

การประมวลผลข้อมูลภาพ  
(DIGITAL IMAGE PROCESSING)

ผู้ช่วยศาสตราจารย์ ดร.กฤษณะ ชินสาร

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์  
มหาวิทยาลัยบูรพา อ.เมือง จ.ชลบุรี  
2550

## คำนำ

เอกสารประกอบการสอนเล่มนี้ใช้ประกอบการเรียนการสอนวิชา 310453 การประมวลผลภาพเบื้องต้น (Introduction to Digital Image Processing) โดยในรายวิชานี้จะทำให้ทราบถึง ทฤษฎี และ การโปรแกรมข้อมูลภาพ (Digital Image) ด้วยคอมพิวเตอร์ ได้แก่ หลักการเบื้องต้นทางการประมวลผลข้อเชิงตัวเลขเบื้องต้น เช่น การแทนข้อมูลการแปลงโดเมน การนิยามและแบบจำลองทางสถิติ และการแทนข้อมูลแบบหลายระดับ การสร้างข้อมูลภาพและการประมวลผลขั้นต้น

การดำเนินการแบบเฉพาะจุดและกลุ่มจุดภาพรบกวน การปรับปรุงคุณภาพข้อมูล การกรองและการสร้างภาพย้อนกลับ การแยกคุณลักษณะเด่น การแบ่งข้อมูล การหาขอบวัตถุภาพ เนื้อภาพ การวิเคราะห์ภาพ พื้นทีรูปร่างและการจำแนกข้อมูล และการบีบอัดข้อมูลภาพ

ผู้จัดทำหวังเป็นอย่างยิ่งว่าเอกสารประกอบการสอนเล่มนี้ จะช่วยให้ผู้ศึกษาเข้าใจหลักการ ทฤษฎี และ การโปรแกรมเพื่อการประมวลผลภาพ ทั้งนี้ก็เพื่อให้โปรแกรมที่เขียนขึ้นสามารถใช้งานได้เต็มประสิทธิภาพ

สุดท้ายนี้ผู้จัดทำ ขอขอบพระคุณ ครู-อาจารย์ทุกท่านที่เมตตาสั่งสอนจนสามารถเขียนเอกสารประกอบการสอนเล่มนี้ ขอขอบคุณ คณะอาจารย์ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยบูรพา ทุกท่านที่กรุณาให้คำแนะนำและคำปรึกษาเกี่ยวกับการเขียนเอกสารประกอบการสอนซึ่งช่วยให้เอกสารการสอนเล่มนี้มีความสมบูรณ์ยิ่งขึ้น

กฤษณะ ชินสาร, Ph.D.

ตุลาคม 2548

## สารบัญ

คำนำ . . . . .	I
สารบัญตาราง . . . . .	VI
สารบัญรูป . . . . .	VII
1 บทนำ . . . . .	1
1.1 บทนำ . . . . .	1
1.2 ภาพและรูป (Images and Pictures) . . . . .	2
1.3 ความหมายการประมวลผลข้อมูลภาพ . . . . .	2
1.4 ทำไมจึงต้องประมวลผลภาพ . . . . .	3
1.5 ข้อมูลภาพคืออะไร (What is an Image?) . . . . .	4
1.5.1 ขนาดของข้อมูลภาพ (Image size) . . . . .	6
1.5.2 ความคมชัดของภาพ (Image Resolution) . . . . .	7
1.5.3 ความสว่างของภาพ (Brightness) . . . . .	7
1.6 การประมวลผลข้อมูลภาพเบื้องต้น . . . . .	8
1.6.1 ตัวแบบการประมวลผลภาพอย่างง่าย . . . . .	8
1.6.2 การคอนโทซ์เซชัน (Image Quantization) . . . . .	8
1.6.3 การดำเนินการในการประมวลผลข้อมูลภาพ . . . . .	8
1.6.4 ตัวดำเนินการเชิงเส้น . . . . .	9
1.6.5 ตัวดำเนินการแบบกองซ้อน: Stacking operator . . . . .	10
1.7 สรุป . . . . .	13
2 Image Transformations . . . . .	16
2.1 บทนำ . . . . .	17
2.2 การคูณเวกเตอร์ . . . . .	17
2.3 Unitary transform . . . . .	18
2.4 การประยุกต์ใช้เมตริกซ์ $U$ และ $V$ . . . . .	19
2.4.1 Applications: Independent Component Analysis . . . . .	20
2.5 Fourier Transform . . . . .	24
2.6 สรุป . . . . .	26
3 Statistical Description of Images . . . . .	27
3.1 บทนำ . . . . .	29
3.2 ทำไมจึงต้องสถิติ . . . . .	29
3.3 ตัวแปรสุ่ม (Random variable) . . . . .	30

3.4	ความน่าจะเป็นของเหตุการณ์ (Probability of Events) . . . . .	30
3.5	ฟังก์ชันการแจกแจงและฟังก์ชันความหนาแน่นความน่าจะเป็น (Probability Distribution and Density Functions) . . . . .	31
3.5.1	การแจกแจงเวกเตอร์สุ่ม (Distribution of a Random Vector) . . .	31
3.5.2	ฟังก์ชันการแจกแจงความน่าจะเป็นร่วมและความหนาแน่นความน่า จะเป็นร่วม (Joint Distribution and Probability Density Functions) . . . . .	32
3.6	ค่าคาดหวัง (Expectation) . . . . .	34
3.7	การทดสอบความอิสระของข้อมูล . . . . .	34
3.8	สรุป . . . . .	35
4	การปรับปรุงภาพ Image Enhancement . . . . .	36
4.1	บทนำ . . . . .	37
4.2	การดำเนินการบนโดเมนภาพเชิงพื้นที่ (Spatial domain method) . . . . .	38
4.3	การดำเนินการบนโดเมนภาพเชิงความถี่ (Frequency domain method) . . .	39
4.4	การปรับปรุงภาพด้วยฮิสโตแกรม (Histogram) . . . . .	39
4.5	การปรับปรุงค่าฮิสโตแกรมสำหรับภาพ . . . . .	40
4.5.1	Histogram Equalization . . . . .	42
4.5.2	ขั้นตอนการทำ Histogram Equalization . . . . .	42
4.5.3	การดำเนินการด้วย Histogram แบบอื่นๆ . . . . .	44
4.6	การหาผลต่างของภาพ Image Subtraction . . . . .	44
4.7	การดำเนินการบนโดเมนภาพเชิงพื้นที่ (Mask processing or Spatial do- main method) . . . . .	45
4.8	ตัวกรองปรับเรียบ (Smoothing Filters) . . . . .	45
4.8.1	ตัวกรองปรับเรียบแบบเชิงเส้น (Linear Smoothing Filters) . . . . .	45
4.8.2	ตัวกรองปรับเรียบแบบไม่เชิงเส้น (Non-linear Smoothing Filters) .	47
4.9	ตัวกรองปรับสว่าง (Sharpen Filters) . . . . .	48
4.9.1	ตัวกรองความถี่สูง (High-pass filters (Linear Filter)) . . . . .	48
4.9.2	การหาตัวกรองความถี่สูงด้วยวิธีการหาอนุพันธ์ . . . . .	48
4.10	สรุป . . . . .	52
5	การกู้คืนภาพ Image Restoration . . . . .	59
5.1	บทนำ . . . . .	60

5.2	ตัวแบบของการเสื่อมของภาพ	
	Model of Image degradation . . . . .	60
5.2.1	สมมติฐานของตัวกรอง (Assumption of $H[.] = h(x, y)$ ) . . . . .	61
5.3	ตัวแบบของการเสื่อมของภาพแบบต่อเนื่อง . . . . .	62
5.4	ตัวแบบของการเสื่อมของภาพแบบไม่ต่อเนื่อง . . . . .	62
5.5	การคำนวณหาภาพต้นฉบับ $f(x, y)$ . . . . .	64
5.5.1	การกู้คืนภาพด้วยเทคนิค Principal Component Analysis: PCA . . . . .	64
5.6	บทประยุกต์: การกู้คืนภาพต้นฉบับด้วยเทคนิค PCA เมื่อไม่ทราบค่า $H[.]$ . . . . .	65
5.7	บทประยุกต์: การกู้คืนภาพต้นฉบับด้วยเทคนิค PCA เมื่อไม่ทราบค่า $H[.]$ . . . . .	68
6	การแยกภาพ และการหาขอบวัตถุในภาพ	
	Image Segmentation และ Edge Detection . . . . .	74
6.1	บทนำ . . . . .	75
6.2	วัตถุประสงค์และความจำเป็นการแยกภาพ . . . . .	75
6.3	วิธีการการแยกภาพ: Segmentation Method . . . . .	76
6.4	วิธีการหาขอบวัตถุในภาพ (Edge-based approaches) . . . . .	76
6.4.1	ขั้นตอนในการหาขอบวัตถุในภาพ . . . . .	76
6.4.2	การหาจุดภาพ (Point Detection) . . . . .	77
6.4.3	การหาสายเส้น (Line Detection) . . . . .	78
6.4.4	การหาขอบวัตถุ . . . . .	80
6.4.5	Thresholding . . . . .	85
6.5	การแยกภาพโดยวิธีหาพื้นที่ (Region-based approaches) . . . . .	87
6.5.1	Region growing method . . . . .	89
6.5.2	Region splitting and merging . . . . .	90
7	การเข้ารหัส และการบีบอัดข้อมูลภาพ	
	Image coding and Compression . . . . .	92
7.1	บทนำ . . . . .	93
7.2	วัตถุประสงค์ของการเข้ารหัสและการบีบอัดข้อมูลภาพ . . . . .	93
7.3	Huffman Coding . . . . .	93
7.4	Run-Length Encoding: RLE . . . . .	96
	บรรณานุกรม . . . . .	98
	Biography . . . . .	102
	ภาคผนวก . . . . .	103

## สารบัญตาราง

4.1	การคำนวณ Histogram Equalization . . . . .	43
7.1	รหัส Huffman สำหรับการเข้ารหัสจากรูปที่ 7.2 . . . . .	96

## สารบัญรูป

1.1	ตัวอย่างข้อมูลภาพที่จัดเก็บด้วยรายละเอียดที่ต่างกัน . . . . .	4
1.2	การกำจัดสัญญาณรบกวน . . . . .	5
1.3	ภาพระดับเทา . . . . .	14
1.4	ตัวอย่างข้อมูลภาพที่จัดเก็บด้วยรายละเอียดที่ต่างกัน . . . . .	15
2.1	The cocktail party problem. . . . .	21
2.2	การแจกแจงความน่าจะเป็นแบบต่างๆ . . . . .	22
2.3	การวิเคราะห์องค์ประกอบอย่างอิสระสำหรับแหล่งซึ่งมีค่า Kurtosis แบบผสม	24
2.4	ความถี่พื้นฐานของสัญญาณ $f_1$ $f_2$ และ $f_3$ ที่มาผสมกันเป็นสัญญาณเดียว .	26
3.1	ความแข็งของเจลาตูรีน . . . . .	29
3.2	ฟังก์ชันความหนาแน่นความน่าจะเป็น . . . . .	33
4.1	การปรับปรุงภาพ . . . . .	37
4.2	การปรับปรุงภาพบนโดเมนเชิงพื้นที่และ โดเมนเชิงความถี่ . . . . .	38
4.3	ภาพ 8 ระดับ การแจกแจงความถี่ และ ฮิสโตแกรมของภาพ . . . . .	40
4.4	Pseudo code to compute the histogram of an image. . . . .	41
4.5	ภาพและฮิสโตแกรมของภาพ (a) ภาพตัวหนังสือต้นฉบับ (b) ฮิสโตแกรม ของภาพต้นฉบับ (a), (c) ภาพหลังจากการทำ Histogram Equalization และ (d) ฮิสโตแกรมใหม่ของภาพ . . . . .	53
4.6	การหาผลต่างภาพ (a) ภาพต้นฉบับ (b) ภาพต้นฉบับหลังจากผ่านการปรับ เรียบ (Smoothing Filter) (c) ผลต่างระหว่างภาพทั้งสอง . . . . .	54
4.7	การปรับปรุงภาพโดยการดำเนินการเชิงพื้นที่กับจุดภาพรอบข้าง . . . . .	54
4.8	การปรับเรียบเชิงเส้น (a) ภาพต้นฉบับ (b) ตัวกรองขนาด 3x3 ค่าถ่วงน้ำ หนักเท่ากัน (c) ตัวกรองขนาด 5x5 ค่าถ่วงน้ำหนักเท่ากัน (d) ตัวกรอง ขนาด 7x7 ค่าถ่วงน้ำหนักเท่ากัน (e) ตัวกรองขนาด 9x9 ค่าถ่วงน้ำหนักเท่า กัน (f) ตัวกรองแบบวงกลมขนาด 5x5 รัศมี 2.5 ดังสมการ 4.17 . . . . .	55
4.9	การปรับเรียบแบบไม่เชิงเส้นสำหรับหน้าต่างขนาด 3x3 . . . . .	56
4.10	การปรับปรุงภาพโดยตัวกรองความถี่สูงด้วยหน้าต่างตัวกรองขนาด 3x3 . . .	57
4.11	การปรับปรุงภาพ Lena ด้วยตัวกรองความถี่สูงในสมการ 4.21 . . . . .	58

5.1	สัญญาณรบกวนบนภาพ (a) ภาพต้นฉบับ (b) เพิ่มสัญญาณรบกวนแบบ Salt-and-Pepper และ (c) ภาพมัวจากตัวกรองในสมการ 5.1 . . . . .	71
5.2	ตัวแบบของการสีกกร่อนของภาพ . . . . .	72
5.3	การประยุกต์ใช้ PCA ในการกู้คืนภาพแบบไม่รู้ภาพต้นฉบับ . . . . .	72
5.4	การประยุกต์ใช้ PCA ในการกู้คืนภาพแบบไม่รู้ภาพต้นฉบับ . . . . .	73
6.1	วัตถุนบนภาพพื้นหลัง . . . . .	76
6.2	สุนัขลายจุดบนภาพพื้นหลัง . . . . .	77
6.3	ขอบของวัตถุในภาพ . . . . .	78
6.4	การหาความไม่ต่อเนื่องบนภาพ โดยการดำเนินการกับกลุ่มจุดภาพ เมื่อ $f'_5 =  R  = h_1f_1 + h_2f_2 + h_3f_3 + h_4f_4 + h_5f_5 + h_6f_6 + h_7f_7 + h_8f_8 + h_9f_9$ . . . . .	79
6.5	การใช้ตัวกรองกับภาพนำเข้า และ การเลือกภาพผลลัพธ์ . . . . .	81
6.6	การใช้อนุพันธ์กับภาพ . . . . .	82
6.7	การหา Gradient บนภาพ (a) ภาพต้นฉบับ, (b) ผลลัพธ์จาก Gradient ในแนวแกน $y$ , $ G_y $ , (c) ผลลัพธ์จาก Gradient ในแนวแกน $x$ , $ G_x $ , (d) ผลลัพธ์จริง $M(x, y) =  G_x  +  G_y $ . . . . .	83
6.8	การหาวัตถุและขอบของวัตถุ ด้วยการกำหนดค่า Threshold ที่แตกต่างกัน . . . . .	85
6.9	การหาวัตถุเมื่อภาพมีฮิสโตแกรมเป็น Uniform . . . . .	86
6.10	การหาวัตถุเมื่อภาพมีฮิสโตแกรมหลายระดับ (Multi-Level histogram) . . . . .	87
6.11	การแบ่งภาพออกเป็นภาพย่อย เพื่อให้ภาพย่อยเหล่านั้นมีฮิสโตแกรมเพียง 2 กลุ่ม . . . . .	88
6.12	การแบ่งภาพออกเป็นพื้นที่ย่อย . . . . .	89
6.13	Splitting procedure . . . . .	90
6.14	Merging procedure . . . . .	91
7.1	การเข้ารหัสและการบีบอัดข้อมูลภาพ . . . . .	94
7.2	Huffman Encoding และ กำหนดค่าบิตในแต่ละกิ่งของต้นไม้ . . . . .	95

# บทที่ 1

## บทนำ

### 1.1 บทนำ

ในอดีตเมื่อประมาณ 20-30 ปีก่อนหน้านี้ การประมวลผลข้อมูลด้วยระบบคอมพิวเตอร์ ไม่ว่าจะเป็น การประมวลผลแฟ้มข้อมูล (File Processing) การประมวลผลฐานข้อมูล (Database Processing) หรือ การประมวลคลังข้อมูล (Data-Warehouse Processing) จะเน้นการประมวลผลข้อมูลชนิดข้อความเป็นหลัก (Text Processing) กล่าวคือ ข้อมูลที่ใช้ในการประมวลผล จะอยู่ในรูปของ ASCII file format และ แฟ้มข้อมูลจะมีรูปแบบการจัดการเป็นเขตข้อมูล (Fields) และ ระเบียบ (Records) เป็นต้น แต่ในปัจจุบัน เป็นที่ทราบกันทั่วไปว่า ความก้าวหน้าทางด้านวิทยาการคอมพิวเตอร์ และ เทคโนโลยีสารสนเทศ ที่เกี่ยวกับการประมวลผลข้อมูลภาพกำลังเข้ามามีบทบาทในชีวิตประจำวันมากยิ่งขึ้น เช่น คนส่วนมากมีการใช้กล้องถ่ายรูปชนิดฟิล์มลดลง แต่หันมาใช้กล้องดิจิทัลมากยิ่งขึ้น หรือ โทรศัพท์มือถือจากเดิมที่เคยใช้โทรศัพท์เพียงอย่างเดียว ก็ได้รับการพัฒนา ให้มีความสามารถในการส่งข้อความ (SMS) การส่งภาพเคลื่อนไหว (MMS) ตลอดจน การถ่ายภาพดิจิทัล ได้ เป็นต้น ดังนั้น จึงความจำเป็นอย่างยิ่งที่ผู้เรียนควรจะต้องทราบถึง ศาสตร์ความรู้เกี่ยวกับการประมวลผลภาพ (Digital Image Processing)

ในบทนี้ จะกล่าวถึง ความหมาย และ ความสำคัญของการประมวลผลภาพ ความสว่างของข้อมูลภาพ การแทนค่าข้อมูลภาพ ความหมายของรายละเอียดของภาพ การประมวลผลภาพ และ ตัวดำเนินการในการประมวลผลภาพ ตลอดจนเมตริกซ์การแปลงเบื้องต้น สำหรับการประมวลผลภาพพอสั่งเขป เพื่อให้ผู้อ่านได้เกิดความคุ้นเคย และเข้าใจหลักการเบื้องต้นของการประมวลผลภาพ ซึ่งผู้เขียนเองจะพยายามเรียบเรียงอย่างค่อยเป็นค่อยไป เพื่อให้ผู้อ่านได้เข้าใจมากที่สุด ทั้งนี้ ปกติแล้วสำหรับวิชาการประมวลผลภาพจะมีตำราให้ศึกษาจำนวนมาก แต่ส่วนใหญ่แล้วจะเป็นภาษาอังกฤษ ซึ่งอาจจะก่อให้เกิดความสับสนแก่ผู้อ่านบ้าง ดังนั้น ประสบการณ์ด้านการประมวลผลภาพของผู้เขียนที่มีมากกว่า 10 ปี คงพอสามารถช่วยผู้อ่านได้เข้าใจเบื้องต้นได้บ้าง

## 1.2 ภาพและรูป (Images and Pictures)

จากที่ได้กล่าวไปในหัวข้อก่อนหน้านี้ ทำให้ทราบว่ามนุษย์เรานั้น กำลังให้ความสนใจกับการสร้างมโนภาพ (Visualization) ที่มนุษย์มองเห็นและสามารถสัมผัสได้ ซึ่งสิ่งต่างๆ เหล่านี้กระจายอยู่รอบตัวเรา การพิจารณาสิ่งที่อยู่รอบข้างตัวเรานั้น ไม่ใช่การพิจารณาเพียงเพื่อแบ่งแยกหรือจัดกลุ่มให้กับสิ่งที่เรามองเห็นเท่านั้น แต่เรายังต้องมองไปให้ถึงการพิจารณาถึงความแตกต่างของสิ่งต่างๆ ที่มองเห็นนั้นด้วย

มนุษย์เรานั้นมีความสามารถและมีทักษะพื้นฐานมาแต่กำเนิดแล้วเกี่ยวกับการมองเห็น ยกตัวอย่างเช่น เราสามารถบ่งบอกใบหน้ามนุษย์ที่เรามองเห็น เราสามารถแยกแยะความแตกต่างของสี เราสามารถประมวลผลสิ่งต่างๆ ที่เรามองเห็นได้อย่างรวดเร็ว

แต่อย่างไรก็ตาม วัตถุ หรือ สิ่งต่างๆ (Object) ในโลกนั้นล้วนแต่มีการเปลี่ยนแปลงตัวเองอยู่ตลอดเวลา เช่น

- การเปลี่ยนแปลงเกี่ยวกับการมองเห็นซึ่งขึ้นกับ เวลา (กลางวัน หรือ กลางคืน) ความสว่างของแสงอาทิตย์ (มีเมฆ หรือ ฟ้าใส) หรือ การตกกระทบของเงาบนวัตถุนั้นๆ
- การเปลี่ยนแปลงจากสภาพแวดล้อมตามกาลเวลาที่เปลี่ยนไป โดยการศึกษาในหัวข้อนี้ต้องใช้ช่วงระยะเวลาที่ค่อนข้างนานพอสมควร

ในหนังสือเล่มนี้เราจะศึกษาเฉพาะภาพเดี่ยว (Single Image) ซึ่งเป็นรูป (Picture) ที่ใช้เป็นตัวแทนของบางสิ่งบางอย่าง เช่น รูปบุคคล รูปสัตว์ รูปสิ่งแวดล้อมต่างๆ หรือรูปทางด้านการแพทย์ เท่านั้น

## 1.3 ความหมายการประมวลผลข้อมูลภาพ

**นิยาม** การประมวลผลข้อมูลภาพ หรือ การประมวลผลภาพ (Image Processings) หมายถึง การจัดการและการวิเคราะห์สารสนเทศของภาพโดยใช้คอมพิวเตอร์ โดยมีวัตถุประสงค์เพื่อ

1. ปรับปรุงหรือเพิ่มสารสนเทศของรูปเพื่อการแปลความหมายของมนุษย์ที่ดีขึ้น
2. ทำให้รูปนั้นๆ เปลี่ยนแปลงในทางที่ดีขึ้นเพื่อให้เหมาะสมกับการประมวลผล หรือ การรับรู้ด้วยเครื่องจักรอย่างอัตโนมัติ (Autonomous machine perception)

ตัวอย่างขั้นตอนวิธีของการจัดการและการวิเคราะห์สารสนเทศของภาพ เช่น การแปลงภาพ (Image Transformation) การนิยามภาพ (Image Description) การกรองภาพ (Image Filters) การคืนคืนภาพ (Image Restoration) การปรับปรุงคุณภาพของภาพ (Image

Enhancement) การแบ่งภาพและการหาขอบวัตถุในภาพ (Image Segmentation and Edge Detection) และ การบีบอัดข้อมูลภาพ (Image Compression) เป็นต้น

## 1.4 ทำไมจึงต้องประมวลผลภาพ

จากที่ได้กล่าวมาข้างต้น การประมวลผลภาพเชิงตัวเลข หรือเรียกสั้นๆ ว่า การประมวลผลภาพ (Digital Image Processing or Image Processing) คือ การประมวลผลของคอมพิวเตอร์ เพื่อเปลี่ยนแปลงธรรมชาติของภาพเชิงตัวเลข (Digital Image) ดังนั้น การประมวลผลภาพถูกพัฒนาขึ้นมาเพื่อตอบปัญหาหลักๆ เช่น

- การทำดิจิทัลไจซ์ และ การเข้ารหัสข้อมูลภาพเพื่อการส่ง การพิมพ์ และ การจัดเก็บอย่างมีประสิทธิภาพ ปัญหานี้ เนื่องจากรูปต้นฉบับ (Original picture) นั้นเป็นข้อมูลเชิงอนาล็อก (Analog picture) ซึ่งเป็นข้อมูลที่มีความต่อเนื่อง ยังไม่สามารถประมวลผลได้ด้วยคอมพิวเตอร์ จึงจำเป็นต้องใช้การทำดิจิทัลไจซ์เพื่อทำการสุ่มข้อมูล และ แปลงข้อมูลดังกล่าวนี้ ให้เป็นข้อมูลเชิงตัวเลข (Discrete or Digital image) เพื่อการประมวลผลสำหรับเครื่องคอมพิวเตอร์ต่อไป
- การปรับปรุงคุณภาพของภาพ เช่น การเพิ่มความคมชัดให้กับขอบของวัตถุในภาพทำให้ภาพดูสว่างมากขึ้น ดังตัวอย่างในภาพที่ 1.1 โดยภาพทางซ้ายมือ คือ ภาพต้นฉบับ และเมื่อผ่านขั้นตอนการปรับปรุงภาพโดยการเพิ่มความสว่าง ทำให้ภาพมีความคมชัดมากขึ้น
- การกำจัดสัญญาณรบกวน จากตัวอย่างในภาพที่ 1.2 พบว่า ภาพด้านซ้ายมือเป็นภาพต้นฉบับที่มีสัญญาณรบกวนแบบเกลือและพริกไทย (Salt-and-Pepper Noise) สำหรับรายละเอียดจะกล่าวอีกครั้งในบทหลังๆ เมื่อนำภาพดังกล่าวไปผ่านกระบวนการกำจัดสัญญาณรบกวน ทำให้ได้คืนภาพที่มีความสมบูรณ์เกือบเหมือนจริงดังภาพขวามือ เป็นต้น ซึ่งภาพที่คืนมาได้นั้น จะทำให้การแปลความต่อสารสนเทศที่อยู่บนข้อมูลภาพในปัญหาต่างๆ ทำได้ง่ายขึ้น
- การแบ่งภาพ และ การนิยามภาพ ถือเป็นขั้นตอนสำคัญสำหรับระบบการเรียนรู้ และ มองเห็นด้วยเครื่องจักร (Machine Learning and Vision) เพราะ ขั้นตอนวิธีนี้ จะช่วยให้คอมพิวเตอร์หรือเครื่องจักรสามารถประมวลผลกับส่วนต่างๆ ที่แบ่งออกมาได้ง่ายขึ้น



(ก) ภาพต้นฉบับ

(ข) ภาพหลังการปรับความสว่าง

รูปที่ 1.1: ตัวอย่างข้อมูลภาพที่จัดเก็บด้วยรายละเอียดที่ต่างกัน

## 1.5 ข้อมูลภาพคืออะไร (What is an Image?)

**นิยาม** ภาพโมนอโครม (Monochrome Image) คือ ค่าความเข้มของแสงในสองมิติ ซึ่งกำหนดโดยฟังก์ชัน  $f(x, y)$  เมื่อ  $x$  และ  $y$  เป็นค่าตำแหน่งเชิงพื้นที่ (Spatial coordinates) และ  $f()$  ณ ตำแหน่ง  $x$  และ  $y$  คือค่าความเข้ม หรือ ความสว่างของแสง

ในกรณีที่ เป็นภาพหลายสี (Multicolour Image) ค่าของ  $f(x, y)$  จะเป็นเวกเตอร์ซึ่งจะบอกถึงความสว่างของภาพ ณ ตำแหน่ง  $(x, y)$  เช่น  $[R, G, B]$  เมื่อ  $R, G,$  และ  $B$  คือ ค่าความสว่างของจุดภาพของสีแดง สีเขียว และ สีน้ำเงิน ตามลำดับ

**นิยาม** ข้อมูลภาพเชิงตัวเลข (Digital Image),  $f(x, y)$  คือ ข้อมูลภาพซึ่งไม่ต่อเนื่อง (Discrete) ทั้งในด้านค่าตำแหน่งเชิงพื้นที่ และ ความสว่างของแสงในตำแหน่งนั้นๆ

ในทางด้านคอมพิวเตอร์เราจะแทนข้อมูลภาพ  $f(x, y)$  ด้วยแถวลำดับสองมิติชนิดจำนวนเต็ม (2D Integer Array) และ ค่าความสว่างของแต่ละจุด  $(x, y)$  เราเรียกว่า ค่าระดับเทา หรือ ค่าความเข้มของภาพ

สำหรับสมาชิกในแต่ละจุด  $(x, y)$  ของแถวลำดับ เราเรียกว่า จุดภาพ (pixel หรือ pel) ซึ่งย่อมาจาก Picture Element ขนาดของข้อมูลภาพจะกำหนดโดย  $M \times N$  ดังนั้น เราจึงสามารถเขียนความสัมพันธ์ของข้อมูลภาพเชิงตัวเลข ได้ดังนี้



(ก) ภาพต้นฉบับ



(ข) ภาพหลังการกำจัดสัญญาณรบกวน

รูปที่ 1.2: การกำจัดสัญญาณรบกวน

$$f(x, y) = \begin{bmatrix} f(0, 0) & f(0, 1) & \cdots & f(0, N) \\ f(1, 0) & f(1, 1) & \cdots & f(1, N) \\ \vdots & \vdots & \ddots & \vdots \\ f(M, 0) & f(M, 1) & \cdots & f(M, N) \end{bmatrix} \quad (1.1)$$

เมื่อ  $2 \leq f(x, y) \leq G$  และ  $G$  เป็นเลขจำนวนเต็มที่อยู่ในรูปกำลังของสอง ความเข้มของแสง  $f(x, y)$  ณ ตำแหน่ง  $(x, y)$  จะขึ้นกับจำนวนของบิต (Bits) ที่ใช้แทนค่าในแต่ละภาพ ยกตัวอย่าง เช่น

- ถ้าใช้ 1 บิต ในการแทนแต่ละจุดภาพ  $2^1 = 2$  จะเรียกว่า ภาพสองระดับ (Binary image)
- ถ้าใช้ 2 บิต ในการแทนแต่ละจุดภาพ  $2^2 = 4$  จะเรียกว่า ภาพระดับเทาสี่ระดับ (Four-levelled grey scale image)
- ถ้าใช้ 3 บิต ในการแทนแต่ละจุดภาพ  $2^3 = 8$  จะเรียกว่า ภาพระดับเทาแปดระดับ (Eight-levelled grey scale image)
- ถ้าใช้ 4 บิต ในการแทนแต่ละจุดภาพ  $2^4 = 16$  จะเรียกว่า ภาพระดับเทาสิบหกระดับ (Sixteen-levelled grey scale image)

- ถ้าใช้ 8 บิต ในการแทนแต่ละจุดภาพ  $2^8 = 256$  จะเรียกว่า ภาพระดับเทาสองร้อยห้าสิบหกระดับ (Grey scale image)

ภาพที่ 1.3 แสดงความสว่างของภาพที่จัดเก็บในหลายระดับและใช้จำนวนบิตต่อจุดภาพไม่เท่ากัน

### 1.5.1 ขนาดของข้อมูลภาพ (Image size)

เพื่อให้ง่ายต่อการวิจัยด้านการประมวลผล จะพบว่าขนาดของข้อมูลภาพโดยทั่วไป จะแทนในรูปของ 512x512, 256x256, หรือ 64x64 เป็นต้น ทั้งนี้ เนื่องจากตัวเลขดังกล่าวข้างต้นนั้น จัดอยู่ในรูปกำลังของสองทำให้การโปรแกรมสะดวกมากยิ่งขึ้น แต่ถ้าจะนำไปใช้งานจริงขนาดของข้อมูลภาพก็ไม่จำเป็นต้องอยู่ในรูปกำลังของสอง

จำนวนของบิต  $b$  ที่ต้องการสำหรับจัดเก็บข้อมูลภาพขนาด  $N \times N$  สำหรับภาพ  $2^m$  ระดับคือ

$$b = N * N * m \quad (1.2)$$

ยกตัวอย่าง เช่น ข้อมูลภาพระดับเทา 256 ( $m = 8$ ) ระดับ ขนาด 512x512 จุดภาพ ถ้าเป็นภาพสองระดับ ขนาดของแฟ้มข้อมูลโดยประมาณ คือ

$$\begin{aligned} 512 * 512 * 1 &= 262144 \text{ bits} \\ &= 32768 \text{ bytes} \\ &= 32.768 \text{ Kb} \\ &\approx 0.033 \text{ Mb} \end{aligned}$$

(ในที่นี้ ใช้การคำนวณโดยประมาณ  $1\text{KB} \approx 1000 \text{ bytes}$  และ  $1\text{Mb} \approx 1,000,000 \text{ bytes}$ ) สำหรับภาพระดับเทา 256 ระดับ ขนาดของแฟ้มข้อมูลควรจะเป็น

$$\begin{aligned} 512 * 512 * 1 &= 262144 \text{ bytes} \\ &= 262.144 \text{ Kb} \\ &\approx 0.262 \text{ Mb} \end{aligned}$$

และ ถ้าเป็นภาพสีจริง ซึ่งแต่ละจุดภาพจะต้องใช้ 3 ไบต์ในการแสดงสารสนเทศของสี ซึ่งขนาดของภาพ ควรจะเป็น

$$\begin{aligned} 512 * 512 * 3 &= 786432 \text{ bytes} \\ &= 786.432 \text{ Kb} \\ &\approx 0.786 \text{ Mb} \end{aligned}$$

จากทั้ง 3 ตัวอย่างที่ได้กล่าวมา ก็ถือได้ว่าภาพสีนั้น มีขนาดใหญ่มาก จึงมีงานวิจัยจำนวนมากที่พยายามจะลดขนาดของ  $N$  และ  $m$  โดยต้องไม่ทำให้สารสนเทศของข้อมูลภาพสูญหายไป

### 1.5.2 ความคมชัดของภาพ (Image Resolution)

ความคมชัดของภาพ จะเป็นพารามิเตอร์สำหรับการอธิบายถึงรายละเอียดของภาพที่จะแสดงผล และ จัดเก็บ จะมีการวัดค่าเป็นจำนวนจุดภาพต่อนิ้ว (dot per inch: dpi) ซึ่งจะบอกกว่าในขนาดของภาพ 1 นิ้ว จะแบ่งออกเป็นกี่จุดภาพ เช่น 150 dpi หมายความว่าใน 1 ตารางนิ้ว จะมีทั้งหมด 150 จุดในแนวนอน และ 150 จุดในแนวตั้ง ภาพที่ 1.4 แสดงภาพเดียวกัน แต่ถูกจัดเก็บด้วยรายละเอียดที่ต่างกัน ทำให้ความคมชัดในการแสดงผลแตกต่างกัน นอกจากนี้ ความคมชัดของภาพยังขึ้นกับ จำนวนบิตที่ใช้ในการแทนแต่ละจุดภาพ ซึ่งได้กล่าวมาแล้วในหัวข้อก่อนหน้านี้

### 1.5.3 ความสว่างของภาพ (Brightness)

ในโลกสามมิติ (3D world) ของวัตถุแต่ละชนิด ในแต่ละด้านของวัตถุจะมีความสามารถหรือคุณลักษณะทางกายภาพใน การรับแสง การดูดซับแสง และ การสะท้อนแสง ที่แตกต่างกัน ซึ่งทำให้เรามองเห็นวัตถุต่างๆ ในมุมมองต่างๆ นั้นมีความสว่างของแสงที่แตกต่างกัน ดังนั้น เมื่อเราจะทำการแปลงภาพจากโลกสามมิติหรือสองมิติไปอยู่ในคอมพิวเตอร์ แต่ละจุดภาพซึ่งให้ค่าความสว่างที่แตกต่างกันย่อมมีค่าความเข้มของแสงหรือความสว่างของภาพที่แตกต่างกัน ซึ่ง ค่าความสว่างของภาพ (Brightness) จะถูกแทนอยู่ในรูปของค่าตัวเลขจำนวนเต็มของแต่ละจุดภาพ  $f(x, y)$  (Pixel value) ดังตัวอย่างในภาพที่ 1.1 ซึ่งแสดงไว้แล้วก่อนหน้านี้

## 1.6 การประมวลผลข้อมูลภาพเบื้องต้น

### 1.6.1 ตัวแบบการประมวลผลภาพอย่างง่าย

จากที่ได้กล่าวมาข้างต้น ข้อมูลภาพ  $f(x, y)$  สามารถเขียนเป็นความสัมพันธ์ทางคณิตศาสตร์ใหม่ได้ ดังนี้

$$f(x, y) = i(x, y)r(x, y) \quad (1.3)$$

เมื่อ

$i(x, y)$  คือ ค่า Illumination component โดย  $0 \leq i(x, y) \leq \infty$

$r(x, y)$  คือ ค่า reflectance component โดย  $0 \leq r(x, y) \leq 1$

ในทางทฤษฎีแล้วจะพบว่า  $0 \leq f(x, y) \leq \infty$  กล่าวคือ ค่าความสว่างของภาพในแต่ละจุดภาพมีค่าตั้งแต่ศูนย์ ถึง  $+\infty$  หรือ กล่าวอีกอย่างหนึ่งว่า ความสว่างของภาพสามารถเป็นไปได้ไม่มีขอบเขตจำกัด นั้นเอง แต่ในทางปฏิบัติ หรือ ด้านการโปรแกรมด้วยคอมพิวเตอร์แล้ว เราจะกำหนดค่าความสว่างของแต่ละจุดภาพไว้เป็น  $L_{min} \leq f(x, y) \leq L_{max}$  นั่นคือ เราจะต้องการสุ่มอย่างไม่ต่อเนื่อง (Discrete Sampling) เพื่อทำการแบ่งภาพ  $0 \leq f(x, y) \leq \infty$  ระดับ ให้ไปอยู่ในช่วง  $L_{min} \leq f(x, y) \leq L_{max}$  ระดับ

### 1.6.2 การคอนโทซ์เซชัน (Image Quantization)

จากข้อมูลที่ได้จากการสุ่มอย่างไม่ต่อเนื่อง เราจะได้ข้อมูลภาพอยู่ในช่วง  $L_{min} \leq f(x, y) \leq L_{max}$  หรือ  $[L_{min}, L_{max}]$  จากนั้นนำข้อมูลเข้าสู่ขั้นตอนการคอนโทซ์เซชัน เช่น ถ้าต้องการแปลงจาก  $[L_{min}, L_{max}] \Rightarrow [0, 255]$  หรือ  $[L_{min}, L_{max}]$  ระดับ ให้เป็นภาพ 256 ระดับ สามารถทำได้ตามความสัมพันธ์ ดังนี้

$$[L_{min}, L_{max}] \Rightarrow [(L_{min}, a_1), (a_1, a_2), \dots, (a_G, L_{max})] \Rightarrow [0, 1, \dots, 255] \quad (1.4)$$

เมื่อ  $G$  คือ จำนวนระดับของการคอนโทซ์เซชัน ซึ่งในที่นี้  $G = 256$

### 1.6.3 การดำเนินการในการประมวลผลข้อมูลภาพ

หลักการที่ใช้ในการประมวลผลข้อมูลภาพ คือ การแปลงข้อมูลภาพ (Image Transformation) ซึ่งเราต้องการตัวดำเนินการ (Operators) และ ข้อมูลภาพต้นฉบับเป็นตัวถูกดำเนินการ

(Operand) หรือ ข้อมูลนำเข้า (Input) เพื่อสร้างข้อมูลภาพใหม่ (Output) ซึ่งเป็นคำตอบของการแปลงดังกล่าว และเพื่อให้ง่ายและสะดวกต่อการเข้าใจ ในหนังสือเล่มนี้ จะเน้นการอธิบายเฉพาะการแปลงแบบเชิงเส้น (Linear Transformation) ด้วยตัวดำเนินการเชิงเส้น (Linear Operators) เท่านั้น

#### 1.6.4 ตัวดำเนินการเชิงเส้น

พิจารณา  $O$  เป็นตัวดำเนินการสำหรับการแปลงข้อมูลภาพ และ กำหนดให้  $f$  เป็นข้อมูลภาพ,  $O(f)$  เป็นผลลัพธ์ของการดำเนินการ  $O$  บน  $f$  เราจะเรียก  $O$  ว่าเป็นเชิงเส้น ถ้า :

$$O[af + bg] = aO[f] + bO[g] \quad (1.5)$$

เมื่อ  $f$  และ  $g$  เป็นข้อมูลภาพ และ  $a$  และ  $b$  เป็นสเกลาร์

ตัวดำเนินการที่กล่าวมาข้างต้น จะถูกนิยามในรูปของตัวดำเนินการสำหรับจุดภาพ (Point Spread Function) กล่าวคือ ผลลัพธ์ที่เราจะได้ออกมานั้นจะได้จากการใช้ตัวดำเนินการกับจุดภาพที่กำลังสนใจจุดใดๆ (point source:  $p$ ) ดังสมการ

$$O[p] = \text{point spread function} \quad (1.6)$$

หรือ

$$O[\delta(x - \alpha, y - \beta)] = h(x, \alpha, y, \beta) \quad (1.7)$$

เมื่อ  $\delta(x - \alpha, y - \beta)$  คือ จุดภาพใดๆ ซึ่งจุดศูนย์กลางอยู่ที่  $(\alpha, \beta)$  จากสมการ (1.7) สมมติ  $O()$  เป็นเชิงเส้น เราจะได้ว่า

$$O[a\delta(x - \alpha, y - \beta)] = ah(x, \alpha, y, \beta) \quad (1.8)$$

เมื่อ  $a$  คือ จำนวนเท่าของความสว่าง กล่าวคือ เราสามารถเพิ่มความสว่างให้กับจุดภาพ (Brighter) เมื่อมาถึงจุดนี้ เราจะพบว่าข้อมูลภาพ (Image) ก็คือ การรวมกันของจุดภาพ (point sources) หรือ pixels ซึ่งแต่ละจุดต่างก็จะมีค่าความสว่างประจำตัว ซึ่งกล่าวได้อีกอย่างว่าข้อมูลภาพ  $g(x, y)$  ก็คือ ผลรวมของจุดภาพต้นฉบับ ของตัวดำเนินการที่ดำเนินการ  $h(x, \alpha, y, \beta)$  กับจุด  $f(x, y)$  ซึ่งสามารถเขียนใหม่ได้ ดังนี้

$$g(\alpha, \beta) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} N - 1 f(x, y) h(x, \alpha, y, \beta) \quad (1.9)$$

เมื่อ  $g(\alpha, \beta)$  เป็นภาพผลลัพธ์,  $f(x, y)$  เป็นภาพนำเข้า (ต้นฉบับ) และ ขนาดของภาพเท่ากับ  $N \times N$

### 1.6.5 ตัวดำเนินการแบบกองซ้อน: Stacking operator

นิยาม ตัวดำเนินการแบบกองซ้อน คือ การดำเนินการที่ยอมให้การดำเนินการกับแถวลำดับของภาพ  $N \times N$  ทำได้เช่นเดียวกับการดำเนินการบนเวกเตอร์ขนาด  $N^2 \times 1$  ในทำนองเดียวกัน การดำเนินการกับเวกเตอร์  $N^2 \times 1$  ก็สามารถทำได้เช่นเดียวกันกับเมทริกซ์ของภาพ  $N \times N$

ตัวอย่าง กำหนดให้ข้อมูลภาพ  $f$  มีขนาด  $3 \times 3$  จงแสดงตัวอย่างการคำนวณให้อยู่ในรูปของเวกเตอร์

กำหนดให้

$$f = \begin{pmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{pmatrix} \quad (1.10)$$

เรา นิยาม เวกเตอร์  $V_n$  และ เมทริกซ์  $N_n$  สำหรับ  $n = 1, 2, 3$  ดังนี้

$$V_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad V_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad V_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad (1.11)$$

$$N_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad N_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad N_3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (1.12)$$

เราสามารถ คำนวณภาพผลลัพธ์ ได้จาก

$$\begin{aligned} \mathbf{f} &= \sum_{n=1}^N N_n f V_n \\ &= N_1 f V_1 + N_2 f V_2 + N_3 f V_3 \end{aligned} \quad (1.13)$$

เราสามารถ แยกการคำนวณแต่ละส่วนได้ดังนี้

$$\begin{aligned}
N_1 f V_1 &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} f_{11} \\ f_{21} \\ f_{31} \end{pmatrix} = \begin{pmatrix} f_{11} \\ f_{21} \\ f_{31} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{1.14}
\end{aligned}$$

ทำนองเดียวกัน

$$N_2 f V_2 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ f_{12} \\ f_{22} \\ f_{32} \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad N_3 f V_3 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ f_{13} \\ f_{23} \\ f_{33} \end{pmatrix}, \tag{1.15}$$

## 1.7 สรุป

ในบทนี้ได้กล่าวถึง วัตถุประสงค์ และ ความจำเป็นของการประมวลผลภาพ ความหมายของข้อมูลภาพ และ การดำเนินการเบื้องต้นกับข้อมูลภาพ ซึ่งจากที่ได้บรรยายมาตลอดทั้งบทนี้ คงพอทำให้ผู้อ่านพอเข้าใจพื้นฐานทางคณิตศาสตร์ ซึ่งจำเป็นอย่างมากสำหรับงานทางด้าน การประมวลผลภาพ แต่ก็คงไม่ยากเกินไป และ สำหรับเนื้อหาจริงๆ ของการประมวลผลภาพนั้น จะกล่าวในบทถัดไป

### แบบฝึกหัด

- 1.



(ก) ภาพปกติขนาด 256 ระดับเทา



(ข) ภาพสองระดับ



(ค) ภาพสี่ระดับเทา



(ง) ภาพแปดระดับเทา



(จ) ภาพสิบหกระดับเทา



(ฉ) ภาพสามสิบสองระดับเทา



(ช) ภาพหกสิบสี่ระดับเทา



(ซ) ภาพหนึ่งร้อยสิบแปดระดับ

รูปที่ 1.3: ภาพระดับเทา



(ก) ภาพปกติขนาด 256x256



(ข) ภาพขนาดร้อยละ 50 ของรูปปกติ



(ค) ภาพขนาดร้อยละ 25 ของรูปปกติ



(ง) ภาพขนาดร้อยละ 12.5 ของรูปปกติ

รูปที่ 1.4: ตัวอย่างข้อมูลภาพที่จัดเก็บด้วยรายละเอียดที่ต่างกัน

## บทที่ 2

### Image Transformations

#### เนื้อหาสาระ

1. การคูณเวกเตอร์
2. Unitary transform
3. การประยุกต์ใช้เมตริกซ์  $U$  และ  $V$
4. Fourier Transform

#### วัตถุประสงค์

1. เพื่อศึกษาหลักการพื้นฐานเกี่ยวกับการแปลงภาพเบื้องต้น
2. เพื่อพัฒนาโปรแกรมแปลงภาพ

#### กิจกรรมการเรียนการสอน

1. บรรยาย อภิปรายซักถาม และ ยกตัวอย่าง
2. สรุปเนื้อหารายวิชา

#### การวัดผล

1. ซักถามความเข้าใจหลังการเรียน

## Image Transformations

### 2.1 บทนำ

การแปลงภาพ (Image Transformation) หมายถึง กระบวนการในการเปลี่ยนวิธีการพิจารณา หรือ การวิเคราะห์ภาพ เช่น การแปลงข้อมูลจากโดเมนของเวลาเป็นโดเมนของความถี่ หรือ การเพิ่มการเพิ่มมิติของข้อมูลให้สูงขึ้น ทำให้การวิเคราะห์แยกองค์ประกอบของภาพทำได้สะดวกยิ่งขึ้น เป็นต้น ซึ่งการแปลงดังกล่าว จะยังคงสารสนเทศภายในภาพไว้เช่นเดิม กล่าวคือ ข้อมูลภาพซึ่งถูกแปลงเข้าไปเป็นโดเมนอื่นแล้วจะยังสามารถแปลงกลับมาเป็นข้อมูลเดิมได้ โดยไม่มีการเปลี่ยนแปลงสารสนเทศเมื่อแปลงกลับมา

ดังนั้น ในบทนี้จะกล่าวถึง ขั้นตอน และ วิธีการสำหรับการสร้างเครื่องมือที่สำคัญ และ จำเป็นสำหรับการประมวลผลภาพเชิงเส้น (Linear Image Processing) หรือกล่าวอีกอย่างหนึ่ง ก็คือ หนังสือเล่มนี้จะเน้นกล่าวถึงการแปลงเชิงเส้นบนภาพ หรือ การพยายามอธิบายภาพในรูปของการประมวลผลเชิงเส้นให้มากที่สุด โดยจะเริ่มตั้งแต่ การคูณเวกเตอร์ (Vector Multiplication)

### 2.2 การคูณเวกเตอร์

พิจารณาเวกเตอร์สองเวกเตอร์ขนาด  $N \times 1$

$$\begin{aligned} \mathbf{u}_i^T &= (u_{i1}, u_{i2}, \dots, u_{iN}) \\ \mathbf{v}_j^T &= (v_{j1}, v_{j2}, \dots, v_{jN}) \end{aligned}$$

การคูณกันระหว่างเวกเตอร์ทั้งสอง สามารถนิยามได้ ดังนี้

$$\mathbf{u}_i \mathbf{v}_j^T = \begin{pmatrix} u_{i1} \\ u_{i2} \\ \vdots \\ u_{iN} \end{pmatrix} (v_{j1}, v_{j2}, \dots, v_{jN}) = \begin{pmatrix} u_{i1}v_{j1} & u_{i1}v_{j2} & \cdots & u_{i1}v_{jN} \\ u_{i2}v_{j1} & u_{i2}v_{j2} & \cdots & u_{i2}v_{jN} \\ \vdots & \vdots & \ddots & \vdots \\ u_{iN}v_{j1} & u_{iN}v_{j2} & \cdots & u_{iN}v_{jN} \end{pmatrix} \quad (2.1)$$

นั่นคือ ผลคูณของสองเวกเตอร์ คือ เมตริกซ์ขนาด  $N \times N$  ซึ่งก็คือ ข้อมูลภาพซึ่งได้กล่าวไว้ในบทที่ 1 นั้นเอง ดังนั้น ในกรณีทั่วไปของการประมวลผลภาพเชิงเส้นบนภาพ  $f$  ก็จะสามารถเขียนได้ ดังนี้

$$g = h_c^T f h_r \quad (2.2)$$

นอกจากนี้ ยังสามารถหาอินเวอร์สของเมตริกซ์  $h_c^T$  และ  $h_r$  เพื่อแก้สมการที่กล่าวมาข้างต้น เพื่อหาค่า  $f$  จากภาพ  $g$  โดยการคูณทั้งสองข้างของสมการด้วย  $(h_c^T)^{-1}$  และ  $(h_r)^{-1}$  ดังนี้

$$(h_c^T)^{-1} g (h_r)^{-1} = (h_c^T)^{-1} h_c^T f h_r (h_r)^{-1} = f \quad (2.3)$$

นั่นคือ ถ้า:

$$f = (h_c^T)^{-1} g (h_r)^{-1} \quad (2.4)$$

แล้ว

$$f = \sum_{i=1}^N \sum_{j=1}^N g_{ij} \mathbf{u}_i \mathbf{v}_j^T \quad (2.5)$$

### 2.3 Unitary transform

ถ้า เมตริกซ์  $h_c$  และ  $h_r$  ซึ่งถูกเลือกมาเป็น *unitary* แล้ว จากสมการ 2.2 จะได้ว่า  $g$  เป็น *unitary transform* ของ  $f$  และ  $f$  เป็น *unitary transform* ของ  $g$

และ เมตริกซ์  $U$  จะเรียกว่า *unitary* ถ้าอินเวอร์สเป็นทรานสโพสของ complex conjugate ดังสมการ

$$U U^{T*} = I \quad (2.6)$$

เมื่อ  $I$  เป็นเมตริกซ์หนึ่งหน่วย (Unit matrix) แต่โดยปกติแล้วเราจะแทนยกกำลัง  $T^*$  ด้วย  $H$  และ ถ้าสมาชิกในเมตริกซ์เป็นจำนวนจริง เราจะแทน *unitary* ด้วยเมตริกซ์ตั้งฉาก (Orthogonal matrix)

ดังนั้น ถ้า  $h_c^T$  และ  $h_r$  ใน(2.2) ต่างเป็น *unitary* แล้วเราสามารถหาอินเวอร์สของ  $g$  ได้ ดังนี้

$$f = h_c g h_r^H \quad (2.7)$$

และ เพื่อให้ง่ายเราจะเขียนแทน  $h_c$  ด้วย  $U$  และ แทน  $h_r$  ด้วย  $V$  ดังนั้น ข้อมูลภาพ  $f$  สามารถเขียนใหม่ได้ ดังนี้

$$f = U g V^H \quad (2.8)$$

จากสมการ (2.6) เราพบว่า เมตริกซ์  $U$  จะเป็นเมตริกซ์ *unitary* ถ้า ผลคูณ (dot product) ของสองแถวใดๆ ต้องเป็นศูนย์ ในขณะที่ค่าผลคูณเวกเตอร์บนตัวมันเองจะต้องเป็นหนึ่ง

## 2.4 การประยุกต์ใช้เมตริกซ์ $U$ และ $V$

การประยุกต์ใช้งานเมตริกซ์  $U$  และ  $V$  มีอย่างแพร่หลาย อย่างหนึ่งซึ่งมีการนำไปใช้กันมาก คือ การนำเมตริกซ์  $U$  และ  $V$  มาช่วยลดจำนวนสมาชิกที่จะใช้ในการแทนข้อมูลภาพ  $f$  ซึ่งเดิมจะต้องใช้ขนาดข้อมูลเท่ากับ  $N^2$  ซึ่งจะทำให้ได้โดยการเลือก เมตริกซ์  $U$  และ  $V$  ที่ จะทำให้การแปลงบน  $g$  เป็นเมตริกซ์ทแยงมุม (Diagonal matrix) ดังนั้น โดยใช้สมการ (2.5) เราสามารถแทนข้อมูลภาพ  $f$  ด้วยสมาชิกเพียง  $N$  ตัวโดยการพิจารณาค่า  $g$  ซึ่งไม่เป็นศูนย์ วิธีนี้เรียกว่า Singular Value Decomposition (SVD) ของข้อมูลภาพ โดย

$$g = U \Lambda^{1/2} V^T \quad (2.9)$$

เมื่อ  $U$  และ  $V$  เป็นเมตริกซ์จัตุรัสขนาด  $N \times r$  และ  $\Lambda^{1/2}$  เป็นเมตริกซ์ทแยงมุมขนาด  $r \times r$

ตัวอย่าง ถ้า  $\Lambda$  เป็นเมตริกซ์  $2 \times 2$  และ  $\Lambda^m$  เป็นกำลังที่  $m$  ของเมตริกซ์ไม่ใช่ศูนย์ (Non-zero matrix) ของเมตริกซ์  $\Lambda$  จงแสดงว่า

$$\Lambda^{-1/2} \Lambda \Lambda^{-1/2} = I \quad \text{and} \quad \Lambda^{-1/2} \Lambda^{-1/2} = I$$

วิธีทำ

$$\begin{aligned}
 \Lambda^{-1/2}\Lambda\Lambda^{-1/2} &= \begin{pmatrix} \lambda_1^{-1/2} & 0 \\ 0 & \lambda_2^{-1/2} \end{pmatrix} \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \begin{pmatrix} \lambda_1^{-1/2} & 0 \\ 0 & \lambda_2^{-1/2} \end{pmatrix} \\
 &= \begin{pmatrix} \lambda_1^{-1/2} & 0 \\ 0 & \lambda_2^{-1/2} \end{pmatrix} \begin{pmatrix} \lambda_1^{1/2} & 0 \\ 0 & \lambda_2^{1/2} \end{pmatrix} \\
 &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}
 \end{aligned}$$

ในทำนองเดียวกัน เราก็จะสามารถแสดงได้ว่า  $\Lambda^{-1/2}\Lambda^{-1/2} = I$

### 2.4.1 Applications: Independent Component Analysis

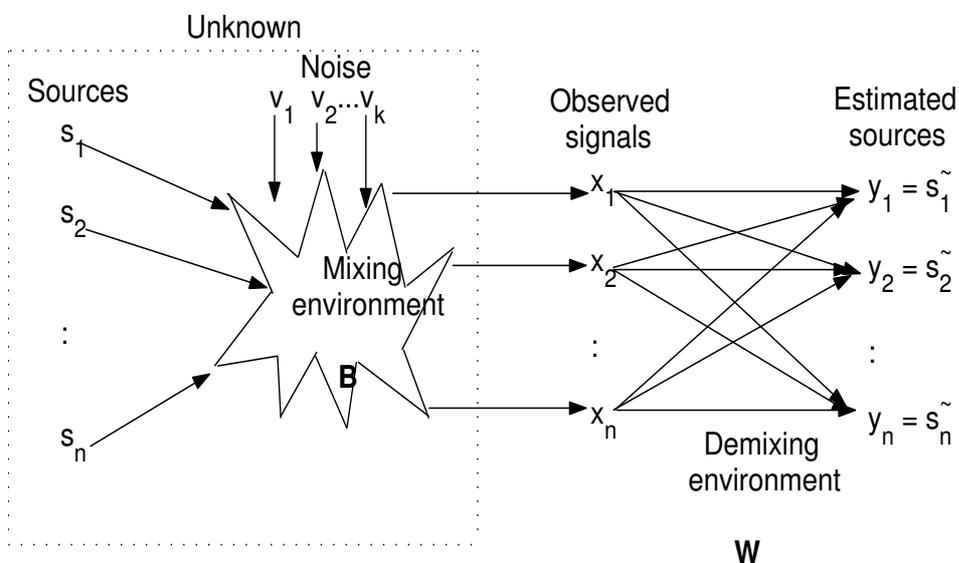
การวิเคราะห์องค์ประกอบอย่างอิสระ (Independent Component Analysis: ICA) เป็นวิธีการที่แพร่หลายมากในการแก้ปัญหา การแยกกลับแหล่งสัญญาณแบบไม่รู้แหล่งที่มา (Blind Source Separation: BSS)

ตัวอย่างที่พบได้ง่ายที่สุดของปัญหา ICA คือ *the cocktail party problem* สมมติในห้องๆ หนึ่ง มีคนจำนวนหนึ่งอยู่ในห้อง และ ทุกคนก็พูดพร้อมๆ กัน ผ่านไมโครโฟนที่เตรียมไว้สำหรับเสียง ณ ตำแหน่งๆ ต่างๆ ในห้อง ในที่นี้ สมมติ มีคนทั้งหมด  $n$  คน และมี  $m$  ไมโครโฟน ดังแสดงในรูป 2.1 (for  $n = m$ ). แต่ละไมโครโฟน  $Mic_j$  จะได้รับสัญญาณเสียงที่บันทึกตามเวลา คือ  $x_j(k)$ , เมื่อ  $1 \leq j \leq m$  และ  $t$  คือ เวลา ซึ่งแต่ละสัญญาณที่บันทึกตามเวลานั้นจะเป็นการดำเนินการเชิงเส้นระหว่างแหล่งสัญญาณ  $s_i(k)$ , ( $1 \leq i \leq n$ ) กับ เมตริกซ์การรวม  $B$ , ดังสมการ

$$x_j(k) = \sum_{i=1}^n b_{ji}s_i(k) \quad (2.10)$$

เมื่อ  $b_{ji}$ ,  $1 \leq j, i \leq n$  เป็นค่าถ่วงน้ำหนักซึ่งจะขึ้นกับระยะห่างระหว่างไมโครโฟนกับผู้พูด Hyvarinen:2000 และ ถ้าแหล่งสัญญาณ  $s_i$  อยู่ใกล้กับตัวรับสัญญาณ  $Mic_j$  จะทำให้เมตริกซ์การรวม  $B$  เป็น *diagonal* ซึ่งจะไม่ถือว่าเป็นปัญหา ICA โดยปกติแล้ว สำหรับปัญหา ICA สมาชิกของเมตริกซ์  $B$  จะเป็น nonsingular และ จะมีการหมุน (permuted) นอกจากนั้น ฟังก์ชันการแจกแจงความน่าจะเป็น (the probability density function) ของ  $s_i(k)$  ก็จะไม่ทราบค่า มีเพียงสมมติฐานเดียวที่จะกำหนดให้กับปัญหา *the cocktail party problem* คือ ทุกแหล่งสัญญาณ  $s_i(k)$  จะอิสระจากกัน (identically and independently distributed (iid))

และ แหล่งสัญญาณ (Source signals) สัญญาณที่ได้รับ (observed signals) และ สัญญาณที่จะถูกแยกกลับ (recovered signals) ต้องมีค่าเฉลี่ยเป็นศูนย์ (zero mean)  $E[s] = E[x] = E[y] = 0$ . รายละเอียดสามารถอ่านได้ใน [1, 9, 10].



รูปที่ 2.1: The cocktail party problem.

เป้าหมายของปัญหา ICA คือ การแยกกลับ หรือ การคืนคืนแหล่งสัญญาณ  $\tilde{s} = y = Wx$  จากสัญญาณที่ได้รับ (observed signal)  $x = Bs$ , เมื่อ แต่ละองค์ประกอบของสัญญาณที่จะถูกแยกกลับ (recovered signals)  $y_i(k)$  มีการแจกแจงเป็นแบบ *iid* ตั้งสมการแสดงความสัมพันธ์

$$\tilde{s} = y = Wx = WBs = B^{-1}Bs = Is = s \quad (2.11)$$

ถ้า  $s_i$  และ  $s_j$  มีการแจกแจงแบบ independently distributed,

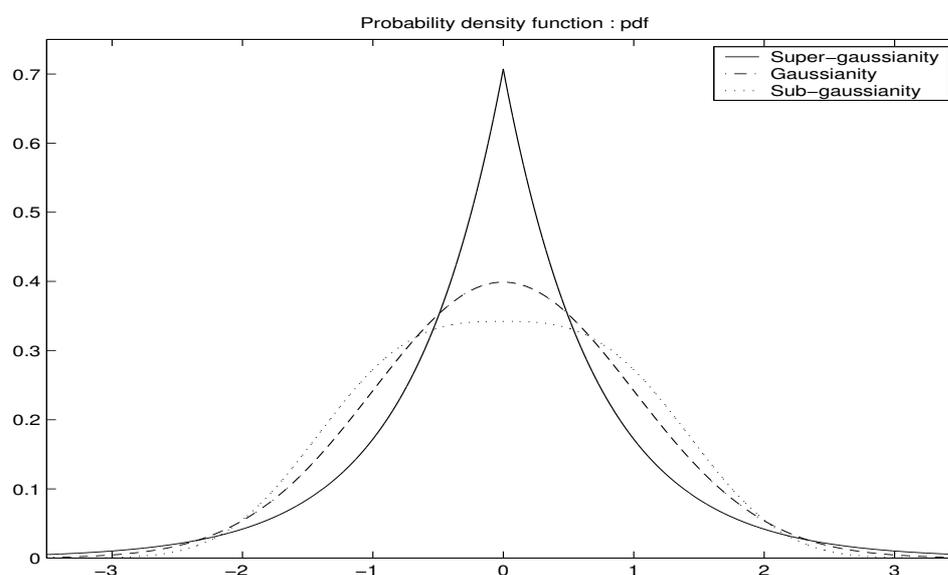
$$E[ss^T] = I. \quad (2.12)$$

ค่า Kurtosis ของ  $s_i$  ซึ่งเป็นค่าซึ่งบอกถึงลักษณะการกระจายตัวของแต่ละแหล่ง สามารถนิยามได้ ดังนี้

$$kurt(s) = \frac{E[s^4]}{(E[s^2])^2} - 3 \quad (2.13)$$

คุณสมบัติของค่า Kurtosis

- ถ้ามีค่าเป็นบวก หมายถึง แหล่งนั้นมีการแจกแจงแบบ Super-Gaussianity
- ถ้ามีค่าเป็นศูนย์ หมายถึง แหล่งนั้นมีการแจกแจงแบบ Gaussianity
- ถ้ามีค่าเป็นลบ หมายถึง แหล่งนั้นมีการแจกแจงแบบ Sub-Gaussianity



รูปที่ 2.2: การแจกแจงความน่าจะเป็นแบบต่างๆ

หลังจาก linear transformation ด้วยเมตริกซ์  $B$ , เราจะได้

$$\mathbf{x} = \mathbf{B}\mathbf{s} \quad (2.14)$$

or

$$\begin{aligned} x_1 &= b_{11}s_1 + b_{12}s_2 + b_{13}s_3 \\ x_2 &= b_{21}s_1 + b_{22}s_2 + b_{23}s_3 \\ x_3 &= b_{31}s_1 + b_{32}s_2 + b_{33}s_3. \end{aligned} \quad (2.15)$$

ดังนั้น ค่า Kurtosis ของ  $k_i$  ก็จะเปลี่ยนไปเป็น

$$kurt(x_i) = b_{i1}^4 kurt(s_1) + b_{i2}^4 kurt(s_2) \quad (2.16)$$

เราจะประยุกต์ใช้ SVD เพื่อทำการลดสหสัมพันธ์ระหว่างองค์ประกอบในสัญญาณที่ได้รับ

$$\tilde{\mathbf{x}} = \mathbf{Z}\mathbf{x} = \mathbf{Z}\mathbf{B}\mathbf{s} \quad (2.17)$$

เมื่อ

$$\mathbf{Z} = \mathbf{D}^{-1/2}\mathbf{V}^T, \quad (2.18)$$

$$\mathbf{D}^{1/2} = \begin{bmatrix} \frac{1}{\sqrt{d_1}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{d_2}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{d_3}} \end{bmatrix}, \quad (2.19)$$

และ

$$\mathbf{V}^T = \begin{bmatrix} v_{11}^{max} & v_{21} & v_{31} \\ v_{12} & v_{22} & v_{32}^{max} \\ v_{13} & v_{23}^{max} & v_{33} \end{bmatrix}. \quad (2.20)$$

สำหรับ ค่าผลคูณที่มีค่าน้อยๆ  $v_{12}, v_{13}, v_{21}, v_{22}, v_{31}$  และ  $v_{33}$  เราสามารถข้ามการพิจารณาได้ ดังนั้น เราสามารถประมาณค่าใหม่ได้ ดังนี้

$$\mathbf{Z} \approx \begin{bmatrix} \frac{v_{11}^{max}}{\sqrt{d_1}} & 0 & 0 \\ 0 & 0 & \frac{v_{32}^{max}}{\sqrt{d_2}} \\ 0 & \frac{v_{23}^{max}}{\sqrt{d_3}} & 0 \end{bmatrix} = \begin{bmatrix} z_{11} & 0 & 0 \\ 0 & 0 & z_{32} \\ 0 & z_{23} & 0 \end{bmatrix}. \quad (2.21)$$

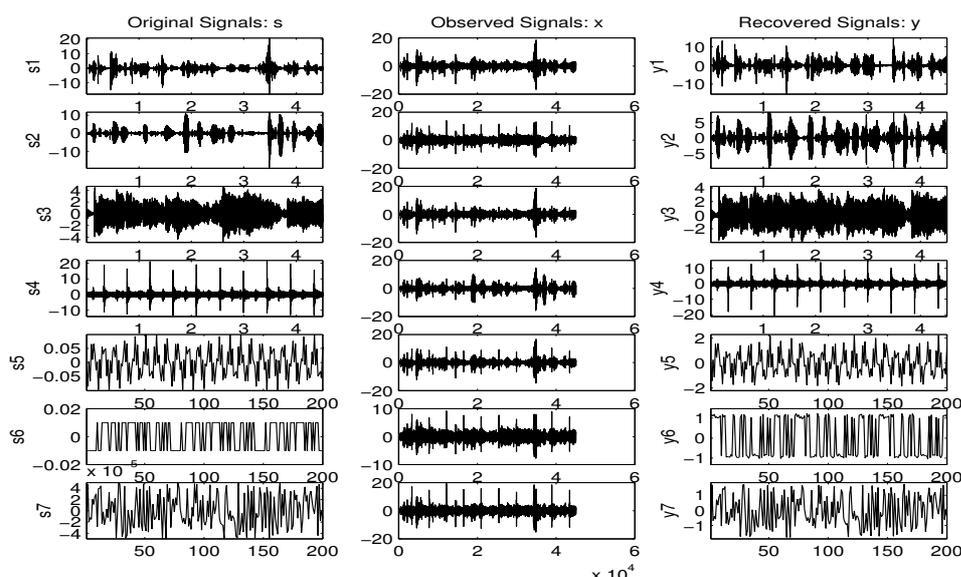
ดังนั้น  $\mathbf{Z}$  ก็จะเป็นเมตริกซ์ที่ตั้งฉาก (orthogonal matrix) ดังนั้น หลังการทำแปลงย้อนกลับ (de-transformation) ด้วย  $\mathbf{Z}$  เราจะได้

$$\begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ \tilde{x}_3 \end{bmatrix} = \begin{bmatrix} z_{11}b_{11}s_1 \\ z_{32}b_{22}s_2 \\ z_{23}b_{33}s_3 \end{bmatrix}. \quad (2.22)$$

และ ค่า Kurtosis ของ  $\tilde{x}_i$  คือ

$$\begin{bmatrix} Kurtosis(\tilde{x}_1) \\ Kurtosis(\tilde{x}_2) \\ Kurtosis(\tilde{x}_3) \end{bmatrix} = \begin{bmatrix} z_{11}^4 b_{11}^4 Kurtosis(s_1) \\ z_{32}^4 b_{22}^4 Kurtosis(s_2) \\ z_{23}^4 b_{33}^4 Kurtosis(s_3) \end{bmatrix}. \quad (2.23)$$

นั่นคือ ค่า Kurtosis จะถูกแปลงกลับมาเป็นค่าเดิมหลังจากกระบวนการแปลงด้วย SVD



รูปที่ 2.3: การวิเคราะห์องค์ประกอบอย่างอิสระสำหรับแหล่งซึ่งมีค่า Kurtosis แบบผสม

## 2.5 Fourier Transform

การแปลงฟูรีเยร์ (Fourier transform) เป็นการแปลงที่ใช้กันแพร่หลายในงานประมวลผลภาพ ทั้งนี้เนื่องจาก การแปลงฟูรีเยร์มีความสัมพันธ์กับทฤษฎีระบบเชิงเส้น (Linear system theory) กล่าวคือ ข้อมูลนำเข้า (Input) จะถูกนำไปผ่านระบบเชิงเส้น (Linear system) เพื่อสร้างคำตอบ (Output) โดยในระบบเชิงเส้นนั้นจะเป็นการทำคอนโวลูชัน หรือ การคูณอย่างง่ายนั่นเอง การแปลงฟูรีเยร์ได้ถูกนำเสนอครั้งแรกโดยนักคณิตศาสตร์ชาวฝรั่งเศส ชื่อ Baptiste Joseph Fourier (ค.ศ. 1786-1830) โดยมีจุดประสงค์เพื่อแก้ปัญหาคณิตศาสตร์เชิงอนุพันธ์ของลาปลาซซ์ (Laplace's differential equation) ซึ่งอธิบายเรื่องการนำพาความร้อน (Conduction of heat

หรือ distribution of temperature) สำหรับข้อมูลซึ่งไม่จำกัด (infinite conduction sheet) โดยการแปลงฟูรีเยร์จะถูกจัดให้อยู่ในรูปอนุกรมของฟังก์ชัน Sine และ Cosine

สำหรับสัญญาณหนึ่งมิติ  $f(t)$  จะสามารถแทนด้วยอนุกรมฟูรีเยร์ของสัญญาณดังกล่าวได้ดังนี้

$$f(t) = \frac{A_0}{2} + \sum_{n=1}^{\infty} A_n \cos(2\pi n f_0 t) + \sum_{n=1}^{\infty} B_n \sin(2\pi n f_0 t) \quad (2.24)$$

สัมประสิทธิ์  $A_n$  และ  $B_n$  สามารถหาได้จากการอินทิเกรตบนคาบพื้นฐาน  $T$  (fundamental period) ของฟังก์ชัน  $f(t)$

$$A_n = \frac{2}{T} \int_0^T f(t) \cos(2\pi n f_0 t) dt, \quad \text{for } n \geq 0 \quad (2.25)$$

และ

$$B_n = \frac{2}{T} \int_0^T f(t) \sin(2\pi n f_0 t) dt, \quad \text{for } n \geq 0 \quad (2.26)$$

เมื่อ  $f_0 = 1/T$  รูปที่ 2.4 แสดงให้เห็นว่าสัญญาณที่เราได้รับ  $f(t)$  ประกอบด้วยสัญญาณ 3 สัญญาณที่มีความถี่พื้นฐาน  $f_0$  ที่ต่างกัน จากหลักการดังกล่าวนี้ทำให้เราสามารถวิเคราะห์หาความถี่พื้นฐานต่างๆ ของสัญญาณได้โดยใช้การแปลงฟูรีเยร์จากโดเมนเชิงพื้นที่เป็นโดเมนเชิงความถี่ (Spatial to Frequency domain)

การแปลงฟูรีเยร์จากโดเมนเชิงพื้นที่เป็นโดเมนเชิงความถี่ เขียนได้ ดังนี้

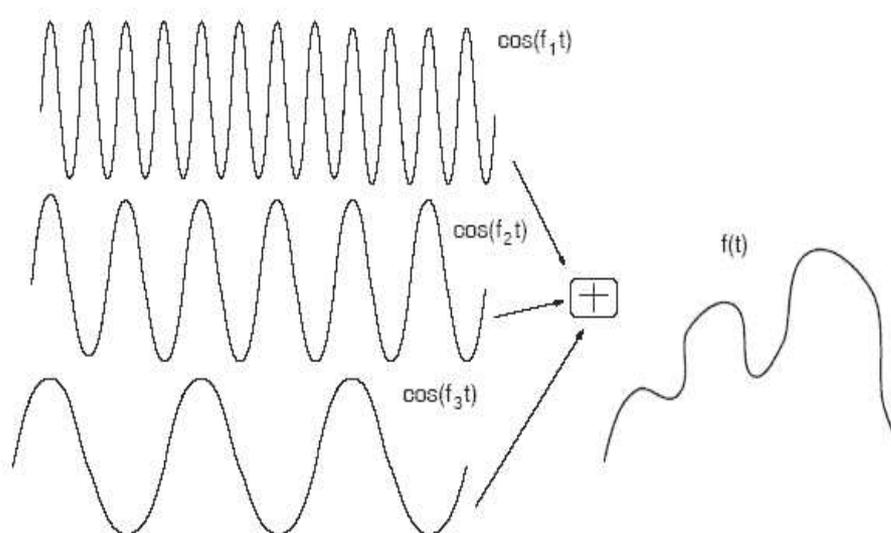
$$F(f(x)) = F(u) = \int_{-\infty}^{\infty} f(x) e^{-j2\pi u x} dx \quad (2.27)$$

และ อินเวอร์สของการแปลงฟูรีเยร์ คือ

$$F^{-1}(f(u)) = F(x) = \int_{-\infty}^{\infty} f(u) e^{j2\pi u x} dx \quad (2.28)$$

### การใช้การแปลงฟูรีเยร์

1. ใช้ลบความถี่ที่ไม่ต้องการของสัญญาณ



รูปที่ 2.4: ความถี่พื้นฐานของสัญญาณ  $f_1$   $f_2$  และ  $f_3$  ที่มาผสมกันเป็นสัญญาณเดียว

2. การดำเนินการในโดเมนของความถี่นั้น ทำได้ง่ายและรวดเร็วกว่าโดเมนเชิงพื้นที่  
ในทางปฏิบัติเราสามารถประมาณการคำนวณได้ ดังนี้

$$e^{\pm j2\pi ux} = \cos(2\pi ux) \pm j \sin(2\pi ux) \quad (2.29)$$

## 2.6 สรุป

ในบทนี้ ได้กล่าวถึงการแปลงข้อมูลภาพอย่างง่าย โดยเน้นที่การแปลงเชิงเส้นเป็นหลัก และอินเวอร์สของการแปลงซึ่งจะทำให้เราสามารถแปลงข้อมูลภาพไป-และ-กลับได้ นอกจากที่กล่าวในบทนี้แล้ว ยังมีการแปลงภาพอีกหลายวิธีแต่จะมีความซับซ้อนมากจึงจะยังไม่กล่าวถึงในบทนี้

## บทที่ 3

### Statistical Description of Images

#### เนื้อหาสาระ

1. ทำไมจึงต้องสถิติ
2. ตัวแปรสุ่ม (Random variable)
3. ความน่าจะเป็นของเหตุการณ์ (Probability of Events)
4. ฟังก์ชันการแจกแจงและฟังก์ชันความหนาแน่นความน่าจะเป็น (Probability Distribution and Density Functions)
5. ค่าคาดหลัง (Expectation)
6. ความเป็นอิสระของสัญญาณ (Independence of Signals)

#### วัตถุประสงค์

1. เพื่ออธิบายความรู้พื้นฐานทางสถิติ และ การนำสถิติมาใช้ในการประมวลผลภาพ
2. เพื่อศึกษาการเขียนโปรแกรมประมวลผลภาพโดยใช้หลักการทางสถิติ

#### กิจกรรมการเรียนการสอน

1. บรรยาย อภิปรายซักถาม และ ยกตัวอย่าง
2. สรุปเนื้อหารายวิชา

## การวัดผล

1. ซักถามความเข้าใจหลังการเรียน

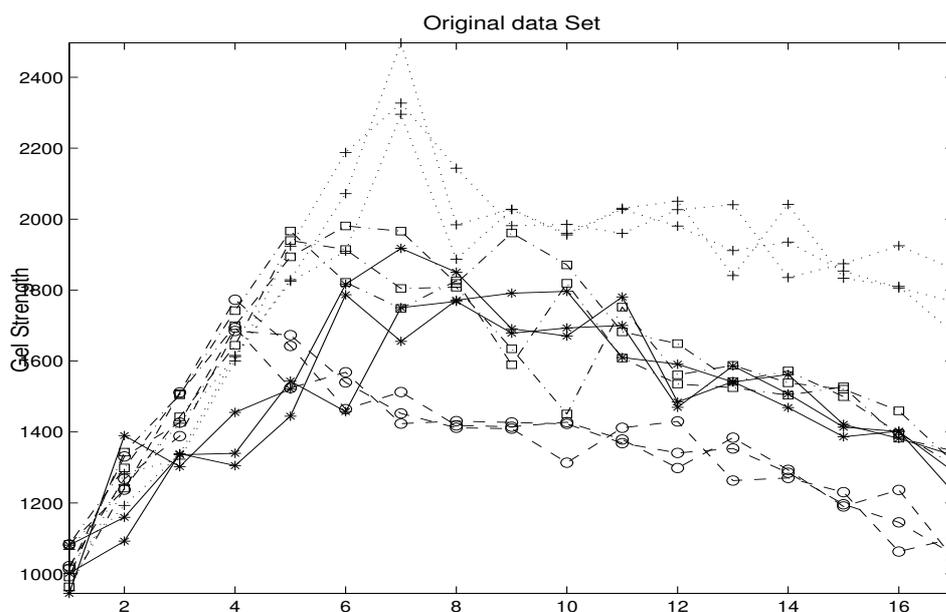
## Statistical Description of Images

### 3.1 บทนำ

ในบทนี้ จะกล่าวถึง สถิติศาสตร์และสถิติพื้นฐานที่จำเป็นสำหรับการนำไปอธิบายข้อมูลภาพ โดยพิจารณาจากงานการประมวลผลภาพเป็นหลัก เช่น ตัวแปรสุ่ม เวกเตอร์สุ่ม ฟังก์ชันความหนาแน่นความน่าจะเป็น ฟังก์ชันการแจกแจงความน่าจะเป็น ค่าคาดหวัง ความแปรปรวน ความแปรปรวนร่วม สหสัมพันธ์ และ ความเป็นอิสระระหว่างกันของข้อมูล เป็นต้น

### 3.2 ทำไมจึงต้องสถิติ

ในโปรแกรมประยุกต์ต่างๆ ที่เกี่ยวข้องกับการจัดการภาพ เช่น ภาพ X-ray การวิเคราะห์เซลล์มะเร็ง ภาพการจราจร (Traffic scene image) และ ภาพจากการทดลองเกี่ยวกับการวัดความแข็งแรงของโครงสร้างของซูริมิเจล ในรูปที่ 3.1 [7] [8]



รูปที่ 3.1: ความแข็งแรงของเจลซูริมิ

จากรูป 3.1 เราพบว่า

1. แต่ละภาพ หรือ แต่ละเส้น ในการทดลองทั้งหมด อาจจะแตกต่างกัน, และ

2. ในเวลาเดียวกัน หลายๆ ภาพอาจจะมีคุณลักษณะบางประการที่เหมือนกัน

ดังนั้น เราจำเป็นต้องการใช้สถิติเพื่อ

1. แยกกลุ่มของภาพโดยใช้ความแตกต่างกันทางกายภาพ
2. หาคุณลักษณะบางประการที่เหมือนกันของข้อมูลภาพ
3. ใช้คุณสมบัติในข้อ 2 เพื่อแทนข้อมูลภาพด้วยจำนวนบิตที่ต่ำที่สุดเพื่อลดขนาดภาพ
4. ภาพแต่ละภาพในข้อ 3 เมื่อแทนข้อมูลด้วยจำนวนบิตที่ต่ำที่สุดแล้ว จะต้องไม่มีสหสัมพันธ์ซึ่งกันและกัน (Uncorelated)
5. เพื่อสร้างกลับภาพ (reconstruction of the images) ที่มีค่าความคาดเคลื่อนต่ำที่สุด

### 3.3 ตัวแปรสุ่ม (Random variable)

นิยาม

พื้นที่สุ่ม *random field* คือ ฟังก์ชันเชิงพื้นที่สำหรับการกำหนดค่าตัวแปรสุ่มแต่ละตัวลงไปในพื้นที่ต่างๆ

ตัวแปรสุ่ม *random variable* คือ ค่าใดๆ ซึ่งถูกกำหนดได้จากการทดลองอย่างสุ่ม

การอธิบายตัวแปรสุ่ม โดยปกติจะอธิบายในรูปของฟังก์ชันการแจกแจง (*distribution functions*) ซึ่งก็คือ ความน่าจะเป็นของเหตุการณ์ที่กำลังเกิดขึ้น

### 3.4 ความน่าจะเป็นของเหตุการณ์ (Probability of Events)

กำหนดให้  $A, B, \dots, N$  เป็นเหตุการณ์ใดๆ ความน่าจะเป็น (probability) ของแต่ละเหตุการณ์ คือ ค่าจำนวนจริงระหว่าง 0 ถึง 1 โดยแทนด้วยสัญลักษณ์  $P[A], P[B], \dots, P[N]$  โดยถ้าเหตุการณ์นั้นเกิดขึ้นแน่นอน ค่าความน่าจะเป็นของเหตุการณ์นั้นจะมีค่าเป็น 1 ในทางตรงกันข้าม ถ้าเหตุการณ์นั้นไม่เกิดขึ้นแน่นอน ค่าความน่าจะเป็นของเหตุการณ์ก็จะมีค่าเท่ากับ 0 และ ถ้า  $A$  และ  $B$  เป็นเหตุการณ์ซึ่งเติมเต็มซึ่งกันและกัน (complementary) แล้ว ผลรวมของค่าความน่าจะเป็นของเหตุการณ์ทั้งสองต้องมีค่าเป็น 1 ดังสมการ (3.1)

$$P[B] = 1 - P[A] \quad (3.1)$$

ความน่าจะเป็นร่วมของเหตุการณ์  $A$  และ  $B$  (joint probability of events) เขียนแทนด้วย  $P[AB]$  หรือ  $P[A \text{ and } B]$  คือ ความน่าจะเป็นซึ่งทั้งสองเหตุการณ์จะเกิดร่วมกัน และเกิดพร้อมๆ กัน สามารถเขียนแทน ได้ดังสมการ (3.2)

$$P[A \text{ and } B] = P(A \cap B) \quad (3.2)$$

### 3.5 ฟังก์ชันการแจกแจงและฟังก์ชันความหนาแน่นความน่าจะเป็น (Probability Distribution and Density Functions)

#### 3.5.1 การแจกแจงเวกเตอร์สุ่ม (Distribution of a Random Vector)

กำหนดให้ ตัวแปรสุ่มทุกตัวเป็นตัวแปรสุ่มชนิดต่อเนื่อง (continuous-valued random variable) และ กำหนดให้  $\mathbf{x}$  เป็นเวกเตอร์สุ่ม (random vector) ความสัมพันธ์ระหว่างตัวแปรสุ่มและเวกเตอร์สุ่ม จะเป็นดังนี้

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}. \quad (3.3)$$

เมื่อตัวประกอบ  $x_1, x_2, x_3, \dots, x_n$  ของเวกเตอร์  $\mathbf{x}$  เป็นตัวแปรสุ่มชนิดต่อเนื่อง กำหนดให้เวกเตอร์  $\hat{\mathbf{x}}$  เป็นตัวแทนของเวกเตอร์  $\mathbf{x}$  นั่นคือ

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \vdots \\ \hat{x}_n \end{bmatrix}. \quad (3.4)$$

ตัวประกอบ  $\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n$  เป็นค่าจำนวนจริง และ ค่าความน่าจะเป็นของเหตุการณ์จะเขียนได้ ดังนี้

$$\mathbf{x} \leq \hat{\mathbf{x}} : x_1 \leq \hat{x}_1, x_2 \leq \hat{x}_2, \dots, x_n \leq \hat{x}_n$$

ซึ่งเป็นฟังก์ชันของ  $\hat{\mathbf{x}}$  เรียกว่า ฟังก์ชันการแจกแจงสะสม (*cumulative distribution function: cdf*) ของเวกเตอร์สุ่ม  $\mathbf{x}$  หรือ  $F$  สำหรับค่าฟังก์ชันการแจกแจงสะสมของตัวแปร  $x_i$  ณ จุด  $\hat{x}_i$  สามารถนิยามในรูปของความน่าจะเป็นซึ่ง  $x_i \leq \hat{x}_i$  ได้ดังนี้

$$F_{\mathbf{x}}(\hat{\mathbf{x}}) \equiv P_{x_1, x_2, \dots, x_n}(\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n) \equiv P(\mathbf{x} \leq \hat{\mathbf{x}}) \quad (3.5)$$

เมื่อ  $-\infty \leq \hat{\mathbf{x}} \leq \infty$  ซึ่งเป็นที่ทราบกันดีว่าสำหรับตัวแปรชนิดต่อเนื่อง ค่าฟังก์ชันการแจกแจงสะสมจะอยู่ในช่วง  $0 \leq F_{\mathbf{x}}(\hat{\mathbf{x}}) \leq 1$

แต่โดยปกติแล้ว ฟังก์ชันความน่าจะเป็นจะนิยามในรูปของฟังก์ชันความหนาแน่นความน่าจะเป็น (*probability density function : pdf*) มากกว่าการใช้ในรูปของฟังก์ชันการแจกแจงสะสม ฟังก์ชันความหนาแน่นความน่าจะเป็น  $p_{\mathbf{x}}(\hat{\mathbf{x}})$  ของตัวแปรสุ่ม  $x$  จะหาได้จากอนุพันธ์ของฟังก์ชันการแจกแจงสะสมเทียบกับตัวแปรสุ่มทุกตัว  $p_{\mathbf{x}}(\hat{\mathbf{x}}) = \frac{dF(x)}{dx_i}$  แต่สำหรับกรณีหลายตัวแปร (*multivariate*) ค่าฟังก์ชันความหนาแน่นความน่าจะเป็น  $p_{\mathbf{x}}(\hat{\mathbf{x}})$  จะได้จากการหาอนุพันธ์ของฟังก์ชันการแจกแจงสะสมเทียบกับตัวแปรสุ่มทุกตัวในเวกเตอร์สุ่ม  $\mathbf{x}$ :

$$p_{\mathbf{x}}(\hat{\mathbf{x}}) = \frac{\partial}{\partial x_1} \frac{\partial}{\partial x_2} \dots \frac{\partial}{\partial x_n} F_{\mathbf{x}}(\hat{\mathbf{x}}) \quad (3.6)$$

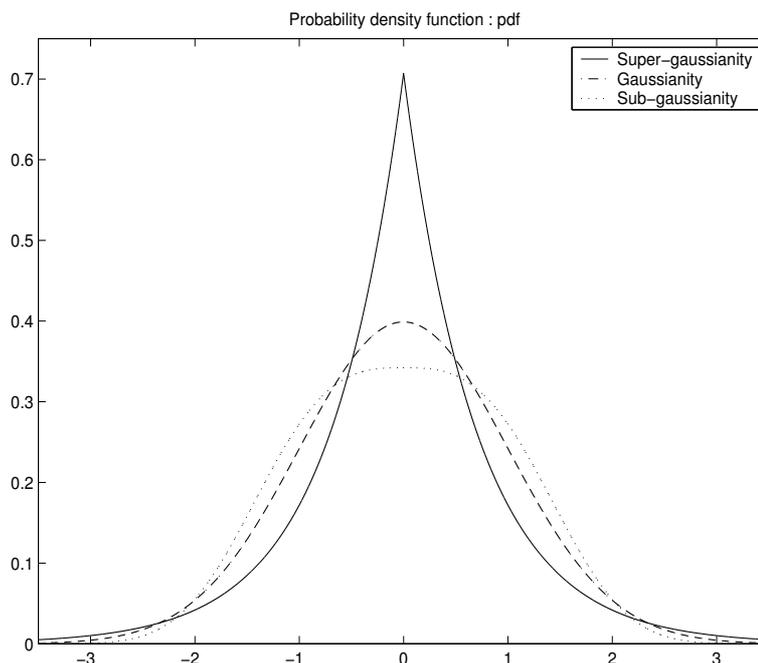
ในทางตรงกันข้าม สมมติเวกเตอร์  $\mathbf{x}$  เป็นเวกเตอร์ขนาด  $n$  มิติดังนิยามในสมการ (3.3) ค่าฟังก์ชันการแจกแจงสะสมของเวกเตอร์  $\mathbf{x}$  สามารถนิยามได้ดังสมการ (3.7)

$$F_{\mathbf{x}}(\mathbf{x}_0) = P(\mathbf{x} \leq \mathbf{x}_0) = \int_{-\infty}^{\infty} p_{\mathbf{x}}(\mathbf{x}_0) \quad (3.7)$$

### 3.5.2 ฟังก์ชันการแจกแจงความน่าจะเป็นร่วมและความหนาแน่นความน่าจะเป็นร่วม

#### (Joint Distribution and Probability Density Functions)

ถ้า  $\mathbf{x}$  และ  $\mathbf{y}$  เป็นเวกเตอร์สุ่มทั้งคู่และ เวกเตอร์สุ่มทั้งสองอาจมีมิติข้อมูลไม่เท่ากัน จะพบว่า เวกเตอร์  $\mathbf{x}$  และ  $\mathbf{y}$  สามารถนำมาเชื่อมต่อกันเป็นซูเปอร์เวกเตอร์ (*supervector*)  $\mathbf{z}^T = (\mathbf{x}^T, \mathbf{y}^T)$  ค่าฟังก์ชันการแจกแจงสะสมของซูเปอร์เวกเตอร์ จะเรียกว่า ฟังก์ชันการแจกแจงร่วม (*joint distribution function*) ของ  $\mathbf{x}$  และ  $\mathbf{y}$  ซึ่งสามารถเขียนเป็นสมการได้ดังสมการ (3.8) :



รูปที่ 3.2: ฟังก์ชันความหนาแน่นความน่าจะเป็น

$$F_{x,y}(\hat{x}, \hat{y}) = P(x \leq \hat{x}, y \leq \hat{y}) \quad (3.8)$$

ในทำนองเดียวกันกับฟังก์ชันความหนาแน่นความน่าจะเป็น (pdf) ซึ่งได้อธิบายไปในหัวข้อก่อนหน้า สำหรับฟังก์ชันความหนาแน่นความน่าจะเป็นร่วม (joint pdf)  $p_{x,y}(\hat{x}, \hat{y})$  ของเวกเตอร์  $x$  และ เวกเตอร์  $y$  ก็จะสามารถนิยามในรูปของอนุพันธ์ของฟังก์ชันการแจกแจงร่วม  $F_{x,y}(\hat{x}, \hat{y})$  เทียบกับ ทุกๆ องค์ประกอบภายในเวกเตอร์สุม  $x$  และ  $y$  ดังนั้น ความสัมพันธ์ระหว่างฟังก์ชันความหนาแน่นความน่าจะเป็นร่วม กับ ฟังก์ชันการแจกแจงร่วม ก็จะสามารถเขียนได้ดังสมการ (3.10)

$$p_{x,y}(\hat{x}, \hat{y}) = \frac{\partial}{\partial x_1} \frac{\partial}{\partial x_2} \cdots \frac{\partial}{\partial x_n} \frac{\partial}{\partial y_1} \frac{\partial}{\partial y_2} \cdots \frac{\partial}{\partial y_m} F_{x,y}(\hat{x}, \hat{y}) \quad (3.9)$$

และ

$$F_{x,y}(\hat{x}, \hat{y}) = \int_{-\infty}^{\hat{x}} \int_{-\infty}^{\hat{y}} p_{x,y}(\xi, \sigma) d\sigma d\xi \quad (3.10)$$

### 3.6 ค่าคาดหวัง (Expectation)

ค่าคาดหวัง (*expectation*) หรือ ค่าเฉลี่ย (*mean*) ของเวกเตอร์สุ่ม  $\mathbf{x}$  เป็นเวกเตอร์ของค่าคงที่ (constant vector)  $\mathbf{m}$  ซึ่งนิยาม ดังนี้

$$\mathbf{m} = E[\mathbf{x}] = \int_{-\infty}^{\infty} \mathbf{x} p_{\mathbf{x}}(\mathbf{x}) d\mathbf{x} \quad (3.11)$$

และ ค่าเฉลี่ยของตัวประกอบที่  $i$ th ของเวกเตอร์สุ่ม  $\mathbf{x}$  จะมีค่าเป็น

$$m_i = E[x_i] = \int_{-\infty}^{\infty} x_i p_{x_i}(x_i) dx_i \quad (3.12)$$

ค่าความแปรปรวน (Variance) ของเวกเตอร์สุ่ม  $\mathbf{x}$  สามารถนิยามในรูปของผลต่างระหว่างค่าคาดหวัง กับ ตัวแปรสุ่ม ยกกำลังสอง ดังนี้

$$\begin{aligned} \sigma^2 \equiv Var(x_i) &= E[(x_i - m_i)^2] \\ &= \int_{-\infty}^{\infty} (x_i - m_i)^2 p_{x_i}(x_i) dx_i \end{aligned} \quad (3.13)$$

ความแปรปรวนร่วม (Covariance) ระหว่างสองเวกเตอร์ นิยามได้ ดังนี้

$$c_{ij} = E[(f_i - m_{f_i})(f_j - m_{f_j})], \quad \forall i, j, \quad i \neq j \quad (3.14)$$

### 3.7 การทดสอบความอิสระของข้อมูล

ตัวอย่างการนำสถิติไปใช้อย่างหนึ่งในเรื่องการประมวลผลภาพ คือ การใช้สถิติในการทดสอบความเป็นอิสระ (Independence) ระหว่างสัญญาณสองสัญญาณ กล่าวคือ ในการทดสอบความเป็นอิสระของสองสัญญาณ  $y_i$  และ  $y_j$  นั้น ค่าของ  $y_i$  และ  $y_j$  จะต้องเป็นอิสระทางสถิติตลอดช่วงเวลาของสัญญาณที่ใช้ในการทดสอบ ซึ่งก็มีวิธีการทางสถิติมากมาย ที่ใช้ในการทดสอบค่าความเป็นอิสระทางสถิติ เช่น การทดสอบความเป็นอิสระของตัวแปรสุ่ม  $y_i$  และ  $y_j$  โดยใช้ฟังก์ชันความหนาแน่นความน่าจะเป็นของตัวแปรสุ่ม  $y_i$  ดังนี้ กำหนดให้  $p(y_i, y_j)$  ฟังก์ชันความหนาแน่นความน่าจะเป็นร่วมของตัวแปรสุ่ม  $y_i$  และ  $y_j$ , และ  $p_i(y_i)$  ฟังก์ชันความหนาแน่นความน่าจะเป็นส่วนริมของ  $y_i$  ดังนั้น

$$p_i(y_i) = \int_0^\infty p(y_i, y_j) dy_j$$

และ

$$p(y_i, y_j) = p_i(y_i)p_j(y_j) \tag{3.15}$$

แต่ในทางปฏิบัติ เป็นการยากที่จะทำการทดสอบความเป็นอิสระของ  $y_i$  และ  $y_j$  โดยใช้ ฟังก์ชันความหนาแน่นความน่าจะเป็น  $p(y_i, y_j)$ ,  $p_i(y_i)$ , และ  $p_j(y_j)$  ดังนั้น เราจึงนิยมใช้ค่าสถิติซึ่งคำนวณได้ง่ายกว่า คือ ค่าความสัมพันธ์ระหว่างตัวแปร (correlation) กล่าวคือ สองตัวแปรสุ่ม  $y_i$  และ  $y_j$  จะไม่มีความสัมพันธ์กัน (uncorrelated) คำว่า ตัวแปรสุ่มทั้งสองไม่มีความสัมพันธ์กัน หมายถึง ค่าของตัวแปรสุ่มทั้งสองไม่ขึ้นต่อกัน ในที่นี้ ถ้า ความแปรปรวนร่วม (covariance) ของตัวแปรสุ่มทั้งสองเป็นศูนย์แล้ว แสดงว่า ตัวแปรสุ่มทั้งสองไม่มีความสัมพันธ์กัน ซึ่งความแปรปรวนร่วมสามารถคำนวณได้จากค่าคาดหวังของความสัมพันธ์ (correlated expected values) กับ ผลคูณของค่าคาดหวัง  $y_i$  และ  $y_j$  ดังนี้

$$E[y_i y_j] - E[y_i]E[y_j] = 0 \tag{3.16}$$

**หมายเหตุ** สำหรับตัวแปรสุ่มสองตัวซึ่งอิสระจากกัน แล้ว ตัวแปรสุ่มทั้งสองจะไม่มีความสัมพันธ์กัน ในทางตรงกันข้าม ถ้าตัวแปรสุ่มทั้งสองไม่อิสระจากกัน ไม่ได้หมายความว่า ตัวแปรสุ่มทั้งสองจะอิสระจากกัน

### 3.8 สรุป

ในบทนี้ ได้กล่าวถึง คณิตศาสตร์และ สถิติพื้นฐานที่จำเป็นสำหรับการประมวลผลภาพ เพื่อเป็นการปรับพื้นฐานของผู้เรียน และ แสดงให้เห็นความสำคัญของ คณิตศาสตร์และสถิติต่อการประมวลผลภาพ และ หวังว่าผู้เรียนจะสามารถนำความรู้พื้นฐานเหล่านี้ไปใช้ในบทถัดไป

## บทที่ 4

### การปรับปรุงภาพ Image Enhancement

#### เนื้อหาสาระ

1. การดำเนินการบนโดเมนภาพเชิงพื้นที่ และ เชิงความถี่
2. การปรับปรุงภาพด้วยฮิสโตแกรม (Histogram)
3. การหาผลต่างของภาพ Image Subtraction
4. ตัวกรองปรับเรียบ (Smoothing Filters)
5. ตัวกรองปรับสว่าง (Sharpen Filters)

#### วัตถุประสงค์

1. เพื่ออธิบายกระบวนการการปรับปรุงภาพ
2. เพื่ออธิบายการเขียนโปรแกรมตามวิธีที่ได้ศึกษาไว้

#### กิจกรรมการเรียนการสอน

1. บรรยาย อภิปรายซักถาม และ ยกตัวอย่าง
2. สรุปเนื้อหารายวิชา

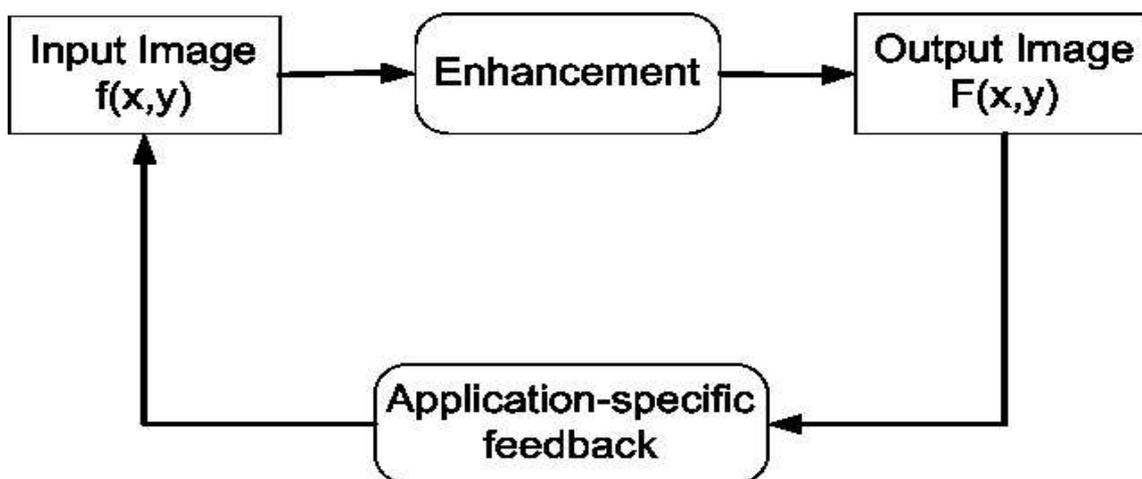
#### การวัดผล

1. ซักถามความเข้าใจหลังการเรียน

## การปรับปรุงภาพ Image Enhancement

### 4.1 บทนำ

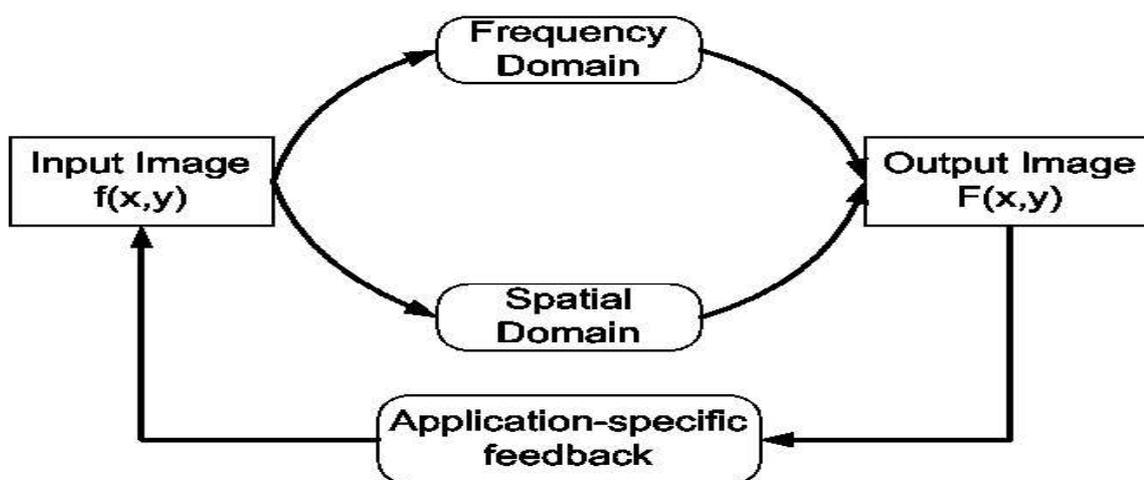
การปรับปรุงภาพ (Image Enhancement) เป็นกระบวนการซึ่งเกี่ยวข้องกับการพยายามทำให้ภาพดูดีขึ้น ซึ่งในความเป็นจริงแล้วเราไม่ทราบว่าแท้จริงแล้วภาพนั้นๆ ควรจะเป็นอย่างไร แต่เราสามารถบอกได้ว่าเมื่อเราดำเนินการบางอย่างกับภาพนั้นแล้ว จะทำให้เรารู้สึกได้ว่าภาพนั้นดูดีขึ้นหรือไม่ ซึ่งพิจารณาได้จาก ยกตัวอย่างเช่น สามารถทำให้เพิ่มรายละเอียดของภาพมากยิ่งขึ้น สิ่งที่ไม่ต้องการในภาพถูกกำจัดออกไป หรือ ความสว่างของภาพดีขึ้น เป็นต้น ในรูปที่ 4.1 แสดงให้เห็นขั้นตอนการปรับปรุงภาพอย่างง่าย โดยการแปลงข้อมูลนำเข้า (Input Image) เป็นภาพผลลัพธ์ (Output Image) โดยใช้ฟังก์ชัน Enhancement และ ภาพผลลัพธ์ที่ได้ดังกล่าวนั้น ก็จะสามารถนำไปใช้งานต่างๆ ได้ตามต้องการ หรือ สามารถนำกลับไปผ่านฟังก์ชัน Enhancement อีกก็ได้



รูปที่ 4.1: การปรับปรุงภาพ

การปรับปรุงภาพ มี 2 วิธีหลักๆ คือ

- Spatial domain method คือ การดำเนินการบนโดเมนภาพเชิงพื้นที่
- Frequency domain method คือ การดำเนินการบนโดเมนภาพเชิงความถี่



รูปที่ 4.2: การปรับปรุงภาพบนโดเมนเชิงพื้นที่และ โดเมนเชิงความถี่

ในทำนองเดียวกัน ก้กับการปรับปรุงภาพเบื้องต้น ภาพที่ได้รับการปรับปรุงแล้วสามารถนำไปใช้ได้ตามวัตถุประสงค์ หรือ สามารถนำกลับเข้ามาสู่ขั้นตอนการปรับปรุงอีกครั้งก็ได้ ถ้าเราพิจารณาเห็นแล้วว่าภาพที่ปรับปรุงได้ยังไม่มีคุณภาพดีพอ

## 4.2 การดำเนินการบนโดเมนภาพเชิงพื้นที่ (Spatial domain method)

การดำเนินการปรับปรุงภาพเชิงพื้นที่ คือ การดำเนินการกับค่าระดับเทาของจุดภาพโดยตรง (Gray values of the pixels) ซึ่งการดำเนินการของการประมวลผลภาพเชิงพื้นที่นั้น จะแบ่งออกเป็น 3 กลุ่มที่สำคัญ คือ

1. การแปลง (Transforms) คือ การแทนค่าของจุดภาพต่างๆ ในรูปแบบอื่นๆ แต่ค่าของจุดภาพเหล่านั้นยังมีค่าเท่าเดิม ซึ่งการแปลงนั้นเป็นวิธีการที่นิยมใช้กันอย่างแพร่หลาย และ ให้ประสิทธิภาพในการทำงานสูง ยกตัวอย่างเช่น การแปลงข้อมูลจากข้อมูลภาพเชิงพื้นที่ ไปเป็น ข้อมูลภาพเชิงความถี่ และก็จะมีการดำเนินการบางอย่างกับข้อมูลภาพนั้นบนโดเมนเชิงความถี่ จากนั้น ก็จะมีการแปลงข้อมูลภาพดังกล่าวนั้น (ซึ่งอยู่ในโดเมนของความถี่) กลับมายังโดเมนของพื้นที่
2. การประมวลผลจุด (Point Processing)
3. การประมวลผลเป็นพื้นที่ (Neighborhood Processing)

### 4.3 การดำเนินการบนโดเมนภาพเชิงความถี่ (Frequency domain method)

นอกจากการปรับปรุงภาพในโดเมนภาพเชิงพื้นที่แล้ว ยังมีการปรับปรุงภาพอีกวิธีหนึ่ง ซึ่งได้รับความนิยมมาก คือ การปรับปรุงภาพ บนโดเมนของความถี่ โดยเริ่มจาก ในปี ค.ศ 1807 นักคณิตศาสตร์ชาวฝรั่งเศส ชื่อ Jean Baptiste Joseph Fourier ได้นำเสนอผลงานเกี่ยวกับการนำและการแพร่ความร้อนต่อ Institute de France ซึ่งในการนำเสนอดังกล่าว Joseph Fourier ได้แสดงให้เห็นว่า ทุกๆ สัญญาณที่มีคาบ (periodic signal) สามารถแทนได้อยู่ในรูปของอนุกรมของไซน์ (series of sinusoids) ซึ่งแนวคิดดังกล่าวได้ถูกนำไปใช้อย่างแพร่หลาย และรู้จักโดยทั่วไปว่า อนุกรมของฟูริเยร์ (Fourier series) การปรับปรุงภาพในโดเมนของความถี่ มีขั้นตอนที่สำคัญๆ ดังนี้

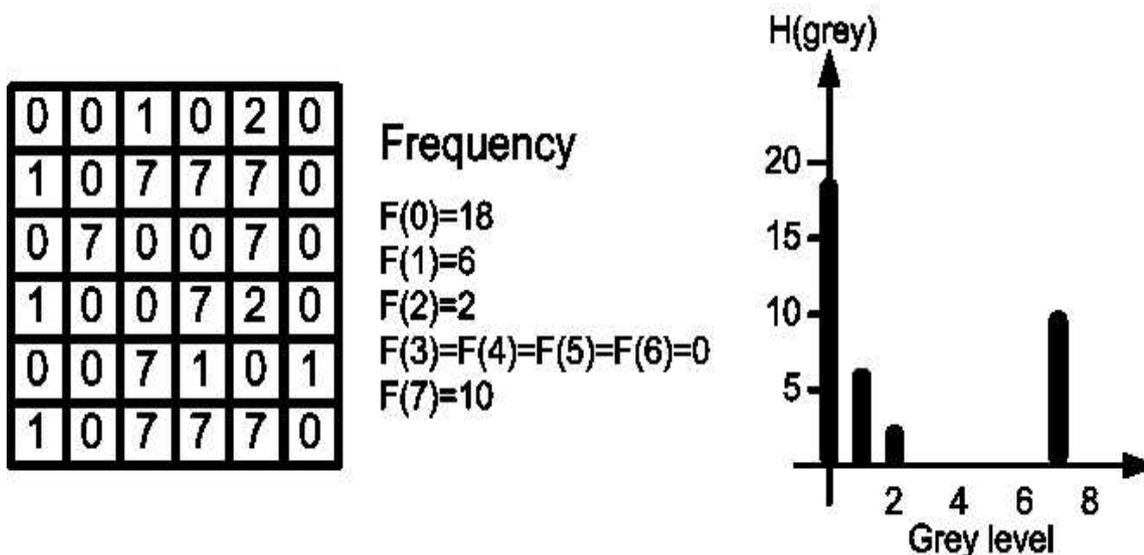
1. การแปลงข้อมูลจากโดเมนภาพเชิงพื้นที่ เป็น โดเมนภาพเชิงความถี่
2. การปรับปรุงภาพบนโดเมนของความถี่ เช่น การลบความถี่พื้นฐาน  $F_0$  ของสัญญาณรบกวนซึ่งเราไม่ต้องการออก
3. การแปลงข้อมูลภาพจากโดเมนภาพเชิงความถี่ เป็น โดเมนภาพเชิงพื้นที่

### 4.4 การปรับปรุงภาพด้วยฮิสโตแกรม (Histogram)

ฮิสโตแกรมของภาพ (Histogram of an image) คือ ฟังก์ชันเต็มหน่วย สำหรับการนับจำนวนของจุดภาพซึ่งมีค่าระดับเทาเดียวกัน และ เมื่อ พิจารณา เราจะพบว่า ฟังก์ชันฮิสโตแกรมสามารถแปลงให้อยู่ในช่วง  $[0,1]$  สำหรับทุกๆ ค่าระดับเทา (เมื่อหาผลรวมของค่าฮิสโตแกรมทุกค่า) นั่นคือ เราสามารถพิจารณาฟังก์ชันฮิสโตแกรม ในรูปของ ฟังก์ชันความหนาแน่นความน่าจะเป็น (*Probability density function*) ดังตัวอย่างในรูปที่ 4.3 หรือ กล่าวอีกอย่างหนึ่งคือ เราสามารถหาความน่าจะเป็นของค่าระดับเทาบนภาพ ได้จากค่าของฮิสโตแกรม ดังนี้

$$\sum_{i=0}^{G_{max}} h_i = 1 \quad (4.1)$$

การคำนวณค่าฮิสโตแกรมสามารถแสดงเป็นขั้นตอนวิธีสำหรับการโปรแกรม ได้ดัง รูปที่ 4.4



รูปที่ 4.3: ภาพ 8 ระดับ การแจกแจงความถี่ และ ฮิสโตแกรมของภาพ

#### 4.5 การปรับปรุงค่าฮิสโตแกรมสำหรับภาพ

ในบางครั้ง การกระจายตัวของค่าฮิสโตแกรมของภาพยังไม่ดีพอที่จะทำให้เราสามารถวิเคราะห์ภาพได้อย่างถูกต้อง ยกตัวอย่างเช่น ในภาพที่ 4.5(a) ภาพต้นฉบับที่ได้มานั้นจะค่อนข้างมืด ยกแก่การประมวลผล และ ยิ่งไปกว่านั้น เมื่อทำการวาดฮิสโตแกรมของภาพออกมามาดังในรูป 4.5(b) ก็เห็นได้ชัดว่า ค่าความเข้มของจุดภาพในรูป 4.5(a) นั้นจะมีค่าอยู่ระหว่าง  $[0-40]$  นั่นคือ ภาพต้นฉบับที่ได้มาจะมีมืด หรือ ความเข้มน้อย ซึ่งวิธีการหนึ่งที่ได้รับการนิยมนิยม คือ การทำให้ค่าของฮิสโตแกรมของภาพทุกค่าเท่ากัน (Histogram Equalization) หรือ การขยายฮิสโตแกรมออกไปทางด้านใดด้านหนึ่ง ยกตัวอย่างเช่น ในที่นี้ จะทำการขยายไปทางด้านขวา เพื่อเพิ่มความสว่างให้ภาพ ซึ่งก็จะได้ภาพผลลัพธ์ออกมาตามรูปที่ 4.5(c) และ มีค่าฮิสโตแกรมของภาพใหม่ดังรูป 4.5(d)

จากที่ได้กล่าวมาข้างต้น เราสามารถดำเนินการปรับปรุงภาพโดยใช้ฮิสโตแกรมของภาพได้ด้วยขั้นตอนวิธี ดังต่อไปนี้

- Suppose the grey levels in the original image are given by the values a variable  $r$  obtains and in the new image by the values a variable  $s$  obtains
- Then we would like to find a transformation  $s = T(r)$ , such that  $p_r(r)$  into

**Algorithm: Histogram****Input:** Input signal,  $x$ **Output:** Histogram of Images  $h$ **begin**     $[r, c] = \text{size}(x);$     **For**  $i = 1:r$         **For**  $j = 1:c$              $\text{Hist}\{x[i,j]\} = \text{Hist}\{x[i,j]\} + 1$         **Endfor**    **Endfor**    **For**  $g = 1:G_{\max}$          $\text{Hist}[g] = \text{Hist}[g] / (M * N)$     **Endfor****end.**

รูปที่ 4.4: Pseudo code to compute the histogram of an image.

$p_s(s)$ .

- Since  $p_r(r)$  is the *pdf* of RV  $r$ , the number of pixels with grey level values in the range  $r$  to  $r + dr$  is  $p_r(r)dr$ .
- The transform we seek will transform this range to  $[s, s + ds]$ .
- The total number of pixels in  $[s, s + ds]$  will remain the same but in the enhanced image this number will be  $p_s(s)ds$

นั่นคือ

$$p_r(r)dr = p_s(s)ds \quad (4.2)$$

### 4.5.1 Histogram Equalization

การทำค่าฮิสโตแกรมให้เท่ากัน (Histogram Equalization) คือ กระบวนการสำหรับการกระจายค่าระดับเทาบนภาพที่จะทำให้ค่าระดับเทาทุกๆ ค่ามีโอกาสเกิดขึ้นเท่ากัน  $p_s(s) = c$  หรือ กว่ อีกร้อยกก็คือ การทำค่าฮิสโตแกรมให้เท่ากัน คือ การทำให้ฮิสโตแกรมของภาพผลลัพธ์มีการกระจายตัวแบบยูนิฟอร์ม (*Uniformly distribution*) นั่นคือ

$$\int_0^s cds = \int_0^r p_r(r)dr \Rightarrow s = \frac{1}{c} \int_0^r p_r(x)dx \quad (4.3)$$

### 4.5.2 ขั้นตอนการทำ Histogram Equalization

ก่อนการดำเนินการทุกอย่าง ต้องมั่นใจว่าข้อมูลภาพที่จะทำ Histogram Equalization นั้น ต้องอยู่ในรูปเต็มหน่วย (discrete case) ดังนั้น ถ้าข้อมูลภาพยังอยู่ในรูปต่อเนื่อง (continuous case) ให้แปลงให้อยู่ในรูปเต็มหน่วย

จากที่กล่าวมา กระบวนการ Histogram Equalization คือการหาฟังก์ชันการแทนค่า (mapping function)  $M(r)$  ซึ่งจะใช้แทนค่าฟังก์ชันความหนาแน่นความน่าจะเป็นของภาพนำเข้า (input PDF of image)  $p_r(r)$  ด้วยฟังก์ชันความหนาแน่นความน่าจะเป็นของภาพผลลัพธ์  $p_s(s)$  ซึ่งมีการกระจายตัวแบบยูนิฟอร์ม (*Uniformly distribution*) นั่นคือ

$$p_s(s) = \left\{ p_r(r) \frac{dr}{ds} \right\}_{r=M^{-1}(s)} \quad (4.4)$$

และ

$$p_s(s) = \begin{cases} \frac{1}{Z-W} & \text{for } W \leq s < Z \\ 0 & \text{otherwise} \end{cases} \quad (4.5)$$

จะพบว่า สมการ 4.4 สามารถเขียนใหม่ในรูปของการเปลี่ยนแปลงของความน่าจะเป็น (Differential probability) ดังสมการ 4.6

$$p_s(s)ds = p_r(r)dr \quad (4.6)$$

ทำการหาปฏิยานุพันธ์ทั้งสองข้างของสมการ 4.6 ได้ดังสมการ ที่ 4.7

$$\int_{h \in R} p_r(h)dh = \int_{v \in S} p(v)dv \quad (4.7)$$

แทนสมการ 4.5 ในสมการ 4.7, และ เราจะได้  $s$  สำหรับหาฟังก์ชันการแทนค่า ดังนี้

$$s = M(r) = (Z - W) \int_{h \in R} p_r(h)dh + W \quad \text{for } W \leq s \leq Z \quad (4.8)$$

หรือ

$$s_k = M_{r_k} = Gmax \sum_{i=0}^k hist_i \quad \text{for } 0 \leq k \leq Gmax \quad (4.9)$$

ตัวอย่าง ในตารางที่ ข้อมูลภาพเทา 8 ระดับ ขนาด 16384 จุดภาพ มีค่าฮิสโตแกรมของแต่ละจุดภาพ ดังสดมภ์ที่ 1 และ 2 ในตาราง ที่ 4.1 ซึ่งพบว่า ภาพนั้นค่อนข้างมืด หรือ ฮิสโตแกรมกระจายตัวไม่สม่ำเสมอ จึงทำการปรับปรุงฮิสโตแกรมของภาพใหม่ โดยใช้สมการ (4.9) และ ได้ฮิสโตแกรมของภาพใหม่ ดังแสดงในสดมภ์ที่ 5

ตารางข้อมูลรายปี				
Grey Levels	$n_i$	$h_i$	Equation (4.9)	M(a)
0	1116	0.0681	0.476	0
1	4513	0.2754	2.405	2
2	5420	0.3308	4.720	4
3	2149	0.1312	5.678	6
4	1389	0.0848	6.232	6
5	917	0.0560	6.624	7
6	654	0.0399	6.903	7
7	222	0.0138	7	7
$n_t$	16384			

ตารางที่ 4.1: การคำนวณ Histogram Equalization

### 4.5.3 การดำเนินการด้วย Histogram แบบอื่นๆ

นอกจากวิธีการ Histogram Equalization แล้ว เรายังสามารถใช้ค่าสถิติอื่นๆ สำหรับการปรับปรุงภาพ เช่น การใช้ ค่าเฉลี่ย (mean) และ ส่วนเบี่ยงเบนมาตรฐาน (standard deviation) ของจุดภาพภาพในหน้าต่างที่กำหนด (Interesting window or region) ดังนี้

1. Let  $m(x, y)$  be the mean grey value inside a window centred at  $(x, y)$ ,
2.  $\sigma(x, y)$  be the variance of the pixel inside a window, and
3.  $f(x, y)$  is the value of pixel  $(x, y)$

และ เราสามารถปรับปรุงค่าความแปรปรวนภายในหน้าต่างได้โดย

$$g(x, y) = A[f(x, y) - m(x, y)] + m(x, y) \quad (4.10)$$

เมื่อ  $A$  คือ สเกลาร์

และ บางครั้ง เราสามารถเลือก *amplification factor*  $A$  สำหรับ  $\sigma(x, y)$  ดังนี้

$$A = \frac{kM}{\sigma(x, y)} \quad (4.11)$$

เมื่อ  $k$  คือ ค่าคงที่ และ  $M$  เป็นค่าเฉลี่ยของระดับเทาของภาพ

### 4.6 การหาผลต่างของภาพ Image Subtraction

การหาผลต่างระหว่างภาพ (Image Subtraction) มีวัตถุประสงค์เพื่อหาความแตกต่างระหว่าง  $f(x, y)$  และ  $h(x, y)$  แม้ว่าจะเป็นวิธีการที่ง่ายและไม่ซับซ้อนแต่นิยมใช้มาก โดยเฉพาะอย่างยิ่งในวงการแพทย์ เช่น การวิเคราะห์ภาพจากเครื่อง CT-Scanner ซึ่งการวิเคราะห์ภาพการแพทย์ดังกล่าว ยกตัวอย่างเช่น ต้องการสแกนสมองคนไข้ แพทย์จะเริ่มจากการสแกนสมองในสภาวะปกติก่อน จากนั้น จะทำการฉีดสีให้กับคนไข้ และ จากนั้นจะทำการสแกนสมองซ้ำอีกครั้ง ในตำแหน่งเดิม และ ขั้นตอนสุดท้ายจะนำภาพที่ได้จากการสแกนทั้งสองครั้งมาหาผลต่างกัน ก็จะทำให้ทราบว่าคนไข้มีอาการผิดปกติหรือไม่ การหาผลต่างของสมการสามารถหาได้โดยง่ายดังนี้

$$g(x, y) = f(x, y) - h(x, y) \quad (4.12)$$

## 4.7 การดำเนินการบนโดเมนภาพเชิงพื้นที่ (Mask processing or Spatial domain method)

พิจารณาภาพ  $f(x, y)$  เมื่อจะถูกดำเนินการปรับปรุงภาพเชิงพื้นที่โดยการดำเนินการกับกลุ่มจุดภาพที่อยู่รอบข้าง (Area or mask processing method) โดย ตัวดำเนินการ  $T$  ซึ่ง  $f(x, y)$  ก็คือ ข้อมูลนำเข้า และ  $F(x, y)$  คือ ภาพซึ่งผ่านการปรับปรุงด้วย  $T$  ดังนั้น  $F(x, y) = T[f(x, y)]$  และ พิจารณารูปที่ 4.7 เมื่อพิจารณาเฉพาะจุดภาพที่เรากำลังสนใจ  $f5'$  จะคำนวณได้ ดังนี้

$$f5' = h1f1 + h2f1 + \dots + h8f8 + h9f9 \quad (4.13)$$

ข้อสังเกต

1. สำหรับการดำเนินการหาค่า  $f5'$  ที่เป็นไปตามสมการ 4.13 เราจะเรียกว่า การดำเนินการแบบเชิงเส้น (Linear filter)
2. สำหรับการดำเนินการที่นอกเหนือจากนี้ เราเรียกว่า การดำเนินการแบบไม่เป็นเชิงเส้น (Non-linear filter) เช่น  $f5' = \max(f_k, k = 1, \dots, 9)$

## 4.8 ตัวกรองปรับเรียบ (Smoothing Filters)

วิธีการนี้ นำมาใช้เพื่อลดสัญญาณรบกวน (Noise) และ เป็นการประมวลผลก่อน (Preprocessing) สำหรับการประมวลผลขั้นตอนอื่น เช่น การแยกวัตถุออกจากภาพ (Segmentation) ต่อไปตัวกรองการปรับเรียบนี้มีทั้งแบบที่เป็นเชิงเส้น และ ไม่เป็นเชิงเส้น

### 4.8.1 ตัวกรองปรับเรียบแบบเชิงเส้น (Linear Smoothing Filters)

#### Uniform filter

ตัวกรองประเภทนี้ ภาพผลลัพธ์ (output image) ที่ได้จะอยู่บนหลักการพื้นฐานของการหาค่าเฉลี่ยในบริเวณที่กำหนด (local averaging) ของภาพต้นฉบับกับตัวกรองที่กำหนดให้ (input filter) เมื่อกำหนดให้ ทุกๆ ค่าภายในตัวกรองมีค่าเท่ากัน โดยตัวกรองอาจเป็นได้ทั้งแบบ ตัวกรองสี่เหลี่ยม (rectangular case) และ แบบ วงกลม (circular case) ดังสมการที่ (4.15) และ

(4.17) ซึ่งในที่นี่กำหนดรัศมีวงกลมเท่ากับ 2.5 ตามลำดับ โดยตัวกรองทั้งสองตัวขนาดเท่ากันคือ เท่ากับ 5

$$h_{rect}(x, y) = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (4.14)$$

$$h_{rect}(x, y) = \frac{1}{25} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (4.15)$$

$$h_{rect}(x, y) = \frac{1}{49} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (4.16)$$

$$h_{circ}(x, y) = \frac{1}{25} \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix} \quad (4.17)$$

จากทั้งสามสมการพบว่า  $\sum h(x, y) = 1$  และ ภาพผลลัพธ์ที่ได้  $g(x, y) = \frac{1f_1+1f_2+\dots+1f_9}{9}$  สำหรับตัวกรองขนาด 3 เป็นต้น

### Triangular filter

สำหรับตัวกรองแบบนี้ ภาพผลลัพธ์ที่ได้ จะอยู่บนหลักการของการหาค่าเฉลี่ยบนพื้นที่ที่กำหนดของตัวกรองที่กำหนดให้ และ ค่าภายในตัวกรองจะไม่เท่ากัน ยกตัวอย่าง เช่น ตัวกรองในสมการที่ (4.18), (4.19) และ (4.20)

$$h_{rect}(x, y) = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad (4.18)$$

$$h_{rect}(x, y) = \frac{1}{81} \begin{bmatrix} 1 & 2 & 3 & 2 & 1 \\ 2 & 4 & 6 & 4 & 2 \\ 3 & 6 & 9 & 6 & 3 \\ 2 & 4 & 6 & 4 & 2 \\ 1 & 2 & 3 & 2 & 1 \end{bmatrix} \quad (4.19)$$

$$h_{circ}(x, y) = \frac{1}{25} \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 2 & 2 & 2 & 0 \\ 1 & 2 & 5 & 1 & 1 \\ 0 & 2 & 2 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (4.20)$$

## 4.8.2 ตัวกรองปรับเรียบแบบไม่เชิงเส้น (Non-linear Smoothing Filters)

### ตัวกรองมัธยฐาน (Median filter)

ค่ามัธยฐานทางสถิติ เป็นหนึ่งในสามวิธีในการหาค่ากลางทางสถิติ (อีกสองวิธี คือ ค่าเฉลี่ย และ ค่าฐานนิยม) ซึ่งได้รับความนิยมในการนำไปประยุกต์ใช้งานอย่างแพร่หลาย สำหรับค่าตัวกรองมัธยฐาน จะอยู่บนหลักการของการเลื่อนหน้าต่าง (moving a window) ไปบนข้อมูลภาพ (เช่นเดียวกับการทำ convolution) และ คำนวณค่าคำตอบของจุดภาพที่กำลังสนใจจากค่ามัธยฐานของค่าความสว่างในหน้าต่างที่กำหนด (input window) วิธีการคำนวณค่ามัธยฐาน คือ การนำค่าความสว่างของจุดภาพภายในหน้าต่างที่กำหนดมาเรียงลำดับจากน้อยไปมาก สำหรับหน้าต่างขนาด  $J \times K$  ถ้า  $J * K$  เป็นจำนวนคี่ ค่ามัธยฐานจะอยู่ในตำแหน่งที่  $(J * K + 1)/2$  และ ถ้า  $J * K$  เป็นจำนวนคู่ ค่ามัธยฐานจะคำนวณได้จาก ค่าเฉลี่ยความสว่างของจุดภาพในตำแหน่งที่  $(J * K)/2$  และ  $(J * K + 1)/2$  ดังตัวอย่าง

## 4.9 ตัวกรองปรับสว่าง (Sharpen Filters)

เป็นตัวกรองที่ใช้กันทั่วไป เมื่อต้องการเน้นให้เห็นส่วนของภาพที่มีการเปลี่ยนแปลงอย่างทันทีทันใด เช่น บริเวณของขอบวัตถุในภาพ (Edges) และ ลายเส้นที่อยู่บนภาพ เป็นต้น

### 4.9.1 ตัวกรองความถี่สูง (High-pass filters (Linear Filter))

ตัวกรองชนิดนี้ สมาชิกที่อยู่ภายในหน้ากากตัวกรอง (Mask) จะมีค่าเป็นบวกบริเวณใกล้กึ่งกลาง และ จะมีค่าเป็นลบถ้าอยู่บริเวณรอบนอก เช่น

$$h_{rect}(x, y) = \frac{1}{9} \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad (4.21)$$

รูป 4.10 เป็นตัวอย่างภาพ และ หน้ากากตัวกรองความถี่สูงอย่างง่ายขนาด 3x3 จะเห็นว่าค่าของแต่ละจุดภาพบนภาพผลลัพธ์

1. จะมีค่าของจุดภาพเท่ากับศูนย์ ( $= 0$ ) ถ้าบริเวณจุดภาพนั้นๆ ของภาพนำเข้า มีค่าเท่ากัน
2. จะมีค่าของจุดภาพมากกว่าศูนย์ ( $> 0$ ) ถ้าค่าของจุดภาพทางซ้ายน้อยกว่าค่าของจุดภาพทางขวา  $f(x-1, y) < f(x, y)$  หรือ ถ้าค่าของจุดภาพทางบนน้อยกว่าค่าของจุดภาพทางล่าง  $f(x, y-1) < f(x, y)$  และ
3. ในทางตรงกันข้าม จุดภาพบนภาพผลลัพธ์จะมีค่าของจุดภาพน้อยกว่าศูนย์  $< 0$  ถ้ามีการเปลี่ยนแปลงค่าจุดภาพของภาพนำเข้าจากมากไปน้อย เช่น  $f(x-1, y) > f(x, y)$  หรือ  $f(x, y-1) > f(x, y)$  และ ในทางปฏิบัติ ถ้าค่าของจุดภาพบนภาพผลลัพธ์มีค่าน้อยกว่าศูนย์ ให้ทำการเปลี่ยนค่าของคำตอบให้อยู่ในช่วง  $[0 - 255]$

### 4.9.2 การหาตัวกรองความถี่สูงด้วยวิธีการหาอนุพันธ์

จากที่นิสิตได้เรียนวิชาแคลคูลัส (Calculus) ในปีแรกของการเรียนในระดับมหาวิทยาลัย นิสิตหลายคนมีความสงสัยว่าเราจะสามารถนำวิชาแคลคูลัสไปใช้ได้เมื่อไหร่ ในบทนี้จะแสดงให้เห็นถึงการนำแคลคูลัสไปใช้ในการประมวลผลภาพ แคลคูลัส หรือ การหาอนุพันธ์ เป็นวิชาว่าด้วยการหาอัตราการเปลี่ยนแปลงของตัวแปรตามเทียบกับตัวแปรต้น หรือ ตัวแปรอิสระ ยกตัวอย่าง เช่น

$$y = x^2 \quad (4.22)$$

เมื่อ  $y$  คือ ตัวแปรตาม และ  $x$  คือ ตัวแปรอิสระ หรือ ตัวแปรต้น ถ้าเราต้องการทราบว่า เมื่อค่าของตัวแปรต้น  $x$  เปลี่ยนไปเท่ากับ  $\Delta x = x_2 - x_1$  แล้ว ค่าตัวแปรตาม  $y_2$  จะมีค่าเท่ากับเท่าใด ซึ่งเราจะสามารถคำนวณได้จากอัตราการเปลี่ยนแปลงของ  $y$  เทียบกับ  $x$  ดังนี้

$$\begin{aligned} \frac{dy}{dx} &= \frac{dx^2}{dx} \\ &= 2x \end{aligned} \quad (4.23)$$

นั่นคือ เมื่อ  $\Delta x$  เปลี่ยนไปหน่วยเท่ากับ 1 ค่าอัตราการเปลี่ยนแปลงของ  $y$  จะเป็น 2 เท่าของ  $x$

จากที่ได้กล่าวถึงวิชาแคลคูลัสมาข้างต้น จะเห็นได้ว่าเราสามารถนำการหาอนุพันธ์มาใช้ในการประมวลผลภาพได้ กล่าวคือ ภาพนำเข้า (Input Image) สำหรับการประมวลผลจะประกอบด้วยจุดภาพ (pixels หรือ picture elements) ขนาด  $M \times N$  จุดภาพ ซึ่งรายละเอียดเรื่องนี้ได้กล่าวไว้แล้วในบทที่ 1 ในแต่ละแถวของจุดภาพ ( $1 \leq i \leq M$ ) ค่าความเข้มสีหรือค่าระดับเทาของแต่ละจุดภาพจะมีค่าไม่เท่ากัน หรือ กล่าวอีกอย่างหนึ่งคือ จุดภาพสองจุดภาพที่อยู่ติดกันอาจจะมีค่าระดับเทาที่แตกต่างกัน หรือ มีการเปลี่ยนแปลงค่าระดับเทาเกิดขึ้นบนภาพ ดังนั้น เราจึงสามารถนำการหาอนุพันธ์บนภาพนำเข้าได้ โดยวิธีการที่นิยมใช้ในการหาอนุพันธ์คือ การหาค่า Gradient  $\nabla f$  โดย

$$\nabla f(x, y) = \begin{pmatrix} \frac{\partial f(x, y)}{\partial x} \\ \frac{\partial f(x, y)}{\partial y} \end{pmatrix} \quad (4.24)$$

ค่า Gradient  $\nabla f$  เป็นเวกเตอร์ ที่มีขนาดและทิศทางดังนี้

$$magnitude(\nabla f) = \sqrt{\frac{\partial f^2(x, y)}{\partial x} + \frac{\partial f^2(x, y)}{\partial y}} \quad (4.25)$$

และ

$$direction(\nabla f) = \tan^{-1} \frac{\partial f(x, y)}{\partial y} / \frac{\partial f(x, y)}{\partial x} \quad (4.26)$$

โดย ค่า Magnitude ของ  $\nabla f$  จะบอกถึงค่าระดับ หรือ อัตราของการเปลี่ยนแปลงของค่าระดับเทาของจุดภาพ และ ค่าทิศทางของ  $\nabla f$  จะมีทิศทางตั้งฉากกับแนวขอบของวัตถุบนภาพเสมอ

### การคำนวณค่า Gradient

การคำนวณค่า Gradient จะได้มาจากการประมาณความแตกต่างบนภาพ  $f(x, y)$  เมื่อ กำหนดให้  $f(x, y)$  เป็นค่าไม่ต่อเนื่อง (Discrete or interger value) ดังนี้

$$\frac{\partial f(x, y)}{\partial x} = \frac{f(x + \Delta x, y) - f(x, y)}{\Delta x} = f(x + 1, y) - f(x, y), (\Delta x = 1) \quad (4.27)$$

$$\frac{\partial f(x, y)}{\partial y} = \frac{f(x, y + \Delta y) - f(x, y)}{\Delta y} = f(x, y + 1) - f(x, y), (\Delta y = 1) \quad (4.28)$$

### ตัวอย่าง

$Z_1$	$Z_2$	$Z_3$
$Z_4$	$Z_5$	$Z_6$
$Z_7$	$Z_8$	$Z_9$

สมมติว่าต้องการหาค่า Gradient ของภาพ ณ ตำแหน่ง  $Z_5$  จะสามารถประมาณค่า Gradient ได้ ดังนี้

$$\frac{\partial f(x, y)}{\partial x} = Z_6 - Z_5 \quad (4.29)$$

$$\frac{\partial f(x, y)}{\partial y} = Z_8 - Z_5 \quad (4.30)$$

$$(4.31)$$

และ

$$\text{magnitude}(\nabla f) = \sqrt{(Z_6 - Z_5)^2 + (Z_8 - Z_5)^2} \quad (4.32)$$

นั่นคือ เราสามารถประมาณหน้ากากรวดสำหรับค่า Gradient สำหรับ  $\frac{\partial f(x, y)}{\partial x}$  และ  $\frac{\partial f(x, y)}{\partial y}$  ดังกล่าวได้ ดังนี้

$$\frac{\partial f(x, y)}{\partial x} = \begin{bmatrix} -1 & 1 \end{bmatrix} \quad (4.33)$$

และ

$$\frac{\partial f(x, y)}{\partial y} = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \quad (4.34)$$

### การประมาณค่า Gradient แบบอื่นๆ

กำหนดให้

$$\frac{\partial f(x, y)}{\partial x} = f(x, y) - f(x + 1, y + 1), (\Delta x = 1) \quad (4.35)$$

$$\frac{\partial f(x, y)}{\partial y} = f(x + 1, y) - f(x, y + 1), (\Delta y = 1) \quad (4.36)$$

และสำหรับภาพ  $f(x, y)$  ต่อไปนี้

$Z_1$	$Z_2$	$Z_3$
$Z_4$	$Z_5$	$Z_6$
$Z_7$	$Z_8$	$Z_9$

เราจะสามารถประมาณ ค่า Gradient ณ ตำแหน่ง  $Z_5$  ได้ดังนี้

$$\frac{\partial f(x, y)}{\partial x} = Z_5 - Z_9 \quad (4.37)$$

$$\frac{\partial f(x, y)}{\partial y} = Z_6 - Z_8 \quad (4.38)$$

$$(4.39)$$

และ จะสามารถเขียนเป็นหน้ากากตัวกรอง สำหรับ  $\frac{\partial f(x, y)}{\partial x}$  และ  $\frac{\partial f(x, y)}{\partial y}$  ดังกล่าวได้ ดังนี้ (Roberts Filter)

$$\frac{\partial f(x, y)}{\partial x} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad (4.40)$$

และ

$$\frac{\partial f(x, y)}{\partial y} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad (4.41)$$

### Prewitt Filter

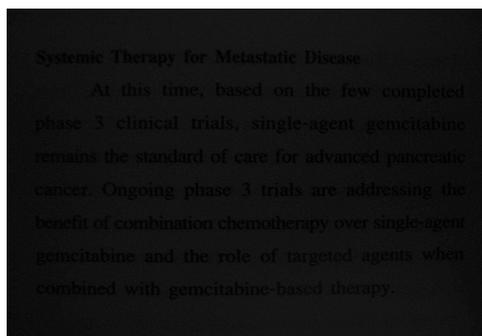
นอกจากนี้ ยังมีหน้ากากตัวกรองอื่นที่น่าสนใจซึ่งเกิดมาจากการหาอนุพันธ์ เช่น Prewitt Filter ดังนี้

-1	-1	-1	-1	0	1
0	0	0	-1	0	1
1	1	1	-1	0	1

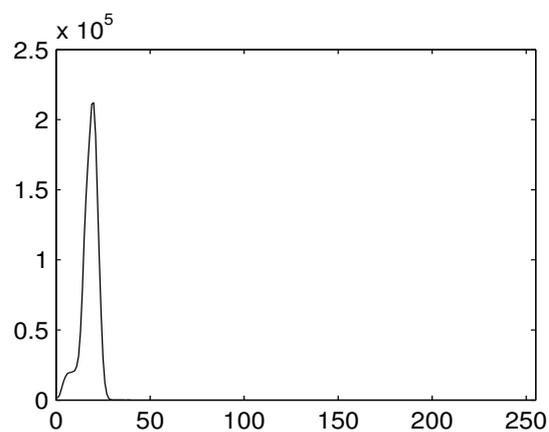
-1	-2	-1	-1	0	1
0	0	0	-2	0	2
1	2	1	-1	0	1

## 4.10 สรุป

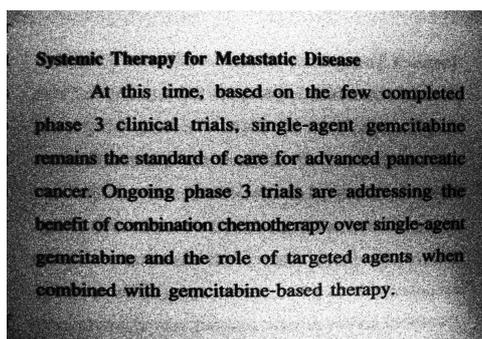
ในบทนี้ ได้กล่าวถึง การดำเนินการบนโดเมนภาพเชิงพื้นที่ และ เชิงความถี่ การปรับปรุงภาพด้วยฮิสโตแกรม (Histogram equalization) การหาผลต่างของภาพ Image Subtraction ตัวกรองปรับเรียบ (Smoothing Filters) ตัวกรองปรับสว่าง (Sharpen Filters) และ การหาตัวกรองความถี่สูงด้วยวิธีการหาอนุพันธ์ เนื้อหาทั้งหมดที่กล่าวมาเป็นบทนำสำหรับการปรับปรุงภาพ ซึ่งจะช่วยให้ผู้เรียนสามารถประยุกต์ใช้ต่อไปได้ในอนาคต



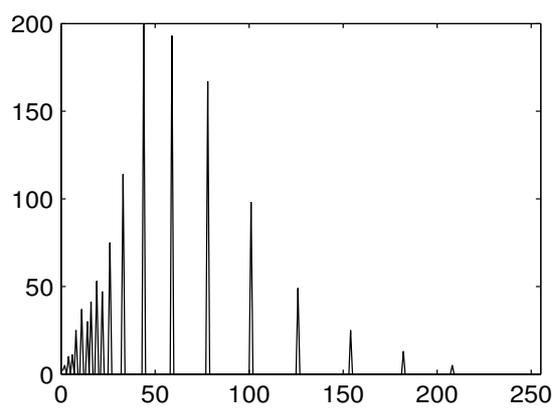
(a)



(b)

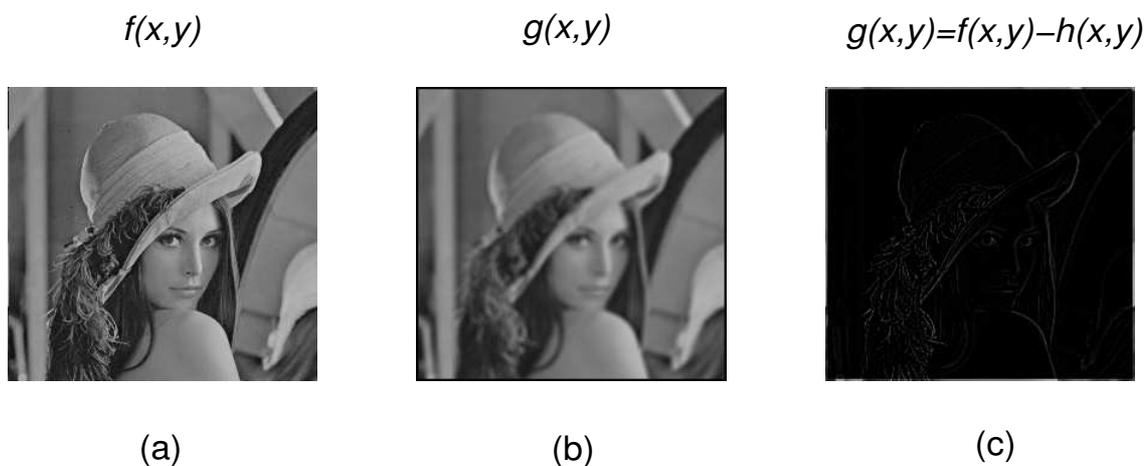


(c)

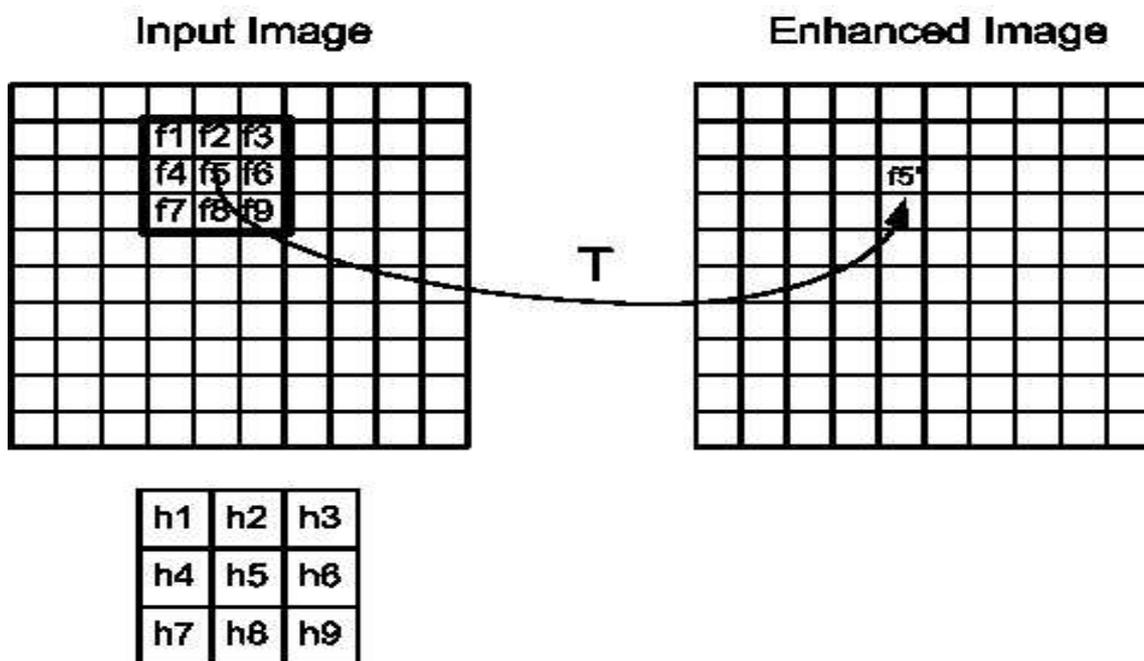


(d)

รูปที่ 4.5: ภาพและฮิสโตแกรมของภาพ (a) ภาพตัวหนังสือต้นฉบับ (b) ฮิสโตแกรมของภาพต้นฉบับ (c) ภาพหลังจากการทำ Histogram Equalization และ (d) ฮิสโตแกรมใหม่ของภาพ



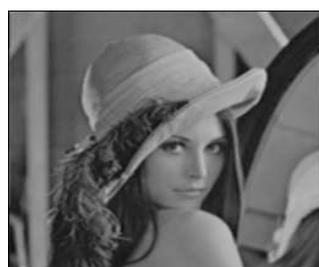
รูปที่ 4.6: การหาผลต่างภาพ (a) ภาพต้นฉบับ (b) ภาพต้นฉบับหลังจากผ่านการปรับเรียบ (Smoothing Filter) (c) ผลต่างระหว่างภาพทั้งสอง



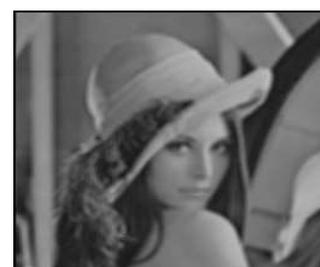
รูปที่ 4.7: การปรับปรุงภาพโดยการดำเนินการเชิงพื้นที่กับจุดภาพรอบข้าง



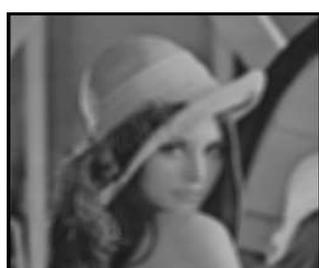
(a)



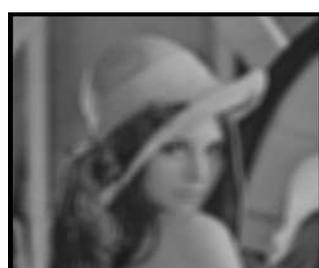
(b)



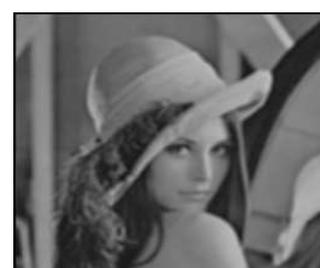
(c)



(d)

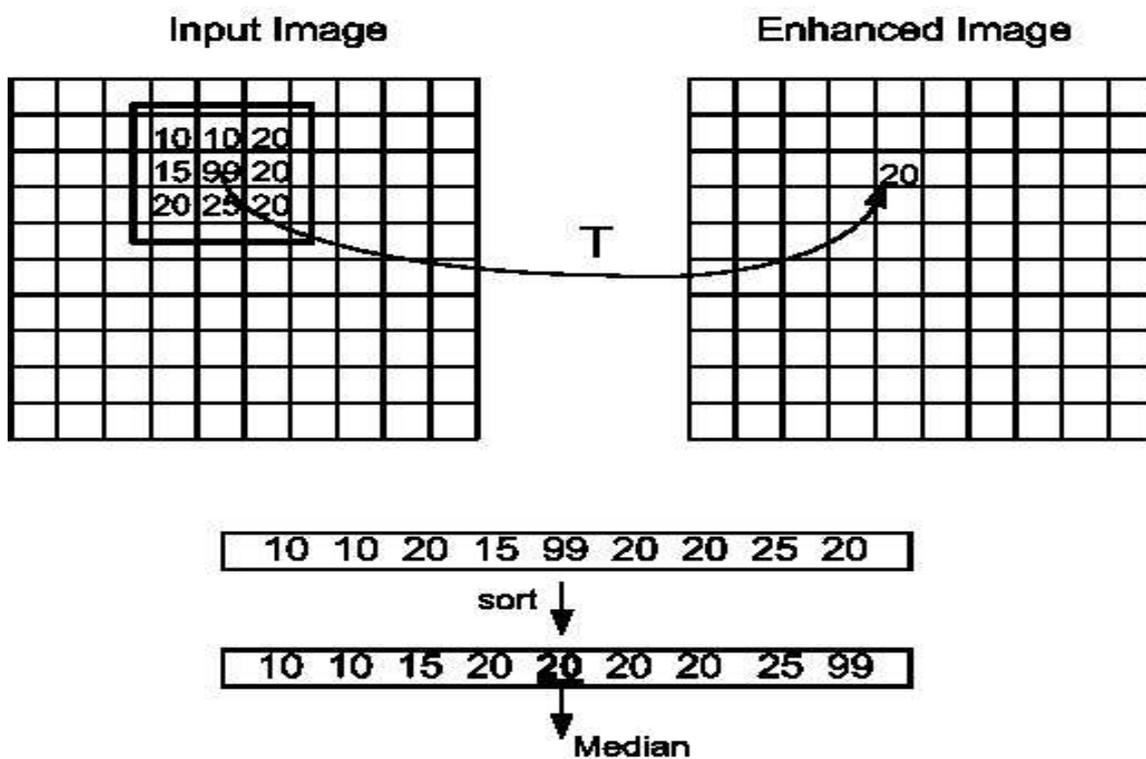


(e)

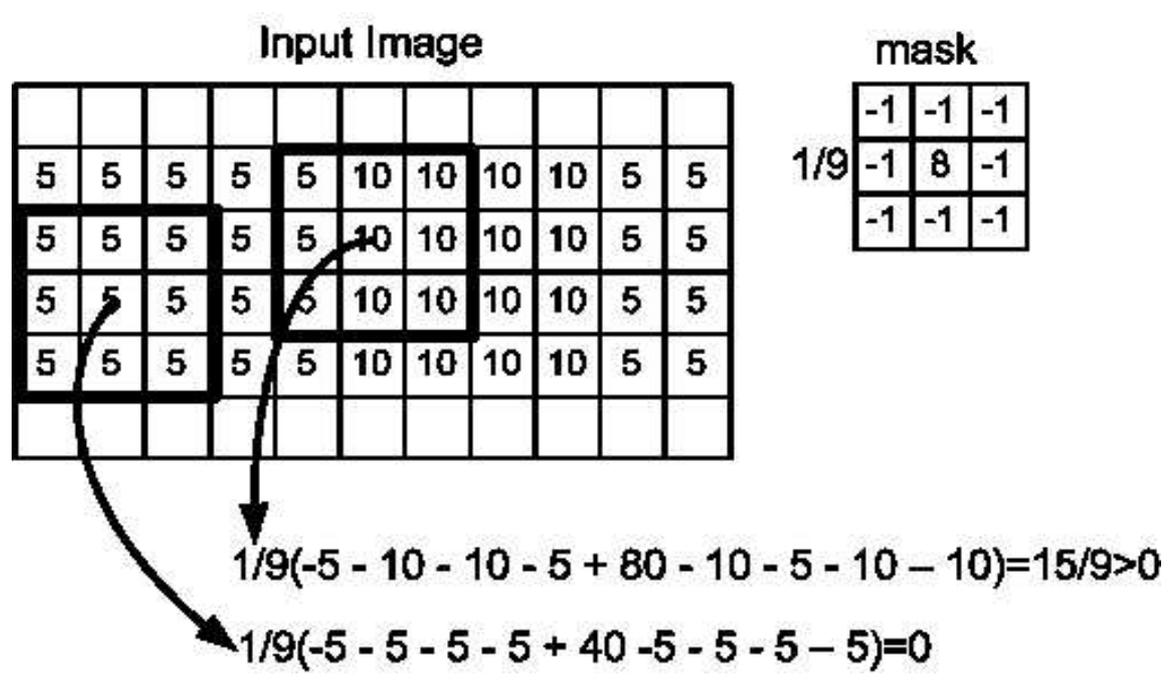


(f)

รูปที่ 4.8: การปรับเรียบเชิงเส้น (a) ภาพต้นฉบับ (b) ตัวกรองขนาด  $3 \times 3$  ค่าถ่วงน้ำหนักเท่ากัน (c) ตัวกรองขนาด  $5 \times 5$  ค่าถ่วงน้ำหนักเท่ากัน (d) ตัวกรองขนาด  $7 \times 7$  ค่าถ่วงน้ำหนักเท่ากัน (e) ตัวกรองขนาด  $9 \times 9$  ค่าถ่วงน้ำหนักเท่ากัน (f) ตัวกรองแบบวงกลมขนาด  $5 \times 5$  รัศมี 2.5 ตั้งสมการ 4.17



รูปที่ 4.9: การปรับเรียบแบบไม่เชิงเส้นสำหรับหน้าต่างขนาด 3x3



รูปที่ 4.10: การปรับปรุงภาพโดยตัวกรองความถี่สูงด้วยหน้ากาทัวกรองขนาด 3x3

Input Image



(a)

High-pass Image



(b)

รูปที่ 4.11: การปรับปรุงภาพ Lena ด้วยตัวกรองความถี่สูงในสมการ 4.21

## บทที่ 5

### การกู้คืนภาพ Image Restoration

#### เนื้อหาสาระ

1. Model the degradation process
2. Apply the inverse process to recover the original image

#### วัตถุประสงค์

1. อธิบายทฤษฎีการทำให้สีจางของข้อมูลภาพต้นฉบับ
2. อธิบายวิธีการกู้คืนภาพต้นฉบับ

#### กิจกรรมการเรียนการสอน

1. บรรยาย อภิปรายซักถาม และ ยกตัวอย่าง
2. สรุปเนื้อหารายวิชา

#### การวัดผล

1. ซักถามความเข้าใจหลังการเรียน

## การกู้คืนภาพ Image Restoration

### 5.1 บทนำ

การกู้คืนภาพ (Image Restoration) เป็นกระบวนการที่เกี่ยวข้องกับ การศึกษาวิธีการในการลบหรือการลดการเสื่อมของภาพ (Removal or reduction of degradation of images) ซึ่งเกิดขึ้นได้จากขั้นตอนการได้มาซึ่งภาพ เช่น สัญญาณรบกวนบนภาพ ภาพที่ได้มาไม่มีความคมชัด ซึ่งวิธีการที่จะกู้คืนนั้นจะทำได้ทั้งบนโดเมนเชิงพื้นที่ และ โดเมนเชิงความถี่ ซึ่งได้กล่าวไว้แล้วในบทก่อนหน้า สำหรับเอกสารประกอบการสอนนี้จะกล่าวถึงสองหัวข้อหลักๆ คือ ตัวแบบสำหรับการเสื่อมของภาพ และ กระบวนการในการกู้คืนภาพต้นฉบับ

### 5.2 ตัวแบบของการเสื่อมของภาพ Model of Image degradation

บนโดเมนเชิงพื้นที่ (Spatial Domain) มีข้อมูลภาพ  $f(x, y)$  ซึ่งถูกทำการคอนโวลูชัน (Convolution) กับตัวกรองเชิงพื้นที่  $h(x, y)$  ในบางครั้งจะได้ภาพผลลัพธ์ที่มีการเสื่อม ยกตัวอย่างเช่น ถ้า  $h(x, y)$  ประกอบด้วยข้อมูลแถวเดียว และมีค่าเป็นหนึ่งทั้งหมด ทำให้ภาพผลลัพธ์ที่ได้ จะออกมาเป็นภาพมัว และมีทิศทางมัวไปตามทิศทางของตัวกรองที่กำลังดำเนินการ ยกตัวอย่าง เช่น ตัวกรอง  $H[] = h(x, y)$  ในสมการ 5.1 เป็นตัวกรองที่มีข้อมูลแถวเดียวในทิศะแยงมุม เมื่อนำตัวดำเนินการกล่าวไปกระทำกับภาพ  $f(x, y)$  ทำให้ได้ภาพผลลัพธ์ที่มัว และมีทิศทางเยื้องไปในทิศะแยงมุมตามตัวกรอง  $h(x, y)$  ดังรูปที่ 5.1(c) แต่ถ้าภาพต้นฉบับถูกเพิ่มสัญญาณรบกวน ประเภท Salt-and-Pepper เข้าไปก็จะได้ผลลัพธ์ตามรูปที่ 5.1(b) เป็นต้น

$$h(x, y) = \frac{1}{5} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.1)$$

ดังนั้น จะสามารถเขียนเป็นสมการความสัมพันธ์ของการเสื่อมได้ดังนี้

$$g(x, y) = f(x, y) * h(x, y) \quad (5.2)$$

สำหรับการสีกกร่อนที่เกิดจากการคอนโวลูชันเพียงอย่างเดียว อย่างไรก็ตามสมการที่ 5.2 ไม่ได้ครอบคลุมทุกกรณี ยกตัวอย่างเช่น กรณีภาพสีกกร่อนในรูปที่ 5.1(b) ในภาพนั้นมีการเพิ่มของสัญญาณรบกวน (Noise) นั่นคือ เราจะเขียนสมการความสัมพันธ์ของการสีกกร่อนได้ดังนี้

$$g(x, y) = f(x, y) * h(x, y) + n(x, y) \quad (5.3)$$

จากที่ได้กล่าวมาเป็นลำดับ จะพบว่า การกู้คืนภาพข้อกำหนดที่น่าสนใจ คือ

1. วัตถุประสงค์ เพื่อคำนวณหาภาพต้นฉบับ  $f(x, y)$  จากภาพสีกกร่อน  $g(x, y)$
2. ข้อมูลที่กำหนดให้ คือ ภาพสีกกร่อน  $g(x, y)$  และ ตัวกรอง  $h(x, y)$
3. ข้อมูลที่โจทย์ ไม่กำหนดให้ คือ  $n(x, y)$

### 5.2.1 สมมติฐานของตัวกรอง (Assumption of $H[.] = h(x, y)$ )

#### Linear

กล่าวคือ

$$h(f_1 + f_2) = h(f_1) + h(f_2), h(kf) = kh(f) \quad (5.4)$$

#### Shift Invariant

กล่าวคือ

$$\text{IF } H[f(x, y)] = g(x, y) \text{ THEN } H[f(x - a, y - b)] = g(x - a, y - b) \quad (5.5)$$

เมื่อ  $H[.] = h(x, y)$  นั่นคือ ถ้ามีการเลื่อนตำแหน่งของข้อมูลนำเข้าจะทำให้ผลลัพธ์เลื่อนไปในระยะที่เท่ากันเสมอ

### 5.3 ตัวแบบของการสีกกร่อนของภาพแบบต่อเนื่อง

กำหนดให้  $f(x, y)$  เขียนได้ดังสมการ ต่อไปนี้

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(a, b) \delta(x - a, y - b) da db \quad (5.6)$$

คำนวณหา  $H[f(x, y)]$

$$\begin{aligned} H[f(x, y)] &= H\left[\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(a, b) \delta(x - a, y - b) da db\right] \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} H[f(a, b) \delta(x - a, y - b)] da db \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(a, b) H[\delta(x - a, y - b)] da db \end{aligned} \quad (5.7)$$

จากสมมติฐานของตัวกรองทากล่าวมาข้างต้น ถ้ากำหนดให้  $h(x, y)$  มีคุณสมบัติเป็น Shift Invariant ดังนั้น  $H[\delta(x, y)] = h(x, y) \Rightarrow H[\delta(x - a, y - b)] = h(x - a, y - b)$

ดังนั้น สำหรับการสีกกร่อนของภาพแบบต่อเนื่อง กรณีที่สัญญาณรบกวนไม่มีผลต่อการสีกกร่อน จะได้ตัวแบบดังสมการ

$$\begin{aligned} H[f(x, y)] &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(a, b) H[\delta(x - a, y - b)] da db \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(a, b) h(x - a, y - b) da db \\ g(x, y) &= f(x, y) * h(x, y) \end{aligned} \quad (5.8)$$

สำหรับการสีกกร่อนของภาพแบบต่อเนื่อง กรณีที่สัญญาณรบกวนมีผลต่อการสีกกร่อน จะได้ตัวแบบดังสมการ

$$g(x, y) = f(x, y) * h(x, y) + n(x, y) \quad (5.9)$$

### 5.4 ตัวแบบของการสีกกร่อนของภาพแบบไม่ต่อเนื่อง

กำหนดให้  $f(x, y)$  และ  $h(x, y)$  เป็นฟังก์ชันไม่ต่อเนื่อง จะได้ว่า

$$g_d(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f_d(m, n) h_d(x - m, y - n) + n_d(x, y) \quad (5.10)$$

ซึ่งเราสามารถเขียนอธิบายการคอมโวลูชัน ในสมการ 5.10 ในรูปของเมตริกซ์ 1 มิติ ได้ดัง

$$g_d(x) = \sum_{m=0}^{M-1} f_d(m) h_d(x - m) + n_d(x) \quad (5.11)$$

ดังนั้น

$$\begin{aligned} g_d(0) &= f_d(0)h_d(0) + f_d(1)h_d(-1) + \cdots + f_d(M-1)h_d(-M+1) + n_d(0) \\ g_d(1) &= f_d(0)h_d(1) + f_d(1)h_d(0) + \cdots + f_d(M-1)h_d(-M+2) + n_d(1) \\ &\vdots \\ g_d(M-1) &= f_d(0)h_d(M-1) + f_d(1)h_d(M-2) + \cdots + f_d(M-1)h_d(0) + n_d(M-1) \end{aligned} \quad (5.12)$$

หรือ ในรูปของเมตริกซ์

$$\begin{aligned} \begin{bmatrix} g_d(0) \\ g_d(1) \\ \vdots \\ g_d(M-1) \end{bmatrix} &= \begin{bmatrix} h_d(0) & h_d(-1) & \cdots & h_d(-M+1) \\ h_d(1) & h_d(0) & \cdots & h_d(-M+2) \\ \vdots & \vdots & \ddots & \vdots \\ h_d(M-1) & h_d(M-2) & \cdots & h_d(0) \end{bmatrix} \begin{bmatrix} f_d(0) \\ f_d(1) \\ \vdots \\ f_d(M-1) \end{bmatrix} \\ &+ \begin{bmatrix} n_d(0) \\ n_d(1) \\ \vdots \\ n_d(M-1) \end{bmatrix} \end{aligned} \quad (5.13)$$

หรือ

$$g = \mathbf{H}f + n \quad (5.14)$$

## 5.5 การคำนวณหาภาพต้นฉบับ $f(x, y)$

การหาภาพต้นฉบับ  $f(x, y)$  จากภาพสีกร่อน  $g(x, y)$  นั้น ทำได้หลายวิธี เช่น

1. คำนวณหา  $f(x, y)$  จาก  $f = H^{-1}g$  ในวิชาพีชคณิตเชิงเส้น (Linear Algebra) แต่เป็นวิธีที่มีความยุ่งยาก และ ซับซ้อน
2. คำนวณจากการประมาณ Eigenvector ซึ่งสะดวกและรวดเร็วกว่า หรือ เรียกว่า Principal Component Analysis

### 5.5.1 การกู้คืนภาพด้วยเทคนิค Principal Component Analysis: PCA

ประกอบด้วยขั้นตอนสำคัญๆ ดังนี้

#### Step 1.

ทำเมตริกซ์  $H$  ให้เป็นเมตริกซ์ทแยงมุม โดย  $H = VDV^{-1}$  เมื่อ  $D$  เป็นเมตริกซ์ทแยงมุม และ คำนวณหา ค่าไอเกน  $\lambda(k)$  และ ไอเกนเวกเตอร์  $v(k)$  ของ  $H$  โดยมีคุณสมบัติ คือ

$$Hv(k) = \lambda(k)v(k)$$

ถ้ากำหนดให้

$$V = \begin{bmatrix} v(0) & v(1) & \cdots & v(M-1) \end{bmatrix} \quad (5.15)$$

ดังนั้น

$$H = VDV^{-1} = V \begin{bmatrix} \lambda(0) & 0 & \cdots & 0 \\ 0 & \lambda(1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \lambda(M-1) \end{bmatrix} V^{-1} \quad (5.16)$$

เมื่อ

$$V^{-1} = \frac{1}{M}[v^*(0), v^*(1), \dots, v^*(M-1)] \text{ และ } v^*(0) \text{ คือ Conjugate ของ } v(0)$$

**Step 2**

เขียนแทนสมการ  $g = Hf$  ใหม่โดยใช้  $H = VDV^{-1}$

$$\begin{aligned} g &= Hf \\ &= VDV^{-1}f \end{aligned} \quad (5.17)$$

ดังนั้น

$$\begin{aligned} f &= H^{-1}g \\ &= V^{-1}D^{-1}Vg \end{aligned} \quad (5.18)$$

## 5.6 บทประยุกต์: การกู้คืนภาพต้นฉบับด้วยเทคนิค PCA เมื่อไม่ทราบค่า $H[.]$

ในกรณีที่ทราบค่าเมตริกซ์  $H[.]$  โดยเราทราบเพียงภาพสีกร่อน  $g(x, y)$  เราสามารถประมาณ  $f(x, y)$  จาก ค่าไอเกน และ ไอเกนเวกเตอร์ ของ  $g(x, y)$  ตามขั้นตอน ดังต่อไปนี้ [11]:

1. Transform  $g(x, y)$  into row major
2. Compute the mean for each row  $mean_i$
3. Generate the zero-mean matrix  $z$  for each row  $i$ ,

$$z(x, i) = g(x, i) - mean_i \quad (5.19)$$

where  $1 \leq i \leq M$ .

4. Compute the covariance matrix **Cov** for  $z$ ,

$$\mathbf{Cov}(z) = E[zz^T] \quad (5.20)$$

where  $E[.]$  is an expectation value and  $z^T$  is a transpose matrix of  $z$ . Then the dimensions of  $\mathbf{Cov}(z)$  are  $N$  by  $N$ .

5. Compute an eigenvalue,  $D$ , and eigenvector,  $V$ , for the covariance matrix **Cov**.

6. Project  $Z$  onto new axis by eigen-vector  $V$ ,

$$P = v^T z \quad (5.21)$$

where  $V^T$  is a transpose matrix of  $V$

จากขั้นตอนวิธีที่นำเสนอ เมื่อสิ้นสุดกระบวนการ จะได้  $P$  ซึ่งเป็นภาพที่กู้คืนจากภาพที่ ลีกรหรือ ดังตัวอย่างในรูปที่ 5.4

จากรูปที่ 5.4 กำหนดให้  $S$  ในสดมภ์ที่หนึ่ง เป็นภาพของสัญญาณเสียงต้นฉบับจำนวน 3 เสียง ซึ่งเป็นไปตามสมการ 5.31 ในสดมภ์ที่ 2 เป็นภาพของสัญญาณเสียงที่ได้รับการผสม เข้ามา  $X = AS$  จำนวน 3 เสียง จะเห็นได้ว่าในภาพของสัญญาณเสียงในสดมภ์ที่ 2 ทั้ง 3 ภาพนั้น มีลักษณะโครงร่างที่เหมือนกัน ไม่สามารถแยกหรือเข้าใจได้ว่าเสียงอะไรบ้าง ส่วนใน สดมภ์ที่ 3 เป็นภาพของสัญญาณเสียงที่ได้รับการกู้คืนจำนวน 3 เสียง ซึ่งพิจารณาแล้วมีความ คล้ายกับเสียงต้นฉบับ เพียงแต่อาจมีการหมุนเปลี่ยนสลับตำแหน่งของสัญญาณเท่านั้น

สมมติ ในปัญหาหนึ่งที่กำลังสนใจ ได้มีสัญญาณที่ถูกผสมมาแล้ว  $X$  และ บอกเพียงว่า  $X = AS$  แต่ในปัญหาดังกล่าว เราจะไม่ทราบค่าฟังก์ชันการแจกแจงความน่าจะเป็นของ  $S$  และ ไม่มีการกำหนดค่าเมตริกซ์การรวม  $A$  มาให้ กล่าวอีกอย่างคือ ไม่ทราบค่า  $S$  และ  $A$  (Unknown system) ในทางปฏิบัติ ยกตัวอย่างในความเป็นจริงของปัญหานี้ คือ การแยก เสียงของสมองมนุษย์ ซึ่งจะทราบกันดีว่าในความเป็นจริง เสียงแต่ละเสียงที่มนุษย์เราได้ยิน มานั้น มนุษย์เราจะไม่ทราบค่า ฟังก์ชันการแจกแจงความน่าจะเป็นของแต่ละเสียง  $s_i$  และ ยิ่ง ไปกว่านั้น เสียงแต่ละเสียงเกิดขึ้นในตำแหน่งที่ต่างกัน (ต่างกันทั้งระยะซัด และ ทิศทาง) ทำให้เราได้ยินเสียงแต่ละเสียงได้ชัดไม่เท่ากัน ปัจจัยในเรื่องของตำแหน่งการเกิดของเสียง ก็คือ เมตริกซ์การรวม  $A$  นั่นเอง ดังนั้น เสียงที่มนุษย์เราได้ยิน คือ เสียงที่ได้รับการผสมมาแล้ว คือ  $X$  และ เป็นหน้าที่ของสมองในการแยกหรือการกู้คืนเสียงต้นฉบับ  $S$  เพื่อจะได้รับรู้ว่าเสียง ที่เรากำลังได้ยินนั้นหมายถึงอะไร คำถาม ก็คือ ภายในสมองของเรานั้น กู้คืนสัญญาณที่เรา ได้ยินได้อย่างไร และ เราสามารถสร้างตัวแบบทางคณิตศาสตร์เพื่อกู้คืนสัญญาณที่เราได้ยินมา นั้นได้หรือไม่

กำหนดให้

$$[S] = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{cases} 0.1 * \sin(400 * t) * \cos(30 * t); \\ 0.01 * \sin(\sin(500 * t + 9 * \cos(40 * t))); \\ 0.0001 * \text{rand}(1, \text{length}(t)) - 1; \end{cases} \quad (5.22)$$

และ

$$A = \begin{bmatrix} 0.9775 & 0.6187 & 0.5531 \\ 0.9845 & 0.6292 & 0.4508 \\ 0.6927 & 0.0615 & 0.2504 \end{bmatrix} \quad (5.23)$$

จากสมการการแปลงเชิงเส้น (Linear Transformation)

$$X = AS \quad (5.24)$$

ผลจากการแปลงเชิงเส้น จะได้ภาพของสัญญาณเสียง 3 ภาพ ดังแสดงในสไลด์ที่ 2 ของรูปที่ 5.4 จากที่กล่าวมา วัตถุประสงค์ของปัญหานี้คือ ต้องการให้หาวิธีการในการกู้คืนสัญญาณ  $S$  จากสัญญาณที่ได้รับมาคือ  $X$  ดังแสดงในสไลด์ที่ 3 ของรูปที่ 5.4 ปัญหานี้ เราเรียกว่า **ปัญหาการแยกต้นฉบับกลับโดยไม่รู้แหล่งที่มา (Blind Source Separation)** โดยวิธีการกู้คืนภาพต้นฉบับ  $S$  สามารถสร้างเป็นสมการความสัมพันธ์ได้ดังนี้

$$X = AS$$

ถ้าต้องการกู้คืน  $S$  จะต้องประมาณ  $W$  เพื่อ

$$\tilde{S} = WX = WAS \quad (5.25)$$

ถ้า สามารถประมาณหา  $W$  ซึ่งเป็น pseudo-inverse ของ  $A$  จะได้

$$\tilde{S} = WX = WAS = A^{-1}AS = IS = S \quad (5.26)$$

เมื่อ  $I$  คือ เมทริกซ์เอกลักษณ์ และ  $W = A^{-1}$  คือ pseudo-inverse ของ  $A$  จากการศึกษาพบว่า ถ้า  $s_i$  ตั้งฉากกับ  $s_j$  จะได้ว่าไอเกนเวกเตอร์ของ  $X$  จะเป็น pseudo-inverse ของ  $A$  แต่ถ้า  $s_i$  ไม่ตั้งฉากกับ  $s_j$  ไอเกนเวกเตอร์ของ  $X$  จะเป็นตัวประมวลผลขั้นต้นที่เหมาะสม (preprocessing) สำหรับการหมุนแกนของสัญญาณ  $X$  ให้กลับมาอยู่ในทิศทางที่ควรจะเป็น จากนั้นก็จะสามารถใช้การประมวลผลขั้นต่อนอื่นที่มีความซับซ้อนสูงกว่าเพื่อแยกสัญญาณกลับต่อไป เช่น High-order statistics , Kullback-Leibler divergence เป็นต้น

$$\begin{aligned} f &= H^{-1}g \\ &= V^{-1}D^{-1}Vg \end{aligned} \quad (5.27)$$

## 5.7 บทประยุกต์: การกู้คืนภาพต้นฉบับด้วยเทคนิค PCA เมื่อไม่ทราบค่า $H[.]$

ในกรณีที่ไมทราบค่าเมตริกซ์  $H[.]$  โดยเราทราบเพียงภาพสีกกร่อน  $g(x, y)$  เราสามารถประมาณ  $f(x, y)$  จาก ค่าไอเกน และ ไอเกนเวคเตอร์ของ  $g(x, y)$  ตามขั้นตอน ดังต่อไปนี้ [11]:

1. Transform  $g(x, y)$  into row major
2. Compute the mean for each row  $mean_i$
3. Generate the zero-mean matrix  $z$  for each row  $i$ ,

$$z(x, i) = g(x, i) - mean_i \quad (5.28)$$

where  $1 \leq i \leq M$ .

4. Compute the covariance matrix  $Cov$  for  $z$ ,

$$Cov(z) = E[zz^T] \quad (5.29)$$

where  $E[.]$  is an expectation value and  $z^T$  is a transpose matrix of  $z$ . Then the dimensions of  $Cov(z)$  are  $N$  by  $N$ .

5. Compute an eigenvalue,  $D$ , and eigenvector,  $V$ , for the covariance matrix  $Cov$ .
6. Project  $Z$  onto new axis by eigen-vector  $V$ ,

$$P = v^T z \quad (5.30)$$

where  $V^T$  is a transpose matrix of  $V$

จากขั้นตอนวิธีที่นำเสนอ เมื่อสิ้นสุดกระบวนการ จะได้  $P$  ซึ่งเป็นภาพที่กู้คืนจากภาพที่ สีกหรือ ดังตัวอย่างในรูปที่ 5.4

จากรูปที่ 5.4 กำหนดให้  $S$  ในสดมภ์ที่หนึ่ง เป็นภาพของสัญญาณเสียงต้นฉบับจำนวน 3 เสียง ซึ่งเป็นไปตามสมการ 5.31 ในสดมภ์ที่ 2 เป็นภาพของสัญญาณเสียงที่ได้รับการผสม เข้ามา  $X = AS$  จำนวน 3 เสียง จะเห็นได้ว่าในภาพของสัญญาณเสียงในสดมภ์ที่ 2 ทั้ง 3 ภาพนั้น มีลักษณะโครงร่างที่เหมือนกัน ไม่สามารถแยกหรือเข้าใจได้ว่าเสียงอะไรบ้าง ส่วนใน

สดมภ์ที่ 3 เป็นภาพของสัญญาณเสียงที่ได้รับการสุ่มจำนวน 3 เสียง ซึ่งพิจารณาแล้วมีความคล้ายกับเสียงต้นฉบับ เพียงแต่อาจมีการหมุนเปลี่ยนสลับตำแหน่งของสัญญาณเท่านั้น

สมมติ ในปัญหาหนึ่งที่กำลังสนใจ ได้มีสัญญาณที่ถูกผสมมาแล้ว  $X$  และ บอกเพียงว่า  $X = AS$  แต่ในปัญหาดังกล่าว เราจะไม่ทราบค่าฟังก์ชันการแจกแจงความน่าจะเป็นของ  $S$  และ ไม่มีการกำหนดค่าเมตริกซ์การรวม  $A$  มาให้ กล่าวอีกอย่างคือ ไม่ทราบค่า  $S$  และ  $A$  (Unknown system) ในทางปฏิบัติ ยกตัวอย่างในความเป็นจริงของปัญหานี้ คือ การแยกเสียงของสมอมนุษย์ ซึ่งจะทราบกันดีว่าในความเป็นจริง เสียงแต่ละเสียงที่มนุษย์เราได้ยินมานั้น มนุษย์เราจะไม่ทราบค่า ฟังก์ชันการแจกแจงความน่าจะเป็นของแต่ละเสียง  $s_i$  และ ยิ่งไปกว่านั้น เสียงแต่ละเสียงเกิดขึ้นในตำแหน่งที่ต่างกัน (ต่างกันทั้งระยะซัด และ ทิศทาง) ทำให้เราได้ยินเสียงแต่ละเสียงได้ชัดไม่เท่ากัน ปัจจัยในเรื่องของตำแหน่งการเกิดของเสียง ก็คือเมตริกซ์การรวม  $A$  นั่นเอง ดังนั้น เสียงที่มนุษย์เราได้ยิน คือ เสียงที่ได้รับการผสมมาแล้ว คือ  $X$  และ เป็นหน้าที่ของสมอในการแยกหรือการสุ่มเสียงต้นฉบับ  $S$  เพื่อจะได้รับรู้ว่าเสียงที่เรากำลังได้ยินนั้นหมายถึงอะไร คำถาม ก็คือ ภายในสมอของเรานั้น ผู้คืนสัญญาณที่เราได้ยินได้อย่างไร และ เราสามารถสร้างตัวแบบทางคณิตศาสตร์เพื่อผู้คืนสัญญาณที่เราได้ยินมานั้นได้หรือไม่ กำหนดให้

$$[S] = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{cases} 0.1 * \sin(400 * t) * \cos(30 * t); \\ 0.01 * \text{sign}(\sin(500 * t + 9 * \cos(40 * t))); \\ 0.0001 * \text{rand}(1, \text{length}(t)) - 1; \end{cases} \quad (5.31)$$

และ

$$A = \begin{bmatrix} 0.9775 & 0.6187 & 0.5531 \\ 0.9845 & 0.6292 & 0.4508 \\ 0.6927 & 0.0615 & 0.2504 \end{bmatrix} \quad (5.32)$$

จากสมการการแปลงเชิงเส้น (Linear Transformation)

$$X = AS \quad (5.33)$$

ผลจากการแปลงเชิงเส้น จะได้ภาพของสัญญาณเสียง 3 ภาพ ดังแสดงในสดมภ์ที่ 2 ของรูปที่ 5.4 จากที่กล่าวมา วัตถุประสงค์ของปัญหานี้คือ ต้องการให้หาวิธีการในการสุ่มสัญญาณ  $S$  จากสัญญาณที่ได้รับมาคือ  $X$  ดังแสดงในสดมภ์ที่ 3 ของรูปที่ 5.4 ปัญหานี้ เราเรียกว่า ปัญหา

การแยกต้นฉบับกลับโดยไม่รู้แหล่งที่มา (Blind Source Separation) โดยวิธีการกู้คืนภาพต้นฉบับ  $S$  สามารถสร้างเป็นสมการความสัมพันธ์ได้ดังนี้ กำหนดให้

$$X = AS$$

ถ้าต้องการกู้คืน  $S$  จะต้องประมาณ  $W$  เพื่อ

$$\tilde{S} = WX = WAS \quad (5.34)$$

ถ้า สามารถประมาณหา  $W$  ซึ่งเป็น pseudo-inverse ของ  $A$  จะได้

$$\tilde{S} = WX = WAS = A^{-1}AS = IS = S \quad (5.35)$$

เมื่อ  $I$  คือ เมทริกซ์เอกลักษณ์ และ  $W = A^{-1}$  คือ pseudo-inverse ของ  $A$  จากการศึกษาพบว่า ถ้า  $s_i$  ตั้งฉากกับ  $s_j$  จะได้ว่าไอเกนเวกเตอร์ของ  $X$  จะเป็น pseudo-inverse ของ  $A$  แต่ถ้า  $s_i$  ไม่ตั้งฉากกับ  $s_j$  ไอเกนเวกเตอร์ของ  $X$  จะเป็นตัวประมวลผลขั้นต้นที่เหมาะสม (preprocessing) สำหรับการหมุนแกนของสัญญาณ  $X$  ให้กลับมาอยู่ในทิศทางที่ควรจะเป็น จากนั้นก็จะสามารถใช้การประมวลผลขั้นต่อนอื่นที่มีความซับซ้อนสูงกว่าเพื่อแยกสัญญาณกลับต่อไป เช่น High-order statistics, KL-divergence เป็นต้น



(a)

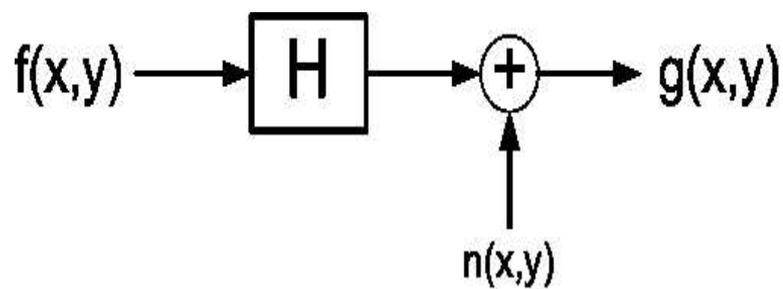


(b)

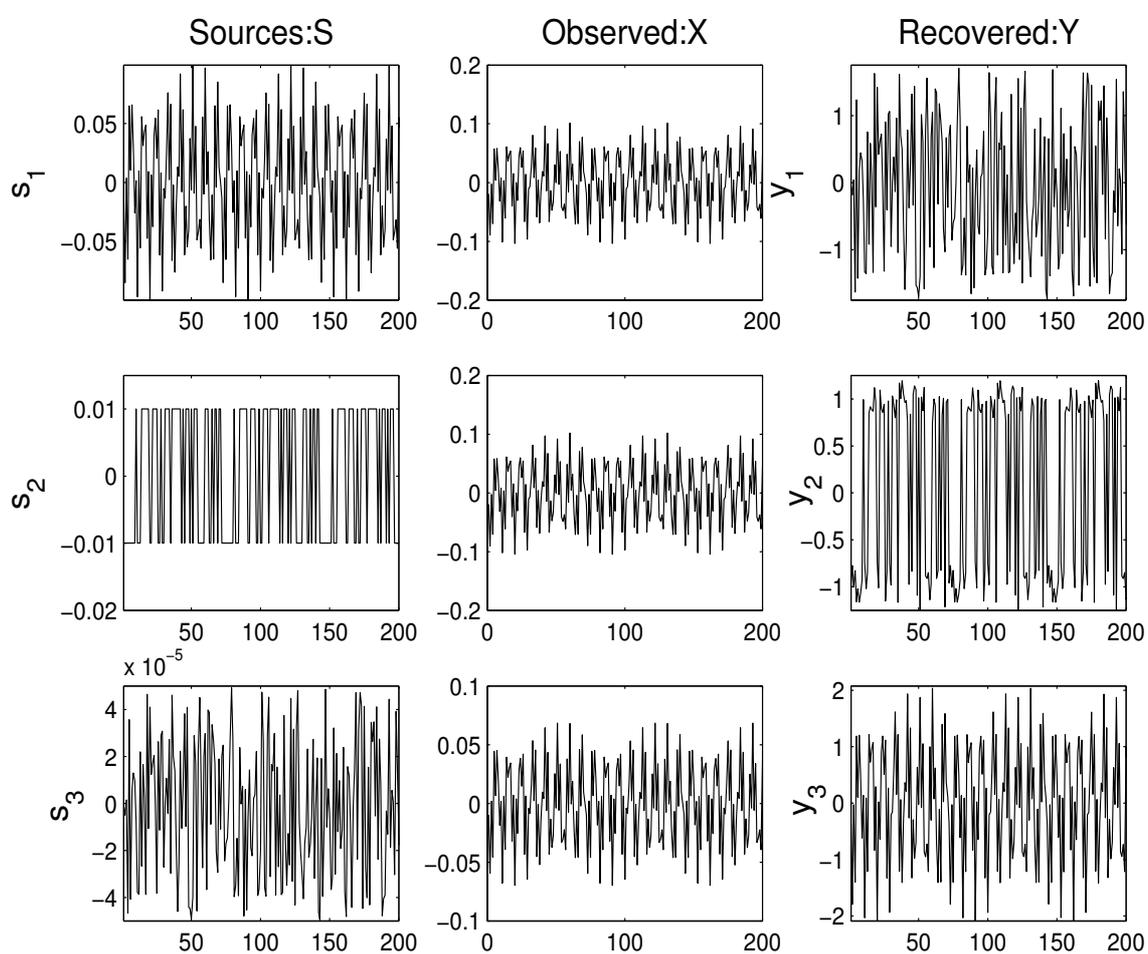


(c)

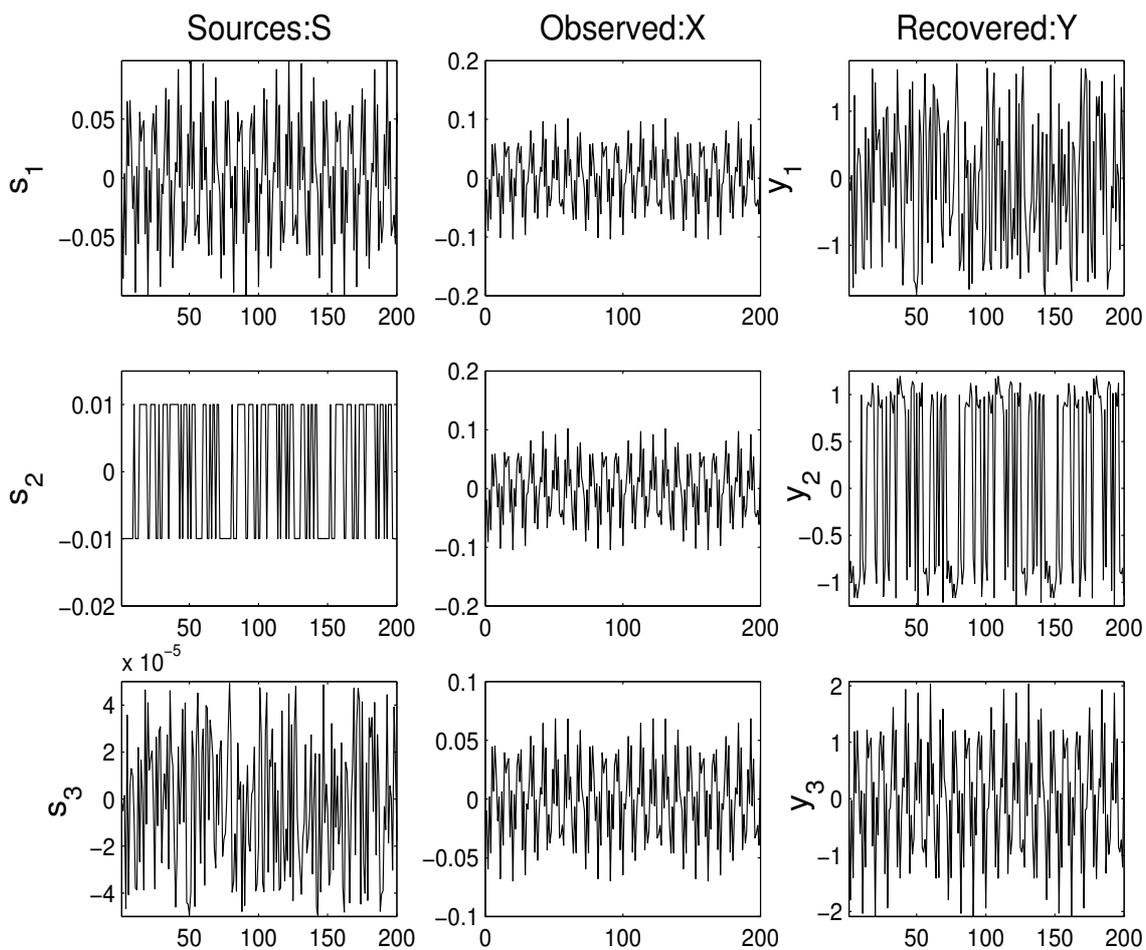
รูปที่ 5.1: สัญญาณรบกวนบนภาพ (a) ภาพต้นฉบับ (b) เพิ่มสัญญาณรบกวนแบบ Salt-and-Pepper และ (c) ภาพมั่วจากตัวกรองในสมการ 5.1



รูปที่ 5.2: ตัวแบบของการสีกกร่อนของภาพ



รูปที่ 5.3: การประยุกต์ใช้ PCA ในการกู้คืนภาพแบบไม่รู้ภาพต้นฉบับ



รูปที่ 5.4: การประยุกต์ใช้ PCA ในการกู้คืนภาพแบบไม่ระบุภาพต้นฉบับ

## บทที่ 6

### การแยกภาพ และ การหาขอบวัตถุในภาพ Image Segmentation และ Edge Detection

#### เนื้อหาสาระ

1. วัตถุประสงค์ของการแยกภาพ
2. วิธีการสำหรับการแยกภาพ
3. วิธีการสำหรับการหาขอบวัตถุในภาพ

#### วัตถุประสงค์

1. อธิบายทฤษฎีหลักการแยกภาพ และ การหาขอบวัตถุในภาพ
2. อธิบายวิธีเขียนโปรแกรมสำหรับ การแยกภาพ และ การหาขอบวัตถุในภาพ

#### กิจกรรมการเรียนการสอน

1. บรรยาย อภิปรายซักถาม
2. ยกตัวอย่าง
3. สรุปเนื้อหารายวิชา

#### การวัดผล

1. ซักถามความเข้าใจหลังการเรียน

## การแยกภาพ และการหาขอบวัตถุในภาพ Image Segmentation และ Edge Detection

### 6.1 บทนำ

การแยกภาพ และการหาขอบวัตถุในภาพ มีวัตถุประสงค์เพื่อสกัดสารสนเทศจากภาพนำเข้า และ สารสนเทศที่ได้บนภาพผลลัพธ์จะต้องน้อยกว่าสารสนเทศที่ปรากฏอยู่บนภาพนำเข้า แต่ สารสนเทศที่สกัดออกมาได้นั้น จะต้องมีความสัมพันธ์กัน เพื่อให้สามารถนำไปใช้เป็นข้อมูล นำเข้าในโมดูลอื่น ที่เกี่ยวข้องในการประมวลผลภาพ เช่นการตรวจจับวัตถุเคลื่อนไหวในภาพ เป็นต้น กล่าวอีกอย่างหนึ่งคือ การแยกภาพ และการหาขอบวัตถุในภาพ เป็นการประมวลผลขั้นต้น (preprocessing) สำหรับการประมวลผลภาพ

### 6.2 วัตถุประสงค์และความจำเป็นการแยกภาพ

1. วัตถุประสงค์หลักของการแยกภาพ คือ การแบ่งข้อมูลภาพออกเป็นส่วนๆ เช่น ส่วนของวัตถุที่อยู่ในภาพ (Objects) และ ส่วนของภาพพื้นหลัง (Background) ดังรูป ที่ 6.1 เป็นภาพของกรรไกร และ ประแจ อยู่บนพื้นหลัง ซึ่งก่อนจะทำการประมวลผลภาพ ส่วนของกรรไกร และ ประแจ จะต้องได้รับการแยกออกมาจากส่วนพื้นหลังก่อน
2. ผลลัพธ์ที่ได้จากขั้นตอนการแยกภาพ มีความสำคัญต่อประสิทธิภาพของการวิเคราะห์ภาพ (Image Analysis) กล่าวคือ ถ้าสามารถแยกส่วนของข้อมูลภาพได้อย่างถูกต้อง ร้อยละของความถูกต้องของกระบวนการวิเคราะห์ภาพ (เช่น การรู้จำภาพ) ก็จะสูงขึ้นตาม
3. โดยปกติ ขั้นตอนการแยกภาพ เป็นขั้นตอนที่มีความยากและซับซ้อนมาก ยกตัวอย่าง ในรูปที่ 6.2 ให้พิจารณาว่าเป็นรูปอะไร ดังนั้น ถ้าต้องการเพิ่มความถูกต้องให้การแยกภาพ สามารถทำได้โดย
  - (a) ต้องให้ความรู้พื้นฐานที่เพียงพอกับงานที่กำลังดำเนินงานอยู่
  - (b) ต้องมีการควบคุมสภาพแวดล้อม เช่น ในโรงงานอุตสาหกรรม
  - (c) ต้องเลือกอุปกรณ์ตรวจจับ (Sensors) ที่เหมาะสมกับการใช้งาน



รูปที่ 6.1: วัตถุบนภาพพื้นหลัง

### 6.3 วิธีการการแยกภาพ: Segmentation Method

1. วิธีการหาขอบ (Edge-based approaches)
  - Use the boundaries of regions to segment the image
  - Detect abrupt changes in intensity (discontinuities)
2. วิธีหาพื้นที่ (Region-based approaches)
  - Use similarity among pixels to find different regions

ในทางทฤษฎี วิธีการแยกภาพทั้งสองให้ผลลัพธ์ที่มีประสิทธิภาพเท่ากัน แต่ในทางปฏิบัติแล้ว ผลลัพธ์อาจแตกต่างกันบ้าง รูปที่ 6.3 ทางซ้ายมือ แสดงภาพต้นฉบับซึ่งมีภาพของบ้านและต้นไม้ และ ทางขวามือ คือ บริเวณของขอบของวัตถุในภาพ

### 6.4 วิธีการหาขอบวัตถุในภาพ (Edge-based approaches)

#### 6.4.1 ขั้นตอนในการหาขอบวัตถุในภาพ

1. Filtering (smoothing - noise reduction)
2. Enhancement (gradient magnitude)



รูปที่ 6.2: สุนัขลายจุดบนภาพพื้นหลัง

3. Detection (thresholding)
4. Localization (location, orientation)

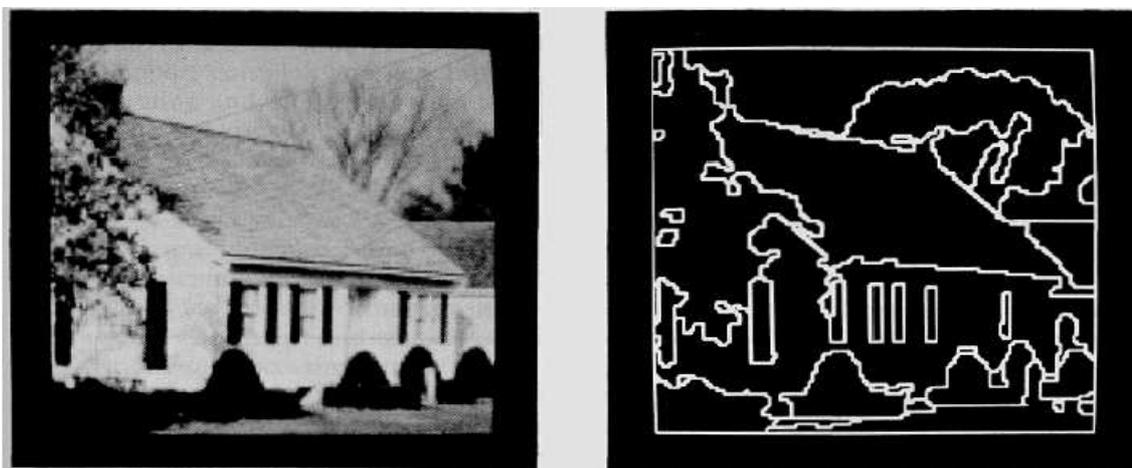
โดยการกรอง (Filtering) และการปรับปรุงภาพ (Enhancement) ได้กล่าวไปแล้วในบทก่อนหน้า ในบทนี้จะกล่าวถึงการตรวจจับ (Detection) เมื่อตรวจจับพบวัตถุได้จะทำให้เราได้ตำแหน่ง (Location) ของวัตถุนั้นภาพในทันที ซึ่งเราก็จะสามารถแยกวัตถุออกมาได้

#### 6.4.2 การหาจุดภาพ (Point Detection)

แนวคิดเบื้องต้นสำหรับการหาจุดภาพ การหาลายเส้น หรือ การหาวัตถุในภาพ ก็คือ การหาตำแหน่ง  $(x, y)$  ที่ทำให้เกิดความไม่ต่อเนื่อง (Discontinuity) ของค่าความเข้มของจุดภาพ (pixel intensity) ในภาพ ซึ่งประกอบด้วยขั้นตอนหลัก 2 ขั้นตอน คือ

1. Apply a mask over the image
2. Apply thresholding: If  $(|R| > T)$ , then *discontinuity* !! ดังรูปที่ 6.4

ตัวอย่าง หน้ากากสำหรับหาจุดภาพ



รูปที่ 6.3: ขอบของวัตถุในภาพ

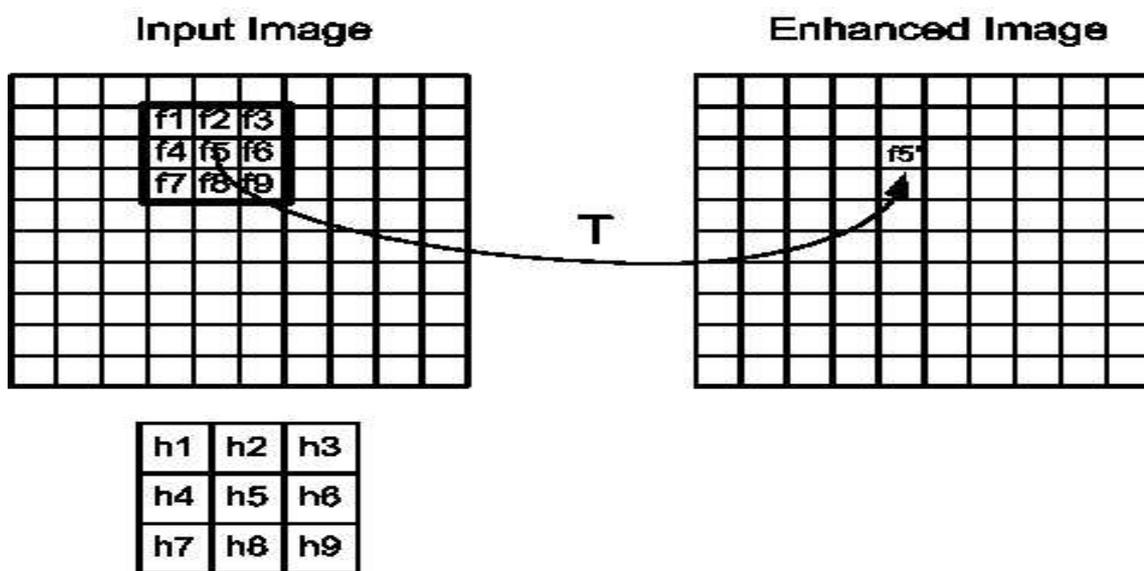
$$\begin{array}{|c|c|c|c|c|} \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 10 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline \end{array} * \begin{array}{|c|c|c|} \hline -1 & -1 & -1 \\ \hline -1 & 8 & -1 \\ \hline -1 & -1 & -1 \\ \hline \end{array} = \begin{array}{|c|c|c|c|c|} \hline - & - & - & - & - \\ \hline - & 72 & -9 & 0 & - \\ \hline - & -9 & -9 & 0 & - \\ \hline - & 0 & 0 & 0 & - \\ \hline - & - & - & - & - \\ \hline \end{array} \quad (6.1)$$

จากตัวอย่างภาพที่กำหนดให้ ในสมการ 6.1 ในสดมภ์ทางซ้าย ถูกดำเนินการด้วยตัวกรองความถี่สูง แล้วได้ภาพผลลัพธ์ในสดมภ์ทางขวา จะเห็นว่า จำนวนจุดภาพที่มีการเปลี่ยนค่าความเข้มจุดภาพจะขึ้นกับ ค่า Threshold ( $T$ ) กล่าวคือ

1. จะได้ 4 จุดภาพ ถ้า  $0 < T \leq 9$
2. จะได้ 1 จุดภาพ ถ้า  $9 < T \leq 72$
3. จะได้ 0 จุดภาพ ถ้า  $T > 72$

### 6.4.3 การทลายเส้น (Line Detection)

ตัวกรองสำหรับหาเส้นบนภาพ ในทิศทางต่างๆ



รูปที่ 6.4: การหาความไม่ต่อเนื่องบนภาพ โดยการดำเนินการกับกลุ่มจุดภาพ เมื่อ  $f'_5 = |R| = h_1f_1 + h_2f_2 + h_3f_3 + h_4f_4 + h_5f_5 + h_6f_6 + h_7f_7 + h_8f_8 + h_9f_9$

-1	-1	-1	-1	2	-1	-1	-1	2	2	-1	-1
2	2	2	-1	2	-1	-1	2	-1	-1	2	-1
-1	-1	-1	-1	2	-1	2	-1	-1	-1	-1	2

Horizontal      Vertical      +45°      -45°

(6.2)

ตัวอย่างการใช้งาน

Horizontal line		Mask		Convolved image										
0	0	0	0	0	*	-1	-1	-1	=	-	-	-	-	-
1	1	1	1	1		2	2	2		-	6	6	6	-
0	0	0	0	0		-1	-1	-1		-	-	-	-	-

(6.3)

Vertical line		Mask		Convolved image										
0	0	1	0	0	*	-1	-1	-1	=	-	-	-	-	-
0	0	1	0	0		2	2	2		-	0	0	0	-
0	0	1	0	0		-1	-1	-1		-	-	-	-	-

(6.4)

จากสมการ 6.3 เป็นตัวอย่างการหาเส้นในแนวนอนบนภาพที่มีเส้นตรงในแนวนอนอยู่ ซึ่งภาพผลลัพธ์ที่ได้ จะเห็นแนวเส้นตรงที่ชัดเจนออกมาทันที ในทางตรงกันข้ามในสมการที่ 6.4 เป็นการใช้ตัวกรองการหาเส้นในแนวนอน บนภาพที่มีเส้นตรงในแนวตั้ง ทำให้ผลการดำเนินงานออกมามีค่าเป็นศูนย์

แต่ในทางปฏิบัติในการเขียนโปรแกรมบนคอมพิวเตอร์ ซึ่งมีความสามารถในการทำงานที่สูงและรวดเร็วกว่ามนุษย์ เราจะใช้ตัวกรองทุกตัวที่มี (ที่เหมาะสม) ดำเนินการบนภาพนำเข้า แล้วเลือกผลลัพธ์ที่เหมาะสมที่สุด โดย

$$R(x, y) = \max(R_1(x, y), R_2(x, y), R_3(x, y), R_4(x, y)) \quad (6.5)$$

และ

IF  $R(x, y) > T$  Then *discontinuity*

ดังรูปที่ 6.5

#### 6.4.4 การหาขอบวัตถุ

ขอบวัตถุในภาพสามารถคำนวณได้จากอนุพันธ์ของจุดภาพรอบข้าง ซึ่งจุดที่จะเป็นขอบของวัตถุ นั้น จะคำนวณได้จาก

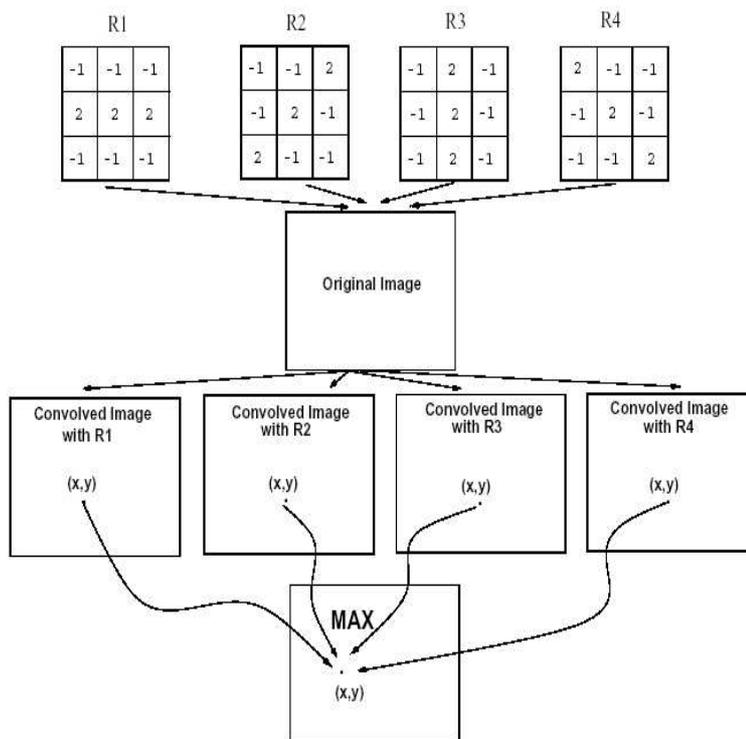
1. detecting local maxima or minima of the first derivative
2. detecting the zero-crossing of the second derivative

จากรูปที่ 6.6 เป็นตัวอย่างของภาพมีลักษณะเป็นแถว ซึ่งในแต่ละแถวของจุดภาพ จะมีเพียง 2 สี ดังแสดงในแถวที่ 2 ของรูปที่ 6.6 เมื่อหาอนุพันธ์อันดับที่หนึ่ง และ อนุพันธ์อันดับที่สอง ของแต่ละแถวของจุดภาพเทียบกับการเปลี่ยนตำแหน่งของจุดภาพในแนวนอน ( $\frac{\partial f(x,y)}{\partial x}$  และ  $\frac{\partial^2 f(x,y)}{\partial x^2}$ ) จะได้ผลลัพธ์ดังแสดงในแถวที่ 3 และ 4 ของรูปที่ 6.6

#### Computing edges using the first derivative

การหาค่า Gradient  $\nabla f$  โดย

$$\nabla f(x, y) = \begin{pmatrix} \frac{\partial f(x,y)}{\partial x} \\ \frac{\partial f(x,y)}{\partial y} \end{pmatrix} \quad (6.6)$$



รูปที่ 6.5: การใช้ตัวกรองกับภาพนำเข้า และ การเลือกภาพผลลัพธ์

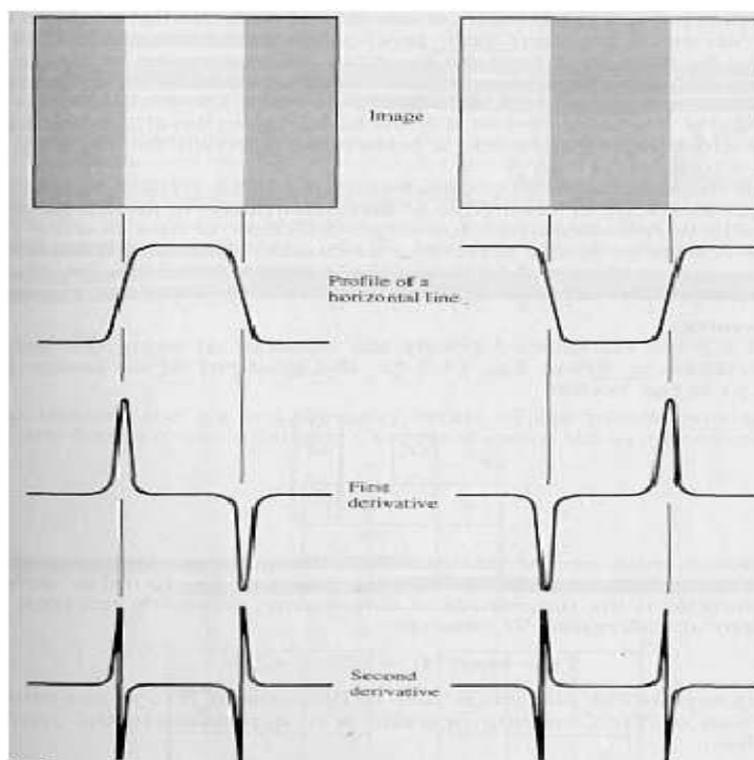
ค่า Gradient  $\nabla f$  เป็นเวกเตอร์ ที่มีขนาดและทิศทางดังนี้

$$\begin{aligned}
 \text{magnitude}(\nabla f) &= \sqrt{\frac{\partial f^2(x, y)}{\partial x} + \frac{\partial f^2(x, y)}{\partial y}} \\
 &= \sqrt{G_x^2 + G_y^2} \\
 &\approx |G_x| + |G_y|
 \end{aligned} \tag{6.7}$$

และ

$$\text{direction}(\nabla f) = \tan^{-1} \frac{\partial f(x, y)}{\partial y} / \frac{\partial f(x, y)}{\partial x} \tag{6.8}$$

ขั้นตอนวิธี สำหรับการเขียนโปรแกรม



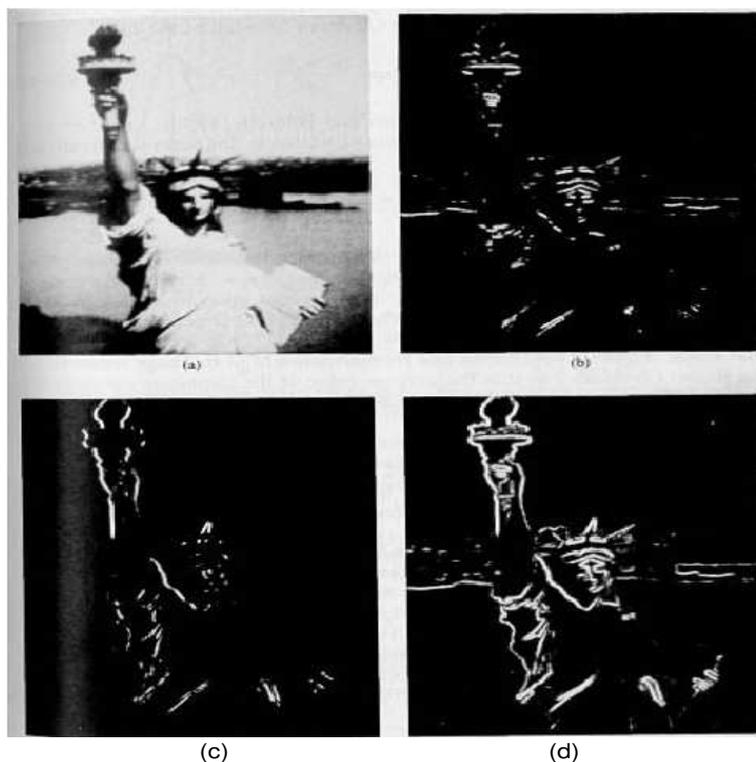
รูปที่ 6.6: การใช้อนุพันธ์กับภาพ

1.  $G_x = f(x, y) * M_x(x, y)$
2.  $G_y = f(x, y) * M_y(x, y)$
3.  $M(x, y) = |G_x| + |G_y|$
4.  $direction(\nabla f) = \tan^{-1}(G_y/G_x)$
5. IF  $M(x, y) \geq T$  then discontinuity

ตัวอย่างดังแสดงในรูปที่ 6.7

#### Properties of the magnitude of gradient

1. ค่า Magnitude ของ  $\nabla f$  จะบอกถึงค่าระดับ หรือ อัตราของการเปลี่ยนแปลงของค่าระดับเทาของจุดภาพ



รูปที่ 6.7: การหา Gradient บนภาพ (a) ภาพต้นฉบับ, (b) ผลลัพธ์จาก Gradient ในแนวแกน  $y$ ,  $|G_y|$ , (c) ผลลัพธ์จาก Gradient ในแนวแกน  $x$ ,  $|G_x|$ , (d) ผลลัพธ์จริง  $M(x, y) = |G_x| + |G_y|$

2. ค่า magnitude ของ gradient เป็น isotropical operator นั่นคือ สามารถคำนวณหาการเปลี่ยนแปลงในทุกทิศทาง
3. ค่าทิศทางของ  $\nabla f$  จะมีทิศทางตั้งฉากกับแนวขอบของวัตถุบนภาพเสมอ

### Computing edges using the second derivative

การหาอนุพันธ์อันดับที่สองสามารถคำนวณได้โดยใช้ Laplacian method ดังสมการ

$$\nabla^2 f(x, y) = \frac{\partial^2 f(x, y)}{\partial^2 x} + \frac{\partial^2 f(x, y)}{\partial^2 y} \quad (6.9)$$

การคำนวณอนุพันธ์อันดับที่สองสำหรับข้อมูลภาพจะมีความซับซ้อนค่อนข้างมาก จึงมีการนำเสนอวิธีการประมาณ การคำนวณอนุพันธ์อันดับที่สอง  $\nabla^2 f(x, y)$  ดังนี้

$$\frac{\partial^2 f(x, y)}{\partial^2 x} = f(x - 1, y) - 2f(x, y) + f(x + 1, y) \quad (6.10)$$

$$\frac{\partial^2 f(x, y)}{\partial^2 y} = f(x, y - 1) - 2f(x, y) + f(x, y + 1) \quad (6.11)$$

ดังนั้น

$$\nabla^2 f(x, y) = -4f(x, y) + f(x - 1, y) + f(x + 1, y) + f(x, y - 1) + f(x, y + 1) \quad (6.12)$$

ถ้ากำหนดให้หน้ากากรองขนาด 3x3 มีสมาชิกดังต่อไปนี้

z1	z2	z3
z4	z5	z6
z7	z8	z9

แล้ว จะสามารถประมาณ  $\nabla^2 f(x, y)$  ได้จาก

$$\nabla^2 f(x, y) = -4z_5 + (z_4 + z_6 + z_2 + z_8) \quad (6.13)$$

ดังนั้น หน้ากากรองสำหรับการหาขอบวัตถุ คือ (Laplacian mask)

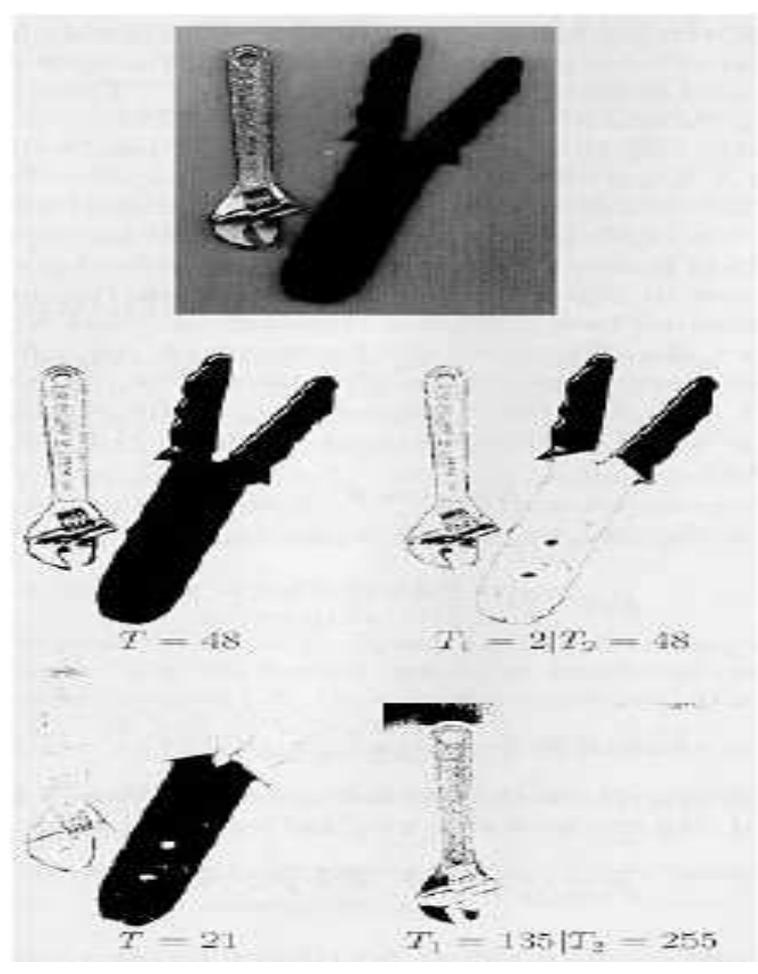
0	1	0
1	-4	1
0	1	0

### Properties of the second derivative

1. It is an isotropic operator
2. It is cheaper to implement (one mask only)
3. It does not provide information about edge direction
4. It is more sensitive to noise (differentiates twice)
5. To reduce the noise effect, the image should be first smoothed with a low-pass filter

### 6.4.5 Thresholding

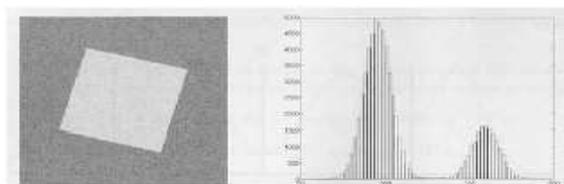
นอกจากวิธีที่กล่าวมาข้างต้น เช่น การกรองด้วยหน้ากาก และการหา Gradient บนภาพแล้ว ในการคำนวณจริง พื้นที่ หรือ ขอบของพื้นที่ สามารถหาได้จากการกำหนดค่า Threshold บนภาพ ดังรูป 6.8 จากรูปจะเห็นว่า การกำหนดค่า Threshold ระดับเท่าที่ต่างกัน จะได้ผลลัพธ์ที่ต่างกัน



รูปที่ 6.8: การหาวัตถุและขอบของวัตถุ ด้วยการกำหนดค่า Threshold ที่แตกต่างกัน

### การเลือกค่า Threshold ด้วย Histogram ของภาพ

- Regions with uniform intensity give rise to strong peaks in the histogram ดังรูป 6.9



รูปที่ 6.9: การหาวัตถุเมื่อภาพมีฮิสโตแกรมเป็น Uniform

#### ขั้นตอนวิธีสำหรับการโปรแกรม เมื่อภาพมีฮิสโตแกรมเป็น Uniform

$$\begin{aligned} \text{IF } f(x, y) \leq T \text{ Then } f(x, y) = 255 \\ \text{Else } f(x, y) = 0 \end{aligned} \quad (6.14)$$

- Multilevel thresholding is also possible (although more difficult in practices)

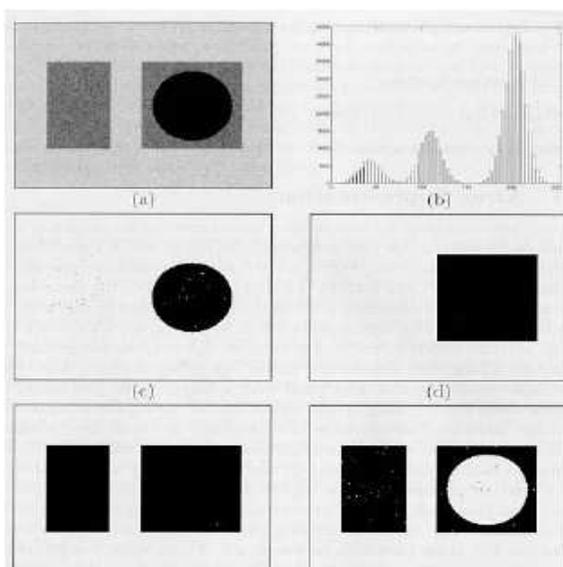
#### ขั้นตอนวิธีสำหรับการโปรแกรม เมื่อภาพมีฮิสโตแกรมหลายระดับ

$$\begin{aligned} \text{IF } f(x, y) \leq T_1 \text{ Then } f(x, y) = 255 \\ \text{Else IF } T_1 < f(x, y) \leq T_2 \text{ Then } f(x, y) = 128 \\ \text{Else } f(x, y) = 0 \end{aligned} \quad (6.15)$$

- ในทางปฏิบัติ ค่า threshold ที่ดี นั้น จะสามารถเลือกได้จาก การพิจารณาค่าของฮิสโตแกรม ซึ่ง มีจุดยอดที่สูง (tall) มีฐานฮิสโตแกรมที่แคบ (narrow) ฮิสโตแกรมมีความสมมาตร (symmetric) และ ระหว่างฮิสโตแกรมที่ค่อนข้างลึก (separated by deep valleys)

### Adaptive thresholding

ในบางครั้ง บนภาพระดับเทาหนึ่งภาพ มีความแตกต่างกันของค่าระดับเทาจำนวนมาก ทำให้การกำหนดค่า Threshold จำนวน 2-3 ชุดนั้น จะไม่สามารถแยกวัตถุ ออกมาได้ อีกทั้ง ถ้ากำหนดให้ที่ค่า Threshold หลายๆ ก็ยากแก่การกำหนดเงื่อนไขในโปรแกรมคอมพิวเตอร์ ดังนั้น จึงมีการนำเสนอให้มีการแบ่งภาพออกเป็นส่วนย่อย โดยใช้หลักการที่สำคัญ ดังนี้



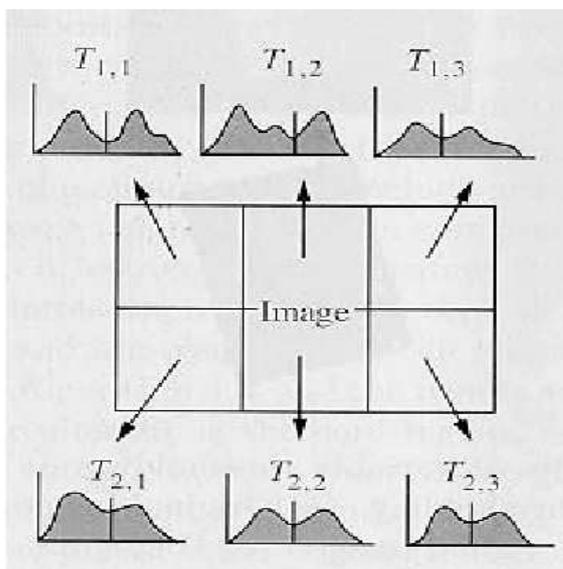
รูปที่ 6.10: การหาวัตถุเมื่อภาพมีฮิสโตแกรมหลายระดับ (Multi-Level histogram)

1. แบ่งข้อมูลภาพออกเป็น  $m \times n$  ส่วน
2. หาค่า Threshold,  $T_{ij}$ , ที่เหมาะสมสำหรับ แต่ละภาพย่อยที่ถูกแบ่งออกมา
3. การแบ่งภาพย่อยจะวนรอบการประมวลผลจนกว่าในแต่ละภาพย่อยจะมีฮิสโตแกรมเพียงสองฮิสโตแกรม ดังรูป 6.11
4. ดังนั้น เราก็จะสามารถใช้วิธีการที่นำเสนอมาก่อนหน้านี้ได้ทันที

## 6.5 การแยกภาพโดยวิธีหาพื้นที่ (Region-based approaches)

จากที่ได้กล่าวมาทั้งหมด จุดด้อยของการแยกภาพด้วยฮิสโตแกรมจะพอสรุปได้ดังนี้

1. วิธีการของฮิสโตแกรมจะพิจารณาเฉพาะค่าของจุดภาพเพียงอย่างเดียว ไม่มีการพิจารณาถึงหลักการเชิงพื้นที่ของภาพ
2. วิธีการของฮิสโตแกรมจะไม่ให้ความสำคัญกับความจริงที่ว่า จุดภาพที่อยู่ใกล้กันนั้นจะมีค่าความเข้มสีใกล้เคียงกัน



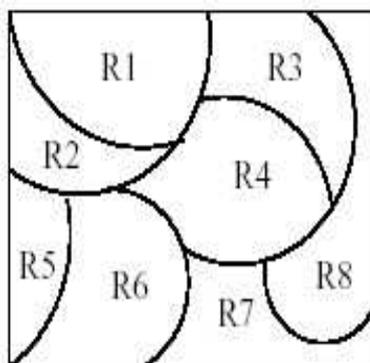
รูปที่ 6.11: การแบ่งภาพออกเป็นภาพย่อย เพื่อให้ภาพย่อยเหล่านั้นมีฮิสโตแกรม เพียง 2 กลุ่ม

3. ดังนั้น วิธีการหาพื้นที่จึงถูกนำเสนอขึ้นมาเพื่อเติมเต็มจุดด้อยของวิธีฮิสโตแกรม

#### หลักการ

- แบ่งภาพออกเป็นพื้นที่ย่อยๆ (sub-regions),  $R_1, R_2, \dots, R_n$  ดังรูป
- โดยมีคุณสมบัติ ดังนี้
  - $R_1 \cup R_2 \cup \dots \cup R_n = R$
  - $R_i$  จะต้องเชื่อมติดกัน
  - $R_i \cap R_j = \phi$
  - ความน่าจะเป็นของพื้นที่  $R_i$ ,  $P(R_i)$  จะต้องเป็นจริง หรือ มีค่าเกิดขึ้นแน่นอน
  - ความน่าจะเป็นของพื้นที่  $R_i$  และ  $R_j$ ,  $P(R_i \cup R_j)$  จะต้องเป็นเท็จ หรือ ไม่มีค่าเกิดขึ้นแน่นอน

การแยกภาพโดยวิธีหาพื้นที่ ประกอบด้วย 2 วิธีหลักๆ คือ



รูปที่ 6.12: การแบ่งภาพออกเป็นพื้นที่ย่อย

1. Region growing method
2. Region splitting and Merging method

### 6.5.1 Region growing method

#### หลักการ

1. เริ่มต้นจากการเลือกจุดภาพหนึ่งจุดเพื่อเป็นจุดเริ่มต้น (seed) หลังจากนั้น จะเพิ่มจุดภาพใหม่เข้าไปที่ละจุด
2. ขั้นตอนวิธี
  - (a) เลือกจุดภาพที่สนใจในการหาพื้นที่ 1 จุด
  - (b) ตรวจสอบค่าความเข้มของจุดภาพนั้น กับจุดภาพรอบข้าง ถ้ามีค่าความเข้มในระดับเดียวกัน ให้ เพิ่มเข้ามาในพื้นที่นั้น
  - (c) วนกลับไปทำงานในขั้นตอน (b) สำหรับจุดภาพใหม่ที่ดูถูกเพิ่มเข้ามา และ จะหยุดเมื่อไม่มีจุดภาพอื่นเพิ่มเข้ามาแล้ว
3. ในทางปฏิบัติ เราสามารถเลือก จุดภาพเริ่มต้นได้มากกว่า 1 จุดภาพ

#### หลักการเลือกจุดภาพเริ่มต้น

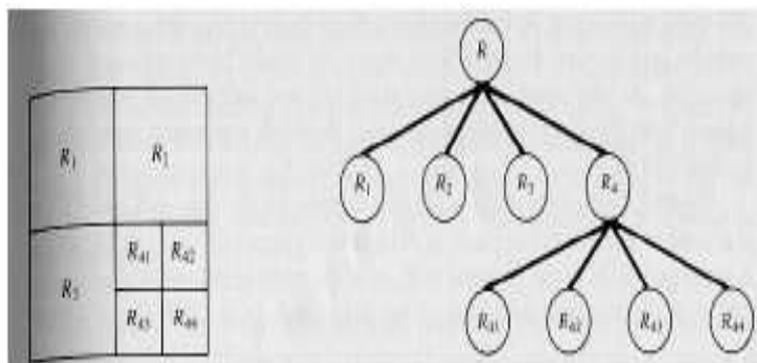
1. การเลือกจุดภาพเริ่มต้นขึ้นกับธรรมชาติของปัญหา

2. ถ้าไม่มีสารสนเทศใดๆ เกี่ยวกับภาพ ให้เลือกใช้ฮิสโตแกรมเข้ามาช่วย

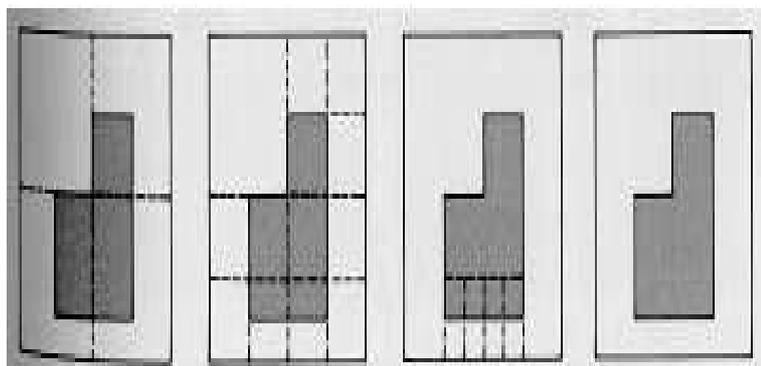
## 6.5.2 Region splitting and merging

### หลักการ

1. Split the image into a number of regions
2. Merge and/or split these regions again until the resulted regions satisfy the required properties
3. Splitting procedure
  - (a) If  $P(R) = False$ , split  $R$  into four quadrants
  - (b) If  $P$  is *false* on any quadrant, then subsplit
4. Merging procedure
  - (a) Merging is also needed to combine regions having similar properties



รูปที่ 6.13: Splitting procedure



รูปที่ 6.14: Merging procedure

สรุปขั้นตอนวิธีที่สำคัญสำหรับวิธีการ **Region splitting and merging**

1. Split into four disjointed quadrants any region  $R_i$  เมื่อ  $P(R_i) = False$
2. Merge any adjacent regions  $R_j$  และ  $R_k$  ถ้า  $P(R_j \cup R_k) = True$
3. Stop when no further merging or splitting is possible

## บทที่ 7

### การเข้ารหัส และ การบีบอัดข้อมูลภาพ Image coding and Compression

#### เนื้อหาสาระ

1. หลักการการเข้ารหัสและถอดรหัส
2. Huffman Encoding
3. Run-Length Encoding

#### วัตถุประสงค์

1. เพื่อศึกษาทฤษฎีและวิธีการเข้ารหัส และ ถอดรหัส
2. เพื่อพัฒนาโปรแกรมสำหรับการเข้ารหัส และ ถอดรหัส

#### กิจกรรมการเรียนการสอน

1. บรรยาย อภิปรายซักถาม
2. ยกตัวอย่าง
3. สรุปเนื้อหารายวิชา

#### การวัดผล

1. ซักถามความเข้าใจหลังการเรียน

## การเข้ารหัส และการบีบอัดข้อมูลภาพ Image coding and Compression

### 7.1 บทนำ

เป็นที่ทราบกันดีว่าแฟ้มข้อมูลภาพ (Image files) นั้นมีขนาดใหญ่มาก จึงมีความเป็นที่จะต้องทำให้แฟ้มข้อมูลมีขนาดเล็กลงเพื่อประหยัดพื้นที่ในการจัดเก็บ (Storage) และ ประหยัดเวลาในการส่ง (file transfers) โดยการบีบอัดและการเข้ารหัสข้อมูลภาพนั้น มี 2 ชนิด คือ ชนิดที่ยอมให้มีการสูญเสียข้อมูลบางส่วนของภาพ (Lossy compression) และ ชนิดที่ไม่ยอมให้สูญเสียข้อมูลภาพ (Lossless compression) ซึ่งผู้อ่านต้องพิจารณาตัดสินใจเลือกใช้เองว่า จะเลือกวิธีแบบบีบอัดและการเข้ารหัสวิธีใด โดยปกติแล้ว การบีบอัดข้อมูลภาพชนิดไม่ยอมให้สูญเสียข้อมูลภาพ จะนิยมใช้กับ ภาพที่ต้องการใช้ในทางกฎหมาย ทางด้านวิทยาศาสตร์ หรือ ภาพที่มีนัยสำคัญต่อการเมือง ซึ่งถ้ายอมให้มีการสูญเสียข้อมูลบางส่วนไป อาจทำให้กระบวนการพิสูจน์ต่างๆ เหล่านั้นไม่มีนัยสำคัญได้

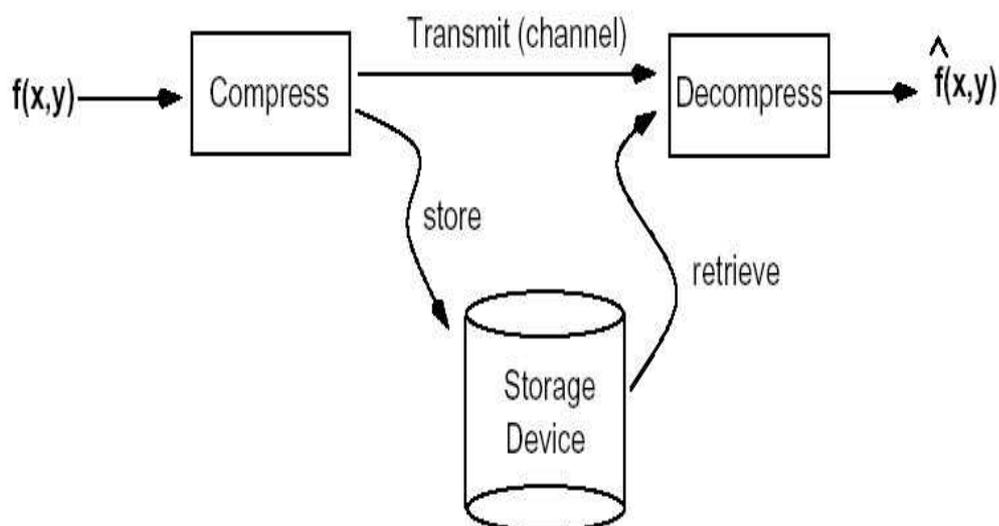
### 7.2 วัตถุประสงค์ของการเข้ารหัสและการบีบอัดข้อมูลภาพ

1. The goal of image compression is the reduction of the amount of data required to represent a digital image.
2. The idea is to remove redundant data from the image (i.e., data which do not affect image quality significantly)
3. Image compression is very important for image storage and image transmission

### 7.3 Huffman Coding

แนวคิดสำหรับการเข้ารหัสแบบ Huffman นั้นจะง่ายและตรงไปตรงมา ดังนี้

1. แทนการใช้การเข้ารหัสชนิดความยาวคงที่ (Fixed-length code) (8 บิต ต่อ 1 จุดภาพ) เป็นการเข้ารหัสแบบความยาวไม่คงที่ (Variable-length code)
2. รหัสที่สั้นจะใช้แทนค่าของจุดภาพระดับเทาที่มีความน่าจะเป็นสูง



รูปที่ 7.1: การเข้ารหัสและการบีบอัดข้อมูลภาพ

ยกตัวอย่าง เช่น มีภาพระดับเทา 4 ระดับ ซึ่งจะมีค่าในแต่ละจุดภาพ เป็น 0, 1, 2, 3 โดยแต่ละจุดภาพมีค่าความน่าจะเป็นเท่ากับ 0.2, 0.4, 0.3 และ 0.1 ตามลำดับ จากตารางที่ จะแสดงให้เห็นถึงการเข้ารหัสสำหรับภาพทั้งแบบความยาวในการแทนคงที่ และ แบบความยาวในการแทนไม่คงที่

ค่าระดับเทา	ความน่าจะเป็น	ความยาวในการแทนคงที่	ความยาวในการแทนไม่คงที่
0	0.2	00	000
1	0.4	01	1
2	0.3	10	01
3	0.1	11	001

พิจารณาว่าจากตารางที่ผ่านมาจะสามารถบีบอัดข้อมูลได้อย่างไร จะพบว่า แต่ละค่าระดับเทาจะมีรหัสที่เฉพาะตัว จำนวนบิตเฉลี่ยที่ใช้ในการแทนข้อมูลเท่ากับ

$$(0.2 * 3) + (0.4 * 1) + (0.3 * 2) + (0.1 * 3) = 1.9$$

จะเห็นว่า เมื่อเราให้รหัสที่ยาวที่สุด (3 บิต) ใช้แทนจุดภาพที่มีความน่าจะเป็นต่ำสุด (0.2 และ 0.1) และ ให้รหัสที่สั้นที่สุด (1 บิต) ใช้แทนจุดภาพที่มีความน่าจะเป็นสูงสุด ดังนั้น เราจะได้

จำนวนบิตเฉลี่ยในการแทนจุดภาพทั้งหมดเท่ากับ 1.9 บิต ซึ่งน้อยกว่า 2 บิต ซึ่งก็ถือได้ว่าเป็นผลลัพธ์ที่เป็นที่น่าพอใจ

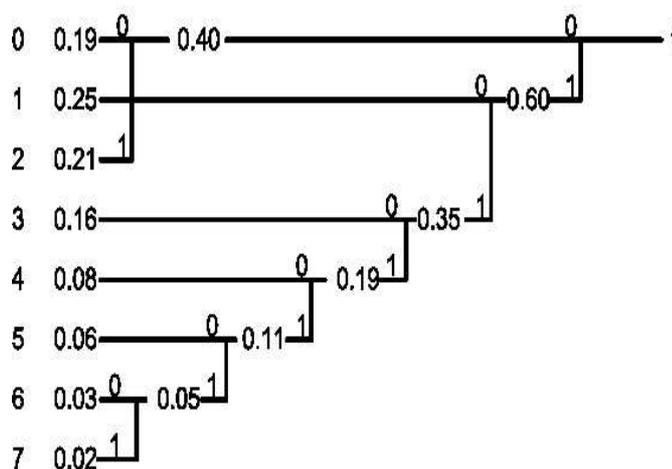
#### ขั้นตอนวิธีสำหรับการเข้ารหัสแบบ Huffman

1. หาค่าความน่าจะเป็นของแต่ละระดับเทาบนภาพ
2. สร้าง Binary Tree โดยใช้ความน่าจะเป็น
3. กำหนดค่า 0 และ 1 สำหรับแต่ละกิ่งของต้นไม้
4. อ่านรหัสจากบนลงล่าง

ตัวอย่าง สำหรับภาพระดับเทา 8 ระดับ ซึ่งถ้าใช้การแทนรหัสแบบความยาวคงที่จะใช้ 3 บิตในการแทน ถ้าใช้การเข้ารหัสแบบ Huffman สมมติให้ค่าระดับเทาแต่ละค่า มีค่าความน่าจะเป็น ตามตาราง ต่อไปนี้

Gray value	0	1	2	3	4	5	6	7
ค่าความน่าจะเป็น	0.19	0.25	0.21	0.16	0.08	0.06	0.03	0.02

รูปที่ 7.2 แสดงการสร้างต้นไม้สำหรับการเข้ารหัสแบบ Huffman และ รหัสที่จะใช้แทนค่าระดับเทาของแต่ละจุดภาพ จะมีขั้นตอนในการทำดังนี้ คือ เริ่มต้นจากรหัส 1 ของมุมบนขวา แล้วย้อนกลับไปหารหัส โดยรายละเอียดนั้นจะแสดงในตารางที่ 7.1



รูปที่ 7.2: Huffman Encoding และ กำหนดค่าบิตในแต่ละกิ่งของต้นไม้

Grey Value	Huffman Code
0	00
1	10
2	01
3	110
4	1110
5	11110
6	111110
7	111111

ตารางที่ 7.1: รหัส Huffman สำหรับการเข้ารหัสจากรูปที่ 7.2

จากตารางที่ 7.1 คำนวณค่าเฉลี่ยจำนวนบิตที่ใช้ในการแทนรหัส เท่ากับ

$$(0.19 \times 2) + (0.25 \times 2) + (0.21 \times 2) + (0.16 \times 3) + (0.08 \times 4) + (0.06 \times 5) + (0.03 \times 6) + (0.02 \times 6) = 2.7$$

ซึ่ง  $2.7 < 3$  นั่นคือ จำนวนบิตที่ใช้แทนค่าจุดภาพด้วยวิธี Huffman ลดลงอย่างมีนัยสำคัญ การถอดรหัส (Decoding) พิจารณาสายอักขระของรหัสบิต ต่อไปนี้

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

เนื่องจากในตารางที่ 7.1 ไม่มีรหัสใดเลยที่ถูกแทนที่ด้วย 1 และ 11 ดังนั้น เราจะเริ่มถอดรหัสจาก 3 บิตแรก และ ได้ค่าระดับเทา เท่ากับ 3 และ รหัสตัวถัดไปที่จะถอดได้ก็คือ 1110 ดังนั้น จากสายอักขระของรหัสบิตที่กำหนดให้ ก็จะถูกถอดออกได้ ดังนี้

$$\underbrace{1\ 1\ 0}_3 \underbrace{1\ 1\ 1\ 0}_4 \underbrace{0\ 0\ 0\ 0}_0 \underbrace{0\ 0}_0 \underbrace{1\ 0}_1 \underbrace{0\ 1}_2 \underbrace{1\ 1\ 1\ 1\ 0}_5$$

## 7.4 Run-Length Encoding: RLE

แนวคิดพื้นฐานสำหรับการเข้ารหัสแบบ Run-Length คือ การเข้ารหัสจากจำนวนการซ้ำของสายอักขระแต่ละตัว การเข้ารหัสแบบ RLE เป็นมาตรฐานที่ใช้ในการส่งโทรสาร (Facsimile transmission)

**หลักการ**

1. หลักการหาจุดที่มีสีเหมือนกันและอยู่ติดกัน แล้วนำมารวมเหลือจุดเดียว และ บันทึกจุดเฉพาะค่าจุดสีเดิมไว้
2. RLE จะตรวจสอบข้อมูลที่ละบรรทัด ทำให้ภาพที่ประกอบด้วยเส้นตามแนวนอน จะบีบได้มากกว่าเส้นแนวตั้ง

ตัวอย่าง

aaaaaaaaabaaaaabbbbbbaaab

เมื่อเข้ารหัสแล้วก็ได้ดังนี้

8ab5a7b3ab

## บรรณานุกรม

- [1] S.-I.Amari, A.Cichocki, and H.H.Yang. A New Learning Algorithm for Blind Signal Separation, *MIT Press*, pp.757-763, 1996.
- [2] J.Bernd, *Digital Image Processing*, Springer-Verlag, 1997.
- [3] G.A.Baxes, *Digital Image Processing: Principles and Applications*, John Wiley & Sons, Inc., 1994.
- [4] M.Petrou and P.Bosdogianni, *Image Processing: The Fundamentals*, John Wiley & Sons Ltd, 1999.
- [5] C.W.Therrien, *Decision Estimation And Classification: An Introduction to Pattern Recognition and Related Topics*, John Wiley & Sons, 1989.
- [6] K.Chinnasarn, C.Lursinsap, and V.Palade, Blind Extraction of Mixed Kurtosis Signed Signal using Partial Observations and Low Complexity Activation Functions, *the International Journal of Computational Intelligence and Applications* in the special issue on "Computational Intelligence for Signal and Image Processing", Vol.4, No.2. (June 2004.)
- [7] K.Chinnasarn, S.Chinnasarn, and D.L.Pyle, Surimi Gel Pattern Identification Using Eigen-Pattern and Similarity Analysis, *Proceedings of the 9<sup>th</sup> National Computer Science and Computer Engineering Conference: NCSEC2005.*, 2005.
- [8] S.Chinnasarn, K.Chinnasarn, and D.L.Pyle, Comparison of Similarity Methods for Classification of Surimi Gel Strength Patterns based on Principal Component Analysis, *Proceedings of the Joint Conference on Computer Science and Software Engineering, JCSSE2005*, pp. 55-61, 2005.
- [9] A.Cichocki and S.-I.Amari. *Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications*, John Wiley & Sons, Ltd., 2002.
- [10] P.Comon. Independent Component Analysis: A New Concept?, *Signal Processing*, Vol. 36. pp. 287-314, 1994.
- [11] Duda.R.O, Hart.P.E., and Stork.D.G. *Pattern Classification*, 2<sup>nd</sup> edition, John Wiley & Sons, Inc., 2001.

- [12] A.Hyvarinen and E.Oja. Independent Component analysis: algorithms and applications, *Neural Networks*. Vol. 13 pp. 411-430, 2000.
- [13] R.Weeks, Jr., *Fundamentals of Electronics Image Processing*, SPIE/IEEE Series on Imaging Science and Engineering, IEEE PRESS, 1998.
- [14] S.Barnett, *Matrix Methods for Engineers and Scientists*, McGRAW-Hill Book Company (UK) Limited, 1979.

## ดรรชนี

- Adaptive thresholding, 86
- amplification factor, 44
- Background, 75
- Blind Source Separation, 20, 67
- cocktail party problem, 20
- Convolution, 47, 60, 63
- Covariance, 34
- CT-Scanner, 44
- Decoding, 96
- Degradation, 60
- Derivative filter, 48
- Digital Image, 4
- Discontinuity, 77
- Distribution of a Random Vector, 31
- dot product, 19
- Edge Detection, 75
- Edges, 48
- Eigenvalue, 65
- Eigenvector, 65
- Enhancement, 77
- Filtering, 77
- Fourier transform, 24
- Frequency domain, 39
- Gradient, 50, 81
- High-order statistics, 67
- High-pass filters, 48
- Histogram, 39
- Histogram Equalization, 42
- Image Analysis, 75
- Image Enhancement, 37
- Image Resolution, 7
- Image Restoration, 60
- Image Segmentation, 75
- Image size, 6
- Image Subtraction, 44
- Image Transformation, 17
- Independence, 34
- Independent Component Analysis, 20
- joint distribution function, 32, 34
- Kullback-Leibler divergence, 67
- Kurtosis, 23
- Laplacian, 83
- Line Detection, 78
- Linear Image Processing, 17
- Linear Smoothing Filters, 45
- Mask processing method, 45
- Merging, 90
- Monochrome Image, 4
- Multicolor Image, 4
- Noise, 45, 61
- Non-linear Smoothing Filters, 47
- Objects, 75
- Operand, 9
- Operator, 8

orthogonal matrix, 23

PCA, 65

Picture Element, 4

pixel, 4

Point Detection, 77

Point Spread Function, 9

Preprocessing, 45

Prewitt Filter, 52

Probability density function, 39

Probability of Events, 30

pseudo-inverse, 67

Random variable, 30

Region growing, 89

Region-based, 87

Roberts filter, 51

Run-Length Encoding, 96

Salt-and-Pepper Noise, 71

Segmentation, 45

Sharpen Filters, 48

Singular Value Decomposition, 19

Smoothing Filters, 45

Spatial coordinates, 4

Spatial Domain, 60

Splitting, 90

Stack operator, 10

sub-regions, 88

supervector, 32

Thresholding, 85

Transformation, 8

Triangular filter, 46

Uniformly distributed, 42

unitary transform, 18

Variance, 34

Vector Multiplication, 17

What is an Image, 4

## **Biography**

**Name:** Assistant Professor Krisana CHINNASARN, Ph.D.

### **Educations:**

- Ph.D.(Computer Science), Department of Mathematics, Chulalongkorn University, Thailand, (June 2000 - March 2004)
- Ph.D.(Visiting student), Oxford University Computing Laboratory, Oxford, UNITED KINGDOM, (October 2002 - June 2003).
- M.Sc.(Computer Science and Information Technology), King Mongkut's Institute of Technology Ladkrabang, Thailand. (June 1994 - November 1997).
- B.Sc.(Statistics), Srinakarinwirot University, Mahasarakham, Thailand. (June 1989 - March 1992).

### **Research Area:**

- Artificial Neural Network
- Algorithm Analysis and Design
- Data Mining
- Digital Signal & Image Processing
- Decision Support & Making System
- Fuzzy Logic & System
- Genetic Algorithm
- Pattern Classification and Recognition

### **Scholarship:**

- Thai Government for Ph.D. program
- Burapha University for M.Sc. program

## ภาคผนวก

เอกสารประกอบการบรรยาย