AUTOMATED QUALITY INSPECTION ON TILE BORDER DETECTION USING VISION SYSTEM

YAP SHER KEE

BACHELOR OF MECHATRONICS ENGINEERING WITH HONOURS

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2019

AUTOMATED QUALITY INSPECTION ON TILE BORDER DETECTION USING VISION SYSTEM

YAP SHER KEE

A report submitted
in partial fulfillment of the requirements for the degree of
Bachelor of Mechatronics Engineering with Honours

Faculty of Electrical Engineering

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2019

DECLARATION

I declare that this thesis entitled "AUTOMATED QUALITY INSPECTION ON TILE BORDER DETECTION USING VISION SYSTEM" is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature : --

Name : --

Date : --

APPROVAL

I hereby declare that I have checked this report entitled “AUTOMATED QUALITY INSPECTION ON TILE BORDER DETECTION USING VISION SYSTEM” and in my opinion, this thesis it complies the partial fulfillment for awarding the award of the degree of Bachelor of Mechatronics Engineering with Honours.

Signature : ....

Supervisor Name : ....

Date : ....

DEDICATIONS

To my beloved mother and father

ACKNOWLEDGEMENTS

I wish to express my sincere appreciation to my Final Year Project supervisor, Dr. Hairol Nizam Bin Mohd Shah for his support and suggestion throughout the semester. He has given many innovative and useful ideas for cultivating my interest. He has lent a hand to me for my further development of creativity in engineering through this project. I would also like to express my sincere thanks to the panels, PM Dr. Ahmad Zaki bin Hj Shukor, Puan Norafizah binti Abas and Puan Fadilah binti Abdul Azis for giving some useful advice and comment to improve my project. Without their continued support and interest, this project would not have been same as presented here.

I also want to show my gratitude to my parents and my family for continually pushing me forward in my pursuits as an engineer. They have been there for me through everything and given positive support as I grow. I also appreciate my fellow course mates who have been with me over the past few years, their support and assistance have been key in helping me.

ABSTRACT

Most of the ceramic tile industry still doing the quality control by manually. The accuracy of the manual inspection by human is lower due to the harsh industrial environment with noise, extreme temperature and humidity. A camera should replace the human eyes to recognise the defect tile effectively. Thus, a suitable method have to investigate for implementing this function. This project aim to design and develop an automated quality inspection in ceramic tile industry using vision system. The performance in term of detection accuracy for the system is analysed. An Imaging Source Series CMOS industrial camera is used to capture the tile border. The system is implemented in the MATLAB software. Image processing with Canny edge detection technique and morphological operation are used to segment and extract the tile border edge. The threshold level of image processing, focus and iris of camera and illumination of the light are adjusted to improve the performance of the system. The system developed is only to detect cracks occur on the edge of the tile border, middle crack such as pinhole is not included. The overall automation process involves image capturing, image processing, defect detection algorithms and decision making. The defect detection algorithms are developed to differentiate the defective tile. The automated quality inspection for the defect detection of tile border using vision system based on the background subtraction method and gradient variation of the tile border edge are presented in this research. The system using background subtraction method has achieved 50% accuracy in identify the status of tile since it consist of many limitation. By evaluate the gradient variation on the tile border edge, the accuracy of the defect detection has achieved 80% in identify the tile condition. The performance of the second method is relatively strong since the process of detection is considered in many aspects. In future, a consistent workspace such as in a production line can be achieved and reduce the error. The good and defect tiles will be classified and divided to different place by design a conveyer sorting system.

ABSTRAK

Kebanyakan industi jubin seramik masih menggunakan kawalan kualiti secara manual. Ketepatan pemeriksaan secara manual oleh manusia adalah rendah kerana persekitaran perindustrian yang dipenuhi dengan bunyi bising, suhu yang melampau dan kelembapan yang terlalu tinggi atau rendah. Kamera patut digunakan dengan mengantikan mata manusia untuk mengenali kecacatan di jubin dengan berkesan. Oleh itu, keadah yang sesuai harus dikaji demi melaksanakan fungsi ini. Objektif projek ini adalah untuk mereka cipta sistem pemeriksaan kualiti automatik dalam industri jubin seramik menggunakan penglihatan mesin. Prestasi sistem ini dianalisis berdasarkan ketepatan pengesanannya. Kamera industri Imaging Source CMOS digunakan dalam pengambilan gambar sisi jubin. Sistem ini dilaksanakan dalam perisian MATLAB. Pemprosesan gambar dengan mengunakan teknik "*Canny edge detection*" dan "*morphological operation*" untuk membahagi dan memperolehi hujung sisi jubin. Tahap ampang untuk pemprosesan gambar, tumpuan dan iris kamera serta pencahayaan lampu dilaraskan untuk menambahbaikkan prestasi sistem ini. Sistem ini dicipta hanya untuk mengesan retakan di hujung jubin, retakan di tengah jubin seperti lubang adalah tidak dapat dikesan. Keseluruhan proses automatik ini merangkumi pengambilan gambar, pemprosesan gambar, algoritma pengesanan kecacatan dan pengambilan keputusan. Algoritma pengesanan kecacatan ini direka cipta untuk memebezakan jubin yang mempunyai keccacatan. Pemeriksaan kualiti automatik untuk pengesanan kecacatan hujung jubin adalah berdasarkan kaedah "*background subtraction*" dan "*gradient variation*" dalam hujung jubin dibentangkan dalam kajian ini. Sistem yang menggunakan "*background subtraction*" telah mencapai ketepatan sebanyak 50% dalam memastikan keadaan jubin memandangkan ia mempunyai banyak pengehadan. Dengan menilaikan "*gradient variation*" di hujung jubin, ketepatan pengesanan kecacatan telah mencapai 80% dalam megenalpastikan keadaan jubin. Prestasi keadah kedua adalah lebih baik disebabkan oleh proses ini merangkumi banyak aspek dalam penilaian. Pada masa akan datang, ruang kerja yang tetap seperti dalam kawasan produksi dapat dicapai dan mengurangkan kesilapan sistem. Jubin dalam keadaan baik dan mempunyai kecacatan dapat dibezakan dan dibahagikan ke tempat belainan dengan menghasilkan satu sistem pengasingan dan penghantaran.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

MATLAB	-	Matrix Laboratory
ROI	-	Regions of Interest
LED	-	Light-emitting Diode
HPC	-	Performance Cluster
CUDA	-	Compute Unified Device Architecture
PC	-	Personal Computer
USB	-	Universal Serial Bus
CMOS	-	Complementary Metal Oxide Semiconductor
RGB	-	Red, Green, Blue
GPU	-	Graphics Processing Unit
RAM	-	Random Access Memory
HDD	-	Hard Disk Drive
VRAM	-	Video Random Access Memory
OS	-	Operating System
m	-	Meter
IoT	-	Internet of Things

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Project Introduction

In this industry 4.0 era, automated system is indispensable to a modern manufacturing industry. The need of the mankind has demanded increase in productivity with the improved quality of the products. This has led to innovations, and these innovations have transformed the traditional manufacturing to advanced manufacturing. Quality control is a step in manufacturing to ensure customers receive a good product that meet their needs and free from defects. Consumers will face the risk if the quality of product is done in the wrong way [1]. There are several methods of quality checking. Each method has its own advantages and disadvantages. Contact type equipment consumes more time for quality check than non-contact type [2].

The conventional quality checking of a component is made by taking one sample out of a lot to ensure the quality of the particular lot. Such quality checking method may lead to rejection of the whole lot or even acceptance of defective parts. In order to ensure the quality of the product, each component has to undergo quality check, which raises the need of in-process inspection. In-process inspection ensures successful control over the quality of the component, reduces the quality check time, ensure inspection of each component and reliability as well as the efficiency of the system [3].

To fulfil the automated and in-process inspection, machine vision can be applicate. A machine vision system is a type of technology that enables a computing device to inspect, evaluate and identify still or moving images. The main problem in developing efficient machine vision is to translate the human visual perception into sequential and logical operations. In purpose to find some other ways for defect detecting the image processing methods are developed [4]. Edge detection is an image processing technique to find the boundaries of objects in images by detecting the discontinuities in brightness. In the ceramic tiles industry, the tile border can be found out using this method to identify the defect condition. Figure 1.1 shows the tile in the industry required a quality check of the tile border defect.

Figure 1.1: Quality Check of Tile Edge Defect [5]

There are many object analysis functions in MATLAB which are detect edges, circles and lines. MATLAB which developed by MathWorks is a multi-model numerical computing environment and possessed own programming language [6]. Image processing and computer vision is one of the product in MATLAB. Algorithm development is medium for image processing and computer vision due to each situation is unique and good solutions require many kinds of iterations on design [7]. It provided an Image Processing Toolbox contains many kinds of set of reference algorithms and workflow apps that can be applied for image processing, analysis, visualization, and algorithm development. It able to carry out image processing such as image segmentation, image enhancement and noise reduction. Many visualization functions and applications to explore images and produce histograms as well as manipulate regions of interest (ROIs) [8].

1.2 Motivation

Most production process is automated in ceramic tile industry, yet the final step which is quality inspection still monitored manually. Failures on ceramic tile always causing by human error. The manual inspection on detecting the defect is based on human decision. Inspectors may feel eyes fatigue and tired causing the fail in inspection [9]. The failures in product will make customer on risk and the vendor, installer and some implicated person have to take responsible on it.

Failure in quality inspection will cause the defect product shipped out to customers. Customer will complain and ask for recalled which will cause the company loss of money. Besides, customers not only ask for the product that failed specifications or had to be recalled but the company's operations in general also will be question for the quality. The product may not be accepted if the company under excessive warning letter. The image and reputation of the company will be destroyed and cause loss in customers' trust and business in this competitive field. Associated the financial impact due by lost sales, lower production with increased production costs and material cost increased. Company will suffer tremendous loss for failing in quality control so that a proper solution should be implemented to solve the error in inspection of defect product [10].

1.3 Problem Statements

This research is to create an automated quality inspection system using vision system. This system is to check the border of ceramic tile whether it is good or defected by automated process instead of human monitored. Quality check by human causes a lot of time consumed and low efficiency. This also causes a high cost consumed in production since manually quality check cannot give feedback on the trend of defect and quality of the product cannot be improved efficiency.

In current research of tiles automated inspection system, the camera normally placed on the top for inspecting the surface of tiles. The inspection will only focus on the top side of tiles while the bottom side condition unable to detect. Camera intensity is the main element that need to concern in a vision system. If intensity is too high or low, an exposure and dark image will be formed respectively. The image with noise also influence the performance of the inspection system. With the aid of light source can improve the system performance but sometime the light which directly illuminate the tiles will provide unwanted reflection and influence the output image. The factors that will influence the performance of the system should be found out to reduce the error occur.

1.4 Objectives

This research has the following objectives:

1. To design and develop the automated quality inspection for the defect detection of tile border using vision system.
2. To obtain the characteristic of crack based on the tile border line.
3. To analyse the performance of the tiles border line detection in term of accuracy.

1.5 Scope

This research contain some scopes to narrow down the area of study. The defect detection algorithms developed is only tested on 10 pieces 30cmx30cm plain ceramic tile with different crack area on border. The distance between camera and tiles always fixed during the testing which is 2 m. The defect type to detect in the system is only focus on the crack defect of tile border. The system developed is only to detect cracks occur on the edge of the tile border, middle crack such as pinhole is not included. The tiles is always in static position for inspection. An Imaging Source CMOS industrial camera is used to capture the images. The focus and iris of the camera are adjusted manually to control the clarity and exposure of the image captured. MATLAB software is used for design the image processing, defect detection algorithms and data analysis of the image acquired. A laptop is used for executing the system program in the MATLAB.

CHAPTER 2

LITERATURE REVIEW

2.1 Background Theories

This section is basically explained about the background theories of the project. Vision inspection system and pixels are defined in the subsections. These theories are aided in developing the research.

2.1.1 Vision Inspection System

Vision inspection systems is an image-based automated inspection that provide convenience for a variety of industrial and manufacturing used. This kind of systems are now become commonly applied in automated inspection, quality control, sorting, robot guidance and so on.

The inspection systems required a camera or multiple cameras and even video and lighting. Vision systems are able to perform measuring, verifying parts are in the correct position and recognising the shape of parts at high speeds. Computer software can used for processing images captured when want to assess to capture data. Vision system is intelligent enough to do decision-making for controlling an operator in doing some output. These systems can provide a constant stream of information when embedded into the production line.

Any industries in which quality control is important suitable to apply these vision inspection systems. These system enable data collecting that help in efficiency improvement for manufacturing lines, sorting, packing and other tasks. Furthermore, the data obtained by the system can determine problems with the manufacturing line or other function to improve efficiency, stop bad processes and identify defective products.

The design of the vision inspection systems can be specialized to meet the needs of many industries due to these systems integrate various kinds of technologies. Therefore, a lot of companies have used this technology for quality control purposes and even security purposes. Industries using vision inspection systems involved in a wider field such as automation, robotics, pharmaceuticals, packaging, automotive, food and beverage, semiconductors, medical imaging and much more.

Overall, the advantages of vision inspection systems consist of production improvements, uptime increased and cost reduction. Industries allow conducting completely inspection of parts for quality control purposes using vision systems. These system guarantee that all products will meet the customers' needs and specifications [11].

2.1.2 Pixels

The smallest element in an image is called pixel which is also known as pel. Each pixel represent a value. The value of the pixel is between 0 and 255 in an 8-bit gray scale image. The value of a pixel at any point correspond to the intensity of the light photons bombardment at that point. Each pixel contain a value proportional to the light intensity at that specific location. The value 0 means absence of light which denotes dark. It further means that whenever a value of pixel is 0 indicate that black colour would be formed at that point. For white colour is value 255.

There may be thousands of pixels that joined together and make up an image. The image is zoomed to the level that some pixels division able to see. Figure 2.1 have shown to explain this feature.

Figure 2.1: Pixels Division in Zoomed Image [12]

Next, an image is defined as a two-dimensional matrix. In this case, the number of pixels would correspond to the number of rows multiply with number of columns. It can be said that the number of (x,y) coordinate pairs make up the total number of pixels. The mathematically represented as below:

$$Total\ number\ of\ pixels = number\ of\ rows \times number\ of\ columns \quad (2.1)$$

A binary image is a digital image that has only two possible values for each pixel which are 1 and 0. Typically, the two colors used for a binary image are black and white. Value 0 represent black while value 1 represent white [12].

2.2 Review of Previous Related Works

This section is discussed about the review of previous related works done by other researchers. Methods for defect detection, edge detection algorithms, type of defects, the processors used and light sources applied in vision system are reviewed and compared for finding the advantages and limitations of various aspect before implementing the own project.

2.2.1 Methods for Defect Detection

Ceramic tile defect can be detected by many methods. The more traditional method is detected by human eyes. A research is conducted by F. Ozkan [9] using an eye-tracker to assess workers when detect the surface defect of ceramic tile. The workers will mark a straight line on a surface of tile when they realise the defect occur. The marked tile will be sorted automatic by sensor detected. The research found that an expert worker have shown a systematic pattern when inspect the ceramic tile on conveyer. A good inspectors are evaluated by accuracy and speed are relatively high and make many brief eye fixations during the time they have to inspect. For a novice worker, the accuracy and speed will relatively low. Human eyes can detect some obvious defects but not all recognizable by human eyes. Human resources also expensive to use and sometime they are not accurate enough for visual controlling [13].

Time of Flight Diffraction (TOFD) technique using ultrasonic sensor is one of a non-destructive testing in the quality control of ceramic tile. It used to do the mapping of edges position. Ultrasound is transmitted and reflected in a normal direction if the tile is perfect. Diffracted wave will emitted with a wide angular range when a crack is detected. This method saving production cost but may affected by temperature, dust, vibration, humidity, roughness and movement of tile [14]. A laser speckle photometry also applied in ceramic industries for detection of micro-cracks on the surface of ceramic. A speckle pattern will varies based on the thermal and mechanical excitation of object. Heat will be distributed on the surrounding of defect area. Camera is positioning above the object act as a detector of the pattern change to recognise the defect such as crack [15].

There are many research of defect detection on ceramic tile with automated visual inspection system based on image processing method. The detection speed and accuracy rate are high which extremely improves the accuracy, stability and efficiency of product detection [3]. Image processing can be done within production line process. The cameras scan the tile quickly without stalling it and fixing position [13]. Image processing required less time to do defect detection and many algorithms have available to classify different type of defect [16]. In the research of Y. C. Samarawickrama by using Matlab image processing technique to detect the surface defect of tile, the accuracy have reached 96.36% from 110 sample and the rate of perform image processing and result delivering are just consumed 2 seconds [17]. The comparison of the methods for defect detection is shown in Table 2.1.

Table 2.1: Comparison of Methods for Defect Detection

Researcher	Methods used	Advantage	Limitation
F. Ozkan and B. Ulutas [9] E. Golkar et al. [13]	Human	•Decision making •Easy to train	•Eyes fatigue and tired causing the fail in inspection.
H. M. Elbehiery et al. [14]	Ultrasonic sensor	•Destructive Testing •Not affect future usefulness	•Diffracted wave depended. •Do not have image reference.
B. Bendjus et al. [15]	Laser speckle photometry	•Speckle image show defect •More details	•Reflective index, angle between camera and laser needed to do future adjustment.
R. Kiran et al. [3] E. Golkar et al. [13] A. N. Shire [16] Y. C. Samarawickrama and C. D. Wickramasinghe [17]	Image processing	•High detection speed and accuracy rate •Improves the accuracy and the stability of detection	•Intensity change will affect the performance.

2.2.2 Edge Detection Algorithms

Edge detection is the first step in image processing. Its function is to identify the discontinuities intensity in the boundaries of a homogeneous regions in an image. It is important for the next step which is edge extraction. Edge detection make the discontinuities apparently for the edge extraction. There are many edge detectors developed to produce edges that provide for faster and precise recognition of object from an object greyscale image [17,18].

One of the most practical and frequently used edge detection algorithms is Canny edge detector. Canny operator can be said as an optimal detector. It function as make the image smoothly and find the gradient to eliminate insignificant edges with thresholding. This method obtained very good results in detected cracks, scratches, spots and blobs defect on tile [19]. Matlab software have the Canny edge detector library which can apply to do the image processing purpose. The automated ceramic tile surface defect detection system achieve an higher accuracy of 96.36% with using this technique in Matlab [17].An automated visual inspection system of ceramic tile border defect also applied Canny technique to focus the line gradient for histogram subtraction with the aid of morphological filter for eliminate noise [13].

Laplacian of gaussian operator is a second derivative edge detector can be used for edge linking method [16]. Other kinds of gaussian edge detectors such as canny is more complex. The operator could find the proper edge location and test wider areas around the pixel. In a study of real time defect detection method for high speed bar in coil from S. H. Choi et al. [20], they used laplacian filter to isolate image according to gray levels. By using this technique, they unable to recognise the orientation of the edge. It show where the gray level intensity function changes, the orientation of edge is not finding. Thus, this kind of technique inaccurate for detect corner and curves [18].

Sobel and Prewitt operator is the first stage of edge detection to evaluate the derivatives of image intensity. The operators are simple and used to detect the edges and orientation but they do not have accurate sensitivity to noise [18]. The study from E. Golkar et al. [13] have conducted an algorithm of ceramic tile length and width defect detection by using Prewitt and Sobel techniques in beginning step of vertical line extraction. The study found that the fluctuation of Prewitt's method is almost same with the real size changes measured by digital caliber. An approximate deviation of 1.44% is show on the maximum relative-error of both techniques.

M. Roushdy [18] have found that Canny edge detector are better than Laplacian of Gaussian while Laplacian of Gaussian is better than Prewitt and Sobel based on noisy image. Table 2.2 have compared the edge detection algorithms.

Table 2.2: Comparison of Edge Detection Algorithms

Researcher	Methods used	Advantage	Limitation
Z. Hocenski et al. [19]	Canny	•Better detection in noise condition •Smoothing the image reduces the noise	•Complex.
E. Golkar et al. [13]			
Y. C. Samarawickrama [17]			
M. Roushdy [18]	Sobel and Prewitt	•Simple	•Inaccurate detection sensitivity in case of noise.
E. Golkar et al. [13]			
A. N. Shire et al. [16]	Laplacian of Gaussian	•Finds correct place of edge	•Unable to recognise the orientation of the edge. •Inaccurate for detect corner and curves.
S. Choi et al.[20]			
M. Roushdy [18]			

2.2.3 Type of Defect

Ceramic tile will face some defect cause by the mechanical damaging during the production line. Research have found few patterns of defects from the existing defect detection methods and are shown in Table 2.3.

Table 2.3: Type of Ceramic Tile Defects [20,21]

Type of Defects	Description
Cracks	Break down, slit of tile
Pinhole	Scattered isolated pinpoint spot, craters, small bubbles
Corner	Break down of tile corner
Edge	Break down of edge
Blob	Water drop spot on tile surface
Dirt	Dust or glaze residue
Scratch	Scratch on tile surface
Size	Incorrectly length, width, thickness of tile

Corner and edge defect have detected by obtained angle change using thresholding and contour tracing algorithm and dot product formula to identify the tile corner and edge defect [22,23,24].

Edge crack, curvature and size defect can be detected by edge extraction techniques with looking pixels in the image where edges are likely to occur by looking for discontinuities in gradients and edge linking to produce descriptions of edges [12,21,24,25].

An abnormal reflected light used to detect the cracks defect. A normal tile is expected to reflect light at the same angle. For a defects from an uneven surface like cracks will reflect light in a different direction to the rest of the surface. It can be detected by placing the camera at where the abnormal reflected light occur, any light reflect to the camera considered as defect tile [21]. The type of defect are compared in Table 2.4.

Table 2.4: Comparison of Type of Defect

<table>
<tr><th>Methods used</th><th>Description</th><th>Differences</th></tr>
<tr><td>Edge Crack</td><td>Break down of edge.
</td><td>•Most of the edge detection method can detect.
•High accuracy without depending on the specification of camera.</td></tr>
<tr><td>Corner Crack</td><td>Break down of tile corner.
</td><td rowspan="2">•Need to train the system with the good tiles.
•Low accuracy due to uncertainty highly depend to the resolution of the camera.</td></tr>
<tr><td>Size</td><td>Incorrectly length, width, thickness of tile.</td></tr>
</table>

2.2.4 Processor

Automated visual inspection system based on computer vision need to be processed and come out a decision. Computing speed is considered as an important factor to reduce the time consume in manufacturing. As a demand of precise and fast processing speed, there are some technologies applied to improve the system.

The study of Z. Hocenski et al. [26] have suggested an idea of using parallel computing which are computer clusters. Computer cluster is a set of computer work at a same time and function like a unique computer. They often connected through local network in a same area. They supply great processing power and memory space for faster processing. High-performance cluster (HPC) as shown in Figure 2.2 is implemented and LAM/MPI library is used to shorter the processed time. It is a communication tools that run a parallel program on a memory system.

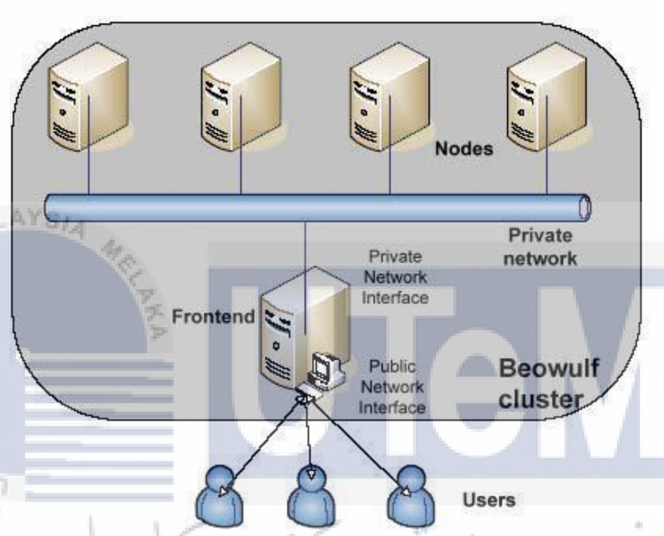


Figure 2.2: HPC Cluster as Beowulf Cluster [26]

Another parallel processing is develop algorithm using Compute Unified Device Architecture (CUDA). Parallel processing used to reduce the computing time and enhance the performance on Graphical Processing Unit (GPU). Parallel algorithm increase the speed in spot defect and crack defect detection compared to sentential algorithms [23,26].

Most of the research used an ordinary personal computer (PC) as a processing hardware. For example, PC is used to control the camera for image acquisition, motor and processing the image according to specific algorithm [12,27]. PC is a simple hardware that can transfer the information to controller and doing image processing [3]. M. Bohlool and S.Taghanaki [29] conducted an automated visual inspection system based on simple PC, camera and image processing software for defect detection. It is cost-efficiently, more economic and accessible for common medium industries. The processors are compared and list out in Table 2.5.

Table 2.5: Comparison of Processor

Researcher	Methods used	Advantage	Limitation
Z. Hocenski et al. [26]	Computer Cluster	•Great processing power and memory space. •Shorter the processed time parallel program.	•Required many computer.
Z. Hocenski et al. [24]	CUDA (Compute Unified Device Architecture)	•Parallel processing reduce the computing time enhance the performance on GPU.	•Only reduce if the algorithm can be parallel processing.
K. Ragab and N. Alsharay [27]			
E. Golkar et al. [13]	PC (ordinary personal computer)	•Simple hardware transfer information to controller and run algorithm. •Economic and accessible for common medium industries.	•Low processing power.
R. Kiran et al. [3]			
B. You et al. [28]			
M. Bohlool and S. R. Taghanaki [29]			

2.2.5 Light Sources

In a vision system, image acquisition is a function that directly influence the decision making of a system. Capturing a minimum noise images can reduce the error on the results. Thus, light resources considered as an important factor that will affect the efficiency of image acquisition. Halogen and LED lights are the most common illuminate resources in the vision inspection system.

Z. Hocenski et al. [30] had conducted a study about simulation models for several constructions and combinations of halogen lighting sources for visual inspection of ceramic tiles. Halogen light bulb of 150W is used as a lighting source. Due to it does not change fast the light intensity after being switched on and becoming incandescent. The increase of lighting sources increase the illumination and a suitable light geometry present a uniform illumination on the tile.

By using LED light source, B. You et al. [28] found that right recognition rate can reach 99% of ceramic tile detection system based on image processing. The maximum relative-error of both techniques shows an approximate deviation of 1.44% in [13]. For a recent research conducted by Z. Hocenski et al. [31], it used LED panel in the system and found that it was almost the same effect achieved as ordinary light bulbs. LED is selected due to its lower power consumption which energy that will consume on one panel is almost 1.5W. LED is a semiconductor that converts electrical energy directly into light without produce heat like light bulbs.

According to the illumination simulation of LED lighting panels for different diode distances and different heights from camera and panel, it was found that the tile defects are best observed by placing two panels with white LEDs light, set above lateral of observed tile. Solution with the panel positioning directly above the tiles gives bad results, because of the reflection surface of the tiles. LEDs of other colour is not consider to use because of the tiles lighted in different colour are less visible on image shot with digital camera.

Illumination quality is vital because of the several algorithms for fault detection methods based on pixel intensity comparison. This illumination mostly used on detecting black dots distortion and cracks which cause by mechanical damaging. Most bright tile colours is used without high contrast. The research claims that illumination uniformity depends on light sources configuration or geometry and that illumination increases by using more lighting sources. The unwanted reflection from the tile surface should be concerned to prevent reflection to the digital camera applied for automated visual inspection [29,30]. Comparison of light sources used in vision system are presented in Table 2.6.

Table 2.6: Comparison of Light Sources Used in Vision System

<table>
<tr><th>Researcher</th><th>Methods used</th><th>Advantage</th><th>Limitation</th></tr>
<tr><td>Z. Hocenski et al. [30]</td><td>Halogen light</td><td>• The light intensity does not change fast after being switched.</td><td>•Higher power consumption 150W.
•Sensitivity to mechanical damage.
•Much heat produced.</td></tr>
<tr><td>Z. Hocenski et al. [31]</td><td rowspan="3">LED light</td><td rowspan="3">•Lower power consumption 1.5W.
•Converts electrical energy directly into light.
•Small heat generated.
•Resistance to physical stress and strain.</td><td rowspan="3">•High cost.</td></tr>
<tr><td>B. You et al. [28]</td></tr>
<tr><td>E. Golkar et al. [13]</td></tr>
</table>

2.3 Summary

Overall, the previous related works' reviews are done and summarise in Figure 2.3 and taken as reference for this project. Some comparison are made to explore and investigate the information of quality inspection field in ceramic industry.

From the methods for defect detection, image processing method have high detection speed and accuracy rate compare to other method such as human inspection, sensor and laser. Image processing improved the accuracy and stability of the detection since it only required one time setting after trained and do not need adjustment for different type of tiles.

Canny edge detector will be used to find the edge of the tiles. Due to it can find the edge more accurate in a noise condition. It smoother the image to reduce the noise and find the gradient to eliminate insignificant edges with thresholding.

From the type of tiles' defect, edge crack is commonly happen in the industry and most of the research show it can be detected successfully. Ordinary personal computer is a simple hardware to use for the image processing task since the process to carry out in this project is one in a time. LED light panel have high intensity compared to halogen light with low power. It is economic applied in this project.

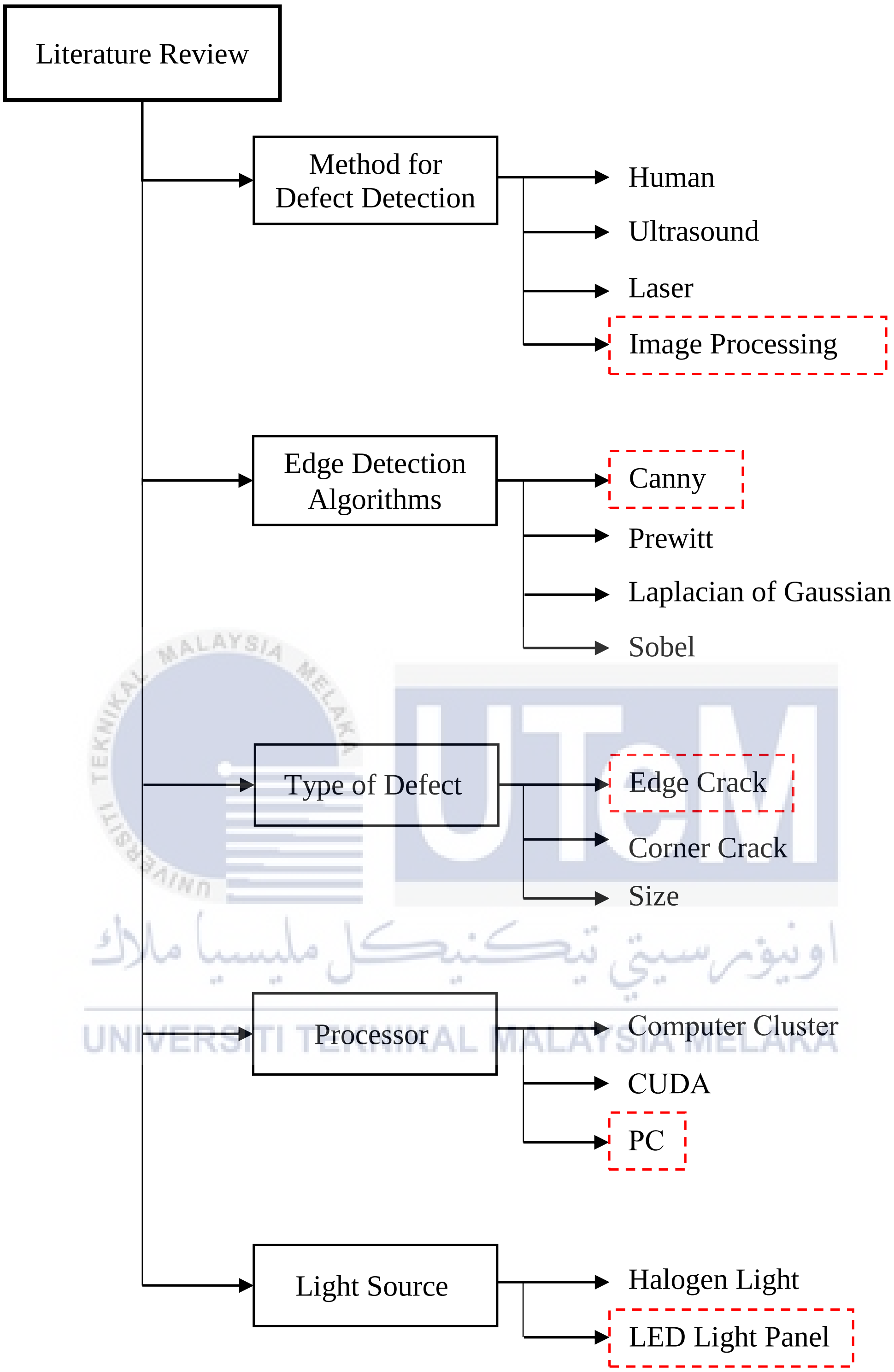


Figure 2.3: Summay of Literature Review

CHAPTER 3

METHODOLOGY

3.1 Project Flowchart

A project flowchart is shown in this section to present the overview of the project. The flowchart explains briefly the processes that are required to be taken to conduct this project from planning, gathering information, fabrication and performance testing. The project flowchart is shown in Figure 3.1.

After confirmed the title, the path planning for inspecting the quality of ceramic tile border is start researched. The literature review based on the title is carried out by searching the problems that encountered on previous related works. The components and techniques are searched for fabrication of hardware and software. The hardware included the apparatus setup while software included the algorithms designed. Adjustment and modification are done for developed the system successfully. After the system is developed, some improvement should be done to obtain a better result. Performance analysis based on the accuracy is carried out after the results obtained. The Gantt chart of this project can be referred to Appendix D in page 70.

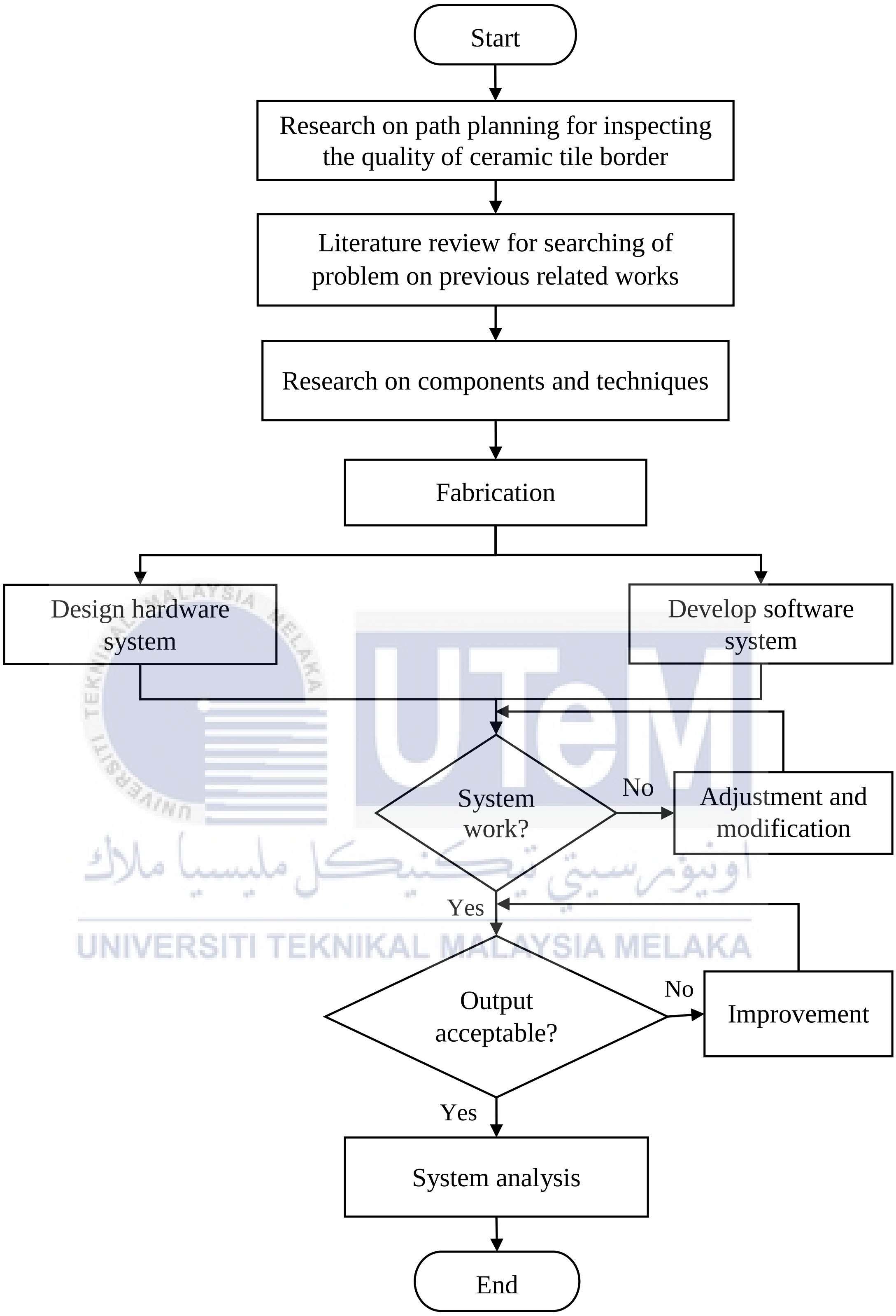


Figure 3.1: Project Flowchart

3.2 Design of System

Overall design criteria for the vision based automated quality inspection system designed and methods for analysing the system performance is exhibited on the flowchart in this section. The details in the flowchart will be discussed later in this chapter. The system design flowchart for this research is shown in Figure 3.2.

The flowchart shows the system design included hardware and software. The hardware will be constructed followed by the software development. Hardware included camera stand, camera and light sources which needed to be setup for the experiment carried out afterward. Software used is MATLAB for developing the algorithms of the system. The camera must be accessed into the software at the beginning of the project for getting the image information.

By getting the image information, the image processing is done to extract the border line. The border line is used for the defect detection purpose. The defect detection algorithms are developed to detect the border line pattern. There are two defect detection algorithms used in this project. The first method applied the background subtraction technique while the second method is based on the gradient variation of the border edge. The adjustment of the light illumination and camera are done until a better results obtained. The analysis on the results are done and the data obtained are recorded in this project.

The next subsections have described about the image processing method used in the system and the algorithms developed for the defect detection.

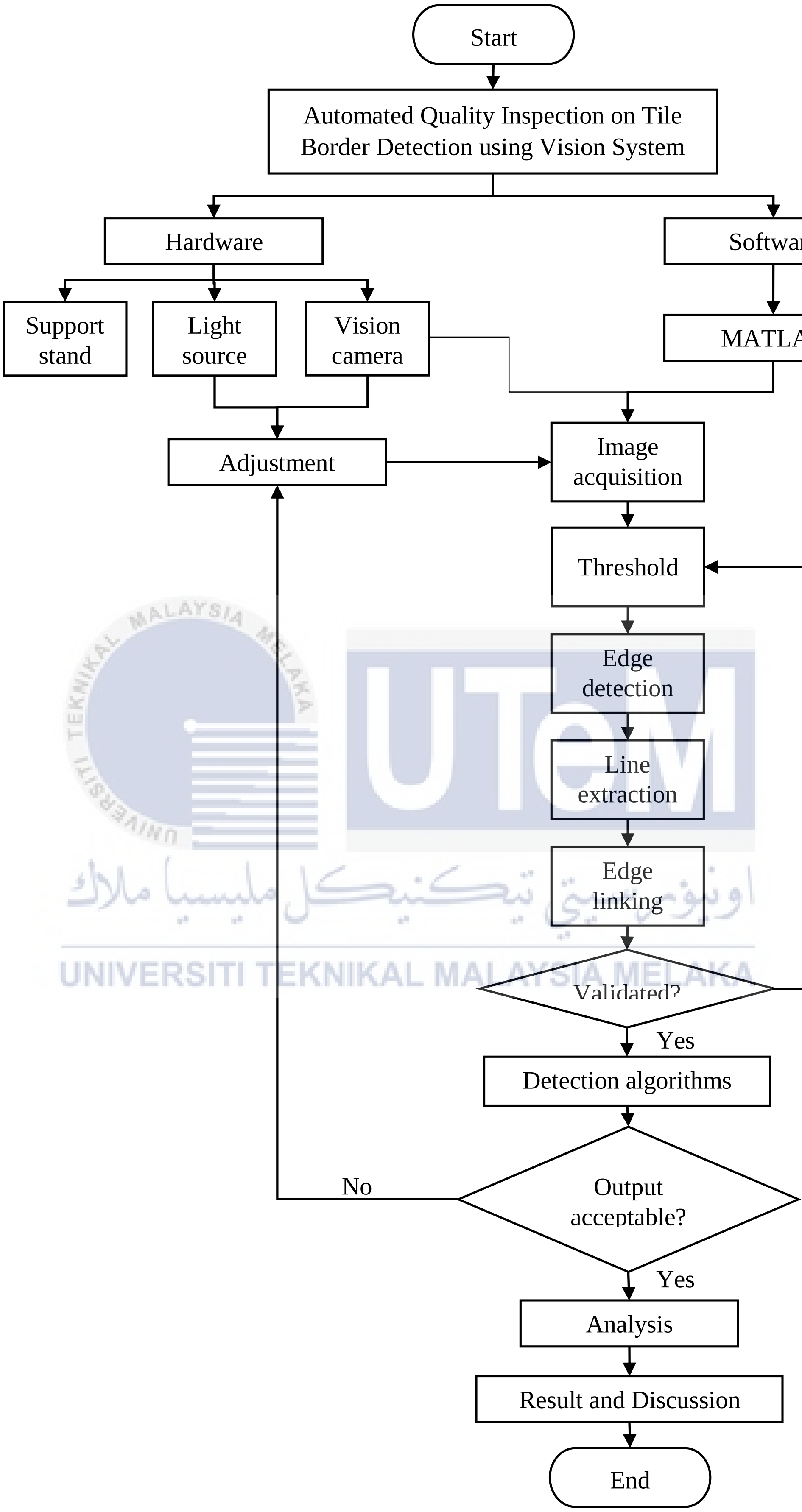


Figure 3.2: System Design Flowchart

3.2.1 Image Processing

The edge of the tile border is obtained during the image processing and the image obtained is for the detection. The image captured from the camera is undergo thresholding to form a binary image for transform the image to a logical data. Canny edge detection is applied on the image to segment the edge. The edge line is extracted and morphological operation is performed on that line for enhancing the border line to get a better result in the defect detection.

3.2.2 Defect Detection Algorithms

There are two algorithms developed to perform the defect detection on ceramic tile border. Background subtraction method is a technique of detection based on modelling the background of the image, setting the background and detecting changes that occur. A reference image should be used to act as a background in this method. The reference image used in this project is a perfect tile border. If a testing image consists any changes with the background, the changes are detected and is evaluated to find the crack. A big changes occur on the image is detected as a crack.

The other method is the detection based on the gradient variation on the edge of the tile border. Graph of the coordinate of the top and bottom edge are plotted first. The polynomial curve fitting method is applied to perform the best fit line on every 20 interval points along the edge. The best fit line on every 20 interval points are drawn using the linear equation. To improve the numerical properties of both the polynomial and the fitting algorithm in the MATLAB, centering and scaling values which are the mean and standard deviation are found to apply the standard normal distribution in the linear line equation. It centers x at zero and scales it to have unit standard deviation. After that, the gradient is obtained from the linear equation and is plotted for observed the variation along the edge. A significant changes on the graph is detected and is considered as a crack if fulfil the requirement set.

3.3 Hardware Description

In this section, the hardware used in the poject are described. The specification of camera, lens and laptop are listed in the subsection.

3.3.1 Industrial Camera

An industrial camera plays an important role in vision inspection system. The camera used to capture the image for processing later. For vision system, it is important for the camera to have sufficient resolution and field of view to ease object identification process. A low resolution camera will not able to catch the details of an object while a small field of view will cause the wrong defect detection. Taking these factors into consideration, DFK72AUC02 USB 2.0 CMOS Industrial Camera from Imaging Source series is selected in this research. The camera is shown in Figure 3.3. It has sufficiently high resolution and wide field of view. It also has high processing speed that allows it to track a moving object easily. The hardware specification of the camera is listed in Table 3.1[32].

Figure 3.3 Imaging Source Series of DFK72AUC02 USB 2.0 CMOS Industrial Camera

Table 3.1: List of Specification for DFK72AUC02 USB 2.0 CMOS Industrial Camera [32]

General Behaviour	
Dynamic range	8 bit
Video formats frame rate (maximum)	2,592×1,944 (5 MP) Y800 @ 6 fps 2,592×1,944 (5 MP) RGB32 @ 6 fps
Interface (Optical)	
IR cut filter	Yes
Sensor type	CMOS
Sensor specification	Aptina MT9P031
Shutter	rolling
Format	$^{1}/_{2.5}$ inch
Pixel size	H: 2.2 µm, V: 2.2 µm
Lens mount	C/CS
Interface (Electrical)	
Interface	USB 2.0 (forwards compatible)
Supply voltage	4.5 VDC to 5.5 VDC
Current consumption	approximate 250 mA or 5 VDC
Interface (Mechanical)	
Dimensions	H: 36 mm, W: 36 mm, L: 25 mm
Mass	70 g
Adjustments	
Shutter	$^{1}/_{10,000}$ s to 30 s
Gain	0 dB to 18 dB
White balance	-6 dB to 6 dB
Environmental	
Temperature (operating)	-5 °C to 45 °C
Temperature (storage)	-20 °C to 60 °C
Humidity (operating)	20 % to 80 % (non-condensing)
Humidity (storage)	20 % to 95 % (non-condensing)

3.3.2 Lens

The lens will be used in this project is HF35HA-1B Fujinon Lens. It supports a high-resolution camera up to 1.5 megapixels and enables a faithful image input. It has compact, lightweight and robust design which able to support environments such as vibration. It can support various systems. The lens also provides the lock tab for fixing the movement of focus and iris. Table 3.2 have listed out the specification for HF35HA-1B Fujinon Lens.

Table 3.2: List of Specification for HF35HA-1B Fujinon Lens

Focal length	35 mm
Focus	Manual
Iris Range	F1.6 - F22
Iris	Manual
Focus Length	∞-0.25m

Figure 3.4: HF35HA-1B Fujinon Lens [33]

3.3.3 Laptop

Laptop is used for developing and executing the MATLAB program. The specification for laptop is listed in Table 3.3.

Table 3.3: List of Specification for Laptop

Model Name	ASUS A555LD-XX685H
Processor	Intel® Core™ i5-5200U CPU @ 2.20GHz
GPU	Intel® HD Graphic 5500, NV GT 820M
Display	15.6 HD(GL,LED)
Memory (RAM)	8.00 GB DDR3
HDD	1TB 5400R SATA
VRAM	2GB DDR3
OS	Windows 8.1

3.4 Software Description

3.4.1 MATLAB

MATLAB which developed by MathWorks is a multi-model numerical computing environment and possessed own programming language [6]. MATLAB R2018b version 9.5 is applied in this research. The MATLAB software environment allows users to develop the image processing algorithms. The software consist many set of the reference algorithms of different type of edge detection technique that can be used to do the analysis of image.

3.4.2 Image Acquisition Toolbox

Image Acquisition Toolbox in MATLAB can be used to access much more features of the Imaging Source industrial cameras which means it support the DFK72AUC02 USB 2.0 CMOS Industrial Camera. Users allow to acquire data directly in MATLAB using Image Acquisition Toolbox shown in Figure 3.5 through the consistent interface with the camera. Then, analysis can be done easier by using these features [34].

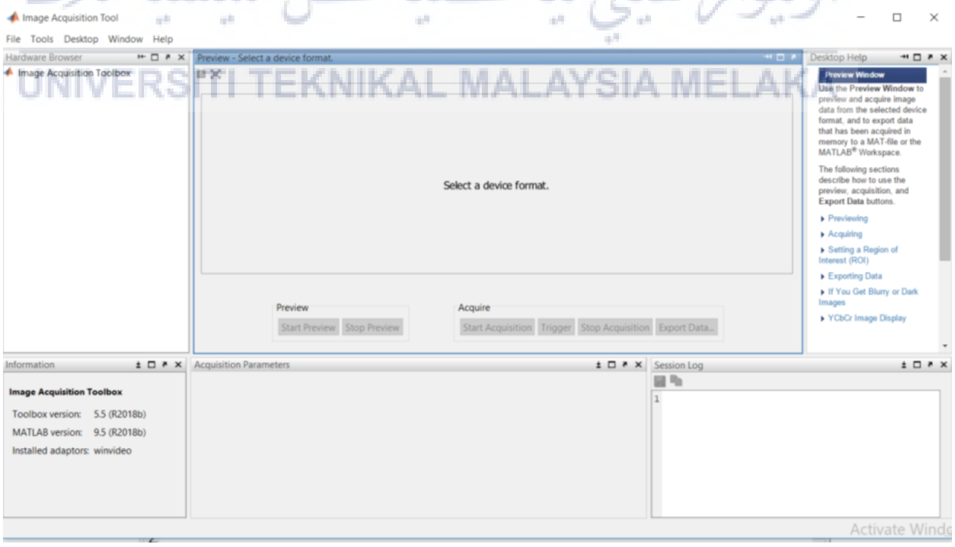


Figure 3.5: MATLAB Image Acquisition Toolbox

3.5 Research Design

The experiments are conducted for fulfilling the objectives of this project. 3 experiments are carried out which involve testing the detection algorithms based on background subtraction method, detection algorithms based on edge gradient variation and analysis of both the defect detection algorithms. Experiment 1 and 2 are fulfilled the objective 1 which the automated quality inspection on tile border using vision system is designed and developed successfully. Objective 2 also fulfilled by experiment 1 and 2 which the characteristic of the cracks pattern are obtained for developing the defect detection method. To fulfil the objective 3, experiment 3 is done in analysing the performance of the tiles border line detection in term of accuracy for both methods. The fulfilment of objectives based on the experiments are shown in Table 3.4.

Table 3.4: Fulfilment of Objectives Based on the Experiments

	Objective 1	Objective 2	Objective 3
Experiment 1	√	√	
Experiment 2	√	√	
Experiment 3			√

3.5.1 Experiment 1: Detection Algorithms Based on Background Subtraction Method

Experiment 1 is conducted to detect the tile border crack area by design a background subtraction algorithms. The experiment is setup as shown in Figure 3.6.

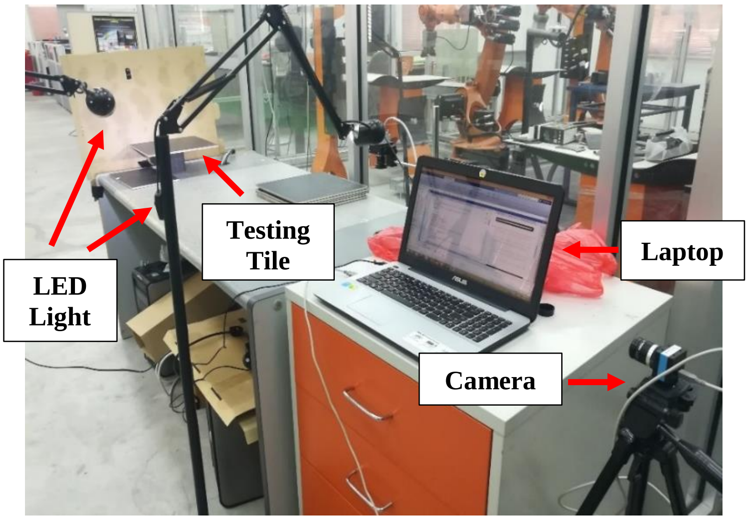


Figure 3.6: Experiment Setup

A laptop is used to run the MATLAB program for implementing the vision system. The industrial camera is interfaced with MATLAB and image is acquired using the image acquisition tool. 1 good tile is taken as a reference and 10 sample tiles include 3 good and 7 crack are taken to detect. A tile is placed in front of the camera. The distance between the tile and camera is 2m. Two LED light is used for illuminating the tile border.

First, a good tile border is taken as a reference for the defect detection afterward. The tile border is acquired and the ROI is found and set as constant. Camera focus and iris as well as light are adjusted to get a clear image. The ROI image is acquired. The image is first change to a binary image. Threshold level is adjusted to obtain a high performance image. Canny edge detection is applied to segment and extract the edge. Threshold level is adjusted to filter the disturbance around the edge. Morphologically dilate operation is performed to link the edge. Then, morphologically filled operation is done to fill the region in the edge. Camera focus and iris as well as light are adjusted again if the edge is not extract nicely. Step from image acquired until morphological operations are repeated. The coordinate of the edge is obtained. The reference image is created. The process is presented in Figure 3.7.

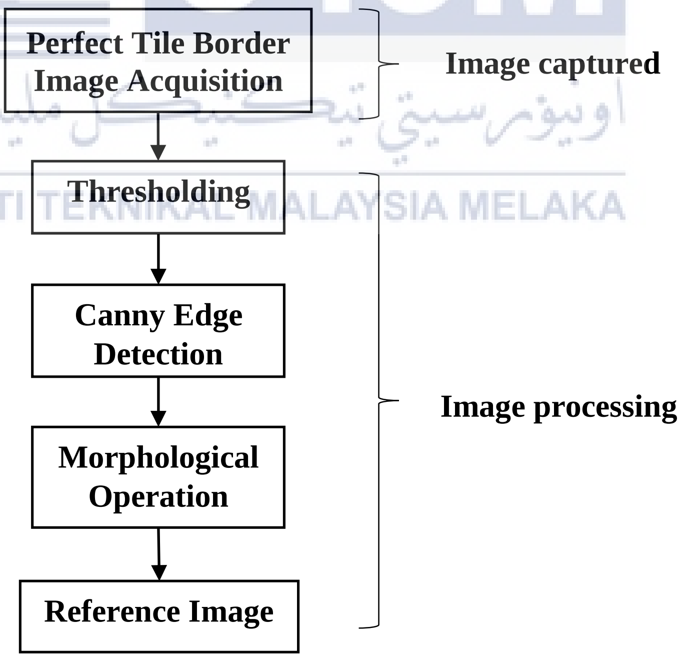


Figure 3.7: Reference Image Flowchart

After the reference image is processed successfully, 10 testing tiles are taken to conduct the defect detection. A testing tile is acquired with the same ROI as reference. The tile is go through the process of image acquired until morphological operation as reference. Detection algorithms is designed to detect the crack by comparing with the reference image as background. After the morphological operation, the coordinate of the edge of the image is obtained and then the image is resized to reference size and translated to move to the reference image position. Both reference and testing tile border image are compared. The differences between the images are obtained. The unused narrow line and tiny dot in the image are eliminated by adjusted the range of the pixels numbers to left only the data which needed to conduct the identification. The crack area is highlighted and the condition is identified. The steps are repeated to test the other 9 testing tiles as presented in Figure 3.8.

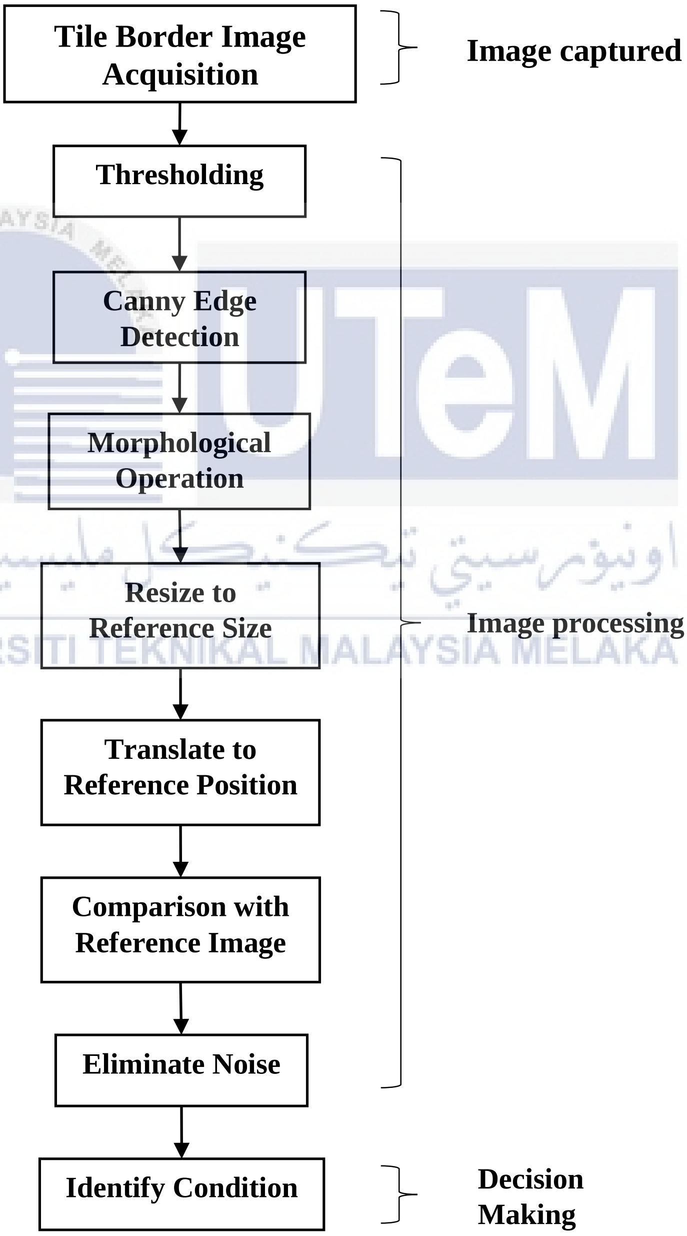


Figure 3.8: Testing Image Flowchart

The system is identified whether the tile is good or defected. A table is drawn to record the data obtained. After that, the accuracy of the defect detection system is analysed.

Accuracy will be calculated as shown in (3.1).

$$Accuracy = \frac{a}{b} \times 100\% \qquad (3\text{-}1)$$

Where, a = Number of correct detection

b = Total number of sample tiles

3.5.1.1 Parameters

The parameters in experiment 1 are listed in Table 3.5. The parameters for hardware, software and the condition applied are presented. The hardware included processor, ceramic tile and camera while software included the MATLAB and some algorithms setting. The noise elimination for the images after comparison is set. The condition such as distance and illumination are also mentioned in the list.

Table 3.5: List of Parameter of Experiment 1

Items		Parameters
Processor	Laptop Core GPU CPU RAM	ASUS A555LD-XX685H Intel® Core™ i5-5200U Intel® HD Graphic 5500 2.20GHz 8.00 GB
MATLAB	Version	9.5
Reference tile		1 piece good 30cmx30cm ceramic tiles
Sample tile		10 pieces 30cmx30cm ceramic tiles (3 good and 7 crack border)
Camera	Format Resolution	RGB32 2592x1944
Distance between camera and sample		2m
Illumination		2 x 3W 220V LED light bulbs
ROI position		[37 1261 2528 226]
Binary image	Threshold	0.80
Canny operator	Threshold	0.999
Noise elimination	Maximum noise area Minor Axis Length	1200 20 pixels

3.5.1.2 Equipment

The equipment used in experiment 1 are listed in Table 3.6. The equipment included the hardware such as laptop, camera, camera stand, ceramic tile and light. For software, MATLAB is used for executing the system. A reference tile is used in this experiment.

Table 3.6: List of Equipment for Experiment 1

No.	Equipment	Quantity
1.	Laptop	1
2.	MATLAB software	-
3.	Industrial camera	1
4.	Camera stand	1
5.	Good reference tile	1
6.	Good tiles samples	3
7.	Crack tiles samples	7
8.	LED light	2

3.5.2 Experiment 2: Detection Algorithms Based On Edge Gradient Variation

Experiment 2 is conducted to detect the tile border defect by developing a detection algorithms based on the variation of edge gradient. The experiment setup is same as experiment 1 which show in Figure 3.6.

First, a tile border image is acquired and the edge detection is processed with the same step as shown in experiment 1. The difference is that no reference sample is required in this experiment. All sample tiles which include 3 good and 7 crack tile are go through the same detection algorithms. The morphologically filled operation is skipped to save processing time and only the edge is obtained for the defect detection.

After the edge detection, the edge coordinate is found by taking all the first and last value 1 pixels along the column of the image pixels for the top and bottom edge respectively. Both top and bottom edge coordinate graphs is plotted. Then, polynomial curve fitting method is applied on the graphs to perform the best fit line on every 20 interval points. The best fit line on every 20 interval points are drawn using the linear equation. To improve the numerical properties of both the polynomial and the fitting algorithm in the MATLAB, centering and scaling values which are the mean and standard deviation are found to apply the standard normal distribution in the linear line equation. It centers x at zero and scales it to have unit standard deviation.

Mean and standard deviation for each of the 20 points in the edge coordinate graph are found by using equation as shown in (3.2) and (3.3) respectively. Standard normal distribution is obtained from equation as shown in (3.4). These values are substituted to the linear equation that exhibit in (3.5) to draw a straight line.

$$\mu = \frac{\sum x}{n} \tag{3-2}$$

$$\sigma = \frac{\sum(x - \mu)}{n - 1} \tag{3-3}$$

$$z = \frac{x - \mu}{\sigma} \tag{3-4}$$

$$y = p_1 z + p_2 \tag{3-5}$$

Where,
μ = mean
x = value of x-axis
n = number of value of x-axis
σ = standard deviation
z = standard normal distribution
y = value of y-axis
p_1 = slope or gradient of the line
p_2 = y-intercept point

The line is drawn on every 20 points interval and shown in a graph. p_1 from equation (3.5) which are the gradient of the lines are taken and plotted in a graph. The mode of the gradient along the points is found and the range of a good edge's slope is set not more than 1.5 of the mode. The coordinate of the gradient that exceed the range are labelled and shown in a graph. The gradient that out of the range are grouped according to the consecutive points. The groups that more than 11 consecutive value are considered a crack occur at the area. The consecutive values are represent the pixel values of the image. Therefore, the gradients out of range that have the consecutive pixel values more than 11 are considered a crack is occur on the image. The crack area is circled in the tile border image. The flowchart of the process of this algorithms is shown in Figure 3.9.

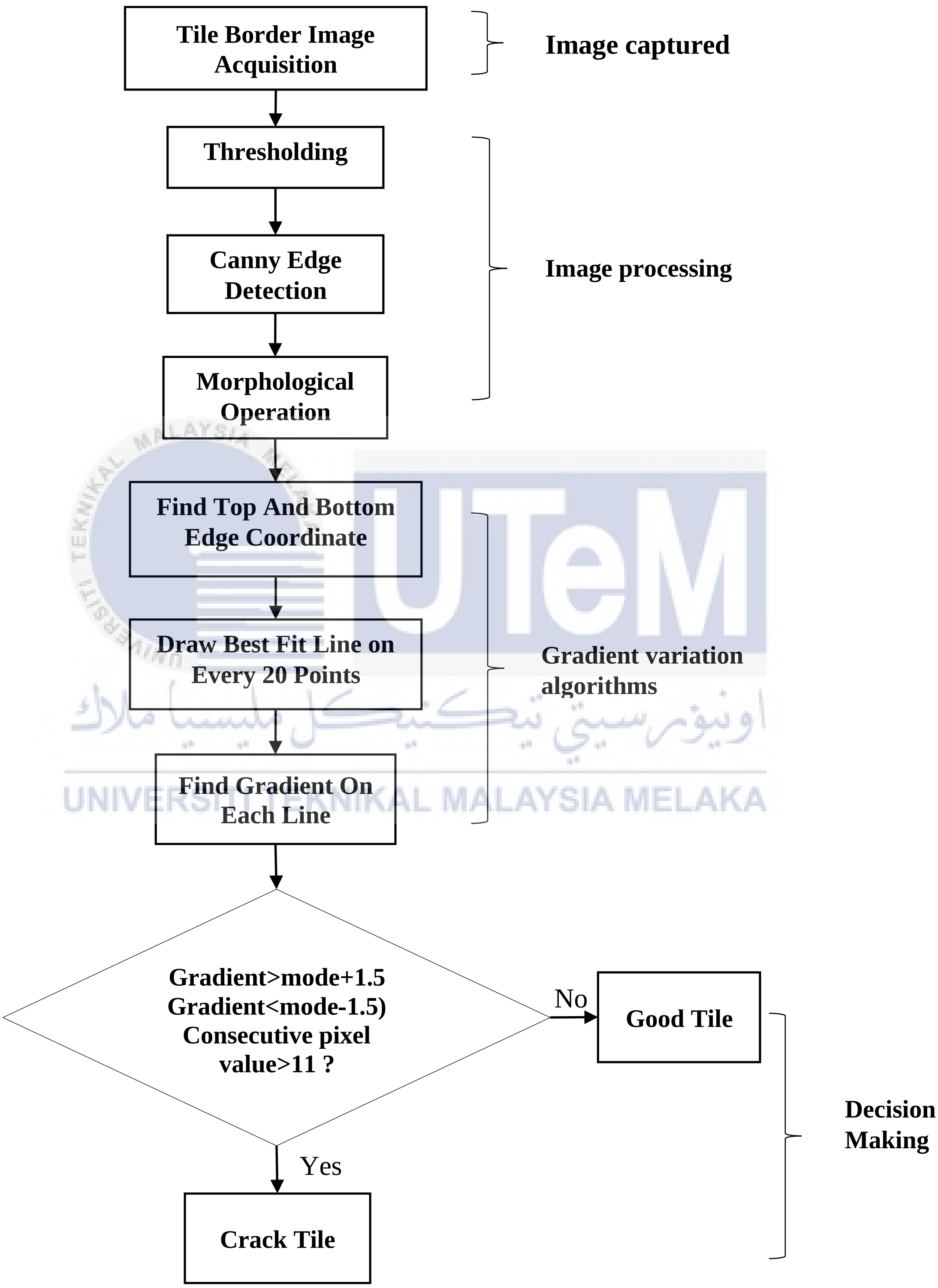


Figure 3.9 Detection Algorithms Based on Edge Gradient Variation Flowchart

The system is identified whether the tile is good or defected. A table is drawn to record the data obtained. After that, the accuracy of the defect detection system is analysed. Accuracy will be calculated as shown in (3.1).

3.5.2.1 Parameters

The parameters in experiment 2 are listed in Table 3.7. The parameters for hardware, software and the condition applied are presented. The hardware included processor, ceramic tile and camera while software included the MATLAB and some algorithms setting. The range of crack have set in this experiment. The condition such as distance and illumination are also mentioned in the list.

Table 3.7: List of Parameter of Experiment 2

Items		Parameters
Processor	Laptop Core GPU CPU RAM	ASUS A555LD-XX685H Intel® Core™ i5-5200U Intel® HD Graphic 5500 2.20GHz 8.00 GB
MATLAB	Version	9.5
Sample		10 pieces 30cmx30cm ceramic tiles (3 good 7 crack border)
Camera	Format Resolution	RGB32 2592x1944
Distance between camera and sample		2m
Illumination		2 x 3W 220V LED light bulbs
ROI position		[37 1261 2528 226]
Binary image	Threshold	0.80
Canny operator	Threshold	0.999
Range of crack	Gradient of edge Width	> mode +1.5 or < mode -1.5 >11 consecutive pixels

3.5.2.2 Equipment

The equipment used in experiment 2 are listed in Table 3.8. The equipment included the hardware such as laptop, camera, camera stand, ceramic tile and light. For software, MATLAB is used for executing the system. A reference tile is not required in this experiment.

Table 3.8: List of Equipment for Experiment 2

No.	Equipment	Quantity
1.	Laptop	1
2.	MATLAB software	-
3.	Industrial camera	1
4.	Camera stand	1
5.	Good Tiles samples	3
6.	Crack Tiles samples	7
7.	LED light	2

3.5.3 Experiment 3: Analysis of Both the Defect Detection Algorithms

From the experiment 1 and experiment 2, the tile border cracks are detected. In this experiment, the accuracy of both system are compared. Although both system can detect the cracks, the result obtained from them have some different. Since the sample used are the same, the performance in term of defect detection accuracy are compared and the ability to detect the type and size of the crack are determined. The data are recorded in the table.

3.5.3.1 Parameters

The parameters in experiment 3 are listed in Table 3.9. The parameters are the combination of the previous experiments. The parameters for hardware, software and the condition applied are presented. The hardware included processor, ceramic tile and camera while software included the MATLAB and some algorithms setting. The condition such as distance and illumination are also mentioned in the list.

Table 3.9: List of Parameter of Experiment 3

Items		Parameters
Processor	Laptop Core GPU CPU RAM	ASUS A555LD-XX685H Intel® Core™ i5-5200U Intel® HD Graphic 5500 2.20GHz 8.00 GB
MATLAB	Version	9.5
Reference tile		1 piece good 30cmx30cm ceramic tiles
Sample		10 pieces 30cmx30cm ceramic tiles (3 good 7 crack border)
Camera	Format Resolution	RGB32 2592x1944
Distance between camera and sample		2m
Illumination		2 x 3W 220V LED light bulbs
ROI position		[37 1261 2528 226]
Binary image	Threshold	0.80
Canny operator	Threshold	0.999
Noise elimination	Maximum noise area Minor Axis Length	1200 20 pixels
Range of crack	Gradient of edge Width	> mode+1.5 or < mode-1.5 >11 consecutive pixels

3.5.3.2 Equipment

The equipment used in experiment 3 are listed in Table 3.10. The equipment included the hardware such as laptop, camera, camera stand, ceramic tile and light. For software, MATLAB is used for executing the system. This is the combination of the previous experiments.

Table 3.10: List of Equipment for Experiment 3

No.	Equipment	Quantity
1.	Laptop	1
2.	MATLAB software	-
3.	Industrial camera	1
4.	Camera stand	1
5.	Good reference tile	1
6.	Good tiles samples	3
7.	Crack tiles samples	7
8.	LED light	2

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Detection Algorithms Based on Background Subtraction Method

This section is to show the results obtained from the detection algorithms based on background subtraction method. The detailed explanations of the process to obtain the results are presented in subsection. The MATLAB coding or algorithms are developed for conducting this experiment and can be referred to Appendix A in page 58.

4.1.1 Reference Tile Border Image

A perfect tile border is taken as a reference. The region of interest (ROI) of the image is acquired as shown in Figure 4.1.

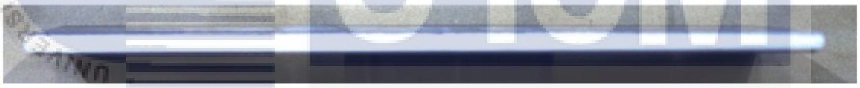


Figure 4.1: Perfect Tile Border

Image processing is done on the image. The binary image with threshold level 0.80 is shown in Figure 4.2.

Figure 4.2: Binary Image of the Tile Border

After the Canny edge detection with threshold level 0.999, the image is obtained as shown in Figure 4.3.

Figure 4.3: Edge Segmentation using Canny Method

The edge is linked by apply the morphologically dilate operation. The output is shown in Figure 4.4. The edge is smoother for enhance the image.

Figure 4.4: Edge Linking after Morphologically Dilate Operation

The edge is filled by perform the morphologically filled operation as shown in Figure 4.5. The reference image is created to act as the background in the defect detection system using background subtraction method.

Figure 4.5: Reference Image

4.1.2 Defect Detection

The tile border defect detection system is conduct on 10 samples. The output of a good and a crack sample tile are presented in this section. Sample tile A which is a good tile and D which is a crack tile are taken to present the results. Figure 4.6 and Figure 4.7 have shown the image of the tile border of good and crack tile.

Figure 4.6: Good Tile Sample

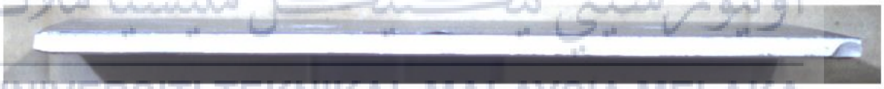


Figure 4.7: Crack Tile Sample

The samples undergo the same edge detection process as the process to obtain the reference image. The good and crack tile image is obtained and shown in Figure 4.8 and 4.9.

Figure 4.8: Good Tile Border Image

Figure 4.9: Crack Tile Border Image

The sample image is resized and translated to move to the reference image position for facilitating and reduce the error of the comparison afterward. After the dimension of both reference and sample image is adjusted to be the same, the both image is compared. The outputs are shown in Figure 4.10 and 4.11. The colour line on the image is the differences with the background image.

Figure 4.10: Comparison between Reference and Good Tile

Figure 4.11: Comparison between Reference and Crack Tile

The differences between the images are obtained. The output of good tile is shown in Figure 4.12 while crack tile is shown in Figure 4.13.

Figure 4.12: Differences between Reference and Good Tile

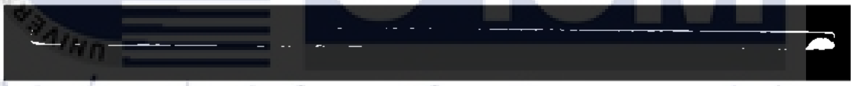


Figure 4.13: Differences between Reference and Crack Tile

It can be seen that is some narrow line and disturbance occur on the image. The crack is presented by a large area difference that shown at the right bottom side in Figure 4.13. After eliminate the unused line and dot, the image left only the data which needed to conduct the identification. The outputs are shown in Figure 4.14.

Figure 4.14: Differences between Reference and Good Tile after Noise Elimination

Figure 4.15: Differences between Reference and Crack Tile after Noise Elimination

The accuracy of the defect detection obtained high after remove the noise. The tile condition is identified. The good testing tile is not detected any defect while the crack area is highlighted in pink on the crack testing tile. Figure 4.16 and 4.17 are presented the detection result.

Figure 4.16: No defect is Detected

Figure 4.17: Defect is Detected

The output of all the sample tiles are recorded in Table 4.1.

Table 4.1: Table of Detection Result of Experiment 1

Sample Tile	Real Condition	Detection Result	True/False
A	Good	Good	True
B	Good	Good	True
C	Good	Good	True
D	Defect	Defect	True
E	Defect	Good	False
F	Defect	Good	False
G	Defect	Defect	True
H	Defect	Good	False
I	Defect	Good	False
J	Defect	Good	False

Some tiles' defect are not detected due to the size is small and be removed in the process of noise elimination. The size of the cracks are in the range of the noise as shown in parameter table. For example, the sample tile E has a small crack located at the left bottom side of the border as circled. Before the noise elimination process, a thicker line area is shown on left side of Figure 4.18 which is the area that a crack occurred.

Figure 4.18: Differences between Reference and Crack Tile E before Noise Elimination

After process the noise elimination, all of the lines and dots are eliminated as shown in Figure 4.19. The area of the crack is also removed since its area and minor axis length are small enough in the range of noise and be considered as a part of noise in the system.

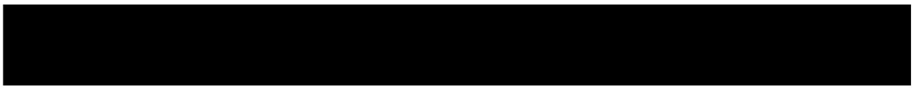


Figure 4.19: Differences between Reference and Crack Tile E after Noise Elimination

The corner crack also cannot be detected as the crack is not obvious by comparing between two images. This is because the crack is also white in colour and same pixels value with the background reference image. This caused the differences between them is not large. Thus, it will be mistaken as good. For example, sample tile F has a crack at the right corner of the border. After the edge detection and morphological filled operation which shown in Figure 4.20, the right corner side can be seen that is almost same with a good tile border and no much difference with the reference image. Therefore, the crack cannot be detected well.

Figure 4.20: Sample Tile F image after Edge Detection and Morphological Filled Operation

Besides, the other crack that always be ignored is the crack occurred at the top of the border. This is because the top crack is presented in white after the edge detection. It also same concept with the failure in detect corner crack. For instance, sample tile H has a crack on top of the border. Colour of white caused it has be mistaken as a normal edge. Figure 4.21 has shown the image of sample tile H and the area where the crack should be occurred is shown. It is ignored by the system which considered it as a normal border edge.

Figure 4.21: Sample Tile H image after Edge Detection and Morphological Filled Operation

The accuracy of the defect detection for good tile, defect tile and overall system are calculated using equations shown in (3.1) and recorded in Table 4.2.

Table 4.2: Table of Accuracy of Experiment 1

	Good	Defect	Total
Number of tested tiles, b	3	7	10
Number of defected tiles obtained	0	2	2
Number of correct detection, a	3	2	5
Number of wrong detection	0	5	5
Accuracy (%)	100	28.57	50

From the result of detection shown in Table 4.2, the good tile is always in correct detection while the defect tile is only obtained 28.57% accurate in the detection which 2 out of 7 testing tiles are in correct detection. Overall, the accuracy of the system is 50%.

4.2 Detection Algorithms Based on Edge Gradient Variation

The MATLAB coding or algorithms are developed for conducting this experiment and can be referred to Appendix B in page 62.

A good and crack sample tiles' results are presented in this section. The tiles acquired are undergo the edge detection to extract the tile border edge. The sample tile used as shown in Figure 4.22 and 4.23. The edge obtained of the border tiles are shown in Figure 4.24 and 4.25.

Figure 4.22: Good Tile Sample

Figure 4.23: Crack Tile Sample

Figure 4.24: Good Tile Border Edge

Figure 4.25: Crack Tile Border Edge

The top and bottom edge coordinate graph are plotted as shown in Figure 4.26 and 4.27. The x-axis of the graph is the number of pixel column along the image while y-axis show the first and last pixels value 1 along each column of the image for obtain the top and bottom edge coordinate respectively. From both of the good testing tile edge coordinate, the line is almost constant across the graph. From the top edge coordinate of the crack testing tile, the variation of the slope is considered constant while the bottom edge coordinate have a big variation of slope at the end of the graph.

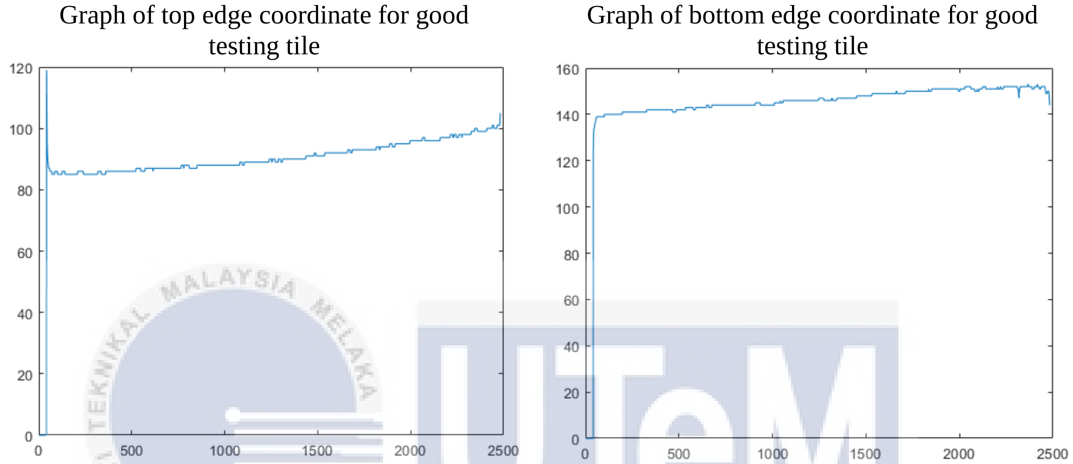


Figure 4.26: Graph of Edge Coordinate for the Good Testing Tile Border

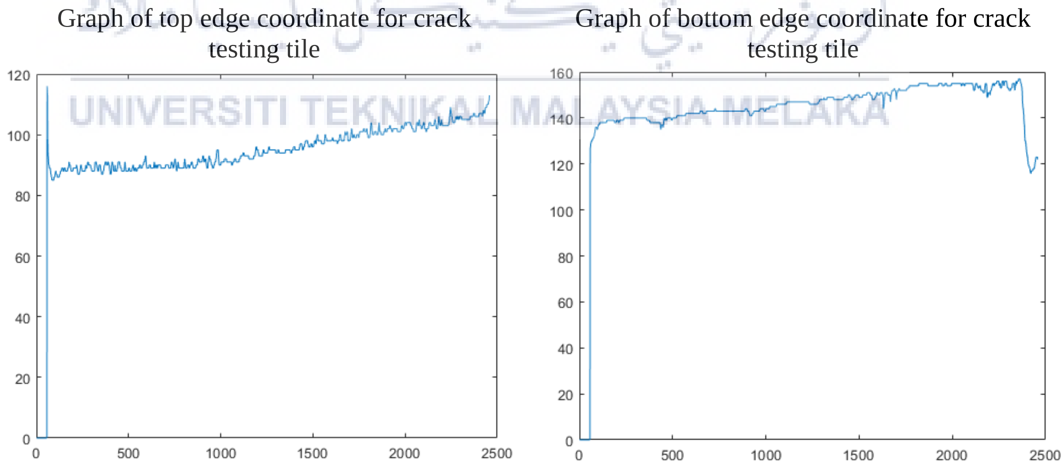


Figure 4.27: Graph of Edge Coordinate for the Crack Testing Tile Border

The best fit lines are obtained from the equation shown in 3.2 to 3.5. The lines are drawn on every 20 intervals of each point in the edge coordinate graph as shown in Figure 4.28 and 4.29.

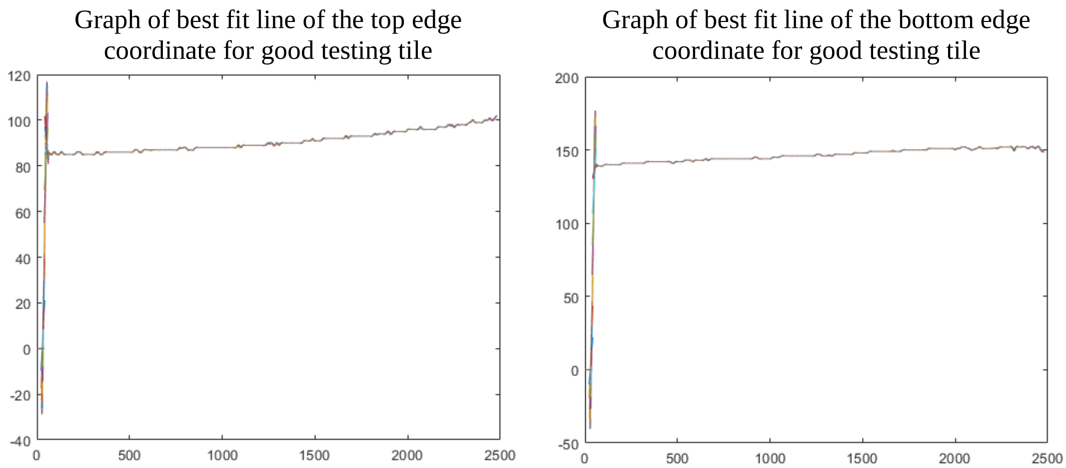


Figure 4.28: Graph Of Best Fit Line of the Edge Coordinate for the Good Testing Tile

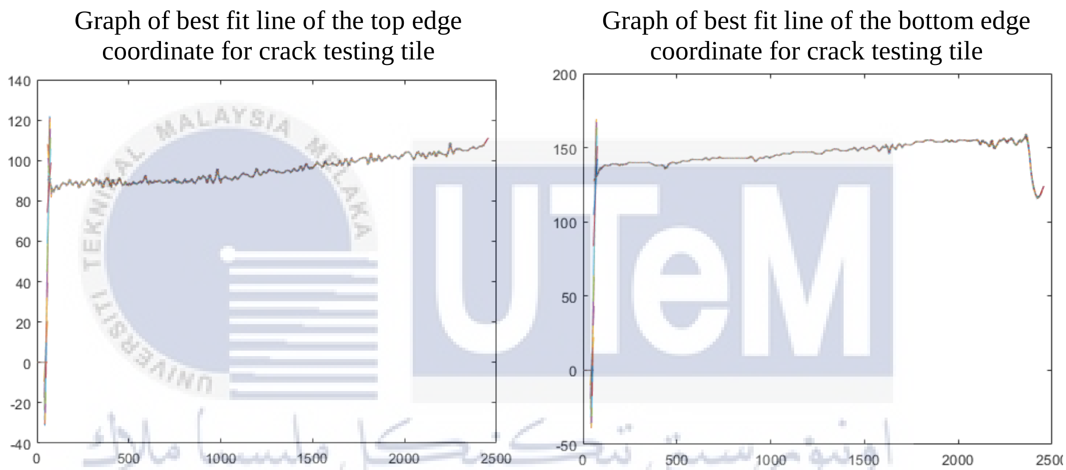


Figure 4.29: Graph of Best Fit Line of the Edge Coordinate for the Crack Testing Tile

From the line equation, the gradients of each line are obtained. The graph of gradient variation for every 20 points in top and bottom edge are plotted as shown in Figure 4.30 and 4.31. The lines of the range are drawn on the gradient graph. It is facilitated to observe the gradient that out of the range. From the good testing tile, the gradient are inside the range. While the crack testing tile, there are some points that out of the range. The gradients at the starting points are always high and out of the range. This is because the corner of the tile have detected like a curve since the edge detection is not extract the edge in 90 degree perfectly. These gradients are become higher due to the curve and are ignored although they are out of the range.

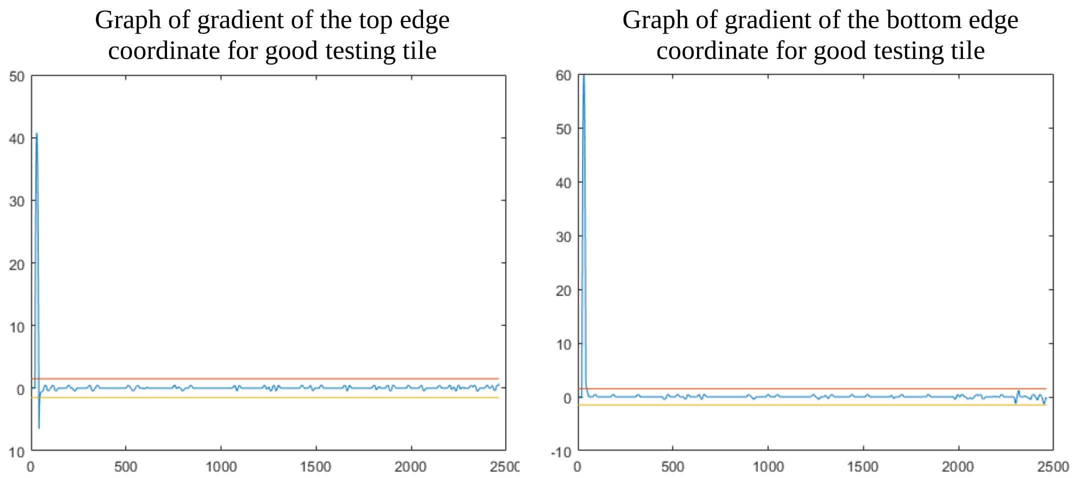


Figure 4.30: Graph of Gradient of the Edge Coordinate for the Good Testing Tile

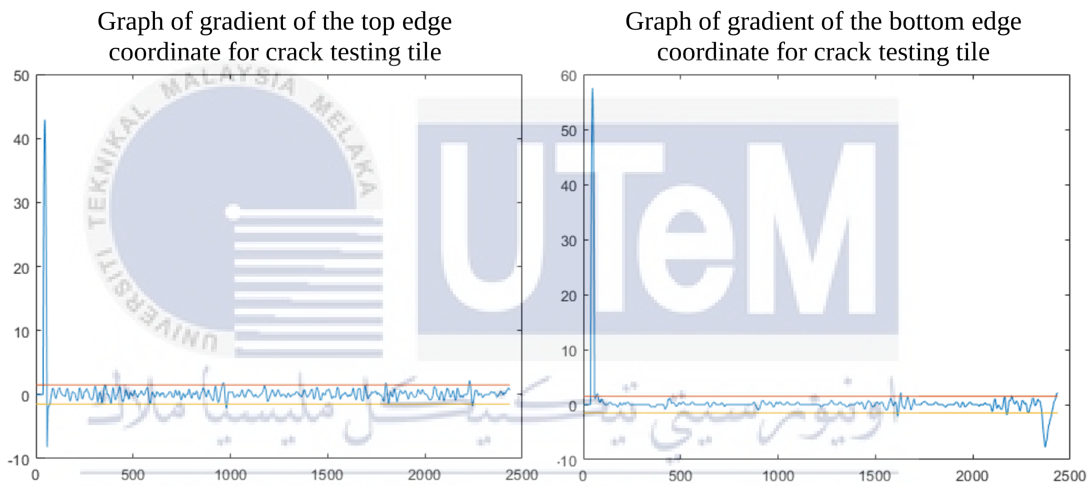


Figure 4.31: Graph of Gradient of the Edge Coordinate for the Crack Testing Tile

The points that are out of range are labelled as red dot at the edge coordinate graph as shown in Figure 4.32 and 4.33. The below is the gradient graph for easily to view.

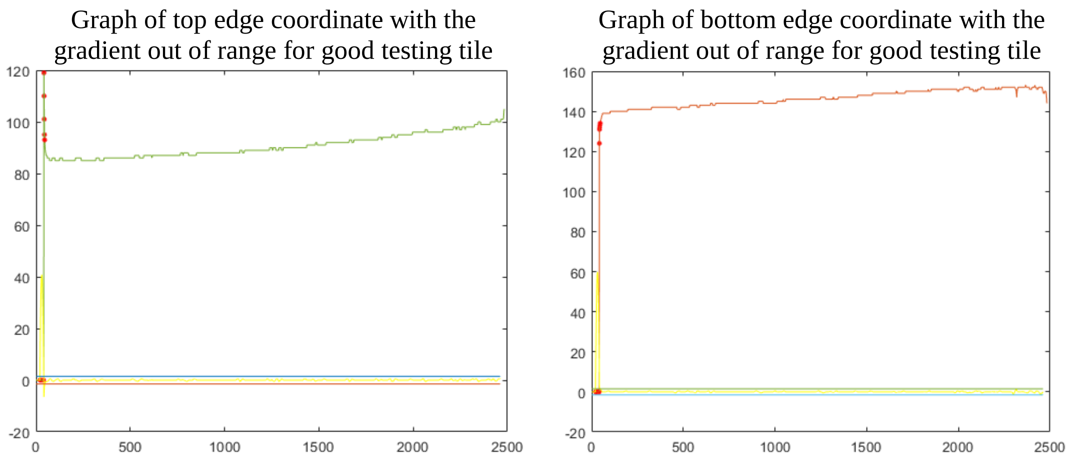


Figure 4.32: Graph of Edge Coordinate with the Gradient Out of Range for Good Testing Tile

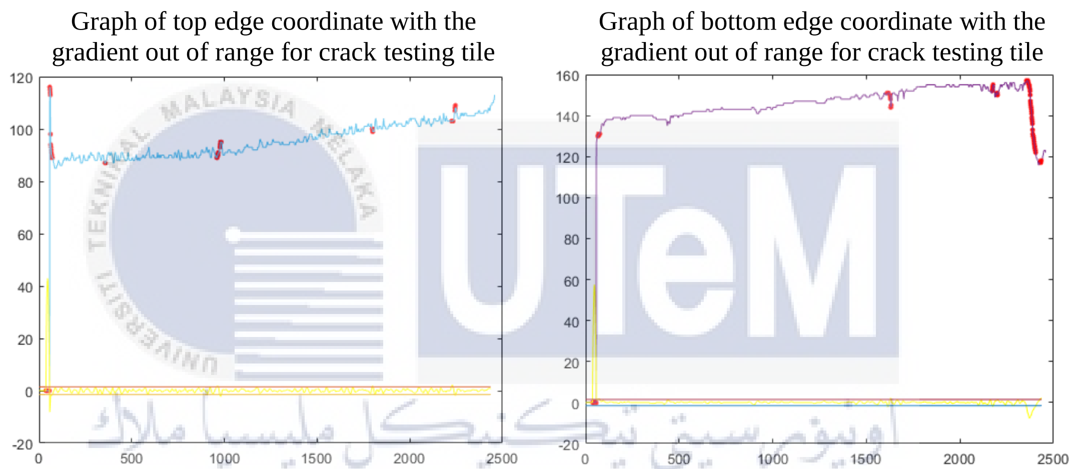


Figure 4.33: Graph of Edge Coordinate with the Gradient Out of Range for Crack Testing Tile

By filtered the gradients that consecutive points are less than 11, the area of the points are circled in the tile border image and are shown. In Figure 4.34 and 4.35. The circled area is the crack area on the tile border. There is no circle on the tile image represent that it is a good tile.

Figure 4.34: Output of Good Testing Tile

Figure 4.35: Output of Defect Testing Tile

The output data of all the sample tile are recorded and the table is shown in Table 4.3. Based on the defect detection program in MATLAB, the mode of gradient of all the testing tiles are found out which is always 0 along the top and bottom edge. Thus, the range of no defects for the gradients are less than 1.5 and more than -1.5 and vice versa.

Table 4.3: Table of Detection Result of Experiment 2

Tile	Real Condition	Gradient out of range (gradient > 1.5) (gradient < -1.5)			No. of consecutive pixels(coordinate)	No. of crack found	True/ False
A	Good	Top	:	-	-	0	True
		Bottom	:	-	-		
B	Good	Top	:	1.60 to 4.09	13 (2416 to 2428)	1	False
		Bottom	:	-	-		
C	Good	Top	:	-	-	0	True
		Bottom	:	-	-		
D	Defect	Top	:	-7.70 to -1.61	14 (52 to 65)	2	True
		Bottom	:	-1.60 to -8.15	48 (2365 to 2412)		
E	Defect	Top	:	-	-	1	True
		Bottom	:	1.61 to 2.34	14 (224 to 237)		
F	Defect	Top	:	-	-	1	True
		Bottom	:	-3.61 to -1.58	16 (2451 to 2466)		
G	Defect	Top	:	-3.74 to -1.53	18 (823 to 840)	5	True
				1.58 to 3.39	18 (959 to 976)		
		Bottom	:	-5.75 to -1.52	30 (1256 to 1285)		
				1.56 to 2.75	24 (1311 to 1334)		
				1.64 to 2.93	16 (1365 to 1380)		
H	Defect	Top	:	1.57 to 2.80	13 (1064 to 1076)	4	True
				-2.51 to -1.64	12 (1162 to 1173)		
				1.77 to 2.77	13 (2231 to 2243)		
		Bottom	:	1.51 to 9.31	16 (106 to 121)		
I	Defect	Top	:	-	-	0	False
		Bottom	:	-	-		
J	Defect	Top	:	1.61 to 12.64	12 (2465 to 2476)	2	True
		Bottom	:	1.68 to 3.33	17 (123 to 139)		

The accuracy of the defect detection for good tile, defect tile and overall system are calculated using equations shown in (3.1) and recorded in Table 4.4.

Table 4.4: Table of Accuracy of Experiment 2

	Good	Defect	Total
Number of tested tiles, b	3	7	10
Number of defected tiles obtained	1	6	7
Number of correct detection, a	2	6	8
Number of wrong detection	1	1	2
Accuracy (%)	66.67	85.71	80

From the result obtained, one of the factor that caused wrong detection is due to the surface stain on the border. During the edge detection, the stain is considered as cracks. For example, the good tile B which shown in Figure 4.36 is contain some stain at the right side of the border. It has mistaken as a defect tile since the stain has taken as a crack by the system. This issue also occurred in the crack tiles and caused more cracks are found.

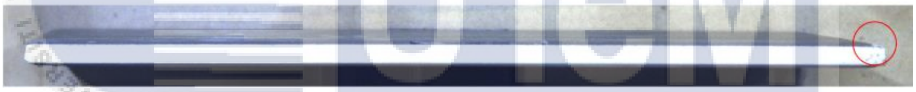


Figure 4.36: Good Sample Tile B Detection Result

Besides the surface stain, some of the cracks found are more than the number of crack that has on the border due to the gradients variation along the axis where the cracks are occurred exceed the positive range and negative range. The system considered the points are different area. Figure 4.37 shows that the system has circled 5 area of cracks on the tile border. There are only 2 cracks on the border in real.

Figure 4.37: Sample Tile G with 5 Circled

A crack tile is detected as a good tile from the results obtained. The wrong detected crack tile is acquired and shown in Figure 4.38. It has a crack on right corner of the border. It can be seen that the crack shows in the image acquired is same intensity with the neighbouring normal edge. This is because of the external light source from top which are the ceiling light.

Figure 4.38: Sample Tile I with Right Border Corner Crack

During the edge detection, the edge extracted is smooth and almost the same coordinate with the good condition edge. Thus, the gradients on those points are not exceeded the range of defect. Figure 4.39 shows the edge detection result and the circled area is represent the crack area that has not be detected by the system.

Figure 4.39: Edge of Sample Tile I

4.3 Analysis of Accuracy for Both Defect Detection Algorithms

In this section, the detection result of both system are compared and analysed. The tiles used for detecting are the same between both of the system but the result is not the same. Table 4.5 has shown the results of both experiments. All of the image of sample tiles and the output can be referred to Appendix C in page 70.

Table 4.5: Table of Detection Results of Each Testing Tile in Experiment 1 and 2

Sample Tile	Real Condition	Detection result		Differences occur
		Method 1	Method 2	
A	Good	Good	Good	-
B	Good	Good	Defect	√
C	Good	Good	Good	-
D	Defect	Defect	Defect	-
E	Defect	Good	Defect	√
F	Defect	Good	Defect	√
G	Defect	Defect	Defect	-
H	Defect	Good	Defect	√
I	Defect	Good	Good	-
J	Defect	Good	Defect	√

From the results show in Table 4.5, a good tile is detected as a crack tile using method 2 which is the detection based on the edge gradient variation while the tile is detected correctly in method 1 which is the background subtraction method. This is due to the surface stain on the tile border has interfered the detection of the system using method 2. During the edge detection, the stain is considered as cracks since the stain on the border line caused the discontinuities in brightness. The output of the edge detection of a sample tile with stain on right side of the border is shown in Figure 4.40. The gradient become higher due to the curve line. This issue also occurred in the crack tiles and caused more cracks are found. This is not an issue in method 1 because the small curve line difference can be removed during the noise elimination process.

Figure 4.40: Edge of Sample Tile B

Most of the defect tile has detected as a good tile using method 1 compared to the method 2. In method 1, most of the tile detected wrongly due to the small cracks unable recognised by the system. The crack is small and has removed in the process of noise elimination. The crack area is in the range of the noise area and minor axis length. For example of testing tile E, there is no defect found in the first detection method while a small crack is on the left bottom of the tile. The detection results are shown in Figure 4.41 and 4.42. This is due to the method 1 has a limitation on detect the crack area less than 1200 and minor axis length less than 20 pixels. The small crack is able to detect by applying the second method. The crack is detected by evaluate the slope of the border edge. If the gradient of the edge is higher and occur consecutively on an area of the edge, a crack may detected on that area. It is filtered by taking only the gradient beyond the range that set as good tile and number of consecutive points.

Figure 4.41: Tile E Result Using Method 1

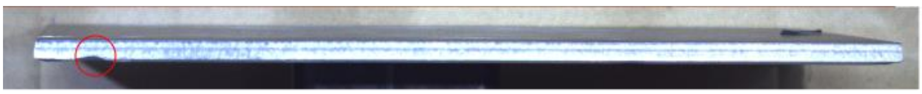


Figure 4.42: Tile E Result Using Method 2

Method 1 has the limitation on detect the corner crack. For example, testing tile F, the crack cannot be found by applying method 1. A crack is found on right of the corner by applying method 2. Both of the results are shown in Figure 4.43 and

4.44. The crack is not obvious by comparing between two images. Since the crack is also white in colour and same pixels value with the background reference image. Thus, it will be mistaken as good. It can be detected by evaluating the gradient variation of the edge since the crack make curve that obtained higher gradient. However, if the crack is not curve enough, it is also failed detected using method 2 since the gradient is lower and considered in the good range.

Figure 4.43: Tile F Result Using Method 1

Figure 4.44: Tile F Result Using Method 2

Although both methods are given a correct result for testing tile G, method 1 only found 1 out of the 2 crack on the tile border. The crack is found on bottom of the tile while the top crack is unable to detect as shown in Figure 4.45. This may cause by the small area crack and also the top crack is white. Small area caused it to be eliminated while colour of white caused it has mistaken as a normal edge. Both top and bottom edge can be detected by method 2 but it has circled 5 areas of crack in the image as shown in Figure 4.46. This is due to the gradients variation along the axis where the cracks are occurred exceed the positive range and negative range. The system considered the points are located in different area.

Figure 4.45: Tile F Result Using Method 1

Figure 4.46: Tile F Result Using Method 2

Overall, the method 2 has higher performance than method 1 since the results of the defect detection using method 2 is more accurate. Method 1 has the limitation in detect small crack, top crack and corner crack. The small crack will be eliminated as recognised as a noise. The crack on top always white since the external light sources are presented on top which are the ceiling light. The colour of white affect the detection when compared with reference image. The corner crack also the same concept as the colour of white at the crack affect the crack not obvious when comparing between testing tile and reference tile. For method 2, the surface stain and the crack corner with low gradient are the factor of failure in detection.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

In conclusion, the objectives of this project are achieved through the experiments carried out. The vision system that automated conduct the quality inspection on ceramic tile border has been developed. The system applied the defect detection algorithms based on the background subtraction method and edge gradient variation are designed and developed successfully. The defect detection algorithms based on the background subtraction method is carried out in experiment 1. The accuracy of this method is achieved 50% in the detection of 10 good and crack sample tiles. The defect detection algorithms based on the edge gradient variation is developed and tested in experiment 2. The system achieved 80% of accuracy in detect 10 different condition of tiles.

The characteristic of crack patterns are obtained based on the tile border line. From the detection using background subtraction method, the cracks found are represented by the large area of differences with the reference image. The small area is not considered as a crack and is filtered. The gradients variation that dropped or rose significantly along the edge with group of consecutive points are considered as the cracks' area occur on those points.

By carrying out experiment 3, the performance of the defect detection system in vision based automated quality inspection for tiles border in term of accuracy is analysed. The results of two methods which conduct in experiment 1 and 2 are compared and analysed. The first method contains many limitations such as unable to determine the small cracks, top edge and corner edge. In second method, it is only failed in detect the unclean tiles and crack that with lower gradient. The first method has ability in removed the unwanted lines or dots in an image. An unclean tile with stain on the border can be ignored by using this method. Overall, the second system is higher performance than the first method.

5.2 Future Works

For future improvements, a consistent working environment and position set up should be considered to avoid the error in detection. A consistent workspace such as a production line can be achieved and reduce the error. The good and defect tiles will be classified and divided to different place by design a conveyer sorting system. The relation of the occurrence of defect tiles can be investigate from all of the data recorded in database. An IoT system can be developed for tracking the data of the product to do various type of analysis. The both methods applied in this project can be combined by taking each ability of them to improve the accuracy of detection. To detect four side of the border, it will consume four times of the processing time. It can be enhanced by using four cameras at the same time to detect one complete tile at once. One more camera can be added on the top to detect the surface defect. The space to do the inspection can be reduced by using a wide angle camera. The angle of view increased, the distance between the camera and the testing sample can be reduced. The time consumed for the tile border defect detection of each edge is long processing by a laptop. It can be faster by applying a high performance processor. The accuracy can be improved by testing more sample and type of defect. The other defect type is possible to detect by using the MATLAB environment and some further research should be done for developing the algorithms.

.

REFERENCES

[1] "Quality Control In Manufacturing | Graphic Products." [Online]. Available: https://www.graphicproducts.com/articles/quality-control-in-manufacturing/. [Accessed: 15-Nov-2018].

[2] L. Blau, P., Martin, R., & Riester, "A comparison of several surface finish measurement methods as applied to ground ceramic and metal surfaces," United States, 1996.

[3] R. Kiran, H. J. Amarendra, and S. Lingappa, "Vision system in quality control automation," vol. 03008, no. 144, pp. 1–12, 2018.

[4] S. Vasilic and Z. Hocenski, "The edge detecting methods in ceramic tiles defects detection," *IEEE Int. Symp. Ind. Electron.*, vol. 1, pp. 469–472, 2006.

[5] "mml2.jpg (640×343)." [Online]. Available: https://innovate.dreamcatcher.asia/images/mml2.jpg. [Accessed: 05-Dec-2018].

[6] "MATLAB - MathWorks - MATLAB & Simulink." [Online]. Available: https://www.mathworks.com/products/matlab.html?s_tid=hp_products_matlab . [Accessed: 15-Nov-2018].

[7] "Image Processing and Computer Vision - MATLAB & Simulink Solutions - MATLAB & Simulink." [Online]. Available: https://www.mathworks.com/solutions/image-video-processing.html. [Accessed: 15-Nov-2018].

[8] "Image Processing Toolbox - MATLAB." [Online]. Available: https://www.mathworks.com/products/image.html. [Accessed: 15-Nov-2018].

[9] F. Ozkan and B. Ulutas, "Use of an eye-tracker to assess workers in ceramic tile surface defect detection," *Int. Conf. Control. Decis. Inf. Technol. CoDIT 2016*, pp. 88–91, 2016.

[10] S. Asotra, A. Cossin, and A. Yacobi, "Costs of Failure in Product Quality.," *Pharm. Technol.*, vol. 36, no. 4, pp. 110–118, 2012.

[11] AIA, "What Are Vision Inspection Systems? - Vision Online," 2014. [Online]. Available: https://www.visiononline.org/vision-resources-details.cfm/vision-resources/What-Are-Vision-Inspection-Systems/content_id/4586. [Accessed: 15-Nov-2018].

[12] "CONCEPT OF PIXEL CONCEPT OF PIXEL Pixel."

[13] E. Golkar, A. Patel, L. Yazdi, and A. S. Prabuwono, "Ceramic Tile Border Defect Detection Algorithms in Automated Visual Inspection System," vol. 7, no. 6, pp. 542–550, 2011.

[14] H. M. Elbehiery, A. H. Alaa, and T. E. Muhammad, "Quality Control Enhancement via Non-Destructive," pp. 1130–1133, 2004.

[15] B. Bendjus, U. Cikalova, and L. Chen, "Crack detection of ceramics based on laser speckle photometry," *J. Ceram. Sci. Technol.*, vol. 8, no. 1, pp. 73–80, 2017.

[16] A. N. Shire, M. M. Khanapurkar, and R. S. Mundewadikar, "Plain ceramic tiles surface defect detection using image processing," *Int. Conf. Emerg. Trends Eng. Technol. ICETET*, pp. 215–220, 2011.

[17] Y. C. Samarawickrama and C. D. Wickramasinghe, "Matlab based Automated Surface Defect Detection System for Ceramic Tiles using Image Processing," *6th Natl. Conf. Technol. Manag.*, pp. 34–39, 2017.

[18] M. Roushdy, "Comparative study of edge detection algorithms applying on the grayscale noisy image using morphological filter," *GVIP J.*, vol. 6, no. 4, pp. 17–23, 2006.

[19] Z. Hocenski, S. Vasilic, and V. Hocenski, "Improved canny edge detector in ceramic tiles defect detection," *IECON Proc. (Industrial Electron. Conf.*, pp. 3328–3331, 2006.

[20] S. Choi, J. Yun, B. Seo, Y. Park, and S. Kim, "Real-Time Defects Detection Algorithm for High-Speed Steel Bar in Coil.," *Enformatika*, vol. 21, no. January, pp. 66–70, 2007.

[21] C. Boukouvalas, J. Kittler, R. Marik, M. Mirmehdi, and M. Petrou, "Ceramic tile inspection for colour and structural defects," *Techni*, pp. 390–399, 1995.

[22] R. Mishra, C. L. Chandrakar, and R. Mishra, "Surface Defects Detection for Ceramic Tiles Using Image Processing and Morphological Techniques," *IJREAS*, vol. 2, no. 2, pp. 1307–1322, 2012.

[23] F. S. Najafabadi and H. Pourghassem, "Corner defect detection based on dot product in ceramic tile images," *Proc. - 2011 IEEE 7th Int. Colloq. Signal Process. Its Appl. CSPA 2011*, pp. 293–297, 2011.

[24] Ž. Hocenski, T. Matić, and I. Vidović, "Technology transfer of computer vision defect detection to ceramic tiles industry," *Proc. 2016 Int. Conf. Smart Syst.*

Technol. SST 2016, pp. 301–305, 2016.

[25] T. Matić, I. Vidović, and Ž. Hocenski, “Real time contour based ceramic tile edge and corner defects detection,” *Teh. Vjesn.*, vol. 20, no. 6, pp. 1063–1070, 2013.

[26] Z. Hocenski, S. Vasilic, and V. Hocenski, “Cluster analysis in ceramic tiles images orientation,” *Proc. Int. Conf. Inf. Technol. Interfaces, ITI*, pp. 619–624, 2006.

[27] K. Ragab and N. Alsharay, “Developing Parallel Cracks and Spots Ceramic Defect Detection and Classification Algorithm Using CUDA,” *Proc. - 2017 IEEE 13th Int. Symp. Auton. Decentralized Syst. ISADS 2017*, pp. 255–261, 2017.

[28] B. You, C. Yan, M. Qiao, and J. Xu, “Research on detection system of ceramic tile based on image processing,” *ICEMI 2009 - Proc. 9th Int. Conf. Electron. Meas. Instruments*, vol. 3, pp. 628–632, 2009.

[29] M. Bohlool and S. R. Taghanaki, “Cost-efficient automated visual inspection system for small manufacturing industries based on SIFT,” *2008 23rd Int. Conf. Image Vis. Comput. New Zealand, IVCNZ*, pp. 0–5, 2008.

[30] Ž. Hocenski, A. Dizdar, and H. Verica, “Illumination Design of a Control System for Visual Inspection of Ceramic Tiles,” *IEEE Int. Symp. Ind. Electron.*, pp. 1093–1097, 2008.

[31] Ž. Hocenski, K. Sobol, and R. Mijaković, “LED panel illumination design of a control system for visual inspection of ceramic tiles,” *IEEE Int. Symp. Ind. Electron.*, vol. 2, no. 2, pp. 1663–1667, 2010.

[32] “DFK 72AUC02 - USB 2.0 color industrial camera.” [Online]. Available: https://www.theimagingsource.com/products/industrial-cameras/usb-2.0-color/dfk72auc02/. [Accessed: 13-Nov-2018].

[33] “DF6HA-1B | 1/2" 1.5 Megapixel Series | Factory Automation | Optical Devices | Fujifilm USA.” [Online]. Available: https://www.fujifilmusa.com/products/optical_devices/machine-vision/2-3-15/hf35ha-1b/index.html. [Accessed: 05-Dec-2018].

[34] “The Imaging Source Series of Cameras - Image Acquisition Toolbox Adaptor Kit for industrial cameras - Third-Party Products & Services - MATLAB & Simulink.” [Online]. Available: https://www.mathworks.com/products/connections/product_detail/the-

imaging-source-cameras.html. [Accessed: 15-Nov-2018].

APPENDICES

APPENDIX A MATLAB CODING FOR BACKGROUND SUBTRACTION ALGORITHMS

```
%Image acquisition
vid = videoinput('winvideo', 1, 'RGB32_2592x1944');
src = getselectedsource(vid);
vid.FramesPerTrigger = 1;
vid.ROIPosition = [37 1261 2528 226];
start(vid);
pause(1)
T = getdata(vid);
%imwrite(T,'C:\Users\sher kee\Documents\Sem7\FYP1\MATLAB file\reference.png')
imshow(T)
```

```
%T = imread('reference.png'); %read image file ****
%imshow(T)
BW = im2bw(T,0.80); %convert RGB image to binary image
figure;
imshow(BW)
```

```
CN = edge(BW,'Canny',0.999); %perform Canny edge detection operation
figure;
imshow(CN)
```

```
Ed = imdilate(CN,strel('disk',2));  %Dilate the edges
imshow(Ed)
```

```
b = imfill(Ed,'holes'); %fill empty space inside
imshow(b)
```

```
imwrite(b,'C:\Users\sher kee\Documents\Sem7\FYP1\MATLAB file\referencefill.png')
```

```
a = imread('referencefill.png'); %read ref image
[xr,yr]= xyimage(a);    %first xy point
[xrl,yrl] = fxyimage(a);    %last xy point
witdhr = xrl - xr;  %length
```

```
%Image acquisition
vid = videoinput('winvideo', 1, 'RGB32_2592x1944');
src = getselectedsource(vid);
vid.FramesPerTrigger = 1;
vid.ROIPosition = [37 1261 2528 226];
start(vid);
pause(5)
T = getdata(vid);
save('C:\Users\sher kee\Documents\Sem7\FYP1\MATLAB file\Sample1.mat', 'T');
imshow(T)
```

```
%Edge Detection
%T=imread('TileF.png')
%imshow(T)
BW = im2bw(T,0.80); %convert RGB image to binary image
figure;
imshow(BW)
CN = edge(BW,'Canny',0.999); %perform Canny edge detection operation
figure;
imshow(CN)
```

```
Ed = imdilate(CN,strel('disk',2));  %Dilate the edges
imshow(Ed)
b = imfill(Ed,'holes'); %fill empty space inside
imshow(b)
```

```
%Comparison
[xs,ys] = xyimage(b);
[xsl,ysl] = fxyimage(b);
witdhs = xsl - xs;
scale = witdhr / witdhs;
newb = imresize(b,scale);   %resize to ref size

[xs,ys] = xyimage(newb);
xt = xr - xs
yt = yr - ys
c = imtranslate(newb,[xt, yt]); %translate sample to ref position
imshowpair(a,c)

%make the c same dimension as ref
if scale < 1
    newc = [c,zeros(size(c,1),size(a,2)-size(c,2));zeros(size(a,1)-size(c,1),size(a,2))];
    d = a-newc;
else
    newa = [a,zeros(size(a,1),size(c,2)-size(a,2));zeros(size(c,1)-size(a,1),size(c,2))];
    d =newa-c;
end

imshow (d) %image of not overlap
area = bwareaopen(d,1200); %eliminate small area
cc = bwconncomp(d);
s = regionprops('table',cc,'area','MinorAxisLength');
l = labelmatrix(cc);
bw2 = ismember (l,find([s.MinorAxisLength]>20));   %eliminate narrow line
imshow (bw2)    %image of crack
%highlight crack area
if scale>=1
    imshowpair(c,bw2)
else
    imshowpair(newc,bw2)
end
```



```
%determine crack
x=0;
cond='good';

while x<size(bw2,2)
    x=x+1;
    y=0;
    while y<size(bw2,1)
        y=y+1;
        if bw2(y,x)>0
            cond='crack';
            break
        end
    end
end
final=cond
```

Function to process the image, find the edge coordinate

```
function [fxc,fyc] = xyimage(i) %find top xy to find length and translation
x = 0;
yc = 0;
xc= 0;
a= 1;
b = 1;
fxc = 0;
while x < size(i,2)
        x=x+1;
        y=0;
        while y < size(i,1)
            y=y+1;
            if i(y,x) == 1
                yc(x) = y;
                break
            end
        end

end

x=0;
while x < size(i,1)
        x=x+1;
        y=0;
        while y < size(i,2)
            y=y+1;
            if i(x,y) == 1
                xc(x) = y;
                break
            end
        end

end
while a < size(xc,2)
    if xc(a) > 0
        fxc(b) = xc(a);
        b=b+1;
    end
   a=a+1;
end

while a < size(yc,2)
    if yc(a) > 0
        fyc(b) = yc(a);
        b=b+1;
    end
   a=a+1;
end
T=1:length(fyc);
plot(T,fyc)
fyc = mode(fyc);
fxc = mode(fxc);
end
```

```
function [fxc,fyc] = fxyimage(i) %find bottom xy to find length and translation
x = 0;
yc = 0;
xc= 0;
a= 1;
b = 1;
fxc = 0;
while x < size(i,2)
        x=x+1;
        y=0;
        while y < size(i,1)
            y=y+1;
            if i(y,x) == 1
                yc(x) = y;
            end
        end

end

x=0;
while x < size(i,1)
        x=x+1;
        y=0;
        while y < size(i,2)
            y=y+1;
            if i(x,y) == 1
                xc(x) = y;
            end
        end

end

while a < size(xc,2)
    if xc(a) > 0
        fxc(b) = xc(a);
        b=b+1;
    end
   a=a+1;
end

while a < size(yc,2)
    if yc(a) > 0
        fyc(b) = yc(a);
        b=b+1;
    end
   a=a+1;
end
T=1:length(fyc);
plot(T,fyc)
fyc = mode(fyc);
fxc = mode(fxc);
end
```

APPENDIX B MATLAB CODING FOR DEFECT DETECTION BASED ON GRADIENT VARIATION ON BORDER EDGE

```
%Image acquisition
vid = videoinput('winvideo', 1, 'RGB32_2592x1944');
src = getselectedsource(vid);
vid.FramesPerTrigger = 1;
vid.ROIPosition = [37 1261 2528 226];
start(vid);
pause(5)
T = getdata(vid);
save('C:\Users\sher kee\Documents\Sem7\FYP1\MATLAB file\Sample1.mat', 'T');
imshow(T)
%imwrite(T,'C:\Users\sher kee\Documents\Sem7\FYP1\MATLAB file\Cracktile1.png')
```

```
%Edge detection
%clear all
T=imread('TileA1.png')
imshow(T)
BW = im2bw(T,0.80); %convert RGB image to binary image
figure;
imshow(BW)
CN = edge(BW,'Canny',0.999); %perform Canny edge detection operation
figure;
imshow(CN)
i = imdilate(CN,strel('disk',2));  %Dilate the edges
imshow(i)
```

```
%Defect Detection
x = 0; x1=0;
yc = 0; yc1=0;
xc= 0; xc1=0;
a= 1; a1=1;
b = 1; b1=1;
fxc = 0; fxc1=0;

%Find top edge coordinate and gradient
while x < size(i,2)
        x=x+1;
        y=0;
        while y < size(i,1)
            y=y+1;
            if i(y,x) == 1
                yc(x) = y;
                break
            end
        end

end
```

```
x=0;
while x < size(i,1)
        x=x+1;
        y=0;
        while y < size(i,2)
            y=y+1;
            if i(x,y) == 1
                xc(x) = y;
                break
            end
        end
```

```
end

while a < size(xc,2)
    if xc(a) > 0
        fxc(b) = xc(a);
        b=b+1;
    end
   a=a+1;
end

a=1;%
b=1;%
while a < size(yc,2)
    if yc(a) > 0
        fyc(a) = yc(a);
        %b=b+1;
    end
   a=a+1;
end
xaxis=1:length(fyc);
plot(xaxis,fyc)
figure;

a=1;
grad=0;
while a<size(fyc,2)-20
x=xaxis(a:a+20);
y=fyc(a:a+20);

[p,S,mu] = polyfit(x,y,1);
grad(a) = p(1);
f = polyval(p,x,[],mu);
plot(x,f,'-')
hold on
a=a+1;
end
hold off

grad
g=mode(grad)
[c,d]=max(grad)
plot(1:length(grad),grad)
hold on
plot([1 length(grad)],[g+1.5 g+1.5])
hold on
plot([1 length(grad)],[g-1.5 g-1.5])
hold off

v=1;
a=1;
k=0;
while v<length(grad)
    if  grad(v)>g+1.5 || grad(v) < g-1.5
        k(a)=v;
        a=a+1;
        plot(xaxis,fyc)
        hold on
        plot (v,fyc(v),'.','markersize',10,'Color','red')
    end
    v=v+1;
end
```

```
hold on
plot (1:length(grad),grad,'Color','y')
hold on
plot([1 length(grad)],[g+1.5 g+1.5])
hold on
plot([1 length(grad)],[g-1.5 g-1.5])

%Find bottom edge coordinate and gradient
while x1 < size(i,2)
        x1=x1+1;
        y1=0;
        while y1 < size(i,1)
            y1=y1+1;
            if i(y1,x1) == 1
                yc1(x1) = y1;
            end
        end

end

x1=0;
while x1 < size(i,1)
        x1=x1+1;
        y1=0;
        while y1 < size(i,2)
            y1=y1+1;
            if i(x1,y1) == 1
                xc1(x1) = y1;
            end
        end
end

while a1 < size(xc1,2)
    if xc1(a1) > 0
        fxc1(b1) = xc1(a1);
        b1=b1+1;
    end
   a1=a1+1;
end

a1=1;%
b1=1;%
while a1 < size(yc1,2)
    if yc1(a1) > 0
        fyc1(a1) = yc1(a1);
    end
   a1=a1+1;
end
xaxis1=1:length(fyc1);
figure;
plot(xaxis1,fyc1)
figure;

a=1;
grad1=0;
while a<size(fyc1,2)-20
    x=xaxis1(a:a+20);
    y=fyc1(a:a+20);

    [p,S,mu] = polyfit(x,y,1);
    grad1(a) = p(1);
    f = polyval(p,x,[],mu);
    plot(x,f,'-')
    %text(x(1),y(1),num2str(p(1)))
    hold on
    a=a+1;
```

```
end
hold off

grad1
g1=mode(grad1)
[c1,d1]=max(grad1)

plot(1:length(grad1),grad1)
hold on
plot([1 length(grad1)],[g1+1.5 g1+1.5])
hold on
plot([1 length(grad1)],[g1-1.5 g1-1.5])
hold off

v1=1;
a1=1;1
k1=0;
while v1<length(grad1)
    if grad1(v1)>g1+1.5 || grad1(v1) < g1-1.5
        k1(a1)=v1;
        a1=a1+1;
        plot(xaxis1,fyc1)
        hold on
        plot (v1,fyc1(v1),'.','markersize',10,'Color','red')
    end
    v1=v1+1;
end

hold on
plot (1:length(grad1),grad1,'Color','y')
hold on
plot([1 length(grad1)],[g+1.5 g+1.5])
hold on
plot([1 length(grad1)],[g-1.5 g-1.5])

hold off
group=0; group1=0;
group = mat2cell( k, 1, diff( [0, find(diff(k) ~= 1), length(k)] )) ;
group1 = mat2cell( k1, 1, diff( [0, find(diff(k1) ~= 1), length(k1)] )) ;
```



```
%Draw circle on image
a=1; a1=1;
b=1; b1=1;
x=0; x1=0;
y=0; y1=0;
r=0; r1=0;
ca=0; ca1=0;
h=0; h1=0;
s=0; s1=0;
xunit=0; xunit1=0;
yunit=0; yunit1=0;

%hold off
imshow(T)
while a <= size(group,2)
    h=max(group{a});
    s=min(group{a});
    r(a)=(h-s)/2;

    if a~=1&&r(a) >5
        x(a)=round(s+r(a));
        y(a)=fyc(x(a));
        hold on
        th = 0:pi/50:2*pi;
        xunit = r(a) * 10 * cos(th) + x(a);
```

```
        yunit = r(a) * 10 *sin(th) + y(a);
        h = plot(xunit, yunit,'Color','red');
        hold off
        ca=ca+1;
    end
    a=a+1;
end

while a1 <= size(group1,2)
    h1=max(group1{a1});
    s1=min(group1{a1});
    r1(a1)=(h1-s1)/2;


    if a1~=1 && r1(a1) > 5
        x1(a1)=round(s1+r1(a1));
        y1(a1)=fyc1(x1(a1));
        hold on
        th = 0:pi/50:2*pi;
        xunit1 = r1(a1) * 10 * cos(th) + x1(a1);
        yunit1 = r1(a1) * 10 *sin(th) + y1(a1);
        h1 = plot(xunit1, yunit1,'Color','red');
        hold off
        ca1=ca1+1;
    end
    a1=a1+1;
end
```

```
%determine crack
ca
ca1
if ca==0 && ca1==0
    cond='good';
else
    cond='crack';
end

final=cond
```

APPENDIX C SAMPLE TILES AND THE DETECTION OUTPUT

Tile A:

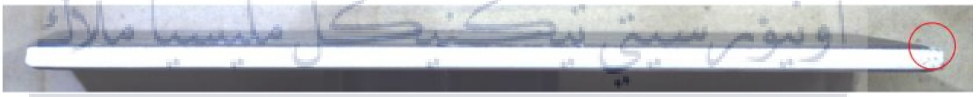
Output of method 1:

Output of method 2:

Tile B:

Output of method 1:

Output of method 2:



Tile C:

Output of method 1:

Output of method 2:

Tile D:

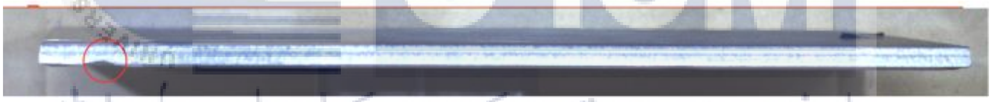
Output of method 1:

Output of method 2:

Tile E:

Output of method 1:

Output of method 2:



Tile F:

Output of method 1:

Output of method 2:

Tile G:

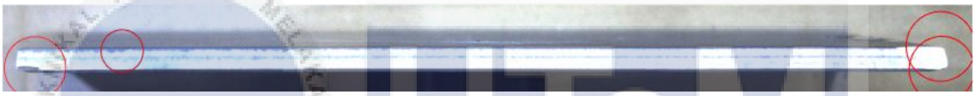
Output of method 1:

Output of method 2:

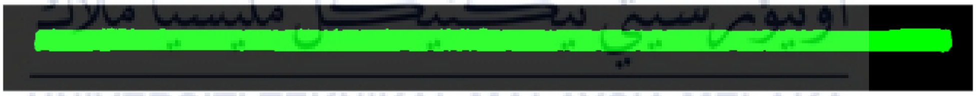
Tile H:

Output of method 1:

Output of method 2:



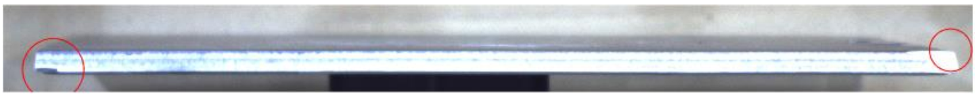
Tile I:



Tile J:

Output of method 1:

Output of method 2:



APPENDIX D GANTT CHART

Task	2018				2019					
	Sept	Oct	Nov	Dec	Jan	Feb	Mar	April	May	June
Research on title concept										
Literature review on project title										
Searching of problem on previous related works										
Verify objective to conduct project										
Research on hardware and software										
Design methodology										
Present proposal										
Finalise proposal report										
Conduct experiment										
Experiment 1										
Experiment 2										
Experiment 3										
Record all the observation and result										
Making discussion and conclusion										
Complete report										
Presentation on project										