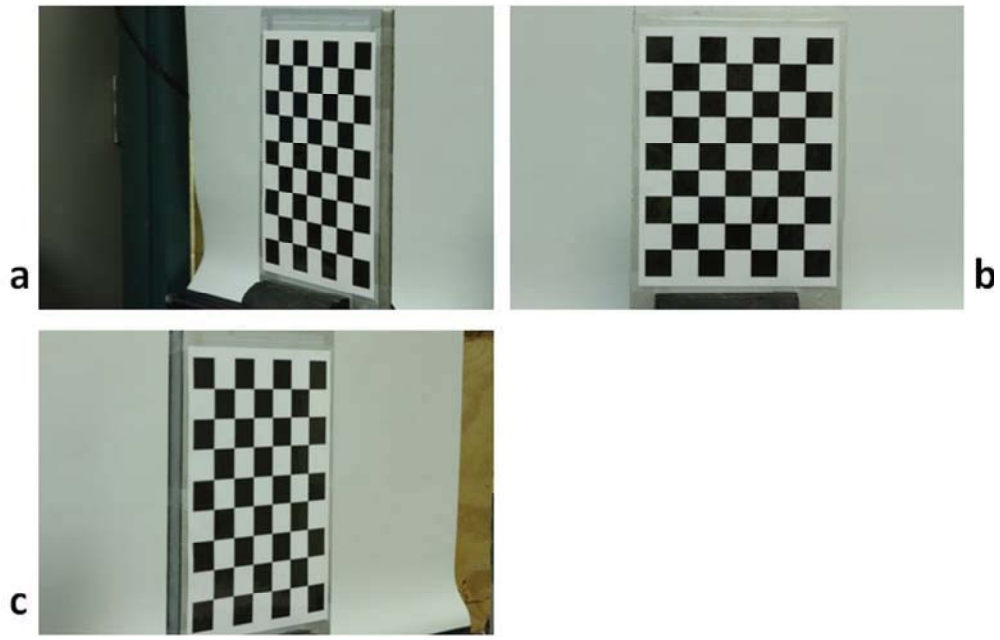


Figure 3.14 Three images of the checker board taken from three different view angles.

Table 3.27 shows the forward and backward lens distortion parameters extracted from three small fine JPEG images of Figure 3.14. The corresponding parameter values of the two models have opposite signs and about the same absolute values. Although the images in Figures 3.12 and 3.14 are taken using the same camera, Tables 3.24 and 3.27 show that the obtained lens distortion models are different. It is because the checker board image in Figure 3.14 does not cover a large part of the sensor plane. In order to obtain an accurate lens distortion model, the checker board image should cover a large part of the sensor plane. On the other hand, since lens distortion is an inherent property of

a lens, a previous model can be used if the checker board does not cover a large part of the sensor plane.

Table 3.27 Forward and backward lens distortion models from three small fine JPEG images.

Forward model		**Backward model**		**Forward model (Caltech Toolbox)**	
$\hat{K}_1$	0.97488	K_1	-0.96674	$\hat{K}_1$	1.03847
$\hat{K}_2$	-82.67	K_2	81.289	$\hat{K}_2$	-70.62044
$\hat{P}_1$	-4.5415×10^{-3}	P_1	4.4948×10^{-3}	$\hat{P}_1$	4.96×10^{-3}
$\hat{P}_2$	7.0542×10^{-4}	P_2	-7.4709×10^{-4}	$\hat{P}_2$	0.01078
$\hat{S}_1$	8.7278×10^{-3}	S_1	-8.6511×10^{-3}	$\hat{K}_3$	0.00000
$\hat{S}_2$	-1.5795×10^{-4}	S_2	2.4218×10^{-4}		

Table 3.28 shows the RMS errors of the image coordinates and the world coordinates. Tables 3.25, 3.26 and 3.28 show that, when the view angle is more away from the normal direction (i.e., 90°), the errors increase. Since this system is to be used for measurement, we care more about the errors in the world coordinates. It is caused by different depth the checker board crossed. The image is sharper when the object is closer to the focal plane. When some part of the object is away from the plane it will be unclear. Since circle of confusion is defined by resolution of human eyes (see section 2.2.3), although the object looks sharp and clear in depth of field, error is still exist. On the other hand, more experiments indicate that the focal length does not affect the accuracy of camera calibration.

Table 3.28 The RMS errors of the estimated image coordinates and world coordinates of the corner points in the a, b and c images of Figure 3.14 when the small fine JPEG of these images are used.

RMS (pixel)	Small Fine JPEG	
	u axis	v axis
Image a	0.8995	0.60932
Image b	0.5154	0.53235
Image c	0.66252	0.64802

RMS (mm)	Small Fine JPEG	
	x axis	y axis
Image a	5.0016	44.168
Image b	1.2879	0.67063
Image c	4.7775	50.636

The three Euler angles of these three images are

$$\begin{Bmatrix}\theta_1\\ \theta_2\\ \theta_3\end{Bmatrix} = \begin{Bmatrix}86.2592^\circ\\ 237.3688^\circ\\ 83.0962^\circ\end{Bmatrix}, \begin{Bmatrix}344.1428^\circ\\ 184.8496^\circ\\ 343.8202^\circ\end{Bmatrix}, \begin{Bmatrix}273.5464^\circ\\ 229.6947^\circ\\ 275.1557^\circ\end{Bmatrix}$$

Figure 3.15 shows the three locations of the camera.

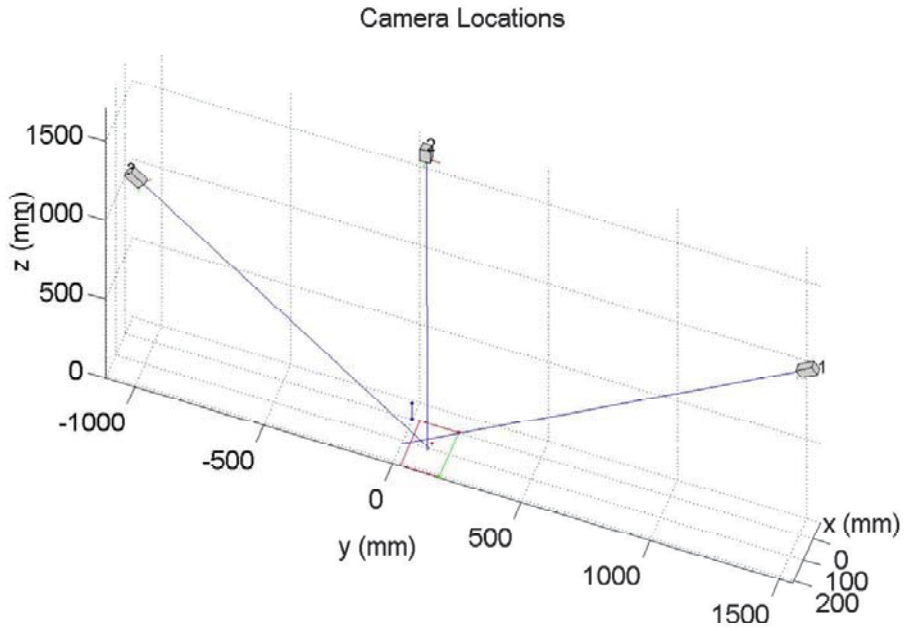


Figure 3.15 The three locations of the camera.

3.2.4 Comparison of Different Cameras

As shown in Figures 3.16 and 3.17, the image captured by a Canon EOS-7D camera with a Canon EF lens has less lens distortion than that by a Sony DSC-TX1 camera with a Carl Zeiss Vario-Tessar lens.

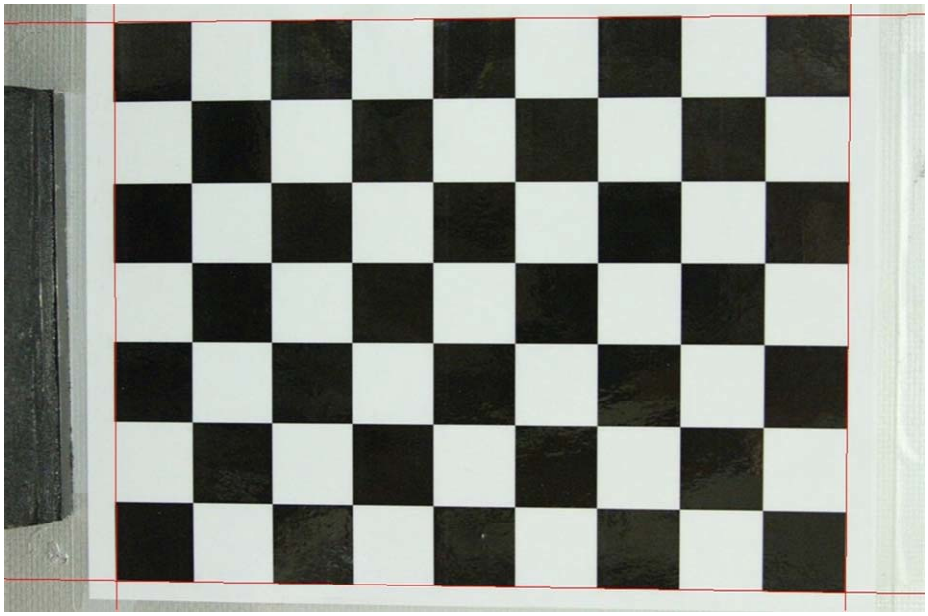


Figure 3.16 Image captured by a Canon EOS-7D camera with a Canon EF lens.

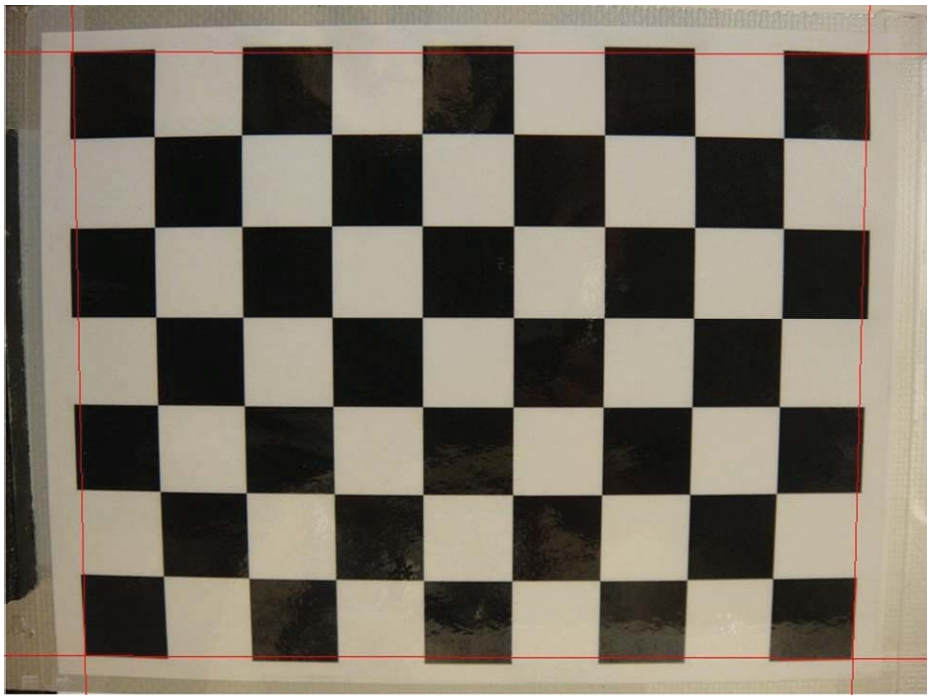


Figure 3.17 Image captured by a Sony DSC-TX1camera with a Carl Zeiss Vario-Tessar lens.

Figure 3.18 are three images of the checker board captured by a Sony DSC-TX1 camera. After calibration, the forward and backward distortion models are shown in Table 3.29.Tables 3.24 and 3.29 show that the lens distortion of the Carl Zeiss Vario-Tessar lens is worse than that of the Canon EF lens.

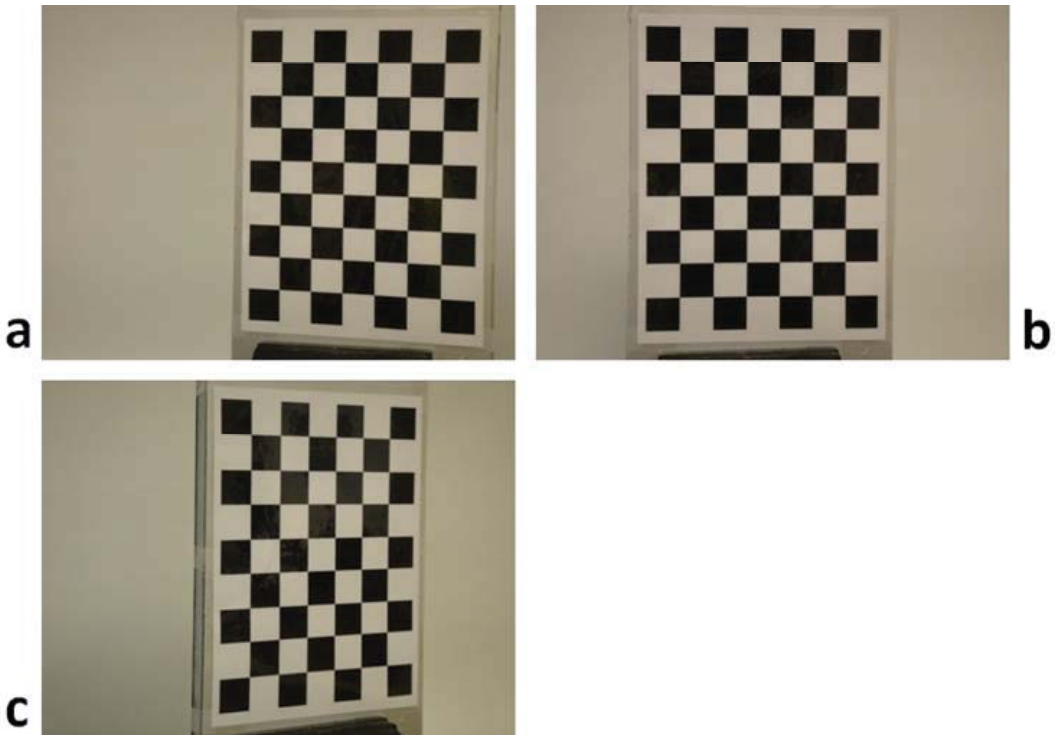


Figure 3.18 Three images of the checker board captured by a Sony DSC-TX1 camera.

Table 3.29 Forward and backward lens distortion models of the Carl Zeiss Vario-Tessar lens of a Sony DSC-TX1 camera.

Forward model		**Backward model**		**Forward model (Caltech Toolbox)**	
$\hat{K}_1$	0.37964	K_1	-0.37531	$\hat{K}_1$	0.56790
$\hat{K}_2$	2.8268	K_2	-2.8881	$\hat{K}_2$	-3.35626
$\hat{P}_1$	-0.040082	P_1	0.039834	$\hat{P}_1$	0.01885
$\hat{P}_2$	-7.3387×10^{-3}	P_2	7.346×10^{-3}	$\hat{P}_2$	0.01404
$\hat{S}_1$	0.029788	S_1	-0.029525	$\hat{K}_3$	0.00000
$\hat{S}_2$	0.01019	S_2	-0.010218		

3.2.5 Video Recording Quality and Extraction of Image Frames from Video Files

For Canon EOS-7D cameras, one can choose to use the NTSC or the PAL format for video recording. As mentioned in Section 2.3.1, they use the same formula to calculate Y (luminance, brightness, or gray value).

Table 3.30 shows different movie recording sizes that can be chosen for a Cannon EOS-7D camera. The 1920×1080 size is called the full high-definition recording quality, the 1280×720 size the high-definition recording quality, and the 640×480 size the standard recording quality. Since static images captured by the camera are used to do calibration and frames from movies are used for dynamic measurements, they have different sizes and this problem needs to be solved before using the intrinsic and extrinsic parameters obtained from camera calibration for measurement. A static image or a video frame with $a\times b$ pixels has a dimension ratio a/b. As shown in Table 3.23, all static images taken by Canon EOS-7D cameras have the same dimension ratio of 1.5. But video frames from different video files may have different dimension ratios, as shown in Table 3.30. When calculating the Sensor Pixels Coefficient e for frames from a video file, the full screen direction should be considered. For example, if $a/b>1.5$, the screen in the first direction is fully filled with image and the sensor pixels coefficient is $e=5184/a$. If $a/b<1.5$, the screen in the second direction is fully filled with image and the sensor pixels coefficient is $e=3456/b$.

The frame rate (or frame frequency) is often expressed in frames per second (FPS). There are several main frame rate standards: 24 FPS for movie films [102]; 25 FPS for the PAL color TV system [103]; and 30 FPS for the NTSC color TV system [104].

Table 3.30 Movie recording properties of Canon EOS-7D cameras.

Dimensions ($pixel \times pixel$)	**Frame Rate (FPS)**[1]	**Full Screen Direction**[2]	**Sensor Pixels Coefficient**
1280×720	60	1	4.05
640×480	60	2	7.2
1920×1080	30	1	2.7
1920×1080	24	1	2.7

[1]For a Canon EOS-7D camera, the actual frame rates are 59.94 FPS in the 60 FPS mode, 29.97 FPS in the 30 FPS mode, and 23.976 FPS in the 24 FPS mode.

[2]1 means the first direction (i.e., *a* in $a \times b$) and 2 means the second direction (i.e., *b* in $a \times b$).

The step after video recording is to extract frames from the video file. Some commercial software (like the ZoomBrowser EX of Canon EOS-7D cameras) can be used to extract frames, and then the frames can be processed as singular pictures. As mentioned before, because static images and frames from video files have different dimensions, they have different lengths between two adjacent pixels. Hence, one needs to use (3.2.1) or (3.2.2) to unify the units into millimeter.

3.2.6 Comparison of Post Processing Using Canon Software

The Digital Photo Professional software came with Canon EOS-7D cameras can convert RAW files to other formats and can also reduce noise and correct lens distortion. In this software, the maximal magnitudes of both luminance noise reduction and chrominance noise reduction are 20, but the default noise reduction magnitudes change

with the image-recording quality of RAW files and other settings of the camera (e.g., white balance), as shown in Figure 3.19.

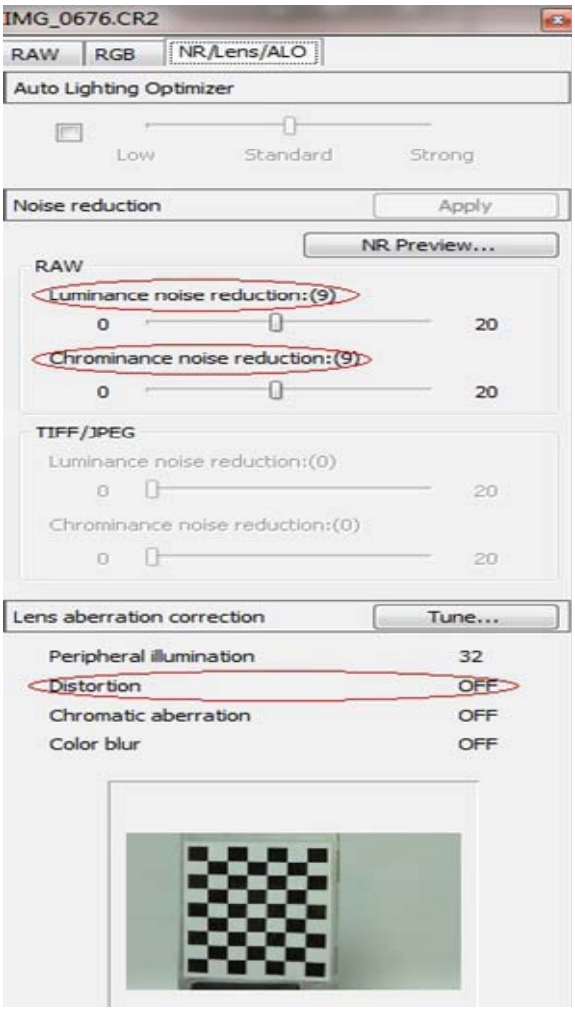


Figure 3.19 Tool palette of the Digital Photo Professional software.

One can click the button "Tune" then click "Distortion" to turn on the distortion correction. Tables 3.31 and 3.32 compare the RMS errors of image coordinates and world coordinates from the use of small fine JPEG format images, 8 bit TIFF format images converting from small RAW files, 8 bit TIFF images with magnitudes of both luminance and chrominance noise reductions equal 20, and 8 bit TIFF images with distortion

correction "on" by this software. Images shown in Figure 3.12 are processed to obtain Tables 3.31 and 3.32.

Table 3.31 RMS errors in image coordinates from Figure 3.12.

RMS (pixel)	Small Fine JPEG		8bit TIFF from Small RAW		8bit TIFF with Noise Reduction		8bit TIFF with Distortion Correction	
	u axis	v axis	u axis	v axis	u axis	v axis	u axis	v axis
Image a	0.7382	0.3898	0.8034	0.4009	0.8019	0.4033	1.0623	0.4429
Image b	0.5752	0.3213	0.5980	0.3191	0.5936	0.3105	0.5864	0.3125
Image c	0.5141	0.3283	0.5427	0.3336	0.5390	0.3318	0.5189	0.3317

Table 3.32 RMS errors in world coordinates from Figure 3.12.

RMS (mm)	Small Fine JPEG		8bit TIFF from Small RAW		8bit TIFF with Noise Reduction		8bit TIFF with Distortion Correction	
	x axis	y axis	x axis	y axis	x axis	y axis	x axis	y axis
Image a	0.7781	2.2171	0.7775	2.2166	0.7798	2.2142	0.8146	2.2510
Image b	1.2919	0.6602	1.3060	0.6719	1.3099	0.6718	1.3272	0.6987
Image c	2.2289	6.9538	2.2335	6.9868	2.2358	6.9766	2.2513	6.9996

The results show that, with noise reduction using the Digital Photo Professional software, the errors may be smaller than those from the original TIFF images converted from Small RAW files, but not as accurate as those from JEPG format files. However, distortion correction generally increases the errors. It may be caused by the unknown lens

distortion model used in the Digital Photo Professional. So we do not recommend the use of the Digital Photo Professional. Since most non-SLR digital cameras do not have the RAW format and the JPEG format is not bad at all in terms of precision and convenience, we recommend the use of JPEG images.

3.2.7 Comparison of Image Recording Quality

Next we compare the use of different image recording qualities. Due to limitations on memory and computational time, we only compare small fine/normal JPEG images, medium fine/normal JPEG images, and 8bit TIFF images converting from small and medium RAW files. Due to differences in dimensions, all units in image coordinates are unified into pixels on the image sensor. Images shown in Figure 3.12 are processed to obtain Tables 3.33 to 3.35.

Table 3.33 RMS errors in image coordinates from Figure 3.12.

RMS (pixel in image sensor)	Small Fine JPEG		Small Normal JPEG		Medium Fine JPEG		Medium Normal JPEG	
	u axis	v axis	u axis	v axis	u axis	v axis	u axis	v axis
Image a	1.4764	0.7796	1.4140	0.7284	1.5429	0.6912	1.4793	0.7677
Image b	1.1504	0.6426	1.1274	0.6420	1.1717	0.6570	1.1448	0.6513
Image c	1.0282	0.6566	1.0164	0.6582	1.0725	0.6837	1.0397	0.6711

Table 3.34 RMS errors in world coordinates from Figure 3.12.

RMS (mm)	Small Fine JPEG		Small Normal JPEG		Medium Fine JPEG		Medium Normal JPEG	
	x axis	y axis	x axis	y axis	x axis	y axis	x axis	y axis
Image a	0.7781	2.2171	0.7713	2.2112	0.7654	2.2068	0.7760	2.2259
Image b	1.2919	0.6602	1.2882	0.6623	1.2818	0.6635	1.2764	0.6631
Image c	2.2289	6.9538	2.2382	6.9385	2.2264	6.9360	2.2257	6.9668

Table 3.35 RMS errors in image coordinates and world coordinates from Figure 3.12.

RMS	8bit TIFF from Small RAW (pixel in image sensor)		8bit TIFF from Medium RAW (pixel in image sensor)		8bit TIFF from Small RAW (mm)		8bit TIFF from Medium RAW (mm)	
	u axis	v axis	u axis	v axis	x axis	y axis	x axis	y axis
Image a	1.6068	0.8018	1.4992	0.7749	0.7775	2.2166	0.7757	2.2193
Image b	1.1960	0.6382	1.1412	0.6467	1.3060	0.6719	1.2841	0.6615
Image c	1.0854	0.6672	1.0773	0.6856	2.2335	6.9868	2.2343	6.9741

From Tables 3.33 to 3.35, one can see that the use of higher quality JPEG images results in better accuracy. Since a lot of commercial cameras only support JPEG format, we recommend the use of high quality (i.e., high pixel density) JPEG images. However, because processing of higher quality JPEG images requires more computational time, we will use only Medium JPEG images for later comparisons in order to demonstrate theories and concepts without too much computational efforts. More experiments show

that if the checker board covers more area of the sensor plane, the errors in image coordinates will reduce, but the errors in world coordinates will increase.

3.3 Four-Point Calibration Method

As shown by (3.1.15), at least 5 object points on the model plane are needed in order to have a unique solution for the homography $\{H\}$. That means a checker board should have at least 5 feature points. For the kind of checker board we used, at least two black squares and two white squares are required, as shown in Figure 3.20.

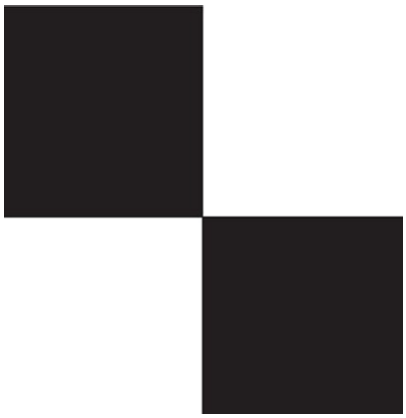


Figure 3.20 A smallest checker board for camera calibration.

In this section, we present a method that can create an extra virtual feature point from four corner points of any rectangle to add an orthogonal constraint in order to solve for $\{H\}$. Then the requirement is reduced to 4 feature points of an arbitrary rectangle without any black and white grids. However, because of fewer points, it is difficult to obtain an accurate lens distortion model. Although one can increase the total number of corner points by using more than 3 calibration images to obtain a lens distortion model, this method is only recommended for obtaining intrinsic and extrinsic parameters, not the

lens distortion model. Since rectangles often exist in indoor and outdoor environments, a camera with known lens distortion models can be easily calibrated using a rectangle with four clear corner points and known lengths.

3.3.1 Image Point of a Rectangle's Center

An independent virtual image point can be created for the center of the rectangular model plane. For a rectangle with four corner points at $\{p_1\} = \{x_0, y_0, z_0\}^T$ with $z_0 = 1$, $\{p_2\} = \{x_0 + \Delta x, y_0, z_0\}^T$, $\{p_3\} = \{x_0 + \Delta x, y_0 + \Delta y, z_0\}^T$, and $\{p_4\} = \{x_0, y_0 + \Delta y, z_0\}$, its center $\{p_c\} = \{x_c, y_c, z_0\}^T$ can be found as

$$\{p_c\} = \{x_0 + \Delta x/2, y_0 + \Delta y/2, z_0\}^T = \beta\{p\} \tag{3.3.1}$$

$$\mathbf{p} \equiv (\mathbf{p}_1 \times \mathbf{p}_3) \times (\mathbf{p}_2 \times \mathbf{p}_4) \tag{3.3.2}$$

where bold face letters represent vectors, e.g., $\mathbf{p}_1 = x_0\mathbf{i}_x + y_0\mathbf{i}_y + z_0\mathbf{i}_z$ and $\mathbf{p} = \bar{x}\mathbf{i}_x + \bar{y}\mathbf{i}_y + \bar{z}\mathbf{i}_z$. It follows from (3.3.1) that $\beta \equiv z_0/\bar{z}$ ($\bar{z}$ is the z component of $\mathbf{p}$), and hence $\mathbf{p}$ goes through the center of the rectangle and β can be used to find the intersection point of $\mathbf{p}$ and the model plane.

It follows from (2.3.20) that the projected image coordinates $\{P_i\}$ of the four corners can be described as

$$\{P_i\} \equiv \begin{Bmatrix} u_i \\ v_i \\ 1 \end{Bmatrix} = \frac{1}{\zeta_i}[H]\{p_i\}, \quad i = 1, 2, 3, 4 \tag{3.3.3}$$

where the 3×3 homography matrix $[H]$ is defined in (2.3.20) and it is valid for any point on the image plane. From (3.3.1)-(3.3.3), we obtain

$$\{P_c\} \equiv \begin{Bmatrix} u_c \\ v_c \\ 1 \end{Bmatrix} = \frac{1}{\zeta_c}[H]\{p_c\} = \alpha[H]\{p\}, \quad \alpha \equiv \frac{\beta}{\zeta_c} \tag{3.3.4}$$

It follows from the properties of cross products of vectors (as shown in Appendix C) that

$$([M]\{a\}) \times ([M]\{b\}) = |[M]|[M]^{-T}(\{a\} \times \{b\}) \tag{3.3.5}$$

where $[M]$ is an arbitrary 3×3 matrix, and $\{a\}$ and $\{b\}$ are arbitrary 3×1 vectors. Hence, the relationship between $\mathbf{P}_c (= u_c\mathbf{i}_1 + v_c\mathbf{i}_2 + \mathbf{i}_3)$ and $\mathbf{P}_i (= u_i\mathbf{i}_1 + v_i\mathbf{i}_2 + \mathbf{i}_3)$ can be obtained as

$$\begin{aligned}
&(\mathbf{P}_1 \times \mathbf{P}_3) \times (\mathbf{P}_2 \times \mathbf{P}_4) \\
&\quad = \left(\frac{1}{\zeta_1}[H]\{p_1\} \times \frac{1}{\zeta_3}[H]\{p_3\}\right) \times \left(\frac{1}{\zeta_2}[H]\{p_2\} \times \frac{1}{\zeta_4}[H]\{p_4\}\right) \\
&\quad = k([H]\{p_1\} \times [H]\{p_3\}) \times ([H]\{p_2\} \times [H]\{p_4\}) \qquad \left(k \equiv \frac{1}{\zeta_1\zeta_2\zeta_3\zeta_4}\right) \\
&\quad = k(|[H]|[H]^{-T}(\mathbf{p}_1 \times \mathbf{p}_3)) \times (|[H]|[H]^{-T}(\mathbf{p}_2 \times \mathbf{p}_4)) \\
&\quad = k|[H]|^2([H]^{-T}(\mathbf{p}_1 \times \mathbf{p}_3)) \times ([H]^{-T}(\mathbf{p}_2 \times \mathbf{p}_4)) \\
&\quad = k|[H]|^2|[H]^{-T}|[H](\mathbf{p}_1 \times \mathbf{p}_3) \times (\mathbf{p}_2 \times \mathbf{p}_4) \\
&\quad = k|[H]|[H](\mathbf{p}_1 \times \mathbf{p}_3) \times (\mathbf{p}_2 \times \mathbf{p}_4)
\end{aligned} \tag{3.3.6}$$

It follows from (3.3.4) and (3.3.6) that the image coordinates of the rectangle's center can be found without any information outside the image as

$$\{P_c\} = \mu\{P\}, \quad \mathbf{P} \equiv (\mathbf{P}_1 \times \mathbf{P}_3) \times (\mathbf{P}_2 \times \mathbf{P}_4), \quad \mu \equiv \frac{\alpha}{k|H|} \tag{3.3.7}$$

where

$$\{P_c\} \equiv \begin{Bmatrix} u_c \\ v_c \\ 1 \end{Bmatrix}, \ \{P_i\} \equiv \begin{Bmatrix} u_i \\ v_i \\ 1 \end{Bmatrix} \tag{3.3.8}$$

In other words, **P** goes through the image point of the rectangle's center and μ is the scale to find the point $\{P_c\}$ as the intersection point. Hence, only four corner points are needed because (3.3.1) and (3.3.7) provide the world and image coordinates of the fifth point to agree the orthogonal constraint.

3.3.2 Comparison of Different Points Result

To verify the accuracy of the Four-Point calibration algorithm, we consider a camera with the parameters shown in (3.1.45a) and (3.1.45b). A smallest element for five- and four-point calibrations are picked and shown in Figure 3.21.

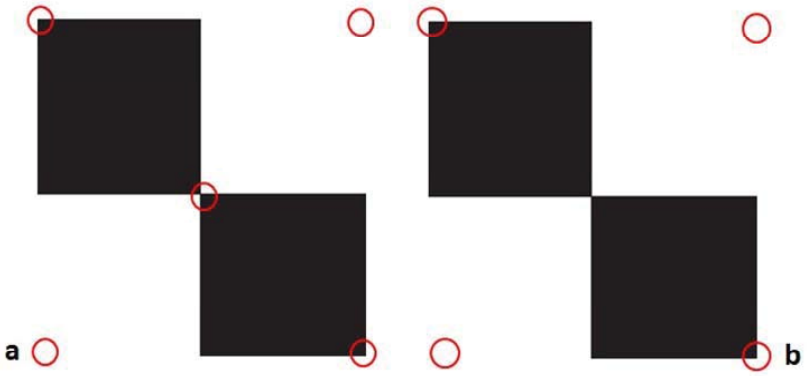


Figure 3.21 The smallest element for calibration using: (a) 5 corner points, and (b) 4 corner points.

Tables 3.36 to 3.43 show the results from camera calibrations using 140 points, 5 points, and 4 points with no noise (i.e., $\sigma = 0$). In simulations, the optimized function value increases with the number of corner points. However, more corner points result in better calibration results, and less points may result in ill-conditioned problems.

Table 3.36 Minimization of the objective function F_0 using different numbers of corner points.

Objective function	140 points	5 points	4 points
F_0	1.3372×10^{-5}	2.3845×10^{-8}	5.5516×10^{-10}
$F_{0\min}$	2.6353×10^{-7}	1.5716×10^{-9}	2.0062×10^{-10}

Table 3.37 The estimated forward lens distortion models using different numbers of points.

Forward model	$\hat{K}_1$	$\hat{K}_2$	$\hat{P}_1$	$\hat{P}_2$	$\hat{S}_1$	$\hat{S}_2$
Assumed	-0.23	0.2	0.0	0.0	0.0	0.0
140 Points	-0.23	0.2	-4.7909×10^{-7}	-1.9462×10^{-8}	5.41×10^{-7}	-4.9183×10^{-7}
5 Points	-0.23	0.20001	4.502×10^{-8}	-1.3772×10^{-7}	-6.6416×10^{-8}	2.4323×10^{-7}
4 Points	-0.23	0.2	-1.9854×10^{-8}	-1.1041×10^{-8}	-2.0811×10^{-8}	-2.9475×10^{-8}

Table 3.38 The estimated backward lens distortion models using different numbers of points.

Forward model	K_1	K_2	P_1	P_2	S_1	S_2
Assumed	-	-	-	-	-	-
140 Points	0.23124	-0.077803	-5.3734×10^{-7}	-5.292×10^{-7}	6.6182×10^{-7}	9.5093×10^{-7}
5 Points	0.23254	-0.079723	9.022×10^{-5}	-1.0265×10^{-4}	-9.0192×10^{-5}	1.0268×10^{-4}
4 Points	0.23254	-0.079725	9.0284×10^{-5}	-1.0269×10^{-4}	-9.0272×10^{-5}	1.0269×10^{-4}

Table 3.39 The estimated intrinsic parameters using different numbers of points.

Parameters (pixel)	α	β	γ	u_0	v_0
Assumed	1250	900	1.0908	255	255
140 points	1250	900	1.0908	255	255
5 points	1250	900	1.0908	255	255
4 points	1250	900	1.0908	255	255

Table 3.40 The estimated camera location vectors using different numbers of points.

Translation vectors (mm)	$\{\tau\}_1^T$	$\{\tau\}_2^T$	$\{\tau\}_3^T$
Assumed	{-90 105 500}	{-90 105 510}	{-105 105 525}
140 points	{-90 105 500}	{-90 105 510}	{-105 105 525}
5 points	{-90 105 500}	{-90 105 510}	{-105 105 525}
4 points	{-90 105 500}	{-90 105 510}	{-105 105 525}

Table 3.41 The estimated [*T*] of the first camera orientation using different numbers of points.

Rotational matrix	$[T]_1$
Assumed	$\begin{bmatrix} 0.99528 & -0.092332 & 0.029809 \\ -0.076723 & -0.93703 & -0.34072 \\ 0.059391 & 0.33682 & -0.93969 \end{bmatrix}$
140 points	$\begin{bmatrix} 0.99528 & -0.092332 & 0.029809 \\ -0.076723 & -0.93703 & -0.34072 \\ 0.059391 & 0.33682 & -0.93969 \end{bmatrix}$
5 points	$\begin{bmatrix} 0.99528 & -0.092332 & 0.029809 \\ -0.076723 & -0.93703 & -0.34072 \\ 0.059391 & 0.33682 & -0.93969 \end{bmatrix}$
4 points	$\begin{bmatrix} 0.99528 & -0.092332 & 0.029809 \\ -0.076723 & -0.93703 & -0.34072 \\ 0.059391 & 0.33682 & -0.93969 \end{bmatrix}$

Table 3.42 The estimated [*T*] of the second camera orientation using different numbers of points.

Rotational matrix	$[T]_2$
Assumed	$\begin{bmatrix} 0.99997 & -3.3039\times10^{-4} & -7.5961\times10^{-3} \\ 3.3039\times10^{-4} & -0.99622 & 0.086824 \\ -7.5961\times10^{-3} & -0.086824 & -0.99619 \end{bmatrix}$
140 points	$\begin{bmatrix} 0.99997 & -4.0162\times10^{-4} & -7.6775\times10^{-3} \\ 2.6487\times10^{-4} & -0.99624 & 0.086613 \\ -7.6835\times10^{-3} & -0.086613 & -0.99621 \end{bmatrix}$
5 points	$\begin{bmatrix} 0.99997 & -3.3039\times10^{-4} & -7.5962\times10^{-3} \\ 3.3039\times10^{-4} & -0.99622 & 0.086824 \\ -7.5962\times10^{-3} & -0.086824 & -0.99619 \end{bmatrix}$
4 points	$\begin{bmatrix} 0.99997 & -3.3034\times10^{-4} & -7.5961\times10^{-3} \\ 3.3034\times10^{-4} & -0.99622 & 0.086824 \\ -7.5961\times10^{-3} & -0.086824 & -0.99619 \end{bmatrix}$

Table 3.43 The estimated [*T*] of the third camera orientation using different numbers of points.

Rotational matrix	$[T]_3$
Assumed	$\begin{bmatrix} 0.9446 & 0.22189 & -0.24184 \\ 0.28014 & -0.929 & 0.24184 \\ -0.17101 & -0.2962 & -0.93969 \end{bmatrix}$
140 points	$\begin{bmatrix} 0.94462 & 0.22182 & -0.24184 \\ 0.28013 & -0.92892 & 0.24216 \\ -0.17093 & -0.2965 & -0.93961 \end{bmatrix}$
5 points	$\begin{bmatrix} 0.9446 & 0.22189 & -0.24184 \\ 0.28014 & -0.929 & 0.24184 \\ -0.17101 & -0.2962 & -0.93969 \end{bmatrix}$
4 points	$\begin{bmatrix} 0.9446 & 0.22189 & -0.24184 \\ 0.28014 & -0.929 & 0.24184 \\ -0.17101 & -0.2962 & -0.93969 \end{bmatrix}$

Using (3.1.5a)-(3.1.5g) and Tables 3.41-3.43 we obtain the three Euler angles to be

$$\text{140 points: } \begin{Bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{Bmatrix} = \begin{Bmatrix} 5.0000^\circ \\ 159.9996^\circ \\ 10.0001^\circ \end{Bmatrix}, \begin{Bmatrix} 5.0004^\circ \\ 185.0031^\circ \\ 5.0004^\circ \end{Bmatrix}, \begin{Bmatrix} 45^\circ \\ 200.0004^\circ \\ 29.9998^\circ \end{Bmatrix}$$

$$\text{5 points: } \begin{Bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{Bmatrix} = \begin{Bmatrix} 5.0000^\circ \\ 159.9996^\circ \\ 10.0001^\circ \end{Bmatrix}, \begin{Bmatrix} 5.0001^\circ \\ 185.0031^\circ \\ 5.0001^\circ \end{Bmatrix}, \begin{Bmatrix} 45^\circ \\ 200.0004^\circ \\ 29.9998^\circ \end{Bmatrix}$$

$$\text{4 points: } \begin{Bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{Bmatrix} = \begin{Bmatrix} 5.0000^\circ \\ 159.9996^\circ \\ 10.0001^\circ \end{Bmatrix}, \begin{Bmatrix} 5.0000^\circ \\ 185.0031^\circ \\ 5.0000^\circ \end{Bmatrix}, \begin{Bmatrix} 45^\circ \\ 200.0004^\circ \\ 29.9998^\circ \end{Bmatrix}$$

Tables 3.44-3.51 show the calibration results using 140 points, 5 points, and 4 points with a random noise with $\sigma = 0.1$ being added to the distorted image coordinates $(\tilde{u}, \tilde{v})$. Since the influence of a random noise vector of four or five entries is very

different from that of a random noise vector of 140 entries. However, results show that the use of four corner points works fine.

Table 3.44 Minimization of the objective function F_0 using different numbers of corner points.

Objective function	140 points	5 points	4 points
F_0	7.5104	0.23359	0.23359
$F_{0\min}$	7.3706	0.032845	0.032834

Table 3.45 The estimated forward lens distortion models using different numbers of points.

Forward model	$\hat{K}_1$	$\hat{K}_2$	$\hat{P}_1$	$\hat{P}_2$	$\hat{S}_1$	$\hat{S}_2$
Assumed	-0.23	0.2	0.0	0.0	0.0	0.0
140 Points	-0.23144	0.21168	7.0611×10^{-5}	5.5258×10^{-5}	-2.4598×10^{-4}	-1.406×10^{-5}
5 Points	-0.44122	0.65192	2.3601×10^{-3}	0.043285	-5.2232×10^{-3}	-0.048577
4 Points	-0.34325	0.32213	2.3148×10^{-3}	0.030299	2.853×10^{-3}	-0.033256

Table 3.46 The estimated backward lens distortion models using different number of points.

Forward model	K_1	K_2	P_1	P_2	S_1	S_2
Assumed	-	-	-	-	-	-
140 Points	0.2331	-0.095551	-5.3067×10^{-5}	-3.3349×10^{-5}	2.2464×10^{-4}	-5.2765×10^{-5}
5 Points	0.42977	-0.48014	-3.3308×10^{-3}	-0.043184	6.2514×10^{-3}	0.048629
4 Points	0.32769	-0.12036	-3.4212×10^{-3}	-0.029756	-1.8402×10^{-3}	0.0327

Table 3.47 The estimated intrinsic parameters using different numbers of points.

Parameters (pixel)	α	β	γ	u_0	v_0
Assumed	1250	900	1.0908	255	255
140 points	1251.2	900.94	0.95257	254.91	254.81
5 points	1433.4	1034.3	-9.2651	200.86	216.24
4 points	1434.7	1034.2	-8.4816	201.6	216.1

Table 3.48 The estimated camera location vectors using different numbers of points.

Translation vectors (mm)	$\{\tau\}_1^T$	$\{\tau\}_2^T$	$\{\tau\}_3^T$
Assumed	{-90 105 500}	{-90 105 510}	{-105 105 525}
140 points	{-89.956 105.11 500.51}	{-89.958 105.11 510.47}	{-104.96 105.09 525.47}
5 points	{-66.798 126.42 575.34}	{-66.442 127.12 587.51}	{-80.856 127.97 605.67}
4 points	{-67.406 126.36 574.54}	{-67.114 127.18 587.24}	{-81.572 128.04 605.67}

Table 3.49 The estimated [*T*] of the first camera orientation using different numbers of points.

Rotational matrix	$[T]_1$
Assumed	$\begin{bmatrix} 0.99528 & -0.092332 & 0.029809 \\ -0.076723 & -0.93703 & -0.34072 \\ 0.059391 & 0.33682 & -0.93969 \end{bmatrix}$
140 points	$\begin{bmatrix} 0.99528 & -0.092389 & 0.029574 \\ -0.076843 & -0.93694 & -0.34093 \\ 0.059207 & 0.33705 & -0.93962 \end{bmatrix}$
5 points	$\begin{bmatrix} 0.99585 & -0.088465 & -0.02122 \\ -0.089821 & -0.91909 & -0.38368 \\ 0.01444 & 0.38399 & -0.92322 \end{bmatrix}$
4 points	$\begin{bmatrix} 0.99609 & -0.085081 & -0.023628 \\ -0.087566 & -0.91731 & -0.38843 \\ 0.011374 & 0.38898 & -0.92118 \end{bmatrix}$

Table 3.50 The estimated [*T*] of the second camera orientation using different numbers of points.

Rotational matrix	$[T]_2$
Assumed	$\begin{bmatrix} 0.99997 & -3.3039\times10^{-4} & -7.5961\times10^{-3} \\ 3.3039\times10^{-4} & -0.99622 & 0.086824 \\ -7.5961\times10^{-3} & -0.086824 & -0.99619 \end{bmatrix}$
140 points	$\begin{bmatrix} 0.99997 & -4.0162\times10^{-4} & -7.6775\times10^{-3} \\ 2.6487\times10^{-4} & -0.99624 & 0.086613 \\ -7.6835\times10^{-3} & -0.086613 & -0.99621 \end{bmatrix}$
5 points	$\begin{bmatrix} 0.99907 & -0.018418 & -0.038986 \\ -0.015141 & -0.99646 & 0.082727 \\ -0.040372 & -0.082059 & -0.99581 \end{bmatrix}$
4 points	$\begin{bmatrix} 0.99932 & -0.014263 & -0.033867 \\ -0.011456 & -0.99659 & 0.081669 \\ -0.034916 & -0.081226 & -0.99608 \end{bmatrix}$

Table 3.51 The estimated [*T*] of the third camera orientation using different numbers of points.

Rotational matrix	$[T]_3$
Assumed	$\begin{bmatrix} 0.9446 & 0.22189 & -0.24184 \\ 0.28014 & -0.929 & 0.24184 \\ -0.17101 & -0.2962 & -0.93969 \end{bmatrix}$
140 points	$\begin{bmatrix} 0.94462 & 0.22182 & -0.24184 \\ 0.28013 & -0.92892 & 0.24216 \\ -0.17093 & -0.2965 & -0.93961 \end{bmatrix}$
5 points	$\begin{bmatrix} 0.94118 & 0.19211 & -0.27797 \\ 0.26192 & -0.93451 & 0.24101 \\ -0.21347 & -0.29964 & -0.92987 \end{bmatrix}$
4 points	$\begin{bmatrix} 0.94058 & 0.19629 & -0.2771 \\ 0.26526 & -0.93418 & 0.23864 \\ -0.21202 & -0.29796 & -0.93073 \end{bmatrix}$

Using (3.1.5a)-(3.1.5g) and Tables 3.49-3.51 we obtain the three Euler angles to be

$$\text{140 points: } \begin{Bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{Bmatrix} = \begin{Bmatrix} 4.9711^\circ \\ 159.9862^\circ \\ 9.9775^\circ \end{Bmatrix}, \begin{Bmatrix} 5.0105^\circ \\ 184.9833^\circ \\ 5.0141^\circ \end{Bmatrix}, \begin{Bmatrix} 44.9538^\circ \\ 200.0088^\circ \\ 29.9557^\circ \end{Bmatrix}$$

$$\text{5 points: } \begin{Bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{Bmatrix} = \begin{Bmatrix} 356.8344^\circ \\ 157.4015^\circ \\ 2.1536^\circ \end{Bmatrix}, \begin{Bmatrix} 25.2327^\circ \\ 185.2468^\circ \\ 26.1966^\circ \end{Bmatrix}, \begin{Bmatrix} 49.0735^\circ \\ 201.5854^\circ \\ 35.4669^\circ \end{Bmatrix}$$

$$\text{4 points: } \begin{Bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{Bmatrix} = \begin{Bmatrix} 356.5190^\circ \\ 157.0992^\circ \\ 1.6749^\circ \end{Bmatrix}, \begin{Bmatrix} 22.5231^\circ \\ 185.0748^\circ \\ 23.2610^\circ \end{Bmatrix}, \begin{Bmatrix} 49.2648^\circ \\ 201.4511^\circ \\ 35.4346^\circ \end{Bmatrix}$$

For experimental comparison, three images of the checker board taken by a fixed Canon EOS-7D camera and shown in Figure 3.22 are to be used. These images are captured with ISO speed at 4000, *f*-number at *f*/8, shutter speed at 1/100 second, and a

focal length of 50mm. The image format is the medium fine JPEG, which has 3456×2304 pixels. The two red rectangles select 48 (8×6) feature points and 35 (7×5) feature points, the nine circles show 9 chosen feature points, the five yellow crosses indicate 5 feature points, and the 5 feature points without the center point reduce to 4 feature points.

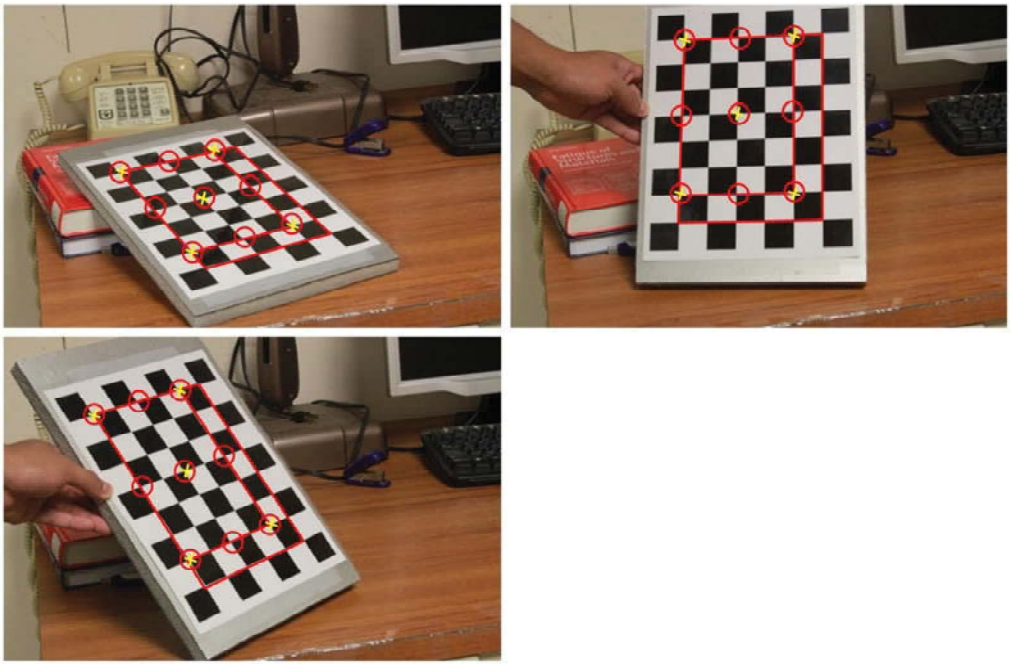


Figure 3.22 Three images used for comparing results using different numbers of corner points.

As shown before, it is difficult to obtain accurate lens distortion models when only a few corner points are used. So that we only compare estimated intrinsic and extrinsic parameters here. Tables 3.52-3.54 compare the intrinsic parameters and camera location vectors estimated using different numbers of feature points.

Table 3.52 The estimated intrinsic parameters using different numbers of feature points (in pixel).

Parameters (pixel)	α	β	γ	u_0	v_0
48 points	7525.4	7653.4	-10	1738	1160
35 points	7462.6	7609.9	-10	1738	1162
9 points	7492.5	7630.1	4.6702	1738	1161
5 points	7683.5	7741.8	10	1738	1162
4 points	7717.9	7563.4	-10	1738	1162

Table 3.53 The estimated intrinsic parameters using different numbers of feature points (in mm).

Parameters (mm)	α	β	γ	u_0	v_0
48 points	48.5580	49.4946	-0.0645	11.2145	7.5017
35 points	48.1528	49.2133	-0.0645	11.2145	7.5147
9 points	48.3457	49.3440	0.0301	11.2145	7.5082
5 points	49.5781	50.0663	0.0645	11.2145	7.5147
4 points	49.8001	48.9126	-0.0645	11.2145	7.5147

Table 3.54 The estimated camera location vectors using different numbers of feature points.

Translation vectors (mm)	$\{\tau\}_1^T$	$\{\tau\}_2^T$	$\{\tau\}_3^T$
48 points	{-114.04 -105.19 1436.2}	{66.14 -316.65 1229.5}	{-160.09 -274.25 1232.1}
35 points	{-114.06 -105.15 1427}	{66.39 -316.67 1222.5}	{-160.13 -274.21 1225.7}
9 points	{-113.67 -105.15 1430}	{66.871 -316.68 1224.3}	{-159.65 -274.48 1229.2}
5 points	{-113.25 -106.36 1458.1}	{67.866 -319.14 1240.3}	{-158.32 -276.74 1249.1}
4 points	{-112.8 -109.11 1452.2}	{66.998 -324.04 1211.8}	{-157.26 -279.91 1230}

Using (3.1.5a)-(3.1.5g) the three Euler angles of each of the three camera orientations are estimated as

$$\text{48 points: } \begin{Bmatrix}\theta_1\\ \theta_2\\ \theta_3\end{Bmatrix} = \begin{Bmatrix}3.5430^\circ\\ 119.3735^\circ\\ 31.7486^\circ\end{Bmatrix}, \begin{Bmatrix}340.9796^\circ\\ 160.6220^\circ\\ 341.6415^\circ\end{Bmatrix}, \begin{Bmatrix}44.8118^\circ\\ 133.8410^\circ\\ 73.8112^\circ\end{Bmatrix}$$

$$\text{35 points: } \begin{Bmatrix}\theta_1\\ \theta_2\\ \theta_3\end{Bmatrix} = \begin{Bmatrix}3.7948^\circ\\ 119.2112^\circ\\ 31.8680^\circ\end{Bmatrix}, \begin{Bmatrix}341.2018^\circ\\ 160.5651^\circ\\ 341.8052^\circ\end{Bmatrix}, \begin{Bmatrix}44.7170^\circ\\ 133.6919^\circ\\ 73.6619^\circ\end{Bmatrix}$$

$$\text{9 points: } \begin{Bmatrix}\theta_1\\ \theta_2\\ \theta_3\end{Bmatrix} = \begin{Bmatrix}3.6814^\circ\\ 119.2624^\circ\\ 31.7217^\circ\end{Bmatrix}, \begin{Bmatrix}340.3603^\circ\\ 160.5135^\circ\\ 340.9192^\circ\end{Bmatrix}, \begin{Bmatrix}44.7773^\circ\\ 133.9062^\circ\\ 73.6762^\circ\end{Bmatrix}$$

$$\text{5 points: } \begin{Bmatrix}\theta_1\\ \theta_2\\ \theta_3\end{Bmatrix} = \begin{Bmatrix}3.7073^\circ\\ 119.6150^\circ\\ 31.6912^\circ\end{Bmatrix}, \begin{Bmatrix}336.2872^\circ\\ 161.4543^\circ\\ 336.7172^\circ\end{Bmatrix}, \begin{Bmatrix}45.9544^\circ\\ 134.0153^\circ\\ 74.7626^\circ\end{Bmatrix}$$

$$\text{4 points: } \begin{Bmatrix}\theta_1\\ \theta_2\\ \theta_3\end{Bmatrix} = \begin{Bmatrix}3.4897^\circ\\ 120.6745^\circ\\ 31.5944^\circ\end{Bmatrix}, \begin{Bmatrix}324.9076^\circ\\ 163.2637^\circ\\ 325.4678^\circ\end{Bmatrix}, \begin{Bmatrix}48.0807^\circ\\ 134.3276^\circ\\ 77.1012^\circ\end{Bmatrix}$$

These results show that the intrinsic and extrinsic parameters estimated using the four-point calibration method agrees well with those using more feature points. Hence, the four-point method can be used for fast and convenient camera calibration if the camera's forward and backward lens distortion models are known.

CHAPTER 4

STATIC AND DYNAMIC MEASUREMENT

After camera calibration, a system of two or more cameras can be used for noncontact static/dynamic measurements. In this chapter we use two Canon EOS-7D DSLR cameras as an example system to demonstrate how to experimentally obtain 3D coordinates of points on a static/dynamic structure. Two methods of creating distinctive object points on a structure under measurement are examined. In one method, we adhere flat circular retro-reflective markers on the object, which is a time-consuming and burdensome task. For the other method, we use a holographic laser pointer to project a grid of laser points on the object, which is easy and fast for preparation. However, because the projected laser points do not move with the object, it can only be used for static measurements.

4.1 Measurement Theory

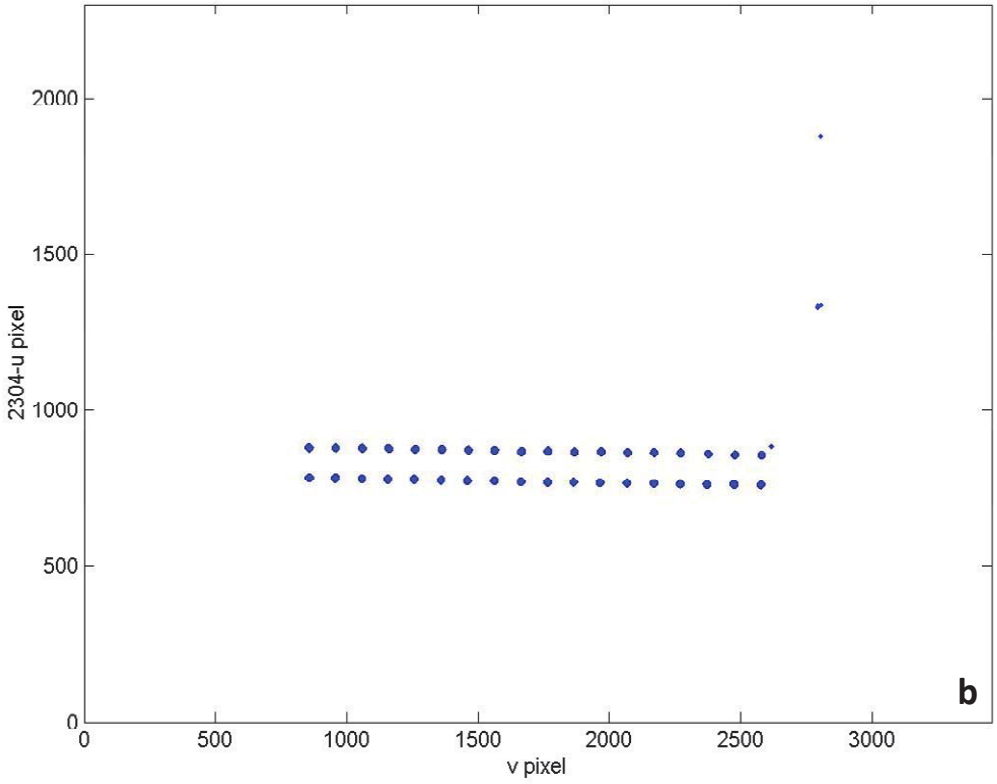
4.1.1 Use of Circular Retro-Reflective Markers

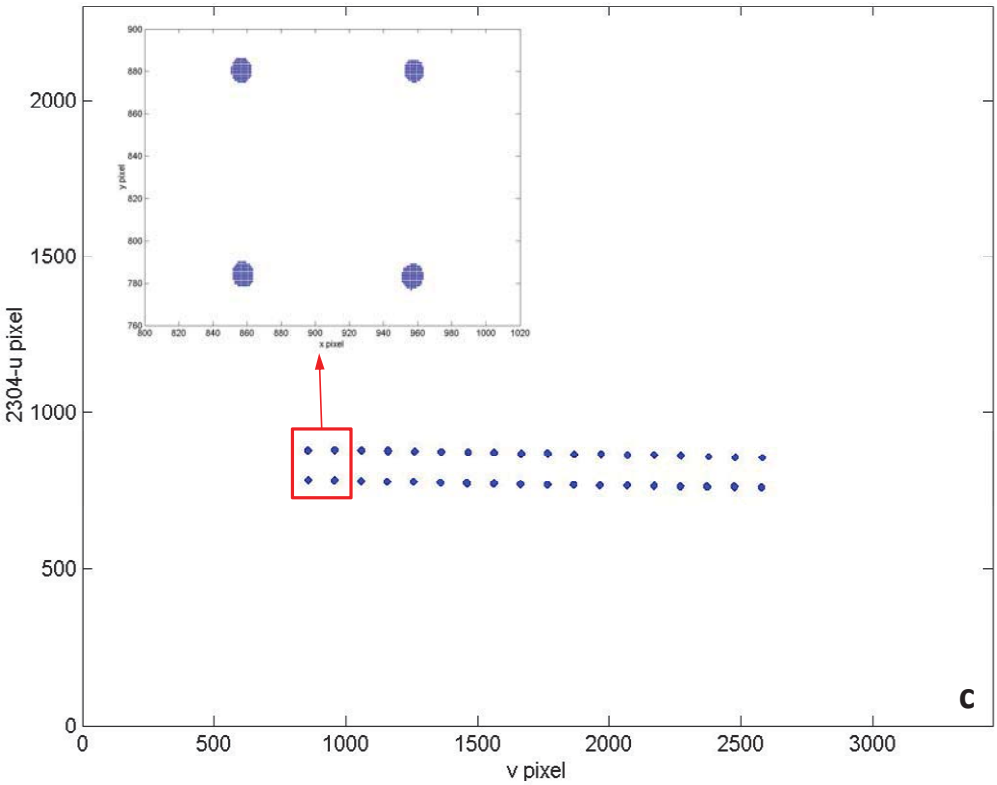
When the structure under measurement is in a regular/dark environment, it is better to shine a bright light on the structure to make the retro-reflective makers have high grey value circular/elliptical image areas on the image or video frame. However, flashlight is not recommended because it often causes the whole image too bright. The center of each of these circular/elliptical areas represents one interest point and can be located by averaging (with data smoothing or shape fitting) the pixel locations of the whole circular/elliptical area. After the center of a marker is identified on each of the images taken at a sequence of time instants by two or more cameras, the time-varying 3D

coordinates and the motion path of this marker can be traced by triangulation techniques, as shown in Chapter 3. However, some marker images may intersect or overlap, and the challenge is how to trace each of these markers from one image to another. With comprehensive image analysis and tracking of all of these interest points, the time-varying 3D configurations and animation of the structure can be obtained.

The first step of image processing is to use (2.3.3) to calculate the grey-value distribution of an image. Image noise can be created by the sensor and circuitry of a digital camera, dust in the air, the high frequency movement of the object, and some unexpected reflective objects, and Figure 4.1a shows an example. Similar to noise reduction in the corner detection process shown in Chapter 2, a 21×21 Laplacian of Gaussian (*LoG*) filter is applied here to reduce noise. The *LoG* filter can be approximated by the mask shown in (2.3.13) or (2.3.14). After noise reduction, each marker image takes up a group of pixels in the image. In order to find a unique position for each maker, a method to converge the group of pixels into one representative point needs to be developed, and the marker center would be the best choice.


a


2000
1500
1000
500
0
2304-u pixel
0
500
1000
1500
2000
2500
3000
v pixel
b


2000
1500
1000
500
0
2304-u pixel
0
500
1000
1500
2000
2500
3000
v pixel
c
y pixel
x pixel

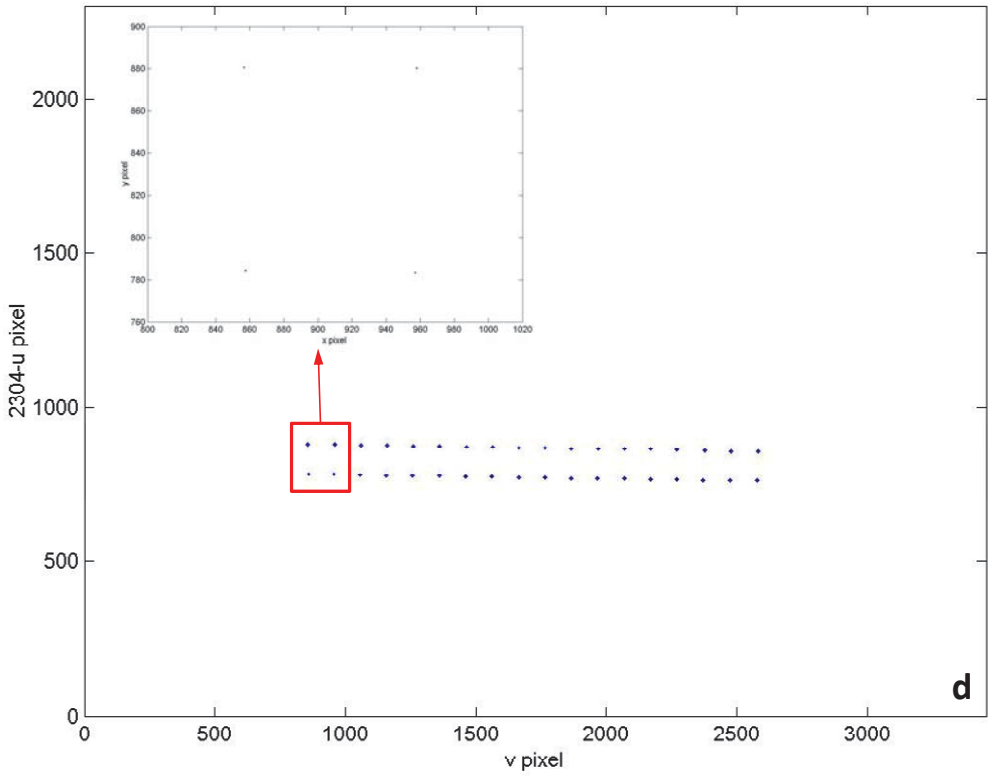


Figure 4.1 a) An original image showing two lines of markers (18 markers in each line, marker diameter being 3mm) on a beam, b) detection without noise filtering, c) interest points after noise filtering, and d) the identified 36 marker centers.

Interest points can be extracted by processing the distribution of grey values. Firstly, the grey value of each pixel is compared with a threshold named the first threshold, which is chosen to be the averaged grey value of all pixels in the image plus a chosen value (e.g., 160 for Figure 4.1a). If the grey value is larger than the first threshold, it can be considered as an interest point candidate. Subsequently, an LoG mask is used to smooth the grey value. If the summation of weighted grey values of all pixels inside the LoG mask area is larger than a second threshold (e.g., the summation of all grey values in the LoG mask area times the first threshold), this point is a real interest point (see Figure 4.1c). Otherwise it can be consider as a noise point. Since multiplication is only

performed for a small *LoG* window for each interest point candidate, this algorithm is efficient.

If noise filtering is not done, some unexpected noises may appear (see Figure 4.1b). There are three groups of noise in Figure 4.1b: the one close to the beam is from the screw fixing the beam, the other two are from some reflective objects in the background. Although we will use a blackboard or black cloth as the background in static/dynamic measurement, the noise filtering is still recommended.

In order to represent each marker by its center, the area of each marker image needs to be determined first. Here we present two methods to find the marker area. One is an area search method, and the other is a line search method.

For the area search method, the average area of all marker images needs to be roughly estimated first. Firstly, we search the whole image along the *u* direction to find the internal interest pixels and the total number of pixel lines of each marker area. If the four adjacent pixels of an interest pixel are all interest pixels (i.e., pixel index=1), it is an internal interest pixel and the line length counter W_t is increased by one. If some of the four adjacent pixels are not interest pixels (i.e., pixel index=0) and W_t is not zero, this interest pixel is considered as a line end and then the line counter *t* is increased by 1, the total internal pixel counter *W* is increased by W_t, and W_t is set back to zero. After the whole image is searched, the average line length along the *u* direction is W/t, which is about the average radius of all marker images times $\pi/2$. Hence, the average diameter is $(4W)/(\pi t)$. Since the pixels on marker edges are not counted, the window length along the *u* direction, W_{in1}, is estimated as

$$W_{in1} = \text{round}\left(4 \times \frac{W}{\pi t} + 5\right) \tag{4.1.1}$$

where W_{in1} is in pixel. The window length along the v direction, W_{in2}, can be obtained using the same approach. Then the $W_{in1} \times W_{in2}$ rectangular window can be used to search and enclose each marker image area one by one by performing area search, as shown in Figure 4.2. If the sum of pixel indices of pixels on the window's four edges is not zero, it indicates that the marker image area intersects with the window, as shown in Figure 4.2a. If the sum of pixel indices of pixels on the window's four edges is zero, it indicates that the window encloses the whole marker image area, as shown in Figure 4.2b. Then the averaged coordinates $(\bar{u}, \bar{v})$ of all interest pixels in the window can be taken as the marker center. Of course, one can fit all the interest pixels in the window into an ellipse (i.e., the projection of a circular area) and then obtain a better estimation of the marker center location, but it requires much more computation. After the center of a marker image is obtained, we set this marker image's all pixel indices to zero. After all pixel indices are set to zero, the centers of all marker images are found, as shown in Figure 4.1d.

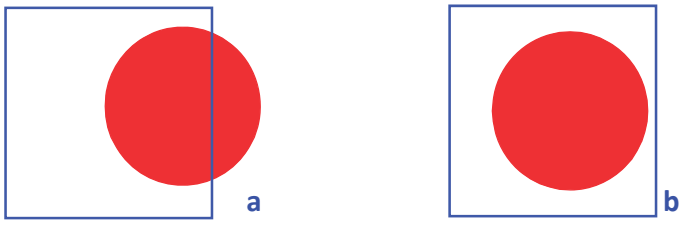


Figure 4.2 The window encloses the marker image area: a) no, and b) yes.

The line search method can be used to locate image areas of circular, elliptical, or even polygonal markers. In this method, marker image areas are located one by one by line searching, and it does not use the rectangular window shown in Figure 4.2a. The line search starts at an interest pixel (i.e., pixel index=1) and ends at a non-interest pixel. In order to stop the line search when a marker image area is completely located, we use $flag_1$ to indicate the number of interest pixels on the next pixel line, as shown in Figure 4.3a. If $flag_1 \neq 0$, the next pixel line needs to be searched for possible pixels belonging to the current marker image area. As shown in Figure 4.3b, if the pixel index of the pixel point on the next line and right next to the beginning pixel of the current line is one, search upward until a non-interest pixel is located and treated as the beginning pixel of the new line. If the pixel index of the pixel point on the next line and right next to the beginning pixel of the current line is zero, search downward until an interest pixel is located and treated as the beginning pixel of the new line. After the line search is stopped (i.e., $flag_1 = 0$), the averaged coordinates $(\overline{u}, \overline{v})$ of all interest pixels of this marker image area can be taken as the marker's center and then this marker image's all pixel indices are set to zero.

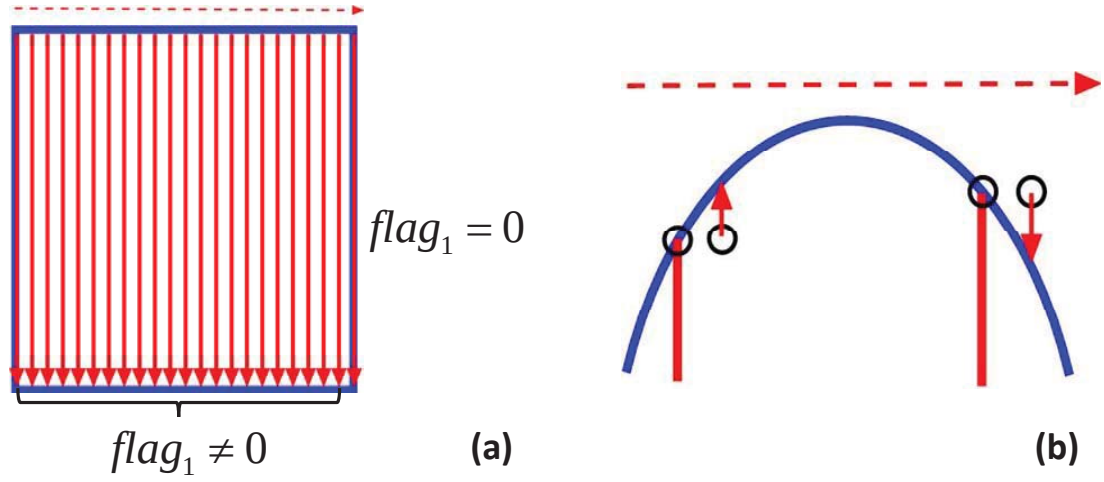


Figure 4.3 The line search method: a) condition to stop line search, and b) search for the beginning pixel of a new pixel line.

Because a marker image area may not be smooth, some special cases need to be considered. Figure 4.4 shows two special cases, where A is a searched area, B is an area to be searched, and C is a missed area. Note that the indices of interest pixels in A have been set to 0. In order to locate all areas of a marker image like that in Figure 4.4a, $flag_0$ is used to indicate the number of interest pixels on the previous pixel line. If $flag_0 \neq 0$, a reverse search to the left is needed. For the case shown in Figure 4.4b, if $flag_1$ of the current line is smaller than that of the previous line, continue search until C is covered. If both lines have about the same $flag_1$, there are no missed areas.

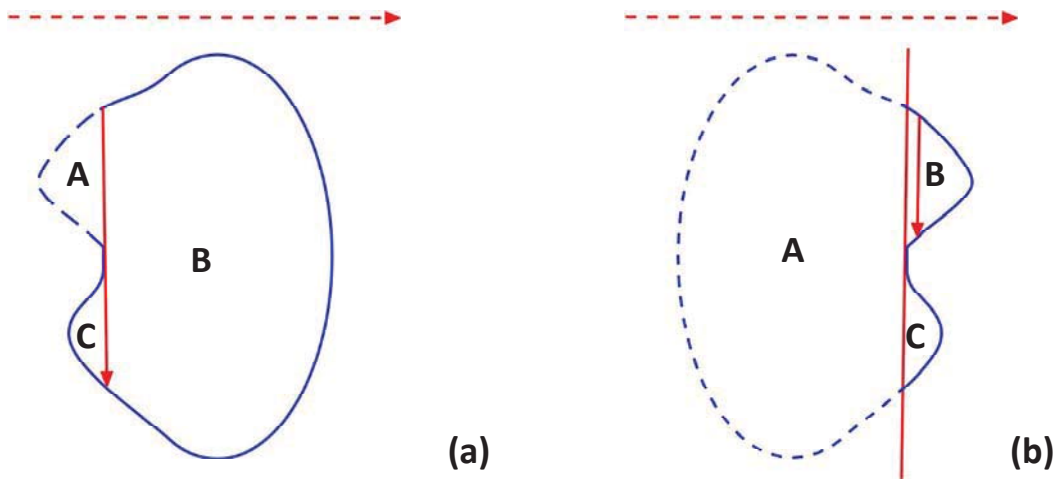


Figure 4.4 Special marker images for the line search method: a) missing areas before the searching line, and b) missing areas after the searching line.

Although some more complicated marker image areas may exist, they rarely occur for real markers. Even if such a marker image exists, the missing areas often have a small number of interest points.

In general, the area search method is faster than the line search method, but the area search method works only if each marker image area is separated from other image areas by 5 or more pixels. The area search method can avoid locating a large noisy area as a marker area, but it requires the size differences between marker image areas to be small. In addition, if a checker board with a white background is used for camera calibration, the corner points can be automatically located using this area search method (see Figure 2.6c). The line search method only requires each marker image area being separated from other image areas by one or more pixels, but it costs more computational time. Moreover, it cannot avoid locating a large noisy area as a marker area, but it can locate image areas of markers of arbitrary shapes and sizes. Since a structure under

measurement may range over a large depth of field of the camera, the captured marker image areas can be large (when markers are close to the camera) or small (when markers are far away). Hence, the line search method is very suitable for vibration testing of structures, especially large highly flexible structures [2].

4.1.2 Use of Holographic Laser Points

The use of a holographic laser pointer to project bright dots and/or patterned shapes on a structure under measurement can release the user of the proposed camera-based measurement system from the time-consuming work of sticking many retro-reflective markers on the structure. The patterns projected from a holographic laser pointer can be used not only for corner detection but also for more complicated feature detection. Figure 4.5 shows a holographic laser pointer with five (one on the pointer) grid heads for creating different dots and shapes. Figure 4.6 shows some projected dots and shapes using different grid heads. A complex distribution of dots and shapes can be used for advanced point and feature extraction to enable simultaneous tracking of a group of points in a series of consecutive images.

Figure 4.5 A holographic laser pointer with different grid heads.

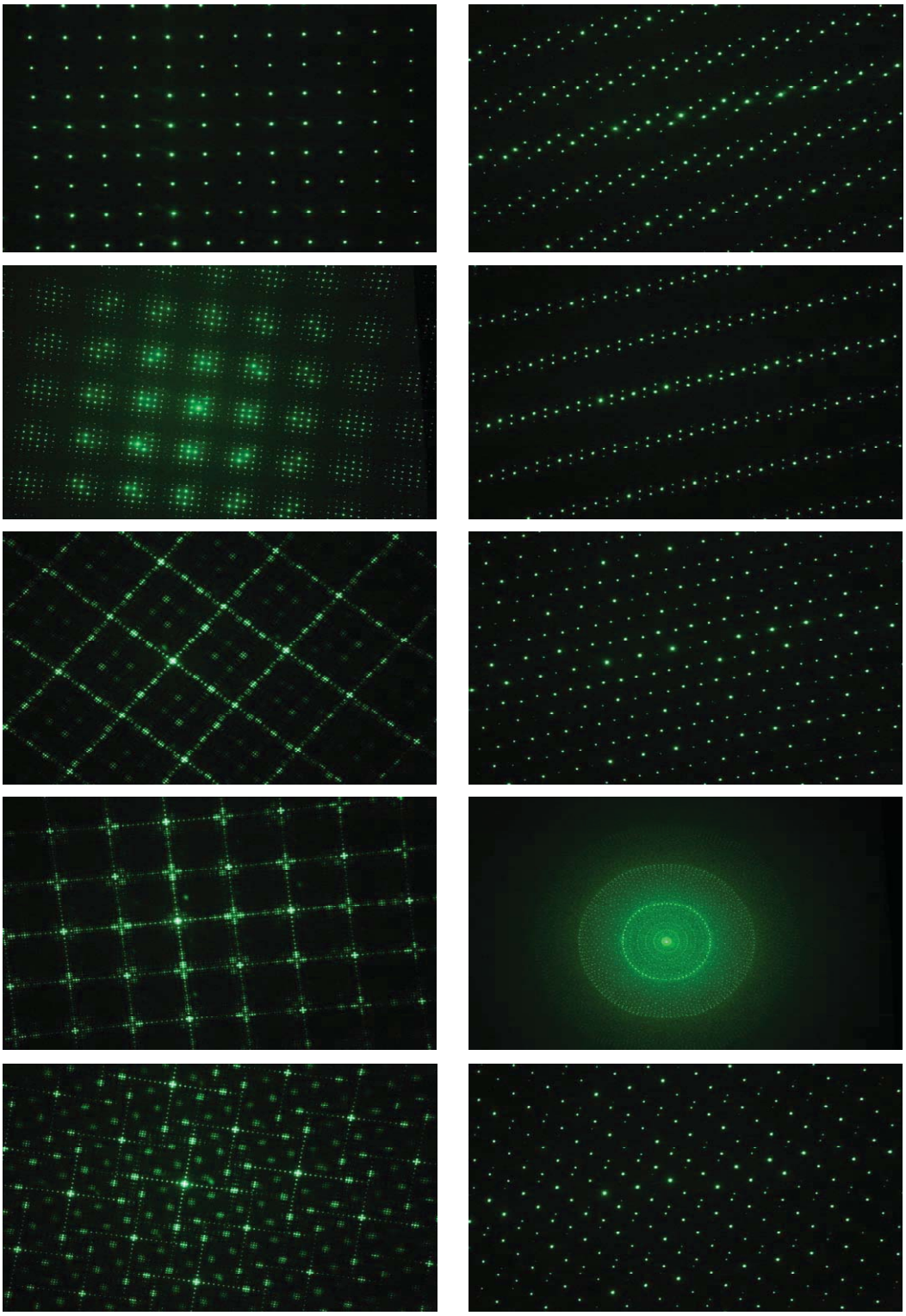


Figure 4.6 Some projected dots and shapes by using different grid heads.

4.1.3 Measurement Applications

After camera calibration, the 5 intrinsic parameters, 6 extrinsic parameters, and two lens distortion models for each camera in the camera-based measurement system (e.g., see Figure 1.1) are available for measurement applications. The 3D physical coordinates (x,y,z) of an object point (e.g., a retro-reflective marker, a laser point, or a feature point) on an object under measurement can be determined from $K(\geq 2)$ 2D images captured by K cameras, as shown next. It follows from (1.1.1) that

$$\zeta \begin{Bmatrix} u \\ v \\ 1 \end{Bmatrix} = \underset{3\times3}{[A]}[\underset{3\times4}{[T]}\{\tau\}] \begin{Bmatrix} x \\ y \\ z \\ 1 \end{Bmatrix} \tag{4.1.2}$$

Based on the [A] in (1.1.1), the first two equations and the third one of (4.1.2) can be separated as

$$\zeta \begin{Bmatrix} u \\ v \end{Bmatrix} = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \end{bmatrix} [\underset{3\times4}{[T]}\{\tau\}] \begin{Bmatrix} x \\ y \\ z \\ 1 \end{Bmatrix} \tag{4.1.3}$$

$$\zeta = \{T_{31} \quad T_{32} \quad T_{33} \quad \zeta_0\}\{x \;\; y \;\; z \;\; 1\}^T \tag{4.1.4}$$

Hence, (4.1.3) can be written in the following alternative form by using (4.1.4) as:

$$\zeta \begin{Bmatrix} u \\ v \end{Bmatrix} = \begin{bmatrix} uT_{31} & uT_{32} & uT_{33} & u\zeta_0 \\ vT_{31} & vT_{32} & vT_{33} & v\zeta_0 \end{bmatrix} \begin{Bmatrix} x \\ y \\ z \\ 1 \end{Bmatrix} = \begin{bmatrix} 0 & 0 & u \\ 0 & 0 & v \end{bmatrix} [[T]\{\tau\}] \begin{Bmatrix} x \\ y \\ z \\ 1 \end{Bmatrix} \tag{4.1.5}$$

Subtracting (4.1.5) from (4.1.3) yields

$$[C]_{2\times 4}\begin{Bmatrix} x \\ y \\ z \\ 1 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}, \quad [C] \equiv \begin{bmatrix} \alpha & \gamma & u_0 - u \\ 0 & \beta & v_0 - v \end{bmatrix} [[T]\{\tau\}] \tag{4.1.6}$$

where the undistorted image-plane coordinates (u, v) are obtained from the distorted image-plane coordinates $(\tilde{u}, \tilde{v})$ (from corner detection) using the camera's backward lens distortion model shown in (3.1.42). Because each image only provides 2 equations but there are 3 unknowns in (4.1.6), at least two images captured by two cameras from different orientations are needed. When *K* cameras are used, the *2K* equations can be written as

$$[M]_{2K\times 4}\{X\}_{4\times 1} = \{0\}_{2K\times 1}, \quad [M] \equiv \begin{bmatrix} [C]_1 \\ \vdots \\ [C]_K \end{bmatrix}, \{X\} \equiv \begin{Bmatrix} x \\ y \\ z \\ 1 \end{Bmatrix} \tag{4.1.7}$$

By least-squares fitting (see Appendix B), one can obtain $\{\bar{X}\}$ with $\|\{\bar{X}\}\| = 1$, and the answer is

$$\{X\} = \frac{\{\bar{X}\}}{\bar{X}_4} \tag{4.1.8}$$

where $\bar{X}_4$ is the 4*th* element of $\{\bar{X}\}$.

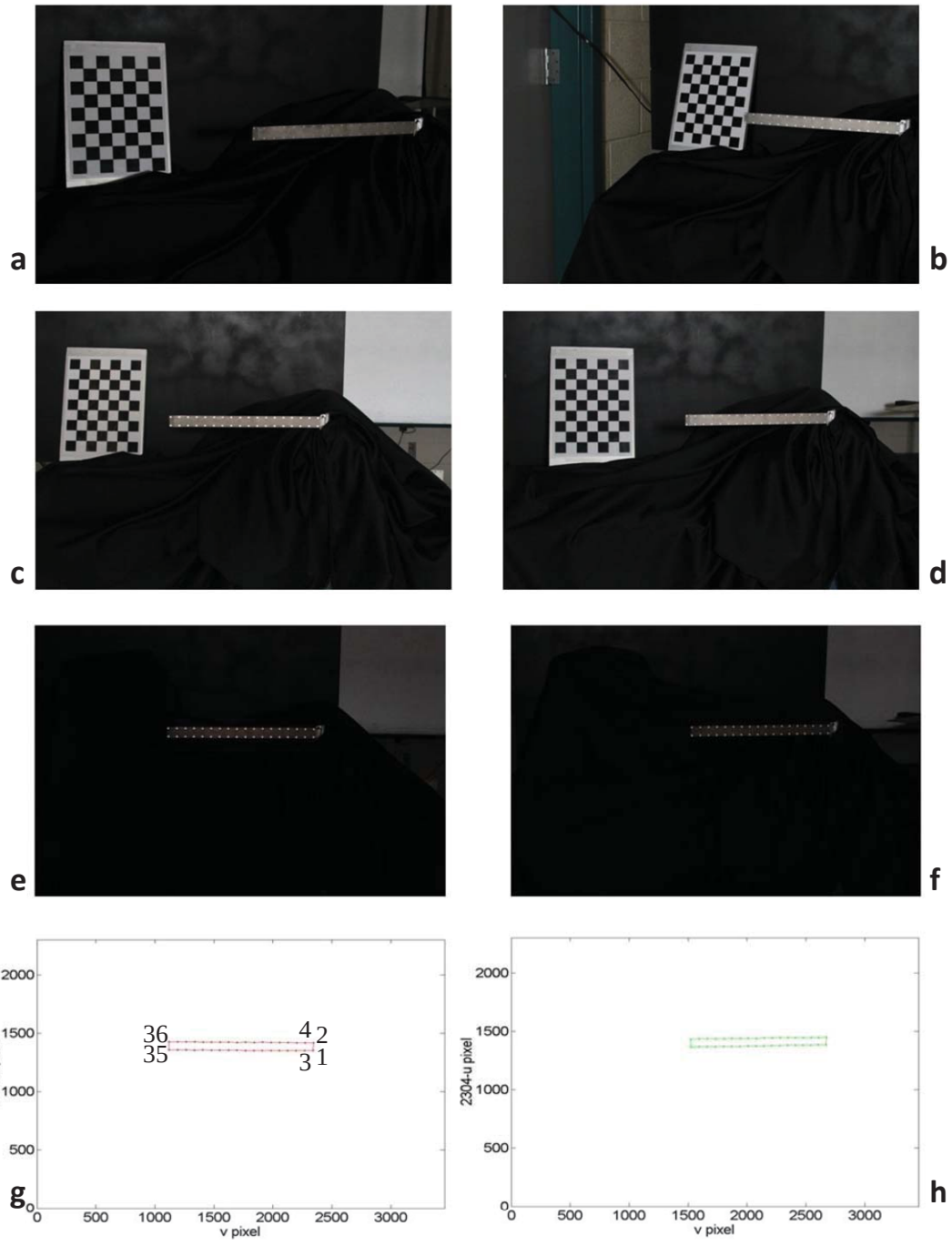
4.2 Static Measurement

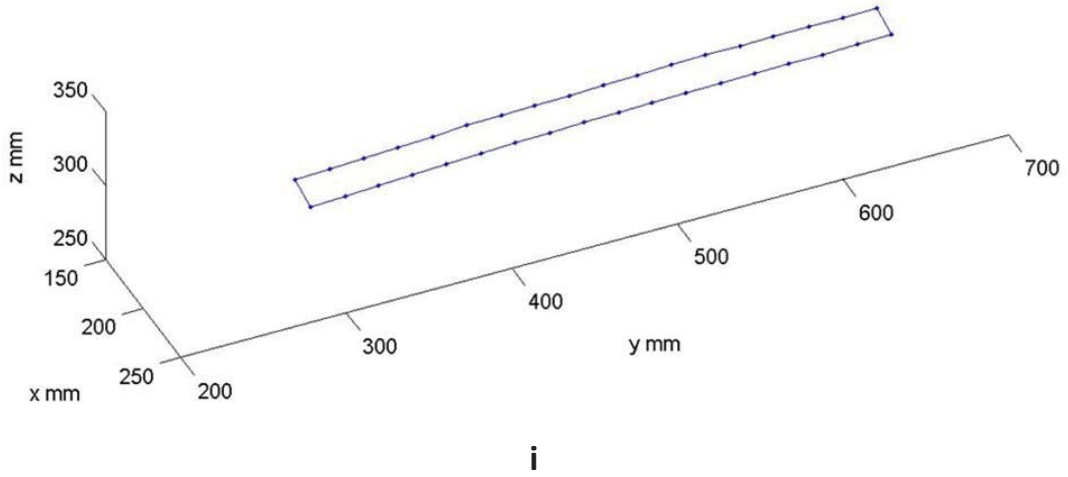
4.2.1 Use of Circular Retro-Reflective Markers

For static measurement, since the object does not move, one can use just one camera to perform 3D reconstruction. For this approach to work, at least two images of

the measured structure are needed for 3D reconstruction, and the extrinsic parameters (i.e., the location and orientation with respect to the camera) of each of the two images need to be as same as those of one of the checker board images used for camera calibration.

In Figure 4.7, we use only one camera with ISO speed at 2000, *f*-number at *f*/8, shutter speed at 1/100 second, and a focal length of 33mm. Moreover, the image format is JPEG, and the image-recording quality is medium fine with 3456×2304 pixels. Figure 4.7a-4.7d show the four images used for camera calibration. The camera setup and hence extrinsic parameters of Figures 4.7e and 4.7f are the same as those of Figures 4.7c and 4.7d, respectively. The beam is represented by the 36 circular retro-reflective markers arranged in two lines (see also Figure 4.1a). After camera calibration using the checker board images in Figures 4.7a-4.7d, the intrinsic parameters of the camera and the extrinsic parameters of Figures 4.7c and 4.7d (the same as those of Figures 4.7e and 4.7f, respectively) are available for measurement applications. The marker centers in Figures 4.7e and 4.7f can be extracted as described in Section 4.1.1. The first thresholds we used for Figures 4.7e and 4.7f are the average grey value of all pixels in each image plus 120. Note that, in the world coordinates, the x axis points downward, the y axis points to the right, and the z axis points out of the checker board.


a
b
c
d
e
f
g
h
36
35
4
2
3
1
2304-u pixel
v pixel
0
500
1000
1500
2000
2500
3000



i

Figure 4.7 a-d) images for camera calibration, e,f) images of markers on the beam, g,h) image coordinates of connected marker centers of (e) and (f), and i) 3D reconstruction of the connected marker centers in the world coordinates.

In order to guarantee the marker centers extracted from Figures 4.7e and 4.7f have one-to-one correspondence in order to perform 3D reconstruction, the marker centers extracted from both images need to be arranged in the same order. It is better to arrange and number markers by starting from a fixed end. For example, we start with the two markers at the fixed right end and number the lower marker as the first and the upper one as the second, and then number other marker centers sequentially, as shown in Figure 4.7g. These marker centers can be connected to show the beam profile, as shown in Figure 4.7g. The marker centers in Figure 4.7h need to be numbered in the same sequence in order to have a correct 3D reconstruction. Then, the static beam geometry can be reconstructed using Figures 4.7g and 4.7h and the method described in Section 4.1.3, as shown in Figure 4.7i in the world coordinate system.

After the physical coordinates (x, y, z) of all marker centers are obtained, one can use them to recalculate the image coordinates (u, v). Comparing the so-obtained (u, v) with those in Figure 4.7g, we obtain the reverse RMS errors to be 0.0344 pixels for *u* and 1.2672 pixels for *v*. Comparing the so-obtained (u, v) with those in Figure 4.7h, we obtain the reverse RMS errors to be 0.0071 pixels for *u* and 1.2363 pixels for *v*. The large error for *v* is caused by the large change of view angles between the two images along the *v* direction. Further discussions about this will be shown later in Section 4.3.2.

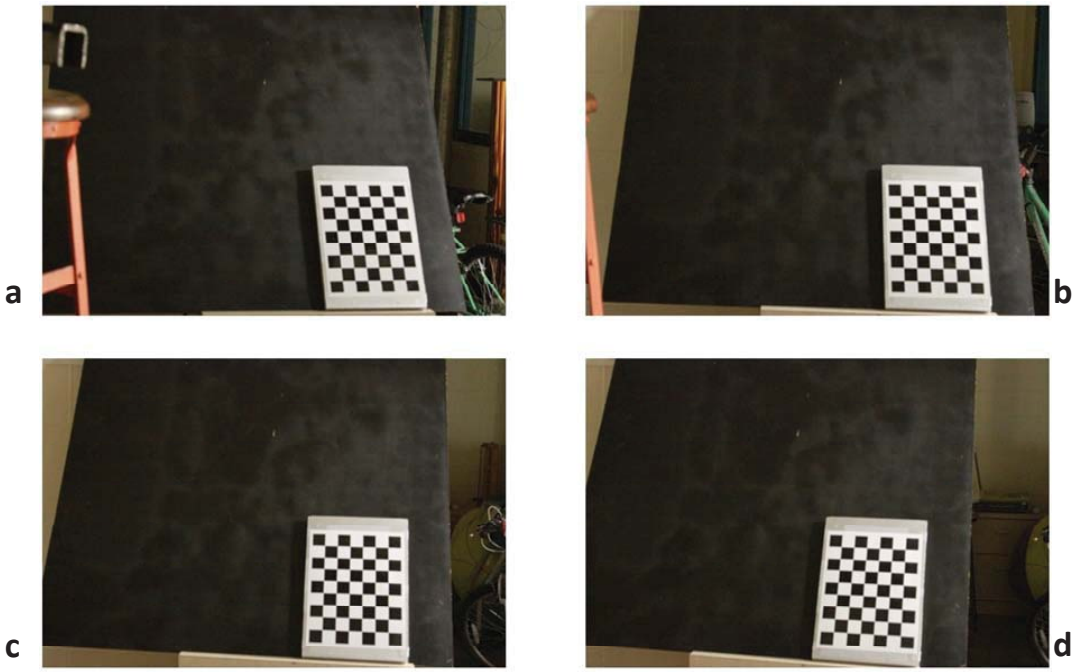
4.2.2 Use of Holographic Laser Points

When projected laser points are used, one can use the same image processing method used for processing images of retro-reflective markers. However, the challenge is how to correlate points on different images. This problem also exists for a complex distribution of many retro-reflective markers. A general approach is to use a feature vector to describe each point [104-106]. However, this method may not work well for repetitive patterns. Hence, a reference object needs to be used, or a special algorithm for each type of pattern is required. Of course, one can always number all interest points manually, but it is too time-consuming.

For Figure 4.8, we use one Canon EOS-7D camera with ISO speed at 6400, *f*-number at *f*/6.3, shutter speed at 1/4 second, and a focal length of 41mm. Moreover, the image format is JPEG, and the image-recording quality is medium fine with 3456×2304 pixels. Figure 4.8a-4.8d show the four images used for camera calibration. The camera setup and hence extrinsic parameters of Figures 4.8e and 4.8f are the same as those of Figures 4.8c and 4.8d, respectively.

Because the only purpose of this experiment is to show the feasibility of using holographic laser points for 3D reconstruction, the two distributions of laser points in Figures 4.8e and 4.8f are chosen to ease numbering of points. Here a square mask is used to ensure only 16 laser points from the holographic laser pointer go through, and the projections are set at two different angles, as shown in Figure 4.8e and 4.8f. To prevent *diffraction* [107,108], the mask needs to be not too small and not too close to the grid head of the laser pointer.

For the points in Figures 4.8e and 4.8f, we scan the image along the *v* direction (i.e., to the right) and number the located point centers in sequence. Then, the point centers in both images are numbered in the same sequence. For a complicated distribution of laser points, a feature detection method [109-111] needs to be used to obtain a consistent point numbering for both images, or one can number the points manually.



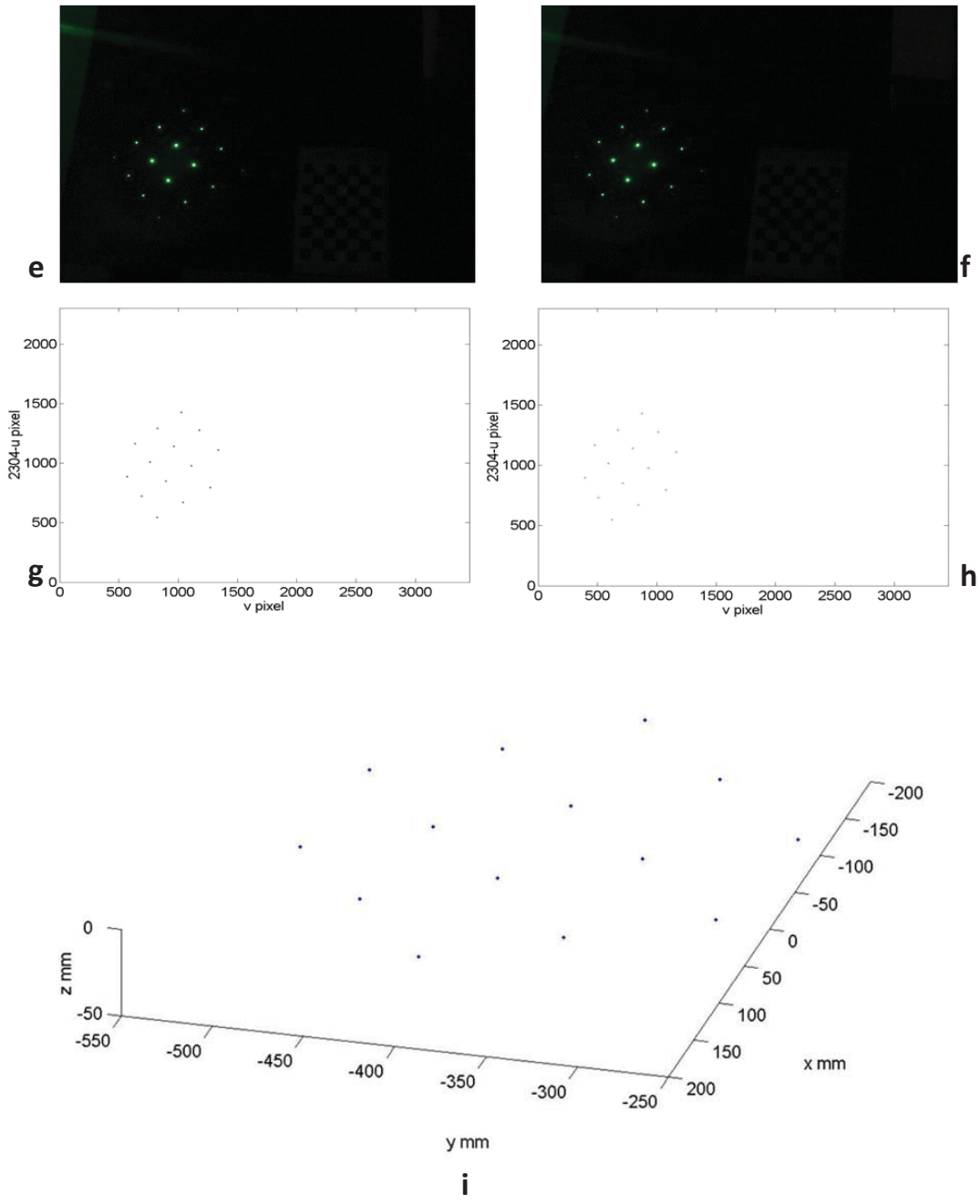


Figure 4.8 a-d) checker board images for camera calibration, e,f) images of projected laser points, g,h) image coordinates of laser point centers of (e) and (f), and i) 3D reconstruction of the laser point centers in the world coordinates.

Figures 4.8e and 4.8f show that some laser points are weak in brightness and the four inner points are very bright. If the first threshold is set too low, the four inner points will appear like large markers. If the first threshold is set too high, the weak points will be missed. In order to avoid the error caused by the brighter area, the first threshold is set to be the average grey value of all pixels in each image plus 60. Figures 4.8g and 4.8h show the so-obtained laser point centers, and Figure 4.8i shows the 3D reconstruction of the laser point centers in the world coordinates. Note that two weak corner points do not pass the first threshold and are missing in Figures 4.8g-4.8i.

After the physical coordinates (x, y, z) of all laser point centers are obtained, one can use them to recalculate the image coordinates (u, v). Comparing the so-obtained (u, v) with those in Figure 4.8g, we obtain the reverse RMS errors to be 0.0107 pixels for *u* and 0.4078 pixels for *v*. Comparing the so-obtained (u, v) with those in Figure 4.8h, we obtain the reverse RMS errors to be 0.0064 pixels for *u* and 0.4056 pixels for *v*. These results show that the use of holographic laser points results in better measurement accuracy. Again, the large error for *v* is caused by the large change of view angles between the two images along the *v* direction.

If the square mask is not used, lots of projected points will cover the whole image plane. After numbering them by feature detection and 3D reconstruction, the physical coordinates of all laser points on the structure and the background will be obtained. If the background is far from the structure, the laser points on the structure can be separated by examining the discontinuity of the z coordinate values.

4.3 Dynamic Measurement

4.3.1 Synchronization of Multiple Cameras

As shown in (4.1.7), at least two images simultaneously captured by two cameras from different locations/orientations are needed in order to estimate the world coordinates (x, y, z) of an object point. To have corresponding images/frames captured by different cameras at the same time, the cameras need to be synchronized. Unfortunately, synchronization of multiple cameras is one of the most critical issues in camera-based noncontact dynamic measurement.

Since the drift of camera clock crystals is specified by design, different models of cameras normally have different time delays after pressing shutter buttons. In order to guarantee the synchronization, all the cameras used in the system should have the same model with the same settings. If manual modes are available for some settings, they should be used. Otherwise the different time delays of different non-manual modes may cause non-synchronous capturing of images/frames. The following settings are recommended: manual shooting mode, manual focus mode (*AF mode*), stabilizer being off, same ISO speed, same *f*-number, same shutter speed, standard picture style, same color temperature (expresses *white balance*), auto lighting optimizer being off, and the same self-timer (*drive mode*). Figures 4.9 and 4.10 show a Canon EOS-7D camera's quick control screen and its LCD panel with some recommended settings.

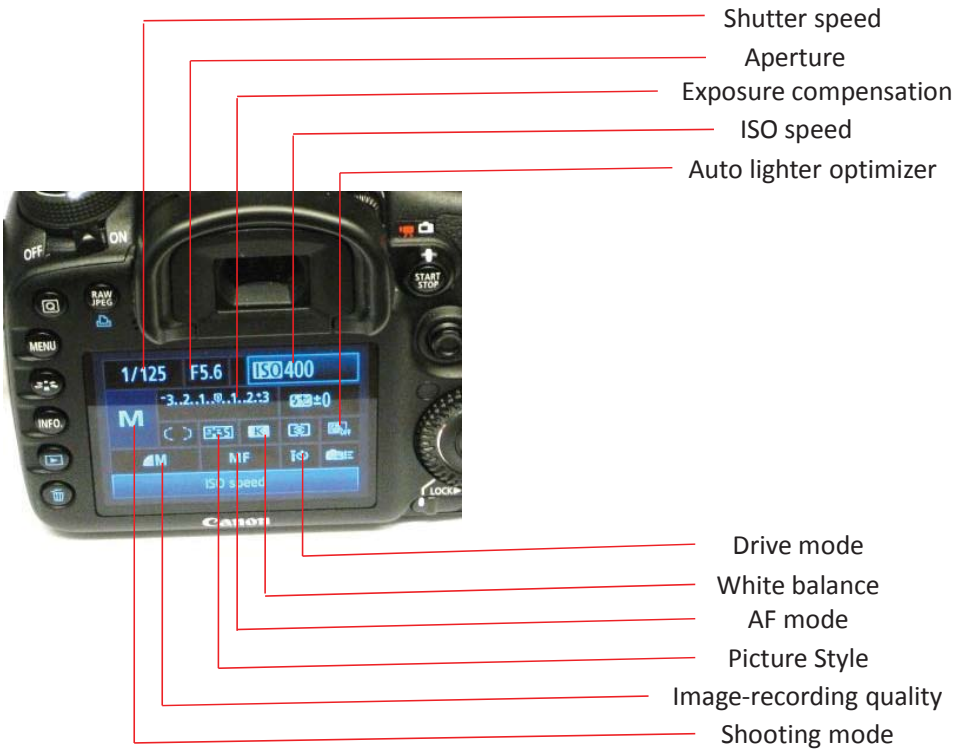


Figure 4.9 The quick control screen of a Canon EOS-7D camera.

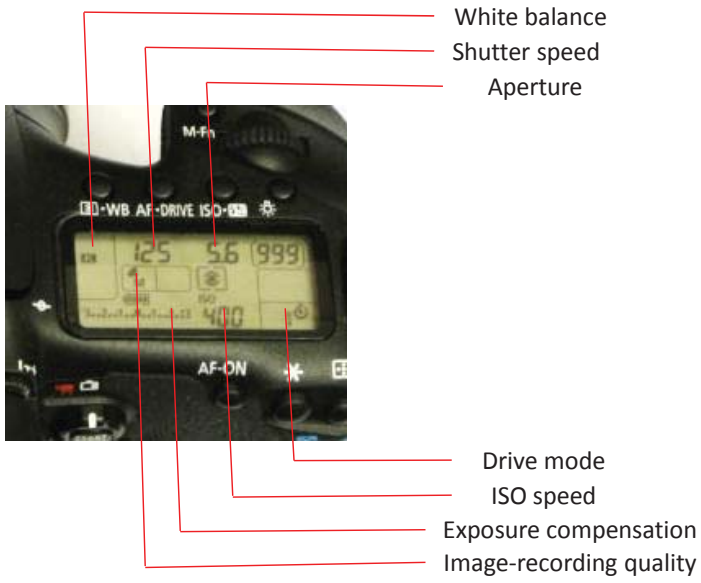


Figure 4.10 The top LCD panel of a Canon EOS-7D camera.

White balance (or color balance, gray balance, neutral balance) globally adjusts intensities of RGB colors to make *neutral colors* correct. Where *neutral colors* are the colors appeared to be without color, such as beige, ivory, taupe, black, gray, and white. *White balance* is counted by the color temperature (measured in Kelvin) of the light. The color temperature is considered as "warm" (side of red color) for 2,000 K to 4,000 K, "daylight" (combination of RGB) for 4,000 K to 7,000K, and "cool" (side of blue color) for 7,000 K to 11,000K [112]. If all the cameras used in the measurement system are set to auto white balance (AWB), it is hard to make sure each of the cameras uses the same amount of time for calculating the color temperature of its light source. For static measurement, the calculation of color temperature does not affect the result. For dynamic measurement, it will cause serious no synchronization problems. So the color temperatures of all cameras need to be set at the same number (e.g., 3,000K in our experiments).

To assure synchronization of all cameras, a wireless Canon RC-6 remote controller (see Figure 3.9) is used, but limitations on its emission angle and path length to each camera exist. Some wired remote controllers (e.g. Canon RS 80N3) can also be used. Unfortunately, more cables need to be connected to all cameras from the controller, and experiments show that unknown time delays between different cameras' initiations of recording still exist. Our experiments show that the synchronization by using a wireless remote controller is much better for a small number of cameras.

4.3.2 Dynamic Measurement System and Measurement Results

Before measurement the cameras need to be calibrated. Figure 4.11 shows the setup of two Canon EOS-7D cameras with a checker board for camera calibration and a hinged-free cantilevered flexible beam with 36 retro-reflective markers to be measured during vibration. The horizontal distance between the centers of two adjacent markers is about 20mm. First, the checker board is positioned at three or more different positions for camera calibration. Then, we use the K2007E0 miniature shaker from the Modal Shop, Inc. to excite the cantilevered beam to vibrate. By using these two cameras to record videos, the consecutive deformed configurations of the beam can be extracted by processing the two video files frame by frame.

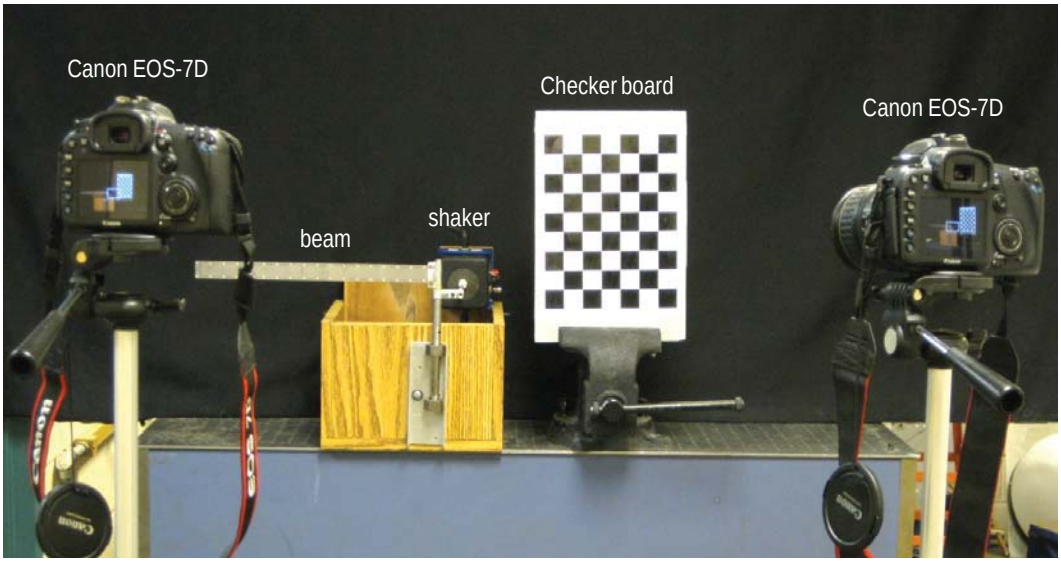
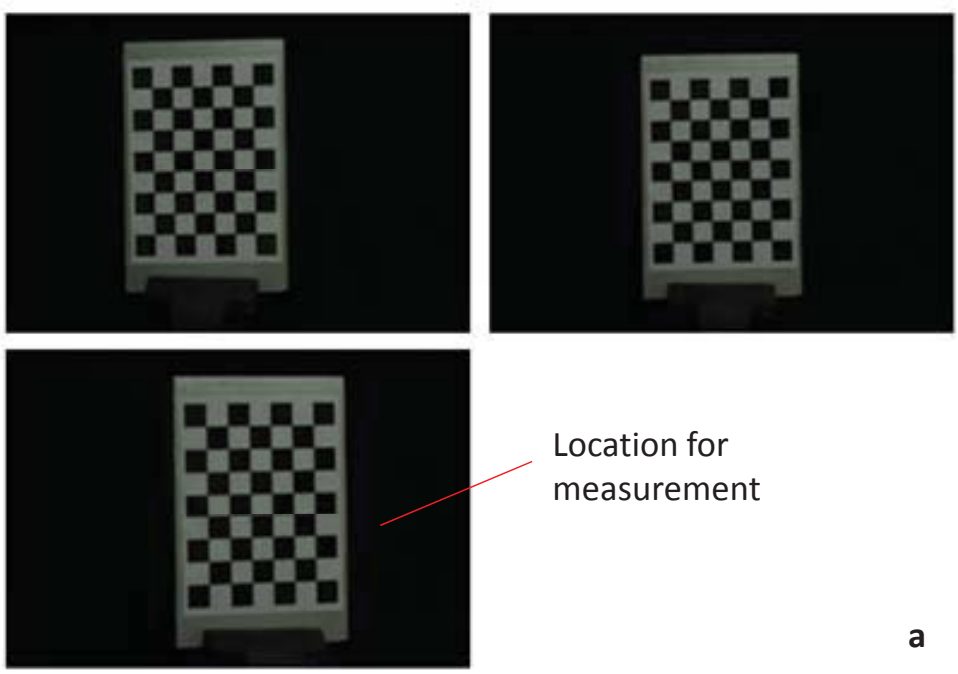


Figure 4.11 Experimental setup for camera calibration and static/dynamic measurements.

In order to have a long depth of field and guarantee synchronization, the settings of both cameras are: ISO speed at 6400, *f*-number at *f*/8, shutter speed at 1/320 second, and color temperature at 3000 K. The image format for camera calibration is JPEG with

the medium fine image-recording quality and 3456×2304 pixels. The focal length of the first camera is around 75mm and that of the second camera is around 70mm. Each video frame has 1280×720 pixels, and a frame rate of 60 FPS is used for video recording. Other settings follow the discussions presented in Section 4.3.1. Moreover, the excitation voltage created using the MATLAB command "sound" and delivered from a laptop computer's headphone jack to the electromagnetic shaker is $y = 0.3\cos(6\pi t)$. So that the beam's vibration period is 1/3 second, and hence 20 image fames per vibration period can be used for 3D reconstruction and showing dynamic deformations.

Figure 4.12 shows the images used for cameras calibration. The extrinsic parameters (i.e., the camera location and orientation with respect to the world coordinate system xyz) corresponding to the third image of each camera are used for defining the world coordinate system xyz (see Figure 1.2) and for measurement.



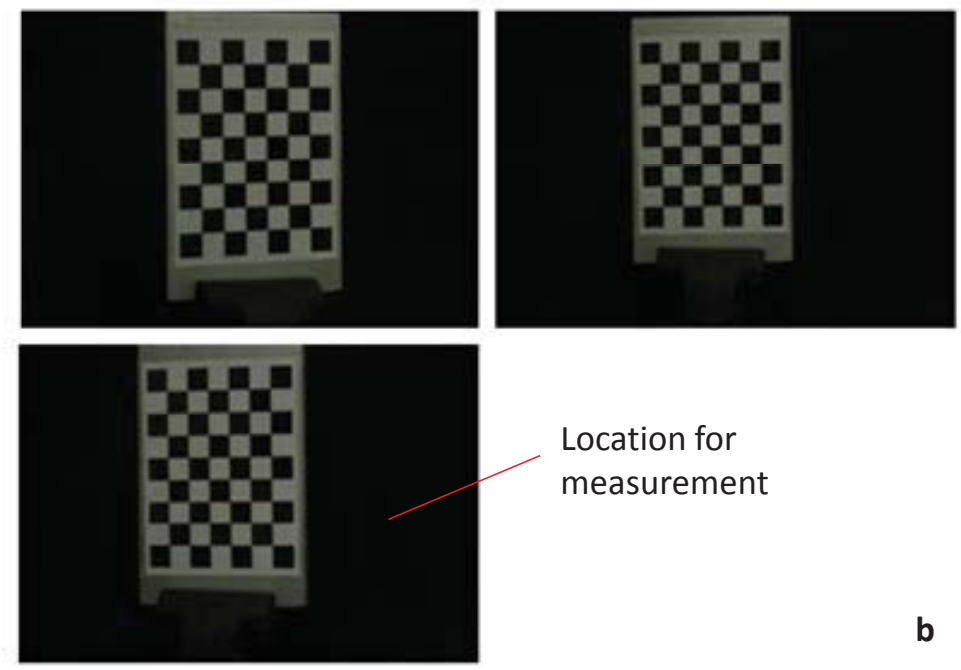


Figure 4.12 Images for camera calibration: a) first camera, and b) second camera.

The location vector $\{\tau\}$ and the three Euler angles for the rotation matrix $[T]$ corresponding to the third images in Figure 4.12 are:

$$\text{First camera:} \quad \{\tau\} \equiv \begin{Bmatrix} \xi_0 \\ \eta_0 \\ \zeta_0 \end{Bmatrix} = \begin{Bmatrix} 70.613 \\ -129.33 \\ 1954.7 \end{Bmatrix} mm; \quad \begin{Bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{Bmatrix} = \begin{Bmatrix} 276.9250^\circ \\ 191.4409^\circ \\ 8.3395^\circ \end{Bmatrix};$$

$$\text{Second camera:} \quad \{\tau\} \equiv \begin{Bmatrix} \xi_0 \\ \eta_0 \\ \zeta_0 \end{Bmatrix} = \begin{Bmatrix} 137.29 \\ -190.12 \\ 1920 \end{Bmatrix} mm; \quad \begin{Bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{Bmatrix} = \begin{Bmatrix} 76.8078^\circ \\ 195.1533^\circ \\ 347.2987^\circ \end{Bmatrix};$$

Figure 4.13 shows the first frames extracted from the two cameras' video files and the marker centers extracted from the two first frames.

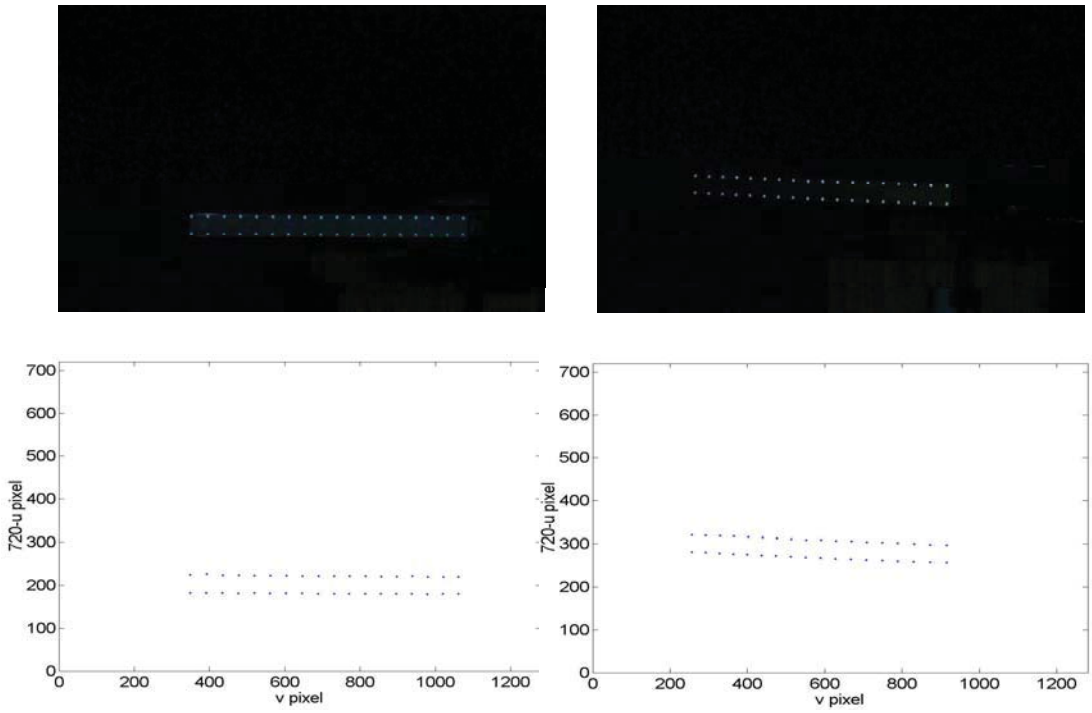


Figure 4.13 Two first frames from the two video files and the extracted marker centers.

Figure 4.14 shows the 3D reconstruction of 11 consecutive deformed beam geometries corresponding to Frames 7-17 of the two video files, and it covers one half of the vibration period.

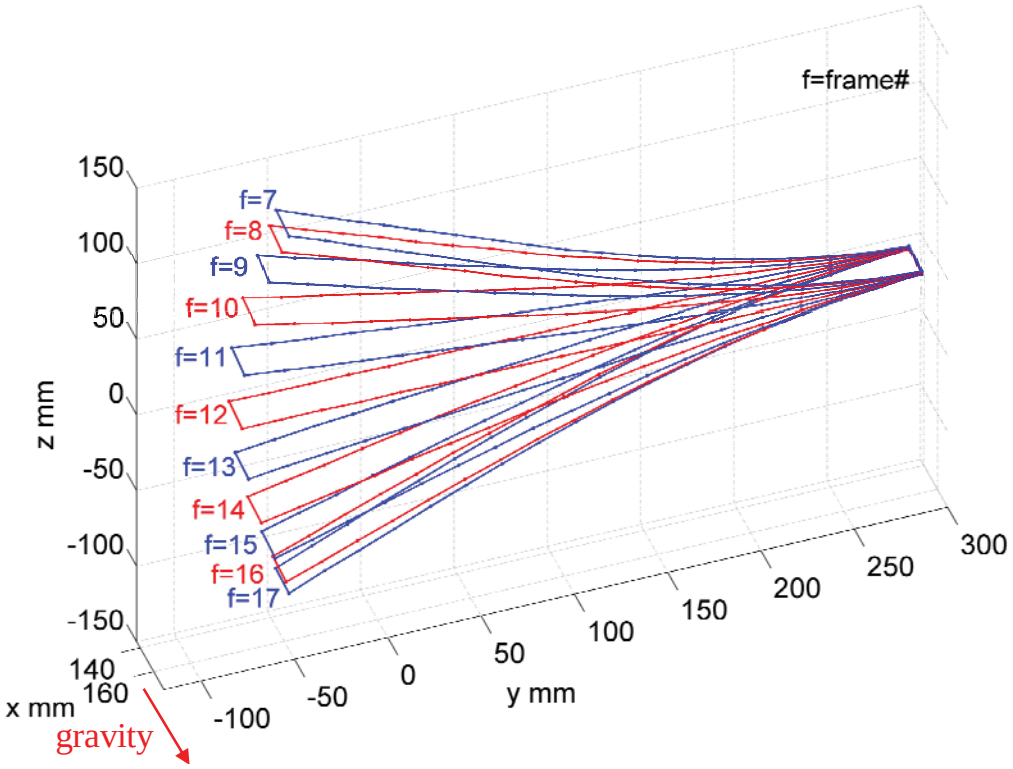


Figure 4.14 The 11 measured consecutive deformed beam geometries.

Table 4.1 lists the reverse RMS errors of image coordinates of frames 1-38. It shows that the errors along the *u* direction are smaller than those along the *v* direction. As discussed in Section 2.2.3, an area smaller than the circle of confusion is perceived as one point by human eyes. Nevertheless, except for an object on the focal plane (see Figure 2.2), errors always exist even if the object is inside the depth of field. To reduce this error, all parts of the measured structure need to be as close as possible to the focal plane. Also, a longer depth of field always provides higher measurement accuracy. As discussed in Section 4.2 Static Measurement, the big errors of *v* are caused by the change of view angles along the *v* direction during cameras calibration and the angles between cameras during measurement. Hence it is better to have a small change of the view angle for

cameras calibration and a small angle between cameras for measurement, and to keep the optic axis ζ of each cameras as perpendicular to the calibration plane and the measurement plane as possible.

Table 4.1 Reverse RMS errors of image coordinates of frames 1-38.

Frame number	Reverse RMS errors (pixel)			
	Image 1		Image 2	
	u axis	v axis	u axis	v axis
1	0.0462	1.5695	0.0092	1.7103
2	0.0394	1.5712	0.0029	1.7107
3	0.0329	1.5110	0.0078	1.6445
4	0.0278	1.3309	0.0084	1.4468
5	0.0260	1.2347	0.0077	1.3427
6	0.0230	0.9843	0.0033	1.0732
7	0.0209	0.8605	0.0020	0.9387
8	0.0182	0.7095	0.0027	0.7760
9	0.0167	0.6240	0.0034	0.6856
10	0.0159	0.5881	0.0036	0.6465
11	0.0174	0.6517	0.0033	0.7149
12	0.0192	0.7142	0.0036	0.7799
13	0.0234	0.8411	0.0049	0.9183
14	0.0303	1.0107	0.0082	1.1010
15	0.0363	1.1511	0.0119	1.2547
16	0.0450	1.3321	0.0175	1.4509
17	0.0475	1.3824	0.0191	1.5061
18	0.0497	1.4284	0.0200	1.5554
19	0.0556	1.5745	0.0221	1.7145

Frame number	Reverse mean absolute errors (pixel)			
	Image 1		Image 2	
	u axis	v axis	u axis	v axis
20	0.0502	1.5290	0.0162	1.6649
21	0.0455	1.5467	0.0088	1.6843
22	0.0393	1.5731	0.0030	1.7133
23	0.0327	1.5164	0.0083	1.6497
24	0.0277	1.3221	0.0083	1.4373
25	0.0263	1.2664	0.0088	1.3775
26	0.0226	1.0171	0.0052	1.1066
27	0.0210	0.8677	0.0020	0.9472
28	0.0181	0.7149	0.0029	0.7817
29	0.0162	0.6241	0.0033	0.6836
30	0.0161	0.6066	0.0035	0.6668
31	0.0166	0.6287	0.0034	0.6892
32	0.0197	0.7357	0.0036	0.8035
33	0.0238	0.8497	0.0051	0.9265
34	0.0314	1.0477	0.0088	1.1412
35	0.0355	1.1336	0.0116	1.2354
36	0.0446	1.3291	0.0174	1.4480
37	0.0479	1.3855	0.0195	1.5089
38	0.0491	1.4303	0.0195	1.5581

Figure 4.15 shows that the RMS errors vary with the vibration at the same frequency. This phenomenon is caused by the deviation of the beam's location from the focal plane of the two cameras, which can also be considered as the change of view angles between the two cameras.

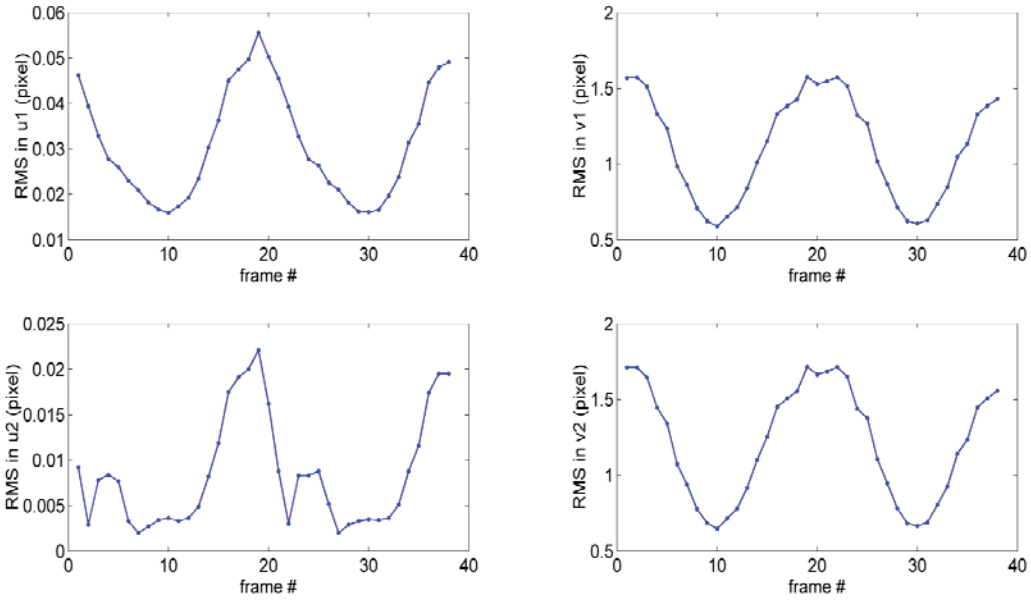


Figure 4.15 Reverse RMS errors of image coordinates of frames 1-38.

The distance between two adjacent retro-reflective markers on the beam is about 20mm. Since the markers were stuck on the beam by hand, the distance may not be exactly 20mm. In order to check the measurement accuracy in the world coordinates, the center distance between the two markers on the moving tip is calculated and listed in Table 4.2.

Table 4.2 The center distance between the two markers on the moving tip from frames 1-38.

Frame number	Distance (mm)	Frame number	Distance (mm)	Frame number	Distance (mm)
1	20.7592	14	20.5620	27	20.7267
2	20.7210	15	20.6937	28	20.5870
3	20.6958	16	20.6624	29	20.6081
4	20.7160	17	20.7884	30	20.6927
5	20.7022	18	20.9060	31	20.5724
6	20.6217	19	20.8847	32	20.6185
7	20.6728	20	20.7158	33	20.5507
8	20.5860	21	20.7514	34	20.6046
9	20.5952	22	20.6803	35	20.6072
10	20.5696	23	20.6511	36	20.6841
11	20.6269	24	20.6817	37	20.7306
12	20.6341	25	20.5519	38	20.7676
13	20.5984	26	20.5633		

The standard deviation of the measured distances is calculated from Table 4.2 as

$$\sigma = \sqrt{\frac{\sum_{i=1}^{38}(x_i - \overline{x})^2}{38}} = 0.0847\text{mm} \tag{4.3.1}$$

where $\overline{x}$ is the mean number of x_i. Since the maximum measured length of this cantilevered flexible beam is around $L \approx 20\text{mm} \times (38/2\text{-}1) = 340\text{mm}$. The measurement accuracy can be presented as the standard deviation divided by the maximum measured length as

$$\frac{\sigma}{L} = 2.4912 \times 10^{-4} \tag{4.3.2}$$

The time-varying coordinates of the middle point on the beam tip can be estimated as averaging the coordinates of the two tip markers. Setting frame 2 as the neutral position, Figure 4.16 shows the time-varying displacement of the middle point.

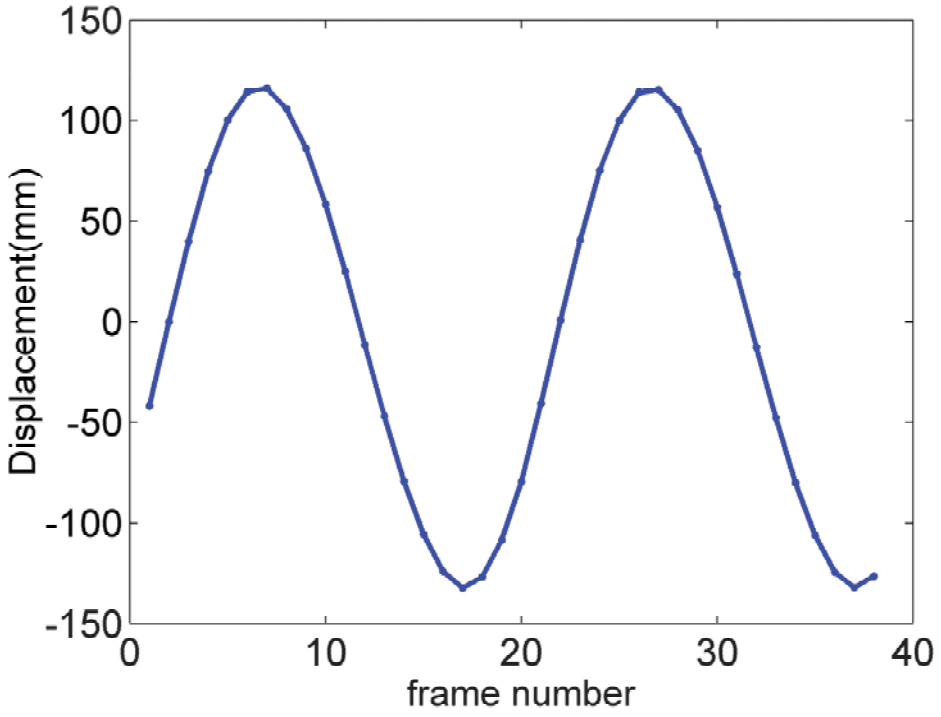


Figure 4.16 Displacement of the middle point between the two tip markers from frames 1-38.

By using these displacements, the amplitudes of the middle point during each period are listed in Table 4.3.

Table 4.3 Amplitudes of the middle point.

Frames used	Amplitude (mm)
Frame 7 and Frame 17	244.1486
Frame 17 and Frame 27	243.5733
Frame 27 and Frame 37	243.3680
Frame 37 and Frame 47	243.9724
Frame 47 and Frame 57	244.5451

As discussed in Section 4.1.1, some marker images may intersect or overlap at times, especially for rotated or highly deflected flexible structures. In addition, some view angles with weak light source may miss some markers. Hence one needs to pre-define a total number of markers to help the system check it. Moreover, cameras need to be appropriately located to ensure that at least two cameras can see all the markers at any time. Then one can use only the cameras seeing all markers to reconstruct the 3D positions.

In order to double check the robustness of the proposed measurement system, we also tested a hinged-free L-shaped flexible beam. As shown in Figure 4.17, the longer arm of the L-shaped beam is along the horizontal direction with its right end connected to the shaker, and the shorter arm is perpendicular to the longer arm with a free end. The distances between two adjacent markers are about 7.5mm and 20mm, as shown in Figure 4.17. There are 32 markers on the longer arm, 22 markers on the shorter arm, and 4 markers on the joint part. So there are 58 markers in total. All the settings are the same as those used for the cantilevered beam except the excitation signal to the shaker.

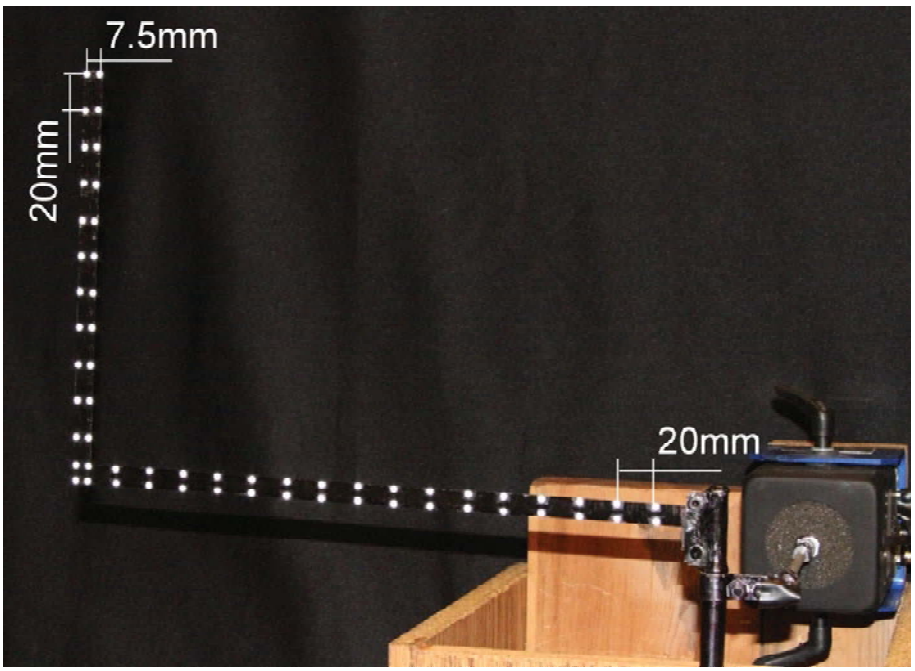


Figure 4.17 A hinged-free L-shaped flexible beam for vibration testing.

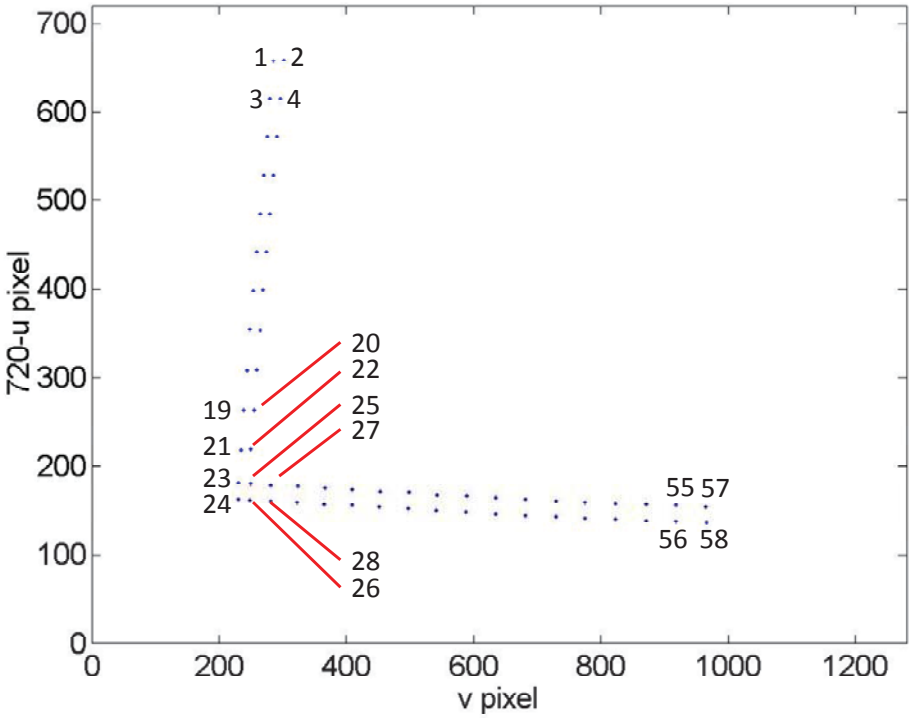


Figure 4.18 Numbering of the marker centers on the L-shaped beam.

In order to assure the marker centers on frames from different cameras have one-to-one correspondence, the marker centers extracted from two different images should be numbered in the same sequence. Our approach is to start the numbering from the static end. As shown in Figure 4.18, we number the upper point on the static end as point 57, the lower one as point 58. Then we compare the distances between point 58 and the other neighboring points and set the shortest one as point 56 and the second shortest one as point 55. Similarly, we compare the distances between point *i* and other neighboring points and set the point with the shortest distance as point *i-2* the second shortest one as point *i-3*. Repeat this process until points 24 and 23 are reached, and then treat point 23 as the starting point *i*. Then, again, we compare the distances between point *i* and other neighboring points and set the point with the shortest distance as point *i-2* and the one with the second shortest distance as point *i-1*. Repeat the process until point 1 is reached. Then one can use the measurement theory presented in Section 4.1.3 to process frames from the two cameras to reconstruct the deformed L-frame geometries for dynamic animation.

When the excitation voltage created using the MATLAB command “sound” and delivered from a laptop computer’s headphone jack to the electromagnetic shaker is $y = 0.3\cos(4\pi t)$, the L-shaped beam’s vibration period is 1/2 second, and hence 30 image fames per vibration period can be used for 3D reconstruction and showing dynamic deformations. Figure 4.19 shows the 3D reconstruction of 16 consecutive deformed L-shaped beam geometries corresponding to Frames 8-23 of the two video files, and it covers one half of the vibration period.

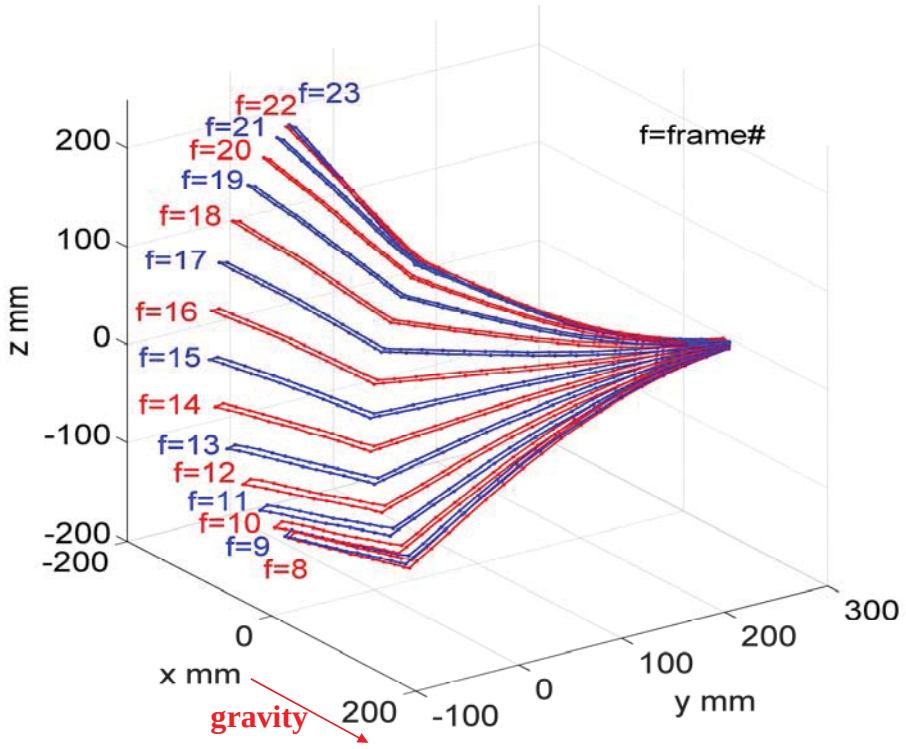


Figure 4.19 The 16 measured consecutive deformed structural geometries.

It follows from (4.3.1) that the standard deviation of the measured distance between the two tip markers on the moving tip is $\sigma = 0.0249\text{mm}$. With the length of the longer arm being $L \approx 325\text{mm}$, the measurement accuracy can be presented as $\sigma / L = 7.6731 \times 10^{-5}$.

When the excitation voltage is $y = 0.3\cos(10\pi t)$, the L-shape beam's second mode shape is excited. The L-shaped beam's vibration period is 1/5 second, and hence 12 image fames per vibration period can be used for 3D reconstruction and showing dynamic deformations. Figure 4.20 shows the 3D reconstruction of 6 consecutive deformed L-shaped beam geometries corresponding to Frames 4-9 of the two video files, and it covers one half of the vibration period. Note the curve near the fixed end is caused

by the connection with the shaker. Figure 4.19 also shows this curve, but it is not as obvious.

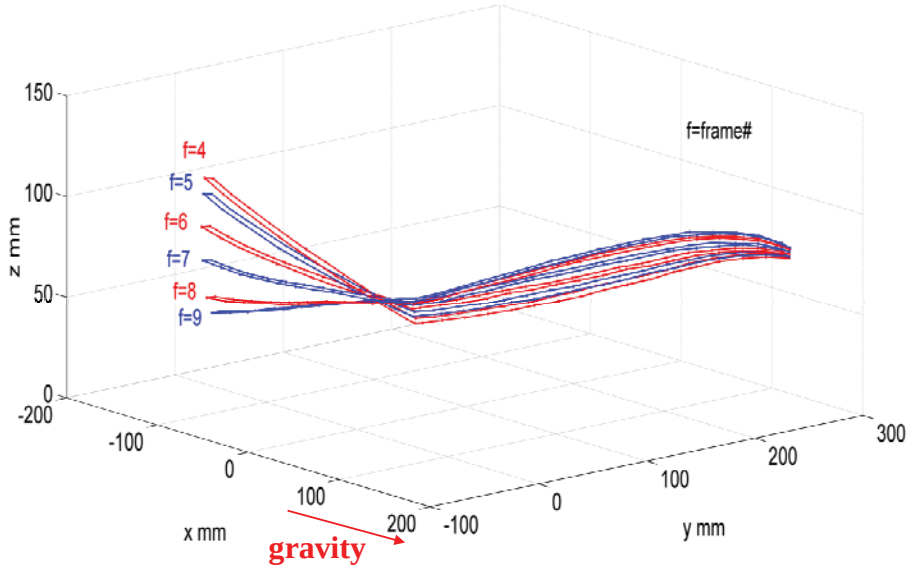


Figure 4.20 The 6 measured consecutive deformed structural geometries.

It follows from (4.3.1) that the standard deviation of the measured distance between the two tip markers on the moving tip is $\sigma = 0.0208\text{mm}$. With the length of the longer arm being $L \approx 325\text{mm}$, the measurement accuracy can be presented as $\sigma / L = 6.3865 \times 10^{-5}$.

4.3.3 Conclusions

Comparing with the measurement accuracy of the Eagle-500 motion analysis system ranging from 1/200 to 1/3000, the proposed system is capable of a higher measurement accuracy ranging from 1/4000 to 1/16000 in our experimental cases. However, the highest frame rate of the Eagle-500 motion analysis system is 2000 FPS,

but that of the proposed system is only 60 FPS. On the other hand, each camera of the Eagle-500 motion analysis system costs more than $10,000, but each camera of the proposed system costs about $1,500. Furthermore, with the developed image processing methods and the measurement theory, one can use cheaper cameras for accurate vibration measurements.

CHAPTER 5

SUMMARY AND RECOMMENDATIONS

5.1 Summary

In this thesis, we presented theoretical and experimental studies and procedure for development of a high-precision easy-to-use noncontact vibration measurement system based on the use of high-speed high-resolution digital cameras and advanced image processing techniques. Compared with the Eagle-500 motion analysis system in the Structural Mechanics and Controls Laboratory of MU, numerical and experimental results show that the proposed system has higher measurement accuracy.

In order for users to obtain high measurement accuracy in using such a camera-based noncontact measurement system, we provide the following guidelines for setting up such a camera system. (1) The checker board for camera calibration and the structure to be measured should be located around the center of and within the depth of field of each camera. (2) The angle between each two cameras should be small. (3) Set each camera's stabilizer and auto lighting optimizer to off, shooting mode and focus mode to manual, and picture style to standard. (4) Set all cameras to have the same ISO speed, *f*-number, shutter speed, color temperature, self-timer, image format, image-recording quality, and video recording quality. The recommended settings for each camera are ISO speed at 6400, *f*-number at *f*/8, shutter speed at 1/320 second, color temperature at 3000, self-timer at a 2-second delay, image format as JPEG, image-recording quality as medium fine, video recording quality at 1280×720, and a frame of 60 FPS. Focal lengths of cameras do not need to be the same. Each camera's location, settings and focal length need to be fixed during camera calibration process and static/dynamic measurement.

The next step after setting up all cameras of the noncontact measurement system is to calibrate all cameras. Since all cameras are set up and fixed, one needs to position the calibration model plane (a checker board here) at 3 (or more) different angles and locations for cameras to capture pictures using a wireless remote controller in order to avoid camera vibration. In order to reduce the influence of the depth of field, it is better to keep the checker board about parallel to the sensor plane.

After capturing the pictures of the checker board for camera calibration, one needs to switch the camera to the movie shooting mode and then use the remote controller to start capturing video images of the dynamic structure under measurement with retro-reflective markers on it. Using a wireless remote controller here is to avoid camera vibration and to guarantee synchronization of multiple cameras.

Subsequently, one needs to transport the pictures and videos from all cameras to a computer for post-processing. The frames of a video file can be extracted using the Canon utility software "ZoomBrowser EX". The static images of the checker board are used for camera calibration as shown in Chapter 3, and the frames from video files of the structure under measurement are used to find centers of the markers on the structure and then to perform 3D reconstruction of the consecutive deformed geometries and dynamic animation. The dynamic deformations of one cantilevered beam and one L-shaped beam were measured using the proposed camera-based noncontact measurement system. The experimental results confirm the accuracy and flexibility of the system.

5.2 Recommendations for Future Study

Although experimental results show that the proposed camera-based measurement system has high measurement accuracy, there are still rooms for improvement. For example, the light source used in our experiments is a simple fluorescent lamp, which causes light reflection for some materials. The light reflection may change a marker's image shape on the photo sensor, or may even create several extra image points on the sensor and makes it hard to find a marker's center. If this phenomenon happens, one cannot automatically process all frames from a video file, and then one needs to process frame by frame with different thresholds and noise filtering. In order to avoid the reflection problem, we suggest using some other light sources (such as an LED ring light around the camera lens). However, since big contrast between markers and the background is needed for better image processing results, the light source cannot be too bright. One can also use more cameras to avoid the reflection problem. When the image captured by a camera appears to have less or more marker points, one can ignore this image and use images from other cameras. But at least two cameras are required to work for each frame. After solving the reflection problem, all marker centers can be automatically extracted from all video frames.

Another thing hinders automatic processing of markers and 3D reconstruction is point tracking between two consecutive frames. The marker centers need to be arranged in the same order for all frames before 3D reconstruction. Although one can use a special algorithm to arrange marker center points for a special structure, for an arbitrary structure with many random markers, we recommend using feature detection to arrange marker points.

When the holographic laser points are used, an algorithm to extract projected points on the measured structure is required. If the background is far from the structure, the points on the structure and those on the background can be separated by their distances to the camera. Then one can discard those points projected on the background in order to obtain the 3D deformed structural configuration.

Since the wireless remote control sensor is in the front of each Canon EOS-7D camera, a wireless remote controller is difficult to simultaneously start several widely distributed cameras. In order to simultaneously start recording of several cameras at different view angles with respect to the structure under measurement, a wired controller is recommended for future research. The critical issue of using a wired controller is how to synchronize several cameras without time delay between cameras or even with a specified time delay. It is hard to have all cameras start recording at the same time because signals need time to move from one camera to another. One possible approach is to specify a time delay between two adjacent cameras and the time delay needs to be an integer times of the sampling time interval. After video frames are extracted, use only those frames after all cameras start to record. More software and hardware developments are needed for this approach.

To extract the image coordinates of a non-isolated image point from a 2D image, a *feature extraction method* is needed. Feature-based search is to find another image patch that looks like a known image patch, and it will be an essential technology over next decades [113]. A local feature is an image pattern which differs from its immediate neighborhood. Local features can be points, edges, or small image patches. The image properties commonly used for feature extraction are intensity, color, and texture.

Typically, some measurements are taken from a region centered on a local feature and converted into descriptors for various applications. Features are typically found at an informative location that has high variability in its intensity pattern. The search for robust texture-based photogrammetry is still wide open and is worth further study.

REFERENCES

[1] Shabana, A.A., 1997. "Flexible multibody dynamics: Review of past and recent developments," Multibody System Dynamics 1, 189–222.

[2] Pai, P.F., 2007. Highly Flexible Structures: Modeling, Computation and Experimentation, AIAA, Reston, Virginia.

[3] Jenkins, C.M., 2001. Gossamer Spacecraft: Membrane and Inflatable Structures Technology for Space Applications, AIAA, Reston, Virginia.

[4] Ambrosio, J. and Goncalves, J., 2001."Complex flexible multibody systems with application to vehicle dynamics," Multibody System Dynamics 6(2), 163-182.

[5] Zhao, X., Maisser, P., and Wu, J., 2007. "A new multibody modeling methodology for wind turbine structures using a cardanic joint beam element," Renewable Energy 32, 532-546.

[6] Hodges, D.H. and Yu, W., 2007. "A rigorous, engineer-friendly approach for modeling realistic composite rotor blades," Wind Energy 10, 179-193.

[7] Faltinsen, O., 1990. Sea Loads on Ships and Offshore Structures, Cambridge University Press.

[8] Pai, P.F., Ramanathan, S., Hu, J., Chernova, D.K., Qian, X., and Wu, G., 2010. "Camera-based noncontact metrology for static/dynamic testing of flexible multibody systems," Measurement Science and Technology 21(8), 085302 (14pp).

[9] Atkinson, K.B., 1996. Close Range Photogrammetry and Machine Vision, Whittles Publishing, Scotland, UK.

[10] Gruen, A., 1997. "Fundamentals of videogrammetry — A review," Human Movement Science 16 (2-3),155-187.

[11] Mikhail, E.M., Bethel, J.S., and McGlone, J.C. 2001. Introduction to Modern Photogrammetry, John Wiley & Sons, New York.

[12] Photomodeler User's Manual 6.3, EOS Systems Inc., Vancouver, Canada, 2009. (http://www.photomodeler.com).

[13] Greaves, J., EVaRT 4.6 User's Manual, Motion Analysis Corporation, Santa Rosa, CA.
(http://www.motionanalysis.com).

[14] User's Manual, Vicon Industries, Hauppauge, NY.

(http://www.vicon-cctv.com)

[15] V-STARS User's Manual, Geodetic Services Inc., Melbourne, FL, 2000. (http://www.geodetic.com)

[16] Qian, X., Du, X., and Pai, P.F.,2010. "Experimental nonlinear dynamics of a highly flexible spinning beam using a 3D motion analysis system," 51st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Orlando, Florida, April 12-15.

[17] Ganci, G. and Brown, J., 2000. "Developments in non-contact measurement using videogrammetry," Boeing Large Scale Metrology Seminar.

[18] Black, J.T. and Pappa, R.S., 2003. "Videogrammetry using projected circular targets: proof-of-concept test," NASA/TM-2003-212148.

[19] Zhang, Z., 2000. "A flexible new technique for camera calibration," IEEE Transactions on Pattern Analysis and Machine Intelligence 22(11), 1330-1334.

[20] Moravec, H.,1980. "Obstacle avoidance and navigation in the real world by a seeing robot rover," Tech Report CMU-RI-TR-3 Carnegie-Mellon University, Robotics Institute.

[21] Harris, C.G. and Stephens, M., 1988. "A combined corner and edge detector," Proceedings of The Fourth Alvey Vision Conference, Manchester, pp. 147-151.

[22] Shi, J. and Tomasi, C., 1994. "Good features to track," 9th IEEE Conference on Computer Vision and Pattern Recognition, Springer.

[23] Tomasi, C. and Kanade, T.,2004. "Detection and tracking of point features," Pattern Recognition 37, 165-168.

[24] Foerstner, W. and Gulch, E., 1987. "A fast operator for detection and precise location of distinct points, corners and centres of circular features," ISPRS Intercommission Workshop, Interlaken.

[25] Wang, H. and Brady, M., 1995. "Real-time corner detection algorithm for motion estimation," Image and Vision Computing 13(9), 695-703.

[26] Smith, S.M. and Brady, J.M., 1997. "SUSAN - a new approach to low level image processing," International Journal of Computer Vision 23(1), 45-78.

[27] Smith, S.M. and Brady, J.M., 1997. "Method for digitally processing images to determine the position of edges and/or corners therein for guidance of unmanned vehicle," UK Patent 2272285, Proprietor: Secretary of State for Defense, UK.

[28] Trajkovic, M. and Hedley, M., 1998. "Fast corner detection," Image and Vision Computing 16(2), 75-87.

[29] Rosten, E. and Drummond, T., 2006. "Machine learning for high-speed corner detection," European Conference on Computer Vision.

[30] Schmid, C., Mohr, R., and Bauckhage, C., 1998. "Comparing and evaluating interest points," International Conference on Computer Vision, pp. 230-235.

[31] Schmid, C., Mohr, R., and Bauckhage, C., 2000. "Evaluation of interestpoint detectors," International Journal of Computer Vision37(2),151–172.

[32] Tsai, R.Y., 1987. "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," IEEE Journal of Robotics and Automation 3(4), 323-344.

[33] Maybank, S.J. and Faugeras, O.D., 1992. "A theory of self-calibration of a moving camera," International Journal of Computer Vision 8(2), 123-152.

[34] Hartley, R.I., 1994. "An algorithm for self calibration from several views," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 908–912.

[35] Fischler, M.A. and Bolles, R.C., 1981. "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," Communications of the Association for Computing Machinery 24, 381-395.

[36] Wen, J. and Scbweitzer, G.,1991. "Hybrid calibration of CCD cameras using artificial neural nets," IEEE International Joint Conference Neural Networks, vol. 1, pp. 337-342.

[37] Czaplewski, R.L.,1992. "Misclassification bias in areal estimates, "Photogrammetric Engineering and Remote Sensing58(2), 189-192.

[38] Caprile, B. and Torre, V., 1990. "Using vanishing points for camera calibration," International Journal of Computer Vision 4(2), 127-140.

[39] Kanatani, K., 1992. "Statistical analysis of focal-length calibration using vanishing points," IEEE Transactions on Robotics and Automation 8(6), 767-775.

[40] Weiss, R.S., Nakatani, H., and Riseman, E.M.,1990. "An error analysis for surface orientation from vanishing points," IEEE Transactions on Pattern Analysis Machine Intelligence 12(12), 1179-1185.

[41] Easy Camera Calibration Tool, 2001, Microsoft Corporation, Redmond, WA.

(http://research.microsoft.com/en-us/downloads/7e9de40f-06db-452c-a0f2-4fabb4f20f52/)

[42] Bouguet, J., Camera Calibration Toolbox for Matlab, Computer Vision Research Group, California Institute of Technology, Pasadena, CA.
(http://www.vision.caltech.edu/bouguetj/calib_doc/)
(http://robots.stanford.edu/cs223b04/JeanYvesCalib/index.html#links)

[43] Heikkila, J., 1996. Camera Calibration Toolbox for Matlab, University of Oulu, Finland.
(http://www.ee.oulu.fi/~jth/calibr/)

[44] Willson, R. and Tsai, R.Y.,1995. Camera Calibration Software, Carnegie Mellon University, Pittsburgh, PA.
(http://www.cs.cmu.edu/afs/cs.cmu.edu/user/rgw/www/TsaiCode.html)

[45] Echigo, T.,1990. "A camera calibration technique using three sets of parallel lines," Machine Vision and Applications 3(3), 159-167.

[46] Shabana, A.A., 2005. Dynamics of Multibody Systems, 3rd Edition, Cambridge University Press, New York.

[47] Bauchau, O.A., 2010. Flexible Multibody Dynamics, Springer Verlag, New York.

[48] Huygens, C., 1690. Traité de la Lumière, Pieter van der Aa, Leyden, Netherlands.

[49] Jenkins, F.A. and White, H.E., 1976. Fundamentals of Optics, 4th edition, McGraw-Hill, New York.

[50] Larmore, L., 1965. Introduction to Photographic Principles, 2nd edition, Dover Publications, New York, pp. 161-166.

[51] Jacobson, R.E., Ray, S.F., Atteridge, G.G., and Axford, N.R., 2000. "The geometry of image formation," The Manual of Photography: Photographic and Digital Imaging, 9th edition, Focal Press, Oxford.

[52] Ray, S.F., 2002. Applied Photographic Optics, 3rd edition, Focal Press, Oxford, pp. 217-220.

[53] TV Transmission Standard, U.S. Federal Communication Commission Rules and Regulations Part 73.

[54] "Rec. ITU-R BT.709: Parameter Values for the HDTV Standards for Production and International ProgrammeExchange," Radiocommunication Sector, International Telecommunication Union, 2002.

[55] "Rec. ITU-R BT.601: Encoding Parameters of Digital Television for Studios," Radiocommunication Sector, International Telecommunication Union, 1994.

[56] Bretzner, L. and Lindeberg, T., 1998. "Feature tracking with automatic selection of spatial scales," Computer Vision and Image Understanding 71(3), 385-392.

[57] Lindeberg, T., 1998. "Feature detection with automatic scale selection," International Journal of Computer Vision 30(2), 77-116.

[58] Parker, J.R., 2010. Algorithms for Image Processing and Computer Vision, 2nd Edition, Wiley Publishing, pp. 26-29.

[59] Nakamura, J., 2005. Image Sensors and Signal Processing for Digital Still Cameras, CRC Press.

[60] Johnson, J.B., 1927. "Thermal agitation of electricity in conductors," Proceedings of the American Physical Society: Minutes of the Philadelphia Meeting, Phys. Rev. 29, pp. 367-368.

[61] Johnson, J.B., 1928. "Thermal agitation of electricity in conductors – details of the experiment," Phys. Rev. 32, p. 97.

[62] Nyquist, H., 1928. "Thermal agitation of electric charge in conductors – the theory," Phys. Rev. 32, p. 110.

[63] Mancini, R., 2002. "Op amps for everyone," Application Notes, Texas Instruments, p. 148.

[64] Ohta, J., 2008. Smart CMOS Image Sensors and Applications, CRC Press.

[65] MacDonald, L., 2006. Digital Heritage. Butterworth-Heinemann.

[66] Shapiro, L.G. and Stockman, G.C., 2001. Computer Vision, Prentice-Hall.

[67] Boncelet, C., 2005. "Image noise models," in Handbook of Image and Video Processing by A.C. Bovik, Academic Press.

[68] Covington, M.A., 2007. Digital SLR Astrophotography, Cambridge University Press.

[69] Hartley, R. and Zisserman, A., 2003. Multiple View Geometry in Computer Vision, Cambridge University Press, pp. 155–157.

[70] Guenther, R., 1990. Modern Optics, John Wiley & Sons, New York, p. 130.

[71] "Comparison of optical aberrations," in Edmund Optics, retrieved on March 26, 2012.

[72] Debevec, P., 1996. Modeling and Rendering Architecture from Photographs, PhD thesis, p. 28.

[73] Horenstein, H., 2005. Black and White Photography: A Basic Manual, p. 55.

[74] Koehler, H., 1949. "Grundsaetzliches zum Fernrohrsehen," Deutsche Optische Wochenschrift 35(6), 41.

[75] Weng, J., Cohen, P., and Herniou, M., 1992. "Camera calibration with distortion models and accuracy evaluation," IEEE Transactions Pattern Analysis and Machine Intelligence 14(10), 965-980.

[76] Euler, L., 1776. "Formulae generales pro translatione quacunque corporum rigidorum," Novi Commentarii academiae scientiarum Petropolitanae 20. pp. 189–207.

[77] Kane, T.R., Likins, P.W., and Levinson, D.A., 1983. Spacecraft Dynamics, McGraw-Hill, New York.

[78] Shuster, M.D., 1993. "A survey of attitude representations," Journal of Astronautical Sciences 41(4), 439–517.

[79] Marandi, S.R., and Modi, V.J., 1987. "A preferred coordinate system and the associated orientation representation in attitude dynamics," Acta Astronautica 15, 833-843.

[80] Hartley, R., 1995. "In defense of the 8-point algorithm," Proceedings of the 5th International Conference on Computer Vision, pp. 1064–1070.

[81] Björck, A., 1996. Numerical Methods for Least Squares Problems, SIAM, Philadelphia. pp. 341-342.

[82] Cauchy, A., 1847. Méthode générale pour la résolution des systèmes d'équations simultanées. pp. 536–538.

[83] Levenberg, K., 1944. "A method for the solution of certain non-linear problems in least squares," Quarterly of Applied Mathematics 2, 164-168.

[84] Marquardt, D.W., 1963. "An algorithm for least-squares estimation of nonlinear parameters," Journal of the Society for Industrial and Applied Mathematics 11(2), 431-441.

[85] Celis, M., Dennis, J.E., and Tapia, R.A., 1985. "A trust region strategy for nonlinear equality constrained optimization," SIAM, Philadelphia, pp. 71-82.

[86] Conrady, A.B., 1919. "Decentering lens systems," Monthly notices of the Royal Astronomical Society 79, 384-390.

[87] Brown, D.C., 1966. "Decentering distortion of lenses," Photometric Engineering 32(3), 444-462.

[88] Faig, W., 1975. "Calibration of close-range photogrammetry systems: Mathematical formulation," Photogrammetric Engineering and Remote Sensing 41(12), 1479-1486.

[89] Manual of Photogrammetry, 4th Edition, American Society of Photogrammetry, 1980.

[90] Wei, G. and Ma, S., 1994. "Implicit and explicit camera calibration: Theory and experiments," IEEE Transactions on Pattern Analysis and Machine Intelligence 16(5), 469-480.

[91] Heikkila, J. and Silven, O., 1997. "A four-step camera calibration procedure with implicit image correction," Proceedings of IEEE Computer Vision and Pattern Recognition, pp. 1106-1112.

[92] Hughes, C., Glavin, M., Jones, E., and Denny, P., 2008. "Review of geometric distortion compensation in fish-eye cameras," IET Irish Signals and Systems Conference (ISSC), NUI Galway, pp. 162-167.

[93] Candocia, F.M., 2006. "A scale-preserving lens distortion model and its application to image registration," Proceedings of the 2006 Florida Conference in Recent Advances in Robotics 1, pp. 1-6.

[94] Mallon, J. and Whelan, P., 2004. "Precise radial un-distortion of images," Proceedings of the 17th International Conference on Pattern Recognition 1, pp. 18-21.

[95] "Electronic still picture imaging — Removable memory — Part 2: TIFF/EP image data format," ISO 12234-2:2001.

[96] "Graphic technology — Prepress digital data exchange — Tag image file format for image technology (TIFF/IT)," ISO 12639:1998.

[97] "Information technology -- Digital compression and coding of continuous-tone still images: Requirements and guidelines," ISO/IEC 10918-1:1994.

[98] "Information technology -- Digital compression and coding of continuous-tone still images: Compliance testing," ISO/IEC 10918-2:1995.

[99] "Information technology -- Digital compression and coding of continuous-tone still images: Extensions," ISO/IEC 10918-3:1997.

[100] "Information technology -- Digital compression and coding of continuous-tone still images: Registration of JPEG profiles, SPIFF profiles, SPIFF tags, SPIFF colour spaces, APPn markers, SPIFF compression types and Registration Authorities (REGAUT)," ISO/IEC 10918-4:1999.

[101] "Information technology -- Digital compression and coding of continuous-tone still images: JPEG File Interchange Format (JFIF)," ISO/IEC FDIS 10918-5.

[102] Brownlow, K., 1980. "Silent films: What was the right speed?" Sight & Sound 49(3), 164-167.

[103] "Recommendation ITU-R BT.472-3: Video-frequency characteristics of a television system to be used for the international exchange of programmes between countries that have adopted 625-line colour or monochrome systems," Radiocommunication Sector, International Telecommunication Union, 1990.

[104] Edgerton, G., 2009. The Columbia History of American Television, Columbia University Press, pp. 51-52.

[105] Schwarz, H., Rutishauser, H., and Stiefel, E., 1973. Numerical Analysis of Symmetric Matrices. Prentice-Hall, p. 117.

[106] Strang, G., 1988. Linear Algebra and its Applications, Harcourt, p. 337.

[107] Cajori, F., 1899. A History of Physics in its Elementary Branches, including the Evolution of Physical Laboratories, MacMillan Company, New York.

[108] Feynman, R., 1963. Lectures in Physics, Vol, 1, p. 30-1, Addison Wesley Publishing Company, Reading, Mass.

[109] Zhang, Z., Deriche, R., Faugeras, O., and Luong, Q.T., 1995. "A robust technique for matching two uncalibrated images through the recovery of the epipolar geometry," Artificial Intelligence Journal 78, 87-119.

[110] Dufournaud, Y., Schmid, C., and Horaud, R., 2004. "Image matching with scale adjustment," Computer Vision and Image Understanding 93(2), 175-194.

[111] Kanazawa, Y. and Kanatani, K., 2002."Robust image matching under a large disparity," Proceedings of the Workshop on Science of Computer Vision , Okayama, Japan, pp. 46-52.

[112] Peterson, B., 2010. Understanding Exposure: How to Shoot Great Photographs with Any Camera, 3rd Edition, Watson-Guptill Publications Inc., New York.

[113] Parker, J.R., 2011. Algorithms for Image Processing and Computer Vision, 2nd Ed., Wiley Publishing, Indianapolis, IN (http://www.wiley.com/go/jrparker).

Appendix A: Singular Value Decomposition in Image Processing

The two eigenvalues and eigenvectors of the 2×2 Harris matrix in (2.3.7) can be easily calculated, but their relations to corner points and edges need some derivations and discussions. The method of singular value decomposition is very useful for this purpose.

By singular value decomposition, the 2×2 Harris matrix can be decomposed into

$$[G]=[U][S][V]^T \tag{A.1}$$

where $[U]$ and $[V]$ are real unitary matrices, and $[S]$ is a real diagonal matrix with diagonal singular values λ_1 and λ_2 ($\lambda_1 \geq \lambda_2$). Because $[G]$ is symmetric, we have

$$[G][G]^T=[G]^T[G]\equiv[\tilde{G}] \tag{A.2}$$

Because $[U]$ and $[V]$ are unitary matrices, it follows from (A.1) and (A.2) that

$$\begin{aligned}[G]^T[G]&=[V][S]^T[U]^T[U][S][V]^T=[V][S]^T[U]^{-1}[U][S][V]^{-1}\\&=[V]([S]^T[S])[V]^{-1}\end{aligned} \tag{A.3}$$

and hence

$$[\tilde{G}][V]=[V]([S]^T[S]) \tag{A.4}$$

The explicit form of (A.4) is given by

$$\begin{bmatrix}\tilde{G}_{11} & \tilde{G}_{12}\\ \tilde{G}_{21} & \tilde{G}_{22}\end{bmatrix}\begin{bmatrix}v_{11} & v_{12}\\ v_{21} & v_{22}\end{bmatrix}=\begin{bmatrix}v_{11} & v_{12}\\ v_{21} & v_{22}\end{bmatrix}\begin{bmatrix}\lambda_1^2 & 0\\ 0 & \lambda_2^2\end{bmatrix} \tag{A.5}$$

Hence, we obtain

$$\begin{aligned}\begin{bmatrix}\tilde{G}_{11} & \tilde{G}_{12}\\ \tilde{G}_{21} & \tilde{G}_{22}\end{bmatrix}\begin{Bmatrix}v_{11}\\ v_{21}\end{Bmatrix}&=\lambda_1^2\begin{Bmatrix}v_{11}\\ v_{21}\end{Bmatrix}\\ \begin{bmatrix}\tilde{G}_{11} & \tilde{G}_{12}\\ \tilde{G}_{21} & \tilde{G}_{22}\end{bmatrix}\begin{Bmatrix}v_{12}\\ v_{22}\end{Bmatrix}&=\lambda_2^2\begin{Bmatrix}v_{12}\\ v_{22}\end{Bmatrix}\end{aligned} \tag{A.6}$$

In other words, λ_1^2 and λ_2^2 are eigenvalues of $[\tilde{G}]$, and the columns of $[V]$ are eigenvectors of $[\tilde{G}]$.

Similar to (A.3), we have

$$\begin{aligned}[G][G]^T &= [U][S][V]^T[V][S]^T[U]^T = [U][S][V]^{-1}[V][S]^T[U]^{-1} \\ &= [U]([S][S]^T)[U]^{-1}\end{aligned} \tag{A.7}$$

Hence, λ_1^2 and λ_2^2 are eigenvalues of $[\tilde{G}]$, and the columns of $[U]$ are also eigenvectors of $[\tilde{G}]$. In other words, we have

$$[U] = [V] \tag{A.8}$$

and

$$[G] = [U][S][V]^T = [U][S][U]^T = [U][S][U]^{-1} \Rightarrow [G][U] = [U][S] \tag{A.9}$$

Hence, we conclude that λ_1 and λ_2 are eigenvalues of $[G]$, and the columns of $[U](=[V])$ are eigenvectors of $[G]$.

According to the principal axis theorem [105,106], λ_1 and $\{v_{11}, v_{21}\}$ represent the greatest changing rate of grey values and its direction, as shown next. It follows from (2.3.6) and (2.3.7) that

$$S_{SD} = \{x \quad y\}[G]\begin{Bmatrix} x \\ y \end{Bmatrix} = \left\langle I_x^2 \right\rangle x^2 + 2\left\langle I_x I_y \right\rangle xy + \left\langle I_y^2 \right\rangle y^2 \tag{A.10}$$

Also, it follows from (A.9) and (A.10) that

$$S_{SD} = \{\tilde{x} \quad \tilde{y}\}[S]\begin{Bmatrix} \tilde{x} \\ \tilde{y} \end{Bmatrix}, \quad \begin{Bmatrix} \tilde{x} \\ \tilde{y} \end{Bmatrix} \equiv [U]^T \begin{Bmatrix} x \\ y \end{Bmatrix} \tag{A.11}$$

In other words, the $[U]$ transforms S_{SD} into an ellipse with λ_1 and λ_2 being the major and minor axes in the direction of $\tilde{x}$ and $\tilde{y}$ axes. In corner detection, $\lambda_1 >> \lambda_2 \approx 0$

indicates that the point is on an edge, $\lambda_1 \approx \lambda_2 \approx 0$ indicates that the point is on a flat area, and $\lambda_1 \approx \lambda_2 >> 0$ indicates that the point is a corner point.

Appendix B: Solving $[M]\{X\}=\{0\}$ by Least-Squares Fitting

Equations (3.1.15) and (3.1.23) for camera calibration and (4.1.7) for measurement applications are all in the form $[M]\{X\}=\{0\}$, where $[M]$ is an $m\times n$ matrix with $m>n$ and $\{X\}$ is an $n\times 1$ unknown vector. Because there are more equations than unknown variables, we show here how to obtain the answer with the minimal error by least-squares fitting and why $\{X\}$ corresponds to the column of the right singular matrix of $[M]$ with the smallest singular value.

By singular value decomposition, one can obtain $[M]=[U][D][V]^T$, where $[U]$ is an $m\times m$ orthogonal matrix, $[V]$ is an $n\times n$ orthogonal matrix, and $[D]$ is an $m\times n$ diagonal matrix with diagonal elements $D_{11}\geq D_{22}\geq\cdots\geq D_{kk}\geq 0$ $(k\leq n)$. Here, D_{ii} $(i=1,...,k)$ are singular values of $[M]$, and $k=n$ if all equations are linearly independent. Because $[U]$ and $[V]$ are orthogonal matrices, we have $[U]^{-1}=[U]^T$ and $[V]^{-1}=[V]^T$. Hence, one can show that

$$\begin{aligned}&[M]^T[M]\{X\}=\{0\}\\&[M]^T[M]=[V][D]^T[U]^T[U][D][V]^T\\&\qquad=[V][D]^T[D][V]^T=[V][Q][V]^T,\ [Q]\equiv[D]^T[D]\end{aligned}\tag{B.1}$$

Similar to (A.3), because $[M]^T[M][V]=[V][Q]$, diagonals of $[Q]$ are eigenvalues of $[M]^T[M]$ and columns of $[V]$ are eigenvectors of $[M]^T[M]$. Hence, the solution of $\{X\}$ can be factored into $\{X\}=a_1\{v_1\}+\cdots+a_k\{v_k\}$, where $\{v_i\}$ is the ith column of $[V]$ and $a_i(i=1,...,k)$ are unknown constants. Hence, we have

$$\begin{aligned}\{0\} &= [M]^T[M]\{X\} = \underset{n\times n}{[V]}\underset{n\times n}{[Q]}\underset{n\times n}{[V]^T}\underset{n\times 1}{\{X\}} \\ &= a_1[V][Q][V]^T\{v_1\} + \cdots + a_k[V][Q][V]^T\{v_k\}\end{aligned} \tag{B.2}$$

Because $\{v_j\}(j = 1,...,k)$ are linearly independent unit vectors and are orthogonal to each other,

$$\begin{aligned}[V][Q][V]^T\{v_j\} &= [V][Q]\{0,\cdots,0,\| v_j \|^2, 0,\cdots,0\}^T \\ &= [V]\{0,\cdots,0, Q_{jj}\| v_j \|^2, 0,\cdots,0\}^T \\ &= Q_{jj}\| v_j \|^2 \{v_j\} \\ &= D_{jj}^2\{v_j\}\end{aligned} \tag{B.3}$$

Substituting (B.3) into (B.2) yields

$$[M]^T[M]\{X\} = a_1D_{11}^2\{v_1\} + \cdots + a_kD_{kk}^2\{v_k\} \approx \{0\} \tag{B.4}$$

The optimal solution of $\{X\}$ should correspond to the minimal length of the left-hand vector. Since $\{v_j\}(j = 1,...,k)$ are linearly independent,

$$\begin{aligned}&\| a_1D_{11}^2\{v_1\} + \cdots + a_kD_{kk}^2\{v_k\} \|^2 \\ &= a_1^2D_{11}^4\| v_1 \|^2 + \cdots + a_k^2D_{kk}^4\| v_k \|^2 \\ &= a_1^2D_{11}^4 + \cdots + a_k^2D_{kk}^4\end{aligned} \tag{B.5}$$

Because $a_i(i = 1,...,k)$ are arbitrary unknown constants, we have $(a_1^2D_{11}^4 + \cdots + a_k^2D_{kk}^4)_{smallest} = a_k^2D_{kk}^4$ with D_{kk}^2 being the smallest eigenvalue of $[M]^T[M]$. Hence, the solution of $\{X\}$ corresponds to the eigenvector of $[M]^T[M]$ with the smallest eigenvalue, and it is also the column of the right singular matrix $[V]$ of $[M]$ with the smallest singular value.

Appendix C: Transformation of Cross Product

$$([M]\{a\})\times([M]\{b\}) = |[M]|[M]^{-T}(\{a\}\times\{b\})$$

Here we prove the transformation of cross product shown in (3.3.5). For a 3×3 matrix $[M]$, we define

$$[M] \equiv \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} \equiv \begin{bmatrix} \{M_1\}^T \\ \{M_2\}^T \\ \{M_3\}^T \end{bmatrix} \tag{C.1}$$

where $\{M_1\}$, $\{M_2\}$ and $\{M_3\}$ are the three rows of $[M]$.The inverse matrix of $[M]$ is

$$[M]^{-1} = \frac{adj([M])}{|[M]|} \tag{C.2}$$

where $|[M]|$ is the determinant of $[M]$. $adj([M])$ is the adjoint matrix of $[M]$, which is defined as

$$adj([M]) \equiv [C]^T \tag{C.3}$$

$[C]$ is the matrix of cofactors given by

$$[C] \equiv \begin{bmatrix} \begin{vmatrix} M_{22} & M_{23} \\ M_{32} & M_{33} \end{vmatrix} & \begin{vmatrix} M_{23} & M_{21} \\ M_{33} & M_{31} \end{vmatrix} & \begin{vmatrix} M_{21} & M_{22} \\ M_{31} & M_{32} \end{vmatrix} \\ \begin{vmatrix} M_{32} & M_{33} \\ M_{12} & M_{13} \end{vmatrix} & \begin{vmatrix} M_{11} & M_{13} \\ M_{31} & A_{33} \end{vmatrix} & \begin{vmatrix} M_{31} & M_{32} \\ M_{11} & M_{12} \end{vmatrix} \\ \begin{vmatrix} M_{12} & M_{13} \\ M_{22} & M_{23} \end{vmatrix} & \begin{vmatrix} M_{13} & M_{11} \\ M_{23} & M_{21} \end{vmatrix} & \begin{vmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{vmatrix} \end{bmatrix} \equiv \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \tag{C.4}$$

For the cross product of two vectors, we have

$$\{a\}\times\{b\} = \begin{vmatrix} \mathbf{i}_1 & \mathbf{i}_2 & \mathbf{i}_3 \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{vmatrix} = \begin{Bmatrix} a_2b_3 - b_2a_3 \\ a_3b_1 - b_3a_1 \\ a_1b_2 - b_1a_2 \end{Bmatrix} \tag{C.5a}$$

$$
\begin{aligned}
([M]\{a\})\times([M]\{b\}) &= \begin{vmatrix} \mathbf{i}_1 & \mathbf{i}_2 & \mathbf{i}_3 \\ \{M_1\}^T\{a\} & \{M_2\}^T\{a\} & \{M_3\}^T\{a\} \\ \{M_1\}^T\{b\} & \{M_2\}^T\{b\} & \{M_3\}^T\{b\} \end{vmatrix} \\
&= \begin{Bmatrix} \{M_2\}^T\{a\}\{M_3\}^T\{b\} - \{M_2\}^T\{b\}\{M_3\}^T\{a\} \\ \{M_3\}^T\{a\}\{M_1\}^T\{b\} - \{M_3\}^T\{b\}\{M_1\}^T\{a\} \\ \{M_1\}^T\{a\}\{M_2\}^T\{b\} - \{M_1\}^T\{b\}\{M_2\}^T\{a\} \end{Bmatrix}
\end{aligned}
\tag{C.5b}
$$

For the component along $\mathbf{i}_1$, we have

$$
\begin{aligned}
&\{M_2\}^T\{a\}\{M_3\}^T\{b\} - \{M_2\}^T\{b\}\{M_3\}^T\{a\} \\
&= \left(M_{21}a_1 + M_{22}a_2 + M_{23}a_3\right)\left(M_{31}b_1 + M_{32}b_2 + M_{33}b_3\right) \\
&-\left(M_{21}b_1 + M_{22}b_2 + M_{23}b_3\right)\left(M_{31}a_1 + M_{32}a_2 + M_{33}a_3\right) \\
&= \cancel{M_{21}M_{31}a_1b_1 - M_{21}M_{31}b_1a_1} + M_{21}M_{32}a_1b_2 - M_{21}M_{32}b_1a_2 + M_{21}M_{33}a_1b_3 - M_{21}M_{33}b_1a_3 \\
&+M_{22}M_{31}a_2b_1 - M_{22}M_{31}b_2a_1 + \cancel{M_{22}M_{32}a_2b_2 - M_{22}M_{32}b_2a_2} + M_{22}M_{33}a_2b_3 - M_{22}M_{33}b_2a_3 \\
&+M_{23}M_{31}a_3b_1 - M_{23}M_{31}b_3a_1 + M_{23}M_{32}a_3b_2 - M_{23}M_{32}b_3a_2 + \cancel{M_{23}M_{33}a_3b_3 - M_{23}M_{33}b_3a_3} \\
&= \begin{vmatrix} M_{21} & M_{22} \\ M_{31} & M_{32} \end{vmatrix}\left(a_1b_2 - b_1a_2\right) + \begin{vmatrix} M_{23} & M_{21} \\ M_{33} & M_{31} \end{vmatrix}\left(a_3b_1 - b_3a_1\right) + \begin{vmatrix} M_{22} & M_{23} \\ M_{32} & M_{33} \end{vmatrix}\left(a_2b_3 - b_2a_3\right)
\end{aligned}
\tag{C.6}
$$

From (C.4) and (C.6), we obtain

$$
\begin{aligned}
&\{M_2\}^T\{a\}\{M_3\}^T\{b\} - \{M_2\}^T\{b\}\{M_3\}^T\{a\} \\
&= C_{13}\left(a_1b_2 - b_1a_2\right) + C_{12}\left(a_3b_1 - b_3a_1\right) + C_{11}\left(a_2b_3 - b_2a_3\right) \\
&= C_{11}\left(a_2b_3 - b_2a_3\right) + C_{12}\left(a_3b_1 - b_3a_1\right) + C_{13}\left(a_1b_2 - b_1a_2\right)
\end{aligned}
\tag{C.7a}
$$

Similarly, for the components along $\mathbf{i}_2$ and $\mathbf{i}_3$ we have

$$
\begin{aligned}
&\{M_3\}^T\{a\}\{M_1\}^T\{b\} - \{M_3\}^T\{b\}\{M_1\}^T\{a\} \\
&= C_{21}\left(a_2b_3 - b_2a_3\right) + C_{22}\left(a_3b_1 - b_3a_1\right) + C_{23}\left(a_1b_2 - b_1a_2\right)
\end{aligned}
\tag{C.7b}
$$

$$
\begin{aligned}
&\{M_1\}^T\{a\}\{M_2\}^T\{b\} - \{M_1\}^T\{b\}\{M_2\}^T\{a\} \\
&= C_{31}\left(a_2b_3 - b_2a_3\right) + C_{32}\left(a_3b_1 - b_3a_1\right) + C_{33}\left(a_1b_2 - b_1a_2\right)
\end{aligned}
\tag{C.7c}
$$

It follows from (C.5b), (C.7a)-(C.7b), (C.4) and (C.5a) that

$$
\begin{aligned}
([M]\{a\})\times([M]\{b\}) &= \begin{Bmatrix} C_{11}(a_2b_3-b_2a_3)+C_{12}(a_3b_1-b_3a_1)+C_{13}(a_1b_2-b_1a_2) \\ C_{21}(a_2b_3-b_2a_3)+C_{22}(a_3b_1-b_3a_1)+C_{23}(a_1b_2-b_1a_2) \\ C_{31}(a_2b_3-b_2a_3)+C_{32}(a_3b_1-b_3a_1)+C_{33}(a_1b_2-b_1a_2) \end{Bmatrix} \\
&= \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \begin{Bmatrix} a_2b_3-b_2a_3 \\ a_3b_1-b_3a_1 \\ a_1b_2-b_1a_2 \end{Bmatrix} \\
&= [C](\{a\}\times\{b\})
\end{aligned}
\tag{C.8}
$$

From (C.2) and (C.3), we obtain

$$
[C] = \left(adj[M]\right)^T = \left(\left|[M]\right|[M]^{-1}\right)^T = \left|[M]\right|[M]^{-T} \tag{C.9}
$$

Substituting (C.9) into (C.8) yields

$$
([M]\{a\})\times([M]\{b\}) = \left|[M]\right|[M]^{-T}(\{a\}\times\{b\}) \tag{C.10}
$$