

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การวิเคราะห์ภาพรหัสแถบ 2 มิติ  
( Two-Dimensional Bar Code Image Analysis )



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาอิเล็กทรอนิกส์  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2546

เลขหมู่.....  
เลขทะเบียน..... 55798  
วัน,เดือน,ปี 25 พ.ค. 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์อื่นใด  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



การวิเคราะห์ภาพรหัสแถบ 2 มิติ  
( Two-Dimensional Bar Code Image Analysis )



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาอิเล็กทรอนิกส์  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2546

ภาควิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การวิเคราะห์ภาพรหัสแถบ 2 มิติ

(Two-Dimensional Bar Code Image Analysis)

ผู้จัดทำ

นางสาว สิริพร เรืองวัฒนไพศาล รหัส 42010347



อาจารย์ที่ปรึกษา

(รศ.ดร. สุรพันธ์ เอื้อไพบูลย์)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงการ การวิเคราะห์ภาพรหัสแถบ 2 มิติ

(Two-Dimensional Bar Code Image Analysis)

โดย นางสาว ศิริพร เรืองวัฒนไพศาล รหัส 42010347

โครงการนี้ได้รับการตรวจสอบแล้ว พร้อมทั้งจะทำการสอบได้



(รศ. ดร. สุรพันธ์ เอื้อไพบูรณ์)

อาจารย์ที่ปรึกษา



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## การวิเคราะห์ภาพรหัสแถบ 2 มิติ

นางสาว ศิริพร เรืองวัฒนไพศาล รหัส 42010347  
 รศ. ดร. สุรพันธ์ เอื้อไพบูลย์ อาจารย์ที่ปรึกษา  
 ปีการศึกษา 2546

### บทคัดย่อ

เนื่องจากความต้องการที่เพิ่มขึ้นในการจัดเก็บข้อมูลให้ได้มากขึ้น โดยใช้พื้นที่ในการเก็บลดลง ยกตัวอย่างเช่นผลิตภัณฑ์ที่นำออกจำหน่ายในท้องตลาดที่มีจำนวนเพิ่มขึ้น และมีรายละเอียดของผลิตภัณฑ์มากขึ้น จึงได้มีการนำเอารหัสแถบ 2 มิติมาใช้ในการบ่งชี้ผลิตภัณฑ์แทนรหัสแถบแบบแท่ง (1 มิติ) ทั่วไป และวิธีการที่ใช้ในการถอดรหัสรหัสแถบที่นิยมกันมากได้แก่การใช้เครื่องเลเซอร์สแกน ซึ่งจะเสียเวลาในการสแกนและยังมีข้อจำกัดในเรื่องมุมและระยะของการสแกน โครงการนี้จึงขอเสนอวิธีการใช้กล้องถ่ายภาพ และนำภาพไปผ่านกระบวนการทางคอมพิวเตอร์เพื่อวิเคราะห์รูปภาพรหัสแถบ 2 มิติ เพื่อทำการถอดรหัส

ชนิดของรหัสแถบที่ใช้ในโครงการ ได้แก่ MicroPDF417 เริ่มจากรูปภาพที่ถูกถ่ายในมุมที่แตกต่างกันเป็นเกรย์สเกล จะถูกนำมาเปลี่ยนเป็นรูปขาว-ดำ (binary) โดยการเลือกค่าเทรชโฮลด์ (threshold) ที่ดีที่สุด แล้วใช้เทคนิคการตรวจจับขอบของรูป (Edge Detection) เข้ามาช่วยในการกำจัดบริเวณที่ไม่ใช่ส่วนหนึ่งของรหัสแถบออกให้หมด ขั้นตอนสุดท้ายคือการวาร์ป (Polynomial Warping) เพื่อให้รหัสแถบที่อาจมีการโค้งงอหรือบิดเบี้ยวกลับมาเป็นรูปสี่เหลี่ยมผืนผ้า เพื่อที่จะใช้ในการถอดรหัส ผลการทดลองจะได้จากการเปรียบเทียบค่ารหัสที่อ่านออกมาโดยใช้โปรแกรมคอมพิวเตอร์ที่เขียนขึ้น กับค่ารหัสที่แท้จริงของบาร์โค้ดนั้น

## Two-Dimensional Bar Code Image Analysis

Miss Siriporn Ruangwattanapaisarn 42010347

Assoc.Prof.Dr. Surapan Airphaiboon Advisor

Academic Year 2003

### Abstract

The number of products is becoming more and more, as well as its details and specifications. So, high-capacity and small-space bar code is needed. Two-dimensional (2D) bar codes are used as their data capacity is much more than one-dimensional (1D) bar codes. There are several ways to decode the 2D bar code. The commonly used method is laser scanning, which takes time to sweep along the code and there are some limitations of the distance. The camera capturing method is proposed to decode 2D bar code. The bar code images are taken by imaging device and processed by a computer in order to decode the bar codes.

The type of 2D bar codes used in this paper is microPDF417. This paper proposes techniques for detecting codewords from microPDF417 bar code images taken by digital camera in grayscale. First, the input image is converted to a binary image by using optimal threshold. Then apply edge detection for eliminating unwanted regions and noises. The last process is polynomial warping for transforming bar code to a readable pattern, which should be in the rectangular form. The efficiency of the proposed techniques is assessed by the percentage of accuracy of detected codeword value compared to the real value.

### กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ถือเป็นการร่วมมือระหว่างมหาวิทยาลัย The University of Electro-Communications (UEC) ในกรุงโตเกียว ประเทศญี่ปุ่น กับสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เนื่องจากเป็นโครงการที่ต่อเนื่องจากเวลาที่เป็นนักศึกษาแลกเปลี่ยนกับโครงการ JUSST Exchange Program จึงขอขอบคุณผู้เกี่ยวข้องทุกท่าน ได้แก่ รศ.ดร.สุรพันธ์ เอื้อไพบูรณ์ อาจารย์ที่ปรึกษาผู้ให้ความช่วยเหลือและเข้าใจมาตลอด, อาจารย์ทุกท่านและเจ้าหน้าที่ในภาควิชาอิเล็กทรอนิกส์, Prof. Akira Kurematsu ผู้เป็นที่ปรึกษาโครงการ ณ UEC, อาจารย์และเจ้าหน้าที่ทุกท่านที่ดูแล JUSST Exchange Program รวมทั้ง Host family ครอบครัว Shimomura และเพื่อน ๆ ทุกคนที่ทำให้การใช้ชีวิตที่ญี่ปุ่นเป็นประสบการณ์ที่ดี รวมทั้งเพื่อนในเมืองไทยทุกคน และสมาชิกในครอบครัวที่คอยให้กำลังใจและอยู่เป็นเพื่อนในยามศึก

ทั้งนี้ ขอขอบคุณและความคิดพลาดที่อาจปรากฏอยู่ในปริญญานิพนธ์ฉบับนี้ ผู้จัดทำขออ้อมรับข้อตำหนิต่างหมด และจะทำการปรับปรุงแก้ไขให้ถูกต้องต่อไป

ผู้จัดทำ

## สารบัญ

	หน้า
บทคัดย่อ	I
Abstract	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VII
สารบัญรูปภาพ	VIII
บทที่ 1 บทนำ	1
1.1 วัตถุประสงค์ของโครงการ	1
1.2 ขอบเขตของโครงการ	1
1.3 ประโยชน์และผลที่คาดว่าจะได้รับ	1
1.4 กระบวนการประมวลผลภาพ	1
บทที่ 2 รหัสแถบ 2 มิติ และ MicroPDF417	3
2.1 รหัสแถบ 2 มิติ (2D Bar code)	3
2.1.1 ชนิดของรหัสแถบ 2 มิติ	3
2.1.2 คุณสมบัติของรหัสแถบ 2 มิติ	4
2.2 รหัสแถบชนิด MicroPDF417	5
2.2.1 รูปแบบของ MicroPDF417	5
2.2.2 คุณสมบัติของ MicroPDF417	7
2.3 ประโยชน์และการประยุกต์ใช้งานรหัสแถบ 2 มิติ	8
บทที่ 3 หลักการเบื้องต้นและทฤษฎีที่ใช้ในการวิเคราะห์ภาพ	10
3.1 ระบบการมองเห็นภาพ (Vision System)	10
3.1.1 การได้มาซึ่งภาพ (Image Acquisition)	10
3.1.2 กระบวนการประมวลผลภาพ (Image Processing)	10
3.1.3 ผลที่ได้และการแสดงผล (Output or Display)	11
3.2 หลักการเบื้องต้นของการประมวลผลภาพ	11
3.2.1 พิกเซล (Pixel)	11
3.2.2 เกรย์สเกล (Gray scale)	12
3.2.3 ฮิสโตแกรม (Histogram)	12

	หน้า
3.3 การทำเทรชโฮลด์ (Thresholding)	13
3.4 การตรวจจับขอบภาพ (Edge Detection)	15
3.4.1 พื้นฐานของเครื่องมือตรวจจับขอบภาพ (Edge Detectors)	15
3.4.2 วิธีการตรวจจับขอบภาพของโซเบล (Sobel Edge Detector)	17
3.5 มอร์โฟโลยี (Morphology)	19
3.5.1 กระบวนการไคเลชั่น (Dilation)	20
3.5.2 กระบวนการอีโรชั่น (Erosion)	21
3.5.3 กระบวนการปิด (Closing)	22
3.5.4 กระบวนการเปิด (Opening)	22
3.6 การวาร์ป (Warping)	23
3.6.1 ชนิดของการวาร์ป	23
3.6.2 การวาร์ปโดยใช้โพลีโนเมียลลำดับที่ 2 (Second Order Polynomial Warping)	25
บทที่ 4 การออกแบบโปรแกรมที่ใช้วิเคราะห์ภาพรหัสเลข 2 มิติ	28
4.1 การเลือกค่าเทรชโฮลด์และแปลงภาพเป็นไบนารี (Thresholding)	28
4.2 การกำจัดสิ่งรบกวน (Noise Cancellation)	30
4.3 การหมุนภาพให้รหัสเลขวางตัวในแนวนอน (Rotation)	31
4.4 การหามุมของรหัสเลข (Corner Detection)	33
4.5 การทำโพลีโนเมียลวาร์ปลำดับที่ 2 (Second-order Polynomial Warping)	34
4.6 การถอดรหัส (Decoding)	36
บทที่ 5 ผลการประมวลผลภาพและผลการทดลอง	38
5.1 การเลือกค่าเทรชโฮลด์และแปลงภาพเป็นไบนารี (Thresholding)	38
5.2 การกำจัดสิ่งรบกวน (Noise Cancellation)	39
5.3 การหมุนภาพให้รหัสเลขวางตัวในแนวนอน (Rotation)	40
5.4 การหามุมของรหัสเลข (Corner Detection)	41
5.5 การทำโพลีโนเมียลวาร์ปลำดับที่ 2 (Second-order Polynomial Warping)	41
5.6 การถอดรหัส (Decoding)	43
บทที่ 6 สรุปและวิจารณ์	46
6.1 สรุปและวิเคราะห์ผลการทดลอง	46
6.2 แนวทางการพัฒนาและปรับปรุงโปรแกรม	46
6.3 ประโยชน์และการใช้งาน	47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก	หน้า
โปรแกรม (Source Code)	48
รหัสที่อ่านได้จากภาพต้นแบบ 6 ภาพ	
บรรณานุกรม	49



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญตาราง

	หน้า
ตารางที่ 2.1 แสดงชนิดของตัวอักษรและจำนวน โค้ดเวิร์ดที่ใช้	6
ตารางที่ 3.1 แสดงจำนวนจุดควบคุมที่ต้องใช้กับการวาร์ปแบบเชิงเส้น	24
ตารางที่ 3.2 แสดงจำนวนจุดควบคุมที่ต้องใช้กับการวาร์ปแบบไม่เชิงเส้น	25
ตารางที่ 5.1 แสดงคำรหัสที่อ่านได้จากภาพที่ผ่านการประมวลผลแล้วเทียบกับคำรหัสจริง	43



## สารบัญรูปภาพ

	หน้า
รูปที่ 1.1 บล็อกไดอะแกรมรวมแสดงการประมวลผลภาพรหัสแถบ	2
รูปที่ 2.1 ตัวอย่างของสแต็คโค้ด	3
รูปที่ 2.2 ตัวอย่างของเมทริกซ์โค้ด	4
รูปที่ 2.3 โครงสร้างของรหัสแถบชนิด MicroPDF417	5
รูปที่ 2.4 ตัวอย่างของ MicroPDF417 ขนาดต่าง ๆ	7
รูปที่ 2.5 ความแตกต่างของรหัสแถบ 1 มิติกับรหัสแถบที่เป็นพอร์เทเบิลด้าไฟล์	8
รูปที่ 2.6 ตัวอย่างการประยุกต์ใช้งานรหัสแถบ 2 มิติ	9
รูปที่ 3.1 ดัชนีแสดงพิกเซลในเมทริกซ์ภาพ	11
รูปที่ 3.2 ซีดีแกรมของแคร์สเกล 8 ระดับจากเมทริกซ์ 4 x 5	13
รูปที่ 3.3 การเลือกค่าเทรชโฮลแบบเทรชโฮล 1 ค่า	14
รูปที่ 3.4 ผลของการเลือกค่าเทรชโฮล	14
รูปที่ 3.5 การใช้เคิร์ฟฟิลล์ลำดับที่ 1 และ 2 ใน 1 มิติ	16
รูปที่ 3.6 มาสก์ของวีธีโซเบล	18
รูปที่ 3.7 ซูโดคอนไวลูชั่นมาสก์	18
รูปที่ 3.8 ตัวอย่างส่วนโครงสร้าง (structuring element)	19
รูปที่ 3.9 ตัวอย่างการไคเลชั่น	20
รูปที่ 3.10 ตัวอย่างการอีโรชัน	21
รูปที่ 3.11 ตัวอย่างการปิด (Closing)	22
รูปที่ 3.12 ตัวอย่างการเปิด (Opening)	23
รูปที่ 3.13 ตัวอย่างการวาร์ปโดยใช้โพลิโนเมียลลำดับที่ 2	27
รูปที่ 4.1 บล็อกไดอะแกรมแสดงการทำเทรชโฮล	29
รูปที่ 4.2 บล็อกไดอะแกรมการกำจัดสิ่งรบกวนออกจากภาพ	30
รูปที่ 4.3 บล็อกไดอะแกรมการหมุนภาพให้รหัสแถบวางตัวในแนวนอน	32
รูปที่ 4.4 บล็อกไดอะแกรมการหามุมของรหัสแถบ	34
รูปที่ 4.5 บล็อกไดอะแกรมการทำโพลิโนเมียลวาร์ปลำดับที่ 2	35
รูปที่ 4.6 บล็อกไดอะแกรมการถอดรหัสของรหัสแถบ	37
รูปที่ 5.1 ตัวอย่างการทำเทรชโฮลที่ได้ผลดี	38
รูปที่ 5.2 ตัวอย่างการทำเทรชโฮลที่ได้ผลไม่ค่อยดี	39

	หน้า
รูปที่ 5.3 แสดงแต่ละขั้นตอนและผลจากการกำจัดสิ่งรบกวนออกจากภาพ	40
รูปที่ 5.4 ตัวอย่างภาพที่ผ่านการหมุน (Rotation)	40
รูปที่ 5.5 แสดงตำแหน่งมุมที่ตรวจจับได้ (ขยายใหญ่)	41
รูปที่ 5.6 ตัวอย่างของภาพที่ได้จากการวาร์ปภาพต้นแบบในมุมมอง Perspective ซึ่งได้ผลไม่ดี	42
รูปที่ 5.7 แสดงขั้นตอนในการวาร์ปที่ได้ผลดี	42
รูปที่ 5.8 แสดงภาพต้นแบบที่ใช้ทดลองและภาพที่ได้จากการประมวลผลในขั้นตอนสุดท้าย	43



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 1

## บทนำ

## 1.1 วัตถุประสงค์ของโครงการ

เพื่อนำเสนอโปรแกรมแสดงกระบวนการถอดรหัสของรหัสแถบ (Bar code) ในรูปภาพที่ถ่ายด้วยกล้องดิจิทัล โดยการใช้ทฤษฎีการประมวลผลภาพดิจิทัล (Digital image processing) ต่าง ๆ เพื่อช่วยในการแปลงรหัสแถบจากภาพต้นแบบที่อาจมีการบิดเบี้ยวหรือโค้งงอเกิดขึ้นเนื่องจากความผิดพลาดของการถ่าย ไปเป็นภาพที่มีลักษณะเป็นรูปสี่เหลี่ยมผืนผ้าซึ่งเหมาะสมต่อการอ่านค่ารหัส

## 1.2 ขอบเขตของโครงการ

- โปรแกรมที่ใช้ในการประมวลผลคือ MATLAB เวอร์ชัน 6.5 ซึ่งโปรแกรมขั้นตอนการวิเคราะห์ภาพที่นำเสนอในโครงการ อาจไม่สามารถทำการประมวลผลโดยใช้ MATLAB เวอร์ชันที่ต่ำกว่าได้
- ชนิดของรหัสแถบ 2 มิติที่ใช้ ได้แก่ MicroPDF417
- โปรแกรมสามารถใช้ประมวลผลได้กับภาพต้นแบบที่ถ่ายในเกรย์สเกล (Gray scale) เท่านั้น
- ภาพต้นแบบที่นำมาใช้ทดลองมี 6 ภาพ โดยถ่ายจากรหัสแถบ 2 แบบ

## 1.3 ประโยชน์และผลที่คาดว่าจะได้รับ

โปรแกรมที่นำเสนอมีประสิทธิภาพในการแปลงภาพรหัสแถบ 2 มิติให้อยู่ในรูปแบบที่เหมาะสมในการถอดรหัส และทำให้ค่ารหัสที่อ่านได้มีความผิดพลาดน้อยที่สุด

## 1.4 กระบวนการประมวลผลภาพ

สามารถแสดงได้ดังบล็อกไดอะแกรมต่อไปนี้



รูปที่ 1.1 บล็อกโคอะแกรมรวมแสดงการประมวลผลภาพรหัสแถบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

### รหัสแถบ 2 มิติและ MicroPDF417

#### 2.1 รหัสแถบ 2 มิติ (2D Bar code)

รหัสแถบ 2 มิติ (2D Bar code) คือ สัญลักษณ์ที่มีสามารถเก็บข้อมูลได้ทั้งในแกนตั้งและแกนนอน จึงสามารถเก็บข้อมูลได้มากกว่ารหัสแถบแบบ 1 มิติ (1D Bar code หรือ Linear bar code) และยังมีขนาดที่เล็กกว่าอีกด้วย โดยข้อมูลจะถูกเข้ารหัสเป็นค่าของรหัสแถบ ต่างกับรหัสแถบแบบ 1 มิติที่เป็นเพียงตัวเรียกข้อมูลซึ่งจะต้องถูกเก็บในหน่วยความจำอีกที ดังนั้นจึงอาจกล่าวได้ว่ารหัสแถบ 2 มิติเป็นแฟ้มข้อมูลเคลื่อนที่ (Portable data file; PDF)

##### 2.1.1 ชนิดของรหัสแถบ 2 มิติ

รหัสแถบ 2 มิติจำแนกออกเป็น 2 ประเภท ได้แก่

##### 1. สแต็กโค้ด (Stacked code)

เป็นรหัสแถบที่ถูกพัฒนามาจากรหัสแถบแบบ 1 มิติ โดยมีการแบ่งตามแนวยาวของรหัสออกเป็นแถว ๆ จึงถูกเรียกว่าเป็นสัญลักษณ์แบบหลายแถว (Multi-row symbology) ชนิดของรหัสแถบนี้ที่นิยมมากที่สุด ได้แก่ PDF417 ซึ่งภายหลังได้มีการพัฒนาไปสู่อีกชนิดหนึ่งให้มีขนาดเล็กลงเพื่อใช้ในงานที่มีการจำกัดพื้นที่มาก ๆ เรียกว่า MicroPDF417 นอกจากนี้ยังมีชนิด Code 49 และ Code 16K



รูปที่ 2.1 ตัวอย่างของสแต็กโค้ด

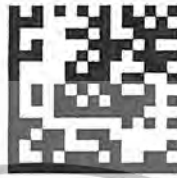
การอ่านค่ารหัสแถบนี้ สามารถใช้เครื่องเลเซอร์สแกนเนอร์ (Laser scanner), เครื่องซีซีดีสแกนเนอร์ (Charge-Coupled Device scanner; CCD scanner) หรือเครื่องถ่ายภาพ 2 มิติ (2D imaging device)

## 2. เมทริกซ์โค้ด (Matrix code)

มีรูปแบบที่หลากหลายมาก โดยอาจจะเป็นแบบวงกลม สี่เหลี่ยม หรือหกเหลี่ยม ซึ่งสามารถเก็บข้อมูลได้มากกว่าแบบแรก และยังสามารถอ่านในทิศทางใดก็ได้ ตัวอย่างรหัสแถบแบบนี้ได้แก่ Data Matrix, MaxiCode, Aztec Code, QR Code, และ Code One



MaxiCode



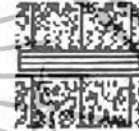
Data Matrix



Aztec Code



QR Code



Code One

รูปที่ 2.2 ตัวอย่างของเมทริกซ์โค้ด

การอ่านรหัสแถบแบบเมทริกซ์โค้ด สามารถใช้ได้เพียงเครื่องซีซีดีสแกนเนอร์ (CCD scanner) หรือเครื่องถ่ายภาพ 2 มิติ (2D imaging device) เท่านั้น

### 2.1.2 คุณสมบัติของรหัสแถบ 2 มิติ

รหัสแถบแบบ 2 มิติมีคุณสมบัติที่เหมือนและแตกต่างกับแบบ 1 มิติดังต่อไปนี้

#### 1. สามารถสแกนได้รอบทิศทาง

รหัสแถบทั้งประเภทสแควร์โค้ดและเมทริกซ์โค้ดสามารถอ่านได้โดยการใช้เครื่องซีซีดีสแกนเนอร์ ซึ่งมีคุณสมบัติสแกนได้รอบทิศทาง ความจริงแล้วรหัสแถบแบบ 1 มิติก็สามารถสแกนได้รอบทิศทางเช่นกัน โดยใช้เครื่องเลเซอร์สแกนเนอร์

#### 2. เป็นพอร์เทเบิลดาต้าไฟล์ (Portable Data File; PDF)

หรือเพิ่มข้อมูลเคลื่อนที่ ซึ่งหมายถึง ข้อมูลทั้งหมดได้ถูกบันทึกอยู่ในรหัสแถบแล้ว ต่างกับรหัสแถบแบบ 1 มิติที่เป็นเพียงตัวเรียกข้อมูลซึ่งจะต้องถูกเก็บในหน่วยความจำที่แยกต่างหาก

### 3. มีรหัสตรวจจับและแก้ไขความผิดพลาด (Error detection and correction)

รหัสแถบ 2 มิติเกือบทุกชนิดจะมีรหัสตรวจจับและแก้ไขความผิดพลาดในกรณีที่รหัสแถบเสียหาย โดยจะถูกเข้ารหัสพร้อมกับข้อมูลและเป็นส่วนหนึ่งของรหัสแถบ ระดับของการแก้ไขความผิดพลาด (Error correction) จะสูงหรือต่ำขึ้นอยู่กับจำนวนข้อมูลที่ถูกรหัส คุณภาพของการพิมพ์รหัสแถบ สภาพและระยะเวลาการใช้งาน ซึ่งยิ่งระดับการแก้ไขความผิดพลาดสูง ก็ยิ่งเพิ่มขนาดของรหัสแถบให้ใหญ่ขึ้นไปด้วย การแก้ไขความผิดพลาดของรหัสแถบ ก็คือ การเพิ่มข้อมูลเดิมซ้ำลงไปบนรหัสแถบ จึงมีอีกชื่อหนึ่งว่า การซ้ำข้อมูล (Data Redundancy)

#### 2.2 รหัสแถบชนิด MicroPDF417

คิดค้นโดยบริษัท Symbol Technologies รหัสแถบชนิดนี้จัดอยู่ในประเภทสแควร์โค้ด ซึ่งถูกพัฒนาจากรหัสแถบชนิด PDF417 ให้มีขนาดเล็กลงและสามารถประยุกต์ใช้กับงานที่มีพื้นที่เล็กมาก ๆ ได้ ดังนั้น MicroPDF417 จึงมีคุณสมบัติและข้อกำหนดเหมือนกับชนิด PDF417 ทุกประการ ยกเว้นจะสามารถเก็บข้อมูลได้น้อยกว่า และจะต้องมีจำนวนแถว คอลัมน์ และรหัสแก้ไขความผิดพลาดที่สัมพันธ์กัน

##### 2.2.1 รูปแบบของ MicroPDF417

โครงสร้างของรหัสแถบชนิดนี้ สามารถแสดง ได้ดังรูป

Quiet Zone	RAP Column	Codeword Column	Codeword Column	RAP Column	Stop Row	Quiet Zone
------------	------------	-----------------	-----------------	------------	----------	------------

โครงสร้างแบบ 2 คอลัมน์ข้อมูล

Quiet Zone	Codeword Column	RAP Column	Codeword Column	Codeword Column	RAP Column	Stop Row	Quiet Zone
------------	-----------------	------------	-----------------	-----------------	------------	----------	------------

โครงสร้างแบบ 3 คอลัมน์ข้อมูล

Quiet Zone	RAP Column	Codeword Column	Codeword Column	RAP Column	Codeword Column	Codeword Column	RAP Column	Stop Row	Quiet Zone
------------	------------	-----------------	-----------------	------------	-----------------	-----------------	------------	----------	------------

โครงสร้างแบบ 4 คอลัมน์ข้อมูล

#### รูปที่ 2.3 โครงสร้างของรหัสแถบชนิด MicroPDF417

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.3 สามารถอธิบายรายละเอียดของส่วนต่าง ๆ ได้ดังนี้

- Quiet Zone  
คือ บริเวณที่ไม่มีสีคำอยู่เลย ซึ่งจะอยู่บริเวณขอบด้านหน้าและด้านหลังของรหัสแถบ ใช้เป็นพื้นที่เผื่อไว้ในการพิมพ์รหัสแถบ
- RAP (Row Addressing Pattern) Column  
เป็นคอลัมน์เริ่มต้นของรหัสแถบ ใช้กำหนดรูปแบบของรหัสแถบนั้น ๆ เพื่อให้เครื่องสแกนรหัสแถบทราบว่ารหัสแถบนั้นมีกี่แถว ก็คอลัมน์ เป็นต้น
- Codeword Column หรือคอลัมน์ของข้อมูล  
โค้ดเวิร์ด เปรียบได้กับ 1 ตัวอักษรของรหัสแถบ สามารถเป็นได้ทั้งสีขาวและสีดำ ซึ่งจะมีจำนวนตามชนิดและจำนวนตัวอักษรของข้อมูลที่เข้ารหัสบนรหัสแถบนั้น ตารางที่ 2.1 แสดงชนิดของตัวอักษรที่ต้องการเข้ารหัส กับจำนวนของโค้ดเวิร์ดที่ต้องใช้ในการเข้ารหัสข้อมูล 1 ตัวอักษร

Character Type	Characters	Codewords
Alpha	1	.5263
Numeric	1	.3448
ASCII	1	.8333

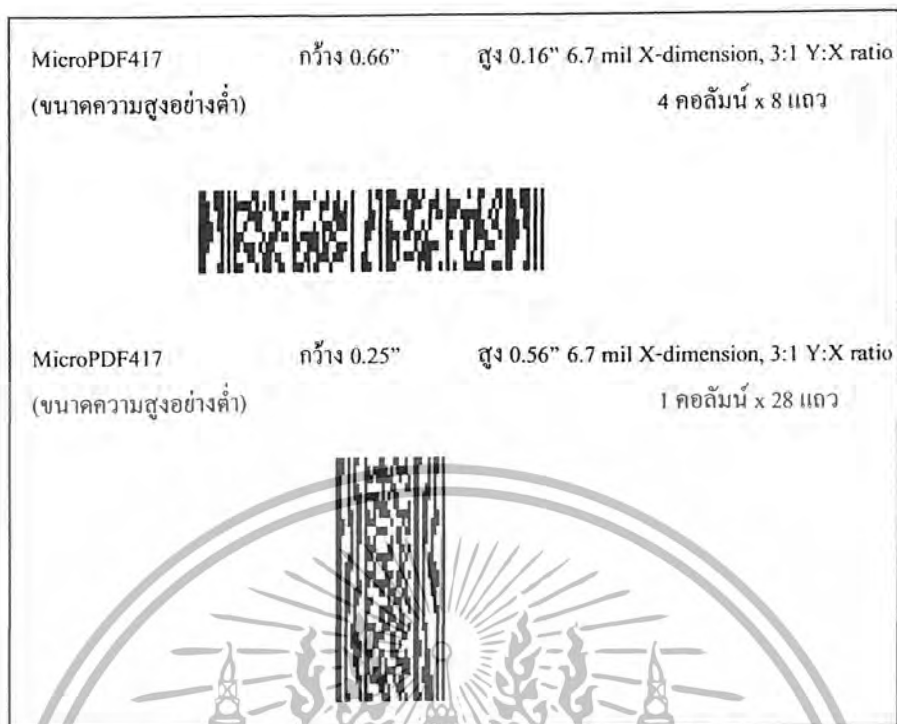
ตารางที่ 2.1 แสดงชนิดของตัวอักษรและจำนวนโค้ดเวิร์ดที่ใช้

รหัสแถบชนิดนี้สามารถมีได้หลายขนาด โดยที่ขนาดของรหัสแถบขึ้นอยู่กับชนิดและจำนวนข้อมูลที่ถูกเข้ารหัส, ขนาดความกว้างของโมดูลหรือมิติในแกน X (X-dimension), และ ระดับของการแก้ไขความผิดพลาด ดังตัวอย่างในรูปที่ 2.4

ขนาดความกว้างของโมดูล (Module)<sup>1</sup> หรือมิติในแกน X (X-dimension)<sup>2</sup> จะเป็นตัวกำหนดความสูงของโมดูล (Y-dimension) โดยที่อัตราส่วนความสูงต่อความกว้าง (Aspect Ratio) Y : X ของทุกโมดูลจะต้องมีค่าอย่างน้อย 3 : 1

<sup>1</sup> โมดูล (Module) คือ โค้ดเวิร์ดที่แคบที่สุดของรหัสแถบ

<sup>2</sup> X-dimension คือ ความกว้างของ 1 โมดูล



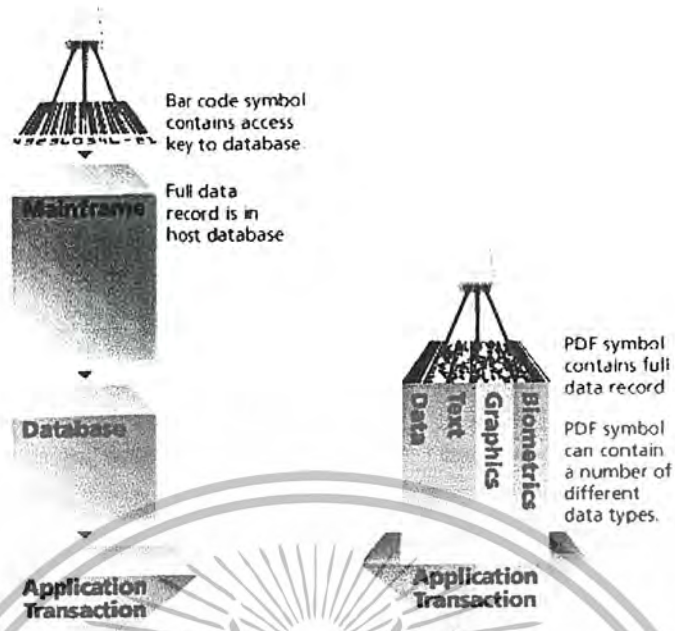
รูปที่ 2.4 ตัวอย่างของ MicroPDF417 ขนาดต่าง ๆ

โดยทั่วไป มีการกำหนดค่าพารามิเตอร์ของ MicroPDF417 ดังนี้

- Columns : จำนวนคอลัมน์ข้อมูล (1 - 4)
- X-unit : ความกว้างของ 1 โมดูล (รหัสที่เล็กที่สุด) ในหน่วย mils (4 - 200)
- Bar Height/Width : อัตราส่วนของความสูงต่อความกว้างของรหัสแถบ (2 - 5)

### 2.2.2 คุณสมบัติของ MicroPDF417

1. ต้องมีจำนวนแถว คอลัมน์ และรหัสแก้ไขความผิดพลาดที่สัมพันธ์กันตามที่กำหนด ซึ่งสามารถมีสูงสุดได้ 4 คอลัมน์ x 44 แถว
2. สามารถเก็บข้อมูลได้สูงถึง 150 ไบท์ หรือ 250 ตัวอักษร หรือ หมายเลข 366 หลัก
3. เป็นพอร์เทเบิลค้ำไพล์ ที่สามารถทำการอ่านข้อมูลจากรหัสแถบได้เลย โดยไม่ต้องทำการเก็บข้อมูลลงในหน่วยความจำเพิ่มเติมเหมือนรหัสแถบ 1 มิติ และข้อมูลทำการเข้ารหัสยังสามารถเป็นรูปภาพ ตัวอักษร ตัวเลข หรือข้อมูลอื่น ๆ เช่นข้อมูลเสียงได้ ความแตกต่างของรหัสแถบ 1 มิติกับรหัสแถบที่เป็นพอร์เทเบิลค้ำไพล์สามารถแสดงได้ดังรูป



- รูปที่ 2.5 ความแตกต่างของรหัสแถบ 1 มิติกับรหัสแถบที่เป็นพอร์เทเบิลโค้ดไฟล์
4. มีความทนทานต่อการสูญหายของข้อมูลอื่นเนื่องมาจากความเสียหายของรหัสแถบ โดยที่รหัสแถบสามารถเสียหายได้ถึงเกือบครึ่งของพื้นที่และยังสามารถถอดรหัสออกมาได้อย่างถูกต้อง
  5. สามารถใช้เทคนิคการพิมพ์ได้เหมือนกันรหัสแถบ 1 มิติ

### 2.3 ประโยชน์และการประยุกต์ใช้งานรหัสแถบ 2 มิติ

ในปัจจุบันได้มีการใช้งานรหัสแถบแบบ 2 มิติเพิ่มมากขึ้น ตัวอย่างการนำไปใช้งานของรหัสแถบชนิดนี้มีดังต่อไปนี้

1. ชีบ่งผลิตภัณฑ์ที่มีขนาดเล็ก
2. ใช้เก็บข้อมูลที่ต้องการเคลื่อนย้าย โดยเฉพาะไปยังพื้นที่ที่ห่างไกล คือใช้งานเป็นพอร์เทเบิลโค้ดไฟล์
3. ใช้ในการจัดเรียงสินค้าและพัสดุที่จะขนส่ง เช่น บริษัท UPS ที่ได้มีการใช้งานรหัสแถบแบบ MaxiCode เพื่อใช้ระบุจุดหมายปลายทาง หมายเลขการขนส่ง และหมายเลขพัสดุที่จะทำการส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ใช้บัตรเครดิตประจำตัวของพนักงานหรือบุคคลทั่วไป ดังเช่นในหน่วยงาน U.S. Department of Defense (DOD) ซึ่งได้ออกบัตรเครดิตประจำตัวที่ใช้รหัสแถบชนิด PDF417 ให้แก่นายทหารไปแล้วมากกว่า 4 ล้านบัตร โดยข้อมูลที่ถูกเข้ารหัสบนบัตรได้แก่ ชื่อ ยศ หมายเลขประจำตัว สิทธิพิเศษในร้านสวัสดิการทหาร และรูปถ่าย นอกจากนี้ยังได้มีการใช้กับใบอนุญาตขับรถในหลายประเทศอีกด้วย



ใบอนุญาตขับรถของประเทศฟิลิปปินส์ที่ใช้รหัส PDF417

รูปที่ 2.6 ตัวอย่างการประยุกต์ใช้งานรหัสแถบ 2 มิติ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### บทที่ 3

## หลักการเบื้องต้นและทฤษฎีที่ใช้ในการวิเคราะห์ภาพ

### 3.1 ระบบการมองเห็นภาพ (Vision System)

ระบบการมองเห็นภาพ (Vision System) นั้นจะหมายความรวมถึงทุกสิ่งที่จะต้องการได้มาซึ่งรหัสดิจิทัลที่ใช้แทนภาพนั้น ๆ การปรับปรุงเปลี่ยนแปลงข้อมูล และรวมทั้งการนำเสนอภาพที่ได้หลังการปรับปรุงแล้วด้วย ซึ่งระบบการมองเห็นภาพนี้อาจมีความยุ่งยากซับซ้อนต่างกันตามลักษณะการใช้งาน แต่ก็สามารถแบ่งเป็นขั้นตอนกว้าง ๆ ได้ 3 ขั้นตอน คือ การได้มาซึ่งภาพ (Image Acquisition), กระบวนการประมวลผลภาพ (Image Processing), ผลที่ได้และการแสดงผล (Output or Display)

#### 3.1.1 การได้มาซึ่งภาพ (Image Acquisition)

การได้มาซึ่งภาพนี้จะหมายถึง การแปลงภาพที่เราเห็นในลักษณะทางกายภาพนั้น ให้เป็นเซตของข้อมูลทางดิจิทัล ซึ่งจะอยู่ในรูปของเมทริกซ์ (Matrix) ซึ่งเซตของข้อมูลนี้จะถูกส่งไปยังหน่วยประมวลผลต่อไป การได้มาซึ่งภาพสามารถพิจารณาได้เป็น 4 ขั้นตอน คือ

##### - การส่องสว่าง (Illumination)

การส่องสว่างเป็นตัวแปรสำคัญที่มีผลกระทบต่ออินพุทที่จะป้อนให้แก่ระบบการมองเห็นภาพ เราจำเป็นต้องออกแบบการส่องสว่างให้เหมาะสมกับการใช้งานแต่ละประเภท ทั้งนี้ ชนิดและวิธีการของแหล่งกำเนิดแสงจะมีผลต่อกำลังงานแสง และจะส่งผลกระทบต่อกระบวนการประมวลผลภาพและผลลัพธ์ที่ได้

- รูปแบบของภาพหรือการโฟกัสภาพ (Image formation or Focusing)
- การตรวจจับภาพหรือการเซนซิง (Image detection or Sensing)
- รูปแบบผลของสัญญาณที่ได้จากกล้อง (Formatting camera output signal)

#### 3.1.2 กระบวนการประมวลผลภาพ (Image Processing)

หน้าที่พื้นฐานของกระบวนการประมวลผลภาพ ได้แก่ การสร้างภาพใหม่โดยแยกแยะส่วนที่ต้องการหรือสนใจกับสิ่งรบกวนออกจากกัน (Noise Elimination), การตรวจหาขอบภาพ (Edge Detection), และการกรอง(Filtering) เป็นต้น



### 3.2.2 เกรย์สเกล (Gray scale)

หากเราต้องการค่าข้อมูลที่ละเอียดมากขึ้น ก็จำเป็นที่จะต้องเพิ่มจำนวนบิตในการแสดงค่าของแต่ละพิกเซล ยกตัวอย่างเช่น หากแบ่งความเข้มของการส่องสว่างให้มี 4 ระดับ ก็ต้องใช้เลขฐานสอง 2 บิต และ 4 บิตสำหรับ 16 ระดับ, 8 บิตสำหรับ 256 ระดับ ซึ่งจำนวนระดับที่ใช้ในเกรย์สเกลนี้มักเป็นเลขยกกำลังของ 2 ค่าต่ำสุดคือ 0 กำหนดให้เป็นสีดำ และ 1 หรือตัวเลขที่น้อยกว่าค่าสูงสุดของเกรย์สเกลอยู่ 1 (เช่น 15 สำหรับเกรย์สเกล 16 ระดับ) แทนสีขาว ค่าที่กำหนดให้ในแต่ละพิกเซลมักเป็นจำนวนเต็ม

ในยุคแรกๆ ของระบบการมองเห็นภาพ (Vision System) จะใช้ระบบเลขฐานสอง แต่ในปัจจุบันเทคโนโลยีไมโครโปรเซสเซอร์เข้ามามีบทบาทมากขึ้น การแบ่งระดับเป็น 16, 64 หรือ 256 เป็นเรื่องธรรมดา แต่ทั้งนี้ในการมองเห็นของมนุษย์ จะสามารถแยกแยะความแตกต่างได้เพียง 10-15 ระดับเท่านั้น การแบ่งโดยละเอียดเป็น 64 หรือ 256 ระดับ อาจจะไปประยุกต์ใช้กับงานการประมวลผลภาพหรือกระบวนการอุตสาหกรรมอื่น ๆ

จะเห็นว่าจำนวนเกรย์สเกลจะเป็นตัวจำกัดรายละเอียดของภาพ โดยทั่วไปแล้ว ยิ่งแบ่งเกรย์สเกลเป็นหลายระดับก็จะเป็นการเพิ่มคุณภาพของภาพด้วย และการเพิ่มจำนวนพิกเซล เช่น จาก  $35 \times 35$  ไปเป็น  $256 \times 256$  ก็จะเป็นการเพิ่มความละเอียด (Resolution) และรายละเอียด (Detail) ของภาพเช่นกัน จะเห็นว่าจะแตกต่างจากการขยายภาพ (Zoom) คือ การZoomเป็นการเพิ่มการขยายของแต่ละพิกเซลให้ใหญ่ขึ้น ไม่ได้เป็นการเพิ่มจำนวน

### 3.2.3 ฮิสโตแกรม (Histogram)

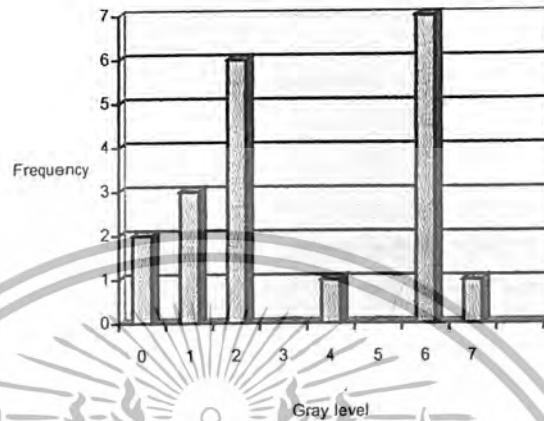
ฮิสโตแกรม เป็นการแสดงให้เห็นถึงความถี่ของการนับจำนวนพิกเซล ที่ความเข้มของแสงแต่ละค่าของรูปภาพเกรย์สเกลนั้น ๆ โดยแกน x ในกราฟแสดงค่าเกรย์สเกล ส่วนแกน y แสดงจำนวนพิกเซล

ขั้นตอนการสร้างฮิสโตแกรมมีดังนี้

1. การทำดิจิทัลภาพ (Digitizing)
2. นับจำนวนพิกเซลในแต่ละค่าความเข้มแสงของเกรย์สเกล
3. วาดกราฟระหว่างจำนวนพิกเซลกับระดับเกรย์สเกล

รูปร่างของฮิสโตแกรมสามารถบอกลักษณะบางประการของภาพได้ เช่น หากฮิสโตแกรมมีลักษณะที่แคบ ก็จะหมายถึงการขาดคอนทราสต์ (Contrast) ในภาพ

นอกจากนี้ ฮิสโตแกรมยังมีประโยชน์มากในการกำหนดค่าเทรชโฮล (Threshold) เพื่อใช้แปลงจากภาพแบบเกรย์สเกลให้เป็นไบนารี (Binary) หรือภาพขาว-ดำ และยังใช้ในการปรับปรุงบางส่วนของสเปกตรัมของเกรย์สเกลได้อีกด้วย



รูปที่ 3.2 ฮิสโตแกรมของเกรย์สเกล 8 ระดับจากเมทริกซ์ 4 x 5

### 3.3 การทำเทรชโฮล (Thresholding)

การทำเทรชโฮล เป็นวิธีหนึ่งในการแยกส่วนภาพ (Segmentation) ให้ออกเป็น 2 ส่วนใหญ่ ๆ คือ ส่วนที่เราต้องการหรือสิ่งที่เราสนใจ (Object) กับส่วนที่เป็นพื้นหลัง (Background) โดยกำหนดค่าของเกรย์สเกลที่แน่นอนขึ้นมาเพื่อให้เป็นจุดตัด เพื่อแบ่งภาพบริเวณที่มีค่าเกรย์สเกลที่เราไม่ต้องการออกจากบริเวณที่ต้องการ ค่าที่ใช้เป็นจุดตัดนั้นอาจเรียกได้ว่าค่าเทรชโฮล (Threshold)

การทำเทรชโฮลโดยส่วนใหญ่แล้ว จะใช้ในกรณีที่ต้องการแปลงจากภาพแบบเกรย์สเกลไปเป็นภาพไบนารี พิกเซลที่มีค่าระดับเกรย์สเกลต่ำกว่าค่าเทรชโฮลจะถูกเปลี่ยนให้เป็น '0' หรือเป็นสีดำ ส่วนพิกเซลที่มีค่าระดับเกรย์สเกลสูงกว่าค่าเทรชโฮลจะถูกเปลี่ยนให้เป็น '1' หรือเป็นสีขาว ดังสมการที่ (3.1) ซึ่งเป็นการเลือกค่าเทรชโฮลแบบเทรชโฮล 1 ค่า (Single Threshold)

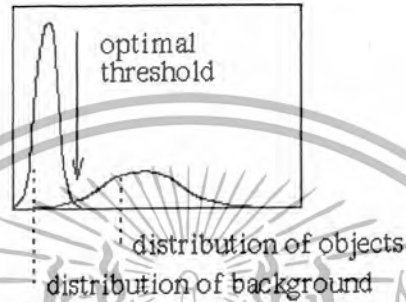
$$g(i, j) = \begin{cases} 1 & \text{for } f(i, j) \geq T \\ 0 & \text{for } f(i, j) < T \end{cases} \quad (3.1)$$

โดยที่  $g(i, j)$  คือ ภาพไบนารี

$f(i, j)$  คือ ภาพเกรย์สเกล

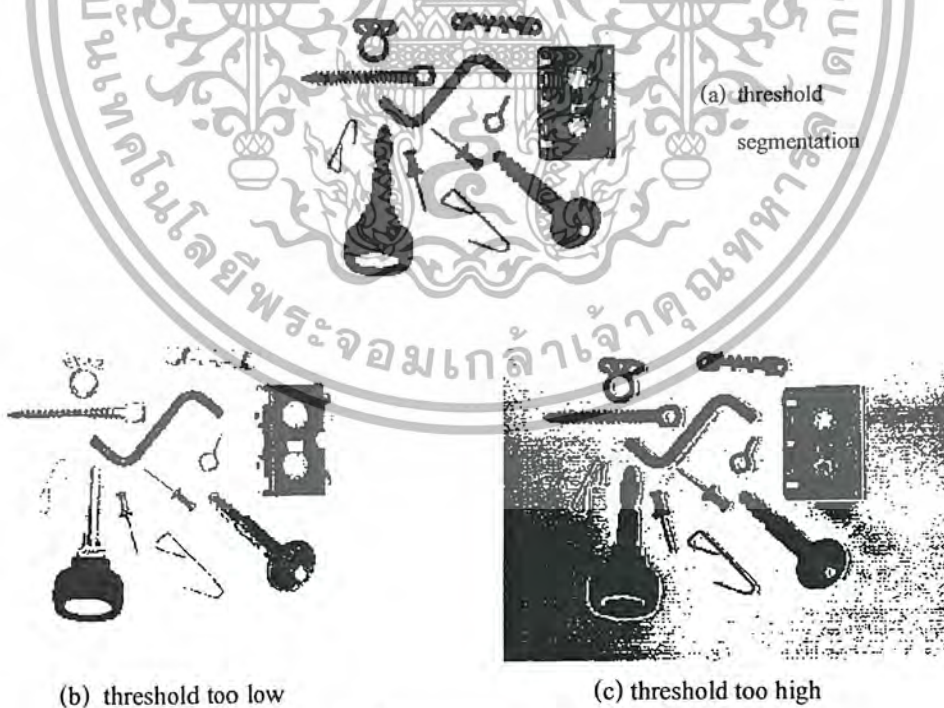
$T$  คือ ค่าเทรชโฮล

การเลือกค่าเทรชโฮลที่ดีที่สุด (Optimal threshold) มีความสำคัญมาก ซึ่งค่าเทรชโฮลที่เหมาะสมของแต่ละภาพก็จะมีค่าแตกต่างกันไป โดยทั่วไป เราสามารถหาค่าเทรชโฮลอย่างง่าย ๆ ได้โดยดูจากฮิสโตแกรม จากรูปที่ 3.3 แสดงฮิสโตแกรมของภาพ ๆ หนึ่ง โดยที่การกระจาย (Distribution) ของพิกเซลแบ่งออกเป็น 2 กลุ่ม ได้แก่กลุ่มที่เป็นพื้นหลัง (background) กับกลุ่มที่ต้องการ (objects) โดยค่าเทรชโฮลจะถูกเลือก ณ บริเวณจุดตัด หรือบริเวณที่เป็นจุดต่ำสุดระหว่าง 2 กายอดนั่นเอง



รูปที่ 3.3 การเลือกค่าเทรชโฮลแบบเทรชโฮล 1 ค่า

ผลของการเลือกค่าเทรชโฮลสามารถดูได้จากรูปที่ 3.4



รูปที่ 3.4 ผลของการเลือกค่าเทรชโฮล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนี้ยังมีการกำหนดใช้ค่าเทรชโฮลหลายค่าต่อหนึ่งภาพได้ด้วย ซึ่งเรียกว่า Multi-Thresholding ซึ่งผลลัพธ์ที่ได้จะไม่ใช่ภาพไบนารี แต่อาจเป็นภาพเกรย์สเกลหรืออื่น ๆ

$$\begin{aligned}
 g(i, j) &= 1 && \text{for } f(i, j) \in D_1 \\
 &= 2 && \text{for } f(i, j) \in D_2 \\
 &= 3 && \text{for } f(i, j) \in D_3 \\
 &= 4 && \text{for } f(i, j) \in D_4 \\
 &\dots && \\
 &= n && \text{for } f(i, j) \in D_n \\
 &= 0 && \text{otherwise}
 \end{aligned}
 \tag{3.2}$$

### 3.4 การตรวจจับขอบภาพ (Edge Detection)

เครื่องมือที่ช่วยในการตรวจจับขอบภาพ สามารถแบ่งออกได้เป็น 2 ประเภท คือ

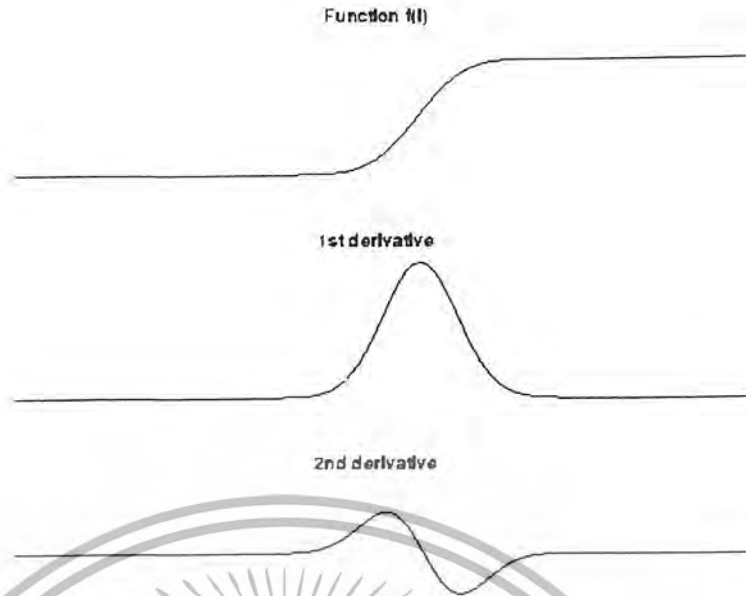
- 1.) ตัวตรวจจับพิกเซลขอบ (Edge Pixel Detectors) ซึ่งจะค้นหาพิกเซลของภาพที่เป็นส่วนหนึ่งของขอบภาพ
- 2.) ตัวตรวจจับเส้นขอบ (Edge Line Detectors) จะค้นหาพิกเซลของภาพที่เป็นส่วนหนึ่งของเส้นในภาพ

#### 3.4.1 พื้นฐานของเครื่องมือตรวจจับขอบภาพ (Edge Detectors)

ขอบภาพ (Edge) เป็นสิ่งที่อยู่ในภาพ ๆ หนึ่ง ซึ่งจะมีความแตกต่างของสีหรือคอนทราสต์ (Contrast) สูง การตรวจจับขอบภาพเป็นที่นิยมเพื่อใช้สำหรับการแยกของสิ่งต่าง ๆ บนภาพออกจากกันเป็นส่วน ๆ (Image Segmentation) นอกจากนี้ การแยกส่วนภาพโดยใช้ขอบภาพนี้ยังมีจุดเด่นคือ จำนวนข้อมูลจะลดลงอย่างมากโดยที่องค์ประกอบของภาพที่ทำการตรวจจับขอบภาพยังอยู่ครบถ้วน

เนื่องจากขอบภาพเป็นบริเวณที่มีความถี่สูงเป็นส่วนมาก ดังนั้น ตามทฤษฎีแล้ว เราสามารถใช้ตัวกรองความถี่สูง (High pass filter) แต่ในทางปฏิบัติแล้ว การตรวจจับขอบภาพจะกระทำในสเปซโดเมน (Spatial domain)

และเนื่องจากขอบภาพเป็นบริเวณที่มีเกรเดียนต์ (Gradient) การส่องสว่างสูง ดังนั้นเราสามารถใช้อนุพันธ์เฟอเรนเชียล (Differential) หรือเดริเวทีฟ (Derivatives) เพื่อคำนวณหาขอบภาพได้ รูปต่อไปนี้ แสดงกรณีการใช้เดริเวทีฟ กับเวกเตอร์ 1 มิติ



รูปที่ 3.5 การใช้เกรเดียนต์ลำดับที่ 1 และ 2 ใน 1 มิติ

ดังนั้น จากรูปจะเห็นว่าจุดที่เป็นขอบภาพจะเป็นตำแหน่งสูงสุดในการเกรเดียนต์ลำดับที่ 1 (1st Derivative) หรือเป็นจุดตัดแกนอนที่ 0 (Zero-crossing) ในการเกรเดียนต์ลำดับที่ 2 (2nd Derivative) แต่สำหรับรูปภาพ ซึ่งเป็น 2 มิติ เราจะต้องหาเทคนิคที่การเกรเดียนต์ใน 2 มิติ สำหรับฟังก์ชันดิสครีต 1 มิติ (Discrete one-dimensional function),  $f(i)$  การเกรเดียนต์อันดับ 1 หาได้จาก

$$\frac{d f(i)}{d(i)} = f(i+1) - f(i) \quad (3.3)$$

จากสมการข้างต้น ทำให้เกิดมาสก์ (Mask) ที่ใช้ในการตรวจจับขอบภาพ เพื่อใช้คำนวณหาค่าเกรเดียนต์ลำดับ 1 กับเกรเดียนต์ลำดับ 2 และสำหรับการทำเกรเดียนต์อันดับ 1 กับรูป 2 มิติ มีมาสก์ที่เป็นที่นิยมอยู่ 2 ประเภท คือ แบบพรีวิตคอมพาส (Prewitt compass edge detection) กับ แบบเกรเดียนต์ (Gradient edge detection)

แบบพรีวิตคอมพาสโดยปกติจะใช้ 8 มาสก์ในการตรวจจับขอบภาพ ซึ่งแต่ละมาสก์จะสามารถตรวจจับทิศทางที่ต่างกันของขอบภาพและขนาดของขอบ

ถ้าสำหรับแบบเกรเดียนต์ซึ่งใช้มาสก์เพียง 2 มาสก์ จะเป็นที่นิยมมากกว่า มาสก์หนึ่งจะใช้หาค่าเกรเดียนต์ในแนวแกน x ( $G_x$ ) อีกหนึ่งมาสก์จะใช้หาเกรเดียนต์ในแนวแกน y ( $G_y$ ) ผลรวมของเกรเดียนต์ทั้งสองหาได้จาก

$$|G| = \sqrt{G_x^2 + G_y^2} \quad (3.4)$$

และโดยปกติจะประมาณค่าเป็น

$$|G| = |G_x| + |G_y| \tag{3.5}$$

ในการนำไปใช้งาน ปกติแบบเกรเดียนต์จะให้แค่ค่าขนาดของขอบภาพเท่านั้น แต่ก็สามารถคำนวณหาทิศทางของขอบภาพได้จาก

$$\theta = \arctan(G_y/G_x) - 3\pi/4 \tag{3.6}$$

มาตรฐานที่เป็นที่นิยมสำหรับแบบเกรเดียนต์ ได้แก่วิธีของ Sobel, Roberts Cross และ Prewitt

หลังจากคำนวณหาค่าขนาดของเกรเดียนต์จากการใช้เคอริเวทีฟลำดับที่ 1 แล้ว จะต้องทำการหาตำแหน่งของพิกเซลที่จะเป็นส่วนหนึ่งของขอบภาพด้วย วิธีที่ง่ายที่สุดคือการทำเทรซโซลในรูปที่ผ่านการหาเกรเดียนต์มาแล้ว โดยให้พิจารณาว่าพิกเซลที่มีค่าเกรเดียนต์มากกว่าค่าเทรซโซลเป็นส่วนหนึ่งของขอบภาพ ซึ่งโดยทั่วไปจะให้ค่าความกว้างของขอบเพียง 1 พิกเซลเท่านั้น ยังมีวิธีที่ดีกว่าในการหาค่าขอบภาพ คือ ตัวตรวจจับขอบภาพแบบแคณนี่ (Canny edge detector) ซึ่งใช้เกรเดียนต์ก่อน แล้วจึงทำการหาตำแหน่งพิกเซลของขอบภาพจากการใช้ non-maximal suppression และ hysteresis tracking

สำหรับการหาขอบภาพโดยใช้เคอริเวทีฟลำดับที่ 2 นั้น ได้แก่วิธีของมาร์ (Marr edge detector) หรือที่รู้จักในชื่อ Zero Crossing Detector ซึ่งเคอริเวทีฟลำดับที่ 2 นั้นคำนวณหาได้จาก ตัวกรองความถี่ลาปลาเซียนของเกาส์เซียน (Laplacian of Guassian filter; LoG filter) การใช้ลาปลาเซียนมีข้อดีคือ สามารถหาลำโพงไอโซทรอปิก (Isotropic) ของภาพได้ ตัวอย่างเช่น ขนาดขอบภาพสามารถหาได้อย่างอิสระจากทิศทางของขอบภาพโดยใช้เพียงแค่ 1 มาตรฐานเท่านั้น วิธีนี้หาตำแหน่งของขอบภาพได้จากจุดตัดแกนที่ 0 (Zero crossing) ในภาพจาก LoG ความหนาของขอบจะหนาและควบคุมได้จากการเปลี่ยนแปลงค่าเกาส์เซียน

ปัญหาที่สำคัญของการตรวจจับขอบภาพคือจะอ่อนไหวต่อสิ่งรบกวน (Noise) มาก โดยสิ่งรบกวนจะถูกลบขยายขนาดขึ้นเมื่อทำการคำนวณหาเคอริเวทีฟ ดังนั้นบางวิธี เช่น วิธีของมาร์ และ แคณนี่ จึงเสนอให้มีการทำภาพให้เรียบก่อนที่จะคำนวณหาเคอริเวทีฟโดยใช้เกาส์เซียน

ในที่นี้จะขออธิบายรายละเอียดเฉพาะวิธีของโซเบลเท่านั้น

### 3.4.2 วิธีการตรวจจับขอบภาพของโซเบล (Sobel Edge Detector)

วิธีการนี้เป็นที่นิยมมากสำหรับการหาขอบภาพแบบเกรเดียนต์ ซึ่งจะเน้นให้เห็นถึงบริเวณของภาพที่มีค่าเกรเดียนต์สูง หรือก็คือบริเวณขอบภาพนั่นเอง

โดยการใช่มาสก์ (Mask หรือ Convolution mask) ที่มีขนาด  $3 \times 3$  ดังรูปที่ 3.6 ซึ่งมาสก์ที่สองจะเป็นมาสก์ที่หนึ่งที่หมุนไป 90 องศา

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

Gx

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

Gy

รูปที่ 3.6 มาสก์ของวิริโซเบล

มาสก์ทั้ง 2 นี้ถูกออกแบบมาเพื่อใช้หาค่าเกรเดียนต์ตามแนวแกนนอนและแกนตั้ง ได้แก่ Gx และ Gy ตามลำดับ มาสก์เหล่านี้สามารถแก้ไขได้อย่างอิสระหากต้องการหาค่าเกรเดียนต์ในแกนใดแกนหนึ่ง ขนาดและมุมของเกรเดียนต์เป็นคังสมการที่ (3.4), (3.5) และ (3.6)

การหาค่าขนาดเกรเดียนต์ตามสมการที่ (3.4), (3.5) นั้นสามารถคำนวณได้ง่ายกว่าหากรวมกันเป็นมาสก์เดียว เพื่อใช้กับภาพอินพุท เรียกมาสก์นี้ว่าซูโคคอนโวลูชันมาสก์ (Pseudo-Convolution mask) ดังรูปที่ 3.7

$P_1$	$P_2$	$P_3$
$P_4$	$P_5$	$P_6$
$P_7$	$P_8$	$P_9$

รูปที่ 3.7 ซูโคคอนโวลูชันมาสก์

จากมาสก์ข้างต้น สามารถคำนวณหาค่าขนาดเกรเดียนต์ได้จากสมการ

$$||G|| = |(P_1 + 2 \times P_2 + P_3) - (P_7 + 2 \times P_8 + P_9)| + |(P_3 + 2 \times P_6 + P_9) - (P_1 + 2 \times P_4 + P_7)| \quad (3.7)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีของโซเบลจะใช้เวลาคำนวณนานกว่าวิธีเกรเดียนต์แบบอื่น ๆ ซึ่งใช้มาสก์ขนาดเล็กกว่าแค่  $2 \times 2$  แต่การที่ใช้มาสก์ขนาดใหญ่มีข้อดีคือ จะไม่อ่อนไหวต่อสิ่งรบกวนมากนัก จึงนับว่าวิธีของโซเบลจะดีกว่าของโรเบิร์ตส์ ครอสส์ (Roberts Cross)

บางครั้งการใช้วิธีของโซเบลจะทำให้เส้นขอบภาพที่หาได้ค่อนข้างหนา ดังนั้นการทำให้เส้นขอบเล็กลง (Thinning) จึงจำเป็นในบางกรณี

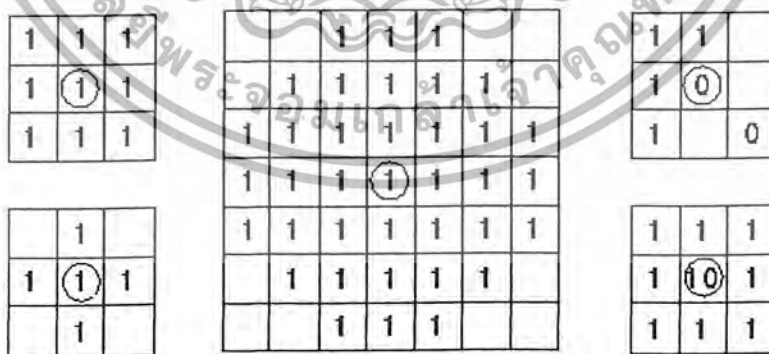
### 3.5 มอร์โฟโลยี (Morphology)

มอร์โฟโลยี หรือ Mathematical Morphology ถูกประยุกต์ใช้ในงานที่เกี่ยวข้องกับการประมวลผลภาพดิจิทัลหลายแบบ เช่น การตรวจจับขอบภาพ (Edge detection), การกำจัดสิ่งรบกวน (Noise removal), การปรับคุณภาพของภาพ (image enhancement) และ การแยกส่วนภาพ (Image segmentation) โดยเฉพาะจะถูกใช้งานมากกับภาพไบนารี

กระบวนการพื้นฐานของมอร์โฟโลยีคือ ไดเลชัน (Dilation) และอีโรชัน (Erosion) โดยจะรับอินพุต 2 อินพุต คือ รูปภาพที่ต้องการทำมอร์โฟโลยี กับ ส่วนโครงสร้าง (Structuring element) หรือเคอร์เนล (Kernel)

โดยทั่วไปภาพอินพุตจะแบ่งเป็น 2 ส่วนสำหรับภาพไบนารี คือ ส่วนสีขาว เป็นส่วนของฉากหน้า (Foreground) กับส่วนสีดำ เป็นส่วนของฉากหลัง (Background) ในบางงานอาจสลับกัน โดยให้สีดำเป็นฉากหน้า ส่วนสีขาวเป็นพื้นหลังก็ได้

ส่วนโครงสร้างนี้ จะเป็นส่วนที่นำไปกระทำกับภาพอินพุต ซึ่งจะมีโครงสร้างเป็นเซตของไบนารี (โดยทั่วไปอาจเป็นภาพไบนารีก็ได้) แต่ส่วนโครงสร้างนี้จะเล็กกว่ามาก และในงานส่วนใหญ่จะถูกกำหนดให้มีรูปร่างที่แน่นอน เช่น เป็น สี่เหลี่ยมจัตุรัส (square), รูปเพชร (diamond), หรือ รูปวงกลม (ball) เป็นต้น ดังแสดงในรูปที่ 3.8



รูปที่ 3.8 ตัวอย่างส่วนโครงสร้าง (structuring element)

มอร์โฟโลยีมีหลักการทำงานคือ จะนำส่วนโครงสร้างมาทาบลงบนภาพอินพุทในแต่ละพิกเซลโดยใช้จุดกึ่งกลางเป็นจุดกำหนดการทาบ แล้วทำการเปรียบเทียบแต่ละพิกเซลของทั้งสองอินพุท ซึ่งผลลัพธ์จะขึ้นอยู่กับชนิดของกระบวนการที่ใช้ ดังที่จะอธิบายรายละเอียดต่อไป

3.5.1 กระบวนการไคเลชั่น (Dilation)

หรือการไคเลท (Dilate) อธิบายได้อย่างง่าย ๆ ว่า ไคเลชั่นคือการเพิ่มจำนวนพิกเซลในส่วนฉากหน้าของภาพอินพุท ซึ่งมีผลทำให้พื้นที่ของฉากหน้านั้น ๆ ขยายใหญ่ขึ้น และรูหรือส่วนสีดำบนฉากหน้าสีขาวเล็กลงหรือหายไป

หากจะอธิบายให้เป็นหลักการสำหรับความหมายของการ ไคเลชั่น ก็คือ สมมุติให้ X แทนภาพอินพุทที่ต้องการทำมอร์โฟโลยี

K แทนส่วนโครงสร้าง

$K_x$  แทนส่วนโครงสร้างที่ทาบลงบนภาพอินพุท โดยที่จุดกึ่งกลางของส่วนโครงสร้างอยู่ ณ ตำแหน่ง x ของภาพอินพุท

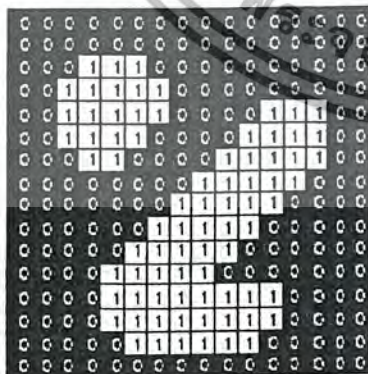
ไคเลชั่นของ X โดย K หาได้จาก จำนวนองค์ประกอบทั้งหมดของ x ที่การอินเตอร์เซก (Intersection) ของ  $K_x$  กับ X ไม่เป็นศูนย์

หมายถึง หาก ณ พิกเซล x ที่ทาบ  $K_x$  ลงไปมีค่าไม่เป็นศูนย์ ก็จะทำให้เกิดการถ่ายรูปแบบส่วนโครงสร้าง K ลงบน X

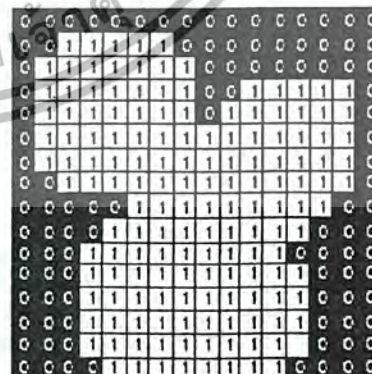
ตัวอย่างเช่นใช้ส่วน โครงสร้างรูปสี่เหลี่ยมจัตุรัสขนาด 3 x 3 รูปที่ 3.9 (a) กับภาพอินพุทดังรูปที่ 3.9 (b) จะ ได้ผลของการ ไคเลชั่น ดังรูปที่ 3.9 (c)

1	1	1
1	1	1
1	1	1

(a) ส่วน โครงสร้างรูปสี่เหลี่ยมจัตุรัส 3 x 3



(b) อินพุท



(c) เอาท์พุท

รูปที่ 3.9 ตัวอย่างการไคเลชั่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.2 กระบวนการอีโรชั่น (Erosion)

หรือการอีโรด (Erode) เป็นการทำให้จำนวนพิกเซลในฉากหน้าของภาพอินพุท น้อยลง บริเวณรูและพื้นหลังจะขยายใหญ่ขึ้น

สมมติให้  $X$  แทนภาพอินพุทที่ต้องการทำมอร์โฟโลยี

$K$  แทนส่วนโครงสร้าง

$Kx$  แทนส่วนโครงสร้างที่ทาบบนภาพอินพุท โดยที่จุดกึ่งกลางของส่วน โครงสร้างอยู่ ณ ตำแหน่ง  $x$  ของภาพอินพุท

อีโรชั่นของ  $X$  โดย  $K$  หาได้จาก จำนวนองค์ประกอบทั้งหมดของ  $x$  ของ  $Kx$  ที่ เป็นซับเซต (Subset) ของ  $X$

หมายถึง หาก  $x$  พิกเซล  $x$  ที่ทาบบน  $Kx$  ลงไปมีค่าไม่เป็นศูนย์ ตำแหน่งที่พิกเซล ของ  $Kx$  กับ  $X$  มีค่าเท่ากันก็จะถูกคงไว้ พิกเซลที่เหลือจะถูกทำให้มีค่าเท่าพิกเซลของ พื้นหลัง (หรือ 0 สำหรับภาพไบนารี)

ตัวอย่างเช่นใช้ส่วน โครงสร้างรูปสี่เหลี่ยมจัตุรัสขนาด  $3 \times 3$  รูปที่ 3.10 (a) กับ ภาพอินพุทดังรูปที่ 3.10 (b) จะ ได้ผลของการอีโรชั่น ดังรูปที่ 3.10 (c)

1	1	1
1	1	1
1	1	1

(a) ส่วน โครงสร้างรูป สี่เหลี่ยมจัตุรัส  $3 \times 3$



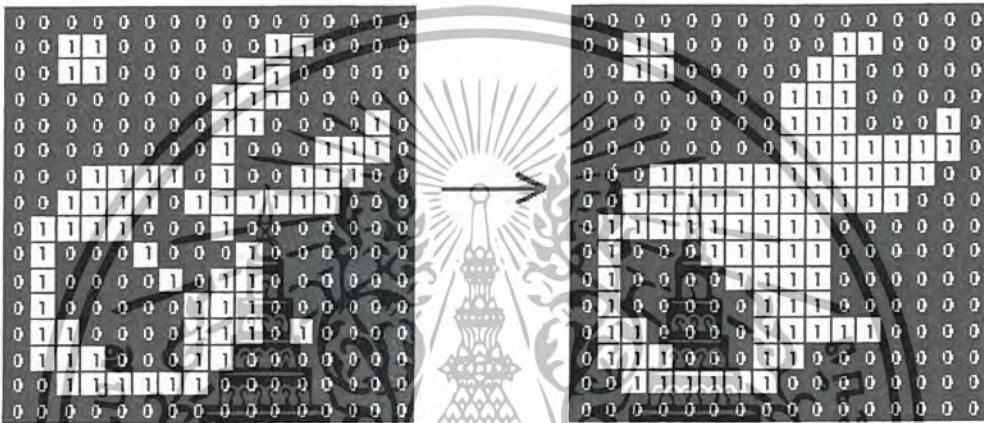
(b) อินพุท

(c) เอ้าท์พุท

รูปที่ 3.10 ตัวอย่างการอีโรชั่น

### 3.5.3 กระบวนการปิด (Closing)

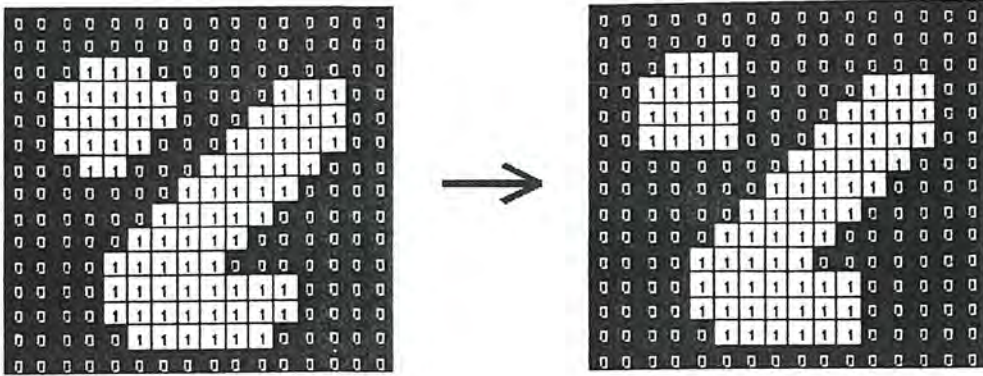
คือ กระบวนการไคเลชันตามคีย์อีโรชัน โดยใช้ส่วนโครงสร้างเหมือนกันทุกประการ ซึ่งมีผลทำให้จำนวนพิกเซลของพื้นหลังลดลง โดยหารูปแบบของกลุ่มพิกเซลของพื้นหลังนั้นเหมือนกันและมีขนาดเท่ากับส่วนโครงสร้างที่ใช้ บริเวณพื้นหลังนั้นก็ยังคงรูปร่างเดิมอยู่ แต่หากกลุ่มพิกเซลนั้นมีรูปแบบไม่เหมือนกัน หรือเล็กกว่าส่วนโครงสร้างที่ใช้ก็จะถูกปิดโดยค่าพิกเซลของฉากหน้า กระบวนการนี้ใช้มากในการปิดรูบนพื้นที่ที่เราสนใจ ตัวอย่างของการปิด แสดงดังรูปต่อไปนี้ ซึ่งใช้ส่วนโครงสร้างเป็นรูปสี่เหลี่ยมจัตุรัส  $3 \times 3$



รูปที่ 3.11 ตัวอย่างการปิด (Closing)

### 3.5.4 กระบวนการเปิด (Opening)

คือ กระบวนการอีโรชันตามคีย์ไคเลชัน โดยใช้ส่วนโครงสร้างเหมือนกันทุกประการ ซึ่งมีผลทำให้จำนวนพิกเซลของฉากหน้าลดลง โดยหารูปแบบของกลุ่มพิกเซลของฉากหน้านั้นเหมือนกันและมีขนาดเท่ากับส่วนโครงสร้างที่ใช้ บริเวณฉากหน้านั้นก็ยังคงรูปร่างเดิมอยู่ แต่หากกลุ่มพิกเซลนั้นมีรูปแบบไม่เหมือนกัน หรือเล็กกว่าส่วนโครงสร้างที่ใช้ก็จะถูกปิดโดยค่าพิกเซลของพื้นหลัง กระบวนการนี้ใช้มากในการกำจัดพิกเซลที่ไม่ต้องการ ตัวอย่างของการปิด แสดงดังรูปต่อไปนี้ ซึ่งใช้ส่วนโครงสร้างเป็นรูปสี่เหลี่ยมจัตุรัส  $3 \times 3$  ซึ่งโดยปกติแล้ว ส่วนโครงสร้าง  $3 \times 3$  จะใช้กันมากที่สุดในการทำมอร์โฟโลยี



รูปที่ 3.12 ตัวอย่างการเปิด (Opening)

### 3.6 การวาร์ป (Warping)

การวาร์ปเป็นการทรานส์ฟอร์มทางเรขาคณิต 2 มิติ เพื่อให้รูปอินพุตที่มีโครงสร้างบิดเบี้ยวหรือโค้งงอ (Distorted image) ให้เป็นรูปเอาต์พุตที่มีลักษณะดีขึ้นใกล้เคียงกับรูปแบบที่ต้องการมากที่สุด

เราจะแทนรูปต้นแบบ (source) ที่มีการผิดรูป (Distortion) ว่าภาพอินพุต และแทนรูปที่ต้องการให้มีลักษณะตามต้องการหรือภาพจุดหมาย (destination) ว่า ภาพเอาต์พุต

ความสัมพันธ์ระหว่างภาพต้นแบบและภาพที่ต้องการถูกแทนด้วยคู่อันดับ ซึ่งจะต้องมีการกำหนดตำแหน่งของคู่อันดับทั้งบนภาพต้นแบบให้ภาพจุดหมาย เพื่อให้การวาร์ปออกมาตามต้องการ ซึ่งคู่อันดับนี้จะเรียกว่าจุดควบคุม (Control point) ในการวาร์ปจะต้องมีการกำหนดสมการหรือรูปแบบที่แทนความสัมพันธ์ของภาพอินพุตและภาพเอาต์พุต ซึ่งสมการโพลิโนเมียล (Polynomial) จะถูกใช้มากที่สุด สมการโพลิโนเมียลลำดับที่ 1 (First order polynomial) ทำให้ภาพเอาต์พุตเกิดการยืดหรือถ่าง (Stretching) สมการโพลิโนเมียลลำดับที่ 2 (Second order polynomial) ใช้กับภาพที่มีการโค้งหรืองอเพียงครั้งเดียวเหมือนรูปพาราโบลา (Parabola) ส่วนสมการโพลิโนเมียลลำดับที่ 3 (Third order polynomial) ใช้กับภาพที่มีการโค้งหรืองอเป็นรูปตัว S (S-shape)

ในการตัดสินใจว่าจะใช้การวาร์ปแบบใด จะต้องพิจารณาก่อนว่ารูปอินพุตมีลักษณะเป็นอย่างไร แล้วจึงเลือกลำดับของการวาร์ป (หรือลำดับของสมการโพลิโนเมียล) เป็นอันดับที่สุดที่จะเลือกใช้ลำดับของการวาร์ปที่ต่ำที่สุดที่เหมาะสมกับงานของเรา เนื่องจากการวาร์ปในลำดับที่สูงขึ้นอาจทำให้ผลออกมาผิดไปจากที่คาดเอาไว้ และยังใช้เวลานานกว่ามากอีกด้วย ให้กำหนดจำนวนคู่อันดับของภาพต้นแบบและภาพจุดหมายมากจนพอที่จะทำให้ได้ภาพเอาต์พุตตามต้องการ

#### 3.6.1 ชนิดของการวาร์ป

การวาร์ปแบ่งออกเป็น 2 ชนิดใหญ่ ๆ คือ เชิงเส้น (Linear) และที่ไม่เป็นเชิงเส้น (Non-Linear)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1. แบบเชิงเส้น

เป็นการวาง (Map) จุดของฉากหน้าจากภาพต้นแบบลงบนภาพจุดหมาย ซึ่งในการวาร์ปแบบเชิงเส้นนี้ เฉพาะจุดของฉากหน้าเท่านั้นที่จะถูกวาร์ป ดังนั้นจึงทำให้มีความเร็วในการประมวลผลภาพสูง เนื่องจากส่วนใหญ่บริเวณฉากหน้าจะมีพื้นที่น้อยมากเมื่อเทียบกับพื้นหลัง การวาร์ปแบบเชิงเส้นมีคุณสมบัติของการรักษาเส้นให้ตรง ดังนั้นจึงใช้ในการวาร์ปภาพสี่เหลี่ยมจัตุรัสหรือสี่เหลี่ยมผืนผ้า ให้เป็นสี่เหลี่ยมขนานเป็ยกปุ่นหรือคางหมู โดยปรับมุมและขนาดตามที่ต้องการ

การวาร์ปแบบนี้สามารถแบ่งย่อยออกอีก ได้ เป็น 3 ชนิด คือ

### - Affine-1st order

โมเดลนี้ใช้สมการโพลีโนเมียลลำดับที่ 1 สามารถเรียกได้อีกชื่อว่าการวาร์ปแบบ Rubber Sheet จำนวนจุดควบคุมเพียงแต่ 2 คู่เท่านั้นเพื่อให้ได้อย่างน้อย 3 คู่

### - Helmert หรือ Orthogonal Warp

การวาร์ปแบบนี้ควรมีจำนวนจุดควบคุมเพียงแต่ 2 คู่เท่านั้นเพื่อให้ได้ผลที่ดีที่สุด ส่วนใหญ่จะใช้ในการสเกลขนาด (scale) หรือหมุน (rotate) เท่านั้น และจะไม่มีกรณีการบิดเบือนระหว่างการวาร์ปอีกด้วย ใช้ในงานทางด้านวิศวกรรมทั่วไป

### - Projective

จะกำหนดจุดควบคุมที่แน่นอน คือ จุดมุมทั้ง 4 ถึงจะถูกจัดอยู่ในประเภทเชิงเส้น แต่ความจริงแล้วจะใช้สมการ โพลีโนเมียลลำดับที่ 2 ในการคำนวณค่าของจุดควบคุมที่ 4 ซึ่งการคำนวณจุดนี้มีความสำคัญมากและใช้ฐานข้อมูลขนาดใหญ่

ทุกโมเดลข้างต้น ควรมีจำนวนจุดควบคุมไม่เกินค่าต่ำสุดตามที่กำหนดในตารางด้านล่าง เนื่องจากการใช้จุดควบคุมมากกว่าที่กำหนดจะทำให้เกิดความผิดพลาด (residuals หรือ errors) ขึ้นได้

โมเดล	จำนวนจุดควบคุม
Affine (1st order)	3
Helmert	2
Projective	4

ตารางที่ 3.1 แสดงจำนวนจุดควบคุมที่ต้องใช้กับการวาร์ปแบบเชิงเส้น

## 2. แบบไม่เชิงเส้น

คือการวาร์ปแบบ Affine ที่มีลำดับสูงกว่า 1 นั่นเอง ซึ่งจะมีตั้งแต่ลำดับที่ 2 ถึง 5 ใช้ในการปรับเส้นโค้งและไม่มีคุณสมบัติในการรักษาความตรงของเส้นตรง

เมื่อลำดับของการวาร์ปเพิ่มขึ้น จำนวนของเส้นโค้งที่จะวาร์ปได้ก็เพิ่มตามไปด้วยอัตราเดียวกัน การวาร์ปลำดับที่ 2 ใช้กับเส้นโค้ง 1 เส้น เช่นพาราโบลา การวาร์ปลำดับที่ 3 ใช้กับเส้นโค้ง 2 เส้นในรูปของตัว S ลำดับที่ 4 ใช้กับเส้นโค้ง 3 เส้น และลำดับที่ 5 ใช้กับเส้นโค้ง 4 เส้น

การวาร์ปแบบนี้ ทุกพิกเซลจะถูกคำนวณโดยใช้สมการโพลีโนเมียล ดังนั้นจึงใช้เวลาค่อนข้างนาน

การกำหนดจำนวนจุดควบคุมค่าที่สุดจะเป็นดังตารางที่ 3.2 ซึ่งการวาร์ปแบบนี้จะต่างกับแบบเชิงเส้น ตรงที่จุดควบคุมยิ่งมากจะยิ่งส่งผลให้ผลลัพธ์ที่ได้ตรงตามต้องการมากขึ้น

โมเดล	จำนวนจุดควบคุม
Second order	6
Third order	10
Forth order	15
Fifth order	21

ตารางที่ 3.2 แสดงจำนวนจุดควบคุมที่ต้องใช้กับการวาร์ปแบบไม่เชิงเส้น

### 3.6.2 การวาร์ปโดยใช้โพลีโนเมียลลำดับที่ 2 (Second Order Polynomial Warping)

การวาร์ปนี้เป็นกรวาร์ปแบบไม่เชิงเส้น ซึ่งในการแปลงจุดพิกเซลจากภาพต้นแบบไปยังภาพจุดหมายนั้นจะต้องใช้การแม็พจุดแบบย้อนหลัง (Reverse Address Mapping) คือ การกำหนดจุดพิกัดพิกเซลของภาพจุดหมายที่ต้องการเอาไว้  $(x,y)$  แล้วใช้สมการโพลีโนเมียลคำนวณหาตำแหน่งพิกัดจากภาพต้นแบบที่ควรใช้  $(u,v)$  เพื่อคำนวณหาค่าเมทริกซ์ที่ใช้ในการทรานส์ฟอร์ม (Transform) หรือการแปลงค่าพิกัด

การแม็พค่าพิกัด (Address mapping) ทำได้โดยใช้สมการโพลีโนเมียลลำดับที่ 2 ซึ่งเป็นดังสมการ (3.8) และ (3.9)

$$u = a_0 + a_1 x + a_2 y + a_3 x^2 + a_4 xy + a_5 y^2 \quad (3.8)$$

$$v = b_0 + b_1 x + b_2 y + b_3 x^2 + b_4 xy + b_5 y^2 \quad (3.9)$$

หรือสามารถเขียนในรูปของเวกเตอร์ได้เป็น

$$u = [a_0 \ a_1 \ a_2 \ a_3 \ a_4 \ a_5] \begin{bmatrix} 1 \\ x \\ y \\ x^2 \\ xy \\ y^2 \end{bmatrix} \quad (3.10)$$

$$v = [b_0 \ b_1 \ b_2 \ b_3 \ b_4 \ b_5] \begin{bmatrix} 1 \\ x \\ y \\ x^2 \\ xy \\ y^2 \end{bmatrix} \quad (3.11)$$

หากกำหนดให้  $\mathbf{a} = [a_0 \ a_1 \ a_2 \ a_3 \ a_4 \ a_5]$  และ  $\mathbf{b} = [b_0 \ b_1 \ b_2 \ b_3 \ b_4 \ b_5]$  โดยที่  $\mathbf{a}$  และ  $\mathbf{b}$  เป็นเวกเตอร์ที่เรียกว่า สัมประสิทธิ์การให้น้ำหนัก (Weighting Coefficient) ในการที่จะหรรณส์ฟอร์มพิกัด จะต้องหาค่าในเวกเตอร์เหล่านี้ด้วย ซึ่งตามปกติแล้ว การเลือกค่าสัมประสิทธิ์การให้น้ำหนักจะเลือกตามหลัก Mean-square error ระหว่าง พิกัดในภาพต้นแบบที่ใช้ควบคุมการจาร์ป ( $u_m, v_m$ ) และพิกัดของภาพต้นแบบ ( $u, v$ ) สำหรับเซตของ  $(x_m, y_m)$  โดยที่  $1 \leq m \leq M$  ความผิดพลาดจะเกิดขึ้นน้อยที่สุด ถ้า

$$\mathbf{a} = ((\mathbf{A}'\mathbf{A})^{-1}\mathbf{A}')\mathbf{u} \quad (3.12)$$

$$\mathbf{b} = ((\mathbf{A}'\mathbf{A})^{-1}\mathbf{A}')\mathbf{v} \quad (3.13)$$

โดย  $\mathbf{u} = [u_1, u_2, \dots, u_M]$  และ  $\mathbf{v} = [v_1, v_2, \dots, v_M]$  เป็นจุดควบคุมของภาพต้นแบบ และ

$$\mathbf{A} = \begin{bmatrix} 1 & u_1 & v_1 & u_1^2 & u_1 v_1 & v_1^2 \\ 1 & u_2 & v_2 & u_2^2 & u_2 v_2 & v_2^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & u_M & v_M & u_M^2 & u_M v_M & v_M^2 \end{bmatrix} \quad (3.14)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากที่ได้กล่าวไว้แล้ว การวาร์ปลำดับที่ 2 นี้จะต้องใช้จุดควบคุมอย่างน้อย 6 จุด จากทั้งภาพต้นแบบและภาพจุดหมาย

เมื่อได้ทำการวาร์ปโดยคำนวณค่าพิกัดตามสมการข้างต้นเรียบร้อยแล้ว ยังต้องใช้การ Interpolation หรือการประมาณค่าพิกัดที่คำนวณได้ซึ่งไม่เป็นจำนวนเต็มให้เป็นค่าจำนวนเต็มเพื่อที่จะแมปได้อย่างถูกต้อง

ตัวอย่างของการวาร์ปสามารถดูได้จากรูปที่ 3.12



รูปที่ 3.13 ตัวอย่างการวาร์ปโดยใช้โพลีโนเมียลลำดับที่ 2

## บทที่ 4

### การออกแบบโปรแกรมที่ใช้วิเคราะห์ภาพรหัสแถบ 2 มิติ

ในการออกแบบโปรแกรมที่ใช้วิเคราะห์และประมวลผลภาพรหัสแถบ 2 มิติ ชนิด MicroPDF417 นั้น ได้เลือกโปรแกรม MATLAB เวอร์ชัน 6.5 มาใช้ โดยโปรแกรมที่ออกแบบนั้น จะรับอินพุตเป็นภาพรหัสแถบที่ถ่ายในมุมต่าง ๆ กันจึงทำให้เกิดการผิดรูป (Distortion) ภาพเหล่านี้ ถูกถ่ายโดยกล้องดิจิทัลในเกรย์สเกล โปรแกรมการประมวลผลภาพที่น่าเสนอ สามารถแบ่งย่อยได้ เป็น 6 กระบวนการ ดังนี้

1. การหาค่าเทรชโฮลที่ดีที่สุดแล้วแปลงให้เป็นภาพไบนารี (Thresholding)
2. การกำจัดสิ่งรบกวน (Noise Cancellation) โดยใช้เทคนิคการตรวจจับขอบภาพ (Edge detection)
3. การหามุมที่รหัสแถบวางตัวใน 2 มิติ แล้วทำการหมุน (Rotatation) ให้รหัสแถบวางตัวในแนวนอน
4. ทำการหาพิคคองมุมทั้ง 4 ของรหัสแถบ (Corner Detection)
5. ทำการวาร์ป โดยใช้วิธีโพลีโนเมียลลำดับที่ 2 (Second-order Polynomial Warping) เพื่อให้รหัสแถบอยู่ในรูปสี่เหลี่ยมผืนผ้า
6. อ่านค่ารหัสแถบออกมาเป็นค่าในแต่ละ ไกด์เวอร์ด (Decoding) ซึ่งมีค่าเป็น 0 ถึง 1 ลำดับการประมวลผลหลักของโปรแกรม สามารถดูได้จากบล็อกไดอะแกรมรวมแสดงการประมวลผลภาพรหัสแถบ (รูปที่ 1.1)

กระบวนการย่อยทั้ง 6 สามารถอธิบายโดยละเอียด ดังต่อไปนี้

#### 4.1 การเลือกค่าเทรชโฮลและแปลงภาพเป็นไบนารี (Thresholding)

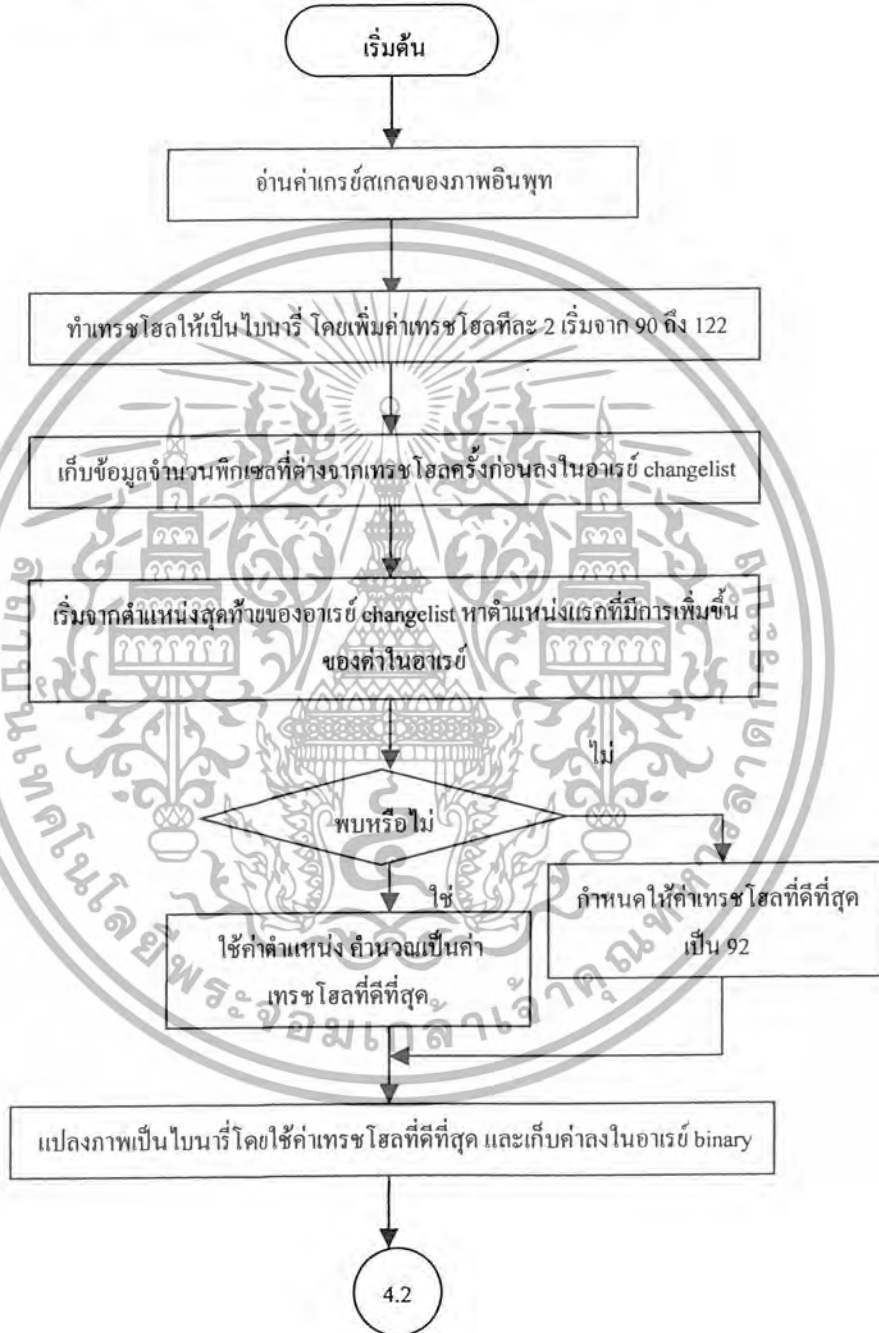
##### จุดประสงค์

เพื่อหาค่าเทรชโฮลที่ดีที่สุด (Optimal Threshold) และนำไปใช้แปลงภาพอินพุตจากเกรย์สเกลให้เป็นภาพไบนารี โดยที่ต้องให้รายละเอียดของรหัสแถบคงเหลืออยู่บนภาพให้มากที่สุดเท่าที่จะเป็นไปได้

##### หลักการทำงาน

โปรแกรมจะเลือกค่าเทรชโฮลที่ดีที่สุดของแต่ละภาพอินพุตโดยอัตโนมัติ โดยจากการทดลองแล้วพบว่า ค่าเทรชโฮลที่ให้ผลที่ดีที่สุดจะอยู่ในช่วง 90 ถึง 120 (ค่าความเข้มแสงในเกรย์สเกล) เมื่อรับภาพอินพุตเข้า เครื่องคอมพิวเตอร์จะทำการอ่านให้รูปในรูปของเมทริกซ์ แล้วโปรแกรมจะสั่งให้เครื่องทำการแปลงภาพเป็นไบนารีโดยใช้ค่าเทรชโฮลตั้งแต่ 90 - 120 โดยเพิ่มค่าเทรชโฮลที่

ละ 2 แล้วทำการเก็บค่าเมทริกซ์หลังจากการแปลงทุกค่า ทำการเปรียบเทียบว่าระหว่างการเพิ่มค่าเทรชโฮลแต่ละครั้งนั้น มีพิกเซลที่เปลี่ยนค่าไปทั้งหมดกี่พิกเซล ตามหลักของการเลือกค่าเทรชโฮลที่ดีที่สุดที่จุดนั้น (Optimal Thresholding) จะต้องเป็นค่าค่าที่สุทธระหว่างค่าขอบของการกระจายความเข้มพื้นหลังและฉากหน้าในฮิสโตแกรม ดังนั้นจุดที่มีการเปลี่ยนค่าพิกเซลน้อยที่สุดจะถูกเลือกให้เป็นค่าเทรชโฮลที่ดีที่สุด แล้วใช้ในการเปลี่ยนจากภาพเกรย์สเกลให้เป็น ไบนารี



รูปที่ 4.1 บล็อกไดอะแกรมแสดงการทำเทรชโฮล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

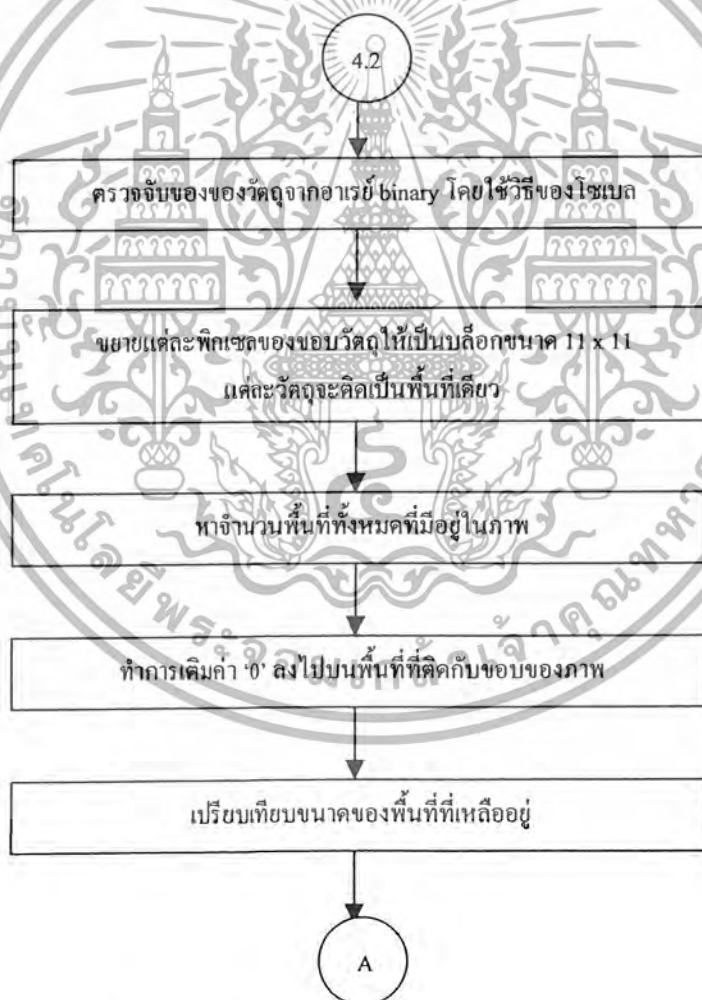
#### 4.2 การกำจัดสิ่งรบกวน (Noise Cancellation)

##### จุดประสงค์

เพื่อใช้กำจัดบริเวณที่ไม่ใช่รหัสแถบออกจากภาพให้หมด

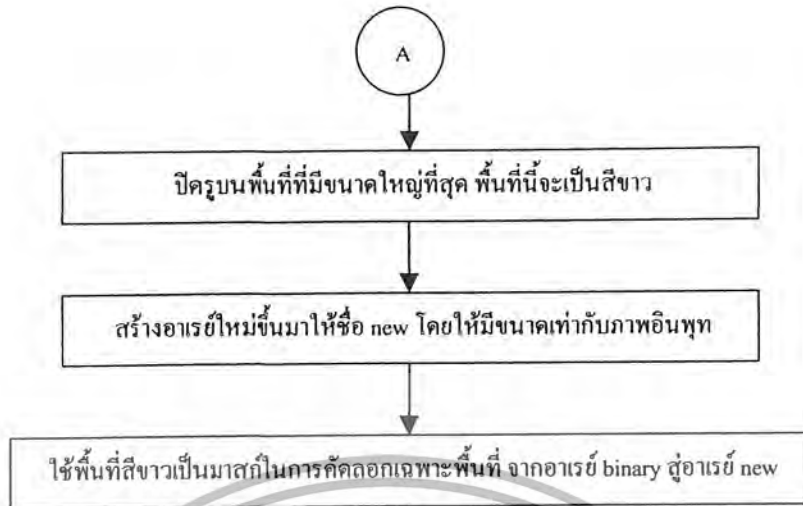
##### หลักการทํางาน

- ทำการหาขอบของวัตถุในภาพโดยใช้วิธีของ โซเบล (Sobel Edge Detection)
- ขยายพิกเซลขอบให้เป็นบล็อกขนาด  $11 \times 11$  เพื่อให้ขอบของสิ่งเดียวกันติดกันเป็นพื้นที่เดียว (region)
- กำจัดพื้นที่ที่ติดกับเส้นขอบของภาพ (border) โดยการเติม (fill) ค่า '0' ให้กับพื้นที่นั้น
- เปรียบเทียบพื้นที่ของวัตถุที่เหลือในภาพ
- เนื่องจากวัตถุที่ใหญ่ที่สุดจะเป็นรหัสแถบ ดังนั้นจะกำจัดวัตถุที่เหลือออกให้หมดจากภาพได้โดยการสร้างภาพใหม่ขึ้นมา ให้มีขนาดเท่าภาพอินพุต แต่จะลอกเฉพาะบริเวณพื้นที่ของรหัสแถบเท่านั้น ขั้นตอนนี้จะสามารถกำจัด Spray noise ได้



รูปที่ 4.2 บล็อกไดอะแกรมการกำจัดสิ่งรบกวนออกจากภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 บล็อกไดอะแกรมการกำจัดสิ่งรบกวนออกจากภาพ (ต่อ)

#### 4.3 การหมุนภาพให้รหัสแถบวางตัวในแนวนอน (Rotation)

##### จุดประสงค์

เพื่อให้การตรวจหามุมของรหัสแถบในขั้นตอนต่อไปทำได้ง่ายขึ้น จึงต้องทำให้บริเวณของรหัสแถบวางตัวในแนวนอน โดยที่ไม่จำเป็นต้องทำมุม 0 องศากับแกนอน เพียงแต่ต้องให้การวางตัวของรหัสแถบจากกระบวนการนี้ ไม่เป็นรูปแบบที่มุมบนไปอยู่ส่วนครึ่งล่างของภาพ หรือกลับกัน ซึ่งจะทำให้การตรวจหามุมในขั้นตอนถัดไปทำได้ยาก

##### หลักการทำงาน

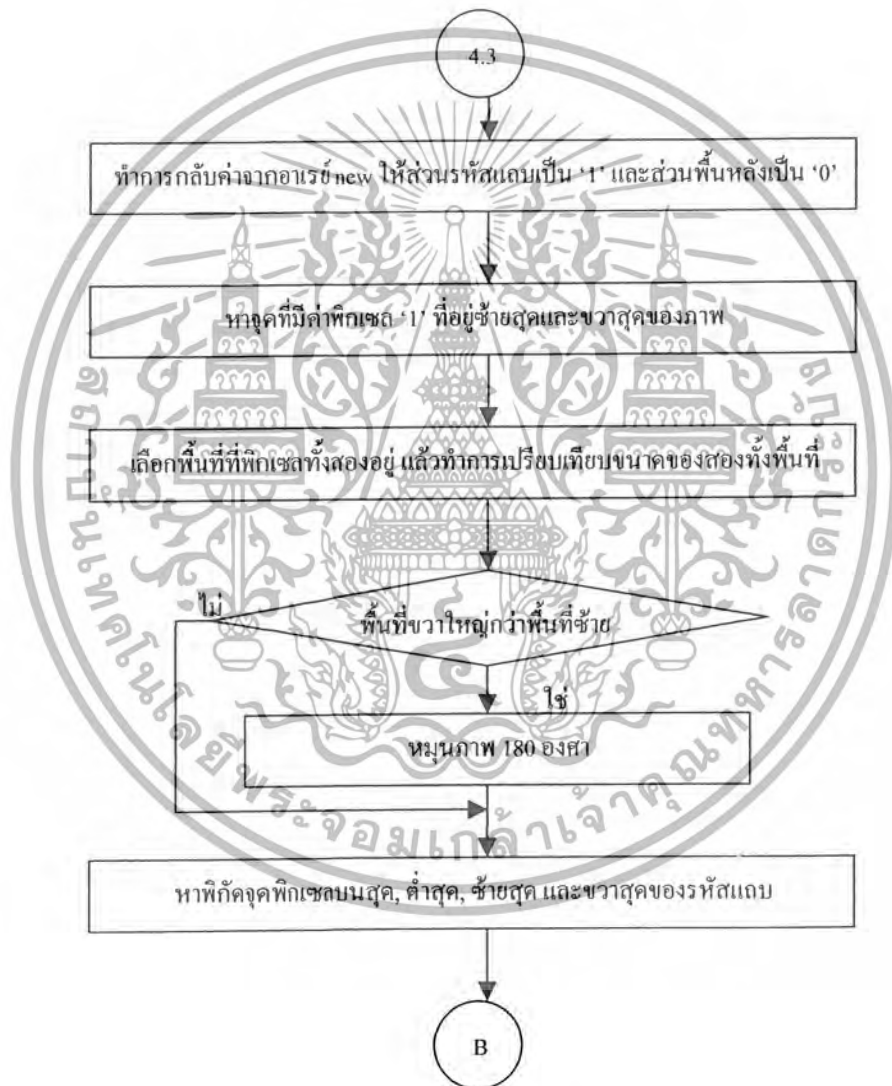
- จากรูปแบบของรหัสแถบ MicroPDF417 ตามตัวอย่างในรูปที่ 2.4 นั้น ด้านซ้ายสุดของรหัสแถบจะเป็นคอลัมน์ RAP (Row Addressing Pattern Column) ซึ่งจะมีความกว้างของแถวมากกว่าแถวหยุด (Stop Row) ที่อยู่ขวาสุดทุกรหัสแถบ ดังนั้นการจะทำให้รหัสแถบวางตัวในแนวนอนก็เพียงเปรียบเทียบขนาดพื้นที่ของแถวซ้ายและแถวขวานั้นเอง หากรหัสแถบบนภาพอินพุตถูกถ่ายแบบกลับหัว คือคอลัมน์ RAP ไปอยู่ด้านขวาของภาพ ให้ทำการหมุนภาพ 180 องศา

- หากค่าจุดบนสุด, ต่ำสุด, ซ้ายสุด และขวาสุดของพื้นที่รหัสแถบ ซึ่งจะทำให้รู้ว่ารหัสแถบมีลักษณะบิดเบือนหรือไม่อย่างไร และยังสำคัญต่อการหาค่ามุมที่จะใช้หมุนภาพเพื่อให้อยู่ในแนวนอนอีกด้วย

- จากค่าพิกัดของทั้ง 4 จุดที่กล่าวข้างต้น สามารถแบ่งกลุ่มของรูปแบบรหัสแถบได้เป็น 2 กลุ่ม คือ

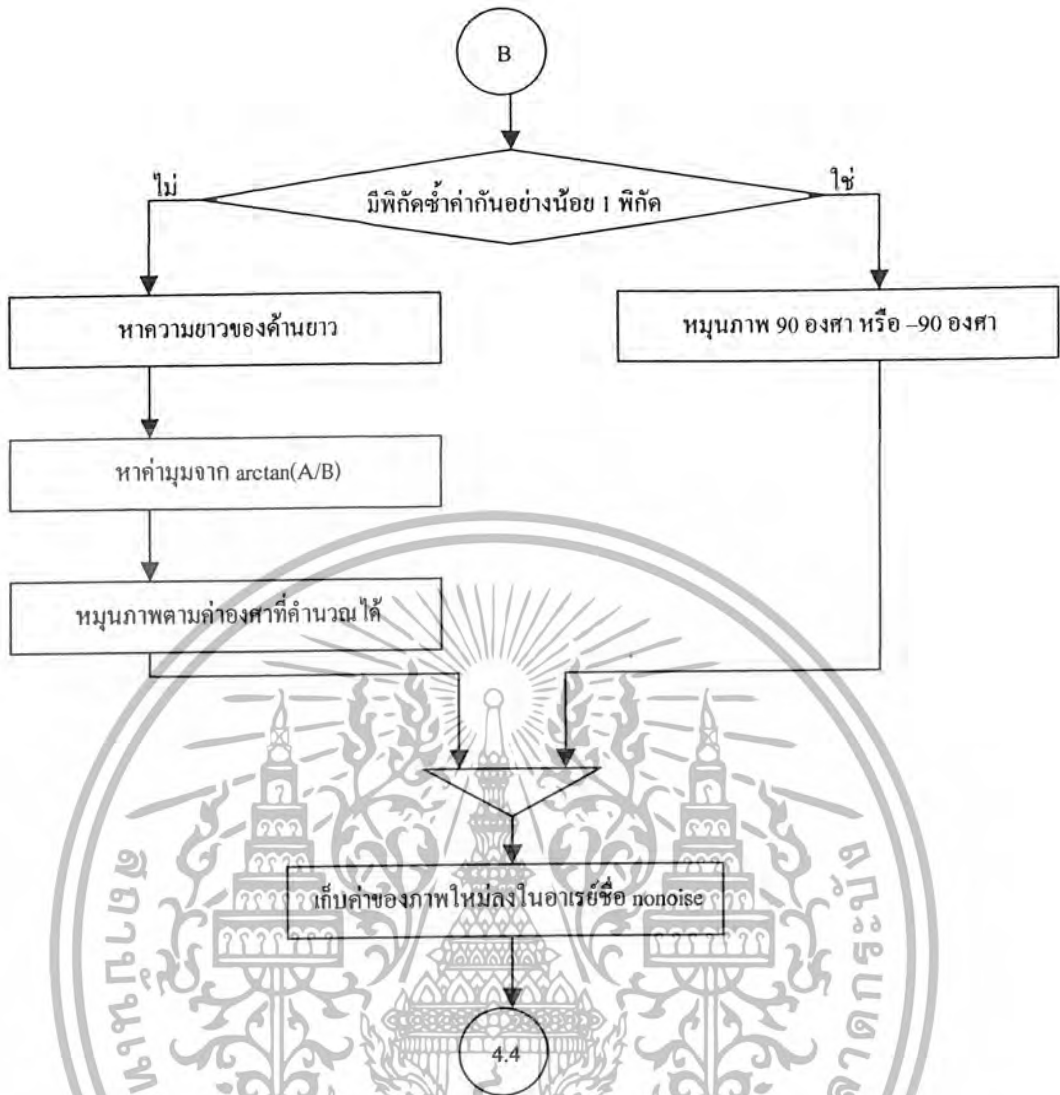
กลุ่มที่ 1 จะมีย่าน้อยพิกัดหนึ่งที่ซ้ำกับพิกัดของอีกจุด ซึ่งแสดงว่ารหัสแถบมีการบิดรูป เป็นหรือคล้ายแบบสี่เหลี่ยมขนมเปียกปูน ซึ่งจะไม่สามารถกำหนดจุดที่จะสร้างสมการตรีโกณเพื่อหามุมได้ จึงต้องทำการหมุน 90 องศา หรือ -90 องศา (ทวนเข็มนาฬิกา)

กลุ่มที่ 2 จะไม่มีพิกัดใดเลยที่เท่ากัน ดังนั้นการกำหนดจะที่จะสร้างสมการตรีโกณเพื่อหา มุมนั้นง่ายกว่า แล้วจึงหมุนภาพตามมุมที่คำนวณได้



รูปที่ 4.3 บล็อกไดอะแกรมการหมุนภาพให้รหัสแถบวางตัวในแนวนอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 บล็อกโคแอดอร์เนกชันการหมุนภาพให้รหัสแถบวางตัวในแนวนอน (ต่อ)

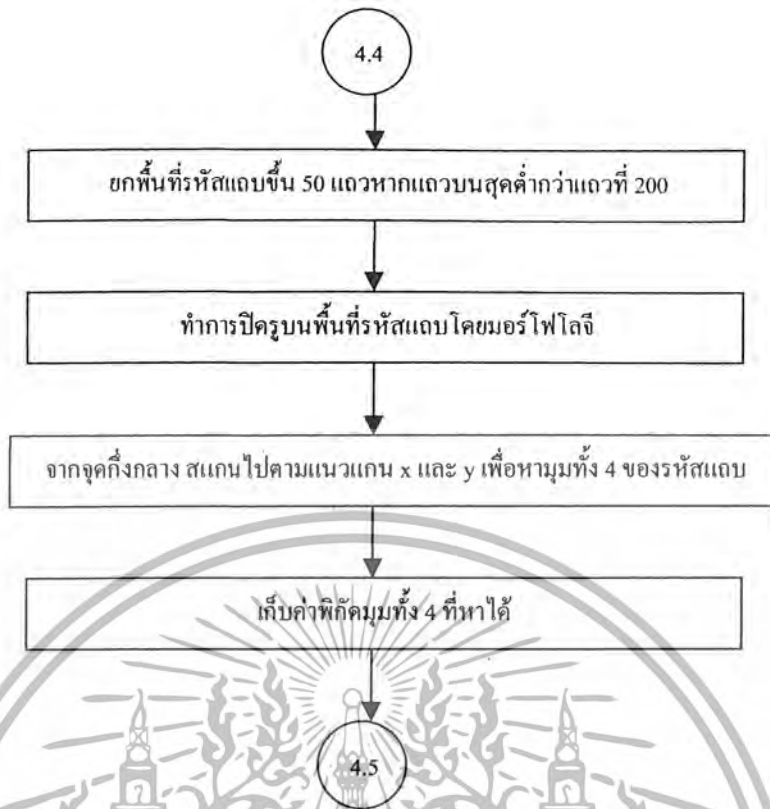
#### 4.4 การหามุมของรหัสแถบ (Corner Detection)

##### จุดประสงค์

เพื่อใช้เป็นจุดควบคุมในการวาร์ปในขั้นตอนต่อไป

##### หลักการทํางาน

จากจุดกึ่งกลางของพื้นที่รหัสแถบที่ต่อเนื่องกันเป็นพื้นที่เดียวโดยการทำการปิด (Closing) ให้สแกนตามแนวแกน x ไปทางซ้ายจนสุดพื้นที่ แล้วสแกนตามแนวแกน y ไปทางด้านบนและล่างเพื่อหาจุดมุมบนซ้าย และล่างซ้ายตามลำดับ ในทำนองเดียวกัน ทำการสแกนตามแนวแกน x ไปทางขวาจนสุด แล้วสแกนตามแกน y ไปด้านบนและล่างเพื่อหาจุดมุมบนขวาและล่างขวา



รูปที่ 4.4 บล็อกโคอะแกรมการหามุมของรหัสแถบ

#### 4.5 การทำพอลิโนเมียลสแควร์ปล้ำดับที่ 2 (Second-order Polynomail Warping)

##### จุดประสงค์

เพื่อทำการแปลงภาพรหัสแถบที่บิดเบือนหรือผิดรูป (Distorted Image) จากการถ่ายภาพให้เป็นในรูปสี่เหลี่ยมผืนผ้าซึ่งเป็นลักษณะเดิมของรหัสแถบ และเป็นรูปแบบที่เหมาะสมต่อการถอดรหัสมากที่สุด

##### หลักการทำงาน

- กำหนดจุดควบคุมในภาพอื่นพุทของกระบวนการและจุดควบคุมในภาพเอ้าท์พุท โดยในภาพเอ้าท์พุทให้กำหนดเป็นรูปสี่เหลี่ยมผืนผ้า จุดควบคุมนี้จะมี 8 จุดในแต่ละภาพ ได้แก่จุดมุมทั้ง 4 และอีก 4 จุดที่กึ่งกลางแต่ละด้าน
- ทำการคำนวณหาเมทริกซ์ของเอ้าท์พุทตามทฤษฎีการวาร์ป
- กำจัด Noise เล็ก ๆ ที่อยู่ตามขอบของรหัสแถบ โดยใช้ส่วนโครงสร้างเป็นรูปสี่เหลี่ยมจัตุรัสขนาด  $3 \times 3$  ภาพสุดท้ายจะเรียบขึ้น



รูปที่ 4.5 บล็อกไดอะแกรมการทำโพลีโนเมียลวาร์ปลำดับที่ 2

#### 4.6 การถอดรหัส (Decoding)

##### จุดประสงค์

เพื่อทำการอ่านค่าในแต่ละรหัส (Codeword) ของรหัสแถบ แล้วเก็บค่าไว้เพื่อเปรียบเทียบประสิทธิภาพการประมวลผลภาพทั้งหมด โดยการคำนวณจำนวนพิกเซลที่แตกต่าง

##### หลักการทํางาน

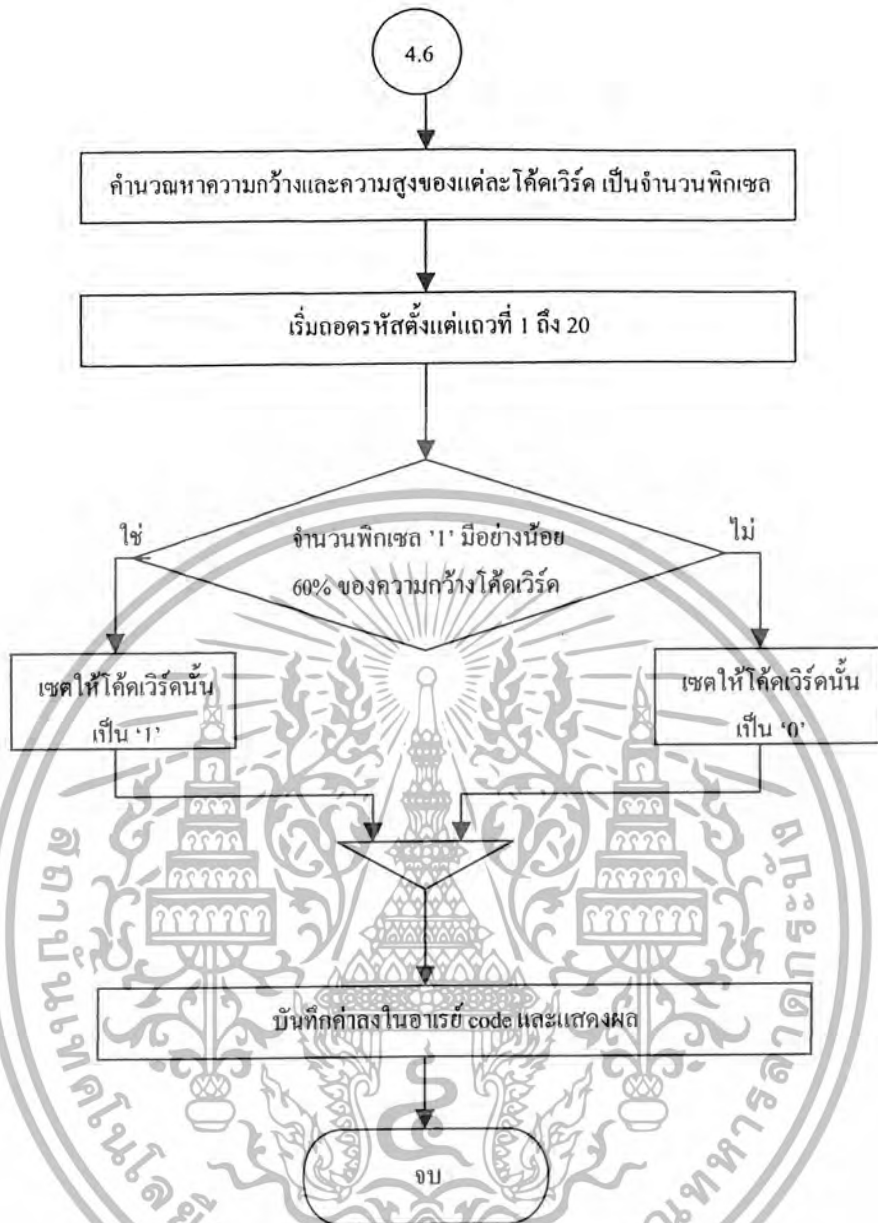
- ให้นำค่าความสูงและความกว้างของแต่ละโค้ดเวิร์ด หลังจากที่เราทราบค่าความสูงและความกว้างของภาพรหัสแถบที่ประมวลผลแล้ว ให้นำค่ามิติของภาพรหัสแถบหารด้วยจำนวนแถว คือ 20 แถว และจำนวนของโค้ดเวิร์ดในแต่ละแถว คือ 55 โค้ดเวิร์ด สำหรับชนิดของ MicroPDF417 ที่นำมาใช้ทดลอง

- ทำการบันทึกค่ารหัสในแถวที่ 1 - 20 โดยให้สแกนในแถวพิกเซลที่ 2 จากพิกเซลเริ่มต้นแถวใหม่นั้น เพื่อป้องกันความผิดพลาดในหลาย ๆ กรณี การอ่านค่ารหัสแถบให้อ่านจากซ้ายไปขวา และจากบนลงล่าง

- หากจำนวนพิกเซล '1' มีอย่างน้อย 60% ของความกว้างโค้ดเวิร์ด (เนื่องจากบริเวณรหัสแถบยังเป็นสีขาวอยู่) ให้โค้ดเวิร์ดนั้นเป็น '1' หากน้อยกว่าให้เซตเป็น '0'

- ทำการกลับค่าโค้ดเวิร์ดที่อ่านได้ โดยผลลัพธ์สุดท้ายนี้ ค่า '0' จะแทนส่วนสีดำของรหัสแถบ '1' จะแทนส่วนพื้นสีขาวของรหัสแถบ

สามารถดูขั้นตอนการถอดรหัสได้จากบล็อกโคอะแกรมแสดงการทำงานในหน้าถัดไป



รูปที่ 4.6 บล็อกโค้ดอะแกรมการถอดรหัสของรหัสแถบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### ผลการประมวลภาพและผลการทดลอง

ในการทดสอบประสิทธิภาพของโปรแกรม จะใช้ภาพต้นแบบทั้งหมด 6 ภาพ ที่ถูกถ่ายมาจากรหัสแถบชนิด MicroPDF417 จำนวน 2 แบบรหัส ๆ ละ 3 ภาพ ผลการประมวลภาพในแต่ละขั้นตอนจะแสดงดังต่อไปนี้

#### 5.1 การเลือกค่าเทรชโวลและแปลงภาพเป็นไบนารี (Thresholding)

ในการเลือกค่าเทรชโวลให้อยู่ในช่วง 90 ถึง 120 ให้ผลเป็นที่น่าพอใจสำหรับภาพส่วนใหญ่ที่นำมาทดลอง แต่ยังมีภาพหนึ่ง คือภาพต้นแบบที่ 1 ที่ไม่สามารถหาจุดค่าสุดของการเปลี่ยนแปลงจำนวนพิกเซลได้ ซึ่งมีผลทำให้ค่าเทรชโวลที่เลือกค่าไป ดังแสดงในรูปต่อไปนี้



(a) ภาพต้นแบบที่ 2



(b) ภาพที่ได้จากการเทรชโวล

รูปที่ 5.1 ตัวอย่างการทำเทรชโวลที่ได้ผลดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(a) ภาพต้นแบบที่ 1



(b) ภาพที่ได้จากการเทรซโฮล

รูปที่ 5.2 ตัวอย่างการทำเทรซโฮลที่ได้ผลไม่ค่อยดี

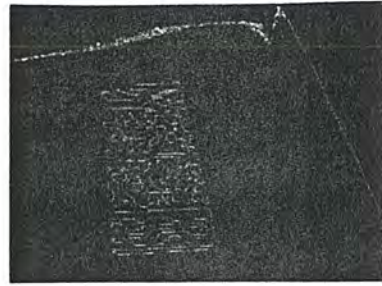
จากรูปที่ 5.2 (b) สังเกตว่าบริเวณที่ถูกครีชีอยู่นั้นมีขนาดของโคคเวรีคของรหัสแถบเล็กกลงจากภาพต้นแบบในรูป (a)

## 5.2 การกำจัดสิ่งรบกวน (Noise Cancellation)

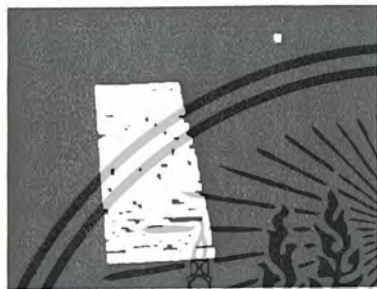
สิ่งรบกวน (Noise) ในทุกภาพถูกกำจัดได้หมดอย่างมีประสิทธิภาพ ดังรูปที่ 5.3 จะแสดงขั้นตอนในการกำจัดสิ่งรบกวน ตั้งแต่การตรวจจับขอบของภาพ, การขยายให้เป็นบล็อก, และภาพที่ผ่านกระบวนการกำจัดสิ่งรบกวนแล้ว



(a) ภาพเทรซโฮลจากต้นแบบที่ 5



(b) ภาพแสดงขอบวัตถุ



(c) ภาพแสดงการทำบล็อก หลังจากลบพื้นที่ที่ติดกับขอบแล้ว



(d) ภาพหลังจากการกำจัด Noise

รูปที่ 5.3 แสดงแต่ละขั้นตอนและผลจากการกำจัดถึงบริเวณออกจากภาพ

### 5.3 การหมุนภาพให้รหัสแถบวางตัวในแนวนอน (Rotation)

ในขั้นนี้ ไม่มีการกำหนดค่ามุมที่แน่นอนของรหัสแถบ เพียงแต่ต้องการให้รหัสแถบวางตัวในแนวนอนเท่านั้น ซึ่งก็ได้ผลดีในทุก ๆ ภาพ



(a) ภาพต้นแบบที่ 5 ก่อนการหมุน

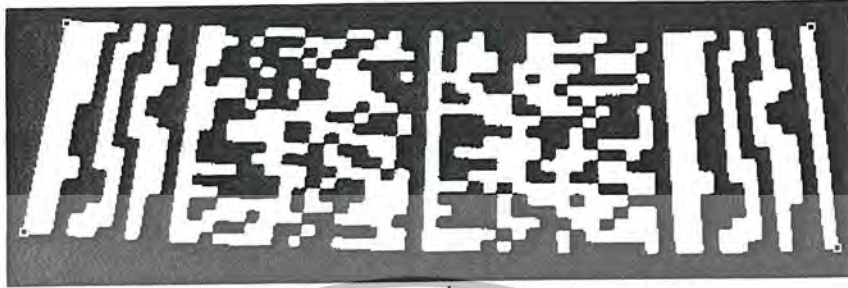


(b) ภาพหลังการหมุน

รูปที่ 5.4 ตัวอย่างภาพที่ผ่านการหมุน (Rotation)

#### 5.4 การหามุมของรหัสแถบ (Corner Detection)

การตรวจหามุมของรหัสแถบจากโปรแกรมนี้ให้ผลดี สามารถตรวจจับได้ในทุกภาพต้นแบบ แสดงได้ดังรูปที่ 5.5



รูปที่ 5.5 แสดงตำแหน่งมุมที่ตรวจจับได้ (ขยายใหญ่)

แต่จากการทำการทดลองที่ผ่านมา การตรวจหามุมจะอ่อนไหวมากต่อการเปลี่ยนแปลงค่าพารามิเตอร์ เช่น การเปลี่ยนค่าเทรชโฮลเพียงเล็กน้อย มีผลทำให้ตรวจหาค่ามุมผิดไปอย่างมาก จึงต้องมีการควบคุมค่าพารามิเตอร์ต่างๆ ให้ดี

#### 5.5 การทำโพลีโนเมียลวอร์ปลำดับที่ 2 (Second-order Polynomial Warping)

ผลของการวอร์ปสามารถแบ่งวิเคราะห์ที่ได้เป็น 2 กลุ่ม

##### 1. กลุ่มที่ภาพต้นแบบมีลักษณะเป็นเพอร์สเปคทีฟ (Perspective)

Perspective เป็นมุมมองจากการถ่ายภาพ โดยส่วนที่อยู่ใกล้กล้องถ่ายรูปมากกว่าจะมีขนาดใหญ่ ส่วนที่อยู่ไกลออกไปจะมีขนาดเล็กลง ซึ่งทำให้ขนาดความกว้างของแต่ละส่วนของรหัสแถบไม่เท่ากัน ถือว่าเป็นการผิดรูป (distortion) อย่างหนึ่ง ซึ่งการวอร์ปไม่สามารถแก้ไขตรงจุดนี้ได้เลย ทำให้ผลออกมาไม่ดี ดูตัวอย่าง ได้จากรูปที่ 5.6 ซึ่งเป็นผลที่ได้จากการวอร์ปภาพต้นแบบที่ 5

2. กลุ่มที่ภาพต้นแบบมีลักษณะโค้งงอหรือไม่ก็ได้ (distorted) แต่จะต้องไม่มีลักษณะเป็นเพอร์สเปคทีฟ ผลของการวอร์ปเป็นที่น่าพอใจ ดังตัวอย่างในรูปที่ 5.7 จะแสดงแต่ละขั้นตอนของการวอร์ปภาพต้นแบบที่ 1 ซึ่งจะเห็นว่าขอบของรูป (a) จะโค้ง ส่วนในรูป (c) จะเป็นเส้นตรง



รูปที่ 5.6 ตัวอย่างของภาพที่ได้จากการวาร์ปภาพต้นแบบในมุมมอง Perspective ซึ่งได้ผลไม่ดี



รูปที่ 5.7 แสดงขั้นตอนในการวาร์ปที่ได้ผลดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 5.6 การถอดรหัส (Decoding)

### การวัดผลการทดลองและผลการทดลอง

เมื่อภาพผ่านการวาร์ปแล้ว จะถูกนำมาเข้าโปรแกรมถอดรหัสเพื่ออ่านค่าในแต่ละโค้ดเวิร์ด (Codeword) ออกมา แล้วนำไปทำการเปรียบเทียบกับค่าโค้ดเวิร์ดจริงที่อ่านได้จากรูปต้นแบบ จากการทดลองทั้งหมด 6 ภาพ โดยที่ภาพต้นแบบที่ 1, 2, 6 ถูกถ่ายมาจากรหัสแถบเดียวกัน ส่วนภาพต้นแบบที่ 3, 4, 5 ถูกถ่ายมาจากรหัสแถบอีกอันหนึ่ง ดังรูปที่ 5.8 แสดงภาพต้นแบบ และภาพที่ได้จากการประมวลผลในขั้นตอนสุดท้ายซึ่งจะใช้ในการอ่านค่าโค้ดเวิร์ด และจะได้ผลการทดลองเป็นดังตารางที่ 5.1

ภาพต้นแบบที่	จำนวนโค้ดเวิร์ดที่อ่านผิด (จาก 1100)	เปอร์เซ็นต์ความผิดพลาด (error)	เปอร์เซ็นต์ความแม่นยำ (accuracy)
1	7	0.64	99.36
2	10	0.91	99.09
3	171	15.55	84.45
4	143	13.00	87.00
5	591	53.73	46.27
6	558	50.73	49.27
ค่าเฉลี่ย	246.67	22.43	77.57

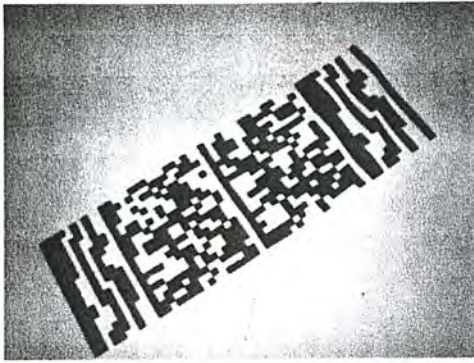
ตารางที่ 5.1 แสดงการรหัสที่อ่านได้จากภาพที่ผ่านการประมวลผลแล้วเทียบกับรหัสจริง



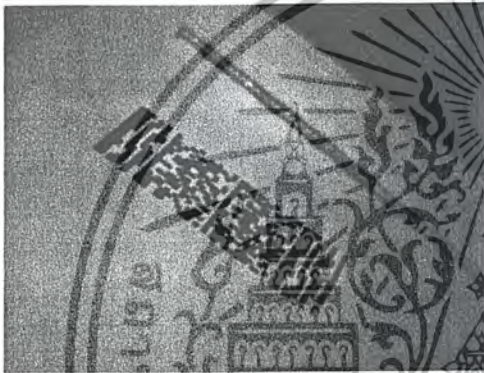
(a) ภาพต้นแบบที่ 1 (ซ้าย) และภาพที่ได้จากการประมวลผลในขั้นตอนสุดท้าย (ขวา)

รูปที่ 5.8 แสดงภาพต้นแบบที่ใช้ทดลองและภาพที่ได้จากการประมวลผลในขั้นตอนสุดท้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(b) ภาพต้นแบบที่ 2 (ซ้าย) และภาพที่ได้จากการประมวลผลในขั้นตอนสุดท้าย (ขวา)



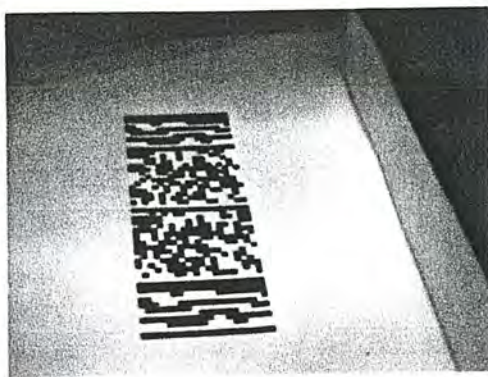
(c) ภาพต้นแบบที่ 3 (ซ้าย) และภาพที่ได้จากการประมวลผลในขั้นตอนสุดท้าย (ขวา)



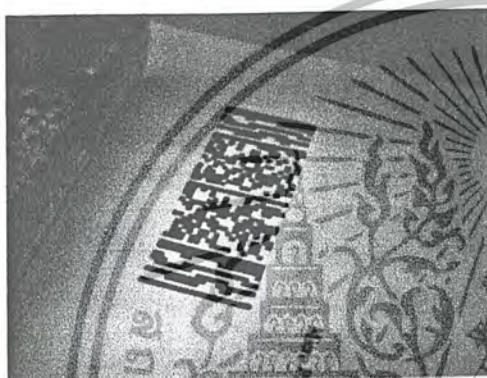
(d) ภาพต้นแบบที่ 4 (ซ้าย) และภาพที่ได้จากการประมวลผลในขั้นตอนสุดท้าย (ขวา)

รูปที่ 5.8 แสดงภาพต้นแบบที่ใช้ทดลองและภาพที่ได้จากการประมวลผลในขั้นตอนสุดท้าย (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(c) ภาพต้นแบบที่ 5 (ซ้าย) และภาพที่ได้จากการประมวลผลในขั้นตอนสุดท้าย (ขวา)



(f) ภาพต้นแบบที่ 6 (ซ้าย) และภาพที่ได้จากการประมวลผลในขั้นตอนสุดท้าย (ขวา)

รูปที่ 5.8 แสดงภาพต้นแบบที่ใช้ทดลองและภาพที่ได้จากการประมวลผลในขั้นตอนสุดท้าย (ต่อ)

## บทที่ 6

### สรุปและวิจารณ์

#### 6.1 สรุปและวิเคราะห์ผลการทดลอง

จากการนำภาพต้นแบบ 6 ภาพมาประมวลผลโดยใช้โปรแกรมการวิเคราะห์ภาพรหัสแถบ 2 มิติที่ออกแบบขึ้น เพื่อที่จะทดสอบดูว่าโปรแกรมเหมาะสมและมีประสิทธิภาพหรือไม่อย่างไร ทำให้พบว่า

- โปรแกรมในขั้นตอนของทำเทรซโฮลกับภาพส่วนใหญ่สามารถทำงานได้ดี แต่จะมีจุดด้อยคือ หากตรวจไม่พบค่าการเพิ่มขึ้นน้อยที่สุดของจำนวนพิกเซลเมื่อทำการปรับค่าเทรซโฮลขั้นที่ 2 ในช่วง 90 - 122 ค่าเทรซโฮลที่ดีที่สุดจะเลือกจากค่าต่ำสุดของการเพิ่มจำนวนพิกเซลของช่วงที่เลือก คือ 92 ซึ่งไม่ใช่ค่าระหว่าง 2 ค่ายอดและอาจค้างกินไปและจะทำให้ได้ผลของการเทรซโฮลไม่ค่อยดี

- ในส่วนของโปรแกรมการกำจัดสิ่งรบกวน และ โปรแกรมการหมุนภาพเพื่อให้มีรูปแบบที่เหมาะสมต่อกระบวนการต่อไป สามารถทำงานได้ดีมาก

- โปรแกรมตรวจหาค่ามุมของรหัสแถบ ทำงานได้ดี แต่จะมีความอ่อนไหวต่อการเปลี่ยนแปลงพารามิเตอร์ของภาพมาก เช่น หากค่าเทรซโฮลเปลี่ยนไปเพียงเล็กน้อย ก็จะทำให้หาค่ามุมผิดไปจากค่ามุมจริงได้ ซึ่งจะต้องระวังควบคุมค่าพารามิเตอร์ต่างๆ ให้เหมาะสม

- โปรแกรมการวาร์ปโดยใช้โพลีโนเมียลลำดับที่ 2 นั้น ทำงานได้ดีกับกรณีที่ภาพต้นแบบที่ถูกถ่ายมาไม่เป็นเพอร์สเปกทีฟ ไม่ว่าจะมีการโค้งงอของขอบของรหัสแถบเป็นรูปพาราโบลาหรือไม่ก็ตาม แต่หากเป็นภาพเพอร์สเปกทีฟแล้ว ขนาดของแต่ละโค้ดเวิร์ดจะไม่เท่ากันตลอดทั้งภาพ ซึ่งการใช้วาร์ปจะไม่สามารถแก้ปัญหานี้ได้ จะมีผลทำให้การถอดรหัสผิดพลาดอย่างมาก ข้อเสียอีกจุดหนึ่งคือ การวาร์ปแบบนี้จะกินเวลาก่อนข้านาน

- ค่าที่ได้จากการถอดรหัสหรือคือค่าในแต่ละโค้ดเวิร์ดของรหัสแถบนั้น จะมีความแม่นยำถูกต้องหรือไม่ขึ้นอยู่กับผลที่ได้จากกระบวนการข้างต้นทั้งหมด โดยเฉพาะการวาร์ป ดังนั้นรูปภาพที่เป็นเพอร์สเปกทีฟจะมีเปอร์เซ็นต์ความผิดพลาดสูง และไม่เหมาะที่จะนำมาใช้กับโปรแกรมที่ออกแบบขึ้นในโครงการนี้

#### 6.2 แนวทางการพัฒนาและปรับปรุงโปรแกรม

ส่วนที่ต้องการการปรับปรุงคือการวาร์ป โดยโปรแกรมควรจะสามารถเลือกใช้ระหว่างการทรานส์ฟอร์มหรือการวาร์ปในแบบเชิงเส้นกับแบบไม่เชิงเส้น ขึ้นอยู่กับลักษณะของรหัสแถบ

ในรูปภาพที่ตรวจจับได้ เพื่อให้โปรแกรมสามารถรับอินพุทเป็นภาพหลาย ๆ แบบได้และมีความ  
แม่นยำในการถอดรหัสสูง

### 6.3 ประโยชน์และการใช้งาน

สามารถนำโปรแกรมที่ออกแบบไปประยุกต์ใช้งานจริง ในการถอดรหัสรหัสแถบแบบใด ๆ ก็ตาม โดยการเปลี่ยนแปลงค่าพารามิเตอร์ของโปรแกรม หรือสามารถนำเทคนิคที่เสนอในโครง  
งานไปประยุกต์ใช้กับการทำงานทางด้านการประมวลผลภาพ (Image Processing) ในรูปแบบอื่น ๆ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
1 %----Find optimal threshold----
2
3 %Increase threshold value from 90 to 122 in steps of 2, the first decrease in rising of
4 %segmented region occurs when approach the optimal theshold
5
6 a = imread('bc5.bmp');
7 [m,n] = size(a);
8 previous = ones(m,n);
9 changelist = [];
10 for threshold = 90:2:122,
11     next = ones(m,n);
12     next(find(a < threshold)) = 0;
13     change = sum(sum(previous - next));
14     changelist = [changelist change];
15     previous = next;
16 end;
17
18 j = size(changelist,2);
19 count = 0;
20 for i = 2:j-1,
21     if changelist(i+1) >= changelist(i),
22         count = count + 1;
23     elseif count < 2,
24         count = 0;
25     else
26         break;
27     end;
28 end;
29
30 if (i == j-1) & (count == j-2),
31     [C,I] = min(changelist);
32     optimal_t = 88 + (I*2);
33 elseif (i == j-1) & (count < 2),
34     disp('Error : Can not find threshold in the selected range!')
35 else
36     optimal_t = 88 + ( (i+1) * 2 );
37 end;
38
39 next = ones(m,n);
40 next(find(a < optimal_t)) = 0;
41 binary = next;
42
43 figure,imshow(a);
44 figure,imshow(binary);
45
46
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
1 %----- Eliminate noise by edge detection -----
2
3 %Detect edges by "sobel" technique and change the edge line into block
4 edge_line = edge(binary,'sobel');
5 figure,imshow(edge_line)
6 [x,y] = find(edge_line == 1);
7 edge_block = edge_line;
8 z = size(x);
9 for i = 1:z(1),
10     if (x(i) > 5 & x(i) <= m-5 & y(i) > 5 & y(i) <= n-5),
11         edge_block(x(i)-5:x(i)+5,y(i)-5:y(i)+5) = 1;
12
13     elseif (x(i) < 6 & y(i) < 6),
14         edge_block(1:x(i)+5,1:y(i)+5) = 1;
15
16     elseif (x(i) < 6),
17         edge_block(1:x(i)+5,y(i)-5:y(i)+5) = 1;
18
19     elseif (y(i) < 6),
20         edge_block(x(i)-5:x(i)+5,1:y(i)+5) = 1;
21
22     elseif (x(i) > m-5 & y(i) > n-5),
23         edge_block(x(i)-5:m,y(i)-5:n) = 1;
24
25     elseif (x(i) > m-5),
26         edge_block(x(i)-5:m,y(i)-5:y(i)+5) = 1;
27
28     elseif (y(i) > n-5),
29         edge_block(x(i)-5:x(i)+5,y(i)-5:n) = 1;
30     end;
31 end;
32 figure,imshow(edge_block)
33
34 %Get rid of all edge blocks that are connected to border of image (except spray noise)
35
36 binary1 = binary;          %'binary1' - barcode is white, background is black
37 for i = 1:m,
38     for j = 1:n,
39         if (binary(i,j) == 0), binary1(i,j) = 1;
40         else binary1(i,j) = 0;
41         end;
42     end;
43 end;
44
45 [L,numobj] = bwlabel(edge_block,8);
46 fill = binary;
47 for i = 1:numobj,
48     [x,y] = find(bwlabel(edge_block) == i);
49     connect1 = find(x == 1 | x == m);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
50 connect2 = find(y == 1 | y == n);
51 if ~(isempty(connect1) | isempty(connect2)),
52     region = bwselect(binary1,y,x,8);
53     [r,c] = find(region == 1);
54     fill = bwfill(fill,c,r,8);    %flood fill of '1' into black region
55     s = size(x);
56     for i = 1:s(1),
57         edge_block(x(i),y(i)) = 0;
58     end;
59 end;
60 end;
61 figure,imshow(fill)
62
63 %Get rid of noise that the region size is smaller than bar code region
64 %including spray noise
65
66 [L,numobj] = bwlabel(edge_block,8);
67 for i = 1:numobj,
68     [x(i),y(i)] = size(find(bwlabel(edge_block) == i));
69 end;
70 [s,obj] = max(x);
71 [r,c] = find(bwlabel(edge_block) == obj);
72 region = bwselect(edge_block,c,r,8);
73 region = bwfill(region,'holes',8);
74 [r,c] = find(region == 1);
75 new = ones(m,n);
76 s = size(r);
77 for i = 1:s(1),
78     new(r(i),c(i)) = binary(r(i),c(i));
79 end;
80 figure,imshow(new)
81
```

```
1  %--Rotate bar code to the horizontal position, which is easy to find corners--
2
3  %Scan from left and right, the biggest region between left & right side is considered
4  %to be left hand-side of barcode
5
6  nonoise = new;          %'nonoise' - barcode is white, background is black
7  for i = 1:m,
8      for j = 1:n,
9          if (new(i,j) == 0), nonoise(i,j) = 1;
10             else nonoise(i,j) = 0;
11         end;
12     end;
13 end;
14
15 %Find first white pixel from left side
16 left = [m n/2];
17 for i = 1:m,
18     for j = 1:n/2,
19         if (nonoise(i,j) == 1),
20             if ( j<=left(2) ),
21                 left = [i j];
22             end;
23         end;
24     end;
25 end;
26 l_region = bwselect(nonoise,left(2),left(1),8);
27 [p,q] = find(l_region == 1);
28 l_size = size(p);
29
30 %Find first white pixel from right side
31 right = [m n/2];
32 for i = 1:m,
33     for j = n-1:n/2,
34         if (nonoise(i,j) == 1),
35             if ( j>=right(2) ),
36                 right = [i j];
37             end;
38         end;
39     end;
40 end;
41 r_region = bwselect(nonoise,right(2),right(1),8);
42 [p,q] = find(r_region == 1);
43 r_size = size(p);
44
45 %If right region is bigger than left region, rotate 180 degrees
46 if r_size(1) > l_size(1),
47     nonoise = imrotate(nonoise,180);
48     figure,imshow(nonoise)
49 end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
50
51 %Find first white pixel from top
52 top = [m/2 n];
53 for i = 1:m/2,
54     for j = 1:n,
55         if (nonoise(i,j) == 1),
56             if ( i<=top(1) ),
57                 top = [i j];
58             end;
59         end;
60     end;
61 end;
62 t_region = bwselect(nonoise,top(2),top(1),8);
63
64 %Find first white pixel from bottom
65 bottom = [m/2 n];
66 for i = m:-1:m/2,
67     for j = 1:n,
68         if (nonoise(i,j) == 1),
69             if ( i>=bottom(1) ),
70                 bottom = [i j];
71             end;
72         end;
73     end;
74 end;
75 b_region = bwselect(nonoise,bottom(2),bottom(1),8);
76
77 %Find new first white pixel from left side
78 left = [m n/2];
79 for i = 1:m,
80     for j = 1:n/2,
81         if (nonoise(i,j) == 1),
82             if ( j<=left(2) ),
83                 left = [i j];
84             end;
85         end;
86     end;
87 end;
88 l_region = bwselect(nonoise,left(2),left(1),8);
89
90 %Find new first white pixel from right side
91 right = [m n/2];
92 for i = 1:m,
93     for j = n:-1:n/2,
94         if (nonoise(i,j) == 1),
95             if ( j>=right(2) ),
96                 right = [i j];
97             end;
98     end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
99     end;
100    end;
101    r_region = bwselect(nonoise,right(2),right(1),8);
102
103    %The bar code patterns are divided into 2 groups, for making them lie in horizontal position
104
105    % 1st group : 2 detected points have the same value (e.g. top = left & bottom = right)
106
107    %Find the amount of '1' pixels in the middle row of image, if the pixels are less than 180,
108    %means bar code lies in vertical, then rotate by 90 degrees to make it lie in horizontal position
109
110    if (top == left & bottom == right) | (top == right & l_region == r_region),
111        [i,j] = find(nonoise(m/2,:) == 1);
112        s = size(i);
113        if s(2) < 180,
114            nonoise = imrotate(nonoise,90,'crop');
115        end;
116    elseif (top == right & bottom == left) | (bottom == right & l_region == r_region),
117        [i,j] = find(nonoise(m/2,:) == 1);
118        s = size(i);
119        if s(2) < 180,
120            nonoise = imrotate(nonoise,-90,'crop');
121        end;
122
123    % 2nd group : none of the points has the same value
124
125    %Find distance between top-left and top-right, the longer distance will indicate the
126    %rotation angle to make bar code lie in horizontal position
127    else
128        [i,j] = find(nonoise(m/2,:) == 1);
129        s = size(i);
130        if s(2) > 180, % If the middle row has '1' more than 180 pixels,
131            break, % doesn't need to find angle to rotate
132        end;
133
134        A = left(1) - top(1);
135        B = left(2) - top(2);
136        t_l = realsqrt(A^2 + B^2); % 't_l' is distance between top and left
137        A = top(1) - right(1);
138        B = top(2) - right(2);
139        t_r = realsqrt(A^2 + B^2); % 't_r' is distance between top and right
140
141        if t_l > t_r,
142            point = left;
143        else
144            point = right;
145        end;
146        A = top(1) - point(1);
147        B = top(2) - point(2);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
48   radian = atan(A/B);           %angle in radian
49   degree = radian*180/pi;      %angle in degree
50   nonoise = imrotate(nonoise,degree,'bilinear','crop'); %Bilinear interpolation gives
51   figure,imshow(nonoise)      %gray scale (0-1)
52   end;
53
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
1 %----corner detection-----
2 %Input image is 'nonoise' (grayscale image)
3
4 %The bar code region should be in the middle of the image.
5 %If the bar code region is too low, it will be raised
6 [p,q] = find(nonoise == 1);
7 if min(p) > 200,
8     new = zeros(m,n);
9     for i = 200:m,
10        for j = 1:n,
11            new(i-50,j) = nonoise(i,j);
12        end;
13    end;
14    figure,imshow(new)
15    nonoise = new;
16 end;
17
18 %Performing dilation and erosion (Close operation) to make the bar code connect
19 %into one region with the same corner points
20 %Make the new picture 'new' with wider space to prevent error when performing dilate and erode
21 new = zeros(m+100,n+100);
22 for i = 1:m,
23     for j = 1:n,
24         new(i+50,j+50) = nonoise(i,j);
25     end;
26 end;
27
28 %Perform dilation and erosion
29 %From this step, 'bw'-binary image is used as an input to find corners
30 se = ones(50,50); %'se'-matrix controlling thickness of dilation
31 bw = dilate(new,se);
32 bw = erode(bw,se);
33 new = zeros(m,n);
34 for i = 1:m,
35     for j = 1:n,
36         new(i,j) = bw(i+50,j+50); %Change image back to the same size as original
37     end;
38 end;
39 bw = new;
40 figure,imshow(bw)
41
42 %Find corners from 'bw'
43 c = zeros(1,4); %Detected corner values will be stored in 'c' & 'd'
44 d = zeros(1,4);
45
46 %Find upper left corner from middle point to left side and up
47 x = m/2;
48 y = n/2;
49 for i = 1:10, %Make loop to prevent error in the case that
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
50 while bw(x,y) == 1,           %bar code is not 0 degree to the horizontal line
51     y = y-1;
52 end;
53 y = y+1;
54 while bw(x,y) == 1,
55     x = x-1;
56 end;
57 x = x+1;
58
59 if (bw(x-1,y-2:y+2) == 0) & (bw(x+1,y-1) == 1), %This case prevent error if bar code
60     x = x+1;           %pattern is curve when there is no '1'
61     y = y-1;           %pixel left in the upper row
62 elseif (bw(x-1,y-2:y+2) == 0), %Check the upper row, if there's no '1' pixel left, stop
63     break,
64 elseif (bw(x-1,y-2:y-1) == 0), %The region which slopes from right to left will
65     while bw(x-1,y+2) == 1, %be detected one pixel at a time by scanning upward
66         x = x-1;           %until there is not '1' pixel left in the upper row
67         y = y+1;
68     end;
69     x = x+1;
70     y = y-1;
71 end;
72
73 end;
74 c(1) = x;
75 d(1) = y;
76
77 %Find upper right corner from middle point to right side and up
78 x = m/2;
79 y = n/2;
80 for i = 1:10,
81     while bw(x,y) == 1,
82         y = y+1;
83     end;
84     y = y-1;
85     while bw(x,y) == 1,
86         x = x-1;
87     end;
88     x = x+1;
89
90 if (bw(x-1,y-2:y+2) == 0) & (bw(x+1,y+1) == 1), %This case prevent error if bar code
91     x = x+1;           %pattern is curve when there is no '1'
92     y = y+1;           %pixel left in the upper row
93 elseif (bw(x-1,y-2:y+2) == 0),
94     break,
95 elseif (bw(x-1,y+1:y+2) == 0), %The region which slopes from left to right will
96     while bw(x-1,y-2) == 1, %be detected one pixel at a time by scanning upward
97         x = x-1;           %until there is no '1' pixel left in the upper row
98         y = y-1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
99     end;
100    x = x+1;
101    y = y+1;
102    end;
103
104 end;
105 c(2) = x;
106 d(2) = y;
107
108 %Find lower left corner from middle point to left side and down
109 x = m/2;
110 y = n/2;
111 for i = 1:10,
112     while bw(x,y) == 1,
113         y = y-1;
114     end;
115     y = y+1;
116     while bw(x,y) == 1,
117         x = x+1;
118     end;
119     x = x-1;
120
121     if (bw(x+1,y-2:y+2) == 0) & (bw(x-1,y-1) == 1), %This case prevent error if bar code
122         x = x-1; %pattern is curve when there is no '1'
123         y = y-1; %pixel left in the lower row
124     elseif (bw(x+1,y-2:y+2) == 0),
125         break,
126     elseif (bw(x+1,y-2:y-1) == 0), %The region which slopes from left to right will
127         while bw(x+1,y+2) == 1, %be detected one pixel at a time by scanning downward
128             x = x+1; %until there is no '1' pixel left in the lower row
129             y = y+1;
130         end;
131         x = x-1;
132         y = y-1;
133     end;
134
135 end;
136 c(3) = x;
137 d(3) = y;
138
139 %Find lower right corner from middle point to right and down
140 x = m/2;
141 y = n/2;
142 for i = 1:10,
143     while bw(x,y) == 1,
144         y = y+1;
145     end;
146     y = y-1;
147     while bw(x,y) == 1,
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
148     x = x+1;
149     end;
150     x = x-1;
151
152     if (bw(x+1,y-2:y+2) == 0) & (bw(x-1,y+1) == 1),    %this case prevent error if bar code
153         x = x-1;                                        %pattern is curve when there is no '1'
154         y = y+1;                                        %pixel left in the lower row
155     elseif (bw(x+1,y-2:y+2) == 0),
156         break,
157     elseif (bw(x+1,y+1:y+2) == 0),    %The region which slopes from right to left will
158         while bw(x+1,y-2) == 1,    %be detected one pixel at a time by scanning downward
159             x = x+1;    %until there is no '1' pixel left in the lower row
160             y = y-1;
161         end;
162         x = x-1;
163         y = y+1;
164     end;
165
166 end;
167 c(4) = x;
168 d(4) = y;
169
170 %Display the detected corners
171 window = [1 1 1 1 1
172           1 0 0 0 1
173           1 0 0 0 1
174           1 0 0 0 1
175           1 1 1 1 1];
176 show = nonoise;
177 for i = 1:4,
178     show( c(i)-2:c(i)+2 , d(i)-2:d(i)+2 ) = window;
179 end;
180 figure,imshow(show)
181
182 %Corner values
183 up_left = [c(1) d(1)];
184 up_right = [c(2) d(2)];
185 lo_left = [c(3) d(3)];
186 lo_right = [c(4) d(4)];
187
```



```
1 %---Polynomial warping---
2 % This process will follow the proper x & y coordinate (vertical & horizontal)
3
4 % template landmark points (input image or distorted image)
5 U = zeros(8,1);
6 V = zeros(8,1);
7
8 % target landmark points (output image)
9 X = zeros(8,1);
10 Y = zeros(8,1);
11
12 %- Assign input control points -%
13
14 V(1:4,1) = [ up_left(1); up_right(1); lo_left(1); lo_right(1) ];
15 U(1:4,1) = [ up_left(2); up_right(2); lo_left(2); lo_right(2) ];
16
17 up_width = up_right(2) - up_left(2);
18 mid_x1 = round( up_left(2) + up_width/2 );
19 lo_width = lo_right(2) - lo_left(2);
20 mid_x2 = round( lo_left(2) + lo_width/2 );
21 if up_width < lo_width,
22     width = up_width;
23     mid_x = mid_x1;
24     u_left = up_left(2); % width of output image (u_left to u_right)
25     u_right = up_right(2);
26 else
27     width = lo_width;
28     mid_x = mid_x2;
29     u_left = lo_left(2);
30     u_right = lo_right(2);
31 end;
32
33 l_h = lo_left(1) - up_left(1);
34 mid_y1 = round( up_left(1) + l_h/2 );
35 r_h = lo_right(1) - up_right(1);
36 mid_y2 = round( up_right(1) + r_h/2 );
37 if l_h < r_h,
38     height = l_h; % height of output image
39     mid_y = mid_y1;
40 else
41     height = r_h;
42     mid_y = mid_y2;
43 end;
44
45
46 % Other 4 points are assigned from the middle points of bar code in vertical & horizontal axis
47 % Use the 'bw' image after Close operation for finding the rest control points
48 for i = 1:m/2,
49     if bw(i,mid_x1) == 1,
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
50     V(5,1) = i;
51     U(5,1) = mid_x1;
52     break;
53     end;
54 end;
55
56 for i = m:-1:m/2,
57     if bw(i,mid_x2) == 1,
58         V(6,1) = i;
59         U(6,1) = mid_x2;
60         break;
61     end;
62 end;
63
64 for j = 1:n/2,
65     if bw(mid_y1,j) == 1,
66         V(7,1) = mid_y1;
67         U(7,1) = j;
68         break;
69     end;
70 end;
71
72 for j = n:-1:n/2,
73     if bw(mid_y2,j) == 1,
74         V(8,1) = mid_y2;
75         U(8,1) = j;
76         break;
77     end;
78 end;
79
80
81 %- Assign output control points -%
82
83 up = up_left(1);
84 lo = up + height;
85 Y = [ up; up; lo; lo; up; lo; mid_y; mid_y ];
86 X = [ u_left; u_right; u_left; u_right; mid_x; mid_x; u_left; u_right ];
87
88
89 %-- Create data matrix A --%
90
91 A = zeros(8,6);
92 A(:,1) = ones(8,1);
93 A(:,2) = X;
94 A(:,3) = Y;
95 A(:,4) = X.^2;
96 A(:,5) = X.*Y;
97 A(:,6) = Y.^2;
98
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
99 temp = inv(A*A)*A';
100 aa = temp*U; % weighting coefficient a, 'aa'
101 bb = temp*V; % weighting coefficient b, 'bb'
102
103 output = zeros(m,n);
104 data = nonoise;
105
106 % Compute output matrix, which is converted to binary image in this step
107 for i=1:n-1,
108     for j=1:m-1,
109         vect = [1 i j i^2 i*j j^2];
110         xori = vect * aa;
111         yori = vect * bb;
112         xoriint = fix(xori);
113         yoriint = fix(yori);
114         difx = xori-xoriint;
115         dify = yori-yoriint;
116         if (xoriint>=n) | (yoriint >=m) | (xoriint<1) | (yoriint<1),
117             output(j,i)=0;
118         else
119             output(j,i) = round( (1-dify) * ((1-difx)*data(yoriint,xoriint) + ...
120                 difx*data(yoriint,xoriint+1)) + ...
121                 dify*( (1-difx)*data(yoriint+1,xoriint) + ...
122                 difx*data(yoriint+1,xoriint+1)));
123         end;
124     end;
125 end;
126
127 % Show 8 control points
128 window = [1 1 1 1
129           1 0 0 1
130           1 0 0 1
131           1 0 0 1
132           1 1 1 1];
133 show_cp = bw;
134 for i = 1:8,
135     show_cp( V(i)-2:V(i)+2 , U(i)-2:U(i)+2 ) = window;
136 end;
137 figure,imshow(show_cp)
138
139 figure,imshow(nonoise)
140 figure,imshow(output)
141
142 [x,y] = find(output == 1);
143 up = min(x);
144 lo = max(x);
145 left = min(y);
146 right = max(y);
147 crop = imcrop(output,[left up right-left lo-up]);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
148 figure,imshow(crop)           %crop region from 'output'  
149  
150 %Use morphological opening for eliminating noise after image being transformed  
151 se = strel('square',3);  
152 final = imopen(crop,se);  
153 figure, imshow(final), title('final output')  %'final' is used for detecting codeword  
154 m_c = size(final,1);           %height of final picture  
155 n_c = size(final,2);           %width of final picture  
156
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
1 %---- Detect sequence of codewords -----
2
3 %** this program is used with 2 column (55 codewords/row) by 20 rows symbol**
4
5 crop = final;
6 height = m_c/20;           %height of one codeword (not int)
7 width = n_c/55;           %width of one codeword (not int)
8 for i = 1:height,
9     if crop( i,floor(width-2) ) == 1,
10         break,
11     end;
12 end;
13 line = i + 2;             %row to be scan
14 num_code = round( (n_c) / width ); %'num_code' has to be equal to 55
15 code = zeros(20,num_code);
16
17 %If the number of '1' pixels in that codeword is more or equal to 60% of width,
18 %that codeword has value '1' (which is black part of bar code)
19
20 for i = 1:19,             %Decode from row 1 - 19
21     x = 1;
22     row = crop( floor(line),1:n_c );
23     for j = 1:num_code,
24         if x+width > size(row,2) + 2,
25             break,
26         end;
27         if sum( row( floor(x) : round(x+width-1) ) ) >= floor(width*0.6),
28             code(i,j) = 1;
29         else
30             code(i,j) = 0;
31         end;
32         x = x + width;
33     end;
34     line = line + height;
35 end;
36
37 i = 20;                 % To prevent error in the last row detection in the case that
38 line = floor(line) - 2; % bar code region is sloping
39 for r = 1:3,
40     x = 1;
41     row = crop( line,1:n_c );
42     for j = 1:num_code,
43         if x+width > size(row,2) + 2,
44             break,
45         end;
46         if sum( row( floor(x) : round(x+width-1) ) ) >= floor(width*0.6),
47             last(r,j) = 1;
48         else
49             last(r,j) = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
50     end;
51     x = x + width;
52     end;
53     line = line + 1;
54     end;
55     s = sum(last');
56     [C,l] = max(s);
57     code(20,:) = last(l,:);
58
59     for i = 1:20,
60         for j = 1:55,
61             if (code(i,j) == 0), code(i,j) = 1;
62             else code(i,j) = 0;
63             end;
64         end;
65     end;
66     code          % Show detected codewords
67
68     %Change 'final' image back to '0' for black region, '1' for white region
69     for i = 1:m_c,
70         for j = 1:n_c,
71             if (final(i,j) == 0), final(i,j) = 1;
72             else final(i,j) = 0;
73             end;
74         end;
75     end;
76
```



```
>> *--bcl.bmp (input image)--
>> code
```

code =

Columns 1 through 15

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

Columns 16 through 30

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

Columns 31 through 45

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

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

```

Columns 46 through 55

```

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

```



>>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
>> *--bc2.bmp (input image two)--
>> code
```

code =

Columns 1 through 15

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

Columns 16 through 30

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

Columns 31 through 45

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

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

```

Columns 46 through 55

```

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

```



>>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
>> %--bc3.bmp (input image three)--
>> code
```

code =

Columns 1 through 15

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

Columns 16 through 30

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

Columns 31 through 45

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

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

```

Columns 46 through 55

```

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

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

>> \*--bc4.bmp (input image four)--

>> code

code =

Columns 1 through 15

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

Columns 16 through 30

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

Columns 31 through 45

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

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

```

Columns 46 through 55

```

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

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
>> %--bc5.bmp (input image five)--
>> code
```

code =

Columns 1 through 15

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

Columns 16 through 30

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

Columns 31 through 45

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

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

```

Columns 46 through 55

```

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

```



>>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

>> \*--bc6.bmp (input image six)--

>> code

code =

Columns 1 through 15

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

Columns 16 through 30

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

Columns 31 through 45

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

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

```

Columns 46 through 55

```

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

```



>>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บรรณานุกรม

- [1] AIM, Inc., *International Symbology Specification-MicroPDF417*, AIM, 1998.
- [2] I. Pitas, *Digital Image Processing Algorithms and Application*, John Wiley & Sons, 2000.
- [3] William K. Pratt, *Digital Image Processing*, John Wiley & Sons, 1991.
- [4] ดวงพร ฉัตรวีระชัยกิตฺร, นพวรรณ โททัตสา, การรู้จำภาพใบหน้าคน (*Face Recognition*), ปริญญานิพนธ์สำหรับปริญญาวิศวกรรมศาสตรบัณฑิต สาขาอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2543.
- [5] Siriporn R., and Akira Kurematsu, *2-Dimensional Bar Code Image Analysis*, Proceedings of The UEC International Mini-Conference for Exchange Students X (JUSST Program), 2003.
- [6] <http://www-cs-students.stanford.edu/~robles/ee368/skinsegment.html>
- [7] <http://selectservices.bentley.com/technotes/technotes/8254.htm>
- [8] <http://www.hibcc.org/autodupn/docs/SmallPackageDraft.doc>
- [9] <http://www.aimglobal.org/>
- [10] <http://www.adams1.com/pub/russadam/stack.html>
- [11] [http://www.currentdirections.com/facts/sizing\\_applications\\_for\\_2d\\_symbols.html](http://www.currentdirections.com/facts/sizing_applications_for_2d_symbols.html)
- [12] [http://www.ean.com.au/hipr/html/hipr\\_top.html](http://www.ean.com.au/hipr/html/hipr_top.html)
- [13] [http://www.symbol.com/products/barcode\\_scanners/](http://www.symbol.com/products/barcode_scanners/)