

กระบวนการจัดรอยขีดข่วนบนภาพคิวอาร์โค้ด

นายกมล หอมขจร

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2555

บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2555 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ที่ส่งผ่านทางบัณฑิตวิทยาลัย

The abstract and full text of theses from the academic year 2011 in Chulalongkorn University Intellectual Repository(CUIR) are the thesis authors' files submitted through the Graduate School.

A PROCESS TO REMOVE SCRATCHES FROM QR CODE IMAGES

Mr. Kamon Homkajorn

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science Program in Computer Science

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2012

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์

กระบวนการจัดรอยขีดข่วนบนภาพคิวอาร์โค้ด

โดย

นายกมล หอมขจร

สาขาวิชา

วิทยาศาสตร์คอมพิวเตอร์

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

รองศาสตราจารย์ ดร.สาธิต วงศ์ประทีป

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้บัณฑิตวิทยานิพนธ์ฉบับนี้เป็นส่วน
หนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

.....คณบดีคณะวิศวกรรมศาสตร์

(รองศาสตราจารย์ ดร.บุญสม เลิศหิรัญวงศ์)

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ

(รองศาสตราจารย์สมชาย ประสิทธิ์จตุระกุล)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

(รองศาสตราจารย์ ดร.สาธิต วงศ์ประทีป)

.....กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.เศรษฐา ปานงาม)

.....กรรมการภายนอกมหาวิทยาลัย

(ผู้ช่วยศาสตราจารย์ ดร.วิไลพร แซ่ลี้)

กมล หอมขจร : กระบวนการขจัดรอยขีดข่วนบนภาพคิวอาร์โค้ด.

(A PROCESS TO REMOVE SCRATCHES FROM QR CODE IMAGES

อ.ที่ปรึกษาวิทยานิพนธ์หลัก : รศ.ดร.สาธิต วงศ์ประทีป, 122 หน้า.

งานวิจัยนี้นำเสนอเกี่ยวกับกระบวนการขจัดรอยขีดข่วนบนภาพ QR Code โดยที่มีการนำเทคนิคการประมวลผลภาพ (Digital image processing) ซึ่งมีขั้นตอนประกอบไปด้วยการแบ่งแยกภาพโดยใช้แบบจำลองสี HSV (Color segmentation using HSV Model) เพื่อทำการแยกความต่างของสีของ QR Code กับรอยขีดข่วนแล้วนำภาพรอยที่ได้เข้าสู่การทำ Binarization ซึ่งเป็นขั้นตอนการแปลงภาพรอยให้เป็นภาพไบนารี (Binary Image), การทำ Morphological Image Processing โดยใช้เทคนิค Opening ร่วมกับ Closing เพื่อปรับปรุงโครงสร้างของภาพรอยที่ได้ให้มีความสมบูรณ์มากยิ่งขึ้นแล้วนำตำแหน่งรอยที่ได้ไปใช้ในการลบรอยออกจากภาพที่มีรอยขีดข่วนแล้วเข้าสู่ขั้นตอนการกำหนดพื้นที่ที่สนใจ (Region of Interest : ROI) เพื่อให้ได้ QR Code ที่ไม่มี Quiet Zone แล้วทำการวาดเส้นตาราง (Grids Generation) เพื่อกำหนดพื้นที่ย่อยของ QR Code โดยใช้ขนาดโมดูลของ QR Code ในเวอร์ชันที่ใช้ในการทดลองเมื่อได้พื้นที่ของโมดูลย่อยแล้วทำการวิเคราะห์กลุ่มสมาชิกของพื้นที่ย่อยเพื่อทำการเติมเต็มส่วนของ QR Code ที่เสียหายในแต่ละโมดูล (Reconstruction QR Code) และในขั้นตอนสุดท้ายเมื่อผ่านกระบวนการต่างๆ จะได้ภาพผลลัพธ์ของ QR Code ที่สามารถนำไปอ่านเพื่อถอดรหัสโดยใช้ซอฟต์แวร์ที่มีอยู่ได้อย่างมีประสิทธิภาพ

ภาควิชา..... วิทยาศาสตร์คอมพิวเตอร์..... ลายมือชื่อนิสิต.....
 สาขาวิชา..... วิศวกรรมคอมพิวเตอร์..... ลายมือชื่ออ.ที่ปรึกษาวิทยานิพนธ์หลัก.....
 ปีการศึกษา..... 2555.....

5470104921: COMPUTER SCIENCE

KEYWORDS: QR CODE/2D BARCODE/QUICK RESPOND CODE

KAMON HOMKAJORN: A PROCESS TO REMOVE SCRATCHES FROM QR CODE IMAGES. ADVISOR: ASSOC. PROF. SARTID VONGPRADHIP, Ph.D., 122 pp.

This thesis presents a process to remove scratches from QR Code Image. The Digital Image Processing technique involves and breaks down into several processes: HSV (Color segmentation using HSV Model), differentiate color of two-dimensional QR code, segment the scratches locate on QR code, and convert the damage (scratches) into Binary image. Morphological Image processing techniques for Opening Closing with the restructuring of the tracks in order to make the mark even come close to perfect mark. Then compare the mark with damage QR code with scratches, this lead to Region of Interest: ROI. ROI technique will remove quietzone and generate Grids Generation to determine segmentation area of QR code. The tiny area (module) after segmentation is complete, group of pixels of each tiny area is analyze in order to fill and fix up the damage QR code of each module. Through various processes, reconstruction QR Code is the last process and as the result, the product of the last process is ready for the QR code ready software to decode with efficiency.

Department : Computer Engineering Student's Signature

Field of Study : Computer Science Advisor's Signature

Academic Year : 2012

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี ขอกราบขอบพระคุณรองศาสตราจารย์ ดร.สาธิต วงศ์ประทีป ซึ่งเป็นอาจารย์ที่ปรึกษาวิทยานิพนธ์ ท่านได้กรุณาสละเวลาและดูแลการวิจัย ให้คำแนะนำเกี่ยวกับการทำวิจัย และการสนับสนุนเป็นอย่างดีจนทำให้การวิจัยในครั้งนี้สำเร็จ ออกมาด้วยดี

ขอกราบขอบพระคุณคณะกรรมการสอบวิทยานิพนธ์ทุกท่านเป็นอย่างสูง ได้แก่ รองศาสตราจารย์สมชาย ประสิทธิ์จูตระกูล ผู้ช่วยศาสตราจารย์ ดร.เศรษฐา ปานงาม และ ผู้ช่วยศาสตราจารย์ ดร.วิไลพร แซ่ลี้ ที่กรุณาเสียสละเวลา ให้คำแนะนำในการจัดทำวิทยานิพนธ์ ฉบับนี้

ขอขอบพระคุณอาจารย์ประจำภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย ทุกท่านเป็นอย่างสูงที่ให้ข้อคิดและแนวทางในการวิจัย ขอขอบคุณเจ้าหน้าที่ประจำภาควิชา วิศวกรรมคอมพิวเตอร์ทุกท่าน เพื่อนๆทุกคนในห้องวิจัย Digital System Engineering Laboratory (DSEL) รวมถึงเพื่อนๆ ในรุ่น CS 41 ทุกคนที่เป็นกำลังใจ และให้ความช่วยเหลือตลอดจน คำแนะนำต่างๆที่เป็นประโยชน์ต่อการวิจัย

ขอกราบขอบพระคุณ คุณพ่อ-คุณแม่ และญาติ ๆ ที่ให้การสนับสนุนและเป็นกำลังใจที่ดี เสมอ ขอขอบคุณ คุณสเตฟาน เอนชเนอร์และคุณมาติน ชูเลอร์ ที่ช่วยให้คำปรึกษาเกี่ยวกับเทคนิค การเขียน โปรแกรมรวมทั้งช่วยอำนวยความสะดวกในการไปร่วมการประชุมนานาชาติที่ ต่างประเทศ ขอขอบคุณ คุณวิรุศต์ โกศลวิจักขณ์ ที่ช่วยในการเรียบเรียงและตรวจสอบ ภาษาอังกฤษให้ ขอขอบคุณ คุณลลิตา สันติวรรักษ์ ที่ช่วยแนะแนวทางการเรียนและการวิจัยต่างๆ

ท้ายที่สุด ผู้วิจัยขอขอบคุณสุภาพบุรุษเรื่องศิลป์และคุณมหศักดิ์ เกตุนำที่คอยติดตามแก้ไข งานวิจัย การปรับปรุงพรีเซนเทชัน และเปเปอร์ที่ไปนำเสนอ ขอขอบคุณท่านอื่น ๆ ที่มีได้กล่าวชื่อ ไว้ ณ ที่นี้ที่มีส่วนทำให้วิทยานิพนธ์สำเร็จได้ด้วยดี

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ง
บทคัดย่อภาษาอังกฤษ	จ
กิตติกรรมประกาศ.....	ฉ
สารบัญ	ช
สารบัญตาราง	ฎ
สารบัญภาพ	ฏ

บทที่

1	บทนำ.....	1
1.1	ความเป็นมาและความสำคัญของปัญหา	1
1.2	วัตถุประสงค์ของการวิจัย	2
1.3	ขอบเขตของการวิจัย.....	2
1.4	ข้อตกลงเบื้องต้น	3
1.5	ประโยชน์ที่คาดว่าจะได้รับ	4
1.6	ลำดับขั้นตอนในการเสนอผลการวิจัย	4
2	ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	5
2.1	ข้อมูลทั่วไปเกี่ยวกับบาร์โค้ด.....	5
2.2	คิวอาร์โค้ด (QR Code).....	9
2.3	กระบวนการเข้ารหัส QR Code	14
2.3.1	การวิเคราะห์กระแสข้อมูล Input	14
2.3.2	การเข้ารหัสข้อมูล (Data encoding).....	16
2.3.3	Error Correction Coding	16
2.3.4	ทำการเตรียม Structure final message	17
2.3.5	ขั้นตอน Module placement in matrix	18
2.3.6	ขั้นตอนการวาง Data masking.....	19
2.3.7	ขั้นตอนการวาง Format information และ Version information.....	22

2.4	ขั้นตอนในการถอดรหัส QR Code	25
2.4.1	ทำการแปลงภาพที่นำเข้าโดยการแปลงเป็นภาพขาวดำ.....	25
2.4.2	ทำการค้นหา Finder Pattern	25
2.4.3	ทำการพิจารณาค่าแห่งการหมุน	26
2.4.4	พิจารณาระยะทางของ D	26
2.4.5	ทำการคำนวณหาค่า Nominal X Dimension	27
2.4.6	สมการที่ใช้ในการคำนวณ Version (V).....	27
2.4.7	เงื่อนไขการคำนวณ Version.....	27
2.4.8	การคำนวณสำหรับใน Version 1.....	28
2.4.9	Global Threshold พื้นที่ตัวอย่าง	32
2.4.10	ทำการถอดรหัส Format information.....	32
2.4.11	ในกรณีที่ Format information.....	32
2.4.12	ทำการ XOR ส่วนของ Data Mask Pattern	32
2.4.13	ทำการพิจารณาสัญลักษณ์ของ Codewords.....	32
2.4.14	ทำการจัดลำดับของ Codeword.....	32
2.4.15	ทำการตรวจหาข้อผิดพลาด.....	33
2.4.16	ทำการ Restore ข้อมูลต้นฉบับ.....	33
2.4.17	ทำการแบ่ง Data bit stream.....	33
2.4.18	ทำการถอดรหัส	33
2.5	พื้นฐานการประมวลผลภาพดิจิทัล	33
2.5.1	รูปแบบของภาพดิจิทัล	33
2.5.2	กระบวนการในการสร้างภาพดิจิทัล	34
2.5.3	คุณสมบัติของสี	37
2.5.4	แบบจำลองสี (Color Models).....	37
2.5.5	การแปลงค่าสีจากแบบจำลองสี RGB ไปแบบจำลองสี HSV	41
2.5.6	การแปลงค่าสีจากแบบจำลองสี HSV ไปแบบจำลองสี RGB.....	44
2.6	ประเภทของการประมวลผลภาพ (Image processing).....	45

2.6.1	การกระทำกับจุดของภาพ (Point Processing)	45
2.6.2	การกระทำกับภาพเฉพาะส่วน (Local Processing).....	46
2.6.3	การกระทำกับภาพทั้งหมด (Global Processing).....	48
2.6.3.1	การหาค่าเทรชโวลด์หนึ่งระดับ (Single Thresholding)	48
2.6.3.2	การหาค่าเทรชโวลด์หลายระดับ (Multilevel Thresholding)	49
2.7	มอร์โฟโลยี (Mophology).....	50
2.7.1	การขยายภาพ (Dilation)	50
2.7.2	การเซาะภาพ (Erosion).....	52
2.7.3	การเปิดภาพ (Opening) และการปิดภาพ (Closing)	53
2.8	การแปลงข้อมูลภาพสองมิติ (Two Dimension Geomatic Transformation).....	56
2.8.1	การเลื่อนจุดพิกัดภาพ (Translation).....	56
2.8.2	การหมุนภาพ (Rotation).....	56
2.8.3	การย่อและขยายภาพ (Scaling).....	57
2.9	เกณฑ์การวัดประสิทธิภาพ	59
2.9.1	การวัดค่าความเหมือนของภาพ (Similarity Measurement)	59
2.9.2	เกณฑ์การวัดเวลาที่ใช้ในการประมวลผล (Time of Process)	60
2.10	เอกสารและงานวิจัยที่เกี่ยวข้อง	60
3	วิธีดำเนินการวิจัย.....	63
3.1	ภาพรวมของทั้งหมดของงานวิจัย	63
3.2	การแปลงแบบจำลองสีจาก RGB ไปยัง HSV.....	66
3.3	กระบวนการแยกองค์ประกอบของภาพ	68
3.3.1	การแยกองค์ประกอบของรอยขีดข่วน	68
3.3.2	การเตรียมภาพนำเข้า (Input Image Preparation).....	71
3.3.3	กระบวนการลบรอยขีดข่วน (Scratches Removing).....	71
3.3.4	การกรองสัญญาณรบกวน (Filtering).....	73
3.3.5	การแบ่งพื้นที่ย่อยของ QR Code (Grid Generating).....	74
3.3.6	กระบวนการเติมเต็มโมดูลภาพใน QR Code (Reconstruction QR Code)	76
3.3.7	การถอดรหัสข้อมูล QR Code (Decoding QR Code).....	79

บทที่	หน้า
3.4	เกณฑ์การวัดประสิทธิภาพ 80
3.4.1	การวัดค่าความเหมือนของภาพ (Similarity Measurement) 80
3.4.2	การวัดเวลาที่ใช้ในการประมวลผล..... 80
4	การทดลองและผลการทดลอง 81
4.1	เครื่องมือที่ใช้ในการทดลอง 81
4.2	ผลการทดลอง..... 82
5	สรุปผลการวิจัย 98
5.1	บทสรุป..... 98
5.2	ข้อเสนอแนะ..... 100
	รายการอ้างอิง 101
	ภาคผนวก 104
	ประวัติผู้เขียนวิทยานิพนธ์..... 122

สารบัญตาราง

ตารางที่	หน้า
2-1 แสดงความจุของ QR Code แบ่งตามประเภทข้อมูล.....	10
2-2 Mode Indicators ของ QR Code	14
2-3 แสดงเวอร์ชันและความจุของ QR Code ในแต่ละเวอร์ชัน	14
2-4 จำนวนบิต Character Count Indicator ของ QR Code	16
2-5 แสดงระดับ Error correcting ใน QR Code	17
2-6 แสดงเงื่อนไขที่ใช้ในการเลือก Pattern	20
2-7 Indicators ของ Error Correction Level ใน QR Code	22
2-8 แสดงค่าระดับสีใน HSV color model.....	42
3-1 ตัวอย่างของภาพใน HSV เมื่อมีการเปลี่ยนค่าสี.....	67
4-1 แสดงผลลัพธ์ของรอยขีดข่วนจากปากกาขนาด 0.4 mm	82
4-2 แสดงผลลัพธ์ของรอยขีดข่วนจากปากกาขนาด 2.0 mm	84
4-3 แสดงผลลัพธ์ของรอยขีดข่วนจากปากกาขนาด 3.5 mm	86
4-4 แสดงผลลัพธ์ของรอยขีดข่วนจากปากกาขนาด 5 mm	88
4-5 ผลการทดลองเฉลี่ยในรอยขีดข่วนขนาด 0.4 mm	90
4-6 ผลการทดลองเฉลี่ยในรอยขีดข่วนขนาด 2 mm	92
4-7 ผลการทดลองเฉลี่ยในรอยขีดข่วนขนาด 3.5 mm	94
4-8 ผลการทดลองเฉลี่ยในรอยขีดข่วนขนาด 5 mm	96

สารบัญภาพ

ภาพที่	หน้า
2-1 บาร์โค้ดแบบ 1 มิติ	5
2-2 ตัวอย่างบาร์โค้ดแบบสแต็ก PDF417	6
2-3 ตัวอย่างบาร์โค้ด 2 มิติ และตำแหน่งของ Finder Pattern	7
2-4 ตัวอย่างการใช้งานบาร์โค้ด 2 มิติ	8
2-5 การถอดรหัส QR Code โดยใช้โทรศัพท์มือถือ	9
2-6 ตัวอย่างการใช้ QR Code ในบริการ Favorite Place	9
2-7 จำนวนของโมดูลใน QR Code Version 1 และ Version 2	11
2-8 แสดงมุมมองของ QR Code ที่สามารถถอดรหัสได้	11
2-9 โครงสร้างของ QR Code ตามมาตรฐานสัญลักษณ์ QR Code 2005	12
2-10 โครงสร้างของ Finder Pattern ใน QR Code	12
2-11 แสดงรูปแบบการวาง Data Codeword และ Error correction codewords	18
2-12 ทิศทางการวางข้อมูลในทิศทางขึ้นและลง	18
2-13 ตัวอย่างทิศทางการวางในรูปแบบปกติและรูปแบบไม่ปกติ	19
2-14 ตัวอย่างการวางบิตในจุดที่ใกล้เคียงกับ Alignment pattern	19
2-15 รูปแบบ Data mask pattern ทั้งหมด	21
2-16 แผนภาพขั้นตอนการวาง Data Mask	21
2-17 ตำแหน่งการวาง Format information	23
2-18 ตำแหน่งของ Version information	24
2-19 ทิศทางการวาง Version information	24
2-20 แสดงแผนผังการถอดรหัส QR Code	25
2-21 การแสดงไลน์ ใน Finder Pattern	26
2-22 ตำแหน่งของ Finder pattern และ Version information	28
2-23 ตำแหน่งของ Finder patterns และ Alignment pattern ในการ Sampling grids	29
2-24 จุดศูนย์กลางของ Alignment Pattern	30
2-25 สัญลักษณ์ด้านบนซ้ายของ QR Code	31
2-26 สัญลักษณ์ในบริเวณขวาล่างของ QR Code	32

ภาพที่	หน้า
2-27 ตัวอย่างรูปภาพในรูปแบบ Bitmap และ Vector	34
2-28 ตัวอย่างการทำ Image Sampling.....	35
2-29 ภาพที่ได้จากอัตราการสุ่มที่แตกต่างกัน.....	35
2-30 ภาพในระดับเทาที่มีค่าตั้งแต่ 0 -255	36
2-31 ภาพที่มีระดับเทา 8 bits 4 bits 2 bits และ 1 bit.....	36
2-32 ค่าความยาวของคลื่นสีที่ตามนุษย์สามารถมองเห็น	37
2-33 มุมสีดำในแบบจำลองสี RGB.....	38
2-34 มุมสีขาวในแบบจำลองสี RGB	38
2-35 แบบจำลองสี HSV	39
2-36 HSV Curve	40
2-37 แสดงภาพต้นฉบับและผลลัพธ์ที่ได้จากการปรับค่า HSV	41
2-38 ตัวอย่างภาพในแบบจำลองสี RGB และ แบบจำลองสี HSV	43
2-39 การกระทำต่อจุดภาพ (Point Processing)	45
2-40 การกระทำกับภาพเฉพาะส่วน (Local Processing).....	46
2-41 แสดงการทำ Zero Padding.....	48
2-42 การเลือกค่าเทรชโวลด์จากภาพฮิสโตแกรม.....	49
2-43 ภาพผลลัพธ์จาก Single Thresholding และ Multilevel Thresholding	50
2-44 ตัวอย่างการขยายภาพ (Dilation)	51
2-45 ตัวอย่างการเซาะภาพ (Erosion).....	52
2-46 ตัวอย่างการทำการเปิด (Opening)	53
2-47 ตัวอย่างการทำการปิด (Closing)	54
2-48 ตัวอย่างการประยุกต์ใช้มอร์โฟโลยี.....	55
2-49 การหมุนภาพจากจุดกำเนิด.....	56
2-50 ภาพตัวอย่างการทำการย่อขยายภาพ (Scaling)	58
3-1 ภาพรวมทั้งหมดของงานวิจัย.....	65
3-2 ตัวอย่างภาพที่นำเข้า (Input Image)	66
3-3 ภาพที่นำเข้าไปในแบบจำลองสี RGB	66
3-4 ผลลัพธ์การแปลงค่าสีจากแบบจำลองสี RGB ไปยัง HSV	66

ภาพที่	หน้า
3-5 ค่าระดับเทาของ Saturation Layer.....	68
3-6 แผนภูมิฮิสโตแกรมของภาพระดับเทาในส่วนของ Saturation.....	69
3-7 ภาพไบนารีของรอยขีดข่วน	69
3-8 รอยขีดข่วนที่มีการซึมของหมึก	69
3-9 ช่องว่างในภาพรอยขีดข่วน	70
3-10 ผลลัพธ์ที่ได้จากการทำการปิดภาพ (Closing) กับภาพรอยขีดข่วน	70
3-11 ภาพ QR Code ในส่วนของ V ในแบบจำลอง HSV	71
3-12 การอ้างอิงตำแหน่งเพื่อทำการลบรอยขีดข่วน	71
3-13 ภาพที่ได้หลังจากการลบรอยรอยขีดข่วน	72
3-14 ภาพไบนารีของ QR Code เมื่อผ่านกระบวนการลบรอยขีดข่วน	72
3-15 ผลลัพธ์จากการทำการเปิดภาพ (Opening)	73
3-16 ลักษณะของขอบภาพ QR Code ที่ผ่านการเปิดภาพ (Opening).....	73
3-17 ผลลัพธ์จากการทำการปิดภาพ (Closing).....	74
3-18 ทิศทางการค้นหาหมุม และพื้นที่ที่สนใจ (ROI).....	75
3-19 ผลลัพธ์จากการแบ่งพื้นที่ย่อยบน QR Code	75
3-20 ผลลัพธ์การเติมเต็ม โมดูลย่อยของภาพ QR Code	76
3-21 การเติมเต็มส่วนของ Finder Pattern ที่เสียหายด้วยภาพ Finder Pattern ดั้งเดิม	77
3-22 ภาพ QR Code ที่ผ่านการเติมเต็มในส่วน Finder Pattern	77
3-23 การเติมเต็ม Timing Pattern โดยใช้ภาพต้นแบบ	78
3-24 การเพิ่มพื้นที่ Quiet Zone ให้กับภาพ QR Code	78
3-25 กระบวนการถอดรหัสภาพ QR Code.....	79
4-1 แผนภาพค่า NC เฉลี่ยในรอยขีดข่วนขนาด 0.4 mm	90
4-2 แผนภาพค่าเวลาที่ใช้ในการประมวลผลเฉลี่ยในรอยขีดข่วนขนาด 0.4 mm.....	91
4-3 แผนภาพค่าเวลาที่ใช้ในการถอดรหัสเฉลี่ยในรอยขีดข่วนขนาด 0.4 mm.....	91
4-4 แผนภาพค่า NC เฉลี่ยในรอยขีดข่วนขนาด 2 mm	92
4-5 แผนภาพค่าเวลาที่ใช้ในการประมวลผลเฉลี่ยในรอยขีดข่วนขนาด 2 mm.....	93
4-6 แผนภาพค่าเวลาที่ใช้ในการถอดรหัสเฉลี่ยในรอยขีดข่วนขนาด 2 mm.....	93
4-7 แผนภาพค่า NC เฉลี่ยในรอยขีดข่วนขนาด 3.5 mm	94

ภาพที่	หน้า
4-8 แผนภาพค่าเวลาที่ใช้ในการประมวลผลเฉลี่ยในรอยขีดข่วนขนาด 3.5 mm.....	95
4-9 แผนภาพค่าเวลาที่ใช้ในการถอดรหัสเฉลี่ยในรอยขีดข่วนขนาด 3.5 mm.....	95
4-10 แผนภาพค่า NC เฉลี่ยในรอยขีดข่วนขนาด 5 mm.....	96
4-11 แผนภาพค่าเวลาที่ใช้ในการประมวลผลเฉลี่ยในรอยขีดข่วนขนาด 5 mm.....	97
4-12 แผนภาพค่าเวลาที่ใช้ในการถอดรหัสเฉลี่ยในรอยขีดข่วนขนาด 5 mm.....	97

บทที่ 1

ส่วนประกอบของวิทยานิพนธ์

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันนี้บาร์โค้ดเป็นสิ่งที่ทุกคนคุ้นเคยซึ่งพบเห็นได้ในชีวิตประจำวัน [1] ซึ่งเทคโนโลยีบาร์โค้ดได้เข้ามามีส่วนช่วยในการอำนวยความสะดวกในงานต่างๆ อาทิ ทางด้านการทำธุรกิจการค้าขาย โดยการนำบาร์โค้ดมาติดกับบรรจุภัณฑ์ต่างๆ ของสินค้า ซึ่งในตัวบาร์โค้ดจะมีการบรรจุข้อมูลรายละเอียดที่สำคัญของสินค้า ได้แก่ รหัสผลิตภัณฑ์ ชื่อผลิตภัณฑ์ และราคาของผลิตภัณฑ์ ทั้งนี้เพื่อช่วยในกระบวนการสืบค้นข้อมูลในการจัดการสต็อก ทำให้ทราบยอดจำนวนคงเหลือและจำนวนผลิตภัณฑ์ที่ถูกจำหน่ายได้อย่างรวดเร็ว และแม่นยำ ซึ่งบาร์โค้ด 1 มิติ มีลักษณะเป็นแถบประกอบด้วยเส้นสีดำและเส้นสีขาวสลับกัน บาร์โค้ด 1 มิติ ที่พบเห็นได้ทั่วไปตามสินค้าในห้างสรรพสินค้า [2] ได้แก่ UPC EAN-1 หรือ ISBN โดยบาร์โค้ดเหล่านี้บรรจุข้อมูลได้น้อยซึ่งบรรจุได้เพียง 20 ตัวอักษร ในระยะเวลาต่อมาบาร์โค้ด 2 มิติ ได้รับการพัฒนาขึ้นเพื่อแก้ปัญหาความจุข้อมูลในบาร์โค้ด 1 มิติ ในบาร์โค้ด 2 มิติ ได้ถูกออกแบบให้สามารถบรรจุได้ทั้งแนวตั้งและแนวนอน ทำให้สามารถเก็บข้อมูลได้ถึง 4000 ตัวอักษร รองรับชนิดข้อมูลที่หลากหลายขึ้นทำให้สามารถจัดเก็บข้อมูลที่เป็นภาษาญี่ปุ่น เกาหลี จีน ไทย เป็นต้น อีกทั้งบาร์โค้ด 2 มิติ ยังมีคุณสมบัติในการเรียกคืนข้อมูลที่เสียหายได้ เรียกว่า Error Correcting ช่วยทำให้ลดข้อผิดพลาดที่เกิดจากกระบวนการถอดรหัส บาร์โค้ด 2 มิติ มีหลากหลายรูปแบบ เช่น PDF417 Data Matrix Maxi Code และ QR Code คิวอาร์โค้ด (QR Code) ย่อมาจาก Quick Response Code เป็นบาร์โค้ด 2 มิติที่ได้รับการยอมรับในระดับสากลดังที่เห็นได้จากการมีมาตรฐานมารองรับในปัจจุบัน [3] QR Code ถูกสร้างและพัฒนาเรื่อยมาโดยบริษัท DensoWave [4] ครั้งแรกถูกพัฒนาเพื่อให้เป็นสัญลักษณ์ที่ง่ายต่อการตีความหมายด้วยเครื่องสแกนเนอร์ และในปัจจุบันสามารถทำการอ่าน QR Code ได้จากซอฟต์แวร์ที่มีติดตั้งอยู่ในมือถือทั่วไป QR Code ได้รับความนิยมน้อยกว่าหลายทั้งในอุตสาหกรรมการผลิตอุปกรณ์ชิ้นส่วนอุปกรณ์อิเล็กทรอนิกส์หรือตามผลิตภัณฑ์ต่างๆ นอกจากนี้ QR Code ยังถูกนำไปใช้ในการประชาสัมพันธ์ต่างๆ เพื่ออำนวยความสะดวกให้กับลูกค้า QR Code ก็ได้ถูกนำไปใช้ที่หลากหลายในงานต่างๆ เช่น การนำระบบที่มี QR Code มาช่วยในการค้นหาร้านค้าต่างๆ ในห้างสรรพสินค้า การนำไปประยุกต์ใช้งานใน Social Network [5] ทั้งนี้เนื่องด้วย QR Code มีประโยชน์อย่างมากทั้งในด้านความจุของข้อมูล และประเภทของข้อมูลที่หลากหลาย เช่น ข้อมูลที่เป็นตัวเลข ตัวอักษร สัญลักษณ์ต่างๆ ตัวอักษร Kanji Kana และ Control Code ด้วยความสามารถในการจัดเก็บข้อมูลได้มากกว่าบาร์โค้ดแบบหนึ่งมิติหลายเท่าตัว ความเร็วที่มากกว่าในการสแกน

QR Code เพื่อทำการถอดรหัส และ QR Code ยังมีส่วน Error Correcting [3] ที่ช่วยในกระบวนการกู้คืนข้อมูลที่เสียหายได้ตามเวอร์ชันของ QR Code อีกด้วย ถึงแม้ว่า QR Code จะมีการออกแบบมาเพื่อการป้องกันความเสียหายซึ่งตัว QR Code นั้นจะมีขอบกรอบเมื่อองค์ประกอบสำคัญที่ใช้ในกระบวนการถอดรหัสถูกรบกวนโดยรอยขีดข่วนที่มีขนาดรอยขีดข่วน และความเข้มสีใกล้เคียงกับสีของโมดูลต่างๆ ของ QR Code ทำให้ซอฟต์แวร์ที่ใช้ในการถอดรหัสไม่สามารถทำการแยกแยะโมดูลสีขาวและสีดำได้ กระบวนการค้นหา QR Code ล้มเหลว จึงส่งผลให้กระบวนการในการถอดรหัสไม่สามารถทำได้ งานวิจัยนี้นำเสนอการศึกษาเกี่ยวกับกระบวนการขจัดรอยขีดข่วนบนภาพ QR Code โดยที่มีการนำเทคนิคการประมวลผลภาพ (Digital Image Processing) ซึ่งมีขั้นตอนประกอบไปด้วยการแบ่งแยกภาพโดยใช้แบบจำลองสี HSV (Color Segmentation using HSV Model) เพื่อทำการแยกความต่างของสี QR Code กับรอยขีดข่วนแล้ว ทำการนำภาพรอยที่ได้เข้าสู่อการทำ Binarization ซึ่งเป็นขั้นตอนการแปลงภาพรอยให้เป็นภาพไบนารี (Binary Image), การทำ Morphological Image Processing โดยใช้เทคนิค Opening ร่วมกับ Closing เพื่อปรับปรุงโครงสร้างของภาพรอยที่ได้ให้ มีความสมบูรณ์มากยิ่งขึ้นแล้วนำตำแหน่งรอยที่ได้ไปใช้ในการลบรอยออกจากภาพที่มีรอยขีดข่วน จากนั้นเข้าสู่ขั้นตอนการกำหนดพื้นที่ที่สนใจ (Region of Interest :ROI) เพื่อให้ได้ QR Code ที่ไม่มี Quiet Zone แล้วทำการวาดเส้นตาราง (Grids Generation) เพื่อกำหนดพื้นที่ย่อยของ QR Code โดยใช้ขนาดของโมดูลของ QR Code ที่ใช้ในการทดลอง เมื่อได้พื้นที่ของโมดูลย่อยแล้ว ทำการวิเคราะห์กลุ่มสมาชิกของพิกเซลเพื่อทำการเติมเต็มส่วน QR Code ที่เสียหายในแต่ละโมดูล (Reconstruction QR Code) เป็นขั้นตอนสุดท้าย เมื่อผ่านกระบวนการต่างๆ ภาพผลลัพธ์ของ QR Code จะสามารถนำไปอ่านเพื่อถอดรหัสโดยใช้ซอฟต์แวร์ที่มีอยู่ได้อย่างมีประสิทธิภาพ

1.2 วัตถุประสงค์ของการวิจัย

เพื่อศึกษาออกแบบและนำเสนอวิธีการที่เหมาะสมในการปรับปรุงภาพ QR Code ที่มีความเสียหายจากรอยขีดข่วน โดยการใช้วิธีการประมวลผลภาพ เพื่อให้ได้ภาพ QR Code ที่สามารถทำการถอดรหัสได้

1.3 ขอบเขตของการวิจัย

1. เป็นการปรับปรุงภาพ QR Code ที่เสียหายจากรอยขีดข่วนซึ่ง QR Code Reader ทัวไปไม่สามารถอ่านได้

2. หาวิธีการประมวลผลภาพหลาย ๆ วิธีเพื่อให้ได้ QR Code ผลลัพธ์ที่สามารถนำไปอ่านเพื่อถอดรหัสได้
3. ภาพ QR Code ที่ใช้ในการทดลองต้องเป็นภาพถ่ายที่ได้จากมุมตรง และสภาพแสงสม่ำเสมอ
4. การถอดรหัสจะใช้ Library ZXing version 2.0
5. QR Code ที่ใช้ในการทดลองคือ QR Code Version 2 ใน Level L
6. เกณฑ์การวัดคุณภาพที่ใช้มี 2 แบบ คือ
 - a. เกณฑ์การวัดประสิทธิภาพของการคำนวณของเครื่องคอมพิวเตอร์โดยวัดเวลาการทำงานของกระบวนการประมวลผลของ Clock cycle
 - b. การใช้เกณฑ์วัดคุณภาพของภาพด้วยค่า NC (Normal Correlation)
7. ขั้นตอนการประมวลผลภาพ QR Code ในงานวิจัยนี้ใช้ C++ ร่วมกับ OpenCV Library version 4.2.2
8. ขั้นตอนในการถอดรหัสภาพ QR Code ในงานวิจัยนี้ใช้ Zxing เวอร์ชัน 2.0

1.4 ข้อตกลงเบื้องต้น

1. ภาพต้นฉบับ (Original Image) QR Code หมายถึง ภาพหลักที่มีความสมบูรณ์ สกุล PNG มีขนาดกว้าง 216 พิกเซล และสูง 216 พิกเซล
2. ภาพที่นำเข้า (Input Image) หมายถึง ภาพ QR Code ที่มีรอยขีดข่วน สกุล JPEG มีขนาดกว้าง 378 พิกเซล และสูง 378 พิกเซล
3. รอยขีดข่วน หมายถึง รอยที่เกิดจากการใช้ปากกาถูกลิ้น ปากกาเคมี ซึ่งมีขนาดเส้นตั้งแต่ 0.4 mm – 5 mm ซึ่งรอยขีดข่วนนั้นต้องไม่เป็นสีดำ แต่เป็นสีที่มีค่าความอิ่มตัวของสี (Saturation) ตั้งแต่ 60 % - 100% ในสถานะแสงที่เอื้ออำนวย สีของรอยขีดข่วน เช่น สีน้ำเงินเข้ม เขียวเข้ม แดงเข้ม เป็นต้น โดยที่ปริมาณของรอยขีดข่วนต่อพื้นที่ QR Code ที่มากกว่า Error Correction ของ QR Code ในเวอร์ชันที่ใช้ในการทดลอง หรือปริมาณของรอยขีดที่มีปริมาณน้อยกว่าระดับ Error Correction ของ QR Code ในเวอร์ชันที่ใช้ในการทดลอง แต่รอยขีดข่วนนั้นรบกวน

ในองค์ประกอบสำคัญของ QR Code ซึ่งส่งผลให้ QR Code นั้นๆ เกิดความเสียหายไม่สามารถทำการถอดรหัสโดยใช้ QR Code Reader [12] ได้ โดยที่ QR Code ที่ใช้จะต้องเป็น QR Code ที่มีสีดำและอยู่บนพื้นหลังที่เป็นสีขาว

4. เวลาในการประมวลผล (Process Time) หมายถึง ระยะเวลาระหว่างการนำเข้าภาพถึงขั้นตอนสุดท้ายที่ได้ผลลัพธ์ที่พร้อมสำหรับการถอดรหัส
5. เวลาในการถอดรหัส (Decoding Time) หมายถึง เวลาที่ถอดรหัส QR Code โดยใช้ Zxing Library

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. เพื่อให้ได้กระบวนการในการจัดรอยขีดข่วนบนภาพ QR Code
2. เป็นแนวทางการวิจัยด้านการพัฒนากระบวนการที่ช่วยให้สามารถอ่านเพื่อถอดรหัส QR Code ที่เสียหายจากรอยขีดข่วนตามข้อตกลงเบื้องต้นได้
3. เป็นแนวทางพัฒนาร่วมกับบาร์โค้ด (Barcode) รูปแบบอื่นๆ
4. เพื่อเป็นแนวทางในการพัฒนา Application ที่ใช้ในโทรศัพท์มือถือต่อไป

1.6 ลำดับขั้นตอนในการเสนอผลการวิจัย

1. ศึกษาทฤษฎี หลักการพื้นฐานที่ใช้ในการวิจัย และงานวิจัยที่เกี่ยวข้อง
2. ศึกษาเทคนิคต่างๆ ที่มีอยู่ถึงแนวคิดหลักการ ข้อดีและข้อบกพร่องของแต่ละเทคนิคที่อยู่ในขอบเขต
3. ออกแบบและนำเสนอวิธีการในการจัดรอยขีดข่วนบนภาพคิวอาร์โค้ด
4. ทดสอบและปรับปรุงวิธีการที่นำเสนอ
5. วิเคราะห์ผลการทดลอง
6. สรุปผลและเรียบเรียงวิทยานิพนธ์

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในส่วนนี้เป็นแนวคิดและทฤษฎีที่เกี่ยวกับ QR Code และการประมวลผลภาพดิจิทัล ซึ่งจะกล่าวถึงข้อมูลทั่วไปเกี่ยวกับบาร์โค้ด โครงสร้างส่วนประกอบของ QR Code และขั้นตอนการถอดรหัสของภาพ QR Code พื้นฐานการประมวลผลภาพดิจิทัล ข้อมูลสีในแบบจำลองสี RGB และแบบจำลองสี HSV การแปลงแบบจำลองสี RGB ไปยัง HSV การประมวลผลภาพกับรูปร่าง โครงสร้างภาพ (Morphological Image Processing) การประมวลผลภาพไบนารี (Binary image) และการปรับเปลี่ยนรูปร่างของภาพ ตามลำดับ

2.1 ข้อมูลทั่วไปเกี่ยวกับบาร์โค้ด

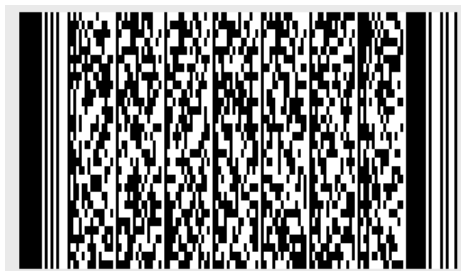


ภาพที่ 2-1 บาร์โค้ดแบบ 1 มิติ

ทุกวันนี้มีการใช้งานบาร์โค้ดอย่างแพร่หลายหลายในธุรกิจต่างๆ อาทิ เช่น การขนส่งพัสดุ การจัดการคลังสินค้า การใช้ในตัวโดยสารของสายการบินต่างๆ [6] และที่พบมากที่สุดคือในบรรจุภัณฑ์ของสินค้าในชีวิตประจำวัน โดยทั่วไปบาร์โค้ดที่ติดอยู่บนผลิตภัณฑ์เหล่านี้จะมีการเก็บเพียงรหัสสินค้านั้นๆ ซึ่งเมื่อลูกค้าทำการชำระเงิน พนักงานขายจะทำการสแกนบาร์โค้ดโดยใช้เครื่องอ่านบาร์โค้ด ซึ่งเครื่องอ่านบาร์โค้ดจะส่งรหัสบาร์โค้ดที่สแกนได้ไปยังเครื่องคอมพิวเตอร์แม่ข่ายเพื่อประมวลผลร่วมกับฐานข้อมูลของสินค้าชิ้นนั้น จากนั้นระบบจะแสดงราคาให้พนักงานขายทราบ จะเห็นได้ว่าพนักงานขายไม่จำเป็นต้องพิมพ์รหัสสินค้าเพื่อสืบค้นราคาแต่ใช้เพียงการสแกนบาร์โค้ด ซึ่งบาร์โค้ดช่วยประหยัดเวลาในการทำรายการอย่างมาก โดยที่บาร์โค้ดที่เราพบเห็นทั่วไปในบรรจุภัณฑ์ของสินค้าในร้านขายของชำหรือห้างสรรพสินค้าจะเป็นบาร์โค้ดที่เป็นแถบเส้นสีขาวสีดำสลับกัน เราเรียกว่า บาร์โค้ด 1 มิติ (One Dimension Barcode) ดังภาพ 2-1 ซึ่งบาร์โค้ด 1 มิติ มีหลายประเภทแต่ที่พบมากในปัจจุบัน อาทิเช่น UPC (Universal Product Code), EAN (European Article Number) และ ISBN (International Standard Book Number) บาร์โค้ด ISBN จะพบอย่างแพร่หลายบนปก หรือบนสันของหนังสือทุกๆ เล่มที่มีอยู่ในท้องตลาด ในบาร์โค้ด 1 มิติสามารถเก็บข้อมูลได้ประมาณ 13 อักขระซึ่งพอสำหรับการใช้ในการเก็บรหัสของสินค้าเท่านั้น

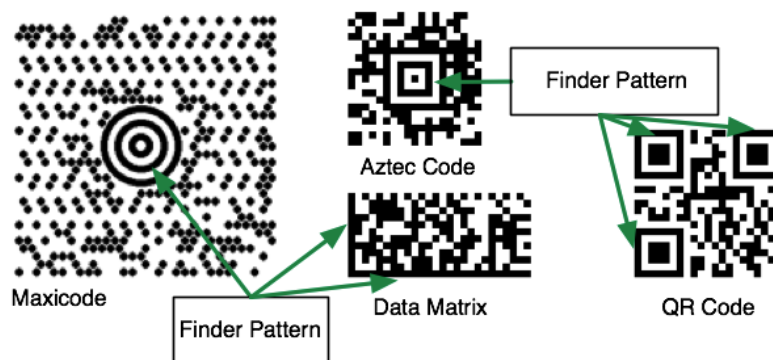
เมื่อความจุที่ค่อนข้างจำกัด ในการใช้งานจำเป็นต้องมีการใช้งานร่วมกับฐานข้อมูลที่มีอยู่ในเครื่องแม่ข่ายในกรณีที่ต้องการข้อมูลทีนอกเหนือจากที่มีอยู่ในบาร์โค้ด เพื่อแก้ปัญหาที่พบในบาร์โค้ด 1 มิติ จึงได้มีการพัฒนาบาร์โค้ด 2 มิติ (Two Dimension Barcode) ขึ้น โดยมีการออกแบบเป็นเมทริกซ์ที่สามารถแทนการเก็บข้อมูลทั้งแนวตั้งและแนวนอนทำให้ความจุในการเก็บข้อมูลเพิ่มขึ้น บาร์โค้ด 2 มิติสามารถแบ่งได้ 2 ประเภทดังต่อไปนี้

1. บาร์โค้ดประเภทสแต็ก (Stacked Barcode) จะมีลักษณะเป็นการนำบาร์โค้ด 1 มิติหลายๆ แถวมาวางซ้อนต่อกัน ในกระบวนการถอดรหัสขึ้นเมื่อมีการสแกนบาร์โค้ดจะมีการขยายปรับความกว้างของบาร์โค้ดก่อนการถอดรหัส ทิศทางการอ่านบาร์โค้ดแบบสแต็กนี้จะทำได้เพียงทิศทางเดียวเท่านั้นทิศทางในการอ่านจะต้องเป็นไปตามทิศทางดังต่อไปนี้ จากทางขวาไปทางซ้ายทางซ้ายไปทางขวา จากล่างขึ้นบน และจากบนลงล่าง ตัวอย่างบาร์โค้ดแบบสแต็กได้แก่ PDF417 (Portable Data File) ดังภาพ 2-2



ภาพที่ 2-2 ตัวอย่างบาร์โค้ดแบบสแต็ก PDF417

2. บาร์โค้ดประเภทเมตริกซ์ (Matrix Barcode) จะมีลักษณะที่แตกต่างจากบาร์โค้ดแบบสแต็กอย่างมากในบาร์โค้ดประเภทนี้จะมีลักษณะเด่นคือจะมีองค์ประกอบที่ช่วยในการค้นหาบาร์โค้ดเรียกว่า Finder Pattern ดังภาพ 2-3 หน้าที่สำคัญของ Finder Pattern คือ ช่วยทำให้การอ่านบาร์โค้ดทำได้อย่างรวดเร็วและช่วยเพิ่มโอกาสในการอ่านสำเร็จแม้บาร์โค้ดจะตั้งอยู่ในตำแหน่งที่กลับด้าน เอียง หรือหมุนไปในทิศต่างๆ นอกจากนี้ในบาร์โค้ดประเภทเมตริกซ์นี้จะใช้พื้นที่น้อยกว่าบาร์โค้ดแบบสแต็กจึงได้ถูกนำไปใช้ในพื้นที่จำกัด [1] ตัวอย่างบาร์โค้ดประเภทเมตริกซ์นี้ได้แก่ Maxicode, Data Matrix, Aztec Code, QR Code เป็นต้น



ภาพที่ 2-3 ตัวอย่างบาร์โค้ด 2 มิติ และตำแหน่งของ Finder Pattern

ประโยชน์ของบาร์โค้ด 2 มิติ เนื่องจากความจุในการบรรจุข้อมูลที่มีมากขึ้น ความเร็วในการสแกนที่รวดเร็ว จึงได้มีการใช้งานอย่างแพร่หลาย ตัวอย่างการใช้งานบาร์โค้ดสองมิตินี้มีดังต่อไปนี้

1. ใช้ในด้านอุตสาหกรรมการผลิตรถยนต์ โดยการนำบาร์โค้ดมาติดตามอะไหล่ เพื่อเก็บข้อมูลของอะไหล่ชิ้นนั้น ชื่อ รหัสชิ้นส่วน ประเภทของอะไหล่
2. ใช้งานในด้านการผลิตอุปกรณ์อิเล็กทรอนิกส์ โดยมีการนำบาร์โค้ดมาติดบนชิป (Chip) บนแผงวงจรอิเล็กทรอนิกส์เพื่อเก็บรายละเอียดขององค์ประกอบต่างๆ ที่มีอยู่ในแผงวงจรนั้นๆ
3. ใช้งานในด้านการขนส่งสินค้า บาร์โค้ด 2 มิติถูกนำมาใช้อย่างแพร่หลายในกล่องพัสดุ และบนใบรับส่งสินค้าโดยที่บาร์โค้ดจะมีการบรรจุข้อมูลผู้รับ ผู้ส่ง วันเวลาที่ส่ง จำนวน ของสินค้า เจ้าหน้าที่ที่สามารถดูข้อมูลได้ง่ายโดยการสแกนที่บาร์โค้ด
4. ใช้ในด้านการแพทย์ ในประเทศญี่ปุ่นได้มีการนำบาร์โค้ดมาทำแท็กติดข้อมือผู้ป่วย เพื่อบ่งบอกถึงชื่อผู้ป่วย ที่อยู่ ประวัติของผู้ป่วย อาการของผู้ป่วย ยาที่ผู้ป่วยแพ้ นอกจากนี้ในประเทศออสเตรเลียได้มีการนำบาร์โค้ดมาใช้ในหลอดทดลอง ที่ใช้ในการทดสอบเลือดและสารต่างๆในห้องทดลองอีกด้วย
5. ใช้ในด้านการปศุสัตว์ มีการนำบาร์โค้ดไปติดที่ป้ายติดหูสัตว์ (Ear Tag) หรือสายรัดที่หางของสัตว์เลี้ยง (Tail Tag) เพื่อใช้ในการตรวจสอบเลขทะเบียนประจำตัวอายุ และสายพันธุ์ ซึ่งมีการใช้งานอย่างแพร่หลายในประเทศออสเตรเลีย

6. ใช้ในระบบการจองตั๋วต่างๆ เมื่อมีการทำรายการจองตั๋ว ผู้จองทำรายการผ่านทางแอปพลิเคชันบนโทรศัพท์มือถือ จากนั้นเมื่อเสร็จสิ้นการทำรายการผู้จองจะได้รับบาร์โค้ดที่สามารถนำไปใช้ในการเข้าร่วมงานได้ บริการที่เห็นได้ในปัจจุบันได้แก่ บัตรการแสดงคอนเสิร์ต บัตรโดยสารรถไฟ เป็นต้น
7. ใช้บนนามบัตร เนื่องด้วยบาร์โค้ด 2 มิติ มีความจุที่มากเพียงพอที่จะเก็บข้อมูลของรายชื่อโทรศัพท์ได้ เมื่อนำบาร์โค้ดไปติดลงบนนามบัตรทำให้ผู้รับสามารถที่สแกนผ่านโทรศัพท์มือถือ และเมื่อแสกนเรียบบร้อยข้อมูลการติดต่อนั้นจะจัดเก็บลงในสมุดรายชื่อในโทรศัพท์ทันที ช่วยลดข้อผิดพลาดในการป้อนหมายเลขโทรศัพท์และข้อมูลการติดต่อของคนสำคัญได้
8. ใช้ในการประชาสัมพันธ์ โฆษณาต่างๆ เช่น ในประเทศเกาหลี ได้หวัน ญี่ปุ่น และในประเทศไทย ได้มีการนำบาร์โค้ด 2 มิติมาใช้ในการประชาสัมพันธ์ ทั้งในหนังสือพิมพ์ โปสเตอร์ หรือป้ายโฆษณาขนาดใหญ่ โดยส่วนใหญ่แล้วข้อมูลที่เก็บในบาร์โค้ดจะเป็นข้อมูล URL ของผลิตภัณฑ์นั้นๆ
9. ด้านสินค้าในชีวิตประจำวัน พบเห็นได้ตามสินค้าที่มีอยู่ในซูเปอร์มาเก็ตต่างๆ โดยบาร์โค้ด 2 มิติ ได้ถูกนำมาใช้ในการเก็บข้อมูลโภชนาการที่มีในอาหารนั้นๆ

นอกเหนือจากตัวอย่างที่ได้กล่าวมาข้างต้น บาร์โค้ด 2 มิติ ยังมีประโยชน์อีกมากมาย ตัวอย่างของการใช้งานบาร์โค้ด 2 มิติ ดังภาพ 2-4



ภาพที่ 2-4 ตัวอย่างการใช้งานบาร์โค้ด 2 มิติ

2.2 คิวอาร์โค้ด (QR Code)

QR Code ย่อมาจากคำว่า Quick Response Code เป็นหนึ่งในบาร์โค้ด 2 มิติ ได้ถูกคิดค้นโดยบริษัท Denso Wave ในปี 1994 แต่เดิมได้ถูกออกแบบมาเพื่อใช้ในอุตสาหกรรมการผลิตยานยนต์ (Automotive industry) ในประเทศญี่ปุ่น ใช้ในการติดตามหมายเลขชิ้นส่วนต่างๆ ของอะไหล่ยานยนต์ ต่อมา QR Code เริ่มมานิยมใช้ในงานอื่นๆ เนื่องด้วยความสามารถที่โดดเด่นของ QR Code คือ อ่านได้รวดเร็ว และความจุที่สูงกว่าบาร์โค้ดที่มีขนาดเท่าๆ กัน หรือความจุที่มากกว่าบาร์โค้ด 1 มิติหลายเท่าตัว QR Code ได้ถูกนำมาใช้อย่างแพร่หลายในงานประชาสัมพันธ์ผลิตภัณฑ์ต่างๆ สมาร์ทโฟนในปัจจุบันมักจะมาพร้อมกับซอฟต์แวร์ที่สามารถถอดรหัส QR Code แต่ถึงแม้ไม่มีมาให้ก็สามารถติดตั้งซอฟต์แวร์เพื่อทำการสแกน QR Code ได้ การอ่าน QR Code ทำได้โดยการใช้กล้องที่มีกับโทรศัพท์ถ่ายภาพ QR Code จากนั้นซอฟต์แวร์จะทำการถอดรหัสและแสดงให้ผู้ใช้งานทราบผลลัพธ์ทางหน้าจอโทรศัพท์ทันที ดังภาพ 2-5



ภาพที่ 2-5 การถอดรหัส QR Code โดยใช้โทรศัพท์มือถือ



ภาพที่ 2-6 ตัวอย่างการใช้ QR Code ในบริการ Favorite Place

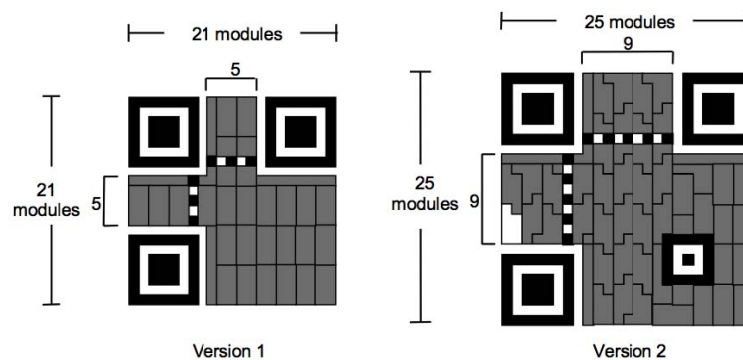
ไม่ใช่เพียงในเอเชียเท่านั้นที่มีความนิยมใน QR Code ในหลายประเทศในทวีปยุโรป [7] ได้มีการใช้ QR Code อย่างแพร่หลายเช่นกัน ข้อมูลที่บันทึกอยู่ใน QR Code ส่วนมากจะเป็น URL (Uniform Resource Locator) ซึ่งเป็นที่อยู่ของเว็บเพจที่มีข้อมูลอยู่ เช่น ข้อมูลโฆษณาการของผลิตภัณฑ์ ข้อมูลทั่วไป หรือในบางกรณีอาจจะเล่นเพลงหรือวิดีโอ เพื่อประกอบการตัดสินใจในการซื้อสินค้า เป็นต้น Google ได้มีการนำ QR Code มาใช้ในบริการที่มีชื่อว่า Favorite Place [1] ซึ่งบริการนี้ร้านค้าที่ร่วมรายการจะนำสติ๊กเกอร์ QR Code ดังภาพ 2-6 ซึ่งภายใน QR Code

จะบรรจุ URL ที่ลิงก์ไปยังข้อมูลต่างๆ เช่น คำวิจารณ์คำติชม คำแนะนำจากลูกค้าคนอื่นๆ เคยใช้บริการจากร้านแห่งนี้ ทำให้ผู้ที่ใช้บริการได้รับข้อมูลที่เป็นประโยชน์ต่อการตัดสินใจต่อไป สำหรับ QR Code กับระบบปฏิบัติการบนโทรศัพท์มือถือ [7] อย่างที่กล่าวมาบางส่วนว่า QR Code สามารถสแกนผ่านสมาร์ตโฟน ซึ่งในปัจจุบันนี้สมาร์ตโฟนมีหลากหลายระบบปฏิบัติการ อาทิ ระบบปฏิบัติการ Google's Android และระบบปฏิบัติการ iOS ของ iPhone, iPod และ iPad จำเป็นต้องมีการติดตั้งซอฟต์แวร์ที่ใช้ในการอ่านเพื่อถอดรหัส ซึ่งหาได้ใน PlayStore สำหรับ Google's Android ส่วนระบบปฏิบัติการ iOS สามารถดาวน์โหลดได้ที่ AppStore ด้วยการใช้จ่าย QR Code Scanner นอกเหนือจาก ทั้งสองระบบปฏิบัติการแล้ว Nintendo 3DS เครื่องเล่นเกมสก็ที่ได้รับความนิยมก็ได้มีการนำ QR Code มาใช้ในการดำเนินเรื่องของเกมส์อีกด้วย BlackBerry เป็นระบบปฏิบัติการที่ได้ทำการติดตั้งความสามารถในการสแกนพร้อมทั้งสามารถสร้าง QR Code มาให้พร้อมใช้งานโดยที่ใน BlackBerry จะใช้ QR Code ในการแลกเปลี่ยนหมายเลข Pin ซึ่งเป็นหมายเลขประจำตัวที่ใช้ในโปรแกรมสนทนาที่ชื่อว่า BBM (BlackBerry Messenger) ส่วนใน Windows Phone 7.5 ก็ได้มีการติดตั้งความสามารถพร้อมใช้งานเช่นกัน โดยที่สามารถทำการสแกนภาพ QR Code ผ่านทางโปรแกรมที่ใช้ในการค้นหาที่มีชื่อว่า Bing จะเห็นได้ว่าแทบจะทุกระบบปฏิบัติการที่ใช้อยู่สมาร์ตโฟนในปัจจุบันนี้สามารถสแกน QR Code จึงเป็นสาเหตุหนึ่งที่ทำให้ QR Code เป็นที่นิยมในกลุ่มผู้ใช้สมาร์ตโฟน จากภาพที่ 2-5 และ 2-6 จะเห็นได้ว่า QR Code ซึ่งเป็นบาร์โค้ด 2 มิติสามารถอ่านข้อมูลผ่านโทรศัพท์มือถือได้อย่างรวดเร็ว QR Code ยังมีความสามารถในการเก็บข้อมูลได้หลากหลายรูปแบบ ได้แก่ ข้อมูลตัวเลข (Numeric), ตัวอักษร (Alphanumeric), ตัวอักษร Kana, Kanji, สัญลักษณ์ (Symbols) และข้อมูลประเภทไบนารี (Binary) และ Control Code จากมาตรฐาน QR Code [3] ได้ระบุความจุของ QR Code ไว้ดังตารางต่อไปนี้

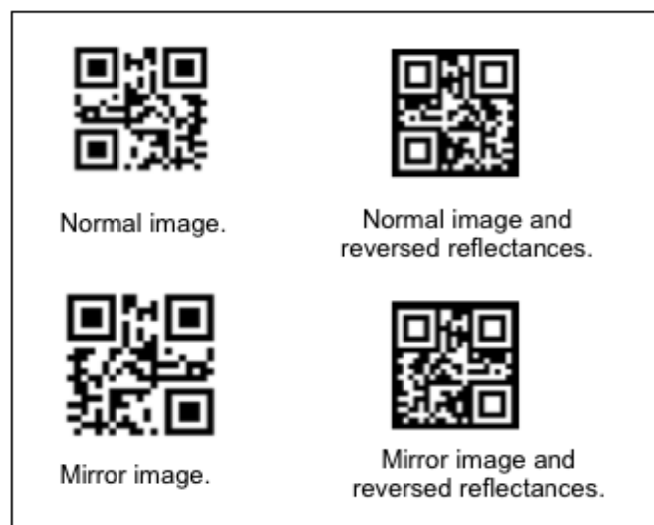
ตารางที่ 2-1 แสดงความจุของ QR Code แบ่งตามประเภทข้อมูล

ประเภทข้อมูล	จำนวนข้อมูลสูงสุด
Numeric (0-9)	7,089 characters
Alphanumeric (0-9, A-Z,\$%*+/-/)	4,296 characters
8Bit byte (Binary)	2,953 bytes
Kanji, Kana	1,817 characters

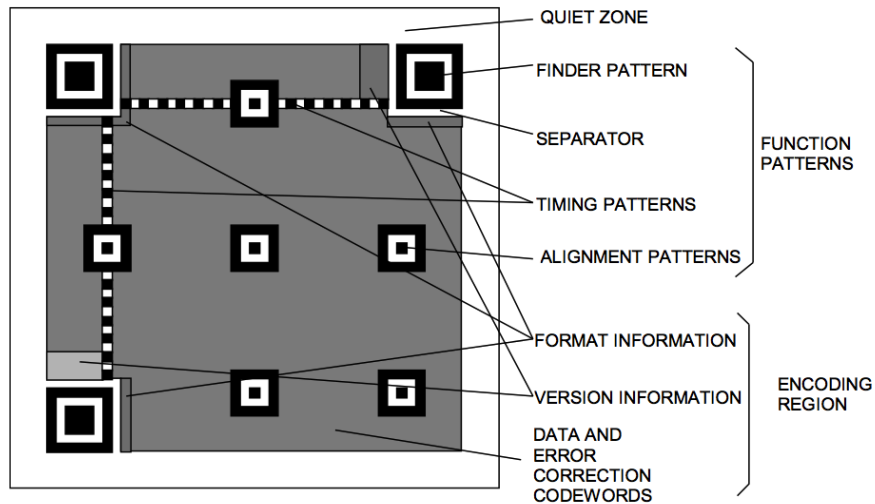
QR Code จะมีการแทนความหมายของสัญลักษณ์ไบนารี โดยที่โมดูลสีดำจะแทนที่ด้วย 1 และขาวจะถูกแทนที่ด้วย 0 สำหรับจำนวนของสัญลักษณ์ของ QR Code นั้นจะขึ้นอยู่กับเวอร์ชัน โดยที่จะเริ่มที่ 21 x 21 โมดูล ถึง 177 x 177 โมดูล ในเวอร์ชัน 1 ถึง 40 ซึ่งจะมีการเพิ่มทีละ 4 โมดูล เมื่อมีเวอร์ชันที่สูงขึ้น ดังนั้นใน QR Code ในเวอร์ชัน 40 ซึ่งมีทั้งหมด 177 x 177 โมดูล ซึ่งเกิดจาก $\{(39*4)+21\}*\{(39*4)+21\}$ [8] ดังภาพ 2-7 ใน QR Code ยังมีคุณสมบัติการสะท้อนกลับ (Reflectance reversal) ใน QR Code โมดูลสีดำจะแทนด้วย 0 และโมดูลขาวจะแทนด้วย 1 นอกจากนี้ QR Code ยังมีคุณสมบัติ Mirror Imaging ซึ่งใน International Standard [3] ได้ระบุตำแหน่งของสัญลักษณ์ของ QR Code ได้อย่างชัดเจน อาทิ ตำแหน่งของ Finder Pattern แต่อย่างไรก็ตามเมื่อมองในมุมกลับกันทุกๆ โมดูลของ QR Code จะสลับจากซ้ายไปขวาซึ่งมุมมองดังกล่าวซอฟต์แวร์ที่ใช้ในการถอดรหัสจะสามารถทำการถอดรหัสได้ ดังภาพ 2-8



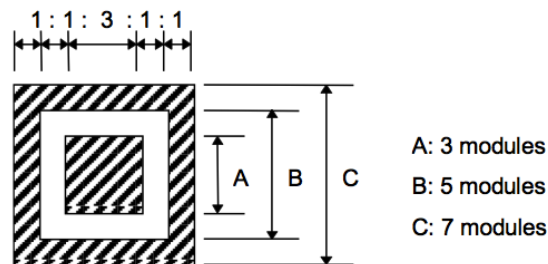
ภาพที่ 2-7 จำนวนของโมดูลใน QR Code Version 1 และ Version 2



ภาพที่ 2-8 แสดงมุมมองของ QR Code ที่สามารถถอดรหัสได้



ภาพที่ 2-9 โครงสร้างของ QR Code ตามมาตรฐานสัญลักษณ์ QR Code 2005



ภาพที่ 2-10 โครงสร้างของ Finder Pattern ใน QR Code

จากภาพ 2-9 แสดง QR Code ตามมาตรฐาน [3] ซึ่งจะมีลักษณะเป็นสี่เหลี่ยมจัตุรัสซึ่งในแต่ละโมดูลย่อยก็มีลักษณะที่เป็นสี่เหลี่ยมจัตุรัส โดยที่ประกอบด้วยองค์ประกอบ 2 ส่วนดังนี้

1. Function Pattern จะเป็นองค์ประกอบที่ไม่ได้มีส่วนในการจัดเก็บข้อมูลของ QR Code แต่ Function Pattern จะเป็นองค์ประกอบที่สำคัญที่ใช้ในกระบวนการอ่านเพื่อถอดรหัสภาพ QR Code ซึ่งประกอบด้วย
 - a. Quiet Zone เป็นบริเวณที่เป็นพื้นที่ว่างสีขาวรอบๆ QR Code ซึ่งมีขนาดความกว้าง 4 โมดูล โดยที่ลักษณะสีพื้นของ Quiet Zone จะมีสีเดียวกับโมดูลสีขาวหรือสีว่างของ QR Code ช่วยทำให้ซอฟต์แวร์ที่ใช้ในการถอดรหัสสามารถค้นหา QR Code ได้อย่างรวดเร็วขึ้นซึ่งขนาดของ Quiet Zone ใน QR Code จะมีขนาดเป็น 4 เท่าของโมดูลย่อย

- b. Finder Pattern มีไว้เพื่อใช้ในการตรวจจับตำแหน่งของ QR Code ซึ่งจะแสดงอยู่ที่ มุมบนซ้าย , มุมบนขวา และมุมล่างซ้าย ดังที่แสดงในรูปที่ 2-9 และแต่ละ Finder Pattern จะประกอบไปด้วยโมดูลสีดำ 7x 7 โมดูล ซ้อนทับด้วยโมดูล สีขาว 5x5 และซ้อนทับด้วยโมดูลดำ 3x3 ซึ่งหากพิจารณาตามความกว้างของโมดูลจะได้ 1:1:3:1:1 ซึ่งได้แสดงในภาพที่ 2-10 Finder Pattern มีความสำคัญอย่างมากในการกระบวนการถอดรหัส หากมี Noise ครอบคลุมในส่วนนี้จะทำให้การถอดรหัสล้มเหลว
 - c. Separator เป็นโมดูลที่มีความกว้าง 1 โมดูลซึ่งมีสีขาว จะอยู่บริเวณระหว่าง Finder Pattern และส่วนของ Encoding Region ดังแสดงในภาพ 2-9
 - d. Timing Patterns ประกอบด้วยโมดูลสีดำสลับกับโมดูลสีขาว ซึ่งจะมีการเริ่มและสิ้นสุดที่โมดูลสีดำ โดยเริ่มต้นจากพื้นที่ของ Separator และสิ้นสุดที่ Separator อีกฝั่งซึ่งจากลักษณะของ Timing Pattern ทำให้ทราบถึงขนาดและเวอร์ชัน ของ QR Code ได้ หน้าที่ของ Timing Pattern นั้นช่วยในการตรวจจับพิกัดของสัญลักษณ์เพื่อใช้สำหรับการถอดรหัส
 - e. Alignment Patterns จะมีใน QR Code เวอร์ชันตั้งแต่ 2 เป็นต้นไป ซึ่งแต่ละ Alignment Pattern นั้นจะประกอบไปด้วย โมดูลสีดำขนาด 5x5 ซ้อนทับด้วยโมดูลขาวขนาด 3x3 และซ้อนทับด้วยโมดูลดำขนาด 1x1 ซึ่งจำนวนของ Alignment Pattern จะขึ้นอยู่กับเวอร์ชันของ QR Code หน้าที่ของ Alignment Pattern นั้นจะมีประโยชน์ในภาพที่มีลักษณะเอียงซึ่งช่วยให้สามารถอ่านเพื่อถอดรหัสได้ถูกต้องด้วย Decoder Application
2. Encoding Region จะเป็นองค์ประกอบที่มีความสำคัญกับการเข้ารหัสข้อมูลซึ่งจะแบ่งย่อยได้ดังต่อไปนี้
- a. Format Information มีไว้เพื่อจัดเก็บข้อมูล Data Type , Binary Indicator ของ Error Correction Level และ Data Mask ที่ใช้ใน QR Code นั้นๆ เพื่อใช้ในขั้นตอนการถอดรหัส
 - b. Version Information จะปรากฏในเวอร์ชัน 7 ขึ้นไป ซึ่งเป็นส่วนที่เก็บข้อมูลเวอร์ชันของ QR Code
 - c. Data and Error Correction Codewords ส่วนที่ใช้ในการเก็บข้อมูลของ QR Code ร่วมกับข้อมูลที่ใช้ใน Error Correction ซึ่งเป็นส่วนที่มีพื้นที่มากที่สุดใน QR Code

2.3 กระบวนการเข้ารหัส QR Code

2.3.1 การวิเคราะห์กระแสข้อมูล Input

โดยมีการระบุความแตกต่างของแต่ละตัวอักษรที่จะทำการเข้ารหัส โดยใน QR Code จะมีความสามารถที่ชื่อว่า Extended Channel Interpretation feature ในมาตรฐานของ QR Code 2005 [3] ได้ระบุโหมดที่ใช้ใน QR Code ดังตารางที่ 2-2 ทั้งนี้เพื่อให้ผลในการเข้ารหัสเป็นไปอย่างมีประสิทธิภาพ และเหมาะกับประเภทข้อมูลในขั้นตอนการแปลงจากกระแสข้อมูลเป็นข้อมูลไบนารีจะมีการสลับโหมดหากมีความจำเป็น จากนั้นจำเป็นต้องระบุ Error Detection และ Correction Level ซึ่งถ้าตอนที่สร้างไม่ได้เจาะจงเวอร์ชัน ในขั้นตอนนี้จะเลือกเวอร์ชันที่ต่ำที่สุดที่สอดคล้องกับข้อมูลซึ่งเวอร์ชันและความจุของ QR Code แต่ละเวอร์ชันแสดงรายละเอียดดังตารางที่ 2-3

ตารางที่ 2-2 Mode Indicators ของ QR Code

Mode	QR Code symbols
Numeric	0001
Alphanumeric	0010
Byte	0100
Kanji	1000
Structured Append	0011
Terminator	0000

ตารางที่ 2-3 แสดงเวอร์ชันและความจุของ QR Code ในแต่ละเวอร์ชัน

Version	Number of modules side	Function pattern modules	Format information / version information modules	Data modules except	Data Capacity
1	21	202	31	208	26
2	25	235	31	359	44
3	29	243	31	567	70
4	33	251	31	807	100

5	37	259	31	1079	134
6	41	267	31	1383	172
7	45	390	31	1568	196
8	49	398	67	1936	242
9	53	406	67	2336	292
10	57	414	67	2768	346
11	61	422	67	3232	404
12	65	430	67	3728	466
13	69	438	67	4256	532
14	73	611	67	4651	581
15	77	619	67	5243	655
16	81	627	67	5867	733
17	85	635	67	6523	815
18	89	643	67	7211	901
19	93	651	67	7931	991
20	97	659	67	8683	1085
21	101	882	67	9252	1156
22	104	890	67	10068	1258
23	109	898	67	10916	1364
24	113	906	67	11796	1474
25	117	914	67	12708	1588
26	121	922	67	13652	1706
27	125	930	67	14628	1828
28	129	1203	67	15371	1921
29	133	1211	67	16411	2051
30	137	1219	67	17483	2185
31	141	1227	67	18587	2323
32	145	1235	67	19723	2465
33	149	1243	67	20891	2611

34	153	1251	67	22091	2761
35	157	1574	67	23008	2876
36	161	1582	67	24272	3034
37	165	1590	67	25568	3196
38	168	1598	67	26896	3362
39	173	1606	67	28256	3532
40	177	1614	67	29648	3706

2.3.2 การเข้ารหัสข้อมูล (Data encoding)

ในการเข้ารหัสข้อมูลให้ทำการแปลงข้อมูลที่เป็นตัวอักษรไปเป็น Bit Stream ซึ่งเป็นไปตามข้อบังคับของการแปลงข้อมูล [3] ทำการแทรก Mode indicators ด้านหน้าในทุกๆ Segment จากนั้นทำการจบ Bit stream ด้วย Terminator แล้วทำการแบ่ง Bit stream เป็น Codewords โดยแต่ละชุดของ Codewords จะมีความยาว 8 บิต จากนั้นทำการเพิ่ม Codewords ให้ครบตามเวอร์ชันที่กำหนด ตัวอย่าง เช่น หากต้องการเข้ารหัสข้อมูลที่เป็น Alphanumeric จำเป็นต้องเลือกโหมด Alphanumeric ในตาราง 2-2 คือ 0010 ซึ่งจะต้องมาคูณจำนวนบิตในการนับ [8] ดังแสดงรายละเอียดในตาราง 2-4

ตารางที่ 2-4 จำนวนบิต Character Count Indicator ของ QR Code

Version	Numeric Mode	Alphanumeric Mode
1 ถึง 9	10	9
10 ถึง 26	12	11
27 ถึง 40	14	13

2.3.3 Error Correction Coding

เมื่อทำการแบ่ง Codeword ที่ได้ตามจำนวนบิตที่ต้องการแล้ว ใน QR Code จะใช้อัลกอริทึมของ Reed-Solomon Error Correcting [3] ซึ่งในทุกๆ บิตของ Codeword จะมีการคำนวณ Error Correction Codewords [8] จากนั้นจะนำ Error Correction Codewords ที่ได้มาต่อท้ายข้อมูลเป็นลำดับต่อไป ซึ่งใน QR Code จะแบ่งระดับของ Error Correcting ดังแสดงในตาราง 2-5

ตารางที่ 2-5 แสดงระดับ Error correcting ใน QR Code

Error Correcting Level	Recovery Capacity % (approx.)
L	7
M	15
Q	25
H	30

2.3.4 การเตรียม Structure final message

ในการเตรียม Structure final messageทำได้โดยการแทรกส่วนของ Data Codewords และ Error Correction Codeword ในแต่ละบล็อกซึ่งประกอบไปด้วยขั้นตอนดังต่อไปนี้

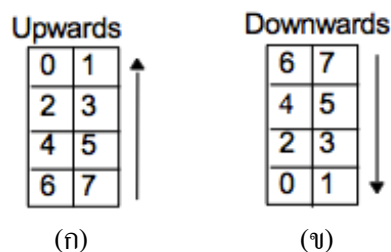
- 2.3.4.1 ทำการแบ่ง Data Codewords ตามลำดับโดยแบ่งเป็น n บล็อกตามที่ระบุไว้ใน [6] ซึ่งจะถูกกำหนดตามเวอร์ชัน และระดับของ Error Correcting
- 2.3.4.2 ในทุกๆ Data Codewords จะต้องมีปริมาณ Corresponding block ของ Error Correction Codeword [3]
- 2.3.4.3 ทำการเริ่มการเรียงข้อมูลตามลำดับ โดยเริ่มจาก Data block 1, Codeword 1 ตามด้วย Data block 2, Codeword 2 ทำไปเรื่อยๆ จนถึง Data block และ Codeword ตัวสุดท้าย จากนั้นจะทำการต่อข้อมูลด้วย Error Correction Codeword ตามลำดับดังภาพ 2-11 ซึ่งจากรายการลำดับที่แสดงในภาพ 2-11 จะได้ Final Message Codeword จะได้ดังนี้ D1, D12, D23, D35, D2, D13, D24, D36, ... D11, D22, D33, D45, D34, D46, E1, E23, E45, E67, E2, E24, E46, E68, ... E22, E44, E66, E88.

	Data codewords				Error correction codewords			
Block 1	D ₁	D ₂	D ₁₁	E ₁	E ₂	E ₂₂
Block 2	D ₁₂	D ₁₃	D ₂₂	E ₂₃	E ₂₄	E ₄₄
Block 3	D ₂₃	D ₂₄	D ₃₃	D ₃₄	E ₄₅	E ₄₆
Block 4	D ₃₅	D ₃₆	D ₄₅	D ₄₆	E ₆₇	E ₆₈

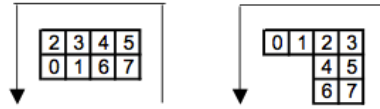
ภาพที่ 2-11 แสดงรูปแบบการวาง Data Codeword และ Error correction codewords

2.3.5 ขั้นตอน Module placement in matrix

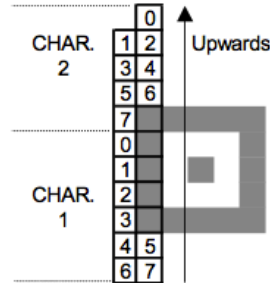
เป็นการนำโมดูลที่ได้จาก Final message มาวางลงในเมตริกซ์ ซึ่งในการวางโมดูลจะต้องทำร่วมกับส่วนประกอบที่เป็น Function Patterns ของ QR Code ซึ่งประกอบด้วย Finder pattern, Separators, Timing pattern และในบางเวอร์ชันจำเป็นต้องคำนึงถึง Alignment patterns สัญลักษณ์ของ QR Code ที่จะทำให้การวางนั้นแยกได้ 2 ประเภทคือ การวางสัญลักษณ์แบบปกติ และการวางแบบไม่ปกติซึ่งใน QR Code 2005 [3] ได้ระบุไว้คือ การวางสัญลักษณ์ทั้งสองชนิดจะถูกวางในตำแหน่งที่ขึ้นกับลักษณะเฉพาะตัวของสัญลักษณ์นั้น และต้องขึ้นกับตำแหน่งของ Function Pattern ด้วย ใน Codewords ส่วนมากจะอยู่ในรูปแบบ 2 x 4 ซึ่งจะมีตำแหน่งที่สามารถวางได้ในสองทิศทาง คือในทิศแนวตั้ง ซึ่งจะได้ ขนาด 2 โมดูลในแนวความกว้าง และ 4 โมดูลในแนวความสูงและถ้าจำเป็นต้องเปลี่ยนทิศทางเป็นแนวนอน Codeword จะถูกวางในทิศทางในแนวความกว้าง 4 โมดูล และในแนวความสูง 2 โมดูล สำหรับการวางสัญลักษณ์ที่ไม่ปกติเราจะใช้เมื่อต้องการเปลี่ยนทิศทางในการวางโมดูล ซึ่งจะถูกใช้เมื่อเข้าใกล้ Alignment Pattern หรือ Function Pattern ต่างๆ ดังแสดงในภาพที่ 2-12, ภาพที่ 2-13 และภาพที่ 2-14



ภาพที่ 2-12 ก) ทิศทางการวางข้อมูลในทิศทางขึ้นและ ข) ลง



ภาพที่ 2-13 ตัวอย่างทิศทางการวางในรูปแบบปกติและรูปแบบไม่ปกติ



ภาพที่ 2-14 ตัวอย่างการวางบิตในจุดที่ใกล้เคียงกับ Alignment pattern

2.3.6 ขั้นตอนการวาง Data masking

ในการทำ Data masking จำเป็นต้องมีการประเมินการเลือกรูปแบบ Masking ที่เหมาะสมกับในอัตราส่วนที่สม่ำเสมอระหว่างโมดูลที่เป็นสีขาวและโมดูลที่เป็นสีดำ ตามข้อมูลใน QR Code [3] เนื่องจากอัตราส่วน Bit pattern ของ Finder Pattern ซึ่งมีค่าเท่ากับ 1011101 ดังนั้น Data masking ที่ได้จะไม่สามารถมี Bit pattern เหมือนกับ Finder Pattern เพื่อป้องกันปัญหาที่จะเกิดขึ้นจึงจำเป็นต้องทำตามขั้นตอนดังต่อไปนี้

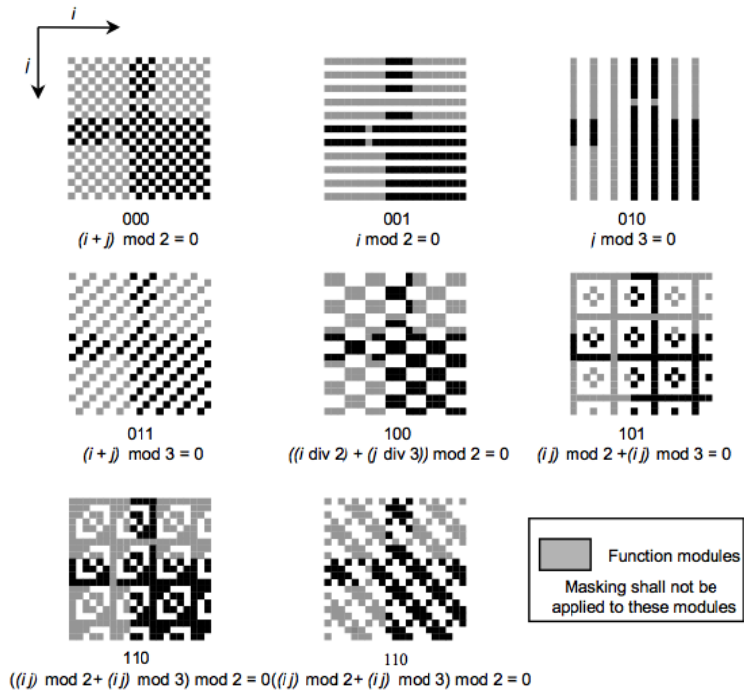
- 2.3.6.1 Data masking จะไม่ถูกนำไปวางในพื้นที่ที่เป็น Function patterns ซึ่งประกอบด้วย Quiet Zone, Finder Pattern, Separator, Timing patterns และ Alignment Patterns ซึ่งการทำ Data masking จะทำในส่วนของ Data และ Error correction Codewords ที่อยู่ใน Encoding Region เท่านั้น
- 2.3.6.2 ทำการแปลงโมดูลที่อยู่บริเวณ Encoding Region โดยที่มองข้ามส่วนของ Format Information และ Version Information โดยทำการ สลับค่าตามผลที่ได้จากการใช้ XOR ในผลที่ได้จากการทำ XOR ทำให้เกิดการสลับกันของโมดูลสีขาวและดำได้ซึ่งขึ้นอยู่กับรูปแบบของ Data Mask Pattern ที่เลือกใช้

- 2.3.6.3 ทำการประเมินผลลัพธ์ที่ได้จากการแปลง Pattern โดยการให้คะแนน Penalty กับรูปแบบที่ไม่ต้องการ โดยทำการให้คะแนนในทุกๆ ผลลัพธ์ที่ได้จากการแปลง
- 2.3.6.4 ทำการเลือก Mask Pattern ที่จะใช้โดยดูที่ Mask Pattern ใดที่มีคะแนน Penalty น้อยที่สุด

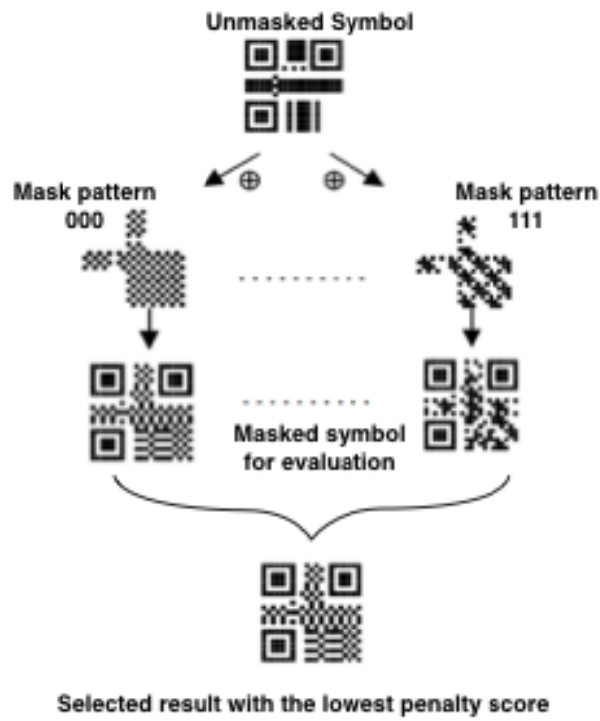
จากตารางที่ 2-6 แสดงเงื่อนไขที่ใช้ในการเลือกรูปแบบ Data Masking โดยอ้างอิงถึงบิตที่จะใช้ใน Format Information และใช้เป็นเงื่อนไขในการสร้าง Mask Pattern สำหรับ Mask pattern ที่สร้างขึ้นจะถูกกำหนดเป็นโมดูลสี่ค่าซึ่งอยู่ในพื้นที่ของ Encoding region (โดยไม่นับรวมพื้นที่ที่จะใช้สำหรับ Format Information และ Version Information) ซึ่งจะดูจากเงื่อนไขที่เป็นจริง i ในที่นี้แทนแถวของโมดูลที่สนใจ และ j แทนตำแหน่งของคอลัมน์ของโมดูลที่สนใจ ซึ่ง $(i,j) = (0,0)$ คือตำแหน่งซ้ายสุด เช่น ใน Mask Pattern ลำดับที่ 000 ในตำแหน่ง (10,10) จะได้ $(10+10) \bmod 2 = 0$ ให้ความหมายว่าเป็นจุดที่ต้องสลับสีบิตจากสีเดิม ที่ตำแหน่ง (9,10) จะได้ $(9+10) \bmod 2 = 1$ แปลว่า บิตสีเดิมจนทำครบทุก Pattern รูปแบบ Data mask pattern ที่มีทั้งหมดแสดงในภาพที่ 2-15 และภาพรวมขั้นตอนการวาง Data Mask แสดงในภาพที่ 2-16

ตารางที่ 2-6 แสดงเงื่อนไขที่ใช้ในการเลือก Pattern

Data mask pattern reference for QR Code symbols	Condition
000	$(i+j) \bmod 2 = 0$
001	$i \bmod 2 = 0$
010	$j \bmod 3 = 0$
011	$(i+j) \bmod 3 = 0$
100	$((i \text{ div } 2) + (j \text{ div } 3)) \bmod 2 = 0$
101	$(i+j) \bmod 2 + (i+j) \bmod 3 = 0$
110	$((i+j) \bmod 2 + (i+j) \bmod 3) \bmod 2 = 0$
111	$((i+j) \bmod 2 + (i+j) \bmod 3) \bmod 2 = 0$



ภาพที่ 2-15 รูปแบบ Data mask pattern ทั้งหมด [3]



ภาพที่ 2-16 แผนภาพขั้นตอนการวาง Data Mask [3]

2.3.7 ขั้นตอนการวาง Format information และ Version information

ซึ่งเป็นขั้นตอนสุดท้ายของการสร้าง QR Code โดยมีรายละเอียดของขั้นตอนดังต่อไปนี้

2.3.7.1 Format information เป็นส่วนที่มีความยาวทั้งสิ้น 15 บิต ที่ประกอบด้วย 5 บิต คือ Data bits และ อีก 10 บิต คือ Error correction bits ซึ่งคำนวณได้จาก BCH (15,5) [3] โดย 2 บิตแรกในส่วนของ Data bits จะมีการบอกถึง Indicator ของระดับ Error Correction ซึ่งเป็นไปตาม ตารางที่ 2-7

ตารางที่ 2-7 Indicators ของ Error Correction Level ใน QR Code

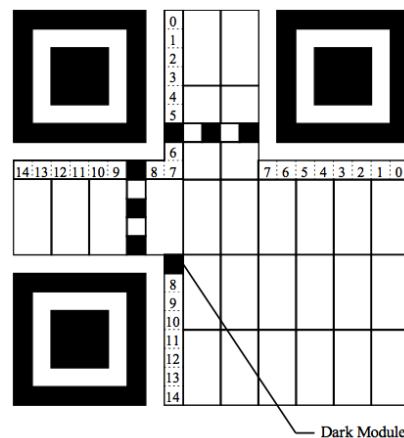
Error Correction Level	Binary indicator
L	10
M	00
Q	11
H	10

ตั้งแต่ตำแหน่งที่ 3 ถึงตำแหน่งที่ 5 ของชุดข้อมูลของ Format information จะมีใส่หมายเลข indicator ของ data mask pattern ดังตารางที่ 2-6 เมื่อมีการทำ Error correction จะได้ ข้อมูลของ Error correction bits ซึ่งมีจำนวนทั้งสิ้น 10 บิต จากนั้นนำ Data bits จำนวน 5 บิตมาต่อท้ายแล้วทำการ XOR 15 bit Error Corrected Format Information กับ Mask Pattern 101010000010010 ในการรวม Error Correction และ Data Mask Pattern จำเป็นต้องแน่ใจว่าข้อความบิตที่ได้ไม่เป็นหมายเลข 0 ทั้งชุด

ผลลัพธ์จากการคำนวณเพื่อสร้าง Format information จะถูกวางตามพื้นที่ที่เหมาะสมดังภาพ 2-17 ซึ่งในการวาง Format information จะมีการวางทั้งหมด 2 ที่เพื่อให้ในกระบวนการถอดรหัส ใน Format information จะมี Significant bit ซึ่งมีความจำเป็นต้องวางไว้ที่โมดูลที่ตำแหน่ง 0 และในโมดูลตำแหน่งที่ 14 และในโมดูลตำแหน่ง $(4V + 9, 8)$ ซึ่ง V แทนหมายเลข Version ซึ่งในตำแหน่งนี้จะต้องเป็นสีดำเสมอ ซึ่งจะไม่ถูกมองว่าเป็นส่วนหนึ่งของ Format information

ตัวอย่างของ Format information

สมมติว่าจะเลือก Error Correction level M	:	00
เลือก Data mask pattern reference	:	101
ตัวอย่างข้อมูล	:	00101
BCH bit	:	0011011100
Unmasked bit sequence	:	001010011011100
Mask pattern สำหรับ XOR operation	:	101010000010010
Format information module pattern	:	100000011001110



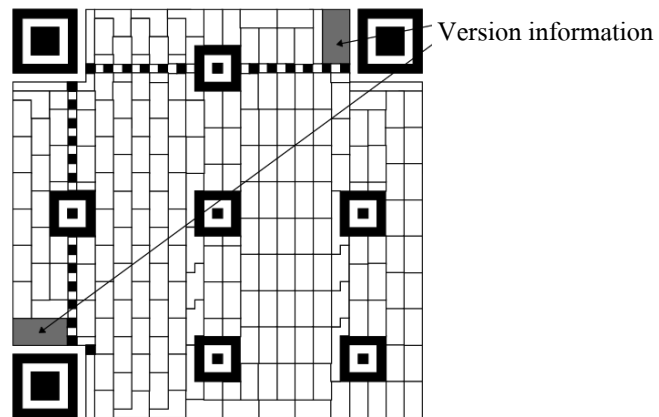
ภาพที่ 2-17 ตำแหน่งการวาง Format information

2.3.7.2 Version information สำหรับ Version information จะมีใน QR Code เวอร์ชันตั้งแต่ 7 เป็นต้นไปซึ่งใน Version information จะประกอบด้วยหมายเลขทั้งหมด 18 bits ซึ่งประกอบด้วย Data bits 5 บิต และอีก 12 บิต จะเป็น Error correction bit โดยคำนวณได้จาก $[3](18,6)$ สำหรับใน QR Code ในเวอร์ชันที่ต่ำกว่า 7 นั้นในส่วนของ Version information จะเป็น 0 ทั้งหมด เมื่อได้ผลจากการคำนวณ Version information แล้วจะนำไปวางที่ตำแหน่งที่จองไว้สำหรับ Version information ดังที่แสดงในภาพที่ 2-18 พื้นที่เก็บ Version information จะมีทั้งหมดสองตัวอย่างเช่นใน Format information เพื่อประโยชน์ในเรื่องของ Redundancy ซึ่ง Version information จะอยู่ที่ตำแหน่งของโมดูลที่ 0 และ Significant bit จะอยู่ที่ตำแหน่งโมดูล 17 ซึ่งแสดงให้เห็นในภาพที่ 2-19

ตัวอย่างของข้อมูล Version information

สมมติหมายเลข Version : 7
 ตัวอย่างข้อมูล : 000111
 BCH bits : 110010010100
 Format information module pattern : 000111110010010100

ใน Version information ตัวที่ 1 จะมีพื้นที่ทั้งหมด 6 x 3 module block ซึ่งอยู่ที่ตำแหน่งซ้ายของ Finder pattern ที่อยู่ฝั่งขวาบน โดยที่จะอยู่ติดกับ Separator และ Version information ตำแหน่งที่ 2 จะมีขนาด 3 x 6 module block ซึ่งจะอยู่เหนือ Finder Pattern ที่อยู่ในตำแหน่งล่างซ้าย และอยู่เหนือส่วนของ Separator



ภาพที่ 2-18 ตำแหน่งของ Version information

0	3	6	9	12	15
1	4	7	10	13	16
2	5	8	11	14	17

0	1	2
3	4	5
6	7	8
9	10	11
12	13	14
15	16	17

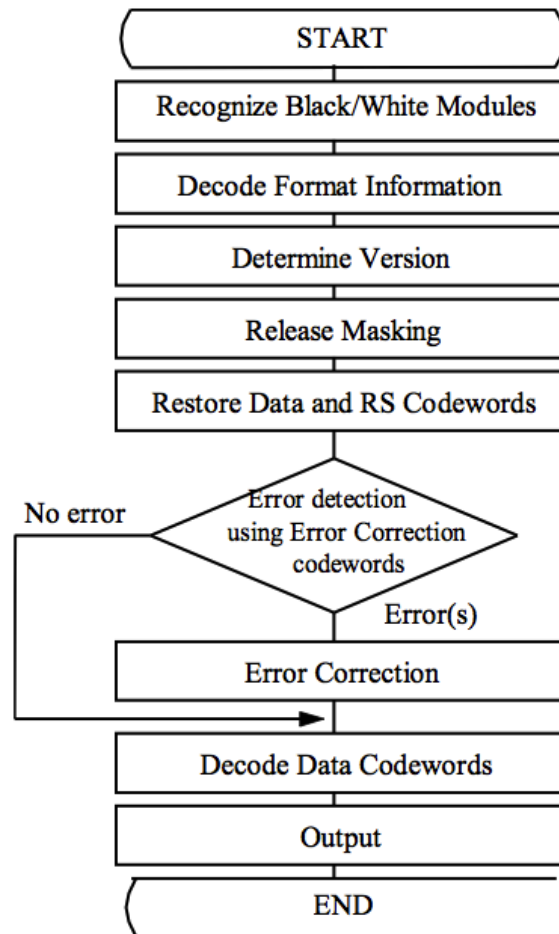
Version Information in lower left

Version Information in upper right

ภาพที่ 2-19 ทิศทางการวาง Version information

2.4 ขั้นตอนในการถอดรหัส QR Code

ขั้นตอนในการถอดรหัส QR Code นั้น ใน QR Code 2005 [6] ได้ระบุไว้ว่าเป็นขั้นตอนที่ตรงข้ามกับการ Encode ซึ่งแสดงในภาพที่ 2-20

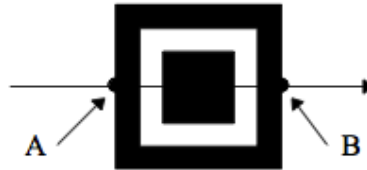


ภาพที่ 2-20 แสดงแผนผังการถอดรหัส QR Code

รายละเอียดของขั้นตอนการถอดรหัส QR Code มีดังต่อไปนี้

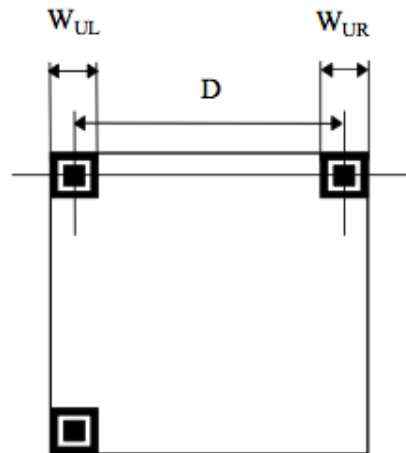
- 2.4.1 ทำการแปลงภาพที่นำเข้าโดยการแปลงเป็นภาพขาวดำ โดยใช้ Global Threshold
- 2.4.2 ทำการค้นหา Finder Pattern ใน QR Code [6] จะมีใน 3 แห่งที่ลักษณะโมดูลของทั้ง 3 จะประกอบไปด้วย dark-light-dark-light-dark โดยมีอัตราส่วนตามความกว้างของโมดูลย่อย คือ 1:1:3:1:1

- 2.4.2.1 ในการค้นหาจะต้องแน่ใจว่าได้ จุด A ซึ่งเป็นจุดเริ่มต้น และจุด B โดยที่จุดทั้งสองจะเป็นจุดที่อยู่บนขอบนอกของ Finder Pattern ที่สนใจ จากนั้นทำการหาจุดศูนย์กลางของทั้งสองจุดซึ่งผลที่ได้จะ ได้จุดศูนย์กลางของ Finder Pattern เราจะได้แกน x ที่ต้องการ ดังภาพที่ 2-21



ภาพที่ 2-21 การแสกนไลน์ ใน Finder Pattern

- 2.4.2.2 ทำขั้นตอน 2.4.2.1 ซ้ำอีก 2 ครั้งโดยเลือกคอลัมน์ที่ผ่านจุดกึ่งกลางโดยที่สนใจในแกน y
- 2.4.2.3 ในการเลือกจุดศูนย์กลางของ Finder pattern นั้นทำได้ด้วยการวาดเส้นจากตำแหน่ง A ไป B โดยเริ่มที่ขอบนอกสุดของ Finder pattern ทำการหาจุดศูนย์กลางซึ่งจะได้แกน x ที่สนใจ จากนั้นทำการลากเส้นในแกน y จากนั้นจุดศูนย์กลางของ Finder pattern คือ จุดที่เส้นตัดกัน
- 2.4.2.4 ทำซ้ำในขั้นตอน 2.4.2.1 และ 2.4.2.3 เพื่อหา Finder pattern ที่เหลืออีก 2 ที่
- 2.4.2.5 ถ้าไม่สามารถหาพื้นที่ที่สนใจได้ให้ทำการทำการ Reverse module ก่อนที่จะทำกระบวนการที่ 2.4.2.1 และในการถอดรหัสก็จะยึดตามนั้น
- 2.4.2.6 ในกรณีที่พบ Finder Pattern เพียง 1 ตัว กระบวนการถอดรหัสนั้นจะมอง QR Code นั้นเป็น Micro QR Code
- 2.4.3 ทำการพิจารณาคำแหน่งการหมุน โดยการวิเคราะห์จุดศูนย์กลางของกลุ่มของ Finder pattern เพื่อให้ทราบว่า Finder pattern ที่อยู่ในตำแหน่งด้านซ้ายบน และเพื่อให้ทราบมุมการหมุนของสัญลักษณ์ด้วย
- 2.4.4 พิจารณาระยะทางของ D ในขั้นตอน 2.4.1 ที่ทำได้จากจุดศูนย์กลางของ Finder pattern มุมซ้ายบนถึงจุดศูนย์กลางของ Finder pattern ขวาบน โดยที่ความกว้างของ Finder pattern ทั้งสองตัวแทนด้วย W_{UL} ในฝั่งบนซ้าย และ W_{UR} ในฝั่งบนขวา ดังแสดงในภาพ 2-22



ภาพที่ 2-22 แสดง Finder Pattern ที่อยู่ด้านบนของ QR Code

2.4.5 ทำการคำนวณหาค่า Nominal X Dimension ของสัญลักษณ์ QR Code โดยใช้สมการที่ 2-1

$$X = \frac{(W_{UL} + W_{UR})}{14} \quad (2-1)$$

2.4.6 สมการที่ใช้ในการคำนวณ Version (V) ของ QR Code สามารถทำได้โดยใช้สมการที่ (2-2)

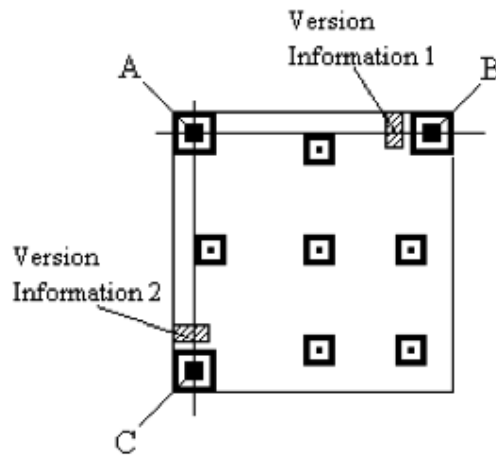
$$V = \left\lfloor \left[\left(\frac{D}{X} \right) - 10 \right] \right\rfloor \quad (2-2)$$

2.4.7 เงื่อนไขการคำนวณ Version ถ้าผลลัพธ์จากการคำนวณได้ 6 หรือน้อยกว่านั้น แสดงว่าผลลัพธ์ที่ได้นั้นเป็นหมายเลข Version แต่ในกรณีที่ได้ออกมาคือ 7 หรือมากกว่านั้นให้ทำการใช้กระบวนการดังต่อไปนี้ในการหา Version ของภาพ QR Code นั้นๆ

2.4.7.1 ทำการหารความกว้าง W_{UR} ด้วย 7 เพื่อคำนวณหาขนาดของโมดูล CP_{UR} โดยใช้สมการดังต่อไปนี้

$$CP_{UR} = \frac{W_{UR}}{7} \quad (2.3)$$

2.4.7.2 ทำการ Sampling Guide Line AB และ AC จากจุด A จุด B และจุด C โดยการลากเส้นผ่านทั้งสามจุดของ Finder Pattern ดังที่ได้แสดงไว้ในภาพที่ 2-22 ผลที่ได้จากการหาเส้น Guide ที่ผ่านกึ่งกลางโมดูล โดยพื้นที่ของ Version Information 1 สามารถทำการคำนวณได้จากจุดที่ที่ขนานกับเส้นที่ Guid ซึ่งขนาดของโมดูล CP_{UR} ค่าใดนารีจะถูกพิจารณาจากโมดูลสีขาว (Light Pattern) หรือ โมดูลสีดำ (Dark Pattern) จากกระบวนการทำ Sampling Grid



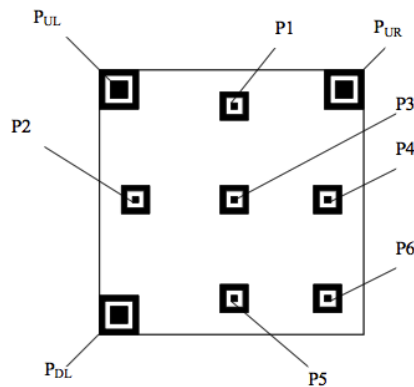
ภาพที่ 2-22 ตำแหน่งของ Finder pattern และ Version information

2.4.7.3 ทำการหา Version โดยการตรวจหาได้โดยการหา Error Correction ตามในตาราง Anex D.2 [3]

2.4.7.4 ถ้าหาก Error ที่พบมากกว่า Error correction ที่ได้มีการตรวจพบให้ทำการคำนวณความกว้างที่ Finder pattern ที่อยู่ในตำแหน่งด้านข้างซ้ายโดยที่มีขั้นตอนเหมือนกับในขั้นตอน 2.4.2.1, 2.4.2.2 และ 2.4.2.3 โดยที่จะทำการถอดรหัสข้อมูล Version จาก Version information ในตำแหน่งที่ 2 ซึ่งคือตำแหน่งที่แสดงในภาพที่ 2-22

2.4.8 การคำนวณสำหรับใน Version 1 ให้ทำการกำหนดค่าของ X โดยที่เป็นการประมาณระยะห่างระหว่างโมดูลสีขาวและสีดำในบริเวณ Timing patterns ซึ่งในทำนองเดียวกันการคำนวณโดยให้ Y เป็นการประมาณช่องว่างของจุดกึ่งกลางของโมดูลสีดำและโมดูลสีขาวในส่วนของ Timing pattern ในฝั่งซ้าย ทำการ Sampling

grid โดยการลากเส้นแนวนอนผ่าน Timing pattern ด้านบนด้วยเส้นที่ขนานกันในตำแหน่งช่องว่างแนวตั้งของจุด Y ซึ่งจะประกอบด้วยเส้นถึง 6 เส้นเหนือ Timing pattern และทุกเส้นใต้ Timing pattern จะเป็น Version ของ QR Code ในขั้นตอนต่อไป ให้ทำการลากเส้นตามแนวตั้งผ่าน Timing pattern ทางฝั่งซ้ายโดยที่เส้นจะขนานกันในตำแหน่งที่เป็นช่องว่างแนวนอนของ X ซึ่งจะประกอบไปด้วยเส้นอ้างอิงในแนวตั้งทั้ง 6 และในทุกๆ เส้นทางขวาของ Timing pattern ถูกกำหนดให้เป็นไปตาม Version ของ QR Code สำหรับ QR Code ใน Version ที่มากกว่าให้พิจารณา กู้อันดับกึ่งกลางของแต่ละ Alignment pattern ซึ่งดูได้ที่ Annex E [3] ในการทำ Sampling grid ระหว่างจุดตั้งในภาพที่ 2-23



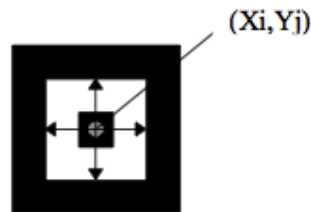
ภาพที่ 2-23 ตำแหน่งของ Finder patterns และ Alignment pattern ในการ Sampling grids

- 2.4.8.1 ทำการหาความกว้างของ W_{UL} โดยการแบ่ง Finder Pattern ด้านบนซ้าย P_{UL} ด้วย 7 ในการหา CP_{UL} สามารถคำนวณได้จากสมการที่ 2.4 ซึ่งแสดงดังต่อไปนี้

$$CP_{UL} = \frac{W_{UL}}{7} \quad (2.4)$$

- 2.4.8.2 ทำการพิจารณาคู่อันดับระหว่าง Alignment Pattern P1 กับ Alignment Pattern P2 จากภาพ 2-22 โดยการดูที่จุดศูนย์กลาง A ใน Finder Pattern ฝั่งซ้ายบน P_{UL} โดยที่เส้นขนานกับเส้น AB และเส้น AC รวมถึงขนาดของ CP_{UL} ด้วย

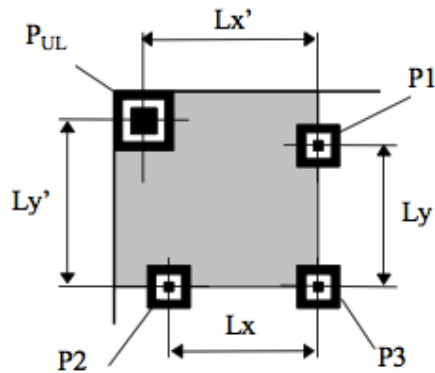
- 2.4.8.3 ทำการหาพื้นที่ขาวภายใน Alignment Pattern ที่จุด P1 และ P2 โดยเริ่มจากจุดกึ่งกลางของ Alignment Pattern เพื่อทำการหา X_i, Y_j ดังแสดงในภาพที่ 2-24



ภาพที่ 2-24 จุดศูนย์กลางกลางของ Alignment Pattern

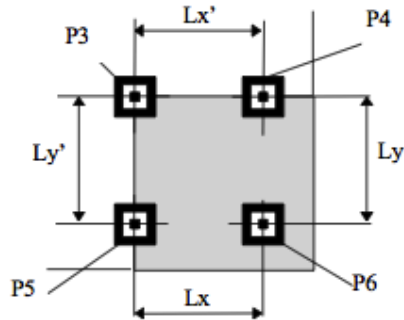
- 2.4.8.4 ทำการประมาณการจุดศูนย์กลางกลางของ Alignment pattern ที่ P3 โดยขึ้นอยู่กับจุดศูนย์กลางกลางของ Finder Pattern บนซ้าย P_{UL} และจุดศูนย์กลางกลางของ Alignment pattern P1 และ P2 หาได้จาก 2.4.8.3
- 2.4.8.5 ทำการหาจุดศูนย์กลางกลางของ Alignment pattern P3 โดยใช้วิธีเดียวกันในขั้นตอนที่ 2.4.8.3
- 2.4.8.6 ทำการหา L_x ซึ่งคือระยะทางระหว่างจุดศูนย์กลางกลางของ Alignment Pattern P2 ไปยัง P3 และ L_y คือระยะห่างจากจุดศูนย์กลางกลางของ Alignment Pattern P1 ไปยัง P3 จากนั้นทำการแบ่ง L_x และ L_y ด้วยการกำหนดพื้นที่ว่างของ Alignment pattern โดยใช้สมการโดยที่กำหนดให้ AP คือพื้นที่ว่างใน Module ของ Alignment pattern ซึ่งดูได้จากใน (Table E.1) [3] และในแนวเดียวกันในการหา L_x' ซึ่งเป็นระยะทางในแนวอนของจุดกึ่งกลางของ Finder Pattern P_{UL} กับ จุดศูนย์กลางกลางของ Alignment pattern P1 และ L_y' ซึ่งเป็นระยะทางแนวตั้งฉากจากจุดศูนย์กลางกลางของ Finder Pattern ด้านบนซ้าย P_{UL} กับ จุดศูนย์กลางกลางของ Alignment pattern P2 จากนั้นทำการ แบ่ง L_x' และ L_y' เพื่อให้ได้ค่า CP_x' ของตำแหน่งด้านบนและ CP_y' ในฝั่งซ้ายบนของภาพสัญลักษณ์ QR Code โดยใช้สมการดังนี้
- $$CP_x' = L_x' / (\text{คอดัมน์ของคู่อันดับจุดศูนย์กลางกลางของ Alignment pattern P1} - \text{คอดัมน์ของคู่อันดับของจุดศูนย์กลางกลางของ Finder pattern P}_{UL})$$
- ซึ่งเป็น Finder pattern ที่อยู่ทางด้านซ้ายบน

$CPy' = Ly' /$ (แถวของคู่อันดับของจุดศูนย์กลางใน Alignment pattern
P2 - แถวของคู่อันดับของจุดศูนย์กลางของ Finder Pattern P_{UL}



ภาพที่ 2-25 สัญลักษณ์ด้านบนซ้ายของ QR Code

- 2.4.8.7 ในการพิจารณาพื้นที่ Sampling grid ที่คลุมที่บริเวณด้านบนซ้าย โดยที่จะขึ้นอยู่กับ CPx , CPx' , CPy และ CPy' ซึ่งแสดงพื้นที่ของสัญลักษณ์ QR Code ในตำแหน่งด้านบนซ้าย
- 2.4.8.8 ในการทำ Sampling grids ในพื้นที่ฝั่งบนขวา ก็มีวิธีการในการดำเนินการในทางเดียวกัน (โดยที่ตีเส้นปกคลุม ด้านบนขวา ของ Finder pattern P_{UR} ในส่วนของ Alignment pattern P1, P3 และ P4) และในทางพื้นที่ด้านล่างซ้าย มีการปกคลุมในส่วนของ Finder pattern P_{UL} , ในส่วน Alignment pattern P2, P3 และ P5 ของสัญลักษณ์ QR Code
- 2.4.8.9 สำหรับใน Alignment pattern ที่จุด P6 จากภาพที่ 2-25 การประมาณคู่ของจุดศูนย์กลางในระดับของ CPx' และ CPy' ซึ่งเป็นค่าที่ได้มาจากพื้นที่ว่างของ Alignment pattern โดยที่จุด P3, P4 และ P5 โดยที่จุดดังกล่าวคือจุดศูนย์กลางของ Alignment pattern เหล่านั้น



ภาพที่ 2-26 สัญลักษณ์ในบริเวณขวาล่างของ QR Code

- 2.4.8.10 ทำขั้นตอน 2.4.8.5 จนถึง 2.4.8.8 เพื่อใช้ในการทำ sampling grid ในส่วนพื้นที่ด้านล่างขวา ของ QR Code
- 2.4.8.11 ตรวจสอบการทำ sampling grid ว่ามีพื้นที่ใดที่ยังไม่ครอบคลุมบ้าง โดยวิธีเดียวกันนี้
- 2.4.9 Global Threshold พื้นที่ตัวอย่างขนาด 3x3 พิกเซลจากนั้นทำการแบ่งเป็น 2 กลุ่มคือกลุ่มที่มีสีเข้มแทนด้วยเลข 1 และแทน โมดุลที่มีสีขาวเป็น 0
- 2.4.10 ทำการถอดรหัส Format information ที่อยู่ใกล้กับ Finder pattern ด้านซ้ายบนโดยการดูที่ Error Correction Level และข้อมูลของ Data Mask Pattern ที่ใช้บนสัญลักษณ์บริเวณนั้น ถ้า Error ที่พบมากกว่าระดับของ Error Correction Level การถอดรหัส Format information จะทำการย้ายไปสนใจในส่วนของ Format information ที่อยู่บริเวณ ใกล้กับ Finder pattern ขวาบน และล่างซ้ายตามลำดับ
- 2.4.11 ในกรณีที่ Format information ไม่สามารถทำการถอดรหัสได้ให้ทำการอ่านในทิศทางตรงกันข้าม และทำการถอดรหัสในแบบภาพ Mirror ของตำแหน่ง Row และ Column ที่เป็นคู่อันดับ Transposed
- 2.4.12 ทำการ XOR ส่วนของ Data Mask Pattern กับบริเวณพื้นที่ของ Data ที่ถูกเข้ารหัสของ QR Code เพื่อทำการถอด Data Mask และการเรียกคืนบริเวณส่วนของ Data Codewords และ Error Correction Codewords ในกระบวนการนี้จะเป็นขั้นตอนที่ตรงกันข้ามกับกระบวนการในการเข้ารหัส QR Code
- 2.4.13 ทำการพิจารณาสัญลักษณ์ของ Codewords ตามกฎการวาง [3] (6.7.3)
- 2.4.14 ทำการจัดลำดับของ Codeword ให้อยู่ในรูปแบบ Block โดยทำให้เป็นไปตามรูปแบบของ Symbol Version และ Error Correction Level

- 2.4.15 ทำการตรวจหาข้อผิดพลาด และใช้กระบวนการในการกู้คืนข้อมูลซึ่งแสดงใน Annex B [3] เพื่อให้ได้ข้อมูลที่เป็นไปตาม Error Correction Level
- 2.4.16 ทำการ Restore ข้อมูลต้นฉบับที่มี Bit stream เป็นลำดับของ Data Block
- 2.4.17 ทำการแบ่ง Data bit stream เป็นส่วนๆ โดยทำการเริ่มต้นแต่ละส่วนด้วย Mode indicator และความยาวจะขึ้นอยู่กับคุณสมบัติของ Indicator นั้นๆ
- 2.4.18 ทำการถอดรหัสในแต่ละส่วนที่ได้จากการแบ่ง Data bit stream

2.5 พื้นฐานการประมวลผลภาพดิจิทัล

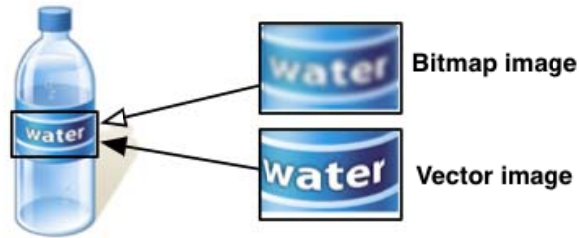
ภาพเพียงหนึ่งภาพสามารถแทนความหมายได้อย่างหลากหลาย ภาพถูกนำมาใช้ในกระบวนการสื่อสารมาตั้งแต่ในยุคเริ่มต้นของมนุษย์จนกระทั่งในปัจจุบันนี้ เทคโนโลยีการถ่ายภาพได้มีการพัฒนามาเป็นในรูปแบบดิจิทัลทั้งการจับเก็บและการแสดงผลทางหน้าจอของคอมพิวเตอร์ สมาร์ทโฟน โทรท์สน์ และอุปกรณ์อิเล็กทรอนิกส์ต่างๆ

2.5.1 รูปแบบของภาพดิจิทัล

2.5.1.1 ภาพในรูปแบบ Bitmap คือภาพที่เกิดจากการประกอบกันของจุดเล็กๆ ที่เรียกว่า พิกเซล (Pixel) โดยที่มีการเรียงต่อกันเป็นตารางเล็กๆ โดยที่ตารางเหล่านั้นจะมีการจัดเก็บค่าสีที่เรียงกันเมื่อมองในภาพรวมแล้วผลของการรวมจุดหลายๆจุดจะกลายเป็นภาพที่สื่อความหมาย เนื่องด้วยตารางเหล่านั้นมีขนาดเล็ก จนดวงตาของมนุษย์ไม่สามารถที่จะแยกแยะรายละเอียดข้อย่อยนั้นได้ ในภาพ Bitmap นี้เมื่อทำการขยายภาพ ตารางที่บรรจุค่าสีจะมีการขยายด้วยจึงทำให้ภาพมีลักษณะที่ไม่คมชัดโดยทั่วไปแล้วภาพ Bitmap จะได้แก่ ภาพถ่ายภาพที่ได้จากเครื่องสแกนเนอร์ ซึ่งประโยชน์ที่ชัดเจนของภาพ Bitmap คือความสามารถในการไล่โทนสีรวมถึงระดับแสงเงาให้มีความใกล้เคียงกับวัตถุจริงมากที่สุดได้

2.5.1.2 ภาพในรูปแบบ Vector คือภาพที่ประกอบด้วย เส้นต่างๆ ทางเรขาคณิต ได้แก่ เส้นตรง เส้นโค้ง รูปทรงทางเรขาคณิตต่างๆ ภาพต่างๆที่ถูกสร้างขึ้นโดยสมการทางคณิตศาสตร์ ดังนั้นในโปรแกรมที่ทำการแสดงผลและแก้ไขภาพ เช่น การขยายภาพ การลดขนาดภาพ จำเป็นต้องนำสมการเหล่านั้นมาคำนวณค่าเพื่อสร้างรูปภาพขึ้นมาใหม่ในทิศทางที่สอดคล้องกับภาพต้นฉบับ ดังนั้นในภาพประเภทนี้จึงตรงข้ามกับภาพ Bitmap โดยที่ภาพ Vector จะสามารถทำการเปลี่ยนแปลงรูปทรงได้โดยที่ไม่ส่งผลต่อคุณภาพของรูปภาพ ในภาพผลลัพธ์ที่ได้จะมีลักษณะคมชัดมากโดยทั่วไปแล้วภาพแบบ Vector จะถูกนำไปใช้กับงานที่

ต้องการความละเอียดสูง เช่น ในการสร้างภาพ 3 มิติ ภาพ Logo ของผลิตภัณฑ์ ภาพในแผนผังต่างๆ เป็นต้น ตัวอย่างของภาพ Bitmap และภาพ Vector แสดงในภาพที่ 2-27



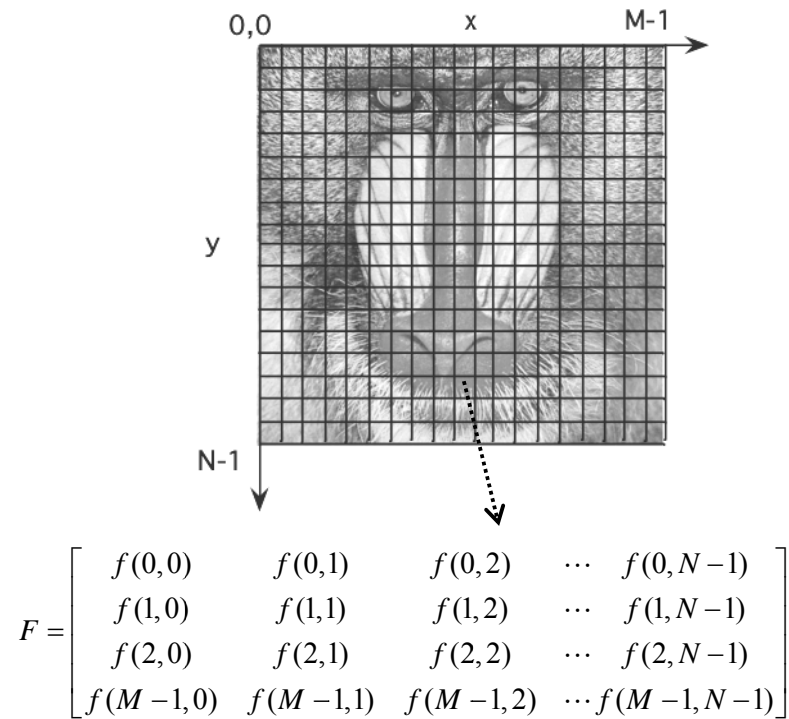
ภาพที่ 2-27 ตัวอย่างรูปภาพในรูปแบบ Bitmap และ Vector

2.5.2 กระบวนการในการสร้างภาพดิจิทัล

กระบวนการในการสร้างภาพดิจิทัลนั้นเราสามารถทำการแบ่งออกได้ 3 ขั้นตอน ซึ่งประกอบด้วยขั้นตอนดังต่อไปนี้

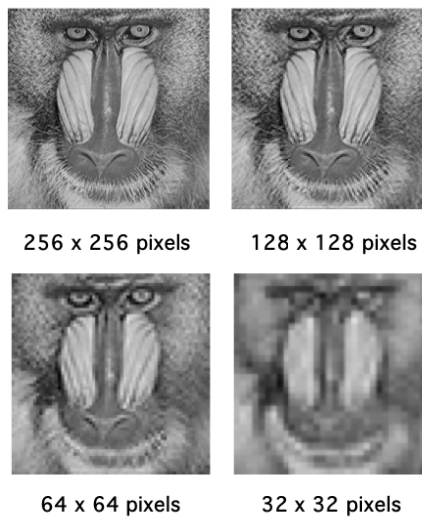
1. การได้มาของภาพ (Image Acquisition) เป็นการบันทึกภาพโดยใช้อุปกรณ์นำเข้าภาพซึ่งโดยทั่วไปภาพเหล่านั้นที่อยู่ในรูปแบบดิจิทัลนั้นจะเป็นสัญญาณที่มีลักษณะต่อเนื่อง (Continuous) ดังนั้นในอุปกรณ์กล้องถ่ายภาพดิจิทัล หรือในกล้องวิดีโอดิจิทัลจะทำหน้าที่ในการแปลงภาพที่เป็นลักษณะ 3 มิติเป็นภาพเชิง 2 มิติเพื่อใช้ในการแสดงผลบนหน้าจอภาพของคอมพิวเตอร์

2. การสุ่มเลือกเพื่อกำหนดตำแหน่ง (Image Sampling) เป็นการแปลงสัญญาณภาพที่ได้เป็นภาพ 2 มิติ ให้เป็นภาพเชิงดิจิทัลโดยที่ทำการสุ่มข้อมูลจากตำแหน่งของภาพที่จุด $f(x, y)$ โดยใช้การกำหนดช่วงที่เหมาะสมกับข้อมูลภาพนั้นๆ โดยผลที่ได้จะเป็นเมตริกซ์ของปริมาณเต็มบวกขนาด $M \times N$ ดังภาพ 2-28 แสดงตัวอย่างผลลัพธ์ที่ได้จากการทำ Image Sampling



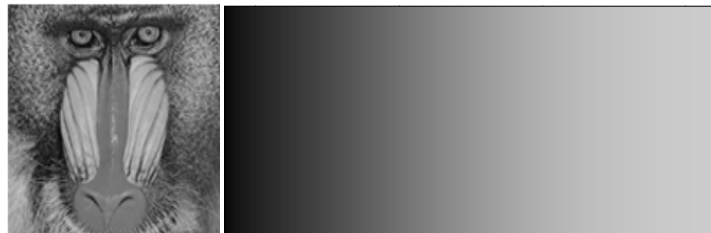
ภาพที่ 2-28 ตัวอย่างการทำ Image Sampling

ภาพผลลัพธ์ที่ได้จากการสุ่มเลือกตำแหน่งภาพที่ต้นนั้นจะทำให้ได้ภาพที่มีรายละเอียดใกล้เคียงกับต้นฉบับ โดยในทางกลับกันถ้าทำการสุ่มโดยใช้ตัวอย่างที่น้อยเกินไปก็จะทำให้ภาพผลลัพธ์ที่ได้มีความผิดเพี้ยนไปจากภาพต้นฉบับ ดังแสดงในภาพที่ 2-29



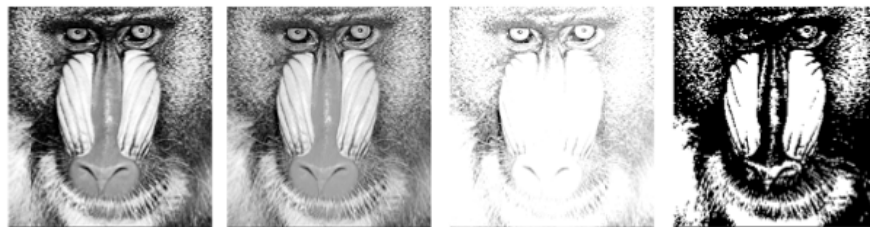
ภาพที่ 2-29 ภาพที่ได้จากอัตราการสุ่มที่แตกต่างกัน

3. การกำหนดระดับของความเข้มของแสง (Image Quantization) เป็นขั้นตอนของการนำภาพที่ได้จากการทำ Image Sampling มาทำการกำหนดระดับความเข้มของสีในระดับเทา ซึ่งในแต่ละจุดของภาพจะประกอบไปด้วยเฉดสีจากสีดำจนถึงสีขาวโดยที่สีดำจะถูกแทนค่าด้วย 0 และสีขาวจะถูกแทนค่าด้วย 255 ในกรณีที่เป็นภาพ 8 บิต ซึ่งจะมีทั้งสิ้น 2^8 เฉดสี (0-255) ดังภาพที่ 2-30 ซึ่งในหน่วยความจำจะมีการเก็บค่าด้วยรหัสฐานสองแทนค่าสีต่างๆ เช่น สีดำจะถูกแทนค่าด้วย 00000000 สีขาวจะถูกแทนค่าด้วย 11111111 โดยที่ค่าที่อยู่ระหว่างสีดำและสีขาวจะเป็นค่าบิตในเลขฐานสองที่เพิ่มขึ้นตามลำดับ



ภาพที่ 2-30 ภาพในระดับเทาที่มีค่าตั้งแต่ 0 -255

จะเห็นได้ว่าแต่ละจุดของภาพจะถูกแทนที่ด้วยตัวเลขที่มีค่า 0 -255 ในภาพระดับเทา แต่ในกรณีที่ต้องการแทนที่ค่าด้วยค่าเพียง 2 ค่า คือ ค่าสีขาวและสีดำ ซึ่งภาพนี้จะใช้พื้นที่ในการเก็บข้อมูลเพียง 1 บิตเท่านั้นซึ่งเรียกว่าภาพขาวดำ (Binary Image) ดังภาพ 2-31 แสดงตัวอย่างภาพระดับเทาที่แตกต่างกัน



8 bits

4 bits

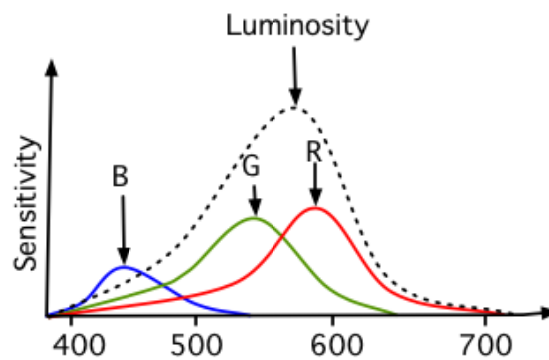
2 bits

1 bit

ภาพที่ 2-31 ภาพที่มีระดับเทา 8 bits 4 bits 2 bits และ 1 bit

2.5.3 คุณสมบัติของสี

สีเกิดจากคุณสมบัติทางกายภาพของแสงซึ่งเป็นส่วนสำคัญที่ทำให้เกิดสีขึ้นในระบบการมองเห็นของมนุษย์โดยธรรมชาติจะมีการตอบสนองไวต่อค่าสีมากกว่าค่าความสว่าง ดังนั้นเมื่อแสงกับระยะทางการมองวัตถุที่ทำให้มนุษย์สามารถแยกแยะสี และจะรับรู้ได้เมื่อแสงนั้นๆ ตกกระทบไปยังวัตถุที่ไม่ดูดซึมแสงสีดังกล่าว แสงที่เรามองเห็นนั้นเป็นพลังงานแม่เหล็กไฟฟ้า โดยมีความยาวคลื่นอยู่ 3 สี ได้แก่ แสงสีแดง (Red) ซึ่งเป็นคลื่นที่มีความยาว 580 นาโนเมตร แสงสีเขียว (Green) ซึ่งมีความยาวคลื่นประมาณ 545 นาโนเมตร และแสงสีน้ำเงิน (Blue) ที่มีความยาวคลื่น 440 นาโนเมตร ดังภาพ 2-32 นอกจากนี้การรับรู้ของมนุษย์ยังสามารถรับรู้ได้ถึงเนื้อสี (Hue) ซึ่งจะบอกถึงสีจริงของสีนั้นๆ เช่น สีแดง เป็นต้น อีกทั้งในการรับรู้ของมนุษย์ยังสามารถบอกถึงความอิ่มตัวของสี (Saturation) ซึ่งจะบอกถึงความรู้สึกที่มีต่อสีได้แก่ สีแดงเข้ม สีแดงอ่อน และสุดท้ายคือค่าความส่องสว่าง (Brightness) มนุษย์สามารถบอกได้ว่าสีนั้นมีมืดหรือสว่างได้



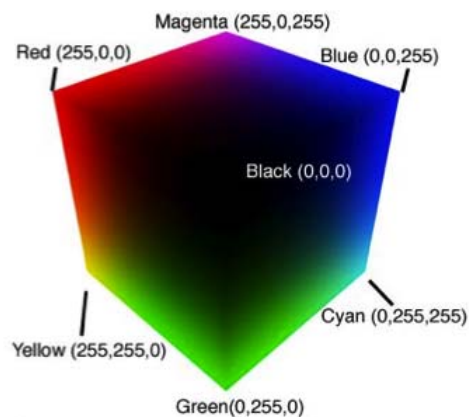
ภาพที่ 2-32 ค่าความยาวของคลื่นสีที่ตามนุษย์สามารถมองเห็น

2.5.4 แบบจำลองสี (Color Models)

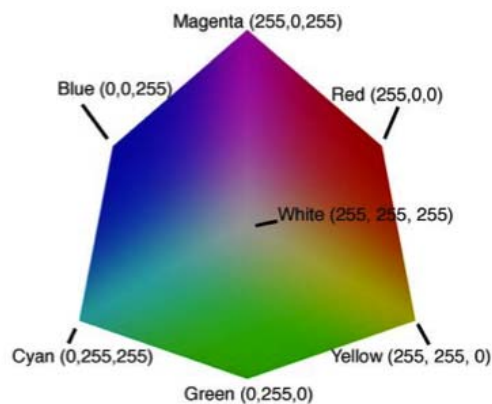
แบบจำลองสีเป็นวิธีในการกำหนดสีต่างๆ ตามมาตรฐาน ซึ่งในปัจจุบันมีหลากหลายให้เลือกใช้ตามงานที่เหมาะสมกับงานเนื่องด้วยคุณสมบัติที่แตกต่างของแต่ละแบบจำลองสี โดยที่ในแต่ละแบบจำลองสีจะมีหลักการนำแม่สีมาผสมเพื่อให้ได้สีอื่นๆ โดยแต่ละแบบจำลองสีจะมีเกณฑ์ที่แตกต่างกัน สำหรับแบบจำลองสีที่ใช้ในงานวิจัยนี้สนใจเพียง 2 แบบจำลองสีเท่านั้น ซึ่งได้แก่ RGB และ HSV โดยมีรายละเอียดดังต่อไปนี้

2.5.4.1 แบบจำลองสี RGB

ในแบบจำลองสีแบบนี้เกิดจากแม่สีหลัก (Primary Color) ที่มีอยู่ 3 สี คือ สีแดง (Red) สีเขียว (Green) น้ำเงิน (Blue) แบบจำลองสี RGB จะมีลักษณะเป็นลูกบาศก์โดยที่แต่ละมุมจะแทนด้วยแม่สีแต่ละสีได้แก่ สีแดง สีเขียว และสีน้ำเงิน ดังภาพที่ 2-33 และ 2-34 จากภาพ 2-33 ณ จุด (0,0,0) ซึ่งจะมีสีดำ และในอีกมุมหนึ่งที่จุด (255,255,255) ซึ่งจะเป็นมุมที่เป็นสีขาวดังแสดงในภาพ 2-34 ในการเลือกค่าสี จำเป็นต้องมีการระบุค่าสีแดง เขียว และน้ำเงิน เช่น ถ้าต้องการเลือกสีเหลือง จะต้องระบุ $R=255, G=255$ และ $B=0$ เมื่อมีการเปลี่ยนค่าสีใดสีหนึ่งในแม่สีก็จะส่งผลให้เกิดสีใหม่ทันทีโดยทั่วไปแบบจำลองสี RGB จะเหมาะกับงานที่ต้องการแสดงผลผ่านหน้าจอคอมพิวเตอร์เช่น ภาพที่ใช้ในเว็บไซต์ต่างๆ อีกทั้งภาพที่เป็นแบบจำลอง RGB นี้ยังใช้ในการจัดเก็บไฟล์ภาพในคอมพิวเตอร์อีกด้วย



ภาพที่ 2-33 มุมสีดำในแบบจำลองสี RGB



ภาพที่ 2-34 มุมสีขาวในแบบจำลองสี RGB

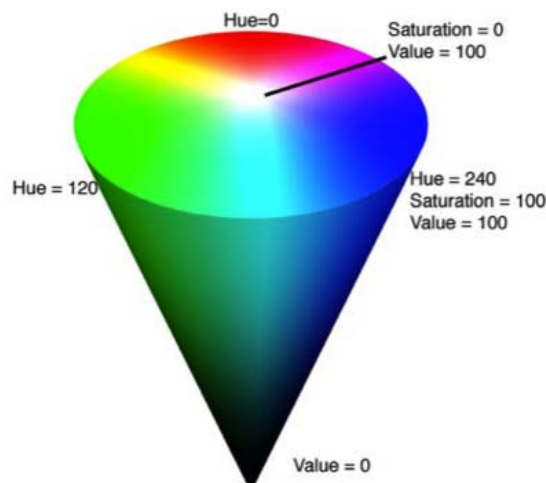
2.5.4.2 แบบจำลองสี HSV

HSV (Hue, Saturation, Value) Color Model ถูกเสนอโดย A.R. Smith (1978) มีวัตถุประสงค์เพื่อให้สะดวกในการใช้สีต่างๆ มากกว่าที่ใช้เฉพาะแม่สีทั้งสามสี HSV เป็นแบบจำลองสีที่ถูกสร้างขึ้นเพื่อเป็นทางเลือกโดยแบบจำลองสี HSV จะมีลักษณะเป็นกรวยสี ดังรูป 2-35 ในแบบจำลองสี HSV เมื่อก้าวถึงสีเหลืองในทางศิลปะจะมีความแตกต่างกัน เมื่อพิจารณาสีเหลืองอ่อน สีเหลืองเข้ม หรือสีน้ำตาลว่ามีความแตกต่างกันอย่างไร จะพบว่าทุกสีคือสีเหลือง ที่มีระดับความเข้มหรือมีความอิ่มตัวที่ต่างกัน ดังนั้นสีในแบบจำลองสี HSV จึงให้ความรู้สึกที่เข้าใจได้แบบจำลองสีนี้ประกอบด้วยค่า 3 ค่า ได้แก่

H หมายถึง Hue คือค่าของสี เช่น สีแดง สีเหลือง สีเขียว วัดเป็นมุมจาก 0 – 360 องศา ซึ่งสีแดง สีเหลือง และสีเขียวจะมีค่าต่างกันสีละ 60 องศา

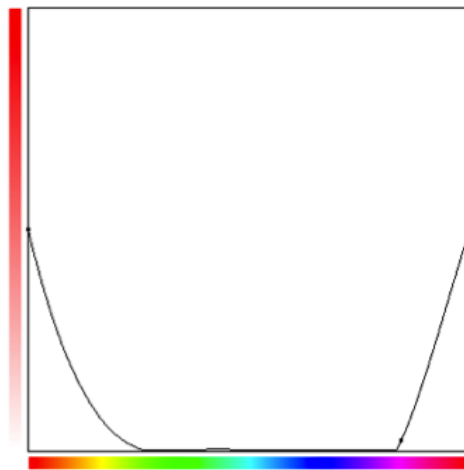
S หมายถึง Saturation คือค่าความอิ่มตัวของเนื้อสีหรือค่าความบริสุทธิ์ของสี ซึ่งมีค่าตั้งแต่ 0 ถึง 100 โดยสีจะมีความอิ่มตัวมากขึ้นเรื่อยๆ เมื่อค่า Saturation มีค่าเพิ่มขึ้น

V หมายถึง Value หรือ Brightness คือ ความสว่างของสีมีค่าตั้งแต่ 0 ถึง 100 โดยภาพจะสว่างมากขึ้นเรื่อยๆ เมื่อ Brightness มีค่าเพิ่มขึ้นเรื่อยๆ เช่น Hue ใดๆ ที่มีค่า Saturation เท่ากับ 0 เมื่อ Value สูงสุดก็คือสีขาว และ Value ต่ำสุดจะได้สีดำ หรือ Hue สีเหลืองที่ Saturation มีค่าเท่ากับ 1 เมื่อ Value สูงสุดจะเป็นสีเหลือง และเมื่อ Value ต่ำสุดคือสีดำ



ภาพที่ 2-35 แบบจำลองสี HSV

แบบจำลองสี HSV นั้นง่ายต่อการทำการเลือกสี [9] เมื่อเปรียบเทียบกับแบบจำลอง RGB ทั้งนี้เนื่องจากในแบบจำลองสี RGB นั้นจะมีความอ่อนไหวต่อการเปลี่ยนแปลงของค่าใดค่าหนึ่งของแม่สี ตัวอย่างในการเลือกสี เช่น ในกรณีที่ต้องการเลือกสีที่อยู่ในโทนสีเดียวกันระหว่างเหลืองอ่อน ในแบบจำลอง RGB ที่มีค่าสีเป็น $R=254$, $G=255$ และ $B=200$ และในแบบจำลอง HSV จะมีค่าสีเป็น $H = 61$ องศา, $S=22\%$ และ $V =100\%$ เมื่อต้องการเปลี่ยนจากสีเหลืองอ่อนไปเป็นสีเขียวอ่อนที่มีระดับความอิ่มสีเท่ากับสีเหลืองอ่อน ในแบบจำลองสี RGB จำเป็นต้องมีการเปลี่ยนค่าทั้ง 3 ค่าเพื่อใช้ในการผสมสีให้ได้สีเขียวอ่อน ได้แก่ $R=199$, $G=255$ และ $B=199$ แต่ใน แบบจำลองสี HSV เมื่อต้องการเลือกสีที่มีค่าความอิ่มสีเดียวกันกับสีเหลืองอ่อน ค่าความสว่างเท่ากันกับสีเหลืองอ่อน แต่ค่าสีที่ต่างกันอย่างเช่นในกรณีนี้ การเลือกสีเขียวอ่อนจะทำได้โดยการเปลี่ยนค่า H หรือ Hue เท่านั้นซึ่งค่าสีเขียวอ่อนในแบบจำลองสี HSV จะมีค่าคือ $H=120$ องศา , $S=22\%$ และ $V=100\%$ จะเห็นได้ว่า HSV นั้นเป็นแบบจำลองสีที่ทำให้ง่ายต่อการเลือกสีหรือแบ่งแยกความต่างของสีได้จากความรู้สึกของผู้ใช้จึงได้มีการใช้งานอย่างแพร่หลายในโปรแกรมที่ใช้ในการแต่งภาพดังตัวอย่างในภาพ 2-36 เป็นผลลัพธ์ที่ได้จากการปรับ HSV Curve แล้วตามในรูปที่ 2-37 ซึ่งเป็นการเลือกเฉพาะส่วนของภาพที่มีสีแดงให้เด่นกว่าสีอื่นๆ



ภาพที่ 2-36 HSV Curve



ภาพที่ 2-37 แสดงภาพต้นฉบับและผลลัพธ์ที่ได้จากการปรับค่า HSV

จากตัวอย่างที่ผ่านมาจะเห็นได้ว่าแบบจำลองสี HSV นั้นจะมีส่วนช่วยให้การจัดการเกี่ยวกับสี การเลือกสี การจับคู่สี ทำได้อย่างสะดวกและให้ผลลัพธ์ตามที่ต้องการ แต่อย่างไรก็ตามการเลือกใช้แบบจำลองสี HSV นี้จำเป็นต้องระวังเนื่องจากสีเทาจากสีดำไล่สีไปยังสีขาวมีค่า Hue และ Saturation เป็น 0 ในบางครั้งอาจจะทำให้เกิดปัญหาได้หากต้องการใช้สีบางสีที่อยู่ในส่วนนี้ โดยหลักการแล้วสีที่มีความอิ่มลีน้อยจะถูกมองเป็นเฉดสีเทา เมื่อทำการปรับปรุงภาพในส่วนที่มีค่า Saturation ต่ำอาจส่งผลให้สูญเสียของข้อมูลในส่วนนี้ได้ ซึ่งปัญหาการสูญเสียข้อมูลนี้จะเกิดจากการบีบอัดข้อมูลเพื่อเขียนไฟล์หรือการแสดงผล สำหรับการใช้งานของ HSV นั้นจำเป็นอย่างยิ่งที่ต้องมีการแปลงเป็น RGB ในกรณีที่ต้องการแสดงผลผ่านหน้าจอคอมพิวเตอร์เนื่องด้วยระบบปฏิบัติการจะรู้จักกับแบบจำลองสี RGB ทั้งนี้เพื่อการแสดงผลที่ถูกต้อง

2.5.5 การแปลงค่าสีจากแบบจำลองสี RGB ไปแบบจำลองสี HSV

เนื่องด้วยค่าสีต่างๆที่ถูกใช้ในไฟล์ภาพที่ในคอมพิวเตอร์ส่วนจัดเก็บอยู่ในรูปแบบของแบบจำลองสี RGB ดังนั้นเมื่อมีความจำเป็นต้องใช้ค่าสีตามแบบจำลองสี HSV จะต้องทำการแปลงค่าสีจาก RGB ไปเป็น HSV โดยเริ่มต้นด้วยการดูค่าสีในแต่ละ Channel ของ RGB หากค่าสีในแต่ละ Channel ของ RGB เป็นค่าที่อยู่ระหว่าง 0-255 ให้ทำการหารค่าดังกล่าวด้วย 255 เพื่อให้ได้เป็นค่าที่อยู่ระหว่าง 0-1 ซึ่งเป็นค่าระดับสีของ Hue เพื่อเตรียมพร้อมในการแปลงค่าสีต่อไปตัวอย่างค่าระดับสีต่างๆ ของแบบจำลองสี HSV แสดงในตารางที่ 2-8

ตารางที่ 2-8 แสดงค่าระดับสีใน HSV color model

สี (Color)	ค่าสี (Hue) ใน HSV
แดง	0
เหลือง	0.1667
เขียว	0.333
น้ำเงิน	0.6667

ในการแปลงค่าจะมีข้อกำหนดดังต่อไปนี้

$$V = \max\{R, G, B\} \quad (2.5)$$

$$\delta = V - \min\{R, G, B\} \quad (2.6)$$

$$S = \frac{\delta}{V} \quad (2.7)$$

จากสมการที่ (2.5) จะเป็นการหาค่า V ซึ่งก็คือค่า Value โดยที่เลือกจากค่าสูงสุดของค่าที่มีอยู่ใน RGB จากนั้นทำการคำนวณโดยใช้สมการที่ (2.7) จะเป็นการหาค่า S ซึ่งก็คือค่า Saturation โดยที่ต้องมีการคำนวณค่า δ จากสมการที่ (2.6) ในขั้นตอนต่อไปจะเป็นการหาผลลัพธ์ค่าสี H หรือ Hue จำเป็นต้องมีการพิจารณาตามเงื่อนไขในสมการ (2.8), (2.9) และ (2.10) ดังต่อไปนี้

$$H = \frac{1}{6} \left(\frac{G-B}{\delta} \right) \quad \text{ถ้า } R=V \quad (2.8)$$

$$H = \frac{1}{6} \left(2 + \frac{B-R}{\delta} \right) \quad \text{ถ้า } G=V \quad (2.9)$$

$$H = \frac{1}{6} \left(4 + \frac{R-G}{\delta} \right) \quad \text{ถ้า } B=V \quad (2.10)$$

ในขั้นตอนสุดท้าย H จะมีค่าติดลบเราต้องทำการเพิ่มค่า H ด้วย 1 ถ้าในกรณีที่ค่าใน (R, G, B) มีค่า $(0, 0, 0)$ และในกรณีที่ $V = \delta = 0$ จะต้องกำหนดให้ (H, S, V) มีค่าเป็น $(0, 0, 0)$ เช่นกัน ตัวอย่างในการคำนวณค่าสี

$$\text{กำหนดค่าสีตัวอย่าง } (R, G, B) = \{0.3, 0.5, 0.6\}$$

$$V = \max\{0.3, 0.5, 0.6\}$$

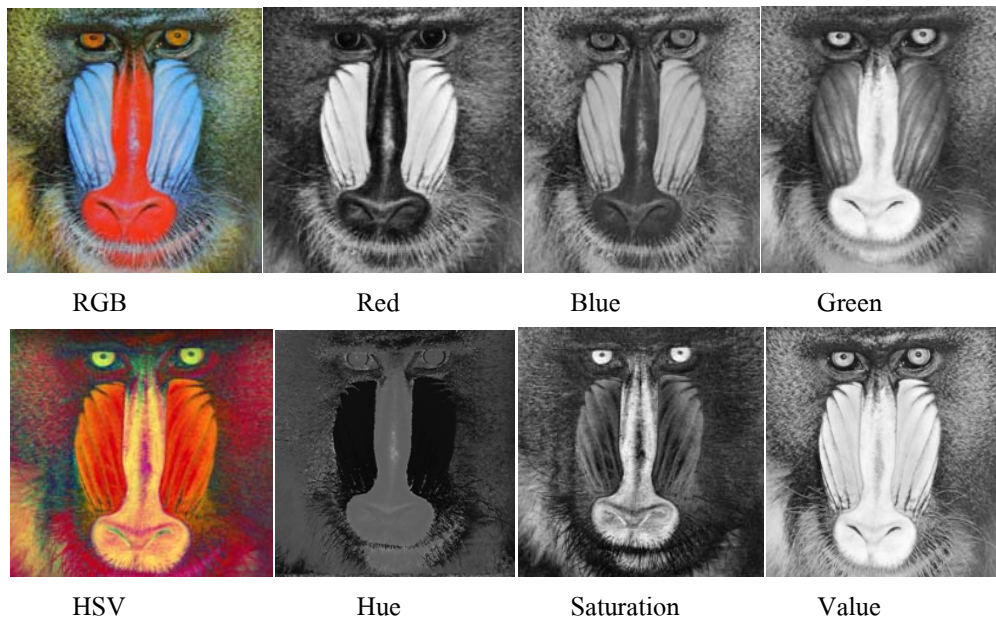
$$\delta = V - \min\{0.3, 0.5, 0.6\} = 0.6 - 0.3 = 0.3$$

$$S = \frac{0.3}{0.6} = 0.50$$

เนื่องด้วย ค่า $B = V$ จะมีการคำนวณค่า H ดังนี้

$$H = \frac{1}{6} \left(4 + \frac{0.3 - 0.5}{0.3} \right) = 0.658$$

ตัวอย่างภาพที่ได้ในแบบจำลองสี RGB และ HSV แสดงดังภาพที่ 2-38



ภาพที่ 2-38 ตัวอย่างภาพในแบบจำลองสี RGB และ แบบจำลองสี HSV

2.5.6 การแปลงค่าสีจากแบบจำลองสี HSV ไปแบบจำลองสี RGB

เพื่อการแสดงผลสีที่ถูกต้องบนหน้าจอคอมพิวเตอร์ แบบจำลองสี RGB เป็นสิ่งที่จำเป็นทั้งนี้เพื่อการแสดงผลที่ถูกต้องขั้นตอนในการแปลงค่าสีจากแบบจำลองสี HSV ไปเป็นแบบจำลองสี RGB นั้นทำได้ดังต่อไปนี้ กำหนดให้ H มีค่าตั้งแต่ 0° ถึง 360° ดังในสมการที่ (2.11) กำหนดให้ค่า S_{HSV} มีค่าระหว่าง 0 ถึง 1 ดังในสมการที่ (2.12) และค่า V มีค่าตั้งแต่ 0 ถึง 1 ดังสมการที่ (2.13) จากนั้นทำการคำนวณค่า C ซึ่งเป็นค่าโครมา (Chroma) แล้วให้ทำการคำนวณหาค่า H' ของสีนั้นโดยใช้สมการที่ (2.15) เมื่อได้ค่า H' แล้วทำการหาค่า X ซึ่งเป็นค่าสูงสุดที่เป็นไปได้ของสีนั้นจากสมการที่ (2.16) คำนวณค่า m โดยใช้สมการที่ (2.18) จากนั้นทำการเลือกค่า R', G', B' ตามเงื่อนไขของ H' ในสมการที่ (2.17) เพื่อให้ได้ค่าที่จะใช้ในการเพิ่มค่าสีให้กับสีใน (R, G, B) เพื่อให้ได้ค่าสีที่ถูกต้องตามแบบจำลองสี RGB

$$H \in [0^\circ, 360^\circ) \quad (2.11)$$

$$S_{HSV} \in [0, 1] \quad (2.12)$$

$$V \in [0, 1] \quad (2.13)$$

$$C = V \times S_{HSV} \quad (2.14)$$

$$H' = \frac{H}{60^\circ} \quad (2.15)$$

$$X = C(1 - |H' \bmod 2 - 1|) \quad (2.16)$$

$$(R', G', B') = \begin{cases} (0, 0, 0) & \text{if } H \text{ is undefined} \\ (C, X, 0) & \text{if } 0 \leq H' < 1 \\ (X, C, 0) & \text{if } 1 \leq H' < 2 \\ (0, C, X) & \text{if } 2 \leq H' < 3 \\ (0, X, C) & \text{if } 3 \leq H' < 4 \\ (X, 0, C) & \text{if } 4 \leq H' < 5 \\ (C, 0, X) & \text{if } 5 \leq H' < 6 \end{cases} \quad (2.17)$$

$$m = V - C \quad (2.18)$$

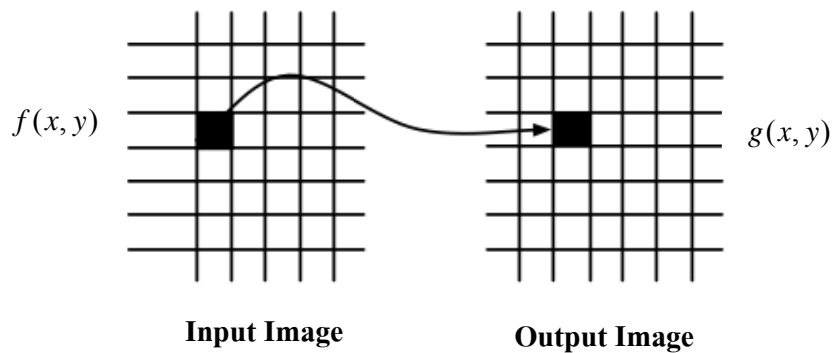
$$(R, G, B = (R' + m, G' + m, B' + m) \quad (2.19)$$

2.6 ประเภทของการประมวลผลภาพ (Image processing)

การประมวลผลภาพดิจิทัลสามารถแบ่งออกเป็น 3 ประเภทดังต่อไปนี้

2.6.1 การกระทำกับจุดของภาพ (Point Processing)

การประมวลผลภาพประเภทนี้จะเป็นการปรับค่าระดับเทาของในแต่ละพิกเซลของภาพโดยจะเป็นการดำเนินการจุดต่อจุดดังแสดงในภาพที่ 2-39 ซึ่งจะเป็นการกระทำเชิงพีชคณิต ซึ่งได้ บวก ลบ คูณ หาร หรือการกระทำทางตรรกศาสตร์ต่างๆ กับจุดในภาพต้นฉบับเพื่อให้ได้ผลลัพธ์ที่ต้องการโดยทั่วไปการกระทำกับจุดภาพได้แก่ การปรับความเข้มของสีหรือสว่างของภาพดิจิทัล เป็นต้น



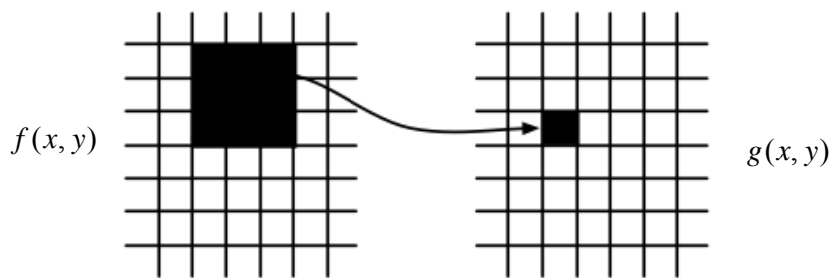
ภาพที่ 2-39 การกระทำต่อจุดภาพ (Point Processing)

$$g(x_i, y_i) = P[f(x_i, y_i)] \quad (2.20)$$

จากสมการที่ (2.20) กำหนดให้ $f(x_i, y_i)$ เป็นภาพที่นำเข้า (Input Image) และ $g(x_i, y_i)$ เป็นภาพผลลัพธ์ (Output Image) เมื่อ P เป็นการกระทำใดๆ ต่อ $f(x_i, y_i)$ เพื่อให้ได้ผลลัพธ์ที่ต้องการ

2.6.2 การกระทำกับภาพเฉพาะส่วน (Local Processing)

สำหรับการประมวลผลภาพแบบนี้เป็นการกระทำกับค่าระดับเทาในจุดของภาพ ผลลัพธ์จะขึ้นอยู่กับค่าระดับความเข้มของระดับเทาในบริเวณข้างเคียงของภาพต้นฉบับ ดังแสดงในภาพที่ 2-40 โดยที่กำหนดให้ $f(x, y)$ เป็นภาพต้นฉบับ และ $g(x, y)$ เป็นภาพผลลัพธ์ ซึ่งจะมีค่าดังแสดงในสมการที่ (2.21)



ภาพที่ 2-40 การกระทำกับภาพเฉพาะส่วน (Local Processing)

$$O(x, y) = T[\text{neighborhood of } I(x, y)] \quad (2.21)$$

ในกระบวนการกระทำกับภาพเฉพาะบริเวณที่นิยมคือ การกรองสัญญาณ ในสเปซเชิงโดเมนโดยที่มีชื่อเรียกว่า การคอนโวลูชัน (Convolution) คอนโวลูชันเป็นขั้นตอนของการกรองสัญญาณแบบ linear filtering ซึ่งจะมีการกำหนดมาสก์ (Mask) ที่ใช้ในการกรองสัญญาณภาพ โดยเมื่อนำมาสก์มาเป็นตัวกำหนดพื้นที่ที่จะทำการคำนวณ โดยที่ทำการคูณค่าในตำแหน่งที่อยู่ในตำแหน่งเดียวกันกับมาสก์ จากนั้นทำการบวกรวมค่าในหลายๆ ตำแหน่ง ตัวอย่างการคอนโวลูชัน (Convolution)

กำหนดเมตริกซ์ A เป็นเมตริกซ์ของภาพแสดงในสมการ (2.22)

$$A = \begin{bmatrix} 1 & 2 & 3 & -1 \\ 6 & -5 & 4 & 3 \\ -2 & 4 & 6 & 7 \\ 5 & 0 & 1 & 3 \end{bmatrix} \quad (2.22)$$

กำหนดให้ B เป็นมาสค์ ที่ใช้เป็น Convolution kernel ดังแสดงในสมการที่ (2.23)

$$B = \begin{bmatrix} 1 & -3 & 5 \\ -2 & 4 & 6 \\ 7 & 8 & -9 \end{bmatrix} \quad (2.23)$$

$$C = \begin{bmatrix} -9 & 8 & 7 \\ 6 & 4 & -2 \\ 5 & -3 & 1 \end{bmatrix} \quad (2.24)$$

ในการคอนโวลูชันเริ่มด้วยการหมุน (Rotate) เมตริกซ์ B ไป 180° ซึ่งจะทำให้ได้เมตริกซ์ C ดังสมการที่ (2.24) จากนั้นทำการหาจุดกึ่งกลางของเมตริกซ์ C ซึ่งตำแหน่งศูนย์กลางของเมตริกซ์ C จะอยู่ที่ (2,2) จากนั้นนำ เมตริกซ์ C วางทับไว้บนเมตริกซ์ A โดยที่เริ่มที่ตำแหน่ง (1,1) ให้ตำแหน่ง (2,2) ของ C ตรงกับตำแหน่ง (1,1) ของเมตริกซ์ A จากนั้นทำการคูณค่าที่อยู่ซ้อนทับ แล้วทำการนำผลคูณรวมกันจะได้เป็นค่า Convolution ของตำแหน่ง (1,1) ของ เมตริกซ์ A ซึ่งคือค่าในตำแหน่ง (1,1) ของ D ซึ่งได้จาก $D(1,1) = (1)(4) + (2)(-2) + (6)(-3) + (-5)(1) = -23$ จากนั้นทำการเลื่อนไปยังตำแหน่งต่อไปของเมตริกซ์ A ในตำแหน่ง (1,2) ซึ่งต้องดำเนินการไปเรื่อยๆจนครบทั้งภาพ ซึ่งตัวอย่างผลลัพธ์ที่ได้จากการทำคอนโวลูชันระหว่างเมตริกซ์ A และ เมตริกซ์ B ซึ่งแสดงดังสมการที่ (2.25)

$$D = \begin{bmatrix} -23 & 57 & -8 & 25 \\ 66 & 8 & 24 & -50 \\ -18 & -24 & 78 & -6 \\ 32 & 36 & -25 & 104 \end{bmatrix} \quad (2.25)$$

เนื่องจากพื้นที่ของมาสค์ในการทำคอนโวลูชันมีการวางโดยค้ำนึ่งถึงจุดกึ่งกลางของมาสค์ดังนั้นในกรณีที่กระทำบริเวณของขอบภาพซึ่งจะต้องมีการมองข้ามส่วนของขอบโดยทำการแทนค่าบริเวณของมาสค์ที่ไม่มีการซ้อนทับด้วย 0 และมีการคำนวณเฉพาะในส่วนที่ซ้อนทับเท่านั้นซึ่งกระบวนการนี้เรียกว่าการทำ Zero Padding ตัวอย่างการทำ Zero Padding ดังภาพที่ 2-41

0	0	0			
0					
0					

ภาพที่ 2-41 แสดงการทำ Zero Padding

2.6.3 การกระทำกับภาพทั้งหมด (Global Processing)

การประมวลผลภาพประเภทนี้ ค่าระดับเทาของภาพในจุดของผลลัพธ์จะขึ้นอยู่กับค่าของระดับเทาของทั้งภาพลักษณะของกระบวนการนี้ ได้แก่ การทำเทรชโฮลดิ้ง (Thresholding) และการทำ Histogram Equalization ซึ่งในการกระทำกับภาพทั้งหมด (Global Processing) เริ่มต้นด้วยการกำหนดให้ $I(x, y)$ เป็นภาพที่นำเข้าและ $O(x, y)$ เป็นผลลัพธ์ที่ได้ซึ่งค่าของจุดภาพของ $O(x, y)$ มีค่าดังสมการที่ (2.26) โดยที่ค่าใน $f(x, y)$ จะใช้ร่วมกันทั้งภาพ

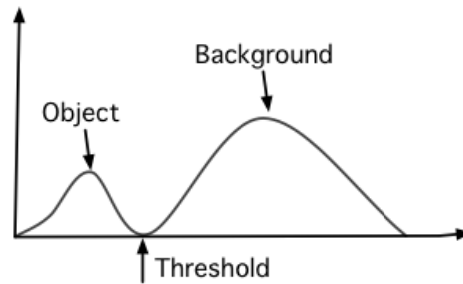
$$O(xy) = G [f(x, y)]$$

2.26)

ในการกำหนดเทรชโฮลด์ (Thresholding) เทคนิคนี้ใช้ในการแบ่งแยกวัตถุ (Object) ที่สนใจกับพื้นหลัง (Background) ซึ่งขั้นตอนนี้เป็นขั้นตอนที่สำคัญที่ใช้ในการสร้างภาพไบนารี โดยที่ค่าเทรชโฮลด์จะมีผลต่อภาพผลลัพธ์ในกรณีที่กำหนดค่าที่น้อยหรือมากเกินไปจะทำให้สูญเสียรายละเอียดบางส่วนของวัตถุที่สนใจ และอีกทั้งยังส่งผลให้มีสัญญาณรบกวน (Noise) ซึ่งโดยทั่วไปการกำหนดเทรชโฮลด์จะพิจารณาจากค่าจากฮิสโตแกรมของภาพ

2.6.3.1 การหาค่าเทรชโฮลด์หนึ่งระดับ (Single Thresholding)

เป็นการแบ่งข้อมูลตามการกระจายของข้อมูลเมื่อพิจารณาจาก ฮิสโตแกรม ซึ่งการเลือกค่าเทรชโฮลด์จะใช้ค่าที่ต่ำสุดที่อยู่ระหว่างกลุ่มข้อมูลสองกลุ่ม โดยค่าที่เลือกจะต้องเหมาะสมที่สุดดังแสดงในภาพที่ 2-42 เพื่อให้ภาพผลลัพธ์ที่ได้ปราศจากสัญญาณรบกวน



ภาพที่ 2-42 การเลือกค่าเทรชโวลด์จากภาพฮิสโตแกรม

เมื่อมีการสร้างฮิสโตแกรมจากภาพระดับเทาของภาพต้นฉบับซึ่งจะได้ข้อมูลทั้งเส้น 2 กลุ่มคือข้อมูลที่จะเป็นข้อมูลที่เป็นวัตถุ จากนั้นทำการเลือกค่าที่จะทำการเทรชโวลด์โดยมีการกำหนดค่า (T) ระดับความเข้มเทาที่ใช้ในการเทรชโวลด์ ดังสมการที่ (2.26) ที่ซึ่ง $f(x, y)$ เป็นระดับเทาในตำแหน่ง (x, y) และ $p(x, y)$ เป็นค่าระดับเทาที่อยู่กึ่งกลางของกลุ่มข้อมูลในตำแหน่ง (x, y) สำหรับภาพผลลัพธ์ $g(x, y)$ ที่ได้จะมีค่าดังสมการที่ (2.28) โดยที่ส่วนที่เป็นวัตถุจะมีค่าเป็น 1 และ 0 แทนค่าของภาพที่เป็นพื้นหลัง

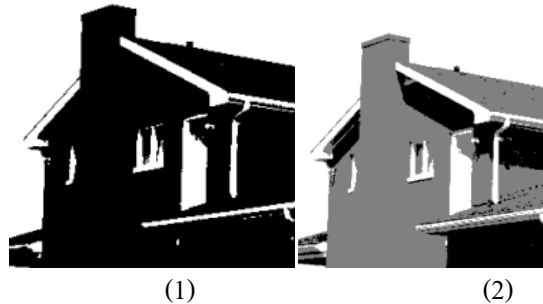
$$T = T [x, y, (p(x, y), f(x, y))] \quad (2.27)$$

$$g(x, y) = \begin{cases} 1 & \text{if } f(x, y) > T \\ 0 & \text{if } f(x, y) \leq T \end{cases} \quad (2.28)$$

2.6.3.2 การหาค่าเทรชโวลด์หลายระดับ (Multilevel Thresholding)

ในการทำการเทรชโวลด์แบบนี้มีการกำหนดค่าที่ใช้ในการทำ เทรชโวลด์มากกว่า 1 ค่า โดยขั้นตอนในการทำเริ่มจากการกำหนดระดับในเทรชโวลด์ตามระดับที่สนใจ แทนด้วย (T_1, T_2) โดยค่าดังกล่าวเป็นค่าที่ได้จากการพิจารณาฮิสโตแกรม ทำการกำหนดให้ $I(x, y)$ คือภาพต้นฉบับ และ $O(x, y)$ เป็นภาพผลลัพธ์โดยได้จากการพิจารณาค่าตามเงื่อนไขที่ระบุในสมการที่ (2.29) ซึ่งภาพผลลัพธ์แสดงดังภาพที่ 2-43 เห็นได้ว่าในภาพ (1) จะเป็นการกำหนดค่าเทรชโวลด์ระดับเดียวจะได้วัตถุที่มีสีดำและสีขาวเท่านั้นส่วนในภาพ (2) จะเป็นการกำหนดค่าเทรชโวลด์หลายระดับจากผลลัพธ์ จะมีระดับเทาอยู่ 2 ระดับบนวัตถุ และพื้นหลังที่เป็นสีขาว

$$O(x, y) = \begin{cases} 1 & \text{if } I(x, y) < T_1 \\ I(x, y) & \text{if } T_1 \leq I(x, y) \leq T_2 \\ 1 & \text{if } I(x, y) > T_2 \end{cases} \quad 2.29$$



ภาพที่ 2-43 ภาพผลลัพธ์จาก Single Thresholding และ Multilevel Thresholding

2.7 มอร์โฟโลยี (Mophology)

มอร์โฟโลยีเป็นคือทฤษฎีเซตซึ่งมีประสิทธิภาพและประโยชน์มากในการประมวลผลภาพ ดิจิตอลมอร์โฟโลยีใช้ในการประมวลผลที่เกี่ยวกับการจัดการรูปร่างและโครงสร้างเพื่อใช้ในการแยกองค์ประกอบของภาพเพื่อการอธิบายรูปร่างหรือบริเวณที่อยู่ในภาพนอกจากนั้นมอร์โฟโลยีนั้นยังมีประโยชน์ในการทำ Pre-processing และ Post-processing [10] ได้แก่การนำมาใช้ในการทำฟิลเตอร์ต่างๆ อีกด้วย ในภาพไบนารี (Binary Image) เซตจะหมายถึงสมาชิกในสเปซ 2 มิติ (Z^2) โดยที่ในแต่ละสมาชิกของเซตคือเวกเตอร์ 2 มิติ ซึ่งก็คือที่ตำแหน่งพิกัด (x, y) ของ Pixel สีดำในภาพ ในกรณีที่ภาพเป็นภาพระดับเทาซึ่งเซตจะมีองค์ประกอบอยู่ใน Z^3 ซึ่งจะมีสมาชิกที่เพิ่มขึ้นมาคือค่าความเข้มซึ่งในเซตที่มีค่ามิติที่สูงขึ้นก็จะมีคุณสมบัติอื่นๆ เช่น องค์ประกอบของภาพที่เปลี่ยนตามเวลาในงานวิจัยนี้ได้มีการใช้มอร์โฟโลยี ซึ่งประกอบด้วยขบวนการดังต่อไปนี้

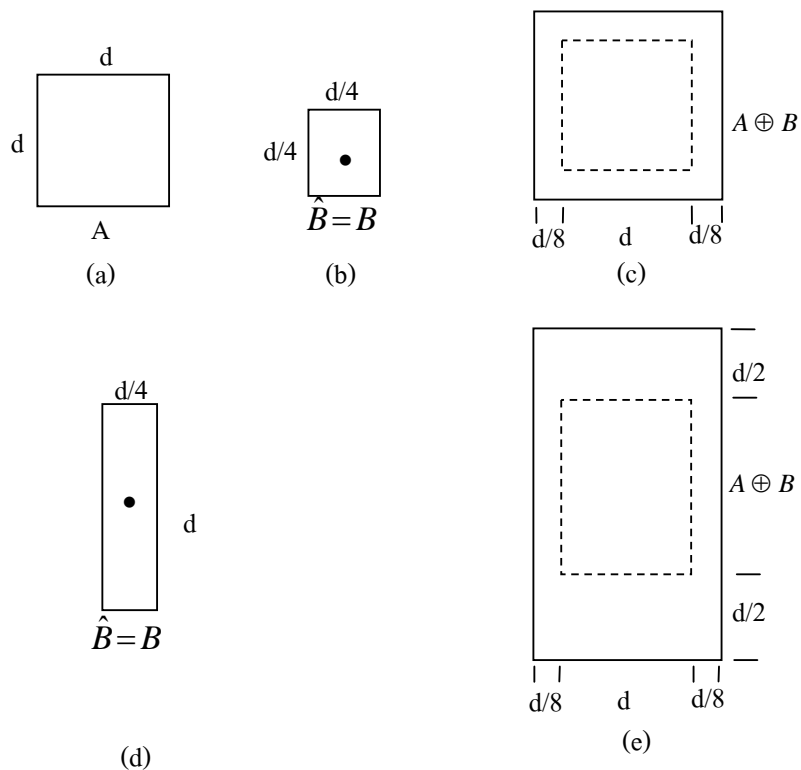
2.7.1 การขยายภาพ (Dilation)

วิธีการประมวลผลภาพนี้เป็นกระบวนการที่ช่วยขยายโครงสร้างและองค์ประกอบของภาพโดยใช้หน่วยโครงสร้าง (Element Structure) ในการทำขยายภาพจะเริ่มด้วยการกำหนดให้เซต A และ B เป็นเซตใน Z^2 และให้ \ominus คือ เซตว่าง ในที่นี้ B คือชื่อหน่วยโครงสร้าง (Element Structure) ในการขยายของ A ด้วย B จะแทนด้วย $A \oplus B$ ซึ่งมีคำจำกัดความดังสมการที่ (2.30) ในการขยายภาพจะประกอบด้วยการทำงานสะท้อน ของ B รอบจุดกำเนิดจากนั้นทำการเลื่อนผลลัพธ์

จากการสะท้อนด้วยระยะทาง x การขยายของ A และ B คือเซตของระยะ x ที่เคลื่อนที่ทั้งหมดโดยที่ \hat{B} และ A โอเวอร์แลปอย่างน้อยหนึ่งสมาชิกที่ไม่เป็น 0 คือสมการที่ (2.31) ในการขยายภาพจะมีลักษณะคล้ายการทำคอนโวลูชัน ซึ่งจะมีการใช้หน่วยโครงสร้างวิ่งไปทั่วทั้งภาพต้นฉบับ

$$A \oplus B = \{x / (\hat{B})_x \cap A \neq \emptyset\} \quad (2.30)$$

$$A \oplus B = \{x / [(\hat{B})_x \cap A] \subseteq A\} \quad (2.31)$$



ภาพที่ 2-44 ตัวอย่างการขยายภาพ (Dilation)

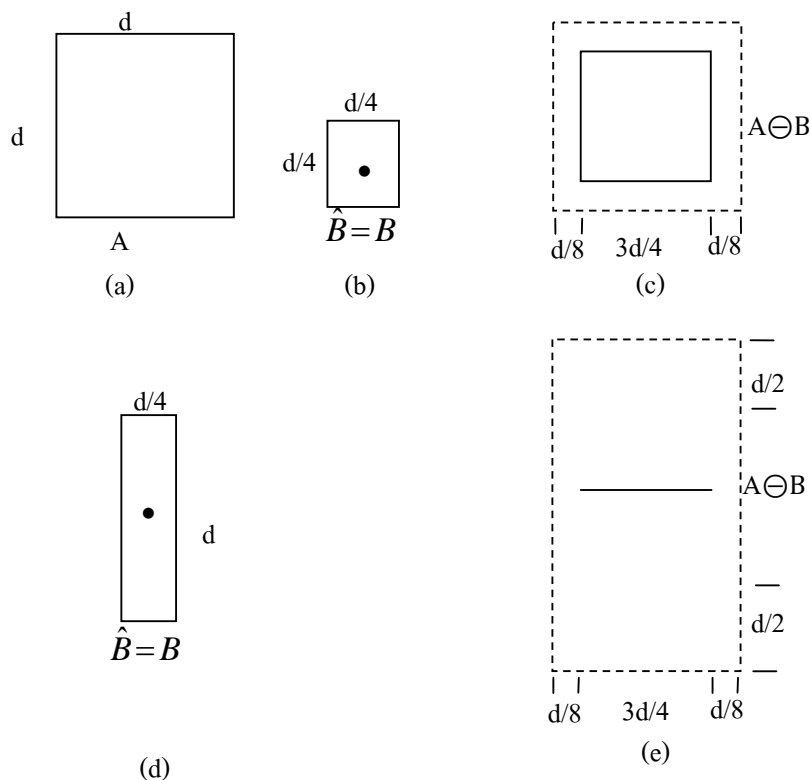
จากภาพที่ 2-44 (a) แสดงเซตดั้งเดิม เมื่อมีการใช้หน่วยโครงสร้างรูปสี่เหลี่ยม ในภาพที่ 2-44(b) กับเซตดั้งเดิมจะได้ผลลัพธ์ในภาพที่ 2-44(c) ซึ่งเส้นประแสดงเซตดั้งเดิมเพื่อใช้ในการอ้างอิงและเส้นทึบแสดงเซตแนวซึ่งเมื่อการเคลื่อนที่ของ \hat{B} ด้วยระยะทาง x ที่เกินจากเซตทำให้เกิดซ้อนทับกันระหว่าง \hat{B} และ A มีค่า 0 จากภาพที่ 2-44 (e) จะเป็นภาพผลลัพธ์ที่เกิดจากหน่วยโครงสร้างจากภาพที่ 2-44 (ก) ที่มีคุณสมบัติในการขยายในแนวดิ่งมากกว่าแนวนอน

2.7.2 การเซาะภาพ (Erosion)

การเซาะภาพเป็นขั้นตอนการทำตรงข้ามกับการขยายภาพ (Dilation) ซึ่งผลลัพธ์ของการทำการเซาะภาพจะได้โครงสร้างภาพที่มีลักษณะมีลักษณะหดลง ทำให้เห็นโครงสร้างของภาพได้ชัดเจน ในการทำการเซาะภาพเซต A และ B เป็นสมาชิกของเซต Z^2 เมื่อทำการเซาะภาพ A ด้วยหน่วยโครงสร้าง B จะถูกแทนด้วย $A \ominus B$ ซึ่งมีคำจำกัดความดังสมการที่ (2.32)

$$A \ominus B = \{x / (\hat{B})_x \subseteq A\} \quad (2.32)$$

จากสมการที่ (2-32) คือ การเซาะภาพ A โดยใช้หน่วยโครงสร้าง B ได้จากเซตของทุกๆ จุดที่ได้จากการเคลื่อนย้าย B ด้วยระยะ x ภายในภาพ A ดังตัวอย่างในภาพที่ 2-45 ในเซต A แทนด้วยเส้นประในภาพที่ 2-45 (c) ใช้ในการอ้างอิง ส่วนเส้นทึบแสดงถึงเซตแนวที่ใช้ในการเคลื่อนที่ของ \hat{B} ไปเป็นระยะ x ที่เกินขอบเขตนี้ถือว่าเซตไม่อยู่ใน A ในภาพที่ 2-45 (e) เป็นผลลัพธ์ที่ได้จากการเซาะด้วยหน่วยโครงสร้างที่อยู่ในภาพที่ 2-45 (d)



ภาพที่ 2-45 ตัวอย่างการเซาะภาพ (Erosion)

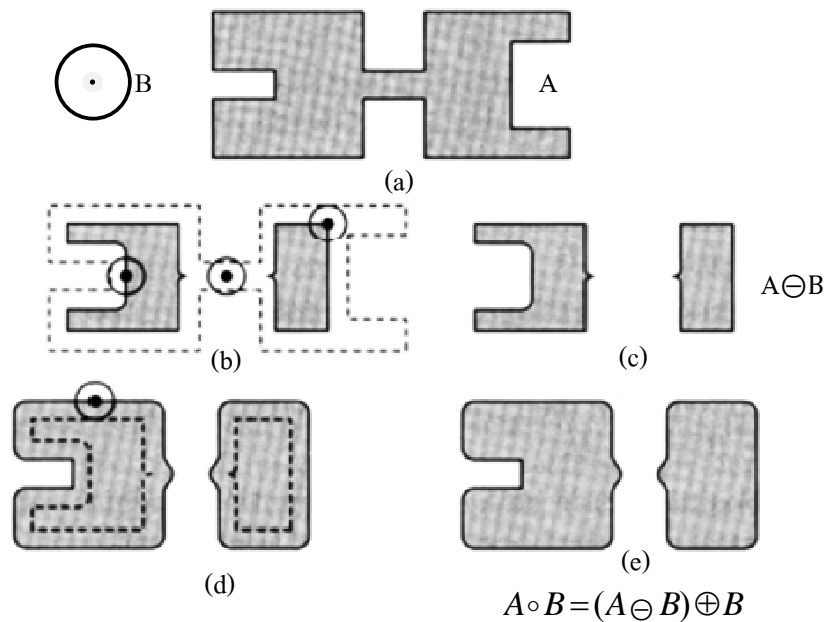
2.7.3 การเปิดภาพ (Opening) และการปิดภาพ (Closing)

การเปิดภาพ (Opening) เป็นกระบวนการที่ช่วยทำให้เส้นรูปร่างของวัตถุมีลักษณะเรียบขึ้น กำจัดส่วนที่เป็นช่องแคบระหว่างวัตถุทำให้วัตถุแยกออกจากกัน ส่วนการปิดภาพ (Closing) จะเป็นกระบวนการที่ตรงกันข้ามกับการเปิด ซึ่งการปิดภาพจะช่วยเติมเต็มช่องว่างเล็กๆ ระหว่างวัตถุ ทำให้วัตถุในภาพมีลักษณะที่เชื่อมต่อกันในส่วนของช่องแคบต่างๆ ของภาพจะถูกเติมเต็มทำให้ได้วัตถุที่มีความสมบูรณ์มากยิ่งขึ้น การเปิดจะมีคำจำกัดความดังสมการที่ (2.33) และการปิดจะมีคำจำกัดความดังสมการที่ (2.34)

$$A \circ B = (A \ominus B) \oplus B \quad (2.33)$$

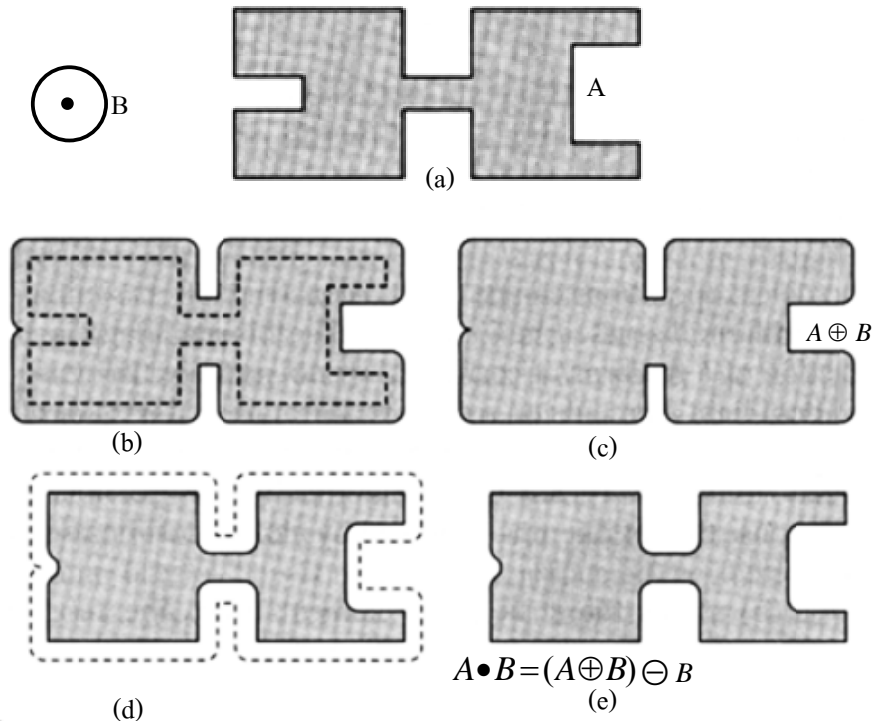
$$A \bullet B = (A \oplus B) \ominus B \quad (2.34)$$

จากสมการที่ (2.33) คือ การเปิดของเซต A ด้วย B แทนด้วย $A \circ B$ ซึ่งกระบวนการที่เริ่มจากการทำการเซาะ (Erosion) A ด้วย B จากนั้นนำผลลัพธ์ที่ได้มาทำการขยาย (Dilation) ด้วย B ส่วนการปิด (Closing) จากสมการที่ (2.34) ผลลัพธ์การปิด A ด้วย B แทนด้วย $A \bullet B$ ซึ่งกระบวนการเริ่มจากการทำการขยาย (Dilation) A ด้วย B เมื่อได้ผลลัพธ์แล้วให้ทำการเซาะ (Erosion) ด้วย B ซึ่งจะได้ผลลัพธ์ที่ต้องการ



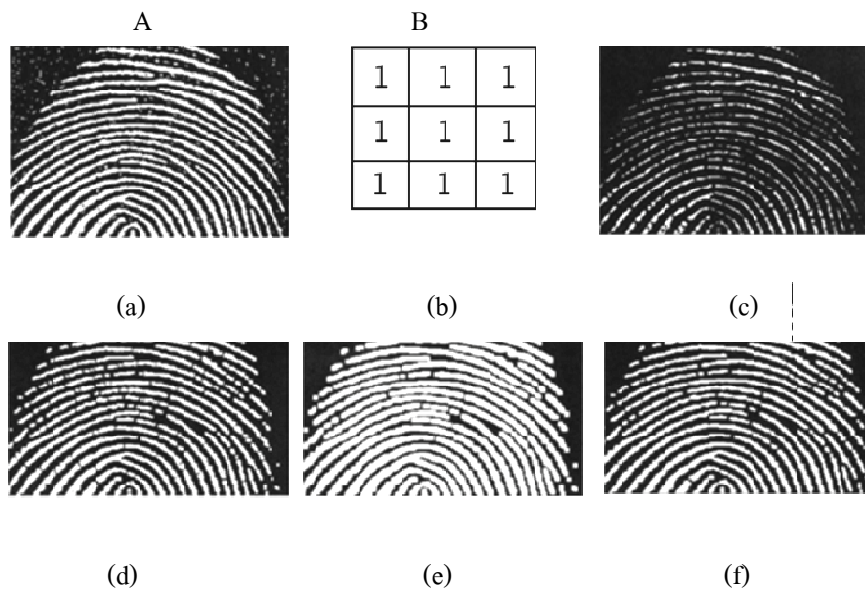
ภาพที่ 2-46 ตัวอย่างการทำการเปิด (Opening)

จากภาพที่ 2-46 (a) แสดงตัวอย่างของการเปิดภาพ โดยให้ A เป็นเซตเริ่มต้น และ B คือหน่วยโครงสร้างที่มีลักษณะเป็นรูปจาน (Disc) ที่มีการระบุจุดศูนย์กลาง ในภาพที่ 2-46 (b) แสดงให้เห็นเส้นรอบวัตถุ A ที่ถูกทำการเซาะด้วย B และได้ผลลัพธ์จากการเซาะดังในภาพที่ 2-46 (c) ต่อมานำผลลัพธ์ที่ได้มาทำการขยายด้วย B ดังแสดงในภาพที่ 2-46 (d) ซึ่งจะทำให้ได้ผลลัพธ์ตามภาพที่ 2-46 (e) จะเห็นได้ว่าภาพผลลัพธ์จะแยกตัวออกจากกันและขอบภาพจะเรียบขึ้น



ภาพที่ 2-47 ตัวอย่างการทำการปิด (Closing)

จากภาพที่ 2-47 แสดงให้เห็นตัวอย่างการทำการปิด โดยเซตตั้งต้นคือ A ในการทำการปิดภาพจะเริ่มด้วยการนำ A มาทำการขยาย (Dilation) ด้วยหน่วยโครงสร้าง B ที่มีลักษณะเป็นจาน (Disc) ซึ่งผลลัพธ์ที่ได้จะได้ดังภาพที่ 2-47 (c) จากนั้นทำการเซาะด้วย B ดังในภาพที่ 2-47 (d) และผลลัพธ์จากการทำการปิดภาพแสดงดังภาพที่ 2-47 (e) จะเห็นได้ว่าผลลัพธ์จะได้วัตถุที่มีขอบภาพที่เรียบและสองวัตถุยังเชื่อมต่อกันอยู่โดยที่การปิดภาพจะมีลักษณะการดำเนินการเหมือนกับการนำลูกบอลมาถลึงรอบวัตถุโดยที่วิ่งอยู่บนเส้นขอบนอก และการเปิดภาพจะเปรียบเสมือนการนำลูกบอลมาถลึงภายในเส้นขอบนอก เมื่อนำมาประยุกต์ใช้กับการประมวลผลภาพจะได้ผลลัพธ์ดังแสดงในภาพที่ 2-48



ภาพที่ 2-48 ตัวอย่างการประยุกต์ใช้มอร์โฟโลยี

จากภาพที่ 2-48 แสดงตัวอย่างของประยุกต์ใช้กับภาพลายมือ ซึ่งในภาพที่ 2-48 (a) คือภาพต้นฉบับ A ที่มีการรบกวนด้วยสิ่งรบกวน (Noise) และในภาพ 2-48 (b) เป็นหน่วยโครงสร้าง B เมื่อทำการเซาะภาพ A ด้วย B จะได้ผลลัพธ์ดังภาพที่ 2-48 (c) ซึ่งสิ่งรบกวนจะถูกกำจัดในกรณีที่สิ่งรบกวนนั้นมีขนาดเล็กกว่าหน่วยโครงสร้าง B จากนั้นทำการเปิดภาพ (Opening) ภาพที่ 2-48 (c) ด้วยหน่วยโครงสร้าง B จะได้ผลลัพธ์ดังภาพที่ 2-48(d) ซึ่งจะเห็นว่าขนาดของลายนิ้วมือจะใกล้เคียงกับภาพต้นฉบับแต่ภาพยังมีช่องว่างระหว่างรอยนิ้วมืออยู่จากนั้นในภาพที่ 2-48 (e) เป็นผลลัพธ์ของการนำภาพที่ 2-48 (d) มาทำการขยายภาพ (Dilation) และในภาพที่ 2-48(f) จะเป็นผลลัพธ์ที่ทำการปิดภาพ (Closing) กับภาพที่ 2-48 (e) ซึ่งจะเห็นได้ว่าการเชื่อมต่อของเส้นลายนิ้วมือเชื่อมต่อกันอย่างสมบูรณ์เมื่อเทียบกับภาพต้นฉบับ และประโยชน์ที่เห็นได้ชัดเจนคือสิ่งรบกวนที่มีอยู่มากในภาพต้นฉบับได้ถูกกำจัดทำให้บริเวณขอบภาพมีความคมชัดขึ้นในงานวิจัยนี้จะมีการใช้เทคนิคการเปิดภาพและปิดภาพช่วยในการเติมเต็มภาพรอยขีดข่วนและเติมเต็มช่องว่างขนาดเล็ก หรือรู (hole) ที่เกิดขึ้นบนภาพ QR Code เมื่อมีการมีการลบรอยขีดข่วน ทั้งนี้เพื่อให้ได้ภาพ QR Code ที่มีความสมบูรณ์มากที่สุด

2.8 การแปลงข้อมูลภาพสองมิติ (Two Dimension Geometric Transformation)

ในการแปลงข้อมูลภาพนี้จะหมายถึงการปรับเปลี่ยนตำแหน่ง การเลื่อนภาพ การหมุนภาพ การบิดภาพ [11] ซึ่งเป็นขั้นตอนที่จำเป็นมากในกระบวนการประมวลผลภาพ ทั้งนี้การแปลงข้อมูลที่กำลังกล่าวในหัวข้อนี้จะเป็นการแปลงข้อมูลในภาพสองมิติเท่านั้น

2.8.1 การเลื่อนจุดพิกัดภาพ (Translation)

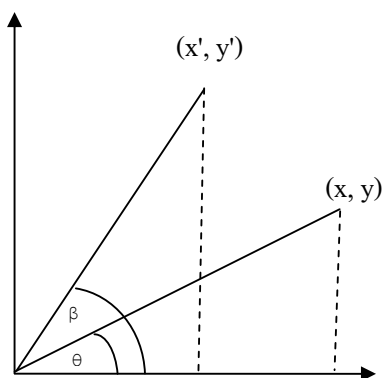
ในการเลื่อนตำแหน่งภาพจะเป็นการเลื่อนตามแนวแกน x ซึ่งจุด x จะเลื่อนไปที่ตำแหน่ง x' และในแนวแกน y จุด y จะเลื่อนไปที่จุด y' โดยที่ในการเลื่อนแต่ละจุดจะเป็นการบวกเพิ่มค่าระยะทาง คือ T_x และ T_y ในการเลื่อนจะใช้ Coordinate เดิมทำให้ตำแหน่งใหม่ของภาพสอดคล้องกับตำแหน่งของภาพเดิม ในการเลื่อนภาพจะได้จากสมการดังต่อไปนี้

$$x' = x + T_x \quad (2.35)$$

$$y' = y + T_y \quad (2.36)$$

2.8.2 การหมุนภาพ (Rotation)

ในการหมุนภาพเป็นวิธีในการปรับตำแหน่งของภาพวัตถุจากตำแหน่ง (x, y) ไปยังภาพในตำแหน่ง (x', y') ซึ่งในการหมุนภาพจำเป็นต้องทำที่จุดหมุนเสมอ ผลลัพธ์ที่ได้จากการหมุนภาพจะยังคงมีรูปร่างลักษณะและขนาดที่เหมือนกับภาพต้นฉบับแต่ตำแหน่งของภาพจะเปลี่ยนไป



ภาพที่ 2-49 การหมุนภาพจากจุดกำเนิด

ในกระบวนการหมุนภาพขนาดของมุมที่หมุนแทนด้วย β และในที่นี้ให้มีจุดหมุนอยู่ที่จุดกำเนิดมุม โดยที่ θ เป็นขนาดมุมที่ได้จากการวัดตามแกนนอนจนถึงจุด (x,y) ซึ่งจะได้สมการที่มีความสัมพันธ์ดังต่อไปนี้

$$x' = r \cos \theta \cos \beta - r \sin \theta \sin \beta \quad (2.37)$$

$$y' = r \sin \theta \cos \beta + r \cos \theta \sin \beta \quad (2.38)$$

กำหนดให้ r คือระยะทางจากจุดกำเนิดถึงจุดที่พิจารณา นอกจากนี้จะได้ว่า

$$x = r \cos \theta \quad (2.39)$$

$$y = r \sin \theta \quad (2.40)$$

จะได้สมการใหม่ดังต่อไปนี้

$$x' = x \cos(\beta) - y \sin(\beta) \quad (2.41)$$

$$y' = x \sin(\beta) + y \cos(\beta) \quad (2.42)$$

2.8.3 การย่อและขยายภาพ (Scaling)

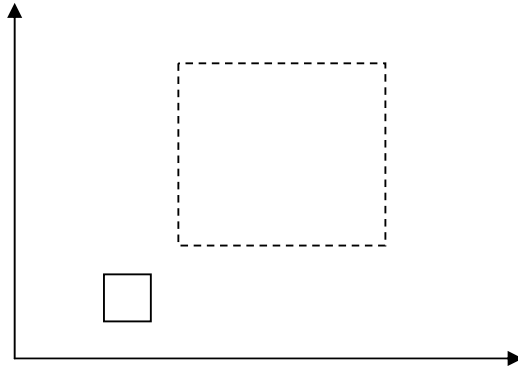
การย่อภาพและการขยายภาพเป็นการเปลี่ยนแปลงขนาดภาพ โดยภาพวัตถุสามารถเปลี่ยนขนาดได้โดยการเปลี่ยนระยะทางระหว่างจุด สำหรับการเปลี่ยนแปลงค่าสามารถทำได้ด้วยการคูณระยะห่างระหว่างจุด (x,y) เดิมกับค่าที่จะทำให้เพิ่มหรือลดขนาดลงสำหรับค่านี้เรียกว่า ค่าสเกลลิงแฟกเตอร์ (Scaling factor) ได้แก่ S_x และ S_y ซึ่งจะมีเงื่อนไขดังต่อไปนี้

ถ้า $0 < S_x, S_y < 1$ คือ การย่อภาพ

ถ้า $S_x, S_y > 1$ คือ การขยายภาพ

ถ้า $S_x = S_y$ คือ แสดงว่าการย่อขนาดและขยายขนาดเป็นไปตามอัตราส่วน

ถ้า $S_x \neq S_y$ คือ แสดงว่าการย่อขนาดและขยายขนาดไม่เป็นไปตามอัตราส่วน



ภาพที่ 2-50 ภาพตัวอย่างการทำการย่อขยายภาพ (Scaling)

ในกรณีที่มีการย่อภาพนั้นมีจุดประจำที่ (Fixed point) ที่จะใช้ในการย่อขยายภาพซึ่งใช้สำหรับเป็นจุดอ้างอิง ถ้ากำหนดให้จุด (x, y) เป็นจุดใด ๆ ในภาพต้นฉบับในการย่อหรือขยายภาพจะทำได้ด้วยการคูณด้วยค่าสเกลลิงแฟกเตอร์ S_x, S_y โดยที่ทำการคูณ S_x ตามระนาบแกน x และ S_y ตามระนาบแกน y จะได้จุดใหม่ (x', y') ดังนี้

เมื่อ p คือจุดใดๆ ของภาพ

$$p = \begin{pmatrix} x \\ y \end{pmatrix} \quad (2.43)$$

เมื่อ p' คือจุดของภาพในตำแหน่งใหม่

$$p' = \begin{pmatrix} x' \\ y' \end{pmatrix} \quad (2.44)$$

เมื่อ S คือ สเกลลิงแฟกเตอร์ (Scaling factor)

$$S = \begin{pmatrix} S_x & 0 \\ 0 & S_y \end{pmatrix} \quad (2.45)$$

จะได้ p' ซึ่งเป็นภาพจุดใหม่คือ

$$\begin{aligned}
 p' &= Sp \\
 &= \begin{pmatrix} S_x & 0 \\ 0 & S_y \end{pmatrix} \\
 &= \begin{pmatrix} S_x & x \\ S_y & y \end{pmatrix} \\
 \begin{pmatrix} x' \\ y' \end{pmatrix} &= \begin{pmatrix} S_x & x \\ S_y & y \end{pmatrix} \tag{2.46}
 \end{aligned}$$

เมื่อ S_x คือค่า Scaling Factor ตามแกน x และ S_y คือค่า Scaling Factor ในแนวตั้ง

2.9 เกณฑ์การวัดประสิทธิภาพ

2.9.1 การวัดค่าความเหมือนของภาพ (Similarity Measurement)

ในการวัดความเหมือนของภาพระหว่างภาพต้นฉบับและภาพผลลัพธ์ที่ได้จะใช้ค่า NC (Normalized Cross Correlation) ซึ่งค่าที่ได้จะมีค่าระหว่าง 0 ถึง 1 ในการคำนวณหาค่า NC จะได้จากสมการดังต่อไปนี้

$$NC(A,B) = 1 - \frac{\sum_{i=1}^n \sum_{j=1}^n (a_{ij} \oplus b_{ij})}{n^2} \tag{2.47}$$

กำหนดให้ A แทนภาพต้นฉบับของ QR Code และ B ภาพผลลัพธ์ QR Code ที่ได้ ซึ่ง A และ B เป็นเมทริกซ์จัตุรัสขนาด $n \times n$ มิติ โดยที่ a_{ij} เป็นสมาชิกแถวที่ i หลักที่ j ของเมทริกซ์ และ b_{ij} เป็นสมาชิกแถวที่ i หลักที่ j ของเมทริกซ์ B และ $a \oplus b = \begin{cases} 0, & a = b \\ 1, & a \neq b \end{cases}$

สำหรับค่าความเหมือนของภาพนั้นเมื่อค่าเข้าใกล้ 1 แสดงว่าภาพผลลัพธ์นั้นมีความใกล้เคียงกับภาพต้นฉบับ และถ้าหากค่า NC มีค่าเท่ากับ 1 หมายความว่าภาพผลลัพธ์มีความเหมือนกับภาพต้นฉบับ 100 %

2.9.2 เกณฑ์การวัดเวลาที่ใช้ในการประมวลผล (Time of Process)

เนื่องด้วยกระบวนการประมวลผลภาพที่ใช้ในงานวิจัยนี้จำเป็นต้องมีการวัดค่าเวลาเพื่อช่วยในการประเมินประสิทธิภาพในการประมวลผล โดยทำการจับเวลาตั้งต้น T_{start} และเวลาที่สิ้นสุด T_{stop} เพื่อให้เวลาที่ใช้ในการวัดมีความถูกต้องในการศึกษาครั้งนี้ได้ใช้เวลาของหน่วยประมวลผลกลาง (CPU Time) โดยใช้สมการดังนี้ในการจับเวลา

$$\Delta T = T_{start} - T_{stop} \quad (2.48)$$

2.10 เอกสารและงานวิจัยที่เกี่ยวข้อง

QR Code เป็นสัญลักษณ์แบบสองมิติถูกพัฒนาจากบริษัท Denso Wave [4] และถูกเผยแพร่ในปี 1994 นอกจาก QR Code แล้วยังมีบาร์โค้ดสองมิติบางประเภทถูกพัฒนาอีกมากมาย เช่น Numeric Alphanumeric Kanji Kana Symbols Binary และ Control Code สูงถึง 7,089 อักษรแปลงเป็น 1 สัญลักษณ์ [3] ในส่วนของกระบวนการประมวลผลด้วยภาพที่ใช้กับ QR Code ส่วนใหญ่จะเป็นรวบรวม และนำเสนอเทคนิคที่มุ่งเน้นในปรับปรุงภาพ QR Code วิธีการวัดผลและนำไปสู่การได้ผลลัพธ์คือ QR Code ที่สามารถทำการถอดรหัสข้อมูลได้ซึ่งผลงานจำนวนหนึ่งได้เสนอออกมา นั้นจะขอกกล่าวในลักษณะสรุปสั้นๆ เพื่อให้เห็นถึงวิธีการต่างๆ ดังนี้

ผลงานวิจัยของ XiongZou และคณะ[12] ซึ่งได้ทำการศึกษาเกี่ยวกับลำดับการประมวลผลภาพประเภทไบนารีร่วมกับ Median filter โดยที่การศึกษาเริ่มต้นที่การเข้าสู่กระบวนการเตรียมภาพเพื่อให้ได้ภาพตั้งต้นที่จะใช้ในการทดลอง จากนั้นทำการเปรียบเทียบระหว่างภาพที่ได้จากการแปลงภาพ QR Code เป็นภาพไบนารีก่อนที่จะใช้ Median filter ที่มีการปรับปรุงอัลกอริทึม (BMF) กับผลลัพธ์ที่ได้จากการนำภาพตั้งต้น แล้วเริ่มที่กระบวนการ Median filter จากนั้นจึงแปลงภาพเป็นไบนารี (MFB) ผลที่ได้คือการทำ BMF จะให้ผลดีที่กว่าทั้งในด้านเวลาและผลลัพธ์ที่ได้ที่ ให้ผลของค่า Bit Error Rate ที่ดีกว่า MFB จากงานวิจัยนี้แสดงให้เห็นว่าการใช้ Median filter นั้นจะให้ผลดีกับภาพที่ได้จากการแปลงภาพให้เป็นภาพแบบไบนารี

ผลงานวิจัยของ Chung-Hua Chu และคณะ[13] ซึ่งได้ทำการศึกษาเกี่ยวกับภาพ QR Code ที่พิมพ์อยู่บนหนังสือพิมพ์หรือสื่อสิ่งพิมพ์ต่างๆ เมื่อมีการถ่ายภาพ QR code ดังกล่าวโดยใช้อุปกรณ์เคลื่อนที่เช่น โทรศัพท์มือถือ ภาพที่ได้มักจะมีลักษณะพรางมัวหากมีการสั่นไหวระหว่างการถ่ายภาพ ปัญหาดังกล่าวส่งผลให้การอ่านเพื่อถอดรหัส QR Code ไม่สามารถทำได้ในงานวิจัยนี้ได้เสนอวิธีการ Extracting Barcode (EB) Algorithm ซึ่งเริ่มจากการประเมินค่าของการสั่นไหวของภาพที่ได้จากกล้องด้วย Point Spread Function(PSF)จากนั้นเข้าสู่กระบวนการ Camera Shake Restoration โดยการเทียบค่าระดับของภาพระดับเทา (Grayscale) ของจุดที่พิจารณากับจุดรอบข้าง ซึ่งค่าที่ได้จะเป็นค่าที่ลดระดับไปเรื่อยๆ ตามลักษณะการเบลอของภาพหากพบค่าที่มีค่าเทียบเท่ากับจุดที่เราพิจารณาให้ทำการเลือกจุดที่พิจารณาเป็นพิกเซลของภาพ QR code เมื่อทำครบทุกจุดจะได้ภาพ QR code ที่มีลักษณะที่ชัดเจนขึ้น จากนั้นเริ่มการแยกส่วนภาพ QR code จากภาพพื้นหลังที่มีลักษณะซับซ้อนโดยการค้นหาจุดของ Finder Pattern ทั้ง 3 จุดซึ่งจะมีอัตราส่วนเป็น 1:1:3:1:1 เสมอ ทำให้ได้พื้นที่ของ QR code และทำการปรับลักษณะของภาพ QR code ให้เป็นรูปสี่เหลี่ยมจัตุรัส ต่อมาใช้กระบวนการ QR Code Reconstruction เพื่อเติมเต็มความสมบูรณ์ของโมดูลใน QR code

ผลงานการวิจัยของ Yoji Kato และคณะ[14] ได้นำเสนอวิธีการในการแก้ปัญหา QR Code ที่มีความละเอียดต่ำ ในกรณีที่ใช้มือถือถ่ายภาพ QR Code ในระยะไกลในภาพ QR Code ที่มีความละเอียดของภาพต่ำจะมีลักษณะของขอบบริเวณโมดูลระหว่างสีขาวและสีดำที่ไม่สามารถแยกแยะออกจากกันได้อย่างชัดเจน ทำให้กระบวนการในการค้นหา QR Code และกระบวนการถอดรหัสมีปัญหาขึ้นในงานวิจัยนี้จึงนำเทคนิค Super-Resolution มาแก้ปัญหาที่พบโดยที่จะใช้วิธีสร้างภาพ QR Code ที่มีความละเอียดสูงจากภาพ QR Code คุณภาพต่ำหลายๆภาพมาวิเคราะห์ลักษณะของค่าในโมดูล QR Code ซึ่งการวิเคราะห์จะเริ่มจากนำเข้าภาพ QR Code ที่มีความละเอียดต่ำจากนั้นเข้าสู่กระบวนการลงทะเบียนรูปแล้วเลือก Pixel ในจุดของภาพที่มีค่าสมบูรณ์มากที่สุดมาใช้ในภาพตั้งต้น และทำการขยายความละเอียดของภาพตามอัตราส่วนของภาพ QR Code จากนั้นเข้าสู่กระบวนการสร้าง QR Code ใหม่โดยการใช้คุณสมบัติของ QR Code และสุดท้ายคือกระบวนการทดสอบเพื่อดูอัตราการ Recognition บน QR Code เพื่อให้ได้ผลลัพธ์ที่สามารถถอดรหัสได้ แต่ลักษณะ QR Code ที่ได้นั้นยังมีโมดูลพิกเซลที่บิดเบี้ยวอยู่ ซึ่งได้กล่าวถึงในส่วนของสิ่งที่ต้องพัฒนาเพิ่มเติมคือ การนำคุณสมบัติการเป็นสี่เหลี่ยมมาใช้เพิ่มเติมจะช่วยให้ได้ผลลัพธ์ที่ดีขึ้น

ผลงานวิจัยของ YinghuiZhing และคณะ [15] ได้ทำการเสนอวิธีการปรับปรุงวิธีที่ใช้ในการแปลงภาพ QR code เป็นภาพไบนารี เมื่อสภาวะแสงที่ไม่สม่ำเสมอ การแปลงภาพไบนารี โดยทั่วไปนั้นการยึดค่าที่เป็นค่าคงที่แบบ Global ที่ใช้ในการทำ Thresholding โดยที่ไม่คำนึงถึง Local Thresholding ส่งผลให้ภาพไบนารีที่ได้นั้นยากต่อการอ่านเพื่อถอดรหัส การแก้ปัญหาในงานวิจัยนี้ได้ทำการพิจารณาค่าต่างๆ ในรูปแบบบิตก้อยของ QR Code แล้วพิจารณาค่าใน Local นั้นๆ ซึ่ง Algorithm ที่ใช้ตามลำดับ ดังต่อไปนี้ Local binarization algorithm ในภาพระดับเทา, Watershed binarization algorithm, Surface fitting Binarization algorithm, The adaptive threshold binarization algorithm และใช้ The background gray-level base binarization algorithm ร่วมกับ Otsu algorithm เป็นขั้นตอนสุดท้าย ผลการทดลองที่ได้แสดงให้เห็นว่าได้ภาพไบนารีที่มีความสมบูรณ์มากกว่าผลลัพธ์ของ Otsu Algorithm กระบวนการที่ใช้ในงานวิจัยนี้ช่วยเพิ่ม Recognition rate ของ QR code ที่มีปัญหาในสภาวะแสงที่ไม่สม่ำเสมอได้

ผลงานวิจัยของ Yue Liu และคณะ [16] ได้ทำการศึกษาเกี่ยวกับการอ่าน QR Code บนโทรศัพท์มือถือซึ่งมักจะมีปัญหาเมื่อถ่ายภาพ QR Code ในสภาวะของแสงที่ไม่สม่ำเสมอ โดยที่ได้เสนอวิธีการในการแก้ปัญหาดังนี้ ทำการแปลงภาพที่ได้จากกล้องมือถือเป็นภาพระดับเทา เนื่องจากลักษณะของ QR Code มีเพียงสัญลักษณ์ที่เป็นสีขาวและสีดำ จากนั้นทำการแปลงภาพระดับเทาให้เป็นภาพไบนารี โดยที่ใช้การวิเคราะห์ค่า Histogram จากภาพระดับเทาแล้วเลือกเกณฑ์ Thresholding ที่เหมาะสมตามลักษณะความสว่างของภาพนั้นๆ เมื่อได้ภาพไบนารีแล้วให้ทำการกำจัด Noise ด้วยการใช้เทคนิค Opening and Closing แล้วนำผลลัพธ์ที่ได้มาทำการระบุตำแหน่งและพื้นที่ของ QR Code โดยการใช้คุณสมบัติที่มีอยู่ใน QR Code คือ Finder pattern ในการค้นหาตำแหน่งที่ตั้งของ QR Code ในกรณีที่ Finder Pattern เสียหายให้เลือกใช้ Timing pattern ช่วยในการหาระยะทางจาก Finder pattern หนึ่งไปยังอีกจุดหนึ่งได้ซึ่งทำให้สามารถคำนวณหาพื้นที่ของ QR Code ได้จากนั้นให้ทำการค้นหา Alignment Pattern โดยเริ่มจากการหาจุดศูนย์กลางของ QR Code เพื่อให้ได้จุดที่จะทำการสร้าง grid เพื่อทำการสร้างแต่ละ Block ใน QR Code แล้วเข้าสู่กระบวนการสุดท้ายคือการทดสอบการถอดรหัสโดยได้มีการเปรียบเทียบผลที่ได้กับ Otsu's method, Ohbuchi's method, QR Code standard's method พบว่าวิธีของงานวิจัยนี้ให้อัตราการ Recognition rate ที่ดีกว่าส่วนเวลาที่เร็วที่สุดคือ Ohbuchi's method

บทที่ 3

วิธีดำเนินการวิจัย

ในบทนี้จะกล่าวถึงการดำเนินการวิจัยซึ่งจะประกอบไปด้วย การนำเข้าภาพต้นฉบับ จากนั้นเข้าสู่กระบวนการแปลงค่าสีจากแบบจำลองสี RGB ไปยังแบบจำลองสี HSV เข้าสู่กระบวนการแยกแยะองค์ประกอบของภาพ (Image Segmentation) โดยใช้แบบจำลองสี HSV จะได้ภาพรอยขีดข่วน นำภาพรอยขีดข่วนดังกล่าวมาเติมเต็มเพื่อให้ได้รอยขีดข่วนที่สมบูรณ์โดยใช้กระบวนการเปิดภาพ (Opening) และปิดภาพ (Closing) เข้าสู่กระบวนการลบรอยออกจากภาพต้นฉบับ นำผลลัพธ์ที่ได้มาเข้าสู่กระบวนการ Grids generating โดยทำการแบ่งพื้นที่ย่อยของ QR Code เพื่อใช้ในกระบวนการเติมเต็มโมดูลภาพใน QR Code (Reconstruction QR Code), นำภาพผลลัพธ์ที่ได้ไปเข้าสู่กระบวนการถอดรหัส QR Code และการวัดประสิทธิภาพของผลลัพธ์จากการวิจัยโดยมีแผนภาพรวมเป็นดังภาพที่ 3-1 ซึ่งประกอบไปด้วยรายละเอียดดังต่อไปนี้

3.1 ภาพรวมของทั้งหมดของงานวิจัย

3.1.1 การแปลงแบบจำลองสีจาก RGB ไปยัง HSV

เริ่มต้นด้วยการนำเข้าไฟล์ภาพ QR Code ที่มีรอยขีดข่วน (Input Image) (จากหัวข้อ 1.4) จากนั้นทำการแปลงค่าสีโดยใช้การแปลงค่าสีจากแบบจำลองสี RGB ไปแบบจำลองสี HSV (จากหัวข้อ 2.5.5)

3.1.2 กระบวนการแยกองค์ประกอบภาพ (Color segmentation using HSV color model)

3.1.2.1 การแยกองค์ประกอบของรอยขีดข่วน (Scratch pattern) โดยการนำผลลัพธ์ที่ได้จากการแปลงค่าสี (ผลลัพธ์ 3.1.1) เลือก S (Saturation Channel) จากนั้นทำการแปลงภาพรอยเป็นภาพไบนารีโดยใช้เทคนิคการทำเทรชโซลด์ แล้วนำผลลัพธ์ที่ได้มาทำการเปิดภาพ (Opening) และการปิดภาพ (Closing) จะได้ภาพรอยที่พร้อมสำหรับขั้นตอนต่อไป

3.1.2.1 การเตรียมภาพที่นำเข้า (Input Image) ในขั้นตอนนี้เริ่มต้นด้วยการนำภาพที่ผ่านการแปลงค่าสี (ผลลัพธ์ 3.1.1) โดยเลือก V (Value Channel)

3.1.3 กระบวนการลบรอยขีดข่วน (Scratches Removing)

เมื่อได้ภาพรอย (ผลลัพธ์ 3.1.2.1) และภาพนำเข้าที่ผ่านกระบวนการเตรียมภาพ (ผลลัพธ์ 3.1.2.2) จากนั้นทำการนำตำแหน่งของจุดพิกเซลของภาพรอย (ผลลัพธ์ 3.1.2.1) มาเทียบกับตำแหน่งของภาพ (ผลลัพธ์ 3.1.2.2) แล้วทำการลบรอยตามตำแหน่งนั้นๆ จากนั้นทำการแปลงค่าภาพที่ได้เป็นภาพไบนารี

3.1.4 การกรองสัญญาณรบกวน (Filtering)

เมื่อได้ภาพที่ปราศจากรอยขีดข่วน (ผลลัพธ์ 3.1.3) จากนั้นทำการเติมเต็มช่องว่างขนาดเล็กที่อยู่บนภาพ QR Code (ผลลัพธ์ 3.1.3) และกรองสัญญาณรบกวนโดยใช้เทคนิคทางมอร์โฟโลยี (Mophology) ซึ่งได้แก่ Dilation, Erosion, Opening และ Closing

3.1.5 การแบ่งพื้นที่ย่อยของ QR Code (Grids Generating)

เริ่มต้นด้วยการกำหนดพื้นที่สนใจ (Region-Of-Interest: ROI) โดยที่ไม่สนใจพื้นที่ในส่วนของ QuietZone ใน QR Code แล้วทำการแปลงพื้นที่ที่สนใจให้เป็นภาพสี่เหลี่ยมจัตุรัสโดยใช้เทคนิคการแปลงข้อมูลภาพสองมิติ (Two Dimension Geomatic Transformation) ทำการตีเส้นกริด (Grid Line) โดยใช้คุณสมบัติของ QR Code ซึ่งใน QR Code แต่ละเวอร์ชันจะมีการกำหนดขนาดของโมดูลอย่างชัดเจน (จากหัวข้อ 2.2)

3.1.6 กระบวนการเติมเต็มโมดูลภาพใน QR Code (Reconstruction QR Code)

เป็นกระบวนการที่ทำการวิเคราะห์ในแต่ละกลุ่มของโมดูลที่ได้จากการตีเส้นกริด (ผลลัพธ์ 3.1.5) ในการเติมเต็มนี้จะใช้เกณฑ์การเลือกจุดพิกเซลสีที่มีมากในแต่ละพื้นที่ย่อย จากนั้นทำการเติมเต็มในส่วนของ Finder Pattern, Timing Pattern, ทำการขยายภาพเพื่อใช้สำหรับพื้นที่ของ Quiet Zone จะทำให้ได้ภาพผลลัพธ์ (Output) และทำการคำนวณค่า NC ระหว่างภาพ QR Code ผลลัพธ์ (Output) กับภาพ QR Code ต้นฉบับ (Original) พร้อมทั้งคำนวณเวลา (Time) ที่ใช้ในการประมวลผล (ที่ได้จาก 3.1.1 ถึง 3.1.6)

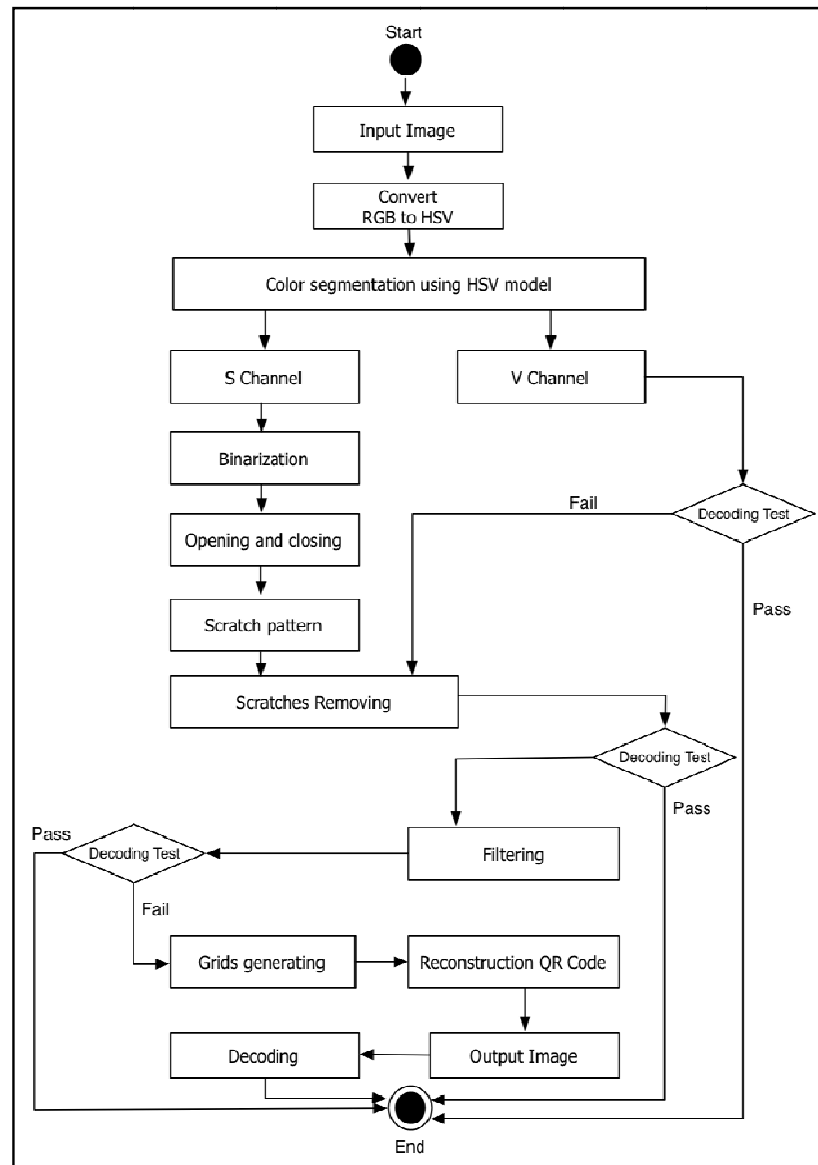
3.1.7 การถอดรหัสภาพคิวอาร์โค้ด (Decoding QR Code)

3.1.7.1 ถอดรหัสภาพคิวอาร์โค้ด

ทำการถอดรหัสภาพผลลัพธ์ (ที่ได้จาก 3.1.6) โดยใช้ Zxing Library [17] ซึ่งเป็นโอเพนซอร์สที่ใช้ในการถอดรหัสบาร์โค้ดสองมิติ

3.1.7.2 จัปเวลาการถอดรหัสภาพคิวอาร์โค้ด

ทำการคำนวณเวลาที่ใช้ในกระบวนการถอดรหัสภาพคิวอาร์โค้ดผลลัพธ์ (ที่ได้จาก 3.1.7.1)



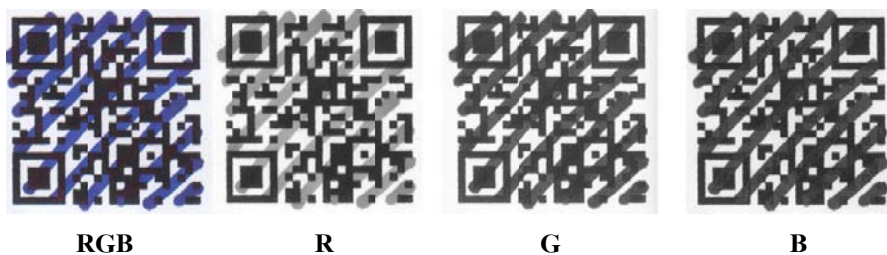
ภาพที่ 3-1 ภาพรวมทั้งหมดของงานวิจัย

3.2 การแปลงแบบจำลองสีจาก RGB ไปยัง HSV

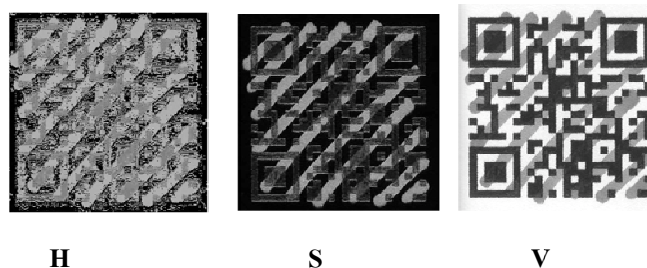
ในงานวิจัยเริ่มต้นด้วยการนำเข้าภาพ (Input Image) ดังภาพที่ 3-2 ซึ่งเป็นภาพคิวอาร์โค้ด (QR Code) ที่มีรอยขีดข่วนดังที่ระบุในข้อตกลงเบื้องต้น (จากหัวข้อ 1.4) เนื่องด้วยการจัดเก็บไฟล์ภาพดิจิทัลและการแสดงผล โดยทั่วไปจะใช้แบบจำลองสี RGB ทั้งนี้เพื่อประโยชน์ในการแสดงผลสีที่ถูกต้องบนหน้าจอคอมพิวเตอร์ แต่เนื่องด้วยข้อจำกัดของแบบจำลองสี RGB ซึ่งมีรายละเอียดในหัวข้อ 2.5.4.1 ที่ในทุก ๆ สีในแบบจำลองสี RGB จะเกิดจากการผสมของแม่สีทั้ง 3 สีที่แตกต่างกัน ดังแสดงในภาพที่ 3-3 ซึ่งการเปลี่ยนแปลงของแม่สีใดแม่สีหนึ่งมีผลต่อค่าสี ซึ่งรวมไปถึงส่งผลต่อการเปลี่ยนแปลงของค่าสีด้วย ดังนั้นในงานวิจัยนี้จะใช้แบบจำลองสี HSV โดยแบบจำลองสี HSV คุณสมบัติที่โดดเด่นกว่า RGB ซึ่งมีรายละเอียดในหัวข้อ 2.5.4.2 ในกระบวนการแปลงแบบจำลองสีจาก RGB ไปยัง HSV ซึ่งมีรายละเอียดดังที่แสดงในหัวข้อ 2.5.5 ผลลัพธ์ของภาพ QR Code เมื่อผ่านกระบวนการแปลงค่าสีแสดงในภาพที่ 3-4



ภาพที่ 3-2 ตัวอย่างภาพที่นำเข้า (Input Image)





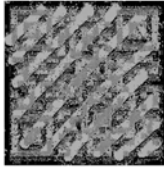
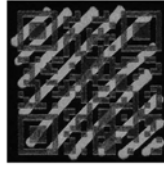






ภาพที่ 3-3 ภาพที่นำเข้าในแบบจำลองสี RGB



ภาพที่ 3-4 ผลลัพธ์การแปลงค่าสีจากแบบจำลองสี RGB ไปยัง HSV

จากภาพ 3-4 แสดงผลภาพที่นำเข้าไปในแบบจำลองสี HSV ซึ่งเมื่อเปรียบเทียบกับภาพที่ได้ในแบบจำลองสี RGB แล้ว เมื่อสังเกตที่ค่าสีในส่วนของ S หรือ Saturation ซึ่งแสดงถึงค่าอิ่มสีของรอยขีดข่วนที่โดดเด่น ซึ่งในค่าอิ่มสีนี้จะมีการเปลี่ยนแปลงที่น้อยกว่าค่า H หรือ Hue ซึ่งก็คือค่าสีในทุกๆ สี เมื่อมีการเปลี่ยนแปลงสีค่า Hue จะเปลี่ยนไป เช่น การเปลี่ยนจากสีน้ำเงินเป็นสีเขียว และในส่วนของ V หรือ Value ซึ่งเป็นค่าความสว่างโดยที่เมื่อมีการเปลี่ยนแปลงของแสงสว่างจะทำให้ค่าใน V เปลี่ยนไปด้วย ตัวอย่างของภาพในแต่ละ Channel ของ HSV เมื่อมีการเปลี่ยนค่าสีจากสีน้ำเงินเป็น สีเขียวแสดงในตารางที่ 3-1

ตารางที่ 3-1 ตัวอย่างของภาพใน HSV เมื่อมีการเปลี่ยนค่าสี

RGB	HSV	Hue	Saturation	Value
				
				

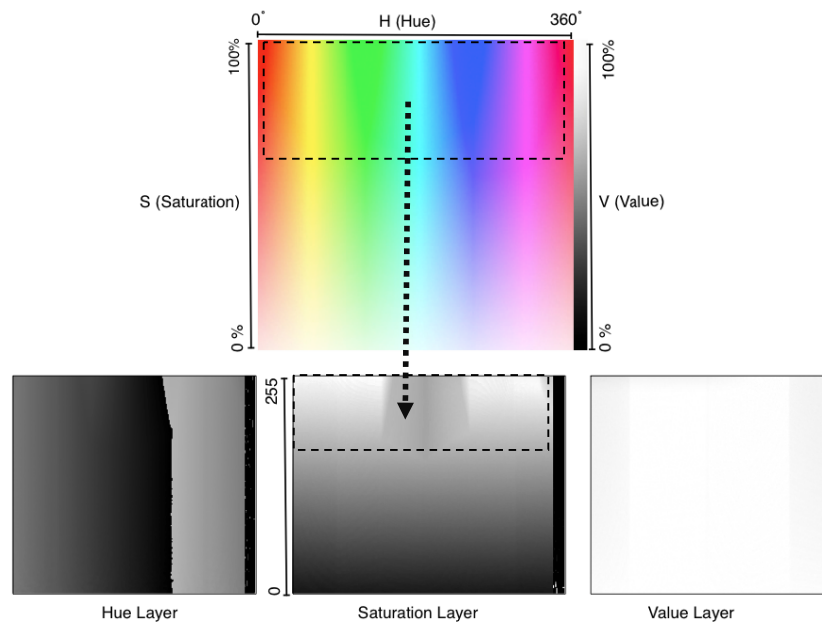
จากตารางที่ 3-1 แสดงให้เห็นว่าค่าสีที่เปลี่ยนไปจากสีน้ำเงินเป็นสีเขียวในแบบจำลองสี RGB ซึ่งเมื่อทำการแปลงค่าสีไปยังแบบจำลอง HSV ในส่วนของ (H) หรือ Hue จะมีค่าที่เปลี่ยนไป ส่งผลให้ค่าระดับเทาในส่วนของ Hue เปลี่ยนไปด้วย ในขณะที่ค่าในส่วนของ V หรือ Value ก็มีการเปลี่ยนแปลงเช่นกัน ทั้งนี้เนื่องด้วยปริมาณแสงที่แตกต่างของภาพที่นำเข้าไป แต่เมื่อสังเกตที่ S หรือ Saturation ของทั้งสองภาพจะยังคงความชัดเจนในส่วนของภาพรอยขีดข่วน เนื่องจากความอิ่มสีของสีทั้งคู่ใกล้เคียงกัน ทั้งนี้เนื่องจาก QR Code มีลักษณะที่มีสีดำ และพื้นหลังที่เป็นสีขาวซึ่งไม่มีความอิ่มสี ส่งผลทำให้ภาพของ Saturation ในบริเวณสีขาวและดำจึงกลมกลืนไปเป็นพื้นหลัง และทำให้ภาพรอยขีดข่วนโดดเด่น ในงานวิจัยนี้จะสนใจภาพในส่วนของ S (Saturation) คือ ตำแหน่งของรอยขีดข่วนที่อยู่บนภาพ QR Code ซึ่งจำเป็นต้องมีการแยกองค์ประกอบภาพเพื่อให้ได้ตำแหน่งรอยที่สมบูรณ์ เพื่อใช้ในการกำจัดรอยขีดข่วนต่อไป

3.3 กระบวนการแยกองค์ประกอบของภาพ

ในขั้นตอนกระบวนการแยกองค์ประกอบของภาพ สามารถแบ่งได้ 3 ขั้นตอนย่อย ดังต่อไปนี้

3.3.1 การแยกองค์ประกอบของรอยขีดข่วน

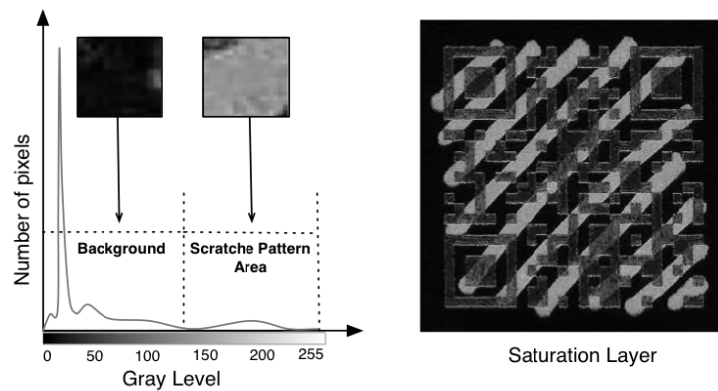
ตามทฤษฎีของแบบจำลองสี HSV (จากหัวข้อ 2.5.5) เมื่อแปลงค่าสีจากแบบจำลองสี RGB ไปยัง HSV ค่าความอิ่มตัวของสี (Saturation) จะมีค่าตั้งแต่ 0% - 100% ซึ่งก็คือ 0 - 255 ในภาพระดับเทาที่แสดงในภาพที่ 3-5 เมื่อพิจารณาค่าอิ่มตัวมาก ในค่าของสีที่มีค่าใกล้เคียง 60% - 100% ได้แก่ แดงเข้ม เขียวเข้ม น้ำเงินเข้ม ชมพูเข้ม เป็นต้น ซึ่งถ้ารอยขีดข่วนมีสีที่มีค่าความอิ่มตัวดังกล่าวจะมีผลทำให้กระบวนการถอดรหัส QR Code ไม่สามารถทำได้เนื่องจากสีดังกล่าวไปรบกวนโมดูลสีดำและสีขาวของ QR Code จึงจำเป็นต้องมีการแยกองค์ประกอบของรอยขีดข่วนที่มีค่าความอิ่มตัวดังกล่าว



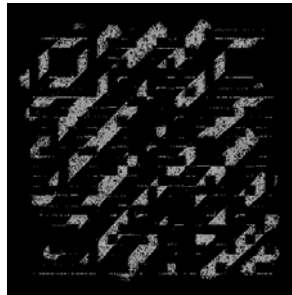
ภาพที่ 3-5 ค่าระดับเทาของ Saturation Layer

จากผลลัพธ์การแปลงแบบจำลองสีในหัวข้อ 3.2 ภาพรอยขีดข่วนที่แสดงในส่วน ของ S (Saturation) ที่ได้จากการแปลงค่าจากแบบจำลองสี RGB นั้นจะมีค่าระดับเทาเป็น 0 - 255 เมื่อนำภาพในส่วนของ S ที่ได้มาสร้างแผนภูมิฮิสโตแกรม (Histogram) จะได้ดังภาพที่ 3-6 เพื่อให้ได้ตำแหน่งของพิกเซลของภาพรอยขีดข่วนจำเป็นต้องมีการทำเทรซโชลด์ตามสมการ(2.28)โดย

ค่าของระดับเทาที่เกิดจากค่าอิมพัลส์ของสีในส่วนของ S ดังในภาพที่ 3-5 เมื่อพิจารณาที่แผนภูมิฮิสโตแกรมค่าของรอยขีดข่วนที่จะอยู่ในช่วงประมาณ $\sim 140 - 255$ ซึ่งคือสีระดับเทาไปถึงสีขาว ซึ่งในค่าของ Saturation ที่ต่ำกว่าช่วงที่ระบุคือส่วนของพื้นหลังของภาพ (Background)

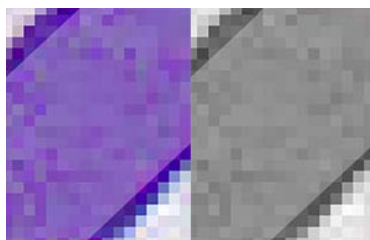


ภาพที่ 3-6 แผนภูมิฮิสโตแกรมของภาพระดับเทาในส่วนของ Saturation



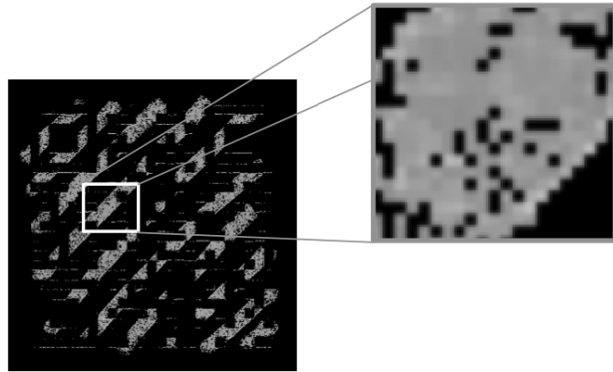
ภาพที่ 3-7 ภาพไบนารีของรอยขีดข่วน

เมื่อทำการเทรซโซลด์ด้วยค่าระดับเทาที่สนใจจะได้ภาพของรอยขีดข่วนที่มีลักษณะเป็นภาพไบนารีดังภาพที่ 3-7 ซึ่งโดยทั่วไปในรอยขีดข่วนที่เกิดจากการใช้ปากกาสีต่างๆ จะมีการซึมของหมึกในบริเวณขอบของรอยขีดข่วนดังในภาพที่ 3-8 แสดงให้เห็นขอบของรอยที่มีการซึมของหมึกสี



ภาพที่ 3-8 รอยขีดข่วนที่มีการซึมของหมึก

เมื่อสังเกตที่ภาพผลลัพธ์ที่ 3-7 จะเห็นได้ว่ามีช่องว่างเล็กๆ จำนวนมากในรอยขีดข่วนซึ่งแสดงให้เห็นในภาพที่ 3-9 ซึ่งช่องว่างเหล่านี้มีความจำเป็นที่ต้องมีการเติมเต็มเพื่อให้ได้ตำแหน่งของรอยขีดข่วนที่สมบูรณ์ขึ้น

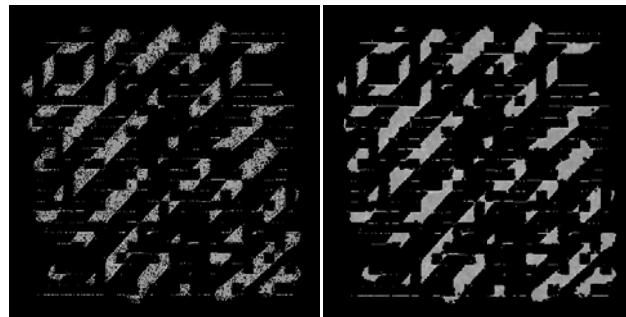


ภาพที่ 3-9 ช่องว่างในภาพรอยขีดข่วน

จากปัญหาในภาพที่ 3-8 และภาพที่ 3-9 จำเป็นต้องมีการใช้เทคนิคการปิดภาพ (Closing) ซึ่งได้กล่าวถึงรายละเอียดใน 2.7.3 โดยในกรณีนี้จะใช้หน่วยโครงสร้างขนาด 3x3 ซึ่งจะช่วยในการเติมเต็มช่องว่างเล็กๆ ในภาพรอยขีดข่วน และช่วยในการขยายขนาดของรอยขีดข่วนเพื่อให้ได้พื้นที่ของรอยที่ครอบคลุมส่วนของรอยที่เกิดจากการซึมของหมึกซึ่งเมื่อผ่านกระบวนการนี้จะทำให้ได้รอยที่มีความสมบูรณ์ดังภาพที่ 3-10

1	1	1
1	1	1
1	1	1

Structure Element



(Before apply Closing)

(After applied Closing)

ภาพที่ 3-10 ผลลัพธ์ที่ได้จากการทำการปิดภาพ (Closing) กับภาพรอยขีดข่วน

3.3.2 การเตรียมภาพนำเข้า (Input Image Preparation)

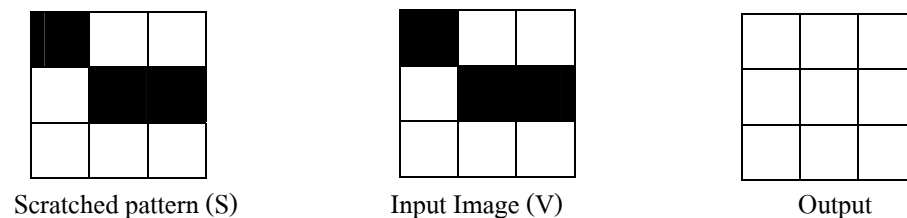
ในขั้นตอนนี้จะเป็นการแปลงภาพที่นำเข้า (Input Image) ซึ่งนำผลลัพธ์ที่ได้จากขั้นตอนที่ 3.2 ซึ่งเป็นการแปลงค่าสีจากแบบจำลองสี RGB ไปยัง HSV โดยในขั้นตอนนี้จะสนใจภาพในส่วนของ V หรือ Value ซึ่งภาพในส่วนนี้จะมีเพียงค่าความมืดและความสว่างโดยมีค่าเป็น 0 % ถึง 100 % แต่ในค่าระดับเทาจะใช้ค่า 0 ถึง 255 ไส้ทอนสีจากสีดำไปสีขาว ในขั้นตอนนี้จะมีการนำภาพที่ได้ในส่วน of V ไปทดสอบการถอดรหัส (Decoding Test) ซึ่งในกรณีที่สามารถถอดรหัสได้ (Pass) ก็ไม่จำเป็นต้องผ่านขั้นตอนอื่นให้จบการทำงานเนื่องจากสามารถถอดรหัสภาพได้แล้ว ส่วนในกรณีที่ถอดรหัสไม่ผ่านให้ทำในขั้นตอนต่อไป ตัวอย่างภาพในส่วน of V แสดงดังภาพที่ 3-11 ซึ่งจะเลือกภาพนี้เป็นภาพตั้งต้นในการลบรอย



ภาพที่ 3-11 ภาพ QR Code ในส่วนของ V ในแบบจำลอง HSV

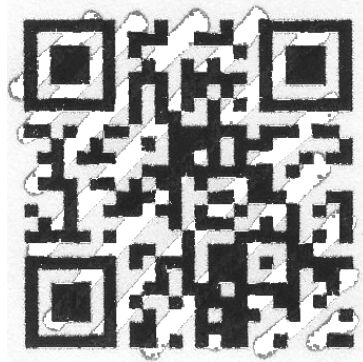
3.3.3 กระบวนการลบรอยขีดข่วน (Scratches Removing)

เมื่อได้ภาพตำแหน่งรอย (ผลลัพธ์จากหัวข้อ 3.3.1) และภาพนำเข้าที่ผ่านการเตรียม (ผลลัพธ์ 3.3.2) ในขั้นตอนนี้จะนำภาพนำเข้าที่ผ่านการเตรียมมาทำการลบรอยโดยการเปลี่ยนค่าสีในตำแหน่งที่มีรอยข่วน ซึ่งการเป็นการปรับค่าของตำแหน่งที่มีรอยให้มีค่าเดียวกับสีของพื้นหลัง ขั้นตอนการลบรอยแสดงดังภาพที่ 3-12



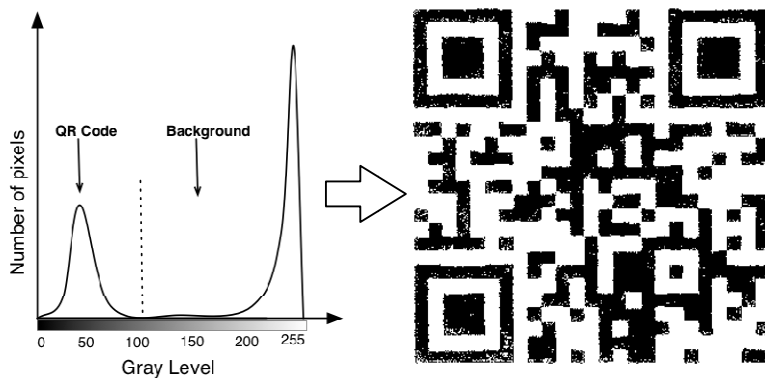
ภาพที่ 3-12 การอ้างอิงตำแหน่งเพื่อทำการลบรอยขีดข่วน

จากภาพที่ 3-12 ในการลบรอยขีดข่วนนั้นจะทำในทุกจุด Pixel โดยที่ทั้งสองภาพจะต้องมีขนาดความกว้างและความสูงเดียวกัน ซึ่งการประมวลผลแบบจุดต่อจุดได้กล่าวถึงรายละเอียดใน 2.6.1 เมื่อทำการแทนที่ค่าสีเพื่อลบรอยขีดข่วนเรียบร้อยแล้วจะได้ผลลัพธ์ดังภาพที่ 3-13



ภาพที่ 3-13 ภาพที่ได้หลังจากการลบรอยรอยขีดข่วน

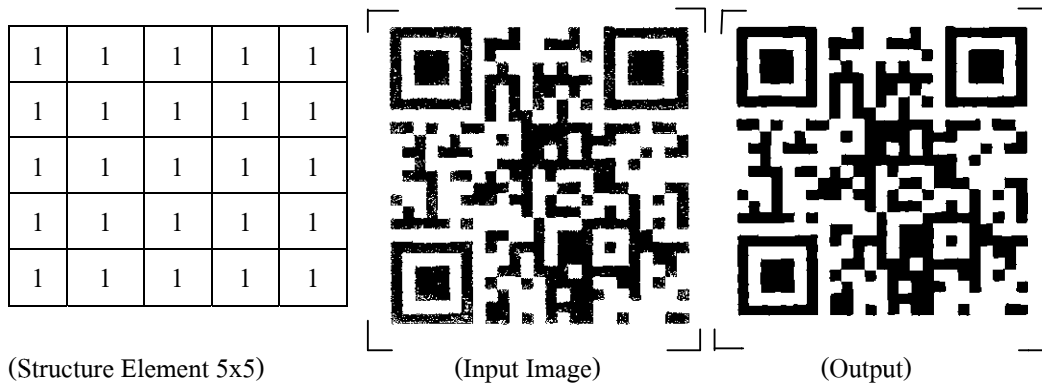
จากภาพ 3-13 เนื่องด้วยภาพตั้งต้น (ผลลัพธ์จาก 3.3.2) เป็นภาพระดับเทาซึ่งมีพื้นหลังยังคงมีค่าหลายระดับในสีระดับเทา และตำแหน่งรอยขีดข่วน (ผลลัพธ์จาก 3.3.1) ที่ใช้ในการลบรายนั้นไม่ครอบคลุมถึงขอบรอยเนื่องจากค่าอิมพัลส์ของสีในขอบรอยขีดข่วนนั้นจะมีค่าใกล้เคียงค่าสีดำมาก ดังนั้นจึงทำให้ผลลัพธ์ในภาพที่ 3-13 มีสัญญาณรบกวน (Noise) ซึ่งสัญญาณรบกวนเหล่านี้จะถูกกำจัดได้ด้วยการทำเทรชโวลด์โดยใช้แผนภูมิฮิสโตแกรมโดยที่จะเลือกเฉพาะค่าที่อยู่ในฝั่งสีดำคือค่าระหว่าง 0 – 100 เพื่อให้ได้เฉพาะภาพ QR Code ผลลัพธ์จากการทำเทรชโวลด์จะได้ภาพไบนารีดังภาพที่ 3-14 จากนั้นนำภาพไบนารีที่ได้ ไปทดสอบการถอดรหัส (Decoding Test) ซึ่งในกรณีที่ถอดรหัสได้ (Pass) ก็ไม่จำเป็นต้องผ่านขั้นตอนอื่นให้จบการทำงานเนื่องจากถอดรหัสภาพได้แล้ว ส่วนในกรณีที่ถอดรหัสไม่ผ่านให้ทำในขั้นตอนต่อไป



ภาพที่ 3-14 ภาพไบนารีของ QR Code เมื่อผ่านกระบวนการลบรอยขีดข่วน

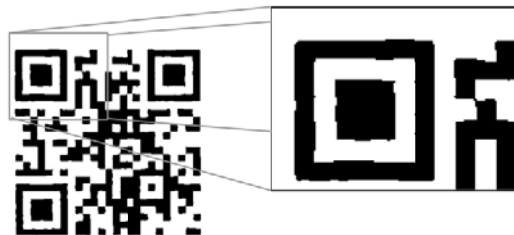
3.3.4 การกรองสัญญาณรบกวน (Filtering)

จากภาพที่ 3-14 เมื่อพิจารณาที่ภาพของ QR Code ที่ได้ในส่วนบริเวณของ โมดูลสีดำ จะมีช่องว่างเกิดขึ้นในขั้นตอนนี้จะเป็นการนำเสนอวิธีการกำจัดช่องว่างเหล่านั้นโดยการใช้เทคนิคทางมอร์โฟโลยี (Morphology) ซึ่งมีรายละเอียดดังที่แสดงในหัวข้อ 2.7 ซึ่งเทคนิคที่ใช้ประกอบด้วย การทำการเปิดภาพ (Opening) และการปิดภาพ (Closing) โดยที่จะใช้หน่วยโครงสร้างขนาด 5x5 เพื่อให้ได้ผลดีในการปิดช่องว่างมีความจำเป็นต้องเลือกหน่วยโครงสร้างที่ใหญ่กว่าช่องว่างที่สนใจ จะทำการเติมเต็มภาพ เมื่อทำการเปิดภาพกับภาพที่ผ่านการลบรอยขีดข่วน (ผลลัพธ์จาก 3.3.3) จะได้ผลลัพธ์ดังภาพที่ 3-15



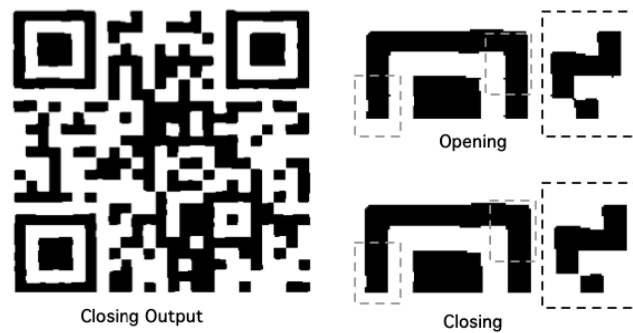
ภาพที่ 3-15 ผลลัพธ์จากการทำการเปิดภาพ (Opening)

จากภาพที่ 3-15 เนื่องด้วยการทำการเปิดภาพ (Opening) จะขยายจุดของภาพในขอบเขตของหน่วยโครงสร้าง (Structure Element) โดยในภาพผลลัพธ์ (Output) ในส่วนของบริเวณภาพของ QR Code จะมีลักษณะหนาขึ้น ช่องว่างที่มีขนาดเล็กกว่าหน่วยโครงสร้างจะถูกเติมเต็ม แต่เมื่อสังเกตที่ภาพผลลัพธ์ (Output) ในส่วนของขอบภาพยังคงมีลักษณะที่ขรุขระ และยังคงมีเส้นบางๆ ที่เชื่อมต่อใน โมดูลของ QR Code ดังแสดงในภาพ 3-16



ภาพที่ 3-16 ลักษณะของขอบภาพ QR Code ที่ผ่านการเปิดภาพ (Opening)

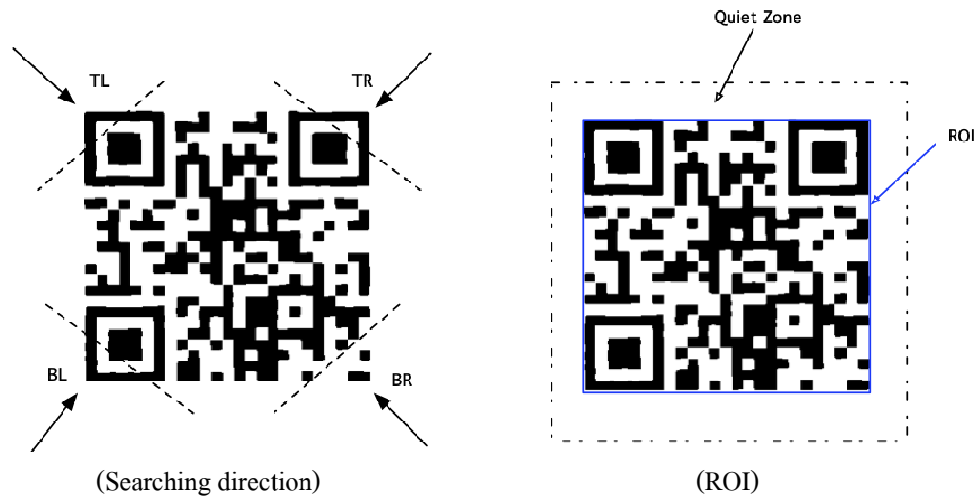
เพื่อให้ได้โครงสร้างของโมดูล QR Code ที่มีขอบของโครงสร้างเรียบจำเป็นต้องมีการใช้เทคนิคการปิดภาพ (Closing) โดยนำภาพผลลัพธ์ที่ได้จากการทำการเปิดภาพและหน่วยโครงสร้าง (Structure Element) ในภาพที่ 3-15 หน่วยโครงสร้างที่ใช้จะเป็นหน่วยโครงสร้างเดียวกับการเปิดภาพ (Opening) เมื่อทำการปิดภาพด้วยโครงสร้างดังกล่าวจะทำให้ได้ภาพ QR Code ที่มีขอบเรียบและส่วนอื่นที่มีลักษณะเป็นเส้นบางๆ จะถูกกำจัด ทำให้ได้ภาพ QR Code ที่มีขอบภาพที่สมบูรณ์ที่มีการแบ่งแยกแต่ละโมดูลชัดเจน ดังแสดงในภาพที่ 3-17 โดยพิจารณาบริเวณเส้นประบนภาพที่แสดงจุดที่มีการปรับปรุงภาพ จากนั้นนำภาพไปนรีที่ได้ ไปทดสอบการถอดรหัส (Decoding Test) ซึ่งในกรณีที่ถอดรหัสได้ (Pass) ก็ไม่จำเป็นต้องผ่านขั้นตอนอื่นให้จบการทำงานเนื่องจากถอดรหัสภาพได้แล้ว ส่วนในกรณีที่ถอดรหัสไม่ผ่านให้ทำในขั้นตอนต่อไป



ภาพที่ 3-17 ผลลัพธ์จากการทำการปิดภาพ (Closing)

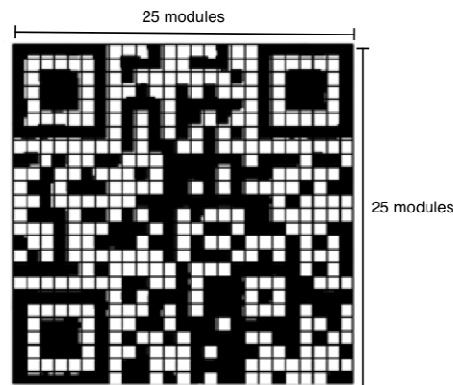
3.3.5 การแบ่งพื้นที่ย่อยของ QR Code (Grid Generating)

ขั้นตอนนี้จะทำการนำภาพ QR Code (ผลลัพธ์จาก 3.3.4) มากำหนดพื้นที่สนใจ (Region-Of-Interest: ROI) ซึ่งในขั้นตอนนี้จะไม่สนใจพื้นที่ของ Quiet Zone (จากหัวข้อ 2.2) ในการกำหนดพื้นที่สนใจในภาพ QR Code จำเป็นต้องทราบมุมของภาพ ซึ่งลักษณะทั่วไปของภาพ QR Code จะมีลักษณะที่เป็นภาพสี่เหลี่ยมซึ่งมีมุมชัดเจนดังนั้นในการค้นหาทำได้โดยการกำหนดทิศทางการค้นหาในแบบทะแยงมุมดังภาพที่ 3-18 โดยมุม TL แทนตำแหน่งพิกัด(x,y) ของมุมบนซ้าย มุม TR แทนตำแหน่งพิกัด (x,y) ของมุมบนขวา, มุม BL แทนตำแหน่งพิกัด(x,y) ของมุมล่างซ้าย และมุม BR แทนตำแหน่งพิกัด (x,y) ของมุมล่างขวาซึ่งเมื่อได้ค่าพิกัดทั้งสี่มุมแล้วจะสามารถกำหนดพื้นที่สนใจได้



ภาพที่ 3-18 ทิศทางการค้นหา และพื้นที่ที่สนใจ (ROI)

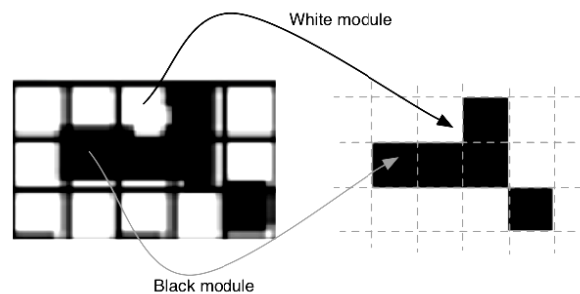
เมื่อได้พื้นที่ที่สนใจ (ROI) แล้วให้ทำการสร้างภาพใหม่ขนาด $M \times N$ โดยที่ $M=N$ ซึ่งในงานวิจัยนี้ได้สร้างภาพใหม่ขนาด 200×200 พิกเซล จากนั้นทำการใช้เทคนิคการแปลงภาพในหัวข้อ 2.8.3 ซึ่งช่วยให้การแปลงภาพในบริเวณของ ROI เป็นขนาดภาพตามที่ต้องการ แล้วทำการเลือกจำนวนโมดูลของ QR Code ในเวอร์ชันที่ทดลองจากตาราง 2-3 ซึ่งแสดงรายละเอียดของจำนวนโมดูลในแต่ละเวอร์ชันของภาพ QR Code ซึ่งในเวอร์ชันที่ทดลองคือเวอร์ชัน 2 ซึ่งจากตารางแสดงจำนวนของโมดูลในแต่ละด้านคือ 25 โมดูล เมื่อทราบจำนวนของโมดูลจะสามารถทำการแบ่งพื้นที่ย่อยของ QR Code ได้เริ่มต้นด้วยการแบ่งพื้นที่ทั้งในแนวคอลัมน์และแถวโดยการแบ่ง QR Code ด้วยจำนวนของโมดูลจะทำให้ได้พื้นที่ย่อยของ QR Code แสดงดังภาพที่ 3-19



ภาพที่ 3-19 ผลลัพธ์จากการแบ่งพื้นที่ย่อยบน QR Code

3.3.6 กระบวนการเติมเต็มโมดูลภาพใน QR Code (Reconstruction QR Code)

การเติมเต็มโมดูลภาพใน QR Code จะเป็นการดำเนินการในส่วนของพื้นที่ย่อยของ QR Code (ผลลัพธ์จาก 3.3.5) ซึ่งในการวิเคราะห์สมาชิกในพื้นที่ย่อยจะใช้เกณฑ์กลุ่มของจุดสี ซึ่งได้แก่ กลุ่มสีดำซึ่งมีค่าเป็น 0 และกลุ่มสีขาว ซึ่งมีค่า 1 จากนั้นทำการปรับค่าในพื้นที่ย่อยตามจำนวนของกลุ่มสีที่มีจำนวนมากในพื้นที่ย่อยนั้นๆ โดยการเติมเต็มจะทำในทุกๆ พื้นที่ย่อยของภาพ QR Code ดังภาพที่ 3-20



(Pixel manipulation)

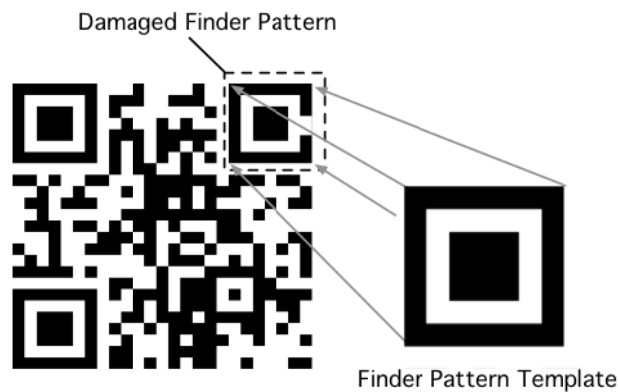


(Output result)

ภาพที่ 3-20 ผลลัพธ์การเติมเต็มโมดูลย่อยของภาพ QR Code

เมื่อทำการเติมเต็มพื้นที่ย่อยของ QR Code ผลลัพธ์ที่ได้คือภาพ QR Code ที่มีความคมชัด ซึ่งในขั้นตอนต่อไปคือการเติมเต็มส่วนของ Finder Pattern เนื่องจากในบางกรณี Finder Pattern ยังคงขาดความสมบูรณ์ตามโครงสร้างมาตรฐานของ QR Code ดังแสดงในภาพ 3-21 โดยทั่วไปในทุกๆ ภาพ QR Code จะมีขนาดของ Finder Pattern ในอัตราส่วน 1:1:3:1:1 ซึ่งเป็นขนาดตามมาตรฐานของ QR Code อีกทั้งตำแหน่งของ Finder จะอยู่ใน 3 มุมเสมอ ดังนั้นในการเติมเต็ม Finder Pattern ทำได้ด้วยการใช้ต้นแบบ (Finder Pattern Template) ของ Finder Pattern

ดั่งภาพที่ 3-21 ซึ่งมีขนาดตามอัตราส่วนของ QR Code กำหนด โดยทำการเติมเต็มทั้ง 3 จุดของ Finder Pattern จะได้ผลลัพธ์ ดั่งภาพที่ 3-22

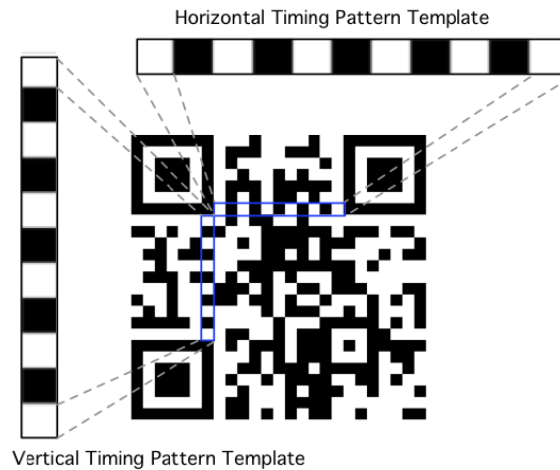


ภาพที่ 3-21 การเติมเต็มส่วนของ Finder Pattern ที่เสียหายด้วยภาพ Finder Pattern ต้นแบบ



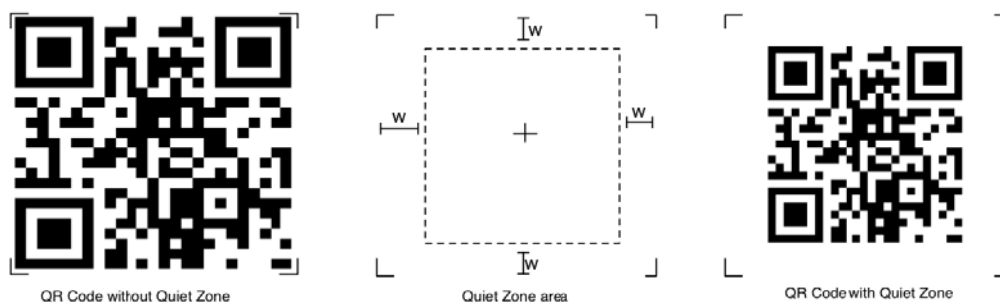
ภาพที่ 3-22 ภาพ QR Code ที่ผ่านการเติมเต็มในส่วน Finder Pattern

เมื่อผ่านการเติมเต็มในส่วนของ Finder Pattern ขั้นตอนต่อไปจะเป็นการเติมเต็มในส่วนของ Timing Pattern ซึ่งจะมีการใช้ต้นแบบของ Timing Pattern (Timing Pattern Template) ซึ่งประกอบด้วย Timing Pattern ในแนวนอน (Horizontal Timing Pattern) และ Timing Pattern ในแนวตั้ง (Vertical Timing Pattern) โดยการเติมเต็มจะดำเนินการตามมาตรฐานของ QR Code ทั้งในตำแหน่งและขนาดของ Timing Pattern ซึ่งได้กล่าวรายละเอียดไว้ใน 2.2 ภาพจำลองการเติมเต็ม Timing Pattern แสดงดั่งภาพที่ 2-23



ภาพที่ 3-23 การเติมเต็ม Timing Pattern โดยใช้ภาพต้นแบบ

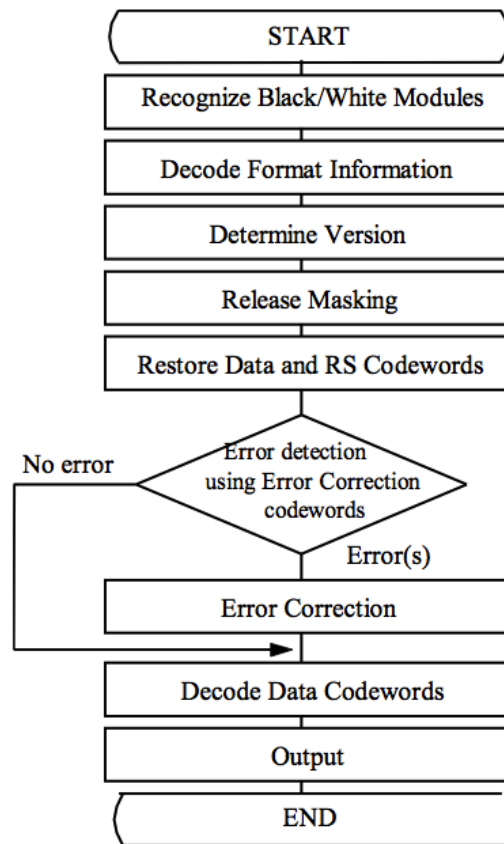
เมื่อผ่านกระบวนการเติมเต็ม Timing Pattern จะได้ภาพผลลัพธ์ที่มีความสมบูรณ์ในขั้นตอนต่อไปเป็นการเพิ่มส่วนของ Quiet Zone ซึ่งเป็นพื้นที่สีขาวรอบๆ QR Code ซึ่งมีความสำคัญมากในกระบวนการถอดรหัสของ QR Code ซึ่งพื้นที่ของ Quiet Zone จะมีขนาดความกว้างของโมดูลเท่ากับ 4 เท่าของขนาดโมดูลย่อยของ QR Code ในการสร้าง Quiet Zone จำเป็นต้องมีการสร้างภาพใหม่โดยให้ภาพใหม่มีขนาดที่ขยายใหญ่ขึ้นโดยมีขนาด 4 เท่าของขนาดโมดูลย่อย จากนั้นทำการนำภาพผลลัพธ์ที่ได้จากการเติมเต็ม Timing Pattern มาวางบริเวณกึ่งกลางของภาพใหม่ซึ่งจะได้ดังภาพที่ 3-24 โดยที่ w คือความกว้างในแต่ละด้านของพื้นที่ Quiet Zone เมื่อผ่านขั้นตอนนี้จะได้ภาพ QR Code ที่มีความสมบูรณ์พร้อมสำหรับการทำการอ่านเพื่อถอดรหัส



ภาพที่ 3-24 การเพิ่มพื้นที่ Quiet Zone ให้กับภาพ QR Code

3.3.7 การถอดรหัสข้อมูล QR Code (Decoding QR Code)

ในการถอดรหัส QR Code ในงานวิจัยนี้ใช้ Zxing Library [17] ซึ่งเป็นโอเพนซอร์สที่ใช้ในการถอดรหัสภาพ QR Code ขั้นตอนในการถอดรหัสภาพ QR Code ได้กล่าวรายละเอียดไว้ในหัวข้อ 2.4 ซึ่งขั้นตอนการถอดรหัสแสดงดังภาพที่ 3-25



ภาพที่ 3-25 กระบวนการถอดรหัสภาพ QR Code [3]

3.4 เกณฑ์การวัดประสิทธิภาพ

3.4.1 การวัดค่าความเหมือนของภาพ (Similarity Measurement)

การวัดความเหมือนของภาพระหว่างภาพต้นฉบับ (Original Image) และภาพผลลัพธ์ (ผลลัพธ์จาก 3.3.6) โดยในการวัดจะเป็นเชิงปริมาณ คือ ใช้ค่า NC (Normalized Cross Correlation) ค่าที่ได้จากการคำนวณจะเป็นค่าที่อยู่ระหว่าง 0 ถึง 1 สำหรับค่าความเหมือนของภาพผลลัพธ์ที่ได้นั้นเมื่อค่าเข้าใกล้ 1 จะหมายถึงภาพผลลัพธ์ที่ได้ใกล้เคียงกับภาพต้นฉบับ (สมการที่ใช้ในการคำนวณค่าอธิบายอยู่ในหัวข้อที่ 2.9.1)

3.4.2 การวัดเวลาที่ใช้ในการประมวลผล

การคำนวณเวลาที่เครื่องคอมพิวเตอร์ที่ใช้ในการประมวลผลในการจัดรอยขีดข่วนบนภาพ QR Code และเวลาในการถอดรหัสภาพ QR Code ผลลัพธ์ที่ได้ด้วย Zxing Library (สมการที่ใช้ในการคำนวณเวลาในการประมวลผลอธิบายอยู่ในหัวข้อที่ 2.9.2)

บทที่ 4

การทดลองและผลการทดลอง

ในบทนี้จะกล่าวถึงเครื่องมือที่ใช้ในการทดลอง ผลการทดลองของกระบวนการจัดรอยขีดข่วนบนภาพคิวอาร์โค้ด โดยทำการทดสอบประสิทธิภาพโดยการคำนวณค่า NC และ Time เปรียบเทียบกัน

4.1 เครื่องมือที่ใช้ในการทดลอง

4.1.1 ซอฟต์แวร์ที่ใช้ในการเขียนโปรแกรมคอมพิวเตอร์ คือ Xcode Version 4.5.2 (4G2008a) ซึ่งเป็นเครื่องมือที่ใช้ในการพัฒนาโปรแกรมบนระบบปฏิบัติการ Mac OSX ร่วมกับ OpenCV [18] (Open Computer Vision) ซึ่งเป็นไลบรารีที่ใช้ในการพัฒนาระบบประมวลผลภาพ โดยสามารถนำไลบรารีนี้ไปรวมกับการพัฒนาโปรแกรมโดยใช้ภาษา C, C++, Java และ Python ซึ่ง OpenCV เป็นโอเพนซอร์สที่พัฒนาโดยบริษัท อินเทล

4.1.2 ภาษาคอมพิวเตอร์ที่ใช้ในการเขียนโปรแกรมคอมพิวเตอร์ คือ C/C++

4.1.3 ฮาร์ดแวร์ที่ใช้ในการทดสอบ คือ ซีพียู Intel Core i7 2.7 GHz 4 MB Cache. หน่วยความจำหลัก 4096 MB ฮาร์ดดิสก์ 500 GB ระบบปฏิบัติการ Mac OS X 10.7.5 (11G63b)

4.1.4 รูปภาพที่ใช้ในการทดลองมีดังนี้

4.1.4.1 ภาพต้นฉบับ (Original Image) QR Code คือ ภาพ QR Code หลัก ที่มีความสมบูรณ์ สกูล PNG มีขนาดกว้าง 216 พิกเซล และสูง 216 พิกเซล ใช้เป็นภาพต้นฉบับในการคำนวณค่า NC




4.1.4.2 ภาพที่นำเข้า (Input Image) คือ ภาพ QR Code ที่มีรอยขีดข่วนขนาด 376x376 พิกเซลจำนวนทั้งหมด 240 ภาพ ได้แก่ สีดำ สีแดง เขียว และน้ำเงิน ซึ่งรอยขีดข่วนแต่ละสีจะมีทั้งสิ้น 15 รูปแบบ สามารถดูตัวอย่างของรอยขีดข่วนในภาคผนวก ก โดยสามารถแบ่งรอยขีดข่วนได้ดังนี้







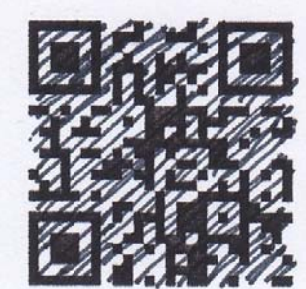


- ก. ภาพรอยขีดข่วนที่เกิดจากปากกาขนาด 0.4 mm จำนวน 60 ตัวอย่าง
- ข. ภาพรอยขีดข่วนที่เกิดจากปากกาขนาด 2 mm จำนวน 60 ตัวอย่าง
- ค. ภาพรอยขีดข่วนที่เกิดจากปากกาขนาด 3.5 mm จำนวน 60 ตัวอย่าง
- ง. ภาพรอยขีดข่วนที่เกิดจากปากกาขนาด 5 mm จำนวน 60 ตัวอย่าง

4.2 ผลการทดลอง

การทดลองนี้จะทำการนำเข้าภาพ QR Code ที่ได้รับความเสียหายจากรอยขีดข่วนขนาดของภาพที่นำเข้าคือ ภาพขนาด 376 x 376 พิกเซล ภาพที่นำเข้าเป็นภาพที่ได้จากการถ่ายภาพผ่านกล้องของโทรศัพท์มือถือซึ่งเป็นภาพถ่ายในมุมมองตรงจากนั้นนำภาพที่นำเข้ามาผ่านกระบวนการกำจัดรอยขีดข่วน ทำการวัดประสิทธิภาพด้วยการวัดค่า NC, เวลาที่ใช้ในการประมวลผล และเวลาที่ใช้ในการถอดรหัสโดยใช้ Zxing โดยใช้เวลาที่เครื่องคอมพิวเตอร์ ภาพตัวอย่างผลลัพธ์บางส่วนของผลการทดลองจำแนกตามสีและขนาดของหัวปากกาที่มีในรอยขีดข่วนแสดงดังนี้

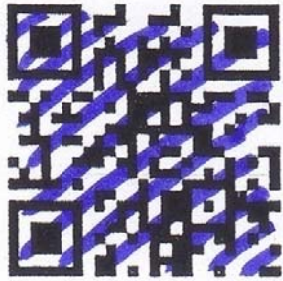
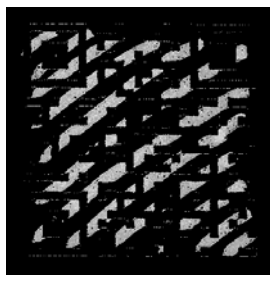




ตารางที่ 4-1 แสดงผลลัพธ์ของรอยขีดข่วนจากปากกาขนาด 0.4 mm







No.	Input Image		Scratches Pattern		Output Image
1					
	Color	Blue	NC		0.9986
			Times	Process	29.2270 ms.
				Decode	1.6230 ms.
	Nib size	0.4 mm.	Decode QR Code		Yes

2				NC		1		
				Color	Green	Times	Process	27.0410 ms.
							Decode	1.6230 ms.
				Nib size	0.4 mm	Decode QR Code		Yes
3				NC		1		
				Color	Red	Times	Process	54.23330 ms.
							Decode	1.0710 ms.
				Nib size	0.4 mm	Decode QR Code		Yes
4				NC		1		
				Color	Black	Times	Process	54.23330 ms.
							Decode	1.0710 ms.
				Nib size	0.4 mm	Decode QR Code		Yes


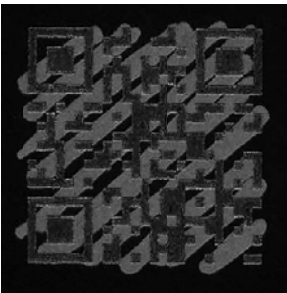




Color	Black	NC		0.9986
		Times	Process	29.7990 ms.
Nib size	0.4 mm.		Decode	1.6440 ms.




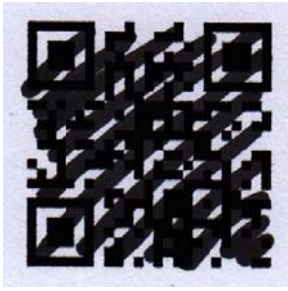
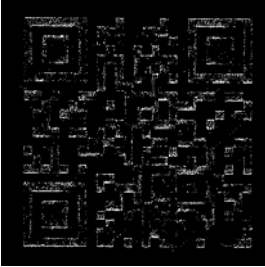

ตารางที่ 4-2 แสดงผลลัพธ์ของรอยขีดข่วนจากปากกาขนาด 2.0 mm

No.	Input Image	Scratches Pattern	Output Image					
1								
				Color	Blue	NC	0.9986	
				Nib size	2 mm.	Times	Process	27.1150 ms.
						Decode	1.7690 ms.	
Decode QR Code		Yes						
2								
				Color	Green	NC	1	
				Nib size	2 mm	Times	Process	34.2180 ms.
						Decode	1.6230 ms.	
Decode QR Code		Yes						





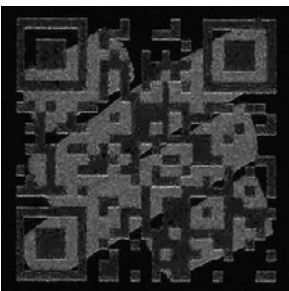

3						
	Color	Red	NC		1	
	Nib size	2 mm	Times	Process	28.6450 ms.	
				Decode	1.0800 ms.	
			Decode QR Code		Yes	
4						
	Color	Black	NC		0.9986	
	Nib size	0.4 mm.	Times	Process	29.7990 ms.	
				Decode	1.6440 ms.	
			Decode QR Code		No	







ตารางที่ 4-3 แสดงผลลัพธ์ของรอยขีดข่วนจากปากกาขนาด 3.5 mm

No.	Input Image		Scratches Pattern		Output Image
1					
	Color	Blue	NC		1.00
			Times	Process	25.4000 ms.
	Nib size	3.5 mm.		Decode	1.5250 ms.
			Decode QR Code		Yes
2					
	Color	Green	NC		1
			Times	Process	32.9590 ms.
	Nib size	3.5 mm		Decode	1.0820 ms.
			Decode QR Code		Yes

3						
	Color	Red	NC		1	
	Nib size	3.5 mm	Times	Process	27.1710 ms.	
				Decode	1.5250 ms.	
			Decode QR Code		Yes	
4						
	Color	Black	NC		0.7799	
	Nib size	3.5 mm.	Times	Process	28.5890 ms.	
				Decode	1.4470 ms.	
			Decode QR Code		No	

ตารางที่ 4-4 แสดงผลลัพธ์ของรอยขีดข่วนจากปากกาขนาด 5 mm

No.	Input Image	Scratches Pattern	Output Image
1			
	Color	Blue	NC
			1.00
			30.0110 ms.
	Nib size	3.5 mm.	0.0300 ms.
			Decode QR Code
			Yes
2			
	Color	Green	NC
			1
			29.0610 ms.
	Nib size	3.5 mm	1.1180 ms.
			Decode QR Code
			Yes

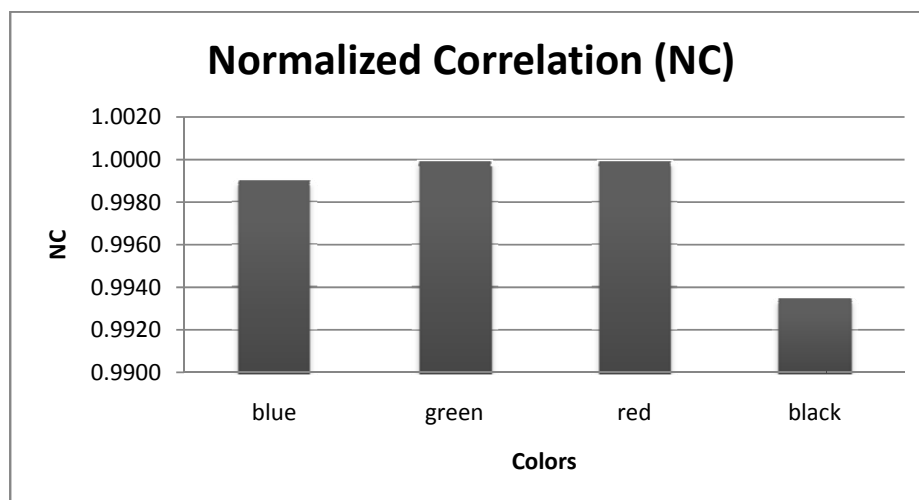
3						
	Color	Red	NC		0.9945	
	Nib size	3.5 mm	Times	Process	52.5170 ms.	
				Decode	2.2120 ms.	
	Decode QR Code			Yes		
4						
	Color	Black	NC		0.8837	
	Nib size	3.5 mm.	Times	Process	30.0610 ms.	
				Decode	1.4100 ms.	
	Decode QR Code			No		

สำหรับผลการทดลองในรอยขีดข่วน สามารถดูรายละเอียดเพิ่มเติมในภาคผนวก ข จากการทดลองสามารถวิเคราะห์ผลการทดลองโดยการหาค่าเฉลี่ยของ NC และ Time ของแต่ละกลุ่มสีและขนาดของปากกาได้ดังนี้

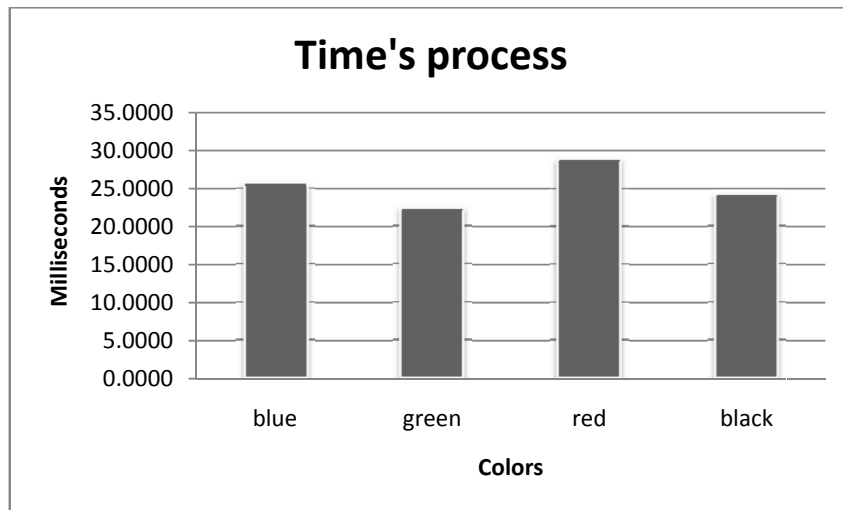
4.2.1 ผลการทดลองในรอยขีดข่วน 0.4 mm โดยทำการหาค่าเฉลี่ยของค่า NC ค่าเวลาที่ใช้ในการประมวลผล (Time 's process) เวลาที่ใช้ในการถอดรหัส (Time's Decode) เปอร์เซ็นต์ผลการถอดรหัสสำเร็จ (Success Rate of Decoding) และเปอร์เซ็นต์การถอดรหัสผิดพลาด(Failure Rate of Docoding) ของผลลัพธ์ที่ได้โดยแสดงค่าเฉลี่ยดังตาราง 4-5 เมื่อเรานำข้อมูลมาสร้างแผนภูมิแสดงค่า NC ดังภาพที่ 4.1, เวลาที่ใช้ในการประมวลผล ดังภาพที่ 4.2 และเวลาที่ใช้ในการถอดรหัสดังภาพที่ 4.3

ตารางที่ 4-5 ผลการทดลองเฉลี่ยในรอยขีดข่วนขนาด 0.4 mm

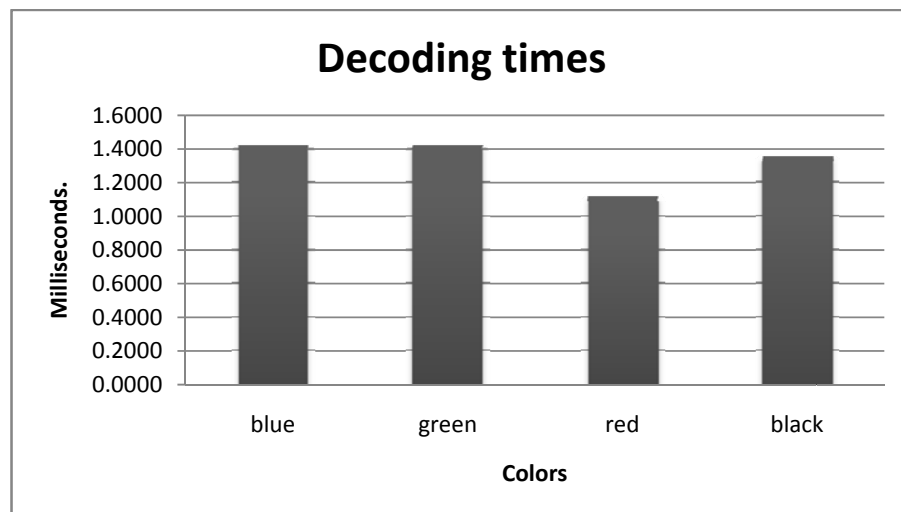
Colors	NC	Time's process	Time 's Decode	Success Rate of Decoding (%)	Failure Rate of Docoding (%)
Blue	0.9990	25.6792	1.4199	100	0
Breen	0.9999	22.4389	1.4205	100	0
Red	0.9999	28.9458	1.1157	100	0
Black	0.9934	24.2836	1.3509	86.67	13.33



ภาพที่ 4-1 แผนภาพค่า NC เฉลี่ยในรอยขีดข่วนขนาด 0.4 mm



ภาพที่ 4-2 แผนภาพค่าเวลาที่ใช้ในการประมวลผลเฉลี่ยในรอยขีดข่วนขนาด 0.4 mm

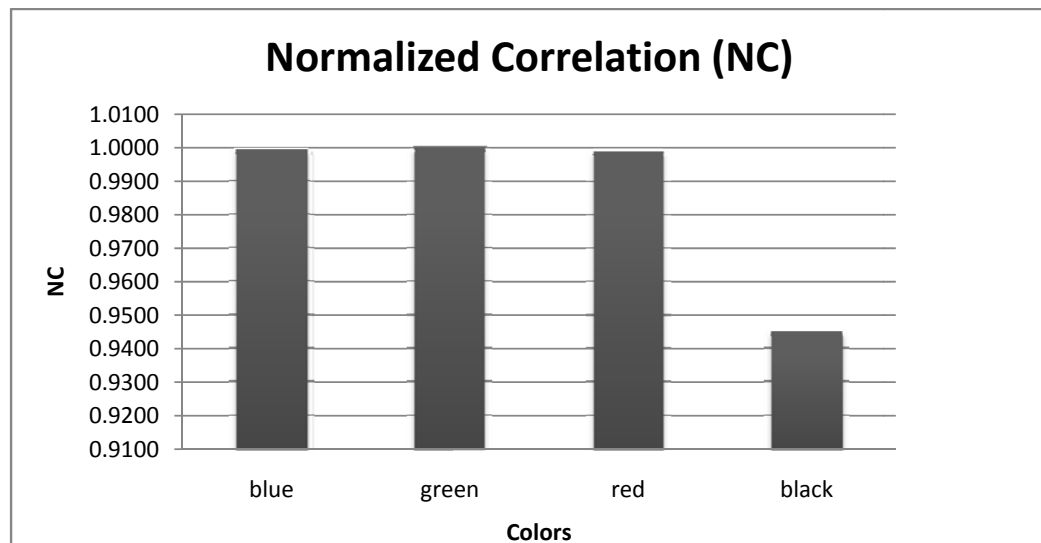


ภาพที่ 4-3 แผนภาพค่าเวลาที่ใช้ในการถอดรหัสเฉลี่ยในรอยขีดข่วนขนาด 0.4 mm

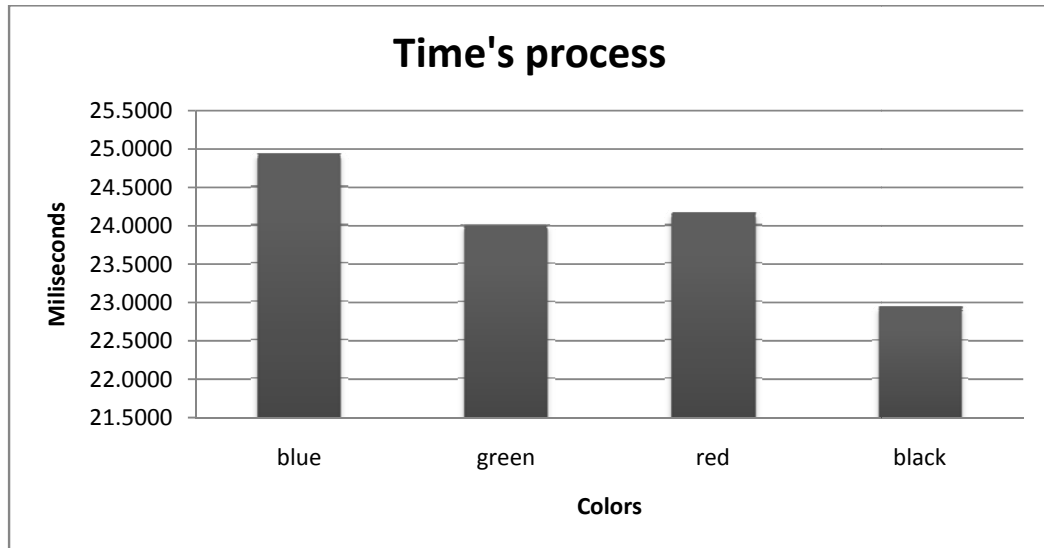
4.2.2 ผลการทดลองในรอยขีดข่วน 2.0 mm โดยทำการหาค่าเฉลี่ยของค่า NC ค่าเวลาที่ใช้ในการประมวลผล (Time's process) เวลาที่ใช้ในการถอดรหัส (Time's Decode) เปอร์เซ็นต์การถอดรหัสสำเร็จ (Success Rate of Decoding) และเปอร์เซ็นต์การถอดรหัสผิดพลาด (Failure Rate of Decoding) ของผลลัพธ์ที่ได้ โดยแสดงค่าเฉลี่ยดังตาราง 4-6 เมื่อนำข้อมูลมาสร้างแผนภูมิแสดงค่า NC ดังภาพที่ 4.4, เวลาที่ใช้ในการประมวลผล ดังภาพที่ 4.5 และเวลาที่ใช้ในการถอดรหัส ดังภาพที่ 4.6

ตารางที่ 4-6 ผลการทดลองเฉลี่ยในรอยขีดข่วนขนาด 2.0 mm

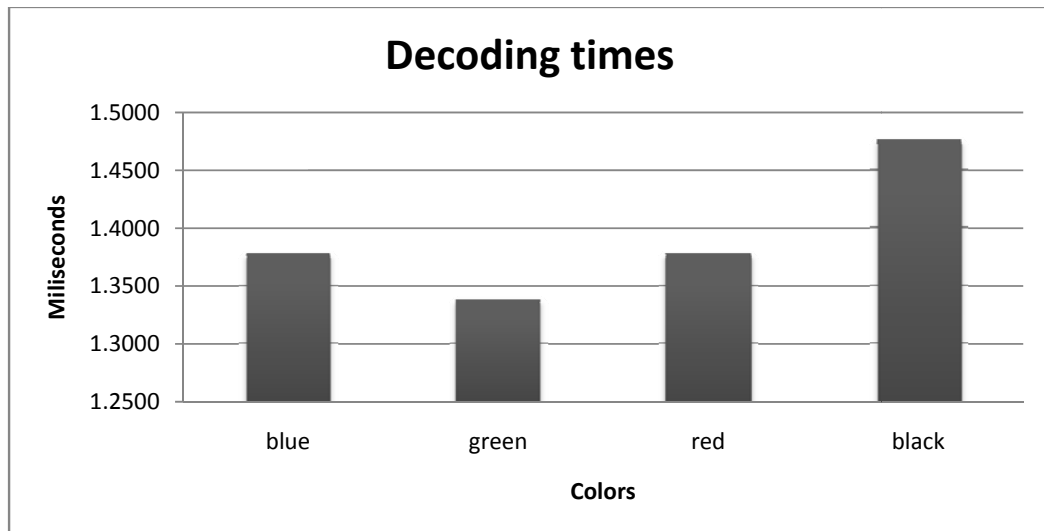
Colors	NC	Time's process	Time 's Decode	Success Rate of Decoding (%)	Failure Rate of Decoding (%)
Blue	0.9992	24.9291	1.3781	100	0
Green	1.0000	24.0059	1.3382	100	0
Red	0.9991	24.1644	1.3778	100	0
Black	0.9449	22.9410	1.4763	0	100.00



ภาพที่ 4-4 แผนภาพค่า NC เฉลี่ยในรอยขีดข่วนขนาด 2 mm



ภาพที่ 4-5 แผนภาพค่าเวลาที่ใช้ในการประมวลผลเฉลี่ยในรอยขีดข่วนขนาด 2 mm

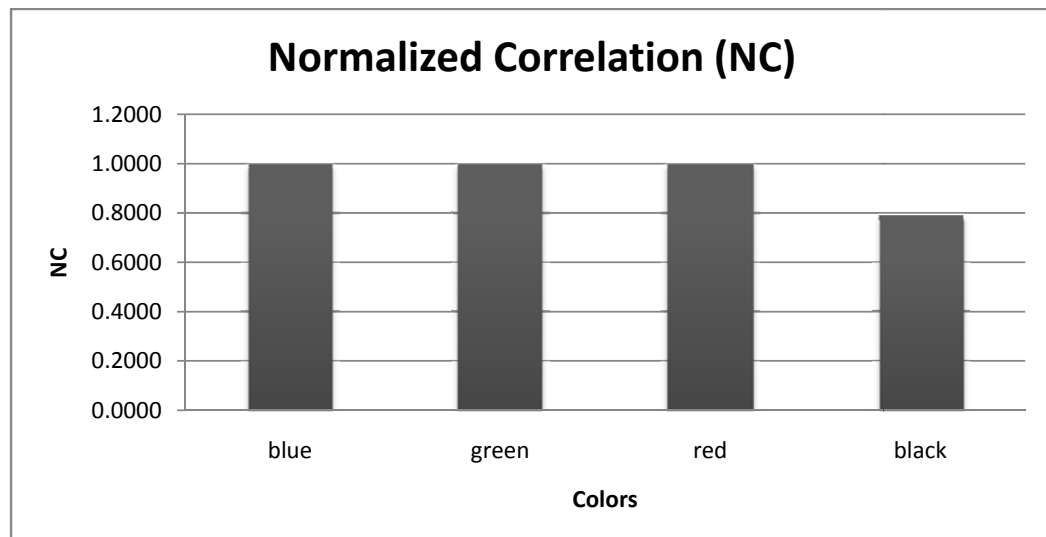


ภาพที่ 4-6 แผนภาพค่าเวลาที่ใช้ในการถอดรหัสเฉลี่ยในรอยขีดข่วนขนาด 2 mm

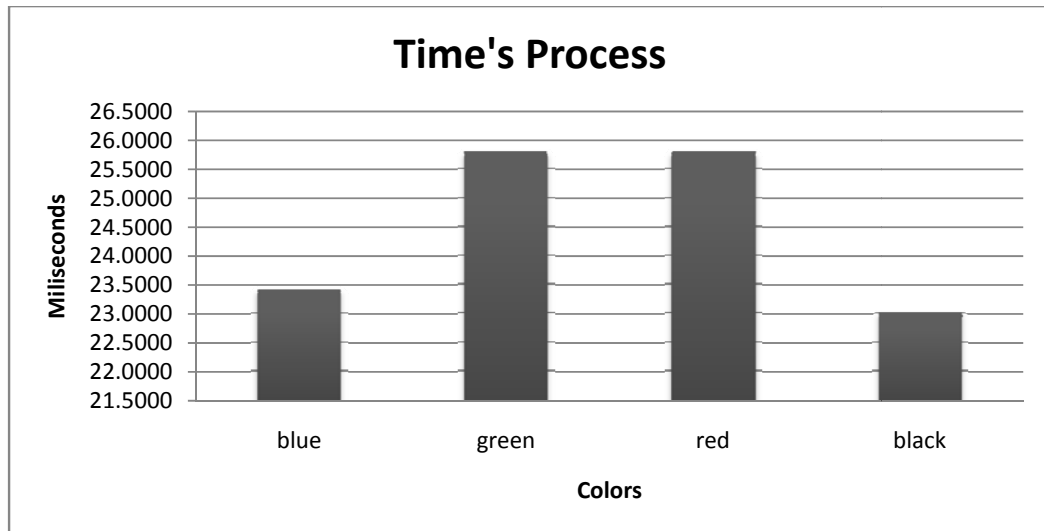
4.2.2 ผลการทดลองในรอยขีดข่วน 3.5 mm โดยทำการหาค่าเฉลี่ยของค่า NC ค่าเวลาที่ใช้ในการประมวลผล (Time's process) เวลาที่ใช้ในการถอดรหัส (Time's Decode) เปอร์เซ็นต์ผลการถอดรหัสสำเร็จ (Success Rate of Decoding) และเปอร์เซ็นต์การถอดรหัสผิดพลาด (Failure Rate of Decoding) ของผลลัพธ์ที่ได้ โดยแสดงค่าเฉลี่ยดังตาราง 4-7 เมื่อนำข้อมูลมาสร้างแผนภูมิแสดงค่า NC ดังภาพที่ 4.7, เวลาที่ใช้ในการประมวลผล ดังรูปที่ 4.8 และเวลาที่ใช้ในการถอดรหัส ดังภาพที่ 4.9

ตารางที่ 4-7 ผลการทดลองเฉลี่ยในรอยขีดข่วนขนาด 3.5 mm

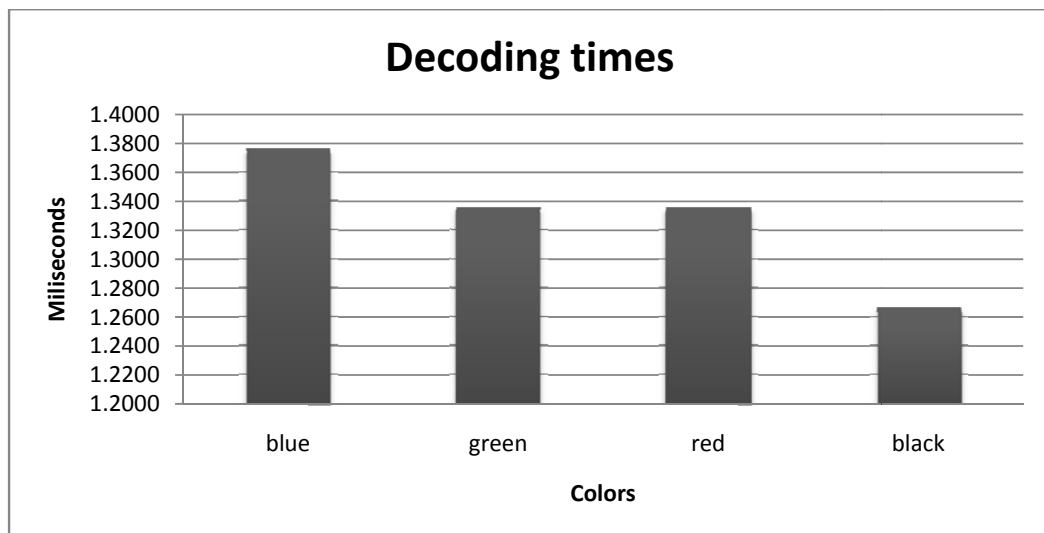
Colors	NC	Time's process	Time 's Decode	Success Rate of Decoding (%)	Failure Rate of Decoding (%)
Blue	0.9998	23.4241	1.3761	100	0
Green	1.0000	25.8081	1.3357	100	0
Red	1.0000	25.8081	1.3357	100	0
Black	0.7886	23.0217	1.2664	0	100



ภาพที่ 4-7 แผนภาพค่า NC เฉลี่ยในรอยขีดข่วนขนาด 3.5 mm



ภาพที่ 4-8 แผนภาพค่าเวลาที่ใช้ในการประมวลผลเฉลี่ยในรอยขีดข่วนขนาด 3.5 mm

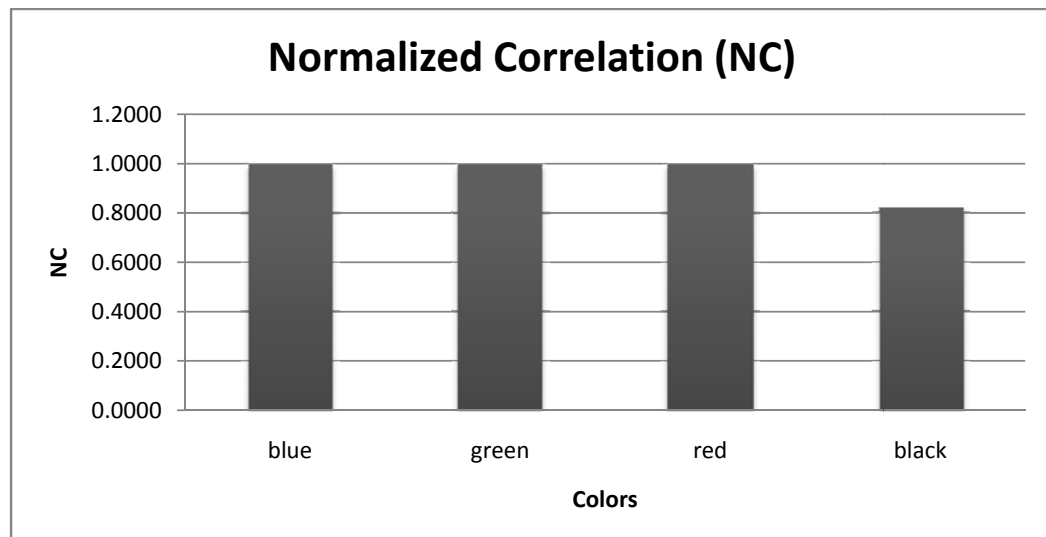


ภาพที่ 4-9 แผนภาพค่าเวลาที่ใช้ในการถอดรหัสเฉลี่ยในรอยขีดข่วนขนาด 3.5 mm

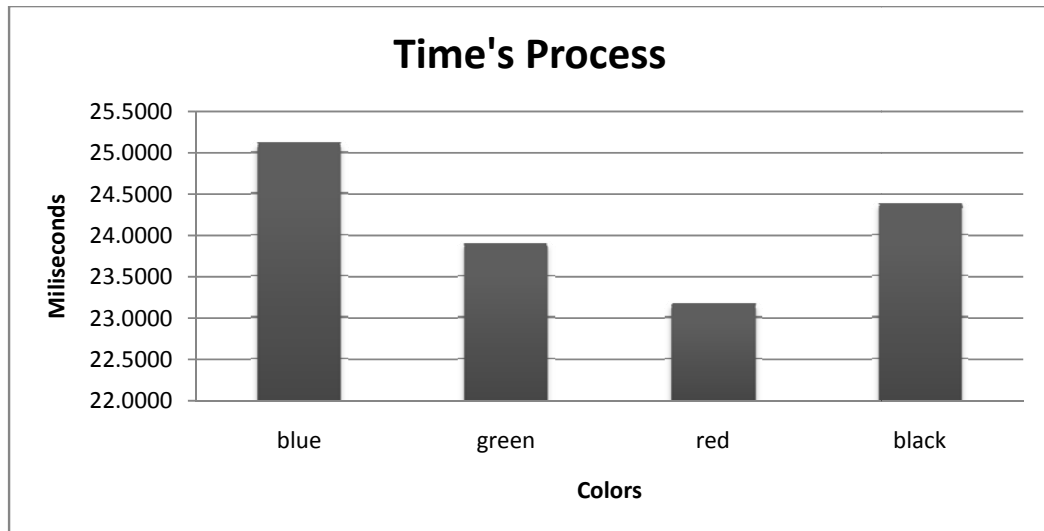
4.2.2 ผลการทดลองในรอยขีดข่วน 5 mm โดยทำการหาค่าเฉลี่ยของค่า NC ค่าเวลาที่ใช้ในการประมวลผล (Time's process) เวลาที่ใช้ในการถอดรหัส (Time's Decode) เปอร์เซ็นต์ผลการถอดรหัสสำเร็จ (Success Rate of Decoding) และเปอร์เซ็นต์การถอดรหัสผิดพลาด (Failure Rate of Decoding) ของผลลัพธ์ที่ได้ โดยแสดงค่าเฉลี่ยดังตาราง 4-8 เมื่อนำข้อมูลมาสร้างแผนภูมิแสดงค่า NC ดังภาพที่ 4.10, เวลาที่ใช้ในการประมวลผล ดังภาพที่ 4.11 และเวลาที่ใช้ในการถอดรหัสดังภาพที่ 4.12

ตารางที่ 4-8 ผลการทดลองเฉลี่ยในรอยขีดข่วนขนาด 5 mm

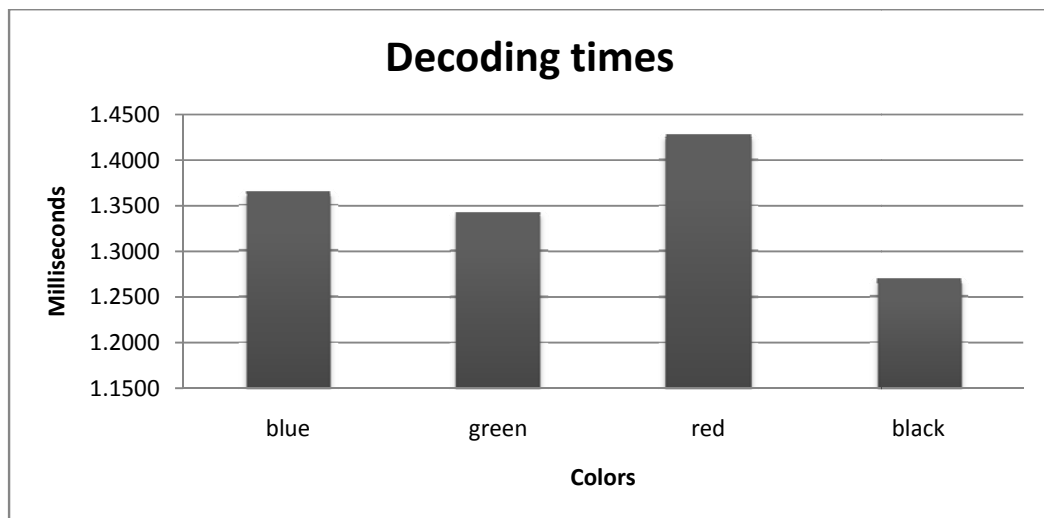
Colors	NC	Time's process	Time 's Decode	Success Rate of Decoding (%)	Failure Rate of Decoding (%)
Blue	0.9998	23.4241	1.3761	100	0
Green	1.0000	25.8081	1.3357	100	0
Red	1.0000	25.8081	1.3357	100	0
Black	0.7886	23.0217	1.2664	0	100



ภาพที่ 4-10 แผนภาพค่า NC เฉลี่ยในรอยขีดข่วนขนาด 5 mm



ภาพที่ 4-11 แผนภาพค่าเวลาที่ใช้ในการประมวลผลเฉลี่ยในรอยขีดข่วนขนาด 5 mm



ภาพที่ 4-12 แผนภาพค่าเวลาที่ใช้ในการถอดรหัสเฉลี่ยในรอยขีดข่วนขนาด 5 mm

บทที่ 5

สรุปผลการวิจัย

ในบทนี้จะกล่าวถึงผลสรุปงานวิจัย ปัญหาที่พบ และขอเสนอแนะอันจะเป็นแนวทางในการพัฒนาวิธีการปรับปรุงภาพ QR Code ที่มีความเสียหายจากรอยขีดข่วนเพื่อให้สามารถอ่านเพื่อถอดรหัสได้

5.1 บทสรุป

เพื่อศึกษาออกแบบและนำเสนอวิธีการที่เหมาะสมในการปรับปรุงภาพ QR Code ที่มีความเสียหายจากรอยขีดข่วน โดยการใช้วิธีการประมวลผลภาพเพื่อให้ได้ภาพ QR Code ที่สามารถทำการถอดรหัสได้ ซึ่งบทสรุปจะแบ่งออกเป็น 2 ส่วน ดังนี้

5.1.1 บทสรุปตามเกณฑ์ขนาดรอยขีดข่วนที่ใช้ในการทดลอง

จากผลการทดลองใน 4.2.1 ซึ่งรอยขีดข่วนเกิดจากปากกาขนาด 0.4 mm ทั้ง 4 สี ได้แก่ สีน้ำเงินเข้ม สีเขียวเข้ม สีแดงเข้ม และสีดำ จากผลลัพธ์ที่ได้เมื่อพิจารณาที่ค่า NC พบว่าจากรอยขีดข่วนใน สีน้ำเงิน สีเขียว และสีแดง ให้ค่า NC เฉลี่ยที่เท่ากับภาพต้นฉบับซึ่งส่งผลให้กระบวนการอ่านเพื่อถอดรหัส QR Code ทำได้อย่างมีประสิทธิภาพในทุกกรณีของทั้ง 3 สี แต่ในรอยขีดข่วนที่มีสีดำจะมีค่า NC เฉลี่ยที่ต่ำกว่าทั้ง 3 สีที่กล่าวข้างต้น เมื่อนำผลลัพธ์ที่ได้จากการกำจัดรอยขีดข่วนสีดำไปทำการถอดรหัสซึ่งภาพที่สามารถถอดรหัสได้มีทั้งสิ้น 13 ภาพ จากทั้งหมด 15 ภาพ คิดเป็น 86.67 เปอร์เซ็นต์ที่สามารถถอดรหัสได้ สำหรับเวลาที่ใช้ในการประมวลผลเฉลี่ยจะอยู่ที่ 24 – 28 milliseconds

จากผลการทดลองใน 4.2.2 ซึ่งเป็นรอยขีดข่วนที่เกิดจากปากกาขนาด 2.0 mm ทั้ง 4 สี ได้แก่ สีน้ำเงินเข้ม สีเขียวเข้ม สีแดงเข้ม และสีดำ เมื่อพิจารณาที่ผลการทดลองพบว่า ผลลัพธ์ในสีดำให้ค่า NC ที่ต่ำกว่าสีอื่นๆ ซึ่งส่งผลให้การถอดรหัสภาพผลลัพธ์จากรอยขีดข่วนสีดำเกิดความล้มเหลวทั้ง 15 กรณี แต่เมื่อนำผลลัพธ์ที่ได้จากสีแดง สีเขียว และสีน้ำเงินที่ให้ค่าเฉลี่ย NC อยู่ระหว่าง 0.9991 ถึง 1 ซึ่งสามารถถอดรหัสผลลัพธ์ที่ได้อย่างมีประสิทธิภาพ สำหรับเวลาที่ใช้ในการประมวลผลเฉลี่ยอยู่ที่ 22 ถึง 24 milliseconds

จากผลการทดลองใน 4.2.3 ซึ่งเป็นรอยขีดข่วนที่เกิดจากปากกาขนาด 3.5 mm ทั้ง 4 สี ได้แก่ สีน้ำเงินเข้ม สีเขียวเข้ม สีแดงเข้ม และสีดำ เมื่อพิจารณาที่ผลการทดลองสามารถสรุปได้คือค่า NC ที่ได้ที่ดีที่สุดคือค่าที่ได้จากการกำจัดรอยขีดข่วนสีแดงและสีเขียว ซึ่งให้ค่า NC เท่ากับ 1 ซึ่งเป็นค่าที่เหมือนกับต้นฉบับ 100 % ส่วนในผลลัพธ์ที่ได้จากรอยขีดข่วนสีน้ำเงินให้ค่า NC

ที่ 0.9998 ซึ่งใกล้เคียงภาพต้นฉบับ ทำให้สามารถทำการถอดรหัสภาพ QR Code ได้ทั้ง 3 สี แต่ในผลลัพธ์ที่ได้จากรอยขีดข่วนสีดำให้ค่า NC ที่ต่ำมากถึง 0.7886 ซึ่งส่งผลให้การถอดรหัสไม่สามารถทำได้ในทุกกรณี

และจากผลการทดลองใน 4.2.4 ซึ่งเป็นรอยขีดข่วนขนาด 5 mm ทั้ง 4 สีซึ่งประกอบไปด้วยสีน้ำเงินเข้ม สีแดงเข้ม สีเขียวเข้ม และสีดำ จากการพิจารณาค่า NC ในผลลัพธ์ที่ได้จากรอยขีดข่วนสีคามีค่าเฉลี่ยอยู่ที่ 0.8224 ซึ่งผลลัพธ์ที่ได้มีความต่างจากภาพต้นฉบับมากส่งผลให้กระบวนการถอดรหัสภาพ QR Code ที่ได้ล้มเหลวในทุกกรณี แต่ในส่วนกรณีของสีอื่นได้แก่สีน้ำเงินเข้ม สีแดงเข้ม และสีเขียวให้ค่า NC ที่เข้าใกล้ต้นฉบับมาก ซึ่งอยู่ระหว่าง 0.9992 ถึง 1 ทำให้กระบวนการถอดรหัสสามารถทำได้อย่างมีประสิทธิภาพ เวลาที่ใช้ในการประมวลผลเฉลี่ยอยู่ที่ 23 – 25 milliseconds และเวลาที่ใช้ในการถอดรหัสอยู่ที่ 1.2695 – 1.4278 milliseconds

5.1.2 บทสรุปตามสีของรอยขีดข่วนที่ใช้ในการทดลอง

ในกรณีรอยขีดข่วนที่มีสีแดง สีเขียว และสีน้ำเงิน หรือสีอื่นๆ ที่มีค่าอิมสี ขนาดของรอยขีดข่วนที่เกิดจากปากกาสีเหล่านี้สามารถถูกขจัดโดยใช้วิธีที่นำเสนอในงานวิจัยนี้ซึ่งแบบจำลองสี HSV ช่วยให้การแยกแยะรอยสีเหล่านี้ได้อย่างมีประสิทธิภาพ ในภาพผลลัพธ์ QR Code ที่ได้มีค่าใกล้เคียงจากภาพต้นฉบับมากส่งผลให้การอ่านเพื่อถอดรหัสทำได้ ส่วนในกรณีของรอยขีดข่วนสีค่านั้นขนาดของรอยจะมีผลต่อกระบวนการ ที่นำเสนอ ในรอยขีดข่วนสีดำขนาด 0.4 mm ผลลัพธ์ที่ได้บางส่วนสามารถถอดรหัสได้ซึ่งคิดเป็น 80 % ของตัวอย่างในการทดลองแต่เมื่อทำการเพิ่มขนาดรอยขีดข่วนเป็น 2 mm, 3.5 mm และ 5mm ผลลัพธ์ที่ได้ไม่สามารถถอดรหัสได้ ทั้งนี้เนื่องจากรอยขีดข่วนสีดำที่มีขนาดใหญ่กว่าโมดูลของ QR Code อีกทั้งในความเข้มที่มีสีเดียวกันกับโมดูลสีดำของภาพ QR Code ซึ่งสีดำกับสีขาวไม่มีค่าความอิมตัวในแบบจำลองสี HSV ดังนั้นเมื่อรอยขีดข่วนที่มีสีดำจึงทำให้การแยกแยะรอยขีดข่วนทำได้ไม่ดีนัก สำหรับเวลาที่ใช้ในการประมวลผลจากการทดลองทั้งสิ้น 240 เวลาที่ใช้ต่อกรณีเฉลี่ยอยู่ที่ 24.47 milliseconds และเวลาเฉลี่ยที่ใช้ในการถอดรหัส QR Code ผลลัพธ์อยู่ที่ 1.35 milliseconds. ซึ่งเมื่อเปรียบเทียบกับเวลาของการถอดรหัสคิวอาร์โค้ดปกติที่ใช้เวลาโดยประมาณอยู่ที่ 1.10 ถึง 1.40 milliseconds จากการทดลองทั้งหมดแสดงให้เห็นว่ากระบวนการที่นำเสนอสามารถกำจัดรอยขีดข่วนที่ไม่ใช่สีดำได้อย่างมีประสิทธิภาพ ซึ่งหากนำกระบวนการที่นำเสนอมาเป็น Pre-Processing ในการเตรียมภาพ QR Code ก่อนการถอดรหัสก็จะสามารถทำให้การถอดรหัสภาพ QR Code ที่เสียหายจากรอยขีดข่วนได้ในเวลาที่เหมาะสม และได้ผลลัพธ์จากการถอดรหัสที่มีประสิทธิภาพ

5.2 ข้อเสนอแนะ

เนื่องด้วยในงานวิจัยนี้มีการกำหนดเวอร์ชันของ QR Code ที่ใช้ในการทดลอง คือ QR Code เวอร์ชัน 2 ดังนั้นในการนำวิธีการที่นำเสนอไปใช้ใน QR Code เวอร์ชันอื่นๆ จำเป็นต้องมีการกำหนดจำนวนโมดูลที่สอดคล้องกับเวอร์ชันนั้นๆ ในขั้นตอนของการการแบ่งพื้นที่ย่อยของ QR Code (Grid Generating) จากผลการทดลองพบว่าวิธีการที่ได้พัฒนาขึ้นมีข้อจำกัดบางประการ ควรปรับปรุงโดยให้มีการทำวิจัยและศึกษาต่อเพื่อให้มีประสิทธิภาพมากยิ่งขึ้น เช่น ในขั้นตอนของการเติมเต็มโมดูลภาพใน QR Code (Reconstruction QR Code) ซึ่งในงานวิจัยนี้ใช้การปรับค่าในพื้นที่ย่อยตามจำนวนของกลุ่มสีที่มีจำนวนมากในพื้นที่ย่อยนั้น ๆ หากมีการนำกระบวนการทาง Machine Learning มาใช้ในตัดสินใจเลือกการปรับค่าสีของโมดูลจะทำให้ได้ผลลัพธ์ที่ได้มีประสิทธิภาพมากยิ่งขึ้น สำหรับในกระบวนการจัดรอยขีดข่วนที่นำเสนอในงานวิจัยนี้สามารถกำจัดรอยขีดข่วนได้อย่างมีประสิทธิภาพในกรณีที่รอยขีดข่วนที่ทำความเสียหายต่อ QR Code มีสีแดง สีเขียว สีน้ำเงิน หรือสีอื่นๆ ที่เกิดจากการผสมสีเหล่านี้ แต่ในกรณีที่รอยขีดข่วนมีสีดำจำเป็นต้องมีการศึกษาเพิ่มเติมเพื่อหาวิธีที่จะมาใช้ในการแก้ปัญหาต่อไป

รายการอ้างอิง

- [1] สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ. แนะนำเทคโนโลยีบาร์โค้ด [ออนไลน์]. 2554. แหล่งที่มา : <http://www.nstda.or.th/nstda-knowledge/2866-2d-barcode> [19 เมษายน 2556]
- [2] Lei Fu. Design of QR Code-based Mall Shopping Guide System. Proceedings of the International Conference on Information Science and Technology (2011) : 450-453.
- [3] ISO/IEC. Information technology-Automatic identification and data capture QR Code 2005 bar code symbology specification. First Edition.18004. Switzerland: ISO copyright office, 2005.
- [4] Denso wave incorporated. About QR Code [Online]. 1994. Available from: <http://www.denso-wave.com/qrcode/index-e.html> [2013, April 20]
- [5] Namiotand, D. and Sneps-Sneppe, M. A New Approach to Advertising in Social Networks Business-Centric Check-ins. Proceedings of the 15th International Conference on Intelligence in Next Generation Network (2011) : 92-96.
- [6] QRreaders. QR Code Boarding Passes [Online]. 2010. Available from : <http://www.qrreaders.org/boarding-passes.html> [2013, March 26]
- [7] Maria Ebling and Ramon Caceres. Bar Codes Everywhere You Look. Proceedings of the IEEE CS (2010) : (4-5).
- [8] สุพัต รุ่งเรืองศิลป์. การพัฒนากิวอาร์โค้ดร่วมกับลายน้ำดิจิทัลเพื่อซ่อนข้อมูล. วิทยานิพนธ์ ปริญญาโทบริหารธุรกิจ สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2554.
- [9] Darrin Cardani. Adventures in HSV Space. The vision and image sciences laboratory Technion-Israel Institute of Technology .

- [10] Rafael Gonzalez and Richard Woods. Digital image processing. Prentice-Hall Publishing, 2001.
- [11] นริศ หนูหอม. การฝังลายน้ำอเนกประสงค์บนรูปภาพดิจิทัลเพื่อตรวจสอบรับรองความเป็นต้นฉบับและพิสูจน์ความเป็นเจ้าของลิขสิทธิ์. วิทยานิพนธ์ปริญญาโท สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2553.
- [12] Xiong, Z., Guo-dong, L. and Jian-min, W. Study on the Sequence of Steps in the QR Code Image Preprocessing. Proceedings of the 2nd International Conference on Future Computer and Communication (2010) : 446 - 449.
- [13] Chung-Hua, C., De-Nian, Y. and Ming-Syan, C. Image Stabilization for 2D Barcode in Handheld Devices. Graduate Institute of Communication Engineering National Taiwan University (2007) : 697-706.
- [14] Yuji, K., Daisuke, D., Tomokazu, T., Ichiro, I. and Hiroshi M. Low resolution QR-code recognition by applying super-resolution using the property of QR-codes. Proceedings of the International Conference on Document Analysis and Recognition (2011) : 992-996.
- [15] Yinghui, Z., Tianlei, G., Deguang, L. and Huaqi L. An Improved Binarization Algorithm of QR Code Image. Proceedings of the Consumer Electronics Communications and Networks (2012) : 2376-2379.
- [16] Yue, L., Ju, Y. and Mingjun, L. Recognition of QR Code with Mobile Phone. Proceedings of the Chinese Control and Decision Conference (2008) : 203-206.
- [17] ZXing. ZXing Multi-format 1D/2D barcode image processing library with clients [Online]. Available from : <https://code.google.com/p/zxing> [2013, March 26]
- [18] OpenCV. OpenCV Open Source Computer Vision [Online]. Available from : <http://opencv.org> [2013, March 26]
















- [19] Gao JerryZeyu. Understanding 2D-BarCode Technology and Application in M-Commerce. Proceedings of the 31st International Computer Software Conference (2007):49-56
- [20] Namiotand, D. and Sneps-Sneppe, M. A New Approach to Advertising in Social Networks Business-Centric Check-ins. Proceedings of the 15th International Conference on Intelligence in Next Generation Network (2011):92-96.
- [21] QRme. QR Code Example [Online]. Available from : <http://qrme.com.au/qr-code-examples> [2013, March 26]
- [22] ประทาน โชติพันธุ์บัณฑิตย์. การซ่อนข้อมูลและความมั่นคงสำหรับเอกสารสิ่งพิมพ์. วิทยานิพนธ์ปริญญาโทบัณฑิต ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2552.
- [23] Vongpradhip S. and Homkajorn K. An Implementation of Process to Remove Scratches from QR Code with Mobile Phone. Proceeding of the International Conference on Information Technology, Electronics and Communications (January 2013):42-46.
- [24] Tobias O. and Seara R. Image Segmentation by Histogram Thresholding Using Fuzzy Sets. IEEE Transactions on Image Processing 11 (December 2002) : 1457-1465.
- [25] Ohbuchi, E., Hanaizumi, H. and Hock A. L. Barcode Readers using the Camera Device in Mobile Phones. Proceeding of the International Conference on Cyberworlds (2004).
- [26] Bi, L., Feng, Z., Liu, M. and Wang, W. Design and Implementation of the Airline Luggage Inspection System Base on Link Structure of QR Code. Internation Symposium on Electornic Comerce and Security (2008):527-530.
- [27] Homkajorn K., Ketcham, M. and Vongpradhip S. A Technique to Remove Scratches from QR Code Images. Proceeding of the International Conference on Computer and Communication Technologies (2012):127-131.
- [28] Cliff Reiter. With J. Image Processing 2 Color Spaces. APL Quote Quad (2004): 3-12.

ภาคผนวก

ภาคผนวก ก











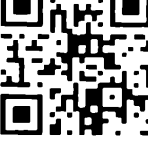









รูปแบบรอยขีดข่วนบนภาพ QR Code

ตาราง ก.1 รูปแบบรอยขีดข่วนทั้ง 15 รูปแบบ


















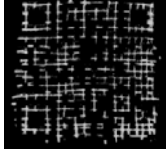


รอยขีดข่วนขนาด 3.5 mm		
		
(1)	(2)	(3)
		
(4)	(5)	(6)
		
(7)	(8)	(9)
		
(10)	(11)	(12)
		
(13)	(14)	(15)

ภาคผนวก ข
แสดงผลการทดลองเพิ่มเติม







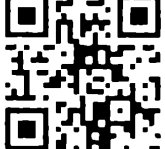













ตาราง ข.2 ตัวอย่างผลการทดลองเพิ่มเติมที่ได้ในรอยปากกาขนาด 0.4 mm สีน้ำเงิน

ภาพ QR Code ที่มีรอยขีดข่วนจากปากกาสีน้ำเงิน ขนาด 0.4 mm.					
ภาพที่นำเข้า	รอยขีดข่วน	กำจัด Noise	ผลลัพธ์	สรุป	
				NC	0.9986
				เวลาประมวลผล	29.227 ms.
				เวลาการถอดรหัส	1.623 ms.
				ถอดรหัส	Yes
				NC	0.9945
				เวลาประมวลผล	32.236 ms.
				เวลาการถอดรหัส	1.535 ms.
				ถอดรหัส	YES
				NC	1
				เวลาประมวลผล	29.396 ms.
				เวลาการถอดรหัส	2.081 ms.
				ถอดรหัส	YES
				NC	1
				เวลาประมวลผล	25.429 ms.
				เวลาการถอดรหัส	1.29 ms.
				ถอดรหัส	YES
				NC	0.9986
				เวลาประมวลผล	25.448 ms.
				เวลาการถอดรหัส	1.337 ms.
				ถอดรหัส	YES







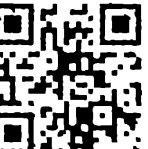













ตาราง ข. 3 ตัวอย่างผลการทดลองเพิ่มเติมที่ได้ในรอยปากกาขนาด 0.4 mm สีแดง

ภาพ QR Code ที่มีรอยขีดข่วนจากปากกาสีแดงขนาด 0.4 mm.					
ภาพที่นำเข้า	รอยขีดข่วน	กำจัด Noise	ผลลัพธ์	สรุป	
				NC	1
				เวลาประมวลผล	54.233 ms.
				เวลาการถอดรหัส	1.071 ms.
				ถอดรหัส	Yes
				NC	1
				เวลาประมวลผล	27.014 ms.
				เวลาการถอดรหัส	0.999 ms.
				ถอดรหัส	YES
				NC	1
				เวลาประมวลผล	33.024 ms.
				เวลาการถอดรหัส	2.119 ms.
				ถอดรหัส	YES
				NC	1
				เวลาประมวลผล	31.607 ms.
				เวลาการถอดรหัส	1.016 ms.
				ถอดรหัส	YES
				NC	1
				เวลาประมวลผล	29.418 ms.
				เวลาการถอดรหัส	1.788 ms.
				ถอดรหัส	YES






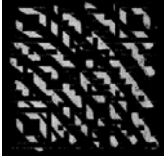











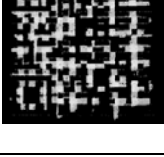


ตาราง ข.4 ตัวอย่างผลการทดลองเพิ่มเติมที่ได้ในรอยปากกาขนาด 0.4 mm สีเขียว

ภาพ QR Code ที่มีรอยขีดข่วนจากปากกาสีเขียวขนาด 0.4 mm.					
ภาพที่นำเข้า	รอยขีดข่วน	กำจัด Noise	ผลลัพธ์	สรุป	
				NC	1
				เวลาประมวลผล	27.041 ms.
				เวลาการถอดรหัส	1.564 ms.
				ถอดรหัส	Yes
				NC	1
				เวลาประมวลผล	23.091 ms.
				เวลาการถอดรหัส	2.515 ms.
				ถอดรหัส	YES
				NC	1
				เวลาประมวลผล	29.221ms.
				เวลาการถอดรหัส	1.23 ms.
				ถอดรหัส	YES
				NC	1
				เวลาประมวลผล	20.42 ms.
				เวลาการถอดรหัส	1.404 ms.
				ถอดรหัส	YES
				NC	1
				เวลาประมวลผล	21.534 ms.
				เวลาการถอดรหัส	1.367 ms.
				ถอดรหัส	YES



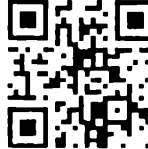



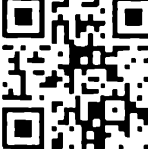


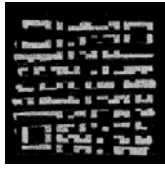
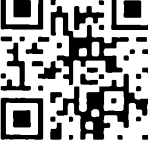









ตาราง ข. 5 ตัวอย่างผลการทดลองเพิ่มเติมที่ได้ในรอยปากกาขนาด 0.4 mm สีดำ

ภาพ QR Code ที่มีรอยขีดข่วนจากปากกาสีดำขนาด 0.4 mm.					
ภาพที่นำเข้า	รอยขีดข่วน	กำจัด Noise	ผลลัพธ์	สรุป	
				NC	0.9986
				เวลาประมวลผล	29.799 ms.
				เวลาการถอดรหัส	1.644 ms.
				ถอดรหัส	Yes
				NC	1
				เวลาประมวลผล	26.115 ms.
				เวลาการถอดรหัส	1.412 ms.
				ถอดรหัส	YES
				NC	1
				เวลาประมวลผล	27.593 ms.
				เวลาการถอดรหัส	1.385 ms.
				ถอดรหัส	YES
				NC	0.9723
				เวลาประมวลผล	20.736 ms.
				เวลาการถอดรหัส	1.478 ms.
				ถอดรหัส	NO
				NC	0.9363
				เวลาประมวลผล	27.39 ms.
				เวลาการถอดรหัส	1.284 ms.
				ถอดรหัส	NO





















ตาราง ข.6 ตัวอย่างผลการทดลองเพิ่มเติมที่ได้ในรอยปากกาขนาด 2.0 mm สีน้ำเงิน

ภาพ QR Code ที่มีรอยขีดข่วนจากปากกาสีน้ำเงินขนาด 2.0 mm.					
ภาพที่นำเข้า	รอยขีดข่วน	กำจัด Noise	ผลลัพธ์	สรุป	
				NC	0.9986
				เวลาประมวลผล	27.115 ms.
				เวลาการถอดรหัส	1.769 ms.
				ถอดรหัส	Yes
				NC	1
				เวลาประมวลผล	24.613 ms.
				เวลาการถอดรหัส	1.417 ms.
				ถอดรหัส	YES
				NC	1
				เวลาประมวลผล	28.95 ms.
				เวลาการถอดรหัส	1.529 ms.
				ถอดรหัส	YES
				NC	0.9986
				เวลาประมวลผล	31.217 ms.
				เวลาการถอดรหัส	1.475 ms.
				ถอดรหัส	YES
				NC	1
				เวลาประมวลผล	22.067 ms.
				เวลาการถอดรหัส	1.30 ms.
				ถอดรหัส	YES



















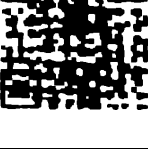

ตาราง ข. 7 ตัวอย่างผลการทดลองเพิ่มเติมที่ได้ในรอยปากกาขนาด 2.0 mm สีแดง

ภาพ QR Code ที่มีรอยขีดข่วนจากปากกาสีแดงขนาด 2.0 mm.					
ภาพที่นำเข้า	รอยขีดข่วน	กำจัด Noise	ผลลัพธ์	สรุป	
				NC	1
				เวลาประมวลผล	28.645 ms.
				เวลาการถอดรหัส	1.08 ms.
				ถอดรหัส	Yes
				NC	1
				เวลาประมวลผล	28.910 ms.
				เวลาการถอดรหัส	1.421 ms.
				ถอดรหัส	YES
				NC	1
				เวลาประมวลผล	20.902 ms.
				เวลาการถอดรหัส	1.309 ms.
				ถอดรหัส	YES
				NC	0.9986
				เวลาประมวลผล	20.579 ms.
				เวลาการถอดรหัส	1.553 ms.
				ถอดรหัส	YES
				NC	1
				เวลาประมวลผล	20.750 ms.
				เวลาการถอดรหัส	1.429 ms.
				ถอดรหัส	YES


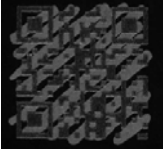








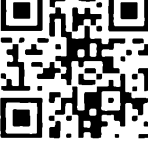









ตาราง ข. 8 ตัวอย่างผลการทดลองเพิ่มเติมที่ได้ในรอยปากกาขนาด 2.0 mm สีเขียว

ภาพ QR Code ที่มีรอยขีดข่วนจากปากกาสีเขียวขนาด 2.0 mm.					
ภาพที่นำเข้า	รอยขีดข่วน	กำจัด Noise	ผลลัพธ์	สรุป	
				NC	1
				เวลาประมวลผล	34.218 ms.
				เวลาการถอดรหัส	1.623 ms.
				ถอดรหัส	Yes
				NC	1
				เวลาประมวลผล	20.976 ms.
				เวลาการถอดรหัส	1.390 ms.
				ถอดรหัส	YES
				NC	1
				เวลาประมวลผล	20.808 ms.
				เวลาการถอดรหัส	1.364 ms.
				ถอดรหัส	YES
				NC	1
				เวลาประมวลผล	22.115 ms.
				เวลาการถอดรหัส	1.501 ms.
				ถอดรหัส	YES
				NC	1
				เวลาประมวลผล	23.250 ms.
				เวลาการถอดรหัส	1.224 ms.
				ถอดรหัส	YES


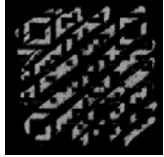












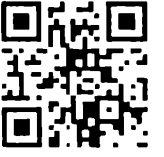


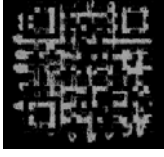


ตาราง ข. 9 ตัวอย่างผลการทดลองเพิ่มเติมที่ได้ในรอยปากกาขนาด 2.0 mm สีดำ

ภาพ QR Code ที่มีรอยขีดข่วนจากปากกาสีดำขนาด 2.0 mm.					
ภาพที่นำเข้า	รอยขีดข่วน	กำจัด Noise	ผลลัพธ์	สรุป	
				NC	0.9543
				เวลาประมวลผล	29.671 ms.
				เวลาการถอดรหัส	1.389 ms.
				ถอดรหัส	NO
				NC	0.9640 ms.
				เวลาประมวลผล	23.782 ms.
				เวลาการถอดรหัส	1.60 ms.
				ถอดรหัส	NO
				NC	0.9751
				เวลาประมวลผล	22.593 ms.
				เวลาการถอดรหัส	1.2240 ms.
				ถอดรหัส	NO
				NC	0.9668
				เวลาประมวลผล	21.027 ms.
				เวลาการถอดรหัส	1.593 ms.
				ถอดรหัส	NO
				NC	0.8629
				เวลาประมวลผล	21.512 ms.
				เวลาการถอดรหัส	0.909 ms.
				ถอดรหัส	NO







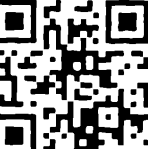










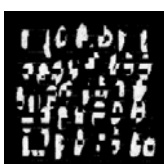


ตาราง ข.10 ตัวอย่างผลการทดลองเพิ่มเติมที่ได้ในรอยปากกาขนาด 3.5 mm สีน้ำเงิน

ภาพ QR Code ที่มีรอยขีดข่วนจากปากกาสีน้ำเงินขนาด 3.5 mm.					
ภาพที่นำเข้า	รอยขีดข่วน	กำจัด Noise	ผลลัพธ์	สรุป	
				NC	1
				เวลาประมวลผล	25.40 ms.
				เวลาการถอดรหัส	1.525 ms.
				ถอดรหัส	Yes
				NC	1
				เวลาประมวลผล	26.954 ms.
				เวลาการถอดรหัส	1.392 ms.
				ถอดรหัส	YES
				NC	1
				เวลาประมวลผล	27.97 ms.
				เวลาการถอดรหัส	1.41 ms.
				ถอดรหัส	YES
				NC	1
				เวลาประมวลผล	20.392 ms.
				เวลาการถอดรหัส	1.298 ms.
				ถอดรหัส	YES
				NC	1
				เวลาประมวลผล	19.924 ms.
				เวลาการถอดรหัส	1.411 ms.
				ถอดรหัส	YES



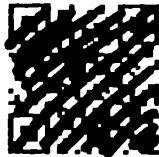



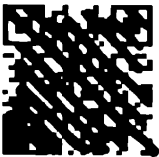




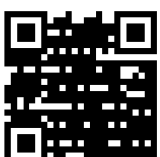






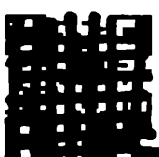
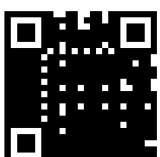
ตาราง ข. 11 ตัวอย่างผลการทดลองเพิ่มเติมที่ได้ในรอยปากกาขนาด 3.5 mm สีแดง

ภาพ QR Code ที่มีรอยขีดข่วนจากปากกาสีแดงขนาด 3.5 mm.				
ภาพที่นำเข้า	รอยขีดข่วน	กำจัด Noise	ผลลัพธ์	สรุป
				NC 1
				เวลาประมวลผล 27.171 ms.
				เวลาการถอดรหัส 1.525 ms.
				ถอดรหัส Yes
				NC 1
				เวลาประมวลผล 28.412 ms.
				เวลาการถอดรหัส 1.411 ms.
				ถอดรหัส YES
				NC 1
				เวลาประมวลผล 29.536 ms.
				เวลาการถอดรหัส 1.407 ms.
				ถอดรหัส YES
				NC 1
				เวลาประมวลผล 25.281 ms.
				เวลาการถอดรหัส 1.415 ms.
				ถอดรหัส YES
				NC 1
				เวลาประมวลผล 24.281 ms.
				เวลาการถอดรหัส 1.415 ms.
				ถอดรหัส YES







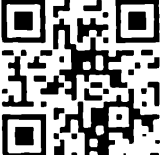



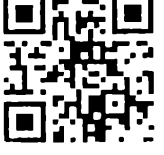









ตาราง ข. 12 ตัวอย่างผลการทดลองเพิ่มเติมที่ได้ในรอยปากกาขนาด 3.5 mm สีเขียว

ภาพ QR Code ที่มีรอยขีดข่วนจากปากกาสีเขียวขนาด 3.5 mm.					
ภาพที่นำเข้า	รอยขีดข่วน	กำจัด Noise	ผลลัพธ์	สรุป	
				NC	1
				เวลาประมวลผล	32.959 ms.
				เวลาการถอดรหัส	1.082 ms.
				ถอดรหัส	Yes
				NC	1
				เวลาประมวลผล	29.323 ms.
				เวลาการถอดรหัส	1.406 ms.
				ถอดรหัส	YES
				NC	1
				เวลาประมวลผล	24.498 ms.
				เวลาการถอดรหัส	1.317 ms.
				ถอดรหัส	YES
				NC	1
				เวลาประมวลผล	27.339 ms.
				เวลาการถอดรหัส	1.30 ms.
				ถอดรหัส	YES
				NC	1
				เวลาประมวลผล	20.222 ms.
				เวลาการถอดรหัส	1.342 ms.
				ถอดรหัส	YES





















ตาราง ข. 13 ตัวอย่างผลการทดลองเพิ่มเติมที่ได้ในรอยปากกาขนาด 3.5 mm สีดำ

ภาพ QR Code ที่มีรอยขีดข่วนจากปากกาสีดำขนาด 3.5 mm.					
ภาพที่นำเข้า	รอยขีดข่วน	กำจัด Noise	ผลลัพธ์	สรุป	
				NC	0.7799
				เวลาประมวลผล	28.589 ms.
				เวลาการถอดรหัส	1.447 ms.
				ถอดรหัส	NO
				NC	0.6996 ms.
				เวลาประมวลผล	23.697 ms.
				เวลาการถอดรหัส	1.246 ms.
				ถอดรหัส	NO
				NC	0.7854
				เวลาประมวลผล	24.555 ms.
				เวลาการถอดรหัส	1.297 ms.
				ถอดรหัส	NO
				NC	0.7716
				เวลาประมวลผล	24.942 ms.
				เวลาการถอดรหัส	1.179 ms.
				ถอดรหัส	NO
				NC	0.7051
				เวลาประมวลผล	21.493 ms.
				เวลาการถอดรหัส	1.029 ms.
				ถอดรหัส	NO





















ตาราง ข.14 ตัวอย่างผลการทดลองเพิ่มเติมที่ได้ในรอยปากกาขนาด 5 mm สีน้ำเงิน

ภาพ QR Code ที่มีรอยขีดข่วนจากปากกาสีน้ำเงินขนาด 5 mm.					
ภาพที่นำเข้า	รอยขีดข่วน	กำจัด Noise	ผลลัพธ์	สรุป	
				NC	1
				เวลาประมวลผล	30.011 ms.
				เวลาการถอดรหัส	1.582 ms.
				ถอดรหัส	Yes
				NC	0.9945
				เวลาประมวลผล	31.259 ms.
				เวลาการถอดรหัส	1.506 ms.
				ถอดรหัส	YES
				NC	1
				เวลาประมวลผล	26.433 ms.
				เวลาการถอดรหัส	1.434 ms.
				ถอดรหัส	YES
				NC	1
				เวลาประมวลผล	20.560 ms.
				เวลาการถอดรหัส	1.267 ms.
				ถอดรหัส	YES
				NC	1
				เวลาประมวลผล	20.888 ms.
				เวลาการถอดรหัส	1.428 ms.
				ถอดรหัส	YES











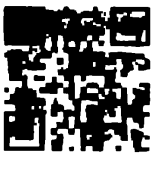



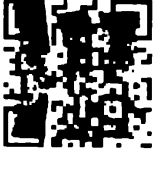




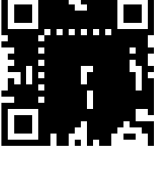
ตาราง ข. 15 ตัวอย่างผลการทดลองเพิ่มเติมที่ได้ในรอยปากกาขนาด 5 mm สีแดง

ภาพ QR Code ที่มีรอยขีดข่วนจากปากกาสีแดงขนาด 5 mm.					
ภาพที่นำเข้า	รอยขีดข่วน	กำจัด Noise	ผลลัพธ์	สรุป	
				NC	0.9945
				เวลาประมวลผล	52.517 ms.
				เวลาการถอดรหัส	2.212 ms.
				ถอดรหัส	Yes
				NC	1
				เวลาประมวลผล	21.591 ms.
				เวลาการถอดรหัส	1.028 ms.
				ถอดรหัส	YES
				NC	1
				เวลาประมวลผล	20.292 ms.
				เวลาการถอดรหัส	0.977 ms.
				ถอดรหัส	YES
				NC	1
				เวลาประมวลผล	21.843 ms.
				เวลาการถอดรหัส	1.383 ms.
				ถอดรหัส	YES
				NC	0.9972
				เวลาประมวลผล	20.384 ms.
				เวลาการถอดรหัส	1.868 ms.
				ถอดรหัส	YES

ตาราง ข. 16 ตัวอย่างผลการทดลองเพิ่มเติมที่ได้ในรอยปากกาขนาด 5 mm สีเขียว

ภาพ QR Code ที่มีรอยขีดข่วนจากปากกาสีเขียวขนาด 5 mm.					
ภาพที่นำเข้า	รอยขีดข่วน	กำจัด Noise	ผลลัพธ์	สรุป	
				NC	1
				เวลาประมวลผล	29.061 ms.
				เวลาการถอดรหัส	1.118 ms.
				ถอดรหัส	Yes
				NC	1
				เวลาประมวลผล	25.582 ms.
				เวลาการถอดรหัส	1.376 ms.
				ถอดรหัส	YES
				NC	1
				เวลาประมวลผล	31.229 ms.
				เวลาการถอดรหัส	0.976 ms.
				ถอดรหัส	YES
				NC	1
				เวลาประมวลผล	20.854 ms.
				เวลาการถอดรหัส	1.188 ms.
				ถอดรหัส	YES
				NC	1
				เวลาประมวลผล	21.47 ms.
				เวลาการถอดรหัส	1.499 ms.
				ถอดรหัส	YES

ตาราง ข. 17 ตัวอย่างผลการทดลองเพิ่มเติมที่ได้ในรอยปากกาขนาด 5 mm สีดำ

ภาพ QR Code ที่มีรอยขีดข่วนจากปากกาสีดำขนาด 5 mm.					
ภาพที่นำเข้า	รอยขีดข่วน	กำจัด Noise	ผลลัพธ์	สรุป	
				NC	0.8837
				เวลาประมวลผล	30.061 ms.
				เวลาการถอดรหัส	1.410 ms.
				ถอดรหัส	NO
				NC	0.9086 ms.
				เวลาประมวลผล	30.567 ms.
				เวลาการถอดรหัส	1.534 ms.
				ถอดรหัส	NO
				NC	0.8879
				เวลาประมวลผล	25.303 ms.
				เวลาการถอดรหัส	1.255 ms.
				ถอดรหัส	NO
				NC	0.910
				เวลาประมวลผล	21.159 ms.
				เวลาการถอดรหัส	1.291 ms.
				ถอดรหัส	NO
				NC	0.7009
				เวลาประมวลผล	25.388 ms.
				เวลาการถอดรหัส	1.111 ms.
				ถอดรหัส	NO

ประวัติผู้เขียนวิทยานิพนธ์

นายกมล หอมขจร เกิดเมื่อวันที่ 16 ธันวาคม พ.ศ. 2527 เกิดที่จังหวัดสุรินทร์ สำเร็จการศึกษาปริญญาวิทยาศาสตรบัณฑิต สาขาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏสุรินทร์ ในปีการศึกษา 2550 จากนั้นได้ทำงานในตำแหน่งเจ้าหน้าที่ฐานข้อมูลในมหาวิทยาลัยราชภัฏสุรินทร์เป็นระยะเวลา 2 ปี โดยมีหน้าที่หลักในการพัฒนาระบบงานฐานข้อมูล และดูแลระบบคอมพิวเตอร์เครือข่ายที่ใช้ในมหาลัย หลังจากนั้นได้มีโอกาสเปลี่ยนงาน โดยได้ร่วมงานกับบริษัทต่างประเทศซึ่งเป็นบริษัทเกี่ยวกับการพัฒนาเว็บไซต์ให้กับองค์กรต่างๆ ในประเทศเยอรมันนีโดยหน้าที่ที่รับผิดชอบคือนักพัฒนาเว็บไซต์ และในปี 2554 ได้เข้าศึกษาต่อในหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ ที่ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในภาคการศึกษาต้นปีการศึกษา 2554