3D Laser Scanner Development and Analysis

2008-2013

JunJie, Liu

Supervisor:

Dr.Yonghuai Liu
Department of Computer Science
Aberystwyth University, Ceredigion SY23 3DB, UK
yyl@aber.ac.uk

October,24 2013

Abstract

This PhD project is a collaboration between Smart Light Devices, Ltd. in Aberdeen and Aberystwyth University on the development of such 3D laser scanners with an ultimate aim to inspect the underwater oil and gas pipes or structure. At the end of this project, a workable and full functional 3D laser scanner is to be developed. This PhD project puts a particular emphasis on the engineering and implementation of the scanner according to real applications' requirements. Our 3D laser scanner is based on the principle of triangulation and its high accuracy over a short range scanning. Accurate 3D data can be obtained from a triangle between the scanner, camera lens, laser source, and the object being scanned. Once the distance between the scanner camera lens and laser source (stereo baseline) is known and the laser projection angle can be measured by the goniometer, all the X, Y, Z coordinates of the object surface can be obtained through trigonometry.

This 3D laser scanner development involves a lot of issues and tasks including image noise removal, laser peak detection, corner detection, camera calibration and 3D reconstruction. These issues and tasks have been addressed, analysed and improved during the PhD period. Firstly, the Sparse Code Shrinkage (SCS) image de-noise is implemented, since it is one of the most suitable de-noising methods for our laser images with dark background and white laser stripe. Secondly, there are already plenty of methods for corner and laser peak detection, it is necessary to compare and evaluate which is the most suitable for our 3D laser scanner. Thus, comparative studies are carried out and their results are presented in this thesis. Thirdly, our scanner is based on laser triangulation, in this case, laser projection angle α and baseline distance D from the centre of the camera lens to laser source plays a crucial role in 3D reconstruction. However, these two parameters are hard to measure directly, and there are no particular tools designed for this purpose. Thus, a new approach is proposed in this thesis to estimate them which combines camera calibration results with the precise linear stage. Fourthly, it is very expensive to customize an accurate positional pattern for camera calibration, due to budget limit, this pattern is printed by a printer or even painted on a paper or white board which is inaccurate and contains errors in absolute distance and location. An iterative camera calibration method is proposed. It can compensate up to 10% error and the calibration parameters remain stable. Finally, in the underwater applications, the light travel angle is changed from water to air which makes the normal calibration method less accurate. Hence, a new approach is proposed to compensate between the estimate and real distance in 3D reconstruction with normal calibration parameters. Experimental results show the proposed methods reduce the distance error in 3D down to $\pm 0.2mm$ underwater. Overall, the developed scanning systems have been successfully applied in several real scanning and 3D modelling projects such as mooring chain, underwater pipeline surface and reducer. Positive feedback has been received from these projects, the scanning results satisfy the resolution and accuracy requirements.

Acknowledgements

This research project would not have been possible without the support of many people. I wish to express my gratitude to my supervisor, Dr. Liu who was abundantly helpful and offered invaluable assistance, support and guidance. Deepest gratitude is also due to the members of the supervisory committee, Prof. Reyer and Dr. Ala without their knowledge and assistance this study would not have been successful. Special thanks also to my friends, especially group members: Anthony Jakas for sharing the literature and invaluable assistance. Not forgetting to his best friends who have always been there. I would also like to convey thanks to the Smart Light Device, Ltd. for providing the financial means and laboratory facilities. I wish to express my love and gratitude to my beloved family for their understanding and endless love, through the duration of my studies.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In 2007, Smart Light Devices, Ltd. and Aberystwyth University agreed to collaborate on a project to develop 3D laser scanners for underwater oil and gas scanning. The project intended to build a 3D laser scanner for oil and gas industry inspection, such as pipeline internal/external surface survey, object structure analyses. The final objective is after three years' research, there will be a workable, robust 3D scanning system developed and tested. It includes the 3D laser scanner components' and software kits development. Any issues, problems, and technology used or invented during the development will be counted into part of PhD research. The great advantages of this project are that all the used and developed methods, theories and programs can be applied in real world applications directly. Most of the problems, bugs and issues in the real applications need to be solved immediately. It is dual learning processes between research and application. In addition, working in the real applications gives us great opportunities to deal with the challenges from real tests such as the cable is broken, the computer gets crashed, software contain bugs which may not happen in the lab. Comparing with the research in a lab, there are more factors to be considered. The complex real applications always come with different and special requirements. The scanner and software needs to be adapted to fit those requirements. Moreover, the system running speed, data storage and environmental issues also need to be taken into account. Overall, this is not only pure research but also implementation engineering and optimization.

In underwater inspection, there are needs for sensors that can determine the structure of an asset. These measurements are usually represented as a 3D digital point cloud and used for engineering calculations, maintenance planning, and ROV (Remotely operated underwater vehicle[1]) localization. The most prevalent technology in this field is sonar.

---

[1] A remotely operated vehicle (ROV) is a tethered underwater vehicle. They are widely used in deep water industries like offshore hydrocarbon extraction. Sometimes, an ROV is called a remotely operated underwater vehicle to distinguish it from remote control vehicles operating on land or in the air. dROVs are unoccupied and highly maneuverable vessels operated by a person aboard. They are linked to the ship by a tether (sometimes referred to as an umbilical cable), a group of cables that carry electrical power, video and data signals back and forth between the operator and the vehicle. High power applications will often use hydraulics in addition to electrical cabling. Most ROVs are equipped with at least a video camera and lights. Additional equipment is commonly added to expand the vehicle's capabilities. These

High frequency multi-beam sonar systems are capable of capturing complete 3D digital point-cloud representations of underwater environments. There are however physical limitations to the resolution capability of these technologies for understanding small and important features of structures as cracks and erosion in concrete structures, welds and dents in metallic marine infrastructure. However, 3D laser scanners are able to capture details of underwater assets that are previously unobtainable.

There is huge market and demand for underwater 3d scanning particularly in oil and gas pipe inspection, survey and science research. Underwater laser scanners provide several clear advantages over sonar systems at relatively short ranges. The measurement resolution is multiple orders of magnitude higher, enabling very dense point clouds. Measurements made from dense laser point clouds will in general be much more accurate than those taken from relatively sparse sonar point clouds. The typical sonar system has a scan range from $0 \sim 1m$ with about 5cm resolution. With laser scanner, the range can be from $0 \sim 3m$ with up to $< 1mm$ resolution. Additionally, laser systems are not affected by confined spaces prone to acoustic echoes. The ideal application for an underwater laser scanner involves short-range, high-detail measurements of specific locations on an asset. Such applications include damage assessment, archaeological documentation, and quality control/inspection.

1.1 Scanning procedure

In this section, a brief description of the entire 3D scanning procedure is presented. Fig.1.1 shows the most used scanner designs with dual laser sources. The camera is placed in the middle of the block and the two laser sources are placed at two sides with a projection angle about 60°. During the scanning, one side of the laser will be turned on and sweep the object surface. For the reason of the camera field of view (FOV), the laser projection angle and object's shape (like a sphere), the scanner will not be able to cover the entire surface. Therefore, by controlling the linear stage to travel in reverse direction with other laser, it is able to sweep the object's rest surface. These two different 3D scans can be merged thereafter so that a larger coverage of the object can be built.

Base on this design, Fig.1.2 shows the main steps of the entire process in 3D scanning. Steps $1 \sim 5$ are to calibrate the whole system and obtain the important parameters for Step 8, such as camera focal length and laser projection angle. They are normally done in the lab before the equipment is sent out. Step 9 may not be necessary in most cases which depends on the client's requirement. In some of the steps such as $2, 5, 7$, they involve image acquisition and processing. Feature detection and noise removal methods are applied to these steps. The details can be found in the later chapters.

Basically, the scanner is made of an optical camera and laser stripe generators, plus positioning tools. For instance, the linear stage holds and moves the scanner to acquiring 3D coordinates in one axis. Development of this 3D scanning system requires a wide range of knowledge including camera calibration, image noise removal, feature detection,

---

may include sonar, magnetometers, a still camera, a manipulator or cutting arm, water samplers, and instruments that measure water clarity, light penetration and temperature.

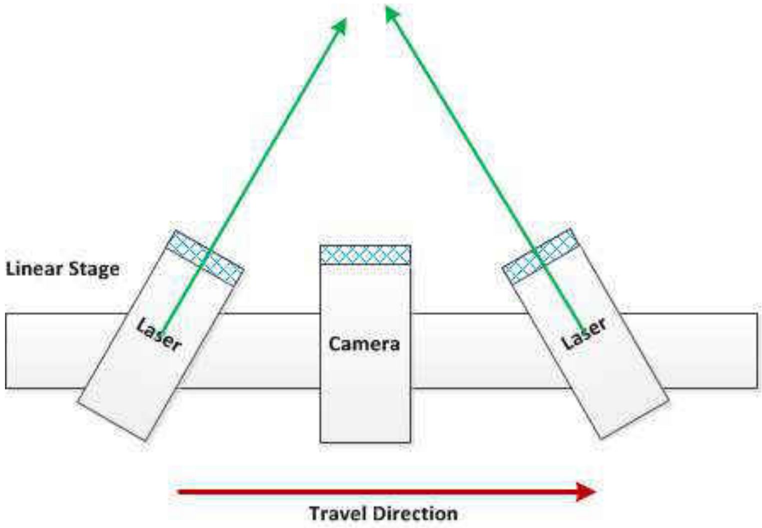


Figure 1.1: Dual lasers 3D scanner design

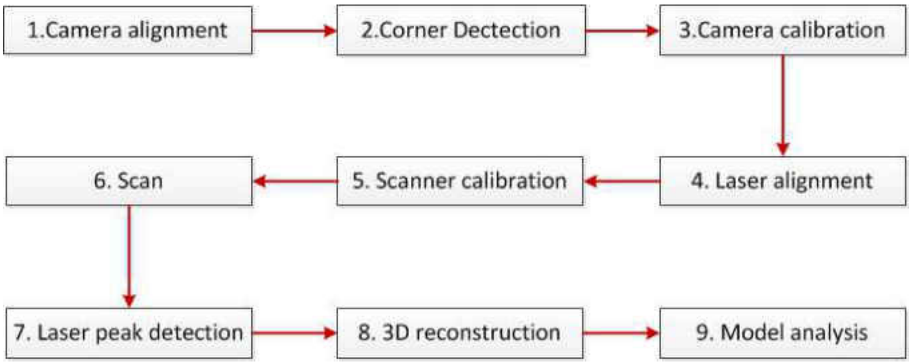


Figure 1.2: The 3D scanning main steps

data acquisition and communication between hardware and software, 3D reconstruction, data storage and visualization, data analysis and classification. Fig.1.3 shows the diagram that contains all tasks, techniques, and topics which have to be achieved, used, developed or relevant to this project.

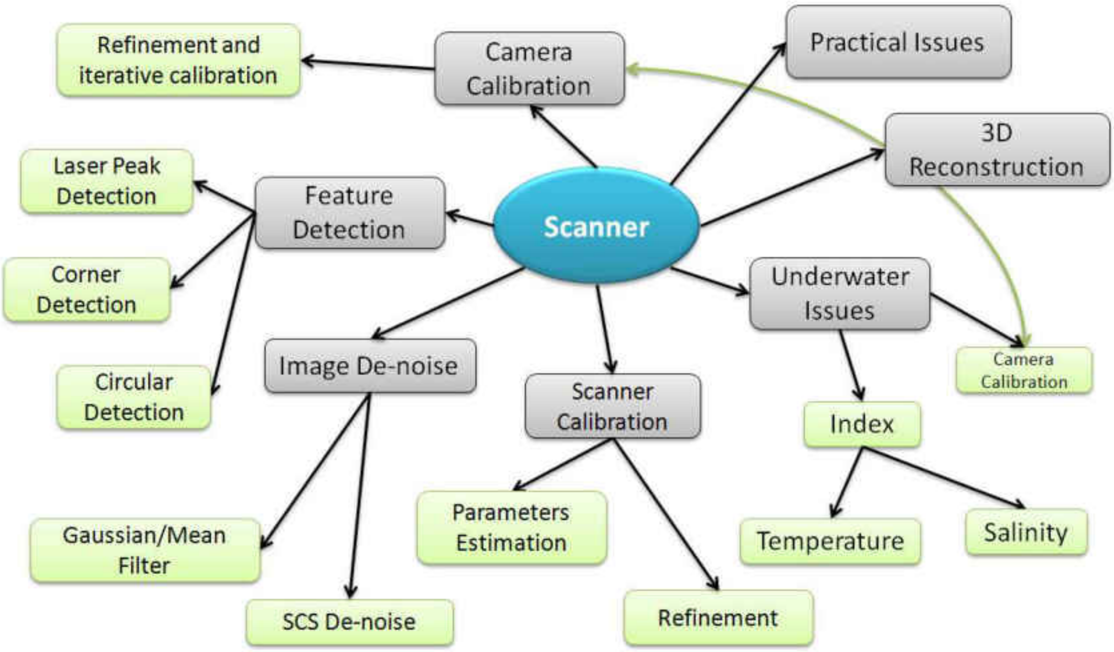


Figure 1.3: The decomposition of the PhD project

Feature detection is a very important preprocessing step for all those relative tasks, including laser peak detection, corner/circle detection which are used for camera calibration. By comparing the well-known existing corner detection methods [49], the most suitable detector for our applications can be selected like in [11]. In the same way, by evaluating the existing laser peak detection methods, the FIR [24] convolution method is the most accurate and adaptable, even though the laser stripe is usually saturated due to the highly reflective surface and the higher laser power.

Camera calibration is another important step: it relates the 3D real world to the projected 2D images and allows the reconstruction of the 3D world from its captured 2D images. To this end, it has two main purposes: (1) find out the intrinsic/extrinsic parameters for achieving geometry information [83, 3]; (2) remove the distortion which comes from the nature of camera's lens. The distortion factors are able to model and remove the lens distortion. These calibration parameters are crucial parts of any 3D laser scanner. The distortion factors can be used for image distortion removal and intrinsic/extrinsic parameters are used for 3D reconstruction in our triangulation based 3D scanner, which is explained in Chapter 5. Even though camera calibration is somewhat mature after dozens of years' research and development, a novel iterative refinement based method is proposed in this thesis for more accurate camera calibration results. The main factor that affects the calibration result is the control point detection and actual calibration pattern design. There are plenty of feature detection methods that can be used nowadays and give us reasonably precise detection. However, the accurate locations of such patterns require precise engineering and are costly to build [16]. Hence, the inaccurate 3D location of control points is considered in the proposed method, which can compensate up to 10% absolute error in x, y, z coordinates based on our currently 3D scanner's setting.

National Research Council of Canada is among the earliest institutes for the development of the triangulation based laser scanning technologies in 1978[2] [54]. The triangulation based laser scanning technologies are relatively easy to implement and cheap to run. Our laser scanner based on this principle as shown in Fig.5.1. The crucial task is to estimate the laser stripe projection angle α and the distance D between the centre of the lens to the laser. These two parameters and the position of laser dots in the image fully determine the shape and size of the triangle and give the location of the laser hit point on the object surface in 3D space. However, the distance D from the centre of the lens to laser and the projection angle α are almost impossible to measure directly, since they are difficult to define and locate inside two physical objects and determine a reference for the measurement of the projection angle. A new approach makes full use of the precise linear stage and the image formation geometry proposed to estimate these two parameters in Chapter 5.

It is a great challenge to apply our scanner in underwater environments. The main difficulty is the water index that is different from air, thus the light bends an angle when it travels from the air into the water as Fig.1.4 shows. Various index values will lead to various bending angles. Temperature, salinity and soluble matter inside (sea/river) water also affect its index and thus light refraction. This kind of refraction affects the entire system including camera calibration and triangulation. More details about refraction, light bending can be found in Chapter 5.4. However, it is still possible to implement a 3D scanner in the underwater environment. Experimental Chapter 6 and real projects Chapter 7.2 show some examples and analysis when our system applied to underwater applications. Overall, this thesis is about the research, developing and building the 3D laser scanning system. All the methods, technologies and related researches serve this ultimate goal. Thus, the project puts special emphasis on the pragmatic development of the scanner, rather than on the advance of the techniques involved. A reliable, robust and ready for real application 3D laser scanner was built after this PhD project. The next chapter will show a brief survey of existing scanning technologies as compared to our system.

1.2 Thesis structure

This thesis is structured and organized as follows based on the scanning procedure:

---

[2] Access on 4th July, 2011, http://en.wikipedia.org/wiki/3D_scanner#Triangulation

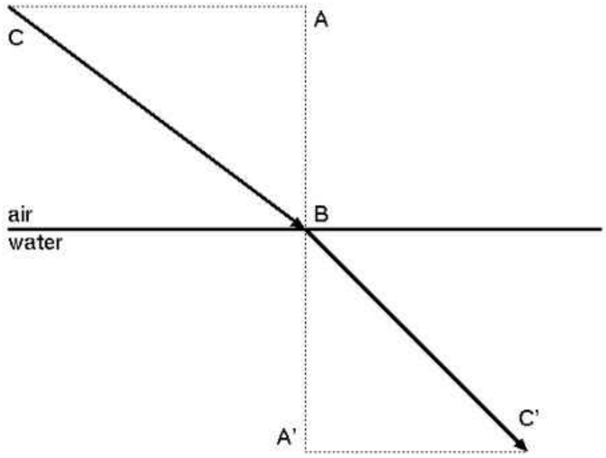


Figure 1.4: Light entering water from the air.

Chapter 3 evaluates the most widely used and latest methods in three image processing tasks: Noise removal, corner and laser peak detection. (1) In the noise removal, the method "Sparse Code Shrinkage image denoise" (SCS will be the abbreviation in the following content) is evaluated. It has better performance to remove Gaussian noise than other commonly used noise removal techniques such as Gaussian, median and Wienier filters. (2) A comparative study is carried out over the most commonly used corner detection methods including Harris, SUSAN corner detectors. The corner detection is an important pre-step for camera calibration and those detected corners are used as 2D input dataset. By the evaluation, the most accurate, robust and resistant to noise method will be selected. (3) To be able to recover the geometry information from the triangulation based laser scanner, the laser peak location must be detected precisely and efficiently. It is necessary to compare these peak detection methods and know how they affect the final 3D model accuracy.

Chapter 4 firstly describes the camera calibration concepts including perspective projection model, distortion model and optimization. After that, a new camera distortion model is proposed. It has the ability to model the lens distortion but not require any prior knowledge about what kind of distortion of the lens was subject to. In addition, an iterative camera calibration method is proposed to compensate the inaccurate 3D object. Even though the 3D object contains up to 10% absolute distance error, the calibration parameters remain stable.

Chapter 5 proposes a new approach to estimate the laser projection angle and baseline length, which is the distance from the laser to the camera's CCD, lens. These two parameters play important roles in our triangulation based 3D laser scanner. They are used to recover the 3D geometry information from the 2D laser stripe images. However, the baseline length is almost impossible to measure directly because of their physical limitation, in which case, firstly, it is difficult, if not impossible, to localize the centres of these components; secondly, even though they can be localized, they lie inside and thus are not accessible for direct distance measurement. The proposed method manipulates other available components like the precise linear stage and related geometry for the estimation of the parameters of interest.

Chapter 6 presents the experimental results based on the techniques and methods described and proposed in the previous chapters. Individual experiment is carried out for each task as follows:

1. Image noise removal
2. Corner detection
3. Laser peak detection
4. Lens distortion modelling
5. Camera calibration result refinement

The performance of the various methods and their impact on the 3D reconstruction accuracy will be evaluated. Finally, all the techniques and methods will bring into the system together for overall evaluation.

Chapter 7 describes all the main scanning projects we did during the PhD study period. The algorithms and methods were applied to applications in the real world. Thus, the reliability, speed, or any other concerns need to be counted in. Compared with the experiments just in the lab, sometimes the real world problems are unpredictable. Thus, this real world scenario provides us with a great opportunity to test our products, algorithms, methods and programs. The feedbacks from the scanning jobs are one of the critical parts in our research.

Chapter 2

Literature Review

2.1 3D Laser scanner's history

At a conference in 1959, Gordon Gould published the term LASER in the paper "The LASER", Light Amplification by Stimulated Emission of Radiation [25]. Gould's linguistic intention was using the "Laser" word particle as a suffix - to accurately denote the spectrum of the light emitted by the LASER device; thus x-rays: xaser, ultraviolet: uvaser, et cetera; none established itself as a discrete term, although "raser" was briefly popular for denoting radio-frequency-emitting devices.

The word laser started as an acronym for "light amplification by stimulated emission of radiation"; in modern usage "light" broadly denotes electromagnetic radiation of any frequency, not only visible light, hence infrared laser, ultraviolet laser, X-ray laser, and so on. Because the microwave predecessor of the laser, the maser, was developed first, devices of this sort operating at microwave and radio frequencies are referred to as "masers" rather than "microwave lasers" or "radio lasers". In the early technical literature, especially at Bell Telephone Laboratories, the laser was called an optical maser; this term is now obsolete. [1]

Since the laser is invented, it was developed for 3D imaging and came into the market very quickly, below is the brief Summary of Milestones, [2]

- 1968 - Academics from Cambridge University produced the prototype of a laser-based digitizing scientific instrument called Sweepnik.
- 1969 - Laser-Scan founded at Madingley Road to produce Sweepniks.
- 1972 - First sale of Sweepnik by Laser-Scan.
- 1972 - Prototype HRD-1 large-screen display and plotter built.
- 1973 - First cartographic software written for the HRD-1, led to sales into early digital mapping.

---

[1] Access on 22nd July, 2011, http://en.wikipedia.org/wiki/Laser#cite_note-3

[2] Access on 22nd July, 2011, http://www.pghardy.net/lsl/lsl_history.html

After that, this laser 3D scanner is widely used, in the meanwhile, different types of 3D scanner were invented. A short survey is presented in the next section.

2.2 Existing 3D scanners survey

A survey of existing 3D scanning technologies is presented in this section. A 3D scanner is a device that samples data on the shape and possibly the appearance (i.e. color) of a real world object or environment (background). The collected data can then be used to construct digital, three dimensional models useful for a wide variety of applications. For example, these devices are used extensively by the entertainment industry in the production of movies and video games. Other common applications of this technology include industrial design, reverse engineering and prototyping, quality control/inspection and documentation of cultural artifacts. A well-established classification divides them into two types: contact and non-contact 3D scanners. Non-contact 3D scanners can be further divided into two main categories: active and passive scanners [3]. There are varieties of technologies that fall under each of these categories. Apparently, the process of contact scanning might modify or damage the scanned object. The consequence is very significant when scanning delicate or valuable objects. Hence, contact scanners have limited applications. In this survey, we focus on the non-contact 3D scanners. There are several techniques used in non-contact 3D scanners to collect data. The following are a few commonly used techniques:

2.2.1 Structured light scanner

Optical 3D measurement technology is one of the most effective methods to acquire object 3D information. It belongs to non-contact measurement with the advantages of non-contact to measure surface and high sampling density. Among the methods, the encoded structured light is widely used in such fields as 3D reconstruction and industrial measurement because of its advantages in high accuracy, high measuring speed, low cost and so on. There are two key steps in this system:

2.2.1.1 Calibration

Before doing the actual scanning, this projector-camera system needs to be precisely calibrated for a successful and accurate reconstruction. This is especially true for multi-view reconstructions, where a given surface point is reconstructed several times and the corresponding reconstructed points have to meet in 3D space. Papers [67, 15, 46] presented some of these projector-camera system calibration methods. In short, the projector can be modelled as an inverse camera (i.e., one in which light travels in the opposite direction as usual). Under this model, calibration proceeds in a similar manner as with cameras. Rather than photographing fixed checker boards, the known checker board patterns are projected and their distorted appearance is photographed when reflected from a diffuse

---

[3] `http://en.wikipedia.org/wiki/3D_scanner`

rigid object. This approach has the advantage of being a direct extension of Zhang's calibration algorithm for the cameras. As a result, much of the software can be shared between camera calibration and projector calibration.

2.2.1.2 Encoding

In the premise of acquiring camera calibration parameters such as focal length, another key problem with the encoded structured light method is to determine image sampling point, which determines the object sampling point, and encoding stripe region (namely projecting angle) in encoding patterns. Encoding methods can be classified into three categories: time encoding, space encoding and direct encoding with their unique merits and drawbacks. Existing time encoding method divides projecting angle by binary code or Gray code [82], that sometimes are combined with phase shift [26, 68] to subdivide projecting angle [72]. Fig.2.1 is one of the typical structured light scanner setups.

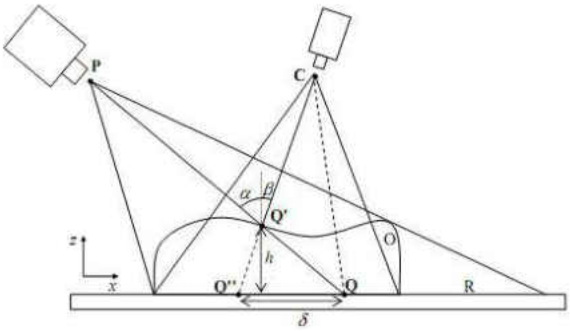


Figure 2.1: Geometry of structured light setup

To better understand light encoding, we can start from very simple "desktop scanner" originally proposed by Bouguet and Perona [6] as shown in Fig.2.2: (a) The scanning set up composed of five primary items: a digital camera, a point-like light source, a stick, two planar surfaces, and a calibration checkerboard (not shown). Note that the light source and camera must be separated so that cast shadow planes and camera rays do not meet at small incidence angles. (b) The stick is slowly waved in front of the point light to cast a planar shadow that translates from left to right in the scene. The position and orientation of the shadow plane, in the world coordinate system, are estimated by observing its position on the planar surfaces. After calibrating the camera, a 3D model can be recovered by triangulation of each optical ray by the shadow plane that first entered the corresponding scene point.

The primary benefit of introducing the projector is to eliminate the mechanical motion required in swept-plane scanning systems (e.g., laser striping or the desktop scanner[6]) and lower down the system cost and complexity. Assuming minimal lens distortion, the projector can be used to display a single column (or row) of white pixels translating against a black background; thus, 1024 (or 768) images would be required


(a) swept-plane scanning apparatus (b) frame from acquired video sequence

Figure 2.2: 3D photography using planar shadows

to assign the correspondences, between camera pixels and projector columns (or rows). After establishing the correspondences and calibrating the system, a 3D point cloud is reconstructed using familiar ray-plane triangulation. However, a simple swept-plane sequence does not fully exploit the projector. Since we are free to project arbitrary 24-bit colour images, one would expect that there exists a sequence of coded patterns, besides a simple translation of a single stripe, that allow the projector-camera correspondences to be assigned in relatively few frames. In general, the identity of each plane can be encoded spatially (i.e., within a single frame) or temporally (i.e., across multiple frames), or with a combination of both spatial and temporal encodings. There are benefits and drawbacks to each strategy. For instance, purely spatial encodings allow a single static pattern to be used for reconstruction, enabling dynamic scenes to be captured. Alternatively, purely temporal encodings are more likely to benefit from redundancy, reducing reconstruction artifacts. A comprehensive assessment of such codes is presented by Salvi [68]

2.2.2 Time-of-flight laser scanner

The time-of-flight 3D laser scanner is another active scanner using laser light to probe the subject. It is similar as a time-of-flight laser range finder. The laser range finder finds the distance of a surface by timing the round-trip time of a pulse of light. A laser is used to emit a pulse of light and the amount of time before the reflected light is seen by a detector is timed. Since the speed c of light is known, the round-trip time determines the travel distance of the light, which is twice the distance between the scanner and the surface. If t is the round-trip time, then the distance can be estimated as: $(c \cdot t)/2$. The accuracy of a time-of-flight 3D laser scanner depends on how precisely we can measure the time t. 3.3 picoseconds (approx.) is the time taken for light to travel 1 millimetre. The laser range finder only detects the distance of one point in its direction of view. Thus, the scanner scans in its entire field of view one point at a time by changing its direction of view in order to scan different points. The view direction of the laser range finder can be changed either by rotating the range finder itself, or by using a system of rotating mirrors. The latter method is commonly used because mirrors are much lighter and can thus be rotated much faster with greater accuracy. Typical time-of-flight 3D laser scanners can measure the distance of from 10,000 to 100,000 points every second.

Clearly, the advantage of this scanner is their capability of operating over very long distances on the order of kilometres. This scanner is thus suitable for scanning large structures like buildings or geographic features. However, compared with other types of scanners, it has disadvantages in accuracy. Due to the high speed of light, it is relatively difficult to time the round-trip time, leading the accuracy of the distance measurement to be relatively low, in the order of millimetres [9].

2.2.3 Triangulation based laser scanner

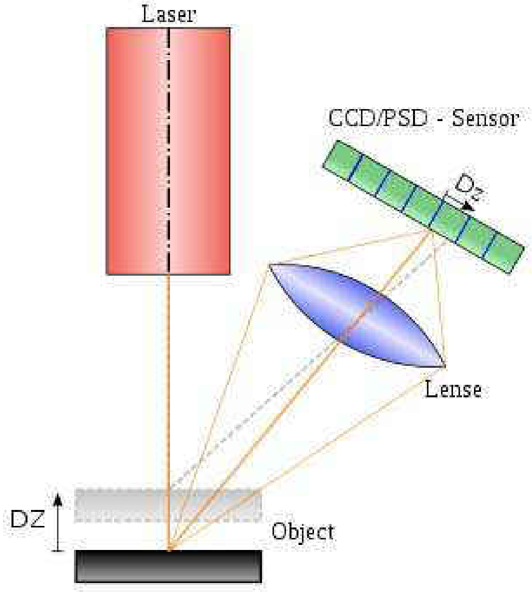


Figure 2.3: Laser-based optical triangulation (a)

The triangulation based 3D laser scanner is the most widely use technique in 3D data acquisition. Basically, it shines a laser spot on the subject and exploits a camera to look for the location of the laser dot. Depending on how far away the laser strikes a surface, the laser dot/stripe appears at different places in the camera's field of view. To recover the geometrical structure of visible surface of objects from 2D images, the laser-based optical triangulation method has been used in 3D scanning system. Fig.2.3 and Fig.2.4 depict the optical geometry of an optical probe (single 3D points) [3]. The laser stripe can be treated as many dots, each dot captured in the image can be reversed from 2D(x, y) point to 3D(X, Y, Z). The relationship among the parameters of interest in this optical probe can be found in the law of cosines. This laser-based triangulation model will be used in the following calculation. Knowing two angles (α, β) of the triangle relative to its base (baseline D) determines the dimension of this triangle.

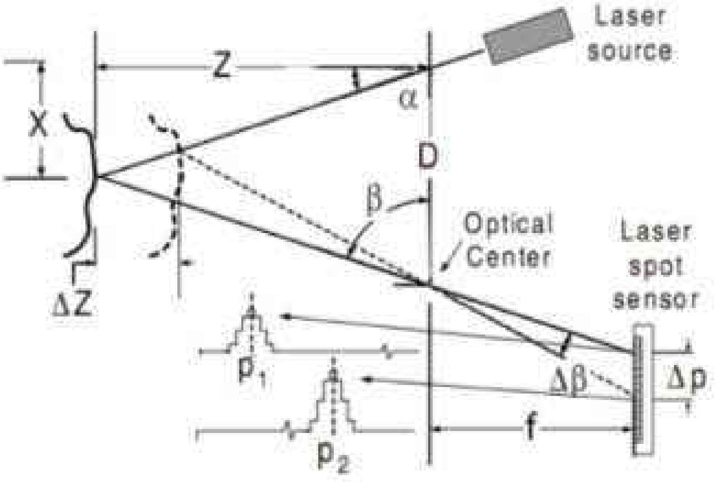


Figure 2.4: Laser-based optical triangulation (b)

2.2.4 Current air 3D scanning system

The evaluation of 3D (three dimensional) scanners is presented on paper [8] for obtaining 3D models or objects which can be used to build virtual reality systems. The evaluation methodology applies the MCDM (Multi Criteria Decision Making) concepts. It helps decision makers to select the best 3D scanner, based on several criteria through comparing the results obtained by applying three different MCDM methods. The 3D scanners evaluated in [8] are shown in Fig.2.5. They cover the three main categories mentioned above:

- Time of flight: Laser Scanner LS880
- Triangulation:
    - Laser based: Laser Scan Arm
    - Projector based: Star Cam FW-3, Atos I, Atos III Model Maker Z

The result shows that the best tool is Laser scanner LS880 from Faro Technologies and the second best is Laser ScanArm from the same company. However, this LS800 based on TOF method has some disadvantages. Firstly, it may generate huge amount of points and increase a lot of workload after scanning. Secondly, because of TOF, the farther the object is to the scanner, the better the result is. This brings the benefits of long distance object scanning since it is easier to measure the 3D depth with relatively longer light travel time. Therefore, for close range ($< 1m$) object scanning, it may not be the best. Laser ScanArm is more suitable for the close range scanning, the only disadvantage will be the error from mechanical arm, and the reading of movement of those arms may not be accurate enough.

Name	Figure	Name	Figure
Star Cam FW-3		Laser ScanArm	
Atos I		Laser Scanner LS880	
Atos III		ZEPHYR KZ 100	
ModelMaker Z			

Figure 2.5: Identification and selection of 3D scanners

2.2.5 Current underwater 3D scanning system

The previous sections show some examples of 3D scanners which is designed to work in air. For underwater 3D scanning, it does need some special design. Images taken from underwater scenes usually suffer from poor contrast. Water-induced contrast decay varies across the scene and is exponential in the depths of scene points, which prevents standard computer vision algorithms from operating properly. Compared with a 3D laser scanner in the air, underwater scanning is much more complicated due to the following factors:

1. The water/air index differences cause light refraction and thus change the entire perspective geometry of the camera as shown in Fig.1.4. The general camera calibration model cannot be applied. In addition, the light refraction affects the laser triangulation.

2. All the equipment needs to be waterproof. The camera and laser source should be able to fit into the cylinder or some waterproof container. At the same time, some of the equipment cannot be put in underwater such as: normal high accuracy linear/rotation stage. So far, we have not found any of these stages that can be operated underwater directly.

3. Due to the complex underwater circumstances, it is necessary to perform image restoration[76], enhancement and colour correction [64].

Fig.2.6 shows one of the underwater circumstances, the colour of the left image is far away from its reality on the right. Depending on the refraction index of water and absorption, the water colour can be green or yellow. Since our system uses a mono camera, this factor still reduces our laser intensity and affects the light capturing. Fig.2.7 shows such crinkle pattern. When the water is very calm, its undulations resemble weak positive and negative lenses, the negative ones diffusing the light, resulting in dark patches while the positive ones focusing the light into bright patches. This effect also causes the creative 'cathedral' rays, sometimes visible. The mud in water will be another circumstance as shown in Fig.2.8.

Figure 2.6: Underwater circumstances: Incorrect Colour

Figure 2.7: Underwater circumstances: Crinkle patterns

There are many factors to be considered and it is much more complicated to build a 3D scanner that can operate underwater. The scanning accuracy or resolution is lower than normal air 3D scanner. A lot of aspects can be improved and more tests need to be done. Furthermore, it is possible to extend the usage of the underwater laser scanning in any transparent medium like glass. In the 3D scanner market, only a few companies

Figure 2.8: Underwater circumstances: The mud

are managing to provide 3D underwater scanning services like 2G underwater scanner[4] which is shown in Fig.2.9. Fig.2.10 shows USL-100 mounted on an underwater robot (ROV). Here is how it works: the head of the sensor sweeps a line of laser light over a surface and obtains measurements of hundreds of points along the line at each position, capturing thousands of points per second. It is designed for scanning cylinder shaped object like oil and gas pipeline. However, this rotation design has its defect to scan non-cylinder shape object. When the motor rotates, the laser may not illuminate the entire surface. Fig.2.11 explains this rotation scanner operating inside a pipeline. The motor rotates at every angle but when the laser hits the flat surface, there are still some blind areas. Camera cannot receive any reflected signal since the laser cannot illuminate in such areas.

OptoPig [5] in Fig.2.12 is also designed for pipeline scanning with much larger diameter. However, this equipment is much more expensive to rent, not many customers can afford.

2.3 Summary

The 3D scanner technologies are already developed and implemented in many fields but here is a huge market for the automatic inspection of the oil and gas pipes. The current techniques are mainly based on human intervention. It is lack of accuracy and speed, and operates on a small scale. There is a large room for modern techniques to play in this aspect but only a few of them that can be applied in the underwater working environment, such as sub-sea inspection, oil industry. Clearly, the time-of-flight scanner is not accurate enough and the type of TOF camera is much more expensive than a normal camera. Structure light scanner system may be able to put into the water but the light source from the projector is much less compact compared with the laser source. In addition, the projector is more expensive and hard to build the waterproof

---

[4] `http://www.2grobotics.com`

[5] `http://www.neo.no/products/optopig.html`

Figure 2.9: The ULS-100

Figure 2.10: ULS-100 attached to ROV

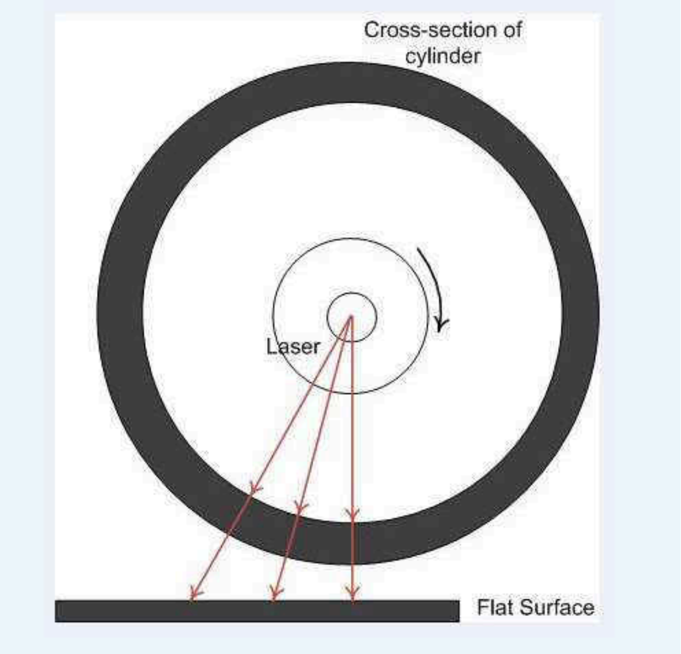


Figure 2.11: The rotational scan by ULS-100

Figure 2.12: The OptoPig

container. Camera-laser triangulation system is the best option to implement in the underwater environment. Compared wither others, the camera laser triangulation system has advantages of low cost, being much easier to develop and apply in different areas. The key to reversing the 2D points in images to 3D is to have the known 3D scene reference location or the moving step in XYZ direction. There are some hand-held 3D scanner such as David 3D scanner[6] and Handyscan 3D scanners[7]. Some of these 3D scanners using known 3D points as references to obtain the laser stripe location in a 3D scene, while some of the others are sticking lots of white dots on the object's surface to do the 3D model assignment and registration. Such methods also use some of the structured light scanner to register 3D data from different views of scanning. Both of them have limitation applying to underwater. It is easy to see that putting 3D reference points or dots is not practical when applying underwater. In our cases, the 3D location is achieved from the precise movement of linear/rotation stage. To sum up, using the camera-laser triangulation based method would be the best to apply in underwater scanning so far.

---

[6] `http://www.david-laserscanner.com`

[7] `http://www.creaform3d.com/en/handyscan3d/products/default.aspx`

Chapter 3

Image processing

Image processing is the pre-stage essential step during the entire scanning. It includes image filtering, noise removal, feature point detection, geometric image transformations or object tracking, etc. The main image processing in this project includes image noise removal and feature detection. Noise removal is the most pre-stage step and applied in all the image processing tasks. Feature detections include laser peak and corner detection. Those detected corners are used in camera calibration to compute camera geometric, optical characteristics (intrinsic parameters) and lens distortion factors. With such information, it is able to reconstruct the 2D laser peaks to 3D points.

This chapter is structured and organized as follows based on the order of these image processing methods applied in the scanning procedure:

The first section is the earliest step from image processing tasks: noise removal. In this section, a Sparse Code Shrinkage denoise method is introduced. A comparative study is carried on using natural and 3D scanner's images in later Chapter.6.1.

The second section is one of the feature detection: corner detection. Chessboard pattern is easy to make and detect. These detected corner location is used for camera calibration such as [79, 83]. A comparative study in later Chapter.6.2 is to find out the best from existing corner detection methods.

In the last section, brief outlines of existing peak detection methods are presented. Laser peak detection is used to extract the peak of the laser stripe in the row of the captured image. Precise peak location in image $P(x_i, y_i)$ is a crucial factor to recover the 3D information in our triangulation based scanner as shown in Fig.2.3 and Fig.2.4. Detailed comparative study is carried on using artificial and 3D scanner's images in later Chapter.6.3

3.1 Sparse Code Shrinkage image denoise

Image noise is the digital equivalent of film grain for analogue cameras. For digital images, this noise appears as random speckles on an otherwise smooth surface and can significantly degrade image quality. Although noise often detracts from an image, it is sometimes desirable since it can add an old-fashioned, grainy look to an image that is reminiscent of early film. Some noise can also increase the apparent sharpness of an image. Noise increases with the sensitivity setting in the camera, the length of the exposure, temperature, and even varies amongst different camera models.

Noise is an inherent property of digital imaging sensors. The laws of physics make it impossible to eliminate noise, and they force a trade-off between noise levels and other properties like sensor size or sensitivity. Photons, for instance, arrive at random intervals, so the simple task of counting them during an exposure– which is the basic function of a pixel in a sensor – is subject to sampling error. When the exposure is shortened or the pixel size is reduced, there are fewer photons to ”average out” the sampling error, so the noise increases relative to the signal.

The image in a digital camera comes from a non-digital component: the CCD or CMOS image sensor. An image sensor is typically comprised of a matrix of light sensors. A light sensor can be thought of as simply a device that converts light into an electric charge. Each square of the image sensor matrix is a photo site, usually with one light sensor ’painted’ on it. A photo site generally corresponds to one pixel in your digital image. When light (photons) strikes the image sensor, electrons are produced. These ”photoelectrons” give rise to analog signals and then converted into digital pixels by an Analog to Digital (A/D) Converter.

Causes of Noise There are a number of sources of noise contamination. Heat generated might free electrons from the image sensor itself, thus contaminating the ”true” photoelectrons. These ”thermal electrons” give rise to a form of noise called thermal noise or dark current.

Another type of noise is more akin to the ’grain’ obtained by using a high ISO film. When we use a higher ISO, we are amplifying the signal we receive from the light photons. Unfortunately, as we amplify the signal, we also amplify the background electrical noise that is present in any electrical system.

In low light environment, there is not enough light to have proper exposure. The longer we allow the image sensor to collect the weak signal, the more background electrical noise it collects. In this case, the background electrical noise may be higher than the signal.

Laser image is a special case with white laser stripe and dark background. Thus, white noise such as dots, lumps have heavier impact to laser image than the normal colour image. Gaussian blur/smooth[58] is a good method to reduce noise. Basically, it is a smooth filter applied to the image. It may decrease the laser stripe intensity or shift the peak location. In our case, most of the noise is coming from the electronic noise. Fig.3.1 is one of the captured laser stripes with a lot of white dots around it. The closer the camera to the electronic equipment, like motor and transformers, the heavier the noise is. The noise appears randomly, they are flying everywhere in the image. While our scanner works in dark environment, the noise becomes much more sensitive and heavier. Sometimes, the scanner needs to continue capturing for a long period and the camera’s CMOS sensor will generate much heavier noise in high temperature after long operation. Because the CCD and CMOS’ manufacturing technically differ, the CMOS sensor would produce much more noise than CCD sensor in normal temperature (e.g.

25°). If the temperature is high, the CCD sensor actually generating more noise than the CMOS sensor. Therefore, Sparse Code Shrinkage image denoise [35] approach is applied in our scanner to remove noise. It is an extended method based on the independent component analysis (ICA) [37]. The detailed description and experimental results can be found in the next sub-section.

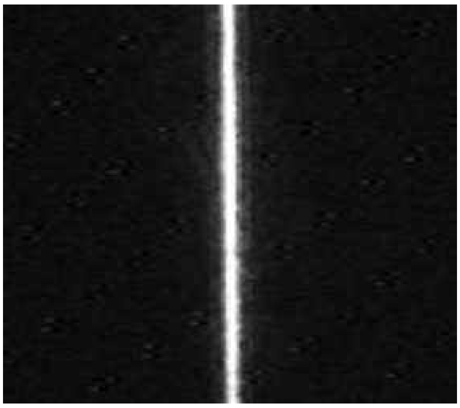


Figure 3.1: An example of noisy laser stripes

Before any mathematical method applies to remove or lower down the noise level, there some tips are given below:

1. Use the high noise resistant cables and connect them properly.
2. Use the proper camera filter to get rid of the light source we do not want.
3. Put the camera as far as possible away from the equipments such as motor, electricity transformer, etc.
4. Make sure that the laser signal is higher than the background/noise. If the laser intensity is too low or the background is too bright, then it is impossible to separate them.

To remove the heavy noise in our 3D scanning system or any other image equipment, the "Sparse Code Shrinkage image denoise" in [33] was implemented. Sparse coding is a coding of the data such that for any given input vector, only a few of the components of the code will be significantly active. The type of coding was originally motivated by neurophysiological considerations, and simulations of the methods based on sparse coding, producing results match measurements from real neurons quite well [60, 61, 33]. In theory, sparse coding is a method for finding a neural network representation of multidimensional data in which only a small number of neurons are significant at the same time. The idea constitutes the rationale behind Sparse Code Shrinkage (SCS) is to use a basis that is more suitable for image denoise. Considering a random vector X, which have been corrupted by additive Gaussian white. Estimating a sparse code for it will give us a transform, which will concentrate much of the energy of the sample vectors into only a few components. If our transform is orthogonal, then the noise will still be Gaussian and white in the transform basis. Thus, we have effectively separated the signal from the noise, and by setting small components to zero, we will remove a large part of the noise but only a little of the signal [32].

Denote by $x = (x_1, x_2, ...x_n)^T$ the observed and n-dimensional random vector that is input to a neural network, and by $s = (s_1, s_2, ...s_n)^T$ is the vector of the transformed component variables, which are the n linear outputs of the network. Denoting further the weight vectors of the neurons by w_i, $i = 1, ...n$, and by $W = (w_1, ...w_n)^T$ the weight matrix whose rows are the weight vectors, the linear relationship is given by

$$s = Wx \tag{3.1.1}$$

We assume here that the number of sparse components, i.e., the number of neurons, equals the number of observed variables, but this need not be the case in general. Sparse coding can now be formulated as a search for weight matrix W such that the components s_i are as "sparse" as possible. A zero-mean random variable s_i is called sparse when it has a probability density function with a peak at zero, and heavy tails; for all practical purposes, sparsity is equivalent to super gaussianity [1] or positive kurtosis[2][42].

3.1.1 Independent Component Analysis(ICA) and Image Data

Sparse coding is closely related to independent component analysis (ICA) [39]. The term "independent" is associated with statistical properties of the components s_i. The model can be defined as follows:

$$x = As. \tag{3.1.2}$$

where x and s are column vectors with entries x_i and s_i respectively, and A is the mixing matrix. j^{th} entry for the vector x can be expressed as:

$$x_j = a_{j1}s_1 + a_{j2}s_2 + ... + a_{jm}s_m \tag{3.1.3}$$

In this model, s_i are latent variables and the mixing process or matrix is assumed to be unknown. From this perspective, ICA is closely related to the method so called Blind Source Separation (BSS). Inverting the relation, one obtains s from Equation 3.1.1: $s = Wx$, with W being the pseudo inverse of A. In this denoise method, ICA is used to extract features from data (mixing matrix) and its inverse W. This extraction procedure provides a more data-dependent representation of given images than any other transformation like principal component analysis (PCA). The main steps in the whole image denoise process are given in the following sections.

---

[1] Super-Gaussianity may be thought of as implying both a sharper peak and heavier tail than the corresponding Gaussian [62]

[2] In probability theory and statistics, kurtosis is a measure of the "peakedness" of the probability distribution of a real-valued random variable, although some sources are insistent that heavy tails, and not peakedness, is what is really being measured by kurtosis. [41] Higher kurtosis means more of the variance is the result of infrequent extreme deviations, as opposed to frequent modestly sized deviations.

3.1.2 Image data pre-processing

Before applying an ICA algorithm on the data, it is usually very useful to do some pre-processing. In this section, we discuss some pre-processing techniques that make the problem of ICA estimation simpler and better conditioned. The image data would be the vector from the original image and all the variables in the vector are randomly chosen. This vector is defined as X.

1. ***Local mean*** When ICA is applied to image data, it usually gives one component representing the local mean image intensity. This component normally has a distribution that is not sparse, often even sub-Gaussian. Thus, it must be treated separately from the other, super-Gaussian components. However, since the component generally has a large variance, it is relatively unaffected by the noise, and a simplification is to simply leave it alone. The most basic and necessary step is to center data X. For example, subtract local mean gray-scale value from each patch and normalize to zero mean and unit variance.

2. ***Normalizing the local variance*** Normalizing the input data to zero mean and unit variance will cause the component statistics to be a lot less super Gaussian than in the non-normalized case, with few components sparser than those given by the Laplace distribution. This can be done in both the estimation of parameters, and in the denoising procedure.

3. ***Whitening*** Another necessary pre-processing strategy in ICA is to first whiten the observed variables. This means that before applying the ICA method, we transform the images' vector X to be $\tilde{X}$ which is white. The whitening transformation is always possible. The popular method for whitening is to use the eigen-value decomposition (EVD) of the covariance matrix $E\{XX^T\} = EDE^T$ and E is the orthogonal matrix of eigenvectors of $E\{XX^T\}$ and D is the diagonal matrix of its eigenvalues. Thus, the whitening can be done by using Equations 3.1.4 and 3.1.5

$$\tilde{x} = ED^{-1/2}E^T x \tag{3.1.4}$$

where the matrix $D^{-1/2}$ is computed by a simple component-wise operation as

$$D^{-1/2} = diag(d_1^{-1/2}, ...diag(d_n^{-1/2})). \tag{3.1.5}$$

3.1.3 FastICA

In modeling image data for ICA, subimages of size $N \times N$, taken from an arbitrary sized image, is considered as the observations of the $nx1$, letting $n = N^2$, random vector x of Eq. 3.1.2. In this view, each x can be considered as a linear combination of the columns of the mixing matrix A, each of which is weighted by the independent components si. For instance, such a subimage would be obtained by superposing the patches of Fig 3.2, each patch being weighted by the corresponding independent component of the sparse code. Hence the patches of Fig 3.2 correspond to the columns of the mixing matrix and the independent components to sparse codes. In the ideal setup, the mixing matrix A is square, i.e. $n = m$ or there as many observed mixtures as the independent components, hence the relation in Eq.3.1.2 can be inverted to s=W.x where the matrix W is called the separating or demixing matrix. What ICA achieves is the estimation of the matrices A and W by just using the observed mixtures x that are indeed subimages written in the vector form. Once such a transformation pair is found, extraction of sparse codes is a simple matter of matrix-vector multiplication through s=W.x.[33].

The FastICA algorithm is a computationally highly efficient method for performing the estimation of ICA. It uses a fixed-point iteration scheme that has been found in independent experiments to be 10-100 times faster than conventional gradient descent methods for ICA. Another advantage of the FastICA algorithm is that it can be used to perform projection pursuit as well, thus providing a general-purpose data analysis method that can be used both in an exploratory fashion and for estimation of independent components (or sources)[34]. It can be derived as an approximate Newton iteration. The basic algorithm is as follows:

1. Choose an initial weight vector W
2. Iterate: $\mathbf{w}^{+} = E\{\mathbf{x}g(\mathbf{w}^T\mathbf{x})\} - E\{g'(\mathbf{w}^T\mathbf{x})\}\mathbf{w}$
3. Normalize: $\mathbf{w} = \mathbf{w}^{+}/\|\mathbf{w}^{+}\|$
4. If not converged, goes back to 2

Finally, the result of separating (weight) and mixing matrix have relationship $W^(-1) = A$, and matrix W should satisfy the orthogonality and done by3.1.6

$$W \leftarrow W(W^TW)^{-1/2} \tag{3.1.6}$$

The natural scene images 3.2 and 3.3 are used to test the FastICA outlined above.

Figure 3.2: ICA test image 1

Figure 3.3: ICA test image 2

After having applied FastICA, the matrix W should look like Fig.3.4. If the test data are totally about synthetic scene such as the buildings, then the matrix would look like Fig.3.5. From these two images, much stronger continuous lines and edges can be seen in the synthetic scene, as expected.

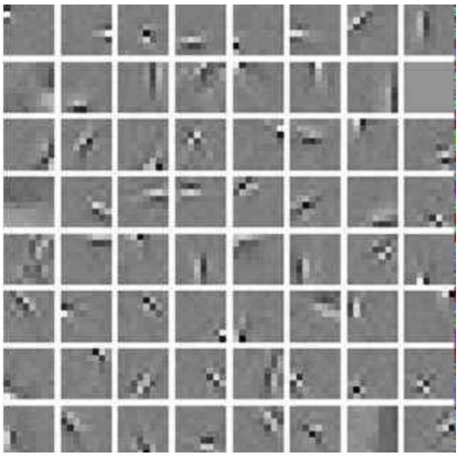


Figure 3.4: The orthogonal sparsifying matrix of natural scene (8 by 8 patch)

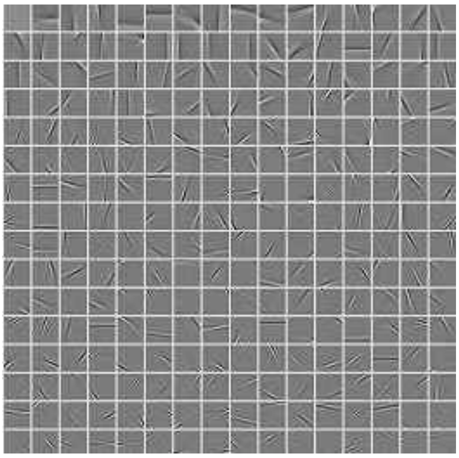


Figure 3.5: The orthogonal sparsifying matrix of synthetic scene (16 by 16 patch)

3.1.3.1 Maximum likelihood de-noising of non-Gaussian random variables

In order to derive maximum-likelihood denoising of a non-Gaussian data corrupted by zero mean Gaussian noise with variance σ^2, the following additive model is considered

$$y = s + v \tag{3.1.7}$$

where y is the corrupted version of non-Gaussian random variable s and v is the noise: $E(v) = 0$ and $std(v) = \sigma$. We want to estimate the original signal s. Denoting by p the probability density of s, and by $f = -\log p$, it's the negative log-density, the maximum likelihood method gives the following estimator for s:

$$\hat{s} = \arg min_u \frac{1}{2\sigma^2}(y-u)^2 + f(u) \tag{3.1.8}$$

Assuming that f is strictly convex and differentiable, this minimization is equivalent to solving the following equation:

$$\frac{1}{\sigma^2}(\hat{s} - y) + f^{'}(\hat{s}) = 0 \tag{3.1.9}$$

or $\hat{s} = g(y)$ with $g^{-1}(u) = u + \sigma^2 f'(u)$

3.1.4 Sparse density models

To use the estimator defined by $\hat{s} = g(y)$, the densities of the s_i need to be modelled with a parameterization that is expressive enough. Sparse data are characterized as being mildly or strongly sparse. The general expressions for such models are given below. Note that the Laplacian density is somewhat between them as shown in Fig.3.6. The form of these models is called shrinkage function $g(u)$ corresponding to the above distributions in Fig.3.6.

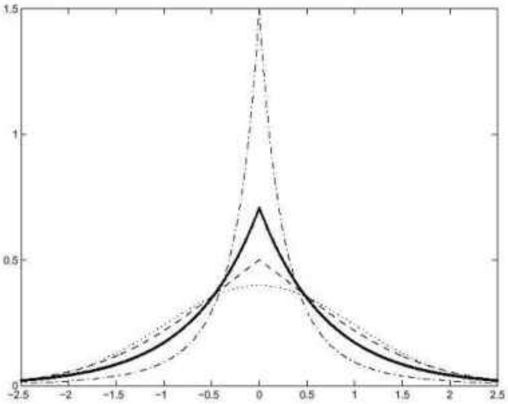


Figure 3.6: Plots of densities corresponding to different models of sparse components. Solid line: Laplace density. Dashed line: a typical moderately sparse(super-Gaussian) density. Dash-dotted line: a typical strongly sparse(super-Gaussian) density. Dotted line: Gaussian density for reference.[due to Hyvaarinen et al. 1999]

3.1.4.1 Mildly sparse densities

For mildly sparse densities, $g(u)$ can be represented as[36]:

$$g(u) = \frac{1}{1+\sigma^2 a} sign(u) max(0, \mid u \mid -b\sigma^2) \tag{3.1.10}$$

where

$$b = \frac{2p_s(0)E\{s^2\} - E\{|s|\}}{E\{s^2\} - [E\{|s|\}]^2} \tag{3.1.11}$$

$$a = \frac{1}{E\{s^2\}}[1 - E\{|s|\}b] \tag{3.1.12}$$

where p_s0 is the value of the density function of s at zero.

3.1.4.2 Strongly sparse densities

For the strongly sparse densities, $g(u)$ can be represented as

$$g(u) = \frac{1}{2d} sign(u) max(0, \frac{|u| - ad}{2} + \frac{1}{2}\sqrt{((|u| + ad)^2 - 4\sigma^2(\sigma + 3))} \tag{3.1.13}$$

where

$$d = \sqrt{E\{s^2\}} \tag{3.1.14}$$

$$\alpha = \frac{2 - k + \sqrt{k(k+4)}}{2k - 1} \quad with \quad k = d^2 p_s(0)^2 \tag{3.1.15}$$

$$a = \sqrt{\alpha(a+1)/2} \tag{3.1.16}$$

and σ^2 is the noise variance in all cases.

3.1.5 Sparse code denoise

The effects of the functions shown in Fig.3.7 is to reduce the absolute value of its argument by an amount determined by the noise level. Small arguments are set to zero. The overall effect is a reduction in Gaussian noise by restoring sparseness back. To determine which model to use, it is suggested in the limiting case of Laplacian density that the value $\sqrt{E\{s^2\}}p_s(0)$ happens to be $1/\sqrt{2}$. The model is mildly sparse below it and strongly sparse above it. However, experience shows that sparse code extracted from real image data happens to follow a strong sparsity.

If the noise variance σ^2 is unknown, one might estimate it, by multiplying by 0.6475 the mean absolute deviation of the y_i corresponding to the very sparsest s_i

The SCS technique is summarized as follows:

1. Build a noise-free set of data z that has the same statistical properties as the n-dimensional data x that we want to denoise.

2. Use data z to estimate the sparse coding transformation $W = W_{opt}$ by estimating and fractionalizing the ICA transform matrix W.

3. For every $i = 1, 2, \cdots, n$, estimate a density model for $s_i = w_i^T z$, using the model described above. Also, estimate the relevant parameters.

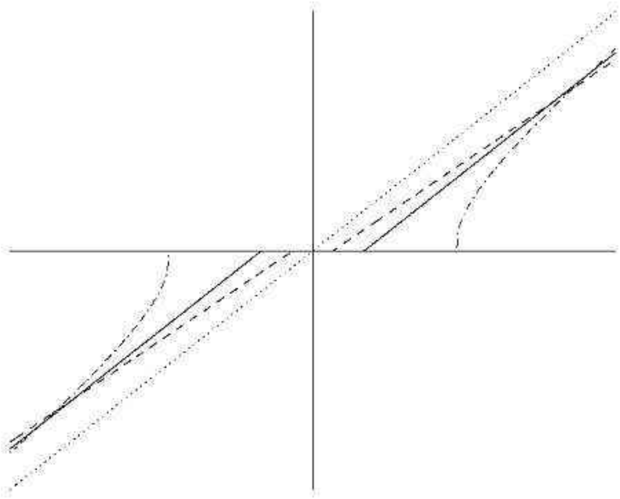


Figure 3.7: Plots of the shrinkage functions. Solid line: shrinkage corresponding to Laplace density. Dashed line: shrinkage corresponding to moderately sparse(super-Gaussian) density. Dash-dotted line: shrinkage corresponding to strongly sparse(super-Gaussian) density. Dotted line: y=x for reference. [due to Hyvaarinen et al. 1999]

4. For every noisy data sample $x(t)$, $t = 1, 2, \cdots, T$, which are samples of a noisy version x, compute the projections on the sparsifying basis:

$$y(t) = W\hat{x}(t) \tag{3.1.17}$$

5. Apply the shrinkage operator g_i corresponding to the density model of s_i to every component $y_i(t)$ of $y(t)$, for every t, obtaining:

$$\hat{s}_i(t) = g_i(y_i(t)); \tag{3.1.18}$$

6. Transform back to original variable to obtain estimates of the noise-free data $x(t)$

$$\hat{x}(t) = W^T\hat{s}(t) \tag{3.1.19}$$

3.2 Corner Detection

Interest point detection is widely used and as an early step in many computer vision tasks such as camera calibration that requires fast and efficient feature extraction and matching. Since the camera calibration is one of the important steps in our scanning process, it is necessary to evaluate and select which corner detection method is the best. In this section, alternative corner measures will be tested and a comparative evaluation of those different techniques is carried out.

3.2.1 Introduction

The use of interest points (corner detectors) to find corresponding points across multiple images and the corresponding points between an image and a real object is a key step in many image processing and computer vision applications. The most notable applications include image matching [40], real-time gesture recognition[45], mouth detection[48] and motion tracking and robot navigation. In the past decades, many interest point detectors developed, one of the earliest is that of Moravec [57], which measures intensity differences by shifting a small mask in eight principal directions. Harris and Stephens[27] later expanded the Moravec operator to include a function that enables intensity variation to be measured in any direction. Many different interest point detectors have been proposed with a wide range of definitions for what points in an image are interesting. Some detectors find points of high local symmetry, others find areas of highly varying texture, while others locate corner points. Corner points are interesting as they are formed from two or more edges and edges usually define the boundary between two different objects or parts of the same object.

There are a significant number of different approaches for detecting corners. Figure 3.8 shows a time-line of the corner detectors have been developed. There are three main approaches for detection of corners in gray scale images: edge-related, topology, and autocorrelation. Most of the corner detectors discussed here can be placed into one of these categories. However, it should be made clear that the classification is somewhat subjective as most of the detectors have multiple interpretations and thus could be classified into more than one category.

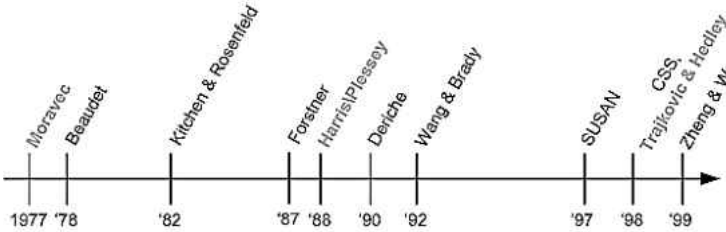


Figure 3.8: Time line of corner detectors

Although some work on comparing and evaluating interesting point detectors have published [70, 75, 84, 44], only [84, 44] provide the comparison of different cornerness measures. Most of them haven't drawn conclusion or evaluation to which cornerness measure is most appropriate in different situations. Though [44] provides a quantitative evaluation on [13, 14, 71, 27] but most of the comparable methods are the extensions based on [27]. In addition, in our 3D scanner, the corner detection result has significant effect on the camera calibration and 3D model's accuracy. Therefore, a comparative study on such kind of effect on camera calibration and 3D reconstruction of the latest corner detectors is necessary and carried out. In section 3.2.2, each of the corner detectors selected for the comparative study is outlined. Section 6.2 presents the evaluation results using both synthetic and real world datasets including our the samples from our underwater scanning system.

3.2.2 Outlines of corner detectors

In this section the corner detectors used in the comparative study in [27, 59, 73, 56] are outlined. These methods are historically significant, widely used, and well suited for a particular application (i.e. real-time).

3.2.2.1 Harris and Stephens / Plessey corner detection

Harris and Stephens improved Moravec's corner detector by considering the differential of the corner scores with respect to direction directly, instead of using shifted patches. This corner detector is an alternative approach used in feature detection. This corner detection defines a measure of corner strength $H(x, y)$, for each pixel(x,y) in the image through constructing an autocorrelation matrix M:

$$M = \begin{bmatrix} A & C \\ C & B \end{bmatrix} \tag{3.2.1}$$

where $A = (\frac{\partial I}{\partial x})^2 \bigotimes w$, $B = (\frac{\partial I}{\partial y})^2 \bigotimes w$, $C = (\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y})^2 \bigotimes w$, and $\bigotimes$ is the convolution operator, w is the Gaussian window [3]. The $H(x, y)$ is defined as:

$$H(x, y) = det(M) - k(trace(M))^2 \tag{3.2.2}$$

$$det(M) = \lambda_1 \lambda_2 = AB - C^2 \tag{3.2.3}$$

$$trace(M) = \lambda_1 + \lambda_2 = A + B \tag{3.2.4}$$

$$M = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} \tag{3.2.5}$$

where

- $\sum$ is over a small region around a corner
- I_x is the gradient with respect to x

---

[3] For clarity in exposition the Gaussian weighting factor $e^{-(x^2+y^2)/(2\sigma^2)}$

- I_y is the gradient with respect to y
- λ_1 and λ_2 are two eigenvalues of matrix M

This matrix is a Harris matrix. If a circular window (or circularly weighted window, such as a Gaussian) is used, then the response will be isotropic. If $\lambda_1 \approx 0$ and $\lambda_2 \approx 0$ then this pixel has no features of interest. If $\lambda_1 \approx 0$ and λ_2 has some large positive value, then an edge is found. If λ_1 and λ_2 have large positive values, then a corner is found.

3.2.2.2 Noble corner detection

This corner detector is strongly based on the Harris corner Detector outlined above. Noble [59] uses the same process as Harris in order to obtain the autocorrelation matrix M. Considering that the variable k in Harris and Stephens measure was too weak and difficult to set up, he proposed a new cornerness measure $H(x, y)$ as follows:

$$H = \frac{det(M)}{trace(M) + \varepsilon} \tag{3.2.6}$$

which removes the parameter k, but introduces the small constant ε to avoid a singular denominator in case of a rank zero divisor.

3.2.2.3 SUSAN corner detector

SUSAN as an acronym stands for Smallest Univalue Segment Assimilating Nucleus. The SUSAN algorithm includes image noise filtering, edge finding and corner detection. This principle is implemented using digital approximation of a circular mask. SUSAN places a circular mask over the pixel to be tested (the nucleus). As shown in Figure 3.9, a circular mask is shown at five image positions, if the brightness of each pixel within a mask is compared with the brightness of that mask's nucleus then an area of the mask can be defined which has the same (or similar) brightness as the nucleus. This area of the mask shall be known as the "USAN", an acronym standing for "Univalue Segment Assimilating Nucleus". In Figure 3.10 each mask from Figure 3.9 is depicted with its USAN shown in white. This concept of each image point having associated with it a local area of similar brightness is the basis for the SUSAN principle. The local area or USAN contains much information about the structure of the image. It is effective in region finding on a small scale. From the size, centroid and second moments of the USAN two dimensional features and edges can be detected. The area of an USAN conveys the most important information about the structure of the image in the region around any point in question. As can be seen from Figures 3.9 and 3.10, the USAN area is at a maximum when the nucleus lies in a flat region of the image surface, it falls to half of this maximum very near a straight edge, and falls even further when inside a corner. It is this property of the USAN's area which is used as the main determinant of the presence of edges and two dimensional features.

Consideration of the above arguments and observation of the examples and results shown in Figures, the SUSAN principle can be formulated as:

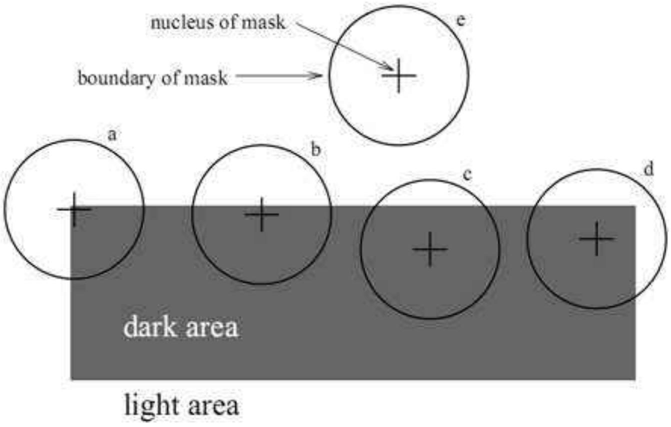


Figure 3.9: Four circular masks at different places on a simple image

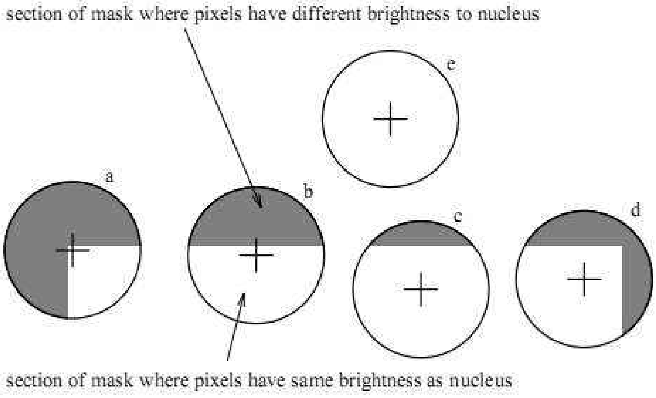


Figure 3.10: Four circular masks with similar colouring; USANs are shown as the white parts of the masks

An image processed to give as output inverted USAN area has edges and two dimensional feature strongly enhanced, with the two dimensional features more strongly enhanced than edges.

The implementation of this principle into a corner detector is similar to an edge detection. Suppose that the mask is M, and pixel in this mask is represented by $\vec{m} \in M$. The nucleus is at $\vec{m}_0$. Every pixel is compared to the nucleus using the comparison function:

$$c(\vec{m}) = e^{\frac{(I(\vec{m})-I(\vec{m}_0))^6}{t}} \tag{3.2.7}$$

where t determines the radius of the mask, and the power of the exponent has been determined empirically. The area of the SUSAN is given by

$$n(M) = \sum_{\vec{m} \in M} c(\vec{m}) \tag{3.2.8}$$

If c is the rectangular function, then n is the number of pixels in the mask which are within t of the nucleus. The response of the SUSAN operator is given by:

$$R(M) = \begin{cases} g - n(M) & \text{if } n(M) < g \\ 0 & \text{otherwise,} \end{cases} \tag{3.2.9}$$

where g is named the "geometric threshold". In other words the SUSAN operator only has a positive score if the area is small enough. The locally smallest USAN can be found using non-maximal suppression, and this is the complete SUSAN operator. The value t determines how similar points have to be in the nucleus before they are considered part of the univalue segment. The value of g determines the minimum size of the univalue segment. If g is large enough, then this becomes an edge detector. For corner detection, two further steps are used. Firstly, the centroid of the USAN is found. A proper corner will have the centroid far from the nucleus. Secondly, all points on the line from the nucleus through the centroid out to the edge of the mask are in the SUSAN.

3.2.2.4 Corner detector based on global and local curvature properties

Corner detector based on global and local curvature properties [11] is an extended version of the CSS method presented in [55] and [56]. The CCS method is a curvature-based detector that detects both fine and coarse feature accurately at low computational cost. To begin with, the curvature, K, from [66] is defined as follows:

$$K(u,\sigma) = \frac{\dot{X}(u,\sigma)\ddot{Y}(u,\sigma) - \ddot{X}(u,\sigma)\dot{Y}(u,\sigma)}{[\dot{X}(u,\sigma)^2 + \dot{Y}(u,\sigma)^2]^{1.5}} \tag{3.2.10}$$

where $\dot{X}(u,\sigma) = x(u) \otimes \dot{g}(u,\sigma)$, $\ddot{X}(u,\sigma) = x(u) \otimes \ddot{g}(u,\sigma)$, $\dot{Y}(u,\sigma) = y(u) \otimes \dot{g}(u,\sigma)$, $\ddot{Y}(u,\sigma) = y(u) \otimes \ddot{g}(u,\sigma)$, and $\otimes$ is the convolution operator, while $g(u,\sigma)$ denotes a Gaussian of width σ and $\dot{g}(u,\sigma)$ and $\ddot{g}(u,\sigma)$ are the first and second derivatives of $g(u,\sigma)$ respectively. The traditional single-scale algorithms detect corners by considering their local properties, and usually either miss fine features or detect noisy pixels as false corners. Compared to the traditional single-scale algorithms, this version of corner detector is to utilize global and local curvature properties, and balance their influence when extracting corners. Its main steps can be summarized as follows:

1. Detect the edges using Canny edge detector to obtain a binary edge map.
2. Extract contours form the edge map. When the edge reaches an end, fill the gap and continue the extraction till the end point is nearly connected to another end point, or marks this point as T-junction corner if the end points are nearly connected to an edge contour, but not to another end.
3. Compute the curvature at a fixed low scale, and regard the local maxima of absolute curvature as corner candidates.

4. Compute a threshold adaptively according to the mean curvature within a region of support. Round corners are removed by comparing the curvature of corner candidates with adaptive threshold.
5. Remove the false corners based on a dynamically recalculated region of support, evaluate the angles of the remaining corners.
6. Finally, consider the end points of open contours, and mark them as corners unless they are very close to another corner.

3.3 Laser Peak Detection

Laser peak detection is used for locating the peak position in a laser image. The object's surface 3D information is recovered by the laser projection angle, camera calibration parameters and peak location in the laser images. Thus, laser peak is one of the most important steps during the entire scanning process. Fig.3.11 shows a typical profile of laser line intensities. Even though the maximum pixel value is around 125, its exact location is uncertain yet and needs to be estimated. Hence, it is necessary to know which laser peak detection method is more accurate and precise in our 3D laser scanning system.

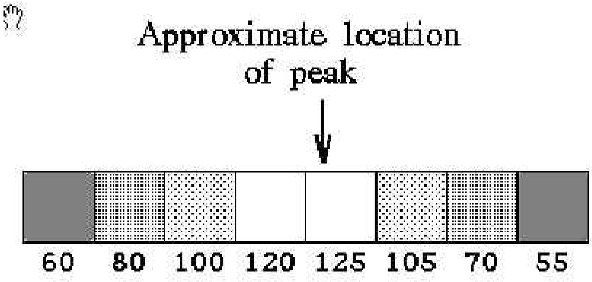


Figure 3.11: Typical intensity values of a laser profile

3.3.1 Outlines of peak detection methods

Fisher and Naidu [22] compared some of the popular detection methods including: (1) Gaussian approximation (**GA**),(2) Center of Mass (**CM**), (3) Linear Interpolation (**LIP**) (4) Parabolic estimator (**PE**) (5) Blais and Rioux detector (**BR**). These methods consider the peak and nearby ($2 \backsim 4$) pixels intensity. However, the laser stripe maybe saturated and cross over ten pixels in the worst case (Image size: 720×576) and they may fail in such condition. In 2004, Josep Forest proposed the FIR method based on FIR filter. It is able to detect the peak in high saturated and noisy laser image. It is interesting to compare these detectors' performance and effect in our 3D model accuracy. Hence, brief outline of these laser peak detectors will be given in this section and a comparative study based on 3D model will be carried out in experiment Section 6.3.

3.3.1.1 Gaussian approximation

This algorithm uses the three highest, contiguous intensity values around the observed peak of the stripe and assumes that the observed peak shape fits a Gaussian profile. This assumption is true as the laser light incident on the scene is known to be nearly Gaussian distributed as Figure 3.12 shows. The sub pixel offset of the peak is given by:

$$\alpha = \frac{\ln(f(x-1)) - \ln(f(x+1))}{2(\ln(f(x-1)) - 2\ln(f(x)) + \ln(f(x+1)))} \tag{3.3.1}$$

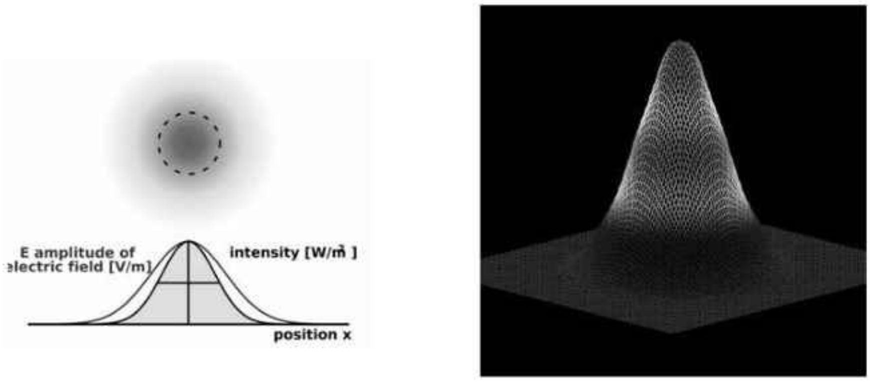


Figure 3.12: Typical Laser beam

As $f(x)$ is intensity of a pixel x and is usually integers in the range 0-255, the log calculation can be performed by table lookup.

3.3.1.2 Center of Mass

The center-of-mass algorithm also assumes that the spread of intensity values across the stripe conform to a Gaussian distribution. Thus, the location of the peak can be computed by a simple weighted-average method. The sub pixel location of the peak is given by:

$$\alpha = \frac{f(x+1) - f(x-1)}{f(x-1) + f(x) + f(x+1)} \tag{3.3.2}$$

3.3.1.3 Linear Interpolation

This method assumes that a simple, linear relationship defines the spread of intensity values before and after the peak. Thus, if the three highest intensities are identified as before, then:
If $f(x+1) > f(x-1)$

$$\alpha = \frac{f(x+1) - f(x-1)}{2 * (f(x) - f(x-1))} \tag{3.3.3}$$

else

$$\alpha = \frac{f(x+1) - f(x-1)}{2 * (f(x) - f(x+1))} \tag{3.3.4}$$

3.3.1.4 Parabolic Estimator

A continuous version of the peak finder is derivable from the Taylor series expansion of the signal near the peak. If the peak is at $f(x+\delta)$ while δ is introduced, α is estimated and we observe the signal at $f(x)$, then we have:

$$\alpha = \frac{f(x-1) - f(x+1)}{2(f(x+1) - 2f(x) + f(x-1))} \tag{3.3.5}$$

3.3.1.5 Blais and Rioux Detectors

Blais and Rioux[4] introduced fourth and eighth order linear filters:

$$\begin{aligned} g_4(x) &= f(x-2) + f(x-1) - f(x+1) - f(x+2) \\ g_8(x) &= f(x-4) + f(x-3) + f(x-2) + f(x-1) - \\ &\quad f(x+1) - f(x+2) - f(x+3) - f(x+4) \end{aligned} \tag{3.3.6}$$

and a second order filter

$$g_2(x) = f(x-1) - f(x+1) \tag{3.3.7}$$

These operators act like a form of numerical derivative operator. The peak position is estimated as above by if $f(x+1) > f(x-1)$

$$\hat{\delta} = \frac{g(x)}{g(x) - g(x+1)} \tag{3.3.8}$$

Or if $f(x+1) < f(x-1)$ we have

$$\hat{\delta} = \frac{g(x-1)}{g(x-1) - g(x)} \tag{3.3.9}$$

3.3.1.6 FIR Filter

Sometimes, when the captured laser image is not so perfect, the maximum intensity points will spread over several pixels. For instance, the intensity values would be $[115, 169, 255, 255, 255, 255, 255, 255, 255, 135]$, which means the laser stripe is saturated in the middle. In the worst case, saturate area will be more than ten pixels wide. Fig.3.133.14 is an example of row data in an high saturated laser stripe. The saturated area crosses seven pixels in this row data. In paper [24], Forest reported a digital filtering techniques (FIR) in order to cope with the scanning of different surface with different optical properties and noise levels including laser saturation. From the point of view of signal processing, it seems reasonable to consider it as the manifestation of a noisy signal, which complies with the principle of superposition. Regarding each row of the stripe image as a signal, a digital low pass filter can be designed with the right cut-off frequency, attenuation and transition band width parameters.

The specific steps so called FIR filter are given below:

1. Estimate the filter coefficients using Matlab *fdatool* function.

2. Compute the convolution of the row signals with the coefficients of the filter estimated in Step.1.

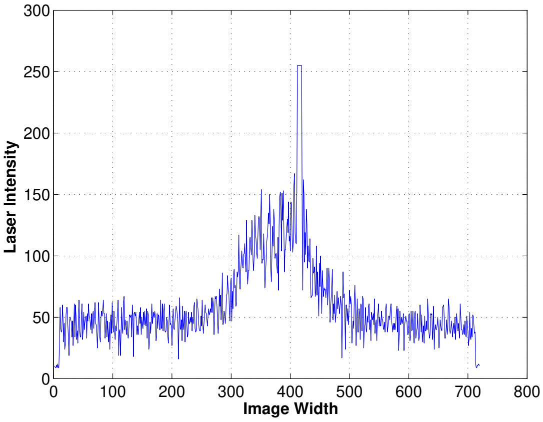


Figure 3.13: Saturated laser row

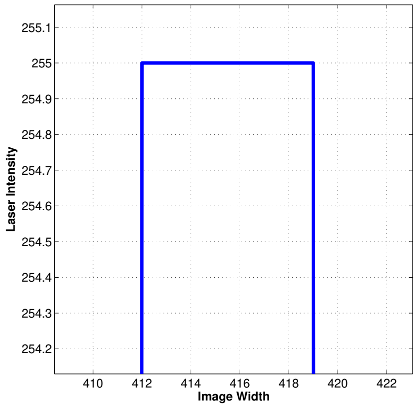


Figure 3.14: Enlarged laser row saturated area

3. Compute first derivative of the filtered signal from Step.2
4. The maximum grey level value is selected
5. The signal is tracked from its maximum, left to right in the image, until the first negative value is found.
6. A straight line is computed between the points corresponding to the first negative and the last positive signal value, and the zero crossing is computed. Fig.3.15 summarise this process and equation 3.3.10 shows how the estimation of the zero crossing is computed.

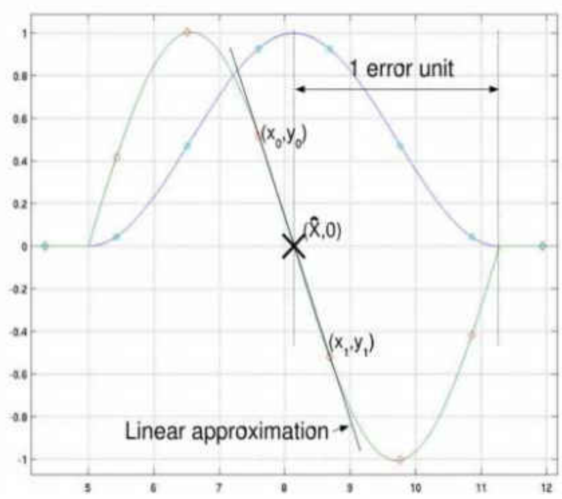


Figure 3.15: First order derivative's zero cross

$$\hat{X} = x_0 - \frac{y_0(x_1 - x_0)}{y_1 - y_0} \tag{3.3.10}$$

Chapter 4

Camera Calibration

Range images are becoming more and more widely used to model 3D environment, such as planning path from the starting to the destination location, and avoid any obstacle on the way to its destination for robot navigation. 3D data can be captured through using the latest laser scanning systems or stereo vision systems. Whichever method is adopted to capture the 3D data, it is usually necessary to model image formation, calibrate camera parameters of interest, and correct the obtained distorted data due to the unavoidable distortion of the physical camera lens. Otherwise, the analysis of the resulting distorted data may provide biased information. Thus, camera calibration and image correction is a necessary and important step for 3D metric measurement.

This chapter is structured as follows: Section 4.1 describes the background and necessity of the camera calibration; Section 4.2 describes the basic categorization of camera calibration, three main lens distortion models and the explanation of some commonly used calibration parameters; Section 4.3 describes the perspective projection model and general calibration procedure; Section 4.4 proposes a fraction distortion model; Section 4.5 proposes an iterative camera calibration method. Both proposed methods' experimental results are presented in Section 6.5.

4.1 Introduction and Background

Camera calibration in the context of three-dimensional data capture is the process of determining the internal camera geometric and optical characteristics (intrinsic parameters) and/or the 3D position and orientation of the camera frame relative to a certain world coordinate system (extrinsic parameters) [79]. The fundamental task of calibration is to find the parameters of the projection model that links the known 3D points and their projections and then rebuild/identify the object using these parameters. The relationship of the points' position in the object space and the corresponding points in the image is determined by the camera imaging geometry. This geometry is characterized by camera intrinsic/extrinsic parameters. The process of camera calibration is to find these parameters, the camera position and orientation in the world coordinate system.

Camera calibration is often used (1) as an early stage in Computer Vision and especially in the field of augmented reality; (2) in the application of Stereo vision where the camera projection matrices of two cameras are used to calculate the 3D world coordinates of a point viewed by both cameras; (3) as an early stage in image processing. When a camera is used, light from the environment is focused on an image plane and captured. This process reduces the dimensions of the data taken in by the camera from three to two (light from a 3D scene is stored in a 2D image). Each pixel in the image plane therefore corresponds to a shaft of light from the original scene. Camera calibration determines which incoming light is associated with each pixel in the resulting image. In an ideal pinhole camera, a simple projection matrix is enough to do this. With more complicated camera systems, errors resulting from misaligned lenses and deformations in their structures can result in more complex distortions in the final image.

4.2 Calibration Methods

The existing camera calibration techniques can be classified broadly into the following four main categories: calibration rig based, self calibration, auto-calibration, and parameter free correction:

- The calibration rig based approaches require the knowledge of correspondences between the 3D calibration rig and its projective image. The rig contains either non-coplanar points [79, 29] or coplanar points [83]. While the approach in [83] is based on the rigidity constraint on the rigid rotation matrix, that in [79] is based on the radial alignment constraint. The calibration usually involves two-steps. In the first step, a crude estimate of the parameters of interest is obtained often with closed form solutions. In the second step, all parameters of interest are iteratively optimized globally through minimizing some error functions. One of the most widely used objective functions is the sum of the squared back-projection errors.

- The self calibration techniques require just a single projective image. The distortion parameters can be estimated using either the projective geometry [17, 18, 20] or in the frequency domain [21]. The former applies the property that the projections of lines are still lines in the corrected image. Thus, the curvatures of line segments in the distorted image are due to the camera distortion. In this case, various objective functions can be constructed: minimising distances from the points to best fit line segments [17], slope variation of the line segments [20], etc. In contrast, the latter observes that the Fourier transform of the signal before and after distortion is correlated which can be defined as the bicoherence. The first order radial distortion parameter can be estimated as one of the candidates sampled from an interval that minimises the bicoherence. The experimental results show that the frequency domain method is by no means comparable to that based on calibration rigs for camera calibration and correction.

- Auto-calibration approaches require at least two projective images of the same scene and are often based on epipolar geometry [65]. To calibrate the intrinsic camera parameters, the constraints on the plane at infinity and the quadrics in that plane are often constructed. Usually, auto calibration approaches do not consider the camera distortion. Even so, it was observed in [5] that it is often difficult to estimate the focal length, the estimation of the principal point is often unstable, and there is an ambiguity in calibrating the focal length, the principal point, and the camera position.

- Parameter free approaches [28] take only the radial distortion into account and estimate the distortion factor for each point. To estimate the distortion factors, a planar calibration grid is required and a constraint on the center of distortion and the homography $\mathbf{H}$ transforming calibration grid points in the Euclidean space into the undistorted image points in pixel coordinates is derived. Interestingly enough, the constraint is very similar to that on the fundamental matrix, which results in the estimation of the first two rows of $\mathbf{H}$. To estimate the third row of $\mathbf{H}$, an assumption of monotonicity of distortion with regard to the radial distance is made. This assumption implies that the neighbouring undistorted points should not differ significantly whose squared difference can be minimised, resulting in the relative parameters being estimated in the least squares sense.

4.2.1 The Lens distortion

The basic camera calibration usually includes two main tasks: 1) find the parameters of the projection model including extrinsic and intrinsic parameters; 2) model and estimate the camera lens distortion parameters so that the captured images can be un-distorted. This section will describe several lens distortion models in Task 2 and explain how they affect the quality of the captured images.

Aberrations are departures of the performance of an optical system from the predictions of paraxial optics. Aberration leads to blurring of the image produced by an image-forming optical system. It occurs when light from one point of an object after transmission through the system does not converge into (or does not diverge from) a single point. [1] There are three main types of aberrations which can affect the image quality. The first one is caused by the imperfect lens shape and manifests itself by radial positional error only, whereas the second and the third types of distortion are generally caused by improper lens and camera assembly and generate both radial and tangential errors in point positions, see Figure 4.1, [80].

1. **Radial Distortion**: Radial distortion causes an inward or outward displacement of giving image point from its ideal location. These types of distortion are mainly caused by flawed radial curvature of the lens elements. A negative radial displacement of the image point is referred to as barrel distortion. It causes outer points to crowd increasingly together and the scale to decrease. A positive radial displacement is referred to as pincushion distortion. It causes outer points to spread

---

[1] Access on 3rd August, 2011, http://en.wikipedia.org/wiki/Aberration_in_optical_systems

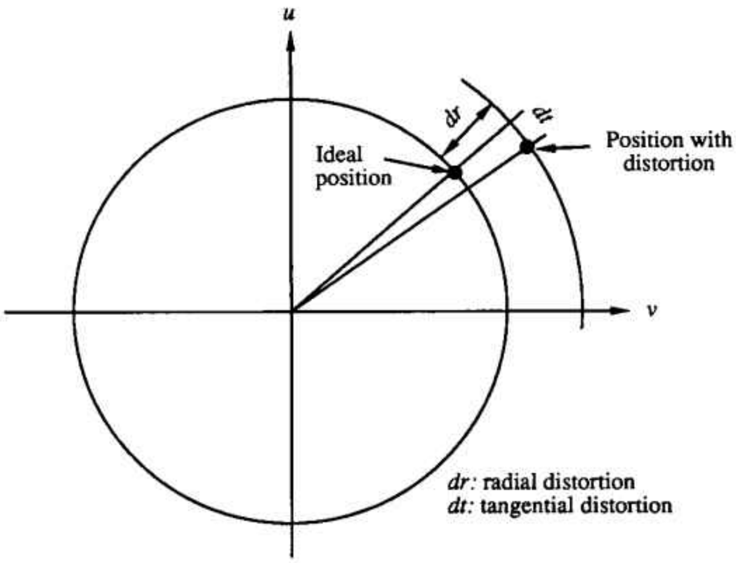


Figure 4.1: Radial and tangential distortions

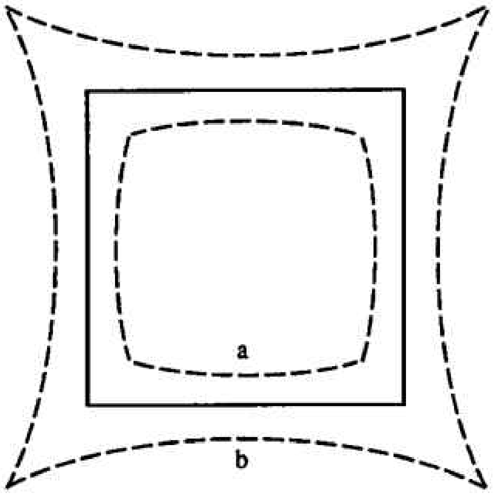


Figure 4.2: Effect of radial distortion. Solid lines: no distortion; dashed lines: with radial distortion (a: negative, b: positive)

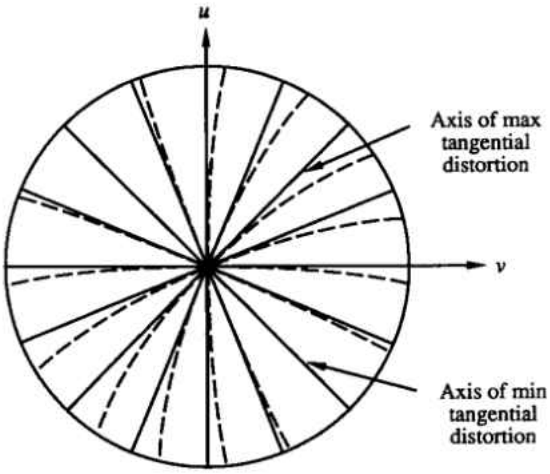


Figure 4.3: Effect of tangential distortion. Solid lines: no distortion; dashed lines: with tangential distortion

and the scale to increase. This type of distortion is strictly symmetric about the optical axis. Fig.4.2 illustrates the effect of radial distortion [80].

2. **Decentring Distortion**: Actual optical systems are subject to various degrees of decentring, that is , the optical centres of the lens elements are not strictly collinear. This defect introduces what is called decentring distortion. This distortion has both radial and tangential components [80].

3. **Prism Distortion** Thin prism distortion arises from imperfection in lens design and manufacturing as well as camera assembly (for example, slight tilt of some lens elements or the image sensing array). These types of distortion can be adequately modelled by the adjunction of thin prism to the optical system, causing additional amounts of radial and tangential distortions [80]. Such distortions can be expressed as

$$\begin{aligned} \delta_{\rho p} &= (i_1\rho^2 + i_2\rho^4 + \cdots)sin(\varphi - \varphi_1) \\ \delta_{tp} &= (i_1\rho^2 + i_2\rho^4 + \cdots)cos(\varphi - \varphi_1) \end{aligned} \tag{4.2.1}$$

where ρ_1 is the angle between the positive u axis and the axis of maximum tangential distortion shown in Fig.4.3

4.2.2 Calibration parameters

Below are the common definitions in [79, 83] of camera calibration parameters including intrinsic and extrinsic parameters of pinhole perspective projection model and used in the entire thesis:

- f : Effective focal length of the pin-hole camera (mm or pixel).

- d_x, d_y : Center to center distance between adjacent sensor elements in X and Y (mm).
- C_x, C_y : Principal point coordinates (that is usually at the image centre).
- N_{cx} : Number of sensor elements in x direction of the camera (sels).
- N_{fx} : Number of pixels in x direction of the frame grabber (pixel).
- s_x : scale factor for uncertainty in the re-sampling of the horizontal scan line in the frame grabber.
- $k_1, k_2 \cdots, k_n$: N-th order radial lens distortion coefficient.
- p_1, p_2 : Tangential lens distortion coefficient.
- R_x, R_y, R_z : Rotation angles around the x, y, and z axes for the transformation between the world and camera coordinate systems,
- t_x, t_y, t_z : Translation components along the x, y, and z axes for the transformation between the world and camera coordinate system.

4.3 Perspective Geometry Model

To understand how the 3D real world can be digitized into a computer, it is necessary to understand the image acquisition process. The role of the camera in machine vision is analogous to that of the eyes in biological systems. For the sake of simplification, the ideal pinhole model camera is often used to approximate this function. It has an infinitesimally small hole, which allows light enters before forming an inverted image on the camera surface facing the hole. Normally, a pinhole camera places the image plane between the focal point of the camera and the object, so that the image is not inverted. This mapping of three dimensions onto two, is called a perspective projection (see Figure 4.4), and perspective geometry is fundamental to the understanding of image formation and analysis.

Based on this perspective projection model (assuming the lens is distortion free), a scene view is formed by projecting 3D points into the image plane by the perspective transformation. This transformation can be represented as Eq.4.3.1 or 4.3.2

$$sm' = A[R|t]M \tag{4.3.1}$$

Or

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{4.3.2}$$

where (X, Y, Z) are the coordinates of a 3D point in the world coordinate system, (u, v) are the coordinates of the projection point in the 2D image plane in pixels. A is called a

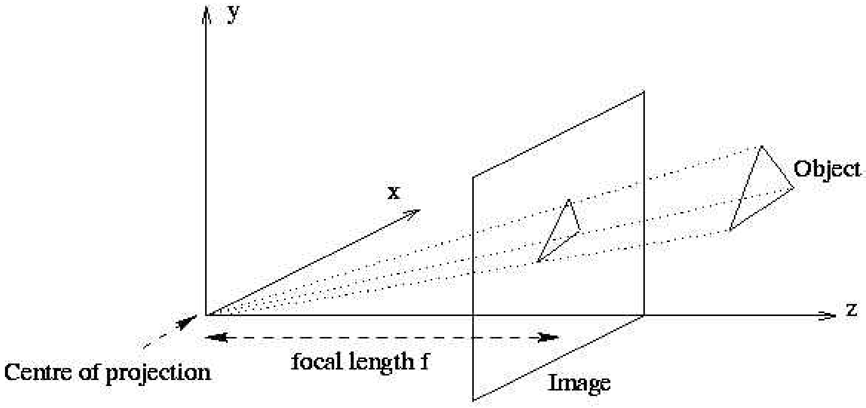


Figure 4.4: Perspective projection in the traditional pinhole camera model

camera matrix, including intrinsic parameters. (c_x, c_y) is a principal point (it is usually the image centre as initial guess), and f_x, f_y are the focal lengths expressed in pixel-related units along x and y axes respectively. Thus, if an image from a camera is scaled by some factor, all of these parameters should be scaled (multiplied/divided, respectively) by the same factor. The matrix of intrinsic parameters does not depend on the scene viewed and, once estimated, can be re-used (as long as the focal length is fixed (in case of zoom lens)). The joint rotation-translation matrix $[R|t]$ is called a matrix of extrinsic parameters. It is used to describe the camera motion around a static scene, or vice versa, rigid motion of an object in front of still camera. That is, $[R|t]$ transforms a point (X, Y, Z) in 3D world coordinate system to a point (x, y, z) in a local camera centred coordinate system, fixed with respect to the camera. The transformation above is equivalent to the following (when $z \neq 0$):

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + t \tag{4.3.3}$$

Hence, a point in 3D world can be transformed to 2D image plane as: [83, 17]

$$\begin{aligned} x' &= x/z \\ y' &= y/z \\ u &= f_x * x' + c_x \\ v &= fy * y' + c_y. \end{aligned} \tag{4.3.4}$$

Real camera lenses usually have distortion and distort the captured images. Mostly, the distortion contains radial distortion and slight tangential distortion. The distortion model that is used for correcting the radial and tangential distortion components is usually known as radial polynomial model [30] and Decentring model [81]. Radial polynomial model is represented as:

$$r_d = r_u(1 + k_1 r_2^2 + k_2 r_u^4 + \cdots + k_p r_u^{2p}). \tag{4.3.5}$$

Decentring polynomial model is represented as:

$$\begin{aligned} x_d &= p_1(r_u^2 + 2x_u^2) + 2p_2 x_u y_u \\ y_d &= p_2(r_u^2 + 2y_u^2) + 2p_1 x_u y_u. \end{aligned} \tag{4.3.6}$$

where (x_d, y_d) is the distorted point and (x_u, y_u) is the distortion free point. Thus, taking Eq. 4.3.5 and Eq. 4.3.6 into account, the coordinate transformation model in Eq. 4.3.4 can be extended with distortion factors:

$$\begin{aligned} x'' &= x'(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_1 x' y' + p_2(r^2 + 2x'^2) \\ y'' &= y'(1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + p_1(r^2 + 2y'^2) + 2p_2 x' y' \\ &\text{where} \quad r^2 = x'^2 + y'^2, \\ u &= f_x * x'' + c_x \\ v &= f_y * y'' + c_y \end{aligned} \tag{4.3.7}$$

k_1, k_2, k_3 are the radial distortion coefficients, and p_1, p_2 are tangential distortion coefficients. Normally, 6th order coefficient is enough, higher-order coefficient is negligible and thus can be omitted.

Altogether, there are 15 unknown parameters if considering distortion model in Eq. 4.3.7 including: 4 intrinsic parameters $A = (f_x, f_y, c_x, c_y)$, 6 extrinsic parameters $R = (R_1, R_2, R_3)$ and $t = (t_x, t_y, t_z)$ and 5 distortion parameters $D = (k_1, k_2, k_3, p_1, p_2)$, while the fraction distortion model described below also contains 5 distortion parameters $D = (k_1, k_2, k_3, k_4, k_5)$ which has 16 parameters instead with the rotation matrix represented by a quaternion in 4 parameters. No matter which model is used to optimize these parameters, the following commonly used objective function [83] is built based on Eq.4.3.1 and Eq.4.3.2:

$$J = \min_{A,D,R_i,t_i} \sum_{i=1}^{n} \sum_{j=1}^{m} \| P_{ij} - \breve{m}(A, D, R_i, T_i, M_j) \|^2 \tag{4.3.8}$$

where $\breve{m}(A, D, R_i, t_i, M_j)$ is the re-projection points M_j in image i. This function minimizes the differences between 3D re-projected points M_j in the image plane i and given image P_{ij}. This non-linear optimization is performed using Levenberg-Marquardt algorithm [50]. The initial guess of $A, (R_i, t_i \mid i = 1...n)$, $(R \mid t)$ can be obtained with closed-form solution and maximum likelihood estimation [83], relaxing the rigidity constraint on the rotation matrix R or in the image formation model in 4.3.7.

4.4 Fraction Distortion Model

The acquisition and analysis of 3D data attract more and more attention from the robot vision community for the representation of the objects of interest in the process of object modelling, classification and recognition [2]. Accurate camera calibration is a prerequisite step for 3D metric measurement using either the latest laser scanning techniques or stereo vision systems. Without correction, the captured data are distorted, which may give an illusion to the shape of the objects of interest. Subsequent analysis of such data is not accurate or even meaningless.

In this section, we propose a novel camera distortion model which attempts to model the overall distortion. In this case, no knowledge is required about what the camera distortion is: radial, decentring, or thin prism. This attempt is practical, since in reality, we probably have little idea about what distortion the captured image is subject to. The actual distortion of the camera is not known and need to be investigated using the various metrics: collinearity constraint, parallel, perpendicularity. On the other hand, the novel model attempts to combat the imaging noise. This is very important to the subsequent data analysis, since all imaging devices introduce some amount of noise caused by point sampling, quantization of measurements, reflective properties, etc.

In order to estimate the parameters in the novel camera distortion model, we employ the Levernburg-Marquardt (LM) algorithm to globally optimize all the parameters of interest through minimizing the sum of squared differences between the transformed distorted projected 3D world points and their given distorted image points. Four parameters are used to model the image formation: focal length (f_x, f_y), aspect ratio s, and the principal point (u_0, v_0). Seven parameters are used to model the camera orientation and position: a quaternion q is used to describe the camera orientation and a 3D vector t is used to describe the camera position in the world coordinate system. Five parameters are used to describe the camera distortions.

For the sake of fair testing, we did not use the results from the Zhang [83] algorithm directly for a comparative study. Instead, we employ again the LM algorithm to globally optimize its estimation results through minimising the sum of squared differences between the transformed distorted projected 3D world points and their given distorted image points with the camera distortion explicitly modelled as both radial and decentring ones in 4 parameters altogether. Such a comparative study is valuable, since it can reveal whether the explicit knowledge of the camera distortion is beneficial to successful image point correction. The performance of camera calibration algorithms is measured from two aspects: (1) Absolute and relative correction errors. This is in contrast with the commonly used average correction error which does not take into account the fact that the distortion in the middle area of images is little, while the distortion in the peripheral area is more pronounced. As such, the average correction error may not be informative to the performance of algorithms; and (2) the collinearity constraint on points. While the collinear points lie on curves in the distorted image, they should lie on the collinear line segments after correction. The collinear fitting errors are measured as: maximum fitting error (MFE), average fitting error (AFE), and the standard deviation of fitting errors (SDFE) of points on different line segments. The experiments based on both synthetic data and real images show that the proposed algorithm produces encouraging camera calibration and correction results.

4.4.1 A new distortion model

The following notations are used throughout this section: lower case letters denote scalars, $|\cdot|$ denotes the absolute value of a scalar, $||\cdot||$ denotes the Euclidean norm of a vector, I is an identify matrix, $a \times b$ denotes the cross product of vectors a and b, $det(A)$ denotes the determinant of the square matrix A, the subscripts w and c denote the 3D points described in the world and camera centred coordinate systems, the subscripts f, u, and d denote the points in the frame buffer, undistorted, and distorted image points, respectively, variables with ˆ signs denote the corrected or estimated ones, and superscript T denotes the transpose of a vector or a matrix.

Inspired by the various camera distortion models summarised in Table 4.1 proposed in the literature and in an attempt to model the overall camera distortion without priorly knowing which distortion the lens is subject to and combat the imaging noise for accurate camera calibration and correction, we propose the following camera distortion model:

$$\begin{aligned} x_{di} &= x_{ui}\tfrac{1+k_2x_{ui}+k_3y_{ui}}{1-k_1r_{ui}^2}, \\ y_{di} &= y_{ui}\tfrac{1+k_4x_{ui}+k_5y_{ui}}{1-k_1r_{ui}^2} \end{aligned} \tag{4.4.1}$$

where k_1, k_2, k_3, k_4, and k_5 are unknown distortion parameters to be calibrated, $P_{di} = (x_{di}, y_{di})^T$ is distorted image point, $P_{ui} = (y_{ui}, y_{ui})^T$ is undistorted image point:

$$\begin{aligned} x_{ui} &= f\tfrac{R_1p_{wi}+t_x}{R_3p_{wi}+t_z}, \\ y_{ui} &= f\tfrac{R_2p_{wi}+t_y}{R_3p_{wi}+t_z} \end{aligned} \tag{4.4.2}$$

R_1, R_2 and R_3 are the three rows of the camera orientation matrix R in the world coordinate system, t_x, t_y and t_z are the three components of the camera position t, and $r_{ui}^2 = x_{ui}^2 + y_{ui}^2$ is the squared radius distance from the principal point $(c_x, c_y)^T$ to the undistorted image point P_{ui}. This model which is called a fraction model (FMC), is applied to the projected 3D world points, and clearly has the property that when $r_u = 0$, the point has no distortion at all. With the increase of r_u, the distortion in the points also increases. This property conforms to the normal observation that the points in the middle of the image are subject to less distortion, while the distortion of the points in the peripheral area is more pronounced.

4.4.2 Calibration algorithm based on fraction distortion model

In this section, we estimate the unknown parameters in the proposed camera distortion model (Equation 4.4.2) described in the last section and then summarise the main steps in the proposed algorithm for camera calibration and correction. To this end, we minimise the sum of the squared differences between the transformed distorted projected 3D world

Table 4.1: Commonly used camera distortion models

Type	Model	Applicability
Radial polynomial model	$r_d = r_u(1 + k_1 r_u^2 + k_2 r_u^4 + \cdots + k_p r_u^{2p})$	Projected/Image points
Radial model [52]	$r_d = r_u(1 + k_1 r_u + k_2 r_u^2)$	Projected points
Radial model [77]	$r_d = r_u(1 + k_1 r_u^2)$	Projected points
Radial division model [23]	$\hat{P}_f = \frac{1}{1+k_1\|\|P_f\|\|^2} P_f$	Image points
Radial model [74]	$\hat{p}_f - C = (P_f - C)\sum_{i=0}^{d} k_i \|\|P_f - C\|\|^i$	Image points
Radial rational model [12]	$\hat{P}_f = \left(\frac{A_1^T x(i,j)}{A_3^T x(i,j)}, \frac{A_2^T x(i,j)}{A_3^T x(i,j)}\right)$	Image points
Decentering model [80]	$x_d = p_1(r_u^2 + 2x_u^2) + 2p_2 x_u y_u$ $y_d = 2p_1 x_u y_u + p_2(r_u^2 + 2y_u^2)$	Projected points
Thin Prism model [80]	$x_d = s_1 r_u^2$ $y_d = s_2 r_u^2$	Projected points

points $p_{wi} = (x_{wi}, y_{wi}, z_{wi})^T$ and their given corresponding distorted projective image points $P_{di} = (x_{di}, y_{di})^T (i = 1, 2, \cdots, n)$. The details of optimization are given as follows.

4.4.3 Pin-hole camera model

The relationship between a 3D world point $p_{wi} = (x_{wi}, y_{wi}, z_{wi})^T$ and its image point $P_{fi} = (x_{fi}, y_{fi})^T$ in the image plane without distortion can be represented as:

$$Z_{ci} \begin{pmatrix} x_{di} \\ y_{di} \\ 1 \end{pmatrix} = A(Rt) \begin{pmatrix} p_{wi} \\ 1 \end{pmatrix} = Hp_{wi} + T \tag{4.4.3}$$

where $H = AR, T = At$, matrix $A = \begin{pmatrix} f_x & 0 & C_x \\ 0 & f_y & C_y \\ 0 & 0 & 1 \end{pmatrix}$ encodes the camera intrinsic parameters, s is the aspect ratio of the pixel, f is the focus length of the camera,f_x and f_y are the camera focal length in X and Y axis directions, (C_x, C_y) is the principal point, and Z_{ci} is the depth of 3D world point p_{wi} in the camera centred coordinate system. These formulas are coincident with the perspective model in Section.4.3 above.

4.4.4 Optimization of all parameters

Taking the proposed camera distortion model in Equation 4.4.1 and the pin-hole camera model Equation 4.4.3 into account, then altogether 16 parameters need to optimized: 4 intrinsic parameters **IP** $= (f_x, f_y, C_x, C_y)$, 7 extrinsic parameters **EP**=(q,t where a quaternion q is used to represent the camera orientation matrix R and three parameters are used to represent camera position t in the world coordinate system), and 5 distortion parameters **DP** $= (k_1, k_2, k_3, k_4, k_5)$. To optimize these 16 parameters, the following objective function is built:

$$J_1 = \min_{\mathbf{IP,EP,DP}} \sum_{i=1}^{n} ((\hat{x}_{fi} - x_{fi})^2 + (\hat{y}_{fi} - y_{fi})^2) \tag{4.4.4}$$

which minimize the differences between the re-projected 3D points $\hat{P}_{fi}$ in the image plane and given image points P_{fi}. The initialization is required for the Levernburg-Marquardt algorithm. The initialization was provided using the intrinsic and extrinsic parameters estimated using the classical Zhang algorithm [83] with distortion parameters all set to 0.

4.4.5 The correction of the distorted image points

From Equation 4.4.1, it can be seen that given the distorted image points $P_{di} = (x_{di}, y_{di})^T$, generally, there is no closed form solution to the undistorted points $P_{ui} = (x_{ui}, y_{ui})^T$. While some researchers turn to the approximation method [30],[29], in this section, we propose using the LM algorithm to correct the distorted points. To this end, the following objective function is built:

$$J_2 = \min_{\hat{x}_{ui},\hat{y}_{ui}} (\hat{x}_{di} - x_{di})^2 + (\hat{y}_{di} - y_{di})^2 \tag{4.4.5}$$

where $\hat{x}_{di} = \hat{x}_{ui}\frac{1+\hat{k}_2\hat{x}_{ui}+\hat{k}_3\hat{y}_{ui}}{1-\hat{k}_1\hat{r}_{ui}}$, $\hat{y}_{di} = \hat{y}_{ui}\frac{1+\hat{k}_4\hat{x}_{ui}+\hat{k}_5\hat{y}_{ui}}{1-\hat{k}_1\hat{r}_{ui}}$, $\hat{r}^2_{ui} = \hat{x}^2_{ui} + \hat{y}^2_{ui}$. The LM algorithm was initialised by the distorted image points. Since the undistorted points around the principal point are always very close to the distorted points and the undistorted points should not be far away from the distorted points, the distorted points thus often provide a good initialization for their optimized correction. As long as both the intrinsic and distortion parameters have been calibrated with reasonable accuracy, then the objective function J_2 usually well poses the image correction problem, leading the distorted points to be accurately corrected.

4.4.6 Summary of the camera calibration and correction algorithm

The main steps in this camera calibration based on the proposed fraction distortion model and correction algorithm can be summarised as follows:

1. Use Equation 4.4.1 to model the camera distortion;
2. Use the improved Zhang algorithm[83] to estimate both the intrinsic and extrinsic parameters;
3. Apply the LM algorithm to minimise the objective function Equation 4.4.4 for an optimal estimation of the camera intrinsic, extrinsic and distortion parameters. The LM algorithm is initialised by the intrinsic and extrinsic parameters that were estimated above by the Zhang algorithm and the distortion parameters that are all set to zero;
4. Apply again the LM algorithm to minimise the objective function Equation 4.4.5 to correct the distorted points with the LM algorithm initialised by the distorted points themselves.

Since the proposed algorithm is based on a fraction distortion model for camera calibration, it is called the FMC algorithm in the rest of this thesis. As it operates directly on 3D-2D correspondences and it has a linear computational complexity $O(n)$ in the number of 3D-2D correspondences used for camera calibration and correction.

4.5 Iterative Camera Calibration

In most camera calibration methods, they assumed that the precise 2D and 3D coordinates of the control points are available. However, in practice, this is not always the case due to fact that the 3D pattern manufacturing process always introduces errors to the pattern. In addition, the accuracy of control points detection is suffered from image distortion. Therefore, this section we propose iterative based camera calibration to reduce the errors from inaccurate 3D pattern and the effect from image distortion.

4.5.1 Some Backgrounds

Based on the above sections, we now consider how to improve existing methods to achieve more accurate calibration results. From Eq.4.3.1 and Eq.4.3.2, it is known that camera calibration requires two sets of input points: (X, Y, Z) and (u, v), where the former is the 3D coordinates of the 3D points in world space, the latter is their 2D projection in the image plane. Clearly, the more accurate 3D input points are, the more precise the calibration results will be. Inaccurate calibration results can be analysed from reliability of these two sets of points.

Precise 3D point's pattern can be classified into two types: coplanar or non-coplanar, as shown in Fig 4.6 and 4.5. Chessboard and circular pattern are normally used for camera calibration, since various constraints can be extracted that can be used to constrain the calibration process. Both of these pattern types have shown to be detectable with sub-pixel accuracy, but the chessboard pattern deals with the misplacement error introduced by radial distortion, which is easier to examine even by eye.

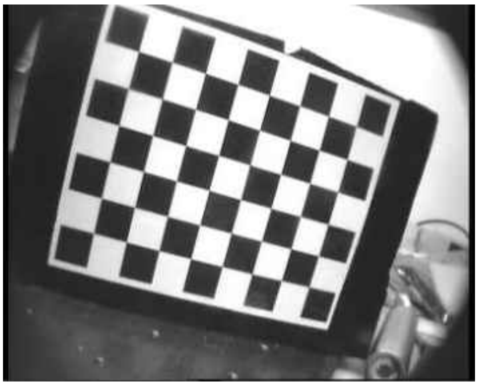


Figure 4.5: Coplanar chess pattern

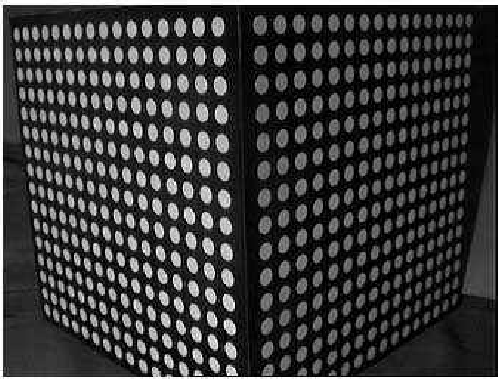


Figure 4.6: Non-coplanar circular pattern[29]

These patterns can be manufactured in precision engineering, which can narrow the error down to $\pm 0.01 mm$. However, such accurate 3D markers are very difficult and expensive to build. It requires expensive tooling and need a remarkable amount of effort for maintenance and verification. Although printing the patterns on a paper is an easy and fast solution to prepare the 3D target, the accuracy is not always perfect. In paper [1], it was mentioned that the normal ink-jet printer was more accurate, showing printing errors greatly below the measurement threshold of the ruler (i.e. sub-millimetre) both with respect to the global chessboard size and its aspect ratio compared to a laser printer. The reason is the different ways in which they work: ink-jet printer has internal linear stage control, which should be more precise compared to rotate the hollow tube (heat roller) in laser printers. However, the 3D pattern printing still contains some errors which are unavoidable. To compensate the error, Jean-Marc [47] proposed a method which reduces this type of error. In his proposed method, the 3D world points (X_w, Y_w, Z_w) are varied and can be adjusted in global optimization.

The steps of this algorithm are as follows:

1. Input original 3D world coordinates (X_{wj}, Y_{wj}, Z_{wj})
2. Perform the standard camera calibration (by [83] or [79]), and compute the re-projection error ξ
3. Input 3D world coordinates as variables into the global optimization. The original coordinates will be regarded as initial guess. Eq.4.3.8 can be extended to be:

$$\sum_{i=1}^{n}\sum_{j=1}^{m}\|P_{ij} - \breve{P}(A, D, R_i, T_i, X_{wj}, Y_{wj}, Z_{wj})\|^2 \tag{4.5.1}$$

By this method, the author claims that even the 3D points contain up to $10mm$ noise, the calibration results are very close to the true solution. More details can be found in the later experimental results section.

In Paper [1], Andrea proposed a similar method to compensate the inaccurate 3D world pattern but using bundle adjustment to deal with a large number of unknown variables. Below are the main steps:

1. Input original 3D world coordinates (X_{wj}, Y_{wj}, Z_{wj})
2. Perform the standard camera calibration by [83] with existing input datasets including detected feature points (like corners), and compute the re-projection error ξ
3. The camera parameters obtained from the previous step are assumed to be correct, the geometry of 3D input points can be rectified by the bundle adjustment for the estimation of only the camera poses and scene (the 3D input points).
4. By the new set of parameters, a smaller re-projection error can be achieved.
5. Repeat Steps 2 $\sim$ 4 until the error ξ converges or the maximum number of iterations has been reached
6. Once the error converges, a further step is needed, specifically we need to rescale the newly created target (3D world input points) to fit the original one. This is necessary since the bundle adjustment step does not guarantee scale invariance. This adjustment step is performed by using the robust closed form point alignment technique by [31].

The key technique in Andrea method is bundle adjustment. Bundle adjustment is the problem of refining a visual reconstruction to produce jointly optimal 3D structure and viewing parameter (camera pose and/or calibration) estimates. Optimal means that the parameter estimates are found by minimizing some cost function that quantifies the model fitting error, and jointly means that the solution is simultaneously optimal with respect to both structure and camera variables. The name refers to the 'bundle' of light rays leaving each 3D feature and converging on each camera centre, which are 'adjust' optimally with respect to both feature and camera positions. Equivalently, unlike independent model methods, which merge partial reconstructions without updating their internal structure-all of the structure and camera parameters are adjusted together 'in one bundle' [78].

After step 2, there are $(X_w, Y_w, Z_w)_n$, n is the number of captured images by the number of pattern points plus 15 calibration parameters that need to be optimized. For example, an 8×6 chessboard pattern in five captured images will bring $48 \times 5 \times 3 = 720$ non-linear parameters to be adjusted. To increase the processing speed, sparse bundle adjustment [51] is used in our application. It is far more efficient compared to the original Levenberg-Marquardt Algorithm.

Datta [16] developed another iterative method to perform the refinement. It considers the camera distortion and pattern position in the images that both affect the feature detection and final calibration results. Here is the algorithm outline:

1. Detect the calibration pattern in the input images.
2. Use the detected control points to estimate the camera parameters.
3. Use the above camera parameters to undistort and un-project input images to a canonical pattern.
4. Localize calibration pattern control points in the canonical pattern.
5. Project back the control points using the estimated camera parameters.
6. Use the projected control points to re-fine the camera parameters using Levenberg-Marquardt.
7. Repeat Steps 3-6 until convergence or reach maximum iteration number.

The advantage of this method is un-project the images to a canonical pattern view, which manifests as the fact to reduce the effect of lens distortion and increase the accuracy of feature detection in the canonical-front view. However, this canonical-front view sometimes is hard to compute and the grid may appear outside the image plane after un-distortion. It really limits the flexibility of using multi orientations of calibration pattern, for example, the fish-eye lens may not be able to apply to this method; the calibration pattern only limits to move/rotate at a small angle to make sure the grid appears in the image after the images being undistorted. Hence, our experiment only considers Andrea method for comparison against the proposed iterative camera calibration.

Usually, the given the 3D world points (X_{wj}, Y_{wj}, X_{wj}) and 2D image points (u_j, v_j) are fixed and used throughout the whole process of camera calibration, assuming that they are accurate. However, in practice, the image points are detected from the distorted images, and thus, are not accurate. This implies that after the camera parameters have been estimated in each iteration of the calibration, the images should be un-distorted and the feature points have to be re-detected from the un-distorted images for more accuracy.

4.5.2 Combination and further development

From the description in the above sections, it can be inferred that Datta's and Andrea's methods are compatible with each other. Thus, we propose a new method which combines the advantage within the above algorithms and considers the input 2D detected feature points altogether. The images used for feature detection always come with more or less lens distortion. Such image distortion affects the feature detection accuracy and brings error in final calibration results. To reduce such lens distortion affection on feature detection like corner or circular detection, it is possible to un-distort the image before the detection. Based on this observation and the previous sections, we propose a combination method to achieve higher accuracy camera calibration.

Here are the main steps in the proposed method:

1. Camera Calibration (for example, Zhang [83] or Tsai [79]) obtains the calibration parameters.
2. Bundle adjustment to have a new set of 3D points which have better fit and minimum re-projection error.
3. A further step is needed, specifically we need to rescale the newly created target (3D world input points) to fit the original one. This is necessary since the bundle adjustment step does not guarantee scale invariance. This adjustment step is performed by using the robust closed form point alignment technique by [31].
4. Perform the calibration again and have a new set of parameters using the new set 3D points and the calibration parameters from step 1 as initial guess.
5. Un-distort the images to extract better corner position. (more precise corner location can be achieved in distortion free images) Such corner position can be used in the next iteration from step 1.
6. Repeat Steps $1 \sim 5$ until the error converges or its maximum number of iterations has been reached.

The calibration result shown in Table.4.2 was obtained after 30 iterations. Total eight chessboard pattern images were used in this example. The image size is 720×576 and there are 48 corners in total. As it can be seen in this table, the re-projection error has been decreased very fast in just two iterations, and parameters including f_x, f_y, C_x, C_y

Round	f_x	f_y	C_x	C_y	RPE
0	672.13252	728.37007	367.49119	311.85001	0.16970
1	676.50042	733.19166	366.45614	312.06166	0.10029
2	678.34047	735.28178	365.84873	312.13756	0.09912
3	679.24575	736.36300	365.41231	312.12981	0.09834
4	679.72589	736.98538	365.06730	312.06847	0.09768
5	679.97993	737.36290	364.78062	311.97429	0.09710
6	680.10293	737.59737	364.53471	311.86128	0.09656
7	680.14710	737.74460	364.31879	311.73866	0.09607
8	680.14339	737.83767	364.12578	311.61234	0.09561
9	680.11083	737.89690	363.95090	311.48610	0.09518
10	680.06130	737.93503	363.79083	311.36234	0.09477
11	680.00228	737.96007	363.64325	311.24255	0.09440
12	679.93844	737.97707	363.50651	311.12764	0.09404
13	679.87272	737.98925	363.37939	311.01814	0.09371
14	679.80694	737.99860	363.26099	310.91432	0.09340
15	679.74221	738.00639	363.15061	310.81632	0.09311
16	679.67919	738.01340	363.04769	310.72413	0.09283
17	679.61827	738.02011	362.95178	310.63772	0.09257
18	679.55963	738.02680	362.86250	310.55696	0.09233
19	679.50337	738.03365	362.77951	310.48173	0.09211
20	679.44949	738.04073	362.70253	310.41190	0.09190
21	679.39797	738.04810	362.63128	310.34729	0.09170
22	679.34875	738.05576	362.56551	310.28774	0.09151
23	679.30176	738.06372	362.50499	310.23309	0.09134
24	679.25693	738.07196	362.44951	310.18318	0.09117
25	679.21416	738.08046	362.39886	310.13782	0.09102
26	679.17339	738.08920	362.35284	310.09687	0.09088
27	679.13454	738.09816	362.31127	310.06016	0.09074
28	679.09752	738.10733	362.27395	310.02753	0.09062
29	679.06225	738.11666	362.24072	309.99884	0.09050

Table 4.2: Iterative camera calibration example, RPE: re-projection error, unit: pixel

converged after 26 iterations. Fig.4.7 shows more clearly how quickly the re-projection error drops. From these result we can see that the detected feature points plays a key role for accurate camera calibration.

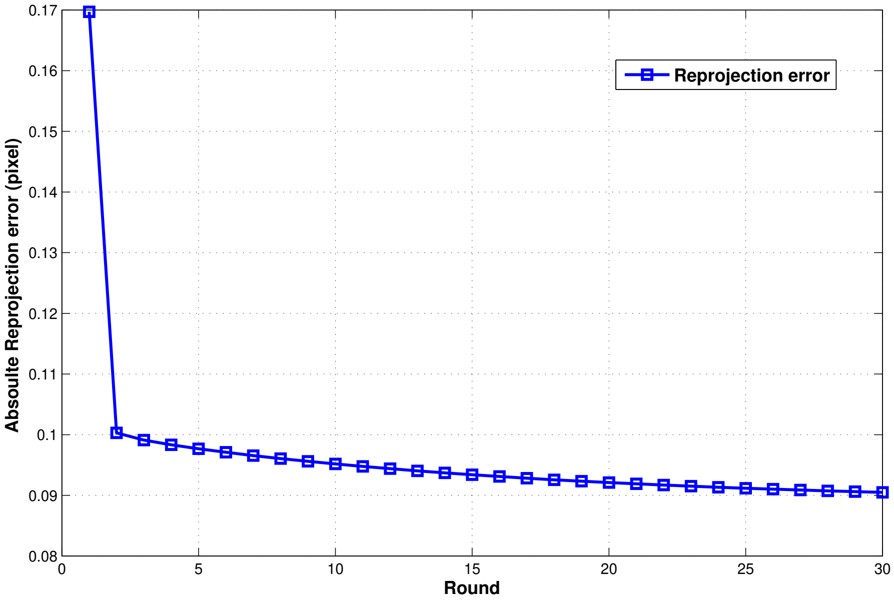


Figure 4.7: Re-projection error as a function of iteration numbers in iterative calibration

Chapter 5

Scanner Calibration

This chapter presents our 3D laser scanner calibration which related to camera calibration and the final 3D reconstruction. The scanner calibration aims to estimate two most important parameters: laser stripe projection angle α and baseline length D. Assuming the baseline D, the laser projection angle α and camera intrinsic and extrinsic parameters are known, then the 3D reconstruction should be straightforward by the trigonometry. However, these two parameters can not be measured directly and there is no special tools designed for that. A new approach to estimate the angle α and baseline length D is proposed in the chapter. Once the laser stripe location P_i in captured image is known, then P_i can be reversed back to 3D world space by simple triangulation. In addition, in this chapter, we also discuss some camera calibration practical issues and propose their resulting solutions.

5.1 Laser Camera Triangulation principles

To recover the geometrical structure of the visible surface of the objects of interest from their 2D images, the laser-based optical triangulation method is used in the 3D scanning system.

Fig.5.1 depicts the optical geometry of a scanning system [3]. Similarly, the laser stripe can be treated as many dots, each dot captured in the image can be reversed from 2D (x, y) to 3D (X, Y, Z). The relationship among the parameters of interest in this optical probe can be found by trigonometry. Two angles (α, β) and an edge (baseline D) determines this triangle completely. By simple trigonometry, the (X, Y, Z) coordinates of the illuminated point on the object can be estimated from its image point. If the laser dot in 2D image has coordinates (x_i, y_i) where x_i is larger than image centre C_x and $\beta < 90°$, then X_i, Y_i, Z_i are determined by Eq. 5.1.1, 5.1.2 and 5.1.3 below:

$$Z_i = \frac{Df}{p + f\cot(\alpha)} \tag{5.1.1}$$

$$Y_i = \frac{Z_i(y_i - C_y)}{f} \tag{5.1.2}$$

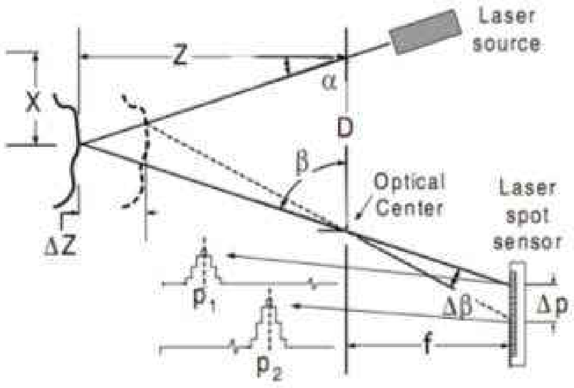


Figure 5.1: Laser-based optical triangulation [3]

$$X_i = X_0 + n\Delta x \tag{5.1.3}$$

where Δx is the linear stage moving step, n is the number of steps moved so far, and X_0 is the standing position. If $x_i < C_x$ and $\beta > 90°$, then (Z_i, Y_i) coordinates can be computed as:

$$\beta = \arctan \frac{p}{f} \tag{5.1.4}$$

where p is the laser dot in camera's CCD sensor, $p = |x_i - C_x|$, β is the angle between the CCD sensor plane and reflected laser beam, and α is the incident angle (or fixed angle) of the laser beam/stripe. Hence, Z_i

$$Z_i = \cos(\alpha)\left(D\tan(\alpha) + \frac{D\cot(\alpha)}{\tan(\alpha - \arctan(\beta))}\right). \tag{5.1.5}$$

Once Z_i is known, Y_i and Z_i can be computed by:

$$Y_i = \frac{Z_i(y_i - C_y)}{f} \tag{5.1.6}$$

$$X_i = X_0 + n\Delta x \tag{5.1.7}$$

The reason why different formulas were used to estimate 3D points coordinates is when the angle β is larger than 90°, the altitude of a triangle is outside, resulting in Z_i being estimated using another formula. X_i is related to linear stage moving step. For an incremental change of distance ΔZ, one measures the incremental angle shift $\Delta\beta$. The angular shift $\Delta\beta$ caused by the displacement of the surface is observed through a shift in the laser dot position $\Delta p = (p_1 - p_2)$. In practice, the main errors come from the estimated measurement of p and baseline length D.

In our case, the laser and camera are placed like Fig.5.3. Ideally, they should be in the same plane on the top of the linear stage. Since the triangulation described in Fig.5.1

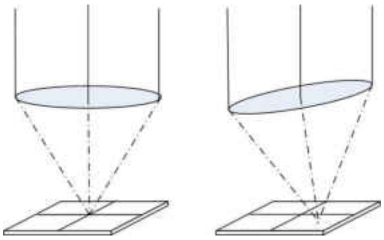


Figure 5.2: Ideal (left) and actual (right) lens position

is based on laser dot model, to be able to apply to laser stripes, it should be projecting parallel to Y coordinate in camera CCD plane as shown in Fig.5.3. Meanwhile, the camera should be placed at a projection angle at 90° exactly, otherwise CCD sensor is not parallel to the base line D, the angle β is meaningless and the formulas above can not apply any more.

Usually, the camera lens optical centre does not always lie on the line perpendicular to and pass through the centre of the CCD sensor plane. It means when the light rays go through the optical lens, they do not exactly focus on the CCD sensor's plane centre. This is the main reason why the calibrated image centre (C_x, C_y) always shifting a bit as shown in Fig.5.2. The whole triangulation accuracy highly depends on the laser projection angle α and baseline D, which is the distance between the camera lens's centre to the laser generator. To have accurate result, their measuring error should be as small as possible. However, it is extremely hard to measure them directly. Therefore, in the next section, a new method is proposed to estimate of these parameters and make full use of the accurate linear movement stage.

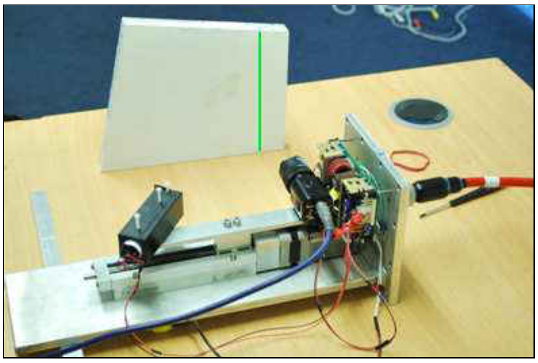


Figure 5.3: Laser scanning setup in air

5.2 Laser stripe Angle and Base Line Estimation

To be able to apply the triangulation method described in the previous sections, parameter baseline length D and laser projection angle α should be precisely measured. In practice, it is challenging to measure these parameters directly due to the underwater environment. [63] proposed a method using lithography film for reference of measurements. They used the **Z** axis to calculate the laser beam angle α by taking two pictures with the same camera. Firstly, they turned on the lasers and pasted the film on the scanner table. Then they took two pictures as shown in Figure 5.4. Finally, they processed these two pictures to find the distance between the centres of the laser stripes with consideration of the fixed film in the pictures and the known centre positions of the circles on it.

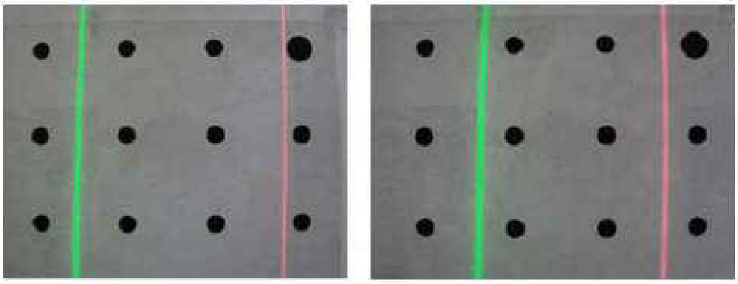


Figure 5.4: Two pictures used to calculate the laser beam angle in [63]

Alternatively, it is possible to use chessboard pattern like 5.12 and colour camera to compute the laser projection angle α. Firstly, we take couples of pictures with laser stripe projecting on the chessboard. Secondly, by the camera calibration parameters, it is able to calculate the chessboard 3D positions and then obtain the laser stripe position in 3D. Once we get this information, the laser projection angle α and tilt angle to camera **Y** axis can be estimated. In this case, it doesn't require the laser stripe perfectly aligned as long as the stripe itself is straight.

However, both of the above methods require either colour camera or secondary light source for the chessboard or lithography film to separate the laser stripe from the calibration pattern background. Most of our cases require mono camera (due to colour aberration underwater) and no other light source can be used. If we use other light source, the laser may become too weak to separate it from the background. If increasing the laser power, the image may become highly saturated. Light filter sometimes may help to reduce such saturation but it also filters out the background pattern that causes the above estimation methods to fail. Hence, a novel method that makes full use of highly accurate linear stage and mono camera is proposed:

1. Mount the scanning system onto the linear stage, pointing the laser and camera to the stage travel direction

2. Turn on the system and capture image to extract the laser stripe. Then the angle β_1 can be computed by the camera's focal length from the calibration. It is able to compute average angle β_1 to reduce the noise effect by multiple images which captured in different times.

3. Control the stage moving step $\triangle d_1$ and capture another image and then angle β_2

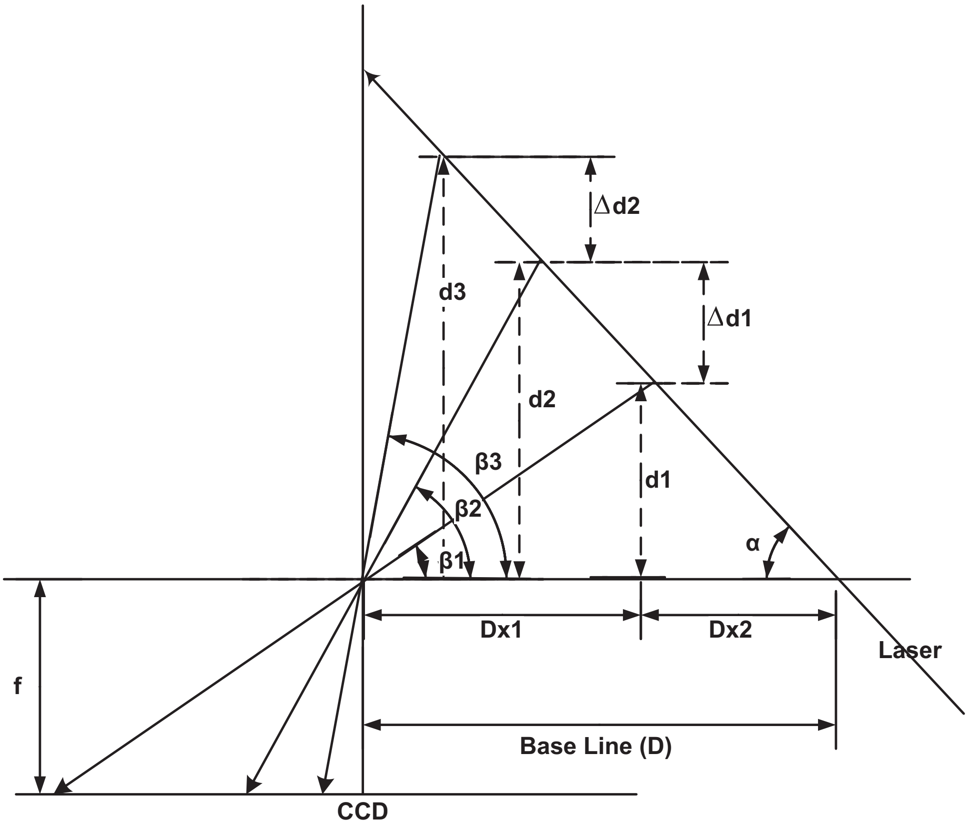


Figure 5.5: Principle of laser-based optical triangulation

can be computed. Again, it is able to compute average angle β_2 to reduce the noise effect by capturing multiple images.

4. Repeat step 3 to get $\triangle d_2$ and angle β_3.

As D and α are unknown, moving distance $\triangle d1$ and $\triangle d2$ can be read from linear stage, f is the focal length. Based on the triangulation, it is known: $\tan(\beta_1) = d_1/D_{x1}$ and $\tan(\alpha) = d_1/D_{x2}$, where $D_{x1} + D_{x2} = D$, hence,

$$D = d_1 \left(\frac{1}{\tan(\beta_1)} + \frac{1}{\tan(\alpha)} \right), D = d_2 \left(\frac{1}{\tan(\beta_2)} + \frac{1}{\tan(\alpha)} \right) \tag{5.2.1}$$

then

$$d_1 = \frac{D}{\left(\frac{1}{\tan(\beta_1)} + \frac{1}{\tan(\alpha)} \right)} \tag{5.2.2}$$

similarly, the d_2 can be derived as

$$d_2 = \frac{D}{\left(\frac{1}{\tan(\beta_2)} + \frac{1}{\tan(\alpha)} \right)} \tag{5.2.3}$$

Also, $d_2 - d_1 = \triangle d_1$, where d_1 is the first round linear stage movement, hence, the following equation holds:

$$D = \frac{\triangle d_1}{\frac{1}{\frac{1}{\tan(\beta_2)} + \frac{1}{\tan(\alpha)}} - \frac{1}{\frac{1}{\tan(\beta_1)} + \frac{1}{\tan(\alpha)}}} \tag{5.2.4}$$

Repeating the same steps as before, mark down the linear movement $\triangle d_2$ and it can be inferred:

$$D = \frac{\triangle d_2}{\frac{1}{\frac{1}{\tan(\beta_3)} + \frac{1}{\tan(\alpha)}} - \frac{1}{\frac{1}{\tan(\beta_2)} + \frac{1}{\tan(\alpha)}}} \tag{5.2.5}$$

Combining the last two equations, unknown variable D can be removed, hence,

$$\frac{\triangle d_2}{\triangle d_1} = \frac{\frac{1}{\frac{1}{\tan(\beta_3)} + \frac{1}{\tan(\alpha)}} - \frac{1}{\frac{1}{\tan(\beta_2)} + \frac{1}{\tan(\alpha)}}}{\frac{1}{\frac{1}{\tan(\beta_2)} + \frac{1}{\tan(\alpha)}} - \frac{1}{\frac{1}{\tan(\beta_1)} + \frac{1}{\tan(\alpha)}}} \tag{5.2.6}$$

In this equation, only $\tan(\alpha)$ is unknown and can be solved. Once it has been solved, the baseline length D can be determined. In real experiments, the linear stage movement can be alike to simplify the equation which means $\frac{\triangle d_2}{\triangle d_1} = 1$. Surely, repeat the process to obtain average would give better optimization results.

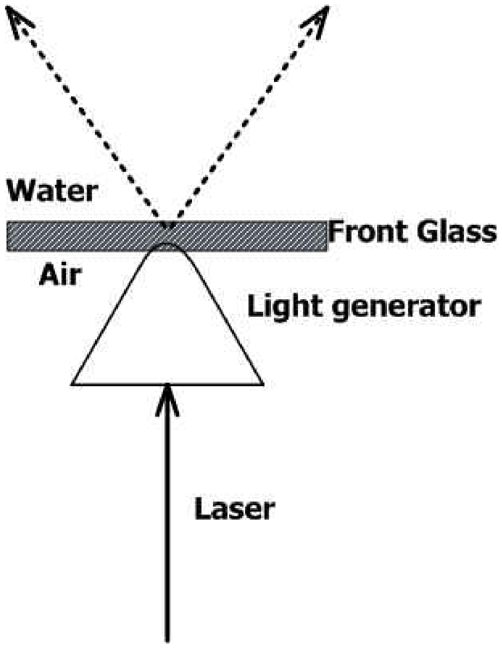


Figure 5.6: Laser generator

5.3 Some Practical issues

5.3.1 Adjust the laser generator

It is important to make sure the laser stripe itself is straight. Otherwise, the laser stripe bending can cause laser dot triangulation model in Fig.5.1 fail to extend as laser stripe model and such geometry trigonometry is no longer stand. As shown in Fig.5.6, adjust the laser and the light generator to make sure the laser itself is straight. There is a way to verify whether the laser is bent: turn on the camera and take a snapshot of laser stripes and perform image un-distortion. Fig.5.7 shows the snapshot and its undistorted image. Once we get the laser peak, the least-squares technique [1] can be applied:

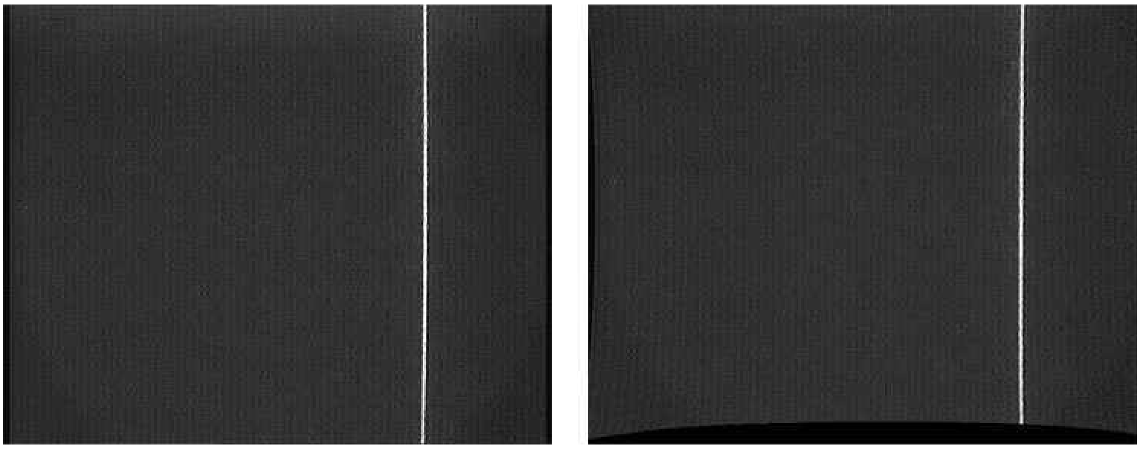


Figure 5.7: Laser stripe adjustment

Certainly, to verify this laser stripe straight or not, each sample point of the peaks $x_i, y_i, i = 1, \dots n.$ can be fit to a straight line model $y = a + bx$ with a total error E defined in Eq.5.3.1 as small as possible. Also, the correlation coefficient R^2 defined in Eq.5.3.11 gives the quality of this least squares fitting to the original data.

$$E(a,b) = \sum_{i=1}^{n}[y_i - (a + bx_i)]^2 \tag{5.3.1}$$

and

$$\frac{\partial(E)}{\partial a} = -2\sum_{i=1}^{n}[y_i - (a + bx_i)] = 0 \tag{5.3.2}$$

$$\frac{\partial(E)}{\partial b} = -2\sum_{i=1}^{n}[y_i - (a + bx_i)]x_i = 0 \tag{5.3.3}$$

---

[1] `http://mathworld.wolfram.com/LeastSquaresFitting.html`

These lead to the equations

$$na + b\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i \tag{5.3.4}$$

$$a\sum_{i=1}^{n} x_i + b\sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i y_i \tag{5.3.5}$$

Solving these equations by matrix inverse, we have

$$a = \frac{\bar{y}(\sum_{i=1}^{n} x_i^2) - \bar{x}\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i^2 - n\bar{x}^2} \tag{5.3.6}$$

where $\bar{x} and \bar{y}$ are the average of x_i, y_i, then

$$b = \frac{(\sum_{i=1}^{n} x_i^2 x_i y_i) - n\bar{x}\bar{y}}{\sum_{i=1}^{n} x_i^2 x_i^2 - n\bar{x}^2} \tag{5.3.7}$$

By [43], they can rewrite by defining the sum of squares:

$$ss_{xx} = \sum_{i=1}^{n}(x_i - \bar{x})^2 \tag{5.3.8}$$

$$ss_{yy} = \sum_{i=1}^{n}(y_i - \bar{y})^2 \tag{5.3.9}$$

$$ss_{xy} = \sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}) \tag{5.3.10}$$

Then, the correlation coefficient R^2 is finally defined as:

$$R^2 = \frac{ss_{xy}^2}{ss_{xx} ss_{yy}} \tag{5.3.11}$$

In the example as shown in Fig.5.7, its correlation coefficient $R^2 = 0.99996$, which means 99.996% of the data that is the closest to the best fit line. Thus, laser stripe is straight in the camera view. Such calculation it is in real time and the laser generator can be adjusted carefully until the correlation coefficient R^2 as close to 100% as possible.

5.3.2 Laser tilts versus camera Y-axis

Laser tilts versus camera Y axis is one of the parameters of the scanner which should be corrected. The laser triangulation model in Fig.5.1 is based on laser dot. To be able to apply to laser stripe and have accurate scanning, the laser stripe should be parallel to the camera Y axis. Hence, it needs very precisely assembling of scanner parts. However, it is possible to compensate laser tilt error from the assembly of the system.

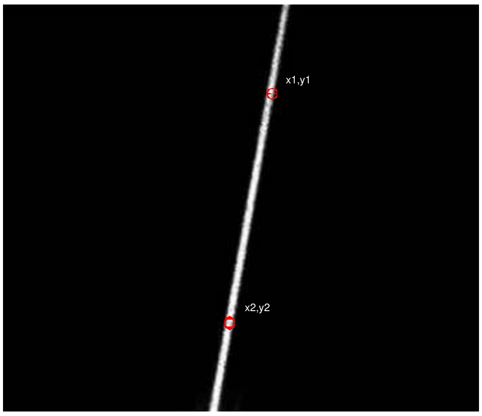


Figure 5.8: Laser strip tilt

The laser tilt can be calculated by linear least-squares fitting. Actually, its tilting angle is the slop of its best fitting line as long as the stripe is straight. Of course, the captured image should be un-distorted in the beginning. Such procedure can be performed in air before the scanner applied underwater. Fig.5.8 shows a tilt laser stripe, its slope is the tilting angle to the camera Y axis, which can be compensated by Eq.5.3.12, where κ is the laser tilt.

$$\begin{aligned} \kappa &= \tfrac{x_2-x_1}{y_2-y_1} \\ x' &= x'_{tilt} - \kappa \times y' \end{aligned} \tag{5.3.12}$$

where $(x_1, y_1), (x_2, y_2)$ are two select points on the laser stripe, x', y' are their corrected point. Such laser tilt calculation can be used in air laser scanning which allow using third part light source, colour laser or even lithography film. In underwater environment, we still need to make sure the laser parallel to camera Y axis as much as possible then our laser stripe and baseline estimation method can be applied.

5.3.3 Camera and linear stage Y coordinate mismatch

The mismatch between the XY axes of the camera and linear stage may happen as illustrated in Fig.5.9. To correct this, we can put a perpendicular white string with heavy metal piece at the end of it. By gravitational attraction, the string will stabilize and straight enough to be the reference. Calculating the slope of the line in the image by the linear least square fitting will tell us how much the camera XY axes have been bent relative to the linear stage, and the camera can then be adjusted to make it perfectly fit with the Y axis vertical to the ground.

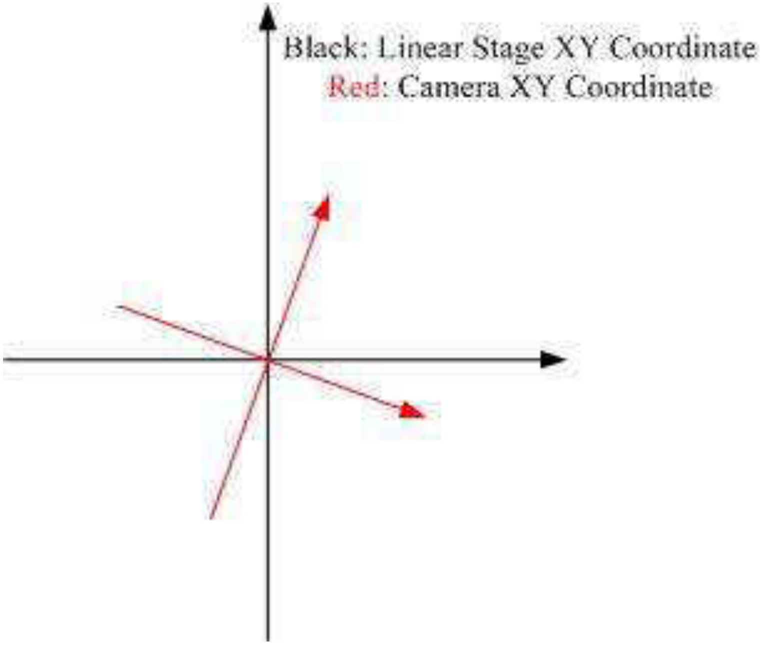


Figure 5.9: XY coordinates mismatch

5.3.4 Angle λ

The angle between the X axis of the linear stage and that of the scanner system as shown in Fig.5.10 would be anther parameter to be calculated. When placing the scanner on the top of the linear stage, it is not easy to make sure that their X-axes are the same. The actual moving distance X' of the scanner needs to be compensated by Eq.5.3.13 as a function of the moving distance X of the linear stage and the angle λ between them.

$$X' = \frac{X}{\cos(\lambda)} \tag{5.3.13}$$

There are two ways to compensate or correct the angle λ as shown in Fig.5.10 and 5.11:

1. Directly measure the distance D_1 and D_2 from both ends of the linear stage of the board which is vertical to the ground, as shown in Fig.5.11 to make sure $D_1 = D_2$, thus, $\lambda = 0$.

2. Place a chessboard on the board and capture its image, then perform corner detection, using camera calibration parameters to un-distort the image and the corners.

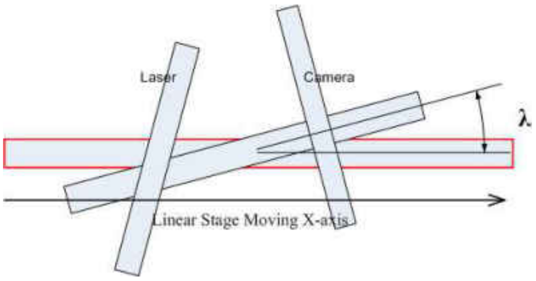


Figure 5.10: The angle between the linear stage and camera in X-axis, which is due to their imprecise assembly.

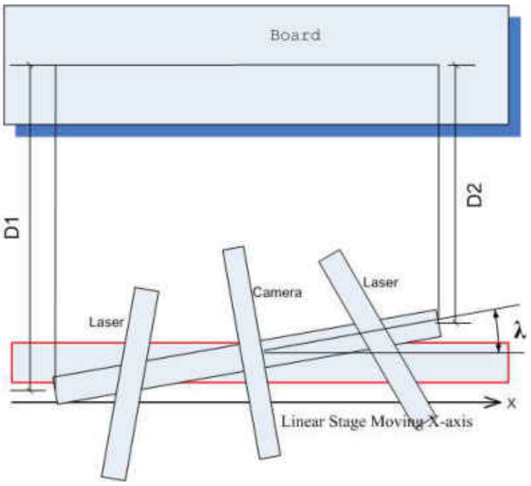


Figure 5.11: Compensate for the angle λ

> The board and chessboard can be adjusted to vertical to the ground carefully, thus the formed lines by the corners on the captured image should be parallel or vertical to each other. Ideally, if the camera Y axis is parallel to the board, the grid made by these corners should have the same size.

Clearly, these two ways can be combined together. One possibility is to adjust the linear stage directly as the first method and then use camera calibration and linear fitting to check if the camera is parallel to the chessboard.

Figure 5.12: Using a chessboard to measure the tilt of the camera.

5.4 Underwater Scanning issues

Water Index There are many factors change the water index, which means that the light refractive angle may change, and thus affect the camera calibration result. It is easier to conduct experiments using an air/water model to obtain the index of refraction of the light relative to the air or glass. The index of refraction is the ratio of the travel velocity of the light in vacuum to that in water: light travels more slowly in water than in vacuum (or in the air). Thus, it is always greater than one. In our case, the laser beam travels from seawater, goes through the front glass and passes the camera lens to CCD through the air. Obviously, the ingredient of the seawater is complex. It includes: temperature, salinity and pressure/stress. Generally, the first two factors are more important.

Theoretically, in various degrees, all transparent media are dispersive. Thus, they bend light as a function of its wavelength. Specifically, in the visible portion of the spectrum (approximately 4300-6900 Angstroms) the index of refraction is generally a decreasing function of the wavelength: the violet light is bent more than the red light. Furthermore, the rate of change of the index of refraction also increases as the wavelength decreases. Table 5.1 shows the results of some measurements of the index of refraction of water, $n(w)$, with respect to the dry air at the same temperature T as water and a pressure of $760mm - Hg$.

Wave Length (Angstroms)	$T = 10°C$	$T = 20°C$	$T = 30°C$
7065	1.3307	1.3300	1.3290
5893	1.3337	1.3330	1.3319
5016	1.3371	1.3364	1.3353
4047	1.3435	1.3427	1.3417

Table 5.1: Index of the refraction of water as a function of the wavelength and water temperature

Another factor is the salinity of the seawater. It does have some effect on the light refraction. In the huge part of the ocean that remains hidden, the seawater is salty, cold, dark and deep. Salinity is usually expressed as grams of salts dissolved in a kilogram of seawater (gm/kg). Average salt content in the ocean is 35 grams per kilogram of the seawater. Table 5.2 shows how the seawater index $n(w)$ increases with regards to salinity for the sodium D-lines (mean: 5893 Angstroms) at 18° [19].

Salinity (gm/kg)	increase in n(w)	example
5	0.00097	northern Baltic Sea
10	0.00194	
15	0.00290	
20	0.00386	bight of Biafra
25	0.00482	
30	0.00577	
35	0.00673	Atlantic surface
40	0.00769	northern Red sea

Table 5.2: Changes in index of refraction due to salinity

The index of refraction is also a function of the water pressure, but the dependence is quite weak because of the relative incompressibility of the water. In fact, over the normal ranges of temperatures (0-30C), the approximate increase in $n(w)$ is 0.000016 when the water pressure increases by one atmosphere [2]. Clearly, the most significant factors that affect $n(w)$ are the wavelength of the light, salinity and temperature of the water. To reduce their effect on the detection accuracy of the laser spot in the image and simulate the working environment of the 3D scanner in subsea, the salinity, average temperature and wavelength of the laser scanning should be measured in advance. A calibration and optimization is done in the water with the same salinity and temperature which remain the index consistent.

5.5 Underwater camera calibration

[77] clearly describes the principle of the normal camera model as presented in Chapter.4.3. The perspective geometry model in the air would be very different from that

---

[2] Atmosphere, unit of pressure: the pressure that will support a column of mercury with 760mm high at sea level and 0 degrees centigrade

underwater. Here the perspective projection model is shown again in Fig.5.13. Fig. 5.14 shows the perspective model underwater as comparison. Apparently, the viewpoint (with dash line) at the bottom is following the normal perspective projection. However, the real viewpoint is moved forward due to the light refraction of the water. Hence, the application of the perspective model underwater may cause error.

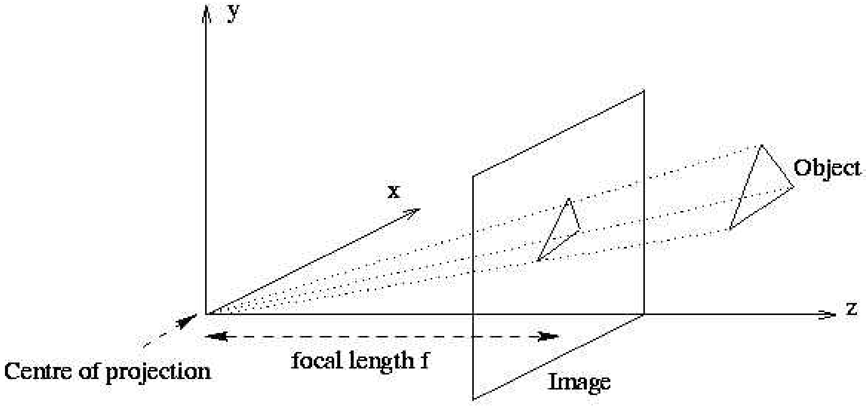


Figure 5.13: Perspective projection in the pinhole camera model

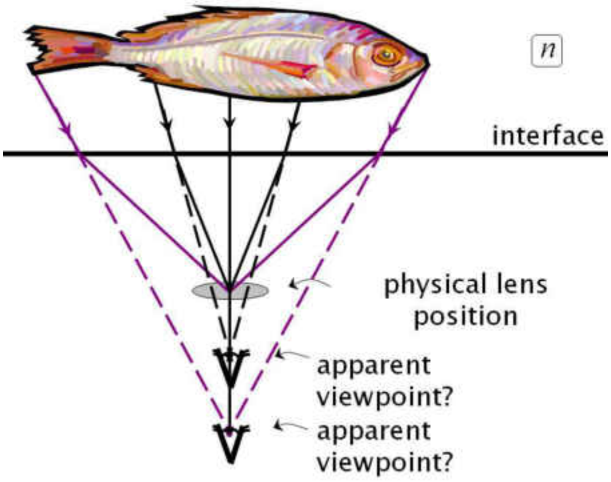


Figure 5.14: Perspective projection model in an underwater environment [77]

Since no standard manufacturing data sheet will provide the camera focal length based on water, it is hard to find a ground truth reference to verify the underwater camera calibration results. However, the normal calibration based on the perspective model may apply to underwater with some corrections. The detail is shown in later section.

5.6 Optimization

To sum up, the angles (α, β), water refraction, lens and CCD position are uncertain factors. Their measurements from the direct survey are not always accurate enough and can easily go wrong. Hence, a new method is proposed to optimize the measurements. It is performed after parameters have been measured and estimated. The method follows the basic triangulation principle: every $2D$ point in the CCD sensory plane or every pixel in the captured image can be referred to a real $3D$ point. If all the factors including the camera's parameters (focal length f, distortion factors) and triangulation factors (angles (α, β), base line D) are precisely measured, the relationship between $2D - 3D$ should be fixed. If not, then the error can be minimized by comparing the real measurement and estimated values. The main steps for the optimization are as follows:

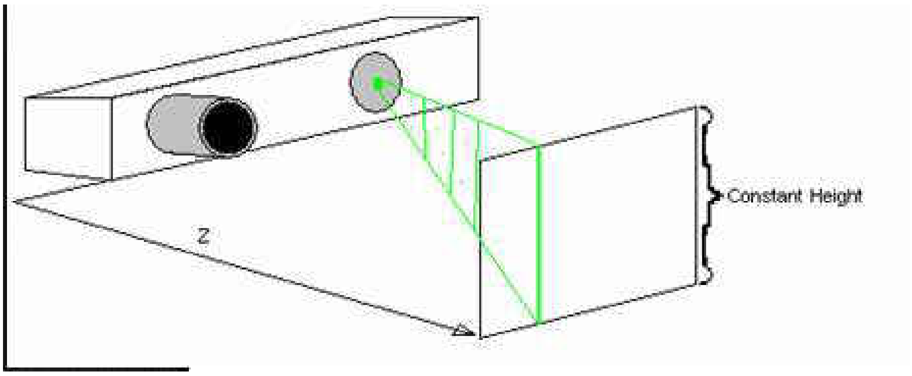


Figure 5.15: Triangulation optimization

1. Mount the laser scanner onto the linear stage. The stage movement is controllable and the minimum step is $0.16um$. Real measurement can be read directly from software.

2. Adjust the laser holder and camera by gradienter to make sure that they are horizontal to the linear stage. At the same time, the laser line should be projecting vertically to the linear stage and the CCD sensor in the camera. This also means that the CCD sensor is vertical to the linear stage.

3. Place a flat object such as a white board vertically in front of the scanner as Fig.5.15

4. Control the linear stage moving forward/backward to the board by a small step of $0.1mm$, for example, record the distance Z from camera's CCD sensor to the board, take a picture in every step of the stage movement. As the linear stage moves, the position in the white board hit by the laser line also changes. The value Z can be estimated by Eq.5.1.1 or Eq.5.1.5. The actual Z can be directly

read from the stage movement. Hence, every estimated Z_e can be corrected by $Z_c = f(Z_e)$.

5. The 3D scene parameter Y can be estimated from Eq.5.1.6, while X comes directly from the linear stage or any other motion sensor.

For certain distance, this $2D$ to $3D$ relationship is reliable. However, as the distance increases, the $3D$ location requires a sub-pixel accuracy in the CCD sensor. As the sensor's inherent capability (CCD resolution) is fixed, the relation is not reliable any more if it exceeds the certain distance. Apparently, to measure farther object accurately, a higher resolution camera should be adopted. Due to all those uncertain factors, the correction of the estimated parameters is normally a necessary step before actually outputting the data. A flat glass/air/water model is proposed in [77]. In our case, we consider distortion coming from the lens and the refraction is fixed, which means most of the distortion is radial. Table.5.3 shows the contrast between real and measured distance Z by the proposed estimation method. The actual measurement Z_a is the distance between the laser stripe to the flat wall. Since we can open the camera's front cap that Z_a is measurable as a reference value. In the example, the linear stage movement is set to $\triangle d = 10mm$, the camera resolution is 720×576, the angle of view is 90° in air, 60° in water, and focal length $f = 764.323$ obtained from the camera calibration method. Substituting these values into the formulas derived before yields baseline $D = 110.840mm$, $tan(\alpha) = 1.967$ and angle $\angle\alpha = 63.057°$.

Actual Z_a	Estimated Z_e	Error ΔZ
150.000	150.248	0.248
160.000	159.512	0.488
170.000	168.594	1.406
180.000	178.012	1.988
190.000	186.015	3.985
200.000	194.812	5.188
210.000	203.455	6.545
220.000	212.078	7.922
230.000	220.657	9.344
240.000	229.165	10.834
250.000	237.579	12.420

Table 5.3: Actual and estimated depth (unit: mm)

Apparently, as the distance increases, the error between actual Z_a and estimated Z_e distance becomes bigger. Fig.5.16 shows that this error is non-linear but fits the second order polynomial with correlation coefficient of 0.9999. Thus, the estimated Z_e can be corrected by the 2nd order polynomial function about the actual Z_a and estimated Z_e. For instance, in the case of Fig.5.16, the correction function is as follows:

$$Z_c = 0.0006 \times Z_e^2 + 0.9166 \times Z_e - 1.462 \tag{5.6.1}$$

where Z_c and Z_e are the distance before and after correction. By this correction, the distance error can be reduced down to $\pm 0.5mm$. However, this fitting can be extended to a higher order, for instance, in 4th order,

$$Z_c = 0.0000001861 \times Z_e^4 - 0.0001535 \times Z_e^3 + 0.04753 \times Z_e^2 - 5.388 \times Z_e + 312.4 \tag{5.6.2}$$

with correlation coefficient of 0.99996, which is very similar to second order fitting. Hence, as long as the correlation coefficient is close to 100%, this fitting is acceptable, no matter in which order. Such correction is necessary and useful taken into account the various uncertain factors in the calibration and measurements of the focal length of the digital camera, principal point, the length of the baseline, among others.

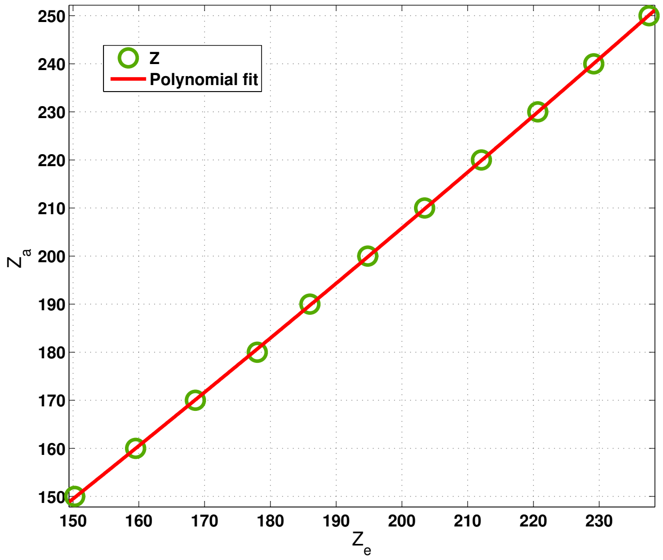


Figure 5.16: Z_a, Z_e and their polynomial fitting

Fig.5.17 shows an example of 3D scanning before and after optimization. The blue and red parts are from different viewpoints of scanning (from the dual laser scanner, left and right laser). Before the optimization, they are displaced and the measurements are not very accurate. In principle, if the 3D reconstructions are accurate, both of them should be superimposed perfectly because we are using the same camera, the object is still but the lasers are shutting at different angles. Obviously, after the optimization, their superimposition is perfect with precise measurement.

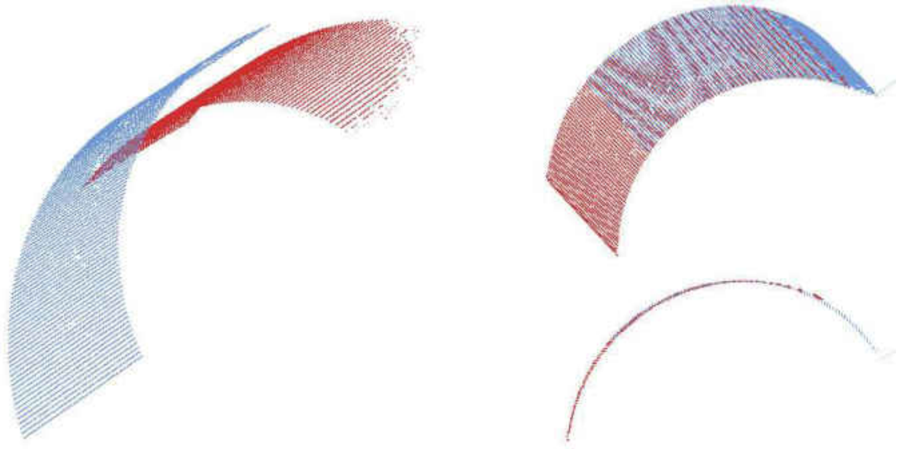


Figure 5.17: Left: mismatch 3D profile, Right: match 3D profile after optimization

Chapter 6

Experimental results and analysis

Our 3D Laser scanner is a whole system composed of several different components. Each component plays an important role in the entire system. It is interesting and important to know each component's impact on the final scanning results. To this end, a systematic approach is adopted for the comparison and study of the effect of a particular component on the final 3D reconstruction. This is in contrast with the previous study: different components are investigated independently. While a particular component is controlled, all others will remain the same. The 3D reconstruction error is measured as the relative error between the ground truth and the estimated one. The ground truth is collected from direct and careful measurement. Individual experiment is carried out based on different components including:

1. Sparse Code Shrinkage image denoise
2. Corner detection
3. Laser peak detection
4. Fraction lens distortion model
5. Iterative camera calibration

Synthetic and real scanning data were used in these experiments. All the experiments are evaluated based on the final 3D result. The final experiment combines all these selected components into the system and compares the 3D result to the design without the selected components. Hence, the entire experiment is organized as two parts: from section 6.1 to section 6.5 are the independent investigation of different components and methods. Section 6.6 investigates the difference on 3D reconstruction between the selected methods/components and traditional methods so that whether such optimization is effective can be investigated.

6.1 Sparse Code Shrinkage image denoise

In this section, a comparative study is carried on the denoising methods using natural and underwater scanned images from our 3D scanner to examine:

1. How noise affects image quality and its feature detection.
2. How noise affects the final 3d reconstruction.
3. Which method is the most effective to reduce noise on the final 3D reconstruction.

The widely and common used Gaussian filter, median filter and Wiener filter were chosen for the comparative study. The parameters of these filters were empirically chosen from the evaluation images.

6.1.1 Natural images

From Fig.6.1 to Fig.6.4 show the denoising results on a grasshopper and the famous "Lena" image corrupted by Gaussian noise with a standard deviation of $\sigma = 0.3$ and $\sigma = 0.5$ (the unity was the standard deviation of a given image). In these figures, Gaussian noise was added as illustrated in the right top image. The next two rows show the denoising results by Gaussian, median, Wiener filter and SCS respectively. The Sparse Code Shrinkage method was applied using the estimated orthogonalized ICA transform (8×8 windows), while Gaussian smoothing with 3×3 masks, median and wiener filtering in 3×3 neighborhood at noise level $\sigma = 0.3$. When the noise level increase to 0.5, all those masks and neighborhood increase to 5×5 to have a better result.

Visually, even at heavy noise level $\sigma = 0.5$, the SCS still produces better noise reduction and retaining the image features. The perceptual quality of SCS is significantly better than the remaining ones and its denoise result is qualitatively very like those for the lower noise level $\sigma = 0.3$. It seems it performs quite similar as a feature detector, in that it retains those features that are clearly visible in the noisy data but cuts out anything that is probably a result of the noise[38]. The Wiener filter gives fair results, although it does not really eliminate the noise but more like image blurring. Visually, Gaussian and median filter distorts the images important feature and sharpness. To have a quantity measurement of these filters' effect on the final 3D result, we compared them based on the real data from our 3D scanner in the next section.

6.1.2 Real scanning data

It is important to know the influence of noise to the final 3D reconstruction accuracy and the image denoising methods' performance in the difficult imaging task. Therefore, we evaluated them based on the real data. The equipment set up is illustrated as Fig.6.5. There are two scanning targets in this scanning experiment: (1) Vertical flat wall and (2) Cup. They include different complexities of geometry and thus can be used to test

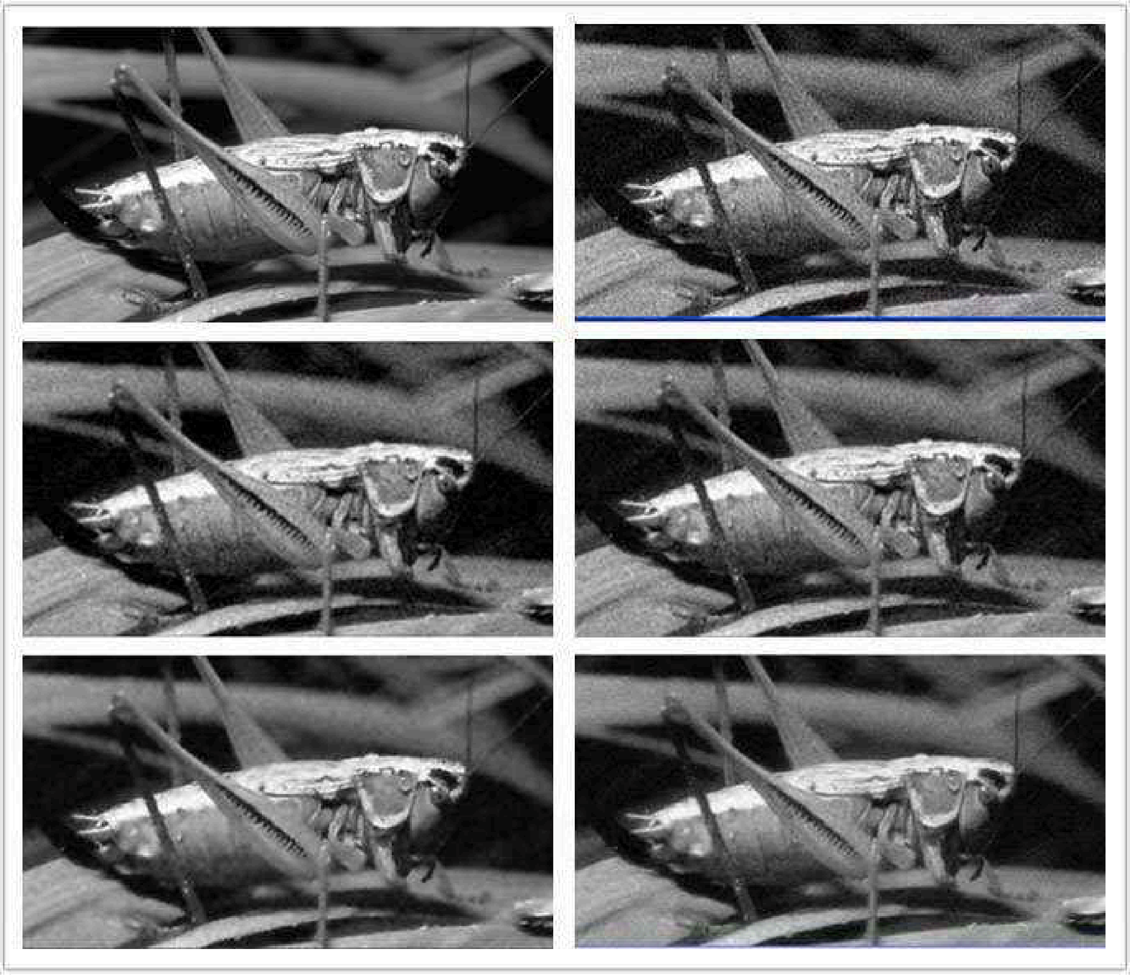


Figure 6.1: Result of denoising a grasshopper image with different methods when the noise level is $\sigma = 0.3$. The Sparse Code Shrinkage method was applied using the estimated orthogonalized ICA transform (8×8 windows), while Gaussian smoothing with 3×3 masks, median and wiener filtering in 3×3 neighborhood. Top left: original image, top right: noise added, second row left: Gaussian smoothed, second row right: median filtered, third row left: wiener filtered, third row right: SCS applied.

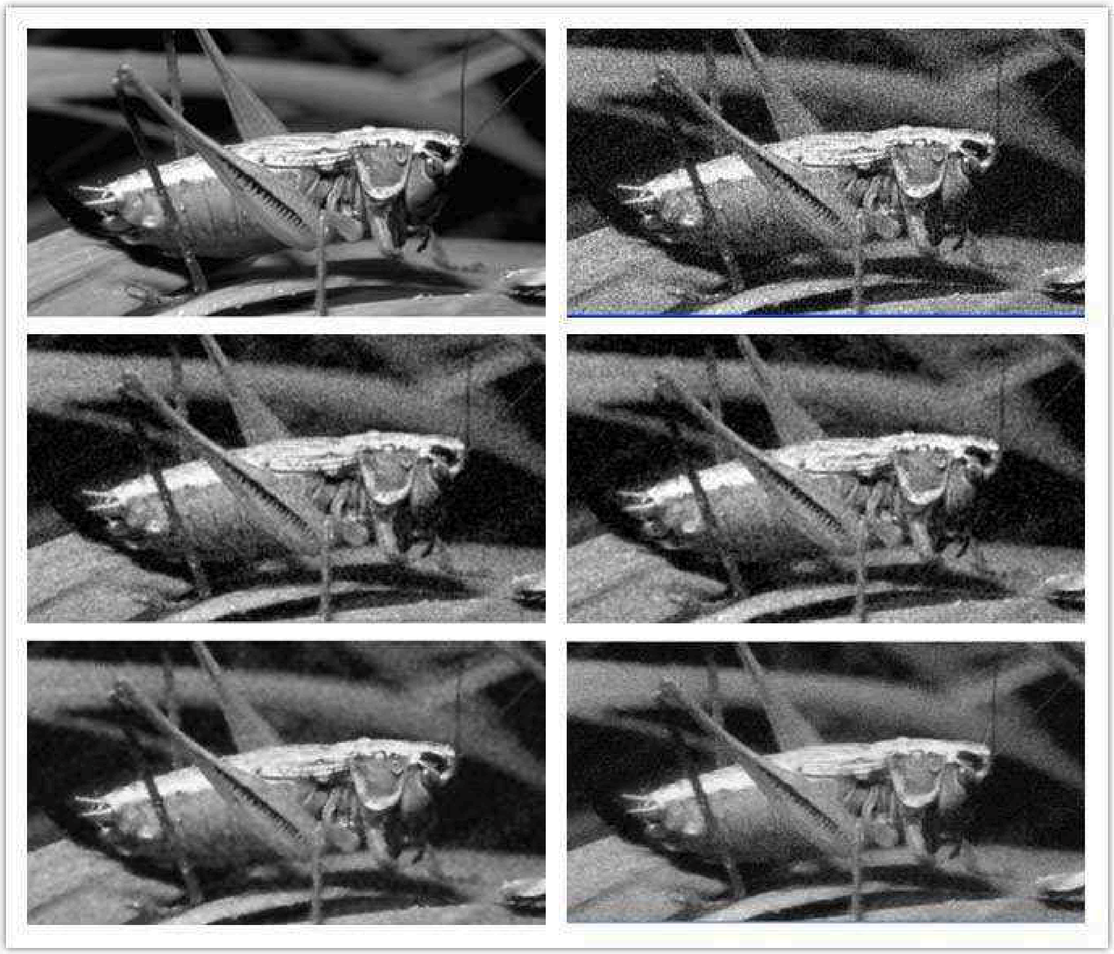


Figure 6.2: Result of denoising a grasshopper image with different methods when the noise level is $\sigma = 0.5$. The Sparse Code Shrinkage method was applied using the estimated orthogonalized ICA transform (8×8 windows), while Gaussian smoothing with 5×5 masks, median and wiener filtering in 5×5 neighborhood. Top left: original image, top right: noise added, second row left: Gaussian smoothed, second row right: median filtered, third row left: wiener filtered, third row right: SCS applied.

Figure 6.3: Result of denoising 'Lena' with different methods when the noise level is $\sigma = 0.3$.The Sparse Code Shrinkage method was applied using the estimated orthogonalized ICA transform (8×8 windows), while Gaussian smoothing with 3×3 masks, median and wiener filtering in 3×3 neighborhood. Top left: original image, top right: noise added, second row left: Gaussian smoothed, second row right: median filtered, third row left: wiener filtered, third row right: SCS applied.

Figure 6.4: Result of denoising 'Lena' with different methods when the noise level is $\sigma = 0.5$. The Sparse Code Shrinkage method was applied using the estimated orthogonalized ICA transform (8×8 windows), while Gaussian smoothing with 5×5 masks, median and wiener filtering in 5×5 neighborhood. Top left: original image, top right: noise added, second row left: Gaussian smoothed, second row right: median filtered, third row left: wiener filtered, third row right: SCS applied.

whether the proposed method is robust for 3D scanning. Planar, circular and cylindrical features are often used to characterize the performance of 3D laser scanners[53]. To eliminate any effect caused by other components such as camera calibration, the noise was only added into the laser stripe images and the evaluation was made only based on their 3D models.

6.1.2.1 Equipment Set Up

This is a test done underwater, all the equipment is waterproof. The camera is Bowtech LCC-600, with a resolution of 720 × 576. The laser projection angle is about 60°. The linear stage moving step is $1mm$. From the scanner to the wall distance is about $20cm$. The wall is painted in white and placed vertically to the scanner. Zhang ZhengYou[83] camera calibration and the proposed scanner parameters estimation procedure (including the baseline D and laser projection angle α) described in Chapter 5 were applied in this experiment. Gaussian approximation laser peak detection [22] method was used to make a fair comparison. Please note the global optimization method discussed in Section 5.6 was not used in both scanning objects for easy implementation. The accuracy may be lower because the normal camera calibration was applied in underwater without optimization. Underwater scanning often comes with heavier noise than in air. For example, Fig.6.6 is the one of the example images. The noise came from water(such as current, sediment, floating particles, sand, mud, clay and salinity) or electronic signal. In most of cases, it is very hard to avoid, hence, to simulate the real applications, different levels of artificial Gaussian noise were added to the images and then denoised by Gaussian filter, median filter, Wiener filter and SCS respectively. After that, laser peak is extracted by Gaussian approximate and reconstructions to 3D model. Each noise level runs 100 times and the average is chosen as the final result. Orthogonal linear regression is fitted to the flat wall 3D models and three measurements were picked from cup model for accurate evaluation.

6.1.2.2 Results and analysis: Vertical Flat Wall

In this part, the scanner was placed to scan a vertical flat wall, therefore, the reconstructed 3D model should be completely flat, too. It is able to estimate 3D model's roughness by fitting an orthogonal linear regression and calculate the perpendicular distance from the fitting to target model [1]. This is similar to the simple linear regression in 2D fits a straight line to a set of data points. Thus, the sum of perpendicular distances (or called orthogonal distances, SOD) and their sum of squared errors (SSE) between the fitting model and the original can be used to measure flat wall 3D model smoothness. Based on the 3D model's smoothness/roughness, it is very easy and sensible to evaluate the denoise method's performance. Fig.6.8 is the visualization of 3D plane fitting on one of the flat wall 3D models. The final generated 3D model contained over 30000 points.

---

[1] Access on 11th Feb, 2012, Fitting an Orthogonal Regression Using Principal Components Analysis http://www.mathworks.co.uk/products/statistics/demos.html?file=/products/demos/shipping/stats/orthoregdemo.html

In considering the 3D models have no redundancy filtering and the amount of 3D points, the SSE and SOD went larger since they are the summed up values.

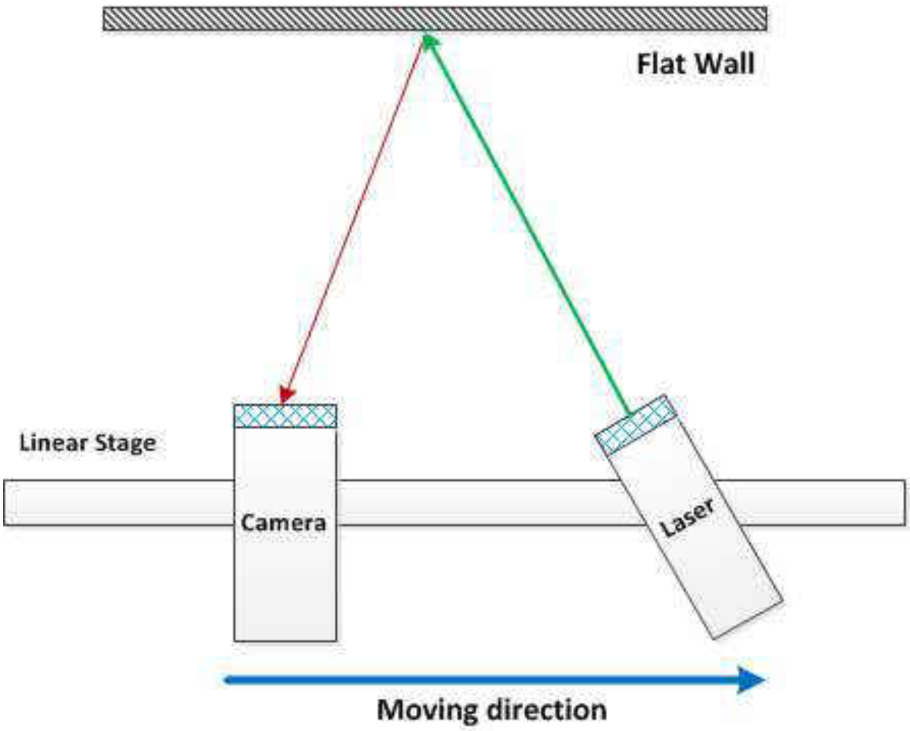


Figure 6.5: The scanner set up for scanning the flat wall

The fitting result is shown on Table.6.1. The first column is the noise level increases from 0.1 to 0.5. In the second column, SSE represents the sum of square errors and SOD represents the sum of orthogonal distances. The column 'Ref' is the reconstruction result by the laser peak without any noise removal technique applied. The next columns are the denoise result by Gaussian, median, wiener and SCS filters respectively. In Gaussian filter, σ_2 represents different smooth strength levels. The parameters of median, wiener and SCS filters have been adjusted to have the best denoise result.

When the noise increases, the SSE and SOD increase sharply. Even the noise level increases just 0.1, the SSE error increases almost 70%. Since we compared the result based on the roughness of the 3D model and the noise directly affects point 3D position that both SSE and SOD errors are very sensitive to the noise. Fig.6.9 and Fig.6.10 show the SSE and SOD errors at different noise levels. The red line is the "Ref" result which is without any denoising method applied. It can be observed from those results that:

1. Overall, SCS is the only method which actually brings down the errors except Gaussian filter(as $\sigma_2 \geqslant 0.6$). Wiener gives similar errors as the 'Ref', but the median filter results are far away from expected.

2. As the Gaussian smooth strength(σ_2) increases, the laser peak images are getting

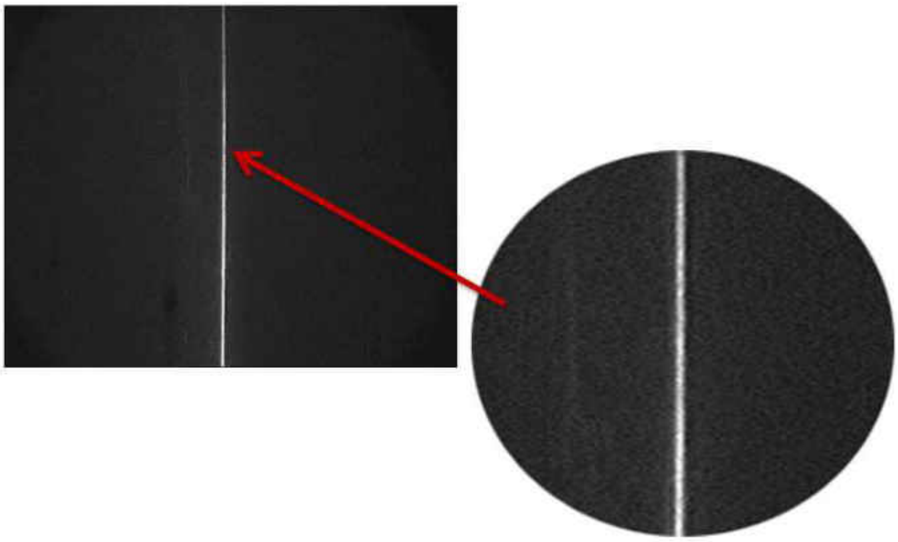


Figure 6.6: An example of captured laser images

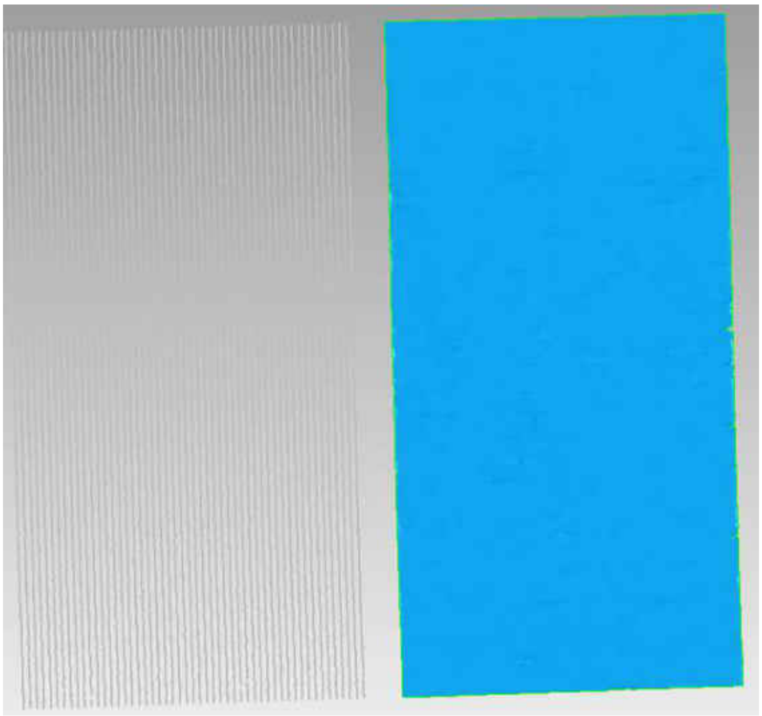


Figure 6.7: Flat wall 3D reconstruction. Left: The reconstructed point cloud from peak detection. Right: The mesh of point cloud

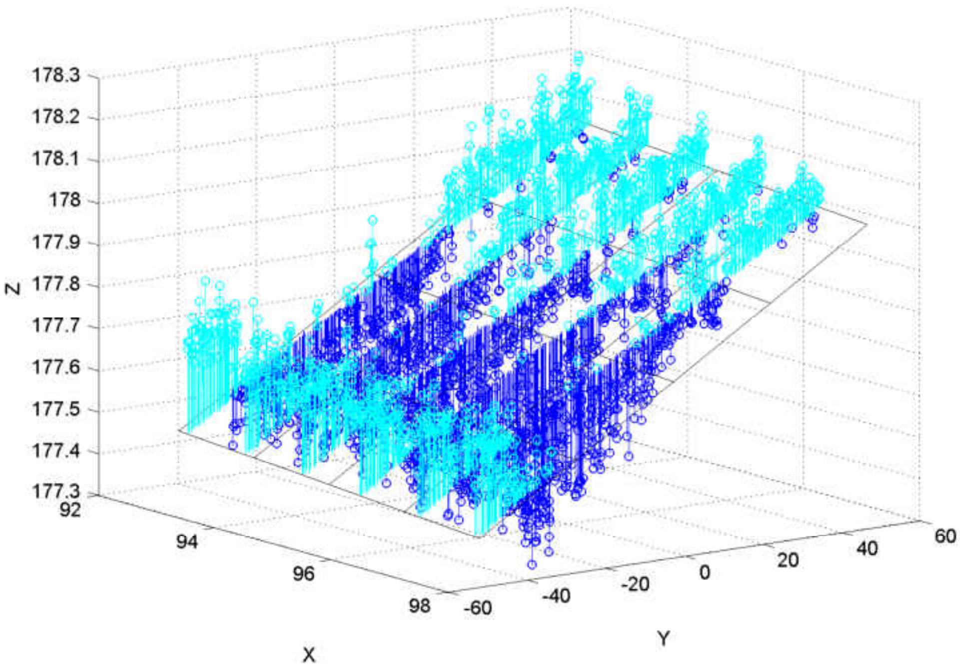


Figure 6.8: The visualization of 3D plane fitting on flat wall 3D model (1% of the total)

Table 6.1: Orthogonal linear regression fitting error in noisy 3D flat wall models. The Sparse Code Shrinkage method was applied using the estimated orthogonalized ICA transform (8 × 8 windows), while the mask in Gaussian smoothing and neighborhood in median and wiener filtering increase from 3 × 3 to 5 × 5 as the noise level increase. σ_1: Noise levels, σ_2: Gaussian smooth levels, SSE: sum of the squared errors, SOD: sum of orthogonal distances

σ_1		Ref	Gaussian			Median	Wiener	SCS
			$\sigma_2 = 0.3$	$\sigma_2 = 0.6$	$\sigma_2 = 0.9$			
0.0	SSE	391.15	-	-	-	-	-	-
0.1		663.83	660.69	620.51	603.88	860.12	664.56	630.52
0.2		675.84	676.57	622.77	603.85	858.69	677.39	639.95
0.3		689.93	690.58	630.29	607.90	856.31	693.47	648.65
0.4		713.81	713.05	633.62	608.20	863.23	719.43	669.92
0.5		735.78	734.53	638.91	611.35	868.42	742.96	688.07
0.0	SOD	2732.90	-	-	-	-	-	-
0.1		3638.15	3625.89	3529.50	3483.53	4071.10	3638.61	3536.92
0.2		3668.04	3669.67	3534.08	3480.55	4070.07	3671.13	3558.81
0.3		3695.58	3669.67	3557.75	3557.75	4062.57	3704.41	3573.48
0.4		3750.56	3748.53	3563.95	3507.88	4080.22	3764.71	3624.93
0.5		3797.44	3793.02	3579.07	3517.64	4091.18	3813.87	3664.38

more blurry and smoother which lower down the flat wall 3D models' roughness. Hence, the SSE and SOD errors are getting smaller as σ_2 increase since the scanning object is flat. A Gaussian blur effect typically generated by convolving an image with a kernel of Gaussian values. Therefore, the laser peaks are shifted during the convolution and such effect is similar as 3D smoothing, the higher strength is, the smoother and flatter the 3D wall model is. The flatter the 3D wall model is, the smaller the SSE, SOD error are.

3. Median filter has the worst result. It has the largest SSE and SOD error in overall experiment. The median filter considers each pixel in the image in turn and looks at its nearby neighbours to decide whether or not it is representative of its surroundings. Instead of simply replacing the pixel value with the mean of neighbouring pixel values, it replaces it with the median of those values. Its performance is not that much better than Gaussian blur for high levels of noise, whereas, for speckle noise and salt and pepper noise (impulsive noise), it is particularly effective. [2] Hence, median filter has poor denoise performance on our laser images corrupted by Gaussian noise because it replaces each pixel with median of its surrounding values.

4. Wiener filter has similar performance as Gaussian filter as $\sigma_2 = 0.3$. Calculation of the Wiener filter requires the assumption that the signal and noise processes are second-order stationary [3]. From this point, Wiener filter is MSE-optimal stationary linear filter. Whereas, SCS exploits the statistical properties of data to be denoised in a more realistic way than other filters. When our image data is rather sparsely distributed, i.e. supergaussian, like our cases, SCS has the ability to smooth the image without losing too much features.

6.1.2.3 Results and analysis: Cup

In this experiment, the scanning has similar set up as the flat wall. It was done in water and equipment was waterproof. The camera resolution is 720×576, the laser projection angle is about 60°. The linear stage moving step is $1mm$. From the scanner to the cup distance is about $15cm$. [83] camera calibration and the proposed step described in Chapter 5 were applied to estimate the scanner parameters (including the baseline D and laser projection angle α). Gaussian approximation peak detection [22] method was used. Fig.6.11 shows one of the scanning images when the laser swept the cup's surface. There are two large black letters: "V" and inverted "C" on the surface and the laser does not reflect in this area. Fig.6.12 shows part of the 3D model and the letter "V" can be clearly recognized. Hence, there are three measurements in this scanning indicated by letters A, B, and C. To simulate the real applications, different levels of artificial Gaussian noise were added to the image and then denoised by Gaussian filter,

---

[2] Access on 20th May 2013 http://en.wikipedia.org/wiki/Median_filter

[3] Access on 20th May 2013 http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/VELDHUIZEN/node15.html

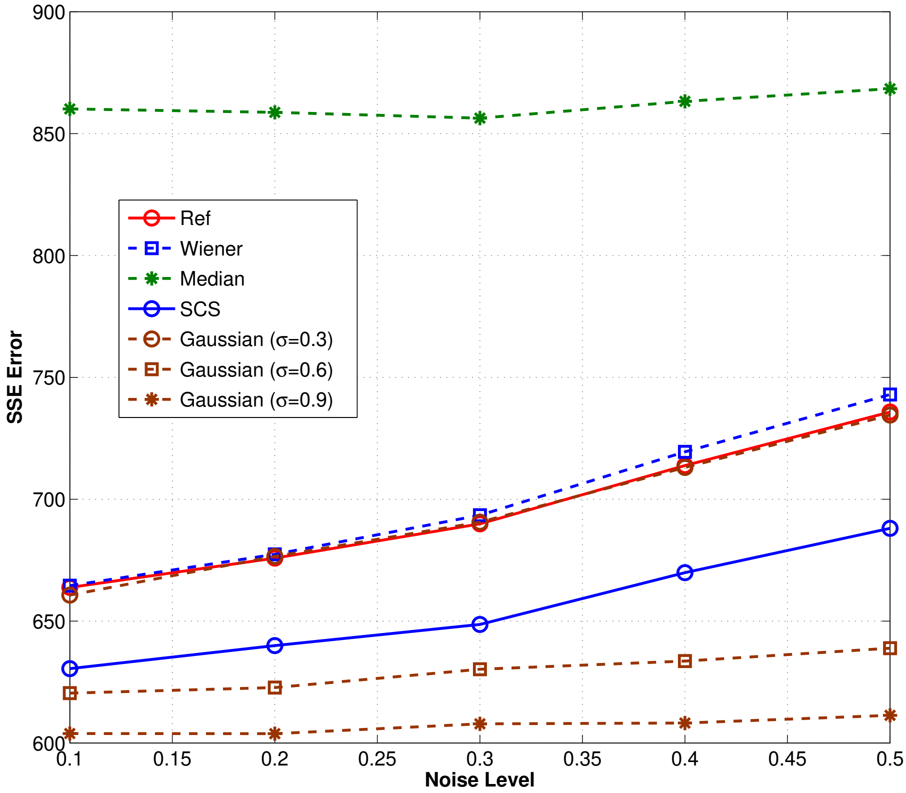


Figure 6.9: Orthogonal linear regression fitting error (SSE) on noisy 3D models

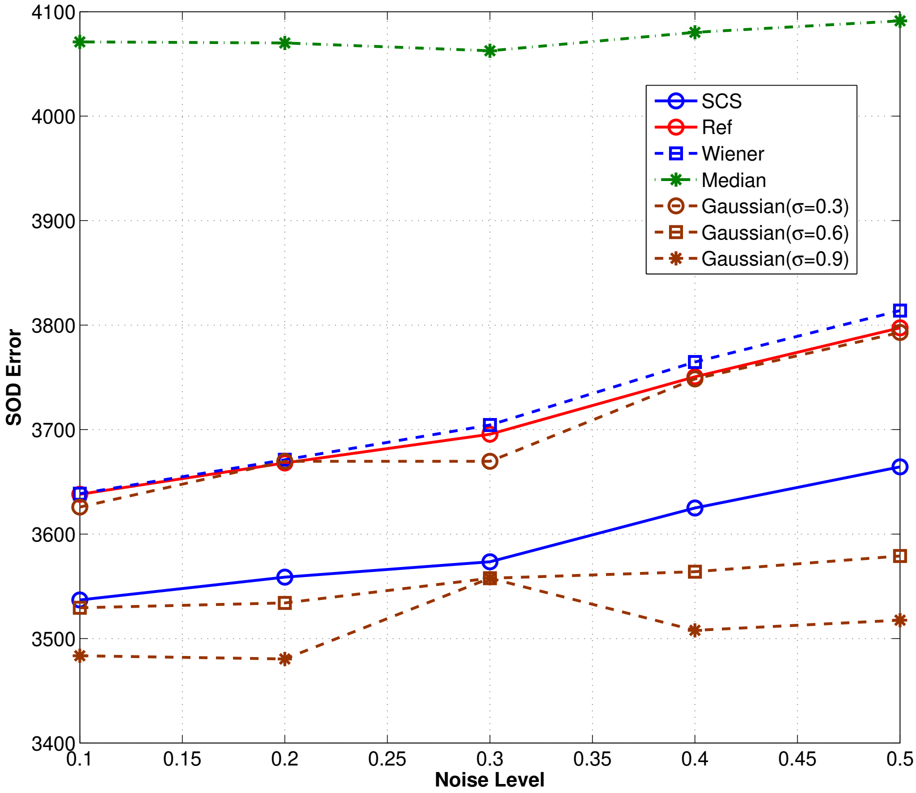


Figure 6.10: Orthogonal linear regression fitting error (SOD) on noisy 3D models

median filter, Wiener filter and SCS respectively. These measurements were taken from the models at different noise levels. The result is shown in Table 6.2 and Figures 6.13, 6.14 and 6.15.

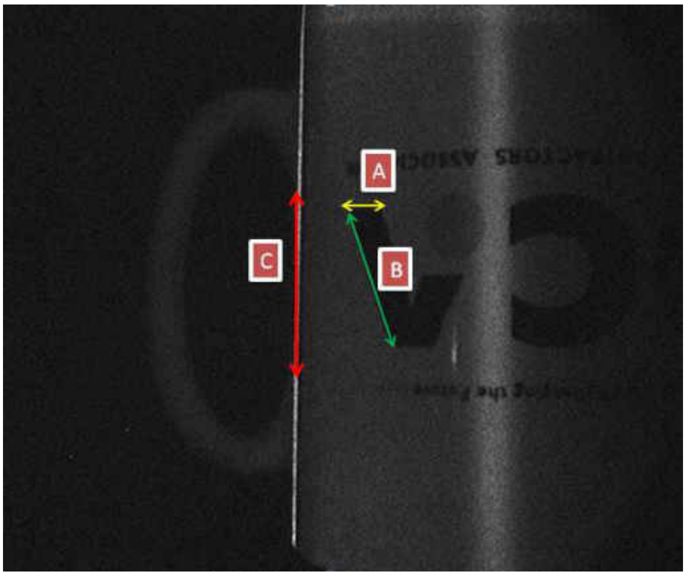


Figure 6.11: Scanning object - Cup and its three measurements: 'A', 'B', and 'C'

In Table.6.2, the first column is the selected measuring distances and σ_1 is the noise level, increases from 0.0 to 0.5. The third column is the real measurement. Column "Ref" means the measurements were taken from the 3D models reconstructed from the noisy images but without applying any denoising methods. The following columns show the measurements de-noised by Gaussian filter, Median filter, Wiener filter and SCS respectively. Since both of these measurements' variation are very small and close, hence, it is easier to compare from their relative error to real measurement. In Table.6.2, all denoise results were compared to real values in RE (Relative error) columns. It can be observed from this table that:

1. Measurement A remain stable at different noise levels in denoising methods. It is because A is almost parallel to the linear stage moving direction and its reading is very close to the stage moving steps.

2. Measurement B and C have much larger variation when noise added. For instance, B is $22.02mm$ without noise corruption but increases to $23.11mm$ when the noise level increase to 0.1. As the noise increases, the measurement B becomes slightly large. For measurement C, it has up to $1mm$ error as noise increases.

3. SCS has an average RE of -2.5% in measurement A, while the other methods have a maximum -3.78% RE. In measurement B, SCS has much smaller RE error even in high noise levels (from 0.3 to 0.5) compared to the others. While SCS

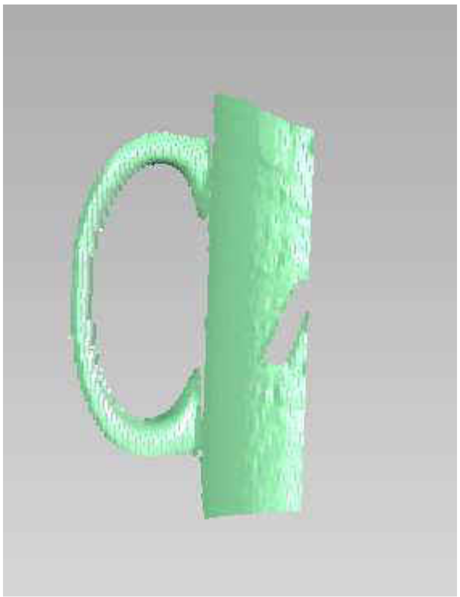


Figure 6.12: The cup's 3D model

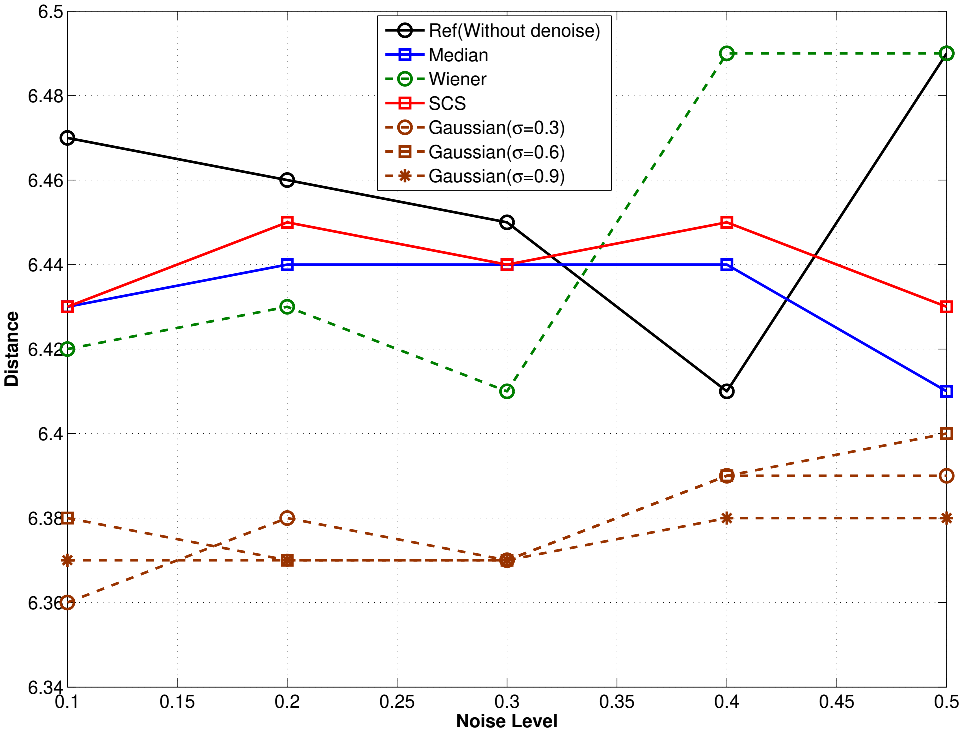


Figure 6.13: Measurement 'A' after denoise. Unit: mm

Table 6.2: Validation experiments result and error in noisy 3D cup model. The Sparse Code Shrinkage method was applied using the estimated orthogonalized ICA transform (8×8 windows), while the mask in Gaussian smoothing and neighborhood in median and wiener filtering increase from 3×3 to 5×5 as the noise level increase. σ_1: Noise levels, σ_2: Gaussian smooth levels, RE: Relative error, STD: standard derivation

	σ_1	Real	Ref	Gaussian						Median	RE	Wiener	RE	SCS	RE
				$\sigma_2 = 0.3$	RE	$\sigma_2 = 0.6$	RE	$\sigma_2 = 0.9$	RE						
A	0.0	6.61	6.33	-	-	-	-	-	-	-	-	-	-	-	-
	0.1	6.61	6.47	6.36	-3.78%	6.38	-3.48%	6.37	-3.63%	6.43	-2.72%	6.42	-2.87%	6.43	-2.72%
	0.2	6.61	6.46	6.38	-3.48%	6.37	-3.63%	6.37	-3.63%	6.44	-2.57%	6.43	-2.72%	6.45	-2.42%
	0.3	6.61	6.45	6.37	-3.63%	6.37	-3.63%	6.37	-3.63%	6.44	-2.57%	6.41	-3.03%	6.44	-2.57%
	0.4	6.61	6.41	6.39	-3.33%	6.39	-3.33%	6.38	-3.48%	6.44	-2.57%	6.49	-1.82%	6.45	-2.42%
	0.5	6.61	6.49	6.39	-3.33%	6.40	-3.18%	6.38	-3.48%	6.41	-3.03%	6.49	-1.82%	6.43	-2.72%
	STD	-	-	0.01	-	0.04	-	0.01	-	0.01	-	0.04	-	0.01	-
B	0.0	20.92	22.02	-	-	-	-	-	-	-	-	-	-	-	-
	0.1	20.92	23.11	23.85	14.01%	24.02	14.82%	24.03	14.87%	23.43	12.00%	23.11	10.47%	22.80	8.99%
	0.2	20.92	23.72	23.59	12.76%	24.18	15.58%	24.17	15.54%	23.56	12.62%	22.97	9.80%	22.96	9.75%
	0.3	20.92	23.57	24.03	14.87%	23.88	14.15%	24.03	14.87%	23.65	13.05%	23.39	11.81%	23.29	11.33%
	0.4	20.92	23.37	23.83	13.91%	23.85	14.01%	23.87	14.10%	23.55	12.57%	23.49	12.29%	23.55	12.57%
	0.5	20.92	23.41	23.63	12.95%	23.73	13.43%	24.03	14.87%	23.56	12.62%	23.86	14.05%	23.45	12.09%
	STD	-	-	0.18	-	0.17	-	0.11	-	0.08	-	0.35	-	0.32	-
C	0.0	34.85	33.94	-	-	-	-	-	-	-	-	-	-	-	-
	0.1	34.85	35.48	36.19	3.85%	36.33	4.25%	36.37	4.36%	35.69	2.41%	35.38	1.52%	35.29	1.26%
	0.2	34.85	35.73	36.20	3.87%	36.50	4.73%	36.55	4.88%	35.88	2.96%	35.54	1.98%	35.34	1.41%
	0.3	34.85	35.80	36.54	4.85%	36.15	3.73%	36.14	3.70%	35.69	2.41%	35.55	2.01%	35.39	1.55%
	0.4	34.85	35.54	36.37	4.36%	36.33	4.25%	36.52	4.79%	35.7	2.44%	35.37	1.49%	35.42	1.64%
	0.5	34.85	35.72	36.19	3.85%	36.56	4.91%	36.69	5.28%	35.69	2.41%	35.38	1.52%	35.37	1.49%
	STD	-	-	0.16	-	0.16	-	0.21	-	0.08	-	0.09	-	0.05	-

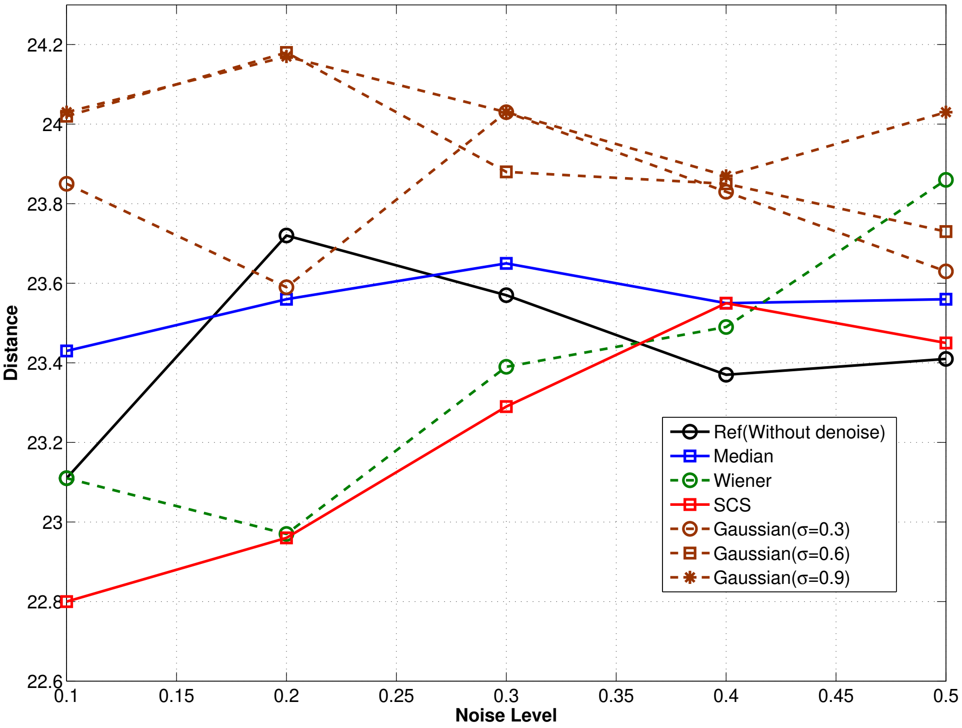


Figure 6.14: Measurement 'B' after denoise. Unit: mm

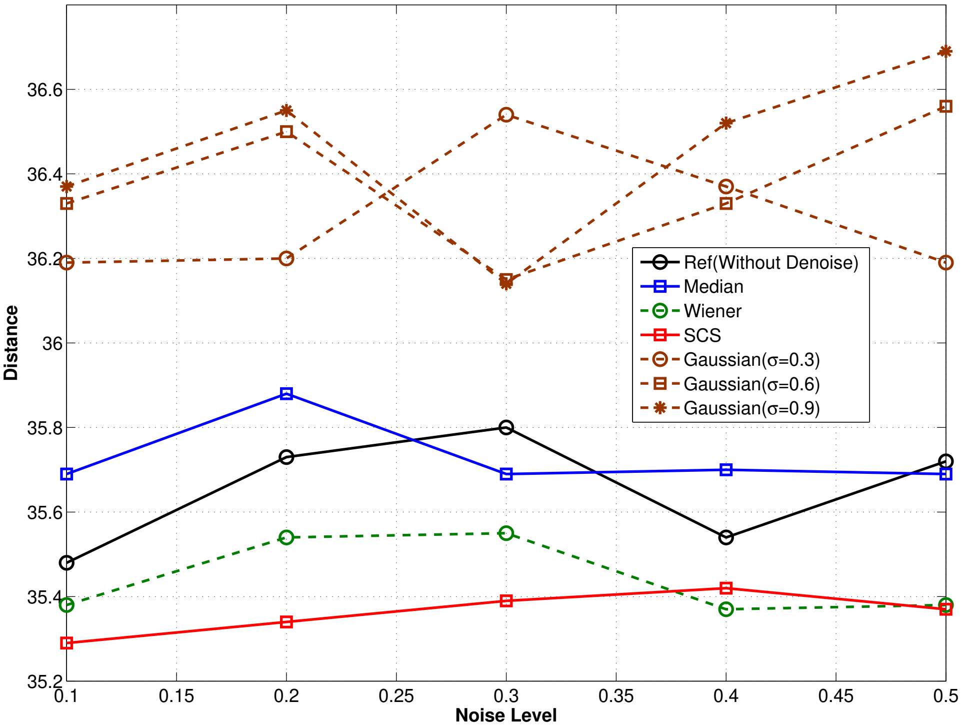


Figure 6.15: Measurement 'C' after denoise. Unit: mm

has a RE of 12.094% at the highest noise level, the Gaussian filter has a RE as high as 15.54%, increasing by almost 2%. As mentioned before, A is parallel to the linear stage moving direction and its measurement error mainly came from X coordinate which is along the stage travel direction. In other words, errors from Y, Z directions have much less impact compared to those on measurement B. In measurement C, SCS maintains the RE error below 1.7%, while Gaussian, Median and Wiener filters have an error up to 5.0%. Wiener filter has similar problem as mentioned in above section, its assumption (the signal and noise processes are second-order stationary) is not quite fitted in our cases. It has been discussed that median filter replaces each pixel with median of its surrounding values in image. Such process cause the laser peak shift or undetectable.

4. It is clear that Gaussian filter has worse performance than the previous experiment. It is due to the scanning object's shape, the Gaussian filter strength directly affects flat wall 3D model's shape and surface's smoothness. As the smooth levels(σ_2) increase, SOD and SSE error are lower. However, it is different in this experiment. As the smooth levels increase, the RE error is larger. When $\sigma_2 = 0.9$, it has the smallest SSE and SOD in the flat wall model but largest RE error in the cup model. As mentioned before, the Gaussian smooth actually blurs the image and reduces the image detail. Thus, it causes the laser peak shifting, which finally affect the 3D model's accuracy. The higher the smooth strength σ_2, the bigger the laser peak shift.

5. In the noise free dataset, the three measurements still have up to $1.1mm$ absolute error compared to real ones. The error may come from the system itself and the light refraction in the water.

6.1.3 Conclusion

In this section, a detailed comparative study of three noise removal methods was carried out using natural images and real 3D scanning images. Denoising is closely linked to pattern recognition. Optimal denoising involves interpreting the noisy signal, and based on that interpretation reconstruction of the original signal. Thus, SCS is suitable to identify the features of the signal, and retains those features which is likely to be the original signal like discarding those which probably are a result of noise. This is where sparse code is useful. Since the important features in most natural signal are sparse that sparse code will detect these features. For a random vector X (Image data, for example), which have been corrupted with additive Gaussian white noise, estimating a sparse code for it will give us a transform, which will concentrate much of the energy of the sample vectors into only a few components. If our transform is orthogonal, then the noise will still be Gaussian and white in the transform basis. Thus, we have effectively separated the signal from the noise, and by setting small components to zero, we will remove a large part of the noise but only a little of the signal [32]. The experimental result is consistent with above theory. When our image data is rather sparsely distributed, i.e. supergaussian, SCS has the ability to smooth the image without losing too much features, although some data is corrupted by synthetic noise.

Sparse Code Shrinkage exploits the statistical properties of the data to be denoised in a more realistic way than Wiener filter that is known to be optimal for Gaussian data and median filter, which does not deal with data statistics. SCS reduces noise without blurring edges or other sharp features as much as Gaussian filter, median filter, or Wiener filter. This is possible by the strongly non-linear nature of the shrinkage operator that takes advantage of the inherent statistical structure of natural images.

Gaussian filter blurs the image but reduces the image detail. It smoothes the 2D image and make the 3D surface much smoother. SSE and SOD errors are related to the flatness of 3D model, hence, it has the best result in flat wall model. However, it has the worst performance in the cup model since such smoothing actually shifts the laser peak position.

Median filter has the largest SSE and SOD error in the overall experiment. It replaces the pixel value with the median of neighbouring pixel values, it should have much better performance for speckle noise and salt and pepper noise (impulsive noise) but not for Gaussian noise. Hence, it performs not as well as Wiener filter and SCS.

In this experiment, the most important image feature is laser peak. The noise did have heavy impact on image quality and final 3D reconstruction accuracy. The noise distorted the laser peak location and effected the estimation of scanner's parameters (including the baseline D and laser projection angle α). In our experiment, the error is up to $3.0mm$ in highest noise level. SCS denoise results provide better surface smoothness and accuracy of 3D reconstruction in the overall comparison. However, the SCS has disadvantage at the computational time. It takes about 25 seconds to denoise an image with 720×576 resolution (orthogonalized ICA transform 8×8 windows) on an Intel Q6600 (2.4GHz) based PC. Since most of our applications do not require real time processing and display, this SCS method is still able to apply in those cases. It is possible to speed up the process by changing the entire program using parallel computing. In addition, the SCS may be able to apply to colour images. Research is underway and results will be reported in the future.

6.2 Corner detection

In this section, the results of all the corner detectors outlined in section .3.2 are presented. Generally, an ideal corner detector should satisfy a number of criteria:

- As many as possible ”True” corners detected;
- As few as possible ”False” corners detected;
- Corner points should be well localized;
- Repeatable: the corners can be detected from the images subject to different perspective projection transformations;
- Robust with respect to noise: the detected corners will not be heavily affected by the imaging noise;
- Computationally efficient.

Our 3D laser scanner is a combined unit with different components and they are connected and related to each other. Corner detection is one of the most important components and it is the preprocessing step for camera calibration and image correction, which affects final 3D reconstruction accuracy. Paper [11] proposed a new corner detector and compared with Harris and SUSAN methods. The proposed CSS method in [11] detects the maximum number of true corners with the fewest false and has the smallest localization error. However, it didn’t validate its performance on the calibration pattern such as the widely used chessboard. Hence, it is very important to evaluate in this section:

1. These corner detectors detection accuracy of camera calibration target, such as chessboard pattern.
2. Such detection result effect on the final 3D reconstruction based on real scanning data.

6.2.1 Experimental setup

Some of the corner detectors can be downloaded from internet such as: SUSAN [4] and CSS[5]. However, both of them only provide pixel level result. Hence, OpenCV’s corner sub-pixel refining method from [6] based on such detected corners was applied to all the detectors for fair comparison.

The test was done in underwater. The Bowtech LCC-600 camera with a resolution of 720×576 was used. The calibration target is a printed and laminated chessboard, sticking on a flat black board. It contains $8 \times 6 = 48$ corners, every corner has $20mm$ distance

---

[4] Access on 20th Feb 2012 http://users.fmrib.ox.ac.uk/ steve/susan/

[5] Access on 20th Feb 2012 http://www.mathworks.com/matlabcentral/fileexchange/7652

[6] Access on 20th Feb 2012 http://opencv.itseez.com/modules/imgproc/doc/feature_detection.html

to each other. 10 images with different orientations of this flat board were captured for Zhang ZhengYou [83] calibration method. To avoid the effect from shadow or the refraction from water, all the rest area in the images has been painted black manually except the chessboard area as shown in Fig.6.16. The calibration parameters can be used for two purposes: (1) Image correction, (2) 3D reconstruction. Therefore, the scanner's parameters estimation procedure proposed in section.5.2 can be applied in this case to verify the corner detection accuracy. The system was placed and set up as the scanner parameters estimation described in section.5.2. The vertical flat board was placed in front of the scanner. The scanner travelled 19 steps in $1mm$ interval, then 19 laser stripe images were captured. To have a fair comparison, Gaussian approximation laser peak detector was used in all cases. Each detector's parameter was adjusted individually to have the best results. There are two main aspects to be verified in this experiment: (1) the laser images used in the scanner parameters estimation process should have "Straight" laser after the correction by the calibration parameters, and (2) the 3D result of those laser stripes should be "Straight" in 3D space, too. Otherwise, non-straightness shows that the error still exists in the undistorted images and 3D reconstruct results and thus, such image calibration and correction, and 3D reconstruction methods are not very accurate.

6.2.2 Results and analysis

The corner detection result are shown in figures from 6.17 to 6.20. The camera calibration result based on these detected corners are shown on Table.6.3. It includes calibration parameters: f_x, f_y, C_x, C_y, re-projection error 'RPE' and pixel error X_e, Y_e (They are the pixel errors between detected corners and their corresponding 3D points re-projected in the image plane in XY direction). As the table shows, Harris and Noble detectors have very close calibration results with very low average pixel and re-projection errors. The pixel error and 'RPE' from SUSAN and CSS are almost $3 \sim 4$ times larger than Harris and Noble.

To have clearer picture about these four corner detectors' performance on our chessboard images, Figures 6.23 and 6.24 show the individual detection results of Picture 0. As we can see, most of the corners detected by Noble(blue cross) are located inside the chessboard area with very few faulty in the image. Harris detected similar amount of true corners as Noble's but with some faulty corners which outside the chessboard area. Since the chessboard corners represent variation in the gradient in the image, Harris and Noble detector take the advantage by measuring the corner strength for each pixel in XY directions. Although both Harris and Noble corner detectors based on the similar principle but Noble's cornerness measure H (See Chapter.3.2.6) is apparent more effective. It detected the maximum number of true chessboard corners with very few or even none faulty.

SUSAN and CSS results are very poor compared to Harris and Noble detectors. Clearly, CSS and SUSAN (Red square and yellow round) detect the single chessboard corner as two individual corners which is significant lower down the calibration accuracy although these paired corners are very close to each other. In addition, it is hard to determine which corner has more precise location than the other on these paired corners does. Instead of using the image's pixel gradient for corner detector, Susan operator places a circular mask over the pixel to be tested (the nucleus) (See Chapter.3.2.2.3) which may detect faulty corners or treat the single chessboard corner to be two separate ones. CSS method also has the same problem because of its early step: Canny edge detector (See Chapter.3.2.2.4). No matter how we changed Canny edge detector's parameters using in CSS, it always treated the chessboard corner as circular instead of cross. Fig.6.21 and 6.22 show two of the example detection results in Picture 0 with canny edge. Apparently, it is impossible to convert these irregular circles between the chessboard grids into individual corners. It might be caused by the blurry grid connection parts in the chessboard images. These incorrect corners' locations have a heavy effect on the calibration result. Susan re-projection error RPE and average pixel error X_e, Y_e in Table.6.3 increase over 234.70% than Noble and Harris result. CSS calibration has even larger error, it increase over 367.91% than Harris and Noble's.

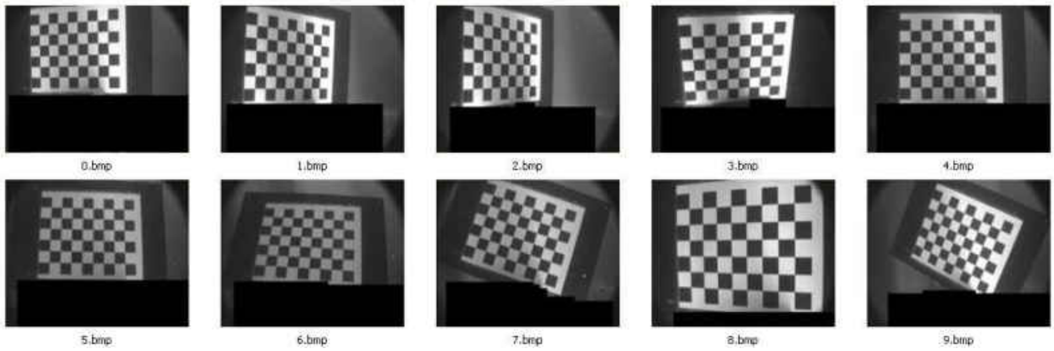


Figure 6.16: Chessboard images for corner detection

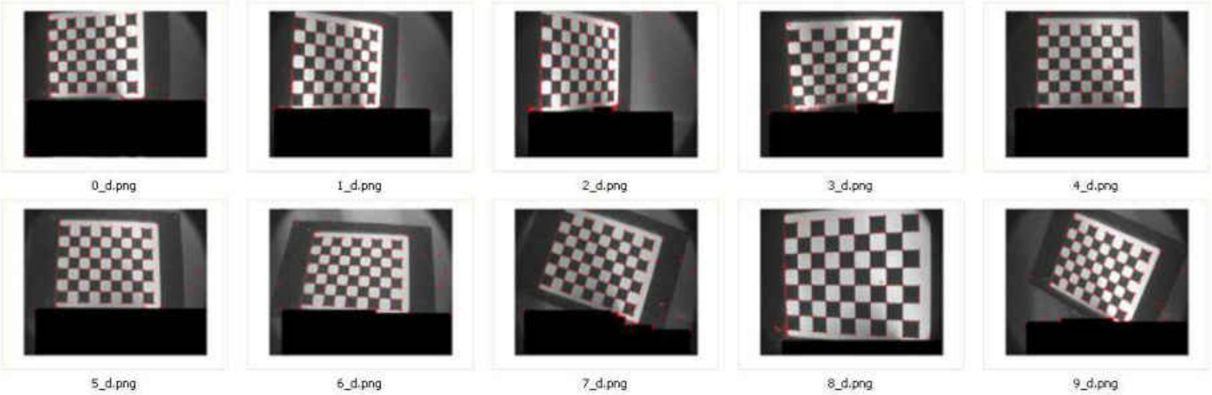


Figure 6.17: Harris Corner detection result on the chessboard images. The image size is 720×576, total $8 \times 6 = 48$ corners, every corner has $20mm$ distance to each other. Gaussian smoothing with 3×3 masks and the subpixel corner location refinement using 11×11 searching windows.

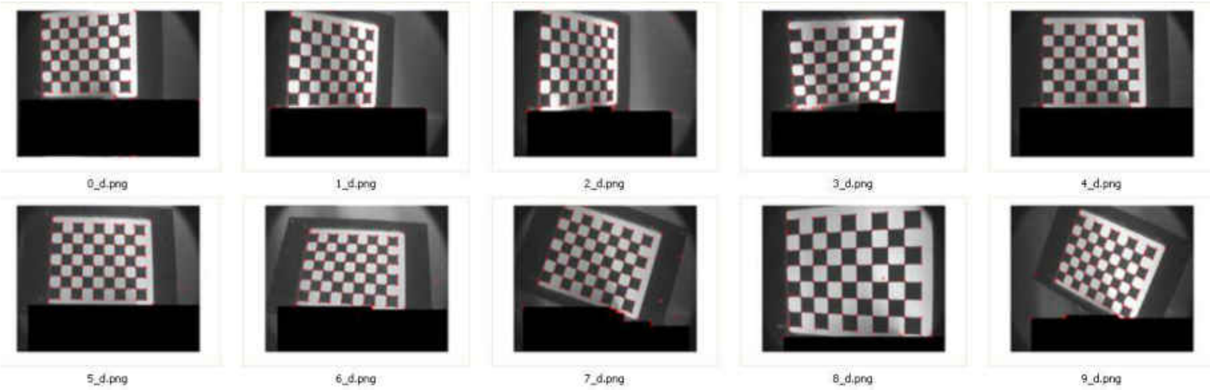


Figure 6.18: Noble Corner detection result on the chessboard images. The image size is 720×576, total $8 \times 6 = 48$ corners, every corner has $20mm$ distance to each other. Gaussian smoothing with 3×3 masks and the subpixel corner location refinement using 11×11 searching windows.

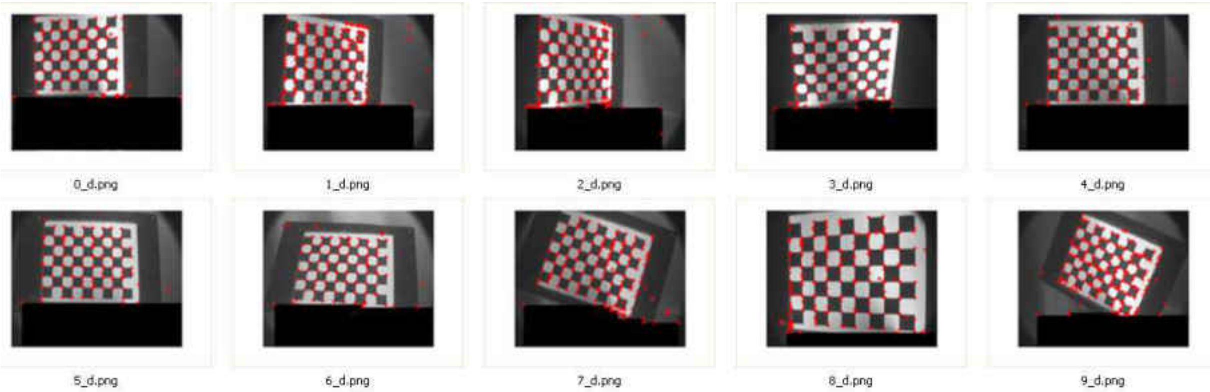


Figure 6.19: SUSAN Corner detection result on the chessboard images. The image size is 720×576, total $8 \times 6 = 48$ corners, every corner has $20mm$ distance to each other. Brightness threshold is 20, distance threshold is 4, 3×3 masks using in the image smoothing

Table 6.3: Camera calibration result with different corner detectors. Image size is 720×576, total $8 \times 6 = 48$ corners, every corner has $20mm$ distance to each other. Gaussian smoothing with 3×3 masks and the subpixel corner location refinement using 11×11 searching windows in Harris and Noble detectors. Susan's brightness threshold is 20, distance threshold is 4, 3×3 masks using in the image smoothing. CSS sensitivity thresholds $L = 0.35$, $H = 0.0$ for the Canny method, minimum ratio $C = 1.5$ of major axis to minor axis of an ellipse; maximum obtuse angle $T = 162.0°$ that a corner when it is detected as a true corner. RPE: Re-projection error

	f_x	f_y	C_x	C_y	RPE	X_e	Y_e
Harris	1003.174	1067.540	341.634	295.416	0.268	0.205	0.172
Noble	1003.173	1067.539	341.634	295.415	0.268	0.205	0.172
SUSAN	1011.253	1075.131	355.623	297.027	0.897	0.522	0.453
CSS	1012.089	1074.732	344.844	290.587	0.986	0.698	0.696

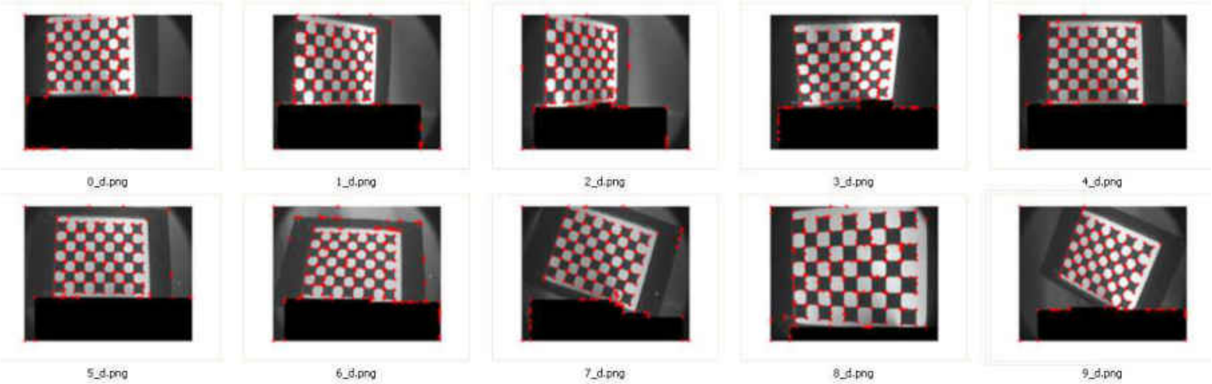


Figure 6.20: CSS Corner detection result on the chessboard images. The image size is 720×576, total $8 \times 6 = 48$ corners, every corner has $20mm$ distance to each other. Main parameters for CSS: sensitivity thresholds $L = 0.35$, $H = 0.0$ for the Canny method, minimum ratio $C = 1.5$ of major axis to minor axis of an ellipse; maximum obtuse angle $T = 162.0°$ that a corner when it is detected as a true corner

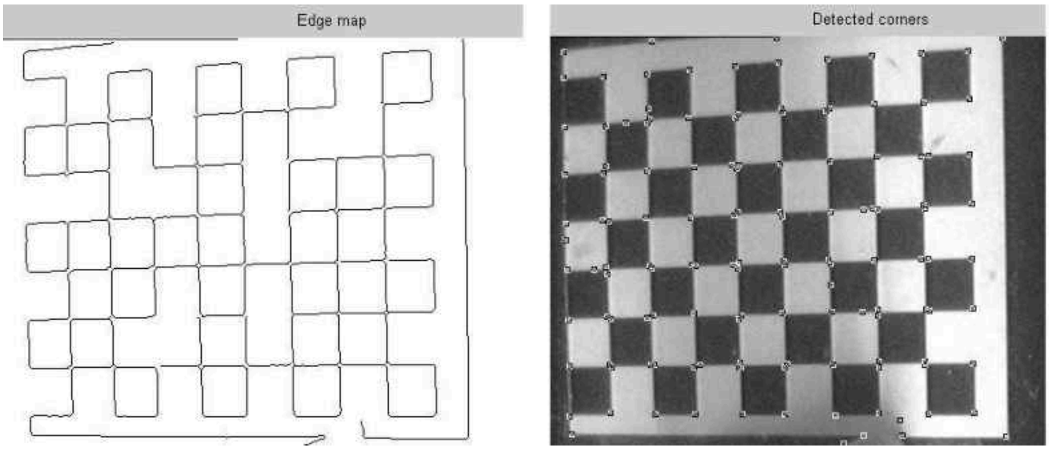


Figure 6.21: CSS corner detection result (a) on Picture 0. Left: detected edge by Canny. Right: CSS corners. Sensitivity thresholds $L = 0.35$, $H = 0.0$ for the Canny method. Main parameters for CSS: minimum ratio $C = 1.5$ of major axis to minor axis of an ellipse; maximum obtuse angle $T = 162.0°$ that a corner when it is detected as a true corner

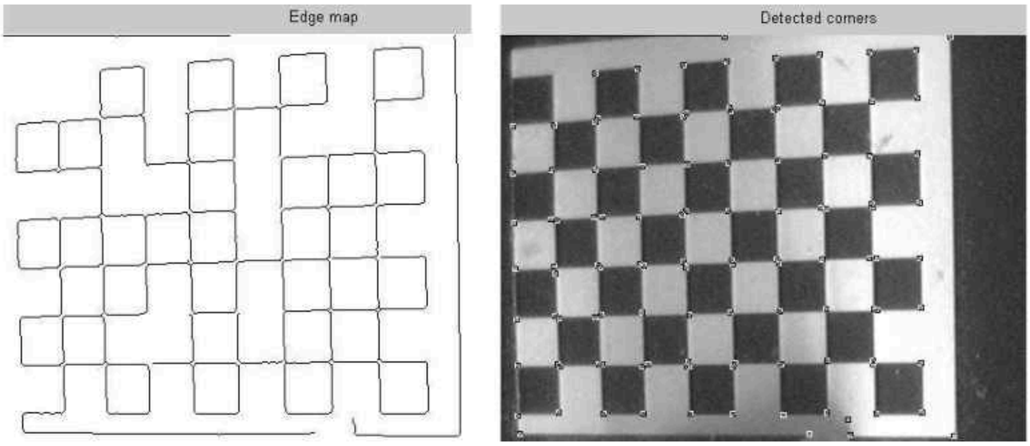


Figure 6.22: CSS corner detection result (b) on Picture 0. Left: detected edge by Canny. Right: CSS corners. Sensitivity thresholds $L = 0.65$, $H = 0.1$ for the Canny method. Main parameters for CSS: minimum ratio $C = 1.7$ of major axis to minor axis of an ellipse; maximum obtuse angle $T = 156.54°$ that a corner when it is detected as a true corner,

The calibration result is used for image undistortion and then applied in scanner parameters estimation. The laser itself was carefully adjusted to be straight and parallel to camera Y axis physically. Hence, these laser stripes should be straight in the images. It may be bent in the image because of the lens distortion but the such distortion is able to correct by the calibration parameters. The captured laser stripe images are shown in Fig.6.25. These laser stripes were extracted by Gaussian approximation detector and then the least square fitting was applied to estimate the laser stripe straightness for each image. Tables 6.4 and 6.5 show the laser stripes' straightness in 2D and 3D space respectively. In the first table, R is the sum of squares of the residuals from the best-fit line and α is the approximation laser stripe angle in the image plane. The smaller R the better the least square fitting is. The closer the α to 90° is, the better the estimate as the laser itself was placed vertically to the camera Y axis.

Harris and Noble have very close results as they based on similar corner detection principle. Their calibration result are very close to each other even Harris has much higher faulty corner detection rate. Therefor, their result of straightness measurement of the laser stripe by image correction are very close, the difference between their results is so tiny even can be ignored. Their corrected laser stripes only have 0.1 degree deviation and fitting error is below 5.4 in average, which is the smallest. SUSAN has the worst result in all cases. It has the biggest linear fitting error and the largest deviation of α. Its average fitting error is 12.86% lager than Harris and Noble's. It can be seen that such out of true calibration result by SUSAN and CSS can not undistort the image accurately. CSS provided slightly smaller maximum and minimum linear fitting error, however, its α deviation and average R are still bigger than either Harris or Noble. This means that the laser stripe after the un-distortion with CSS is unstable and tilting at bigger angle.

Such laser stripes straightness in 3D space can be another main aspect to verify the

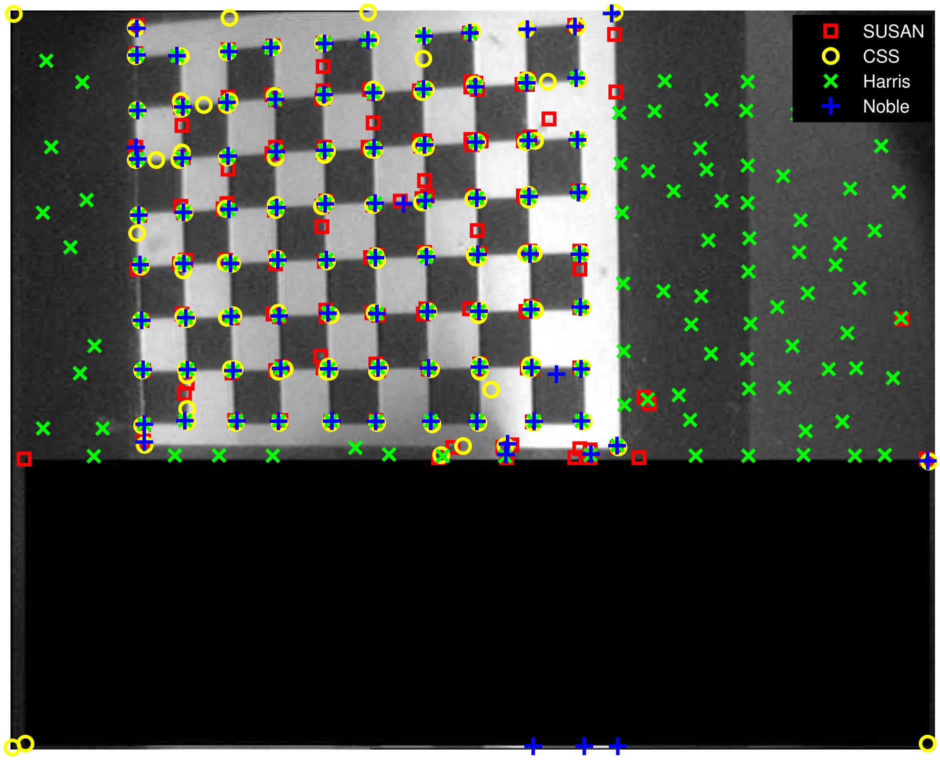


Figure 6.23: Picture_0 corner detection result by different methods. Image size is 720 × 576, total 8 × 6 = 48 corners, every corner has $20mm$ distance to each other. Gaussian smoothing with 3 × 3 masks and the subpixel corner location refinement using 11 × 11 searching windows in Harris and Noble detectors. Susan's brightness threshold is 20, distance threshold is 4, 3 × 3 masks using in the image smoothing. CSS sensitivity thresholds $L = 0.35$, $H = 0.0$ for the Canny method, minimum ratio $C = 1.5$ of major axis to minor axis of an ellipse; maximum obtuse angle $T = 162.0°$ that a corner when it is detected as a true corner.

Table 6.4: Laser stripes' straightness in 2D images. R: fitting error, Unit:pixel. α: the laser stripe angle, STD: standard deviation

	Harris		Noble		SUSAN		CSS	
	α	R	α	R	α	R	α	R
Avg	89.903	5.372	89.903	5.372	89.893	6.063	89.895	5.385
Max	89.915	6.081	89.915	6.081	89.905	6.812	89.905	6.053
Min	89.882	4.926	89.882	4.926	89.875	5.599	89.879	4.908
STD	0.0099	0.3000	0.0099	0.2999	0.0082	0.3355	0.0074	0.3070

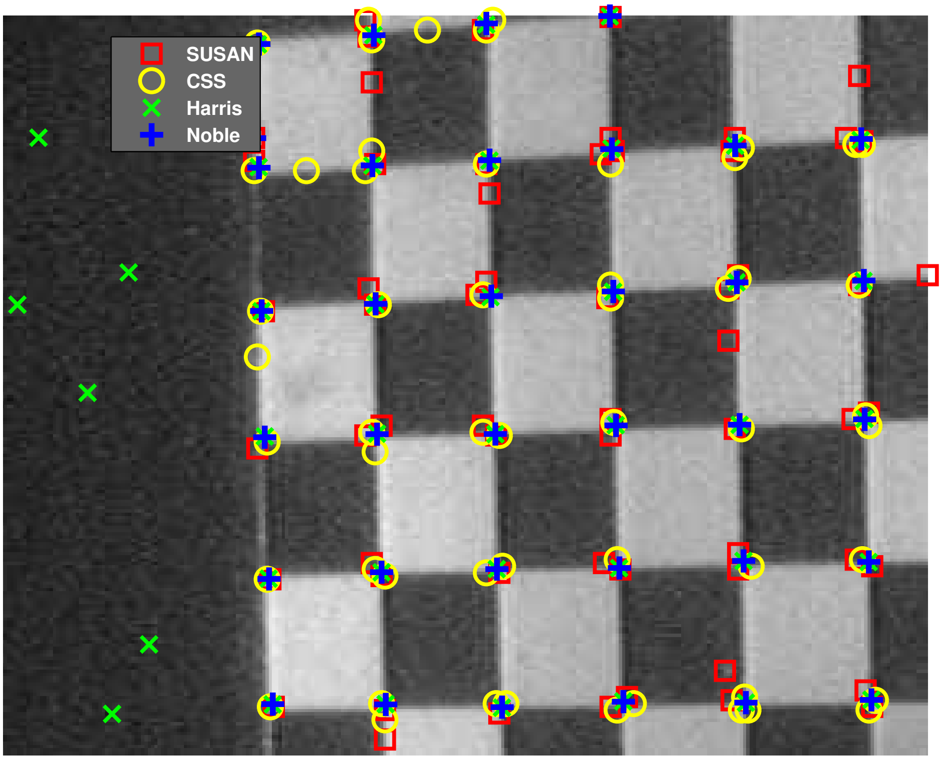


Figure 6.24: Picture_0 corner detection zoomed in result. Picture_0 corner detection result by different methods. Image size is 720 × 576, total 8 × 6 = 48 corners, every corner has 20*mm* distance to each other. Gaussian smoothing with 3 × 3 masks and the subpixel corner location refinement using 11 × 11 searching windows in Harris and Noble detectors. Susan's brightness threshold is 20, distance threshold is 4, 3×3 masks using in the image smoothing. CSS sensitivity thresholds $L = 0.35$, $H = 0.0$ for the Canny method, minimum ratio $C = 1.5$ of major axis to minor axis of an ellipse; maximum obtuse angle $T = 162.0°$ that a corner when it is detected as a true corner.

Table 6.5: Laser stripes' straightness in 3D space. R: fitting error, Unit:mm

	R		
	Avg.	Max	Min
Harris	6.82183E-03	7.12712E-03	6.55633E-03
Noble	6.80162E-03	7.07617E-03	6.44695E-03
SUSAN	7.05875E-03	7.43426E-03	6.77432E-03
CSS	6.92812E-03	7.28983E-03	6.56996E-03

Table 6.6: Absolute error of laser peak 3D reconstruction based on different corner detectors' result. Ref: actual measurement, β : absolute error between the estimated value and actual measurement, Unit:mm; STD: standard deviation

Ref.	Harris	β	Noble	β	SUSAN	β	CSS	β
158.000	158.404	0.404	158.404	0.404	160.520	2.520	159.561	1.561
159.000	159.358	0.358	159.358	0.358	161.477	2.477	160.517	1.517
160.000	160.375	0.375	160.375	0.375	162.496	2.496	160.517	0.517
161.000	161.372	0.372	161.372	0.372	163.496	2.496	161.774	0.774
162.000	162.391	0.391	162.392	0.392	164.517	2.517	163.554	1.554
163.000	163.350	0.350	163.350	0.350	165.475	2.475	164.514	1.514
164.000	164.372	0.372	164.372	0.372	166.496	2.496	165.536	1.536
165.000	165.338	0.338	165.338	0.338	167.465	2.465	166.503	1.503
166.000	166.348	0.348	166.348	0.348	168.475	2.475	167.513	1.513
167.000	167.340	0.340	167.340	0.340	169.468	2.468	168.506	1.506
168.000	168.374	0.374	168.374	0.374	170.503	2.503	169.540	1.540
169.000	169.335	0.335	169.335	0.335	171.465	2.465	170.502	1.502
170.000	170.369	0.369	170.369	0.369	172.500	2.500	171.536	1.536
171.000	171.334	0.334	171.334	0.334	173.467	2.467	172.501	1.501
172.000	172.350	0.350	172.350	0.350	174.481	2.481	173.518	1.518
173.000	173.315	0.315	173.315	0.315	175.447	2.447	174.482	1.482
174.000	174.366	0.366	174.366	0.366	176.496	2.496	175.535	1.535
175.000	175.325	0.325	175.325	0.325	177.456	2.456	176.493	1.493
176.000	176.360	0.360	176.360	0.360	178.492	2.492	177.528	1.528
Avg.	-	0.357	-	0.357	-	2.484	-	1.428
STD.	-	0.023	-	0.023	-	0.020	-	0.280

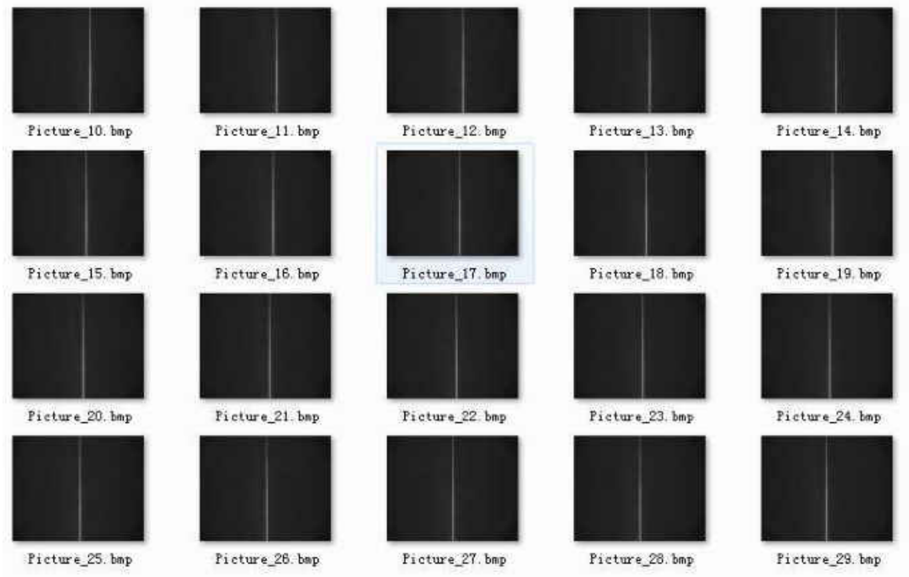


Figure 6.25: The laser stripes images used for image correction

corner detection accuracy. Hence, All the laser stripes are reconstructed into 3D space by the calibration and scanner parameters. The laser stripes' linearity in 3D space and their average fitting results are shown in the Table.6.5. R is the sum of orthogonal distances by the least square fitting. Noble provides the smallest sum of orthogonal distances in average. Its maximum and minimum values are smaller than the others. To have a better idea about the corner detection effect on the final 3D reconstruction accuracy, these laser stripes Z distances are shown in Table.6.6. Since the distance between the camera to the vertical board is measurable then it can be used for verifying the accuracy of 3D reconstruction. The initial distance 'Ref' is set to $158.00mm$ and the scanner travelled 18 steps in $1mm$ interval. Hence, the final distance is $176.00mm$. Table.6.6 shows the laser stripes absolute error between the estimated from reconstruction and actual value in 3D space. As mentioned before, the calibration result also applied in our scanner parameters estimation process, therefore, the error from inaccurate calibration result has been amplified in the laser stripe straightness and distance measurement in 3D. Clearly, Harris and Noble have the best result. They limit the error below $0.4mm$, the average absolute distances are only $0.357mm$, which is 85.63% smaller than SUSAN ($2.484mm$) and 75.00% than CSS ($1.428mm$).

6.2.3 Conclusion

In this section, we carried out a detailed comparative study of four corner detectors using real chessboard images. In summary, Harris operator suffers from poor localization and is computationally expensive but Noble corner detector has improved it. Considering that the variable k in Harris measure was too weak and difficult to set up, Noble proposed the new cornerness measure with great improvement. Noble has achieved a far much better detection rate (true/fault corners) in our chessboard images. It detected the maximum number of true chessboard corners with very few or even none false ones in all cases. Even though their calibration results are almost the same, Harris detector does take much more time to eliminate the corners which are belonged to the chessboard patter.

Harris and Noble corner detection algorithms are superior to SUSAN algorithm. The performance of SUSAN corner detector mainly depends on the similar comparison function 3.2.7 which is not immune to certain factors impacting imaging, such as strong luminance fluctuation, and noises. However, Harris and Noble corner detector is better because it draws a Gaussian smoothing function into the detector, which makes great contribution to improving the stability and reducing the impact of noise and that is why SUSAN finds many false corners in the experiment [10].

CSS method is highly dependent on the edge detector used. Canny edge detector is the default edge detector and used in our experiment. Because of that, CSS treats the individual corner as two connected corners in our cases. Both of CSS and SUSAN need to filter such corners manually to be able to use in camera calibration. It is time consuming and impractical. In additional, it is hard to determine which one has more precise corner location in those connected corners pairs and they may contain some missing detected corners. Hence, Noble corner detection is the most suitable method in the chessboard detection and chosen in our scanning system.

6.3 Laser Peak Detection

This section compared six algorithms for determining the laser peak position of an image line or stripe at sub-pixel level accuracy. A good laser peak detector should satisfy a number of criteria:

1. Laser peak should be well localized at sub-pixel level;
2. Robust with respect to noise;
3. Robust with respect to laser saturation;
4. Computationally efficient;

Since our system and applications are not requiring real time peak detection that criteria $1 \sim 3$ are more important in our cases. Depending on the laser profile, object surface reflectivity and camera resolution, the laser stripe can have variable widths. New technology brings larger camera CCD size and higher image resolution. For example, the latest camera Canon 5D mark III has 21 mega-pixels and its maximum image size is $5,616 \times 3,744$. The laser stripe may have a width of over 50 pixels in the image captured by it. The camera used in our laser scanning system has resolution of 720×576 in water and 1440×1024 in air, which are much lower than the camera with ultra-high image resolution. Such cameras were chosen because of the budget and underwater scanning requirements such as weight, object size, capture speed and signal transmission. Due to the laser projection angle, strength, object surface reflection rate and camera setting, the laser stripe thickness/saturation may vary from $3 \sim 10$ pixels in our underwater laser scanning system. Hence, how these detectors perform in highly saturated laser stripe image has a huge impact on the final 3D reconstruction accuracy. Therefore, this experiment focuses on their performance in the saturated laser stripe in artificial and real images. Their performance in final 3D result will be evaluated based on the artificial images and controllable camera parameters.

6.3.1 Real Laser Stripe Image

Figure 6.26 and 3.13 show one of the heavy noisy images and its example row data in our experiment respectively. Clearly, there is some unwanted stripe pattern on the images, which mainly causes by the low cost USB video capture stick and the overpower laser stripe. Table.6.7 shows its saturated pixel intensity of from rows (image height) $184 \sim 210$. As we can see, the laser stripe saturation area is from $412 \sim 418$(image width) and crossing pixels. The pixels intensity in these areas are 255. It can be inferred from the figures and intensity table that the approximate laser peak locates at 415(image width). It is a big challenge for the peak detectors to find out the accurate peak position as the saturated part crossing over five pixels. Table.6.8 shows this laser example row data's peak detection result by six detectors. Fig.6.27 shows the plotted and enlarged detected results. Clearly, when the laser stripe width is over five pixels, only FIR detected peak precisely. All the rest methods have more than three pixels error. It is because GA, CM, LIP, PE and BR only consider the maximum light intensity and its nearby three to five pixels to locate the peak at sub-pixel level. They are applicable in laser stripe without heavy saturation and the laser energy pattern of such a stripe corresponds to a Gaussian profile. It makes sense to detect the point/peak of maximum light intensity in such case. However, in the cases like Fig.6.26, the maximum light intensity is 255 and cross over five pixels, the laser energy pattern is no longer corresponds to Gaussian profile. Therefore, they detected faulty laser peaks which are shifted to left.

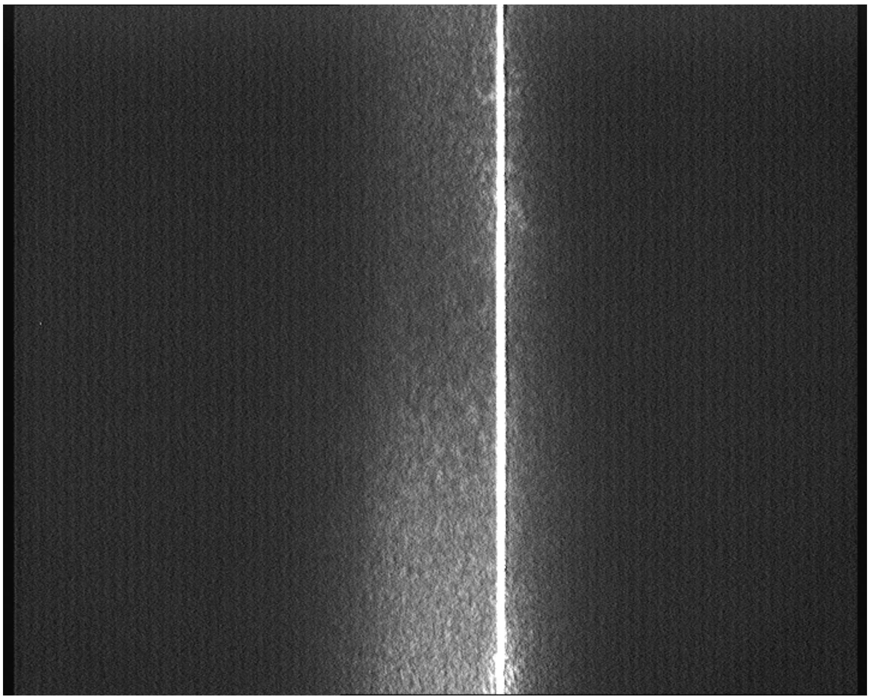


Figure 6.26: Saturated laser stripe image

Table 6.7: Saturated laser stripe image pixel intensity table. Image size: 720 × 576 H: image height, W: image width, Unit: pixel

W H	410	411	412	413	414	415	416	417	418	419	420
186	94.00	107.00	252.00	255.00	255.00	255.00	255.00	255.00	255.00	159.00	43.00
187	81.00	111.00	211.00	255.00	255.00	255.00	255.00	255.00	255.00	182.00	18.00
188	144.00	161.00	187.00	255.00	255.00	255.00	255.00	255.00	255.00	147.00	72.00
189	140.00	158.00	203.00	255.00	255.00	255.00	255.00	255.00	255.00	95.00	67.00
190	114.00	172.00	255.00	255.00	255.00	255.00	255.00	255.00	255.00	178.00	71.00
191	137.00	151.00	255.00	255.00	255.00	255.00	255.00	255.00	255.00	180.00	105.00
192	110.00	167.00	255.00	255.00	255.00	255.00	255.00	255.00	255.00	172.00	87.00
193	117.00	131.00	255.00	255.00	255.00	255.00	255.00	255.00	255.00	115.00	74.00
194	115.00	151.00	255.00	255.00	255.00	255.00	255.00	255.00	255.00	158.00	90.00
195	98.00	136.00	255.00	255.00	255.00	255.00	255.00	255.00	255.00	186.00	69.00
196	152.00	143.00	255.00	255.00	255.00	255.00	255.00	255.00	255.00	167.00	88.00
197	97.00	185.00	255.00	255.00	255.00	255.00	255.00	255.00	255.00	181.00	72.00
198	122.00	164.00	255.00	255.00	255.00	255.00	255.00	255.00	255.00	231.00	77.00
199	135.00	142.00	240.00	255.00	255.00	255.00	255.00	255.00	255.00	150.00	117.00
200	131.00	169.00	222.00	255.00	255.00	255.00	255.00	255.00	255.00	172.00	109.00
201	124.00	181.00	255.00	255.00	255.00	255.00	255.00	255.00	255.00	188.00	84.00
202	129.00	153.00	255.00	255.00	255.00	255.00	255.00	255.00	255.00	138.00	52.00
203	117.00	186.00	255.00	255.00	255.00	255.00	255.00	255.00	255.00	197.00	88.00
204	119.00	175.00	255.00	255.00	255.00	255.00	255.00	255.00	255.00	178.00	104.00
205	150.00	174.00	255.00	255.00	255.00	255.00	255.00	255.00	255.00	160.00	100.00
206	147.00	204.00	255.00	255.00	255.00	255.00	255.00	255.00	255.00	230.00	112.00
207	169.00	173.00	255.00	255.00	255.00	255.00	255.00	255.00	255.00	166.00	125.00
208	176.00	189.00	255.00	255.00	255.00	255.00	255.00	255.00	255.00	151.00	93.00
209	153.00	187.00	255.00	255.00	255.00	255.00	255.00	255.00	255.00	143.00	104.00
210	140.00	157.00	255.00	255.00	255.00	255.00	255.00	255.00	255.00	221.00	100.00

Table 6.8: Saturated laser stripe image peak detection result Image size: 720 × 576. W: image width, Unit: pixel

W	BR	CM	FIR	GA	LIP	PEM
186	413.306	412.061	415.472	412.500	412.500	412.500
187	413.723	412.098	415.499	412.500	412.500	412.500
188	413.536	412.073	415.499	412.500	412.500	412.500
189	412.589	411.122	415.498	411.500	411.500	411.500
190	412.881	411.157	415.018	411.500	411.500	411.500
191	412.607	411.130	415.080	411.500	411.500	411.500
192	412.899	411.193	415.014	411.500	411.500	411.500
193	412.743	411.157	414.969	411.500	411.500	411.500
194	412.758	411.184	415.017	411.500	411.500	411.500
195	413.087	411.172	415.132	411.500	411.500	411.500
196	412.443	411.101	415.059	411.500	411.500	411.500
197	412.684	411.135	414.986	411.500	411.500	411.500
198	413.133	412.020	415.291	412.500	412.500	412.500
199	413.384	412.045	415.494	412.500	412.500	412.500
200	412.565	411.107	415.498	411.500	411.500	411.500
201	412.810	411.154	415.024	411.500	411.500	411.500
202	412.500	411.099	414.965	411.500	411.500	411.500
203	412.588	411.117	415.043	411.500	411.500	411.500
204	412.771	411.118	415.009	411.500	411.500	411.500
205	412.472	411.071	414.960	411.500	411.500	411.500
206	412.953	411.120	415.171	411.500	411.500	411.500
207	412.835	411.094	414.979	411.500	411.500	411.500
208	412.667	411.098	414.888	411.500	411.500	411.500
209	412.835	411.094	414.877	411.500	411.500	411.500
210	412.667	411.098	415.242	411.500	411.500	411.500

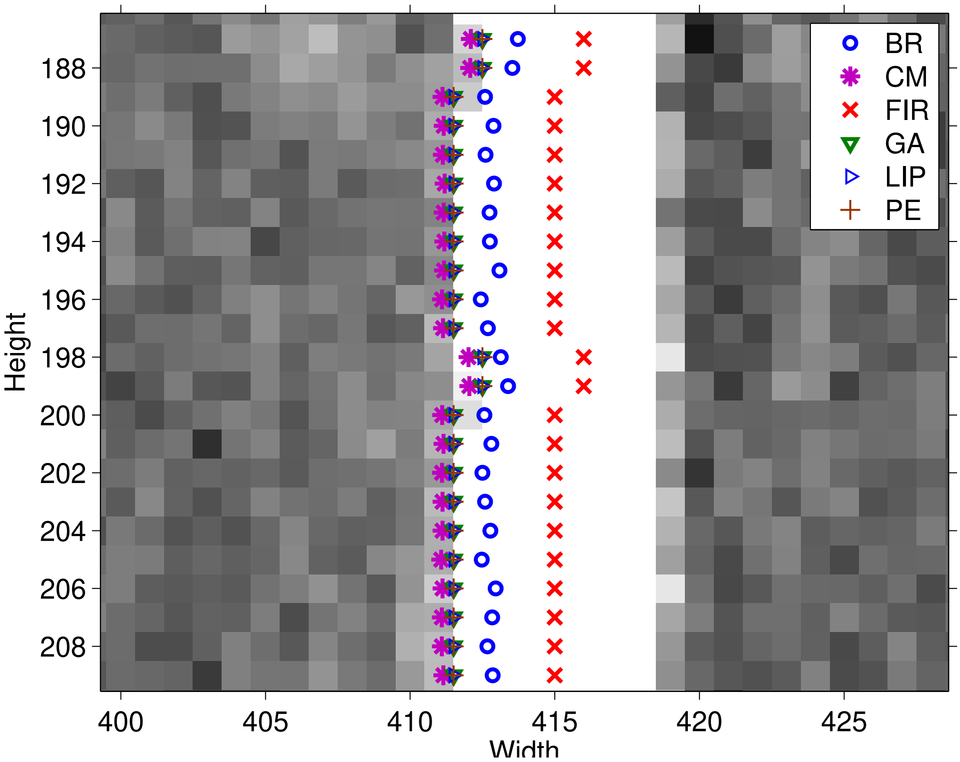


Figure 6.27: Laser peak detection result on saturated image by different methods. Image size: 720 × 576

6.3.2 Artificial Image

Sometimes it is hard to control laser property (such as the stripe width, strength) and find out and determine the ground truth laser peak location. In addition, the scanning object also affects the laser reflecting back to the camera and causes variable light saturation. Hence, in this experiment, controllable artificial laser data is generated to evaluate the peak detectors' performance for laser stripes under different conditions. The effect of laser peak deviation ε in 3D can be computed based on the triangulation equation 5.1.1 and 5.1.5 in section 5.1. Assuming the ground true laser peak (x, y) and its measured value (x', y') has deviation $(\triangle x, \triangle y)$. To be simplified, we set $x < C_x$, $\beta < 90°$, and its 3D coordinates (X, Y, Z) can be estimated as follows:

$$X = X_0 + n\Delta x \tag{6.3.1}$$

$$Z = \frac{Df}{(C_x - x) + f\cot(\alpha)} \tag{6.3.2}$$

$$Y = \frac{Z(y - C_y)}{f} \tag{6.3.3}$$

where Δx is the linear stage moving step, n is the number of steps moved so far, and X_0 is the standing position. The deviation ε between the true (X, Y, Z) and measured value (X', Y', Z') in 3D can be computed from their absolute distance:

$$\varepsilon = \sqrt{(X - X')^2 + (Y - Y')^2 + (Z - Z')^2} \tag{6.3.4}$$

6.3.2.1 Experimental setup

Here is the set-up for this simulation system. The camera and scanner parameters were the same as in the corner detection experiment as shown in Table.6.9.

Table 6.9: Camera and scanner parameters

f_x	f_y	C_x	C_y	Angle α	Base Line D
1002.84	1066.59	341.89	294.18	61.77°	97.30

Where Angle α is the laser projection angle and Base Line D is the laser to camera CCD centre's distance in millimetre. The system was underwater and the mono camera has a resolution of 720 × 576. [83] method was used for camera calibration. Since laser peak detection can be carried out in a row-by-row parallel process, the single row data can be generated as follows, leading the entire laser stripes to be generated:

- The single row data dimension is $[1 \times 720]$
- The maximum intensities(255) were in the middle of the image, varying from $2 \sim 10$ pixels on both sides to simulate real saturation

- Random noise (fewer than ten pixels) was added to the data except the peak to simulate the background noise signal. This is because the noise effect on the saturated part is so weak that can be ignored.
- Repeat the above steps to generate an image [576 × 720] with laser stripe in the middle.

Fig.6.28 shows one of the artificial examples with three pixels saturated laser row data. The enlarged saturated area is shown in Fig.6.29.

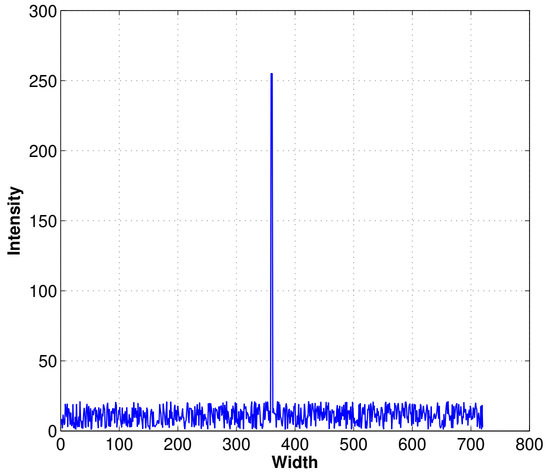


Figure 6.28: Artificial saturated laser row data

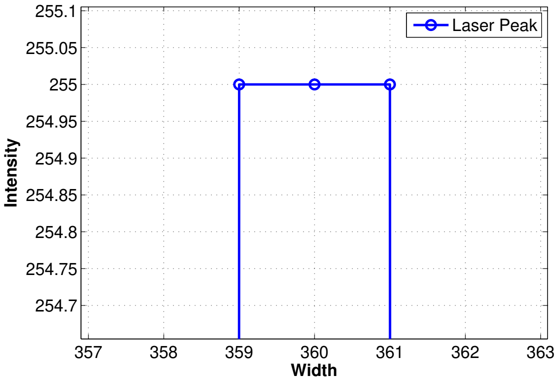


Figure 6.29: Enlarged saturated area in artificial laser row data

6.3.2.2 Results and analysis

The final detection result is shown in Table.6.10. The first column is different saturated laser peak width in pixels. Second column is the ground truth peak location. It contains two values: 360.00 and 359.50 because of the odd and even peak width. Column r is the detected peaks by the six detection methods respectively. σ_1 is the absolute error in pixel in the image plane and σ_2 is the absolute error in millimetre in 3D space after reconstruction.

At low saturated level (below 3 pixels), all methods detect the peak very precisely and their absolute errors σ_1 are below 0.5 pixels. GA, LIP and PEM even have zero pixel error as the laser stripe in two pixels width. GA, LIP and PEM have very similar results in different saturation levels. It is because they all only consider the highest three intensities and fail as $SW >= 3$. When the saturated area larger than three pixels, the σ_1 increase sharply. Only FIR maintains the error well as the saturation getting heavier. Its maximum absolute error is only 0.007 pixels as the saturated laser width increases from 2 $\sim$ 10 pixels. Based on the scanner and camera parameters in Table.6.9 and peak detection result, these laser peaks were reconstructed to 3D and then their deviation can be computed and shown in the column σ_2. Overall, in low saturation, the errors σ_2 are close to zero. When the saturated area has the maximum width of 10 pixels, CM has the largest error: 4.024 pixels in 2D and $1.332mm$ in 3D. FIR not only considers the laser frequency but also uses first order directly to locate the peak. Hence, it provides the best result in the different saturation conditions with the error in 3D being only $0.001mm$.

Table 6.10: Laser peak detection result on artificial laser stripe image. Image size: 720 × 576, Ref: Ground truth peak; SW: Saturated area width; r: Detected peak by different methods; σ_1: Absolute distance error in 2D, Unit: pixel; σ_2: Absolute distance error in 3D, Unit: mm.

SW	Ref	Blais			CM			FIR		
		r	σ_1	σ_2	r	σ_1	σ_2	r	σ_1	σ_2
2	359.500	359.492	0.008	0.001	359.476	0.024	0.009	359.496	0.004	0.001
3	360.000	360.009	0.009	0.001	359.497	0.503	0.172	360.005	0.005	0.001
4	359.500	360.036	0.536	0.160	358.494	1.006	0.338	359.500	0.000	0.000
5	360.000	359.992	0.008	0.009	358.482	1.518	0.502	359.998	0.002	0.001
6	359.500	358.984	0.516	0.166	357.474	2.026	0.669	359.500	0.000	0.000
7	360.000	359.008	0.992	0.325	357.476	2.524	0.825	359.993	0.007	0.001
8	359.500	357.995	1.505	0.491	356.479	3.021	0.995	359.500	0.000	0.000
9	360.000	357.988	2.012	0.659	356.485	3.515	1.163	360.001	0.001	0.001
10	359.500	356.972	2.528	0.834	355.476	4.024	1.332	359.500	0.000	0.000

SW	Ref	GA			LIP			PEM		
		r	σ_1	σ_2	r	σ_1	σ_2	r	σ_1	σ_2
2	359.500	359.500	0.000	0.000	359.500	0.000	0.000	359.500	0.000	0.000
3	360.000	359.500	0.500	0.164	359.500	0.500	0.164	359.500	0.500	0.164
4	359.500	358.500	1.000	0.329	358.500	1.000	0.329	358.500	1.000	0.329
5	360.000	358.500	1.500	0.494	358.500	1.500	0.494	358.500	1.500	0.494
6	359.500	357.500	2.000	0.659	357.500	2.000	0.659	357.500	2.000	0.659
7	360.000	357.500	2.500	0.824	357.500	2.500	0.824	357.500	2.500	0.824
8	359.500	356.500	3.000	0.991	356.500	3.000	0.991	356.500	3.000	0.991
9	360.000	356.500	3.500	1.156	356.500	3.500	1.156	356.500	3.500	1.156
10	359.500	355.500	4.000	1.324	355.500	4.000	1.324	355.500	4.000	1.324

The laser peaks were reconstructed into 3D based on the camera and scanner parameters. From Equations 6.3.1, 6.3.3, 6.3.2 and Table 6.9, assuming the camera and scanner parameters are fixed, laser peak $P(x, y)$ has deviation δ in x (y is the row number), then its 3D accuracy totally relies on the peak detection method. However, the relationship between the peak location error and its 3D reconstruction error is not linear. For example, assuming the ground truth peak location (x_i, y_i) in the row data is from $x_i = 1, 2, 3 \cdots 718, 719, 720$ and $y_i = 1$. If each observed peak has one pixel detection error, then its 3D reconstruction error can be computed by the camera and scanner parameters mentioned above. The result is shown in Fig.6.30. The bottom left corner is set as $(0, 0)$, the width represents the X axis in the image coordinate system. The

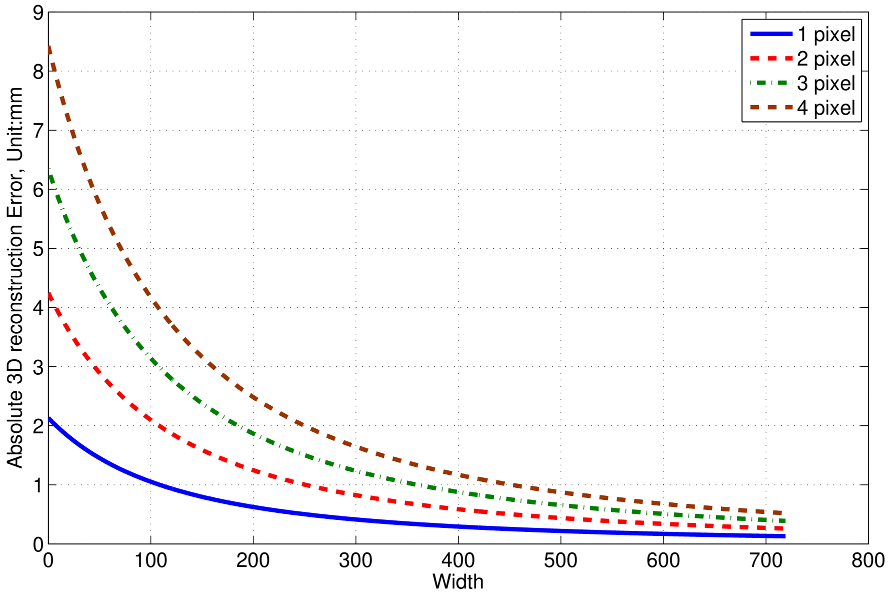


Figure 6.30: Error between observed peak position and ground truth in 3D reconstruction. Observed peak position has an error between 1 $\sim$ 4 pixels. Calibration and 3D reconstruction parameters: $f_x = 1002.84, f_y = 1066.59, C_x = 341.89, C_y = 294.18, \alpha = 61.77°, BaseLine = 97.30mm$

height represents the absolute 3D reconstruction error in different level of laser peak pixel deviation(from 1 $\sim$ 4 pixels) in different X position. In our scanning system , the closer the object to the scanner, the closer the laser stripe is toward the right side of the image as width $>$ 360 pixel; on the contrary, the farther the object, the more extent the laser stripe moves toward the left (width$<$ 360). Clearly, peak detection error has much heavier impact on 3D accuracy as the laser stripe on the image left hand side. In other words, if the distance between the object and scanner increase, then the 3D accuracy drops sharply. When the laser stripe locates larger than 400 in X axis, its absolute 3D error is below $1mm$ even the laser peak location error with large as 3 pixels. However, in the range below 100 in X axis, the absolute 3D error dramatically increases to $8.5mm$. This is consistent with reality, as the distance between the scanner and object increases, each pixel in the image represents a larger distance and causes the 3D reconstruction accuracy decrease. Thus, the scanning object cannot be too far away or too close from the scanner. Too far will lose the accuracy and too close may cause saturation due to the object surface's reflection or too high laser power.

6.3.3 Conclusion

In this experiment we compared the different peak detection methods' performance in different levels of laser saturation. Most of these methods only considering the highest intensity value of the laser stripe are working well over low saturation laser stripes. Obviously, they are not accurate enough in heavily saturated laser stripes. As the laser stripe's width is over 3 pixels, all the methods except FIR start increasing detection error. GA, CM, LIP, PE and BR only consider the maximum light intensity and the nearby three to five pixels to locate the peak in sub-pixel level. They are applicable in laser stripe without heavy saturation since the laser energy pattern of such a stripe corresponds to a Gaussian profile. However, in high resolution size or high saturated laser stripe image, the maximum light intensity(255) may cross over five pixels, the laser energy pattern is no longer corresponds to Gaussian profile. Therefore, they detected faulty laser peaks which are shifted to left in those cases. In this experiment, FIR filter has the best performance in high saturated laser stripe image. The strong point of FIR peak detector is the analysis of the row signals in the frequency domain, which yields to the cut-off frequency and the transition band width for obtaining the coefficients of an optimized filter[24]. Since the laser stripe row data is filtered by convolution that it is easy to estimate the laser peak in the saturated part without shifted.

On the other hand, when the camera and scanner parameters fixed, the laser peak and its 3D position relation is not linear. Hence, depending on where is laser stripe on the image and its detection error, 3D reconstruction differs with various levels of 3D error. One pixel of peak detection error would cause up to $2mm$ error in 3D reconstruction based on current scanner set-up. Therefore, FIR laser detection is the most suitable method in our system because of its strong resistance to laser saturation which often happens in modern camera and underwater scanning environment.

6.4 Fraction Lens Distortion Model

In this section, we evaluate the proposed camera distortion model(FMC) described in section.4.4.1 using (1) real images from [83] in [7] for the camera calibration and image correction; (2) real images from our 3D scanning system on its performance, including the camera calibration, image correction and their effect on the final 3D reconstruction accuracy.

6.4.1 Zhang's data

Since it is hard to have ground truth data that we use the public Zhang's data to evaluate the proposed FMC algorithm for camera calibration and image correction in this section.

6.4.1.1 Experimental setup

The camera to be calibrated is an off the shelf PULNix CCD camera with 6 mm lens. The image resolution is 640×480. The model plane contains a pattern of 8×8 squares. In total, there are 256 corners. The size of the pattern is $17cm \times 17cm$. Five images of the plane under different orientations were taken as shown in 6.31. The camera distortion model was explicitly described as the second order radial and decentring as Eq.6.4.1, instead of just the first-order radial one in the original Zhang's algorithm. We called it "Zhang-RD" in this chapter to differentiate from the original one since the camera may have decentring distortion.

$$\begin{aligned} x_{di} &= x_{ui}(1 + k_1 r_{ui}^2 + k_2 r_{ui}^4) + 2v_1 x_{ui} y_{ui} + v_2(r_{ui}^2 + 2x_{ui}^2)) \\ y_{di} &= y_{ui}(1 + k_1 r_{ui}^2 + k_2 r_{ui}^4) + v_1(r_{ui}^2 + 2y_{ui}^2 + 2v_2 x_{ui} y_{ui}) \end{aligned} \tag{6.4.1}$$

Assuming that the data from Zhang's website is ground truth, in order to simulate the real world imaging noise due to the quantization, different reflectance properties of the object surface, mechanical errors, etc., the Gaussian white noise with standard deviation $\delta = 0.04\tau$ was added to the coordinates of the 3D world points $p_{wi}(i = 1, 2, \cdots 256)$ and $\delta = 0.02\tau$ was added to image points P_{fi} where τ varied from 0 to 25 at intervals of 1, simulating different levels of noise. Each calibration method will run 100 times at each noise level, and their average is calculated as final result. Due to the fact that the chessboard pattern in Fig.6.31 contains multiple lines, it is able to get a further idea about the accuracy of the proposed FMC algorithm. We manually extracted the coplanar points which should lie on 32 lines, segments in each Zhang's image, total 160 in five images. The coplanar points are corrected using the distortion parameters estimated by the Zhang, Zhang-RD and FMC respectively and then we computed the best-fit linear regression coefficients for each line and sum of squares of the residuals from the best-fit line. The performance of the algorithms is measured as: maximum fitting error (MFE), average fitting error (AFE) and the standard deviation of fitting errors (SDFE) of points on these line segments.

---

[7] Access data: 2012-3-12 http://research.microsoft.com/en-us/um/people/zhang/Calib/

6.4.1.2 Results and analysis

The result is shown in Tables 6.11, 6.12, 6.13 and Figures 6.32, 6.33 and 6.34. As mentioned above, each calibration method runs 100 times at each noise level and their average as the final result. The standard deviation σ for each parameter is also shown in above tables. It can be clearly observed from these table and figures that:

1. Zhang and Zhang-RD produce quite similar intrinsic parameters result in different noise levels. It is interesting to point out that FMC produces slightly different principal point (C_x, C_y) but still successfully undistorted the images from the result of collinear fitting. It is because Zhang, Zhang-RD distortion model actually using very close to each other. Also the image correction is the principal point together with distortion factors, different distortion model may lead to different distortion factors and principal point. Besides, f_x, f_y are mostly used for describing the real camera focal length, hence, they should be the same even in different methods. In the experiment result, when $\tau = 0$, all the methods have almost the same result.

2. The proposed FMC algorithm is always more accurate than other algorithms in the sense of correcting the distorted image points when the imaging noise is heavy enough ($\tau > 5$, for example). FMC has much smaller σ in AFE and MFE at the high noise levels ($15 < \tau < 25$), which means FMC has better noise resistance. $SDFE$ describes the discrete levels of line fitting errors. Together with AFE, it can indicate the overall line fitting error. It can be seen that FMC maintains the smallest fitting error for the overall line segments.

3. In the low noise levels (as $\tau <= 5$), all of the methods maintains the calibration parameters error below one pixel. Zhang and Zhang-RD are slightly better than the proposed FMC algorithm. This is because the former two algorithms make full use of the prior knowledge of the camera distortion models. However, as the noise level increasing, the noise weakens the regularities existing in the data about the distortions that the camera was subject to. In this case, they fail to take the advantage of the prior knowledge of the camera distortions for the calibration and correction. This shows that the knowledge of radial and decentring distortions does not necessarily always bring benefits to Zhang and Zhang-RD algorithm for accurate camera calibration and image correction.

4. In different noise levels, FMC, Zhang and Zhang-RD have very close re-projection error (RPE). Thus, it indicates the camera lens doesn't contain decentring distortion or small enough to be ignored. Otherwise Zhang-RD should have smaller RPE than the other two as it fully uses the prior knowledge of the camera distortion model. It is consistent with collinear fitting results. Zhang-RD produces the biggest AFE and MFE errors as the noise increases.

5. Heavy noise usually leads to less accurate estimation of the parameters of interest, which is often within expectation.

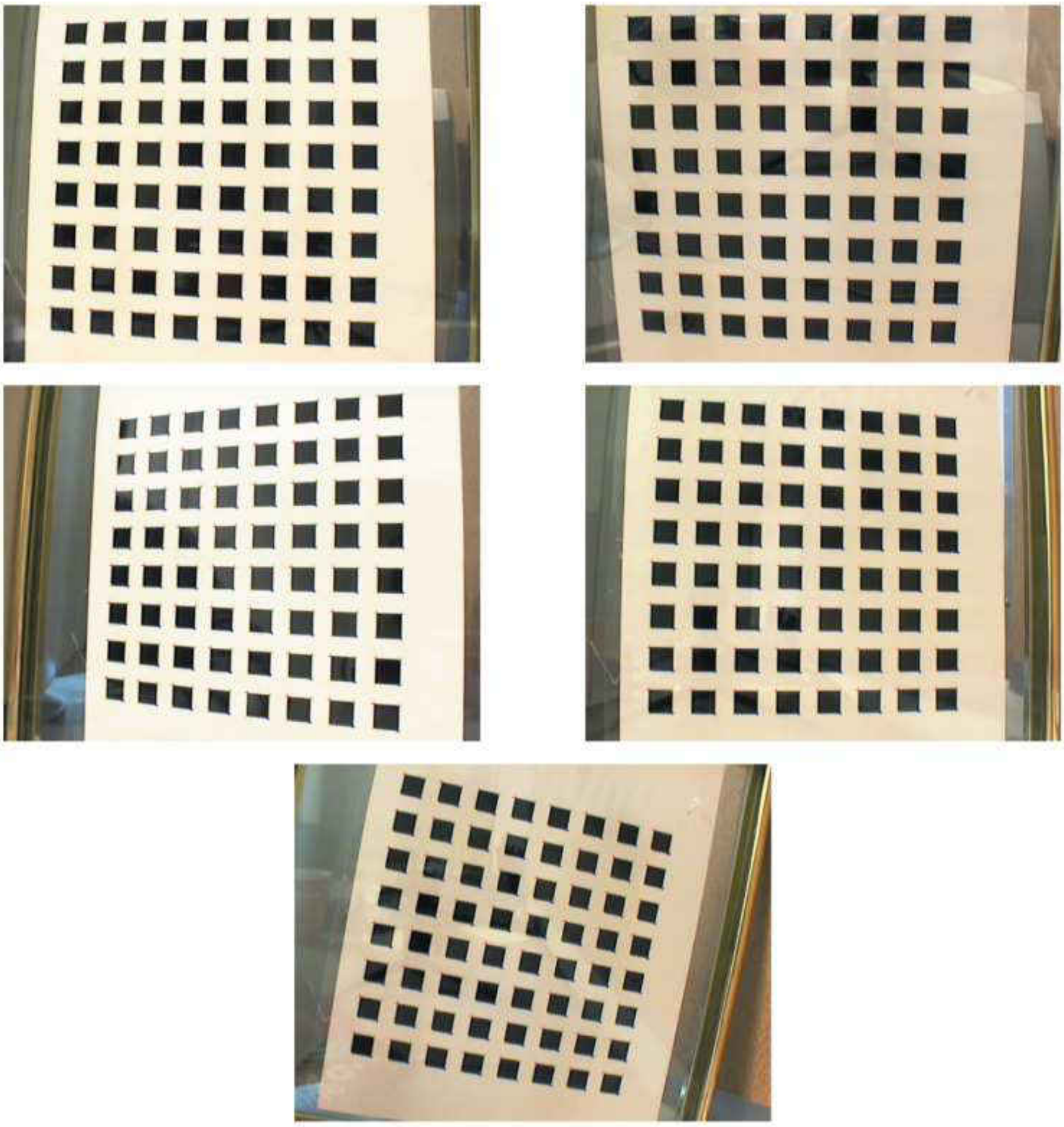


Figure 6.31: Five images of a model plane from [83]

Table 6.11: Calibration and line segment fitting result using Zhang method. The image resolution is 640 × 480. The model plane contains a pattern of 8×8 squares ($17cm \times 17cm$), in total there are 256 corners. Gaussian white noise with standard deviation $\delta = 0.04\tau$ was added to the coordinates of the 3D world points $p_{wi}(i = 1, 2, \cdots 256)$ and $\delta = 0.02\tau$ was added to image points P_{fi} where τ varied from 0 to 25 at intervals of 1. μ: average result for each parameter; σ: standard deviation for each parameter, RPE: re-projection error

	f_x		f_y		C_x		C_y		RPE		AFE		MFE		SDFE
τ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	
0	832.148	0.000	832.183	0.000	304.061	0.000	206.384	0.000	0.337	0.000	0.187	0.000	0.784	0.000	0.135
1	832.183	0.294	832.189	0.326	303.885	0.476	206.511	0.309	0.364	0.003	0.187	0.002	0.785	0.003	0.136
2	832.087	1.064	832.131	1.073	304.234	0.631	206.404	1.201	0.435	0.009	0.190	0.005	0.786	0.007	0.137
3	832.464	0.907	832.497	0.944	303.924	0.907	206.430	1.114	0.535	0.009	0.190	0.005	0.788	0.005	0.136
4	831.652	2.093	831.689	2.050	304.570	1.497	206.405	2.460	0.638	0.015	0.200	0.013	0.819	0.021	0.145
5	832.044	1.748	831.986	1.737	303.840	2.623	206.608	2.002	0.759	0.019	0.200	0.009	0.813	0.044	0.145
6	831.438	2.937	831.451	2.806	304.538	3.309	206.393	2.526	0.891	0.028	0.206	0.032	0.845	0.306	0.151
7	832.551	2.405	832.662	2.524	303.481	2.479	205.917	3.374	1.022	0.021	0.209	0.038	0.859	0.210	0.150
8	832.949	3.172	832.868	3.198	303.903	3.645	206.946	4.431	1.149	0.031	0.218	0.023	0.910	0.249	0.161
9	832.527	3.913	832.622	3.999	302.196	5.334	205.034	2.812	1.277	0.034	0.233	0.028	1.219	0.110	0.186
10	833.378	5.158	833.308	5.062	303.350	5.654	206.886	4.966	1.416	0.039	0.224	0.045	0.934	0.650	0.167
11	831.866	4.370	831.907	4.430	304.135	7.463	205.031	3.997	1.566	0.040	0.260	0.116	1.351	1.283	0.219
12	831.703	4.869	831.808	5.005	304.376	6.782	206.026	4.252	1.690	0.036	0.286	0.070	1.481	0.762	0.250
13	830.850	5.273	831.129	5.626	304.265	5.718	207.319	4.946	1.830	0.052	0.275	0.090	1.753	0.496	0.263
14	832.742	4.619	832.704	4.401	304.849	3.523	204.539	4.850	1.925	0.049	0.302	0.082	1.394	0.725	0.243
15	831.525	6.074	831.768	6.299	306.811	5.335	206.839	5.169	2.101	0.049	0.358	0.059	2.436	0.832	0.354
16	833.553	5.918	833.838	5.667	301.939	6.349	203.265	6.878	2.251	0.051	0.365	0.094	2.091	0.955	0.339
17	830.061	6.459	829.876	6.787	307.711	9.967	206.798	5.196	2.396	0.056	0.359	0.116	2.033	1.295	0.333
18	831.997	7.707	831.870	7.812	302.127	12.507	205.032	7.909	2.512	0.098	0.379	0.158	2.783	3.084	0.400
19	830.989	8.427	831.000	8.532	306.275	9.607	207.057	7.454	2.654	0.095	0.463	0.257	3.103	1.471	0.488
20	829.756	8.712	829.675	8.486	305.457	11.801	204.950	9.035	2.790	0.076	0.472	0.219	3.483	4.639	0.544
21	834.123	10.524	834.065	10.021	302.385	14.058	203.349	13.396	2.938	0.068	0.497	0.165	3.499	2.300	0.552
22	832.421	5.431	832.578	6.033	305.984	11.636	208.185	8.609	3.100	0.092	0.613	0.157	6.625	3.305	0.903
23	831.725	6.325	831.420	6.315	298.930	10.482	203.263	9.732	3.202	0.050	0.574	0.295	6.165	2.412	0.826
24	829.874	12.705	829.917	12.633	303.947	11.110	206.710	13.674	3.340	0.114	0.610	0.370	6.185	3.976	0.900
25	831.215	14.351	831.278	14.255	302.647	16.056	204.747	16.793	3.491	0.120	0.702	0.565	9.454	7.931	1.150

Table 6.12: Calibration and line segment fitting result using Zhang-RD method. The image resolution is 640 × 480. The model plane contains a pattern of 8×8 squares ($17cm \times 17cm$), in total there are 256 corners. Gaussian white noise with standard deviation $\delta = 0.04\tau$ was added to the coordinates of the 3D world points $p_{wi}(i = 1, 2, \cdots 256)$ and $\delta = 0.02\tau$ was added to image points P_{fi} where τ varied from 0 to 25 at intervals of 1. μ: average result for each parameter; σ: standard deviation for each parameter, RPE: re-projection error

	f_x		f_y		C_x		C_y		RPE		AFE		MFE		SDFE
τ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	
0	832.883	0.000	832.820	0.000	304.139	0.000	208.616	0.000	0.334	0.000	0.185	0.000	0.786	0.000	0.144
1	832.928	0.252	832.878	0.250	304.343	0.501	208.558	0.724	0.362	0.005	0.188	0.003	0.785	0.007	0.145
2	832.890	0.403	832.842	0.413	303.973	0.756	208.714	1.040	0.434	0.006	0.188	0.005	0.793	0.006	0.144
3	833.245	1.105	833.250	1.107	303.887	1.397	208.140	2.663	0.529	0.011	0.193	0.009	0.801	0.014	0.146
4	832.799	0.618	832.671	0.680	304.416	2.790	209.824	2.424	0.640	0.015	0.206	0.015	0.909	0.047	0.161
5	833.030	0.905	832.923	0.961	304.735	3.645	208.403	2.586	0.759	0.021	0.203	0.033	0.812	0.346	0.152
6	833.476	2.052	833.431	2.194	303.408	2.263	208.534	4.411	0.883	0.022	0.206	0.029	0.855	0.198	0.156
7	832.122	1.860	832.082	1.885	306.346	5.452	210.555	5.248	1.035	0.022	0.246	0.038	1.196	0.163	0.199
8	833.379	1.913	833.470	1.824	305.449	6.188	209.708	4.731	1.159	0.035	0.252	0.062	0.986	0.481	0.189
9	833.110	3.177	832.769	3.044	303.050	7.988	209.164	3.833	1.289	0.035	0.253	0.095	1.313	0.720	0.210
10	833.529	3.283	833.512	3.277	306.742	5.831	209.490	4.978	1.417	0.033	0.300	0.068	1.597	0.637	0.249
11	833.132	3.192	832.794	3.364	302.449	6.859	208.363	6.941	1.553	0.041	0.284	0.070	1.174	0.783	0.222
12	832.955	2.979	832.840	3.128	306.159	3.960	208.569	7.647	1.683	0.040	0.327	0.065	1.872	0.460	0.279
13	833.637	3.264	833.432	3.411	305.129	6.262	208.283	6.918	1.837	0.042	0.339	0.055	1.652	0.706	0.270
14	832.319	3.699	832.134	3.414	302.293	7.961	205.847	7.296	1.954	0.045	0.381	0.113	1.814	1.025	0.297
15	832.561	4.224	832.410	4.472	303.811	9.357	209.465	9.638	2.097	0.062	0.426	0.098	2.858	1.537	0.425
16	832.812	5.670	832.556	5.808	303.196	11.129	210.300	9.354	2.209	0.083	0.380	0.190	2.277	2.336	0.337
17	830.086	5.507	830.324	5.219	300.389	9.914	206.816	7.928	2.358	0.080	0.512	0.149	4.168	3.162	0.582
18	834.430	6.635	834.230	6.634	304.023	14.965	210.901	13.855	2.493	0.057	0.480	0.154	3.388	1.685	0.504
19	830.728	7.350	830.737	7.326	303.742	12.393	204.751	9.482	2.620	0.073	0.628	0.160	5.266	1.785	0.744
20	832.939	5.525	832.762	5.661	306.235	11.416	207.555	12.022	2.821	0.057	0.637	0.203	3.843	1.754	0.625
21	832.334	8.226	832.552	8.067	300.986	13.534	209.819	13.634	2.883	0.118	0.603	0.362	4.638	4.744	0.735
22	833.399	7.414	833.424	7.248	303.598	12.777	209.697	23.241	3.046	0.078	0.672	0.330	5.481	4.405	0.758
23	832.742	9.037	832.660	8.923	302.526	14.340	210.522	14.869	3.166	0.093	0.722	0.343	6.202	5.838	0.848
24	834.098	10.131	834.307	10.381	305.370	18.509	212.040	16.002	3.325	0.101	0.742	0.453	5.188	6.264	0.809
25	834.058	8.498	833.893	8.610	302.267	17.693	210.363	13.088	3.479	0.102	0.876	0.393	7.917	6.810	1.062

Table 6.13: Calibration and line segment fitting result using FMC method. The image resolution is 640×480. The model plane contains a pattern of 8×8 squares ($17cm \times 17cm$), in total there are 256 corners. Gaussian white noise with standard deviation $\delta = 0.04\tau$ was added to the coordinates of the 3D world points $p_{wi}(i = 1, 2, \cdots 256)$ and $\delta = 0.02\tau$ was added to image points P_{fi} where τ varied from 0 to 25 at intervals of 1. μ: average result for each parameter; σ: standard deviation for each parameter, RPE: re-projection error

	f_x		f_y		C_x		C_y		RPE		AFE		MFE		SDFE
τ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	
0	832.266	0.000	832.217	0.000	298.300	0.000	214.994	0.000	0.333	0.000	0.189	0.000	1.089	0.000	0.164
1	832.278	0.244	832.241	0.251	298.038	1.013	215.298	1.074	0.361	0.004	0.190	0.003	1.101	0.066	0.166
2	832.356	0.566	832.300	0.550	298.112	2.304	215.019	2.373	0.430	0.008	0.193	0.006	1.126	0.145	0.169
3	832.261	0.687	832.226	0.751	297.770	3.125	215.761	3.652	0.530	0.013	0.195	0.009	1.106	0.165	0.168
4	832.235	0.675	832.173	0.664	299.081	4.800	215.550	5.716	0.646	0.015	0.204	0.010	1.141	0.232	0.176
5	832.431	0.935	832.424	0.898	297.179	8.184	216.307	6.733	0.770	0.015	0.211	0.017	1.231	0.362	0.183
6	832.538	1.532	832.379	1.591	300.106	6.534	217.854	7.380	0.889	0.025	0.227	0.022	1.303	0.402	0.197
7	832.593	1.660	832.439	1.732	297.167	9.542	212.990	8.660	1.023	0.041	0.219	0.041	1.172	0.528	0.180
8	831.974	1.806	831.936	1.660	301.388	8.009	215.065	9.772	1.154	0.033	0.245	0.034	1.341	0.622	0.206
9	833.073	2.760	832.866	2.811	295.034	11.222	213.457	13.398	1.279	0.033	0.265	0.047	1.455	0.595	0.214
10	832.349	2.934	832.290	2.894	297.023	10.901	215.868	13.476	1.421	0.031	0.279	0.062	1.775	0.782	0.250
11	832.038	2.095	831.943	2.202	298.263	12.192	214.441	17.201	1.538	0.059	0.267	0.107	1.429	0.741	0.218
12	832.529	3.430	832.531	3.071	296.499	14.010	216.661	20.274	1.676	0.045	0.339	0.098	1.982	0.807	0.292
13	832.245	2.777	832.236	3.029	297.289	18.749	214.924	16.532	1.820	0.057	0.349	0.141	2.082	1.332	0.317
14	831.974	3.543	831.772	3.774	304.055	18.608	210.595	19.374	1.974	0.054	0.382	0.119	2.206	1.435	0.335
15	832.146	3.302	832.322	3.202	295.577	16.791	215.149	18.930	2.077	0.059	0.396	0.094	2.722	1.127	0.387
16	831.984	3.938	831.807	4.213	292.758	15.447	221.349	19.009	2.225	0.068	0.362	0.119	2.160	1.042	0.329
17	831.162	4.231	831.403	4.430	306.635	21.156	210.800	17.808	2.350	0.065	0.421	0.140	2.497	1.315	0.383
18	833.081	4.679	833.127	4.574	299.447	16.924	214.747	29.453	2.510	0.086	0.473	0.213	2.890	1.662	0.424
19	834.188	3.633	833.971	3.711	303.222	32.629	215.571	24.828	2.618	0.087	0.458	0.306	2.417	2.186	0.395
20	831.311	5.214	831.273	5.280	299.349	28.047	208.516	22.621	2.763	0.062	0.511	0.190	3.159	1.911	0.491
21	832.231	4.246	832.316	4.372	298.640	24.195	215.856	25.934	2.926	0.081	0.521	0.258	3.108	1.884	0.472
22	832.731	5.257	832.309	4.945	296.240	24.795	216.841	25.729	3.068	0.085	0.541	0.297	3.420	2.759	0.518
23	832.662	6.153	832.502	6.852	298.135	26.619	219.275	32.884	3.203	0.105	0.584	0.200	3.955	2.351	0.596
24	831.977	5.355	831.913	5.755	296.604	21.370	222.350	32.215	3.312	0.097	0.633	0.221	4.118	2.160	0.615
25	832.161	5.817	832.173	5.628	296.691	27.989	207.961	30.942	3.443	0.106	0.645	0.315	4.127	2.382	0.650

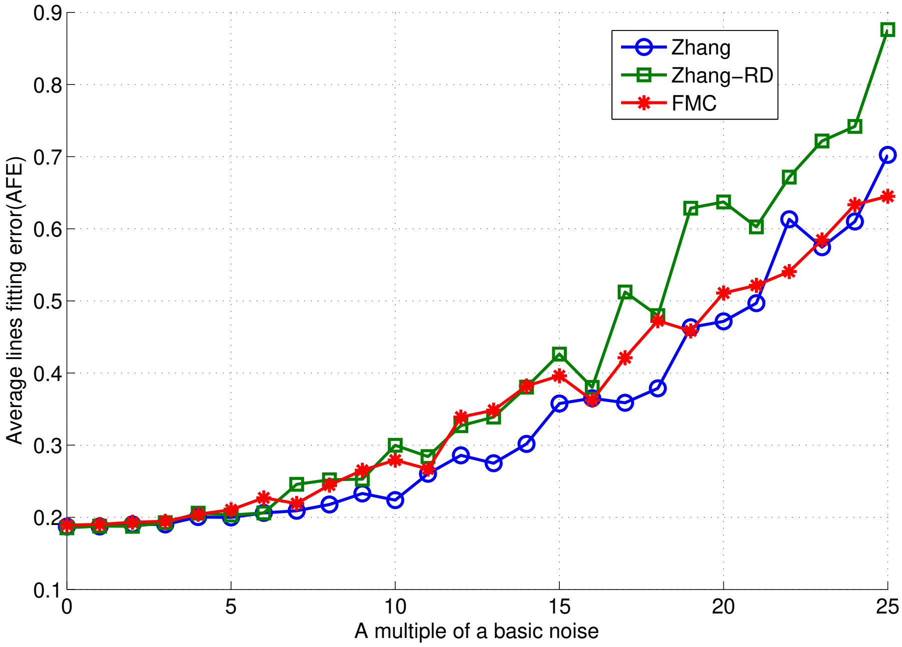


Figure 6.32: The AFE line fitting error using different algorithms based on Zhang's data. The image resolution is 640×480. The model plane contains a pattern of 8×8 squares ($17cm \times 17cm$), in total there are 256 corners. The coplanar points lie on 32 lines segments in each Zhang's image, total 160 in five images. Gaussian white noise with standard deviation $\delta = 0.04\tau$ was added to the coordinates of the 3D world points $p_{wi}(i = 1, 2, \cdots 256)$ and $\delta = 0.02\tau$ was added to image points P_{fi} where τ varied from 0 to 25 at intervals of 1.

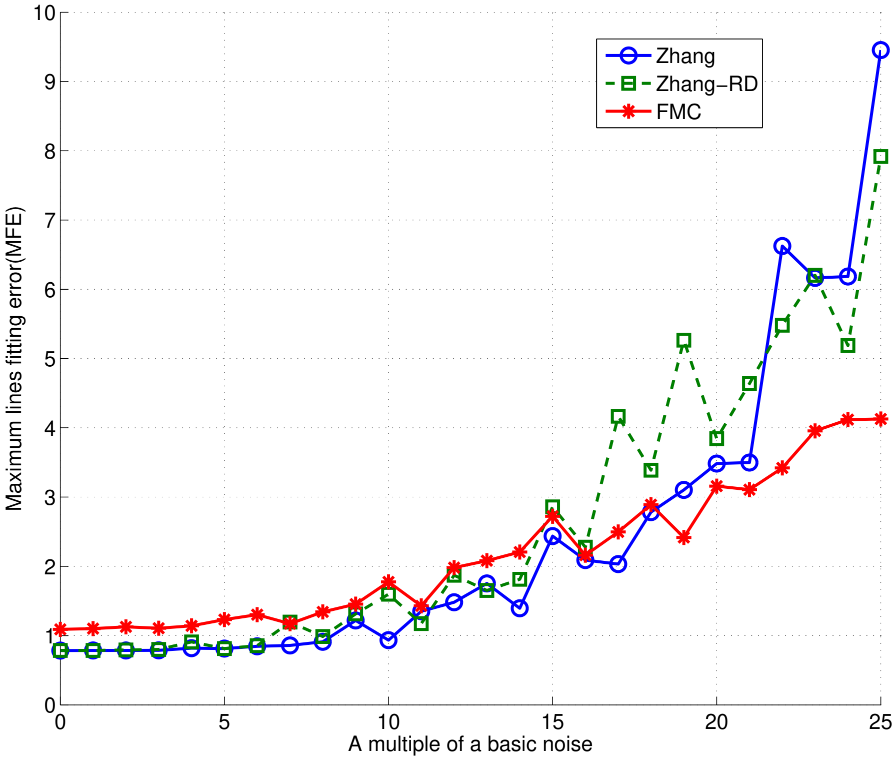


Figure 6.33: The MFE line fitting error using different algorithms based on Zhang's data. The image resolution is 640×480. The model plane contains a pattern of 8×8 squares ($17cm \times 17cm$), in total there are 256 corners. The coplanar points lie on 32 lines segments in each Zhang's image, total 160 in five images. Gaussian white noise with standard deviation $\delta = 0.04\tau$ was added to the coordinates of the 3D world points $p_{wi}(i = 1, 2, \cdots 256)$ and $\delta = 0.02\tau$ was added to image points P_{fi} where τ varied from 0 to 25 at intervals of 1.

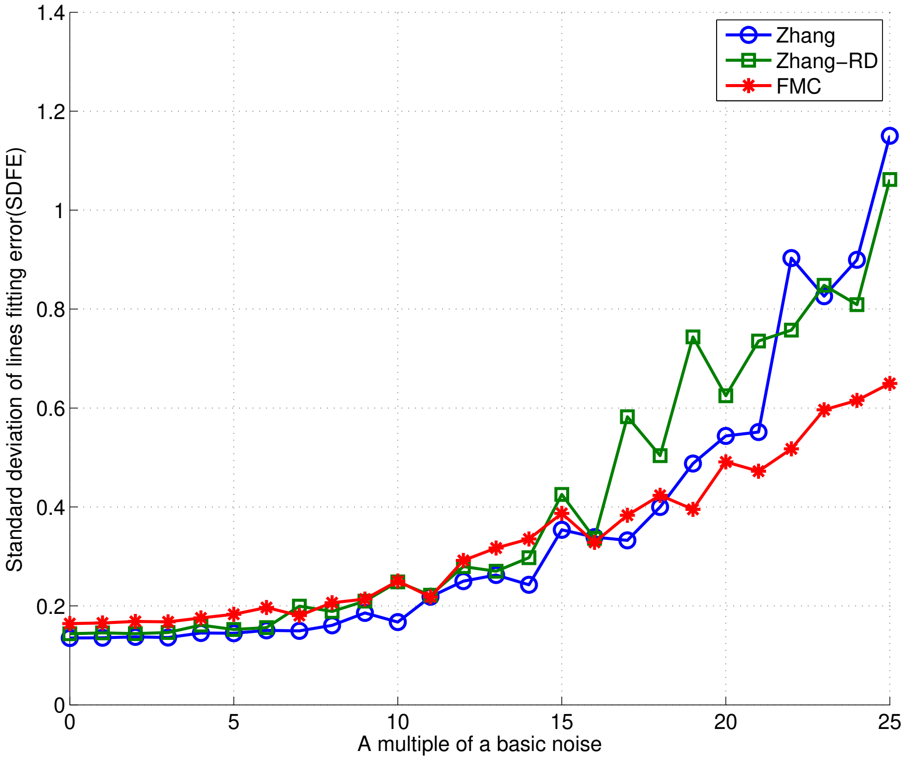


Figure 6.34: The SDFE line fitting error using different algorithms based on Zhang's data. The image resolution is 640×480. The model plane contains a pattern of 8×8 squares $(17cm \times 17cm)$, in total there are 256 corners. The coplanar points lie on 32 lines segments in each Zhang's image, total 160 in five images. Gaussian white noise with standard deviation $\delta = 0.04\tau$ was added to the coordinates of the 3D world points $p_{wi}(i = 1, 2, \cdots 256)$ and $\delta = 0.02\tau$ was added to image points P_{fi} where τ varied from 0 to 25 at intervals of 1.

6.4.2 Applied in 3D scanner data

In the previous section we evaluated the FMC using Zhang's data for camera calibration and image correction, it is important to know the influence of camera calibration and image correction to the final 3D reconstruction accuracy. In this section, we use the real image data from our 3D scanner to evaluate the proposed FMC algorithm for camera calibration, image correction and 3D reconstruction. Since it is hardly to precisely measure the lens distortion that the known 3D measurement can be used to evaluate the proposed method in the 3D reconstruction alternatively.

6.4.2.1 Experimental setup

The equipment set up was similar as section 6.2.1. The image data used in this experiment was from underwater scanning. The camera is Bowtech LCC-600 at a resolution of 720×576. Its original lens is fish-eye (focal length $= 2.9mm$) model and has a field of view (FOV) of $65°$ underwater. To reduce lens distortion effect, a pinhole lens was fitted into this camera. It has a FOV of about $42°$ in water and focal length is $5.3mm$. The calibration target was a printed and laminated chessboard, sticking onto a flat black board. It contains 8×6 corners, every corner has a distance of $20mm$ from each other. 10 chessboard images at different orientations were captured. The scanner was placed and set up for parameter estimation as the procedure described in section 5.2. It contains a vertical flat board in front of the scanner. The scanner travels toward to the flat board. From the starting point to the other end, it travels 105 steps at $1mm$ interval, then total 105 laser stripe images were captured. To simulate the different working environments/-conditions, the Gaussian white noise with standard deviation $\delta = 0.04\tau$ was added to the coordinates of the 3D world points $p_{wi}(i = 1, 2, \cdots 48)$ and $\delta = 0.02\tau$ was added to image points P_{fi} where τ varied from 0 to 25 at intervals of 1, simulating different levels of noise. Each calibration method will run 100 times on each noise level, and their average is calculated as the final result. The calibration parameters may be different under different noise levels. Thus, as the calibration parameters change, it is necessary to perform the scanner parameter estimation described in section 5.2 again to obtain a new set of scanner parameters including baseline D and angle α. Once we get the scanner parameters and calibration results, laser peak detection(Gaussian approximation) was applied to the corrected images and reconstruct the 3D model. There are two main aspects to be verified under these calibration algorithms: (1) the laser stripes in the 2D images should be "Straight" after correction using the distortion factors; and (2) The Z-axis distance should be as close as possible to the real in the 3D reconstruction.

6.4.2.2 Results and analysis

The calibration and laser stripes linear fitting results are shown in Tables 6.14, 6.15, 6.16 and Figures 6.35, 6.36, 6.37, 6.39,6.38 and 6.40. From these tables and figures, it can be observed that: Zhang, Zhang-RD, and FMC have all successfully calibrated the camera. The RPE (re-projection error) is very similar to each other at different noise levels.

However, the laser stripe correction shows different results. Since the laser stripe has been carefully adjusted to be straight physically, thus, the laser stripe in the images should have the same property. Based on that, we undistorted the images using three methods' calibration results and then linear fitting was performed on the laser stripes extracted from the images. The linear fitting error may look larger due to the fact: (1) the images were taken from underwater with some unavoidable noise, (2) calibration result may contain some error when applies air calibration methods into underwater and (3) this process only using FMC without any other components like FIR laser peak detection involved. Figure 6.35 and 6.36 show the average result: FMC has smaller fitting error and the laser stripes are closer to 90° than the other two methods. At low noise level(like $\tau < 5$), all the corrected stripes angles are minimized between $89.82^\circ \sim 89.84^\circ$ which are very close to reality. Even the noise level increases to $\tau = 25$, their angles' error is still below 0.3°. In order to get a closer look, Figures 6.37, 6.39,6.38 and 6.40 show the individual angles of the laser stripes captured at different linear stage positions by the three algorithms at different noise levels. It is clearer that as noise increases, FMC has the smallest deviation between the fitting angles and the real 90°, which means FMC has better image correction result than the other two. As the noise increases from $\tau = 0 \cdots 25$, the angles' deviation are limited to 0.3° as the laser stripes in different image positions. Average angle results Zhang and Zhang-RD are similar, however, when τ is high (like $\tau > 20$) and the laser stripe number is over 80, the maximum angles' deviation is over 0.8° by Zhang algorithm and 30° by Zhang-RD algorithm. This linear fitting result is consistent with the previous section result using Zhang's data: (1) FMC has better noise resistance; (2) the knowledge of radial and decentring distortions does not necessarily always bring benefits to Zhang and Zhang-RD algorithm for the image undistortion.

Table 6.14: Calibration result by Zhang method and real 3D scanning data corrupted by different multiples of a basic noise. The image size is 720×576 and calibration target contains $8 \times 6 = 48$ corners, every corner has a distance of $20mm$ from each other. Gaussian white noise with standard deviation $\delta = 0.04\tau$ was added to the coordinates of the 3D world points $p_{wi}(i = 1, 2, \cdots 48)$ and $\delta = 0.02\tau$ was added to image points P_{fi} where τ varied from 0 to 25 at intervals of 1. μ: average result for each parameter, σ: standard deviation for each parameter, RPE: re-projection error

	f_x		f_y		C_x		C_y		RPE	
τ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
0	1002.722	0.000	1066.884	0.000	348.681	0.000	292.736	0.000	0.270	0.000
5	1002.392	1.243	1066.501	1.294	348.942	1.231	293.477	1.232	0.664	0.040
10	1003.010	2.774	1067.175	2.779	350.093	1.691	296.004	2.162	1.224	0.060
15	1000.272	4.186	1064.157	4.442	356.045	3.220	300.527	3.702	1.839	0.123
20	1005.101	5.667	1067.928	5.663	360.504	4.447	297.403	3.514	2.394	0.145
25	1004.485	6.187	1068.166	6.612	361.601	5.422	312.405	5.082	3.065	0.144

To have better knowledge how these calibration results and image correction by these three algorithms affect the final 3D reconstruction accuracy, all these 2D laser peaks are

Table 6.15: Calibration result by Zhang-RD method and real 3D scanning data corrupted by different multiples of a basic noise. The image size is 720×576 and calibration target contains $8 \times 6 = 48$ corners, every corner has a distance of $20mm$ from each other. Gaussian white noise with standard deviation $\delta = 0.04\tau$ was added to the coordinates of the 3D world points $p_{wi}(i = 1, 2, \cdots 48)$ and $\delta = 0.02\tau$ was added to image points P_{fi} where τ varied from 0 to 25 at intervals of 1. μ: average result for each parameter, σ: standard deviation for each parameter, RPE: re-projection error.

	f_x		f_y		C_x		C_y		RPE	
τ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
0	1003.148	0.000	1067.513	0.000	341.642	0.000	295.417	0.000	0.268	0.000
5	1003.180	1.079	1067.447	1.118	344.186	1.608	293.925	1.203	0.661	0.048
10	1003.744	2.882	1067.484	2.969	350.369	2.640	297.183	2.200	1.233	0.077
15	1003.271	3.308	1067.204	3.487	349.143	4.052	294.571	3.964	1.800	0.118
20	1003.164	4.378	1066.155	4.313	355.453	6.596	299.245	4.742	2.427	0.116
25	1002.372	6.821	1064.435	7.131	360.319	8.204	296.295	5.051	3.006	0.156

Table 6.16: Calibration result by FMC method and real 3D scanning data corrupted by different multiples of a basic noise. The image size is 720×576 and calibration target contains $8 \times 6 = 48$ corners, every corner has a distance of $20mm$ from each other. Gaussian white noise with standard deviation $\delta = 0.04\tau$ was added to the coordinates of the 3D world points $p_{wi}(i = 1, 2, \cdots 48)$ and $\delta = 0.02\tau$ was added to image points P_{fi} where τ varied from 0 to 25 at intervals of 1. μ: average result for each parameter, σ: standard deviation for each parameter, RPE: re-projection error.

	f_x		f_y		C_x		C_y		RPE	
τ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
0	1003.045	0.000	1067.614	0.000	342.684	0.000	299.454	0.000	0.267	0.000
5	1003.545	1.069	1068.084	1.155	345.916	3.977	301.023	3.770	0.649	0.038
10	1004.558	2.390	1069.021	2.564	353.043	8.157	295.096	7.415	1.248	0.061
15	1006.796	3.817	1071.431	4.007	366.779	10.689	298.359	10.115	1.812	0.103
20	1004.926	4.662	1069.714	4.871	351.495	16.494	300.930	12.784	2.447	0.147
25	1006.842	5.746	1070.410	6.201	352.013	24.475	291.665	16.102	3.022	0.174

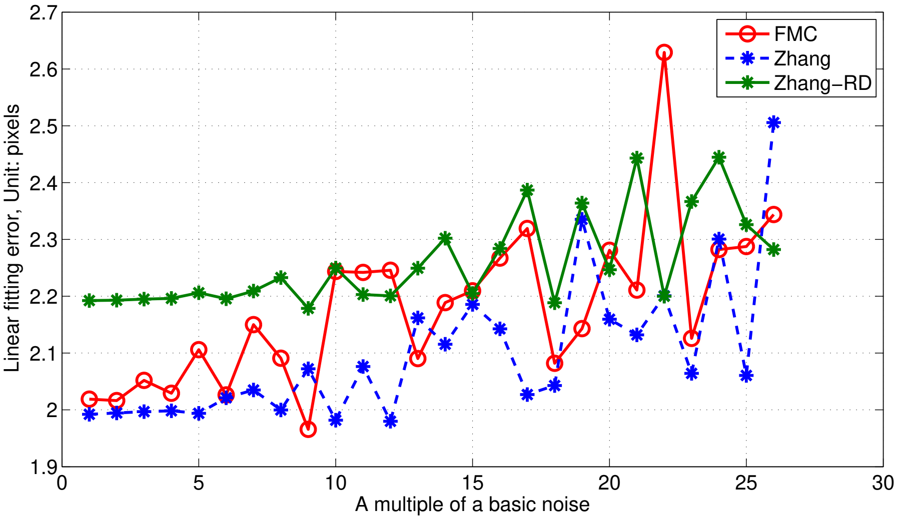


Figure 6.35: The average linear fitting error after the 2D image correction using different calibration algorithms with various multiples of a basic noise. The image size is 720 × 576, calibration target contains $8 \times 6 = 48$ corners, every corner has a distance of $20mm$ from each other. The scanner travels 105 steps at $1mm$ interval. Total there are 105 images. Gaussian white noise with standard deviation $\delta = 0.04\tau$ was added to the coordinates of the 3D world points $p_{wi}(i = 1, 2, \cdots 48)$ and $\delta = 0.02\tau$ was added to image points P_{fi} where τ varied from 0 to 25 at intervals of 1.

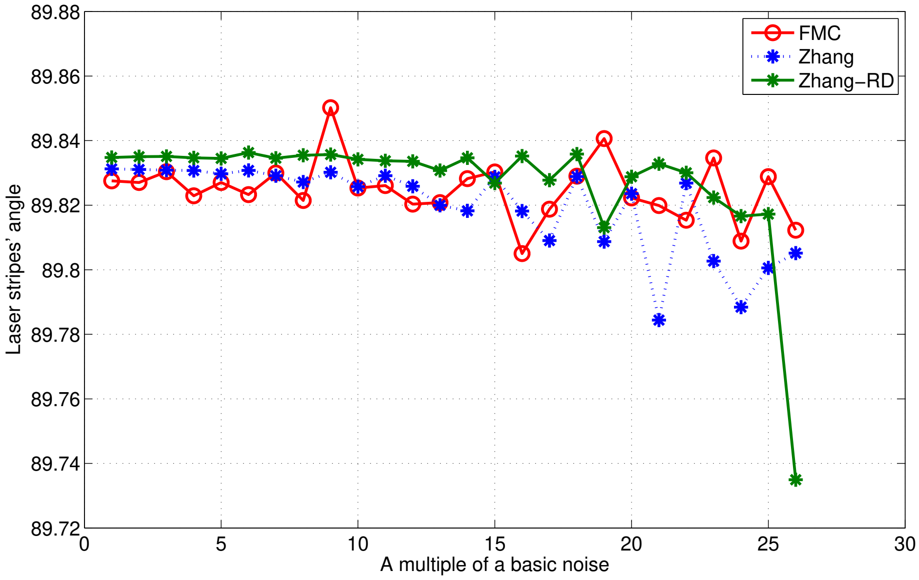


Figure 6.36: The average angle of laser stripes at each linear stage moving step based on the linear fitting after the image correction using different calibration algorithms with various multiples of a basic noise. The image size is 720×576, calibration target contains $8 \times 6 = 48$ corners, every corner has a distance of $20mm$ from each other. The scanner travels 105 steps at $1mm$ interval. Total there are 105 images. Gaussian white noise with standard deviation $\delta = 0.04\tau$ was added to the coordinates of the 3D world points $p_{wi}(i = 1, 2, \cdots 48)$ and $\delta = 0.02\tau$ was added to image points P_{fi} where τ varied from 0 to 25 at intervals of 1.

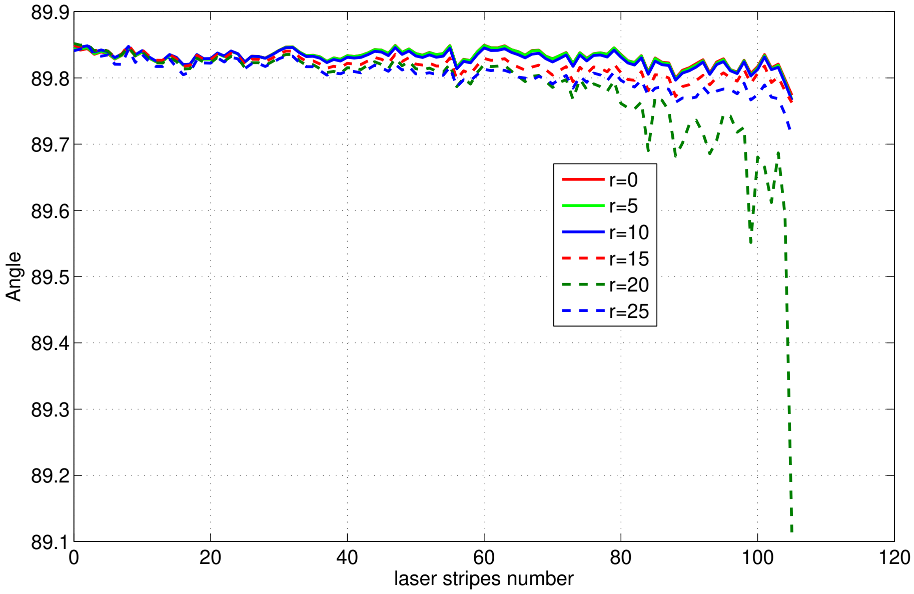


Figure 6.37: The average angle of laser stripes at each linear stage moving step based on the linear fitting after the image correction using Zhang algorithm with various multiples of a basic noise. The image size is 720 × 576, calibration target contains $8 \times 6 = 48$ corners, every corner has a distance of $20mm$ from each other. The scanner travels 105 steps at $1mm$ interval. Total there are 105 images. Gaussian white noise with standard deviation $\delta = 0.04\tau$ was added to the coordinates of the 3D world points $p_{wi}(i = 1, 2, \cdots 48)$ and $\delta = 0.02\tau$ was added to image points P_{fi} where τ varied from 0 to 25 at intervals of 1.

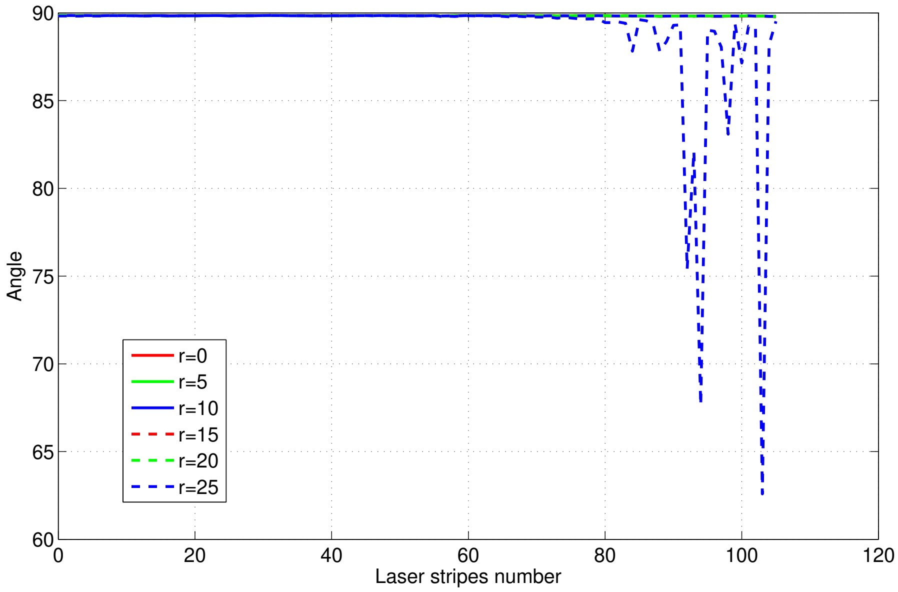


Figure 6.38: The average angle of laser stripes at each linear stage moving step based on the linear fitting after the image correction using Zhang-RD algorithm with various multiples of a basic noise.(**A**) The image size is 720×576, calibration target contains $8 \times 6 = 48$ corners, every corner has a distance of $20mm$ from each other. The scanner travels 105 steps at $1mm$ interval. Total there are 105 images. Gaussian white noise with standard deviation $\delta = 0.04\tau$ was added to the coordinates of the 3D world points $p_{wi}(i = 1, 2, \cdots 48)$ and $\delta = 0.02\tau$ was added to image points P_{fi} where τ varied from 0 to 25 at intervals of 1.

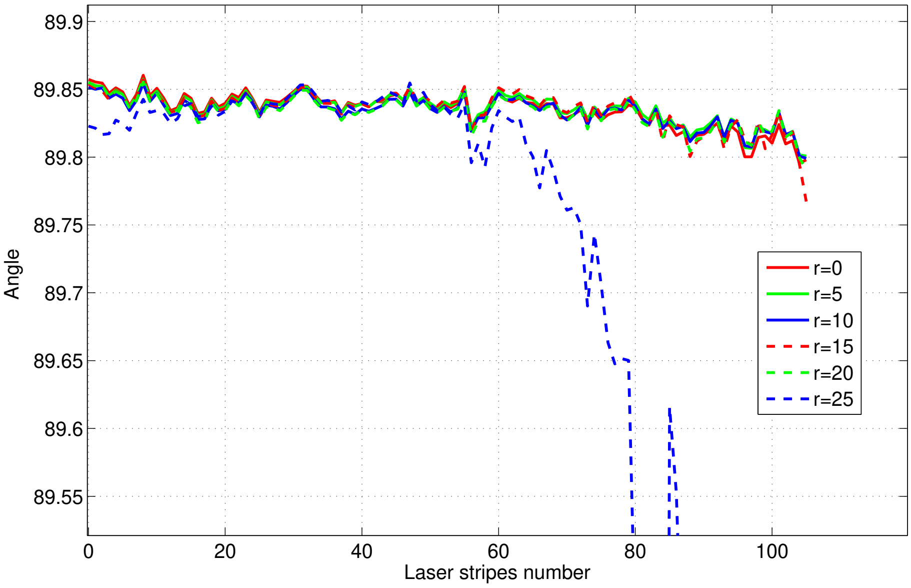


Figure 6.39: The average angle of laser stripes at each linear stage moving step based on the linear fitting after the image correction using Zhang-RD algorithm with various multiples of a basic noise.(**B**). The image size is 720×576, calibration target contains $8 \times 6 = 48$ corners, every corner has a distance of $20mm$ from each other. The scanner travels 105 steps at $1mm$ interval. Total there are 105 images. Gaussian white noise with standard deviation $\delta = 0.04\tau$ was added to the coordinates of the 3D world points $p_{wi}(i = 1, 2, \cdots 48)$ and $\delta = 0.02\tau$ was added to image points P_{fi} where τ varied from 0 to 25 at intervals of 1.

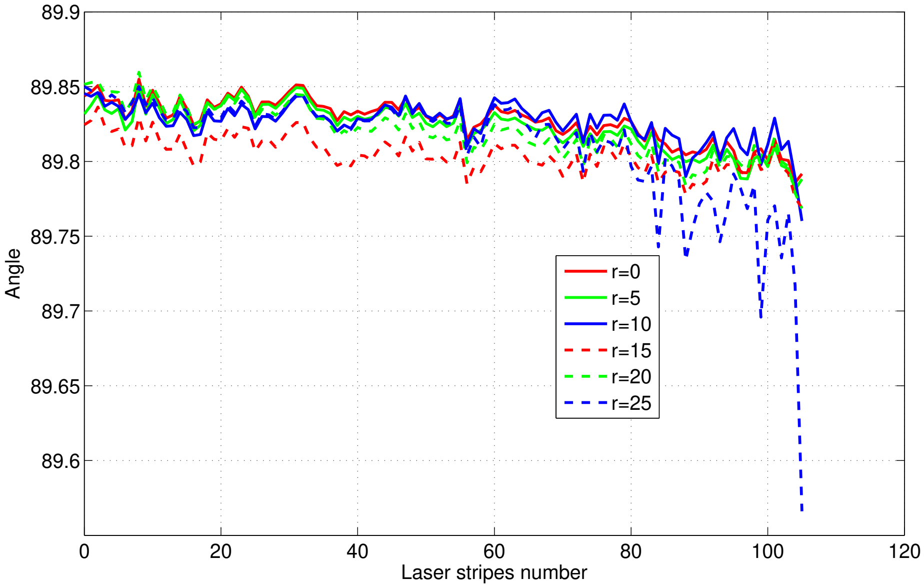


Figure 6.40: The average angle of laser stripes at each linear stage moving step based on the linear fitting after the image correction using FMC algorithm with various multiples of a basic noise. The image size is 720×576, calibration target contains $8 \times 6 = 48$ corners, every corner has a distance of $20mm$ from each other. The scanner travels 105 steps at $1mm$ interval. Total there are 105 images. Gaussian white noise with standard deviation $\delta = 0.04\tau$ was added to the coordinates of the 3D world points $p_{wi}(i = 1, 2, \cdots 48)$ and $\delta = 0.02\tau$ was added to image points P_{fi} where τ varied from 0 to 25 at intervals of 1.

Table 6.17: Absolute error Δ and standard deviation σ of laser peak 3D reconstruction based on Zhang algorithm with various multiples of a basic noise. The image size is 720×576, calibration target contains $8 \times 6 = 48$ corners, every corner has a distance of $20mm$ from each other. The scanner travels 105 steps at $1mm$ interval. Gaussian white noise with standard deviation $\delta = 0.04\tau$ was added to the coordinates of the 3D world points $p_{wi}(i = 1, 2, \cdots 48)$ and $\delta = 0.02\tau$ was added to image points P_{fi} where τ varied from 0 to 25 at intervals of 1. Ref: actual measurement, τ: multiple noise, μ estimated peak location. Unit:mm

	$\tau = 0$			$\tau = 5$			$\tau = 10$			$\tau = 15$			$\tau = 20$			$\tau = 25$		
Ref	μ	Δ	σ	μ	Δ	σ	μ	Δ	σ	μ	Δ	σ	μ	Δ	σ	μ	Δ	σ
148.00	148.82	0.82	0.08	148.87	0.87	0.08	148.81	0.81	0.07	149.19	1.19	0.07	149.27	1.27	0.07	149.57	1.57	0.06
158.00	158.71	0.71	0.09	158.76	0.76	0.09	158.70	0.70	0.09	159.08	1.08	0.09	159.17	1.17	0.09	159.46	1.46	0.08
168.00	168.71	0.71	0.11	168.76	0.76	0.11	168.70	0.70	0.11	169.08	1.08	0.11	169.17	1.17	0.11	169.46	1.46	0.11
178.00	178.71	0.71	0.12	178.76	0.76	0.12	178.70	0.70	0.12	179.09	1.09	0.12	179.17	1.17	0.13	179.47	1.47	0.13
188.00	188.69	0.69	0.14	188.74	0.74	0.14	188.68	0.68	0.14	189.06	1.06	0.15	189.15	1.15	0.15	189.44	1.44	0.16
198.00	198.65	0.65	0.15	198.70	0.70	0.15	198.64	0.64	0.15	199.01	1.01	0.16	199.10	1.10	0.16	199.39	1.39	0.18
208.00	208.62	0.62	0.14	208.67	0.67	0.14	208.61	0.61	0.14	208.98	0.98	0.15	209.08	1.08	0.15	209.35	1.35	0.17
218.00	218.57	0.57	0.15	218.62	0.62	0.15	218.57	0.57	0.16	218.94	0.94	0.17	219.04	1.04	0.16	219.29	1.29	0.19
228.00	228.64	0.64	0.14	228.68	0.68	0.14	228.63	0.63	0.14	229.00	1.00	0.15	229.12	1.12	0.14	229.34	1.34	0.17
238.00	238.67	0.67	0.16	238.71	0.71	0.16	238.67	0.67	0.17	239.04	1.04	0.18	239.17	1.17	0.17	239.37	1.37	0.21
248.00	248.69	0.69	0.16	248.73	0.73	0.16	248.70	0.70	0.16	249.06	1.06	0.17	249.22	1.22	0.16	249.38	1.38	0.19
253.00	253.71	0.71	0.16	253.75	0.75	0.16	253.72	0.72	0.16	254.08	1.08	0.17	254.26	1.26	0.16	254.40	1.40	0.18
Avg	-	0.68	-	-	0.73	-	-	0.68	-	-	1.05	-	-	1.16	-	-	1.41	-

Table 6.18: Absolute error Δ and standard deviation σ of laser peak 3D reconstruction based on Zhang-RD algorithm with various multiples of a basic noise. The image size is 720×576, calibration target contains $8 \times 6 = 48$ corners, every corner has a distance of $20mm$ from each other. The scanner travels 105 steps at $1mm$ interval. Gaussian white noise with standard deviation $\delta = 0.04\tau$ was added to the coordinates of the 3D world points $p_{wi}(i = 1, 2, \cdots 48)$ and $\delta = 0.02\tau$ was added to image points P_{fi} where τ varied from 0 to 25 at intervals of 1. Ref: actual measurement, τ: multiple noise, μ estimated peak location. Unit:mm

	$\tau = 0$			$\tau = 5$			$\tau = 10$			$\tau = 15$			$\tau = 20$			$\tau = 25$		
Ref	μ	Δ	σ	μ	Δ	σ	μ	Δ	σ	μ	Δ	σ	μ	Δ	σ	μ	Δ	σ
148.00	147.85	0.15	0.07	148.26	0.26	0.04	148.31	0.31	0.07	148.38	0.38	0.07	148.62	0.62	0.07	148.55	0.55	0.09
158.00	157.74	0.26	0.08	158.26	0.26	0.07	158.20	0.20	0.09	158.38	0.38	0.05	158.62	0.62	0.09	158.55	0.65	0.10
168.00	167.74	0.26	0.10	168.15	0.15	0.08	168.20	0.20	0.11	168.27	0.27	0.07	168.51	0.51	0.11	168.65	0.65	0.11
178.00	177.74	0.26	0.12	178.15	0.15	0.08	178.20	0.20	0.12	178.27	0.27	0.07	178.51	0.51	0.12	178.65	0.65	0.12
188.00	187.72	0.28	0.14	188.13	0.13	0.10	188.18	0.18	0.13	188.25	0.25	0.08	188.49	0.49	0.14	188.67	0.67	0.13
198.00	197.68	0.32	0.15	198.09	0.09	0.12	198.14	0.14	0.13	198.21	0.21	0.08	198.44	0.44	0.15	198.70	0.70	0.14
208.00	207.65	0.35	0.15	208.06	0.06	0.12	208.11	0.11	0.12	208.18	0.18	0.09	208.41	0.41	0.15	208.71	0.71	0.12
218.00	217.61	0.39	0.17	218.02	0.02	0.14	218.07	0.07	0.12	218.14	0.14	0.10	218.36	0.36	0.16	218.72	0.72	0.14
228.00	227.67	0.33	0.16	228.08	0.08	0.12	228.14	0.14	0.13	228.20	0.20	0.08	228.42	0.42	0.15	228.61	0.61	0.12
238.00	237.70	0.30	0.18	238.11	0.11	0.12	238.18	0.18	0.16	238.24	0.24	0.10	238.45	0.45	0.18	238.51	0.51	0.15
248.00	247.72	0.28	0.18	248.13	0.13	0.13	248.21	0.21	0.16	248.26	0.26	0.10	248.46	0.46	0.17	248.41	0.41	0.14
253.00	252.73	0.27	0.18	253.15	0.15	0.13	253.23	0.23	0.16	253.29	0.29	0.10	253.48	0.48	0.17	253.34	0.34	0.15
Avg	-	0.29	-	-	0.13	-	-	0.19	-	-	0.26	-	-	0.48	-	-	0.59	-

Table 6.19: Absolute error Δ and standard deviation σ of laser peak 3D reconstruction based on FMC algorithm with various multiples of a basic noise. The image size is 720×576, calibration target contains $8 \times 6 = 48$ corners, every corner has a distance of $20mm$ from each other. The scanner travels 105 steps at $1mm$ interval. Gaussian white noise with standard deviation $\delta = 0.04\tau$ was added to the coordinates of the 3D world points $p_{wi}(i = 1, 2, \cdots 48)$ and $\delta = 0.02\tau$ was added to image points P_{fi} where τ varied from 0 to 25 at intervals of 1. Ref: actual measurement, τ: multiple noise, μ estimated peak location. Unit:mm

	$\tau = 0$			$\tau = 5$			$\tau = 10$			$\tau = 15$			$\tau = 20$			$\tau = 25$		
Ref	μ	Δ	σ	μ	Δ	σ	μ	Δ	σ	μ	Δ	σ	μ	Δ	σ	μ	Δ	σ
148.00	148.00	0.00	0.06	148.28	0.28	0.06	148.09	0.09	0.08	148.42	0.42	0.07	148.21	0.21	0.06	148.54	0.54	0.08
158.00	158.11	0.11	0.08	158.18	0.18	0.05	158.20	0.20	0.09	158.31	0.31	0.09	158.31	0.31	0.06	158.43	0.43	0.09
168.00	168.11	0.11	0.10	168.18	0.18	0.07	168.20	0.20	0.11	168.31	0.31	0.11	168.31	0.31	0.08	168.43	0.43	0.11
178.00	178.11	0.11	0.11	178.18	0.18	0.07	178.20	0.20	0.12	178.31	0.31	0.13	178.31	0.31	0.09	178.43	0.43	0.12
188.00	188.13	0.13	0.14	188.16	0.16	0.08	188.22	0.22	0.14	188.29	0.29	0.16	188.33	0.33	0.10	188.41	0.41	0.14
198.00	198.18	0.18	0.15	198.12	0.12	0.09	198.26	0.26	0.15	198.24	0.24	0.18	198.37	0.37	0.10	198.36	0.36	0.15
208.00	208.21	0.21	0.15	208.09	0.09	0.10	208.30	0.30	0.14	208.20	0.20	0.17	208.40	0.40	0.10	208.33	0.33	0.14
218.00	218.26	0.26	0.18	218.06	0.06	0.11	218.34	0.34	0.16	218.16	0.16	0.19	218.43	0.43	0.10	218.29	0.29	0.15
228.00	228.20	0.20	0.16	228.13	0.13	0.11	228.29	0.29	0.14	228.21	0.21	0.17	228.35	0.35	0.11	228.36	0.36	0.13
238.00	238.18	0.18	0.18	238.19	0.19	0.12	238.26	0.26	0.17	238.25	0.25	0.21	238.29	0.29	0.15	238.39	0.39	0.16
248.00	248.17	0.17	0.18	248.23	0.23	0.13	248.26	0.26	0.16	248.26	0.26	0.20	248.24	0.24	0.16	248.42	0.42	0.14
253.00	253.17	0.17	0.18	253.26	0.26	0.13	253.25	0.25	0.16	253.28	0.28	0.19	253.21	0.21	0.15	253.44	0.44	0.14
Avg	-	0.15	-	-	0.17	-	-	0.24	-	-	0.27	-	-	0.31	-	-	0.40	-

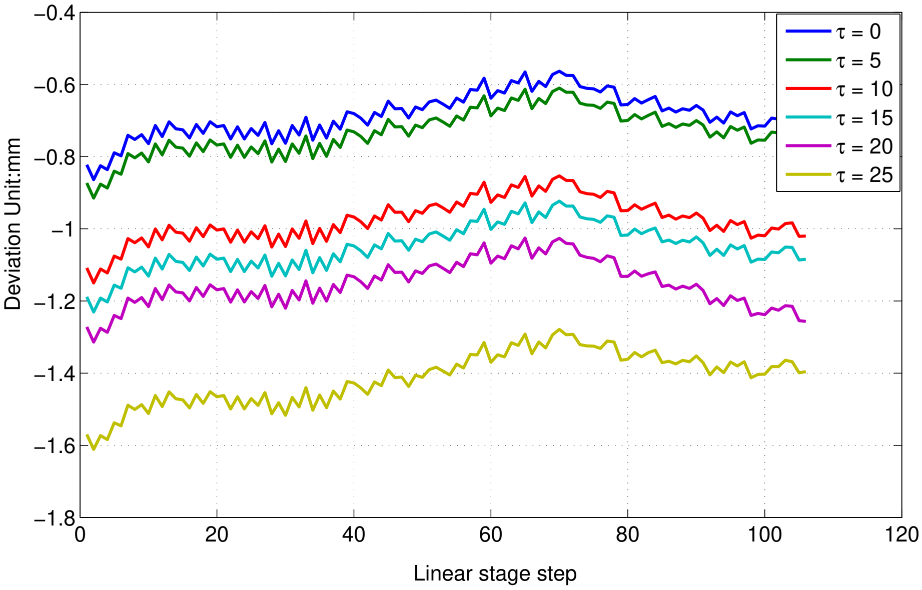


Figure 6.41: The laser stripes 3D reconstruction deviation using Zhang algorithms with various multiples of a basic noise. The scanner travels 105 steps at $1mm$ interval. Gaussian white noise with standard deviation $\delta = 0.04\tau$ was added to the coordinates of the 3D world points $p_{wi}(i = 1, 2, \cdots 48)$ and $\delta = 0.02\tau$ was added to image points P_{fi} where τ varied from 0 to 25 at intervals of 1.

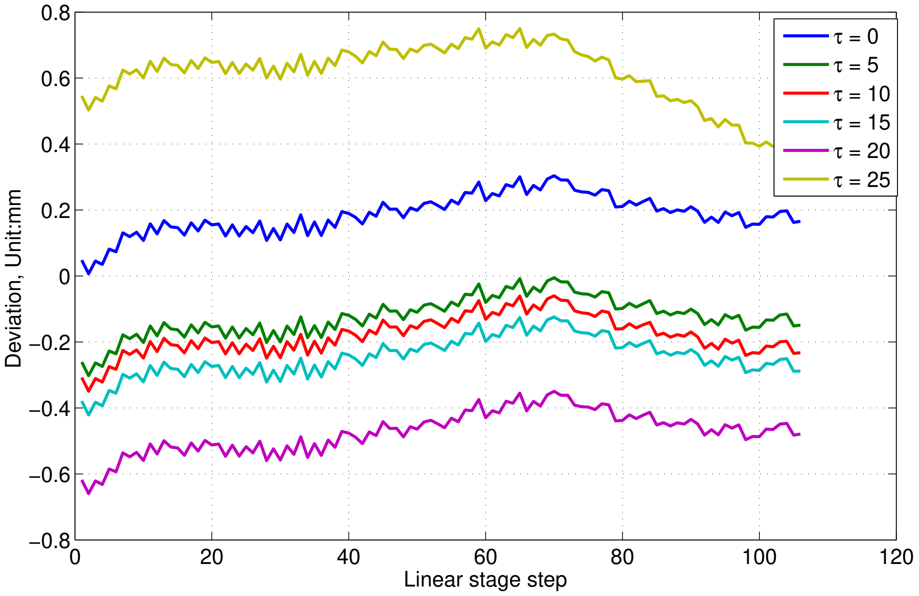


Figure 6.42: The laser stripes 3D reconstruction deviation based on Zhang-RD algorithm with various multiples of a basic noise. The scanner travels 105 steps at $1mm$ interval. Gaussian white noise with standard deviation $\delta = 0.04\tau$ was added to the coordinates of the 3D world points $p_{wi}(i = 1, 2, \cdots 48)$ and $\delta = 0.02\tau$ was added to image points P_{fi} where τ varied from 0 to 25 at intervals of 1.

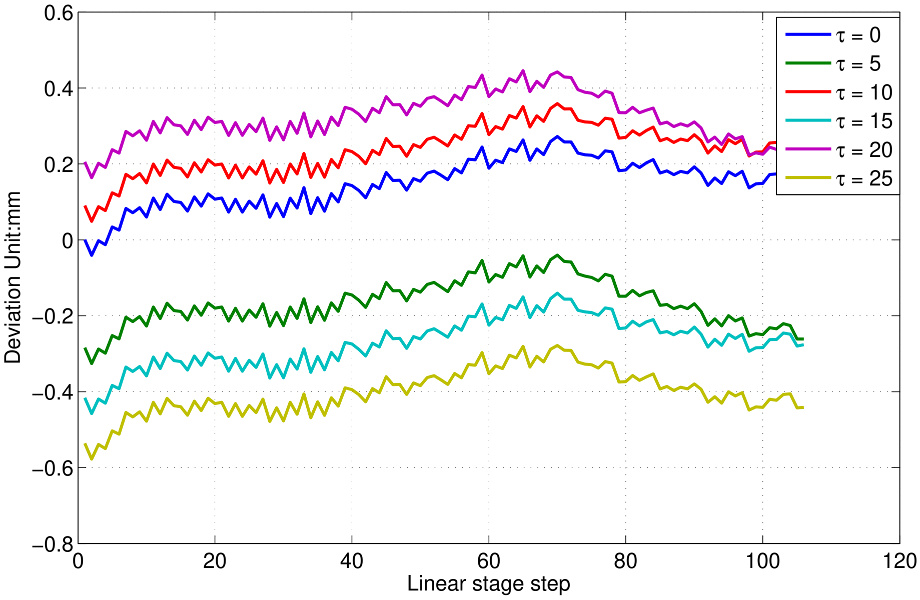


Figure 6.43: The laser stripes 3D reconstruction deviation using FMC algorithms with various multiples of a basic noise. The scanner travels 105 steps at $1mm$ interval. Gaussian white noise with standard deviation $\delta = 0.04\tau$ was added to the coordinates of the 3D world points $p_{wi}(i = 1, 2, \cdots 48)$ and $\delta = 0.02\tau$ was added to image points P_{fi} where τ varied from 0 to 25 at intervals of 1.

reconstructed to 3D. The initial distance D between camera CCD lens to the vertical board is measurable and set to be $148.0mm$. The linear stage was controlled to move backward from the board in $1mm$ step. After 105 steps, the distance D was increased to $253mm$. The absolute error Δ between the actual value (REF) and estimation (μ) and standard deviation (σ) at each noise levels are shown in Tables 6.17, 6.18, 6.19. The differences between 'REF' and μ at different noise levels at each linear stage step are shown on Figures 6.41, 6.42, and 6.43. It can be observed from these tables and figures that:

1. Zhang algorithm has the biggest deviation and absolute error in all noise levels. In the noise free dataset, its absolute error is up to $0.82mm$. When $\tau = 25$, it increases to $1.57mm$. In Fig.6.41, the deviation is between $-1.5 \sim 0.5mm$, none of them is close to range $\pm 0.5mm$.

2. Zhang-RD has much smaller error in considering the decentring distortion. Its average absolute error is only $0.29mm$ over the noise free data, the derivation is between $-0.4 \sim 0.8mm$ in Fig.6.42. It can be presumed that this lens has decentring distortion and Zhang-RD making full use of this prior knowledge which lens distortion model subject to.

3. FMC has better result without prior knowledge of the camera distortion models. It can be seen from Table.6.19, FMC has the smallest absolute error in average. In the noise free data, its maximum absolute error is only $0.26mm$ in average. Even in the high noise level (like $\tau > 15$), its maximum absolute error is only $0.54mm$. FMC's deviation in Fig.6.43 is between $-0.5 \sim 0.5mm$ which is 66.7% smaller than Zhang and 33.3% smaller than Zhang-RD. This shows that the FMC model is powerful for the characterization and correction of the camera distortions.

4. The 3D reconstruction requires accurate calibration parameters and lens distortion factors. In our case, the linear stage parameters estimation also requires the same information because these scanner parameters are hard to measure them directly. Therefor, the final 3D reconstruction result amplified the error coming from incorrect calibration parameters and distortion factors. Such inaccurate calibration result and distortion factor causes error in the scanner parameters including baseline 'D' and laser shutting angle α estimation. Eventually, all these errors affects the triangulation based laser scanner 3D reconstruction accuracy which is consistent with the experimental result showing above. It is completely feasible for FMC to calibrate and correct the cameras without knowing the distortions that they are subject to.

6.4.3 Conclusion

In this section, we evaluated an algorithm for camera calibration and correction. This algorithm is based on a novel camera distortion model which attempts to model the overall distortion of the camera without knowing which distortion of the lens subject to and combat the imaging noise for more accurate camera calibration and correction results. To estimate the parameters of interest, we globally optimized an objective function about the sum of the back projection errors using the LM algorithm. For the initialization of the LM algorithm, the classical Zhang algorithm [83] was implemented. After both the intrinsic and distortion parameters have been calibrated, the distorted image points have been corrected using again the LM algorithm initialized by the distorted image points themselves. This LM algorithm minimizes the squared difference between the distorted points and the transformed corrected points. As long as both the intrinsic and distortion parameters have been calibrated with reasonable accuracy, then the minimization is usually successful.

We experimentally demonstrated that the prior knowledge about the camera distortion does not necessarily always bring benefits to algorithms for accurate camera calibration and correction. In reality, we probably have little idea about what distortion the captured image is subject to. The actual distortion of the camera is not known and need to be investigated. Camera distortion model is critical for camera calibration and correction. What the accurate camera calibration and correction algorithm requires is a powerful model to characterise the overall camera distortion and effectively resist the imaging noise in the data. A comparative study based on both real images from Zhang and the 3D scanner captured ones corrupted by reasonable levels of noise shows that the proposed algorithm successfully calibrated and corrected the distorted image points. Fractional model is more powerful than the polynomial model in describing the distortions that the camera was subject to.

In additional, we compared 3D reconstruction result based on the calibration parameter with our 3D image data. It is shown that FMC does have better performance compared to Zhang and Zhang-RD. The experimental results show that the proposed algorithm successfully characterises the camera overall distortion, producing encouraging results for camera calibration and correction and giving us the smallest deviation between the estimation value and actual 3D measurement.

6.5 Iterative Camera Calibration

In this section, we evaluate the proposed iterative camera calibration method described in section.4.5 using synthetic and real images. The chessboard was used as the calibration patterns in this chapter. The corners were detected by the same method to make sure that the results are comparable. The Zhang-RD implemented in OpenCV [7] and Andrea [1] method were selected to compare with the proposed method. Please note Andrea method only considered the 3D input points as unknown variables while the proposed method consider the 3D and the detected 2D feature points altogether.

6.5.1 Synthetic Data

To have a prior knowledge about the camera calibration result, synthetic data was generated to evaluate proposed iterative camera calibration method. The closer the calibration result to the prior setting the better.

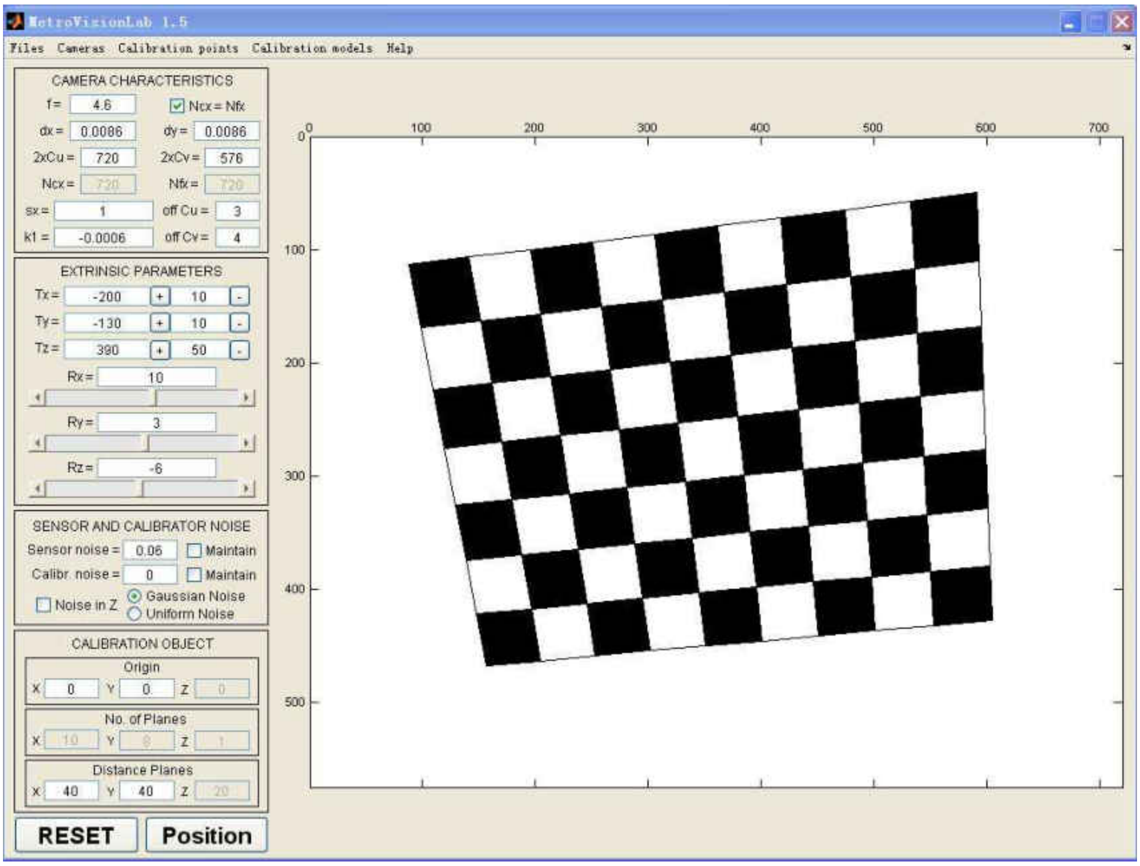
6.5.1.1 Experimental setup

METROVISIONLAB package [69] was used in this section to generate synthetic chessboard pattern images. Fig.6.44 shows its user interface. It is easy to control all the calibration parameters, chessboard size, synthetic noise and generate synthetic image.

In order to test how image resolution, focal length, and image noise affect the camera calibration and thus the accuracy of 3D reconstruction, two sets of synthetic data $(\mathbf{A}, \mathbf{B})$ were generated using the following parameters:

Here are the settings for the synthetic data **A**: Image size is 720×576, $C_x = 360.0$, $C_y = 288.0$, focal length $f = 4.6mm$, $d_x = 0.0086$, $d_y = 0.0086$ and offset of principal point is $C_u = 3.0$, $C_v = 4.0$. Camera sensor and the calibration gauge object noise are generated by random Gaussian noise, 0.02 and 0.01 respectively. The chessboard contains 8×6 grids, each grid size is $40 \times 40mm$. We generated five planes to simulate five images captured from the real chessboard. Each camera capturing position's rotation and translation vectors settings is shown in Table 6.20. The rotation and translation vectors unit are degree and mm respectively.

For the synthetic data **B**: Image size is 1024×800, $C_x = 512$, $C_y = 400$, focal length $f = 8mm$, $d_x = 0.0084$, $d_y = 0.0086$. Camera sensor and the calibration gauge object noise are generated by random Gaussian noise, 0.04 and 0.02 respectively which are higher than **A**. Offset for the coordinates of principal point is $(4.0, 5.0)$. The chessboard grids number and size are the same as setting **A**. Five planes was generated to simulate five images captured from the real chessboard. Each camera capturing position's rotation and translation vectors settings is shown in Table 6.22. Focal length (f_x, f_y) unit conversion from millimeter to pixel are $f_x(pixel) = f_x(mm)/d_x$ and $f_y(pixel) = f_y(mm)/d_y$ in both settings.



CAMERA CHARACTERISTICS

f = 4.6 ☑ Ncx = Nfx

dx = 0.0086 dy = 0.0086

2xCu = 720 2xCv = 576

Ncx = 720 Nfx = 720

sx = 1 off Cu = 3

k1 = -0.0006 off Cv = 4

EXTRINSIC PARAMETERS

Tx = -200 + 10 -

Ty = -130 + 10 -

Tz = 390 + 50 -

Rx = 10

Ry = 3

Rz = -6

SENSOR AND CALIBRATOR NOISE

Sensor noise = 0.06 ☐ Maintain

Calibr. noise = 0 ☐ Maintain

☐ Noise in Z ● Gaussian Noise ○ Uniform Noise

CALIBRATION OBJECT

Origin X 0 Y 0 Z 0

No. of Planes X 10 Y 8 Z 1

Distance Planes X 40 Y 40 Z 20

Figure 6.44: Synthetic chessboard generator by the MetrovisionLab package

Table 6.20: Synthetic camera orientation and position in setting (**A**)

♯	R_x	R_y	R_z	T_x	T_y	T_z
0	8.00	0.00	0.00	-160.00	-150.00	430.00
1	11.00	6.00	-1.00	-190.00	-120.00	380.00
2	14.00	-14.00	-14.00	-150.00	-120.00	330.00
3	-7.00	-11.00	-11.00	-170.00	-170.00	380.00
4	10.00	3.00	-6.00	-200.00	-130.00	390.00

Table 6.21: Calibration results of different methods from test (**A**). Image size is 720 × 576, $C_x = 360.0$, $C_y = 288.0$, focal length $f = 4.6mm$, $d_x = 0.0086$, $d_y = 0.0086$ and offset of principal point is $C_u = 3.0$, $C_v = 4.0$. Camera sensor and the calibration gauge object noise are generated by random Gaussian noise, 0.02 and 0.01 respectively. The chessboard contains 8 × 6 grids, each grid size is $40 \times 40mm$. Est: estimated value, σ: absolute error between estimated value and reference. RPE: re-projection error.

		Zhang-RD		Andrea		Proposed	
Para.	Ref.	Est.	σ	Est.	σ	Est.	σ
f_x	534.88	535.98	1.10	534.93	0.05	534.33	0.55
f_y	534.88	536.64	1.76	535.93	1.05	535.27	0.39
C_x	363.00	360.96	2.04	359.92	3.08	360.06	2.94
C_y	292.00	291.69	0.31	294.52	2.83	294.26	2.26
RPE	-	0.211	-	0.117	-	0.111	-
SSE	-	4.431	-	3.016	-	2.955	-

Table 6.22: Synthetic camera orientation and position in setting (**B**)

♯	R_x	R_y	R_z	T_x	T_y	T_z
0	20.00	-14.00	1.00	-180.00	-120.00	400.00
1	26.00	0.00	-1.00	-170.00	-130.00	430.00
2	-15.00	4.00	8.00	-180.00	-190.00	610.00
3	-1.00	-17.00	0.00	-180.00	-140.00	490.00
4	-1.00	-14.00	13.00	-120.00	-150.00	460.00

Table 6.23: Calibration results of different methods from test (**B**). Image size is 1024 × 800, $C_x = 512$, $C_y = 400$, focal length $f = 8mm$, $d_x = 0.0084$, $d_y = 0.0086$. Camera sensor and the calibration gauge object noise are generated by random Gaussian noise, 0.04 and 0.02 respectively which are higher than **A**. Offset for the coordinates of principal point is $(4.0, 5.0)$. The chessboard contains 8 × 6 grids, each grid size is $40 \times 40mm$. Est: estimated value, σ: absolute error between estimated value and reference. RPE: re-projection error.

		Zhang-RD		Andrea		Proposed	
Para.	Ref.	Est.	σ	Est.	σ	Est.	σ
f_x	952.38	951.35	1.02	951.36	1.02	952.92	0.54
f_y	930.23	928.83	1.40	928.70	1.53	930.09	0.13
C_x	516.00	514.79	1.21	511.51	4.49	515.62	0.38
C_y	405.00	403.49	1.51	405.39	1.51	407.93	2.93
RPE	-	0.224	-	0.121	-	0.114	-
SSE	-	8.585	-	5.723	-	5.587	-

6.5.1.2 Results and analysis

Tables 6.21 and 6.23 show the calibration results and absolute error between calculated result and reference by the three methods. The last two rows are re-projection errors (RPE) and the sum of squared error (SSE [8]). From above tables it can be seen that:

1. The proposed method performs better than the others even in the heavier noise setting **B**. The proposed method calibration has more precise parameters f_x, f_y, C_x, and C_y.
2. In both setting, the proposed and Andrea RPE error is smaller than Zhang-RD's method. The estimated focal length by proposed method is only about ±0.5 pixel off the ground truth value. This shows that the iteration calibration is really useful for higher accurate camera calibration.
3. It is well known that the calibration accuracy depends on the 2D(detected corners) and 3D target(calibration target' 3D coordinates) data. According to [1] and [47], the calibration target' 3D coordinates include systematic error which generated during the manufacture process and it can be compensated by iterative method. In each iteration, once plausible camera parameters are obtained, we assume them to be correct and evaluate a more accurate target geometry by using a bundle adjustment technique to estimate only camera poses and scene, and then new 3D target data is generated. Since the estimated target is scaled towards the theoretical one at each step during bundle adjustment, the final camera calibration could be subject to an absolute scaling error that is not avoidable as the real measures of the target are not known. Still, this error is averaged over the printing error of each corner and has shown to be very small in our experience. The proposed method takes a further step in considering image distortion effect on corner detection. In each iteration, new corners will be detected again based on the images corrected by the current distortion factors. It makes sense the corner detection should perform better on distortion free images. Hence, the proposed method has about $3 \sim 10\%$ smaller RPE and SSE error than the others.

6.5.2 Real chessboard pattern images

As you may have noticed in Fig.6.45, there are some indentations on the chessboard grid. They are caused by the 2D/3D rendering by the PC graphic card and hard to avoid when the chessboard images are generated. Consequently, such indentations may introduce error to the camera calibration. To eliminate the possible synthetic error from chessboard image generation and have better knowledge about the performance of the proposed method in real applications, the scanner's chessboard pattern images are used in this section.

---

[8] SSE: the error in distorted image coordinates between the measured location of a feature point in the image plane and its re-projected point through the calibrated model.

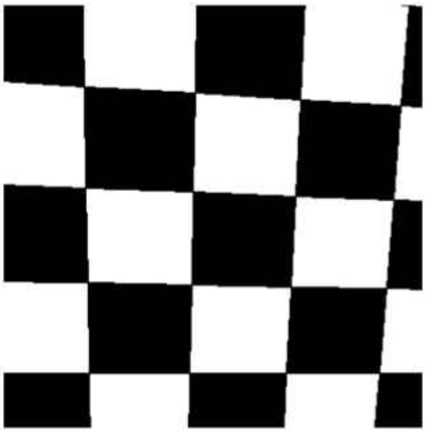


Figure 6.45: MetrovisionLab synthetic chessboard sample

The section is divided into two parts: (1) Firstly, the real chessboard images from underwater scanning were used to test the Zhang-RD, Andrea and the proposed algorithm's performance. (2) Secondly, artificial noise was added to the images or 3D input points respectively to simulate the difficult tasks in real environments.

6.5.2.1 Noise free images

This experiment used chessboard pattern images without artificial noise added to evaluate the proposed method's performance.

6.5.2.1.1 Experimental setup The camera has 1/3 inch CCD, image resolution is 720×576, principal point $C_x = 360, C_y = 288$ and its offset are unknown. The camera's focal length $f = 6.0mm$, $d_x = 0.0086$, $d_y = 0.0083$. Chessboard grid size is 12×8, each grid is $20mm$ by $20mm$. Eight pictures were taken in different positions. The chessboard was printed on a normal A4 paper and laminated by two thin layer plastic. In this way, we can assume the input 3D targets (chessboard pattern) are planar. In reality, even after the lamination, the surface of the pattern cannot be guaranteed 100% flat and corner to corner distance may not always be $20mm$ due to the mechanical problems form printer or rugosity of the paper itself. This is also the main reason to introduce the proposed method to reduce the side effect caused by low cost handmade chessboard calibration pattern.

6.5.2.1.2 Results and analysis The calibration results by Zhang-RD, Andrea and proposed methods are shown in the Table.6.24. It can be seen from the table that:

1. Camera focal length (f_x, f_y) shown in the table are in pixel and they can be converted to millimeter unit. In Zhang-RD is $f_x = 672.132, f_y = 728.37(pixels)$

Table 6.24: Calibration results of different methods using real data. The camera has 1/3 inch CCD, image resolution is 720 × 576, principal point $C_x = 360, C_y = 288$ and its offset are unknown. The camera's focal length $f = 6.0mm$, $d_x = 0.0086$, $d_y = 0.0083$. Chessboard grid size is 12 × 8, each grid is $20mm$ by $20mm$

	f_x	f_y	C_x	C_y	RPE	SSE
Zhang-RD	672.132	728.370	367.491	311.850	0.170	2.765
Andrea	679.449	738.040	362.702	310.412	0.092	0.811
Proposed	679.489	738.001	363.191	311.112	0.091	0.798

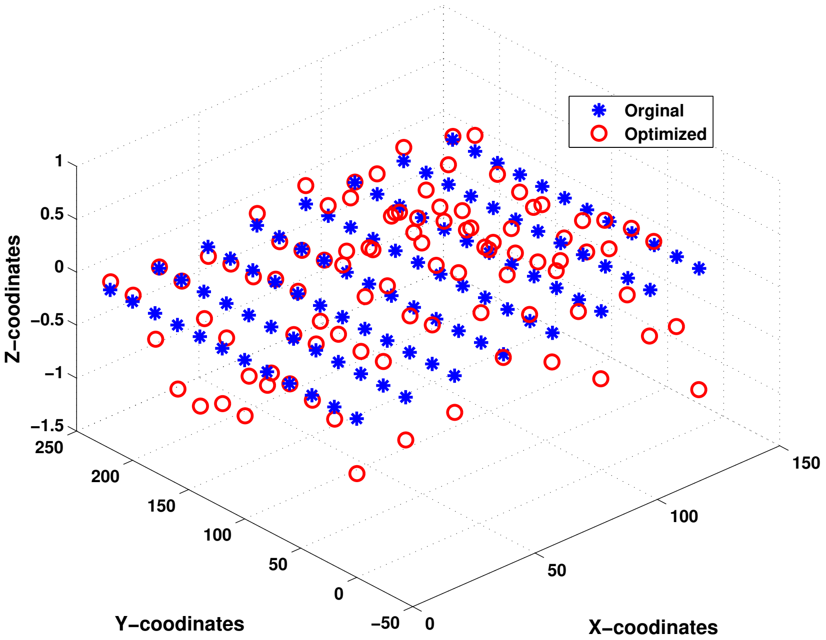


Figure 6.46: Refined world points in 3D view, Red Circles: Optimized 3D world points, Blue Stars: The original points

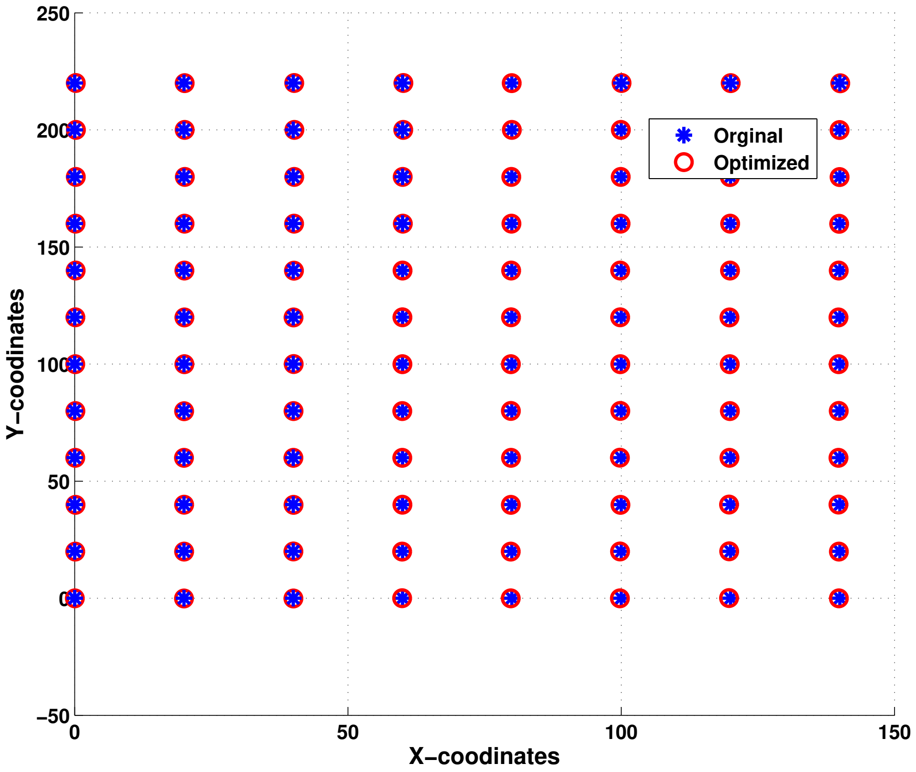


Figure 6.47: Refined world points in 2D view, Red Circles: Optimized 3D world points, Blue Stars: The original points

and they converted to millimeter unit is $672.132 \times 0.0084 = 5.780(mm), 728.37 \times 0.0083 = 6.045(mm)$, their average is $5.913mm$. Andrea has slightly better result: $f = 5.918mm$. In contrast, the proposed method and calibrated the same parameter as $f = 5.985mm$, which is much closer to the factory focal length value: $6.0mm$.

2. The proposed method reduced re-projection error(RPE) about 43% than Zhang-RD. In addition, its RPE and SSE error apparently smaller than Zhang-RD and Andrea. As mentioned in above sections, the printed chessboard is laminated and placed on a flat board, normally, we can set the initial Z-coordinate in 3D world pattern to zero. However, the ground truth is not always as explained above, since the paper itself and lamination may cause error in Z axis. Andrea and proposed method are trying to correct or compensate such inaccurate 3D world points by iterative process. Fig.6.46 shows the 3D world points after the iterative refinement. It can be seen that the maximum error in Z-coordinate is about $1mm$, which is quite close to ground truth because such home-made chessboard's surface is hard to guaranty 100% flat and even. The thickness of the paper is no more than $1mm$ which measured by caliper. Fig.6.47 shows the flat view in XY-plane, most of the points remain the same position with maximum $\pm 0.3mm$ shifts. Similar experiments will be performed again in later section.6.5.3 to verify these calibration parameters resulting in the 3D reconstruction.

6.5.2.2 Artificial noise added

In this section, artificial noise was added to simulate the difficult tasks in real environments.

6.5.2.2.1 Experimental setup This experiment can be divided into two parts:

Firstly, we used the same captured images but added different levels of Gaussian white noise from $\sigma = 0.1 \sim 0.5$. Fig.6.48 shows one of the chessboard pattern images with different levels of noise. Clearly, when the noise increases to $\sigma = 0.5$, it is a big challenge to detect accurate corners.

Secondly, due to fact that the manufacturing process of the 3D pattern always introduce errors to the calibration pattern. This part investigates how such inaccurate 3D pattern affects the camera calibration and correction. We assumed the chessboard pattern is accurate in the image plane, and then the random noise was added to the 3D world points to simulate the error coming from manufacturing or any other source. Then the performance of each calibration method can be tested as the input 3D world points are inaccurate.

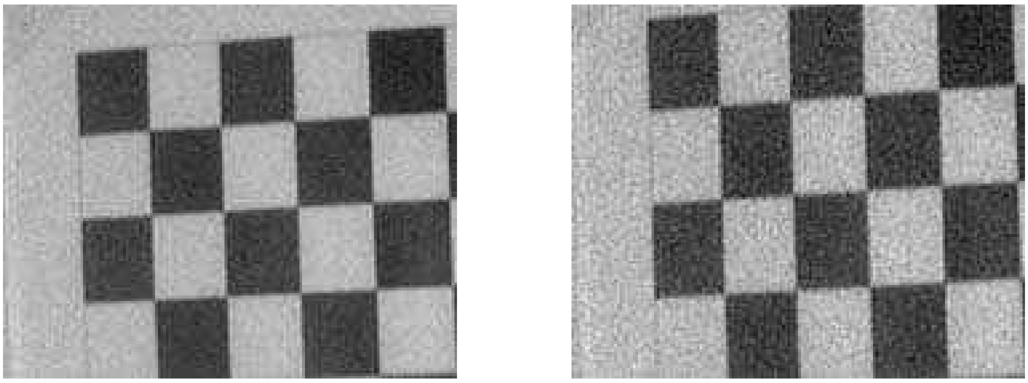
The chessboard grid size is $20mm$ and the added noise is varied from $1\% \sim 10\%$ in absolute distance (in $X - Y$ coordinates), which causing maximum $2.0mm$ off to the original position. In both experiments, each noise level will run 100 times and the average will be the final result.

6.5.2.2.2 Results and analysis: corrupted images The example noisy images are shown in Fig.6.48, and the calibration results are shown in Table 6.25. Individual calibration parameter comparison result chart is shown in figures from 6.49 to 6.52. To have a more objective evaluation and eliminate the effect of the corner detection method, all the methods used the same detection and noise removal method.

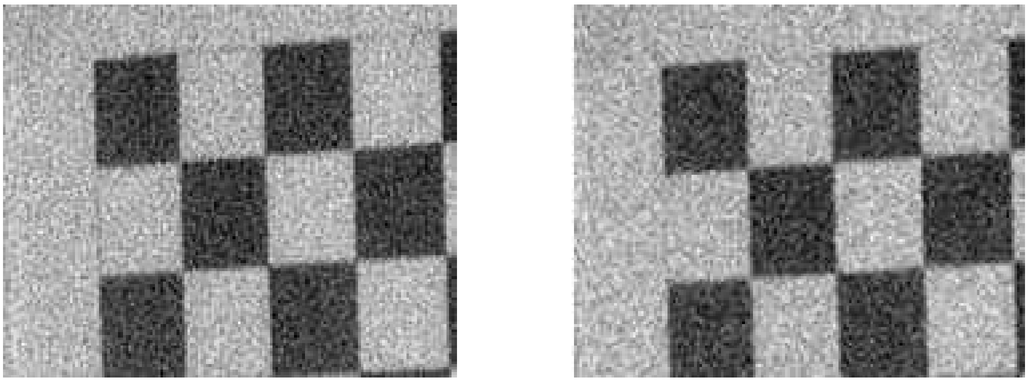
As shown in these figures and tables: (1) in the low level noise $(0 \sim 0.2)$, all the results are fairly similar. The calibration parameters in Zhang-RD, Andrea, and the proposed method have a maximum difference of $2.2, 2.0$ and 1.14 pixels. When the noise level increases to 0.5, the results are not stable any more, particularly C_x and C_y have dramatic changes. The RPE error also increases as the noise level increases. However, Andrea and proposed method provide smaller re-projection errors than Zhang-RD. (2) θ in Table 6.25 is the standard deviation for all the parameters from noise $(0 \sim 0.5)$. It can be seen STD increase as the noise level increase. Zhang-RD has the smallest STD for all calibration parameters but has the largest STD for RPE which indicates its calibration result is not stable and reliable in the high noise levels (like $\sigma > 0.2$). (3) The noise do has heavy impact on the corner detection and brought error into the detected corners, therefore, in the Eq.4.3.8, the Levenberg-Marquardt algorithm takes much more time to converge with these inaccurate corners. (4) It can be observed that as the noise levels increase, the maximum error of calibration parameters is about ten pixels, which is less than 3% deviation. However, RPE error increases over 200%. It is because re-projected points on image plane are hard to match the noisy 2D corners. Obviously, in the heavy noise corrupted images, corner detection method plays a much more important role during the entire calibration process. Thus, in the next experiment, we introduce different noises into 3D world points to evaluate the performance of these calibration methods.

6.5.2.2.3 Results and analysis: corrupted 3D points In this experiment, reasonable artificial noise is added into the 3D world points to simulate real environment and test the proposed method performance in such difficult task. The chessboard has $12 \times 8 = 96$ corners. Distance between each corner is designed to be $20mm$. The introduced different noise levels are at $0.2, 0.6, 1.0, 1.4, 2.0$ (Unit:mm) in $X - Y$ coordinates, which is up to $2.0mm$ (10%) off from the original points in XY plane in our case. Fig.6.53 shows the corrupted 3D world points with different noise levels on the XY plane. Since the noisy data was generated randomly, thus, the calibration needs to run 100 times for each noise level and the average will be used as the final result.

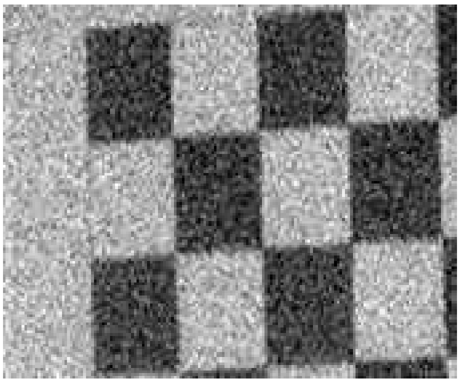
Table 6.26 and Figures from 6.56 to 6.55 clearly show when the added noise increases in 3D space, the most important calibration parameters by Zhang-RD method can not remain stable, while the results by Andrea and proposed methods are more consistent. For instance, when the noise level increases from 0% to 5% which has up to $1mm$ off (Chessboard grid size is 20mm by 20mm) in 3D space, the maximum deviation of f_x is 4.179 pixels by Zhang-RD, 1.022 by Andrea and 0.967 pixel by the proposed method which is less than a single pixel. Even the noise level increases to 10%, which has 2mm off from the original position in 3D space, the proposed method maintains the value well,



(a) $\sigma = 0.1$ (b) $\sigma = 0.2$



(c) $\sigma = 0.3$ (d) $\sigma = 0.4$



(e) $\sigma = 0.5$

Figure 6.48: Artificial noise added to the chessboard images

Table 6.25: Calibration results of different methods from the corrupted image data. Different levels of Gaussian white noise from $\sigma = 0.1 \sim 0.5$ was added to the same captured images. Image size is 720×576, principal point $C_x = 360, C_y = 288$ and its offset are unknown. The camera's focal length $f = 6.0mm$, $d_x = 0.0086$, $d_y = 0.0083$. Chessboard grid size is 12×8, each grid is $20mm$ by $20mm$. μ, θ: average and standard deviation of each calibration parameter.

		f_x		f_y		C_x		C_y		RPE	
Noise Level		μ	θ	μ	θ	μ	θ	μ	θ	μ	θ
$\sigma = 0.0$	Zhang-RD	672.132	0.000	728.370	0.000	367.491	0.000	311.850	0.000	0.170	0.000
	Andrea	679.449	0.000	738.040	0.000	362.702	0.000	310.412	0.000	0.092	0.000
	Proposed	679.489	0.000	738.001	0.000	363.191	0.000	311.112	0.000	0.091	0.000
$\sigma = 0.1$	Zhang-RD	671.499	1.308	727.682	1.396	367.103	0.668	311.937	0.539	0.209	0.040
	Andrea	679.686	2.072	738.368	2.223	362.879	0.961	309.492	0.952	0.141	0.017
	Proposed	677.880	1.661	736.296	1.793	363.728	1.661	310.023	1.799	0.096	0.008
$\sigma = 0.2$	Zhang-RD	673.298	2.605	729.512	2.810	369.327	1.110	311.794	1.221	0.317	0.037
	Andrea	679.878	3.177	738.454	3.388	364.805	2.761	309.312	1.523	0.259	0.031
	Proposed	678.945	2.303	737.437	2.395	364.325	3.970	310.506	4.013	0.142	0.015
$\sigma = 0.3$	Zhang-RD	675.664	2.750	732.401	3.037	365.108	2.785	313.940	1.769	0.453	0.062
	Andrea	681.369	6.349	740.366	6.809	362.323	3.517	312.664	2.858	0.384	0.039
	Proposed	680.480	3.817	739.298	4.085	362.125	6.001	313.860	5.200	0.195	0.030
$\sigma = 0.4$	Zhang-RD	674.682	4.225	731.183	4.475	368.679	2.542	313.528	2.158	0.581	0.104
	Andrea	687.645	8.044	746.488	8.612	360.508	4.564	309.931	3.467	0.507	0.048
	Proposed	684.589	5.388	741.744	5.660	366.425	7.698	314.553	7.140	0.250	0.045
$\sigma = 0.5$	Zhang-RD	674.774	4.432	731.289	4.866	366.742	3.413	311.296	3.501	0.716	0.118
	Andrea	681.094	9.506	740.356	10.172	357.637	6.440	308.252	4.898	0.630	0.069
	Proposed	681.583	5.284	741.533	5.874	356.534	7.484	307.094	7.572	0.314	0.055

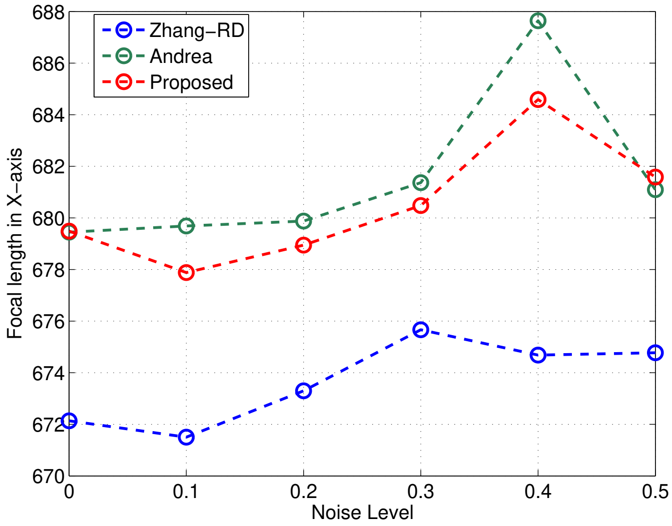


Figure 6.49: Calibrated parameter f_x of different methods from images corrupted by artificial noise. Image size is 720×576, principal point $C_x = 360, C_y = 288$ and its offset are unknown. Different levels of Gaussian white noise from $\sigma = 0.1 \sim 0.5$ was added to the same captured images.

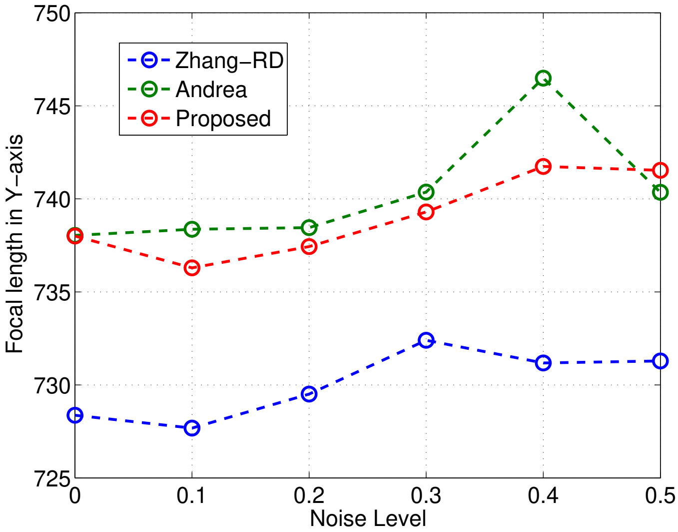


Figure 6.50: Calibrated parameter f_y of different methods from images corrupted by artificial noise. Image size is 720×576, principal point $C_x = 360, C_y = 288$ and its offset are unknown. Different levels of Gaussian white noise from $\sigma = 0.1 \sim 0.5$ was added to the same captured images.

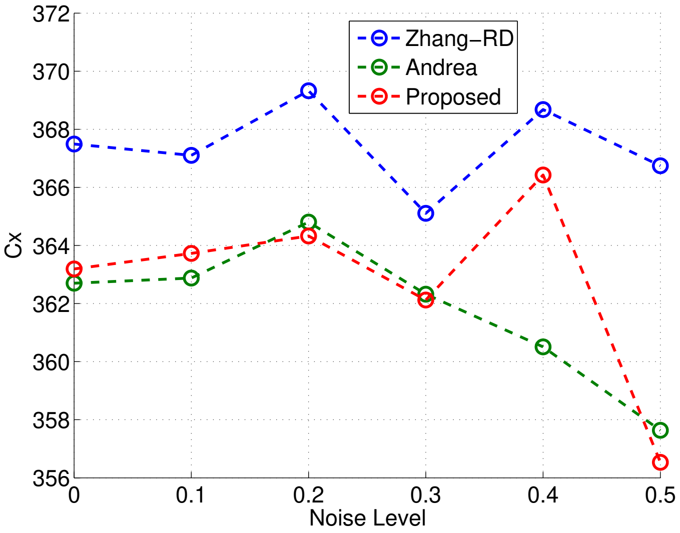


Figure 6.51: Calibrated parameter C_x of different methods from images corrupted by artificial noise. Image size is 720×576, principal point $C_x = 360, C_y = 288$ and its offset are unknown. Different levels of Gaussian white noise from $\sigma = 0.1 \sim 0.5$ was added to the same captured images.

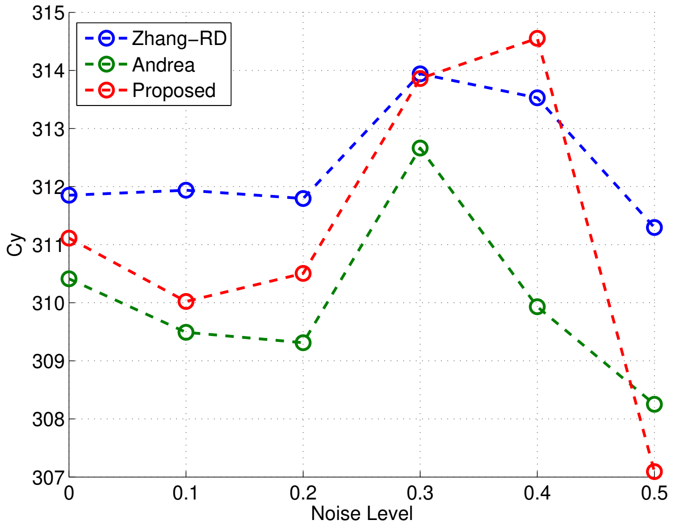
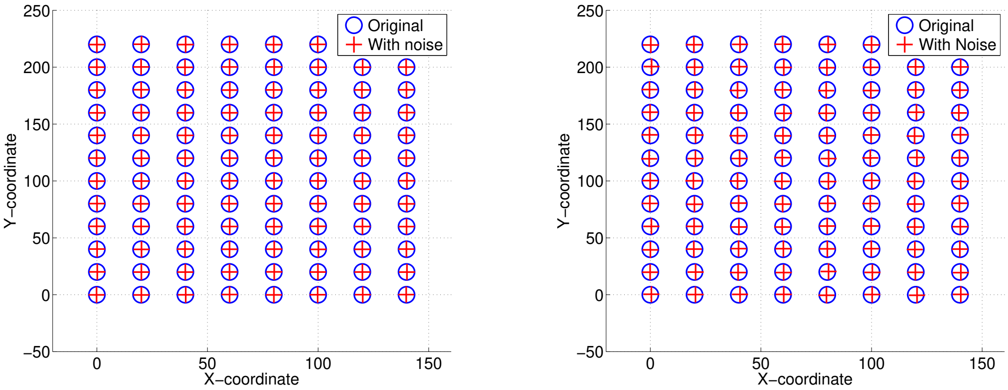
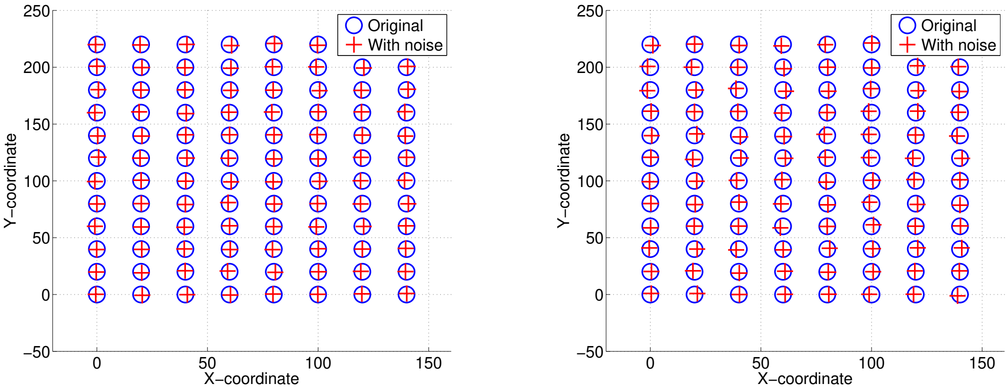


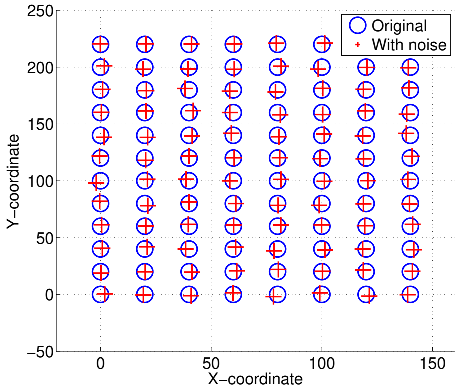
Figure 6.52: Calibrated parameter C_y of different methods from images corrupted by artificial noise. Image size is 720×576, principal point $C_x = 360, C_y = 288$ and its offset are unknown. Different levels of Gaussian white noise from $\sigma = 0.1 \sim 0.5$ was added to the same captured images.



(a) 1% (b) 3%



(c) 5% (d) 7%



(e) 10%

Figure 6.53: Different percentages of synthetic noise added to the XY-plane. The chessboard has $12 \times 8 = 96$ corners. Distance between each corner is designed to be $20mm$. The introduced different noise levels are at $0.2, 0.6, 1.0, 1.4, 2.0$ (Unit:mm) in $X - Y$ coordinates, which is up to $2.0mm$ (10%) off from the original points in XY plane.

Table 6.26: Calibration results of different methods with 3D world points corrupted by different levels of noise. The chessboard has $12 \times 8 = 96$ corners, each grid is $20mm$ by $20mm$. Distance between each corner is designed to be $20mm$. The introduced different noise levels are at $0.2, 0.6, 1.0, 1.4, 2.0$ (Unit:mm) in $X - Y$ coordinates, which is up to $2.0mm$ (10%) off from the original points in XY plane. Image size is 720×576, principal point $C_x = 360, C_y = 288$ and its offset are unknown. μ, θ: average and standard deviation of each calibration parameter.

		f_x		f_y		C_x		C_y		RPE	
	Noise Level	μ	θ	μ	θ	μ	θ	μ	θ	μ	θ
Zhang-RD	1%	673.464	0.276	729.811	0.295	366.659	0.188	311.410	0.231	0.299	0.010
	3%	673.003	0.767	729.179	0.838	367.593	0.654	313.598	0.785	0.760	0.036
	5%	669.285	1.268	725.276	1.369	365.026	0.928	310.009	1.321	1.238	0.064
	7%	669.459	1.432	724.951	1.569	367.724	1.536	309.666	1.711	1.730	0.088
	10%	667.537	2.657	723.226	2.876	365.547	2.238	307.967	2.481	2.465	0.134
Andrea	1%	678.523	0.035	737.043	0.030	363.656	0.185	310.467	0.226	0.078	0.001
	3%	678.589	0.095	737.071	0.077	363.942	0.485	310.209	0.364	0.079	0.001
	5%	678.427	0.215	736.939	0.192	362.998	1.102	310.285	1.409	0.078	0.001
	7%	678.561	0.291	736.852	0.236	363.846	1.342	311.882	2.102	0.081	0.003
	10%	677.619	0.370	735.960	0.327	359.466	1.835	312.551	2.053	0.083	0.003
Proposed	1%	678.522	0.310	736.936	0.342	363.642	0.200	311.224	0.143	0.080	0.003
	3%	678.737	0.293	737.118	0.316	363.528	0.748	311.526	0.671	0.082	0.003
	5%	678.919	0.408	737.317	0.424	363.851	1.123	310.735	0.812	0.082	0.004
	7%	679.196	0.639	737.624	0.719	364.684	1.920	311.928	1.903	0.083	0.004
	10%	678.247	0.689	736.457	0.646	361.741	1.993	312.684	2.216	0.085	0.006

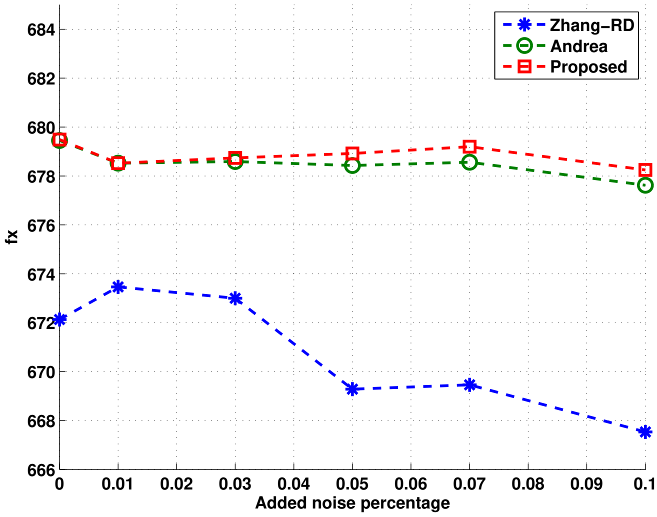


Figure 6.54: The calibrated parameter f_x of interest as a function of artificial noise added in 3D space. The chessboard has $12 \times 8 = 96$ corners. Distance between each corner is designed to be $20mm$. The introduced different noise levels are at $0.2, 0.6, 1.0, 1.4, 2.0$ (Unit:mm) in $X - Y$ coordinates, which is up to $2.0mm$ (10%) off from the original points in XY plane.

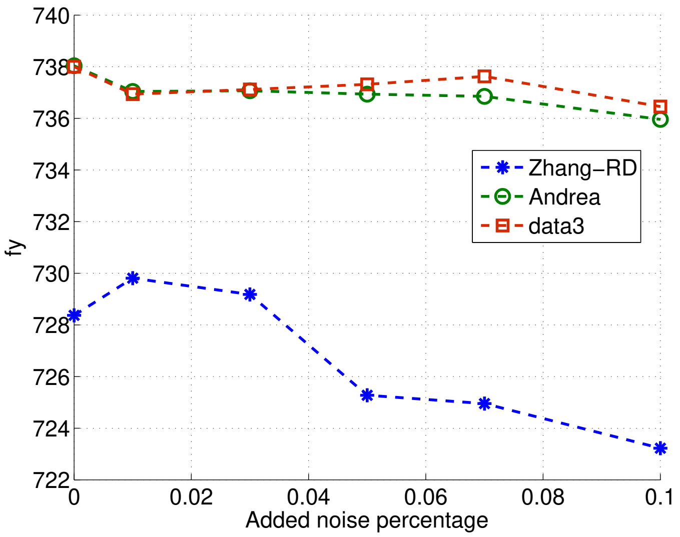


Figure 6.55: The calibrated parameter f_y of interest as a function of artificial noise added in 3D space. The chessboard has $12 \times 8 = 96$ corners. Distance between each corner is designed to be $20mm$. The introduced different noise levels are at $0.2, 0.6, 1.0, 1.4, 2.0$ (Unit:mm) in $X - Y$ coordinates, which is up to $2.0mm$ (10%) off from the original points in XY plane.

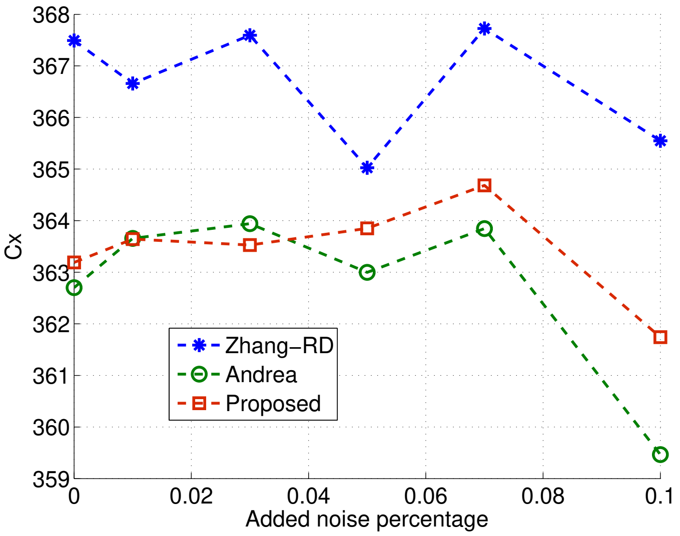


Figure 6.56: The calibrated parameter C_x of interest as a function of artificial noise added in 3D space. The chessboard has $12 \times 8 = 96$ corners. Distance between each corner is designed to be $20mm$. The introduced different noise levels are at $0.2, 0.6, 1.0, 1.4, 2.0$ (Unit:mm) in $X - Y$ coordinates, which is up to $2.0mm$ (10%) off from the original points in XY plane.

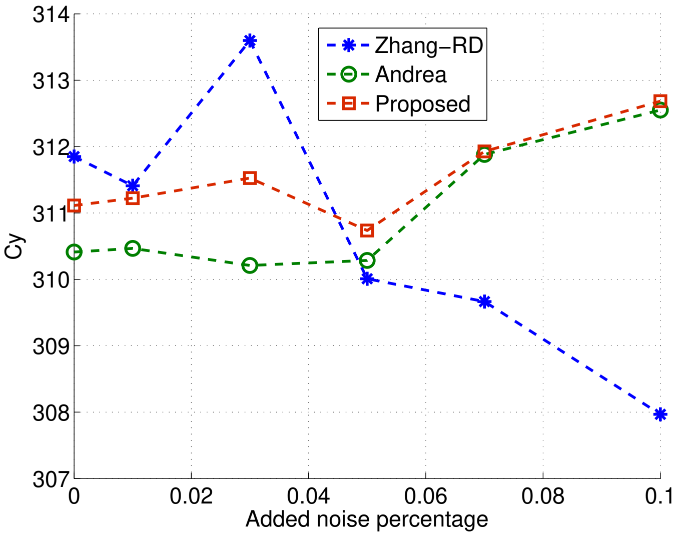


Figure 6.57: The calibrated parameter C_y of interest as a function of artificial noise added in 3D space. The chessboard has $12 \times 8 = 96$ corners. Distance between each corner is designed to be $20mm$. The introduced different noise levels are at $0.2, 0.6, 1.0, 1.4, 2.0$ (Unit:mm) in $X - Y$ coordinates, which is up to $2.0mm$ (10%) off from the original points in XY plane.

the deviation is limited to about 1.8 pixels compared to 6.5 by Zhang-RD method and 3.5 pixels by Andrea method. θ is the standard deviation of each calibration parameter during the 100 times running, the small the better. Zhang-RD has largest standard deviation θ than the other two in all parameters. Andrea has the smallest θ but its average result μ is bigger than the proposed method. To sum up, such noisy 3D points do have side effect on the calibration. If the detected 2D corners are accurate but it is easily to see that they can't be precisely matched during the optimization by LM method. However, the iteration process in proposed and Andrea method significantly reduce such effect.

6.5.3 Applied in 3D scanner data

Camera calibration have heavy impact to the image correct and final 3D reconstruction accuracy. In this section, we use the real image data from our 3D scanner to evaluate the proposed iterative algorithm for camera calibration and 3D reconstruction.

6.5.3.1 Experimental setup

The same procedure and measurement in section 6.4.2 will be used for this evaluation.

6.5.3.2 Results and analysis

The calibration, line segment fitting and 3D reconstruction results can be found on Tables 6.27, 6.28, 6.29, 6.30 and Figures 6.58, 6.59, 6.60,6.61, 6.62, 6.63. It can be observed from these tables and figures that: (1) Andrea and proposed methods produce much better calibration result of intrinsic, extrinsic parameters and 3D reconstruction. (2) As expected, while the noise getting heavier, their re-projection and 3D reconstruction errors increase. (3) While the noise is heavy enough (like $\tau > 10$), the re-projection error from Zhang-RD, Andrea method and proposed method increases rapidly from $1.233 \sim 3.006$, $0.279 \sim 0.633$ and $0.158 \sim 0.304$ respectively. Zhang-RD has almost ten times larger error than proposed method. It is clear that the proposed method benefited from the distortion free images for corner detection and it is more robust to noise than Andrea and Zhang-RD methods. Its re-projection error only increases about 0.2 as the noise level increases from $\tau = 0$ to $\tau = 25$.

The linear fitting results of the undistorted laser stripe images can be found in Figures 6.58, 6.59, 6.39, 6.38 6.60, 6.61. Overall, the laser stripes straightness by the proposed method are nearly 90° in different noise levels and their deviation is limited to 0.3° in average. Once we have the camera and scanner parameters, these laser stripes were reconstructed into 3D. The results are shown in Tables 6.18, 6.29, 6.30 and Figures 6.42 6.62, 6.63. The proposed method has the smallest average absolute error between the estimated and actual distances. It is only $0.15mm$ in the noise free data set. Compared to $0.29mm$ by Zhang-RD and $0.19mm$ by Andrea, it reduced by 48.27% and 26.7% respectively. Similar result can be found in the noisy dataset in the above tables. It is clearly shown in the 3D reconstruction deviation in above figures that the proposed method has limited errors between $-0.3 \sim 0.4mm$ in all noise levels, which is smaller than either Andrea's $-0.6 \sim 0.6mm$ or Zhang-RD's $-0.4 \sim 0.8mm$. Apparently, the noise do have heavy impact to the calibration, specially the noise was added to 3D corner coordinates. The noise increases the difficulty for the optimization in the non-iterative process like Zhang-RD to converge and the result is not stable while the noise levels increases. In addition, our scanner's parameters estimation is highly depended on the camera calibration result, hence, the final 3D error is larger. From above figures and table we can notice that as the distance between the scanner and the flat board increase, the 3D reconstruction absolute error is increasing and unstable too. It is because the noise has much heavier impact to peak location as the object distance increase. As we mentioned in section 6.3.2.2, the relation between 3D reconstruction error and laser peak is not linear but exponential growth when the camera calibration parameters are known and fixed.

6.5.4 Conclusion

In this section, we have proposed a novel algorithm for camera calibration and image correction. This algorithm is based on the iteration procedure which considers the inaccuracy of 3D pattern (like chessboard) and the lens distortion effect to the feature point detection. A comparative study based on synthetic data, real image and data from our

Table 6.27: Calibration result using Andrea method and the 3D scanner's data corrupted by different multiples of a basic noise. The image size is 720×576 and calibration target contains $8 \times 6 = 48$ corners, every corner has a distance of $20mm$ from each other. Gaussian white noise with standard deviation $\delta = 0.04\tau$ was added to the coordinates of the 3D world points $p_{wi}(i = 1, 2, \cdots 48)$ and $\delta = 0.02\tau$ was added to image points P_{fi} where τ varied from 0 to 25 at intervals of 1. μ: average result for each parameter, σ: standard deviation for each parameter, RPE: re-projection error

	f_x		f_y		C_x		C_y		RPE	
τ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
0	1019.571	0.000	1086.079	0.000	346.498	0.000	293.412	0.000	0.109	0.000
5	1020.878	1.630	1086.940	2.014	344.454	10.137	292.692	9.349	0.167	0.011
10	1019.930	3.689	1085.633	4.550	347.502	24.013	289.551	17.805	0.279	0.023
15	1020.596	5.460	1086.196	7.513	357.589	32.183	294.104	34.195	0.392	0.028
20	1018.891	5.059	1082.891	7.150	352.649	33.752	298.021	38.694	0.519	0.042
25	1012.404	8.456	1078.985	9.675	366.158	53.710	303.311	40.316	0.633	0.045

Table 6.28: Calibration result using proposed iterative method and the 3D scanner's data corrupted by different multiples of a basic noise. The image size is 720×576 and calibration target contains $8\times6 = 48$ corners, every corner has a distance of $20mm$ from each other. Gaussian white noise with standard deviation $\delta = 0.04\tau$ was added to the coordinates of the 3D world points $p_{wi}(i = 1, 2, \cdots 48)$ and $\delta = 0.02\tau$ was added to image points P_{fi} where τ varied from 0 to 25 at intervals of 1. μ: average result for each parameter, σ: standard deviation for each parameter, RPE: re-projection error

	f_x		f_y		C_x		C_y		RPE	
τ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
0	1019.903	0.000	1086.512	0.000	347.048	0.000	292.904	0.000	0.112	0.000
5	1019.465	0.763	1086.062	1.134	343.501	7.406	283.283	11.858	0.128	0.011
10	1020.511	1.580	1085.725	2.685	365.786	20.413	292.043	15.873	0.158	0.020
15	1020.091	3.233	1086.573	3.506	334.849	23.321	316.435	31.373	0.204	0.024
20	1020.711	2.512	1086.274	5.218	339.599	25.667	332.895	18.633	0.245	0.028
25	1013.530	7.005	1078.160	6.843	409.536	40.818	273.827	35.613	0.304	0.032

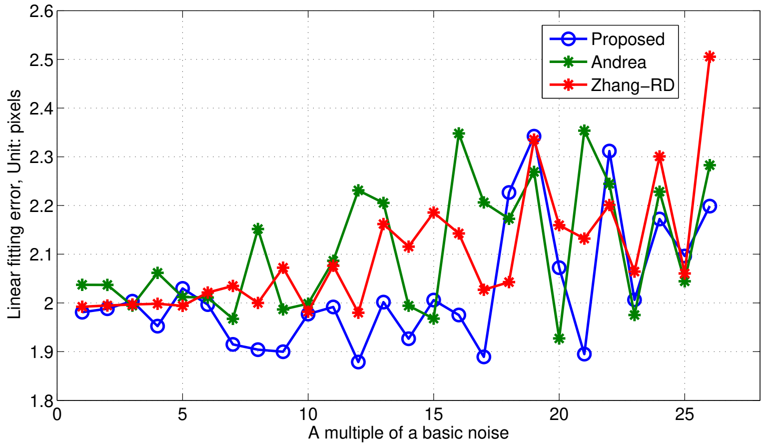


Figure 6.58: The average linear fitting error of laser stripes after the image correction using different calibration algorithms at each multiple of a basic noise. The scanner travels 105 steps at 1mm interval. Gaussian white noise with standard deviation $\delta = 0.04\tau$ was added to the coordinates of the 3D world points $p_{wi}(i = 1, 2, \cdots 48)$ and $\delta = 0.02\tau$ was added to image points P_{fi} where τ varied from 0 to 25 at intervals of 1.

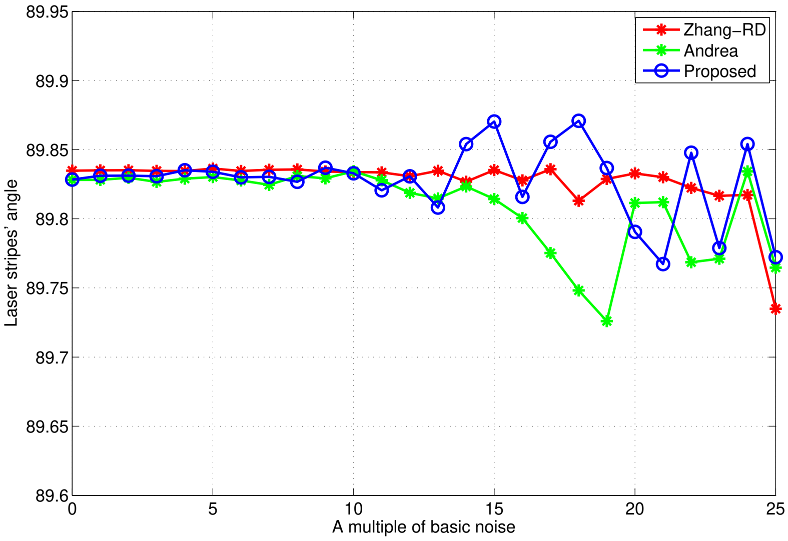


Figure 6.59: The average linear fitting angle of laser stripes after the image correction using different calibration algorithms at each multiple of a basic noise. The scanner travels 105 steps at 1mm interval. Gaussian white noise with standard deviation $\delta = 0.04\tau$ was added to the coordinates of the 3D world points $p_{wi}(i = 1, 2, \cdots 48)$ and $\delta = 0.02\tau$ was added to image points P_{fi} where τ varied from 0 to 25 at intervals of 1.

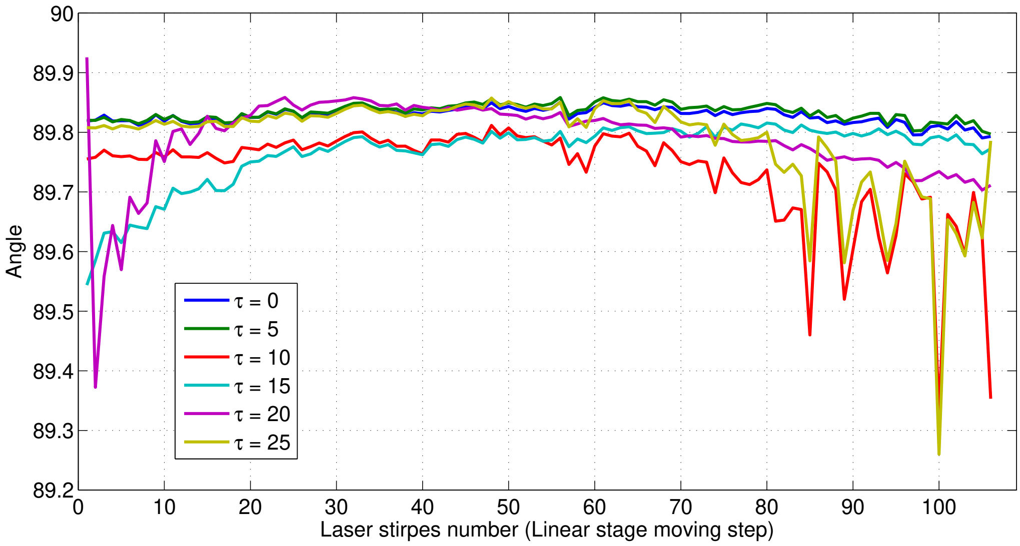


Figure 6.60: The average angle of laser stripes at each linear stage moving step based on the linear fitting after the image correction using Andrea algorithm with various multiples of a basic noise. The scanner travels 105 steps at 1mm interval. Gaussian white noise with standard deviation $\delta = 0.04\tau$ was added to the coordinates of the 3D world points $p_{wi}(i = 1, 2, \cdots 48)$ and $\delta = 0.02\tau$ was added to image points P_{fi} where τ varied from 0 to 25 at intervals of 1.

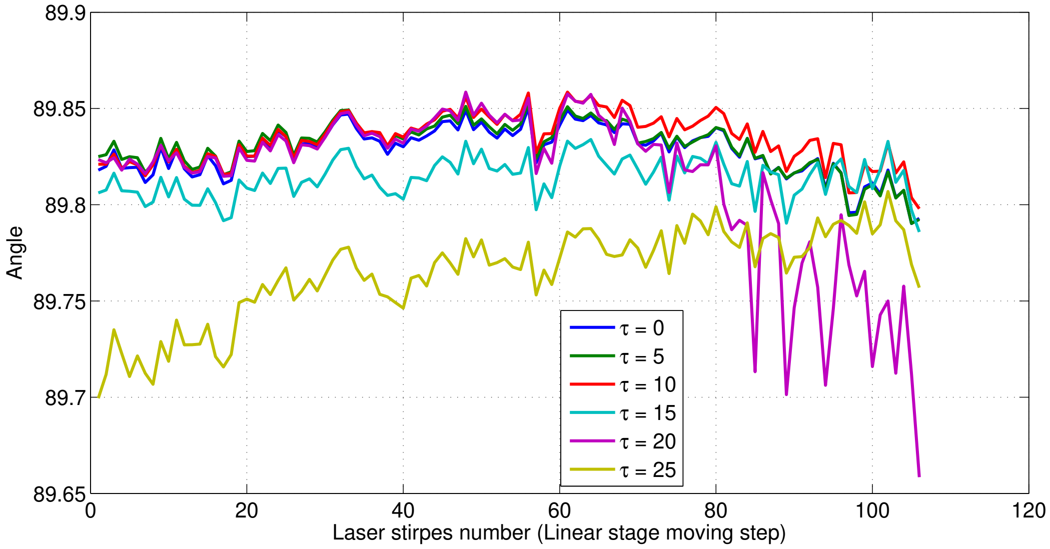


Figure 6.61: The average angle of laser stripes at each linear stage moving step based on the linear fitting after the image correction using proposed algorithm with various multiples of a basic noise. The scanner travels 105 steps at 1mm interval. Gaussian white noise with standard deviation $\delta = 0.04\tau$ was added to the coordinates of the 3D world points $p_{wi}(i = 1, 2, \cdots 48)$ and $\delta = 0.02\tau$ was added to image points P_{fi} where τ varied from 0 to 25 at intervals of 1.

Table 6.29: Absolute error Δ and standard deviation σ of laser peak 3D reconstruction based on Andrea algorithm with various multiples of a basic noise. Gaussian white noise with standard deviation $\delta = 0.04\tau$ was added to the coordinates of the 3D world points $p_{wi}(i = 1, 2, \cdots 48)$ and $\delta = 0.02\tau$ was added to image points P_{fi} where τ varied from 0 to 25 at intervals of 1. Ref: actual measurement, τ: multiple noise, μ estimated peak location. Unit:mm

	$\tau = 0$			$\tau = 5$			$\tau = 10$			$\tau = 15$			$\tau = 20$			$\tau = 25$		
Ref	μ	Δ	σ	μ	Δ	σ	μ	Δ	σ	μ	Δ	σ	μ	Δ	σ	μ	Δ	σ
148.00	148.03	0.03	0.08	148.39	0.39	0.08	148.36	0.36	0.11	148.14	0.14	0.07	148.63	0.63	0.11	148.40	0.40	0.11
158.00	158.16	0.16	0.09	158.27	0.27	0.10	158.22	0.22	0.12	158.27	0.27	0.08	158.52	0.52	0.08	158.57	0.57	0.11
168.00	168.18	0.18	0.11	168.26	0.26	0.12	168.21	0.21	0.13	168.28	0.28	0.11	168.52	0.52	0.09	168.60	0.60	0.14
178.00	178.18	0.18	0.12	178.25	0.25	0.12	178.22	0.22	0.13	178.29	0.29	0.14	178.51	0.51	0.12	178.61	0.61	0.19
188.00	188.21	0.21	0.14	188.23	0.23	0.14	188.21	0.21	0.14	188.31	0.31	0.18	188.50	0.50	0.20	188.62	0.62	0.27
198.00	198.25	0.25	0.15	198.18	0.18	0.15	198.18	0.18	0.14	198.35	0.35	0.21	198.45	0.45	0.26	198.63	0.63	0.32
208.00	208.27	0.27	0.15	208.14	0.14	0.14	208.19	0.19	0.12	208.38	0.38	0.22	208.43	0.43	0.31	208.60	0.60	0.36
218.00	218.30	0.30	0.17	218.10	0.10	0.16	218.19	0.19	0.13	218.42	0.42	0.25	218.39	0.39	0.37	218.56	0.56	0.41
228.00	228.22	0.22	0.16	228.16	0.16	0.15	228.31	0.31	0.11	228.36	0.36	0.25	228.47	0.47	0.41	228.39	0.39	0.44
238.00	238.18	0.18	0.20	238.18	0.18	0.19	238.41	0.41	0.13	238.33	0.33	0.31	238.51	0.51	0.51	238.23	0.23	0.54
248.00	248.15	0.15	0.21	248.19	0.19	0.20	248.50	0.50	0.13	248.32	0.32	0.32	248.55	0.55	0.56	248.06	0.06	0.60
253.00	253.13	0.13	0.21	253.20	0.20	0.20	253.56	0.56	0.14	253.31	0.31	0.32	253.58	0.58	0.56	253.04	0.04	0.60
Avg	-	0.19	-	-	0.21	-	-	0.30	-	-	0.31	-	-	0.51	-	-	0.44	-

Table 6.30: Absolute error Δ and standard deviation σ of laser peak 3D reconstruction based on the proposed algorithm with various multiples of a basic noise. Gaussian white noise with standard deviation $\delta = 0.04\tau$ was added to the coordinates of the 3D world points $p_{wi}(i = 1, 2, \cdots 48)$ and $\delta = 0.02\tau$ was added to image points P_{fi} where τ varied from 0 to 25 at intervals of 1. Ref: actual measurement, τ: multiple noise, μ estimated peak location. Unit:mm

	$\tau = 0$			$\tau = 5$			$\tau = 10$			$\tau = 15$			$\tau = 20$			$\tau = 25$		
Ref	μ	Δ	σ	μ	Δ	σ	μ	Δ	σ	μ	Δ	σ	μ	Δ	σ	μ	Δ	σ
148.00	148.05	0.05	0.05	147.99	0.01	0.07	147.95	0.05	0.09	148.20	0.20	0.09	148.20	0.20	0.10	148.32	0.32	0.10
158.00	157.93	0.07	0.09	157.88	0.12	0.06	157.83	0.17	0.10	158.09	0.09	0.08	158.10	0.10	0.11	158.23	0.23	0.11
168.00	167.92	0.08	0.11	167.87	0.13	0.07	167.82	0.18	0.11	168.10	0.10	0.10	168.11	0.11	0.12	168.25	0.25	0.14
178.00	177.91	0.09	0.12	177.87	0.13	0.08	177.82	0.18	0.12	178.11	0.11	0.10	178.14	0.14	0.12	178.27	0.27	0.15
188.00	187.89	0.11	0.14	187.85	0.15	0.09	187.80	0.20	0.13	188.10	0.10	0.11	188.15	0.15	0.13	188.27	0.27	0.18
198.00	197.83	0.17	0.14	197.80	0.20	0.11	197.77	0.23	0.12	198.08	0.08	0.10	198.15	0.15	0.13	198.24	0.24	0.19
208.00	207.79	0.21	0.14	207.77	0.23	0.12	207.75	0.25	0.10	208.08	0.08	0.10	208.18	0.18	0.12	208.22	0.22	0.18
218.00	217.73	0.27	0.16	217.73	0.27	0.14	217.74	0.26	0.09	218.07	0.07	0.10	218.21	0.21	0.13	218.17	0.17	0.19
228.00	227.78	0.22	0.15	227.79	0.21	0.13	227.83	0.17	0.10	228.17	0.17	0.13	228.26	0.26	0.11	228.23	0.23	0.15
238.00	237.78	0.22	0.16	237.82	0.18	0.13	237.90	0.10	0.13	238.26	0.26	0.16	238.28	0.28	0.13	238.27	0.27	0.13
248.00	247.76	0.24	0.17	247.83	0.17	0.14	247.96	0.04	0.13	248.31	0.31	0.17	248.27	0.27	0.13	248.24	0.24	0.12
253.00	252.76	0.24	0.17	252.84	0.16	0.14	253.01	0.01	0.13	253.33	0.33	0.17	253.27	0.27	0.13	253.21	0.21	0.13
Avg	-	0.15	-	-	0.16	-	-	0.16	-	-	0.15	-	-	0.20	-	-	0.25	-

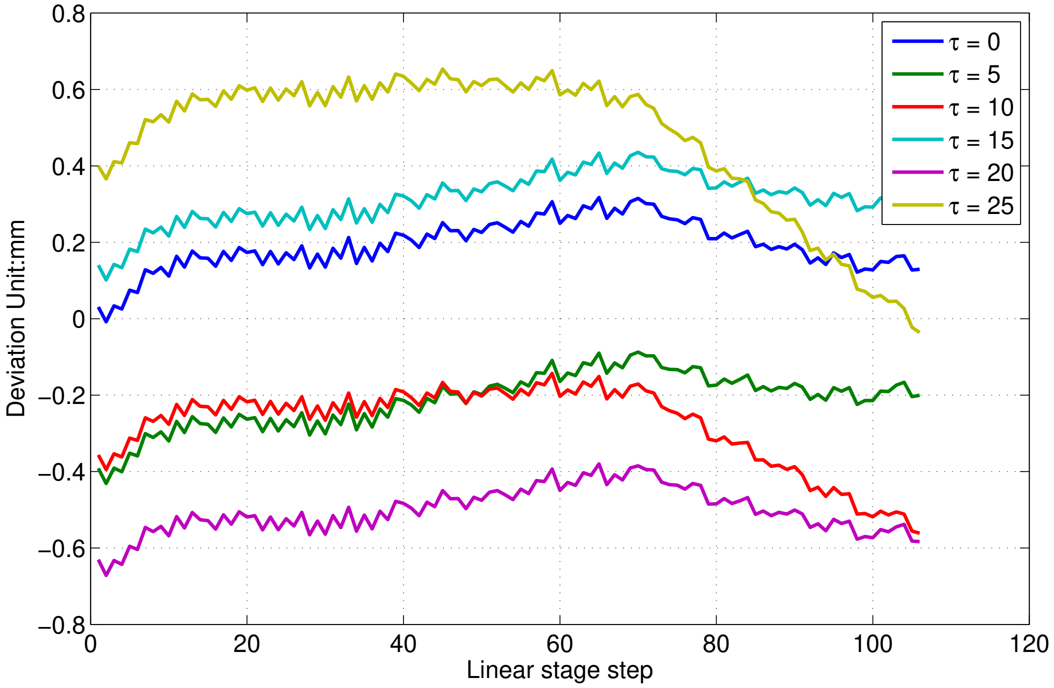


Figure 6.62: The laser stripes 3D reconstruction error as a function of the linear stage moving step based on Andrea algorithm with various multiples of a basic noise. The scanner travels 105 steps at 1mm interval. Gaussian white noise with standard deviation $\delta = 0.04\tau$ was added to the coordinates of the 3D world points $p_{wi}(i = 1, 2, \cdots 48)$ and $\delta = 0.02\tau$ was added to image points P_{fi} where τ varied from 0 to 25 at intervals of 1.

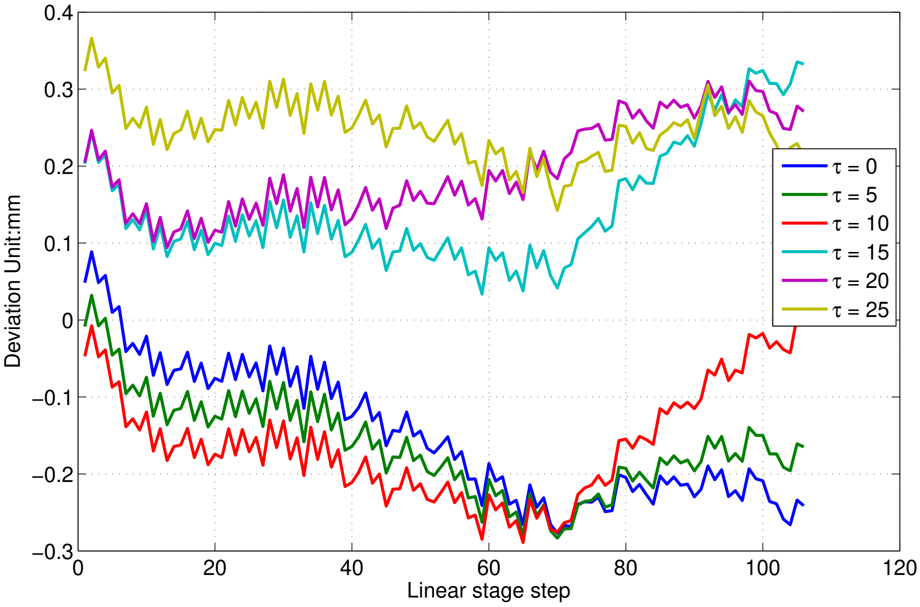


Figure 6.63: The laser stripes 3D reconstruction error as a function of the linear stage moving step based on the proposed algorithm with various multiples of a basic noise. The scanner travels 105 steps at 1mm interval. Gaussian white noise with standard deviation $\delta = 0.04\tau$ was added to the coordinates of the 3D world points $p_{wi}(i = 1, 2, \cdots 48)$ and $\delta = 0.02\tau$ was added to image points P_{fi} where τ varied from 0 to 25 at intervals of 1.

3D scanner corrupted by reasonable levels of noise shows that the proposed algorithm successfully calibrated, corrected the distorted image and feature points and precisely reconstructed the captured laser stripes in 3D space.

As we all known the 3D reconstruction result's accuracy is related to accurate calibration parameters including distortion factors and well estimated scanner parameters in our triangulated scanning system. Since our estimation of scanner parameters is based on the knowledge of camera calibration result, hence, inaccurate calibration amplifies such error in 3D space. The proposed method benefited from performing the corner detection on distortion free images and taking account of the errors coming from low cost made calibration board. Its absolute error and standard deviation are much smaller than Zhang-RD and Andrea's.

Noble corner detector was selected in the experiment. The calibration pattern distortion does affect its gradient based search process for sub-pixel localization of control points. Zhang-RD has the worst calibration and image correction performance in the overall experiment because the corners were inaccurate since they detected from distorted images. The corner detection do have better performance in distortion free image. That is why Andrea's has better result than Zhang-RD. However, the manufacturing process of the 3D pattern will always introduce errors to the pattern. In addition, such precise 3D pattern is quite expensive. In most cases, the printed and laminated pattern is still the first choice. Thus, inaccurate 3D pattern bring error into the calibration. This is another reason why Zhang-RD has the worst performance. Andrea hasn't considered such effect from inaccurate input 3D coordinates. The proposed method has the best result because it considers both conditions and using iterative method to refine the input 2D/3D coordinates by (1) the corner detection run on the distortion free images at each round; (2) it used the LM and bundle adjustment to refine the inaccurate 3D coordinates. In other words, it directly save our budget since the precise calibration pattern(like chessboard) is very expensive, particularly it need to be customized to fit for our projects.

The iteration procedure also has its disadvantage: it takes more time to compute. From the experiment, iteration in Andrea and proposed methods converged very quickly, normally less than ten rounds. The proposed method takes about 2.2 seconds in each round in a quad core Intel processor (Q6600 2.4 GHz) based PC. Andrea takes about 3.4 seconds. The proposed method reduces about 50% computational time.

6.6 Overall system performance evaluation

This experiment aims for evaluating the performance with or without the components investigated in the above sections. We are trying to compare and investigate the optimal settings of different components with those and without optimization on the accuracy of 3D reconstruction, so that we can gain knowledge whether optimal components will lead to an optimal system.

6.6.1 Experimental setup

Similar procedure and measurement in section 6.4.2 is used for the evaluation. Table 6.31 shows the settings (A) and (B) with different combination of methods for scanning tasks. Setting (A) evaluates the existing and general methods for camera calibration, image noise removal, feature detection and 3D reconstruction. The window size for Gaussian smoothing varies from 3×3 to 7×7 as the noise increases. Setting (B) applies all the methods mentioned and investigated in the above sections. As we discussed in above sections, SCS denoise is heavily time consuming procedure. It takes around 25 seconds to denoise a image with 720×576 resolution. It will be very long time if considering the amount of chessboard and laser stripes images, SCS noise removal applying in each iteration process and 100 times running at different noise levels. Hence, we only took the pictures every $10mm$ instead of taking very millimetre, which gave us 11 images and the last step is $5mm$ because the limit of linear stage's length. In total there are 12 laser stripe images using in this experiment. To simulate the different working environments/conditions, the Gaussian white noise with standard deviation $\delta = 0.04\tau$ was added to the 3D world points, $p_{wi}(i = 1, 2, \cdots 48)$ and $\delta = 0.04\tau$ was added to the chessboard and laser stripe images, where τ varied from 0 to 25 at intervals of 1 simulating different levels of noise. The noise added to the 2D images is higher than previous experiments(in this experiment $\delta = 0.04\tau$, the previous are $\delta = 0.02\tau$) because it was directly added to the image itself instead of the detected 2D points. If the Gaussian noise adding to 2D points has standard deviation $\delta = 0.02\tau$, where $\tau = 0 \cdots 25$ then the corners have a maximum of 0.5 pixel shifted. It is reasonable and heavy enough to test the calibration method but not for the test of the corner detection and noise removal method. Hence, heavier noise $\delta = 0.4\tau$ was added to the images in this experiment. Each setting will run 100 times at different noise levels, and their average was calculated as final results.

6.6.2 Results and analysis

The calibration results based on setting (A) and (B) are shown in Tables 6.32 and 6.33. The proposed setting (B) produces much more stable and noise resistant results. The re-projection error is much smaller using setting (B) than (A). The standard deviation σ of each parameter in setting (B) is almost ten times smaller than in (A). It is clear that these parameters average result μ are more stable and their standard deviation (σ) are much smaller in setting (B) than (A). For example, the intrinsic parameters f_x, f_y

Table 6.31: Two different settings applied with 3D scanner data

Setting	A	B
Noise removal	Gaussian Smooth	SCS
Corner detection	Noble	Noble
Laser peak detection	Gaussian approximation	FIR
Distortion model	Radial + Decentring	Fraction
Calibration	Zhang-RD/OpenCV	Proposed iterative
Optimization	None	Yes

Table 6.32: Calibration result with setting (A) based on the 3D scanner's data corrupted by different multiples of a basic noise. The window size for Gaussian smoothing varies from 3×3 to 7×7 as the noise increases. Gaussian white noise with standard deviation $\delta = 0.04\tau$ was added to the 3D world points, $p_{wi}(i = 1, 2, \cdots 48)$ and $\delta = 0.04\tau$ was added to the chessboard and laser stripe images, where τ varied from 0 to 25 at intervals of 1. μ: average result for each parameter, σ: standard deviation for each parameter, RPE: re-projection error

	f_x		f_y		C_x		C_y		RPE	
τ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
0	1003.632	0.000	1068.132	0.000	343.578	0.000	291.598	0.000	0.229	0.000
5	1003.674	0.578	1067.984	0.658	352.248	2.562	282.799	2.123	0.609	0.037
10	1003.052	2.778	1068.616	3.077	343.604	6.744	294.003	5.266	1.230	0.115
15	1011.472	4.947	1082.465	5.190	326.729	9.749	264.735	14.227	1.702	0.234
20	1008.332	7.792	1069.681	8.759	321.777	11.148	302.528	14.194	2.385	0.222
25	1003.185	7.195	1066.618	7.544	345.812	17.784	306.883	25.689	2.919	0.382

Table 6.33: Calibration result with setting (B) based on the 3D scanner's data corrupted by different multiples of a basic noise. Gaussian white noise with standard deviation $\delta = 0.04\tau$ was added to the 3D world points, $p_{wi}(i = 1, 2, \cdots 48)$ and $\delta = 0.04\tau$ was added to the chessboard and laser stripe images, where τ varied from 0 to 25 at intervals of 1. μ: average result for each parameter, σ: standard deviation for each parameter, RPE: re-projection error

	f_x		f_y		C_x		C_y		RPE	
τ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
0	1018.796	0.000	1084.733	0.000	344.809	0.000	300.027	0.000	0.110	0.000
5	1017.759	0.220	1083.588	0.345	348.664	4.101	297.564	4.469	0.137	0.001
10	1018.271	0.361	1084.170	0.601	350.092	7.759	298.857	8.824	0.138	0.002
15	1018.635	0.617	1084.606	0.833	343.505	12.327	297.754	12.909	0.142	0.003
20	1018.477	0.860	1084.192	1.200	350.931	12.949	296.782	13.882	0.144	0.005
25	1017.801	0.863	1084.172	1.105	348.332	13.461	294.955	15.589	0.150	0.003

Table 6.34: Absolute error Δ and standard deviation σ of laser peak 3D reconstruction based on setting (A) with various multiples of a basic noise. Gaussian white noise with standard deviation $\delta = 0.04\tau$ was added to the 3D world points, $p_{wi}(i = 1, 2, \cdots 48)$ and $\delta = 0.04\tau$ was added to the chessboard and laser stripe images, where τ varied from 0 to 25 at intervals of 1. Ref: actual measurement, τ: multiple noise, μ estimated peak location. Unit:mm

	$\tau = 0$			$\tau = 5$			$\tau = 10$			$\tau = 15$			$\tau = 20$			$\tau = 25$		
Ref	μ	Δ	σ	μ	Δ	σ	μ	Δ	σ	μ	Δ	σ	μ	Δ	σ	μ	Δ	σ
148.000	149.010	1.010	0.073	149.480	1.480	0.090	148.370	0.370	0.072	149.100	1.100	0.151	147.350	0.650	0.144	151.350	3.350	0.127
158.000	158.910	0.910	0.085	159.380	1.380	0.098	158.270	0.270	0.082	159.000	1.000	0.148	157.240	0.760	0.131	161.240	3.240	0.123
168.000	168.910	0.910	0.105	169.380	1.380	0.113	168.270	0.270	0.102	169.000	1.000	0.149	167.240	0.760	0.136	171.240	3.240	0.126
178.000	178.910	0.910	0.110	179.380	1.380	0.112	178.260	0.260	0.109	179.000	1.000	0.134	177.240	0.760	0.137	181.240	3.240	0.118
188.000	188.870	0.870	0.125	189.340	1.340	0.121	188.230	0.230	0.126	188.960	0.960	0.126	187.190	0.810	0.150	191.190	3.190	0.125
198.000	198.800	0.800	0.126	199.270	1.270	0.115	198.170	0.170	0.121	198.890	0.890	0.108	197.130	0.870	0.163	201.110	3.110	0.128
208.000	208.750	0.750	0.119	209.220	1.220	0.099	208.120	0.120	0.103	208.830	0.830	0.087	207.060	0.940	0.179	211.040	3.040	0.138
218.000	218.660	0.660	0.119	219.140	1.140	0.096	218.060	0.060	0.086	218.760	0.760	0.080	216.950	1.050	0.197	220.920	2.920	0.166
228.000	228.710	0.710	0.122	229.190	1.190	0.094	228.130	0.130	0.108	228.780	0.780	0.080	226.930	1.070	0.207	230.900	2.900	0.199
238.000	238.690	0.690	0.144	239.170	1.170	0.112	238.150	0.150	0.125	238.750	0.750	0.102	236.830	1.170	0.243	240.790	2.790	0.252
248.000	248.650	0.650	0.151	249.150	1.150	0.118	248.150	0.150	0.122	248.720	0.720	0.132	246.700	1.300	0.298	250.630	2.630	0.310
253.000	253.640	0.640	0.140	254.150	1.150	0.112	253.160	0.160	0.116	253.710	0.710	0.157	251.630	1.370	0.309	255.540	2.540	0.318
Avg	-	0.79	-	-	1.27	-	-	0.20	-	-	0.88	-	-	0.96	-	-	3.02	-

Table 6.35: Absolute error Δ and standard deviation σ of laser peak 3D reconstruction based on setting (B) with various multiples of a basic noise. Gaussian white noise with standard deviation $\delta = 0.04\tau$ was added to the 3D world points, $p_{wi}(i = 1, 2, \cdots 48)$ and $\delta = 0.04\tau$ was added to the chessboard and laser stripe images, where τ varied from 0 to 25 at intervals of 1. Ref: actual measurement, τ: multiple noise, μ estimated peak location. Unit:mm

	$\tau = 0$			$\tau = 5$			$\tau = 10$			$\tau = 15$			$\tau = 20$			$\tau = 25$		
Ref	μ	Δ	σ	μ	Δ	σ	μ	Δ	σ	μ	Δ	σ	μ	Δ	σ	μ	Δ	σ
148.000	148.040	0.040	0.067	148.340	0.340	0.072	148.140	0.140	0.071	148.390	0.390	0.050	148.380	0.380	0.081	148.120	0.120	0.134
158.000	157.930	0.070	0.054	158.220	0.220	0.081	158.020	0.020	0.082	158.270	0.270	0.060	158.260	0.260	0.087	158.010	0.010	0.124
168.000	167.930	0.070	0.059	168.210	0.210	0.087	168.010	0.010	0.089	168.260	0.260	0.070	168.250	0.250	0.095	168.000	0.000	0.111
178.000	177.930	0.070	0.062	178.200	0.200	0.097	178.000	0.000	0.100	178.250	0.250	0.070	178.240	0.240	0.100	177.990	0.010	0.099
188.000	187.910	0.090	0.061	188.170	0.170	0.093	187.970	0.030	0.104	188.220	0.220	0.091	188.210	0.210	0.105	187.950	0.050	0.093
198.000	197.860	0.140	0.073	198.120	0.120	0.078	197.920	0.080	0.098	198.170	0.170	0.103	198.160	0.160	0.103	197.900	0.100	0.080
208.000	207.810	0.190	0.081	208.080	0.080	0.062	207.880	0.120	0.079	208.130	0.130	0.100	208.110	0.110	0.087	207.860	0.140	0.069
218.000	217.750	0.250	0.098	218.000	0.000	0.065	217.800	0.200	0.074	218.050	0.050	0.113	218.040	0.040	0.086	217.790	0.210	0.091
228.000	227.780	0.220	0.081	228.030	0.030	0.072	227.830	0.170	0.089	228.080	0.080	0.104	228.080	0.080	0.099	227.830	0.170	0.093
238.000	237.780	0.220	0.092	237.990	0.010	0.080	237.790	0.210	0.093	238.040	0.040	0.105	238.030	0.030	0.105	237.790	0.210	0.124
248.000	247.750	0.250	0.100	247.950	0.050	0.079	247.750	0.250	0.093	247.980	0.020	0.115	247.980	0.020	0.108	247.750	0.250	0.122
253.000	252.730	0.270	0.087	252.940	0.060	0.095	252.740	0.260	0.113	252.980	0.020	0.119	253.000	0.000	0.130	252.760	0.240	0.133
Avg	-	0.16	-	-	0.12	-	-	0.12	-	-	0.16	-	-	0.15	-	-	0.13	-

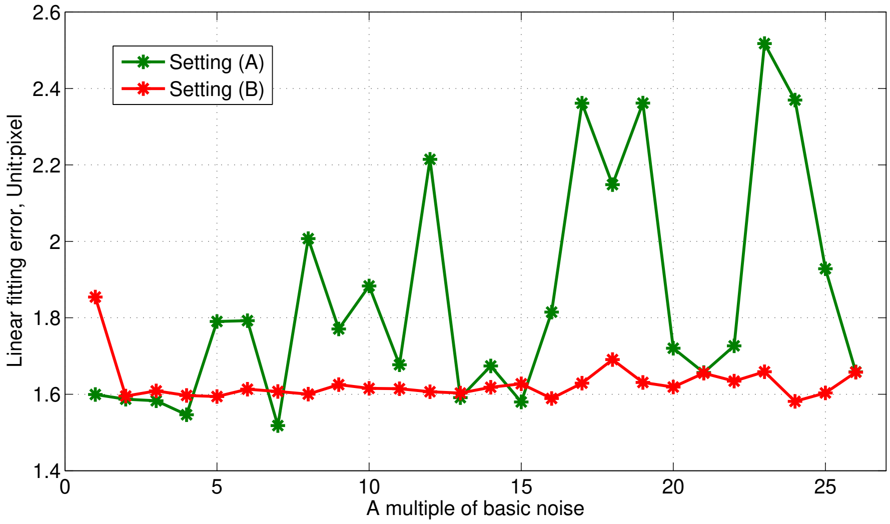


Figure 6.64: The average linear fitting error of laser stripes after the image correction using different settings at each multiple of a basic noise. Gaussian white noise with standard deviation $\delta = 0.04\tau$ was added to the 3D world points, $p_{wi}(i = 1, 2, \cdots 48)$ and $\delta = 0.04\tau$ was added to the chessboard and laser stripe images, where τ varied from 0 to 25 at intervals of 1.

vary within 1.5 pixels in Setting (B), which is much smaller than 16 pixels in (A). In the highest noise level, the re-projection error in setting (B) is only 0.04 bigger than its noise free data. It increases by only 36.37% as the noise vary from $\tau = 0$ to $\tau = 25$, compared to setting (A) has increased over 13 times.

These intrinsic and distortion parameters were used to undistort the laser stripe images and then performed the peak detection. After then, linear fitting was applied, the fitting error and angles are shown in Figures 6.64, 6.65, 6.66 and 6.67. From these figures we can observe that setting (B) undistorted images are better than those by setting (A). The fitting errors are smaller and the laser stripes' angles are closer to the expected 90°(vertical in camera Y-axis) in setting (B). For example, line fitting errors based on setting (B) are around 1.6 in Fig.6.64. This appears much smaller than setting (A), which fitting error varies from 1.5 to 2.5.

Both of these two sets of laser peak are reconstructed to 3D using their corresponding camera and scanner parameters respectively. The results are shown in Tables 6.34 and 6.35. It is clear that the 3D reconstruction based on setting (B) has smaller absolute error and standard derivation. Its maximum absolute error is $0.39mm$ and the average deviation in different noise levels is below 0.15. However, the 3D reconstructed based on setting (A) has a maximum deviation of $3.35mm$ and the average deviation is over 0.3 in different noise levels. It is clear that the result of Setting (B) is more stable in different noise levels and closer to ground truth values.

From these results, it can be seen that each component plays important role in the

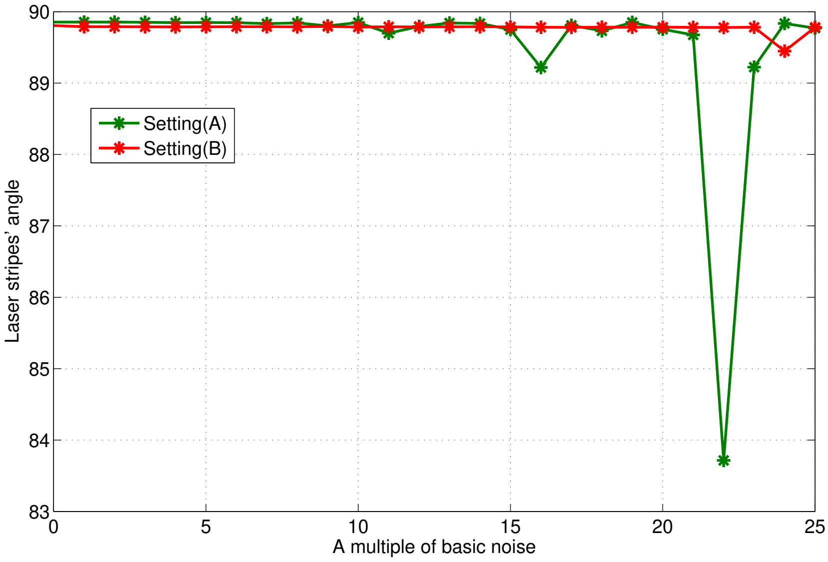


Figure 6.65: The average angle of laser stripes for 10 moving steps based on the linear fitting after the image correction based on different settings at each multiple of a basic noise. Gaussian white noise with standard deviation $\delta = 0.04\tau$ was added to the 3D world points, $p_{wi}(i = 1, 2, \cdots 48)$ and $\delta = 0.04\tau$ was added to the chessboard and laser stripe images, where τ varied from 0 to 25 at intervals of 1.

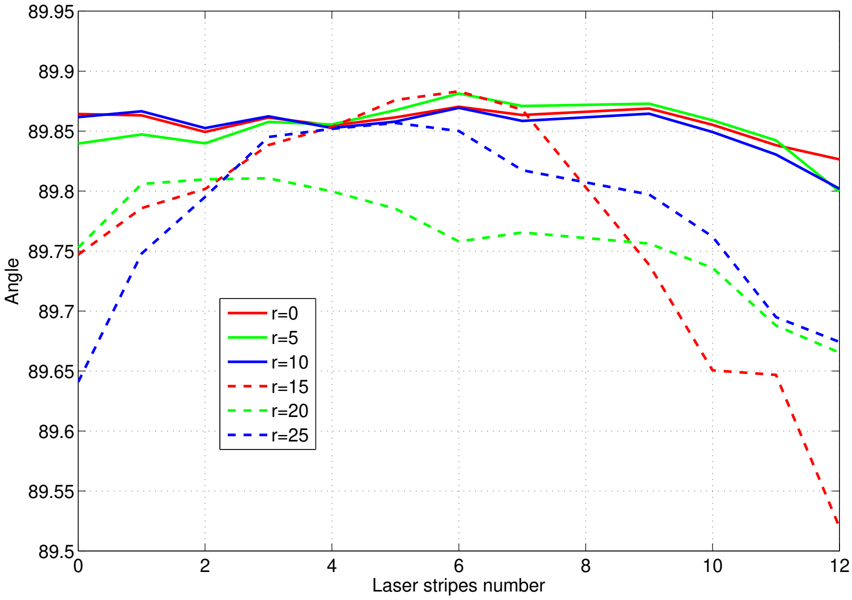


Figure 6.66: The average angle of laser stripes at each linear stage moving step based on the linear fitting after the image correction with setting (A) with various multiples (τ) of a basic noise. Gaussian white noise with standard deviation $\delta = 0.04\tau$ was added to the 3D world points, $p_{wi}(i = 1, 2, \cdots 48)$ and $\delta = 0.04\tau$ was added to the chessboard and laser stripe images, where τ varied from 0 to 25 at intervals of 1.

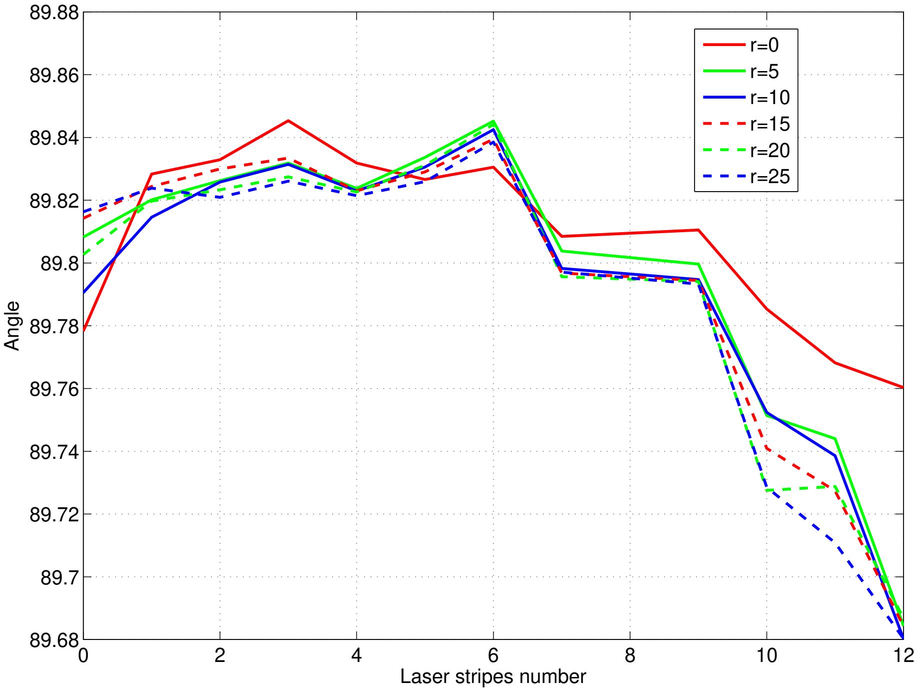


Figure 6.67: The average angle of laser stripes at each linear stage moving step based on the linear fitting after the image correction with setting (B) with various multiples (τ) of a basic noise. Gaussian white noise with standard deviation $\delta = 0.04\tau$ was added to the 3D world points, $p_{wi}(i = 1, 2, \cdots 48)$ and $\delta = 0.04\tau$ was added to the chessboard and laser stripe images, where τ varied from 0 to 25 at intervals of 1.

overall scanner system. The final result benefited from the advantage at each *optimal* component in (B) than (A). Fox example, FIR has the ability to detect accurate peak in highly saturated laser stripe image while a general method like Gaussian approximation may fail. In such case, inaccurate peaks cause inaccurate scanner parameters' such as the laser projection angle α estimation and lead to inaccurate 3D reconstruction. It is harder to determine how much the improvement of each component in the final result using (B) than (A) but when each component performs better, the final system is more advanced than just using the general methods.

6.6.3 Conclusion

In this experiment, we evaluated the performance of the 3D laser scanning system based on different settings (A) and (B). Setting (B) has all the selected components for our scanner. It is clear that the selected Setting (B) has much better performance than (A). The only common method among them is corner detection, the rest methods/components are different and have been already evaluated in previous sections. A comparative study based on real images from 3D scanner corrupted by reasonable levels of noise shows that the proposed algorithms successfully calibrated and corrected the distorted image points. In additional, we compared 3D reconstruction result based on the calibration parameters with our 3D image data. The results show that the setting (B) has more advantages than the general methods in (A). The overall system has benefited from the advantage of the proposed methods/components. It has smaller image correction error, stronger noise resistance and better 3D reconstruction accuracy. By carefully selecting different components mentioned above and system optimization, our scanning system is able to achieve $0.15mm$ accuracy at close range scanning ($148 \sim 253mm$) underwater.

Since we have investigated different components, including image noise removal methods, corner detection, laser peak detection, camera calibration, and scanner calibration based on the real 3D scanning data. It can be observed from those results that:

1. To find out how the noise affect the image quality and our final 3D scanning result, a detailed comparative study of noise removal methods was carried out using the nature and real scanning images. From the result we can see the noise do have the heaviest impact on our 3D scanning result. It can be observed from Table 6.2, it has up to $2.8mm$ deviation of the three example measurements as the noise level increases. In Table 6.1, even the noise just increases by 0.1, the SSE and SOD errors increase by $61.12\%, 29.43\%$ after SCS denoise respectively. It is apparently larger to other components in our scanning system. In addition, noise has significant heavy impact on the performance of all the other components. For instance, it affects the accuracy of corner detection and peak detection. These inaccurate corners affects the camera calibration result. Such results and peak locations also affect the scanner parameters estimation and final 3D reconstruction. The experimental result shows Sparse Code Shrinkage method is only method retain the image feature without blurring too much and reduce too much image detail as the noise levels

increasing. It is because that SCS exploits the statistical properties of the data to be denoised in a more realistic and deals with data statistics.

2. Laser peak detection plays the second important role because it is directly related to the scanner parameter estimation and 3D reconstruction. Due to the fact that our scanner parameters D and α are not directly measurable, and the accuracy of such parameters estimation proposed in this thesis is mainly depended on the laser peak detection and calibration result. Hence, the error come from laser peak will be amplified in the final 3D reconstruction. In addition, these two parameters have directly affect to our triangulation based 3D reconstruction accuracy. For example, from Fig.6.30 we can see, as the laser peak detection error increases to 4 pixels, the final 3D may contain up to $8mm$ error based on our current system settings with fixed calibration parameters. Of course, the largest error only happens when the laser is very close to the image edge. All the rest points have much smaller 3D reconstruction error. Meanwhile, laser stripe saturation easily happens because of the scanning environment, laser profile, scanning object's surface and so on. If the laser stripe is saturated, for example, when the laser peak crosses over 3 pixels, all these methods start increasing detection error except FIR. It is because GA, CM, LIP, PE and BR only consider the maximum light intensity and the nearby three to five pixels to locate the peak in sub-pixel level. However, the laser row data is filtered by convolution and first order derivate in FIR method. Then, it is much easier to estimate the laser peak even in the highly saturated laser stripe.

3. Image correction and camera calibration also play very important roles and they are related to each other. We proposed a fraction lens distortion model(FMC) and iterative camera calibration in this thesis. Both of them significantly improved the camera calibration and image correction accuracy. In reality, we probably have little idea about what kind of the distortion the captured image is subject to and the actual distortion of the camera lens is not known and need to be investigated. FMC has the ability to model the overall distortion of the camera lens. From the experimental result, it is clear that fractional model is more powerful than the polynomial model in describing the distortions that the camera was subject to. In the other hand, 2D/3D coordinates of the control points sometimes are shifted due to lens distortion effect on the image feature detection or manufacturing process of the 3D pattern. The proposed iterative camera calibration method reduces errors from incorrect 2D/3D coordinates by (1) the corner detection run on distortion free images in each round; (2) it used the LM and bundle adjustment. Fraction distortion model and iterative camera calibration have similar effect on the final 3D result. It can be seen from Tables 6.17, 6.18 and 6.16, FMC distortion model significantly reduces the distortion effect on the 3D accuracy. For example, as the multiple noise increases to $\tau = 25$, the average error is $1.41mm$ by radial distortion, $0.61mm$ by radial + decentering distortion (Zhang-RD), and $0.42mm$ by FMC. Iterative camera calibration is able to reduce the error from an inaccurate calibration pattern. As shown in Table 6.30, as $\tau = 25$, the average error is lowered

to $0.25mm$.

4. The iterative camera calibration experiment simulated how the corner detection accuracy affect the final result within the noisy chessboard images. The fraction distortion model experiment also has similar simulation: the noise directly added to the detected corners. As the noise level increases to $\tau = 25$, the corners has maximum 0.5 pixel offset in the image plane. It can be seen that choosing different distortion models or calibration methods has heavier impact on the final 3D result than corner detector at the same noise level. Hence, corner detection has less impact compared to other components.

5. As all the components combined together, the final error became even lower. the optimization of each component does help the optimization of the whole system. From Table 6.34 and 6.35, it can be seen that the average error drops from $3.02mm$ in Setting (A) to $0.24mm$ in Setting (B). The rest error as a function of τ is also smaller compared to those in Table 6.30.

Chapter 7

Real World Scanning Projects

This chapter contains all the main scanning projects we did during the PhD study. The algorithms and methods are applied in real applications. Thus, the reliability, speed, or any other concerns need to be counted in. Compared with the experiments just in the lab, sometimes the real world problems are unpredictable. Thus, this real world scenario provides us with a great opportunity to test our products, algorithms, methods and programs. The feedbacks from the scanning jobs are one of the critical parts in our research and enable us to keep on improving our techniques and products.

7.1 Air scanning

The air scanning project includes one main project that we did on April 2011.

7.1.1 Mooring Chain Scanning - 2011

7.1.1.1 User Requirement

This project planned to scan some broken links of mooring chain and used for stress analysis to find out why they broke and how much the force was needed to make it break. Another purpose is to have these mooring chains' 3D models as a record in case that any damage occurs to them in the future and used for claiming insurance. The client required $0.05mm$ resolution and $0.5mm$ accuracy. Hence, the scanner set to be $0.045mm$ resolution to make sure the final data has enough density. Fig.7.1 shows the customized 3D laser scanners including the supporting metal cubic frame.

The scanner was placed at one side of the metal cubic frame. The mooring chains can be lifted up or down inside the frame by the crane. We can rotate the frame to have four scanning positions, and such scanning process is called vertical scan. To be able to have a wider view of scanning, we used dual laser scanner as shown in Fig. 7.1. In each position, the linear stage travels left and right (up and down in vertical direction), in this way it is able to cover about 180° scanning view. This project required the 3D models to be constructed as individual links in each chain. Thus, we need to have a horizontal scanning to capture inner sections on the top and bottom of each link. This

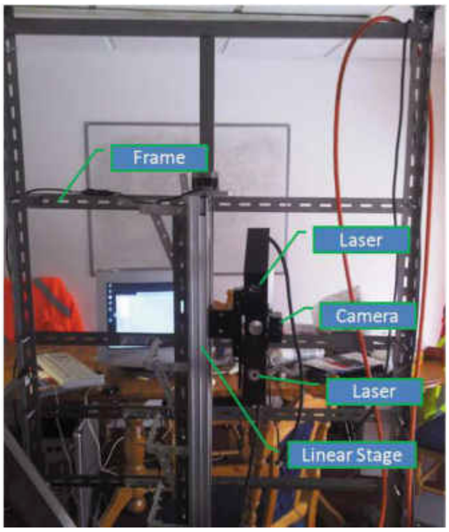


Figure 7.1: The customized mooring chain 3D scanner

was due to the fact that the links are connected (hooked) to each other. Each link's top and bottom were covered by the links next to it. They had to be laid on the floor with a particular pose to be able to scan the covered parts (covered by its neighbouring links). The shadow on left image on Fig.7.2 explains the covered parts and right image is the 3D model contains the covered parts. The scanner had a high speed mono ethernet camera and is able to capture up to 60 frames per second and two lasers (called left or right laser). After the horizontal and vertical scan, we can combine and merge all the 3D models from different position scans and create a single linked model of each chain and their individual links.

7.1.1.2 Some Issues

There are some issues during the scanning process: firstly, these mooring chains are very heavy, each link weighs over 80kg, and each chain contains over twelve links. Thus, every movement of the chains requires a machine's help and slows down our scanning process a lot. In addition, the horizontal scan requires the links in a particular pose in order to scan the hidden part. This means that only human force can adjust them in the pose what we are looking for. This also slows down the entire scanning process a lot even we had three people available to help. Secondly, the surface of the chain is very rough, the laser cannot reflect the enough signal back to the camera even the lens' iris is fully opened. We had to use a black curtain to cover up the entire frame and paint the chain on white (The client only requires the 3D model of the chains and colour is not an issue). Another problem is due to their huge weight, we have to use a lifter to lift it up and drag on the concrete floor. Every time we dragged the chain, the chain's surface including colour and marks has been slightly changed because of the friction from the

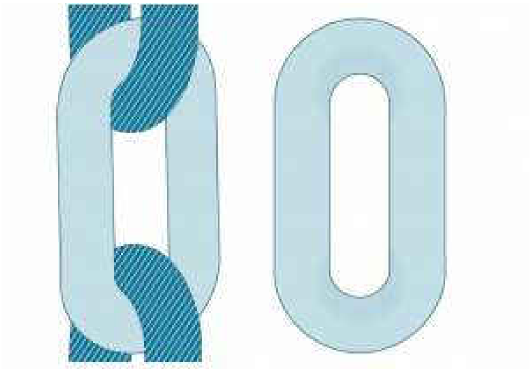


Figure 7.2: Hidden part (the shaded regions) of the mooring chain links.

ground and thus affects our 3D model surface. Of course, these slightly changed surfaces affect the final fusion of 3D models from different parts of the chain. However, such slight changes/damages to the surface are unavoidable and the client understood and accepted that.

Finally, some problems discussed in previous chapters happened in this project. For example, the angle λ discussed in Chapter 5.3.4 needs to be estimated due to the mechanical design. As shown in Fig.5.10, there is a small angle λ between the X-axis of the scanner and linear stage travel direction. Normally, this angle will be very small, even it can be ignored. In this case, because the stage was lifted up instead of lying on the floor or table horizontally. The position and weight of the scanner increased the angle λ and it needed to be counted in for better scanning accuracy. This error can be compensated by applying the Eq.5.3.13.

For the reason of confidentiality, the actual scanned 3D models on the field cannot be displayed in this thesis. Therefore, only some of links scanning done in the lab based on totally the same principle and procedure are shown in this chapter. Here is equipment set up: the linear stage is $986mm$ long, scanning step (linear stage moving step) is $0.045mm$, the capture frame rate is 60 FPS, the image resolution is 1400×1024. The camera has an ability to control the darkness, contrast, brightness, etc., which helps reduce the imaging noise very well. Fig.7.4 shows the scanning laser stripe example. The laser itself is a little bit thicker but image background is clean and dark enough. This laser stripe is thicker due to the saturation, however, FIR laser peak detection method described in 3.3 is still able to detect the peak in the centre well as shown in Fig. 7.5.

Fig.7.3 shows a demonstration broken link to be scanned. Figs.7.6 and 7.7 show its raw 3D models which were taken from four different views from the dual laser scans. Clearly, there are some noisy lumps coming from some un-wanted light sources or back-

Figure 7.3: A broken link to be scanned

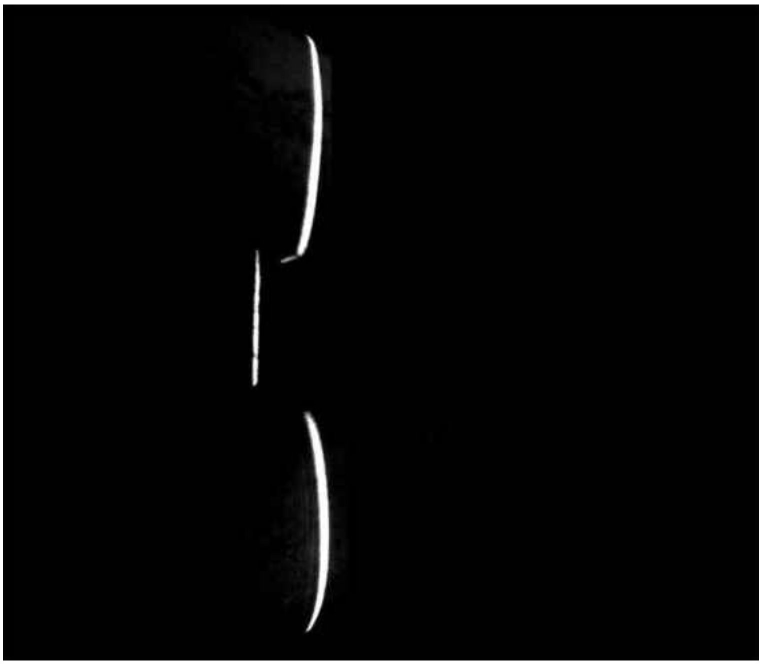


Figure 7.4: An example of the laser stripe of the mooring chain during its scanning

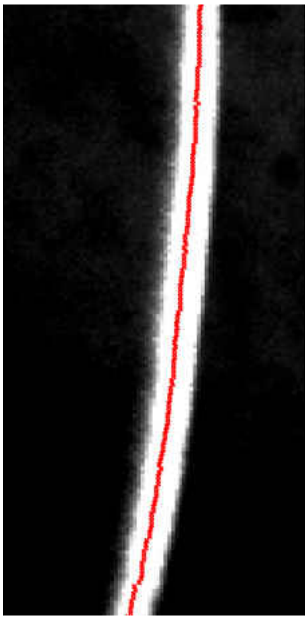


Figure 7.5: An example of peak detection on a laser stripe. Red: laser peak

ground in the 3D models and they can be deleted manually. These 3D models were adjusted manually firstly and then used global alignment to minimize the registration error. This process was done by third party software such as Geomagic Studio, Rapid Form. The challenge during this process is that some areas in the 3D model have too much noise and contain some un-wanted points which lead these multi-view registration results far away from expectation. However, some of these un-wanted points can be deleted manually and carefully. The noise may come from the rough object surface and background light source. Some unwanted noisy points were generated from the inaccurate peaks caused by too heavy laser saturation. When the intersection angle between the laser stripe and object surface was too small and the laser stripe becomes wider than expected (saturated). In such case, it may produce inaccuracy peaks.

Fig.7.8 shows this broken link's 3D model combining all eight scans. This demonstration scanning was done in the lab and only vertical scans were taken since there was no hidden part of this individual link. In our measurement, the deviation of this scanning is about $\pm 0.5mm$ compared to the direct measurement. In summary, the error may come from two parts:

1. The surface of the object is too rough and some tiny features cannot be picked up.
2. The laser's thickness. Obviously thinner laser stripe can pick up more details than thick ones. However, to generate strong enough laser beams for such low reflection surface(too rough), specialized lasers were required and consume much higher power. Hence, the laser thickness became higher. It may cause some tiny

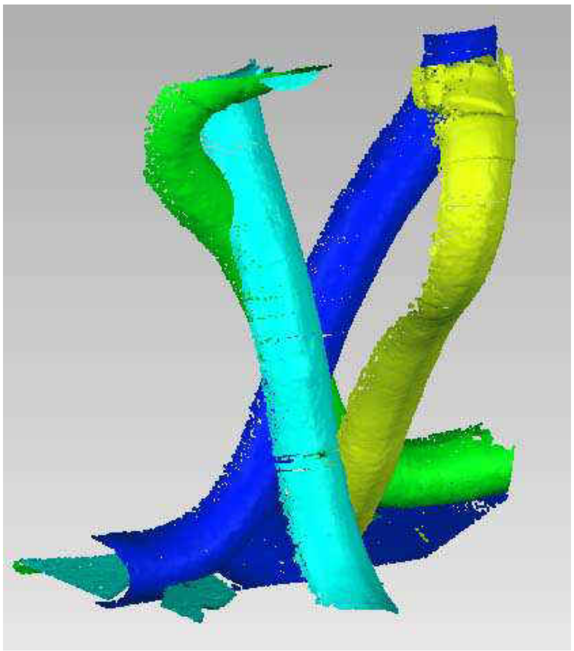


Figure 7.6: Four different 3D views of the mooring chain links from the left laser scanner

piece of the edge of the links not be picked up and the deviation may be slightly bigger.

However, our customer is satisfied with existing accuracy and resolution of our 3D model. It is understandable to scan objects with such rough surface and heavy weight in a very short period, the result is acceptable.

By scanning these mooring chains, our system was pushed to limit. The system ran about 10 hours per day, 5 days a week for 4 weeks. Some practical issues that we hadn't met before came up. For example, the workshop was open, windy and full of dust, which made our linear stage stuck for many times. It had to be cleaned before using again. Such problem has never happened in the lab. The generated data was huge. For example, in the vertical scan, at each position the left laser scanner has captured more than 200000 pictures and the generated 3D file size was over 400MB. The amount of data could be doubled if counting in right laser scans. It took a lot of time to process the captured data. For instance, it took about one hour to finish laser peak detection from just one position scanning file (200000 pictures in total) on a PC with Intel Core I7 3.20GHz using parallel computing method (actual detection rate is about $56f/s$, image resolution: 1400 × 1024). Therefore, it took up to five to six days just to finish one link, and there are over twelve links for each chain, eight chains altogether. We had not only just huge files stored on our hard drives but also applied our scanner to the real applications. It is a great opportunity to test techniques we developed and the programs we made. Luckily, our scanning was in air that eliminates a lot of hassle than

Figure 7.7: Four different 3D views of the mooring chain links from the right laser scanner

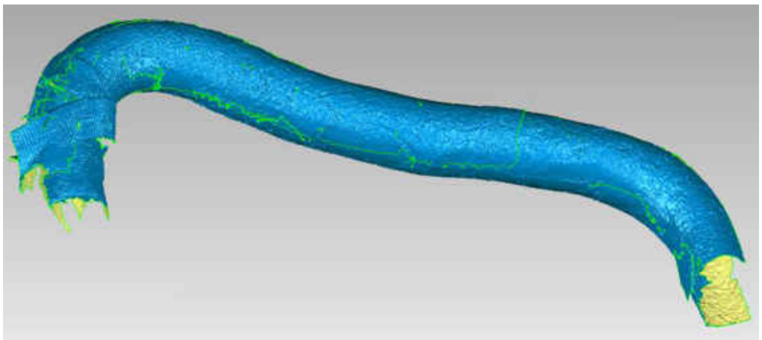


Figure 7.8: The 3D model of the broken link

scanning underwater. Furthermore, the mooring chain scan can be applied underwater, the biggest problem is how to deploy the tools, for example, positioning the mooring chain or claiming our scanner in the chain.

7.1.2 Well head Scanning - 2010

A well head is a general term used to describe the component at the surface of an oil or gas well that provides the structural and pressure-containing interface for the drilling and production equipment.[1]. Normally it is a bottom connector to Christmas tree which distributes the oil or gas. Thus, this well head is one of the most important parts in these oil and gas components. Anything wrong or even tiny leakage will lose money or even stop the whole production.

7.1.2.1 User Requirement

This scanning job is to build the 3D model for inspecting the damage to the well head and the inner thread. Fig.7.9 shows the well head we need to scan. This metal bar on the top of the well head holds the rotation stage and the scanner. The rotation stage will be placed on as much centre of the well head as possible, which make the rotation centre overlap with the centre of the well head.

Figure 7.9: The well head to be scanned

7.1.2.2 Design and scanning

Since the demo scanning is in air and we don't have to worry about how to build the waterproof rotation stage. Otherwise, it will cause a real big problem. The scanner set up is as follows: the image size is 720×576 and the scan step is about $0.1°$ degree. The diameter of the well head is 890mm. A 5p, 10p, and 50p coins on the inner thread is used to check if the scanner can pick them up and their measurement accuracy.

---

[1] http://en.wikipedia.org/wiki/Wellhead

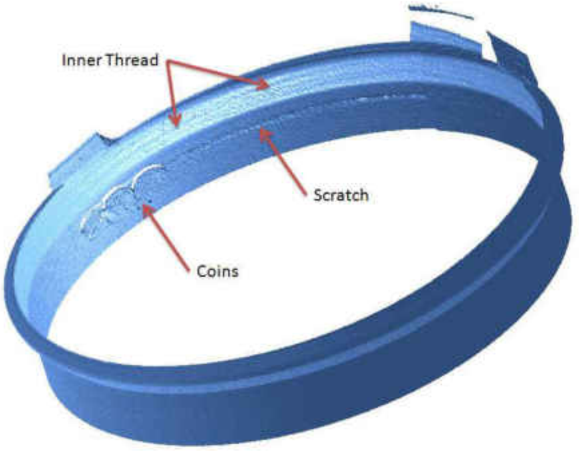
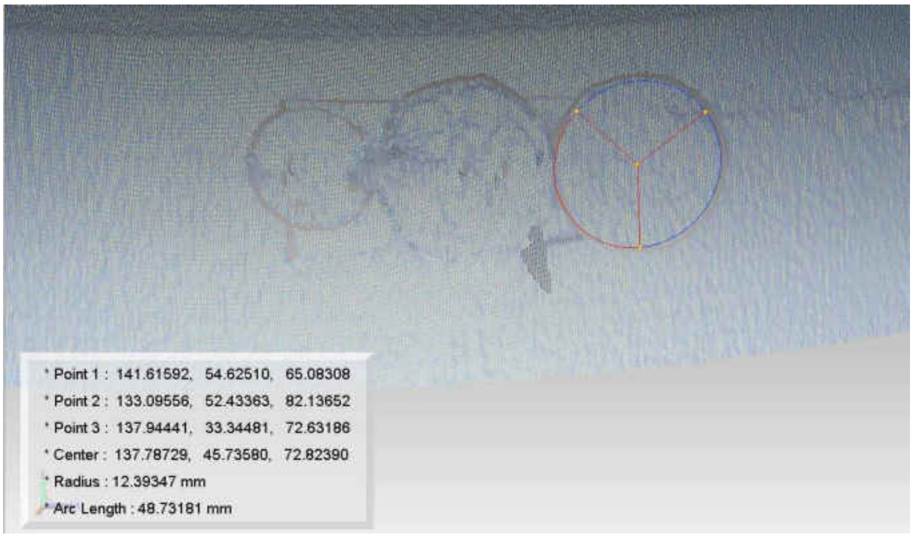


Figure 7.10: The 3D model of the well head



```
* Point 1 :  141.61592,   54.62510,   65.08308
* Point 2 :  133.09556,   52.43363,   82.13652
* Point 3 :  137.94441,   33.34481,   72.63186
* Center :  137.78729,   45.73580,   72.82390
* Radius : 12.39347 mm
* Arc Length : 48.73181 mm
```

Figure 7.11: The measurement of the coins in the 3D model of the well head

Fig.7.10 shows the 3D model of this well head. The inner thread and coins can be clearly seen on the 3D model. Fig.7.11 shows the measurement of this pound coin's radius and circumference measurement. As the figure shows $R = 12.39347mm$, which is about $0.14mm$ off from $R = 12.25mm$, which is the radius of a real 10p coin. [2]. In this case, the relative error is 1.18%. It is well over the estimate since this just a demo, the equipment is not well designed particularly the rotation stage, it is just a simple gear motor. To improve the reflection properties of the well head, it was painted with high refraction paint. This painting process thus created some scratches as shown in the 3D model in Fig.7.10. Fig.7.12 more clearly explains the noise and reflection during the scanning. However, this is demo scanning, all the equipment set-up was for demonstration purpose, and even the scanning well head itself was designed for that. Therefore, the painting and refraction issues can be solved in the field. It was a pity that this project didn't run after the demonstration because of customer's tight financial budget in 2010.

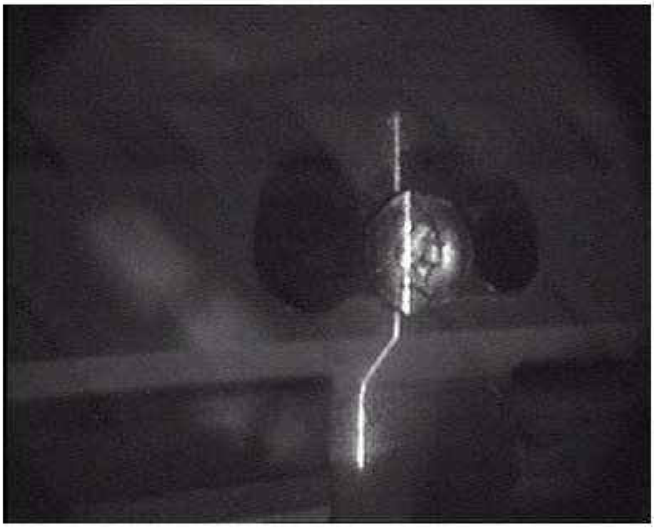


Figure 7.12: The scanning noise and reflection of the well head

7.2 Underwater Scanning

This is another part of the real scanning projects but in underwater environments. Compared to the previous projects operated in the air, scanning an object inside water is really challenging. The main difficulty lies in that the index of the water is completely different from that in the air. Since the air is lightweight, transparent and the refraction index is so small, thus the bending of light travel through the air will normally not be

---

[2] WiKi: Ten pence (British coin) http://en.wikipedia.org/wiki/Ten_pence_%28British_coin%29

considered. However, underwater is different since the light is bent, and different area's water has slightly different indexes that imply that the bending angles will be different. In addition, the different wavelength of light will change the bending angle of light. The complex environment of underwater, such as temperature, salinity and visibility will also affect the final results. Thus, it is a great challenge for us to transform our scanner from the air to underwater.

7.2.1 Laser stripe in water

To demonstrate the laser scanning operates underwater, some images, which come from real scanning, are shown and different circumstances will be explained. The laser stripe in Fig.7.13 is too thick, saturated and has a lot of refraction. It is still able to detect the peak of this laser stripe but with very poor accuracy and rough 3D looking.

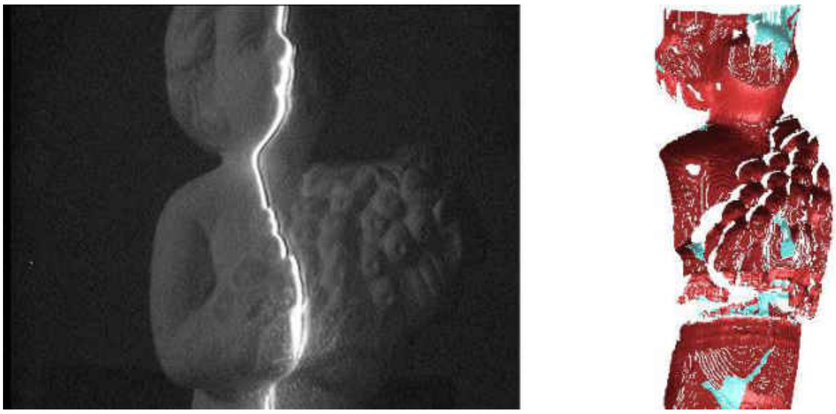


Figure 7.13: Left: Too thick and double layer laser stripe; Right: its 3D profile (Scan step: 0.25mm/s)

Fig.7.14 shows a significant improvement than the last result in Fig.7.13 after increasing the scanning step and adjusting the laser thickness. This laser beam is much thinner, focused and without double layer. Its 3D profile is smooth and detailed. Apparently, smaller moving steps and thinner laser stripes have a huge effect on the final 3D scanning results. Hence, it is believed that the scanning accuracy depends on different object sizes, requirement of resolutions, the laser stripe width and moving steps. To scan the surface of small objects like one pound coin, this laser stripe should be very thin and moving step should be $<= 0.1$mm/s. On the contrary, to scan large objects, high power laser, high resolution camera are needed and the moving step can be as large as $1 \sim 5$mm/s.

Actually, from these two examples, it can be seen that there is still a lot of noise if the image is enlarged. To be able to capture laser stripe by mono cameras, all the other light sources should be switched off to reduce the noise. Adjusting the exposure time, ISO value and darkness can help to lower down the noise level. Unfortunately, most of high speed cameras don't contain manual control of CCD's exposure time and ISO

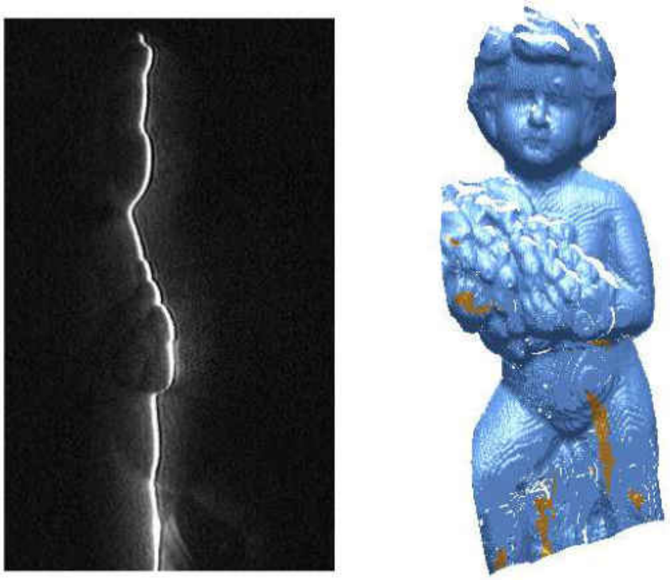


Figure 7.14: Left: Thinner laser stripe; Right: its 3D profile (Scan step: 0.1mm/s)

value. The capture speed of those coming with full manual control and high resolution camera such as Nikon D300 is just three frames per second at full resolution. Hence, there is always conflict between capture speed and noise in laser scanning applications.

On the other hand, these images from laser scanning always include high contrast, which means the laser power cannot be too high or too low. Within a certain distance, too high power laser will lead the laser stripe to be too thick and over contrasted. Too low power laser will lead it hard to separate the background noise and laser stripe because the reflection light from the object of interest is too low. However, by carefully selecting the equipment and remoulding, our laser scanner is successfully applied to real world applications with about $\pm 0.5mm$ accuracy or even lower depends the scan distance and laser projection angle. The detail of these projects will be given in the sub-section below. These scanners have already been used for Oil&Gas industrial inspection.

7.2.2 Reducer Scanning - 2009

This damaged reducer is part of the cooling system in a nuclear power station. The project intended to generate the 3D model for inspection and build a replacement reducer based on the damaged part. The reducer is a cylinder with one end bigger than the other and filled full of pure water. Fig.7.15 shows the scanners designed to work for this reducer survey. It is a customized scanning system that contains the laser, digital camera, motion sensor, rotation and linear stage. The scanner is mounted at the front of the rotation stage that is then connected to the linear stage.

7.2.2.1 User Requirements

The generated 3D model is for damage inspection. The client required precise 3D model and measurement analysis. Since the reducer has a cylindrical shape with unequal ends then the analyses including the 3D centre line of this cylinder and radius of each cross-section. With this centre line and radius, it is able to compute each cross-section's ovality. Such cylinder shape object also can unwrap/open to be flat model and generate contours image which displays the damage sections/parts intuitively.

7.2.2.2 Design and scanning

It is impossible to scan the whole pipe in one go. Hence, the system is designed to work like this: the rotation stage scans one session by 360°, then the linear stage drives the scanner forward and rotates again to have another session. In total, there are six sessions to be scanned. The 3D coordinate(X, Y, Z) are taken from the following steps:

1. **Y** is the step size (in mm) of arm's moving direction
2. Assume $P_i(x_i, y_i)$ is the peak location in the image, and R_x, R_y, R_z is the rotation angle in XYZ axis. In this case, R_y is set to the rotation axis, R_x, R_z are set to zero. Each step of R_y is 0.1°. Hence, the peak location $P_i(x_i, y_i)$ in 3D $P_w(X_i, Y_i, Z_i)$can be computed by the camera calibration parameters and scanner parameters. Please note that $X_i = 0$, and Y_i is the step size, Z_i is the distance between the pipe's wall to the camera lens.
3. Once we have P_w, it can be transformed to the world coordinate system P'_w by the rotation matrix R as show in Eq. 7.2.1

$$[R]P_w = P'_w \tag{7.2.1}$$

There are some problems happening when we did the scanning:

1. The rotation arm was not stable when it turned in the water because the arm itself was long (about 1 meter) and heavy. The scanner mounted on the arm's end also increased the entire arm's weight. Because of weight force, it was hard to make sure that the arm was always staying at the same rotation center during the scanning and it was a little bent down.
2. Even we adjusted the rotation arm and scanner carefully, it was hard or even impossible to make sure that the arm was at the center of the reducer during the scanning process. Because of that, in the 3D reconstruction, different sessions would overlap and the session itself did not form a perfect circle. Fig.7.17 shows a mismatch session in 3D display. It means that the rotation center points had been shifted during the 360 degree rotation. That was caused by the camera/arm shift or the arm doesn't place at the centre of the scanner as Fig.7.16. The error can be corrected by spiral function. Fig.7.18 shows the overlap parts between two

sessions. As we can see, there are two layers. The only way to solve those problems is to make a function which can adjust the session in X, Y, Z directions.

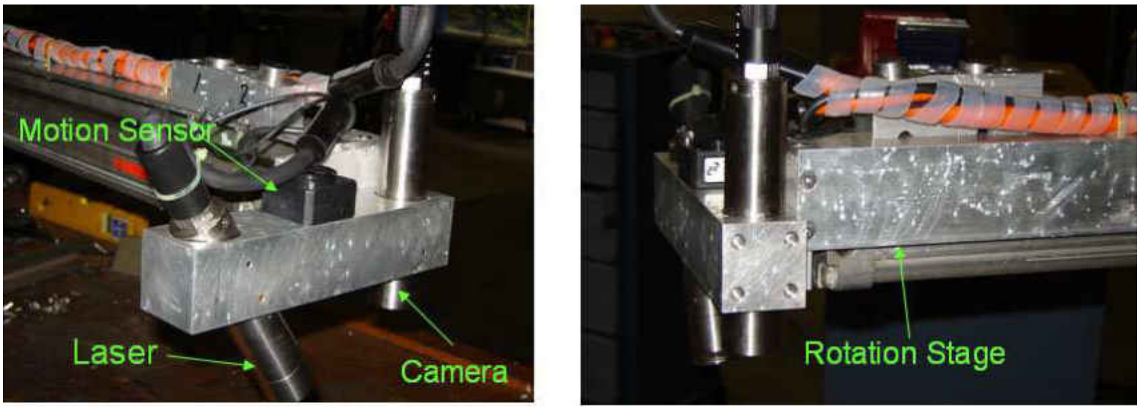


Figure 7.15: Front part of the scanner(a)

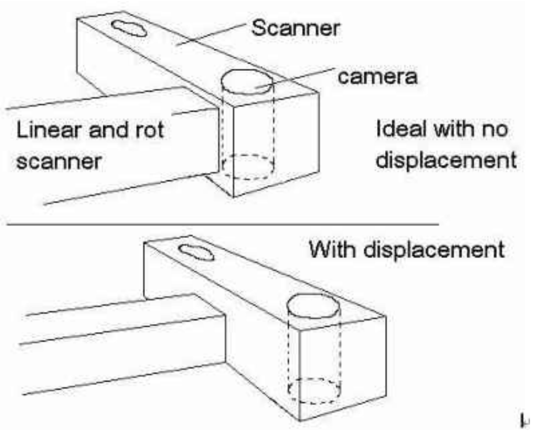


Figure 7.16: Front part of the scanner(b)

All the individual sessions were joined together as a complete reducer after the scanning finished. Fig.7.19 shows different sessions which are marked by different colours (it is not the real model for the work due to confidentiality, but it is similar). Fig.7.20 shows the parts of the final 3D model. It can be seen that there are some indentations on the bottom of the pipe. From the 3D model, the object's size, the indentation's location and shape can be extracted. A contour diagram of the pipe can be built. Fig.7.21 shows the contour diagram of the 3D model in Fig.7.20. The left hand side shows the scale of depth, the blue areas in the figure shows the indentations of the pipe and the colour variation indicates the changes in depth.

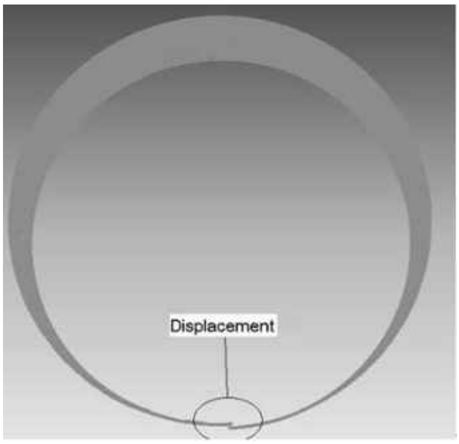


Figure 7.17: Mismatch session

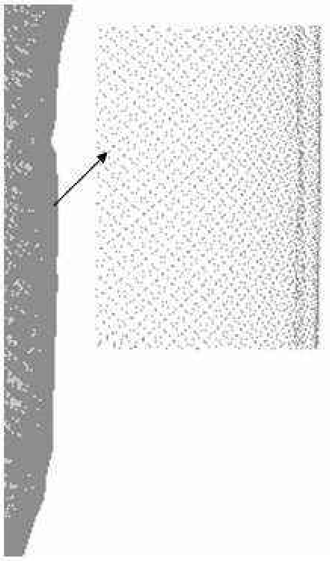


Figure 7.18: A snapshot of the mismatch session

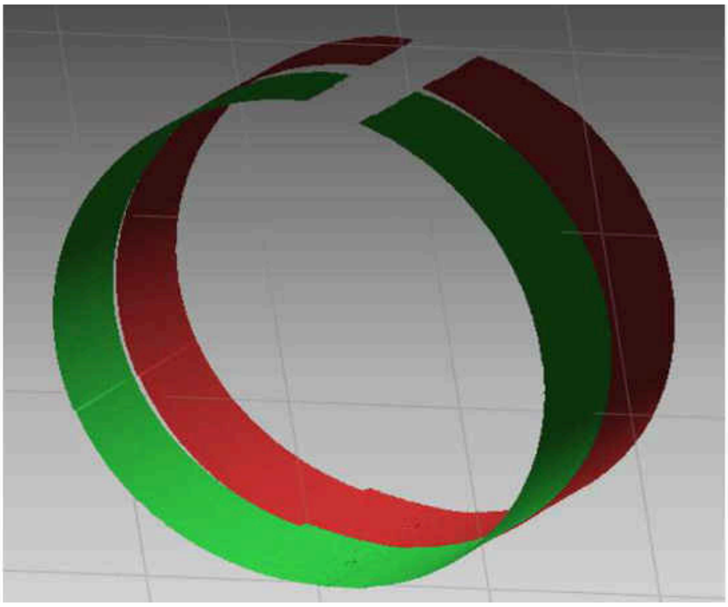


Figure 7.19: Joined 3D reducer models together. Various colours represent different sections

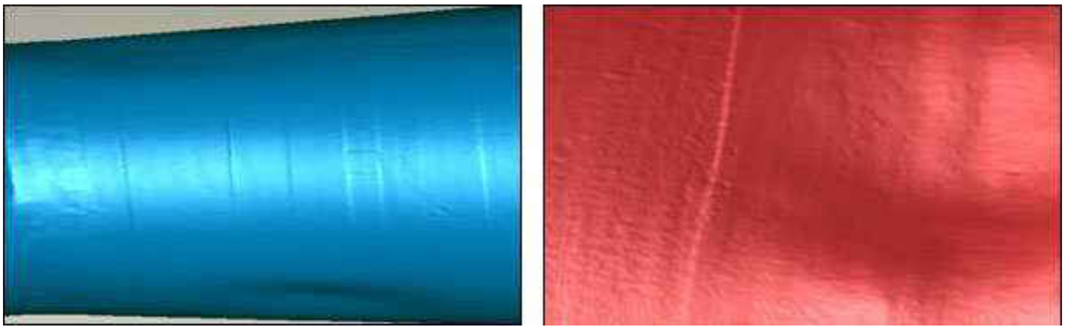


Figure 7.20: Contour diagram. Left: The surface of a scanned pipe. Right: Zoomed image of the damage/corrosion

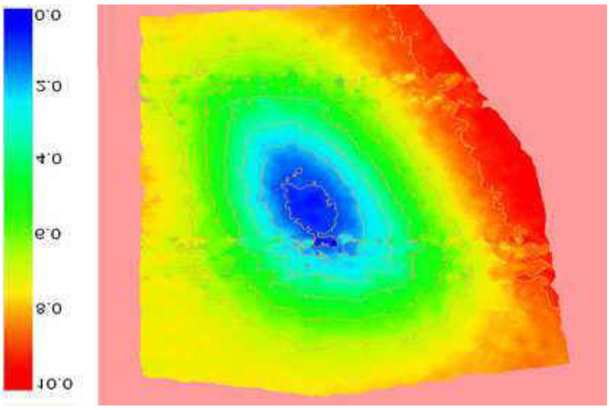


Figure 7.21: Contours highlighting the damage/corrosion

7.2.2.3 Measurement Analysis - Radius and Centre Line

The reducer is columnar. In order to know how much the reducer bends and the shape has been changed compared to the original design, it is necessary to compute the 3D centre points. The difficulty comes from three main parts:

1. To be simplified, the reducer (columnar) is formed of millions of cycles in Y direction. Every $0.1mm$ step can be formed as a cycle. For instance, if $Y = 200mm$ the 3D points in cross-section which has $|Y_i - 200| <= 0.1$ can be treated as one cycle since they have very similar Y coordinates and almost planar. Nevertheless, the numbers of points in different cycles are different. In addition, these points do not lie exactly on the same ring, see Fig.7.22 and Fig.7.23.
2. In the connection parts, there is some overlap, also, the overlap would be happened in the whole ring or just some parts of it as shown in Fig. 7.18. Due to that, the different layers would have different centre points.

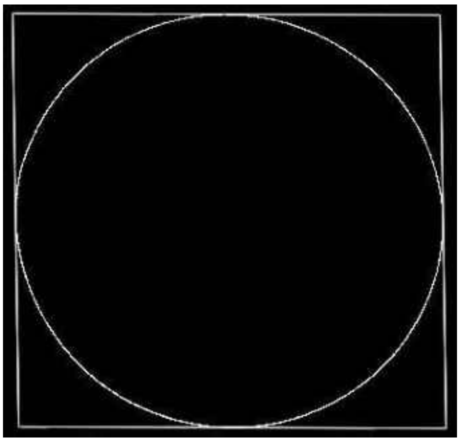


Figure 7.22: One of the crossing section of the reducer

Figure 7.23: The points on the crossing section

Thus, to compute a precise centre line, we need to consider the frequency of the centre points in all layers. The centre points can be computed as shown in Fig.7.24: Two lines can be formed through two pairs of the three points, the first passes through the first two points P_1 and P_2. Line b passes through the next two points P_2 and P_3. The equations of these two lines are

$$y_a = m_a(x - x_1) + y_1 \quad and \quad y_b = m_b(x - x_2) + y_2 \tag{7.2.2}$$

where m is the slope of the line given by:

$$m_a = \frac{y_2 - y_1}{x_2 - x_1} \quad and \quad m_b = \frac{y_3 - y_2}{x_3 - x_2} \tag{7.2.3}$$

The center of the circle is the intersection of the two lines perpendicular to and passing through the midpoints of the lines (P_1, P_2) and (P_2, P_3). A line perpendicular to a line with slope m has slope $\frac{-1}{m}$, thus the equation of the lines perpendicular to lines a and b and passing through the midpoints of (P_1, P_2) and (P_2, P_3) are

$$y'_a = -\frac{1}{m_a}\{x - \frac{x_1 + x_2}{2}\} + \frac{y_1 + y_2}{2} \tag{7.2.4}$$

$$y'_b = -\frac{1}{m_b}\{x - \frac{x_2 + x_3}{2}\} + \frac{y_2 + y_3}{2} \tag{7.2.5}$$

The three points should be picked with large angles. In other words, we cannot choose three points that are close to each other in 3D space. The best instance is that the three points form an equilateral triangle. For each cross section, there are many points, which can generate multiple centres. Some of them are far away from the centre point cloud because of the noise, thus, they will not be considered, only the most dense centre points and their average are chosen for the actual centre. After we got the centre points in the same cross section that has multiple layers. The multiple layers cause multiple centre points, in this case, the point with the highest frequency of occurrence is chosen as the 3D centre point of this cross section.

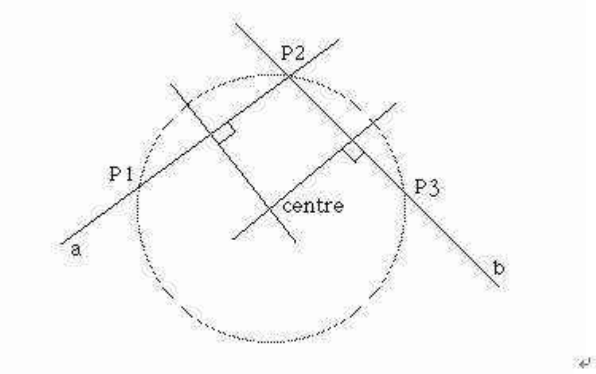


Figure 7.24: Centre of a circle

7.2.2.4 Measurement Analysis - Ovality

It is able to compute the ovality of this cross section by the center points. Fig.7.26 shows one of the examples. To be simplified, the maximum and minimum radii are considered the same, that means it would be a circle but actually, it is not. As a reference solution, the property of the ovality would be given in our project including the maximum radius and its perpendicularity radius. If the reducer is a column, then they should be equal. If not, that means that this section has aberration. Fig.7.25 is the short radius in ellipse measurement of the entire reducer in absolute distance. The green line is ideal radius and the blue is the real. Clearly, there is some variation between the real and original design, this ovality calculation gives a numerical way for the measurement and inspection.

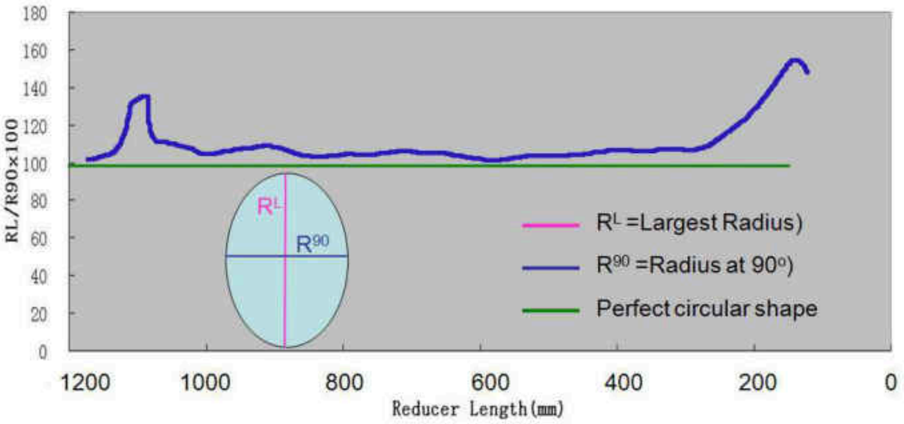


Figure 7.25: Ovality measurements of the Reducer

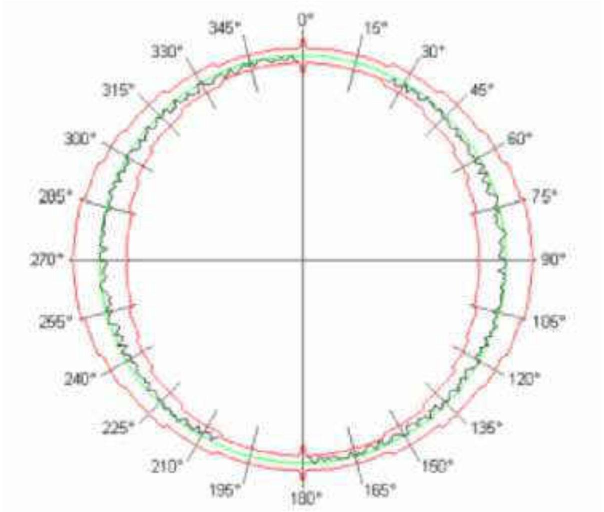


Figure 7.26: Cross section's Ovality

7.2.2.5 Measurement Analysis - Unwrap the 3D reducer (Column)

In order to present the shape aberration of the reducer, the cylindrical shape 3D model is un-wrapped ("opened") to a flat model and paint in colour according to the surface's depth. It is easier to find out the aberration and measuring from this diagram. Fig.7.27 shows the unwrapped example and Fig.7.28 shows the contour image from the flat model. The deep red parts would be the bumps of the reducer that have higher deviation compared to the entire surface. This kind of information is what our customer was looking for. With these diagrams and 3D models, they can easily find out the weak parts of the reducer and repair them or even replace them with a new one.

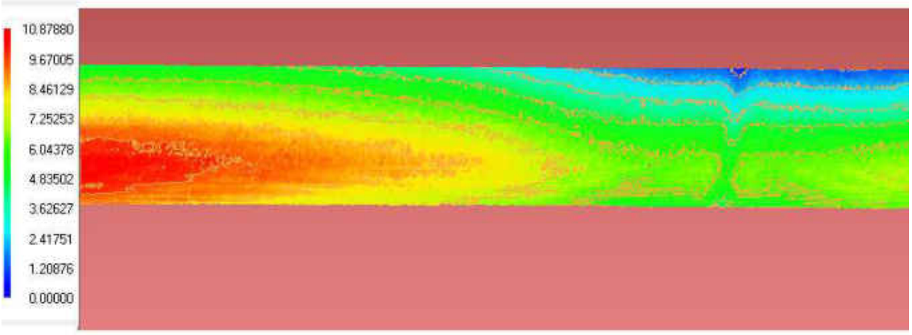


Figure 7.27: Reducer's contour extraction

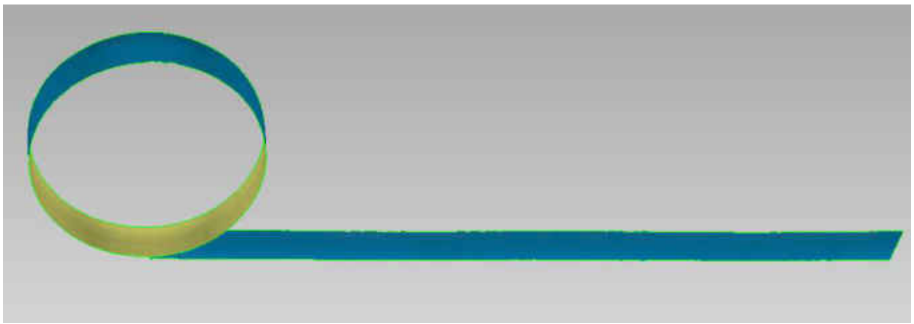


Figure 7.28: Reducer's unwrapped

The basic steps of the un-wrap computation are as follows:

1. Compute the centre line of the whole reducer
2. Compute the maximum and minimum radius of the whole reducer
3. Compute the angle α between the maximum and minimum radii displayed in Fig. 7.29
4. Use this angle to compute the points in that cross section's arc length. Fox example, point A in cross section i, we know the radius and centre by the pre-calculation,

then we need to determine whether the angle β is larger than 180 or not. As shown in Fig.7.30, to compute the angle with known three edges, the equation is:

$$cos(\beta) = \frac{c^2 + a^2 - b^2}{2 * a * c} \tag{7.2.6}$$

then, the angle is $\beta = \arccos(cos(\beta))$

5. When the angle is known, then circumference of circle can be computed directly. Once we know the circumference that the circle can be un-wrapped.

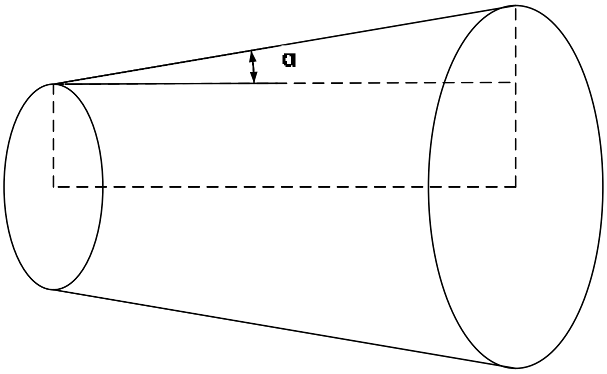


Figure 7.29: Angle between maximum and minimum radii

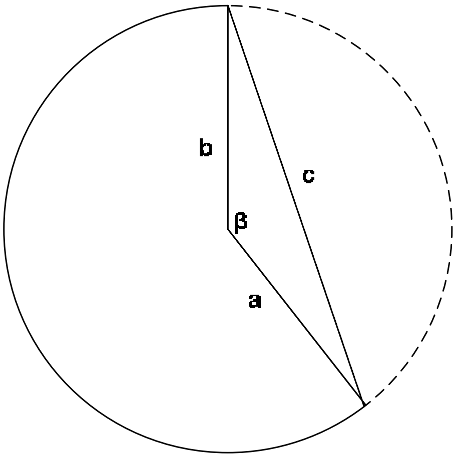


Figure 7.30: The arc needs to be calculated

7.2.3 Pipe's surface Scanning - 2010

This is an application required to mount the scanner on a Remotely Operated Vehicle (ROV) in order to scan a damaged pipeline's surface.

7.2.3.1 User Requirements

The scanner composed of one laser source and one digital camera has a limitation in this application, since it can only see about half of the required surface of the object in Fig.7.32. It is impossible to scan the entire surface in one go. If so, then the scanner must be placed much farther. In addition, the ROV always floats underwater whose stability is not good enough. To reduce the effect of the movement of the ROV and have a larger area of scanning profile, the quick scan should be performed. To this end, a dual laser scanner is designed, illustrated in Fig.7.31. In this design, the left and right laser sources have a field of view of around 90° respectively. The left and right profiles captured are combined together to acquire about a 180° profile in one scan.

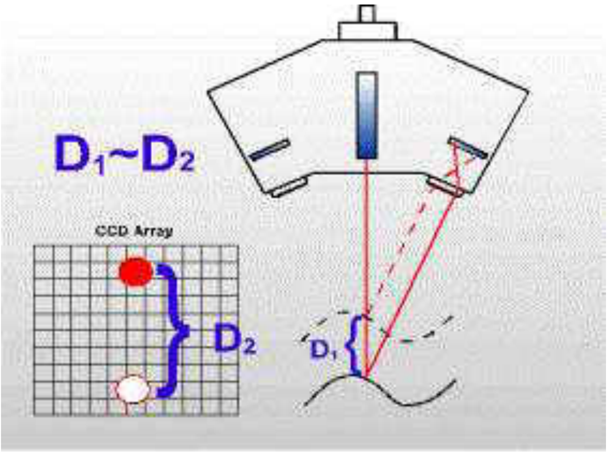


Figure 7.31: A model for dual laser scanner

7.2.3.2 Design and scanning

Fig.7.32 shows a prototype of the dual laser scanner which is under test underwater. It is mounted on a hydraulic linear stage to scan the white pipe in front of it. That mimics the situation running in subsea. The scanning purpose is to get the profile of the bending/damaged part of the pipe. Fig.7.34 shows the final design of the dual laser scanner. To reduce the refraction from the laser under water and front glass, the shutter is designed to only allow the central laser line to pass. As we discussed in the previous chapters, temperature, salinity, water index, etc., were all taken into consideration and simulated in the tests. Fig.7.33 shows one of the 3D profiles after scanning. The measuring radius of the hole at the top left is $5.19mm$ and its original radius is $5.0mm$ with $\pm 0.05mm$ tolerances. In this case, the measuring error is controlled below 4% and our customer was happy with this accuracy.

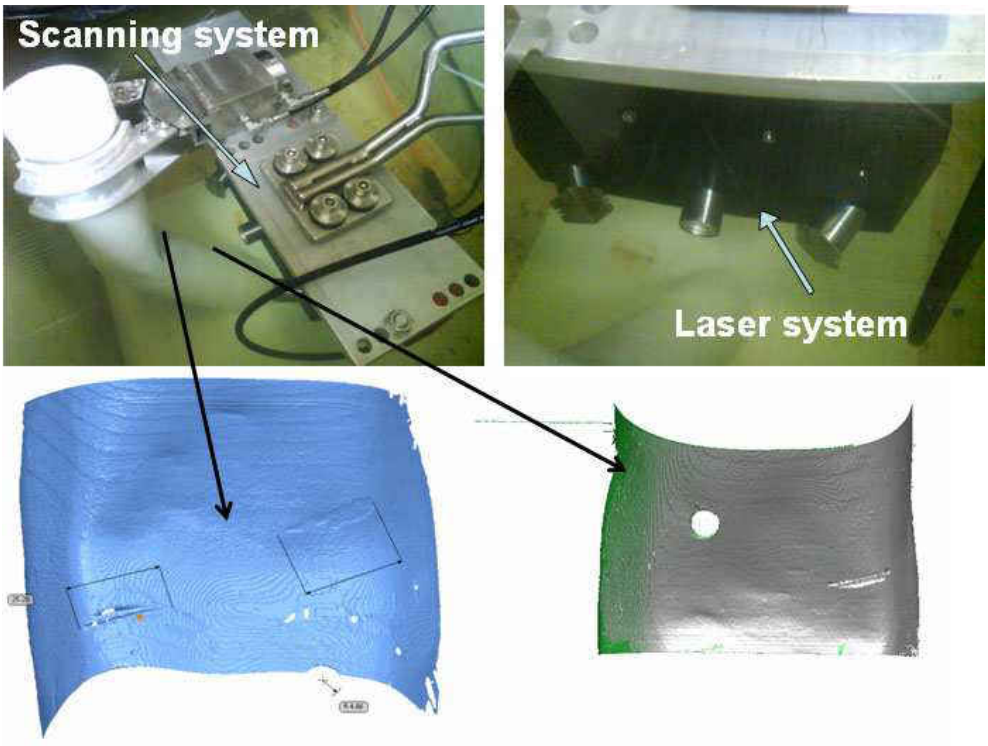
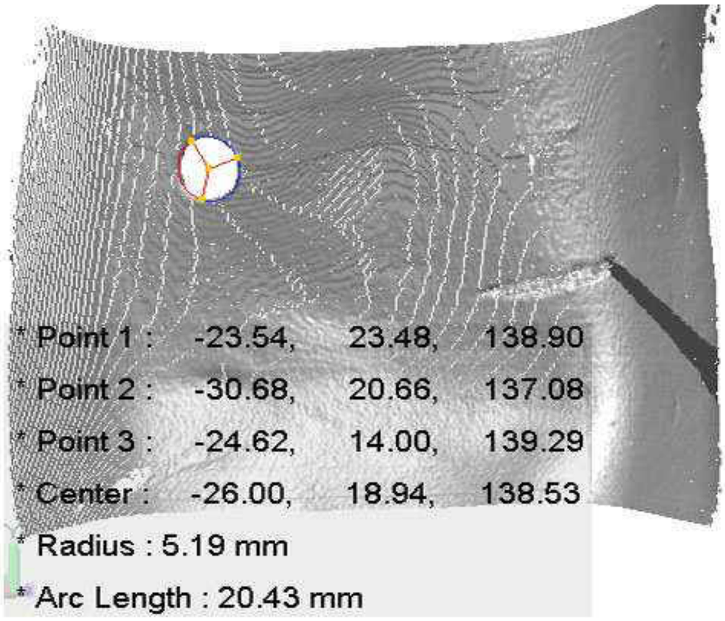


Figure 7.32: The prototype of dual laser scanner under testing. Top row: the scanner and the damage/bent part need to be scanned. Bottom row: The 3D model of the surface



```
* Point 1 :   -23.54,    23.48,    138.90
* Point 2 :   -30.68,    20.66,    137.08
* Point 3 :   -24.62,    14.00,    139.29
* Center :    -26.00,    18.94,    138.53
* Radius : 5.19 mm
* Arc Length : 20.43 mm
```

Figure 7.33: 3D profile and measurement

Figure 7.34: The final design of the dual laser scanner

Chapter 8

Conclusion

We have successfully built a workable 3D laser scanning system by the end of the period for my PhD study. It involves a lot of technical details, such as image processing, camera calibration, mechanical design, components/equipment selection and installation, and eventually applying the developed system in real world applications. All these small aspects in the developing process of this 3D laser scanning system require a wide range of knowledge about electronics, mechanical control, optics, programming, and so on. Their configuration and subsequent data processing also affect the final 3D reconstruction accuracy.

For the image processing, I implemented the Sparse Code Shrinkage image de-noising method which is more suitable for our laser images rather than the normal de-noising methods like Gaussian noise removal. The only disadvantage is its processing speed. However, in most of our cases, we have enough time to process our data and then this technique is still good to apply. Meanwhile, to choose the best corner and laser peak detection method, the comparison and evaluation study is performed between the most well-known and modern methods, only the best fit methods have been chosen in our laser scanning system.

To reduce the cost of building the scanner, the iterative method is proposed for accurate camera calibration from a low cost and not very precise calibration pattern. There are two key components in this method: (1) The world coordinates of the chessboard pattern are considered as unknown variables and re-estimated in the global optimization; (2) The feature (corner) detection is affected by lens distortion, it is much more accurate if the detection performs on distortion-free image. Hence, to reduce such effect, the corners' locations are re-detected on the undistorted image again once the current calibration parameters are obtained. By this procedure, even with an inaccurate pattern board with up to 10% off in absolute distance, the calibration results maintain stable.

Laser stripe projection angle and baseline distance from the center of lens to laser are the most important parameters in our triangulation based 3D scanning system. However, they are difficult, even impossible, to measure directly due to the fact that the center of camera lens and the laser source are extremely difficult to localize. To combat this difficulty, combining the camera calibration parameters and precise linear stage, a new approach is proposed to estimate these parameters which only require at minimum three movements of the entire scanner and capture three laser images. Of course, repeating this approach will increase the estimation accuracy.

During the entire 3D scanner development, practical issues arise either from those imperfect mechanical designs or extremely hard to control hardware, such as the position of laser light generator, laser stripe tilt, scanner block tilt relative to the linear stage, handling extremely large numbers of images, data size, cable connection etc. The entire system is built from scratch and applied in the field. This thesis addressed and proposed workable solutions for these issues. By the end of the research period, our scanner system is able to achieve $\pm 0.05mm$ accuracy in $35 \sim 60cm$ scanning distance in air (with a precise linear stage and 1400×1024 resolution camera) and $\pm 0.15mm$ accuracy in $14.8 \sim 25.3cm$ scanning distance underwater (with a linear stage and 720×576 resolution camera). However, when utilizing the higher resolution camera, thinner laser stripe and close scanning distance, the accuracy can be even higher. Current 3d scanner set up and accuracy is acceptable and satisfies the requirements of our applications in oil and gas industry which demand accurate object scanning.

It is possible to apply our 3D laser scanner for underwater objects. The most important issue is the index of water different from the air. The light path is from water to the camera front sapphire glass, and then to CCD. Obviously, when the light goes through the sapphire glass, it bends by a very tiny angle which is depending on the thickness and flatness of the glass. This angle is so tiny that can be ignored in the calculation normally. In our case, the light travel from water to air and from air to water is bent. This is the main source of error for underwater scanning when directly using the air camera calibration methods. The water index is affected by many variables like temperature, salinity, impurity etc., hence, the system calibration must be performed in the identical water as in the field. By careful mounting the scanner and controlling the linear stage, the scanning error can be minimized down to $0.2mm$. This project has a clear real world application in mind. Thus, it especially emphasizes the development of a workable system, rather than just the advance of science involved. All the equipment, software, technology we implemented have to be taken into account and can be tested and applied in the real world. In these commercial applications, they all have different requirements and purposes. It is much easier for us to evaluate our scanning system and have feedback during these projects. The problems are discovered quickly and fixed in the field and make our scanner more stable, compatible and more accurate. Further research improves the scanning system. A number of real world projects clearly show that the developed two generations of 3D laser scanning systems have been successfully applied to scan the real world objects with sufficient speed and accuracy. On the whole, the project has achieved the objectives set up.

In conclusion, the main contributions can be summarized as:

1. A comparative study of noise removal methods.
2. A comparative study of corner detection methods.
3. A comparative study of laser peak detection techniques.

4. Have proposed a novel camera distortion model.
5. Have proposed a novel method for estimation of the laser projection angle and the distance between the laser source and the camera center.
6. Have proposed a novel iterative based camera calibration method to compensate the error from inaccuracy 2D and 3D coordinates of the control points.
7. Carried out a comparative study of effect of various noise removal, corner detection, laser peak detection, camera distortion model and camera calibration methods on the accuracy of 3D reconstruction.
8. Have optimized the components of the laser scanning system for better accuracy.
9. Have developed two prototypes of laser scanners with different designs.
10. Have applied the developed 3D laser scanners to scan the real world objects and solutions have been proposed to the various issues arising from these applications.

There is a huge market for 3D scanning for inspection and quality assurance in the gas and oil industry, for example, gas/oil pipe inner/outer surface damage inspection and oil platform structure analyses. However, the 3D scanners developed in this market are limited and expensive. This is mainly because it is challenging for the development of underwater 3D scanners, since it involves a large number of issues which may be unpredictable and whose impact is not yet known exactly. Due to the fact that the seawater has different absorption rates for lights with different wavelength and generate blurring effects, which require image enhancement to correct the captured images. All those lead to difficulty in applying the colour laser in our underwater laser scanning systems. Whereas our underwater 3D scanner uses the green laser, which has a lower absorption rate in seawater and will reflect most of the light. Together with the high resolution underwater mono camera, our scanners have successfully been applied in our real underwater scanning projects and achieve $0.2mm$ accuracy. Our scanners have been applied to several projects in Norway, Dubai, and Aberdeen. The scanner must be adjusted to fit for those projects' special requirements due to the variation of the scanning object's size, shape, environment or the deployed equipment. Sometimes it is very hard to deploy our tools on the scanning object since everything is underwater and diver, hydraulic systems are necessary. The scanning object size and shape vary every time which require a special support holder. More analysis of underwater scanning results and interaction among various components inside the system should be taken such as the underwater calibration, image enhancement, and compensation for the light bending angle when it travels between water and air. Research is underway and results will be reported in the future.

Chapter 9

Appendix

9.1 Publications

Publication notice: The following papers have been published. The research and the writing of these papers were performed within the Department of Computer Science at Aberystwyth University and Smart Light Devices, Ltd. as an integral part of the PhD programme of studies under the supervision of Dr.Yonghuai Liu, Professor Reyer Zwiggelaar and Dr Ala Al-Obaidi.

1. Liu, Y.; Al-Obaidi, A.; Jakas, A. & Liu, J. Jain, L.; Wu, X.; Liu, H.; Gu, D.; Howlett, R. J. & Liu, Y. (Eds.) A Fraction Distortion Model for Accurate Camera Calibration and Correction. Robot Intelligence, Springer London, 2010, 169-190

2. Junjie Liu, Anthony Jakas, Ala Al-Obaidi, Yonghuai Liu. Practical issues and Development of Underwater 3D Laser Scanners. Proceedings of 15th IEEE International Conference on Emerging Technologies and Factory Automation, 13-16 September 2010.

3. Junjie Liu Anthony Jakas, Ala Al-Obaidi, Yonghuai Liu. A comparative study of different corner detection methods. Proceedings of The 8th IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA2009), December 15-18, 2009.

4. Yonghuai Liu, Ala Al-Obaidi, Anthony Jakas, Junjie Liu. Accurate Camera Calibration Using the Collinearity Constraint. Proceedings of The 8th IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA2009), December 15-18, 2009.

5. Junjie Liu, Ningfeng Wei, Yonghuai Liu. Accurate camera calibration for 3D data acquisition: a comparative study. Robot Navigation: Strategies, Algorithms and Motion Planning, Editor: Daiki It, Nova Science Publishers, Inc. 2008.

9.2 Programmings

9.2.1 Parallel computing

Nowadays, multi-cores CPU and Solid-state drive make the image capture and processing much faster than before. The hardware's capabilities require special programming skill to utilize fully. Parallel computing (it is a form of computation in which many calculations are carried out simultaneously, operating on the principle that large problems can often be divided into smaller ones, which are then solved concurrently ("in parallel"). There are several different forms of parallel computing: bit-level, instruction level, data, and task parallelism. Parallelism has been employed for many years, mainly in high-performance computing, but interest in it has grown lately due to the physical constraints preventing frequency scaling. As power consumption (and consequently heat generation) by computers has become a concern in recent years, parallel computing has become the dominant paradigm in computer architecture, mainly in the form of multi-core processors. Parallel programming [1] is one of the key programming skills especially in real time computer vision system. Assume we have a high resolution, high speed camera, just like one of our cameras Genie HM1400[2] whose maximum capture speed in full resolution 1400 × 1024 is 60 fps and the total data rate transmission is up to 92MB/s(From camera to PC). Apparently, by the default in C/C++, the program runs in serial which cannot exert the full power of multi-cores of CPU or deal with such huge data input and processing. The traditional C/C++ program runs step by step, which means it can only deal with one image in a single core of the CPU, the other cores are not operating at all. For a quad core CPU, parallel computing can run four tasks at the same time, which means it can process four images.

OpenMP[3](Open Multi-Processing) is an application programming interface (API) that supports multi-platform shared memory multiprocessing programming in C, C++ and Fortran on many architectures, including Unix and Microsoft Windows platforms. It consists of a set of compiler directives, library routines, and environment variables that influence run-time behaviour. It is supported in Visual Studio 2008 by default. I tested it in the Sparse code de-noise method. The sparse coding transformation matrix W is extracted from 13 training images set and its sub-window size is 8×8, patch number is 6000. It iterates 100 times in the entire process. It total spent is 189.98 seconds (Dual-Core E5200) which is 50% faster than the program without OpenMP. Of course, there should be a lot optimization skill in my sparse code de-noise program: the matrix operation and the mathematic function. It would be huge topic to combine mathematic library such as Lapack(Linear Algebra PACKage) with parallel computing.

---

[1] `http://en.wikipedia.org/wiki/Parallel_computing`

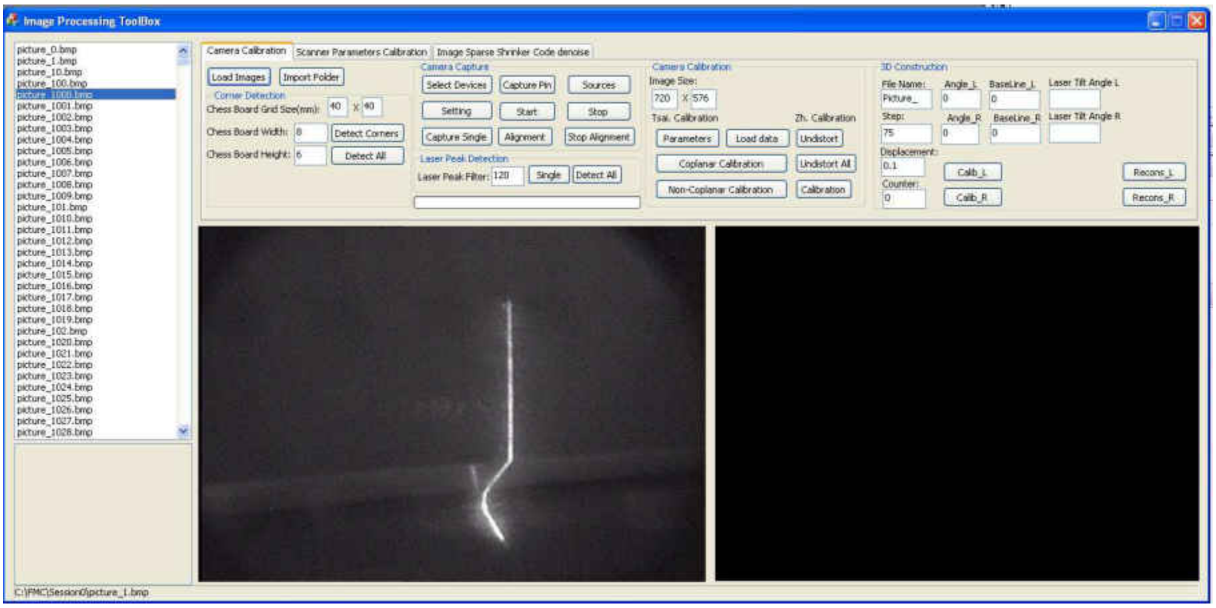
[2] `http://www.dalsa.com/mv/products/cameradetail.aspx?partNumber=CR-GM0x-H140x`

[3] `http://openmp.org/wp/`

9.2.2 Thread

To a certain extent, parallel computing is multi-threads but distribute in different CPUs. In Microsoft Windows System, there are two main types of thread, one is called GUI thread, which could be the main thread of any GUI application which is used for receiving/responding to the user input, and the other one is called Worker thread which is designed for a long calculation processing in the background. In our application, by the worker thread the program still can respond to user's order even there are huge image data being processed at the same time. Fig.9.1 is my program for image processing.

Compiled with worker thread and OpenMP, the image processing speed is up, for instance, the laser peak detection would be the most time consuming function, now its detection speed is 45 frames per second at low CPU usage around 40%. However, the hard drive capacity limits the detection speed since a hard drive in normal spindle speed at 7200 RPM determines the seek time or latency. In addition, this detection performs too many file IO: read the image, detect the laser peak and then write to hard drive. In future, I will test my program performance in a solid state hard drive which gives us much high capability in file IO.



Chess Board Grid Size(mm): 40 X 40
Chess Board Width: 8
Chess Board Height: 6
Laser Peak Filter: 120
Image Size: 720 X 576
File Name: Picture_
Angle_L: 0
BaseLine_L: 0
Step: 75
Angle_R: 0
BaseLine_R: 0
Displacement: 0.1
Counter: 0

C:\FMC\Session0\picture_1.bmp

Figure 9.1: The image processing toolbox

Bibliography

[1] A. Albarelli, E. Rodolà, and A. Torsello. Robust camera calibration using inaccurate targets. In *Proceedings of the British Machine Vision Conference*, pages 16.1–16.10. BMVA Press, 2010. ISBN 1-901725-40-5. doi:10.5244/C.24.16.

[2] M. Andreetto, N. Brusco, and G. M. Cortelazzo. Automatic 3d modeling of textured cultural heritage objects. *IEEE TRANSACTIONS ON IMAGE PROCESSING*, 13: 354–369, 2004.

[3] J.-A. Beraldin, F. Blais, M. Rioux, J. Domey, L. Gonzo, F. D. Nisi, F. Comper, D. Stoppa, M. Gottardi, and A. Simoni. Optimized position sensors for flying-spot active triangulation systems. *3D Digital Imaging and Modeling, International Conference on*, 0:29, 2003. doi: http://doi.ieeecomputersociety.org/10.1109/IM.2003.1240229.

[4] F. Blais and M. Rioux. Real-time numerical peak detector. *Signal Processing*, 11 (2):145–155, Sept. 1986.

[5] S. Bougnoux. From projective to euclidean space under any practical situation, a criticism of self-calibration. *Computer Vision, IEEE International Conference on*, 0:790, 1998. doi: http://doi.ieeecomputersociety.org/10.1109/ICCV.1998.710808.

[6] J.-Y. Bouguet and P. Perona. 3d photography on your desk. In *Computer Vision, 1998. Sixth International Conference on*, pages 43 –50, jan 1998. doi: 10.1109/ICCV.1998.710699.

[7] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[8] R. Catalan, E. Perez, and B. Perez. Evaluation of 3d scanners to develop virtual reality applications. In *Electronics, Robotics and Automotive Mechanics Conference, 2007. CERMA 2007*, pages 551 –556, sept. 2007. doi: 10.1109/CERMA.2007.4367744.

[9] R. B. Catalan, E. I. Perez, and B. Z. Perez. Evaluation of 3d scanners to develop virtual reality applications. *Electronics, Robotics and Automotive Mechanics Conference*, 0:551–556, 2007. doi: http://doi.ieeecomputersociety.org/10.1109/CERMA.2007.58.

[10] J. Chen, L. hui Zou, J. Zhang, and L. hua Dou. The comparison and application of corner detection algorithms. *Journal of Multimedia*, 4(6), 2009. URL `http://ojs.academypublisher.com/index.php/jmm/article/view/0406435441`.

[11] X. Chen He and N. H. C. Yung. Corner detector based on global and local curvature properties. *International Society for Optical Engineering*, vol. 47:p. 057008, 2009.

[12] D. Claus and A. W. Fitzgibbon. A rational function lens distortion model for general cameras. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, CVPR '05, pages 213–219, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7695-2372-2. doi: 10.1109/CVPR.2005.43. URL `http://dx.doi.org/10.1109/CVPR.2005.43`.

[13] K. D. S. B. Coleman, S.A. Concurrent edge and corner detection. *IEEE ICIP, San Antonio, USA*, pages pp. 273–276, 2007.

[14] S. Coleman, B. Scotney, and D. Kerr. Integrated edge and corner detection. *Image Analysis and Processing, International Conference on*, 0:653–658, 2007. doi: http://doi.ieeecomputersociety.org/10.1109/ICIAP.2007.80.

[15] H. Cui, N. Dai, T. Yuan, X. Cheng, and W. Liao. Calibration algorithm for structured light 3d vision measuring system. In *CISP '08: Proceedings of the 2008 Congress on Image and Signal Processing, Vol. 2*, pages 324–328, Washington, DC, USA, 2008. IEEE Computer Society. ISBN 978-0-7695-3119-9. doi: http://dx.doi.org/10.1109/CISP.2008.720.

[16] A. Datta, J.-S. Kim, and T. Kanade. Accurate camera calibration using iterative refinement of control points. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 1201 –1208, 27 2009-oct. 4 2009. doi: 10.1109/ICCVW.2009.5457474.

[17] Devernay, O. Faugeras, and I. S. Antipolis. Automatic calibration and removal of distortion from scenes of structured environments. In *In SPIE, volume 2567*, 1995.

[18] F. Devernay and O. Faugeras. Straight lines have to be straight. *Machine Vision and Applications*, 13(1):14–24, 2001. URL `http://www.springerlink.com/index/10.1007/PL00013269`.

[19] N. E. Dorsey. *Properties of Ordinary Water-Substance.* 03/1942.

[20] M. T. El-melegy. Nonmetric lens distortion calibration: closed-form solutions, robust estimation and model selection. In *In Ninth International Conference on Computer Vision (ICCV 2003*, pages 554–559, 2003.

[21] H. Farid and A. C. Popescu. Blind removal of lens distortion, 2001.

[22] R. B. Fisher and D. K. Naidu. A comparison of algorithms for subpixel peak detection. In *in Image Technology, Advances in Image Processing, Multimedia and Machine Vision*, pages 385–404. Springer-Verlag, 1996.

[23] A. W. Fitzgibbon. Simultaneous linear estimation of multiple view geometry and lens distortion. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

[24] J. Forest, J. Salvi, E. Cabruja, and C. Pous. Laser stripe peak detector for 3d scanners. a fir filter approach. *Pattern Recognition, International Conference on*, 3: 646–649, 2004. ISSN 1051-4651. doi: http://doi.ieeecomputersociety.org/10.1109/ICPR.2004.1334612.

[25] R. G. Gould. The laser, light amplification by stimulated emission of radiation. page 128. The Ann Arbor Conference on Optical Pumping, the University of Michigan, June 1959.

[26] J. Guhring. Dense 3-d surface acquisition by structured light using off-the-shelf components. In *Proc. Videometrics and Optical Methods for 3D Shape Measurement*, pages 220–231, 2001.

[27] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151, 1988.

[28] R. Hartley and S. B. Kang. Parameter-free radial distortion correction with center of distortion estimation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(8):1309 –1321, aug. 2007. ISSN 0162-8828. doi: 10.1109/TPAMI.2007.1147.

[29] J. Heikkila. Geometric camera calibration using circular control points. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(10):1066 – 1077, oct 2000. ISSN 0162-8828. doi: 10.1109/34.879788.

[30] J. Heikkila and O. Silven. A four-step camera calibration procedure with implicit image correction. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 1106 –1112, jun 1997. doi: 10.1109/CVPR.1997.609468.

[31] B. K. P. Horn, H. Hilden, and S. Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *JOURNAL OF THE OPTICAL SOCIETY AMERICA*, 5(7):1127–1135, 1988.

[32] P. Hoyer. Independent component analysis in image denoising, 1999.

[33] A. Hyvärinen. Sparse code shrinkage: denoising of nongaussian data by maximum likelihood estimation. *Neural Comput.*, 11:1739–1768, October 1999. ISSN 0899-7667. doi: 10.1162/089976699300016214. URL `http://portal.acm.org/citation.cfm?id=334369.334379`.

[34] A. Hyvärinen and U. Köster. Fastisa: A fast fixed-point algorithm for independent subspace analysis. In *ESANN*, pages 371–376, 2006.

[35] A. Hyvärinen and E. Oja. Independent component analysis: algorithms and applications. *Neural Netw.*, 13:411–430, May 2000. ISSN 0893-6080. doi: 10.1016/S0893-6080(00)00026-5. URL `http://portal.acm.org/citation.cfm?id=351654.351659`.

[36] A. Hyvarinen, P. Hoyer, and E. Oja. In *Intelligent Signal Processing.* IEEE Press, 1999.

[37] A. Hyvärinen, P. Hoyer, and E. Oja. Image denoising by sparse code shrinkage. In *Intelligent Signal Processing.* IEEE Press, 1999.

[38] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis.* Adaptive and Learning Systems for Signal Processing, Communications, and Control. J. Wiley, 2001. ISBN 9780471405405. URL `http://books.google.co.uk/books?id=96D0ypDwAkkC`.

[39] A. Hyvärinen, J. Hurri, P. O. Hoyer, A. Hyvärinen, J. Hurri, and P. O. Hoyer. Independent component analysis. In *Natural Image Statistics*, volume 39 of *Computational Imaging and Vision*, pages 151–175. Springer London, 2009. ISBN 978-1-84882-491-1.

[40] M. S. Islam, A. Sluzek, and L. Zhu. Detecting and matching interest points in relative scale. *MG&V*, 14:259–283, January 2005. ISSN 1230-0535. URL `http://portal.acm.org/citation.cfm?id=1140432.1140434`.

[41] D. N. Joanes and C. A. Gill. Comparing measures of sample skewness and kurtosis. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 47(1):183–189, 1998. ISSN 1467-9884. doi: 10.1111/1467-9884.00122. URL `http://dx.doi.org/10.1111/1467-9884.00122`.

[42] M. Kendall and A. Stuart. *The Advanced Theory of Statistics*, volume 2. Griffin, 1977.

[43] J. F. Kenney and E. S. Keeping. *Linear Regression and Correlation.* Mathematics of Statistics, Pt, 1962.

[44] D. Kerr, S. Coleman, and B. Scotney. Comparing cornerness measures for interest point detection. *International Machine Vision and Image Processing Conference*, 0:105–110, 2008. doi: http://doi.ieeecomputersociety.org/10.1109/IMVIP.2008.28.

[45] T. Kirishima, K. Sato, and K. Chihara. Realtime gesture recognition by selective control of visual interest points. *Syst. Comput. Japan*, 37:100–111, August 2006. doi: 10.1002/scj.v37:9. URL `http://portal.acm.org/citation.cfm?id=1144556.1144557`.

[46] D. Lanman and G. Taubin. Build your own 3d scanner: 3d photograhy for beginners. In *SIGGRAPH '09: ACM SIGGRAPH 2009 courses*, pages 1–87, New York, NY, USA, 2009. ACM.

[47] J.-M. Lavest, M. Viala, and M. Dhome. Do we really need an accurate calibration pattern to achieve a reliable camera calibration? In *ECCV (1)'98*, pages 158–174, 1998.

[48] H. Y. L. X. Li, W. Mouth detection based on interest points. *Proc Chinese Control Conf*, pages 610–613, 2007.

[49] J. Liu, A. Jakas, A. AI-Obaidi, and Y. Liu. A comparative study of different corner detection methods. In *Proceedings of the 8th IEEE international conference on Computational intelligence in robotics and automation*, CIRA'09, pages 509–514, Piscataway, NJ, USA, 2009. IEEE Press. ISBN 978-1-4244-4808-1. URL `http://portal.acm.org/citation.cfm?id=1811259.1811363`.

[50] M. Lourakis. levmar: Levenberg-marquardt nonlinear least squares algorithms in C/C++. [web page] `http://www.ics.forth.gr/~lourakis/levmar/`, Jul. 2004. [Accessed on 31 Jan. 2005.].

[51] M. A. Lourakis and A. Argyros. SBA: A Software Package for Generic Sparse Bundle Adjustment. *ACM Trans. Math. Software*, 36(1):1–30, 2009. doi: http://doi.acm.org/10.1145/1486525.1486527.

[52] L. Ma, Y. Chen, and K. Moore. Flexible camera calibration using a new analytical radial undistortion formula with application to mobile robot localization. In *Intelligent Control. 2003 IEEE International Symposium on*, pages 799 –804, oct. 2003. doi: 10.1109/ISIC.2003.1254738.

[53] D. Mackinnon, B. Carrier, J.-A. Beraldin, and L. Cournoyer. Gd&t-based characterization of short-range non-contact 3d imaging systems. *Int. J. Comput. Vision*, 102(1-3):56–72, Mar. 2013. ISSN 0920-5691. doi: 10.1007/s11263-012-0570-3. URL `http://dx.doi.org/10.1007/s11263-012-0570-3`.

[54] R. Mayer. *Scientific Canadian: invention and innovation from Canada's National Research Council*. Raincoast Books, 1999. ISBN 9781551922669. URL `http://books.google.com/books?id=5yvbAAAAMAAJ`.

[55] F. Mokhtarian and A. K. Mackworth. A theory of multiscale, curvature-based shape representation for planar curves. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14:789–805, August 1992. ISSN 0162-8828. doi: 10.1109/34.149591. URL `http://portal.acm.org/citation.cfm?id=138774.138775`.

[56] F. Mokhtarian and R. Suomela. Robust image corner detection through curvature scale space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:1376–1381, 1998. ISSN 0162-8828. doi: http://doi.ieeecomputersociety.org/10.1109/34.735812.

[57] H. P. Morevec. Towards automatic visual obstacle avoidance. In *Proceedings of the 5th international joint conference on Artificial intelligence - Volume 2*, pages 584–584, San Francisco, CA, USA, 1977. Morgan Kaufmann Publishers Inc. URL `http://portal.acm.org/citation.cfm?id=1622943.1622947`.

[58] M. Nixon and A. S. Aguado. *Feature Extraction & Image Processing, Second Edition*. Academic Press, 2nd edition, 2008. ISBN 0123725380, 9780123725387.

[59] J. Noble. Finding corners. volume Vol. 6, Elsevier, May 1988.

[60] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996. URL `http://www.ncbi.nlm.nih.gov/pubmed/8637596`.

[61] B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: a strategy employed by v1? *Vision Research*, 37(23):3311–3325, 1997. URL `http://linkinghub.elsevier.com/retrieve/pii/S0042698997001697`.

[62] J. A. Palmer, K. Kreutz-Delgado, and S. Makeig. Strong sub-and super-gaussianity. In *Proceedings of the 9th international conference on Latent variable analysis and signal separation*, LVA/ICA'10, pages 303–310, Berlin, Heidelberg, 2010. Springer-Verlag. ISBN 3-642-15994-X, 978-3-642-15994-7. URL `http://portal.acm.org/citation.cfm?id=1929142.1929185`.

[63] A. Peiravi and B. Taabbodi. A reliable 3d laser triangulation-based scanner with a new simple but accurate procedure for finding scanner parameters. *System*, 6(5): 80–85, 2010.

[64] F. Petit, A.-S. Capelle-Laize, and P. Carre. Underwater image enhancement by attenuation inversionwith quaternions. *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, 0:1177–1180, 2009. doi: http://doi.ieeecomputersociety.org/10.1109/ICASSP.2009.4959799.

[65] M. Pollefeys, R. Koch, and L. Van Gool. Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. In *Computer Vision, 1998. Sixth International Conference on*, pages 90 –95, jan 1998. doi: 10.1109/ICCV.1998.710705.

[66] A. Rattarangsi and R. T. Chin. Scale-based detection of corners of planar curves. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14:430–449, April 1992. ISSN 0162-8828. doi: http://dx.doi.org/10.1109/34.126805. URL `http://dx.doi.org/10.1109/34.126805`.

[67] F. Sadlo, T. Weyrich, R. Peikert, and M. Gross. A Practical Structured Light Acquisition System for Point-Based Geometry and Texture. In M. Pauly and M. Zwicker, editors, *Proceedings of the Eurographics Symposium on Point-Based Graphics '05*, pages 89–98. Eurographics Association, June 2005. ISBN 3-905673-20-7.

[68] J. Salvi, J. PagÃÍs, and J. Batlle. Pattern codification strategies in structured light systems. *Pattern Recognition*, 37(4):827 – 849, 2004. ISSN 0031-3203. doi: 10.1016/j.patcog.2003.10.002. URL `http://www.sciencedirect.com/science/article/pii/S0031320303003303`. <ce:title>Agent Based Computer Vision</ce:title>.

[69] D. Samper, J. Santolaria, A. C. Majarena, and J. J. Aguilar. Comprehensive simulation software for teaching camera calibration by a constructivist methodology. *Measurement*, 43(5):618 – 630, 2010. ISSN 0263-2241. doi: 10.1016/j.measurement.2010.01.009. URL `http://www.sciencedirect.com/science/article/pii/S0263224110000151`. <ce:title>IMEKO XIX World Congress Part 1 âĂŞ Advances in Fundamental and Applied Metrology</ce:title>.

[70] C. Schmid, R. Mohrand, and C. Bauckhage. Comparing and evaluating interest points. In *Proceedings of the Sixth International Conference on Computer Vision*, ICCV '98, pages 230–, Washington, DC, USA, 1998. IEEE Computer Society. ISBN 81-7319-221-9. URL `http://portal.acm.org/citation.cfm?id=938978.939117`.

[71] J. Shi and C. Tomasi. Good features to track. Technical report, Ithaca, NY, USA, 1993.

[72] F. C. Skotheim. Structured light projection for accurate 3d shape determination. *ICEM12- 12th International Conference on Experimental Mechanics*, 29 August - 2 September, 2004.

[73] S. M. Smith and J. M. Brady. A new approach to low level image processing. *Int. J. Comput. Vision*, 23:45–78, May 1997. ISSN 0920-5691. doi: 10.1023/A:1007963824710. URL `http://portal.acm.org/citation.cfm?id=258049.258056`.

[74] J.-P. Tardif, P. Sturm, and S. Roy. Self-calibration of a general radially symmetric distortion model. In *Proceedings of the 9th European Conference on Computer Vision, Graz, Austria*, volume 4 of *Lecture Notes in Computer Science*, pages 186–199. Springer, may 2006. URL `http://perception.inrialpes.fr/Publications/2006/TSR06`.

[75] P. Tissainayagam and D. Suter. Assessing the performance of corner detectors for point feature tracking applications. *Image and Vision Computing*, 22:663–679, 2004.

[76] T. Treibitz and Y. Y. Schechner. Active polarization descattering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31:385–399, 2009. ISSN 0162-8828. doi: http://doi.ieeecomputersociety.org/10.1109/TPAMI.2008.85.

[77] T. Treibitz, Y. Y. Schechner, and H. Singh. Flat refractive geometry. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:1–8, 2008. doi: http://doi.ieeecomputersociety.org/10.1109/CVPR.2008.4587844.

[78] B. Triggs, P. Mclauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment âĂŞ a modern synthesis. In *Vision Algorithms: Theory and Practice, LNCS*, pages 298–375. Springer Verlag, 2000.

[79] R. Y. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. pages 221–244, 1992.

[80] J. Weng, P. Cohen, and M. Herniou. Camera calibration with distortion models and accuracy evaluation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14(10):965 –980, oct 1992. ISSN 0162-8828. doi: 10.1109/34.159901.

[81] J. Weng, P. Cohen, and M. Herniou. Camera calibration with distortion models and accuracy evaluation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14(10):965 –980, oct 1992. ISSN 0162-8828. doi: 10.1109/34.159901.

[82] H. B. Wu, Y. Chen, M. Y. Wu, C. R. Guan, and X. Y. Yu. 3d measurement technology by structured light using stripe-edge-based gray code. *Journal of Physics: Conference Series*, 48(1):537, 2006. URL `http://stacks.iop.org/1742-6596/48/i=1/a=101`.

[83] Z. Zhang. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(11):1330–1334, 2000. ISSN 0162-8828. doi: http://dx.doi.org/10.1109/34.888718.

[84] M. Zuliani, C. Kenney, and B. S. Manjunath. A mathematical comparison of point detectors. *Computer Vision and Pattern Recognition Workshop*, 11:172, 2004. ISSN 1063-6919. doi: http://doi.ieeecomputersociety.org/10.1109/CVPR.2004.282.