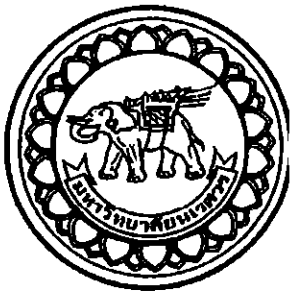


การปรับปรุงตัวอักษรที่ได้จากการสแกน

Scanned Text Enhancement

นายวันชัย กิสิน รหัส 46360103

นายประวิทย์ วงศ์จักรคำ รหัส 46361960

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 25 พ.ค. 2553
เลขทะเบียน 150091571
เลขเรียกหนังสือ ปร. จ426ก 2549
มหาวิทยาลัยนเรศวร

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2549

ใบรับรองโครงงานวิศวกรรม

หัวข้อโครงงาน	การปรับปรุงตัวอักษรที่ได้จากการสแกน		
ผู้ดำเนินโครงงาน	นายวันชัย	กิสิน	รหัส 46360103
	นายประวิทย์	วงศ์จักรคำ	รหัส 46361960
อาจารย์ที่ปรึกษา	ดร.พนมขวัญ	ริยะมงคล	
สาขาวิชา	วิศวกรรมคอมพิวเตอร์		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2549		

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์

คณะกรรมการสอบโครงงานวิศวกรรม

..... ประธานกรรมการ

(ดร.พนมขวัญ ริยะมงคล)

..... กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.สุชาติ แย้มเม่น)

..... กรรมการ

(อาจารย์ศิริพร เดชะศิลารักษ์)

หัวข้อโครงงาน	การปรับปรุงตัวอักษรที่ได้จากการสแกน		
ผู้ดำเนินโครงงาน	นายวันชัย	กิสิน	รหัส 46360103
	นายประวิทย์	วงศ์จักรคำ	รหัส 46361960
อาจารย์ที่ปรึกษา	ดร.พนมขวัญ	ริยะมงคล	
สาขาวิชา	วิศวกรรมคอมพิวเตอร์		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2549		

บทคัดย่อ

ปัจจุบันการจัดเก็บเอกสารด้วยวิธีการสแกนได้รับความนิยมมากขึ้น แต่ตัวอักษรที่ได้จากการสแกนนั้นอาจจะยังมีความไม่สมบูรณ์ ทั้งนี้อาจเกิดมาจากเอกสารต้นฉบับเอง หรือเกิดปัญหาจากกระบวนการสแกนเอกสารเอง จึงทำให้ตัวอักษรที่ได้เลือนราง หรืออาจมีความเข้มมากเกินไป ทำให้อ่านได้ยาก จึงจำเป็นต้องมีการปรับปรุงเพื่อทำให้ตัวอักษรสามารถอ่านได้ง่ายยิ่งขึ้น โครงงานนี้ได้นำเอาวิธีการ Dilation, Erosion, Opening และ Closing มาใช้ในการปรับปรุงตัวอักษรที่ได้จากการสแกน โดยใช้โปรแกรม MATLAB ผลลัพธ์ที่ได้เมื่อใช้ Closing แล้วตามด้วย Dilation กับตัวอักษรบางหรือเลือนราง จะทำให้ตัวอักษรหนาขึ้นและอ่านได้ง่ายขึ้น เมื่อใช้ Opening แล้วตามด้วย Erosion กับตัวอักษรหนาหรือติดกัน จะทำให้ตัวอักษรบางลงและอ่านได้ง่ายขึ้น

Project Title	Scanned Text Enhancement		
Name	Mr.Wanchai	Kisin	ID 46360103
	Mr.Prawit	Wongjakkum	ID 46361960
Project Advisor	Dr.Panomkhawn	Riyamongkol	
Major	Computer Engineering		
Department	Electrical and Computer Engineering		
Academic Year	2006		

.....

Abstract

The results form scanning text may be separated, not clear, or too thick characters. It is difficult to read. So, characters should be improved. This project used Dilation, Erosion, Opening, and closing to enhance text. MATLAB is a tool to develop scanned text enhancement program. From the project results, Thin or blur characters can be enhanced by using Closing followed by Dilation. Moreover, thick or too close characters can be enhanced by using Opening followed by Erosion.

กิตติกรรมประกาศ

โครงงานวิศวกรรมคอมพิวเตอร์นี้สำเร็จได้ด้วยดีก็เนื่องจากความอนุเคราะห์จากท่านอาจารย์ที่ปรึกษาโครงงาน คือ อาจารย์พนมขวัญ ริยะมงคล ที่ให้ความกรุณาแนะนำวิธีในการทำงานให้เข้าใจถึงการศึกษาอย่างเป็นระบบขั้นตอน อีกทั้งสละเวลาเพื่อตรวจสอบการทำงานและชี้แนวทางแก้ไขในทุกขั้นตอนตลอดการทำโครงงาน และขอบคุณเพื่อนๆ ที่ไม่ได้เอ่ยนามที่คอยให้คำแนะนำต่างๆ

วันชัย กิสิน

ประวิทย์ วงศ์จักรคำ

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ก

บทคัดย่อภาษาอังกฤษ ..... ข

กิตติกรรมประกาศ ..... ค

สารบัญ ..... ง

สารบัญตาราง ..... ฉ

สารบัญรูป ..... ช

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและความสำคัญของปัญหา 1

1.2 วัตถุประสงค์ 1

1.3 เป้าหมายและขอบเขตของโครงงาน 1

1.4 ขั้นตอนการดำเนินงาน 2

1.5 แผนการดำเนินงาน 2

1.6 ผลที่คาดว่าจะได้รับ 3

1.7 รายละเอียดงบประมาณของโครงงาน 3

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง 4

2.1 ภาพดิจิตอล (Digital Image) 4

2.2 มาตรฐานของสี 10

2.3 รูปแบบของไฟล์ภาพ (Digital file format) 17

2.4 รูปแบบของภาพ 21

2.5 การทำงานของ Scanner 22

2.6 กระบวนการวัดทางสถิติ (Statistical Operations) 23

2.7 การประมวลผลภาพกับรูปร่างและโครงร่างของภาพ (Morphological Image Processing) 26

2.8 การกรองข้อมูลภาพโดยใช้ค่ามัธยฐาน (Median filtering) 34

สารบัญ (ต่อ)

หน้า

บทที่ 3 วิธีดำเนินงาน 35

3.1 บล๊อคไดอะแกรม 35

3.2 การเตรียมข้อมูลภาพสำหรับการปรับปรุงตัวอักษรที่ได้จากการสแกน 35

3.3 วิเคราะห์ลักษณะตัวอักษรในภาพ 39

3.4 ออกแบบ Structure element โดยวิเคราะห์จากลักษณะตัวอักษร 41

3.5 ออกแบบการทดลอง 42

บทที่ 4 ผลการทดลองและวิเคราะห์ผลการทดลอง 43

4.1 ผลการทดลองสำหรับภาพที่มีลักษณะตัวอักษรบางหรือเลือนรางแบบไม่มี noise 43

4.2 ผลการทดลองสำหรับภาพที่มีลักษณะตัวอักษรบางหรือเลือนรางแบบมี noise 62

4.3 ผลการทดลองสำหรับภาพที่มีลักษณะตัวอักษรหนาหรือติดกันแบบไม่มี noise 81

4.4 ผลการทดลองสำหรับภาพที่มีลักษณะตัวอักษรหนาหรือติดกันแบบมี noise 90

4.5 วิเคราะห์ผลการทดลอง 99

บทที่ 5 สรุปผล 102

5.1 สรุปผลการทดลอง 102

5.2 ปัญหาและอุปสรรค 103

5.3 ข้อเสนอแนะ 103

เอกสารอ้างอิง 104

ภาคผนวก 105

ภาคผนวก ก 106

ภาคผนวก ข 111

ประวัติผู้เขียนโครงงาน 129

สารบัญตาราง

ตารางที่ หน้า

1.1 แผนการดำเนินงาน 2

2.1 ตารางแสดงค่าในภาพระดับเทา 25

2.2 ตารางแสดงค่าในภาพขาวดำหลังการทำ Thresholding 25

2.3 ตารางแสดงค่าในภาพระดับเทา 34

2.4 ตารางแสดงค่าในภาพระดับเทาที่ผ่าน median filter ขนาด 3x3 34

สารบัญรูป

รูปที่ หน้า

2.1 การเรียงกันเป็นวงกลมของสี 10 ระดับ 4
2.2 ระบบสี ที่สร้างจาก Munsell system 5
2.3 พิกเซลของภาพ 5
2.4 พิกเซลสามารถที่จะมองเห็นแต่ละพิกเซลได้โดยการขยายภาพที่เป็นดิจิตอล 6
2.5 ภาพ เอกสาร 8" x 10" 8
2.6 Dynamic range 10
2.7 ระบบสี RGB 11
2.8 การแสดงสีระบบ Additive 12
2.9 การผสมกันของแม่สีทั้งสาม 12
2.10 การแสดงสีแบบ Subtractive 12
2.11 แสดงระบบสี HSV 13
2.12 แสดงระบบสี HSL 15
2.13 ระบบสี CMY 15
2.14 ระบบสี CMYK 16
2.15 คุณภาพของภาพตามจำนวนบิต 21
2.16 ภาพที่ถูกสร้างแบบบิตแมป 21
2.17 ภาพที่ถูกสร้างด้วย เวกเตอร์ 22
2.18 สแกนเนอร์ 23
2.19 แสดงการไล่สีระดับเทา (Gray scale) 24
2.20 แสดงจำนวน 4-connected และ 8-connected 26
2.21 ตัวอย่างสำหรับตัวดำเนินการแบบขยาย (Dilation) 27
2.22 แสดงการดำเนินการแบบขยาย(Dilation) 27
2.23 ผลลัพธ์หลังจากกระบวนการเสร็จสิ้น 29
2.24 ตัวอย่างสำหรับตัวดำเนินการแบบหด (Erosion) 29
2.25 แสดงการดำเนินการแบบหด (Erosion) 30
2.26 ผลลัพธ์หลังจากกระบวนการเสร็จสิ้น 31
2.27 ผลลัพธ์ของการทำ (ค) Erosion และ (ง) Dilation 31
2.28 ตัวอย่างสำหรับตัวดำเนินการแบบเปิด (Opening) 32

สารบัญรูป(ต่อ)

รูปที่ หน้า

2.29 ตัวอย่างสำหรับตัวดำเนินการแบบปิด (Closing) 33
2.30 ผลลัพธ์ของการทำ (ค)Opening และ (ง)Closing 34
3.1 บล็อคไดอะแกรม 35
3.2 แผนผังโปรแกรมของการหาค่า Threshold (T) 37
3.3 แผนผังโปรแกรมของการทำ Thresholding 38
3.4 ลักษณะของตัวอักษรมีลักษณะบางหรือเลือนรางแบบไม่มี noise 39
3.5 ลักษณะของตัวอักษรมีลักษณะบางหรือเลือนรางแบบมี noise 40
3.6 ลักษณะของตัวอักษรมีลักษณะหนาหรือติดกันแบบไม่มี noise 40
3.7 ลักษณะของตัวอักษรมีลักษณะหนาหรือติดกันแบบมี noise 41
3.8 Structure Element สำหรับตัวอักษรที่มีลักษณะบางหรือเลือนราง 41
3.9 Structure Element สำหรับตัวอักษรที่มีลักษณะหนาหรือติดกัน 42
4.1 ภาพระดับเทาต้นฉบับของตัวอักษรบางหรือเลือนรางแบบไม่มี Noise 43
4.2 ภาพขาวดำของตัวอักษรบางหรือเลือนรางแบบไม่มี Noise 43
4.3 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบไม่มี Noise เทียบกับ ผลลัพธ์จากการใช้ Dilation ด้วย Structure element แบบที่ 1 จากรูปที่ 3.8 44
4.4 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบไม่มี Noise เทียบกับ ผลลัพธ์จากการใช้ Dilation ด้วย Structure element แบบที่ 2 จากรูปที่ 3.8 45
4.5 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบไม่มี Noise เทียบกับ ผลลัพธ์จากการใช้ Dilation ด้วย Structure element แบบที่ 3 จากรูปที่ 3.8 46
4.6 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบไม่มี Noise เทียบกับ ผลลัพธ์จากการใช้ Dilation ด้วย Structure element แบบที่ 4 จากรูปที่ 3.8 47
4.7 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบไม่มี Noise เทียบกับ ผลลัพธ์จากการใช้ Dilation ด้วย Structure element แบบที่ 5 จากรูปที่ 3.8 48
4.8 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบไม่มี Noise เทียบกับ ผลลัพธ์จากการใช้ Dilation ด้วย Structure element แบบที่ 6 จากรูปที่ 3.8 49
4.9 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบไม่มี Noise เทียบกับ ผลลัพธ์จากการใช้ Closing ด้วย Structure element แบบที่ 1 จากรูปที่ 3.8 50

สารบัญรูป(ต่อ)

รูปที่ หน้า

4.10 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบไม่มี Noise เทียบกับ
ผลลัพธ์จากการใช้ Closing ด้วย Structure element แบบที่ 2 จากรูปที่ 3.8 51

4.11 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบไม่มี Noise เทียบกับ
ผลลัพธ์จากการใช้ Closing ด้วย Structure element แบบที่ 3 จากรูปที่ 3.8 52

4.12 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบไม่มี Noise เทียบกับ
ผลลัพธ์จากการใช้ Closing ด้วย Structure element แบบที่ 4 จากรูปที่ 3.8 53

4.13 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบไม่มี Noise เทียบกับ
ผลลัพธ์จากการใช้ Closing ด้วย Structure element แบบที่ 5 จากรูปที่ 3.8 54

4.14 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบไม่มี Noise เทียบกับ
ผลลัพธ์จากการใช้ Closing ด้วย Structure element แบบที่ 6 จากรูปที่ 3.8 55

4.15 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบไม่มี Noise เทียบกับ
ผลลัพธ์จากการใช้ Closing + Dilation ด้วย Structure element แบบที่ 1 จากรูปที่ 3.8..... 56

4.16 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบไม่มี Noise เทียบกับ
ผลลัพธ์จากการใช้ Closing + Dilation ด้วย Structure element แบบที่ 2 จากรูปที่ 3.8..... 57

4.17 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบไม่มี Noise เทียบกับ
ผลลัพธ์จากการใช้ Closing + Dilation ด้วย Structure element แบบที่ 3 จากรูปที่ 3.8 58

4.18 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบไม่มี Noise เทียบกับ
ผลลัพธ์จากการใช้ Closing + Dilation ด้วย Structure element แบบที่ 4 จากรูปที่ 3.8 59

4.19 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบไม่มี Noise เทียบกับ
ผลลัพธ์จากการใช้ Closing + Dilation ด้วย Structure element แบบที่ 5 จากรูปที่ 3.8 60

4.20 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบไม่มี Noise เทียบกับ
ผลลัพธ์จากการใช้ Closing + Dilation ด้วย Structure element แบบที่ 6 จากรูปที่ 3.8..... 61

4.21 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบมี Noise 62

4.22 ผลลัพธ์การลบ Noise โดยใช้ Median filter 62

4.23 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบมี Noise เทียบกับ
ผลลัพธ์จากการใช้ Dilation ด้วย Structure element แบบที่ 1 จากรูปที่ 3.8..... 63

4.24 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบมี Noise เทียบกับ
ผลลัพธ์จากการใช้ Dilation ด้วย Structure element แบบที่ 2 จากรูปที่ 3.8..... 64

สารบัญรูป(ต่อ)

รูปที่ หน้า

4.25 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบมี Noise เทียบกับ
ผลลัพธ์จากการใช้ Dilation ด้วย Structure element แบบที่ 3 จากรูปที่ 3.8 65
4.26 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบมี Noise เทียบกับ
ผลลัพธ์จากการใช้ Dilation ด้วย Structure element แบบที่ 4 จากรูปที่ 3.8 66
4.27 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบมี Noise เทียบกับ
ผลลัพธ์จากการใช้ Dilation ด้วย Structure element แบบที่ 5 จากรูปที่ 3.8 67
4.28 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบมี Noise เทียบกับ
ผลลัพธ์จากการใช้ Dilation ด้วย Structure element แบบที่ 6 จากรูปที่ 3.8 68
4.29 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบมี Noise เทียบกับ
ผลลัพธ์จากการใช้ Closing ด้วย Structure element แบบที่ 1 จากรูปที่ 3.8 69
4.30 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบมี Noise เทียบกับ
ผลลัพธ์จากการใช้ Closing ด้วย Structure element แบบที่ 2 จากรูปที่ 3.8 70
4.31 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบมี Noise เทียบกับ
ผลลัพธ์จากการใช้ Closing ด้วย Structure element แบบที่ 3 จากรูปที่ 3.8 71
4.32 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบมี Noise เทียบกับ
ผลลัพธ์จากการใช้ Closing ด้วย Structure element แบบที่ 4 จากรูปที่ 3.8 72
4.33 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบมี Noise เทียบกับ
ผลลัพธ์จากการใช้ Closing ด้วย Structure element แบบที่ 5 จากรูปที่ 3.8 73
4.34 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบมี Noise เทียบกับ
ผลลัพธ์จากการใช้ Closing ด้วย Structure element แบบที่ 6 จากรูปที่ 3.8 74
4.35 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบมี Noise เทียบกับ
ผลลัพธ์จากการใช้ Closing + Dilation ด้วย Structure element แบบที่ 1 จากรูปที่ 3.8 75
4.36 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบมี Noise เทียบกับ
ผลลัพธ์จากการใช้ Closing + Dilation ด้วย Structure element แบบที่ 2 จากรูปที่ 3.8 76
4.37 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบมี Noise เทียบกับ
ผลลัพธ์จากการใช้ Closing + Dilation ด้วย Structure element แบบที่ 3 จากรูปที่ 3.8 77
4.38 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบมี Noise เทียบกับ
ผลลัพธ์จากการใช้ Closing + Dilation ด้วย Structure element แบบที่ 4 จากรูปที่ 3.8 78

สารบัญรูป(ต่อ)

รูปที่ หน้า

4.39 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบมี Noise เทียบกับ
ผลลัพธ์จากการใช้ Closing + Dilation ด้วย Structure element แบบที่ 5 จากรูปที่ 3.8 79

4.40 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบมี Noise เทียบกับ
ผลลัพธ์จากการใช้ Closing + Dilation ด้วย Structure element แบบที่ 6 จากรูปที่ 3.8 80

4.41 ภาพต้นฉบับตัวของอักษรหนาหรือติดกันแบบไม่มี Noise 81

4.42 ภาพต้นฉบับของตัวอักษรหนาหรือติดกันแบบไม่มี Noise เทียบกับ
ผลลัพธ์จากการใช้ Erosion ด้วย Structure element แบบที่ 1 จากรูปที่ 3.9.. 82

4.43 ภาพต้นฉบับของตัวอักษรหนาหรือติดกันแบบไม่มี Noise เทียบกับ
ผลลัพธ์จากการใช้ Erosion ด้วย Structure element แบบที่ 2 จากรูปที่ 3.9 83

4.44 ภาพต้นฉบับของตัวอักษรหนาหรือติดกันแบบไม่มี Noise เทียบกับ
ผลลัพธ์จากการใช้ Erosion ด้วย Structure element แบบที่ 3 จากรูปที่ 3.9 84

4.45 ภาพต้นฉบับของตัวอักษรหนาหรือติดกันแบบไม่มี Noise เทียบกับ
ผลลัพธ์จากการใช้ Erosion ด้วย Structure element แบบที่ 4 จากรูปที่ 3.9 85

4.46 ภาพต้นฉบับของตัวอักษรหนาหรือติดกันแบบไม่มี Noise เทียบกับ
ผลลัพธ์จากการใช้ Opening + Erosion ด้วย Structure element แบบที่ 1 จากรูปที่ 3.9 86

4.47 ภาพต้นฉบับของตัวอักษรหนาหรือติดกันแบบไม่มี Noise เทียบกับ
ผลลัพธ์จากการใช้ Opening + Erosion ด้วย Structure element แบบที่ 2 จากรูปที่ 3.9 87

4.48 ภาพต้นฉบับของตัวอักษรหนาหรือติดกันแบบไม่มี Noise เทียบกับ
ผลลัพธ์จากการใช้ Opening + Erosion ด้วย Structure element แบบที่ 3 จากรูปที่ 3.9 88

4.49 ภาพต้นฉบับของตัวอักษรหนาหรือติดกันแบบไม่มี Noise เทียบกับ
ผลลัพธ์จากการใช้ Opening + Erosion ด้วย Structure element แบบที่ 4 จากรูปที่ 3.9 89

4.50 ภาพต้นฉบับของตัวอักษรหนาหรือติดกันแบบมี Noise 90

4.51 ผลลัพธ์การลบ Noise โดยใช้ Median filter 90

4.52 ภาพต้นฉบับของตัวอักษรหนาหรือติดกันแบบมี Noise เทียบกับ
ผลลัพธ์จากการใช้ Erosion ด้วย Structure element แบบที่ 1 จากรูปที่ 3.9 91

4.53 ภาพต้นฉบับของตัวอักษรหนาหรือติดกันแบบมี Noise เทียบกับ
ผลลัพธ์จากการใช้ Erosion ด้วย Structure element แบบที่ 2 จากรูปที่ 3.9. 92

สารบัญรูป(ต่อ)

รูปที่ หน้า

4.54 ภาพต้นฉบับของตัวอักษรหนาหรือติดกันแบบมี Noise เทียบกับ
ผลลัพธ์จากการใช้ Erosion ด้วย Structure element แบบที่ 3 จากรูปที่ 3.9.....93
4.55 ภาพต้นฉบับของตัวอักษรหนาหรือติดกันแบบมี Noise เทียบกับ
ผลลัพธ์จากการใช้ Erosion ด้วย Structure element แบบที่ 4 จากรูปที่ 3.9.....94
4.56 ภาพต้นฉบับของตัวอักษรหนาหรือติดกันแบบมี Noise เทียบกับ
ผลลัพธ์จากการใช้ Opening + Erosion ด้วย Structure element แบบที่ 1 จากรูปที่ 3.9 95
4.57 ภาพต้นฉบับของตัวอักษรหนาหรือติดกันแบบมี Noise เทียบกับ
ผลลัพธ์จากการใช้ Opening + Erosion ด้วย Structure element แบบที่ 2 จากรูปที่ 3.9 96
4.58 ภาพต้นฉบับของตัวอักษรหนาหรือติดกันแบบมี Noise เทียบกับ
ผลลัพธ์จากการใช้ Opening + Erosion ด้วย Structure element แบบที่ 3 จากรูปที่ 3.9 97
4.59 ภาพต้นฉบับของตัวอักษรหนาหรือติดกันแบบมี Noise เทียบกับ
ผลลัพธ์จากการใช้ Opening + Erosion ด้วย Structure element แบบที่ 4 จากรูปที่ 3.9 98

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันมีเทคโนโลยีทางด้านเอกสารอิเล็กทรอนิกส์ได้เข้ามามีบทบาทในสังคมมนุษย์มากขึ้น การสแกนเอกสารเป็นวิธีหนึ่งที่ทำให้งานเกี่ยวกับเอกสารเป็นไปอย่างรวดเร็ว เนื่องจากเราไม่จำเป็น ต้องพิมพ์เอกสารนั้นใหม่ทั้งชุด หลังจากการสแกนเอกสารมาเพื่อทำเป็นเอกสารอิเล็กทรอนิกส์จะได้ผลลัพธ์จากการสแกนออกมาอยู่ในรูปแบบของไฟล์รูปภาพ แต่ตัวอักษรที่ได้จากมาจากการสแกนนั้นอาจจะยังไม่สมบูรณ์ ทั้งนี้อาจเกิดมาจากเอกสารต้นฉบับเอง หรือเกิดจากปัญหาของกระบวนการสแกนเอกสารเอง จึงทำให้ตัวอักษรหรือข้อความที่ได้มีความเลือนราง ตัวอักษรบางตัวอาจจะขาดความสมบูรณ์ หรืออาจมีความเข้มมากเกินไป ทำให้ตัวอักษรหรือข้อความเหล่านั้นอ่านได้ยาก

ทางผู้ทำโครงงานจึงได้มีแนวคิดที่จะทำการปรับปรุงตัวอักษรหรือข้อความในไฟล์รูป ภาพที่ได้จากการสแกนให้ดีขึ้นเพื่อให้ง่ายและสะดวกสบายต่อการอ่านและนำไปใช้งานมากขึ้น โดยการใช้อัลกอริทึมต่างๆ มาช่วยในการปรับปรุงตัวอักษรเหล่านี้

1.2 วัตถุประสงค์

1.2.1 เพื่อปรับปรุงตัวอักษรหรือข้อความจากไฟล์ภาพสี ภาพระดับเทา หรือภาพขาวดำที่ได้จากการสแกนเอกสารให้สมบูรณ์และสามารถอ่านได้ง่ายขึ้น

1.2.2 เพื่อศึกษาและเปรียบเทียบอัลกอริทึมที่ใช้ในการจัดการกับตัวอักษรหรือข้อความในไฟล์รูปภาพที่ได้จากการสแกน

1.2.3 เพื่อหาอัลกอริทึมที่เหมาะสมในการจัดการกับปัญหาแต่ละปัญหาเกี่ยวกับข้อความในไฟล์รูปภาพ เช่น ตัวอักษรขาดความสมบูรณ์ ตัวอักษรมีความเลือนราง เป็นต้น

1.2.4 เพื่อพัฒนาเทคนิคการปรับปรุงตัวอักษร

1.3 เป้าหมายและขอบเขตของโครงงาน

1.3.1 พัฒนาการปรับปรุงตัวอักษรที่อยู่ในไฟล์ภาพสี ระดับเทา และขาวดำ ที่มีสีขาวเป็นพื้นหลัง ด้วยอัลกอริทึมแบบต่างๆ โดยได้ผลลัพธ์อยู่ในรูปภาพขาว-ดำ

1.3.2 ใช้โปรแกรม MATLAB เป็นเครื่องมือช่วยในการปรับปรุงตัวอักษรจากไฟล์ภาพ

1.3.3 ข้อความหรือตัวอักษรในไฟล์รูปภาพที่ได้จากการสแกนมีความสมบูรณ์และสามารถอ่านได้ง่ายขึ้น

1.4 ขั้นตอนการดำเนินงาน

1.4.1 วางแผนการทำงานและกำหนดระยะเวลาของกิจกรรมต่างๆ

1.4.2 ค้นคว้าและศึกษาเทคนิคและทฤษฎีที่จำเป็นสำหรับการปรับปรุงตัวอักษรที่ได้จากการสแกน

1.4.3 นำความรู้ที่ได้จากการศึกษามาประยุกต์ใช้ในการปรับปรุงตัวอักษร

1.4.4 เขียนโปรแกรมโดยใช้ MATLAB

1.4.5 เปรียบเทียบอัลกอริทึมว่าอัลกอริทึมใดทำให้ตัวอักษรสมบูรณ์ที่สุด

1.4.6 สรุปผลที่ได้จากการศึกษาและแนวทางการพัฒนาที่ได้จากการศึกษา

1.4.7 จัดทำรูปเล่มโครงงาน

1.5 แผนการดำเนินงาน

ตารางที่ 1.1 แผนการดำเนินงาน

กิจกรรม	ปี 2548		ปี 2549									
	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.
วางแผนการทำงานและกำหนดระยะเวลาของกิจกรรมต่างๆ	←	→										
ค้นคว้าและศึกษาเทคนิคและทฤษฎีที่จำเป็นสำหรับการปรับปรุงตัวอักษรที่ได้จากการสแกน		←	→									
นำความรู้ที่ได้จากการศึกษามาประยุกต์ใช้ในการปรับปรุงตัวอักษร				←	→							
เขียนโปรแกรมโดยใช้ MATLAB							←	→				
เปรียบเทียบอัลกอริทึมว่าอัลกอริทึมใดทำให้ตัวอักษรสมบูรณ์ที่สุด								←	→			

กิจกรรม	ปี 2548		ปี 2549									
	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.
สรุปผลที่ได้จากการศึกษาและแนวทางการพัฒนาที่ได้จากการศึกษา										←	→	
จัดทำรูปเล่มโครงงาน											←	→

1.6 ผลที่คาดว่าจะได้รับ

1.6.1 สามารถศึกษาและเลือกใช้อัลกอริทึมที่ดีที่สุด ในการปรับปรุงตัวอักษรที่ได้จากการสแกนได้

1.6.2 สามารถปรับปรุงตัวอักษรที่ได้จากการสแกนซึ่งเป็นภาพสี ระดับเทา หรือขาวดำให้มีความสมบูรณ์ขึ้น

1.6.3 สามารถนำโครงงานนี้ไปใช้ประโยชน์ในการปรับปรุงภาพในรูปแบบอื่นๆได้

1.6.4 สามารถส่งเสริมการพัฒนาเทคโนโลยีที่สามารถใช้งานได้ โดยใช้ทุนต่ำ

1.7 รายละเอียดงบประมาณของโครงงาน

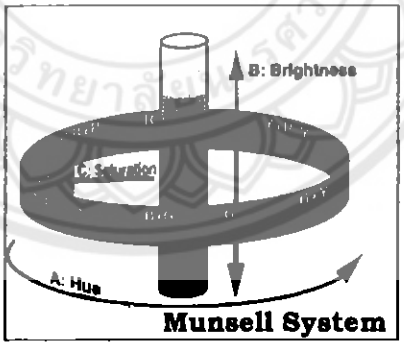
1.7.1 ค่าเอกสารและค่าเข้าเล่ม	900 บาท
1.7.2 ค่าหมึกพิมพ์	500 บาท
1.7.3 ค่ากระดาษ	200 บาท
1.7.4 ค่าแผ่นซีดี	100 บาท
1.7.5 ค่าวัสดุอื่น ๆ	300 บาท
รวมเป็นเงิน	2,000 บาท

หมายเหตุ ทุกรายการสามารถถัวเฉลี่ย

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

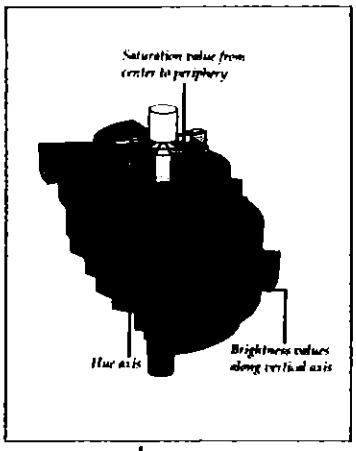
2.1 ภาพดิจิตอล (Digital Image)

ภาพดิจิตอล คือ ภาพที่ได้จากการถ่ายด้วยกล้องดิจิตอล หรือการสแกนภาพโดยสแกนเนอร์ ภาพดิจิตอล (Digital photography) แตกต่างจาก Analog photography หลายแง่มุมคือ ความละเอียด, สี, คุณภาพชิ้นงานหรือขั้นตอนการทำงานและการจัดการ ตลอดจนถึงค่าใช้ ง่าย ในธรรมชาติแสงสีเป็นพลังงานคลื่นแม่เหล็กไฟฟ้าที่มีสมบัติของคลื่น, สมบัติเฉพาะทางไฟฟ้า และแม่เหล็กอยู่รวมกัน โดยที่มีความยาวคลื่นที่ตามองเห็นในช่วง 380-780 nm ในช่วงความยาวคลื่นที่ตามองเห็นแตกต่างกันนี้คือ แสงสีที่ต่างกันในสเปกตรัมของสี ดวงตามนุษย์สามารถมอง เห็นสีที่แตกต่างกันได้จากที่ดวงตาของมนุษย์มีเซลล์รับแสงสีชนิด Cone ที่ต่างกัน 3 ชนิดคือ Red, Blue, Green Cone ที่ดูดแสงสีในช่วงคลื่น 570, 445, 535 nm การแปลผลสีอื่นๆ ที่ต่างไปจากนี้จะเกิดขึ้นได้จากการผสมแสงสีทั้ง 3 ในอัตราส่วนต่างๆ กัน โดยทฤษฎีแสงสีของ Albert Munsell หรือ Munsell system กล่าวไว้ว่าสีจะมีความแตกต่างกัน (Hue) 10 ระดับคือ แดง, เหลืองแดง, เหลือง, เขียวเหลือง, เขียว, น้ำเงินเขียว, น้ำเงิน, ม่วงน้ำเงิน, ม่วง และ แดงม่วง ทั้ง 10 ระดับ จะเรียงเป็นวงกลมดังรูป



รูปที่ 2.1 การเรียงกันเป็นวงกลมของสี 10 ระดับ

ในแต่ละระดับของสีที่ต่างกันจะมีความอิ่มตัวของสี 10 ระดับ และมีความแตกต่างกันในแง่ของความสว่างของสีอีก 10 ระดับ ทำให้ Munsell system สร้างระบบสีได้ดังรูป



รูปที่ 2.2 ระบบสี ที่สร้างจาก Munsell system

2.1.1 รูปร่างของภาพ (Image Shape)

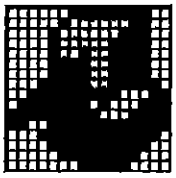
วัตถุที่มีอยู่ตามธรรมชาติและที่มนุษย์สร้างขึ้นจะมีรูปร่างที่แตกต่างกันไป ทั้งที่เป็นรูปทรงเรขาคณิตและไม่เป็นรูปทรงเรขาคณิต ในศาสตร์ของการประมวลผลภาพนั้น การกำหนดขอบเขตของภาพทุกภาพให้อยู่ในรูปสี่เหลี่ยม (Rectangular image model) เป็นวิธีที่ใช้กันมากที่สุด เนื่องจากทำให้การอ่านภาพ การจัดเก็บข้อมูลภาพในหน่วยความจำและการแสดงภาพออกทางอุปกรณ์ต่างๆ เป็นไปได้อย่างมีประสิทธิภาพ

การเก็บข้อมูลภาพลงในหน่วยความจำของคอมพิวเตอร์ ที่สามารถทำได้โดยการจองหน่วยความจำของเครื่องไว้ในรูปของตัวแปรอาร์เรย์ (Array) โดยค่าในแต่ละช่องของอาร์เรย์ที่แสดงถึงคุณสมบัติของจุดภาพ (pixel) และตำแหน่งของช่องอาร์เรย์เป็นตัวกำหนดตำแหน่งของจุดภาพ

2.1.2 พิกเซล (Pixel)

พิกเซล (Pixel) เป็นคำผสมของคำว่า "Picture" กับคำว่า "Element" หรือหน่วยพื้นฐานของภาพที่เทียบได้กับ "จุดภาพ" 1 จุด โดยแต่ละพิกเซลเปรียบได้กับสี่เหลี่ยมเล็กๆ ที่บรรจุค่าสี โดยถูกกำหนดตำแหน่งไว้บนเส้นกริดของแนวแกน x และแกน y หรือในตารางเมตริกซ์สี่เหลี่ยม ซึ่งภาพบิตแมปจะประกอบด้วยพิกเซลหลายๆ พิกเซล

พิกเซลของภาพเฉดสีขาว/ดำ



พิกเซลของภาพสี

รูปที่ 2.3 พิกเซลของภาพ

จำนวนพิกเซลของภาพแต่ละภาพเรียกว่า ความละเอียดในการแสดงผล (Resolution) โดยจะเทียบจำนวนพิกเซลกับความยาวต่อนิ้ว ดังนั้นจะมีหน่วยเป็น พิกเซลต่อนิ้ว (ppi : pixels per inch) หรือจุดต่อนิ้ว (dpi : dot per inch) ถ้าภาพขนาดเท่ากันแต่มีความละเอียดต่างกัน แสดงว่าจำนวนของพิกเซลต่างกัน และขนาดของจุดพิกเซลก็ต่างกันด้วย

2.1.3 ความละเอียดในการแสดงผล (Resolution)

คำนี้สามารถจะใช้ได้กับสถานการณ์ที่แตกต่างกัน เช่น ความละเอียดของการแสดงผลของเครื่องพิมพ์ หรือว่าความละเอียดในการแสดงผลของจอภาพ ดังนั้นความละเอียดในการแสดงผลจึงหมายถึง จำนวนหน่วยต่อพื้นที่

ความละเอียดในการแสดงผล เป็นความสามารถในการปรับระยะในการแสดงความละเอียดของภาพดิจิตอล ระยะห่างของความถี่ในการแสดงภาพ (ความถี่ในการทำวิธีการสุ่มภาพ(sampling)) จะถูกระบุในรูปของความละเอียดในการแสดงผล ซึ่งจะหมายถึงจุดต่อนิ้ว (dpi : dot per inch) หรือพิกเซลต่อนิ้ (ppi : pixels per inch) เป็นคำทั่วไปที่ใช้เรียกหรือบ่งบอกว่ามีการแสดงภาพอยู่ที่ระดับความละเอียดในการแสดงผลที่เท่าไร แต่อยู่ในขอบเขตที่จำกัด การเพิ่มความถี่การทำวิธีการสุ่มภาพ (sampling) ก็เป็นการเพิ่มความละเอียดในการแสดงผลด้วยเช่นกัน

รูปที่ 2.4 พิกเซลสามารถที่จะมองเห็นแต่ละพิกเซลได้โดยการขยายภาพที่เป็นดิจิตอล

Digital photograph เป็นภาพที่เกิดจากการเรียงต่อกันของพิกเซลเป็นภาพบิตแมป ซึ่งจะมีลักษณะที่เทียบเท่ากับภาพถ่ายหรือฟิล์มที่ถ่ายจากกล้องที่มีผลึกของเกลือเงินเฮไลด์ (Silver halide) กระจายอยู่ ความละเอียดของภาพถ่ายหรือฟิล์มขึ้นกับขนาดของผลึก ความละเอียดของภาพดิจิตอล (Digital photography) ก็ขึ้นกับจำนวนพิกเซลต่อพื้นที่หรือค่าความละเอียดในการแสดงผล นอกจากยังบอกเป็นพิกเซลต่อนิ้ว (ppi : pixels per inch) แล้วยังบอกจุดต่อนิ้ว (dpi : dot per inch) โดยความจริงนั้น จุดต่อนิ้ว (dpi : dot per inch) คือ ค่าความละเอียดในการแสดงผลที่จะใช้บอกความสามารถของอุปกรณ์เอาต์พุตแบบ Screening ทั้ง halftone (คือ Image setter) และ Dithering (Inkjet) เนื่องจากอุปกรณ์เหล่านี้จะสร้างภาพโดยการพ่นจุด และ พิกเซลต่อนิ้ว(ppi : pixels per inch) คือค่าความละเอียดในการแสดงผลที่ใช้สำหรับ ภาพดิจิตอล (Digital photography) ด้วยเหตุผลข้างต้น จึงเห็น

ว่าจำเป็นต้องมีการคำนวณค่าความละเอียดในการแสดงผลของภาพจากอุปกรณ์อินพุต และทำการตกแต่งให้เหมาะสมกับอุปกรณ์เอาต์พุต เพื่อได้ภาพที่ได้เหมาะสมที่สุด อุปกรณ์อินพุต เช่น กล้องดิจิตอลที่มีความสามารถแตกต่างกัน โดยการบันทึกความละเอียดของภาพตั้งแต่ 756x504 พิกเซล (Kodak DC50) ถึง 2000 x 3000 พิกเซล (Kodak DS505) สำหรับสแกนเนอร์จะใช้หน่วย จุดต่อนิ้ว (dpi : dot per inch) เหมือนอุปกรณ์เอาต์พุตในการบอกความละเอียด ที่การคำนวณค่าความละเอียดในการแสดงผล สำหรับการสแกนภาพเพื่อให้ได้คุณภาพที่ดีใช้สูตร

$$\text{Resolution} = \frac{\text{output size} \times \text{output device resolution dpi}}{\text{Original size}} \qquad (2.1)$$

ค่า Output size และ Original size เราทราบอยู่แล้ว แต่ค่า Output device resolution นั้น จะต้องมีการคำนวณในกรณีที่อุปกรณ์เอาต์พุตเป็นแบบ Screening (Inkjet หรือ Image setter) เนื่องจากอุปกรณ์พวกนี้พิมพ์งานแบบ Dot coverage method ค่าความละเอียดในการแสดงผล หน่วยเป็น line per inchs (lpi) สำหรับ Image setter ที่ซึ่งแต่ละ lpi ซึ่งสามารถสร้างจุดได้ต่างกัน ตั้งแต่ 10-30 lasers marks ในการคำนวณค่า output resolution สำหรับ Image setter นั้นจะต้องทำการคูณ output device resolution ที่มีหน่วยเป็น lpi กับค่า lasers masks per square cell ของเครื่องพิมพ์ สำหรับ Inkjet ซึ่งจะบอกความสามารถด้านความละเอียดเป็น DPI นั้น จุดที่สร้างจาก Inkjet จะมีความเล็กกว่าพิกเซล โดยประมาณ 1 พิกเซล จะเท่ากับ 3 จุดของ Inkjet ดังนั้น output resolution ของ Inkjet จะเท่ากับค่า output device resolution/3 ตัวอย่างเช่น ถ้าต้องการพิมพ์งานจากภาพด้วยเครื่อง Inkjet ขนาด 600 dpi จากภาพขนาด 4x5 นิ้ว และต้องการให้งานที่พิมพ์มีขนาดเท่าเดิม ที่ต้องสแกนด้วยความละเอียดเท่ากับ 4 x (600/3) / 4 = 200 dpi

อย่างไรก็ตามหลังจากการได้ภาพทั้งจากกล้องดิจิตอลหรือสแกนเนอร์ก็ตาม เราสามารถปรับค่าความละเอียดได้อีกด้วย คือการทำ interpolation โดยการปรับนี้จะทำได้ทั้งการเพิ่มจำนวนพิกเซลเข้าไปให้กับภาพเรียกว่า "upsampling" และ ทำการปรับ- ลดภาพ โดยตัดพิกเซลออกเรียกว่า "downsampling" การปรับจะทำให้ภาพมีขนาดใหญ่ขึ้นหรือเล็กลง วิธีการ interpolation จะมีหลายแบบ ดังนี้ nearest neighbor, bilinear และ bicubic

สำหรับงานที่แสดงผลภาพออกทางจอภาพ นั้น เนื่องจากจอภาพ ส่วนใหญ่แสดงผลได้ 72 จุดต่อนิ้ว (dpi : dot per inch) ดังนั้นการสแกนจะตั้งค่าที่ 72 จุดต่อนิ้ว (dpi : dot per inch) วิธีการนี้จะได้ภาพขนาดเท่ากับปกติ การสแกนด้วยค่าที่มากกว่านี้จะทำให้ได้ภาพที่มีขนาดใหญ่ขึ้น

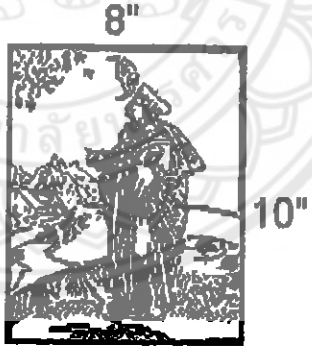
สมมุติให้ภาพ(Image)เป็นตัวแปรแบบอาร์เรย์ขนาด M x N (M แถว และ N คอลัมน์) ที่ใช้เก็บภาพขนาด M x N จุด (M จุดในแนวนอน และ N จุดในแนวตั้ง) ค่าสี (หรือความสว่าง ในกรณีที่

เป็นภาพระดับเทา(grey level)) ของจุดภาพในแถวที่ 5 คอลัมน์ที่ 4 จะตรงกับค่าของ Image (5,4) จะเห็นว่าเราใช้ตำแหน่งของจุดภาพทั้งสองแกนเป็นตัวชี้ค่าข้อมูลในอาร์เรย์

จากการใช้หน่วยความจำเพื่อทำการเก็บภาพในลักษณะที่กล่าวมานั้น เนื้อที่ในการเก็บภาพสามารถคำนวณได้จาก M x N x g เมื่อ g เป็นจำนวนเต็มที่แทนจำนวนบิตของข้อมูลในแต่ละจุดภาพ ตัวอย่างถ้า g มีค่าเท่ากับ 8 บิต

เราจะสามารถเก็บความแตกต่างของระดับสีที่เป็นไปสูงสุด 256 ระดับ ค่า M และ N จะเป็นตัวบอกถึงความละเอียดของภาพ สำหรับคอมพิวเตอร์ทั่วไปในระบบ VGA (Video Graphic Array) จะมีขนาด 640x480, 800x600 และ 1024x768 จุด เป็นต้น การกำหนดความละเอียดจะขึ้นอยู่กับงานที่จะใช้ในงานบางอย่างใช้ความละเอียดแค่ 30 x 50 จุด ก็พอแล้วแต่ในงานบางชนิดใช้ความละเอียดถึง 1000 x 1000 จุด ก็ยังไม่พอ

ปกติแล้วในการเก็บข้อมูลภาพโดยเครื่องมือต่าง ๆ จะเก็บตามมาตรฐานของโทรทัศน์ซึ่งมีอัตราส่วน x ต่อ y เท่ากับ 4:3 สำหรับเครื่องมือเก็บข้อมูลภาพที่ไม่เป็นไปตามอัตราส่วน 4:3 เมื่อนำภาพนี้ไปแสดงในจอภาพมาตรฐาน ก็จะทำให้ภาพที่แสดงนั้นมีขนาดจุดภาพไม่เป็นสี่เหลี่ยมจัตุรัส เช่น ในบางระบบอาจจะใช้ความละเอียดในการแสดงมีค่าเท่ากับ 640 x 512 ซึ่งจะทำให้ขนาดของจุดภาพที่ได้มีขนาดของด้านกว้างมีความยาวมากกว่าด้านสูง ซึ่งลักษณะดังกล่าวนี้เป็นหัวข้อที่ต้องสนใจสำหรับการเขียนโปรแกรมทางด้านกราฟิกและการจัดการข้อมูล



รูปที่ 2.5 ภาพ เอกสาร 8" x 10"

ภาพ เอกสาร 8" x 10" ถูกเก็บความละเอียดที่ 300 dpi (dot per inch) ความหมายคือ pixel dimension เป็น 2400 พิกเซล (8"x 300dpi) ต่อ 3000 พิกเซล (10" x 300 dpi)

2.1.4 บิต (BIT)

Bit ย่อมาจาก Binary Digit หมายถึง หน่วยความจำที่เล็กที่สุดของคอมพิวเตอร์ประกอบด้วยตัวเลข 2 จำนวน คือ 0 หมายถึงปิด และ 1 หมายถึงเปิด หรือสีขาวและสีดำ

ความลึกของบิต (Bit Depth)

ความลึกของบิต หมายถึง จำนวนบิตที่ใช้ในแต่ละพิกเซล ในกราฟิกแบบบิตแมปสีของพิกเซลถูกบันทึกโดยใช้บิต ถ้าใช้สีมากก็แสดงสีได้มากขึ้น ซึ่งถ้ามีหน่วยความจำ 2 บิต ในการเก็บรวบรวมข้อมูล สามารถใช้สีได้ทั้งหมด 2 เท่ากับ 4 สี คือ สามารถกลับสีได้ 4 วิธี คือ 00, 01, 10 และ 11 ถ้ามี 2 บิต สามารถสร้างสีให้กับพิกเซลทั้งหมด 4 เฉดสี

จำนวนสีสูงสุดที่เป็นไปได้ของแต่ละจุดภาพขึ้นอยู่กับจำนวนบิตที่ใช้ เมื่อมีการกำหนดให้ขนาดของบิตต่อจุด มากขึ้นจะทำให้จำนวนของสีมากขึ้นด้วย ตัวอย่างเช่น

1 บิต = 2^1=2 สี

2 บิต = 2^2=4 สี

4 บิต = 2^4=16 สี

8 บิต = 2^8=256 สี

16 บิต = 2^{16}=65536 สี เป็นต้น

ภาพขาวดำ (Binary image) อยู่ในรูปพิกเซลที่แต่ละพิกเซลจะมี 1 บิต ซึ่งแสดงได้ 2 ระดับสี คือ ขาวและดำ โดยค่า 0 จะเป็นสีดำ และ 1 จะเป็นสีขาว หรืออาจจะตรงกันข้าม

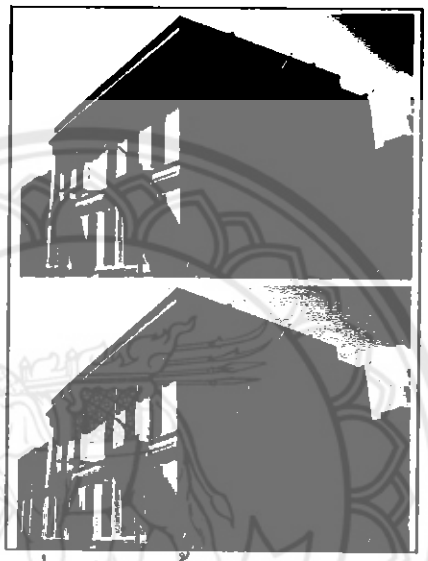
ภาพระดับเทา (Gray-scale image) เป็นการเรียงของพิกเซลที่ใช้ข้อมูลแบบ multiple bits อยู่ในช่วงระหว่าง 2- 8 บิต หรือมากกว่านั้น

ภาพสี (Color image) แบบทั่วไปนั้นจะมีค่า bit depth อยู่ในช่วง 8 – 24 บิต หรือมากกว่า ภาพที่มี 24 บิต นั้นคือ บิตจะถูกแบ่งออกเป็น 3 กลุ่ม 8 สำหรับสีแดง 8 สำหรับสีเขียว 8 สำหรับสีน้ำเงิน สีทั้งหมดจะถูกรวมกันเพื่อแสดงสีอื่นๆ 24 บิต ภาพสามารถแสดงค่าสีได้ถึง 16.7 ล้านสี (2^{24}) สำหรับ สแกนเนอร์นั้นได้เพิ่ม จำนวนบิตในการจับภาพเอกสารเป็น 10 บิต หรือมากกว่านั้น เพราะบ่อยครั้งหลังจากจับภาพเอกสารที่ 8 บิต จะมี noise รวมเข้าไปด้วย และเพื่อให้ภาพที่ออกนั้นมีความเหมือนใกล้เคียงกับที่มนุษย์ต้องการ อาจจะมีการใช้สีถึง 36, 48, 64 บิต แต่ในคอมพิวเตอร์ทั่วๆ ไปมักจะใช้สีไม่เกิน 32 บิต

สำหรับการแสดงข้อมูลภาพที่มีขนาด 1 บิตและ 8 บิตนั้นจะมีการทำงานที่จะใกล้เคียงกัน เนื่องจากหน่วยประมวลผลจะไม่สามารถจัดการกับข้อมูลที่เป็นบิตเดี่ยวๆ ได้ ดังนั้นในการแสดงข้อมูลออกทางจอภาพตัวโปรเซสเซอร์จะทำการก็อปปี้ข้อมูลทั้ง 8 บิต (1 ไบต์) ส่งให้กับจอภาพซึ่งในกรณีที่พิกเซลมีขนาด 1 บิต เมื่อโปรเซสเซอร์จะทำงานกับบิตแรกที่ต้องการแล้วก็จะทำการก็อปปี้ข้อมูลชุดใหม่ทันทีโดยที่ไม่เกี่ยวกับข้อมูลอีก 7 บิตที่เหลือส่วนในกรณีพิกเซลที่มีขนาด 8 บิต โปรเซสเซอร์จะทำการก็อปปี้ข้อมูลจุดใหม่ก็ต่อเมื่อโปรเซสเซอร์ทำงานกับทุกบิตแล้ว

ตัวอย่างสำหรับระบบที่มีความละเอียดเท่ากับ 800x600 และมีขนาด 16 บิตต่อพิกเซล จะสามารถแสดงสีได้ทั้งหมด 65536 ระดับและต้องใช้เนื้อที่ในการเก็บเท่ากับ 800x600x16 บิต

Dynamic Range คือ ช่วงระหว่างความต่างของสีที่สว่างที่สุดและ มืดที่สุดของภาพ ถ้ามีค่า Dynamic Range สูง จะทำให้ภาพที่ได้มานั้นมีความคมชัดสูง แม้ว่า Dynamic range จะไม่เกี่ยวข้องกับจำนวนของสีเลยก็ตาม Dynamic Range นั้นยังอธิบายความสามารถของระบบดิจิตอล ว่าระบบนั้นว่ามีความสามารถที่จะแสดงให้เห็นถึงสีที่ประกอบกันอยู่ ด้วยความสามารถนี้ก็เป็นส่วนสำคัญสำหรับเอกสารที่จะต้องใช้การแสดงลำดับชั้นสีที่ต่างกัน และสำหรับภาพนั้นก็อาจสำคัญอันดับแรกในเรื่องของคุณภาพที่ต้องการ



รูปที่ 2.6 Dynamic range: ภาพที่อยู่ด้านบนนั้นมี Dynamic range กว้างแต่จำกัดจำนวนการแสดงของสี ภาพล่าง มี Dynamic range ที่แคบแต่มีจำนวนของสีที่มากกว่า

2.2 มาตรฐานของสี

มาตรฐานของสีที่ใช้อยู่ในปัจจุบันมีอยู่หลายระบบด้วยกัน ทั้งนี้จะขึ้นอยู่กับการนำไปใช้ แต่โดยทั่วไปแล้วทุกมาตรฐานจะมีแนวคิดเดียวกันคือ การแทนจุดสีด้วยจุดที่อยู่ภายในสเปส 3 มิติ โดยจะมีแกนอ้างอิงสำหรับจุดสีนั้นในสเปสซึ่งแต่ละแกนจะมีความเป็นอิสระต่อกัน ตัวอย่างเช่นในระบบ RGB จะมีแกนสีคือ แกนสีแดง เขียว และน้ำเงิน ส่วนในระบบ HLS จะมีแกนเป็น ค่าสี (hue) ความสว่าง (lightness) และความบริสุทธิ์ของสี (saturation) สำหรับตัวอย่างระบบสีที่นิยมใช้กันได้แก่ ระบบ RGB HSV (Hue Saturation Value) และ HLS (Hue Lightness Saturation)

2.2.1 ระบบสี RGB

ระบบสี RGB เป็นระบบสีของแสง ซึ่งเกิดจากการหักเหของแสงผ่านแท่งแก้วปริซึมจะเกิดแถบสีที่เรียกว่า สีรุ้ง (Spectrum) ซึ่งจะแยกสีตามที่สายตามองเห็นได้ 7 สี คือ แดง แสด เหลือง เขียว น้ำเงิน คราม ม่วง ซึ่งเป็นพลังงานอยู่ในรูปของรังสี ที่มีช่วงคลื่นที่สายตา สามารถมองเห็นได้ แสงสี

ม่วงมีความถี่คลื่นสูงที่สุด ส่วนคลื่นแสงที่มีความถี่สูงกว่าแสงสีม่วง เรียกว่า อัลตราไวโอเลต (Ultra Violet) และคลื่นแสงสีแดง มีความถี่คลื่นต่ำที่สุด คลื่นแสงที่ต่ำกว่าแสงสีแดง เรียกว่า อินฟราเรด (Infrared) คลื่นแสงที่มีความถี่สูงกว่าสีม่วง และต่ำกว่าสีแดงนั้น สายตาของมนุษย์ไม่สามารถรับได้ และเมื่อศึกษาดูแล้วแสงสีทั้งหมดเกิดจากแสงสี 3 สี คือ สีแดง (Red) สีน้ำเงิน (Blue)และสีเขียว (Green)ทั้งสามสีถือเป็นแม่สีของแสง เมื่อนำมาฉายรวมกันจะทำให้เกิดสีใหม่ อีก 3 สี คือ สีแดงมาเจนต้า (Magenta) สีฟ้าไซแอน (Cyan) และสีเหลือง (Yellow) และถ้าฉายแสงสีทั้งหมดรวมกันจะได้แสงสีขาว จากคุณสมบัติของแสงนี้เราได้นำมาใช้ประโยชน์ทั่วไป ในการบันทึกภาพวิดีโอ การฉายภาพยนตร์ ภาพโทรทัศน์ การสร้างภาพเพื่อการนำเสนอทางจอคอมพิวเตอร์ และการจัดแสงสีในการแสดง เป็นต้น

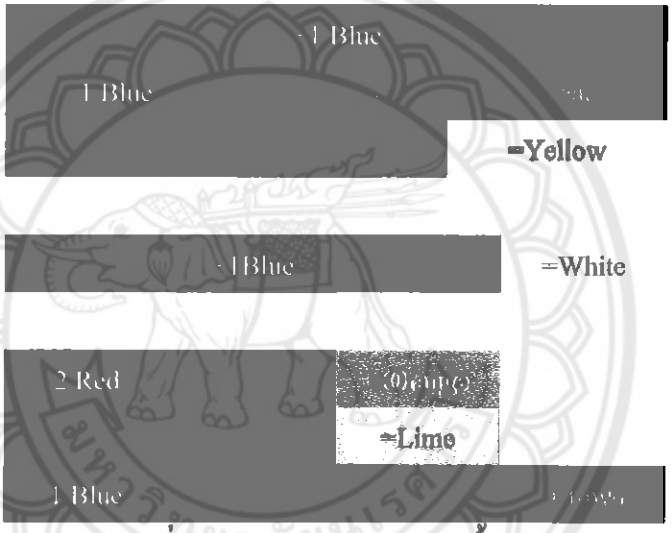
รูปที่ 2.7 ระบบสี RGB

แบบ Additive

ระบบสี RGB เป็นระบบสีที่เกิดจากการรวมกันของแสงสีแดง เขียวและน้ำเงิน โดยมีการรวมกันแบบ Additive ซึ่งโดยปกติจะนำไปใช้ในจอภาพแบบ CRT (Cathode ray tube) ปกติเมื่อพูดถึงสี มักจะนึกถึงแม่สี 3 สี แต่อย่างไรก็ตาม การใช้สีกับงานกราฟิกในคอมพิวเตอร์ จะมีรายละเอียดหลายประการ ซึ่งเป็นลักษณะเฉพาะ ดังนั้นจึงควรทราบระบบสีของคอมพิวเตอร์ก่อนระบบสีของคอมพิวเตอร์ จะเกี่ยวข้องกับการแสดงผลแสงที่แสดงบนจอคอมพิวเตอร์ โดยมีลักษณะการแสดงผล คือ ถ้าไม่มีแสดงผลสีใดเลย บนจอภาพจะแสดงเป็น “สีดำ” ถ้าหากสีทุกสีแสดงผลพร้อมกัน จะเห็นสีบนจอภาพเป็น "สีขาว" ส่วนสีอื่นๆ เกิดจากการแสดงสีหลายๆ สี แต่มีค่าแตกต่างกัน ซึ่งการแสดงผลลักษณะนี้ เรียกว่า การแสดงสีระบบ Additive สีในระบบ Additive ประกอบด้วยสีหลัก 3 สี (เช่นเดียวกับแม่สี) คือ สีแดง (Red) สีเขียว (Green) และ สีน้ำเงิน (Blue) เรียกรวมกันว่า RGB ซึ่งมีรูปแบบการผสมสีของ RGB ดังนี้

รูปที่ 2.8 การแสดงสีระบบ Additive

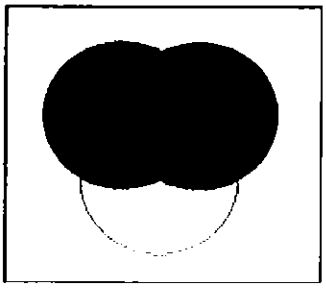
การผสมกันของแม่สีทั้งสาม



รูปที่ 2.9 การผสมกันของแม่สีทั้งสาม

แบบ Subtractive

ระบบสี Subtractive มีลักษณะที่ตรงข้ามกับระบบ Additive โดยสีแต่ละสีจะได้จากการลบสีต่างๆ ออกไปจากระบบ หากไม่มีการแสดงสีใดๆ จะแสดงผลเป็นสีขาว ขณะที่การแสดงสีทุกสี จะปรากฏเป็นสีดำ



รูปที่ 2.10 การแสดงสีแบบ Subtractive

ในการใช้งานระบบสีRGB ยังมีการสร้างมาตรฐานที่แตกต่างกันออกไปที่นิยมใช้งานได้แต่ RGB_{CIE} และ RGB_{NTSC}

ระบบสีแบบ RGB ของ CIE

เป็นระบบสีที่พัฒนาขึ้นโดย CIE (Commission International l 'Eclairage) ซึ่งอ้างอิงสีด้วยสีแดงที่ 700 nm สีเขียวเท่ากับ 546.1 nm และสีน้ำเงิน 435.8 nm

ระบบสีแบบ RGB ของ NTSC

เป็นระบบที่พัฒนาโดย NTSC (National Television System Committee) เพื่อใช้สำหรับการแสดงภาพของจอภาพแบบ CRT เป็นมาตรฐานสำหรับผู้ผลิตแบบ CRT ให้มีลักษณะเดียวกัน

2.2.2 ระบบสี HSV

ระบบสี HSV (Hue Saturation Value) เป็นการพิจารณาสีโดยใช้ Hue Saturation และ Value ซึ่ง

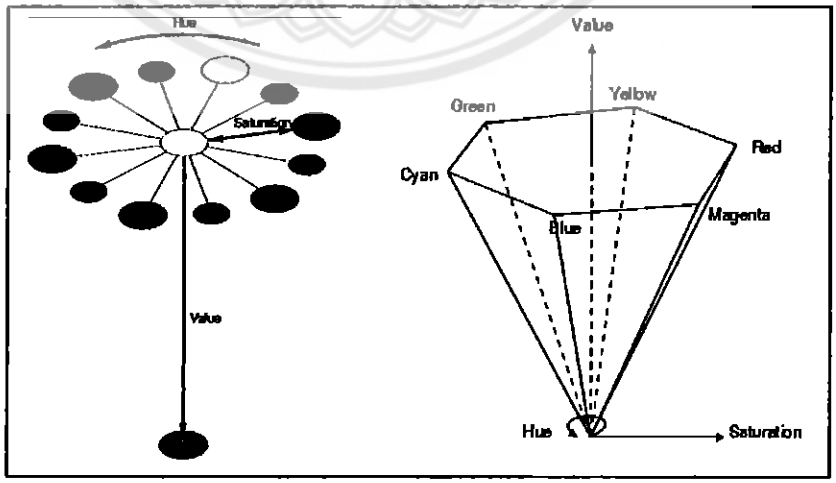
Hue คือค่าสีของสีหลัก (แดง เขียวและน้ำเงิน) ในทางปฏิบัติจะอยู่ระหว่าง 0 และ 255 ซึ่งถ้า Hue มีค่าเท่ากับ 0 จะแทนสีแดง และเมื่อ Hue มีค่าเพิ่มขึ้นเรื่อยๆ สีก็จะเปลี่ยนแปลงไปตามค่าสเปกตรัมของสีจนถึง 256 จึงจะกลับมาเป็นสีแดงอีกครั้งซึ่งสามารถแทนให้อยู่ในรูปขององศาได้ดังนี้คือ สีแดง = 0 องศา สีเขียวเท่ากับ 120 องศา สีน้ำเงินเท่ากับ 240 องศา

Hue สามารถคำนวณได้จากระบบสี RGB ได้ดังนี้

$$red_h = red - \min(red, green, blue) \tag{2.2}$$

$$green_h = green - \min(red, green, blue) \tag{2.3}$$

$$blue_h = blue - \min(red, green, blue) \tag{2.4}$$



รูปที่ 2.11 แสดงระบบสี HSV

จากลักษณะโมเดลของระบบ Hue พบว่าจะมีค่าอย่างน้อยหนึ่งค่าที่จะเท่ากับ 0 แต่ถ้ามีสองค่าเท่ากับ 0 แล้ว Hue จะเป็นมุมของสี (ค่าสี) มีค่าเป็นไปตามสี่ที่สามและถ้าทั้งสามสีมีค่าเท่ากับ 0 แล้วจะทำให้ไม่มีค่าของ Hue หรือสีที่ได้จะมีค่าเท่ากับสีขาวนั่นเอง ตัวอย่างเช่น จอภาพขาว-ดำ ถ้าเกิดมีสีใดสีหนึ่งมีค่าเท่ากับ 0 จะทำให้ค่าสีที่ได้เป็นไปตามสีที่เหลือ การให้น้ำหนักในการพิจารณาเมื่อสีแดงมีค่าเท่ากับ 0

$$Hue = \frac{(240 \times blue) + (120 \times green)}{blue_h + green_h} \tag{2.5}$$

Saturation คือความบริสุทธิ์ของสีซึ่งถ้า Saturation มีค่าเท่ากับ 0 แล้วสีที่ได้จะไม่มี Hue ซึ่งจะเป็นสีขาวล้วนแต่ถ้า Saturation มีค่าเท่ากับ 255 แสดงว่าจะไม่มีแสงสีขาวผสมอยู่เลย

Saturation สามารถคำนวณได้ดังนี้

$$Satuation = \frac{\max(red, green, blue) - \min(red, green, blue)}{\max(red, green, blue)} \tag{2.6}$$

Value คือความสว่างของสี ซึ่งสามารถวัดได้โดยค่าความเข้มของความสว่างของแต่ละสีที่ประกอบกันสามารถคำนวณได้จาก

$$value = \max(red, green, blue) \tag{2.7}$$

2.2.3 ระบบสีแบบ HSL

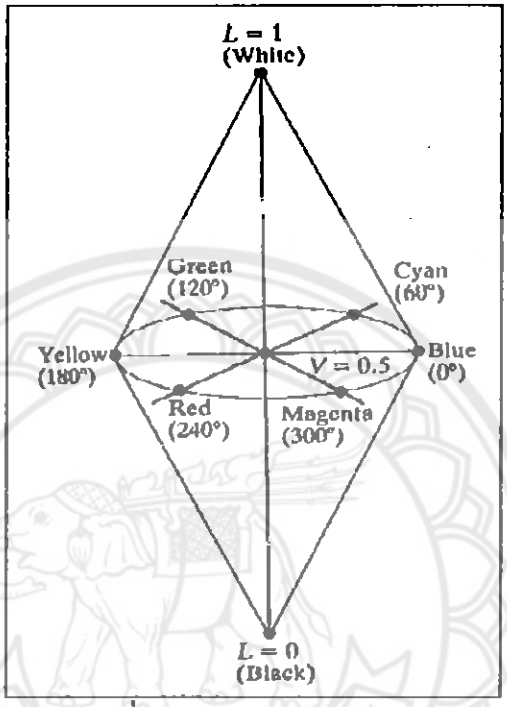
ระบบสีแบบ HSL (Hue saturation lightness) พัฒนาโดย Teletromix Incorporated จะมีลักษณะคล้ายกับ HSV ดังนี้คือสีของระบบจะขึ้นอยู่กับ Hue Lightness และ Saturation

Hue คือ ค่าของสีหลักซึ่งมีสีน้ำเงินอยู่ที่ 0 องศา สีเขียวอยู่ที่ 120 องศา และสีแดงอยู่ที่ 240 องศา

Lightness คือ ค่าความสว่างซึ่งจะมีค่าเปลี่ยนแปลงตามแนวแกน L โดยที่ L = 0 จะเป็นสีดำ L = 1 จะเป็นสีขาว สามารถคำนวณได้ดังนี้

$$lightness = \frac{\max(red, green, blue) - \min(red, green, blue)}{2} \tag{2.8}$$

ค่า Lightness บ่งถึงความสว่างและมืด ค่า Hue จะมีค่าตั้งแต่ 0 องศา ถึง 360 องศา ระบบสี Lab เป็นอีกระบบที่ค่า L ชี้ถึง lightness ค่า a เป็นสมดุลระหว่างสี Green กับ Magenta ค่า b เป็นสมดุลระหว่างสี Blue กับ Yellow และระบบสี XYZ ที่มีมาตรฐานค่า Y คือความ Lightness ค่า XZ บ่งถึงค่า Hue และ Saturation ระบบ XYZ เป็นระบบที่ใช้กันในงานสีระดับสูง



รูปที่ 2.12 แสดงระบบสี HSL

Saturation คือความบริสุทธิ์ของสีสามารถหาได้ดังนี้คือ

$$suturation = \begin{cases} \dfrac{\max(red,green,blue)+\min(red,green,blue)}{\max(red,green,blue)-\min(red,green,blue)} & if\ L <= 0.5 \\ \dfrac{\max(red,green,blue)-\min(red,green,blue)}{2-\max(red,green,blue)-\min(red,green,blue)} & if\ ohterwise \end{cases} \quad (2.9)$$

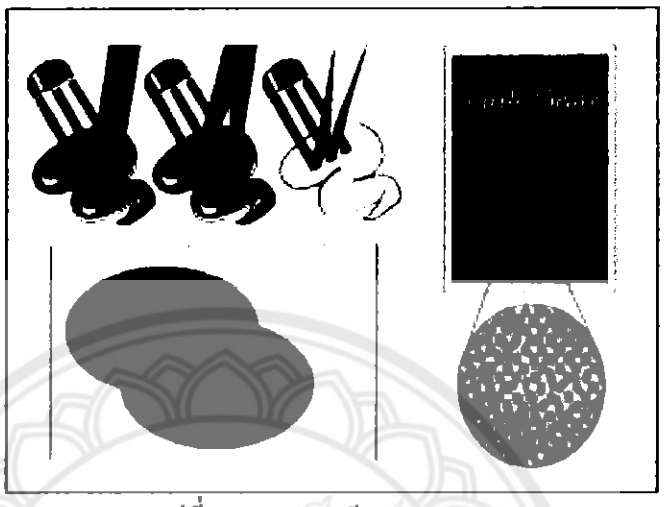
2.2.4 ระบบสีแบบ CMY

CMY (Cyan Magenta Yellow) เป็นระบบสีที่พัฒนาขึ้นมาใช้สำหรับการพิมพ์ภาพสีโดยมีสีหลักคือสี Cyan Magenta และ Yellow ซึ่งเรียกว่า Subtractive primaries Color (สีแดง เขียวและน้ำเงิน เรียกว่า Additive primaries Color) ระบบสีแบบ CMY สามารถหาได้โดยการนำเอาสีในระบบ RGB ลบกับสีขาวดังนี้คือ

$$C = 1 - R \quad (2.10)$$

$$M = 1 - G \quad (2.11)$$

$$Y = 1 - B \quad (2.12)$$



รูปที่ 2.13 ระบบสี CMY

ระบบสี CMY จะนำไปใช้สำหรับการพิมพ์ภาพสีแต่ยังไม่ดีเท่าที่ควร **เนื่องจากว่ายังไม่สามารถสร้างสีดำได้อย่างถูกต้อง ดังนั้นจึงมีการใช้ระบบ CMYK แทนโดย**

$$C = C - K \quad (2.13)$$

$$M = M - K \quad (2.14)$$

$$Y = Y \ K \quad (2.15)$$

$$K = \min(C, M, Y) \quad \text{K เป็นสีที่ 4 แทนสีดำ} \quad (2.16)$$

ถ้าจะต้องส่งไปพิมพ์งาน จะต้องทำการเก็บงาน (Save) ให้อยู่ในรูปแบบ (Mode) CMYK ไม่ใช่ RGB

การผสมสีแบบ CMYK

Magenta paint +	=Red
+ Cyan paint	=Green
Cyan paint + Magenta paint	=blue

รูปที่ 2.14 ระบบสี CMYK

2.2.5 ระบบสีแบบ YUV

ระบบสีแบบ YUV ใช้สำหรับโทรทัศน์แบบ PAL และ SECAM ซึ่งยังมีใช้อยู่ในหลาย ๆ ประเทศโดย Y คือค่าความสว่างของภาพ ส่วนสัญญาณ U และ V เป็นสัญญาณที่เก็บค่าสีของภาพ ต่อมาได้มีระบบ YIQ มาใช้แทน เนื่องจากพบว่าสัญญาณ I และ Q สามารถลด Bandwidth ได้มากกว่าสัญญาณ U และ V ในขณะที่ได้ภาพที่มีคุณภาพเท่ากัน

2.2.6 ระบบสีแบบ YIQ

ระบบสีแบบ YIQ เป็นระบบที่ใช้ใน TV Broadcasting สำหรับ NTSC ประโยชน์หลักก็เพื่อให้ใช้งานได้กับโทรทัศน์แบบขาว-ดำ โดยที่ y คือความสว่างของภาพ ส่วน I และ Q จะเป็นสัญญาณที่เข้ารหัสสีของภาพไว้ ดังนั้นสำหรับโทรทัศน์ขาว- ดำ นั้นก็สามารถใช้ค่า Y ค่าเดียวแล้วสามารถได้ภาพที่สมบูรณ์

2.2.7 ระบบสีแบบ XYZ

ระบบสีแบบ XYZ เป็นระบบสีที่ CIE ได้กำหนดให้มีขึ้นเป็นมาตรฐาน เนื่องจากในระบบสี RGB ยังไม่สามารถสร้างสีที่เป็นไปได้ทั้งหมด ดังนั้นจึงได้มีตั้งระบบสี XYZ ซึ่งเป็นระบบสีที่สมมุติขึ้น

2.3 รูปแบบของไฟล์ภาพ (Digital file format)

ภาพดิจิตอลที่จัดเก็บ (save) ในรูปแบบต่าง ๆ กันมีอยู่มากมายดังนี้ TIFF, BMP, GIF, JPEG, PSD ฯลฯ

2.3.1 ไฟล์สกุล GIF (CompuServe Graphics Interchange Format)

เป็นไฟล์ชนิดบิตแมป สามารถนำมาแสดงเป็นรูปกราฟิกได้ทุกระบบและเก็บรายละเอียดสีได้ไม่เกิน 8 บิต มีความละเอียดของจุดสี (Pixel) สูงสุด 64,000 x 64,000 จุด ไฟล์ .GIF มีขนาดไม่ใหญ่มากนักแต่มีข้อจำกัดคือ สามารถแสงดสีได้สูงสุด 256 สีเท่านั้น เว็บเพจที่มีรูปเคลื่อนไหวต่าง ๆ จะมีนามสกุล .GIF โดยนำเอาภาพต่าง ๆ เป็นเฟรมมาผ่านโปรแกรมทำภาพเคลื่อนไหวเช่น GIF animation แล้วจะได้ภาพเคลื่อนไหวตามต้องการ

ไฟล์สกุล GIF เป็นไฟล์กราฟิกมาตรฐานที่ทำงานบนอินเทอร์เน็ต มักจะใช้เมื่อ

- ต้องการไฟล์ที่มีขนาดเล็ก
- จำนวนสีและความละเอียดของภาพไม่สูงมากนัก
- ต้องการพื้นแบบโปร่งใส
- ต้องการแสดงผลแบบโครงร่างก่อน แล้วค่อยแสดงผลแบบละเอียด

- ต้องการนำเสนอภาพแบบภาพเคลื่อนไหว

จุดเด่น

- มีขนาดไฟล์ต่ำ
- สามารถทำพื้นของภาพให้เป็นพื้นแบบโปร่งใสได้ (Transparent)
- มีระบบแสดงผลแบบหยาบและค่อยๆ ขยายไปสู่ละเอียดในระบบ Interlace
- มีโปรแกรมสนับการสร้างจำนวนมาก
- เรียกดูได้กับ Graphics Browser ทุกตัว
- ความสามารถด้านการนำเสนอแบบภาพเคลื่อนไหว (Gif Animation)

จุดด้อย

- แสดงสีได้เพียง 256 สี

ไฟล์ .GIF มี 2 สกุล ได้แก่

- GIF87 พัฒนาขึ้นในปี ค.ศ. 1987

เป็นไฟล์กราฟิกรุ่นแรกที่สนับสนุนการนำเสนอบนอินเทอร์เน็ต เป็นไฟล์ที่มีขนาดเล็กและแสดงผลสีได้เพียง 256 สี และกำหนดให้แสดงผลแบบโครงร่างได้ (Interlace)

- GIF89A พัฒนาขึ้นในปี ค.ศ. 1989

เป็นไฟล์กราฟิกที่พัฒนาต่อจาก GIF87 โดยเพิ่มความสามารถการแสดงผลแบบพื้นโปร่งใส (Transparent) และการสร้างภาพเคลื่อนไหว (GIF Animation) เป็นไฟล์กราฟิกที่มีความสามารถพิเศษโดยนำเอาไฟล์ภาพหลายๆ ไฟล์มารวมกันและนำเสนอภาพเหล่านั้นโดยอาศัยการหน่วงเวลามีการใส่รูปแบบการนำเสนอลักษณะต่างๆ (Effects) ในลักษณะภาพเคลื่อนไหว

นอกจากนี้ ยังมีคุณสมบัติพิเศษอีก 3 ประการ คือ

- **Interlaced** คือ การแสดงภาพออกเป็นช่วงๆ แทนที่จะไล่จากบนลงล่างเหมือนปกติ
- **Progressive** คือ การแสดงภาพหยาบก่อนแล้วจึงค่อยๆ ชัดขึ้น
- **Transparent** คือ การแสดงภาพโดยไม่มีพื้นฉากหลัง

2.3.2 ไฟล์สกุล JPG (Joint Photographer's Experts Group)

ไฟล์ JPEG หรือ JPG ถูกพัฒนาขึ้นเพื่อใช้งานกับภาพที่มีสีสันสดใสและมีความละเอียดสูงมาก สามารถแสดงสีได้ถึง 16.7 ล้านสี ไฟล์มีขนาดเล็กมาก นิยมใช้กันมาก

เป็นอีกไฟล์หนึ่งที่นิยมใช้บนอินเตอร์เน็ต มักใช้กรณี

- ภาพที่ต้องการนำเสนอมีความละเอียดสูง และใช้สีจำนวนมาก (สนับสนุนถึง 24 บิต)
- ต้องการบีบไฟล์ตามความต้องการของผู้ใช้

- ไฟล์ชนิดนี้มักจะใช้กับภาพถ่ายที่นำมาสแกน และต้องการนำไปใช้บนอินเตอร์เน็ต เพราะให้ความคมชัดและความละเอียดของภาพสูง

จุดเด่น

- สนับสนุนสีได้ถึง 24 บิต
- สามารถกำหนดค่าการบีบไฟล์ได้ตามที่ต้องการ
- มีระบบแสดงผลแบบหยาบและค่อยๆ ขยายไปสู่ละเอียดในระบบ Progressive
- มีโปรแกรมสนับสนุนการสร้างจำนวนมาก
- เรียกดูได้กับ Graphics Browser ทุกตัว
- ตั้งค่าการบีบไฟล์ได้ (compress files)

จุดด้อย

- ทำให้พื้นของรูปโปร่งใสไม่ได้

JPEG เป็นไฟล์รูปภาพที่ถูกบีบขนาดเหมือนกันไฟล์ GIF จะมีขนาดเล็กกว่าไฟล์รูปแบบ GIF มาก โดยใช้เทคนิค LOSSY ซึ่งคุณภาพของภาพจะต่ำลงมาตามขนานที่บีบ แต่ไม่จำกัดจำนวนสี มักใช้ในลักษณะของภาพถ่าย

ข้อเสียของการบีบไฟล์ (Compress File)

กำหนดค่าการบีบไฟล์ไว้สูง (1 - 10) แม้ว่าจะช่วยให้ขนาดของไฟล์มีขนาดต่ำ แต่ก็มีข้อเสียคือ เมื่อมีการส่งภาพจากเครื่องแม่ข่าย (Server) นั้นไปแสดงผลที่เครื่องลูกข่าย (Client) จะทำให้การแสดงผลช้ามาก เพราะต้องเสียเวลาในการคลายไฟล์ ดังนั้นการเลือกค่าการบีบไฟล์ ควรกำหนดให้เหมาะสมกับภาพแต่ละภาพ

2.3.3 ไฟล์สกุล PNG (Portable Network Graphics)

เป็นไฟล์ที่ถูกพัฒนาขึ้นมาแทนไฟล์ GIF และไฟล์ JPG ในอนาคต ไฟล์ PNG สามารถใช้ได้กับทุกระบบปฏิบัติการ

จุดเด่น

- สนับสนุนสีได้ถึงตามค่า True color (16 บิต, 32 บิต หรือ 64 บิต)
- สามารถกำหนดค่าการบีบไฟล์ได้ตามที่ต้องการ
- มีระบบแสดงผลแบบหยาบและค่อยๆ ขยายไปสู่ละเอียด (Interlace)
- สามารถทำพื้นโปร่งใสได้

จุดด้อย

- หากกำหนดค่าการบีบไฟล์ไว้สูง จะใช้เวลาในการคลายไฟล์สูงตามไปด้วย แต่ขนาดของไฟล์จะมีขนาดต่ำ
- ไม่สนับสนุนกับ Graphic Browser รุ่นเก่า สนับสนุนเฉพาะ IE 4 และ Netscape 4

- ความละเอียดของภาพและจำนวนสีขึ้นอยู่กับ Video Card
- โปรแกรมสนับสนุนในการสร้างมีน้อย

สรุปก็คือ รูปแบบของภาพที่นิยมก็คือ .jpg กับ .gif เพราะว่าไฟล์มีขนาดเล็กมากและ .gif สามารถนำมาทำเป็นภาพเคลื่อนไหวได้ อย่างไรก็ตามยังมีอีกรูปแบบหนึ่งคือ .png แต่ยังมีปัญหาคือ browser บางตัวยังไม่สามารถแสดงผลได้

2.3.4 ไฟล์สกุล BMP (MS-Windows bitmap format)

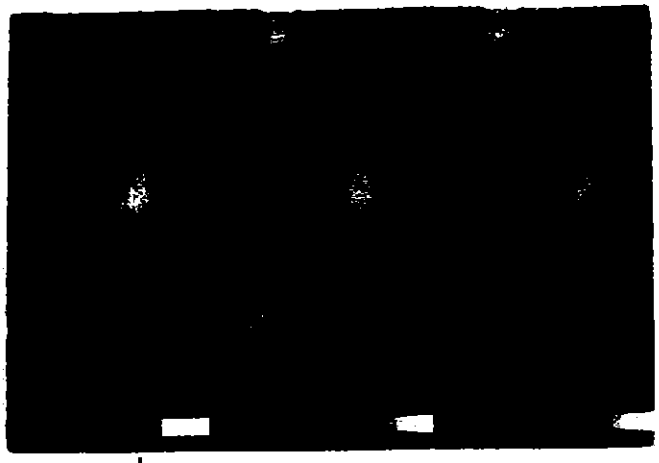
ไฟล์ BMP เป็นรูปแบบพื้นฐานที่ใช้งานได้ดีกับโปรแกรมที่ทำงานภายใต้วินโดว์ ซึ่งมีความเร็วสูง เป็นไฟล์มีขนาดใหญ่ เพราะไม่ได้รับการบีบย่อข้อมูลเหมือนกับ .GIF หรือ .JPG ไฟล์ BMP ที่เห็นบ่อยๆ คือ ภาพวอลล์เปเปอร์ที่แสดงบนจอภาพของวินโดว์ โดยสามารถแสดงได้ตั้งแต่ 2, 16, 256 และ 16 ล้านสี

2.3.5 ไฟล์สกุล TIFF (Tagged Image File Format)

สามารถใช้กับระบบปฏิบัติการหลายชนิด ที่สามารถจัดเก็บภาพสีทั้งโหมด Index color, RGB และ CMYK รวมถึงขาวดำ และระดับเทา (Gray-scale) ภาพส่วนใหญ่จะเป็นภาพที่เกิดจากการสแกนรูปภาพต่าง ๆ และจะจัดเก็บเป็นไฟล์นามสกุล .TIF

ไฟล์แบบ TIFF เป็นรูปแบบที่มีคุณภาพความคมชัดของภาพสูงที่สุด ไม่ว่าจะย่อหรือขยายภาพคุณภาพที่แสดงก็ยังคงเดิม แต่จะมีขนาดใหญ่เนื่องจากว่ามีการรวมเอาข้อมูลจากบิตแมป วันที่และเวลาที่ไฟล์ถูกสร้าง รวมทั้งซอฟท์แวร์ที่ใช้ เหมาะสมในวงการสื่อสิ่งพิมพ์

แต่ในทางปฏิบัติ การทำงานกับภาพจะใช้รูปแบบการจัดเก็บที่เป็นสากลนิยมคือ JPEG และ GIF ความแตกต่างของ 2 รูปแบบ ในทางปฏิบัติคือ การจัดเก็บงานในรูปแบบ GIF นั้นจะทำให้งานมีขนาดความจุเล็ก เนื่องจากรูปที่จัดเก็บแบบ GIF เป็น Index color คือให้สีน้อยกว่าปกติ ส่วนรูปที่จัดเก็บแบบ JPEG จะสามารถควบคุมคุณภาพจัดเก็บได้ 3 ระดับคือ ต่ำ กลางและสูง ถ้าภาพที่ต้องการจัดเก็บมีความละเอียดต่ำ การจัดเก็บแบบ GIF และ JPEG จะมองไม่เห็นความแตกต่าง แต่ภาพ JPEG จะมีความจุมากกว่า GIF และการจัดเก็บอีกรูปแบบคือ PSD เป็นรูปแบบของ Photoshop การจัดเก็บวิธีนี้จะทำให้ภาพเก็บข้อมูลทุกอย่างรวมถึงการทำงาน layers ด้วยทำให้ง่ายต่อการทำงานเมื่อเปิดภาพขึ้นมาทำงานใหม่ สำหรับคุณภาพของภาพดิจิตอลถูกกำหนดโดย จำนวนสีที่ใช้, ชนิดของการจัดเก็บ, จำนวน level, จำนวนบิตของภาพ และความละเอียด (Resolution) ความหมายของจำนวนสีและชนิดของการจัดเก็บได้กล่าวไว้แล้ว สำหรับจำนวน level คือ ระดับการไล่สีจากจุดที่อ่อนสู่จุดที่เข้ม ภาพที่ใช้จำนวน level มาก จะให้คุณภาพที่ดีดังรูป โดยส่วนใหญ่ภาพที่มีคุณภาพดีปกติจะใช้ 256 levels แต่อุปกรณ์ Input อาจสร้างภาพได้คุณภาพมากกว่านี้คือ 1,024 levels ถึง 16,384 levels



รูปที่ 2.15 คุณภาพของภาพตามจำนวนบิต

สำหรับจำนวนบิตของภาพจะมีผลต่อคุณภาพของภาพ โดยที่ภาพ 1 บิต จะแสดงผลของสีได้ 2 สีคือ ขาว-ดำ ในแต่ละพิกเซลในขณะทำงาน 8 บิต จะสามารถสร้างความเข้ม-จาง (Gray-scale) ให้กับภาพได้ 256 ระดับ (level) ภาพระดับ RGB คือ ภาพที่ประกอบจากภาพ 8 บิต 3 channel รวมกัน จึงถือเป็นภาพ 24 บิต ($(28)^3$) และเป็นภาพที่ต่างจาก CMYK และ Monochrome ภาพ 24 บิต (ภาพ 32 บิต) (8 บิต) ซอฟแวร์ที่ใช้ตกแต่งภาพ เช่น Photoshop จะเป็นซอฟแวร์ที่สามารถจัดการกับภาพ 24 บิต 8 บิต และ 32 บิต

2.4 รูปแบบของภาพ

บิตแมป (Bitmap) หรือ Raster จะประกอบไปด้วยพิกเซลที่เรียงต่อเนื่องกัน ภายในแต่ละพิกเซลจะมีองค์ประกอบที่ใช้ในการแสดงสี รูปทรง รูปแบบไฟล์ เราเรียกองค์ประกอบของพิกเซลนี้ว่า บิต (Bit) ภาพบิตแมปนั้นเราอาจเรียกว่า ราสเตอร์ (Raster) หรือเพนต์ไทป์ (Paint-type) ก็ได้ สำหรับข้อดีของบิตแมป คือ จะมีความเร็วในการแสดงสูงและใช้หน่วยความจำน้อยกว่า ตัวอย่างโปรแกรมกราฟิกประเภทบิตแมป เช่น Adobe Photoshop , Fractal Design Painter , Paint Shop Pro, L-View เป็นต้น

รูปที่ 2.16 ภาพที่ถูกสร้างแบบบิตแมป

เวกเตอร์ (Vector) ภาพประกอบเวกเตอร์นั้นจะใช้สมการทางคณิตศาสตร์ เป็นตัวสร้างภาพ โดยรวมเอกออปเจ็คหลายๆ ชนิดมาผสมกันเป็นรูปต่างๆ กันได้มากมาย ซึ่งภาพเวกเตอร์จะมีความละเอียดในการแสดงสูงมาก ไม่ว่าเราจะย่อหรือขยาย จะไม่ทำให้ภาพเพี้ยนไปได้ แต่การแสดงผลจะช้ามาก ตัวอย่างโปรแกรมกราฟิกประเภทเวกเตอร์ เช่น Adobe Illustrator, Macromedia Freehand, Corel Draw เป็นต้น

รูปที่ 2.17 ภาพที่ถูกสร้างด้วยเวกเตอร์

2.5 การทำงานของสแกนเนอร์ (Scanner)

สแกนเนอร์ มีหลักการทำงาน คือ เครื่องอ่านภาพจะทำการอ่านภาพโดยอาศัยการสะท้อนหรือการส่องผ่านของแสงกับ ภาพต้นฉบับที่ทึบแสง หรือโปร่งแสง ให้กระทบกับ แถบของอุปกรณ์ไวแสง (Photosensitive) ซึ่งมีชื่อในทางเทคนิคว่า Charge-Couple Device (CCD) ตัว CCD จะรับแสงดังกล่าวลงไปเก็บไว้ในเส้นเล็กของเซลล์ และจะแปลงคลื่นแสง ของแต่ละเซลล์เล็กๆ ให้กลายเป็นคลื่นความต่างศักย์ ซึ่งจะแตกต่างไปตามอัตราส่วนของระดับความเข้มของแสงแต่ละจุด

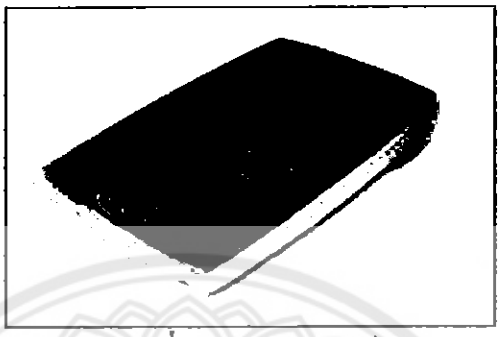
ตัวแปลงสัญญาณอนาล็อกเป็นดิจิตอล(Analog to Digital Converter) จะแปลงคลื่นความต่างศักย์ให้เป็นข้อมูลในรูปแบบที่คอมพิวเตอร์เข้าใจ ขณะเดียวกันโปรแกรมในการอ่านจะควบคุมการทำงานของเครื่องอ่านภาพให้รับข้อมูลเข้ามา และจัดเก็บรูปแบบเป็นแฟ้มข้อมูลของภาพระบบคอมพิวเตอร์ต่อไป

ภาพจากการสแกน

ภาพในคอมพิวเตอร์จะอยู่ในรูปแบบดิจิตอลคอมพิวเตอร์แทนส่วนเล็กๆ ของภาพที่เรียกว่า พิกเซล(Pixels) ขนาดของไฟล์ภาพ จะประกอบด้วยจำนวนพิกเซลเป็นร้อยเป็นพันคอมพิวเตอร์จะบันทึกค่าความเข้มและค่าสีของพิกเซลแต่ละพิกเซลด้วยจำนวน 1 บิตหรือหลายๆบิต จำนวนของพิกเซลจะเป็นตัวแสดงถึงความละเอียดและถ้ามีจำนวนบิตต่อพิกเซลมาก สีที่ได้ก็จะมากตามไปด้วย

รูปแบบการเก็บข้อมูลมีหลายระบบเช่น 1 บิต, 8 บิต และ 24 บิต โดยถ้าเป็นข้อมูลแบบ 1 บิต จะใช้สำหรับเก็บข้อมูลต่อพิกเซล 2 สถานะ คือ 1 และ 0 ซึ่งจะแสดงสีได้เฉพาะขาวกับดำ แต่ถ้า

เป็น 8 บิตจะใช้ความแตกต่างของสีถึง 256 ระดับ การรวมแม่สีมีเทคนิคที่เรียกว่า Dithering ซึ่งจะแสดงสีได้ไม่เหมือนกับความจริงที่เรามองเห็นได้ สำหรับระบบ 24 บิตจะให้ภาพที่มีสีใกล้เคียงจริงมากที่สุด เรียกว่า Photo-Realistic โดยจะแบ่ง 24 บิตเป็น 3 ส่วนคือ แดง, เขียว, น้ำเงิน ส่วนละ 8 บิต เมื่อรวมทั้ง 3 ส่วนเข้ากันแล้ว จะสามารถแสดงสีได้ถึง 16.7 ล้านสี



รูปที่ 2.18 สแกนเนอร์

การอ่านภาพสี CCD ของเครื่องอ่านภาพจะมีการประมวลผล โดยอาศัยโครงสร้างของแม่สี 3 สีคือ แดง, เขียว และน้ำเงิน ในทางเทคนิคจะเรียกว่า RGB ในโครงสร้างสีแบบ RGB นี้ แต่ละสีที่เกิดขึ้นจะประกอบด้วยแม่สีทั้ง 3 สีรวมอยู่ด้วยกันในค่าที่ต่างกันไป สีดำเกิดขึ้นจากการไม่มีแสงสี ขาว ในทำนองเดียวกัน สีขาวก็เกิดจากแสงแม่สีทั้ง 3 อยู่ในระดับสูงสุดเท่าๆกัน (100 เปอร์เซ็นต์ของ RGB) และระดับแสงเท่าๆกันของทั้ง 3 แม่สีจะเกิดแสงสีเทา (Gray-scale)

2.6 กระบวนการวัดทางสถิติ (Statistical Operations)

ที่ได้กล่าวมาแล้วเกี่ยวกับการรับภาพเข้าสู่คอมพิวเตอร์และรูปแบบของการนำภาพที่อยู่ในหน่วยความจำไปแสดงผลด้วยอุปกรณ์ต่างๆ จะเห็นได้ว่ามีกระบวนการทำได้หลาย ๆ วิธี เพื่อที่ จะให้ได้ผลตามที่ต้องการ

ในการแบ่งกระบวนการประมวลผลเกี่ยวกับภาพแบ่งได้หลักๆ 3 อย่างด้วยกันคือ ระดับต่ำ, ระดับปานกลาง และระดับสูง โดยขึ้นอยู่กับการกระทำเกี่ยวกับบิตของภาพ ซึ่งอาจต้องผ่านการประมวลระดับต่ำ, ระดับปานกลาง และ ระดับสูง ตามลำดับเพื่อให้ได้ผลลัพธ์ออกมาดีที่สุด

การประมวลระดับต่ำ จะจัดการกับภาพแบบขาว-ดำ โดยปกติจะทำการสร้างภาพที่สองขึ้นมาโดยให้มีเฉพาะข้อมูลที่ต้องการ ส่วนไหนที่ไม่ต้องการจะทำการตัดออกไป

การประมวลระดับปานกลาง เป็นการประมวลเกี่ยวกับการบ่งบอกว่าภาพมีลักษณะรูปร่าง พื้นที่ หรือจุดของภาพจากการประมวลผลแบบระดับต่ำ

การประมวลระดับสูง เป็นการรู้ลักษณะทั่วๆ ไปที่จำเป็นของภาพ เช่น มีการเชื่อมต่อกันของรูปร่างอะไรบ้าง เพื่อที่จะทำให้รู้รูปร่างที่แท้จริงของวัตถุ ผลของการประมวลผลในระดับนี้จะนำไปสู่การวิเคราะห์ภาพด้วย

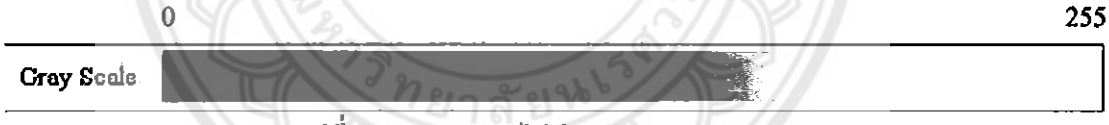
ในส่วนของนี้จะกล่าวถึงการประมวลผลภาพในระดับอย่างต่ำ ซึ่งจะขึ้นอยู่กับตำแหน่งต่างของพิกเซล

2.6.1 การแปลงภาพสู่ระดับสีเทา (Grey-Level Transformations)

การแปลงภาพสีเป็นระดับเทาได้นั้นทำได้หลายวิธีเช่น การใช้สมการ 2.17 แปลงค่า RGB ให้เป็นค่าเฉลี่ยแล้วแทนลงไปในพิกเซลนั้นๆซึ่งจะได้จำนวนบิตที่ใช้แทนระดับความเข้มของภาพเท่ากับ 24 บิตเหมือนเดิม หรืออีกวิธีโดยการเปลี่ยนจากภาพสี RGB เป็นภาพ Gray Scale ซึ่งจำนวนบิตที่ใช้แทนระดับความเข้มของภาพจะเหลือ 8 บิต โดยจะมีการคูณด้วยค่าคงที่ไปที่แต่ละสีของ RGB ซึ่งค่าคงที่นั้นโดยความจริงแล้วอาจจะไม่ใช่ตัวเลขที่ตายตัวเสมอไปแต่โดยทั่วไปแล้วมักเป็นดังสมการที่ 2.17

$$I = 0.299 \times red + 0.587 \times green + 0.114 \times blue \qquad (2.17)$$

โดยที่ I คือค่าของความเข้มมีค่าอยู่ในช่วงตั้งแต่ 0 (สีดำ) ไปจนถึง 255 (สีขาว) แสดงได้ดังรูปที่ 2.19



รูปที่ 2.19 แสดงการไล่สีระดับเทา (Gray scale)

2.6.2 การทำ Thresholding

ใช้เพื่อ เปลี่ยนทิศทางของภาพ โดยทำการแยกกลุ่มของภาพออกเป็นส่วน ๆ

กระบวนการทำ โดยปกติ การทำ Threshold จำทำการเปรียบเทียบกับค่าของ threshold โดยถ้ามากกว่าจะเปลี่ยนค่าพิกเซลที่ตำแหน่งนั้นให้เป็นค่าสูงสุด และถ้าน้อยกว่าจะเปลี่ยนค่าพิกเซลตำแหน่งนั้นให้เป็นค่าต่ำสุด จึงทำให้ได้ภาพมีแค่สองระดับคือ สูงสุด หรือ ต่ำสุด

ห้องสมุดคณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร
ป.ร.
15009157 ๑๔๒๓
๒๕๔๙

ตัวอย่างการแปลงภาพระดับเทา เป็นภาพขาวดำ

ตารางที่ 2.1 ตารางแสดงค่าในภาพระดับเทา

47	230	170	237	71	219	124	30	53	113
66	111	40	170	74	145	132	44	78	187
199	190	141	224	160	103	170	241	198	196
162	195	192	15	111	160	2	97	224	162
88	152	210	175	229	62	155	157	22	54
216	223	117	196	27	6	250	74	1	83
190	231	88	99	39	28	195	197	145	7
118	206	95	252	114	52	54	136	215	13
1	145	88	183	97	72	152	91	92	117
228	4	28	253	245	207	28	173	210	39

จากตารางสามารถคำนวณค่า Threshold ได้เป็น 128 เมื่อนำไปเข้ากระบวนการ Thresholding จะได้ค่าออกมา ดังตาราง

ตารางที่ 2.2 ตารางแสดงค่าในภาพขาวดำหลังการทำ Thresholding

0	1	1	1	0	1	0	0	0	0
0	0	0	1	0	1	1	0	0	1
1	1	1	1	1	0	1	1	1	1
1	1	1	0	0	1	0	0	1	1
0	1	1	1	1	0	1	1	0	0
1	1	0	1	0	0	1	0	0	0
1	1	0	0	0	0	1	1	1	0
0	1	0	1	0	0	0	1	1	0
0	1	0	1	0	0	1	0	0	0
1	0	0	1	1	1	0	1	1	0

ความผิดพลาดจากการทำ Threshold

โดยปกติความผิดพลาดจะเกิดจากการผลกระทบจากพื้นหน้าหรือพื้นหลังของภาพโดยที่เมื่อทำการแยกระดับของสีเทาออกเป็นกลุ่ม ความผิดพลาดหลัก ๆ ที่พบได้

ชนิดที่ 1 ทุกพิกเซลไม่เข้ามาอยู่ตามกลุ่มที่ต้องการ

ชนิดที่ 2 บางพิกเซล ไม่ควรจะเข้ามาอยู่ในกลุ่มที่ไม่ต้องการ

กระบวนการ Threshold ถูกนำไปใช้ในการสร้างภาพขาวดำ โดยอาจมีกระบวนการต่างๆ เพิ่มเข้ามาด้วยเพื่อทำให้คุณภาพของภาพดียิ่งขึ้น

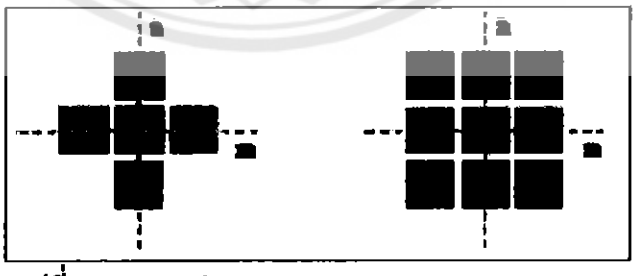
2.7 การประมวลผลภาพกับรูปร่างและโครงร่างของภาพ (Morphological Image Processing)

การประมวลผลภาพกับรูปร่างและโครงร่างของภาพ (Morphological Image Processing) เป็นเทคนิคพื้นฐานสำหรับการประมวลผลภาพที่เกี่ยวกับรูปร่างของวัตถุ และค่าภายในของแต่ละพิกเซลในรูปภาพ จะขึ้นอยู่กับการเปรียบเทียบระหว่างพิกเซลของภาพที่รับเข้า (input image) กับพิกเซลรอบข้างโดยในการทำงานกับรูปภาพนั้นจะเป็นภาพที่ถูกเก็บอยู่ในลักษณะของภาพไบนารี่ (Binary image) และนำภาพนั้นเข้ามาสู่กระบวนการคือสามารถสร้าง Morphology operation ในการทำงานเบื้องต้นนั้นจะต้องทำการเลือกขนาดและรูปร่างของพิกเซลรอบข้าง ซึ่งจะได้อธิบายเกี่ยวกับหลักการของ Morphology operation เบื้องต้น ในกระบวนการทางด้านการประมวลผลภาพ (Image processing) นั้นใช้กระบวนการพื้นฐานทางคณิตศาสตร์ในการทำ Morphology นั้นจะประกอบด้วยการ intersection , union และ complement โดยกระบวนการนั้นจะกระทำกับภาพที่มีคุณลักษณะเป็นภาพไบนารี่ (Binary image) ตามที่กล่าวไปข้างต้นจำนวน 2 ภาพด้วยกัน โดยภาพหนึ่งจะทำหน้าที่เป็น Structuring element หรือ Mask ซึ่งในกระบวนการทำงานจะประกอบด้วย Terminology พื้นฐานดังนี้

- **Foreground** ในภาพไบนารี่ (Binary image) จะประกอบไปด้วยพิกเซลที่มีค่าเท่ากับ 1 ซึ่งใช้แทนส่วนของวัตถุใน Image โดยจะเห็นเป็นสีขาว

- **Background** ใน ภาพไบนารี่ (Binary image) จะประกอบไปด้วยพิกเซลที่มีค่าเท่ากับ 0 ซึ่งใช้แทนพื้นหลังของ Image เมื่อดูภาพนั้นจะเห็นพื้นหลังเป็นสีดำ

- **Connectivity** นิยามที่บอกว่าวัตถุที่อยู่ในภาพนั้นเชื่อมต่อกัน ซึ่งอาจจะเชื่อมต่อกันด้วยลักษณะแบบ 4-connected หรือ 8-connected



รูปที่ 2.20 แสดงจำนวน 4-connected และ 8-connected

- **Morphology** นั้นจะเป็นการศึกษาเกี่ยวกับรูปร่างของวัตถุในภาพโดยการใช้ Structuring element ประมวลผลร่วมกับ Input image ได้ Output image ที่มีขนาดเดียวกัน Operation ที่สำคัญของ Morphology ได้แก่ Dilation และ Erosion

- **Neighborhood** คือ กลุ่มของพิกเซลในภาพซึ่งมีความสัมพันธ์กับพิกเซลในตำแหน่งที่กำลังประมวลผลหรือที่สนใจอยู่ สามารถกำหนดได้โดย structuring element หรือ Connectivity

- **Object** คือ กลุ่มของพิกเซลในภาพที่เชื่อมต่อกันเป็นรูปร่างเป็นวัตถุหรือก็คือ Connected component นั่นเอง

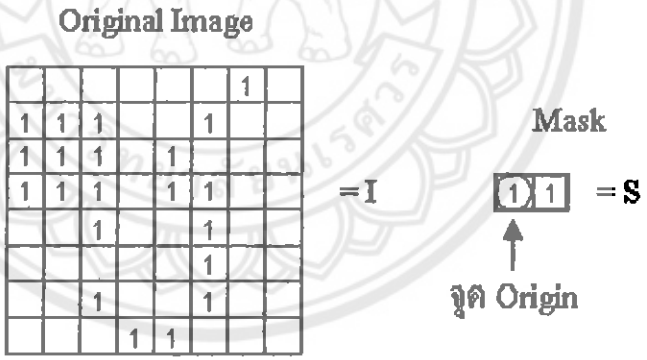
- **Structuring element** คือ เมตริกซ์ที่ใช้กำหนดขนาดและรูปร่างของ Neighborhood ซึ่งใช้สำหรับ Morphology operation โดยที่สามารถมีขนาดและรูปร่างใดๆ ก็ได้

สำหรับ Operation ที่สำคัญของ Morphology ได้แก่

1. Dilation
2. Erosion
3. Opening
4. Closing

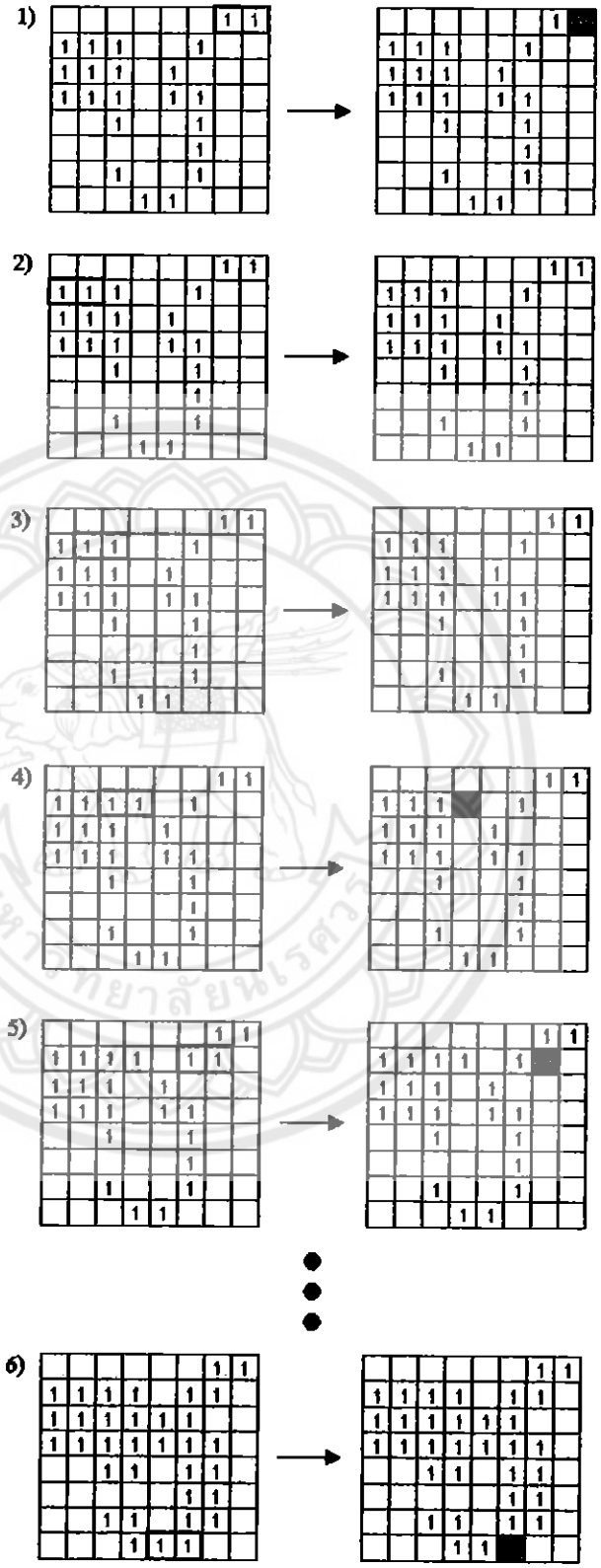
2.7.1 **การขยาย (Dilation)** คือ กระบวนการที่จะทำให้พื้นที่ของวัตถุมีขนาดขยายใหญ่ขึ้น ในการขยายภาพภาพไบนารี่ (Binary image) I โดยใช้ Mask S สามารถใช้สัญลักษณ์แทนได้ดังนี้

$$I \oplus S = \bigcup_{s \in S} I_s \tag{2.18}$$

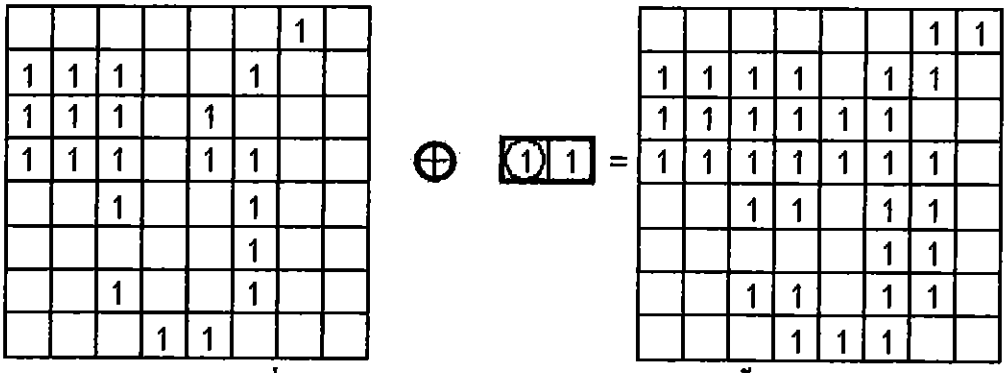


รูปที่ 2.21 ตัวอย่างสำหรับตัวดำเนินการแบบขยาย (Dilation)

เริ่มจากแถวบนคอลัมน์ซ้ายสุด วางจุด Origin ไว้ที่พิกเซลที่มีค่าเป็น 1 พิกเซลเพื่อนบ้านใดๆ ที่ตรงกับ Mask ให้เปลี่ยนค่าเป็น 1 ด้วย ทำไปเรื่อยๆ จนครบทุกพิกเซล



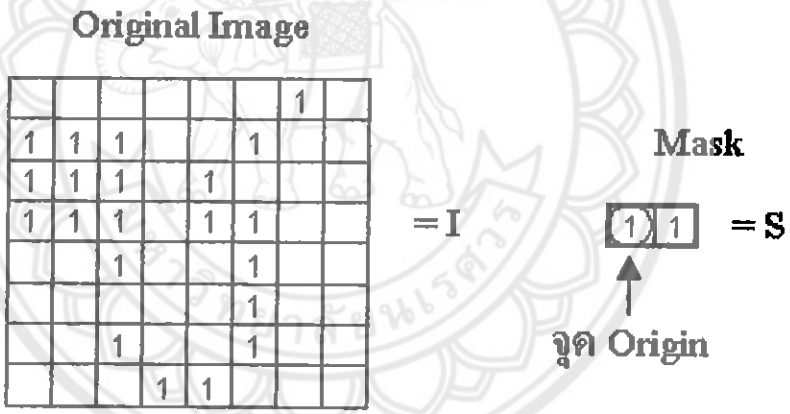
รูปที่ 2.22 แสดงการดำเนินการแบบขยาย (Dilation)



รูปที่ 2.23 ผลลัพธ์หลังจากกระบวนการเสร็จสิ้น

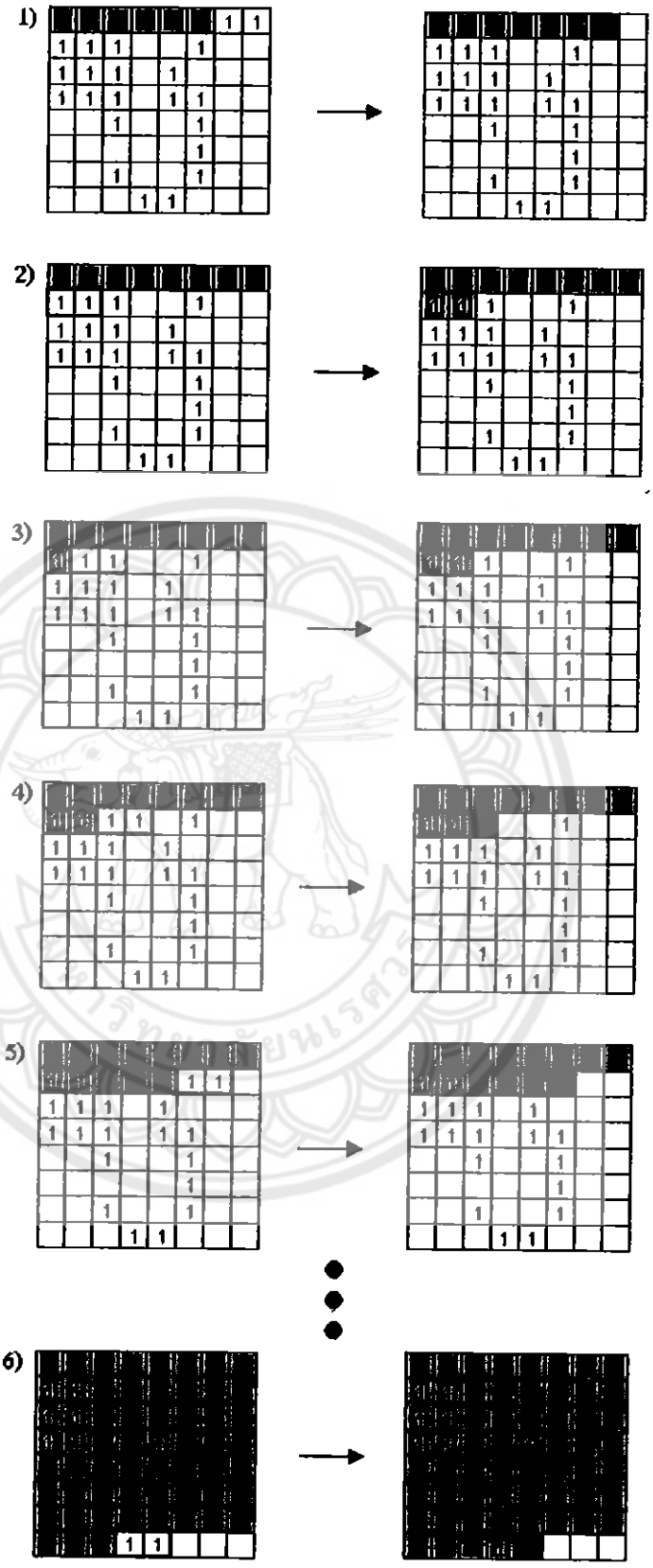
2.7.2 การหด (Erosion) คือ กระบวนการที่จะทำให้พื้นที่ของวัตถุมีขนาดเล็กลงการหดภาพภาพไบนารี่ (Binary image) I โดยใช้ Mask S สามารถใช้สัญลักษณ์แทนได้ดังนี้

$$I \ominus S = \{i \mid i + s \text{ in } I, \forall s \text{ in } S\} \tag{2.19}$$

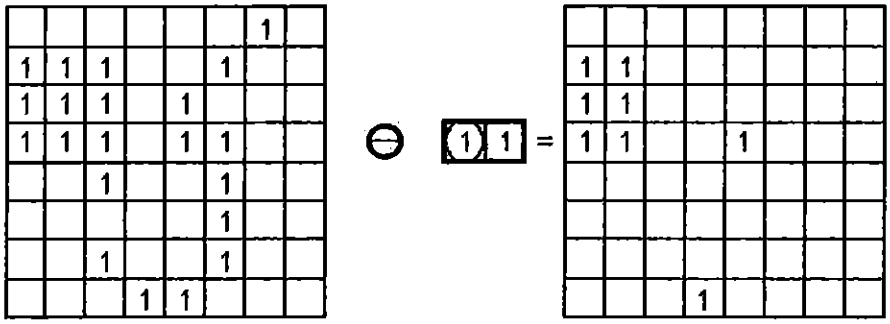


รูปที่ 2.24 ตัวอย่างสำหรับตัวดำเนินการแบบหด (Erosion)

เริ่มจากแถวบนคอลัมน์ซ้ายสุด วางจุด Origin ไว้ที่พิกเซลที่มีค่าเป็น 1 พิกเซลเพื่อนบ้านใดๆ ที่ไม่เหมือนกับ Mask ให้ลบทิ้งหรือเปลี่ยนค่าเป็น 0 ทำไปเรื่อยๆ จนครบทุกพิกเซล

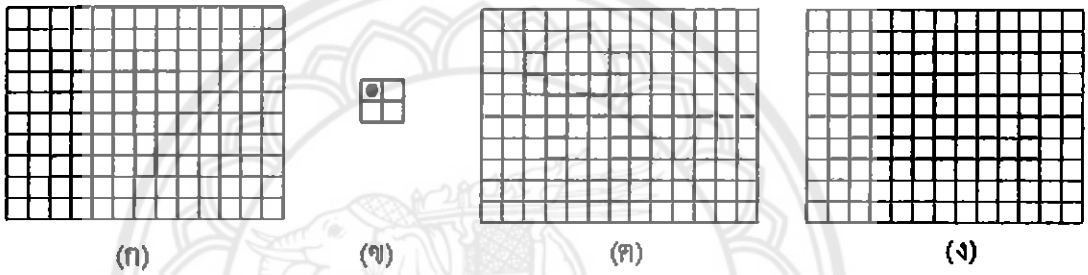


รูปที่2.25 แสดงการดำเนินการแบบหด (Erosion)



รูปที่ 2.26 ผลลัพธ์หลังจากกระบวนการเสร็จสิ้น

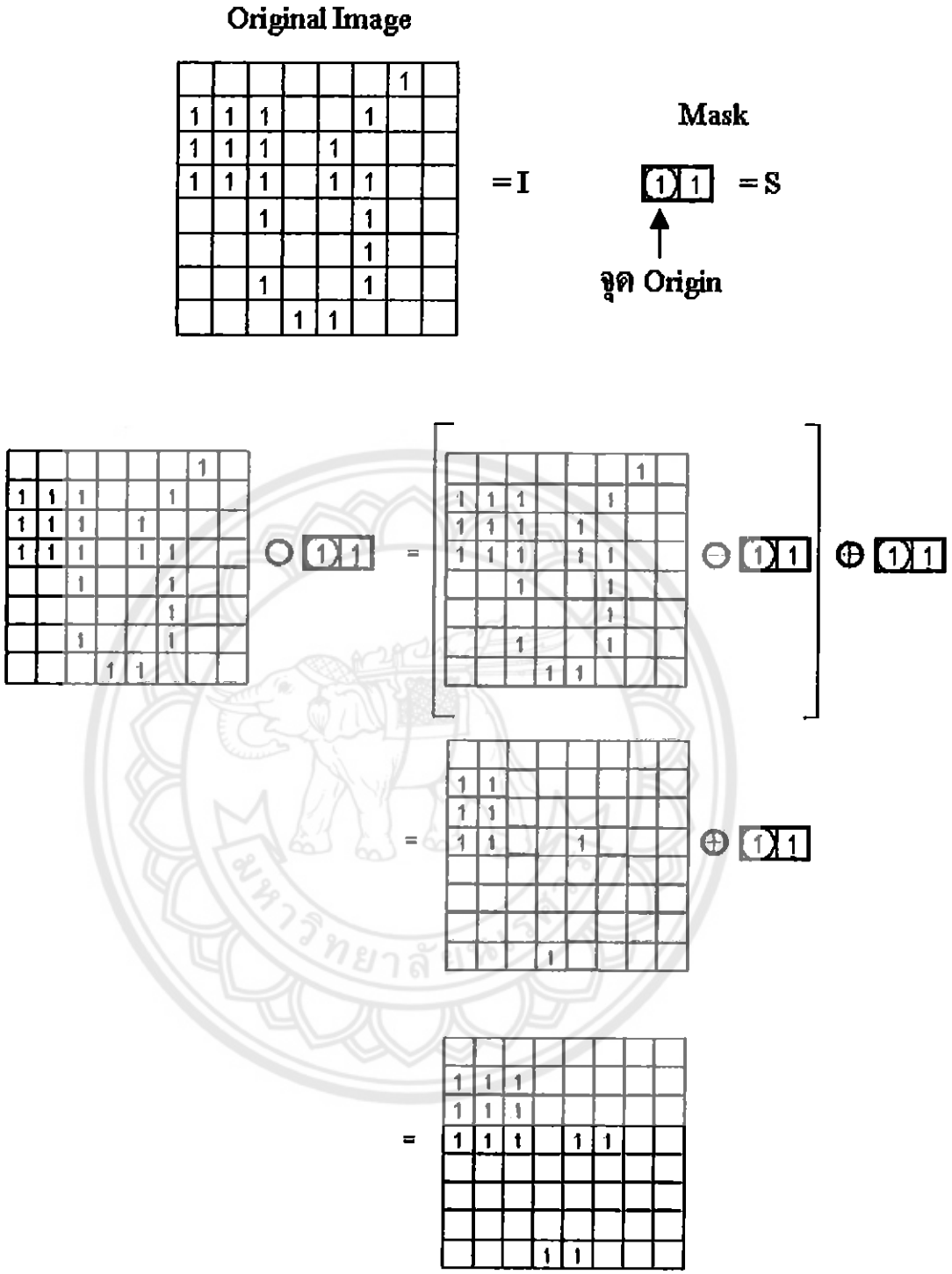
Operation ทั้งสองนี้ให้ผลลัพธ์ต่างกันตามรูปที่ 2.27



รูปที่ 2.27 ผลลัพธ์ของการทำ (ค) Erosion และ (ง) Dilation

2.7.3 การเปิด (Opening) คือการทำ Combination ระหว่าง Dilation กับ Erosion โดยจะทำ Erosion ก่อนแล้วจึงตามด้วย Dilation Operation กระบวนการกำจัดส่วนเล็กๆ ที่ยื่นออกมานอกขอบของวัตถุซึ่งการเปิดภาพภาพไบนารี่ (Binary image) I โดยใช้ Mask S เป็นการใช้ตัวดำเนินการแบบหด (Erosion) ก่อนแล้วจึงใช้ตัวดำเนินการแบบขยาย (Dilation) ซึ่งสามารถใช้สัญลักษณ์แทนได้ดังนี้

$$I \bigcirc S = (I \ominus S) \oplus S \tag{2.20}$$

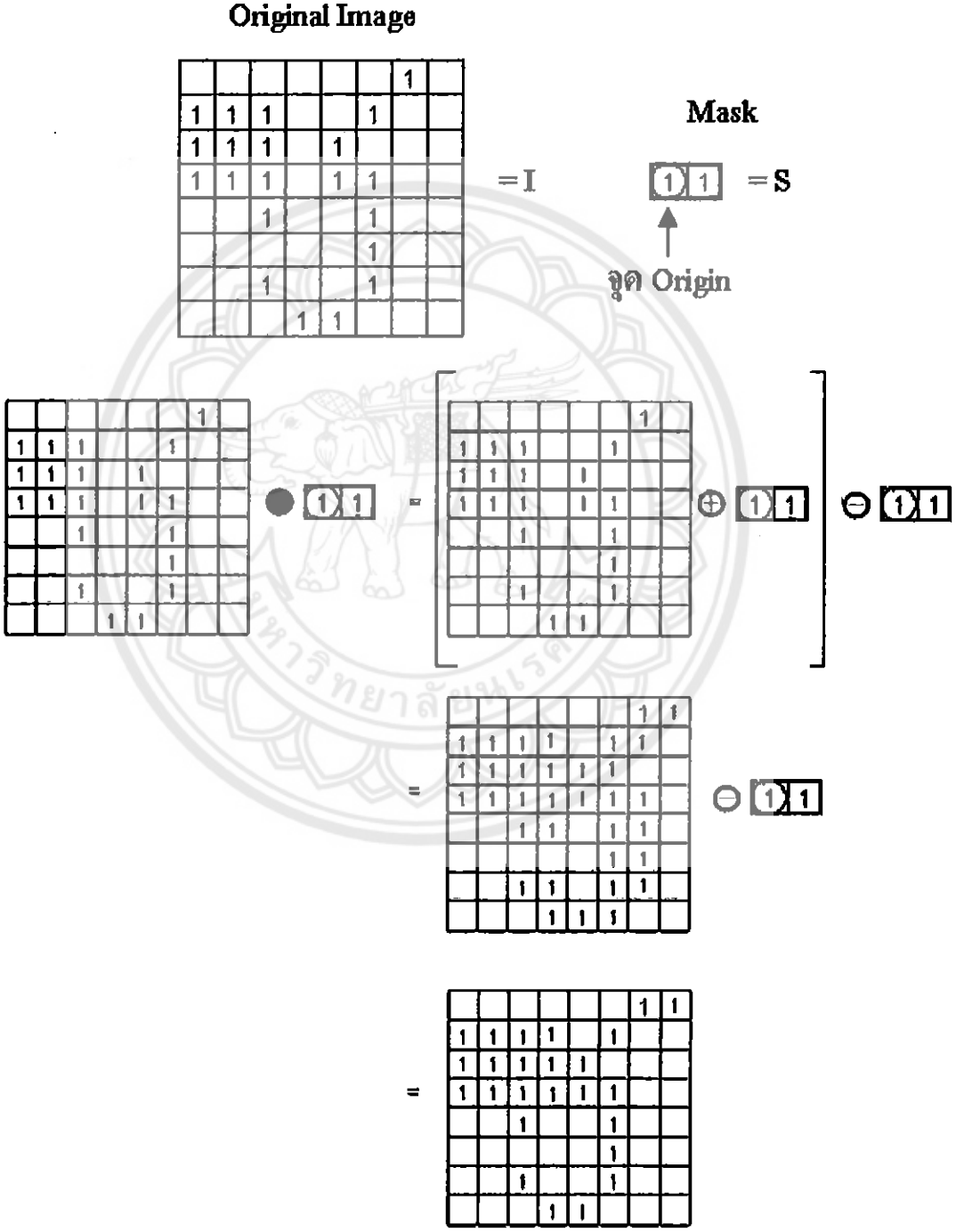


รูปที่ 2.28 ตัวอย่างสำหรับตัวดำเนินการแบบเปิด (Opening)

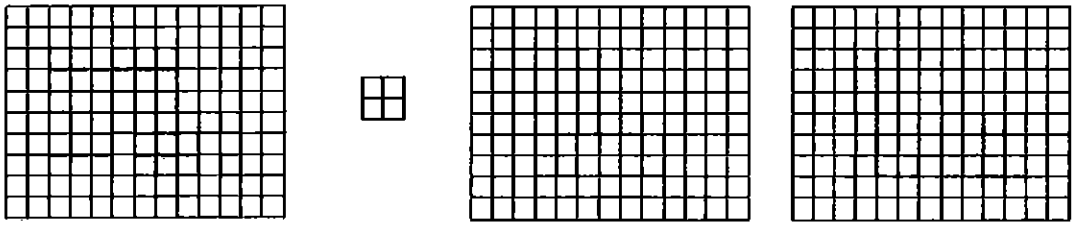
2.7.4 การปิด (Closing) คือ การทำ Combination ระหว่าง Dilation กับ Erosion โดยจะทำ Dilation ก่อนแล้วจึงตามด้วย Erosion Operation กระบวนการนี้จะทำการปิดรูหรือ noise ที่อยู่ภายในวัตถุและกำจัดส่วนเว้าตามขอบซึ่งการปิดภาพภาพไบนารี่ (Binary image) I โดยใช้ Mask S เป็น

การใช้ตัวดำเนินการแบบขยาย (Dilation) ก่อนแล้วจึงใช้ตัวดำเนินการแบบหด (Erosion) ซึ่งสามารถใช้สัญลักษณ์แทนได้ดังนี้

$$I \bullet S = (I \oplus S) \ominus S \tag{2.21}$$



รูปที่ 2.29 ตัวอย่างสำหรับตัวดำเนินการแบบปิด (Closing)



รูปที่ 2.30 ผลลัพธ์ของการทำ (ค)Opening และ (ง)Closing

2.8 การกรองข้อมูลภาพโดยใช้ค่ามัธยฐาน (Median filtering)

วิธีการนี้จะนำเอาความเข้มแสงของจุดที่ตรงกันในภาพต่างๆ มาเรียงลำดับ (sort) จากน้อยไปหามาก จากนั้นจะเลือกค่าที่อยู่ตรงกลางไปใช้ หากจำนวนภาพทั้งหมดเป็นจำนวนคู่ ค่าทั้งสองที่อยู่ตรงกลางจะนำมาหาค่าเฉลี่ย วิธีการนี้จะต้องใช้การเรียงลำดับซึ่งเป็นกระบวนการที่ใช้เวลาในการคำนวณสูง แต่ข้อดีคือไม่สูญเสียความคมชัด

ตัวอย่าง การใช้ median filter กับภาพระดับเทา

ตารางที่ 2.3 ตารางแสดงค่าในภาพระดับเทา

149	53	106	55	174	115	155	21	31	59
108	97	78	164	54	11	4	116	115	61
131	200	223	82	214	7	4	113	183	13
85	174	4	245	160	80	48	90	228	20
110	118	196	185	34	3	150	39	70	163
58	145	248	105	53	98	15	172	65	49
148	203	252	190	155	174	94	178	221	215
194	15	201	68	161	24	161	186	59	44
135	154	112	112	94	9	183	122	205	44
163	13	127	238	147	156	177	141	232	254

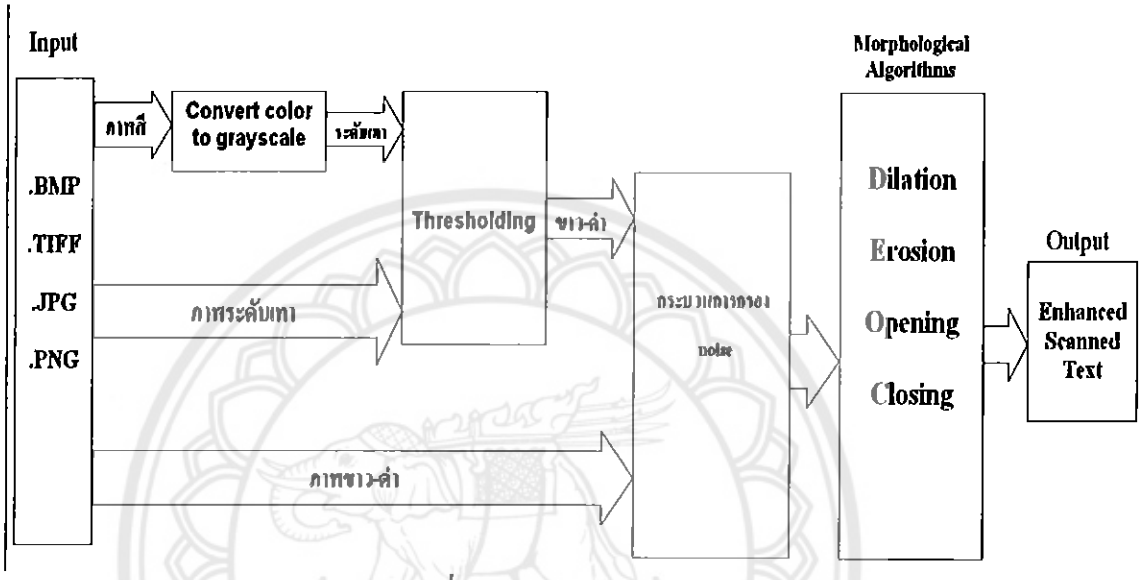
เมื่อใช้ median filter ขนาด 3x3 จะได้ผลลัพธ์ดังนี้

ตารางที่ 2.4 ตารางแสดงค่าในภาพระดับเทาที่ผ่าน median filter ขนาด 3x3

0	78	55	55	54	11	11	21	31	0
97	108	97	106	82	54	21	113	61	31
97	108	164	160	82	48	48	113	113	20
110	131	185	185	82	48	48	90	90	20
85	118	174	160	98	53	80	70	70	49
110	148	190	185	105	94	98	94	163	65
58	194	190	161	105	98	161	161	172	49
135	154	154	155	112	155	161	178	178	44
15	135	112	127	112	156	156	177	141	44
0	112	112	112	94	94	122	141	122	0

บทที่ 3
วิธีดำเนินงาน

3.1 บล๊อคไดอะแกรม



รูปที่ 3.1 บล๊อคไดอะแกรม

3.2 การเตรียมข้อมูลภาพสำหรับการปรับปรุงตัวอักษรที่ได้จากการสแกน

3.2.1 ทำการสแกนภาพหรือเอกสารที่เป็นข้อความ

นำภาพหรือเอกสารมาทำการสแกนผ่านเครื่องสแกน โดยให้มีอัตราการขยายของภาพเป็น 100% โดยจัดเก็บเป็นภาพสี ภาพระดับเทา หรือภาพขาวดำ และ มีนามสกุลเป็น .BMP, .JPEG, .PNG หรือ .TIFF

3.2.2 การแปลงภาพสี RGB ให้เป็น ภาพระดับเทา (Gray scale)

1) ทำการอ่านไฟล์ภาพสีโดยใช้คำสั่ง imread

2) ทำการแปลงจากภาพสีเป็นภาพระดับเทา โดยใช้คำสั่ง rgb2gray

3.2.3 การแปลงภาพระดับเทา (Gray scale image) ให้เป็น ภาพขาว- ดำ (Binary image)

วิธีการคำนวณหา ค่า Threshold

คำนวณหาค่า Threshold โดยใช้ Automatic Thresholding โดยมีขั้นตอนการคำนวณ ดังนี้

1. เลือกค่า T เริ่มต้น โดยให้ T เป็นค่าเฉลี่ย หรือ ค่ากลาง ของทั้งภาพ

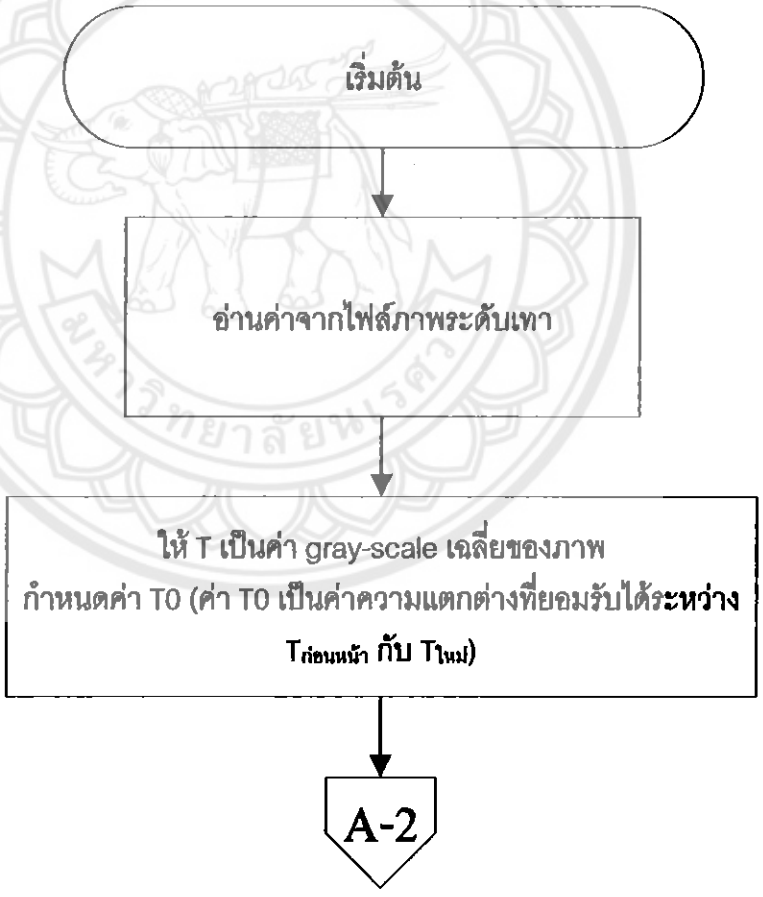
2. แยกค่า gray-level ของภาพออกเป็น สองกลุ่ม กลุ่มแรก(G_1) ให้เก็บค่า gray-level ที่น้อยกว่า หรือเท่ากับ ค่า T ส่วนกลุ่มที่ 2 (G_2) ให้เก็บค่า gray-level มากว่า ค่า T

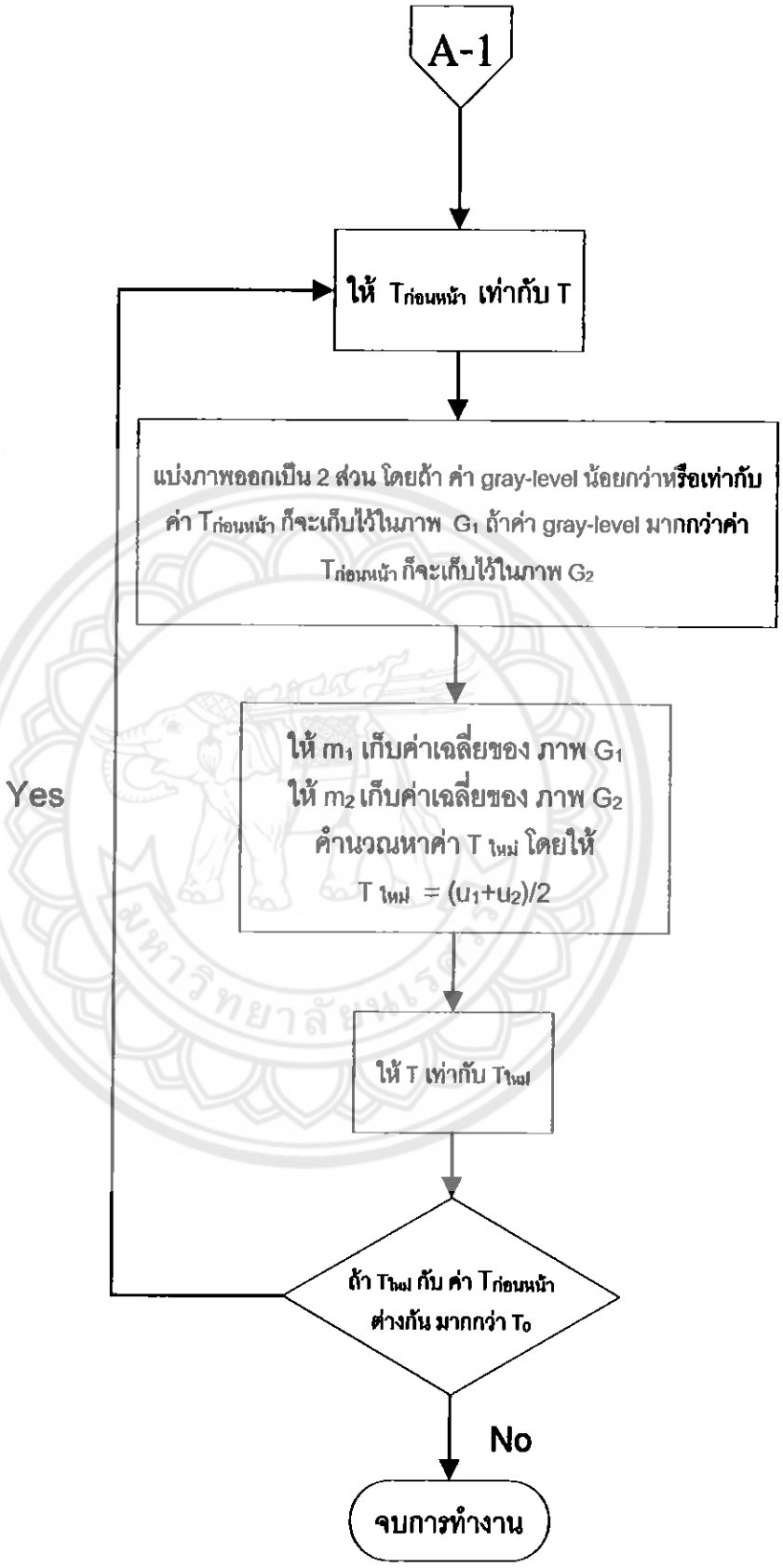
3. หา ค่าเฉลี่ยใน G_1 ให้เป็น μ_1 และ ค่าเฉลี่ยใน G_2 ให้เป็น μ_2

4. คำนวณหาค่า T ใหม่ โดยให้

$$T = \frac{(\mu_1 + \mu_2)}{2} \tag{3.1}$$

5. ทำซ้ำข้อ 2 โดยใช้ค่า T จากข้อ 4 เป็นตัวแยกค่า gray-level ทำซ้ำไปจนกว่าผลต่างระหว่างค่า T ที่ได้ กับค่า T ก่อนหน้า น้อยกว่า หรือเท่ากับ T_0 (ปกติ $T_0 = 0$)



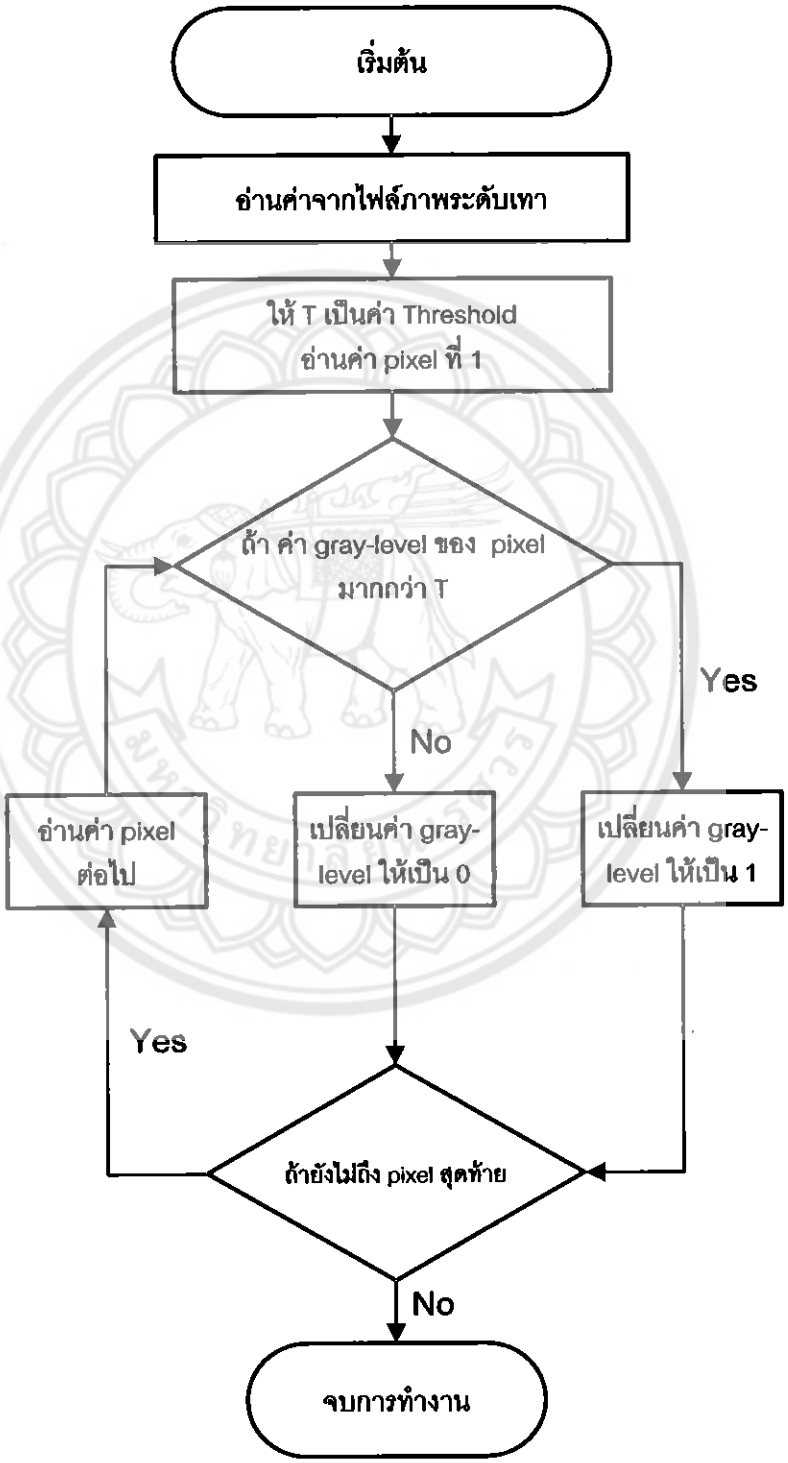


รูปที่ 3.2 แผนผังโปรแกรมของการหาค่า Threshold (T)

วิธีการแปลงภาพระดับเทา (Gray scale image) ให้เป็น ภาพขาว- ดำ (Binary image)

เมื่อได้ค่า Threshold มาแล้วนำมาใช้ในการแปลงภาพระดับเทาให้เป็นภาพขาว-ดำ โดย

1. ค่า gray-level ในภาพที่มีค่าน้อยกว่า ค่า Threshold เปลี่ยนให้มีค่าเป็น 0
2. ค่า gray-level ในภาพที่มีค่ามากกว่า ค่า Threshold เปลี่ยนให้มีค่าเป็น 1



รูปที่ 3.3 แผนผังโปรแกรมของการทำ Thresholding

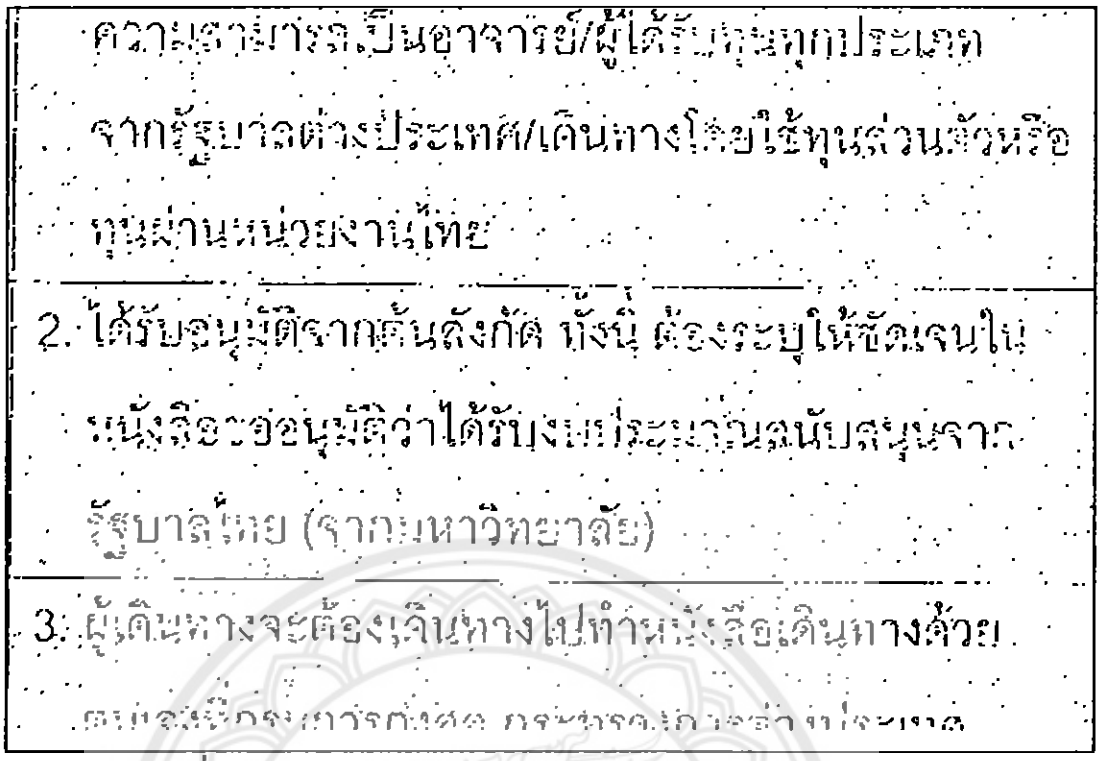
3.3 วิเคราะห์ลักษณะตัวอักษรในภาพ

วิเคราะห์ลักษณะปัญหาที่เกิดขึ้นกับตัวอักษรในภาพ และวิเคราะห์หาวิธีการปรับปรุงภาพที่ได้จะมีลักษณะปัญหาเด่นๆอยู่ 2 ลักษณะคือ

3.3.1 ลักษณะของตัวอักษรมีลักษณะบางหรือเลือนราง ซึ่งก็มีทั้งแบบมี noise และไม่มี noise

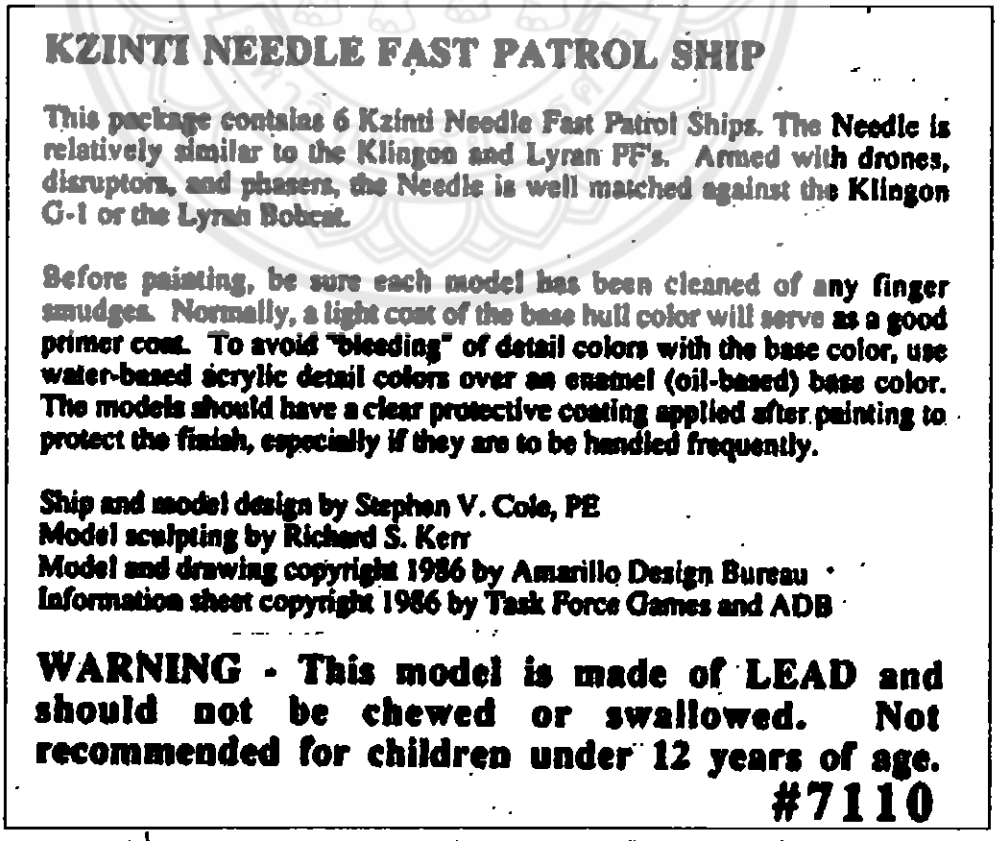
ความสามารถเป็นอาจารย์/ผู้ได้รับทุนทุกประเภท จากรัฐบาลต่างประเทศ/เดินทางโดยใช้ทุนส่วนตัวหรือ ทุนผ่านหน่วยงานไทย
2. ได้รับอนุมัติจากต้นสังกัด ทั้งนี้ ต้องระบุให้ชัดเจนใน หนังสือขออนุมัติว่าได้รับงบประมาณสนับสนุนจาก รัฐบาลไทย (จากมหาวิทยาลัย)
3. ผู้เดินทางจะต้องเดินทางไปทำหนังสือเดินทางด้วย

รูปที่ 3.4 ลักษณะของตัวอักษรมีลักษณะบางหรือเลือนรางแบบไม่มี noise

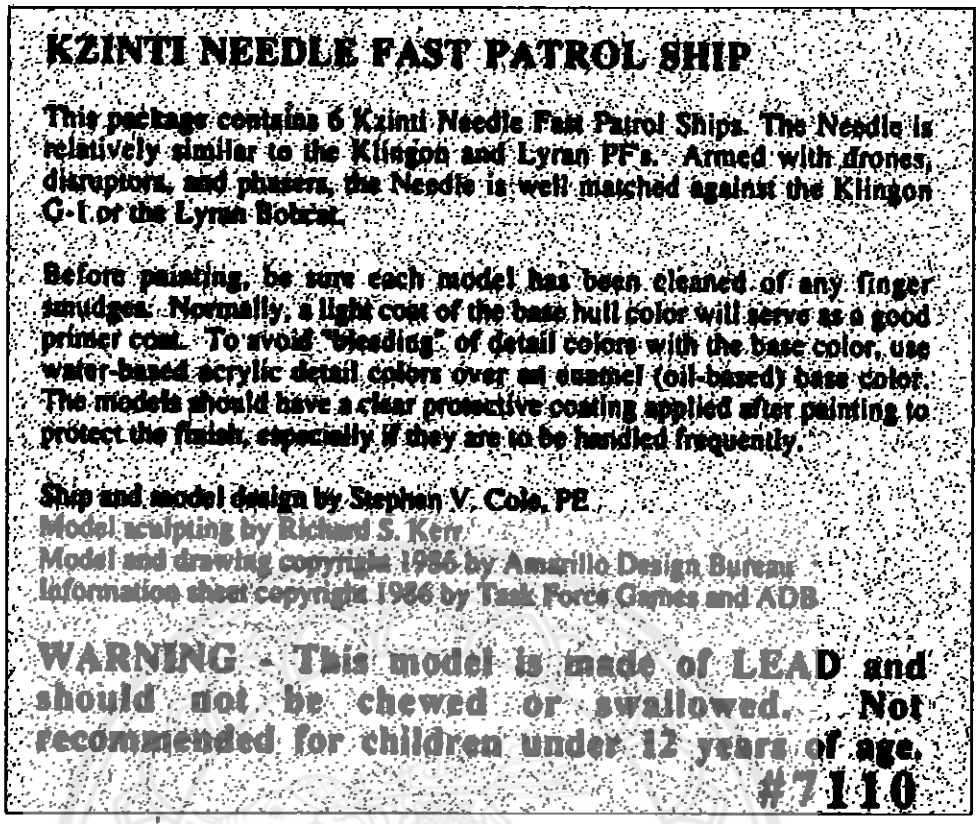


รูปที่ 3.5 ลักษณะของตัวอักษรมีลักษณะบางหรือเลือนรางแบบมี noise

3.3.2 ลักษณะของตัวอักษรมีลักษณะหนาหรือติดกัน ซึ่งก็มีทั้งแบบมี noise และไม่มี noise



รูปที่ 3.6 ลักษณะของตัวอักษรมีลักษณะหนาหรือติดกันแบบไม่มี noise


KZINTI NEEDLE FAST PATROL SHIP

This package contains 6 Kzinti Needle Fast Patrol Ships. The Needle is relatively similar to the Klingon and Lyran PF's. Armed with drones, disruptors, and phasers, the Needle is well matched against the Klingon G-1 or the Lyran Bobcat.

Before painting, be sure each model has been cleaned of any finger smudges. Normally, a light coat of the base hull color will serve as a good primer coat. To avoid "bleeding" of detail colors with the base color, use water-based acrylic detail colors over an enamel (oil-based) base color. The models should have a clear protective coating applied after painting to protect the finish, especially if they are to be handled frequently.

Ship and model design by Stephen V. Cole, PE.
Model sculpting by Richard S. Kerr
Model and drawing copyright 1986 by Amarillo Design Bureau
Information sheet copyright 1986 by Task Force Games and ADB

WARNING - This model is made of LEAD and should not be chewed or swallowed. Not recommended for children under 12 years of age.

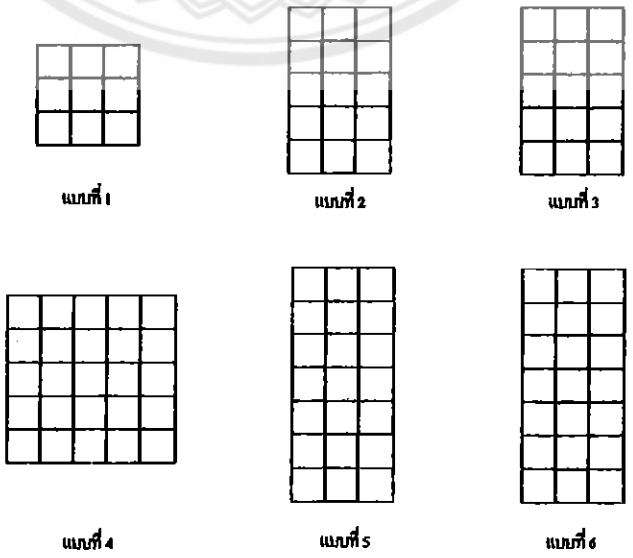
#7110

รูปที่ 3.7 ลักษณะของตัวอักษรมีลักษณะหนาหรือติดกันแบบมี noise

3.4 ออกแบบ Structure element โดยวิเคราะห์จากลักษณะตัวอักษร

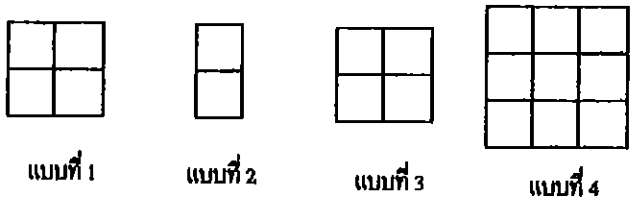
จากการสังเกตลักษณะของตัวอักษร ซึ่งมีลักษณะเส้นของตัวอักษรในแนวตั้งจะยาวกว่าในแนวนอน และเส้นของตัวอักษรขาดเป็นช่วงๆ โดยได้ Structure element ดังนี้

3.4.1 Structure Element สำหรับตัวอักษรที่มีลักษณะตัวอักษรบางหรือเลือนราง



รูปที่ 3.8 Structure Element สำหรับตัวอักษรที่มีลักษณะบางหรือเลือนราง

3.4.2 Structure Element สำหรับตัวอักษรที่มีลักษณะตัวอักษรหนาหรือติดกัน



รูปที่ 3.9 Structure Element สำหรับตัวอักษรที่มีลักษณะหนาหรือติดกัน

3.5 ออกแบบการทดลอง

3.5.1 ทดลองกำจัด noise สำหรับภาพที่มี noise โดยทดลองใช้ median filter และ opening

3.5.2 ทำการทดลองเพื่อดูผลของ Operation ทั้ง 4 ซึ่งได้แก่ dilation, erosion, closing และ opening กับแต่ละ Structure element ตามที่ออกแบบไว้

3.5.3 สังเกตผลจากการทดลอง ใช้ dilation, erosion, opening และ closing กับภาพที่นำมาทดสอบทั้ง 2 ลักษณะ โดยใช้ structure element ที่ได้ออกแบบไว้ แล้วเลือก operation ที่เหมาะสมกับภาพที่นำมาปรับปรุง

แบบที่ 1 ตัวอักษรที่มีลักษณะบาง ใช้วิธีดังนี้ในการปรับปรุง

- Dilation
- Closing
- Closing + Dilation

แบบที่ 2 ตัวอักษรที่มีลักษณะหนา ใช้วิธีการดังนี้ในการปรับปรุง

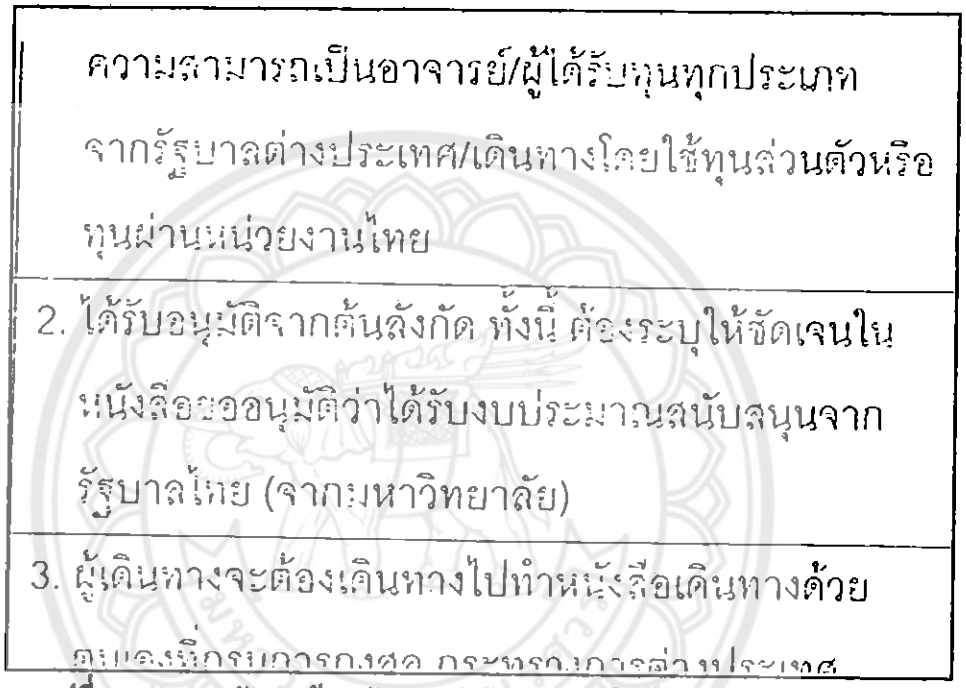
- Erosion
- Opening + Erosion

บทที่ 4

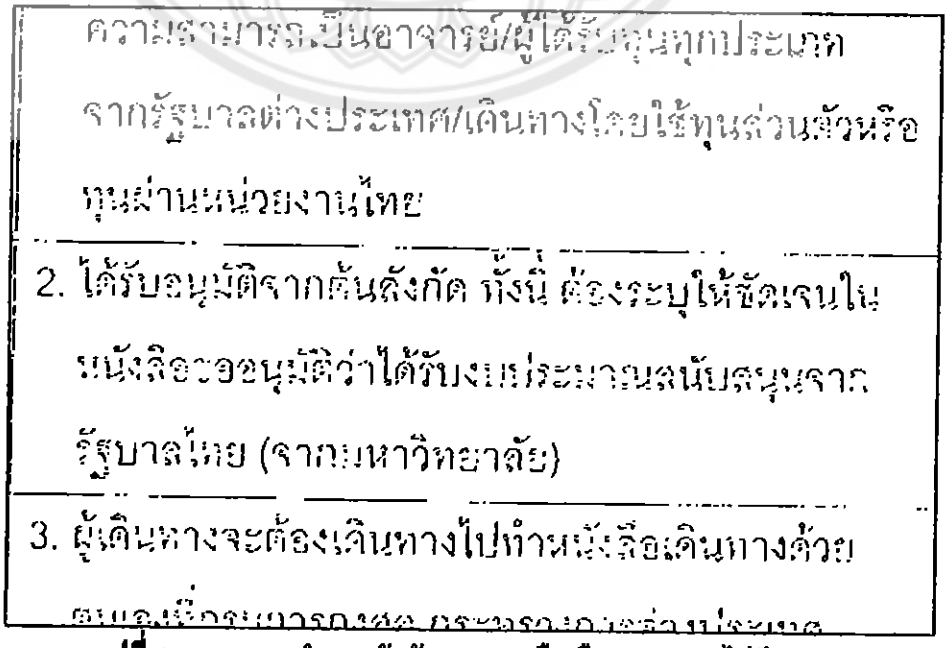
ผลการทดลองและวิเคราะห์ผลการทดลอง

4.1 ผลการทดลองสำหรับภาพที่มีลักษณะตัวอักษรบางหรือเลือนรางแบบไม่มี noise

ข้อมูลภาพที่นำมาใช้ในการทดลอง : ภาพระดับเทาขนาด 438x662 พิกเซล รูปแบบการจัดเก็บเป็น .jpg

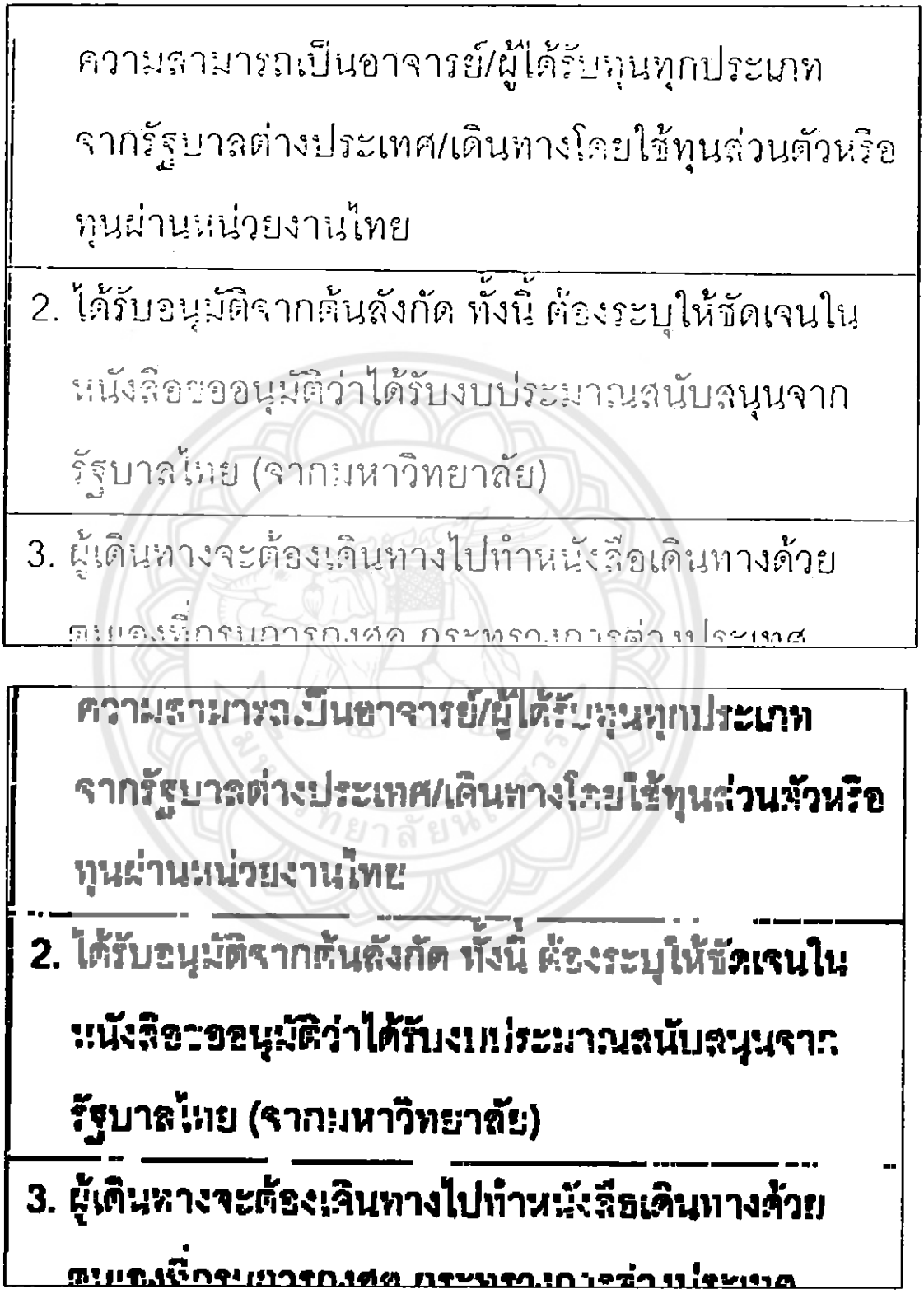


รูปที่ 4.1 ภาพระดับเทาต้นฉบับของตัวอักษรบางหรือเลือนรางแบบไม่มี Noise

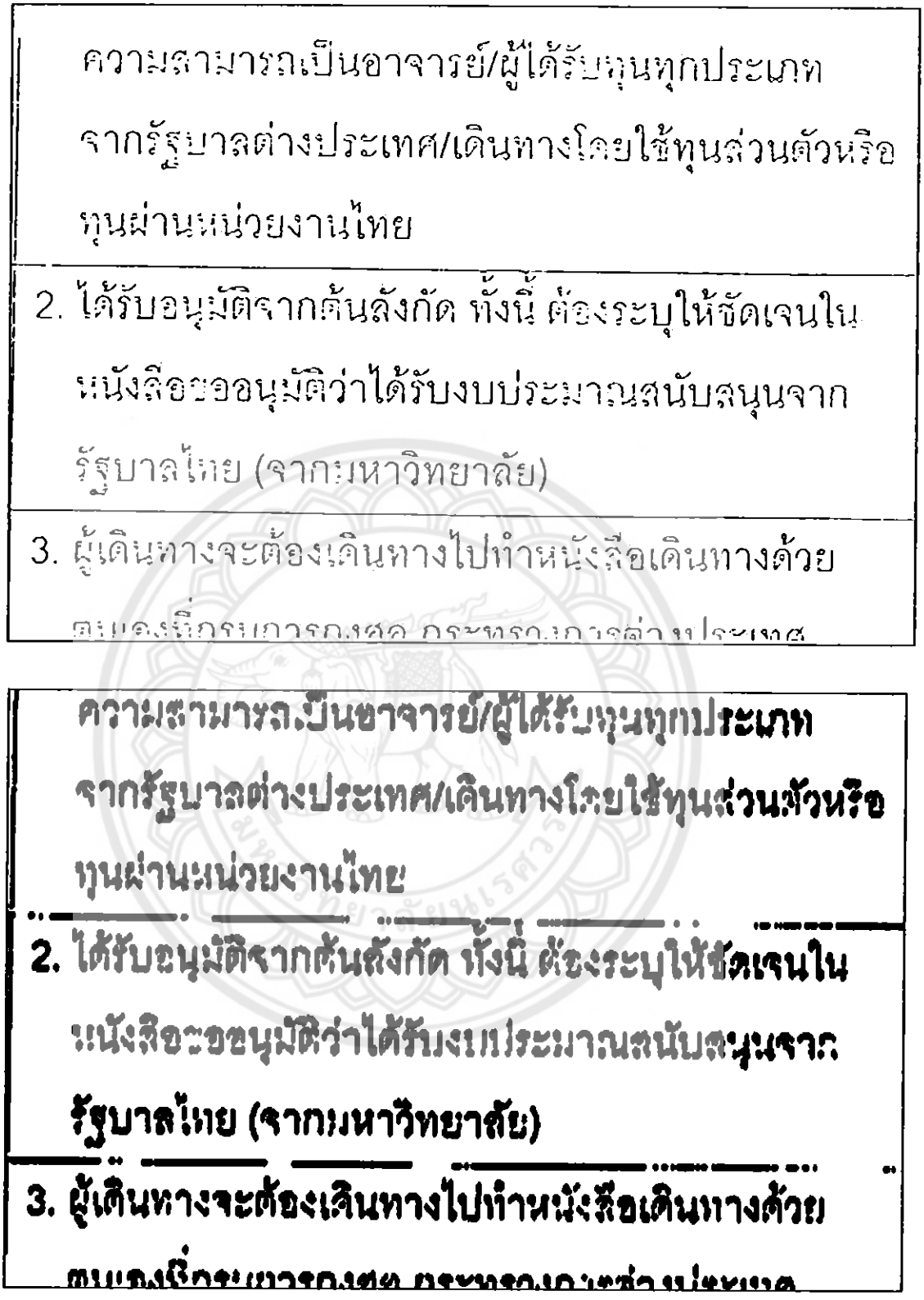


รูปที่ 4.2 ภาพขาวดำของตัวอักษรบางหรือเลือนรางแบบไม่มี Noise

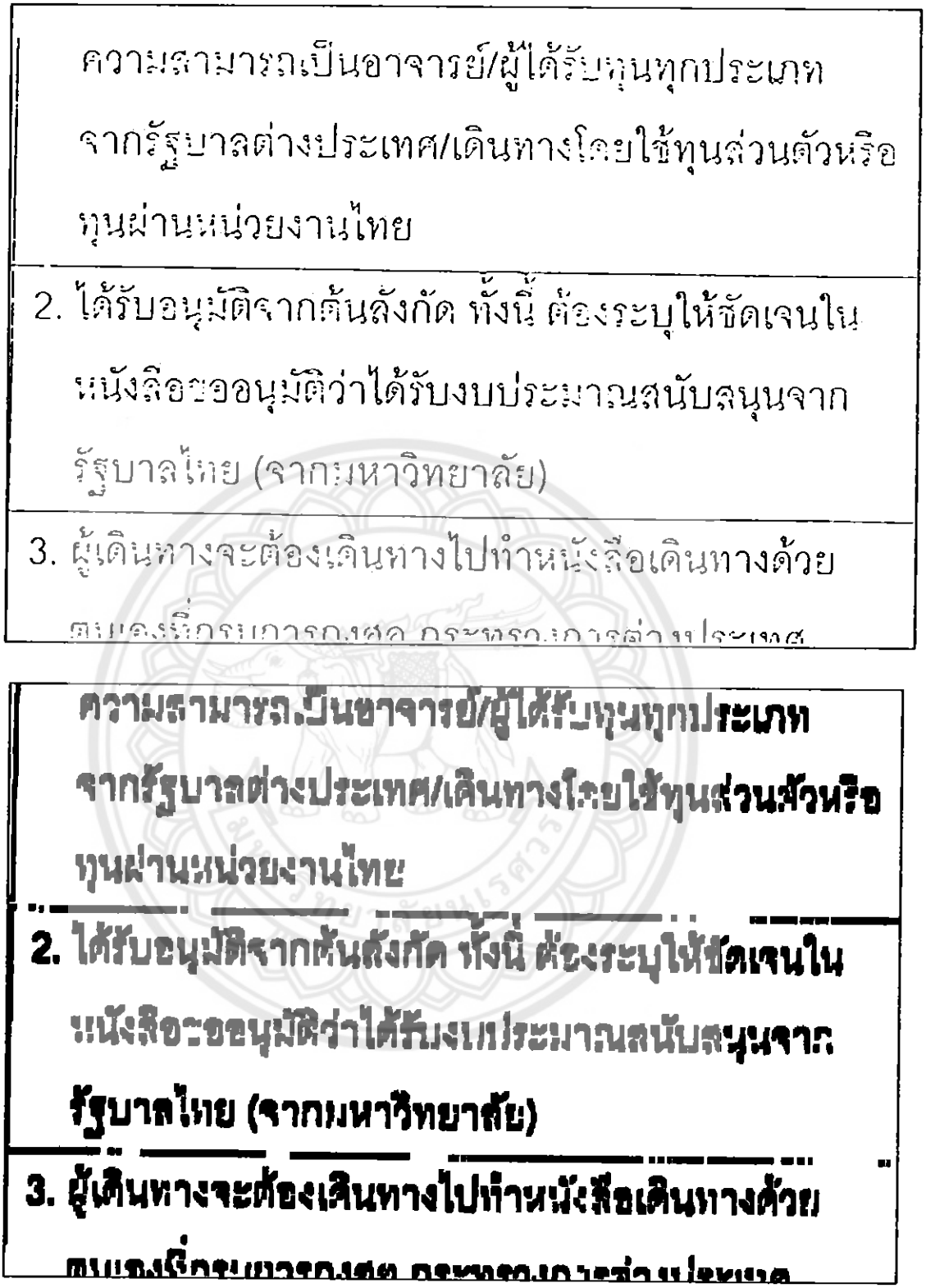
4.1.1 ผลลัพธ์จากการใช้ Dilation



รูปที่ 4.3 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบไม่มี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Dilation ด้วย Structure element (ล่าง)



รูปที่ 4.4 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบไม่มี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Dilation ด้วย Structure element (ล่าง)

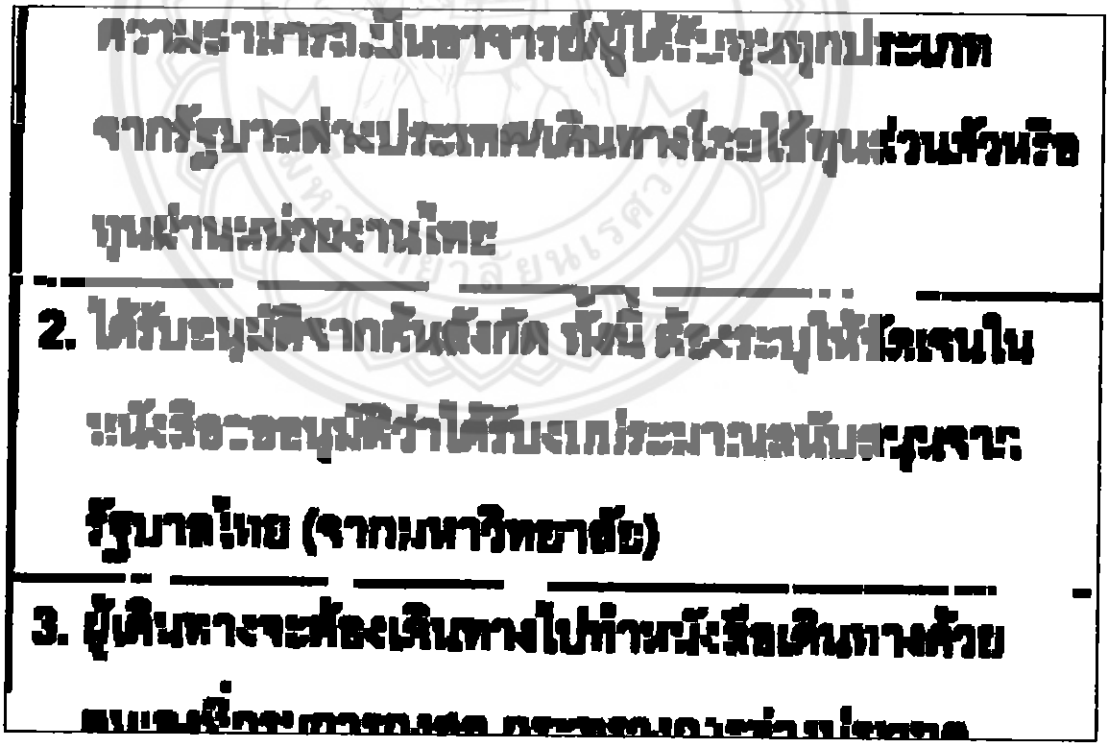


รูปที่ 4.5 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบไม่มี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Dilation ด้วย Structure element (ล่าง)

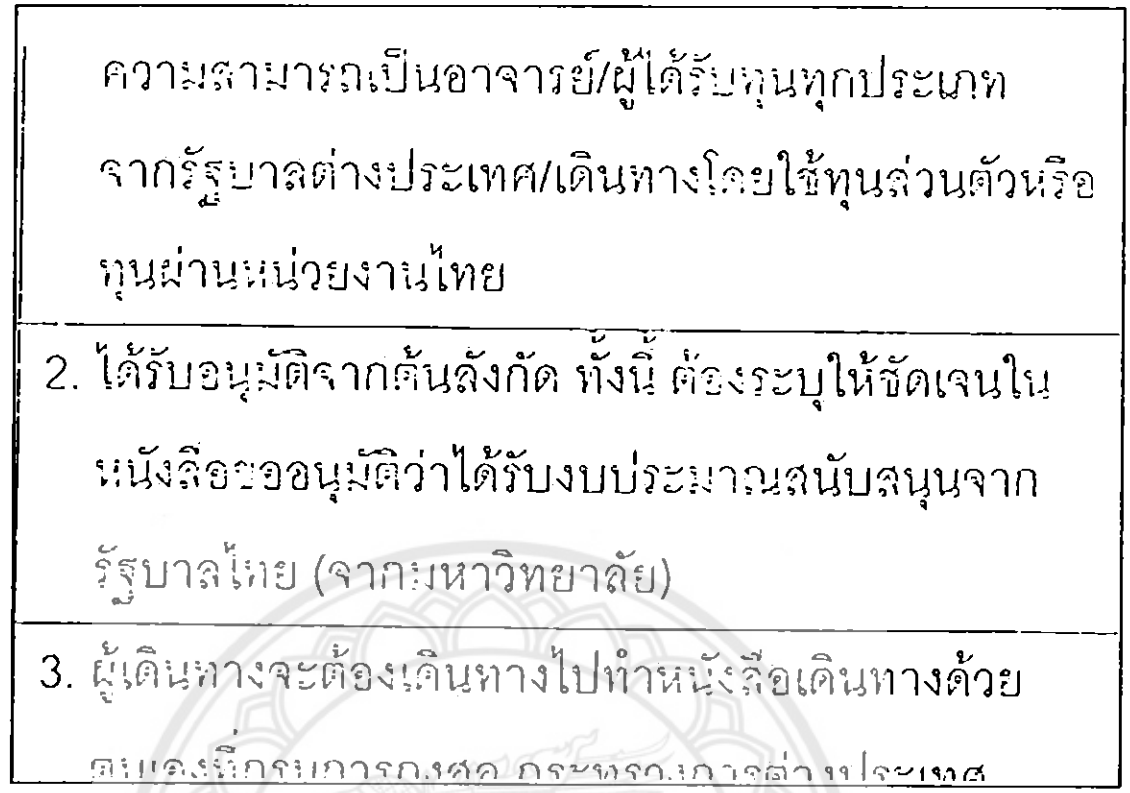
ความสามารถเป็นอาจารย์/ผู้ได้รับทุนทุกประเภท
จากรัฐบาลต่างประเทศ/เดินทางโดยใช้ทุนส่วนตัวหรือ
ทุนผ่านหน่วยงานไทย

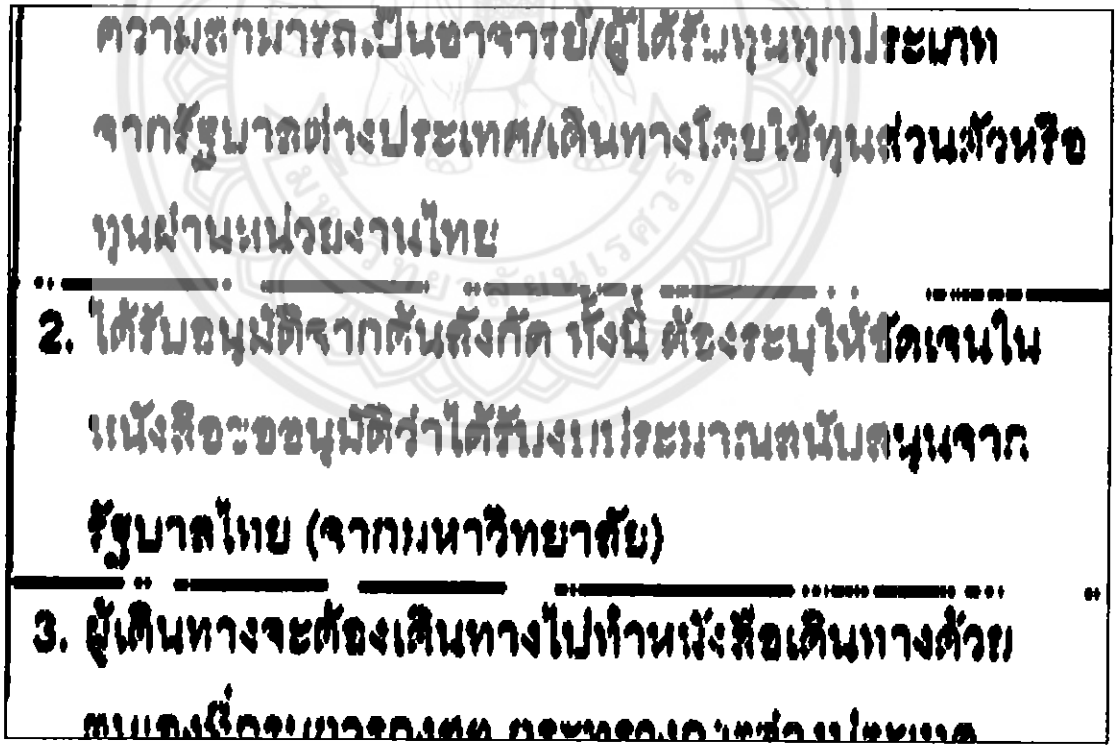
2. ได้รับอนุมัติจากต้นสังกัด ทั้งนี้ ต้องระบุให้ชัดเจนใน
หนังสืออนุมัติว่าได้รับงบประมาณสนับสนุนจาก
รัฐบาลไทย (จากมหาวิทยาลัย)

3. ผู้เดินทางจะต้องเดินทางไปทำหนังสือเดินทางด้วย
ตนเองที่กรมการกงสุล กระทรวงการต่างประเทศ

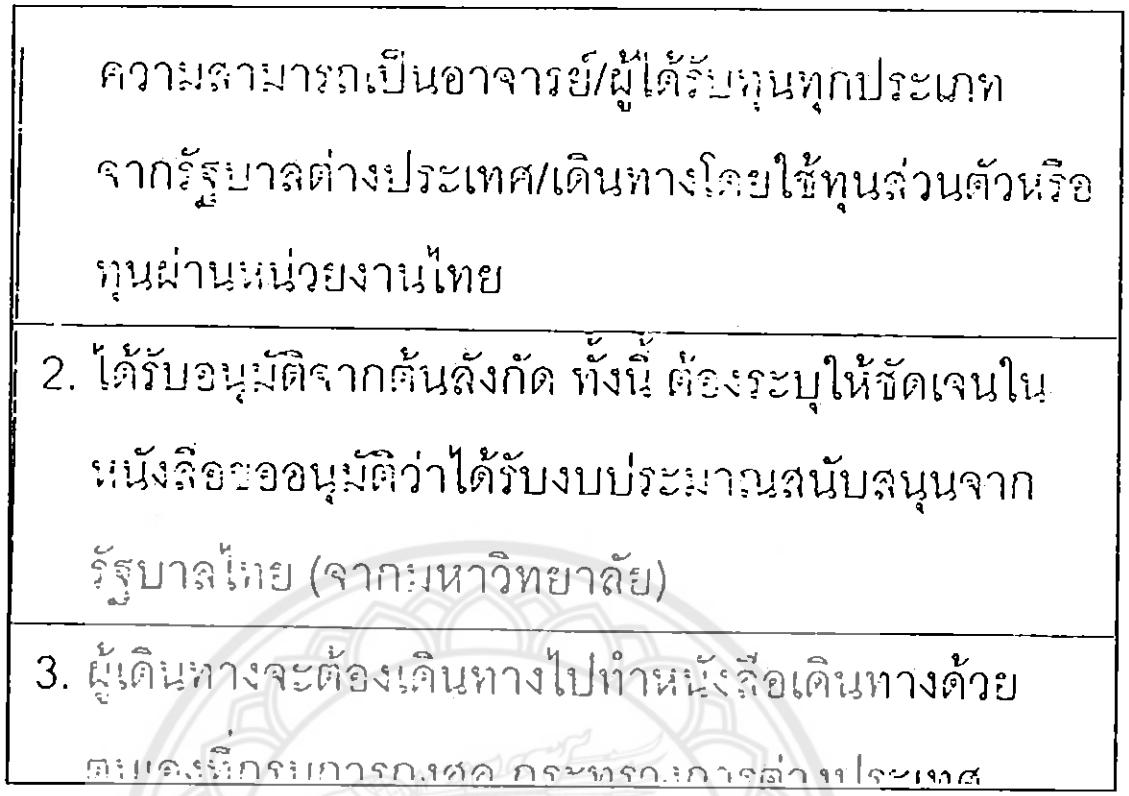


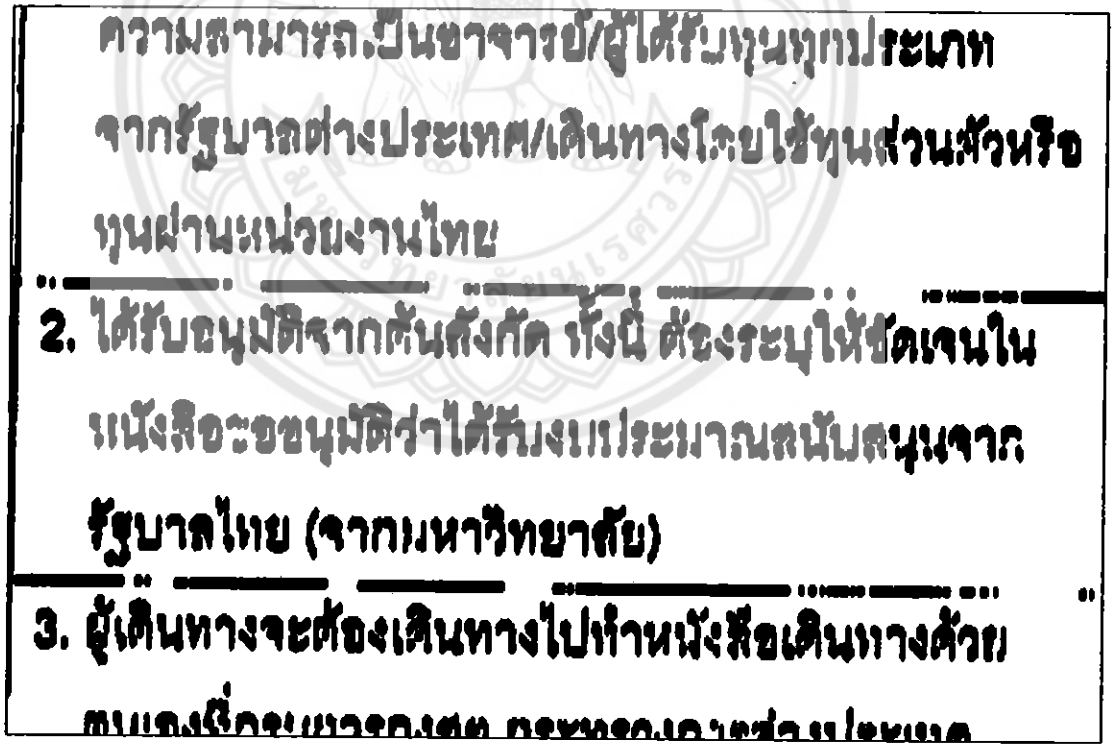
รูปที่ 4.6 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบไม่มี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Dilation ด้วย Structure element (ล่าง)





รูปที่ 4.7 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบไม่มี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Dilation ด้วย Structure element (ล่าง)





รูปที่ 4.8 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบไม่มี Noise (บน) เทียบกับ

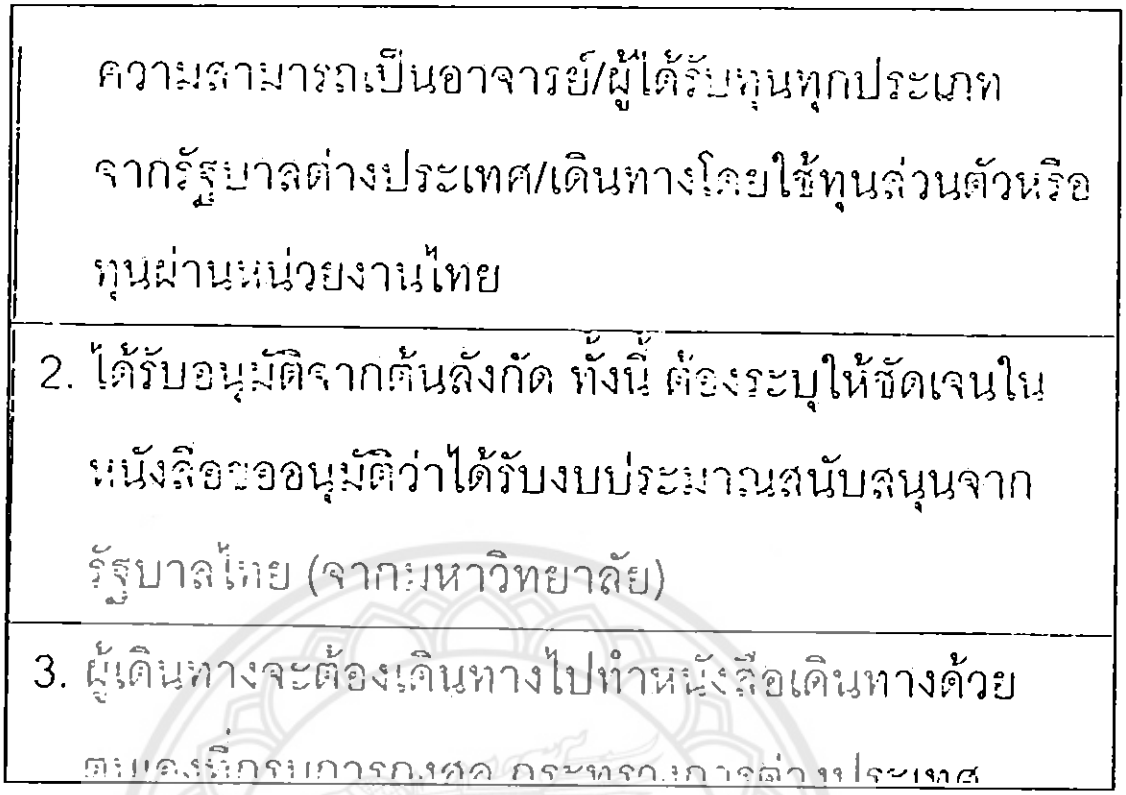
ลัพธ์จากการใช้ Dilation ด้วย Structure element (ล่าง)

4.1.2 ผลลัพธ์จากการใช้ Closing

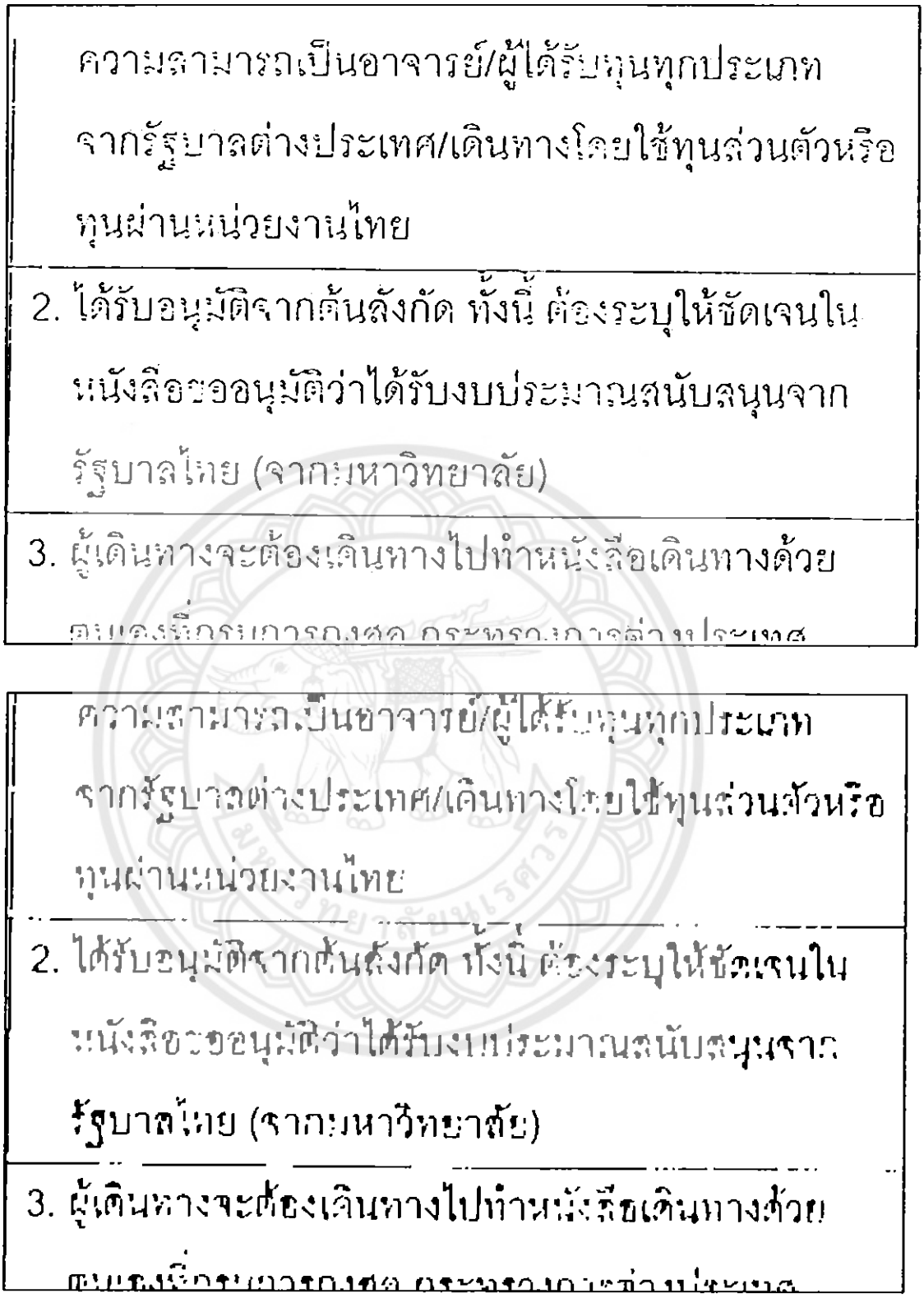
ความสามารถเป็นอาจารย์/ผู้ได้รับทุนทุกประเภท จากรัฐบาลต่างประเทศ/เดินทางโดยใช้ทุนส่วนตัวหรือ ทุนผ่านหน่วยงานไทย
2. ได้รับอนุมัติจากต้นสังกัด ทั้งนี้ ต้องระบุให้ชัดเจนใน หนังสือขออนุมัติว่าได้รับงบประมาณสนับสนุนจาก รัฐบาลไทย (จากมหาวิทยาลัย)
3. ผู้เดินทางจะต้องเดินทางไปทำหนังสือเดินทางด้วย ตนเองที่กรมการกงสุล กระทรวงการต่างประเทศ

ความสามารถเป็นอาจารย์/ผู้ได้รับทุนทุกประเภท จากรัฐบาลต่างประเทศ/เดินทางโดยใช้ทุนส่วนตัวหรือ ทุนผ่านหน่วยงานไทย
2. ได้รับอนุมัติจากต้นสังกัด ทั้งนี้ ต้องระบุให้ชัดเจนใน หนังสือขออนุมัติว่าได้รับงบประมาณสนับสนุนจาก รัฐบาลไทย (จากมหาวิทยาลัย)
3. ผู้เดินทางจะต้องเดินทางไปทำหนังสือเดินทางด้วย ตนเองที่กรมการกงสุล กระทรวงการต่างประเทศ

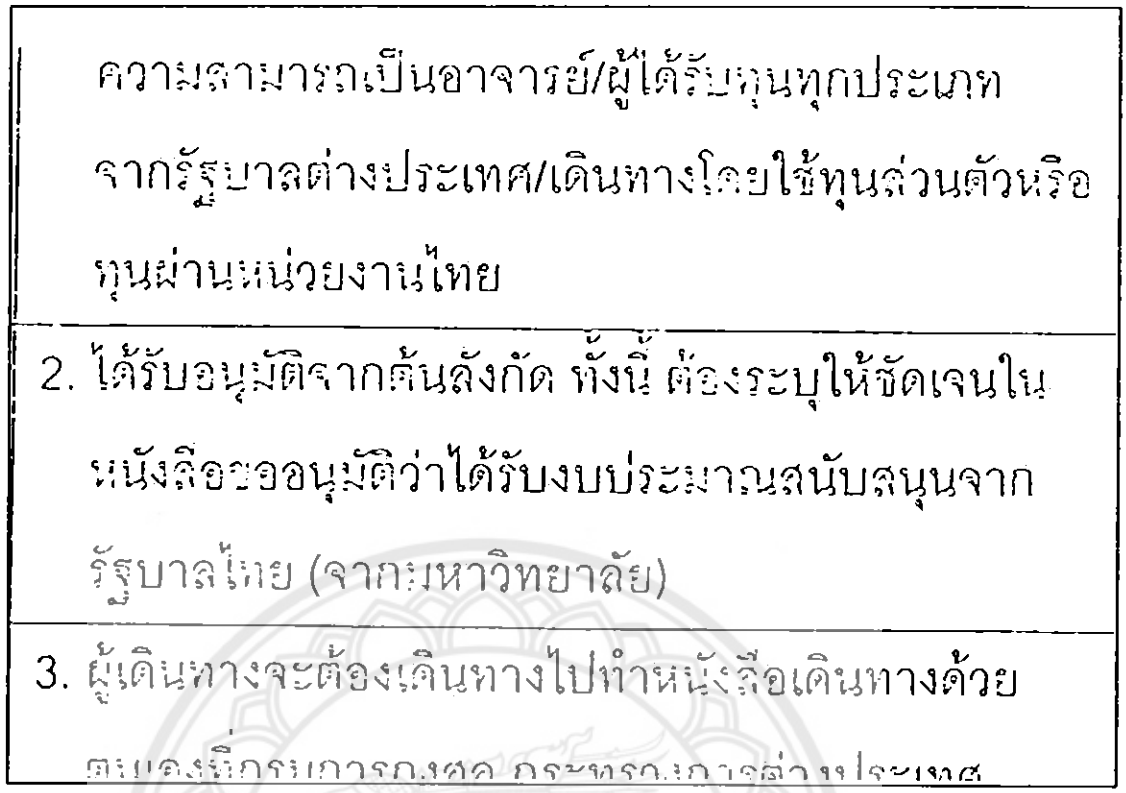
รูปที่ 4.9 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบไม่มี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Closing ด้วย Structure element ▦ (ล่าง)

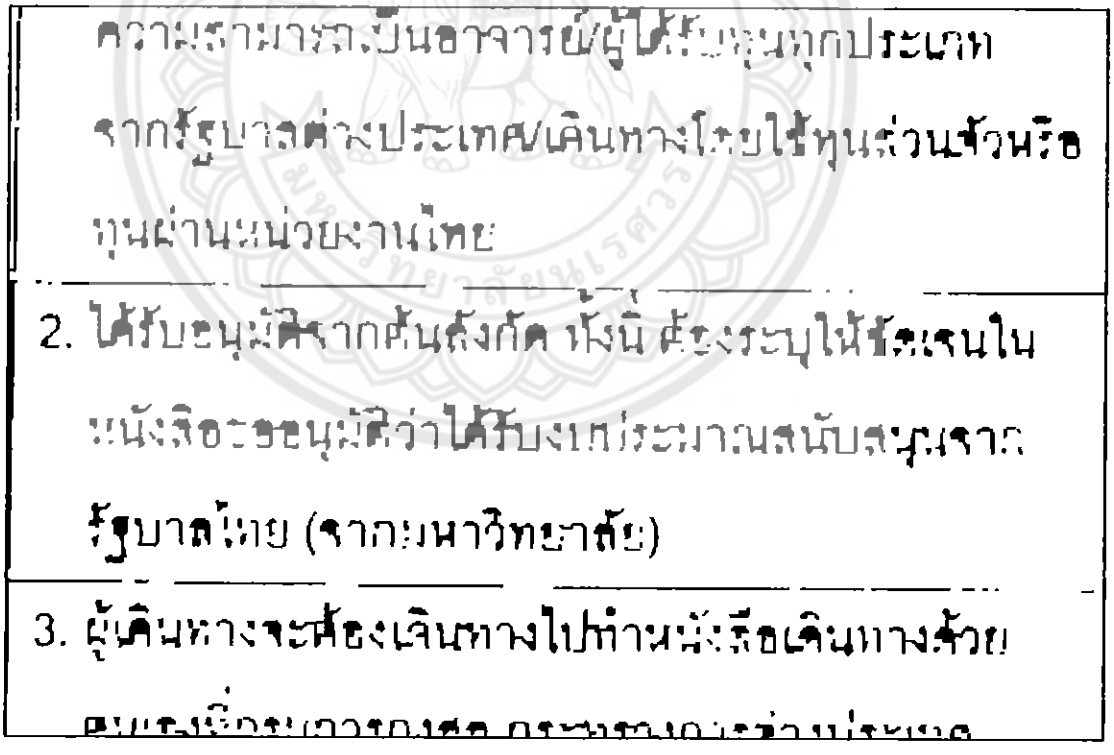


รูปที่ 4.10 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบไม่มี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Closing ด้วย Structure element (ล่าง)



รูปที่ 4.11 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบไม่มี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Closing ด้วย Structure element (ล่าง)



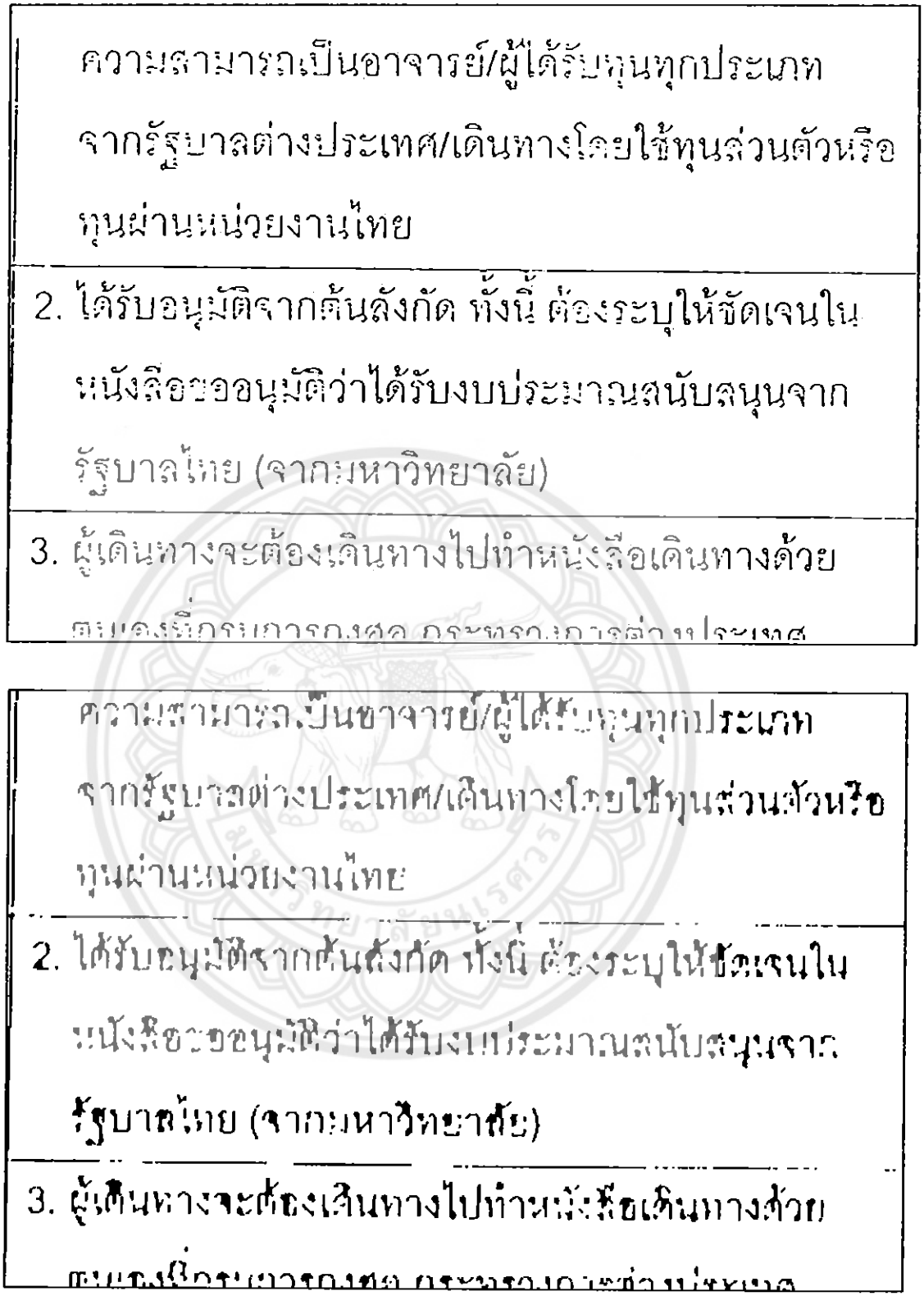


รูปที่ 4.12 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบไม่มี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Closing ด้วย Structure element (ล่าง)

ความสามารถเป็นอาจารย์/ผู้ได้รับทุนทุกประเภท จากรัฐบาลต่างประเทศ/เดินทางโดยใช้ทุนส่วนตัวหรือ ทุนผ่านหน่วยงานไทย
2. ได้รับอนุมัติจากต้นสังกัด ทั้งนี้ ต้องระบุให้ชัดเจนใน หนังสืออนุมัติว่าได้รับงบประมาณสนับสนุนจาก รัฐบาลไทย (จากมหาวิทยาลัย)
3. ผู้เดินทางจะต้องเดินทางไปทำหนังสือเดินทางด้วย

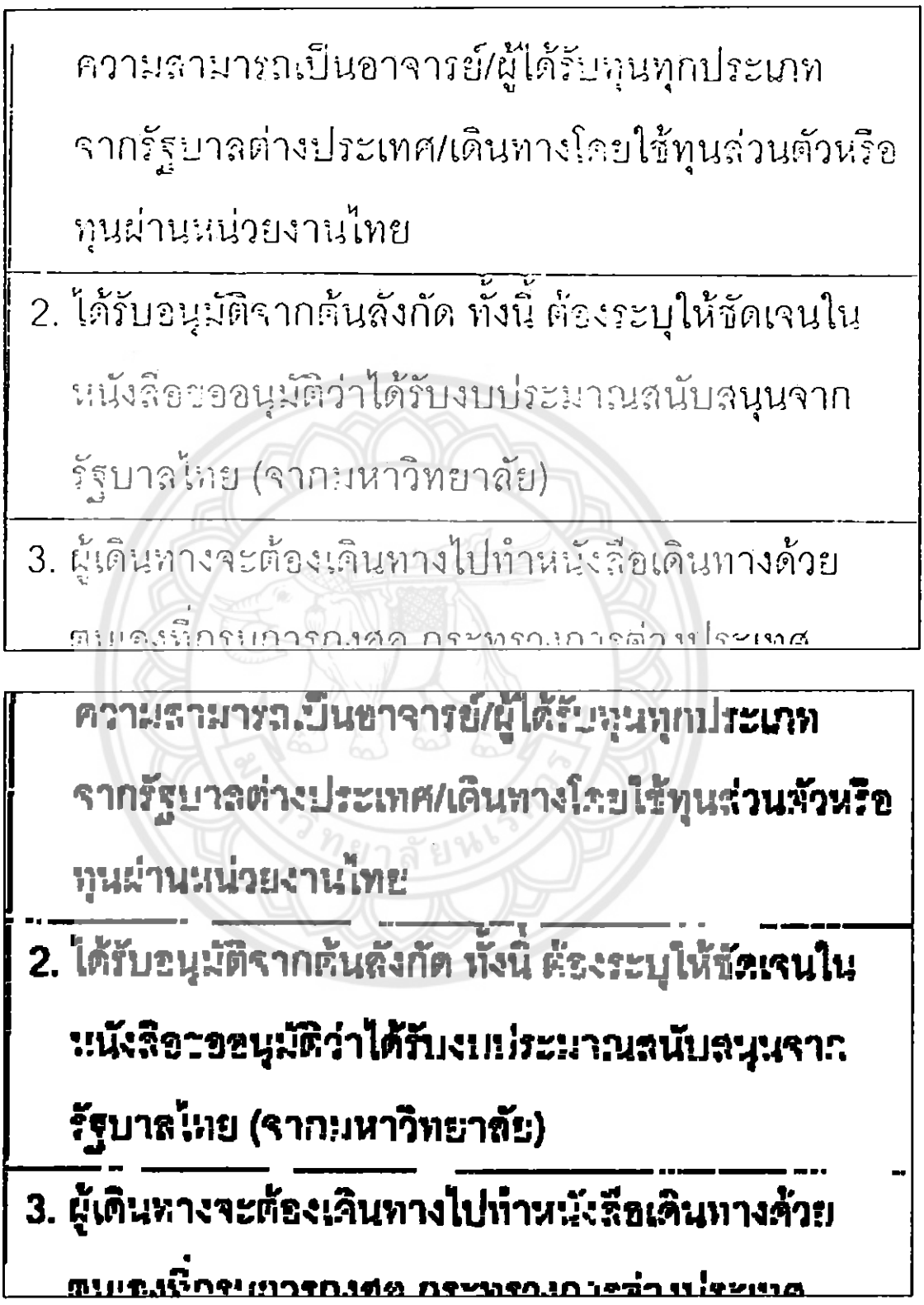
ความสามารถเป็นอาจารย์/ผู้ได้รับทุนทุกประเภท จากรัฐบาลต่างประเทศ/เดินทางโดยใช้ทุนส่วนตัวหรือ ทุนผ่านหน่วยงานไทย
2. ได้รับอนุมัติจากต้นสังกัด ทั้งนี้ ต้องระบุให้ชัดเจนใน หนังสืออนุมัติว่าได้รับงบประมาณสนับสนุนจาก รัฐบาลไทย (จากมหาวิทยาลัย)
3. ผู้เดินทางจะต้องเดินทางไปทำหนังสือเดินทางด้วย

รูปที่ 4.13 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบไม่มี Noise (บน) เทียบกับ ผลลัพธ์จากการใช้ Closing ด้วย Structure element (ล่าง)

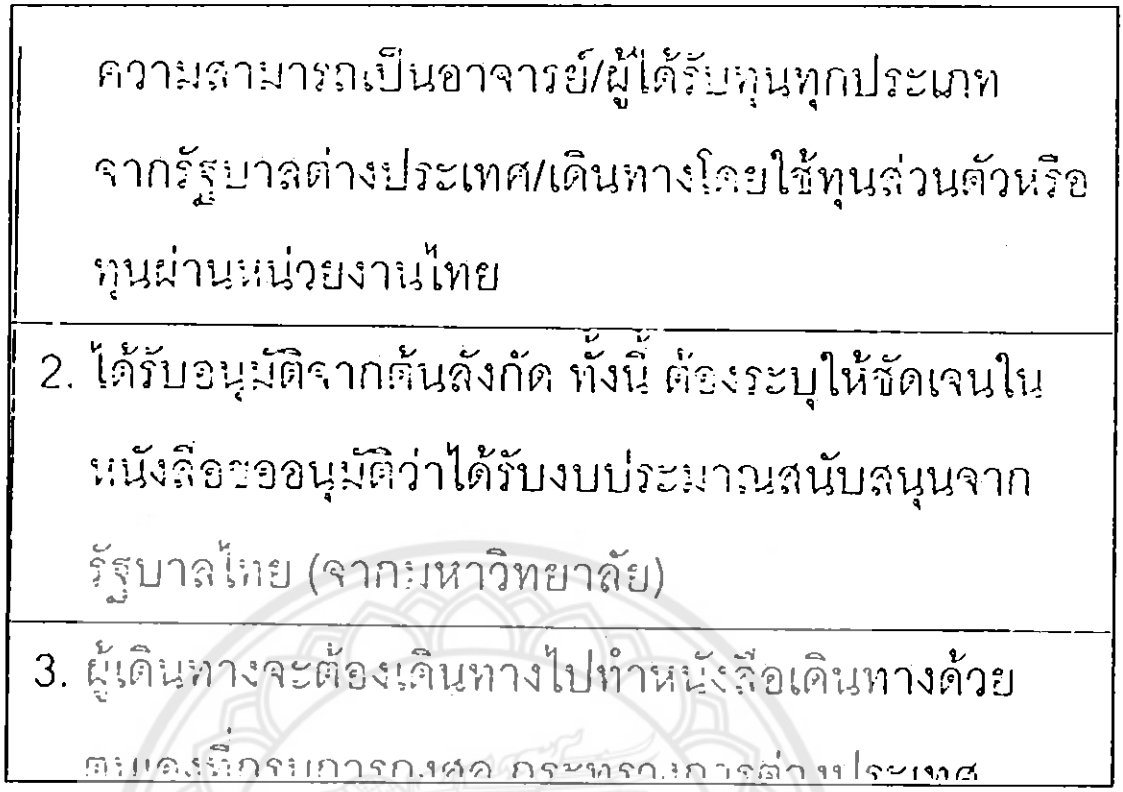


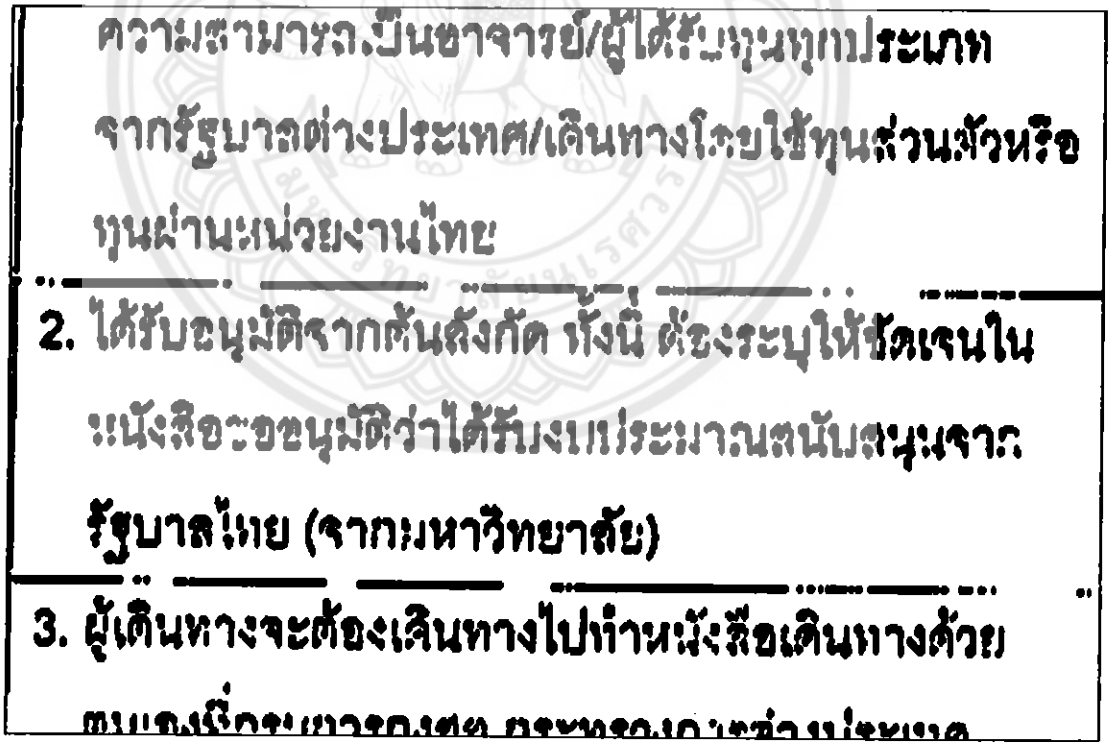
รูปที่ 4.14 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบไม่มี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Closing ด้วย Structure element (ล่าง)

4.1.3 ผลลัพธ์จากการใช้ Closing + Dilation

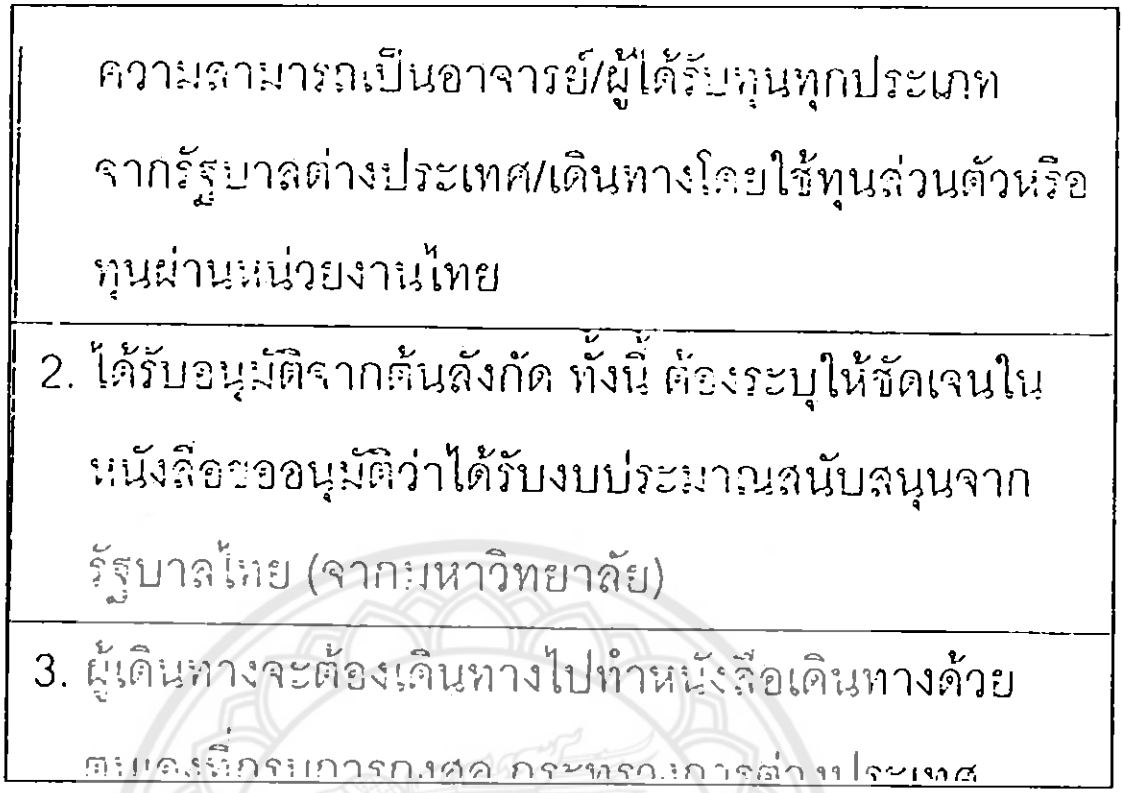


รูปที่ 4.15 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบไม่มี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Closing + Dilation ด้วย Structure element ⊞ (ล่าง)

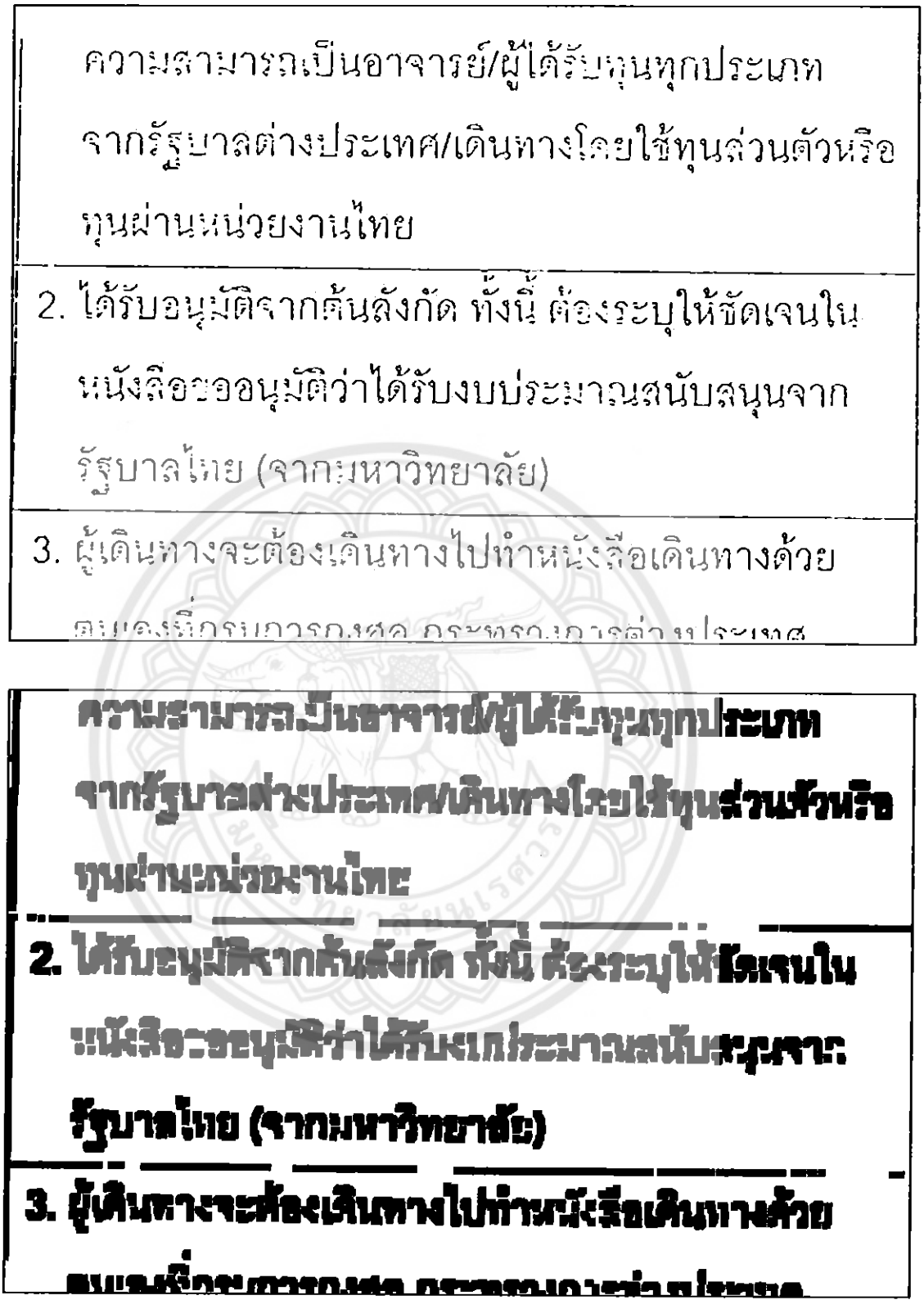




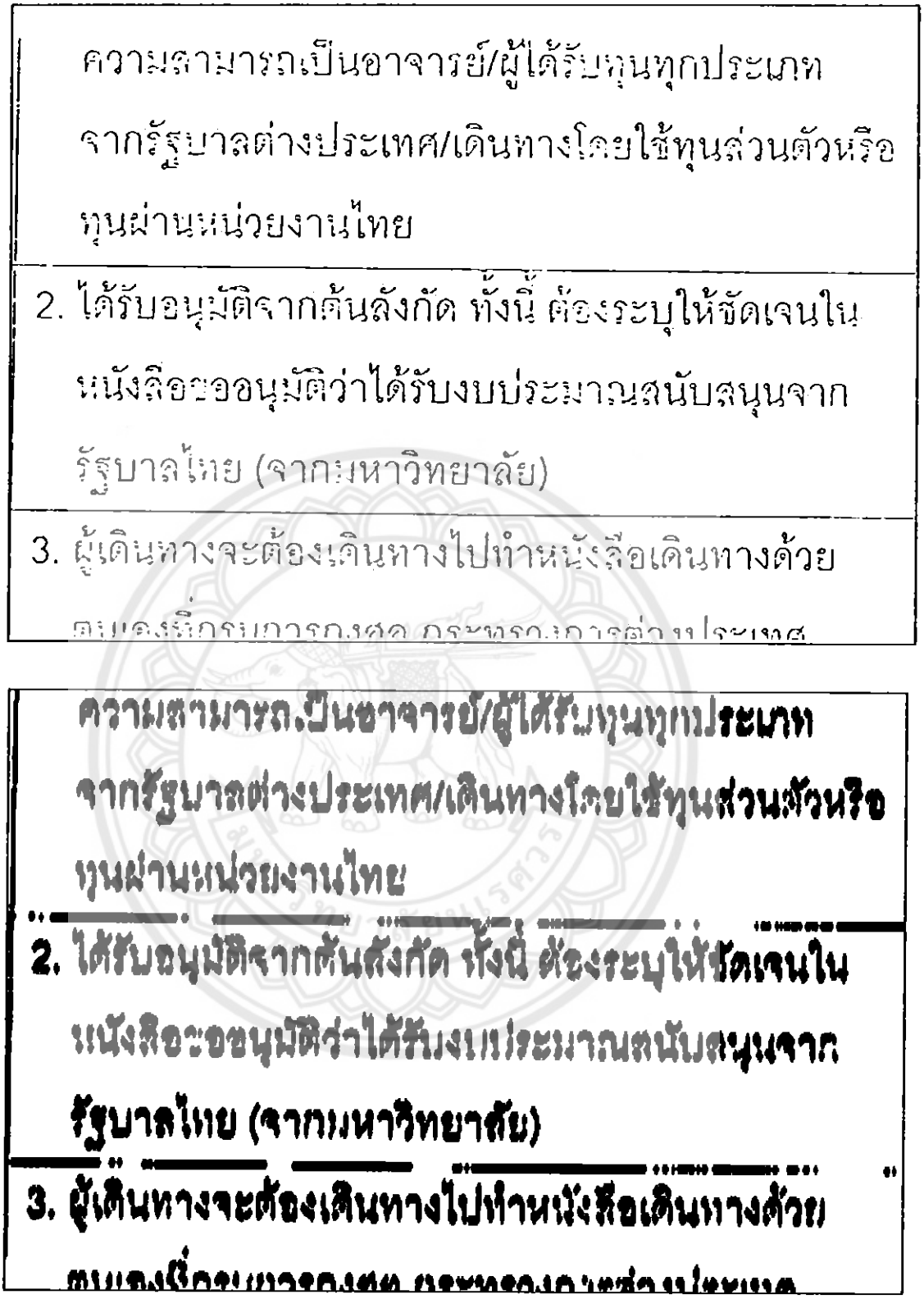
รูปที่ 4.16 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบไม่มี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Closing + Dilation ด้วย Structure element (ล่าง)



รูปที่ 4.17 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบไม่มี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Closing + Dilation ด้วย Structure element (ล่าง)



รูปที่ 4.18 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบไม่มี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Closing + Dilation ด้วย Structure element (ล่าง)

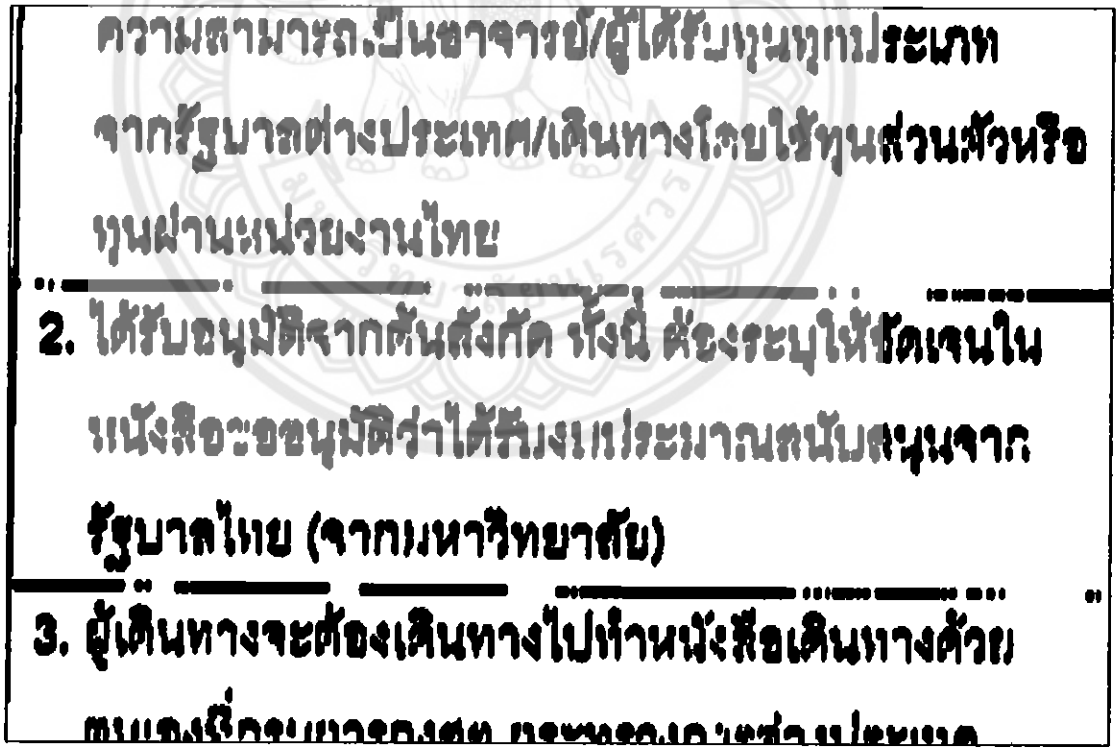


รูปที่ 4.19 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบไม่มี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Closing + Dilation ด้วย Structure element (ล่าง)

ความสามารถเป็นอาจารย์/ผู้ได้รับทุนทุกประเภท
จากรัฐบาลต่างประเทศ/เดินทางโดยใช้ทุนส่วนตัวหรือ
ทุนผ่านหน่วยงานไทย

2. ได้รับอนุมัติจากต้นสังกัด ทั้งนี้ ต้องระบุให้ชัดเจนใน
หนังสือขออนุมัติว่าได้รับงบประมาณสนับสนุนจาก
รัฐบาลไทย (จากมหาวิทยาลัย)

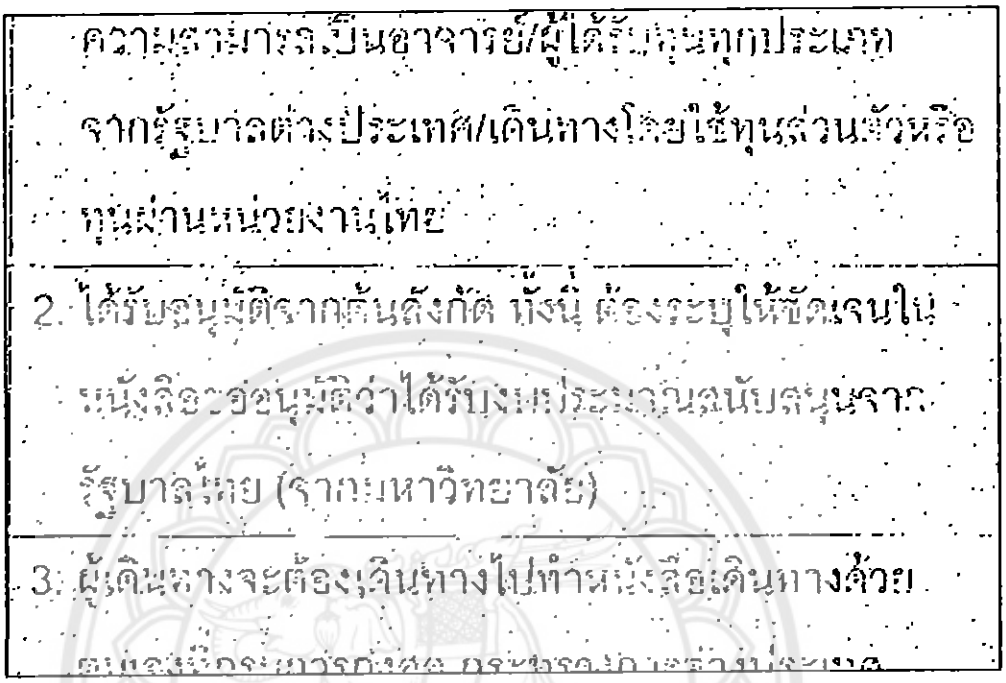
3. ผู้เดินทางจะต้องเดินทางไปทำหนังสือเดินทางด้วย
ตนเองที่กรมการกงสุล กระทรวงการต่างประเทศ



รูปที่ 4.20 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบไม่มี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Closing + Dilation ด้วย Structure element (ล่าง)

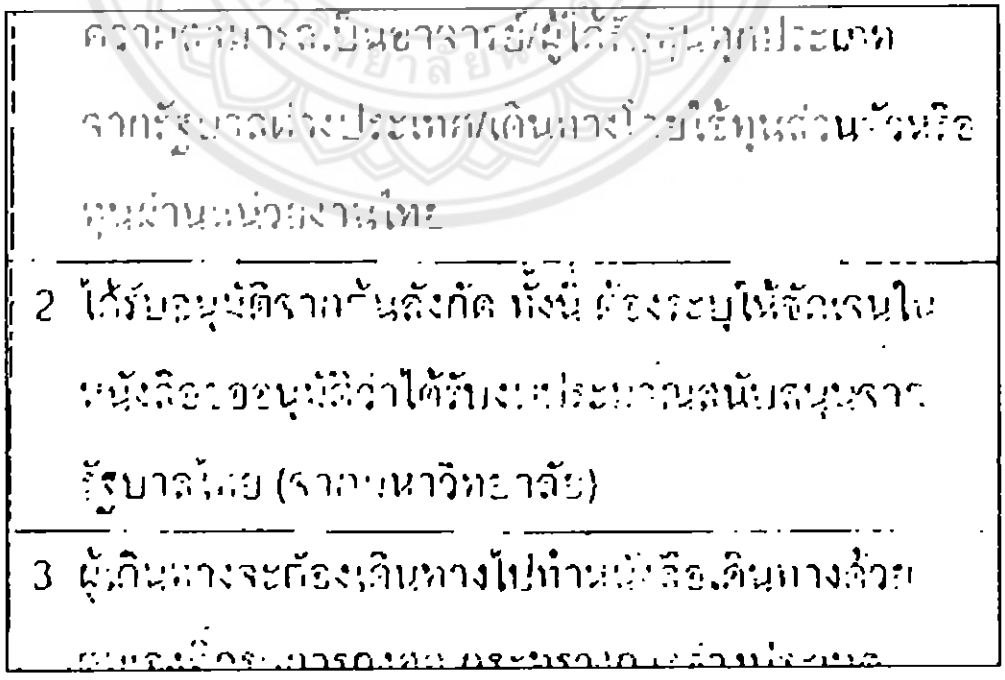
4.2 ผลการทดลองสำหรับภาพที่มีลักษณะตัวอักษรบางหรือเลือนรางแบบมี noise

ข้อมูลภาพที่นำมาใช้ในการทดลอง : ภาพระดับเทาขนาด 438x662 พิกเซล รูปแบบการจัดเก็บเป็น .jpg ใช้ noise ชนิด salt & pepper



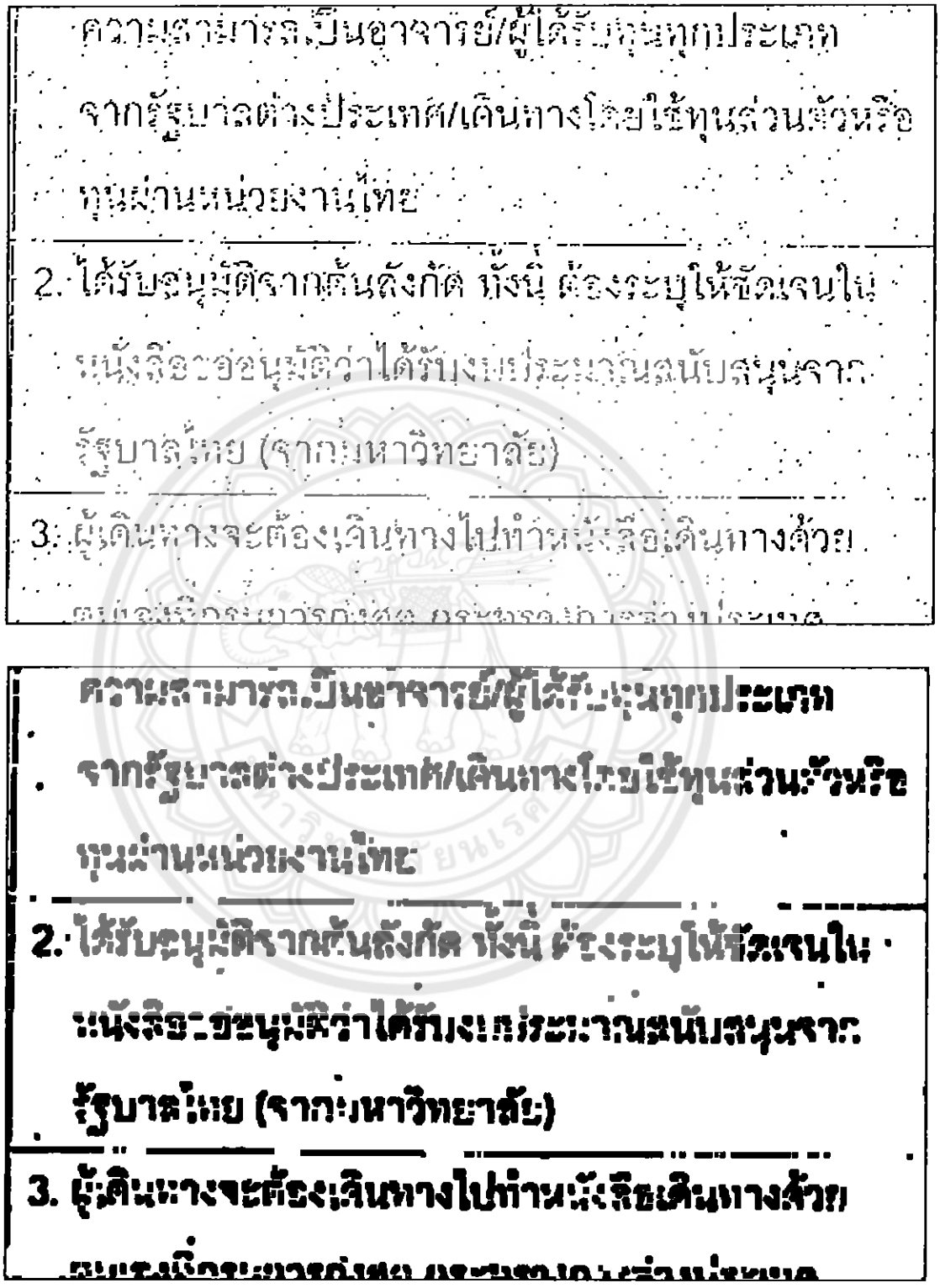
รูปที่ 4.21 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบมี Noise

ผลลัพธ์จากการลบ Noise โดยใช้ Median Filter

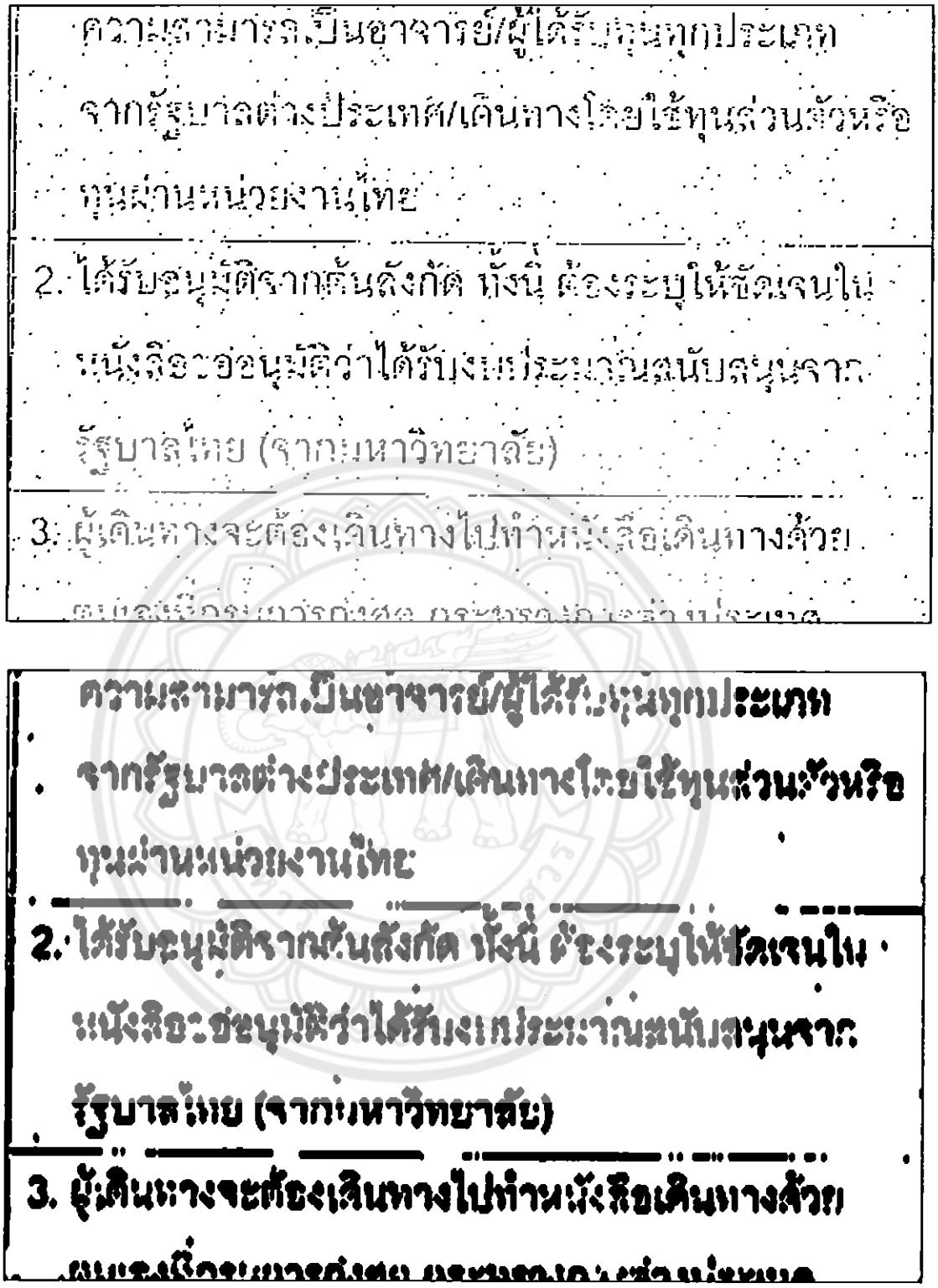


รูปที่ 4.22 ผลลัพธ์การลบ Noise โดยใช้ Median Filter

4.2.1 ผลลัพธ์จากการใช้ Dilation



รูปที่ 4.23 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบมี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Dilation ด้วย Structure element (ล่าง)



ความสามารถเป็นอาจารย์/ผู้ได้รับทุนทุกประเภท
จากรัฐบาลต่างประเทศ/เดินทางโดยใช้ทุนส่วนตัวหรือ
ทุนผ่านหน่วยงานไทย

2. ได้รับอนุมัติจากต้นสังกัด ทั้งนี้ ต้องระบุให้ชัดเจนใน
หนังสืออนุมัติว่าได้รับงบประมาณสนับสนุนจาก
รัฐบาลไทย (จากมหาวิทยาลัย)

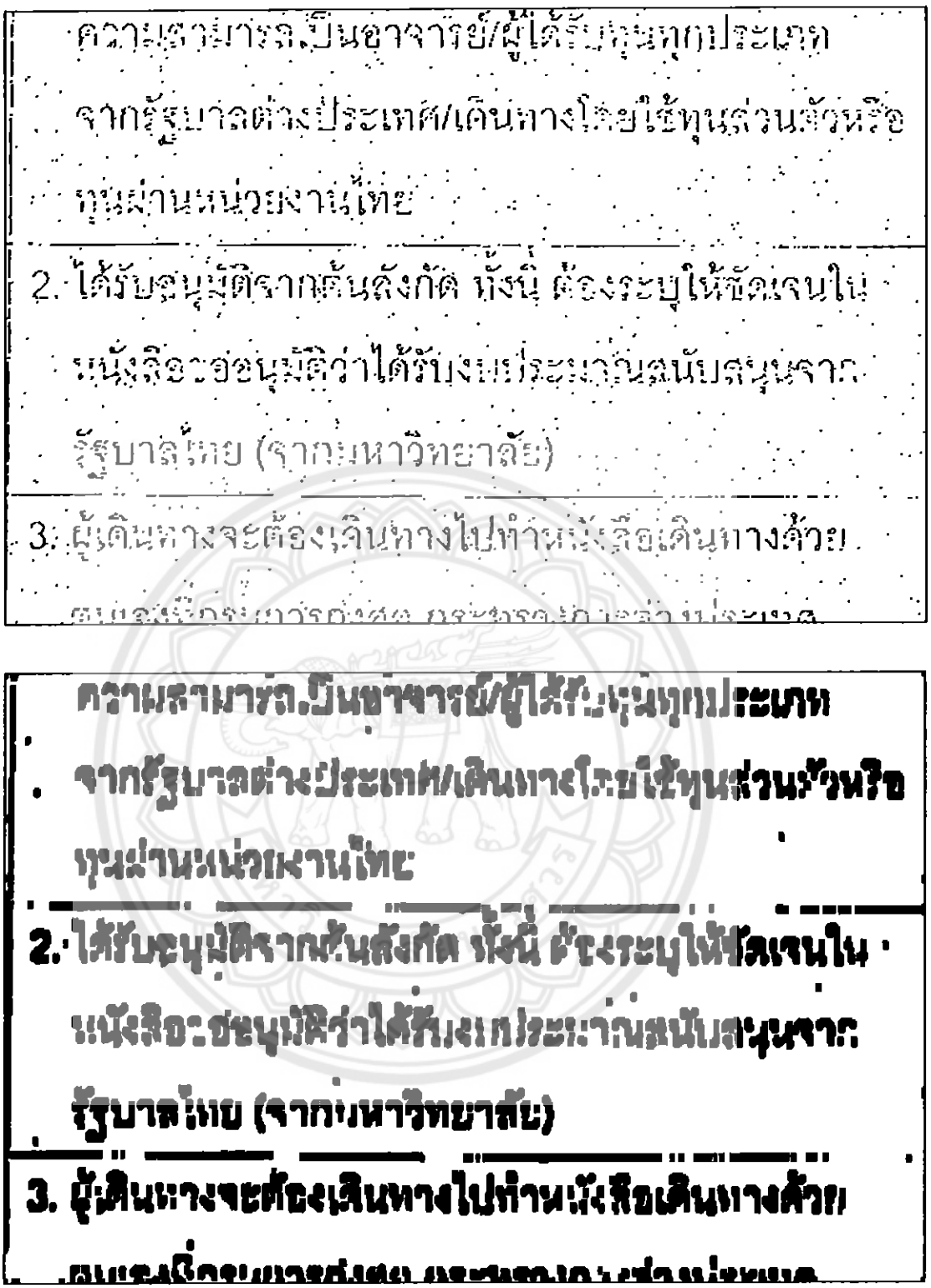
3. ผู้เดินทางจะต้องเดินทางไปทำหนังสือเดินทางด้วย

ความสามารถเป็นอาจารย์/ผู้ได้รับทุนทุกประเภท
จากรัฐบาลต่างประเทศ/เดินทางโดยใช้ทุนส่วนตัวหรือ
ทุนผ่านหน่วยงานไทย

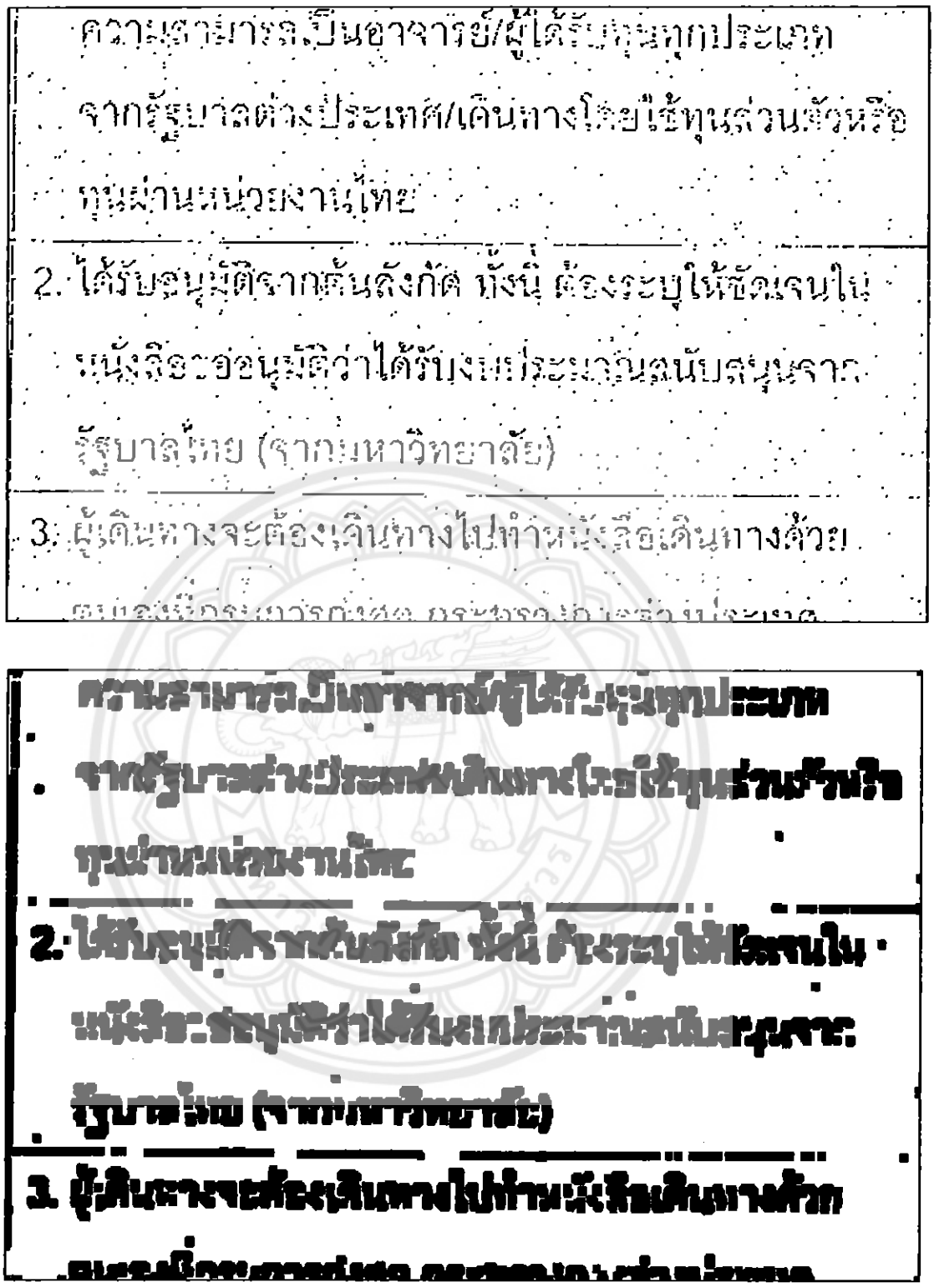
2. ได้รับอนุมัติจากต้นสังกัด ทั้งนี้ ต้องระบุให้ชัดเจนใน
หนังสืออนุมัติว่าได้รับงบประมาณสนับสนุนจาก
รัฐบาลไทย (จากมหาวิทยาลัย)

3. ผู้เดินทางจะต้องเดินทางไปทำหนังสือเดินทางด้วย

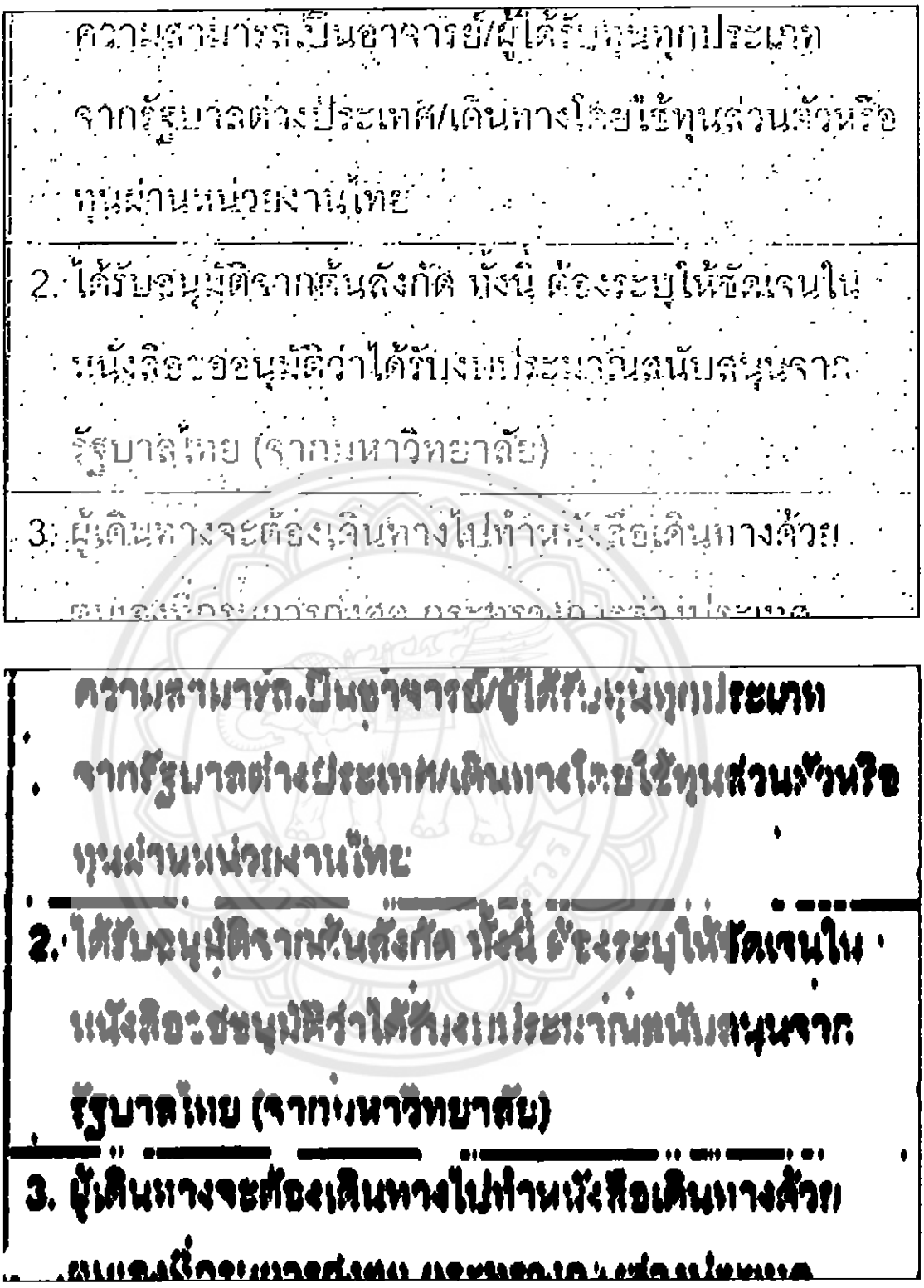
รูปที่ 4.24 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบมี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Dilation ด้วย Structure element (ล่าง)



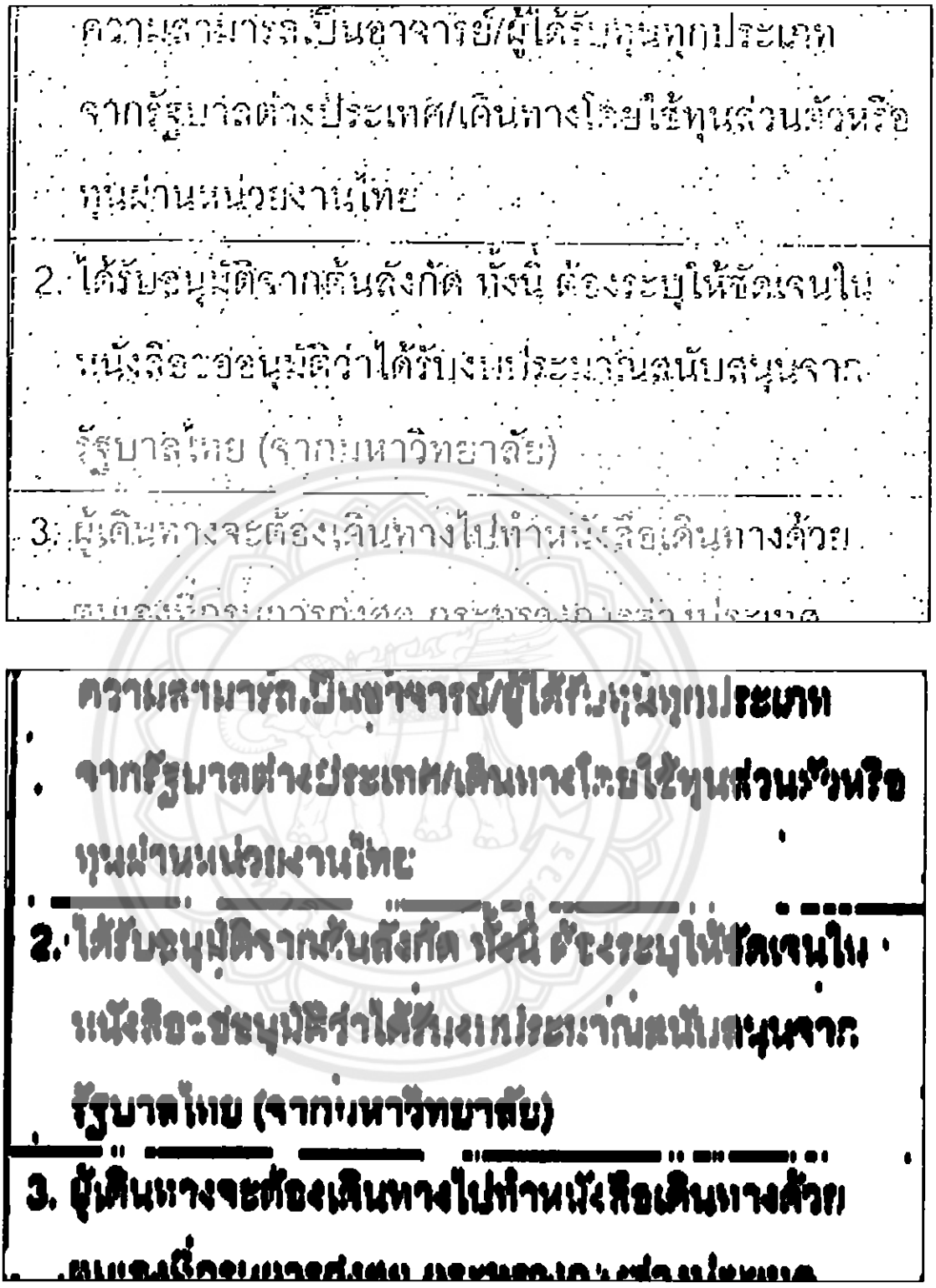
รูปที่ 4.25 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบมี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Dilation ด้วย Structure element (ล่าง)



รูปที่ 4.26 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบมี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Dilation ด้วย Structure element (ล่าง)

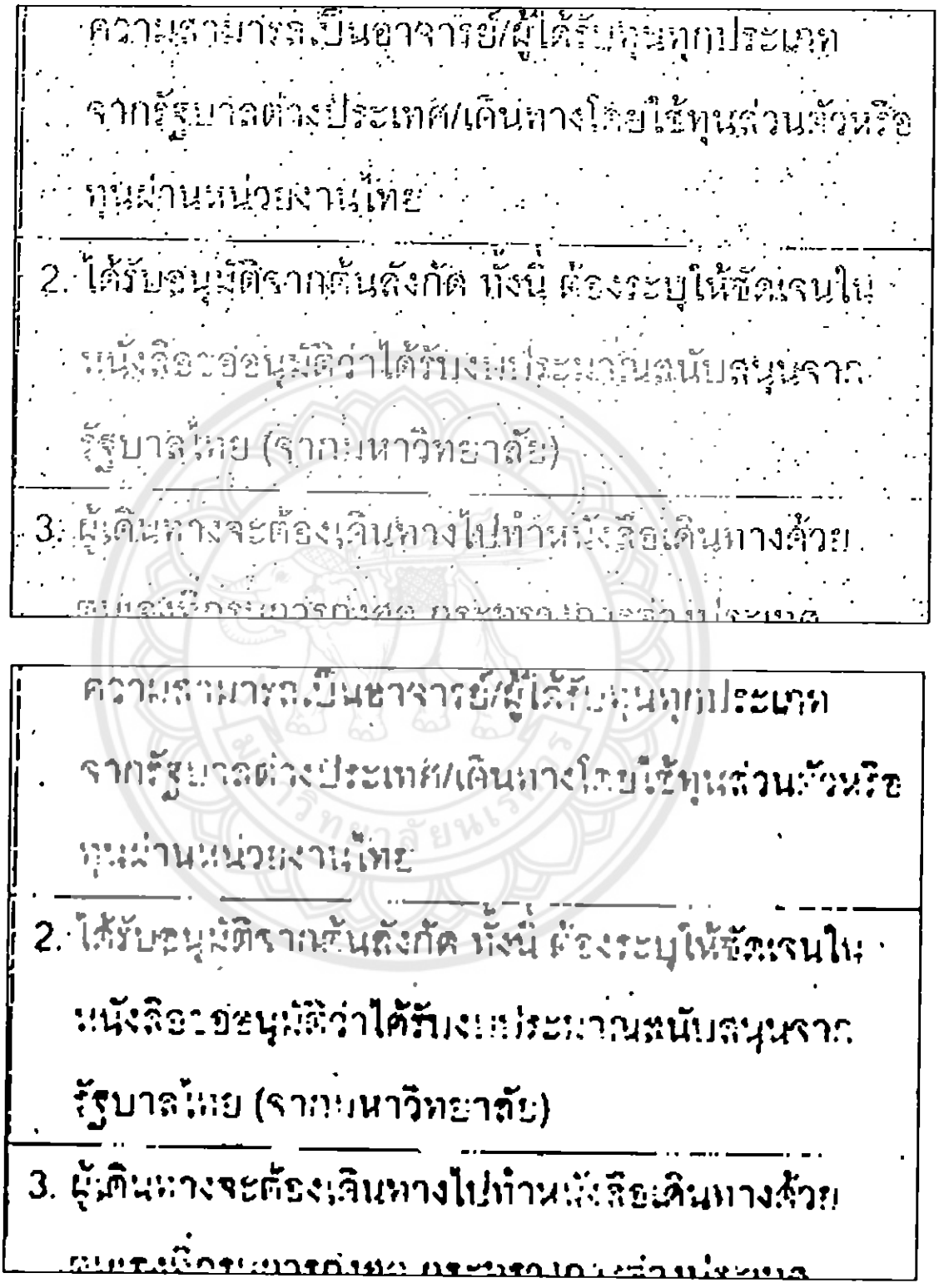


รูปที่ 4.27 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบมี Noise (บน) เทียบกับ ผลลัพธ์จากการใช้ Dilation ด้วย Structure element (ล่าง)

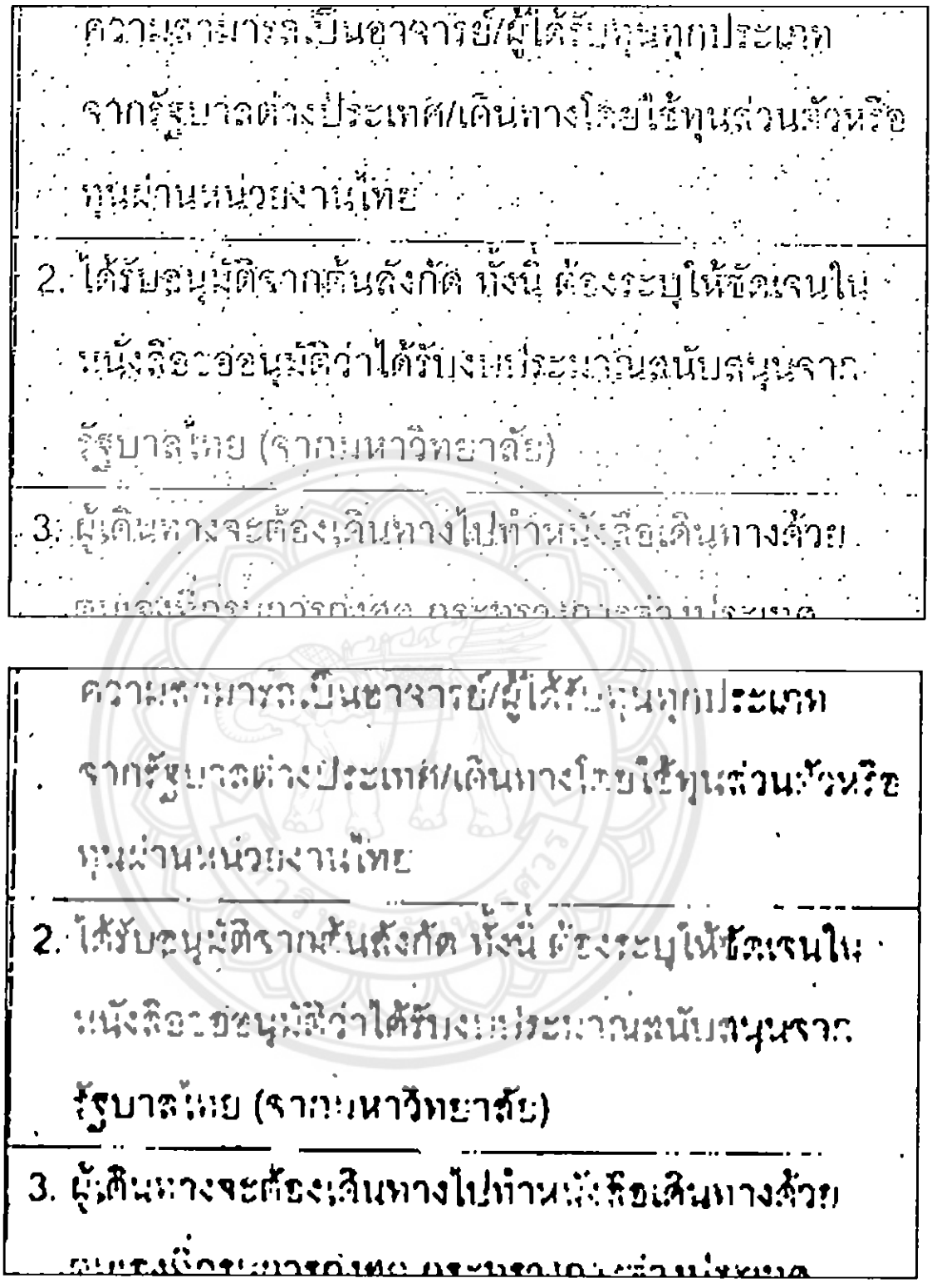


รูปที่ 4.28 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบมี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Dilation ด้วย Structure element (ล่าง)

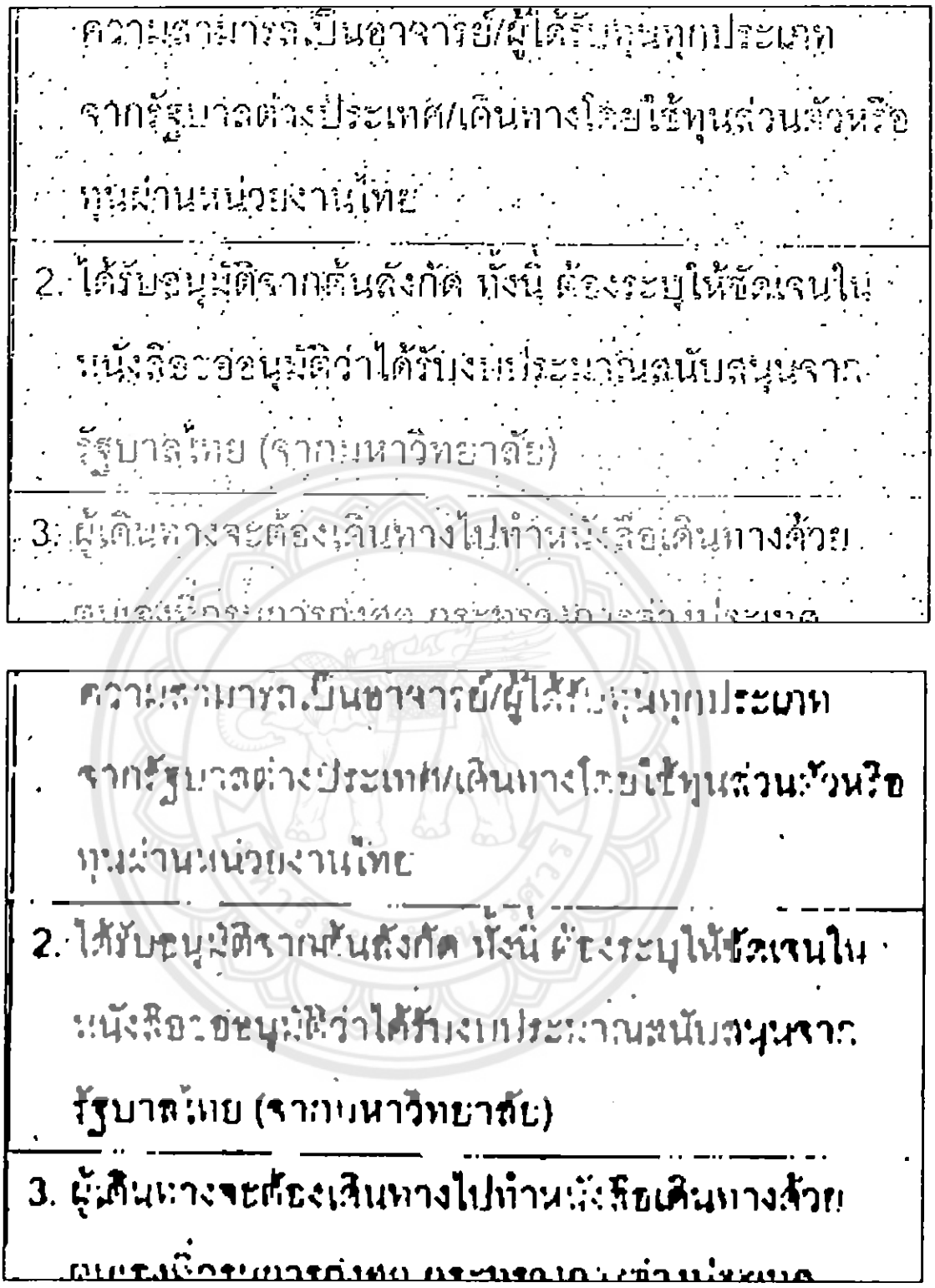
4.2.2 ผลลัพธ์จากการใช้ Closing



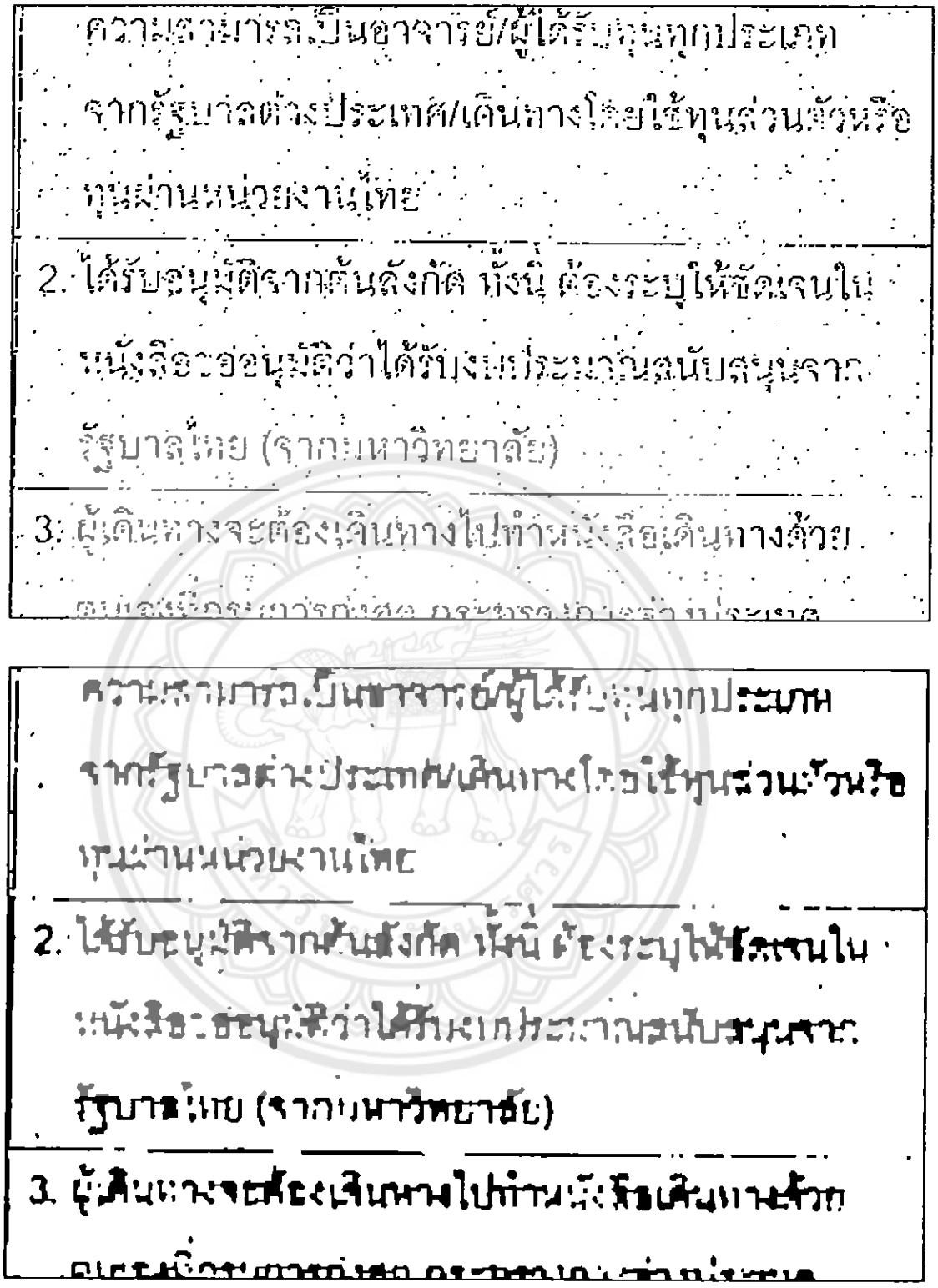
รูปที่ 4.29 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบมี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Closing ด้วย Structure element ⊞ (ล่าง)



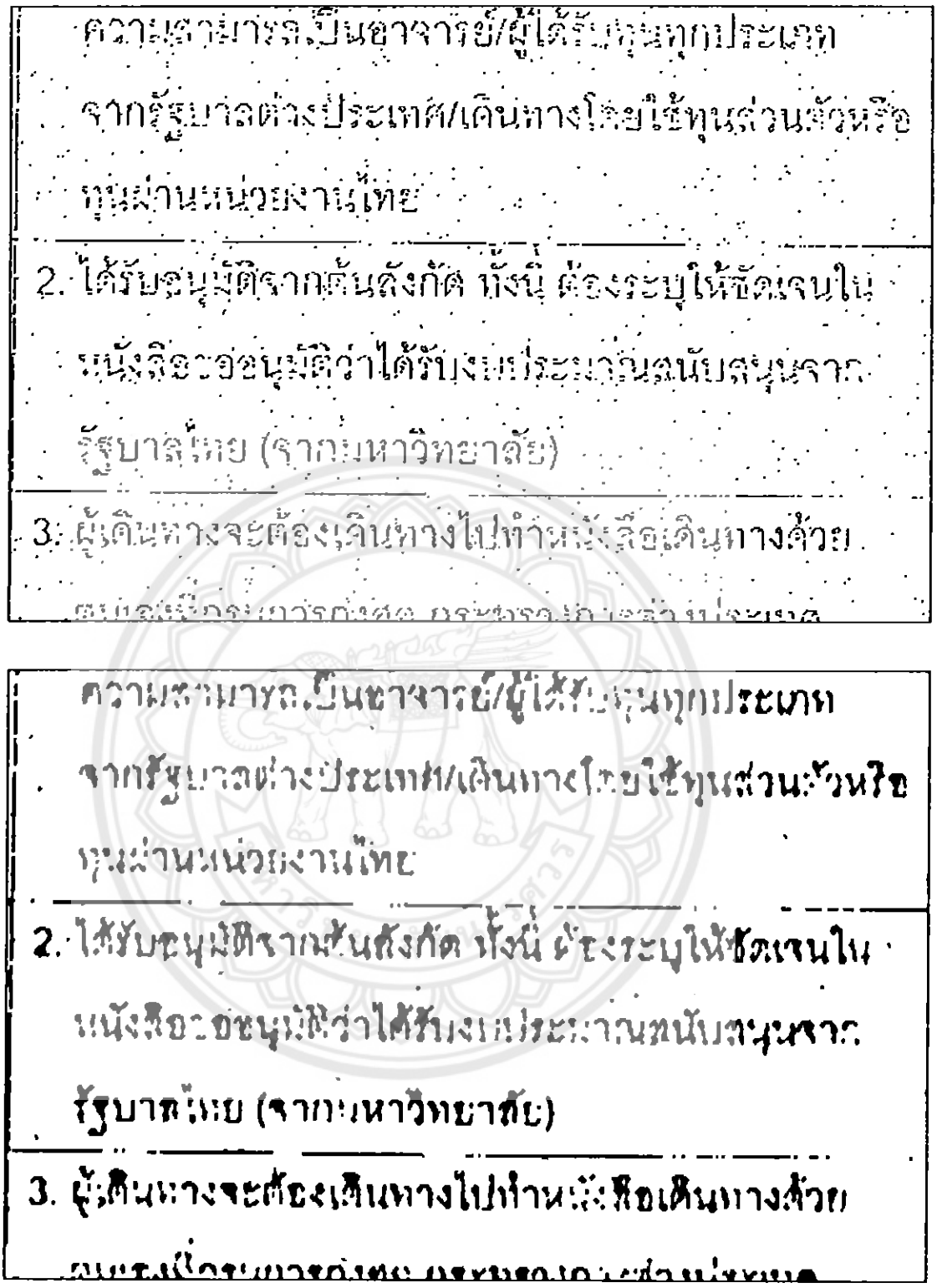
รูปที่ 4.30 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบมี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Closing ด้วย Structure element (ล่าง)



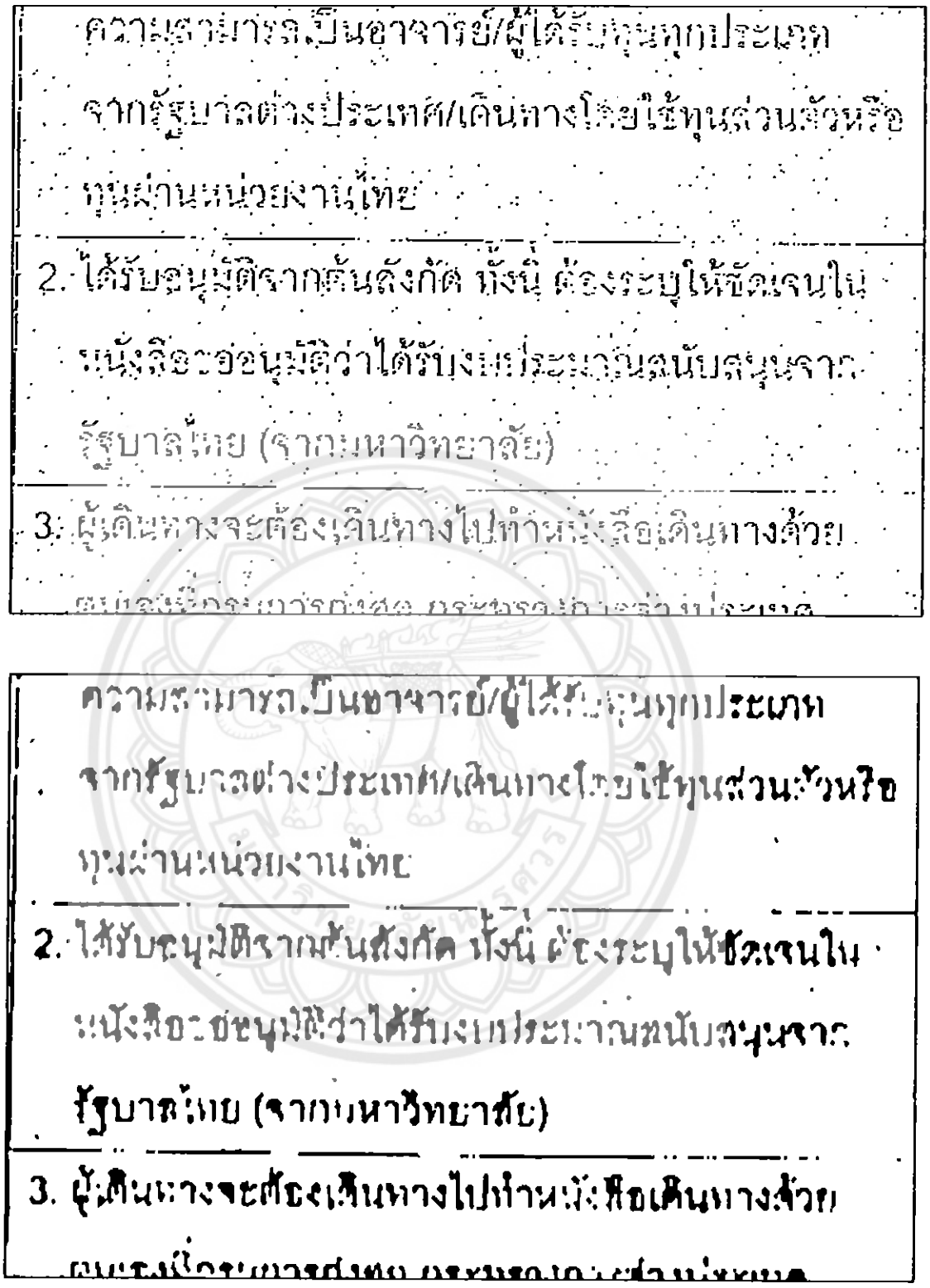
รูปที่ 4.31 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบมี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Closing ด้วย Structure element (ล่าง)



รูปที่ 4.32 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบมี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Closing ด้วย Structure element (ล่าง)



รูปที่ 4.33 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบมี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Closing ด้วย Structure element (ล่าง)



ความสามารถเป็นอาจารย์/ผู้ได้รับทุนทุกประเภท
จากรัฐบาลต่างประเทศ/เดินทางโดยใช้ทุนส่วนตัวหรือ
ทุนผ่านหน่วยงานไทย

2. ได้รับอนุมัติจากต้นสังกัด ทั้งนี้ ต้องระบุให้ชัดเจนใน
หนังสืออนุมัติว่าได้รับงบประมาณสนับสนุนจาก
รัฐบาลไทย (จากมหาวิทยาลัย)

3. ผู้เดินทางจะต้องเดินทางไปทำหน้าที่อเดินทางด้วย
ตนเองเป็นการส่วนตัว และจะต้องไม่ขัดต่อ

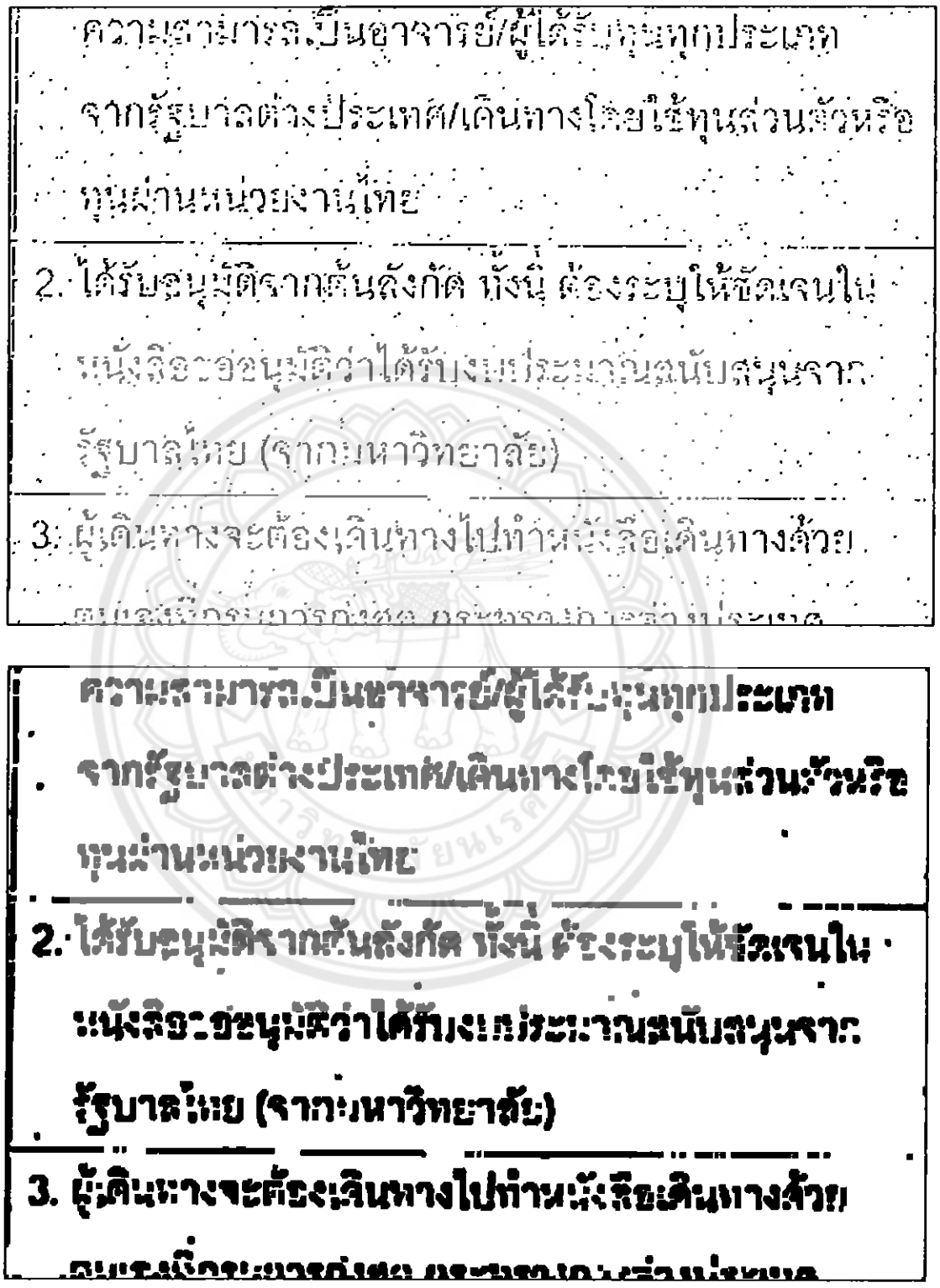
ความสามารถเป็นอาจารย์/ผู้ได้รับทุนทุกประเภท
จากรัฐบาลต่างประเทศ/เดินทางโดยใช้ทุนส่วนตัวหรือ
ทุนผ่านหน่วยงานไทย

2. ได้รับอนุมัติจากต้นสังกัด ทั้งนี้ ต้องระบุให้ชัดเจนใน
หนังสืออนุมัติว่าได้รับงบประมาณสนับสนุนจาก
รัฐบาลไทย (จากมหาวิทยาลัย)

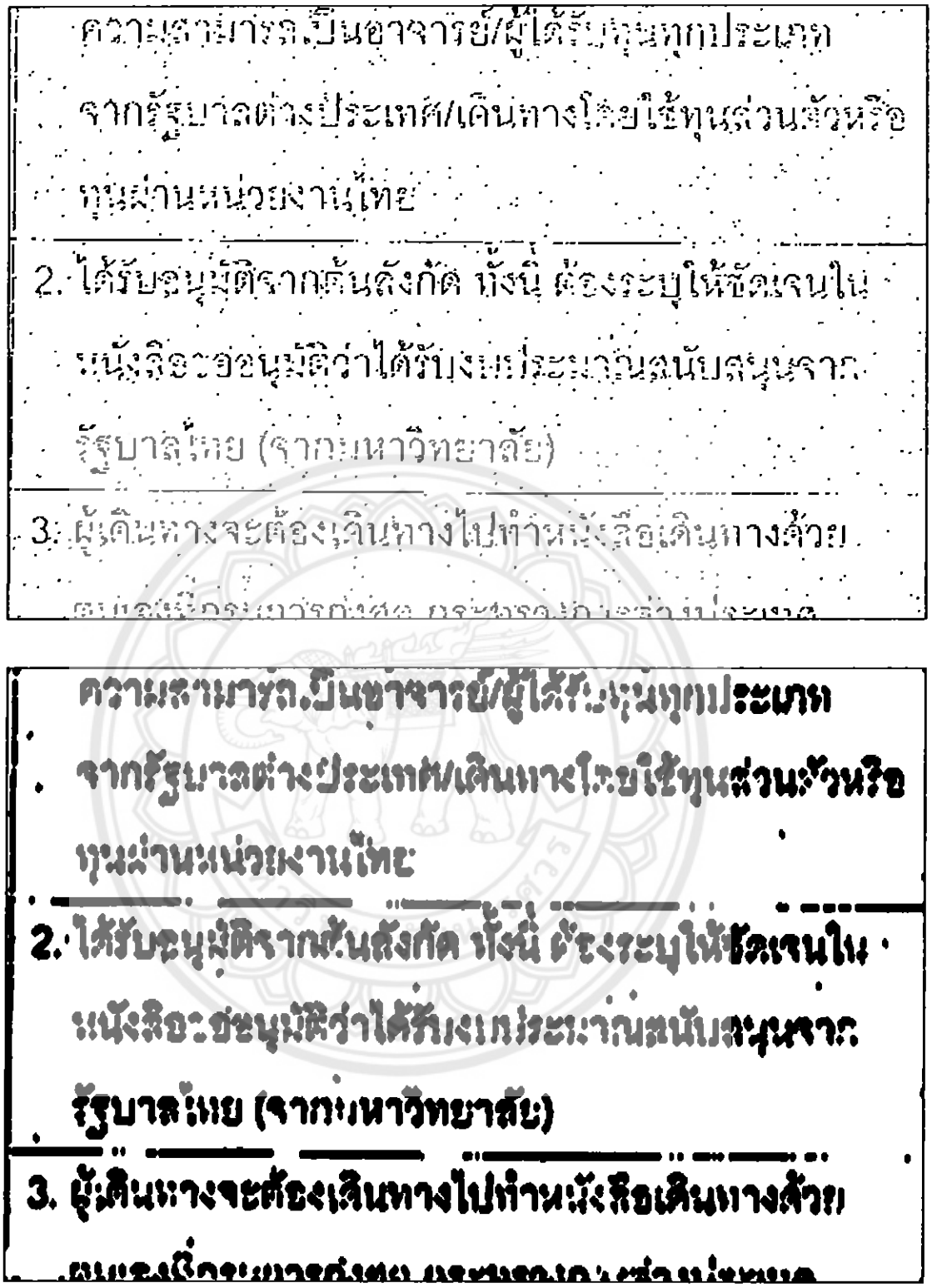
3. ผู้เดินทางจะต้องเดินทางไปทำหน้าที่อเดินทางด้วย
ตนเองเป็นการส่วนตัว และจะต้องไม่ขัดต่อ

รูปที่ 4.34 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบมี Noise (บน) เทียบกับ
ผลลัพธ์จากการใช้ Closing ด้วย Structure element (ล่าง)

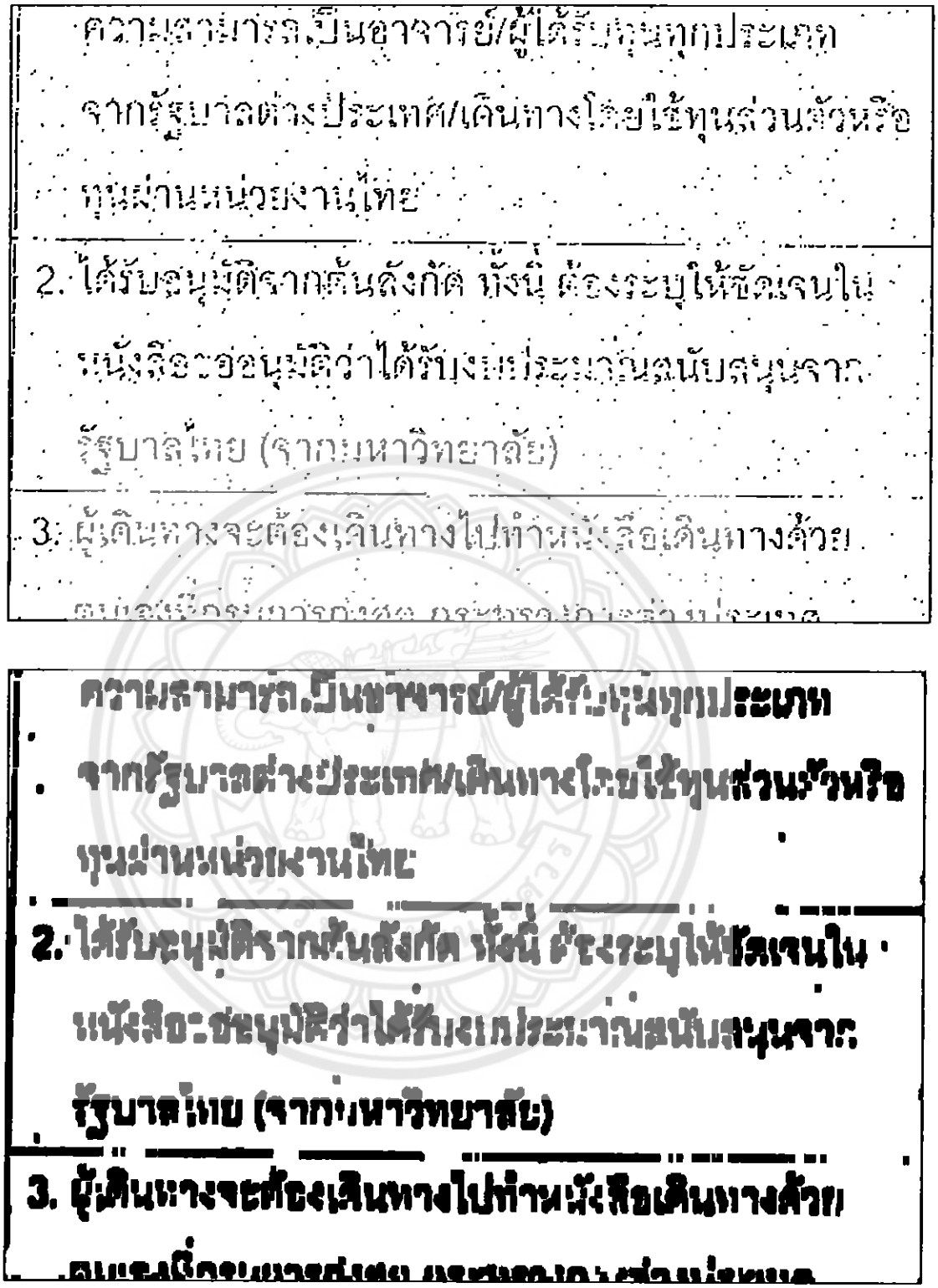
4.2.3 ผลลัพธ์จากการใช้ Closing + Dilation



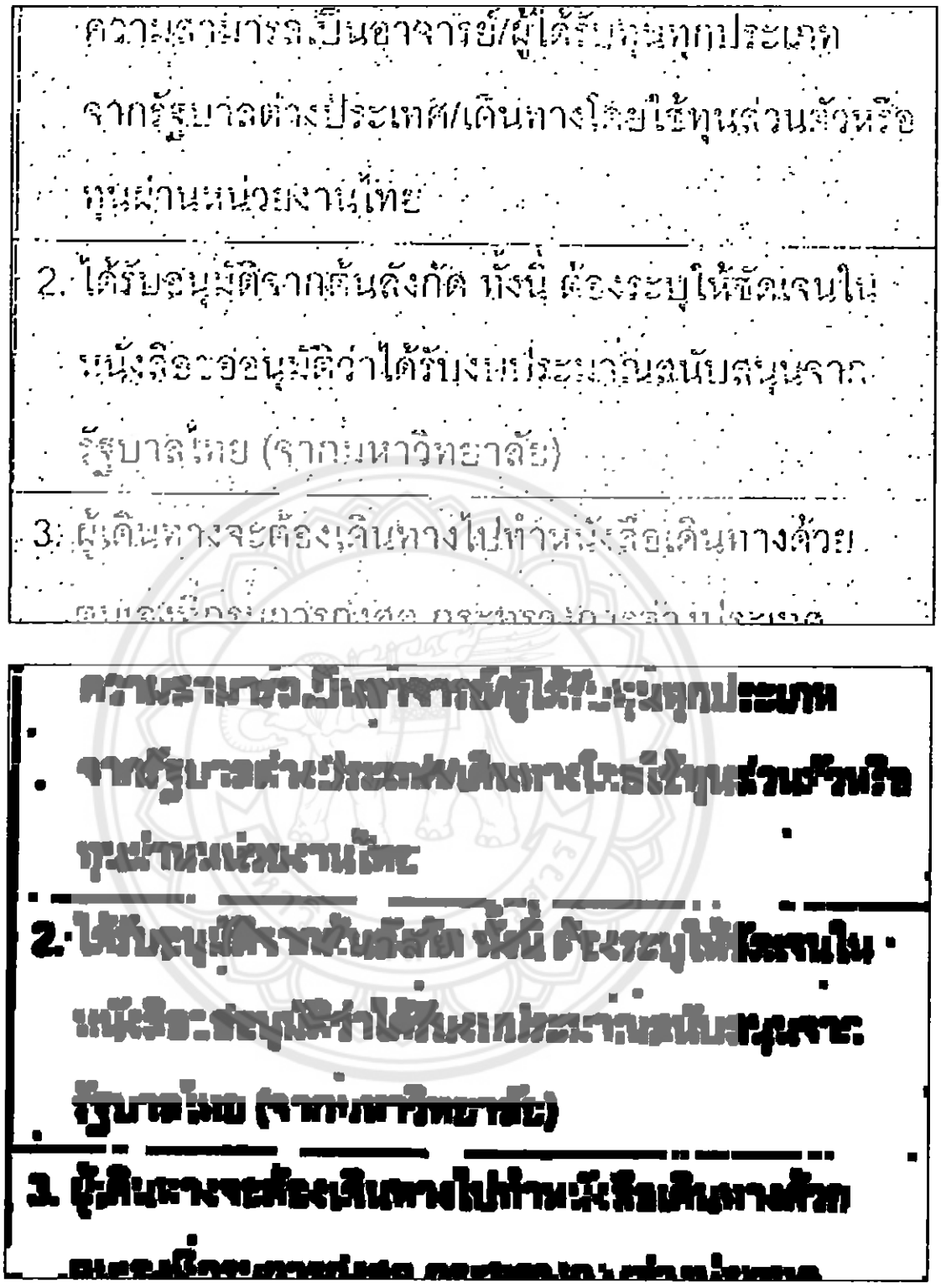
รูปที่ 4.35 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบมี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Closing + Dilation ด้วย Structure element (ล่าง)



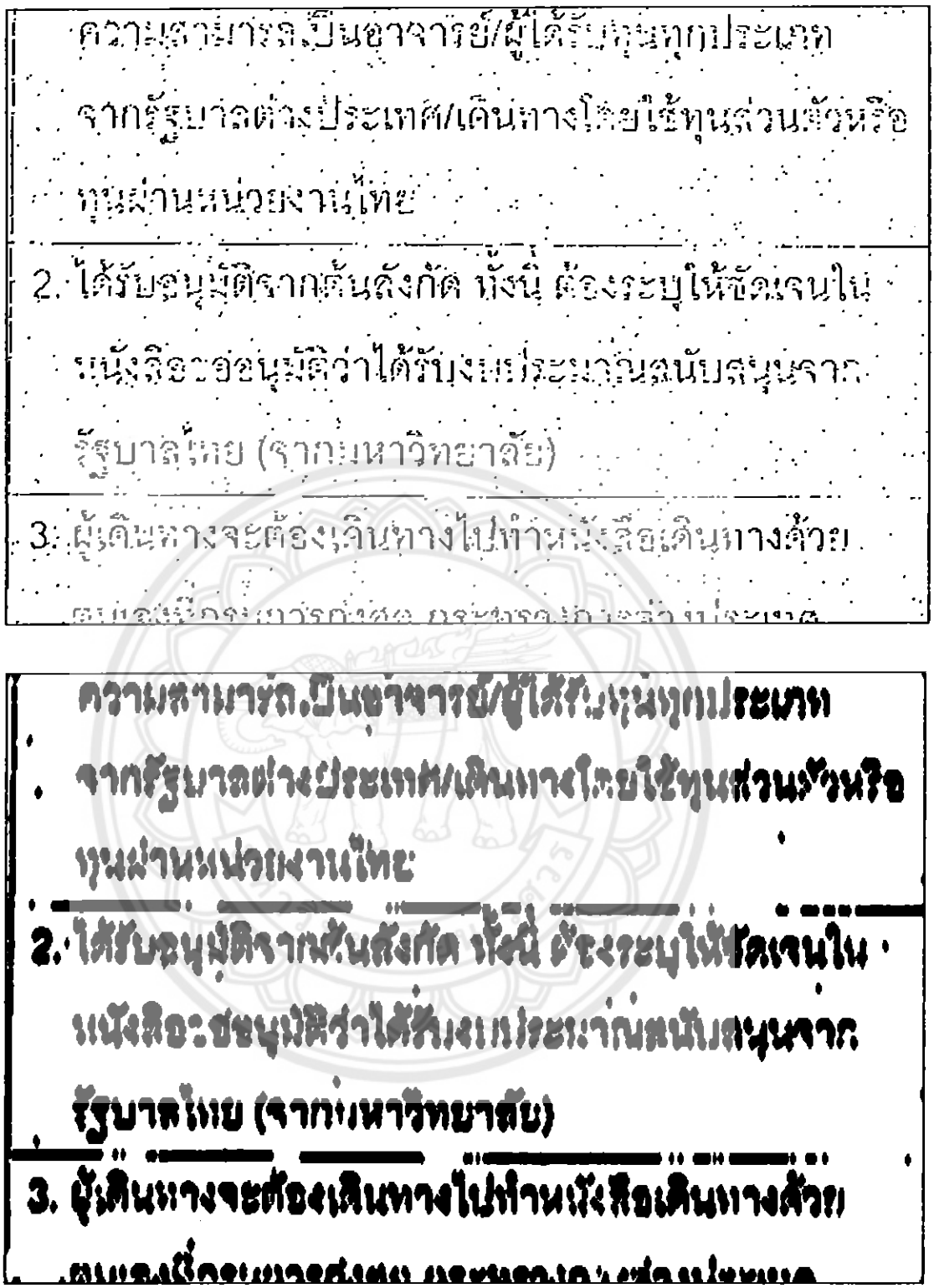
รูปที่ 4.36 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบมี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Closing + Dilation ด้วย Structure element (ล่าง)



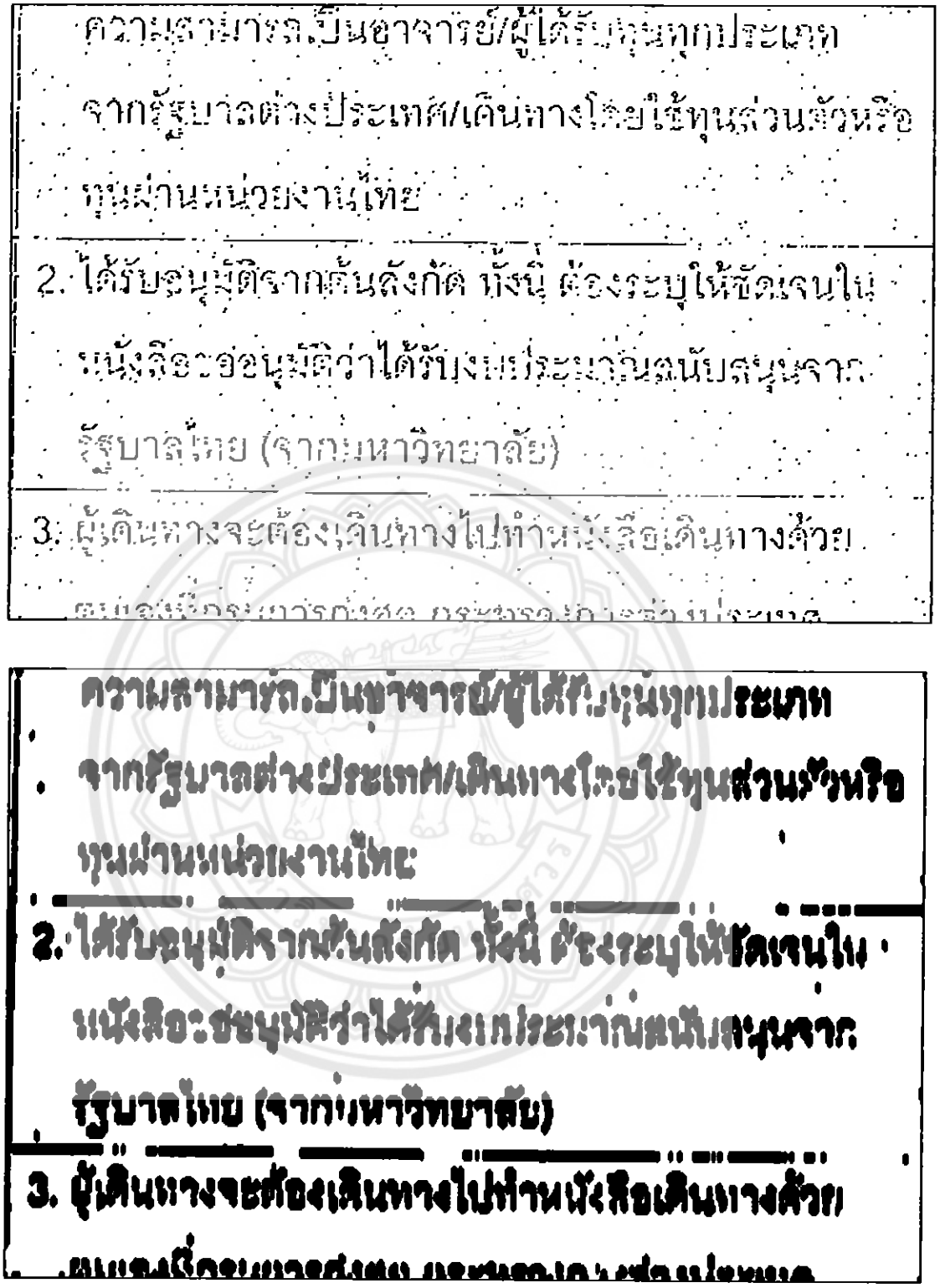
รูปที่ 4.37 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบมี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Closing + Dilation ด้วย Structure element (ล่าง)



รูปที่ 4.38 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบมี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Closing + Dilation ด้วย Structure element (ล่าง)



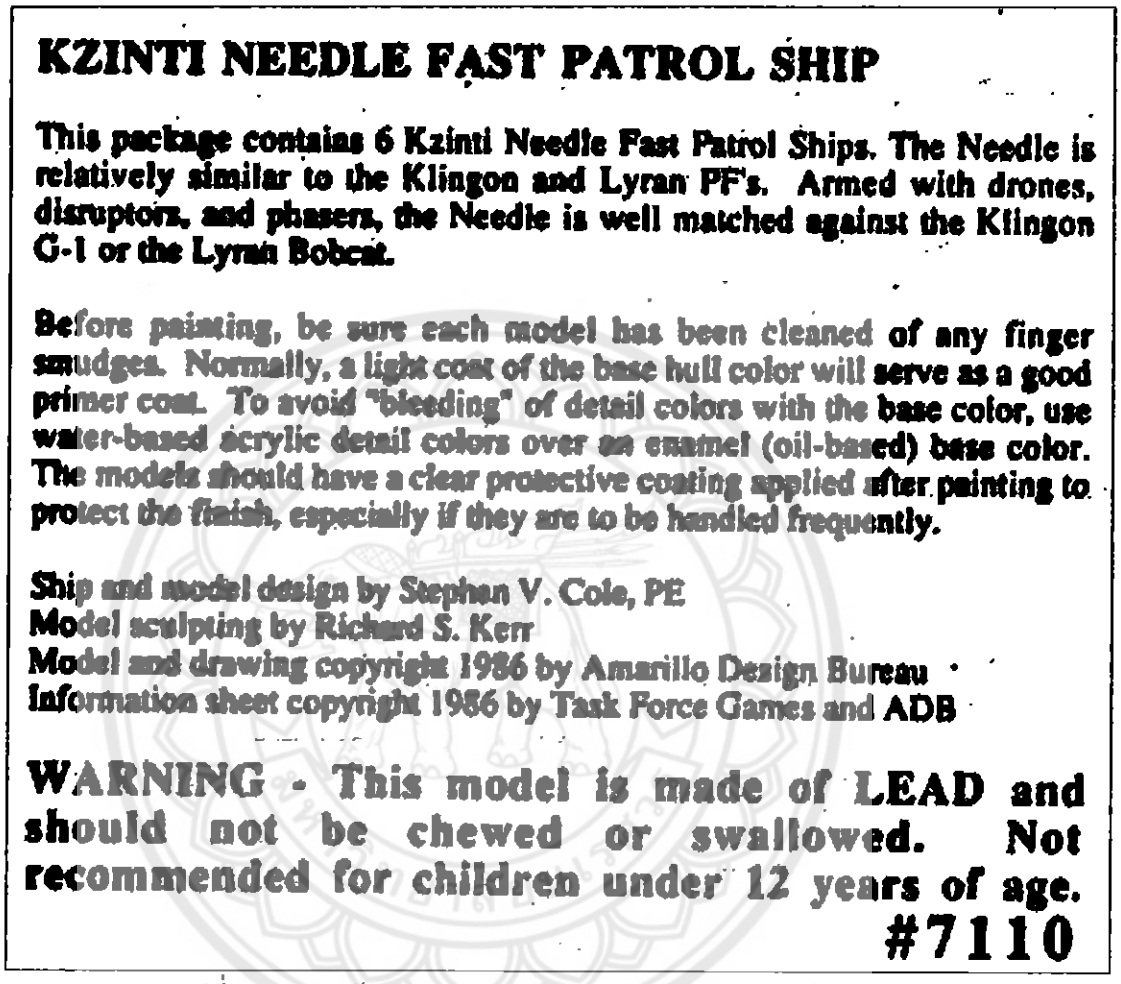
รูปที่ 4.39 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบมี Noise (บน) เทียบกับ ผลลัพธ์จากการใช้ Closing + Dilation ด้วย Structure element (ล่าง)



รูปที่ 4.40 ภาพต้นฉบับของตัวอักษรบางหรือเลือนรางแบบมี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Closing + Dilation ด้วย Structure element (ล่าง)

4.3 ผลการทดลองสำหรับภาพที่มีลักษณะตัวอักษรหนาหรือติดกันแบบไม่มี noise

ข้อมูลภาพที่นำมาใช้ในการทดลอง : ภาพขาวดำขนาด 641x734 พิกเซล รูปแบบการจัดเก็บเป็น .bmp


KZINTI NEEDLE FAST PATROL SHIP

This package contains 6 Kzinti Needle Fast Patrol Ships. The Needle is relatively similar to the Klingon and Lyran PF's. Armed with drones, disruptors, and phasers, the Needle is well matched against the Klingon G-1 or the Lyran Bobcat.

Before painting, be sure each model has been cleaned of any finger smudges. Normally, a light coat of the base hull color will serve as a good primer coat. To avoid "bleeding" of detail colors with the base color, use water-based acrylic detail colors over an enamel (oil-based) base color. The models should have a clear protective coating applied after painting to protect the finish, especially if they are to be handled frequently.

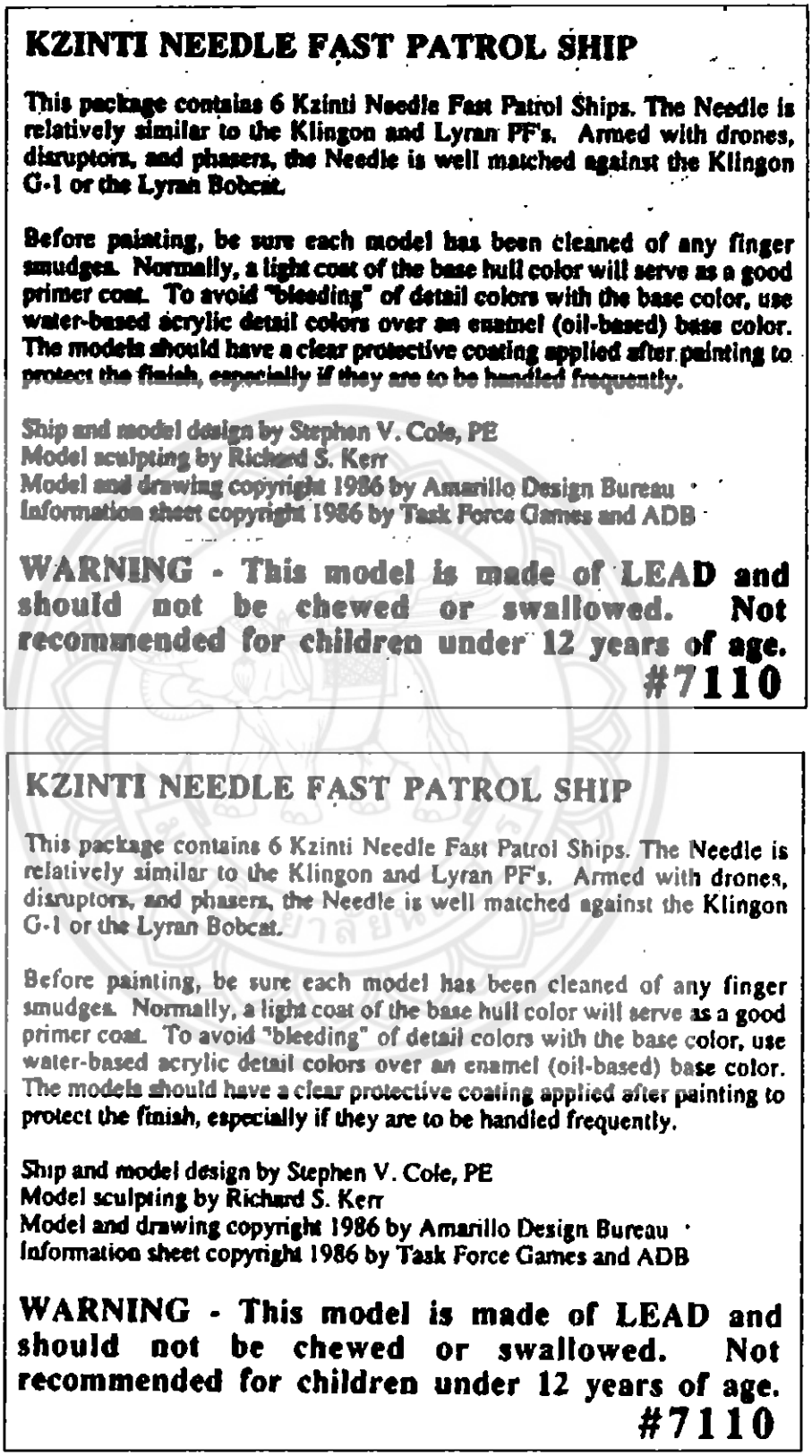
Ship and model design by Stephan V. Cole, PE
Model sculpting by Richard S. Kerr
Model and drawing copyright 1986 by Amarillo Design Bureau
Information sheet copyright 1986 by Task Force Games and ADB

WARNING - This model is made of LEAD and should not be chewed or swallowed. Not recommended for children under 12 years of age.

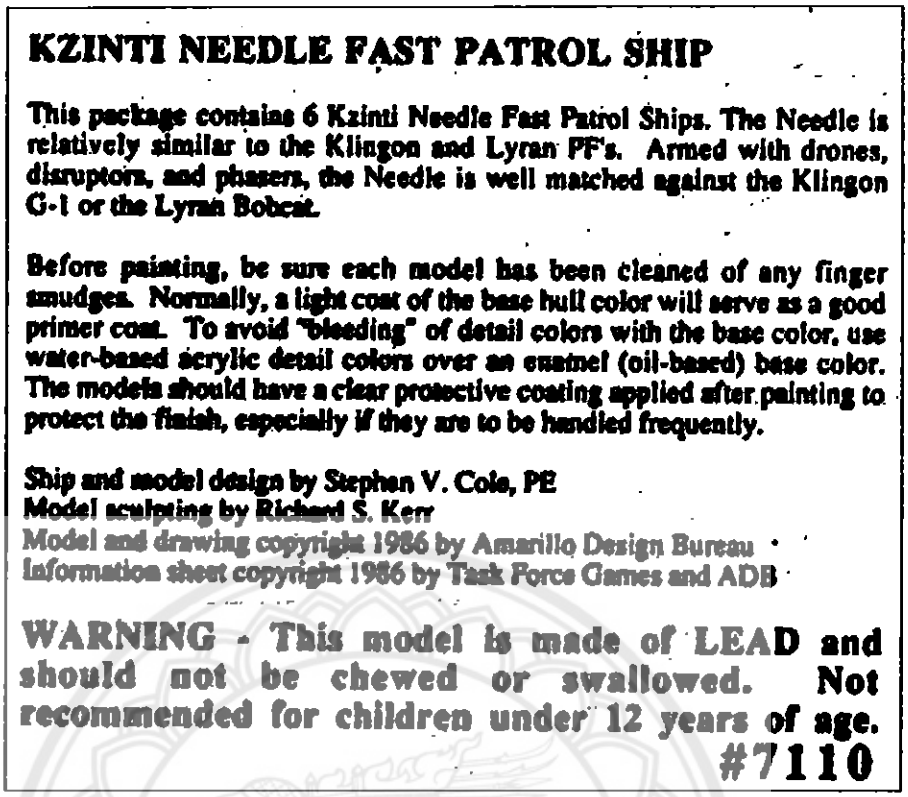
#7110

รูปที่ 4.41 ภาพต้นฉบับของตัวอักษรหนาหรือติดกันแบบไม่มี Noise

4.3.1 ผลลัพธ์จากการใช้ Erosion



รูปที่ 4.42 ภาพต้นฉบับของตัวอักษรหนาหรือติดกันแบบไม่มี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Erosion ด้วย Structure element (ล่าง)



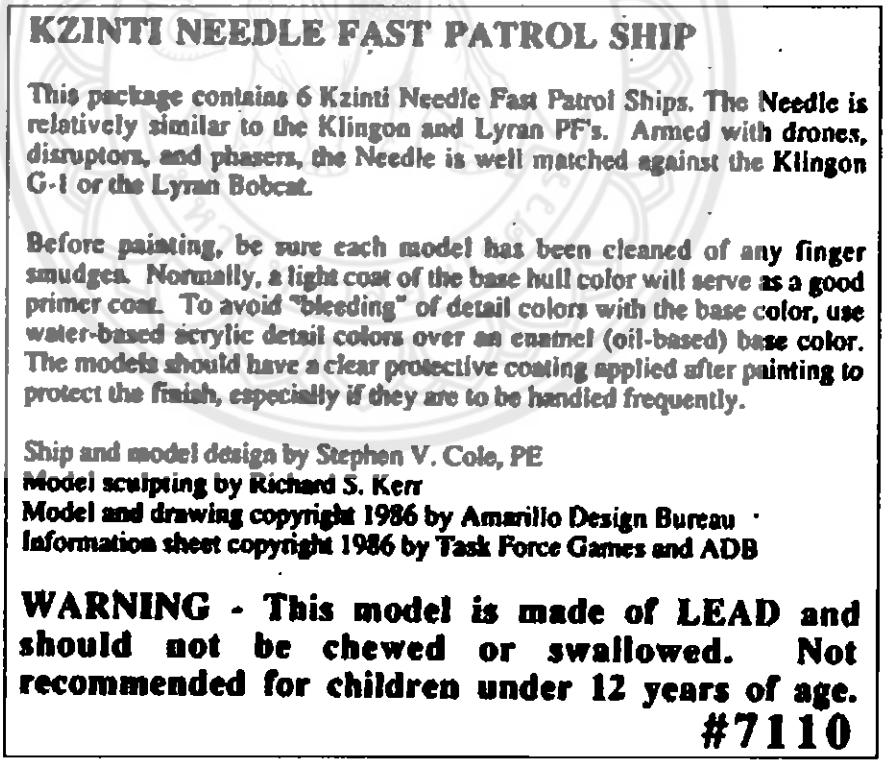
KZINTI NEEDLE FAST PATROL SHIP

This package contains 6 Kzinti Needle Fast Patrol Ships. The Needle is relatively similar to the Klingon and Lyran PF's. Armed with drones, disruptors, and phasers, the Needle is well matched against the Klingon G-1 or the Lyran Bobcat.

Before painting, be sure each model has been cleaned of any finger smudges. Normally, a light coat of the base hull color will serve as a good primer coat. To avoid "bleeding" of detail colors with the base color, use water-based acrylic detail colors over an enamel (oil-based) base color. The models should have a clear protective coating applied after painting to protect the finish, especially if they are to be handled frequently.

Ship and model design by Stephen V. Cole, PE
Model sculpting by Richard S. Kerr
Model and drawing copyright 1986 by Amarillo Design Bureau
Information sheet copyright 1986 by Task Force Games and ADB

WARNING - This model is made of LEAD and should not be chewed or swallowed. Not recommended for children under 12 years of age.
#7110



KZINTI NEEDLE FAST PATROL SHIP

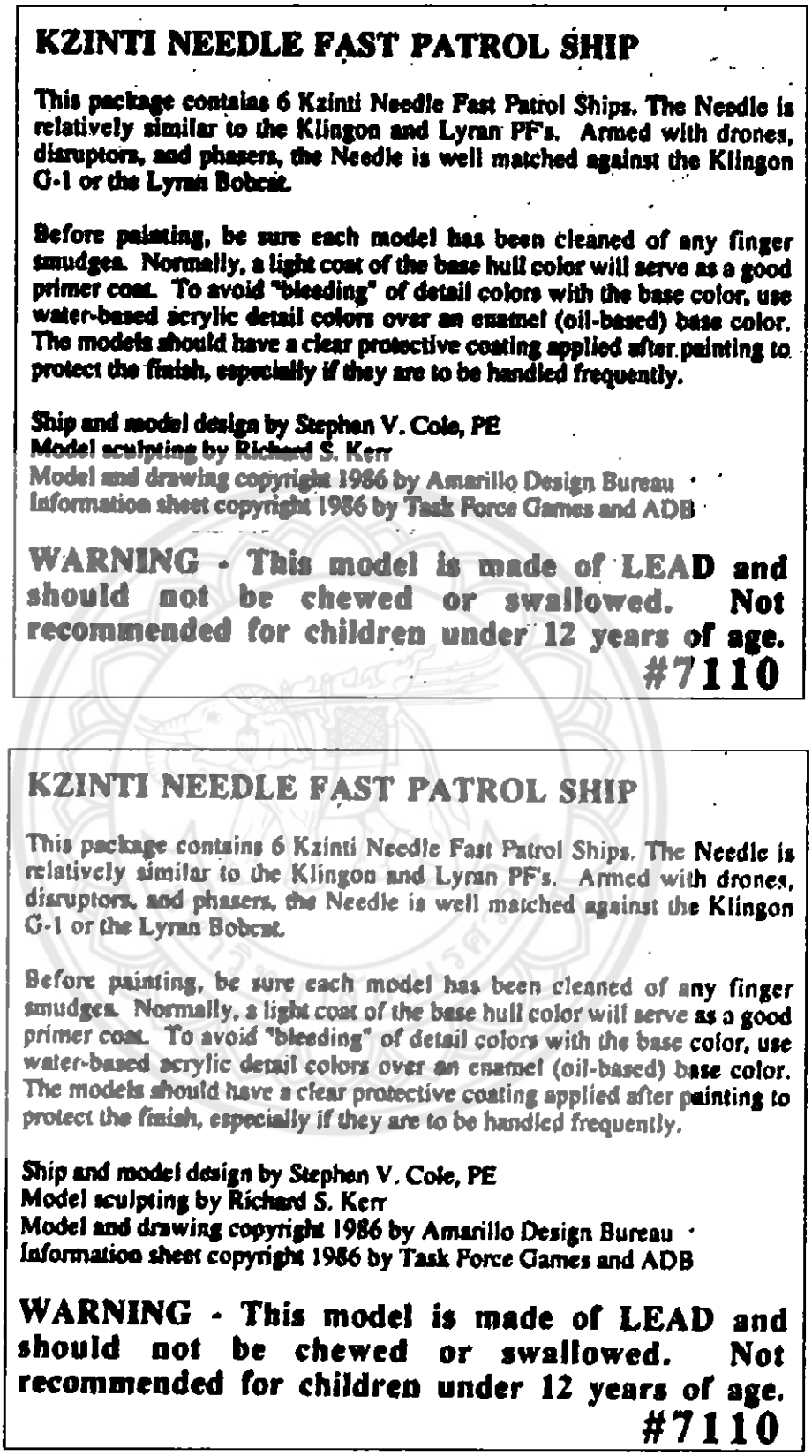
This package contains 6 Kzinti Needle Fast Patrol Ships. The Needle is relatively similar to the Klingon and Lyran PF's. Armed with drones, disruptors, and phasers, the Needle is well matched against the Klingon G-1 or the Lyran Bobcat.

Before painting, be sure each model has been cleaned of any finger smudges. Normally, a light coat of the base hull color will serve as a good primer coat. To avoid "bleeding" of detail colors with the base color, use water-based acrylic detail colors over an enamel (oil-based) base color. The models should have a clear protective coating applied after painting to protect the finish, especially if they are to be handled frequently.

Ship and model design by Stephen V. Cole, PE
Model sculpting by Richard S. Kerr
Model and drawing copyright 1986 by Amarillo Design Bureau
Information sheet copyright 1986 by Task Force Games and ADB

WARNING - This model is made of LEAD and should not be chewed or swallowed. Not recommended for children under 12 years of age.
#7110

รูปที่ 4.43 ภาพต้นฉบับของตัวอักษรหนาหรือติดกันแบบไม่มี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Erosion ด้วย Structure element (ล่าง)



KZINTI NEEDLE FAST PATROL SHIP

This package contains 6 Kzinti Needle Fast Patrol Ships. The Needle is relatively similar to the Klingon and Lyran PF's. Armed with drones, disruptors, and phasers, the Needle is well matched against the Klingon G-1 or the Lyran Bobcat.

Before painting, be sure each model has been cleaned of any finger smudges. Normally, a light coat of the base hull color will serve as a good primer coat. To avoid "bleeding" of detail colors with the base color, use water-based acrylic detail colors over an enamel (oil-based) base color. The models should have a clear protective coating applied after painting to protect the finish, especially if they are to be handled frequently.

Ship and model design by Stephen V. Cole, PE
Model sculpting by Richard S. Kerr
Model and drawing copyright 1986 by Amarillo Design Bureau
Information sheet copyright 1986 by Task Force Games and ADB

WARNING - This model is made of LEAD and should not be chewed or swallowed. Not recommended for children under 12 years of age.

#7110

KZINTI NEEDLE FAST PATROL SHIP

This package contains 6 Kzinti Needle Fast Patrol Ships. The Needle is relatively similar to the Klingon and Lyran PF's. Armed with drones, disruptors, and phasers, the Needle is well matched against the Klingon G-1 or the Lyran Bobcat.

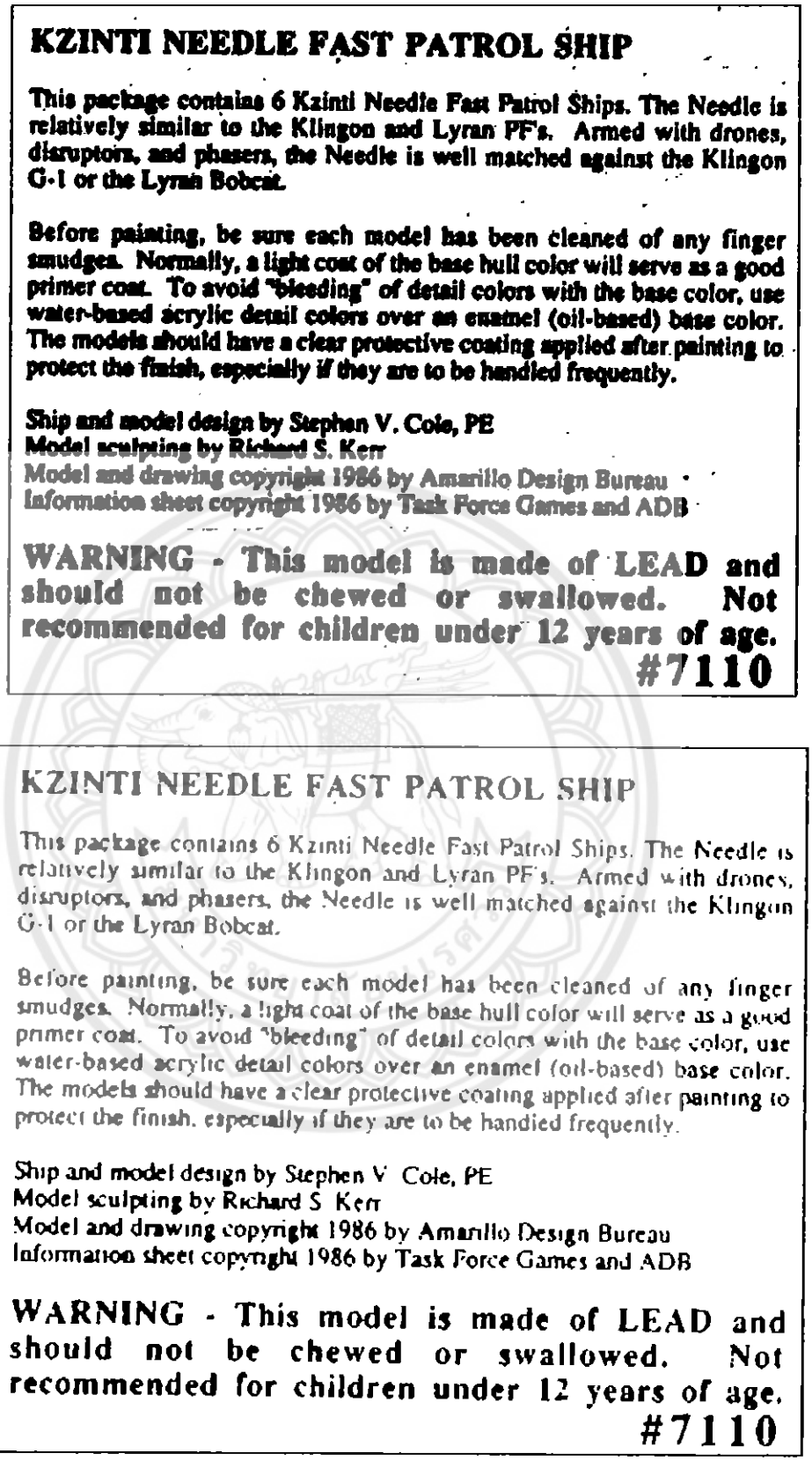
Before painting, be sure each model has been cleaned of any finger smudges. Normally, a light coat of the base hull color will serve as a good primer coat. To avoid "bleeding" of detail colors with the base color, use water-based acrylic detail colors over an enamel (oil-based) base color. The models should have a clear protective coating applied after painting to protect the finish, especially if they are to be handled frequently.

Ship and model design by Stephen V. Cole, PE
Model sculpting by Richard S. Kerr
Model and drawing copyright 1986 by Amarillo Design Bureau
Information sheet copyright 1986 by Task Force Games and ADB

WARNING - This model is made of LEAD and should not be chewed or swallowed. Not recommended for children under 12 years of age.

#7110

รูปที่ 4.44 ภาพต้นฉบับของตัวอักษรหนาหรือติดกันแบบไม่มี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Erosion ด้วย Structure element (ล่าง)



KZINTI NEEDLE FAST PATROL SHIP

This package contains 6 Kzinti Needle Fast Patrol Ships. The Needle is relatively similar to the Klingon and Lyran PF's. Armed with drones, disruptors, and phasers, the Needle is well matched against the Klingon G-1 or the Lyran Bobcat.

Before painting, be sure each model has been cleaned of any finger smudges. Normally, a light coat of the base hull color will serve as a good primer coat. To avoid "bleeding" of detail colors with the base color, use water-based acrylic detail colors over an enamel (oil-based) base color. The models should have a clear protective coating applied after painting to protect the finish, especially if they are to be handled frequently.

Ship and model design by Stephen V. Cole, PE
Model sculpting by Richard S. Kerr
Model and drawing copyright 1986 by Amarillo Design Bureau
Information sheet copyright 1986 by Task Force Games and ADB

WARNING - This model is made of LEAD and should not be chewed or swallowed. Not recommended for children under 12 years of age.

#7110

KZINTI NEEDLE FAST PATROL SHIP

This package contains 6 Kzinti Needle Fast Patrol Ships. The Needle is relatively similar to the Klingon and Lyran PF's. Armed with drones, disruptors, and phasers, the Needle is well matched against the Klingon G-1 or the Lyran Bobcat.

Before painting, be sure each model has been cleaned of any finger smudges. Normally, a light coat of the base hull color will serve as a good primer coat. To avoid "bleeding" of detail colors with the base color, use water-based acrylic detail colors over an enamel (oil-based) base color. The models should have a clear protective coating applied after painting to protect the finish, especially if they are to be handled frequently.

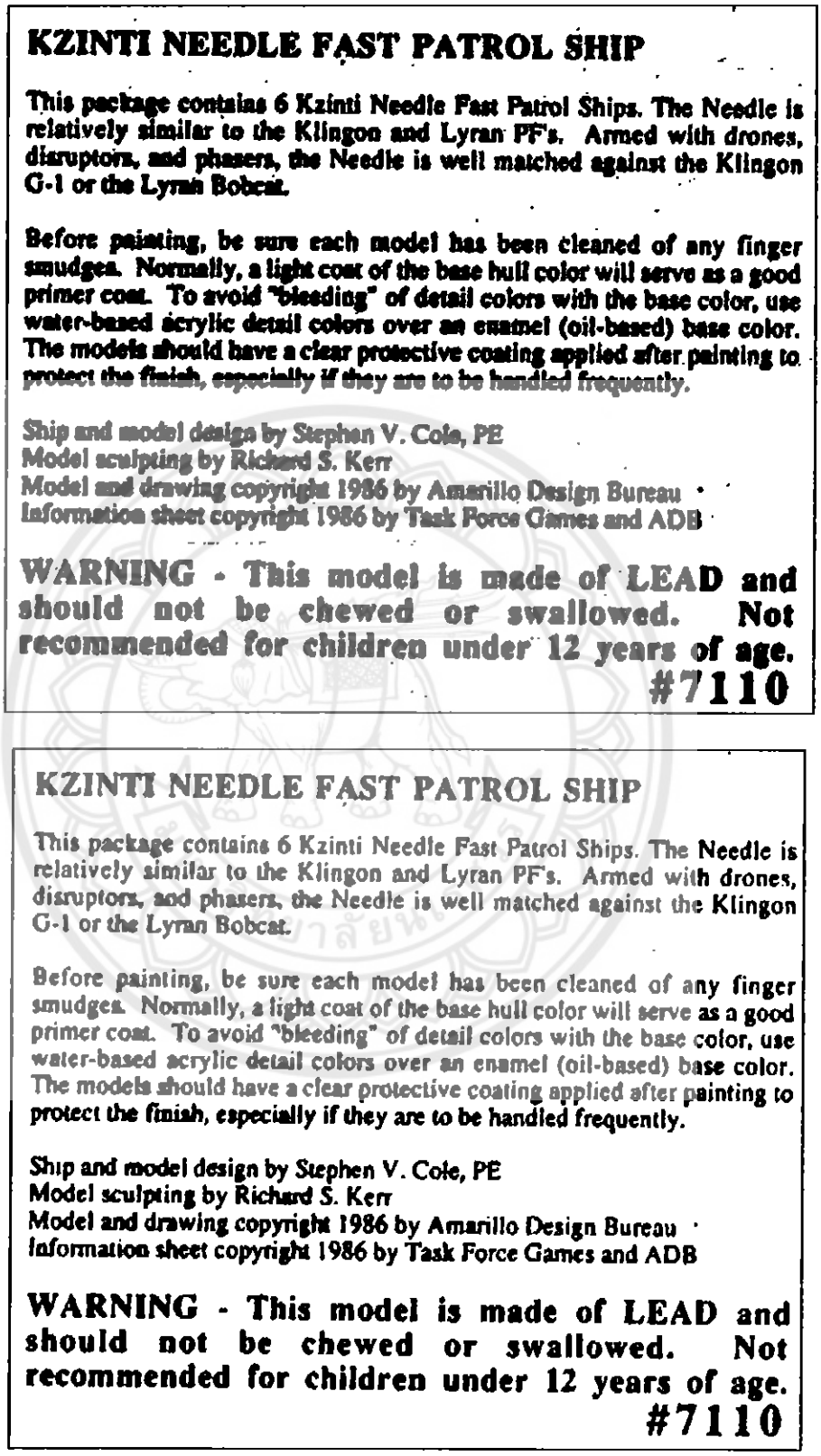
Ship and model design by Stephen V Cole, PE
Model sculpting by Richard S Kerr
Model and drawing copyright 1986 by Amarillo Design Bureau
Information sheet copyright 1986 by Task Force Games and ADB

WARNING - This model is made of LEAD and should not be chewed or swallowed. Not recommended for children under 12 years of age.

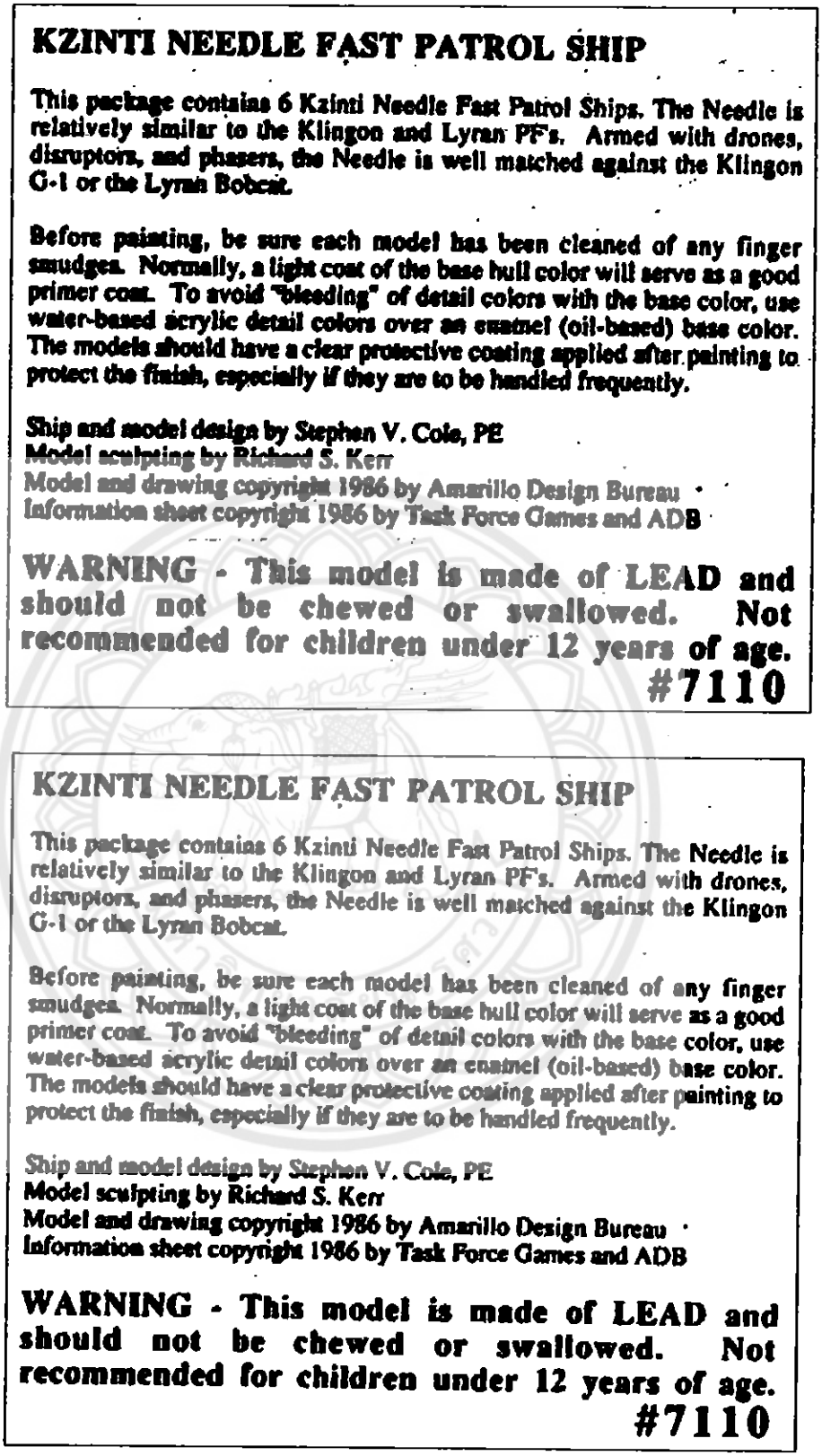
#7110

รูปที่ 4.45 ภาพต้นฉบับของตัวอักษรหนาหรือติดกันแบบไม่มี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Erosion ด้วย Structure element ⊞ (ล่าง)

4.3.2 ผลลัพธ์จากการใช้ Opening + Erosion



รูปที่ 4.46 ภาพต้นฉบับของตัวอักษรหนาหรือติดกันแบบไม่มี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Opening + Erosion ด้วย Structure element (ล่าง)



KZINTI NEEDLE FAST PATROL SHIP

This package contains 6 Kzinti Needle Fast Patrol Ships. The Needle is relatively similar to the Klingon and Lyran PF's. Armed with drones, disruptors, and phasers, the Needle is well matched against the Klingon G-1 or the Lyran Bobcat.

Before painting, be sure each model has been cleaned of any finger smudges. Normally, a light coat of the base hull color will serve as a good primer coat. To avoid "bleeding" of detail colors with the base color, use water-based acrylic detail colors over an enamel (oil-based) base color. The models should have a clear protective coating applied after painting to protect the finish, especially if they are to be handled frequently.

Ship and model design by Stephen V. Cole, PE
Model sculpting by Richard S. Kerr
Model and drawing copyright 1986 by Amarillo Design Bureau
Information sheet copyright 1986 by Task Force Games and ADB

WARNING - This model is made of LEAD and should not be chewed or swallowed. Not recommended for children under 12 years of age.

#7110

KZINTI NEEDLE FAST PATROL SHIP

This package contains 6 Kzinti Needle Fast Patrol Ships. The Needle is relatively similar to the Klingon and Lyran PF's. Armed with drones, disruptors, and phasers, the Needle is well matched against the Klingon G-1 or the Lyran Bobcat.

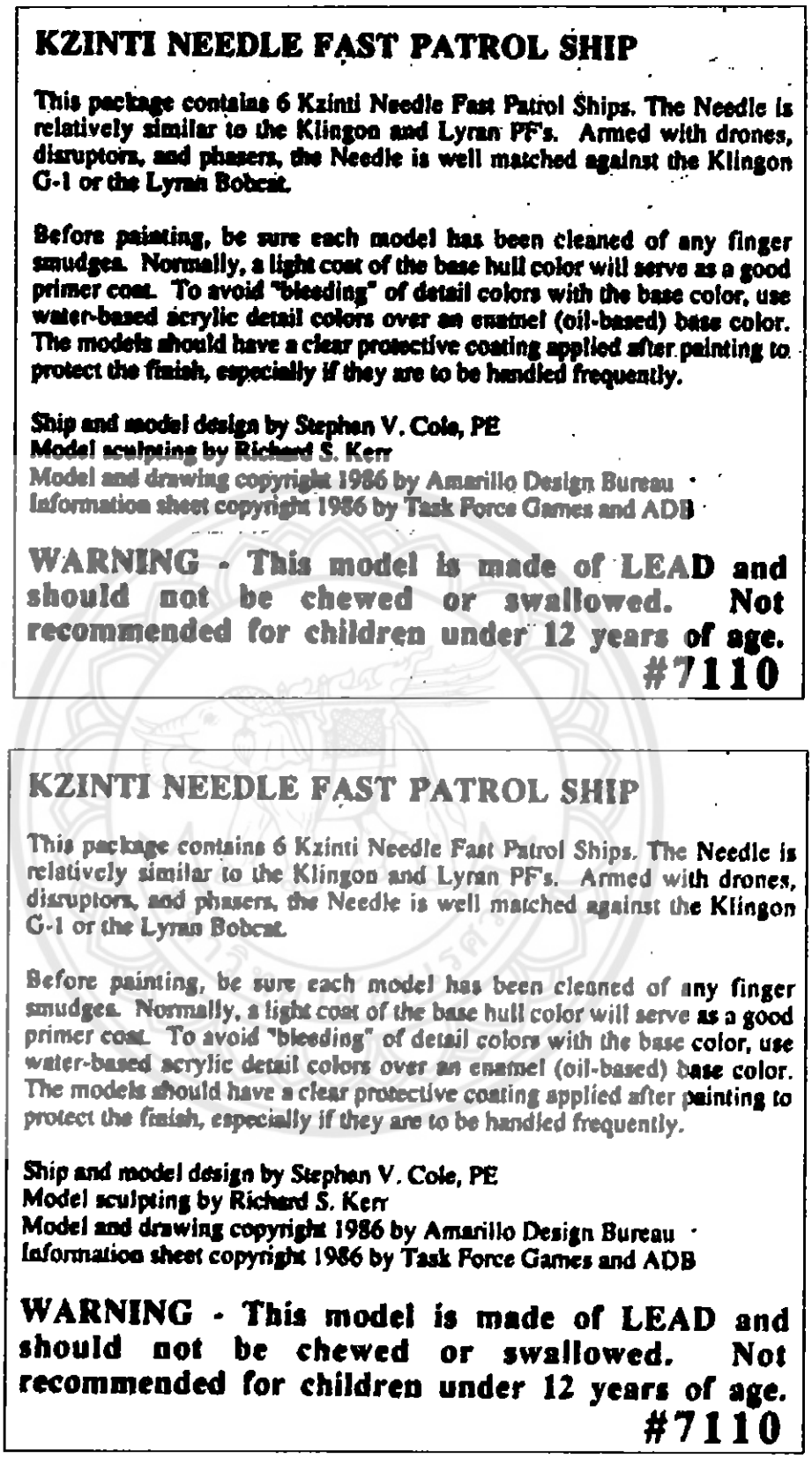
Before painting, be sure each model has been cleaned of any finger smudges. Normally, a light coat of the base hull color will serve as a good primer coat. To avoid "bleeding" of detail colors with the base color, use water-based acrylic detail colors over an enamel (oil-based) base color. The models should have a clear protective coating applied after painting to protect the finish, especially if they are to be handled frequently.

Ship and model design by Stephen V. Cole, PE
Model sculpting by Richard S. Kerr
Model and drawing copyright 1986 by Amarillo Design Bureau
Information sheet copyright 1986 by Task Force Games and ADB

WARNING - This model is made of LEAD and should not be chewed or swallowed. Not recommended for children under 12 years of age.

#7110

รูปที่ 4.47 ภาพต้นฉบับของตัวอักษรหนาหรือติดกันแบบไม่มี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Opening + Erosion ด้วย Structure element $\begin{smallmatrix}1\\1\end{smallmatrix}$ (ล่าง)



KZINTI NEEDLE FAST PATROL SHIP

This package contains 6 Kzinti Needle Fast Patrol Ships. The Needle is relatively similar to the Klingon and Lyran PF's. Armed with drones, disruptors, and phasers, the Needle is well matched against the Klingon G-1 or the Lyran Bobcat.

Before painting, be sure each model has been cleaned of any finger smudges. Normally, a light coat of the base hull color will serve as a good primer coat. To avoid "bleeding" of detail colors with the base color, use water-based acrylic detail colors over an enamel (oil-based) base color. The models should have a clear protective coating applied after painting to protect the finish, especially if they are to be handled frequently.

Ship and model design by Stephen V. Cole, PE
Model sculpting by Richard S. Kerr
Model and drawing copyright 1986 by Amarillo Design Bureau
Information sheet copyright 1986 by Task Force Games and ADB

WARNING - This model is made of LEAD and should not be chewed or swallowed. Not recommended for children under 12 years of age.

#7110

KZINTI NEEDLE FAST PATROL SHIP

This package contains 6 Kzinti Needle Fast Patrol Ships. The Needle is relatively similar to the Klingon and Lyran PF's. Armed with drones, disruptors, and phasers, the Needle is well matched against the Klingon G-1 or the Lyran Bobcat.

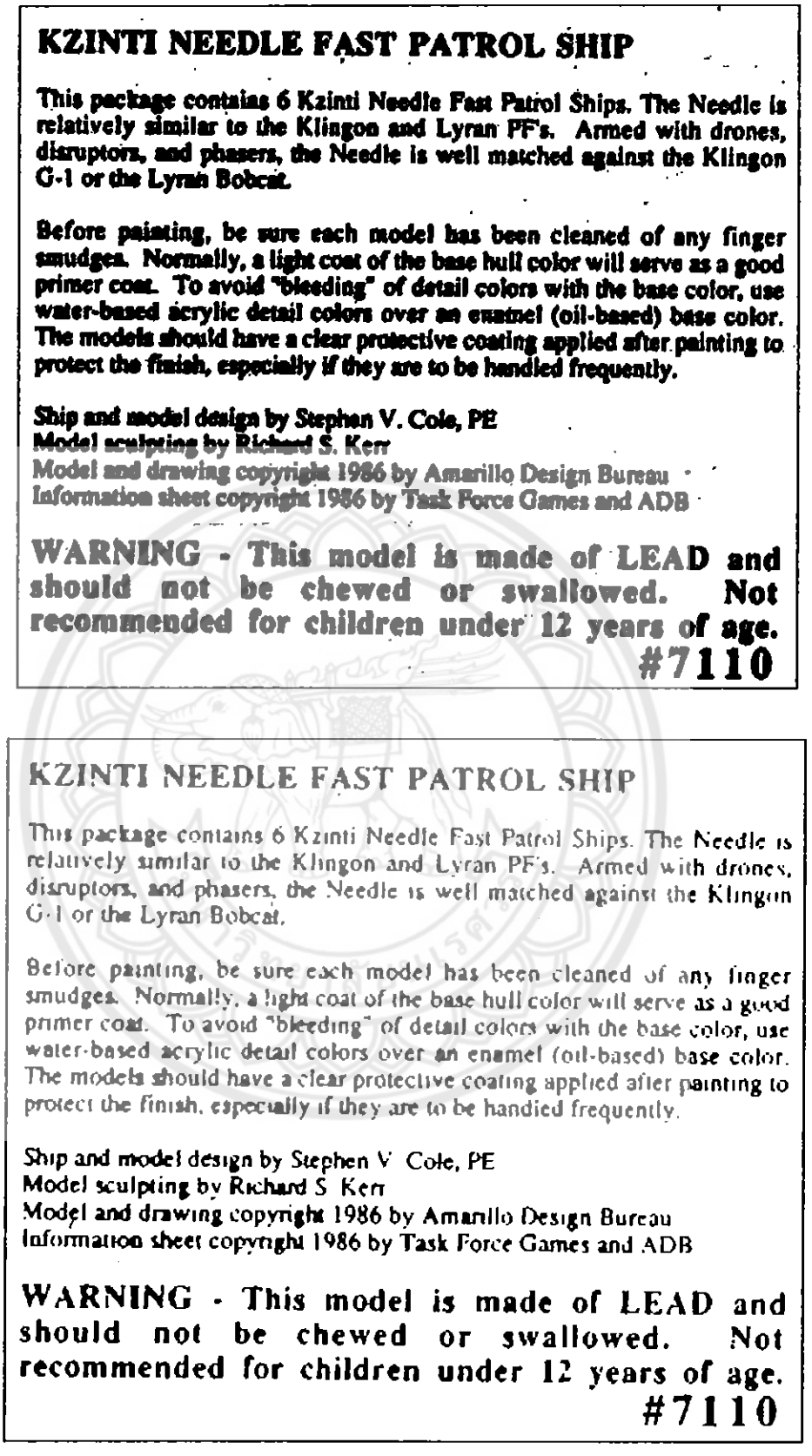
Before painting, be sure each model has been cleaned of any finger smudges. Normally, a light coat of the base hull color will serve as a good primer coat. To avoid "bleeding" of detail colors with the base color, use water-based acrylic detail colors over an enamel (oil-based) base color. The models should have a clear protective coating applied after painting to protect the finish, especially if they are to be handled frequently.

Ship and model design by Stephen V. Cole, PE
Model sculpting by Richard S. Kerr
Model and drawing copyright 1986 by Amarillo Design Bureau
Information sheet copyright 1986 by Task Force Games and ADB

WARNING - This model is made of LEAD and should not be chewed or swallowed. Not recommended for children under 12 years of age.

#7110

รูปที่ 4.48 ภาพต้นฉบับของตัวอักษรหนาหรือติดกันแบบไม่มี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Opening + Erosion ด้วย Structure element (ล่าง)



KZINTI NEEDLE FAST PATROL SHIP

This package contains 6 Kzinti Needle Fast Patrol Ships. The Needle is relatively similar to the Klingon and Lyran PF's. Armed with drones, disruptors, and phasers, the Needle is well matched against the Klingon G-1 or the Lyran Bobcat.

Before painting, be sure each model has been cleaned of any finger smudges. Normally, a light coat of the base hull color will serve as a good primer coat. To avoid "bleeding" of detail colors with the base color, use water-based acrylic detail colors over an enamel (oil-based) base color. The models should have a clear protective coating applied after painting to protect the finish, especially if they are to be handled frequently.

Ship and model design by Stephen V. Cole, PE
Model sculpting by Richard S. Kerr
Model and drawing copyright 1986 by Amarillo Design Bureau
Information sheet copyright 1986 by Task Force Games and ADB

WARNING - This model is made of LEAD and should not be chewed or swallowed. Not recommended for children under 12 years of age.

#7110

KZINTI NEEDLE FAST PATROL SHIP

This package contains 6 Kzinti Needle Fast Patrol Ships. The Needle is relatively similar to the Klingon and Lyran PF's. Armed with drones, disruptors, and phasers, the Needle is well matched against the Klingon G-1 or the Lyran Bobcat.

Before painting, be sure each model has been cleaned of any finger smudges. Normally, a light coat of the base hull color will serve as a good primer coat. To avoid "bleeding" of detail colors with the base color, use water-based acrylic detail colors over an enamel (oil-based) base color. The models should have a clear protective coating applied after painting to protect the finish, especially if they are to be handled frequently.

Ship and model design by Stephen V Cole, PE
Model sculpting by Richard S Kerr
Model and drawing copyright 1986 by Amarillo Design Bureau
Information sheet copyright 1986 by Task Force Games and ADB

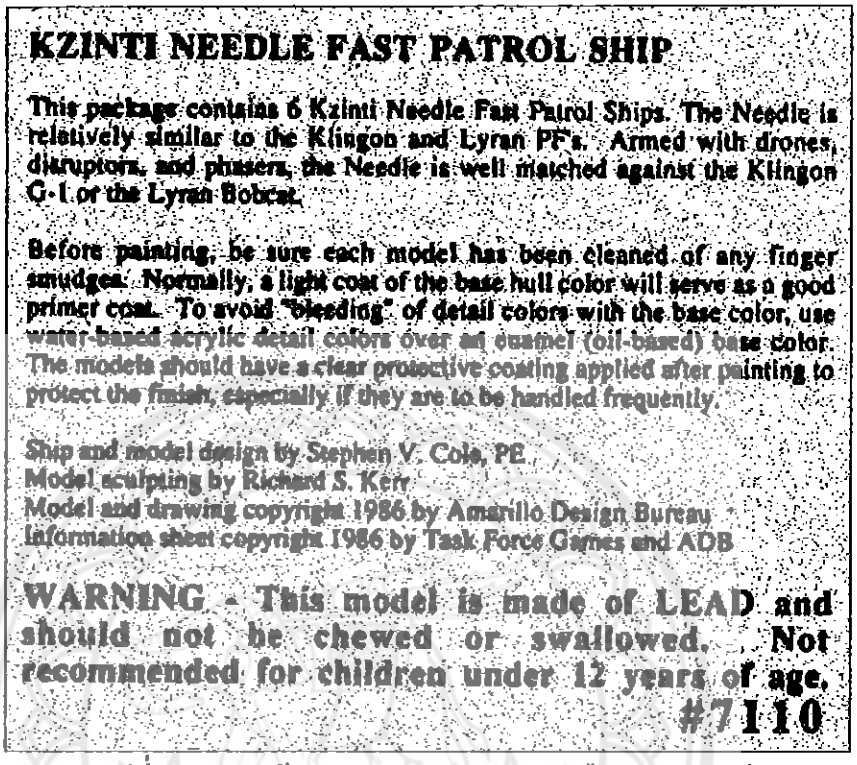
WARNING - This model is made of LEAD and should not be chewed or swallowed. Not recommended for children under 12 years of age.

#7110

รูปที่ 4.49 ภาพต้นฉบับของตัวอักษรหนาหรือติดกันแบบไม่มี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Opening + Erosion ด้วย Structure element ⊞ (ล่าง)

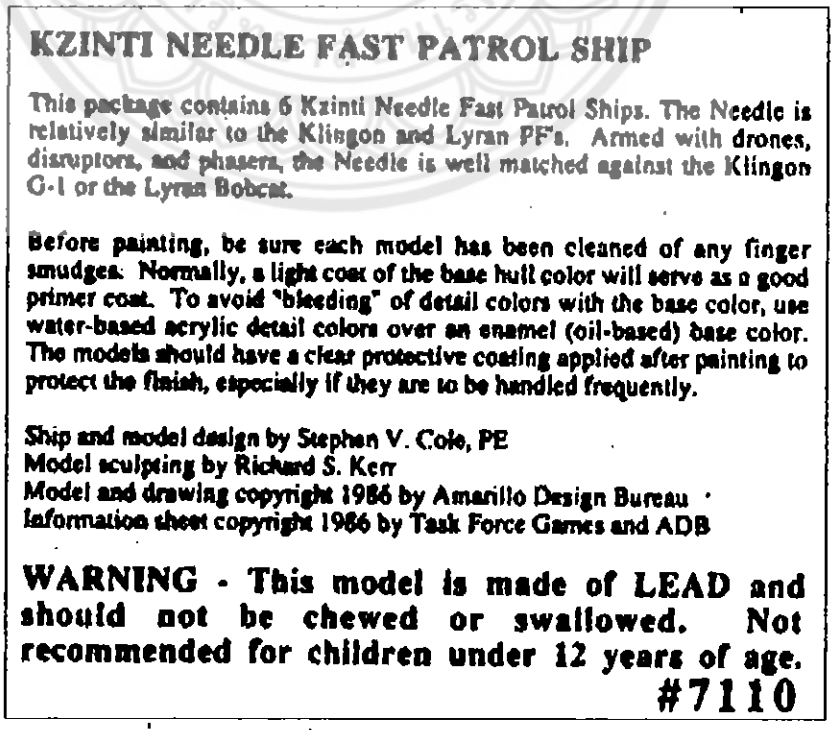
4.4 ผลการทดลองสำหรับภาพที่มีลักษณะตัวอักษรหนาหรือติดกันแบบมี noise

ข้อมูลภาพที่นำมาใช้ในการทดลอง : ภาพขาวดำขนาด 641x734 พิกเซล รูปแบบการจัดเก็บเป็น .bmp ใช้ noise ชนิด salt & pepper



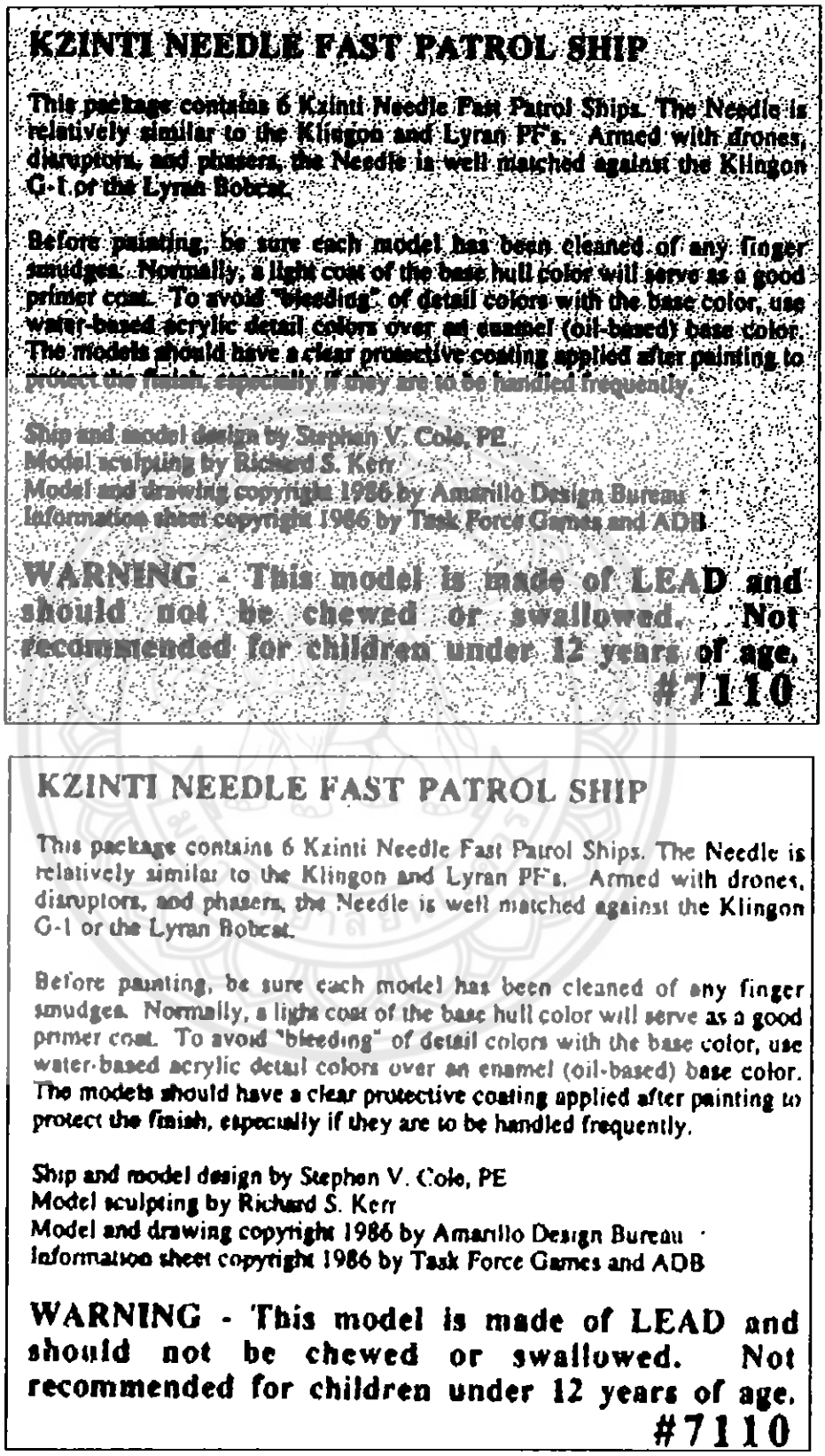
รูปที่ 4.50 ภาพต้นฉบับของตัวอักษรหนาหรือติดกันแบบมี Noise

ผลลัพธ์จากการลบ Noise โดยใช้ Median Filter

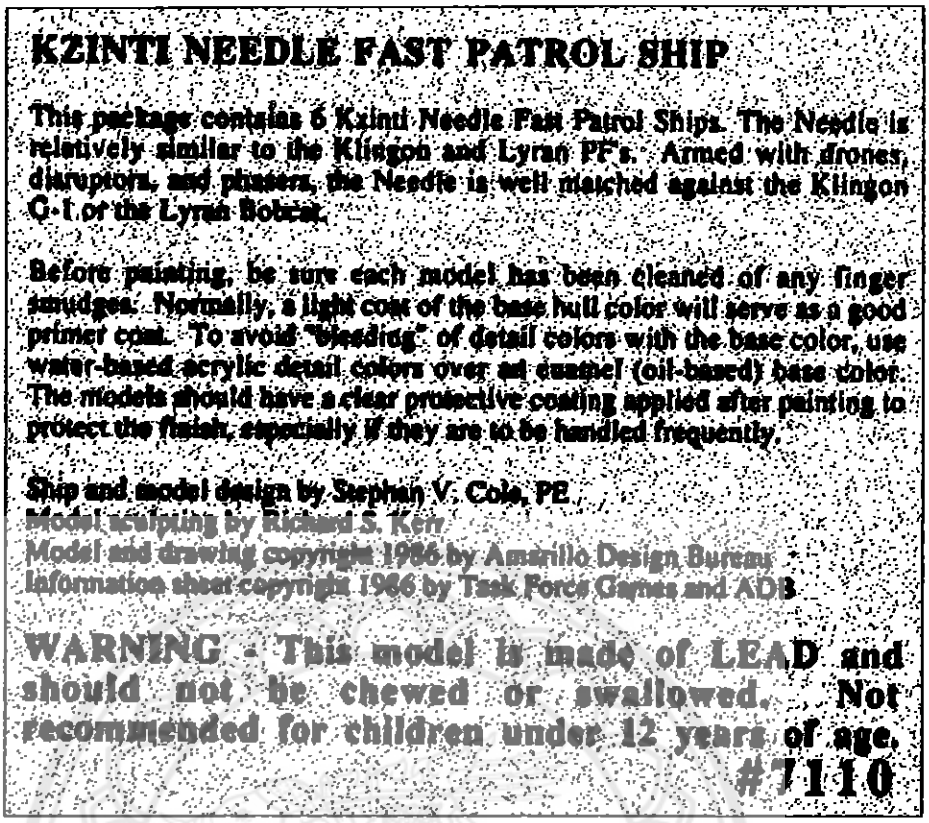


รูปที่ 4.51 ผลลัพธ์การลบ Noise โดยใช้ Median Filter

4.4.1 ผลลัพธ์จากการใช้ Erosion



รูปที่ 4.52 ต้นฉบับของตัวอักษรหนาหรือติดกันแบบมี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Erosion ด้วย Structure element (ล่าง)



KZINTI NEEDLE FAST PATROL SHIP

This package contains 6 Kzinti Needle Fast Patrol Ships. The Needle is relatively similar to the Klingon and Lyran PF's. Armed with drones, disruptors, and phasers, the Needle is well matched against the Klingon G-1 or the Lyran Bobcat.

Before painting, be sure each model has been cleaned of any finger smudges. Normally, a light coat of the base hull color will serve as a good primer coat. To avoid "bleeding" of detail colors with the base color, use water-based acrylic detail colors over an enamel (oil-based) base color. The models should have a clear protective coating applied after painting to protect the finish, especially if they are to be handled frequently.

Ship and model design by Stephen V. Cole, PE
Model sculpting by Richard S. Kerr
Model and drawing copyright 1986 by Amarillo Design Bureau
Information sheet copyright 1986 by Task Force Games and ADB

WARNING - This model is made of LEAD and should not be chewed or swallowed. Not recommended for children under 12 years of age.

#7110

KZINTI NEEDLE FAST PATROL SHIP

This package contains 6 Kzinti Needle Fast Patrol Ships. The Needle is relatively similar to the Klingon and Lyran PF's. Armed with drones, disruptors, and phasers, the Needle is well matched against the Klingon G-1 or the Lyran Bobcat.

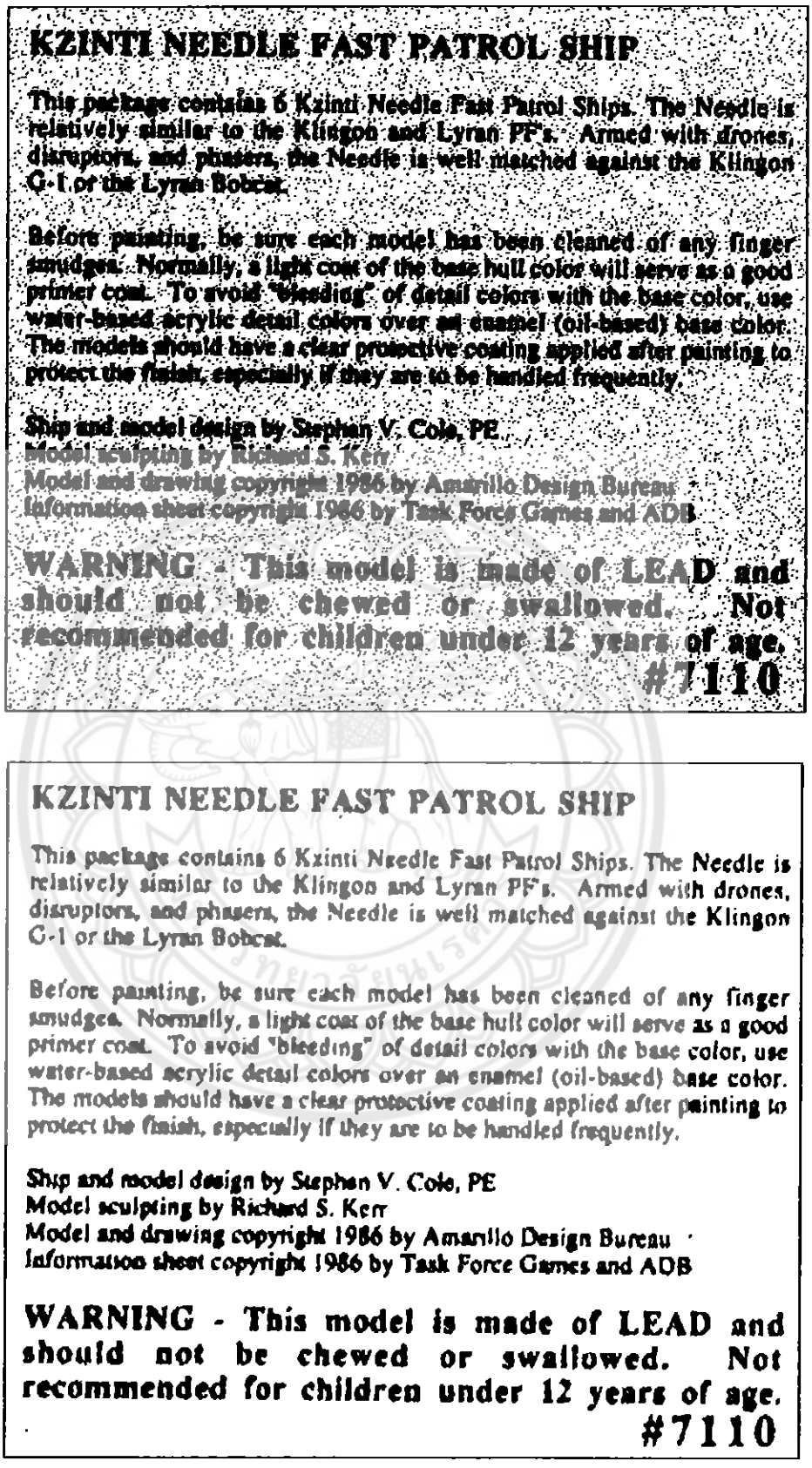
Before painting, be sure each model has been cleaned of any finger smudges. Normally, a light coat of the base hull color will serve as a good primer coat. To avoid "bleeding" of detail colors with the base color, use water-based acrylic detail colors over an enamel (oil-based) base color. The models should have a clear protective coating applied after painting to protect the finish, especially if they are to be handled frequently.

Ship and model design by Stephen V. Cole, PE
Model sculpting by Richard S. Kerr
Model and drawing copyright 1986 by Amarillo Design Bureau
Information sheet copyright 1986 by Task Force Games and ADB

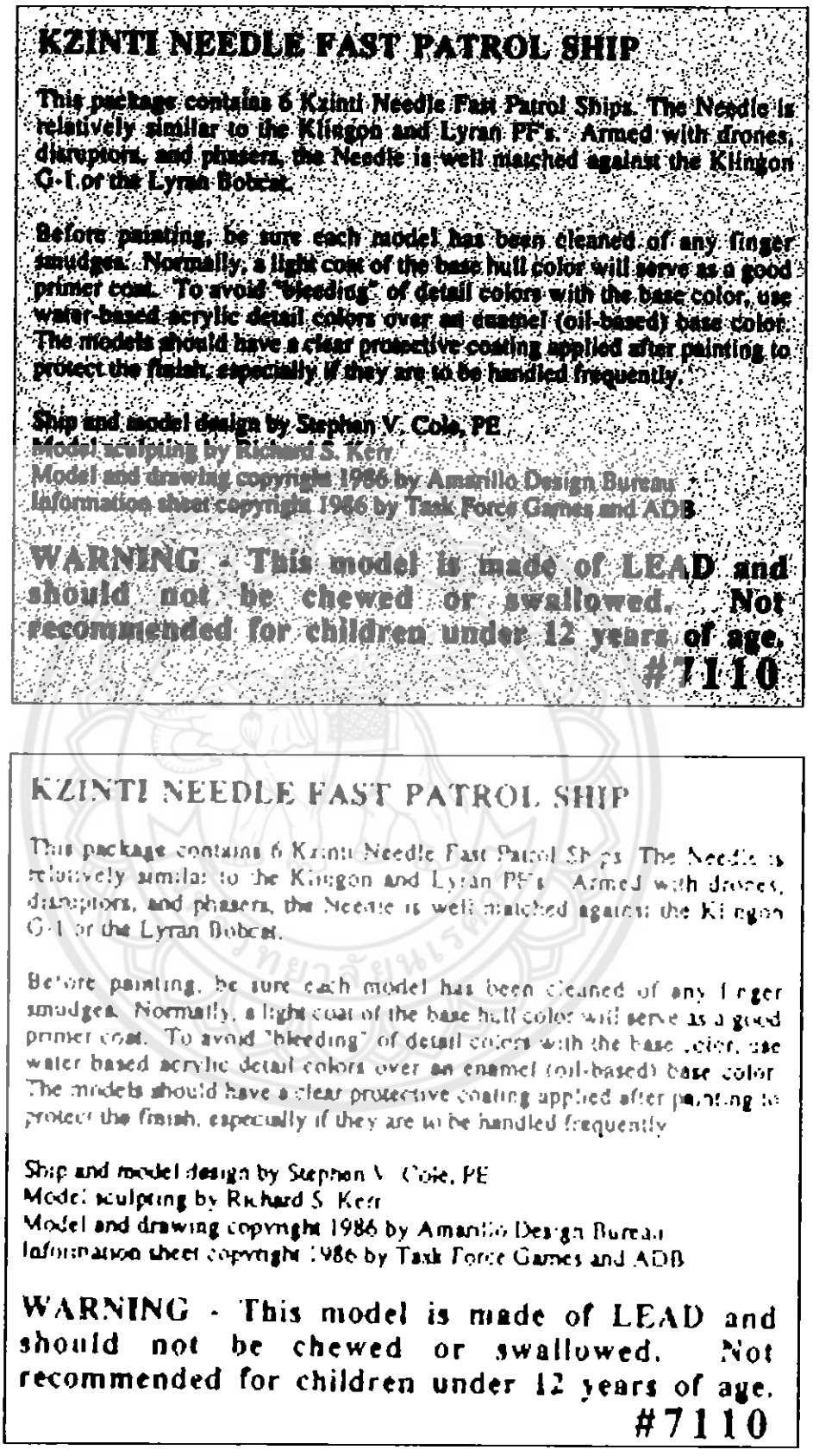
WARNING - This model is made of LEAD and should not be chewed or swallowed. Not recommended for children under 12 years of age.

#7110

รูปที่ 4.53 ต้นฉบับของตัวอักษรหนาหรือติดกันแบบมี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Erosion ด้วย Structure element (ล่าง)

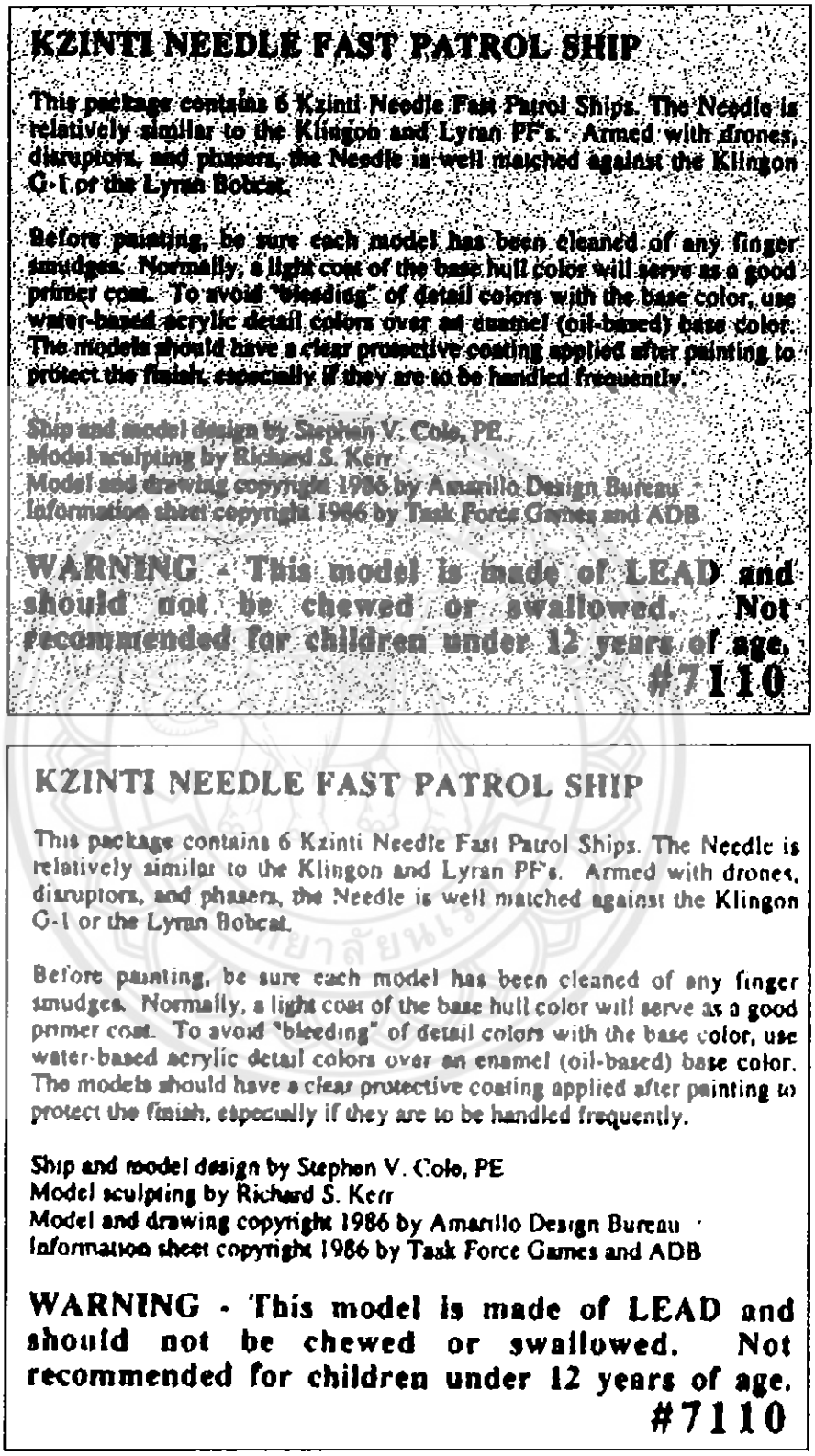


รูปที่ 4.54 ต้นฉบับของตัวอักษรหนาหรือติดกันแบบมี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Erosion ด้วย Structure element ⊞ (ล่าง)

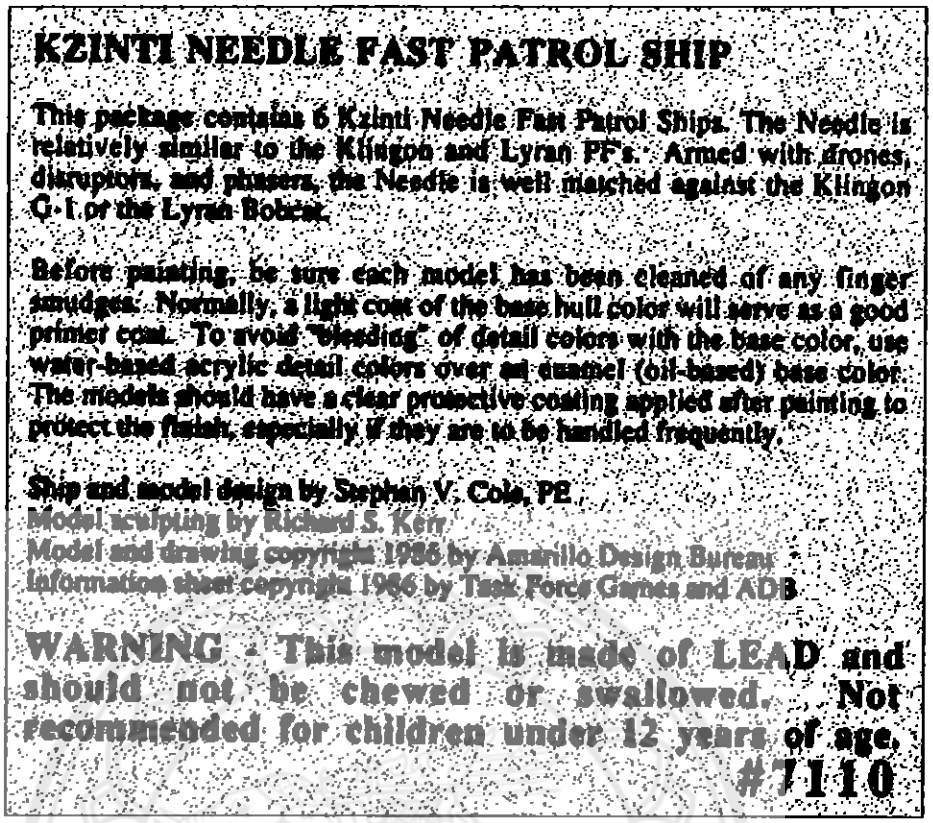


รูปที่ 4.55 ต้นฉบับของตัวอักษรหนาหรือติดกันแบบมี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Erosion ด้วย Structure element ⊞ (ล่าง)

4.4.2 ผลลัพธ์จากการใช้ Opening + Erosion



รูปที่ 4.56 ต้นฉบับของตัวอักษรหนาหรือติดกันแบบมี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Opening + Erosion ด้วย Structure element ⊞ (ล่าง)



KZINTI NEEDLE FAST PATROL SHIP

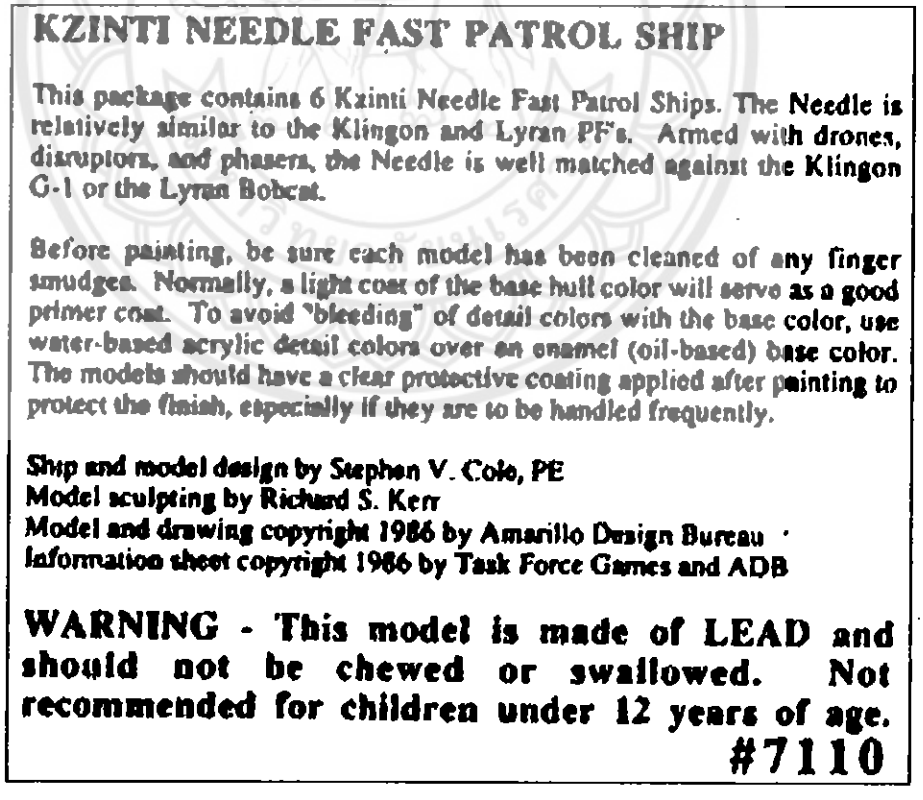
This package contains 6 Kzinti Needle Fast Patrol Ships. The Needle is relatively similar to the Klingon and Lyran PF's. Armed with drones, disruptors, and phasers, the Needle is well matched against the Klingon G-1 or the Lyran Bobcat.

Before painting, be sure each model has been cleaned of any finger smudges. Normally, a light coat of the base hull color will serve as a good primer coat. To avoid "bleeding" of detail colors with the base color, use water-based acrylic detail colors over an enamel (oil-based) base color. The models should have a clear protective coating applied after painting to protect the finish, especially if they are to be handled frequently.

Ship and model design by Stephen V. Cole, PE
Model sculpting by Richard S. Kerr
Model and drawing copyright 1986 by Amarillo Design Bureau
Information sheet copyright 1986 by Task Force Games and ADB

WARNING - This model is made of LEAD and should not be chewed or swallowed. Not recommended for children under 12 years of age.

#7110



KZINTI NEEDLE FAST PATROL SHIP

This package contains 6 Kzinti Needle Fast Patrol Ships. The Needle is relatively similar to the Klingon and Lyran PF's. Armed with drones, disruptors, and phasers, the Needle is well matched against the Klingon G-1 or the Lyran Bobcat.

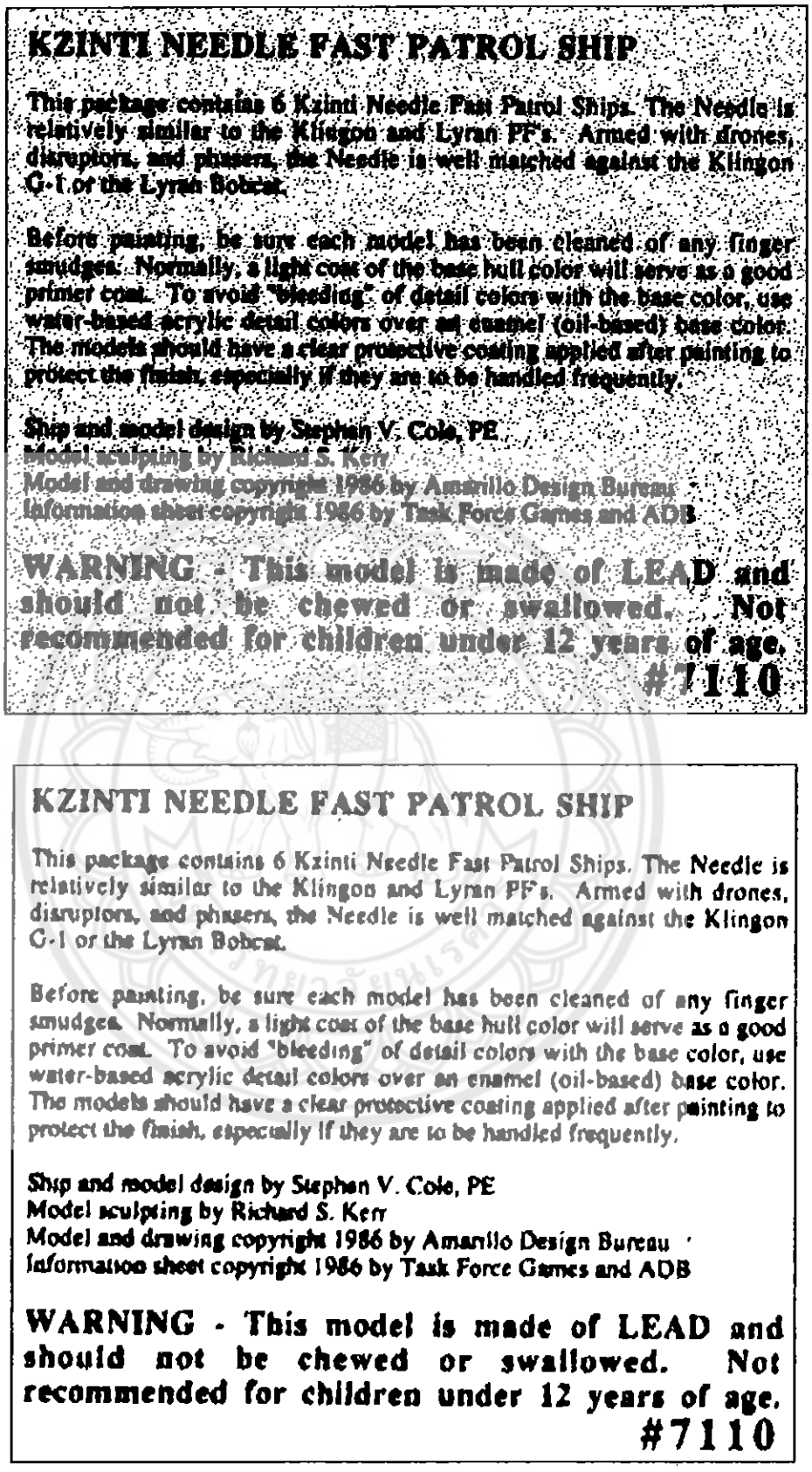
Before painting, be sure each model has been cleaned of any finger smudges. Normally, a light coat of the base hull color will serve as a good primer coat. To avoid "bleeding" of detail colors with the base color, use water-based acrylic detail colors over an enamel (oil-based) base color. The models should have a clear protective coating applied after painting to protect the finish, especially if they are to be handled frequently.

Ship and model design by Stephen V. Cole, PE
Model sculpting by Richard S. Kerr
Model and drawing copyright 1986 by Amarillo Design Bureau
Information sheet copyright 1986 by Task Force Games and ADB

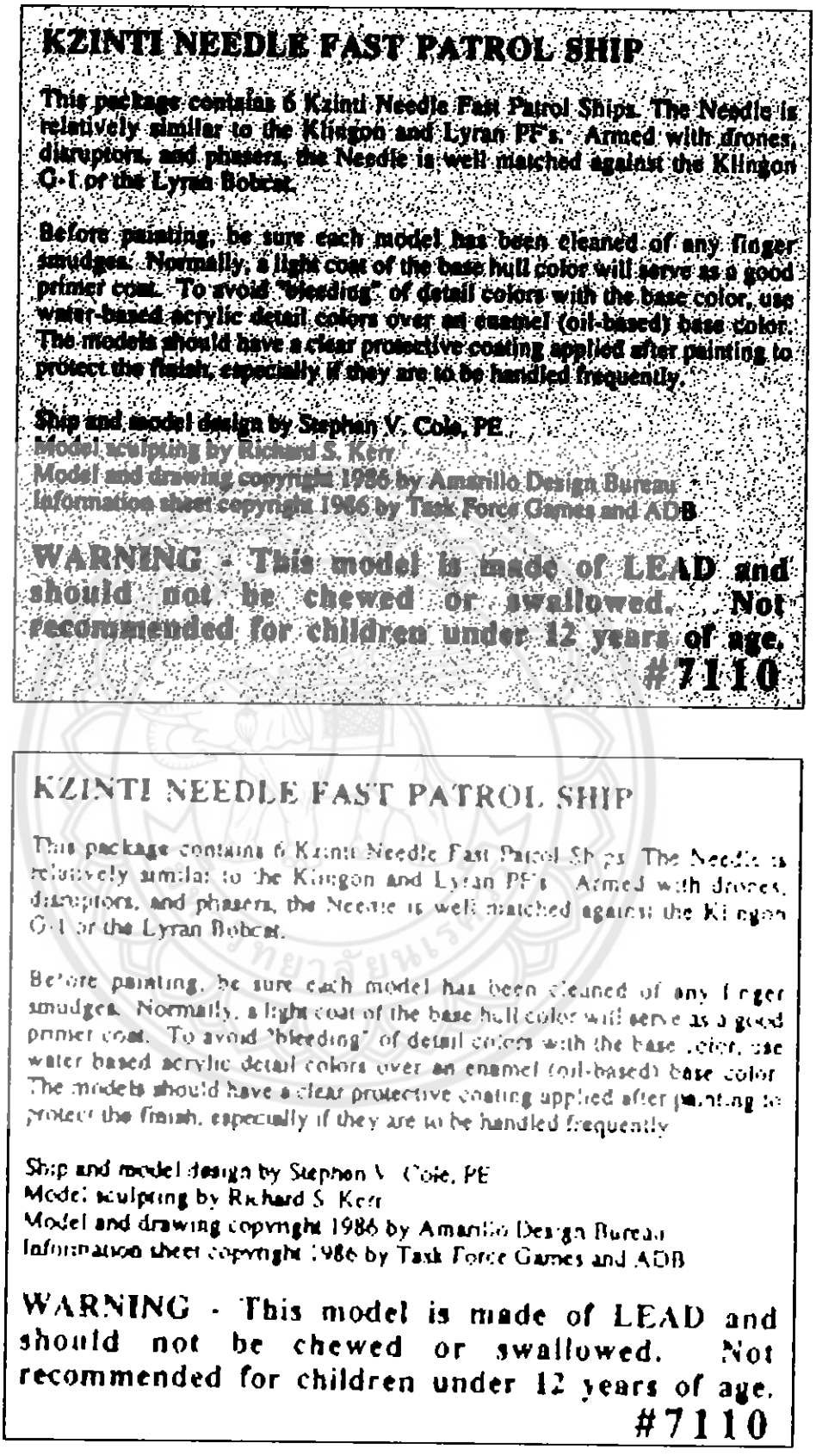
WARNING - This model is made of LEAD and should not be chewed or swallowed. Not recommended for children under 12 years of age.

#7110

รูปที่ 4.57 ต้นฉบับของตัวอักษรหนาหรือติดกันแบบมี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Opening + Erosion ด้วย Structure element ⊟ (ล่าง)



รูปที่ 4.58 ต้นฉบับของตัวอักษรหนาหรือติดกันแบบมี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Opening + Erosion ด้วย Structure element ⊞ (ล่าง)



รูปที่ 4.59 ต้นฉบับของตัวอักษรหนาหรือติดกันแบบมี Noise (บน) เทียบกับผลลัพธ์จากการใช้ Opening + Erosion ด้วย Structure element ⊞ (ล่าง)

4.5 วิเคราะห์ผลการทดลอง

4.5.1 สำหรับภาพตัวอักษรบางหรือเลือนราง

- Structure element แบบที่1

1. เมื่อทำการ Dilation จะทำให้เส้นของตัวอักษรจะหนาขึ้นทั้งเส้นแนวนอน เส้นแนวตั้งและ เส้นแนวทแยง

2. เมื่อทำการ Closing จะทำให้เส้นของตัวอักษรที่ขาดออกจากกันไม่มากติดกัน

3. เมื่อทำการ Closing+Dilation จะทำให้ตัวหนังสือหนาขึ้นและเส้นของตัวอักษรที่ขาดออจากกันติดกันมาขึ้นกว่าการทำ Closing เพียงอย่างเดียว

- Structure element แบบที่ 2

1. เมื่อทำการ Dilation จะทำให้เส้นในแนวนอนหนากว่าเส้นในแนวตั้ง แต่ส่วนเส้นตัวอักษรที่ขาดออกจากกันจะชิดกันมากขึ้น

2. เมื่อทำการ Closing จะทำให้เส้นของตัวอักษรในแนวตั้งติดกัน

3. เมื่อทำการ Closing + Dilation ลักษณะของเส้นในแนวตั้งจะติดกันมากกว่า structure element แบบที่ 1 ตัวอักษรก็จะหนามากขึ้น

- Structure element แบบที่ 3

1. เมื่อทำการ Dilation ให้ผลคล้ายกับ structure element แบบที่ 2 แต่ตัวอักษรจะมีลักษณะเป็นเหลี่ยมมากกว่า structure element แบบที่ 2

2. เมื่อทำการ Closing ให้ผลคล้ายกับ structure element แบบที่ 2 แต่ตัวอักษรจะมีลักษณะเป็นเหลี่ยมมากกว่า structure element แบบที่ 2

3. เมื่อทำการ Closing + Dilation ให้ผลคล้ายกับ structure element แบบที่ 2 แต่ตัวอักษรจะมีลักษณะเป็นเหลี่ยมมากกว่า structure element แบบที่ 2

- Structure element แบบที่ 4

1. เมื่อทำการ Dilation จะทำให้เล่นของตัวอักษรมีลักษณะหนามากทั้งในแนวตั้ง แนวนอน และแนวทแยง

2. เมื่อทำการ Closing จะทำให้บางส่วนของตัวอักษรติดกันจนเป็นจุดดำขนาดใหญ่

3. เมื่อทำการ Closing + Dilation จะทำให้ตัวอักษรหนามากและบางส่วนของตัวอักษรเป็นจุดสีดำขนาดใหญ่ ทำให้อ่านได้ยาก

- Structure element แบบที่ 5

1. เมื่อทำการ Dilation จะทำให้เส้นในแนวนอนหนามาก ถ้าเส้นในแนวนอนติดกันมากก็จะทำให้เส้นติดกัน เช่น ตัวอักอักษร ติดกับ สระ อี ในส่วนของเส้นตัวอักษรในแนว ตั้งที่ขาดออกจากกันก็จะติดกันมากกว่า structure element แบบที่ 1, แบบที่2 และแบบที่ 3

2. เมื่อทำการ Closing จะทำให้บางส่วนของตัวอักษรติดกันจนเป็นจุดดำแต่เล็กกว่า structure element แบบที่ 3

3. เมื่อทำการ Closing + Dilation จะทำให้เส้นในแนวตั้งเชื่อมติดกันได้ดี แต่ตัวอักษรกับวรรณยุกต์ยังติดกันอยู่ แต่ลักษณะของเส้นจะไม่เรียบ

- Structure element แบบที่ 6

1. เมื่อทำการ Dilation จะทำให้ตัวอักษรติดกับวรรณยุกต์มากกว่า Structure element แบบที่ 5 เส้นในแนวตั้งเชื่อมกันได้ดี

2. เมื่อทำการ Closing ผลที่ได้คล้ายกับ Structure element แบบที่ 5 แต่จุดดำจะใหญ่เล็กน้อย แต่เล็กกว่า Structure element แบบที่ 4

3. เมื่อทำการ Closing+ Dilation ลักษณะของเส้นจะหนากว่า Structure element แบบที่ 5 แต่จะเรียบกว่า Structure element แบบที่ 5

4.5.2 สำหรับตัวอักษรหนาหรือติดกัน

- Structure element แบบที่ 1

1. เมื่อทำการ Erosion จะทำให้ตัวอักษรบางลง

2. เมื่อทำการ Opening + Erosion มีลักษณะใกล้เคียงกับ Erosion

- Structure element แบบที่ 2

1. เมื่อทำการ Erosion ตัวอักษรมีลักษณะบางกว่าต้นฉบับแต่หนากว่า Structure element แบบที่ 1

2. เมื่อทำการ Opening + Erosion มีลักษณะใกล้เคียงกับ Erosion

- Structure element แบบที่ 3

1. เมื่อทำการ Erosion ตัวอักษรจะบางลงกว่า Structure element แบบที่ 1 แต่หนากว่า Structure element แบบที่ 1

2. เมื่อทำการ Opening + Erosion มีลักษณะใกล้เคียงกับ Erosion

- Structure element แบบที่ 4

1. เมื่อทำการ Erosion จะทำให้ตัวอักษรบางมากและเส้นของตัวอักษรบางเส้นขาด

2. เมื่อทำการ Opening + Erosion มีลักษณะใกล้เคียงกับ Erosion

บทที่ 5
สรุปผล

5.1 สรุปผลการทดลอง

5.1.1 ผลของ Dilation, Erosion, Opening และ Closing ที่มีต่อตัวอักษร

- Dilation จะทำให้ตัวอักษรมีความหนามากขึ้น
- Erosion จะช่วยลดความหนาของตัวอักษร ทำให้ตัวอักษรบางลง
- Opening ช่วยลบ noise และช่วยแยกตัวอักษรที่ติดกันออกจากกัน
- Closing ช่วยเชื่อมส่วนของตัวอักษรที่ขาดให้ติดกันและลบ noise ที่อยู่ในตัวอักษร

5.1.2 ผลของ Erosion และ Opening ในการลบ noise เมื่อเทียบกับ Median filter

- Median Filter สามารถลบ noise ในภาพได้ดีกว่า Erosion และ Opening

5.1.3 Structure Element มีผลต่อรูปร่างของตัวอักษร

- ถ้า Structure Element มีลักษณะยาวในแนวตั้ง มีผลต่อความสูงของตัวอักษรและมีผลตัวต่อความหนาของตัวอักษรในแนวนอน
- ถ้า Structure Element มีลักษณะยาวในแนวนอน มีผลต่อความกว้างของตัวอักษรและมีผลตัวต่อความหนาของตัวอักษรในแนวตั้ง

5.1.4 การนำ Dilation, Erosion, Opening และ Closing มาช่วยในการปรับปรุงคุณภาพของตัวอักษร

- **การปรับปรุงตัวอักษรที่มีลักษณะบางหรือเลือนรางแบบไม่มี noise** เมื่อใช้ Closing แล้ว ตามด้วย Dilation โดยใช้ Structure element จะทำให้ตัวอักษรอ่านได้ง่ายมากที่สุดจากผลการทดลองทั้งหมด
- **การปรับปรุงตัวอักษรที่มีลักษณะบางหรือเลือนรางแบบมี noise** ต้องทำการลบ noise ออกก่อนด้วย median filter แล้วจึงดำเนินการเช่นเดียวกับการปรับปรุงตัวอักษรที่มีลักษณะบางหรือเลือนรางแบบไม่มี noise จะทำให้ตัวอักษรอ่านได้ง่ายมากที่สุดจากผลการทดลองทั้งหมด

- **การปรับปรุงตัวอักษรที่มีลักษณะหนาหรือติดกันแบบไม่มี noise เมื่อใช้** Opening แล้วตามด้วย Erosion โดยใช้ Structure Element **จะทำให้ตัวอักษรอ่านได้ง่ายมากที่สุดจากผลการทดลองทั้งหมด**

- **การปรับปรุงตัวอักษรที่มีลักษณะหนาหรือติดกันแบบมี noise ต้องทำการลบ** noise ก่อนด้วย median filter **แล้วจึงดำเนินการเหมือนกับการปรับปรุงตัวอักษรที่มีลักษณะหนาหรือติดกันแบบไม่มี** noise **จะทำให้ตัวอักษรอ่านได้ง่ายมากที่สุดจากผลการทดลองทั้งหมด**

5.2 ปัญหาและอุปสรรค

- ภาพที่มีตัวอักษรบางหรือเลือนรางเมื่อมี noise จะทำให้เกิดปัญหาในการปรังปรุง

- ปัญหาในการประเมินคุณภาพของภาพที่เป็นผลลัพธ์จากการทดลอง **เนื่องจากผลลัพธ์ที่มีคุณลักษณะที่คล้ายๆกัน**

- ขนาดและรูปร่างลักษณะของตัวอักษรมีผลต่อการเลือกใช้ structure **element จึงต้องออกแบบ** structure element **ใหม่สำหรับตัวอักษรที่มีขนาดและรูปร่างต่างกันออกไป**

- ตัวอักษรบางไม่สามารถปรับปรุงให้ดีขึ้นได้ เนื่องจากมีความเสียหาย**มากจนเกินไป**

5.3 ข้อเสนอแนะ

- ก่อนการแปลงภาพจากระดับเทาเป็นขาวดำ ถ้าหากมีการปรับปรุง**คุณภาพของภาพระดับเทาก่อน**แปลงเป็นขาวดำก็จะทำให้ ภาพขาวดำที่ได้จากการแปลงมีการสูญเสีย**ของข้อมูลภาพน้อยลง**

เอกสารอ้างอิง

[1] G.J. Awcock and R.Tromas. **Applied Image Processing.** London : Macmilian Press Ltd. 1995.

[2] Andrian Low. **Introductory Computer Vision and Image Processing.** Singapore : McGraw-Hill Book Company. 1991.

[3] Maher A., Sid-Ahmed. **Image Processing.** Singapore : McGraw-Hill Book Company. 1994.

[4] Rafael C. Gonzalez and Richard E. Woods. **Digital Image Processing.** 2nd Ed., New Jersey : Prentice-Hall,Inc. 2001.

[5] Ronald N. Bracewell. **Two-Dimensional Imaging.** New Jersey : Prentice-Hall,Inc. 1995.

ภาคผนวก

ภาคผนวก ก
โค้ดโปรแกรม

1. โค้ดของการคำนวณหาค่า Threshold

```
function T = threshold(I)
[row col] = size(I);
%กำหนดค่า T เริ่มต้นโดยใช้ค่าเฉลี่ยของค่า pixel ในภาพ I
T = mean(mean(I));
t0 = 0;    Tb = 0;
a1 = 0;    a2 = 0;
%คำนวณหาค่า Treshold(T) ไปจนกว่าความแตกต่างระหว่าง T ใหม่กับ T เก่าต่างกันน้อยกว่า t0
while abs(T-Tb)>t0,
        Tb = T;%Tb เป็นค่า T ก่อนทำการคำนวณ(ค่า T ก่อนหน้า)
        G1 = uint8(I>T).*I;%G1 เก็บค่า pixel ใน I ที่มีค่ามากกว่า T
        G2 = uint8(I<=T).*I;%G2 เก็บค่า pixel ใน I ที่มีค่าน้อยกว่าหรือเท่ากับ T
        for i=1:row,
            for j=1:col,
                if I(i,j) > T,
                    a1 = a1+1;%a2 คือ จำนวน pixel ใน G1
                end
            end
        end
        a2 = (row*col)-a1;%a2 คือ จำนวน pixel ใน G2  ( a1+a2 = จำนวน pixel ทั้งหมดใน I )
        u1 = sum(sum(G1))/a1;%หาค่าเฉลี่ยใน G1
        u2 = sum(sum(G2))/a2;%หาค่าเฉลี่ยใน G2
        T = (u1+u2)/2;%ค่า T หลังจากทำการคำนวณ(ค่า T ใหม่)
        %reset ค่าของตัวแปร a1,a2,G1,G2
        a1=0;    a2=0;
        G1 = 0;  G2 = 0;
end
^return ค่า T เป็น ชนิด double
T = double(T);
```

2. โค้ดของการแปลงภาพระดับเทาเป็นภาพขาวดำ

```
function BW = gray2bw(I)

%I เป็นภาพระดับเทา(gray-scale image)
[m n] = size(I);
L = zeros(m,n);
level = threshold(I);

%แปลงระดับเทา เป็นขาวดำ โดยเมื่อ ค่า pixel ใน I มีค่าน้อยกว่าหรือเท่ากับค่า Threshold ให้
เปลี่ยนค่าเป็น 0
%ถ้าค่า pixel ใน I มีค่ากว่าค่า Treshold ให้เปลี่ยนค่าเป็น 1
for a=1:m,
        for b=1:n,
                if I(a,b)<=level
                        L(a,b)=0;
                else
                        L(a,b)=1;
                end
        end
end

%ส่งค่า output ออกเป็นแบบ binary(0-1)
BW = logical(L);
```

3. โค้ดของการทำ Dilation

```
function D = dilate(A,S);


I = A;
[row col] = size(I);
D = I;
[m n] = size(S);


%เปลี่ยน structure element ให้เป็นแบบขนาด เลขคี่xเลขคี่
if mod(m,2) == 0 || mod(n,2)==0,
        tmp = zeros(m+mod(n,2),n+mod(m,2));
        tmp(1+~mod(m,2):m+~mod(m,2),1+~mod(n,2):n+~mod(n,2)) = S;
        S = tmp;
end
%คำนวณหาขอบเขตของ neightbor pixel ที่ใช้
[ms ns] = size(S);
ci = ceil(ms/2);          cj = ceil(ns/2);
ii= ci-1;                 jj= cj-1;
istart = ci;              iend  = row-ii;
jstart = cj;              jend  = col-jj;
%ทำการ dilation โดยเมื่อเจอ pixel ที่เป็น 0 จึงค่อยทำการ dilation
for i=istart:iend,
        for j=jstart:jend,
                if I(i,j) == 0,
                        D(i-ii:i+ii,j-jj:j+jj) = D(i-ii:i+ii,j-jj:j+jj).*~S;
                end
        end
end
%ส่งค่า output ออกเป็นแบบ binary(0-1)
D = logical(D);
```

4. โค้ดของการทำ Erosion

```
function E = erode(A,S);


I = A;
[row col] = size(I);
E = I;
[m n] = size(S);
%เปลี่ยน structure element ให้เป็นแบบขนาด เลขคี่xเลขคี่
if mod(m,2) == 0 || mod(n,2)==0,
        tmp = zeros(m+mod(n,2),n+mod(m,2));
        tmp(1+~mod(m,2):m+~mod(m,2),1+~mod(n,2):n+~mod(n,2)) = S;
        S = tmp;
end
%คำนวณหาขอบเขตของ neightbor pixel ที่ใช้
[ms ns] = size(S);
ci = ceil(ms/2);          cj = ceil(ns/2);
ii= ci-1;                 jj= cj-1;
istart = ci;              iend  = row-ii;
jstart = cj;              jend  = col-jj;


% ทำการ Erosion โดยเมื่อเจอ pixel ที่อยู่ในขอบเขตของ structure element มีคนเป็น 0
% pixel ที่ตำแหน่ง i,j จะมีค่าเป็น 0 นอกจากนี้จะมีค่าเป็น 1
for i=istart:iend,
        for j=jstart:jend,
                b = I(i-ii:i+ii,j-jj:j+jj).*S;
                E(i,j) = any(any(b));
        end
end
%ส่งค่า output ออกเป็นแบบ binary(0-1)
E = logical(E);
```

5. โค้ดของการทำ Opening

```
function O = open(I,SE)

%ทำการ Erosion ก่อน
E  = erode(I,SE);
%หลังจากทำ Erosion ก็ทำการ Dilation
%ผลลัพธ์ที่ได้จะเป็นผลของการ Opening
O = dilate(E,SE);
```

6. โค้ดของการทำ Closing

```
function C = close(I,SE)

%ทำการ Dilation ก่อน
D  = dilate(I,SE);
%หลังจากทำ Dilation ก็ทำการ Erosion
%ผลลัพธ์ที่ได้จะเป็นผลของการ Closing
C = erode(D,SE);
```

ภาคผนวก ข
คำสั่ง MATLAB ที่ใช้ในการเขียนโปรแกรม

1. ตัวดำเนินการ เครื่องหมายพิเศษ และตัวดำเนินการเปรียบเทียบตรรกะ

โปรแกรม MATLAB ได้กำหนดตัวดำเนินการ เครื่องหมายเปรียบเทียบ และตัวดำเนินการเปรียบเทียบทางตรรกะมาใช้ในการคำนวณเพื่ออำนวยความสะดวกในการเขียนโปรแกรม ดังนี้

ตัวดำเนินการเมตริกซ์

ลักษณะดำเนินการ	ตัวดำเนินการ	รูปแบบของ MATLAB
การบวก	+	A+B
การลบ	-	A-B
การคูณ	*	A*B
การคูณเชิงสมาชิก	.*	A.*B
การหารทางขวา	/	A/B
การหารทางซ้าย	\	A\B
การหารเชิงสมาชิก	./	A./B
การยกกำลัง	Ab	A^B
การยกกำลังสมาชิก	A.b	A.^B

ตัวดำเนินการเปรียบเทียบและตรรกะ

ลักษณะดำเนินการ	ตัวดำเนินการ	รูปแบบของ MATLAB
น้อยกว่า	<	x<10
น้อยกว่าหรือเท่ากับ	<=	x<=10
มากกว่า	>	x>10
มากกว่าหรือเท่ากับ	>=	x>=10
เท่ากับ	= =	x= =1
ไม่เท่ากับ	~=	x~=5
และ	&	x>2&y<1
หรือ	|	x>2|y<1
ไม่	~	~x

เครื่องหมายพิเศษ

เครื่องหมายพิเศษหรือสัญลักษณ์ที่นิยมใช้บ่อยในโปรแกรม MATLAB ส่วนใหญ่จะเป็นเครื่องหมาย % และ ; เครื่องหมาย % จะแสดงหมายเหตุโดยโปรแกรม MATLAB จะมีไม่กระทำคำสั่งที่ถัดจากเครื่องหมาย % ในบรรทัดเดียวกัน ส่วนใหญ่เครื่องหมาย % มักใช้เพื่ออธิบายให้ทราบถึงโปรแกรมที่ใช้ในคำสั่งว่าคือโปรแกรมอะไร มีรูปแบบในการทำงาน และใช้งานได้อย่างไร เพื่อให้ผู้ใช้โปรแกรมสามารถเข้าใจถึงโปรแกรมที่เขียนขึ้น

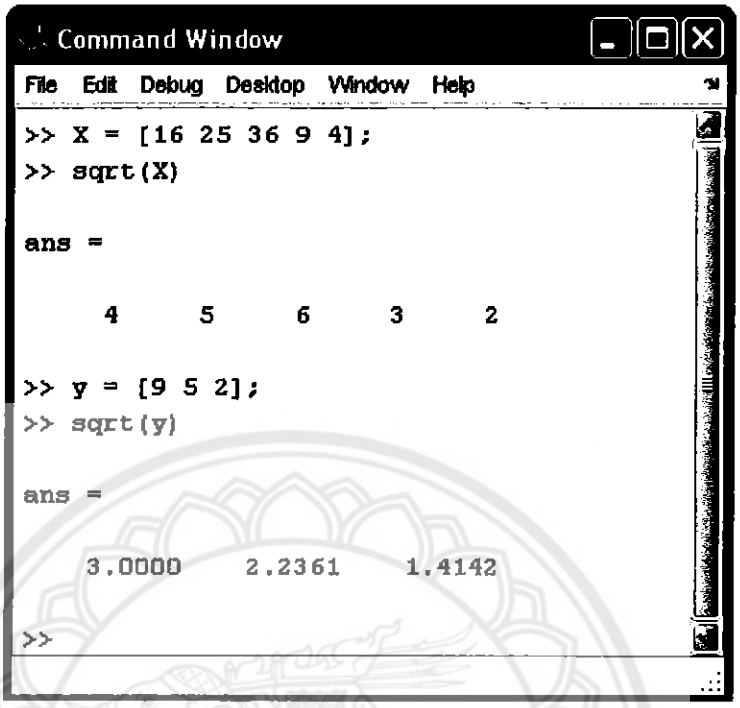
เครื่องหมาย ; จะใช้แยกเมตริกซ์หรือคำสั่งหรือไม่ให้พิมพ์ค่าผลลัพธ์ที่ได้จากการประมวลผล ในการเขียนโปรแกรม MATLAB ส่วนมากจะพบเครื่องหมาย ; อยู่หลังคำสั่งซึ่งเครื่อง หมายนี้จะแยกคำสั่งแต่ละคำสั่งออกจากกัน เมื่อทำการป้อนคำสั่งหลายๆ คำสั่งไว้ในบรรทัดเดียวกันโปร-แกรม MATLAB ก็จะทำการประมวลผลทีละคำสั่งแยกกัน และถ้าทำการป้อนคำสั่งต่างๆ ในโปรแกรมโดยไม่ใช้เครื่องหมาย ; ในการประมวลผลของโปรแกรม MATLAB จะช้ามากเพราะจะต้องทำการพิมพ์ค่าผลลัพธ์ที่ได้ในแต่ละคำสั่งออกมาที่หน้าต่างคำสั่ง นอกเหนือจากนี้ยังสามารถใช้ในการแยกค่าระหว่างแถวและหลักของเมตริกซ์ได้อีกด้วย นอกจากเครื่องหมาย % และ ; แล้วยังมีเครื่องหมายพิเศษอื่นๆ เช่น

เครื่องหมาย	รายละเอียด
.	จุดทศนิยม
()	กำหนด subscripts
=	กำหนดค่า
[]	สร้างเวกเตอร์และเมตริกซ์
:	สร้างเวกเตอร์
...	กระทำคำสั่งยังบรรทัดต่อไป
,	แยก element ภายในเมตริกซ์ subscripts

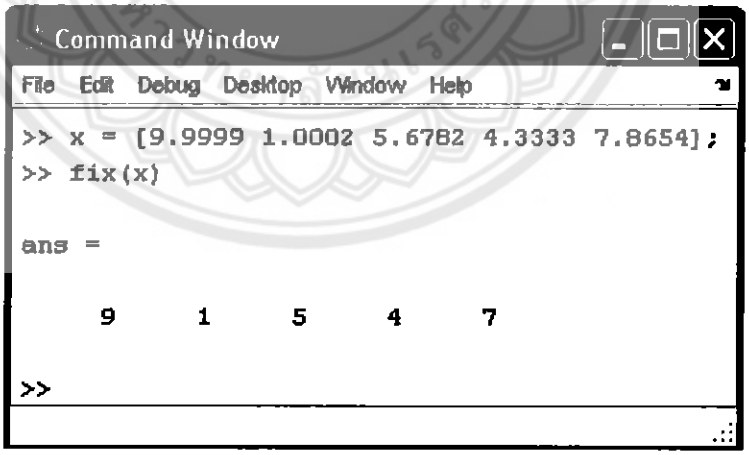
2. ฟังก์ชันในการคำนวณทางพีชคณิต

การเขียนในโปรแกรม MATLAB มักจะทำการคำนวณทางพีชคณิตในด้านต่างๆ บ่อยครั้ง เช่น การหารากที่สอง (square root) การหาค่าสัมบูรณ์ การปัดเศษ เป็นต้น ทำให้โปรแกรมที่เขียนขึ้นมามีความยาวมากและเสียเวลาในการเขียนโปรแกรม ดังนั้นโปรแกรม MATLAB จึงมีฟังก์ชันที่ใช้สำหรับการคำนวณทางพีชคณิตดังนี้

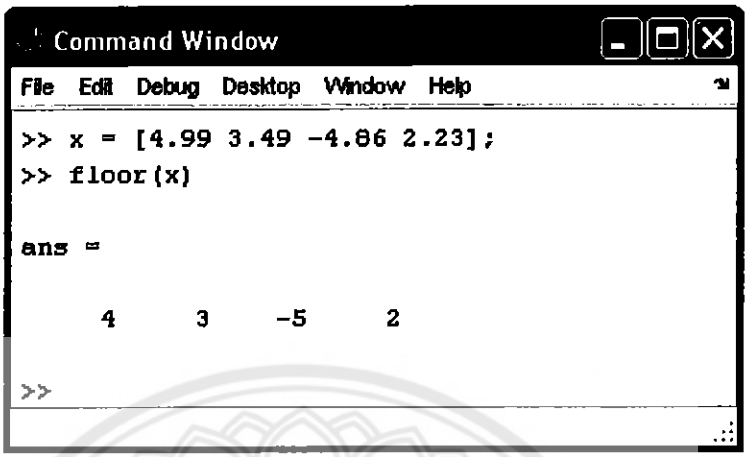
sqrt(x) สามารถใช้ฟังก์ชันนี้ในการหารากที่สองได้ตัวอย่างเช่น



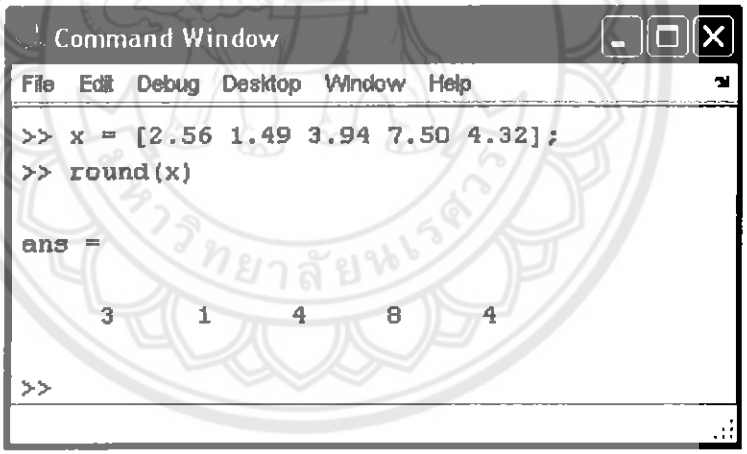
fix(x) ใช้ฟังก์ชันนี้เพื่อหาค่าจำนวนเต็มใดๆ ที่ใกล้เคียงกับค่า x โดยกำหนดให้ค่านั้นเข้าใกล้ 0 มากที่สุด ตัวอย่างเช่น



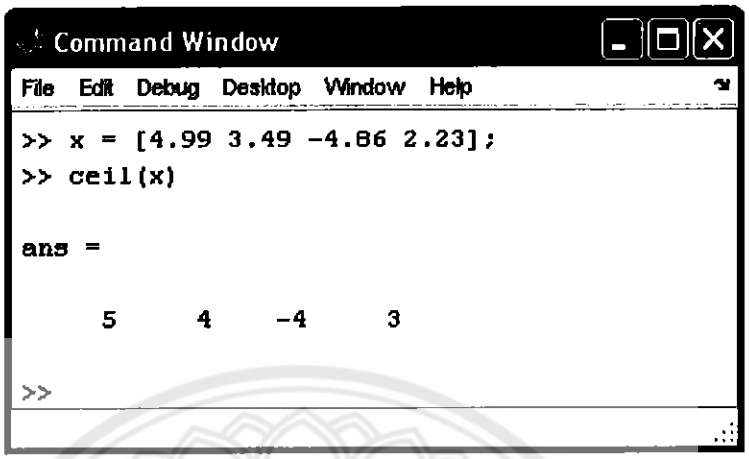
floor(x) ใช้ฟังก์ชันนี้เพื่อหาค่าจำนวนเต็มใดๆ ที่ใกล้เคียงกับค่า x โดยค่านั้นเข้าใกล้ค่า $-\infty$ มากที่สุด ตัวอย่างเช่น



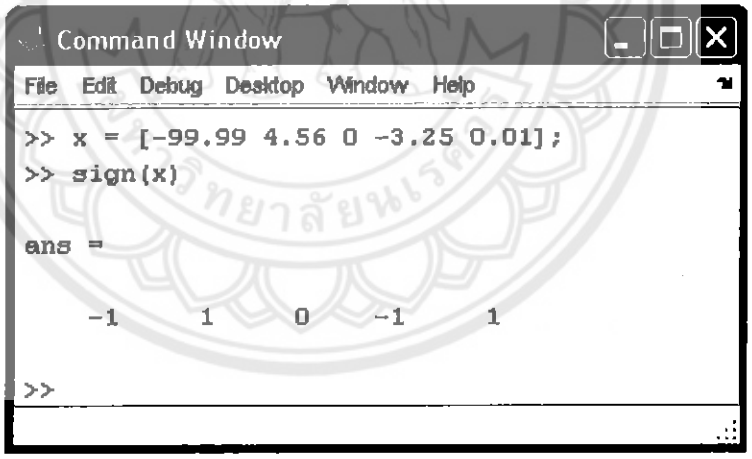
round(x) ใช้ฟังก์ชันนี้เพื่อหาค่าจำนวนเต็มใดๆ ที่ใกล้เคียงกับค่า x ด้วยนัยสำคัญของจุดทศนิยม ตัวอย่างเช่น



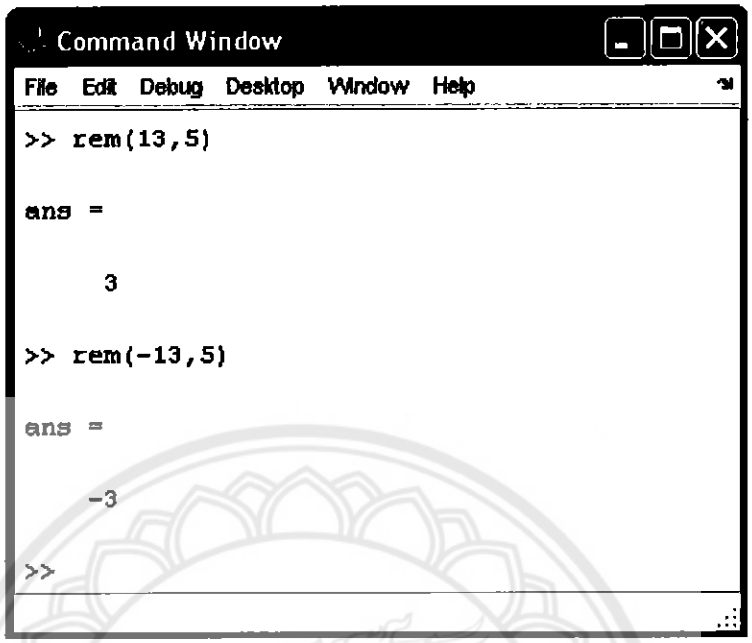
ceil(x) ใช้ฟังก์ชันนี้เพื่อหาค่าจำนวนเต็มใดๆ ที่ใกล้เคียงกับค่า x โดยค่านั้นเข้าใกล้ค่า ∞ มากที่สุด ตัวอย่างเช่น



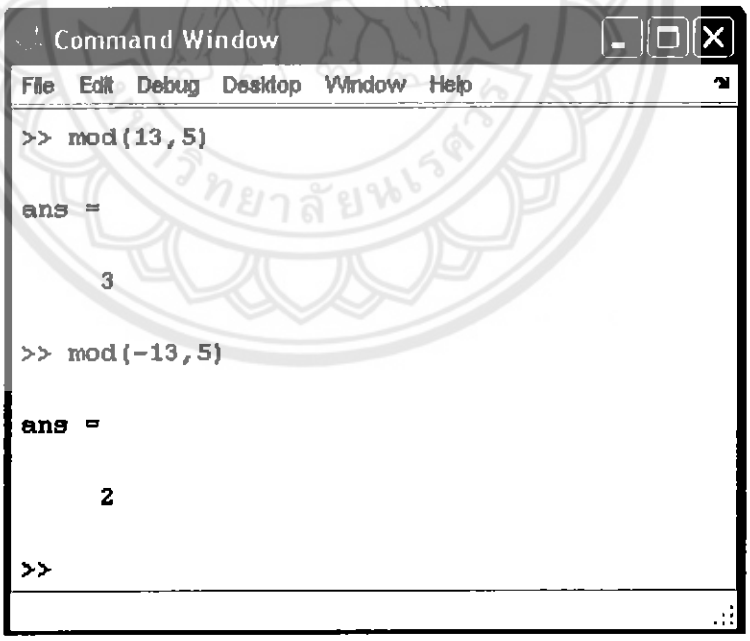
sign(x) ฟังก์ชันนี้ใช้สำหรับการกำหนดค่า x ใดๆ ให้มีค่าเป็น -1, 0 หรือ 1 โดยพิจารณาจาก ถ้าค่า x มีค่าน้อยกว่าศูนย์ให้เป็น -1 หรือ ถ้าค่า x มีค่าเท่ากับศูนย์ให้เป็นศูนย์ หรือถ้าค่า x มีค่ามากกว่าศูนย์ให้เป็น 1 ตัวอย่างเช่น



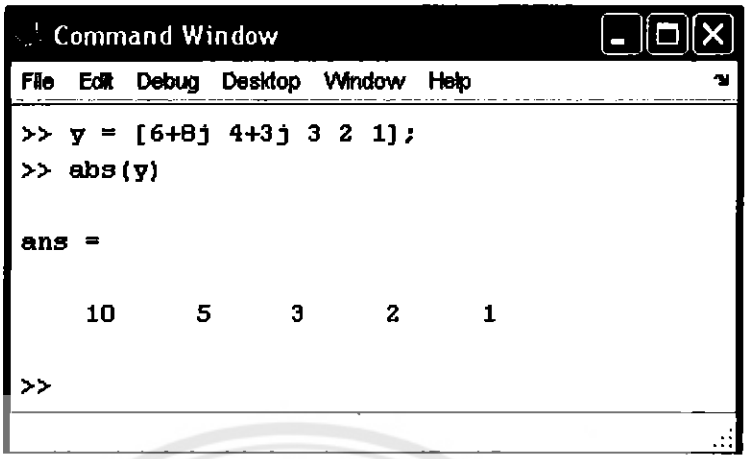
rem(x,y) ฟังก์ชันนี้หาค่าเศษที่เหลือจากการหารกันระหว่างค่า x และ y ตัวอย่างเช่น



mod(x,y) ฟังก์ชันนี้หาค่าเศษที่เหลือจากการหารกันระหว่างค่า x และ y ตัวอย่างเช่น



abs(x) สามารถใช้ฟังก์ชันนี้ในการคำนวณค่า Absolute ตัวอย่างเช่น



ฟังก์ชันที่ใช้ในการคำนวณหาค่า Absolute ของค่าจำนวนจริงจะอาศัยพื้นฐานของสมการการคำนวณเมื่อ X= real + imaginary(j) = a + bj โดย a และ b เป็นจำนวนจริงใดๆ ดังนั้น

$$\text{abs}(X) = \sqrt{a^2 + b^2}$$

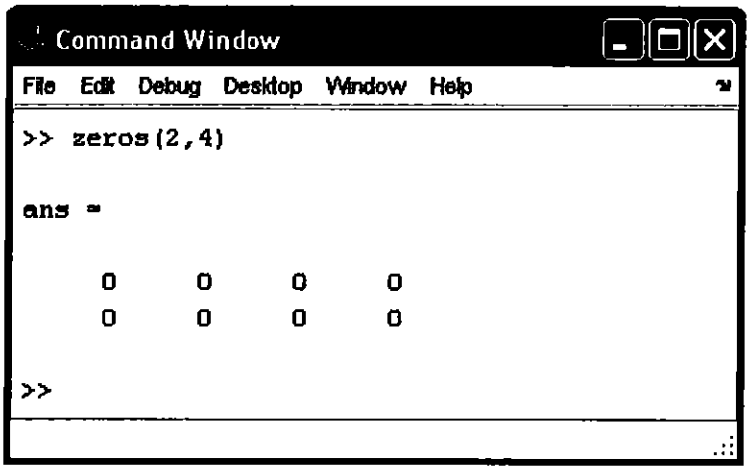
3. อาร์เรย์มาตรฐาน

โปรแกรม MATLAB จะมีฟังก์ชันต่างๆ สำหรับสร้างอาร์เรย์มาตรฐานเพื่อความรวดเร็วและสะดวกต่อการใช้งาน อาร์เรย์ที่เป็นเมตริกซ์พิเศษเหล่านี้ ประกอบด้วยเมตริกซ์ที่เป็นศูนย์ทั้งหมด เมตริกซ์ที่เป็นหนึ่งทั้งหมด เมตริกซ์เอกลักษณ์ เมตริกซ์ของการสุ่มจำนวน เมตริกซ์แนวทแยงมุม และเมตริกซ์ที่ให้ค่าคงที่เฉพาะ ตัวอย่างเช่น

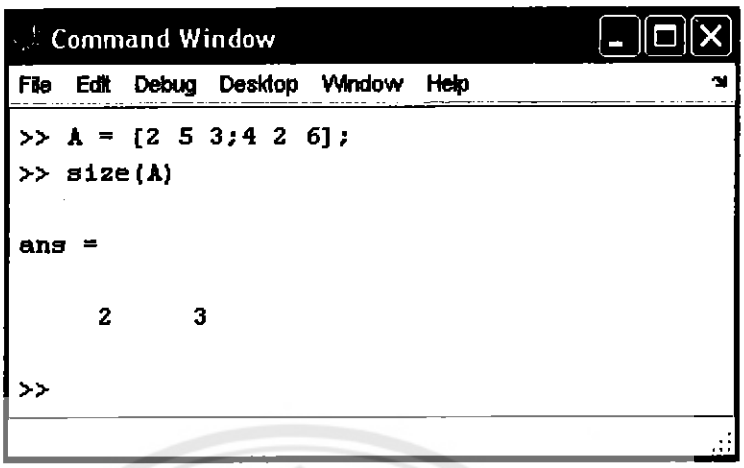
เมตริกซ์ที่เป็นศูนย์ทั้งหมด : จะใช้คำสั่งดังต่อไปนี้

zeros(n) คำสั่งนี้ยะใช้สร้างเมตริกซ์ศูนย์ซึ่งเป็นเมตริกซ์จัตุรัสที่มีขนาด n×n

zeros(m,n) คำสั่งนี้จะใช้สร้างเมตริกซ์ศูนย์ที่เป็นเมตริกซ์แบบ rectangular ที่มีขนาด m×n ตัวอย่างเช่น



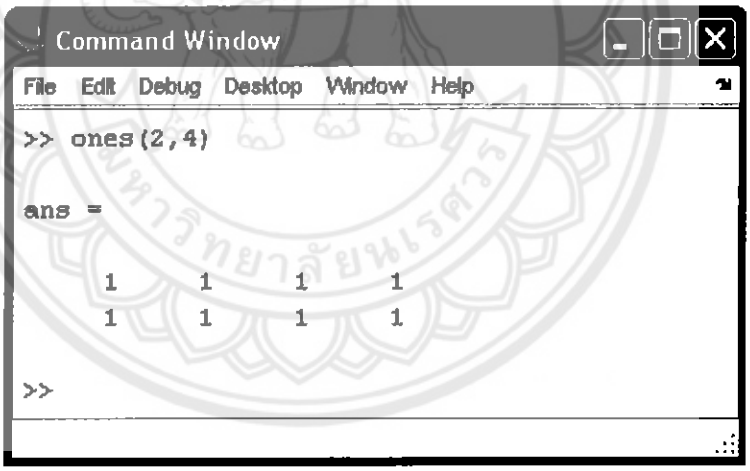
size(A) คำสั่งนี้ใช้หาขนาดของเมตริกซ์ A ใดๆ ตัวอย่างเช่น



เมตริกซ์ที่เป็นหนึ่งทั้งหมด : จะใช้คำสั่งดังต่อไปนี้

ones(n) คำสั่งนี้ใช้สร้างเมตริกซ์ที่มีค่าเป็นหนึ่งทั้งหมดซึ่งเป็นเมตริกซ์จัตุรัสที่มีขนาด n×n

ones(m,n) zeros(m,n) คำสั่งนี้จะใช้สร้างเมตริกซ์หนึ่งที่เป็นเมตริกซ์แบบ rectangular ที่มีขนาด m×n ตัวอย่างเช่น



4. โครงสร้างโปรแกรมพื้นฐาน

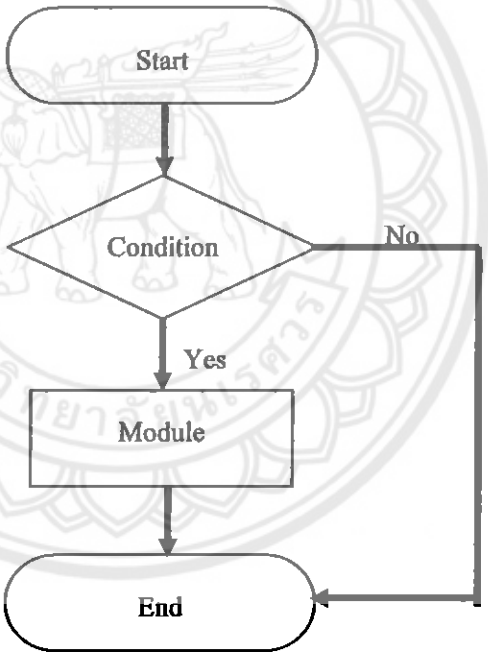
4.1 โครงสร้างแบบมีเงื่อนไข

กระบวนการทำงานแบบมีเงื่อนไขในโปรแกรม MATLAB จะเป็นการควบคุมให้มีการเลือกการทำงานในโมดูลใดโมดูลหนึ่งจากหลายๆโมดูลตามเงื่อนไขที่กำหนดการทำงานในลักษณะนี้เรียกว่าเป็นโครงสร้างแบบ if โครงสร้างแบบมีเงื่อนไขนี้จะแบ่งออกเป็น

4.1.1 เลือกทำหรือไม่ มีโครงสร้างดังนี้

```
if      Condition
        Module1
End
```

หากเงื่อนไขเป็นจริงโมดูล 1 ก็จะถูกกระทำแต่ถ้าเงื่อนไขเป็นเท็จก็จะข้ามไปที่ End การทำงานของโครงสร้างนี้อธิบายด้วยแผนผังโปรแกรมนี้ ได้ดังนี้



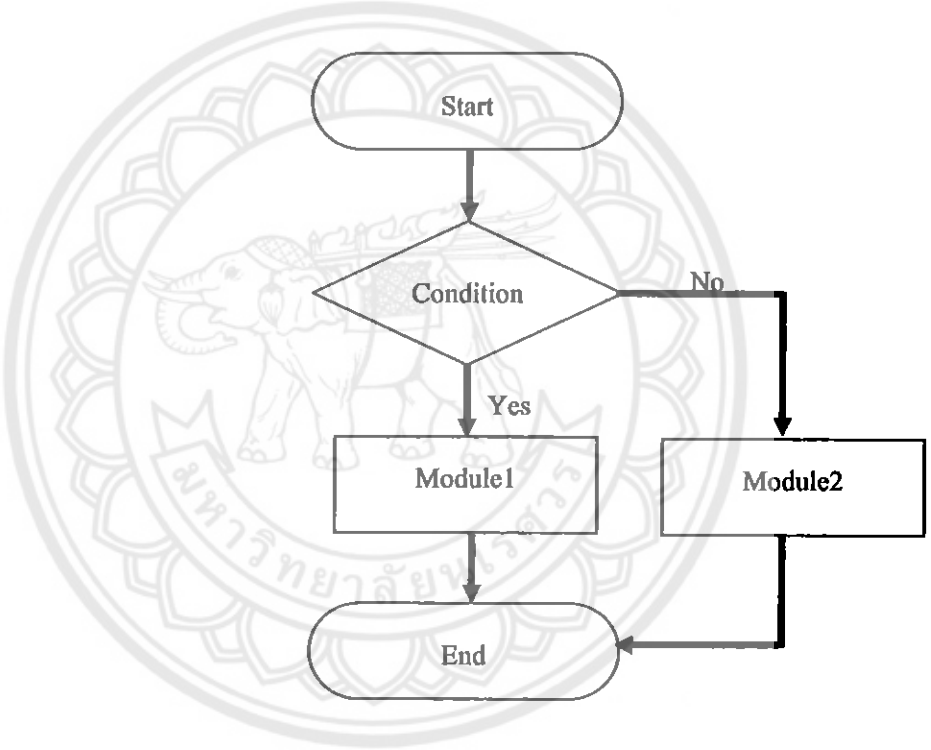
ตัวอย่างเช่น

```
if      x > 0
        y = sin(x)
end
```

4.1.2 เลือกทำอย่างใดอย่างหนึ่งจาก 2 โครงสร้างที่มีตัวอย่างเช่น

```
if      Condition
        Module1
else
        Modeule2
end
```

ถ้าหากเงื่อนไขเป็นจริงจะกระทำโมดูล 1 แต่ถ้าเงื่อนไขเป็นเท็จก็จะกระทำโมดูล 2 การทำงานของโครงสร้างนี้อธิบายด้วยแผนผังโปรแกรมดังนี้



ตัวอย่างเช่น

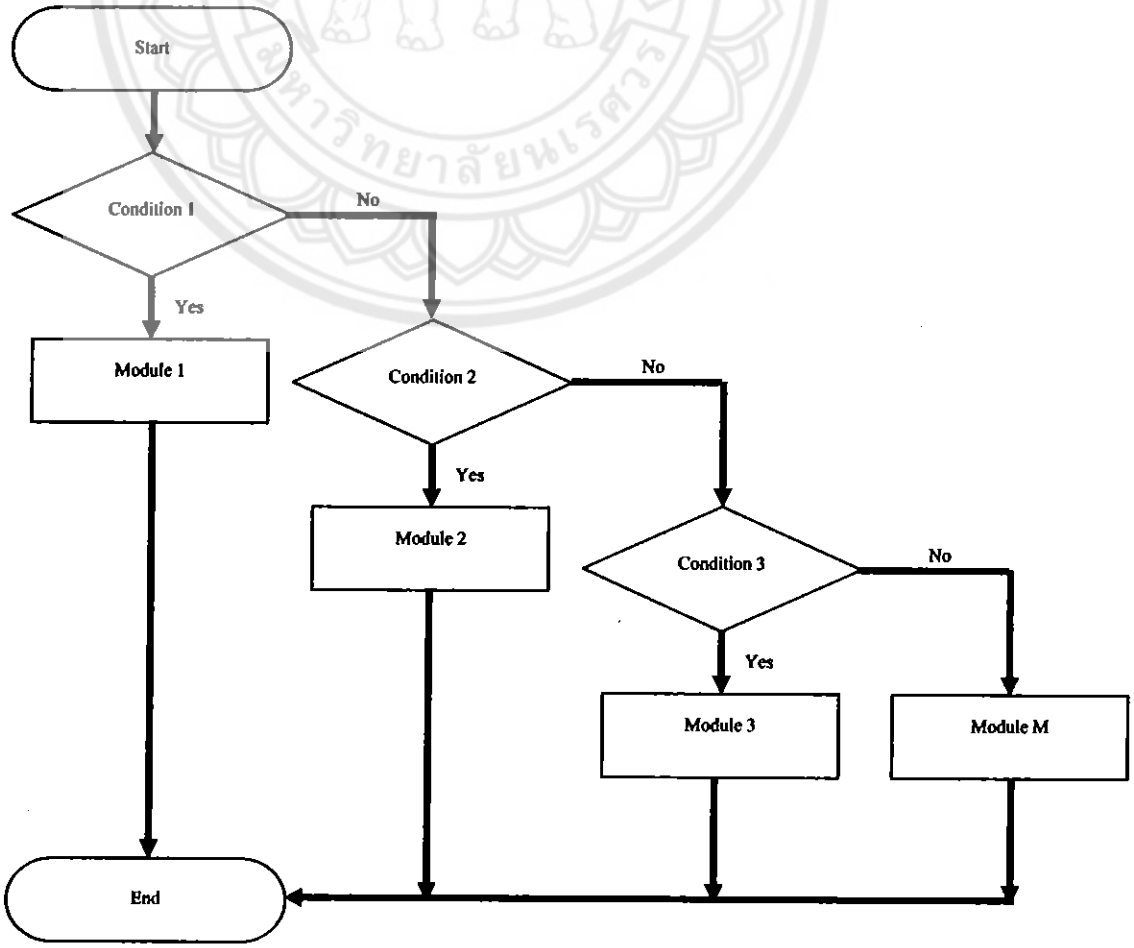
```
if      x > 0
        y = sin(x);
else
        y = cos(x);
end
```

4.1.3 ถ้าโปรแกรมที่ต้องการเขียนมีความซับซ้อนและต้องตัดสินใจ จะใช้เงื่อนไป
ดังต่อไปนี้คือ เลือกอย่างใดอย่างหนึ่งจากหลายๆ statement ซึ่งมีโครงสร้างดังนี้

```
if          Condition 1
            Module 1
elseif      Condition 2
            Module 2
elseif      Condition N
            Module n
else
            Module M
end
```

โครงสร้างนี้จะเป็นการเลือกทำงานจากโมดูลต่างๆ เพียงโมดูลเดียวเมื่อเงื่อนไขเป็นจริง แต่ถ้าเงื่อนไขทั้งหมดไม่เป็นจริงก็จะกระทำโมดูล M ถ้ากำหนดให้เงื่อนไขต่างๆ มี 3 เงื่อนไข การทำงานของโครงสร้างนี้อธิบายด้วยแผนผังโปรแกรมได้ดังนี้

ตัวอย่างต่อไปนี้เป็นตัวอย่างการเขียนโปรแกรมโดยการใช้โครงสร้างเงื่อนไข



```
If          x > 100
            y = sqrt(x);
elseif      x < 100
            y = log(x);
else
            y = 0;
end
```

4.2 โครงสร้างการทำงานแบบซ้ำ

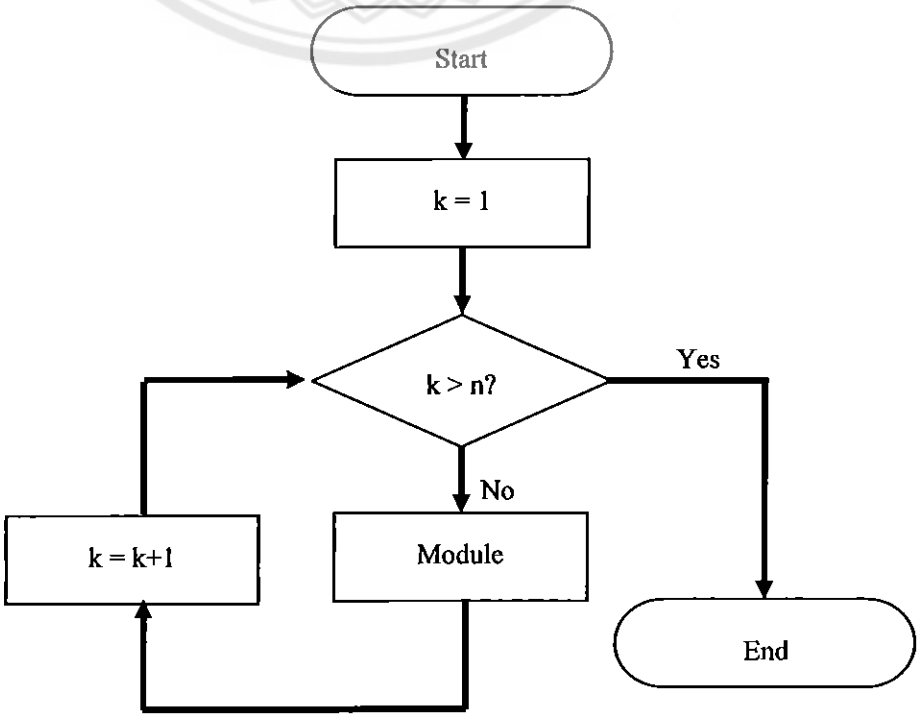
โครงสร้างการทำงานแบบซ้ำจะมีอยู่ 2 แบบคือ การทำงานซ้ำที่ได้กำหนดตัวแปรควบคุมจำนวนรอบซึ่งเรียกว่า โครงสร้าง for และการทำงานที่ใช้เงื่อนไขควบคุมการวนรอบซึ่งเรียกว่าโครงสร้าง while โครงสร้างทั้งสองนี้อธิบายได้ดังนี้

4.2.1 โครงสร้าง for

โครงสร้าง for จะเป็นการทำงานแบบซ้ำที่กำหนดจำนวนรอบที่แน่นอนในการกระทำ ตัวอย่างเช่น

```
for     k = 1:n
        Module
end
```

คำสั่งนี้จะกระทำโมดูลตามจำนวนรอบ n ที่กำหนด เมื่อจำนวนรอบที่กระทำมีมากกว่า n รอบก็จะข้ามไปที่ End การทำงานของโครงสร้างนี้สามารถอธิบายด้วยแผนผังโปรแกรมดังนี้



ตัวอย่างเช่น

```
for     k = 1:10
        x = (k+1)*3;
        y(k) = sqrt(x);
end
```

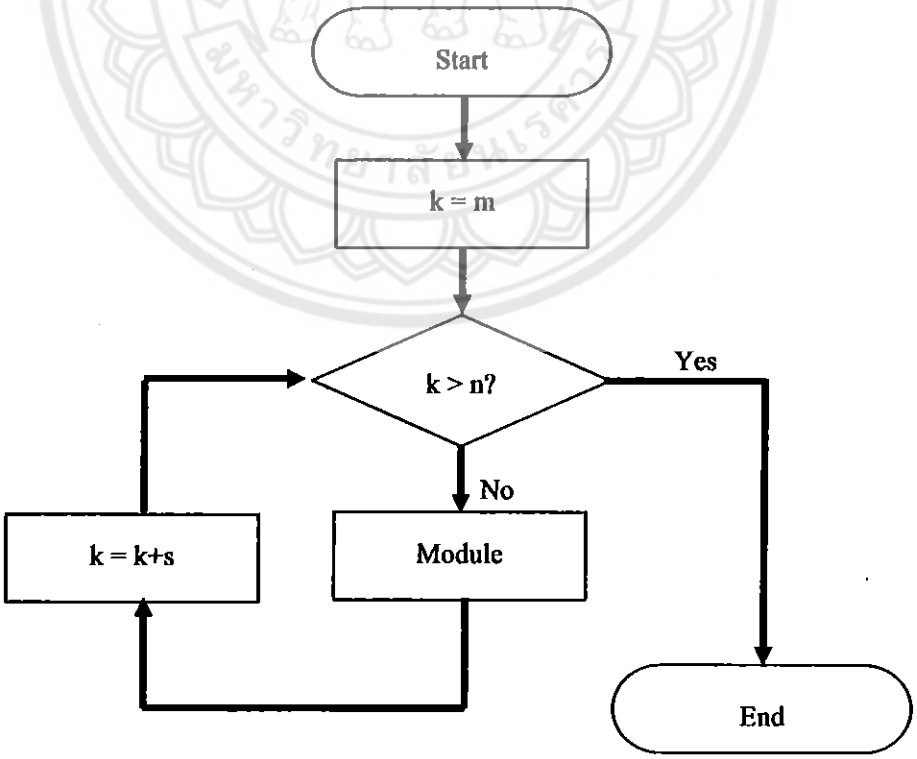
นอกจากนี้รูปแบบของการกำหนดเงื่อนไข for จากตัวอย่างข้างต้นแล้วเรายังสามารถกำหนดในรูปแบบโดยใช้ Colon(:) ได้ดังนี้

for k = ค่าเริ่มต้น:ค่าที่เพิ่มขึ้น:ค่าสิ้นสุด

ซึ่งมีรูปแบบโครงสร้างของคำสั่งดังนี้

```
for     k = m:s:n
        Module
End
```

คำสั่งนี้จะกระทำโมดูลตามเงื่อนไข for ที่กำหนด เมื่อจำนวนที่กระทำมีมากกว่า n ก็จะข้ามไปที่ End การทำงานโครงสร้างนี้สามารถอธิบายด้วยแผนผังโปรแกรมดังนี้



ตัวอย่างเช่น

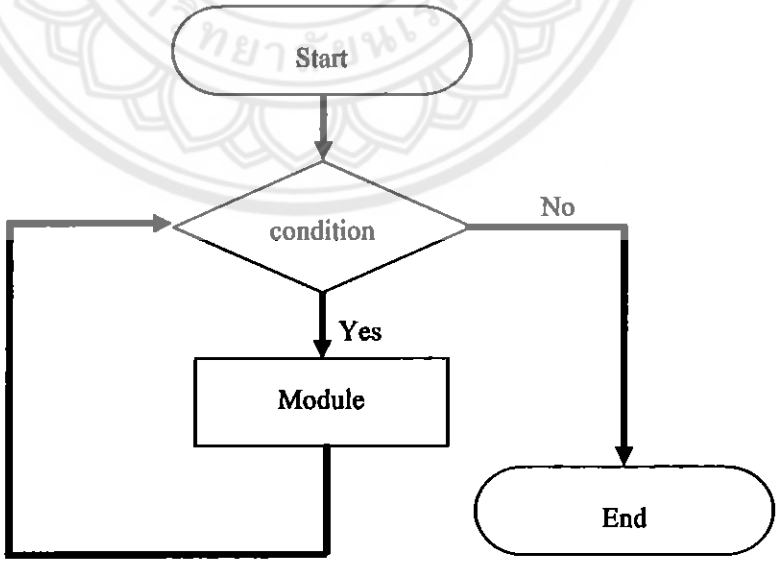
```
for     k = 0:3:300
        x = (k+1)*3;
        y(k) = sin(x);
end
```

4.2.2 โครงสร้าง while

โครงสร้าง while เป็นการทำงานโดยใช้เงื่อนไขควบคุมการวนรอบโดยเงื่อนไขที่กำหนดจะต้องมรการเปลี่ยนแปลง เพราะถ้าเงื่อนไขที่กำหนดไม่มีการเปลี่ยนแปลงก็จะทำให้เกิดการวนรอบไม่มีที่สิ้นสุด กล่าวคือ โปรแกรมจะกระทำไม่จำกัดจำนวนรอบจนกว่าเงื่อนไขจะเป็นเท็จ โดยโครงสร้างนี้จะตรวจสอบเงื่อนไขก่อนกระทำคำสั่ง การทำงานของโครงสร้างนี้เป็นดังนี้

```
while       Condition
            Module
End
```

คำสั่งนี้จะกระทำ Module จนกว่าเงื่อนไขจะเป็นเท็จจึงจะข้ามไปที่คำสั่ง end การทำงานของโครงสร้างนี้สามารถอธิบายด้วยแผนผังโปรแกรมได้ดังนี้



ตัวอย่างเช่น

```
while       x < 100
            x = x+1;
            y = sin(x);
end;
```

4.2.3 โครงสร้าง switch-case

โครงสร้างนี้เป็นโครงสร้างแบบมีเงื่อนไขที่กระทำตามรูปแบบ statement switch ที่กำหนดในแต่ละcase การประมวลผล case แรกจนถึง case สุดท้ายจะสัมพันธ์กับ statement switch เสมอ

```
switch  Condition
case        Condition 1
            Module 1
case        Condition 2
            Module 2
Otherwise
end
```

ตัวอย่างเช่น

```
t = [0:100]*pi/50;
data = input('Type sin, cos or tan: ','s');
response = lower(data);
switch response
case 'sin'
        si = sin(t);
        plot(si);
case 'cos'
        co = cos(t);
        plot(co);
case 'tan'
        ta = tan(t);
```

```
        plot(ta);
    otherwise
        display('You have not enter a proper choice.')
    end
```

5. ฟังก์ชันทางสถิติ

5.1 ค่าเฉลี่ยหรือมัชฌิมเลขคณิต (Mean)

การวัดค่าทางสถิติด้วยวิธีการหาค่าเฉลี่ยนะเป็นวิธีที่นิยมใช้กันมาก ค่าเฉลี่ยจะมีค่าเท่ากับผลรวมของข้อมูลทั้งหมดแล้วหารด้วยจำนวนข้อมูลซึ่งสามารถอธิบายได้ด้วยสมการดังนี้

$$\overline{X} = \frac{\sum X_i}{n}$$

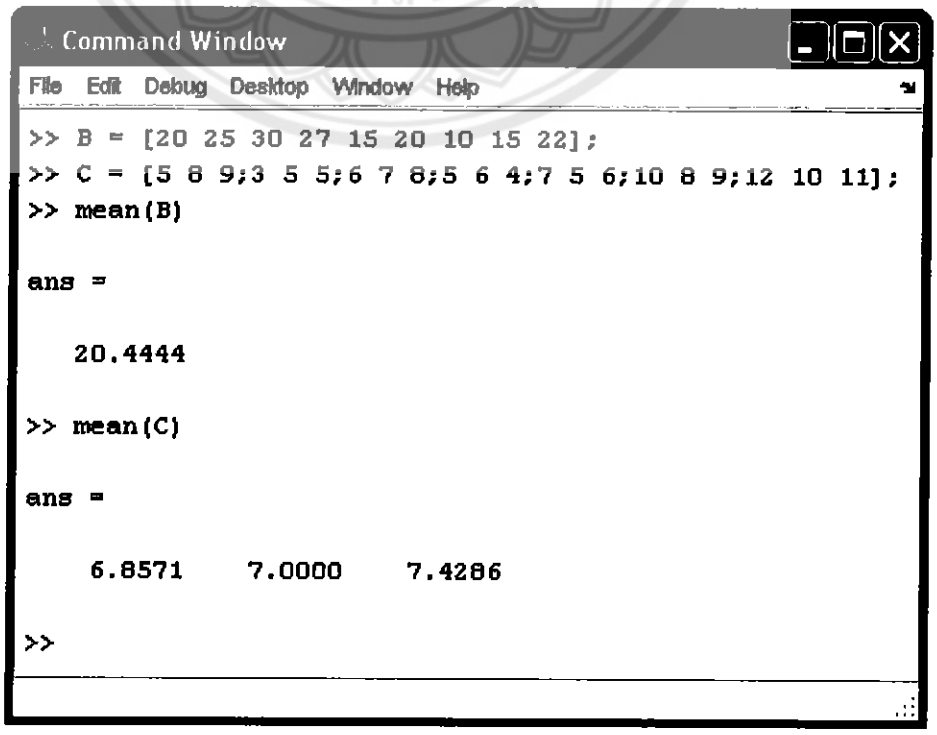
เมื่อ $\overline{X}$ คือค่าเฉลี่ยของกลุ่มตัวอย่าง

X_i คือค่าข้อมูลแต่ละตัว

n คือจำนวนข้อมูลทั้งหมด

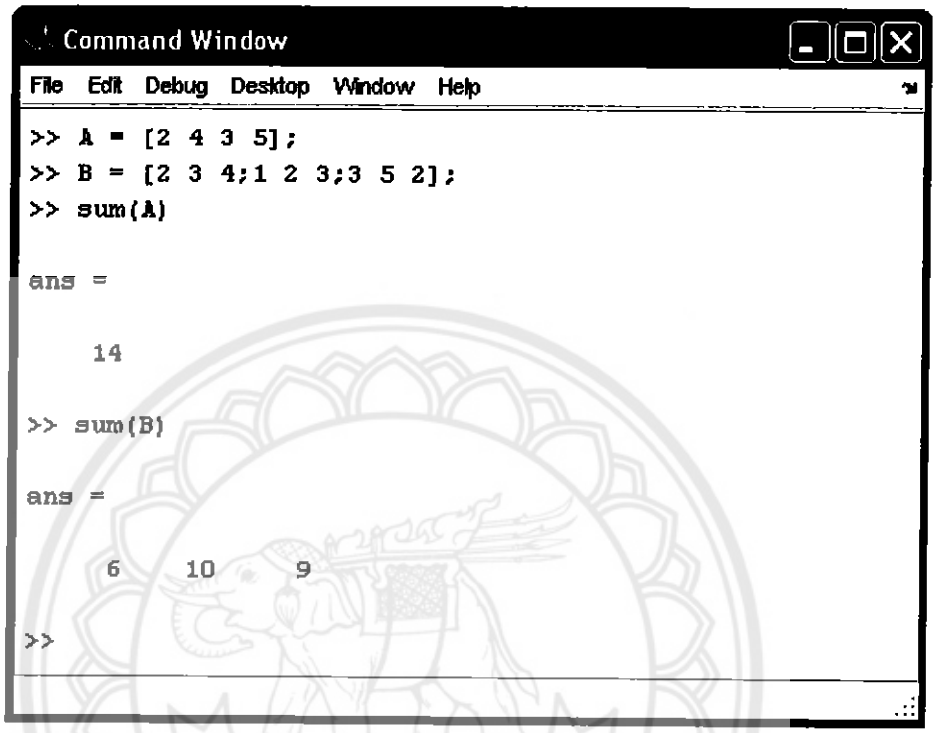
โปรแกรม MATLAB มีฟังก์ชันหรือคำสั่งที่ใช้สำหรับคำนวณค่าเฉลี่ยคือ mean ซึ่งมีโครงสร้างดังต่อไปนี้

mean(A) คือ การหาค่าเฉลี่ยของ A ถ้า A เป็นเวกเตอร์ค่าผลลัพธ์ที่ได้จากการคำนวณจะเป็นค่าสเกลาร์ ถ้า A เป็นเมตริกซ์ค่าผลลัพธ์ที่ได้จากการคำนวณจะเป็นเวกเตอร์แถวที่ได้จากค่าเฉลี่ยในแต่ละหลักของเมตริกซ์ A ตัวอย่างเช่น



5.2 ผลรวม (sum)

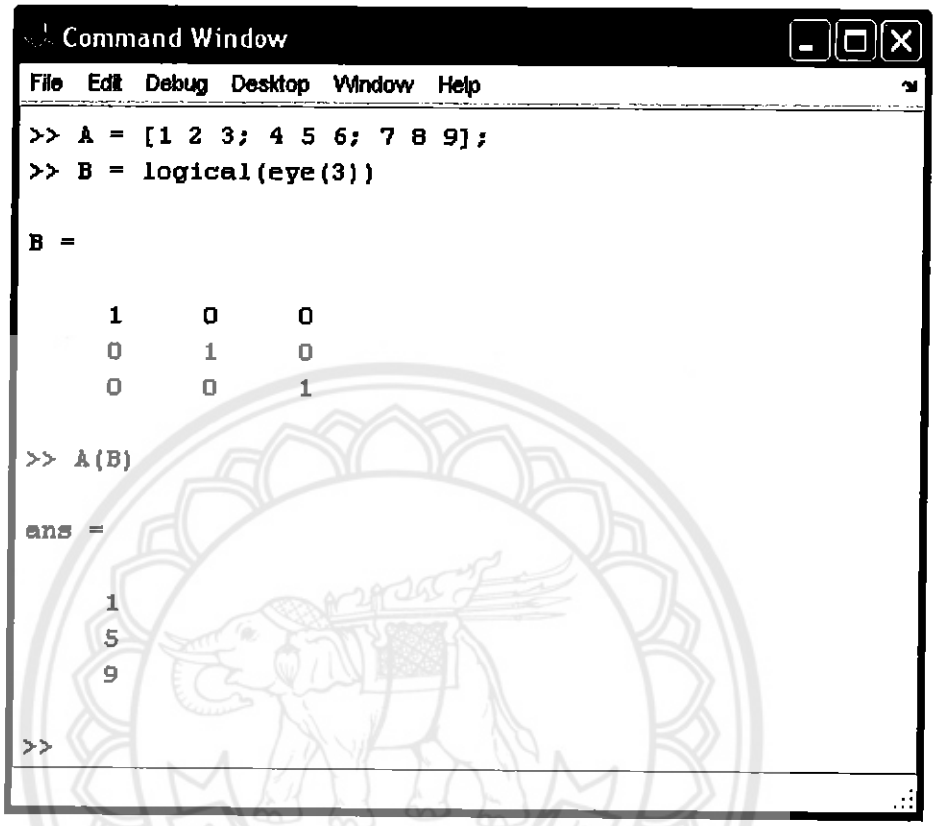
sum(A) คือการคำนวณผลรวมของข้อมูล A ถ้า A เป็นเวกเตอร์ ค่าผลรวมของ A จะเป็นค่าสเกล่าร์ ที่เป็นผลรวมของทุกๆ element ใน A ถ้า A เป็นเมตริกซ์ ค่าผลรวมของ A จะเป็นเวกเตอร์แถวที่เป็นผลรวมในแต่ละหลักของเมตริกซ์ A ตัวอย่างเช่น



5.3 any(a) คือ จะเช็คดูว่าสมาชิกในอาร์เรย์มีอย่างน้อย 1 ตัวที่ไม่เป็น 0 จะ return ฟังก์ชันอื่นๆ ที่ใช้ในการเขียนโปรแกรม

6. Binary

Logical คือ แปลงค่าให้เป็น Binary ตัวอย่างเช่น



uint8 คือ คำสั่งในการเปลี่ยนข้อมูลให้เป็นชนิด unsigned integer ขนาด 8 บิต

7. การเขียนฟังก์ชัน

function คือ คำสั่งในการสร้างฟังก์ชันใน M-file ตัวอย่างเช่น

```
function [mean,stdev] = stat(x)
n = length(x);
mean = sum(x)/n;
stdev = sqrt(sum((x-mean).^2/n));
```

ประวัติผู้เขียนโครงงาน

ชื่อ นายวันชัย กิสิน

ภูมิลำเนา 45/2 ม.6 ต.แก่งโสภา อ.วังทอง จ.พิษณุโลก

ประวัติการศึกษา

- จบมัธยมศึกษาจากโรงเรียนทรัพย์ไพรวัลย์วิทยาคม จ.พิษณุโลก
- ปัจจุบันกำลังศึกษาอยู่ระดับปริญญาตรีชั้นปีที่ 4 สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail: foobfeeb@hotmail.com

ชื่อ นายประวิทย์ วงศ์จักรคำ

ภูมิลำเนา 19 ม.12 ต.แม่ถอด อ.เถิน จ.ลำปาง

ประวัติการศึกษา

- จบมัธยมศึกษาจากโรงเรียนเถินวิทยา จ.ลำปาง
- ปัจจุบันกำลังศึกษาอยู่ระดับปริญญาตรีชั้นปีที่ 4 สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail: one_cpe@hotmail.com