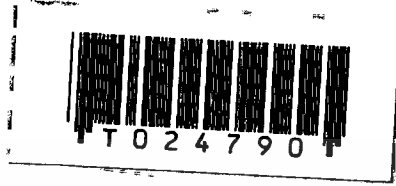


การหาขอบภาพโดยใช้แบบจำลองฟัซซีและนิเวรอลเน็ตเวิร์ค
Edge Detection using Fuzzy and Neural Network Model



สาริต อินทจักร์
MR. SATHIT INTAJAG

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2538

ISBN 974-621-440-3

ลิขสิทธิ์ของบัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เลขหม.....

เลขทะเบียน..... 24790

วัน, เดือน, ปี..... 18 ส.ค. 2539

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ หากสนใจที่จะนำเอกสารนี้ไปใช้ กรุณาติดต่อขอขออนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

EDGE DETECTION USING FUZZY AND NEURAL NETWORK MODEL



A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE
MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING
GRADUATE SCHOOL
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
1995
ISBN 974-621-440-3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	การหาขอบภาพโดยใช้แบบจำลองฟัซซี่เซตและนิวรอลเน็ตเวิร์ค
นักศึกษา	นายสาธิต อินทจักร์
อาจารย์ผู้ควบคุมวิทยานิพนธ์	ผศ.ดร. กิตติ ไพฑูรย์วัฒนกิจ
ระดับการศึกษา	วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า
ภาควิชา	วิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร ลาดกระบัง
พ.ศ.	2538

บทคัดย่อ

การหาขอบภาพในวิทยานิพนธ์ฉบับนี้ จะใช้แบบจำลองฟัซซี่เซตและนิวรอลเน็ตเวิร์คเป็นเครื่องมือในการหาขอบโดยอาศัยคุณสมบัติของเนื้อภาพ (Texture) เป็นตัวบอกถึงความเป็นขอบภาพของค่าความเข้มสำหรับจุดภาพใดๆ (Pixels) ในภาพว่าจะมีความเป็นขอบอยู่มากน้อยเพียงใด ซึ่งได้ใช้ฟัซซี่ลอจิกเป็นตัววิเคราะห์ในส่วนนี้ เนื่องจากฟัซซี่ลอจิกมีความเหมาะสมกับงานที่มีลักษณะเช่นนี้ นั่นคือถ้าจุดภาพมีการเปลี่ยนแปลงค่าความเข้มอย่างทันทีทันใด ค่าความเป็นไปได้หรือระดับความเป็นสมาชิก (Grade or Degree of Membership) ของฟัซซี่เซตของการเป็นขอบภาพจะสูงและจะใช้คุณสมบัติการสร้างความสัมพันธ์ระหว่างข้อมูลแบบไม่เป็นเชิงเส้น (Nonlinear Mapping) ของนิวรอลเน็ตเวิร์คมาหาความสัมพันธ์ระหว่างระดับความเป็นสมาชิกของฟัซซี่เซตว่าเป็นขอบภาพหรือไม่ และในวิทยานิพนธ์นี้ได้เสนอแบบจำลองของฟัซซี่ลอจิกร่วมกับนิวรอลเน็ตเวิร์คที่เรียกว่า "ฟัซซี่นิวรอลเน็ตเวิร์ค (Fuzzy Neural Network: FNN)" ที่ใช้ในการหาขอบภาพด้วย สำหรับการทดสอบหาขอบภาพได้ใช้กับภาพทั้งที่เป็นภาพที่จำลองขึ้นมาและภาพจริงที่มีระดับความเข้มของสีเทา (Grey-Scale) อยู่ในช่วง [0-255] และขอบภาพที่ได้จะทำการประเมินความถูกต้องโดยการสร้างภาพอ้างอิงที่รู้ตำแหน่งของขอบภาพอย่างแน่นอนแล้ววัดความถูกต้องของขอบภาพที่หาได้โดยวิธีนี้ และได้มีการเปรียบเทียบผลของวิธีที่เสนอกับการหาขอบภาพโดยวิธีอื่นๆ ซึ่งนิยมใช้กันทั่วไป เช่น Sobel, Marr, Gabor, Edge Relaxation เป็นต้น ซึ่งจากผลการเปรียบเทียบปรากฏว่าวิธีที่เสนอจะให้ผลลัพธ์ที่ถูกต้องและแน่นอนกว่า เนื่องจากวิธีที่เสนอจะหาขอบภาพอย่างอัตโนมัติ โดยผู้ใช้ไม่จำเป็นต้องกำหนดค่าเทรชโฮลด์ หรือค่าตัวแปรใดๆ

กิติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลงได้ด้วยดีก็เพราะได้รับความเมตตาจากท่านผู้ช่วยศาสตราจารย์ ดร. กิตติไพฑูรย์วัฒนกิจ ที่ได้กรุณาตรวจสอบความถูกต้องและให้คำแนะนำแก่ผู้วิจัยตลอดมา ผู้วิจัยรู้สึกทราบบซึ้งและขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณ กรมอุตุนิยมวิทยา และกองสำรวจทรัพยากรธรรมชาติด้วยดาวเทียม สำนักงานการวิจัยแห่งชาติ กระทรวงวิทยาศาสตร์เทคโนโลยีและการพลังงาน ที่กรุณาให้ภาพถ่ายดาวเทียมในการศึกษาและทดลอง

นายสาธิต อินทจักร

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญภาพ.....	VIII
บทที่ 1	
ความรู้พื้นฐานเกี่ยวกับการประมวลผลภาพดิจิทัล	1
-การประมวลผลภาพเชิงตัวเลข.....	1
-การประมวลผลก่อน.....	2
-การหาขอบภาพ	2
-การวัดประสิทธิภาพของขอบภาพ.....	22
-การหา Texture.....	23
-สรุป.....	37
บทที่ 2	
พีชชีเซต	38
-บทนำ.....	38
-การแทนข้อมูลโดยพีชชีเซต	38
-ฟังก์ชันความเป็นสมาชิก.....	40
-ตัวดำเนินการพีชชีเซต.....	45
-ส่วนขยายของพีชชีเซต.....	47
-แบบจำลองพีชชีเซต.....	48
-ความคลุมเครือ.....	58
-สรุป.....	60
บทที่ 3	
การปรับปรุงภาพ	62
-บทนำ.....	62
-การประมวลผลภาพแบบจุด	62

สารบัญ (ต่อ)

	หน้า
-การกำจัดสัญญาณรบกวน	81
-การเน้นขอบภาพ	86
-สรุป.....	87
บทที่ 4	
นิเวศน์เน็ตเวิร์ค.....	88
-บทนำ.....	88
-ความรู้พื้นฐานทางกายภาพของเซลล์ประสาท.....	88
-แบบจำลองโครงข่ายเซลล์ประสาท.....	92
-การเรียนรู้ของโครงข่าย	96
-การแพร่กระจายกลับ	99
-สรุป.....	109
บทที่ 5	
พีชชีนิเวศน์เน็ตเวิร์ค	111
-บทนำ.....	111
-ปัญหาการออกแบบระบบพีชชีเซต และนิเวศน์เน็ตเวิร์ค.....	111
-การลดปัญหาในการออกแบบระบบพีชชีเซต และนิเวศน์เน็ตเวิร์ค.....	112
-การสอนโครงข่ายพีชชีนิเวศน์	112
-สรุป.....	115
บทที่ 6	
การหาขอบภาพ	116
-บทนำ.....	116
-การหาขอบภาพโดยการประมวลผลก่อน โดยใช้พีชชีเซต.....	116
-การหาขอบภาพโดยพีชชีเซต	125
-การหาขอบภาพโดยนิเวศน์เน็ตเวิร์ค	135
-การหาขอบภาพโดยพีชชีนิเวศน์เน็ตเวิร์ค	141
-ผลการทดลองและการเปรียบเทียบ.....	157
-สรุป.....	176

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการเรียนการสอนเท่านั้น ไม่สามารถนำเนื้อหาไปใช้เชิงพาณิชย์ได้

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่ 7

สรุปและวิจารณ์	178
เอกสารอ้างอิง.....	180
ภาคผนวก ก. แผนภูมิขั้นตอนการหาขอบภาพ โดยพีชชีเซต.....	185
ภาคผนวก ข. แผนภูมิขั้นตอนการหาขอบภาพ โดย DABP.....	186
ภาคผนวก ค. อัลกอริทึมการสอน โครงข่ายพีชชีนิวรอลเน็ตเวิร์ค.....	187
ภาคผนวก ง. บทความและผลงานวิจัยที่ได้รับการตีพิมพ์	193
ประวัติผู้เขียน	194



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
1.1 การกำหนดชนิดของขอบภาพ	20
1.2 แสดงผลการคำนวณค่าตัวแปรทางสถิติของ Cooccurrence เมตริกซ์.....	35
1.3 แสดงผลการคำนวณค่าตัวแปรทางสถิติของ Cooccurrence เมตริกซ์.....	36
3.1 แสดงขอบเขตของกฎแต่ละข้อของอินพุทและเอาต์พุท.....	76
4.1 แสดงการปรับค่าถ่วงน้ำหนักสำหรับกฎการสอนแบบต่างๆ.....	97
6.1 แสดงค่าตัวแปรแต่ละตัวของข้อมูลภาพ.....	145
6.2 เปรียบเทียบประสิทธิภาพของตัวดำเนินการหาขอบภาพของแต่ละวิธีสำหรับรูปที่สร้างขึ้น	166
6.3 เปรียบเทียบประสิทธิภาพของตัวดำเนินการหาขอบภาพของแต่ละวิธีสำหรับรูปแผนที่.....	175



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

รูปที่		หน้า
1.1	แสดงลักษณะขอบภาพแบบต่างๆ	3
1.2	แสดงขอบภาพชนิดต่างๆ	4
1.3	แสดงตัวดำเนินการหาขอบภาพของ Laplacian.....	9
1.4	แสดงลักษณะของจุด Zero-Crossing ที่ใช้ในการกำหนดขอบภาพ.....	10
1.5	ตัวดำเนินการหาขอบภาพของ Marr และ Hildreth.....	11
1.6	ตัวดำเนินการหาขอบภาพของ Haralick.....	13
1.7	ฟังก์ชันคี่ Gabor แบบสองมิติ.....	15
1.8	ผลของทิศทางของตัวกรอง θ ที่มีผลตอบสนองกับ $\sigma \cdot \omega$	15
1.9	แสดงตำแหน่งของ a, b, c และขอบภาพ e	18
1.10	แสดงตำแหน่งของขอบภาพที่จะนำมาพิจารณาร่วมเพื่อใช้ในการกำหนดชนิดของความเป็นขอบภาพ.....	19
1.11	แสดงลักษณะของเนื้อภาพ.....	24
1.12	ภาพระดับสีเทาที่มีค่าความเข้ม 4 ระดับ.....	27
1.13	รูปแบบทั่วไปของ Cooccurrence เมตริกซ์	27
1.14	แสดง Cooccurrence เมตริกซ์ที่ระยะทาง $d=1$ โดยมีทิศทางต่างๆ กัน.....	28
1.15	ค่าความน่าจะเป็นของ Cooccurrence เมตริกซ์ตามแนวอนและระยะทาง $d=1$	28
1.16	ฮิสโตแกรมแบบสองมิติโดย Cooccurrence เมตริกซ์ที่ระยะทาง $d = 1$	36
2.1	ค่าความเป็นสมาชิกของเซต A ในกรณีที่เป็นเซตธรรมดา.....	39
2.2	ค่าความเป็นสมาชิกของเทอมเซต A ในกรณีที่เป็นฟัซซี่เซต	40
2.3	แสดงกราฟฟังก์ชันความเป็นสมาชิกของตัวแปร “ความสูง” ที่ประกอบด้วย 3 เทอมเซต.....	41
2.4	แสดงกราฟฟังก์ชันความเป็นสมาชิกชนิดต่างๆ.....	45
2.5	Hedge ของตัวแปรอุณหภูมิ.....	48
2.6	บล็อกไดอะแกรมของแบบจำลองฟัซซี่ลอจิก.....	48
2.7	แผนผังการควบคุม Stream Turbine.....	50
2.8	แสดงเทอมเซตของตัวแปรของตัวแปรต่างๆที่ใช้ในระบบการควบคุม Stream Turbine	51
2.9	แสดงเมตริกซ์ FAM ของระบบควบคุม STREAM TURBINE.....	53
2.10	แสดงการคำนวณค่าความเป็นสมาชิกของเทอมเซต COOL และ NORMAL	54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่	หน้า
2.11	แสดงการคำนวณค่าความเป็นสมาชิกของเทอมเซต LOW และ OK 54
2.12	แสดงกฎที่เป็นจริงของ FAM เมื่อ $T[t] = 190 \text{ }^{\circ}\text{C}$ และ $P[t] = 100 \text{ Kgmm}^{-2}$ 55
2.13	แสดงขั้นตอนการประมาณค่า $I[t]$ โดยคำนวณจากค่าความจริงที่มีค่าต่ำสุด 56
2.14	แสดงการ Defuzzification โดยใช้ค่าสูงสุดของค่าอัตราความเป็นสมาชิก 57
2.15	แสดงการ Defuzzification โดยวิธี COA 58
3.1	แสดงลักษณะของฟังก์ชันที่ใช้ในการยืดภาพให้คมชัดขึ้น 62
3.2	การปรับปรุงภาพโดยวิธีการยืดช่วงของภาพ 64
3.3	การแปลงข้อมูลภาพโดยการตัดสัญญาณรบกวน 65
3.4	การแปลงข้อมูลภาพระดับสีเทา ไปเป็นภาพไบนารี 65
3.5	ผลของการปรับปรุงภาพโดยวิธีการตัดสัญญาณรบกวน 67
3.6	ลักษณะของฟังก์ชันการแปลงแบบ Digital Negative 67
3.7	ผลของการกลับของมุลภาพ 68
3.8	แสดงกราฟลักษณะการเน้นข้อมูลบางช่วง 69
3.9	ผลของการปรับปรุงภาพโดยวิธีการเน้นข้อมูลภาพเป็นช่วงๆ 70
3.10	ผลของการปรับปรุงภาพโดยวิธีการเน้นข้อมูลเฉพาะบางบิต 72
3.11	ผลของการปรับปรุงภาพโดยวิธีการตัดทอนข้อมูลภาพเฉพาะบางบิต 74
3.12	ผลของการปรับปรุงภาพโดยวิธีการบีบข้อมูลในบางช่วง 75
3.13	บล็อกไดอะแกรมของกฎฟัซซีที่ใช้ในการปรับคอนแทรสต์ภาพ 76
3.14	กฎฟัซซีที่ใช้ในการปรับปรุง 77
3.15	เปรียบเทียบผลของการประมวลผลภาพพร้อมทั้งแสดงการแจกแจง ของความถี่สะสมของแต่ละวิธีซึ่งแสดงอยู่ได้ภาพ 80
3.16	สรุปวิธีการปรับปรุงภาพแบบจุด 81
3.17	แสดงตัวกรองสัญญาณเชิงเส้น 82
3.18	แสดงผลลัพธ์ของการประมวลผลภาพเพื่อการกำจัดสัญญาณรบกวนซึ่งมีสองชนิดคือ Gaussian และ ยูนิฟอร์ม 85
3.19	วินโดว์ที่เป็นตัวกรองสัญญาณชนิดที่ยอมให้ความถี่สูงผ่าน 86
3.20	แสดงผลลัพธ์ของการเน้นของภาพโดยใช้ตัวกรอง High-Pass 86
4.1	การสื่อสารข้อมูลในระบบประสาท 89
4.2	โครงสร้างของเซลล์ประสาทและการส่งสัญญาณกระแสประสาท 89

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญญภาพ(ต่อ)

รูปที่	หน้า
4.3	แสดงลักษณะการเกิดศักย์ไฟฟ้าขณะทำงาน 91
4.4	แสดงภาวะต่างๆ ของเซลล์ประสาท 91
4.5	แสดงศักย์ไฟฟ้าขณะทำงานที่เกิดจากการรวมกันของซิแนปส์ 91
4.6	แบบจำลองเซลล์ประสาทของ McCulloch-Pitts และ โครงข่ายลอจิกพื้นฐาน 93
4.7	แบบจำลองโครงข่ายเซลล์ประสาทเทียม 93
4.8	Activation Function แบบชนิดต่างๆ 95
4.9	โครงข่าย Feedforward แบบชั้นเดียว 98
4.10	โครงข่ายป้อนกลับแบบเวลาไม่ต่อเนื่องชั้นเดียว 98
4.11	โครงข่าย Multilayer Perceptron ที่มี 3 ชั้น 99
4.12	แสดงความสัมพันธ์ระหว่างตัวแปร x กับ y 100
4.13	แสดงความสัมพันธ์ระหว่างตัวแปร x กับ y โดยมีชั้นฮิดเดนเป็นตัวเชื่อม 100
4.14	แผนภูมิการสอนแบบ Error Back-propagation ที่มี 1 ชั้น 105
5.1	แสดงแบบจำลองพีชชีนิวโรลเน็ตเวิร์คที่ใช้ในการหาขอบภาพ 113
5.2	แสดงโครงข่าย FNN ที่เอาท์พุทและค่าถ่วงน้ำหนักเป็นพีชชีเซต 113
6.1	ลักษณะของฟังก์ชัน Contrast Intensification ที่ใช้ในการเน้นภาพเมื่อค่า α ต่างๆ กัน 117
6.2	กราฟของฟังก์ชัน π ที่มี K ช่วง 119
6.3	แสดงขอบภาพที่ได้จากวิธีการประมวลผลภาพก่อน โดยใช้พีชชีเซต 124
6.4	แสดงขอบภาพที่ได้จากวิธีการประมวลผลภาพก่อน โดยใช้พีชชีเซต 125
6.5	แสดงโครงสร้างขอบภาพ 126
6.6	แสดงการแบ่งพื้นที่ของข้อมูลภาพที่ถูกแบ่งออกเป็นสองส่วน 128
6.7	อินพุทของโครงข่าย 135
6.8	ข้อมูลที่ใช้ในการสอนเน็ตเวิร์ค 136
6.9	แบบจำลองโครงข่ายนิวโรลที่ใช้ในการหาขอบภาพ 137
6.10	ขั้นตอนการกำหนดค่าเอาท์พุทของแต่ละ โหนดของชั้นที่ 2-4 138
6.11	แสดงผลของตัวแปรที่ได้จากการวิเคราะห์เนื้อหาของแต่ละตัว 144
6.12	ผลของค่าการแพร่กระจายและค่าส่วนเบี่ยงเบนมาตรฐาน ที่ใช้ค่าแวนโวนุ่มสู่ส่วนกลางต่างกัน 144
6.13	แสดงฟังก์ชันความเป็นสมาชิกของตัวแปรเอนโทรปี 150

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญญภาพ(ต่อ)

รูปที่	หน้า
6.14	แสดงกราฟฟังก์ชันความเป็นสมาชิกของตัวแปรการแพร่กระจาย และส่วนเบี่ยงเบนมาตรฐาน 151
6.15	แสดงผลของตัวแปรแต่ละตัวที่ใช้โครงข่ายพีชชีนิวรอลหาขอบภาพ 152
6.16	แสดงขั้นตอนการแปลงที่เมตริกซ์ย่อยของค่าส่วนเบี่ยงเบนมาตรฐาน ไปเป็นค่าความเป็นสมาชิกของขอบภาพ 152
6.17	แสดงโครงข่ายพีชชีนิวรอลที่ใช้ในการหาขอบภาพ 154
6.18	แผนภูมิการสอนโครงข่ายพีชชีนิวรอลที่ใช้กฎการเรียนรู้เซลล์ดำแบบพีชชี 155
6.19	แสดงแพทเทิร์นที่ จากข้อมูลภาพขนาด 32 x 32 156
6.20	แสดงภาพที่สร้างขึ้นเพื่อนำมาใช้ทดสอบวิธีการหาขอบภาพแต่ละวิธี 159
6.21	ขอบภาพของรูปที่สร้างขึ้นซึ่งได้จากวิธีพีชชีนิวรอลเน็ตเวิร์ค 159
6.22	ขอบภาพของรูปที่สร้างขึ้นซึ่งได้จากวิธีการหาขอบภาพโดยนิวรอลเน็ตเวิร์ค 160
6.23	ขอบภาพของรูปที่สร้างขึ้นซึ่งได้จากวิธีการหาขอบภาพโดยพีชชีเซต 160
6.24	ขอบภาพของรูปที่สร้างขึ้นซึ่งได้จากวิธีของ Robert 161
6.25	ขอบภาพของรูปที่สร้างขึ้นซึ่งได้จากวิธีของ Prewitt 161
6.26	ขอบภาพของรูปที่สร้างขึ้นซึ่งได้จากวิธีของ Sobel 162
6.27	ขอบภาพของรูปที่สร้างขึ้นซึ่งได้จากวิธีของ Frei-Chen 162
6.28	ขอบภาพของรูปที่สร้างขึ้นซึ่งได้จากวิธีของ Kirsch 163
6.29	ขอบภาพของรูปที่สร้างขึ้นซึ่งได้จากวิธีของ Laplacian 163
6.30	ขอบภาพของรูปที่สร้างขึ้นซึ่งได้จากวิธีของ Marr และ Hildreth 164
6.31	ขอบภาพของรูปที่สร้างขึ้นซึ่งได้จากวิธีของ Canny 164
6.32	ขอบภาพของรูปที่สร้างขึ้นซึ่งได้จากวิธีของ Gabor 165
6.33	ขอบภาพของรูปที่สร้างขึ้นซึ่งได้จากวิธีของ Relaxation 165
6.34	ภาพแผนที่ 168
6.35	ขอบของภาพแผนที่ที่ได้จากวิธีพีชชีนิวรอลเน็ตเวิร์ค 168
6.36	ขอบของภาพแผนที่ที่ได้จากวิธีนิวรอลเน็ตเวิร์ค 169
6.37	ขอบของภาพแผนที่ที่ได้จากวิธีพีชชีเซต 169
6.38	ขอบของภาพแผนที่ซึ่งได้จากวิธีของ Prewitt 170
6.39	ขอบของภาพแผนที่ซึ่งได้จากวิธีของ Sobel 170
6.40	ขอบของภาพแผนที่ซึ่งได้จากวิธีของ Frei-Chen 171

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

รูปที่		หน้า
6.41	ขอบของภาพแผนที่ซึ่งได้จากวิธีของ Kirsch	171
6.42	ขอบของภาพแผนที่ซึ่งได้จากวิธีของ Laplacian	172
6.43	ขอบของภาพแผนที่ซึ่งได้จากวิธีของ Marr-Hildreth	172
6.44	ขอบของภาพแผนที่ซึ่งได้จากวิธีของ Canny	173
6.45	ขอบของภาพแผนที่ซึ่งได้จากวิธีของ Gabor	173
6.46	ขอบของภาพแผนที่ซึ่งได้จากวิธีของ Relaxation	174



บทที่ 1 การประมวลผลภาพเชิงตัวเลข

ภาพที่นำมาประมวลผลด้วยคอมพิวเตอร์จะถูกแทนด้วยตัวเลขที่อยู่ในรูปของเมตริกซ์ แต่ภาพที่ได้จากตัวรับสัญญาณ จะอยู่ในรูปของฟังก์ชัน $f(x,y)$ ที่ต่อเนื่องในระนาบสองมิติ (คือแกน X และ Y) ดังนั้นภาพเชิงตัวเลขจึงได้จากการแปลงฟังก์ชันต่อเนื่อง $f(x,y)$ ให้เป็นฟังก์ชันไม่ต่อเนื่อง $g(x,y)$ โดยการแบ่ง $f(x,y)$ ออกเป็นช่วงๆ (Quantized) ซึ่งผลลัพธ์ได้ก็คือค่าความเข้มของจุดภาพ (Pixel หรือ Picture Element) ที่มีอยู่ L ระดับ รวมทั้งค่าพิกัด x, y โดยค่าเหล่านี้จะเป็นเลขจำนวนเต็มที่ไม่เป็นลบ โดยทั่วไปจุดภาพแต่ละจุดจะเป็นสมาชิกของเมตริกซ์ที่มีขนาด M แถว N คอลัมน์ ($M \times N$) เพราะฉะนั้น x, y จะมีค่าอยู่ในช่วง $(1 \leq y \leq M, 1 \leq x \leq N)$ และจำนวนช่วงของระดับความเข้มของจุดภาพ L จะบ่งถึงระดับความละเอียดของภาพเชิงตัวเลข (Digital Image) โดยทั่วไปแล้ว L จะมีค่าความเข้มเท่ากับ 256 ระดับ ซึ่งจะทำให้ค่าของจุดภาพอยู่ในช่วง $[0-255]$ โดยใช้ 8 บิต ($2^8 = 256$) สำหรับเก็บข้อมูลภาพในแต่ละจุดและขนาดของเมตริกซ์ เป็น 256×256 ในกรณีที่ต้องการภาพที่มีความละเอียดของระดับความเข้มสูงๆ L อาจต้องการจำนวนบิตสำหรับเก็บข้อมูลมากกว่า 8 บิต คืออาจเป็น 16 หรือ 24 บิต โดยค่าระดับความเข้มของจุดภาพจะเท่ากับ 2^{16} และ 2^{24} ตามลำดับ โดยที่ขนาดของเมตริกซ์ก็จะเพิ่มขึ้นเป็น 512×512 , และ 1024×1024 เป็นต้น

วิธีการประมวลผลภาพเชิงตัวเลขที่ทำให้คอมพิวเตอร์สามารถรู้จักวัตถุภายในภาพได้นั้น พอจะแบ่งได้สองระดับด้วยกันคือ การประมวลผลภาพในระดับต่ำ (Low-level Image Processing) และการประมวลผลภาพในระดับสูง (High-level Image Processing)

การประมวลผลภาพในระดับต่ำ เป็นการประมวลผลเชิงตัวเลขเกือบทั้งหมด เพื่อหาตัวแปรต่างๆ มาอธิบายข้อมูลภาพ โดยมีจุดประสงค์เพื่อนำตัวแปรเหล่านี้ไปใช้ในการประมวลผลในระดับสูงต่อไป ซึ่งการประมวลผลในระดับสูงก็คือการทำให้คอมพิวเตอร์รู้จักและเข้าใจภาพได้ ในที่นี้จะพิจารณาเฉพาะการประมวลผลระดับต่ำซึ่งโดยทั่วไปจะประกอบไปด้วย การประมวลผลก่อน (Image Pre-processing), การกำจัดสัญญาณรบกวน, การทำให้ภาพคมชัด, การหาขอบภาพ, สำหรับการแบ่งแยกวัตถุภายในภาพ (Image Segmentation), การจำแนกภาพ (Image Classification) และการบีบอัดข้อมูลภาพ (Image Compression) จะไม่กล่าวถึงในที่นี้

ความแตกต่างของการประมวลผลภาพระดับต่ำและระดับสูงคือ ข้อมูลที่นำมาใช้ในการประมวลผลภาพ การประมวลผลภาพระดับต่ำจะใช้ค่าความสว่างของจุดภาพโดยตรง ส่วนการประมวลผลภาพในระดับสูงข้อมูลที่จะนำมาประมวลผลจะถูกแสดงในรูปของสัญลักษณ์ โดยสัญลักษณ์เหล่านี้จะแสดงถึงสิ่งต่างๆ ที่อยู่ในภาพ โดยใช้ตัวแปรที่ได้จากการประมวลผลในระดับต่ำมาอธิบายถึงสัญลักษณ์เหล่านี้ เช่น ขนาดของวัตถุ, รูปร่าง, และความสัมพันธ์กันระหว่างวัตถุต่างๆ ภายในภาพ

จะเห็นได้ว่า การประมวลผลในระดับต่ำนั้นมีความสำคัญมาก สำหรับที่จะทำให้คอมพิวเตอร์สามารถเข้าใจภาพได้ แต่ในปัจจุบันการประมวลผลภาพในระดับต่ำยังไม่มีวิธีใดที่สามารถแก้ปัญหา

ได้อย่างสมบูรณ์ ในที่นี้ได้เสนอวิธีการใหม่ๆ ที่จะนำมาช่วยแก้ปัญหาที่คือ ฟิชซีเซต [1-2], นิวรอลเน็ตเวิร์ค [3-4], และยังมีวิธีการอื่นๆ อีกที่ไม่ได้กล่าวถึงรายละเอียดในที่นี้เช่น จีเนติกอัลกอริทึม, (Genetic Algorithm [5]) และ การจำลองแอนนิลลิ่ง (Simulated Annealing) [6] เป็นต้น วิธีการเหล่านี้มีความยืดหยุ่นสูง และสามารถนำมาแก้ปัญหาเหล่านี้ได้อย่างมีประสิทธิภาพ

1.1 การประมวลผลก่อน

วิธีการประมวลผลก่อนเป็นการดำเนินการกับภาพในระดับต่ำสุด ซึ่งเป็นการเตรียมข้อมูลเพื่อนำไปใช้ในขั้นตอนถัดไป โดยข้อมูลเข้าและผลลัพธ์จะเป็นค่าความเข้มของจุดภาพ ซึ่งข้อมูลเข้าเป็นข้อมูลภาพเชิงตัวเลขที่ยังไม่เคยผ่านการประมวลผลใดๆ มาก่อน

จุดประสงค์ของการประมวลผลภาพก่อนพอจะจำแนกได้ดังนี้

1. ปรับปรุงข้อมูลภาพในส่วนที่ไม่คมชัด
2. กำจัดสัญญาณรบกวน
3. คอนทราสต์หรือปรับปรุงคุณสมบัติของการมองเห็น เช่น การปรับค่าความเข้มของจุดภาพ การกลับข้อมูลภาพ เป็นต้น การประมวลผลข้อมูลในลักษณะเช่นนี้ก็เพื่อแสดงผลภาพอาจจะเป็นบนจอภาพ หรือทางเครื่องพิมพ์ก็ได้
4. การแบ่งแยกข้อมูลภาพออกจากสีพื้น โดยใช้ค่าขีดเริ่มเปลี่ยนหรือค่าเทรชโฮลด์ (Threshold)
5. การแปลงข้อมูลภาพในทางเรขาคณิต (Geometric Correction) เช่น การหมุนภาพ, การเปลี่ยนแปลงขนาดของภาพ, การแก้ไขตำแหน่งของจุดภาพ
6. การแก้ไขข้อมูลภาพในส่วนที่มีการบิดเบี้ยว

1.2 การหาขอบภาพ

การหาขอบภาพเป็นการหาขอบเขตของวัตถุภายในภาพ ซึ่งขอบเขตของวัตถุจะเป็นคุณสมบัติที่เด่นและมีความสำคัญมากที่สุดที่จะนำไปสู่การรู้จักวัตถุนั้นๆ โดยคอมพิวเตอร์ จะเห็นได้ว่าการหาขอบภาพในที่นี้ก็คือ การดึงคุณลักษณะโครงร่างที่เด่นของวัตถุออกมานั่นเอง ลักษณะที่เด่นของวัตถุที่เรามองเห็นกันโดยทั่วไปก็คือส่วนที่เป็นสันหรือส่วนที่เป็นขอบของวัตถุ เมื่อส่วนนี้มีแสงมาตกกระทบก็就会有ความสว่างมากกว่าส่วนอื่นๆ ของวัตถุ ซึ่งลักษณะเช่นนี้ก็เกิดขึ้นในฉงพเชิงตัวเลขที่จะนำมาประมวลผลในที่นี้เช่นกัน

ขอบภาพเชิงตัวเลขคือ ฟังก์ชันค่าความเข้มของจุดภาพที่มีการเปลี่ยนแปลงค่าความเข้มอย่างทันทีทันใด และจุดที่เป็นขอบภาพของวัตถุจะแบ่งพื้นผิวของวัตถุหนึ่งออกจากวัตถุอีกชิ้นหนึ่งหรืออาจจะแบ่งพื้นผิวภายในวัตถุเดียวกัน หรืออาจจะแบ่งแยกวัตถุออกจากสีพื้นของภาพ โดยทั่วไปลักษณะของขอบภาพที่ดีจะต้องมีลักษณะดังต่อไปนี้

เอกสารนี้เป็นความบางคือ จุดที่เป็นขอบภาพจะต้องมีความกว้างเพียงจุดเดียวเท่านั้น ใช้ประโยชน์ด้านการคำนวณว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. มีความต่อเนื่อง โดยจุดของขอบภาพในวัตถุเดียวกันควรมีความต่อเนื่องกัน ถ้าจุดที่เป็นขอบภาพมีอยู่เพียงจุดเดียว โดยไม่ต่อเนื่องกับจุดใดในย่านใกล้เคียงเลยอาจจะเป็นขอบภาพที่ไม่สมบูรณ์หรืออาจจะเป็นสัญญาณรบกวนก็ได้
3. ตำแหน่งของจุดที่เป็นขอบภาพควรมีความถูกต้อง โดยขอบที่มีความถูกต้องนั้นจะต้องอยู่ในส่วนที่มีค่าความเข้มของจุดภาพสูงกว่า

ในรูปที่ 1.1 ได้แสดงขอบภาพในลักษณะต่างๆ โดยรูป ก เป็นข้อมูลภาพที่เป็นไบนารีที่มีขอบภาพปรากฏในแนวตั้ง ในรูป ข เป็นขอบภาพที่มีความถูกต้องสมบูรณ์คือ ขอบภาพมีความบาง, จุดที่เป็นขอบภาพมีความต่อเนื่องกัน, และตำแหน่งของจุดขอบภาพก็มีความถูกต้อง โดยขอบภาพในข้อมูลภาพชุดนี้เกิดจากการเปลี่ยนแปลงค่าความเข้มจาก 1 ไป 0 และตำแหน่งของขอบภาพที่ถูกต้องควรอยู่ในส่วนที่มีค่าความเข้มสูงกว่าดังรูป ข (โดยจุดที่เป็นขอบภาพจะแทนด้วย 1 และส่วนที่เป็นสีพื้นจะแทนด้วย 0) ในรูป ค จุดที่เป็นขอบภาพอยู่ในตำแหน่งที่ผิดเพราะขอบภาพไปอยู่ในตำแหน่งที่ค่าความเข้มต่ำกว่าซึ่งผิดไปจากความเป็นจริงส่วนใน รูป ง และรูป จ เป็นขอบภาพที่ไม่ถูกต้องสมบูรณ์เนื่องจากขาดคุณสมบัติข้อใดข้อหนึ่งหรือมากกว่าของคุณสมบัติทั้งสามข้อข้างต้น

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

ก. ข้อมูลภาพ

ข. ขอบภาพที่ถูกต้องสมบูรณ์

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

ค. ขอบภาพที่มีตำแหน่งผิด

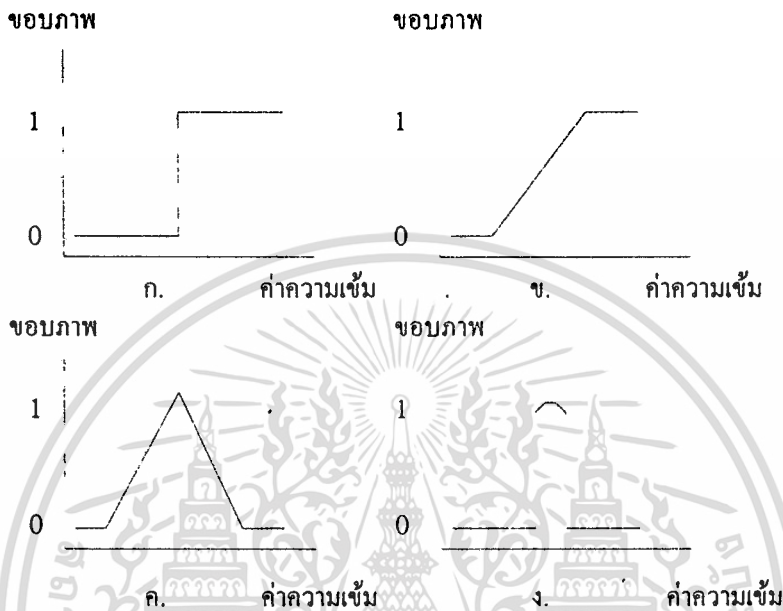
ง. ขอบภาพหนา (กว้าง 2 จุด)

จ. ขอบภาพที่ไม่ต่อเนื่อง

รูปที่ 1.1 แสดงลักษณะขอบภาพแบบต่างๆ

รูปแบบของขอบภาพที่ปรากฏในรูปภาพโดยทั่วไป จะประกอบไปด้วยขอบภาพหลายชนิด ดังแสดงในรูปที่ 1.2 ซึ่งรูปที่ 1.2ก เป็นขอบภาพที่มีการเปลี่ยนแปลงอย่างทันทีทันใด ขอบภาพชนิดนี้ส่วนมากไม่วากริณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะอยู่ในภาพที่มนุษย์จำลองขึ้นมา ส่วนขอบภาพในรูปภาพโดยทั่วไปจะเป็นขอบภาพที่มีการเปลี่ยนแปลงค่าความเข้มที่ละน้อย ดังแสดงในรูปที่ 1.2ข, รูปที่ 1.2ค ขอบภาพที่ค่าความเข้มค่อยๆ เพิ่มขึ้นหรือลดลงทีละน้อยในลักษณะของรูปหน้าจั่ว, และรูปที่ 1.2ง แสดงขอบภาพที่เป็นเส้น



รูปที่ 1.2 แสดงขอบภาพชนิดต่างๆ

การเปลี่ยนแปลงหรือความไม่ต่อเนื่องในฟังก์ชันค่าความเข้มของจุดภาพที่เกิดจากการส่องสว่าง หรือ ลักษณะทางกายภาพของตัววัตถุเอง (เช่น ลักษณะของพื้นผิว, รูปทรง) คุณสมบัติเหล่านี้จะเป็นปัจจัยพื้นฐานที่สำคัญของรูปภาพที่จะบอกถึงขอบเขตทางกายภาพของวัตถุที่อยู่ในภาพนั้น ฟังก์ชันจุดภาพ $g(x,y)$ ที่เป็นขอบภาพจะมีการเปลี่ยนแปลงจากค่าความเข้มหนึ่ง ไปยังอีกค่าความเข้มหนึ่ง โดยค่าความเข้มนั้นจะต้องแตกต่างกันอย่างมีนัยสำคัญ คำว่าแตกต่างกันอย่างมีนัยสำคัญของค่าความเข้มในที่นี้จะมีตัววัดมากมายหลายวิธี และตัววัดความแตกต่างเหล่านี้จะเรียกว่าตัวดำเนินการหาขอบภาพซึ่งจะอธิบายแต่ละวิธีดังนี้

1.2.1 การขอบภาพโดยใช้อนุพันธ์อันดับที่หนึ่ง

วิธีแรกที่จะเสนอสำหรับการหาจุดภาพที่เป็นขอบคือ การแปลงเกรเดียนต์แบบไม่ต่อเนื่องบนข้อมูลภาพเชิงตัวเลข [7] ซึ่งเป็นการหาค่าตำแหน่งค่าความเข้มของจุดภาพที่มีการเปลี่ยนแปลง และเกรเดียนต์ $\nabla f(x,y)$ ที่เป็นฟังก์ชันต่อเนื่องในระนาบสองมิติซึ่งสามารถกำหนดได้โดย

$$\nabla f(x,y) = \left(\frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} \right) \quad (1.1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีแรกที่จะเสนอสำหรับการหาจุดภาพที่เป็นขอบคือ การแปลงเกรเดียนต์แบบไม่ต่อเนื่องบนข้อมูลภาพเชิงตัวเลข [7] ซึ่งเป็นการหาตำแหน่งค่าความเข้มของจุดภาพที่มีการเปลี่ยนแปลง และเกรเดียนต์ $\nabla f(x, y)$ ที่เป็นฟังก์ชันต่อเนื่องในระนาบสองมิติซึ่งสามารถกำหนดได้โดย

$$\nabla f(x, y) = \left(\frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} \right) \quad (1.1)$$

สำหรับขนาดสามารถกำหนดได้ดังนี้

$$|\nabla f(x, y)| = \sqrt{\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2} \quad (1.2)$$

ส่วนทิศทางของเวกเตอร์เกรเดียนต์คือ

$$\theta = \tan^{-1} \left(\frac{\partial f / \partial y}{\partial f / \partial x} \right) \quad (1.3)$$

เนื่องจากการหาขอบภาพเป็นการประมวลผลแบบไม่ต่อเนื่อง ดังนั้นจึงต้องใช้อนุพันธ์ย่อยแบบไม่ต่อเนื่องตามทิศทางที่ตั้งฉากกับแกน x และ y [9] ซึ่งสามารถกำหนดได้ดังนี้

$$\begin{aligned} \nabla_x g(x, y) &= g(x, y) - g(x-1, y) \\ \nabla_y g(x, y) &= g(x, y) - g(x, y-1) \end{aligned} ; \begin{bmatrix} x-1, y & x, y \\ x-1, y-1 & x, y-1 \end{bmatrix} \quad (1.4)$$

และสำหรับทิศทางหาได้จากสมการที่ (1.5)

$$\nabla g(x, y) = g(x, y) \cos \theta + g(x, y) \sin \theta \quad (1.5)$$

ส่วนขนาดโดยประมาณของเกรเดียนต์ $g(x, y)$ สามารถกำหนดได้ดังนี้

$$|\nabla g(x, y)| = \sqrt{(\nabla_x g(x, y))^2 + (\nabla_y g(x, y))^2} \quad (1.6)$$

การหาขอบภาพโดยใช้เกรเดียนต์ในทางปฏิบัติจะมีลักษณะที่แตกต่างกันไปเช่น วิธีของ Roberts [9] สามารถเขียนเป็นสมการได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$W_1 = \frac{1}{2+\sqrt{2}} \begin{bmatrix} 1 & 0 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} \\ 1 & 0 & -1 \end{bmatrix} \quad W_2 = \frac{1}{2+\sqrt{2}} \begin{bmatrix} -1 & -\sqrt{2} & -1 \\ 0 & 0 & 0 \\ 1 & \sqrt{2} & 1 \end{bmatrix}$$

และ Kirsch [9] ซึ่งเป็นเกรเดียนต์ที่มีทิศทางแปดทิศโดยแต่ละทิศทางมีมุมต่างกัน 45 องศา ดังแสดงข้างล่าง

$$W_1 = \begin{bmatrix} 3 & 3 & -5 \\ 3 & 0 & -5 \\ 3 & 3 & -5 \end{bmatrix}, \quad W_2 = \begin{bmatrix} 3 & -5 & -5 \\ 3 & 0 & -5 \\ 3 & 3 & 3 \end{bmatrix}, \quad W_3 = \begin{bmatrix} -5 & -5 & -5 \\ 3 & 0 & 3 \\ 3 & 3 & 3 \end{bmatrix}, \quad W_4 = \begin{bmatrix} -5 & -5 & 3 \\ -5 & 0 & 3 \\ 3 & 3 & 3 \end{bmatrix}$$

$$W_5 = \begin{bmatrix} -5 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & 3 & 3 \end{bmatrix}, \quad W_6 = \begin{bmatrix} 3 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & -5 & 3 \end{bmatrix}, \quad W_7 = \begin{bmatrix} 3 & 3 & 3 \\ 3 & 0 & 3 \\ -5 & -5 & -5 \end{bmatrix}, \quad W_8 = \begin{bmatrix} 3 & 3 & 3 \\ 3 & 0 & -5 \\ 3 & -5 & -5 \end{bmatrix}$$

การนำวินโดว์ W ของแต่ละวิธีมาหาขอบภาพ จะต้องนำวินโดว์ทุกตัวของวิธีนั้นๆ มาคอนโวลูชันหรือการประสาน (Convolution) [11] กับข้อมูลภาพ ซึ่งสามารถแสดงในรูปของสมการได้ดังนี้

$$h_k(x, y) = \sum_i \sum_j w_k(i, j) g(x+i, y+j) \quad (1.10)$$

ที่ $h_k(x, y)$ คือจุดภาพ ณ ตำแหน่ง x, y ที่ผ่านการคอนโวลูชันแล้วด้วยวินโดว์ที่ k
 $w_k(i, j)$ เป็นสมาชิกตัวที่ i, j ของวินโดว์ตัวที่ k

เมื่อดำเนินการคอนโวลูชันระหว่างข้อมูลภาพกับวินโดว์ของวิธีที่จะนำมาหาขอบภาพครบทุกวินโดว์แล้วก็นำ $h_k(x, y)$ สำหรับ k ทุกตัวมาทำการเปรียบเทียบกัน ณ พิกัด x, y เดียวกันเพื่อเลือกค่าที่สูงที่สุด และนำค่าที่ได้มาเทียบกันกับค่าเทรชโฮลด์คือ ถ้าน้อยกว่าค่าเทรชโฮลด์จุดภาพนั้นก็ไม่ใช่ขอบภาพ อัลกอริทึมที่ 1-1 จะแสดงขั้นตอนต่างๆ ในการหาขอบภาพที่ได้อธิบายมาแล้วข้างต้น

อัลกอริทึมที่ 1-1. แสดงขั้นตอนการหาขอบภาพโดยใช้วินโดว์มาคอนโวลูชันกับข้อมูลภาพ

กำหนดให้ T เป็นค่าเทรชโฮลด์ที่ผู้ใช้จะต้องกำหนดเอง
 K เป็นจำนวนวินโดว์ที่จะนำมาหาขอบภาพ เช่น Roberts, Prewitt, Sobel, และ Frei-Chen มี $K=2$ ส่วนวิธีของ Kirsch มี $K=8$ เป็นต้น

Mw, Nw เป็นจำนวนแถวและจำนวนคอลัมน์ของวินโดว์ที่ k ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 $e(x, y)$ เป็นสมาชิกของเซตของขอบภาพ E

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void Edge_Detection_by_Convolution()
{
    for(int x = 0; x < M; x++)
        for(int y = 0; y < N; y++)
            {
                h_max = 0;
                for(int k = 0; k < K; k++)
                    {
                        for(int i = 1; i < Mw; i++)
                            for(int j = 1; j < Nw; j++)
                                h_k += w_k[i][j] * g[x+i][y+j];
                                if (h_k > h_max)
                                    h_max = h_k;
                    }
                if (h_max > T)
                    e[x][y] = EDGE;
                else
                    e[x][y] = NO_EDGE;
            }
}

```

สำหรับทิศทางของเกรเดียนต์ (θ) จะสอดคล้องกับ ทิศทางของวินโดว์ที่ k ที่ทำให้ $h_k(x, y)$ ในสมการที่ (1.10) มีค่าสูงสุด โดยทิศทางของวินโดว์ในที่นี้จะขึ้นอยู่กับวิธีที่เลือกใช้ เช่น วิธีของ Prewitt, Sobel, และ Frei-Chen นั้น W_1 จะใช้หาขอบภาพที่อยู่ในแนวตั้ง ($\theta = 90^\circ$) ส่วน W_2 จะใช้หาขอบภาพในแนวนอน ($\theta = 180^\circ$) ส่วนวิธีของ Kirsch จะมีทั้งหมดแปดทิศทางด้วยกันคือ 0, 45, 90, 135, 180, 225, 270, และ 315 องศา

1.2.2 การหาขอบภาพโดยใช้ออนุพันธ์อันดับที่สอง

การหาขอบภาพโดยวิธีนี้จะไม่มีการพิจารณาทิศทางของขอบภาพ แต่จะสนใจเฉพาะขนาดของความเป็นขอบภาพ (เกรเดียนต์) เท่านั้น ดังนั้นวินโดว์ที่ใช้หาขอบภาพโดยวิธีนี้จะมีเพียงวินโดว์เดียว ซึ่งมีวิธีต่างๆ ดังต่อไปนี้ที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2.2.1 ตัวดำเนินการ Laplacian [9]

ตัวดำเนินการหาขอบภาพ Laplacian ของฟังก์ชันสองมิติ $g(x,y)$ ซึ่งเป็นฟังก์ชันไม่ต่อเนื่องที่จุด (x, y) ที่ได้มาจากการประมาณอนุพันธ์เชิงเส้นของฟังก์ชันต่อเนื่องดังสมการที่ (1.11)

$$\nabla^2 f(x, y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (1.11)$$

สำหรับในกรณีของเกรเดียนต์ของ $\Delta^2 g(x, y)$ ที่ได้จากการประมาณสมการที่ (1.11) เพื่อใช้ในการหาขอบภาพภายในพื้นที่ขนาด 3×3 จุด สามารถกำหนดได้ดังนี้

$$\nabla^2 g(x, y) = 4g(x, y) - (g(x+1, y) + g(x-1, y) + g(x, y+1) + g(x, y-1)) \quad (1.12)$$

ในการกำหนด Laplacian เชิงตัวเลข สัมประสิทธิ์ที่คุณอยู่กับพิกัดที่ต้องการหาขอบภาพจะต้องมีค่าเป็นบวกเสมอส่วนสัมประสิทธิ์ในตำแหน่งอื่นๆ อาจจะมีค่าเป็นบวกหรือลบก็ได้ เนื่องจากอนุพันธ์ของ Laplacian ผลรวมของสัมประสิทธิ์ทั้งหมดจะต้องเป็นศูนย์เสมอ ซึ่งในกรณีนี้ถ้าจุดภาพที่อยู่ในพื้นที่ขนาด 3×3 มีค่าเท่ากันหมดเมื่อคอนโวลูชันกับตัวดำเนินการนี้ผลลัพธ์ที่ได้ก็คือศูนย์ ซึ่งแสดงว่าจุดนี้ไม่ใช่ขอบภาพอย่างแน่นอนในรูปที่ 1.3 แสดงวินโดว์ของ Laplacian ในรูปแบบต่างๆ ทั้งที่ใช้จุดภาพในบริเวณใกล้เคียง 4 จุดดังรูปที่ 1.3.1 และ 8 จุดดังรูปที่ 1.3.2 กับ 1.3.3 ซึ่งจะมีผลต่อขอบภาพต่างกัน

$$1) \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad 2) \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad 3) \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

รูปที่ 1.3 แสดงตัวดำเนินการหาขอบภาพของ Laplacian

การหาขอบภาพจะเลือกวินโดว์ตัวใดตัวหนึ่งจากรูปที่ 1.3 แล้วนำไปคอนโวลูชันโดยอัลกอริทึมที่ 1-1

1.2.2.2 Marr และ Hildreth [10]

การหาขอบภาพโดยวิธีนี้จะใช้ตัวกรองสัญญาณชนิดที่ให้ความถี่บางช่วงผ่าน (Band Pass) ซึ่งใช้ฟังก์ชันเกาส์เซียน (Gaussian) ในระนาบสองมิติ ดังสมการที่ (1.13)

$$w(i, j) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{i^2+j^2}{2\sigma^2}} \quad (1.13)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์อื่นใด
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้เผยแพร่ลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ σ เป็นค่าส่วนเบี่ยงเบนมาตรฐานที่บอกถึงความกว้างของช่วงความถี่ที่จะให้ผ่าน

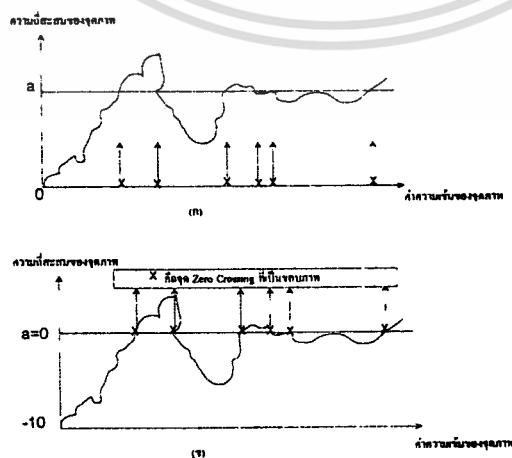
ณ. จุดใดๆ ในภาพที่ค่าความเข้มมีการเปลี่ยนแปลงอย่างทันทีทันใด เมื่อเราหาอนุพันธ์อันดับที่สอง ตรงจุดนี้จะได้ผลลัพธ์เป็นศูนย์ หรือที่เรียกว่าซีโรครอสซิง (Zero Crossing) ซึ่งเป็นกรณีเฉพาะของระดับที่ ตกคร่อมที่จุด a ใดๆ (Level-Crossing [11]) โดยในการนิยามของ Zero Crossing นั้น a มีค่าเท่ากับศูนย์ ดัง แสดงในรูปที่ 1.4 จากแนวความคิดอันนี้ Marr และ Hildreth จึงนำมาสร้างเป็นตัวดำเนินการหาขอบภาพ โดยการหาจุดที่เป็น Zero Crossing ของอนุพันธ์อันดับที่สองของค่าความเข้มข้อมูลภาพ ซึ่งสามารถเขียน เป็นสมการได้ดังนี้

$$\begin{aligned} h(x, y) &= \frac{\partial^2 [w(i, j) \otimes g(x, y)]}{\partial i \partial j} \\ &= \frac{\partial^2 w(i, j)}{\partial i \partial j} \otimes g(x, y) \end{aligned} \quad (1.14)$$

ที่ \otimes เป็นตัวดำเนินการคอนโวลูชัน

$$\text{และ} \quad \frac{\partial^2 w(i, j)}{\partial i \partial j} = \frac{-1}{\pi \sigma^4} \left(1 - \frac{i^2 + j^2}{2\sigma^2} \right) e^{-\frac{(i^2 + j^2)}{2\sigma^2}} \quad (1.15)$$

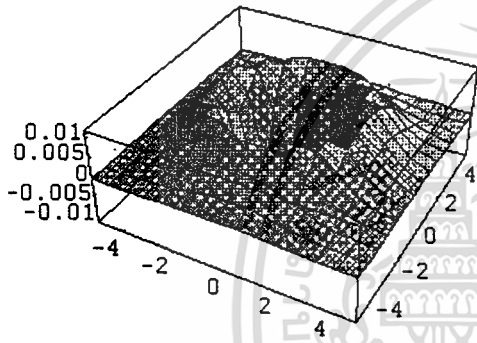
ค่าของ i และ j จะขึ้นอยู่กับขนาดของวินโดว์ที่นำมาใช้ในการหาขอบภาพ เช่น ถ้าใช้วินโดว์ขนาด 3×3 ค่าของ i และ j จะอยู่ในช่วง -1 ถึง 1 และถ้าใช้วินโดว์ขนาด 11×11 i และ j จะมีค่าอยู่ในช่วง -5 ถึง 5 เป็นต้น จากสมการที่ (1.15) ทำให้เราสามารถกำหนดวินโดว์เพื่อใช้ในการหาขอบภาพได้ และนำวินโดว์ที่ได้ไปคอนโวลูชันกับข้อมูลภาพโดยใช้อัลกอริทึม 1-1 สำหรับวินโดว์ขนาด 11×11 และลักษณะรูปร่างทาง กราฟฟิกส์ของมันแสดงในรูปที่ 1.5



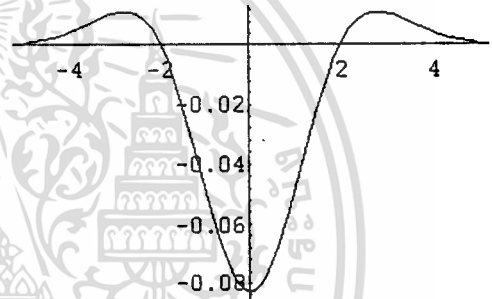
เอกสารรูปที่ 1.4 (ก) ระดับที่ตกคร่อมที่จุด a ใดๆ, (ข) Zero-Crossing ที่ทำให้หาขอบภาพได้ง่ายกว่า (ก) ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.81e-06	2.25e-05	1.09e-04	3.25e-04	6.14e-04	7.57e-04	6.14e-04	3.25e-04	1.09e-04	2.25e-05	2.81e-06
2.25e-05	1.69e-04	7.57e-04	2.07e-03	3.61e-03	4.31e-03	3.61e-03	2.07e-03	7.57e-04	1.69e-04	2.25e-05
1.09e-04	7.57e-04	3.01e-03	6.96e-03	1.00e-02	1.08e-02	1.00e-02	6.96e-03	3.01e-03	7.57e-04	1.09e-04
3.25e-04	2.07e-03	6.96e-03	1.12e-02	6.37e-03	6.09e-04	6.37e-03	1.12e-02	6.96e-03	2.07e-03	3.25e-04
6.14e-04	3.61e-03	1.00e-02	6.37e-03	-2.44e-02	-4.78e-02	-2.44e-02	6.37e-03	1.00e-02	3.61e-03	6.14e-04
7.57e-04	4.31e-03	1.08e-02	6.09e-04	-4.78e-02	-8.28e-02	-4.78e-02	6.09e-04	1.08e-02	4.31e-03	7.57e-04
6.14e-04	3.61e-03	1.00e-02	6.37e-03	-2.44e-02	-4.78e-02	-2.44e-02	6.37e-03	1.00e-02	3.61e-03	6.14e-04
3.25e-04	2.07e-03	6.96e-03	1.12e-02	6.37e-03	6.09e-04	6.37e-03	1.12e-02	6.96e-03	2.07e-03	3.25e-04
1.09e-04	7.57e-04	3.01e-03	6.96e-03	1.00e-02	1.08e-02	1.00e-02	6.96e-03	3.01e-03	7.57e-04	1.09e-04
2.25e-05	1.69e-04	7.57e-04	2.07e-03	3.61e-03	4.31e-03	3.61e-03	2.07e-03	7.57e-04	1.69e-04	2.25e-05
2.81e-06	2.25e-05	1.09e-04	3.25e-04	6.14e-04	7.57e-04	6.14e-04	3.25e-04	1.09e-04	2.25e-05	2.81e-06

(ก)



(ข)



(ค)

รูปที่ 1.5 ตัวดำเนินการหาขอบภาพของ Marr และ Hildreth ที่ $\sigma = 1.4$ (ก) วินโดว์ขนาด 11×11 (ข) แสดงรูปร่างทางกราฟฟิกส์ของวินโดว์ในรูป (ก), (ค) เป็นรูปด้านข้างของรูป (ก)

วิธีนี้ได้มีผู้นำมาพัฒนาต่ออีกมากในที่นี้จะกล่าวถึงเฉพาะที่วิธีที่นิยมใช้กัน โดยทั่วไปคือ วิธีของ Haralick [12] และวิธีของ Canny [13] โดยตัวดำเนินการหาขอบภาพของ Haralick สามารถกำหนดได้ดังนี้

$$w(i, j) = A \left(1 - k \frac{i^2 + j^2}{\sigma^2} \right) e^{-\frac{i^2 + j^2}{2\sigma^2}} \tag{1.16}$$

เมื่อ A เป็นค่าคงที่ที่ใช้ในการเพิ่มค่าที่อยู่ภายในวินโดว์ เนื่องจากเทอมที่ถัดจากค่าคงที่ A มีค่าน้อยมาก และค่า k เป็นค่าคงที่ที่ทำให้ผลรวมของค่าถ่วงน้ำหนักแต่ละตัวในวินโดว์มีค่าเท่ากับศูนย์ ($\sum_{i,j} w(i, j) = 0$) ใน

รูปที่ 1.6 เป็นวินโดว์ของ Haralick ที่มีขนาด 11×11 โดย $k = 0.5$, $\sigma = 1.4$, และ $A = 1$ สำหรับตัวดำเนินการหาขอบภาพของ Canny [13] ได้ใช้ตัวกรองสัญญาณของเกาส์เซียนแบบสองมิติตั้งสมการที่เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านกักรนี้ (1.17)

ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$G(i, j) = e^{-\left(\frac{i^2 + j^2}{2\sigma^2}\right)} \quad (1.17)$$

และสามารถหาค่าแหน่งของขอบภาพได้โดยสมการที่ (1.18)

$$\text{ถ้า } \frac{\partial^2 G(i, j)}{\partial n^2} \otimes g(x, y) = 0 \quad \text{แสดงว่าเป็นขอบภาพ} \quad (1.18)$$

$$\frac{\partial G(i, j)}{\partial n} = n \cdot \nabla G(i, j) \quad (1.19)$$

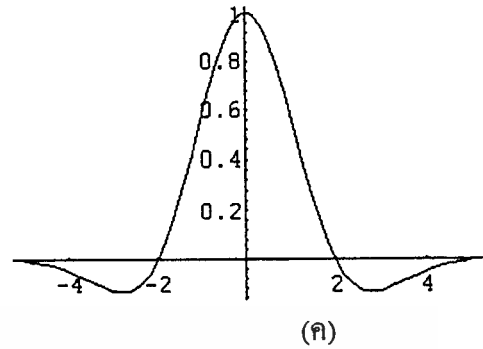
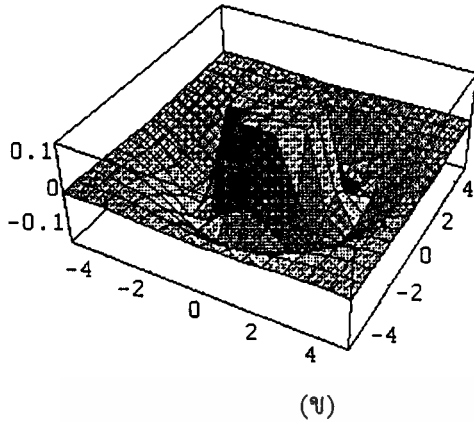
โดยที่

$$n = \frac{\nabla(G(i, j) \otimes g(x, y))}{|\nabla(G(i, j) \otimes g(x, y))|} \quad (1.20)$$

เมื่อ n เป็นทิศทางที่ตั้งฉากกับจุดที่เป็นขอบภาพจะเห็นว่าสมการที่ (1.18) จะมีลักษณะคล้ายกับสมการที่ (1.14) ขอบภาพที่ได้จากวิธีของ Canny จะมีตำแหน่งถูกต้อง และวิธีนี้ยังใช้ได้กับภาพที่มีสัญญาณรบกวนชนิดที่เป็นความถี่สูง (White Noise) ด้วย

-3.39e-05	-2.71e-04	-1.31e-03	-3.92e-03	-7.42e-03	-9.14e-03	-7.42e-03	-3.92e-03	-1.31e-03	-2.71e-04	-3.39e-05
-2.71e-04	-2.04e-03	-9.14e-03	-2.50e-02	-4.36e-02	-5.20e-02	-4.36e-02	-2.50e-02	-9.14e-03	-2.04e-03	-2.71e-04
-1.31e-03	-9.14e-03	-3.64e-02	-8.40e-02	-1.21e-01	-1.30e-01	-1.21e-01	-8.40e-02	-3.64e-02	-9.14e-03	-1.31e-03
-3.92e-03	-2.50e-02	-8.40e-02	-1.35e-01	-7.69e-02	-7.36e-03	-7.69e-02	-1.35e-01	-8.40e-02	-2.50e-02	-3.92e-03
-7.42e-03	-4.36e-02	-1.21e-01	-7.69e-02	2.94e-01	5.77e-01	2.94e-01	-7.69e-02	-1.21e-01	-4.36e-02	-7.42e-03
-9.14e-03	-5.20e-02	-1.30e-01	-7.36e-03	5.77e-01	1.00e+00	5.77e-01	-7.36e-03	-1.30e-01	-5.20e-02	-9.14e-03
-7.42e-03	-4.36e-02	-1.21e-01	-7.69e-02	2.94e-01	5.77e-01	2.94e-01	-7.69e-02	-1.21e-01	-4.36e-02	-7.42e-03
-3.92e-03	-2.50e-02	-8.40e-02	-1.35e-01	-7.69e-02	-7.36e-03	-7.69e-02	-1.35e-01	-8.40e-02	-2.50e-02	-3.92e-03
-1.31e-03	-9.14e-03	-3.64e-02	-8.40e-02	-1.21e-01	-1.30e-01	-1.21e-01	-8.40e-02	-3.64e-02	-9.14e-03	-1.31e-03
-2.71e-04	-2.04e-03	-9.14e-03	-2.50e-02	-4.36e-02	-5.20e-02	-4.36e-02	-2.50e-02	-9.14e-03	-2.04e-03	-2.71e-04
-3.39e-05	-2.71e-04	-1.31e-03	-3.92e-03	-7.42e-03	-9.14e-03	-7.42e-03	-3.92e-03	-1.31e-03	-2.71e-04	-3.39e-05

(ก)



รูปที่ 1.6 ตัวดำเนินการหาขอบภาพของ Haralick ที่ $\sigma = 1.4$ (ก) วินโดว์ขนาด 11×11 (ข) แสดงรูปร่างทางกราฟฟิกของวินโดว์ในรูป (ก), (ค) เป็นรูปด้านข้างของรูป (ข)

1.2.3 การหาขอบภาพโดยใช้ฟังก์ชัน Gabor

ฟังก์ชัน Gabor ในที่นี้เป็นฟังก์ชันที่ใช้ในการกรองสัญญาณภาพ ซึ่งเป็นแบบจำลองการคำนวณที่พัฒนามาจากการทดลองทางด้านชีววิทยาและจิตวิทยาที่เกี่ยวข้องกับระบบการมองเห็น ที่มีส่วนใกล้เคียงกับสัตว์เลี้ยงลูกด้วยนม [14] โดยทั่วไปตัวกรองสัญญาณชนิดฟังก์ชัน Gabor นี้จะมีผลทำให้ความไม่แน่นอนในข้อมูลมีค่าต่ำสุด [15] จากผลอันนี้ทำให้ฟังก์ชัน Gabor ถูกนำไปใช้ในการประมวลผลระดับต่ำในหลายขั้นตอนด้วยกัน เช่น การปรับปรุงภาพ [16], การหาขอบภาพ [14], การแบ่งแยกวัตถุภายในภาพโดยเนื้อภาพ (Texture Classification [17-18]) เป็นต้น

ฟังก์ชัน Gabor ได้มาจากการรวมกันของฟังก์ชันเกาส์เซียน และไซน์ ดังสมการที่ (1.21) ซึ่งเป็นฟังก์ชันแบบหนึ่งมิติ

$$S(x) = e^{\left(-\frac{1}{2} \frac{(x-x_0)^2}{\sigma} + j\omega_x x \right)} \quad (1.21)$$

ที่ σ และ x_0 เป็นส่วนเบี่ยงเบนมาตรฐานและค่าเฉลี่ยของฟังก์ชันเกาส์เซียนที่เกิดขึ้นภายในหนึ่งรอบ และ ω_x คือความถี่ของสัญญาณไซน์

ในสมการที่ (1.21) สามารถแยกได้สองส่วนคือ ส่วนที่เป็นฟังก์ชันคี่กับส่วนที่เป็นฟังก์ชันคู่ (โดยที่จุดศูนย์กลางหรือค่าเฉลี่ยของแต่ละรอบเท่ากับศูนย์ ($x_0 = 0$)) ซึ่งแสดงในสมการที่ (1.22) และ (1.23) ตามลำดับ

$$S_{\text{odd}}(x) = e^{-\frac{1}{2} \frac{(x)^2}{\sigma}} \sin(\omega_x x) \quad (1.22)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$S_{even}(x) = e^{-\frac{1}{2}\left(\frac{x}{\sigma}\right)^2} \cdot \cos(\omega_x x) \quad (1.23)$$

ในการหาขอบภาพจะใช้ฟังก์ชันที่ [14] เนื่องจากฟังก์ชันที่เป็นสองมิติสามารถปรับทิศทางให้ตั้งฉากกับขอบภาพได้ (ดังแสดงในรูปที่ 1.7) ซึ่งจะทำให้มีการตอบสนองสูงสุดต่อสัญญาณที่มีการเปลี่ยนแปลงอย่างทันทีทันใด แต่ฟังก์ชันคู่จะสมมาตรที่จุดกำเนิดทำให้ขอบภาพที่ได้มีตำแหน่งที่ผิดไปจากความเป็นจริง โดยขอบภาพจริงๆ มีเพียงเส้นเดียวแต่ผลที่ได้จาก Gabor ฟังก์ชันคู่ขอบภาพจะถูกแบ่งออกเป็นสองเส้นขนานกันไป

ตัวกรองสัญญาณที่เป็นฟังก์ชันคู่ของ Gabor ชนิดสองมิติ โดยมีทิศทางของตัวกรองสัญญาณตามมุม θ สามารถกำหนดได้โดย

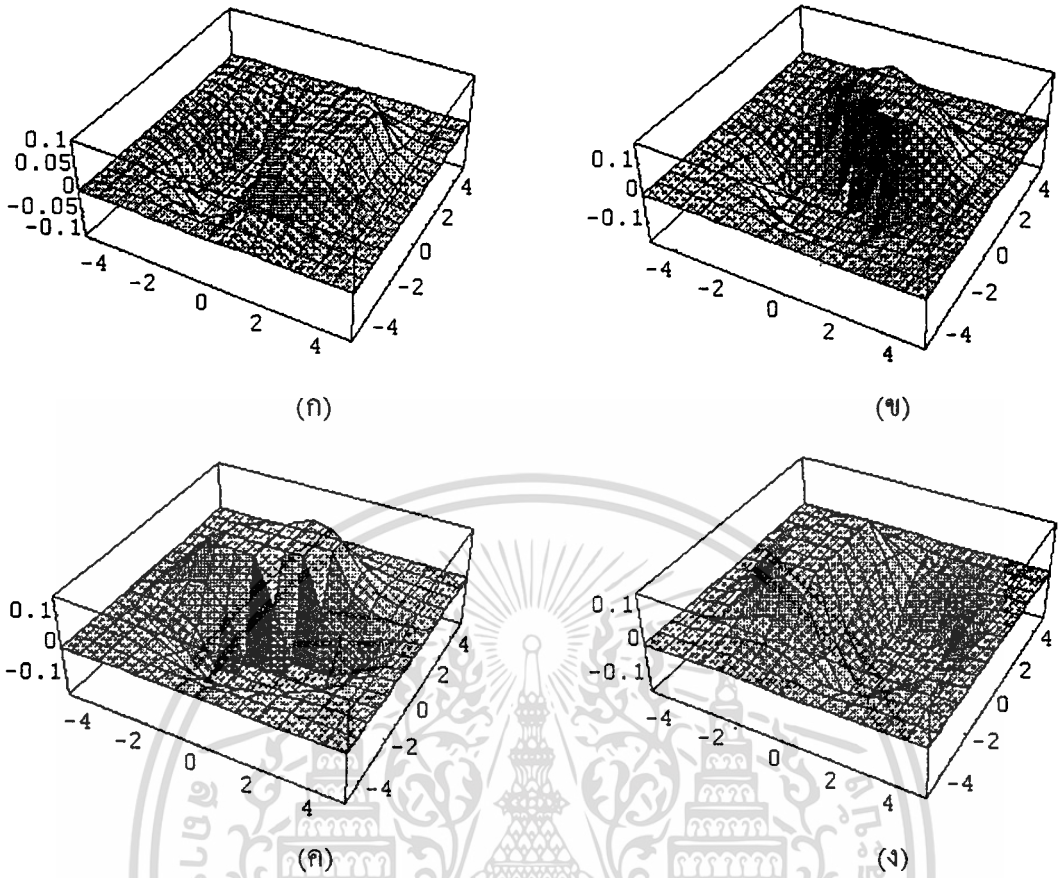
$$S_{odd}(x, y) = e^{-\left(\frac{x^2 + y^2}{2\sigma^2}\right)} \sin(\omega(x \cos \theta + y \sin \theta)) \quad (1.24)$$

ที่ ω เป็นขนาดของความถี่เชิงมุมของเวกเตอร์ในระนาบสองมิติ (สำหรับลักษณะของฟังก์ชันรูปที่ 1.7)

ในการหาขอบภาพ การเลือกทิศทางของตัวกรองสัญญาณให้เหมาะสมกับจุดภาพในตำแหน่งต่าง ๆ นั้นเป็นสิ่งสำคัญ แต่ในทางปฏิบัติจะทำได้ยากมาก ในที่นี้จะพิจารณาถึงผลกระทบของทิศทางของตัวกรองสัญญาณภาพที่เป็นฟังก์ชันคู่ของ Gabor แบบสองมิติในสมการที่ (1.24) ซึ่งมีทิศทางตามมุม θ โดยจะพิจารณาจากจุดศูนย์กลาง $(x = 0, y = 0)$ ของขอบภาพชนิดที่มีการเปลี่ยนแปลงค่าความเข้มอย่างทันทีทันใด (ซึ่งในที่นี้แทนโดย $U(x, y)$) ตามแกน x ซึ่งกำหนดได้โดย

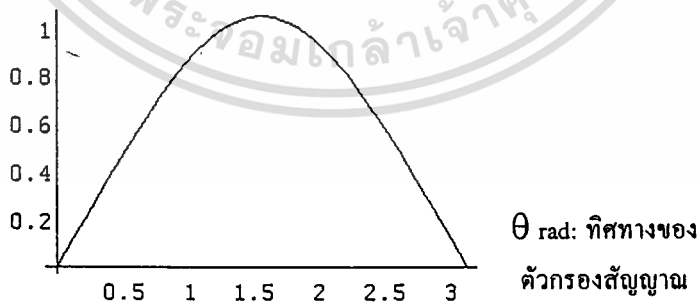
$$\begin{aligned} U(x, y) \otimes S_{odd}(x, y) \Big|_{(x=0, y=0)} &= \int_{-\infty}^0 \int_0^{+\infty} S_{odd}(x, y) dx dy \\ &= \left(\omega \sin(\theta) \sigma^2 - \frac{(\omega \sin(\theta))^3 \sigma^4}{3} + \frac{(\omega \sin(\theta))^5 \sigma^6}{15} - \frac{(\omega \sin(\theta))^7 \sigma^8}{105} \right) \times \\ &\quad \left(\sigma \sqrt{2\pi} e^{-\left(\frac{-\omega^2 \cos^2(\theta) \sigma^2}{2}\right)} \right) \end{aligned} \quad (1.25)$$

จากสมการที่ (1.25) สามารถนำมาแสดงในรูปของกราฟที่แสดงถึงผลการตอบสนองของทิศทาง θ กับผลคูณของ σ และ ω ดังแสดงในรูปที่ 1.8



รูปที่ 1.7 ฟังก์ชัน Gabor แบบสองมิติ ที่ $\sigma = 3/2$, $\omega = 2/3$, (ก) $\theta = 0^\circ$, (ข) $\theta = 45^\circ$ (ค) $\theta = 90^\circ$ (ง) $\theta = 135^\circ$

ผลการตอบสนองของ
ตัวกรองสัญญาณ



รูปที่ 1.8 ผลของทิศทางของตัวกรอง θ ที่มีผลตอบสนองกับ $\sigma \cdot \omega$

จากรูปที่ 1.8 จะเห็นว่าการตอบสนองของตัวกรองสัญญาณที่เป็นฟังก์ชัน Gabor แบบสองมิติที่มีผลต่อขอบภาพแบบที่มีการเปลี่ยนแปลงค่าอย่างทันทีทันใดนั้น ทิศทางของตัวกรองสัญญาณที่แท้จริง (คือทิศทางที่ตั้งฉากกับทิศทางของขอบภาพ) ทำให้ตัวกรองสัญญาณมีการตอบสนองสูงสุด ในกรณีที่เราจำกัดให้ $\sigma \cdot \omega$ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นค่าคงที่ การตอบสนองของตัวกรองสัญญาณต่อทิศทางที่ไม่ใช่ทิศทางที่แท้จริงจะมีผลการตอบสนองน้อยกว่าของตัวกรองสัญญาณ ที่เป็นทิศทางที่แท้จริงของขอบภาพ ดังนั้นฟังก์ชัน Gabor ที่เป็นตัวกรองสัญญาณแบบสองมิติ และมีทิศทางตั้งฉากกับขอบภาพโดยมี $\sigma \cdot \omega \approx 1$ จะทำให้ตัวกรองสัญญาณนี้มีการตอบสนองต่อขอบภาพดีที่สุด

วิธีหนึ่งที่เป็นไปได้ในการหาทิศทางที่แท้จริงสำหรับการหาขอบภาพคือ การใช้ตัวกรองสัญญาณหลายๆ ตัว โดยแต่ละตัวมีทิศทางต่างๆ กัน แต่วิธีนี้จะทำให้เสียเวลาในการคำนวณมาก ทางเลือกหนึ่งที่มีประสิทธิภาพมากและมีความถูกต้องพอสมควรคือ การประมาณทิศทางของขอบภาพสำหรับแต่ละจุดที่ต้องการหาโดยใช้ตัวกรองสัญญาณเกาส์เซียนแบบสองมิติ (ดังสมการที่ (1.17)) ไปคอนโวลูชันกับภาพและหามุมของเกรเดียนต์ ซึ่งมุมที่ได้จะเป็นมุมที่เป็นทิศทางของขอบภาพ เมื่อเอามุมนี้ไปสร้างตัวกรองสัญญาณจะต้องบวกด้วย $\pi/2$ เพื่อให้ทิศทางของตัวกรองสัญญาณตั้งฉากกับขอบภาพ

ขั้นตอนการหาขอบภาพโดยใช้ฟังก์ชัน Gabor

1. ใช้ฟังก์ชันเกาส์เซียนในสมการที่ (1.17) คอนโวลูชันกับข้อมูลภาพดังสมการข้างล่าง

$$R(x, y) = G(i, j, \frac{\sigma}{\sqrt{2}}) \otimes g(x, y) \quad (1.26)$$

2. นำ $R(x, y)$ ไปหาทิศทางของขอบภาพโดยใช้เกรเดียนต์ ซึ่งสามารถหาได้ดังนี้ สมมติให้ $R(x, y)$ เป็นเมตริกซ์ขนาด 3×3 ดังนั้นเกรเดียนต์ตามทิศทางของแกน x (G_x) และ แกน y (G_y) กำหนดได้ดังนี้

$$R = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{bmatrix}$$

$$G_x = (a_7 + a_8 + a_9) - (a_1 + a_2 + a_3) \quad (1.27)$$

$$G_y = (a_3 + a_6 + a_9) - (a_1 + a_4 + a_7) \quad (1.28)$$

ซึ่งทิศทางของขอบภาพเมื่อเทียบกับแกน x สามารถกำหนดได้ดังนี้

$$\alpha = \tan^{-1} \left(\frac{G_x}{G_y} \right) \quad (1.29)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่ 1.1 สมมติให้ $R = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ และทิศทางของขอบภาพสามารถหาได้ดังนี้

$$\begin{aligned} G_x &= (1+1+1) - (0+0+1) \\ &= 2 \\ G_y &= (1+1+1) - (0+0+1) \\ &= 2 \end{aligned}$$

ดังนั้นทิศทางของขอบภาพเมื่อเทียบกับแกน x คือ

$$\begin{aligned} \alpha &= \tan^{-1}\left(\frac{2}{2}\right) \\ &= 45^\circ \end{aligned}$$

3. กำหนดทิศทางของตัวกรองสัญญาณ

$$\theta = \alpha + 90 \quad (1.30)$$

4. สร้างวินโดว์ของตัวกรองสัญญาณโดยใช้สมการที่ (1.24) แล้วนำมาคอนโวลูชันกับ $R(x, y)$ ดังสมการข้างล่าง

$$h(x, y) = R(x, y) \otimes S_{odd}\left(x, y, \frac{\sigma}{\sqrt{2}}, 2\omega, \theta\right) \quad (1.31)$$

5. นำ $h(x, y)$ ไปผ่านค่าเทรสโฮลเพื่อกำหนดขอบภาพ

1.2.4 การหาขอบภาพโดยวิธี Relaxation

การหาขอบภาพโดยวิธี Relaxation [19-20] จะใช้ข้อมูลของความเป็นขอบภาพในย่านใกล้เคียงมาปรับระดับความเชื่อมั่นของขอบภาพที่กำลังพิจารณา วิธีที่ใช้ในการหาขอบภาพในที่นี้คือ วิธีการพิจารณาขอบภาพที่เป็นรอยแยก (Crack Edge) ที่เกิดขึ้นระหว่างจุดเชื่อมต่อของค่าความเข้มที่มีการเปลี่ยนแปลง ถ้ากำหนดให้ $g(x, y)$ และ $g(x', y')$ เป็นจุดภาพในย่านใกล้เคียงกัน ดังนั้นระดับความเข้ม (S) ของขอบสามารถกำหนดได้ดังนี้ S จะวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$s = |g(x, y) - g(x', y')| \quad (1.32)$$

การหาขอบภาพที่เป็นรอยแยกสามารถทำได้โดย การหาระดับความเข้มของขอบภาพ $S(e)$ แล้วนำมาหาค่าเทรสโพลในตัวอย่างข้างล่างแสดงการหาระดับความเข้มของขอบภาพแบบรอยแยก

$$\begin{bmatrix} 4 & & \\ 1 & 3 & 3 \\ 0 & & \end{bmatrix} \xrightarrow{s} \begin{bmatrix} 1 & & \\ 2 & & 0 \\ 3 & & \end{bmatrix}$$

กำหนดให้ค่าเทรสโพลเท่ากับ 1 ดังนั้นระดับความเข้มของขอบภาพที่เป็นรอยแยก $S(e)$ จะเท่ากับ 2 และ 3 ตามลำดับ

เมื่อหาขอบภาพแบบรอยแยกได้แล้วเราจะใช้ขั้นตอนของ Relaxation เพื่อปรับปรุงผลของขอบภาพ โดยการปรับระดับค่าความเป็นขอบภาพ ซึ่งจะใช้ข้อมูลของขอบภาพในบริเวณใกล้เคียงเป็นตัวพิจารณาถึงระดับความเชื่อมั่นว่าเป็นขอบภาพหรือไม่ ระดับความเชื่อมั่นนี้ได้จากค่าความเป็นขอบภาพของจุดภาพในบริเวณใกล้เคียง และระดับความเชื่อมั่นนี้จะมีการปรับตัวมันเองอย่างอัตโนมัติ คือถ้าจุดภาพที่กำลังพิจารณามีความเป็นขอบภาพต่ำ แต่ขอบภาพในบริเวณใกล้เคียงมีค่าระดับความเชื่อมั่นของความเป็นขอบภาพสูงวิธี Relaxation สามารถปรับเพิ่มค่าความเป็นขอบภาพของจุดนี้ได้เอง ในทางตรงกันข้ามถ้าจุดใดไม่ใช่ขอบภาพที่แท้จริง วิธีนี้ก็ลดค่าความเชื่อมั่นของความเป็นขอบภาพลงมาได้ โดยอาศัยระดับความเชื่อมั่นของขอบภาพในบริเวณใกล้เคียงเช่นกัน ดังนั้นระดับความเชื่อมั่นของความเป็นขอบภาพในที่นี้จะมีค่าเพิ่มขึ้น, ลดลง, หรือ อาจจะคงตัวก็ได้

ระดับความเชื่อมั่นของความเป็นขอบภาพ $C(e)$ ได้จากการคำนวณขอบภาพแบบรอยแยกดังที่ได้อธิบายมาแล้วข้างต้น การกำหนดค่า $C(e)$ จะต้องใช้ข้อมูลของขอบภาพในย่านใกล้เคียง 3 จุด ด้วยกัน และข้อมูลของขอบภาพที่นำมาใช้จะต้องทำให้มีค่าอยู่ในช่วง $[0,1]$ สมมุติขอบภาพทั้ง 3 จุดมีระดับความเชื่อมั่นของขอบภาพเป็น $a, b,$ และ c ตามลำดับ โดย $a \geq b \geq c$ และระดับความเชื่อมั่นแต่ละระดับคำนวณได้จากสมการที่ (1.32) ในรูปที่ 1.9. แสดงตำแหน่งของระดับความเชื่อมั่น $a, b,$ และ c

$$\begin{array}{c} a \\ \hline b \quad \frac{e}{c} \end{array}$$

กำหนดให้ $m = \text{Max}(a, b, c, q)$ ที่ q เป็นขอบเขตล่างของระดับความเชื่อมั่นของขอบภาพ (โดยที่ $q > 0$) ระดับความเชื่อมั่นของความเป็นขอบภาพที่ใช้ในการปรับปรุงขอบภาพในที่นี้มีด้วยกัน 4 ระดับคือ

-ระดับที่ 1. (ไม่ใช่ขอบภาพ)	$\text{conf}(0) = (m-a)(m-b)(m-c)$
-ระดับที่ 2. (มีขอบภาพอยู่ 1 จุด)	$\text{conf}(1) = a(m-b)(m-c)$
-ระดับที่ 3. (มีขอบภาพอยู่ 2 จุด)	$\text{conf}(2) = ab(m-c)$
-ระดับที่ 4. (มีขอบภาพอยู่ 3 จุด)	$\text{conf}(3) = abc$

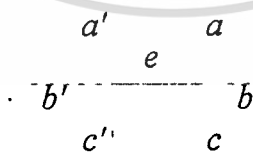
การเลือกว่าขอบภาพที่กำลังพิจารณา อยู่ในระดับความเชื่อมั่นชนิดใดสามารถดำเนินการได้โดยการเลือกค่าระดับความเชื่อมั่นที่มีค่ามากที่สุดจากทั้ง 4 ชนิด ($e = \text{Max}(\text{conf}(0), \text{conf}(1), \text{conf}(2), \text{conf}(3))$)

การกำหนดค่าของตัวแปร q เป็นการบังคับชนิดของความเชื่อมั่นที่เป็นศูนย์ ซึ่งจะเกิดขึ้นในกรณีที่ค่าความเชื่อมั่นของความเป็นขอบภาพ a, b , และ c มีค่าต่ำมากๆ เช่น $a = 0.05, b = 0.03, c = 0.01$, และ $q = 0.1$ $\text{conf}(0)$ จะมีค่าสูงที่สุด ดังนั้นขอบภาพที่กำลังพิจารณาอยู่จะมีระดับ ความเชื่อมั่นเป็นระดับที่ 1. ส่วนกรณีอื่นๆ เช่น $a = 1, b = 0.05, c = 0.01$ และ $q = 0.1$ $\text{conf}(1)$ จะมีค่าสูงสุด ดังนั้นขอบภาพนี้จะมีระดับความเชื่อมั่นเป็นระดับที่ 2. เป็นต้น

ชนิดของความเป็นขอบภาพแบ่งเป็น 3 ชนิดด้วยกันคือ

1. D คือความเป็นขอบภาพมีค่าลดลง
2. I ความเป็นขอบภาพมีค่าเพิ่มขึ้น
3. U ความเป็นขอบภาพคงที่

การกำหนดชนิดของความเป็นขอบภาพ จะต้องใช้ระดับความเชื่อมั่นของขอบภาพสองจุดที่อยู่ในบริเวณใกล้เคียงกัน ดังแสดงในรูปที่ 1.10



รูปที่ 1.10 แสดงตำแหน่งของขอบภาพที่จะนำมาพิจารณาร่วมเพื่อใช้ในการกำหนดชนิดของความเป็นขอบภาพ

จากรูปที่ 1.10 จะต้องหาชนิดของระดับความเชื่อมั่น e และ e' โดยใช้ข้อมูลของขอบภาพในบริเวณใกล้เคียง a, b, c และ a', b', c' ตามลำดับ เมื่อกำหนดชนิดของระดับความเชื่อมั่นของ e และ e' ได้แล้วก็นำมาหาชนิดของความเป็นขอบภาพ โดยใช้ข้อมูลในตารางที่ 1.1

ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 1.1 การกำหนดชนิดของขอบภาพ

	D	I	U
e/e'	1/1	2/2	1/2
	1/3	2/3	3/3
	1/4	2/4	3/4
			4/4

ในตารางที่ 1.1 ความเป็นขอบภาพเป็นชนิด D ก็ต่อเมื่อชนิดของระดับความเชื่อมั่นของขอบภาพ $e = 1$ และ $e' = 1$ หรือ $e = 1$ และ $e' = 3$ หรือ $e = 1$ และ $e' = 4$ สำหรับความเป็นขอบภาพชนิด I และ U ก็กำหนดได้ในลักษณะเดียวกัน

การกำหนดชนิดของความเป็นขอบภาพก็เพื่อใช้ในการปรับปรุงค่า $C(e)$ ให้มีค่าเป็น 0 หรือ 1 (0 : ไม่ใช่ขอบภาพ, 1 : เป็นขอบภาพ) ซึ่งสามารถดำเนินการได้ดังนี้

$$C_{k+1}(e) = \begin{cases} \text{Min}(1, C_k(e) + d), & \text{ถ้า } e \text{ เป็นชนิด I} \\ C_k(e), & \text{ถ้า } e \text{ เป็นชนิด U} \\ \text{Max}(0, C_k(e) - d), & \text{ถ้า } e \text{ เป็นชนิด D} \end{cases} \quad (1.33)$$

ตัวแปร d ใช้ในการเพิ่มหรือลดระดับความเชื่อมั่น $C_k(e)$ โดยที่ $d > 0$ และ k เป็นดัชนีของจำนวนครั้งที่ใช้ในการปรับปรุง ในสมการที่ (1.33) เป็นการปรับปรุงค่าระดับความเชื่อมั่นของความเป็นขอบภาพโดยอัตโนมัติ คือถ้ามีขอบภาพที่มีค่าระดับความเชื่อมั่นของความเป็นขอบภาพสูงจะส่งผลไปถึงขอบภาพที่มีค่าระดับความเชื่อมั่นต่ำ โดยจะดึงค่าระดับความเชื่อมั่นที่ต่ำให้สูงขึ้นจนเป็นขอบภาพ และในทางตรงกันข้ามเมื่อมีจุดที่มีระดับเชื่อมั่นของความเป็นขอบภาพต่ำสมการที่ (1.33) จะลดค่าอื่นลงมาด้วยเพื่อกำจัดจุดที่ไม่ใช่ขอบภาพที่แท้จริง อัลกอริทึมที่ 1-2. เป็นขั้นตอนการหาขอบภาพโดยวิธี Relaxation

อัลกอริทึมที่ 1-2. แสดงขั้นตอนการหาขอบภาพโดยใช้วิธี Relaxation

กำหนดให้ $L1, L2, L3, L4, L1', L2', L3', L4'$ เป็นระดับความเชื่อมั่นของความเป็นขอบภาพ ทั้ง 4 ระดับ ของขอบภาพ e และ e'

void Edge_Relaxation()

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 { ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

k = 0;

$$c_k(e) = \frac{S(e)}{\text{Max}_{\forall e}(S(e))};$$

q = 0.1; /* กำหนดค่า q */

d = 0.1; /* กำหนดค่า d */

while ((c_k(e) != 1) || (c_k(e) != 0))

{

k = k + 1;

a = abs(e-a);

b = abs(e-b);

c = abs(e-c);

a' = abs(e'-a');

b' = abs(e'-b');

c' = abs(e'-c');

m = Max(a, b, c, q);

m' = Max(a', b', c', q);

L1 = (m-a)*(m-b)*(m-c);

L2 = a*(m-b)*(m-c);

L3 = a*b*(m-c);

L4 = a*b*c;

L1' = (m'-a')*(m'-b')*(m'-c');

L2' = a*(m'-b')*(m'-c');

L3' = a*b*(m'-c');

L4' = a*b*c';

e = Max(L1, L2, L3, L4);

e' = Max(L1', L2', L3', L4');

if ((e == 1) && ((e' == 1) || (e' == 3) || (e' == 4)))

/* e มีความเป็นขอบภาพเป็นชนิด D */

c_k(e) = Max(0, - d);

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if ((e == 2) && ((e' == 2) || (e' == 3) || (e' == 4)))
    /* e มีความเป็นขอบภาพเป็นชนิด I */
     $c_k(e) = \text{Min}(1, + d);$ 
if (((e == 1) || (e == 3) || (e == 4)) && ((e' == 2) || (e' == 3) || (e' == 4)))
    /* e มีความเป็นขอบภาพเป็นชนิด U */
     $c_k(e) = c_k(e);$ 
} /* end while loop */
} /* end procedure edge relaxation */

```

ในอัลกอริทึมที่ 1-2. จะเกิดกรณีวนลูปแบบไม่รู้จบขึ้น ถ้า e มีความเป็นขอบภาพเป็นชนิด U ในกรณีนี้สามารถแก้ปัญหาได้โดยการกำหนดจำนวนลูปที่แน่นอนให้กับดัชนี k สำหรับตัวแปร d และ q กำหนดได้จากการทดลอง โดยตัวแปรทั้งสองตัวนี้จะขึ้นอยู่กับการแจกแจงของข้อมูลภาพและจะมีผลต่อจำนวนลูปของ k ด้วย คือถ้ากำหนดค่า d และ q ที่เหมาะสมได้แล้วจะทำให้จำนวนรอบในการวนลูปน้อยลงและขอบภาพที่ได้ก็มีความคมชัด ในทางตรงกันข้ามถ้ากำหนดค่า d และ q ไม่เหมาะสมกับภาพนั้นๆ แล้วจะทำให้ขอบภาพที่ได้ไม่สมบูรณ์และจะมีสัญญาณรบกวนปนออกมาด้วย

1.3 การวัดประสิทธิภาพของขอบภาพ

สำหรับการวัดประสิทธิภาพของตัวดำเนินการหาขอบภาพ [21] ในแต่ละวิธีสามารถกำหนดได้ดังนี้

$$P_e = \frac{n_1}{n_0} \quad (1.34)$$

ที่ n_0 เป็นจำนวนจุดของขอบภาพที่แท้จริง n_1 เป็นจำนวนจุดของขอบภาพที่ได้จากการหาขอบภาพของแต่ละวิธี สำหรับค่า P_e นี้จะแสดงถึงประสิทธิภาพของตัวดำเนินการหาขอบภาพนั้นคือถ้า P_e มีค่าเข้าใกล้ 1 แสดงว่าวิธีนั้นมีประสิทธิภาพสูงแต่ถ้า P_e น้อยกว่า 1 มากๆ แสดงว่าจุดภาพที่หาได้น้อยกว่าจุดภาพของขอบภาพที่แท้จริง และในทางตรงกันข้ามคือ P_e มากกว่า 1 แสดงว่าจำนวนจุดภาพของขอบภาพที่หาได้มากกว่าขอบภาพที่แท้จริง อีกวิธีหนึ่งที่ใช้วัดประสิทธิภาพความถูกต้องของการหาขอบภาพในแต่ละวิธีมีดังนี้

$$P = \frac{1}{\text{Max}(N_I, N_D)} \sum_{i=1}^{N_D} \frac{1}{1 + \alpha d_i^2} \quad (1.35)$$

ที่ d_i เป็นระยะทางระหว่างขอบภาพที่หาได้ในแต่ละวิธีกับขอบภาพที่แท้จริง, α เป็นค่าคงที่ซึ่งในที่นี้มีค่า α ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เท่ากับ 1, และ N_I และ N_D เป็นจำนวนจุดของขอบภาพที่แท้จริงกับ ขอบภาพที่หาได้ในแต่ละวิธีตามลำดับ ค่า P แสดงถึงความถูกต้องของตำแหน่งขอบภาพ, นั่นคือถ้า P เท่ากับ 1 แสดงว่าขอบภาพที่ได้มีความถูกต้องของตำแหน่งของขอบภาพสูงและถ้า P น้อยกว่า 1 มากๆ ความถูกต้องของตำแหน่งขอบภาพก็จะยิ่งลดลง

1.4 เนื้อภาพ

เนื้อภาพ (Texture) ในที่นี้หมายถึง คุณสมบัติที่ใช้แทนพื้นผิวของวัตถุที่ปรากฏอยู่ในภาพ ซึ่งคุณสมบัติอันนี้จะอยู่ในรูปของความสัมพันธ์กันขององค์ประกอบพื้นฐานของเนื้อภาพ(Texture Primitives หรือบางครั้งอาจจะเรียกสั้นๆ ว่า Texels ซึ่งมาจากคำว่า Texture Elements) และองค์ประกอบที่เล็กที่สุดของเนื้อภาพก็คือจุดภาพ ดังนั้นคุณสมบัติที่กล่าวมานี้จะหมายถึงความสัมพันธ์ของกลุ่มจุดภาพที่ใช้แสดงถึงลักษณะพื้นผิวของวัตถุซึ่งก็คือตัวแปรต่างๆ ที่ใช้อธิบายถึงโครงสร้างของพื้นผิววัตถุนั้นเอง สำหรับขนาดของกลุ่มจุดภาพที่เป็นองค์ประกอบพื้นฐานของเนื้อภาพจะขึ้นอยู่กับการใช้งาน [22] เช่น ถ้านำไปใช้ในการหาขอบภาพกลุ่มของจุดภาพอาจอยู่ในกรอบขนาด 3×3 [23] แต่ถ้าต้องการวิเคราะห์เนื้อภาพเพื่อทำการแบ่งแยกวัตถุภายในภาพ หรือเพื่อนำไปใช้ในการรู้จักรูปแบบของวัตถุในภาพโดยคอมพิวเตอร์ (Pattern Recognition) ขนาดกลุ่มของจุดภาพก็จะใหญ่ขึ้น เป็นต้น

ขนาดที่แท้จริงของกลุ่มจุดภาพที่เป็นองค์ประกอบพื้นฐานของเนื้อภาพ จะแสดงถึงความเป็นเนื้อเดียวกันของข้อมูลภายในกลุ่ม โดยข้อมูลกลุ่มนี้อาจจะเป็นส่วนหนึ่งของวัตถุหรืออาจจะเป็นวัตถุชิ้นหนึ่งที่ปรากฏอยู่ในภาพซึ่งมีคุณสมบัติคล้ายๆ กัน การกำหนดขนาดที่แท้จริงของกลุ่มจุดภาพที่เป็นองค์ประกอบพื้นฐานของเนื้อภาพนั้นในทางปฏิบัติทำได้ยากมาก เนื่องจากจุดภาพกลุ่มนี้จะต้องมีคุณสมบัติที่เหมือนกัน (เช่น การแจกแจงของข้อมูลภายในกลุ่มจะต้องเป็นไปในทิศทางเดียวกัน) แต่เราไม่สามารถหาตัวแปรที่จะมาตัดสินใจถึงความเป็นเนื้อเดียวกันของข้อมูลชุดนี้ได้เหมาะสม ซึ่งในทางอุดมคติแล้วเนื้อภาพจะมีลักษณะเป็นเชิงสุ่ม แต่จากข้อเท็จจริงเราพบว่าเนื้อภาพจะก้ำกึ่งระหว่างแบบจำลองที่กำหนดขึ้นมากับส่วนที่เป็นเชิงสุ่ม [24] สำหรับแบบจำลองที่สร้างขึ้นมาจะใช้ในการหาความสัมพันธ์ระหว่างแบบจำลองกับเนื้อภาพ เพื่อใช้ในการหาค่าตัวแปรที่จะนำมาใช้อธิบายคุณสมบัติของเนื้อภาพเช่นเดียวกับการวิเคราะห์เนื้อภาพในทางอุดมคติ และเทคนิคต่างๆ ที่ใช้ในการสร้างแบบจำลองของเนื้อภาพแต่ละวิธีจะขึ้นอยู่กับวัตถุประสงค์ที่จะนำไปใช้งาน [25] ซึ่งส่วนใหญ่จะนำไปใช้ในการทำให้คอมพิวเตอร์รู้จักรูปแบบต่างๆ ของวัตถุที่ปรากฏอยู่ในภาพ ในที่นี้เราจะเสนอเฉพาะกรณีการวิเคราะห์เนื้อภาพที่เป็นเชิงสุ่มซึ่งส่วนใหญ่จะอาศัยวิธีการทางสถิติศาสตร์ในการกำหนดตัวแปรต่างๆ และวิธีที่ใช้ในการวิเคราะห์เนื้อภาพที่จะกล่าวถึงในที่นี้ประกอบด้วย Autocorrelation Function, และ Cooccurrence Matrices

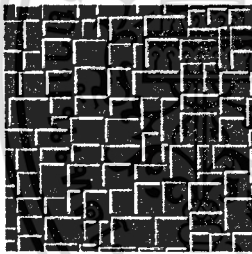
1.4.1. Autocorrelation Function

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

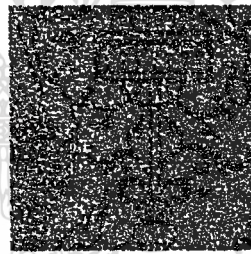
เมื่อเราพิจารณาความสัมพันธ์ของเนื้อหาเกี่ยวกับขนาดของกลุ่มจุดภาพที่เป็นองค์ประกอบพื้นฐานของเนื้อหาจะได้ว่าถ้าขนาดของกลุ่มจุดภาพมีขนาดใหญ่แสดงว่าเนื้อหาที่มีความหยาบ (ดังรูปที่ 1.11ก) และถ้าขนาดของกลุ่มจุดภาพที่เป็นองค์ประกอบพื้นฐานของเนื้อหาที่มีขนาดเล็กกว่าย่อมหมายถึงเนื้อหาที่มีความละเอียดกว่า (ดังรูปที่ 1.11ข) สิ่งที่ใช้อธิบายถึงลักษณะคร่าวๆ โครงที่เป็นขนาดของกลุ่มจุดภาพที่เป็นองค์ประกอบพื้นฐานของเนื้อหาคือ Autocorrelation Function นั่นคือถ้ากลุ่มจุดภาพขนาดใหญ่ ค่าของฟังก์ชัน Autocorrelation จะค่อยๆ ลดลงอย่างช้าๆ เมื่อทำการเลื่อนตำแหน่งของวินโดว์ (ซึ่งมีการดำเนินการคล้ายกับการคอนโวลูชันจะแตกต่างกันเพียงแค่วินโดว์จะไม่ซ้อนทับกัน) และในกรณีที่กลุ่มจุดภาพมีขนาดเล็กค่า Autocorrelation ก็จะลดลงอย่างรวดเร็ว ในสมการที่ (1.36) จะเป็นฟังก์ชัน Autocorrelation ที่ตำแหน่ง p, q

$$C_{gg}(p, q) = \frac{MN}{(M-p)(M-q)} \frac{\sum_{i=1}^{M-p} \sum_{j=1}^{N-q} g(i, j)g(i+p, j+q)}{\sum_{i=1}^M \sum_{j=1}^N g^2(i, j)} \quad (1.36)$$

โดยที่ $g(i, j)$ เป็นจุดภาพที่ตำแหน่ง i, j ซึ่งอยู่ภายในวินโดว์ขนาด $M \times N$



(ก)



(ข)

รูปที่ 1.11 แสดงลักษณะของเนื้อหา (ก) เนื้อภาพหยาบ (ข) เนื้อภาพละเอียด

1.4.2. Cooccurrence Matrices

การอธิบายเนื้อหาโดยวิธีนี้จะอาศัยการเกิดขึ้นซ้ำๆ กันของลักษณะรูปร่างของระดับความเข้มสีเทาในเนื้อหา โดยลักษณะรูปร่างของระดับความเข้มสีเทานี้จะมีการเปลี่ยนแปลงอย่างรวดเร็วมากเมื่อเทียบกับระยะทางในเนื้อหาที่มีความละเอียด ส่วนเนื้อหาที่มีความหยาบมันจะมีการเปลี่ยนแปลงอย่างช้าๆ [26] การเกิดขึ้นร่วมกันของระดับความเข้มสีเทาภายในวินโดว์ขนาด $M \times N$ สามารถเขียนให้อยู่ในรูปของความถี่สัมพันธ์ $P_{\theta,d}(i, j)$ ที่แสดงถึงการเกิดขึ้นร่วมกันของค่าความเข้มระดับสีเทา i และ j และอยู่ในย่านเดียวกันโดยทั้งสองจุดอยู่ห่างกันเป็นระยะทาง d ตามทิศทาง θ องศา สำหรับค่า $P_{\theta,d}(i, j)$ สามารถเขียนให้อยู่ในรูปของเมตริกซ์ที่เรียกว่า Cooccurrence เมตริกซ์ ซึ่งเมตริกซ์นี้จะเป็นเมตริกซ์สมมาตร ในสมการที่ (1.37) แสดงการคำนวณค่าความถี่สัมพันธ์ $P_{\theta,d}(i, j)$ ที่ทิศทาง θ จะแบ่งออกเป็นช่วงๆ โดยแต่ละช่วงจะมีมุมต่างกัน 45 องศา

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{aligned}
P_{0^\circ,d}(i,j) &= \#\{(k,l),(m,n)\} / \\
&\quad k-m=0, |l-n|=d, \\
&\quad I(k,l)=i, I(m,n)=j\} \\
P_{45^\circ,d}(i,j) &= \#\{(k,l),(m,n)\} / \\
&\quad k-m=d, l-n=-d, \\
&\quad k-m=-d, l-n=d, \\
&\quad I(k,l)=i, I(m,n)=j\} \\
P_{90^\circ,d}(i,j) &= \#\{(k,l),(m,n)\} / \\
&\quad |k-m|=d, l-n=0, \\
&\quad I(k,l)=i, I(m,n)=j\} \\
P_{135^\circ,d}(i,j) &= \#\{(k,l),(m,n)\} / \\
&\quad k-m=d, l-n=d, \\
&\quad k-m=-d, l-n=-d, \\
&\quad I(k,l)=i, I(m,n)=j\}
\end{aligned} \tag{1.37}$$

โดยที่ # เป็นจำนวนครั้งที่ค่าระดับความเข้มสีเทา i และ j เกิดขึ้นร่วมกันภายในวินโดว์ขนาด $M \times N$ ส่วน k, l, m , และ n เป็นดัชนีที่ชี้ไปยังตำแหน่งของจุดภาพ

จากสมการที่ (1.37) สามารถนำมาเขียนเป็นอัลกอริทึมของการกำหนด Cooccurrence เมตริกซ์ได้ดังนี้

อัลกอริทึมที่ 1-3. เป็นขั้นตอนในการกำหนด Cooccurrence เมตริกซ์

กำหนดให้ $X[M,N]$	เป็นเมตริกซ์ของข้อมูลภาพที่มีขนาด $M \times N$
P_H	เป็นเมตริกซ์ของ Cooccurrence ที่มีทิศทางการแนวอนที่ $\theta = 0^\circ$
P_V	เป็นเมตริกซ์ของ Cooccurrence ที่มีทิศทางการแนวตั้งที่ $\theta = 90^\circ$
P_{SD}	เป็นเมตริกซ์ของ Cooccurrence ที่มีทิศทางการแนวเฉียงที่ $\theta = 45^\circ$
P_{LD}	เป็นเมตริกซ์ของ Cooccurrence ที่มีทิศทางการแนวเฉียงที่ $\theta = 135^\circ$
d	เป็นระยะทางระหว่างค่าระดับความเข้มสีเทาที่ i และที่ j
k, l, m, n	เป็นดัชนีที่ชี้ไปยังตำแหน่งของจุดภาพในเมตริกซ์
g_{Min}	เป็นค่าต่ำสุดของจุดภาพในเมตริกซ์ X
g_{Max}	เป็นค่าสูงสุดของจุดภาพในเมตริกซ์ X

void Cooccurrence_Matrix(MATRIX X, int d) การศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

{ ไม่ว่าการณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* ตั้งค่าเริ่มต้นของสมาชิกเมตริกซ์  $P_H$ ,  $P_V$ ,  $P_{SD}$ , และ  $P_{LD}$  ทุกตัวให้มีค่าเป็นศูนย์ */
for(int m = gMin; m < gMax; m++)
    for(int n = gMin; n < gMax; n++)
    {
         $P_H[m][n] = 0;$ 
         $P_V[m][n] = 0;$ 
         $P_{SD}[m][n] = 0;$ 
         $P_{LS}[m][n] = 0;$ 
    }

/* กำหนดค่า Cooccurrence เมตริกซ์ */
for(int k = gMin; k < gMax; k++)
    for(int l = gMin; l < gMax; l++)
        for(int m = gMin; m < gMax; m++)
            for(int n = gMin; n < gMax; n++)
            {
                /*  $P_{0,d}(i, j)$  */
                if (k-m == 0 && (abs(l-n) == d))
                     $P_H[X[m][n]][X[k][l]] += 1;$ 

                /*  $P_{45,d}(i, j)$  */
                if ((k-m == d) && (l-n == -d)) || ((k-m == -d) && (l-n == d))
                     $P_{SD}[X[m][n]][X[k][l]] += 1;$ 

                /*  $P_{90,d}(i, j)$  */
                if (abs(k-m) == d && (l-n == 0))
                     $P_V[X[m][n]][X[k][l]] += 1;$ 

                /*  $P_{135,d}(i, j)$  */
                if ((k-m == d) && (l-n == d)) || ((k-m == -d) && (l-n == -d))
                     $P_{LD}[X[m][n]][X[k][l]] += 1;$ 
            }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
} /* จบอัลกอริทึม Cooccurrence_Matrix */
```

จากอัลกอริทึมที่ 1-3. สามารถนำมาสร้าง Cooccurrence เมตริกซ์ดังตัวอย่างที่ 1.2

ตัวอย่างที่ 1.2 แสดงการคำนวณ Cooccurrence เมตริกซ์ที่มีระยะทาง $d=1$ ในรูปที่ 1.12 แสดงข้อมูลภาพที่มีค่าระดับความเข้มสีเทา 4 ระดับ ($i, j = 0, 1, 2,$ และ 3) และภาพมีขนาด 4×4 จุด

$$\begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 2 & 2 & 2 \\ 2 & 2 & 3 & 3 \end{bmatrix}$$

รูปที่ 1.12 ภาพระดับสีเทามีค่าความเข้ม 4 ระดับ

จากข้อมูลภาพในรูปที่ 1.12 สามารถกำหนดตำแหน่งของค่าระดับความเข้มสีเทาที่เป็นรูปแบบต่างๆ ไปของ Cooccurrence เมตริกซ์ดังแสดงในรูปที่ 1.13

		ระดับความเข้มสีเทา			
		0	1	2	3
ระดับความเข้มสีเทา	0	#(0,0)	#(0,1)	#(0,2)	#(0,3)
	1	#(1,0)	#(1,1)	#(1,2)	#(1,3)
	2	#(2,0)	#(2,1)	#(2,2)	#(2,3)
	3	#(3,0)	#(3,1)	#(3,2)	#(3,3)

รูปที่ 1.13 รูปแบบทั่วไปของ Cooccurrence เมตริกซ์

ในรูปที่ 1.14 แสดงค่า Cooccurrence เมตริกซ์ของข้อมูลภาพในรูปที่ 1.12 โดยใช้อัลกอริทึมที่ 1-3. ในการคำนวณ

$$P_H = \begin{bmatrix} 4 & 2 & 1 & 0 \\ 2 & 4 & 0 & 0 \\ 1 & 0 & 6 & 1 \\ 0 & 0 & 1 & 2 \end{bmatrix}$$

$$P_{SD} = \begin{bmatrix} 4 & 1 & 0 & 0 \\ 1 & 2 & 2 & 0 \\ 0 & 2 & 4 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$P_V = \begin{bmatrix} 6 & 0 & 2 & 0 \\ 0 & 4 & 2 & 0 \\ 2 & 2 & 2 & 2 \\ 0 & 0 & 2 & 0 \end{bmatrix}$$

(ก) $\theta = 90^\circ$

$$P_{LD} = \begin{bmatrix} 2 & 1 & 3 & 0 \\ 1 & 2 & 1 & 0 \\ 3 & 1 & 0 & 2 \\ 0 & 0 & 2 & 0 \end{bmatrix}$$

(ง) $\theta = 135^\circ$

รูปที่ 1.14 แสดง Cooccurrence เมตริกซ์ที่ระยะทาง $d=1$ โดยมีทิศทางต่างๆ กัน

จากรูปที่ 1.14 $P_{\theta=0^\circ, d=1}(0,0)$ จะแทนจำนวนครั้งที่ระดับค่าความเข้มสีเทา $i=0$, และ $j=0$ เกิดขึ้นร่วมกันทั้งภาพซึ่งมีทั้งหมด 4 ครั้ง และสมาชิกตัวอื่นๆ ในเมตริกซ์ P_H หรือ P_V , P_{LD} , และ P_{SD} ก็กำหนดได้ในลักษณะเดียวกันตามเงื่อนไขในสมการที่ (1.37)

จาก Cooccurrence เมตริกซ์หรือฮิสโตแกรมแบบสองมิติทำให้สามารถกำหนดฟังก์ชันความหนาแน่นของความน่าจะเป็นแบบมีเงื่อนไขร่วมกันในอันดับที่สอง [21, 24] ได้ดังนี้

$$P_{\theta, d}(i, j) = \frac{P_{\theta, d}(i, j)}{\sum_{v_i} \sum_{v_j} P_{\theta, d}(i, j)} \quad (1.38)$$

จากรูปที่ 1.14ก เราสามารถคำนวณค่าความน่าจะเป็นของ Cooccurrence เมตริกซ์ได้ดังรูปที่ 1.15

$$P_H = \begin{bmatrix} \frac{4}{24} & \frac{2}{24} & \frac{1}{24} & 0 \\ \frac{2}{24} & \frac{4}{24} & 0 & 0 \\ \frac{1}{24} & 0 & \frac{6}{24} & \frac{1}{24} \\ 0 & 0 & \frac{1}{24} & \frac{2}{24} \end{bmatrix}$$

รูปที่ 1.15 ค่าความน่าจะเป็นของ Cooccurrence เมตริกซ์ตามแนวอนและระยะทาง $d=1$

จากฟังก์ชันความหนาแน่นของความน่าจะเป็นแบบมีเงื่อนไขร่วมกันในอันดับที่สองของ Cooccurrence เมตริกซ์ ทำให้สามารถกำหนดตัวแปรทางสถิติศาสตร์เพื่อนำไปใช้ในการวิเคราะห์เนื้อหาของได้ 14 ตัวแปรด้วยกัน [26] แต่ที่นำไปใช้ในการจำแนกวัตถุที่อยู่ในภาพทั่วไปส่วนใหญ่จะใช้ตัวแปรเหล่านี้เพียงไม่กี่ตัวแปรเท่านั้น เช่น การใช้ลักษณะของเนื้อหาในการจำแนกวัตถุในภาพถ่ายดาวเทียมของไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Haralick จะใช้ตัวแปรเหล่านี้เพียง 5 ตัว [26], การจำแนกชนิดของเมฆโดยใช้นิวรอลเน็ตเวิร์ค [27] จะใช้ตัวแปรเหล่านี้เพียง 9 ตัว เป็นต้น การเลือกใช้ตัวแปรแต่ละตัวต้องพิจารณาถึงคุณสมบัติเฉพาะตัวของมัน สำหรับตัวแปรต่างๆ ทั้ง 14 ตัว [26] สามารถกำหนดได้ดังนี้

1. ค่าโมเมนต์เชิงมุมอันดับที่สอง (f_1)

$$f_1 = \sum_i \sum_j (p_{\theta,d}(i, j))^2 \quad (1.39)$$

2. ค่าคอนทราสต์ (Contrast : f_2)

$$f_2 = \sum_{n=0}^{Ng-1} n^2 \sum_{\substack{i=1 \\ |i-j|=n}}^{Ng} \sum_{j=1}^{Ng} p_{\theta,d}(i, j) \quad (1.40)$$

ที่ Ng เป็นจำนวนระดับความเข้มสีเทา

3. สหสัมพันธ์ (f_3)

$$f_3 = \sum_{i=1}^{Ng} \sum_{j=1}^{Ng} \frac{(ij)p_{\theta,d}(i, j) - \mu_x \mu_y}{\sigma_x \sigma_y} \quad (1.41)$$

$$\text{ที่ } \mu_x = \sum_{i=0}^{Ng-1} i \sum_{j=0}^{Ng-1} p_{\theta,d}(i, j) \quad (1.42)$$

$$\mu_y = \sum_{j=0}^{Ng-1} j \sum_{i=0}^{Ng-1} p_{\theta,d}(i, j) \quad (1.43)$$

$$\sigma_x^2 = \sum_{i=0}^{Ng-1} (i - \mu_x)^2 \sum_{j=0}^{Ng-1} p_{\theta,d}(i, j) \quad (1.44)$$

$$\sigma_y^2 = \sum_{j=0}^{Ng-1} (j - \mu_y)^2 \sum_{i=0}^{Ng-1} p_{\theta,d}(i, j) \quad (1.45)$$

4. ผลรวมกำลังสองของความแปรปรวน (f_4)

$$f_4 = \sum_{\forall i} \sum_{\forall j} (i - \mu)^2 p_{\theta,d}(i, j) \quad (1.47)$$

$$\text{ที่ } \mu = \sum i \sum j p_{\theta,d}(i, j) \quad (1.48)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. โมเมนต์ความแตกต่างแบบผกผัน (f_5)

$$f_5 = \sum_{\forall i} \sum_{\forall j} \frac{1}{1+(i-j)^2} p_{\theta,d}(i,j) \quad (1.49)$$

6. ผลรวมของค่าเฉลี่ย (f_6)

$$f_6 = \sum_{i=2}^{2Ng} i p_{x+y}(i) \quad (1.50)$$

ที่ $p_{x+y}(k) = \sum_{i=1}^{Ng} \sum_{j=k-i}^{Ng} p_{\theta,d}(i,j); \quad k = 2, 3, 4, \dots, 2Ng$ (1.51)

7. ผลรวมของความแปรปรวน (f_7)

$$f_7 = \sum_{i=2}^{2Ng} (i - f_6)^2 p_{x+y}(i) \quad (1.52)$$

8. ผลรวมของเอนโทรปี (f_8)

$$f_8 = - \sum_{i=2}^{2Ng} p_{x+y}(i) \log(p_{x+y}(i)) \quad (1.53)$$

9. เอนโทรปี (f_9)

$$f_9 = - \sum_{\forall i} \sum_{\forall j} p_{\theta,d}(i,j) \log(p_{\theta,d}(i,j)) \quad (1.54)$$

10. ความแปรปรวนของผลต่าง (f_{10})

$$f_{10} = \sum_{k=0}^{Ng-1} (k - \mu_{x-y})^2 p_{x-y}(k) \quad (1.55)$$

ที่ $p_{x-y}(k) = \sum_{i=1}^{Ng} \sum_{j=|i-k|}^{Ng} p_{\theta,d}(i,j); \quad k = 0, 1, 2, \dots, Ng-1$ (1.56)

$$\mu_{x-y} = \sum_k^{Ng-1} k p_{x-y}(k) \quad (1.57)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11. เอนโทรปีของความแตกต่างของระดับความเข้มสีเทา (f_{11})

$$f_{11} = -\sum_{i=0}^{Ng-1} p_{x-y}(i) \log(p_{x-y}(i)) \quad (1.58)$$

12, 13. การวัดข้อมูลภาพด้วยค่าสหสัมพันธ์

$$f_{12} = \frac{HXY - HXY1}{\text{Max}(HX, HY)} \quad (1.59)$$

$$f_{13} = \sqrt{(1 - e^{-2(HXY2 - HXY)})} \quad (1.60)$$

ที่ $HXY = -\sum_i \sum_j p_{\theta,d}(i, j) \log(p_{\theta,d}(i, j)) \quad (1.61)$

$$HX = -\sum_i p_x(i) \log(p_x(i)) \quad (1.62)$$

$$p_x(i) = \sum_j p_{\theta,d}(i, j) \quad (1.63)$$

$$HY = -\sum_j p_y(j) \log(p_y(j)) \quad (1.64)$$

$$p_y(j) = \sum_i p_{\theta,d}(i, j) \quad (1.65)$$

$$HXY1 = -\sum_i \sum_j p_{\theta,d}(i, j) \log(p_x(i)p_y(j)) \quad (1.66)$$

$$HXY2 = -\sum_i \sum_j p_x(i)p_y(j) \log(p_x(i)p_y(j)) \quad (1.67)$$

14. สัมประสิทธิ์ที่มีค่าสูงสุดของสหสัมพันธ์ (f_{14})

$$f_{14} = \sqrt{\text{ค่าไอเกนของ } Q} \quad (1.68)$$

$$\text{ที่ } Q_{(i,j)} = \sum_k \frac{p_{\theta,d}(i, k)p_{\theta,d}(j, k)}{p_x(i)p_y(j)}$$

จากสมการที่ (1.35) - (1.67) สามารถนำมาเขียนเป็นอัลกอริทึมเพื่อคำนวณค่าตัวแปรทางสถิติ 13 ตัวได้ดังนี้

อัลกอริทึมที่ 1-3. แสดงขั้นตอนการคำนวณค่าตัวแปรทางสถิติ 13 ตัว

/* P เป็น Cooccurrence เมตริกซ์ที่มีระยะทาง d และมุม θ ใดๆ */

{

/* f1 - f13 เป็นตัวแปรทางสถิติ */

float f1 = 0, f2 = 0, f3 = 0, f4 = 0, f5 = 0, f6 = 0, f7 = 0;

float f8 = 0, f9 = 0, f10 = 0, f11 = 0, f12 = 0, f13 = 0;

/* ตัวแปรที่ใช้ในการถ่ายโอนค่า */

float sum_j, sum_x, sum_y, $\mu = 0$, $\mu_x = 0$, $\mu_y = 0$, $\mu_{x-y} = 0$;

float $\sigma_x = 0$, $\sigma_y = 0$, HXY1 = 0, HXY2 = 0, HX = 0, HY = 0, MN = 0, Pr[gMax][gMax];

int n, k;

/* จองหน่วยความจำให้กับเวกเตอร์ P_{x-y} จำนวน gMax ตัว */

float * $P_{x-y} = \text{new float}[gMax]$,

/* จองหน่วยความจำให้กับเวกเตอร์ P_{x+y} จำนวน $2 * gMax$ ตัว */

* $P_{x+y} = \text{new float}[2 * gMax]$,

/* จองหน่วยความจำให้กับเวกเตอร์ P_x จำนวน gMax */

* $P_x = \text{new float}[gMax]$,

/* จองหน่วยความจำให้กับเวกเตอร์ P_y จำนวน gMax */

* $P_y = \text{new float}[gMax]$;

for(int i=gMin; i<=gMax; i++)

$P_{x-y}[i] = 0$; /* เซตค่าเริ่มต้นของเวกเตอร์ P_{x-y} ทุกตัวให้เป็นศูนย์ */

for(i=gMin; i<2*gMax; i++)

$P_{x+y}[i] = 0$; /* เซตค่าเริ่มต้นของเวกเตอร์ P_{x+y} ทุกตัวให้เป็นศูนย์ */

/* หาจำนวนคู่ของจุดภาพ i และ j ที่เกิดขึ้นร่วมกันทั้งหมด */

for(i = gMin; i <= gMax; i++)

for(int j = gMin; j <= gMax; j++)

MN += P[i][j];

for(i = gMin; i <= gMax; i++)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 {
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

sum_j = 0;
sum_x = 0;
sum_y = 0;
for(j = gMin; j <= gMax; j++)
{
    Pr[i][j] = P[i][j] / MN;          /* คำนวณค่าความน่าจะเป็น */
    k = abs(i - j);
    Px-y[k] += Pr[i][j];
    n = i + j;
    Px+y[n] += Pr[i][j];
    sum_j += j * Pr[i][j];
    sum_x += Pr[i][j];
    sum_y += Pr[j][i];

    f1 += Pr[i][j] * Pr[i][j];      /* คำนวณค่าโมเมนต์เชิงมุมอันดับที่สอง */
    /* คำนวณค่าโมเมนต์ความแตกต่างแบบผกผัน */
    f5 += (1.0 / (1 + (i - j) * (i - j))) * Pr[i][j];
    /* คำนวณค่าเอนโทรปี */
    if (Pr[i][j] > 0)
        f9 += -Pr[i][j] * log(Pr[i][j]);
}

μ += i * sum_j;
μx += i * sum_x;
μy += i * sum_y;
Px[i] = sum_x;
Py[i] = sum_y;
}

for(i = gMin; i <= gMax; i++)
    μx+y += i * Px+y[i];

```

```

{
for(j = gMin; j <= gMax; j++)
{
f3 += (i -  $\mu_x$ ) * (j -  $\mu_y$ ) * Pr[i][j];          /* คำนวณค่าสหสัมพันธ์ */
/* คำนวณค่าผลรวมกำลังสองของความแปรปรวน */
f4 += (i -  $\mu$ ) * (i -  $\mu$ ) * Pr[i][j];
if ( (Px[i] > 0) && (Py[j] > 0) )
{
HXY1 += -Pr[i][j] * log(Px[i] * Py[j]);
HXY2 += -Px[i] * Py[j] * log(Px[i] * Py[j]);
}
}
f2 += i * i * Px-y[i];          /* คำนวณค่าคอนแทรสต์ */
f10 += (i -  $\mu_{x-y}$ ) * (i -  $\mu_{x-y}$ ) * Px-y[i];          /* คำนวณค่าความแปรปรวนของผลต่าง */
/* คำนวณค่าเอนโทรปีของความแตกต่างของระดับความเข้มสีเทา */
if (Px-y[i] > 0)
f11 += -Px-y[i] * log(Px-y[i]);
 $\sigma_x$  += (i -  $\mu_x$ ) * (i -  $\mu_x$ ) * Px[i];
 $\sigma_y$  += (i -  $\mu_y$ ) * (i -  $\mu_y$ ) * Py[i];
if (Px[i] > 0)
HX += -Px[i] * log(Px[i]);
if (Py[i] > 0)
HY += -Py[i] * log(Py[i]);
}
if ( ( $\sigma_x$  > 0) && ( $\sigma_y$  > 0) )
f3 = f3 / (sqrt( $\sigma_x$ ) * sqrt( $\sigma_y$ ));          /* คำนวณค่าสหสัมพันธ์ */
/* คำนวณตัววัดข้อมูลด้วยค่าสหสัมพันธ์ */
if (HX >= HY)
f12 = (f9 - HXY1)/HX;
else
f12 = (f9 - HXY1)/HY;
f13 = sqrt(fabs(1.0 - exp(2.0 * (HXY2 - f9))));

```

```

for(i = gMin; i <= (2 * gMax); i++)
{
    f6 += i * Px+y[i];           /* หาผลรวมของค่าเฉลี่ย */
    if (Px+y[i] > 0)
        f8 += -Px+y[i] * log(Px+y[i]);   /* หาค่าผลรวมของเอนโทรปี */
}
for(i = gMin; i <= (2*gMax); i++)
    f7 += (i - f8) * (i - f8) * Px+y[i];   /* หาค่าผลรวมของความแปรปรวน */
} /* สิ้นสุดการทำงานของฟังก์ชัน Statistic_Variables */

```

ในการทดสอบ Cooccurrence เมตริกซ์นั้น จะใช้ข้อมูลที่ได้จำลองและข้อมูลภาพจริง ในกรณีข้อมูลจำลองได้แสดงในตารางที่ 1.2 ซึ่งผลการคำนวณค่าตัวแปรทางสถิติของ Cooccurrence เมตริกซ์ทั้ง 13 ตัวของรูปที่ 1.14ก โดยใช้อัลกอริทึมที่ 1-4

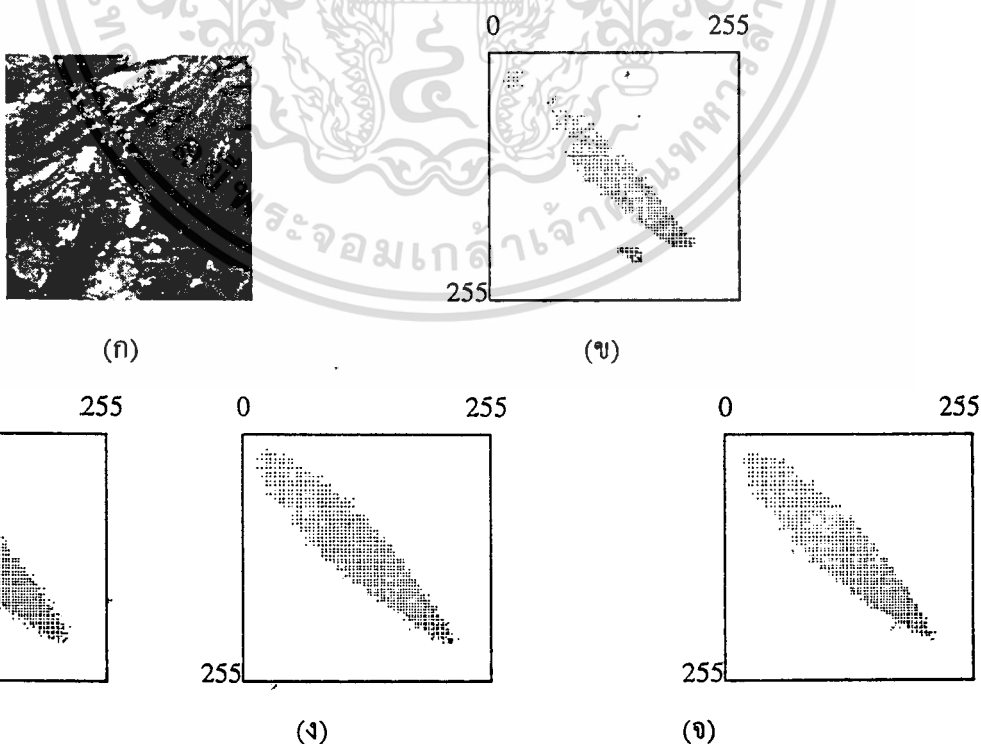
ตารางที่ 1.2 แสดงผลการคำนวณค่าตัวแปรทางสถิติของ Cooccurrence เมตริกซ์ในรูปที่ 1.14ก

ตัวแปร	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	f11	f12	f13
ค่าของ ตัวแปร	0.14	0.58	0.70	2.30	0.80	2.58	4.34	1.70	2.09	0.40	0.82	-0.37	1.74

ส่วนในกรณีของภาพจริงนั้นได้แสดงในตารางที่ 1.3 ซึ่งเป็นผลลัพธ์ของการคำนวณค่าตัวแปรทางสถิติของ Cooccurrence เมตริกซ์ของรูปที่ 1.16ก ซึ่งเป็นภาพถ่ายดาวเทียมแบนด์ที่ 4 ของดาวเทียม NOAA ภาพนี้ถ่ายเมื่อวันที่ 30 มกราคม 2536 ค่าระดับความเข้มสีเทาของภาพนี้อยู่ในช่วง 0-255 ในตารางที่ 1.3 จะแสดงผลของตัวแปรทั้ง 4 ทิศทาง คือ 0, 45, 90 และ 135 องศา ที่ระยะทาง $d = 1$ ในรูปที่ 1.16ข-1.16จ แสดงข้อมูลของ Cooccurrence เมตริกซ์ของรูปที่ 1.16ก โดยการแสดงค่าความถี่สะสมที่เกิดขึ้นร่วมกันของค่าความเข้มจุดภาพที่ i และ j

ตารางที่ 1.3 แสดงผลการคำนวณค่าตัวแปรทางสถิติของ Cooccurrence เมตริกซ์ในรูปที่ 1.16ข-1.16จ

ตัวแปร	$\theta = 0^\circ$	$\theta = 45^\circ$	$\theta = 90^\circ$	$\theta = 135^\circ$
f1	0.003950	0.003060	0.003385	0.002822
f2	82.496696	161.008972	124.535881	176.369751
f3	0.802237	0.766765	0.781602	0.763924
f4	271825760.00	270195840.00	270156640.00	269947232.00
f5	0.258357	0.204972	0.224916	0.189956
f6	244.995270	244.953537	244.786880	244.949478
f7	64187.621094	64023.195312	64004.410156	63994.613281
f8	4.809479	4.907531	4.866956	4.930624
f9	5.966467	6.255219	6.140471	6.317950
f10	41.461372	83.206612	64.213562	88.498375
f11	2.623438	2.982866	2.837941	3.054271
f12	-0.541033	-0.475854	-0.501154	-0.456897
f13	0.938790	0.976531	0.966046	0.977649



รูปที่ 1.16 ซีตโตแกรมแบบสองมิติโดย Cooccurrence เมตริกซ์ที่ระยะทาง $d = 1$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 (ก) อินพุต, (ข) $\theta = 0^\circ$, (ค) $\theta = 45^\circ$, (ง) $\theta = 90^\circ$, (จ) $\theta = -135^\circ$
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุป

การประมวลผลภาพเชิงตัวเลขจะประกอบด้วย การประมวลผลระดับต่ำและการประมวลผลระดับสูง การประมวลผลระดับต่ำจะเป็นการประมวลผลกับข้อมูลภาพเชิงตัวเลขโดยตรง เพื่อที่จะคำนวณหาค่าตัวแปร ซึ่งจะนำไปใช้ในการวิเคราะห์ภาพโดยการประมวลผลในระดับสูงต่อไป

จากหัวข้อต่างๆ ที่ได้กล่าวมาแล้วนั้น เป็นการประมวลผลภาพในระดับต่ำ ซึ่งประกอบด้วย การหาขอบภาพ และการวิเคราะห์เนื้อหาภาพ สำหรับการหาขอบภาพเป็นขั้นตอนที่สำคัญของการวิเคราะห์ภาพเชิงตัวเลข เพื่อที่จะหาขอบเขตของวัตถุแต่ละชิ้นและเป็นการแบ่งแยกวัตถุด้วย ส่วนการวิเคราะห์เนื้อหาภาพเป็นการคำนวณหาค่าของตัวแปรเพื่อนำมาอธิบายพื้นผิวของวัตถุ ซึ่งส่วนใหญ่จะใช้ในการแบ่งแยกวัตถุที่อยู่ในภาพ หรืออาจจะใช้ในการจำแนกวัตถุต่างๆ ที่อยู่ภายในภาพก็ได้



บทที่ 2 ฟัซซีเซต

ฟัซซีเซตเป็นเซตที่แสดงถึงความสัมพันธ์ของสมาชิกภายในกลุ่มแต่ละตัวกับค่าจำกัดความของเซตนั้นๆ โดยความสัมพันธ์นี้จะถูกแสดงในลักษณะของระดับความเป็นสมาชิกที่มีค่าอยู่ในช่วง $[0, 1]$ แทนที่จะแสดงว่าสมาชิกตัวใดเป็นสมาชิกหรือไม่เป็นสมาชิกของเซตนั้นอย่างเซตธรรมดา (ซึ่งมีค่าเป็น $\{0, 1\}$) ถ้าเรากำหนดให้ U เป็นเซตเอกภพสัมพัทธ์ และฟัซซีเซต A มีสมาชิกของเซตเป็น $x_1, x_2, x_3, \dots, x_n$ โดยที่ x_i ($i=1, 2, 3, \dots, n$) เป็นค่าฟัซซีเซต ดังนั้นฟัซซีเซต A สามารถแสดงในรูปแบบของความสัมพันธ์ในฟัซซีเซตได้โดย $\mu_A(x_i)$ ความสัมพันธ์เช่นนี้ในทางทฤษฎีของฟัซซีเซตจะเรียกว่า ฟังก์ชันความเป็นสมาชิก (Membership Function) และสามารถเขียนเป็นสัญลักษณ์ทางคณิตศาสตร์ได้โดย

$$\mu_A(x_i) : U \rightarrow [0, 1] \quad (2.1)$$

โดยที่ $\mu_A(x_i)$ ก็คือฟังก์ชันความเป็นสมาชิกของเซต A

ฟังก์ชันความเป็นสมาชิกของเซตใดก็จะสอดคล้องกับค่าจำกัดความของเซตนั้น และค่าจำกัดความเหล่านี้ก็คือตัวแปรต่างๆ ของฟัซซีเซตที่อยู่ภายใต้ขอบเขตของเซตเอกภพสัมพัทธ์หรือระบบใดๆ ที่เราสนใจ

ตัวแปรฟัซซี (Fuzzy Variable) หรือบางครั้งอาจจะเรียกว่าตัวแปร Linguistic [28-29] ซึ่งก็คือฟัซซีเซตใดๆ ของระบบที่เราสนใจดังที่ได้กล่าวมาแล้วข้างต้น ตัวอย่างเช่น ถ้าเราพิจารณา “อุณหภูมิ” เป็นตัวแปรฟัซซีและถ้าค่าของตัวแปรของเซตนี้คือ {ต่ำ, ปานกลาง, สูง} ซึ่งจะเรียกเซตของค่าตัวแปรฟัซซีนี้ว่า เทอมเซต (Term Set) หรือเซตย่อย และนอกจากนี้ตัวแปรฟัซซีแต่ละตัวอาจจะมีส่วนขยายของ (Hedges หรือ Qualifiers) เพื่อปรับค่าตัวแปรให้มีความให้มีความยืดหยุ่นมากยิ่งขึ้น เช่น จากค่าของตัวแปรอุณหภูมิ “ต่ำ→ต่ำมาก”, “สูง→ค่อนข้างสูง” เป็นต้น คำว่า “มาก” หรือ “ค่อนข้าง” ในที่นี้เป็นส่วนขยายของตัวแปรฟัซซี ซึ่งในทางปฏิบัติสามารถจะแทนด้วยฟังก์ชันทางคณิตศาสตร์ได้ซึ่งจะกล่าวถึงรายละเอียดในหัวข้อถัดไป

2.1 การแทนข้อมูลโดยฟัซซีเซต

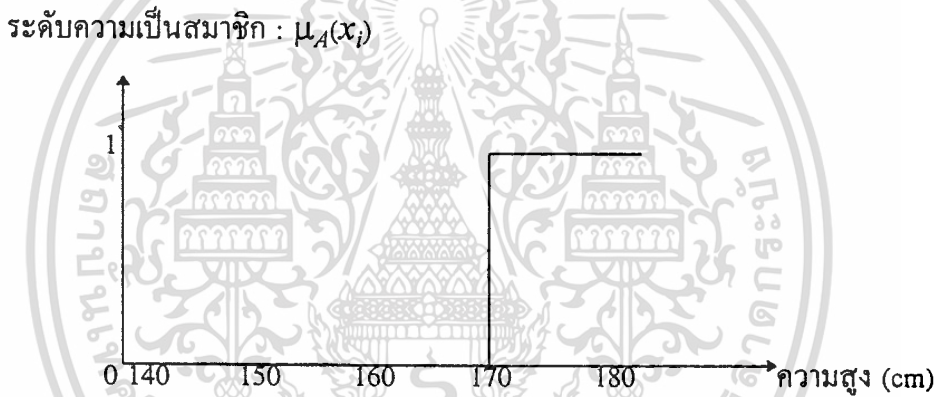
เพื่อให้เข้าใจการแทนข้อมูลโดยฟัซซีเซตได้ง่ายขึ้น เราจะยกตัวอย่างของเหตุการณ์ที่เกิดขึ้นในชีวิตประจำวันคือเราจะพิจารณาความสูงของผู้ชายไทย โดย “ความสูง” ในที่นี้จะเป็นตัวแปรฟัซซีเซต และเพื่อให้เข้าใจความหมายของฟัซซีเซตได้เด่นชัดยิ่งขึ้นเราจะเปรียบเทียบการแทนข้อมูลของฟัซซีเซตกับเซตธรรมดา การแปลงข้อมูลเพื่อคำนวณค่าความเป็นสมาชิกของเซตธรรมดาคงต้องมีการกำหนดค่าเทรซโฮลด์ที่จะนำมาเป็นตัวตัดสินว่าสมาชิกตัวใดในเซตเอกภพสัมพัทธ์ที่เป็นสมาชิกเซตที่กำลังพิจารณาอยู่บ้าง สำหรับในตัวอย่างของความสูงของผู้ชายไทยสมมุติอยู่ในช่วง 140-180 cm (โดเมนของเซตเอกภพสัมพัทธ์) แต่ใน

ที่นี้เราสนใจคนที่สูง ถ้าเรากำหนดให้เซต A เป็นเซตของคนสูง (“สูง” ในกรณีของพืชซึ่งเซตจะเป็นเทอมเซต) ดังนั้นเซต A ในกรณีของเซตธรรมดาจะต้องมีค่าขีดเริ่มเปลี่ยนหรือค่าเทรชโฮลด์ (Threshold) ซึ่งในที่นี้จะกำหนดให้เท่ากับ 170 cm นั่นคือถ้าชายไทยคนใดมีความสูงมากกว่าหรือเท่ากับ 170 cm จะถือว่าคนนั้นสูง (เป็นสมาชิกของเซต A) ซึ่งสามารถเขียนเป็นฟังก์ชันคุณลักษณะ (Characteristic Function) หรือฟังก์ชันความเป็นสมาชิกได้ดังนี้

$$\mu_A(x_i) = \{x_i \geq 170\} \quad (2.2)$$

โดยที่ x_i คือความสูงของผู้ชายไทยคนที่ i

ในสมการที่ (2.2) สามารถนำมาเขียนเป็นกราฟของความเป็นสมาชิกได้ดังรูปที่ 2.1



รูปที่ 2.1 ค่าความเป็นสมาชิกของเซต A ในกรณีที่เป็นเซตธรรมดา

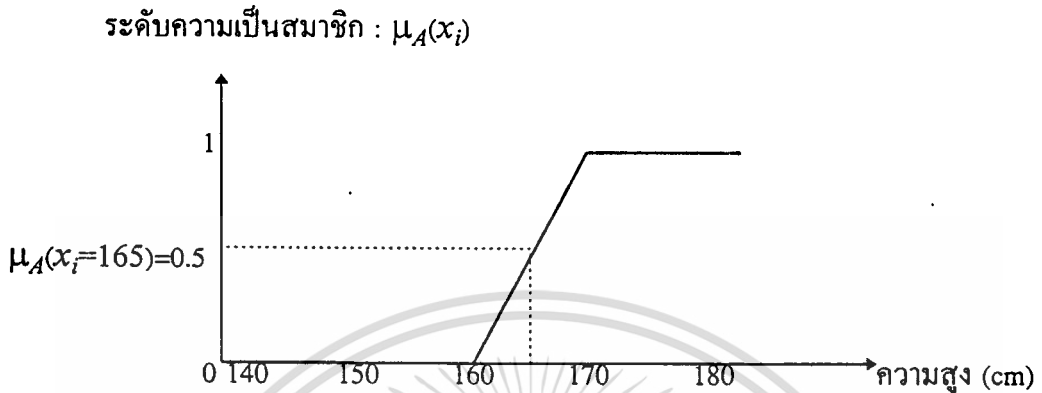
ในกราฟรูปที่ 2.1 บนแนวแกนตั้งแทนค่าระดับความเป็นสมาชิกที่มีค่าเป็น 0 (ไม่เป็นสมาชิกของเซต A) และ 1 (เป็นสมาชิกของเซต A) และจะเห็นว่าคนที่มีความสูงมากกว่าหรือเท่ากับ 170 cm ขึ้นไปเท่านั้นจึงจะถือว่าผู้ชายคนนั้นสูง นอกจากนั้นจะไม่ใช้สมาชิกของเซต A

ในกรณีของพืชซึ่งเซตฟังก์ชันความเป็นสมาชิกจะมีความยืดหยุ่นกว่า และสอดคล้องกับความเป็นจริงมากกว่าเซตธรรมดา ดังในสมการที่ (2.3)

$$\mu_A(x_i) = \begin{cases} \frac{1}{(170-160)}(x_i - 160); & \text{ถ้า } 160 \leq x_i \leq 170 \\ 1; & \text{ถ้า } x_i > 170 \\ 0; & \text{ในกรณีอื่นๆ} \end{cases} \quad (2.3)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในสมการที่ (2.3) เป็นกำหนดค่าความเป็นสมาชิกให้กับเทอมเซตของคนสูง และในกราฟรูปที่ 2.2 แสดงกราฟระดับความเป็นสมาชิกของสมการนี้



รูปที่ 2.2 ค่าความเป็นสมาชิกของเทอมเซต A ในกรณีที่เป็นฟัซซีเซต

เมื่อเปรียบเทียบฟังก์ชันความเป็นสมาชิกในกราฟรูปที่ 2.1 กับ รูปที่ 2.2 จะเห็นว่ากราฟแสดงความเป็นสมาชิกของฟัซซีเซตจะมีความต่อเนื่องในค่าระดับความเป็นสมาชิกในช่วง $[0, 1]$ เช่น คนที่มีความสูง 165 cm ก็จะได้ว่าเป็นคนสูงเช่นกัน แต่เขาจะมีระดับความเป็นสมาชิกของเซตของคนสูงเท่ากับ 0.5 ถ้าเปรียบเทียบกับเซตธรรมดาคนที่มีความสูง 165 cm จะไม่เป็นสมาชิกของเซต A ทั้งที่เขามีความสูงแตกต่างจากคนสูงไม่กี่เซนติเมตรเท่านั้น หรือแม้แต่คนที่มีความสูง 169 cm ซึ่งมีความสูงต่างจากคนที่ถือว่าสูงเพียง 1 cm เท่านั้นแต่ในเซตธรรมดาจะถือว่าเขาไม่ใช่คนสูงซึ่งจะไม่ตรงกับความเป็นจริง แต่สำหรับการแทนข้อมูลโดยฟัซซีเซตจะถือว่าคนนี้เป็นเซตของคนสูงด้วยค่าระดับความเป็นสมาชิกเท่ากับ 0.9 (คือเกือบจะเป็นคนสูงเลยทีเดียว)

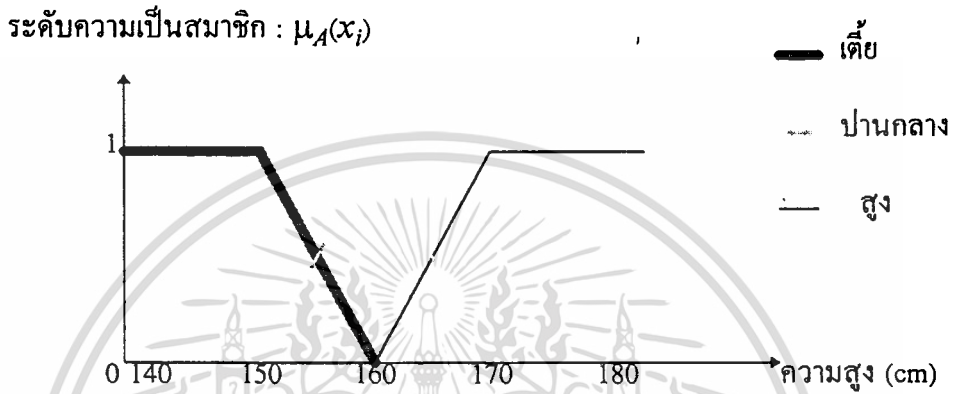
2.2 ฟังก์ชันความเป็นสมาชิก

ฟังก์ชันความเป็นสมาชิกหรือฟังก์ชันคุณลักษณะในกรณีของฟัซซีเซตไม่ใช่แค่การ Normalization ข้อมูลอย่างที่หลายๆ คนเข้าใจกัน แต่ฟังก์ชันนี้จะต้องถูกกำหนดขึ้นตามคำจำกัดความของเทอมเซตที่มีความยืดหยุ่นและตรงตามความเป็นจริง แต่ก่อนที่จะพิจารณาถึงฟังก์ชันความเป็นสมาชิกชนิดต่างๆ เราจะไปดูถึงคุณสมบัติบางประการของฟัซซีเซตก่อน

- เซตเอกภพสัมพัทธ์

เมื่อพิจารณาถึงกราฟรูปที่ 2.2 แกนบนของกราฟจะแทน โดเมนของฟัซซีเซต ส่วนแกนตั้งจะแทนค่าระดับความเป็นสมาชิก แต่ในกราฟรูปที่ 2.2 มีเทอมเซตเพียงเทอมเดียว ซึ่งเป็นเทอมของคนสูงเท่านั้น เล็กส่วนนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานาน นอนยาดเห็นาเบไซบระเยชิตานการค่า เพื่อให้ครอบคลุมกลุ่มประชากรที่เราสนใจจึงเพิ่มเทอมเซตอีกสองเทอมคือเทอมเซตของคน “เตี้ย” และไมวารณเดตทั้งสน ออกทงหามมเหตดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เทอมเซตของคนที่มีความสูง “ปานกลาง” ดังรูปที่ 2.3 ซึ่งในกรณีที่โดเมนของตัวแปรฟัซซีที่ครอบคลุมประชากรทั้งหมดที่ทำการศึกษา เราจะเรียกโดเมนของตัวแปรนี้ว่า เซตเอกภพสัมพัทธ์ และในแต่ละเทอมเซตก็จะมีโดเมนของตัวเองเช่น เทอมเซตของคนเตี้ยจะมีโดเมนอยู่ในช่วง 140-160 cm, เทอมเซตของคนที่มีความสูงปานกลางจะมีโดเมนอยู่ในช่วง 150-170 cm, และเทอมเซตของคนที่สูงจะมีโดเมนอยู่ในช่วง 160-180 cm



รูปที่ 2.3 แสดงกราฟฟังก์ชันความเป็นสมาชิกของตัวแปร “ความสูง” ที่ประกอบด้วย 3 เทอมเซต

- ซัพพอร์ตเซต

ในกรณีที่ x_i ($\forall i$) ในบางเทอมเซตมีค่าน้อยกว่าโดเมนของมัน เช่น ในเทอมเซตของคนเตี้ยซึ่งมีโดเมนอยู่ในช่วง [140-160] เมื่อทำการวัดความสูงของประชากรกลุ่มนี้ปรากฏว่าความสูงที่วัดได้จริงๆ ของพวกเขาอยู่ในช่วง [145-160] ซึ่งช่วงของข้อมูลที่วัดได้จริงๆ จะเรียกว่า ซัพพอร์ตเซต (Support Set) ของเทอมเซตนั้นๆ สำหรับซัพพอร์ตเซตจะมีประโยชน์ในการปรับขนาดของเทอมเซตโดยอัตโนมัติ

- การเทรซโฮลด์ฟัซซีเซต

การเทรซโฮลด์หรืออัลฟาคัท (Alpha-Cut, หรือ α -Cut) ของฟัซซีเซตจะมีความสัมพันธ์กับซัพพอร์ตเซต คือค่าระดับของ α ($0 \leq \alpha \leq 1$) จะเป็นค่าเทรซโฮลด์ภายในขอบเขตโดเมนของเทอมเซตที่ค่า α เป็นค่าขอบเขตล่างของซัพพอร์ตเซต หรืออาจจะเป็นค่าใดๆ ที่เราสนใจในช่วง $[0, 1]$ α -Cut มีอยู่สองชนิดด้วยกันคือ

1. Strong α -Cut ซึ่งกำหนดได้โดย

$$\mu_A(x_i) = \begin{cases} \mu_A(x_i); & \text{ถ้า } \mu_A(x_i) \geq \alpha \\ 0; & \text{ในกรณีอื่นๆ} \end{cases} \quad (2.4)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Weak α -Cut สามารถกำหนดได้โดย

$$\mu_A(x_i) = \begin{cases} \mu_A(x_i); & \text{ถ้า } \mu_A(x_i) > \alpha \\ 0; & \text{ในกรณีอื่นๆ} \end{cases} \quad (2.5)$$

เมื่อ A เป็นเทอมเซตใดๆ ประโยชน์ของ α -Cut มีอยู่สองประการด้วยกัน อย่างแรกคือ Strong α -Cut ใช้ในการกำหนดซัพพอร์ตเซต อย่างที่สองคือทำให้มีความยืดหยุ่นสำหรับระบบการควบคุมที่ใช้ฟัซซี่เซตในกรณีที่มีการอินเตอร์เซกชันกันหลายเทอมซึ่งจะได้อธิบายในหัวข้อของกฎฟัซซี่เซตต่อไป ตอนนี้เราจะพิจารณาถึงฟังก์ชันความเป็นสมาชิกแบบต่างๆ ที่จะใช้ในการประมาณค่าระดับความเป็นสมาชิกของแต่ละเทอมเซต โดยมีทั้งแบบที่ประมาณค่าเป็นเชิงเส้น เช่น รูปสามเหลี่ยม, รูปสี่เหลี่ยมคางหมู และแบบที่ประมาณค่าโดยฟังก์ชันต่อเนื่อง เช่น S , S^{-1} , ระฆังคว่ำ เป็นต้น ซึ่งมีรายละเอียดดังนี้

2.2.1 สมการเชิงเส้น

ฟังก์ชันความเป็นสมาชิกแบบสมการเชิงเส้นที่ใช้ในแปลงค่าความสัมพันธ์ของโดเมนไปยังเรนจ์สำหรับข้อมูลที่มีความสัมพันธ์เป็นเชิงเส้น ดังสมการที่ (2.3)

2.2.2 รูปสามเหลี่ยม

การกำหนดฟังก์ชันความเป็นสมาชิกโดยใช้รูปสามเหลี่ยม นิยมใช้กันมากในระบบควบคุมที่ใช้ฟัซซี่เซต เทอมเซตที่กำหนดโดยฟังก์ชันนี้จะต้องมีค่าที่เหมาะสมที่สุดอยู่เพียงค่าเดียวที่ทำให้ค่าความเป็นสมาชิกเท่ากับหนึ่ง ส่วนค่าอื่นๆ จะมีค่าความเป็นสมาชิกลดลงเรื่อยๆ เมื่อยิ่งห่างจากค่านี้มากขึ้น ในตัวอย่างของตัวแปรความสูง เทอมเซต “ปานกลาง” มีฟังก์ชันความเป็นสมาชิกเป็นรูปสามเหลี่ยม ถ้ากำหนดให้ $a \leq b \leq c$ เมื่อ a , b , และ c เป็นเลขจำนวนจริงใดๆ ดังนั้นสมการของฟังก์ชันความเป็นสมาชิกของรูปสามเหลี่ยมสามารถกำหนดได้ดังนี้

$$\mu_A(x_i) = \begin{cases} \frac{1}{b-a}(x_i - a); & \text{ถ้า } a \leq x_i \leq b \text{ และ } a \neq b \\ \frac{1}{c-b}(c - x_i); & \text{ถ้า } b < x_i \leq c \text{ และ } b \neq c \\ 0; & \text{สำหรับกรณีอื่นๆ} \end{cases} \quad (2.6)$$

2.2.3 รูปสี่เหลี่ยมคางหมู

สำหรับฟังก์ชันความเป็นสมาชิกที่กำหนดด้วยสมการรูปสี่เหลี่ยมคางหมู นิยมใช้ในระบบควบคุมที่ใช้ฟัซซี่เซตเช่นกัน แต่จะพบน้อยกว่าฟังก์ชันความเป็นสมาชิกที่กำหนดด้วยสมการรูปสามเหลี่ยม ฟังก์ชันความเป็นสมาชิกที่จะกำหนดด้วยสมการรูปสี่เหลี่ยมคางหมูจะต้องมีช่วงของค่าที่เหมาะสมมากที่สุดอยู่กลุ่ม

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนึ่งที่ทำให้ค่าความเป็นสมาชิกของเทอมเซตนั้นๆ มีค่าเท่ากับหนึ่ง นอกจากนั้นจะมีค่าความเป็นสมานชิกน้อยลงเรื่อยๆ เมื่อยิ่งห่างจากข้อมูลกลุ่มนี้ ถ้ากำหนดให้ $a \leq b \leq c \leq d$ เมื่อ $a, b, c,$ และ d เป็นเลขจำนวนจริงใดๆ ดังนั้นสมการของฟังก์ชันความเป็นสมาชิกของรูปสี่เหลี่ยมคางหมูสามารถกำหนดได้ดังนี้

$$\mu_A(x_i) = \begin{cases} \frac{1}{b-a}(x_i - a); & \text{ถ้า } a \leq x_i \leq b \text{ และ } a \neq b \\ 1; & \text{ถ้า } b < x_i \leq c \\ \frac{1}{d-c}(d - x_i); & \text{ถ้า } c < x_i \leq d \text{ และ } c \neq d \\ 0; & \text{สำหรับกรณีอื่นๆ} \end{cases} \quad (2.7)$$

2.2.4 เส้นโค้งรูปตัว S

ฟังก์ชันความเป็นสมาชิกที่กำหนดด้วยเส้นโค้งรูปตัว S (S-Curve, Sigmoid หรือ Logistic) เป็นการกำหนดความสัมพันธ์ระหว่างโดเมนไปยังเรนจ์แบบไม่เป็นเชิงเส้น ตัวแปรที่เหมาะสมที่จะใช้ฟังก์ชันนี้คือพวกอายุการใช้งานของอุปกรณ์ต่างๆ, ตัวแปรส่วนใหญ่ที่ใช้ในทางการประมวลผลภาพ [1] เป็นต้น ซึ่งตัวแปรพวกนี้มักจะมีความสัมพันธ์แบบไม่เป็นเชิงเส้นที่เป็นรูปตัว S หรือ S^{-1} ถ้ากำหนดให้ $a \leq b \leq c$ เมื่อ $a, b,$ และ c เป็นเลขจำนวนจริงใดๆ ดังนั้นสมการของฟังก์ชันความเป็นสมาชิกของเส้นโค้งรูปตัว S สามารถกำหนดได้ดังสมการที่ (2.8) และ S^{-1} สามารถคำนวณได้โดยหาค่าคอมพลิเมนต์ของสมการที่ (2.8) ($S^{-1} = 1 - \mu_A(x_i)$ โดยที่ $\mu_A(x_i)$ ได้จากสมการที่ (2.8))

$$\mu_A(x_i) = \begin{cases} 0; & \text{ถ้า } x_i \leq a \\ 2 \left(\frac{x_i - a}{c - a} \right)^2; & \text{ถ้า } a < x_i \leq b \\ 1 - 2 \left(\frac{x_i - a}{c - a} \right)^2; & \text{ถ้า } b < x_i \leq c \\ 1; & \text{ถ้า } x_i > a \end{cases} \quad (2.8)$$

2.2.5 รูปประฆังคว่ำ

เทอมเซตที่ใช้ฟังก์ชันความเป็นสมาชิกแบบรูปประฆังคว่ำ จะมีคุณสมบัติของข้อมูลคล้ายกับสมการรูปสามเหลี่ยม แต่ข้อมูลของโดเมนที่แปลงไปยังเรนจ์จะมีความสัมพันธ์ในลักษณะไม่เป็นเชิงเส้น สมการของรูปประฆังคว่ำมีด้วยกันหลายสมการแต่ที่ใช้กันมากคือ ฟังก์ชัน π [1, 30], ฟังก์ชัน Gaussian และ ฟังก์ชัน Beta ซึ่งสามารถกำหนดได้ดังนี้

- ฟังก์ชัน π คำนวณได้จากสมการที่ (2.8)

$$S(x_i; a, b, c) = \begin{cases} 0; & \text{ถ้า } x_i \leq a \\ 2 \left(\frac{x_i - a}{c - a} \right)^2; & \text{ถ้า } a < x_i \leq b \\ 1 - 2 \left(\frac{x_i - a}{c - a} \right)^2; & \text{ถ้า } b < x_i \leq c \\ 1; & \text{ถ้า } x_i > a \end{cases}$$

โดยที่ $b = \frac{c - a}{2}$

ดังนั้นฟังก์ชัน π สามารถกำหนดได้โดย

$$\mu_A(x_i) = \begin{cases} S(x_i; c - b, c - \frac{b}{2}, c); & \text{ในกรณีที่ } x_i \leq c \\ 1 - S(x_i; c, c + \frac{b}{2}, c + b); & \text{ในกรณีที่ } x_i > c \end{cases} \quad (2.9)$$

- ฟังก์ชัน Gaussian

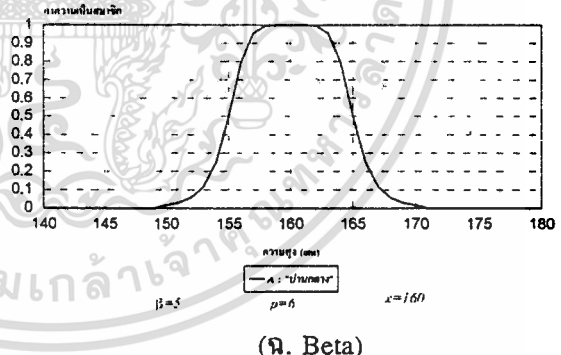
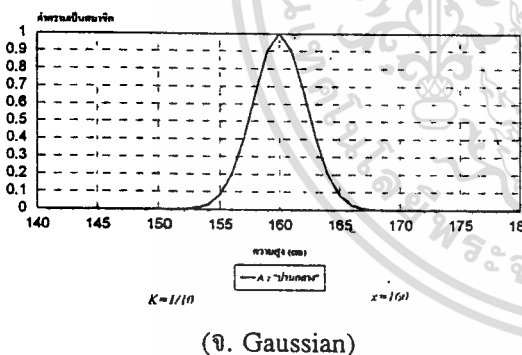
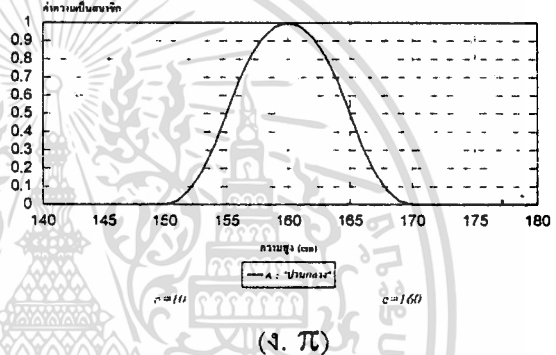
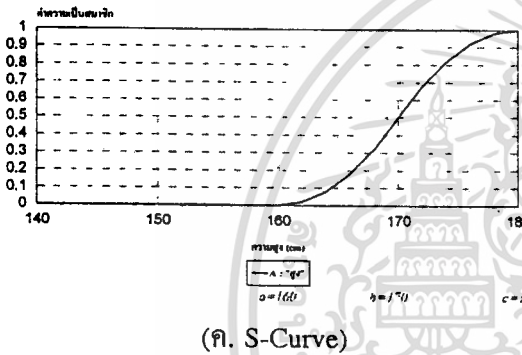
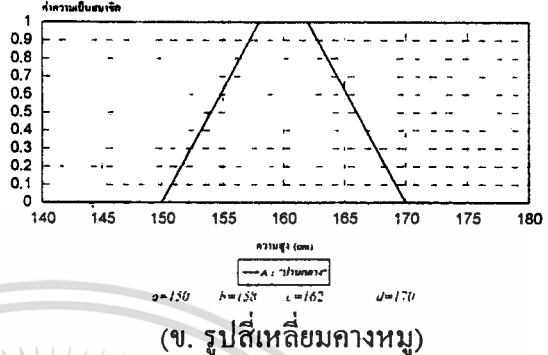
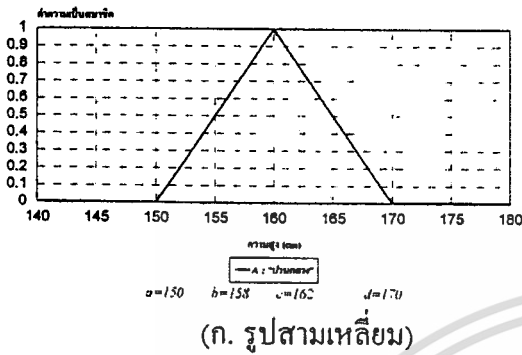
$$\mu_A(x_i) = e^{-K(\bar{x} - x_i)^2} \quad (2.10)$$

โดยที่ $K = \frac{1}{2\sigma^2}$ ซึ่งเป็นโดเมนของฟังก์ชันเซต A ที่มีค่าเฉลี่ยเป็น \bar{x} และส่วนเบี่ยงเบนมาตรฐาน σ

- ฟังก์ชัน Beta ฟังก์ชันนี้จะมีคุณสมบัติคล้ายกับสมการรูปสี่เหลี่ยมคางหมู คือจะมีช่วงของข้อมูลที่เหมาะสมกับเทอมเซตนั้นๆ อยู่กลุ่มหนึ่งที่ทำให้ค่าความเป็นสมาชิกของเทอมเซตมีค่าเข้าใกล้ 1 ซึ่งของมุลกลุ่มนี้จะมีความกว้างของช่วงเท่ากับค่า β ซึ่งถ้าค่า $p \rightarrow \infty$ จะทำให้ข้อมูลในช่วง β มีค่าเข้าใกล้ 1 มากยิ่งขึ้น โดยมีจุดศูนย์กลางอยู่ที่ \bar{x} ดังแสดงในสมการที่ (2.11)

$$\mu_A(x_i) = \frac{1}{1 + \left(\frac{x_i - \bar{x}}{\beta} \right)^p} \quad (2.11)$$

จากสมการที่ (2.6) ถึง (2.11) ในรูปที่ 2.4 แสดงรูปภาพของฟังก์ชันความเป็นสมาชิกในแต่ละเทอมเซต ซึ่งมีทั้งฟังก์ชันแบบเชิงเส้น และแบบต่อเนื่อง



รูปที่ 2.4 แสดงกราฟฟังก์ชันความเป็นสมาชิกชนิดต่างๆ

2.3 ตัวดำเนินการฟัซซีเซต

ตัวดำเนินการของฟัซซีเซตส่วนใหญ่แล้วจะมีตัวดำเนินการคล้ายกับเซตธรรมดา เช่น อินเตอร์เซกชัน, ยูเนียน, คอมพลีเมนต์ เป็นต้น ซึ่งตัวดำเนินการเหล่านี้จะใช้ในการอนุมานฟัซซีเซตเพื่อประโยชน์ในการตัดสินใจโดยการประมาณค่าของตัวแปร ในที่นี้เราจะเสนอตัวดำเนินการของฟัซซีเซตบางส่วนเฉพาะที่ใช้ในแบบจำลองการตัดสินใจของระบบควบคุมโดยใช้ฟัซซีเซตเท่านั้น ส่วนตัวดำเนินการอื่นของฟัซซีเซตสามารถค้นคว้าได้จาก [31-34] สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยามให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดให้ U เป็นเซตเอกภพสัมพัทธ์, A, B เป็นทอมนเซตใดๆ ที่อยู่ใน U และตัวดำเนินการพื้นฐานต่างๆ สามารถกำหนดได้ดังนี้

2.3.1 เซตย่อย

A เป็นเซตย่อยของ B ($A \subseteq B$) ก็ต่อเมื่อ

$$\mu_A(x_i) \leq \mu_B(x_i); \quad \forall x_i \in U \quad (2.12)$$

2.3.2 คอมพลีเมนต์

A เป็นคอมพลีเมนต์ของ B ($A=B'$) ก็ต่อเมื่อ

$$\mu_A(x_i) = \mu_{B'}(x_i) = 1 - \mu_B(x_i); \quad \forall x_i \in U \quad (2.13)$$

2.3.3 อินเตอร์เซกชันของ A และ B ($A \cap B$) กำหนดได้โดย

$$\mu_{A \cap B}(x_i) = \text{Min}(\mu_A(x_i), \mu_B(x_i)); \quad \forall x_i \in U \quad (2.14)$$

2.3.4 ยูเนียนของ A กับ B ($A \cup B$) กำหนดได้โดย

$$\mu_{A \cup B}(x_i) = \text{Max}(\mu_A(x_i), \mu_B(x_i)); \quad \forall x_i \in U \quad (2.15)$$

2.3.5 A ยกกำลังด้วยค่า α (A^α) กำหนดได้โดย

$$\mu_{A^\alpha}(x_i) = (\mu_A(x_i))^\alpha; \quad \forall x_i \in U \quad (2.16)$$

2.3.6 Concentration ของเซต A กำหนดได้โดย

$$\text{CON}(A) \Rightarrow \mu_{\text{CON}(A)}(x_i) = (\mu_A(x_i))^2; \quad \forall x_i \in U \quad (2.17)$$

2.3.7 Dilation ของเซต A กำหนดได้โดย

$$\text{DIL}(A) \Rightarrow \mu_{\text{DIL}(A)}(x_i) = (\mu_A(x_i))^{0.5}; \quad \forall x_i \in U \quad (2.18)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

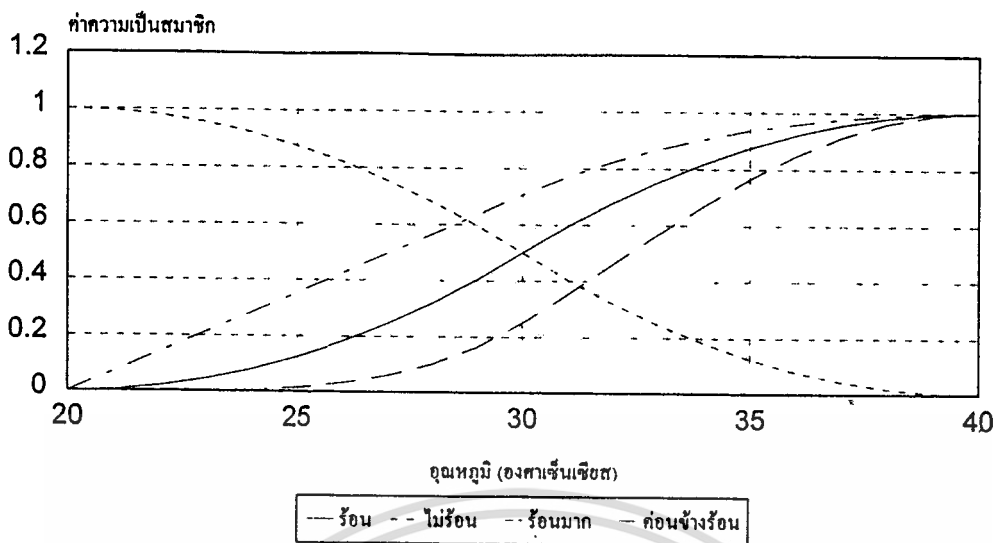
2.3.8 Contrast Intensification ของเซต A กำหนดได้โดย

$$INT(A) \Rightarrow \mu_{INT(A)}(x_i) = \begin{cases} 2(\mu_A(x_i))^2; & \text{ถ้า } 0 \leq \mu_A(x_i) \leq 0.5 \\ 1 - 2(1 - \mu_A(x_i))^2; & \text{ถ้า } 0.5 < \mu_A(x_i) \leq 1 \end{cases} \quad (2.19)$$

2.4 ส่วนขยายของฟัซซี่เซต

ส่วนขยาย (Hedge) ของฟัซซี่เซตจะมีคุณสมบัติคล้ายกับค่าคุณศัพท์ในประโยคที่เราใช้พูดคุยติดต่อกัน ซึ่งจะขยายให้มีใจความของประโยคให้ได้ความหมายที่ชัดเจนยิ่งขึ้นหรือทำให้ความหมายเปลี่ยนไป สำหรับ Hedge ของฟัซซี่เซตนั้นจะขยายขอบเขตของตัวแปรฟัซซี่เซต เพื่อให้การประมาณค่าของตัวแปรมีความถูกต้องมากยิ่งขึ้น เมื่อขยายขอบเขตใดด้วย Hedge จะส่งผลให้รูปร่างของฟังก์ชันความเป็นสมาชิกของเซตนั้นๆ เปลี่ยนไปและในบางกรณีอาจจะทำให้ได้เซตใหม่ขึ้นมา Hedge แต่ละตัวจะเป็นตัวดำเนินการของฟัซซี่เซต เช่น Hedge คำว่า “มาก” Zadeh [28, 34] ได้ให้ความหมายของคำนี้ไว้โดยการยกกำลังค่าความเป็นสมาชิกทุกตัวในเซตด้วยสอง ซึ่งก็คือตัวดำเนินการ Concentration ของฟัซซี่เซต ส่วน Hedge ของคำว่า “ค่อนข้าง” Zadeh [28, 34] ได้ให้ความหมายของคำนี้ไว้โดยการถอดรากที่สองของค่าความเป็นสมาชิกทุกตัวในเซตหรือก็คือตัวดำเนินการ Dilation ของฟัซซี่เซตนั่นเอง

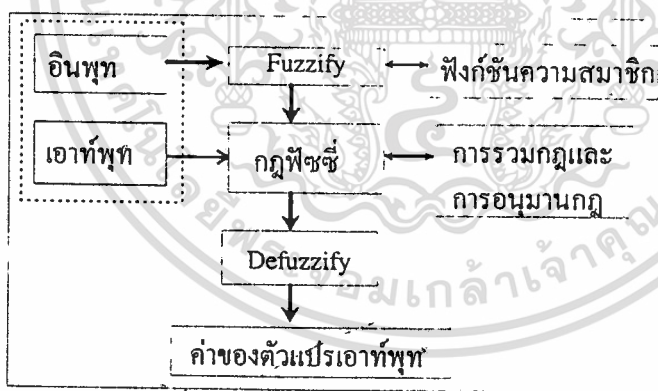
สาเหตุที่ Zadeh แทนความหมายของ Hedge ของคำว่า “มาก” และ “ค่อนข้าง” ด้วยตัวดำเนินการยกกำลังคือยกกำลังด้วย 2 และ 0.5 ตามลำดับ ก็เพื่อทำการแปลงค่าความเป็นสมาชิกให้เหมาะสมกับเซตนั้นๆ ยิ่งขึ้น อย่างเช่น Hedge คำว่า “มาก” เมื่อนำไปขยายขอบเขตใด แสดงว่าค่อนข้างจะแน่ใจว่าข้อมูลของเซตนั้นต้องการความถูกต้องสูง และเพื่อให้แน่ใจว่าค่าระดับความเป็นสมาชิกของเซตนี้มีความถูกต้องสูงจริงๆ จึงทำการลดค่าระดับความเป็นสมาชิกลงมาโดยตัวดำเนินการ Concentration ส่วน Hedge คำว่า “ค่อนข้าง” แสดงถึงความไม่แน่ใจในผลตอบสนองของเซตนั้นๆ จึงทำการเพิ่มค่าระดับความเป็นสมาชิกให้มากขึ้นโดยการยกกำลังด้วย 0.5 ความจริงความหมายของ Hedge เป็นการแปลความหมายตามอำเภอใจ ดังนั้นคำว่า “มาก” อาจจะไม่จำเป็นต้องยกกำลังสองก็ได้ เช่นอาจจะยกกำลังด้วย 1.3 หรือ 1.8 เป็นต้น สำหรับเซตตัวอื่นๆ ที่มีกบพ้อยคือตัวดำเนินการคอมพลิเมนต์หรือ Hedge คำว่า “ไม่”, ส่วน Hedge ที่ใช้ในการเปรียบเทียบเช่น “มากกว่า” หรือ “สูงกว่า”, “น้อยกว่า” หรือ “ต่ำกว่า” เป็นต้น รูปที่ 2.5 แสดง Hedge ของตัวอุณหภูมิ ซึ่งเป็นอุณหภูมิห้อง



รูปที่ 2.5 Hedge ของตัวแปรอุณหภูมิ

2.5 แบบจำลองฟัซซี่

จากหัวข้อที่กล่าวมาแล้วทั้งหมดในเรื่องของฟัซซี่เซต จะถูกนำมาใช้ในแบบจำลองฟัซซี่ (Fuzzy Model) [35-36] ซึ่งเป็นแบบจำลองที่ใช้ในการอนุมานค่าของตัวแปร ซึ่งมีขั้นตอนการทำงานดังบล็อกไดอะแกรมในรูปที่ 2.6



รูปที่ 2.6 บล็อกไดอะแกรมของแบบจำลองฟัซซี่ลอจิก

จากบล็อกไดอะแกรมของแบบจำลองของฟัซซี่ลอจิก สามารถแยกอธิบายเป็นขั้นตอนได้ดังนี้

2.5.1 อินพุตและเอาต์พุตของแบบจำลองฟัซซี่ลอจิกก็คือ ตัวแปรของฟัซซี่เซต ซึ่งตัวแปรแต่ละตัว อาจจะมีเทอมเซตได้หลายเทอม อินพุตซึ่งเป็นตัวแปรของระบบ (State Variables) อาจจะมีที่ตัวก็ได้ซึ่งจะขึ้นอยู่กับระบบ สำหรับตัวแปรของเอาต์พุตหรือที่เรียกว่า Action Variables ซึ่งเป็นตัวแปรที่ถูกประมาณค่า โดยแบบจำลองของฟัซซี่ลอจิกเพื่อนำไปใช้ในการออกแบบกฎฟัซซี่เพื่อใช้ในการควบคุมระบบต่อไป ปัญหาที่เกิดขึ้นในขั้นตอนนี้ก็คือ การกำหนดจำนวนเทอมเซตของแต่ละตัวแปรว่าควรจะมีสักกี่เทอม จึงจะ ไม่ว่าจะกรณีใดข้างต้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำให้การอนุมานค่าของตัวแปรเอาต์พุตมีความใกล้เคียงค่าของตัวแปรที่แท้จริงมากที่สุด ซึ่งจุดนี้ผู้ออกแบบแบบจำลองฟัซซี่จะต้องมีความรู้เกี่ยวกับระบบที่กำลังออกแบบอยู่พอสมควร

2.5.2 Fuzzify เป็นขั้นตอนการแปลงค่าของตัวแปรของอินพุตไปเป็นค่าระดับความเป็นสมาชิกของแต่ละเทอมเซตโดยใช้ฟังก์ชันความเป็นสมาชิกที่ได้เสนอไปแล้ว การเลือกใช้ฟังก์ชันความเป็นสมาชิก เราต้องเลือกให้เหมาะสมกับคุณสมบัติของตัวแปร เช่นถ้าตัวแปรมีคุณสมบัติเป็นเชิงเส้น จะต้องเลือกฟังก์ชันความเป็นสมาชิกที่เป็นเชิงเส้นด้วยซึ่งอาจจะเป็นสมการรูปสามเหลี่ยม และรูปสี่เหลี่ยมคางหมู ส่วนตัวแปรที่มีคุณสมบัติไม่เป็นเชิงเส้นก็ต้องใช้ฟังก์ชันความเป็นสมาชิกที่ไม่เป็นเชิงเส้นด้วยเช่น ฟังก์ชันรูปตัว S, π , Beta, และ Gaussian

2.5.3 กฎฟัซซี่ (Fuzzy Rules) จะประกอบด้วยประพจน์ของฟัซซี่ โดยกฎฟัซซี่จะอยู่ในรูปของ IF-THEN เช่น

IF X is *small* THEN Y is *positive* .

ที่ X และ Y เป็นตัวแปรฟัซซี่ ส่วน *small* และ *positive* เป็นเทอมเซตที่สอดคล้องกับตัวแปร X , Y และประพจน์ของฟัซซี่เซตคือ X is *small* และ Y is *positive* ประพจน์ที่ตามหลัง IF จะเป็นส่วนของเงื่อนไขที่เรียกว่า Antecedent ซึ่งก็คือส่วนของอินพุตของแบบจำลองฟัซซี่ลอจิกนั่นเอง และประพจน์ที่ตามหลัง THEN จะเรียกว่า Consequent หรือส่วนของเอาต์พุตของแบบจำลองฟัซซี่ลอจิก Antecedent อาจประกอบด้วยหลายประพจน์ ซึ่งจำนวนของประพจน์จะขึ้นอยู่กับจำนวนตัวแปรของอินพุตและจำนวนของเทอมเซตของตัวแปรแต่ละตัว เช่นถ้าอินพุตมีสองตัวแปรในส่วนของ Antecedent จะมีสองประพจน์เชื่อมต่อกันในกฎหนึ่งกฎ และในการเชื่อมต่อกันของประพจน์จะต้องมีตัวเชื่อมซึ่งในแบบจำลองฟัซซี่ลอจิกจะมีตัวเชื่อมอยู่สองตัวด้วยกันคือ ยูเนียน, และ อินเตอร์เซกชัน เช่น ในกรณีที่เชื่อมด้วยยูเนียน

IF X is *small* OR T is *cool* THEN Y is *positive*

และในกรณีที่เชื่อมด้วยอินเตอร์เซกชัน

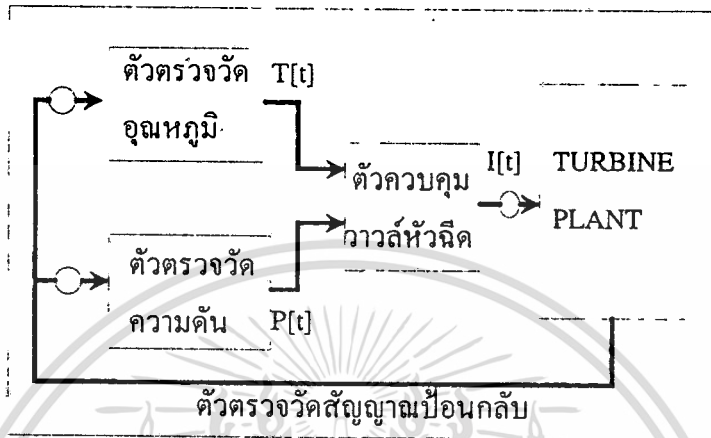
IF X is *small* AND T is *cool* THEN Y is *positive*

นอกจากนี้ในกฎฟัซซี่เทอมเซตอาจจะมี Hedge มาขยายเช่น *very* และ *large*

IF X is *very small* THEN Y is *large positive*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

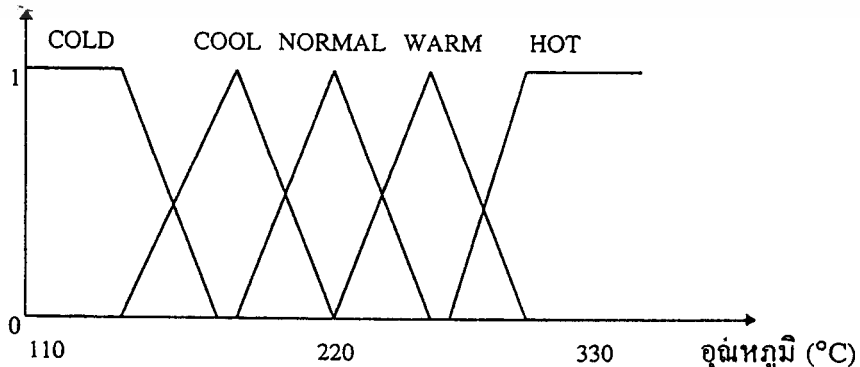
สำหรับจำนวนกฎในแบบจำลองฟัซซี่ลอจิกของระบบหนึ่งๆ จะขึ้นอยู่กับจำนวนของตัวแปรอินพุทและจำนวนเทอมของแต่ละตัวแปร เช่นถ้าอินพุทมีสองตัวแปรและตัวแปรแต่ละตัวมี 5 เทอมเซต ซึ่งจำนวนกฎทั้งหมดจะเท่ากับ 25 กฎ ตัวอย่างแบบจำลองฟัซซี่ลอจิกที่ใช้กับระบบของ Stream Turbine [36] ซึ่งมีแผนผังการควบคุมดังรูปที่ 2.7



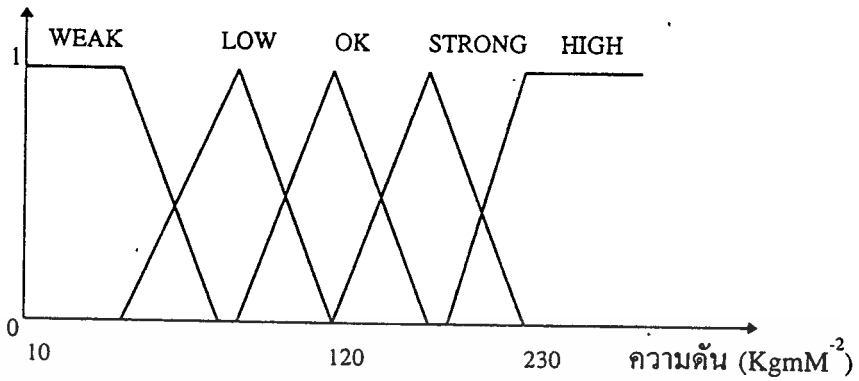
รูปที่ 2.7 แผนผังการควบคุม Stream Turbine

จากแผนผังของกระบวนการควบคุมของ Stream Turbine อินพุทของแบบจำลองฟัซซี่ลอจิกมีตัวแปรสองตัวคือ อุณหภูมิ และความดัน ส่วนเอาต์พุทมีตัวแปรหนึ่งตัวคือการเปิดปิดวาล์ว ตัวแปรของอินพุทแต่ละตัวจะมีเซตอย่างละ 5 เทอมคือ ถ้ากำหนดให้ $T[t]$ เป็นตัวแปรของอุณหภูมิที่วัดได้ในเวลา t ใดๆ ซึ่งมีค่าของตัวแปรนี้อยู่ 5 ค่าด้วยกันคือ $T[t] = \{cold, cool, normal, warm, hot\}$ และสำหรับตัวแปรของความดันที่วัดได้ในเวลา t ใดๆ ซึ่งมีค่าของตัวแปรนี้อยู่ 5 ค่าเช่นกันคือ $P[t] = \{weak, low, ok, strong, high\}$ และตัวแปรของเอาต์พุท $I[t]$ มี 7 เทอมเซตคือ $I[t] = \{NL, NM, NS, ZR, PS, PM, PL\}$ ในรูปที่ 2.8 แสดงเทอมเซตของตัวแปรทั้ง 3 ตัวของระบบการควบคุม Stream Turbine

ระดับความเป็นสมาชิก

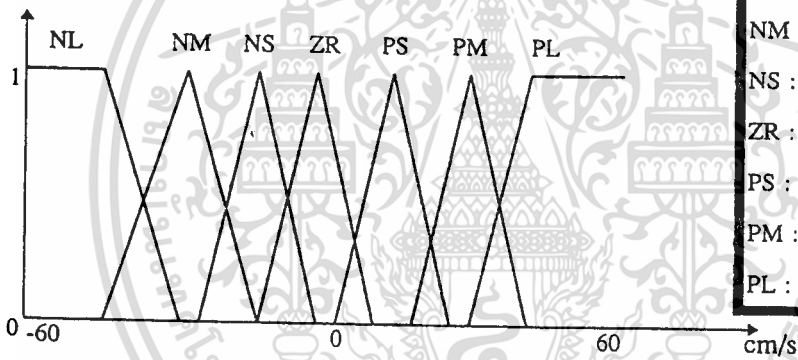


ระดับความเป็นสมาชิก



(ข)

ระดับความเป็นสมาชิก



- NL : Negative Large
- NM : Negative Medium
- NS : Negative Small
- ZR : Approximately Zero
- PS : Positive Small
- PM : Positive Medium
- PL : Positive Large

(ค)

รูปที่ 2.8 แสดงเทอมเซตของตัวแปร (ก) อุณหภูมิ (ข) ความดัน (ค) การปิดเปิดวาล์ว

เมื่อออกแบบเทอมเซตของตัวแปรแต่ละตัวได้แล้ว ก็นำเทอมเซตของตัวแปรแต่ละตัวมาเขียนเป็นกฎฟัซซี่ได้ดังนี้

- IF $T[t]$ is cold AND $P[t]$ is weak THEN $I[t]$ is positive large
- IF $T[t]$ is cool AND $P[t]$ is weak THEN $I[t]$ is positive large
- IF $T[t]$ is normal AND $P[t]$ is weak THEN $I[t]$ is positive medium
- IF $T[t]$ is warm AND $P[t]$ is weak THEN $I[t]$ is positive medium
- IF $T[t]$ is hot AND $P[t]$ is weak THEN $I[t]$ is positive small
- IF $T[t]$ is cold AND $P[t]$ is low THEN $I[t]$ is positive medium
- IF $T[t]$ is cool AND $P[t]$ is low THEN $I[t]$ is positive medium
- IF $T[t]$ is normal AND $P[t]$ is low THEN $I[t]$ is positive small

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IF $T[t]$ is <i>warm</i> AND $P[t]$ is <i>low</i>	THEN $I[t]$ is <i>positive small</i>
IF $T[t]$ is <i>hot</i> AND $P[t]$ is <i>low</i>	THEN $I[t]$ is <i>positive small</i>
IF $T[t]$ is <i>cool</i> AND $P[t]$ is <i>ok</i>	THEN $I[t]$ is <i>positive small</i>
IF $T[t]$ is <i>cool</i> AND $P[t]$ is <i>ok</i>	THEN $I[t]$ is <i>approximately zero</i>
IF $T[t]$ is <i>normal</i> AND $P[t]$ is <i>ok</i>	THEN $I[t]$ is <i>approximately zero</i>
IF $T[t]$ is <i>warm</i> AND $P[t]$ is <i>ok</i>	THEN $I[t]$ is <i>negative small</i>
IF $T[t]$ is <i>hot</i> AND $P[t]$ is <i>ok</i>	THEN $I[t]$ is <i>negative medium</i>
IF $T[t]$ is <i>cold</i> AND $P[t]$ is <i>strong</i>	THEN $I[t]$ is <i>negative small</i>
IF $T[t]$ is <i>cool</i> AND $P[t]$ is <i>strong</i>	THEN $I[t]$ is <i>negative medium</i>
IF $T[t]$ is <i>normal</i> AND $P[t]$ is <i>strong</i>	THEN $I[t]$ is <i>negative small</i>
IF $T[t]$ is <i>warm</i> AND $P[t]$ is <i>strong</i>	THEN $I[t]$ is <i>negative medium</i>
IF $T[t]$ is <i>hot</i> AND $P[t]$ is <i>strong</i>	THEN $I[t]$ is <i>negative large</i>
IF $T[t]$ is <i>cold</i> AND $P[t]$ is <i>high</i>	THEN $I[t]$ is <i>negative medium</i>
IF $T[t]$ is <i>cool</i> AND $P[t]$ is <i>high</i>	THEN $I[t]$ is <i>negative medium</i>
IF $T[t]$ is <i>normal</i> AND $P[t]$ is <i>high</i>	THEN $I[t]$ is <i>negative medium</i>
IF $T[t]$ is <i>warm</i> AND $P[t]$ is <i>high</i>	THEN $I[t]$ is <i>negative large</i>
IF $T[t]$ is <i>hot</i> AND $P[t]$ is <i>high</i>	THEN $I[t]$ is <i>negative large</i>

จากกฎทั้ง 25 กฎของระบบนี้ สามารถนำมาเขียนเป็นเมตริกซ์ที่แสดงถึงความสัมพันธ์ร่วมกันของตัวแปรได้ดังรูปที่ 2.9 ซึ่งเมตริกซ์จะถูกเรียกว่า Fuzzy Associative Memory (FAM) [37] FAM จะมีมิติเท่ากับจำนวนตัวแปรของอินพุต ซึ่งในตัวอย่างนี้มีสองตัวแปรดังนั้น FAM จะมีเมตริกซ์เป็นสองมิติ ขนาด $M \times N$ เมื่อ M แทนจำนวนเทอมเซตของตัวแปรอุณหภูมิ และ N เป็นจำนวนเทอมเซตของตัวแปรความดัน และสมาชิกของเมตริกซ์นี้คือค่าของตัวแปรเอาต์พุตทั้ง 7 เทอม ส่วนในกรณีอื่นๆ เช่นถ้าตัวแปรของอินพุตมี 3 ตัว FAM จะมี 3 มิติและขนาดของตาราง 3 มิตินี้จะเป็น $M \times N \times P$ เมื่อ M , N , และ P เป็นจำนวนเทอมเซตของแต่ละตัวแปร เป็นต้น

อุณหภูมิ / ความดัน	COLD	COOL	NORMAL	WARM	HOT
WEAK	PL	PL	PM	PM	PS
LOW	PM	PM	PS	PS	PS
OK	PS	ZR	ZR	NS	MS
STRONG	NS	NM	NS	NM	NL
HIGH	NM	NM	NM	NL	NL

รูปที่ 2.9 แสดงเมตริกซ์ FAM ของระบบควบคุม STREAM TURBINE

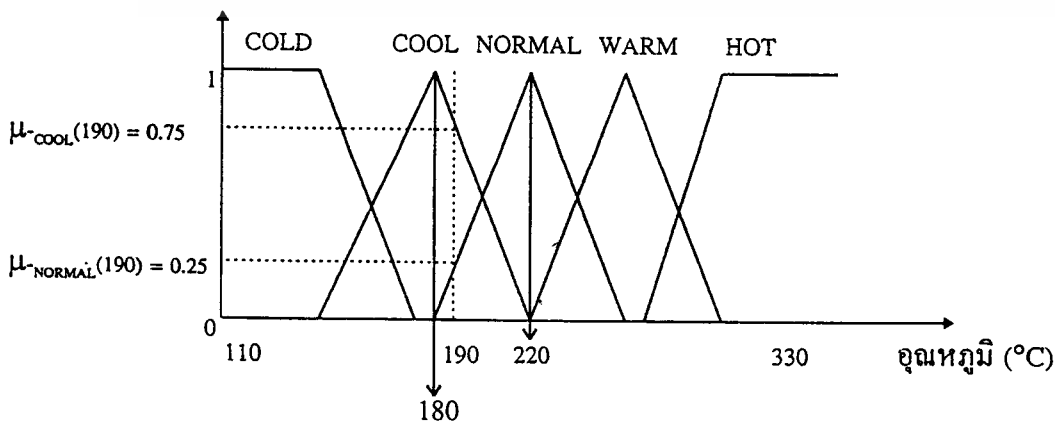
2.5.4 Defuzzify

Defuzzification เป็นกระบวนการประมาณค่าของตัวแปรเอาต์พุต โดยการอนุมานจากค่าความจริงในส่วนของ Antecedent ของแบบจำลองฟัซซี่ลอจิก โดยค่าที่ได้จะเป็นเลขจำนวนจริงที่อยู่ในโดเมนของตัวแปรเอาต์พุต และค่านี้จะเป็นค่าคาดหวัง (Expected Value) ของตัวแปรฟัซซี่เซต ที่จะนำไปใช้ในการควบคุมระบบต่อไป สำหรับการหาค่าคาดหวังของฟัซซี่เซตโดยการ Defuzzification นั้นมีอยู่หลายวิธีด้วยกันคือ

2.5.4.1 กำหนดจากค่าความจริงที่มีค่าต่ำสุด (เฉพาะเทอมเซตที่เป็นสมาชิกเท่านั้น, $\mu_A(x_i) > 0$ เมื่อ A เป็นเทอมเซตใดๆ) ของ Antecedent จากตัวอย่างแบบจำลองฟัซซี่ลอจิกที่ใช้กับระบบ Stream Turbine ในขั้นตอนของกฎฟัซซี่สามารถประมาณค่า I ที่เวลา t ใดๆ ได้ดังนี้

สมมุติอุณหภูมิที่วัดได้ที่เวลา t เท่ากับ $190\text{ }^{\circ}\text{C}$ และความดันที่เวลาเดียวกับที่วัดอุณหภูมิเท่ากับ 100 KgM^{-2} จากกฎฟัซซี่ที่เทอมเซต COOL และ NORMAL ที่ถูกนำมาพิจารณา ดังนั้นเราสามารถประมาณค่าระดับการเปิดปิดวาล์วได้ดังนี้ดังรูปที่ 2.10

ระดับความเป็นสมาชิก



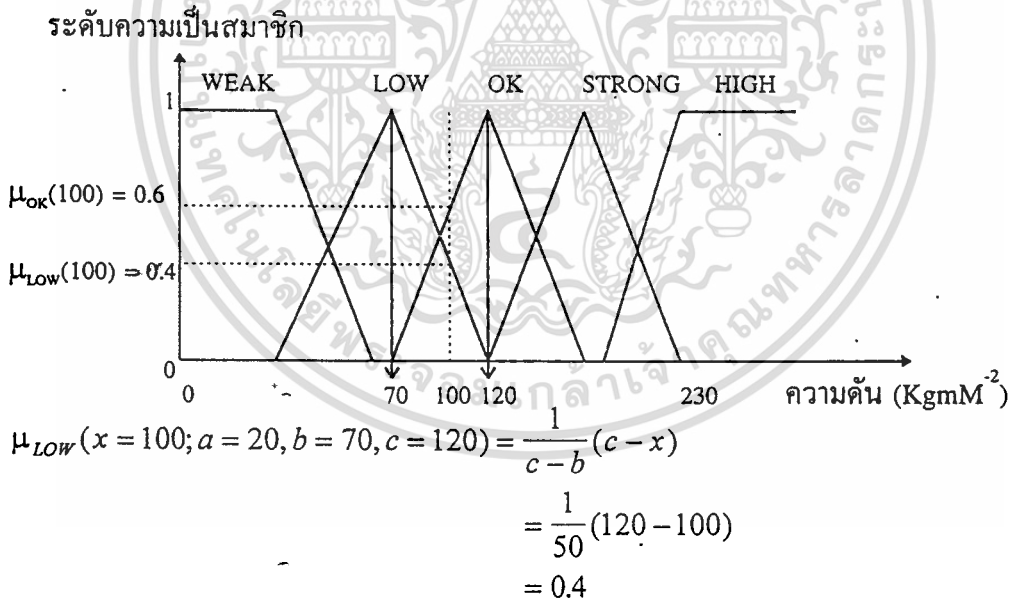
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{aligned}\mu_{COOL}(x=190; a=135, b=180, c=220) &= \frac{1}{c-b}(c-x) \\ &= \frac{1}{40}(220-190) \\ &= 0.75\end{aligned}$$

$$\begin{aligned}\mu_{NORMAL}(x=190; a=180, b=220, c=260) &= \frac{1}{b-a}(x-a) \\ &= \frac{1}{40}(190-180) \\ &= 0.25\end{aligned}$$

รูปที่ 2.10 แสดงการคำนวณค่าความเป็นสมาชิกของเทอมเซต COOL และ NORMAL

ที่อุณหภูมิ 190 °C ไม่ได้เป็นสมาชิกของเทอมเซต COLD, WARM, และ HOT สำหรับความดันที่ 100 KgmM⁻² นั้นจากกฎฟัซซี่จะมีเทอมเซต LOW และ OK ที่ถูกนำมาพิจารณาดังรูปที่ 2.11



$$\begin{aligned}\mu_{OK}(x=100; a=70, b=120, c=170) &= \frac{1}{b-a}(x-a) \\ &= \frac{1}{50}(100-70) \\ &= 0.6\end{aligned}$$

รูปที่ 2.11 แสดงการคำนวณค่าความเป็นสมาชิกของเทอมเซต LOW และ OK

เอกสารนี้เป็นเอกสารที่ลงวันเวลาสำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ความดัน 100 KgmM^{-2} ไม่เป็นสมาชิกของเทอมเซต WEAK, STRONG, และ HIGH ดังนั้นที่ $T[t] = 190$ °C และ $P[t] = 100 \text{ KgmM}^{-2}$ จะส่งผลให้เทอมเซต PM, ZR, และ PS เป็นเทอมที่ใช้ประมาณค่า $I[t]$ พิจารณามเมตริกซ์ FAM ในรูปที่ 2.9 กฎฟuzzyที่เป็นจริงมีอยู่ 4 กฎด้วยกันดังแสดงในรูปที่ 2.12 (กฎที่เป็นจริงจะถูกล้อมกรอบด้วยสี่ดำเข้ม)

อุณหภูมิ ความดัน	COLD	COOL	NORMAL	WARM	HOT
WEAK	PL	PL	PM	PM	PS
LOW	PM	PM	PS	PS	PS
OK	PS	ZR	ZR	NS	MS
STRONG	NS	NM	NS	NM	NL
HIGH	NM	NM	NM	NL	NL

รูปที่ 2.12 แสดงกฎที่เป็นจริงของ FAM เมื่อ $T[t] = 190$ °C และ $P[t] = 100 \text{ KgmM}^{-2}$

เราสามารถประมาณค่าตัวแปรเอาต์พุต $I[t]$ ด้วยค่าความจริงที่มีค่าต่ำสุดได้ดังรูปที่ 2.13

2.5.4.2 กำหนดจากค่าความจริงที่มีค่าสูงสุดของ Antecedent จากขั้นตอนการประมาณค่าตัวแปรเอาต์พุตในรูปที่ 2.13 จะมีความแตกต่างจากการ Defuzzification โดยใช้ค่าความจริงที่มีค่าต่ำสุดในหัวข้อ 4.1 เฉพาะขั้นตอนการคำนวณค่า $\mu_{\text{INFERENCE}}$ เท่านั้นซึ่งสามารถคำนวณได้ดังนี้

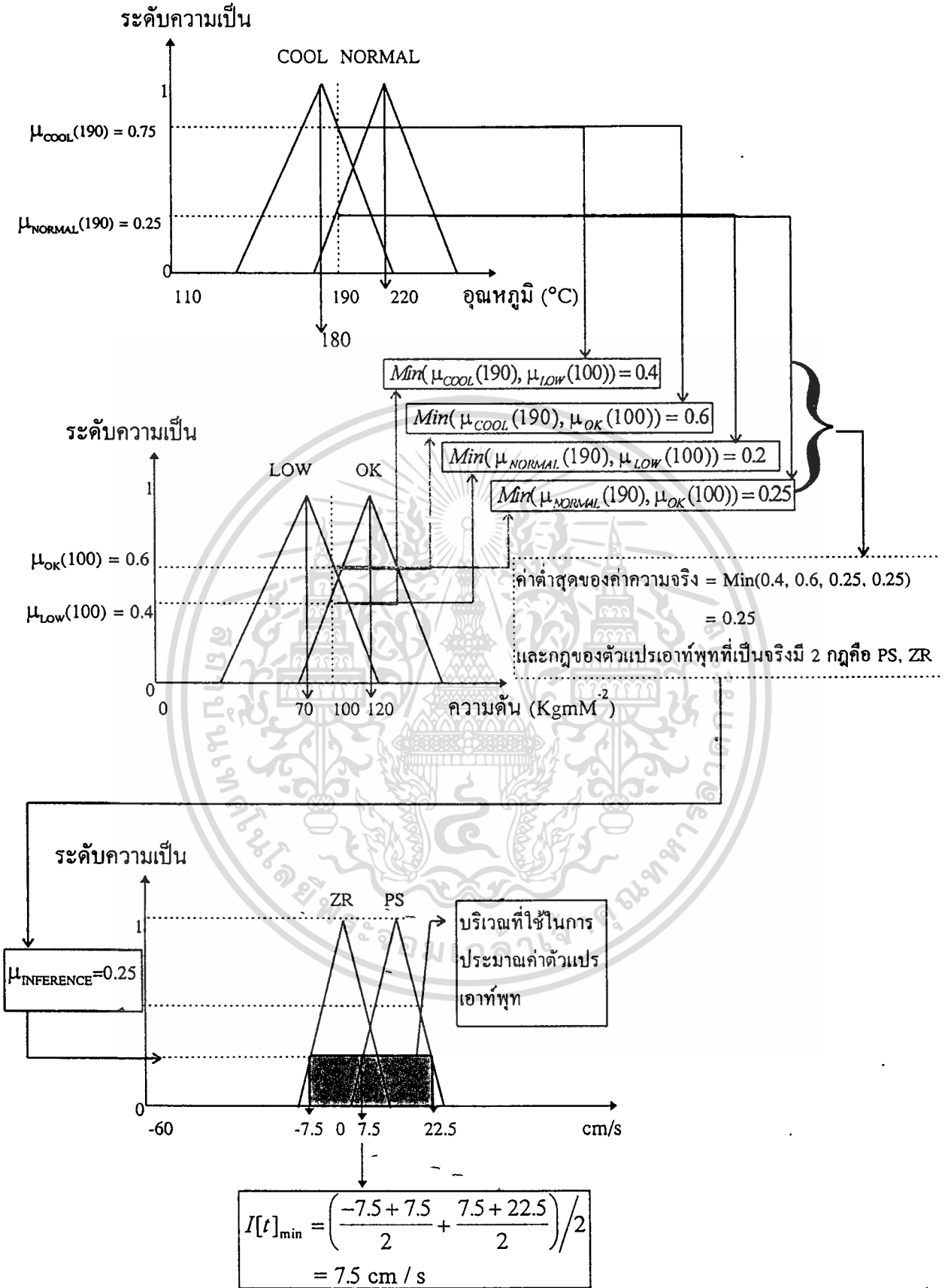
$$\begin{aligned}\mu_{\text{INFERENCE}} &= \text{Max}(0.4, 0.6, 0.25, 0.25) \\ &= 0.6\end{aligned}$$

กฎที่ใช้ในการประมาณค่า $I[t]$ ในกรณีการ Defuzzification โดยวิธีนี้คือ

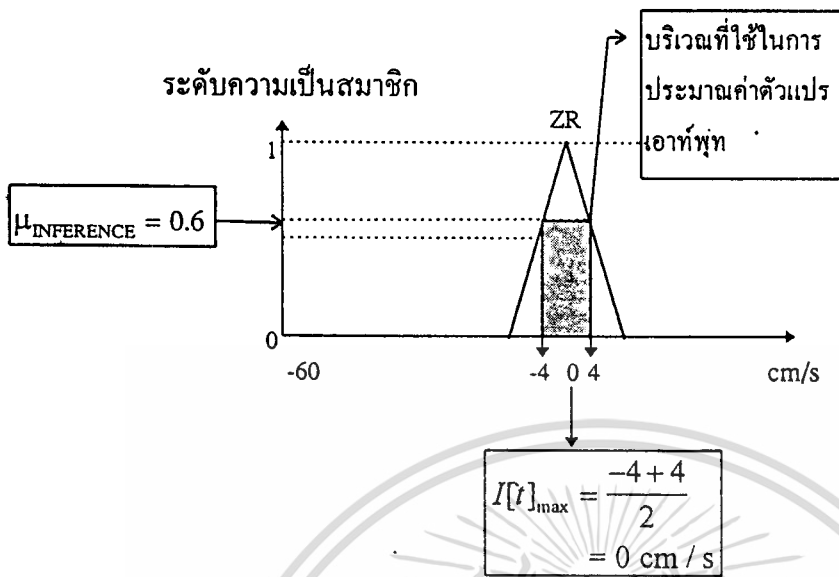
IF $T[t]$ is cool AND $P[t]$ is ok THEN $I[t]$ is approximately zero

เนื่องจากกฎข้อนี้ให้ค่าความจริงที่มีค่าสูงสุด $\mu_{\text{INFERENCE}} = 0.6$ ดังนั้นสามารถคำนวณค่าประมาณของ $I[t]$ ได้ดังรูปที่ 2.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารรูปที่ 2.13 แสดงขั้นตอนการประมาณค่า $I[t]$ โดยคำนวณจากค่าความจริงที่มีค่าต่ำสุดขั้นตอนการคำนวณว่ากรณีใดข้างต้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 แสดงการ Defuzzification โดยใช้ค่าสูงสุดของค่าอัตราความเป็นสมาชิก

2.5.4.3 คำนวณจากค่าเฉลี่ยของกฎที่มีค่าความจริงสูงสุดหรือที่เรียกว่าวิธี Mean of Maxima : MOM วิธีนี้คล้ายกับวิธีที่ใช้ค่าระดับความเป็นสมาชิกที่มีค่าสูงสุดในหัวข้อ 4.2 เพียงแต่เพิ่มค่าระดับความเป็นสมาชิกของกฎเดียวกันเป็นตัวหารซึ่งจากตัวอย่างของการควบคุมระบบ Stream Turbine สามารถคำนวณได้ดังนี้

$$I[t]_{MOM} = \frac{-4 + 4}{0.6 + 0.6} = 0 \text{ cm / s}$$

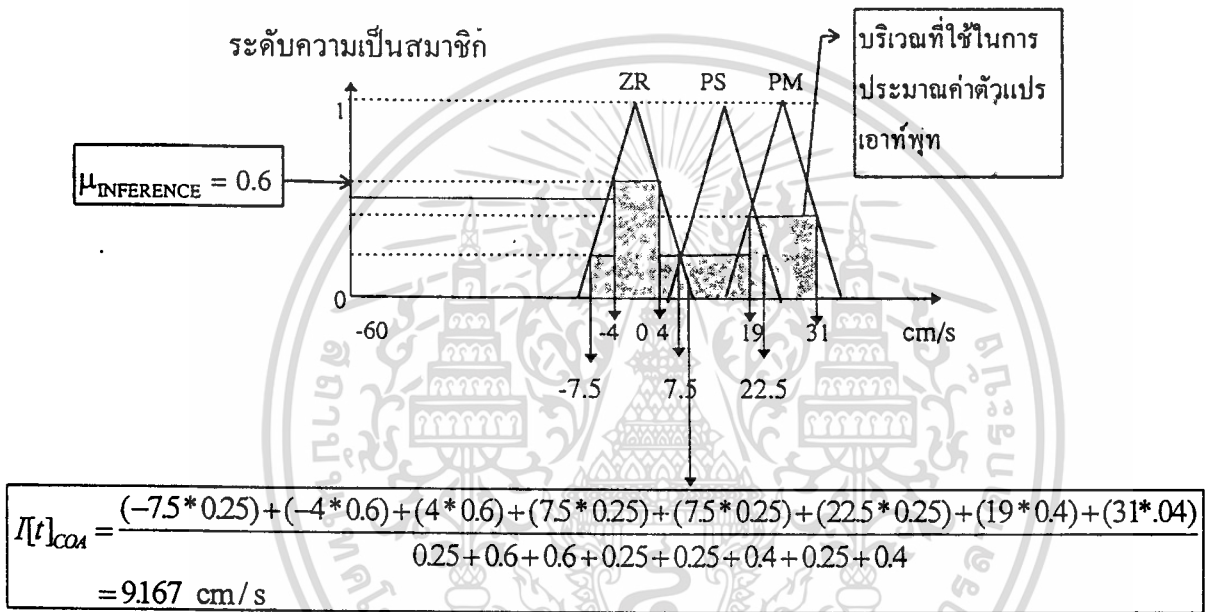
2.5.4.4 การหาค่าจุดศูนย์กลางถ่วง (Centroid หรือ Center of Area : COA) [38-39] เป็นการประมาณค่าของตัวแปรเอาท์พุทโดยการหาค่าจุดศูนย์กลางของพื้นที่ของกฎที่เป็นจริง ซึ่งลักษณะการคำนวณจะมีความแตกต่างจากสองวิธีแรก คือแทนที่จะประมาณค่าตัวแปรของเอาท์พุทโดยใช้ค่าความจริงของกฎที่มีค่าความเป็นสมาชิกต่ำสุดหรือสูงสุด ก็จะใช้ค่าความจริงของกฎทุกข้อที่มีค่าระดับความเป็นสมาชิกมากกว่าศูนย์ ซึ่งในกรณีนี้กฎทุกข้อที่ถูกล้อมกรอบด้วยเส้นทึบของตาราง FAM ในรูปที่ 2.12 ในตัวอย่างของระบบควบคุม Stream Turbine จะถูกนำมาประมาณค่าตัวแปร $I[t]$ ทุกกฎ ซึ่งสามารถดำเนินการได้ดังนี้

จากขั้นตอนการประมาณค่าตัวแปรเอาท์พุทในรูปที่ 2.13 กฎที่ใช้ในการประมาณค่า $I[t]$ ในกรณีการ Defuzzification โดยวิธีนี้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 IF $T[t]$ is cool AND $P[t]$ is low THEN $I[t]$ is positive medium
 ไม่ว่าจะกรณีใดทั้งสน ออกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IF $T[t]$ is cool AND $P[t]$ is ok THEN $I[t]$ is approximately zero
 IF $T[t]$ is normal AND $P[t]$ is low THEN $I[t]$ is positive small
 IF $T[t]$ is normal AND $P[t]$ is ok THEN $I[t]$ is approximately zero

ซึ่งมีค่า $\mu_{\text{INFERENCE}}$ จะมี 4 ค่าด้วยกันคือ $\mu_{\text{INFERENCE}} = \{0.4, 0.6, 0.25, 0.25\}$ ดังนั้นสามารถคำนวณค่าประมาณของ $I[t]$ ได้ดังรูปที่ 2.15



รูปที่ 2.15 แสดงการ Defuzzification โดยวิธี COA

จากวิธีการ defuzzification ที่ได้กล่าวมาแล้วทั้ง 4 วิธี วิธี COA เป็นวิธีที่นิยมใช้ในกระบวนการควบคุมมากกว่าวิธีอื่นๆ เนื่องจากวิธีนี้ใช้ข้อมูลข่าวสารของค่าความจริงของทุกกฎที่เป็นจริง ($\mu_A(x) > 0$) ทำให้ค่าที่ได้มีความน่าเชื่อถือและเหมาะสมมากกว่าวิธีอื่นๆ ที่ได้กล่าวมาแล้ว นอกจากนี้ยังมีวิธีอื่นๆ อีกที่ค่าตัวแปรบางตัวสามารถปรับค่าได้เช่น วิธี Basic Defuzzification Distribution : BADD [40] และวิธี Semi Linear Defuzzification : SLIDE [41] เป็นต้น

2.8 ความคลุมเครือ

ความคลุมเครือหรือที่เรียกว่า Fuzziness เป็นระดับหรือปริมาณที่ไม่สามารถกำหนดขอบเขตให้กับตัวแปรฟัซซี่เซตได้อย่างแน่นอน (บางครั้งอาจจะเรียกว่า Vagueness) ซึ่งค่าของตัวเหล่านี้จะมีความคลุมเครืออยู่แล้วโดยธรรมชาติ เนื่องจากเราไม่สามารถกำหนดขอบเขตให้กับตัวแปรบางตัวได้อย่างแน่นอน เช่น

เมื่อเราวัดอุณหภูมิ (ไม่ว่าจะเป็นอุณหภูมิของระบบใด) เราไม่สามารถกำหนดขอบเขตได้อย่างแน่นอนว่า อุณหภูมิเท่าไรจึงจะเรียกว่าร้อน และอุณหภูมิกี่องศาจึงจะเหมาะสมกับระบบนั้นๆ ในทางปฏิบัติได้ทำการประมาณตัวแปรของข้อมูลโดยการแบ่งข้อมูลออกเป็นเซตย่อยหรือเทอมเซต เพื่อหาค่าถ่วงน้ำหนัก (ระดับความเป็นสมาชิก) ว่าเทอมใดมีโอกาสถูกต้องสูงสุด แต่อย่างไรก็ตามเทอมเซตต่างๆ ที่สร้างขึ้นก็เป็นการคาดคะเนจากประสบการณ์หรือการทดลองหรืออาจจะอาศัยข้อมูลจากผู้เชี่ยวชาญ ดังนั้นจึงมีการวัดความคลุมเครือของฟัซซีเซตโดยใช้ค่าดัชนีความคลุมเครือ (Index of Fuzziness : H) ซึ่งค่าของดัชนีนี้จะบอกได้ว่าเทอมเซตใดเป็นเทอมเซตที่ถูกออกแบบได้อย่างเหมาะสม คือถ้าเทอมเซตใดสามารถออกแบบได้ครอบคลุมข้อมูลทั้งหมดได้อย่างเหมาะสมแล้ว ค่าความเป็นสมาชิกจะเข้าใกล้ศูนย์ (ในกรณีที่ไม่ได้เป็นสมาชิก) และเข้าใกล้หนึ่ง (ในกรณีที่เป็นสมาชิก) ซึ่งถ้าเป็นดังนี้ค่าดัชนีความคลุมเครือจะมีค่าต่ำมาก ($H \rightarrow 0$) ในทางตรงกันข้ามถ้าเทอมเซตใดไม่สามารถระบุได้อย่างแน่ชัดว่าเป็นสมาชิกหรือไม่ ($\mu_A(x) \rightarrow 0.5$) ค่าดัชนีความคลุมเครือของเทอมเซตนั้นจะมีค่ามาก ($H \rightarrow \infty$) Ebanks [42] ได้แนะนำไว้ว่าคุณสมบัติที่ดีของตัววัดดัชนีความคลุมเครือควรมีคุณสมบัติดังนี้

เมื่อ A, B เป็นเทอมเซตใดๆ ใน U

1. Sharpness ในกรณีที้ออกแบบเทอมเซต A ได้อย่างเหมาะสม

$$H(A) = 0 \Leftrightarrow \mu_A(x) = 0 \text{ OR } 1 \quad \forall x \in U$$

2. Maximality เมื่อกำหนดเทอมเซต A ไม่เหมาะสมกับข้อมูลชุดนั้นๆ จะทำให้

$$H(A) \rightarrow \infty \Leftrightarrow \mu_A(x) = 0.5 \quad \forall x \in U$$

3. Resolution ถ้ามีเทอมเซต A' เป็นเทอมเซตที่มีคุณสมบัติ Sharpness ของเทอมเซต A จะได้ว่า

$$H(A) \geq H(A')$$

4. Symmetry

$$H(A) = H(1 - \bar{A})$$

$$\text{โดยที่ } \mu_{1-\bar{A}}(x) = 1 - \mu_A(x) \quad \forall x \in U$$

5. Valuation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$H(A \cup B) + H(A \cap B) = H(A) + H(B)$$

ในปี ค.ศ. 1968 Zadeh [43] ได้เสนอตัววัดความคลุมเครือไว้ดังนี้

$$H_{ZE}(A, P) = -\sum_{i=1}^n \mu_i p_i \log p_i \quad (2.20)$$

โดยที่ $\mu_i \in A$ ซึ่งเป็นเทอมเซต และ $p_i \in P$ เป็นเซตของความน่าจะเป็นที่ใช้ข้อมูลชุดเดียวกับเทอมเซต A เนื่องจากวิธีวัดความคลุมเครือของ Zadeh ไม่เป็นไปตามคุณสมบัติที่ Ebanks ได้ตั้งไว้ทั้ง 5 ข้อ วิธีวัดความคลุมเครือที่มีคุณสมบัติของ Ebanks ครบทั้ง 5 ข้อถูกเสนอครั้งแรกโดย De Luca และ Termini [42] ซึ่งมีสมการดังนี้

$$H_{DTE}(A) = -K \sum_{i=1}^n \mu_i \log \mu_i + (1 - \mu_i) \log(1 - \mu_i) \quad (2.21)$$

ที่ K เป็นค่าคงที่ที่ใช้ในการ Normalization

สำหรับตัวดำเนินการวัดความคลุมเครือวิธีอื่นๆ ที่มีคุณสมบัติครบทั้ง 5 ข้อ [42] คือวิธีของ Ebanks [44], Pal และ Pal [45], Bhandari และ Pal [46], และวิธีของ Pal และ Bezdeh [47-48] และวิธีอื่นๆ ที่มีคุณสมบัติบางข้อขาดไป [42] เช่น วิธีของ Yager [49], และ Kaufmann [33] เป็นต้น

สรุป

ฟัซซี (Fuzzy) ตามรูปศัพท์มีอยู่สองความหมายคือ เป็นผอย, ขนปุย “Like or having Fuzz” ส่วนอีกความหมายหนึ่งคือ เบลูเอเลือน, คลุมเครือ, ไม่ชัดเจน “Blurred or Unclear ideas” ซึ่งฟัซซีเซตในที่นี้จะตรงกับความหมายที่สองคือเป็นเซตที่มีความคลุมเครือ (Fuzziness) คำว่า “คลุมเครือ” ในที่นี้หมายถึงความไม่ชัดเจนของตัวแปรฟัซซีที่เราไม่สามารถกำหนดขอบเขตให้กับตัวแปรได้อย่างแน่นอน เช่น ในประโยคที่ว่า “วัยกลางคน” ซึ่งมีตัวแปรฟัซซีคือ อายุ จะเห็นได้ว่าเราไม่สามารถกำหนดขอบเขตได้ชัดเจนลงไปว่า ช่วงอายุเท่าไรจึงจะเป็นวัยกลางคน แต่ในทางฟัซซีเซตสามารถแทนตัวแปรอายุจากประโยค “วัยกลางคน” (เป็นเทอมเซต) ได้โดยใช้แบบจำลองทางคณิตศาสตร์ที่เรียกว่า “ฟังก์ชันความเป็นสมาชิก” ซึ่งเป็นแนวความคิดที่ต่างจากเซตธรรมดา (Crisp Set) ก็ตรงฟังก์ชันความเป็นสมาชิกนี้เอง คือฟัซซีเซตจะมีแนวคิดของ “บางส่วนอาจจะเป็นสมาชิกของเซต” แต่ในกรณีของเซตธรรมดาจะบอกว่า “เป็นหรือไม่เป็น” สมาชิกเท่านั้น ซึ่งตรงจุดนี้เองที่ทำให้ฟัซซีเซตมีความยืดหยุ่นกว่าเซตธรรมดา

ฟัซซีเซตสามารถนำมาประยุกต์ใช้งานได้จริง จะเห็นได้จากปัจจุบันมีสิ่งอำนวยความสะดวกหลายชนิดที่ใช้ฟัซซีลอจิกเป็นตัวควบคุม เช่น เครื่องปรับอากาศ, เครื่องซักผ้า, ลิฟท์, เตาไมโครเวฟ, กล้องถ่ายรูป

ไม่ว่าการณ์ใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นต้น ซึ่งผู้ที่นำฟuzzyลอจิกมาประยุกต์ใช้เป็นคนแรกคือ Mamdani [50] และได้มีการพัฒนาต่อมาจนเป็นระบบฟuzzy (Fuzzy Systems) ที่ใช้กันอยู่ในปัจจุบัน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3 การปรับปรุงภาพ

วิธีการปรับปรุงภาพจะประกอบด้วยเทคนิคต่างๆ เพื่อที่จะปรับปรุงภาพ หรือทำการแปลงภาพให้อยู่ในรูปแบบที่ดีกว่าเพื่อทำการวิเคราะห์โดยมนุษย์หรือคอมพิวเตอร์ โดยเฉพาะในงานทางด้านการแพทย์ การแยกวัตถุภายในภาพ และการจดจำวัตถุภายในภาพ สำหรับประโยชน์ที่ได้รับจากการปรับปรุงภาพในการวิเคราะห์ภาพด้วยคอมพิวเตอร์ คือการกำจัดสัญญาณรบกวน, การเน้นขอบภาพ, การปรับภาพเพื่อให้เห็นความแตกต่างของวัตถุภายในภาพ จะทำให้คอมพิวเตอร์สามารถดึงลักษณะต่างๆ ของวัตถุภายในภาพ (Feature Extraction) ได้ดีขึ้น

3.1 การประมวลผลภาพแบบจุด

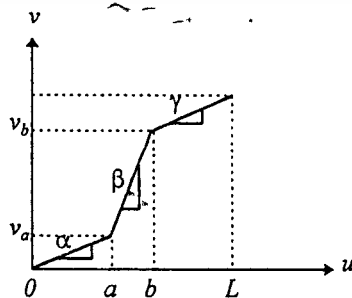
การประมวลผลภาพแบบจุด [8, 21] เป็นวิธีการปรับปรุงภาพแบบหนึ่ง ที่ใช้วิธีการแปลงข้อมูลภาพระดับสีเทา (Gray Level) จากโดเมน $u \in [0, L]$ ไปเป็น $v \in [0, L]$ โดยใช้ฟังก์ชันในการแปลงดังสมการที่ (3.1)

$$v = f(u) \quad (3.1)$$

โดยที่ u และ v เป็นค่าของจุดภาพ สำหรับการแปลงข้อมูลภาพแบบต่างๆ มีวิธีการดังนี้

3.1.1 การยืดภาพให้คมชัด

ภาพที่ใช้โดยทั่วไปจะมีค่าอยู่ในช่วง 0-255 (ใช้ 8 บิตสำหรับเก็บข้อมูล) แต่บางครั้งข้อมูลที่ได้มามีคอนทราสต์อยู่ในช่วงที่ค่อนข้างต่ำ การกระจายของข้อมูลอยู่ในช่วงที่แคบทำให้ภาพไม่คมชัด ดังนั้นการยืดภาพจะช่วยเน้นข้อมูลภาพ ในช่วงที่เราต้องการจะเน้นเพื่อให้เห็นความแตกต่างระหว่างระดับสีเทาให้มากขึ้น ในรูปที่ 3.1 แสดงช่วงต่างๆ ที่ต้องการจะยืด



รูปที่ 3.1 แสดงลักษณะของฟังก์ชันที่ใช้ในการยืดภาพให้คมชัดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า และจากรูปที่ 3.1 สามารถนำมาเขียนเป็นสมการได้ดังนี้ ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$v = \begin{cases} \alpha u, & 0 \leq u < a \\ \beta (u - a) + v_a, & a \leq u < b \\ \gamma (u - b) + v_b, & b \leq u < L \end{cases} \quad (3.2)$$

ที่ α , β , และ γ เป็นค่าความชันของแต่ละช่วง ถ้าต้องการยืดข้อมูลภาพในช่วงใดก็ให้ค่าความชันในช่วงนั้นมีค่ามากกว่าหนึ่ง สำหรับ a และ b เป็นค่าขอบเขตของแต่ละช่วง โดยที่ $a \approx \frac{L}{3}$ และ $b \approx \frac{2L}{3}$ เมื่อ L เป็นค่าคงที่ที่เท่ากับ 255 ซึ่งเป็นค่าสูงสุดของภาพ 8 บิต ในการปรับปรุงภาพโดยวิธีนี้จะต้องคำนวณค่าความเข้มสูงสุดของภาพที่ต้องการปรับปรุงก่อนเพื่อใช้ในการคำนวณค่าของตัวแปร a , b , α , β , และ γ ดังอัลกอริทึมที่ 3-1.

อัลกอริทึมที่ 3-1. แสดงขั้นตอนการยืดภาพให้คมชัด

```
void Contrast_Stretch(IMAGE I)
/* IMAGE เป็นตัวแปรชนิด struct ซึ่งสามารถกำหนดได้ดังนี้
* typedef struct
* {
*     BYTE **data; // เป็นการจองหน่วยความจำไว้เก็บข้อมูลภาพซึ่งเป็นเมตริกซ์-
*                  // ขนาด ROWS x COLS
*     int ROWS, COLS; // ขนาดของภาพ ROWS x COLS
*     int MIN, MAX; // ค่าความเข้มต่ำสุด (MIN) และ สูงสุดของภาพ (MAX)
* }IMAGE;
*/
{
    BYTE a, b; // ตัวแปร a, b ในสมการที่ (3.2)
    BYTE va, vb, L=255;
    float alpha, beta, gamma; // ตัวแปร  $\alpha$ ,  $\beta$  และ  $\gamma$  ในสมการที่ (3.2)

    a = I.MAX/3;
    b = (2*I.MAX)/3;
    va = L/3;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

alpha = va/(float)(a);
beta = (vb-va)/(float)(b-a);
gamma = (L-vb)/(I.MAX-b);
for(int r=0; r<I.ROWS; r++)
    for(int c=0; c<I.COLS; c++)
    {
        BYTE m=I.data[r][c];

        if(m<a)
            I.data[r][c] = (BYTE)(alpha*m);
        if((m>=a) && (m<b))
            I.data[r][c] = (BYTE)(beta*(m-a)+va);
        if((m>=b) && (m<L))
            I.data[r][c] = (BYTE)(gamma*(m-b)+vb);
    }
} /* สิ้นสุดอัลกอริทึมการขยายช่วงของข้อมูลภาพ */

```

จากอัลกอริทึมที่ 3-1. ได้นำมาปรับปรุงภาพดังแสดงในรูปที่ 3.2 ซึ่งเป็นภาพเด็กที่มีขนาด 256 x 256 (ROWS = 256, COLS = 256) และค่าสูงสุดเท่ากับ (MAX = 75), ค่าต่ำสุดเท่ากับ (MIN = 12)



(ก)



(ข)

รูปที่ 3.2 การปรับปรุงภาพโดยวิธีการยืดช่วงของภาพ (ก) อินพุท (ข) ผลลัพธ์ที่ได้จากอัลกอริทึมที่ 3-1.

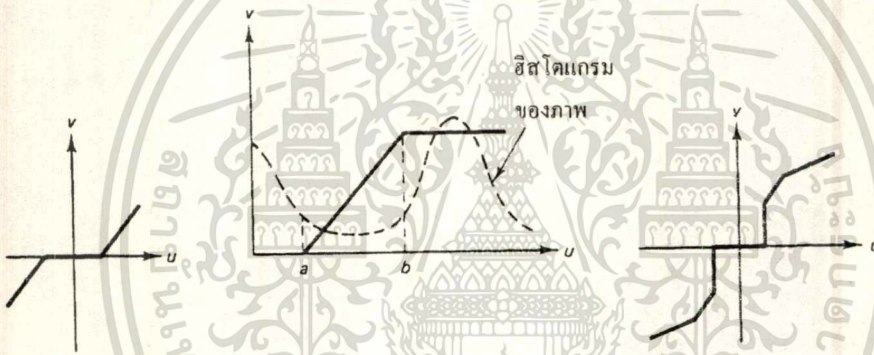
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 การตัดสัญญาณรบกวน และเทรซโฮลด์

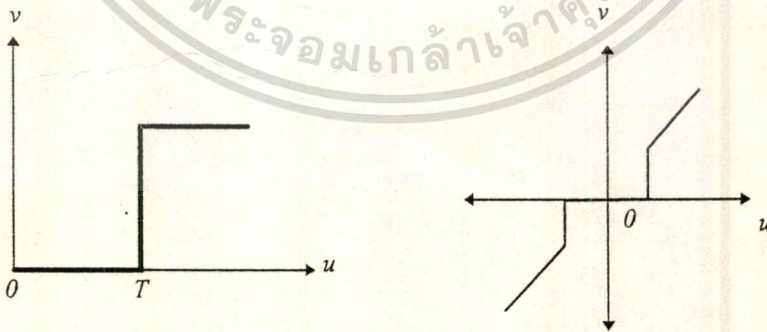
วิธีตัดสัญญาณรบกวนเป็นกรณีเฉพาะของการบีดภาพให้คมชัด โดยที่ค่าความชัน $\alpha = \gamma = 0$ (ดูกราฟในรูปที่ 3.3) วิธีนี้ใช้กำจัดสัญญาณรบกวนเมื่อเราทราบช่วงที่มีสัญญาณรบกวนที่อยู่นอกช่วง $[a, b]$ ซึ่งสามารถเขียนเป็นสมการได้ดังนี้

$$v = \begin{cases} 0, & 0 \leq u < a \\ \beta u, & a \leq u \leq b \\ L, & u > b \end{cases} \quad (3.3)$$

การทำภาพให้เป็นภาพไบนารีโดยใช้ค่าพิคคเป็นกรณีเฉพาะของสมการนี้โดย $a = b = t$ (ดูกราฟรูปที่ 3.4)



รูปที่ 3.3 การแปลงข้อมูลภาพโดยการตัดสัญญาณรบกวน



รูปที่ 3.4 การแปลงข้อมูลภาพระดับสีเทาไปเป็นภาพไบนารี

ในการกำจัดช่วงที่มีสัญญาณรบกวนออกไปจากภาพ เราต้องกำหนดช่วงของค่าความเข้มที่ต้องการจะเน้น นั่นคือช่วง $[a, b]$ และช่วงที่ต้องการกำจัด (ค่าความเข้มที่อยู่นอกช่วง $[a, b]$) การเน้นข้อมูลในช่วง $[a, b]$ ให้สว่างขึ้นหรือมืดลงขึ้นอยู่กับค่าที่กำหนดตัวแปร β ซึ่งมีคุณสมบัติเช่นเดียวกับ β ในสมการที่ (3.2) ซึ่งขั้นตอนการประมวลผลโดยวิธีนี้ได้แสดงไว้ในอัลกอริทึมที่ 3-2.
 เมื่อกำหนดค่าต่างๆแล้ว ออกพิมพ์ออกมาให้ชัดเจน และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อัลกอริทึมที่ 3-2. เป็นขั้นตอนของการปรับปรุงภาพโดยวิธีการตัดสัญญาณรบกวน

```

void Clipping(IMAGE I)
{
    int a, b; // ช่วงของข้อมูลที่ใช้ในการกำจัดสัญญาณรบกวน
    BYTE L=255;
    float beta; // ค่า  $\beta$  ที่ใช้เน้นข้อมูลภาพในช่วงที่ต้องการ

    printf("Maximum Pixel Value := %d", I.MAX);
    printf("Minimum Pixel Value := %d", I.MIN);

    printf("Lower Bound := ");
    scanf("%d", &a);
    printf("Upper Bound := ");
    scanf("%d", &b);
    printf("The Slop Beta := ");
    scanf("%f", &beta);

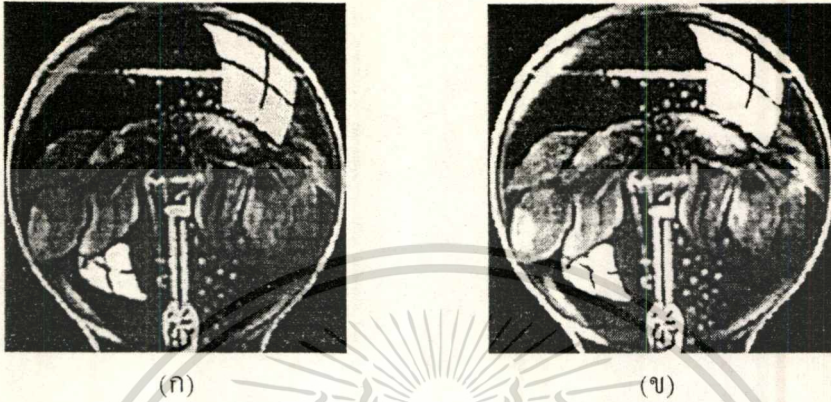
    for(int r=0; r<I.ROWS; r++)
        for(int c=0; c<I.COLS; c++)
        {
            BYTE m=I.data[r][c];

            if(m<a)
                I.data[r][c] = 0;
            if((m>=a) && (m<b))
                I.data[r][c] = (BYTE)(beta*m);
            if((m>=b) && (m<L))
                I.data[r][c] = L;
        }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

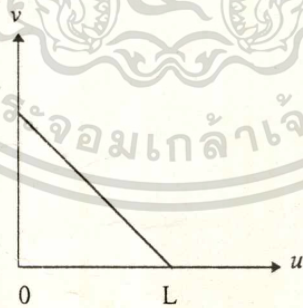
ในรูปที่ 3.5 แสดงผลของการปรับปรุงภาพโดยวิธีการตัดสัญญาณรบกวนและการเทรซโฮลด์ข้อมูลภาพ โดยภาพที่ใช้เป็นภาพพลาการ์ดที่ขนาดของภาพเท่ากับ 256×256 จุด ค่าความเข้มสูงสุดเท่ากับ 255 และค่าความเข้มต่ำสุดเท่ากับ 0



รูปที่ 3.5 ผลของการปรับปรุงภาพโดยวิธีการตัดสัญญาณรบกวน (ก) อินพุท (ข) ผลลัพธ์ที่ได้จากการประมวลผลโดยใช้อัลกอริทึมที่ 3-2. ที่ $a = 40$, $b = 180$, และ $\beta = 1.4$

3.1.3 การกลับข้อมูลภาพ

การกลับภาพเชิงตัวเลขหรือการอินเวิร์ตข้อมูลภาพ (Digital Negative) สามารถทำได้โดยการคอมพลิเมนต์ภาพระดับสีเทาด้วยการแปลงข้อมูลในรูปที่ 3.6



รูปที่ 3.6 ลักษณะของฟังก์ชันการแปลงแบบ Digital Negative

สำหรับสมการในการแปลงสามารถเขียนได้โดย

$$v = L - u \quad (3.4)$$

อัลกอริทึมที่ 3-3. แสดงขั้นตอนของการกลับข้อมูลภาพ

เอกสารนี้เป็นเอกสารทวงเงินไวสาหรับการเขงานเพื่อการศึกษาเท่านั้น ไม่นุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void Reversal_Digital_Image(IMAGE I)
{
    for(int r=0; r<I.ROWS; r++)
        for(int c=0; c<I.COLS; c++)
            I.data[r][c] = L - I.data[r][c];
}

```

ในรูปที่ 3.7 แสดงผลของการกลับข้อมูลภาพ ซึ่งได้จากการประมวลผลโดยอัลกอริทึมที่ 3-3. ภาพที่ใช้เป็นภาพผู้หญิงซึ่งขนาดของภาพเท่ากับ 256 x 256 จุด ค่าความเข้มต่ำสุดของภาพนี้เท่ากับ 0 และค่าความเข้มสูงสุดเท่ากับ 255



รูปที่ 3.7 ผลของการกลับของมุลภาพ (ก) อินพุท (ข) ผลลัพธ์ที่ได้จากการประมวลผลโดยอัลกอริทึมที่ 3-3.

3.1.4 การเน้นค่าความเข้มเป็นช่วงๆ

วิธีการเน้นค่าความเข้มเป็นบางช่วงๆ (Level Slicing) เป็นการเน้นค่าความเข้มในช่วง $[a, b]$ ใดๆ ที่สนใจให้เด่นขึ้นมาโดยการลดความสำคัญของข้อมูลที่อยู่นอกช่วงให้เป็นสีพื้น ซึ่งการเน้นค่าของข้อมูลภาพบางช่วงสามารถทำได้ดังนี้

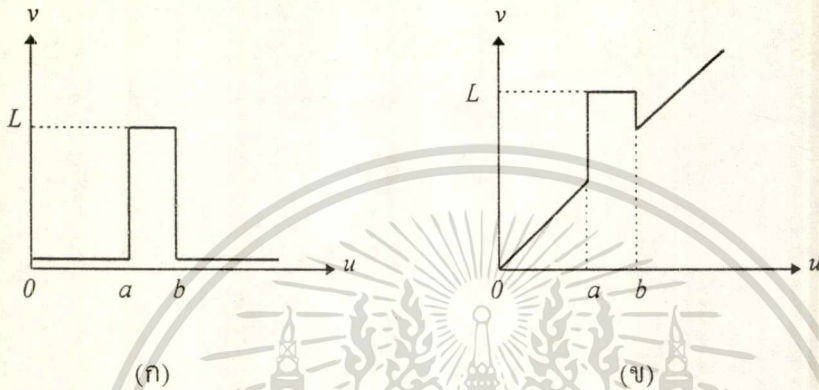
$$v = \begin{cases} L, & a \leq u \leq b \\ 0, & \text{ในกรณีอื่นๆ} \end{cases} \quad (3.5)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และในกรณีที่ไม้ตัดสี่เหลี่ยมออกสามารถทำได้ดังสมการที่ (3.6)

$$v = \begin{cases} L, & a \leq u \leq b \\ u, & \text{ในกรณีอื่น} \end{cases} \quad (3.6)$$

ในรูปที่ 3.8 แสดงกราฟลักษณะการแปลงข้อมูลในสมการที่ (3.5) และ (3.6) ตามลำดับ



รูปที่ 3.8 แสดงกราฟลักษณะการเน้นข้อมูลบางช่วง

(ก) เป็นการเน้นโดยกำจัดสี่เหลี่ยม (ข) เป็นกรณีที่มีสี่เหลี่ยม

การแปลงข้อมูลภาพโดยวิธีนี้เป็นการแยกสิ่งที่เราสนใจที่อยู่ในภาพ โดยเราต้องทราบถึงช่วง $[a, b]$ ของระดับความเข้มของวัตถุที่สนใจ ในอัลกอริทึมที่ 3-4. แสดงขั้นตอนของการประมวลผลโดยวิธีนี้

อัลกอริทึมที่ 3-4. แสดงขั้นตอนการดำเนินการปรับปรุงภาพ โดยวิธีการเน้นค่าความเข้มเป็นช่วงๆ

```
void Slicing_Digital_image(IMAGE I)
{
    int a, b;
    BYTE L=255;

    printf("Lower Bound := ");
    scanf("%d", &a);
    printf("Upper Bound := ");
    scanf("%d", &b);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for(int r=0; r<I.ROWS; r++)
    for(int c=0; c<I.COLS; c++)
    {
        BYTE m=I.data[r][c];

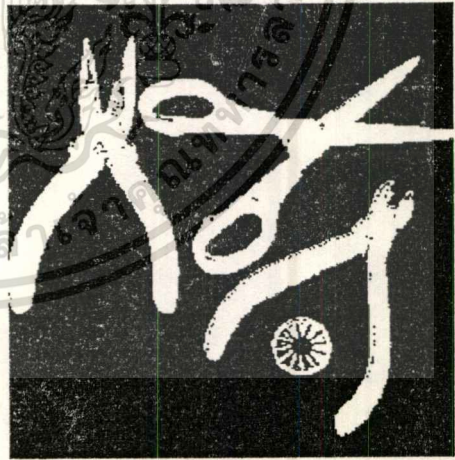
        if(m<a)
            I.data[r][c] = 0;
        if((m>=a) && (m<=b))
            I.data[r][c] = L;
        if(m>b)
            I.data[r][c] = 0;
    }
}

```

สำหรับผลลัพธ์ที่ได้จากการประมวลผล โดยอัลกอริทึมนี้แสดงในรูปที่ 3.9 ซึ่งเป็นภาพเครื่องมือ โดยขนาดของภาพเท่ากับ 256 x 256 จุด และค่าความเข้มของจุดภาพอยู่ในช่วง [6, 254]



(ก)



(ข)

รูปที่ 3.9 ผลของการปรับปรุงภาพโดยวิธีการเน้นข้อมูลภาพเป็นช่วงๆ (ก) อินพุท (ข) ผลลัพธ์ที่ได้จากการประมวลผลโดยอัลกอริทึมที่ 3-4. โดยที่ $a = 20$, $b = 140$

3.1.5 การเน้นข้อมูลภาพเฉพาะบางบิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าข้อมูลภาพมีการแจกแจงที่อยู่ในรูปแบบเดียวกัน (Uniform) และจุดภาพแต่ละจุดใช้จำนวนบิตเท่ากับ B บิตในการเก็บข้อมูลภาพ และเมื่อต้องการดึงข้อมูลภาพบิตที่ n มาแสดงสามารถทำได้ดังนี้

$$\text{ให้ } u = k_1 2^{B-1} + k_2 2^{B-2} + \dots + k_n 2^{B-n} + \dots + k_{B-1} 2 + k_B \quad (3.7)$$

$$\text{โดยที่ } k_n = i_n - 2i_{n-1}$$

ดังนั้นผลลัพธ์ที่เราต้องการสามารถกำหนดได้โดย

$$v = (i_n - 2i_{n-1})L$$

$$i_n = \text{Int} \left[\frac{u}{2^{B-n}} \right] \quad n = 1, 2, 3, \dots, B \quad (3.8)$$

โดยที่ *Int* เป็นการแปลงค่าเป็นเลขจำนวนเต็ม สำหรับการปรับปรุงภาพโดยวิธีนี้มีขั้นตอนการประมวลผลดังอัลกอริทึมที่ 3-5. และผลลัพธ์ที่ได้แสดงในรูปที่ 3.10 ซึ่งเป็นภาพเอ็กซ์เรย์รูปกระโหลกศีรษะโดยขนาดของภาพเท่ากับ 256 x 256 จุด และมีค่าความเข้มของจุดภาพอยู่ในช่วง [6, 248]

อัลกอริทึมที่ 3-5. แสดงขั้นตอนการประมวลผลภาพโดยวิธีการเน้นข้อมูลภาพเฉพาะบางบิต

```
void BitExtraction_Digital_Image(IMAGE I)
{
    int L=255, i;
    BYTE B=8; // B เป็นจำนวนบิตที่ใช้ในการเก็บข้อมูลภาพ

    printf("Maximum Pixel Value := %d", I.MAX);
    printf("Minimum Pixel Value := %d", I.MIN);
    printf("n-th Bit to extracted n := ");
    scanf("%d", &i);

    for(int r=0; r<I.ROWS; r++)
        for(int c=0; c<I.COLS; c++)
        {
```

```

int u = (int)(m/pow(2, B-i));
int ui_1 = (int)(m/pow(2, B-i-1));
I.data[r][c] = (u - 2*ui_1)*L;

```



(ก)

(ข)

รูปที่ 3.10 ผลของการปรับปรุงภาพโดยวิธีการเน้นข้อมูลเฉพาะบางบิต (ก) อินพุต (ข) ผลลัพธ์ที่ได้จากการประมวลผลโดยอัลกอริทึมที่ 3-5. ที่ $n = 7$

3.1.6 การตัดทอนข้อมูลภาพบางบิต

การแปลงข้อมูลภาพโดยวิธีนี้มีด้วยกันสองลักษณะคือ

3.1.6.1 การกำจัดบิตที่มีนัยสำคัญสูง ซึ่งสามารถทำได้ดังนี้

$$v = 2u \text{MODULO}(L+1), \quad 0 \leq u \leq L \quad (3.9)$$

3.1.6.2 การกำจัดบิตที่มีนัยสำคัญต่ำ ซึ่งสามารถคำนวณได้โดย

$$v = 2 \text{Int} \left[\frac{u}{2} \right] \quad (3.10)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการประมวลผลภาพโดยวิธีนี้ได้แสดงไว้ในอัลกอริทึมที่ 3-6. สำหรับภาพที่ใช้ในการทดสอบอัลกอริทึมนี้เป็นภาพเอ็กซ์เรย์ที่บริเวณฝ่ามือ (รูปที่ 3.11) โดยภาพมีขนาด 256 x 256 จุด และค่าความเข้มของภาพนี้อยู่ในช่วง [5, 84]

อัลกอริทึมที่ 3-6. แสดงขั้นตอนการปรับปรุงภาพโดยวิธีการตัดทอนข้อมูลบางบิต

```
void BitRemoval_Digital_Image(IMAGE I)
{
    printf("Maximum Pixel Value := %d", I.MAX);
    printf("Minimum Pixel Value := %d", I.MIN);

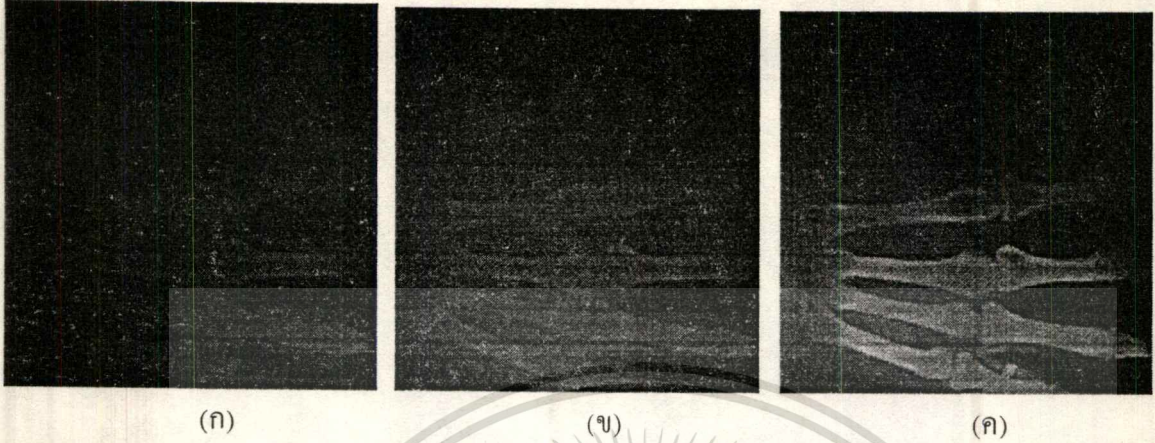
    int i;
    printf("1. LSB or 2. MSB");
    scanf("%d", &i);
    printf("%s Bit Removal", i==1 ? "Least-Significant-Bit":"Most-Significant-Bit");

    if(i==1)
        for(int r=0; r<I.ROWS; r++)
            for(int c=0; c<I.COLS; c++)
            {
                BYTE m=I.data[r][c];

                I.data[r][c] = (BYTE)(2*(m/2));
            }
    else
        for(int r=0; r<I.ROWS; r++)
            for(int c=0; c<I.COLS; c++)
            {
                BYTE m=I.data[r][c];
```

```
                I.data[r][c] = (BYTE)(2*(m%(I.MAX+1)));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 ผลของการปรับปรุงภาพโดยวิธีการตัดทอนข้อมูลภาพเฉพาะบางบิต (ก) อินพุต (ข) ผลของการกำจัดบิตที่มีนัยสำคัญสูง (ค) ผลของการกำจัดบิตที่มีนัยสำคัญต่ำ

3.1.7 การบีบข้อมูลภาพในบางช่วง

ข้อมูลภาพบางช่วงอาจจะกว้างเกินไป ดังนั้นเราสามารถปรับคอนทราสต์ภาพได้โดยการใช้ลอการิทึมในการแปลงดังนี้

$$v = c \text{Log}_{10}(1 + u), \quad u > 0$$

$$c \equiv \frac{L}{\text{Log}_{10}(1 + L)} \quad (3.11)$$

สำหรับค่า L ในที่นี้จะเป็ค่าความเข้มสูงสุดของข้อมูลภาพ สำหรับขั้นตอนการปรับปรุงภาพโดยวิธีนี้ได้แสดงไว้ในอัลกอริทึมที่ 3-7. และผลลัพธ์ที่ได้จากการประมวลผลภาพโดยอัลกอริทึมที่ 3-7. ได้แสดงไว้ในรูปที่ 3.12 ซึ่งเป็นภาพแมลงที่ขนาดของภาพ 256 x 256 จุด และค่าความเข้มของภาพอยู่ในช่วง $[0, 251]$

อัลกอริทึมที่ 3-7. แสดงขั้นตอนการปรับปรุงภาพโดยวิธีการบีบข้อมูลในบางช่วง

```
void RangeCompress_Digital_Image(IMAGE I)
```

```
{
```

```
    float v;
```

```
    printf("Maximum Pixel Value := %d", I.MAX);
```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

```
    printf("Minimum Pixel Value := %d", I.MIN);
```

ไม่ว่ากรรมใดๆทั้งสิ้น อีกทั้งห้ามเผยแพร่เนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

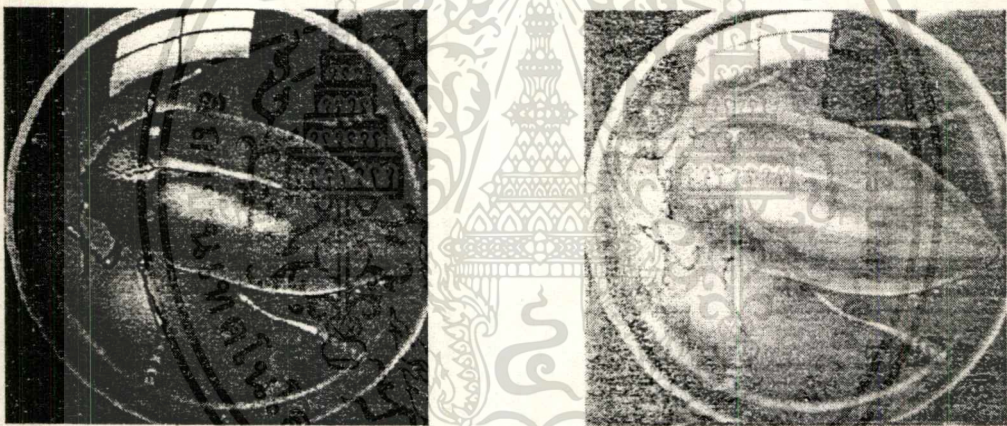
```

v = (I.MAX/(log10(1+I.MAX)));

for(int r=0; r<I.ROWS; r++)
    for(int c=0; c<I.COLS; c++)
    {
        BYTE m=I.data[r][c];

        I.data[r][c] = (BYTE)(v*log10(1+m));
    }
}

```



(ก)

(ข)

รูปที่ 3.12 ผลของการปรับปรุงภาพโดยวิธีการบีบข้อมูลในบางช่วง (ก) อินพุท (ข) ผลลัพธ์ที่ได้จากการประมวลผลโดยใช้อัลกอริทึมที่ 3-7.

3.1.8 การปรับปรุงภาพโดยใช้กฎพีชชี

การปรับปรุงภาพโดยใช้กฎพีชชี [52] เป็นการปรับปรุงภาพแบบ Contrast Stretch โดยใช้สมการดังต่อไปนี้

$$v = \left(K - \left(\frac{MAX - u}{Fd} \right)^{Fc} / K \right) * L \quad (3.12)$$

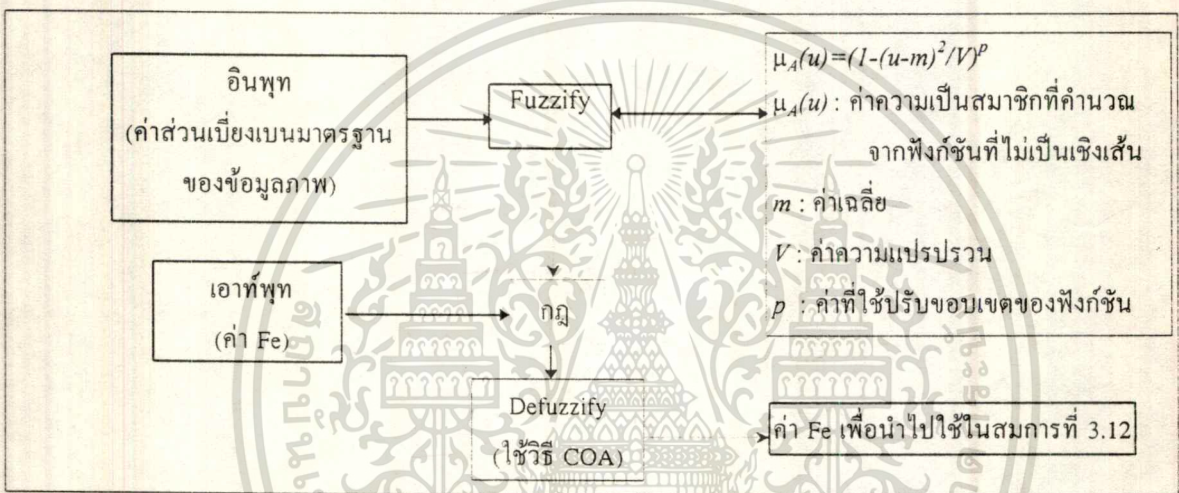
ที่ MAX เป็นค่าความเข้มสูงสุดของข้อมูลภาพ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$Fd = \frac{MAX - MID}{0.5^{1/Fe}} ; \quad MID = MIN + \frac{MAX - MIN + 1}{2} \quad (3.13)$$

MIN เป็นค่าความเข้มต่ำสุดของข้อมูลภาพ

$$K = \left(\frac{MAX - MIN}{Fd} \right)^{Fe} \quad (3.14)$$

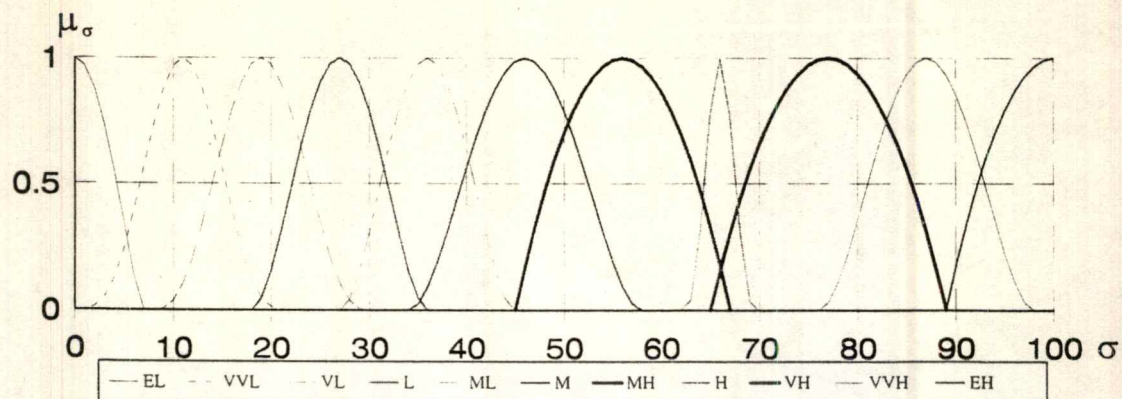
Fe เป็นตัวแปรที่ต้องการหาโดยใช้กฎฟัซซี่ ซึ่งสามารถหาได้โดยใช้บล็อกไดอะแกรมดังรูปที่ 3.13 และขอบเขตของกฎแต่ละข้อแสดงในตารางที่ 3.1



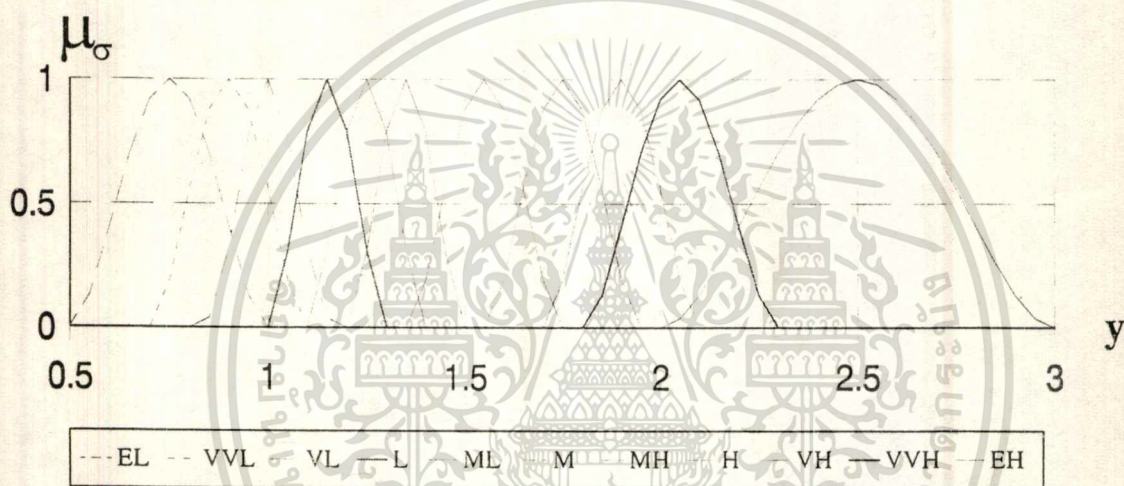
รูปที่ 3.13 บล็อกไดอะแกรมของกฎฟัซซี่ที่ใช้ในการปรับคอนทราสต์ภาพ

ตารางที่ 3.1 แสดงขอบเขตของกฎแต่ละข้อของอินพุตและเอาต์พุต

อินพุต						เอาต์พุต			
ลำดับที่	ชื่อกฎ	m	V	p	ขอบเขต	ชื่อกฎ	m	V	p
1	EL	0	49	1.5	0 ≤ σ ≤ 7	EL	0.75	0.0625	2
2	VVL	11	121	4	0 ≤ σ ≤ 22	VVL	0.9	0.04	2
3	VL	19	121	3	8 ≤ σ ≤ 30	VL	1	0.04	2
4	L	27	81	2	18 ≤ σ ≤ 36	L	1.15	0.0225	2
5	ML	36	81	2	27 ≤ σ ≤ 45	ML	1.25	0.0225	2
6	M	46	144	2	34 ≤ σ ≤ 58	M	1.35	0.0225	2
7	MH	56	121	1	45 ≤ σ ≤ 67	MH	1.55	0.04	2
8	H	66	16	4	62 ≤ σ ≤ 70	H	1.75	0.04	2
9	VH	77	144	1	65 ≤ σ ≤ 89	VH	1.9	0.04	2
10	VVH	87	121	2	76 ≤ σ ≤ 98	VVH	2.05	0.0625	2
11	EH	100	121	2	σ > 89	EH	2.5	0.25	2



(ก)



(ข)

รูปที่ 3.14 กฎฟัซซีที่ใช้ในการปรับปรุง (ก) กฎของอินพุท (ข) กฎของเอาต์พุท

และความสัมพันธ์ของกฎอินพุทและเอาต์พุทสามารถกำหนดได้ดังนี้

กฎที่ 1: IF (σ คือ EL) THEN (Fe คือ EH)

กฎที่ 2: IF (σ คือ VVL) THEN (Fe คือ H)

กฎที่ 3: IF (σ คือ VL) THEN (Fe คือ VH)

กฎที่ 4: IF (σ คือ L) THEN (Fe คือ MH)

กฎที่ 5: IF (σ คือ ML) THEN (Fe คือ ML)

กฎที่ 6: IF (σ คือ M) THEN (Fe คือ M)

กฎที่ 7: IF (σ คือ MH) THEN (Fe คือ L)

กฎที่ 8: IF (σ คือ H) THEN (Fe คือ VVH)

กฎที่ 9: IF (σ คือ VH) THEN (Fe คือ VL)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ห้ามทำมั่วหรือดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- กฎที่ 10: IF (σ คือ VVH) THEN (Fe คือ VVL)
- กฎที่ 11: IF (σ คือ EH) THEN (Fe คือ EL)

สำหรับผลลัพธ์ของการปรับปรุงภาพโดยวิธีนี้ได้แสดงในรูปที่ 3.15x ซึ่งเป็นการเปรียบเทียบกับ การปรับปรุงภาพโดยวิธีฮิสโตแกรมอีควอลไลเซชัน (ในหัวข้อ 3.1.9) และผลลัพธ์ที่ได้จากการทดลองกับ หลายๆ ภาพปรากฏว่าวิธีการปรับปรุงภาพโดยพีชชีเซตจะมีคุณภาพของเอาท์พุทที่ดีกว่า [51] รูปที่ 3.15ก เป็นภาพข้างที่อยู่ในน้ำโดยขนาดของภาพนี้เท่ากับ 256 x 256 จุด และค่าความเข้มของจุดภาพอยู่ในช่วง [0, 87]

3.1.9 การปรับปรุงภาพด้วยวิธีฮิสโตแกรมอีควอลไลเซชัน

วิธีฮิสโตแกรมอีควอลไลเซชัน (Histogram Equalization) เป็นวิธีหนึ่งที่ยอมรับใช้ในการปรับปรุงภาพ ฮิสโตแกรมของรูปภาพเป็นการแทนความถี่สะสมของระดับความเข้มของสีเทาที่ปรากฏในรูปภาพ วิธีนี้เป็นการขยายช่วงของข้อมูลภาพให้กว้างขึ้นเพื่อให้ภาพมีความคมชัดขึ้น ซึ่งจุดประสงค์ของวิธีนี้จะเหมือนกับวิธีที่ใช้พีชชีเซตที่นำค่าความเป็นสมาชิกมาคูณค่าด้วย α แต่ในวิธี Histogram Equalization นี้เป้าหมายของมันก็คือ ทำให้ผลลัพธ์ของรูปภาพมีการแจกแจงเป็น Uniform ในวิธีการของฮิสโตแกรมเราจะพิจารณาค่าของจุดภาพ $u \geq 0$ ($u \in U$ เมื่อ U เป็นเซตของจุดภาพในภาพใดๆ) เป็นตัวแปรเชิงสุ่ม ดังนั้นให้ $p_U(u)$ เป็นฟังก์ชันความหนาแน่นของความน่าจะเป็นแบบต่อเนื่อง (Continuous Probability Density Function) และ $P_U(u) = F[U \leq u]$ เป็นฟังก์ชันการแจกแจงความน่าจะเป็นแบบสะสม (Cumulative Probability Distribution Function) ดังนั้นความน่าจะเป็นของตัวแปรเชิงสุ่ม v สามารถกำหนดได้ว่า

$$v = F_U(u) = \int_0^u P_U(u) du \tag{3.15}$$

ซึ่ง v จะมีการแจกแจงเป็น uniform ซึ่งมีค่าอยู่ในช่วง $(0 \leq v \leq 1)$ การนำเอาวิธี Histogram Equalization มาใช้กับภาพดิจิทัล จะสมมติให้ u เป็นข้อมูลของรูปภาพที่มีระดับความเข้มของสีเทา L ระดับ $x_i; i=0, 1, 2, \dots, L-1$ เพราะฉะนั้นความน่าจะเป็นของจุดภาพ x_i ที่จะปรากฏในภาพจะกำหนดได้โดยให้ $h(x_i)$ เป็นฮิสโตแกรมของข้อมูลภาพซึ่งเป็นจำนวนของจุดภาพที่ระดับความเข้มของสีเทา (x_i) ดังนั้น

$$p_U(x_i) = \frac{h(x_i)}{\sum_{i=0}^{L-1} h(x_i)} \tag{3.16}$$

ค่าความน่าจะเป็นแบบสะสม v สามารถกำหนดได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$v = \sum_{x_i=0}^k p_U(x_i) \quad (3.17)$$

ในการหาผลลัพธ์ v' ซึ่งเป็นตัวแปรเชิงสุ่มที่ใช้แทนค่าความเข้มที่ผ่านการปรับปรุงแล้ว และสามารถคำนวณได้ดังนี้

$$v' = \frac{(L-1)(v - v_{\min})}{1 - v_{\min}} + 0.5 \quad (3.18)$$

ที่ v_{\min} เป็นค่าความน่าจะเป็นของ v มีค่าต่ำสุด และในที่นี้ข้อมูลภาพมีขนาด 8 บิต เพราะฉะนั้น $L=255$ สำหรับขั้นตอนการปรับปรุงภาพโดยวิธี Histogram Equalization แสดงในอัลกอริทึมที่ 3-8. และผลลัพธ์ที่ได้แสดงในรูปที่ 3.15ค

อัลกอริทึมที่ 3-8. แสดงขั้นตอนการปรับปรุงภาพโดยวิธี Histogram Equalization

```
void Eequalization(IMAGE I)
{
#define SIZE 256
float prob, accum_prob[SIZE], min=0;
WORD his_value[SIZE];
unsigned x, y;

for(x=0; x<SIZE; x++)
{
    his_value[x] = 0;
    accum_prob[x] = 0;
}

for(x=0; x<I.ROWS; x++)
    for(y=0; y<I.COLS; y++)
        his_value[I.data[x][y]] += 1; // จำนวนค่าความถี่สะสม

prob = 0.0;
x = I.ROWS;
y = I.COLS;
min = (float)(x*y-1);
```

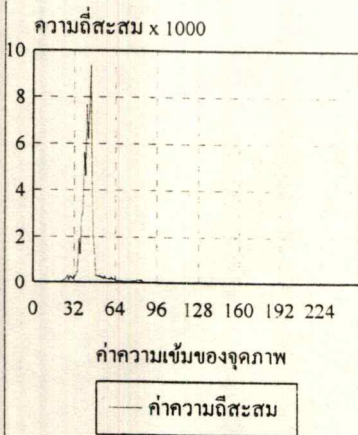
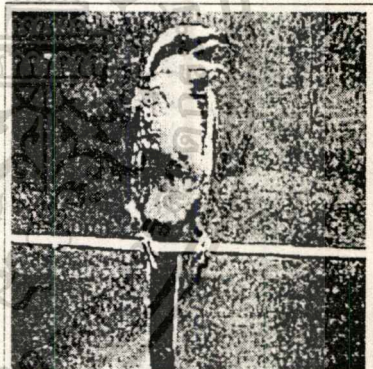
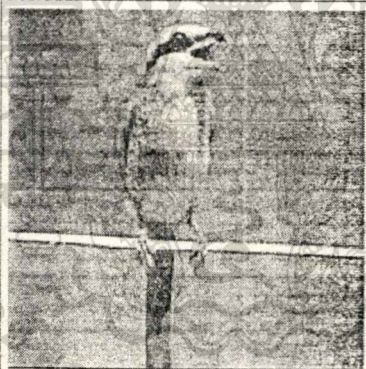
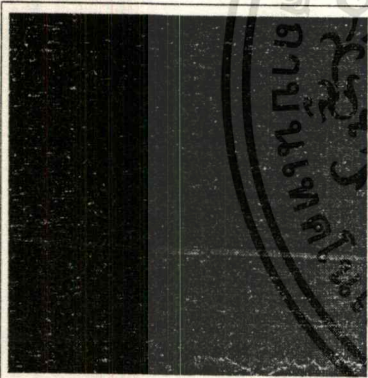
```

for(x=0; x<SIZE; x++)
{
    prob += his_value[x]/(min+1);           // คำนวณค่าความน่าจะเป็น
    accum_prob[x] = prob;                 // กำหนดค่าความน่าจะเป็นแบบสะสม
}

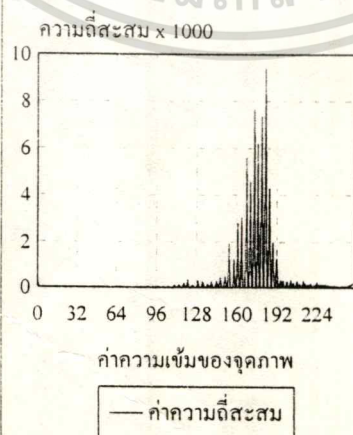
min = accum_prob[0];

for(x=0; x<I.ROWS; x++)
    for(y=0; y<I.COLS; y++)
    {
        prob = (accum_prob[I.data[x][y]]-min)/(1-min)*255.0+0.5;
        I.data[x][y] = (unsigned char)prob;
    }
}

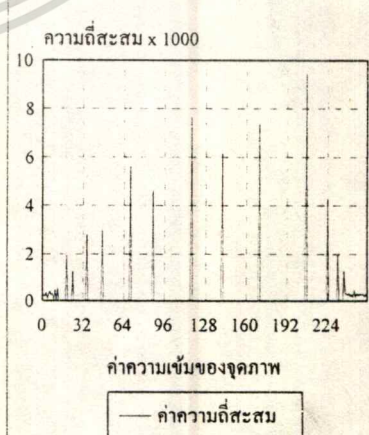
```



(ก)



(ข)

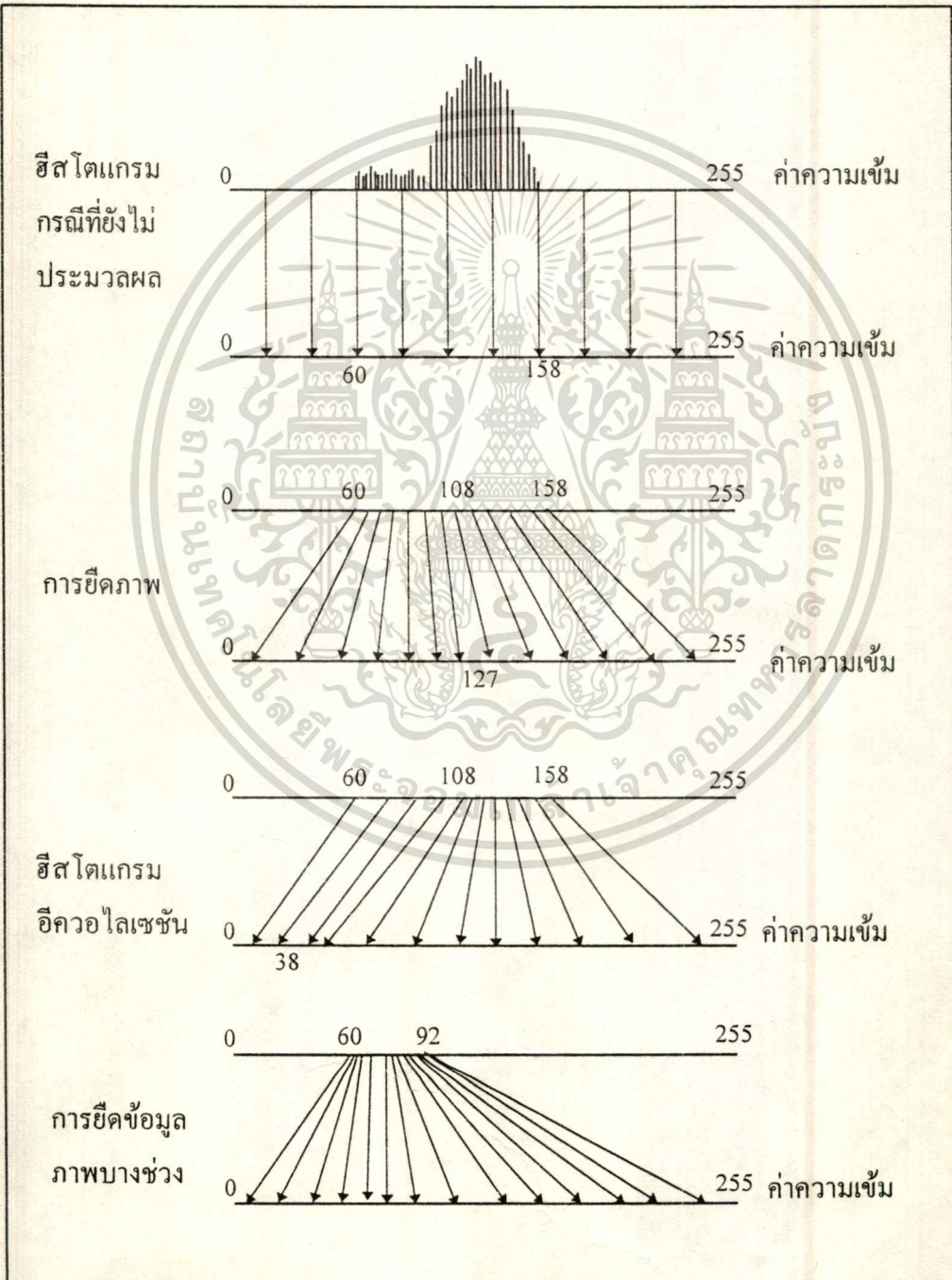


(ค)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.15 เปรียบเทียบผลของการประมวลผลภาพพร้อมทั้งแสดงการแจกแจงของความถี่สะสมของแต่ละวิธีซึ่งแสดงอยู่ใต้ภาพ (ก) อินพุท (ข) การปรับปรุงภาพโดยพีชคณิต (ค) การปรับปรุงภาพโดยวิธี Histogram Equalization

การปรับปรุงภาพแบบจุดที่เสนอมาพอจะสรุปลักษณะต่างๆ ได้ดังรูปที่ 3.16



3.2 การกำจัดสัญญาณรบกวน

สัญญาณรบกวนที่ปะปนอยู่ในภาพส่วนใหญ่จะเกิดจากตัวรับสัญญาณไฟฟ้า, การขยายสัญญาณภาพ, และความผิดพลาดที่เกิดขึ้นของช่องรับสัญญาณ, การกำจัดสัญญาณรบกวนมีวิธีการต่างๆ ดังนี้

3.2.1 การกำจัดสัญญาณรบกวนด้วยตัวกรองสัญญาณเชิงเส้น สัญญาณรบกวนโดยทั่วไปจะมีความถี่สูงกว่าสัญญาณภาพปกติ สัญญาณรบกวนประเภทนี้สามารถกำจัดออกได้โดยการใช้ตัวกรองสัญญาณที่ยอมให้ความถี่ต่ำผ่านไปได้ ซึ่งสามารถกำหนดได้ดังนี้

$$W = \left(\frac{1}{b+2} \right)^2 \begin{bmatrix} 1 & b & 1 \\ b & b^2 & b \\ 1 & b & 1 \end{bmatrix} \quad (3.19)$$

โดยที่ b เป็นจำนวนจริงที่มากกว่าศูนย์ วินโดว์ในสมการที่ (19) เป็นวินโดว์ขนาด 3×3 ซึ่งในกรณีที่ $b = 1$ เมื่อนำตัวกรองสัญญาณไปคอนโวลูชันกับภาพก็จะเหมือนกับการทำการเฉลี่ยข้อมูลภาพภายในวินโดว์ขนาด 3×3 (ดูรูปที่ 3.17ก และผลลัพธ์ที่ได้แสดงในรูปที่ 3.18ง) และในกรณีที่ $b = 2$ ตัวกรองสัญญาณนี้จะถูกเรียกว่าตัวกรองสัญญาณ Gaussian (รูปที่ 3.17ข และผลลัพธ์ที่ได้แสดงในรูปที่ 3.18จ) การกำจัดสัญญาณรบกวนด้วยตัวกรองสัญญาณที่เป็นเชิงเส้นจะใช้ได้ผลกับสัญญาณรบกวนที่มีความต่อเนื่อง อย่างเช่น สัญญาณรบกวนที่มีการแจกแจงแบบ Gaussian (รูปที่ 3.18ข) และสัญญาณรบกวนที่มีการแจกแจงแบบยูนิฟอร์ม (รูปที่ 3.18ค)

$$W_{average} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (ก) \quad W_{gaussian} = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad (ข)$$

รูปที่ 3.17 แสดงตัวกรองสัญญาณเชิงเส้น (ก) ตัวกรองที่ใช้เฉลี่ยสัญญาณ (ข) ตัวกรอง Gaussian

3.2.2 การกำจัดสัญญาณรบกวนด้วยวิธีที่ไม่เป็นเชิงเส้น เทคนิคอันนี้จะลดสัญญาณรบกวนและยังคงรายละเอียดของภาพได้ดีกว่า การกำจัดสัญญาณรบกวนด้วยตัวกรองเชิงเส้น สำหรับการกำจัดสัญญาณรบกวนด้วยวิธีที่ไม่เป็นเชิงเส้นประกอบด้วยวิธีต่างๆ ดังนี้

3.2.2.1 วิธีการเน้นโครง

เอกสารการกำจัดสัญญาณรบกวนด้วยวิธีการเน้นโครงหรือเอาท์ไลเออร์ (Outlier) [8] สามารถดำเนินการได้โดยการคอนโวลูชันดังสมการข้างล่างนี้ แผลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{bmatrix}
 \cdot & \cdot & \cdot & \cdot & \dots \\
 \cdot & o_1 & o_2 & o_3 & \dots \\
 \cdot & o_8 & x & o_4 & \dots \\
 \cdot & o_7 & o_6 & o_5 & \dots \\
 \cdot & \cdot & \cdot & \cdot & \dots \\
 \cdot & \cdot & \cdot & \cdot & \dots \\
 \cdot & \cdot & \cdot & \cdot & x_{MN}
 \end{bmatrix}
 \Rightarrow IF \left(x - \frac{1}{8} \sum_{i=1}^8 o_i \right) > \varepsilon \text{ THEN } x = \frac{1}{8} \sum_{i=1}^8 o_i \quad (3.20)$$

โดยค่า ε เป็นค่าเทรชโฮล การกำจัดสัญญาณรบกวนโดยวิธีนี้แต่ละจุดภาพ x จะถูกเปรียบเทียบกับค่าเฉลี่ยของจุดภาพที่อยู่รอบข้าง 8 จุด ถ้าขนาดของความแตกต่างมากกว่าค่าเทรชโฮล ε จะถือว่าจุดภาพนั้นเป็นสัญญาณรบกวน ในอัลกอริทึมที่ 3-9. แสดงขั้นตอนการกำจัดสัญญาณรบกวนโดยวิธีนี้ และผลลัพธ์ที่ได้แสดงในรูปที่ 3.18จ

อัลกอริทึมที่ 3-9. ขั้นตอนการกำจัดสัญญาณรบกวนโดยวิธี Outlier

```

void Outlier(IMAGE I)
{
    int E; // เป็นค่าเทรชโฮล ( $\varepsilon$  ในสมการที่ 3.20)
    int K = 3; // K เป็นขนาดของวินโดว์ (K x K)
    int mark = K/2;

    printf("\nThreshold = ");
    scanf("%d", &E);

    for(int i = 0; i < I.ROWS; i++)
        for(int j = 0; j < I.COLS; j++)
            {
                float sum = 0;

                for(int k = 0; k < K; k++)
                    for(int l = 0; l < K; l++)
                        {
                            if( (mark != k) || (mark != l) )

```

```

sum += I.data[i+k][j+1];
    }
sum = sum/(float)(K*K-1);
if( (I.data[i+mark][j+mark] - (int)sum) > E )
    I.data[i+mark][j+mark] = (int)sum;
}
}

```

3.2.2.2 ตัวกรองสัญญาณค่ามัธยฐาน

ตัวกรองสัญญาณด้วยค่ามัธยฐาน (Median Filter) เป็นตัวกรองสัญญาณอีกวิธีหนึ่งที่นิยมใช้กันทั่วไป เพราะวิธีนี้นอกจากจะกำจัดสัญญาณรบกวนได้แล้วก็ยังสามารถคงสภาพของขอบภาพได้ดีอีกด้วย และวิธีการก็ไม่ยุ่งยากซับซ้อน ซึ่งสามารถดำเนินการได้ตั้งอัลกอริทึมที่ 3-10. และผลลัพธ์ของการกำจัดสัญญาณรบกวนด้วยตัวกรองมัธยฐานได้แสดงในรูปที่ 3.18

อัลกอริทึมที่ 3-10. แสดงขั้นตอนการกำจัดสัญญาณรบกวนด้วยตัวกรองสัญญาณด้วยค่ามัธยฐาน โดยจะดำเนินการภายในกรอบขนาด 3×3

```

unsigned char Median_Filter (int W[3][3])
{
    int K=3; /* ขนาดของกรอบ */
    int Array[9];

    for (int i = 0; i < K; i++)
        for (int j = 0; j < K; j++) /* แปลงข้อมูลที่อยู่ในรูปของเมตริกซ์- */
            Array[K*i+j] = W[i][j]; /* ให้อยู่ในรูปของเวกเตอร์ */

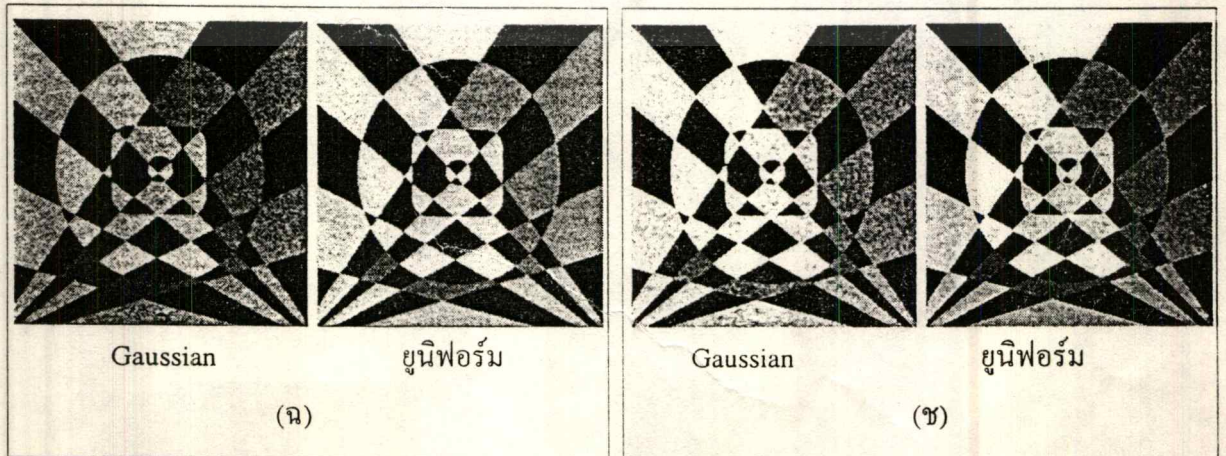
    /* ฟังก์ชัน Quick Sort ใช้ในการเรียงลำดับข้อมูลที่อยู่ตัวแปร Array โดยใช้ฟังก์ชัน Compare เป็นตัว
    เปรียบเทียบค่าที่อยู่ใน Array */
    qsort(Array, K*K, sizeof(int), Compare);

    return Array[(K*K)/2]; /* ส่งค่ามัธยฐานคืนสู่ฟังก์ชันหลัก */
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 3.18ก เป็นรูปที่สร้างขึ้นโดยไม่มีสัญญาณรบกวนและขนาดของรูปเท่ากับ 256 x 256 จุด รูปที่ 3.18ข เป็นรูปที่ถูกเพิ่มสัญญาณรบกวนชนิดที่มีการแจกแจงแบบ Gaussian โดยมีค่าส่วนเบี่ยงเบนมาตรฐานเท่ากับ 5 ส่วนรูปที่ 3.18ค เป็นรูปที่ถูกเพิ่มสัญญาณรบกวนชนิดที่มีการแจกแจงแบบยูนิฟอร์มที่มี ส่วนเบี่ยงเบนมาตรฐานเท่ากับ 5 เช่นกัน สำหรับผลลัพธ์ของการกำจัดสัญญาณรบกวนของแต่ละวิธีจะถูกแสดงเป็นสองภาพคือภาพซ้ายมือจะเป็นผลจากการกำจัดสัญญาณรบกวนที่เป็นแบบ Gaussian และทางขวาจะเป็นผลลัพธ์ของการกำจัดสัญญาณรบกวนที่เป็นยูนิฟอร์ม



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า รูปที่ 3.18 แสดงผลลัพธ์ของการประมวลผลภาพเพื่อการกำจัดสัญญาณรบกวนซึ่งมีสองชนิดคือ Gaussian

และ ยูนิฟอร์ม (ก) อินพุทที่ไม่มีสัญญาณรบกวน (ข) ภาพที่ถูกเพิ่มสัญญาณรบกวน Gaussian ด้วยค่าความแปรปรวนเท่ากับ 25 (ค) ภาพที่ถูกเพิ่มสัญญาณรบกวนยูนิฟอร์มด้วยค่าความแปรปรวนเท่ากับ 25 (ง- จ.) เป็นการกำจัดสัญญาณรบกวนด้วยตัวกรองเชิงเส้นโดยที่ $b = 1$ และ 2 ตามลำดับ (ฉ. - ช.) เป็นการกำจัดสัญญาณรบกวนด้วยตัวกรองสัญญาณที่ไม่เป็นเชิงเส้นโดยที่รูป (ฉ) เป็นการกำจัดสัญญาณรบกวนด้วยวิธี Outlier และ (ช) เป็นการกำจัดสัญญาณรบกวนด้วยตัวกรองสัญญาณมัธยฐาน

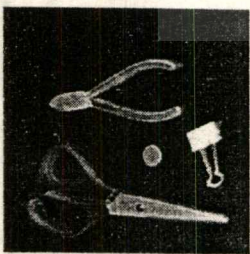
3.3 การเน้นขอบภาพให้คมชัด

การเน้นขอบภาพให้คมชัดสามารถดำเนินการได้โดยการใช้ตัวกรองสัญญาณที่ยอมให้ความถี่สูงผ่าน (High-Pass Filter) ในรูปที่ 3.19 เป็นวินโดว์ขนาด 3×3 ซึ่งเป็นตัวกรองสัญญาณที่ยอมให้ความถี่สูงผ่าน [8] ที่ใช้ในการปรับขอบภาพให้คมชัด สำหรับการดำเนินการเน้นขอบภาพให้คมชัดสามารถทำได้โดยการนำวินโดว์ในรูปที่ 3.19 ไปคอนโวลูชันกับภาพ

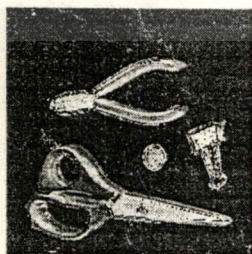
$$W_1 = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad W_2 = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad W_3 = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

รูปที่ 3.19 วินโดว์ที่เป็นตัวกรองสัญญาณชนิดที่ยอมให้ความถี่สูงผ่าน

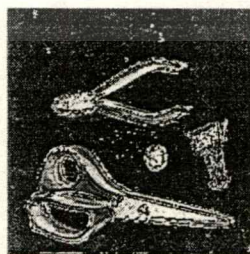
จะเห็นว่าวินโดว์ในรูปที่ 3.19 ทั้ง 3 วินโดว์จะมีลักษณะคล้ายกับวินโดว์ของ Laplacian ที่ใช้ในการหาของภาพในบทที่ 1 แต่วินโดว์ที่ใช้ในการเน้นขอบภาพในที่นี้ผลรวมของสมาชิกภายในวินโดว์จะไม่เท่ากับศูนย์ จากตัวกรองสัญญาณในรูปที่ 3.29 ได้นำมาประมวลผลเพื่อเน้นขอบภาพดังแสดงในรูปที่ 3.20 ซึ่งเป็นภาพเครื่องมือที่ขนาดของภาพเท่ากับ 256×256 จุด



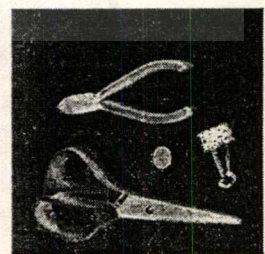
(ก)



(ข)



(ค)



(ง)

รูปที่ 3.20 แสดงผลลัพธ์ของการเน้นขอบภาพโดยใช้ตัวกรอง High-Pass ในรูปที่ 3.19 (ก) อินพุท (ข) ผล

จากการใช้ตัวกรอง W_1 (ค) ผลจากการใช้ตัวกรอง W_2 (ง) ผลจากการใช้ตัวกรอง W_3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุป

การปรับปรุงภาพเป็นการแปลงข้อมูลภาพ โดยฟังก์ชันที่ใช้ในการแปลงนี้จะต้องมีคุณสมบัติที่ทำให้ภาพที่ได้มีรูปแบบที่ดีขึ้น อย่างเช่นทำให้ภาพคมชัดขึ้น และการปรับปรุงภาพยังส่งผลถึงการวิเคราะห์ภาพโดยคอมพิวเตอร์ด้วย คือการปรับปรุงภาพสามารถจะช่วยให้การวิเคราะห์มีความถูกต้องยิ่งขึ้น เนื่องจากภาพที่ผ่านการปรับปรุงแล้วจะมีสัญญาณรบกวนน้อยลง แต่การปรับปรุงภาพบางครั้งอาจจะทำให้ข้อมูลที่สำคัญของภาพเกิดการสูญหายได้เหมือนกันถ้าไม่มีการพิจารณาเลือกวิธีที่เหมาะสมกับแต่ละภาพ เช่นในการกำจัดสัญญาณรบกวนถ้าใช้ตัวกรองสัญญาณขนาดใหญ่เกินไปจะทำให้ขอบภาพไม่คมชัด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4 นิวรอลเน็ตเวิร์ค

หลายศตวรรษมาแล้วที่มนุษย์ได้ประดิษฐ์และพัฒนาเครื่องจักรขึ้นมา เราจะเห็นว่าเครื่องจักรเหล่านี้จะถูกใช้สำหรับงานที่มีความยุ่งยากซับซ้อนและเป็นงานที่ซ้ำๆ ซากๆ ซึ่งถ้าใช้คนทำอาจจะทำให้เกิดความผิดพลาดได้ง่าย ในสมัยแรกๆ อาจจะเริ่มจากเครื่องจักรที่มีการทำงานที่ไม่ยุ่งยากซับซ้อน เช่น คาน, รอก เป็นต้น และต่อมาได้ประดิษฐ์สิ่งที่มีความซับซ้อนขึ้นมาเรื่อยๆ เช่น เครื่องจักรกล, หุ่นยนต์, และคอมพิวเตอร์ เป็นต้น จนกระทั่งปัจจุบันวิศวกรและนักวิทยาศาสตร์แขนงต่างๆ ได้พยายามที่จะพัฒนาเครื่องคอมพิวเตอร์ให้มีความฉลาดยิ่งขึ้น โดยพยายามให้เครื่องคอมพิวเตอร์มีการเรียนรู้และสามารถตัดสินใจได้เองหรือเป็นส่วนช่วยในการตัดสินใจในขั้นต้นได้ ระบบแบบจำลองโครงข่ายเซลล์ประสาทหรือระบบแบบจำลองนิวรอลเน็ตเวิร์ค (Artificial Neural Network Systems: ANNS) เป็นตัวอย่างหนึ่งที่จะทำให้เครื่องคอมพิวเตอร์ต่างๆ มีศักยภาพเพิ่มขึ้น

เมื่อเราพิจารณาถึงมนุษย์และสัตว์จะเห็นว่าสามารถรู้จักภาพและวิเคราะห์แยกแยะเสียงได้ดีกว่าคอมพิวเตอร์ที่ทันสมัยที่สุดแม้จะเป็นคอมพิวเตอร์ที่ถูกสร้างขึ้นมาเพื่องานทางด้านนี้โดยเฉพาะก็ตาม และนักวิจัยในสาขานี้ได้คาดหวังกกันว่าระบบ ANN จะเป็นคลื่นลูกใหม่ที่ใช้ในการประมวลผลข่าวสารสำหรับใช้แทนหรือสนับสนุนการคำนวณแบบเดิม ANNS จะช่วยเสริมความสามารถของคอมพิวเตอร์สำหรับการประมวลผลในระบบใหญ่ๆ ซึ่งระบบนี้สามารถเรียนรู้และตัดสินใจสำหรับกระบวนการของระบบนั้นๆ ได้

สำหรับการทำงานของระบบ ANN จะถอดแบบมาจากการทำงานของระบบสมองของมนุษย์ คือจะมีการส่งผ่านข้อมูลกัน โดยการเชื่อมต่อของโหนดกันเป็นโครงข่ายร่างแหอย่างหนาแน่น และมีการทำงานในลักษณะขนาน ซึ่งในหัวข้อถัดไปจะอธิบายลักษณะการทำงานของระบบเซลล์ประสาทของมนุษย์เพื่อจะได้เทียบเคียงการทำงานและทำให้สามารถเข้าใจการทำงานของ ANNS ได้ดีขึ้น

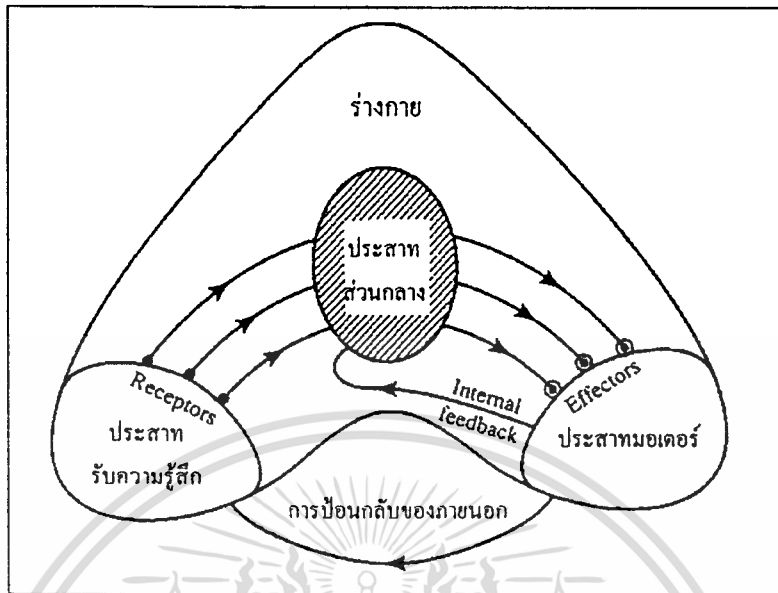
4.1 ความรู้พื้นฐานทางกายภาพของระบบเซลล์ประสาท

ระบบประสาทมีหน้าที่ควบคุมรักษาสมดุลภายในร่างกายให้คงที่ ระบบประสาทของมนุษย์เป็นระบบที่มีโครงสร้างที่ซับซ้อนซึ่งจะประกอบด้วยเซลล์ประสาท (Neurons) ประมาณ 10^{11} เซลล์ และเซลล์ประสาทแต่ละเซลล์จะเชื่อมต่อกับเซลล์ประสาทตัวอื่นอีกประมาณ 10^4 เซลล์ ระบบประสาทมีอยู่ด้วยกัน 3 ส่วน (รูปที่ 4.1 [53]) คือ

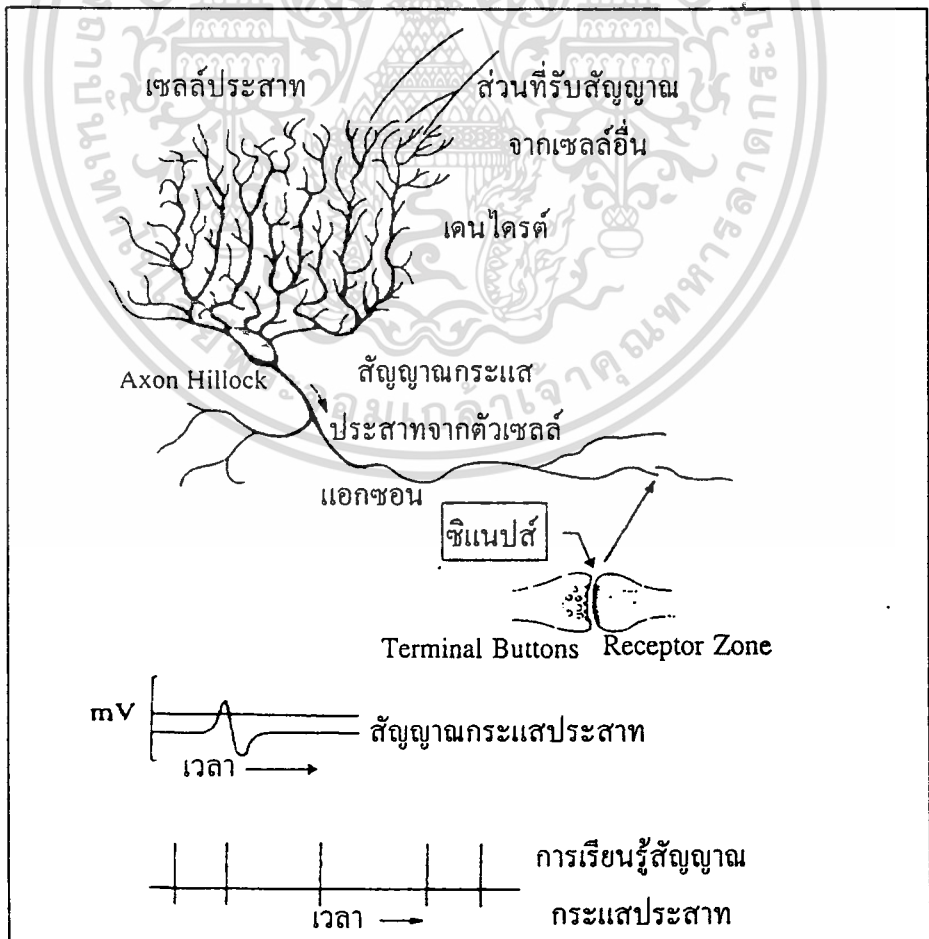
1. ระบบประสาทส่วนกลาง (Central Nervous System)
2. ระบบประสาทรับความรู้สึก (Sensory Systems)
3. ระบบประสาทมอเตอร์ (Motor Systems)

จากรูปที่ 4.1 ระบบประสาทส่วนกลางจะทำหน้าที่วิเคราะห์ข้อมูลต่างๆ ที่รับมาจากระบบประสาทรับความรู้สึกที่ประกอบด้วยเซลล์ประสาทรับความรู้สึก (Sensory Receptor) ซึ่งจะรับรู้เกี่ยวกับการเปลี่ยนแปลงของสิ่งแวดล้อมรอบตัว และจะส่งข่าวสารไปยังระบบประสาทส่วนกลาง และเมื่อประสาทส่วนกลางประมวลผลแล้วก็จะส่งสัญญาณไปควบคุมสั่งการระบบประสาทมอเตอร์ของอวัยวะแสดงผล (Effector) ให้

ทำงาน ซึ่งจะทำให้เกิดการตอบสนองต่อการเปลี่ยนแปลงทั้งภายในและภายนอกร่างกาย



รูปที่ 4.1 การสื่อสารข้อมูลในระบบประสาท [53]



เอกสารนี้เป็นเอกสารที่ 4.2 โครงสร้างของเซลล์ประสาทและการส่งสัญญาณกระแสประสาท [53] ในด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

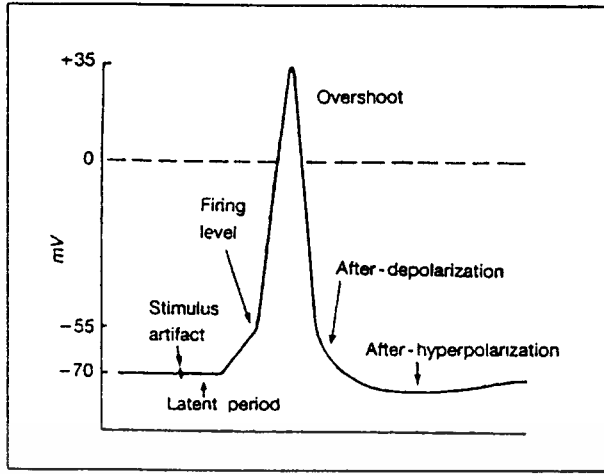
เซลล์ประสาทแต่ละเซลล์จะประกอบด้วย ตัวเซลล์ (Cell Body) และแขนงที่แยกออกจากตัวเซลล์ ซึ่งได้แก่ แอกซอน (Axon) และ เดนไดรต์ (Dendrite) (รูปที่ 4.2 [53]) แอกซอนมีจุดเริ่มต้นจากบริเวณส่วนที่หนาที่สุดของเซลล์ประสาทที่เรียกว่า Axon Hillock เซลล์ประสาทแต่ละเซลล์จะมีแอกซอนเพียงหนึ่งอัน ปลายของแอกซอนจะแยกเป็นแขนงหลายเส้นและไปสิ้นสุดที่ Terminal Buttons และที่จุดสิ้นสุดนี้จะมีตัวรับกระแสประสาท (Nerve Impulse) ที่เรียกว่า Receptor Zone ซึ่งต่ออยู่กับเดนไดรต์ของเซลล์ประสาทตัวอื่น กระแสประสาทหรือข้อมูลที่อยู่ในรูปของการเปลี่ยนแปลงทางไฟฟ้าเคมี จะไปกระตุ้นให้ตัวเซลล์ส่งสัญญาณไปตามแอกซอน ดังนั้นแอกซอนจะทำหน้าที่นำสัญญาณกระแสประสาทออกจากตัวเซลล์ไปสู่เซลล์ประสาทตัวอื่น และเดนไดรต์จะทำหน้าที่รับกระแสประสาทจากเซลล์อื่นเข้าสู่ตัวเซลล์

คุณสมบัติของเซลล์ประสาทโดยทั่วไปคือ จะมีความไวต่อการถูกกระตุ้น เมื่อเซลล์ประสาทถูกกระตุ้นจะทำให้เกิดการเปลี่ยนแปลงสมมูลย์ของไอออนบริเวณ Receptor Zone ซึ่งจะทำให้เกิดความต่างศักย์ขึ้น และเมื่อความต่างศักย์ของจุดนี้สูงกว่าค่าเทรชโฮลด์ก็ทำให้เกิดการส่งสัญญาณกระแสประสาทไปยังเซลล์อื่น

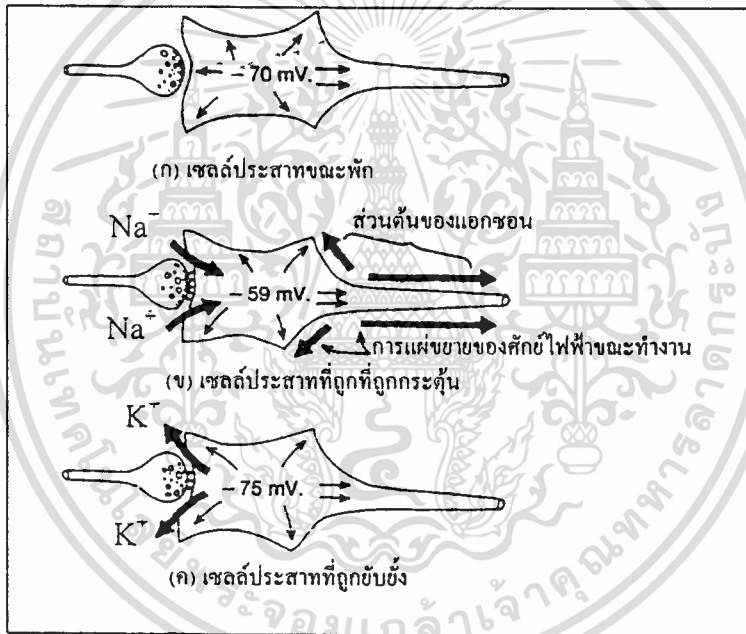
ความต่างศักย์ของเยื่อหุ้มเซลล์ประสาทขณะพักประมาณ -60 ถึง -90 mV ถ้ามีการกระตุ้นเซลล์ประสาทจะมีการแก้โพลาไรซ์ (Depolarization) ของเยื่อหุ้มเซลล์ และหลังจากเกิดการแก้โพลาไรซ์จนความต่างศักย์เพิ่มขึ้นเป็น 15 mV แล้ว จะทำให้อัตราการแก้โพลาไรซ์เพิ่มขึ้นอย่างรวดเร็ว ซึ่งจุดนี้จะเรียกว่า Firing Level (รูปที่ 4.3 [54]) เมื่อความต่างศักย์เพิ่มขึ้นจนถึงจุดสูงสุด (≈ 35 mV) จะทำให้อัตราการแก้โพลาไรซ์ลดลงอย่างรวดเร็ว ($\approx 70\%$) และจะค่อยๆ ลดลงกลับเข้าสู่สภาวะปกติ หลังจากเข้าสู่สภาวะปกติแล้ว ความต่างศักย์จะลดลงต่ำกว่าระดับปกติซึ่งจะเรียกช่วงนี้ว่า After Hyperpolarization ลำดับของการเปลี่ยนแปลงความต่างศักย์ทั้งหมดนี้เรียกว่า ศักย์ไฟฟ้าขณะทำงาน (Action Potential)

กระแสประสาทที่ส่งผ่านจากเซลล์หนึ่งไปยังอีกเซลล์หนึ่ง จะต้องผ่านรอยต่อระหว่าง Terminal Bottom และ Receptor Zone ซึ่งจะเรียกรอยต่อนี้ว่า ซิแนปส์ (Synapse) การส่งผ่านกระแสประสาทกันระหว่างเซลล์ (Neurotransmission) ซิแนปส์จะปล่อยสารเคมีเพื่อไปกระตุ้น (Excitatory) สำหรับบางกรณีและบางกรณีก็ปล่อยสารเคมีเพื่อยับยั้ง (Inhibitory) และในขณะใดขณะหนึ่งซิแนปส์จะยอมให้เกิดขึ้นเพียงกรณีเดียวเท่านั้น คือถ้าไม่กระตุ้นก็จะปล่อยสารเพื่อยับยั้งการส่งผ่านกระแสประสาท ในรูปที่ 4.4 แสดงสภาวะต่างๆ ของเซลล์ประสาท [54] ในกรณีที่ซิแนปส์เกิดสภาวะกระตุ้นความต่างศักย์ระหว่างผนังเซลล์จะลดลงเป็น -58 mV จะทำให้เกิดการแก้โพลาไรซ์ขึ้น ซึ่งถ้าสูงพอถึงภาวะ Firing Level ก็จะทำให้เกิดศักย์ไฟฟ้าขณะทำงานขึ้น และการที่จะเกิดสภาวะเช่นนี้ขึ้นได้จะต้องอาศัยซิแนปส์ 10 ถึงหลายร้อยตัวในเวลาเดียวกันโดยขบวนการ Summation ซึ่งซิแนปส์เพียงตัวเดียวจะไม่สามารถกระตุ้นเซลล์ประสาทให้ถึงภาวะ Firing Level ได้ ในรูปที่ 4.5 แสดงศักย์ไฟฟ้าขณะทำงานที่เกิดขึ้นจากการรวมกันของซิแนปส์จำนวนต่างๆ กัน [54]

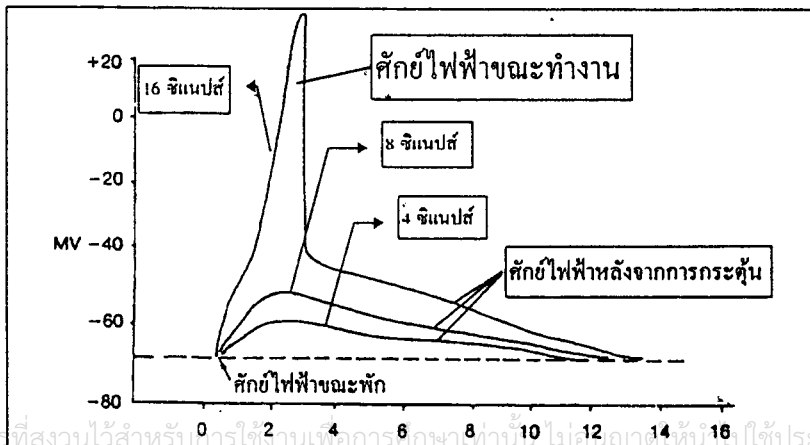
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 แสดงลักษณะการเกิดศักย์ไฟฟ้าขณะทำงาน



รูปที่ 4.4 แสดงภาวะต่างๆ ของเซลล์ประสาท (ก) ขณะพัก, (ข) ขณะกระตุ้น, และ (ค) ขณะถูกยับยั้ง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ... ใช้ประโยชน์ด้านการค้า...
รูปที่ 4.5 แสดงศักย์ไฟฟ้าขณะทำงานที่เกิดจากการรวมกันของจินแนปส์

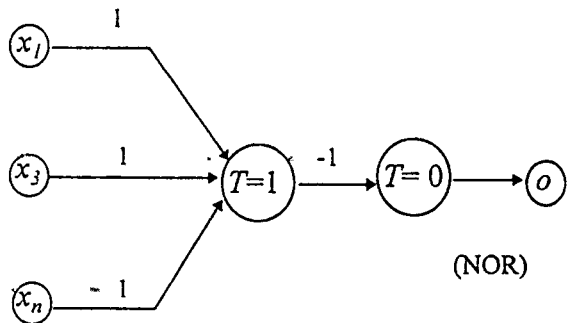
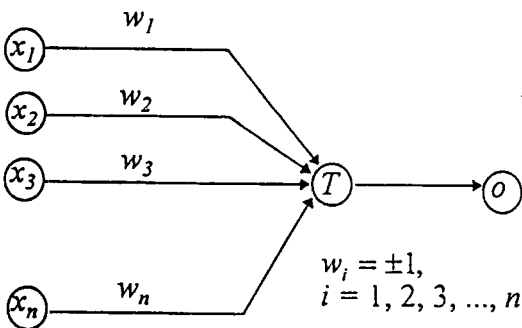
ในสภาวะกระตุ้นจะทำให้เกิดการส่งผ่านสัญญาณกระแสประสาท ไอออนของโซเดียม (Na^+), โพแทสเซียม (K^+) และ คลอไรด์ (Cl^-) จะซึมผ่านเยื่อหุ้มเซลล์ได้มากขึ้น ส่วนในสภาวะยับยั้ง โพแทสเซียม และคลอไรด์ จะไหลออกมาจากเซลล์และโซเดียมไม่มีการไหลเข้าเซลล์ซึ่งเป็นเหตุให้ประจุบวกภายในเซลล์ลดลงทำให้ความต่างศักย์ภายในเซลล์เป็นลบมากขึ้น (ประมาณ -75 mV) ซึ่งจะส่งผลให้เซลล์ประสาทถูกกระตุ้นยากขึ้นกว่าปกติ คือต้องใช้ความต่างศักย์เพิ่มขึ้นเป็น 16 mV จึงจะสามารถกระตุ้นเซลล์ประสาทให้ถึง Firing Level ได้ ซึ่งระยะนี้จะเรียกว่าระยะการดื้อ (Refractory Period)

4.2 แบบจำลองโครงข่ายเซลล์ประสาท

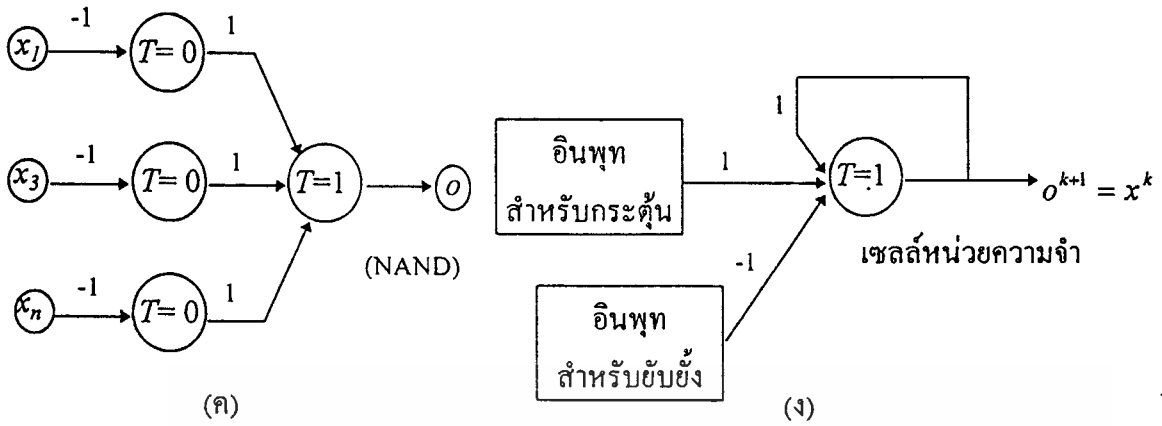
ในหัวข้อที่แล้วเราได้กล่าวถึงการทำงานของระบบประสาทโดยทั่วไป ของสัตว์เลี้ยงลูกด้วยนม ในหัวข้อนี้จะมาพิจารณากันถึงระบบการทำงานของแบบจำลองระบบประสาทที่จะใช้ในการประมวลผล โดยเครื่องคอมพิวเตอร์เพื่อนำไปใช้ควบคุม (รักษาสมดุล) ระบบต่างๆ แบบจำลองที่จะกล่าวมาเป็นแบบแรกในที่นี้ถูกเสนอโดย McCulloch และ Pitts ในปีค.ศ. 1943 ซึ่งแบบจำลองของเซลล์ประสาทแสดงในรูปที่ 4.6ก อินพุต x_i (สำหรับ $i = 1, 2, 3, \dots, n$) จะมีค่าเป็น $\{0, 1\}$ ซึ่งจะขึ้นอยู่กับสัญญาณอินพุตสัญญาณจากเซลล์อื่นในขณะนั้นว่าจะมีหรือไม่มีสัญญาณ ส่วนสัญญาณที่จะส่งต่อไปยังเซลล์ถัดไปซึ่งเป็นเซลล์ส่งผล (เซลล์ของผลลัพธ์) จะแทนด้วย o และ Firing Level ของแบบจำลองนี้ถูกกำหนดโดย

$$o^{k+1} = \begin{cases} 1 & \text{ถ้า } \sum_{i=1}^n w_i x_i^k \geq T \\ 0 & \text{ถ้า } \sum_{i=1}^n w_i x_i^k < T \end{cases} \quad (4.1)$$

โดยที่ $k = 0, 1, 2, \dots$ เป็นช่วงเวลาแบบไม่ต่อเนื่อง w_i เป็นค่าถ่วงน้ำหนักที่เชื่อมต่อกับอินพุตที่ i ซึ่งถ้า $w_i = +1$ แสดงถึงสถานะการกระตุ้นของซินแนปส์ และถ้า $w_i = -1$ ซินแนปส์จะมีการยับยั้งการส่งผ่านสัญญาณ และ T เป็นค่าความต่างศักย์เทรชโฮลด์หรือขีดเริ่มเปลี่ยน ซึ่งถ้าค่าผลรวมของผลคูณระหว่างค่าถ่วงน้ำหนักกับสัญญาณอินพุตจะต้องมากกว่า T จึงจะมีสัญญาณผ่านไปยังเซลล์อื่นได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 (ก) ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
 (ข)

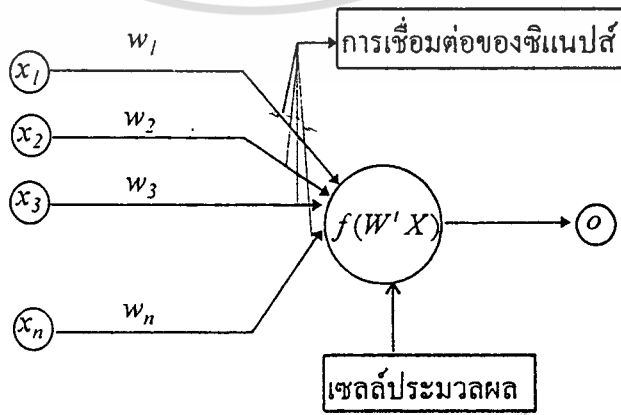


รูปที่ 4.6 แบบจำลองเซลล์ประสาทของ McCulloch-Pitts และ โครงข่ายลอจิกพื้นฐาน

(ก) โค้ดอะแกรมแบบจำลอง (ข) เกต NOR (ค) เกต NAND (ง) เซลล์หน่วยความจำ

แบบจำลองของ McCulloch-Pitts แม้จะเป็นโครงข่ายแบบง่าย ๆ แต่แบบจำลองนี้สามารถใช้แทนตัวดำเนินการทางลอจิกได้ คือตัวดำเนินการ NOT, OR, และ AND เมื่อเรากำหนดค่าถ่วงน้ำหนักและค่าเทรชโฮลด์ที่เหมาะสมให้กับโครงข่าย ในรูปที่ 4.6ข และ ค. เป็นแบบจำลองที่ใช้แทนตัวดำเนินการซึ่งเป็น Gates ที่ใช้ในพีชคณิตบูลีนคือ NOR และ NAND ซึ่งมีอินพุตสามตัว และในรูปที่ 4.6ง เป็นเซลล์ที่ใช้แทนหน่วยความจำซึ่งเป็นโครงข่ายที่มีการป้อนกลับจากเอาต์พุต

ANN ที่จะกล่าวถึงต่อไปนี้เป็นแบบจำลองที่ใช้เป็นต้นแบบในการศึกษาแบบจำลองและระบบการเรียนรู้ชนิดต่างๆ ที่จะกล่าวถึงต่อไป ซึ่งโครงข่ายนี้จะคล้ายกับแบบจำลองของ McCulloch-Pitts ส่วนที่ต่างกันคือ ค่าของตัวแปรต่างๆ ที่ใช้ในแบบจำลอง เนื่องจากค่าของตัวแปรในแบบจำลอง ANN เป็นเลขจำนวนจริง และค่าถ่วงน้ำหนักได้จากการเรียนรู้ ซึ่งแบบจำลองนี้แสดงในรูปที่ 4.7



รูปที่ 4.7 แบบจำลองโครงข่ายเซลล์ประสาทเทียม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้เข้าไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.7 แสดงโครงข่ายการเชื่อมต่อของแบบจำลองเซลล์ประสาทที่สามารถสอนให้โครงข่ายตัดสินใจได้ โดยมี x_i เป็นสัญญาณอินพุต และ w_i เป็นค่าถ่วงน้ำหนักที่ได้จากการสอนโครงข่าย และแต่ละโหนดในโครงข่ายจะใช้แทนเซลล์ประสาทแต่ละเซลล์ ซึ่งบางครั้งจะเรียกว่าหน่วยประมวลผลพื้นฐาน (Process Element Unit) และมีซีแนปส์ซึ่งจะเชื่อมต่อโหนดเพื่อใช้ในการส่งผ่านสัญญาณ การกระตุ้นหรือยับยั้งสัญญาณจะขึ้นอยู่กับค่าถ่วงน้ำหนัก w_i และสำหรับสัญญาณเอาต์พุตสามารถคำนวณได้ดังนี้

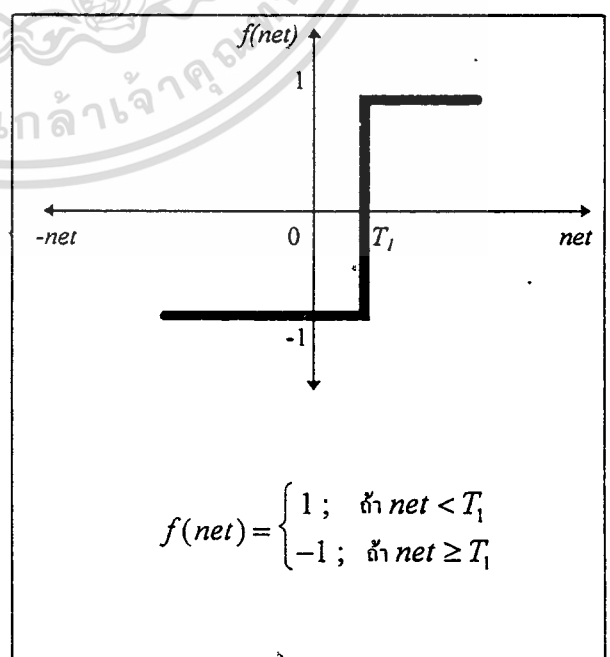
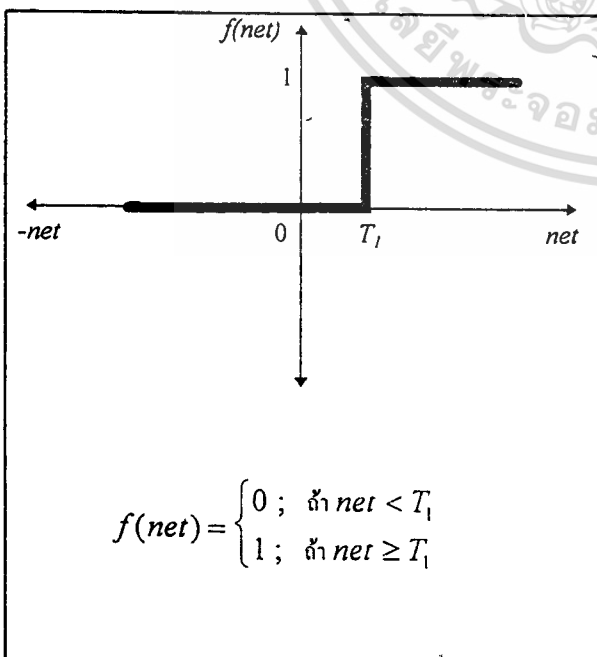
$$o = f(W'X) \quad (4.2)$$

โดยที่ W เป็นเวกเตอร์ของค่าถ่วงน้ำหนักซึ่งสามารถกำหนดได้ดังนี้

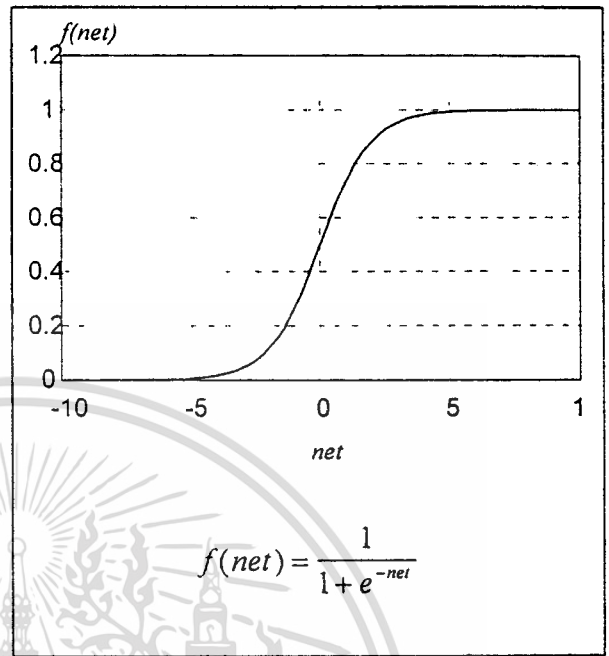
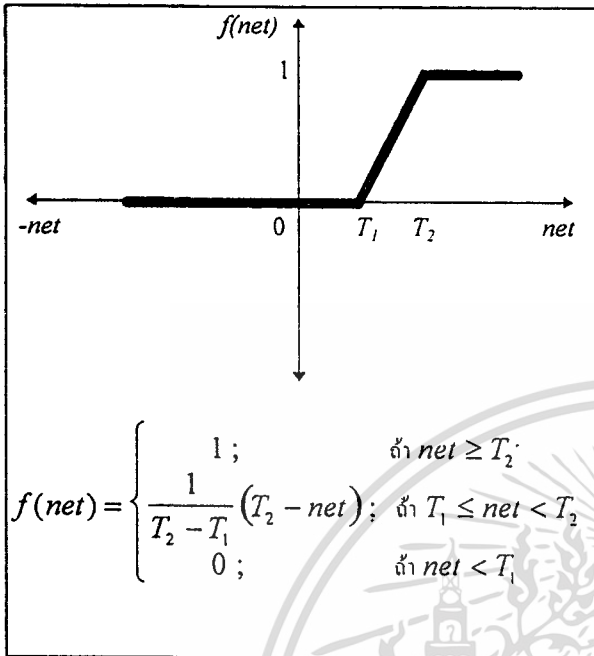
$$W \equiv [w_1, w_2, w_3, \dots, w_n]'$$

และ X เป็นเวกเตอร์อินพุต $X \equiv [x_1, x_2, x_3, \dots, x_n]'$ เมื่อ f เป็นตัวดำเนินการทรานสโพสค์ของเมทริกซ์ฟังก์ชันกำหนดสัญญาณเอาต์พุตในสมการที่ (4.2) ถูกเรียกว่าฟังก์ชันการเร่งเร้าหรือแอคทีเวชันฟังก์ชัน (Activation Function) [55-58] ซึ่งมีคุณสมบัติคล้ายกับกราฟของศักย์ไฟฟ้าขณะทำงานในรูปที่ 4.3 Activation Function มีด้วยกันสองชนิดคือ ชนิดที่เป็นเชิงเส้นและชนิดที่ไม่เป็นเชิงเส้นดังแสดงในรูปที่ 4.8

กำหนดให้ $net = W'X = \sum_{i=1}^n w_i x_i$ และ T_1, T_2 เป็นค่าเทรชโฮลด์

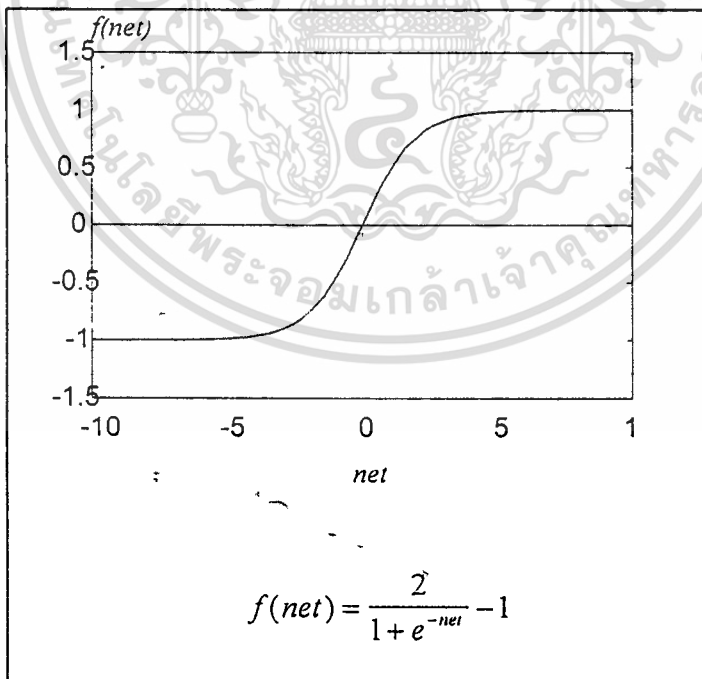


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
(ก) (ข)
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก)

(ง)



(จ)

รูปที่ 4.8 Activation Function แบบชนิดต่างๆ (ก) Threshold Logic (ง) Bipolar Binary (ค) Linear

Threshold (จ) Sigmoid (จ) Bipolar Continuous

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับ Activation Function ในรูปที่ 4.8 ที่เป็นแบบไม่ต่อเนื่อง (รูปที่ 4.8ก - ค.) และแบบต่อเนื่อง (รูปที่ 4.8ง - จ.) ค่าที่ได้จากฟังก์ชันเหล่านี้มีสองลักษณะ [53] คือ สองสภาพขั้วหรือไบโพลาร์ (Bipolar) ซึ่งโดเมนของ $f(net)$ จะอยู่ในช่วงปิด $[-1, 1]$ และหนึ่งสภาพขั้วหรือยูนิโพลาร์ (Unipolar) ซึ่งโดเมนของ $f(net)$ จะอยู่ในช่วงปิด $[0, 1]$ การที่จะกำหนดว่า Activation Function ใดคุณสมบัติที่สุดคงเป็นไปได้ เนื่องจาก ANNS ไม่มีโครงข่ายที่แน่นอน นั่นคือโครงข่ายหนึ่งจะใช้ได้กับปัญหาใดปัญหาหนึ่งเท่านั้น ดังนั้นการเลือก Activation Function ใดจะต้องพิจารณาให้เหมาะสมกับปัญหานั้นๆ

4.3 การเรียนรู้ของโครงข่าย

การเรียนรู้ของ ANNS จะมีประสิทธิภาพเพียงใดนั้นขึ้นอยู่กับค่าถ่วงน้ำหนักของโครงข่าย ซึ่งการสอน (Training) โครงข่ายก็คือการหาค่าถ่วงน้ำหนักที่เหมาะสมให้แก่โครงข่ายนั้นๆ วิธีการสอน ANNS มีอยู่สองแบบด้วยกันคือ

1. การสอนแบบชี้นำหรือดูแล (Supervised Learning) การสอนโดยวิธีนี้จะกำหนดเซตของการสอนให้กับโครงข่ายซึ่งเซตนี้ประกอบด้วยอินพุตและเอาต์พุตที่ต้องการ (Output Desired) เมื่อป้อนอินพุตให้กับโครงข่ายๆ ก็จะทำให้การประมวลผลจนได้คำตอบและค่าถ่วงน้ำหนักออกมาชุดหนึ่ง สำหรับคำตอบที่ได้จากโครงข่ายจะถูกนำมาคำนวณค่าความผิดพลาดโดยวัดเป็นระยะทางว่ามีความห่างจากคำตอบที่ต้องการของอินพุตในชุดเดียวกันมากน้อยเพียงใด ถ้ายังมีความผิดพลาดสูงอยู่ก็จะมีค่าปรับค่าถ่วงน้ำหนัก และทำการสอนต่อไปจนกว่าค่าความผิดพลาดระหว่างคำตอบของโครงข่ายกับเอาต์พุตที่ต้องการมีค่าน้อยพอที่จะยอมรับได้จึงจะหยุดการสอน และค่าถ่วงน้ำหนักที่ได้ก็จะเป็นเหมือนฟังก์ชันที่ใช้ในการแปลงข้อมูล

2. การสอนแบบไม่มีการชี้นำหรือไม่มีการดูแล (Unsupervised Learning) การสอนโดยวิธีนี้จะป้อนอินพุตเข้าสู่โครงข่าย และภายในโครงข่ายจะมีเอาต์พุตโหนดอยู่หลายโหนดด้วยกัน โดยแต่ละโหนดจะแทนกลุ่มของข้อมูลที่มีคุณสมบัติเหมือนกัน เมื่อป้อนอินพุตเข้าสู่โครงข่ายๆ จะคำนวณค่าความสัมพันธ์ที่มีอยู่ภายในเซตของอินพุต โดยอาศัยค่าถ่วงน้ำหนักเป็นตัวแยกความแตกต่างของอินพุตไปเก็บไว้ในโหนดเอาต์พุตของโครงข่าย การสอนโดยวิธีนี้จะไม่สามารถระบุได้ว่าเอาต์พุตโหนดใดเป็นของข้อมูลกลุ่มไหน ผู้ใช้จะต้องกำหนดเอง ซึ่งต่างจากการสอนแบบชี้นำที่โครงข่ายสามารถระบุกลุ่มของเอาต์พุตได้อย่างแน่นอน

การสอนโครงข่ายเซลล์ประสาทแบบจำลอง เป็นการหาฟังก์ชันการแปลงและฟังก์ชันการแปลงที่ได้จะมีคุณสมบัติไม่เป็นเชิงเส้น ซึ่งฟังก์ชันการแปลงของ ANN ในที่นี้คือเซตของค่าถ่วงน้ำหนักของโครงข่าย ดังนั้นฟังก์ชันการแปลงจะมีศักยภาพมากน้อยเพียงใดนั้นจะขึ้นอยู่กับค่าถ่วงน้ำหนักของโครงข่ายนั้นๆ ว่ามีเสถียรภาพมากน้อยเพียงใด และค่าถ่วงน้ำหนักคำนวณได้จากการสอนโครงข่าย ซึ่งการสอนโครงข่ายมีหลายแบบด้วยกันเช่น กฎการสอนของ Hebb, กฎการสอนแบบ Perceptron ของ Rosenblatt, กฎการสอนแบบเดลด้า, กฎการสอนของ Widrow-Hoff, กฎการสอนโดยใช้สหสัมพันธ์, กฎการสอนแบบ Winner-Take-All, และกฎการสอนแบบ Outstar ของ Grossberg ซึ่งกฎการสอนเหล่านี้ได้กล่าวไว้อย่างละเอียดใน [53, 57, 58] และในตารางที่ 4.1 ได้สรุปกฎการสอนโครงข่ายของวิธีต่างๆ ไว้

ตารางที่ 4.1 แสดงการปรับค่าถ่วงน้ำหนักสำหรับกฎการสอนแบบต่างๆ

กฎการสอน	การปรับค่าถ่วงน้ำหนัก	ค่าเริ่มต้นของ ค่าถ่วงน้ำหนัก	การ สอน	Activation Function
Hebb	$\Delta w_{ij} = cf(W_i' X)x_j$	0	U	Any
Preceptron	$\Delta w_{ij} = c(d_i - f(W_i' X))x_j$	จำนวนจริงใดๆ	S	Binary Bipolar or Binary Unipolar
Delta	$\Delta w_i = c(d_i - o_i)f'(W_i' X)X$	จำนวนจริงใดๆ	S	ฟังก์ชันต่อเนื่อง
Widrow-Hoff	$\Delta w_{ij} = c(d_i - W_i' X)x_j$	จำนวนจริงใดๆ	S	Any
Correlation	$\Delta w_{ij} = cd_i x_j$	0	S	Any
Winner-Take-All	$\Delta w_{mj} = c(x_j - w_{mj})$ m : โหนดที่เป็น Winner	Random Normalized	U	ฟังก์ชันต่อเนื่อง
Outstar	$\Delta w_{ij} = c(d_i - w_{ij})$	0	S	ฟังก์ชันต่อเนื่อง

c เป็นค่าคงที่ของอัตราการเรียนรู้ [56-58]

S : การสอนแบบชี้หน้า, U : การสอนแบบไม่มีการชี้หน้า

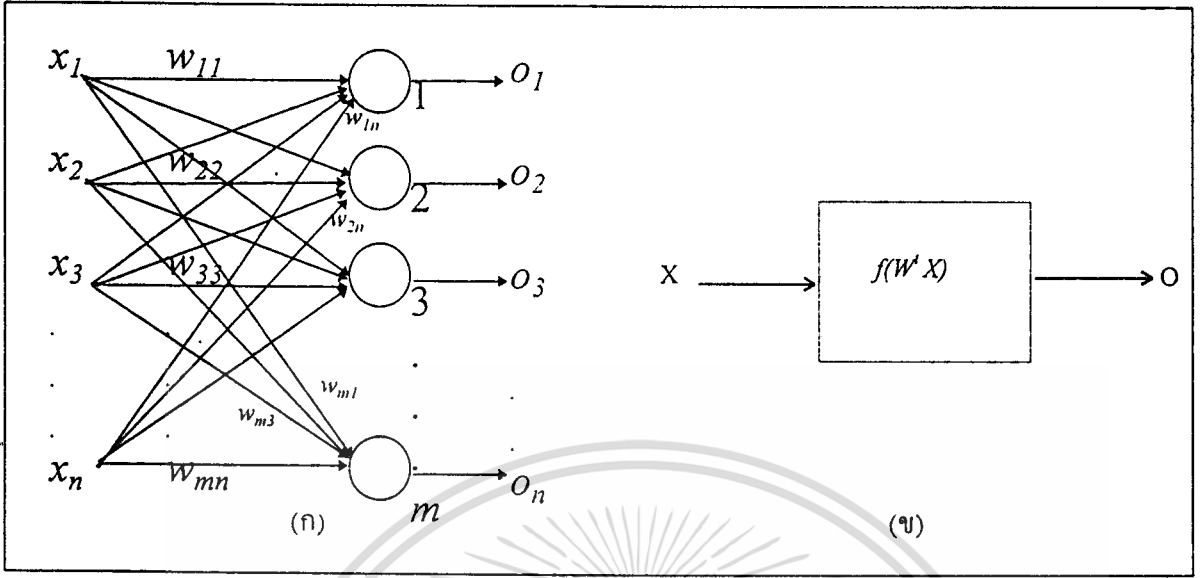
Δw_{ij} : ค่าถ่วงน้ำหนักที่ถูกปรับค่าและมีการเชื่อมต่อกันระหว่างอินพุตโหนดที่ i และ เอาท์พุทโหนดที่ j

4.4 แบบจำลองของ ANNs

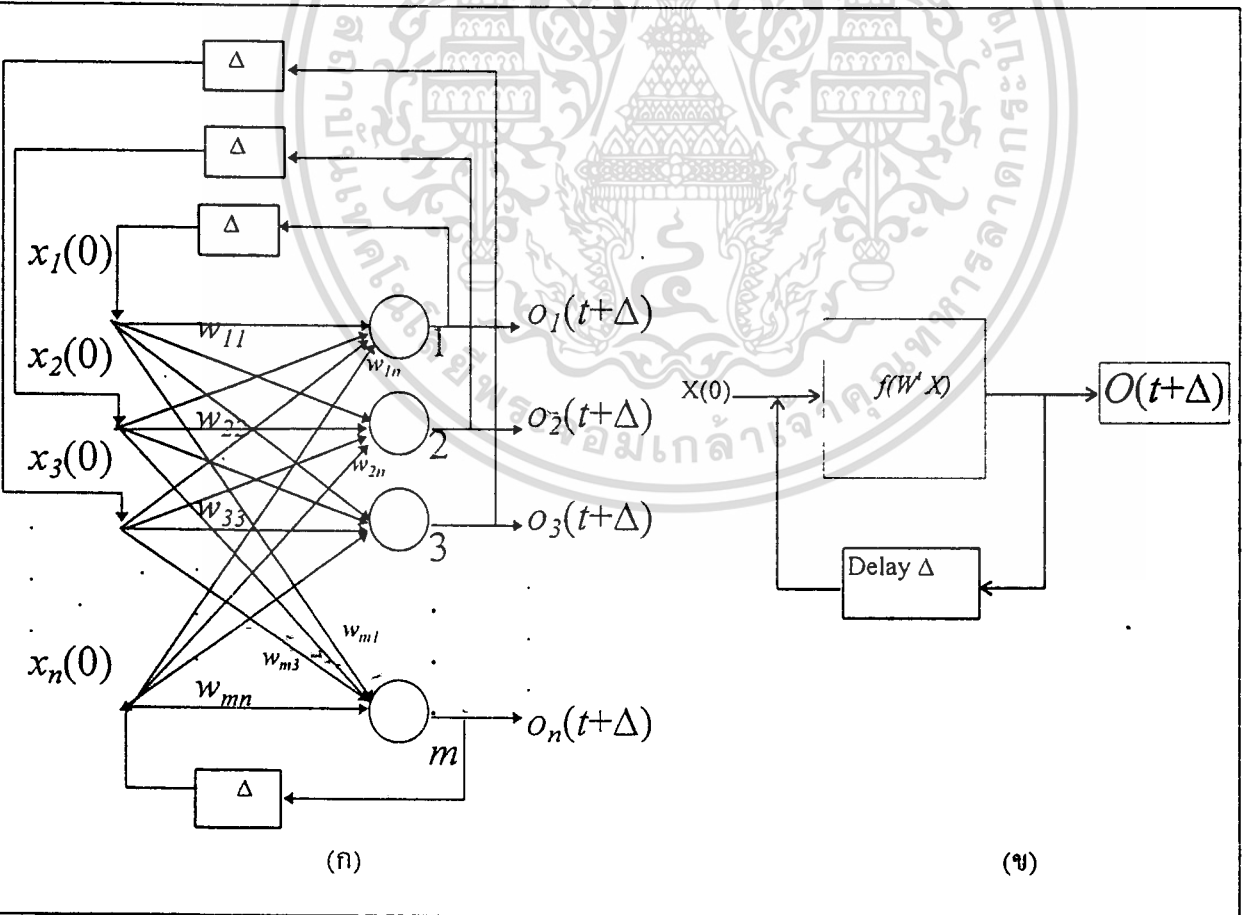
จากที่ได้กล่าวมาแล้วเกี่ยวกับโครงข่ายเซลล์ประสาท ทั้งที่เป็นแบบจำลองและเป็นโครงข่ายของเซลล์ประสาทจริงของมนุษย์ จะมีการเชื่อมต่อกันของโหนดในลักษณะของโครงข่ายอย่างหนาแน่น เพื่อให้โครงข่ายสามารถเรียนรู้และสามารถจดจำสิ่งที่เรียนรู้มาแล้วได้ ซึ่งการเชื่อมโยงของโครงข่ายจะมีสองลักษณะด้วยกันคือ

1. โครงข่ายที่ส่งสัญญาณไปข้างหน้า (Feedforward Networks) โครงข่ายชนิดนี้จะประกอบด้วยชั้นต่างๆ ของโครงข่ายโดยชั้นแรกจะเป็นอินพุตและชั้นสุดท้ายเป็นชั้นของเอาท์พุท ส่วนระหว่างชั้นอินพุตกับเอาท์พุทอาจจะมีหรือไม่มีชั้นที่แทรกอยู่ภายในอีกก็ได้ ซึ่งจะขึ้นอยู่กับอัลกอริทึมที่ใช้ในการสอนโครงข่าย เช่นถ้าเป็นโครงข่าย Perceptron แบบหลายชั้น (Multilayer Perceptron) ก็จะมีชั้นที่อยู่ระหว่างอินพุตกับเอาท์พุทอีก ซึ่งอาจจะมีมากกว่าหนึ่งชั้นก็ได้ ส่วนโครงข่าย Self-Organizing Map ของ Kohonen [60] จะมีเพียงชั้นของอินพุตกับเอาท์พุทเท่านั้น การเชื่อมต่อกันระหว่างชั้นของโครงข่ายแบบ Feedforward จะมีค่าน้ำหนักเป็นตัวเชื่อม และสัญญาณอินพุตที่เข้ามาจะถูกส่งไปตามทิศทางของลูกศรจนถึงชั้นของเอาท์พุทโดยไม่มีมีการป้อนกลับดังรูปที่ 4.9

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 โครงข่าย Feedforward แบบชั้นเดียว (ก) ลักษณะการเชื่อมโยง (ข) บล็อกไดอะแกรม

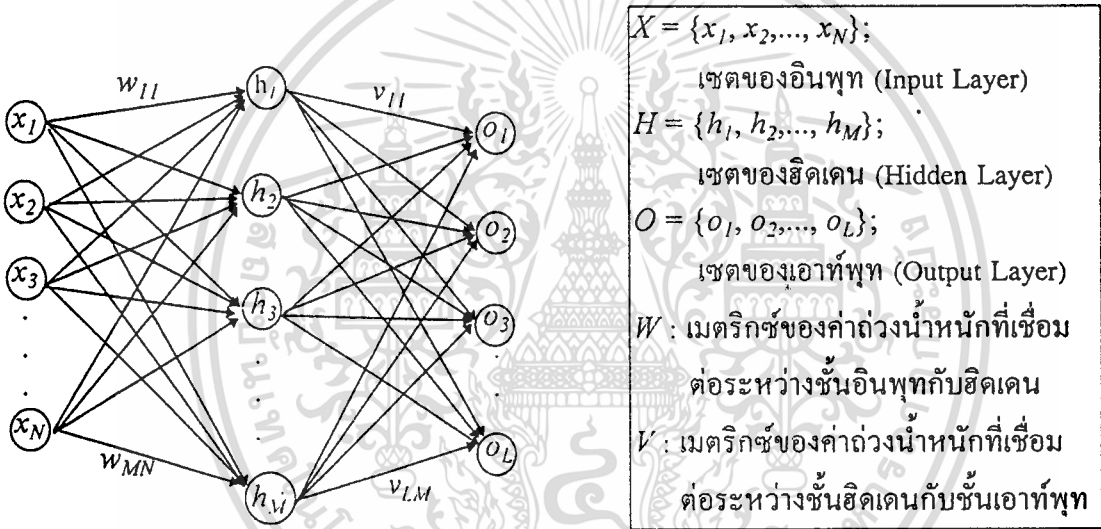


รูปที่ 4.10 โครงข่ายป้อนกลับแบบเวลาไม่ต่อเนื่องชั้นเดียว (ก) ลักษณะการเชื่อมโยง (ข) บล็อกไดอะแกรม
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. โครงข่ายที่มีการป้อนกลับ (Feedback Networks) ในส่วนแรกของโครงข่ายนี้จะเป็นโครงข่าย Feedforward เหมือนกับแบบแรก และส่วนที่เพิ่มเข้ามาคือส่วนของการป้อนกลับดังแสดงในรูปที่ 4.10 และการป้อนกลับจะมีการหน่วงเวลาไปจากเวลาเดิมเท่ากับ Δ ซึ่งโครงข่ายในรูปที่ 4.10 จะเรียกว่า โครงข่าย รีเคอร์เรนท์ (Recurrent Networks) [53]

4.5 การแพร่กระจายกลับ

การแพร่กระจายกลับหรือแบคโพรพาเกชัน (Back-propagation) เป็นขั้นตอนที่ใช้สอนโครงข่ายแบบ Multilayer Perceptron ซึ่งเป็นแบบจำลองโครงข่ายเซลล์ประสาทที่มีการเชื่อมโยงกันเป็นโครงข่ายแบบเป็นชั้นๆ ดังในรูปที่ 4.11

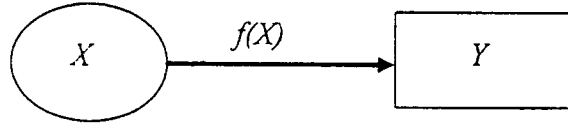


รูปที่ 4.11 โครงข่าย Multilayer Perceptron ที่มี 3 ชั้น

ซึ่งโครงข่ายนี้มีการเชื่อมโยงกัน 3 ชั้น ซึ่งประกอบด้วยชั้นของอินพุตซึ่งมีเซลล์ประสาทอยู่ N โหนด ถัดมาเป็นชั้นของฮิดเดนหรือชั้นภายใน (Hidden Layer) ซึ่งประกอบด้วยโหนดต่างๆ จำนวน M โหนด และสุดท้ายคือชั้นของเอาต์พุตซึ่งมีโหนดต่างๆ อยู่ L โหนด โครงข่ายแบบ Multilayer Perceptron ในรูปที่ 4.11 แต่ละโหนดในชั้นเดียวกันจะไม่มี การเชื่อมต่อกัน การเชื่อมโยงกันจะมีเฉพาะระหว่างชั้นเท่านั้น และการเชื่อมโยงนี้จะต่อถึงกันทุกโหนด โครงข่ายแบบ Multilayer Perceptron ไม่จำเป็นต้องมีสามชั้นเหมือนในรูปที่ 4.11 อาจจะมีจำนวนชั้นมากกว่านี้ก็ได้ซึ่งอาจมีสี่ชั้น [61] โดยการเพิ่มชั้นฮิดเดนเข้าไปอีกหนึ่งชั้น หรือถ้าต้องการจำนวนชั้นมากกว่านี้ก็สามารถทำได้โดยการเพิ่มชั้นของฮิดเดน

ชั้นฮิดเดน [53, 58, 60] เป็นตัวเพิ่มความสามารถให้แก่โครงข่ายเซลล์ประสาท โครงข่ายแบบ Multilayer Perceptron ถ้าไม่มีชั้นฮิดเดนก็จะกลายเป็นโครงข่ายแบบ Perceptron อย่างเช่นโครงข่ายแบบ Perceptron ที่เสนอโดย Rosenblatt ซึ่ง Minsky และ Papert ได้แสดงให้เห็นว่าโครงข่ายเซลล์ประสาท

จำลองนี้เป็นโครงข่ายที่มีความสัมพันธ์ระหว่างอินพุตกับเอาต์พุตในลักษณะเชิงเส้นซึ่งไม่สามารถจะแก้ปัญหาเอ็กคลูซีฟสออร์ (Exclusive OR : XOR) ได้ [60] ในกรณีที่ไม่มีชั้นฮิดเดนความสัมพันธ์ของฟังก์ชันการแปลงดังแสดงในตัวอย่างข้างล่างนี้

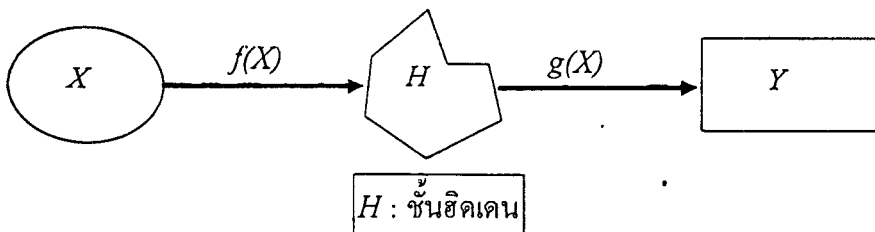


รูปที่ 4.12 แสดงความสัมพันธ์ระหว่างตัวแปร X กับ Y

จากรูปที่ 4.12 เมื่อ X เป็นตัวแปรอิสระ Y เป็นตัวแปรตามและ f(X) เป็นฟังก์ชันการแปลงจาก X ไปยัง Y ในกรณีที่ X เป็นการเป็นอิสระต่อกันแบบเชิงเส้น (Linearly Independent) เราสามารถคำนวณฟังก์ชันการแปลง f(X) ได้ ตัวอย่างเช่น ถ้า $\alpha_1 = \{0, 1\}$ และ $\alpha_2 = \{1, 0\}$ เมื่อ $X = [\alpha_1, \alpha_2]$ และ $Y' = [1, 1]$ ดังนั้นเราสามารถคำนวณฟังก์ชันการแปลงได้ดังนี้

$$X \cdot f(X) = Y \rightarrow \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

เมื่อฟังก์ชันการแปลง $f(X) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ แต่ถ้าตัวแปร X ไม่เป็นอิสระต่อกันแบบเชิงเส้น (Linearly Dependent) เราไม่สามารถจะกำหนดฟังก์ชันการแปลง f(X) ได้ เช่นถ้า $\alpha_1 = \{0, 0\}$ และ $\alpha_2 = \{1, 1\}$ เมื่อ $X = [\alpha_1, \alpha_2]$ และ $Y' = [1, 1]$ จะได้ว่า $X = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$ ซึ่งในกรณีนี้เราไม่สามารถคำนวณฟังก์ชัน f(X) ที่สอดคล้องกับ Y ได้ และในกรณีนี้จะเหมือนกับปัญหา XOR แต่ถ้าเพิ่มชั้นฮิดเดนเข้าไปในรูปที่ 4.12 จะทำให้สามารถแก้ปัญหาที่เป็น Linearly Dependent นี้ได้ [53, 60] (ดูรูปที่ 4.13)



รูปที่ 4.13 แสดงความสัมพันธ์ระหว่างตัวแปร X กับ Y โดยมีชั้นฮิดเดนเป็นตัวเชื่อม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการหาความสัมพันธ์จาก X ไปยัง Y โดยใช้ชั้นของฮิดเดนในทาง ANN จะทำให้ฟังก์ชัน $f(X)$ และ $g(X)$ ไม่เป็นเชิงเส้นซึ่ง $f(X)$ และ $g(X)$ ในที่นี้ก็คือค่าถ่วงน้ำหนักที่ได้จากการสอนโครงข่ายนั่นเอง การที่จะกำหนดชั้นของฮิดเดนว่าในโครงข่ายหนึ่งๆ ควรมีฮิดเดนกี่ชั้น และแต่ละชั้นประกอบด้วยกี่โหนด นั้นไม่มีกฎเกณฑ์หรือทฤษฎีที่แน่นอน ดังนั้นการกำหนดจำนวนชั้นและจำนวนโหนดของแต่ละชั้นของฮิดเดนสามารถทำได้โดยการทดลอง

วิธีการของกฎเดลต้าเอนกประสงค์

กฎเดลต้า (Delta Rule) ถูกพัฒนาขึ้นมาเพื่อใช้สอนโครงข่ายเซลล์ประสาทจำลอง [58] ซึ่งครั้งแรกใช้ในการสอนโครงข่ายเซลล์ประสาทจำลอง Perceptron ซึ่งจะเป็นกฎการสอน Perceptron แบบต่อเนื่อง และต่อมาได้พัฒนาการสอนเดลต้าให้ใช้ได้กับโครงข่ายที่มีการเชื่อมต่อกันหลายชั้นอย่าง Multilayer Perceptron จึงเรียกกฎเดลต้าที่ถูกพัฒนาขึ้นใหม่นี้ว่า กฎเดลต้าเอนกประสงค์ (Generalized Delta Rule: GDR)

ขั้นตอนการสอนโครงข่าย Multilayer Perceptron จะเรียกว่า ขั้นตอนการสอนแบบค่าความผิดพลาดแพร่กระจายกลับหรือ Error Back-propagation ซึ่งเป็นการแพร่กระจายกลับของค่าความผิดพลาดที่เกิดขึ้นในชั้นเอาต์พุตที่ต้องการกับเอาต์พุตที่คำนวณได้ โดยคำนวณย้อนกลับจากชั้นเอาต์พุตผ่านชั้นฮิดเดนตลอดมาจนถึงชั้นอินพุต เพื่อทำการปรับค่าถ่วงน้ำหนัก จากรูปที่ 4.11 เมื่อทำการสอนโครงข่ายนี้ ซึ่งมีการสอนแบบซึ้นาดังนั้นในการสอนจะต้องป้อนเซตของข้อมูลที่จะใช้สอนซึ่งประกอบด้วยเซตของอินพุต และเซตของเอาต์พุตที่ต้องการซึ่งจะสอดคล้องกับเซตของอินพุต กำหนดให้ P เป็นจำนวนเซตทั้งหมดที่ใช้ในการสอนโครงข่าย ดังนั้นเซตของอินพุตและเอาต์พุตที่ต้องการทั้งจะมีอยู่ P เซต, ถ้าให้ D เป็นเซตของเอาต์พุตที่ต้องการจะได้ว่า $D_p = \{d_1, d_2, d_3, \dots, d_L\}$, เมื่อ $p = 1, 2, 3, \dots, P$

W เป็นเมตริกซ์ของค่าถ่วงน้ำหนักที่เชื่อมโยงระหว่างชั้นของอินพุตกับชั้นฮิดเดนซึ่ง W มีขนาด M แถว และ N คอลัมน์

$$W = \begin{bmatrix} w_{11} & w_{12} & w_{13} & \dots & w_{1N} \\ w_{21} & w_{22} & w_{23} & \dots & w_{2N} \\ w_{31} & w_{32} & w_{33} & \dots & w_{3N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_{M1} & w_{M2} & w_{M3} & \dots & w_{MN} \end{bmatrix}$$

V เป็นเมตริกซ์ของค่าถ่วงน้ำหนักที่เชื่อมโยงระหว่างชั้นฮิดเดนไปยังชั้นเอาต์พุต ซึ่งมีขนาด L แถว และ M คอลัมน์

$$V = \begin{bmatrix} v_{11} & v_{12} & v_{13} & \dots & v_{1M} \\ v_{21} & v_{22} & v_{23} & \dots & v_{2M} \\ v_{31} & v_{32} & v_{33} & \dots & v_{3M} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ v_{L1} & v_{L2} & v_{L3} & \dots & v_{LM} \end{bmatrix}$$

และ $f(net)$ เป็น Activation Function ใดๆ ดังที่แสดงในรูปที่ 4.8

เมื่อทำการป้อนเซตของข้อมูลที่ใส่สอนเข้าสู่โครงข่ายจะสามารถคำนวณ โหนดของฮิดเดนที่ m ได้ดังนี้

$$\begin{aligned} net_m &= \sum_{n=1}^N w_{mn} x_n \\ h_m &= f(net_m) \end{aligned} \quad (4.3)$$

และสามารถคำนวณเอาต์พุตโหนดที่ l ได้ดังนี้

$$\begin{aligned} net_l &= \sum_{m=1}^M v_{lm} x_m \\ o_l &= f(net_l) \end{aligned} \quad (4.4)$$

เมื่อคำนวณชั้นของเอาต์พุตครบทุกโหนดแล้ว ขั้นตอนต่อไปเป็นการปรับค่าถ่วงน้ำหนักของโครงข่าย โดยใช้ค่าผลรวมของค่าความคลาดเคลื่อนกำลังสอง ระหว่างค่าของผลลัพธ์ที่ได้จากโครงข่ายกับค่าเอาต์พุตที่ต้องการของแพทเทิร์นที่ p ซึ่งสามารถคำนวณได้ดังนี้

$$E_p = \frac{1}{2} \sum_{l=1}^L (o_{pl} - d_{pl})^2 \quad (4.5)$$

ดังนั้นในการปรับค่าถ่วงน้ำหนักที่เชื่อมต่อระหว่างชั้นเอาต์พุตกับชั้นฮิดเดนสำหรับแพทเทิร์นที่ p (H_p) สามารถคำนวณได้ดังนี้

$$\begin{aligned} \Delta V &= -\eta \frac{\partial E_p}{\partial V} \\ &= \alpha V + \eta \delta_p^l H_p \end{aligned} \quad (4.6)$$

โดยที่ $\delta_p^l = (D_p - O_p) O_p (1 - O_p)$ ซึ่งเป็นความคลาดเคลื่อนภายในชั้นของเอาต์พุต

α เป็นค่าโมเมนตัม [53, 58]

เอกสารนี้เป็นเอกสารประกอบการเรียนรู้ (Learning Rate) ซึ่งเป็นค่าคงที่ [53, 58] หน้าไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และการปรับค่าถ่วงน้ำหนักที่เชื่อมต่อระหว่างชั้นฮิดเดนกับชั้นอินพุตสามารถคำนวณได้ดังนี้

$$\Delta W = \alpha IV + \eta \delta_p'' X_p \quad (4.7)$$

โดยที่ $\delta_p'' = H_p(1-H_p)(\delta_p' V)$ ซึ่งเป็นค่าความคลาดเคลื่อนภายในชั้นฮิดเดน สำหรับ α และ η จะมีคุณสมบัติเหมือนกับสมการที่ (4.6)

ในกรณีที่มีโครงข่ายมี I ชั้นสามารถปรับค่าถ่วงน้ำหนักโดยวิธีค่าความผิดพลาดแพร่กระจายกลับได้ดังนี้

$$\Delta W' = \alpha W' + \eta \delta_p' X_p'^{-1} \quad (4.8)$$

เมื่อ W^i เป็นเมตริกซ์ค่าถ่วงน้ำหนักที่อยู่ระหว่างชั้นที่ i กับ $i-1$ ($i = 1, 2, 3, \dots, I$), $X_p'^{-1}$ เป็นเซตของอินพุตแพทเทิร์นที่ p จากชั้นที่ $i-1$ และ δ_p' เป็นเวกเตอร์ของความคลาดเคลื่อนสำหรับชั้นที่ i สำหรับชั้นเอาต์พุตหรือชั้นที่ I สามารถคำนวณได้ดังนี้

$$\delta_p' = (D_p - O_p') O_p' (1 - O_p') \quad (4.9)$$

โดยที่ O_p' คือชั้นของเอาต์พุต และสำหรับความคลาดเคลื่อนของชั้นที่ i ใดๆ โดยที่ $i \neq I$ สามารถคำนวณได้ดังนี้

$$\delta_p' = O_p' (1 - O_p') (\delta_p^{i+1} W^{i+1}) \quad (4.10)$$

โดยที่ O_p' คือเอาต์พุตชั้นที่ i แพทเทิร์นที่ p และสำหรับ δ_p^{i+1} และ W^{i+1} เป็นค่าความคลาดเคลื่อนและเมตริกซ์ค่าถ่วงน้ำหนักของชั้นถัดไป

ปัจจัยที่ทำให้ประสิทธิภาพการสอนโครงข่ายด้วยวิธี Back-propagation เพิ่มมากขึ้นมีดังนี้

- การกำหนดค่าเริ่มต้นให้กับเมตริกซ์ถ่วงน้ำหนัก

ก่อนที่จะทำการสอนโครงข่าย Multilayer Perceptron โดยวิธี Error Back-propagation จำเป็นต้องกำหนดค่าเริ่มต้นให้กับเมตริกซ์ค่าถ่วงน้ำหนักที่เชื่อมโยงระหว่างชั้นทุกชั้น โดยค่านี้จะเป็นเลขจำนวนจริงที่มีค่าน้อยๆ ที่ได้มาจากการสุ่ม ค่าเริ่มต้นของเมตริกซ์เชื่อมโยงจะมีผลต่อเวลาที่ใช้ในการสอนและอาจจะส่งผลถึงค่าถ่วงน้ำหนักหลังจากที่ได้สอนโครงข่ายไปแล้วว่าจะมีศักยภาพมากน้อยเพียงใดในการแปลงข้อมูลอินพุตไปสู่เอาต์พุต ในอัลกอริทึมที่ 4-1 แสดงขั้นตอนการกำหนดค่าเริ่มต้นให้กับเมตริกซ์ค่าถ่วงน้ำหนัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น มิอนุญาตให้นำไปใช้ประโยชน์ในการค้า
 อลกอริทึมที่ 4-1 ขั้นตอนการกำหนดค่าเริ่มต้นให้กับเมตริกซ์ค่าถ่วงน้ำหนัก
 เมื่อกำหนดค่าเริ่มต้นให้กับเมตริกซ์ค่าถ่วงน้ำหนักแล้ว จะต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดให้
 ROW เป็นจำนวนแถวของเมตริกซ์ถ่วงน้ำหนัก
 COL เป็นจำนวนคอลัมน์ของเมตริกซ์ถ่วงน้ำหนัก
 W เป็นเมตริกซ์ของค่าถ่วงน้ำหนักที่เชื่อมโยงระหว่างชั้นใดๆ

```
void Initial_Weights (Matrix W, int ROW, int COL)
{
  for (int i = 0; i < ROW; i++)
    for (int j = 0; j < COL; j++)
      W[i][j] = 1 - random (101) / 53.0;
}
```

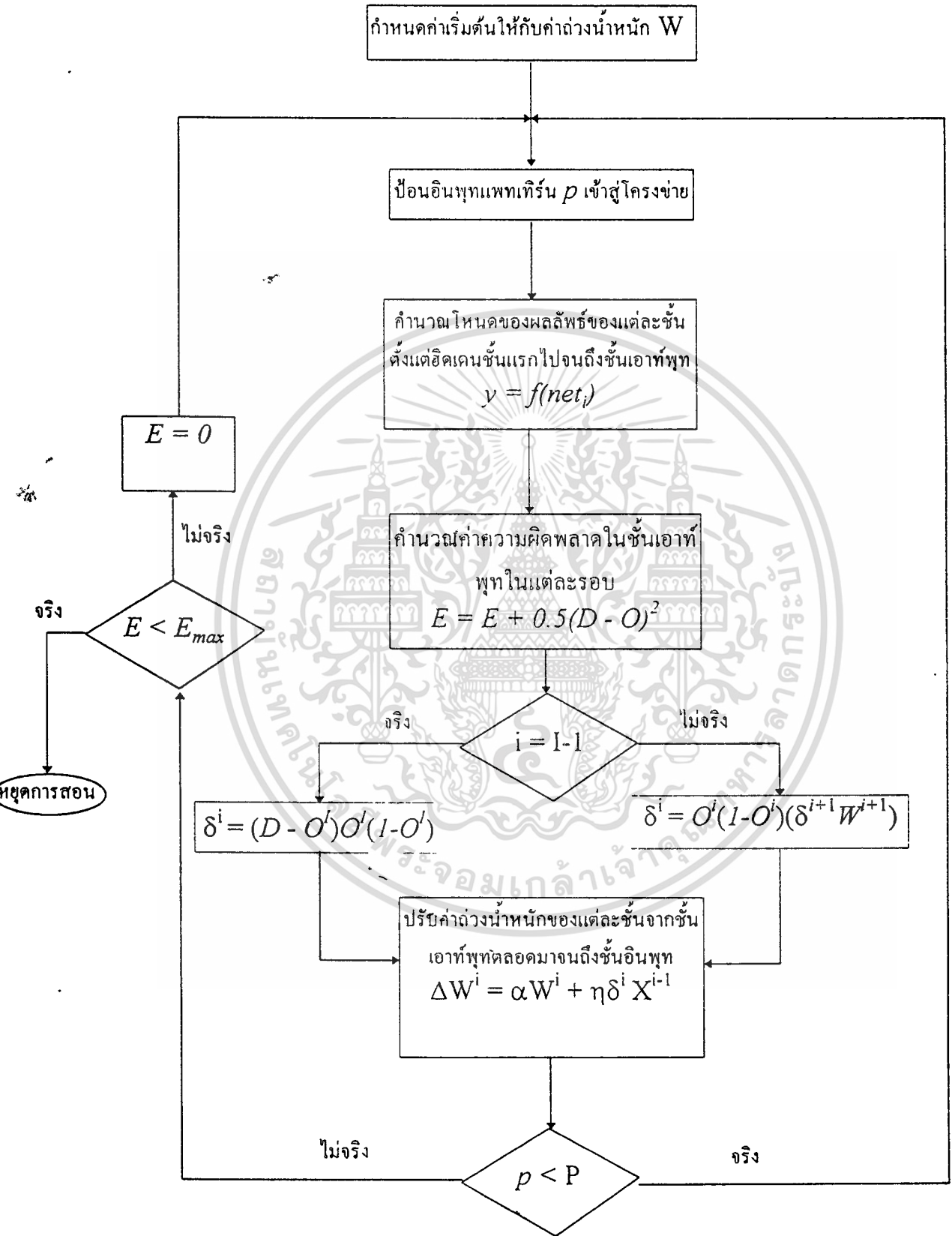
- อัตราเร็วในการเรียนรู้ (Learn Rate : η) ซึ่งโดยทั่วไปแล้วค่านี้จะเป็นค่าคงที่ที่มีค่าอยู่ระหว่าง 0.05 - 0.25 [60] แต่มีงานวิจัยบางส่วนที่ค่าอัตราเร็วในการเรียนรู้มีการปรับค่าได้เช่น กำหนดให้อัตราเร็วในการเรียนรู้ผันแปรตามจำนวนรอบที่สอน [62], อัตราเร็วในการเรียนรู้ผันแปรตามค่าความผิดพลาดเฉลี่ยของระบบ [63] และ ค่าอัตราเร็วในการเรียนรู้ผันแปรตามค่าความแปรปรวนของแต่ละแพทเทิร์น [61]

- โมเมนตัม (Momentum : α) สำหรับกฎการสอนแบบเจเนอรัลไลซ์เดลต้า โดยทั่วไปค่าโมเมนตัมอาจจะใช้หรือไม่ใช้ก็ได้ แต่ค่านี้จะมีคุณสมบัติช่วยป้องกันการแกว่ง (Oscillate) ของระบบ [57-58] โดยค่านี้จะสัมพันธ์กับค่าอัตราเร็วในการเรียนรู้ คือถ้าอัตราเร็วในการเรียนรู้มีค่ามากแต่ค่าโมเมนตัมมีค่าน้อยจะทำให้ระบบโครงข่ายเกิดการแกว่ง ส่วนกรณีอื่นๆ โครงข่ายจะไม่เกิดการแกว่งแต่จะมีผลต่อเวลาที่ใช้ในการสอน

- ไบแอส (Bias) โหนดไบแอสสำหรับโครงข่ายโดยทั่วไปอาจจะมีหรือไม่ก็ได้ ในกรณีของโครงข่ายที่ใช้ในระบบที่เป็น Real-Time แล้วจะไม่ค่อยมีการใช้โหนดไบแอสเนื่องจากจะเสียเวลาในการคำนวณ และถ้ามีการออกแบบโครงข่ายอย่างเหมาะสมแล้วทอมไบแอสไม่จำเป็นต้องมี แต่ถ้าโครงข่ายมีจำนวนโหนดไม่เหมาะสมคือมีจำนวนโหนดน้อยไป การเพิ่มโหนดไบแอสเข้าไปจะทำให้โครงข่ายสามารถจดจำได้ดีขึ้น โหนดไบแอสจะถูกเพิ่มเข้าไปเป็นโหนดๆ หนึ่งในแต่ละชั้น ยกเว้นในชั้นของผลลัพธ์และโหนดไบแอสที่เพิ่มเข้าไปมักจะเป็นค่าคงที่ ผลของการเพิ่มโหนดไบแอสเข้าไปในโครงข่ายจะทำให้ประสิทธิภาพการสอนดีขึ้น [64]

ขั้นตอนการสอน

การสอนโครงข่าย Multilayer Perceptron ด้วยวิธี Back-propagation มีขั้นตอนการสอนดังอัลกอริทึมที่ 4-2 ซึ่งเป็นโครงข่ายแบบไม่มีไบแอสและในรูปที่ 4.14 เป็นแผนภูมิของขั้นตอนการสอนของอัลกอริทึมนี้ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตเห็นไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 4.14 แผนภูมิการสอนแบบ Error Back-propagation ที่มี I ชั้น
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ E_{max} เป็นค่าความผิดพลาดต่ำสุดที่พอจะยอมรับได้ ซึ่งในการสอนโครงข่ายในรอบหนึ่งๆ (Epoch) ก็คือการสอนชุดของแพทเทิร์นทั้งหมด P แพทเทิร์น ถ้าค่าความผิดพลาดของระบบ E ยังมีค่ามากกว่าค่าความผิดพลาดที่ตั้งไว้ E_{max} ก็จะนำข้อมูลชุดเดิมมาทำการสอนใหม่จนกว่าค่าความผิดพลาด E จะน้อยกว่าค่า E_{max}

อัลกอริทึมที่ 4-2 แสดงขั้นตอนการสอนโครงข่ายโดยวิธี Back-propagation

กำหนดให้

```
float ***W; /* W[i][j][k] เป็นค่าถ่วงน้ำหนักของชั้นที่ i ที่เชื่อมต่อระหว่างอินพุตโหนดที่ k
            กับเอาต์พุต j โหนด */
float **Layer; /* Layer[i][j] เป็นชั้นที่ i โหนดที่ j */
float **Delta; /* Delta[i][j] เป็นค่าความผิดพลาดของชั้นที่ i โหนดที่ j */
int *ROW; /* ROW[i] : ชั้นที่ i มีจำนวนโหนดเท่ากับ ROW[i] */
int *COL; /* COL[i] : เป็นจำนวนคอลัมน์ของเมตริกซ์ถ่วงน้ำหนักชั้นที่ i */
int I; /* จำนวนชั้นทั้งหมดของโครงข่าย */

void Set_Input_Pattern(Vector X) /* ป้อนเซตของอินพุตเข้าสู่ชั้นอินพุตของโครงข่าย */
{
    Layer[0] = X; /* ป้อนเวกเตอร์อินพุตให้กับเวกเตอร์ของชั้นอินพุต */
}

void Forward_Propagate() /* คำนวณผลลัพธ์ของแต่ละชั้น */
{
    Vector input;
    float net;

    for (int i = 0; i < (I-1); i++)
    {
        input = Layer[i];
        for (int j = 0; j < ROW[i]; j++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น $net = 0;$ อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for (int k = 0; k < COL[i]; k++)
    net = net + input[k]*W[i][j][k];

/* คำนวณ โหนดผลลัพธ์โดยใช้ฟังก์ชัน Sigmoid */
Layer[i+1][j] = 1 / (1 + exp(-net));
}

```

```

float SSE() /* คำนวณค่าความผิดพลาดของชั้นเอาต์พุต */
{
    float sse = 0;

    for (int j = 0; j < ROW[I-1]; j++)
    {
        Ep[j] = Layer[I-1][j] * (Dp[j] - Layer[I-1][j]) * (1 - Layer[I-1][j]);
        sse = sse + Ep[j] * Ep[j];
    }

    return sse;
}

```

```

void Adjust_Weights() /* ปรับค่าถ่วงน้ำหนัก */
{
    float sum;

    /* ปรับค่าถ่วงน้ำหนักในชั้นเอาต์พุต */
    for (int j = 0; j < ROW[I-1]; j++)
        for (int k = 0; k < COL[I-1]; k++)
            W[I-2][j][k] = a * W[I-2][j][k] + h * Ep[j] * Layer[I-2][k];
}

```

```

/* คำนวณค่าความผิดพลาดของชั้นที่ I-1 */

```

```

for (int j = 0; j < COL[I-1]; j++)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    sum = 0;
    for (int k = 0; k < ROW[I-1]; k++)
        sum += Ep[k] * W[I-2][k][j];
    Delta[I-2][j] = Layer[I-2][j] * (1 - Layer[I-2][j]) * sum;
}
/* ปรับค่าถ่วงน้ำหนักสำหรับชั้นที่ i ใดๆ โดยที่ i ≠ I */
for (int i = I - 3; i >= 0; i--)
{
    for (int j = 0; j < ROW[i]; j++)
        for (int k = 0; k < COL[i]; k++)
            W[i][j][k] = a * W[i][j][k] + h * Delta[i+1][j] * Layer[i][k];
    /* คำนวณค่าความผิดพลาดของชั้นที่ i */
    if (i > 0)
        for (int j = 0; j < COL[i]; j++)
        {
            sum = 0;
            for (int k = 0; k < ROW[i]; k++)
                sum += Ep[k] * W[i][k][j];
            Delta[i][j] = Layer[i][j] * (1 - Layer[i][j]) * sum;
        }
}
}

void Back_Propagation_Training()
{
    for (int i = 0; i < I; i++)                /* I เป็นจำนวนชั้นทั้งหมดของโครงข่าย */
        Initail_Weights(W[i], ROW[i], COL[i]); /* กำหนดค่าถ่วงน้ำหนัก */

    /* ถ้าค่าความผิดพลาดยังมากกว่าค่าความผิดพลาดต่ำสุดที่ตั้งเอาไว้ให้ทำการสอนโครงข่ายต่อไป */
    เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
    ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

```

while (Error > ERROR_MIN)
{
    p = random(P);          /* สุ่มแพทเทิร์นที่ p จากแพทเทิร์นที่ใช้สอนทั้งหมด P เซต */
    Set_Input_Pattern(X[p]); /* ป้อนอินพุตแพทเทิร์นที่ p เข้าสู่โครงข่าย */
    Forward_Propagate();    /* คำนวณผลลัพธ์ของแต่ละชั้น */
    error = SSE();          /* คำนวณค่าความผิดพลาด */
    Adjust_Weights();       /* ปรับค่าถ่วงน้ำหนัก */
}
}

```

เมื่อสอนโครงข่ายจนอยู่ในสถานะที่เสถียรแล้วก็สามารถนำค่าถ่วงน้ำหนักที่ได้มาหาคำตอบได้ดัง
อัลกอริทึมที่ 4-3

อัลกอริทึมที่ 4-3 ขั้นตอนการนำค่าถ่วงน้ำหนักมาใช้งาน

```

void Back_Propagation()
{
    for (int p=0; p < P; p++) /* P เป็นจำนวนแพทเทิร์นทั้งหมดของระบบ */
    {
        Set_Input_Pattern(X[p]); /* ป้อนอินพุตแพทเทิร์นที่ p เข้าสู่โครงข่าย */
        Forward_Propagate(); /* คำนวณผลลัพธ์ของแต่ละชั้น และผลลัพธ์ของระบบจะอยู่ที่
                               ชั้นที่ I (Layer[I-1] */
    }
}

```

สรุป

ANNS เป็นเครื่องมือที่ช่วยให้คอมพิวเตอร์มีความสามารถมากขึ้น (คือนอกเหนือจากการคำนวณด้วยหลักทางคณิตศาสตร์ทั่วไปและการเก็บข้อมูล) โดยเฉพาะการประมวลผลข่าวสารที่มีความยุ่งยากซับซ้อน เช่นการทำให้คอมพิวเตอร์สามารถรู้จักภาพและเข้าใจภาษาพูดของมนุษย์ ซึ่งในงานเช่นนี้จะต้องมีการตัดสินใจที่ต้องอาศัยความรู้และประสบการณ์ ซึ่งถ้าใช้เทคนิคทางคณิตศาสตร์ธรรมดาแก้ปัญหานี้ก็ทำให้มีความระบบมีความซับซ้อนมาก แต่ถ้าใช้ ANNS ก็จะช่วยลดความยุ่งยากลงได้มากทีเดียว แต่อย่าลืมว่าในปัจจุบัน ANNS เป็นเสมือนฟังก์ชันการแปลงเท่านั้น ดังนั้นเมื่อมีข้อมูลที่เข้ามาใหม่และมีการแจกแจงที่ต่างเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากเดิมมาก (ต่างจากข้อมูลที่เคยสอนไว้มาก) ระบบนี้ก็จะมีความคลาดเคลื่อนสูงเช่นกัน ซึ่งจุดอ่อนอันนี้ในปัจจุบันเป็นที่สนใจของนักวิจัยในสาขาต่างๆ ที่จะพยายามปรับปรุงโครงสร้างให้มีความสามารถมากขึ้น

แต่อย่างไรก็ตาม ANNS ก็ยังมีประโยชน์อยู่มาก เพียงแต่เราต้องเลือกปัญหาให้เหมาะสมกับระบบก็จะช่วยลดความซับซ้อนของระบบลงได้มาก อย่างเช่นการทำให้คอมพิวเตอร์จัดจำรูปแบบของตัวอักษรแต่ละตัว ที่เป็นทั้งตัวพิมพ์และลายมือเขียน ซึ่งได้รับความสำเร็จในหลายภาษาไม่ว่าจะเป็นภาษาอังกฤษ, ญี่ปุ่น, หรือ จีน และอีกไม่นานคงจะได้เห็น ANNS ที่ประยุกต์ใช้กับภาษาไทยอย่างได้ผล



บทที่ 5 ฟัชชีนิวรอลเน็ตเวิร์ค

แบบจำลองโครงข่ายเซลล์ประสาทแบบฟัชชีหรือฟัชชีนิวรอลเน็ตเวิร์ค (Fuzzy Neural Networks: FNN) ที่เสนอในบทนี้ เป็นโครงข่ายที่ใช้ประมวลผลสัญญาณฟัชชี และส่วนของการเชื่อมโยงภายในโครงข่ายก็จะเป็นฟัชชีเซตด้วย ก่อนที่เราจะพิจารณาถึงขั้นตอนการเรียนรู้ของ FNN เราจะอธิบายถึงปัญหาในการออกแบบของฟัชชีเซตและนิวรอลเน็ตเวิร์คก่อน เพื่อแสดงให้เห็นว่าการใช้ FNN มีประสิทธิภาพและความยืดหยุ่นสูงกว่าฟัชชีและนิวรอลเน็ตเวิร์คแบบธรรมดาอย่างไร

5.1 ปัญหาของการออกแบบระบบของฟัชชีเซต และนิวรอลเน็ตเวิร์ค

ในการประมาณค่าตัวแปรที่ต้องการ โดยใช้ระบบฟัชชีได้อธิบายไว้แล้วในบทที่ 2 ซึ่งค่าประมาณที่ได้จะมีความคลาดเคลื่อนน้อยมากถ้าผู้ออกแบบระบบมีความรู้ความชำนาญในระบบนั้นๆ แต่ถ้าหากว่าผู้ออกแบบไม่มีความเข้าใจในระบบที่กำลังออกแบบอยู่อย่างเพียงพอ จะส่งผลให้ค่าประมาณของตัวแปรที่ต้องการที่จะนำไปควบคุมระบบมีความคลาดเคลื่อนสูง ซึ่งทำให้ระบบขาดเสถียรภาพได้หรือระบบอาจจะไม่สามารถดำเนินการต่อไปได้ ระบบฟัชชีที่กล่าวถึงมาแล้วข้างต้นก็คือการออกแบบเทอมเซตหรือกฎแต่ละข้อว่าควรจะใช้พื้นที่ในโดเมนมากน้อยเพียงใด และฟังก์ชันที่ใช้กำหนดค่าความเป็นสมาชิกของกฎจะต้องสอดคล้องกับคุณสมบัติของตัวแปร และขั้นตอนมาที่จะต้องหาความสัมพันธ์ระหว่างตัวแปรอินพุทกับตัวแปรเอาต์พุท โดยความสัมพันธ์อันนี้จะอยู่ในรูปของตาราง FAM และขั้นตอนการออกแบบระบบฟัชชีที่สำคัญอีกขั้นหนึ่งที่ผู้ออกแบบต้องศึกษาอย่างละเอียดเช่นกันคือขั้นตอน Defuzzification เราต้องเลือกวิธีการ Defuzzifies ให้เหมาะสมกับระบบเช่น ถ้าระบบต้องการค่าตัวแปรที่มีความคลาดเคลื่อนน้อย เราต้องเลือกวิธีการ Defuzzify ที่ให้ความถูกต้องสูงเช่น COA แต่ถ้าระบบต้องการความเร็วและไม่ต้องการความแม่นยำมากนักก็อาจจะเลือกวิธีที่ความถูกต้องของค่าประมาณที่ได้น้อยลงมาแต่ใช้เวลาในการคำนวณน้อย เช่น MOM, MIN, MAX, เป็นต้น

สำหรับปัญหาที่พบในการออกแบบนิวรอลเน็ตเวิร์คก็มีความซับซ้อนไม่น้อยไปกว่าระบบฟัชชีเลยทีเดียว โดยเริ่มจากการกำหนดขนาดของโครงข่าย (ซึ่งในบทนี้จะกล่าวถึงเฉพาะโครงข่ายที่ใช้การสอนแบบ Back Propagation) ซึ่งเป็นขั้นตอนที่ยากและต้องใช้เวลาพอสมควร เนื่องจากการกำหนดขนาดของโครงข่ายยังไม่มีกฎเกณฑ์หรือทฤษฎีที่แน่นอนมาใช้เป็นเกณฑ์ในการกำหนดจำนวนชั้นของโครงข่ายและจำนวนโหนดในแต่ละชั้น ดังนั้นผู้ออกแบบเริ่มจากการกำหนดจำนวนโหนดของชั้นอินพุทและเอาต์พุทขึ้นมาก่อน แล้วจึงกำหนดชั้นฮิดเดนขึ้นมาและทดลองสอนโครงข่ายดู ถ้าโครงข่ายไม่สามารถหาความสัมพันธ์ระหว่างอินพุทกับเอาต์พุทได้ก็ทำการเพิ่มจำนวนโหนดเข้าไปในชั้นฮิดเดนหรืออาจจะเพิ่มชั้นฮิดเดนขึ้นมาอีกชั้นก็ได้ ซึ่งในขั้นตอนนี้จะต้องทดลองสอนโครงข่าย และเพิ่มจำนวนโหนดจนกระทั่งโครงข่ายสามารถหาความสัมพันธ์จากอินพุทไปยังเอาต์พุทได้ แต่ถ้าฟังก์ชันความสัมพันธ์ (ค่าถ่วงน้ำหนัก) ยังไม่อยู่ในสถานะเสถียร อาจจะแก้ปัญหานี้ได้โดยการเพิ่มโหนดไปอัสหรืออาจจะเพิ่มโหนดในชั้นของฮิดเดน ถ้าฟังก์ชันความ

สัมพันธ์อยู่ในสถานะที่เสถียรแล้ว โหนดไบอัสไม่จำเป็นต้องมีก็ได้ และปัญหาอีกอย่างหนึ่งก็คือแพทเทิร์นที่จะนำมาสอนจะต้องมีความถูกต้องสามารถเป็นตัวแทนของข้อมูลทั้งหมดได้เป็นอย่างดี เพราะถ้าแพทเทิร์นที่ใช้สอนเป็นตัวแทนข้อมูลทั้งหมดของระบบด้วยแล้ว ก็จะทำให้การสอนมีประสิทธิภาพมากขึ้นและเวลาที่ใช้ในการสอนก็น้อยลงด้วย

5.2 การลดปัญหาในการออกแบบระบบทั้งของฟัซซี่เซต และนิวรอลเน็ตเวิร์ค

หลังจากที่ได้พิจารณาถึงปัญหาที่เกิดขึ้นสำหรับการออกแบบทั้งในส่วนของระบบฟัซซี่ และนิวรอลเน็ตเวิร์คแล้ว เราจะมาพูดถึงแนวทางอันหนึ่งที่ใช้แก้ปัญหาเหล่านี้กันคือการใช้โครงข่ายฟัซซีนิวรอลซึ่งมีผู้เสนอแบบจำลองการเรียนรู้ของโครงข่ายนี้ไว้หลายวิธีด้วยกันเช่น การใช้นิวรอลเน็ตเวิร์คมาสร้างกฎฟัซซี่ [65] การใช้ตัวดำเนินการ Max และ Min ในฟัซซี่เซตแทนตัวดำเนินการบวกและคูณในนิวรอลเน็ตเวิร์ค [66-67] และแบบจำลองที่จะเสนอในที่นี้เป็นการดัดแปลงและปรับปรุงมาจากการสอนเซลล์แบบฟัซซี่ [68-69] และโครงข่ายฟัซซีนิวรอล [70] สำหรับโครงข่ายฟัซซีนิวรอลในวิทยานิพนธ์ฉบับนี้ เป็นการริเริ่มประยุกต์ใช้กับการหาขอบภาพ โดยมีลักษณะการเชื่อมโยงของโครงข่ายดังแสดงในรูปที่ 5.1

จากแผนภูมิโครงข่ายในรูปที่ 5.1 สามารถแบ่งได้เป็นสองส่วนที่มีความคาบเกี่ยวกันคือ ส่วนที่เป็นฟัซซี่เซต และส่วนที่เป็นนิวรอลเน็ตเวิร์ค

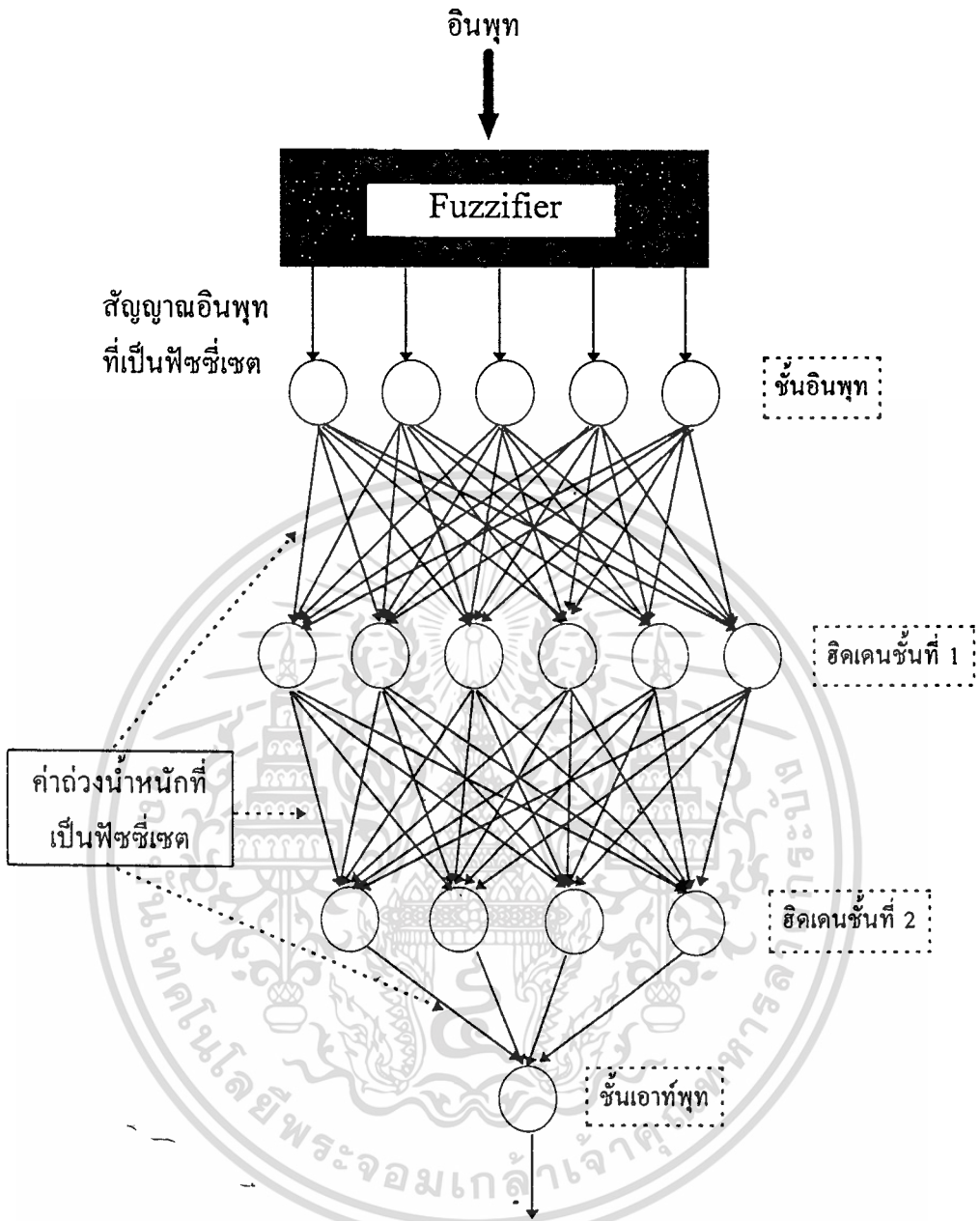
5.2.1 ส่วนที่เป็นฟัซซี่เซต ซึ่งจะดัดแปลงจากระบบฟัซซี่ในบทที่ 2 คือจะดึงเอาเฉพาะส่วนของ Fuzzifier มาใช้เพื่อแปลงข้อมูลในส่วนของอินพุตให้เป็นค่าฟัซซี่เซต ดังนั้นการใช้โครงข่ายฟัซซีนิวรอลจะตัดปัญหาการสร้างความสัมพันธ์จากอินพุตไปยังเอาต์พุต (ตาราง FAM) และการ Defuzzifies ซึ่งทั้งสองส่วนนี้จะโอนไปให้ส่วนของโครงข่ายดำเนินการแทน

5.2.2 ส่วนที่เป็นนิวรอลเน็ตเวิร์ค โดยส่วนนี้จะรับสัญญาณอินพุตที่เป็นฟัซซี่เซตจากขั้นตอนการ Fuzzify เข้ามา และส่วนนี้จะทำหน้าที่แทนการสร้างความสัมพันธ์ระหว่างอินพุตกับเอาต์พุต รวมไปถึงการ Defuzzify ในระบบฟัซซี่ด้วย สำหรับการพิจารณาขนาดของโครงข่ายยังคงใช้หลักการเดียวกันกับการพิจารณาขนาดของโครงข่ายเซลล์ประสาทธรรมดา แต่ในส่วนของแพทเทิร์นที่ใช้ในการสอนจะถูกนำมาประมวลผลก่อนในขั้นตอนการ Fuzzify ทำให้แพทเทิร์นที่ใช้สอนอยู่ในรูปแบบที่เป็นยูนิฟอร์ม ซึ่งจะส่งผลให้การเรียนรู้และการจำได้ของโครงข่ายมีประสิทธิภาพมากยิ่งขึ้น

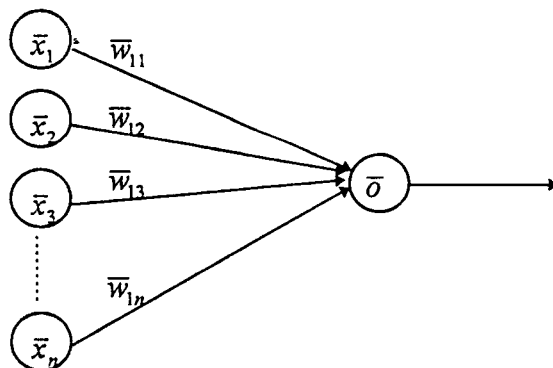
5.3 การสอนโครงข่ายฟัซซีนิวรอล

ฟัซซีนิวรอลเน็ตเวิร์ค ที่จะเสนอในที่นี้เป็นชนิดที่อินพุตและค่าถ่วงน้ำหนักในแต่ละชั้นเป็นฟัซซี่เซต และโครงข่ายนี้จะถูกสอนโดยกฎเซลล์แบบฟัซซี่ (Fuzzified Delta Rule) [68-69] ซึ่งลักษณะของโครงข่ายอย่างง่ายแสดงในรูปที่ 5.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.1 แสดงแบบจำลองฟัซซีนิวรอลเน็ตเวิร์คที่ใช้ในการหาขอบภาพ



รูปที่ 5.2 แสดงโครงข่าย FNN ที่เอาต์พุตและค่าถ่วงน้ำหนักเป็นฟัซซี่เซต

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของเจ้าของเนื้อหา ไม่อนุญาตให้นำไปใช้หรือเผยแพร่ในเชิงพาณิชย์ การค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงข่ายพีชชีนิวโรลในรูปที่ 5.2 ใช้การสอนแบบชี้เป้าซึ่งเซตที่ใช้สอนก็เป็นพีชชีเซตเช่นกัน นั่นคือ $T_p = \{\bar{X}_p, \bar{D}_p\}$; $p = 1, 2, 3, \dots, P$ เมื่อ P เป็นจำนวนแพทเทิร์นทั้งหมดที่จะทำการสอน (สำหรับเครื่องหมายบาร์ที่อยู่บนตัวอักษรภาษาอังกฤษตัวใดแสดงว่าตัวนั้นเป็นพีชชีเซต) สำหรับ Activation Function ของโครงข่ายพีชชีนิวโรลนี้ จะใช้ฟังก์ชัน Sigmoid ซึ่งสามารถกำหนดได้ดังนี้

$$\bar{o}_{pj} = f(\bar{y}_{pj}) = \frac{1}{1 + e^{-\bar{y}_{pj}}} \quad (5.1)$$

โดยที่ $\bar{y}_{pj} = \sum_{i=1}^n \bar{w}_{ji} \bar{x}_{pi}$

การเรียนรู้ของโครงข่ายพีชชีนิวโรลที่อินพุตและค่าถ่วงน้ำหนักเป็นพีชชีเซต จะใช้กฎการสอนแบบพีชชีเคลต้าซึ่งมีรายละเอียดดังนี้

กฎการสอนเคลต้าแบบพีชชี

การวัดความคลาดเคลื่อนของค่าผลลัพธ์ที่ได้จากโครงข่ายสามารถทำได้ดังนี้

$$\bar{E}_p = \frac{1}{2} \sum_{p=1}^P (\bar{O}_p - \bar{D}_p)^2 \quad (5.2)$$

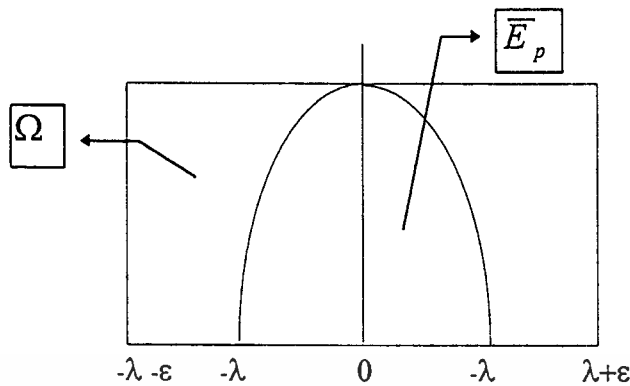
ในการเรียนรู้ของโครงข่ายเราต้องการให้ค่าความคลาดเคลื่อนเท่ากับศูนย์ ($\bar{E} \rightarrow 0$) แต่สำหรับ FNN ที่อินพุตและค่าถ่วงน้ำหนักเป็นพีชชีเซตแล้วค่าความคลาดเคลื่อนของโครงข่ายจะไม่เท่ากับศูนย์ ถึงแม้ว่า $\bar{O}_p = \bar{D}_p$ ก็ตามทั้งนี้เป็นผลมาจากกรลบก้นของพีชชี ดังนั้นจึงต้องตั้งเกณฑ์ขึ้นมาเพื่อจะบอกให้โครงข่ายหยุดการเรียนรู้เมื่อใด

กำหนดให้ ช่วงของเซตของผลลัพธ์ที่ต้องการ \bar{D}_p อยู่ในช่วง $[\bar{d}_{p1}, \bar{d}_{p2}]$ ดังนั้นเมื่อ $\bar{O}_p = \bar{D}_p$; $\forall p \in P$ จะได้ว่าช่วงของเซตของค่าความคลาดเคลื่อนจะอยู่ในช่วง $[-\lambda, \lambda]$ โดยที่

$$\lambda = \frac{1}{2} \sum_{p=1}^P (\bar{d}_{p1} - \bar{d}_{p2})^2 \quad (5.3)$$

และกำหนดให้ $\varepsilon > 0$ ซึ่งเป็นค่าที่เบี่ยงเบนไปจากค่าคาดหวังของ \bar{E}_p เมื่อ $\bar{O}_p = \bar{D}_p$; $\forall p \in P$ ดังนั้นการสอนโครงข่ายจะสิ้นสุดลงเมื่อ \bar{E}_p เป็นสมาชิกของ Ω

เอกสารนี้เป็นเอกสารที่ $\Omega = [-\lambda - \varepsilon, \lambda + \varepsilon] \times [0, 1]$ ารศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ (5.4) ด้านการค้า ไม่ว่าการณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



สำหรับการปรับค่าถ่วงน้ำหนักสามารถดำเนินการได้ดังนี้

$$\Delta \bar{W}(t+1) = \bar{W} + \eta \Delta \bar{W}(t) \quad (5.5)$$

โดยที่ $t = 1, 2, 3, \dots$ เป็นจำนวนครั้งที่ทำการสอน และ η เป็นค่าคงที่ที่ใช้แทนอัตราการเรียนรู้ และ

$$\begin{aligned} \Delta \bar{W}(t) &= \frac{\partial \bar{E}}{\partial \bar{W}(t)} \\ &= \sum_{p=1}^P (\bar{O}_p - \bar{D}_p)(\bar{O}_p)(1 - \bar{O}_p) \bar{X}_{pi} \end{aligned} \quad (5.6)$$

สำหรับโปรแกรมของโครงข่ายฟuzzyนิวรอล ที่มีการสอนโดยใช้กฎเดลด้าแบบฟuzzyได้แสดงไว้ในภาคผนวก ก. และแผนภูมิการสอน โครงข่ายนี้ที่ใช้ในการหาขอบภาพจะแสดงไว้ในบทถัดไป

สรุป

โครงข่ายฟuzzyนิวรอลที่ใช้กฎการสอนเดลด้าแบบฟuzzy โดยมีค่าถ่วงน้ำหนักที่เชื่อมโยงระหว่างชั้น เป็นค่าฟuzzyเซตด้วย สามารถใช้แทนระบบฟuzzyหรือนิวรอลเน็ตเวิร์คแบบธรรมดาได้ โดยโครงข่ายนี้จะช่วยลดปัญหาในการออกแบบระบบฟuzzyเซต และจะช่วยเพิ่มประสิทธิภาพการเรียนรู้และการจำได้ของโครงข่ายด้วย

บทที่ 6 การหาขอบภาพ

วิธีการหาขอบภาพที่ได้เสนอไปแล้วในบทที่ 1 เป็นวิธีการหาขอบภาพที่นิยมใช้กันทั่วไป และเป็นวิธีที่รู้จักกันดีสำหรับผู้ที่สนใจในทางด้านการประมวลผลภาพเชิงตัวเลข ไม่ว่าจะเป็นตัวดำเนินการหาขอบภาพของ Sobel, Marr, Gabor และการหาขอบภาพโดยวิธี Relaxation ซึ่งวิธีเหล่านี้จะใช้เปรียบเทียบกับวิธีที่จะเสนอซึ่งได้แก่วิธีการหาขอบภาพโดยการประมวลผลก่อนโดยใช้ฟังก์ชันเซต [71], การหาขอบภาพโดยฟังก์ชันเซต, การหาขอบภาพโดยใช้นิวรอลเน็ตเวิร์ค [61], และการหาขอบภาพโดยใช้ฟังก์ชันนิวรอลเน็ตเวิร์ค ซึ่งการหาขอบภาพโดยวิธีหลังสุดนี้จะเป็นการพิจารณาขอบภาพจากตัวแปรที่ได้จากเนื้อหาภาพคือค่า Dispersion และค่าส่วนเบี่ยงเบนมาตรฐานของค่าความเข้มจุดภาพที่อยู่ภายในวินโดว์ขนาด 2×2 และในการเปรียบเทียบวิธีการหาขอบภาพของวิธีที่กล่าวมาแล้วในบทที่ 1 กับวิธีที่จะเสนอได้ทำการวัดความถูกต้องของตำแหน่งและเปอร์เซ็นต์ขอบภาพที่ได้ซึ่งข้อมูลภาพที่นำมาทดสอบได้เพิ่มสัญญาณรบกวนทั้งที่มีการแจกแจงแบบยูนิฟอร์ม และแบบ Gaussian สำหรับขั้นตอนการหาขอบภาพของแต่ละวิธีจะได้เสนอรายละเอียดในหัวข้อถัดไป

6.1 การหาขอบภาพโดยการประมวลผลก่อนโดยใช้ฟังก์ชันเซต

การหาขอบภาพโดยวิธีนี้จะใช้ขั้นตอนการประมวลผลก่อน ซึ่งเป็นวิธีการปรับปรุงภาพแบบจุดโดยใช้ฟังก์ชันเซตซึ่งได้เสนอไปแล้วในบทที่ 3 (หัวข้อ 3.1.8 เรื่องการปรับปรุงภาพโดยฟังก์ชัน) และนำภาพที่ผ่านการปรับปรุงแล้วมาทำการกำจัดสัญญาณรบกวน โดยใช้ตัวกรองสัญญาณมัลติเรขาคณิต (ซึ่งได้กล่าวถึงแล้วในหัวข้อ 3.2.2.2) เนื่องจากในขั้นตอนการปรับปรุงภาพจะมีผลต่อสัญญาณรบกวนด้วยคือถ้าภาพมีสัญญาณรบกวนการปรับปรุงภาพจะส่งผลให้สัญญาณรบกวนปรากฏชัดเจนขึ้น หลังจากนั้นจะแบ่งค่าความเข้มของจุดภาพออกเป็น K ช่วงแล้วใช้ฟังก์ชัน π มาเน้นข้อมูลภาพสำหรับส่วนที่เป็นขอบของวัตถุให้มีความคมชัดขึ้นแล้วจึงทำการหาขอบภาพ ซึ่งขั้นตอนต่างๆ ของวิธีการหาขอบภาพโดยการประมวลผลก่อน โดยใช้ฟังก์ชันเซต มีรายละเอียดดังนี้

6.1.1 การปรับปรุงภาพโดยฟังก์ชันเซต

ในขั้นตอนนี้จะคล้ายกับขั้นตอนการปรับปรุงภาพแบบจุดโดยฟังก์ชันเซตที่ได้เสนอไปแล้ว สำหรับการปรับปรุงภาพที่นำมาใช้ในการหาขอบภาพ จะเพิ่มส่วนที่ใช้ในการเน้นขอบภาพให้ปรากฏชัดเจนขึ้นมาโดยใช้ส่วนขยายของฟังก์ชันเซต ซึ่งเป็นตัวดำเนินการ Contrast Intensification และขั้นตอนการปรับปรุงภาพโดยฟังก์ชันเซตสำหรับการหาขอบภาพจะเริ่มจากการคำนวณฟังก์ชันความเป็นสมาชิก ซึ่งสามารถคำนวณได้ดังนี้

$$\mu_A(x_{mn}) = \left(1 + \frac{|\hat{x} - x_{mn}|}{Fd} \right)^{-Fe} \quad (6.1)$$

$$Fd = \frac{\hat{x} - mid}{0.5^{-\sqrt{Fe}} - 1}$$

$$mid = min + (max - min + 1) / 2$$

min : เป็นค่าต่ำสุดของค่าความเข้มจุดภาพสำหรับภาพหนึ่งๆ และ max : เป็นค่าสูงสุดของค่าความเข้มจุดภาพในภาพเดียวกัน

x_{mn} : เป็นค่าความเข้มของจุดภาพที่ตำแหน่ง m, n

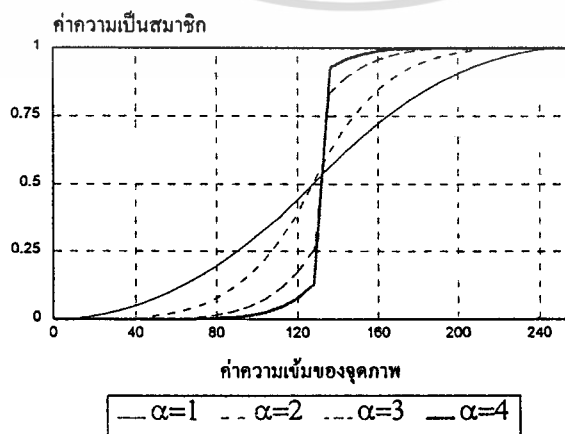
Fe : เป็นค่าคงที่ซึ่งเป็นส่วนขยายของฟังก์ชันเซต ในที่นี้ Fe มีค่าเท่ากับ 2

\hat{x} : ค่านี้ขึ้นอยู่กับทางเลือกใช้ฟังก์ชัน S หรือ π กรณีที่ใช้ฟังก์ชัน S ซึ่งส่วนใหญ่จะใช้ในการปรับปรุงภาพ \hat{x} จะเท่ากับค่าสูงสุดของจุดภาพของภาพนั้นๆ ($\hat{x} = max$) และในกรณีที่ใช้ฟังก์ชัน π (ซึ่งจะใช้ในขั้นตอนถัดไป) $\hat{x} = mid$

สำหรับฟังก์ชันความเป็นสมาชิกในขั้นตอนนี้จะใช้ฟังก์ชัน S เมื่อคำนวณค่าความเป็นสมาชิกของแต่ละจุดภาพได้แล้วก็ทำการเน้นค่าความเป็นสมาชิกของจุดภาพนั้นๆ โดยการใช้ตัวดำเนินการ Contrast Intensification ซึ่งสามารถกำหนดได้ดังนี้

$$\mu'_A(x_{mn}) = \begin{cases} 2(\mu_A(x_{mn}))^\alpha; & 0 \leq \mu_A(x_{mn}) \leq 0.5 \\ 1 - 2(1 - \mu_A(x_{mn}))^\alpha; & 0.5 < \mu_A(x_{mn}) \leq 1 \end{cases} \quad (6.2)$$

เมื่อ $\mu'_A(x_{mn})$ เป็นค่าความเป็นสมาชิกของจุดภาพที่ผ่านการเน้นค่าความเป็นสมาชิกที่อยู่ในช่วงกลางๆ (บริเวณ 0.5 หรือที่เรียกว่าจุด Crossover : x_C) เมื่อค่า α ยิ่งสูงขึ้นข้อมูลภาพก็จะถูกลู่เข้าสู่ภาพไบนารี ซึ่งลักษณะของสมการที่ (6.2) แสดงในกราฟรูปที่ 6.1



รูปที่ 6.1 ลักษณะของฟังก์ชัน Contrast Intensification ที่ใช้ในการเน้นภาพเมื่อค่า α ต่างๆ กัน
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และเมื่อทำการเน้นข้อมูลภาพแล้วก็จะทำการแปลงค่าความเป็นสมาชิกของจุดภาพที่ผ่านตัวดำเนินการ Contrast Intensification ไปเป็นค่าความเข้มของจุดภาพโดยใช้สมการที่ (6.3)

$$x'_{mn} = L - Fd \left(\frac{1}{(\mu'_A(x_{mn}))^{-1/F_e}} - 1 \right) \quad (6.3)$$

โดยที่ L เป็นค่าคงที่ซึ่งเป็นค่าความเข้มสูงสุดของข้อมูลภาพที่มีขนาด 8 บิต ($L=255$)

6.1.2 การกำจัดสัญญาณรบกวน ขั้นตอนนี้มีความจำเป็นมากเพราะถ้าข้อมูลภาพมีสัญญาณรบกวนปะปนอยู่ด้วย อาจจะทำให้ขอบภาพผิดไปจากความเป็นจริงได้ สำหรับการกำจัดสัญญาณรบกวนที่ใช้ในที่นี้เป็นตัวกรองสัญญาณมัลติฐานภายในวินโดว์ขนาด 3×3 ดังที่ได้กล่าวไว้แล้วในบทที่ 3

6.1.3 ทำการแบ่งข้อมูลภาพออกเป็น K ช่วง โดยพิจารณาจากฮิสโตแกรมของข้อมูลภาพ และนำข้อมูลภาพแต่ละช่วงมาเน้นภาพโดยใช้ฟังก์ชันเซตอีกครั้งหนึ่ง ซึ่งจะใช้ฟังก์ชัน π ในการคำนวณค่าความเป็นสมาชิกของข้อมูลภาพ สำหรับค่าความเป็นสมาชิกในแต่ละช่วง $l_k; k = 1, 2, 3, \dots, K$ สามารถกำหนดได้ดังนี้

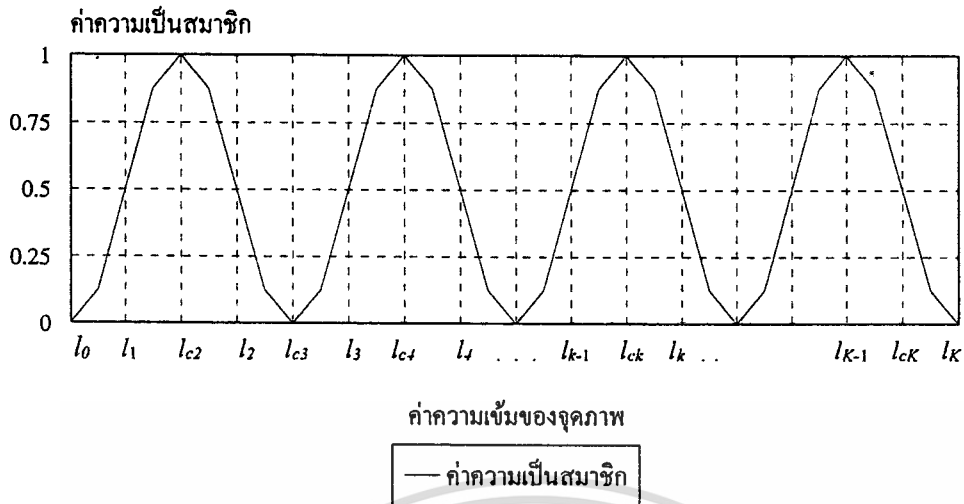
$$\mu_A(x_{mn}) = \begin{cases} G_\pi(x_{mn}; l_2 - l_1, l_{c2}); & x_{mn} \leq l_2 \\ 1 - G_\pi(x_{mn}; l_3 - l_2, l_{c3}); & l_2 < x_{mn} \leq l_3 \\ G_\pi(x_{mn}; l_4 - l_3, l_{c4}); & l_3 < x_{mn} \leq l_4 \\ \dots & \dots \\ 1 - G_\pi(x_{mn}; l_k - l_{k-1}, l_{ck}); & l_{k-1} < x_{mn} \leq l_k \\ \dots & \dots \\ 1 - G_\pi(x_{mn}; l_{K-1} - l_{K-2}, l_{c(K-1)}); & x_{mn} > l_{K-2} \end{cases} \quad (6.4)$$

โดยที่ $l_{ck} = (l_k + l_{k-1})/2; k = 1, 2, 3, \dots, K$

$$G_\pi(x_{mn}; l_k - l_{k-1}, l_{ck}) = \left(1 + \frac{|l_{ck} - x_{mn}|}{Fd} \right)^{-F_e}; \quad l_{k-1} < x_{mn} \leq l_k \quad (6.5)$$

ในรูปที่ 6.2 แสดงลักษณะของฟังก์ชันความเป็นสมาชิกทั้ง K ช่วง เมื่อคำนวณค่าความเป็นสมาชิกของแต่ละจุดภาพได้แล้วก็นำค่าความเป็นสมาชิกที่ได้ไปเน้นโดยใช้สมการที่ (6.2) อีกครั้งหนึ่ง และนำค่าความเป็นสมาชิกที่ผ่านการดำเนินการ Contrast Intensification ในสมการที่ (6.2) แล้วไปแปลงเป็นค่าความเข้มของจุดภาพโดยใช้สมการที่ (6.3)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.2 กราฟของฟังก์ชัน π ที่มี K ช่วง

6.1.4 หากขอภาพ จากขั้นตอนที่ 6.1.3 จะทำให้ภาพระดับสีเทาเกือบจะเป็นภาพไบนารี ดังนั้นในการหาค่าขอบภาพสามารถดำเนินการได้โดยใช้สมการที่ (6.6)

$$e(m, n) = \bigcup_{m, n} x_{mn}'' \tag{6.6}$$

โดยที่ $x_{mn}'' = |x_{mn}'' - \max(x_{ij}''); (i, j) \in W_{3 \times 3}$

x_{mn}'' : เป็นจุดภาพที่ได้จากขั้นตอน 6.1.3

$\max(x_{ij}')'$: เป็นค่าสูงสุดของวินโดวขนาด 3×3 ($W_{3 \times 3}$)

จากขั้นตอนการหาขอบภาพโดยการประมวลผลก่อนโดยใช้พีซีเซตทั้ง 4 ขั้นตอน สามารถเขียนเป็นอัลกอริทึมได้ดังนี้

อัลกอริทึมที่ 6-1. แสดงขั้นตอนการประมวลผลภาพโดยการประมวลผลก่อนโดยใช้พีซีเซต

```
#define CROSSOVER 0.5 // จุด Crossover ของข้อมูลภาพ
// ฟังก์ชัน Contrast Intensification ใช้ในการเน้นข้อมูลภาพ
float Contrast_Intensification(float Mu, float alpha)
{
    if(Mu <= CROSSOVER)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 Mu = 2*pow(Mu, alpha);
 ไม่ว่ากรณีใดๆทางสน ออกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else
    Mu = 1-2*pow(1-Mu, alpha);

return Mu;
}
/* ฟังก์ชัน Membership_Function ใช้ในการกำหนดค่าความเป็นสมาชิก ซึ่งสามารถใช้คำนวณได้ทั้ง
* ฟังก์ชัน S และ  $\pi$  ขึ้นอยู่กับตัวแปร x_hat ถ้า x_hat เป็นค่าสูงสุดของข้อมูลฟังก์ชันนี้จะให้ค่าเป็นฟังก์ชัน
* S แต่ถ้า ค่า x_hat เป็นค่ากึ่งกลางของข้อมูลฟังก์ชันนี้จะให้ค่าเป็นฟังก์ชัน  $\pi$ 
*/
float Membership_Function(int x_hat, int x, float Fd, float Fe)
{
    float Mu;

    Mu = pow(1+abs(x_hat-x)/Fd, -Fe);

    return Mu;
}
/* ฟังก์ชัน Membership_Value_to_Pixel ใช้ในการแปลงข้อมูลจากค่าความเป็นสมาชิกไปเป็นค่าความเข้ม
* ของจุดภาพ
*/
int Membership_Value_to_Pixel(float Mu, float Fd, float Fe)
{
    int x, L=255;

    Mu = pow(Mu, -1/Fe);
    x = L-Fd*(1/Mu - 1);

    return x;
}

void Edge_Detection1(IMAGE I)
/* เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
* IMAGE เป็นโครงสร้างข้อมูลชนิด struct ซึ่งมีสมาชิกเช่นเดียวกับ IMAGE ในอัลกอริทึมที่ 3-1. */
    ไม่สามารถดึงทางสน ออกทางห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

```

float  Fd, Fe, Mu;
int    x_hat, mid;

// กำหนดค่าตัวแปรของฟังก์ชันความเป็นสมาชิก โดยในขั้นตอนแรกจะใช้ฟังก์ชัน S
x_hat = I.MAX;
mid = I.MIN+(I.MAX-I.MIN+1)/2;
Fe = 2;
Fd = (x_hat-mid)/(pow(0.5, -1/Fe)-1);
for(int m=0; m<I.nRows; m++)
    for(int n=0; n<I.nCols; n++)
    {
        Mu = Membership_Function(x_hat, I.data[m][n], Fd, Fe);
        Mu = Contrast_Intensification(Mu, 2);
        I.data[m][n] = Membership_Value_to_Pixel(Mu, Fd, Fe);
    }
// กำจัดสัญญาณรบกวนด้วยตัวกรองสัญญาณมัลฐาน
Median_Filter(I);
// กำหนดตัวแปรที่จะใช้ในการเน้นภาพโดยใช้ฟังก์ชัน  $\pi$ 
int    K, *I;
printf("\nK = ");
scanf("%d", &K);
I = new int[K+1];
// กำหนดค่าความเข้มซึ่งมีทั้งหมด K+1 ช่วง
for(m=0; m<=K; m++)
{
    int    k;

    printf("I[%d] = ", m);
    scanf("%d", &k);

    I[m] = k;
}

```

// เหน้ข้อมูลภาพครั้งที่สองโดยใช้ฟังก์ชัน π ที่แบ่งข้อมูลภาพออกเป็น K ช่วง

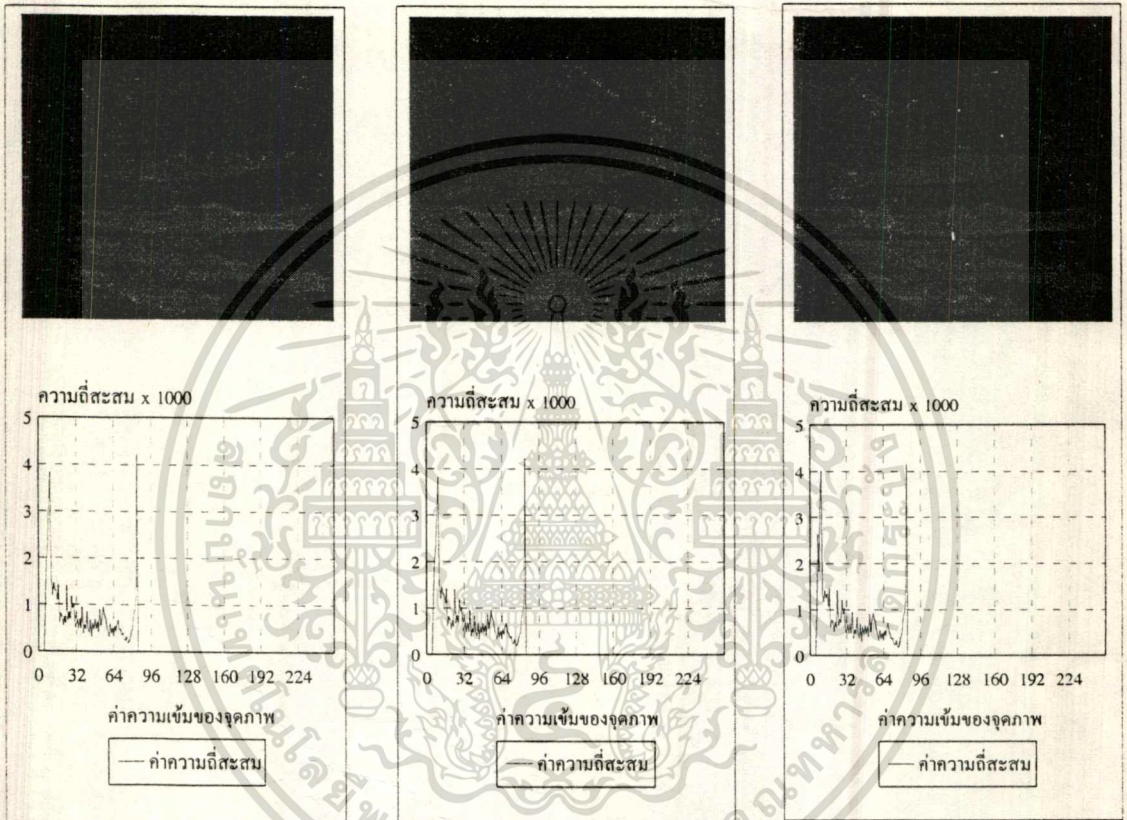
```

for(m=0; m<I.nRows; m++)
    for(int n=0; n<I.nCols; n++)
    {
        for(int k=1; k<=K; k++)
            if(I.data[m][n] <= l[k])
            {
                mid = l[k-1]+(l[k]-l[k-1]+1)/2;
                x_hat = mid;
                Fe = 2;
                Fd = (l[k]-mid)/(pow(0.5, -1/Fe)-1);
            }
        Mu = Membership_Function(x_hat, I.data[m][n], Fd, Fe);
        if(!((k-1)%2))
            Mu = 1 - Mu;
        Mu = Contrast_Intensification(Mu, 2);
        I.data[m][n] = Membership_Value_to_Pixel(Mu, Fd, Fe);
    }
// การหาขอบภาพภายในวินโดว์ขนาด 3 x 3
int    KERNEL=3;
for(m=0; m<I.nRows; m++)
    for(int n=0; n<I.nCols; n++)
    {
        int    max=0, W[3][3];
        for(int i=0; i<KERNEL; i++)
            for(int j=0; j<KERNEL; j++)
            {
                W[i][j] = I.data[m+i][n+j];
                max = W[i][j] > max? W[i][j] : max;
            }
    }

```

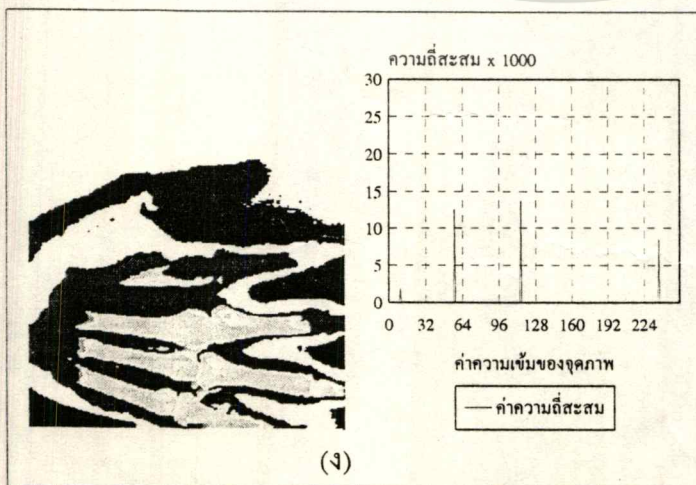
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อี $e[m][n] = \text{abs}(I.data[m][n]-max)$; อังอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับขั้นตอนการหาขอบภาพโดยวิธีการประมวลผลก่อนโดยใช้พีชคณิตในอัลกอริทึมที่ 6-1. ได้นำมาทดสอบหาขอบภาพเอ็กซ์เรย์ ซึ่งเป็นภาพมือที่ประกอบด้วยเนื้อเยื่อชั้นต่างๆ และกระดูกดังแสดงในรูปที่ 6.3

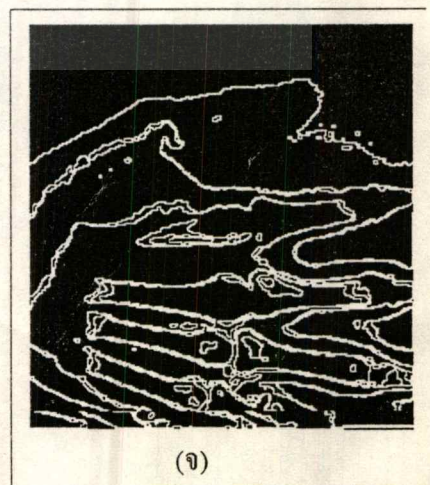


(ก)

(ค)



(ง)



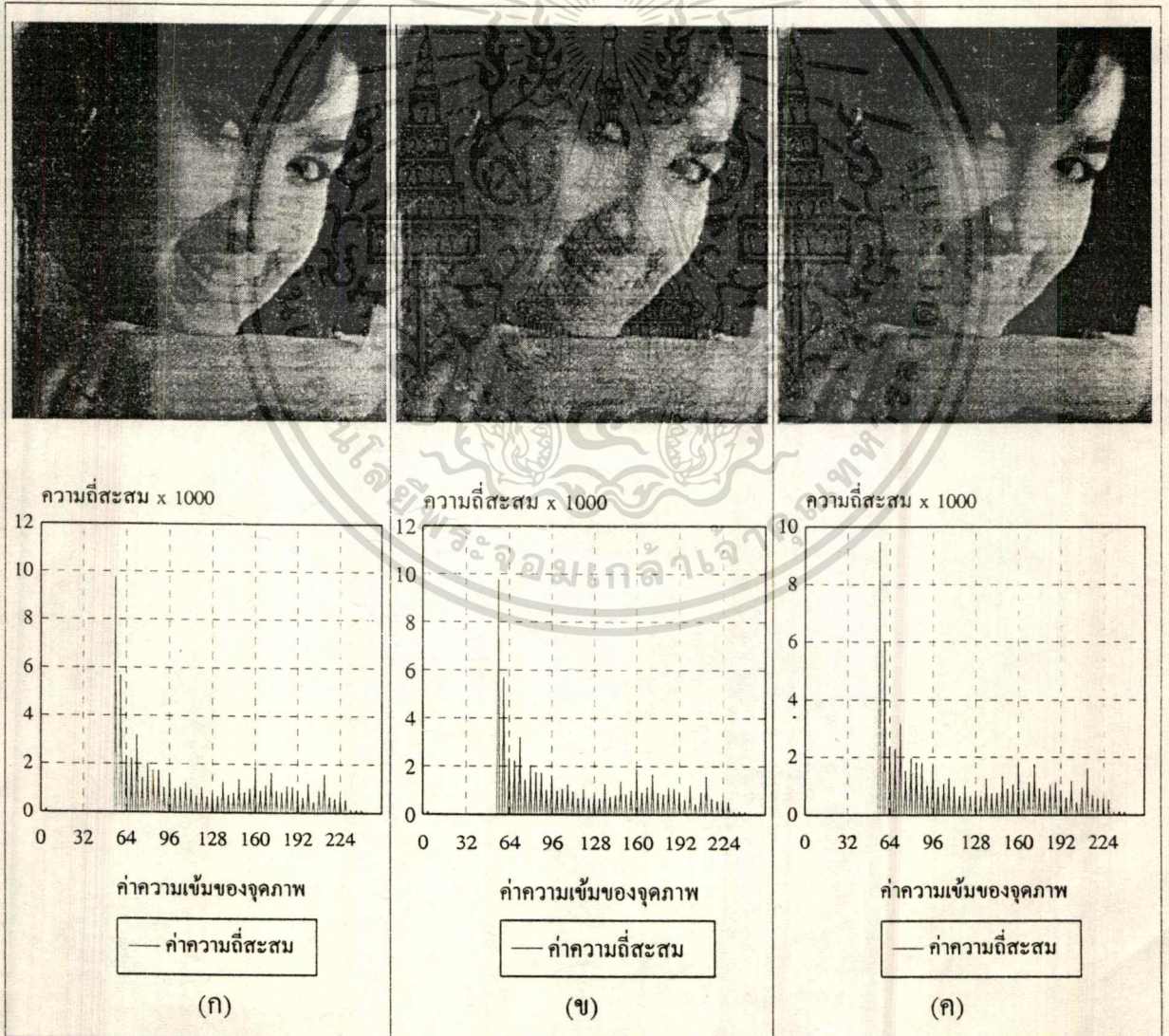
(จ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 6.3 แสดงขอบภาพที่ได้จากวิธีการประมวลผลภาพก่อนโดยใช้พีซีเซต

- (ก) อินพุท
- (ข) ภาพที่ผ่านการประมวลผลโดยใช้พีซีเซต
- (ค) ภาพที่ผ่านการกำจัดสัญญาณรบกวนโดยใช้ตัวกรองสัญญาณมัลติชาน
- (ง) รูปที่ผ่านการเน้นข้อมูลภาพโดยใช้ฟังก์ชัน π
- (จ) ขอบภาพ

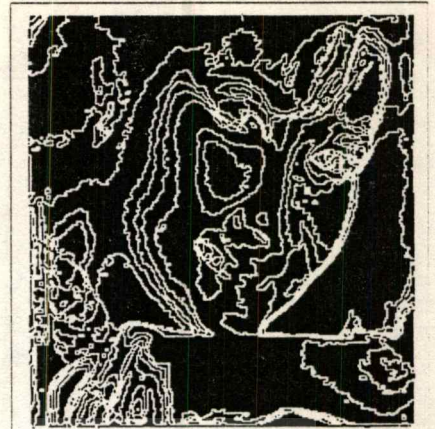
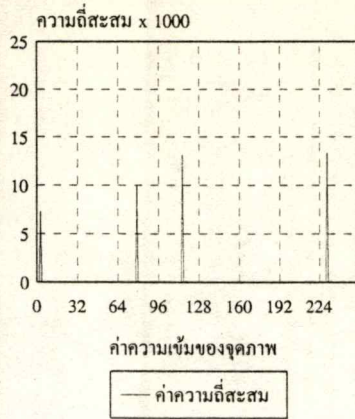
ในรูปที่ 6.3ง ข้อมูลภาพจะถูกแบ่งออกเป็น 5 ช่วง($K=5$) โดยพิจารณาจากฮิสโตแกรมในรูปที่ 6.3ค ซึ่งจำนวนช่วงของข้อมูล K จะขึ้นอยู่กับจำนวนวัตถุที่ปรากฏอยู่ในภาพ ในรูปที่ 6.4 แสดงขอบภาพที่ได้จากวิธีนี้อีกรูปหนึ่ง ที่ขนาดของภาพเท่ากับ 256×256 จุด สำหรับการวิจารณ์ผลลัพธ์ของวิธีนี้จะกล่าวถึงในหัวข้อของการเปรียบเทียบผลลัพธ์และสรุปผล



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ง)



(จ)

รูปที่ 6.4 แสดงขอบภาพที่ได้จากวิธีการประมวลผลภาพก่อน โดยใช้พีชชีเซต

- (ก) อินพุต
- (ข) ภาพที่ผ่านการประมวลผลโดยใช้พีชชีเซต
- (ค) ภาพที่ผ่านการกำจัดสัญญาณรบกวน โดยใช้ตัวกรองสัญญาณมัลชวาน
- (ง) รูปที่ผ่านการเน้นข้อมูลภาพโดยใช้ฟังก์ชัน π โดยที่ $K = 5$
- (จ) ขอบภาพ

6.2 การหาขอบภาพโดยพีชชีเซต

ในหัวข้อนี้จะเสนอการหาขอบภาพโดยใช้พีชชีลอจิก ซึ่งจะพิจารณาความเป็นขอบภาพร่วมกับการวิเคราะห์เนื้อหาภาพ วิธีนี้จะช่วยขจัดความยุ่งยากในการกำหนดค่าตัวแปรต่างๆ ที่เหมาะสมสำหรับใช้ในการหาขอบภาพ (เช่น ค่าเทรชโฮลด์ในการหาขอบภาพโดยวิธีเกรเดียนต์และลาปลาเซียน, และค่าส่วนเบี่ยงเบนมาตรฐานในการหาขอบภาพโดยวิธีลาปลาเซียนเกาส์เซียน) เมื่อไม่ทราบถึงข้อมูลข่าวสารต่างๆ ที่เกี่ยวกับภาพนั้นๆ มาก่อน ในที่นี้จะใช้กฎพีชชีในการเลือกโครงสร้างของขอบภาพที่มีอยู่ 20 โครงสร้าง เพื่อให้ได้โครงสร้างขอบภาพที่เหมาะสมกับข้อมูลภาพที่บริเวณนั้นมากที่สุด แล้วจึงจะพิจารณาความเป็นขอบภาพของจุดภาพนั้นๆ ว่าเป็นขอบภาพหรือไม่โดยอาศัยลักษณะเนื้อหาภาพ, ค่าความเป็นขอบภาพ, และค่าดัชนีพีชชีจากการทดสอบการหาขอบภาพโดยวิธีนี้สามารถใช้ได้กับภาพโดยทั่วไป และขอบภาพที่ได้จากวิธีนี้จะมี ความถูกต้องและขอบที่ได้จะบางกว่า เมื่อเปรียบเทียบกับวิธีการหาขอบภาพโดยวิธีอื่นๆ เช่น Sobel, Canny, Gabor, และการหาขอบภาพโดยวิธี Relaxation ที่ได้เสนอไปแล้วในบทที่ 1

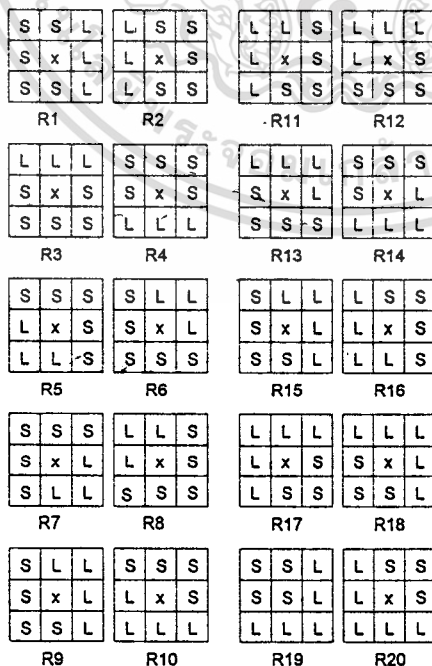
ในช่วงหลายปีที่ผ่านมา บทความที่เกี่ยวข้องกับการหาขอบภาพที่ใช้แบบจำลองทางคณิตศาสตร์ได้มีการเผยแพร่มากมาย (ดังที่ได้กล่าวมาแล้วในบทที่ 1) อย่างไรก็ตามอัลกอริทึมการหาขอบภาพที่ใช้แบบจำลองทางคณิตศาสตร์เหล่านี้ สามารถหาขอบภาพได้เพียงไม่กี่ชนิด ตัวอย่างเช่นตัวดำเนินการหาขอบภาพที่ใช้กับขอบภาพที่มีลักษณะเป็นแบบเปลี่ยนแปลงทันทีทันใด หรือ ขึ้นบันได แทนจะไม่มีผลเมื่อนำไปหาขอบภาพที่มีลักษณะเป็นแบบที่ค่าความเข้มของจุดภาพเพิ่มขึ้นหรือลดลงทีละน้อย หรือเป็นแบบหน้าจั่ว

นอกจากนี้แล้วตัวแปรต่างๆ ในบางแบบจำลองทางคณิตศาสตร์สามารถกำหนดได้ยากมาก เมื่อเราไม่ทราบถึงลักษณะการแจกแจงของข้อมูลภาพ ในปัจจุบันอัลกอริทึม ในการหาขอบภาพที่อาศัยพื้นฐานทางความรู้ของมนุษย์ (เช่น ระบบฐานความรู้: Knowledge-base System) [72] ได้ถูกเสนอ ไว้หลายวิธีด้วยกัน เช่น การหาขอบภาพโดยใช้เน็ตติกอัลกอริทึม [5], การหาขอบภาพโดยการจำลองแอนนิลลิ่ง [6], การหาขอบภาพโดยใช้ฟัซซี่เซต [2, 71], และการหาขอบภาพโดยใช้นิรอลเน็ตเวิร์ค [3, 4, 61] เป็นต้น ซึ่งวิธีการเหล่านี้มีความอ่อนตัวสูงทำให้สามารถใช้หาขอบภาพกับข้อมูลภาพโดยทั่วไปได้ และขอบภาพที่ได้ก็อยู่ในเกณฑ์ที่ดี

การหาขอบภาพที่จะเสนอในที่นี้ จะอาศัยแนวความคิดของฟัซซี่ลอจิก ในลักษณะของระบบผู้เชี่ยวชาญ (Expert System) โดยใช้กฎฟัซซี่แทนความรู้ของมนุษย์ในการเลือกโครงสร้างขอบภาพ (กฎในแต่ละข้อ) ให้เหมาะสมกับข้อมูลภาพในบริเวณนั้นๆ และเมื่อเลือกกฎที่เหมาะสมกับจุดภาพได้แล้วก็ทำการพิจารณาความเป็นขอบภาพโดยใช้การวิเคราะห์เนื้อหาภาพ [73] เพื่อพิจารณาว่าจุดภาพนั้นเป็นขอบภาพหรือไม่ โดยพิจารณาร่วมกับค่าความเป็นสมาชิกและดัชนีของความไม่แน่นอน (Index of Fuzziness) เพื่อให้ได้ขอบภาพที่แท้จริง การออกแบบการหาขอบภาพโดยใช้ฟัซซี่ลอจิกรายละเอียดดังนี้

6.2.1 กฎฟัซซี่

เซตของกฎฟัซซี่ IF-THEN ถูกออกแบบขึ้นโดยพื้นฐานความสัมพันธ์ระหว่างแต่ละจุดภาพกับจุดภาพในบริเวณใกล้เคียงอีกแปดจุด โครงสร้างของขอบภาพหรือกฎแต่ละข้อถูกแสดงในรูปที่ 6.5 ซึ่งมีอยู่ด้วยกันทั้งหมด 20 กฎ



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ที่ 6.5 แสดงโครงสร้างขอบภาพเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อใช้ในการหาความสัมพันธ์ระหว่างกฎกับคุณภาพ โดยกฎฟัซซี่ IF-THEN มีลักษณะดังนี้

IF (พื้นที่โดยเฉลี่ย $(\sum_i S'i)$ ของกฎที่ j มีค่าน้อยที่สุด และพื้นที่โดยเฉลี่ย $(\sum_i L'i)$ ของกฎที่ j มีค่ามากที่สุด)
THEN กฎที่ j จะเป็นโครงสร้างที่มีความสัมพันธ์กับคุณภาพนั้น

โดยที่ $S'i$ และ $L'i$ คือค่าความเข้มของระดับสีเทาซึ่งเป็นอนุพันธ์ที่เทียบกับจุด x เมื่อ $L'i$ แทนจุดภาพที่ i ที่แสดงด้วยตัวอักษร L ในรูปที่ 6.5 ซึ่งเป็นพื้นที่ที่มีค่าความเข้มมากกว่า $S'i$ ส่วน $S'i$ แทนจุดภาพที่ i ที่แสดงด้วยตัวอักษร S ในรูปที่ 6.5

พื้นที่ในกฎแต่ละกฎที่แสดงในรูปที่ 6.5 (ทั้ง L และ S) เป็นตัวแปรฟัซซี่เซตส่วนคำว่า "มาก" และ "น้อย" ในที่นี้เป็นเทอมเซตของฟัซซี่เซตหรือ Label

6.2.2 ฟังก์ชันความเป็นสมาชิก

ฟังก์ชันความเป็นสมาชิกของค่าความเข้มที่ใช้ในที่นี้คือ ฟังก์ชันรูปประฆังที่เป็นแบบฟังก์ชัน β ในกรณีของพื้นที่ที่มีค่าความเข้มน้อย ($S'i$) ค่าความเป็นสมาชิกสามารถกำหนดได้ดังนี้

$$\mu_S(S'i) = \frac{1}{1 + \left(\frac{S'i - \min}{(\max - \min) / 8} \right)^p} \quad (6.7)$$

โดยที่ \min : เป็นค่าต่ำสุดของ $S'i$ และ $L'i$
 \max : เป็นค่าสูงสุดของ $S'i$ และ $L'i$
 p : เป็นค่าคงที่ซึ่งในที่นี้มีค่าเท่ากับ 4

ในสมการที่ (6.7) จะมีคุณสมบัติเหมือนกับฟังก์ชัน β ในสมการที่ (2.11) ในบทที่ 2 ซึ่งในสมการนี้ค่า $\beta = (\max - \min) / 8$ ส่วนพื้นที่ที่มีค่าความเข้มมาก ($L'i$) ค่าความเป็นสมาชิกของ L สามารถกำหนดได้โดย

$$\mu_L(L'i) = 1 - \mu_S(L'i) \quad (6.8)$$

6.2.3 การเลือกกฎฟัซซี่เซต

ในการพิจารณาว่าจุดภาพที่ตำแหน่งใดๆ จะมีความเป็นสมาชิกของศักยภาพความเป็นขอบภาพของกฎที่ j ($\mu_p(x)$) ด้วยค่าความเป็นสมาชิกมากน้อยเพียงใดสามารถกำหนดได้โดยใช้ตัวดำเนินการอินเตอร์เอกสสารบนเอกสสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซกชันของฟuzzyเซต ตัวอย่างเช่น โครงสร้างขอบภาพในกฎ R1 ในรูปที่ 6.5 สามารถกำหนด $\mu_{P_j}(x)$ ได้ดังนี้

$$\begin{aligned} \mu_{P_j}(x) = & \text{Min}(\mu_S(S'_{00}), \mu_S(S'_{01}), \mu_L(L'_{02}), \\ & \mu_S(S'_{10}), \mu_L(L'_{12}), \\ & \mu_S(S'_{20}), \mu_S(S'_{21}), \mu_L(L'_{22})) \end{aligned} \tag{6.9}$$

ที่ S'_{ij} และ L'_{ij} เป็นจุดภาพในย่านใกล้เคียงของ X ซึ่งได้ผ่านการดิฟเฟอเรนซ์เทียบกับ x มาแล้วโดย i, j เป็นตำแหน่งของจุดภาพที่มีค่าอยู่ในช่วง $[0-2]$ สำหรับฟังก์ชัน Min เป็นตัวดำเนินการอินเตอร์เซกชัน หรือตัวดำเนินการ "AND" ในฟuzzyเซต [20]

เมื่อได้ค่าความเป็นสมาชิกของศักยภาพความเป็นขอบภาพ $\mu_{P_j}(X) ; j=1,2, 3, \dots, 20$ แล้วก็ทำการหาความสัมพันธ์ระหว่าง $\mu_{P_j}(X)$ ของทุกกฎกับจุดภาพที่ x ในข้อมูลภาพโดยใช้ตัวดำเนินการยูเนียนหรือตัวดำเนินการ "OR" ในฟuzzyเซตซึ่งจะแทนโดยฟังก์ชัน Max [20] ดังนั้นค่าความเป็นสมาชิกของความเป็นขอบภาพ ณ. จุดที่ สามารถเขียนได้ดังนี้

$$\begin{aligned} \mu_E(x) &= \mu_{P_1}(x) \cup \mu_{P_2}(x) \cup \mu_{P_3}(x) \dots \mu_{P_{20}}(x) \\ &= \text{Max}(\mu_{P_1}(x), \mu_{P_2}(x), \mu_{P_3}(x), \dots, \mu_{P_{20}}(x)) \end{aligned} \tag{6.10}$$

ที่ $\mu_E(x)$ คือค่าความเป็นสมาชิกของความเป็นขอบภาพที่จุด x ใดๆ ในภาพ

6.2.4 การตัดสินใจว่าจุดภาพเป็นขอบภาพหรือไม่

จากขั้นตอนที่ 6.2.3 ทำให้เราทราบว่าจุดภาพ x มีโครงสร้างความเป็นขอบภาพตามกฎข้อที่ j ใดๆ และจะนำโครงสร้างขอบภาพตามลักษณะของกฎข้อที่ j มาวิเคราะห์เนื้อหาภาพ โดยแบ่งพื้นที่ภายในกฎออกเป็นสองส่วนดังตัวอย่าง สมมุติจุดภาพ x ที่สอดคล้องกับกฎ R1 ซึ่งจะได้พื้นที่สองส่วนดังรูปที่ 6.6



รูปที่ 6.6 แสดงการแบ่งพื้นที่ของข้อมูลภาพที่ถูกแบ่งออกเป็นสองส่วน

จากนั้นนำข้อมูลภาพที่จุด x และจุดภาพในบริเวณใกล้เคียงอีกแปดจุดมาวิเคราะห์เนื้อหาภาพซึ่งตัวแปรที่ได้จากการวิเคราะห์เนื้อหาภาพนี้มีอยู่หลายตัวด้วยกัน ในที่นี้ได้เลือกเฉพาะตัวแปรที่มีผลต่อการหาขอบภาพเท่านั้น ซึ่งตัวแปรเหล่านี้จะใช้วัดความแตกต่างกันของพื้นที่ทั้งสองในรูปที่ 6.6 โดยพื้นที่ที่ระบายด้วยสีทึบจะเป็นเอกส่วนนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ส่วนที่มีค่าความเข้มของจุดภาพสูงกว่าส่วนที่ระบายด้วยสีที่สว่างกว่า และตัวแปรของเนื้อหาภาพก็จะบอกถึงเมวาร์ณเดจทงสน อักทงทามมให้ดดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความเป็นเนื้อเดียวกันของแต่ละพื้นที่ และจะบอกถึงความเข้มของจุดภาพที่แตกต่างกันระหว่างพื้นที่ทั้งสอง ซึ่งตัวแปรเหล่านี้ได้แก่ ค่าเฉลี่ย และค่าส่วนเบี่ยงเบนมาตรฐานของจุดภาพในแต่ละพื้นที่ (ส่วนที่ระบายด้วย สีทึบและสีสว่าง) และของทั้งสองพื้นที่รวมกันแล้วนำมาพิจารณาถึงความเป็นขอบภาพร่วมกับตัวแปรที่ได้จากการประมวลผลโดยใช้พีชคณิตคือค่าความเป็นขอบภาพ $\mu_E(x)$ และค่าที่บอกถึงความคลุมเครือในข้อมูลภาพ ซึ่งในที่นี้ใช้วิธีของ De Luca และ Termini ในสมการที่ (2.21) ในกรณีที่ใช้ในการพิจารณาค่าความเป็นขอบภาพสามารถกำหนดได้ดังนี้

$$H(A) = \left(\frac{1}{n} \ln 2\right) \sum_{i=1}^n \mu_A(x_i) \log \mu_A(x_i) + (1 - \mu_A(x_i)) \log(1 - \mu_A(x_i)) \quad (6.11)$$

เมื่อ n เป็นจำนวนของข้อมูลในวินโดวขนาด 3×3 , A เป็นเซตของข้อมูลภาพในวินโดวขนาด 3×3 ซึ่งก็คือ S^i และ L^i จากตัวแปรทั้งหมดที่กล่าวมาเราสามารถนำมาเขียนเป็นอัลกอริทึมสำหรับหาขอบภาพได้ดังอัลกอริทึมที่ 6-2 และสำหรับแผนภูมิของอัลกอริทึมนี้จะแสดงในภาคผนวก ก.

อัลกอริทึมที่ 6-2. เป็นขั้นตอนการหาขอบภาพโดยใช้พีชคณิตลอจิก

```
#define Num_of_RULE 20
// กฎที่ใช้ในการหาโครงสร้างของข้อมูลภาพซึ่งมีอยู่ 20. ข้อ
char RULES[20][3][3] = {
    {{1, 1, 0}, {1, 1, 0}, {1, 1, 0}}, //กฎข้อที่ 1.
    {{1, 0, 0}, {1, 1, 0}, {1, 1, 1}}, //กฎข้อที่ 2.
    {{0, 0, 1}, {0, 1, 1}, {1, 1, 1}}, //กฎข้อที่ 3.
    {{1, 1, 1}, {1, 1, 1}, {0, 0, 0}}, //กฎข้อที่ 4.
    {{1, 0, 0}, {1, 1, 0}, {1, 1, 0}}, //กฎข้อที่ 5.
    {{1, 1, 1}, {0, 1, 1}, {0, 0, 0}}, //กฎข้อที่ 6.
    {{0, 0, 1}, {0, 1, 1}, {0, 1, 1}}, //กฎข้อที่ 7.
    {{0, 0, 0}, {0, 1, 1}, {1, 1, 1}}, //กฎข้อที่ 8.
    {{0, 0, 0}, {1, 1, 0}, {1, 1, 1}}, //กฎข้อที่ 9.
    {{1, 1, 1}, {1, 1, 0}, {0, 0, 0}}, //กฎข้อที่ 10.
    {{1, 1, 0}, {1, 1, 0}, {1, 0, 0}}, //กฎข้อที่ 11.
    {{0, 1, 1}, {0, 1, 1}, {0, 0, 1}}, //กฎข้อที่ 12.
    {{1, 1, 0}, {1, 1, 0}, {0, 0, 0}}, //กฎข้อที่ 13.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ $\{(0, 0, 0), \{1, 1, 0\}, \{1, 1, 0\}\}$ ไม่อนุญาต //กฎข้อที่ 14. ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้าม $\{(0, 1, 1), \{0, 1, 1\}, \{0, 0, 0\}\}$ ถึงเจ้าของ //กฎข้อที่ 15. รังที่มีการนำไปใช้

```

({0, 0, 0}, {0, 1, 1}, {0, 1, 1}), //กฎข้อที่ 16.
({1, 1, 1}, {0, 1, 1}, {0, 0, 1}), //กฎข้อที่ 17.
({1, 1, 1}, {1, 1, 0}, {1, 0, 0}), //กฎข้อที่ 18.
({0, 1, 1}, {0, 1, 1}, {0, 1, 1}), //กฎข้อที่ 19.
({0, 0, 0}, {1, 1, 1}, {1, 1, 1}); //กฎข้อที่ 20.

```

// ฟังก์ชันที่ใช้ในการหาค่าความเป็นสมาชิกของค่าความเข้มของจุดภาพ

```
float Beta_Function(int B, int C, int xin)
```

// B และ C : เป็นค่าต่ำสุดและค่าสูงสุดของจุดภาพที่ผ่านการหาค่าอนุพันธ์มาแล้ว

```

{
    if(xin>=B && xin<C)
    {
        float s = (float)(C-B)/8.0;
        s = 1.0 + (float)pow((xin-B)/s, 4.0);
        return (1.0/s);
    }
    return 0.0;
}

```

// คำนวณค่าดัชนีของความคลุมเครือโดยใช้ฟังก์ชันของ De Luca และ Termini

```
float Entropy_DTE(int K, float *Mu)
```

// Mu เป็นเวกเตอร์ของค่าความเป็นสมาชิกที่มีอยู่ K ตัว

```

{
    float Sn=0;

    for(int i=0; i<K; i++)
    {
        if(Mu[i]>0 && Mu[i]<1)
            Sn = Sn + ((-1)*Mu[i]*log(Mu[i]) - (1.0-Mu[i])*(log(1.0-Mu[i])));
    }

    Sn = (1.0/(float)(K))*(float)log(2)*Sn;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

}

// ฟังก์ชันที่ใช้ในการหาขอบภาพโดยฟัซซี่เซต

BYTE Edge_Detection_by_Fuzzy_logic(BYTE I[3][3], float SD)

// I เป็นข้อมูลภาพที่อยู่ภายในวินโดว์ขนาด 3 x 3

// SD เป็นค่าส่วนเบี่ยงเบนมาตรฐานของภาพ

{

int D[3][3], K=3;

BYTE E, max = 0, min=255;

float Mu[9], sum=0, SDL;

for(int i=0; i<K; i++)

for(int j=0; j<K; j++)

{

// หาอนุพันธ์ของข้อมูลภาพภายในวินโดว์ขนาด 3 x 3 เทียบกับจุดภาพที่อยู่ตรงกลาง (ตำแหน่ง I[1][1])

D[i][j] = abs(I[1][1] - I[i][j]);

max = Max(max, D[i][j]); // หาค่าสูงสุด

min = Min(min, D[i][j]); // หาค่าต่ำสุด

sum = sum + I[i][j];

}

float mean = sum/(float)(K*K); // คำนวณค่าเฉลี่ยของข้อมูลภาพ

sum = 0.0;

for(i=0; i<K; i++)

for(int j=0; j<K; j++)

{

Mu[K*i+j] = Beta_Function(min, max, D[i][j]); // คำนวณค่าความเป็นสมาชิก

sum += ((I[i][j]-mean)*(I[i][j]-mean));

}

SDj = sqrt(sum/(float)(K*K)); // คำนวณค่าส่วนเบี่ยงเบนมาตรฐาน

max = 0;

float M1=2, M2=0;

// วนลูปเพื่อหาโครงสร้างของขอบภาพหรือกฎที่เหมาะสมกับข้อมูลในวินโดวขนาด 3 x 3

```
for(i=0; i<Num_of_RULE; i++)
```

```
{
```

```
    M1 = 2;
```

```
    for(int j=0; j<K; j++)
```

```
    {
```

```
        for(int k=0; k<K; k++)
```

```
        {
```

```
            if(RULES[i][j][k])
```

```
                M1 = Min(M1, Mu[K*j+k]);
```

```
            else
```

```
                M1 = Min(M1, 1.0 - Mu[K*j+k]);
```

```
        }
    }
}
if(M1 > M2)
```

```
{
```

```
    M2 = M1;
```

```
    max = i;
```

```
}
```

```
}
```

```
float mean1=0, mean2=0;
```

```
int count=0;
```

```
sum = 0.0;
```

```
for(i=0; i<K; i++)
```

```
    for(int j=0; j<K; j++)
```

```
    {
```

```
        if(RULES[max][i][j])
```

```
        {
```

```
            M1 = Mu[K*i+j];
```

```
            mean1 += (float)I[i][j];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกิจกรรมเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้เผยแพร่ข้อมูลนี้ไปยังผู้อื่นต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        count++;
    }
    else
    {
        M1 = 1.0 - Mu[K*i+j];
        mean2 += (float)I[i][j];
    }
    sum = sum + M1;
    Mu[K*i+j] = M1;
}
mean1 = mean1/(float)(count); // หาค่าเฉลี่ยของจุดภาพบริเวณที่ค่าความเข้มสูง (L'i)
mean2 = mean2/(float)(K*K-count); // หาค่าเฉลี่ยของจุดภาพบริเวณที่ค่าความเข้มต่ำ (S'i)
if(M2>0.0001)
    M2 = sum/(float)(count);
float SDL=0, SDS=0; // กำหนดตัวแปรของส่วนเบี่ยงเบนมาตรฐานของ L'i และ S'i
for(i=0; i<K; i++)
    for(int j=0; j<K; j++)
    {
        if(RULES[max][i][j])
            SDL += (I[i][j]-mean1)*(I[i][j]-mean1);
        else
            SDS += (I[i][j]-mean2)*(I[i][j]-mean2);
    }
SDL = sqrt(SDL/count); // หาค่าส่วนเบี่ยงเบนมาตรฐานของจุดภาพบริเวณที่ค่าความเข้มสูง (L'i)
SDS = sqrt(SDS/(K*K-count)); // หาค่าส่วนเบี่ยงเบนมาตรฐานของจุดภาพบริเวณที่
// ค่าความเข้มต่ำ (S'i)
sum = Entropy_DTE((K*K), Mu); // คำนวณค่าดัชนีความคลุมเครือ
if(sum > 0)
    M1 = pow(M2, sum); /* ปรับค่าความเป็นสมาชิกของความเป็นขอบภาพซึ่ง

```

- * ถ้าทั้งบริเวณ S'i และ L'i มีความเป็นเนื้อเดียวกันค่าดัชนีความคลุมเครือจะมีค่าต่ำมาก ซึ่งจะส่งผลให้ค่า
- * ดัชนีความเป็นขอบภาพมีค่าสูง ในทางกลับกันจะทำให้ระดับความเป็นขอบภาพเพิ่มขึ้นเล็กน้อยหรืออาจจะ

* ลดลงถ้าค่าดัชนีความคลุมเครือมีค่ามากกว่า 1

```
*/
if( ((mean1-mean2) < (SD*M1)) && ((SDL+SDS) > SDj) )
    E = NO_EDGE;
else
    E = EDGE;
return E;
}
```

จากอัลกอริทึม 6-2. นิพจน์ที่ใช้ในการกำหนดขอบภาพคือ

```
if( ((mean1-mean2) < (SD*M1)) && ((SDL+SDS) > SDj) )
    E = NO_EDGE;
else
    E = EDGE;
```

ซึ่งสามารถสรุปได้ดังนี้

1. ค่าความเป็นสมาชิกของความเป็นขอบภาพ $\mu_E(X)$ ในกฎที่ j จากสมการที่ (6.10) จะต้องมิต่ำสูงที่สุด ($\mu_E(X) \rightarrow 1$)
2. ค่า $H(A)$ ของกฎจะต้องมีค่าต่ำ ($H(A) \rightarrow 0$)
3. ส่วนเบี่ยงเบนมาตรฐานของพื้นที่ทั้งสองส่วน (SDS : σ_s และ SDL : σ_l) ของกฎที่ j จะต้องมิต่ำ (ค่าความเข้มของจุดภาพในแต่ละพื้นที่ที่มีความใกล้เคียงกัน)
4. ส่วนเบี่ยงเบนมาตรฐานของกฎที่ j (SDj : σ_j) จะต้องมิต่ำสูง (ค่าความเข้มของจุดภาพทั้งสองพื้นที่ต่างกันมาก)
5. ความแตกต่างของค่าเฉลี่ยของแต่ละพื้นที่ (mean1 และ mean2) มีค่าสูง

ถ้าจุด x ใดๆ ในภาพมีคุณสมบัติตามเงื่อนไขข้างต้นจุดภาพนั้นก็จะขอบภาพ และในทางตรงกันข้ามก็จะไม่ใช่ขอบภาพ สำหรับผลลัพธ์ของวิธีนี้จะกล่าวถึงในหัวข้อของผลการทดลองและเปรียบเทียบ และปัญหาต่างๆ จะอธิบายในหัวข้อการประเมินผลและวิจารณ์

6.3 การหาขอบภาพโดยนิรอรลเน็ตเวิร์ค

การหาภาพโดยใช้นิรอรลเน็ตเวิร์ค [61] ที่จะเสนอในหัวข้อนี้เป็นนิรอรลเน็ตเวิร์คที่มีโครงข่ายการเชื่อมโยงแบบ Multilayer Perceptron ที่มี 4 ชั้น โดยแต่ละชั้นจะมีจำนวนโหนดเท่ากับ 9, 20, 15, และ 1 ตามลำดับ และได้ใช้เทคนิคการสอนแบบ Back Propagation [57-60] ที่มีการคัดแปลงให้เหมาะสมกับการหาขอบภาพ เพื่อลดเวลาในการสอนโครงข่ายและเพิ่มประสิทธิภาพในการจำได้ของโครงข่ายหลังจากที่ได้สอนไปแล้ว ซึ่งจะเรียกโครงข่ายที่เราได้ปรับปรุงขึ้นมาใหม่นี้ว่า การแพร่กระจายกลับที่มีการปรับตัวเองแบบพลวัต (Dynamic Adaptive Back Propagation : DABP) [61] ซึ่งได้มีการปรับปรุงและเปลี่ยนแปลงจากวิธีการสอนแบบ Back Propagation แบบมาตรฐาน 4 ส่วนด้วยกันคือ

- อัตราเร็วในการเรียนรู้ ให้ผันแปรตามค่าความแปรปรวนของแต่ละแพทเทิร์น
- โมเมนต์ดัมให้แปรค่าตามค่าเบี่ยงเบนมาตรฐานของแพทเทิร์นทั้งหมดสำหรับการสอนในแต่ละรอบ
- ไปอัสของ Activation Function แปรค่าตามค่าเฉลี่ยของแต่ละแพทเทิร์น
- การกำหนดผลลัพธ์ในชั้นสุดท้ายให้เปลี่ยนตามค่าเฉลี่ยของแต่ละแพทเทิร์น

ผลจากการใช้ DABP ทำให้โครงข่ายมีศักยภาพในการเรียนรู้สูงขึ้นเป็นอย่างมาก ขั้นตอนการหาขอบภาพโดยใช้นิรอรลเน็ตเวิร์คที่มีการสอนแบบ DABP มีรายละเอียดดังนี้

6.3.1 อินพุทของนิรอรลเน็ตเวิร์ค

การหาขอบภาพโดยใช้นิรอรลเน็ตเวิร์คจะดำเนินการในลักษณะคอนโวลูชัน โดยใช้วินโดว์ขนาด 3×3 และวินโดว์จะถูกแปลงไปเป็นเวกเตอร์เพื่อที่จะส่งไปยังชั้นอินพุทของโครงข่ายดังรูปที่ 6.7

$$X_i = \begin{bmatrix} x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 \\ x_7 & x_8 & x_9 \end{bmatrix} \Rightarrow \vec{V}_i = \{x_1, x_2, x_3, \dots, x_5, \dots, x_9\}$$

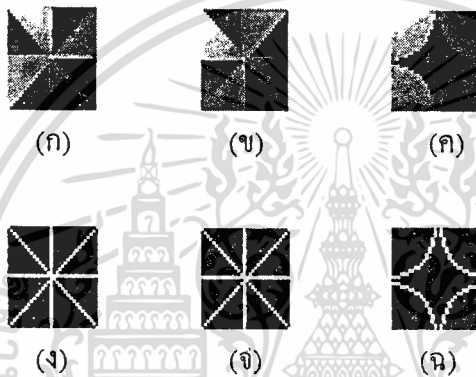
รูปที่ 6.7 อินพุทของโครงข่าย

จุดภาพตรงกลาง (x_5) ในเวกเตอร์จะถูกนำมาพิจารณาว่าใช่ขอบภาพหรือไม่โดยตรวจสอบกับค่าเอ๊าท์พุทที่ต้องการ d_i ถ้า $d_i = 0.1$ แสดงว่า x_5 ไม่ใช่ขอบภาพแต่ถ้า $d_i = 0.9$ แสดงว่า x_5 จะต้องเป็นขอบภาพ สำหรับข้อมูลที่ส่งไปยังชั้นอินพุทของนิรอรลเน็ตเวิร์คที่ใช้วิธีการสอนแบบที่ต้องมีการชี้แนะจะส่งไปแบบคู่ลำดับดังสมการ (6.12)

$$A = \{(X_1, d_1), (X_2, d_2), (X_3, d_3), \dots, (X_p, d_p)\} \quad (6.12)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ A เป็นเซตที่ใช้ในการสอนที่ประกอบด้วยคู่ลำดับทั้งหมด P คู่, โดยที่ X_i เป็นวินโดวที่ถูกละเลงไปเป็น \bar{V}_i เวกเตอร์ ส่วน d_i เป็นจุดภาพที่เป็นขอบ ในรูปที่ 6.8 แสดงข้อมูลที่ใช้ในการสอนโครงข่ายของเซต A โดยที่ X_i จะประกอบด้วย 3 ภาพ (รูปที่ 6.8ก - 6.8ค) และ d_i จะประกอบด้วย 3 ภาพข้างล่าง (รูปที่ 6.8ง - 6.8จ) อินพุตแพทเทิร์น X_i ในรูป ก ขอบภาพจะเป็นเส้นตรงและมี 8 ทิศทาง และภาพนี้จะสมนัยกับแพทเทิร์นของขอบภาพที่ต้องการในภาพที่ ง ส่วน X_i ในรูป ข จะผกผันกับภาพที่ ก และจะสมมูลกับแพทเทิร์นของขอบภาพที่ต้องการในรูป จ และรูป ค ขอบภาพจะเป็นเส้นโค้งซึ่งจะสอดคล้องกับแพทเทิร์นของขอบภาพในรูป ฉ สำหรับอินพุตแพทเทิร์น X_i (รูป ก - ค) จะถูกทำให้เป็นบรรทัดฐานก่อนที่จะส่งเข้าไปที่ชั้นอินพุต



รูปที่ 6.8 ข้อมูลที่ใช้ในการสอนเน็ตเวิร์ค

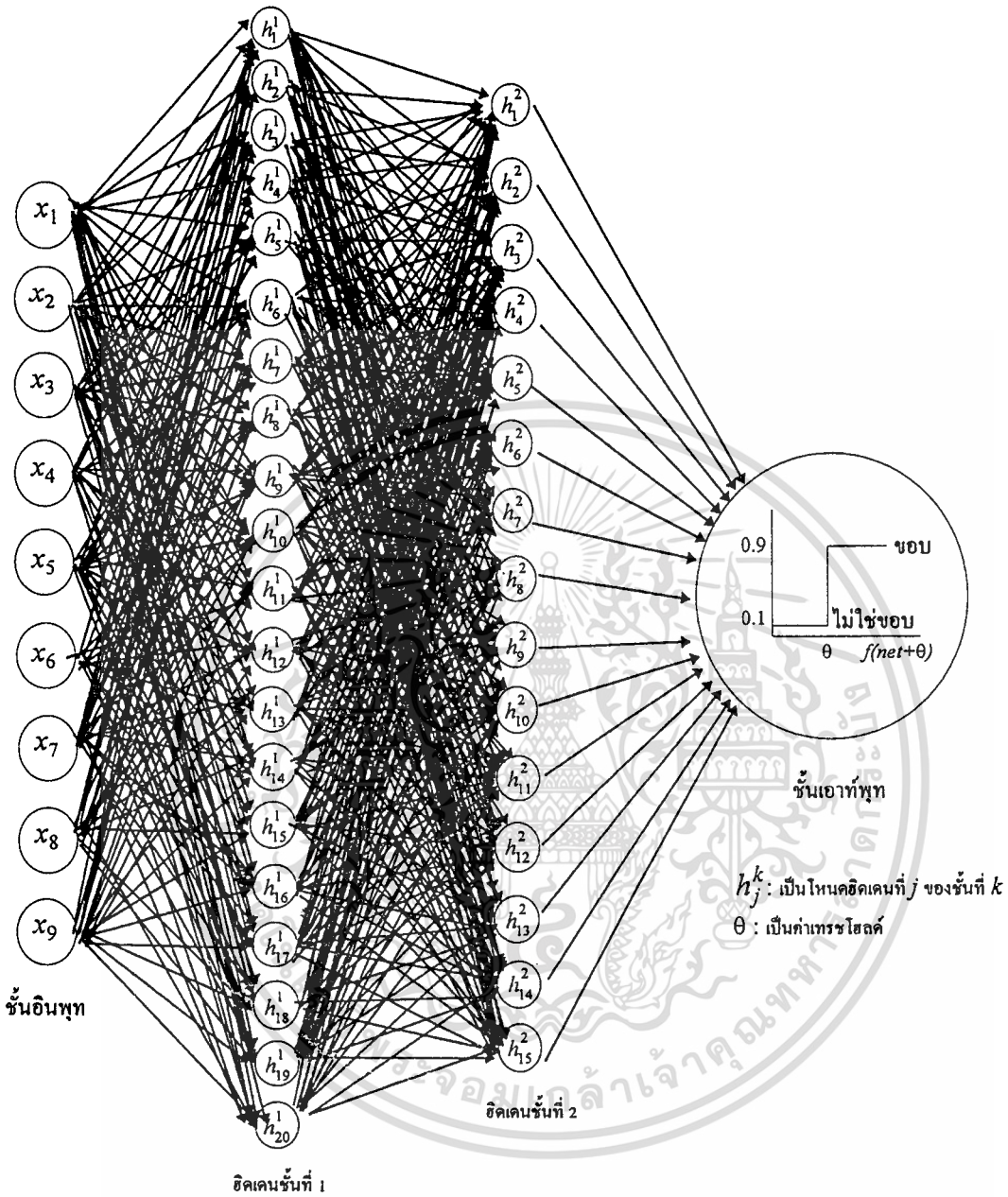
6.3.2 การคำนวณค่าเองที่พุดในแต่ละชั้น

ในรูปที่ 6.9 เป็นรูปโครงสร้างของนิวรอลเน็ตเวิร์คที่มีการเชื่อมโยงของโครงข่ายแบบ Multilayer Perceptron ซึ่งมี 4 ชั้น โดยชั้นแรกเป็นอินพุตซึ่งมีหน่วยประมวลผลเท่ากับ 9 โหนด, และข้อมูลที่อยู่ในชั้นอินพุต ทุกโหนดจะถูกส่งต่อไปยังฮิดเดนชั้นแรกซึ่งประกอบด้วยเซลล์ประมวลผล 20 เซลล์ เมื่อข้อมูลในฮิดเดนชั้นแรกประมวลผลเสร็จก็จะส่งข้อมูลในแต่ละโหนดไปยังฮิดเดนชั้นที่สองซึ่งมีเซลล์ประมวลผลอยู่ 15 เซลล์ และสุดท้ายข้อมูลในชั้นที่สองของฮิดเดนทุกโหนดจะถูกส่งไปยังชั้นของเอาต์พุตที่มีอยู่หนึ่งโหนด ซึ่งในโหนดนี้จะเทรซโวลต์ค่าเอาต์พุตให้เป็น 0.1 (ไม่ใช่ขอบ) และ 0.9 (ขอบ)

ผลลัพธ์ของโหนดในชั้นของฮิดเดนจนถึงชั้นเอาต์พุต (ชั้นที่ 2-4) จะถูกกำหนดโดย Activation Function ในที่นี้จะใช้ซิกมอยด์ฟังก์ชัน (Sigmoid function) ซึ่งเป็นฟังก์ชันที่ใช้ในการกำหนดค่าของผลลัพธ์ โดยมีการใช้ค่าไบอัสซึ่งสามารถกำหนดได้ดังนี้

$$f(\text{net}_j + \theta) = \frac{1}{1 + e^{-(\text{net}_j + \theta)}} \quad (6.13)$$

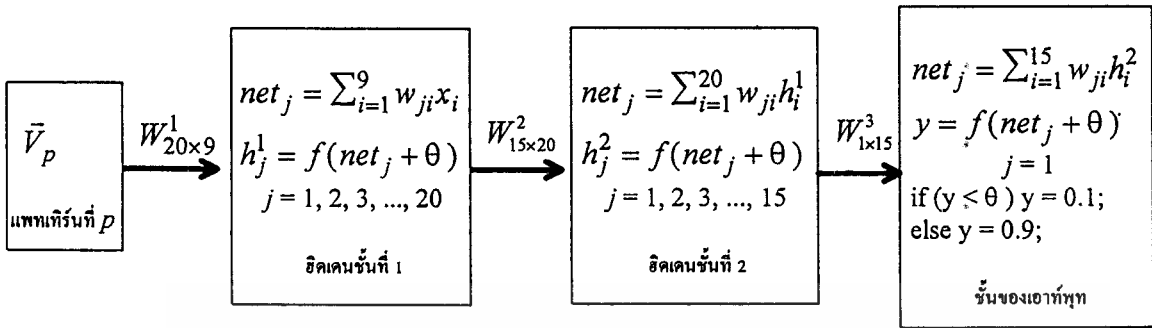
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.9 แบบจำลองโครงข่ายนิเวรอลที่ใช้ในการหาขอบภาพที่แต่ละโหนดของแต่ละชั้นจะเชื่อมต่อกันทุกโหนดในชั้นถัดไป

ที่ $f(\text{net}_j+\theta)$ เป็นแอกติเวชันฟังก์ชันที่เป็นอินเนอร์โปรดักซ์ (Inner Product) ของโหนดที่ j ระหว่างชั้นของอินพุทกับค่าถ่วงน้ำหนัก สำหรับการคำนวณค่าของเอาต์พุทในแต่ละชั้นจะแสดงในรูปที่ 6.10 และ θ เป็นค่าไบอัสที่ผันแปรตามค่าเฉลี่ยของแต่ละแพทเทิร์นและจะใช้ค่านี้เป็นค่าเทรซโซลด์ในชั้นเอาต์พุทเพื่อกำหนดค่าของผลลัพธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



W_{rc}^k : เมตริกซ์ของค่าถ่วงน้ำหนักที่เชื่อมต่อระหว่างชั้นที่ $k-1$ กับชั้นที่ k ซึ่งมีขนาด r แถว c คอลัมน์

w_{ji} : เป็นค่าถ่วงน้ำหนักที่เชื่อมต่อระหว่างโหนดที่ i ของชั้นที่ $k-1$ ไปยังโหนดที่ j ของชั้นที่ k และเป็นสมาชิกของเมตริกซ์ W_{rc}^k

รูปที่ 6.10 ขั้นตอนการกำหนดค่าเอาต์พุตของแต่ละ โหนดของชั้นที่ 2-4

6.3.3 การปรับค่าถ่วงน้ำหนักเพื่อใช้ในการหาขอบภาพ

กลไกที่สำคัญของนิวรอลเน็ตเวิร์คก็คือ การปรับค่าถ่วงน้ำหนักเพื่อหาความสัมพันธ์ระหว่างอินพุตกับเอาต์พุตที่ต้องการ โดยยอมให้มีความคลาดเคลื่อนที่น้อยที่สุด ซึ่งหลักการอันนี้ก็คือนี่ก็คือ GDR เนื่องจากในหัวข้อนี้เราได้นำเสนอ DABP เพื่อเพิ่มประสิทธิภาพในการสอนและการจำได้ของตัวเน็ตเวิร์ค

การคำนวณค่าความผิดพลาดเพื่อนำไปปรับค่าถ่วงน้ำหนักใน DABP ค่าความแปรปรวนของผลลัพธ์ในเน็ตเวิร์คจะถูกกำหนดให้เป็นค่าอัตราเร็วในการเรียนรู้ (η_p) โดยค่านี้อาจเปลี่ยนแปลงทุกแพทเทิร์นซึ่งกำหนดได้ดังนี้

$$\eta_p = \frac{E_p}{S} ; E_p = \sum_{p=1}^S (d_p - y_p)^2 \quad (6.14)$$

ที่ E_p เป็นผลรวมกำลังสองของค่าความผิดพลาด (Sum Square Error) ระหว่างค่าผลลัพธ์ที่ต้องการ (ซึ่ง d_p ตัวนี้ก็คือค่าคาดหวังของแพทเทิร์นที่ p) กับค่าผลลัพธ์ที่ได้จากเน็ตเวิร์ค (y_p) ของแพทเทิร์นที่ p ส่วน S เป็นจำนวนแพทเทิร์นที่ได้ทำการสอนไปแล้ว ในสมการที่ (6.15) เป็นสมการที่ใช้ในการปรับค่าถ่วงน้ำหนักที่เวลา $t+1$

$$W_p^k(t+1) = \alpha W_p^k(t) + \eta_p \delta_p^k X_p^{k-1} \quad (6.15)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ X_p^{k-1} เป็นเวกเตอร์ของอินพุตที่ส่งมาจากชั้นที่ $k-1$
 $W_p^k(t+1)$ เป็นเมตริกซ์ของค่าถ่วงน้ำหนักของชั้นที่ k แพทเทิร์นที่ p
 α เป็นค่าโมเมนต์ดัมที่มีการผันแปรค่าตามค่าส่วนเบี่ยงเบนมาตรฐานของแพทเทิร์นทั้งหมดสำหรับการสอนในแต่ละรอบ ซึ่งสามารถคำนวณค่าได้ดังนี้

$$\alpha = \sqrt{\frac{E_p}{P}} \tag{6.16}$$

δ_p^k เป็นเวกเตอร์ของค่าความคลาดเคลื่อนของชั้นที่ k และแพทเทิร์นที่ p ซึ่งในชั้นของผลลัพธ์สามารถคำนวณได้ดังนี้

$$\delta_p^k = (d_p - y_p)y_p(1 - y_p) \tag{6.17}$$

และความคลาดเคลื่อนในชั้นฮิดเดนสามารถกำหนดได้ดังนี้

$$\delta_p^k = \bar{Y}_p^k(1 - \bar{Y}_p^k)(\delta_p^{k+1}W_p^{k+1}) \tag{6.18}$$

เมื่อ \bar{Y}_p^k เป็นเวกเตอร์ของผลลัพธ์ในชั้นที่ k สำหรับแพทเทิร์นที่ p

จากขั้นตอนการหาขอบภาพโดยใช้นิวรอลเน็ตเวิร์คที่มีการสอนแบบ DABP ที่ได้เสนอก่อนหน้านี้สามารถนำมาเขียนเป็นอัลกอริทึมสำหรับใช้สอนโครงข่ายเพื่อให้โครงข่ายรู้จักขอบภาพ ดังอัลกอริทึมที่ 6-3 และสำหรับแผนภูมิการทำงานของอัลกอริทึมนี้ได้แสดงในภาคผนวก ข.

อัลกอริทึมที่ 6-3. แสดงขั้นตอนการสอนโครงข่ายแบบ DABP ให้รู้จักขอบภาพ

```
#define ERROR_MIN 0.001 // กำหนดค่าความผิดพลาดต่ำสุดที่ยอมรับให้เกิดขึ้นได้
#define I 4 // กำหนดจำนวนชั้นของโครงข่าย
#define N 3 // จำนวนรูปที่นำมาใช้สอนโครงข่ายซึ่งมีอยู่ 3 รูป(ตั้งในรูปที่ 6.13)
int KERNEL 3 // กำหนดขนาดของวินโดว์
float mean; // ค่าเฉลี่ยของแต่ละแพทเทิร์น
```

```
void DABP_Training(IMAGE *I)
```

{เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for(int k = 0; k < I-1; k++)
    Initial_Weights(W[k], ROW[i], COL[i]); // กำหนดค่าเริ่มต้นของค่าถ่วงน้ำหนัก

float   error = 1000;
float   *V = new float[KERNEL*KERNEL];
int     D;    // เป็นค่าของผลลัพธ์ที่ต้องการ
while (error > ERROR_MIN)
{
    for(int i = 0; i < I.ROWS; i++)
        for(int j = 0; j < I.COLS; j++)
        {
            int n = random(N); // สุ่มรูปภาพที่จะทำการสอน
            mean = 0;
            for(k = 0; k < KERNEL; k++) // แปลงข้อมูล
                for(int l = 0; l < KERNEL; l++) // ให้อยู่ในรูปของเวกเตอร์
                {
                    V[KERNEL*k+l] = (I[n].data[i+k][j+l]-I[n].MIN)/
                                        (I[n].MAX-I[n].MIN);
                    mean += V[KERNEL*k+l];
                }
            mean /= (KERNEL*KERNEL);
            D = I[n].Output[i+1][j+1];
            Set_Input_Pattern(V, D); // ป้อนอินพุตและเอาต์พุตที่
                                     // ต้องการเข้าสู่โครงข่าย
            Forward_Propagate(); // คำนวณค่าผลลัพธ์ของแต่ละชั้น
            error = SSE(); // คำนวณค่าความผิดพลาด
            Set_Learning_Rate(); // กำหนดค่าอัตราการเรียนรู้
            Adjust_Weights(); // ปรับค่าถ่วงน้ำหนัก
        }
    Set_Momentum(); // กำหนดค่าโมเมนตัม
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการสอนโครงข่ายแบบ DABP ในอัลกอริทึมที่ 6-3. เป็นการปรับปรุงวิธีการสอนแบบ Back Propagation ดังที่ได้เสนอไปแล้วในอัลกอริทึมที่ 4-2. ในบทที่ 4 แต่ในอัลกอริทึมที่ 6-3. จะมีการเพิ่มฟังก์ชันเข้ามาอีกสองฟังก์ชันคือ ฟังก์ชัน Set_Learning_rate และ Set_Momentum ใช้ในการกำหนดค่าอัตราการเรียนรู้และค่าโมเมนต์ดัมโดยสามารถกำหนดได้ดังสมการที่ (6.14) และ (6.16) ตามลำดับ ส่วนฟังก์ชัน Forward_Propagate จะมีการปรับปรุงจากฟังก์ชันเดิมในอัลกอริทึม 4-2. เล็กน้อยซึ่งสามารถคำนวณได้ดังรูปที่ 6.10

6.4 การหาขอบภาพโดยฟิซซีนิวรอลเน็ตเวิร์ค

โครงข่ายฟิซซีนิวรอลที่ได้เสนอไปแล้วในบทที่ 5 จะถูกนำมาประยุกต์ใช้สำหรับหาขอบภาพ โดยใช้ตัวแปรที่ได้จากการวิเคราะห์เนื้อหาภาพเป็นอินพุทของโครงข่าย ซึ่งตัวแปรที่ได้จากการวิเคราะห์เนื้อหาภาพ มีอยู่หลายตัวด้วยกันดังที่กล่าวไว้ในบทที่ 1 แต่การหาขอบภาพโดยวิธีนี้จะเลือกเอาเฉพาะตัวแปรที่มีผลต่อขอบภาพเท่านั้น และตัวแปรนี้จะถูกแปลงไปเป็นค่าความเป็นขอบภาพโดยขั้นตอนของระบบฟิซซี เพื่อให้โครงข่ายฟิซซีนิวรอลเป็นตัวตรวจสอบว่าค่าความเป็นขอบภาพจุดใดบ้างเป็นหรือไม่เป็นขอบภาพ โดยโครงข่ายฟิซซีนิวรอลจะทำหน้าที่เหมือนตัวกำหนดค่าเทรชโฮลด์แบบหลายระดับ ที่สามารถใช้หาขอบภาพที่มีความเข้มต่าง ๆ กันได้ ซึ่งจะทำให้ขอบภาพที่ได้มีความสมบูรณ์ครบถ้วนถึงแม้ขอบของวัตถุในภาพจะมีค่าความเข้มต่าง ๆ กันก็ตาม

6.4.1 การเลือกตัวแปรที่ได้จากเนื้อหาภาพ

ตัวแปรต่าง ๆ ที่ได้จากวิเคราะห์เนื้อหาภาพในบทที่ 1 จะมีคุณสมบัติต่างกันไปในที่นี้เราต้องการตัวแปรที่จะบอกถึงความเป็นขอบภาพซึ่งตัวแปรเหล่านี้ได้แก่ ค่าเอนโทรปี (สมการที่ 6.20), ค่าการแพร่กระจาย (Dispersion; สมการที่ 6.21) และค่าส่วนเบี่ยงเบนมาตรฐาน (สมการที่ 6.22) ส่วนตัวแปรอื่น ๆ จะมีผลต่อขอบภาพน้อยมาก เช่นค่าเฉลี่ย, ค่ามัธยฐาน และค่าฐานนิยม สำหรับค่าตัวแปรแต่ละตัวสามารถคำนวณได้ดังนี้

กำหนดให้	N	เป็นจำนวนข้อมูลทั้งหมด
	H	เป็นอะเรย์ของฮิสโตแกรมข้อมูลภาพ
	Pr	เป็นค่าความน่าจะเป็นของความถี่สะสมของค่าความเข้มจุดภาพ
	X	เป็นอะเรย์ของค่าความเข้มจุดภาพที่มีอยู่ N ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\text{ค่าเฉลี่ย} \quad \bar{X} = \frac{1}{N} \sum_{i=1}^N x_i \quad (6.19)$$

$$\text{ค่าเอนโทรปี} \quad H(Pr_i) = Pr_i * \log(1 - Pr_i) \quad (6.20)$$

การกำหนดค่าความถี่สะสม, ค่าความน่าจะเป็น, ค่าเฉลี่ย (Mean), ค่ามัธยฐาน (Median), ค่าฐานนิยม (Mode), และค่าเอนโทรปีของค่าความเข้มจุดภาพ สามารถคำนวณได้ด้วยอัลกอริทึมที่ 6-4

อัลกอริทึมที่ 6-4 เป็นขั้นตอนการคำนวณค่าตัวแปรของค่าความเข้มจุดภาพ

```
void Histogram(BYTE * X)
{
    for(int i=0; i<N; i++)
        H[i] = 0; // กำหนดค่าเริ่มต้นของอะเรย์
    mean = 0;
    for(int i=0; i<N; i++)
    {
        H[ X[i] ]++; // คำนวณค่าความถี่สะสม
        mean += X[i];
    }
    mean /= N; // คำนวณค่าเฉลี่ย
    gsort(X, N, sizeof(BYTE), Compare); // เรียงลำดับค่าความเข้มจุดภาพ
    median = X[N/2] // คำนวณค่ามัธยฐาน
    int i=0; Entropy=0;
    for(i=0; i<N; i++)
    {
        if(j<H[i]);
        {
            j = H[i];
            mode = X[i]; // คำนวณค่าฐานนิยม
        }
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี การนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย
ไม่ว่าการณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

if(H(i))

Entropy -= Pr[i]*log(1-Pr[i]); // คำนวณค่าเอนโทรปีของข้อมูล

}

}

จากค่าเฉลี่ยและค่าความน่าจะเป็น(Pr_i) ในอัลกอริทึมที่ 6-4. สามารถนำมาคำนวณค่าส่วนเบี่ยงเบนมาตรฐาน (σ) และค่าการแพร่กระจาย (D) ได้ดังนี้

$$D = \sum_{i=1}^N |x_i - \text{mean}| Pr_i \quad (6.21)$$

$$\sigma = \sum_{i=1}^N (x_i - \text{mean})^2 Pr_i \quad (6.22)$$

จากตัวแปรต่าง ๆ ของเนื้อหาภาพที่กล่าวมาแล้วคือ ค่าเฉลี่ย, ค่ามัธยฐาน, ค่าฐานนิยม, ค่าเอนโทรปี, ค่าการแพร่กระจาย และค่าส่วนเบี่ยงเบนมาตรฐาน จะนำมาพิจารณาว่าตัวแปรตัวใดมีผลต่อขอบภาพอย่างไร ซึ่งได้แสดงผลของตัวแปรแต่ละตัวไว้ในรูปที่ 6.11 ซึ่งเป็นภาพเด็กผู้หญิงที่สแกนเข้ามาและภาพมีขนาด 255 x 255 ในการคำนวณค่าตัวแปรต่าง ๆ จะดำเนินการภายในวินโดว์ขนาด 2 x 2 โดยการคอนโวลูชัน


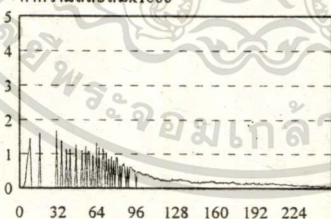

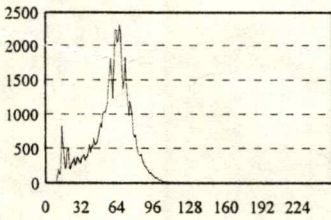
จากรูปที่ 6.11 ภาพที่ได้ในรูป (ข)-(ง) จะเหมือนกับภาพที่ผ่านตัวกรองสัญญาณในการกำจัดสัญญาณรบกวนในบทที่ 3 โดยรูป (ข) เป็นตัวกรองสัญญาณที่เป็นเชิงเส้น ส่วนรูป (ค) และ (ง) จะเป็นตัวกรองสัญญาณแบบไม่เชิงเส้น สำหรับภาพ (จ)-(ช) จะเห็นส่วนที่เป็นขอบภาพชัดเจน โดยเริ่มจากภาพ (จ) ซึ่งเป็นภาพที่ได้จากค่าเอนโทรปี โดยตัวแปรตัวนี้จะอธิบายถึงความเป็นเนื้อเดียวกันของค่าความเข้มจุดภาพภายในวินโดว์ 2 x 2 ถ้าข้อมูลภาพภายในวินโดว์เป็นเนื้อเดียวกัน ค่าเอนโทรปีจะมีค่าต่ำ (ไม่มีขอบภาพอยู่) แต่ถ้าค่าความเข้มของจุดภาพไม่เป็นเนื้อเดียวกันจะทำให้ค่าเอนโทรปีมีค่าสูง ซึ่งอาจจะมีขอบภาพอยู่ในวินโดว์นี้ก็ได้ ซึ่งจะต้องพิจารณาตัวแปรอื่น ๆ ประกอบด้วย และภาพในรูป (จ) ได้จากการนำค่าเอนโทรปีไปคูณกับค่าคงที่ (คือ 128) เพื่อทำการขยายสัญญาณ ภาพ (ฉ) เป็นภาพที่ได้จากตัวแปรที่วัดค่าการแพร่กระจายโดยใช้ค่าเฉลี่ยเป็นค่าจุดศูนย์กลาง ถ้าข้อมูลภายในวินโดว์เป็นเนื้อเดียวกันคือ เกือบเท่ากับค่าเฉลี่ย จะส่งผลให้ค่านี้ต่ำ แต่ถ้าข้อมูลภาพยังห่างจากค่าเฉลี่ยมากเท่าใดจะยิ่งทำให้ค่าการแพร่กระจายสูงขึ้น ซึ่งจะเห็นเป็นขอบขึ้นมาดังแสดงในภาพ (ฉ) และในภาพ (ช) ซึ่งเป็นภาพที่ได้จากค่าส่วนเบี่ยงเบนมาตรฐาน ซึ่งจะมีคุณสมบัติคล้ายกับค่าการแพร่กระจาย แต่ค่านี้จะมีผลต่อขอบภาพมากกว่า [21]

เหมาะสมสำหรับการหาขอบภาพมากกว่าค่าเฉลี่ยและค่าฐานนิยม เนื่องจากค่าเฉลี่ยเป็นค่าที่ประมาณขึ้นมา จากข้อมูลภาพไม่ใช่ค่าความเข้มของจุดภาพจริงๆ ส่วนค่าฐานนิยมจะตอบสนองต่อสัญญาณรบกวนที่มีความถี่สูงถึงแม้จะได้ขอบภาพที่ชัดเจนกว่าก็ตาม

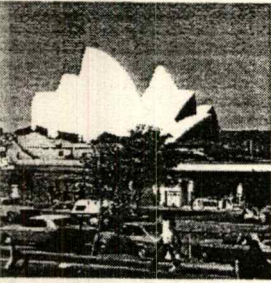
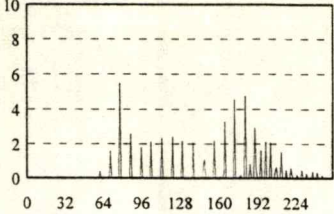
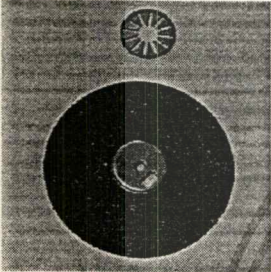
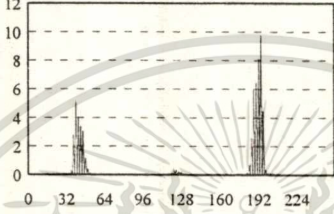

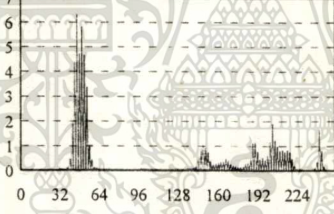
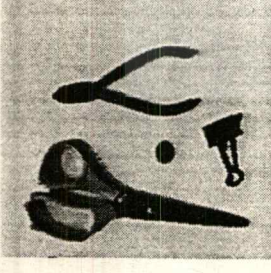
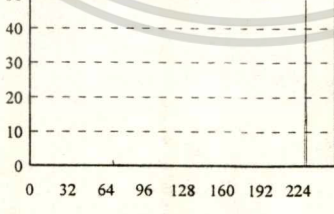
ขั้นตอนสำคัญของการใช้โครงข่ายพีชชีนัวร์อล จะอยู่ที่การเตรียมข้อมูลเพื่อจะส่งเข้าไปในโครงข่ายหรือขั้นตอนการ Fuzzifier นั่นเอง ซึ่งขั้นตอนนี้จะต้องอาศัยความรู้และประสบการณ์เกี่ยวกับระบบนั้นๆ พอสมควรเพื่อจะได้กำหนดฟังก์ชันความเป็นสมาชิกขึ้นมาได้เหมาะสมกับตัวแปรแต่ละตัว จากขั้นตอนที่แล้วเราได้พิจารณาถึงตัวแปรที่จะใช้ในการหาขอบภาพซึ่งมี 3 ตัวด้วยกัน คือ เอนโทรปี, การแพร่กระจาย, และส่วนเบี่ยงเบนมาตรฐาน ซึ่งในหัวข้อนี้จะพิจารณาถึงรายละเอียดของตัวแปรแต่ละตัวเพื่อใช้ในการกำหนดฟังก์ชันความเป็นสมาชิก

ในการสร้างฟังก์ชันของความเป็นสมาชิกของตัวแปรแต่ละตัว จะใช้ข้อมูลภาพจำนวน 30 ภาพ ที่มีการแจกแจงต่างๆ กันไป และในภาพหนึ่งๆ จะมีสองแบบด้วยกันคือแบบที่เป็นภาพปกติและอีกแบบหนึ่งเป็นภาพผกผันของภาพเดียวกันซึ่งภาพนี้จะมีลักษณะตรงกันข้ามกับภาพปกติ เพื่อศึกษาถึงคุณสมบัติของตัวแปรแต่ละตัวว่ามีการเปลี่ยนแปลงค่าอย่างไร ในตารางที่ 6.1 แสดงค่าตัวแปรแต่ละตัวของข้อมูลภาพซึ่งได้นำมาแสดงไว้เพียง 20 ภาพ จากข้อมูลภาพทั้งหมด 30 ภาพ

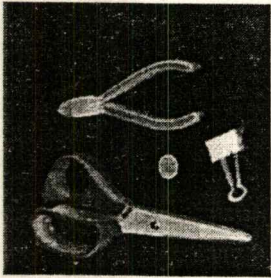
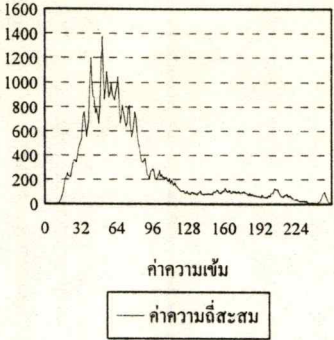
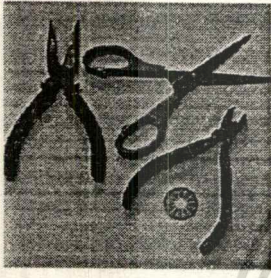
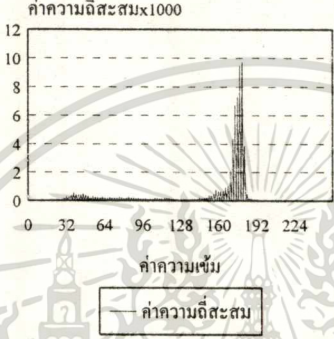
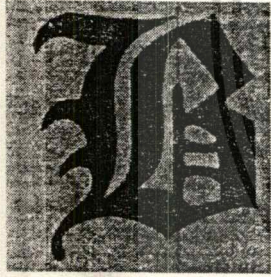
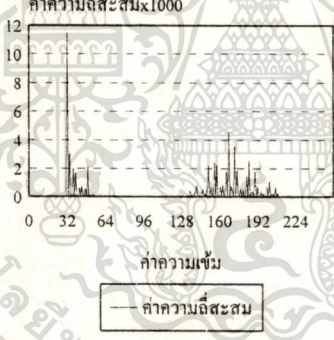
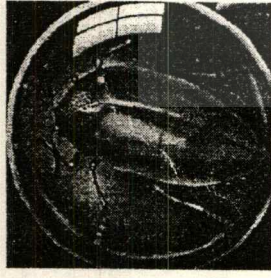
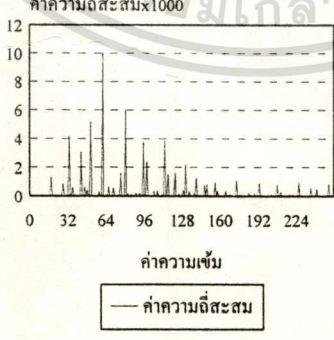
ตารางที่ 6.1 แสดงค่าตัวแปรแต่ละตัวของข้อมูลภาพ

รูปภาพ	ฮิสโตแกรม	ค่าตัวแปร
	<p>ค่าความถี่สะสม x 1000</p>  <p>ค่าความเข้ม</p> <p>— ค่าความถี่สะสม</p>	<p>ค่าของตัวแปรทั้งภาพ</p> <p>ค่าต่ำสุด=3 ค่าสูงสุด=255 ค่าเฉลี่ย=109.27 ค่ามัชฌิมฐาน=157 ค่าฐานนิยม=255 ค่าการแพร่กระจาย=57.58 ค่าส่วนเบี่ยงเบนมาตรฐาน=70.27</p> <p>ค่าของตัวแปรภายในวินโดว์ขนาด 2 x 2</p> <p>ค่าเอนโทรปี = [0, 1.39] ค่าการแพร่กระจาย = [0, 76.0] ส่วนเบี่ยงเบนมาตรฐาน = [0, 85.51]</p>
	<p>ความถี่สะสม</p>  <p>ค่าความเข้ม</p> <p>— ค่าความถี่สะสม</p>	<p>ค่าของตัวแปรทั้งภาพ</p> <p>ค่าต่ำสุด = 9 ค่าสูงสุด = 120 ค่าเฉลี่ย = 59.00 ค่ามัชฌิมฐาน = 64 ค่าฐานนิยม = 67 ค่าการแพร่กระจาย = 14.00 ค่าส่วนเบี่ยงเบนมาตรฐาน = 18.00</p> <p>ค่าของตัวแปรภายในวินโดว์ขนาด 2 x 2</p> <p>ค่าเอนโทรปี = [0, 1.39] ค่าการแพร่กระจาย = [0, 76.0] ส่วนเบี่ยงเบนมาตรฐาน = [0, 85.51]</p>

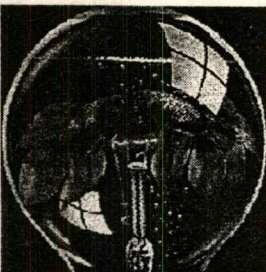
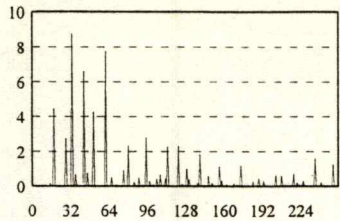
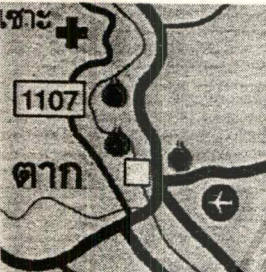
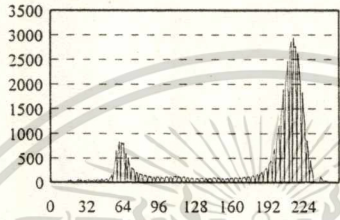

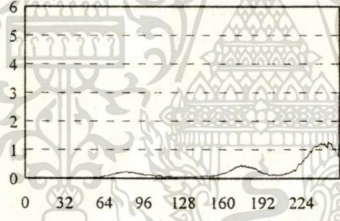

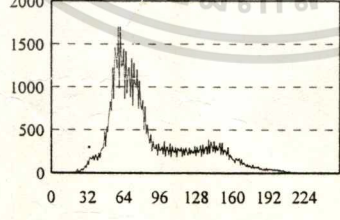
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น เมื่อมีเวลาให้เข้าไปใช้ประโยชน์ด้านการค้า
 เมื่อกิจกรรมใดทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปดสิ่งเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	<p>ค่าความถี่สะสมx1000</p>  <p>ค่าความถี่</p> <p>— ค่าความถี่สะสม</p>	<p>ค่าของตัวแปรทั้งภาพ</p> <p>ค่าต่ำสุด = 46 ค่าสูงสุด = 255 ค่าเฉลี่ย = 165.00 ค่ามัธยฐาน = 186 ค่าฐานนิยม = 255 ค่าการแพร่กระจาย = 47.52 ค่าส่วนเบี่ยงเบนมาตรฐาน = 56.93</p> <p>ค่าของตัวแปรภายในวินโดว์ขนาด 2 x 2</p> <p>ค่าเอนโทรปี = [0, 1.39] ค่าการแพร่กระจาย = [0, 92.50] ส่วนเบี่ยงเบนมาตรฐาน = [0, 92.50]</p>
	<p>ค่าความถี่สะสมx1000</p>  <p>ค่าความถี่</p> <p>— ค่าความถี่สะสม</p>	<p>ค่าของตัวแปรทั้งภาพ</p> <p>ค่าต่ำสุด = 2 ค่าสูงสุด = 250 ค่าเฉลี่ย = 138.5 ค่ามัธยฐาน = 124 ค่าฐานนิยม = 194 ค่าการแพร่กระจาย = 65.75 ค่าส่วนเบี่ยงเบนมาตรฐาน = 69.87</p> <p>ค่าของตัวแปรภายในวินโดว์ขนาด 2 x 2</p> <p>ค่าเอนโทรปี = [0, 1.39] ค่าการแพร่กระจาย = [0, 110.00] ส่วนเบี่ยงเบนมาตรฐาน = [0, 110.00]</p>
	<p>ค่าความถี่สะสมx1000</p>  <p>ค่าความถี่</p> <p>— ค่าความถี่สะสม</p>	<p>ค่าของตัวแปรทั้งภาพ</p> <p>ค่าต่ำสุด = 0 ค่าสูงสุด = 254 ค่าเฉลี่ย = 114.9 ค่ามัธยฐาน = 130 ค่าฐานนิยม = 46 ค่าการแพร่กระจาย = 70.88 ค่าส่วนเบี่ยงเบนมาตรฐาน = 73.89</p> <p>ค่าของตัวแปรภายในวินโดว์ขนาด 2 x 2</p> <p>ค่าเอนโทรปี = [0, 1.39] ค่าการแพร่กระจาย = [0, 115.50] ส่วนเบี่ยงเบนมาตรฐาน = [0, 115.50]</p>
	<p>ค่าความถี่สะสมx1000</p>  <p>ค่าความถี่</p> <p>— ค่าความถี่สะสม</p>	<p>ค่าของตัวแปรทั้งภาพ</p> <p>ค่าต่ำสุด = 0 ค่าสูงสุด = 255 ค่าเฉลี่ย = 204 ค่ามัธยฐาน = 125 ค่าฐานนิยม = 230 ค่าการแพร่กระจาย = 40.74 ค่าส่วนเบี่ยงเบนมาตรฐาน = 54.81</p> <p>ค่าของตัวแปรภายในวินโดว์ขนาด 2 x 2</p> <p>ค่าเอนโทรปี = [0, 1.39] ค่าการแพร่กระจาย = [0, 66.75] ส่วนเบี่ยงเบนมาตรฐาน = [0, 53.39]</p>

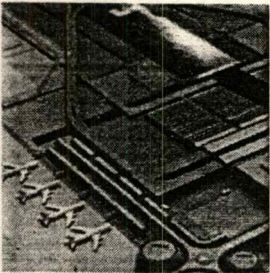
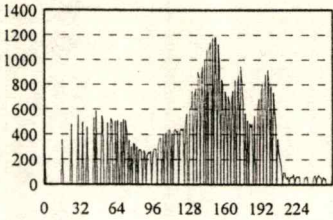

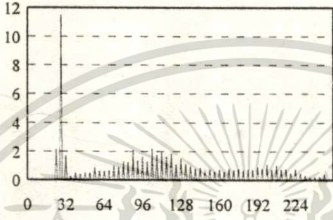

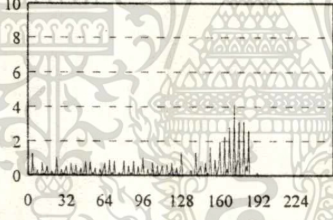

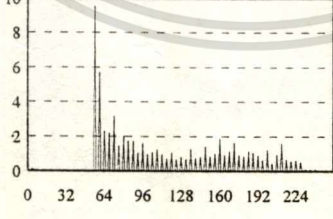
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		<p>ค่าของตัวแปรทั้งภาพ</p> <p>ค่าต่ำสุด = 0 ค่าสูงสุด = 255 ค่าเฉลี่ย = 77.41 ค่ามัธยฐาน = 130 ค่าฐานนิยม = 51 ค่าการแพร่กระจาย = 34.47 ค่าส่วนเบี่ยงเบนมาตรฐาน = 47.19</p> <p>ค่าของตัวแปรภายในวินโดวขนาด 2 x 2</p> <p>ค่าเอนโทรปี = [0, 1.39] ค่าการแพร่กระจาย = [0, 60.75] ส่วนเบี่ยงเบนมาตรฐาน = [0, 65.80]</p>
		<p>ค่าของตัวแปรทั้งภาพ</p> <p>ค่าต่ำสุด = 6 ค่าสูงสุด = 254 ค่าเฉลี่ย = 153.85 ค่ามัธยฐาน = 122 ค่าฐานนิยม = 180 ค่าการแพร่กระจาย = 33.62 ค่าส่วนเบี่ยงเบนมาตรฐาน = 44.57</p> <p>ค่าของตัวแปรภายในวินโดวขนาด 2 x 2</p> <p>ค่าเอนโทรปี = [0, 1.39] ค่าการแพร่กระจาย = [0, 79.0] ส่วนเบี่ยงเบนมาตรฐาน = [0, 91.27]</p>
		<p>ค่าของตัวแปรทั้งภาพ</p> <p>ค่าต่ำสุด = 22 ค่าสูงสุด = 224 ค่าเฉลี่ย = 120.80 ค่ามัธยฐาน = 123 ค่าฐานนิยม = 32 ค่าการแพร่กระจาย = 61.13 ค่าส่วนเบี่ยงเบนมาตรฐาน = 65.33</p> <p>ค่าของตัวแปรภายในวินโดวขนาด 2 x 2</p> <p>ค่าเอนโทรปี = [0, 1.39] ค่าการแพร่กระจาย = [0, 79.60] ส่วนเบี่ยงเบนมาตรฐาน = [0, 58.75]</p>
		<p>ค่าของตัวแปรทั้งภาพ</p> <p>ค่าต่ำสุด = 0 ค่าสูงสุด = 251 ค่าเฉลี่ย = 93.79 ค่ามัธยฐาน = 129 ค่าฐานนิยม = 61 ค่าการแพร่กระจาย = 41.87 ค่าส่วนเบี่ยงเบนมาตรฐาน = 53.11</p> <p>ค่าของตัวแปรภายในวินโดวขนาด 2 x 2</p> <p>ค่าเอนโทรปี = [0, 1.39] ค่าการแพร่กระจาย = [0, 97.50] ส่วนเบี่ยงเบนมาตรฐาน = [0, 97.50]</p>


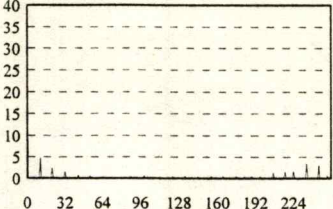
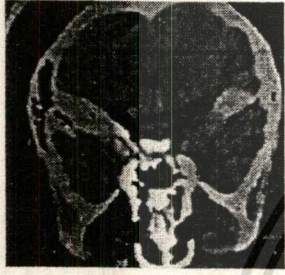
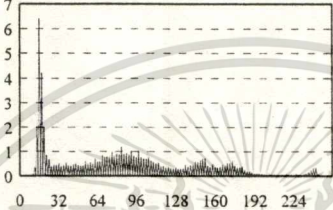
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	<p>ค่าความถี่สะสมx1000</p>  <p>ค่าความเข้ม</p> <p>— ค่าความถี่สะสม</p>	<p>ค่าของตัวแปรทั้งภาพ</p> <p>ค่าต่ำสุด = 0 ค่าสูงสุด = 251 ค่าเฉลี่ย = 84.35 ค่ามัธยฐาน = 131 ค่าฐานนิยม = 33 ค่าการแพร่กระจาย = 50.40 ค่าส่วนเบี่ยงเบนมาตรฐาน = 61.92</p> <p>ค่าของตัวแปรภายในวินโดว์ขนาด 2 x 2</p> <p>ค่าเอนโทรปี = [0, 1.39] ค่าการแพร่กระจาย = [0, 116.5] ส่วนเบี่ยงเบนมาตรฐาน = [0, 116.5]</p>
	<p>ความถี่สะสม</p>  <p>ค่าความเข้ม</p> <p>— ค่าความถี่สะสม</p>	<p>ค่าของตัวแปรทั้งภาพ</p> <p>ค่าต่ำสุด = 0 ค่าสูงสุด = 255 ค่าเฉลี่ย = 179.20 ค่ามัธยฐาน = 128 ค่าฐานนิยม = 215 ค่าการแพร่กระจาย = 48.70 ค่าส่วนเบี่ยงเบนมาตรฐาน = 59.50</p> <p>ค่าของตัวแปรภายในวินโดว์ขนาด 2 x 2</p> <p>ค่าเอนโทรปี = [0, 1.39] ค่าการแพร่กระจาย = [0, 62.5] ส่วนเบี่ยงเบนมาตรฐาน = [0, 89.71]</p>
	<p>ความถี่สะสมx1000</p>  <p>ค่าความเข้ม</p> <p>— ค่าความถี่สะสม</p>	<p>ค่าของตัวแปรทั้งภาพ</p> <p>ค่าต่ำสุด = 0 ค่าสูงสุด = 255 ค่าเฉลี่ย = 200.59 ค่ามัธยฐาน = 128 ค่าฐานนิยม = 255 ค่าการแพร่กระจาย = 48.25 ค่าส่วนเบี่ยงเบนมาตรฐาน = 59.27</p> <p>ค่าของตัวแปรภายในวินโดว์ขนาด 2 x 2</p> <p>ค่าเอนโทรปี = [0, 1.39] ค่าการแพร่กระจาย = [0, 73.89] ส่วนเบี่ยงเบนมาตรฐาน = [0, 76.75]</p>
	<p>ความถี่สะสม</p>  <p>ค่าความเข้ม</p> <p>— ค่าความถี่สะสม</p>	<p>ค่าของตัวแปรทั้งภาพ</p> <p>ค่าต่ำสุด = 0 ค่าสูงสุด = 255 ค่าเฉลี่ย = 86.82 ค่ามัธยฐาน = 131 ค่าฐานนิยม = 58 ค่าการแพร่กระจาย = 34.14 ค่าส่วนเบี่ยงเบนมาตรฐาน = 43.86</p> <p>ค่าของตัวแปรภายในวินโดว์ขนาด 2 x 2</p> <p>ค่าเอนโทรปี = [0, 1.39] ค่าการแพร่กระจาย = [0, 58.66] ส่วนเบี่ยงเบนมาตรฐาน = [0, 40.00]</p>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	<p>ความถี่สะสม</p>  <p>ค่าความเข้ม</p> <p>— ค่าความถี่สะสม</p>	<p>ค่าของตัวแปรทั้งภาพ</p> <p>ค่าต่ำสุด = 0 ค่าสูงสุด = 255 ค่าเฉลี่ย = 137.73 ค่ามัธยฐาน = 146 ค่าฐานนิยม = 152 ค่าการแพร่กระจาย = 40.45 ค่าส่วนเบี่ยงเบนมาตรฐาน = 50.44</p> <p>ค่าของตัวแปรภายในวินโดว์ขนาด 2 x 2</p> <p>ค่าเอนโทรปี = [0, 1.39] ค่าการแพร่กระจาย = [0, 99.5] ส่วนเบี่ยงเบนมาตรฐาน = [0, 99.5]</p>
	<p>ความถี่สะสมx1000</p>  <p>ค่าความเข้ม</p> <p>— ค่าความถี่สะสม</p>	<p>ค่าของตัวแปรทั้งภาพ</p> <p>ค่าต่ำสุด = 0 ค่าสูงสุด = 252 ค่าเฉลี่ย = 106.6 ค่ามัธยฐาน = 128 ค่าฐานนิยม = 28 ค่าการแพร่กระจาย = 53.09 ค่าส่วนเบี่ยงเบนมาตรฐาน = 63.93</p> <p>ค่าของตัวแปรภายในวินโดว์ขนาด 2 x 2</p> <p>ค่าเอนโทรปี = [0, 1.39] ค่าการแพร่กระจาย = [0, 98.0] ส่วนเบี่ยงเบนมาตรฐาน = [0, 98.0]</p>
	<p>ความถี่สะสมx1000</p>  <p>ค่าความเข้ม</p> <p>— ค่าความถี่สะสม</p>	<p>ค่าของตัวแปรทั้งภาพ</p> <p>ค่าต่ำสุด = 0 ค่าสูงสุด = 252 ค่าเฉลี่ย = 106.53 ค่ามัธยฐาน = 100 ค่าฐานนิยม = 0 ค่าการแพร่กระจาย = 59.60 ค่าส่วนเบี่ยงเบนมาตรฐาน = 66.43</p> <p>ค่าของตัวแปรภายในวินโดว์ขนาด 2 x 2</p> <p>ค่าเอนโทรปี = [0, 1.39] ค่าการแพร่กระจาย = [0, 68.0] ส่วนเบี่ยงเบนมาตรฐาน = [0, 78.17]</p>
	<p>ความถี่สะสมx1000</p>  <p>ค่าความเข้ม</p> <p>— ค่าความถี่สะสม</p>	<p>ค่าของตัวแปรทั้งภาพ</p> <p>ค่าต่ำสุด = 0 ค่าสูงสุด = 244 ค่าเฉลี่ย = 114.73 ค่ามัธยฐาน = 140 ค่าฐานนิยม = 56 ค่าการแพร่กระจาย = 48.65 ค่าส่วนเบี่ยงเบนมาตรฐาน = 55.03</p> <p>ค่าของตัวแปรภายในวินโดว์ขนาด 2 x 2</p> <p>ค่าเอนโทรปี = [0, 1.39] ค่าการแพร่กระจาย = [0, 118.0] ส่วนเบี่ยงเบนมาตรฐาน = [0, 118.0]</p>

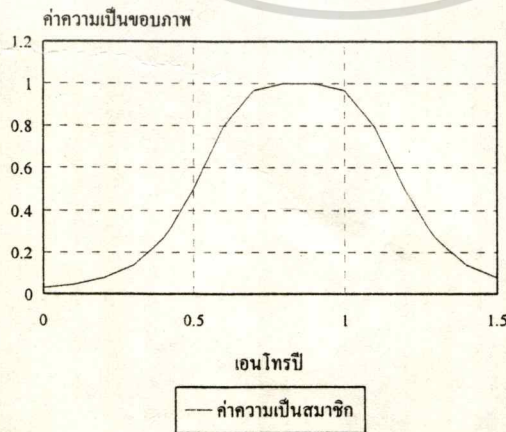
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	<p>ความถี่สะสม x 1000</p>  <p>ค่าความเข้ม</p> <p>— ค่าความถี่สะสม</p>	<p>ค่าของตัวแปรทั้งภาพ</p> <p>ค่าต่ำสุด = 0 ค่าสูงสุด = 252</p> <p>ค่าเฉลี่ย = 63.53 ค่ามัธยฐาน = 147</p> <p>ค่าฐานนิยม = 0 ค่าการแพร่กระจาย = 80.76</p> <p>ค่าส่วนเบี่ยงเบนมาตรฐาน = 92.70</p> <p>ค่าของตัวแปรภายในวินโดวขนาด 2 x 2</p> <p>ค่าเอนโทรปี = [0, 1.39]</p> <p>ค่าการแพร่กระจาย = [0, 77.0]</p> <p>ส่วนเบี่ยงเบนมาตรฐาน = [0, 87.0]</p>
	<p>ค่าความถี่สะสม x 1000</p>  <p>ค่าความเข้ม</p> <p>— ค่าความถี่สะสม</p>	<p>ค่าของตัวแปรทั้งภาพ</p> <p>ค่าต่ำสุด = 6 ค่าสูงสุด = 248</p> <p>ค่าเฉลี่ย = 85.00 ค่ามัธยฐาน = 126</p> <p>ค่าฐานนิยม = 16 ค่าการแพร่กระจาย = 49.2</p> <p>ค่าส่วนเบี่ยงเบนมาตรฐาน = 59.52</p> <p>ค่าของตัวแปรภายในวินโดวขนาด 2 x 2</p> <p>ค่าเอนโทรปี = [0, 1.39]</p> <p>ค่าการแพร่กระจาย = [0, 75.0]</p> <p>ส่วนเบี่ยงเบนมาตรฐาน = [0, 75.0]</p>

จากการสังเกตค่าตัวแปรแต่ละตัวในตารางที่ 6.1 สามารถนำมาสร้างเป็นฟังก์ชันความเป็นสมาชิกของตัวแปรแต่ละตัวได้ดังนี้

ค่าเอนโทรปีของภาพจะอยู่ในช่วงกลาง ๆ ของข้อมูลดังนั้นจึงใช้ฟังก์ชัน β ดังสมการที่ (6.23) และลักษณะของฟังก์ชันแสดงในรูปที่ 6.13

$$\mu_H(x) = \frac{1}{1 + \frac{(x - 0.85)^4}{0.015}} \tag{6.23}$$

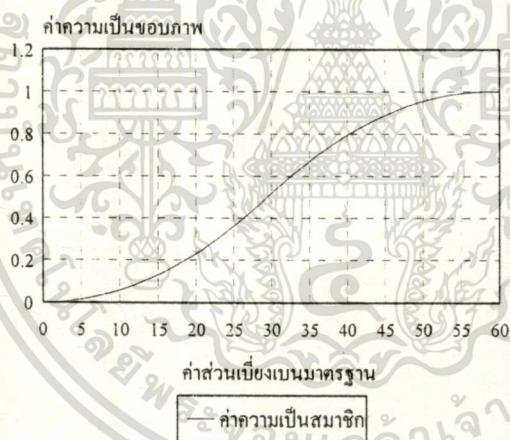


เอกสารนี้เป็นเอกสารที่ 6.13 แสดงฟังก์ชันความเป็นสมาชิกของตัวแปรเอนโทรปีให้นำไปใช้ประโยชน์ด้านการคำนวณว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับค่าการแพร่กระจายและส่วนเบี่ยงเบนมาตรฐานจะมีคุณสมบัติเหมือนกันคือ ถ้าตัวแปรเหล่านี้ยังมีค่าสูงย่อมแสดงว่าข้อมูลในวินโดว์มีความแตกต่างกัน ซึ่งในบริเวณนี้อาจจะมีขอบภาพอยู่ (ซึ่งจะต้องพิจารณาขอบภาพในบริเวณใกล้เคียงด้วย เพราะถ้ามีค่าสูงอยู่เพียงจุดเดียวโดยไม่เชื่อมต่อกับขอบภาพจุดอื่นค่านี้จะกลายเป็นสัญญาณการรบกวนไป) จากคุณสมบัติอันนี้จึงเลือกใช้ฟังก์ชัน S ในการคำนวณค่าความเป็นสมาชิก ซึ่งสามารถกำหนดได้ดังนี้

$$\mu_A(x) = \begin{cases} 0; & \text{ถ้า } x \leq \min \\ 2 \left(\frac{x - \min}{\max - \min} \right)^2; & \text{ถ้า } \min < x \leq \text{mid} \\ 1 - 2 \left(\frac{x - \max}{\max - \min} \right)^2; & \text{ถ้า } \text{mid} < x \leq \max \\ 1; & \text{ถ้า } x > \max \end{cases} \quad (6.24)$$

โดยที่ $\text{mid} = (\min + \max)/2$ และลักษณะของฟังก์ชันนี้แสดงในรูปที่ 6.14

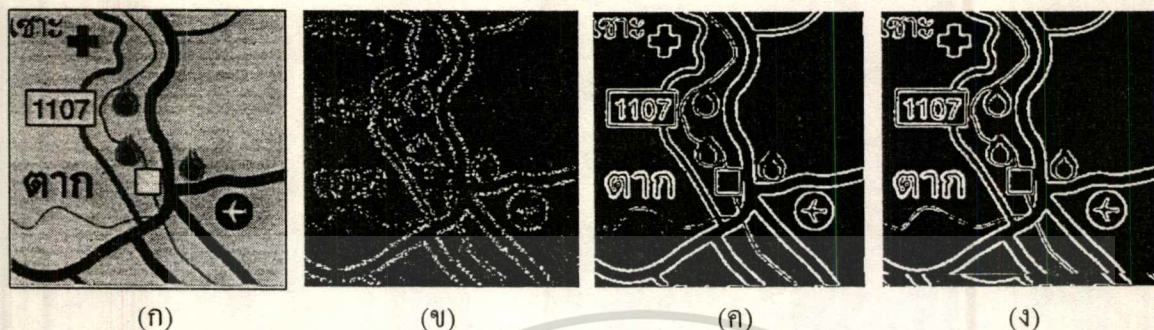


รูปที่ 6.14 แสดงกราฟฟังก์ชันความเป็นสมาชิกของตัวแปรการแพร่กระจายและส่วนเบี่ยงเบนมาตรฐาน

ค่า \min และ \max ในสมการที่ (6.24) คือค่าต่ำสุด และค่าสูงสุดของค่าการแพร่กระจาย หรือค่าส่วนเบี่ยงเบนมาตรฐานที่ได้จากข้อมูลภาพทั้งภาพตามลำดับ

เมื่อออกแบบฟังก์ชันความเป็นสมาชิกของแต่ละตัวแปรได้แล้วก็จะทำการเลือกว่าตัวแปรตัวใดมีความเหมาะสมสำหรับการหาขอบภาพมากที่สุด โดยใช้โครงข่ายฟัซซีนิวรอล โดยนำตัวแปรทั้งสามตัวมาทดสอบกับโครงข่ายฟัซซีนิวรอลที่ได้เสนอไว้แล้วในบทที่ 5 จากการทดสอบพบว่าค่าส่วนเบี่ยงเบนมาตรฐานมีความเหมาะสมในการหาขอบภาพมากที่สุด, รองลงมาคือค่าการแพร่กระจาย, และค่าเอนโทรปีจะมีความเหมาะสมกับการหาขอบภาพโดยโครงข่ายฟัซซีนิวรอลน้อยที่สุด ดังแสดงในรูปที่ 6.15 และเมื่อเลือกเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวแปรที่จะใช้ในการหาขอบภาพโดยใช้โครงข่ายฟัซซีนิรอลได้แล้ว ขั้นตอนต่อไปจะเป็นการสอนโครงข่ายฟัซซีนิรอลให้สามารถหาขอบภาพได้



รูปที่ 6.15 แสดงผลของตัวแปรแต่ละตัวที่ใช้โครงข่ายฟัซซีนิรอลหาขอบภาพ
 (ก) ภาพอินพุต (ข) ขอบภาพที่ได้จากตัวแปรเอนโทรปี
 (ค) ขอบภาพที่ได้จากตัวแปรค่าการกระจาย (ง) ขอบภาพที่ได้จากค่าส่วนเบี่ยงเบนมาตรฐาน

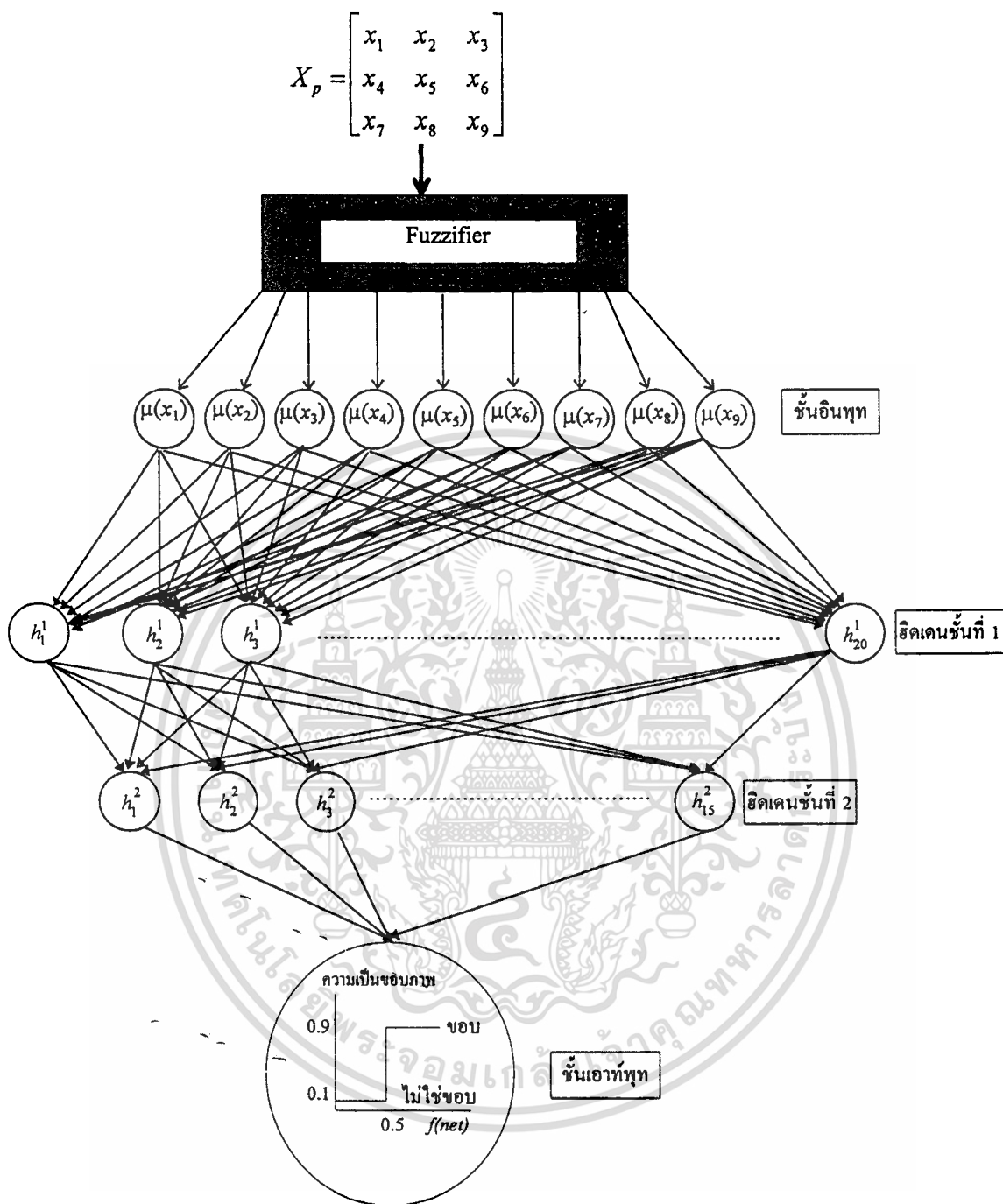
6.4.2 การสอนโครงข่ายฟัซซีนิรอลให้รู้จักขอบภาพ

การสอนโครงข่ายฟัซซีนิรอลเน็ตเวิร์คให้รู้จักขอบภาพ จะต้องใช้แพทเทิร์นในการสอนซึ่งเป็นแพทเทิร์นเดียวกับที่ใช้สอนนิรอลเน็ตเวิร์คให้หาขอบภาพในหัวข้อ 6.3 ซึ่งแสดงในรูปที่ 6.8 แพทเทิร์นส่วนที่เป็นอินพุตทั้ง 3 รูป จะถูกนำมาหาเมตริกซ์ค่าส่วนเบี่ยงเบนมาตรฐานภายในวินโดว์ขนาด 2×2 โดยการคอนโวลูชันทุกจุดภาพและผลลัพธ์ที่ได้มี 3 เมตริกซ์ที่มีขนาดเดียวกับข้อมูลภาพ เมตริกซ์ของค่าส่วนเบี่ยงเบนมาตรฐานทั้ง 3 เมตริกซ์นี้จะถูกป้อนเข้าสู่โครงข่ายฟัซซีนิรอล โดยจะป้อนเป็นเมตริกซ์ย่อยขนาด 3×3 ในลักษณะของการคอนโวลูชันเหมือนกับการป้อนอินพุตของนิรอลเน็ตเวิร์ค (ในหัวข้อ 6.3.1) ต่างกันเพียงแต่โครงข่ายฟัซซีนิรอลนั้นอินพุตจะเป็นค่าความเป็นขอบภาพ ส่วนนิรอลเน็ตเวิร์คอินพุตจะเป็นค่าความเข้มของจุดภาพ และเมตริกซ์ย่อยของค่าส่วนเบี่ยงเบนมาตรฐานจะถูกแปลงเป็นเวกเตอร์ของความ เป็นขอบภาพโดยใช้ฟังก์ชันความเป็นสมาชิกในสมการที่ (6.24) ดังแสดงในรูปข้างล่าง

$$X_p = \begin{bmatrix} x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 \\ x_7 & x_8 & x_9 \end{bmatrix} \xrightarrow{\text{Fuzzify}} \bar{V}_p = \{\mu(x_1), \mu(x_2), \mu(x_3), \dots, \mu(x_5), \dots, \mu(x_9)\}$$

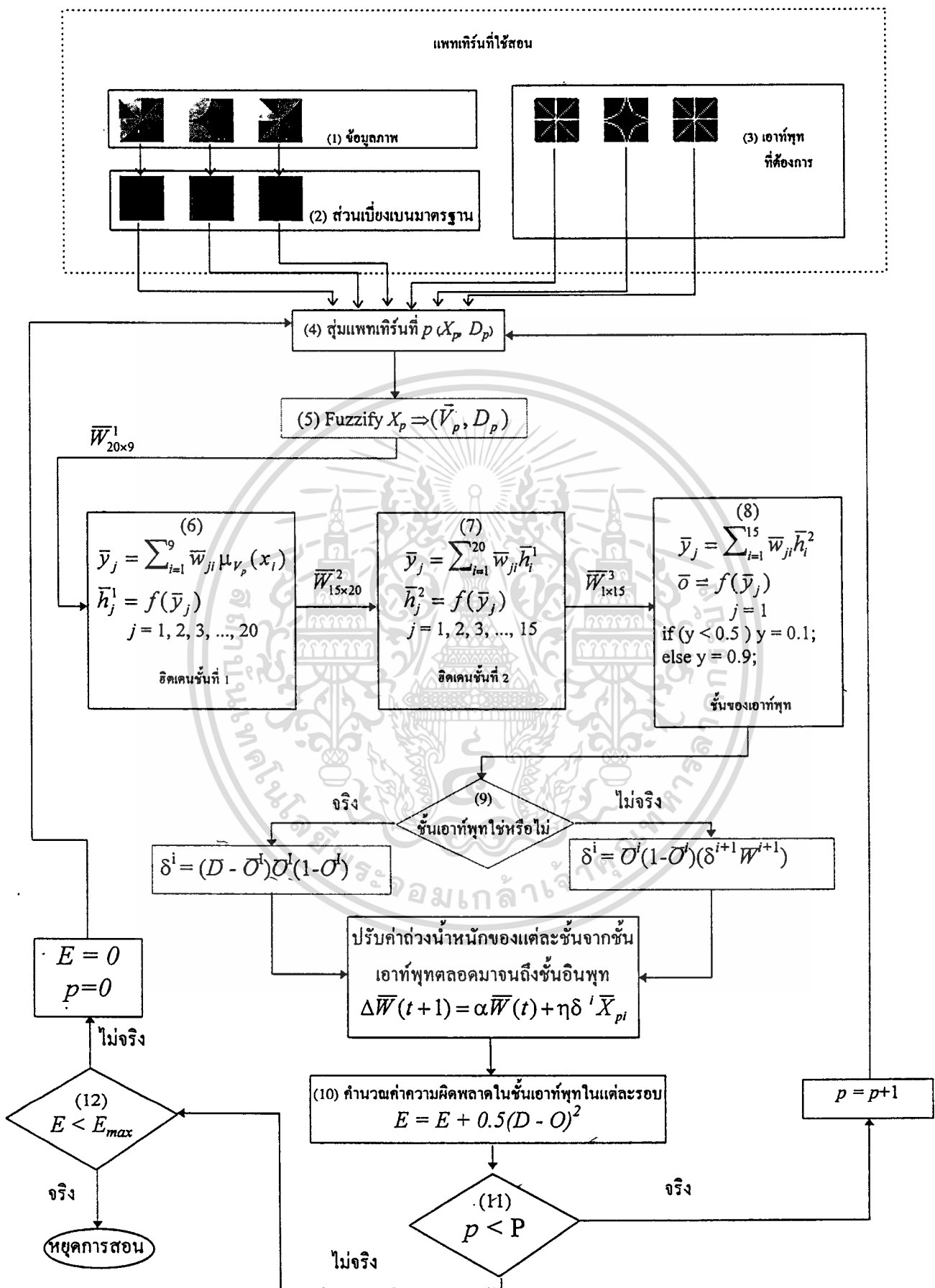
รูปที่ 6.16 แสดงขั้นตอนการแปลงที่เมตริกซ์ย่อยของค่าส่วนเบี่ยงเบนมาตรฐานไปเป็นค่าความเป็นสมาชิกของขอบภาพที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

\bar{V}_p เป็นเวกเตอร์ของค่าความเป็นขอบภาพที่จะใช้ป้อนเข้าสู่โครงข่ายฟัซซีนิวรอล ซึ่งเวกเตอร์นี้มีขนาดเดียวกับอินพุทเวกเตอร์ \bar{V}_i ในรูปที่ 6.7 และค่าของ \bar{V}_p ก็อยู่ในช่วงเดียวกันด้วยคือ $\mu_{V_p}(x_i) \in [0,1]$ ดังนั้นส่วนที่เป็นโครงข่ายจึงมีขนาดเดียวกันด้วยคือ ชั้นอินพุทมี 9 โหนด, ฮิดเดนชั้นที่ 1 มี 20 โหนด, ฮิดเดนชั้นที่ 2 มี 15 โหนด และชั้นเอาต์พุทมีโหนดเดียว สำหรับการเชื่อมโยงของโครงข่ายฟัซซีนิวรอลก็จะเหมือนกับโครงข่ายในรูปที่ 6.9 สำหรับโครงข่ายฟัซซีนิวรอลที่ใช้ในการหาขอบภาพซึ่งมีอยู่ 4 ชั้นได้แสดงในรูปที่ 6.17 โดยโครงข่ายชั้นอินพุทจะรับข้อมูลจากขั้นตอนการ Fuzzify ซึ่งค่าที่รับมาแต่ละตัวจะแสดงถึงความเป็นขอบภาพของจุดภาพแต่ละจุดที่สอดคล้องกับภาพอินพุทที่ตำแหน่งเดียวกัน และค่าเหล่านี้จะถูกส่งต่อไปยังชั้นฮิดเดนโดยผ่านฟังก์ชันการแปลง (เมตริกซ์ของค่าถ่วงน้ำหนักซึ่งมีขนาด $20 \times 9 : \bar{W}_{20 \times 9}^1$) ที่เชื่อมโยงระหว่างชั้นอินพุทกับฮิดเดนชั้นที่ 1 และเมื่อแต่ละโหนดในฮิดเดนชั้นที่ 1 ได้รับสัญญาณแล้วก็จะส่งต่อไปยังฮิดเดนชั้นที่ 2 โดยผ่านค่าถ่วงน้ำหนักที่เป็นเมตริกซ์ขนาด 15×20 ($\bar{W}_{15 \times 20}^2$) ซึ่งเชื่อมต่อจากชั้นฮิดเดนที่ 1 ไปยังชั้นฮิดเดนที่ 2 และเมื่อทุกโหนดในชั้นฮิดเดนที่ 2 ได้รับสัญญาณแล้วก็จะส่งต่อไปยังชั้นเอาต์พุท โดยผ่านเมตริกซ์ของค่าถ่วงน้ำหนักที่มีขนาด 1×15 ($\bar{W}_{1 \times 15}^3$) และเฉพาะในชั้นโหนดเอาต์พุทจะมีการทรานส์โวลต์สัญญาณ $f(net)$ คือถ้า $f(net)$ มากกว่าหรือเท่ากับ 0.5 แสดงว่าอินพุทที่ป้อนเข้ามาเป็นขอบภาพ แต่ถ้า $f(net)$ น้อยกว่า 0.5 แสดงว่าอินพุทที่เข้ามาไม่ใช่ขอบภาพ



รูปที่ 6.17 แสดงโครงข่ายฟัซซีนิวรอลที่ใช้ในการหาขอบภาพ

สำหรับขั้นตอนการสอนโครงข่ายฟัซซีนิวรอลจะแสดงในแผนภูมิรูปที่ 6.18 ส่วนอัลกอริทึมในการสอนโครงข่ายนี้ได้แสดงในภาคผนวก ค.



รูปที่ 6.18 แผนภูมิการสอนโครงข่ายพีชชีนินทรอลที่ใช้กฎการเรียนรู้เคลด้าแบบพีชชี

สำหรับขั้นตอนการสอนโครงข่ายพีชคณิตนิเวรอลในรูปที่ 6.18 สามารถอธิบายได้ดังนี้

- (1) เป็นข้อมูลที่ใช้สอนโครงข่ายซึ่งเป็นข้อมูลเดียวกับที่ใช้สอนนิเวรอลเน็ตเวิร์คในหัวข้อ 6.3
- (2) แปลงข้อมูลภาพในขั้นตอนที่ (1) ให้เป็นค่าส่วนเบี่ยงเบนมาตรฐานของข้อมูลภาพภายในวินโดว์ขนาด 2×2 โดยใช้ค่ามัธยฐานเป็นค่าศูนย์กลางแทนการใช้ค่าเฉลี่ย
- (3) ขอบภาพที่ต้องการ โดยแต่ละภาพจะสอดคล้องกับภาพที่ใช้สอนในข้อ (1)
- (4) ทำการสุ่มข้อมูลภาพที่ถูกแปลงเป็นค่าส่วนเบี่ยงเบนมาตรฐานในข้อ (2) ทั้ง 3 ภาพ ถ้าสุ่มได้รูปใดก็จะดึงข้อมูลของขอบในข้อ (3) ที่สอดคล้องกันมา เพื่อระบุว่าแพทเทิร์นที่ได้มาเป็นขอบภาพหรือไม่ คือถ้าสุ่มได้รูปที่ 1 ก็จะดึงข้อมูลของขอบภาพในรูปที่ 1 เช่นกัน, ถ้าสุ่มได้รูปที่ 2 ก็จะดึงข้อมูลของขอบภาพในรูปที่ 2, และถ้าสุ่มได้รูปที่ 3 ก็จะดึงข้อมูลของขอบภาพในรูปที่ 3 เมื่อรู้ว่าจะใช้ข้อมูลจากรูปใดแล้ว ก็ทำการดึงแพทเทิร์นที่ p โดยที่ p เป็นดัชนีของแพทเทิร์นที่ใช้สอนดังแสดงในรูปที่ 6.19 ($p = 1, 2, 3, \dots, P$, เมื่อ P เป็นจำนวนแพทเทิร์นทั้งหมดที่ใช้สอน ซึ่งในที่นี้ $P = 900$ ($M-2 \times N-2 = 30 \times 30$), $M \times N$ เป็นขนาดของรูปที่ใช้สอน)

	0	1	2	3	29	30	31
0								
1		1	2	3	29	30	
2		31	32	33	59	60	
3		61	62	63	89	90	
.		
.		
.		
29		831	832	833	859	860	
30		861	862	863	899	900	
31								

รูปที่ 6.19 แสดงแพทเทิร์นที่ จากข้อมูลภาพขนาด 32×32

และทำการดึงข้อมูลที่ล้อมรอบ p อีก 8 จุด จากข้อมูลของส่วนเบี่ยงเบนมาตรฐานที่สุ่มได้ในข้อ (2) ซึ่งจะได้เมตริกซ์ย่อย X_p ขนาด 3×3 สำหรับข้อมูลของขอบภาพในข้อ (3) จะดึงมาเฉพาะจุด p เท่านั้นซึ่งจะเก็บข้อมูลนี้ไว้ในตัวแปร D_p

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไปว่ากรรณิดงข้างสืบ อีกทั้งหวั่นเป็นให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(5) นำข้อมูลของส่วนเบี่ยงเบนมาตรฐานที่อยู่ในรูปเมตริกซ์ย่อย X_p จากข้อ (4) มาหาค่าความเป็นขอบภาพของแต่ละจุด โดยการแปลงข้อมูลในเมตริกซ์ย่อย X_p ไปเป็นเวกเตอร์ \vec{V}_p ดังที่แสดงไว้ในรูปที่ 6.16 และทำการป้อนเซตที่ใช้สอน $\{ \vec{V}_p, D_p \}$ เข้าสู่โครงข่าย

(6) คำนวณค่าของแต่ละโหนดในชั้นฮิดเดน สำหรับชั้นแรกมีอยู่ 20 โหนด

(7) คำนวณค่าของแต่ละโหนดในชั้นฮิดเดน ซึ่งในชั้นที่สองมีอยู่ 15 โหนด

(8) คำนวณโหนดเอาต์พุต และนำค่าเอาต์พุตที่ได้ไปผ่านค่าทรานซไฮลด์ ซึ่งมีค่าเท่ากับ 0.5 นั่นคือถ้าเอาต์พุตที่ผ่าน Activation Function มาแล้วมีค่าน้อยกว่า 0.5 แสดงว่าแพทเทิร์นที่ p ใดๆ ไม่ใช่ขอบ แต่ถ้าว่านเอาต์พุตมีค่ามากกว่า 0.5 แสดงว่าแพทเทิร์นที่ p เป็นขอบภาพ

(9) ปรับค่าถ่วงน้ำหนักที่เชื่อมต่อระหว่างชั้นทุกชั้น โดยใช้ค่าโมเมนตัม (α) และค่าอัตราการเรียนรู้ (η)

(10) คำนวณผลรวมกำลังสองของค่าความผิดพลาด

(11) ตรวจสอบว่าสอนหมดทุกแพทเทิร์นหรือยัง ถ้ายังก็ทำการสุ่มแพทเทิร์นต่อไป แต่ถ้าสอนครบทุกแพทเทิร์นแล้วก็จะทำการตรวจสอบต่อไปในข้อ (12)

(12) ตรวจสอบว่าค่าความผิดพลาดรวมของทุกแพทเทิร์น มีค่าน้อยกว่าค่าความผิดพลาดที่ยอมรับได้หรือยัง ถ้ายังก็ทำการสอนรอบต่อไป แต่ถ้าค่าความผิดพลาดอยู่ในโดเมนที่ยอมรับได้แล้วก็จะหยุดการสอนและเก็บค่าถ่วงน้ำหนักทั้งหมดเอาไว้ใช้งานต่อไป

ก่อนที่จะเริ่มต้นการสอนโครงข่ายพีชชีนิวรอลตามขั้นตอนต่างๆ ที่ได้กล่าวมาแล้ว จะต้องมีการกำหนดค่าถ่วงน้ำหนักไว้ก่อน โดยการสุ่มขึ้นมา และเซตของค่าถ่วงน้ำหนักที่เชื่อมโยงระหว่างชั้นทุกชั้นมีค่าอยู่ในช่วง $[0, 1]$

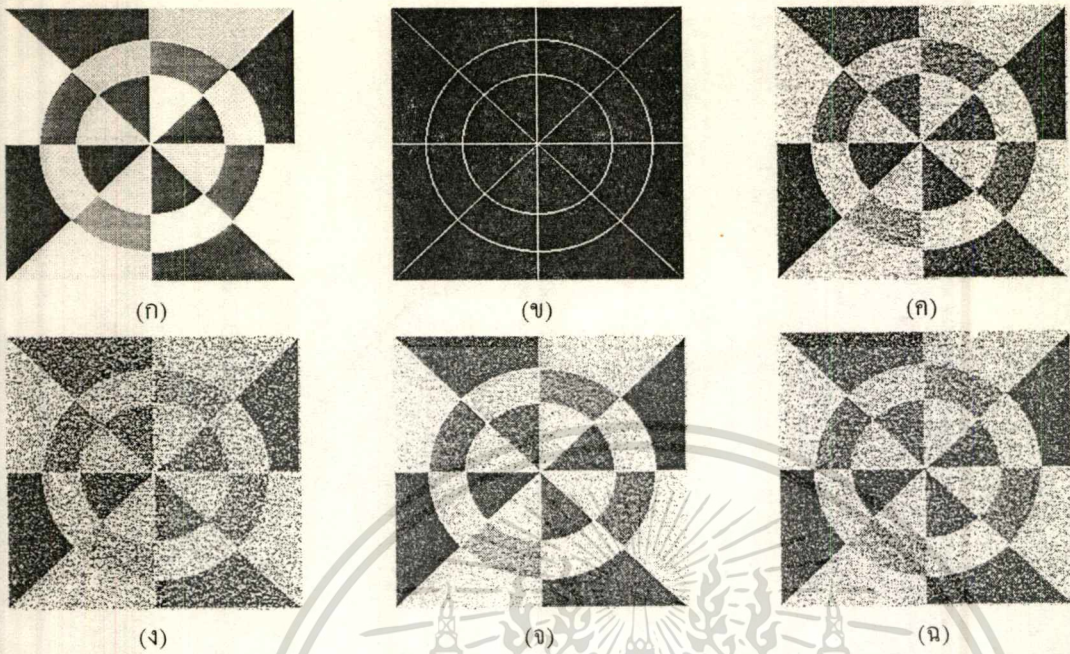
6.5 ผลการทดลองและการเปรียบเทียบ

การหาขอบภาพโดยวิธีต่างๆ ที่ได้เสนอไปแล้ว เป็นการพัฒนาวธีการหาขอบภาพโดยใช้ระบบฐานความรู้เกี่ยวกับขอบภาพมาเป็นเงื่อนไข เพื่อที่จะระบุว่าจุดภาพใดเป็นขอบหรือไม่เป็นขอบภาพ โดยเริ่มพัฒนาจาก การหาขอบภาพโดยการประมวลผลก่อน, การหาขอบภาพโดยพีชชีเซต, การหาขอบภาพโดยนิวรอลเน็ตเวิร์ค และการหาขอบภาพโดยพีชชีนิวรอลเน็ตเวิร์ค ตามลำดับ วิธีการหาขอบภาพเหล่านี้จะพัฒนาไปสู่ระบบการหาขอบภาพโดยอัตโนมัติคือ ผู้ใช้ไม่ต้องดำเนินการใดๆ ทั้งสิ้นในช่วงระหว่างการพัฒนาประมวลผล ผู้ใช้เพียงแต่ป้อนภาพอินพุตเข้าไปในระบบ วิธีการเหล่านี้จะดำเนินการเองทั้งหมดจนกระทั่งได้ผลลัพธ์เป็นขอบภาพที่สมบูรณ์และถูกต้องออกมา พร้อมทั้งจะนำไปใช้งานในขั้นตอนการวิเคราะห์ภาพในระดับสูงต่อไป

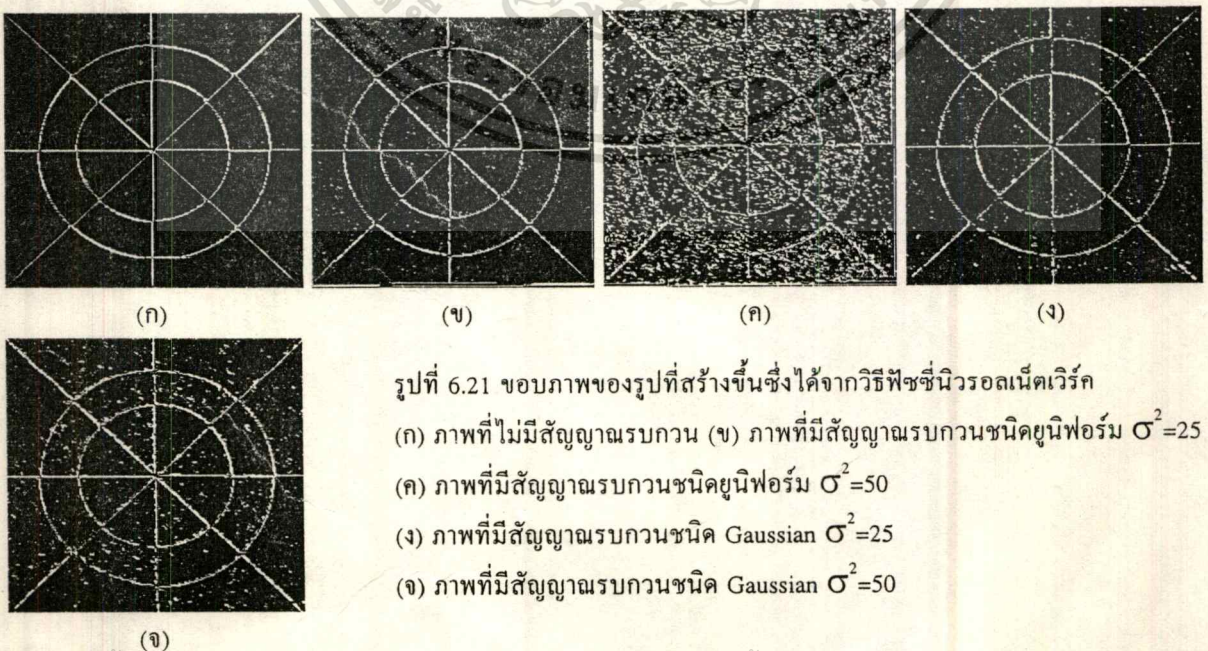
การเปรียบเทียบผลลัพธ์ของวิธีการที่เสนอ กับวิธีการหาขอบภาพโดยทั่วไปที่ได้กล่าวไปแล้วในบทที่ 1 จะมีการเปรียบเทียบเฉพาะสามวิธีหลังของวิธีที่ได้เสนอไป ส่วนวิธีการหาขอบภาพโดยการประมวลผลก่อนจะไม่นำมาเปรียบเทียบ เนื่องจากการหาขอบภาพโดยวิธีการนี้จะได้ขอบภาพในลักษณะคอนทราสต์

(Contour Edge) มากกว่าจะเป็นขอบภาพธรรมดาทั่วไป ซึ่งถ้านำไปใช้งานจริง ผู้ใช้ต้องทราบถึงช่วงของข้อมูลที่เป็นขอบภาพอย่างคร่าวๆ ก่อนจึงจะได้ขอบภาพที่ถูกต้อง ส่วนขอบภาพที่ได้จากสามวิธีหลังคือ การหาขอบภาพโดยพีชชีเซต, การหาขอบภาพโดยนิเวรอลเน็ตเวิร์ค และการหาขอบภาพโดยพีชชีนิเวรอลเน็ตเวิร์ค วิธีการเหล่านี้จะให้ผลลัพธ์ที่เป็นขอบภาพปกติที่เรามองเห็นได้โดยทั่วไป ในการเปรียบเทียบนอกจากจะใช้วิธีสังเกตจากของภาพที่ได้แล้ว เรายังวัดความถูกต้องของขอบภาพที่ได้ในแต่ละวิธี 2 ลักษณะด้วยกันคือ การวัดความถูกต้องของตำแหน่งขอบภาพ และการวัดเปอร์เซ็นต์ของขอบภาพที่หาได้โดยเทียบกับขอบภาพจริง ซึ่งวิธีการวัดเหล่านี้ได้กล่าวถึงไปแล้วในบทที่ 1 (ในหัวข้อ 1.3) และภาพที่นำมาทดสอบมีสองชนิดด้วยกันคือ ภาพที่สร้างขึ้นมาซึ่งภาพนี้จะมีต้นแบบของขอบภาพที่ถูกต้องแน่นอน และภาพอีกชนิดหนึ่งเป็นภาพจริงที่สแกนเข้ามา ซึ่งภาพทั้งสองชนิดนี้จะมีขนาด 256×256

เพื่อทดสอบความเชื่อถือได้ในผลลัพธ์ที่ได้จากวิธีต่างๆ ว่ามีการตอบสนองต่อสัญญาณรบกวนมากน้อยเพียงใด ดังนั้นภาพที่จะนำมาทดสอบจึงถูกเพิ่มสัญญาณรบกวนเข้าไป โดยสัญญาณรบกวนที่นำมาใช้ในที่นี้มีสองชนิดด้วยกันคือ สัญญาณรบกวนที่มีการแจกแจงแบบยูนิฟอร์มและ Gaussian สำหรับสัญญาณรบกวนที่มีการแจกแจงแบบ Gaussian จะมีค่าเฉลี่ยเท่ากับศูนย์ และสัญญาณรบกวนทั้งสองชนิดที่ถูกเพิ่มเข้าไปในภาพนี้จะมีค่าความแปรปรวนเท่ากับ 25 และ 50 สำหรับภาพที่สร้างขึ้น (รูปที่ 6.20) ซึ่งเป็นภาพที่มีขอบชัดเจนแบบขั้นบันได, สำหรับภาพโดยทั่วไปจะใช้ภาพแผนที่ (รูปที่ 6.33) ซึ่งเป็นภาพที่ได้มาจากสแกนเนอร์ โดยขอบภาพของภาพนี้จะเป็นแบบเพิ่มขึ้นหรือลดลงทีละน้อยในลักษณะหน้าจั่วและมีการเพิ่มสัญญาณรบกวนทั้งสองชนิดเข้าไปอีกด้วยค่าความแปรปรวน 20 และ 40 สำหรับผลการวัดความถูกต้องของขอบในแต่ละภาพที่ได้จากวิธีต่างๆ จะแสดงในตารางที่ 6.2 และ 6.3 ตามลำดับ ในการวัดความถูกต้องของขอบภาพโดยทั่วไปจะไม่มีขอบภาพต้นแบบที่ถูกต้องแน่นอน ดังนั้นในที่นี้จะใช้ขอบภาพที่ได้จากวิธีของ Robert มาเป็นต้นแบบสำหรับเปรียบเทียบ

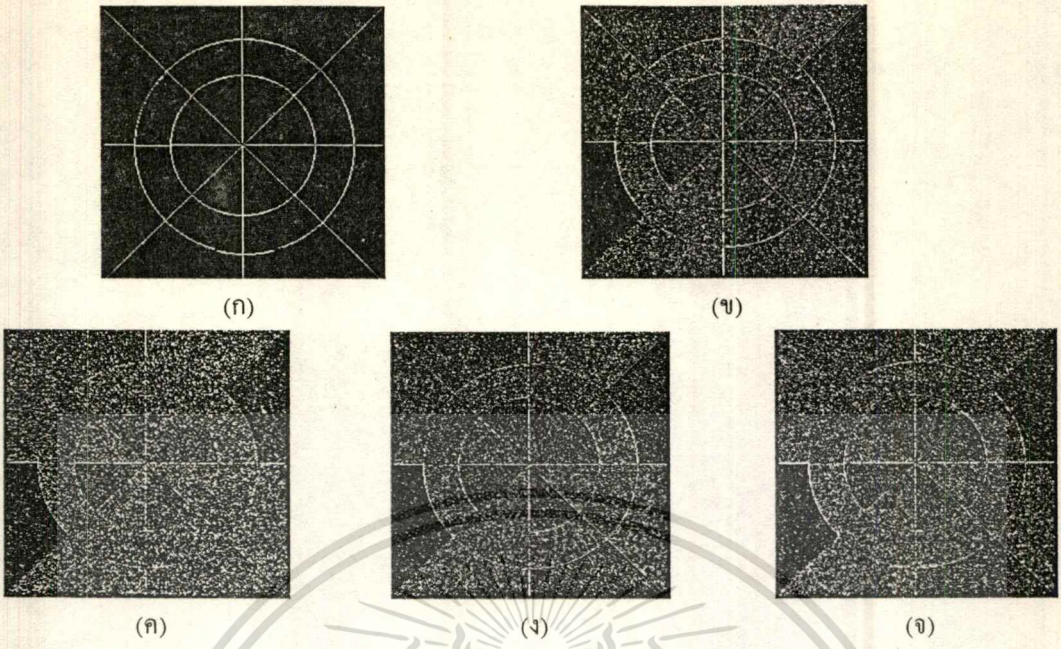


รูปที่ 6.20 แสดงภาพที่สร้างขึ้นเพื่อนำมาใช้ทดสอบวิธีการหาขอบภาพแต่ละวิธี
 (ก) ภาพที่ไม่มีสัญญาณรบกวน (ข) ขอบภาพต้นแบบ
 (ค) ภาพที่มีสัญญาณรบกวนที่มีการแจกแจงแบบยูนิฟอร์ม โดยความแปรปรวนเท่ากับ 25
 (ง) ภาพที่มีสัญญาณรบกวนที่มีการแจกแจงแบบยูนิฟอร์ม โดยความแปรปรวนเท่ากับ 50
 (ฉ) ภาพที่มีสัญญาณรบกวนที่มีการแจกแจงแบบ Gaussian โดยความแปรปรวนเท่ากับ 25
 (ฉ) ภาพที่มีสัญญาณรบกวนที่มีการแจกแจงแบบ Gaussian โดยความแปรปรวนเท่ากับ 50



รูปที่ 6.21 ขอบภาพของรูปที่สร้างขึ้นซึ่งได้จากวิธีพีชชีนิวโรลเน็ทเวิร์ค
 (ก) ภาพที่ไม่มีสัญญาณรบกวน (ข) ภาพที่มีสัญญาณรบกวนชนิดยูนิฟอร์ม $\sigma^2=25$
 (ค) ภาพที่มีสัญญาณรบกวนชนิดยูนิฟอร์ม $\sigma^2=50$
 (ง) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=25$
 (ฉ) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=50$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

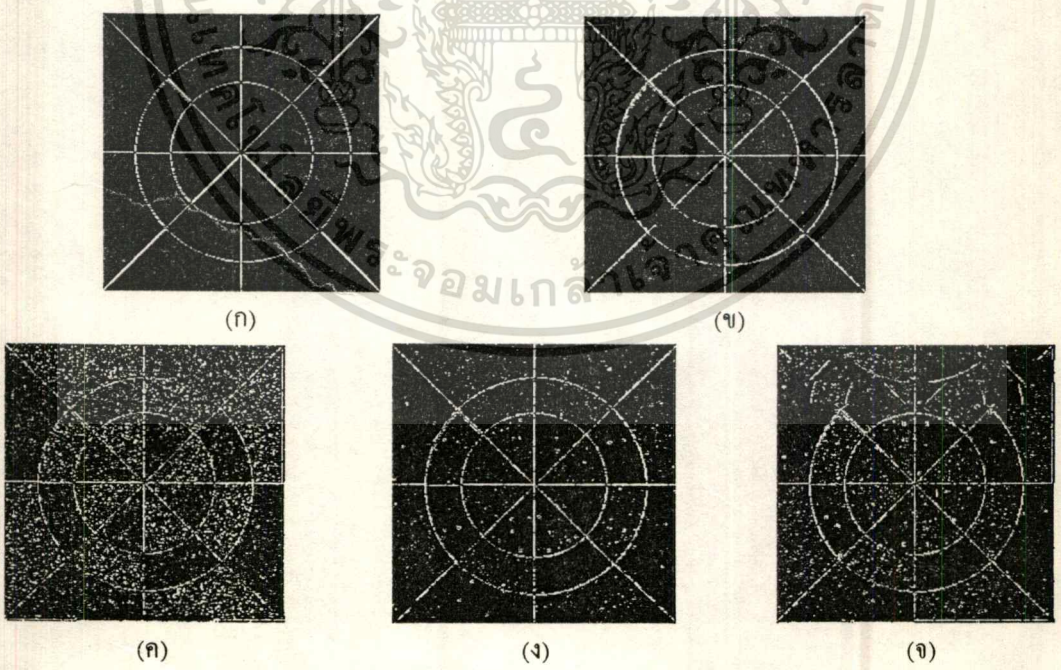


รูปที่ 6.22 ขอบภาพของรูปที่สร้างขึ้นซึ่งได้จากวิธีการหาขอบภาพโดยนิรอลเน็คเวิร์ค

(ก) ภาพที่ไม่มีสัญญาณรบกวน (ข) ภาพที่มีสัญญาณรบกวนชนิดยูนิฟอร์ม $\sigma^2=25$

(ค) ภาพที่มีสัญญาณรบกวนชนิดยูนิฟอร์ม $\sigma^2=50$ (ง) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=25$

(จ) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=50$

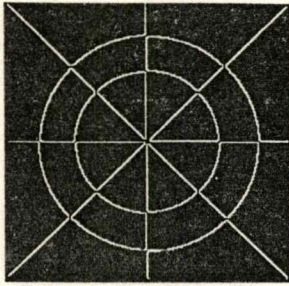


รูปที่ 6.23 ขอบภาพของรูปที่สร้างขึ้นซึ่งได้จากวิธีการหาขอบภาพโดยพีชชีเซต

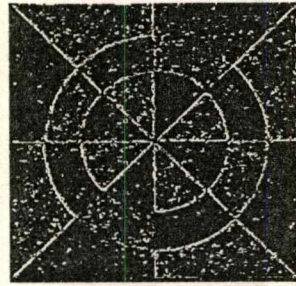
(ก) ภาพที่ไม่มีสัญญาณรบกวน (ข) ภาพที่มีสัญญาณรบกวนชนิดยูนิฟอร์ม $\sigma^2=25$

(ค) ภาพที่มีสัญญาณรบกวนชนิดยูนิฟอร์ม $\sigma^2=50$ (ง) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=25$

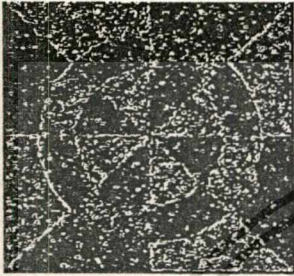
(จ) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=50$



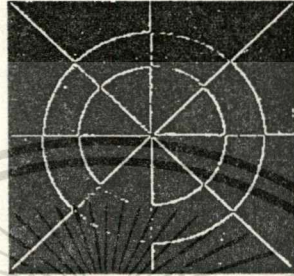
(ก)



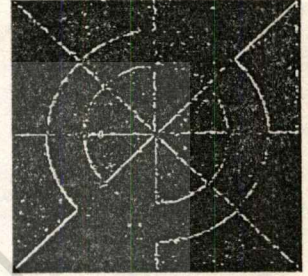
(ข)



(ค)



(ง)



(จ)

รูปที่ 6.24 ขอบภาพของรูปที่สร้างขึ้นซึ่งได้จากวิธีของ Robert

(ก) ขอบของภาพที่ไม่มีสัญญาณรบกวน

(ข) ภาพที่มีสัญญาณรบกวนชนิดยูนิฟอร์ม $\sigma^2=25$

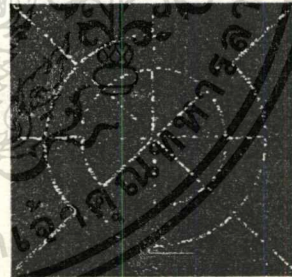
(ค) ภาพที่มีสัญญาณรบกวนชนิดยูนิฟอร์ม $\sigma^2=50$

(ง) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=25$

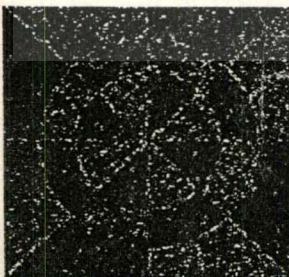
(จ) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=50$



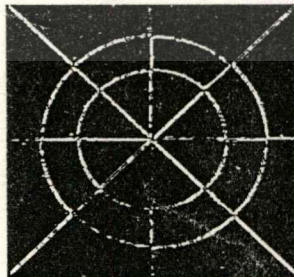
(ก)



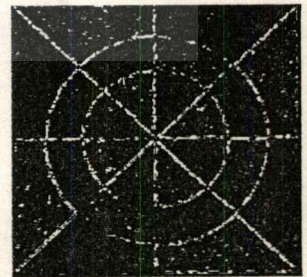
(ข)



(ค)



(ง)



(จ)

รูปที่ 6.25 ขอบภาพของรูปที่สร้างขึ้นซึ่งได้จากวิธีของ Prewitt

(ก) ขอบของภาพที่ไม่มีสัญญาณรบกวน

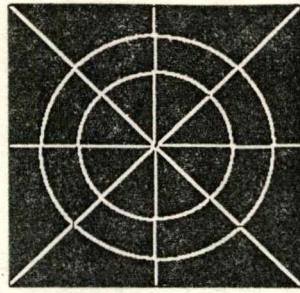
(ข) ภาพที่มีสัญญาณรบกวนชนิดยูนิฟอร์ม $\sigma^2=25$

(ค) ภาพที่มีสัญญาณรบกวนชนิดยูนิฟอร์ม $\sigma^2=50$

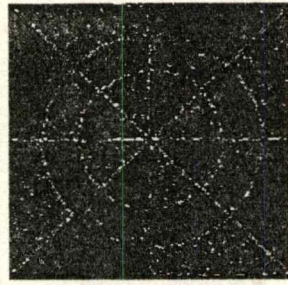
(ง) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=25$

(จ) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=50$

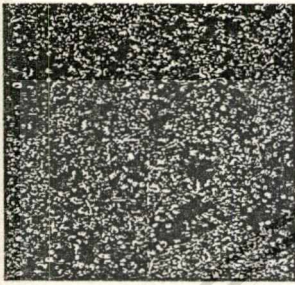
ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



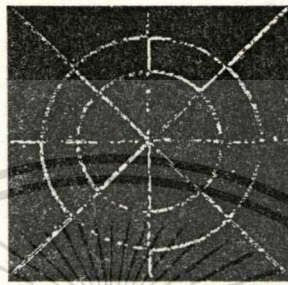
(ก)



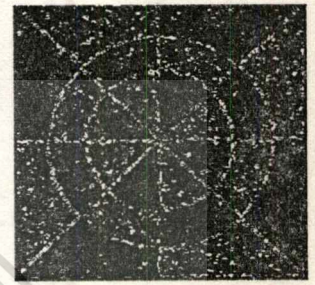
(ข)



(ค)



(ง)



(จ)

รูปที่ 6.26 ขอบภาพของรูปที่สร้างขึ้นซึ่งได้จากวิธีของ Sobel

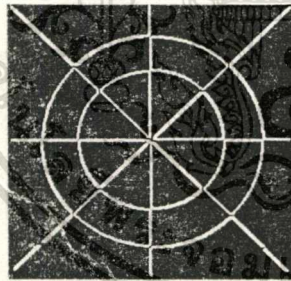
(ก) ขอบของภาพที่ไม่มีสัญญาณรบกวน

(ข) ภาพที่มีสัญญาณรบกวนชนิดยูนีฟอร์ม $\sigma^2=25$

(ค) ภาพที่มีสัญญาณรบกวนชนิดยูนีฟอร์ม $\sigma^2=50$

(ง) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=25$

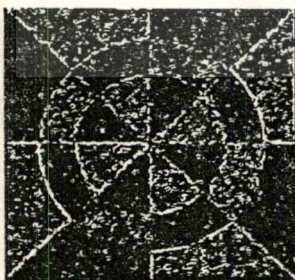
(จ) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=50$



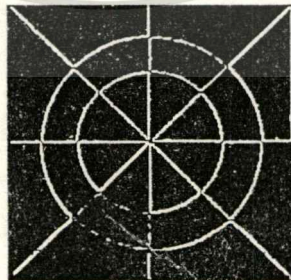
(ก)



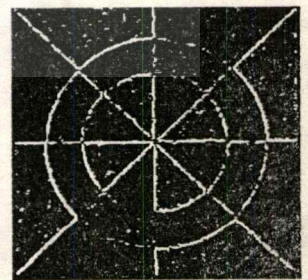
(ข)



(ค)



(ง)



(จ)

รูปที่ 6.27 ขอบภาพของรูปที่สร้างขึ้นซึ่งได้จากวิธีของ Frei-Chen

(ก) ขอบของภาพที่ไม่มีสัญญาณรบกวน

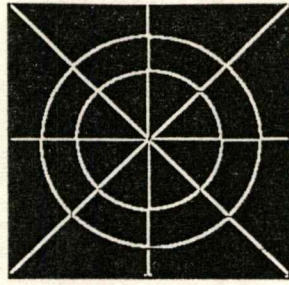
(ข) ภาพที่มีสัญญาณรบกวนชนิดยูนีฟอร์ม $\sigma^2=25$

(ค) ภาพที่มีสัญญาณรบกวนชนิดยูนีฟอร์ม $\sigma^2=50$

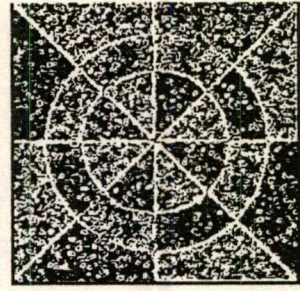
(ง) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=25$

(จ) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=50$

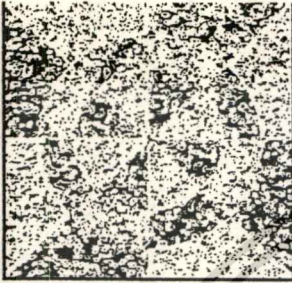
เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีกรนำไปใช้



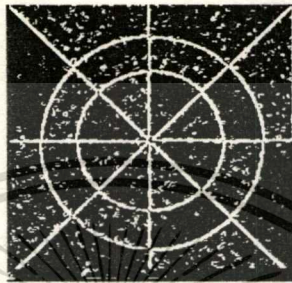
(ก)



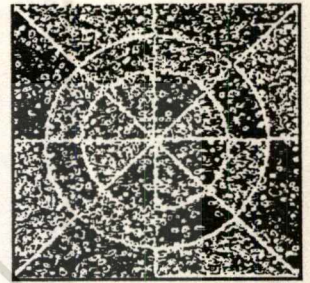
(ข)



(ค)



(ง)



(จ)

รูปที่ 6.28 ขอบภาพของรูปที่สร้างขึ้นซึ่งได้จากวิธีของ Kirsch

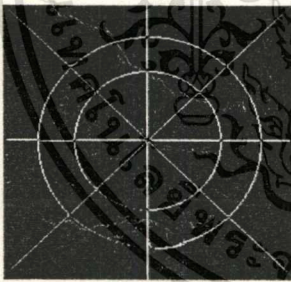
(ก) ขอบของภาพที่ไม่มีสัญญาณรบกวน

(ข) ภาพที่มีสัญญาณรบกวนชนิดยูนีฟอร์ม $\sigma^2=25$

(ค) ภาพที่มีสัญญาณรบกวนชนิดยูนีฟอร์ม $\sigma^2=50$

(ง) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=25$

(จ) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=50$



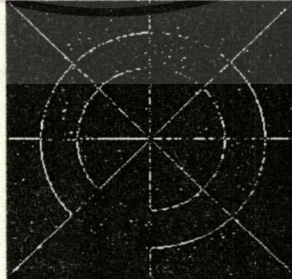
(ก)



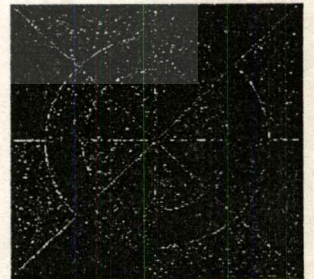
(ข)



(ค)



(ง)



(จ)

รูปที่ 6.29 ขอบภาพของรูปที่สร้างขึ้นซึ่งได้จากวิธีของ Laplacian

(ก) ขอบของภาพที่ไม่มีสัญญาณรบกวน

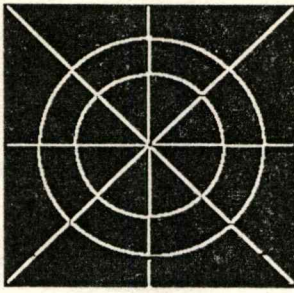
(ข) ภาพที่มีสัญญาณรบกวนชนิดยูนีฟอร์ม $\sigma^2=25$

(ค) ภาพที่มีสัญญาณรบกวนชนิดยูนีฟอร์ม $\sigma^2=50$

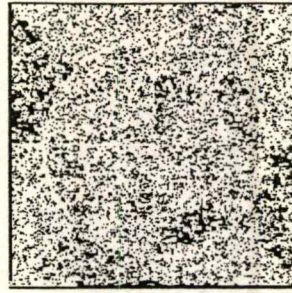
(ง) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=25$

(จ) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=50$

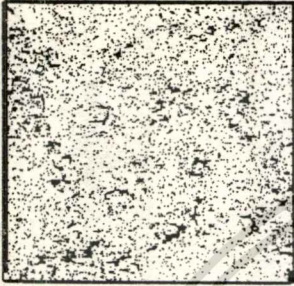
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับวารสารงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



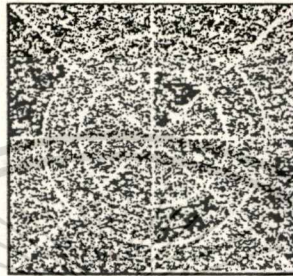
(ก)



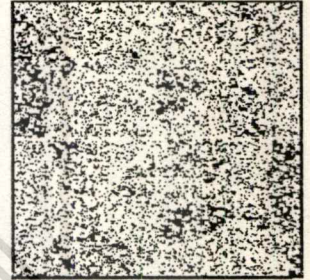
(ข)



(ค)



(ง)



(จ)

รูปที่ 6.30 ขอบภาพของรูปที่สร้างขึ้นซึ่งได้จากวิธีของ Marr และ Hildreth

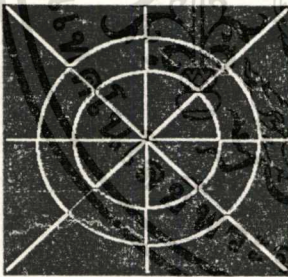
(ก) ขอบของภาพที่ไม่มีสัญญาณรบกวน

(ข) ภาพที่มีสัญญาณรบกวนชนิดยูนิฟอร์ม $\sigma^2=25$

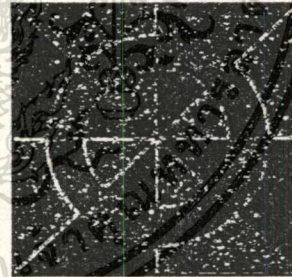
(ค) ภาพที่มีสัญญาณรบกวนชนิดยูนิฟอร์ม $\sigma^2=50$

(ง) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=25$

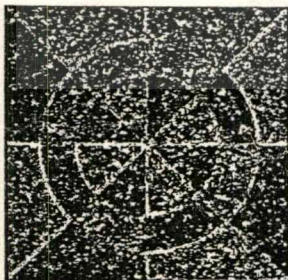
(จ) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=50$



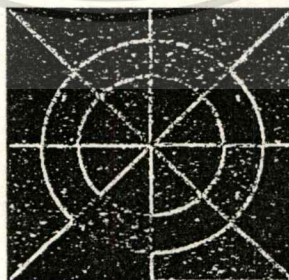
(ก)



(ข)



(ค)



(ง)



(จ)

รูปที่ 6.31 ขอบภาพของรูปที่สร้างขึ้นซึ่งได้จากวิธีของ Canny

(ก) ขอบของภาพที่ไม่มีสัญญาณรบกวน

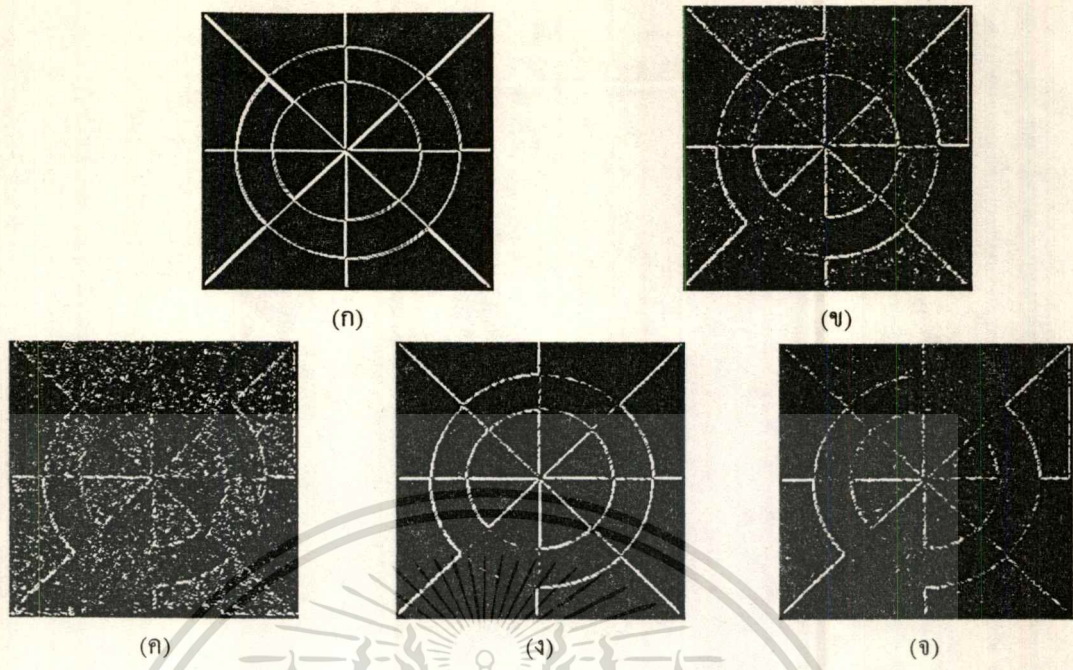
(ข) ภาพที่มีสัญญาณรบกวนชนิดยูนิฟอร์ม $\sigma^2=25$

(ค) ภาพที่มีสัญญาณรบกวนชนิดยูนิฟอร์ม $\sigma^2=50$

(ง) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=25$

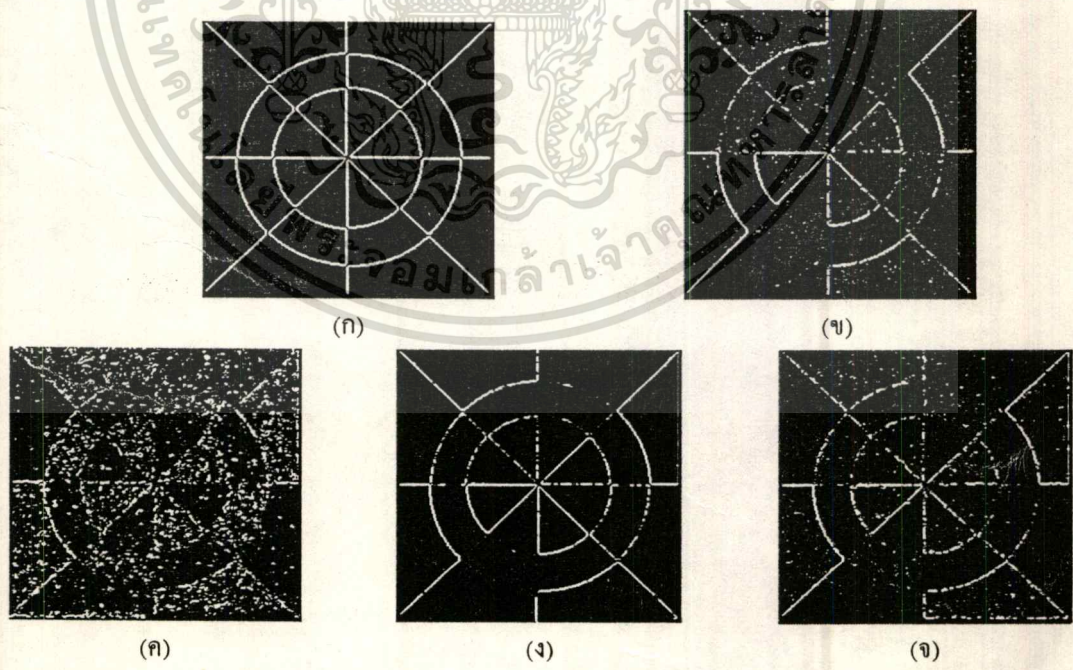
(จ) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=50$

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ของภาควิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.32 ขอบภาพของรูปที่สร้างขึ้นซึ่งได้จากวิธีของ Gabor

- (ก) ขอบของภาพที่ไม่มีสัญญาณรบกวน
- (ข) ภาพที่มีสัญญาณรบกวนชนิดยูนิฟอร์ม $\sigma^2=25$
- (ค) ภาพที่มีสัญญาณรบกวนชนิดยูนิฟอร์ม $\sigma^2=50$
- (ง) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=25$
- (ฉ) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=50$
- (จ) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=50$



รูปที่ 6.33 ขอบภาพของรูปที่สร้างขึ้นซึ่งได้จากวิธีของ Relaxation

- (ก) ขอบของภาพที่ไม่มีสัญญาณรบกวน
- (ข) ภาพที่มีสัญญาณรบกวนชนิดยูนิฟอร์ม $\sigma^2=25$
- (ค) ภาพที่มีสัญญาณรบกวนชนิดยูนิฟอร์ม $\sigma^2=50$
- (ง) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=25$
- (ฉ) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=50$
- (จ) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=50$

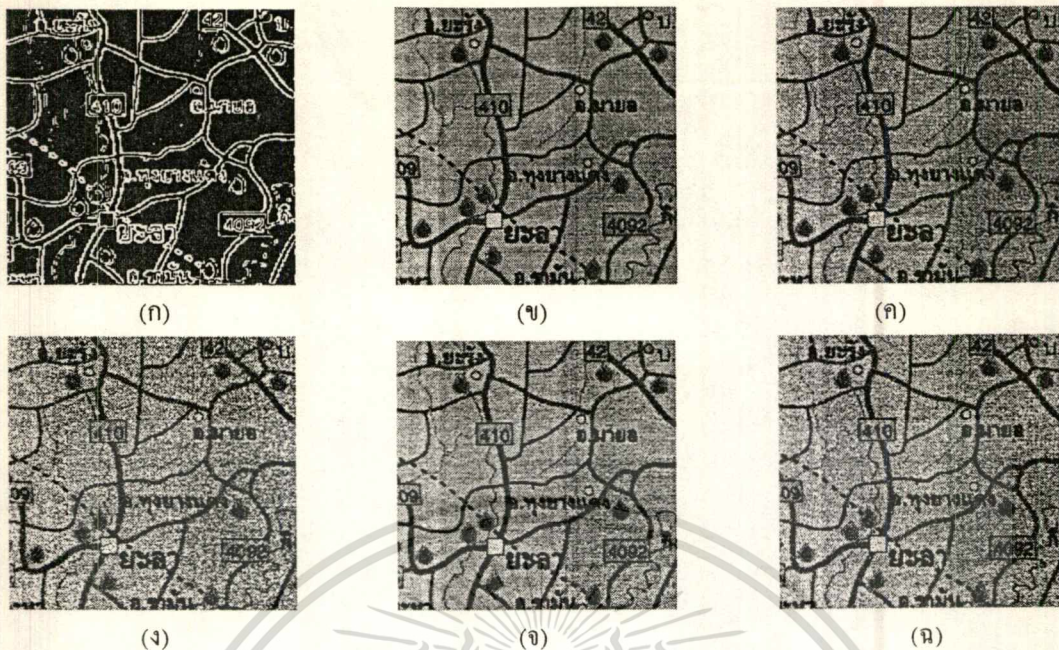
เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.2 เปรียบเทียบประสิทธิภาพของตัวดำเนินการหาขอบภาพของแต่ละวิธีสำหรับรูปที่สร้างขึ้น

วิธีการหาขอบภาพ	รูปที่	ค่าตัวแปร	ค่าเทรชโฮลด์	P_e	P^*
พีชชีนิวรอลเน็ตเวิร์ค	6.21ก	-	-	1.294671	0.438264
	6.21ข	-	-	1.688610	0.126937
	6.21ค	-	-	4.364159	0.049585
	6.21ง	-	-	1.840125	0.128417
	6.21จ	-	-	2.357367	0.108845
นิวรอลเน็ตเวิร์ค	6.22ก	-	-	0.879278	0.844589
	6.22ข	-	-	5.605920	0.140683
	6.22ค	-	-	10.022664	0.067201
	6.22ง	-	-	6.978261	0.106987
	6.22จ	-	-	8.007863	0.089305
พีชชีเซต	6.23ก	-	-	0.927845	0.785146
	6.23ข	-	-	0.944033	0.800590
	6.23ค	-	-	5.535153	0.131034
	6.23ง	-	-	1.419056	0.526080
	6.23จ	-	-	2.247456	0.325999
Robert	6.24ก	-	40	1.314061	0.167564
	6.24ข	-	40	2.998512	0.105828
	6.24ค	-	60	4.660962	0.055491
	6.24ง	-	60	1.324699	0.179834
	6.24จ	-	60	3.409806	0.096730
Prewitt	6.25ก	-	5	2.832562	0.262742
	6.25ข	-	200	0.5579093	0.152182
	6.25ค	-	200	1.933395	0.082793
	6.25ง	-	100	1.965772	0.264010
	6.25จ	-	150	1.691952	0.206408
Sobel	6.26ก	-	10	2.815911	0.124026
	6.26ข	-	200	0.826549	0.080492
	6.26ค	-	150	8.148936	0.041108
	6.26ง	-	150	1.170675	0.117757
	6.26จ	-	170	1.999075	0.083083
Frei-Chen	6.27ก	-	30	2.374653	0.287111
	6.27ข	-	50	2.051804	0.234003

Frei-Chen	6.27ก	-	60	4.380204	0.086283
	6.27ง	-	35	2.336263	0.275001
	6.27จ	-	50	2.264107	0.218599
Kirsch	6.28ก	-	50	2.808048	0.271789
	6.28ข	-	250	13.329787	0.058823
	6.28ค	-	250	21.678538	0.036663
	6.28ง	-	250	4.821462	0.160501
	6.28จ	-	250	14.99889	0.056325
Laplacian	6.29ก	-	50	0.895930	0.262731
	6.29ข	-	70	1.205360	0.111299
	6.29ค	-	100	1.795097	0.055928
	6.29ง	-	60	0.763182	0.171151
	6.29จ	-	70	0.280759	0.111249
Marr และ Hildreth	6.30ก	$\sigma=1.06$	-	2.641938	0.283794
	6.30ข	$\sigma=1.06$	-	23.410731	0.034602
	6.30ค	$\sigma=1.06$	-	25.471785	0.032130
	6.30ง	$\sigma=1.06$	-	19.913969	0.041443
	6.30จ	$\sigma=1.06$	-	23.467160	0.034381
Canny	6.31ก	$\sigma=1.062$	5	2.406105	0.292778
	6.31ข	$\sigma=1.062$	10	2.259482	0.134915
	6.31ค	$\sigma=1.062$	15	7.852451	0.066866
	6.31ง	$\sigma=1.062$	5	3.312673	0.182359
	6.31จ	$\sigma=1.062$	10	2.487049	0.130010
Gabor	6.32ก	$\sigma=0.9, \omega=1.11$	0.1	2.230342	0.250526
	6.32ข	$\sigma=0.9, \omega=1.11$	1	2.553654	0.084787
	6.32ค	$\sigma=0.9, \omega=1.11$	2	3.305735	0.57667
	6.32ง	$\sigma=0.9, \omega=1.11$	1	1.644311	0.118724
	6.32จ	$\sigma=0.9, \omega=1.11$	1	1.106383	0.109546
Relaxation	6.33ก	$d=0.1, q=0.15$	-	1.813154	0.296854
	6.33ข	$d=0.1, q=0.15$	-	1.612396	0.200527
	6.33ค	$d=0.1, q=0.15$	-	4.014801	0.067988
	6.33ง	$d=0.1, q=0.15$	-	1.448659	0.284172
	6.33จ	$d=0.1, q=0.15$	-	1.675301	0.186373

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.34 ภาพแผนที่
 (ก) ขอบภาพของวิธี Robert ที่ใช้เป็นต้นแบบในการเปรียบเทียบกับวิธีอื่น
 (ข) ภาพที่ไม่มีสัญญาณรบกวน (ค) ภาพที่มีสัญญาณรบกวนชนิดยูนีฟอร์มปนอยู่โดย $\sigma^2=15$
 (ง) ภาพที่มีสัญญาณรบกวนชนิดยูนีฟอร์มปนอยู่โดย $\sigma^2=30$
 (จ) ภาพที่มีสัญญาณรบกวนชนิด Gaussian ปนอยู่โดย $\sigma^2=15$
 (ฉ) ภาพที่มีสัญญาณรบกวนชนิด Gaussian ปนอยู่โดย $\sigma^2=30$



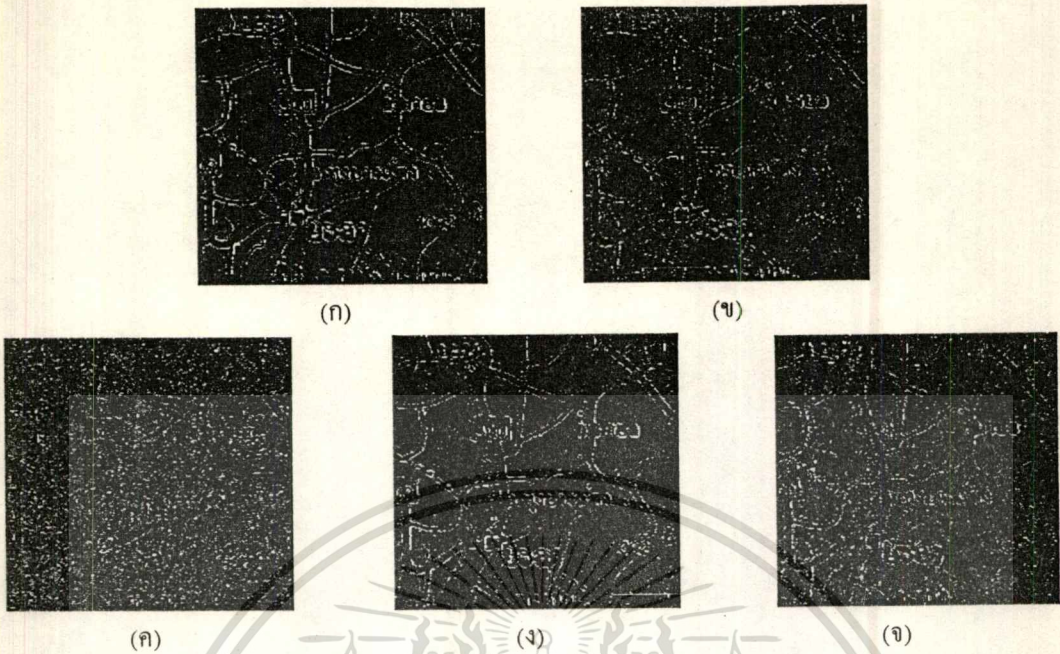
(ก) (ข)



(ค) (ง) (จ)

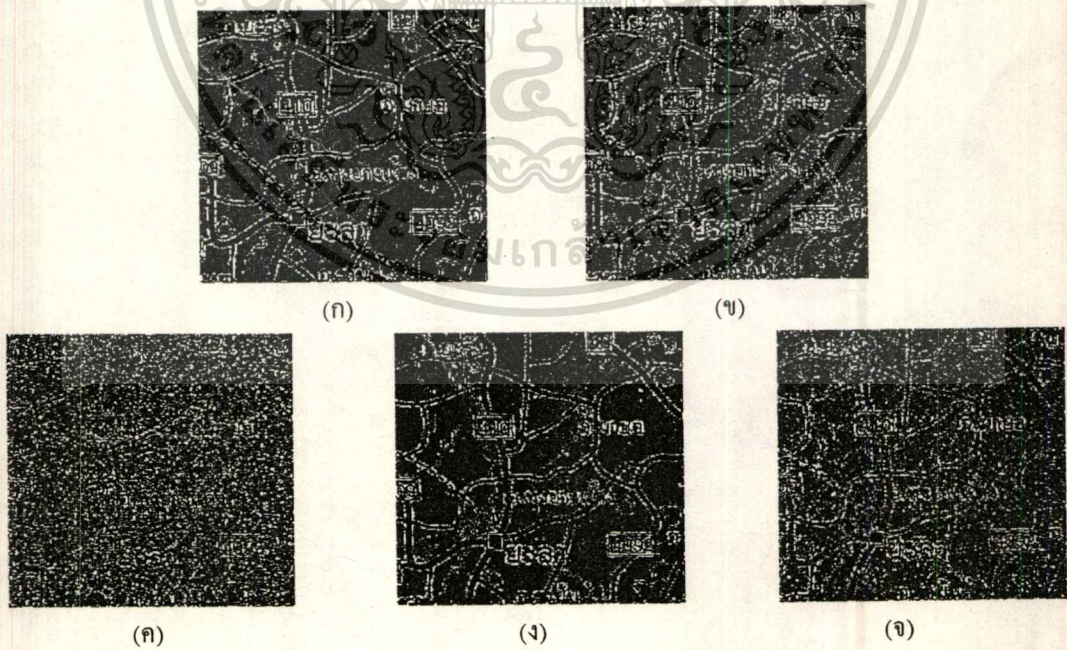
รูปที่ 6.35 ขอบของภาพแผนที่ที่ได้จากวิธีฟิชเชอร์เน็ท

(ก) ภาพที่ไม่มีสัญญาณรบกวน (ข) ภาพที่มีสัญญาณรบกวนชนิดยูนีฟอร์ม $\sigma^2=15$
 (ค) ภาพที่มีสัญญาณรบกวนชนิดยูนีฟอร์ม $\sigma^2=30$ (ง) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=15$
 (จ) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=30$



รูปที่ 6.36 ขอบของภาพแผนที่ที่ได้จากวิธีนิเวรอลเน็ตเวิร์ค

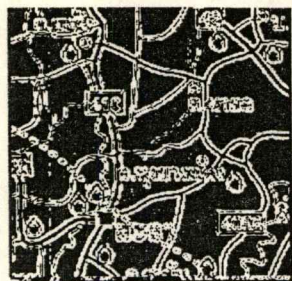
- (ก) ภาพที่ไม่มีสัญญาณรบกวน
- (ข) ภาพที่มีสัญญาณรบกวนชนิดยูนีฟอร์ม $\sigma^2=15$
- (ค) ภาพที่มีสัญญาณรบกวนชนิดยูนีฟอร์ม $\sigma^2=30$
- (ง) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=15$
- (จ) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=30$



รูปที่ 6.37 ขอบของภาพแผนที่ที่ได้จากวิธีพีชชีเซต

- (ก) ภาพที่ไม่มีสัญญาณรบกวน
- (ข) ภาพที่มีสัญญาณรบกวนชนิดยูนีฟอร์ม $\sigma^2=15$
- (ค) ภาพที่มีสัญญาณรบกวนชนิดยูนีฟอร์ม $\sigma^2=30$
- (ง) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=15$ คำ
- (จ) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=30$ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

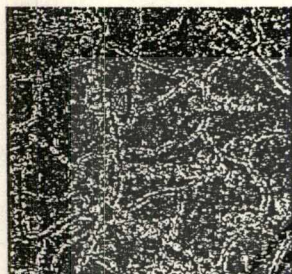
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับอาจารย์ใช้ภายในห้องเรียนเท่านั้น ไม่สามารถเผยแพร่หรือใช้เพื่อการศึกษารายอื่นได้



(ก)



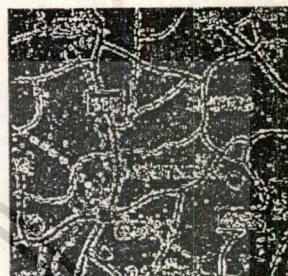
(ข)



(ค)



(ง)



(จ)

รูปที่ 6.38 ขอบของภาพแผนที่ซึ่งได้จากวิธีของ Prewitt

(ก) ภาพที่ไม่มีสัญญาณรบกวน

(ข) ภาพที่มีสัญญาณรบกวนชนิดยูนิฟอร์ม $\sigma^2=15$

(ค) ภาพที่มีสัญญาณรบกวนชนิดยูนิฟอร์ม $\sigma^2=30$

(ง) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=15$

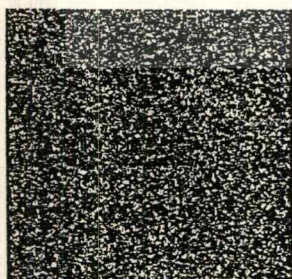
(จ) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=30$



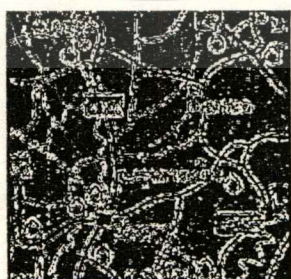
(ก)



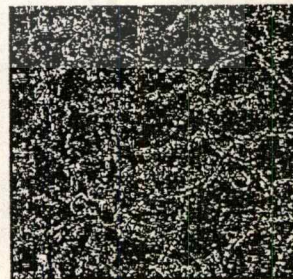
(ข)



(ค)



(ง)



(จ)

รูปที่ 6.39 ขอบของภาพแผนที่ซึ่งได้จากวิธีของ Sobel

(ก) ภาพที่ไม่มีสัญญาณรบกวน

(ข) ภาพที่มีสัญญาณรบกวนชนิดยูนิฟอร์ม $\sigma^2=15$

(ค) ภาพที่มีสัญญาณรบกวนชนิดยูนิฟอร์ม $\sigma^2=30$ (ง) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=15$

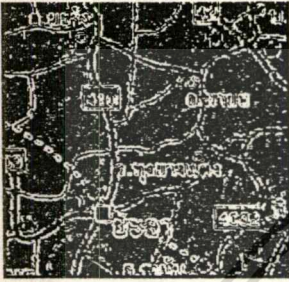
(จ) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=30$ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก)



(ข)



(ค)



(ง)



(จ)

รูปที่ 6.40 ขอบของภาพแผนที่ซึ่งได้จากวิธีของ Frei-Chen

(ก) ภาพที่ไม่มีสัญญาณรบกวน

(ข) ภาพที่มีสัญญาณรบกวนชนิดยูนีฟอร์ม $\sigma^2=15$

(ค) ภาพที่มีสัญญาณรบกวนชนิดยูนีฟอร์ม $\sigma^2=30$

(ง) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=15$

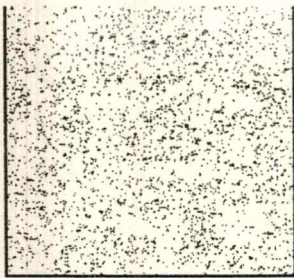
(จ) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=30$



(ก)



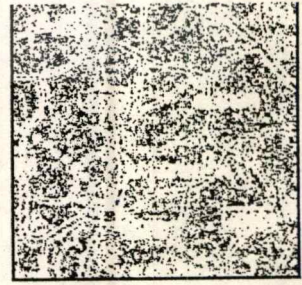
(ข)



(ค)



(ง)



(จ)

รูปที่ 6.41 ขอบของภาพแผนที่ซึ่งได้จากวิธีของ Kirsch

(ก) ภาพที่ไม่มีสัญญาณรบกวน

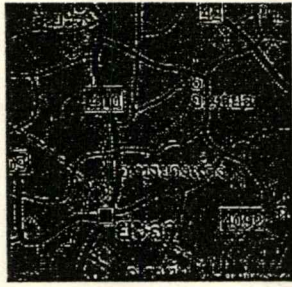
(ข) ภาพที่มีสัญญาณรบกวนชนิดยูนีฟอร์ม $\sigma^2=15$

(ค) ภาพที่มีสัญญาณรบกวนชนิดยูนีฟอร์ม $\sigma^2=30$

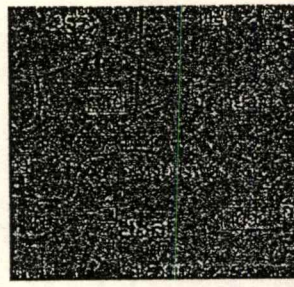
(ง) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=15$

(จ) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=30$

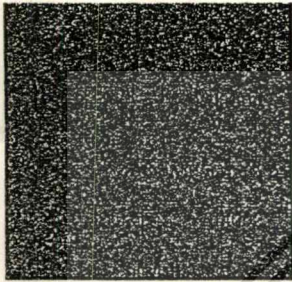
เอกสารนี้เป็นเอกสารทศงานไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



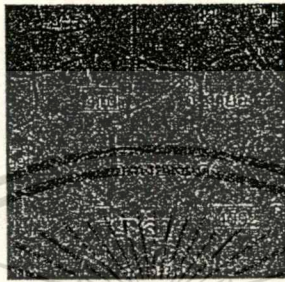
(ก)



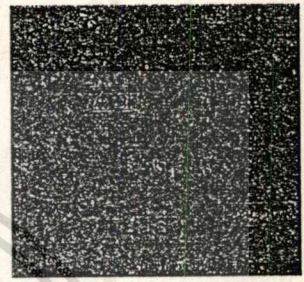
(ข)



(ค)



(ง)



(จ)

รูปที่ 6.42 ขอบของภาพแผนที่ซึ่งได้จากวิธีของ Laplacian

(ก) ภาพที่ไม่มีสัญญาณรบกวน

(ข) ภาพที่มีสัญญาณรบกวนชนิดยูนิฟอรม $\sigma^2=15$

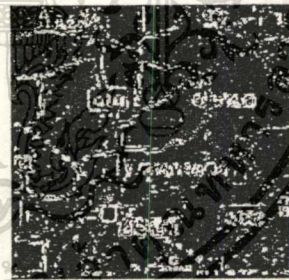
(ค) ภาพที่มีสัญญาณรบกวนชนิดยูนิฟอรม $\sigma^2=30$

(ง) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=15$

(จ) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=30$



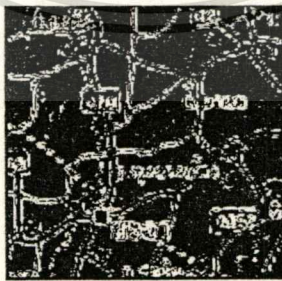
(ก)



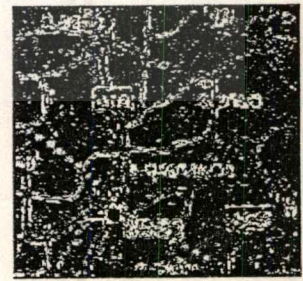
(ข)



(ค)



(ง)



(จ)

รูปที่ 6.43 ขอบของภาพแผนที่ซึ่งได้จากวิธีของ Marr และ Hildreth

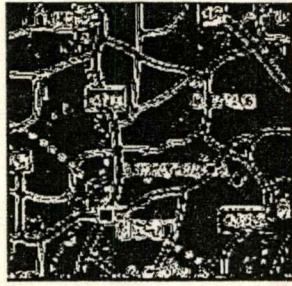
(ก) ภาพที่ไม่มีสัญญาณรบกวน

(ข) ภาพที่มีสัญญาณรบกวนชนิดยูนิฟอรม $\sigma^2=15$

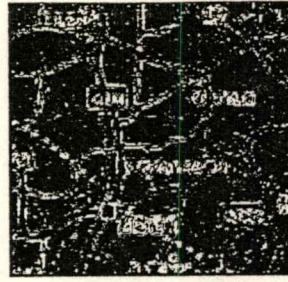
(ค) ภาพที่มีสัญญาณรบกวนชนิดยูนิฟอรม $\sigma^2=30$

(ง) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=15$

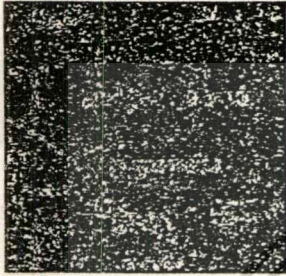
เอกสาร(จ) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=30$ การศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



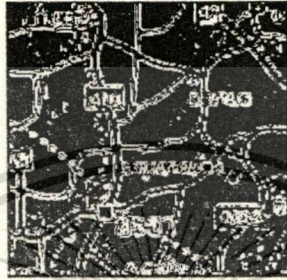
(ก)



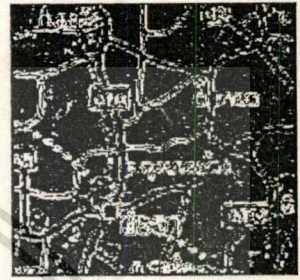
(ข)



(ค)



(ง)



(จ)

รูปที่ 6.44 ขอบของภาพแผนที่ซึ่งได้จากวิธีของ Canny

(ก) ภาพที่ไม่มีสัญญาณรบกวน

(ข) ภาพที่มีสัญญาณรบกวนชนิดยูนิฟอร์ม $\sigma^2=15$

(ค) ภาพที่มีสัญญาณรบกวนชนิดยูนิฟอร์ม $\sigma^2=30$

(ง) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=15$

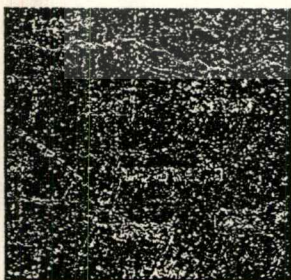
(จ) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=30$



(ก)



(ข)



(ค)



(ง)



(จ)

รูปที่ 6.45 ขอบของภาพแผนที่ซึ่งได้จากวิธีของ Gabor

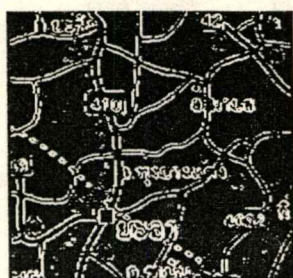
(ก) ภาพที่ไม่มีสัญญาณรบกวน

(ข) ภาพที่มีสัญญาณรบกวนชนิดยูนิฟอร์ม $\sigma^2=15$

(ค) ภาพที่มีสัญญาณรบกวนชนิดยูนิฟอร์ม $\sigma^2=30$

(ง) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=15$

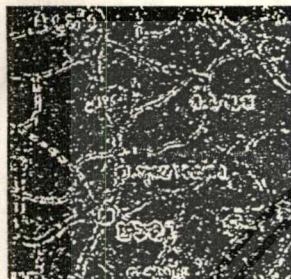
(จ) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=30$



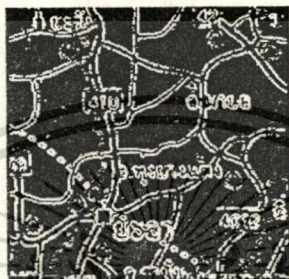
(ก)



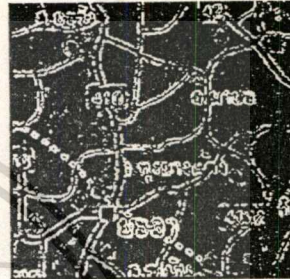
(ข)



(ค)



(ง)



(จ)

รูปที่ 6.46 ขอบของภาพแผนที่ซึ่งได้จากวิธี Relaxation

- (ก) ภาพที่ไม่มีสัญญาณรบกวน (ข) ภาพที่มีสัญญาณรบกวนชนิดยูนิฟอร์ม $\sigma^2=15$
 (ค) ภาพที่มีสัญญาณรบกวนชนิดยูนิฟอร์ม $\sigma^2=30$ (ง) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=15$
 (จ) ภาพที่มีสัญญาณรบกวนชนิด Gaussian $\sigma^2=30$

ตารางที่ 6.3 เปรียบเทียบประสิทธิภาพของตัวดำเนินการหาขอบภาพของแต่ละวิธีสำหรับรูปแผนที่

วิธีการหาขอบภาพ	รูปที่	ค่าตัวแปร	ค่าเทรชโฮลด์	P_e	P
พีชชีนิวรอลเน็ตเวิร์ค	6.35ก	-	-	0.960179	0.709080
	6.35ข	-	-	1.309097	0.498903
	6.35ค	-	-	2.141867	0.264428
	6.35ง	-	-	1.149729	0.597912
	6.35จ	-	-	1.549036	0.404214
นิวรอลเน็ตเวิร์ค	6.36ก	-	-	0.262729	0.134845
	6.36ข	-	-	0.197132	0.093433
	6.36ค	-	-	0.381704	0.081344
	6.36ง	-	-	0.227172	0.116770
	6.36จ	-	-	0.227257	0.093234
พีชชีเซต	6.37ก	-	-	0.540733	0.381987
	6.37ข	-	-	0.647064	0.336266
	6.37ค	-	-	1.366344	0.206574
	6.37ง	-	-	0.542515	0.372286
	6.37จ	-	-	0.818822	0.277752
Prewitt	6.38ก	-	80	1.514596	0.406153
	6.38ข	-	120	1.115411	0.401097
	6.38ค	-	140	1.364477	0.296662
	6.38ง	-	110	1.227936	0.399586
	6.38จ	-	140	0.992617	0.370334
Sobel	6.39ก	-	100	1.333503	0.480082
	6.39ข	-	120	1.212576	0.417945
	6.39ค	-	150	1.756195	0.234648
	6.39ง	-	120	1.812067	0.448920
	6.39จ	-	100	1.359725	0.334146
Frei-Chen	6.40ก	-	30	1.572811	0.407252
	6.40ข	-	40	1.359470	0.403434
	6.40ค	-	60	1.032671	0.371766
	6.40ง	-	40	1.335115	0.401641
	6.40จ	-	50	1.155720	0.387768
Kirsch	6.41ก	-	254	2.162424	0.367672
	6.41ข	-	254	3.616344	0.243741

Kirsch	6.41ก	-	254	5.025034	0.191354
	6.41ง	-	254	2.915818	0.290164
	6.41จ	-	254	4.264936	0.214613
Laplacian	6.42ก	-	40	0.478276	0.161662
	6.42ข	-	50	1.448489	0.189762
	6.42ค	-	70	1.759080	0.178834
	6.42ง	-	50	1.047522	0.214034
	6.42จ	-	60	1.435675	0.184540
Marr และ Hildreth	6.43ก	$\sigma=1.061$	-	1.133232	0.352181
	6.43ข	$\sigma=1.061$	-	0.679396	0.257130
	6.43ค	$\sigma=1.061$	-	0.892736	0.260610
	6.43ง	$\sigma=1.061$	-	0.871775	0.321624
	6.43จ	$\sigma=1.061$	-	0.938730	0.329516
Canny	6.44ก	$\sigma=1.062$	5	0.978785	0.535557
	6.44ข	$\sigma=1.062$	6	0.736762	0.363545
	6.44ค	$\sigma=1.062$	7	0.947980	0.283662
	6.44ง	$\sigma=1.062$	5	0.926426	0.486593
	6.44จ	$\sigma=1.062$	7	0.587152	0.295486
Gabor	6.45ก	$\sigma=0.5, \omega=0.5$	1	0.910811	0.579346
	6.45ข	$\sigma=0.5, \omega=0.5$	2	0.544807	0.326291
	6.45ค	$\sigma=0.5, \omega=0.5$	2	1.358028	0.306631
	6.45ง	$\sigma=0.5, \omega=0.5$	1.2	1.071623	0.542525
	6.45จ	$\sigma=0.5, \omega=0.5$	2	0.784114	0.363630
Relaxation	6.46ก	$d=0.15, q=0.05$	-	0.810506	0.560506
	6.46ข	$d=0.15, q=0.05$	-	0.788442	0.533252
	6.46ค	$d=0.15, q=0.05$	-	0.951460	0.465887
	6.46ง	$d=0.15, q=0.05$	-	0.938221	0.626528
	6.46จ	$d=0.15, q=0.05$	-	0.914291	0.561270

สรุป

จากตารางที่ 6.2-6.3 และรูปที่ 6.21-6.46 สามารถสรุปได้ดังนี้

- ประสิทธิภาพโดยรวมของวิธีการหาขอบภาพที่ได้เสนอไปแล้วทั้ง 3 วิธี การหาขอบภาพโดยใช้ฟิชชันนิวรอลเน็ตเวิร์กจะให้ผลลัพธ์ของขอบภาพดีที่สุด เนื่องจากวิธีนี้สามารถใช้หาขอบภาพอย่างได้ผลกับภาพทั่วไปมากกว่าวิธีที่ใช้ฟิชชันเซตหรือนิวรอลเน็ตเวิร์กเพียงอย่างเดียว

ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การตอบสนองต่อสัญญาณรบกวนสำหรับวิธีที่เสนอกันทั้ง 3 วิธี การหาขอบภาพโดยใช้นิวรอลเน็ตเวิร์คมีการตอบสนองต่อสัญญาณรบกวนมากที่สุด เนื่องจากอินพุทที่ป้อนเข้าไปเป็นค่าระดับความเข้มของจุดภาพโดยตรง ซึ่งไม่เหมือนกับการหาขอบภาพโดยพีชชีนิวรอลเน็ตเวิร์คและการหาขอบภาพโดยพีชชีเซตที่มีการหาค่าของตัวแปร เพื่ออธิบายความเป็นขอบภาพก่อนที่จะตัดสินใจว่าจุดภาพที่กำลังพิจารณาเป็นขอบภาพใช่หรือไม่ และเมื่อเปรียบเทียบกับวิธีธรรมดา (รูปที่ 6.24-6.34 และรูปที่ 6.38-6.46) ผลลัพธ์ที่ได้จากวิธีที่เสนอ โดยเฉพาะการหาขอบภาพโดยพีชชีนิวรอลเน็ตเวิร์ค จะให้ผลลัพธ์ของขอบภาพอยู่ในเกณฑ์ที่น่าพอใจ สำหรับวิธีธรรมดานั้น การตอบสนองของสัญญาณรบกวนจะขึ้นอยู่กับประสิทธิภาพของตัวดำเนินการหาขอบภาพเองและค่าเทรชโฮลด์ ซึ่งถ้าตัวดำเนินการใดตอบสนองต่อขอบภาพได้ดีการกำหนดค่าเทรชโฮลด์ที่เหมาะสมจะทำให้สามารถกำจัดสัญญาณรบกวนออกไปได้ แต่วิธีการหาขอบภาพแบบธรรมดา ค่าเทรชโฮลด์จะเป็นค่าคงที่เพียงค่าเดียว ถ้าขอบภาพบริเวณใดต่ำกว่าค่าเทรชโฮลด์ก็จะทำให้ขอบภาพในช่วงนั้นหายไปด้วย แต่วิธีการหาขอบภาพโดยใช้พีชชีนิวรอลเน็ตเวิร์คค่าเทรชโฮลด์ที่ใช้ในการกำหนดขอบภาพจะมีหลายระดับขึ้นอยู่กับข้อมูลของภาพนั้นๆ ดังนั้นแม้ขอบภาพจะอยู่ในช่วงค่าความเข้มต่ำวิธีนี้ก็ยังสามารถตรวจจับขอบภาพได้

- สัญญาณรบกวนที่มีการแจกแจงแบบยูนิฟอร์ม จะมีผลกระทบต่อตัวดำเนินการหาขอบภาพมากกว่าสัญญาณรบกวนที่มีการแจกแจงแบบ Gaussian ดังจะเห็นได้จากผลลัพธ์ของแต่ละวิธีที่แสดงในรูปที่ 6.21-6.46

- สำหรับการวัดประสิทธิภาพและความถูกต้องของขอบภาพที่ได้จะพิจารณาได้จากค่า P_e และ P (ซึ่งกล่าวในขงที่ 1 หน้า 22) โดยขอบภาพที่มีความถูกต้องสูงทั้งค่า P_e และ P จะต้องมีค่าเข้าใกล้ 1 จากตารางที่ 6.2 การหาขอบภาพโดยใช้พีชชีเซตสามารถหาขอบภาพได้ถูกต้องมากที่สุด สำหรับตารางที่ 6.3 การหาขอบภาพโดยพีชชีนิวรอลเน็ตเวิร์คสามารถหาขอบภาพได้ถูกต้องมากที่สุด

- ความสะดวกในการใช้งาน วิธีหาขอบภาพที่ได้เสนอไปแล้วทั้ง 3 วิธี ผู้ใช้ไม่ต้องดำเนินการใดๆ ในขั้นตอนการหาขอบภาพ เพียงแต่ป้อนข้อมูลภาพเข้าไปตัวดำเนินการหาขอบภาพที่เสนอจะทำการหาขอบภาพเองโดยอัตโนมัติ ส่วนวิธีธรรมดาผู้ใช้ต้องกำหนดค่าเทรชโฮลด์ที่เหมาะสมให้กับแต่ละภาพซึ่งต้องใช้เวลานานพอสมควร และบางวิธียังต้องคอยปรับค่าตัวแปรให้ใช้ได้กับแต่ละภาพอีกด้วย เช่นวิธีของ Marr-Hildreth จะต้องปรับค่า σ , วิธีของ Canny จะต้องปรับค่า σ , ส่วนวิธี Gabor จะต้องมีการปรับทั้งค่า σ และ ω และสำหรับวิธี Relaxation จะต้องมีการปรับค่า d และค่า q ที่เหมาะสมมีฉะนั้นจำนวนรูปที่จะใช้ในการคำนวณขอบภาพก็จะมาก

บทที่ 7 สรุปลงและวิจารณ์

การหาขอบภาพเป็นขั้นตอนที่สำคัญ ของการวิเคราะห์ข้อมูลเชิงตัวเลขเพื่อที่จะหาขอบของวัตถุที่ปรากฏอยู่ในภาพ และขอบของวัตถุที่ได้จากกระบวนการนี้จะนำไปใช้แบ่งแยกวัตถุแต่ละชนิดเพื่อทำให้คอมพิวเตอร์รู้จักวัตถุชนิดนั้นๆ ต่อไป

วิธีการหาขอบภาพโดยทั่วไปผู้ใช้จะต้องกำหนดค่าเทรชโฮลด์ขึ้นมาเช่นวิธีของ Sobel, Frei-Chen, Laplacian เป็นต้น หรือบางวิธีจะต้องมีการกำหนดค่าตัวแปรต่างๆ เพื่อให้เหมาะสมกับแต่ละภาพเช่นวิธีของ Marr-Hildreth, Canny, Gabor, Relaxation เป็นต้น ซึ่งการกำหนดค่าเทรชโฮลด์หรือค่าตัวแปรต่างๆ ของแต่ละวิธีในแต่ละภาพจะต้องเวลานานพอสมควร จึงจะได้ขอบภาพที่สมบูรณ์และถูกต้อง อย่างเช่นในกรณีที่ภาพมีสัญญาณรบกวนปะปนอยู่ด้วยและเป็นภาพที่มีขอบอยู่ในช่วงที่มีค่าความเข้มต่ำ การกำหนดค่าเทรชโฮลด์ซึ่งเป็นค่าคงที่เพียงค่าเดียวจะดำเนินการได้ยากมากคือ เมื่อต้องการกำจัดสัญญาณรบกวนจะต้องกำหนดให้ค่าเทรชโฮลด์ให้มีค่าสูง ซึ่งจะทำให้ขอบภาพที่อยู่ในช่วงค่าความเข้มต่ำๆ หายไป และในทางตรงกันข้ามถ้าต้องการให้ขอบภาพในช่วงค่าความเข้มต่ำๆ ไม่สูญหายไปก็ต้องกำหนดค่าเทรชโฮลด์ให้ต่ำ ซึ่งจะทำให้ขอบภาพมีสัญญาณรบกวนและจะส่งผลให้ขอบภาพที่อยู่ในช่วงค่าความเข้มสูงหนาขึ้นอีกด้วย

จากปัญหาต่างๆ ที่เกิดขึ้นในขั้นตอนการหาขอบภาพโดยวิธีธรรมดาที่กล่าวมาแล้วข้างบน เราจึงพัฒนาวิธีการหาขอบภาพขึ้นมาใหม่โดยอาศัยฐานความรู้เกี่ยวกับขอบภาพมาเป็นตัวแปรสำคัญในการพัฒนาครั้งนี้โดยใช้ ฟิชชีเซต และนิรอลเนตเวิร์คเป็นเครื่องมือ ซึ่งวิธีที่พัฒนาขึ้นมามี 4 วิธีด้วยกัน โดยเริ่มจากวิธีการหาขอบภาพโดยการประมวลผลก่อนโดยใช้ฟิชชีเซต, การหาขอบภาพโดยฟิชชีเซต, การหาขอบภาพโดยนิรอลเนตเวิร์ค และการหาขอบภาพโดยฟิชชีนิรอลเนตเวิร์ค ตามลำดับ ซึ่งแต่ละวิธีมีหลักการและปัญหาต่างๆ ดังนี้

- การหาขอบภาพโดยการประมวลผลก่อนโดยใช้ฟิชชีเซต ซึ่งวิธีนี้จะใช้ฟิชชีเซตมาปรับภาพเพื่อให้ใกล้เคียงกับภาพใบนารีมากที่สุดแต่ยังคงสภาพของขอบเอาไว้แล้วจึงทำการหาขอบภาพ ขั้นตอนสำคัญและปัญหาของวิธีนี้จะอยู่ที่การแบ่งข้อมูลออกเป็นช่วงๆ เพื่อทำการปรับภาพ ซึ่งถ้ามีการแบ่งช่วงของข้อมูลอย่างถูกต้องแล้วจะได้ขอบภาพที่สมบูรณ์ทีเดียว

- การหาขอบภาพโดยฟิชชีเซต วิธีนี้จะใช้กฎฟิชชีมาหาความสัมพันธ์ของจุดภาพกับโครงสร้างของขอบภาพที่กำหนดขึ้นซึ่งประกอบด้วย 20 โครงสร้าง กฎฟิชชีจะหาโครงสร้างของขอบภาพที่เหมาะสมให้กับจุดภาพแต่ละจุดโดยพิจารณาพร้อมกับจุดภาพในบริเวณใกล้เคียงอีก 8 จุด แล้วมาวิเคราะห์จุดภาพนั้นว่าเป็นขอบภาพหรือไม่โดยใช้ค่าความเป็นขอบภาพ, ค่าเฉลี่ย, ค่าส่วนเบี่ยงเบนมาตรฐาน, ค่าดัชนีฟิชชี ของจุดภาพภายในวินโดว์ขนาด 3×3 ปัญหาของวิธีนี้จะอยู่ที่ขั้นตอนการหาโครงสร้างที่เหมาะสมให้กับจุดภาพคือถ้ามีสัญญาณรบกวนปะปนเข้ามาจะทำให้เกิดความผิดพลาดขึ้นได้

- การหาขอบภาพโดยนิรอลเนตเวิร์ค วิธีนี้จะใช้นิรอลเนตเวิร์คมาหาขอบภาพโดยป้อนค่าความเข้มของจุดภาพที่เป็นวินโดว์ขนาด 3×3 และนิรอลเนตเวิร์คจะสร้างฟังก์ชันความสัมพันธ์(ค่าถ่วงน้ำหนัก)

ขึ้นมาเพื่อระบุว่าจุดภาพใดบางเป็นขอบภาพ แต่วิธีนี้จะมีปัญหาอย่างมากเมื่อภาพมีสัญญาณรบกวนหรือถ้าข้อมูลภาพที่ต้องการหาขอบมีการแจกแจงที่ผิดไปจากข้อมูลที่ใช้สอนนิเวรอลเน็ตเวิร์คมากๆ โครงข่ายอาจจะหาขอบภาพไม่ได้เลยก็ได้ ปัญหาจึงอยู่ที่การเลือกแพทเทิร์นที่ใช้สอนให้เหมาะสม

- การหาขอบภาพโดยใช้แบบจำลองพีชชีนิเวรอลเน็ตเวิร์ค เป็นการหาขอบภาพโดยใช้ตัวแปรที่ได้จากการวิเคราะห์เนื้อภาพ (Texture) โดยตัวแปรนี้จะมีคุณสมบัติที่บอกถึงความเป็นขอบภาพ สำหรับพีชชีเซตจะถูกนำมาใช้ในการคำนวณระดับของความเป็นขอบภาพของตัวแปรที่ได้จากเนื้อภาพว่าค่าของตัวแปร ณ จุดภาพนั้นๆ มีความเป็นขอบภาพมากน้อยเพียงใด และค่าระดับความเป็นขอบภาพที่ได้จากขั้นตอนของพีชชีเซตก็จะถูกสร้างความสัมพันธ์ไปยังขอบภาพโดยใช้คุณสมบัติการสร้างความสัมพันธ์แบบไม่เป็นเชิงเส้นของนิเวรอลเน็ตเวิร์ค สำหรับขอบภาพที่ได้จากวิธีนี้จัดว่าอยู่ในเกณฑ์ที่ดี เมื่อเปรียบเทียบกับวิธีการหาขอบภาพที่รู้จักกันโดยทั่วไปเช่นวิธีของ Sobel, Marr-Hildreth, Gabor และ Relaxation เป็นต้น และที่สำคัญวิธีนี้จะเป็นการหาขอบภาพแบบอัตโนมัติ ดังนั้นผู้ใช้จะไม่ต้องดำเนินการใดๆ ทั้งสิ้นในขั้นตอนการหาขอบภาพ ผู้ใช้เพียงแค่ป้อนข้อมูลภาพเข้าไป ระบบการหาขอบภาพโดยวิธีนี้ก็จะทำการประมวลผลจนได้ผลลัพธ์เป็นขอบภาพที่ถูกต้อง ซึ่งเราได้ทำการวัดความถูกต้องของขอบภาพใน 2 ลักษณะด้วยกันคือ การวัดความถูกต้องในตำแหน่งของขอบภาพ และการวัดเปอร์เซ็นต์ของขอบภาพที่หาได้โดยเทียบกับขอบภาพจริง สำหรับปัญหาที่อาจจะเกิดขึ้นได้ในการหาขอบภาพคือ ถ้าขอบของภาพที่มีการแจกแจงผิดไปจากเดิมมากๆ อาจจะทำให้ขอบภาพบางส่วนหายไปได้ สำหรับการแก้ไขในจุดนี้คือทำการวิเคราะห์ฟังก์ชันความเป็นสมาชิกของความเป็นขอบภาพใหม่ และทำการสอนโครงข่ายโดยข้อมูลภาพที่ทำให้โครงข่ายนี้ไม่สามารถหาขอบได้อย่างสมบูรณ์ใหม่

เอกสารอ้างอิง

- [1] S. K. Pal and Dwijesh K. Dutta Majumder, Fuzzy Mathematical Approach to Pattern Recognition, New Delhi: Wiley Eastern Limited, 1986.
- [2] Chin-Wang Tao and W. E. Thompson, "A Fuzzy If-Then Approach to Edge Detection," IEEE International Conference on Fuzzy Systems, Vol. 2, pp. 1356-1360, March 28-April 1, 1993.
- [3] S. Lu and A. Szeto, "Hierarchical Artificial Neural Networks for Edge Enhancement," Pattern Recognition, Vol. 26, No. 8, pp. 1149-1163, 1993.
- [4] J. K. Paik, J. C. Brailean, and A. K. Katsaggelos, "An Edge Detection Algorithm Using Multi-State Adalines," Pattern Recognition, Vol. 25, No. 12, pp. 1495-1504, 1992.
- [5] S. M. Bhandakar, V. Zhang and W. D. Potter, "An Edge Detection Technique Using Genetic Algorithm Based Optimization," Pattern Recognition, Vol. 27, No. 9, pp. 1159-1180, 1994.
- [6] H. L. Tan, S. B. Gelfand, and B. J. Delp, "A Cost Minimization Approach to Edge Detection Using Simulated Annealing," IEEE Trans. Patt. Anal. Machine Intell., Vol. 14, pp. 3-18, Jan 1992.
- [7] J. C. Simon and R. M. Haralick (Eds.), Digital Image Processing, pp. 105-148, D. Reidel Publishing Company, 1981.
- [8] W. K. Pratt, Digital Image Processing, Second Edition, USA: Jon Wiley & Sons, Inc., 1991.
- [9] R. C. Gonzalez and R. E. Woods, Digital Image Processing, USA: Addison-Wesley, 1992.
- [10] D. Marr and E. Hildreth, "Theory of Edge Detection," Proc. R. Soc. Lond., B 207, pp. 187-217, 1980.
- [11] A. Papoulis, Probability Random Variables and Stochastic Processes, Third Edition, Singapore: McGraw-Hill, Inc. pp. 603-612. 1991.
- [12] R. M. Haralick, "Digital Step Edges from Zero Crossing of the Second Directional Derivatives," IEEE Trans. Patt. Anal. Machine Intell., Vol. PAMI-5, pp. 58-68, 1984.
- [13] J. F. Canny, "A Computational Approach to Edge Detection," IEEE Trans. Patt. Anal. Machine Intell., Vol. 8, No. 6, pp. 679-698, 1986.
- [14] R. Mehrota, K. R. Namudur, and N. Ranganathan, "Gabor Filter-Based Edge Detection," Pattern Recognition, Vol. 25, No. 12, pp. 1479-1494, 1992.
- [15] J. G. Daugman, "Uncertainty Relation for Resolution in Space, Spatial Frequency and Orientation Optimized by Two-Dimensional Visual Cortical Filters," J. Opt. Soc. Am. A2(2), pp. 1160-1169, 1991.

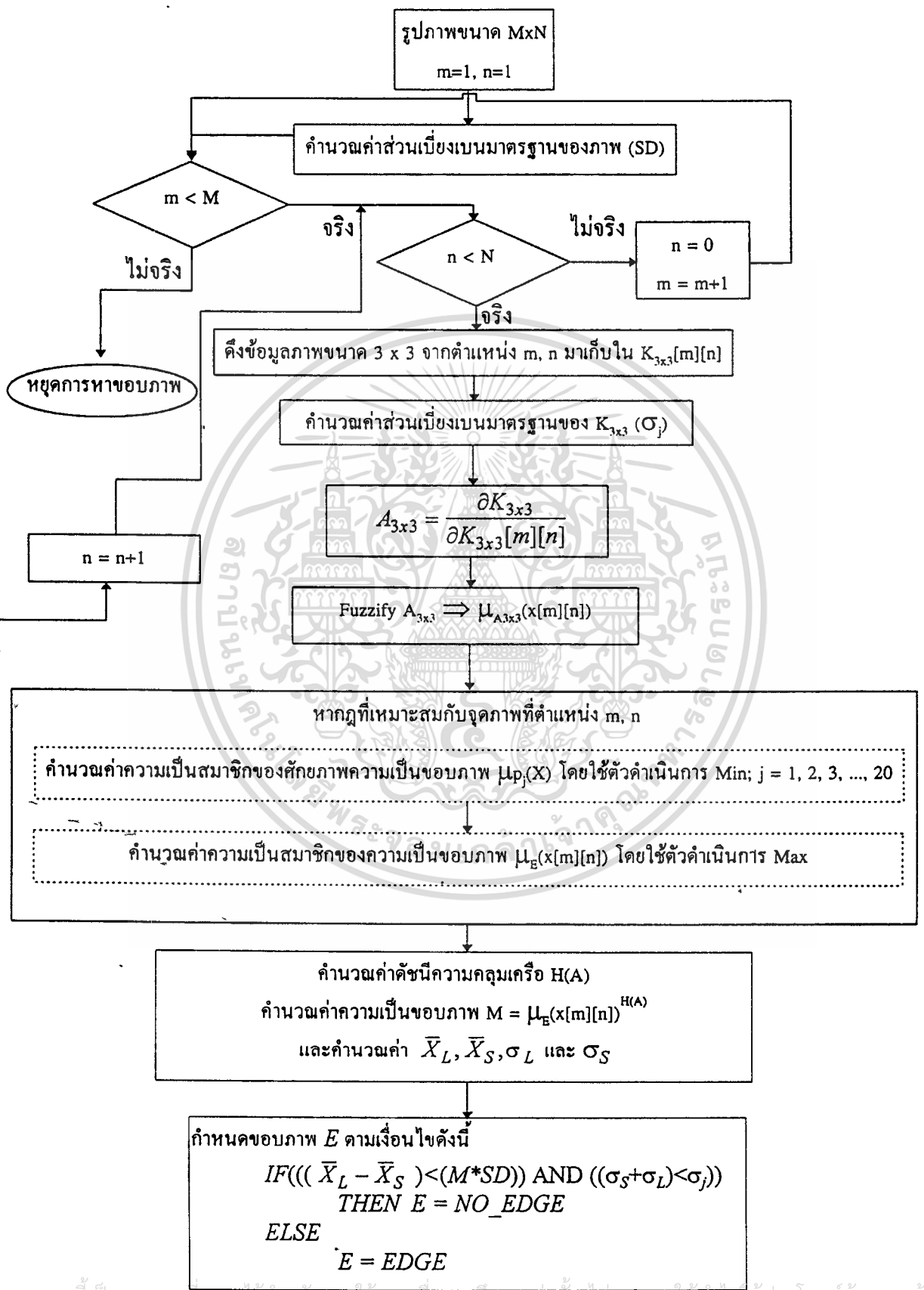
- [16] M. Lindenbaum, M. Fischer, and A. Bruckstein, "On Gabor's Contribution to Image Enhancement," *Pattern Recognition*, Vol. 27, No. 1, pp. 1-8, 1994.
- [17] A. C. Bovik, M. Clark, and W. S. Geisler, "Multichannel Texture Analysis Using Localized Spatial Filters," *IEEE Trans. Patt. Anal. Machine Intell.*, Vol. 12, No. 1, pp. 55-73, 1990.
- [18] D. Dunn, W. E. Higgins, and J. Wakeley, "Texture Segmentation Using 2-D Gabor Elementary Functions," *IEEE Trans. Patt. Anal. Machine Intell.*, Vol. 16, No. 2, pp. 130-149, 1994.
- [19] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing Analysis and Machine Vision*, UK: Chapman and Hall Computing, 1993.
- [20] D. H. Ballard, and C. M. Brown, *Computer Vision*, New Jersey, Prentice-Hall, Inc., 1982.
- [21] A. K. Jain, *Fundamentals of Digital Image Processing*, USA., Prentice-Hall, Inc., 1989.
- [22] R. M. Haralick, "Statistic and Structural Approaches to Texture," *Proceeding IEEE*, Vol. 67, No. 5, pp. 788-804, 1979.
- [23] Tsai-Hong, C. R. Dyes, and A. Rosenfeld, "Texture Primitive Extraction Using an Edge-based Approach," *IEEE Tran. Syst., Man, Cybern.*, Vol. SMC-10, No. 10, pp. 659-675, 1980.
- [24] L. Van Gool, P. Dewaele, and A. Oosterlinck, "Survey Texture Analysis Anno 1983," *Computer Vision, Graphics, and Image Processing* Vol. 29, pp. 336-357, 1985.
- [25] R. M. Haralick, and L. G. Shapiro, *Computer and Robot Vision Vol. I*, USA., Addison-Wesley, pp. 453-507, 1992.
- [26] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Texture Features for Image Classification," *IEEE Trans. Syst., Man, Cybern.*, Vol. 3, pp. 610-621, 1973.
- [27] J. Lee, R. C. Weger, S. K. Sengupta, and R. M. Welch, "A Neural Network Approach to Cloud Classification," *IEEE Trans. Geosc., Remote Sensing*, Vol. 28, No. 5, 1990.
- [28] L. A. Zadeh, "Outline of a New Approach to Analysis of Complex Systems and Decision Process," *IEEE Trans. Syst. Man Cybern.*, Vol. SMC-3, No. 1, pp. 28-44, Jan., 1973.
- [29] L. A. Zadeh, "The Concept of The Linguistic Variable and Its Application to Approximate Reasoning, Part 1," *Information Sciences*, Vol 8, pp. 199-249, 1975.
- [30] L. A. Zadeh, "Calculus of Fuzzy Restrictions," L. A. Zadeh, K. S. Fu, K. Tanaka, and M. Shimura, (Eds), *Fuzzy Set and Their Applications to Cognitive and Decision Processes*, Academic Press, London, 1975.
- [31] L. A. Zadeh, "Fuzzy Sets," *Information and Control*, Vol. 8, pp. 338-353, 1965.
- [32] H. J. Zimmermann, *Fuzzy Set Theory and Its Applications*, Second, Revised Edition, USA.: Kluwer Academic Publishers, 1991.

- [33] A. Kaufmann, Introduction to The Theory of Fuzzy Subsets, Vol. 1, New York, 1975.
- [34] L. A. Zadeh, "A Fuzzy-Set-Theoretic Interpretation of Linguistic Hedges," J. of Cybernetics, Vol. 2, No. 2, pp. 4-34, 1972.
- [35] E. Cox, "Adaptive Fuzzy Systems," IEEE Spectrum, pp. 67-70, February 1993.
- [36] E. Cox, "Fuzzy Fundamentals," IEEE Spectrum, pp. 58-61, October 1992.
- [37] B. Kosko, Neural Networks and Fuzzy Systems, A Dynamical Systems Approach to Machine Intelligence, Prentice Hall, Englewood Cliffs, NJ., 1992.
- [38] C. C. Lee, "Fuzzy Logic in Control Systems: Fuzzy Logic Controller, Part I," IEEE Trans. Syst., Man, Cybern., Vol. 20, pp. 404-418, 1990.
- [39] C. C. Lee, "Fuzzy Logic in Control Systems: Fuzzy Logic Controller, Part II," IEEE Trans. Syst., Man, Cybern., Vol. 20, pp. 419-435, 1990.
- [40] D. P. Filev, and R. R. Yager, "A Generalized Defuzzification Method Under BAD Distributions," Int. J. Intelligent Systems, Vol. 6, pp. 689-697, 1991.
- [41] R. R. Yager, and D. P. Filev, "SLIDE: A Simple Adaptive Defuzzification Method," IEEE Trans., Fuzzy Syst., Vol. 1, No. 1, pp. 69-78, 1993.
- [42] N. R. Pal, and J. C. Bezdek, "Measuring Fuzzy Uncertainty," IEEE Trans., Fuzzy Syst., Vol. 2, No. 2, pp. 107-118, 1994.
- [43] L. A. Zadeh, "Probability Measures of Fuzzy Events," J. Math. Analysis and Applicat., Vol. 23, pp. 421-427, 1968.
- [44] B. R. Ebanks, "On Measures of Fuzziness and Their Representations," J. Math. Anal. and Appl., Vol. 94, pp. 24-37, 1983.
- [45] N. R. Pal, and S. K. Pal, "Object-Background Segmentation Using New Definitions of Entropy," IEE Pro., 136, pt. E, pp. 284-295, 1989.
- [46] D. Bhandari and N. R. Pal, "Some New Information Measures for Fuzzy Sets," Inform. Sci., 1992.
- [47] N. R. Pal, J. C. Bezdek, and R. Hemasinha, "Uncertainty Measures for Evidential Reasoning I: A Review," Int. J. Approx. Reasoning, Vol. 7, pp. 165-183, 1992.
- [48] N. R. Pal, J. C. Bezdek, and R. Hemasinha, "Uncertainty Measures for Evidential Reasoning II: A New Measure," Int. J. Approx. Reasoning, Vol. 8, pp. 1-16, 1993.
- [49] R. R. Yager, "On Measures of Fuzziness and Negation, Part I: Membership in The Unit Interval," Int. J. Gen. Syst., Vol. 5, pp. 221-229, 1979.

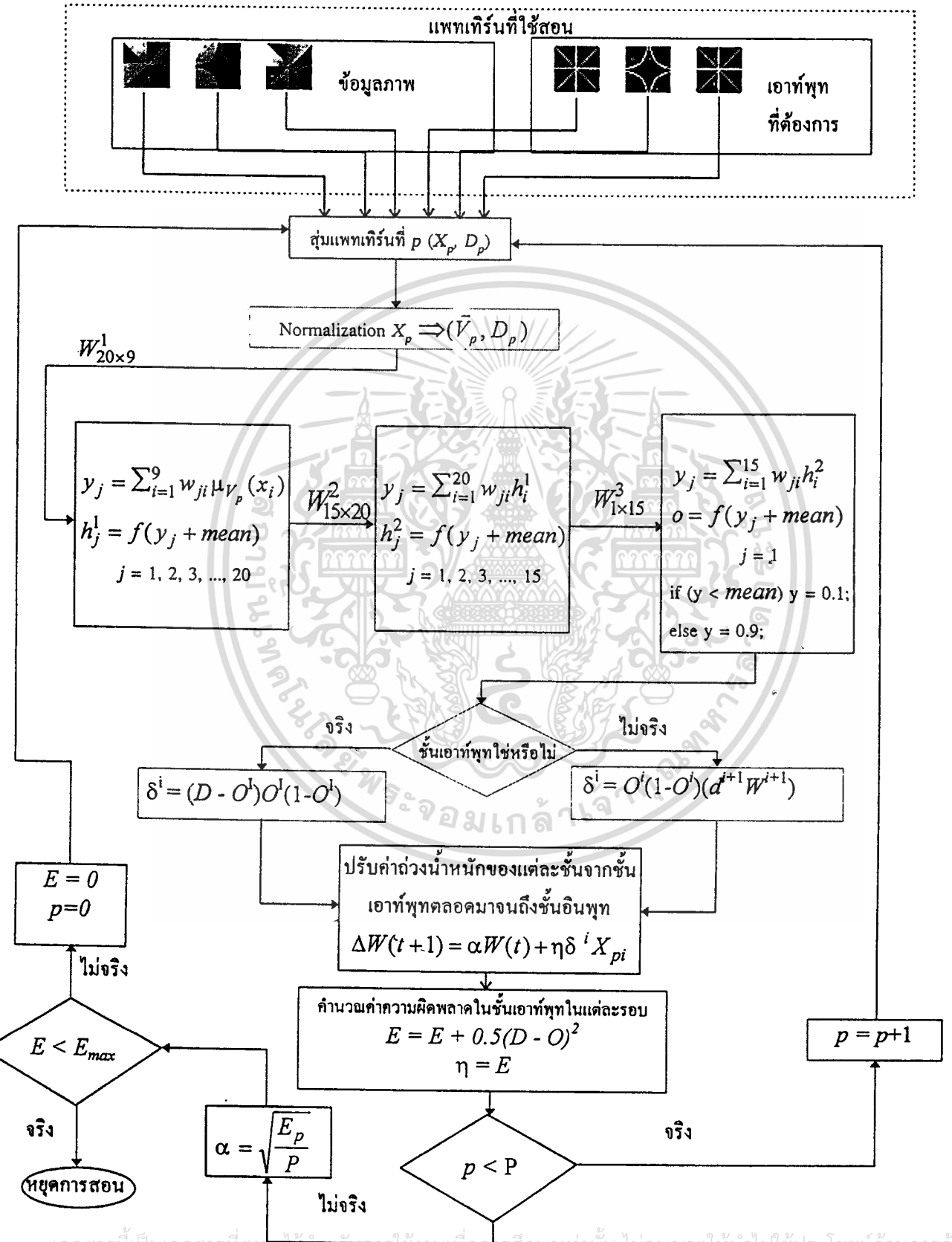
- [50] E. H. Mamdani, and S. Assilian, "An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller," Fuzzy Reasoning and Its Applications, Edited by E. H. Mamdani, and B. R. Gaines, Academic Press Inc. London, pp. 311-323, 1981.
- [51] กิตติ ไพฑูรย์วัฒนกิจ และ สาริต อินทจักร์, "การปรับปรุงภาพแบบส่วนย่อยโดยฟัซซี่เซต," การประชุมใหญ่ทางวิชาการประจำปี 2536 วิศวกรรมสถานแห่งประเทศไทยฯ, (27-30 พฤศจิกายน 2536) หน้า 591- 600.
- [52] กิตติ ไพฑูรย์วัฒนกิจ และ สาริต อินทจักร์, "การประยุกต์กฎฟัซซี่เพื่อการปรับปรุงภาพ," วารสารสมาคมคอมพิวเตอร์แห่งประเทศไทยในพระบรมราชูปถัมภ์, ปีที่ 20, ฉบับที่ 105 (มกราคม-กุมภาพันธ์ 2537) หน้า 32-40.
- [53] J. M. Zurada, Introduction to Artificial Neural Systems, West Publishing Company, Singapore, 1992.
- [54] ราตรี สุตทรวง, ประสาทศรัรวิทยา, สำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย, พิมพ์ครั้งที่ 2, (2535).
- [55] A. A. Kempka, "Activating Neural Networks : Part I," AI Expert, June, 1994.
- [56] A. A. Kempka, "Activating Neural Networks : Part II," AI Expert, Aug., 1994.
- [57] D. E. Rumelhart, and J. L. McClelland, Exploration in Parallel Distributed Processing; A Handbook of Model Programs; and Exercises, MIT Press, Massachusetts, USA., 1988.
- [58] D. E. Rumelhart, J. L. McClelland, and the PDP Research Group, Parallel Distributed Processing; Exploration in the Microstructure of Cognition, Volume 1: Foundation, MIT Press, Massachusetts, USA., 1986.
- [59] R. P. Lippmann, "An Introduction to Computing with Neural Nets," IEEE ASSP Mag., pp. 4-22, April 1987.
- [60] J. A. Freeman, and D. M. Skapara, Neural Networks, Applications, and Programming Techniques, Addison Wesley, USA., 1991.
- [61] กิตติ ไพฑูรย์วัฒนกิจ สาริต อินทจักร์ และ ไพลีน บุญเดช, "นิรอลเน็ตเวิร์คบนทรานสพิวดอร์กับการหาขอบภาพ," ประชุมใหญ่วิชาการทางวิศวกรรมประจำปี 2537, วิศวกรรมสถานแห่งประเทศไทยฯ, (2537) หน้า EE 75-90.
- [62] T. Kohonen, Self-Organization and Associative Memory, Springer-Verlag, USA., 1989.
- [63] กิตติ ไพฑูรย์วัฒนกิจ และ เศรษฐพล ลิมปราชญา, "การใช้นิรอลเน็ตเวิร์คเพื่อจดจำแพทเทิร์น," วารสารสำนักงานคณะกรรมการวิจัยแห่งชาติ, ปีที่ 25, เล่มที่ 1, (ม.ค.-มิ.ย. 2536) หน้า 45-67.
- [64] J. E. Dayhoff, Neural Network Architectures an Introduction, Van Nostrand Reinhold, 1990.

- [65] L. H. Tsoukalas, A. Ikononopoulos, and E. Uhrig, "Fuzzy Neural Control," Ed. Cihan H. Dagli, *Artificial Neural Networks for Intelligent Manufacturing*, pp. 413-434, Chapman&Hall, London, 1994.
- [66] Kaoru Hirota and Witold Pedrycz, "OR/AND Neuron in Modeling Fuzzy Set Connectives," *IEEE Trans. Fuzzy Syst.*, Vol. 2, No. 2, May 1994.
- [67] Hon Keung Kwan, Yaling Cai, and Bin Zhang, "Membership Function Learning in Fuzzy Classification," *Int. J. Electronics*, Vol. 74, No. 6, pp. 845-850, 1993.
- [68] M. Caudill, *Understanding Neural Networks: Computer Explorations*, Vol. 2, pp. 255-276, 1991.
- [69] J. J. Buckley and Yoichi Hayashi, "Fuzzy Neural Networks," Ed. by R. R. Yager and L. A. Zadeh, *Fuzzy Sets, Neural Networks, and Soft Computing*, Van Nostrand Reinhold, New York, pp. 233-249, 1994.
- [70] Yoichi Hayashi, J. J. Buckley, and E. Czogala, "Fuzzy Neural Network with Fuzzy Signals and Weights," *Int. J. Intelligent Syst.*, Vol. 8, pp. 527-537, 1993.
- [71] สาริต อินทจักร, นิมิตร บุญภิรมย์, และกิตติ ไพฑูรย์วัฒนกิจ, "การประมวลผลภาพดิจิทัลกับทฤษฎีฟuzzyเซต," การประชุมทางวิชาการวิศวกรรมไฟฟ้า ครั้งที่ 16, 2536, หน้า 625-629.
- [72] R. M. Haralick and J. S. Lee, "Context Dependent Edge Detection and Evaluation," *Pattern Recognition*, Vol. 23, No. 12, pp. 1-9, 1990.
- [73] D. J. Park, K. M. Man, and Rae-Hong Park, "Edge Detection in Noisy Images Based on The Co-occurrence Matrix," *Pattern Recognition*, Vol. 27, No. 6, pp. 765-775, 1994.

ภาคผนวก ก. แผนภูมิขั้นตอนการหาขอบภาพโดยใช้ฟัซซี่เซต



ภาคผนวก ข. แผนภูมิขั้นตอนการหาขอบภาพโดยใช้ DABP



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ภายใต้การใช้นี้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมีการดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก. อัลกอริทึมการสอนโครงข่ายพีชคณิตวีรลเนตเวิร์ค

```

#include <iostream.h>
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <math.h>

#define NUM_LAYERS 4
#define Max(a,b) ((a) > (b) ? (a) : (b))
#define Min(a,b) ((a) < (b) ? (a) : (b))

typedef float Type;
typedef Type * Vector;
typedef Vector * Matrix;

Type errDesired=0.01;

class Layer
{
    friend class BPN;
protected:
    Vector outputs, hidOut, errors;
    Matrix weights, last_delta;
    int ounits, hunits;
public:
    void init_layer(int o, int h);
    void init_wd();
};

void Layer::init_layer(int o, int h)
{
    ounits = o;
    hunits = h;
    errors = new Type[h];
    outputs = new Type[o];
    hidOut = new Type[h];
    weights = new Vector[h];
    last_delta = new Vector[h];
    for(int i=0; i<h; i++)
    {
        weights[i] = new Type[o];
        last_delta[i] = new Type[o];
    }
}

void Layer::init_wd()
{
    for(int i=0; i<hunits; i++)
        for(int j=0; j<ounits; j++)
        {
            weights[i][j] = 13.0*(1.0-random(100)/51.0);
            last_delta[i][j] = (1.0-random(100)/51.0);
        }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

class BPN
{
private:
    Layer      *layer;
    Type      alpha, eta, mean, SD;
    Vector outDesired;
public:
    BPN()
    {
        alpha = 0.05;
        eta = 2.0;
    }
    Type AccessSD()
    {
        return SD;
    }
    void insert_layer(Matrix);
    void init_weights();
    void set_input(int I[3][3], Vector, Vector);
    Vector propagate_layer(Layer *, Vector);
    void propagate_forward();
    void compute_output_error();
    void backpropagate_error(Layer *);
    void propagate_backward();
    void adjust_weights();
};

void BPN::insert_layer(Matrix N)
{
    Layer q;

    layer = new Layer[NUM_LAYERS-1];
    for(int i=0; i<NUM_LAYERS-1; i++)
    {
        q.init_layer(N[i][0], N[i][1]);
        layer[i] = q;
    }
    outDesired = new Type[N[i-1][1]];
}

void BPN::init_weights()
{
    for(int i=0; i<NUM_LAYERS-1; i++)
        layer[i].init_wd();
}

void BPN::set_input(int I[3][3], Vector input, Vector output)
{
    int K=3, max=0, min=255, D[3][3];

    for(int i=0; i<K; i++)
        for(int j=0; j<K; j++)
        {
            D[i][j] = abs(I[1][1] - I[i][j]);
            max = Max(max, D[i][j]);
            min = Min(min, D[i][j]);
        }
    if(max != min)

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าในกรณีใดๆ หากต้องการนำเอกสารนี้ไปใช้ซ้ำ กรุณาติดต่อขอขออนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    float y = -1.0;
    float x = (float)(max - min);
    float c = -1.0 * (float)max * (y/x);
    for(i=0; i<K; i++)
        for(int j=0; j<K; j++)
            {
                input[K*i+j] = (y/x * D[i][j] + c);
            }
}
else
{
    for(i=0; i<K; i++)
        for(int j=0; j<K; j++)
            input[K*i+j] = 0.0;
}
memcpy(layer[0].outputs, input, sizeof(Type)*layer[0].ounits);
memcpy(outDesired, output, sizeof(Type)*layer[NUM_LAYERS-2].hunits);
}

Vector BPN::propagate_layer(Layer *p, Vector inputs)
{
    Vector current;
    Vector connects;
    long double sum;

    current = p->hidOut;
    for(int i=0; i<p->hunits; i++)
    {
        sum = 0.0;
        connects = p->weights[i];
        for(int j=0; j<p->ounits; j++)
        {
            sum = sum + inputs[j] * connects[j];
        }
        current[i] = 1.0/(1.0 + expl(-sum));
        if( (p->hunits == layer[NUM_LAYERS-2].hunits) )
        {
            if(current[i] < 0.5)
                current[i] = 0.1;
            else
                current[i] = 0.9;
        }
    }
    p->outputs = current;
    return current;
}

void BPN::propagate_forward()
{
    Vector input=layer[0].outputs;

    for(int i=0; i<NUM_LAYERS-1; i++)
    {
        input = propagate_layer(&(layer[i]), input);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void BPN::backpropagate_error(Layer *p)
{
    Vector senders;
    Vector receivers;
    Vector connects;
    Type sum;

    senders = p->errors;
    receivers = p->outputs;
    for(int i=0; i<p->ounits; i++)
    {
        sum = 0.0;
        for(int j=0; j<p->hunits; j++)
        {
            connects = p->weights[j];
            sum = sum + senders[j] * connects[i];
        }
        receivers[i] = sum * receivers[i] * (1.0-receivers[i]);
    }
}

void BPN::propagate_backward()
{
    for(int i=NUM_LAYERS-2; i>0; i--)
    {
        backpropagate_error(&(layer[i]));
        memcpy(layer[i-1].errors, layer[i].outputs, sizeof(Type)*layer[i].ounits);
    }
}

void BPN::compute_output_error()
{
    Vector outputs=layer[NUM_LAYERS-2].hidOut;
    Vector errors =layer[NUM_LAYERS-2].errors;

    for(int i=0; i<layer[NUM_LAYERS-2].hunits; i++)
        errors[i] = outputs[i]*(1.0-outputs[i])*(outDesired[i]-outputs[i]);
}

void BPN::adjust_weights()
{
    Matrix A, B;
    Vector units;
    Vector hidDelta;
    Vector errors;

    A = layer[NUM_LAYERS-2].last_delta;
    B = layer[NUM_LAYERS-2].weights;
    units = layer[NUM_LAYERS-3].hidOut;
    errors = layer[NUM_LAYERS-2].errors;
    for(int i=0; i<layer[NUM_LAYERS-2].hunits; i++)
    {
        Type bias, sum=0.0;

        for(int j=0; j<layer[NUM_LAYERS-2].hunits; j++)
            sum += errors[j];
        bias = eta*sum + alpha*A[i][layer[NUM_LAYERS-2].ounits-1];
        B[i][layer[NUM_LAYERS-2].ounits-1] += bias;
    }
}

```

```

A[i][layer[NUM_LAYERS-2].ounits-1] = bias;

for(j=0; j<layer[NUM_LAYERS-2].ounits; j++)
{
    A[i][j] = eta * (errors[i]*units[j]) + (alpha*A[i][j]);
    B[i][j] = B[i][j] + A[i][j];
}
}
for(i=NUM_LAYERS-3; i>=0; i--)
{
    A = layer[i].last_delta;
    B = layer[i].weights;
    hidDelta = layer[i+1].outputs;
    units = layer[i].outputs;
    for(int j=0; j<layer[i].hunits; j++)
    {
        Type bias, sum=0.0;

        sum = hidDelta[j];
        bias = eta*sum + alpha*A[j][layer[i].ounits-1];
        B[j][layer[i].ounits-1] += bias;
        A[j][layer[i].ounits-1] = bias;

        for(int k=0; k<layer[i].ounits; k++)
        {
            A[j][k] = eta*(hidDelta[j]*units[k])+(alpha*A[j][k]);
            B[j][k] = B[j][k] + A[j][k];
        }
    }
}

void main(void)
{
    BPN          b;
    Matrix       A;
    Vector       input;
    Type         output, error;
    int          KERNEL=3;

    A = new Vector[NUM_LAYERS-1];
    for(long i=0; i<NUM_LAYERS-1; i++)
        A[i] = new Type[2];
    A[0][0] = 9;    A[0][1] = 20;
    A[1][0] = 20;  A[1][1] = 15;
    A[2][0] = 15;  A[2][1] = 1;

    b.insert_layer(A);
    b.init_weights();

    input = new Type[9];

    BYTE **inIm=new BYTE*[4];
    inIm[0]=readImage32x32("pbn_line.img");
    inIm[1]=readImage32x32("pincurve.img");
    inIm[2]=readImage32x32("pin_line.img");
    inIm[3]=readImage32x32("pb_curve.img");

```

```

BYTE **outIm;
outIm=new BYTE *[4];
outIm[0]=readImage32x32("pl_edge.img");
outIm[1]=readImage32x32("pc_edge.img");
outIm[2]=readImage32x32("pl_edge.img");
outIm[3]=readImage32x32("pc_edge.img");

BYTE *temp=readImage32x32("pl_edge.img");

error = 10.0;

for(long p=0; error > 0.0010;)
{
    short m,n=0;

    error = 0.0;
    for(int row=0; row<XMAX-2; row++)
        for(int col=0; col<YMAX-2; col++)
        {
            n = random(4);
            m=n;
            int I[3][3];
            for(int j =0; j<KERNEL; j++)
                for(int k=0; k<KERNEL; k++)
                    I[j][k] = inIm[n][XMAX*(j+row)+(col+k)];
            if(outIm[m][XMAX*(row+1)+(col+1)])
                output = 0.9;
            else
                output = 0.1;
            b.set_input(I, input, &output);
            b.propagate_forward();
            temp[XMAX*row+col] = (BYTE)(b.get_output()*255);
            b.compute_output_error();
            b.propagate_backward();
            p++;
            b.adjust_weights();
        }
}
}

```

ภาคผนวก ง. บทความและงานวิจัยที่ได้รับการตีพิมพ์

1. สาริต อินทจักร์ และกิตติ ไพฑูรย์วัฒนกิจ, "ระบบฐานข้อมูลภาพ," วารสารสมาคมคอมพิวเตอร์แห่งประเทศไทยในพระบรมราชูปถัมภ์, ปีที่ 20, ฉบับที่ 100/มีนาคม-เมษายน, หน้า 35-41, 2536
2. กิตติ ไพฑูรย์วัฒนกิจ และสาริต อินทจักร์, "การปรับปรุงภาพอย่างอัตโนมัติโดยใช้พีซีเซต," วารสาร ทางวิชาการวิศวกรรมสถานแห่งประเทศไทยฯ, ปีที่ 46, ฉบับที่ 6, หน้า 86-89, 2536
3. กิตติ ไพฑูรย์วัฒนกิจ และสาริต อินทจักร์, "การประมวลผลก่อนสำหรับภาพสีเทียม," วารสารสมาคมคอมพิวเตอร์แห่งประเทศไทยในพระบรมราชูปถัมภ์, ปีที่ 20, ฉบับที่ 102/กรกฎาคม-สิงหาคม หน้า 41-46, 2536
4. สาริต อินทจักร์ นิมิตร บุญภิรมย์ และกิตติ ไพฑูรย์วัฒนกิจ, "การประมวลผลภาพดิจิทัลกับทฤษฎีพีซีเซต", การประชุมทางวิชาการวิศวกรรมไฟฟ้า ครั้งที่ 16, หน้า 625-629, 2536
5. กิตติ ไพฑูรย์วัฒนกิจ และสาริต อินทจักร์, "การปรับปรุงแบบส่วนย่อยโดยพีซีเซต," ประชุมใหญ่วิชาการทางวิศวกรรม ประจำปี 2536 วิศวกรรมสถานแห่งประเทศไทยฯ, หน้า 591-600, 2536
6. กิตติ ไพฑูรย์วัฒนกิจ และสาริต อินทจักร์, "การประยุกต์ทฤษฎีพีซีเซตเพื่อการปรับปรุงภาพ," วารสารสมาคมคอมพิวเตอร์แห่งประเทศไทยในพระบรมราชูปถัมภ์, ปีที่ 20, ฉบับที่ 105/มกราคม-กุมภาพันธ์, หน้า 32-40, 2537
7. กิตติ ไพฑูรย์วัฒนกิจ สาริต อินทจักร์ และไพสิน บุญเดช, "นิเวรอลเน็ตเวิร์คบนทรานสพิวเตอร์กับการหาขอบภาพ", ประชุมใหญ่วิชาการทางวิศวกรรมประจำปี 2537 วิศวกรรมสถานแห่งประเทศไทยฯ, หน้า EE 75-90, 2537
8. กิตติ ไพฑูรย์วัฒนกิจ และสาริต อินทจักร์, "การปรับปรุงภาพเทอร์มัลอินฟราเรด," วารสาร NECTEC, หน้า 15-24, ฉบับที่ 4: มกราคม-มีนาคม 2538
9. กิตติ ไพฑูรย์วัฒนกิจ และสาริต อินทจักร์, "การประมวลผลภาพเชิงตัวเลข : ตอนที่ 1. (การหาขอบภาพ)," วารสาร NECTEC, หน้า 28-42, ฉบับที่ 5: เมษายน-มิถุนายน 2538

ประวัติผู้เขียน

ชื่อผู้เขียน : นายสาธิต อินทจักร์
 วัน เดือน ปีเกิด : วันที่ 19 เดือน ธันวาคม พ.ศ. 2512
 สถานที่เกิด : 39 หมู่ที่ 3 ต. พุ่งขมิ้น อ. นาหม่อม จ. สงขลา
 วุฒิกศีกษาระดับปริญญาตรี : วิทยาศาสตร์บัณฑิต สาขาสถิติศาสตร์
 สถานที่สำเร็จการศึกษา : มหาวิทยาลัยรามคำแหง
 ปีที่สำเร็จการศึกษา : 2533
 ผลงานทางวิชาการที่ได้รับการตีพิมพ์ :

1. ระบบฐานข้อมูลภาพ
2. การปรับปรุงภาพอย่างอัตโนมัติโดยใช้พีซีเซต
3. การประมวลผลก่อนสำหรับภาพสีเทียม
4. การประมวลผลภาพดิจิทัลกับทฤษฎีพีซีเซต
5. การปรับปรุงแบบส่วนย่อยโดยพีซีเซต
6. การประยุกต์ทฤษฎีพีซีเซตเพื่อการปรับปรุงภาพ
7. นวัตกรรมเน็ตเวิร์คบนทรานสฟิวเคอร์กับการหาขอบภาพ
8. การปรับปรุงภาพเทอร์มัลอินฟราเรด
9. การประมวลผลภาพเชิงตัวเลข : ตอนที่ 1. (การหาขอบภาพ)

รางวัลหรือทุนที่เคยได้รับ : ทุนโครงการข้างเผือก
 ประสบการณ์การทำงาน : โปรแกรมเมอร์, ผู้ช่วยนักวิจัย
 อาชีพปัจจุบัน : นักวิเคราะห์ระบบ