ROBUST EDGE DETECTION DESIGN FOR HSA MISSING GASKET INSPECTION USING PARTICLE SWARM OPTIMIZATION

THUNYATIP VIPASETHARAT

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF ENGINEERING IN DATA STORAGE TECHNOLOGY
ENGLISH PROGRAM
INTERNATIONAL COLLEGE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2013
KMITL-2013-IC-M-005-019

หัวข้อวิทยานิพนธ์	การออกแบบการค้นหาขอบอย่างสมบูรณ์ของกระบวนการตรวจจับการสูญหายของแก็สเค็ทโดยใช้วิธีการหาค่าที่ดีที่สุดแบบพาร์ทิเคิลสวอร์มออพทิไมเซชัน
นักศึกษา	นางสาวธัญทิพย์ วิภาเศรษฐรัตน์
รหัสประจำตัว	54600705
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
ภาควิชา	เทคโนโลยีการบันทึกข้อมูล
พ.ศ.	2556
อาจารย์ที่ปรึกษาวิทยานิพนธ์	ผศ.ดร.สมยศ เกียรติวณิชวิลัย

บทคัดย่อ

แก็สเค็ทในกระบวนการประกอบ Head Stack (HSA) เป็นชิ้นส่วนสำคัญของฮาร์ดดิสค์ไดร์ฟ เนื่องจากเป็นชิ้นส่วนที่ดัดแปลงมาเพื่อทำหน้าที่สำคัญในการป้องกันปัญหาการรั่วซึมของอากาศระหว่างการประกอบฮาร์ดดิสค์ไดร์ฟ (HDD) ในปัจจุบัน กระบวนการตรวจสอบแก็สเค็ทนั้นทำโดยอาศัยทักษะของผู้ปฏิบัติงานในกระบวนการผลิต ซึ่งปฏิบัติไปตามรายการการตรวจสอบที่ถูกกำหนดมาให้เน้นการตรวจสอบการจัดวางให้ถูกตำแหน่งของแก็สเค็ทด้วย อย่างไรก็ตามปัจจุบันเราพบว่าการตรวจสอบด้วยสายตาคนนั้นทำให้ไม่มีความน่าเชื่อถือ และมีผลที่ผิดพลาดเกิดขึ้นในกระบวนการประกอบฮาร์ดดิสค์ไดร์ฟ (HDD) ดังในรายงานการตรวจพบความล้มเหลวที่กระบวนการประกอบฮาร์ดดิสค์ พบว่า 68 เปอร์เซ็นต์ของความล้มเหลวนั้นมีความเกี่ยวข้องกับแก็สเค็ทใน HSA ไม่ว่าจะเป็นการสูญหายหรือการประกอบไม่ถูกตำแหน่ง ซึ่งส่งผลกระทบถึงคุณภาพของผลิตภัณฑ์ ดังนั้นเพื่อก้าวล่วงปัญหาเหล่านี้ การตรวจสอบโดยใช้การมองเห็นอย่างอัตโนมัติจึงถูกนำมาดัดแปลง งานวิจัยนี้มุ่งเน้นการตรวจสอบแก็สเค็ทโดยประยุกต์ใช้การตรวจสอบโดยอัตโนมัติร่วมกับการออกแบบตัวแปรในกระบวนการตรวจจับขอบ พาร์ทิเคิลสวอมออพทิไมเซชั่นถูกนำมาประยุกต์ใช้เพื่อหาค่าที่ดีที่สุดของตัวแปรของการตรวจจับขอบแบบแคนนี ซึ่งก็คือค่าเบี่ยงเบนมาตรฐานของฟิลเตอร์และค่า threshold เพื่อทำให้การตรวจสอบเป็นไปอย่างแม่นยำที่สุดในการพัฒนาระบบการตรวจสอบนี้ ทั้งนี้ การแปรผันของแสงจากการเก็บภาพ ได้ถูกนำมาพิจารณาร่วมด้วยในการออกแบบฟังก์ชั่นการตรวจสอบของระบบ เพื่อความสมบูรณ์และบรรลุปัญหาการตรวจสอบเมื่อมีความผันแปรของความเข้มแสงมาเกี่ยวข้อง ผลการทดลองได้พิสูจน์ถึงประสิทธิภาพของอัลกอริทึมที่นำเสนอ เทียบกับการตรวจสอบตามแบบวิธีธรรมดา

Thesis	Robust Edge Detection Design for HSA Missing Gasket Inspection using Particle Swarm Optimization
Student	Miss Thunyatip Vipasetharat
Student ID.	54600705
Degree	Master of Engineering
Program	Data Storage Technology
Year	2013
Thesis Advisor	Asst. Prof. Dr. Somyot Kaitwanidvilai

ABSTRACT

In this paper, the robust edge detection filter design for gasket inspection is proposed. Gasket of HSA (Head Stack Assembly) is an important component in HDD (Hard Disk Drive); it is mainly adopted to prevent the air leak. The current gasket inspection is carried out by a skilled operator, following a pre-determined 'check list' which focuses on the alignment of gasket. However, the visual inspection by human causes unreliable and inaccurate results which impact to the product quality. To overcome this problem, automatic visual inspection is adopted. This research is focused on the inspection of the gasket by applying an automatic visual inspection with the new design of parameters of edge detection filter. Particle Swarm Optimization (PSO) is adopted to determine the optimal parameters of the Canny edge detection filter, i.e. standard deviation, threshold to make the best inspection accuracy of the developed inspection system. In addition, variation of light intensity by performing several captured images is included to the fitness function of the proposed design to achieve the robustness against the variation of light intensity. Experimental results verify the effectiveness of the proposed algorithm compared to the conventional design

Acknowledgements

I would like to extend my profound thanks to Seagate Technology (Thailand) Ltd., National Science and Technology Development Agency (NSTDA), and College of Data Storage Innovation, King Mongkut's Institute of Technology Ladkrabang for co-sponsorship, scholarship funding, and all supports throughout this research. I also would like to thank Asst. Prof. Dr. Somyot Kaitvanidwilai, my advisor for his kind support, especially on technical consultancy.

Finally, I would like to thank my family and my friends for all love.

Thunyatip Vipasetharat

Table of Contents

Page

Thai Abstract I
English Abstract II
Acknowledgements III
Table of Contents IV
List of Tables VI
List of Figures VII

Chapter 1 Introduction 1
1.1 Background and Problems 1
1.2 Objectives 2
1.3 Scope of Research 2

Chapter 2 Literature Review 4

Chapter 3 Edge Detection Technique and PSO 14
3.1 Edge Detection 14
3.2 Edge Thresholding 15
3.3 Canny Edge Detector 16
3.4 Particle Swarm Optimization (PSO) 18

Chapter 4 HSA Gasket Edge Detection 21
4.1 Input Original Images 21
4.2 Image preprocessing by applying Canny method for gasket edge detection scenarios 25
4.3 Adopt PSO to generate the first population 26
4.4 Feature Extraction 26
4.5 Adopt PSO to find the next particle 27

Chapter 5 Experimental Results 29
5.1 Results and Discussion 29

Chapter 6 Conclusion 46

Table of Contents (Cont.)

Page

Reference.. 48

Appendix 50

Biography 65

List of Tables

Table Page

2.1 The mean squared error of all techniques..........5

5.1 The edge images of gray scale with specific ROI from the Canny filter with the optimal values determined by the PSO..........33

5.2 The edge images of original grayscale preprocessing by calculating the average light intensity x interval values [0.5-1.0] obtained from the Canny filter with the optimal values determined by the PSO..........35

5.3 The edge images of original grayscale obtained from the Canny filter with the optimal values determined by the PSO..........38

5.4 The edge images of uninstalled gasket images obtained from the Canny filter with the optimal values determined by the PSO..........41

List of Figures

Figure Page

1.1 A gasket located on the HSA 1

1.2 Leak failure analysis summaries in FY2012 [data from Seagate Company] 2

2.1 Work flow of soldering joint inspection 4

2.2 Examples of edge images determined by conventional and proposed filter 5

2.3 Gray scale test image 6

2.4 Edge image using GA-based optimization 6

2.5 PSOTH Algorithm 7

2.6 Grouping a target image into regions Algorithm 8

2.7 The best Segmentation of Test Images Produced by (a, b, c). PSOTH method. (d, e, f). Normal region based algorithm. (g, h, i) *k*-means algorithm 8

2.8 PSO-SRG algorithm 9

2.9 Segmenting unlabeled pixels step 10

2.10 Cameraman image segmentation 10

2.11 Optimized segmentation using PSO and GMAC 11

3.1 Structure of gasket edge detection algorithm 14

4.1 Flow chart of robust edge detection design for HSA missing gasket inspection algorithm 21

4.2 Captured images adopted for the PSO optimization with different light intensity 22

4.2 (a) normal 22

4.2 (b) lower intensity 1 22

4.2 (c) lower intensity 2 22

4.2 (d) lower intensity 3 22

4.2 (e) higher intensity 1 22

4.2 (f) higher intensity 2 22

4.2 (g) higher intensity 3 22

4.3 (a) – (e) Captured gray scale images with specific ROI adopted for the PSO optimization 23

4.4 (a) – (g) Captured gray scale images adopted for the PSO optimization 24

4.5 Gasket and PCB images 25

4.5 (a) Original image from CCD camera 25

4.5 (b) Image by Canny method 25

4.5 (c) Image by Sobel method 25

4.5 (d) Image by Prewitt method 26

4.5 (d) Image by Roberts method 26

List of Figures (Cont.)

Figure Page

4.6 PSO algorithms to find the next particles 27

5.1 Fitness value versus iterations of RGB images 30

5.2 The edge images from the Canny filter with the optimal values determined by the PSO 30

5.2 (a) normal 30

5.2 (b) lower intensity 1 30

5.2 (c) lower intensity 2 30

5.2 (d) lower intensity 3 30

5.2 (e) higher intensity 1 30

5.2 (f) higher intensity 2 31

5.2 (g) higher intensity 3 31

5.3 The edge images from the Canny filter with parameters by trial and error method 31

5.3 (a) normal 31

5.3 (b) lower intensity 1 31

5.3 (c) lower intensity 2 31

5.3 (d) lower intensity 3 32

5.3 (e) higher intensity 1 32

5.3 (f) higher intensity 2 32

5.3 (g) higher intensity 3 32

5.4 Fitness value versus iterations of gray scale images with specific ROI 33

5.5 The edge images of gray scale with specific ROI 35

5.5 (a) the original image with light intensity variation 35

5.5 (b) edge image of (a) from the Canny filter with the optimal values determined by the PSO 35

5.6 Fitness value versus iterations of original gray scale images 37

5.7 The edge images of original gray scale 40

5.7 (a) the original image with light intensity variation 40

5.7 (b) edge image of (a) from the Canny filter with the optimal values determined by the PSO 40

CHAPTER 1
Introduction

1.1 Background and Problems

Gasket is a rectangular rubber that placed cover the print circuit board (PCB) of the HSA (Head Stack Assembly) part as shown in Figure 1.1. The size of the gasket will depend on the design of each HSA product. This part is called as the one of the important components of HSA due to its main role is used to prevent air leak during the assembly of PCB to the base desk of the drive.

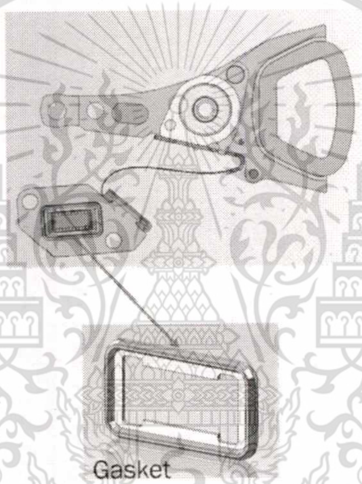


Figure 1.1 A gasket located on the HSA

Nowadays, the gasket inspection is still operated by human that causes long cycle time and high head count consumption leading to high HSA unit cost. Moreover, the visual inspection results in unreliable results due to human error from eyes-based fatique, time consuming, and inaccurate data. Especially, the missing gasket impacts to product quality. The summary report of leak failure that affected from the missing gasket is shown in Figure 1.2. The summary shows that the missing gasket from HSA assembly level decreased leak test yield at the Drive operation. Therefore, the developed gasket detection process is required for leak test yield improvement and quality issue closure.

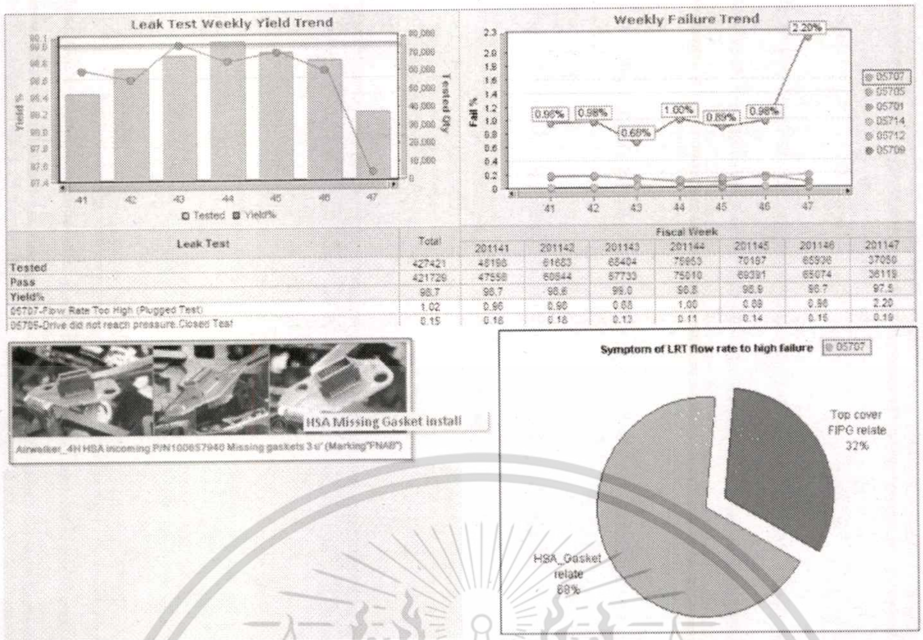


Figure 1.2 Leak failure analysis summaries in FY2012 [data from Seagate Company]

1.2 Objectives

In order to overcome labor cost increasing, the limitations and errors from the existing manual process for missing gasket inspection, this research focuses on the inspection of the gasket by applying an automatic visual inspection instead of manual operation. The new technique of the designing of image filtering parameters such as variance and threshold in order to improve the overall accuracy is proposed. Moreover, one of the main objectives of this research is to design an objective function of the design optimization to include the effects of illumination. Thus, the proposed technique is called as a robust design.

1.3 Scope of Research

1.3.1 Design a robust edge detection using population search algorithm (Particle Swarm Optimization) on MATLAB for HSA missing gasket inspection.

1.3.2 Find the optimal parameters to achieve the best edge detection filter with a variation of light intensity of the images.

1.3.3 Design the cost function that can be used to measure the goodness of the developed system.

This thesis is organized as follows. Chapter 2 describes the literature review of previous research articles which included both the image segmentation techniques and adopted population search algorithm. Chapter 3 describes the image detection filter and Particle swarm optimization technique which are used in the algorithm. The proposed technique is also described in this session. Chapter 4 explains the proposed method and Chapter 5 illustrates the results of the proposed algorithm to detect the gasket. Finally, Chapter 6 presents the conclusion of this research.

CHAPTER 2
Literature Review

There are many research works on the automatic visual inspection applied with image filtering techniques in various applications [1-7]. Many researchers proposed several new techniques to detect the objects. Moreover, some of thre reviewed works presented the applications of the intelligent algorithms to find optimal parameters in image processing. In this chapter, the relative works were reviewed and detailed for declaring the basic theory and the importance of the proposed technique.

In [1], the approach to develop new edge-detection filter based on genetic algorithms for soldering joint inspection was proposed. This paper, "The Development of a new edge detection filter based on genetic algorithm: an application to a soldering joint inspection." [1] was published by Kaitwanidvilai S. and Saenthon A. In their research, the measured data from high accurate inspection, i.e., destructive inspection, are included in the fitness function in their optimization problem. The inspection results from real experiments were performed to verify the effectiveness of their proposed algorithm. The algorithm starts from converting RGB image captured from the X-ray camera to gray scale image. Then, the second step, adjust the intensity by the normalized intensity method. The third step, apply an edge detection filter which is adopted to evaluate the edge of the soldering joint, and then the feature extraction; major length, minor length, bump ratio, and bubble are applied to find the specification of the soldering joint. Moreover, in their proposed technique, GA is adopted to find the proper parameters in the filter mask that is the key learning to show the successful results and gives inspiration to study about image search algorithm techniques. The work flow of soldering joint inspection is shown in Figure 2.1, the example of edge images determined by conventional and their proposed filter and the mean squared error of all techniques are shown in Figure 2.2 and Table 2.1, respectively.

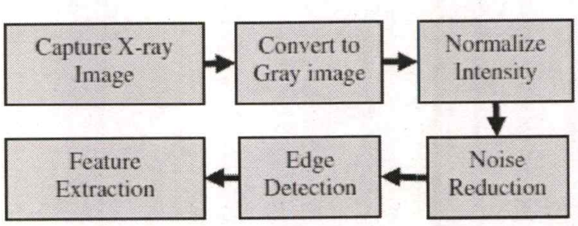


Figure 2.1 Work flow of soldering joint inspection [1]

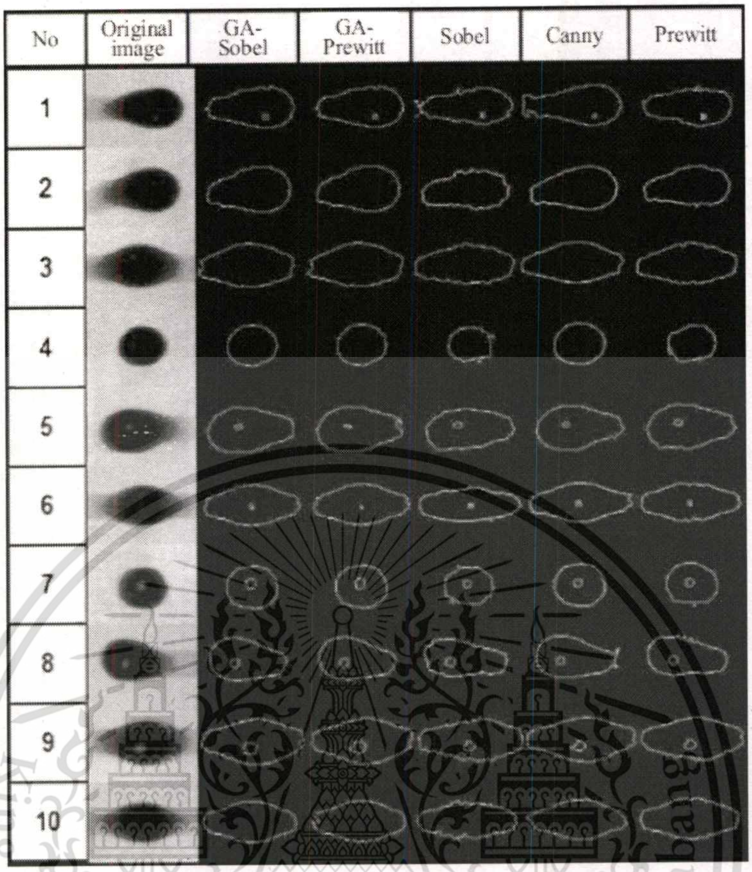


Figure 2.2 Examples of edge images determined by conventional and proposed filter [1]

Table 2.1 The mean squared error of all techniques [1].

Techniques (number of samples = 50)	MSE (major length; $\times 10^{-12}$)	MSE (minor length; $\times 10^{-12}$)	MSE (bump ratio; $\times 10^{-3}$)
Proposed technique with Sobel structure	33.6	21.5	1.1
Proposed technique with Prewitt structure	53.2	22.3	0.84
Sobel filter	763.6	355.5	28.2
Canny filter	682.5	76.0	12.5
Prewitt filter	1,118.9	308.4	28.5

The next research article is “Automatic visual inspection of bump in hard disk drive component using neural network and image processing.”[2] by Kaitwanidvilai S. and Saenthon A. In this technique, image processing is adopted to find the main features of the captured image and the neural network is used to classify the completeness of bump. The main conclusion of this paper states that there is not only one algorithm can be solved all problems in the image segmentation process. However, the proper algorithm can be selected to perform the better result in the image processing work.

In 1993, Bhandarker M, Zhang Y, and Potter D adopted a genetic algorithm (GA) for choosing a minimum cost edge configuration in research "A Genetic Algorithm-based Edge Detection Technique."[3]. There are five factors to be considered in an optimization problem: region dissimilarity, thickness, curvature, edge fragmentation, and the number of edge pixels detected. A simulated annealing algorithm and GA were adopted to determine the best results as shown in Figure 2.3 and the result is shown in Figure 2.4, respectively.

Figure 2.3 Gray scale test image [3]

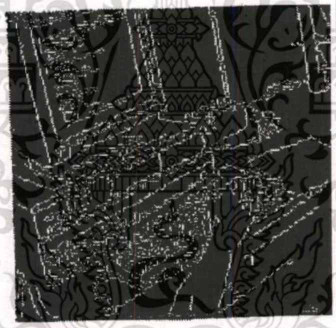


Figure 2.4 Edge image using GA-based optimization [3]

In 2011, Verma O. and Sharma R. used the ACO and the law of universal gravity to tackle the edge detection problem in "An Optimal Edge Detection using Universal Law of Gravity and Ant Colony Algorithm"[4]. The concept of universal gravity is used to calculate the heuristic function which guides the ant towards the most promising solution. The proposed ACO-based image edge detection approach is to utilize a number of ants to move on a 2-D image for constructing a pheromone matrix, each entry of which represents the edge information at each pixel location of the image. The proposed approach is much faster, less time consuming, requires less memory, and more focused on the optimization concerns of the edge detection problem.

The above articles have been referred to the performance of the techniques that used to implement since several decades ago. Recently, there is one of the search methods perform well to solve any optimization problems. This technique is called the Particle Swarm Optimization (PSO). There are many applications applied the PSO to improve effectiveness of problems solving.

In [5], The PSO is adopted as the main algorithm to solve the thresholding problem. In this paper, “A new Optimization-Based Image Segmentation method by Particle Swarm Opmization” [5] was proposed by Fahd M. A. Mohsen, Mohiy M. Hadhoud, and Khalid Amin. The researchers used the PSO to find the best values of thresholds to come out the target images. The function called the new quantitative evaluation function is used to be the objective function for the algorithm of PSO. A new multilevel thresholding method for segmenting images using PSO is developed and be called PSOTH. This algorithm aims to find near-optimal values of thresholds that can give a near-optimal segmentation. The PSO algorithm originates a random swarm of M particles, where each one has k thresholds, and flies them on the search space to find the target partition according to a fitness function. The fitness function they use is Q' function (one of a new quantitative evaluation function that gives more balancing between the number of regions and their homogeneities than the other functions for the segmentation results). The PSO algorithm is shown in Figure 2.5.

Algorithm 1: PSOTH Algorithm
1- **Do** (3 x 3) Low pass filter.
2- **Initialize** randomly each particle to contain k threshold values.
3- **For** each particle, i, **Sort** the threshold values. *// Optional*
3- **Repeat** the following **Until** the number of iterations has been satisfied
(a) **For** each particle i :
(i) **Threshold** the target image into regions using the threshold values of particle i.
(ii) **Compute** the fitness of particle i by using Q' function.
(b) **Find** the global best solution (*gbest*).
(c) **For** each particle i, **Update** the threshold values.
4- **Fix** the gbest as a best solution.
5- **Merge** small regions of the best solution.

Figure 2.5 PSOTH Algorithm [5]

To threshold and group a target image into regions, the principle of normal region based segmentation is applied as well. The algorithm of thresholding and grouping the target image into regions is summarized in Algorithm 2 as shown in Figure 2.6.

Algorithm 2: Grouping a target image into regions.
- **For** each pixel, px, in the target image.
- **If** Pixel, px, is ungrouped **Then**
(i) **Grow** px with its neighboring pixels according to the threshold values of particle i.
(ii) - **Add** the new region to the list of regions.
- **End if**
- **End For**

Figure 2.6 Grouping a target image into regions Algorithm [5]

To prove the effectiveness of the PSOTH algorithm, they applied the PSOTH to many kinds of images and compared it with the other algorithms. Figures 2.7 (a, b, c) illustrates the segmented images produced through applying the PSOTH method to the test images, while Figures 2.7 (d, e, f) and Figures 2.7 (g, h, i) illustrate the segmented images obtained from applying the k-mean algorithm and the region based segmentation to the test images respectively. As human visual observed, it can be seen that the results shown in the figures clearly illustrate that the segmented images obtained by the PSOTH method are better than the segmented images obtained from the region based segmentation and the means algorithm. This means that the PSO algorithm is a good optimization method can be used to segment images if it is complicated with a good fitness function such as Q' function.

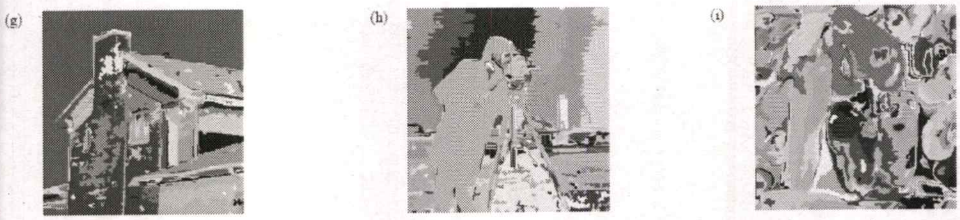


Figure 2.7 The best Segmentation of Test Images Produced by (a, b, c). PSOTH method. (d, e, f). Normal region based algorithm. (g, h, i) *k*-means algorithm [5].

As seen in the results of their research, the PSO is used to be the main algorithm to produce a new optimization-based image segmentation method called PSOTH. One of the advantages is that the PSO is a flexible optimization method. Many objective functions can be used with this method to compare the effectiveness that leads to they are able to propose a new quantitative evaluation function as a fitness function for the algorithm. Finally, they can prove the performance by comparing the results from the PSOTH method to those by the conventional methods.

The same researchers proposed a new segmentation method based on PSO by combining PSO algorithm with one of region-based image segmentation method called Seeded Region Growing (SRG). The title of this paper is “A New Image Segmentation Method Based on Particle Swarm Optimization” [6]. In this paper, the algorithm of PSO in the new method was applied to solve the two problems of SRG method through refining the position and similarity difference value of each seed point. In conclusion, in the proposed method, region merging was applied to merge small regions in the segmented image. The proposed method is called PSO-SRG, which aims to find the best locations for seed points and solve the similarity criteria of pixels in the regions. The PSO-SRG algorithm is summarized as Figure 2.8.

1. Do 3×3 Low pass filter (Pre-processing).
2. Initialize randomly each particle to contains k seed points.
3. Repeat the following steps Until number of iteration or stopping condition is satisfied.
   a. For each particle i :
      1. Segment the image using the seed points of particle i, (using seeded region growing method).
      2. Compute the fitness of particle i, f (i).
   b. Find the global best solution (gbest).
   c. For each particle i, Update the seed points.
4. Segment unlabeled pixels of the best solution, if any.
5. Merge small regions of the best solution.

Figure 2.8 PSO-SRG algorithm [6]

For the optimal segmentation, two steps must be done; segmenting unlabeled pixels and merging small regions. The segmenting unlabeled pixels step is summarized as Figure 2.9

For each Pixel , px, in the target image
If Pixel, px, is unlabeled Then
1. Do Seeded Region Growing algorithm using the seed point px with similarity difference $d=(d_{min}+d_{max})/2$.
2. Add new region to region list.
End If
End For

Figure 2.9 Segmenting unlabeled pixels step [6]

To prove the efficiency of the PSO-SRG method, the best solution of their experiment is compared with the result produced by implementing the conventional region-based image segmentation method. Figure 2.10 (a) shows the result of image segmentation by PSO-SRG method and Figure 2.10 (b) shows the result simulated by region-based method. It can be seen that their proposed algorithm can find more homogeneous regions than the conventional region-based method.

a) With PSO-SRG. b) With Region-based.

Figure 2.10 Cameraman image segmentation [6]

As their research [6], the PSO was merged with other segmentation method; Seeded Region Growing Segmentation (SRG). We can see that the PSO is an algorithm applied to help improve the potential of optimal parameters tuning with many objective functions.

PSO is also applied to the other field such as medical image segmentation. In 2012, J. Umamaheswari and G. Radhamani proposed an optimal approach for medical image segmentation based on the combination of Particle Swarm Optimization (PSO) and Global Minimization by Active Contour in “An Optimal Approach for Medical Image Analysis” [7]. Researchers presented the optimal approach on the combination of PSO and GMAC algorithm. The detailed explanation of the detection system is described in Figure 2.11 which shows an optimal segmentation system. The preprocessing is the first stage before starting the other processes to produce noise suppression and image enhancement. The second process consists of the PSO with GMAC segmentation for irregular brain detection. The performance of the segmentation is compared with other segmentation method.

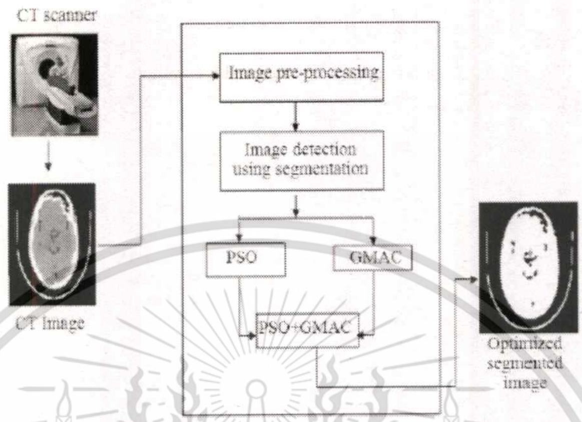


Figure 2.11 Optimized segmentation using PSO and GMAC [7]

In their proposed algorithm [7], the PSO image segmentation is carried out by adjusting the curve parameters according to the undesirable object, the segmentation process is done by the following sequence:

Step 1 Select the curve parameters arbitrarily from the range specified in [7] and create the corresponding level set functions

Step 2 Segment the image by using the curves derived from the generated level set functions

Step 3 Measure the fitness of each curve by computing the fitness function described in the PSO algorithm and determine the best curve

Step 4 Update the curve parameters according to the PSO algorithm equations.

Step 5 Create the level set functions of the new parameters and repeat Step-2

Repeat the Step-3

If the best curve is not changed for more than 10 epochs, produce the segmentation results; else go to Step-4

Nevertheless, from this research, they found the disadvantages of PSO algorithm approaches for the medical application as below:

1. Some limitations in real-time applications due to relatively slow computation time (Possibility for the off-line real-world problems).

2. The problems of dependency on initial conditions, parameter values, difficulty in finding the optimal design parameters, stochastic characteristics of the final outputs

3. The major drawback of PSO, like in other heuristic optimization techniques, it lacks a solid mathematical foundation for analysis. The PSO is a variant of stochastic optimization techniques requiring relatively a longer computation time than mathematical approaches.

4. The problems of dependency on initial point and parameters, difficulty in finding their optimal design parameters and the stochastic characteristic of the final outputs

In the case of medical image segmentation, the aim of research focuses is to study anatomical structure, identify the region of interest, measure abnormality and help doctors in planning for early diagnosis. In this paper [7], they proposed an optimal approach for medical image segmentation based on the combination of PSO and Global Minimization by Active Contour (GMAC) methods. The grouped part from PSO is again treated with GMAC to reduce the complex region of image parts. The simulation result gives that the proposed optimal approach gives efficient results for medical image segmentation based on parametric metrics.

The last article reviewed in this thesis that shows the effectiveness of PSO is presented by P.D. Sathya and R. Kayalvizhi as "PSO-Based Tsallis Thresholding Selection Procedure for Image Segmentation" [8]. In this research, the PSO algorithm is used to find the optimal threshold values which maximize the Tsallis objective function. This paper presented a quick solution to the multilevel image thresholding problems using the PSO algorithm. The number of threshold levels is the dimension of the problem. Its implementation consists of the following steps.

Step 1. *Initialization of parameters:* For a population size p, the particles are randomly generated between the minimum and the maximum limits of the threshold values.

Step 2. *Evaluation of the objective function:* The objective function values of the particles are evaluated using the objective functions.

Step 3. *Initialization of pbest and gbest:* The objective values obtained above for the initial particles of the swarm are set as the initial pbest values of the particles. The best value among all the pbest values is identified as gbest.

Step 4. *Evaluation of velocity:* The new velocity for each particle is computed.

Step 5. *Update the swarm:* The particle position is updated. The values of the objective function are calculated for the updated positions of the particles. If the new value is better than the previous pbest, the new value is set to pbest. Similarly, gbest value is also updated as the best pbest.

Step 6. *Stopping criteria:* If the objective functions are achieved, the positions of particles represented by gbest are the optimal threshold values. Otherwise, the procedure is repeated from step 4.

From the experimental results of this research, nonextensive entropy image thresholding can be proved as a powerful technique for image segmentation. In this paper, a new multilevel thresholding method based on Particle Swarm Optimization (PSO) is proposed which enables in determining the optimal threshold values by maximizing Tsallis objective function. Experiments with several standard test images have proved the robustness of the proposed method. For the accuracy of image segmentation, evaluated through the PSNR measure and the objective function. Compared with the GA method, it shows that thier proposed method is less processing time and more stability.

CHAPTER 3
Edge Detection Technique and PSO

In the previous chapter, many research articles presented the effectiveness of image edge detection algorithms and search algorithm in order to specify and evaluate the results of interesting object. The edge detection and PSO are needed for the proposed technique. Hence, this chapter describes the techniques which are proposed to find the edge of the gasket by finding the optimal parameters tuning that effect to images in this work.

The design phase, workflow is divided into 5 main sections which are input original images, Canny edge detection, and PSO algorithm to find optimal parameters and the gasket edge detected as shown in Figure 3.1.

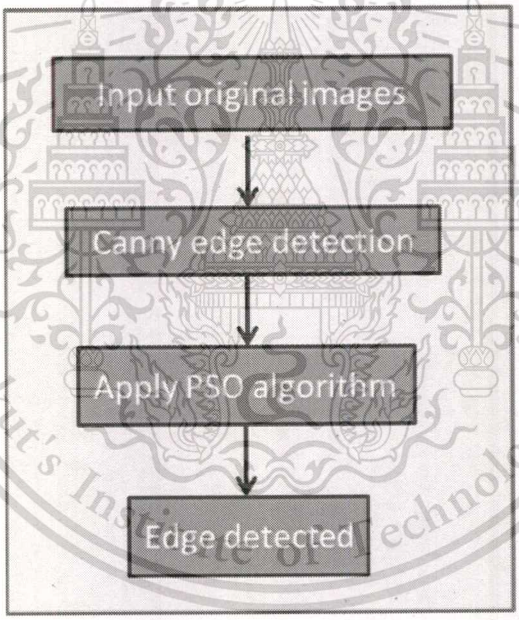


Figure 3.1 Structure of gasket edge detection algorithm

3.1 Edge Detection

Edges are basic image features. They carry useful information about object boundaries which can be used for image analysis, object identification and for image filtering applications as well. Although point and line detection are certainly important in any discussion on image segmentation, edge detection [9-10] is by far the most common approach for detecting meaningful discontinuities in intensities values. Such discontinuities are detected by using first- and second-order derivatives.

The first-order derivative of choice in image processing is the gradient. The gradient of a 2-D function, $f(x,y)$, is defined as the vector

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix} \tag{3.1}$$

The magnitude of this vector is

$$\nabla f = mag(\nabla f) = \left[G_x^2 + G_y^2 \right]^{1/2} \tag{3.2}$$

$$= \left[(\partial f / \partial x)^2 + (\partial f / \partial y)^2 \right]^{1/2}$$

To simplify computation, this quantity is approximated sometimes by omitting the square-root operation,

$$\nabla f \approx G_x^2 + G_y^2 \tag{3.3}$$

Or by using absolute values,

$$\nabla f \approx |G_x| + |G_y| \tag{3.4}$$

These approximations still behave as derivatives; that is, they are zero in areas of constant intensity and their values are proportional to the degree of intensity change in areas whose pixel values are variable. It is common practice to refer to the magnitude of the gradient or its approximations simply as "the gradient."

A fundamental property of the gradient vector is that it points in the direction of the maximum rate of change of f at coordinates (x, y). The angle at which this maximum rate of change occurs is

$$\alpha(x,y) = \tan^{-1}\left(\frac{G_y}{G_x} \right) \tag{3.5}$$

3.2 Edge Thresholding

Most edge detectors produce a gray scale output image ∇f. This image carries information about the edge magnitude. If the edge detector output is large, a local edge is presented. Otherwise, this pixel location corresponds to background. Therefore, a thresholding [9-10] operation is required after edge detection:

$$\nabla f = \begin{cases} 1 & \text{if } \nabla f \geq T \\ 0 & \text{otherwise} \end{cases} \qquad (3.6)$$

The threshold T can be chosen by inspecting the edge detector output histogram, so that only a small percentage of the pixel ∇f above it. Thresholding is global, because T is chosen based on global information and it is applied to the entire image. In many applications, the edge detector output has regions possessing different statistical properties. Therefore, global thresholding may produce thick edges in one region and thin or broken edges in another region. Thus, locally adapted thresholding is desirable. The thresholding operation is still described by ∇f with threshold adapted locally. Several heuristic adaptation techniques can be used.

3.3 Canny Edge Detector

The Canny edge detector [11] is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images. It was developed by John F. Canny in 1986. Canny also produced a computational theory of edge detection explaining why the technique works. To satisfy these requirements Canny used the calculus of variations – a technique which finds the function which optimizes a given function. The optimal function in Canny's detector is described by the sum of four exponential terms, but it can be approximated by the first derivative of a Gaussian.

Stages of the Canny algorithm

3.3.1 Noise reduction

Because the Canny edge detector is susceptible to noise presented in the raw image data, it uses a filter based on a Gaussian (bell) curve, where the raw image is convoluted with a Gaussian filter. The result is a slightly blurred version of the original, which is not affected by a single noisy pixel to any significant degree.

Here is an example of a 5x5 Gaussian filter, used to create the image to the right, with σ = 1.4. (The asterisk denotes a convolution operation.)

$$B = \frac{1}{159}\begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} * A \qquad (3.7)$$

3.3.2 Finding the intensity gradient of the image

An edge in an image may points in a variety of directions, so the Canny algorithm uses four filters to detect horizontal, vertical and diagonal edges in the blurred image. The edge detection operator (Roberts, Prewitt, Sobel for example) returns a value for the first derivative in the horizontal direction (G_x) and the vertical direction (G_y). From this the edge gradient and direction can be determined:

$$G = \sqrt{G_x^{\,2} + G_y^{\,2}} \qquad (3.8)$$

$$\Theta = a\tan 2\left(G_y, G_x\right) \qquad (3.9)$$

Where G can be computed using the hypot function and atan2 is the arctangent function with two arguments. The edge direction angle is rounded to one of four angles representing vertical, horizontal and the two diagonals (0, 45, 90 and 135 degrees for example).

3.3.3 Non-maximum suppression

Non-maximum suppression is an edge thinning technique. Given several estimates of the image gradients, a search is carried out to determine if the gradient magnitude assumes a local maximum in the gradient direction. In some implementations, the algorithm categorizes the continuous gradient directions into a small set of discrete directions, and then moves a 3x3 filter over the output of the previous step (that is, the edge strength and gradient directions). At every pixel, it suppresses the edge strength of the center pixel (by setting its value to 0) if its magnitude is not greater than the magnitude of the two neighbors in the gradient direction. For example,

1. If the rounded gradient angle is zero degrees (i.e. the edge is in the north-south direction) the point will be considered to be on the edge if its gradient magnitude is greater than the magnitudes at the pixels in the north and south directions,

2. If the rounded gradient angle is 90 degrees (i.e. the edge is in the east-west direction) the point will be considered to be on the edge if its gradient magnitude is greater than the magnitudes at pixels on the east and west directions,

3. If the rounded gradient angle is 135 degrees (i.e. the edge is in the north east-south west direction) the point will be considered to be on the edge if its gradient magnitude is greater than the magnitudes at the pixels on the north east and south west directions,

4. If the rounded gradient angle is 45 degrees (i.e. the edge is in the north west-south east direction) the point will be considered to be on the edge if its

gradient magnitude is greater than the magnitudes at the pixels on the north west and south east directions.

In more accurate implementations, linear interpolation is used between the two neighboring pixels that straddle the gradient direction. For example, if the gradient angle is between 45 degrees and 90 degrees interpolation between gradients at the northeast and east pixels will give one interpolated value, and interpolation between the southwest and west pixels will give the other (using the conventions of last paragraph). The gradient magnitude at the central pixel must be greater than both of these for it to be marked as an edge.

3.3.4 Tracing edges though the image and hysteresis thresholding

Large intensity gradients are more likely to correspond to edges than small intensity gradients. It is in most cases impossible to specify a threshold at which a given intensity gradient switches from corresponding to an edge into not doing so. Therefore Canny uses thresholding with hysteresis.

Thresholding with hysteresis requires two thresholds – high and low. Making the assumption that important edges should be along continuous curves in the image allows us to follow a faint section on a given line and to discard a few noisy pixels that do not constitute a line but have produced large gradients. Therefore, we begin by applying a high threshold. This marks out the edges we can be fairly sure are genuine. Starting from these, using the directional information derived earlier, edges can be traced through the image. While tracing an edge, we apply the lower threshold, allowing us to trace faint sections of the edges as long as we find a starting point.

Once this process is complete, the results of a binary image where each pixel is marked as either an edge pixel or a non-edge pixel is obtained. From the complementary output from the edge tracing step, the binary edge map obtained in this way can also be treated as a set of edge curves, which after further processing can be represented as polygons in the image domain.

3.4 Particle Swarm Optimization (PSO)

Particle swarm optimization (PSO) [12-13] is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. PSO optimizes a problem by having a population of candidate solutions, here dubbed particles, and moving these particles around in the search-space according to simple mathematical formulae over the particle's position and velocity. Each particle's movement is influenced by its local best known position but, is also guided toward the best known positions in the search-space, which are updated as better positions are found by other particles. This is expected to move the swarm toward the best solutions.

PSO is originally attributed to Kennedy, Eberhart and Shi and was first intended for simulating social behaviour, as a stylized representation of the movement of organisms in a bird flock or fish school. The algorithm was simplified and it was observed to be performing optimization. The book by Kennedy and Eberhart describes many philosophical aspects of PSO and swarm intelligence. An extensive survey of PSO applications is made by Poli.

PSO is a metaheuristic as it makes few or no assumptions about the problem being optimized and can search very large spaces of candidate solutions. However, metaheuristics such as PSO do not guarantee an optimal solution is ever found. More specifically, PSO does not use the gradient of the problem being optimized, which means PSO does not require that the optimization problem be differentiable as is required by classic optimization methods such as gradient ascent and quasi-newton methods. PSO can therefore also be used on optimization problems that are partially irregular, noisy, change over time, etc.

Algorithm

A basic variant of the PSO algorithm works by having a population (called a swarm) of candidate solutions (called particles). These particles are moved around in the search-space according to a few simple formulae. The movements of the particles are guided by their own best known position in the search-space as well as the entire swarm's best known position. When improved positions are being discovered, these will then come to guide the movements of the swarm. The process is repeated and by doing so it is hoped, but not guaranteed, that a satisfactory solution will eventually be discovered.

Formally, let *f*: $R^n \rightarrow R$ be the cost function which must be minimized. The function takes a candidate solution as an argument in the form of a vector of real numbers and produces a real number as output, which indicates the objective function value of the given candidate solution. The gradient of *f* is not known. The goal is to find a solution *a* for which $f(a) \leq f(b)$ for all b in the search-space, which would mean **a** is the global minimum. Maximization can be performed by considering the function $h = -f$ instead.

Let *S* be the number of particles in the swarm, each having a position $x_i \in R^n$ in the search-space and a velocity $v_i \in R^n$. Let p_i be the best known position of particle *I* and let *g* be the best known position of the entire swarm. A basic PSO algorithm is then:

For each particle *i* = 1,..., *S* do:

Initialize the particle's position with a uniformly distributed random vector: $x_i \sim U(b_{lo}, b_{up})$, where b_{lo} and b_{up} are the lower and upper boundaries of the search-space.

Initialize the particle's best known position to its initial position: $p_i \leftarrow x_i$

If $(f(p_i) < f(g))$ update the swarm's best known position: $g \leftarrow p_i$

Initialize the particle's velocity:

$$v_i \sim U\left(-\left|b_{up} - b_{lo}\right|, \left|b_{up} - b_{lo}\right|\right) \qquad (3.16)$$

Until a termination criterion is met (e.g. number of iterations performed, or a solution with adequate objective function value is found), repeat:

For each particle i = 1,..., S do:

Pick random numbers: $r_p, r_g \sim U(0,1)$

For each dimension d = 1, ..., n do:

Update the particle's velocity:

$$v_{i,d} \leftarrow \omega v_{i,d} + \varphi_p r_p \left(p_{i,d} - x_{i,d}\right) + \varphi_g r_g \left(g_d - x_{i,d}\right) \qquad (3.17)$$

Update the particle's position: $x_i \leftarrow x_i + v_i$

If $(f(x_i) < f(p_i))$ do:

Update the particle's best known position: $p_i \leftarrow x_i$

If $(f(p_i) < f(g))$ update the swarm's best known position: $g \leftarrow p_i$

Now **g** holds the best found solution.

The parameters ω, φ_p, and φ_g are selected by the practitioner and control the behavior and efficacy of the PSO met.

CHAPTER 4
HSA Gasket Edge Detection

The HSA gasket detection algorithm is developed to inspect the missing gasket of the product. Moreover, this algorithm is designed to improve the robustness and accuracy of automatic visual inspection. To design methodology in the algorithm, the edge detection techniques and optimization tools are reviewed in Chapter 2 and 3. Eventually, Canny edge detection filter and Particle Swarm Optimization technique are the most appropriate techniques to be applied for the proposed edge detection algorithm of missing gasket inspection. The proposed scenarios are shown in Figure 4.1.

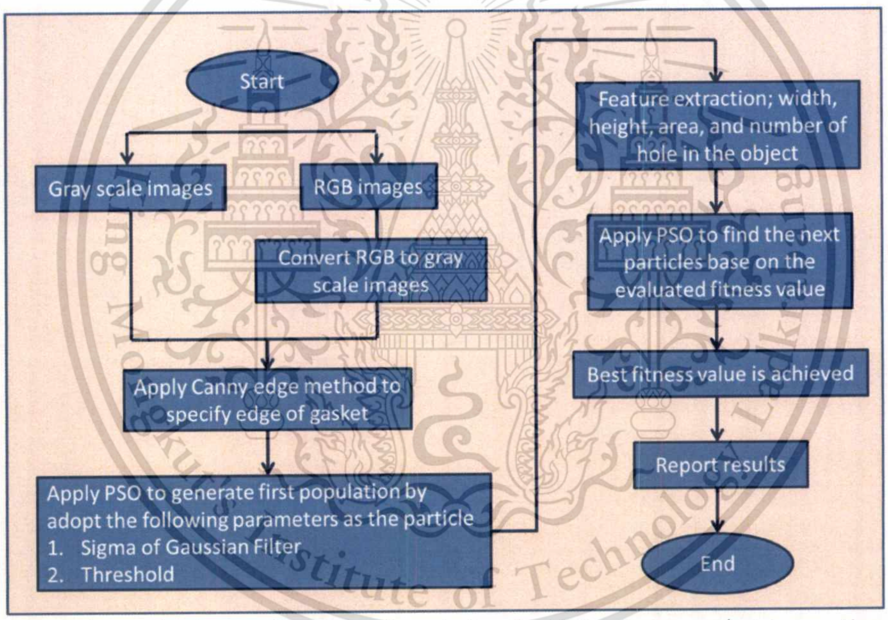


Figure 4.1 Flow chart of robust edge detection design for HSA missing gasket inspection algorithm

4.1 Input Original Images

The prepared samples are captured the image by CCD 5 Megapixels camera, CCTV lens 12.5 mm with full resolution 2456 x 2056 capturing. Pixel size is 3.45 x 3.45 μm. Each image contains the same object which is a gasket. We separate the input images as 3 cases as followings:

4.1.1 RGB images

We adopted 7 images with different light intensity. The original RGB images are shown in Figure 4.2 (a-g).

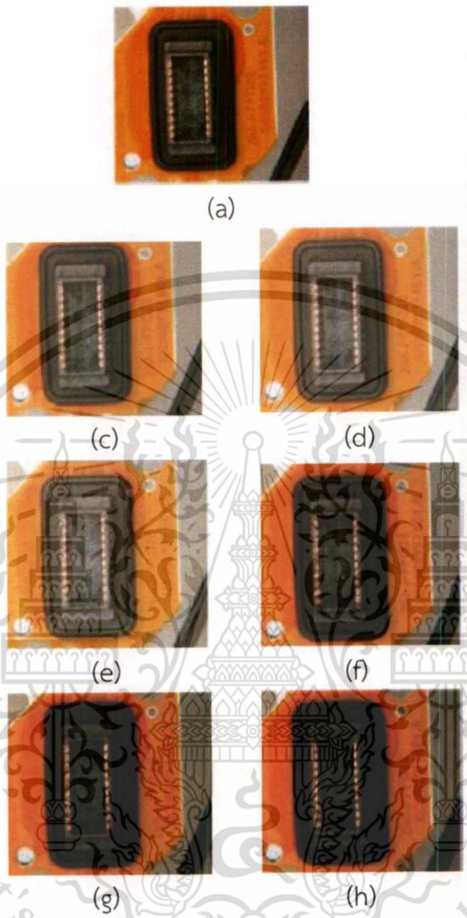
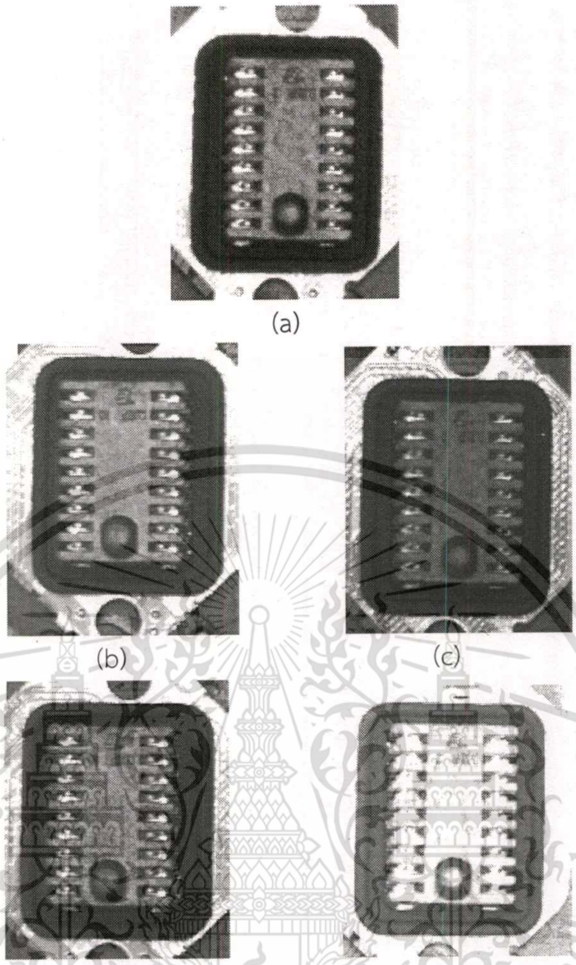


Figure 4.2 Captured images adopted for the PSO optimization with different light intensity: (a) normal (b) lower intensity 1 (c) lower intensity 2 (d) lower intensity 3 (e) higher intensity 1 (f) higher intensity 2 and (g) higher intensity 3

4.1.2 Gray scale images with specific ROI 5 images.

We adopted these 5 images for training the simulated program. The original 5 images are shown in Figure 4.3 (a-e).



(a)

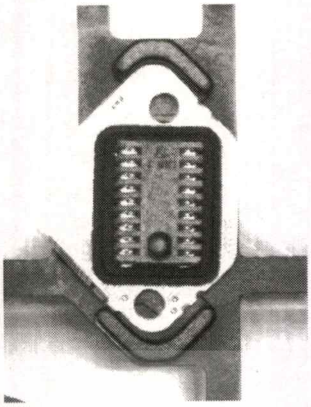
(b) (c)

(d) (e)

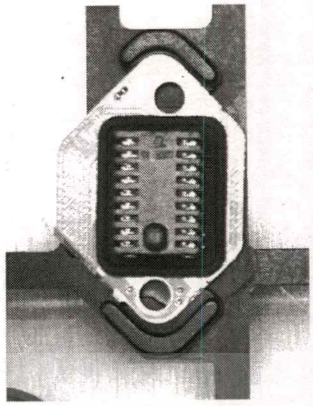
Figure 4.3 (a) – (e) Captured gray scale images with specific ROI adopted for the PSO optimization

4.1.3 Gray scale images without specified ROI, 7 images.

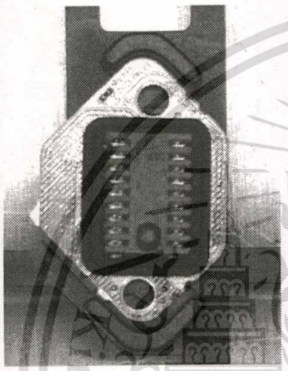
These original 7 images represent the original grayscale images from the actual scenario. Hence, they are adapted for PSO optimization to prove the effectiveness of algorithm in case of noise and environmental features are applied. The original 7 images are shown in Figure 4.4 (a-g).



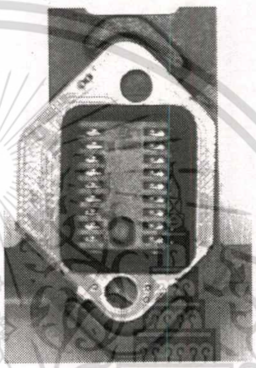
(a)



(b)



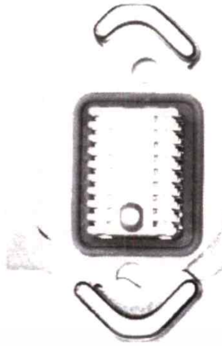
(c)



(d)

(e)

(f)



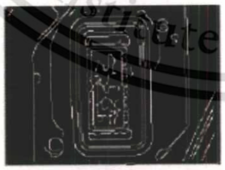
(g)

Figure 4.4 (a) – (g) Captured gray scale images adopted for the PSO optimization

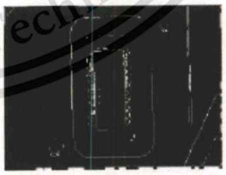
4.2 Image preprocessing by applying canny method for gasket edge detection scenarios

4.2.1 Convert RGB image to gray scale image then apply Canny method. Due to the Canny method has noise filter, this technique is adequate to this application. As seen in Figure 4.5, Comparing with the other conventional filters such as Sobel and Prewitt, the results shown canny method provides the best complete for the edge structural. Therefore, Canny method is applied to all 3 cases in this research.

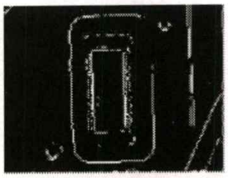
(a)



(b)



(c)


(d)


(e)

Figure 4.5 Gasket and PCB images: (a) Original image from CCD camera, (b) image by Canny method, (c) image by Sobel method, (d) image by Prewitt method, and (e) image by Roberts method

4.2.2 Gray scale 5 and 7 images cases with specific ROI preprocessing then apply the Canny method

We adopted the threshold value for cases of 5 and 7 images by calculating the average light intensity x any value in range [0.6-1.0]. Then the preprocessing in this case is the 3 dilation times to connect the points of the edge of the image due to the captured images are not completed structure.

4.3 Adopt PSO to generate the first population

In the proposed technique, following parameters are adopted as the particle in the optimization problem.

1. Standard deviation of Gaussian filter
2. Threshold value

The first batch of particles will be generated randomly.

4.4 Feature extraction

Feature extraction is applied to find the features and then identify the interesting object. Set the fitness function as

$$fitness = 0.001 \quad \text{If there is no specified object found}$$

$$= \sum_{i=1}^{N} Npixel_i \quad \text{If there is the specified object is found}$$

Where N = number of pixels on edge of the interesting object

Feature adopted to classify the specified object in this research work are

1. Width of gasket edge
2. Height of gasket edge
3. Area of gasket edge
4. Number of pixels on the perimeter of the object *If the maximum perimeter of the objects are in the range of 300-500 pixels* This states that the gasket is found

in the image. As seen in the fitness function, the proposed technique aims to maximize the number of pixels of gasket edges of testing images.

4.5 Adopt the PSO to find the next particle

Based on the evaluated fitness value in 4.4, PSO is applied to find the next particles until the best fitness or stopping criteria are achieved. PSO is a potential algorithm to solve an optimization problem. This approach is a robust stochastic optimization technique based on the movement and intelligence of swarms and applies the concept of social interaction to problem solving. The algorithm diagram is shown in Figure 4.6. Based on the above described approach, the optimization problem can be formulated as:

Find the threshold value and Sigma of Gaussian filter such that the objects are found on the images and the number of the pixels on the edges are maximized.

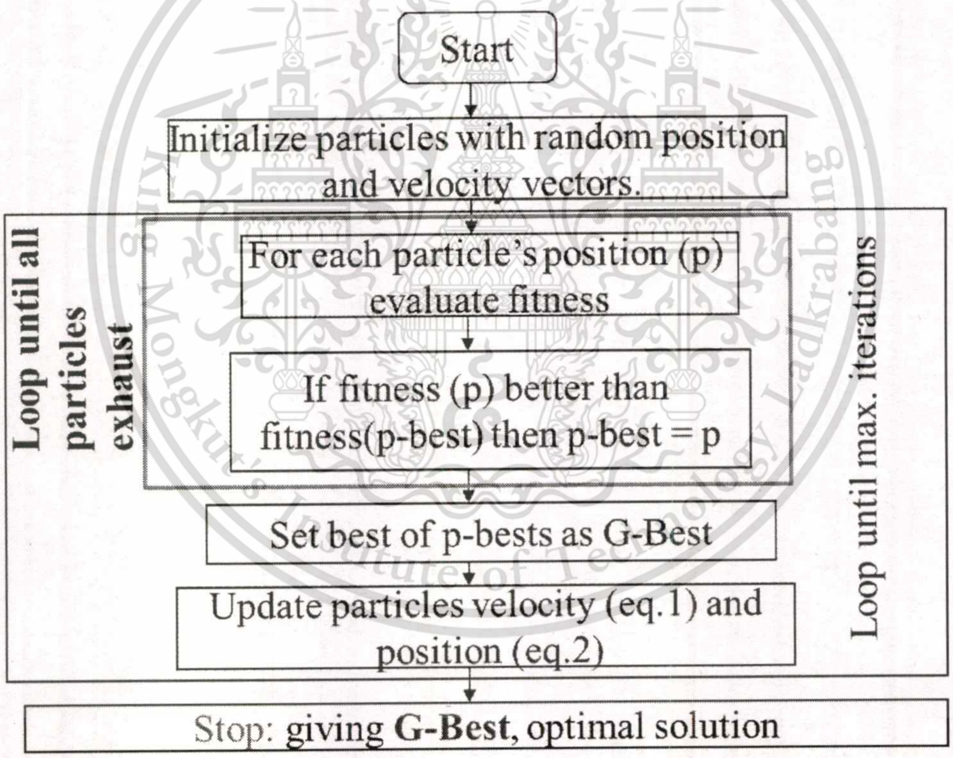


Figure 4.6 PSO algorithms to find the next particles

In addition, the testing images are the images with various brightness. The preparation of these images can be done by adjusting the light intensity of the machine vision. Thus, the optimal parameters found by the PSO is not optimal only the single light intensity, but also for various intensities which we can call the proposed algorithms as a robust edge detection.

CHAPTER 5
Experimental Results

5.1 Results and Discussion

The simulations are set to verify the effectiveness of finding the gasket edge on the original images with a variety of light intensity. These results are used in the inspection process in order to screen the interesting components, identify the missing gasket before Drive process. The devices used in these machine vision are CCD 5 Megapixels camera, CCTV lens 12.5 mm with full resolution 2456 x 2056 capturing. Pixel size of the machine vision is 3.45 x 3.45 μm. and MATLAB tool is used for adopting the PSO and image processing to find the optimal parameters. Then the optimal parameters can be used in the real machine vision. In the proposed experiments, the PSO was adopted to find the optimal threshold and sigma in Canny edge detection filter to make the image processing has robustness against the light intensity variation. There are three cases of experiments, those are RGB training images, Grayscale images with 5 training images, Grayscale images with 7 training images. In this case, the word "training images" states the images with varying light intensity which are used in the optimization problem.

5.1.1 PSO results of RGB images

PSO parameters were used as

Selection function = fitness in Chapter 4

Number of images = 7

Number of population = 6

Maximum iterations = 30

Acceleration constants = 2.1

Lower inertia weights = 0.6

Upper inertia weights = 0.9

Upper and lower bounds of Threshold = [0.1, 0.9]

Upper and lower bounds of Sigma = [0.1, 1.5]

The convergence of solution can be observed by the curve in Figure 5.1. In this figure, the optimal solution is obtained when running the PSO for 7 iterations.

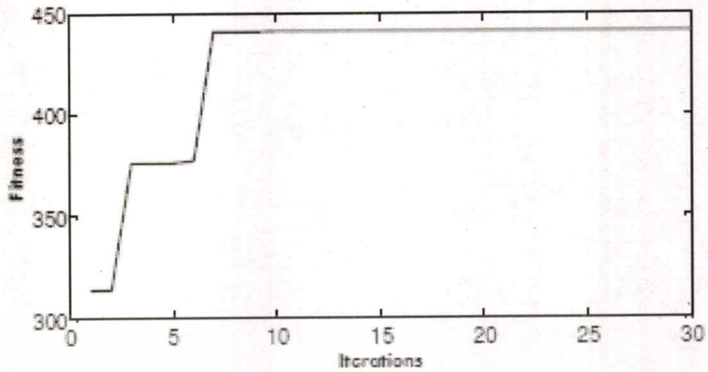


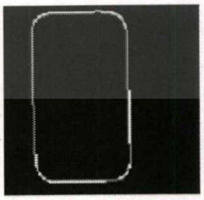
Figure 5.1 Fitness value versus iterations of RGB images

The optimal values of threshold and sigma evaluated by the PSO are 0.3 and 0.859, respectively. When applying these parameters to the Canny filter, the resulting edge images can be obtained as shown in Figure 5.2 (a) - (g).

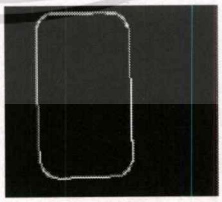
(a)

(b)

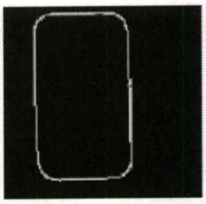
(c)



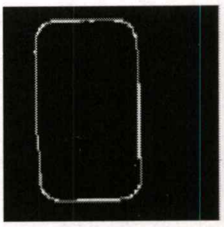
(d)



(e)



(f)



(g)

Figure 5.2 The edge images from the Canny filter with the optimal values determined by the PSO when the light intensity is (a) normal (b) lower intensity 1 (c) lower intensity 2 (d) lower intensity 3 (e) higher intensity 1 (f) higher intensity 2 and (g) higher intensity 3

To show the effectiveness of the proposed algorithm, the performance of conventional edge detection filter design which assigning the filter parameter by considering only one image is investigated. Based on the Figure 5.2 (a), the threshold and sigma values are properly selected as 0.4 and 0.45, respectively by trial and error method. The edge image obtained from these values is almost the same as those by the proposed algorithm. Figure 5.3 (a) shows the edge image of Figure 5.2(a). However, when applying this conventional filter to the other images (Figure 5.3(b) to 5.3(g)), some edge images lack of the interesting object. This means that the conventional filter fails to capture the interesting object due to the effect of variation of the light intensity.

(a)

(b)

(c)

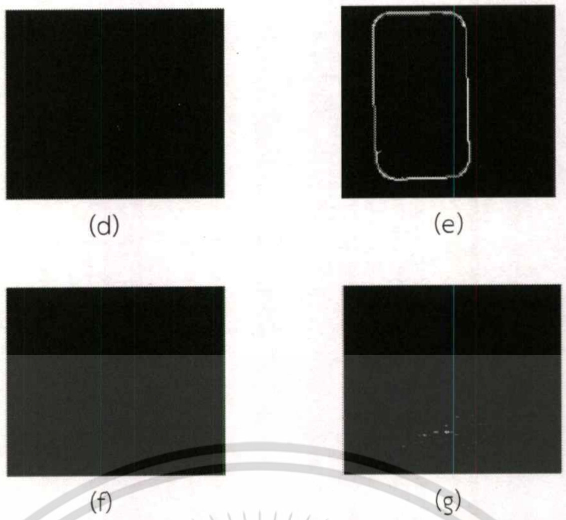


Figure 5.3 The edge images from the Canny filter with parameters by trial and error method when the light intensity is (a) normal (b) lower intensity 1 (c) lower intensity 2 (d) lower intensity 3 (e) higher intensity 1 (f) higher intensity 2 and (g) higher intensity 3

5.1.2 PSO results of gray scale 5 images with specific ROI

Images preprocessing by calculating the average light intensity x 0.9. Then the preprocessing in this case is the 3 dilation times to connect the points of the edge of the image.

PSO parameters were used as

Selection function = fitness as in Chapter 3

Number of images = 5

Number of population = 30

Maximum iterations = 30

Acceleration constants = 2.1

Lower inertia weights = 0.6

Upper inertia weights = 0.9

Canny filter boundary: threshold = [0.1, 0.9] and Standard deviation = [0.1, 1.5]

The convergence of solution can be observed by the curve in Figure 5.4. In this figure, the optimal solution is obtained when running the PSO for 30 iterations.

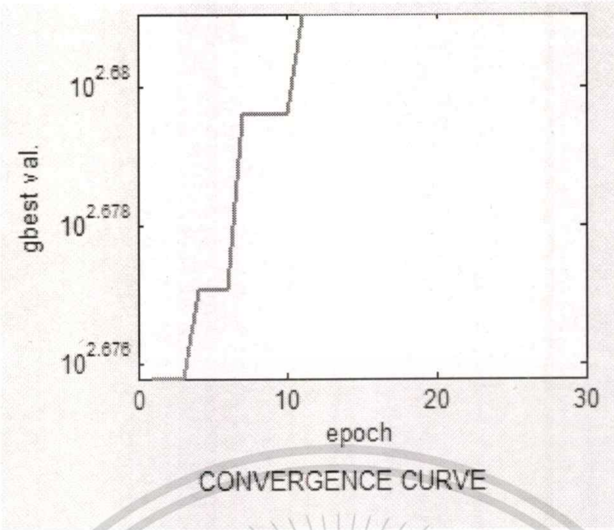
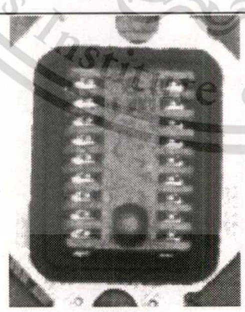
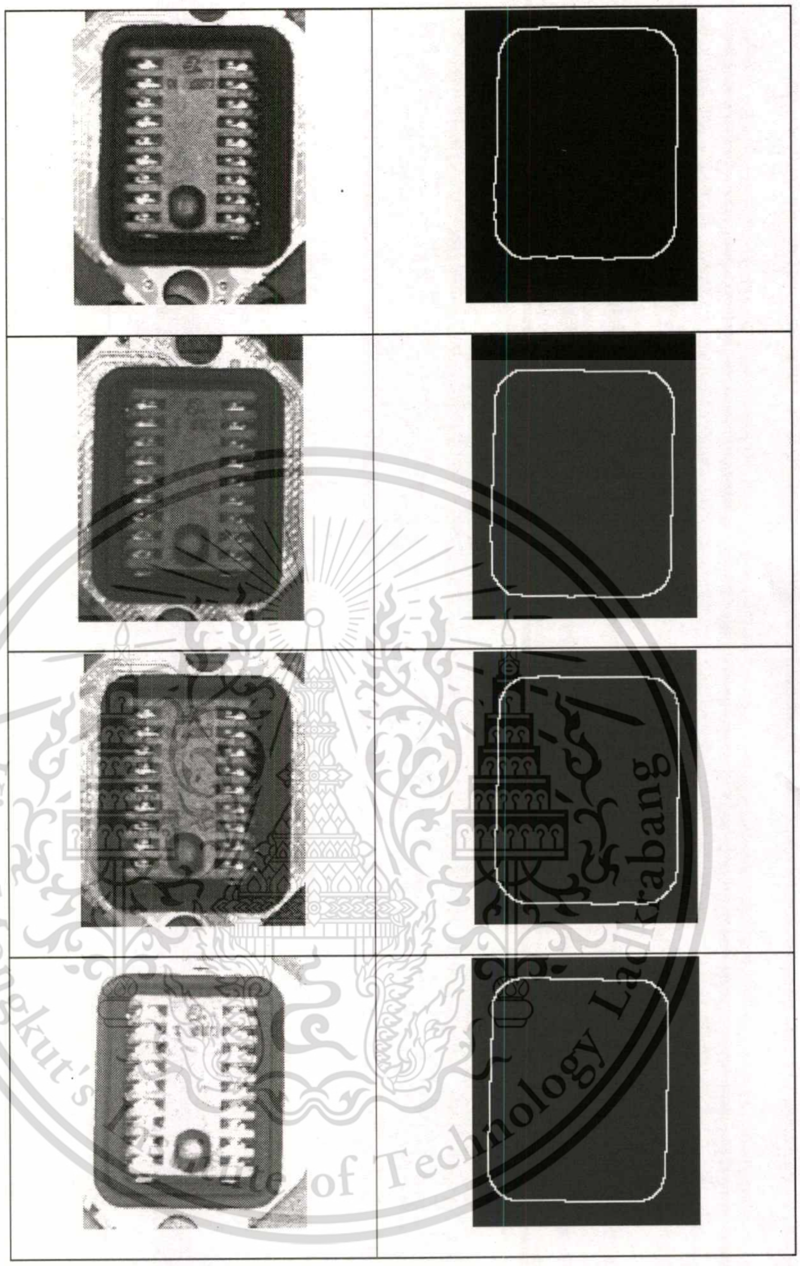


Figure 5.4 Fitness value versus iterations of gray scale images with specific ROI

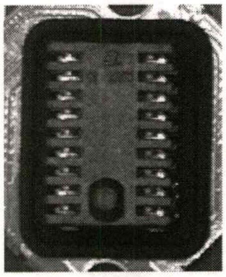
We found that the optimal fitness value is 479.8 and the optimal values of threshold and sigma evaluated by the PSO are 0.867 and 0.987, respectively. When applying these parameters to the Canny filter, the resulting edge images can be obtained as shown in Table 5.1.

Table 5.1 the edge images of gray scale with specific ROI from the Canny filter with the optimal values determined by the PSO

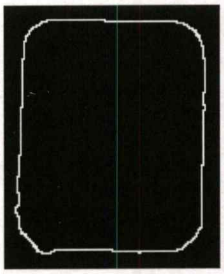
Original image	Image from PSO optimal values
	



To show the effectiveness of the proposed algorithm, we applied the optimal filter designed by the proposed technique to the image with the light intensity differed from the training images. As seen in Figure 5.5, the resulting edge images still be obtained.



(a)

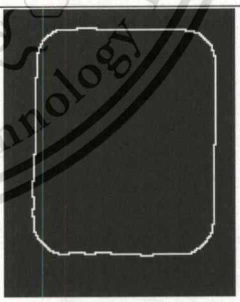


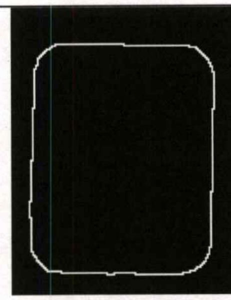
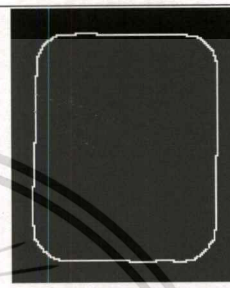
(b)

Figure 5.5 the edge images of gray scale with specific ROI (a) the original image with light intensity variation, (b) edge image of (a) from the Canny filter with the optimal values determined by the PSO

To define the boundary of multiple value for image processing, we simulated the proposed algorithm to images which preprocessing by calculating the average light intensity x interval values 0.5-1.0 (the value can not be more than 1.0). The examples of results can be shown in Table 5.2.

Table 5.2 the edge images of original grayscale preprocessing by calculating the average light intensity x interval values [0.5-1.0] obtained from the Canny filter with the optimal values determined by the PSO

Values	Image from PSO optimal values
1.0	

0.7	
0.6	
0.5	

From the results, we can define the interval values which used for gasket grayscale images processing as [0.6 – 1.0].

5.1.3 PSO results of original gray scale 7 images

Images preprocessing by calculating the average light intensity x 0.9. Then the preprocessing in this case is the 3 dilation times to connect the points of the edge of the image.

PSO parameters were used as

Selection function = the fitness function in Chapter 3

Number of images = 7

Number of population = 30

Maximum iterations = 30

Acceleration constants = 2.1

Lower inertia weights = 0.6

Upper inertia weights = 0.9

Canny filter boundary: threshold = [0.1, 0.9] and Standard deviation = [0.1, 1.5]

The convergence of solution can be observed by the curve in Figure 5.6. In this figure, the optimal solution is obtained when running the PSO for 30 iterations.

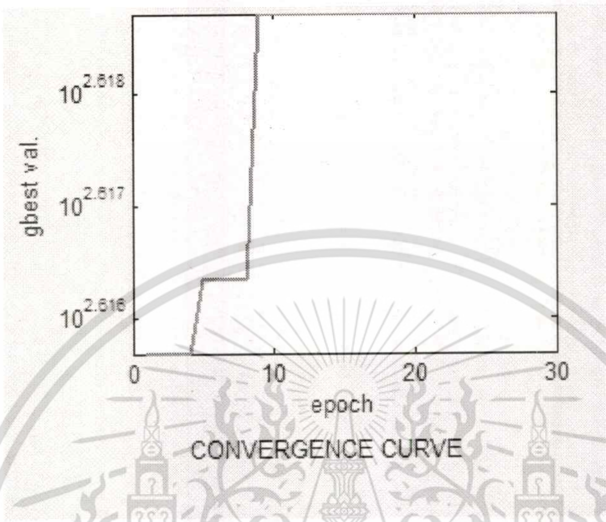
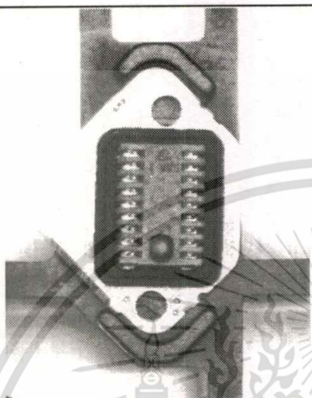
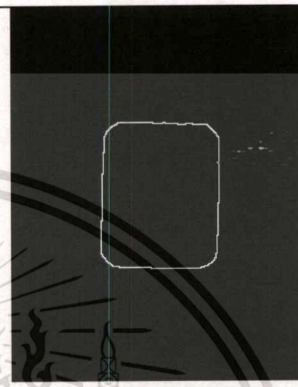
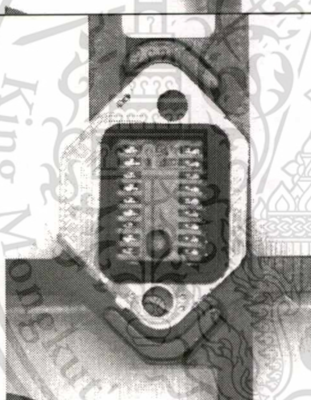
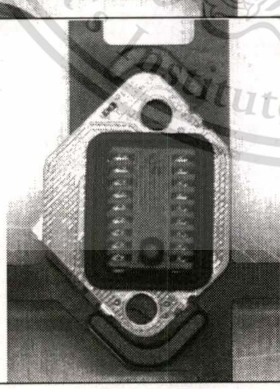
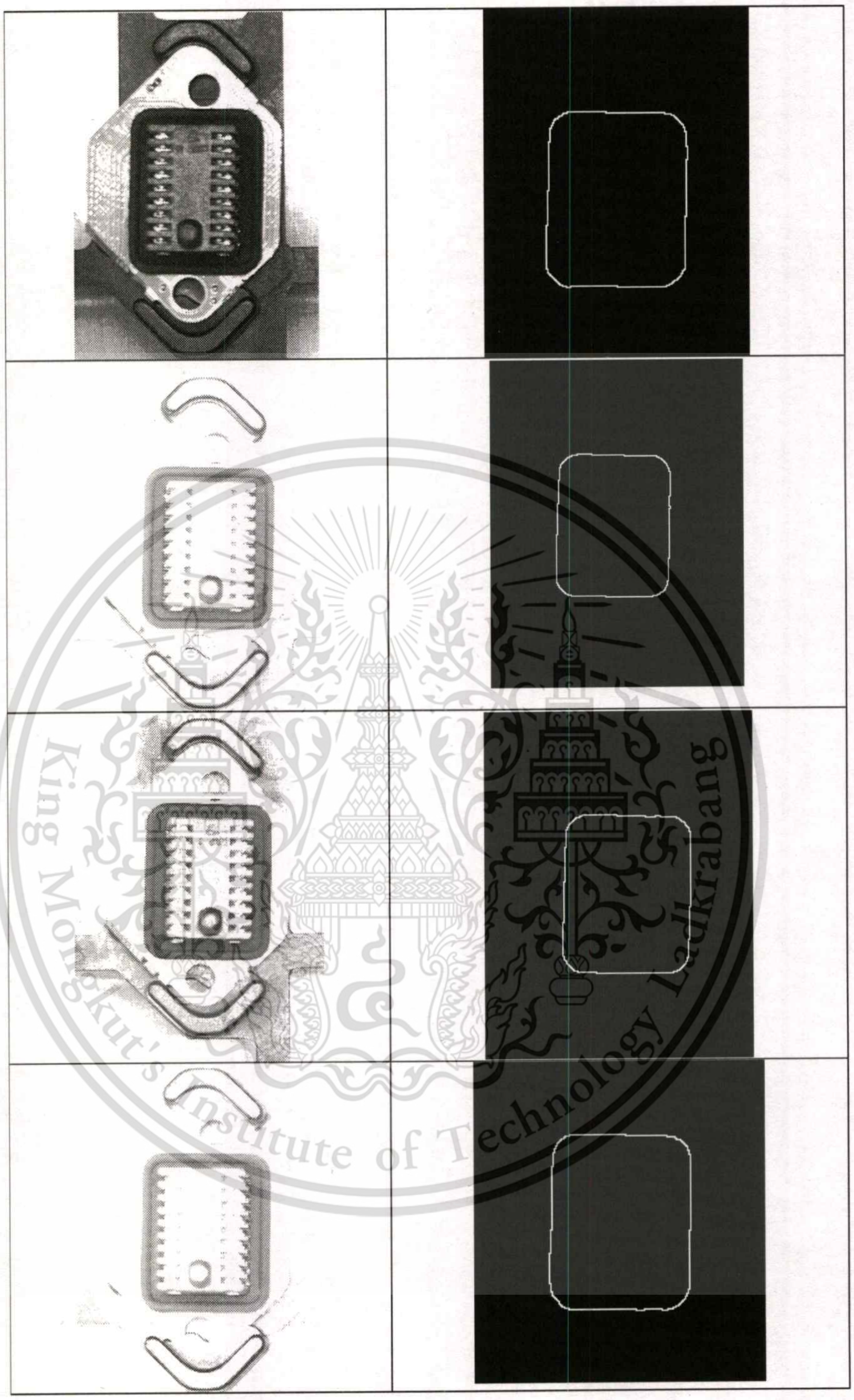


Figure 5.6 Fitness value versus iterations of original gray scale images

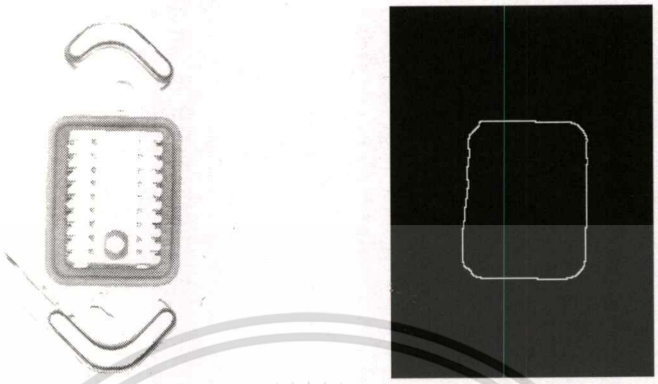
We found that the optimal fitness value is 415.625 and the optimal values of threshold and sigma evaluated by the PSO are 0.6 and 0.99, respectively. When applying these parameters to the Canny filter, the resulting edge images can be obtained as shown in Table 5.3.

Table 5.3 the edge images of original grayscale obtained from the Canny filter with the optimal values determined by the PSO

Original image	Image from PSO optimal values
	
	
	



To show the effectiveness of the proposed algorithm, we applied the optimal filter designed by the proposed technique to the image with the light intensity differed from the training images. As seen in Figure 5.7, the resulting edge images still be obtained.



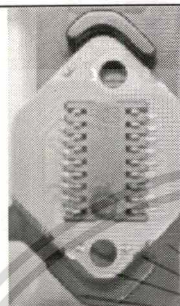
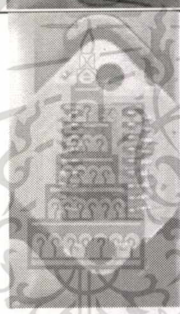
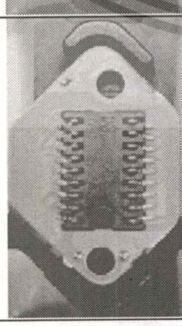
(a) (b)

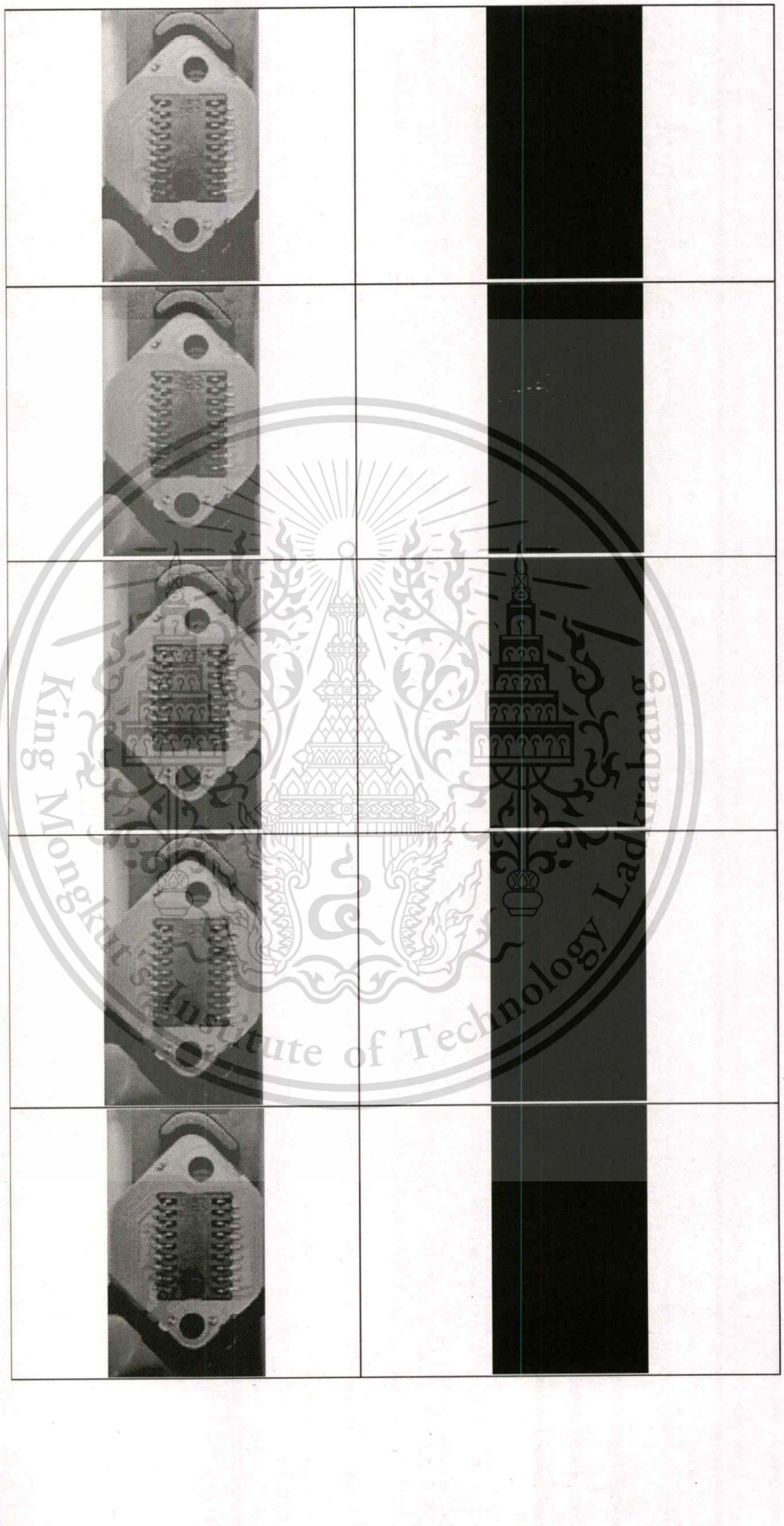
Figure 5.7 the edge images of original gray scale (a) the original image with light intensity variation, (b) edge image of (a) from the Canny filter with the optimal values determined by the PSO

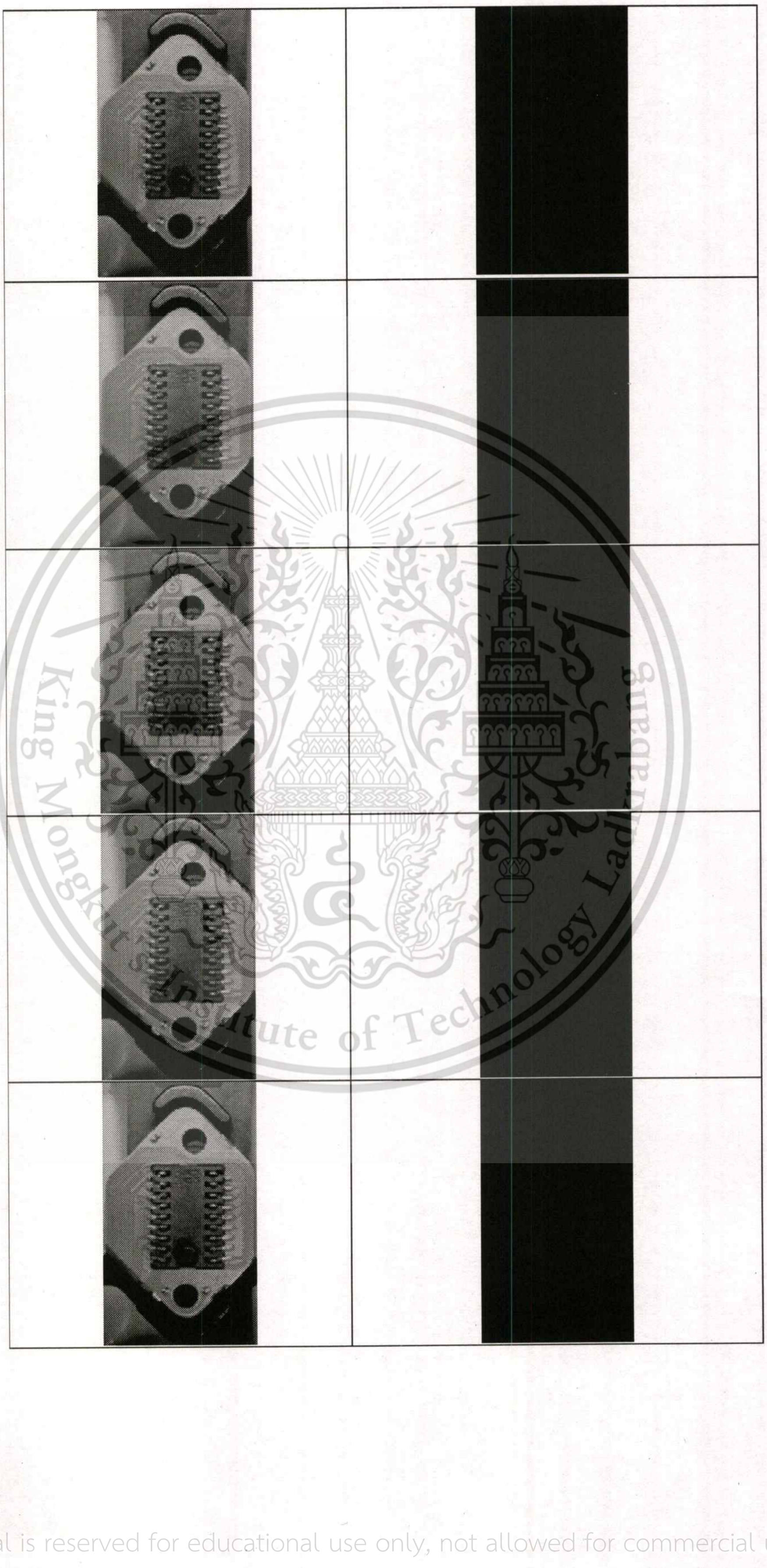
As shown in the experimental results of the proposed technique, it can prove that by using the proposed algorithms, the gasket edge can be classified from PCB clearly and correctly even under the variation in light intensity. The region of interest can be precisely shown by the proposed technique.

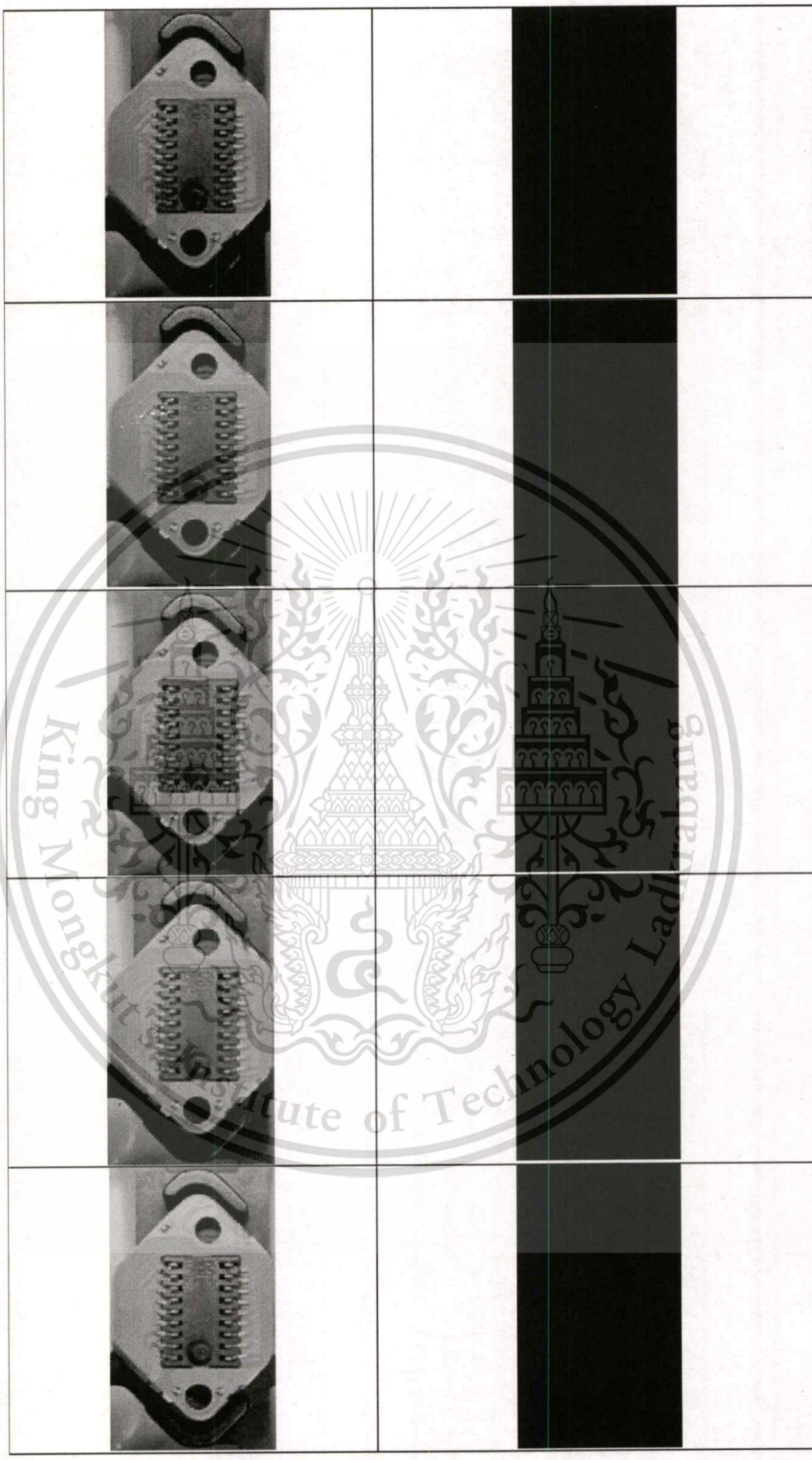
However, the application will be implemented to identify good or not good output or support the automatic inspection in manufacturing process. The result must be indicated to guarantee the quality of HSA (Head Stack Assembly) product. Therefore, to show the reliable of the proposed algorithm that will not cause the production accept the missing gasket condition, we simulated the proposed algorithm to uninstalled gasket with 20 variation of light intensity images. The results are shown in Table 5.4.

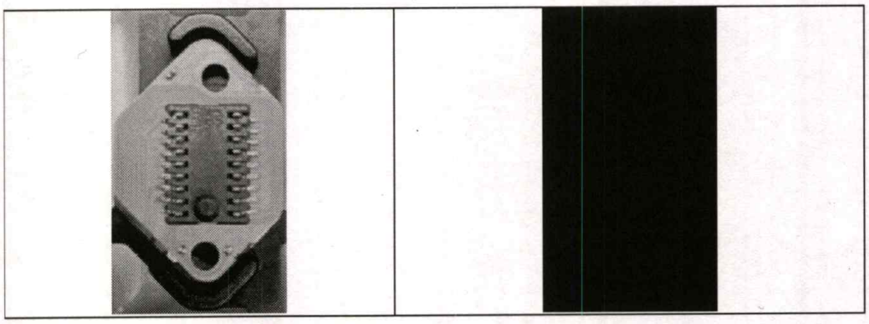
Table 5.4 the edge images of uninstalled gasket images obtained from the Canny filter with the optimal values determined by the PSO

Original image	Image from PSO optimal values
	
	
	
	









The results show that the gasket edge is not found from the proposed algorithm simulation. Thus, this detection is reliable for the automatic visual inspection that can be implemented in the manufacturing process.

Experimental results can confirm the robustness against light intensity variation of the proposed filter. Hence, the algorithm can be considered for using in an automatic visual inspection. In contrary, the conventional filter which the parameters were selected by considering one image performs a good detection in the normal image. However, the conventional filter may fail to capture the object when the light intensity is changed far from that of the original image.

CHAPTER 6
Conclusion

The new edge detection algorithm is developed to inspect the gasket by focusing on the gasket edge detection of PCB image. Various image processing techniques are reviewed, i.e. Sobel algorithm, Prewitt algorithm, Roberts algorithm. Among them, Canny method is selected to identify the true edge of gasket from the image background, feature extraction is used to help finding feature of edge detected. In addition, Particle Swarm Optimization (PSO) algorithm is applied to find optimal parameters until the cost function is achieved. Moreover, the image preprocessing method is used to help for image processing as well. The wide ranges of light intensity images are captured from actual environment and then be used in the optimization process. This leads to the resulting filter is robust against the light intensity variance. The proposed technique can be adopted to the inspection process of gasket which the missing gasket is found in Drive assemble process that also impacts to product quality in HDD manufacturing.

At the present, it takes relatively long cycle time to get the result of the inspection and inaccurate data occurs due to human error and fatigue from eyes-base. Moreover, human error is an unpredicted factor in inspection's result.

As shown in the experimental results, it can be seen that the proposed algorithm is capable to identify gasket edge correctly from the image. It also gives an acceptable feature on the image as well. In addition, verification on RGB images, 5 grayscale images with specified ROI and 7 5 grayscale images without specifying ROI concern that the proposed technique is effective. In the RGB images experiments, the results also show that conventional algorithm is not capable to classify the edge of gasket with high variation of light intensity.

Moreover, the proposed technique is reliable to implement in manufacturing process due to the detection cannot found the gasket edge from uninstalled gasket condition that will be prevent the "Accept bad quality" condition.

For this gasket detection algorithm, the PSO is the potential technique to support optimal parameters finding. Compared with the other optimization techniques such as GA (Genetic algorithm) and Neural network the PSO has less complication of algorithm and less parameters to adjust. One of the advantages of this technique is PSO takes real numbers as particles that leads to less time processing and simply to implement. Therefore, PSO is the proper method for gasket detection optimization problem.

In conclusion, this research is aimed to inspect the missing gasket by applying vision technique with the proposed image filtering. The PSO is adopted to find the proper parameters in the filter mask so that the filter can be applied functionary even in the situation of light intensity variation. Comparison results between the proposed and conventional technique verify that the proposed technique can enhance the performance and reliability of the inspection process for this application scenario.

REFERENCES

[1] A. Saenthon, S. Kaitvanidvilai "Development of new edge detection filter based on genetic algorithm: an application to a soldering joint inspection", Springer-Varlag London Limited, 2009.

[2] A. Saenthon, S. Kaitvanidvilai, "Automatic visual inspection of bump in hard disk drive component using neural network and image processing", AU Journal of Technology, 2011. pp. 188-195.

[3] S.M. Bhandarkar, Y. Zhang, W.D. Potter, "A Genetic Algorithm-based Edge Detection Technique", International Joint Conference on Neural Networks, 1993. pp. 2995-2999.

[4] O.P. Verma, R. Sharma, "An Optimal Edge Detection using Universal Law of Gravity and Ant Colony Algorithm", World Congress on Information and Communication Technologies, 2011, pp. 507-511.

[5] F.M.A. Mosen, M.M. Hadhoud, K. Amin, "A new Optimization-Based Image Segmentation method by Particle Swarm Opmization", International Journal of Advanced Computer Science and Applications (IJACSA), Special Issues on Image Processing and Analysis, 2011, pp. 10-18.

[6] F.M.A. Mosen, M.M. Hadhoud, K. Amin, "A New Image Segmentation Method Based on Particle Swarm Optimization", The International Arab Journal of Information Technology, Vol. 9, No. 5, 2012, pp. 487-493.

[7] J. Amamaheswari, G. Radhamani, "An Optimal Approach for Medical Image Analysis", American Journal of Bioinformatics 1 (1), 2012, pp. 64-69.

[8] P.D. Sathya, R. Kayalvizhi, "PSO-Based Tsallis Thresholding Selection Procedure for Image Segmentation", International Journal of Computer Applications (0975-8887), Vol. 5, No. 4, 2010, pp. 39-46.

[9] Pitas, Ioannis, *Digital Image Processing Algorithms and Applications*, 1st edition, John Wiley and Sons, 1993.

[10] McAndrew, Alasdair, *Introduction to Digital Image Processing*, 3rd, 2004.

[11] S.E. Umbaugh, *Digital Image Analysis and Processing*, 2005.

[12] Wang, F. Yue, *Advance in Computational Intelligence: Theory and Application*, 2006.

[13] Q. Bai, "Analysis of Particle Swarm Optimization Algorithm", Computer and Information Science (CCSE), Vol. 3, No. 1, 2010, Pp. 180-183.

APPENDIX

HSA Gasket Edge Detection Algorithm

```
% f6.m

% Schaffer's F6 function

% commonly used to test optimization/global minimization problems

%

% z = 0.5+ (sin^2(sqrt(x^2+y^2))-0.5)/((1+0.01*(x^2+y^2))^2)

function [out]=f6(in)

  x=in(:,1); %threshold

  y=in(:,2);

sumtemp=0;

temp=0;

for i=1:size(x,1)

sumtemp=0;

temp=0;

boundL=435;

boundH=445;

%neutral

I = imread('D:\normal.jpg');

L = rgb2gray(I);

v=edge(L,'canny',x(i),y(i));

CC = bwconncomp(v);

numPixels = cellfun(@numel,CC.PixelIdxList);
```

```
[biggest,idx] = max(numPixels);
CC = bwconncomp(v);
numPixels = cellfun(@numel,CC.PixelIdxList);
[biggest,idx] = max(numPixels);
X = regionprops(CC,'Area');
for j=1:size(X,1)
    if X(j).Area > boundL && X(j).Area < boundH
    else
        v(CC.PixelIdxList{j}) = 0;
    end;
end;
CC1 = bwconncomp(v);
numPixels = cellfun(@numel,CC1.PixelIdxList);
[biggest,idx] = max(numPixels);
[a,b]=size(biggest);
if (a ==0) || (b==0)
   temp=1;
else
   temp=biggest;
end
sumtemp=sumtemp+temp;
%Low1
L1 = imread('D:\lowb1.jpg');
Low1 = rgb2gray(L1);
v=edge(Low1,'canny',x(i),y(i));
```

```
CC = bwconncomp(v);
numPixels = cellfun(@numel,CC.PixelIdxList);
[biggest,idx] = max(numPixels);
CC = bwconncomp(v);
numPixels = cellfun(@numel,CC.PixelIdxList);
[biggest,idx] = max(numPixels);
X = regionprops(CC,'Area');
for j=1:size(X,1)
    if X(j).Area > boundL && X(j).Area < boundH

    else
        v(CC.PixelIdxList{j}) = 0;
    end;
end;
CC1 = bwconncomp(v);
numPixels = cellfun(@numel,CC1.PixelIdxList);
[biggest,idx] = max(numPixels);
[a,b]=size(biggest);
if (a ==0) || (b==0)
   temp=1;
else
   temp=biggest;
end
sumtemp=sumtemp+temp;
%Low2
```

```
L1 = imread('D:\lowb2.jpg');
Low1 = rgb2gray(L1);
v=edge(Low1,'canny',x(i),y(i));
CC = bwconncomp(v);
numPixels = cellfun(@numel,CC.PixelIdxList);
[biggest,idx] = max(numPixels);
CC = bwconncomp(v);
numPixels = cellfun(@numel,CC.PixelIdxList);
[biggest,idx] = max(numPixels);
X = regionprops(CC,'Area');
for j=1:size(X,1)
    if X(j).Area > boundL && X(j).Area < boundH
    else
        v(CC.PixelIdxList{j}) = 0;
    end;
end;
CC1 = bwconncomp(v);
numPixels = cellfun(@numel,CC1.PixelIdxList);
[biggest,idx] = max(numPixels);
[a,b]=size(biggest);
if (a ==0) || (b==0)
   temp=1;
else
   temp=biggest;
end
```

```
sumtemp=sumtemp+temp;
%Low1
L1 = imread('D:\lowb3.jpg');
Low1 = rgb2gray(L1);
v=edge(Low1,'canny',x(i),y(i));
CC = bwconncomp(v);
numPixels = cellfun(@numel,CC.PixelIdxList);
[biggest,idx] = max(numPixels);
CC = bwconncomp(v);
numPixels = cellfun(@numel,CC.PixelIdxList);
[biggest,idx] = max(numPixels);
X = regionprops(CC,'Area');
for j=1:size(X,1)
    if X(j).Area > boundL && X(j).Area < boundH
    else
        v(CC.PixelIdxList{j}) = 0;
    end;
end;
CC1 = bwconncomp(v);
numPixels = cellfun(@numel,CC1.PixelIdxList);
[biggest,idx] = max(numPixels);
[a,b]=size(biggest);
if (a ==0) || (b==0)
   temp=1;
else
```

```
  temp=biggest;
end
sumtemp=sumtemp+temp;
%high1
H1 = imread('D:\highb1.jpg');
High1 = rgb2gray(H1);
v=edge(High1,'canny',x(i),y(i));
CC = bwconncomp(v);
numPixels = cellfun(@numel,CC.PixelIdxList);
[biggest,idx] = max(numPixels);
CC = bwconncomp(v);
numPixels = cellfun(@numel,CC.PixelIdxList);
[biggest,idx] = max(numPixels);
X = regionprops(CC,'Area');
for j=1:size(X,1)
   if X(j).Area > boundL && X(j).Area < boundH
   else
      v(CC.PixelIdxList{j}) = 0;
   end;
end;
CC1 = bwconncomp(v);
numPixels = cellfun(@numel,CC1.PixelIdxList);
[biggest,idx] = max(numPixels);
[a,b]=size(biggest);
if (a ==0) || (b==0)
```

```
  temp=1;
else
  temp=biggest;
end
sumtemp=sumtemp+temp;
%high1
H1 = imread('D:\highb2.jpg');
High1 = rgb2gray(H1);
v=edge(High1,'canny',x(i),y(i));
CC = bwconncomp(v);
numPixels = cellfun(@numel,CC.PixelIdxList);
[biggest,idx] = max(numPixels);
CC = bwconncomp(v);
numPixels = cellfun(@numel,CC.PixelIdxList);
[biggest,idx] = max(numPixels);
X = regionprops(CC,'Area');
for j=1:size(X,1)
   if X(j).Area > boundL && X(j).Area < boundH
   else
      v(CC.PixelIdxList{j}) = 0;
   end;
end;
CC1 = bwconncomp(v);
numPixels = cellfun(@numel,CC1.PixelIdxList);
[biggest,idx] = max(numPixels);
```

```
[a,b]=size(biggest);
if (a ==0) || (b==0)
  temp=1;
else
  temp=biggest;
end
sumtemp=sumtemp+temp;
%high3
H1 = imread('D:\highb3.jpg');
High1 = rgb2gray(H1);
v=edge(High1,'canny',x(i),y(i));
CC = bwconncomp(v);
numPixels = cellfun(@numel,CC.PixelIdxList);
[biggest,idx] = max(numPixels);
CC = bwconncomp(v);
numPixels = cellfun(@numel,CC.PixelIdxList);
[biggest,idx] = max(numPixels);
X = regionprops(CC,'Area');
for j=1:size(X,1)
    if X(j).Area > boundL && X(j).Area < boundH

    else
        v(CC.PixelIdxList{j}) = 0;
    end;
end;
```

```
CC1 = bwconncomp(v);
numPixels = cellfun(@numel,CC1.PixelIdxList);
[biggest,idx] = max(numPixels);
[a,b]=size(biggest);
if (a ==0) || (b==0)
   temp=1;
else
   temp=biggest;
end
sumtemp=sumtemp+temp;
xx(i) = sumtemp/7;
end;
out=xx';

% demopsobehavior.m
% demo of the pso.m function
% the pso tries to find the minimum of the f6 function, a standard
% benchmark
%
% on the plots, blue is current position, green is Pbest, and red is Gbest
% Brian Birge
% Rev 3.0
% 2/27/06
clear all
close all
```

```
clc
help demopsobehavior
warning off
functnames = {'ackley','alpine','DeJong_f2','DeJong_f3','DeJong_f4',...
        'Foxhole','Griewank','NDparabola',...
        'Rastrigin','Rosenbrock','f6_1','f6mod','tripod',...
        'f6_bubbles_dyn','f6_linear_dyn','f6_spiral_dyn'};
disp('Static test functions, minima don''t change w.r.t. time/iteration:');
disp(' 1) Ackley');
disp(' 2) Alpine');
disp(' 3) DeJong_f2');
disp(' 4) DeJong_f3');
disp(' 5) DeJong_f4');
disp(' 6) Foxhole');
disp(' 7) Griewank');
disp(' 8) NDparabola (for this demo N = 2)');
disp(' 9) Rastrigin');
disp('10) Rosenbrock');
disp('11) Schaffer f6');
disp('12) Schaffer f6 modified (5 f6 functions translated from each other)');
disp('13) Tripod');
disp(' ');
disp('Dynamic test functions, minima/environment evolves over time/iteration:');
disp('14) f6_bubbles_dyn');
disp('15) f6_linear_dyn');
```

```
disp('16) f6_spiral_dyn');
%functchc=input('Choose test function ? ');
functchc=11;
functname = functnames{functchc};
disp(' ');
disp('1) Intense graphics, shows error topology and surfing particles');
disp('2) Default PSO graphing, shows error trend and particle dynamics');
disp('3) no plot, only final output shown, fastest');
%plotfcn=input('Choose plotting function ? ');
plotfcn=2;
if plotfcn == 1
   plotfcn = 'goplotpso4demo';
   shw     = 1;   % how often to update display
elseif plotfcn == 2
   plotfcn = 'goplotpso';
   shw     = 1;   % how often to update display
else
   plotfcn = 'goplotpso';
   shw     = 0;   % how often to update display
end
% set flag for 'dynamic function on', only used at very end for tracking plots
dyn_on = 0;
if functchc==15 | functchc == 16 | functchc == 17
   dyn_on = 1;
end
```

```
%xrng=input('Input search range for X, e.g. [-10,10] ? ');
%yrng=input('Input search range for Y ? ');
%xrng=[0.3,0.4];
%yrng=[0.4,0.5];
xrng=[0.3, 0.9];
yrng=[0.2, 0.9];
disp(' ');
% if =0 then we look for minimum, =1 then max
  disp('0) Minimize')
  disp('1) Maximize')
 % minmax=input('Choose search goal ?');
 minmax=1;
 % minmax=0;
  disp(' ');
  %mvden = input('Max velocity divisor (2 is a good choice) ? ');
  mvden =2;
  disp(' ');
  %ps    = input('How many particles (24 - 30 is common)? ');
  ps = 5;
  disp(' ');
  disp('0) Common PSO - with inertia');
  disp('1) Trelea model 1');
  disp('2) Trelea model 2');
  disp('3) Clerc Type 1" - with constriction');
  %modl  = input('Choose PSO model ? ');
```

```
modl  =0;
% note: if errgoal=NaN then unconstrained min or max is performed
if minmax==1
   %  errgoal=0.97643183; % max for f6 function (close enough for termination)
    errgoal=NaN;
else
   % errgoal=0; % min
    errgoal=NaN;
end
minx = xrng(1);
maxx = xrng(2);
miny = yrng(1);
maxy = yrng(2);
%---------------------------------------------------------------------------
dims=2;
varrange=[];
mv=[];
for i=1:dims
    varrange=[varrange;minx maxx];
    mv=[mv;(varrange(i,2)-varrange(i,1))/mvden];
end
ac      = [2.1,2.1];% acceleration constants, only used for modl=0
Iwt     = [0.9,0.6];  % intertia weights, only used for modl=0
epoch   = 30; % max iterations
wt_end  = 100; % iterations it takes to go from Iwt(1) to Iwt(2), only for modl=0
```

```
errgrad = 1e-99;   % lowest error gradient tolerance

errgraditer=100; % max # of epochs without error change >= errgrad

PSOseed = 0;    % if=1 then can input particle starting positions, if= 0 then all
random

% starting particle positions (first 20 at zero, just for an example)

 PSOseedValue = repmat([0],ps-10,1);

psoparams=...

 [shw epoch ps ac(1) ac(2) Iwt(1) Iwt(2) ...

  wt_end errgrad errgraditer errgoal modl PSOseed];

% run pso

% vectorized version

[pso_out,tr,te]=pso_Trelea_vectorized(functname, dims,...

    mv, varrange, minmax, psoparams,plotfcn,PSOseedValue);

%----------------------------------------------------------------------

% display best params, this only makes sense for static functions, for dynamic

% you'd want to see a time history of expected versus optimized global best

% values.

disp(' ');

disp(' ');

disp(['Best fit parameters: ']);

disp([' cost = ',functname,'( [ input1, input2 ] )']);

disp(['---------------------------']);

disp(['     input1 = ',num2str(pso_out(1))]);

disp(['     input2 = ',num2str(pso_out(2))]);

disp(['       cost = ',num2str(pso_out(3))]);
```

```
disp(['   mean cost = ',num2str(mean(te))]);
disp([' # of epochs = ',num2str(tr(end))]);
%% optional, save picture
%set(gcf,'InvertHardcopy','off');
%print -dmeta
%print('-djpeg',['demoPSOBehavior.jpg']);
I = imread('D:\normal.jpg');
L = rgb2gray(I);
v=edge(L,'canny',pso_out(1),pso_out(2));
%imshow(v);
%r=bwlabel(v);
%stats = imfeature(r,'all');
%idx = find([stats.Area] > 210);
%BW2 = ismember(r,idx);
CC = bwconncomp(v);
numPixels = cellfun(@numel,CC.PixelIdxList);
[biggest,idx] = max(numPixels);
%to show the gasket
BW1 = zeros(size(v));
BW1(CC.PixelIdxList{idx}) = 1;
figure, imshow(BW1);
```

Author's Biography

Name - Surname: Ms. Thunyatip Vipasetharat
Date of Birth: November 28'1985
Place of Birth: Bangkok, Thailand
Education: 2008 Bachelor of Mechatronics Engineering, Production Engineering Department, King Mongkut's University of Technology Thonburi.

Work Experience:
2010 - Current Tooling Replication Engineer, Seagate Technology (Thailand) Ltd.
2008 - 2010 Design Engineer, Nissan Technical Center South East Asia (Thailand).

Publication:
Thunyatip Vipasetharat, Somyot Kaitwanidvilai, "Robust Edge Detection Design for HSA Missing Gasket Inspection using Particle Swarm Optimization", iEECON 2014.