



**A Measurement of Pitch Angle and Roll Angle of Hard Disk Head in the HSA
using Image Processing Techniques**

Fu Dongjin

**A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of
Master of Engineering in Computer Engineering**

Prince of Songkla University

2010

Copyright of Prince of Songkla University

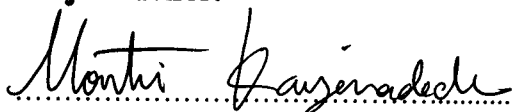
i

Thesis Title A Measurement of Pitch Angle and Roll Angle of Hard Disk Head in the HSA using Image Processing Techniques

Author Mr. Fu Dongjin

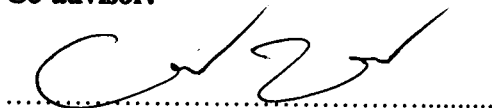
Major Program Computer Engineering

Major Advisor:



(Assoc. Prof. Dr. Montri Karnjanadecha)

Co-advisor:

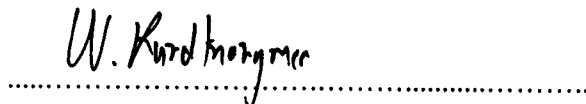


(Dr. Anant Choksuriwong)

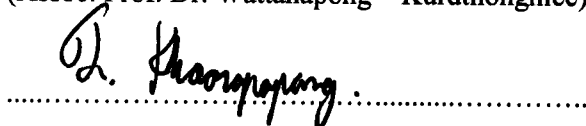
Examining Committee:

P. PhukpattaranontChairperson

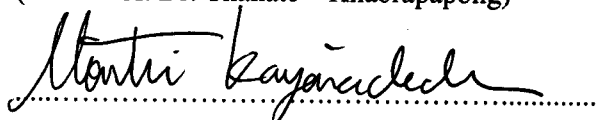
(Asst. Prof. Dr. Pornchai Phukpattaranont)



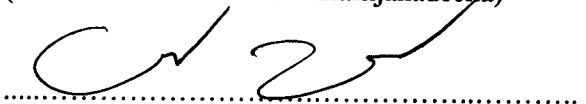
(Assoc. Prof. Dr. Wattanapong Kurdthongmee)



(Asst. Prof. Dr. Thanate Khaorapapong)

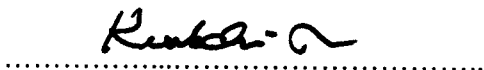


(Assoc. Prof. Dr. Montri Karnjanadecha)



(Dr. Anant Choksuriwong)

The Graduate School, Prince of Songkla University, has approved this thesis as partial fulfillment of the requirements for the Master of Engineering Degree in Computer Engineering.



(Assoc. Prof. Dr. Kerkchai Thongnoo)

Dean of Graduate School

Thesis Title A Measurement of Pitch Angle and Roll Angle of Hard Disk Head in the HSA using Image Processing Techniques
Author Mr. Fu Dongjin
Major Program Computer Engineering
Academic Year 2010

ABSTRACT

Since the floating height between a Hard Disk Head (HDH) and platter is very narrow, we must make sure that both are parallel with the horizontal plane in the Head Stack Assembly (HSA). It is not easy to make the HDH parallel because the HDH is small size, and the HDH's pitch angle and roll angle can only vary between -0.05° and 0.05° .

Currently, laser measurement system is used to measure HDH's pitch and roll angles in production-line. Laser system can offer good resolution, but it is expensive. Therefore, we develop an image processing system to measure these angles with acceptable resolution.

The image processing measurement system begins by obtaining clear images of an HDH using high-quality camera and microscope, and the subsequent processings include image cropping to crop useless area, super resolution to obtain higher resolution image, image enhancement to enhance image, edge detection to detect edge, robust contour following method to find the main edge, edge auto-segmentation to segment edge automatically, and line fitting to compute angle.

The resolution of the proposed system is 0.0226° for the pitch angle, and 0.0288° for the roll angle, according to our current experimental environment.

Keywords: hard disk head, HSA, pitch angle, roll angle, image processing

ACKNOWLEDGEMENT

First of all, this work is partially supported by Graduate School and Faculty of Engineering, Prince of Songkla University.

I would like to extend my sincere gratitude to my supervisor, Assoc. Prof. Dr. Montri Karnjanadecha, for his patience, encouragement, and professional instructions on this thesis. My co-advisor, Dr. Anant Choksuriwong, has helped and given several good suggestions on my work.

High tribute shall be paid to the internal committees Asst. Prof. Dr. Pornchai Phukpattaranont, Asst. Prof. Dr. Thanate Khaorapong, Assoc. Prof. Dr. Montri Karnjanadecha, and Dr. Anant Choksuriwong, also to the external committee Assoc. Prof. Dr. Wattanapong Kurdthongmee. They have drawn out their precious time for my thesis.

I feel grateful to all the lecturers in Prince of Songkla University who once offered me valuable courses and advice during my study. In addition, great thanks should to Dr. Andrew Davison for English correction partly on this report.

I feel grateful to all staffs in Faculty of Engineering who have helped me, especially to Bongkot Prucksapong at Department of Computer Engineering. I also would like to thank officer Chanawaatt Choosang, from Department of Biology, Faculty of Science, for his kindness direction and help during the image acquisition using microscope.

Great thanks should go to my friends who have put considerable time and effort into their comments on the draft. Especially thanks give to Rattana Saelee, Budsara Sakulsujirapa, and Hayatul Waha. They have put a lot of time on helping me and being translators during image acquisition, because I could not speak Thai language.

Finally, I am indebted to my parents, sisters, and other relatives for their continuous support and encouragement.

Fu Dongjin

CONTENTS

Chapter 1. Introduction.....	1
1.1. Background and problem statement.....	1
1.2. Objective.....	2
1.3. Scope.....	2
Chapter 2. Theory.....	3
2.1. Hard disk and hard disk head.....	3
2.2. HSA.....	4
2.3. Laser measurement system.....	5
2.4. Super resolution.....	6
2.5. Image enhancement.....	11
2.6. Edge detection.....	12
2.6.1 Roberts cross operator.....	13
2.6.2 Sobel operator.....	13
2.6.3 Prewitt operator.....	14
2.6.4 Canny operator.....	15
2.7. Contour following.....	20
2.8. Line fitting.....	22
2.8.1 Least squares method.....	22
2.8.2 Eigenvalue-eigenvector method.....	24
2.8.3 Function cvFitline.....	25
2.9. Resolution specification.....	26
Chapter 3. Experiment and analysis.....	29
3.1. Image acquisition.....	30
3.2. Image cropping.....	32
3.3. Super resolution.....	32
3.4. Image enhancement.....	33
3.5. Edge detection.....	34
3.6. Robust contour following.....	35

3.7.	Edge auto-segmentation.....	36
3.8.	Line fitting.....	37
3.9.	Comparison and analysis.....	38
3.9.1	Image format comparison.....	38
3.9.2	Image adjustment comparison.....	39
3.9.3	Line fitting method comparison.....	40
3.9.4	Super resolution affection comparison.....	41
3.10.	Results.....	42
Chapter 4. Summary.....		44
4.1.	Conclusion.....	44
4.2.	Recommendations for future work.....	45
References.....		46
Vitae.....		49

LIST OF TABLES

Table 2-1	Direction guide for contour following.....	22
Table 2-2	Region details of pitch angle and roll angle.....	28
Table 3-1	Angle results of different image format.....	39
Table 3-2	Angle results of different adjustment threshold.....	40
Table 3-3	Angle results of different line fitting method.....	40
Table 3-4	Angle results of super resolution affection.....	41
Table 3-5	Computing time of super resolution affection.....	41
Table 3-6	Pitch angle results taken by rotating step by step.....	42

LIST OF FIGURES

Figure 1-1	Hard disk internals.....	1
Figure 1-2	Pitch angle and roll angle.....	2
Figure 2-1	The structure of an HDH.....	4
Figure 2-2	Head stack assembly.....	5
Figure 2-3	Laser measurement system.....	6
Figure 2-4	Common imaging system.....	6
Figure 2-5	Basic premise for super resolution.....	7
Figure 2-6	The necessity of interpolation in HR sensor grid.....	9
Figure 2-7	Aliasing relationship between LR image and HR image.....	10
Figure 2-8	Divide a semicircle into 5 regions.....	18
Figure 2-9	Result images of contour following.....	21
Figure 2-10	Eight directions for following.....	22
Figure 2-11	Two offsets types.....	23
Figure 2-12	Compare precision and accuracy.....	26
Figure 2-13	Pitch and roll angle side images of an HDH.....	27
Figure 2-14	Computing angle resolution.....	27
Figure 3-1	Image processing system flow chart.....	29
Figure 3-2	A pitch angle side of an HDH.....	30
Figure 3-3	The board's length and rotation distance.....	31
Figure 3-4	Image acquisition environment.....	31
Figure 3-5	Cropped image.....	32
Figure 3-6	Image processing images.....	33
Figure 3-7	Image adjustment results.....	34
Figure 3-8	Results of different edge detection.....	35
Figure 3-9	Robust contour following images.....	36
Figure 3-10	Image results for edge line select automatically method.....	37
Figure 3-11	Result image of edge detection after using default adjustment.....	39

LIST OF ABBREVIATIONS

HDH	Hard Disk Head
HSA	Head Stack Assembly
SR	Super Resolution
HR	High Resolution
LR	Low Resolution
CFT	Continuous Fourier Transform
DFT	Discrete Fourier Transform
JPEG	Joint Photographic Experts Group
TIFF	Tagged Image File Format
stdev	Standard deviation

CHAPTER 1

INTRODUCTION

1.1. Background and problem statement

A hard disk's main components include the Hard Disk Head (HDH), platter, spindle, actuator, and actuator arm. Different hard disks contain different numbers of HDHs (normally six), collectively known as an HDH stack.



Figure 1-1. Hard disk internals [1].

HDH is one of the key parts in a hard disk because of its function and sensitive. Since HDH has to write electric data to platter and read magnetic data from platter, the floating height-the distance between the HDH and the platter-is between 0.127 and $0.305 \mu m$. This height must be maintained within a proper range by making both HDH and platter parallel with horizontal plane in the Head Stack Assembly (HSA). Otherwise, there will be a head crash. In the HSA, the HDH and platter are parallel to the horizontal plane to maintain this height. It is easy to make the platter parallel, but not so for the HDH because of its small size.

An HDH's pitch angle (Figure 1-2 (a)) and roll angle (Figure 1-2 (b)) are the angles between one side of the HDH and the local magnetic field. Both pitch angle and roll angle must be in the range -0.05° to 0.05° , and if either exceeds this range, then the HDH stack is rejected. Yaw angle is the angle between the HDH's heading and magnetic north. Variation in yaw angle does not affect floating height, so it will not be considered here.

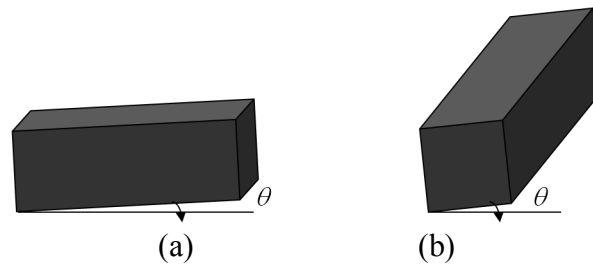


Figure 1-2. Pitch angle and roll angle.

Currently, the pitch and roll angles of an HDH are normally measured using a laser measurement system. The laser system can offer good resolution, it is expensive. This research presents an attempt to replace the laser system with an image processing system.

1.2. Objective

The objective is to measure the pitch and roll angles of the HDH using image processing techniques.

1.3. Scope

- 1) An HDH was attached to an actuator arm to simulate HSA production-line.
- 2) The HDH size we used in this report is the length of $1235 \mu m$, width $1010 \mu m$, and thickness $280 \mu m$.

CHAPTER 2

THEORY

2.1. Hard disk and hard disk head

Hard disk [1], also referred to as hard disk drive, is a storage device which stores digitally encoded data on platters. A hard disk's main components include HDH, platter, spindle, actuator, and actuator arm.

Platters are coated on both sides with a special media material designed to store information in the form of magnetic patterns. The platters are mounted by cutting a hole in the center and stacking them onto a spindle. The platters rotate at high speed, driven by a special spindle motor connected to the spindle. HDHs, which are special electromagnetic read/write devices, are mounted onto sliders and used to either record information onto the platter or read information from it. The sliders are mounted onto arms, all of which are mechanically connected into a single assembly and positioned over the surface of the disk by a device called an actuator.

Each surface of a platter in the hard disk can hold tens of billions of individual bits of data. Each platter has two HDHs, one on the top of the platter and one on the bottom. Different hard disks contain different numbers of HDHs (normally six), collectively known as an HDH stack.

Each HDH (Figure 2-1) is comprised of four layers together [2] into a single structure. These four layers are free layer, which is the sensing layer, made of a nickel-iron alloy, and is passed over the surface of the data bits to be read; spacer, which is nonmagnetic, typically made from copper; pinned layer, which is made of cobalt material; and exchange layer, which is typically made from iron and manganese to fix the pinned layer's magnetic orientation. Four layers work together to finish the function of an HDH, which is to convert electronic bits to magnetic pulses and store them on the platters, and then reverse the process when the data needs to be read back.

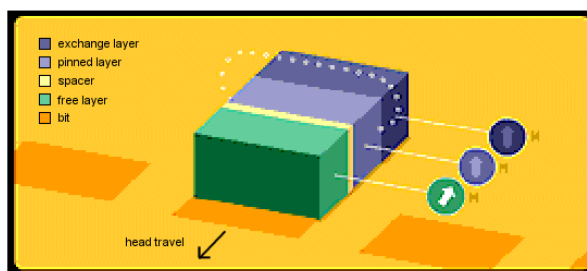


Figure 2-1. The structure of an HDH [2].

Modern HDHs float over the surface of the disk and do all of their work without ever physically touching the platters. The amount of space between the HDH and the platter is called the floating height. The floating height normally is between 0.127 and $0.305 \mu\text{m}$. The HDH assemblies are spring-loaded--using the spring steel of the head arms, which causes the sliders to press against the platters when the disk is stationary. (This is done to ensure that the HDHs don't drift away from the platters; maintaining an exact floating height is essential for correct operation.) When the disk spins up to operating speed, the high speed causes air to flow under the sliders and lift them off the surface of the disk--the same principle of lift that operates on aircraft wings and enables them to fly.

When the PC is shut down, HDHs will auto park on a designated area of the platters, where there is no data, so they will not be damaged during transportation.

2.2. HSA

Assembly considerations are vital to any consideration of the reliability of a drive, due to its very sensitive components. This includes a look at the exterior of the hard disk, a discussion of how the drive is sealed against contamination from the outside air, how drives should be oriented for maximum reliability, and how to make sure the angles of HDH.

Due to both the HDH size and floating height are very small, the HSA (Head Stack Assembly) is assembled in a clean room containing air specially filtered to remove all but the tiniest particles, and the entire hard disk must be manufactured to a high degree of precision and accuracy. The main part of the hard disk is isolated from outside air to ensure that no contaminants get onto the platters, which could cause

damage to the HDHs.

After HDH stack finishes all other prophase assembly sections, it will be set in order (Figure 2-2), waiting for platters to be inserted between each two functionary HDH sides. The floating height, the distance between the HDH and the platter, is between 0.127 and $0.305 \mu\text{m}$. In the HSA, the HDH and platter are parallel to the horizontal plane to maintain this height. It is easy to make the platter parallel, but not so for the HDH because of its small size. This is the primary problem that this thesis has to solve. Otherwise, the HDH will touch the platter causing a head crash.

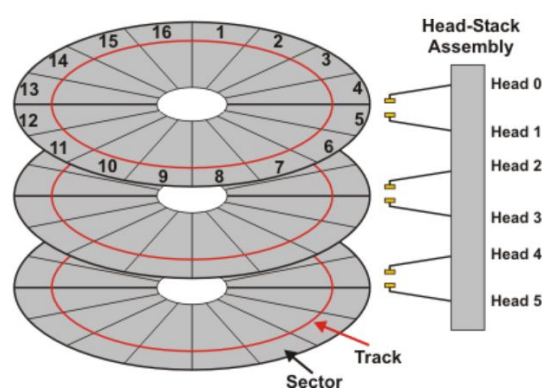


Figure 2-2. Head stack assembly [3].

On the picture above, it is a hard disk with three platters. It contains 6 HDHs, which move synchronously.

An HDH's pitch angle (Figure 1-2 (a)) and roll angle (Figure 1-2 (b)) are the angles between one side of the HDH and the local magnetic field. Both pitch angle and roll angle must be in the range -0.05° to 0.05° , and if either exceeds this range, then the HDH stack is rejected. Yaw angle is the angle between the HDH's heading and magnetic north. Variation in yaw angle does not affect floating height, so it will not be considered here.

2.3. Laser measurement system

Currently, pitch and roll angles of HDH are normally measured using a laser measurement system as shown in Figure 2-3.

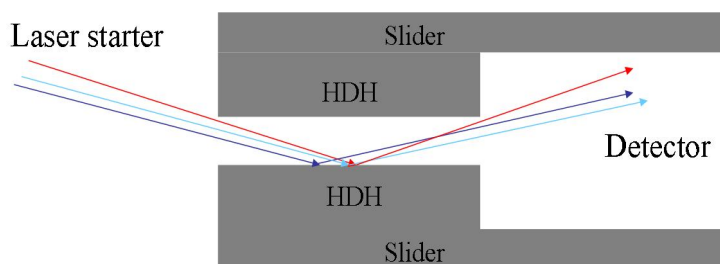


Figure 2-3. Laser measurement system.

Laser starter is a device sending out laser beams. Laser beams are collected by a detector after being reflected off the HDH's surface. Variations in an HDH's angle will affect the direction of the reflected beam, which will arrive at a different position at the detector. By analyzing the beam positions on the detector, the HDH's angle can be computed. Laser system can offer good resolution, while one system also very expensive, about 30,000 USD.

2.4. Super resolution

Super resolution [4]-[11] is a method to remove blur caused by the imaging system (aliasing, motion blur, non-ideal sampling, etc.) as well as recovery of spatial frequency information beyond the diffraction limit of the optical system (Figure 2-4), through the use of image processing algorithms. Normally, it obtains a high resolution (HR) image from many low resolution (LR) images. HR means that the image's pixel density is higher, and therefore an HR image can offer more details.

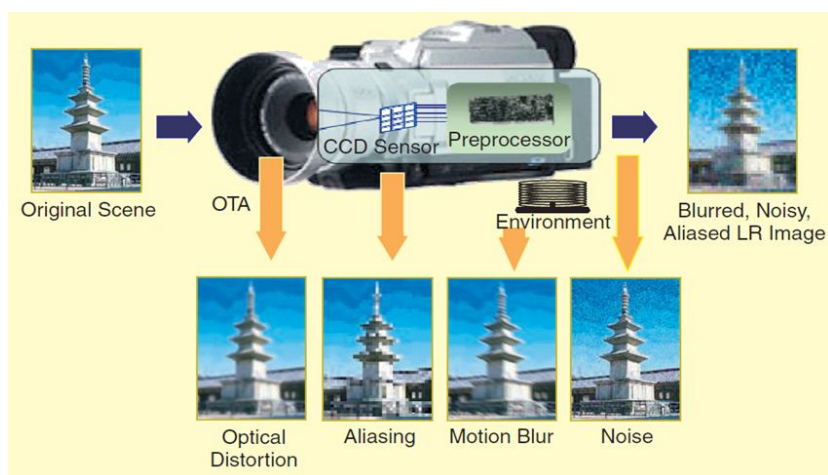


Figure 2-4. Common imaging system [4].

Typically, the LR images represent different looks at the same scene. That is, LR images are aliased as well as shifted with subpixel precision. If the LR images all are shifted by integer units, there is no new information we can use to reconstruct an HR image because that each image contains the same information. But subpixel shifts can offer us new information that can be exploited to obtain an HR image (Figure 2-5). To obtain different looks at the same scene, some relative scene motions must exist among many frames or video sequences. These motions can occur due to the images acquired from orbiting satellites. Super resolution image reconstruction will be successful if we estimate subpixel and combine LR images.

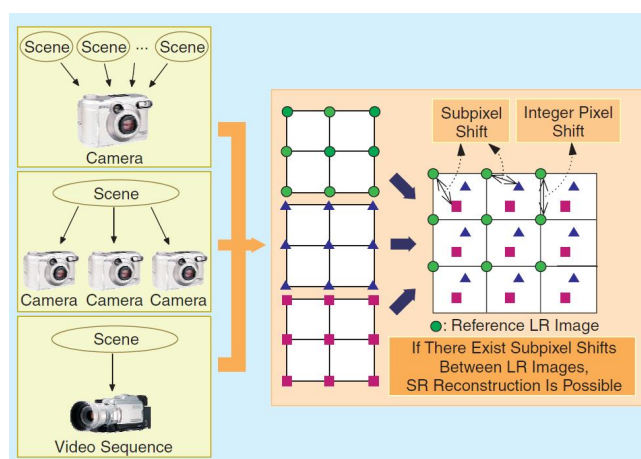


Figure 2-5. Basic premise for super resolution [4].

There are three critical factors affecting super resolution method. Firstly, reliable subpixel motion information is essential as mentioned before, and motion estimates are more important than subpixel motion information. Secondly, observation models have to accurately describe the imaging system and its degradations. Thirdly, restoration methods must provide the maximum potential of all a-priori information. These observations and the discussion of constraints on the solution space suggest the following approaches and actions:

Motion estimation: Existing subpixel motion estimation methods should be studied and extended with emphasis on providing robust performance and reliability measures on motion estimates. Regularized and simultaneous motion estimation methods should be taken into account.

Motion models: Motion models and estimation methods should be elected according to the a-priori knowledge. Model based on motion estimators should be considered as multiple independent motion. As motion is estimated from degraded observation image, the reliability of these estimates should be studied. Motion constraints should be applied to the image restoration step.

Observation models: Observation models which accurately account for degradations occurring in the imaging system constrain the image solution space. They can be broadly divided into the models for still images and for video sequence. To present a basic concept of super resolution, we employ the observation model for still images in this report, and it is rather straightforward and easy to extend the still image model to the video sequence model.

Restoration Algorithms: Restoration algorithms should be able to include reliability measures on individual motion estimates as well as accommodate model based motion estimators.

The desired HR image size $L_1N_1 \times L_2N_2$ is considered as the vector $x = [x_1, x_2, \dots, x_N]^T$, where $N = L_1N_1 \times L_2N_2$. That is to say, x is the ideal undegraded image that is sampled at or above the Nyquist rate from a continuous scene. Make the parameters L_1 and L_2 represent the down-sampling factors of the horizontal and vertical directions in the observation model, respectively. Thus, each observed LR image's size is $N_1 \times N_2$. Let the k th LR image be denoted in the notation as $y_k = [y_{k,1}, y_{k,2}, \dots, y_{k,M}]^T$, for $k = 1, 2, \dots, p$ and $M = N_1 \times N_2$. Now, it is assumed that x remains constant during the acquisition of the multiple LR images, except for any motion and degradation allowed by the model. Therefore, the observed LR images result from warping, blurring, and subsampling operators performed on the HR image x . Assuming that each LR image is corrupted by additive noise, the observation model can be represented as [12], [13]

$$y_k = DB_k M_k x + n_k \quad \text{for } 1 \leq k \leq p \quad (1)$$

where M_k is a warp matrix of size $L_1N_1L_2N_2 \times L_1N_1L_2N_2$, B_k represents a

$L_1 N_1 L_2 N_2 \times L_1 N_1 L_2 N_2$ blur matrix, D stand for a $(N_1 N_2)^2 \times L_1 N_1 L_2 N_2$ subsampling matrix, and n_k is a lexicographically ordered noise vector.

The warping process executed on HR image is actually defined in terms of LR pixel spacing when we estimate it. Therefore, this step requires interpolation if the tiny motion is not the same with the HR grid. An example for the whole translation is shown in Figure 2-6. We set a circle (O) to represent for the original (reference) HR image x , and a triangle (Δ) and a diamond (\diamond) are globally shifted motions of x . If the down-sampling factor is two, a diamond (\diamond) has (0.5, 0.5) subpixel shift for the horizontal and vertical directions, while a triangle (Δ) has a less than (0.5,0.5) shift. Therefore a diamond (\diamond) does not need interpolation from x , but a triangle (Δ) needs since it is not located on the HR grid.

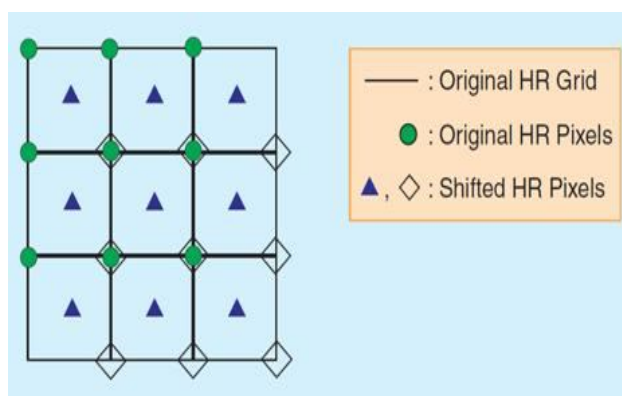


Figure 2-6. The necessity of interpolation in HR sensor grid [4].

Super resolution image reconstruction methods mostly consist of the three stages: registration, interpolation, and restoration. These steps can be implemented separately or simultaneously according to needs. The registration is extensively studied in various fields of image processing [14]-[17]. In the registration stage, the relative shifts between LR images compared to the reference LR image are estimated with fractional pixel accuracy. Obviously, accurate subpixel motion estimation is a very important factor to implement the super resolution successfully. Since the shifts among LR images are arbitrary, the registered HR image will not always be matched with the uniformly spaced HR grid. Thus, interpolation is a necessary stage to obtain an HR image from some LR images. Finally, restoration is employed to remove

blurring and noise.

We describe a major class of super resolution methods which utilize a frequency domain formulation of the super resolution problem. The techniques discussed utilize the shifting property of the Fourier transform to model global translational scene motion, and take advantage of the sampling theory to enable effect restoration made possible by the availability of multiple observation images. The frequency domain approach of super resolution is based on three theories: i) the shifting property of the Fourier transform, ii) the aliasing relationship between the discrete Fourier transform (DFT) of observed LR images and the continuous Fourier transform (CFT) of the original HR image, iii) and the assumption that the original HR image is bandlimited. These properties make it possible to transact the system equation relating the aliased DFT coefficients of the observed LR images to a sample of the CFT of an unknown image. For example, let us assume that there are two 1-D LR signals sampled below the Nyquist sampling rate. Following the above three principles, the unaliased HR signal can be decomposed from the aliased LR signals as shown in Figure 2-7.

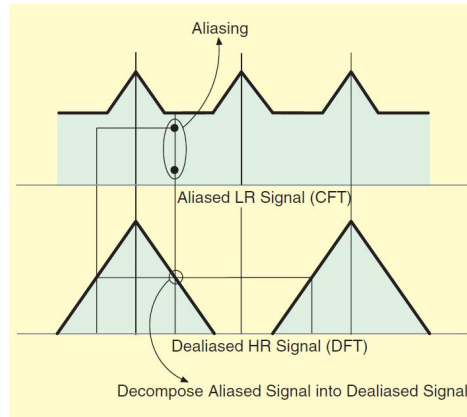


Figure 2-7. Aliasing relationship between LR image and HR image [4].

A continuous HR image is denoted to be $x(t_1, t_2)$, and its CFT is $X(w_1, w_2)$. The global translations yield the k th shifted image of $x_k(t_1, t_2) = x(t_1 + \delta_{k1}, t_2 + \delta_{k2})$, where δ_{k1} and δ_{k2} are arbitrary but known values, and $k = 1, 2, \dots, p$. The CFT of the shifted image, $X_k(w_1, w_2)$, can be defined as

$$X_k(w_1, w_2) = \exp[j2\pi(\delta_{k1}w_1 + \delta_{k2}w_2)]X(w_1, w_2). \quad (2)$$

The shifted image $x_k(t_1, t_2)$ can be sampled to generate the observed LR image $y_k[n_1, n_2]$ from the sampling period T_1 and T_2 . The relationship between the CFT of the HR image and the DFT of the k th LR image can be written as in (3), coming from the aliasing relationship and the assumed bandlimitedness of $X(w_1, w_2)$ ($|X(w_1, w_2)| = 0$ for $|w_1| \geq (L_1\pi/T_1)$, $|w_2| \geq (L_2\pi/T_2)$) [18].

$$\gamma_k[\Omega_1, \Omega_2] = \frac{1}{T_1 T_2} \sum_{n_1=0}^{L_1-1} \sum_{n_2=0}^{L_2-1} X_k \times \left(\frac{2\pi}{T_1} \left(\frac{\Omega_1}{N_1} + n_1 \right), \frac{2\pi}{T_2} \left(\frac{\Omega_2}{N_2} + n_2 \right) \right). \quad (3)$$

Lexicographic ordering is used for the indices n_1 and n_2 on the right-hand side, and k on the left-hand side, to obtain a matrix vector form:

$$Y = \Phi X, \quad (4)$$

where Y is a $p \times 1$ column vector with the k th element of the DFT coefficients of $y_k[n_1, n_2]$, X is a $L_1 L_2 \times 1$ column vector with the samples of the unknown CFT of $x(t_1, t_2)$, and Φ is a $p \times L_1 L_2$ matrix which relates the DFT of the observed LR images to samples of the continuous HR image.

One advantage of the frequency domain approach is its simple theory, because that the relationship between LR images and the HR image is clearly demonstrated. Besides, the frequency method also doesn't ask for complex hardware specification. Both two advantages are important for production-line using.

2.5. Image enhancement

Image enhancement is a big research area in image processing. We employ both median filter and image adjustment to enhance image.

The median filter [19 ~21] is a non-linear digital filtering technique, often used to remove noise from images or other signals. It is particularly useful to reduce speckle noise and salt and pepper noise, and its edge-preserving nature makes it useful in cases where edge blurring is undesirable.

The idea of median filter is to calculate the median of neighbouring pixels' values. This can be done by repeating these steps for each pixel in the image: store the neighbouring pixels in an array (the array should be odd sized); sort the array in numerical order; pick the median from the array as the new pixels value.

Image adjustment [22] adjusts image intensity values to increase the contrast of image. Image adjustment maps the values in intensity input image to new values in output image. We set a threshold (low_in and $high_in$) for input image and another one (low_out and $high_out$) for output image. Then this function will adjust values between low_in and $high_in$ map to values between low_out and $high_out$, while values below low_in and above $high_in$ are clipped. Sometimes an empty matrix is used for either or both input and output thresholds to specify the default of [0 1].

2.6. Edge detection

Edge detection [1] is an important technology in image processing, particularly in the areas of feature detection and feature extraction, to refer to algorithms which aim at identifying points in a digital image at which the image brightness changes sharply or more formally has discontinuities.

The purpose of detecting sharp changes in image brightness is to capture important events and changes in properties. It can be shown that under rather general assumptions for an image formation model, discontinuities in image brightness are likely to correspond to discontinuities in depth, discontinuities in surface orientation, changes in material properties and variations in scene illumination.

Edge detection, which contains many different methods, can be grouped into two categories, search-based and zero-crossing based. The search-based methods detect edges by first computing a measure of edge strength, usually a first-order derivative expression such as the gradient magnitude, and then searching for local directional maxima of the gradient magnitude using a computed estimate of the local orientation of the edge, usually the gradient direction. The zero-crossing based methods search for zero crossings in a second-order derivative expression computed from the image to find edges.

Some edge-detection operators, which are based on second-order derivatives of the intensity, essentially capture the rate of change in the intensity gradient. Thus,

in the ideal continuous case, detection of zero-crossings in the second derivative captures local maxima in the gradient. The early Marr-Hildreth operator is based on the detection of zero-crossings of the Laplace operator applied to a Gaussian-smoothed image. It can be shown, however, that this operator will return false edges corresponding to local minima of the gradient magnitude. Moreover, this operator will give poor localization at curved edges. Hence, this operator is nowadays mainly of historical interest.

The search-based category includes Roberts cross, Sobel, Prewitt and Canny operators. We describe them one by one.

2.6.1 Roberts cross operator

The Roberts cross operator is one of the earliest edge detection methods, which works by calculating the sum of the squares of the differences between diagonally adjacent pixels. This can be accomplished by convolving the image with these two 2*2 kernels:

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \text{ and } \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

Convolving each of the Roberts cross operators can be shown to be the image derivative along the non-zero diagonal, and the combination of those convolved images can also be thought of as the gradient of the two upper pixels to the two lower pixels. Roberts cross is still in use because of the fast speed of computation, but performance compared to other operators is poor, with noise sensitivity a significant problem.

2.6.2 Sobel operator

The Sobel operator is a discrete differentiation operator, computing an approximation of the gradient of the image intensity function, and giving the direction of the largest possible increase from light to dark and the rate of change in that direction. At each point in the image, the result of the Sobel operator is either the corresponding gradient vector or the norm of this vector. The Sobel operator is based

on convolving the image with a small, separable, and integer valued filter in both horizontal and vertical direction and is therefore relatively inexpensive in terms of computing time. On the other hand, the gradient approximation which it produces is relatively crude, in particular for high frequency variations in the image. The Sobel operator at an image point which is in a region of constant image intensity is a zero vector and at a point on an edge is a vector which points across the edge, from darker to brighter values.

Mathematically, the operator uses two 3*3 kernels to convolve with the source image to calculate approximations of the derivatives - one for horizontal changes, and one for vertical. If we define A as the source image, and G_x and G_y are two images which at each point contain the horizontal and vertical derivative approximations, the computation functions are as follows:

$$G_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * A \quad (6)$$

$$G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A \quad (7)$$

The x-coordinate is here defined as increasing in the "right"-direction, and the y-coordinate as the "down"-direction. At each point in the image, the resulting gradient approximations can be combined to give the gradient magnitude, using:

$$G = \sqrt{G_x^2 + G_y^2} \quad (8)$$

Using this information, we can also calculate the gradient's direction:

$$\theta = \arctan\left(\frac{G_y}{G_x}\right) \quad (9)$$

where, for example, θ is 0 for a vertical edge which is darker on the left side.

2.6.3 Prewitt operator

The Prewitt operator is a method to compute the maximum response of a set of convolution kernels finding the local edge orientation of each pixel. Various kernels can be used in this operation. The whole set of 8 kernels is produced by choosing one of the kernels and rotating its coefficients circularly. Each of kernels is sensitive to one edge orientation ranging from 0° to 315° in steps of 45° , where 0° corresponds to the vertical edge. The maximum response for each pixel is taken as the value of the corresponding pixel in the output magnitude image.

The Prewitt edge detector is an appropriate way to estimate the magnitude and orientation of an edge. Although differential gradient edge detection needs a rather time-consuming calculation to estimate the orientation from the magnitudes in the x- and y-directions, the Prewitt edge detection obtains the orientation directly from the kernel with the maximum response. The set of kernels is limited to 8 possible orientations; however experience shows that most direct orientation estimates are not much more accurate. On the other hand, the set of kernels needs 8 convolutions for each pixel, whereas the set of kernel in gradient method needs only 2, one kernel being sensitive to edges in the vertical direction and one to the horizontal direction. The result for the edge magnitude image is very similar with both methods, provided the same convolving kernel is used.

Mathematically, the operator uses two 3×3 kernels to convolve with the source image to calculate approximations of the derivatives. If we define A as the source image, and G_x and G_y are two images which at each point contain the horizontal and vertical derivative approximations, the computation functions are as follows:

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix} * A \quad (10)$$

$$G_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ +1 & +1 & +1 \end{bmatrix} * A \quad (11)$$

2.6.4 Canny operator

John Canny [23], who invented the Canny operator, considered the mathematical problem of deriving an optimal smoothing filter given the criteria of detection, localization and minimizing multiple responses to a single edge. He showed that the optimal filter given these assumptions is a sum of four exponential terms. He also showed that this filter could be well approximated by first-order derivatives of Gaussians. Canny also introduced the notion of non-maximum suppression, which means that given the presmoothing filters, edge points are defined as points where the gradient magnitude assumes a local maximum in the gradient direction. Although his work was done in the early days of computer vision, the Canny edge detector is still a state-of-the-art edge detector. Unless the preconditions are particularly suitable, it is hard to find an edge detector that performs significantly better than the Canny edge detector.

John Canny's intentions were to enhance the many edge detectors already out at the time he started his work. He followed a list of criteria to improve current methods of edge detection. The first and most obvious is low error rate. It is important that edges occurring in images should be detected and that there be NO responses to non-edges. The second criterion is that the edge points be well localized. That's to say, the distance between the edge pixels are found by the Canny detector and the actual edge is to be at a minimum. The third one is to have only one response to each single edge. This was obtained because the first 2 criterions were not substantial enough to completely eliminate the possibility of multiple responses to an edge.

Based on these criterias, the canny edge detector first smoothes the image to eliminate and noise. It then finds the image gradient to highlight regions with high spatial derivatives. The algorithm then tracks along these regions and suppresses any pixel that is not at the maximum (nonmaximum suppression). The gradient array is now further reduced by hysteresis. Hysteresis is used to track along the remaining pixels that have not been suppressed, using two thresholds.

In order to implement the Canny edge detector algorithm, a series of steps must be followed. The first step is to filter out noise in the source image before trying to locate and detect any edges. And because the Gaussian filter can be computed using a simple mask, it is used exclusively in the Canny algorithm. Once a suitable

mask has been chosen, the Gaussian smoothing can be performed using standard convolution methods. A convolution mask must be much smaller than the source image. As a result, the mask is slid over the image, manipulating a square of pixels at a time. The larger the width of the Gaussian mask, the lower is the detector's sensitivity to noise. The localization error in the detected edges also increases slightly as the Gaussian width is increased. The Gaussian mask normally been used is shown below.

$$\frac{1}{115} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

After smoothing the image and eliminating the noise, the next step is to find the edge strength by calculating the gradient of the image. The Sobel operator performs a 2-D spatial gradient measurement on an image. Then, the approximate absolute gradient magnitude (edge strength) at each point can be found. The Sobel operator uses a pair of 3*3 convolution masks, one estimating the gradient in the x-direction (columns) and the other estimating the gradient in the y-direction (rows). They are shown below:

$$\begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \quad \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

G_x G_y

The magnitude of the gradient is then approximated using the formula:

$$|G| = |G_x| + |G_y| \quad (12)$$

Finding the edge direction is trivial once the gradient in the x and y directions are known. However, we will generate an error whenever sum X is equal to zero. So in the code there has to be a restriction set whenever this takes place. Whenever the gradient in the x direction is equal to zero, the edge direction has to be equal to 90 degrees or 0 degrees, depending on what the value of the gradient in the y-direction is equal to. If G_y has a value of zero, the edge direction will equal 0 degrees. Otherwise the edge direction will equal 90 degrees. The formula for finding the edge direction is:

$$\theta = \arctan(G_y/G_x) \quad (13)$$

Once the edge direction is known, the next step is to relate the edge direction to a direction that can be traced in an image. So if the pixels of a 5*5 image are aligned as follows:

```

x  x  x  x  x
x  x  x  x  x
x  x  a  x  x
x  x  x  x  x
x  x  x  x  x

```

By looking at the pixel "a", there are only four possible directions to describe the surrounding pixels: 0 degrees (in the horizontal direction), 45 degrees (along the positive diagonal), 90 degrees (in the vertical direction), or 135 degrees (along the negative diagonal). Therefore, the edge orientation has to be resolved into one of these four directions depending on which direction it is closest to (e.g. if the orientation angle is found to be 3 degrees, make it zero degrees). Think of this as taking a semicircle and dividing it into 5 regions (4 different angles).

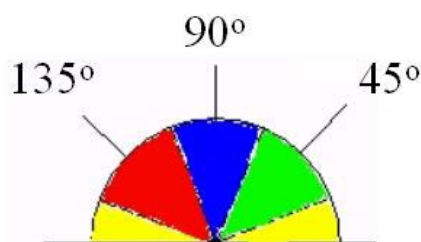


Figure 2-8. Divide a semicircle into 5 regions [24].

Therefore, any edge direction falling within the yellow range (0 to 22.5 and 157.5 to 180 degrees) is set to 0 degrees; any in the green range (22.5 to 67.5 degrees) is set to 45 degrees; any in the blue range (67.5 to 112.5 degrees) is set to 90 degrees; and any in the red range (112.5 to 157.5 degrees) is set to 135 degrees.

Once we have computed a measure of edge strength (typically the gradient magnitude), the next step is to apply a threshold, to decide whether edges are present or not at an image point. The lower the threshold, the more edges will be detected, and the result will be increasingly susceptible to noise, and also to pick out irrelevant features from the image. Conversely a higher threshold may miss subtle edges, or result in fragmented edges.

If the edge thresholding is applied to just the gradient magnitude image, the resulting edges will in general be thick and some type of edge thinning post-processing is necessary. However, for edges detected with non-maximum suppression, the edge curves are thin by definition and the edge pixels can be linked into edge polygon by an edge linking procedure. On a discrete grid, the non-maximum suppression step can be implemented by estimating the gradient direction using first-order derivatives, then rounding off the gradient direction to multiples of 45 degrees. Finally, compare the values of the gradient magnitude in the estimated gradient direction.

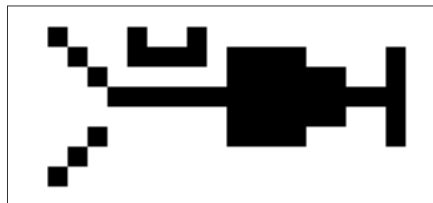
Commonly a used approach to handle the problem of appropriate thresholds for thresholding is by using thresholding with hysteresis, which uses multiple thresholds to find edges. We begin by using the upper threshold to find the start of an edge. Once we have a start point, we then trace the path of the edge through the image

pixel by pixel, marking an edge whenever we are above the lower threshold. We stop marking our edge only when the value falls below our lower threshold. This approach makes the assumption that edges are likely to be in continuous curves, and allows us to follow a faint section of an edge we have previously seen, without meaning that every noisy pixel in the image is marked down as an edge. However, choosing appropriate thresholding parameters is still a problem to solve, and suitable thresholding values may vary over the image.

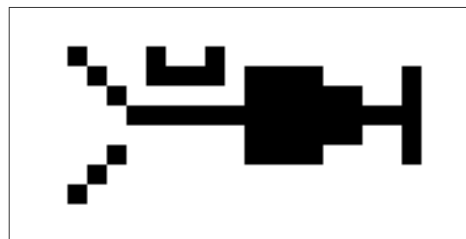
2.7. Contour following

Contour following takes a binary array and returns the sorted row and column coordinates of contour pixels, and ends when the first two and last two point coordinates are the same. Besides, we send back blank image result if the object is less than 3 pixels, because we consider that less than 3 pixels cannot make a contour.

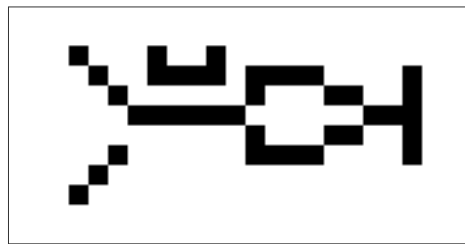
Contour following works as follows. Firstly, we add one blank pixel to each side of the image (Figure 2-9 (b)). Otherwise, the next point maybe empty while we are searching contour. Secondly, we erode the image (Figure 2-9 (c)). Thirdly, we thin the image so that an object without holes shrinks to a minimally connected stroke, and an object with holes shrinks to a ring halfway between the hold and outer boundary (Figure 2-9 (d)).



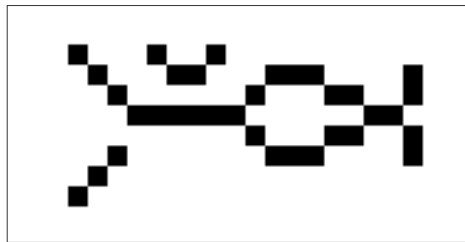
(a)



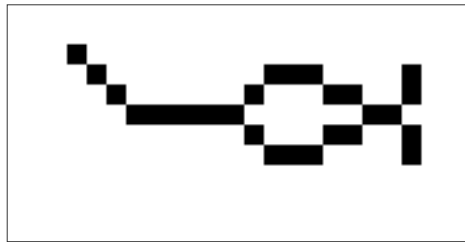
(b)



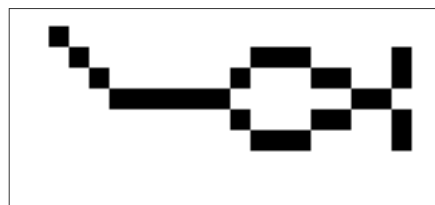
(c)



(d)



(e)



(f)

Figure 2-9. Result images of contour following.

(a) Original image; (b) Add one blank pixel to each side; (c) Eroded image; (d) Thinned image; (e) Contour following image; (f) Final contour following image on the original size image.

Fourthly, after all preparations, we start to follow points from the first top-left point, or any other point you set. There are eight directions to follow for the next pixel (Figure 2-10), and we choose direction 0 as the starting direction. If there is no pixel on direction 0, we will switch to direction 1, 2... 7 till getting a pixel. Otherwise, there is only one alone point, less than 3 points. We will return an empty array to show that we can not obtain a contour. In figure 2-9 (d), we get a point in direction1. We

continue to follow starting from this point, and start anew in direction 6 because our current direction is direction 1 (Table 2-1). The rule to give next direction is: take direction 0 and 1 for example, the opposite direction is direction 4 and 5, while we set the next direction to be direction 6. This means we will start to follow from direction 6, and then direction 7, and the final directions are direction 0 and 1. Before following back, we have tried all other directions, without missing any part of the contour. The reason we do not have to set next direction to be direction 5 for current direction 0 (direction 4 is the opposite direction of direction 0) is because we have followed that pixel already before following current pixel. Therefore, we take direction 0 and 1 to be the same group, no need to be separated. Continuous work by following above rules till the first two and last two point coordinates are the same, and this means that we have followed all contour and avoiding endless loop. The result image is shown in figure 2-9 (e).

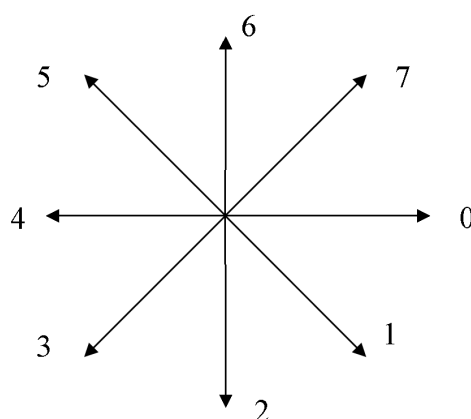


Figure 2-10. Eight directions for following.

Table 2-1. Direction guide for contour following.

Current direction	Next direction
0, 1	6
2, 3	0
4, 5	2
6, 7	4

Fifthly, we delete four blank pixels on all four sides. A clear contour after contour following is shown in Figure 2-9 (f).

2.8. Line fitting

Line fitting, in short, it is one way to get a straight line from a curve line using kinds of algorithm. We describe three methods of line fitting: least squares method, eigenvalue-eigenvector method, and function `cvFitline`.

2.8.1 Least squares method

The least squares method is applied to approximate solutions of overdetermined systems, i.e. systems of equations in which there are more equations than unknowns. The best fit, between modeled data and observed data, in its least-squares sense, is an instance of the model for which the sum of squared residuals has its least value, where a residual is the difference between an observed value and the value provided by the model. Least squares corresponds to the maximum likelihood criterion if the experimental errors have a normal distribution and can also be derived as a method of moments estimator.

Least squares method is a mathematical procedure for finding the best-fitting curve to a given set of points by minimizing the sum of the squares of the offsets of the points from the curve. The sum of the squares of the offsets is used instead of the offset absolute values because this allows the residuals to be treated as a continuous differentiable quantity.

Figure 2-11 shows two kinds of offset. In practice, the vertical offsets from a line (polynomial, surface, hyperplane, etc.) are almost always minimized instead of the perpendicular offsets. This provides a fitting function for the independent variable x that estimates y for a given x (most often what an experimenter wants), allows uncertainties of the data points along the x - and y -axes to be incorporated simply, and also provides a much simpler analytic form for the fitting parameters than would be obtained using a fit based on perpendicular offsets. In addition, the fitting technique can be easily generalized from a best-fit line to a best-fit polynomial when sums of vertical distances are used.

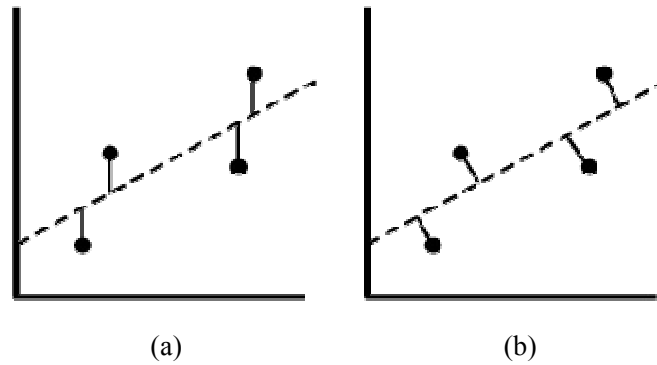


Figure 2-11. Two offsets types.

(a) Vertical offsets; (b) Perpendicular offsets.

Least squares method can fit curve to both line and curve, and we only describe fitting line type. The linear least squares fitting technique, which is the simplest and most commonly applied form, provides a solution to find the best fitting straight line from a set of points.

For a general linear equation, $y = mx + b$, it is assumed that the errors in the y -values are substantially greater than the errors in the x -values. The deviation D can be calculated by following function:

$$D = \begin{vmatrix} \sum (x_i^2) & \sum x_i \\ \sum x_i & i \end{vmatrix} \quad (14)$$

the values for m and b can be determined using deviation D :

$$m = \begin{vmatrix} \sum (x_i * y_i) & \sum x_i \\ \sum y_i & i \end{vmatrix} / D \quad (15)$$

$$b = \begin{vmatrix} \sum (x_i^2) & \sum (x_i * y_i) \\ \sum x_i & \sum y_i \end{vmatrix} / D \quad (16)$$

2.8.2 Eigenvalue-eigenvector method

Covariance is a measure of how much two variables change together, and can be defined as:

$$\text{Cov}(X, Y) = E((X - \mu)(Y - \nu)) \quad (17)$$

Where E is the expected value operator, μ is the expected value of X , and ν is the expected value of Y .

If two variables tend to vary together, then the covariance between the two variables will be positive. On the other hand, if one of them tends to be above its expected value when the other variable is below its expected value, then the covariance between them will be negative [1].

Eigenvalues and eigenvectors for the covariance result are computed. The proper eigenvector corresponds to the biggest eigenvalue. The angle of this line, which is also the HDH pitch angle, is the arctangent of this eigenvector.

2.8.3 Function cvFitline

Open CV, which is an open source library started by Intel company, offers many type algorithms of cvFitline [26]. Here we choose Fair type to fit our edge line to obtain the angle between the edge and horizontal line.

This function has three stages to fit line.

1) Calculate average value of X-coordinates (\bar{x}), Y-coordinates (\bar{y}), square of X-coordinates ($\bar{x^2}$), square of Y-coordinates ($\bar{y^2}$), and X-coordinates multiple Y-coordinates (\bar{xy}). Then we compute dx^2 , dy^2 , and dxy using below equations.

$$dx^2 = \bar{x^2} - \bar{x} * \bar{x} \quad (18)$$

$$dy^2 = \bar{y^2} - \bar{y} * \bar{y} \quad (19)$$

$$dxy = \bar{xy} - \bar{x} * \bar{y} \quad (20)$$

And we get t value by below equation:

$$t = \text{atan2}(2 * dxy, dx^2 - dy^2) / 2 \quad (21)$$

Then, we can get two points' coordinates using value t , x , and y . $x_1=x$, $y_1=y$; $x_2=\sin(t)$, $y_2=\cos(t)$. From both two points, we can describe one line.

2) We compute distance sum between the line and all points. If this sum is bigger than the value we set, for Fair type, we use below equation to compute weight.

$$\rho(r) = C^2 \times [r/C - \log(1+r/C)] \quad (C=1.3998) \quad (22)$$

3) Finally, we use this weight to implement stage 1 again, and obtain a more exact line to points.

2.9. Resolution specification

Whether a method can work or not depends on the resolution it can offer. Resolution is the ability to resolve differences, and it is the smallest possible change of the parameter that can be detected. High resolution means being able to resolve small differences. The acceptable region for the pitch and roll angles of an HDH is between -0.05° and 0.05° . To measure such a small angle, good resolution requires both very good precision and accuracy. The precision of a measurement system is the ability to which repeated measurements under unchanged conditions show the same results. The accuracy of a system is a deviation between the result and the true value. In other words the accuracy can be described as a maximum operating error that can be expected under worst case conditions [1]. In short, good precision needs the same result every time, while good accuracy needs good result, or the result you expect. For explaining better, we take shooting for example. After shooting four times each to two different board, we get the results (Figure 2-12). Compare with two results, the left one obtains better precision because three shooting positions are very near, and the right one obtains better accuracy because it gets higher score.

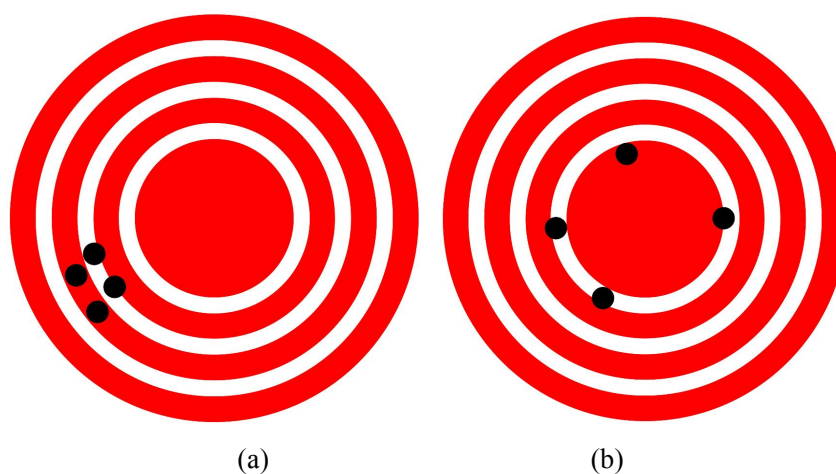


Figure 2-12. Compare precision and accuracy [1].

(a). High precision while low accuracy; (b). High accuracy while low precision.

An HDH's angle can be measured in an image, but we don't know whether it is the correct value or not. To solve this problem, we rotate the HDH by a certain angle. The accuracy can be checked by comparing the angle difference between the two measurements with the angle that the HDH was rotated while taking images.

Different image resolutions and the number of pixels in an HDH's edge lead to different resolutions. According to our experiment situation, the HDH's length is 2538 pixels and width 1992 pixels (Figure 2-13), calculated by using corner coordinates.

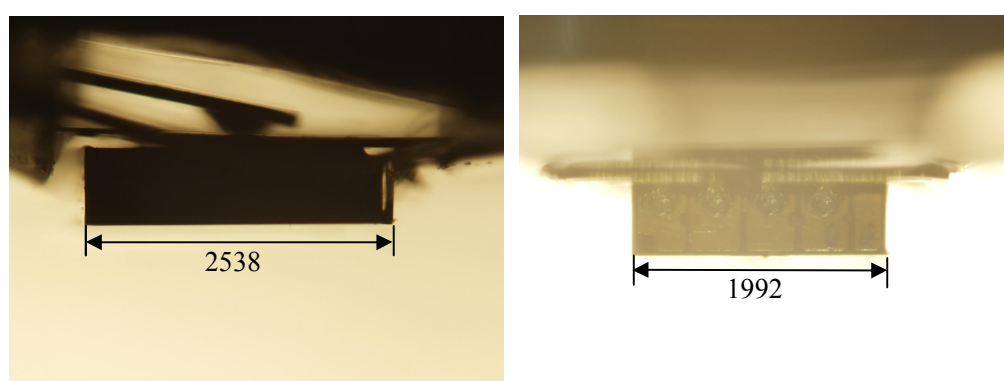


Figure 2-13. Pitch and roll angle side images of an HDH.

We take L2 as the edge of the HDH. If we set L1 to be one pixel (Figure 2-14), a pitch angle resolution (θ_1) of 2538 pixels and a roll angle resolution (θ_2) of 1992 pixels can be obtained respectively by following functions:

$$\theta_1 = \arctan \frac{L1}{L2} = \arctan \frac{1}{2538} = 0.0226^\circ \quad (23)$$

$$\theta_2 = \arctan \frac{L1}{L2} = \arctan \frac{1}{1992} = 0.0288^\circ \quad (24)$$

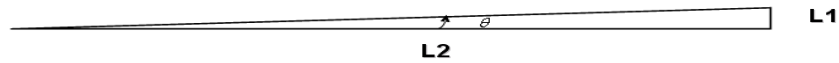


Figure 2-14. Computing angle resolution

Since acceptable angle is between -0.05° and 0.05° , the rejection, uncertain and acceptable regions of the pitch and roll angles are shown in Table 2-2. All angle smaller than -0.0791° and bigger than 0.0791° also will be rejected for pitch angle; all angle smaller than -0.1008° and bigger than 0.1008° also will be rejected for roll angle.

Table 2-2. Region details of pitch angle and roll angle.

Region	Pitch angle	Roll angle
Rejection	$-0.0791^\circ \sim -0.0565^\circ$	$-0.1008^\circ \sim -0.0720^\circ$
Uncertain	$-0.0565^\circ \sim -0.0339^\circ$	$-0.0720^\circ \sim -0.0432^\circ$
Acceptable	$-0.0339^\circ \sim -0.0113^\circ$	$-0.0432^\circ \sim -0.0144^\circ$
	$-0.0113^\circ \sim 0.0113^\circ$	$-0.0144^\circ \sim 0.0144^\circ$
	$0.0113^\circ \sim 0.0339^\circ$	$0.0144^\circ \sim 0.0432^\circ$
Uncertain	$0.0339^\circ \sim 0.0565^\circ$	$0.0432^\circ \sim 0.0720^\circ$
Rejection	$0.0565^\circ \sim 0.0791^\circ$	$0.0720^\circ \sim 0.1008^\circ$

With the same image acquisition device, the resolution of the pitch and roll angles vary only because of different length and width values. Therefore, we only implement the pitch angle on the following experiment.

CHAPTER 3

EXPERIMENT AND ANALYSIS

Image processing [1] is any form of signal processing for which the input is an image, such as photographs or frames of video; the output of image processing can be either an image or a set of characteristics or parameters related to the image. The function of image processing includes improving the image's quality, obtaining useful information from the image. Image processing contains many different methods to achieve its aim: basic geometry transformations, color corrections, digital compositing, high dynamic range imaging by combining multiple images, super resolution, image recognition and image segmentation.

The image processing measurement system (Figure 3-1) starts from image acquisition with high-quality camera and microscope, and the subsequent processings include image cropping to crop useless area, super resolution to obtain higher resolution image, image enhancement to enhance image, edge detection to detect edge, robust contour following to find the main edge, edge auto-segmentation to segment edge automatically, and line fitting to compute angle.

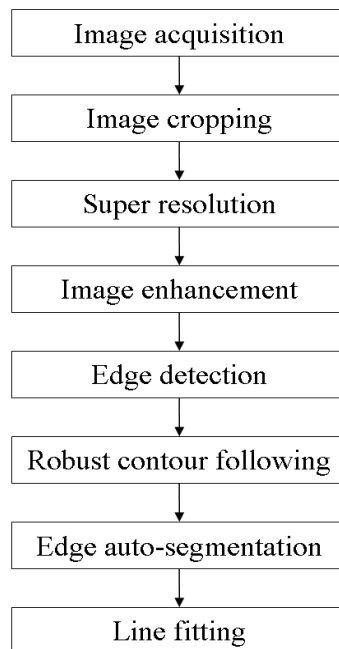


Figure 3-1. Image processing system flow chart.

After computing out resolution specification in section 2.9, we start to implement our image processing system to check whether it can work. Image acquisition is achieved with microscope and high-quality camera. The subsequent processing includes image cropping, super resolution, image enhancement, edge detection, robust contour following, edge auto-segmentation, and line fitting. Finally, we analyse results.

3.1. Image acquisition

We applied an HDH which was attached to an actuator arm to model a production-line HSA environment. Since the HDH size is very small, we captured images using a high-quality digital camera (Olympus DP71) attached to a microscope (Olympus SZX12). We chose the maximum zoom level (90 times), and rotated the camera manually to make the HDH parallel with horizontal line the best we could acting like in a real production-line situation (HDH's acceptable angle is -0.05° to 0.05° in production-line). Proper focus is important, to ensure the image does not have fuzzy areas, especially at the edges. For lights source, we don't admire strong around lights but only strong lights from HDH bottom because that some HDH level's materials will refract strong around lights causing too much useless lines after edge detection. At the same time, different lights intensity also will affect angle results, so we require the same lights intensity or vary in a tiny range. Images with the resolution of 4080×3072 , which is the maximum image resolution that the camera can offer, were acquired (Figure 3-2).

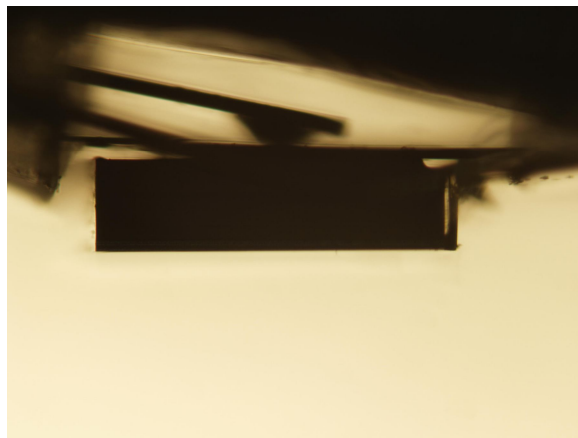


Figure 3-2. A pitch angle side of an HDH.

For computing precision, we take 20 images of our HDH at the same position. After image processing, we can get the HDH angle, but do not know whether this value is correct. Therefore, we turn the HDH 0.0226° (resolution value). Then we check that the angle difference between the starting and rotation positions is in the range of $0.0226^\circ \pm 0.0113^\circ$. There are two main ways to measure angle. The first is with a clinometer directly, and the other one measures rotation distance using a vernier. Considering measuring accuracy, we choose the second method.

Because the angle rotated in each step is very small (0.0226°), we attach the HDH to an about one meter long board, and obtain the rotation angle by measuring the rotation angle of the board (rotate from point A to point B rounding point O (Figure 3-3)). Rotation distance, a , can be computed as follows:

$$a_1 = b * \sin(0.0113^\circ) = 1014 * 0.000197 = 0.20mm \quad (25)$$

$$a = 2 * a_1 = 2 * 0.20mm = 0.40mm \quad (26)$$

For an accurate rotation distance of 0.40mm, we set the length of the board, b , to be 1014mm (Figure 3-3). The rotation distance measured by the vernier with a resolution of 0.02mm, which meets our needs.

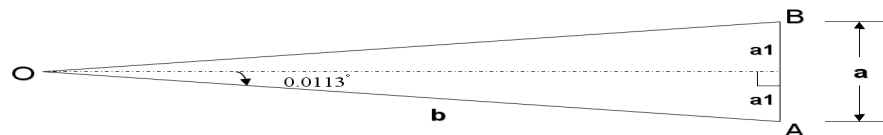


Figure 3-3. The board's length and rotation distance.

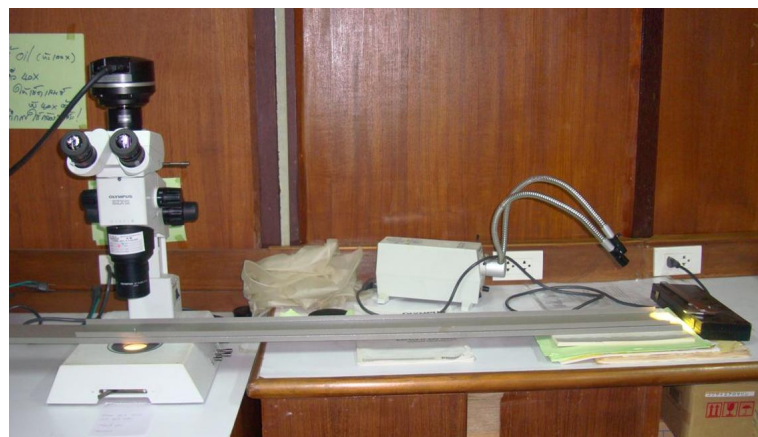


Figure 3-4. Image acquisition environment.

Three group images are taken: one at the starting position and others after rotating the HDH in steps of 0.40mm. Each group consists of twenty images.

After capturing images of an HDH, we start to implement a series of image processing on them, and compute angles from these images. In this thesis, we employ several methods: image cropping, super resolution, image enhancement, edge detection, robust contour following, edge line segmentation, and line fitting.

3.2. Image cropping

Image cropping is the process to crop image starting from the point we set, and to get a rectangle allowed by the length and width we choose, to abandon other areas.

To save processing time, we crop useless areas of the image, and choose interested area of length 2800 and width 150 pixels (Figure 3-5). Since the acceptable angle range is -0.05° to 0.05° , the HDH's position changes very little. Therefore, this rough cropping is available for every images.



Figure 3-5. Cropped image.

3.3. Super resolution

Digital cameras have a limited spatial resolution, dictated by their utilized optical lens and CCD array. Surpassing this limit can be achieved by acquiring and fusing several LR images of the same scene, producing HR images. Super resolution reconstruction produces one or a set of HR images from a sequence of LR images, and it can solve the problem of transcending the limitations of optical imaging systems through the use of image processing algorithms, which presumably are relatively inexpensive to implement.

We divide 20 images of every group into 5 teams, and obtain an HR image of 11200*600 image resolution (Figure 3-6 (a)) from each team of 2800*150 by means of super resolution. This step takes approximately 18.9 seconds.

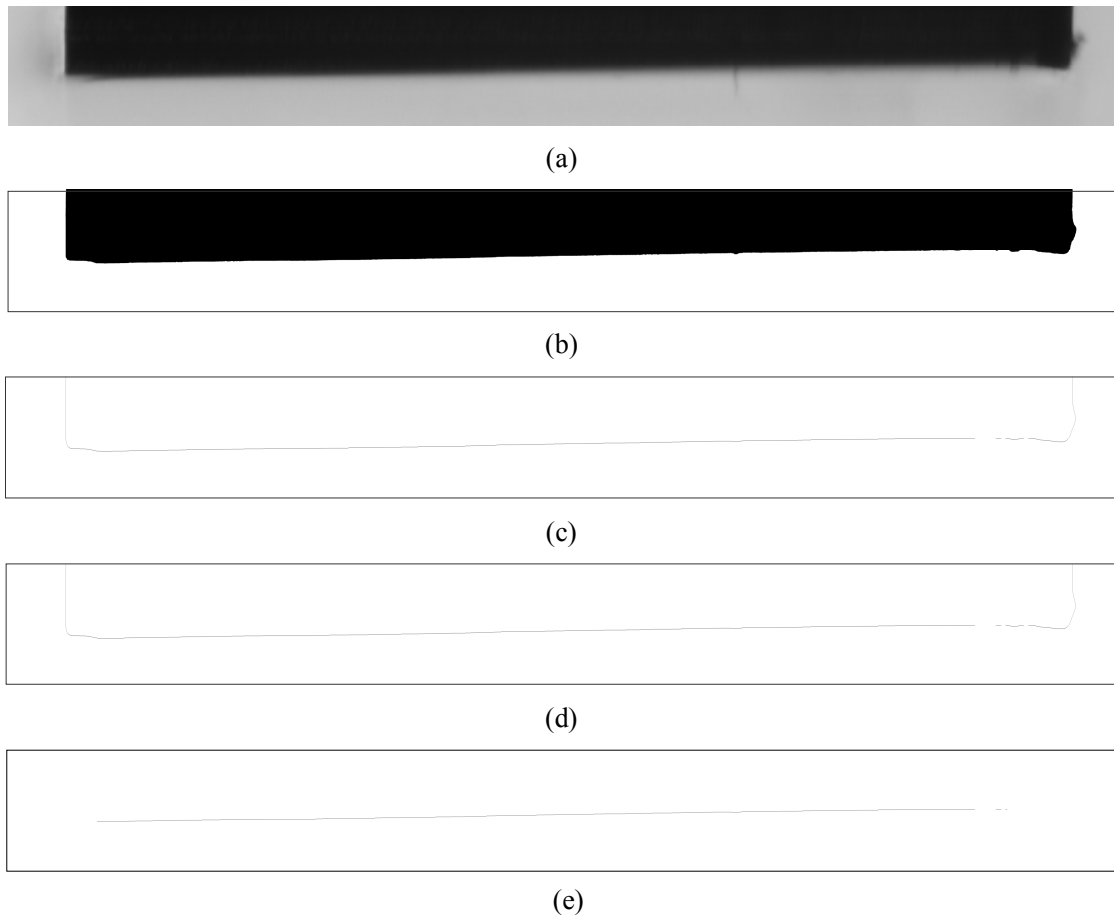


Figure 3-6. Image processing images.

(a) Super resolution image, (b) Enhanced image, (c) Canny image, (d) Robust contour following image, (e) Edge auto-segmentation image.

3.4. Image enhancement

The image's noise, edge and contrast are important factors that influence its quality. We employ median filtering to smooth the noise.

Contrast adjustment linearly scales pixel values between a lower and upper limit. Pixel values above or below this threshold are saturated to the upper or lower limits. The microscope is zoomed in on the HDH by a factor of 90 to obtain a clear image, but then the image shows too much HDH surface detail. It takes a considerable amount of time to find both limits, but once they have been determined, every image obtained under the same conditions will share the same parameters, even after rotating the HDH. During our experiment, we adjust images using a lot different threshold. Then we compare and choose the suitable one for all images. The lower and upper

limits we choose are 0.34 and 0.35. An enhanced image is shown in Figure 3-6 (b) and Figure 3-7 (a).

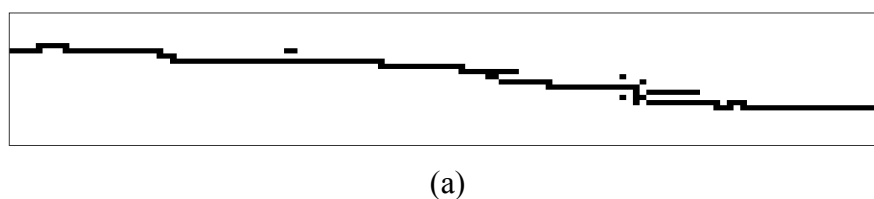
Adjustment parameters 0.34 and 0.35 mean that only adjust values between 0.34 and 0.35, while clip values excess this range. To compare with other parameters, we implement wider parameters 0.30 and 0.39 (Figure 3-7 (b)), and default 0 and 1 (Figure 3-7 (c)) on super resolution image.



Figure 3-7. Image adjustment results. (a) Adjustment of parameters 0.34 and 0.35; (b) Adjustment of parameters 0.30 and 0.39; (c) Default adjustment.

3.5. Edge detection

Edge detection method includes Roberts cross, Sobel, Prewitt, and Canny. For the Canny method, we set the parameter SIGMAR (the standard deviation of the Gaussian filter) to be 55. Figure 3-8 shows part image results of all four methods on the same image.



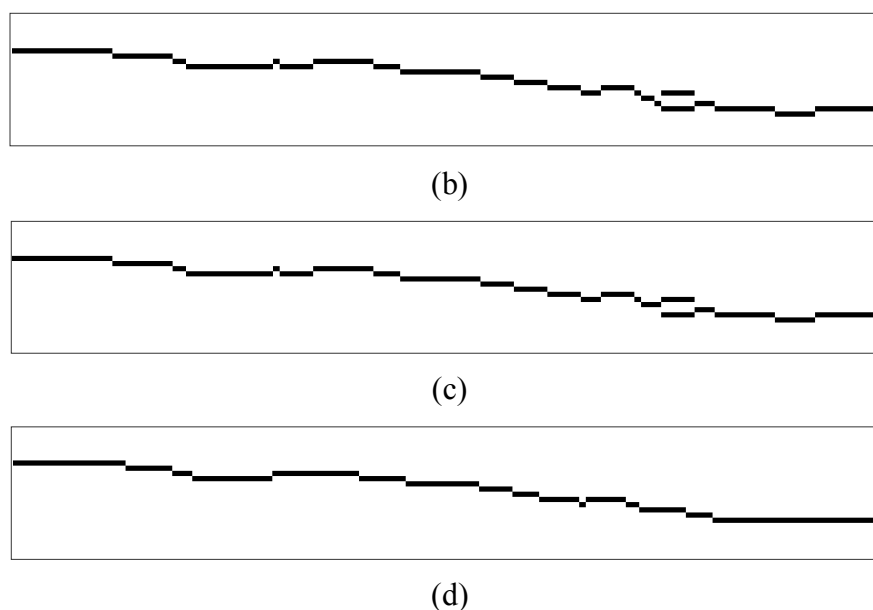


Figure 3-8. Results of different edge detection methods.

(a) Roberts cross method; (b) Sobel method; (c) Prewitt method; (d) Canny method.

Comparing with other methods, the Canny method has the ability to locate and mark the edge that is a minimal distance between the detected edge and a real edge. Also, there is only one response for each edge, which is very useful for our line analysis. So we employ Canny operator, and the result of HDH image is shown in Figure 3-6 (c).

3.6. Robust contour following

Contour following finds every contour point starting from the point we set, up to the contour end point, or when a break appears.

Since sometimes breaks (normally less than 5 breaks) may occur in our edges, we have developed a robust contour following method that fixes breaks. We add two points in the right and right-down directions instead of stopping when a break is encountered (Figure 3-9 (b)). Then contour following is resumed on the current image, while less than ten loops to avoid endless loop. Although a few new points are added to our image, the number is small compared with the more than 10,000 points in each image. Also, the new points are located very close to true points.

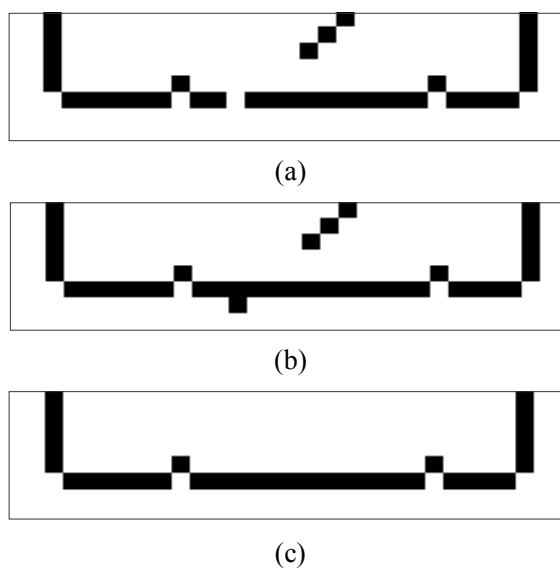


Figure 3-9. Robust contour following images.

(a) Input image, (b) Added two points image, (c) Result image.

A clear HDH contour is shown in Figure 3-6 (d), obtained after applying our contour following method.

3.7. Edge auto-segmentation

With contour following, we can obtain a clear HDH edge line, but we only need one side edge line to compute an angle. We present slop method to segment edge automatically, and it is called edge auto-segmentation.

We start from the center point (as the first point) of the edge and calculate slope of these 50 points (Figure 3-10 (b)) after including 49 points in the right direction; we add the first 25 points to our point list (Figure 3-10 (c)) if the slope result is inside our threshold range, then start the slope computation again starting from the 26th point. This continues until the slope result goes over the threshold result (Figure 3-10 (d)). Then we repeat the slope calculation starting from the center point but progressing to the left. Finally, we reconstruct a new image using the point list we have collected (Figure 3-10 (e) and Figure 3-6 (e)).

Since different thresholds cause to different results, we need to set proper threshold according to different experiment situation. However, this threshold can be settled because the acceptable range (from -0.05° to 0.05°) is fixed.

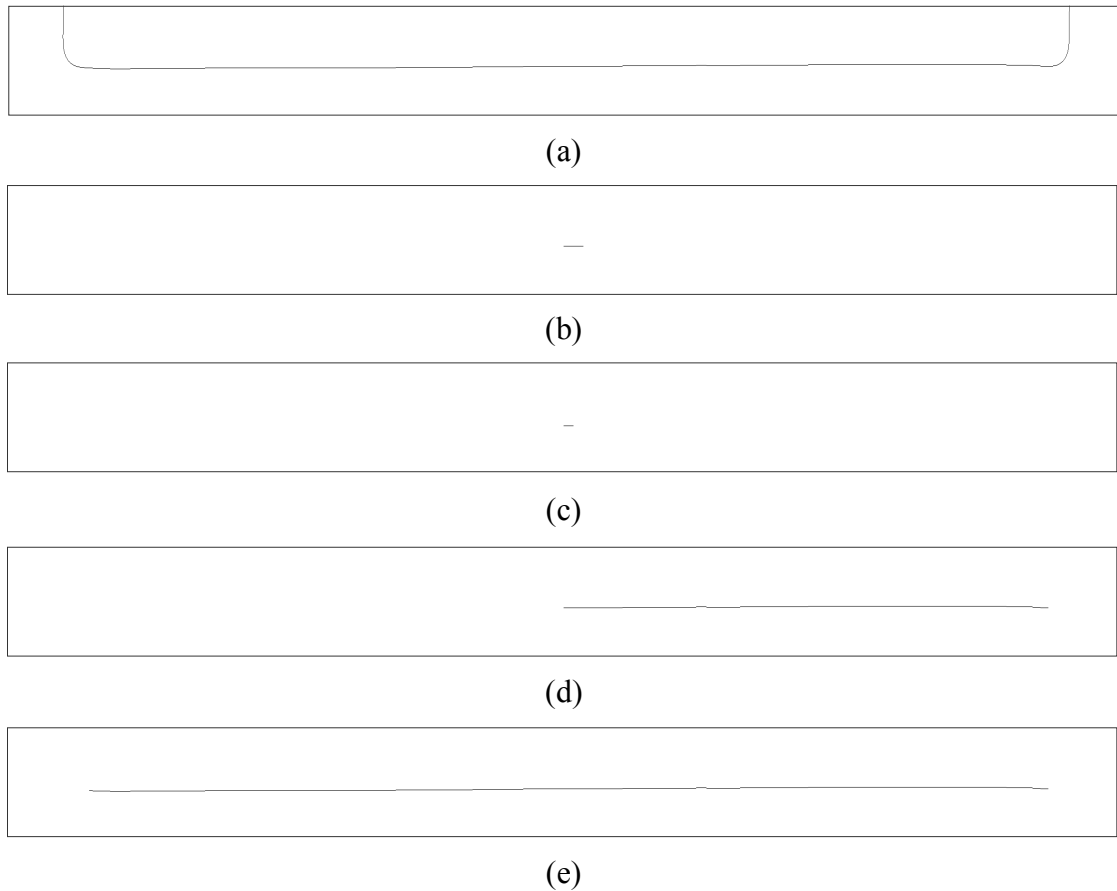


Figure 3-10. Image results for edge auto-segmentation.

- (a) A sample image, (b) 50 point image start from center to right, (c) 25 point image which will be added to list, (d) Edge segmentation from center to right direction, (e) Final image results.

3.8. Line fitting

After getting a clear HDH edge line, we can obtain every point's coordinates. Pitch angle, which is the angle between the HDH's edge line and the image edge, can be computed using these coordinates. There are many measurement methods to fit line: least squares method, eigenvalue-eigenvector method, and function `cvFitline`. We implement all three methods, and will compare their results in section 3.9.

3.9. Comparison and analysis

Many methods are implemented above. We will compare their abilities on the same images, and find out the better algorithm.

3.9.1 Image format comparison

TIFF, tagged image file format, is one of the most popular image file formats used on the web today. The TIFF image file format is based on the lossless compression method, allowing for an image to be stored without losing even a percent of its original data. Moreover, an image in TIFF format can undergo multiple edits with no negative effect on its quality parameters. At the same time, it also takes much space to store data.

JPEG is a standardized image compression mechanism, and stands for Joint Photographic Experts Group, the original name of the committee that wrote the standard. A useful property of JPEG is that the degree of lossiness can be varied by adjusting compression parameters. This means that the image maker can trade off file size against output image quality. Another important aspect of JPEG is that decoders can trade off decoding speed against image quality, by using fast but inaccurate approximations to the required calculations. Some viewers obtain remarkable speedups in this way.

To compare different image formats affect result or not, we also took each 10 images of TIFF format and three different JPEG quality levels (low, med, and high) at the same position. Angle results are shown in Table 3-1.

From the stdev (standard deviation) values, we can say that smaller value offers better precision. Therefore, TIFF format offers the best precision because that its stdev value is the smallest. For the different JPEG format, the higher quality (lower compression) offers better result. Although TIFF format is the best, it needs two much space to store data (35.8 MB per image), and high quality of JPEG format is 2.1MB. Huge store requirement is a big problem on the high speed and big amount production line. Therefore, we choose JPEG high quality format considering of good enough precision.

Table 3-1. Angle results of different image format.

	TIFF	JPEG (High)	JPEG (Med)	JPEG (Low)
1	0.18317°	0.16933°	0.12944°	0.10701°
2	0.18537°	0.17392°	0.12798°	0.09692°
3	0.18665°	0.16914°	0.14210°	0.10264°
4	0.18538°	0.16163°	0.14258°	0.09771°
5	0.18589°	0.15805°	0.14748°	0.10705°
6	0.19739°	0.16561°	0.15008°	0.11390°
7	0.19594°	0.16052°	0.15638°	0.12478°
8	0.19443°	0.17625°	0.15335°	0.12421°
9	0.19133°	0.17782°	0.15459°	0.11436°
10	0.19323°	0.17941°	0.15386°	0.12657°
stdev	0.00515	0.00760	0.01023	0.01373

3.9.2 Image adjustment comparison

The best adjustment parameters we choose are 0.34 and 0.35 (threshold 1), and this function will adjust image intensity values between 0.34 and 0.35 to values we set (normally we use default value, 0 and 1), while values below 0.34 and above 0.35 are clipped.

We set another wider threshold, 0.30 and 0.39 (threshold 2), and implement on the same images with threshold 1. Default adjustment parameters also have been implemented, while it offers us bad result image after detecting edge (Figure 3-11). This image is not suitable to find main edge line using contour following, and worse precision. Therefore, we only need to compare threshold 1 and threshold 2.



Figure 3-11. Result image of edge detection after using default adjustment.

Table 3-2. Angle results of different adjustment threshold.

	Threshold 1	Threshold 2
1	0.431499°	0.434177°
2	0.431921°	0.432394°
3	0.432542°	0.432851°
4	0.433406°	0.431894°
5	0.433716°	0.433955°
stdev	0.000945	0.000987

From Table 3-2, we can see that the stdev value of threshold 1 is smaller than of threshold 2. Therefore, Threshold 1 offers better precision.

3.9.3 Line fitting method comparison

We implement all three different line fitting methods on a group of images, least squares method, eigenvalue-eigenvector method, and function cvFitline. The method which is the minimum stdev value offers better precision.

Table 3-3. Angle results of different line fitting methods.

	Least square method	Eigenvalue-eigenvector	Function cvFitline
1	0.437088°	0.437088°	0.435403°
2	0.434268°	0.434268°	0.433291°
3	0.434729°	0.434729°	0.433796°
4	0.434308°	0.434309°	0.432993°
5	0.435957°	0.435957°	0.434476°
stdev	0.00122	0.00122	0.00097

From Table 3-3, we can see that least square method and eigenvalue-eigenvector method offer the same stdev value (almost the same angle results), and cvFitline offers the minimum stdev value. Therefore, we employ function cvFitline to fit our edge line.

3.9.4 Super resolution affection comparison

In this report, we employ super resolution method to enlarge image resolution. Angle results and computing time information of using and do not is recorded in Table 3-4 and Table 3-5.

Table 3-4. Angle results of super resolution affection.

	No SR	Using SR
1	0.39161°	0.43540°
2	0.38632°	0.43329°
3	0.39044°	0.43380°
4	0.39033°	0.43330°
5	0.39237°	0.43448°
stdev	0.00179	0.00097

Table 3-5. Computing time of super resolution affection.

	No SR (sec)	Using SR (sec)
Image cropping	0.03	0.03
Gray transform	0.03	/
Super Resolution	/	18.95
Image enhancement	0.13	1.42
Edge detection	5.63	89.16
Robust contour following	0.38	4.42
Edge segmentation	0.14	0.43
Line fitting	0.18	0.52
Total	6.49	114.93

From Table 3-4, we can see that stdev value has been changed to 0.00097 from 0.00179. This means that super resolution can improve system's precision.

From Table 3-5, super resolution step takes about 18.95 seconds, and all following steps take much more time than without using it. Finally, total time is also much more than without super resolution method.

The computer specification we used for experiment are: CPU: QuadCore Intel Core 2 Quad Q9400, 2.66GHz (8*333); Mother board: Asus P5QL-E; Display card: ATI Radeon HD 3800 Series (512 MB); EMS memory: 3328 MB; Operating system: Microsoft Windows XP Professional.

In the research of this report, it's not very necessary to employ super resolution because we have obtained high resolution and quality images already. If this image processing system will be implemented in production-line, video format is taken instead of taking single images because that taking video makes it possible to process in real-time. At this same time, video's resolution is needed to be enlarged. Super resolution method will be very useful in this situation.

3.10. Results

After all implementation and discussion above, we choose JPEG high quality format and function cvFitline from line fitting method list.

We took 60 images, consisting of 3 groups each of 20 images. Group (position) 1 was taken at the starting position, and the other two groups after a clockwise rotation step of 0.0226° (0.40mm). We divided 20 images from every position into 5 teams, each team consists of 4 images. Table 3-6 shows the results of all 3 positions.

Table 3-6. Pitch angle results taken by rotating step by step.

	Position 1	Position 2	Position 3
1	0.43540°	0.40414°	0.38155°
2	0.43329°	0.40274°	0.37718°
3	0.43380°	0.40291°	0.37788°
4	0.43300°	0.40415°	0.37778°
5	0.43448°	0.40584°	0.37886°
Minimum	0.43300°	0.40274°	0.37718°
Maximum	0.43540°	0.40584°	0.38155°
Median filter	0.43380°	0.40414°	0.37788°

The resolution of pitch angle in our experiment is 0.0226° . Therefore, we require precision and accuracy both smaller than 0.0226° . The angle differences between the maximum and minimum is 0.0024° , 0.0031° and 0.0044° for each position, so the precision meets our expectations.

Since parameter accuracy has to smaller than 0.0226° , range from -0.0113° to $+0.0113^\circ$ (value $\pm 0.0113^\circ$) means the same accuracy. To check the accuracy, we employ a median filter to choose an angle typical for each position (Table 3-4). Angle 0.43380° was chosen to stand for position 1; angle 0.40414° for position 2; and angle 0.37788° for position 3. We subtracted the angle of position 1 from position 2, and the angle of position 2 from position 3. The resulting differences are 0.0287° and 0.0263° , both within the range $0.0226^\circ \pm 0.0113^\circ$. Therefore, good accuracy has been obtained.

Since both precision and accuracy meet our needs, good resolution (0.0226°) has been obtained.

CHAPTER 4

SUMMARY

This report presents an image processing system to measure pitch and roll angles of HDH in the HSA. In this chapter, we conclude our work and make recommendations for future work.

4.1. Conclusion

Hard disk is a storage device which stores digitally encoded data on platters. A hard disk's main components include HDH, platter, spindle, actuator, and actuator arm.

HDH is one of the key parts in a hard disk because of its function and sensitive. Since HDH has to write electric data to platter and read magnetic data from platter, the floating height-the distance between the HDH and the platter-is between 0.127 and 0.305 μm . This height must be maintained within a proper range by making both HDH and platter parallel with horizontal plane in the HSA. Otherwise, there will be a head crash. It is easy to make the platter parallel, while it is not so for HDH because of its small size. To avoid head crash between platter and HDH during using this hard disk, the acceptable ranges of pitch and roll angles both are between -0.05° and 0.05° .

Although current measurement system, laser techniques system, offers us good resolution, we propose an image processing system to measure pitch and roll angles considering of its high price, 30,000 USD. In this thesis, the cost of camera and microscope are 12,000 and 4,500 USD respectively. Moreover, we will use a lens which offers the same function with microscope, but lower price, if this system is implemented in production-line.

Our measurement system starts from capturing clear images of an HDH using a microscope and a high-quality camera. The subsequent processing includes image cropping, super resolution, image enhancement, edge detection, robust contour following, edge auto-segmentation, and line fitting.

One of the key operator is super resolution, which reconstructs an HR image

from a sequence of LR images. It improves our precision since we combine more information from other LR images.

The allowable vary range of pitch and roll angles both are between -0.05° and 0.05° . Our measurement system offers an acceptable resolution of 0.0226° for pitch angle and 0.0288° for roll angle.

4.2. Recommendations for future work

In this thesis, we only took images of one single HDH each time. An image that contains whole HDH stack (around 6 HDHs) can be captured if we have big enough image resolution camera. Then crop each interested area, and compute them respectively.

If this image processing system is implemented on production-line, we shall calibrate with current laser system. After making the camera parallel with the horizontal plane, every values offered by this system will be correct.

REFERENCES

- [1] <http://wikipedia.org/>. Accessing time: Jan 12, 2010.
- [2] <http://www.pcguides.com/ref/hdd/index.htm>. Accessing time: Jan 12, 2010.
- [3] <http://keppanet.net/firms.com/keppanet/harddisk.htm>. Accessing time: Jan 12, 2010.
- [4] Sung Cheol Park, Min Kyu Park, Moon Gi Kang, "Super-resolution image reconstruction: a technical overview", *IEEE Trans. on Signal Processing*, Vol 20, no. 3, pp. 21 - 36, May 2003.
- [5] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar, "Fast and Robust Multi-frame Super-resolution", *IEEE Trans. on Image Processing*, vol. 13, no. 10, pp. 1327-1344, October 2004.
- [6] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar, "Advances and Challenges in Super-Resolution", Invited Paper, *International Journal of Imaging Systems and Technology*, Special Issue on High Resolution Image Reconstruction, vol. 14, no. 2, pp. 47-57, 2004.
- [7] S. Farsiu, M. Elad, and P. Milanfar, "Multi-Frame Demosaicing and Super-Resolution of Color Images", *IEEE Trans. on Image Processing*, vol. 15, no. 1, pp. 141-159, January 2006.
- [8] P.E. Eren, M.I. Sezan, and A.M. Tekalp, "Robust, object-based high-resolution image reconstruction from low-resolution video," *IEEE Trans. Image Processing*, vol. 6, no. 10, pp. 1446-1451, Oct. 1997.
- [9] A.J. Patti and Y. Altunbasak, "Artifact reduction for set theoretic super resolution image reconstruction with edge adaptive constraints and higher-order interpolants," *IEEE Trans. Image Processing*, vol. 10, no. 1, pp. 179-186, Jan. 2001.
- [10] Sean Borman, Robert Stevenson, "Spatial Resolution Enhancement of Low-Resolution Image Sequences; A Comprehensive Review with Directions for Future Research", July, 1998
- [11] R.Y. Tsai and T.S. Huang, "Multipleframe image restoration and registration," in *Advances in Computer Vision and Image Processing*. Greenwich, CT: JAI Press Inc., 1984, pp. 317-339.

- [12] M. Elad and A. Feuer, "Restoration of a single superresolution image from several blurred, noisy, and undersampled measured images," *IEEE Trans. Image Processing*, vol. 6, no. 12, pp. 1646-1658, Dec. 1997.
- [13] N. Nguyen, P. Milanfar, and G. Golub, "Efficient generalized cross-validation with applications to parametric image restoration and resolution enhancement," *IEEE Trans. Image Processing*, vol. 10, pp. 1299-1308, Sept. 2001.
- [14] V.N. Dvorchenko, "Bounds on (deterministic) correlation functions with applications to registration." *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 5, no. 2, pp. 206-213, 1983.
- [15] Q. Tian and M.N. Huhns, "Algorithm for subpixel registration," *Computer Vision, Graphics, Image Proc.*, vol. 35, pp. 220-233, 1986.
- [16] C.A. Bernstein, L.N. Kanal, D. Lavin, and E.C. Olson, "A geometric approach to subpixel registration accuracy," *Computer Vision, Graphics, and Image Proc.*, vol. 40, pp. 334-360, 1987.
- [17] L.G. Brown, "A survey of image registration techniques," *ACM Comput. Surveys*, vol. 24, no. 4, pp. 325-376, Dec. 1992.
- [18] A.M. Tekalp, *Digital Video Processing*. Englewood Cliffs, NJ: Prentice Hall, 1995.
- [19] Zhang Yepeng, Tao Yuezhen, Fan Zhiyong, "Application of Digital Image Process Technology to the Mouth of Beer Bottle Defect Inspection", *The Eighth International Conference on Electronic Measurement and Instruments*, 2007.
- [20] He Qing-hang, Zhang Zhen-xi, Li Zheng, Xu Zheng-hong, "The Processing of the Degraded Medical Digital Image's Image Enhancement", *Proceedings of the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*. 2005.
- [21] Yong Hoon Lee, Saleem A. Kassam. "Generalized Median Filter and Related Nonlinear Filtering Techniques". *IEEE Trans. On ASSP*. 35(1): 60-69, 1987.
- [22] Peli E, "Contrast in complex images", *Optical Society America*, 17(10), 1990.
- [23] Mohsen Sharifi, Mahmoud Fathy, Maryam Tayefeh Mahmoudi, "A Classified and Comparative Study of Edge Detection Algorithms", *ITCC'02, Proceedings of the International Conference on Information Technology: Coding and Computing*, 2002.
- [24] http://www.pages.drexel.edu/~weg22/can_tut.html. Accessing time: Jan 12, 2010.

- [25] Wing-Nin Leung, Ng C.M, Yu P.C, "Contour following parallel thinning for simple binary images", IEEE International Conference on Systems, Man, and Cybernetics, Volume 3, 8-11, Page(s):1650 - 1655 vol.3, 2000.
- [26] Gary Bradski, Adrian Kaehler, "Learning OpenCV", O'Reilly, 2008.
- [27] Fu Dongjin, Montri Karnjanadecha, and Anant Choksuriwong, "An Angle Measurement of Hard Disk's Head in the HSA using Image Processing", CISE'09, International Conference on Computational Intelligence and Software Engineering, 2009.
- [28] Fu Dongjin, Montri Karnjanadecha, and Anant Choksuriwong, "The Angle Measurement of a Hard Disk Head using Super Resolution", ICIP'10, IEEE International Conference on Image Processing, 2010.(Submitted).

VITAE

Name Mr. Fu Dongjin

Student ID 5010120149

Educational Attainment

Degree	Name of Institution	Year of Graduation
Bachelor of Engineering (Automation Engineering)	JiangXi University of Science and Technology P.R.China.	2007

Scholarship Awards during Enrolment

June, 2008 ~ Oct, 2008: Tuition fee waiving scholarship.

Nov, 2008 ~ Mar, 2009: Tuition fee waiving scholarship.

June, 2009 ~ Oct, 2009: Tuition fee waiving scholarship.

List of Publication and Proceedings

- [1] Fu Dongjin, Montri Karnjanadecha, and Anant Choksuriwong, "An Angle Measurement of Hard Disk's Head in the HSA using Image Processing", CISE'09, *International Conference on Computational Intelligence and Software Engineering*, 2009.
- [2] Fu Dongjin, Montri Karnjanadecha, and Anant Choksuriwong, "The Angle Measurement of a Hard Disk Head using Super Resolution", ICIP'10, *IEEE International Conference on Image Processing*, 2010.(Submitted).