

Copyright  
by  
Zuyun Zhao  
2012

**The Thesis Committee for Zuyun Zhao**  
**Certifies that this is the approved version of the following thesis:**

**Line scan camera calibration for fabric imaging**

**APPROVED BY**  
**SUPERVISING COMMITTEE:**

**Supervisor:**

---

Bugao Xu

---

Jonathan Y. Chen

**Line scan camera calibration for fabric imaging**

**by**

**Zuyun Zhao, B.E.; M.E.**

**Thesis**

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

**Master of Science in Textile and Apparel Technology**

**The University of Texas at Austin**

**May 2012**

## **Abstract**

### **Line scan camera calibration for fabric imaging**

Zuyun Zhao, M.S.T.A.T.

The University of Texas at Austin, 2012

Supervisor: Bugao Xu

Fabric defects inspection is a vital step for fabric quality assessment. Many vision-based automatic fabric defect detection methods have been proposed to detect fabric flaws efficiently and accurately. Because the inspection methods are vision-based, image quality is of great importance to the accuracy of detection result. To our knowledge, most of camera lenses have radial distortion. So our goal in this project is to remove the radial distortion and achieve undistorted images.

Much research work has been done for 2-D image correction, but the study for 1-D line scan camera image correction is rarely done, although line scan cameras are gaining more and wider applications due to the high resolution and efficiency on 1-D data processing. A novel line scan camera correction method is proposed in this project. We first propose a pattern object with mutually parallel lines and oblique lines to each pair of parallel ones. The purpose of the pattern design is based upon the fact that line scan

camera acquires image one line at a time and it's difficult for one scan line to match the "0-D" marked points on pattern. We detect the intersection points between pattern lines and one scan line and calculate their position according to the pattern geometry.

As calibrations for 2-D cameras have been greatly achieved, we propose a method to calibrate 1-D camera. A least-square method is applied to solve the pinhole projection equation and estimate the values of camera parameter matrix. Finally we refine the data with maximum-likelihood estimation and get the camera lens distortion coefficients. We re-project the data from the image coordinate to the world coordinate, using the obtained camera matrix and the re-projection error is 0.68 pixel. With the distortion coefficients ready, we correct captured images with an undistortion equation.

We introduce a term of unit distance in the discussion part to better assess the proposed method. When testifying the undistortion results, we observe corrected image has almost identical unit distance with standard deviation of 0.29 pixels. Compared to the ideal distortion-free unit distance, the corrected image has only 0.09 pixel off the average, which proves the validity of the proposed method.

## Table of Contents

List of Tables .....	viii
List of Figures .....	ix
Chapter 1: Introduction .....	1
1.1 Motivation and Goals.....	1
1.2 Methods for fabric defects inspection.....	2
1.3 Image undistortion .....	9
1.4 Camera calibration .....	10
1.4.1 Area-based camera calibration.....	11
1.4.2 Line-scan camera calibration .....	12
1.5 Structures of the thesis .....	13
Chapter 2: Framework of Image Capturing System by Line-scan Camera .....	15
2.1 Principle of linear scan system .....	15
2.2 System setup .....	16
Chapter 3: Calibration of Line Scan Camera .....	19
3.1 Introduction.....	19
3.2 Calibration.....	19
3.2.1 Pattern design.....	19
3.2.2 Geometric formation.....	21
3.2.3 Image feature points extraction.....	24
3.2.4 1-D to 2-D mapping .....	27
3.2.5 Camera matrix calculation .....	33
3.3 Experimental results.....	35
3.4 Conclusion and discussion.....	37
Chapter 4: Image Undistortion and Correction .....	39
4.1 Radial distortion equation .....	39
4.1.1 2-D camera radial distortion .....	39
4.1.2 1-D camera radial distortion .....	41

4.2	Distorted image correction.....	42
4.3	Image correction result .....	43
Chapter 5: Conclusion.....		53
Bibliography .....		55

## **List of Tables**

Table 1.1 Major types of fabric defects .....	3
Table 3.1 Camera intrinsic and extrinsic parameters .....	37
Table 4.1 Image UD before and after correction .....	46
Table 4.2 UD before and after correction (with a skewed camera plane) .....	48
Table 4.3 Model summary and parameter estimations .....	49
Table 4.4 Test result summary .....	51
Table 4.5 Model summary and parameter estimates .....	51



## List of Figures

Figure 2.1 View of 2-D camera and 1-D camera.....	16
Figure 2.2 Line scan camera .....	17
Figure 2.3 System configuration.....	18
Figure 3.1 Planar pattern for camera calibration .....	20
Figure 3.2 Pattern image segment.....	21
Figure 3.3 Image segment for feature points extraction .....	24
Figure 3.4 Image from one line of pixels.....	24
Figure 3.5 Line data grey scale distribution.....	26
Figure 3.6 Camera inner look .....	28
Figure 3.7 2-D image data establishment .....	30
Figure 3.8 2-D object data establishment .....	33
Figure 3.9 Procedure diagram.....	36
Figure 4.1 Originally captured image .....	44
Figure 4.2 Corrected image.....	44
Figure 4.3 UD before and after correction.....	47
Figure 4.4 UD before and after image correction (with a skewed camera plane) .	49
Figure 4.5 Linear simulation.....	50
Figure 4.6 Linear simulation.....	52

# **Chapter 1: Introduction**

## **1.1 MOTIVATION AND GOALS**

Quality is one of the most important elements to evaluate textile fabrics. It is desirable to produce high quality products to meet customers' requirements and gain competitive ability. Quality inspection of textile fabric is an important problem for fabric manufactures. In the manufacturing facility, typical product is 1 to 3 meters wide and is driven with speeds ranging from 20 to 200 m/min. Fabric defects are traditionally inspected manually, which is very time-consuming and expensive. Moreover, only about 70% of the defected can be detected, even by the most highly trained inspectors during their most efficient time period, and the detected defect rate will decrease when the inspectors work with fatigue. The rate drops as the fabric is moving faster than 30 m/min.

To enhance the accuracy of fabric defects detection, and save human from this tedious and stressful work, automated computer vision based fabric inspection has been proposed [1] [2], and has already received substantial progress. Almost all computer vision based inspection solutions rely on digital cameras to acquire images of the inspected fabrics. To accommodate the moving nature of the fabric at the production line, line scan cameras are often used. This is because line scan camera only captures one line at a time. Images of the fabrics are created as the scanned lines accumulate while the inspected fabric moves across the camera's view. But in order to get accurate defects detection results, we should first make sure the source images are in good condition, i.e., images truthfully reflect the geometrical features or color tones of fabric. That is why most of researchers would like to apply camera calibration before image capturing, to obtain relatively accurate images. Currently, there are multiple papers dedicate to area-based camera calibration, but few researches shed lights on line-scan camera calibration.

In this study, we will focus on the line-scan camera calibration method, and try to found its application in fabric imaging.

## **1.2 METHODS FOR FABRIC DEFECTS INSPECTION**

Generally, the methods for fabric defects inspection can be divided into two categories. One is subjective; the other is objective. Defects exist in many types, and the chances of them to be distinguished by an inspector vary depending on the fabric weaving pattern. In the case of human vision inspection the types of defects that are mostly likely to be detected are ‘Burl Mark’, ‘Drawbacks’, ‘Dropped Pick’, ‘End Out’, ‘Mixed Filling’, ‘Open Reed’, ‘Smash’, ‘Soiled Filling’, ‘Thin Place’, ‘Broken Color Pattern’, ‘Drop Stitches’, ‘Hole’, ‘Missing Yarn’, ‘Mixed Yarn’, ‘Press-off’, and ‘Runner’. Table 1.1 shows a list of images illustrating the various fabric defects. But as we have found, visual evaluation manually can be subjective and inefficient, and thus the result from human vision is not reliable. So a lot of researchers started to put their efforts on objective evaluation system design. Therefore automatic and computer vision based detection methods have emerged and developed very quickly. Basically, the development of automatic fabric defects detection has gone through three steps, from the color image processing technique, to digital images processing, and to the recent artificial neural network (ANN) approach.

**Table 1.1** Major types of fabric defects


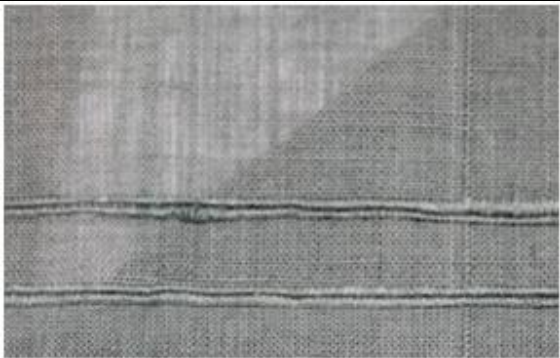
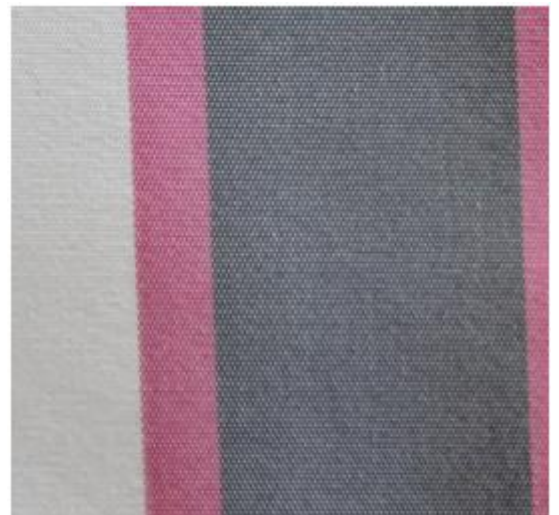





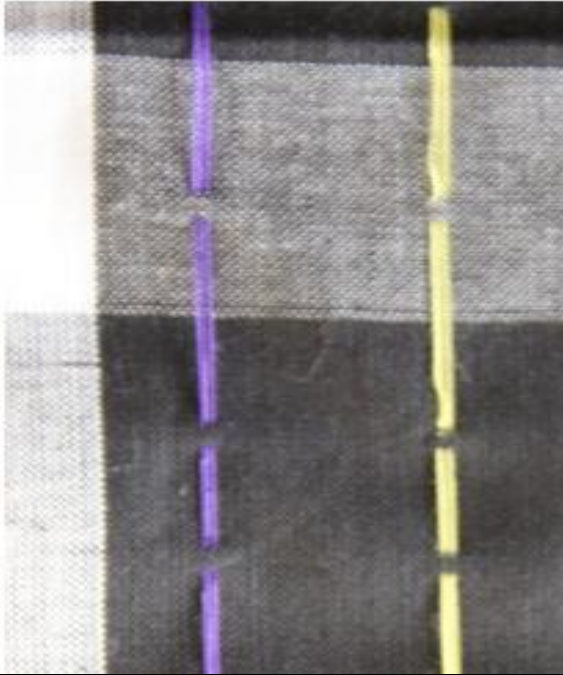

Flawless fabric		Fabric defect	
	Burl mark(1)		
	Burl mark(2)		
	hole		



Table 1.1, cont.

	Thin place	
	drawbacks	

**Table 1.1, cont.**

	End out	
		

For the color image processing level, Takato et al developed a gray-level matching system to inspect fabric defects [1]. But in this paper, to implement the inspection, all image conditions should remain constant, which is hard to achieve. Also the threshold setting was subjective. These problems were solved by Zhang et al.[2] in

1995, when he and his coworkers introduced gray-level statistical and morphological method to do the fabric defect evaluation and classification. Convolution mask method was introduced by Lanes et al. in [3]. Depending on the intensity contrast in the boundary of regular fabric and defects, like spots, knot, and slab, this method used multiple masks to identify the defects. But this approach cannot cover most of the commonly found defects like drawbacks, or missing yarn, for there is not obvious boundary or contrast around these kinds of defects. Northon et al.[4] proposed adaptive threshold and binary filtering method to achieve a high contrast between foreground defects and background fabric. But there is still detectable influence of noise which hinders the separation of defects from fabric.

Although apparent progression has been made in defects inspection through color image processing method, system with this approach is not redundant and the evaluation result is so sensitive to ambient light that one fabric set under different lumen can get totally different defects. This limitation makes the space for the development of digital image processing when researcher began to use frequency-domain theory to solve temporal-domain problems. Methods in this field include cosine transform[5], Fourier Transform[6][7][8][9] , Fast Fourier Transform[10], and wavelet transform[11][12]. Fourier transform has been used by Sari-Saraf et al in [9] to detect fabric defects. This approach used the idea of concentric ring filter in optical processing[13], and rings were concentric with different radii and were used to inspect filling and warp densities. This approach was achieved by performing a one-dimensional signature diagram in the two-dimensional spectrum through integrating the points within each ring in the two-dimensional spectrum. The main advantage was that this approach can effectively get rid of the influence of background noise and obtain clearer defect results because of the

dimensional changes in the fabric structure. The only limitation for this approach was the high-order (2 dimensions) of complexity, which meant longer computational time. A vision-based on-loom fabric inspection system was introduced in [12], using wavelet transform, image fusion and the correlation dimension theory to implement the defect segmentation algorithm, which was based on the idea that local defects in the input images would disrupt the global homogeneity of the background fabric texture. After repeated testing on 3700 images, the inspection system got an overall 89% efficacy with localization accuracy in 0.2 in (the minimum defect size). The shortcoming for this inspection system was the cost because even small textile plants need over 20 looms. Chan and his colleagues [14] used Fourier Transformation to get the frequency spectrum of a fabric and created simulated models to understand the relationship of fabric structure between spatial-domain and frequency-domain. This approach first classified all commonly found defects broadly into four categories: double yarn; missing yarn; webs or broken fabric and yarn density variation and use the central spatial frequency spectrums to analysis the fabric structure and finally extract seven parameters to describe the above four classes of defects. But for the fabric defects are so numerous and complex, simply four classes cannot cover all defects we need to recognize and identify in the practical industrial usage.

Artificial Neural Network (ANN) approach is a newly developed approach in conjunction with image processing to recognize and classify fabric defects and has been proved as a powerful tool when dealing with the recognition and classification [15][16][17]. Lippmann initiated the theory of neural network in [18], and described that ANNs were made up with multiple interconnected neurons that performs either in parallel or connected with weights. They were developed for assessing set marks and parameter



selection. Rajasekaran [19] employed an ANN method combined with image processing technology to identify fabric defects. A direct approach using optical acquisition and an ANN was introduced by Borzone et al. [20] to analyze the acquired fabric data. Good classification rate demonstrated the relevant information on the textile sample. Also the system was very fast since there is no complicated computation in algorithm implementation, thus was suitable for the real-time inspection of fabric defects with a high detection rate. Shady et al. [21] reported an ANN algorithm based method to detect and classify knitted fabric defects. They choose six different knitting defects for study and employed both statistical procedure and Fourier Transform as approaches for feature extraction. After 30 samples testing, they got the conclusion that their method is successful to classify most of the defects, also Fourier Transform feature extraction method show more success compared to the statistical approach when detecting the defect-free fabrics. The Back-propagation Neural Network (BPNN) is a hierarchical feed-forward ANN composed with three or more fully interconnected layers of neurons. Since Tsai and Hu [22] first used the BPNN to classify a fabric image's Fourier spectrum with nine parameters extracted, the BPNN has been the most widely used ANN architecture [23]. Chen et al. [24] used a BP neural network with power spectra to classify fabrics. Shiau et al. [25] reported a method to classify web defects by color image processing using BP neural network. Recently, an approach combining BP Neural Network and Wavelet digital image processing method to detect textile flaws was introduced in [26] by Yin and his members. They narrowed textile defects to two types: one is oil stain; the other is hole. Wavelet is used to separate the background of the pattern of fabric and leave the fabric defects to sub-images. BP neural network training and recognition procedure is implemented to identify types of flaws. They reported an effective detection

results and accurate recognition rate for fabric defects after 32 samples testing. The hybrid approach of putting wavelet transform and BP neural network together to fix the problem to defects detection was also employed by Wong and his colleagues [27], but they focus stitching defect, which is rarely researched, compared to the other more commonly found textile defects. After sample testing, their classification results demonstrated that the proposed method can effectively identify five classes of stitching defects with high recognition and detection accuracy.

### **1.3 IMAGE UNDISTORTION**

After we go through the current search papers on fabric defects inspection, we found that almost all the automatic and objective method is computer vision based. In that case, the quality of source fabric images captured from camera is of great importance to determine the subsequent defects detection accuracy. A poor quality image could be ghost, being blurred or unclear enough. As to the poor images like this, the reason is that the camera wasn't adjusted to a suitable focal length that makes the view of objects not clear, and this problem can be solved when we get through multiple trainings to adjust the focal length to the best we can see. But there is another type of poor images not caused by manually inexperienced, but by the imperfection of camera lens which inevitably happens all the time, called lens distortion. It occurs when a lens produces curved lines where straight lines should take place [28]. Several types of lens distortion exist; however, radial distortion is usually the most severe part of the total lens distortion. Radial distortion actually implies a nonlinear radius mapping from the object to the image: Pincushion distortion is actually simply an exaggerated radius mapping for large radii in comparison with small radii. Conversely, barrel distortion is actually a diminished radius

mapping for large radii in comparison with small radii. So the value of radius directly determines the distortion severity of the image projection from world objects.

Source images obtained from lens distortion will greatly hinder the accuracy of defects detection, especially when we need to measure the size of defects, and compute the overall defect rates of fabric samples. To fix this problem and get good quality source image, image undistortion is only way to help us out.

There have been a lot of effects focused in this area. From the initially photogrammetric method for distortion modeling and removal [29] to the camera calibration based method [30] [31]. In [31], Weng and his colleagues obtained the equation to describe radial distortion, and this equation is simplified by Tsai in his paper [30] for more practical purposes. But both of the above methods need distortion coefficients to correct distorted image, which should be computed from camera calibration.

#### **1.4 CAMERA CALIBRATION**

Camera calibration is a necessary step to get the relationship between the 3D object coordinates and the image coordinates. This transformation is determined in camera calibration procedure by obtaining the unknown parameters of the camera model, including internal calibration parameters and external parameters. The internal parameters include the basic intrinsic camera matrix that affects how sensor samples the scene; and the external parameters indicate the position and orientation of the sensor related to the world coordinate system. According to the dimensional difference of image sensor in camera, line-scan and area-scan, we have different approaches to deal with these two types of cameras.

#### **1.4.1 Area-based camera calibration**

There have been tremendous works on the area-based camera calibration. Many different camera calibration approaches have been reported, such as traditional photogrammetric method [32], computer vision [33][34][35][36][30][37], and self-calibration [38][39].

Photogrammetric calibration is first developed for aerial imaging and surveying. In [40], Brown focused on the close-range photogrammetry to take into account of the variation of lens distortion. The analytical plumb line method is used in this approach to confirm the validity of the theory accounting for variation of distortion with object distance. This approach was reported to excel in operational convenience and be able to achieve relatively accurate results. But recently photographic cameras have been replaced by video cameras.

As to computer vision approach, according to the difference of dimensions of calibration objects, the technique can be divided into two categories: 3D object calibration and 2D planar object calibration.

For 3D object calibration, it's conducted by observing a calibration object with précised 3-D shape and position, usually including two or three planes orthogonal to each other. A technique for three-dimensional camera calibration was reported by Tsai[30] for machine vision metrology using off-the-shelf TV cameras and lenses. He used two-stage technique to achieve efficient computation of camera external parameters and focal length, radial lens distortion, and image scanning parameters. His system was reported to be efficient, accurate, and straightforward to implement in real environment and test

results were described with both accuracy and speed reported. But this approach needs highly accurate calibration apparatus and complicated setup[41].

For 2D planar object calibration, techniques in this category don't need elaborate setup as[30] does and it's required to observe the planar objects shown in multiple different directions. Zhang made apparent contribution to this field in his paper [37], he proposed a flexible new technique to easily calibrate a camera. It only required the camera to observe a planar pattern shown at a few (at least two) different orientations. The planar pattern can be printed on a laser printer with regular marked points that can be measured with great accuracy. It was not necessary to know the plane motion. The proposed method included a closed-form solution, using a nonlinear refinement based on the maximum likelihood criterion. Test results from both computer simulation and real data was reported very well. Compared with classical techniques using expensive equipment, his approach is more flexible and easier to be implemented.

#### **1.4.2 Line-scan camera calibration**

Although there have been a lot of publications talking about area-based camera calibration and the research on this field is relatively mature, however, line-scan camera calibration is rarely talked about [42][43]. Compared to the 2-D matrix cameras, 1-D cameras are more efficient and accurate of the measurement, and 1-D data is easier and faster to process than 2-D images. Right now the application of line-scan cameras for measurement in various areas is increasing, and in our fabric defects inspection, we used line-scan cameras to capture fabric images.

From the scant information available on the subject of linear camera calibration, the approach proposed by Horaud et al [44] is helpful for the subsequent research in this

field. They proposed a multiline calibration method to calculate the external parameters. The calibration pattern had three mutually parallel lines with the fourth line crossing the three lines and making an angle with the direction of three. With this pattern design, they can get the geometric position of the camera with respect to pattern and thus the camera's external parameters matrix. The paper reported by Luna et al [43] proposed a method based the multiple line calibration technique [44], using a calibration pattern with two parallel planes marked by 4 parallel lines for each. With pattern geometry, they can describe the lines on pattern. Using the straight line that is captured by line-scan camera and the lines on pattern, they obtained the relationship of the 3-D world coordinate and the image coordinate, and got the internal and external parameters of the line scan camera through standard calibration procedure based on recursive least squares method. After repeated calibration process for 500 times, they reported the obtained median residual error was 0.28 pixel. But his approach ignored the optical distortion because they fixed camera at long focal length, at approximately 100mm. However in real application, focal length should be flexible and adjustable, in this case, optical distortion could be apparent and cannot be ignored.

## **1.5 STRUCTURES OF THE THESIS**

This thesis will cover four parts, Chapter 2 to Chapter 5, to report our contribution to the research of line-scan camera calibration and image undistortion technique in detail for fabric defects inspection system.

Chapter 2 will discuss the “Framework of image capturing system by line-scan camera”. This chapter will show us the system setup, the linear camera scan principle.

This part will give the readers basic idea of linear scanning system and its difference from commonly used 2-D scan.

Chapter 3 is “Calibration of line-scan camera”. In this part, a novel linear camera calibration algorithm developed by ourselves will be discussed in detail. It consists five steps to solve the calibration problem, object pattern design, geometric formulation, feature extraction on image coordinate, 1-D to 2-D data frame establishment, functional minimization and camera matrix computation. With this method, we can get all camera parameters, including external parameters, internal parameter, and optical distortion coefficients, with the re-projection error less than 0.68 pixels.

Chapter 4 will discuss “Image undistortion and correction”. A calibration-based method is developed for fabric image undistortion. We apply the optical distortion coefficients, which are obtained in the last chapter, to create lens distortion formula, and then derive the correction formula to calculate undistortion coefficients. After the undistortion coefficients are obtained, distorted images can be corrected with undistortion matrix. The last section reports the image correction results after a series of validation tests, we propose the term of UD to reliably evaluate the accuracy of our correction method.

Chapter 5 gives conclusion for this study.

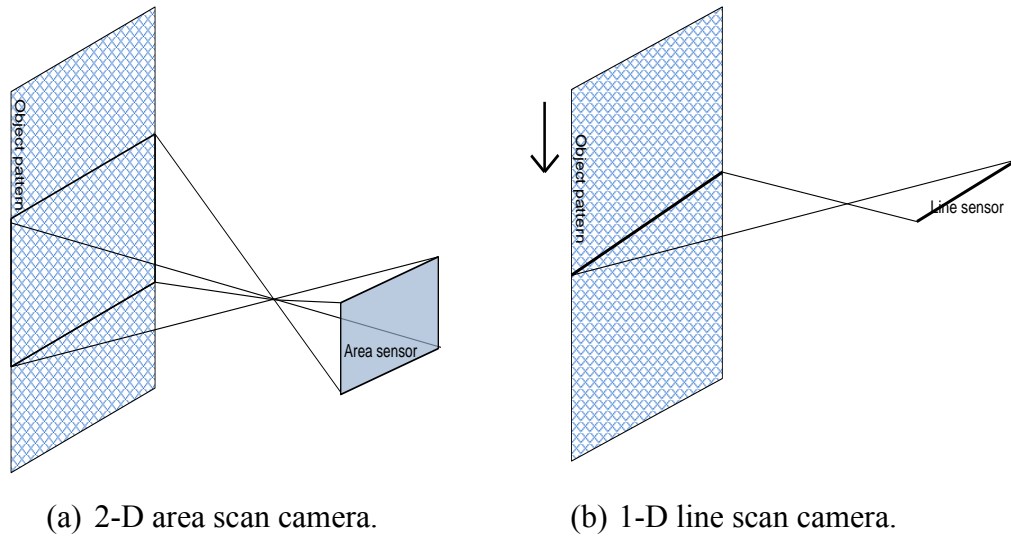
## **Chapter 2: Framework of Image Capturing System by Line-scan Camera**

Application of line-scan camera for measurement is becoming increasingly important because of the accuracy and efficiency of the linear data captured from line-scan camera. This chapter will give us a brief review of the basic principle of line-scan system, and then the setup of the fabric image capturing system will be presented in detail.

### **2.1 PRINCIPLE OF LINEAR SCAN SYSTEM**

Line-scan camera is an image capturing device whose CCD sensor is formed by a single line of photosensitive elements (pixel) (Figure 2.1(b)) rather than rectangular CCD sensor which is used by area-based 2-D cameras. (Figure 2.1(a)) Although the sensor is only one pixel high but can be very wide to three or four thousand pixels. Line-scan camera acquires images line by line, and by moving objects or camera at certain speed, we can obtain a complete 2-D image at the end. This can very useful when inspecting items such as newspaper, or banknotes being printed when passing under the camera very quickly. The image built up can subsequently be analyzed for flaws or defects on the computer.



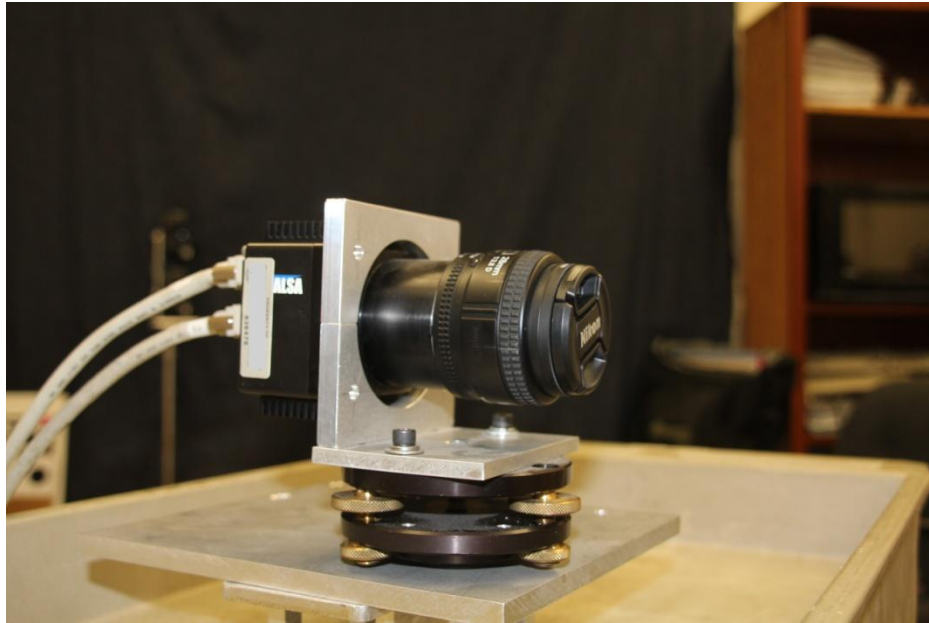


**Figure 2.1** View of 2-D camera and 1-D camera.

One single scanning line is a 1-D mapping of the gray level related to every single point of the captured line and every sudden change in a single point of objects would cause variation, either color or other aspect, of the acquire image. Therefore, line scan camera has better accuracy in measurement than an area sensor. Also linear sensors normally have high resolution than that of 2-D camera.

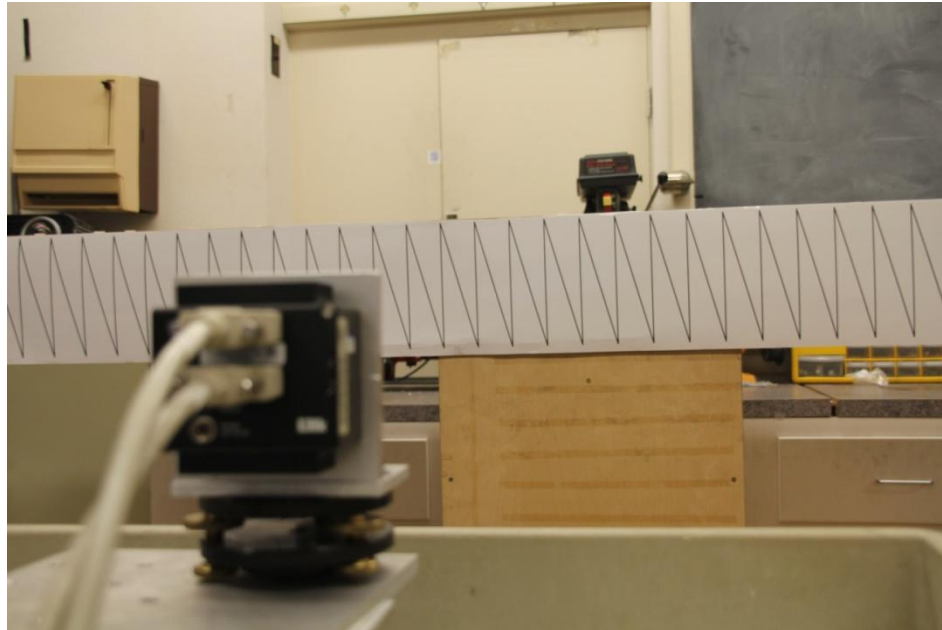
## 2.2 SYSTEM SETUP

This section describes the hardware design for the study. The following Figure 2.2 displays the line scan camera we use. The high-resolution DALSA line scan camera has 2048 pixels on the CCD. A Nikon F-mount lens with effective focal length of 28 mm is used.



**Figure 2.2** Line scan camera

Figure 2.3 illustrates the configuration of our system. The object we put in front of the camera is the pattern we proposed with around 115 pattern lines (including vertical parallel and oblique ones).



**Figure 2.3** System configuration

The camera is controlled by a computer via Camera Link interface, and captured images can be transferred directly to the computer memory, which makes the system more efficient and practical. A desktop computer with Intel Core 2 Quad 2.4 GHz CPU and 4GB RAM is used to control the cameras and receive images from the camera.

## **Chapter 3: Calibration of Line Scan Camera**

### **3.1 INTRODUCTION**

Using - has a few advantages over area based 2-D camera, because of its high resolution and efficiency when dealing with 1-D data. And Line scan camera has been increasingly used in automotive industries [45] and bioengineering [46]. Calibration is an important step before we use the camera for research purpose. But as to our knowledge, there is scant information talking about line scan camera calibration and even though some publications dealing with this problem got some result, their results were conditionally obtained, based upon ignoring a few parameters which are important and can be ignored in our project. On the contrary, calibration methods for 2-D camera have been studied intensively, from the traditional 3-D object calibration to the relatively flexible planar objects calibration. In this chapter, we will introduce a novel method of applying 2-D calibration method to fix the 1-D camera calibration problem. The following are the major steps to solve the calibration problem in our project.

### **3.2 CALIBRATION**

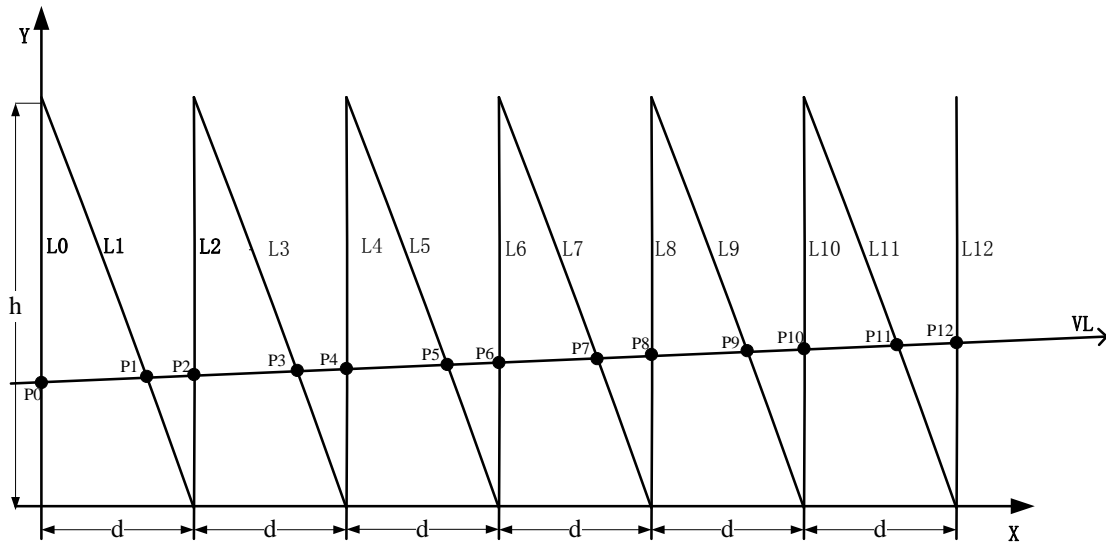
#### **3.2.1 Pattern design**

In the calibration process of the area 2-D camera, the important problem to be fixed is the determination of coordinate projection from world coordinate to image plane with the significant points of the pattern. These significant points could be corners [37], centers of circles [36], or intersection between lines [44]. In the case of line scan camera, since the scan line is invisible, it is very difficult to match the scan line with the specific

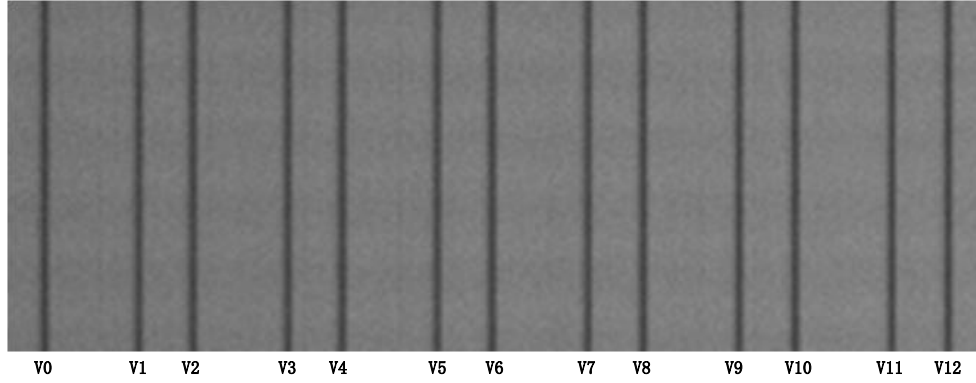
points of the pattern. In our project, we propose a calibration pattern and a method to obtain these points with geometric principle of the pattern and scan line.

The pattern we used is a planar pattern with 58 straight lines and 57 oblique lines connecting each pair of parallel straight lines. To explain how to map the image coordinate to world coordinate, here we only show 7 straight lines and 6 oblique ones in Figure 3.1, where  $L_0, L_2, L_4, L_6, L_8, L_{10}, L_{12}$  and  $Y$  axis are mutually parallel and  $L_1, L_3, L_5, L_7, L_9$ , and  $L_{11}$  make an angle to  $Y$  axis.

Figure 3.2 is the corresponding image segment to show what the scan line “viewed” from the pattern. Since the camera is stationary when we capture the image from pattern, only one line has information and other lines are just the copy of the first line data. So each column of the image represents one point of scan line on the pattern.



**Figure 3.1** Planar pattern for camera calibration



**Figure 3.2** Pattern image segment

### 3.2.2 Geometric formation

Due to the basic structure of the camera, the view line cannot be seen directly, but it can be detected using the geometric information of the pattern we propose. First we assume the view line as VL, (see Figure 3.1), and VL makes cross points with pattern lines when it goes by. We label the cross points as  $P_0, P_1, P_2, \dots, P_{11},$  and  $P_{12}$ , and the segments between the two neighboring cross points as  $P_0P_1, P_1P_2, \dots, P_{10}P_{11}, P_{11}P_{12}$ . As to the image part (see Figure 3.2), as we mentioned in the last section, since each column of image pixels represents only one point of scan line, the black lines in the captured image are the accumulation of intersection points when the scan line meets with the black pattern lines. Thus, we can consider one dark line as one cross point. These cross points can be detected by edge recognition method, which will be discussed in detail in the next section. Here we just assume the extracted image cross points named as  $V_0, V_1, V_2, \dots, V_{11}, V_{12}$ , which is counterpart to the pattern cross points in world coordinate. Correspondently, the distance between the neighboring cross points in captured image are defined as  $V_0V_1, V_1V_2, \dots, V_{10}V_{11}, V_{11}V_{12}$ . According to the pinhole model projection principle, based on the precondition that view line or the one

dimensional CCD slot is parallel to target pattern plane, and that the distance between neighboring points is small enough, we can get the following equation:

$$\frac{P_{n-1}P_n}{P_nP_{n+1}} = \frac{V_{n-1}V_n}{V_nV_{n+1}} \quad (3.1)$$

Computing error is neglect able even if the camera view line is not perfectly parallel to the pattern planes [46]. And in our project, we manually adjust the camera to make the CCD slot as parallel to target plane as possible to reduce the error. Also we create 115 pattern lines within 3.408 meters. That means our computation is based upon each pair of neighboring cross points, in a relatively small area, where the computer error is much less.

$V_{n-1}V_n$  and  $V_nV_{n+1}$  can be easily computed once we have the position of cross points,  $V_{n-1}$ ,  $V_n$ , and  $V_{n+1}$  in image coordinate.

Given the equation (3.1), the relationship between  $P_{n-1}P_n$  and  $P_nP_{n+1}$  is ready for subsequent computation.

To find the position of cross points on patterns,  $P_{n-1}$ ,  $P_n$ , and  $P_{n+1}$  we should resort to the triangle model in Figure 3.1. As the unit width and height of calibration pattern can be measured directly, named as  $d$  and  $h$  respectively, and given the ratio of  $P_{n-1}P_n$  over  $P_nP_{n+1}$ , we can easily calculate the position of  $P_1, P_3, \dots, P_{2n-1}$ , ( $n$  is positive integer) using the following equation:

On  $X$  axis,

$$\begin{aligned} x_{2n-1} &= d \times \frac{P_{n-1}P_n}{P_{n-1}P_n + P_nP_{n+1}} + (n-1) \times d \\ &= d \times \frac{1}{1 + \frac{P_nP_{n+1}}{P_{n-1}P_n}} + (n-1) \times d \end{aligned} \quad (3.2)$$

Given the relationship defined in equation (3.1), we have

$$x_{2n-1} = d \times \frac{1}{1 + \frac{V_n V_{n+1}}{V_{n-1} V_n}} + (n-1) \times d \quad (3.3)$$

On  $Y$  axis,

$$y_{2n-1} = h \times \frac{1}{\frac{1 + P_{n-1} P_n}{P_n P_{n+1}}} = h \times \frac{1}{1 + \frac{V_{n-1} V_n}{V_n V_{n+1}}} \quad (3.4)$$

After we obtain the position of odd cross points, the position of even cross points can also be calculated because both odd and even cross points are theoretically supposed to be in the same line, since the view line is straight. So we can take the slope of odd cross points as overall slope for the view line, when the slope of odd points is easy to find. But in most cases, because of imperfection of CCD slot during manufacturing, view line is not perfectly straight, so we cannot simply use overall slope as the regional slope for each pair of even points. To solve this problem, we take the regional slopes between two neighboring odd points separately for the approximation of the corresponding even points slope and calculate slope for each even cross point using the linearity of its neighboring left and right odd points. As to the start and end even points, since they don't have left and right neighbor, respectively, we apply the nearest slope to their calculation. So we get the following equation:

$$\begin{cases} x_0 = 0, \\ y_0 = P_1 \cdot y + \frac{P_3 \cdot y - P_1 \cdot y}{P_3 \cdot x - P_1 \cdot x} \times (P_0 \cdot x - P_1 \cdot x) \end{cases} \quad (3.5)$$

$$\begin{cases} x_{2n} = n \times d, \\ y_{2n} = y_{2n-1} + \frac{y_{2n+1} - y_{2n-1}}{x_{2n+1} - x_{2n-1}} \times (x_{2n} - x_{2n-1}) \end{cases} \quad (3.6)$$

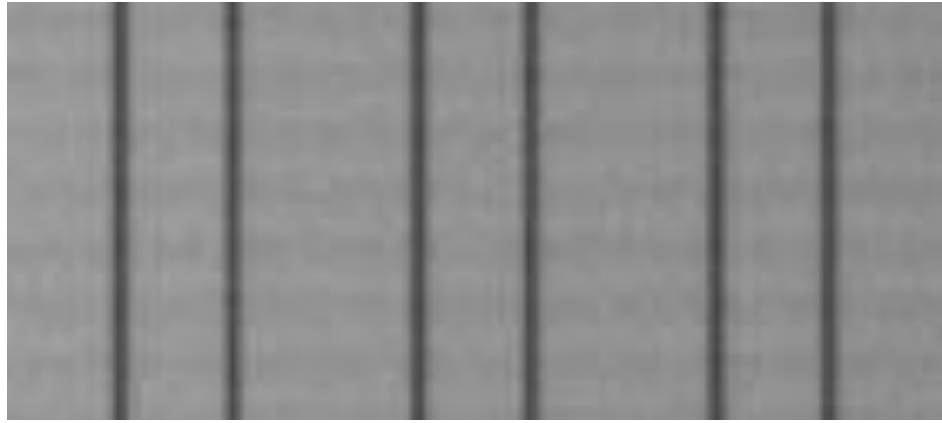


$$\begin{cases} x_{2N} = N \times d, \\ y_{2N} = y_{2N-1} + \frac{y_{2N-1} - y_{2N-3}}{x_{2N+1} - x_{2N-3}} \times (x_{2N} - x_{2N-1}) \end{cases} \quad (3.7)$$

In the above equations,  $n$  refers to the index of unit width of two neighboring parallel lines, and when  $N$  indicates the index of unit width for the last pair of parallel line.

### 3.2.3 Image feature points extraction

As we've already known that the 2-D image we got is only the repeating of one line data captured from object pattern (see Figure 3.3) since we didn't move camera, neither pattern. So the grey value of the pixels in the same column should identical and we randomly choose 10 lines of pixels and calculate the average of the 10 and finally carry out only one line pixel points for subsequent analysis.



**Figure 3.3** Image segment for feature points extraction

The following picture with 1 pixel high is the line data we get from the 2-D image



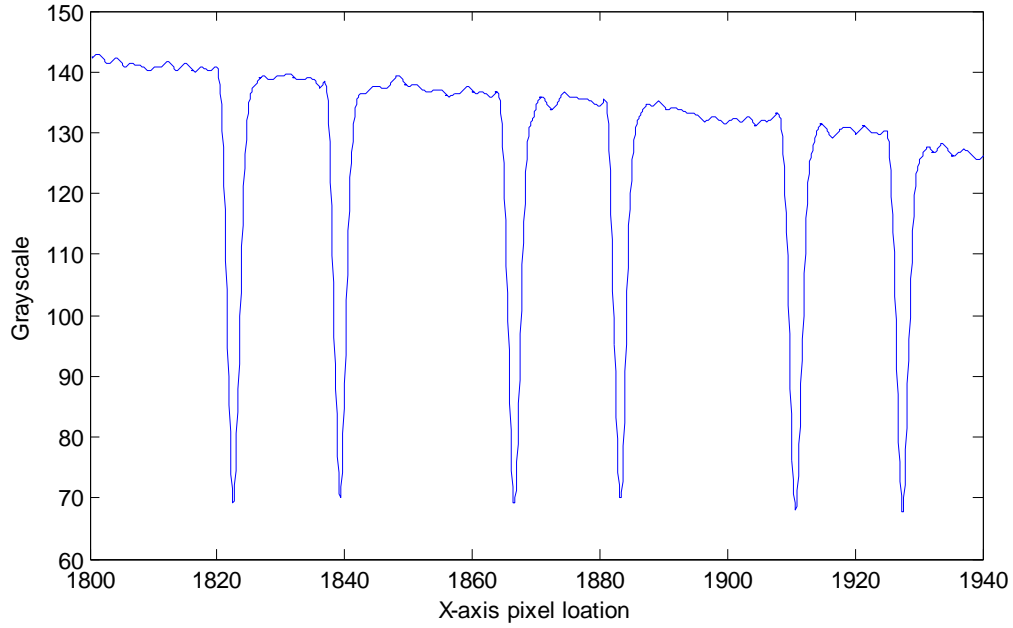
**Figure 3.4** Image from one line of pixels.

To extract the position of the dark cross points in this line, we should first figure out the region of interest (RI) because the one object feature point covers more than one pixel in the image. Using global threshold method, we define the region with all pixels' (pixel numbers larger than 2) grey value less than average grey value of the whole line as the RI, shown as the following expression,

$$\forall a \in A, \quad \text{if } g(a) < \text{Aver}, \text{ then } A \subseteq RI \quad (3.8)$$

Where “A” is the set of continuous pixels, “a” is the pixel point of region “A”, “g(a)” refer to the grey value of “a”, “Aver” refers to the overall average, and “RI” is the sets consisting RI.

After we fix the RI, we set out to calculate the position of cross points. To better express the above line image, we obtain a continuous mathematic function to simulate the discrete image points using B-spline interpolation technique [47]. Following is the plot from the simulation function.



**Figure 3.5** Line data grey scale distribution

From the above plot, we can clearly see that around each cross point, there is an abrupt plunge on the left side and a big jump on the right. So here we use slope information to locate the cross point. First for the left edge, the point whose grey value has biggest negative slope among all the points in the RI, can be regarded as the left edge for cross point, that is

$$\text{if } d(g(a_{\min})) = \min(d(g(a)), a \in A), \quad (3.9)$$

$a_{\min}$  is the left edge position

Using the same method, we can found the right edge according to the following equation:

$$\text{if } d(g(a_{\max})) = \min(d(g(a)), a \in A), \quad (3.10)$$

$a_{\max}$  is the right edge position

So feature point can be considered to be in the middle of the line connected by its left and right edge point.

$$a_{fp} = \frac{1}{2} * (a_{max} + a_{min}), \quad a_{fp} \text{ is the cross point position} \quad (3.11)$$

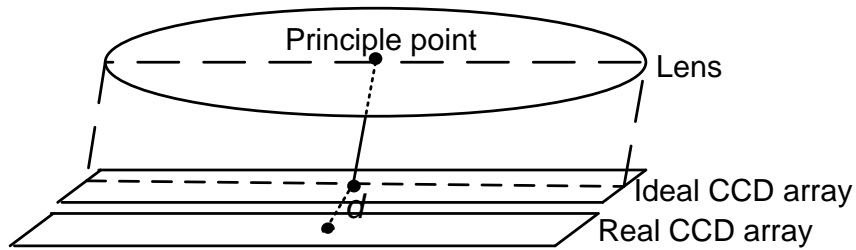
After the multiple times of iteration to get each RI then each cross point, we can obtain all the cross points of intersection between invisible view line and visible pattern line. Those cross points are of high importance for us to create correspondence from image coordinate to world coordinate. So we call them image feature points in the following parts.

### 3.2.4 1-D to 2-D mapping

After we have all the feature points in the calibration pattern and also their corresponding positions in the image coordinate. We can start the camera calibration. Among the much work done on matrix camera calibration, Zhang [37] and Tai [30] created great breakthrough in this field. And Zhang [37] is one more step ahead because his technique on camera calibration is more flexible and easier to use. But the biggest limitation is that the method is only applicable for area based camera calibration, not for line scan camera calibration. Also the current study of line scan camera is rare and incomplete. In this case, we should think about a method to create a set of data which is suitable for 2D camera calibration while does not change the information line scan camera expresses. When we come to this problem, we should know the fact that the positions of feature points in images only show the information along with the view line, which is the one dimensional data. That means, in the image captured with line scan camera, from stationary object, we only have the valid data in horizontal direction, data

in vertical direction is only the accumulation of time, and only to repeat the line data. The line data in horizontal direction is what the line scan camera captured on the very thin CCD slot.

While considering the lens for line scan camera, although we know the case in most cases, the principle point of the lens is not posed exactly on the CCD slot (see Figure 3.6). That means, the distortion of one dimensional image comes from two directions,  $X$  and  $Y$  axis. But practically the deviation of slot from principle in  $Y$  axis, defined as  $d$  is so small (normally less than 2 pixel) that the distortion (distortion coefficient defined as  $k$ ) caused in  $Y$  axis ( $k * d, k \ll 1$ ) could be neglected. In the result part, we will prove that the distortion caused by deviation is neglectable with solid value.

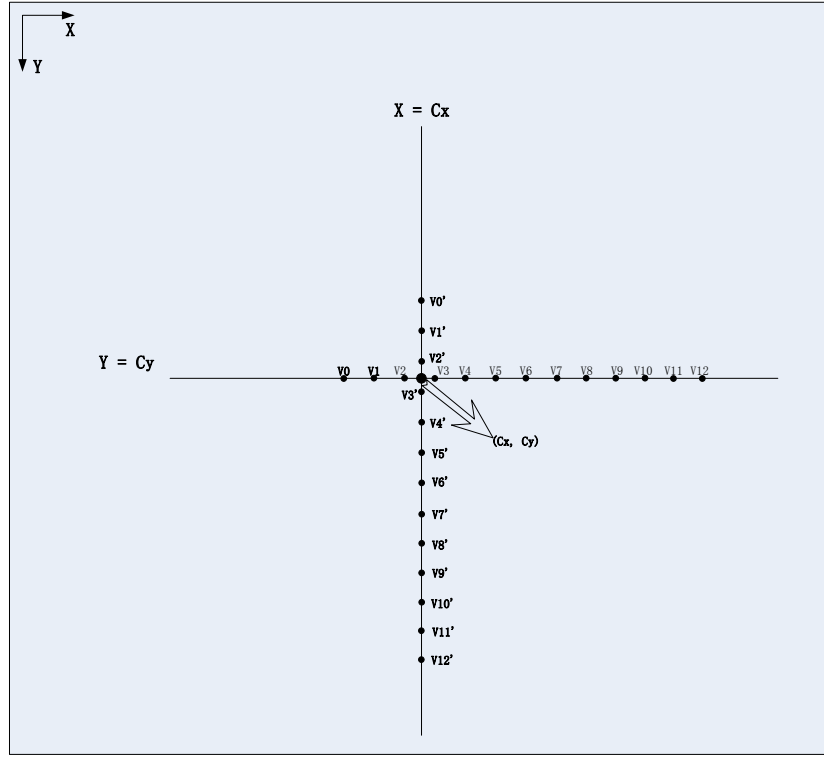


**Figure 3.6** Camera inner look

Before we start to settle this problem, it's necessary for us to understand what type of lens distortion occurs in our line scan camera. Lens distortion is the phenomena that prohibit use of simple pinhole camera models. Several types of lens distortion exist, including tangential and radial distortion. Due to the manufacturing technique, tangential distortion is very little and could be neglected. So here we only consider radial distortion. Radial distortion actually implies a nonlinear radius mapping from the object to the image: And the value of radius directly determines the distortion severity of the image projection from world objects.

After a general introduction of lens distortion, we continue the topic to solve the problem of 1-D calibration. Although the feature points on calibration have two dimensional positions, all the points are supposed to locate in the same scan line. That mean all object feature points are linear correlated. But the current area-based 2-D camera calibration method cannot accept the input data with linear correlation. That's the problem we need to fix, if we plan to apply the relative mature matrix camera calibration method to calibrate our line scan camera. We need to make our 1-D dimensional data have 2 dimensions and linear independent, while keeping results consistent for 1-D calibration.

Our solution is to create a 2-D image coordinate is to add another set of image data in perpendicular dimension which is exactly the same to the original line data captured. That means the lens distortion existing in the horizontal direction will exert the same effect to the vertical direction. According to the lens distortion principle, radial distortion is directly related to the radius, the distance from the image points to principle point, assumed as  $(C_x, C_y)$ . To make sure the created image data in vertical direction has the same radial distortion with that in horizontal direction, we need to put each created image points with as same distance to principle point as corresponding original image points. So the following picture illustrated is the 2-D image with original feature points in horizontal direction and the created feature points array in vertical direction.



**Figure 3.7** 2-D image data establishment

In this way, we successfully convert a 1-D image into 2-D, without changing the properties of the original image; we put the original line data on  $Y = C_y$  of the new 2-D image coordinate, and rotate the line clock wisely by 90 degree according to torsion center  $(C_x, C_y)$  to create another set of data  $X = C_x$ . In  $X = C_x$ , feature points  $V'_0, V'_1, V'_2, \dots, V'_{11}, V'_{12}$  have as the same distance to  $(C_x, C_y)$  as the corresponding points on  $Y = C_y, V_0, V_1, V_2, \dots, V_{11}, V_{12}$ , respectively. That is,

$$\begin{aligned} \forall i \in I, \exists (V'_i \cdot x - C_x)^2 + (V'_i \cdot y - C_y)^2 \\ = (V_i \cdot x - C_x)^2 + (V_i \cdot y - C_y)^2, \end{aligned} \quad (3.12)$$

where  $I$  is the set of two dimensional object feature points.

Therefore distortion effect should be the same due to the same distance to principle point  $(C_x, C_y)$ .

Since principle point  $(C_x, C_y)$  is very important for our technique of 1-D to 2-D mapping, It would be better if we have  $(C_x, C_y)$  available before the mapping. But  $(C_x, C_y)$  are 2 parameters in camera intrinsic matrix which we need to compute, there is no way to get exactly what  $(C_x, C_y)$  is before that calibration. So at first we assume  $(C_x, C_y)$  available and set an initial value as the center point of 2-D image we create. The size of CCD array of our line scan camera is 2048 pixels, so the size of 2-D image we create is 2048×2048 pixels. Thus the initial value of  $(C_x, C_y)$  is set as (1024, 1024). Then we use the least-square method to iteratively compute more accurate  $(C_x, C_y)$  value till we obtain the least square error. The steps in detail are the following.

First we use the initial value of  $(C_x, C_y)$  to do mapping and calibration, after the calibration and we get all the camera parameters including a new set of  $(C_x, C_y)$ . We replace the previous value with the newly obtained  $(C_x, C_y)$  and do mapping and calibration over and over again. The termination condition to end this iteration is that, if the current position of  $(C_x, C_y)$  has the least deviation and stays almost the same with the previous few values, the iteration can be terminated with the current value as the final principle point.

After we get 2-D image coordinate, we can use the same method to create 2-D object feature points by adding another set of data perpendicular to the scan line. Accordingly, we just need to rotate the object line by another 90 degree at the torsion center  $T(x, y, z)$ . Torsion center point could be obtained by equation (1), which connects the image coordinate with world coordinate, given the principle point  $(C_x, C_y)$  available.



To get the rotated data points( $x', y', z' = 0$ ), we define  $R$  as rotation matrix, and  $T$  as the translation vector. According to the transformation principle, we have,

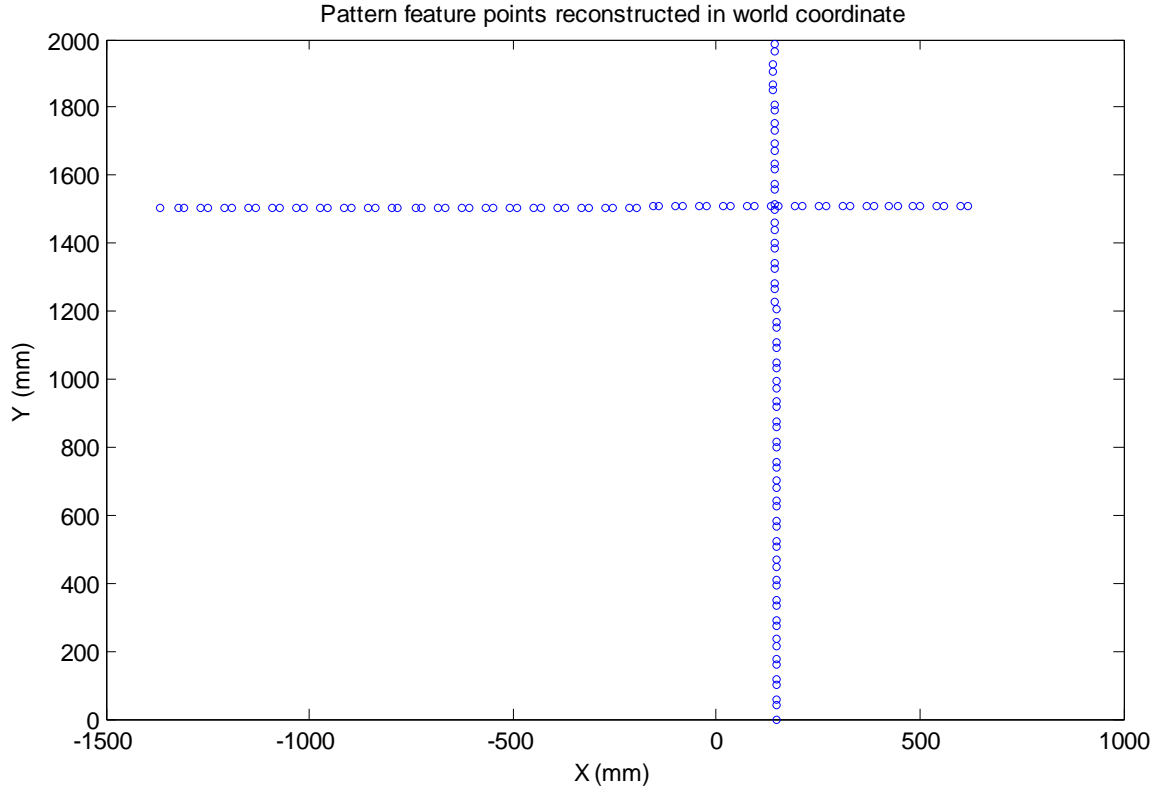
$$\begin{bmatrix} x \\ y \\ z = 0 \end{bmatrix} * R + T = \begin{bmatrix} x' \\ y' \\ z' = 0 \end{bmatrix} \quad (3.13)$$

Where Give the fixed rotation  $R$  is a rotation matrix converted from vector  $r(\alpha = 0, \beta = 0, \gamma = \frac{\pi}{2})$  by the following formula,

$$R = \begin{bmatrix} \cos\beta\cos\gamma & -\cos\alpha\sin\gamma + \sin\alpha\sin\beta\cos\gamma & \sin\alpha\sin\gamma + \cos\alpha\sin\beta\cos\gamma \\ \cos\gamma\sin\gamma & \cos\alpha\cos\gamma + \sin\alpha\sin\beta\sin\gamma & \sin\alpha\cos\gamma + \cos\alpha\sin\beta\sin\gamma \\ -\sin\beta & \sin\alpha\cos\beta & \cos\alpha\cos\beta \end{bmatrix} \quad (3.14)$$

Before we know( $x', y', z' = 0$ ), we should calculate translation  $T$  first, since  $R$  is available. To get  $T$ , we can get use of torsion center which is already know and is the only point which stays at the same position during the rotation. As we know that torsion center is the only point which keeps the same position during the rotation.

With the transformation formula, we can create another set of object feature points corresponding to the original ones. Following is the illustration of 2-D object feature point establishing.



**Figure 3.8** 2-D object data establishment

With the image coordinate and world coordinate feature points ready, we can move to the next step to calculate intrinsic and extrinsic parameters and distortion efficient of the camera.

### 3.2.5 Camera matrix calculation

Given the basic principle of a pinhole model, we have the relationship between 3D point  $M$  and its image projection  $m$  given by:

$$s * m = A[R \quad T]M, \text{ with } A = \begin{bmatrix} f_x & \alpha & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.15)$$

S refers to an arbitrary scale factor,  $(\mathbf{R}, \mathbf{T})$  is the extrinsic parameters in rotation and translation to project world coordinate system to the image coordinate system. A is called intrinsic matrix, with  $(u_0, v_0)$  as the principle point, and  $(f_x, f_y)$  as the focal length in  $u$  and  $v$  axis respectively, and  $\alpha$  as the skew value of the two image axis.

A 2-D image point is denoted by  $m = [u, v]^T$ , and  $M = [X, Y, Z]^T$ , to generalize the equation, we augment the vector of  $m$  into  $[u, v, 1]^T$  and  $M$  into  $[X, Y, Z, 1]^T$ , by denoting the  $i$ th column of the rotation matrix  $\mathbf{R}$  by  $\mathbf{r}_i$ , and considering the  $Z = 0$  for all the object points on the calibration pattern, we have the following equation.

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A \begin{bmatrix} r_1 & r_2 & r_3 & t \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = A \begin{bmatrix} r_1 & r_2 & T \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \quad (3.16)$$

Giving the knowledge that  $r_1$  and  $r_2$  are orthonormal, we have the following equations:

$$\begin{cases} r_1 \cdot r_2 = 0 \\ \|r_1\| = 1 \\ \|r_2\| = 1 \end{cases} \quad (3.17)$$

By using the above three constrains and applying least-square method, we solve the equation with the multiple groups of known image feature points  $[u, v]^T$  and object feature points  $[X, Y, Z = 0]^T$  and obtain the intrinsic matrix A and the extrinsic  $(R, T)$  of the camera.

To refine the camera matrix, we use the Maximun-likelihood estimation method proposed by Zhang [37]. Because the images we captured are interrupted by lens distortion and independent noise, and given  $n$  images of a model plane and there are  $m$  points on each model plane, we can decrease and lens distortion and noise by minimizing the following function:

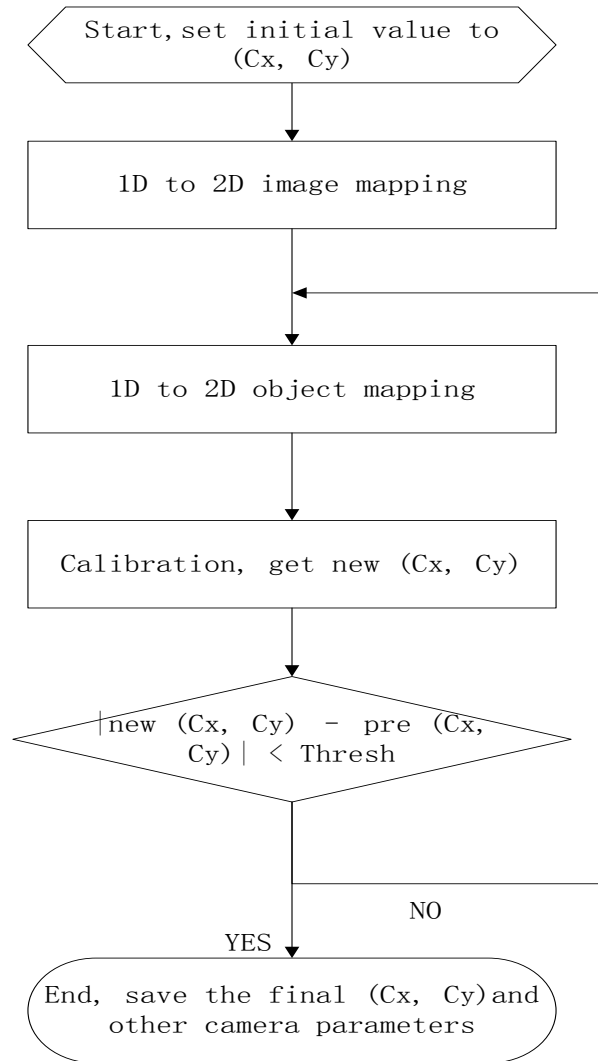
$$\sum_{i=1}^n \sum_{j=1}^m \|m_{ij} - \hat{m}(A, R_i, t_i, M_j)\|^2 \quad (3.18)$$

In the above formula,  $\hat{m}(A, R_i, t_i, M_j)$  is the reprojection of feature points  $M_j$  from world coordinate to image coordinate  $\hat{m}$  and  $m_{ij}$  is the image points captured by camera from model plane. Formula (15) is a nonlinear polynomial, to minimize (15), we should use Levenberg-Marquardt Algorithm [48] to solve the problem, with an initial guess of  $A, \{R_i, t_i | i = 1 \dots n\}$ .

By minimizing the formula (15), we can refine the camera matrix and also obtain the radial distortion  $k_1, k_2,$  and  $k_3,$

### 3.3 EXPERIMENTAL RESULTS

To summarize the iteration procedure we did for refining principle point  $(C_x, C_y)$  and calibration, we give the following diagram to show this procedure,



**Figure 3.9** Procedure diagram

Table 3.1 lists the intrinsic and extrinsic parameters from a successful calibration of line scan camera, with the reprojection error within 0.68 pixels. From this table we can find that  $C_x$  and  $C_y$  are not exactly the same although our feature point data in two directions is identical. The reason is that as we mentioned above, although the CCD slot is in one dimension, the lens designed for line scan camera is the same we use for area-based camera. So the position the principle point could be different in  $X$  and

$Y$  coordinate, when the one dimensional CCD slot doesn't pass the principle point of lens.

Also we assume that there is no tangential distortion in line scan camera due to the modern lens manufacture technique. Besides, as to one dimensional camera, the distortion happens in one direction of lens, which is radial distortion. Focal length is the same in both  $X$  and  $Y$  direction.

**Table 3.1** Camera intrinsic and extrinsic parameters

Camera Parameters	Value
$R_x$ (degree)	-0.008143
$R_y$ (degree)	0.008024
$R_z$ (degree)	-0.003570
$T_x$ (mm)	-1144.4522
$T_y$ (mm)	-68.9639
$T_z$ (mm)	343.9326
$f$ (mm)	261.6441
$\kappa_1$	0.003276
$\kappa_2$	0.000073
$k_3$	0
$c_x$ (pixel)	1021.6435
$c_y$ (pixel)	1022.4045

### 3.4 CONCLUSION AND DISCUSSION

In this part, we discussed the novel method of line scan camera calibration in detail for the fabric defects imaging project. Based on the complete understanding of the specialty of line scan camera, we design the pattern for its calibration. We create multiple parallel lines (around 58 lines) and 57 oblique line connecting each pair of neighboring

parallel lines to obtain around 115 feature point crossed by invisible scan line and visible pattern lines. The 115 image feature points and their corresponding object feature points can create 115 equations where only 14 parameters unknown. The large numbers of equations are sufficient to obtain the parameters from pinhole model with high accuracy. With a linear least-square equation, we estimated the parameters which are refined by maximum likelihood estimation, we obtain all the parameters with the reprojection error only 0.68 pixel.

## Chapter 4: Image Undistortion and Correction

Image quality is greatly influenced by lens distortion. One of the most common visual distortions is radial distortion. This occurs when the magnification of the lens at the edge is different from that at the center of lens. Chapter 3 has shown us the method to obtain distortion coefficient through camera calibration. In this chapter, we'll discuss the procedure to correct the images using distortion coefficients.

### 4.1 RADIAL DISTORTION EQUATION

#### 4.1.1 2-D camera radial distortion

Before we set out studying radial distortion equation, it's necessary to review the projection relationship from 3D world coordinate to the 2D image coordinate, according to the pinhole camera principle. See the following equation (4.1),

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = A[R|T] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (4.1)$$

Where  $\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$  is a point from world coordinate, and  $\begin{bmatrix} u \\ v \end{bmatrix}$  is a corresponding pixel point from image coordinator;  $A = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix}$ , where  $(\alpha, \beta)$  is the focal length of camera in X and Y axis, and  $(u_0, v_0)$  is camera lens principle point;

$R$  is a  $3 \times 3$  rotation matrix converted from rotation vector  $r(\delta, \varepsilon, \theta)$ , as the equation (12) has shown us. To simply the rotation matrix here, we use  $R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{23} & r_{33} \end{bmatrix}$

$T$  is a  $1 \times 3$  translation vector  $T(t_1, t_2, t_3)$ .



So the equation (4.1) can be rewritten as following:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{23} & r_{33} & t_3 \end{bmatrix} \times \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (4.2)$$

Considering the  $Z = 0$  for all the object points on the calibration pattern, we have,

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{23} & t_3 \end{bmatrix} \times \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \quad (4.3)$$

To calculate the right side the above equation, we get,

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha(r_{11}X + r_{12}Y + t_1) + \gamma(r_{21}X + r_{22}Y + t_2) + u_0(r_{31}X + r_{32}Y + t_3) \\ \beta(r_{21}X + r_{22}Y + t_2) + v_0(r_{31}X + r_{32}Y + t_3) \\ r_{31}X + r_{32}Y + t_3 \end{bmatrix} \quad (4.4)$$

First let's do  $v / 1$ , and have,

$$v = \frac{\beta(r_{21}X + r_{22}Y + t_2)}{r_{31}X + r_{32}Y + t_3} + v_0 \quad (4.5)$$

Then  $u / 1$

$$u = \frac{\alpha(r_{11}X + r_{12}Y + t_1) + \gamma(r_{21}X + r_{22}Y + t_2)}{r_{31}X + r_{32}Y + t_3} + u_0 \quad (4.6)$$

Equation (4.5) and (4.6) show us the ideal pinhole projection from world coordinate to image coordinator for X and Y axis, without taking account lens distortion. But in real case, lens distortion always happens. Distortion makes the deviation from the original image point. We define the deviation as  $(d_x, d_y)$ . And the deviated image point  $(u', v')$  is show as the following equation,

$$\begin{cases} u' = u + d_x, \\ v' = v + d_y \end{cases} \quad (4.7)$$

As we've known that radial distortion just means each point is radially distorted and the degree of distortion depends on the distance from the principle point of lens or image center to the certain point. According to the research of Weng [31], we have the following equation,

$$\begin{cases} d_x = (u - u_0) \times (k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots) \\ d_y = (v - v_0) \times (k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots) \\ r^2 = (u - u_0)^2 + (v - v_0)^2 \end{cases} \quad (4.8)$$

where  $k_j, j \in \{1, 2, 3, \dots\}$  is the coefficient of radial distortion. Its value affects how much a point is radially distorted and its sign indicated the type of radial distortion. If  $k_j$  is negative, the resultant distortion is barrel radial distortion; if  $k_j$  is positive, the resultant distortion is pincushion radial distortion.

#### 4.1.2 1-D camera radial distortion

As to 1-D camera, the information in  $Y$  axis is only time accumulation because the one dimensional CCD array is in  $X$  direction. In each line the image points are distorted with a ratio directly influenced by their distance to principle points. And as we discussed in the last chapter that the distance from CCD array to principle point in  $Y$  direction is so small that could be neglected. So the radial distortion in line scan camera happens only in  $X$  direction. That is  $d_y = 0$ ;

The one dimensional radial distortion can also be explained in another way. Since line scan camera only has one dimensional CCD array in  $X$  direction, so the data in  $Y$  direction is meaningless, only showing time accumulation. So,

$$v = v_0$$

And rewrite equation (4.8), we have

$$\begin{cases} d_x = (u - u_0) \times (k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots) & (1) \\ d_y = 0 & (2) \\ r^2 = (u - u_0)^2 & (3) \end{cases} \quad (4.9)$$

As to  $k_j$ , for practical usage and according to Tsai's theory [30],  $k_j, (j > 3)$  is extremely small that we can safely approximate radial distortion by using the first three items of the infinite series of (4.9)-(1). Since we have calculated the radial distortion coefficient  $k_j$  in the last chapter, we'll discuss how to correct the distorted images in the next section.

## 4.2 DISTORTED IMAGE CORRECTION

According to the equation (4.7) and (4.9) in the above section, we have,

$$\begin{aligned} u' &= u + d_x = u + (u - u_0) \times (k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots) \\ &= u - u_0 + u_0 + (u - u_0) \times (k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots) \\ &= (u - u_0) \times (1 + k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots) + u_0 \end{aligned} \quad (4.10)$$

To calculate the corrected point value  $u$ , we rewrite the above equation, and put  $u$  on the left side of the equation.

$$u = \frac{u' - u_0}{1 + k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots} + u_0 \quad (4.11)$$

Here we only consider the first four items of the infinite series of denominator, so the above equation turns to,

$$u = \frac{u' - u_0}{1 + k_1 r^2 + k_2 r^4 + k_3 r^6} + u_0 \quad (4.12)$$

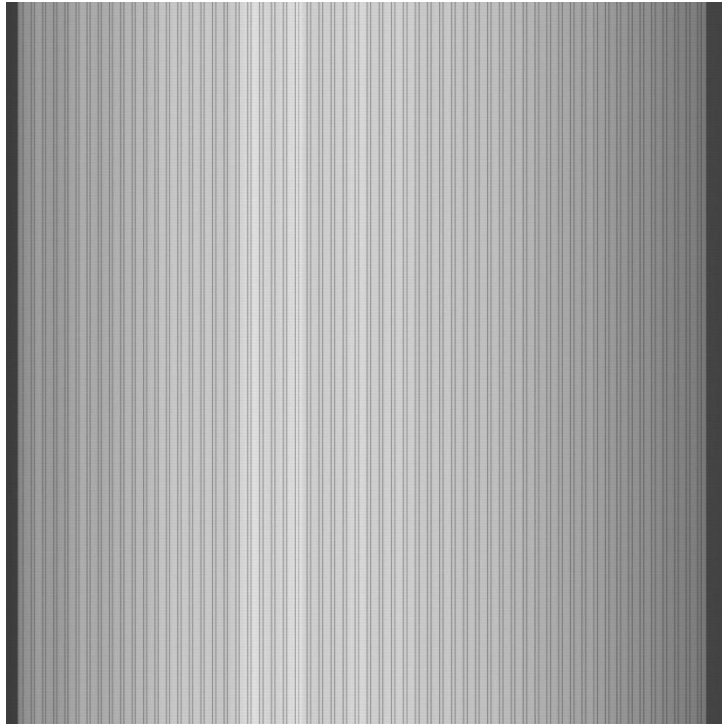
Specifically, we know that each point in the distorted image corresponds to a point in the undistorted image. Given the above equation, we can easily find the corresponding undistorted point from distorted point.

Due to the calculated undistorted points are not integers. We first use B-spline interpolation method to get evenly distributed sub-pixel points, and then apply 1/10 sampling technique to pick out the image points whose fractional part is zero, and finally fit these points to the desired undistorted image coordinate.

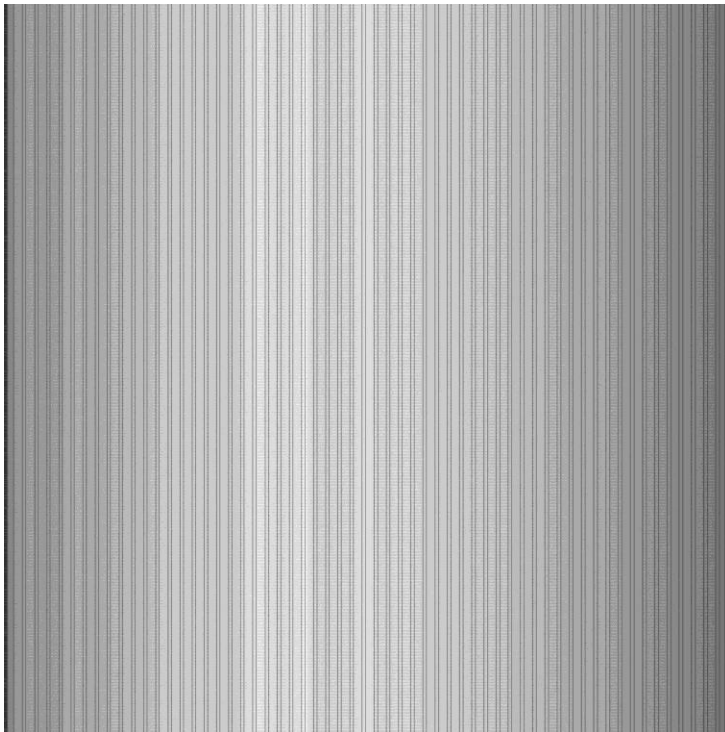
In the next part we'll show image correction results and give relative discussion.

### **4.3 IMAGE CORRECTION RESULT**

After image correction, we obtain the undistorted image in Figure 4.2. Compared to the originally captured image in Figure 4.1, the objects in corrected image become wider in the first impression.



**Figure 4.1** Originally captured image



**Figure 4.2** Corrected image

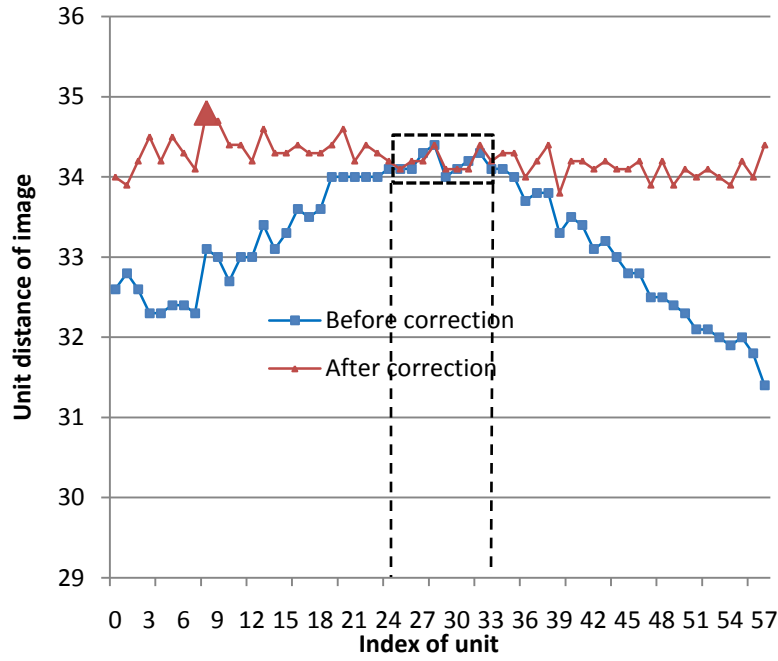
The reason behind is that the lens of our line scan camera cause barrel distortion, the farther image pixel to principle point, the severer distortion. So image points on the boundary have greatest tendency to approach to the principle point. As to the pattern we designed, distance between each pair of parallel lines is identical. But from the image coordinate aspect, when we look at the captured image Figure 4.1, from the left side (define the first line index as 0) and to right side and calculate the distance of neighboring even-index parallel lines. Here we skip all odd-index parallel lines because the odd-index lines are the projection from world coordinate where intersection happens between oblique line and scan line, we have no way to prove the distance between neighboring oblique lines are the same due to the fact that view line is not always horizontally straight. So here we only consider the distance of even-index parallel lines, call Unit Distance (UD) of image, which is also image projection of  $P_0P_2, P_2P_4, \dots, P_{2n-2}P_{2n}, \dots$  (see Figure 3.1) in world coordinate. Following is the Table 4.1 to list all UD's detected from left to the right side of one image before and after image lens correction.

**Table 4.1** Image UD before and after correction

Unit	UD		Unit	UD		Unit	UD	
index	Before	After	index	Before	After	index	Before	After
0	32.6	34	20	34	34.6	40	33.5	34.2
1	32.8	33.9	21	34	34.2	41	33.4	34.2
2	32.6	34.2	22	34	34.4	42	33.1	34.1
3	32.3	34.5	23	34	34.3	43	33.2	34.2
4	32.3	34.2	24	34.1	34.2	44	33	34.1
5	32.4	34.5	25	34.1	34.1	45	32.8	34.1
6	32.4	34.3	26	34.1	34.2	46	32.8	34.2
7	32.3	34.1	27	34.3	34.2	47	32.5	33.9
8	33.1	34.8	28	34.4	34.4	48	32.5	34.2
9	33	34.7	29	34	34.1	49	32.4	33.9
10	32.7	34.4	30	34.1	34.1	50	32.3	34.1
11	33	34.4	31	34.2	34.1	51	32.1	34
12	33	34.2	32	34.3	34.4	52	32.1	34.1
13	33.4	34.6	33	34.1	34.2	53	32	34
14	33.1	34.3	34	34.1	34.3	54	31.9	33.9
15	33.3	34.3	35	34	34.3	55	32	34.2
16	33.6	34.4	36	33.7	34	56	31.8	34
17	33.5	34.3	37	33.8	34.2	57	31.4	34.4
18	33.6	34.3	38	33.8	34.4			
19	34	34.4	39	33.3	33.8			

According to Table 4.1, we draft a diagram to visually show the validity of our correction method. In Figure 4.3, the blue line with square dots refers to image UD before image correction, from this plot we can see that on the center of image there is little lens

distortion and UD remains high at around 34 pixels, but beyond the center region, the UD keep decreasing, with a deviation around 3 pixels in two edges. But after the image being corrected, as the red line in the diagram with triangular dots shows us, most of the triangular dots are around 34 pixels. The points surrounded by black rectangular, from unit index 25 to index 33, are the coincidence of two lines. They indicate that image points within this region are captured without distortion. We defined the corresponding UD as distortion-free UD  $d_{dfree}$ . Based upon the data listed in Table 4.1, we calculate that  $d_{dfree} = 34.16 \text{ pixel}$ .



**Figure 4.3** UD before and after correction

For the distorted image (the blue line), the average UD, is defined as  $d_{pre}$ , and  $d_{pre} = 33.17 \text{ pixel}$ , with the standard deviation,  $STD_{pre} = 0.79 \text{ pixel}$ . The overall deviation of UD to distortion-free UD is  $OVD_{pre} = 1.601 \text{ pixel}$ .



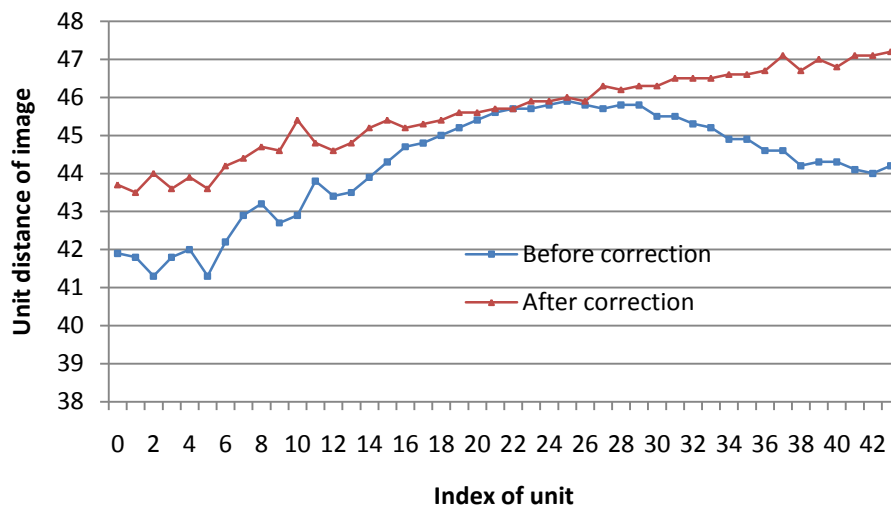
As to the corrected image (the red line), the average UD, is defined as  $d_{post}$ , and  $d_{post} = 34.22 \text{ pixel}$ , with the standard deviation,  $STD_{post} = 0.20 \text{ pixel}$ . The overall deviation of UD to distortion-free UD is  $OVD_{post} = 0.043 \text{ pixel}$ .

To testify the precondition that image plane should be parallel to the pattern plane is not necessary, we captured a couple of images with camera skewed to a certain angle. Using the same method we obtain the UD from captured image before image correction and compare the results with that after correction. The following lists all the UD data

**Table 4.2** UD before and after correction (with a skewed camera plane)

Unit	UD		Unit	UD		Unit	UD	
index	Before	After	index	Before	After	index	Before	After
0	41.9	43.7	15	44.3	45.4	30	45.5	46.3
1	41.8	43.5	16	44.7	45.2	31	45.5	46.5
2	41.3	44	17	44.8	45.3	32	45.3	46.5
3	41.8	43.6	18	45	45.4	33	45.2	46.5
4	42	43.9	19	45.2	45.6	34	44.9	46.6
5	41.3	43.6	20	45.4	45.6	35	44.9	46.6
6	42.2	44.2	21	45.6	45.7	36	44.6	46.7
7	42.9	44.4	22	45.7	45.7	37	44.6	47.1
8	43.2	44.7	23	45.7	45.9	38	44.2	46.7
9	42.7	44.6	24	45.8	45.9	39	44.3	47
10	42.9	45.4	25	45.9	46	40	44.3	46.8
11	43.8	44.8	26	45.8	45.9	41	44.1	47.1
12	43.4	44.6	27	45.7	46.3	42	44	47.1
13	43.5	44.8	28	45.8	46.2	43	44.2	47.2
14	43.9	45.2	29	45.8	46.3			

The following Figure 4.4 is the corresponding diagram to show the difference. Although the camera plane is skewed and the correct UD doesn't remain constant, we still can see the proposed method has successfully removed the radial lens distortion, and pull the distorted points (the blue line with square points) all the way to the right track. The corrected points are almost linearly located, as shown in the red line with triangular points.



**Figure 4.4** UD before and after image correction (with a skewed camera plane)

Using SPSS to do the regression analysis, we find our hypothesis of linearity is significant with R Square larger than 0.95, as the following Table 4.3 shows us.

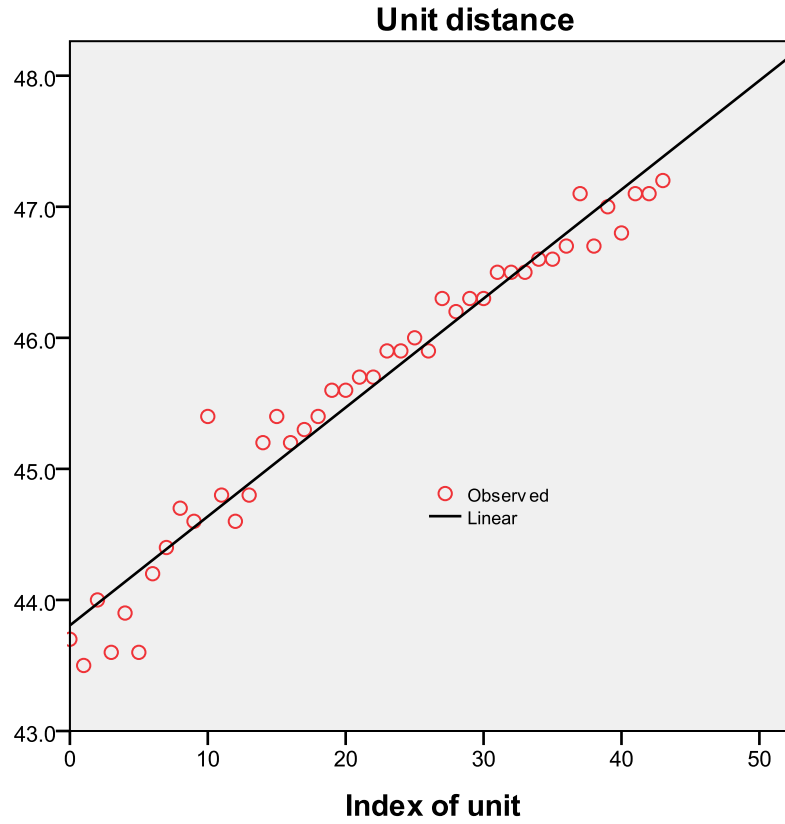
**Table 4.3** Model summary and parameter estimations

Dependent Variable: Y

Equation	Model Summary					Parameter Estimates	
	R Square	F	df1	df2	Sig.	Constant	b1
Linear	.955	899.313	1	42	.000	43.806	.083

The independent variable is X.

And using the curve fitting method, we obtain the curve to simulate the corrected points, shown in Figure 4.5.



**Figure 4.5** Linear simulation

To test the proposed method, we capture 4 images, with 2 images captured from a parallel camera plane and the other two from a skewed camera plane. Table 4.4 is the summary of test results for parallel camera plane.

**Table 4.4** Test result summary

Image	Distortion free	Average UD		UD Standard deviation		UD Overall deviation		
No.	UD	$d_{dfree}$	$d_{pre}$	$d_{post}$	$STD_{pre}$	$STD_{post}$	$OVD_{pre}$	$OVD_{post}$
1	34.16		33.17	34.22	0.79	0.20	1.601	0.043
2	34.21		33.16	34.23	0.85	0.37	1.796	0.139
Mean	33.19		33.17	34.23	0.82	0.29	1.90	0.09

From the above table, we can see that both of the images are corrected with high accuracy. The average deviation from distortion free UD is only 0.09 pixel. Compared to original image without correction, whose deviation is nearly 2 pixels, the quality of corrected image has been greatly modified with the proposed method.

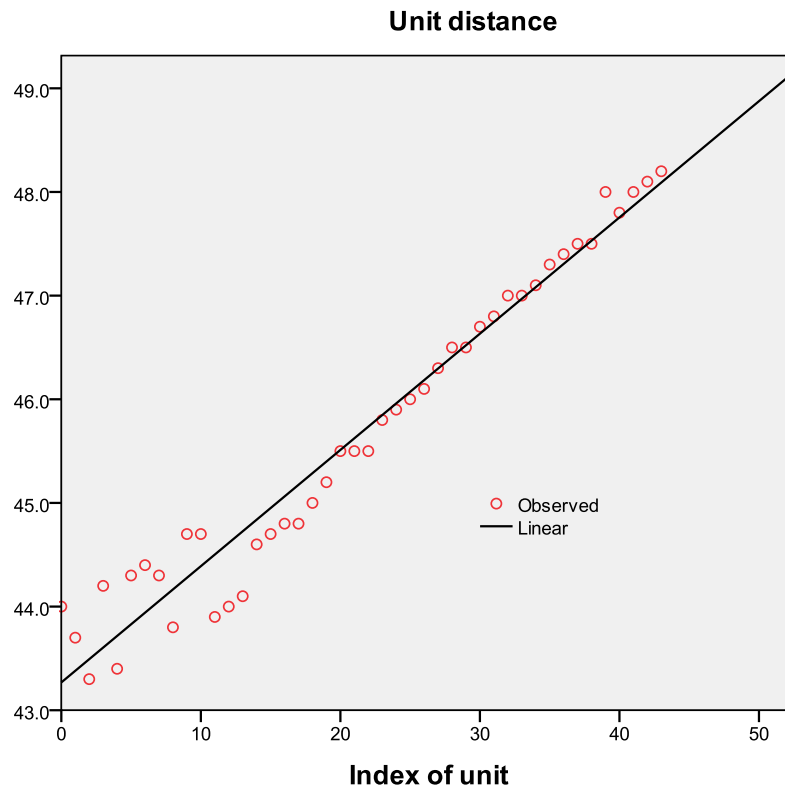
As to the images captured from skewed camera plane, our correction algorithm still works in rectifying the image. We've illustrated one image result and given detailed discussion. To show the consistency of the result, we provide linearity analysis to another corrected image, as shown in the following Table 4.5 and Figure 4.6.

**Table 4.5** Model summary and parameter estimates

Dependent Variable:Y

Equation	Model Summary					Parameter Estimates	
	R Square	F	df1	df2	Sig.	Constant	b1
Linear	.958	950.633	1	42	.000	43.268	.112

The independent variable is X.



**Figure 4.6** Linear simulation

Compare to the first image, the linearity of the second is slightly higher with R Square 0.958. And the linear slope is also higher at 0.112 pixel per unit. That means UD increases more when index increases, which indicates that the skewed angle in the second image is larger than that of the first captured image.

## Chapter 5: Conclusion

Quality is a very important element to evaluate fabric. Large numbers of scholars have been dedicated in the research of fabric defects inspection and made substantiate progress in this field. Many vision-based automatic fabric detection algorithms have been proposed to detect fabric flaw efficiently and with a high accuracy. To secure the accuracy of detection result, image quality should be reliable before we apply any kind of detection methods. But as we know that all camera lens has distortions, especially radial distortion. To remove that distortion and correct image, much research work has been done to do 2-D camera image correction. However as to 1-D line scan camera, there is scant information talking about camera correction although line scan camera is getting wider application due to the high resolution and data processing efficiency.

A novel calibration-based line scan camera correction method is proposed in this study. Same as 2-D camera calibration, pattern design is a vital step before calibration. Pattern with regular marks can be projected to camera plane, and these marks from both pattern plane and image plane can create relationship between world coordinate and image coordinate. But base upon the specialty of line scan camera that it acquires images one line at a time, it's difficult for the one scan line to match the marked points on pattern. So we propose a pattern with many mutual parallel lines and one oblique line to connect each pair of the parallel ones. We consider the intersection points between pattern lines and scan lines as feature points and calculate the position according to the pattern geometry.

As 2-D camera calibration has been greatly researched and the algorithms and techniques to deal with 2-D calibration are relatively mature. We propose a method to

calibrate 1-D camera from 2-D aspect by mapping all 1-D data to 2-D without changing the original properties of 1-D data.

When we have the 2-D data ready, we use least-square method to solve the pinhole projection equation and estimate the values of camera internal parameters and external parameters. Then we refine the data with maximum-likelihood estimation and get the camera lens distortion coefficients, with reprojection error only 0.68 pixel. With the distortion coefficients ready, the distortion equation is applied and modified to obtain undistortion equation. And finally we use modified undistortion equation to correct captured images.

To better testify the undistortion results and prove the validity of proposed method, we introduce a term of unit distance in assessing procedure. In the discussion part, we can see that corrected image has almost identical unit distance with standard deviation only 0.29 pixels. Compare to the ideal distortion-free unit distance, the corrected image has only 0.09 pixel in average apart. To prove the generality of our method, we capture a couple of images with skewed camera plane, the results given in discussion part are also positive to shown the effectiveness of the proposed method.

## Bibliography

- [1] R. Stojanovic and P. Mitropulos, "Real-time vision-based system for textile fabric inspection," *Real-Time Imaging*, vol. 7, no. 6, pp. 507-518, 2001.
- [2] C. S. Cho, B. M. Chung and M. J. Park, "Development of real-time vision-based fabric inspection system," *IEEE Transaction on Industrial Electronics*, vol. 52, no. 4, pp. 1073-1079, 2005.
- [3] M. Takato, T. Mori and Y. Takagi, "Automated Fabric Inspection Using Image Processing Techniques.," *Automated Inspection and High Speed Architectures*, vol. 104, pp. 151-158, 1988.
- [4] X. F. Zhang and R. Bresee, "Fabric Defect Detection and Classification Using Image Analysis," *Textile Res. J.*, vol. 65, no. 1, pp. 1-9, 1995.
- [5] J. S. Lane and S. C. Moure, "Textile Fabric Inspection System," *U.S. Patent 5774177*, 1998.
- [6] L. Northon, M. Bradshaw and A. Jewell, "Machine Vision Inspection of Web Textile Fabric," *Proc. Brithish Machine Vision Conference*, pp. 217-226, 1992.
- [7] A. T. Alam Eldin and H. A. Nour Eldin, "Automated Visiual Inspection of Uniformly Textured Flat Surfaces Using Correlation Analysis," *Proc. of the IASTED International Symposium on Applied Control, Filtering, and Signal Processing*, 1987.
- [8] S. Ribolzi, J. Merckle and J. Gresser, "Real-Time Fault Detection on Textiles using opto-Electronic Processing," *Textile Res. J.*, vol. 63, no. 2, pp. 61-71, 1993.
- [9] E. Shady, "A Computer Vision System for Automated Inspection of Fabrics," *Mater Thesis, Mansoura University*, 1998.
- [10] I. Tsai and M. Hu, "Automatic Inspection of Fabric Defects using an Artifical Neural Network Technique," *Textile Res. J.*, vol. 66, no. 7, pp. 474-482, 1996.
- [11] H. Sari-Sarraf and J. Goddard, "On-line Optical Measurement and Monitoring of Yarn Density in Woven Fabrics," *Automat. Opt. Inspect. Ind. SPIE*, vol. 2899, pp. 444-452, 1996.
- [12] B. Xu, "Identifying Fabric Structures with Fast Fourier Transform Techniques," *Textile Res. J.*, vol. 66, no. 8, pp. 496-506, 1996.
- [13] W. Jasper and H. Potlapalli, "Image Analysis of Mispicks in Woven Fabric," *Textile Res. J.*, vol. 65, no. 11, pp. 683-692, 1995.
- [14] H. Sari-Sarraf and J. Goddard, "Vision System for On-Loom Fabric Inspection," *IEEE Transactions on Industry Applications*, vol. 35, no. 6, November 1999.
- [15] C. Ciamberlini, F. Francini, P. Sansoni and B. Tiribilli, "Defect Detection in Textured Materials by Optical Filtering with Structured Detectors and Self-adaptable



- Masks," *Opt. Eng.*, vol. 35, no. 3, pp. 838-844, 1996.
- [16] C.-h. Chan and G. K. H. Pang, "Fabric Defect Detection by Fourier Analysis," *IEEE Transactions on Industry Applications*, vol. 36, no. 5, September 2000.
  - [17] Y. D. Cai, X. J. Liu and K. C. Chou, "Artificial Neural Network Model for Predicting Membrane Protein Types.," *J. Biomol. Struct. Dyn.*, vol. 18, pp. 607-610, 2001.
  - [18] M. C. Hu and I. S. Tsai, "Fabric Inspection Based on Best Wavelet Packet Bases," *Textile Res. J.*, vol. 70, no. 8, pp. 662-670, 2000.
  - [19] B. S. Jeon and J. H. Bae, "Automatic Recognition of Woven Fabric Patterns by an Artificial Neural Network," *Textile Res. J.*, vol. 73, no. 7, pp. 645-650, 2003.
  - [20] R. P. Lippmann, "An Introduction to Computing with Neural Nets," *IEEE ASSP Magazine*, vol. 4, no. 2, pp. 4-22, April 1987.
  - [21] S. Rajasekaran, "Training-Free Counter Propagation Neural Network for Pattern Recognition of Fabric Defects," *Textile Res. J.*, vol. 67, no. 6, pp. 401-405, 1997.
  - [22] P. Borzone, S. Carosio and A. Durante, "Detecting Fabric Defects with a Neural Network Using Two Kinds of Optical Patterns," *Textile Res. J.*, vol. 72, no. 6, pp. 545-550, 2002.
  - [23] E. Shady, Y. Gawayed, M. Abouiana, S. Youssef and C. Pastore, "Detection and Classification of defects in Knitted Fabric Structures," *Textile Res. J.*, vol. 76, no. 4, pp. 295-300, 2006.
  - [24] I. S. Tsai and M. C. Hu, "Automatic Inspection of Fabric Defects using an Artificial Neural Network Technique," *Textile Res. J.*, vol. 66, no. 7, pp. 474-482, 1996.
  - [25] O. Al-allaf, "Fast BPNN Algorithm for Reducing Convergence Time of BPNN Image Compression," *Proceedings of the 5th International Conference on IT & Multimedia at UNITEN(ICIMU 2011)*, November 2011.
  - [26] P. W. Chen, T. C. Liang, H. F. Yau, W. L. Sun and N. C. Wang, "Classifying Textile Faults with a Back-propagation Neural Network using Power Spectra," *Textile Res. J.*, vol. 68, no. 2, pp. 121-126, 1998.
  - [27] Y. R. Shiau, I. S. Tsai and C. S. Lin, "Classifying Web Defects with a Back-Propagation Neural Network by Color Image Preprocessing," *Textile Res. J.*, vol. 70, no. 7, pp. 633-640, 2000.
  - [28] Y. Yin, W. B. Lu, K. Zhang and L. Jing, "Textile Flaw Detection and Classification by Wavelet Reconstruction and BP Neural Network," *Gload Congress on Interlligent System*, vol. 284, 2009.
  - [29] W. K. Wong, C. W. M. Yuen, D. D. Fan and E. H. K. Fung, "Stitching Defect Detection and Classification using Wavelet Transform and BP Neural Network," *Expert Systems with Applications*, vol. 36, pp. 3845-3856, 2009.
  - [30] J. Pers and S. Kovacic, "Nonparametric, Model-based Radial lens Distortion Correction Using Tilted Camera Assumption," *Proceedings of the Computer Vision*

*Winter Workshop*, pp. 286-295, 2002.

- [31] K. Atkinson, "Close Range Photogrammetry and Machine Vision," *Whittles Publishing*, 1996.
- [32] R. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE Journal of Robotics and Automation*, vol. 4, no. 3, pp. 323-344, 1987.
- [33] J. Weng, P. Cohen and M. Herniou, "Camera Calibration with Distortion Models and Accuracy Evaluation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 10, pp. 965-980, 1992.
- [34] W. Faig, "Calibration of Close-Range Photogrammetry Systems: Mathematical Formulation," *Photogrammetric Eng. and Remote Sensing*, vol. 41, no. 12, pp. 1479-1486, 1975.
- [35] P. Sturm and S. Maybank, "On Plane-based Camera Calibration: A General Algorithm, Singularities, Applications," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 432-437, 1999.
- [36] V. Fremont and R. Chellali, "Direct Camera Calibration Using Two Concentric Circles from a Single View," *Proc. Int'l Conf. Artificial Reality and Telexistence*, pp. 93-98, 2002.
- [37] G. Gurdjos, A. Crouzil and R. Payrissat, "Another Way of Looking at Plane-based Calibration: The Center Circle Constraint," *Proc. European Conf. Computer Vision*, vol. 4, pp. 252-266, 2002.
- [38] J. Heikkila, "Geometric Camera Calibration Using Circular Control Points," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1066-1077, 2000.
- [39] Z. Zhang, "A flexible New Technique for Camera Calibration," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330-1344, 2000.
- [40] M. Pollefeys, R. Koch and L. Gool, "Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameter," *International Journal of Computer Vision*, vol. 32, no. 1, pp. 7-25, 1999.
- [41] R. Hartley, "Self-calibration from Multiple Views with a Rotating Camera," *In European Conference on Computer Vision*, pp. 471-478, 1994.
- [42] D. Brown, "Close-range Camera Calibration," *Photogrammetric Engineering*, vol. 37, no. 8, pp. 855-866, 1971.
- [43] Z. Zhang, "Camera Calibration with One-dimensional Objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 7, pp. 892-899, 2004.
- [44] W. I. Grosky and L. A. Tamburino, "A Unified Approach to the Linear Camera Calibration Problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 663-671, 1990.
- [45] C. A. Luna, M. Mazo, J. L. Lazaro and J. F. Vazquez, "Calibration of Line-Scan

- Cameras," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 8, pp. 2185-2190, 2010.
- [46] R. Horaud, R. Mohr and B. Lorecki, "On Single-scanline Camera Calibration," *IEEE Trans. Robot. Autom.*, vol. 9, no. 1, pp. 71-75, 1993.
- [47] K. Hirahara and K. Ikeuchi, "Detection of Street-parking Vehicles Using Line Scan Camera and Scanning Laser Range Sensor," *Proc. IEEE Intell. Veh. Symp.*, pp. 656-661, 2003.
- [48] C. A. Luna, M. Mazo and J. L. Lazaro, "Method to Measure the Rotation on Angles in Vibrating Systems," *IEEE Trans. Instrum. Meas.*, vol. 55, no. 2, pp. 232-239, 2006.
- [49] S. W. Lee and J. K. Paik, "Image Interpolation Using Adaptive Fast B-spline Filtering," *IEEE Proc. on Acoustics, Speech, and Signal Processing*, vol. 5, pp. 27-30, 1993.
- [50] J. More, "The Levenberg-Marquardt Algorithm, Implementation, and Theory," *Numerical Analysis*, 1977.