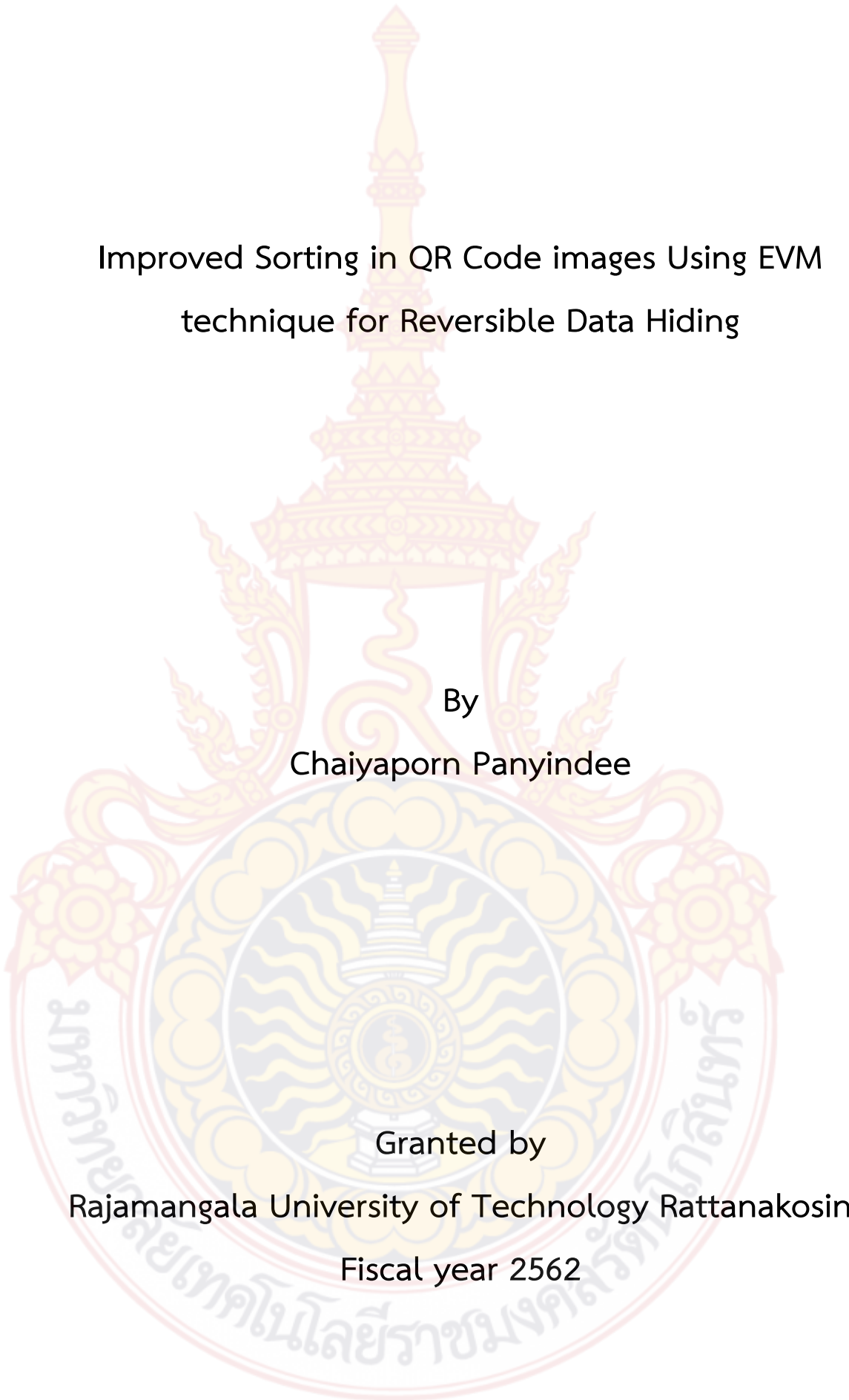




การปรับปรุงวิธีการเรียงลำดับข้อมูลในภาพคิวอาร์โค้ดโดยใช้  
เทคนิค EVM สำหรับการฝังข้อมูลแบบกู้คืนกลับได้

โดย  
ชัยพร ปานยินดี

สนับสนุนงบประมาณโดย  
มหาวิทยาลัยเทคโนโลยีราชมงคลรัตนโกสินทร์  
ประจำปีงบประมาณ 2562



Improved Sorting in QR Code images Using EVM  
technique for Reversible Data Hiding

By  
Chaiyaporn Panyindee

Granted by  
Rajamangala University of Technology Rattanakosin  
Fiscal year 2562

## กิตติกรรมประกาศ

โครงการวิจัยนี้สำเร็จสมบูรณ์ลงได้ด้วยการสนับสนุนงบประมาณโดยมหาวิทยาลัยเทคโนโลยีราชมงคลรัตนโกสินทร์ ประจำปี 2562 ที่ให้การสนับสนุนสถานที่ในการทำวิจัย ตลอดจนเครื่องมือต่าง ๆ ที่ใช้ในการจัดทำโครงการวิจัยนี้ ขอกราบขอบพระคุณ คุณพ่อ คุณแม่ และครอบครัวของผู้เขียน ที่ให้ความห่วงใย ให้การสนับสนุน ให้คำปรึกษา และให้กำลังใจแก่ผู้เขียนตลอดมาสุดท้ายนี้ ขอขอบพระคุณ บุคคลทุกคนที่มีได้เอ่ยนามมา ณ ที่นี้ ที่ได้ให้การสนับสนุน และเป็นกำลังใจแก่ผู้เขียนตลอดมา คุณค่า และประโยชน์อันพึงมาจากโครงการวิจัยนี้ ข้าพเจ้าขอมอบแด่ผู้มีพระคุณทุกท่าน

ชัยพร ปานยินดี

2562



## บทคัดย่อ

รหัสโครงการ : C๔๒/๒๕๖๒

ชื่อโครงการ : การปรับปรุงวิธีการเรียงลำดับข้อมูลในภาพคิวอาร์โค้ดโดยใช้เทคนิค EVM สำหรับการฝังข้อมูลแบบกู้คืนกลับได้

ชื่อนักวิจัย : ชัยพร ปานยินดี (ดร.)

งานวิจัยฉบับนี้นำเสนอหนึ่งการปรับปรุงวิธีการเรียงลำดับข้อมูลในภาพคิวอาร์โค้ดโดยประยุกต์ใช้เทคนิค EVM เพื่อเรียงลำดับค่าความคลาดเคลื่อนจากการทำนายที่สอดคล้องก่อนการฝังลำดับของค่าความคลาดเคลื่อนจากการทำนายถูกคำนวณใหม่ และใช้แทนที่ค่าความแปรปรวนดั้งเดิมซึ่งคำนวณได้จากสี่พิกเซลข้างเคียง เพื่อให้ได้รับค่า PSNR สูงสุดที่เป็นไปได้ ค่าความแปรปรวนใหม่สามารถขยายระดับของพื้นที่พิกเซลได้สูงสุด 30 ระดับ กับ 40 ภาพคิวอาร์โค้ดที่ใช้ทดสอบผลลัพธ์ แสดงให้เห็นการเรียงลำดับที่มีประสิทธิภาพกว่างานในอดีตเร็ว ๆ นี้ โดยเฉพาะการฝังข้อมูลขนาดเล็ก

คำสำคัญ : ค่าความคลาดเคลื่อนจากการทำนาย การฝังข้อมูลแบบกู้คืนกลับได้ คิวอาร์โค้ด EVM

E-mail Address : (chaiyaporn.pan@rmutr.ac.th)

ระยะเวลาโครงการ : ตุลาคม พ.ศ. 2561 – สิงหาคม พ.ศ. 2562



## Abstract

**Code of project** : C42/2562

**Project name** : Improved Sorting in QR Code images Using EVM technique for Reversible Data Hiding

**Researcher name** : Chaiyaporn Panyindee (Dr.)

This research presents a novel strategy for a reversible data-hiding (RDH) algorithm used in QR Code imaging. The linear weighting is used with many effective RDH techniques to embed large amounts of data into QR Code images with low distortion. Many linear weighting values are exploited significantly for the possible capacity. The Experiments with several QR Code versions were carried out and the results were compared to previous work in term of higher embedding capacity, our approach presented in this paper provided better results with lower distortion.

**Keywords:** Linear weighting, Reversible data hiding, QR Code.

---

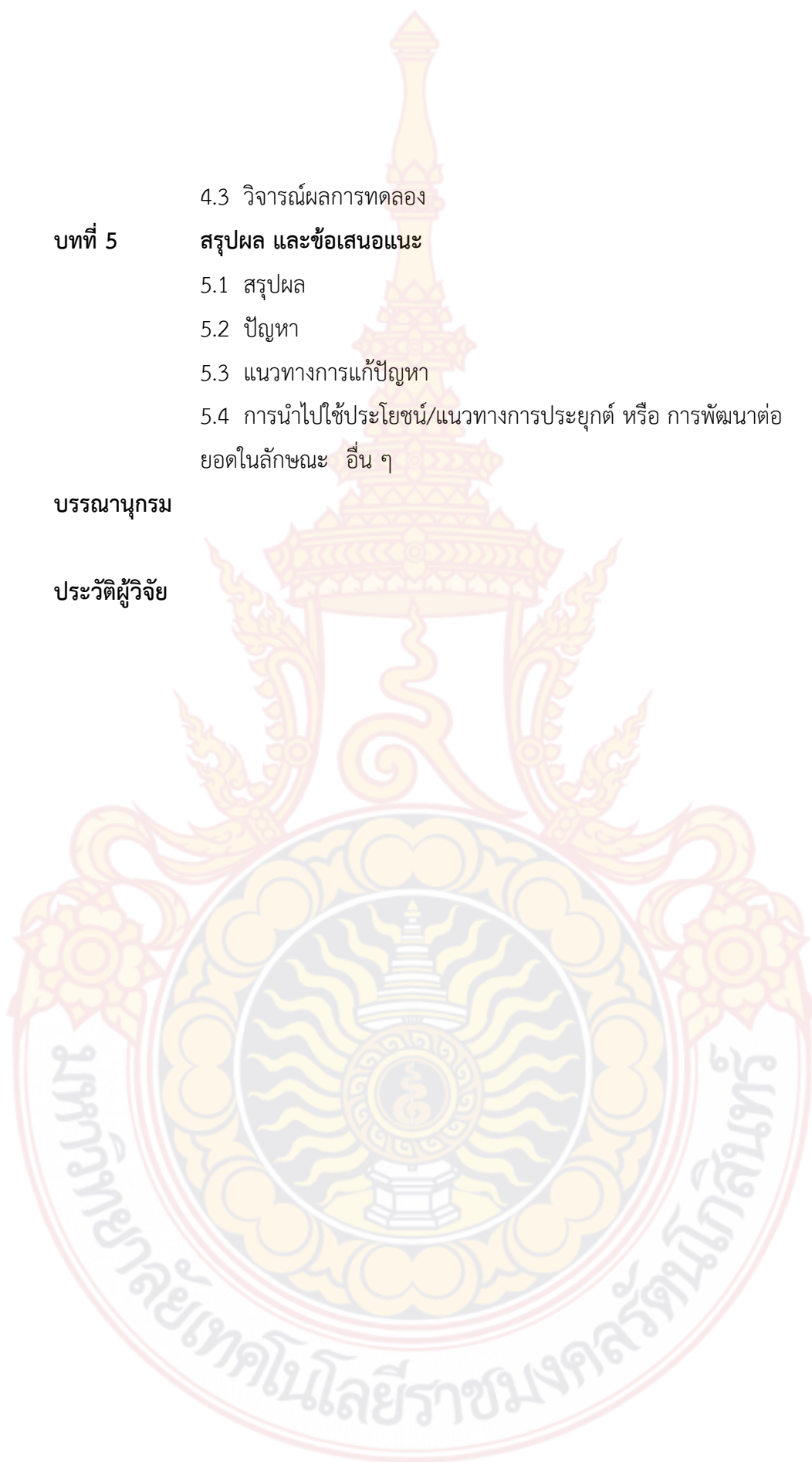
**E-mail Address** : (chaiyaporn.pan@rmutr.ac.th)

**Period of project** : October 2018 - August 2019

## สารบัญ

	หน้า	
กิตติกรรมประกาศ	ค	
บทคัดย่อภาษาไทย	ง	
บทคัดย่อภาษาอังกฤษ	จ	
สารบัญ	ฉ	
สารบัญตาราง	ช	
สารบัญภาพ	ฌ	
<b>บทที่ 1</b>	<b>บทนำ</b>	<b>1</b>
	1.1 ที่มา และความสำคัญ	1
	1.2 วัตถุประสงค์	2
	1.3 ขอบเขต	2
	1.4 ประโยชน์ที่คาดว่าจะได้รับ	2
	1.5 แผนการดำเนินงาน	3
<b>บทที่ 2</b>	<b>ทบทวนวรรณกรรมที่เกี่ยวข้อง</b>	<b>4</b>
	2.1 ความรู้พื้นฐาน	4
	2.2 การฝังข้อมูลแบบกู้คืนกลับได้ (RDH)	4
	2.3 การทำนายค่าพิกเซล (Pixel Prediction)	5
	2.4 การฝัง และการกู้คืนข้อมูล (Embedding and Recovering)	6
	2.5 การเลื่อนฮิสโตแกรม (Histogram Shifting)	8
	2.6 การเรียงลำดับข้อมูล (Data Sorting)	11
	2.7 การทดสอบแบบดับเบิล (Double Modification Testing: DMT)	13
	2.8 การฝังข้อมูลข้อมูลแบบกู้คืนกลับได้สำหรับภาพควอดริโค้ด	13
<b>บทที่ 3</b>	<b>วิธีการที่นำเสนอ</b>	<b>15</b>
	3.1 ขั้นตอนการฝังข้อมูล	18
	3.2 ขั้นตอนการกู้คืนข้อมูล และภาพต้นฉบับ	18
<b>บทที่ 4</b>	<b>ผลการวิจัย และผลการวิเคราะห์</b>	<b>20</b>
	4.1 การประเมินผล	20
	4.2 ผลการทดลอง	21

	4.3	วิจารณ์ผลการทดลอง	21
<b>บทที่ 5</b>		<b>สรุปผล และข้อเสนอแนะ</b>	<b>28</b>
	5.1	สรุปผล	28
	5.2	ปัญหา	28
	5.3	แนวทางการแก้ปัญหา	28
	5.4	การนำไปใช้ประโยชน์/แนวทางการประยุกต์ หรือ การพัฒนาต่อ ยอดในลักษณะ อื่น ๆ	28
<b>บรรณานุกรม</b>			<b>29</b>
<b>ประวัติผู้วิจัย</b>			<b>31</b>



## สารบัญตาราง

ตารางที่	หน้า
1-1 แผนการดำเนินการ	3
2-1 แสดงตัวอย่างการใช้สมการ $d + b$ และ $2d + b$	7
4-1 ผลลัพธ์การฝังข้อมูล ex 1-30 ของภาพ QR Code เวอร์ชัน 40	22
4-2 ผลลัพธ์ค่า PSNR และค่าถ่วงน้ำหนักที่ดีที่สุดสำหรับภาพคิวอาร์โค้ด 40 เวอร์ชัน	26





## สารบัญภาพ

ภาพที่	หน้า
1-1 ภาพคิวอาร์โค้ด	2
2-1 ความรู้พื้นฐานของงานวิจัยนี้	4
2-2 การแบ่งเซตครอส และเซตดอต	6
2-3 แสดงตัวอย่างการเลื่อนฮิสโตแกรม	11
2-4 (ก) ตัวอย่างภาพคิวอาร์โค้ดต้นฉบับขนาด 3x3 พิกเซล (ข) ฝังข้อมูลลงในภาพ (ก) โดยใช้ HS [7]	14
3-1 (ก) การเปรียบเทียบค่า $d$ และ $\mu$ โดยไม่ใช้เทคนิคการเรียงลำดับข้อมูล (Data Sorting)	16
(ข) การเปรียบเทียบค่า $d$ และ $\mu_{LV}$ โดยใช้เทคนิคการเรียงลำดับข้อมูล (Data Sorting)	
(ค) การเปรียบเทียบค่า $d$ และ $\mu_{EVM}$ โดยใช้เทคนิคการเรียงลำดับข้อมูล (Data Sorting)	
[11]	
3-2 ไดอะแกรมกระบวนการฝังข้อมูล และกู้คืนข้อมูล [11]	17
4-1 กราฟแสดงค่า PSNR VS Payload ของวิธีการที่นำเสนอ (EVM+Weight) โดยเปรียบเทียบกับงานในอดีต (LV+Weight) สำหรับ 6 ภาพคิวอาร์โค้ด	25

## บทที่ 1

### บทนำ

#### 1.1 ที่มา และความสำคัญ

การฝังข้อมูลแบบกู้คืนกลับได้ RDH สำหรับภาพเป็นการฝังข้อมูลโดยการโมดิไฟ์พิกเซล เพื่อจัดเก็บข้อมูล โดยที่ภาพต้นฉบับ และข้อมูลที่ถูกฝังก่อนหน้าต้องสามารถกู้คืนกลับได้ทั้งหมด แบบไม่มีการสูญเสีย RDH ถูกนำมาประยุกต์ใช้งานอย่างหลากหลาย เช่น ฝังข้อมูลลับลงในภาพทางการแพทย์ ภาพทางทหาร เป็นต้น ในปัจจุบันภาพคิวอาร์โค้ดได้ถูกนำมาใช้ในการฝังข้อมูลแบบกู้คืนกลับได้ เนื่องจากข้อมูลที่ถูกส่งไปกับภาพคิวอาร์โค้ดนั้นสามารถเข้าถึงได้โดยง่าย ซึ่งข้อมูลบางประเภทยังคงต้องการความเป็นส่วนตัว เช่น เลขที่ธุรกรรม วันเดือนปีเกิด เลขประจำตัวประชาชน จึงต้องอาศัยเทคนิคการฝังข้อมูลแบบกู้คืนกลับได้เพื่อป้องกันการเข้าถึงข้อมูล

RDH เป็นหนึ่งแขนงงานที่ถูกพัฒนาอย่างต่อเนื่อง มีนักวิจัยจำนวนมากนำเสนอวิธีการใหม่ ๆ สำหรับเพิ่มประสิทธิภาพของการฝังข้อมูลที่ยังคงสามารถกู้คืนข้อมูลกลับได้ หนึ่งในวิธีการที่ถูกกล่าวขานคือ Difference Expansion (DE) ซึ่งถูกนำเสนอโดย Tian เทคนิคของพวกเขาถูกปรับปรุงในหลายมิติ เช่น Thodi และ Rodriguez นำเสนอกระบวนการที่มีชื่อว่า Prediction Error Expansion (PEE) โดยได้ปรับปรุงกระบวนการ DE เดิมจากการใช้ค่าความแตกต่างเดิมมาใช้ตัวทำนายแทน กระบวนการของพวกเขาช่วยลดความบิดเบือนอย่างมีนัยสำคัญ สำหรับตัวทำนายรูปแบบใหม่ จึงถูกต่อยอดในอีกหลายงานวิจัย Thodi และ Rodriguez ได้นำเสนออีกหนึ่งการปรับปรุงที่สำคัญ และมีผลต่องานยุคสมัยใหม่คือ Histogram Shift (HS) HS ถูกนำมาใช้แก้ปัญหาพิกเซลที่ซ้อนทับกันของเทคนิค PEE เพื่อลดขนาดของพื้นที่แมป พื้นที่แมปเป็นอีกหนึ่งปัญหาใหญ่ของ RDH เนื่องจากพื้นที่แมปจะต้องถูกฝังไปกับข้อมูล ดังนั้นหากลดขนาดลงได้ พื้นที่ในการฝังจะเพิ่มขึ้น ตามหลายเทคนิค [1] – [4] สามารถลดขนาดของพื้นที่แมปอย่างมีนัยสำคัญ หนึ่งในกลยุทธ์ที่มีประสิทธิภาพคือการใช้เทคนิคการเรียงลำดับข้อมูล (Data Sorting: DS) DS ถูกนำเสนอครั้งแรกโดย Kamstra และ Heijmans พวกเขาเรียงลำดับข้อมูลสำหรับการฝัง โดยใช้ค่าสหสัมพันธ์จากพิกเซลข้างเคียงสี่ตำแหน่ง แนวคิดของการเรียงลำดับข้อมูลก่อนการฝังถูกพัฒนาอย่างต่อเนื่อง [5]-[6]

เมื่อเร็ว ๆ นี้ Pandeyindee และคณะ [7] ได้นำเสนอการใช้หลายเทคนิคสำหรับหนึ่งขั้นตอนวิธี RDH ร่วมกับการปรับภาพเชิงเส้นสำหรับภาพคิวอาร์โค้ดในหลายเวอร์ชัน วิธีการของพวกเขาบรรลุการฝังขนาดใหญ่ในขณะที่ความบิดเบือนต่ำเมื่อเทียบกับงานก่อนหน้า [8] หนึ่งในปัญหาสำคัญสำหรับการจัดเรียงลำดับข้อมูลโดยใช้เพียงสี่พิกเซลข้างเคียงในภาพคิวอาร์โค้ด ดูจากรูปที่ 1 จะพบว่าภาพคิวอาร์โค้ดมีลักษณะของพิกเซลคล้ายกับสัญญาณรบกวน กล่าวคือ ค่าเฉลี่ยของพิกเซลมีการ

เปลี่ยนแปลงอย่างรวดเร็ว เช่น จาก 255 เป็น 0 ซึ่งการใช้ค่าความแปรปรวนจาก 4 ตำแหน่งข้างเคียง ไม่เพียงพอเพื่อให้ได้การเรียงลำดับข้อมูลที่ต้องการ

ในงานวิจัยฉบับนี้จึงนำเสนอหนึ่งการปรับปรุงเทคนิคการเรียงลำดับข้อมูลโดยประยุกต์ใช้เทคนิค EVM ร่วมกับหลายเทคนิค การคำนวณหาค่า EVM ใหม่ ในแต่ละการฝังถูกเพิ่มขอบเขตจนได้รับค่าที่เหมาะสมที่สุดเพื่อให้ได้ค่า PSNR สูงสุดที่เป็นไปได้ การเรียงลำดับข้อมูลที่มีประสิทธิภาพ ค่าความคลาดเคลื่อนจากการทำนายที่มีค่าต่ำควรถูกเรียงขึ้นก่อนค่าที่มีค่าสูงตามค่า EVM เมื่อฝังข้อมูลโดยใช้ HS ความบิดเบือนที่ได้รับจะต่ำตามค่าความคลาดเคลื่อนจากการทำนายที่มีขนาดเล็ก ในทางเดียวกัน เมื่อค่าความคลาดเคลื่อนที่มีขนาดเล็กยังคงช่วยลดปัญหา Overflow หรือในอีกความหมายหนึ่ง ขนาดของพื้นที่ที่สามารถฝังได้สูงขึ้น ในขณะที่พื้นที่แมปมีขนาดเล็กลง



ภาพที่ 1-1 ภาพคิวอาร์โค้ด

### 1.2 วัตถุประสงค์

1. เพื่อแก้ปัญหการเรียงลำดับของขั้นตอนวิธีการฝังข้อมูลแบบกู่คืนกลับได้สำหรับภาพคิวอาร์โค้ด
2. เพื่อให้ฝังข้อมูลได้มากขึ้น และความบิดเบือนลดลง

### 1.3 ขอบเขต

1. ศึกษา และวิเคราะห์ขั้นตอนวิธีการฝังข้อมูลแบบกู่คืนกลับได้สำหรับคิวอาร์โค้ด
2. ใช้ภาพคิวอาร์โค้ดนามสกุล PNG ชนิดสี RGB และภาพคิวอาร์โค้ด 40 เเวอร์ชัน
3. สามารถฝังข้อมูลได้ และกู่คืนกลับได้

### 1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้ขั้นตอนวิธีการใหม่ของการฝังข้อมูลแบบกู่คืนกลับได้สำหรับภาพคิวอาร์โค้ด

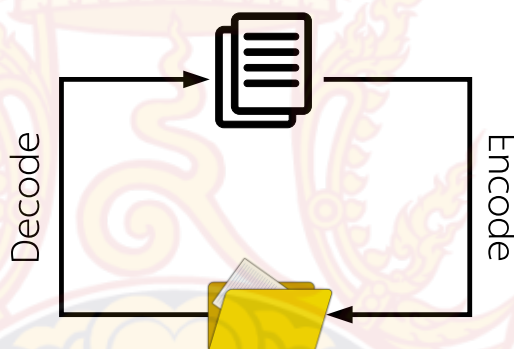


## บทที่ 2

### ทบทวนวรรณกรรมที่เกี่ยวข้อง

ในบทที่ 2 นี้จะเป็นการรวบรวมทฤษฎีการฝังข้อมูลแบบสามารถกู้คืนกลับได้ ซึ่งเป็นส่วนหนึ่งของศาสตร์วิทยาการอำพรางข้อมูลในหลายด้านร่วมกัน และกล่าวถึงงานวิจัยต่าง ๆ ที่เกี่ยวกับ RDH สำหรับภาพควอาร์โค้ด รวมถึงหลักการพื้นฐานที่เกี่ยวข้องกับงานวิจัย

#### 2.1 ความรู้พื้นฐาน



ภาพที่ 2-1 ความรู้พื้นฐานของงานวิจัยนี้

ในส่วนนี้จะอธิบายถึงการฝังข้อมูลแบบกู้คืนกลับได้สำหรับภาพดิจิทัลพื้นฐาน ซึ่งแบ่งออกเป็น 2 ส่วน คือ การเข้ารหัส (Encode) และการถอดรหัส (Decode) ดังภาพที่ 2-1 การเข้ารหัสคือกระบวนการที่ผู้ส่งนำภาพดิจิทัลทำการอำพรางข้อมูลลงในภาพ ซึ่งจะถูกส่งไปยังปลายทางผู้รับจะได้ภาพใหม่ที่มีข้อมูลลับซ่อนอยู่ สำหรับการถอดรหัสเป็นการนำภาพดิจิทัลที่ผู้ส่งอำพรางข้อมูลเข้าสู่กระบวนการย้อนกลับ หรือการกู้คืนข้อมูลเพื่อสามารถนำข้อมูลที่ผู้ส่งอำพรางข้อมูลลงไป และภาพต้นฉบับกลับคืนมา สำหรับงานวิจัยนี้จะกล่าวถึงการอำพรางข้อมูลลงในภาพควอาร์โค้ด 40 เวอร์ชันแบบไม่สูญเสียซึ่งจะกล่าวถึงต่อไป

#### 2.2 การฝังข้อมูลแบบกู้คืนกลับได้ (RDH)

การฝังข้อมูลแบบกู้คืนกลับได้สำหรับภาพดิจิทัลเป็นการฝังข้อมูลโดยมีการเปลี่ยนแปลงพิกเซลเพื่อจัดเก็บข้อมูลที่ต้องการฝัง และสามารถแยกข้อมูลที่ถูกฝังก่อนหน้าออกมาได้ทั้งหมดโดยไม่ทำให้ภาพต้นฉบับเกิดความบิดเบือน โดยใช้การแก้ไขบิตที่มีนัยสำคัญต่ำสุดของพิกเซลให้เป็นค่าของ

ข้อมูลที่ต้องการที่แปลงเป็นเลขฐานสองแล้ว โดยปกติการแก้ไขพิกเซลของภาพจะเก็บข้อมูลได้ 1 บิต หากภาพมีขนาดใหญ่ก็จะสามารถฝังข้อมูลได้มากขึ้นด้วย [6]

### 2.3 การทำนายค่าพิกเซล (Pixel Prediction)

Sachnev และคณะ [6] ได้ใช้ตัวทำนาย Rhombus ในงานวิจัยของเขาเพื่อทำนายค่าพิกเซลที่จุด  $u_{i,j}$  โดยต้องใช้พิกเซลข้างเคียง 4 พิกเซล คือ  $v_{i-1,j}, v_{i+1,j}, v_{i,j+1}$  และ  $v_{i,j-1}$  ทั้ง 5 พิกเซลรวม  $u_{i,j}$  จะเป็นเซตที่ใช้ฝังข้อมูลสำหรับ 1 บิตต่อ 1 พิกเซล โดยพิกเซลทั้งหมดของภาพนั้นจะแบ่งออกเป็น 2 เซต คือ เซตครอส (Cross set) และเซตดอต (Dot set) ซึ่งการฝังเซตครอสแบ่งออกเป็น 2 ขั้นตอน คือการฝังเซตครอสในแถวคี่ และการฝังเซตครอสในแถวคู่ ดังภาพที่ 2-2 ในทางเดียวกัน สำหรับการฝังเซตดอตแบ่งออกเป็น 2 ขั้นตอนเช่นเดียวกัน จะเห็นได้ว่าจากภาพที่ 2-2 เริ่มต้นที่การฝังเซตครอส โดยทำการฝังเซตครอสในแถวคู่ และแถวคี่ เซตดอตจึงเป็นเซตที่ถูกทำนายค่าพิกเซล และในทางกลับกันเมื่อเราฝังในเซตดอต เซตครอสที่ถูกฝังลงไปแล้วจะเป็นเซตที่ถูกทำนายค่าพิกเซล ซึ่งจะเห็นได้ว่าการฝังข้อมูลลงในเซตครอสในแถวคู่ และแถวคี่มีความสัมพันธ์กัน สังเกตว่าเซตดอตที่ถูกทำนายในการฝังเซตครอสในแถวคู่ ครึ่งหนึ่งของการทำนายในแถวคู่นำมาใช้ในการทำนายในแถวคี่ด้วยเช่นกัน แต่ไม่มีการส่งผลกระทบต่อกรฝังข้อมูล และกู้คืนข้อมูล โดยจะได้ค่า  $u'_{i,j}$  เป็นค่าที่ถูกทำนายแล้ว ซึ่งสามารถคำนวณได้ดังสมการนี้

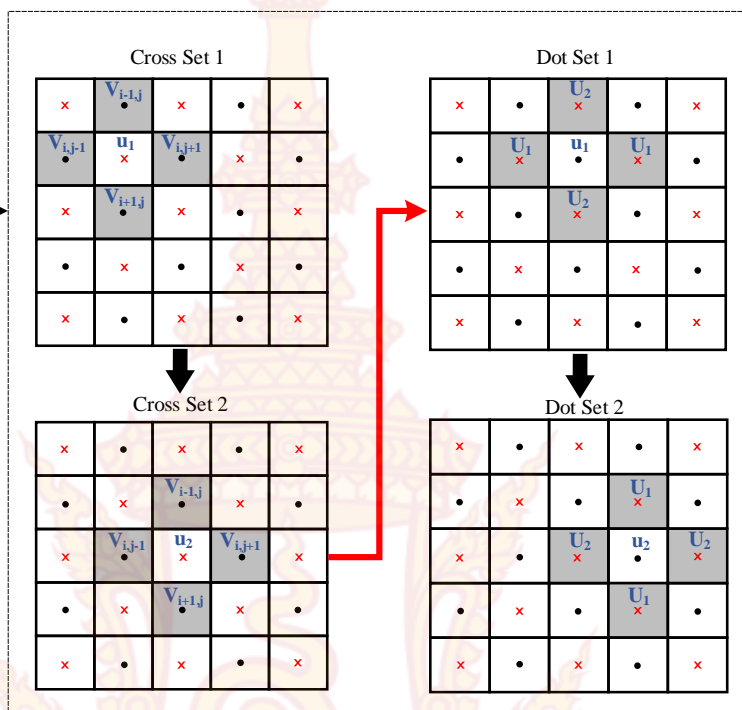
$$u'_{i,j} = \left\lfloor \frac{v_{i,j-1} + v_{i+1,j} + v_{i,j+1} + v_{i-1,j}}{4} \right\rfloor \quad (2-1)$$

สำหรับการกู้คืนข้อมูลทำย้อนกลับจากกระบวนการฝังข้อมูล คือเมื่อฝังเซตดอตเป็นเซตสุดท้าย จึงเริ่มการกู้คืนด้วยเซตดอตก่อน และใช้เซตครอสเป็นเซตที่ถูกทำนายค่าพิกเซล เมื่อทำการกู้คืนเซตดอตทั้ง 2 ขั้นตอนเสร็จแล้ว จึงทำการกู้คืนข้อมูลของเซตครอส เซตดอตจึงเป็นเซตที่ถูกทำนายค่าพิกเซล และจึงได้ภาพต้นฉบับกลับคืนมา

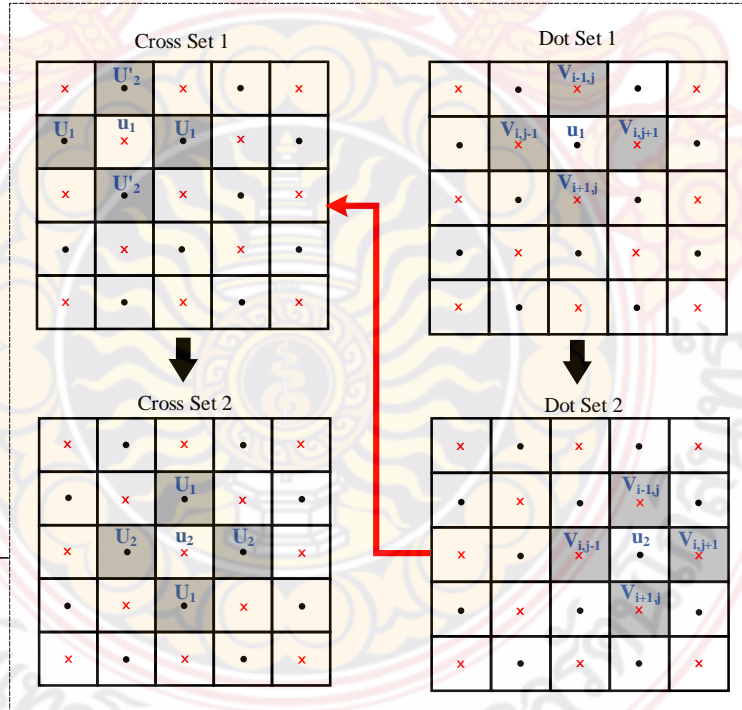
จากนั้นคำนวณค่าความคลาดเคลื่อนจากการทำนาย (Prediction Error)  $d_{i,j}$  ได้จากการนำค่าพิกเซลต้นฉบับ  $u_{i,j}$  มาลบกับค่าพิกเซลที่ถูกทำนาย  $u'_{i,j}$  ซึ่งสามารถคำนวณได้ดังสมการนี้

$$d_{i,j} = u_{i,j} - u'_{i,j} \quad (2-2)$$

Encode



Decode



ภาพที่ 2-2 การแบ่งเซตครอส และเซตดอต

## 2.4 การฝัง และการกู้คืนข้อมูล (Embedding and Recovering)

การขยายค่าความคลาดเคลื่อนจากการทำนาย (Prediction Error Expansion) จำเป็นต่อการฝังข้อมูลลงในพิกเซล [5] ซึ่งสามารถคำนวณได้ดังสมการนี้

$$D_{i,j} = 2d_{i,j} + b \quad (2-3)$$

เมื่อ  $D_{i,j}$  คือ ค่าความคลาดเคลื่อนจากการทำนายที่ถูกขยายแล้ว

$b$  คือ ข้อมูลที่นำมาฝัง (1 บิต)

จากตารางที่ 2-1 สังเกตว่าผลลัพธ์ของสมการ  $d + b$  มีค่าที่ซ้อนทับกัน เช่น ที่  $d = 0, b = 1$  ผลลัพธ์ที่ได้คือ 1 และที่  $d = 1, b = 0$  ผลลัพธ์ที่ได้คือ 1 เช่นเดียวกัน ผลลัพธ์ที่เท่ากันทำให้ไม่สามารถระบุได้ว่าค่าบิตที่ฝังนั้นเป็นเท่าไร ส่งผลให้ไม่สามารถกู้คืนข้อมูลกลับได้ในทางกลับกัน สมการ  $2d + b$  ไม่ซ้อนทับกันทำให้สามารถระบุได้ว่าค่าบิตที่ฝังนั้นเป็นเท่าไร จึงทำให้สามารถกู้คืนข้อมูลกลับได้ Tian [5] จึงได้นำเสนอการขยายค่าความผิดพลาดจากการทำนายขึ้นมาเพื่อหลีกเลี่ยงปัญหาดังกล่าว

ตารางที่ 2-1 แสดงตัวอย่างการใช้สมการ  $d + b$  และ  $2d + b$

$d$	$b$	$d + b$	$2d + b$
-3	0	-3	-6
	1	-2	-5
-2	0	-2	-4
	1	-1	-3
-1	0	-1	-2
	1	0	-1
0	0	0	0
	1	1	1
1	0	1	2
	1	2	3
2	0	2	4
	1	3	5



	1	3	5
3	0	3	6
	1	4	7

จากนั้นนำค่า  $D_{i,j}$  แทนค่าพิกเซลเดิม  $u_{i,j}$  และนำมาบวกกับค่าพิกเซลที่ถูกทำนาย  $u'_{i,j}$  จะได้ค่า  $U_{i,j}$  ซึ่งสามารถคำนวณได้ดังสมการนี้

$$U_{i,j} = D_{i,j} + u'_{i,j} \quad (2-4)$$

สำหรับการกู้คืนข้อมูล และค่าพิกเซลเดิมกลับคืนมา จะทำตรงกันข้ามกับสมการการฝังข้อมูลดังนี้ ค่าความคลาดเคลื่อนจากการทำนายที่ถูกขยายสามารถคำนวณได้ดังสมการนี้

$$D_{i,j} = U_{i,j} - u'_{i,j} \quad (2-5)$$

ข้อมูลที่ถูกฝังลงไปจะคำนวณได้ดังสมการนี้

$$b = D_{i,j} \bmod 2 \quad (2-6)$$

ค่าความคลาดเคลื่อนจากการทำนายสามารถคำนวณได้ดังสมการนี้

$$d_{i,j} = \frac{D_{i,j}}{2} \quad (2-7)$$

ค่าพิกเซลต้นฉบับสามารถคำนวณได้ดังสมการนี้

$$u_{i,j} = u'_{i,j} + d_{i,j} \quad (2-8)$$

## 2.5 การเลื่อนฮิสโตแกรม (Histogram Shifting)

Thodi และ Rodriguez [9] ได้นำเสนอเทคนิคการเลื่อนฮิสโตแกรม เพื่อเพิ่มประสิทธิภาพสำหรับการฝังข้อมูลแบบกู้คืนกลับได้ ส่งผลให้ค่าความบิดเบือนของภาพลดลง โดยมีการกำหนดช่วงค่าขีดแบ่ง (Threshold) คือ  $T_n$  หมายถึง ช่วงค่าขีดแบ่งฝั่งลบ และ  $T_p$  หมายถึง ช่วงค่าขีดแบ่งฝั่งบวก เพื่อเป็นการกำหนดขอบเขตของพิกเซลสำหรับการฝังข้อมูล โดยมีขั้นตอนสำหรับการเลื่อนฮิสโตแกรมในการฝังข้อมูลลงในค่าความคลาดเคลื่อนจากการทำนายสามารถคำนวณได้ดังสมการนี้

$$D_{i,j} = \begin{cases} 2d_{i,j} + b, & \text{if } d_{i,j} \in [T_n; T_p] \\ d_{i,j} + T_p + 1, & \text{if } d_{i,j} > T_p \text{ and } T_p \geq 0 \\ d_{i,j} + T_n, & \text{if } d_{i,j} < T_n \text{ and } T_n < 0 \end{cases} \quad (2-9)$$

สำหรับขั้นตอนวิธีการกู้คืนค่าความคลาดเคลื่อนจากการทำนายสามารถคำนวณได้ดังสมการนี้

$$d_{i,j} = \begin{cases} [D_{i,j} / 2], & \text{if } D_{i,j} \in [2T_n; 2T_p + 1] \\ D_{i,j} - T_p - 1, & \text{if } D_{i,j} > 2T_p + 1 \text{ and } T_p \geq 0 \\ D_{i,j} - T_n, & \text{if } D_{i,j} < 2T_n \text{ and } T_n < 0 \end{cases} \quad (2-10)$$

และการเลื่อนฮิสโตแกรมสำหรับขั้นตอนการกู้คืนข้อมูลที่ฝังลงไปสามารถคำนวณได้ดังสมการนี้

$$b = D_{i,j} \bmod 2, \quad D_{i,j} \in [2T_n; 2T_p + 1] \quad (2-11)$$

จากภาพที่ 2-3 เป็นการแสดงตัวอย่างการเลื่อนฮิสโตแกรม ภาพทางด้านบน คือภาพก่อนการฝังข้อมูล ภาพด้านล่างคือภาพหลังการฝังข้อมูล โดยสีเขียว คือพิกเซลที่สามารถฝังได้ สีส้ม คือพิกเซลที่สามารถเลื่อนค่าได้ และสีแดง คือไม่สามารถเลื่อนค่าได้ เมื่อเราฝังข้อมูลลงในสีเขียว ค่าดังกล่าวจึงเปลี่ยนเป็น 1 ส่งผลให้เกิดการซ้อนทับกันไม่สามารถแยกออกได้ว่าค่าไหนเป็นค่าที่ถูกฝังข้อมูลลงไป หรือค่าไหนเป็นค่าที่ไม่ได้ถูกฝังข้อมูลลงไป เนื่องจากมีค่าที่เป็น 1 เหมือนกันหมด เพื่อแก้ปัญหาเรื่องนี้จึงต้องมีการขยายออกไปเพื่อไม่ให้ซ้ำค่าเดิม ทำให้ 1 ในสีส้มถูกเลื่อนออกไปเป็น 2 จึงทำให้สามารถแยกได้ว่าพิกเซลไหนเป็นพิกเซลที่ถูกฝังข้อมูลลงไป เพื่อแก้ปัญหาเหล่านี้ Thodi และ Rodriguez [9] จึงได้นำเสนอเทคนิคการเลื่อนฮิสโตแกรมขึ้นมา

### 2.5.1 การแก้ปัญหา Overflow/Underflow

การเกิด Overflow และ Underflow [9] ซึ่งเกิดจากการเลื่อนฮิสโตแกรมของค่าความคลาดเคลื่อนจากการทำนาย โดยมีเงื่อนไขสำหรับการเลื่อนฮิสโตแกรมไม่ให้เกิดปัญหา Overflow และ Underflow ได้ดังสมการนี้

$$0 \leq u'_{i,j} + D_{i,j} \leq 255 \quad (2-12)$$

จากเงื่อนไขดังกล่าวจะถูกใช้ก็ต่อเมื่อการหาเซลล์ที่เกิดปัญหา Overflow และ Underflow ซึ่งเมื่อพิกเซลใด ๆ ที่เข้าเงื่อนไขจะถูกบันทึกลงในพื้นที่แมปเพื่อบันทึกพิกเซลที่

255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	1	1	1	1	1	1	1	1	1	1	1	1	1	1	255
255	1	1	1	1	1	1	1	1	1	1	1	1	1	1	255
255	1	1	1	1	1	1	1	1	1	1	1	1	1	1	255
255	1	1	1	0	0	0	0	0	0	0	0	0	1	1	255
255	1	1	1	0	0	0	0	0	0	0	0	0	1	1	255
255	1	1	1	0	0	0	0	0	0	0	0	0	1	1	255
255	1	1	1	0	0	0	0	0	0	0	0	0	1	1	255
255	1	1	1	0	0	0	0	0	0	0	0	0	1	1	255
255	1	1	1	0	0	0	0	0	0	0	0	0	1	1	255
255	1	1	1	0	0	0	0	0	0	0	0	0	1	1	255
255	1	1	1	0	0	0	0	0	0	0	0	0	1	1	255
255	1	1	1	1	1	1	1	1	1	1	1	1	1	1	255
255	1	1	1	1	1	1	1	1	1	1	1	1	1	1	255
255	1	1	1	1	1	1	1	1	1	1	1	1	1	1	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255

Underflow / Overflow

$d + T_p + 1, d + T_n$

$2d + b$

(ก) แสดงตัวอย่างการเลื่อนฮิสโตแกรมก่อนการฝังข้อมูล

255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	2	2	2	2	2	2	2	2	2	2	2	2	2	2	255
255	2	2	2	2	2	2	2	2	2	2	2	2	2	2	255
255	2	2	2	2	2	2	2	2	2	2	2	2	2	2	255
255	2	2	2	1	1	1	1	1	1	1	1	2	2	2	255
255	2	2	2	1	1	1	1	1	1	1	1	2	2	2	255
255	2	2	2	1	1	1	1	1	1	1	1	2	2	2	255
255	2	2	2	1	1	1	1	1	1	1	1	2	2	2	255
255	2	2	2	1	1	1	1	1	1	1	1	2	2	2	255
255	2	2	2	1	1	1	1	1	1	1	1	2	2	2	255
255	2	2	2	1	1	1	1	1	1	1	1	2	2	2	255
255	2	2	2	1	1	1	1	1	1	1	1	2	2	2	255
255	2	2	2	2	2	2	2	2	2	2	2	2	2	2	255
255	2	2	2	2	2	2	2	2	2	2	2	2	2	2	255
255	2	2	2	2	2	2	2	2	2	2	2	2	2	2	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255	255

(ข) แสดงตัวอย่างการเลื่อนฮิสโตแกรมหลังการฝังข้อมูล

ภาพที่ 2-3 แสดงตัวอย่างการเลื่อนฮิสโตแกรม

## 2.6 การเรียงลำดับข้อมูล (Data Sorting)

Kamstra และ Heijmans [5] ได้นำเสนอเทคนิคสำหรับการเรียงลำดับข้อมูล เพื่อให้การฝังข้อมูลมีประสิทธิภาพมากขึ้น เนื่องจากการเรียงลำดับข้อมูลทำให้พิกเซลที่มีค่าน้อยถูกจัดลำดับไว้ด้านหน้า ซึ่งเมื่อฝังข้อมูลลงไปในพื้นที่พิกเซลแล้วทำให้เกิดความบิดเบือนของภาพน้อยลง โดยสามารถคำนวณค่าความแปรปรวนข้างเคียง (Local Variance)  $\mu$  จากพิกเซลข้างเคียง 4 พิกเซล คือ  $v_{i-1,j}, v_{i+1,j}, v_{i,j+1}$  และ  $v_{i,j-1}$  ดังภาพที่ 2-3 โดยเมื่อมีการฝังข้อมูลลงไป ค่า  $\mu$  จะไม่มีการเปลี่ยนแปลง สามารถคำนวณได้ดังสมการที่ (2-13) และการกู้คืนข้อมูลสามารถคำนวณได้ดังสมการที่ (2-14)

$$\mu_{i,j} = \frac{1}{4} \sum_{k=1}^4 (\Delta v_k - \Delta \bar{v}_k)^2 \quad (2-13)$$

$$\mu'_{i,j} = \frac{1}{4} \sum_{k=1}^4 (\Delta v_k - \Delta \bar{v}_k)^2 \quad (2-14)$$

$$\begin{aligned} \text{โดย } \Delta v_1 &= |v_{i,j-1} - v_{i-1,j}|, \Delta v_2 = |v_{i-1,j} - v_{i,j+1}|, \\ \Delta v_3 &= |v_{i,j+1} - v_{i+1,j}|, \Delta v_4 = |v_{i+1,j} - v_{i,j-1}|, \\ \Delta \bar{v}_k &= (\Delta v_1 + \Delta v_2 + \Delta v_3 + \Delta v_4) / 4 \end{aligned}$$

### 2.6.1 การเรียงลำดับข้อมูลโดยใช้เทคนิค EVM

Kotvicha และคณะ [10] ได้นำเสนอหนึ่งเทคนิค EVM (Expand Variance Mean Sorting) เทคนิคดังกล่าวช่วยแก้ปัญหาการใช้สี่ตำแหน่งข้างเคียงสำหรับคำนวณค่า LV ไม่เพียงพอ โดยขยายพิทเชลสำหรับคำนวณค่าเฉลี่ยของค่า LV เดิม สามารถคำนวณได้ดังสมการนี้

$$\bar{\mu}_{i,j}^{ex} = \frac{\sum_{l=-\frac{ex}{2}}^{\frac{ex}{2}} \mu_{i+l,j+l} + \sum_{k=1}^{\frac{ex}{2}} \sum_{l=-\frac{ex}{2}}^{\frac{ex}{2}-2k} (\mu_{i+l,j+l+2k} + \mu_{i+l+2k,j+l})}{\frac{ex^2}{2} + ex + 1} \quad (2-15)$$

$$\bar{\mu}_{i,j}^{ex} = \frac{\sum_{l=-\frac{ex-1}{2}}^{\frac{ex-1}{2}} \mu_{i+l,j+l} + \sum_{k=1}^{\frac{ex-1}{2}} \sum_{l=-\frac{ex+1}{2}}^{\frac{ex+1}{2}-2k} (\mu_{i+l,j+l+2k} + \mu_{i+l+2k,j+l})}{\frac{ex^2}{2} + 2ex - \frac{3}{2}} \quad (2-16)$$

สำหรับการกู้คืนข้อมูลของเทคนิค EVM สามารถคำนวณได้ดังสมการนี้

$$\bar{\mu}_{i,j}^{ex} = \frac{\sum_{l=-\frac{ex}{2}}^{\frac{ex}{2}} \mu_{i+l,j+l} + \sum_{k=1}^{\frac{ex}{2}} \sum_{l=-\frac{ex}{2}}^{\frac{ex}{2}-2k} (\mu_{i+l,j+l+2k} + \mu_{i+l+2k,j+l})}{\frac{ex^2}{2} + ex + 1} \quad (2-17)$$

$$\bar{\mu}_{i,j}^{ex} = \frac{\sum_{l=-\frac{ex-1}{2}}^{\frac{ex-1}{2}} \mu_{i+l,j+l} + \sum_{k=1}^{\frac{ex-1}{2}} \sum_{l=-\frac{ex+1}{2}}^{\frac{ex+1}{2}-2k} (\mu_{i+l,j+l+2k} + \mu_{i+l+2k,j+l})}{\frac{ex^2}{2} + 2ex - \frac{3}{2}} \quad (2-18)$$

งานของพวกเขาให้ผลลัพธ์ที่น่าพึงพอใจสำหรับขนาดพื้นที่แมปที่ลดลง อย่างไรก็ตาม การขยายพิกเซลสำหรับคำนวณค่าการเรียงลำดับ EVM ใหม่ทุกครั้ง ส่งผลโดยตรงต่อเวลาในการประมวลผลที่ต้องใช้ระยะเวลาที่สูงขึ้นตามการขยายในแต่ละระดับ สังเกตว่าพื้นที่ที่เหมาะสมสำหรับการขยายออกมักอยู่ใกล้ ๆ ตำแหน่งศูนย์กลางของการทำนาย ดังนั้นการขยายพื้นที่ออกไปไกลมาก ๆ จึงเป็นสิ่งที่ไม่จำเป็น

## 2.7 การทดสอบแบบดับเบิล (Double Modification Testing: DMT)

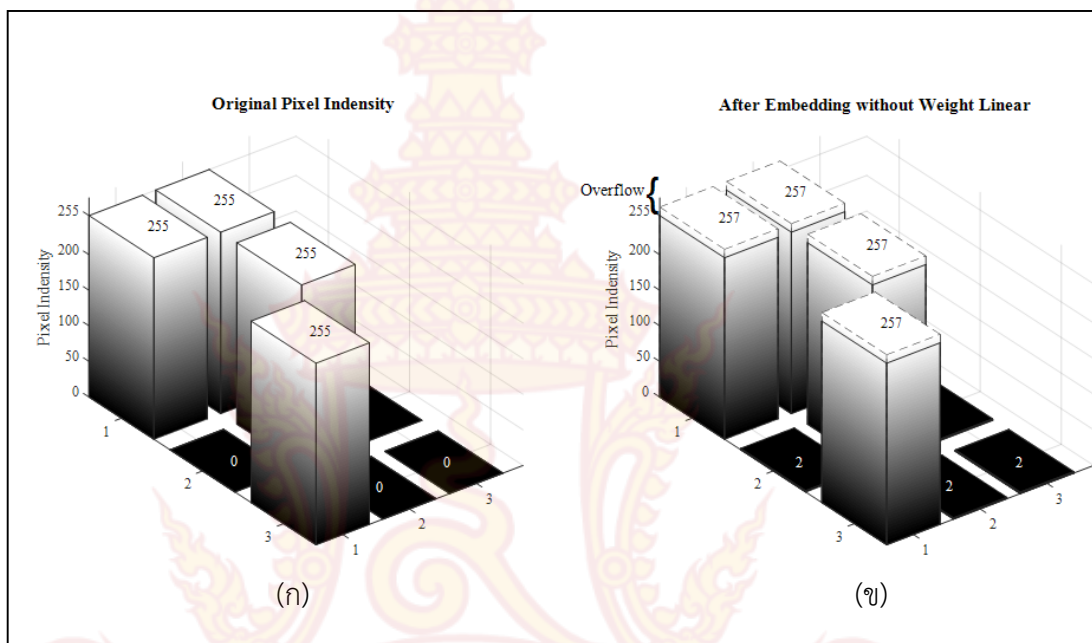
การฝังข้อมูลแบบกู่คืนกลับได้นั้นดังที่กล่าวไปแล้วข้างต้นคือ ปัญหาของการเกิด Overflow และ Underflow จากการขยาย หรือ การใช้เทคนิค HS ซึ่งไม่สามารถหลีกเลี่ยงได้ จึงต้องมีการทำเครื่องหมาย และกำหนดพื้นที่แมปขึ้น โดยขนาดของพื้นที่แมปจะถูกฝังลงไปพร้อมกับข้อมูล เพื่อเป็นการบอกตำแหน่งของการฝังข้อมูล ซึ่งเทคนิค DMT นี้ ยังช่วยให้พื้นที่แมปมีขนาดเล็กลงด้วย สำหรับขั้นตอนการเข้ารหัสถูกแบ่งไว้ 7 เซต และอีก 7 เซต สำหรับขั้นตอนการกู่คืนข้อมูล ดังนี้

1. EE (Expand-Expand) เป็นกลุ่มของการขยายค่าของพิกเซลได้สองครั้งโดยไม่ให้เกิดปัญหา Overflow และ Underflow
2. ES (Expand-Shift) เป็นกลุ่มของการขยายค่าของพิกเซลในครั้งแรก และในครั้งที่สองสามารถเลื่อนค่าได้โดยไม่เกิดปัญหา Overflow และ Underflow
3. SS (Shift-Shift) เป็นกลุ่มที่สามารถเลื่อนค่าได้สองครั้ง โดยไม่เกิดปัญหา Overflow และ Underflow
4. E (Expand) เป็นกลุ่มที่สามารถขยายค่าได้ในครั้งแรก แต่ไม่สามารถปรับเปลี่ยนค่าครั้งที่สองได้ เนื่องจากจะทำให้เกิดปัญหา Overflow และ Underflow
5. S (Shift) เป็นกลุ่มที่สามารถเลื่อนค่าได้เพียงครั้งเดียว แต่ไม่สามารถเลื่อนค่าครั้งที่สองได้ เนื่องจากจะทำให้เกิดปัญหา Overflow และ Underflow
6. NE (None-Expand) เป็นกลุ่มที่ไม่สามารถขยายค่าได้
7. NS (None-Shift) เป็นกลุ่มที่ไม่สามารถเลื่อนค่าได้

## 2.8 การฝังข้อมูลแบบกู่คืนกลับได้สำหรับภาพควอาร์โค้ด

Pandyindee และคณะ [7] ได้นำเสนอการใช้หลายเทคนิคสำหรับหนึ่งขั้นตอนวิธี RDH ร่วมกับการปรับภาพแบบเชิงเส้นสำหรับภาพควอาร์โค้ด 40 เวอร์ชัน อย่างไรก็ตาม การฝังข้อมูลลงในภาพควอาร์โค้ดพบว่าพิกเซลส่วนใหญ่ของภาพมีค่าเฉลี่ยเข้าใกล้ค่าสูงสุด และค่าต่ำสุด คือ 255 และ

0 ตามลำดับ ดังภาพที่ 2-4 (ก) และเมื่อฝังข้อมูลลงไปโดยใช้ HS จึงก่อให้เกิดปัญหา Overflow ดังภาพที่ 2-4 (ข) ทำให้ไม่สามารถฝังข้อมูลลงในพิกเซลได้ เนื่องจากต้องเก็บพื้นที่แมปหลังการฝัง



ภาพที่ 2-4 (ก) ตัวอย่างภาพควอาร์โค้ดต้นฉบับขนาด  $3 \times 3$  พิกเซล

(ข) ฝังข้อมูลลงในภาพ (ก) โดยใช้ HS [7]

จากปัญหาดังกล่าวพวกเขาจึงนำเสนอหนึ่งเทคนิคใหม่สำหรับปรับค่าเฉลี่ยตามขนาดของการฝังข้อมูล เพื่อให้สามารถฝังข้อมูลขนาดใหญ่ได้ ค่าถ่วงน้ำหนักสามารถคำนวณได้ดังสมการนี้

$$w = \frac{255 - C}{\mu_{MAX}} ; C \in \{1, 2, 3, \dots, 255\} \quad (2-19)$$

อย่างไรก็ตาม งานของพวกเขาพบว่าการเรียงลำดับโดยใช้ตำแหน่งข้างเคียงสำหรับการทำนายค่าความแปรปรวนไม่เพียงพอ

### บทที่ 3 วิธีการที่นำเสนอ

หนึ่งปัญหาสำคัญของการเรียงลำดับที่ใช้บริเวณพิกเซล 4 พิกเซลข้างเคียงของภาพที่มีการเปลี่ยนแปลงอย่างรวดเร็วสำหรับคำนวณค่าความแปรปรวนที่สอดคล้อง สังเกตว่าลำดับของค่าความคลาดเคลื่อนจากการทำนายเรียงผิดพลาดในหลายตำแหน่ง ซึ่งพบได้ในงานวิจัยของ Pandyanidee และคณะ [7] งานของพวกเขาแนะนำเสนอหนึ่งขั้นตอนวิธี RDH ร่วมกับอีกหนึ่งกลยุทธ์ใหม่สำหรับการปรับปรุงภาพควอาร์โค้ดโดยใช้การถ่วงน้ำหนักแบบเชิงเส้น (WL) วิธีการของพวกเขาสามารถแก้ปัญหาการฝังที่ก่อให้เกิด Overflow อย่างไรก็ดีตาม การเรียงลำดับข้อมูลโดยใช้ LV (ดังภาพที่ 3-1 (ข)) สำหรับภาพควอาร์โค้ดยังคงให้ผลลัพธ์การทำนายที่ไม่ถูกต้อง ค่าความคลาดเคลื่อนจากการทำนายที่สูงอยู่ก่อนค่าที่ต่ำ ดังนั้นงานวิจัยฉบับนี้แนะนำเสนอหนึ่งการปรับปรุงเทคนิคการเรียงลำดับโดยประยุกต์ใช้เทคนิค EVM ร่วมกับ WL ที่เหมาะสมเพื่อให้ได้การฝังที่มีประสิทธิภาพสูงสุด และมีความบิดเบือนต่ำสุด หนึ่งความแปรปรวนใหม่ตามรูปแบบขยาย EVM [10] สามารถคำนวณได้ดังนี้

$$\bar{\mu}_{i,j}^{ex} = \frac{\sum_{l=-\lfloor ex/2 \rfloor}^{\lfloor ex/2 \rfloor} \mu_{i+l,j+l} + \sum_{k=1}^{\lfloor ex/2 \rfloor} \sum_{l=-\lfloor ex/2 \rfloor}^{\lfloor ex/2 \rfloor - 2k} (\mu_{i+l,j+l+2k} + \mu_{i+l+2k,j+l})}{2\lceil ex/2 \rceil^2 + 2\lceil ex/2 \rceil + 1 - 4(ex \bmod 2)} \quad (3-20)$$

โดยที่  $ex$  คือ ลำดับของการขยายพิกเซลสำหรับคำนวณ EVM ใหม่

$l$  คือ ความยาวของการขยายพิกเซลออกในแต่ละมาร์ก

$k$  คือ จำนวนของระดับการขยายพิกเซล

ข้อเท็จจริงพบว่า การขยายระดับของ  $ex$  (ดังภาพที่ 3-1 (ค)) มีผลต่อการใช้จำนวนพิกเซลที่สูงขึ้นสำหรับการคำนวณ EVM ใหม่ การขยายระดับของ  $ex$  ออกไปเป็นสิ่งที่ดี แต่ในบางเวอร์ชันของภาพควอาร์โค้ด การขยาย  $ex$  ออกไปไกลมาก ๆ มักเป็นสิ่งไม่จำเป็น ซึ่งไม่ได้ช่วยให้ค่า PSNR สูงขึ้น ดังนั้นในงานวิจัยนี้ 30 ระดับของ  $ex$  จึงถูกใช้ในทดสอบ เพื่อให้ได้ผลลัพธ์ที่มีประสิทธิภาพสูงสุด รายละเอียดของขั้นตอนการฝัง และการกู้คืนอธิบายต่อในหัวข้อ 3.1 และ 3.2 ตามลำดับ

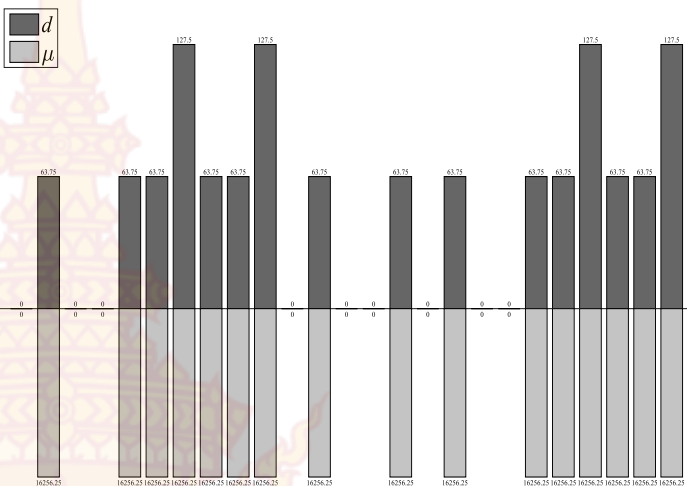


(ก) Unsorting

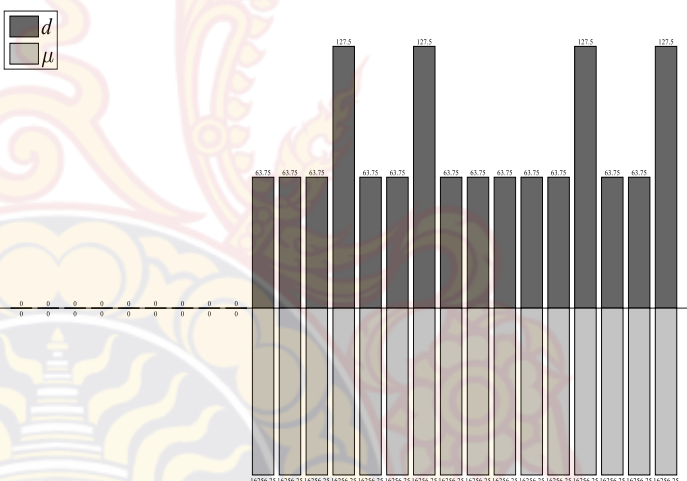
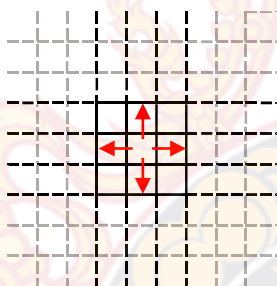


255	255	255	255	255
0	0	255	255	255
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

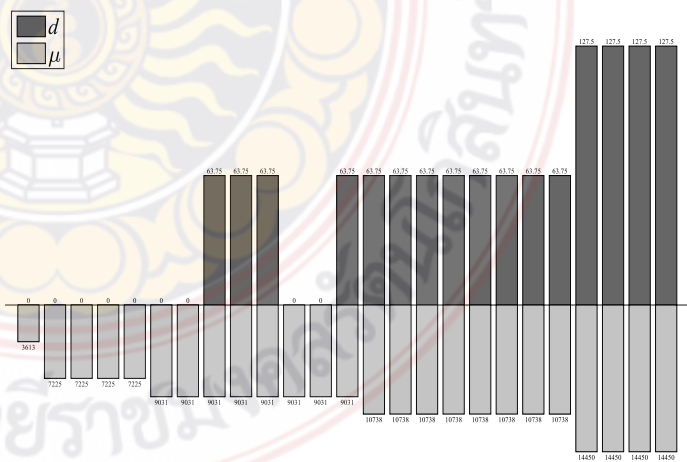
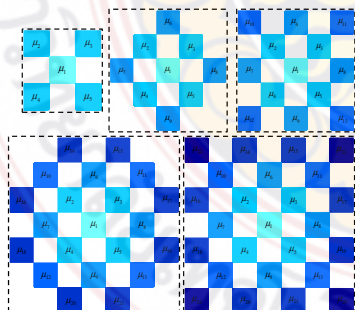
5 x 5



(ข) Sort by Local



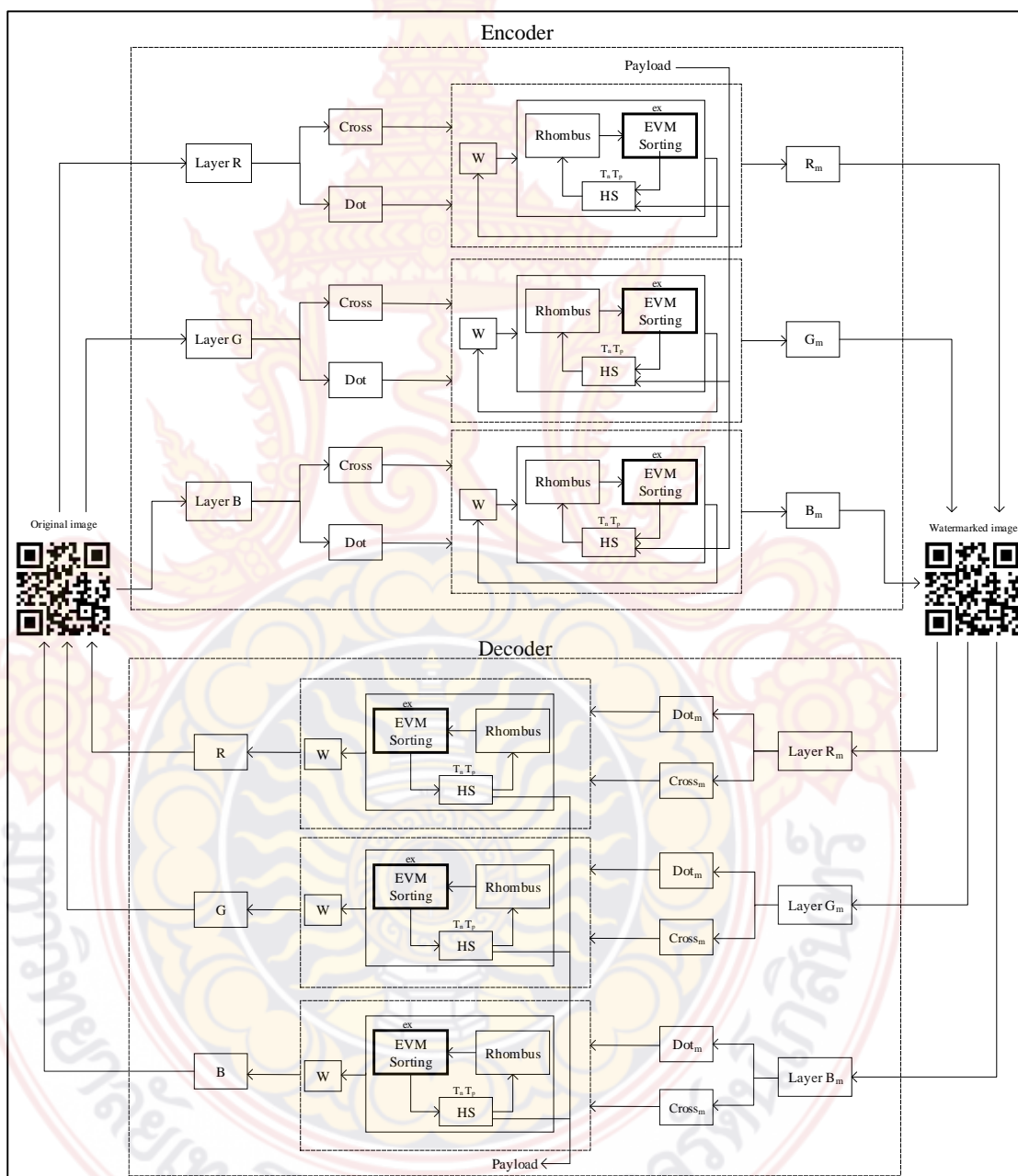
(ค) Sort by EVM



ภาพที่ 3-1 (ก) การเปรียบเทียบค่า  $d$  และ  $\mu$  โดยไม่ใช้เทคนิคการเรียงลำดับข้อมูล (Data Sorting)

(ข) การเปรียบเทียบค่า  $d$  และ  $\mu_{LV}$  โดยใช้เทคนิคการเรียงลำดับข้อมูล (Data Sorting)

(ค) การเปรียบเทียบค่า  $d$  และ  $\mu_{EVM}$  โดยใช้เทคนิคการเรียงลำดับข้อมูล (Data Sorting) [11]



ภาพที่ 3-2 ไดอะแกรมกระบวนการฝังข้อมูล และกู้คืนข้อมูล [11]

ขั้นตอนการทดสอบการฝัง และกู้คืนข้อมูลมีดังนี้

### 3.1 ขั้นตอนการฝังข้อมูล

1. นำภาพควิอาร์โค้ดสีมาแยกเลเยอร์ (Layer) ทั้ง 3 ออกจากกัน (R, G, B)
2. แบ่งพิกเซลในแต่ละเลเยอร์ออกเป็น 2 เซต (ครอสเซต และดอตเซต ตาม [6]) และปรับภาพด้วยค่าถ่วงน้ำหนัก  $W$  เริ่มจากค่าที่ 1 ตาม [7]
3. หาค่าการทำนาย  $u'_{i,j}$  จากสมการที่ (2-1), ค่าความคลาดเคลื่อนจากการทำนาย  $d_{i,j}$  จากสมการที่ (2-2) และค่าความแปรปรวนข้างเคียง  $\mu_{i,j}$  จากสมการที่ (2-13)
4. เรียงลำดับค่าความคลาดเคลื่อนจากการทำนาย  $d_{i,j}$  โดยเรียงตามค่า EVM ใหม่ เริ่มจากค่าที่ 1
5. ใช้เทคนิค DMT ตาม [6] เพื่อตรวจสอบสถานะของทุกพิกเซลก่อนการฝัง
6. หาช่วงค่าขีดแบ่ง ( $T_n$  และ  $T_p$ ) ที่เหมาะสมสำหรับการฝังข้อมูล และฝังข้อมูลโดยใช้เทคนิค HS ตาม [9]
7. ทำซ้ำขั้นตอน 3-6 โดยเรียงลำดับค่าความคลาดเคลื่อนจากการทำนาย  $d_{i,j}$  ตามค่า EVM ถัดไปจนครบ 30 ระดับ
8. ปรับภาพด้วยค่าถ่วงน้ำหนัก  $W$  ค่าถัดไป และทำตามขั้นตอนที่ 4-10 จนถึงค่า  $W$  สุดท้าย
9. แยก LSB เฮดเดอร์ขนาด 208 บิตสำหรับผู้รับเพื่อให้สามารถกู้คืนต้นฉบับ และข้อมูลที่ฝังได้ ซึ่งประกอบไปด้วยเพย์โหลด 56 บิต ค่าขีดแบ่ง ( $T_n$  และ  $T_p$ ) 16 บิต และค่าถ่วงน้ำหนักแบบเชิงเส้น 136 บิต
10. รวมเลเยอร์สีทั้ง 3 เพลน ที่ถูกฝังข้อมูลแล้ว ( $R_m, G_m, B_m$ ) เข้าด้วยกัน เสร็จสิ้นขั้นตอนการฝังข้อมูลสำหรับครอสเซต ในดอตเซตทำเหมือนกันกับครอสเซตตามลำดับ

### 3.2 ขั้นตอนการกู้คืนข้อมูล และภาพต้นฉบับ

1. แยกเลเยอร์สีที่ถูกฝังข้อมูลทั้ง 3 ออกจากกัน ( $R_m, G_m, B_m$ )
2. แยกค่าเฮดเดอร์ขนาด 208 บิตที่กันไว้ก่อนหน้าสำหรับผู้รับของแต่ละเลเยอร์ออกมาซึ่งประกอบด้วยเพย์โหลดขนาด 56 บิต, ช่วงค่าขีดแบ่งขนาด ( $T_n$  และ  $T_p$ ) 16 บิต และค่าถ่วงน้ำหนักแบบเชิงเส้นที่ได้จากการประมวลผลเบื้องต้นขนาด 136 บิต
3. แบ่งพิกเซลของแต่ละเลเยอร์ออกเป็น 2 เซตตามขั้นตอนการฝัง คำนวณหาค่าการทำนาย  $u'$  และค่า EVM โดยเรียงลำดับค่าความคลาดเคลื่อนจากการทำนาย  $d_{i,j}$  ตามค่า EVM

4. ตรวจสอบสถานะของพิกเซลโดยใช้ DMT และกู้คืนข้อมูล กู้คืนค่าการทำนายคลาดเคลื่อน
5. นำภาพคิวอาร์โค้ดที่ได้กู้คืนปรับภาพด้วยค่าถ่วงน้ำหนัก  $W$  เพื่อปรับค่าเฉดสีให้กลับเป็นค่าดั้งเดิม
6. รวมเลขเยอร์สีดั้งเดิมทั้ง 3 เพลน (R, G, B) เข้าด้วยกัน เสร็จสิ้นขั้นตอนการกู้คืนข้อมูลและภาพต้นฉบับ



## บทที่ 4

### ผลการวิจัย และผลการวิเคราะห์

ในบทที่ 4 นี้จะกล่าวถึงวิธีการทดลอง และผลลัพธ์ที่ได้จากขั้นตอนวิธีการฝังข้อมูลแบบกู้คืนกลับได้ โดยใช้การเรียงลำดับแบบ EVM ประยุกต์ใช้ร่วมกับ WL ที่เหมาะสมจากงานวิจัยของ [7] ซึ่งการเรียงลำดับแบบ EVM ได้นำเสนอไว้ในบทที่ 3 และในการทดลองได้ใช้ภาพคิวอาร์โค้ดทั้งหมด 40 เวอร์ชัน เพื่อเป็นมาตรฐานเดียวกัน และครอบคลุมทุกความเป็นไปได้ โดยภาพคิวอาร์โค้ดทั้งหมดถูกสร้างผ่านระบบออนไลน์ [12] และตั้งค่ามาตรฐานสำหรับการสร้างเช่นเดียวกับ [7] กระบวนการทั้งหมดทดสอบโดยคอมพิวเตอร์ PC CPU Core i7-7700 HQ 2.8 GHz, RAM 4 GB, HDD 256 GB บนระบบปฏิบัติการ Windows 10 Pro 64 bit ประมวลผลโดยใช้โปรแกรม MATLAB เวอร์ชัน R2015b

#### 4.1 การประเมินผล

ความบิดเบือนที่เกิดจากการฝังข้อมูลลงในภาพ สามารถวัดประสิทธิภาพของการทดลองได้ โดยการหาค่า PSNR (Peak Signal to Noise Ratio) หรือการหาอัตราส่วนของสัญญาณสูงสุดต่อสัญญาณรบกวน สามารถคำนวณได้ดังสมการนี้

$$PSNR = 10 \log_{10} \left( \frac{255^2}{MSE} \right) \quad (4-1)$$

ผลรวมของผลต่างระหว่างพิกเซลต้นฉบับกับพิกเซลที่มีการฝังข้อมูล หรือค่าความผิดพลาดกำลังสอง สมบูรณ์ MSE (Mean Squared Error) สามารถคำนวณได้ดังสมการนี้

$$MSE = \frac{\sum_{i=1}^N \sum_{j=1}^M |u_{i,j} - U_{i,j}|^2}{N \times M} \quad (4-2)$$

เมื่อ  $N$  คือ จำนวนของพิกเซลตามหลัก

$M$  คือ จำนวนของพิกเซลตามแถว

#### 4.2 ผลการทดลอง

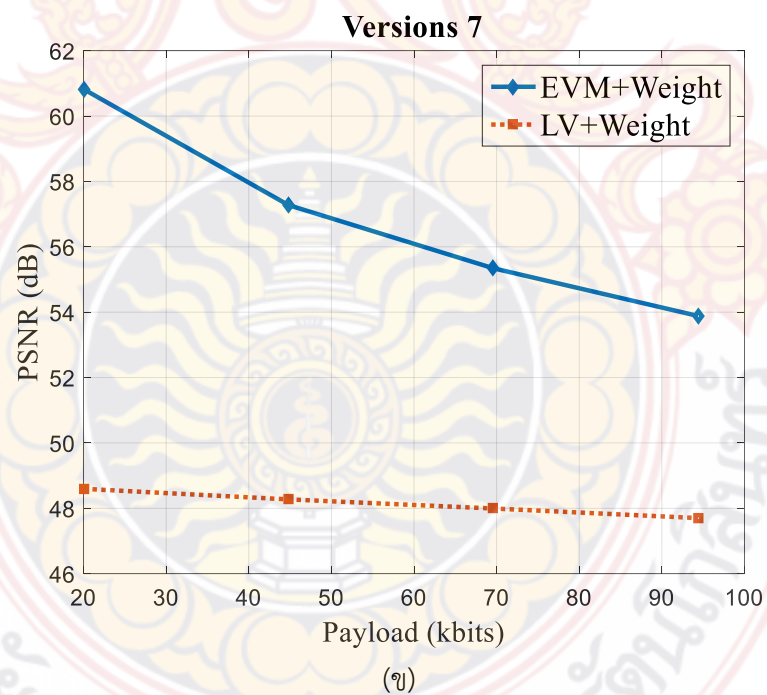
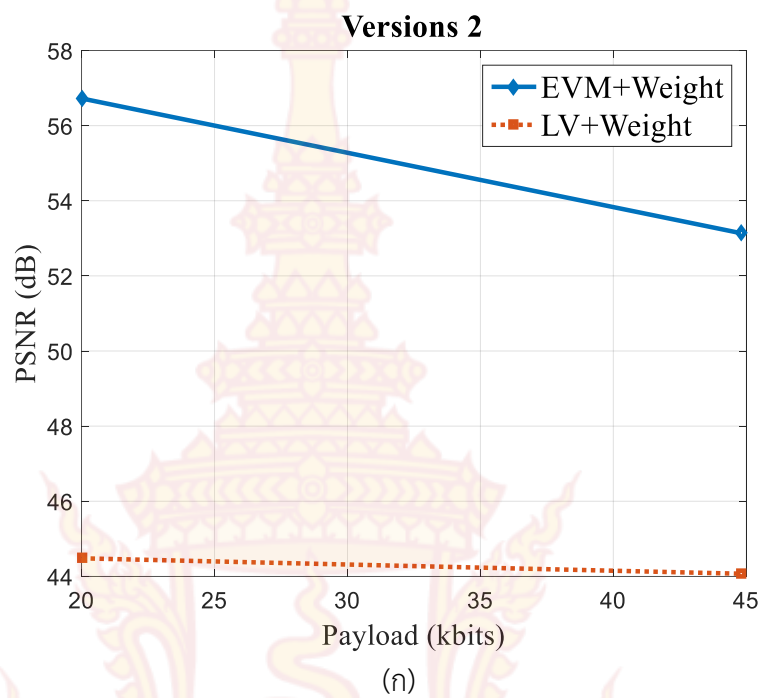
ในการทดสอบการฝังข้อมูลที่ใช้ข้อมูลบิตที่เป็นแบบสุ่ม โดยค่าถ่วงน้ำหนักที่เหมาะสมถูกใช้ร่วมกับกระบวนการ RDH ที่ปรับปรุงประสิทธิภาพการเรียงลำดับ เพื่อให้ได้รับค่า PSNR สูงสุดที่เป็นไปได้ และในส่วนนี้ EVM ใหม่ได้ถูกประยุกต์ใช้จากตารางที่ 4-1 แสดงประสิทธิภาพของวิธีการที่นำเสนอโดยฝังเพย์โหลดหลายขนาด คือ 500, 1000, 2000, 5000 และ 10000 บิต ตามลำดับกับการขยายระดับของ  $ex$  1-30 กับภาพคิวอาร์โค้ดเวอร์ชัน 40 ซึ่งเป็นเวอร์ชันสูงสุดของคิวอาร์โค้ดสังเกตว่า ที่ปริมาณการฝังขนาดเล็ก (500 บิต) ให้ผลลัพธ์ค่า PSNR ที่สูงกว่าการฝังขนาดใหญ่ (10000 บิต) และระดับการขยายที่เหมาะสมมักสูงขึ้นตามขนาดของการฝัง เนื่องมาจาก 2 ปัจจัย คือ ขนาดของภาพ และความแปรปรวนของพิกเซลภายในภาพ

#### 4.3 วิเคราะห์ผลการทดลอง

การประเมินผลจากการฝังข้อมูลลงในภาพคิวอาร์โค้ด 40 เวอร์ชัน จากตารางที่ 4-2 สังเกตได้ว่า ขนาดของการฝังที่ 367.20 กิโลบิต ภาพคิวอาร์โค้ดที่ฝังได้จะเริ่มต้นที่เวอร์ชันที่ 20-40 และระดับการขยาย  $ex$  ให้ผลลัพธ์ค่า PSNR ที่สูงกว่าแบบที่ไม่ได้มีการขยาย จากภาพที่ 4-1 สังเกตได้ว่าที่เพย์โหลดขนาดเล็กให้ความบิดเบือนที่ต่ำมาก ส่งผลให้ค่า PSNR สูงตั้งแต่เริ่มต้นการฝังเมื่อเทียบกับการฝังแบบไม่ได้มีการขยาย  $ex$  [7] กราฟแสดงค่า PSNR แสดงให้เห็นถึงผลลัพธ์ของการบิดเบือนที่ต่ำกว่า และแสดงให้เห็นว่าการเรียงลำดับข้อมูลโดยการขยายพิกเซลสำหรับการทำนายใหม่ให้ผลลัพธ์ที่ต้องเข้าใกล้ค่าความคลาดเคลื่อนจากการทำนายสูงกว่าแบบที่ไม่มีการขยายพิกเซล พิกเซลที่มีการแปรปรวนสูง ๆ ในภาพให้การตอบสนองที่ดีมากสำหรับการประยุกต์ใช้ EVM ร่วมกับ WL ที่เหมาะสม ถึงแม้ว่าการคำนวณจะสูงขึ้น อย่างไรก็ตาม ค่า PSNR ที่ได้รับ และขนาดของการฝังที่ใหญ่ขึ้นเป็นสิ่งที่น่าพึงพอใจสำหรับการนำเสนอ

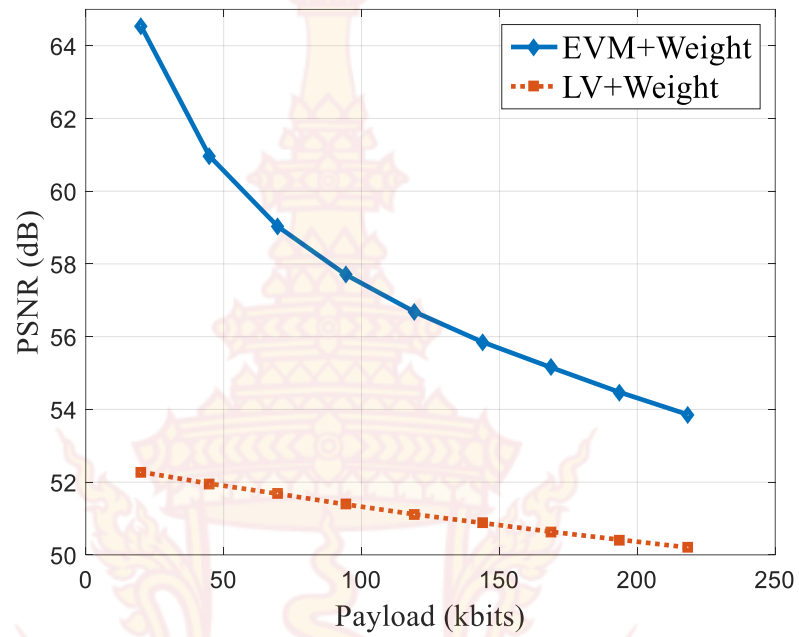
ตารางที่ 4-1 ผลลัพธ์การฝังข้อมูล ex 1-30 ของภาพ QR Code เวอร์ชัน 40

ex	Payload (bits)									
	500		1000		2000		5000		10000	
	PSNR	Weight	PSNR	Weight	PSNR	Weight	PSNR	Weight	PSNR	Weight
1	88.034	0.98431	84.797	0.98431	81.851	0.98431	77.512	0.97647	74.606	0.97647
2	87.734	0.98431	85.023	0.98431	81.731	0.98431	77.557	0.97647	74.583	0.97647
3	87.957	0.98431	84.985	0.98431	81.686	0.98431	77.525	0.97647	74.647	0.97647
4	88.151	0.98431	84.927	0.98431	81.851	0.98431	77.546	0.97647	74.576	0.97647
5	87.957	0.98431	84.927	0.98431	81.842	0.98431	77.581	0.97647	74.639	0.97647
6	88.034	0.98431	84.985	0.98431	81.833	0.98431	77.836	0.97647	74.679	0.97647
7	86.431	0.98431	84.039	0.98431	81.349	0.98431	77.571	0.97647	74.636	0.97647
8	88.073	0.98431	84.946	0.98431	81.946	0.98431	77.627	0.97647	74.629	0.97647
9	88.073	0.98431	84.834	0.98431	81.842	0.98431	77.807	0.97647	74.709	0.97647
10	88.151	0.98431	84.965	0.98431	81.870	0.98431	77.738	0.97647	74.664	0.97647
11	86.485	0.98431	84.293	0.98431	81.357	0.98431	77.585	0.97647	74.613	0.97647
12	88.034	0.98431	84.871	0.98431	81.879	0.98431	77.623	0.97647	74.604	0.97647
13	88.151	0.98431	84.985	0.98431	81.917	0.98431	77.613	0.97647	74.625	0.97647
14	86.431	0.98431	84.023	0.98431	81.459	0.98431	77.595	0.97647	74.636	0.97647
15	86.567	0.98431	84.070	0.98431	81.382	0.98431	77.553	0.97647	74.604	0.97647
16	88.073	0.98431	85.102	0.98431	81.842	0.98431	77.785	0.97647	74.673	0.97647
17	86.299	0.98431	84.101	0.98431	81.416	0.98431	77.616	0.97647	74.601	0.97647
18	86.457	0.98431	84.117	0.98431	81.399	0.98431	77.644	0.97647	74.636	0.97647
19	86.377	0.98431	84.039	0.98431	81.349	0.98431	77.627	0.97647	74.597	0.97647
20	86.325	0.98431	84.023	0.98431	81.416	0.98431	77.641	0.97647	74.641	0.97647
21	88.440	0.98431	85.121	0.98431	81.889	0.98431	77.763	0.97647	74.704	0.97647
22	86.485	0.98431	84.070	0.98431	81.536	0.98431	77.606	0.97647	74.631	0.97647
23	88.073	0.98431	85.043	0.98431	81.974	0.98431	77.595	0.97647	74.622	0.97647
24	88.483	0.98431	84.946	0.98431	81.908	0.98431	77.666	0.97647	74.695	0.97647
25	88.232	0.98431	85.004	0.98431	81.898	0.98431	77.763	0.97647	74.722	0.97647
26	86.457	0.98431	84.117	0.98431	81.510	0.98431	77.634	0.97647	74.623	0.97647
27	88.232	0.98431	85.023	0.98431	81.889	0.98431	77.789	0.97647	74.731	0.97647
28	88.355	0.98431	85.082	0.98431	81.870	0.98431	77.800	0.97647	74.716	0.97647
29	88.073	0.98431	85.043	0.98431	81.861	0.98431	77.818	0.97647	74.724	0.97647
30	88.112	0.98431	85.121	0.98431	81.870	0.98431	77.789	0.97647	74.720	0.97647



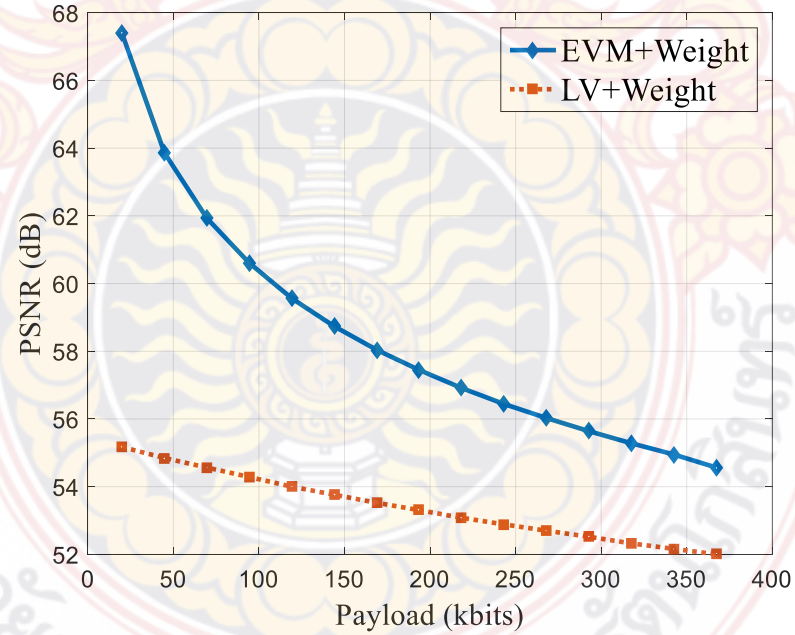


Versions 14

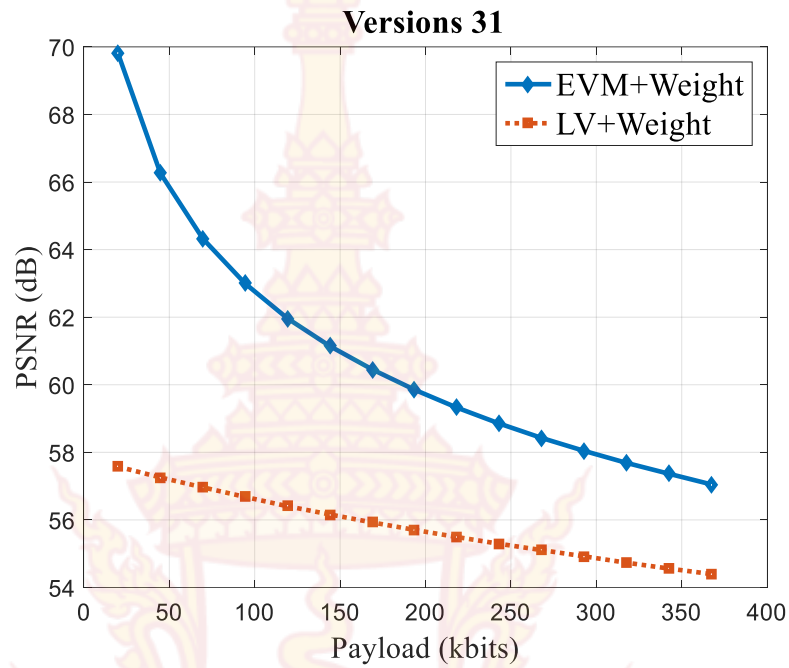


(ค)

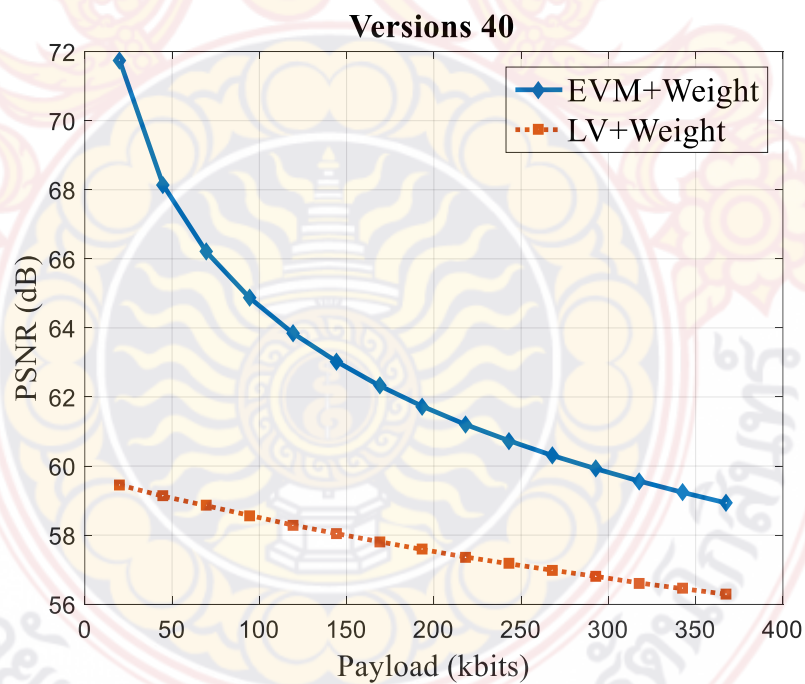
Versions 22



(ง)



(จ)



(ข)

ภาพที่ 4-1 กราฟแสดงค่า PSNR VS Payload ของวิธีการที่นำเสนอ (EVM+Weight)

โดยเปรียบเทียบกับงานในอดีต (LV+Weight) สำหรับ 6 ภาพควอาร์โค้ด

ตารางที่ 4-2 ผลลัพธ์ค่า PSNR และค่าถ่วงน้ำหนักที่ดีที่สุดสำหรับภาพคิวอาร์โค้ด 40 เวอร์ชัน

QR Code Image	Payload (20 kbits)			Payload (193.60 kbits)			Payload (367.20 kbits)		
	PSNR (dB)	Weight	ex	PSNR (dB)	Weight	ex	PSNR (dB)	Weight	ex
Version 1	55.575	0.98824	12	-	-	-	-	-	-
Version 2	56.724	0.98039	30	-	-	-	-	-	-
Version 3	57.703	0.97255	30	-	-	-	-	-	-
Version 4	58.611	0.98039	15	-	-	-	-	-	-
Version 5	59.409	0.97255	30	-	-	-	-	-	-
Version 6	60.134	0.98824	12	-	-	-	-	-	-
Version 7	60.824	0.97255	30	-	-	-	-	-	-
Version 8	61.463	0.98039	15	-	-	-	-	-	-
Version 9	62.068	0.98824	12	-	-	-	-	-	-
Version 10	62.639	0.98824	30	-	-	-	-	-	-
Version 11	63.118	0.98039	8	-	-	-	-	-	-
Version 12	63.625	0.97255	12	53.649	0.97255	30	-	-	-
Version 13	64.056	0.98039	30	53.956	0.97255	8	-	-	-
Version 14	64.547	0.98039	15	54.471	0.97255	5	-	-	-
Version 15	64.957	0.98039	15	54.975	0.98431	30	-	-	-
Version 16	65.336	0.98824	30	55.367	0.98039	30	-	-	-
Version 17	65.710	0.98824	30	55.758	0.98039	12	-	-	-
Version 18	66.076	0.98824	12	56.115	0.97255	30	-	-	-
Version 19	66.419	0.97255	12	56.466	0.98039	5	-	-	-
Version 20	66.792	0.97255	30	56.814	0.98824	30	53.814	0.97255	12
Version 21	67.126	0.98431	30	57.139	0.98824	12	54.208	0.98431	15
Version 22	67.388	0.98039	8	57.447	0.98039	30	54.566	0.98039	12
Version 23	67.719	0.98039	8	57.751	0.98431	30	54.948	0.97255	12
Version 24	68.003	0.98039	15	58.037	0.98039	30	55.252	0.98431	30
Version 25	68.276	0.98039	30	58.322	0.98039	8	55.532	0.98431	30
Version 26	68.554	0.97255	30	58.601	0.97255	30	55.799	0.97255	30
Version 27	68.822	0.97255	8	58.858	0.97255	30	56.068	0.97647	12
Version 28	69.076	0.98039	30	59.117	0.98039	5	56.326	0.97647	12
Version 29	69.309	0.97255	12	59.374	0.98039	8	56.575	0.98039	30
Version 30	69.564	0.98824	12	59.616	0.97255	12	56.816	0.97255	8

QR Code Image	Payload (20 kbits)			Payload (193.60 kbits)			Payload (367.20 kbits)		
	PSNR (dB)	Weight	ex	PSNR (dB)	Weight	ex	PSNR (dB)	Weight	ex
Version 31	69.802	0.97647	30	59.852	0.98824	30	57.053	0.98824	30
Version 32	70.025	0.98039	30	60.074	0.98824	12	57.287	0.97255	12
Version 33	70.284	0.97255	12	60.311	0.98039	15	57.510	0.97255	30
Version 34	70.475	0.97647	30	60.524	0.98824	30	57.725	0.98824	30
Version 35	70.688	0.97647	30	60.738	0.98824	30	57.939	0.98824	30
Version 36	70.896	0.97647	30	60.946	0.98824	30	58.147	0.98824	30
Version 37	71.092	0.98039	30	61.142	0.98824	12	58.354	0.97255	12
Version 38	71.326	0.97255	12	61.353	0.98039	15	58.551	0.97255	30
Version 39	71.487	0.98039	30	61.543	0.97255	30	58.747	0.98039	30
Version 40	71.743	0.98824	15	61.726	0.98039	3	58.935	0.97255	12



## บทที่ 5

### สรุปผล และข้อเสนอแนะ

ในบทที่ 5 นี้จะกล่าวถึงการสรุปผล และข้อเสนอแนะต่าง ๆ ของงานวิจัย โดยแบ่งออกเป็นสรุปผล ปัญหา แนวทางการแก้ปัญหา การนำไปใช้ประโยชน์/แนวทางการประยุกต์ใช้ และการพัฒนาต่อยอดในลักษณะอื่น ๆ

#### 5.1 สรุปผล

งานวิจัยนี้นำเสนอหนึ่งการปรับปรุงกระบวนการ RDH สำหรับภาพคิวอาร์โค้ดโดยประยุกต์ใช้เทคนิค EVM ร่วมกับ WL บนพื้นฐานของขั้นตอนวิธี RDH การปรับระดับที่เหมาะสมถูกใช้ประโยชน์อย่างมีนัยสำคัญต่อความบิดเบือนของภาพหลังการฝัง การขยายพิกเซลสำหรับคำนวณค่าการเรียงลำดับใหม่ของค่าความคลาดเคลื่อนจากการทำนายเป็นสิ่งจำเป็นโดยเฉพาะลักษณะภาพที่มีความแปรปรวนของพิกเซลสูง ๆ วิธีการที่นำเสนอให้ปริมาณการฝังที่สูงกว่า ในขณะที่ความบิดเบือนหลังการฝังเมื่อเปรียบเทียบกับงานเมื่อ เร็ว ๆ นี้

#### 5.2 ปัญหา

ในการทดสอบการฝังข้อมูลพบว่าการเรียงลำดับโดยใช้เทคนิค EVM ให้ผลลัพธ์ที่ดีในการฝังข้อมูลในปริมาณที่น้อยถึงปานกลาง สำหรับการฝังข้อมูลในปริมาณมากจะส่งผลกระทบต่อความบิดเบือนของภาพหลังการฝัง นั่นคือค่า PSNR จะต่ำ เนื่องจากค่าความผิดพลาดจากการทำนายที่มีค่าสูงจะถูกใช้ จึงทำให้มีความบิดเบือนเพิ่มมากขึ้น

#### 5.3 แนวทางการแก้ปัญหา

เพื่อให้การจัดลำดับค่าความคลาดเคลื่อนจากการทำนายมีความถูกต้องมากขึ้น ควรใช้รูปแบบการทำนายรูปแบบอื่นให้เหมาะสมกับภาพในลักษณะต่าง ๆ

#### 5.4 การนำไปใช้ประโยชน์/แนวทางการประยุกต์ หรือ การพัฒนาต่อยอดในลักษณะอื่น ๆ

สำหรับการนำไปใช้ประโยชน์ ในปัจจุบันคิวอาร์โค้ดได้ถูกนำมาใช้อย่างแพร่หลาย เช่น ในทางธุรกิจ การประชาสัมพันธ์ หรือในงานที่ต้องใช้ในการเก็บข้อมูลสำคัญต่าง ๆ ซึ่งข้อมูลเหล่านั้นจำเป็นต้องเก็บเป็นความลับซึ่งสามารถทำได้โดยใช้เทคนิคการฝังข้อมูลแบบกู้คืนกลับได้ สำหรับการต่อ

## บรรณานุกรม

1. J. Tian., 2003, “Reversible Data Embedding Using a Difference Expansion”, **IEEE Trans. Circuits Syst. Video Technol**, Vol. 13, No. 8, pp. 890-896.
2. A. M. Alattar., 2003, “Reversible Watermark using Difference Expansion of Triplets”, **in Proc. Int. Conf. Image Process.**, Barcelona, Spain, Vol. 1, pp. 501-504.
3. A. M. Alattar., 2004, “Reversible Watermark using Difference Expansion of Quads”, **in Proc. Int. Conf.**, Speech, and Signal Processing, pp. III377-III380.
4. A. M. Alattar., 2004. “Reversible Watermark Using the Difference Expansion of a Generalized Integer Transform”, **IEEE Trans. Image Processing**, Vol. 13, No. 8, pp. 1147-1156.
5. L. H. J. Kamstra and A. M. Heijmans., 2005, “Reversible Data Embedding Into Images Using Wavelet Techniques and Sorting”, **IEEE Trans. Image Process.**, Vol. 14, No. 12, pp. 2082-2090.
6. V. Sachnev, H. J. Kim, J. Nam, S. Suresh, and Y.-Q. Shi., 2009, “Reversible Watermarking Algorithm Using Sorting and Prediction”, **IEEE Trans. Circuit and System**, Vol. 19, No. 7, pp. 989-999.
7. ชัยพร ปานยินดี, ชีรนัย ลีลาวิวัฒน์ และภัทรวดี รังสิรักษ์, “การศึกษา และพัฒนาขั้นตอนวิธีการฝังข้อมูลแบบกู้คืนกลับได้โดยใช้การถ่วงน้ำหนักแบบเชิงเส้นสำหรับภาพคิวอาร์โค้ด”, วารสารวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่.
8. ปรีญาทิพย์ แดงมี และวรพล ลีลาเกียรติสกุล. 2015. “การอำพรางข้อมูลลับด้วย Steganography บน QRCode”, **Journal of Information Science and Technology**, Vol. 5, No. 1, pp. 35 – 41.
9. D. M. Thodi and J. J. Rodriguez., 2007, “Expansion Embedding Techniques for Reversible Watermarking”, **IEEE Trans. Image Process.**, Vol. 16, No. 3, pp. 721-730.
10. Aekawit Kotvicha, Parinya Sanguansat, and M. L. Kulthon Kasemsa., 2012, “Expand Variance Mean Sorting for Reversible Watermarking”, **International Journal of Computer and Communication Engineering**, Vol. 1, No.3, pp. 196-199.

11. ชัยพร ปานยินดี, อีรนัย สีสาววัฒน์ และภัทรวดี รังสิรักษ์, “การปรับปรุงวิธีการเรียงลำดับข้อมูลในภาพคิวอาร์โค้ดโดยใช้เทคนิค EVM สำหรับการฝังข้อมูลแบบกู้คืนกลับได้”, วารสารวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่.
12. Project Nayuki. QR Code generator library, ออนไลน์ [4 ตุลาคม 2560], สาระสังเขป สืบค้นจาก : <https://www.nayuki.io/page/qr-code-generator-library>.





ประวัติผู้วิจัย



## ประวัติผู้วิจัย

1. ชื่อ สกุล                      ดร.ชัยพร ปานยินดี
2. ตำแหน่งปัจจุบัน            หัวหน้าสาขาวิชาวิศวกรรมคอมพิวเตอร์
3. หน่วยงานที่สามารถติดต่อได้  
 คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีราชมงคลรัตนโกสินทร์  
 96 หมู่ 3 ตำบลศาลายา อําเภอกุสุมาลย์ จังหวัดนครปฐม 73170  
 หมายเลขโทรศัพท์ที่ทำงาน 0-28894585-7  
 โทรศัพท์มือถือ 081-2850997  
 โทรสาร 0-28894585-7 7  
 ไปรษณีย์อิเล็กทรอนิกส์ (e-mail)  
 chaiyaporn.pan@gmail.com, chaiyaporn.pan@rmutr.ac.th
4. ประวัติการศึกษา  
 ปริญญาเอก สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง หลักสูตร  
 วิศวกรรมศาสตรดุษฎีบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า ปี พ.ศ. 2559  
 ปริญญาโท มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี หลักสูตรวิศวกรรม  
 ศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า ปี พ.ศ.2552  
 ปริญญาตรี มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี หลักสูตรครุศาสตร์  
 อุดสาหกรรมบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า ปี พ.ศ. 2550
5. สาขาวิชาการที่มีคามชำนาญพิเศษ  
 Image processing, information hiding
6. ประสบการณ์ที่เกี่ยวข้องกับการบริหารงานวิจัย  
 - C. Panyindee and W. Chiracharit, “Detection of Lung Tumors in  
 Chest PET/CT images Mapping”, Proceedings of The 2th Biomedical

Engineering International Conference (BMEiCON-2008), Bangkok, Thailand, Nov 10-11, 2008.

- **C. Panyindee** and W. Chiracharit, “Registration of Lung Tumors in Chest PET/CT Images”, Proceedings of International Symposium on Multimedia and Communication Technology (ISMAC 2009), Siam City Hotel, Bangkok, Thailand, January 22-23, 2009.

- **C. Panyindee**, P. Tosranon and C. Pintavirooj, “Robustness of Geodesics to Affine Transformation”, Proceedings of The 4th Biomedical Engineering International Conference (BMEiCON-2011), Chiang Mai, Thailand, Nov 9-11, 2011.

- **C. Panyindee** and C. Pintavirooj, “Reversible Watermarking Algorithm in Application for Medical Images”, Proceedings of The 5th Biomedical Engineering International Conference (BMEiCON-2012), Sunee Grand Hotel, Ubon Ratchathani, Thailand and Champasak, Laos, Dec 5-7, 2012.

- **C. Panyindee** and C. Pintavirooj, “Reversible Watermarking Using Gaussian Weight Prediction and Genetic Algorithm”, Proceedings of The International MultiConference of Engineers and Computer Scientists (IMAC2013), The Royal Garden Hotel, Kowloon, Hong Kong, vol.1, pp. 457-461, Mar. 2013.

- **C. Panyindee** and C. Pintavirooj, “Optimizations Using the Genetic Algorithm for Reversible Watermarking”, Proceedings of The 4th Annual International Conference of the Electrical Engineering/Electronics, Computer, Telecommunications, and Information Technology Association (ECTI-CON'13), Kribi, Thailand, pp. 1-5, May 15-17, 2013.

- **C. Panyindee** and C. Pintavirooj, “QR Codes Application for Reversible Watermarking Algorithm in Biomedical Images”, Proceedings of The 6th Biomedical Engineering International

Conference (BMEiCON-2013), Kribi, Thailand, pp. 155, Oct 23-25, 2013.

- **C. Panyindee**, “DMT-Based Reversible Watermarking for CT Images”, Proceedings of The 7th Biomedical Engineering International Conference (BMEiCON-2015), Pattaya, Thailand, Oct 25-27, 2015.

- **C. Panyindee**, “Appropriate Variance Mean for Sorting Based on a Reversible Watermarking Algorithm”, Proceedings of The International MultiConference of Engineers and Computer Scientists (IMAC2016), The Royal Garden Hotel, Kowloon, Hong Kong, vol.1, Mar. 2016.

