Smart image processing for steel bridge corrosion inspection

Yang, Ya-ching

2010

Yang, Y.-C. (2010). Smart image processing for steel bridge corrosion inspection. Master's thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/46735>

<https://doi.org/10.32657/10356/46735>

Nanyang Technological University

Downloaded on 26 Apr 2023 21:31:10 SGT

SMART IMAGE PROCESSING FOR STEEL BRIDGE CORROSION INSPECTION

YANG, YA-CHING

**SCHOOL OF CIVIL AND ENVIRONMENTAL
ENGINEERING**

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Master of Engineering

2010

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor, Professor Chen, Po-Han and Professor Chang, Luh-Maan, for their support and advice. Their kind guidance stimulated my interest in doing research and made this work possible.

The following list is our teamwork.

Journal Paper:

- [1] Po-Han Che, Ya-Ching Yang, and Luh-Maan Chang (2009), "Automated bridge coating defect recognition using adaptive ellipse approach", *Automation in Construction*, Volume 18, Issue 5, August 2009, Pages 632-643.
- [2] Po-Han Che, Ya-Ching Yang, and Luh-Maan Chang (2009), "Box-and-Ellipse-Based ANFIS (BE-ANFIS) for Bridge Coating Assessment", *Journal of Computing in Civil Engineering*, ASCE 2009, in press.
- [3] Po-Han Che, Ya-Ching Yang, and Luh-Maan Chang (2009), "Illumination Adjustment for Bridge Coating Images Using BEMD-Morphology Approach (BMA)", *Automation in Construction*. (Accepted on 2009)

Conference Paper:

- [1] Chen, P.H., Yang, Y.C., and Chang, L.M. (2008). "Automated Bridge Coating Defect Recognition by Adaptive Ellipse Approach." *Proceedings (abstract proceedings with a CD-ROM) of the 11th East Asia-Pacific Conference on Structural Engineering and Construction (EASEC-11)*, November 19-21, 2008, Taipei, Taiwan, pp. 708-709. (Abstract)
- [2] Po-Han Che, Ya-Ching Yang, and Luh-Maan Chang, (2009) "Steel Bridge Rust Recognition – A Color-Image-Based AI Approach", Fifth International Conference on Construction in the 21st Century (CITC-V), May 2009, Istanbul, Turkey.
- [3] Chen, P.H., Yang, Y.C., and Chang, L.M. (2009). "Illumination Adjustment for Bridge Coating Images Using BEMD-Morphology Approach." *Proceedings of the 3rd International Conference on Construction Engineering and Management / 6th International Conference on Construction Project Management (ICCEM-ICCPM 2009)*, May 27-30, 2009, Jeju, Korea, pp. 224-229. **(This paper received the Excellent Paper Award in ICCEM-ICCPM 2009.)**
- [4] Chen, P.H., Yang, Y.C., and Chang, L.M. (2009). "Box-and-Ellipse-Based Neuro-Fuzzy Approach for Bridge Coating Assessment." *Proceedings of the 3rd*

International Conference on Construction Engineering and Management / 6th International Conference on Construction Project Management (ICCEM-ICCPM 2009), May 27-30, 2009, Jeju, Korea, pp. 257-262.

[5]Chen, P.H., Yang, Y.C., and Chang, L.M. (2009). "Adaptable Ellipse Method for Bridge Coating Defect Recognition." Proceedings of the 3rd International Conference on Construction Engineering and Management / 6th International Conference on Construction Project Management (ICCEM-ICCPM 2009), May 27-30, 2009, Jeju, Korea, pp. 449-456.

[6]Chen, P.H., Yang, Y.C., and Chang, L.M. (2009). "Box-and-Ellipse-based Neuro-Fuzzy Approach for Bridge Coating Assessment." Non-Destructive Testing in Civil Engineering, Nantes, France, July, 2009

Table of Content

ABSTRACT	III
CHAPTER 1 INTRODUCTION.....	1
1.1 Background.....	4
1.2 Problem Statement	4
1.3 Research Objectives	6
1.4 Organization of the Research.....	6
CHAPTER 2 LITERATURE REVIEW.....	8
2.1 Image Processing and Its Application in Infrastructure Condition Assessment 8	
2.1.1 Underground Pipeline	8
2.1.2 Steel bridge Surface Inspection	9
2.2 Color Features.....	10
2.2.1 Color Image.....	11
2.2.2 Color Space and Color Configuration.....	11
2.3 Color Image Segmentation	18
2.3.1 Feature Extraction.....	19
2.3.1.1 Color Similarity	19
2.3.1.2 Histogram	20
2.3.1.3 Entropy	21
2.3.1.4 Principal Component Analysis	21
2.3.2 Threshold Segmentation	24
2.3.3 Clustering Segmentation	25
2.3.3.1 K-Means Algorithm	25
2.3.3.2 Fuzzy C-Means Algorithm	27
2.4 Artificial Intelligence	29
2.4.1 Fuzzy Logic Technique.....	29
2.4.1.1 Fuzzy Sets.....	30
2.4.1.2 Membership Functions.....	31

2.4.1.3 Fuzzy Logical Operations	32
2.4.1.4 Defuzzification	33
2.4.1.5 Fuzzy Inference	35
2.4.2 Artificial Neural Network	35
2.4.2.1 Artificial Neurons	36
2.4.2.2 Learning Algorithm.....	37
2.4.2.3 Framework.....	38
2.4.3 Adaptive Network-Based Fuzzy Inference System (ANFIS)	39
2.4.3.1 Framework of ANFIS.....	40
2.4.3.2 Learning rule	42
2.4.3.3 Sugeno-type fuzzy inference system	45
2.5 Hilbert-Huang Transform.....	46
2.5.1 Hilbert-Huang Transform.....	46
2.5.2 Bidimensional Empirical Mode Decomposition (BEMD).....	48
2.6 Morphological Image Processing.....	52
2.6.1 Dilation and Erosion	52
2.6.2 Opening, Closing, and TopHat Transform.....	53
CHAPTER 3 COMPARISON OF COLOR CONFIGURATION USING THE K-MEANS ALGORITHM.....	56
3.1 Characteristic of Rust Images	56
3.1.1 Definition of a rust color	56
3.1.2 Why not K-Means.....	58
3.1.3 Principal Component Features	60
3.2 Artificial Rust Image	62
3.3 Evaluation the Best Color Feature for Rust Images.....	64
3.3.1 Uniform Illumination	65
3.3.2 Non-uniform Illumination	66
3.3.3 Summary	67
3.4 Test the Power of $L^*a^*b^*$	67
3.5 Summary of Chapter	70

CHAPTER 4 ADAPTIVE ELLIPSE APPROACH (AEA)	71
4.1 Rust Image Preprocessing	71
4.1.1 Automated Detection of Background	71
4.1.2 Illumination Adjustment	73
4.1.3 Definition of rust color by the fundamental ellipse	74
4.2 Adaptive Ellipse Approach.....	75
4.3 Discussion and Comparison	81
4.3.1 Comparison of K-Means in grayscale, RGB, and a^*b^*	81
4.3.2 Comparison of K-Means with AEA	83
4.4 Summary of Chapter	85
CHAPTER 5 BOX-AND-ELLIPSE-BASED NEURO- FUZZY APPROACH (BENFA) FOR BRIDGE COATING ASSESSMENT	87
5.1 Preparation of Training Data	88
5.1.1 Input Features	88
5.1.1.1 Chrominance of the $L^*a^*b^*$ and its eigenvector	88
5.1.1.2 Chrominance of the $L^*a^*b^*$ and intensity L^*	90
5.1.2 Output Designation	92
5.1.2.1 Output Value Setting	92
5.1.2.2 Designation Strategy	94
5.1.2.2.1 Entropy of Chrominance lower than four	94
5.1.2.2.2 Entropy of Chrominance higher than four	95
5.1.3 Illumination Adjustment of Input Image	97
5.1.4 Division of Training Sets	99
5.2 Development of Membership Functions	103
5.2.1 Shape of Membership Function.....	104
5.2.2 Number of Membership Function	105
5.3 Automation of Rust Recognition by ANFIS	107
5.3.1 Picking Sub-ANFIS	107
5.3.2 Automated Determination of Two Thresholds.....	108
5.4 Stepwise BENFA model.....	111

5.5 Applications of BENFA approach	113
5.6 Summary of Chapter	115
CHAPTER 6 Illumination Adjustment for Bridge Coating Images Using BEMD-Morphology Approach (BMA)	117
6.1 Shade and Shadow Elimination by Empirical Mode Decomposition	117
6.2 Highlight Substitution by Rust Color	122
6.2.1 Morphological Reconstruction for Brightness Detection	123
6.2.2 Seed Points of Highlight	124
6.2.3 Highlight Detection	126
6.2.4 Color Substitution	127
6.2.5 Application of Highlight Substitution by Rust Color	128
6.3 Illumination Adjustment Strategy -BMA	131
6.3.1 Development of the BMA	132
6.3.2 Stepwise the BMA	133
6.3.3 Evaluation of the BMA using the K-Means	136
6.4 Summary of Chapter	137
CHAPTER 7 CONCLUSIONS AND RECOMMENDATIONS	139
7.1 Conclusions of Research	139
7.2 Research Contributions	142
7.3 Limitations	143
7.4 Recommendations for Future Work	143
References	145

LIST OF FIGURES

Figure 2.1 ANNs and Fuzzy approach model (Chae and Abraham 2001).....	9
Figure 2.3 24 bits RGB color cube.....	13
Figure 2.4 Unfeasibility of measurement of color similarity in RGB color space.	20
Figure 2.5 Histogram of a grey image (from the official website of Mister Donut)	21
Figure 2.6 Theory and application of the basic global thresholding.....	25
Figure 2.7 A 3*3 grayscale image	26
Figure 2.8 Clustering result of the 3*3 grayscale image by K-Means.....	27
Figure 2.9 Crisp sets and fuzzy sets of TRUE and FALSE (Chi, Yan et al. 1996)	30
Figure 2.10 A normal fuzzy set (Chi, Yan et al. 1996)	31
Figure 2.11 Deriving membership functions from cluster centers and variances. (Chi, Yan et al. 1996)	32
Figure 2.12 Operations on fuzzy sets (Chi, Yan et al. 1996).....	33
Figure 2.13 Defuzzification	34
Figure 2.14 Structure of artificial neuron	37
Figure 2.15 Learning algorithm (Chang and Chang 2007)	38
Figure 2.16 Feedforward network (Chang and Chang 2007)	39
Figure 2.17 Feedback network (Chang and Chang 2007).....	39
Figure 2.18 Structure of ANFIS (Chang and Chang 2007).....	41
Figure 2.19 Two heuristic rules for updating step size k (From (Jang 1993), figure 10).....	45
Figure 2.20 (a) Sugeno fuzzy reasoning. (b) Equivalent ANFIS (from (Jang 1993) fig. 4).....	46
Figure 2.21 The cubic spline upper and the lower envelopes and their mean, m1. (Huang 2005).....	47
Figure 2.22 Repeated Dilations of Marker Image, Constrained by Mask (The MathWorks 2008).....	50
Figure 2.23 Surface interpolation by RBFs	51
Figure 2.24 Application of dilation and erosion on a grayscale image	53
Figure 2.25 Application of opening and closing on a grayscale image.....	54
Figure 2.26 TopHat transform contrast correction(Kasperek 2001)	55
Figure 3.1 Scatter of rust color in HSI color space and RGB color space	57
Figure 3.2 Partition rust color into two parts based on H=50 in the HSI color space	57
Figure 3.3 Rust image and its scatter in the L*a*b* color space.....	59

Figure 3.4 Segmentation results of K-Means	59
Figure 3.5 Magnification of a rust image	60
Figure 3.6 Artificial rust image and its map	63
Figure 3.7 Background texture of artificial image	63
Figure 3.8 Uniformly illuminated artificial rust images.....	64
Figure 3.9 Non-uniformly illuminated artificial rust images.....	64
Figure 3.10 Color difference between non-transformation and transformation	68
Figure 3.11 Irreversibility of transformation from RGB to $L^*a^*b^*$ to RGB	68
Figure 3.12 Rust images for quantifying the power of L^* filtering the light	69
Figure 3.13 The performance of the light filter L^*	70
Figure 4.1 Process of automated detection of background.....	72
Figure 4.2 Result of automated detection of background.....	73
Figure 4.3 Result of background elimination after illumination adjustment.....	74
Figure 4.4 Rust colors and the fundamental ellipse	75
Figure 4.5 Application of automated detection of background and fundamental ellipse in non-uniformly illuminated rust image with blue paint	76
Figure 4.6 Application of automated detection of background and fundamental ellipse in uniformly illuminated rust image with blue paint	77
Figure 4.7 Application of automated detection of background and fundamental ellipse in uniformly illuminated rust image with khaki paint	77
Figure 4.8 Enlargement size is based on the distance between il and the mean of background color	78
Figure 4.9 Flowchart of the Adaptive Ellipse Approach.....	81
Figure 4.10 Comparison of the processed results of a uniformly illuminated image using the K-Means algorithm	82
Figure 4.11 Comparison of the processed results of a non-uniformly illuminated image using the K-Means algorithm	83
Figure 4.12 Processed results using the K-Means algorithm	84
Figure 4.13 Processed results using the adaptive ellipse approach (AEA)	85
Figure 5.1 Result of AEA with enlargement 0.1.....	87
Figure 5.2 Comparison of the results of ANFIS between training with eigenvector and without eigenvector.....	89
Figure 5.3 Comparison of the results between training with (a^*, b^*) and $(a^*, b^*,$ $L^*)$	91
Figure 5.4 Result with setting rust=1, background=0	93
Figure 5.5 Result with setting background=1, rust =0	93
Figure 5.6 Designation of output with $H \leq 4$	95
Figure 5.7 Designation of output with $H > 4$	97

Figure 5.9 Results from different ANFIS trained with different illumination adjustment combinations	99
Figure 5.10 Results of training an ANFIS with different and the same coating images	100
Figure 5.11 Scatters of rust image with different coating's colors.....	101
Figure 5.12 Results of ANFIS with grouping	103
Figure 5.14 Framework of the proposed ANFIS model.....	107
Figure 5.15 Relationship of groups and the border.....	108
Figure 5.16 Output of a sub-ANFIS.....	109
Figure 5.17 Output of every 10% additional colors.....	110
Figure 5.18 Stepwise BENFA model	112
Figure 5.19 Application of BENFA.....	113
Figure 5.20 Comparison of BENFA and FCM in rust intensity recognition	114
Figure 5.21 Comparison of BENFA and FCM in stability	115
Figure 6.1 Application of BEMD on a grey image	118
Figure 6.2 Artificial rust images for BEMD	119
Figure 6.3 Original and BEMD-adjusted artificial rust images segmentation using K-Means in RGB color space	121
Figure 6.4 BEMD adjustment in real color rust image	122
Figure 6.5 Two kinds of highlight on bridge surface	123
Figure 6.6 RGB cube and its transformation in MS diagram (Ortiz 2007)	124
Figure 6.7 Simplification of highlights area on MS diagram	125
Figure 6.8 Contrast enhancement and seed points of highlight.....	125
Figure 6.9 Highlight detection by seed point enlargement.....	126
Figure 6.10 Highlight detection without seed points	127
Figure 6.11 The third times elongated four neighbor relationships	127
Figure 6.12 Rust color substitution of highlight area.....	128
Figure 6.13 Mark of suspicious area	128
Figure 6.14 Highlight substitution by neighboring color and suspicious point detection.....	130
Figure 6.15 Comparison of highlight substitution with original image using the K-Means in the a*b* color configuration	131
Figure 6.16 Combination of BEMD and highlight substitution	132
Figure 6.17 Flowchart of BMA.....	134
Figure 6.18 Application of BMA in uniformly and non-uniformly illuminated images	135
Figure 6.19 Evaluation of BMA in shaded and non-shaded rust image using the K-Means.....	137

LIST OF TABLES

Table 3.1 Eigenvectors of rust and white and blue background color 61

Table 3.2 Eigenvectors of rust image from blue paint bridge..... 62

Table 3.3 Eigenvectors of rust image from white paint bridge..... 62

Table 3.4 Performance evaluation of color spaces by uniformly illuminated
artificial rust images 66

Table 3.5 Performance evaluation of color spaces by non-uniformly illuminated
artificial rust images 66

Table 4.1 Enlargement of fundamental ellipse by trial and error (shade marks
the optimal enlargement percentage) 80

Table 4.2 Processing time for a 256x256 rust image using the K-Means
algorithm..... 83

Table 4.3 Processing times for a 256x256 rust image on the a*b* color plane 85

Table 5.1 Process time of ANFIS without eigenvector and ANFIS with
eigenvector..... 90

Table 5.2 Comparison of different shapes of membership function 105

Table 5.3 Comparison of different number of membership functions 105

Table 5.4 Processing times of BENFA and FCM for a 256x256 rust image on the
a*b* color plane 115

Table 6.1 Accuracy of in artificial images segmentation using K-Means 120

ABSTRACT

Image recognition has been widely utilized in scientific research and prevalently adopted in industries. Application in infrastructure condition assessment includes defect recognition on steel bridge painting and underground sewer systems. Nevertheless, there is still no robust method to overcome the non-uniform illumination problem. The non-uniform illumination problem is arisen from the shades, shadows, and the highlights on a rust image. Although, K-Means, which is a kind of clustering methods according to the differences of each pixel, is recognized as one of the best rust defect recognition methods, it cannot recognize the non-uniform illuminated images and the mild rust color well. Also, there is lack of an automated color image recognition system in this field. The purpose of this research is to attempt to resolve the problems of non-uniform illumination and mild rust color as well as to automate the recognition system.

This research starts with an investigation of 14 color spaces in order to find out a comparatively proper color configuration for non-uniformly illuminated rust image segmentation. Among the 14 color spaces, the color configuration of a^*b^* , which has moderate ability to filter light, is utilized to develop the proposed two models, adaptive ellipse approach (AEA) and box and ellipse-based neural fuzzy approach (BENFA). A color configuration is an individual or a combination of the components of a color space.

In the adaptive ellipse approach (AEA), a rust image is partitioned into three parts, background, rust, and the gradual change color from mild-rust to background. The main idea is to deal with the gradual color change properly for mild rust color extraction. The background colors can be automatically detected from a rust image. A fundamental ellipse is previously defined by the collection of rust colors. The AEA enlarges the

fundamental ellipse to include part of the gradual change in color, and the enlarged size depends on the relationship between the rust color and the color of coating. The AEA is expected to deal with the boundary between background color and rust color properly. In addition, illumination adjustment is adopted in this model in order to overcome the non-uniform illumination problem. Finally, the processing results of AEA are compared to those of the K-Means clusters method to show that AEA could recognize mild rust colors and overcome the non-uniform illumination problem.

When the color distribution is almost parallel to the major axis of the fundamental ellipse, the proposed AEA may not recognize the mild-rust-colors well. Therefore, the box and ellipse-based neural fuzzy approach (BENFA) is proposed to deal with the gradual color change from mild-rust to background. BENFA applies the adaptive-network-based fuzzy inference system (ANFIS), which is a fuzzy inference system trained by a neural network, to describe the gradual change colors. In order to achieve automated detection, the BENFA applies the automated detection of background, illumination adjustment, and the fundamental ellipse to determine the thresholds of serious rust and mild rust. Compared to the Fuzzy C-Means (FCM) method, BENFA could overcome the non-uniform illumination problem and, furthermore, robustly recognize rust intensity. FCM is similar to the K-Means clustering, but the method applies the fuzzy concept on the clustering algorithm.

The third model which is called BEMD-morphology approach (BMA) aims to adjust the color of a non-uniformly illuminated rust image. The objective features of an image could be extracted by the morphological processing, and the bidimensional empirical mode decomposition (BEMD) decomposes a gray image. BMA applies BEMD to mitigate the shade/shadow effect, and applies morphology to substitute the highlight points with neighboring colors. The results show that processing a rust image with the BMA is more reliable than processing without the BMA.

KEYWORDS: Coating defect recognition, image processing, K-Means, adaptive-network-based fuzzy inference system (ANFIS), Fuzzy C-Means, bidimensional empirical mode decomposition (BEMD), morphology

CHAPTER 1 INTRODUCTION

1.1 Background

Digital image processing has impacted almost every area of technical endeavor in some way. From face recognition, thumb print comparison, automated license plate reading, to quality control of manufactured goods, and so forth, image processing has been widely used since late 1960s (Gonzalez and Woods 2001). Application of image processing not only helps human to obtain detailed information, but also provides more objective and accurate reference.

Conventional non-destructive steel bridge coating assessment is carried out through human visual inspection which makes relatively inconsistent and unconvincing results. In North America, civil engineers have utilized image processing for steel bridge coating inspection since late 1990s (AbdelRazig 1999; Chen 2001; Chen, Chang et al. 2002; Chen and Chang 2003; Lee, Chang et al. 2005). The reason of adopting image processing methods for steel bridge coating inspection arose from the steel bridge painting warranty contract in America. In the steel bridge painting warranty contract, it is usually stated that the owner and contractor have to do joint bridge coating inspection a certain period of time (e.g., two years) after the completion of the bridge painting work. If the rust percentage is greater than or equal to a pre-determined percentage (e.g., 1%), the contractor will have to re-do the painting work entirely or partially according to the warranty contract. To ensure the obtained rust percentage is objective and convincing, digital image processing methods are brought into this domain.

1.2 Problem Statement

As computers can distinguish millions of shades of colors which is difficult to achieve by human beings, computer-based image processing becomes more and more

popular. Moreover, computer-aided image processing could provide objective results, if those results are properly processed. However, there is still no algorithm that could process all types of images properly. In previous studies, most steel bridge coating images (i.e., rust images) were processed in gray scale. Since the conversion from color images to grayscale ones would lose information and could affect processed results significantly, color image processing method that simulates human vision will be developed in this research.

Non-uniform illumination problem is always a challenge in the field of image processing. Today, there is no robust algorithm to solve the illumination problem. Unfortunately, rust image acquisition is usually not under ideal condition. Sometimes there are some shadows or shades that enhancing the difficulty of image recognition. A proper processing to resolve this problem is necessary.

Most of the previous research works only focused on the detection of rust defects in area percentages. Previously developed systems lack the ability to recognize the different rust intensities. Recognition of the intensity of rust may provide a flexible assessment. For instance, the more conservative owner can make her/his judgment according to the more conservative evaluation result.

In this research, a method called adaptive ellipse approach (AEA) which is composed of automation detection of background and illumination adjustment will be proposed to achieve mild-rust-color recognition. The second method that combines the illumination-adjusted color, the neural network, and the fuzzy logic system will be proposed. This method is termed the box-and-ellipse-based neuro-fuzzy approach (BENFA). By means of the rich information of color, the model is expected to resolve the non-illumination problem and distinguish the rust intensity. Finally, another approach called BEMD-morphology approach will be presented to adjust the illumination of a bridge coating image.

1.3 Research Objectives

The main objectives of this research are as follows:

1. To find out the proper color configuration for processing color rust image.
2. To construct an automated rust recognition model.
3. To resolve the problem arisen from non-uniform illumination.
4. To develop a feasible rust intensity recognition method.
5. To provide a model that helps construction professionals to make a flexible decision. For example, whether the serious rust or the mild rust color is taken into consideration.

1.4 Organization of the Research

The thesis is composed of six chapters and is organized in the following manner:

Chapter 1 Introduction:

This chapter depicts the background of applying image processing on steel bridge inspection, problem statement, research objectives, and the organization of the report.

Chapter 2 Literature Review:

This chapter presents the prior research efforts in image processing application to Civil Engineering, color image segmentation methods, artificial intelligence, Hilbert-Huang Transform, and morphological processing.

Chapter 3 Comparison of Color Configuration Using the K-Means Algorithm:

This chapter explores the 14 existing color spaces and evaluates the best color space to color rust images. The following work will be processed in the selected color space.

Chapter 4 Adaptive Ellipse Approach:

This chapter proposes the Adaptive Ellipse Approach in detail. From

automated detection of the background to enlargement of the fundamental ellipse, the construction of this model will be depicted in step-by-step procedures.

Chapter 5 Box-and-Ellipse-based Neuro-Fuzzy Approach (BENFA) for Bridge Coating Assessment:

This chapter details the proposed Box-and-Ellipse-based Neuro-Fuzzy Approach (BENFA). In order to automate the rust intensity recognition, the box-and-ellipse is utilized to determine two thresholds of mild rust and serious rust. The architecture of this model will be depicted in step-by-step procedures.

Chapter 6 Illumination Adjustment for Bridge Coating Images Using BEMD-Morphology Approach (BMA):

Chapter 6 is different from the previous two chapters. This chapter aims to adjust the illumination of a bridge coating image, not a rust recognition model. The BEMD-morphology approach (BMA) which aims to mitigate the shade/shadow effect and replace the highlight area with neighboring color will be introduced step by step in this chapter.

Chapter 7 Conclusions and Future Work:

This chapter shows the conclusions based on the research results and describes the future work and expectation.

CHAPTER 2 LITERATURE REVIEW

This chapter starts with the application of image processing in Civil Engineering, and then follows the theories of image processing and recognition.

2.1 Image Processing and Its Application in Infrastructure Condition

Assessment

Maintenance and Operation cost is one of the major component of the total life cycle cost. A cost-effective maintenance method for infrastructure should be developed. One of the conventional non-destructive assessment method is physical inspection which directly assessed in site by professional engineers (Chae and Abraham 2001). However, investigation carried out through human vision is subjective and inconsistent. Moreover, the work is sometimes very dangerous under specific situations, for instance, to investigate the high and cross river steel bridge. Automated inspection is a feasible and safe approach to assess the condition of infrastructure. Therefore, image processing and recognition is brought into this domain.

2.1.1 Underground Pipeline

There are two primary challenges of sewer rehabilitation, the work has to be done in highly developed area and the work is usually performed as the major failure occurs (Federation 1994). Since the sewer rehabilitation is difficult and costly, it is necessary to develop a cost-effective condition assessment method. Chae and Abraham (2001) utilized a multiple Artificial Neural Network (ANNs) to recognize the defections of inner sewer surface by images. The multiple ANNs is composed of different kinds of features, for instance, joints, cracks, corrosion, and other possible

features of sewer inner surface. Each feature has its own neural network. Each output of the neural network is the input of the fuzzy inference system. The overall condition assessment is carried out by fuzzy inference rules. The concept of ANNs and fuzzy algorithm will be introduced in the later sections.

Figure 2.1 ANNs and Fuzzy approach model (Chae and Abraham 2001)

2.1.2 Steel bridge Surface Inspection

Utilize image processing to assess steel bridge rust degree is not a new concept anymore. AbdelRazig (1999) proposed a hybrid model combined artificial neural network and image processing. The hybrid model uses K-Mean algorithm to cluster a grey rust image into two groups, rust and background. A threshold value is decided by the clustering result. An artificial neural network is trained by the previous information. The grey level of each pixel and its difference from image threshold are the input layer, and the clustering result is the output layer. By means of the training process, this model can automate the steel bridge rust degree assessment, simulate experts' knowledge, and make the model more fault-tolerant. However, directly process the grey level of an image cannot deal with non-uniformly illuminated image well.

Chen proposed the neuro-fuzzy recognition approach (NFRA) (Chen 2001). NFRA recommended that to segment a rust image into different areas in accordance with the illumination of the pixels. Each area has its threshold value which is decided by an artificial neural network. Furthermore, fuzzy concept was applied to adjust the illumination value along the boundaries of the different areas. Although this model provided a reasonable approach to deal with non-uniform illumination problem, a grey image which contains only 8 bits shades of light intensity is a natural limitation.

Lee recommended to establishing a complete steel bridge surface assessment system, and most important of all, introduced to process color images (Lee 2005). The first half of the system is to recognize whether the existence of rust in a color image by means of statistics. Cb/Cr color space is chosen based on the author's visual representation. The last half of the system is the recognition of rust area. All the data are projected on a horizontal axis by Hotelling Transform (HT) and segmented by a manipulated threshold.

So far, image processing is not used for steel bridge corrosion detection in practice, so there is no standard operating procedure. However, the paper proposed by Lee (2005) "Performance Comparison of Bridge Coating Defect Recognition Methods." Corrosion 61: pp. 12-20" analyzes some factors which may affect the corrosion recognition results.

2.2 Color Features

In previous research of image processing, most of the images are converted to grey images to speed up the process time or simplify the algorithm. However, advanced computer technology makes color image processing feasible. Compared to grey image processing, color image processing often simplifies object identification and extraction from a scene. Also, humans can discern thousands of color shades and

intensities, but only two dozen shades of grey (Gonzalez and Woods 2001). Therefore, in this research, color image processing is adopted.

2.2.1 Color Image

An image is composed of several picture elements whose abbreviation is pixels. Each pixel has an intensity value. To an 8 bits length grey image, each pixel could have 256 shades of grey color from 0 to 255, that is, from black to white. A color image has three layers. To a 24 bits length color image, each layer contains 8 bits shades of intensity values. The most fundamental color image is RGB image. RGB (red, green, blue) respectively represent the three layers, and all the combinations make different colors that constitute a RGB color space.

Figure 2.2 RGB color image

2.2.2 Color Space and Color Configuration

Since color discrimination is a subjective perception to human, CIE (Commission Internationale de l'Eclairage-the International Commission on Illumination) standardized the specific wavelength values to the three primary colors : blue = 438.5 nm, green = 546.1 nm, and red = 700 nm on 1931. RGB are the primary

colors which can produce other colors. The relative color information and fundamental concept could refer to (Gonzalez and Woods 2001).

The characteristics generally used to distinguish different colors are brightness, hue, and saturation. Brightness embodies the achromatic notion of intensity. Hue represents the pure color perceived by an observer. Saturation refers to the relative purity or the amount of white light mixed with a hue. Hue and saturation taken together are called chromaticity. Brightness does not contain chromaticity information but control the intensity. A color may be characterized by its chromaticity and brightness.

This section details the 14 color spaces assessed in this research: the RGB color space, $I_1I_2I_3$, HSV, HIS, YUV, YIQ, YCbCr, YCgCr, XYZ, $W^*U^*V^*$, $L^*u^*v^*$, $L^*a^*b^*$, and $L^*C^*h^*$ color space. The fourteen color spaces are classified and introduced as following.

Fundamental color space

The RGB color space

The three primary colors constitute the RGB color space which is represented in a 3-dimensional cube showed in Figure 2.3. To RGB color space, if two colors are equivalent, after multiplying or dividing the three components by a constant, the new color is still the same color which only changed its brightness. The reason can be easily observed from the RGB color cube. Therefore, color in RGB color space has high correlation with intensity change. Note that it is impossible to evaluate the similarity of two colors from their distance in RGB color space, since the distance of different colors with similar brightness may be shorter than the same colors with significant different brightness (Cheng, Jiang et al. 2001).

Figure 2.3 24 bits RGB color cube

The rgb color space

Since the RGB color images are sensitive to luminance, object surface reflection, and other photographic conditions(Shih and Liu 2005), rgb color space which normalizes the value of RGB was proposed to minimize the luminance sensitivity. The transformation is defined as following:

$$r = R / (R + G + B) \quad (2-1)$$

$$g = G / (R + G + B) \quad (2-2)$$

$$b = B / (R + G + B) \quad (2-3)$$

The I1I2I3 color space

Another color space was proposed to overcome the shortages of RGB color space, called I1I2I3 color space or Ohta color space(Ohta, Kanade et al. 1980). In order to decorrelate the RGB components, Ohta applied the Karhunen Loeve transformation of R, G, and B data thorough the given 8 images, they found that the color features, I1,I2, and I3 are effective. They recommend that I1 and I2 contain the most features, so in many cases a good segmentation can be down by using only the first two.

$$I1 = (R + G + B) / 3 \quad (2-4)$$

$$I_2 = R - B \quad (2-5)$$

$$I_3 = (2G - R - B) / 2 \quad (2-6)$$

Human perceptual color space

The HSI and HSV color spaces are developed to simulate human vision system. Hue and saturation define chrominance, and intensity or value describes luminance. Since the ability to separate intensity from the color, the human perceptual system is an ideal tool to image processing. Coianiz et al. proposed to use hue filtering to extract the lip region through a weight function, and then use the low-pass spatial filtering to reduce the effect of noise(Coianiz, Torresani et al. 2002).

The HSI color space

HSI color space describes a color by three components: hue, saturation, and intensity. Hue and saturation taken together called chromaticity is the horizontal plane of the HSI model. In the horizontal plane, a color is described as a vector whose angle and length specify hue and saturation accordingly. Intensity is the vertical axis to construct the color space. Color conversion from RGB to HSI is showed as equation 2-7 to equation 2-11 (Gonzalez and Woods 2001).

$$H = \begin{cases} \theta & \text{if } B \leq G \\ 360 - \theta & \text{if } B > G \end{cases} \quad (2-7)$$

In order to confine the range of $H = [0, 360]$,

$$H = H + 360 \text{ if } H < 0 \quad (2-8)$$

$$\theta = \cos^{-1} \left\{ \frac{\frac{1}{2}[(R-G)+(R-B)]}{\left[((R-G))^2 + (R-B)(G-B) \right]^{1/2}} \right\} \quad (2-9)$$

$$S = 1 - \frac{3}{(R+G+B)} [\min(R, G, B)] \quad (2-10)$$

$$I = \frac{1}{3} (R + G + B) \quad (2-11)$$

H: hue , $[0,360]$, S: saturation, $[0,1]$, I: intensity, $[0 1]$.

It is assumed that the RGB values have been normalized to the range $[0 1]$ and that

angle θ is measured with respect to the red axis of HSI space.

The HSV color space

The three components of HSV color space are hue, saturation, and value (also called intensity value). The concept of HSV color space is very similar to HSI color space. Both of the two color spaces describe colors as cone shape whose central axis ranges from white at the top to black at the bottom. Only the shape of the model is different. HSI constructs a double-cone, while HSV is a single cone shape(Wikipedia 2007). The HSV color space is defined as follows(Shih and Liu 2005):

$$\text{Let } \begin{cases} \text{MAX} = \max(R, G, B) \\ \text{MIN} = \min(R, G, B) \\ \delta = \text{MAX} - \text{MIN} \end{cases} \quad (2-12)$$

$$V = \text{MAX} \quad (2-13)$$

$$S = \begin{cases} \frac{\delta}{\text{MAX}} & \text{if MAX} \neq 0 \\ 0 & \text{if MAX} = 0 \end{cases} \quad (2-14)$$

$$H = \begin{cases} 60 \left(\frac{G-B}{\delta} \right) & \text{if MAX} = R \\ 60 \left(\frac{B-R}{\delta} + 2 \right) & \text{if MAX} = G \\ 60 \left(\frac{R-G}{\delta} + 4 \right) & \text{if MAX} = B \end{cases} \quad (2-15)$$

Note that if MAX=0, H is not defined. Also H should be confined within the range of [0, 360].

Video Transmission Efficiency Color Space

The YUV, YIQ, and YCbCr color spaces were developed for video transmission efficiency(Shih and Liu 2005). YUV color space is adopted by the PAL (Phase Alternation by Line) and the SECAM (System Electronique Couleur Avec Memoire). The YIQ color space is used by the NTSC (National Television System Committee). The YCbCr color space is a scaled and offset version of YUV color space. The Y component refers to luminance, and the remaining two components specify chrominance. For instance, Cb means chrominance blue, and Cr is chrominance red.

The YUV Color Space

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.2990 & 0.5870 & 0.1140 \\ -0.1471 & -0.2888 & 0.4359 \\ 0.6148 & -0.5148 & -0.1000 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2-16)$$

The YIQ Color Space

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.2990 & 0.5870 & 0.1140 \\ 0.5957 & -0.2745 & -0.3213 \\ 0.2115 & -0.5226 & 0.3111 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2-17)$$

The YCbCr Color Space

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 65.4810 & 128.5530 & 24.9660 \\ -37.7745 & -74.1592 & 111.9337 \\ 111.9581 & -93.7509 & -18.2072 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2-18)$$

where the R, G, B values are scaled to [0, 1].

The YCgCr Color Space

A new color space YCgCr is applied for face recognition(de Dios and Garcia 2003). It is similar to YCbCr, but using the Cg, chrominance green, component instead of Cb. Scattered in Cg-Cr plane and Cb-Cr plane, the skin color pixels are more condense in Cg-Cr plane. The threshold result showed that using YCgCr and YCbCr was very similar but in some cases YCgCr was better.

$$\begin{bmatrix} Y \\ Cg \\ Cr \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 65.481 & 128.553 & 24.966 \\ -81.085 & 112 & -30.915 \\ 112 & -93.768 & -18.214 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2-19)$$

CIE Uniform Color Space

One of the way to form a color is by X, Y, Z tristimulus values. X, Y, and Z denote the amounts of red, green, and blue needed to form a particular color respectively. The CIE uniform color spaces, $W^*U^*V^*$, $L^*u^*v^*$, $L^*a^*b^*$, and $L^*C^*h^*$ are developed based on the XYZ tristimulus values(Gonzalez and Woods 2001; Shih and Liu 2005).

$$u = 4x/(-2x + 12y + 3) \text{ or } 4X/(X + 15Y + 3Z) \quad (2-20)$$

$$v = 6y/(-2x + 12y + 3) \text{ or } 6Y/(X + 15Y + 3Z) \quad (2-21)$$

The W*U*V* Color Space

The W* component corresponds to luminance, whereas the U*, V* components refer to chrominance.

$$W^* = \begin{cases} 116 \left(\frac{Y}{Y_0} \right)^{\frac{1}{3}} - 16, & \text{if } \frac{Y}{Y_0} > 0.008856 \\ 903.3 \left(\frac{Y}{Y_0} \right) & \text{otherwise} \end{cases} \quad (2-22)$$

$$U^* = 13W^* (u - u_0) \quad (2-23)$$

$$V^* = 13W^* (v - v_0) \quad (2-24)$$

where Y₀, u₀, and v₀ are derived from the reference white stimulus.

The L*u*v* Color Space

The CIE-uv diagram has its deficiency in representing yellow-red colors. L*u*v* was proposed to improve the shortage by adopting the new uv diagram.

$$u' = u \quad (2-25)$$

$$v' = \frac{3}{2}v \quad (2-26)$$

$$L^* = \begin{cases} 116 \left(\frac{Y}{Y_0} \right)^{\frac{1}{3}} - 16 & \text{if } \frac{Y}{Y_0} > 0.008856 \\ 903.3 \left(\frac{Y}{Y_0} \right) & \text{otherwise} \end{cases} \quad (2-27)$$

$$u^* = 13L^* (u' - u_0') \quad (2-28)$$

$$v^* = 13L^* (v' - v_0') \quad (2-29)$$

The L*a*b* Color Space

The L*a*b* system is a good decoupler of luminance (represented by lightness L*) and color (represented by a* for red minus green and b* for green minus blue). The L* component corresponds to brightness ranging from 0 (black) to 100 (white). The a* component measures the color redness (positive value) or greenness (negative value). The b* component measures the color greenness (positive value) or blueness (negative value).

$$L^* = 116f\left(\frac{Y}{Y_0}\right) - 16 \quad (2-30)$$

$$a^* = 500 \left[f\left(\frac{X}{X_0}\right) - f\left(\frac{Y}{Y_0}\right) \right] \quad (2-31)$$

$$b^* = 200 \left[f\left(\frac{Y}{Y_0}\right) - f\left(\frac{Z}{Z_0}\right) \right] \quad (2-32)$$

$$\text{where } f(x) = \begin{cases} x^{\frac{1}{3}} & \text{if } x > 0.008856 \\ 7.787x + \frac{16}{116} & \text{otherwise} \end{cases} \quad (2-33)$$

The L*C*h* Color Space

The CIE L*C*h* color space is a polar color space which is calculated from the L*a*b* scale values. The L*, lightness, value is the same as in L*a*b* color space. The C* value, chroma, and the h* value, hue angle, are calculated from the L*a*b* color space.

$$L^* = 116f\left(\frac{Y}{Y_0}\right) - 16 \quad (2-34)$$

$$C^* = \sqrt{a^{*2} + b^{*2}} \quad (2-35)$$

$$h^* = \arctan \frac{b^*}{a^*} \quad (2-36)$$

$$\text{where } f(x) = \begin{cases} x^{\frac{1}{3}} & \text{if } x > 0.008856 \\ 7.787x + \frac{16}{116} & \text{otherwise} \end{cases} \quad (2-37)$$

Color Configuration

A color configuration is an individual or a combination of the component of the color space(Shih and Liu 2005). Take RGB space as an example, the possible combinations are R, G, B, RG, RB, GB, and RGB. Therefore, every single color space has seven color configurations.

2.3 Color Image Segmentation

Image segmentation is widely applied in many fields. The purpose of

segmentation is to extract the particular object. Generally speaking, different objects have different features. Therefore, feature extraction is the key part of image segmentation. This section first introduces the most common features in the literature, and then follows several segmentation algorithms, such as threshold segmentation, clustering algorithm, fuzzy inference, and ANNs.

2.3.1 Feature Extraction

2.3.1.1 Color Similarity

Most of the image segmentation is based on color homogeneity, that is, color similarity. It is assumed that each item has a color. Therefore, the measurement of color similarity plays an important role to precise segmentation. However, it is important to notice that, based on the color homogeneous assumption the existence of shade may affect the result. Non-uniform illumination problem is still a challenge in the field.

One of the measurement methods is color indexing. Color indexing is a technique retrieving the images whose color compositions are similar to the target image. Traditional color indexing contains the complete color histogram of the images. Stricker and Orengo proposed two methods (Stricker and Orengo 1995). The first one improved the robustness with respect to changes in the form of cumulative histogram. While this method produces only slightly better results than color histogram methods, it is more robust with respect to the quantization parameter of the histograms. The second index contains only dominant features which are the first three moments of each color channel of an image. The similarity function which is used for the retrieval is a weighted sum of the absolute differences between corresponding moments.

Figure 2.4 Unfeasibility of measurement of color similarity in RGB color space.

Another method is also the most intuitive concept, measurement by absolute distance in the same color space. It is important to mention that RGB color space is not suitable in this measurement (Cheng, Jiang et al. 2001). Since the dependency of intensity, the two colors whose distance is very short may be due to their similar intensity, not similar chroma. Figure 2.4 explains the unfeasibility of measurement of color similarity in RGB color space. Both color A and B are grey, but the purple color C is closer to A than B is. Therefore, the color space which decouples the intensity and chroma is proper to be used in this method. For instance, the difference of two colors can be calculated as the Euclidean distance between two color points in $L^*a^*b^*$ color space. The smaller of ΔE_{ab} , the more similar to the two colors.

$$\Delta E_{ab} = \sqrt{(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2} \quad (2-38)$$

2.3.1.2 Histogram

The histogram of a grey image is the frequency of each grey level. The histogram is a discrete function $h(a_k) = n_k$, where r_k is the k th grey level and n_k is the number of

pixels having grey level r_k (Gonzalez and Woods 2001). Each object has its own histogram which is like the identification.

Figure 2.5 Histogram of a grey image (from the official website of Mister Donut)

2.3.1.3 Entropy

In image processing, entropy is similar to the concept of non-homogeneous. Entropy depicts the variety of the color. Therefore, entropy is calculated from the histogram of an image. For instance, the entropy of a single color image is zero. The equation is

$$En(z) = - \sum_{j=1}^J P(a_j) \log_2 P(a_j) \quad (2-39)$$

In this equation: En denotes the entropy, z denotes a image, J specifies the range of the intensity, a denotes the intensity, and P refers to probability of the particular intensity in the image.

2.3.1.4 Principal Component Analysis

Principal Component Analysis (PCA) is a decorrelation procedure by means of the DN values (Digital Number, A positive integer representing the relative reflectance or emittance of an object in a digital image. For 8 bit images, the DN Value lies in the range 0-255.) from an image. It uses eigen analysis for decorrelation. The eigenvector associated with the largest eigenvalue has the same direction as the

first principal component. The eigenvector associated with the second eigenvalue has the same direction as the second principal component (Navulur 2007).

Through PCA, the discriminative and low dimensional features are acquired by projecting the raw data of an image onto the low dimensional space spanned by the principal components. This method is frequently applied in face recognition (Menser and Muller 1999; Xutao, Yudong et al. 2006).

The further application of PCA in face recognition is two dimensional PCA (Zhang, Chen et al. 2005). The classical PCA is used by transforming the 2D images matrices into 1D image vectors, and then to present those 1D vectors with eigenvectors. Zhang proposed “eigenimages” to maintain the spatial information without transforming the 2D images into 1D image vectors. The result showed that “eigenimages” not only has better reconstruction quality for face recognition but also reduces the run time than the classical PCA method.

This report is focus on application of classical PCA. Following is the theoretical mathematics (Shores 2007).

Suppose that Z is the vector or matrix of an image with n pixels:

$$Z = \begin{bmatrix} Z_{1R} & Z_{1G} & Z_{1B} \\ Z_{2R} & Z_{2G} & Z_{2B} \\ \vdots & \vdots & \vdots \\ Z_{nR} & Z_{nG} & Z_{nB} \end{bmatrix} \quad (2-40)$$

Since the eigenvalue problem is suitable to solve a square matrix, the image should be transformed to a covariance matrix.

$$\Sigma = E\{(Z - \mu)(Z - \mu)^T\} \quad (2-41)$$

Where Σ denotes a covariance matrix, and μ is the mean of Z .

The symbol A is to substitute Σ for simplification of writing.

$$A = \Sigma \quad (2-42)$$

Then, the calculation of eigenvalue of an image could be converted into the following equation:

$$Ax = \lambda x \quad (2-43)$$

The scalar λ is called an eigenvalue of the matrix A . The vector x is an eigenvector belonging to the eigenvalue λ . The pair $\{\lambda, x\}$ is called an eigenpair for the matrix A . In German, the word “eigen” means “characteristic”, so the eigenvalues and eigenvectors are also known as characteristic values and characteristic vectors.

To calculate the eigenvalues λ of matrix A is to solve the polynomial equation

$$\det(\lambda I - A) = 0 \quad (2-44)$$

where $\det$ returns the determinant of the matrix and I denotes the identity matrix.

Numerical Example

Find the eigenvalues and eigenvectors for the matrix $A = \begin{bmatrix} 2 & 3 \\ -1 & 6 \end{bmatrix}$

Solution:

$$\begin{aligned} 0 &= \det(\lambda I - A) = \det \begin{bmatrix} \lambda - 2 & -3 \\ 1 & \lambda - 6 \end{bmatrix} \\ &= (\lambda - 2)(\lambda - 6) - (1)(-3) \\ &= \lambda^2 - 8\lambda + 12 \\ &= (\lambda - 3)(\lambda - 5) \end{aligned}$$

Hence the eigenvalues are $\lambda=3, 5$.

$$\lambda=3, \text{ we have } \begin{bmatrix} 2 & 3 \\ -1 & 6 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = 3 \begin{bmatrix} a \\ b \end{bmatrix}$$

Both of the two rows of this matrix equation reduce to the single linear equation $a = 3b$. To find an eigenvector, we are free to choose any value for a . By picking $a=1$ and setting $3b = a$, we find the eigenvector to be:

$$\begin{bmatrix} 1 \\ 1/3 \end{bmatrix}$$

$\lambda=5$, repeat the previous steps to obtain the eigenvector:

$$\begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

2.3.2 Threshold Segmentation

This section introduces the theoretical background of the threshold segmentation. The theoretical portion is based on the explanation of Gonzalez and Woods (Gonzalez and Woods 2001, pp 595-612). Suppose that a histogram corresponds to an image, composed of a dark background and a light object, there should be two dominant grey level peaks. One of the intuitive way to extract the object from the background is to decide a threshold T that partitions the two peaks. Thereupon any point above T is called an object point; otherwise, the point is called a background point.

Since its simplicity of implementation and intuitive properties, threshold segmentation has been widely applied and extended in image segmentation. There are several methods in the literature, one of the methods called basic global thresholding is shown as following Figure 2.6. A global threshold T could be automatically obtained by the following algorithm:

1. Pick an initial estimate for T .
2. According to the value T , there are two regions, G_1 and G_2 , which are divided by the threshold T .
3. Compute the average value T_L and T_H for the pixels in regions G_1 and G_2 .
4. Compute a new threshold value T_i :

$$T_i = \frac{1}{2} (T_L + T_H)$$

(2-45)

5. Repeat the steps from 2 to 4 until the difference of T , $T_{i+1} - T_i$, is smaller than a predefined tolerant value.

Figure 2.6 Theory and application of the basic global thresholding

2.3.3 Clustering Segmentation

Clustering algorithm is to classify the raw data into several groups, and to effectively display the pattern of the system. The core concept is to do classification of the data, and then group the data which has similar property in one group. Hence, data with similar properties will be in the same group, while data in different groups have low similarity (Chang and Chang 2007). This section details two of the popular clustering segmentation methods, K-Means clustering algorithm and Fuzzy C-Means clustering algorithm.

2.3.3.1 K-Means Algorithm

The K-means algorithm is the simplest clustering algorithm. The clustering criterion is base on the distance from the data to the cluster means. The number of

clusters should be determined in advance. In fact, the symbol K refers to the number of clusters. The K centers of group are decided by minimizing the sum of distance between every data to its belonging group center. Generally speaking, the K-Means algorithm can be broken down as the following steps:

1. Randomly select K data as the initial group centers.
2. Assign the remaining $(N-K)$ data to the closest group based on the distance between each data and each group mean, where N is the total number of data.
3. Re-calculate the mean of each new group, and then set them to be the new centers.
4. Repeat the step two and three until the groups are fixed.

Numerical Example

Cluster the grayscale image showed as Figure 2.7 in two groups by means of K-Means.

55	70	188
173	180	192
67	40	186

Figure 2.7 A 3*3 grayscale image

Solution:

1. Randomly select $\{55, 70\}$ as the initial group centers. 55 is the center of group one, while 70 is the center of group two.
2. In the first iteration, group one includes $\{55, 40\}$ and group two includes $\{70, 67, 188, 173, 180, 192, 186\}$.
3. The new center of group one is $(55+40)/2=47.5$. The new center of group two is $(70+67+188+173+180+192+186)/7=150.9$.

4. In the second iteration, group one includes {55, 40, 70, 67} and group two includes {188, 173, 180, 192, 186}. The new center of group one is $(55+40+70+67)/4=58$. The new center of group two is $(188+173+180+192+186)/5=183.8$. In third iteration, there is no further change, so stop the calculation.
5. The clustering result is shown in Figure 2.8:

55	70	188
173	180	192
67	40	186

Figure 2.8 Clustering result of the 3*3 grayscale image by K-Means

The K-Means algorithm is summarized as the following equations:

$$J = \sum_{i=1}^k J_i = \sum_{i=1}^k \sum_{j=1}^n w_{ji} \|X_j - C_i\|^2 \quad (2-46)$$

$$w_{ji} = \begin{cases} 1, & \text{if } \|X_j - C_i\| \leq \|X_j - C_m\|, \forall m \neq j \\ 0, & \text{otherwise} \end{cases} \quad (2-47)$$

$$\sum_{i=1}^k w_{ji} = 1, \forall j = 1, \dots, n \quad (2-48)$$

$$\sum_{i=1}^k \sum_{j=1}^n w_{ji} = n \quad (2-49)$$

$$C_i = \frac{\sum_{j=1}^n w_{ji} X_j}{\sum_{j=1}^n w_{ji}} \quad (2-50)$$

Where k is the number of cluster, n is the number of input vectors, X_j is the j^{th} input vector, C_i is the i^{th} clustering center, J_i is the objective function of the i^{th} cluster. The computation will be stop when J has no further change.

2.3.3.2 Fuzzy C-Means Algorithm

Bezdek firstly propped Fuzzy C-Means to improve the K-Means algorithm

(Bezdek, Ehrlich et al. 1984). The definition of objective function of Fuzzy C-Means algorithm is the same as K-Means algorithm, but its weighted matrix w_{ji} is not binary matrix anymore. The weighted matrix w_{ji} , which is also known as membership function, is constrained in range 0 to 1 corresponding the similarity a point shares with each cluster. Compared with the K-Means algorithm, the Fuzzy C-Means algorithm has the better ability to deal with the boundary data between the different clusters.

The Fuzzy C-Means algorithm is summarized as the following equations:

$$J = \sum_{i=1}^k J_i = \sum_{i=1}^k \sum_{j=1}^n w_{ji}^m \|X_j - C_i\|^2 \quad (2-51)$$

$$\sum_{i=1}^k w_{ji} = 1, \forall j = 1, \dots, n \quad (2-52)$$

$$C_i = \frac{\sum_{j=1}^n w_{ji}^m X_j}{\sum_{j=1}^n w_{ji}^m} \quad (2-53)$$

$$w_{ji} = \frac{1}{\sum_{s=1}^k \left(\frac{\|X_j - C_i\|}{\|X_j - C_s\|} \right)^{\frac{2}{m-1}}} \quad (2-54)$$

Where J_i is the objective function of the i^{th} cluster, m is a weighted number usually set to 2, k is the number of cluster, n is the number of input vectors, X_j is the j^{th} input vector, C_i is the i^{th} clustering center. The computation will be stop when J has no further change.

In general, the Fuzzy C-means algorithm has the following steps:

1. Set the number k , randomly assign the weighted matrix in range zero to one.
2. Compute the center of clusters C_i .
3. Calculate the objective function. If the value of J is below the default tolerant error, stop the iteration. Otherwise, re-compute the weighted matrix w_{ji} and repeat the step 2 and 3.

Extended Application of the Fuzzy C-Means Algorithm

The Fuzzy C-Means algorithm has become popular in image processing application. Yang et al. (2002) proposed two methods, Separate Eigenspace Fuzzy

C-Means (SEFCM) and Coupled Eigen-based Fuzzy C-Means (CEFCM) methods, which are combined principal component transformation (PCT) concept and Fuzzy C-Means cluster method to extract the desired part of the color image (Yang, Hao et al. 2002). SEFCM applied principal eigenspace and residual eigenspace independently to get two segmented images, and then use rule AND to obtain the desired part of the color image. CEFCM directly use eigen-based membership function in fuzzy clustering. The results show that both of the two methods extracting the desired color image are more accurate than using PCT method and FCM method.

Since a conventional FCM algorithm clusters the image pixels based only on their feature space without considering the spatial interactions between neighboring pixels, Liew et al. (2003) proposed a spatial fuzzy clustering algorithm. The proposed algorithm is able to consider the image information both in feature space and the spatial interactions between neighboring pixels. Taking eight-neighborhoods into account, the author define a dissimilarity index which contain a weighting factor controlling the degree of influence of the neighboring pixels. After appropriate pre- and post-processing utilizing the color and the shape property of the lip image, the proposed method successfully segment the lip from a face (Liew, Shu Hung et al. 2003).

2.4 Artificial Intelligence

2.4.1 Fuzzy Logic Technique

The fundamental fuzzy logic theory is that it allows an event to belong to more than one sample space as sharp boundaries between spaces are hardly found. For example, human would rather include some linguistic expressions such as “It is very light”, but the linguistic expressions are hardly quantified. In image processing, it is more proper to treat the transient regions between two areas as fuzzy domains where the pixels

having almost the same or gradual change in color are in fact the expressions of fuzziness.

This section introduces the fuzzy logic from the fuzzy set, fuzzy logical operations, defuzzification to the final fuzzy inference. The theory is based on the explanation of *Chi and Yan et al.* (Chi, Yan et al. 1996). The authors summarized the advantages of fuzzy logic:

1. Fuzzy logic provides a systematic basis for quantifying uncertainty arisen from vagueness and incompleteness of information.
2. The unsharp boundaries between different classes can be easily defined by fuzzy sets.
3. Fuzzy logic is able to process the expert knowledge in a structured and consistent way.
4. When the use of probabilistic methods is not proper to a problem, the method of fuzzy sets are promising.

2.4.1.1 Fuzzy Sets

A fuzzy set is the degree of truth assigned for each of its members. The basic difference of fuzzy sets from the traditional crisp sets is the concept of describing “true” and “false”. Figure 2.9 shows the difference.

Figure 2.9 Crisp sets and fuzzy sets of TRUE and FALSE (Chi, Yan et al. 1996)

A fuzzy set includes a set of ordered pairs, with the form of $\{x, \mu_A(x)\}$. The element $\mu_A(x)$ represents the membership function (or the characteristic function) of x in the fuzzy set A , and it may take any real values in the interval $[0,1]$ corresponded the importance. Figure 2.10 illustrates a normal fuzzy set. A fuzzy set A can be expressed as $A = \{(x, \mu_A(x)) | x \in X\}$, where X is a collection of objects.

Figure 2.10 A normal fuzzy set (Chi, Yan et al. 1996)

2.4.1.2 Membership Functions

Setting and choosing membership functions is critical for a fuzzy pattern recognition model. The triangular, trapezoidal, S-shape, and bell-shaped membership functions are commonly used. The choice of the membership functions is usually problem dependent and subjectively determined. Figure 2.10 shows one example.

Beside the heuristic selection of membership functions, many other techniques have been developed to produce membership functions. For example, Cheng et al. (1997) automatically determined the membership function that corresponding fuzzy event has maximum entropy (Cheng and Chen 1997). The Maximum Entropy Principle implies a fuzzy event contains most information. The membership can be represented by an S-function with three parameters which are suggested using the simulated annealing algorithm to automatically get the three optimal parameters.

Figure 2.11 Deriving membership functions from cluster centers and variances. (Chi, Yan et al. 1996)

2.4.1.3 Fuzzy Logical Operations

The intention of operations on fuzzy sets is to provide the mathematical aspects of fuzzy set theory. The fuzzy set operations include:

1. *Fuzzy intersection*: The intersection of the two fuzzy sets A and B is interpreted as “A AND B”. It takes the minimum value of the two membership grades.

$$A(x) \cap B(x) = \sum \mu_A(x) \wedge \mu_B(x) \quad (2-55)$$

where the symbol $\wedge$ indicates the minimum operator.

2. *Fuzzy union*: The union of the two fuzzy sets A and B is interpreted as “A OR B”. It takes the maximum value of the two membership grades.

$$A(x) \cup B(x) = \sum \mu_A(x) \vee \mu_B(x) \quad (2-56)$$

where the symbol $\vee$ indicates the maximum operator.

3. *Fuzzy complement*: The complement of a fuzzy set A is understood as “NOT (A)”.

$$\bar{A} = \sum 1 - \mu_A(x) \quad (2-57)$$

where $\bar{A}$ signifies the complement of A.

Figure 2.12 Operations on fuzzy sets (Chi, Yan et al. 1996)

2.4.1.4 Defuzzification

Since the fuzzy results cannot be analyzed, defuzzification converts the fuzzy quantities into crisp quantity for further processing. In general, defuzzification reduces the collection of membership function values into a quantity. There are various methods provided by Sivanandam et al. (Sivanandam, Sumathi et al. 2007), the following show the most common three methods:

Figure 2.13 Defuzzification

1. *Max-membership principle*: This method determines the value x^* by selecting the membership function with the maximum value.

$$\mu_A(x^*) \geq \mu_A(x) \text{ for all } x \in X \quad (2-58)$$

2. *Centroid method*: This is the most often applied method which is also known as center of gravity or center of area method. The value x^* is determined by the center of gravity.

$$x^* = \frac{\int x \cdot \mu_A(x) dx}{\int \mu_A(x) dx} \quad (2-59)$$

3. *Weighted average method*: This method can be used only for symmetrical membership functions. The largest membership values of each fuzzy set form

the weights corresponded each mean of fuzzy set.

$$x^* = \frac{\sum \bar{x} \cdot \mu_A(\bar{x})}{\sum \mu_A(\bar{x})} \quad (2-60)$$

Numerical Example:

In Figure 2.13 (c),

$$x^* = \frac{a \cdot (0.9) + b \cdot (0.6)}{0.9 + 0.6}$$

2.4.1.5 Fuzzy Inference

Fuzzy inference is a process of analysis through fuzzy logic. The process converts an input (usually a linguistic description of expert knowledge) into an output through fuzzy computation. In general, fuzzy inference is composed of the following steps:

1. Input fuzzification (including fuzzy set and membership function)
2. Application of fuzzy operations on rule setting
3. Output defuzzification
4. Decision making

2.4.2 Artificial Neural Network

Artificial Neural Networks (ANNs) is an intelligent computational process which simulates the human neural network. Through training the network by the data, the network can accumulate past experiences. ANNs provides a framework to drive the data. Compared with the traditional computation method, ANNs has the better ability to solve a nonlinear problem and the problem which is hard to define the mathematical equations.

The simplest ANN is composed of three layers, input layer, hidden layer, and output layer. Each layer has more than one neuron. A neuron is linked by a weight to every neighboring neuron. Learning is to adjust the weights in the network by the

historical data. This section depicts the artificial neuron, the learning algorithm, and the framework of an ANN. The theoretical portion is based on the description of Chang and Chang (Chang and Chang 2007).

2.4.2.1 Artificial Neurons

An artificial neuron is composed of three main parts, weight, input processor, and activation function.

1. *Weight*: Weight simulates the intensity of link between different biological neurons. The larger weight indicates the stronger link, while the smaller weight means the weaker link.
2. *Input processor*: Input processor simulates how the biological neurons are affected by the stimulation. The simulation is implemented through weighting the input data and then doing further calculation.
3. *Activation function*: Activation function converts the summation of the input information into an output value which is normalized in range [0, 1] or [-1, 1]. There several activation functions, such as threshold function, piece-wise linear function, sigmoid function, and hyperbolic function.

A neuron j can be expressed as the two equations:

$$\text{net}_j = \sum_{i=1}^m w_{ji}x_i + b_j \quad (2-61)$$

$$y_j = F(\text{net}_j) \quad (2-62)$$

Where y_j is the output of the j^{th} neuron, b_j is a bias, net_j is the summation of weighted input and bias, F is the activation function, and w_{ji} is the linkage of the i^{th} input and the j^{th} neuron.

Figure 2.14 Structure of artificial neuron

2.4.2.2 Learning Algorithm

To an ANN, learning means to adjust the weights, that is, training the network. By means of adjusting the weights between the neurons, ANNs stores the knowledge in the linkage weight. After the training is finished, the network contains the knowledge which has certain ability to infer the result of an input.

There are two categories of learning algorithm, supervised learning and unsupervised learning:

1. *Supervised learning*: In supervised learning, each training data includes both an input and an objective output. Each objective output plays an important role to adjust the weights based on the difference between the output of the network and the objective output. Through repeated the training and adjusting the weights, the difference between the output of the network and the objective value will be smaller. When the difference is smaller than a critical value, the learning could be stopped.
2. *Unsupervised learning*: In unsupervised learning, each training data only includes an input without any output. According to the characteristic of the input vector, the network will learn and adjust the weights itself. In fact, this learning algorithm is often applied in clustering. When the feature of the outputs is unobvious, or the inputs are massy and lack of organized, unsupervised learning

algorithm is a promising method.

(a) Supervised learning algorithm

(a) Un-supervised learning algorithm

Figure 2.15 Learning algorithm (Chang and Chang 2007)

2.4.2.3 Framework

The framework and the size of an ANN may affect its learning ability. In general, if the number of the neurons is not enough, the ANN cannot solve a complicate problem; if the number of the neurons is too high, over fitting problem will occur. However, the only method to decide the framework is by trial and error.

There are two categories of framework, feedforward networks and feedback networks. In the feedforward network, the information is passed through a single direction. “Feedforward” means no lateral connections exist between the artificial neurons in a given layer and the information flow does not go back to previous layers; In the feedback network, the information is passed through multiple directions. “Feedback” means the lateral connections may exist between the artificial neurons in a given layer and the information flow may go back to previous layers.

Figure 2.16 Feedforward network (Chang and Chang 2007)

Figure 2.17 Feedback network (Chang and Chang 2007)

2.4.3 Adaptive Network-Based Fuzzy Inference System (ANFIS)

Jang (1993) proposed the Adaptive Network-Based Fuzzy Inference System (ANFIS) (Jang 1993; Chang and Chang 2007). The ANFIS is a systematized framework which is combined the fuzzy algorithm and neural network. Based on the characteristics of a rust image whose mild rust color is hard to define, ANFIS is an

ideal network to develop this research.

2.4.3.1 Framework of ANFIS

Generally speaking, ANFIS utilizes neural network to train a fuzzy inference system. ANFIS has supervised learning capability. An adaptive network refers to part or all of the nodes are adaptive, which means their outputs depend on the parameters pertaining to these nodes, and the learning rules specifies how these parameters should be changed to minimize a prescribed error measure. Jang proposed a hybrid learning rule which combines the gradient method and least squares estimate (LSE) to enhance the performance of network. In order to achieve a desired input-output mapping, the parameters are updated according to the given training data. Figure 2.18 shows the structure of the ANFIS composed of five layers, and note that a square node (adaptive node) has parameters while a circle node (fixed node) has no parameters.

Figure 2.18 Structure of ANFIS (Chang and Chang 2007)

Layer 1: This is an input layer. Given the input x with N dimension, classify the data into M classes. Every node j in this layer is with a membership function $O_{1,ji}$:

$$O_{1,ji} = \mu_j(x_i) = \frac{1}{1 + \left[\left(\frac{x_i - c_{ji}}{a_{ji}} \right)^2 \right]^{b_{ji}}}, \text{ for } i = 1, 2, \dots, N; j = 1, 2, \dots, M \quad (2-63)$$

where $\{a, b, c\}$ is the parameter set of the membership function.

Layer 2: This is a rule setting layer. Through the permutation and combination of the fuzzy set of the input, there are $M^N = P$ rules. Therefore, there are P neurons in this layer to do the fuzzy logic operations which is often used “AND”.

$$O_{2,p} = w_p = \prod_{i=1}^N \mu_{ji}(x_i), \text{ for } j_i = 1, 2, \dots, M_i; p = 1, 2, \dots, P \quad (2-64)$$

Layer 3: It is a normalized layer.

$$O_{3,p} = \bar{w}_p = \frac{w_p}{\sum_{p=1}^P w_p}, \text{ where it is constrained in } [0,1] \quad (2-65)$$

Layer 4: This is the inference layer. The node function is

$$O_{4,p} = \bar{w}_p f_p = \bar{w}_p \left(\sum_{i=0}^N r_{pi} x_i \right), \text{ for } x_0 = 1 \quad (2-66)$$

where r_{pi} is the parameter set. Parameters in this layer will be considered as consequent parameters.

Layer 5: This is an output layer to summarize all incoming signals.

$$O_{5,1} = \sum_{p=1}^P \bar{w}_p f_p = \frac{\sum_{p=1}^P w_p f_p}{\sum_{p=1}^P w_p} \quad (2-67)$$

2.4.3.2 Learning rule

The basic learning rule is based on the gradient decent and the chain rule, which is proposed by Werbos in the 1970s (Werbos 1974). Since the gradient method is notorious for its slowness and tendency to become trapped in local minimum, Chang proposed a hybrid learning rule which combines the gradient method and least squares estimate (LSE) to speed up the learning process.

Assume the adaptive network has one output

$$\text{Output} = F(I, S) \quad (2-68)$$

where I is the set of input variables and S is the set of parameters. If S can be decomposed into two sets $S = \{S1, S2\}$, and there exist a function H that the composite function $H \circ F$ is linear in the element of $S2$, then upon applying H to equation. 2-68

$$H(\text{output}) = H \circ F(I, S) \quad (2-69)$$

The hybrid rule can be decomposed into two parts based on the data flow direction. Along the feedforward direction, the parameters $S2$ are updated by least squares estimate. On the other hand, along the backward pass, the remaining parameters, $S1$, are updated by the gradient method.

Least squares estimate (LSE):

Now the value of S_1 is given. P training data entries are plugged into equation 2-69 and we obtain a matrix equation:

$$AX=B \quad (2-70)$$

where X is unknown vector whose elements are parameters in S_2 . Set $|S_2|=M$, and the dimension of A , X , B are $P \times M$, $M \times 1$, and $P \times 1$ respectively. Usually P is greater than M , so this is an over-determined problem and there is no exact solution. Instead, a LSE of X , X^* , is sought to minimize the square error $\|AX-B\|^2$.

$$X^*=(A^T A)^{-1} A^T B \quad (2-71)$$

where A^T is the transpose of A , $(A^T A)^{-1} A^T$ is the pseudo-inverse of A if ATA is non-singular. The formula is expensive in computation with matrix inverse and ill-defined if ATA is singular. As a result, Chang employed sequential formulas to compute the LSE of X , and then X can be calculated iteratively:

$$X_{i+1} = X_i + S_{i+1} a_{i+1} (b_{i+1}^T - a_{i+1}^T X_i) \quad (2-72)$$

$$S_{i+1} = S_i - \frac{S_i a_{i+1} a_{i+1}^T S_i}{1 + a_{i+1}^T S_i a_{i+1}}, \quad i=0, 1, \dots, P-1 \quad (2-73)$$

The initial conditions are $X_0=0$ and $S_0=\gamma I$, where γ is a large positive number and I is the identity matrix (or unit matrix) of dimension $M \times M$. Let the i th row vector of matrix A be a_i^T , and the i th element of matrix B be b_i . S_i is called covariance matrix, and the least squares estimate X^* is equal to X_P .

Gradient method:

Suppose that an adaptive network has L layers and the k th layer has $\#(k)$ nodes. We denote the i th position of the k th layer by (k,i) , and its output is O_i^k . Since the output of a node depends on its incoming signals and its parameters, we have

$$O_i^k = O_i^k(O_1^{k-1}, \dots, O_{\#(k-1)}^{k-1}, a, b, c, \dots), \text{ where } a, b, c, \text{ etc. are the parameters pertaining}$$

to this node.

Given a training set with P entries, we can define the error measure for the p th ($1 \leq p \leq P$) entry of training data entry as the sum of squared errors:

$$E_p = \sum_{m=1}^{\#(L)} (T_{m,p} - O_{m,p}^L)^2 \quad (2-74)$$

The overall error measure is $E = \sum_{p=1}^P E_p$.

In order to implement gradient decent in E over the parameters space, first we have to calculate the error rate $\frac{\partial E_p}{\partial O}$ for p th training data and for each node output O .

The error rate for the output node at (L, i) can be calculated from equation 2-73:

$$\frac{\partial E_p}{\partial O_{i,p}^L} = -2(T_{i,p} - O_{i,p}^L) \quad (2-75)$$

The error rate for the internal node at (k, i) can be derived by the chain rule

$$\frac{\partial E_p}{\partial O_{i,p}^L} = \sum_{m=1}^{\#(k+1)} \frac{\partial E_p}{\partial O_{m,p}^{k+1}} \frac{\partial O_{m,p}^{k+1}}{\partial O_{i,p}^k} \quad (2-76)$$

where $1 \leq k \leq L-1$. The error rate of an internal node can be expressed as a linear combination of the error rate of the nodes in the next layer.

Now if α is a parameter of an adaptive network, we have

$$\frac{\partial E_p}{\partial \alpha} = \sum_{O^* \in S} \frac{\partial E_p}{\partial O^*} \frac{\partial O^*}{\partial \alpha} \quad (2-77)$$

where S is the set of nodes whose output depend on α . The overall error measure E with respect to α is

$$\frac{\partial E}{\partial \alpha} = \sum_{p=1}^P \frac{\partial E_p}{\partial \alpha} \quad (2-78)$$

Accordingly, the update formula for α is

$$\Delta \alpha = -\eta \frac{\partial E}{\partial \alpha} \quad (2-79)$$

$$\eta = \frac{k}{\sqrt{\sum_{\alpha} \left(\frac{\partial E}{\partial \alpha} \right)^2}} \quad (2-80)$$

where η is a learning rate, and k is the step size. The step size can be changed to speed up convergence. In Matlab, the learning rate and the step size can be adjusted, but the

setting has no significant affection to the training result. Generally speaking, if k is small, convergence will be slow since the gradient must be calculated many times. On the other hand, if k is large, convergence will initially very fast but will oscillate around the optimum. Therefore, the ANFIS updates k according to the following two heuristic rules as shown in Figure 2.19:

1. If the error measure undergoes four consecutive reductions, increase k by $X\%$.
2. If the error measure undergoes two consecutive combinations of one increase and one reduction, that may probably be an oscillation, decrease k by $X\%$.

where $X\%$ can be controlled in Matlab.

Figure 2.19 Two heuristic rules for updating step size k (From (Jang 1993), figure 10)

2.4.3.3 Sugeno-type fuzzy inference system

Takagi and Sugeno's fuzzy if-then rules are used (Takagi and Sugeno 1983) for ANFIS system in Matlab. The output of each rule is a linear combination of input variables plus a constant term, and the final output is the weighted average of each rule's output. For simplicity, we assume there are only two rules for Figure 2.20. Figure 2.20(a) utilizes a two-rule two-input fuzzy inference system to show the fuzzy reasoning, and Figure 2.20(b) shows the corresponding equivalent ANFIS architecture.

Figure 2.20 (a) Sugeno fuzzy reasoning. (b) Equivalent ANFIS (from (Jang 1993) fig.

4)

2.5 Hilbert-Huang Transform

2.5.1 Hilbert-Huang Transform

Hilbert-Huang transform (HHT) contains two parts, empirical mode decomposition (EMD) and the Hilbert spectral analysis (HSA). A signal first decomposes by EMD to get intrinsic mode functions (IMF) in order to get accurate instantaneous frequency which can be derived by HSA. IMF is a function with the same number of extrema and zero crossing, and is symmetric with respect to zero.

HHT is a non-linear, non-stationary and stochastic process, while wavelet transform is a linear, non-stationary process. Huang pointed out that a wrong instantaneous frequency was because of the curve does not has zero mean, or because of the curve is not symmetric with respect to the zero axis. The observation motivated

the EMD process. The EMD is a sifting procedure (see Figure 2.21) :

1. Identify all the local extrema and then connect all the local maxima by a cubic spline line as the upper envelope and all the local minima as the lower envelope.

The mean of the two envelopes is designated as $m1$:

$$x(t)-m1=h1$$

2. The $h1$ should satisfy the definition of IMF, if not then

$h1-m11=h11$, where $m11$ is the mean of upper and lower envelopes of $h1$. Iterate step2 until $h1(k-1)-m1k=h1k$, where $h1k$ is the first IMF.

Figure 2.21 The cubic spline upper and the lower envelopes and their mean, $m1$.

(Huang 2005)

The effects of the iterations make the mean approach zero, and also make amplitude variations of the individual waves more even. Yet the variation of the amplitude represents the physical meaning. The most important issue is to avoid over sifting. To attain the balance of achieving a reasonable small mean and also retaining enough physical meaning in the result component, “stoppage criteria” should be set to determine the number of sifting steps to produce an IMF. The stoppage criteria suggested by Huang are as follows:

1. Sum of difference (SD): the sifting will stop when SD is smaller than a pre-assigned value.

$$SD = \sum_{t=0}^T \frac{|h_{k-1}(t) - h_k(t)|^2}{h_{k-1}^2(t)} \quad (2-81)$$

2. The number of zero-crossings and extrema are equal or at most differing by one.

As the first IMF c_1 is obtained, we can separate c_1 from the rest of the data by $X(t) - c_1 = r_1$. Since the residual, r_1 , still contains information with longer periods, it is treated as the new data and sifted repeatedly.

$$r_1 - c_2 = r_2,$$

$$r_2 - c_3 = r_3,$$

...

$$r_{n-1} - c_n = r_n$$

$$X(t) = \sum_{j=1}^n c_j + r_n \quad (2-82)$$

Then we achieve a decomposition of the data into n IMF modes, and a residual r_n which can be either a constant, a monotonic mean trend, or a curve having only one extrema.

Bhagavatula (2007) directly applied EMD to remove illumination and improve face recognition (Bhagavatula and Savvides 2007). In order to convert a two dimensional grey image into one dimensional data, the author stringed every row or column of an image. This paper points out that “the last two IMF’s contains the majority of the illumination effects”.

2.5.2 Bidimensional Empirical Mode Decomposition (BEMD)

To image processing, data domain is two dimensions. Fortunately, Nunes (2003) proposed bidimensional empirical mode decomposition (BEMD) to decompose a grayscale image (Nunes, Bouaoune et al. 2003; Nunes, Niang et al. 2003; Nunes, Guyot et al. 2005). Nunes also suggested that BEMD can extract inhomogeneous illumination (Nunes, Bouaoune et al. 2003). Based on Nune’s work, the BEMD

adopted in this section has three major parts, extrema detection, surface interpolation, and stoppage criteria.

Extrema detection : Morphological Reconstruction

The result of local extrema detection may be very different due to the different criteria. Sometimes the small vibration is neglected, but sometimes it is an important cue. Morphological reconstruction provides an easily manipulated method to set the criteria of local extrema detection, and the relevant references could be found in (Vincent 1993; Nunes, Guyot et al. 2005; The MathWorks 2008). The morphological reconstruction contains two components, mask and marker (see Figure 2.22). The mask is the original image, and the marker is usually derived from the mask. In this research, the marker is obtained from mask minus one unit. Through repeating dilations of the marker which is constrained by the mask, the result is shown as the blue wave in Figure 2.22. To a two dimensional grey image, the dilation is processed in four neighbors. Finally the local extrema locates on the pixel whose difference between mask and the blue wave is still one unit.

Figure 2.22 Repeated Dilations of Marker Image, Constrained by Mask (The MathWorks 2008)

Surface interpolation (envelope): radial basis functions (RBFs)

After local extrema are obtained, the next step is to approximate the upper envelope and lower envelope. In Matlab, there are several surface interpolation methods for approximating an envelope. We tried surface interpolation by griddata and thin plate, but we find out that the extrema do not fit the surface well. After following Nunes to apply the radial basis functions (RBFs), the extrema locate on the approximated surface. Figure 2.23 shows an example of 50*50 grayscale image surface interpolation by extrema.

Figure 2.23 Surface interpolation by RBFs

Surface interpolation by RBFs is proposed by Carr (1997) to reconstruct medical image (Carr, Fright et al. 1997). The RBFs approximation form is:

$$s(x) = p_m(X) + \sum_{i=1}^n \lambda_i \phi(\|x - x_i\|) \quad (2-83)$$

where p_m is a low-degree polynomial or not present and m denotes the polynomial degree, λ_i is the RBFs coefficient, Φ is a fix function, and $\|\cdot\|$ denotes the Euclidean norm. If there are n extrema points, the approximation could be obtained by solving the linear system:

$$\begin{bmatrix} A & Q \\ Q^T & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ c \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix} \quad (2-84)$$

where $A = (a_{ij}) = (\phi(\|x_i - x_j\|))$, $\langle n \times n \rangle$

$$\phi(r) = r^2 \log r, \quad \text{thin-plate spline} \quad (2-85)$$

$$Q = \begin{bmatrix} 1x_1y_1 \\ 1x_2y_2 \\ \vdots \\ 1x_ny_n \end{bmatrix}, \quad \langle n \times 3 \rangle \quad (2-86)$$

$$\lambda = (\lambda_1, \lambda_2, \dots, \lambda_n)^T, \quad \langle n \times 1 \rangle \quad (2-87)$$

$$c = (c_0, c_1, c_2)^T, \quad \langle 3 \times 1 \rangle \quad (2-88)$$

$$p_1(x) = c_0 + c_1x + c_2y \quad (2-89)$$

$$f = (f_1, f_2, \dots, f_n)^T, \quad <n*1> \quad (2-90)$$

and $Q^T\lambda=0$ is the side condition.

Stoppage criteria: zero mean or $SD < 0.001$

According to Huang's HHT, an IMF is a function with the same number of extrema and zero crossing, and is symmetric with respect to zero. Unfortunately, the criteria are hard to measure for surface. Therefore, in this research an IMF is a surface with zero mean or $SD < 0.001$. Through processing more than a hundred of $100*100$ rust images, usually the IMF is controlled by criteria of zero mean.

2.6 Morphological Image Processing

Morphological image processing is a powerful tool for many computer-vision tasks in grayscale images, such as shape detection, edge detection, skeletonization, segmentation, and pattern recognition (Ortiz, Torres et al. 2002). Morphological image processing utilizes a structuring element (SE) to process all over an image. Through the design of structuring element and morphological operations, objective features of an image could be extracted. This section introduces some operations (Kasperek 2001).

2.6.1 Dilation and Erosion

The dilation and the erosion are the fundamental morphological operations. The dilation of an image X by a structuring B is denoted as $\delta_B(X)$. The dilation of X is defined as the locus of points x which is the origin of B , when B hits X .

$$\delta_B(X) = \{x | B_x \cap X \neq \emptyset\} \quad (2-91)$$

Hence, the dilated value to a specific pixel is the maximum value of the structuring element when its origin is at x .

$$[\delta_B(f)](x) = \max_{b \in B} f(x + b) \quad (2-92)$$

The erosion operation is similar to the dilation. The erosion of an image X by a structuring B is denoted as $\varepsilon_B(X)$. The erosion of X is defined as the locus of points x which is the origin of B , when B is included in X .

$$\varepsilon_B(X) = \{x | B_x \subseteq X\} \quad (2-93)$$

On the contrary, the eroded value to a specific pixel is the minimum value of the structuring element when its origin is at x .

$$[\varepsilon_B(f)](x) = \min_{b \in B} f(x + b) \quad (2-94)$$

Figure 2.24 displays the application of the fundamental morphological operations by a 7×7 SE.

Figure 2.24 Application of dilation and erosion on a grayscale image

2.6.2 Opening, Closing, and TopHat Transform

Based on the fundamental morphological operations, there are more complicated morphological transforms. This section introduces three useful transforms, the opening, closing, and TopHat transform.

The $\gamma_B(f)$ is the opening of image f by a structuring element B , and it is defined that an image is firstly eroded by structuring element B and then dilated by the transposed B , denoted as $\hat{B}$.

$$\gamma_B(f) = \delta_{\hat{B}}[\varepsilon_B(f)] \quad (2-95)$$

The $\Phi_B(f)$ is the closing of image f by a structuring element B , and it is defined that an image is firstly dilated by structuring element B and then eroded by the

transposed B, denoted as $\hat{B}$.

$$\phi_B(f) = \varepsilon_{\hat{B}}[\delta_B(f)] \quad (2-96)$$

Figure 2.25 displays the application of the opening and closing transform by a 7*7 SE.

Figure 2.25 Application of opening and closing on a grayscale image

The TopHat transform utilizes the opening and closing transform to extract features. There are White TopHat (WTH) and Black TopHat (BTH). The WTH can extract the light features from an image, and it is defined as the difference between the original image f and its result of opening.

$$\text{WTH}(f) = f - \gamma(f) \quad (2-97)$$

On the contrary, the BTH can extract dark details from an image, and it is defined as the difference between the closing of an original image and the original image.

$$\text{BTH}(f) = \phi(f) - f \quad (2-98)$$

Furthermore, contrast enhancement could be done by the combination of TopHat transform. By adding the WTH to the original image and then subtracting the BTH, the intensity variety will be sharpened. Figure 2.26 shows the TopHat transform contrast correction using 3 by 3 structuring element.

Figure 2.26 TopHat transform contrast correction(Kasperek 2001)

CHAPTER 3 COMPARISON OF COLOR CONFIGURATION USING THE K-MEANS ALGORITHM

The first step to do the following assessment is to define the rust color for this thesis in advance. Since there is no perfect color space which can be universally used in image processing, selecting the best color space for this research is important. In order to quantify the performance of each color space, artificial rust images are necessary. This chapter first introduces the artificial rust images produced for this research. Then the performance of each color space is evaluated by the K-Means algorithm based on the two kinds of artificial rust images, uniformly illuminated images and non-uniformly illuminated images. Finally, the best color space $L^*a^*b^*$ will be discussed in the last section.

3.1 Characteristic of Rust Images

A successful image processing starts from analyzing the feature of the objective. Since the objective of this report is to recognize rust degree (or intensity) and rust percentage of steel bridge surface, the relationship between rust and background should be analyzed in advance. This section discusses the characteristic of rust images.

3.1.1 Definition of a rust color

In the clustering method, initial points including a rust color and a background color are necessary to stabilize and enhance the clustering result. In order to set the initial point, the representative rust color should be found out in advance.

According to the scatter (Figure 3.1) of the rust color which is cut from real rust images in RGB and HSI color space, it is easier to define a rust color in HSI color

space. In the HSI color space, obviously there are two clusters separately lying on the SI plane and HS plane.

Figure 3.1 Scatter of rust color in HSI color space and RGB color space

Figure 3.2 Partition rust color into two parts based on $H=50$ in the HSI color space

HSI color space is chosen to describe the rust color. Figure 3.2 is parted into two groups based on $H=50$, and the mean and standard deviation are calculated. The result of the partition is shown in the figure. The mean and standard deviation of the blue circle marks are: $m1=(24.4909, 0.4646, 0.4471)$, $s1=(6.5469, 0.1498, 0.1442)$. The mean and standard deviation of the red crosses marks are: $m1=(257.0518, 0.5307,$

0.0251) , $s1=(61.523, 0.1974, 0.0182)$.

After showing the color of the two groups as in RGB color by appropriate software, the blue circle marks represent different shades of brown and the red crosses represent black. Therefore, the rust color for the K-Means initial point is defined as $m1, (H,S,I)=(24.4909, 0.4646, 0.4471)$.

Note that the range of $m1 \pm s1$ in the HSI color space shows the brown color. Converting the two values, $m1$ and $s1$, to RGB color space directly shows the range other than brown. Therefore, if the initial point is set in a range, it must be in the HSI color space.

In summary, the rust color is depicted by mean and variance in the HSI color space:

$$\begin{aligned} \text{mean } (H, S, I) &= (24.4909, 0.4646, 0.4471) \equiv (R, G, B) = \\ &(173.22, 107.76, 60.54) \end{aligned} \quad (3-1)$$

$$\text{standard deviation } (H, S, I) = (6.5469, 0.1498, 0.1442) \quad (3-2)$$

3.1.2 Why not K-Means

Although the K-Means algorithm is considered as one of the simplest and most promising method in previous research, the algorithm in nature is hard to interpret the gradual change area between the rust and the background.

In MATLAB, the K-Means algorithm is stored in the library. In this report, all the codes are implemented in MATLAB. The following discussion is based on the default K-Means algorithm in MATLAB.

Figure 3.3 Rust image and its scatter in the $L^*a^*b^*$ color space

Figure 3.4 Segmentation results of K-Means

Figure 3.3 shows that there is no obvious distinction between rust color and background color. The results of directly using the K-Means algorithm to segment the rust image into two groups is negative (showed in Figure 3.4). The mild-rust-color is classified as the background color. One of the implications of Figure 3.4(a) is that the rust color is scattered much more widely than the background color. This implication

is what inspired this thesis to begin with the elimination of the background color. Also, the characteristic of gradual change fits the application of the fuzzy algorithm. Therefore, the fuzzy inference is a promising method to deal with the mild-rust-color in this research.

Although the K-Means algorithm cannot deal well with the mild-rust-color, it is one of the simplest and most effective methods to cluster data (Lee, Chang et al. 2005). Therefore, this method is adopted to measure the performance of different color spaces. The aim of this thesis is to develop a better algorithm than the K-Means algorithm to improve the recognition result.

3.1.3 Principal Component Features

Through magnifying a rust image, such as Figure 3.5, it was observed that determining whether a pixel belongs to rust is difficult. It is hard to recognize through a single pixel, however, the work will be easier by simultaneously considering its neighboring pixels. PCA is an ideal tool to extract a representative feature from several pixels.

Figure 3.5 Magnification of a rust image

One of the research objects is to identify the rust intensity. In this research, the rust intensity is classified into three classes, serious, mild, and non-rust. The

definitions of the three classes are depicted as follows:

- 1. Serious: only contains rust color
- 2. Mild: contains the gradual change of color from rust to background color.
- 3. Non-rust: only contains background color

The following tables aim to prove that the eigenvectors of the PCA could be applied in matching. The symbols v1, v2, v3 are the eigenvectors of the window which is part of a rust image. v1 corresponds to the largest eigenvalue, and v2 corresponds to the second largest eigenvalue, and so forth. From Table 3.1 to Table 3.3, we can find out that the same features shared the similar eigenvectors which can be used for matching.

Table 3.1 Eigenvectors of rust and white and blue background color

Rust 	v3	v2	v1
r	-0.3496	-0.794	0.4974
g	0.8003	0.0229	0.5991
b	-0.4871	0.6075	0.6274
white back			
r	0.6923	0.4411	0.5711
g	0.0456	-0.8166	0.5754
b	-0.7202	0.3723	0.5854
blue back			
r	0.2113	-0.7887	0.5774
g	-0.7887	0.2113	0.5774
b	0.5774	0.5774	0.5774

Table 3.2 Eigenvectors of rust image from blue paint bridge

Image 	v3	v2	v1	Image 	v3	v2	v1	Image 	v3	v2	v1
r	-0.1475	0.9782	-0.1461	r	-0.2241	0.9612	-0.1607	r	-0.2651	0.9628	-0.0521
g	0.849	0.201	0.4887	g	0.8268	0.2748	0.4908	g	0.8364	0.2565	0.4843
b	-0.5074	0.0519	0.8602	b	-0.5159	0.0229	0.8563	b	-0.4797	-0.0848	0.8733

Table 3.3 Eigenvectors of rust image from white paint bridge

Image 	v3	v2	v1	Image 	v3	v2	v1
r	-0.6888	-0.7236	0.0437	r	0.1307	0.9903	-0.0469
g	0.6999	-0.6481	0.3001	g	0.9582	-0.1141	0.2624
b	-0.1888	0.2373	0.9529	b	-0.2545	0.0792	0.9638

3.2 Artificial Rust Image

An artificial rust image shown in Figure 3.6 is made by copy and paste from the real rust images. The process of creating the artificial rust images has three steps:

1. Create a blank 256*256 artificial rust image, and a 256*256 map.
2. Assign the rust color cut from a real rust image to the artificial rust image, and the map corresponded to the same location of the artificial image is marked 1.
3. The remaining blank pixels are filled with background color, and the map is marked 0.

The map is used for quantifying the error of image processing. It is important to

notice that the artificial images should be stored as ‘BMP’ files, not ‘JPG’ files which may change the contents due to compression.

Figure 3.6 Artificial rust image and its map

Through the three steps, an artificial rust image and its corresponding map are produced for further works. In this research, four kinds of background color, blue, brown, green, and khaki shown in Figure 3.7 are adopted to make the artificial rust images. Figure 3.8 shows the ten uniformly illuminated artificial rust images in this research.

Figure 3.7 Background texture of artificial image

Figure 3.8 Uniformly illuminated artificial rust images

In order to test the ability of dealing with non-uniform illumination problem of a color space, four kinds of artificial non-uniformly illuminated rust images are designed. The artificial non-uniformly illuminated rust images are added artificial white light from four corners respectively by means of the Photoshop. Figure 3.9 shows an example.

Figure 3.9 Non-uniformly illuminated artificial rust images

3.3 Evaluation the Best Color Feature for Rust Images

In order to select the best color space from the 14 color spaces described in chapter 2 for this research, this section utilizes the K-Means algorithm to cluster the uniformly illuminated and non-uniformly illuminated artificial rust images. The error is quantified based on the differences between the clustering result and the corresponding map. The error ε is defined as

$$\varepsilon = \frac{N_e}{N} * 100\% \quad (3-3)$$

where N_e denotes the number of differences between the clustering result and the map, N refers the total number of pixels. Therefore, the accuracy α is defined as

$$\alpha = 100\% - \varepsilon \quad (3-4)$$

3.3.1 Uniform Illumination

Through clustering the 10 uniformly illuminated artificial rust images (see figure 3.8), the color configurations whose accuracy is higher than 95% in all 10 images are shown in Table 3.4.

Following the artificial rust images assessment is the real rust images evaluation. Since the real rust images are still different from the artificial rust images, it is necessary to cluster the real rust images to evaluate the performance of a color space. Therefore, the next step is to cluster 12 real rust images in all the better configurations. The experimental results show that only I2I3, V, UV, I, IQ, Cr, CbCr, CgCr, $L^*a^*b^*$, L^*b^* , a^*b^* , $W^*U^*V^*$, U^* , W^*U^* , W^*V^* , U^*V^* , $L^*u^*v^*$, u^* , L^*u^* , u^*v^* have good results, and thus the ranking is obtained:

$$W^*U^*V^* = U^* = W^*U^* = U^*V^* = L^*u^*v^* = u^* = L^*u^* = u^*v^* > W^*V^* > Lab = ab > Lb = CgCr > Cr = V > UV = CbCr = I = IQ$$

Table 3.4 Performance evaluation of color spaces by uniformly illuminated artificial
rust images

Color Space	The color configurations whose accuracy is higher than 95%
RGB	RGB, B, RB
I1I2I3	I1I2I3, I1I2, I2I3
YUV	YUV, V, UV
YIQ	YIQ, I, YI, IQ
YCbCr	YCbCr, Cr, CbCr
YCgCr	Cr, CgCr
XYZ	Z, XZ
$W^*U^*V^*$	$W^*U^*V^*$, U^* , W^*U^* , W^*V^* , U^*V^*
$L^*u^*v^*$	$L^*u^*v^*$, u^* , L^*u^* , u^*v^*
$L^*a^*b^*$	$L^*a^*b^*$, L^*b^* , a^*b^*

3.3.2 Non-uniform Illumination

By means of clustering the 40 non-uniformly illuminated artificial rust images, it is proven that the following color configurations have certain ability to deal with non-uniform illumination problem. The color configurations whose accuracy is higher than 90% in all 40 images are shown in Table 3.5.

The next step is to cluster the real rust images which are added artificial non-uniform light. If the color space can deal with the non-uniform illumination problem, the result of clusters should not be affected by the light pattern. The experimental results show that only the 6 color configurations have certain ability to deal with the problem arisen from illumination: $L^*a^*b^*$, a^*b^* , U^* , U^*V^* , u^* , u^*v^* .

Table 3.5 Performance evaluation of color spaces by non-uniformly illuminated
artificial rust images

Color Space	The color configurations whose accuracy is higher than 90%
$W^*U^*V^*$	$W^*U^*V^*$, U^* , W^*U^* , U^*V^*
$L^*u^*v^*$	$L^*u^*v^*$, u^* , L^*u^* , u^*v^*
$L^*a^*b^*$	$L^*a^*b^*$, a^*b^*

3.3.3 Summary

The section aims to select the best color space or color configuration for this research. The K-Means algorithm is adopted in this section. Through clustering artificial and real rust images with uniform and non-uniform illumination, the $L^*a^*b^*$, a^*b^* , U^* , U^*V^* , u^* , and u^*v^* are considered as the best color configurations.

Usually, the rust images are stored as RGB images. If we want to process an image in different color space, the image has to be converted into the particular color space based on the RGB values. However, the $W^*U^*V^*$ and $L^*u^*v^*$ color space exists the singular problem arisen from the definition of conversion. Therefore, the best color space in this research is the $L^*a^*b^*$ color space, since its denominator is reference white (non-zero value) that would avoid the singular problem. Finally, the a^*b^* configuration is selected due to its low dimension, which can shorten the processing time, to develop the models in this research.

3.4 Test the Power of $L^*a^*b^*$

According to the previous sections in this chapter, the $L^*a^*b^*$ color space is the first candidate in this research. Since one of the objects is to overcome the non-uniform illumination problem, direct use of the a^*b^* color configuration to filter light is an intuitive and rational method. The section intends to quantify the interpretation of L^* .

The ideal conversion from the RGB color space to the $L^*a^*b^*$ color space is reversible. However, through the conversion from the RGB color space to the $L^*a^*b^*$ color space, and then back to the RGB color space (see figure 3.10(b)), the color which theoretically should be the same is different from the original one. The irreversible characteristic can be observed from Figure 3.10. In fact, the transformation is as Figure 3.11.

Figure 3.10 Color difference between non-transformation and transformation

Figure 3.11 Irreversibility of transformation from RGB to $L^*a^*b^*$ to RGB

If we want to quantify the power of L^* extracting the light part, the two compared images, one without additional light and the other with additional light, should only pass from RGB to $L^*a^*b^*$ once. The error is directly quantified on the a^*b^* plane. Also, we have to ensure the intensity range of relatively uniformly illuminated image falls in $RGB=[1\ 254]$ and $L=[1,99]$, which means the color is well defined. If it is out of the range, some of the colors cannot be defined. The next step is to use the software photoshop to add artificial white light, which contains light and shade. The new image should also fall in the previously mentioned range.

Figure 3.12 Rust images for quantifying the power of L^* filtering the light

In order to quantify the power of L^* filtering the light, the samples for the test should be designed according to the last paragraph. The test of the power of L^* contains the following steps:

1. Prepare a rust image. The Image_ori is the copy of the rust image, and the Image_light is also the copy of the rust image but with additional light and shade (see Figure 3.12).
2. Convert the two images into $L^*a^*b^*$ space, and then subtract the Image_ori from the Image_light.

After the subtraction, in the L^* configuration we should get only the light and shade pattern; in the a^*b^* configuration we should get nothing. It is easier to quantify the error by the change of (a^*, b^*) . The definition of the error is presented from equation 3-5 to 3-9, and Figure 3.13 shows an example.

$$a_s = \text{Image\_light}(a^*) - \text{Image\_ori}(a^*) \quad (3-5)$$

$$b_s = \text{Image\_light}(b^*) - \text{Image\_ori}(b^*) \quad (3-6)$$

Since the range of $a^* = [-100, 100]$, $b^* = [-100, 100]$,

$$a_{\text{error}} = a_s / 200 * 100\% \quad (3-7)$$

$$b_{\text{error}} = b_s / 200 * 100\% \quad (3-8)$$

the change on the a^*b^* plane can be quantified as

$$ab\_error = \sqrt{a\_error^2 + b\_error^2} \quad (3-9)$$

Figure 3.13 The performance of the light filter L^*

Through computing by equation 3-5 to 3-9, the minimum error of Figure 3.13 (c) is around zero and the maximum error is 2.38%.

In summary, through the test in different light intensity, the change on the a^*b^* plane arisen from the change of light is very small, the range is from close to zero to 5%. Therefore, the L^* configuration has superior ability to filter light, and the a^*b^* configuration is adopted in this research.

3.5 Summary of Chapter

This chapter defines a rust color which could be applied in this research. The rust color is utilized as an initial point in the K-Means algorithm. Through processing 50 uniformly and non-uniformly illuminated rust images using the K-Means algorithm in 14 different color spaces, it is found that the $L^*a^*b^*$ color space and the a^*b^* configuration (of the $L^*a^*b^*$ color space) have the best performance in rust image recognition due to their ability in filtering out light effects (or illumination factors). Therefore, the a^*b^* color configuration is adopted in this research.

CHAPTER 4 ADAPTIVE ELLIPSE APPROACH (AEA)

This chapter presents the development and the framework of the Adaptive Ellipse Approach (AEA). The AEA is developed for rust image segmentation based on color homogeneity. The a^*b^* color configuration is adopted in this model. The segmentation results will be compared with the results of the K-Means algorithm.

4.1 Rust Image Preprocessing

Before the implementation of the model, the background color and the rust color should be defined. Also, illumination adjustment should be considered, since the non-uniform illumination problem is always existent in practice.

4.1.1 Automated Detection of Background

To a color rust image, the color of rust area is much more complicated than the background color (see Figure 3.4). This fact inspires this research to start with background color elimination. It has to acknowledge that this model cannot deal with the bridge with brown color paint, since the discriminating criteria is based on chroma (brown or non-brown).

In order to define the area of background color on the a^*b^* plane, this section aims to automatically extract a background area from a rust image. In general, the extraction is based on cutting and selecting, and it could be broken down to the following steps:

Figure 4.1 Process of automated detection of background

1. Divide an image into four equal parts.
2. Select the part whose color mean to the previously defined rust color (in Equation 3-1) is the farthest.
3. Calculate the color entropy En of the part, where $P(a^*, b^*)$ is the probability that will produce the color (a^*, b^*) .

$$En(a^*, b^*) = -\sum P(a^*, b^*) \log_2 P(a^*, b^*) \quad (4-1)$$

4. Repeat step1 to step 3 until the entropy is smaller than 2.5 or the size of the area is smaller than 10×10 .

The termination condition is set based on the analysis of rust images. Most of the entropy of the background texture is below 2.5. Also, in order to define an area of background color, the limitation of the divided size is necessary. Therefore, another terminal criterion ensures the size of the detected background. Figure 4.2 shows the result of the automated detection of background color.

Figure 4.2 Result of automated detection of background

4.1.2 Illumination Adjustment

Non-uniform illumination has always been a problem in image processing and, therefore, its effect should never be neglected. The detection of background colors mentioned in the last section is only based on the available bridge coating images in this research. In order to detect a wide range of background colors, it is necessary to bring illumination adjustment into the context. Illumination adjustment is processed in the RGB color space and is used to adjust the light intensity (or illumination) of rust images so that the light effect on background colors could be better studied and handled.

In the RGB color space, a color will remain the same after multiplying or dividing the three components (R/G/B) by a constant (Cheng, Jiang et al. 2001). Through trials and errors, it is found that the light effect on the background color could be moderately mitigated if the values of the three components (R/G/B) of the background color could be reduced to at least 100 or increased to 255, out of the range of [0, 255]. Therefore, the illumination adjustment is proposed as: (1) identify the largest value (denoted as x) among the three components (R/G/B) of the background color; and (2) multiply the three background color components by $(100/x)$ or by $(255/x)$. Through automated background color detection and illumination adjustment, the complete definition of the background color (with different light intensities considered) could be obtained. Finally,

the background areas on the rust image could be approximated using the background color definition on the a^*b^* plane. Figure 4.3 shows the elimination result of background colors.

Figure 4.3 Result of background elimination after illumination adjustment

4.1.3 Definition of rust color by the fundamental ellipse

Since the adaptive ellipse approach (AEA) is based on the assumption that rust colors of steel bridges are of brown color tones, the segmentation of rust and background colors is according to chrominance similarity. This paper defines the area of rust color tones on the a^*b^* plane using an ellipse, called the fundamental ellipse. The fundamental ellipse is defined according to all the collected rust colors, as shown in Figure 4.4(a). Since non-uniform illumination is always an important issue, illumination adjustment is also considered in this section. In 2000, Adachi et al. proposed to use an ellipse to approximate the flesh color on the UV plane, which is one

of the color configurations of the LUV color space, for face detection (Adachi, Imai et al. 2000). As the rust colors distribution on the a^*b^* plane looks like an ellipse, as shown in Figure 4.4(a), the ellipse approximation is adopted to define the rust colors. The fundamental ellipse is determined by the extrema value along the a^* and the b^* axes. The difference of the maximum and minimum value of the a^* and the b^* decide the minor and major diameters of the fundamental ellipse respectively. The fundamental ellipse is defined as

$$\frac{(a^*-20)^2}{23^2} + \frac{(b^*-27.5)^2}{42.5^2} = 1 \quad (4-2)$$

and displayed in Figure 4.4. In Figure 4.4(b), the fundamental ellipse includes 99.91% of the collected rust colors. The fundamental ellipse shown in Figure 4.4(c) defines 99.89% of the adjusted rust colors using illumination adjustment.

Figure 4.4 Rust colors and the fundamental ellipse

4.2 Adaptive Ellipse Approach

The section details the development of the Adaptive Ellipse Approach (AEA) and its framework.

In section 4.1, automated detection of background defines the background color area, and the fundamental ellipse clarifies the rust colors. Figure 4.5 to Figure 4.7 shows the results of direct application of the two definitions, and the (c) of the three figures display the scatter of color on the a^*b^* color plane. In a scatter, the darker, normal and lighter background colors constitute the region 1; the region 1 is defined by a box shape whose margins are decided by the extreme value of the background colors.

Region 2 is composed of the color which is fallen in the fundamental ellipse. To a scatter, since only the color which falls in region 1 and 2 can be defined, the rest of colors are belonging to region3, called non-defined area.

Figure 4.5 Application of automated detection of background and fundamental ellipse in non-uniformly illuminated rust image with blue paint

Figure 4.6 Application of automated detection of background and fundamental ellipse in uniformly illuminated rust image with blue paint

Figure 4.7 Application of automated detection of background and fundamental ellipse in uniformly illuminated rust image with khaki paint

Although the fundamental ellipse defines the basic rust color, the mild rust colors are still not included. In Figure 4.5(c), 4.6(c), and 4.7(c) the points marked by yellow are the non-defined area which contains both mild rust color and background color. The AEA is focus on the treatment of the non-defined area by means of adjusting the size of the fundamental ellipse.

From the comparison of Figure 4.6 (c) with Figure 4.7 (c), we can find out that the relative relationship between the basic rust color defined by the fundamental ellipse and the background color depends on the paint color. Therefore, each steel bridge should have its own “Adaptive Ellipse” to segment the rust area. The adaptive ellipse is determined by the steps:

1. After automatically defining the background color, there are three groups, background, rust (enclosed by the fundamental ellipse), and the non-defined color.
2. A line is linked by the means of the rust group and the background group, and its intersect with the fundamental ellipse is a new point, called $i1$, for enlarging the fundamental ellipse (see Figure 4.8).

Figure 4.8 Enlargement size is based on the distance between $i1$ and the mean of background color

3. The enlargement percentage of the fundamental ellipse depends on the distance between i_1 and the center of background. Also, the new area of ellipse should not include the background color.

In order to induce the enlargement percentage of the fundamental ellipse, 11 real rust images with different background colors are processed by the previous three steps. Table 5.1 shows the testing results, where symbol X denotes the fail of enlargement, O denotes the success of enlargement, and pink color marks the optimal percentage. Failure or success depends on whether the new ellipse includes the background color or not. All of the results are evaluated by vision. **The optimal enlargement percentages always fall under 50% where is the middle area of the gradual change between rust and background color. Most of the optimal percentages are the second largest success percentage where 50% is the upper limit. Therefore, the enlargement percentage of the AEA is determined by the second largest success percentage and 50% is always set to be the upper limit.**

Table 4.1 Enlargement of fundamental ellipse by trial and error (shade marks the optimal enlargement percentage)

image	80%	70%	60%	50%	40%	30%	20%	10%
1	X	X	X	O	O	O	O	O
2	X	O	O	O	O	O	O	O
3	X	X	X	O	O	O	O	O
4	X	X	X	O	O	O	O	O
5	X	X	X	X	X	X	O	O
6	X	X	X	X	X	X	O	O
7	X	X	X	X	X	X	O	O
8	X	X	X	X	X	O	O	O
9	X	O	O	O	O	O	O	O
10	X	X	O	O	O	O	O	O
11	O	O	O	O	O	O	O	O

Each steel bridge has to run the AEA to produce the adaptive ellipse. Deriving the adaptive ellipse for each steel bridge from a 256*256 rust image spends about 40 seconds. Then this adaptive ellipse can be used to segment the rust area in the same bridge much faster than the K-Means algorithm. Figure 4.9 shows the flowchart of the AEA model.

Figure 4.9 Flowchart of the Adaptive Ellipse Approach

4.3 Discussion and Comparison

4.3.1 Comparison of K-Means in grayscale, RGB, and a^*b^*

In chapter 3, the a^*b^* configuration (of the $L^*a^*b^*$ color space) is found to be the best color coordinate system. Figure 4.10 shows the clustering results of a

uniformly illuminated image sample using the K-Means algorithm under grayscale, RGB, and a^*b^* , respectively. Furthermore, non-uniformly illuminated images are also clustered using K-Means. An example is shown in Figure 4.11. Note that all the results are shown in RGB format. Figure 4.10 shows that both RGB and a^*b^* have better performance than grayscale. Therefore, to produce more accurate recognition results, color information is necessary. Figure 4.11 shows that only the a^*b^* configuration is independent of illumination, which implies that a^*b^* has moderate ability to filter illumination/light factors. In summary, the a^*b^* configuration is found to have the best performance in clustering steel bridge rust images in this paper. Also, all of the results show that the mild rust colors are more likely to be recognized as the background color using the K-Means algorithm.

Figure 4.10 Comparison of the processed results of a uniformly illuminated image using the K-Means algorithm

Figure 4.11 Comparison of the processed results of a non-uniformly illuminated image using the K-Means algorithm

Table 4.2 shows the processing times using the K-Means algorithm for grayscale, RGB, and a^*b^* . Processing an image in a^*b^* takes most time (but still acceptable) due to color space conversion.

Table 4.2 Processing time for a 256x256 rust image using the K-Means algorithm

Time (sec)	Grayscale	RGB	a^*b^*
Average	8.09	11.01	23.23

4.3.2 Comparison of K-Means with AEA

As the K-Means algorithm is one of the most effective segmentation methods in the literature (Lee, Chang et al. 2005), it is selected as the benchmark for the proposed adaptive ellipse approach (AEA). The segmentation result processed by the K-Means algorithm is shown in Figure 4.12. It can be seen that almost all the mild rust colors are considered as the background colors in the K-Means algorithm. This decreases the percentage of rust areas to only 30%, as shown in Figure 4.12(c). This is the

disadvantage of the K-Means algorithm. Generally, the K-Means algorithm can generate good results if the groups of data to be classified are distinct. Since the non-defined colors in this case have vague boundaries with background colors and with rust colors, the segmentation results might be unsatisfactory sometimes. Therefore, a more reliable approach for rust image recognition is required. This is why the adaptive ellipse approach (AEA) is proposed in this paper.

Figure 4.12 Processed results using the K-Means algorithm

Figure 4.13 presents the segmentation result of the rust image in Figure 4.12 using the adaptive ellipse approach (AEA). In the scatter of pixels of a rust image on the a^*b^* plane (Figure 4.13(b)), region 1 denotes the background colors, region 2 indicates the rust colors defined by the fundamental ellipse (Note: The larger ellipse is the adaptive ellipse.), and region 3 (which is inside the adaptive ellipse) implies the mild rust colors defined by AEA. The result in Figure 4.13 shows that AEA has excellent rust recognition ability and could identify a wide range of rust colors. Compared to Figure 4.12(c), Figure 4.13(e) includes the mild rust areas and increases the rust percentage to 65%. The processing time required by AEA is acceptable, as shown in Table 4.3.

Figure 4.13 Processed results using the adaptive ellipse approach (AEA)

Table 4.3 Processing times for a 256x256 rust image on the a^*b^* color plane

Time (sec)	K-Means	AEA
average	23.14	32.32

Although AEA has been proved effective for rust image recognition, there is still a limitation. Since AEA is developed based on color/chrominance similarity on the a^*b^* plane, any color close to brown is considered as rust in this approach. Therefore, if the paint color of steel bridge coating is similar to brown or red colors (i.e., rust colors), the recognition accuracy of AEA will go down. However, it is not a big issue, as in most cases people would choose a paint color distinct from rust colors (or brown/red colors) for steel bridge coating.

4.4 Summary of Chapter

Based on the observation that background colors are much easier to deal with than rust colors, which have a wider coverage of tones and shades of brown (or rust)

colors (from dark rust colors to mild rust colors), the adaptive ellipse approach (AEA) is developed. The fundamental concept is to classify the rust image into three groups: background colors, rust colors, and non-defined colors (such as mild rust colors, etc.). The approach is designed for automatic rust recognition and rust percentage calculation. In AEA, background colors and rust colors defined by the fundamental ellipse could be easily identified, but it takes some effort to process non-defined colors. To classify non-defined colors, the adaptive ellipse, which is enlarged from the fundamental ellipse, is used.

Through experiments, it is proved that the proposed adaptive ellipse approach (AEA) has excellent ability for rust image recognition and could identify a wide range of rust colors. On the contrary, the performance of the popular K-Means algorithm is not as good as that of AEA due to the lack of ability to handle light/illumination factors. Despite all the advantages, AEA has one shortcoming, not being able to properly recognize rust images of brown or red paint colors. However, it would not be a big problem, as usually paints with colors distinct from rust colors will be chosen for steel bridge coating.

**CHAPTER 5 BOX-AND-ELLIPSE-BASED NEURO- FUZZY APPROACH
(BENFA) FOR BRIDGE COATING ASSESSMENT**

Although the adaptive ellipse approach (AEA) provides a method to extract rust color (including mild-rust-color), there is still a shortage. Sometimes the AEA cannot include mild-rust-color well when the color distribution is nearly parallel to the major axis of the fundamental ellipse. As shown in Figure 5.1, this kind of distribution leads to a large non-defined area. Therefore, fuzzy concept is introduced to deal with the gradual change of color from background color to rust color. One of the famous neural fuzzy inference systems, adaptive-network-based fuzzy inference system (ANFIS) (Jang 1993), is applied to build the new proposed model, called box and ellipse-based neuro-fuzzy approach (BENFA). The BENFA applies the automated detection of background and fundamental ellipse of AEA and Fuzzy C-Means (FCM) on ANFIS to automatically recognize serious rust and mild rust area.

Figure 5.1 Result of AEA with enlargement 0.1

5.1 Preparation of Training Data

The first job to start an ANFIS system is to prepare the training data. Since ANFIS is a supervised neural network, each input has to be designated a corresponding output. However, an input data entry to this research is a 256*256 rust image that is too bulky to designate each corresponding output. Therefore, a fast and reasonable method which utilizes the concept of AEA is introduced in this section.

5.1.1 Input Features

The first step is to decide the input of the ANFIS. The candidates of input include chrominance a^* , chrominance b^* , brightness L^* , and the eigenvector of chrominance (a^*, b^*) . The chrominance (a^*, b^*) and brightness L^* are the basic information of a color image. Utilizing eigenvector aims to include the neighboring information.

5.1.1.1 Chrominance of the $L^*a^*b^*$ and its eigenvector

In the first test, we want to decide whether eigenvector is necessary. There are two kinds of input of ANFIS: the first one contains only chrominance feature, (a^*, b^*) , and the second contains chrominance feature (a^*, b^*) and eigenvector of four neighbors. In the first test, three colors of coating, grey, blue, and khaki are included and trained separately, and each ANFIS is trained with three 256*256 rust images. The input-output mapping is produced automatically by the following rules:

1. If the color of pixel falls in the area defined by automated detection background, set the corresponding output as 0.
2. If the color of pixel falls in the fundamental ellipse, set the corresponding output as 1.
3. The remaining pixels which cannot be defined by the previous two rules are called non-defined colors. The non-defined colors are clustered into three groups

by Fuzzy C-Means (FCM), and the one whose center is closest to mean of background is designated as 0.9 while the farthest is designated as 0.1.

Note that each output value ranging from 0 to 1 represents the probability of rust.

After training three ANFIS for grey, blue, and khaki coating separately, we found that training and running the ANFIS trained with eigenvector takes much more time than the one without eigenvector. Figure 6.1 and Table 6.1 shows the result of this test. The result for every image is shown by 3*3 sub-images, and the first one on the left top corner is the original image. The rest eight images are shown with their outputs larger than 0.1, 0.2... to 0.8 respectively. Note that the numbers of membership function are all set as four and used default of the MATLAB.

Figure 5.2 Comparison of the results of ANFIS between training with eigenvector and without eigenvector

Table 5.1 Process time of ANFIS without eigenvector and ANFIS with eigenvector

Time (sec)	ANFIS without eigenvector	ANFIS with eigenvector
training	2~10	400
running	3	75

From Figure 5.2, we can find out that the results from training an ANFIS with eigenvector is not always better than the results form an ANFIS without eigenvector. To the three images in Figure 5.2, only image (b) performs better in the ANFIS training with eigenvector. Also, Table 5.1 shows that the training of ANFIS with eigenvector takes 25 times longer than the training of ANFIS without eigenvector. Therefore, the training input will not include eigenvector of (a^*, b^*) .

5.1.1.2 Chrominance of the $L^*a^*b^*$ and intensity L^*

The second test aims to determine whether the L^* component can resolve the non-uniform illumination problem. Theoretically speaking, the chrominance of a color image will be changed under different intensities of light (see 3.4). The L^* component is expected to control and eliminate the change of chrominance.

In this test, the two ANFIS models are both trained by 36 rust images with size 128*128. The first model contains the features of chrominance a^* and chrominance b^* , and the numbers of membership function are four and four respectively. The second one includes the same features of the first model plus one more feature, the brightness L^* , and the numbers of membership function are four, four, and six respectively. Figure 5.3 shows the results. The result for every image is shown by 3*3 sub-images, and the first one on the left top corner is the original image. The

remaining of eight images are shown with their outputs larger than 0.3, 0.4... to 1 respectively. Note that in this test, the output has opposite setting to the first test. The output of background is set to be 1 while the output of rust is set to be 0. The detail reason of the difference of output will be described in the next section.

Figure 5.3 Comparison of the results between training with (a*, b*) and (a*, b*, L*)

See Figure 5.3, we can find out that if the ANFIS is trained with the L^* component, the output is affected much more seriously by the light pattern than the one trained without the L^* component. Also, training only with a^* and b^* saves much time. Therefore, the inputs of ANFIS are only a^* and b^* .

To summarize the tests of 5.1.1.1 and 5.1.1.2, the input features for the ANFIS model is determined as chrominance a^* and chrominance b^* .

5.1.2 Output Designation

There are two main works in this section. Firstly, we should decide whether to set the rust or the background color as 1, since the output is in the range from 0 to 1. Secondly, design a strategy to automatically assign each input an output value.

5.1.2.1 Output Value Setting

To an ANFIS model, each output should be set a value from zero to one. Based on the observation of Figure 5.4 and Figure 5.5, we can find that although there is no significant difference between setting the output of the background or rust color as 1, but it is easier to explain or recognize the result based on the background color. Since the background color is much simpler than the rust color, evaluating the result from the background color is instinctive. The situation leads us to always eliminate the background color from an image to decide a rust area. Therefore, the training output is one for background and zero for rust.

Figure 5.4 Result with setting rust=1, background=0

(a) Display of the output

(b) Display of the remaining color of (a)

Figure 5.5 Result with setting background=1, rust =0

5.1.2.2 Designation Strategy

In order to simplify the job for creating input and output for ANFIS, the process should be automated. Therefore, automated detection of background is applied in this part. Unfortunately, only an image whose entropy is higher than 2.5, can be processed. Also, an image with entropy higher than but close to 2.5 could not be processed well in this method. Therefore, one threshold should be determined whether an image is suitable for applying automated detection of background method.

First, we apply automated detection of background to about 500 the rust images with size 256×256 , and then check whether the background segmentation contains rust color. All the images which background segmentation contains rust color are collected. After calculation, the maximum value of entropy to the collection is 4.01. Therefore, all the images are classified into two groups by $H=4$, where H refers to entropy of an image. An image whose entropy higher than 4 means it is suitable to use automated background detection. An image with low entropy means the color is simple, that is there is seldom chance of gradual color change between rust and background, so it is grouped into two groups, rust and background, using the K-Means.

All the rust images are classified to two groups according to the entropy:

If $H \leq 4$, use K-Means to designate output

Else ($H > 4$), apply automated detection of background and fundamental ellipse and FCM to assign output

where H refers to entropy of an image.

5.1.2.2.1 Entropy of Chrominance lower than four

$H \leq 4$ represents the color variety is low, so using the K-Means to designate the corresponding output value. If the center of group is closer to the rust defined color (R ,

G, B)=(173.22 107.76 60.54), the group belongs to rust, and every input data in this group is designated an output of 0. The other group belongs to background, and every input data in this group is designated an output of 1. Figure 5.6 shows the designation by RGB images. Figure 5.6(a) is an input entry with 256*256 input data. The K-Means clustering result is displayed by Figure 5.6(b). According to the clustering result, the output designation is shown by RGB images in Figure 5.6(c) and Figure 5.6(d).

Figure 5.6 Designation of output with $H \leq 4$

5.1.2.2 Entropy of Chrominance higher than four

Entropy of Chrominance higher than four, $H > 4$, represents high color variety, so it is meaningful to differentiate rust color into different intensities. Automated detection of background is applied to decide the area of background on the a^*b^* color plane; fundamental ellipse decides the serious rust color.

To an input image, any data that falls in the box area which is formed by automated detection of background method is background color and designated as 1; any data that falls in the fundamental ellipse is serious rust color and designated as 0; the remaining data is clustered by FCM into three groups: probably rust, non-defined

color, probably background, and they are designated as 0.1, 0.5, 0.9, respectively. To an input image, only automated detection of background color and rust color defined by fundamental ellipse are certain, but the remaining data are uncertain. The remaining data may include rust color and background, so the data is further clustered into three groups. The group whose center is closest to the mean of rust color (defined by fundamental ellipse) belongs to probably rust color and is designated as 0.1; the group whose center is farthest to the mean of rust color (defined by fundamental ellipse) belongs to probably background color and is designated as 0.9; the remaining colors could not be defined, so they are designated as 0.5.

Figure 5.7 displays the designation result by RGB images. Figure 5.7(a) is an input entry with 256×256 input data. According to the strategy which is depicted in the last paragraph, the designation of the image in Figure 5.7(a) is scattered on the $a \times b \times$ plane by Figure 5.7(b). Figure 5.7(c) shows the designation result in RGB images.

Figure 5.7 Designation of output with $H > 4$

5.1.3 Illumination Adjustment of Input Image

In order to overcome non-uniformly illumination problem, illumination adjustment is applied for training an ANFIS model. Every pixel of each input image is designated its output by means of the method in 5.1.2, and then the result of designation will be stored as a map.

Illumination adjustment is implemented by multiplying a constant to a RGB image. If a RGB image is multiplied by a constant lower than 1, the image becomes darker as shown from Figure 5.8(b) to Figure 5.8(d); If a RGB image is multiplied by a constant higher than 1, the image becomes brighter as shown from Figure 5.8(e) to Figure 5.8(g). The original input image (Figure 5.8(a)) decides the corresponding output, and then a map is created. The map ensures that the same location has the

same output. According to the map, the input images with illumination adjustment (from Figure 5.8(b) to Figure 5.8(g)) are assigned their corresponding output. The illumination adjustment aims to let the result of rust recognition be independent to different illumination intensities.

Figure 5.8 Input image with illumination adjustment

Figure 5.9 shows that the ANFIS trained with illumination adjustment [0.7~1.3] has the similar result as the ANFIS trained with illumination adjustment [0.7, 0.9, 1, 1.2, 1.3], but better than the ANFIS trained with illumination adjustment [0.7, 1, 1.3], and their training time are 4 hours, 2 hours, and 1 hour respectively. The selection is based on the best or similar performance but with shorter training time. Therefore, each input image IM creates four more images with different brightness, that is IM*0.7, IM*0.9, IM*1.1, IM*1.3.

Figure 5.9 Results from different ANFIS trained with different illumination adjustment combinations

5.1.4 Division of Training Sets

ANFIS is a fuzzy inference system training and running by neural network. As shown in Figure 5.10, we can find out that training with same coatings could get better results. The result implies that the input data should be classified well in advance before training the ANFIS.

(a) Training with different coating's colors, including 3 blues and 9 non-blues.

(b) Training with one coating's color, including 4 blues.

Figure 5.10 Results of training an ANFIS with different and the same coating images

Figure 5.11 Scatters of rust image with different coating's colors

In order to determine how to classify an input data, seven rust images are scattered in Figure 5.11. The color refers to:

Magenta: serious rust color (output=0)

Cyan: probably rust (output=0.1)

Yellow: non-defined color (output=0.5)

Green: probably background (output=0.9)

Blue: background (output=1)

Through the seven scatters in Figure 5.11 we can find out that background color varies more along the b^* axis. Therefore, divide the training image into groups according to their mean b^* of background color. According to the Figure 5.11, the four groups are:

Group1: $\text{mean\_}b^* \leq -10$ (The last row of Figure 5.11)

Gourp2: $-10 < \text{mean\_}b^* \leq 0$ (The third row of Figure 5.11)

Group3: $0 < \text{mean\_}b^* \leq 10$ (The first row of Figure 5.11)

Group4: $10 < \text{mean\_}b^*$ (The second row of Figure 5.11)

Through training with (illumination adjustment, 128×128 , epoch=200)

Group1: $b1 \sim b6$

Gourp2: $gr21, gr23, gr36, w42, w75$

Group3: $w56, w69, w54, w57$

Group4: $w18, w42, w11, w71$

Through training 95 rust images with size 128×128 , and then testing with 12 rust images, the results show good performance (see Figure 5.12).

Figure 5.12 Results of ANFIS with grouping

According to the rust images in hand, images belonging to group four is few, so group four will be combine to group three in this model. Finally, the ANFIS in this research is trained with 256*256 rust images as following:

Group1: b1, b1_adj, b3_adj, b4, b5, b8 (6*5)

Group2: w2, w3, w15, w16, w61, w72, gr8, gr13, gr53 (9*5)

Group3: w4, w11, w18, w37, w39, w45, w54, w58, w69, w74 (10*5)

Each image creates four more images due to illumination adjustment. There are totally 125 images are trained in the ANFIS model.

5.2 Development of Membership Functions

Membership function determines how to explain each input feature, and constructs the rules of fuzzy inference system. The most important settings of

membership function are the shape and number.

5.2.1 Shape of Membership Function

There are eight candidates of shape to ANFIS in Matlab: trimf, trapmf, gaussmf, gauss2mf, gbellmf, psigmf, dsigmf, pimf (MathWorks). These membership functions are classified as following:

1. Formed with straight lines: trimf is the simplest membership function with triangular shape; trapmf is the trapezoidal membership function.
2. Gaussian distribution curve: gaussmf is a simple Gaussian curve, and gauss2mf is a two-sided composite of two different Gaussian curves. The gbellmf is the generalized bell membership function which has one more parameter than the Gaussian membership function. The membership functions have the advantages of being smooth and nonzero at all points.
3. Asymmetric membership function: the previous functions are unable to specify asymmetric membership functions. An asymmetric and close membership function can be synthesized using two sigmoidal functions. The dsigmf is created by the difference between two sigmoidal functions, and the psigmf is the product of two sigmoidal functions.
4. Polynomial based curve: pimf is the Pi curve which is named because of the shape. The pimf is zero on both extremes with a rise in the middle.

In order to choose the shape of membership function, all the candidates are tested with the same training data set and all the conditions except for shape are fixed. Through the test, the ANFIS with trapmf fails in training parameters, and sometimes the ANFIS with trimf and the gaussmf may fail in picking the belonging group (see section 5.3.1). As a result of the elimination of trapmf, trimf, and gaussmf, the remaining candidates should be compared. Table 5.2 shows the root mean square error

(RMSE) and training time of each ANFIS with different membership functions, and the results were from the original four groups. The ANFIS trained with gbellmf has the lowest RMSE, and the ANFIS trained with pimf takes the shortest time to train. Through running all the five ANFIS models, the results seem very similar to each others, so the criteria of comparison are the RMSE and training time. Since training the ANFIS with gbellmf takes only 20 minutes longer than the one with pimf, the gbellmf is determined as the shape of membership function.

Table 5.2 Comparison of different shapes of membership function

Shape	Group1 RMSE	Group2 RMSE	Group3 RMSE	Goup4 RMSE	Training time
gauss2mf	0.1457	0.0989	0.2525	0.2377	25hour 40min
gbellmf	0.1453	0.0970	0.2525	0.2369	23hour
psigmf	0.1454	0.0975	0.2526	0.2373	23hour7min
dsigmf	0.1457	0.0978	0.2525	0.2373	28hour55min
pimf	0.1457	0.0993	0.2525	0.2374	22hour41min

5.2.2 Number of Membership Function

There are two input features, chrominance a^* and chrominance b^* , for ANFIS in this research. Each input feature should be assigned a number of membership functions. Based on Table 5.3, although the set (8, 10) has lower RMSE than the set (4, 6), it takes about 5 times longer. Also, Figure 5.13 shows that the results shown no significant differences.

Table 5.3 Comparison of different number of membership functions

Number of membership functions (a^* , b^*)	Group1 RMSE	Training time (300 epochs)
(4, 6)	0.1457	47744sec=13 hr
(8, 10)	0.1451	228497sec=63hr
(16, 20)	Much lower?	more than three weeks

(a) Number of membership functions $(a^*, b^*)=(4,6)$

(b) Number of membership functions $(a^*, b^*)=(8,10)$

Figure 5.13 Outputs of ANFIS under different number of membership functions

Table 5.3 shows that the RMSE is still high, and one of the reasons may be that the number of membership functions are not enough for image recognition. As a result, the next step is to increase the number. Unfortunately, the number (16, 20) is too large for Matlab. Another reason may be the input-output mapping is not perfectly correct, since the mapping is automatically created and the FCM clustering result may have inconsistent results.

5.3 Automation of Rust Recognition by ANFIS

Through 5.1 and 5.2, three sub-ANFIS can be constructed by the three groups of training data. In order to automatically produce a threshold for segmentation the rust from a rust image, the model shown in Figure 5.14 is developed. The model is composed of three sub-ANFIS according to the mean b^* of background color, and the model ends with automatically creating two thresholds for mild rust and serious rust. This section first depicts how to choose the belonging sub-ANFIS system, and then describes the steps of determination of the two thresholds.

Figure 5.14 Framework of the proposed ANFIS model

5.3.1 Picking Sub-ANFIS

In training stage, all the input images are grouped into three groups as shown in Figure 5.15, and each group creates a sub-ANFIS system. After training, each input image is processed by automated detection of background and illumination adjustment to get the mean b^* of background color. According to the mean b^* value, the belonging sub-ANFIS system is chosen.

Sometimes an image may be designated to the wrong sub-ANFIS, since the mean b^* value locates around the border. In order to solve this problem, if the mean b^* value falls in the border of different groups, then both sub-ANFIS systems should be run. The border is defined as ± 2 units. In detail, if the mean b^* locates among -12 to

-8, then run both ANFIS1 and ANFIS2; if the mean b^* locates among -2 to 2, then run both ANFIS2 and ANFIS3. Finally, adopt the result of the one whose output > 0.8 colors are farther from rust color.

Figure 5.15 Relationship of groups and the border

5.3.2 Automated Determination of Two Thresholds

Each input rust image has a corresponding output map image that is a 256x256 matrix with values ranging from 0 to 1. 1s corresponds to background colors and 0s corresponds to rust colors. Figure 5.16 shows an output example of ANFIS with various threshold values. In this research, two threshold values (one for light or mild rust recognition, and the other for dark or serious rust recognition) will be automatically generated. The following points summarize how the two threshold values are determined:

1. Use the fundamental ellipse and the box area of the automated background detection to determine dark (or serious) rust spots and background, respectively. The pixel values of ANFIS output images range from 0 (i.e., rust) to 1 (i.e., background).

Figure 5.16 Output of a sub-ANFIS

2. To study the relationship between the threshold on pixel values and the recognition result, the change in recognition results is recorded for every 0.1 reduction on the threshold value (e.g., threshold change from $\text{output} \geq 0.9$ to $\text{output} \geq 0.8$ in Figure 5.16). With the reduction on the threshold value, there should be more and more information shown on the image. To get the best threshold values for background and rust clustering, it is important to know whether there would be any rust-color (or brown-color) pixels shown in the next 0.1 reduction on the threshold value. This process starts from $\text{threshold}=1$, followed by reducing the threshold by 0.1 each time. When $\text{threshold}=1$, there is least information shown on the image, i.e., only pixels with a value ≥ 1 (or $\text{output} \geq 1$) are shown. When $\text{threshold}=0$, the entire image is shown.
3. Detect the color information added when the threshold goes from one value (e.g., 0.5) to the next (e.g., 0.4). Compute the mean of the additional color information between two adjacent threshold values, from interval “ $\text{output} \geq 0.9$ – $\text{output} \geq 1$ ” to interval “ $\text{output} \geq 0$ – $\text{output} \geq 0.1$ ”. The means of additional color information for the input image in Figure 5.16 are shown as the red dots in Figure 5.17.

Figure 5.17 Output of every 10% additional colors

4. Selection of threshold for dark (or serious) rust spots (THs): Starting from interval “0-0.1” (i.e., “output $\geq$ 0–output $\geq$ 0.1”), search the first interval whose mean of additional color information is outside the fundamental ellipse (e.g., interval “0.1-0.2” in Figure 5.17). Divide the interval into four equal sub-intervals (e.g., “0.1-0.125,” “0.125-0.15,” “0.15-0.175,” and “0.175-0.2”). Compute the mean of additional color information for each sub-interval. Starting from the sub-interval with the minimum boundary values (e.g., “0.1-0.125”), search the sub-interval with the maximum boundary values that is still within the fundamental ellipse and set this sub-interval’s upper bound as the threshold THs (e.g., if the means of both “0.1-0.125” and “0.125-0.15” are within the fundamental ellipse, set 0.15 as the threshold THs).
5. Selection of threshold for light (or mild) rust spots (THm): Starting from interval “0.9-1” (i.e., “output $\geq$ 0.9–output $\geq$ 1”), search the first interval whose mean of additional color information is outside the box area of background colors (e.g., interval “0.7-0.8” in Figure 5.17). Divide the interval into four equal

sub-intervals (e.g., “0.7-0.725,” “0.725-0.75,” “0.75-0.775,” and “0.775-0.8”).

Compute the mean of additional color information for each sub-interval. Starting from the sub-interval with the minimum boundary values (e.g., “0.7-0.725”), search the sub-interval with maximum boundary values that is still outside the box area and set this sub-interval’s upper bound as the threshold TH_m (e.g., if the means of both “0.7-0.725” and “0.725-0.75” are outside the fundamental ellipse, set 0.75 as the threshold TH_m).

6. Also, as it is found that there might not be sufficient information shown when the threshold is greater than or equal to 0.8 (i.e., $\text{output} \geq 0.8$), THs and TH_m are capped at 0.8.

5.4 Stepwise BENFA model

The stepwise BENFA model is evolved from the image preprocessing of the adaptive ellipse approach (AEA) and adaptive network-based fuzzy inference system (ANFIS). The purpose of stepwise BENFA is to demonstrate the procedure of implementing BENFA. The model contains six steps, from image acquisition to defect recognition and calculation. Each step is described in detail below:

Step1: The first step is to train the ANFIS model. In this research, 120 256x256 rust images are used for training. These images are divided into three groups for the three sub-ANFIS systems according to the means of their background colors along the b^* axis on a^*b^* color plane. Illumination adjustment is considered in the training stage. The inputs for each image include chrominance a^* and chrominance b^* of the $L^*a^*b^*$ color space. The output image has pixel values ranging from 0 (i.e., rust) to 1 (i.e., background). Details on how pixel values are determined are elaborated in Sections 5.1.2.2. The generalized bell curve is adopted as the membership function for the ANFIS. Through some experiments, chrominance a^* has four bell curves in its

membership function and chrominance b^* has six bell curves in its membership function.

Figure 5.18 Stepwise BENFA model

Step2: The second step is image acquisition. Image data can be acquired by using a digital camera.

Step3: For a rust image, the box area of background colors on the a^*b^* plane is determined through automated background detection and illumination adjustment.

Step4: The mean of background colors along the b^* axis should be computed in order to select the right sub-ANFIS. Details about how to select the right sub-ANFIS and run it are shown in Section 5.3.1.

Step5: After Step4, the two threshold values, THs and THm, will be automatically determined (Section 5.3.2).

Step6: Once THs and THm are available, the recognition result with dark (or serious) and light (or mild) rust spots will be generated. The rust (or defect) percentage can be calculated by counting the percentage of rust pixels out of all the pixels in the image.

5.5 Applications of BENFA approach

Through training 120 256x256 images, the box-and-ellipse-based neuro-fuzzy approach (BENFA) model with three sub-ANFIS is ready for use. The training time is about 244 hours and running a 256x256 image takes about 230 seconds on average, using a personal computer with an AMD ATHLON64 3800+X2 2.0GHz CPU and 4096MB memory. The application of BENFA is shown in Figure 5.19. Compared to Figure 5.1, Image 1 in Figure 5.19 shows that BENFA is better than AEA, since BENFA can not only recognize mild rusts but also identify rust intensity. The results of Image 2 in Figure 5.19 show the ability of BENFA to overcome the non-uniform illumination problem. Furthermore, BENFA has the ability to recognize the rust intensity based on rust colors.

Figure 5.19 Application of BENFA

As the Fuzzy C-Means (FCM) algorithm is a very popular segmentation method in the image processing(Chi, Yan et al. 1996), it is selected to compare with the proposed BENFA in this paper. The number of clusters for FCM depends on the entropy of the image, $H(a^*, b^*)$. If the entropy is less than or equal to four, there are two clusters. If the entropy is greater than four, there are three clusters. After comparing BENFA to FCM using 40 rust images, the advantages of BENFA could be summarized as follows:

Figure 5.20 Comparison of BENFA and FCM in rust intensity recognition

1. BENFA is superior to FCM in terms of rust intensity recognition. It can be seen in Figure 5.20 that BENFA properly clustered the image into background, light (or mild) rust spots and dark (or serious) spots, while the clustering results of FCM are not right.
2. BENFA is more stable than FCM. In Figure 5.21, despite some background noises (the unevenness of the surface), BENFA can still generate good recognition results. On the contrary, the results produced by FCM are totally insensible and wrong.

Figure 5.21 Comparison of BENFA and FCM in stability

Table 5.4 shows the process times required for both methods.

Table 5.4 Processing times of BENFA and FCM for a 256x256 rust image on the a*b*

color plane		
Time (sec)	BENFA	FCM
average	228	47

Despite the advantages of BENFA, it still has limitation. As the pre-defined fundamental ellipse contains mostly brown and red colors, BENFA is unable to recognize rust images with brown or red background colors. However, this is not a big issue. For easy identification of rust spots, steel bridge coating would usually avoid brown/red colors.

5.6 Summary of Chapter

Since the AEA sometimes could not well recognize the rust area due to the color distribution, a new and effective bridge coating assessment (or rust recognition) approach, the box-and-ellipse-based neuro-fuzzy approach (BENFA), is proposed. BENFA is based on the a*b* color configuration and combines the illumination

adjustment, the fundamental ellipse of rust colors and the adaptive-network-based fuzzy inference system (ANFIS).

With the pre-defined fundamental ellipse, the box area of background colors of a rust image could be automatically determined. Once the “box” and “ellipse” are available, the background colors and dark (or serious) rust spots could be identified. ANFIS will then be used to process the remaining undefined colors to extract the light (or mild) rust spots. 120 rust images were used to train the ANFIS, which contains three sub-ANFISs.

To verify the effectiveness of BENFA, the popular Fuzzy C-Means (FCM) algorithm is chosen for comparison with BENFA. After processing 40 rust images with various colors of coating, it is found that BENFA is superior to FCM in rust intensity recognition. BENFA is also more stable than FCM in dealing with background noises (unevenness of surface). The only limitation of BENFA is that it is unable to recognize rust images with brown or red background colors due to the definition of the fundamental ellipse. However, it is not a big issue as brown/red colors usually would not be used for steel bridge coating.

CHAPTER 6 Illumination Adjustment for Bridge Coating Images Using BEMD-Morphology Approach (BMA)

Non-uniform illumination is always a challenge in the field of image processing. Shadows/shades and highlights would definitely enhance the difficulty of image recognition. Unfortunately, there is no robust algorithm that could solve the non-uniform illumination problem in rust defect recognition to date (Lee, Chang et al. 2005). Although color image processing mitigates the effect of inhomogeneous illumination, there is no perfect color space which can decompose illumination from chrominance completely. This chapter aims to simultaneously reduce the shades/shadows and highlights effect on a bridge coating image. The proposed method which is called BEMD-Morphology approach (BMA) combines the shade and shadow elimination by applying bidimensional empirical mode decomposition (BEMD) and highlight detection and replacement by morphological processing. The reconstruction results of the BMA will be evaluated by the K-Means algorithm which is the most popular image recognition method.

6.1 Shade and Shadow Elimination by Empirical Mode Decomposition

Sometimes shadows and shades exist on the bridge surface and degrade the bridge image. Since the non-uniform illumination problem troubles the image recognition in many fields, it is a popular issue but so far there is no perfect solution. Fortunately, new knowledge is keep growing. After Norden Huang proposed Hilbert-Huang transform (HHT) (Huang 2005), there are several applications in academic. The BEMD adopted in this section is based on Nune's work (see 2.5.2).

In this research, color image processing is the objective. To BEMD, its limitation is that BEMD can only decompose a grey image. If we want to decompose a color

image, the EMD process should be in five dimensions. In order to simplify the work, the result of a grayscale image using BEMD is applied to adjust a color image. The new image IM_adj is adjusted by the following steps:

1. IM is the shaded RGB image, and IM_{gray} is the grayscale of IM
2. $BEMD(IM_{gray}) = IMF_1 + IMF_2 + \dots + IMF_n + \text{residual}$
3. $ADJ = \text{Combinations of IMF and residual}$
4. $IM\_adj = IM / ADJ * \text{mean}(ADJ)$

The adjustment is based on the assumption that in RGB color space, a color will remain the same after multiplying or dividing the three components (R/G/B) by a constant (Cheng, Jiang et al. 2001). The first IMF contains major color variety information, and the residual mode contains the illumination trend. The following work is to determine how many IMF dominate the illumination component. Figure 6.1 shows the IMFs and residual of an image using the BEMD. Figure 6.1 (a) can be reconstructed through directly adding Figure 6.1 (b) to Figure 6.1 (d).

Figure 6.1 Application of BEMD on a grey image

Since the first work is to verify that the BEMD could eliminate the shades and shadows, artificial rust images are introduced for evaluation again. The artificial

images, as shown in Figure 6.2, are designed as size 100*100 in order to reduce the BEMD process time. There are 13 artificial rust images combined with three kinds of rust patterns and five coating colors. Each artificial image is combined with three different types of shades, and, therefore, there are a total of 39 non-uniformly illuminated artificial rust images for BEMD verification.

Figure 6.2 Artificial rust images for BEMD

All the images are grouped into two group, the first one contains 26 images are the testing group; the second one contains 13 images is for validation. Through BEMD and process the image with their residual and IMFs, the testing group shows that the accuracy is significantly high. The artificial images are low variety, so each image can only extract 2 to 3 IMF and one residual. Through the testing and validation group, the results in Table 6.1 show that the new image adjusted by the residual mode and all the IMF except the first mode could reduce the non-uniform illumination effect.

Table 6.1 Accuracy of in artificial images segmentation using K-Means

Artificial image	KM in RGB accuracy uniform illumination	KM in RGB accuracy shade 1	correct by residual+I MN except first mode	KM in RGB accuracy shade 2	correct by residual+I MN except first mode	KM in RGB accuracy shade 3	correct by residual+I MN except first mode
bu_blue	99.64	57.77	99.88	77.12	99.87	77.31	99.91
bu_khaki	97.27	60.66	87.47	65.62	73.82	67.98	74.92
bu_gray	98.84	64.18	97.18	74.14	97.36	66.23	97.27
bu_gray_dust	97.74	60.51	95.34	74.77	95.44	63	95.42
kha_blue	100	83.15	100	60.33	100	65.64	100
kha_green	99.58	73.52	99.85	40.21	99.85	61.85	99.85
kha_khaki	99.66	73.27	99.5	71.42	99.52	79.76	99.5
kha_gray	99.87	80.86	99.9	76.84	99.93	89.46	99.91
kha_gray_dust	99.73	82.26	99.76	70.1	99.78	71.82	99.82
wh_blue	100	53.33	100	80.86	100	0	100
wh_khaki	96.45	43.51	98.18	39.47	98.05	69.49	97.88
wh_gray	99.49	55.94	99.98	67.91	99.95	81.97	99.96
wh_gray_dust	98.77	45.9	99.73	65.03	99.77	43.85	99.74

Figure 6.3 shows the color artificial image adjustment by BEMD, and their clustering results using the K-Means algorithm in RGB color space. We can notice that the adjustment image is decomposed the inhomogeneous illumination, but the rust area is lighter. Although the accuracy of recognition is high, sometimes the background still exist inhomogeneous illumination. One of the possible reasons is that the code of BEMD in this research cannot extract more modes of IMF due to the stoppage criteria. If using smaller SD to be the stoppage criteria, there will be more modes of IMF but takes longer time. Since the process time is about one hour for a 100*100 image, we don't do further amendment. Note that we use Matlab to process the code.

Figure 6.3 Original and BEMD-adjusted artificial rust images segmentation using K-Means in RGB color space

Figure 6.4 shows the application on real rust images in RGB color space. The adjustment image does not depend on the illumination anymore. To process a 100*100 image by BEMD costs about 55 minutes.

Figure 6.4 BEMD adjustment in real color rust image

6.2 Highlight Substitution by Rust Color

Sometimes we use flash to lighten a rust image. Unfortunately, an uneven coating surface may cause highlight due to the flash. The highlight happens at the location of uneven surface. From observation, an uneven bridge surface is caused by rust. The highlight may indicate the location of rust. There are two kinds of highlight.

One is a protrusion whose neighbor is still background color, but there may be rust under the coating. The other one is a warp of coating which is caused by rust. Figure 6.5 shows the two kinds of highlight on a coating surface. Although the highlight area does not show by rust color, we know it is usually serious rust part. This section aims to substitute the highlight area by neighboring rust color to enhance the rust recognition accuracy. This section first introduces relevant literature, and then the proposed method will be described in detail.

Figure 6.5 Two kinds of highlight on bridge surface

6.2.1 Morphological Reconstruction for Brightness Detection

Ortiz proposed serial methods for highlights elimination using color morphology (Ortiz and Torres 2004; Ortiz and Torres 2005; Ortiz, Torres et al. 2005; Ortiz 2007). In summary, Ortiz proposed to use MS diagram for highlights detection, since the features of highlights are high intensity and low saturation. The M refers to the intensity, and S denotes the saturation. Eq. 6-1 to 6-2 show the definition of the MS diagram.

$$M = \frac{1}{3}(R + G + B) \quad (6-1)$$

$$S = \begin{cases} \frac{1}{2}(2R - G - B), & \text{if } (B + R) \geq 2G \\ \frac{1}{2}(R + G - 2B), & \text{if } (B + R) < 2G \end{cases} \quad (6-2)$$

where $R \in [0, 255]$, $G \in [0, 255]$, and $B \in [0, 255]$. In fact, the MS diagram is a positive projection of all the RGB corners of cubes to the M signal. Figure 6.6 shows the MS diagram.

Figure 6.6 RGB cube and its transformation in MS diagram (Ortiz 2007)

The author draw a conclusion that after contrast enhancement using TopHat transform, all the highlights are located on c3 and c4 lines of the MS diagram between $[M_{sp}, M_{max}]$ for M signal and $[0, S_{sp}]$ for S signal. The M_{sp} and S_{sp} are defined in equation 6-3 and 6-4 respectively.

$$S_{sp} = \frac{M_{max}}{10} \quad (6-3)$$

$$M_{sp} = \frac{2S_{max} - 3M_{max}}{-3} \quad (6-4)$$

where M_{max} and S_{max} are 255. After the detection the highlights, the morphological opening is utilized to reconstruct the color image by surrounding chromatic pixels.

6.2.2 Seed Points of Highlight

The methodology of highlight substitution by rust color in this paper is based on the Ortiz previous works. According to the conclusion of “all the highlights are located on c3 and c4 lines of the MS diagram between $[M_{sp}, M_{max}]$ for M signal and $[0, S_{sp}]$ for S signal (Ortiz 2007)”, a simplified detection is utilized in the thesis. Through calculation, the value of S_{sp} is 25.5 and M_{sp} is 85. According to Figure 6.7, we can find out that it is hard to specifically assign the highlights area on the MS diagram. In order to simplify the work, we suggest defining a box area in the right bottom corner according to the intersection of c3 and S_{sp} . Therefore, after contrast enhancement

using TopHat transform, each pixel which contains the value between [238, 255] for M signal and [0, 25.5] for S signal on MS diagram, as shown in Figure 6.7, is considered as a seed point of highlight. The contrast enhancement using TopHat transform is applied on the M signal as Equation 6-5, and Figure 6.8(c) shows the contrast enhancement of a grayscale image. Figure 6.8(d) to (e) shows the application on a real rust image.

Figure 6.7 Simplification of highlights area on MS diagram

$$M' = M + WTH(M) - BTH(M) \tag{6-5}$$

Figure 6.8 Contrast enhancement and seed points of highlight

6.2.3 Highlight Detection

Highlight has high M and low S, and its M is the highest. It is obvious that the M of rust is the lowest, and then the M of background is the second lowest. Therefore, it is reasonable to assume that the M of highlight should be higher than maximum value of background. After get the seed location of highlight by $M \geq 238$ and $S \leq 25.5$, automated detection of background is applied to get the maximum M value of background, denotes as M4highlight. In order to catch the highlight area, we suggest enlarging the seed area. If MS of the four neighbors of the seed location fall in $M > M4highlight$, then the neighbor pixel is considered as a highlight point (see Figure 6.9 (b)). Furthermore, morphological closing is utilizing to fill the whole of an area as shown in Figure 6.9(c), and Figure 6.9(d) shows the result of highlight detection.

Figure 6.9 Highlight detection by seed point enlargement

For discussion, if the seed points are not detected in advance, the results return bad as shown in Figure 6.10. Therefore, seed point detection is necessary in this research.

Figure 6.10 Highlight detection without seed points

6.2.4 Color Substitution

According to the result after closing (see Figure 6.9(c)), if the value of the pixel is 255, then collect all the third times elongated four neighbor's colors, as the shaded area in Figure 6.11, for every highlight area respectively. The color which is closest to the representative rust color from the collection is chosen as the substitutional color. If there are k highlight areas, then there will be k substitution colors. A representative rust color is the mean of pixel which falls in fundamental ellipse. Figure 6.12 shows the hilight substitution by rust color.

Figure 6.11 The third times elongated four neighbor relationships

Figure 6.12 Rust color substitution of highlight area

Furthermore, we want to mark the highlight area whose substitution is not serious rust color in green color as a suspicious area. This setting is based on the assumption that a highlight is produced by surface warping. Therefore, there are two possible situations of highlight, it is beside the broken area (a warp) or it is not broken yet (a protrusion). If it is beside the broken area, the substitutional color usually is serious rust color. If it is not broken yet, the substitutional color is usually not serious rust color, and it should be considered as a suspicious area. Figure 6.13 shows an example of marking suspicious area in green color.

Figure 6.13 Mark of suspicious area

6.2.5 Application of Highlight Substitution by Rust Color

This section summarizes the method of highlight substitution by rust color. It could be stepwise as follows:

1. Use top-hat to enhance the contrast of M , and then get the MS value.

2. Get the seed location of highlight followed by Ortiz, $M \geq 238$ and $S \leq 25.5$.
3. Apply automated detection of background to get the maximum M value of background, denoted as $M_{4\text{highlight}}$; Apply fundamental ellipse to determine a representative rust color.
4. Enlarge the seed area. If MS of the four neighbors of the seed location fall in $M > M_{4\text{highlight}}$, then the neighbor pixel is considered as a highlight point.
5. Close the whole of the enlargement, and then save as a map. The map is save as binary image where the seed of highlight is marked as 255.
6. According to the map, if the value of the pixel is 255, then collect all the third times elongated four neighbor's colors for every highlight area respectively. Choose the color which is closest to the representative rust color from the collection as the substitutional color. If there are k highlight areas, then there will be k substitution colors.
7. Recognize whether the substitutional color is serious rust color or not. If not, then mark in green color which is considered as a suspicious area.

Figure 6.14 shows the application for real rust images. Unfortunately, the suspicious area detection is only good for blue coating images, but bad for gray coating images. To gray coating images, almost every highlight is considered as a protrusion (mark in green) though it might be a warp. The main reason may be the rust colors of a gray coating are not defined in the fundamental ellipse. If the suspicious point detection is very important in the future, wide variety of rust colors of different coatings should be collected in order to define a new fundamental ellipse.

Figure 6.14 Highlight substitution by neighboring color and suspicious point detection

Figure 6.15 shows the comparison of highlight substitution with original image using the K-Means algorithm. From observation of the results of Figure 6.15(a), the warps which is substituted with serious rust colors provides a more precise and reasonable recognition. The rust percentage area of original image is higher than the

one with reconstruction due to the feature of the K-Means algorithm. The K-Means algorithm simply clusters the data into two groups, and the result depends on the data distribution. Although the rust percentage area of original image is higher, some serious rust parts which shows in highlight are not recognized as rust area.

Figure 6.15 Comparison of highlight substitution with original image using the K-Means in the a^*b^* color configuration

Figure 6.15(b) shows that the rust image contains both the warps and the protrusions. If we ignore the protrusions which are substitutes by background color, the result is shown in Fig. Figure 6.15(b). Furthermore, if the protrusions are necessarily considered as rust, it is also feasible in the proposed method.

6.3 Illumination Adjustment Strategy -BMA

This chapter aims to simultaneously reduce the shades/shadows and highlights effect on a bridge coating image. Section 6.1 introduces bidimensional empirical

mode decomposition method to adjust the shade and shadow of a color image. Section 6.2 proposes to paint the highlight area with rust color or neighboring color, since it may be a rust part without rust color. However, the shades/shadows and highlights probably exist on a bridge coating image in the same time in practice. Although the two sections are separately introduced in this research, it is reasonable and promising to combine the two methods for rust image reconstruction.

Figure 6.16 Combination of BEMD and highlight substitution

6.3.1 Development of the BMA

Although the BEMD method adjusts the illumination of an image from non-uniform to relatively uniform, the highlights area still exists due to its nearly white color. Therefore, we suggest first processing an image with BEMD method proposed in section 6.1 then following the highlight substitution by rust color proposed in section 6.2, and Figure 6.16 displays the results of application. The preliminary test shows that directly apply highlight detection and replacement after BEMD adjustment may not obtain good results, since the BEMD process may affect

the seeds detection for highlights. Therefore, the BEMD-Morphology Approach (BMA) is proposed to solve the situation. Generally speaking, the BMA catches the seeds of highlight before the BEMD adjustment, and then all the following steps are processed in a BEMD-adjusted image.

Through test in several rust images, we find out that seed of highlight should be detected before BEMD processing, and the seed enlargement and background detection should be implemented after BEMD processing. In order to automatically decide the size of seed enlargement (highlight area), the maximum M value of background color, $M_{4\text{highlight}}$, is set to be a threshold. According to the experimental results, a pixel had M value higher than $M_{4\text{highlight}} * 1.1$ is considered as a highlight point.

6.3.2 Stepwise the BMA

According to the test results in section 6.3.1, the BMA as shown in Figure 6.17 is developed for automated illumination adjustment. Figure 6.18 shows an example of application of the BMA. The proposed BMA is detailed as follows:

Step1: Image acquirement: Acquire a shaded or non-shaded RGB rust image.

Step2: BEMD adjustment: Color Image Adjustment by BEMD, and then apply the automated detection of background to get the maximum M value of background denoted as $M_{4\text{highlight}}$.

Step3: Seeds detection: Do the seeds detection for highlight on the original RGB image.

Figure 6.17 Flowchart of BMA

Step4: Highlights detection: Enlarge the seed points. If M value of the four neighbors of the seed location fall in $M > M_{4highlight} * 1.1$, then the neighbor pixel is considered as a highlight point. Close the whole of the enlargement, and then save as a map. The map is saved as binary image where the seed of highlight is marked as 255.

Figure 6.18 Application of BMA in uniformly and non-uniformly illuminated images

Step5: Highlights replacement: According to the map, if the value of the pixel is 255, then collect all the third times elongated four neighbor's colors for every highlight

area respectively. Choose the color which is closest to the representative rust color from the collection as the substitutional color. If there are k highlight areas, then there will be k substitution colors. Recognize whether the substitutional color is serious rust color or not. If not, then mark in green color which is considered as a suspicious area.

Since a non-uniform illuminated images may contain only highlights or both shades/shadows and highlights, the BMA should deal with the two kinds of images well. Figure 6.18 shows that the BMA performs well both in the shaded and non-shaded rust image. Figure 6.18, the example one and example two show that the BMA could eliminate the shades/shadows effect and make the highlight area to be recognized as serious rust; the example three shows that the BMA could also deal with a relatively uniformly illuminated image well.

6.3.3 Evaluation of the BMA using the K-Means

The BEMD-Morphology approach (BMA) aims to adjust a non-uniformly illuminated image. The BEMD reduces the non-uniformly illuminated effect, and the morphological processing replaces the highlight by neighboring color. In order to evaluate the BMA, the K-Means is used to compare the differences between adjusted images and the original images. The left part of Figure 6.19 shows that the BMA can reduce the shade effect and replace the highlight by rust color; the right part shows that the BMA also can replace the highlight by rust color well in a non-shaded image. The rust percentage is not important in this section. Since the K-Means just cluster the data into two groups, the result depends on the data distribution. The K-Means is adopted to do simple segmentation in order to show the improvement in the highlight recognition.

Figure 6.19 Evaluation of BMA in shaded and non-shaded rust image using the K-Means

6.4 Summary of Chapter

In order to mitigate the shades and shadows effect on a color rust image, BEMD is applied in this chapter. Although the BEMD can only process in grayscale image, the results are utilized to adjust a color image. Through processing 39 artificial rust images and some real rust images with artificial shade, we find out that the residual and all the IMF except the first mode could adjust the illumination of a color image to be relative uniform. After applying the adjustment using BEMD, the results show that the adjustment image is decomposed the inhomogeneous illumination, but the rust area will be lighter after adjustment. Although the accuracy of recognition is high, sometimes the background still exists inhomogeneous illumination. One of the possible reasons is that the code of BEMD in this research cannot extract more modes of IMF due to the stoppage criteria. If using smaller SD to be the stoppage criteria, there will be more modes of IMF but takes longer time. Since the process time is

about one hour for a 100*100 image, we don't do further amendment. Note that we use Matlab to process the code.

Highlights sometimes exist in a rust image due to an uneven surface. It is reasonable to assume that the highlights are rust areas. This chapter classifies two kinds of highlights, a protrusion which is surrounded by background color or a warp which neighbors a serious rust area. We proposed to apply morphological processing to catch the seeds of the highlight area, and then enlarge the seeds for highlight detection according to the maximum intensity value of background color. Finally the highlight area will be filled with the neighboring color which is closest to the rust color. The evaluations are assessed by the K-Means in order to show that the proposed method could better explain the highlight area.

The main contribution of this chapter is the BEMD-Morphology Approach (BMA). Most of the previous works individually focused on the mitigation of the shades/shadows effect or elimination of the highlights. The BMA aims to solve both of the problems in this one model. Since the BEMD may affect the seeds detection, the color adjustment using BEMD is after the seeds detection. After the color adjustment using BEMD, the illumination of the color image will be relatively uniform. Then the highlights detection and replacement will be the final step. The evaluation which is processed by the K-Means shows that the BMA could not only mitigate the light effect but also consider the highlight area as the rust area.

BMA mainly resolves the problem of both highlight and shade/shadow. It has to be acknowledged that the available rust images with highlight are few, and there are only two colors of coating. BMA can only provide a way of thinking, and more tests should be done in the future.

CHAPTER 7 CONCLUSIONS AND RECOMMENDATIONS

7.1 Conclusions of Research

With the prevalence and advance of computerized technologies, color image processing has been widely used instead of image processing in grayscale in academic. In the construction industry, the image processing in grayscale has been experimented on determination of rust percentages of steel bridge painting. This progress may lead to automation of steel bridge painting inspection. However, there are still some problems associated with this newly proposed application. Non-uniform illumination is one of the major causes. Also, the rust intensity could not be recognized in previous studies.

In order to resolve these problems, color image processing is adopted to enhance the recognition ability. Since a good color coordinate system decides the performance of color image processing, 14 color spaces are investigated in order to catch the best color configuration for rust image processing. Through processing 50 uniformly and non-uniformly illuminated rust images using the K-Means algorithm, it is found that the $L^*a^*b^*$ color space and the a^*b^* configuration (of the $L^*a^*b^*$ color space) perform comparatively better in rust image recognition due to their ability in filtering out light effects (or illumination factors). Therefore, the a^*b^* color configuration is adopted in this research.

Based on the observation with the naked eye that background colors are much easier to deal with than rust colors, which have a wider coverage of tones and shades of brown (or rust) colors (from dark rust colors to mild rust colors), the adaptive ellipse approach (AEA) is developed. The fundamental concept is to classify the rust image into three groups: background colors, rust colors, and non-defined colors (such as mild rust colors, etc.). The approach is designed for automatic rust recognition and

rust percentage calculation. In AEA, background colors and rust colors defined by the fundamental ellipse could be easily identified, but it takes some effort to process non-defined colors. To classify non-defined colors, the adaptive ellipse, which is enlarged from the fundamental ellipse, is used. Through experiments, it is proved that the proposed adaptive ellipse approach (AEA) has excellent ability for rust image recognition and could identify a wide range of rust colors. On the contrary, the performance of the popular K-Means algorithm is not as good as that of AEA due to the lack of ability to handle light/illumination factors. Despite all the advantages, AEA has one shortcoming, not being able to properly recognize rust images of brown or red paint colors. Since the segmentation is according to color similarity, a pixel with brown color will be recognized as a rust point. However, it would not be a big problem, as usually paints with colors distinct from rust colors will be chosen for steel bridge coating.

When the color distribution is nearly parallel to the major axis of the fundamental ellipse, the AEA cannot recognize the rust area very well. Therefore, the box and ellipse-based neural fuzzy approach (BENFA) is proposed to deal with the gradual color change from rust to background. In the image preprocessing step, background colors and rust colors are defined by the automated detection of background and fundamental ellipse respectively, and the remaining data are called non-defined color. BENFA is developed to deal with the non-defined color by means of utilizing neural fuzzy method. The output of BENFA is from zero to one which makes a gradual color change from rust color to background color. Two thresholds for serious rust and mild rust recognition are determined by means of the fundamental ellipse and the box of automated detection of background color. Observing the output from zero (rust) to one (background), once the mean of the additional color of output falls out of the fundamental ellipse or falls in the box of background color, the thresholds of serious rust and mild rust are determined.

Through processing forty rust images with different coatings, it is shown that the proposed BENFA has outstanding ability for rust image recognition and could identify

the rust intensity by color. On the opposite, the popular Fuzzy C-Means (FCM), which is only clustered data, has no ability to recognize the rust intensity. Also, the result of FCM is data distribution dependence while the BENFA shows the consistency in results. Since the recognition of BENFA is based on chrominance similarity, the limitation is not being able to recognize rust images of brown or red paint colors due to the definition of the fundamental ellipse which also contains most brown or red color.

The adaptive ellipse approach (AEA) and the box and ellipse-based neural fuzzy approach (BENFA) can eliminate the effect of non-uniform illumination. However, using the two methods on an image with non-uniform illumination and uniform illumination may have different results. An optimal model is expected to generate the same result for an image under different lighting/illumination conditions. It is a tough issue in image processing. The BEMD-Morphology Approach (BMA) provides another approach to eliminate the light effect. This model aims to achieve illumination adjustment for bridge coating images. The output of BMA (i.e., the third model) is a color-adjusted rust image, and BMA does not do rust recognition. Most of the previous works individually focused on the mitigation of the shades/shadows effect or elimination of the highlights. The BMA aims to solve both of the problems in this one model.

In order to mitigate the shades and shadows effect on a color rust image, bidimensional empirical mode decomposition (BEMD) is applied. Although the BEMD can only process in grayscale image, we utilize its result to adjust a color image. Through processing 39 artificial rust images and some real rust images with artificial shade, we find out that the residual and all the IMF except the first mode could adjust the illumination of a color image to be relative uniform. Highlights sometimes exist in a rust image due to an uneven surface. It is reasonable to assume that the highlights are rust areas. We proposed to apply morphological processing to catch the seeds of the

highlight area, and then enlarge the seeds for highlight detection according to the maximum intensity value of background color.

Since the BEMD may affect the seeds detection, the color adjustment using BEMD is after the seeds detection. After the color adjustment using BEMD, the illumination of the color image will be relatively uniform. Then the highlights detection and replacement will be the final step. The evaluation which is processed by the K-Means shows that the BMA could not only mitigate the light effect but also consider the highlight area as the rust area.

In summary, AEA and BENFA are the models of rust recognition, and BMA adjusts the color of rust image. The research shows that BENFA is better than AEA in rust recognition. In BMA, the highlight parts are being substituted for serious rust colors, and the shades or shadows are eliminated.

7.2 Research Contributions

The contributions of this research include:

- ◆ This research decides the a^*b^* color configuration through investigating 14 color spaces to be the best color coordinate system for rust recognition.
- ◆ This research utilizes automated detection of background color, illumination adjustment, and fundamental ellipse to form the adaptive ellipse approach (AEA). Through comparison with the K-Means clustering, the proposed AEA not only eliminates light effect but also recognizes mild-rust color well.
- ◆ In order to deal with the gradual change of color well, the box and ellipse-based neural fuzzy approach (BENFA) is proposed. Through comparison with the Fuzzy C-Means clustering, the proposed BENFA not only eliminates light effect but also consistently identifies the rust intensity.
- ◆ Directly filter the non-uniform illumination from a rust image is another approach to

resolve the illumination problem. The proposed BEMD-morphology approach (BMA) aims to adjust a bridge coating image. The BMA could simultaneously filter the shade/shadow and highlight from a color rust image. Furthermore, the highlight points will be replaced by the neighboring color to enhance the recognition accuracy.

7.3 Limitations

- ◆ Since the image segmentation in this research is based on color similarity, a pixel with brown and red color will be considered as a rust point. Consequently, one of the limitations is that the AEA and BENFA could not process the brown or red coating images.
- ◆ Since this research assumes that only rust and coating in a rust image, the two models could not recognize any other objects on the bridge surface (ex. ash or plant).
- ◆ The AEA and BENFA in this research were trained with rust images of steel bridge painting and can only be used to recognize rust images. Before applying the models, the acquisition of images should be classified into two groups, plain coating and coating with rust. The classification work could be achieved according to Lee's PhD dissertation (Lee 2005).

7.4 Recommendations for Future Work

1. From comparison of processing in grayscale image and color images in section 4.3.1, it is shown that color image processing is promising method dealing with non-uniform illumination problem.
2. In the future, the fist model, adaptive ellipse approach (AEA), is recommended that the mechanism of enlarging the fundamental ellipse could be modified. In the thesis, the enlargement percentages along the major and minor axes are the same. If they are different, the results may be improved.

3. Processing a uniformly illuminated image is a relatively easy task. However, most of the algorithms, even the proposed adaptive ellipse approach (AEA) and box-and-ellipse-based neuro-fuzzy approach (BENFA), could not catch the same result from dealing with the same image under different light intensities. The illumination problem dominates the recognition results. A robust method of filtering light component should be explored.

4. In BEMD-morphology approach (BMA), the BEMD may slightly change the color. Although the BEMD could mitigate the non-uniform illumination effect, an image dealt with BEMD still exists some non-uniform illumination. An advanced BEMD could be developed to extract more details.

5. Since the BEMD is a new method in the field of image processing, the processing time is quite long. In the future, a fast BEMD should be explored for real-time inspection use. Also, there is a mature method called wavelet transform which is also a wave analysis method. In the future, the wavelet method could be adopted or compared with the BEMD.

6. Since the proposed AEA and the BENFA could not process the brown or red coating images, it is recommended that the feasible color range should be quantified to clearly define the model limitation.

7. In order to make the model feasible in the practice, it is recommended that the image recognition results of a complete bridge should be compared with the professional engineers.

References

- AbdelRazig, Y. A. (1999). Construction Quality Assessment: A Hybrid Decision Support Model Using Image Processing and Neural Learning for Defect Recognition, Doctor of Philosophy, Purdue University.
- Adachi, Y., A. Imai, et al. (2000). "Extraction of face region by using characteristics of color space and detection of face direction through an eigenspace." The Fourth International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies, 2000.
- Bezdek, J. C., R. Ehrlich, et al. (1984). "FCM: The fuzzy C-Means clustering algorithm." Computers & Geosciences 10(2-3): 191-203.
- Bhagavatula, R. and M. Savvides (2007). "Analyzing Facial Images using Empirical Mode Decomposition for Illumination Artifact Removal and Improved Face Recognition. IEEE International Conference on Acoustics, Speech and Signal Processing, 2007.
- Carr, J. C., W. R. Fright, et al. (1997). "Surface interpolation with radial basis functions for medical imaging." Medical Imaging, IEEE Transactions on 16(1): 96-107.
- Chae, M. J. and D. M. Abraham (2001). "Neuro-Fuzzy Approaches for Sanitary Sewer Pipeline Condition Assessment." Journal of Computing in Civil Engineering 15(1): 4-14.
- Chang, J. L. and L. C. Chang, Eds. (2007). Neural Network. Taipei, 東華.
- Chen, P.-H. (2001). Digital Image Recognition Methods for Infrastructure Surface Coating Assessment, Doctor of Philosophy, Purdue University.
- Chen, P.-H. and L.-M. Chang (2003). "Artificial intelligence application to bridge painting assessment." Automation in Construction 12(4): 431-445.

- Chen, P.-H., Y.-C. Chang, et al. (2002). "Application of Multiresolution Pattern Classification to Steel Bridge Coating Assessment." *Journal of Computing in Civil Engineering* 16(4): 244-251.
- Cheng, H. D. and J.-R. Chen (1997). "Automatically determine the membership function based on the maximum entropy principle." *Information Sciences* 96(3-4): 163-182.
- Cheng, H. D., X. H. Jiang, et al. (2001). "Color image segmentation: advances and prospects." *Pattern Recognition* 34(12): 2259-2281.
- Chi, Z., H. Yan, et al., Eds. (1996). *Fuzzy Algorithms: With Applications to Image Processing and Pattern Recognition. Advances in Fuzzy Systems-Applicatons and Theory*, World Scientific.
- Coianiz, T., L. Torresani, et al. (2002). *2D Deformable Models for Visual Speech Analysis*. Italy.
- de Dios, J. J. and N. Garcia (2003). "Face detection based on a new color space YCgCr." *International Conference on Image Processing*, 2003.
- Federation, W. E. (1994). *Existing sewer evaluation and rehabilitation.*, WEF Manuals of Pract. FD-6: ASCE Manuals and Rep. on Engerg. Pract., Rep No. 62, Alexandria, Va.
- Gonzalez, R. C. and R. E. Woods, Eds. (2001). *Digital Image Processing*, Prentice Hall.
- Huang, N. E. (2005). *The Hilbert-Huang transform in engineering*. New York Taylor & Francis.
- Jang, J. S. R. (1993). "ANFIS: adaptive-network-based fuzzy inference system." *Systems, Man and Cybernetics, IEEE Transactions on* 23(3): 665-685.
- Kasperek, J. (2001). *Real Time Morphological Image Contrast Enhancement in Virtex FPGA. Field-Programmable Logic and Applications*: 430-440.

- Lee, S. (2005). Digital Color Image Processing System for Civil Infrastructure Health Assessment and Monitoring–Steel Bridge Coating Case, Doctor of Philosophy, Purdue University.
- Lee, S., L.-M. Chang, et al. (2005). "Performance Comparison of Bridge Coating Defect Recognition Methods." *Corrosion* 61: pp. 12-20
- Liew, A. W. C., L. Shu Hung, et al. (2003). "Segmentation of color lip images by spatial fuzzy clustering." *Fuzzy Systems, IEEE Transactions on* 11(4): 542-549.
- MathWorks, T. "Foundations of Fuzzy Logic."
- Menser, B. and F. Muller (1999). "Face detection in color images using principal components analysis." *Seventh International Conference on Image Processing and Its Applications*, 1999.
- Navulur, K., Ed. (2007). *Multispectral Image Analysis Using the Object-Oriented Paradigm*.
- Nunes, J. C., Y. Bouaoune, et al. (2003). "Image analysis by bidimensional empirical mode decomposition." *Image and Vision Computing* 21(12): 1019-1026.
- Nunes, J. C., S. Guyot, et al. (2005). "Texture analysis based on local analysis of the Bidimensional Empirical Mode Decomposition." *Machine Vision and Applications* 16(3): 177-188.
- Nunes, J. C., O. Niang, et al. (2003). "Texture analysis based on the bidimensional empirical mode decomposition with gray-level co-occurrence models." *Seventh International Symposium on Signal Processing and Its Applications*, 2003.
- Ohta, Y., T. Kanade, et al. (1980). "Color Information for Region Segmentation." *Computer Graphics and Image Processing* 13(3): 222 - 241.

- Ortiz, F. (2007). "Real-Time Elimination of Brightness in Color Images by MS Diagram and Mathematical Morphology." *Computer Analysis of Images and Patterns*: 458-465.
- Ortiz, F. and F. Torres (2004). "Vectorial morphological reconstruction for brightness elimination in colour images." *Real-Time Imaging* 10(6): 379-387.
- Ortiz, F. and F. Torres (2005). A New Inpainting Method for Highlights Elimination by Colour Morphology. *Pattern Recognition and Image Analysis*: 368-376.
- Ortiz, F., F. Torres, et al. (2002). "Colour Mathematical Morphology For Neural Image Analysis." *Real-Time Imaging* 8(6): 455-465.
- Ortiz, F., F. Torres, et al. (2005). "A Comparative Study of Highlights Detection and Elimination by Color Morphology and Polar Color Models." *Pattern Recognition and Image Analysis*: 295-302.
- Shih, P. and C. Liu (2005). "Comparative Assessment of Content-Based Face Image Retrieval in Different Color Spaces " *International Journal of Pattern Recognition and Artificial Intelligence* 19(7): 873-893.
- Shores, T. S. (2007). *Applied Linear Algebra and Matrix Analysis*. S. Axler and K. A. Ribet.
- Sivanandam, S. N., S. Sumathi, et al. (2007). *Introduction to Fuzzy Logic using MATLAB*.
- Stricker, M. A. and M. Orengo (1995). *Similarity of color images. Storage and Retrieval for Image and Video Databases III*, San Jose, CA, USA, SPIE.
- Takagi, T. and M. Sugeno (1983). "Derivation of Fuzz Control Rules from Human Operator's Control Actions." *IFCA Symp. Fuzzy Inform., Knowledge Representation and Decision Analysis*.
- The MathWorks (2008). "Image Processing Toolbox:Morphological

Reconstruction."

Vincent, L. (1993). "Morphological grayscale reconstruction in image analysis: applications and efficient algorithms." IEEE Transactions on Image Processing, 2(2): 176-201.

Xutao, Z., G. Yudong, et al. (2006). "Face recognition in color images using principal component analysis and fuzzy support vector machines." 1st International Symposium on Systems and Control in Aerospace and Astronautics, 2006.

Yang, J.-F., S.-S. Hao, et al. (2002). "Color image segmentation using fuzzy C-means and eigenspace projections." Signal Processing 82(3): 461-472.

Zhang, D., S. Chen, et al. (2005). Representing Image Matrices: Eigenimages vs. Eigenvectors. Nanjing, Nanjing University.