



ใบรับรองวิทยานิพนธ์
บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์
วิทยาศาสตร์มหาบัณฑิต (วิทยาการคอมพิวเตอร์)
ปริญญา

วิทยาการคอมพิวเตอร์

วิทยาการคอมพิวเตอร์

สาขา

ภาควิชา

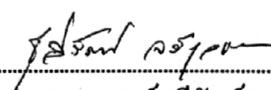
เรื่อง การบีบอัดภาพแบบแฟร็กทัลด้วยการหมุนในองศาอิสระบนขั้นตอนวิธีการค้นหา
โหนดใกล้เคียงที่สุด

Fractal Image Compression Using Free-Rotation Scheme on the Nearest Neighbor
Searching Algorithm

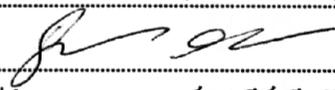
นามผู้วิจัย นายชาวีร์ อิศริยภัทร์

ได้พิจารณาเห็นชอบโดย

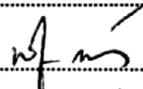
ประธานกรรมการ

( รongศาสตราจารย์สุธีรัตน์ จรัสกุลชัย, D.Sc.)

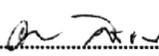
กรรมการ

( ผู้ช่วยศาสตราจารย์อุษณีย์ ธีรวัดน์, วท.ค.)

กรรมการ

( ผู้ช่วยศาสตราจารย์พันธุ์ปิติ เปี่ยมสง่า, D.Sc.)

หัวหน้าภาควิชา

( ผู้ช่วยศาสตราจารย์อุมาพร ศิรชรานนท์, M.S.)

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์รับรองแล้ว

( รongศาสตราจารย์วินัย อางคงหาญ, M.A.)

คณบดีบัณฑิตวิทยาลัย

วันที่ 3 เดือน เมษายน พ.ศ. 2549

วิทยานิพนธ์

เรื่อง

การบีบอัดภาพแบบแฟร็กทัลด้วยการหมุนในองศาอิสระบนขั้นตอนวิธีการค้นหา
โหนดใกล้เคียงที่สุด

Fractal Image Compression Using Free-Rotation Scheme on the Nearest Neighbor
Searching Algorithm

โดย

นายชาวีร์ อิศริยภัทร์

เสนอ

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์
เพื่อความสมบูรณ์แห่งปริญญาวิทยาศาสตรมหาบัณฑิต (วิทยาการคอมพิวเตอร์)

พ.ศ. 2549

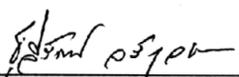
ISBN 974-16-1325-3

ชาวีร์ อิศริยภัทร์ 2549: การบีบอัดภาพแบบแฟร็กทัลด้วยการหมุนในองศาอิสระบน
ขั้นตอนวิธีการค้นหาโทนคโคสที่เสถียรที่สุด ปรินญาวิทยาศาสตร์มหาบัณฑิต (วิทยาการ
คอมพิวเตอร์) สาขาวิชาวิทยาการคอมพิวเตอร์ ภาควิชาวิทยาการคอมพิวเตอร์
ประธานกรรมการที่ปรึกษา: รองศาสตราจารย์สุธีรัตน์ จรัสกุลชัย, D.Sc. 85 หน้า
ISBN 974-16-1325-3

การบีบอัดแบบแฟร็กทัล (fractal compression) เป็นเทคโนโลยีการบีบอัดข้อมูลชนิดใหม่ที่มีแนวโน้มที่จะให้ประสิทธิภาพการบีบอัดสูงกว่าวิธีการบีบอัดแบบดั้งเดิม ภาพจะถูกเข้ารหัสในรูปของฟังก์ชันนิยามความคล้ายระหว่างส่วนต่างๆ เพื่อที่จะให้ได้อัตราส่วนการบีบอัดที่สูงและคุณภาพของภาพที่ดี ฟังก์ชันต้องมีจำนวนน้อย และผลิตความผิดพลาดในระดับต่ำ ดังนั้นจึงมีการนำการหมุนบนจุดรัศ และการพลิก รวมทั้งการปรับค่าระดับสีเทา มาผนวกส่วนประกอบของฟังก์ชันเพื่อเป็นการสร้างความหลากหลายของรูปแบบ อย่างไรก็ตาม การค้นหาความคล้ายสามารถทำได้ในองศาจากเท่านั้น ทั้งนี้เนื่องจากธรรมชาติของการแปลงบนจุดรัศดังกล่าว ความคล้ายตัวเองในทิศทางอื่นจึงอาจถูกมองข้ามไป ส่งผลให้ขั้นตอนวิธีการบีบอัดไม่มีประสิทธิภาพเท่าที่ควรจะเป็น

เพื่อจำกัดข้อจำกัดนี้ วิทยานิพนธ์ฉบับนี้จึงนำเสนอแนวคิดของการหมุนในองศาอิสระบนโดเมนบล็อคด้วยการส่งคองแบบ ซึ่งมีผลให้ความเพี้ยนของฟังก์ชันลดลง และความถูกต้องของภาพสูงขึ้น และยังประยุกต์วิธีนี้เข้ากับขั้นตอนวิธีการค้นหาโทนคโคสที่เสถียรที่สุดซึ่งได้รับการยอมรับว่าเป็นวิธีการค้นหาที่มีประสิทธิภาพสูงสำหรับการทำการบีบอัดแบบแฟร็กทัล ถึงแม้ว่าการหมุนในมุมที่ละเอียดนั้น จะต้องการบิตพิเศษเพื่อใช้นิยามรูปแบบการหมุนที่เพิ่มขึ้นมา แต่จากผลการทดลองพบว่า วิธีนี้สามารถลดจำนวนการแตกควอดทรีพาร์ติชันลงได้ และลดได้มากพอที่จะชดเชยขนาดโดยรวมของบิตพิเศษที่เพิ่มขึ้นมา โดยเฉพาะในช่วงของอัตราการบีบอัดต่ำ ที่ซึ่งคุณภาพของภาพสำคัญกว่าขนาด ดังนั้นขั้นตอนวิธีที่นำเสนอนี้จึงให้ประสิทธิภาพที่ดีกว่าวิธีเดิม ในแง่ของคุณภาพของภาพ และขนาดของไฟล์

ชาวีร์ อิศริยภัทร์
ลายมือชื่อนิติ


ลายมือชื่อประธานกรรมการ

21 / 55 / 45

Chavee Issariyapat 2006: Fractal Image Compression Using Free-Rotation Scheme on the Nearest Neighbor Searching Algorithm. Master of Science (Computer Science), Major Field: Computer Science, Department of Computer Science. Thesis Advisor: Associate Professor Chulcerat Jaraskulchai, D.Sc. 85 pages.
ISBN 974-16-1325-3

Fractal image compression is a new generation of compression technology which claims to be more efficient than typical compression methods. Image is coded in the form of functions of similarity between two different parts. To achieve high compression ratio with high image fidelity, the number of such functions needs to be minimized and each of them must produce as little distortion as possible. Therefore square rotation and flipping has been used, along with the gray-level adjustment, as compositions of functions in order to generate variety of mapping patterns. However, the similarity searching can be done only in perpendicular angles due to the nature of such square isometry. Therefore some better self-similarity in other directions may be skipped and the compression algorithm might not be as efficient as it could be.

To overcome this limitation, this thesis proposed the idea of free-rotation on domain blocks using conformal mapping instead of typical square rotation. This reflected in lower collage error and higher PSNR. Moreover, this method was applied to the nearest neighbor searching scheme which is regarded as a very efficient searching algorithm for fractal image compression. Although more bits were needed to specify more delicate rotation angles, the experimental result showed some advantages in compression efficiency. In general, the number of quadtree partitions that this method can reduce was enough to compensate the total amount of those extra bits. Especially in the low compression range where image quality is concerned. Therefore, the algorithm gave better image fidelity while compression ratio was higher comparing to the typical algorithm.

Chavee Issariyapat

Student's signature

Chulcerat Jaraskulchai

Thesis Advisor's signature

21 / 255 / 49

กิตติกรรมประกาศ

ข้าพเจ้าขอกราบขอบพระคุณ รศ.ดร.ชวลีรัตน์ จรัสกุลชัย ประธานกรรมการที่ปรึกษา ที่ได้ช่วยเหลือในการวางแผนงานวิจัยในวิทยานิพนธ์ฉบับนี้ ตลอดจนให้คำปรึกษา และตรวจสอบแก้ไขข้อบกพร่องต่างๆ ขอกราบขอบพระคุณ ผศ.ดร.อุษณีย์ ลีรัตน์ กรรมการที่ปรึกษาวิชาเอก และ ผศ.ดร.พันธุ์ปิติ เปี่ยมสง่า กรรมการที่ปรึกษาวิชารอง ที่กรุณาให้คำปรึกษาแนะนำ และช่วยเหลือในการทำวิทยานิพนธ์ให้สำเร็จลุล่วงไปด้วยดี

ขอขอบพระคุณ คุณพ่อ คุณแม่ พี่น้อง และเพื่อนๆทุกคน ที่คอยให้ความช่วยเหลือ ให้กำลังใจ ชี้แนะ และสนับสนุนการทำวิทยานิพนธ์จนสำเร็จลุล่วงได้

ชาวีร์ อีสริยภัทร์

กุมภาพันธ์ 2549

สารบัญ

	หน้า
สารบัญ	(1)
สารบัญตาราง	(3)
สารบัญภาพ	(4)
คำนำ.....	1
การตรวจเอกสาร	3
ทฤษฎีพื้นฐานที่เกี่ยวข้อง.....	3
โมเดลของภาพ.....	3
การบีบอัดข้อมูลภาพ.....	4
ตัวแบบของการบีบอัดข้อมูลภาพ.....	6
แฟร็กทัล	7
ไอเอฟเอส	8
การบีบอัดภาพแบบแฟร็กทัล	13
ขั้นตอนวิธีค้นแบบในการเข้ารหัสภาพ	14
ขั้นตอนวิธีในการถอดรหัสภาพ	18
ขั้นตอนวิธีพื้นฐานที่ใช้ในงานวิจัย.....	19
การแปลงเชิงเรขาคณิตบนภาพระดับสีเทา.....	19
โครงสร้างข้อมูลแบบเคคิตรี	23
ตัวดำเนินการนอมัลไลซ์โปรเจกชัน.....	26
การส่งแบบ และ การส่งซวาร์ช-คริสทอฟเฟิล.....	28
นิยามของการส่งแบบ.....	28
การส่งแบบซวาร์ช-คริสทอฟเฟิล	30
งานวิจัยที่เกี่ยวข้อง	33

สารบัญ (ต่อ)

	หน้า
อุปกรณ์และวิธีการ	36
อุปกรณ์	36
วิธีการ	36
แนวคิดของงานวิจัย	36
ฟังก์ชันการหมุนภายในสี่เหลี่ยมจัตุรัส	37
การทดลอง	41
จุดประสงค์การทดลอง	41
ปัจจัยที่มีผลต่อคุณภาพของภาพและอัตราส่วนการบีบอัด	42
ขั้นตอนการทำงานของโปรแกรมบีบอัดภาพ	42
การเก็บข้อมูลลงไฟล์	44
ขั้นตอนการทดลอง	46
ผลการทดลอง	49
การทดลองที่ 1	49
การทดลองที่ 2	50
การทดลองที่ 3	56
การทดลองที่ 4	56
วิจารณ์ผลการทดลอง	58
การทดลองที่ 1	58
การทดลองที่ 2	60
การทดลองที่ 3	73
การทดลองที่ 4	76
สรุป	81
เอกสารและสิ่งอ้างอิง	82

สารบัญตาราง

ตารางที่		หน้า
1	ตัวอย่างของตารางการหมุนที่คำนวณได้สำหรับบล็อกขนาด 4×4	39
2	ส่วนเสกเตอร์ของไฟล์	45
3	ส่วนข้อมูลสำหรับแต่ละพาร์ติชัน ซึ่งจะระบุโดเมนและพารามิเตอร์ของฟังก์ชัน	45
4	แสดงความหมายของแต่ละโหนดการหมุน	47
5	ผลการทดลองของการทดลองที่ 1	49
6	ผลการทดลองของการทดลองที่ 2 ในการบีบอัดภาพ Lenna	50
7	ผลการทดลองของการทดลองที่ 2 ในการบีบอัดภาพ Peppers	51
8	ผลการทดลองของการทดลองที่ 2 ในการบีบอัดภาพ Rock	52
9	ผลการทดลองของการทดลองที่ 2 ในการบีบอัดภาพ Fabric	53
10	ผลการทดลองของการทดลองที่ 2 ในการบีบอัดภาพ Colonel	54
11	ผลการทดลองของการทดลองที่ 2 ในการบีบอัดภาพ France	55
12	ผลการทดลองของการทดลองที่ 3 เมื่อใช้ควอดทรี 3 ระดับ	56
13	ผลการทดลองของการทดลองที่ 3 เมื่อใช้ควอดทรี 4 ระดับ	56
14	ผลการทดลองของการทดลองที่ 4	57

สารบัญภาพ

ภาพที่		หน้า
1	ตัวอย่างการแทนค่าสีด้วยตัวเลขในภาพสเกลสีเทา	3
2	ขั้นตอนของการบีบอัดและการถอดรหัส	6
3	Mandelbrot set และลักษณะความคล้ายตัวเอง	7
4	ตัวอย่างของไอเอฟเอส middle-thirds Cantor set และ Koch curve.....	7
5	แสดงลักษณะของความคล้ายตัวเองในของวัตถุในธรรมชาติ	7
6	ภาพไบเฟิร์นสร้างจากตัวเลข 24 ตัว	8
7	การดำเนินการพื้นฐานของการแปลงสัมพรรค.....	9
8	ตัวอย่างการลู่เข้าสู่จุดตรึงของคอนแทรกชันบน \mathbb{R}^2	10
9	ตัวอย่างไอเอฟเอสที่ประกอบด้วยคอนแทรกชัน T_1, T_2, T_3	11
10	ตัวอย่างแฟร็กทัล Sierpinski Triangle	12
11	ตัวอย่างของการใช้เซตเริ่มต้นที่ต่างกัน แต่ลำดับลู่เข้าสู่แอตแทรกเตอร์เดียวกัน	12
12	แสดงพื้นผิวสามมิติที่ได้จากภาพระดับสีเทา.....	14
13	แสดงตัวอย่างการจับคู่โดเมนและเรนจ์บล็อก	15
14	ตัวอย่างการแบ่งพาร์ติชันแบบควอดทรี	17
15	ตัวอย่างภาพในแต่ละขั้นตอนขณะทำการถอดรหัส	18
16	แสดงการส่งพิกเซลด้วยขั้นตอนวิธีการส่งไปข้างหน้า.....	19
17	แสดงการส่งพิกเซลด้วยขั้นตอนวิธีการส่งย้อนกลับ	20
18	การประมาณค่าแบบไบลิเนียร์	21
19	การประมาณค่าแบบในช่วงแบบไบคิวบิก	22
20	แสดงเคคิทีที่สร้างขึ้นจากการแบ่งตามคีย์แต่ละหลักหมุนเวียนกันไป	23
21	แสดงกรณีการเกิด Bounds Overlap Ball	24
22	แสดงการแทนค่าไฮเปอร์เพลน H ในรูปของ (\vec{v}, μ)	25
23	แสดงความหมายเชิงเรขาคณิตของนอมอลไลซ์โปรเจกชัน	27
24	แสดงตัวอย่างของฟังก์ชัน f ซึ่ง conformal ที่จุด z_0	29
25	ตัวอย่างการส่งคงแบบ $f(z) = \frac{1}{z}$	29

สารบัญภาพ (ต่อ)

ภาพที่		หน้า
26	ตัวอย่างฟังก์ชัน $f(z) = \frac{1+z}{1-z}i$	29
27	แสดงลักษณะการแมพแกนจำนวนจริงของฟังก์ชัน f	31
28	แสดงตัวอย่างฟังก์ชัน f ที่สร้างขึ้นด้วยการส่งแบบซวาร์ช-คริสทอฟเฟิล.....	32
29	พาร์ติชันแบบ Quadtree, HV และ Polygonal.....	34
30	พาร์ติชันแบบ Region-Based	34
31	แสดงการเชื่อมโยงของฟังก์ชันที่ประกอบเป็นการหมุนบนสี่เหลี่ยมจตุรัส.....	37
32	ลักษณะการส่งแบบอนุรักษั่มุมตัดระหว่างสี่เหลี่ยมกับวงกลม.....	38
33	แสดงผลของการหมุนภายในรูปสี่เหลี่ยมด้วยการส่งคงแบบ	41
34	แสดงโดเมนทั้งหมดที่ได้จากการดึงออกมาจากส่วนต่างๆของภาพ.....	43
35	แสดงการเรียงตัวใหม่ของโดเมนที่โหนดใบภายในหลังกระบวนการสร้างเคิลทรี.....	44
36	ภาพที่ใช้ในการทดลอง	46
37	ความสัมพันธ์ระหว่างค่า m กับค่า PSNR ของโหมด 2222	58
38	ความสัมพันธ์ระหว่างค่า m กับค่า PSNR ของโหมด 4444	59
39	ความสัมพันธ์ระหว่างค่า m กับค่า PSNR ของโหมด 6666	59
40	กราฟแสดงอัตราส่วนการบีบอัดและ PSNR ของภาพ Lenna.....	60
41	กราฟแสดงค่าความผิดพลาดที่ยอมรับได้กับจำนวนของฟังก์ชันของภาพ Lenna	62
42	แสดงการแบ่งพาร์ติชันของโหมด 2222 เปรียบเทียบกับโหมด 6666.....	62
43	กราฟแสดงอัตราส่วนการบีบอัดและ PSNR ของภาพ Peppers.....	63
44	กราฟแสดงค่าความผิดพลาดที่ยอมรับได้กับจำนวนของฟังก์ชันของภาพ Peppers.....	64
45	กราฟแสดงอัตราส่วนการบีบอัดและ PSNR ของภาพ Fabric	65
46	กราฟแสดงค่าความผิดพลาดที่ยอมรับได้กับจำนวนของฟังก์ชันของภาพ Fabric	66
47	กราฟแสดงอัตราส่วนการบีบอัดและ PSNR ของภาพ Rock.....	67
48	กราฟแสดงค่าความผิดพลาดที่ยอมรับได้กับจำนวนของฟังก์ชันของภาพ Rock.....	68
49	กราฟแสดงอัตราส่วนการบีบอัดและ PSNR ของภาพ Colonel.....	69
50	กราฟแสดงค่าความผิดพลาดที่ยอมรับได้กับจำนวนของฟังก์ชันของภาพ Colonel.....	70
51	กราฟแสดงอัตราส่วนการบีบอัดและ PSNR ของภาพ France.....	71

สารบัญภาพ (ต่อ)

ภาพที่		หน้า
52	กราฟแสดงความผิดพลาดที่ยอมรับได้กับจำนวนของฟังก์ชันของภาพ France	72
53	กราฟแสดงโหมดการหมุนและเวลาที่ใช้ เมื่อใช้ควอดทรี 3 ระดับ.....	73
54	กราฟแสดงรูปแบบรวมของการหมุนและเวลาที่ใช้ เมื่อใช้ควอดทรี 3 ระดับ	74
55	กราฟแสดงโหมดการหมุนและเวลาที่ใช้ เมื่อใช้ควอดทรี 4 ระดับ.....	74
56	กราฟแสดงรูปแบบรวมของการหมุนและเวลาที่ใช้ เมื่อใช้ควอดทรี 4 ระดับ	75
57	กราฟเปรียบเทียบการบีบอัดแบบแฟร็กทัลกับเจบีคของภาพ Lenna.....	76
58	กราฟเปรียบเทียบการบีบอัดแบบแฟร็กทัลกับเจบีคของภาพ Peppers	76
59	กราฟเปรียบเทียบการบีบอัดแบบแฟร็กทัลกับเจบีคของภาพ Fabric.....	77
60	กราฟเปรียบเทียบการบีบอัดแบบแฟร็กทัลกับเจบีคของภาพ Rock	77
61	กราฟเปรียบเทียบการบีบอัดแบบแฟร็กทัลกับเจบีคของภาพ Colonel	78
62	กราฟเปรียบเทียบการบีบอัดแบบแฟร็กทัลกับเจบีคของภาพ France	78
63	ภาพขั้นตอนการถอดรหัสแบบดั้งเดิม	79
64	ภาพขั้นตอนการถอดรหัสแบบมีการหมุนด้วยการส่งคงแบบส่วน.....	80

การบีบอัดภาพแบบแฟร็กทัลด้วยการหมุนในองศาอิสระบนขั้นตอนวิธีการ ค้นหาโหนดใกล้เคียงที่สุด

Fractal Image Compression Using Free-Rotation Scheme on the Nearest Neighbor Searching Algorithm

คำนำ

ขั้นตอนวิธีที่ใช้ในการบีบอัดขนาดภาพได้ถูกพัฒนามาเป็นเวลานาน ในปัจจุบันเรามีไฟล์ฟอร์แมตสำหรับจัดเก็บข้อมูลภาพให้เลือกใช้มากมายหลายสิบชนิด หากแยกตามลักษณะของคุณภาพของภาพภาพหลังการบีบอัดจะแยกได้เป็น 2 ประเภท คือ การบีบอัดที่ไม่มีการสูญเสีย (Lossless Compression) ฟอร์แมตภาพที่อยู่ในประเภทนี้ เช่น พีซีเอ็กซ์ (PCX), ทีไอเอฟ (TIF), พีเอ็นจี (PNG) ฯลฯ ภาพหลังการบีบอัดจะเหมือนต้นฉบับทุกประการ ไม่มีผิดเพี้ยน การบีบอัดภาพอีกประเภทหนึ่งคือ การบีบอัดที่มีการสูญเสีย (Lossy Compression) ข้อดีของลักษณะการบีบอัดประเภทนี้คือ อัตราการส่วนการบีบอัดที่สูงกว่ามาก แต่ก็ต้องยอมให้มีการปรับเปลี่ยนรายละเอียดเล็กน้อยๆบางอย่างในภาพที่สังเกตได้ยาก เพื่อแลกกับขนาดของข้อมูลที่เล็กลง ฟอร์แมตภาพในประเภทนี้ที่รู้จักกันดี เช่น เจเป็ก (JPG) และ กิฟ (GIF)

เป็นความจริงที่ว่าไม่มีขั้นตอนวิธีการบีบอัดข้อมูลชนิดใดในโลกที่สมบูรณ์แบบ สามารถใช้ลดขนาดข้อมูลในทุกรูปแบบได้ ดังสามารถแสดงให้เห็นได้ด้วยหลักการรังนกพิราบ (pigeon hole principle) ว่าหากเรานำขั้นตอนวิธีนั้นไปใช้ลดขนาดข้อมูลอันหนึ่งได้ผลทำให้ข้อมูลเล็กลง ก็จะต้องมีข้อมูลอย่างน้อยอีกชุดหนึ่งซึ่งเมื่อผ่านขั้นตอนวิธีเดียวกันแล้วขนาดจะใหญ่ขึ้น (Barnsley and Hurd, 1992) เช่นเดียวกัน ไฟล์ฟอร์แมตที่ใช้กันอยู่ในปัจจุบันต่างก็มีข้อดีข้อด้อยด้วยกันทั้งนั้น เช่นการบีบอัดแบบพีซีเอ็กซ์จะเหมาะสำหรับภาพที่มีพื้นหลังเป็นสีเดียวกันติดๆกัน เนื่องจากลักษณะของการเข้ารหัสแบบรันเลนท (run-length encoding) จะทำโดยลดความซ้ำซ้อนของสีที่อยู่ติดกัน หากนำไปใช้กับภาพที่มีความแปรปรวนของสีมาก ก็อาจจะให้ประสิทธิภาพที่ไม่ดีนัก รูปแบบการบีบอัดแบบเจเป็กเองก็อาจให้ผลไม่ค่อยดีนักกับภาพที่ประกอบไปด้วยเส้นขอบหรือมีการเปลี่ยนแปลงของค่าสีแบบฉับพลันจำนวนมาก ทั้งนี้เนื่องจากเจเป็กใช้หลักการของการแปลงโคไซน์แบบไม่ต่อเนื่อง (discrete cosine transform) ภาพที่เหมาะสมสำหรับเจเป็ก จึงเป็นภาพที่มีลักษณะเป็นการไล่โทนสีแบบต่อเนื่อง เช่นภาพถ่ายโดยทั่วไป ขั้นตอนวิธีการบีบอัดภาพจึงยังคง

ต้องมีการพัฒนาอย่างต่อเนื่อง เพื่อให้มีความเหมาะสมครอบคลุมคุณลักษณะของภาพหลากหลายประเภท

การบีบอัดแบบแฟร็กทัล (fractal compression) เป็นเทคโนโลยีการบีบอัดข้อมูลชนิดใหม่ที่อาศัยคุณลักษณะร่วมอย่างหนึ่งของภาพ ที่เรียกว่า ความคล้ายตัวเอง (self-similarity) เป็นส่วนสำคัญในการเข้ารหัส เทคโนโลยีนี้เริ่มเป็นที่สนใจ เนื่องจากในบางกรณีสามารถให้อัตราส่วนการบีบอัดที่สูงมากเมื่อเทียบกับวิธีการบีบอัดแบบอื่น และยังมีแนวโน้มที่จะสามารถพัฒนาต่อไปได้อีก แต่การบีบอัดแบบแฟร็กทัลเองก็ยังมีข้อเสียที่เป็นอุปสรรคสำคัญต่อการนำไปประยุกต์ใช้งานจริง ก็คือในขั้นตอนของการบีบอัดที่ยังต้องใช้เวลาในการประมวลผลที่นานมาก จึงมีงานวิจัยจำนวนมากที่นำเสนอเทคนิคต่างๆ ในการปรับปรุงประสิทธิภาพของการบีบอัดแบบแฟร็กทัล ทั้งทางด้าน การเพิ่มอัตราส่วนการบีบอัด การลดเวลาในการบีบอัด และการเพิ่มคุณภาพของภาพที่ผ่านการบีบอัด

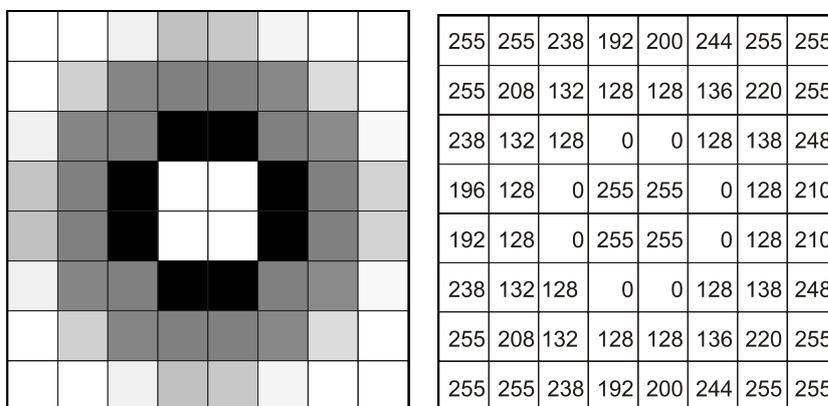
งานวิทยานิพนธ์ชิ้นนี้ เป็นการเสนอแนวทางในการปรับปรุงขั้นตอนวิธีของการบีบอัดแบบแฟร็กทัลชนิดที่มีการสูญเสีย (lossy compression) โดยมีพื้นฐานอยู่บนขั้นตอนวิธีการค้นหาโหนดใกล้เคียงที่สุด (nearest neighbor searching) ที่ถูกเสนอเป็นครั้งแรกโดย Saupe (Saupe, 1995) ซึ่งได้รับการยอมรับว่าเป็นขั้นตอนวิธีที่มีประสิทธิภาพสูงในแง่ของความสามารถในการลดเวลาการประมวลผล งานวิจัยชิ้นนี้เป็นการปรับปรุงขั้นตอนวิธีให้สามารถค้นหาความคล้ายในองศาอื่นๆ นอกเหนือจากการหมุนเป็นมุมฉาก เพื่อเป็นการเพิ่มโอกาสในการจับคู่โดเมนและเรนจ์ที่ดีที่สุด ผลการทดลองพบว่าขั้นตอนวิธีที่เสนอให้ผลลัพธ์ดีกว่าวิธีเดิมในทุกรูปแบบของภาพที่ใช้ทดลองในแง่ของคุณภาพของภาพที่ผ่านการบีบอัด ส่วนอัตราส่วนการบีบอัดนั้น พบว่าขั้นตอนวิธีที่นำเสนอสามารถให้อัตราส่วนการบีบอัดที่สูงกว่าในระดับ PSNR เดียวกันเมื่อตั้งค่าของจำนวนรูปแบบการหมุนในระดับที่เหมาะสม

การตรวจเอกสาร

1. ทฤษฎีพื้นฐานที่เกี่ยวข้อง

1.1 โมเดลของภาพ (Image Formation Model)

โดยทั่วไปภาพก็คือฟังก์ชันในรูปแบบ $f(x,y)$ ที่ให้ค่าเป็นจำนวนจริงบวก ภาพใน ความหมาย โดยทั่วไปอาจมีลักษณะที่ต่อเนื่อง ทั้งทางด้านของพิกัดจุด และค่าสี แต่ในทาง คอมพิวเตอร์ซึ่งข้อมูลมีลักษณะเป็นดิจิทัล โมเดลของภาพในทางคอมพิวเตอร์โดยทั่วไปจึงมี ลักษณะเป็นฟังก์ชันจากระนาบ xy ที่ไม่ต่อเนื่องซึ่งเรียกว่าพิกเซล (pixel) ไปยังจำนวนเต็ม ความหมายของค่าฟังก์ชันที่จุดต่างๆซึ่งใช้แทนค่าสีจะแตกต่างกันออกไปตามตัวแบบสี (color model) ที่เลือกใช้ เช่น RGB, CMYK, HSI สำหรับในงานวิจัยนี้จะใช้ภาพสเกลสีเทา 8 บิต (8 bit-grayscale) ซึ่งค่าของฟังก์ชัน $f(x,y)$ จะเป็นจำนวนเต็มที่มีค่าในช่วงตั้งแต่ 0 ถึง 255 ตามระดับ ความสว่างของจุดสี โดยที่ 0 คือสีดำ และ 255 คือสีขาว



ภาพที่ 1 ตัวอย่างการแทนค่าสีด้วยตัวเลขในภาพสเกลสีเทา

1.2 การบีบอัดข้อมูลภาพ (Image Compression)

ในการจัดเก็บข้อมูลภาพ หากใช้วิธีการบันทึกตามปกติ ในกรณีที่ภาพเป็นรูปสี่เหลี่ยม ข้อมูลภาพจะมีขนาดเท่ากับ ความกว้างของภาพ \times ความสูงของภาพ \times ขนาดข้อมูลสีต่อจุด ทั้งนี้ยังไม่รวมถึงข้อมูลในส่วนเฮดเดอร์ที่บอกถึงขนาดภาพ ตัวแบบสีที่เลือกใช้ และรายละเอียดต่างๆที่จำเป็นในการถอดรหัสข้อมูลกลับมาเป็นภาพ จะเห็นได้ว่าภาพภาพหนึ่งหากบันทึกด้วยวิธีการปกติ อาจมีขนาดใหญ่มากถึงหลายแสนหลายล้านไบต์ ดังนั้นเราจึงนิยมบีบอัดข้อมูลภาพก่อนการจัดเก็บ ซึ่งเทคนิคที่นิยมใช้ในการบีบอัดข้อมูลภาพได้แก่การลดความซ้ำซ้อนของข้อมูล (data redundancy) ประเภทต่างๆ ลักษณะของความซ้ำซ้อนของข้อมูลภาพที่สำคัญมี 3 รูปแบบ (Gonzalez and Woods, 2002) ได้แก่

1.2.1 Coding Redundancy ในการเข้ารหัสเพื่อแทนข้อมูลภาพ บางครั้งอาจเกิดความซ้ำซ้อนของการเข้ารหัสได้ ยกตัวอย่าง เช่น การใช้วิธีบันทึกค่าสีแต่ละสีเป็นสายอักขระที่มีจำนวนของบิตเท่ากันทั้งหมดทั้งๆที่ความถี่ของการปรากฏของสีแต่ละสีมีค่าไม่เท่ากัน จะถือว่าเกิดความซ้ำซ้อนของการเข้ารหัสขึ้น เนื่องจากหากเราใช้ตัวอักษรความยาวบิตไม่คงที่และแทนสีที่ปรากฏบ่อยๆด้วยอักขระที่มีความยาวบิตสั้น แทนสีที่ปรากฏเป็นจำนวนน้อยลงลดหลั่นลงมาด้วยอักขระที่มีความยาวบิตยาวขึ้นตามลำดับ ขนาดรวมของข้อมูลจะสั้นลงกว่าเดิม เทคนิคสำคัญที่ใช้ในการลดความซ้ำซ้อนลักษณะนี้ เช่น การเข้ารหัสฮัฟฟ์แมน (Huffman coding), การเข้ารหัสเชิงจำนวน (Arithmetic coding), การเข้ารหัสแอลแซดดับเบิลยู (LZW)

1.2.2 Interpixel Redundancy โดยทั่วไปจุดภาพที่อยู่ใกล้กันมักจะมีสีสัมพันธ์กันไม่ทางใดก็ทางหนึ่ง การเข้ารหัสภาพโดยไม่คำนึงถึงคุณลักษณะนี้ จึงถือว่าข้อมูลเกิดความซ้ำซ้อนขึ้น เทคนิคการลดความซ้ำซ้อนลักษณะนี้ เช่น การเข้ารหัสแบบลดความยาว (run length encoding), การเข้ารหัสแบบคาดคะเน (predictive coding) โดยการอาศัยฟังก์ชันคาดคะเนในการทายค่าสีที่ควรจะเป็นซึ่งคำนวณได้จากจุดสีบางส่วนที่ปรากฏอยู่ก่อนหน้า ข้อมูลที่จะถูกนำไปเข้ารหัสคือผลต่างระหว่างข้อมูลภาพจริงๆ กับผลของการคาดคะเนด้วยฟังก์ชัน หากฟังก์ชันคะเนที่เลือกใช้นั้นทำงานได้ดี กล่าวคือสามารถทำนายได้ตรงเป็นส่วนใหญ่ เศษเหลือที่นำไปเข้ารหัสก็จะประกอบไปด้วยค่าศูนย์เป็นจำนวนมาก ซึ่งสายอักขระของข้อมูลนี้จะสามารถบีบอัดได้เล็กลงมากด้วยเทคนิคการลดความซ้ำซ้อนของการเข้ารหัส ในการถอดรหัสข้อมูลกลับ ก็เพียงแต่นำค่าที่ถอดรหัสได้ มาบวกด้วยค่าที่คำนวณได้จากฟังก์ชันคาดคะเน (จุดสีที่ฟังก์ชันคาดคะเนต้องใช้ในการคำนวณเป็นจุดสีที่อยู่ก่อน

หน้า ซึ่งถูกถอดรหัสเรียบร้อยแล้ว) ก็จะได้ค่าของจุดสีจริงๆกลับมา ดังนั้นทั้งตัวเข้ารหัสและตัวถอดรหัสจะต้องทำความเข้าใจในเรื่องรูปแบบของฟังก์ชันคาบเดาที่จะเลือกใช้ในแต่ละกรณี

1.2.3 Psychovisual Redundancy ในการใช้งานภาพโดยทั่วไปนั้น เราไม่ได้ต้องการความถูกต้องทุกรายละเอียดของจุดภาพแบบสมบูรณ์แบบ เพราะความสำคัญของภาพไม่ได้ขึ้นอยู่กับค่าจุดสีที่คงที่ตายตัว การที่ค่าจุดสีมีการผิดเพี้ยนไปบ้างเล็กน้อยในระดับที่สังเกตได้ยาก ก็ยังถือว่าเป็นภาพนั้นเป็นภาพเดิมเช่นเดียวกัน ดังนั้นการจัดเก็บข้อมูลภาพแบบไม่ยอมให้ผิดเพี้ยน ทั่วๆที่มีขนาดข้อมูลใหญ่กว่าที่ควรจะเป็นจึงนับเป็นความซ้ำซ้อนส่วนเกินรูปแบบหนึ่ง บางครั้งเราจึงยอมให้มีการปรับเปลี่ยนรายละเอียดบางอย่าง เพื่อที่จะให้ข้อมูลภาพมีขนาดเล็กลง เทคนิคที่สำคัญในการลดความซ้ำซ้อนในรูปแบบนี้ได้แก่ การควอนไทซ์ (quantization) ซึ่งก็คือการลดจำนวนระดับขั้นของข้อมูล ซึ่งทำได้ทั้งในโดเมนเชิงพื้นที่ (spatial domain) เช่นการลดความละเอียดของภาพ, การลดความถี่ของสี และในโดเมนเชิงความถี่ (frequency domain) เช่นการควอนไทซ์ค่าพารามิเตอร์ของการแปลงโคไซน์แบบไม่ต่อเนื่อง (discrete cosine transform) ซึ่งเป็นวิธีที่ใช้ในไฟล์รูปแบบ JPEG

รูปแบบของการบีบอัดข้อมูลภาพ แบ่งกว้างๆได้เป็น 2 ประเภท ได้แก่

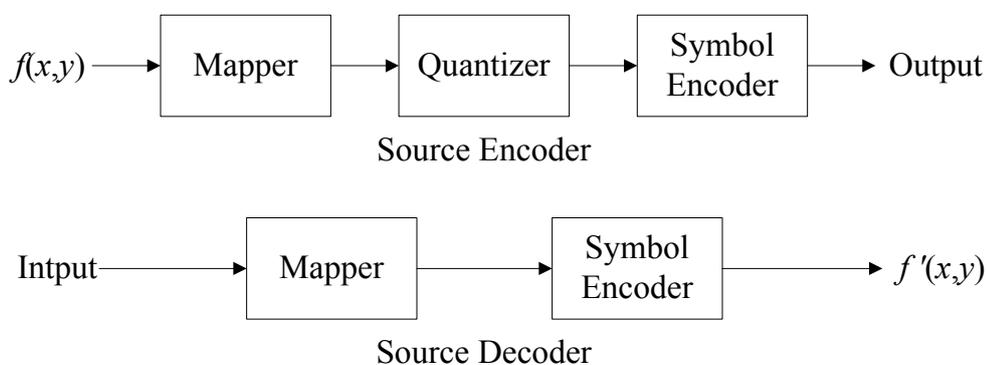
1.) การบีบอัดแบบไม่มีการสูญเสีย (lossless compression) การบีบอัดลักษณะนี้ ข้อมูลที่ผ่านการบีบอัดและขยายออกจะมีลักษณะเหมือนเดิมทุกประการ โดยทั่วไปอัตราการบีบอัดมักจะทำได้ไม่เกิน 3:1 (Gonzalez and Woods, 2002)

2.) การบีบอัดแบบมีการสูญเสีย (lossy compression) ซึ่งจะยอมให้ข้อมูลภาพบางส่วนถูกเปลี่ยนแปลงไปในระดับที่ยังใช้การได้ เพื่อแลกกับอัตราส่วนการบีบอัดที่สูงขึ้นมาก ตั้งแต่ 3:1 ขึ้นไปจนถึง มากกว่า 100:1 แต่ยิ่งบีบอัดมากเท่าไรภาพก็จะมีความผิดเพี้ยนไปจากต้นฉบับมากเท่านั้น

1.3 ตัวแบบของการบีบอัดข้อมูลภาพ (Image Compression Model)

ส่วนประกอบของการเข้ารหัสส่วนแรกคือ mapper ซึ่งทำหน้าที่ในการแปลงข้อมูลภาพให้อยู่ในรูปของอาร์เรย์ของสัมประสิทธิ์ที่มีการลด interpixel redundancy หรือทำให้ความซ้ำซ้อนส่วนนี้ถูกจัดการได้ง่ายขึ้น กระบวนการนี้เป็นกระบวนการที่สามารถทำย้อนกลับได้ ส่วนประกอบถัดมาคือ quantizer ทำหน้าที่ในการลดแม่นยำของค่าสัมประสิทธิ์ เพื่อเป็นการลดขนาดของข้อมูล กระบวนการนี้จะทำให้ข้อมูลเกิดการสูญเสีย และเป็นกระบวนการที่ทำย้อนกลับไม่ได้ ขั้นตอนนี้จะไม่มีในการบีบอัดแบบที่ไม่มีการสูญเสีย ส่วนประกอบสุดท้ายของการเข้ารหัสคือ symbol encoder ซึ่งทำหน้าที่แปลงสัมประสิทธิ์ให้อยู่ในรูปของรหัสที่ลดความซ้ำซ้อนของการเข้ารหัสแล้ว

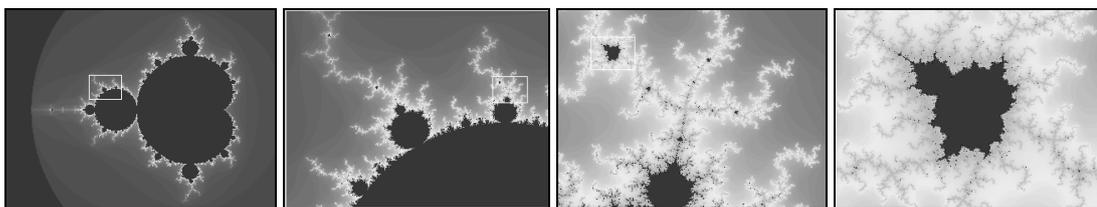
ส่วนขั้นตอนของการถอดรหัสจะเป็นการทำย้อนกลับกระบวนการเดิม เพียงแต่จะไม่มี การย้อนกลับของกระบวนการที่ผ่าน quantizer เท่านั้น ข้อมูลภาพ $f(x,y)$ ที่ได้จากการถอดรหัส อาจจะมีขนาดคลาดเคลื่อนไปจากข้อมูลภาพ $f(x,y)$ ต้นฉบับบ้างหากการบีบอัดนั้นเป็นแบบที่มีการสูญเสีย



ภาพที่ 2 ขั้นตอนของการบีบอัดและการถอดรหัส

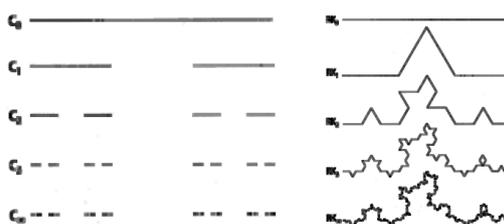
1.4 แฟร็กทัล (Fractal)

ในทางคณิตศาสตร์ ยังไม่มีคำนิยามที่แน่นอนสำหรับสิ่งที่เรียกว่าแฟร็กทัล แต่โดยทั่วไปแล้ว นักคณิตศาสตร์ใช้คำนี้เพื่อเรียกวัตถุพิเศษรูปแบบหนึ่ง ที่มีลักษณะของรายละเอียดปรากฏอยู่ในทุกสเกล และมีลักษณะของความคล้ายตัวเอง (self-similarity) ตัวอย่างของแฟร็กทัลในทางคณิตศาสตร์มีอยู่มากมาย ที่มีชื่อเสียงมากคือ Mandelbrot set (Mandelbrot, 1982) ซึ่งเป็นเซตที่สร้างจากสมการจำนวนเชิงซ้อน $f_c(z) = z^2 + c$

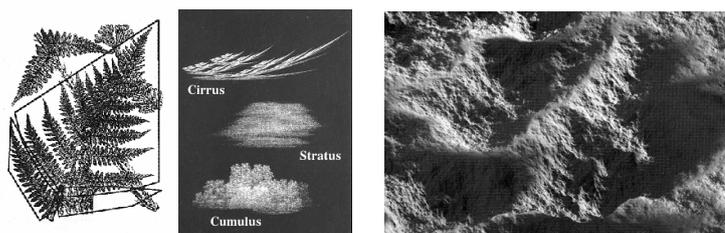


ภาพที่ 3 Mandelbrot set และลักษณะความคล้ายตัวเอง

นอกจาก Mandelbrot set แล้วยังมีสมการทางคณิตศาสตร์อีกมากมายนับไม่ถ้วนที่แสดงลักษณะของแฟร็กทัล เช่น ขั้นตอนวิธีการทำซ้ำเพื่อคำนวณรากของสมการที่รู้จักกันคืออย่างระเบียบวิธีของนิวตัน (Newton's method) เองก็มีลักษณะของแฟร็กทัลแอบแฝงอยู่ (Falconer, 1995) นอกจากนี้ยังมีแฟร็กทัลอีกประเภทหนึ่งซึ่งมีบทบาทต่องานทางด้านการประมวลผลภาพ (image compression) (Barnsley and Hurd, 1992) คือแฟร็กทัลที่เป็นลิมิตของไอเอฟเอส (IFS : Iterated Function System) ซึ่งเชื่อกันว่าเป็นลักษณะที่พบได้ทั่วไปในธรรมชาติ



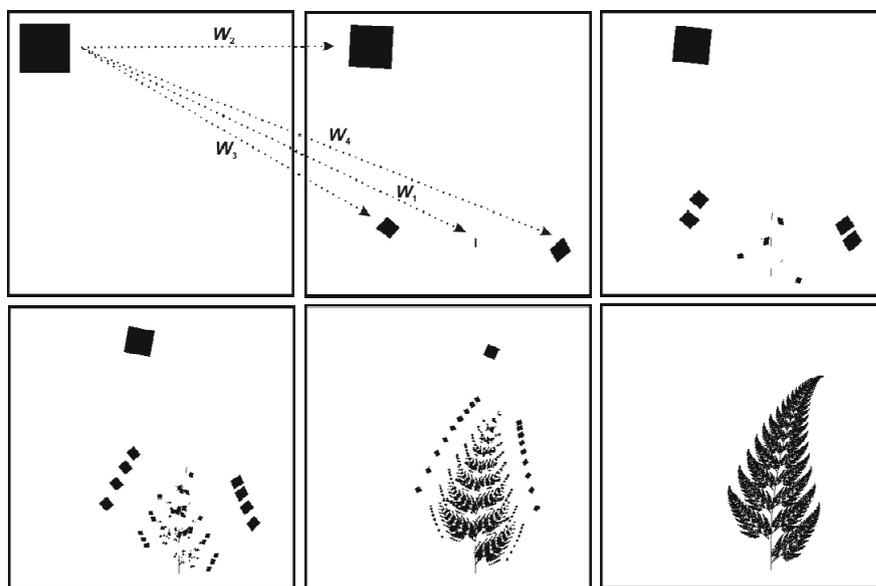
ภาพที่ 4 ตัวอย่างของไอเอฟเอส ซ้าย: middle-thirds Cantor set และ ขวา: Koch curve



ภาพที่ 5 แสดงลักษณะของความคล้ายตัวเองในของวัตถุในธรรมชาติ

1.5 ไอเอฟเอส (IFS: Iterated Function System) คณิตศาสตร์ของการบีบอัดแบบแฟร็กทัล

ไอเอฟเอส เป็นทฤษฎีทางคณิตศาสตร์ที่ได้รับการพัฒนาขึ้นในช่วงต้นทศวรรษที่ 80 โดย Michael Barnsley นักคณิตศาสตร์จากมหาวิทยาลัย Georgia Institute of Technology โดยนิยาม ไอเอฟเอส คือระบบที่ประกอบด้วยเมตริกสเปซ (ในกรณีที่เรากำลังสนใจอยู่คือ \mathbb{R}^2) และชุดของ ฟังก์ชันที่เรียกว่าคอนแทรกชัน (contraction map) ซึ่งก็คือฟังก์ชันที่มีคุณสมบัติในการส่งจุดสองจุดใดๆ ไปเป็นสองจุดที่มีระยะห่างระหว่างกันน้อยลงกว่าเดิม คุณสมบัติอันหนึ่งที่สำคัญของไอเอฟเอสบน \mathbb{R}^2 ก็คือ เมื่อทำการเลือกสับเซตปิดใดๆ มาสร้างเป็นลำดับอนันต์ โดยแมพเซตผ่านชุดของฟังก์ชันในไอเอฟเอสเข้าไปเรื่อยๆ ลำดับนี้จะลู่เข้าหาลิมิตเสมอ และลิมิตจะไม่ขึ้นกับการเลือกสับเซตปิดเริ่มต้น ลิมิตของลำดับที่สร้างโดยไอเอฟเอสนี้ จะเรียกว่าแอตแทรกเตอร์ของไอเอฟเอส (Attractor of IFS) และเมื่อกำหนดไอเอฟเอสใดๆมาให้ เราจะสามารถสร้างภาพที่เป็นแอตแทรกเตอร์ของไอเอฟเอสได้เสมอ (Barnsley, 1988a) ในทางกลับกัน หากกำหนดภาพมาให้ และเราสามารถหาไอเอฟเอสที่มีลิมิตใกล้เคียงภาพนั้นในระดับที่ยอมรับได้ เราก็จะสามารถใช้ค่าพารามิเตอร์ของไอเอฟเอสนั้นแทนข้อมูลภาพได้เสมือนการบีบอัดข้อมูลภาพในรูปแบบหนึ่ง



$$w_i \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_i & b_i \\ c_i & d_i \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \end{bmatrix}$$

i	a_i	b_i	c_i	d_i	e_i	f_i
1	0.00	0.00	0.00	0.16	0.00	0.00
2	0.85	0.04	-0.04	0.85	0.00	1.60
3	0.20	-0.26	0.23	0.22	0.00	1.60
4	-0.15	0.28	0.26	0.24	0.00	0.44

ภาพที่ 6 ภาพใบเฟิร์นสามารถสร้างจากตัวเลขเพียง 24 ตัว ในรูปของฟังก์ชันเชิงเส้น 4 ฟังก์ชัน

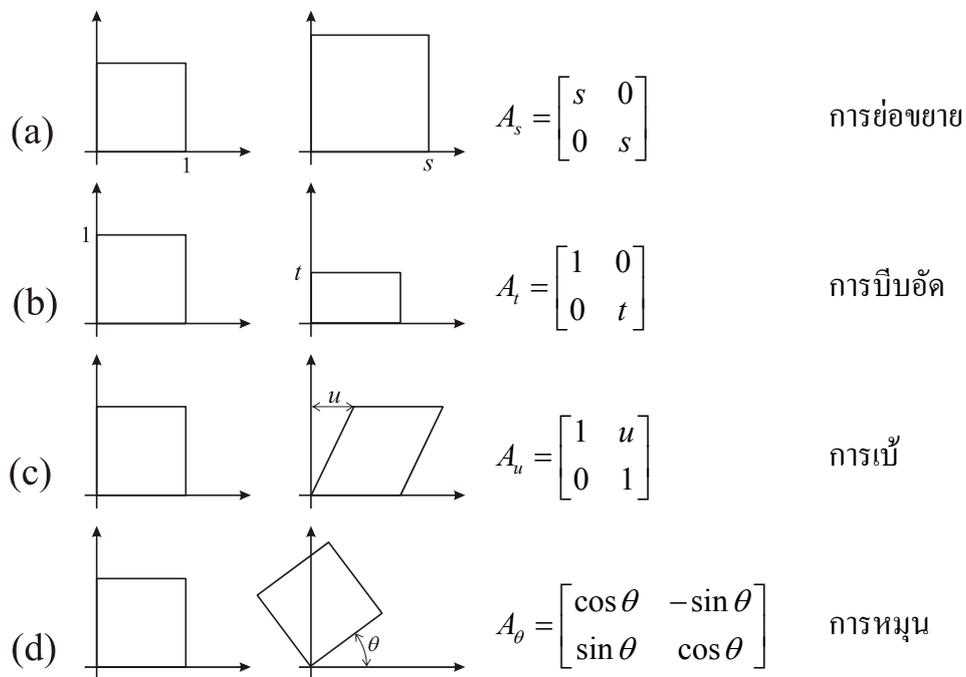
ในการนิยามไอเอฟเอส โดยทั่วไปจะอาศัยฟังก์ชันที่อยู่ในรูปของการย่อขยายขนาด, การบีบ, การเบ้, การหมุน และการเลื่อนตำแหน่ง ฟังก์ชันเชิงเส้นในลักษณะนี้เรียกว่า การแปลงสัมพรรค (affine transformation)

บทนิยาม การแปลงสัมพรรคบน \mathbb{R}^n (affine transformation on \mathbb{R}^n)

พิจารณากรณีของ \mathbb{R}^2 การแปลงสัมพรรคสามารถเขียนได้ในรูปของ

$$w = Ax + B$$

โดยที่ x และ w เป็นเวกเตอร์สองมิติแทนพิกัดจุด, B เป็นเมทริกซ์ขนาด 2×1 บ่งบอกการเลื่อนตำแหน่ง และ A เป็นเมทริกซ์ขนาด 2×2 ที่เป็นผลคูณของเมทริกซ์ต่างๆ ในรูปแบบต่อไปนี้



ภาพที่ 7 การดำเนินการพื้นฐานของการแปลงสัมพรรค (affine transformation)

เห็นได้ชัดว่านิยามนี้สามารถขยายออกไปเป็นนิยามบน \mathbb{R}^n เมื่อ n มีค่าตั้งแต่ 2 ขึ้นไปได้โดยง่าย โดยการขยายขนาดของเมทริกซ์ A เป็น $n \times n$ และกำหนดค่าแต่ละส่วนประกอบของเมทริกซ์ในลักษณะของผลคูณของเมทริกซ์ของการแปลงในแต่ละระนาบ ในการนิยามไอเอฟเอส นอกจากการเป็นการแปลงสัมพรรคแล้ว ฟังก์ชันยังจะต้องสอดคล้องเงื่อนไขที่สำคัญอันหนึ่งที่เรียกว่า คอนแทรคชัน (contraction) ทั้งนี้เพื่อรับประกันการมีอยู่จริงของลิมิตของไอเอฟเอส ซึ่งจำเป็นอย่างยิ่งต่อขั้นตอนวิธีของการบีบอัดภาพแบบแฟร็กทัล

บทนิยาม คอนแทรกชันบน \mathbb{R}^n (contraction on \mathbb{R}^n)

ในกรณีของ \mathbb{R} เราจะกล่าวว่า $f: \mathbb{R} \rightarrow \mathbb{R}$ เป็นคอนแทรกชันบน \mathbb{R} ก็ต่อเมื่อ มีจำนวนจริง s โดยที่ $0 < s < 1$ ทำให้ $|f(x) - f(y)| \leq s|x - y|$ สำหรับทุก $x, y \in \mathbb{R}$ ค่า s ดังกล่าว เรียกว่า แฟกเตอร์การหดตัว (contractivity factor) ของ f

ในกรณีของ \mathbb{R}^n ก็สามารถนิยามได้ในลักษณะคล้ายๆกัน กล่าวคือ $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$ จะเป็น คอนแทรกเตอร์บน \mathbb{R}^n ก็ต่อเมื่อ มีจำนวนจริง s โดยที่ $0 < s < 1$ ทำให้

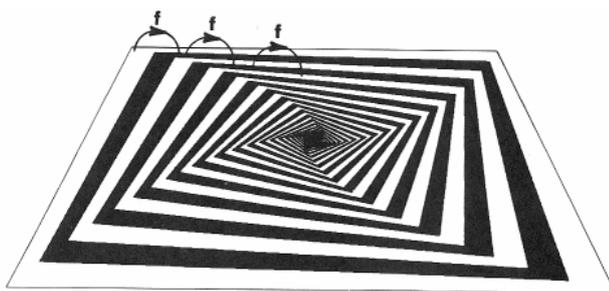
$$\sqrt{\sum_{i=1}^n (f(x_i) - f(y_i))^2} \leq s \cdot \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \text{ สำหรับทุก } (x_1, \dots, x_n), (y_1, \dots, y_n) \in \mathbb{R}^n$$

อาจกล่าวแบบง่ายๆ ได้ว่า คอนแทรกชันก็คือฟังก์ชันที่ส่งจุดสองจุดใดๆ ไปเป็นสองจุดใหม่ที่มีระยะทางใกล้กันกว่าเดิม ทฤษฎีบทต่อไปเป็นการยืนยันว่าหากนำคอนแทรกชันไปแมพจุดใดซ้ำเรื่อยๆ ลำดับของค่าดังกล่าวจะลู่เข้าสู่จุดลิมิตเสมอ และจุดลิมิตนั้นไม่ขึ้นอยู่กับทางเลือกจุดตั้งต้น

ทฤษฎีบท Banach Fixed Point Theorem บน \mathbb{R}^n

ให้ f เป็นคอนแทรกชันบน \mathbb{R}^n แล้ว

1. f จะมีจุดตรึง (fixed point) และมีเพียงจุดเดียวเท่านั้น
2. สำหรับ $a \in \mathbb{R}^n$ ใดๆ ลำดับ $f(a), f^2(a), f^3(a), \dots$ มีลิมิต และลิมิตคือจุดตรึงของ f



ภาพที่ 8 ตัวอย่างการลู่เข้าสู่จุดตรึงของคอนแทรกชันบน \mathbb{R}^2

ในความเป็นจริง เราอาจจะมีคอนแทรกชันมากกว่าหนึ่งตัวหลอมรวมกันเป็นฟังก์ชันอันหนึ่ง เรียกระบบลักษณะนี้ว่า ไอเอฟเอส (IFS: Iterated Function System)

บทนิยาม ไอเอฟเอสบน \mathbb{R}^n

IFS บน \mathbb{R}^n ประกอบด้วยสเปซ \mathbb{R}^n และเซตจำกัดของคอนแทรกชันบน \mathbb{R}^n ทั้งหมดเขียนอยู่ในรูป $\{\mathbb{R}^n; w_1, \dots, w_n\}$



ภาพที่ 9 ตัวอย่างไอเอฟเอสที่ประกอบด้วยคอนแทรกชัน T_1, T_2, T_3

กำหนดให้ $\{\mathbb{R}^n; w_1, \dots, w_n\}$ เป็น IFS บน \mathbb{R}^n และ $H(\mathbb{R}^n)$ เป็นสเปซของของสับเซตปิดมีขอบเขตของ \mathbb{R}^n นิยามฟังก์ชัน $W: H(\mathbb{R}^n) \rightarrow H(\mathbb{R}^n)$ ดังนี้

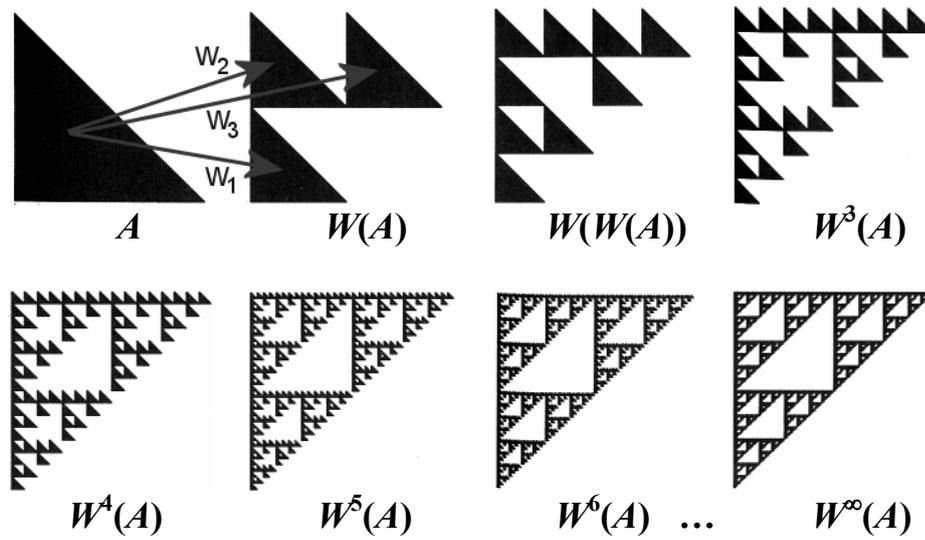
$$W(B) = \bigcup_{i=1}^n w_i(B) \text{ สำหรับทุก } B \in H(\mathbb{R}^n)$$

และด้วยนิยามนี้เอง จึงสามารถใช้ฟังก์ชัน W ในการทำซ้ำ เมื่อกำหนดให้ A เป็นเซตใดๆใน $H(\mathbb{R}^n)$ เซตที่ลำดับ $A, W(A), W^2(A), W^3(A), \dots$ ลู่เข้าหา จะเรียกว่าแอตแทรกเตอร์ (attractor) ของ ไอเอฟเอส

ตัวอย่าง ไอเอฟเอสชื่อ Sierpinski Triangle $\{\mathbb{R}^2; w_1, w_2, w_3\}$ โดยที่

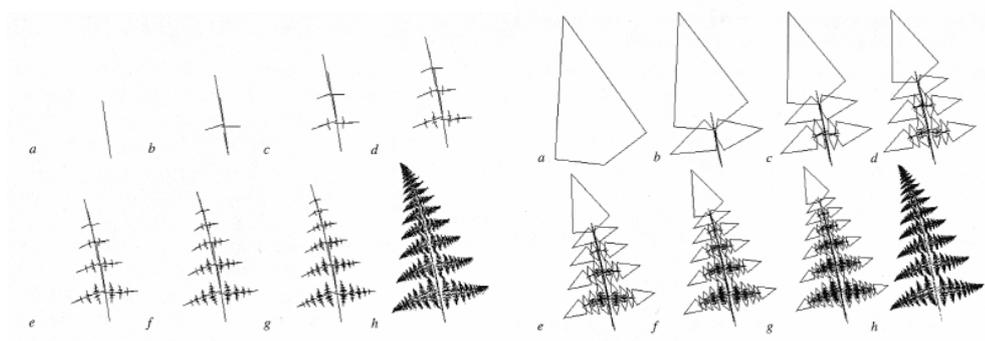
$$\begin{aligned} w_1 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} &= \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ w_2 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} &= \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 50 \end{bmatrix} \\ w_3 \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} &= \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 50 \\ 50 \end{bmatrix} \end{aligned}$$

ทั้ง w_1, w_2 และ w_3 เป็นฟังก์ชันที่ย่อขนาด 50% และเลื่อนตำแหน่ง ทั้งหมดเป็นคอนแทรกชันบน \mathbb{R}^2 ที่มีแฟกเตอร์การหดตัวเท่ากับ 0.5 เมื่อนำฟังก์ชันชุดไปแมพเซตเริ่มต้นอันหนึ่งซ้ำไปเรื่อยๆ จะได้ลำดับของผลลัพธ์ดังภาพที่ 10



ภาพที่ 10 ตัวอย่างแฟร็กทัล Sierpinski Triangle

พิจารณาตัวอย่างของไอเอฟเอสที่มีแอทแทรกเตอร์เป็นภาพไบเฟิร์น ภาพในแต่ละขั้นตอน เกิดจากการแมพรูปในขั้นตอนที่ผ่านมาด้วยการแปลงสัมพรรคจำนวน 4 ตัว สังเกตว่าลักษณะของภาพเริ่มต้นจะไม่มีผลต่อแอทแทรกเตอร์ของไอเอฟเอส (ภาพ h) โดยทฤษฎีแล้ว ไม่ว่าจะเริ่มต้นด้วยรูปทรงลักษณะเช่นใดก็ตาม ท้ายที่สุดแล้วเราจะได้ภาพเดียวกันเสมอ คุณสมบัตินี้ถูกพิสูจน์อย่างเป็นระบบโดยทฤษฎีทางคณิตศาสตร์ใน Barnsley (1988a)



ภาพที่ 11 ตัวอย่างของการใช้เซตเริ่มต้นที่ต่างกัน แต่ลำดับคู่เข้าสู่แอทแทรกเตอร์เดียวกัน

1.6 การบีบอัดภาพแบบแฟร็กทัล (Fractal Image Compression)

แนวคิดในการนำทฤษฎีของไอเอฟเอสมาประยุกต์ใช้ในการบีบอัดข้อมูลภาพ ถูกพัฒนาและเผยแพร่เป็นครั้งแรกโดย Barnsley (1988b) ด้วยสมมติฐานที่ว่าภาพถ่ายโดยทั่วไปจะมีคุณสมบัติสำคัญของแฟร็กทัลอันหนึ่งที่เรียกว่า ความคล้ายตัวเอง (self-similarity) ในการทำการบีบอัดแบบแฟร็กทัลข้อมูลภาพจะถูกเข้ารหัสในรูปของฟังก์ชันที่นิยามขึ้นจากความคล้ายของแต่ละส่วนของภาพเท่านั้น โดยที่ไม่จำเป็นต้องเก็บข้อมูลจุดภาพเลยแม้แต่จุดเดียว ข้อมูลที่ต้องจัดเก็บจึงมีปริมาณน้อยกว่าเดิมมาก ด้วยเหตุนี้จึงสามารถนำมาใช้เป็นขั้นตอนวิธีการบีบอัดภาพได้

ขั้นตอนการบีบอัดข้อมูลมีพื้นฐานอยู่บนทฤษฎีบทที่มีชื่อว่า ทฤษฎีบทคอลลาจ (Collage Theorem) (Barnsley, 1998) ซึ่งเป็นหลักการพื้นฐานของการบีบอัดภาพแบบแฟร็กทัลแทบทุกชนิดในปัจจุบัน

ทฤษฎีบท ทฤษฎีบทคอลลาจ (Collage Theorem)

กำหนดให้ (X, d) เป็นคอมพลีทเมตริกสเปซ (complete metric space) และให้ $\{X; w_1, w_2, \dots, w_n\}$ เป็นคอนแทรกชันที่มีแฟกเตอร์การหดตัว (contractivity factor) เท่ากับ s และให้ K เป็นเซตใดๆในสเปซ $(H(X), h)$ ของคอมแพคต์เซตทั้งหมดของ X ที่มีเมตริกเป็นฮอสดอร์ฟเมตริก (Hausdorff metric) ซึ่งสอดคล้องเงื่อนไข

$$h(K, W(K)) \leq \varepsilon$$

แล้วจะได้ว่า

$$h(K, A) \leq \frac{\varepsilon}{1-s}$$

เมื่อ A เป็นแอตแทรกเตอร์ของไอเอฟเอส $\{X; w_1, w_2, \dots, w_n\}$

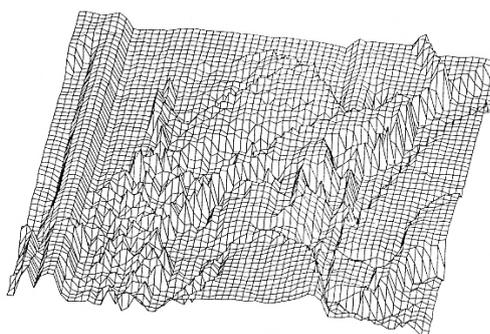
กล่าวอย่างง่ายก็คือ ทฤษฎีบทคอลลาจบอกให้เราทราบว่า เมื่อนำภาพใดภาพหนึ่งมาผ่านการแมพด้วยไอเอฟเอสหนึ่งครั้ง แล้วได้ผลออกมาเป็นภาพใหม่ที่มีความคล้ายภาพต้นฉบับเดิมแล้วแอตแทรกเตอร์ของไอเอฟเอสนั้น ก็จะคล้ายภาพต้นฉบับด้วย ดังนั้นทฤษฎีบทนี้ จึงเปิดโอกาสให้เราสามารถหาไอเอฟเอสได้ง่ายขึ้น โดยการทดลองแมพภาพต้นฉบับด้วยไอเอฟเอสที่ต้องการจะทดสอบเป็นจำนวนเพียงหนึ่งครั้ง หากผลออกมาได้ภาพใหม่ที่มีค่าความผิดเพี้ยนไปจากภาพเดิมฉบับไม่เกิน ε เราก็จะสามารถนำไอเอฟเอสนั้นไปสร้างเป็นภาพด้วยกระบวนการทำซ้ำ (ซึ่งจะกล่าวต่อไป) และจะได้ภาพที่มีความผิดเพี้ยนจากต้นฉบับไม่เกิน $\frac{\varepsilon}{1-s}$ ซึ่งหากเรากำหนดค่า ε ที่

เหมาะสมตั้งแต่แรก ค่าความผิดพลาดในท้ายที่สุดก็จะอยู่ในขอบเขตที่ยอมรับได้ เราก็จะสามารถใช้พารามิเตอร์ของไอเอฟเอสอันนี้เพื่อแทนข้อมูลภาพได้ทันที

ทฤษฎีของ Barnsley สามารถประยุกต์ใช้งานได้จริง แต่ก็ยังไม่เพียงพอที่จะใช้เข้ารหัสภาพแบบอัตโนมัติได้ ตัวอย่างของการบีบอัดข้อมูลด้วยวิธีแฟรactalในยุคแรกๆ จึงเป็นการหาพารามิเตอร์เองด้วยมือ ซึ่งถึงแม้จะได้อัตราส่วนสูงถึงระดับ 1:10000 แต่ก็ต้องใช้เวลานานมาก จนกระทั่ง Jacquin ได้เสนอแนวคิดใหม่ในการใช้ ไอเอฟเอสแบบเฉพาะที่ (Local IFS หรือ Partitioned IFS (PIFS)) (Jacquin, 1992) กล่าวคือ แทนที่จะพยายามค้นหาไอเอฟเอสที่จะใช้แทนภาพทั้งภาพ กลับใช้วิธีแบ่งภาพออกเป็นบล็อกเล็กๆ และค้นหาฟังก์ชันที่เหมาะสมของแต่ละบล็อกแทน เมื่อเสร็จสิ้นกระบวนการเข้ารหัส ก็จะได้ฟังก์ชันจำนวนหนึ่ง โดยฟังก์ชันแต่ละตัวจะมีโดเมนและเรนจ์เป็นของตัวเองที่แตกต่างกันออกไป ซึ่งก็คือส่วนต่างๆ ในภาพนั่นเอง ด้วยแนวคิดนี้ จึงเป็นไปได้ที่จะ โปรแกรมให้คอมพิวเตอร์แบ่งพาร์ติชันเล็กกลงไปเรื่อยๆ จนกว่าจะพบฟังก์ชันสำหรับแต่ละเรนจ์บล็อก ที่ให้ความผิดพลาดน้อยในระดับที่ยอมรับได้

1.7 ขั้นตอนวิธีต้นแบบในการเข้ารหัสภาพ

ภาพระดับสีเทาสามารถถูกมองเป็นพื้นผิว 3 มิติ โดยที่ระนาบ x-y แทนระนาบของภาพ และค่าความสูงทางแกน z คือ ค่าของระดับสีเทา

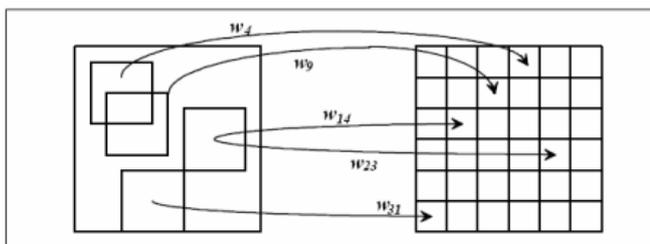


ภาพที่ 12 แสดงพื้นผิวสามมิติที่ได้จากภาพระดับสีเทา

การเข้ารหัสแบบแฟร็กทัลของภาพนี้อาจมองได้ว่าเป็นการพยายามหาชิ้นส่วนบริเวณในภาพสองบริเวณ ซึ่งคล้ายกันในแง่ที่สามารถปรับเปลี่ยนทางเรขาคณิต เช่น หมุน, ย่อขนาด และเลื่อนตำแหน่ง (ทั้งทางแกน x, y, z) เพื่อให้พื้นผิวแนบสนิทกันมากที่สุดเท่าที่จะมากได้ เป้าหมายที่

ต้องการก็คือ สำหรับทุกส่วนของภาพ จะต้องสามารถหาบริเวณในส่วนอื่นของภาพซึ่งมีขนาดใหญ่กว่า ที่มีความคล้ายคลึงกับบริเวณดังกล่าว

ขั้นตอนการเข้ารหัสจะเริ่มจาก นำภาพต้นฉบับมาแตกเป็นพาร์ติชัน แต่ละพาร์ติชันเรียกว่าเรนจ์บล็อกซึ่งเป็นสี่เหลี่ยมจัตุรัสที่ไม่ซ้อนทับกัน แต่ละจุดของภาพจะเป็นสมาชิกของพาร์ติชันเพียงหนึ่งพาร์ติชันเท่านั้น จากนั้นทำการค้นหาบริเวณใดๆ ในภาพต้นฉบับ ที่มีขนาดใหญ่กว่าเรนจ์บล็อก ซึ่งมีความคล้ายมากพอที่จะสามารถนำมาใช้เป็น โดเมนของเรนจ์บล็อกนั้นได้ การพิจารณาความคล้าย จะทำการวัดด้วยฟังก์ชันวัดความแตกต่างระหว่างเรนจ์บล็อก กับส่วนของภาพดั้งเดิม ที่ผ่านการแปลงและลดขนาดให้เท่ากับเรนจ์บล็อกแล้ว ทั้งนี้เพื่อความสะดวกของการค้นหา รูปแบบของฟังก์ชันที่ใช้นิยม ไอเอฟเอสจะถูกจำกัดอยู่ในรูปของการผสมระหว่าง การย่อขนาดลงครึ่งหนึ่ง, การกลับซ้าย-ขวา และการหมุนในแนว 0, 90, 180 และ 270 องศา ดังนั้นจึงมี การแปลงเพียง 8 รูปแบบที่ต้องทดสอบ และใช้วิธีปรับความเปรียบต่างของค่าสี และความสว่างร่วมด้วย (ซึ่งก็คือการย่อขยาย และ การเลื่อนตำแหน่งทางแกน z) ขั้นตอนวิธีจะทำการเลือกบริเวณที่เมื่อผ่านการปรับแต่งดังกล่าวแล้วให้ผลใกล้เคียงเรนจ์บล็อกนั้นมากที่สุด และความผิดพลาดอยู่ในระดับที่ยอมรับได้ เพื่อนำไปใช้เป็น โดเมนของเรนจ์บล็อกนั้นในขั้นตอนของการถอดรหัส



ภาพที่ 13 แสดงตัวอย่างการจับคู่โดเมนและเรนจ์บล็อก

เมื่อเสร็จสิ้นกระบวนการค้นหาสำหรับแต่ละ เรนจ์บล็อก $R_{i,j}$ เราจะได้ฟังก์ชันในรูปแบบ

$$w_{i,j} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ 0 & 0 & p_i \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \\ q_i \end{bmatrix}$$

โดยที่ $a_i, b_i, c_i, d_i, e_i, f_i$ จะบอกถึงลักษณะของการหมุน การเปลี่ยนขนาด และการเลื่อนตำแหน่งในระนาบภาพ ส่วน p_i คือการปรับค่าความเปรียบต่าง และ q_i คือการปรับค่าความสว่าง

ในทางปฏิบัติ นิยมใช้รูทมีนสแควร์ (RMS) เป็นตัววัดความแตกต่างของบล็อกในขั้นตอนของการเข้ารหัส สูตรคำนวณค่าความแตกต่างระหว่างบล็อกขนาด $m \times n$ คือ

$$d_{RMS}(D, R) = \sqrt{\sum_{i=1}^m \sum_{j=1}^n (d_{i,j} - r_{i,j})^2}$$

เมื่อแทนบล็อก D และ R ในรูปของเวกเตอร์ d และ r ที่มีมิติเท่ากับ mn ปัญหานี้จึงอาจมองเป็นปัญหาของการคำนวณค่าของ

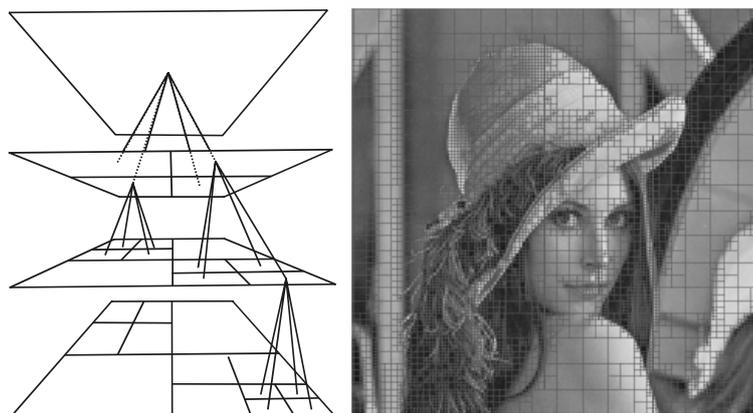
$$E(d, r) = \min_{a, b \in \mathbb{R}} \|r - (ae + bd)\|^2 \quad \text{โดยที่ } e = \underbrace{(1, \dots, 1)}_{mn \text{ components}}$$

จากสมการข้างต้น สูตรสำหรับคำนวณค่าความเปรียบต่างและความสว่าง ที่ให้ค่าผลต่างรูทมีนสแควร์ต่ำที่สุดคือ

$$\text{พารามิเตอร์ความสว่างที่ดีที่สุด} \quad a = \frac{\langle d, r \rangle \langle e, d \rangle - \langle d, d \rangle \langle e, r \rangle}{\langle e, d \rangle^2 - \langle d, d \rangle}$$

$$\text{พารามิเตอร์ความเปรียบต่างที่ดีที่สุด} \quad b = \frac{\langle e, d \rangle \langle e, r \rangle - \langle d, r \rangle}{\langle e, d \rangle^2 - \langle d, d \rangle}$$

โดยที่ $\langle _, _ \rangle$ แทนผลคูณจุด (dot product) ในกระบวนการค้นหาสำหรับแต่ละเรนจ์บล็อก หากไม่พบโดเมนที่ให้ค่าความผิดพลาดในระดับที่ยอมรับได้ ก็จะทำการแบ่งเรนจ์บล็อกนั้นออกเป็น 4 บล็อกเล็กๆ และทำการค้นหาโดเมนสำหรับบล็อกย่อยทั้ง 4 ซ้ำเช่นนี้จนกระทั่งพบโดเมนบล็อกที่เหมาะสมสำหรับทุกๆ บล็อกของพาร์ติชัน วิธีการแบ่งพาร์ติชันแบบนี้เรียกว่า การแบ่งพาร์ติชันแบบควอดทรี (Quadtree Partitioning)

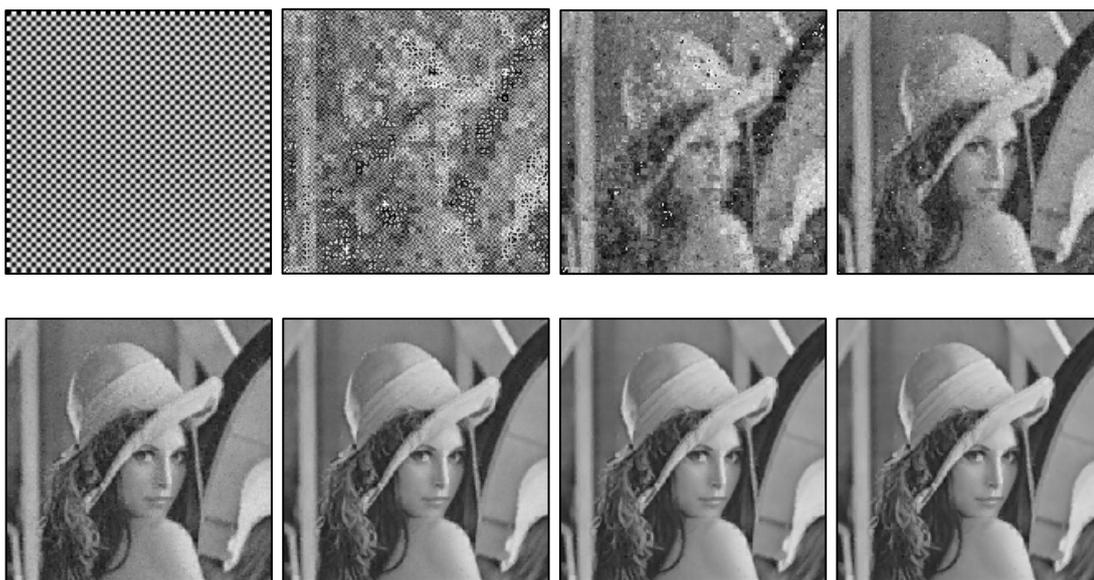


ภาพที่ 14 ตัวอย่างการแบ่งพาร์ติชันแบบควอดทรี

เมื่อได้ฟังก์ชันสำหรับทุกเรนจ์บล็อกครบแล้ว จะทำการจัดเก็บเฉพาะค่าพารามิเตอร์ของแต่ละฟังก์ชันและค่าพิกัดของโดเมนของแต่ละฟังก์ชัน ทั้งนี้ไม่มีความจำเป็นต้องเก็บข้อมูลส่วนหนึ่งส่วนใดของภาพเลย และเพื่อที่จะให้ได้อัตราส่วนของการบีบอัดที่สูงขึ้น จึงจำเป็นต้องอาศัยเทคนิคพื้นฐานในการลดขนาดข้อมูลแบบไม่มีการสูญเสียร่วมด้วย เช่น การเข้ารหัสแบบรันเลนท (Run-Length Encoding) และ การเข้ารหัสแบบฮัฟแมน (Huffman Coding) แบบเดียวกับที่ใช้ในไฟล์ภาพแบบเจเป็็ก

1.8 ขั้นตอนวิธีในการถอดรหัสภาพ

ขั้นตอนการเข้ารหัสเริ่มจากการถอดรหัสข้อมูลเพื่อเตรียมฟังก์ชัน และสร้างภาพเริ่มต้นเตรียมไว้ ซึ่งจะเป็นภาพอะไรก็ได้ เพราะโดยทฤษฎีแล้วภาพเริ่มต้นจะไม่มีผลต่อภาพสุดท้าย ดังที่กล่าวไว้ในหัวข้อ 1.5 ดังนั้นภาพเริ่มต้นอาจจะใช้ภาพสีขาว หรือสีดำทั้งภาพก็ได้ จากนั้นทำการอ่านค่าบริเวณที่เป็นโดเมนของฟังก์ชัน หมุน, ปรับขนาด, ปรับความเปรียบต่าง และความสว่างตามที่นิยามไว้ในฟังก์ชัน และนำไปวาดทับบริเวณที่เป็นเรนจ์ของฟังก์ชันนั้น ทำเช่นนี้กับทุกฟังก์ชันจนครบ ก็จะได้ภาพในขั้นที่ 1 นำภาพขั้นที่ 1 นี้ไปเป็นอินพุทของกระบวนการเดิม กับฟังก์ชันชุดเดิม ก็จะได้ภาพขั้นที่ 2 เมื่อทำซ้ำเช่นนี้ไปเรื่อยๆ ภาพที่ถูกเข้ารหัสไว้ก็จะยิ่งปรากฏชัดเจนขึ้น ทำซ้ำจนกระทั่งภาพใหม่ที่ได้ในขั้นตอนปัจจุบัน กับภาพเก่าในขั้นตอนก่อนหน้า มีค่าของผลต่างรูทมีนสแควร์ น้อยกว่าค่าๆหนึ่งตามแต่กำหนด



ภาพที่ 15 ตัวอย่างภาพในแต่ละขั้นตอนขณะทำการถอดรหัส

อย่างไรก็ตาม ถึงแม้ว่าภาพเริ่มต้นจะไม่มีผลต่อภาพสุดท้าย แต่ภาพเริ่มต้นจะมีผลต่ออัตราการใช้พื้นที่ หากมีการเลือกใช้ภาพเริ่มต้นที่เหมาะสม ก็จะสามารถลดจำนวนรอบในขั้นตอนของการถอดรหัสให้น้อยลงได้ (Moon, Kim and Kim, 2000)

2. ขั้นตอนวิธีพื้นฐานที่ใช้ในงานวิจัย

2.1 การแปลงเชิงเรขาคณิตบนภาพระดับสีเทา

เนื่องจากในงานวิจัยชิ้นนี้มีการสร้างฟังก์ชันในการหมุนภาพในมุมที่ละเอียด เพื่อที่จะให้ภาพที่ผ่านการหมุนมีคุณภาพดีใกล้เคียงภาพเดิม จึงจำเป็นต้องอาศัยขั้นตอนวิธีในการทำการประมาณค่าในช่วง (interpolation) ของการแปลงเชิงเรขาคณิตบนภาพบิตแมต

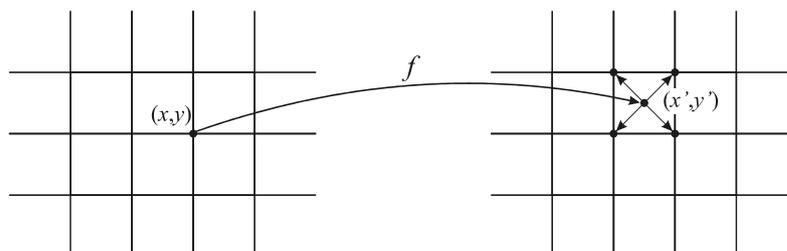
การแปลงเชิงเรขาคณิตบนภาพบิตแมต เป็นการเปลี่ยนความสัมพันธ์เชิงตำแหน่งระหว่างจุดต่างๆในภาพ โดยการระบุความสัมพันธ์ระหว่างตำแหน่งเก่าและตำแหน่งใหม่ของจุดบนภาพในรูปของฟังก์ชัน

$$(x', y') = f(x, y)$$

ในกรณีของภาพบิตแมตซึ่งมีลักษณะเป็นพื้นผิวที่ไม่ต่อเนื่องนั้น เราไม่อาจสามารถเลื่อนแต่ละจุดภาพไปวางในตำแหน่งใหม่โดยตรงได้ เนื่องจากพิกัดของเอาพุตถูกจำกัดให้เป็นจำนวนเต็ม และการประมาณพิกัดที่เป็นเศษทศนิยมให้เป็นจำนวนเต็มก็อาจทำให้เกิดการเหลื่อมล้ำของตำแหน่งจุดและก่อให้เกิดลักษณะที่ไม่พึงประสงค์บนภาพได้ การดำเนินการแปลงบนภาพจึงต้องอาศัยเทคนิคหนึ่งในสองวิธีนี้

2.1.1 การส่งไปข้างหน้า (forward-mapping หรือ pixel carry-over)

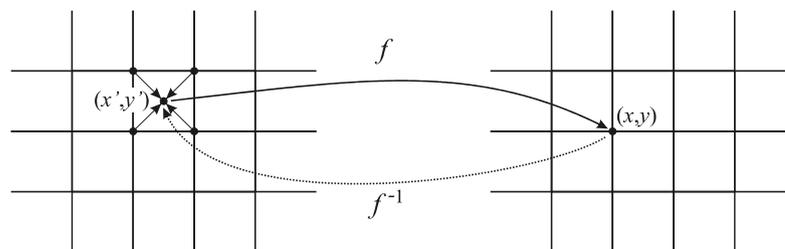
ใช้วิธีการส่งจุดผ่านฟังก์ชันไปยังผลลัพธ์ ในกรณีที่พิกัดที่ถูกส่งไปตกอยู่ระหว่างพิกเซล ก็จะทำการแบ่งค่าของระดับสีเทาของจุดที่ถูกส่งไปนั้นให้กับพิกเซลที่รายล้อมอยู่ โดยอาจจะเฉลี่ยตามระยะทางใกล้เคียงจากพิกเซลเหล่านั้น หรือจะโดยวิธีใดก็ตาม วิธีนี้มีข้อเสียคือมีการคำนวณเป็นจำนวนมาก และอาจมีการคำนวณที่สูญเปล่าเนื่องจากบางจุดในภาพอินพุตอาจจะถูกส่งออกไปนอกกรอบของภาพเอาพุต นอกจากนี้วิธีนี้อาจไม่เหมาะสมสำหรับฟังก์ชันการแปลงในบางรูปแบบ เช่นการขยายภาพ เพราะบางจุดในผลลัพธ์อาจจะถูกเว้นว่างไว้ เนื่องจากไม่มีจุดถูกส่งมาลงในตำแหน่งใกล้เคียงเลขค่าของระดับสีเทาจึงกระจายมาไม่ถึง



ภาพที่ 16 แสดงการส่งพิกเซลด้วยขั้นตอนวิธีการส่งไปข้างหน้า

2.1.2 การส่งย้อนกลับ (backward-mapping หรือ pixel-filling)

วิธีนี้เป็นวิธีที่นิยมใช้กันมากกว่า หลักการก็คือ แทนที่จะอ้างอิงแต่ละพิกเซลของภาพอินพุตเป็นหลัก เราจะสนใจแต่ละพิกเซลในเอาพุตแทน และอาศัยอินเวอร์สของฟังก์ชันในการคำนวณว่าพิกเซลใดในอินพุตที่จะถูกส่งมาลงยังจุดในเอาพุตเหล่านี้ ซึ่งพิกัดที่คำนวณได้อาจเป็นเลขทศนิยม จึงต้องอาศัยข้อมูลจากจุดรอบๆ ในการคำนวณหาค่าประมาณที่ควรจะเป็น ซึ่งเรียกว่าการทำการประมาณค่าในช่วงของระดับสีเทา (gray-level interpolation) วิธีนี้มีข้อดีคือ การคำนวณจะเกิดเท่าที่จำเป็นเท่านั้น คุณภาพของภาพเอาพุตจะขึ้นอยู่กับลักษณะของการประมาณค่าในช่วงที่เลือกใช้



ภาพที่ 17 แสดงการส่งพิกเซลด้วยขั้นตอนวิธีการส่งย้อนกลับ

ขั้นตอนวิธีในการประมาณค่าของฟังก์ชันบนจุดที่พิกัดไม่เป็นจำนวนเต็มนี้มีอยู่หลายวิธี แต่ที่นิยมใช้กันโดยทั่วไปมี 3 วิธี คือ

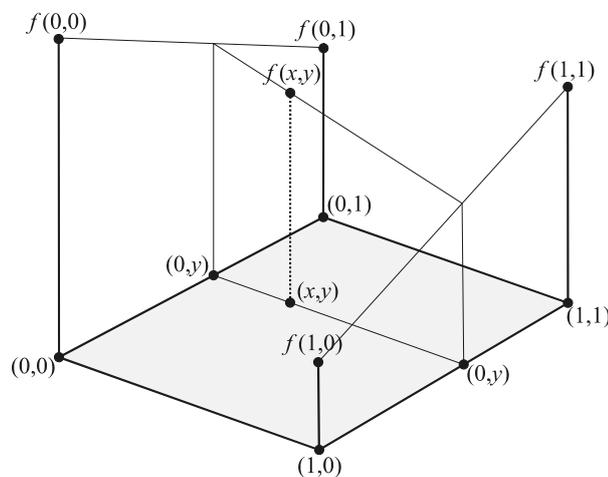
1) การประมาณค่าในช่วงแบบเพื่อนบ้านใกล้เคียงที่สุด (Nearest Neighbor Interpolation)

หลักการคือ ประมาณค่าโดยหิบบค่ามาจากจุดที่มีพิกัดเป็นจำนวนเต็มที่ใกล้เคียงที่สุด วิธีนี้เป็นวิธีที่ง่ายและรวดเร็วมาก แต่ภาพที่ได้มักจะมีคุณภาพที่ไม่ดีนัก เนื่องจากฟังก์ชันที่ได้จากการทำ การประมาณค่าในช่วงนั้นมีลักษณะเป็นขั้นบันไดที่ไม่ต่อเนื่อง ทำให้ปรากฏรอยหยัก (aliasing) ชัดเจนโดยเฉพาะบริเวณขอบของวัตถุในภาพ

2) การประมาณค่าในช่วงแบบไบลิเนียร์ (Bilinear Interpolation)

การประมาณค่าในช่วงแบบไบลิเนียร์ จะเป็นการเฉลี่ยค่าฟังก์ชันจากพิกัดที่เป็นจำนวนเต็มที่อยู่รอบๆ โดยอาศัยการประมาณแบบเชิงเส้นในสองแกน ยกตัวอย่างในกรณีที่ $0 \leq x, y \leq 1$ ดังภาพ การประมาณค่า $f(x, y)$ จะใช้สูตร

$$f(x, y) = (f(1, 0) - f(0, 0))x + (f(0, 1) - f(0, 0))y + (f(1, 1) + f(0, 0) - f(0, 1) - f(1, 0))xy + f(0, 0)$$



ภาพที่ 18 การประมาณค่าแบบไบลิเนียร์

โดยทั่วไปแล้วคุณภาพของภาพที่ได้จากการประมาณค่าในช่วงแบบไบลิเนียร์จะดีกว่าการใช้ การประมาณค่าในช่วงแบบเพื่อนบ้านใกล้เคียงที่สุดอย่างเห็น ได้ชัดและการเกิดการหยักจะน้อยกว่ามาก

การประมาณค่าในช่วงแบบ ไบลิเนียร์ เราจะได้ฟังก์ชันที่มีลักษณะเป็นแผ่นระนาบเอียงในทิศทางต่างๆกันต่อเรียงกันเป็นช่วงๆ เมื่อพิจารณาที่ขอบของแต่ละบล็อกจะเห็นว่า ความชันของฟังก์ชันจะมีลักษณะที่ไม่ต่อเนื่อง ด้วยเหตุนี้จึงอาจส่งผลให้บริเวณที่มีรายละเอียดเล็กๆ ในภาพถูกกลทอนคุณภาพลงได้ การใช้พหุนามที่มีดีกรีสูงกว่าในการประมาณค่าฟังก์ชันจะสามารถลดปัญหานี้ได้ โดยทั่วไปจะนิยมใช้ฟังก์ชันที่อยู่ในรูปของพหุนามกำลัง 3 ของตัวแปรสองตัว ในงานวิจัยชิ้นนี้จึงเลือกใช้การประมาณค่าในช่วงแบบนี้ด้วย

3) การประมาณค่าในช่วงแบบไบคิวบิก (Bicubic Interpolation)

การประมาณค่าในช่วงแบบไบคิวบิก จะอาศัยค่าฟังก์ชันจากจุดที่ล้อมรอบจำนวนทั้งหมด 16 จุดมาใช้ในการคำนวณ ฟังก์ชันที่ใช้ในการประมาณค่าจะอยู่ในรูปของพหุนามของตัวแปร x และ y ที่กำลังสูงสุดเป็น 3

$$f(x, y) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij} x^i y^j$$

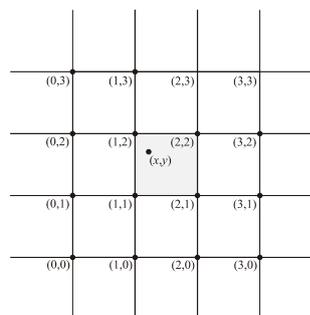
โดยที่ค่าคงที่ a_{ij} ในสูตรนั้นได้มาจากการแก้ระบบสมการเชิงเส้น 16 สมการ 16 ตัวแปร

$$f(3,3) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij} 3^i 3^j \dots\dots\dots(1)$$

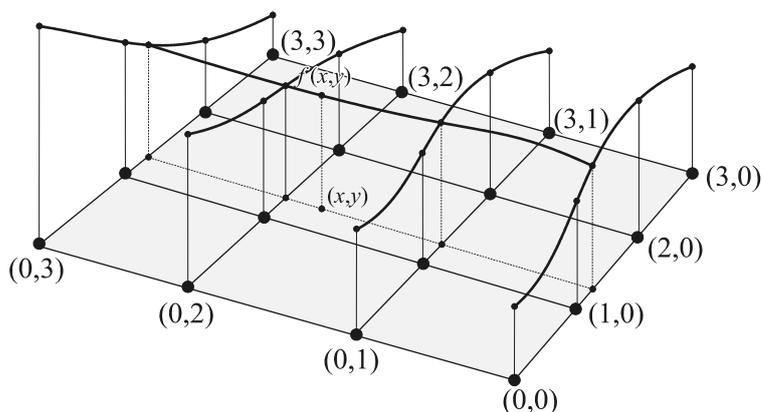
$$f(3,2) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij} 3^i 2^j \dots\dots\dots(2)$$

⋮

$$f(0,0) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij} 0^i 0^j \dots\dots\dots(16)$$



ซึ่งการคำนวณข้างต้นจะเทียบเท่ากับการประมาณค่าในช่วงเพื่อหาพหุนามดีกรี 3 ที่ผ่านจุดที่กำหนดในสองแกนดังภาพ 19



ภาพที่ 19 การประมาณค่าแบบในช่วงแบบไบคิวบิก

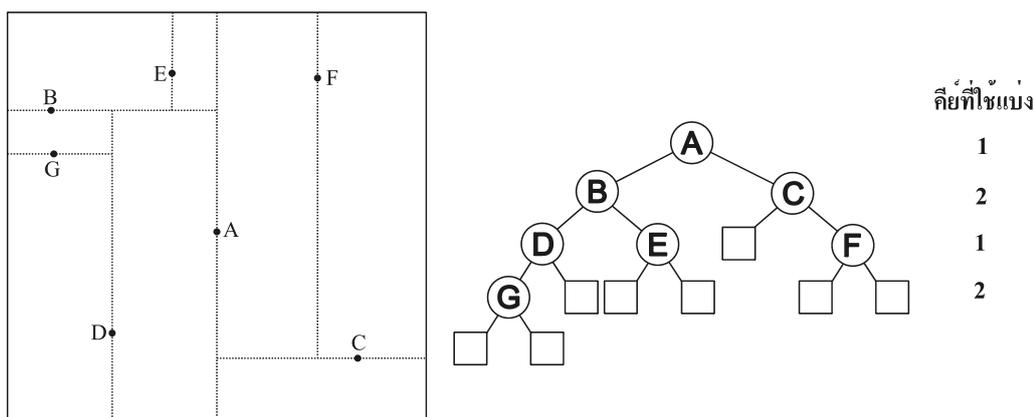
2.2 โครงสร้างข้อมูลแบบเคดีทรี (kd-tree)

เทคนิคหนึ่งในการลดเวลาในการค้นหาของกระบวนการบีบอัดภาพแบบแฟร็กทัลคือการแปลงปัญหาของการจับคู่โดเมนและเรนจ์บล็อก ให้กลายเป็นปัญหาของการค้นหาโหนดใกล้เคียงที่สุดในปริภูมิแบบยูคลิด (Euclidean Space) ซึ่งเป็นปัญหาที่สามารถแก้ได้อย่างมีประสิทธิภาพด้วยโครงสร้างข้อมูลแบบเคดีทรี (kd-tree)

เคดีทรี เป็นโครงสร้างข้อมูลที่ขยายแนวความคิดออกมาจากไบนารีทรี เพื่อให้สามารถรองรับการค้นหาข้อมูลที่ถูกระบุโดยคีย์จำนวนมากว่าหนึ่งหลัก เคดีทรีที่ถูกเสนอเป็นครั้งแรกโดย Bentley (Bentley, 1975) นั้นอาศัยการสร้างทรีเพื่อแบ่งกลุ่มของข้อมูลออกเป็นสองส่วน และทำซ้ำจนกระทั่งเหลือข้อมูลในกลุ่มตามจำนวนที่ต้องการ (อาจมากกว่าหนึ่งในกรณีที่ทำ m -nearest neighbor searching) การแบ่งจะกระทำในคีย์แต่ละหลักหมุนเวียนกันไปตามลำดับ นั่นคือใช้ความสลับพันซ์

$$D = L \bmod k + 1$$

โดยที่ D เป็นลำดับของคีย์ที่จะใช้แบ่ง, L เป็นระดับของทรี (โดยที่โหนดรากมีระดับเป็น 0) และ k เป็นจำนวนหลักของคีย์ การสร้างทรีจะทำโดยการเลือกโหนดขึ้นมาอย่างสุ่ม และนำไปใส่ลงในทรี จนกระทั่งทุกโหนดถูกใส่ลงในทรีจนครบ



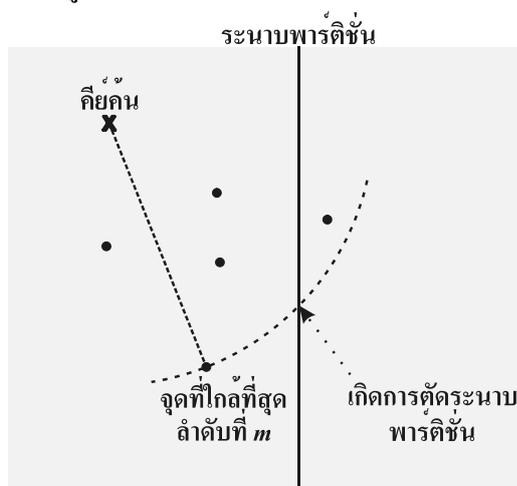
ภาพที่ 20 แสดงเคดีทรีที่สร้างขึ้นจากการแบ่งตามคีย์แต่ละหลักหมุนเวียนกันไป

ทรีที่ได้จากการบวนการนี้มีความเป็นไปได้ที่จะเป็นทรีที่ไม่มีประสิทธิภาพ ในกรณีที่มีข้อมูลมีเป็นจำนวนมาก หรืออาจมีลักษณะไม่สมดุล เคดีทรีแบบที่มีประสิทธิภาพมากกว่าในแง่ของ

การค้นหามีการเปรียบเทียบเรคคอร์ดเป็นจำนวนโดยเฉลี่ยต่ำที่สุดถูกนำเสนอใน (Friedman et al., 1977) โดยมีการปรับปรุงขั้นตอนวิธีในส่วนของวิธีการกำหนดค่าระยะนาบที่จะใช้แบ่งกลุ่มข้อมูล ซึ่งจะใช้คีย์ในหลักที่มีการกระจายตัวของข้อมูลสูงสุดเป็นหลักสำหรับแบ่งกลุ่มแทนที่จะหมุนเวียนแบบเดิม และใช้ค่ามัธยฐานของคีย์นั้นของกลุ่มข้อมูลเป็นค่าสำหรับแบ่งกลุ่ม แต่ละโหนดจะเก็บข้อมูลที่แสดงถึงลักษณะของพาร์ติชัน ยกเว้นโหนดใบจะเก็บตัวเชื่อมโยงไปยังคิวลำดับความสำคัญ (priority queue) ของโหนดที่ใกล้เคียงคีย์คั่นที่สุดจำนวน m โหนด การสร้างทรีด้วยวิธีนี้จะได้เกิดทรีที่มีลักษณะสมดุล โดยใช้เวลาในส่วนของ การสร้างทรีเท่ากับ $O(kM \log M)$

ในส่วนของขั้นตอนการค้นหานั้น จะเริ่มจากการนำคีย์คั่นที่เป็นเวกเตอร์มีขนาดมิติเท่ากับข้อมูลมาเปรียบเทียบกับตัวแบ่งเพื่อคำนวณว่าคีย์คั่นจะอยู่ทางด้านใดของระนาบพาร์ติชัน ซึ่งข้อมูลที่ใกล้เคียงคีย์คั่นก็ควรจะอยู่ด้านนั้นด้วย จากนั้นจึงเลื่อนลงมาและทำซ้ำกระบวนการเดิมจนกระทั่งถึงโหนดใบ ก็จะได้ข้อมูลหรือกลุ่มของข้อมูลที่ใกล้เคียงคีย์คั่นมากที่สุด

ก่อนที่จะทำการคืนค่าข้อมูล m เรคคอร์ดนั้นกลับไป ต้องทำการทดสอบอีกอย่างหนึ่งที่เรียกว่า Bounds Overlap Ball Test เป็นการทดสอบว่าจำเป็นหรือไม่ที่จะต้องพิจารณาข้อมูลบางส่วนที่อยู่อีกฟากหนึ่งของระนาบพาร์ติชัน ซึ่งจะเกิดขึ้นในกรณีที่ระยะห่างระหว่างคีย์คั่นกับข้อมูลที่อยู่ในกลุ่มที่ห่างจากคีย์คั่นมากที่สุดในบรรดาโหนดใกล้เคียงที่สุด m โหนด มีค่ามากกว่าระยะห่างน้อยที่สุดจากคีย์คั่น ไปยังระนาบพาร์ติชัน (ดังภาพที่ 21) ซึ่งทำให้มีความเป็นไปได้ว่า อาจจะมีข้อมูลอีกฟากหนึ่งของระนาบพาร์ติชันที่ใกล้เคียงคีย์คั่นมากกว่าข้อมูลลำดับที่ m ดังกล่าว จึงจำเป็นต้องทำการทดสอบข้อมูลอีกด้านด้วย



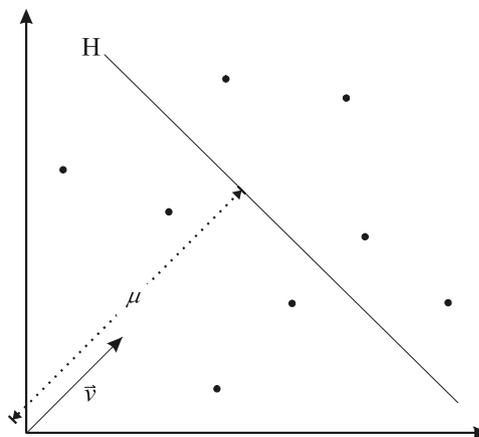
ภาพที่ 21 แสดงกรณีการเกิด Bounds Overlap Ball ทำให้ต้องทดสอบข้อมูลอีกด้านหนึ่งด้วย

ในกรณีที่ใช้เมตริกวัดระยะทางแบบยูคลิดีเนียน การคำนวณระยะห่างระหว่างคีย์คั่นกับข้อมูลสามารถคำนวณได้จากขนาดของเวกเตอร์ผลต่าง สำหรับระยะห่างน้อยที่สุดระหว่างคีย์คั่น $\bar{x} = (x_1, \dots, x_k)$ และพารามิเตอร์ p นั้น คำนวณได้จากสูตร

$$\text{distance}(\bar{x}, p) = \sqrt{\sum_{i=1}^k g_i(\bar{x}, p)^2}$$

$$\text{โดยที่ } g_i(\bar{x}, p) = \begin{cases} x_i - \text{upperbound}_i(p) & ; \text{if } x_i > \text{upperbound}_i(p) \\ \text{lowerbound}_i(p) - x_i & ; \text{if } x_i < \text{lowerbound}_i(p) \\ 0 & ; \text{otherwise} \end{cases}$$

เทคนิคนี้ยังถูกปรับปรุงให้มีประสิทธิภาพมากขึ้น โดยการใช้ระนาบพารามิเตอร์ในแนวอิสระโดยไม่จำเป็นต้องตั้งฉากกับแกนใดแกนหนึ่ง ซึ่งเรียกว่าไฮเปอร์เพลน ไฮเปอร์เพลน H ที่ใช้แบ่งสเปซในแต่ละระดับจะถูกแทนค่าอยู่ในรูปของ (\vec{v}, μ) โดยที่ \vec{v} เป็นเวกเตอร์หนึ่งหน่วยที่ตั้งฉาก H และ μ เป็นระยะทางจากจุดกำเนิดไปตั้งฉาก H



ภาพที่ 22 แสดงการแทนค่าไฮเปอร์เพลน H ในรูปของ (\vec{v}, μ)

การตรวจสอบว่าคีย์คั่น \bar{x} อยู่ด้านใดของไฮเปอร์เพลน H นั้น ทำได้โดยการพิจารณาค่า $\langle \bar{x}, \vec{v} \rangle$ ว่ามีค่ามากกว่าหรือน้อยกว่า μ ส่วนในการตรวจสอบ Bounds Overlap Ball Test ระยะทางจากคีย์คั่น \bar{x} ไปตั้งฉากไฮเปอร์เพลน H คำนวณได้จากสูตร

$$\text{distance}(\bar{x}, H) = |\mu - \langle \bar{x}, \vec{v} \rangle|$$

เห็นได้ชัดว่าขั้นตอนวิธีการกำหนดไฮเปอร์เพลน (\vec{v}, μ) จะมีผลโดยตรงต่อประสิทธิภาพของการค้นหาของทรี ในการสร้างทรีแบบ optimal นั้นเวกเตอร์ \vec{v} จะถูกกำหนดให้เป็นเวกเตอร์หนึ่งหน่วยที่ตั้งฉากกับไอเกนเวกเตอร์ (Eigenvector) ที่ใหญ่ที่สุดของเมทริกซ์ความแปรปรวนร่วมเกี่ยว (covariance matrix) ของเวกเตอร์ข้อมูล ซึ่งเป็นทิศทางที่มีความแปรปรวนของข้อมูลสูงที่สุด (Sproull, 1991)

แต่การสร้างเคสิทรีในแบบที่ให้ประสิทธิภาพในการค้นหาที่ดีที่สุดนี้นั้นเป็นกระบวนการที่ใช้เวลามาก โดยเฉพาะในกรณีที่มีมิติของเวกเตอร์ข้อมูลมีขนาดใหญ่เช่น เวกเตอร์ของบล็อกภาพ วิธีการสร้างเคสิทรีแบบอาศัยฮิวริสติกในการกำหนดทิศทางของระนาบแบ่งสเปซแบบที่เสนอใน (Cardinal, 2001) จะใช้เวลาที่สั้นกว่า ถึงแม้จะไม่ได้ทรีแบบที่ใช้เวลาในการค้นหาที่น้อยที่สุด แต่เมื่อพิจารณาถึงผลรวมของเวลาที่ใช้ในการสร้างทรีและค้นหาแล้วกลับพบว่าเป็นรูปแบบที่เหมาะสมกว่าสำหรับการประยุกต์ใช้ในการบีบอัดภาพแบบแฟร็กทัล ซึ่งจะต้องทำการสร้างเคสิทรีขึ้นมาก่อนที่จะเริ่มต้นกระบวนการบีบอัด

ในขั้นตอนวิธีของ Cardinal ไฮเปอร์เพลนที่จะใช้แบ่งสเปซจะถูกกำหนดให้เป็นระนาบที่ผ่านจุดศูนย์กลางของกลุ่มข้อมูล และวางตัวในแนวตั้งฉากกับแนวที่กลุ่มข้อมูลมีการกระจายตัวออกจากจุดศูนย์กลางมากที่สุด โดยไม่ได้คำนึงว่าไฮเปอร์เพลนนั้นจะต้องแบ่งข้อมูลออกเป็นสองกลุ่มเท่าๆกัน ดังนั้นทรีที่ได้จึงมีความเป็นไปได้ที่อาจจะไม่สมดุล แต่จากการทดลองพบว่าประสิทธิภาพไม่ได้แตกต่างจากทรีแบบที่ดีที่สุดมากนัก แต่มีข้อดีคือใช้เวลาในการสร้างน้อยกว่ามาก โดยรวมแล้วซึ่งวิธีนี้จะทำให้กระบวนการทำงานเร็วขึ้น 1.5 – 3 เท่าเมื่อเทียบกับเคสิทรีแบบที่ดีที่สุด

2.3 ตัวดำเนินการนอร์มัลไลซ์โปรเจกชัน (Normalized Projection Operator)

การนำเคสิทรีมาประยุกต์ใช้ในปัญหาของการบีบอัดภาพแบบแฟร็กทัลถูกเสนอเป็นครั้งแรกใน (Saupe, 1995) ด้วยการแปลงปัญหาของการจับคู่โดเมนและเรนจ์บล็อก ไปเป็นปัญหาของการค้นหาโหนดใกล้เคียงที่สุด แต่เนื่องจากความต่างระหว่างบล็อกในทรี และบล็อกที่จะใช้เป็นตัวค้นนั้น จะถูกพิจารณาภายใต้การปรับค่าความสว่าง และความเปรียบต่างในแบบที่ดีที่สุด ดังนั้นเพื่อที่จะให้กระบวนการค้นหาสามารถมองเห็นความคล้ายในลักษณะนี้ได้ บล็อกเหล่านี้จึงจำเป็นต้องผ่านการแปลงก่อน Saupe ได้พิสูจน์ความสัมพันธ์ต่อไปนี้

$$E(D, R) = \langle R, \Phi(R) \rangle^2 g(\Delta(D, R))$$

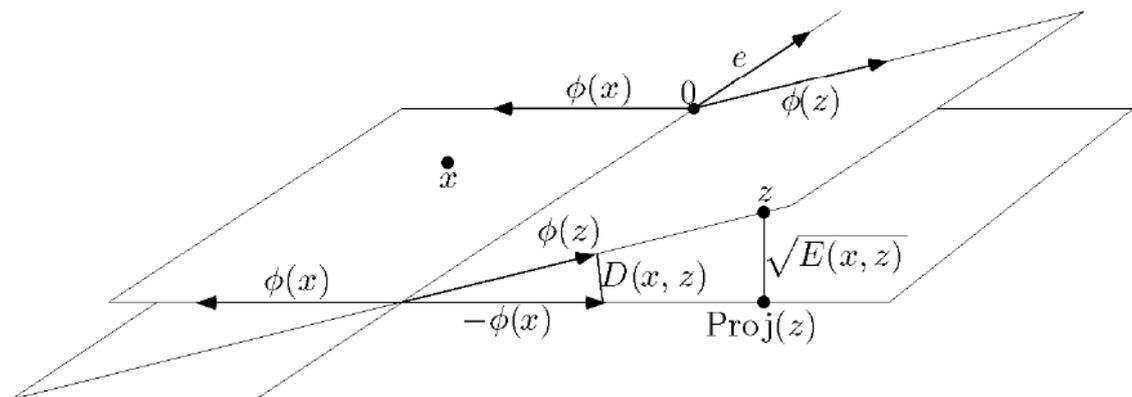
โดยที่ $E(D, R)$ คือค่าที่น้อยที่สุดที่เป็นไปได้ของผลต่างกำลังสองระหว่างโดเมนบล็อก D และเรนจ์บล็อก ภายใต้การปรับเปลี่ยนค่าของความเปรียบต่างและ ความสว่างของโดเมนบล็อก D และฟังก์ชัน g, Δ นิยามโดย

$$g(\Delta) = \Delta^2 \left(1 - \frac{\Delta^2}{4}\right) \quad \text{และ} \quad \Delta(D, R) = \min \{ \|\Phi(R) + \Phi(D)\|, \|\Phi(R) - \Phi(D)\| \}$$

โดยที่ Φ คือตัวดำเนินการนอร์มัลไลซ์โปรเจกชัน (Normalized Projection Operator)

$$\Phi(x) = \frac{x - \langle x, e \rangle e}{\|x - \langle x, e \rangle e\|} \quad \text{เมื่อ} \quad e = \frac{1}{\sqrt{d}} \underbrace{(1, \dots, 1)}_{d \text{ components}}$$

จะเห็นว่าเมื่อกำหนด R คงที่ $E(D, R)$ จะแปรผันตรงกับ $g(\Delta(D, R))$ แต่เนื่องจาก $g(\Delta)$ เป็นฟังก์ชันเพิ่ม การหาค่าต่ำสุดของฟังก์ชัน $g(\Delta)$ จึงสามารถทำได้โดยการหาค่าต่ำสุดของ $\Delta(D, R)$ ดังนั้นปัญหาของการหาค่าต่ำสุดของ $E(D, R)$ ท่ามกลางโดเมนบล็อก D ต่างๆ เมื่อกำหนดบล็อก R ไว้คงที่ จึงสามารถเปลี่ยนรูปไปเป็นปัญหาของการทำ การค้นหาโหนดที่ใกล้เคียงที่สุดของพีเจอร์เวคเตอร์ $\Phi(R)$ ท่ามกลางเซต $\{\pm\Phi(D) \mid D \in \Omega\}$



ภาพที่ 23 แสดงความหมายเชิงเรขาคณิตของตัวดำเนินการนอร์มัลไลซ์โปรเจกชัน

2.4 การส่งคงแบบ (conformal mapping) และการส่งชวาร์ซ-คริสตอฟเฟิล (Schwarz-Christoffel Transformation)

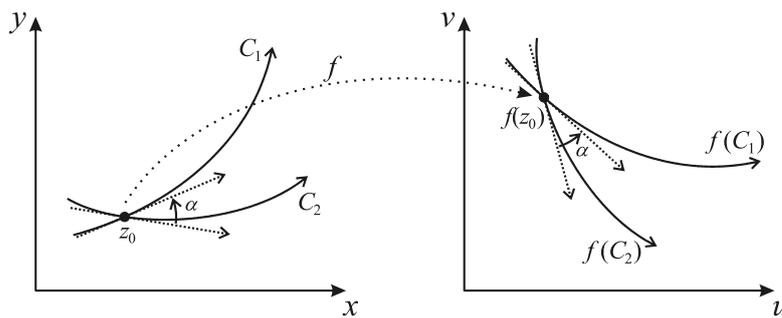
เนื่องจากในงานวิจัยชิ้นนี้ เป็นการพัฒนาขั้นตอนวิธีในการบีบอัดภาพแบบแฟร็กทัล โดยการปรับปรุงฟังก์ชันให้มีลักษณะที่สามารถหมุนในองศาอิสระบนสี่เหลี่ยมจัตุรัส นอกเหนือจากการหมุนในทิศทาง 0, 90, 180, 270 องศาตามแบบปกติ จึงต้องมีการอาศัยทฤษฎีคณิตศาสตร์ในการคำนวณการส่งแบบไม่มีการหักงอบนโพลีกอน ที่เรียกว่าการส่งคงแบบ (conformal mapping)

ทฤษฎีเกี่ยวกับการส่งคงแบบถูกพัฒนาขึ้นโดย George Bernhard Riemann (1826-1866) นักคณิตศาสตร์ชาวเยอรมันซึ่งเป็นผู้ที่มีบทบาทสำคัญอย่างยิ่งในการบุกเบิกทฤษฎีเกี่ยวกับแคลคูลัสของระบบจำนวนเชิงซ้อน Riemann ได้ทำการศึกษาฟังก์ชันบนระนาบจำนวนเชิงซ้อนที่มีคุณสมบัติพิเศษอย่างหนึ่งคือการที่หาอนุพันธ์ได้ (differentiable) และพบว่าภายใต้เงื่อนไขบางประการฟังก์ชันเหล่านี้จะมีคุณสมบัติในการอนุรักษ์มุมตัดระหว่างส่วนโค้งเรียบใดๆ ภายหลัง Riemann ได้ประกาศทฤษฎีบทอันหนึ่งเกี่ยวกับการส่งคงแบบที่มีชื่อเสียงมากที่สุดที่รู้จักกันในนาม Riemann Mapping Theorem

ในปัจจุบันการส่งคงแบบถูกนำไปประยุกต์ใช้ในงานหลากหลายด้าน เช่น การแก้สมการเชิงอนุพันธ์บางรูปแบบ, การสร้างแบบจำลองทางวิศวกรรมทางด้านอากาศพลศาสตร์, การแก้ปัญหาทางอุณหพลศาสตร์, งานทางด้านการประมวลผลภาพ เช่น การแก้ความผิดรูป (distortion) ของภาพ, การแมพภาพลงบนโมเดลสามมิติ, การทำแผนที่โลก ฯลฯ

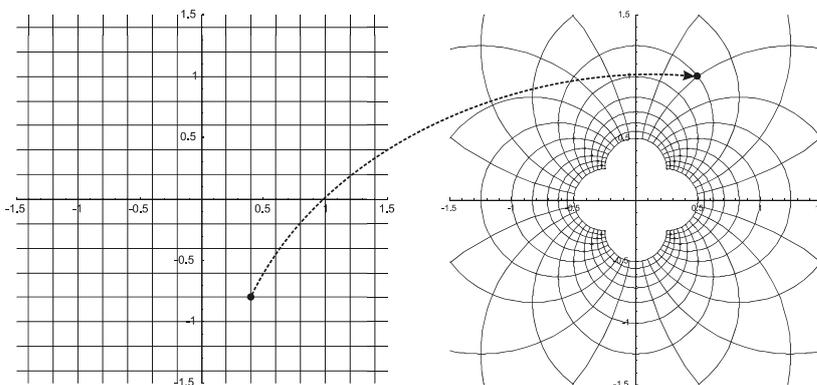
2.4.1 นิยามของการส่งคงแบบ

บทนิยาม กำหนดให้ D เป็นสับเซตของระนาบจำนวนเชิงซ้อน \mathbb{C} จะกล่าวว่าฟังก์ชัน $f: D \rightarrow \mathbb{C}$ เป็นการส่งคงแบบที่จุด $z \in \mathbb{C}$ ก็ต่อเมื่อ f มีค่าที่จุด z และ f อนุรักษ์มุมตัดระหว่างส่วนโค้งเรียบใดๆสองอันที่ผ่านจุด z

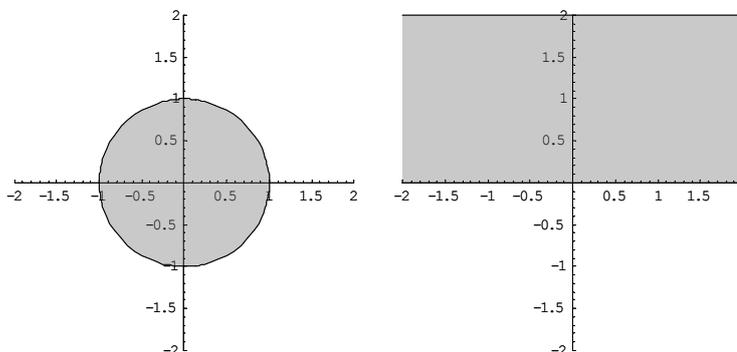


ภาพที่ 24 แสดงตัวอย่างของฟังก์ชัน f ซึ่ง conformal ที่จุด z_0

บทนิยาม ฟังก์ชัน f จะเรียกว่าเป็นการส่งคงแบบก็ต่อเมื่อ f เป็นการส่งคงแบบที่ทุกจุดบน โดเมน



ภาพที่ 25 ตัวอย่าง conformal mapping $f(z) = \frac{1}{z}$



ภาพที่ 26 ตัวอย่างฟังก์ชัน $f(z) = \frac{1+z}{1-z}i$ ซึ่งเป็น conformal mapping ที่ส่งพื้นที่ภายในวงกลม
หนึ่งหน่วยแบบหนึ่งต่อหนึ่งทั่วถึงไปยังครึ่งบนของระนาบจำนวนเชิงซ้อน \mathbb{C}

ทฤษฎีบท Riemann Mapping Theorem บอกให้เราทราบว่าทุกสับเซตของ \mathbb{C} ที่เป็นบริเวณเชื่อมโยงเชิงเดียว (simply-connected) จะสามารถดลึงค์กัน โดยฟังก์ชันวิเคราะห์ (analytic function) แบบหนึ่งต่อหนึ่งทั่วถึงกันได้เสมอ หรือกล่าวแบบจำเพาะเจาะจงก็คือ ทฤษฎีบทนี้รับประกันว่าไม่ว่าเราจะมีโพลีกอนเป็นรูปทรงอะไรก็ตาม จะมีฟังก์ชันวิเคราะห์แบบหนึ่งต่อหนึ่งทั่วถึงจากครึ่งบนของระนาบ \mathbb{C} แบบหนึ่งต่อหนึ่งทั่วถึงส่งไปยังโพลีกอนนั้นเสมอ ทั้งนี้ Riemann Mapping Theorem เพียงแค่กล่าวถึงความมีอยู่จริงของฟังก์ชันดังกล่าว แต่ไม่ได้กล่าวถึงการนิยามขั้นตอนวิธีของการสร้างฟังก์ชันดังกล่าวที่เป็นรูปธรรมนั้น สามารถทำได้โดยอาศัยหลักการที่มีชื่อว่า Schwarz-Christoffel Transformation

2.4.2 การส่งแบบชวาร์ซ-คริสตอฟเฟิล (Schwarz-Christoffel Transformation)

การส่งแบบชวาร์ซ-คริสตอฟเฟิล เป็นหลักการที่ช่วยให้เราสามารถสร้าง การส่งลงแบบจากครึ่งบนของระนาบจำนวนเชิงซ้อน แบบหนึ่งต่อหนึ่งทั่วถึง ไปยังโพลีกอนที่เป็นบริเวณเชื่อมโยงเชิงเดียว ไม่ว่าจะมรูปร่างซับซ้อนเพียงใดและจะมีขอบเขตหรือไม่ก็ตาม ทฤษฎีนี้ถูกตั้งชื่อเพื่อเป็นเกียรติแก่นักคณิตศาสตร์ชาวเยอรมันสองคนที่เป็นผู้ค้นพบ คือ H. A. Schwarz (1843-1921) และ E. B. Christoffel (1829-1900) แนวคิดของการส่งแบบชวาร์ซ-คริสตอฟเฟิลนั้นเริ่มมาจากการพิจารณาฟังก์ชัน f ที่มีอนุพันธ์เป็นฟังก์ชันที่อยู่ในรูป

$$f'(z) = A(z - x_1)^{-\alpha_1} (z - x_2)^{-\alpha_2} \dots (z - x_{n-1})^{-\alpha_{n-1}}$$

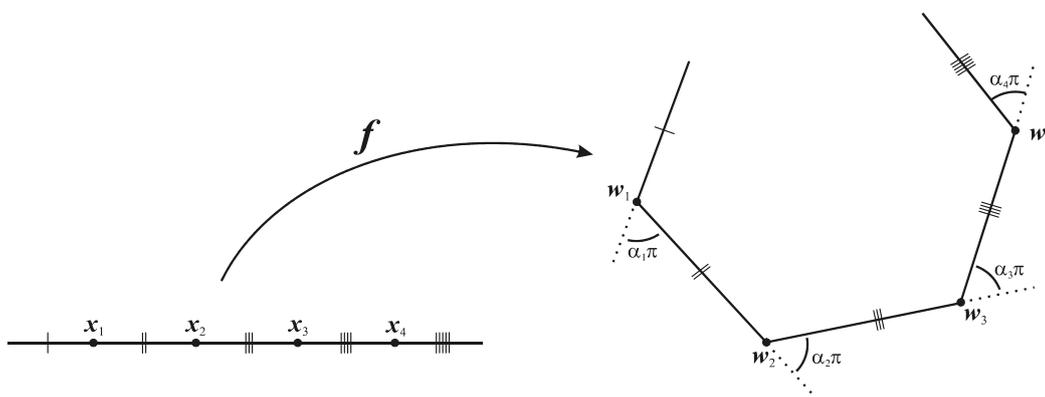
โดยที่ α_i เป็นจำนวนจริงซึ่ง $-1 < \alpha_i < 1$ และ A เป็นจำนวนเชิงซ้อนใดๆ สมมติว่า $z = x$ อยู่บนแกนจำนวนจริงโดยที่ $x_j < x < x_{j+1}$ เมื่อพิจารณา argument หรือค่าของมุมที่ทำกับแกนจำนวนจริง จะพบว่า

$$\begin{aligned} \arg f'(x) &= \arg(A(x - x_1)^{-\alpha_1} (x - x_2)^{-\alpha_2} \dots (x - x_{n-1})^{-\alpha_{n-1}}) \\ &= \arg(A) - \alpha_1 \arg(x - x_1) - \alpha_2 \arg(x - x_2) - \dots - \alpha_{n-1} \arg(x - x_{n-1}) \\ &= \arg(A) - \alpha_{j+1}\pi - \alpha_{j+2}\pi - \dots - \alpha_{n-1}\pi \end{aligned}$$

นั่นคือส่วนของแกนจำนวนจริงในช่วงระหว่าง x_j ถึง x_{j+1} จะถูกส่งไปเป็นเส้นตรงที่ทำมุมกับแกนจำนวนจริงเป็นค่าคงที่เท่ากับ $\arg(A) + \alpha_{j+1}\pi + \alpha_{j+2}\pi + \dots + \alpha_n\pi$ ในขณะเดียวกัน ถ้า x มีค่าในช่วงที่น้อยลง คืออยู่ในช่วง $x_{j-1} < x < x_j$ จะพบว่า

$$\arg f'(x) = \arg(A) - \alpha_j\pi - \alpha_{j+1}\pi - \alpha_{j+2}\pi - \dots - \alpha_{n-1}\pi$$

นั่นคือเมื่อ x มีค่าเพิ่มขึ้นผ่านจุด x_j ค่าฟังก์ชันในช่วงใหม่จะเป็นเส้นตรงที่หักทำมุม $\alpha_j\pi$ กับเส้นตรงเดิม นั่นคือฟังก์ชัน f จะกระทำต่อจุดบนแกนจำนวนจริงในลักษณะดังภาพ



ภาพที่ 27 แสดงลักษณะการแมพแกนจำนวนจริงของฟังก์ชัน f

ดังนั้นจะเห็นว่ารูปร่างของโพลีกอนนั้นถูกกำหนดโดยมุมที่จุดยอด ซึ่งมีความสัมพันธ์โดยตรงกับค่า α_k ต่างๆ และค่าคงที่ A ซึ่งเป็นจำนวนเชิงซ้อนนั้นจะมีบทบาทในการกำหนดขนาดและลักษณะการวางของโพลีกอน กล่าวคือ $\arg A$ จะกำหนดมุมที่โพลีกอนถูกหมุนไป และ $|A|$ จะกำหนดขนาดของสเกล และเนื่องจากสมการข้างต้นเป็นฟังก์ชันในรูปของอนุพันธ์ของ f ดังนั้นฟังก์ชัน f จะมีพจน์ของค่าคงที่อีกเทอมหนึ่งเพิ่มเข้ามาด้วย ซึ่งค่าคงที่นี้เองจะมีผลในการกำหนดตำแหน่งของโพลีกอนบนระนาบจำนวนเชิงซ้อน

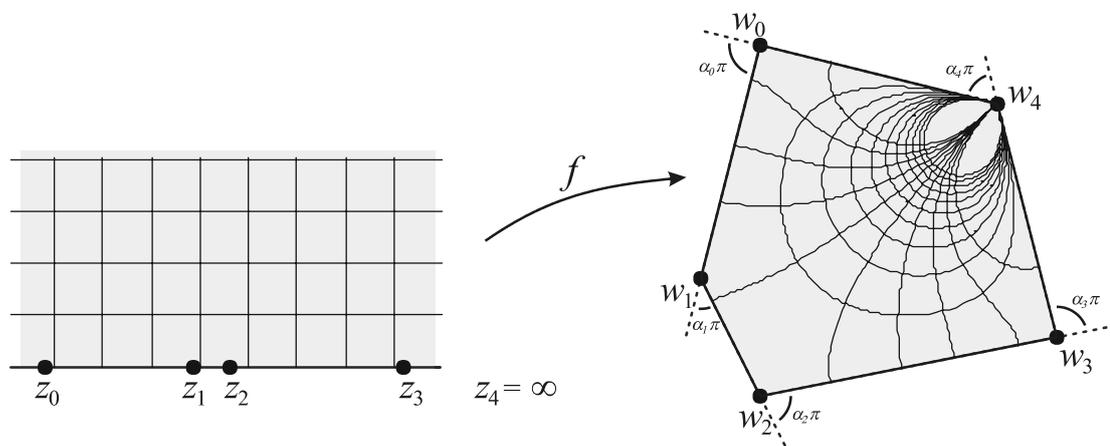
ทฤษฎีบท การส่งแบบชวาร์ซ-คริสตอฟเฟิล

ให้ P เป็นโพลีกอน n ด้าน ที่มีจุดยอดอยู่ที่จุด w_1, \dots, w_n และมีมุมภายนอกที่จุดยอดเป็น $\pi\alpha_1, \dots, \pi\alpha_n$ ตามลำดับ โดยที่ $-1 < \alpha_k < 1$ และ $\sum_{i=1}^n \alpha_i = 2$ เมื่อกำหนดให้ c เป็นค่าคงที่ที่เหมาะสม และเลือก z_i เป็นจุดที่แมพไปลงจุดยอด w_i ของโพลีกอน เมื่อเลือก z_0 จาก z_i ใดๆ ฟังก์ชัน

$$f(z) = f(z_0) + c \int_{z_0}^z \prod_{i=1}^{n-1} (\zeta - z_i)^{-\alpha_i} d\zeta$$

เป็นการส่งแบบที่ส่งครึ่งบนของระนาบจำนวนเชิงซ้อนแบบหนึ่งต่อหนึ่งไปทั่วถึงโพลีกอน P

ทั้งนี้ค่า c ในสมการข้างต้นจะได้มาจากการแทนค่า z ด้วย z_i จุดใดจุดหนึ่ง และแก้สมการอินทิกรัล



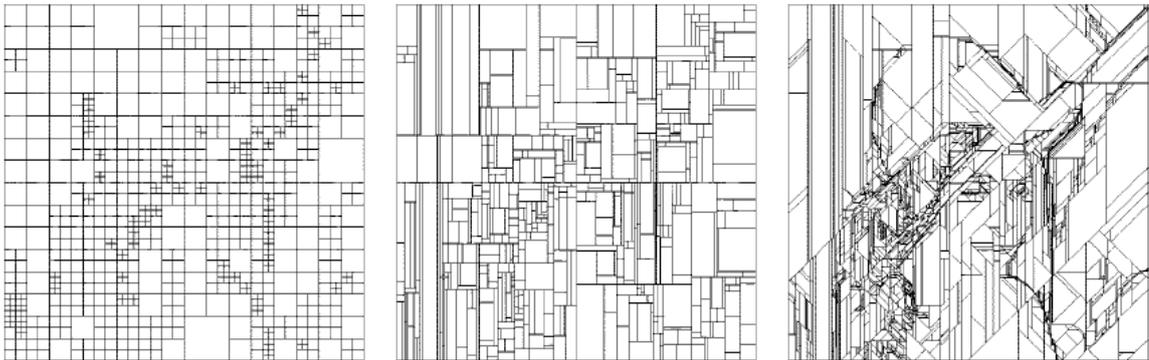
ภาพที่ 28 แสดงตัวอย่างฟังก์ชัน f ที่สร้างขึ้นด้วยหลักการของการส่งแบบซาร์ช-คริสทอฟเฟิล

3. งานวิจัยที่เกี่ยวข้อง

การเข้ารหัสแบบแฟร็กทัลเป็นกระบวนการที่ใช้เวลามาก ในช่วงเวลาสิบปีกว่าปีที่ผ่านมา จึงมีความพยายามที่จะคิดค้นเทคนิคในการลดเวลาในการประมวลผล และเพิ่มคุณภาพของการบีบอัด ซึ่งแยกประเภทโดยสังเขปได้แก่

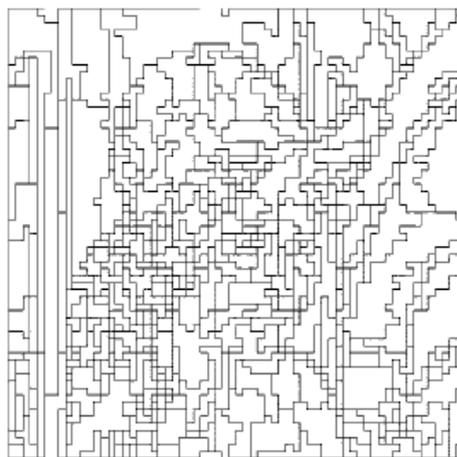
3.1 การแบ่งประเภทของบล็อกด้วยเทคนิคต่างๆ โดยที่การค้นหาจะทำเฉพาะระหว่างบล็อกประเภทเดียวกันหรือใกล้เคียงเท่านั้น ทำให้ขนาดของโดเมนพูลเล็กลง (Jacquin, 1992) เสนอวิธีแบ่งประเภทของบล็อกเป็น 3 คลาส ตามลักษณะปรากฏ คือ บล็อกที่เป็นสีพื้น, บล็อกที่มีเส้นขอบของวัตถุ และบล็อกที่เป็นลวดลาย (texture) และทำการจับคู่เฉพาะบล็อกที่จัดอยู่ในคลาสเดียวกัน (Fisher, 1994) แบ่งประเภทของบล็อกตามลักษณะของการกระจายความเข้มสี โดยคำนวณค่าเฉลี่ยและความแปรปรวนของสีในแต่ละควอดแรนท์ ของบล็อก และแบ่งบล็อกเป็น 72 คลาส ตามลักษณะการเรียงลำดับของควอดแรนท์เมื่อใช้ค่าทั้งสองเป็นเกณฑ์ ซึ่งให้ผลดีในกรณีของการบีบอัดแบบบิตเรตต่ำ (Boss and Jacobs, 1994) พัฒนาต่อจากการแบ่งประเภทบล็อก โดยใช้ไอคิไทป์ (Archetype) ซึ่งเป็นบล็อกที่ให้ค่าผลรวมของค่าความผิดพลาดเทียบกับทุกบล็อกที่เหลือในกลุ่มน้อยที่สุด โดยใช้เป็นตัวแทนของกลุ่มในการตัดสินใจเลือกกลุ่มที่จะทำการค้นหาต่อไป โดยรวมแล้ววิธีนี้จะให้คุณภาพของภาพที่ดีกว่าการเลือกกลุ่มโดยใช้วิธีเดียวกับขั้นตอนวิธีใช้แบ่งประเภทบล็อก การแบ่งโดเมนพูลโดยวิธีคลัสเตอร์ถูกเสนอใน (Wein and Blake, 1996) โดยมีการพัฒนาขั้นตอนวิธีในการคำนวณผลต่างรูทมินัสแควร์ระหว่างคู่บล็อกแบบรวดเร็ว เพื่อใช้ในการสร้างคลัสเตอร์บนเซตของโดเมนบล็อก (Polvere and Nappi, 2000) ใช้อ็องศาของจุดศูนย์กลางมวลของภาพเพื่อเป็นอิทธิพลในการทำการแยกประเภทโดเมนบล็อก ตามความใกล้เคียงขององศาของจุดศูนย์กลางมวล โดยอาศัยแนวคิดที่ว่าภาพที่คล้ายกันมักจะมีองศาของจุดศูนย์กลางมวลใกล้เคียงกัน

3.2 การปรับปรุงรูปแบบพาร์ติชัน ซึ่งมีทั้งการออกแบบจากบนลงล่าง (top-down) เช่น พาร์ติชันแบบควอดทรี (Fisher, 1994), พาร์ติชันแบบ HV (Fisher and Menlove, 1994) และ พาร์ติชันแบบ Polygonal (Reusens, 1994)



ภาพที่ 29 พาร์ติชันแบบ Quadtree, HV และ Polygonal ตามลำดับ

และการออกแบบจากล่างขึ้นบน (bottom-up) เช่น Evolutionary Partitioning (Saupe and Ruhl, 1996)(Ruhl, Hartenstein and Saupe, 1997), Region-Based Partitioning (Thomas and Deravi, 1995) (Hartenstein, Ruhl and Saupe, 2000) ซึ่งอาศัยการแตกภาพออกเป็นพาร์ติชันเล็กๆ และหลอมพาร์ติชันที่อยู่ติดกันเข้าด้วยกันเมื่อการหลอมรวมนั้นสอดคล้องเงื่อนไขที่กำหนด เช่น ค่าความผิดพลาดของพาร์ติชันใหม่อยู่ในระดับที่ยอมรับได้ แนวทางการแบ่งพาร์ติชันแบบนี้จะให้ อัตราส่วนของ Rate-Distortion ที่ดีกว่าการออกแบบจากบนลงล่าง (Hartenstein, Ruhl and Saupe, 2000)



ภาพที่ 30 พาร์ติชันแบบ Region-Based

3.3 การพัฒนาโครงสร้างข้อมูลในการค้นหา (Saupe, 1995) ได้นำเสนอแนวทางในการค้นหาแบบใหม่ ที่มีประสิทธิภาพสูงกว่าเดิมมาก ด้วยการแปลงโดเมนบล็อคเป็นเวกเตอร์ และสร้างโครงสร้างข้อมูลแบบ เคดีทรี (kd-tree) ที่บรรจุโดเมนบล็อคที่ผ่านการแปลงโดยตัวดำเนินการนอร์มอลไลซ์โปรเจกชัน (Normalized Projection Operator) และทำการค้นหาโหนดที่ใกล้เคียงที่สุดกับเรนจ์บล็อคที่ผ่านการแปลงแล้ว วิธีนี้สามารถลดเวลาในการค้นหาสำหรับแต่ละเรนจ์บล็อคเหลือออร์เดอร์ของลอการิทึมจากเดิมที่ใช้เวลาเชิงเส้น (Tong and Wong, 2002) ปรับปรุงขั้นตอนวิธีของ Saupe โดยกำจัดความผิดพลาดที่เกิดจากการปัดเศษ (quantization) ของ ค่าพารามิเตอร์ของความสว่างและความเปรียบต่าง โดยเปลี่ยนสูตรในการแปลงบล็อค ก่อนที่จะใช้ขั้นตอนวิธีเดิม ในการค้นหา (Cardinal, 2001) เสนอแนะวิธีในการสร้างเคดีทรี สำหรับใช้ในกระบวนการบีบอัดแบบแฟร็กทัล โดยอาศัยวิธีสถิติในการกำหนดทิศทางของไฮเปอร์เพลน (hyperplane) ที่ใช้สำหรับการแบ่งสเปซออกเป็นสองส่วนเพื่อแตกออกเป็นไบนารีทรี โดยจะทำการแบ่งกลุ่มในแนวที่ตั้งฉากกับแนวที่มีการกระจายของข้อมูลมากที่สุดก่อน ซึ่งโดยรวมแล้วจึงสามารถลดเวลาในการส่วนของฟรีโปรเซสได้ ทำให้เวลาโดยรวมลดลง แต่ไม่ส่งผลกระทบต่อคุณภาพของภาพนัก เทคนิคการลดเวลาอีกรูปแบบถูกเสนอใน (Lai, Lan and Siu, 2003) โดยการคำนวณค่าคงที่บางอย่างของแต่ละบล็อคว่าไว้ก่อน และนำไปใช้สำหรับการคำนวณขอบเขตล่างของผลต่างรูทมินัสแควร์ระหว่างโดเมนบล็อคและเรนจ์บล็อคในขั้นตอนของการค้นหา เพื่อใช้เป็นเงื่อนไขในการตัดสินใจที่จะไม่ประมวลผลบางโดเมนบล็อคที่มีขอบเขตล่างของความผิดพลาดสูงกว่าความผิดพลาดที่ต่ำที่สุดเท่าที่พบมา

3.4 การแปลง (Au, Liou and Ma, 1997) ได้เสนอวิธีลดมิติของเวกเตอร์ที่ใช้ในขั้นตอนวิธีของ Saupe โดยแปลงบล็อคให้อยู่ในรูปของโดเมนเชิงความถี่ (Frequency Domain) และสร้างพีเจอร์เวกเตอร์จากบริเวณย่านความถี่ต่ำ วิธีคล้ายๆกันนี้ถูกเสนอใน (Truong et al., 2000) โดยการจัดลำดับการคำนวณในโดเมนความถี่ที่เหมาะสม ซึ่งสามารถลดความซ้ำซ้อนของการคำนวณ (Cardinal, 1998) ใช้วิธีแปลงพีเจอร์เวกเตอร์ด้วย การแปลง Karhunen-Loeve ร่วมกับการใช้เคดีทรี ซึ่งสามารถลดเวลาในกระบวนการค้นหาได้เช่นเดียวกัน

อุปกรณ์และวิธีการ

1. อุปกรณ์

ฮาร์ดแวร์	คอมพิวเตอร์ Pentium 4 2.6GHz หน่วยความจำ 512 MB
ซอฟต์แวร์	GNU C compiler, Matlab 7.1, Schwarz-Christoffel toolbox for Matlab

2. วิธีการ

2.1 แนวคิดของงานวิจัย

ในขั้นตอนวิธีการบีบอัดแบบแฟร็กทัลแบบดั้งเดิมนั้น โดเมนและเรนจ์บล็อกที่ใช้กันจะมีลักษณะเป็นรูปสี่เหลี่ยมจัตุรัส ด้วยเหตุนี้ลักษณะการส่งระหว่างโดเมนและเรนจ์บล็อกจึงถูกจำกัดให้มีได้อย่างมากเพียง 8 รูปแบบเท่านั้น กล่าวคือประกอบด้วยการหมุนขึ้นละ 90 องศา รวมกับการพลิกตามแนวเส้นทแยงมุม แต่เนื่องจากโดยธรรมชาติของภาพถ่าย ความคล้ายคลึงไม่จำเป็นต้องเกิดในมุมตั้งฉากเสมอไป ดังนั้นฟังก์ชันที่เหมาะสมกว่าที่อยู่ในมุมอื่นๆจึงอาจถูกมองข้ามไป ประกอบกับข้อจำกัดในเรื่องความหลากหลายของลักษณะโดเมน ทำให้เป็นการยากที่จะปรับปรุงคุณภาพของการบีบอัดข้อมูลให้ดียิ่งขึ้น

เป้าหมายของงานวิจัยชิ้นนี้ คือการพัฒนาขั้นตอนวิธีการจับคู่โดเมนและเรนจ์บล็อกในองศาอื่น ๆ นอกเหนือจากการหมุนเป็นมุมฉาก โดยผนวกเข้ากับขั้นตอนวิธีการค้นหาโหนดใกล้เคียงที่สุดของ Saupe ซึ่งได้รับการยอมรับว่าเป็นขั้นตอนวิธีการทำการบีบอัดแบบแฟร็กทัลที่มีประสิทธิภาพสูง

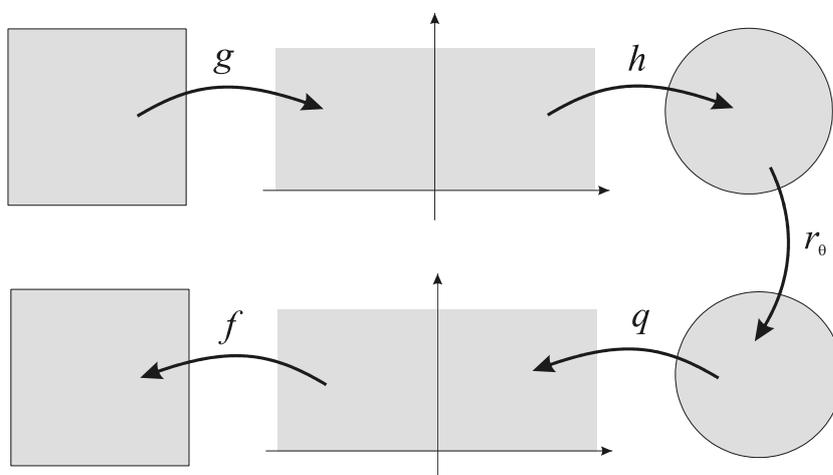
ในความเป็นจริงแล้วนั้น การที่มีลักษณะการหมุนที่หลากหลายขึ้นทำให้ต้องการจำนวนบิตมากขึ้นในการส่งสัญญาณบอกตัวออร์ทอกอนัลให้ทราบถึงรูปแบบเหล่านี้ ขนาดของบิตที่ใช้เข้ารหัสแต่ละพาร์ติชันจึงสูงขึ้นด้วย แต่เนื่องจากการที่มีรูปแบบของการหมุนที่หลากหลายขึ้นกว่าเดิม น่าจะทำให้สามารถพบความคล้ายในระดับของพาร์ติชันที่ใหญ่ขึ้น จึงทำให้จำนวนของพาร์ติชันน้อยลง จึงคาดหวังว่าการแลกเปลี่ยนในจุดนี้น่าจะส่งผลให้ขนาดของไฟล์ที่บีบอัดมีขนาดเล็กลง

2.2 ฟังก์ชันการหมุนภายในสี่เหลี่ยมจัตุรัส

จากการทดลองเบื้องต้นพบว่า ลักษณะการหมุนที่เหมาะสม ควรมีลักษณะที่ก่อให้เกิดความผิดรูป (distortion) ในระดับต่ำที่สุดเท่าที่จะเป็นไปได้ และที่สำคัญจะต้องไม่ทำให้เกิดการหักงอ ไม่เช่นนั้นแล้วโดเมนบล็อกที่มีเส้นขอบทุกชนิดเมื่อผ่านการหมุนจะเกิดเส้นขอบที่หักงอเสมอ ซึ่งเห็นได้ชัดว่าจะเป็นอุปสรรคในการจับคู่โดเมนเข้ากับเรนจ์เนื่องจากลักษณะโดยธรรมชาติของเส้นขอบของวัตถุ โดยทั่วไป มักจะมีแนวโน้มที่จะราบเรียบมากกว่าหักงอ ดังนั้นฟังก์ชันสำหรับใช้ในการนิยามการหมุนแบบของสาอัสระบนสี่เหลี่ยมจัตุรัสในงานวิจัยชิ้นนี้จึงถูกเลือกให้เป็นการส่งคงแบบ (conformal mapping)

ฟังก์ชันการส่งคงแบบของการหมุนภายในสี่เหลี่ยมจัตุรัสที่ต้องการ จะอยู่ในรูปของการประกอบ (composition) ของการส่งคงแบบหนึ่งต่อหนึ่งและทั่วถึงจำนวน 5 ฟังก์ชัน

$$\varphi_\theta = f \circ q \circ r_\theta \circ h \circ g$$



ภาพที่ 31 แสดงการเชื่อมโยงของแต่ละฟังก์ชัน

โดยที่แต่ละฟังก์ชันนิยามได้ดังนี้

2.1.1 ฟังก์ชัน q กำหนดให้เป็นฟังก์ชันโมเบียส ที่นิยามโดย $q(z) = i \frac{1+z}{1-z}$ ซึ่งจะส่งวงกลมหนึ่งหน่วยไปยังครึ่งบนของระนาบจำนวนเชิงซ้อนแบบ 1-1 ทั่วถึงดังภาพที่ 26

2.2.2 ฟังก์ชัน h เป็นผกผัน (inverse) ของ g

2.2.3 ฟังก์ชัน r_θ เป็นการหมุนรอบจุดกำเนิด นิยามโดย $r_\theta(z) = ze^{i\theta}$ โดยที่ θ คือ องศาการหมุน

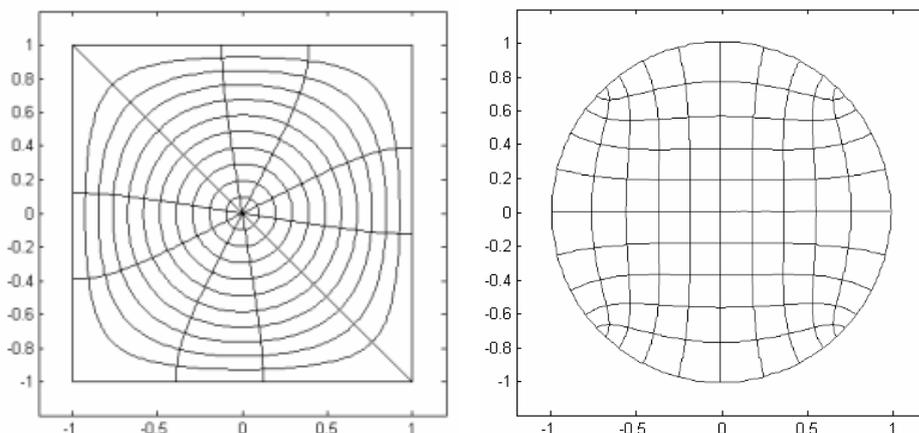
2.2.4 ฟังก์ชัน g เป็นผกผันของ f

2.2.5 ฟังก์ชัน f เป็นฟังก์ชันที่ส่งจากครึ่งบนของระนาบจำนวนเชิงซ้อนไปยังสี่เหลี่ยมจัตุรัส ซึ่งสร้างขึ้นโดยอาศัยหลักการของซวาร์ช-คริสทอฟเฟิล (Schwarz-Christoffel Transformation) ซึ่ง รูปแบบทั่วไปของฟังก์ชันคือ

$$f(z) = f(z_0) + c \int_{z_0}^z \prod_{i=1}^{n-1} (\zeta - z_i)^{-\alpha_i} d\zeta$$

ในกรณีนี้โพลิกอนคือสี่เหลี่ยมจัตุรัสที่มีจุดยอดที่ $-1+i$, $1+i$, $1-i$ และ $-1-i$ ดังนั้นค่าตัวแปร n จึงมีค่าเป็น 4 เนื่องจากมุมที่จุดยอดมีขนาด $\frac{\pi}{2}$ เท่ากันหมด ดังนั้น $\alpha_1 = \alpha_2 = \alpha_3 = \frac{1}{2}$ และเมื่อกำหนดให้จุด -1 , 0 , 1 , ∞ ถูกส่งไปลงยังจุด $1+i$, $-1+i$, $-1-i$, $1-i$, ตามลำดับ ทำการแก้สมการหาค่า c จะได้ว่าฟังก์ชัน f ที่เราต้องการคือ

$$f(z) = -1+i + c \int_0^z \frac{1}{\sqrt{t^3-t}} dt \quad \text{โดยที่ค่าคงที่ } c = \frac{-2i}{\int_0^1 \frac{1}{\sqrt{t^3-t}} dt} \approx 0.7627598$$



ภาพที่ 32 ลักษณะการส่งแบบอนุรักษมุมตัดระหว่างสี่เหลี่ยมกับวงกลม

เนื่องจากฟังก์ชันมีนิยามค่อนข้างซับซ้อนและการประมวลผลใช้เวลานานมาก จึงต้องทำการคำนวณไว้ล่วงหน้าโดยอาศัยโปรแกรม Matlab โดยทำการคำนวณค่าของฟังก์ชัน φ_θ ที่ผ่านการสเกลและเลื่อนตำแหน่งให้เหมาะสมบนสี่เหลี่ยมขนาด 4×4 , 8×8 , 16×16 , 32×32 และ 64×64 พิกเซล โดยกำหนดค่า $\theta = \frac{\pi}{256} j$ โดยที่ $j = 0, \dots, 63$ จากนั้นเก็บค่าที่ได้จากการคำนวณไว้ตารางไว้ ในกรณีที่ $j = 64, \dots, 255$ สามารถคำนวณได้โดยง่ายจากข้อมูลชุดเดิมโดยอาศัยการหมุนรูปสี่เหลี่ยม

ตารางที่ 1 ตัวอย่างของตารางการหมุนที่คำนวณได้สำหรับบล็อกขนาด 4×4 เมื่อ $j = 1, 2, 3$

j	
1	0.0000+0.2535i, 0.9655+0.0000i, 1.9664+0.0000i, 2.7465+0.0000i 0.0000+1.0336i, 0.9877+1.0126i, 1.9874+0.9877i, 3.0000+0.9655i 0.0000+2.0345i, 1.0126+2.0123i, 2.0123+1.9874i, 3.0000+1.9664i 0.2535+3.0000i, 1.0336+3.0000i, 2.0345+3.0000i, 3.0000+2.7465i
2	0.0000+0.3585i, 0.9300+0.0000i, 1.9336+0.0000i, 2.6415+0.0000i 0.0000+1.0664i, 0.9758+1.0255i, 1.9745+0.9758i, 3.0000+0.9300i 0.0000+2.0700i, 1.0255+2.0242i, 2.0242+1.9745i, 3.0000+1.9336i 0.3585+3.0000i, 1.0664+3.0000i, 2.0700+3.0000i, 3.0000+2.6415i
3	0.0000+0.4392i, 0.8933+0.0000i, 1.9015+0.0000i, 2.5608+0.0000i 0.0000+1.0985i, 0.9642+1.0388i, 1.9612+0.9642i, 3.0000+0.8933i 0.0000+2.1067i, 1.0388+2.0358i, 2.0358+1.9612i, 3.0000+1.9015i 0.4392+3.0000i, 1.0985+3.0000i, 2.1067+3.0000i, 3.0000+2.5608i

ขั้นตอนวิธีการทำงานของฟังก์ชันการหมุนแบบไม่หักงอโดยอาศัยการส่งคงแบบและการประมาณค่าในช่วงแบบไบคิวบิกแสดงได้ดังรหัสเทียมนี้

```

imagetype SCRotate(image, rot)
  rotq = (rot-256) div 64 mod 4;
  rotr = (rot-256) mod 64;
  Case rotq of
  0 :   For y = 0 to image.height-1 do
        For x = 0 to image.width-1 do
          output.pixel[x][y] = BicubicInterpolation(image,
            Re(table[rotr][x][y],
              Im(table[rotr][x][y]));
        End;
      End;
  1 :   For x = image.width-1 downto 0 do
        For y = 0 to image.height-1 do
          output.pixel[x][y] = BicubicInterpolation(image,
            Re(table[rotr][image.width-1-
              x][y], Im(table[rotr][
                image.width-1-x][y]));
        End;
      End;
  2 :   For y = 0 to image.height-1 do
        For x = 0 to image.width-1 do
          output.pixel[x][y] = BicubicInterpolation(image,
            Re(table[rotr][ image.width-1-
              x][ image.height-1-y],
              Im(table[rotr][ image.width-1-
                x][ image.height-1-y]));
        End;
      End;
  3 :   For x = 0 to image.width-1 do
        For y = image.height-1 downto 0 do
          output.pixel[x][y] = BicubicInterpolation(image,
            Re(table[rotr][x][
              image.height-1-y],
              Im(table[rotr][x][
                image.height-1-y]));
        End;
      End;
  End;
  Return output;
End;
```



ภาพที่ 33 แสดงผลของการหมุนภายในรูปสี่เหลี่ยมด้วยการส่งคองแบบ

2.3 การทดลอง

2.3.1 จุดประสงค์การทดลอง

ก. เพื่อวัดประสิทธิภาพของขั้นตอนวิธีนำเสนอ เทียบกับขั้นตอนวิธีเดิม โดยการเปรียบเทียบคุณภาพของภาพที่ถอดรหัส (PSNR) ต่ออัตราส่วนการบีบอัด

ข. เปรียบเทียบเวลาที่ใช้ในการประมวลผลระหว่างขั้นตอนวิธีเดิมและขั้นตอนวิธีใหม่

ค. เปรียบเทียบประสิทธิภาพในการบีบอัดแบบเฟิร์กทัลกับการบีบอัดแบบเจบีกับภาพตัวอย่าง

2.3.2 ปัจจัยที่มีผลต่อคุณภาพของภาพและอัตราส่วนการบีบอัด

ก. คุณลักษณะของภาพ

ข. ความละเอียดของการหมุน

ค. ระดับของควอดทรี และ ขนาดของพาร์ติชันที่ใหญ่ที่สุดและเล็กที่สุด

ง. การตั้งค่าความผิดพลาดเฉลี่ยต่อพิกเซลเฉลี่ยที่ยอมรับได้

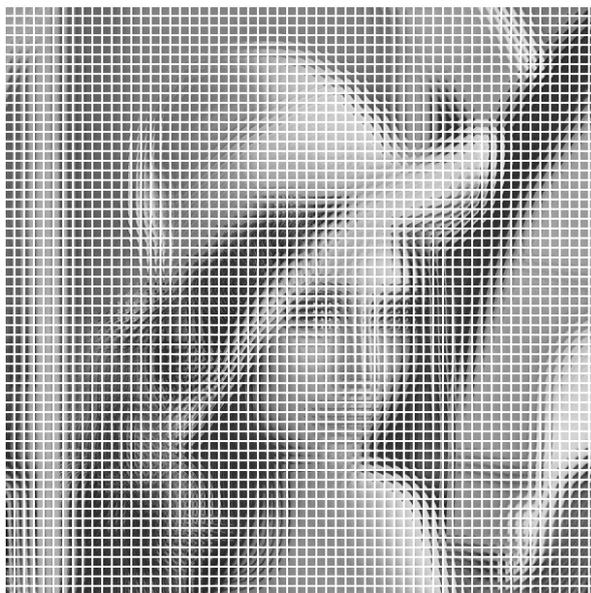
2.3.3 ขั้นตอนการทำงานของโปรแกรมบีบอัดภาพ

ก. อ่านไฟล์ภาพ

ข. คำนวณพารามิเตอร์ไบคิวบิกของภาพ แต่ละจุดของภาพจะมีพารามิเตอร์ไบคิวบิกเป็น floating point จำนวน 16 จำนวนซึ่ง จะทำการคำนวณเก็บไว้โดยขั้นตอนวิธีการประมาณค่าในช่วงแบบไบคิวบิก

ค. สร้างเซตของโดเมน โดยกำหนดค่าของระยะการเลื่อนกรอบการดึงโดเมน domstep เป็นจำนวนเต็มที่มีมากกว่า 0 ตั้งต้นจากมุมบนซ้ายของภาพ ทำการดึงภาพขนาดเท่าขนาดโดเมนที่กำหนดออกมา จากนั้นเลื่อนกรอบไปทางซ้ายขึ้นละ dom_step พร้อมกับดึงภาพออกมา และย่อขนาดลงครึ่งหนึ่ง ทำต่อเนื่องจนกระทั่งกรอบเลื่อนไปสู่ทางขวามือ จึงเพิ่มค่าทางแนวตั้งเท่ากับ domstep ย้อนไปตั้งต้นใหม่ทางซ้ายมือและทำซ้ำแบบเดิม จนกระทั่งเลื่อนลงมาด้านล่างสุด ซึ่งเซตของโดเมนนี้จะมีขนาดเท่ากับ

$$\text{floor}(1+(\text{imagewidth}-\text{blockwidth})/\text{dom_step}) \times \text{floor}(1+(\text{imageheight}-\text{blockheight})/\text{dom_step})$$



ภาพที่ 34 แสดงโดเมนทั้งหมดที่ได้จากการดึงออกมาจากส่วนต่างๆของภาพ

ง. พิจารณาข้อมูลจุดภาพเรียงตัวเป็นเวกเตอร์หนึ่งมิติ และทำการแปลงเวกเตอร์ด้วยตัวดำเนินการนอर्मลไลซ์โปรเจกชัน

จ. สร้างเคทีรีของโดเมนบล็อก โดยอาศัยขั้นตอนวิธีของ Cardinal ตามขั้นตอนดังนี้

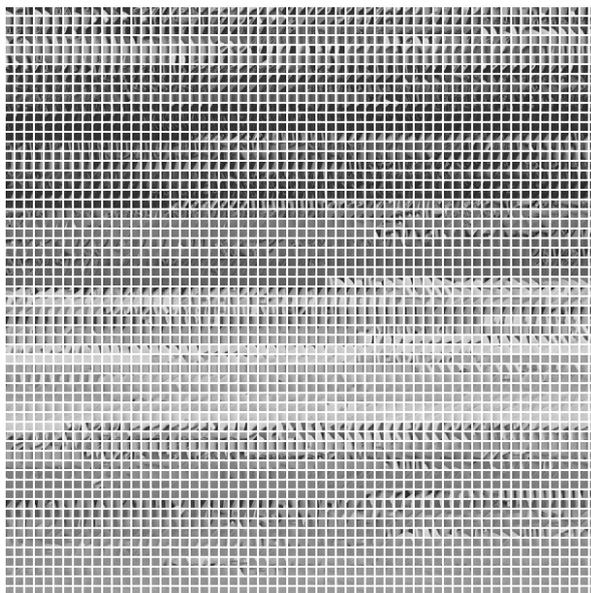
1) กำหนดให้ $\{r_i \in \mathbb{R}^k \mid i = 1 \dots n\}$ เป็นเซตของ พีเจอร์เวกเตอร์ที่ต้องการจะสร้างเคทีรี

$$2) \text{ คำนวณค่าของจุดศูนย์กลางของกลุ่ม } g = \frac{1}{n} \sum_{i=1}^n r_i$$

3) สำหรับแต่ละ $i = 1 \dots n$ คำนวณพิกัดโปรเจกชันของเวกเตอร์ r_i บนทรงกลมหนึ่งหน่วยจุดศูนย์กลางที่ g โดยใช้สูตร $s_i = \frac{r_i - g}{\|r_i - g\|}$

4) ระบายที่จะใช้ตัดเพื่อแบ่งกลุ่มคือระบายที่ผ่านจุด g และตั้งฉากกับเวกเตอร์ $w = \frac{1}{n} \sum_{i=1}^n s_i$

5) ทำซ้ำข้อ 1. - 4. กับข้อมูลทั้งสองด้านของระนาบ จนกระทั่งมีเวกเตอร์เหลืออยู่ในแต่ละด้านไม่เกินจำนวนที่กำหนดไว้



ภาพที่ 35 แสดงการเรียงตัวใหม่ของโดเมนที่โหนดใบบนหลังกระบวนการสร้างเคดิตรี

ฉ. ทำซ้ำข้อ 3 – 5 กับโดเมนบล็อกทุกขนาดที่กำหนด

ช. ค้นหาโดเมนที่เหมาะสมที่สุดสำหรับแต่ละเรนจ์บล็อก โดยเริ่มจากบล็อกขนาดใหญ่ที่สุดก่อน สำหรับแต่ละเรนจ์บล็อก ทำการหมุนตามเข็มนาฬิกาด้วยระดับขั้น θ ตามความละเอียดที่กำหนดโดยจำนวนบิตของการหมุน จากนั้นนำไปแปลงด้วยตัวดำเนินการนอมัลไลซ์โปรเจกชัน ใช้เวกเตอร์ที่ได้เป็นคีย์ในการค้นในเคดิตรี โดยนำไปหาผลคูณจุด (dot product) กับค่าประจำโหนดเริ่มที่โหนดราก หากค่าที่ได้มีค่าน้อยกว่าค่า μ ของเคดิตรีโหนดให้เลื่อนลงมาทางซ้าย มิฉะนั้นให้เลื่อนไปทางขวา ทำเช่นนี้จนกว่าจะถึงโหนดใบบน จากนั้นเริ่มทำการค้นแบบโดยลำดับ (sequential search) เริ่มจาก โหนดนั้น ไปตามลำดับ โดเมนเป็นจำนวนเท่ากับขนาดของกลุ่มของโดเมนกลุ่มนั้น การค้นหาโดยการหมุนโดเมนกลับทิศในระดับมุม $256 - \theta$ และนำไปทดสอบความคล้ายภายใต้การปรับค่าพารามิเตอร์ให้ผลต่างกำลังสองต่ำที่สุดเทียบกับเรนจ์บล็อกนั้นๆ เปรียบเทียบความคล้ายของเรนจ์ที่หมุนในมุมต่างๆกับโดเมนที่ค้นได้สำหรับแต่ละมุม และเลือกใช้โดเมนและมุมการหมุนที่ให้ระดับความคล้ายสูงที่สุดเป็นโดเมนของเรนจ์บล็อกนั้น

ซ. จัดเก็บพารามิเตอร์ของฟังก์ชันลงไฟล์ และบีบอัดข้อมูลทั้งหมดอีกครั้งรอบด้วยการเข้ารหัสแบบฮัฟฟ์แมน

2.3.4 การเก็บข้อมูลลงไฟล์

เพื่อให้การเก็บข้อมูลทำได้โดยมีประสิทธิภาพ การบันทึกค่าลงไฟล์จะทำการระดับบิต ค่าพารามิเตอร์ที่เป็นจำนวนจริงจะต้องถูกควอนไทซ์เพื่อลดระดับขั้นเสียก่อน รูปแบบของการจัดเก็บข้อมูลเป็นดังตารางที่ 2 และ 3

ตารางที่ 2 ส่วนเฮดเดอร์ของไฟล์

ขนาด (บิต)	ค่าที่เก็บ
10	ความกว้างของภาพ
10	ความสูงของภาพ
5	ระยะห่างในการดึงโดเมนบล็อกในแนวตั้ง
5	ระยะห่างในการดึงโดเมนบล็อกในแนวนอน
3	ระบุขนาดพาร์ติชันที่เล็กที่สุด
3	ระบุขนาดพาร์ติชันที่ใหญ่ที่สุด
17	กำหนดความละเอียดของการหมุนในพาร์ติชันแต่ละขนาด
4	กำหนดจำนวนบิตที่ใช้ระบุพารามิเตอร์ความเปรียบต่าง
4	กำหนดจำนวนบิตที่ใช้ระบุพารามิเตอร์ความสว่าง

ตารางที่ 3 ส่วนข้อมูลสำหรับแต่ละพาร์ติชัน ซึ่งจะระบุโดเมนและพารามิเตอร์ของฟังก์ชัน

ขนาด (บิต)	ค่าที่เก็บ
1	บิตสำหรับบอกตัวถอดรหัสว่ามีการแตกควอดทรีหรือไม่ (0=ไม่แตก, 1=แตก) บิตนี้จะไม่ถูกบันทึกถ้าขนาดของพาร์ติชันมีค่าเท่ากับขนาดเล็กที่สุดที่กำหนดในเฮดเดอร์
$\text{ceiling}(\log_2(\text{จำนวนแถวของโดเมน} \times \text{จำนวนหลักของโดเมน}))$	หมายเลขของโดเมนที่สัมพันธ์กับเรนจ์บล็อกนี้ ข้อมูลชุดนี้จะไม่ถูกบันทึกถ้าพารามิเตอร์การปรับความเปรียบต่างของพาร์ติชันนี้เท่ากับ 0
5	พารามิเตอร์ของการปรับความเปรียบต่าง (contrast)
7	พารามิเตอร์ของการปรับความสว่าง (brightness)
เป็นได้ตั้งแต่ 1 – 8 (กำหนดในเฮดเดอร์)	พารามิเตอร์ของการหมุน ข้อมูลชุดนี้จะไม่ถูกบันทึกถ้าพารามิเตอร์การปรับความเปรียบต่างของพาร์ติชันนี้เท่ากับ 0

2.3.5 ขั้นตอนการทดลอง

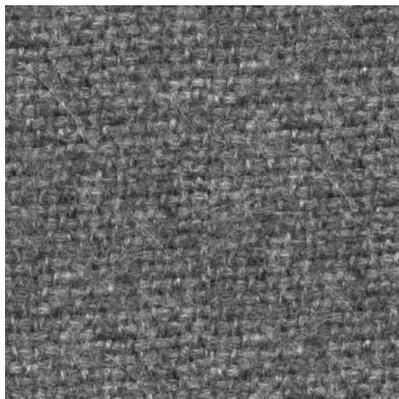
ก. เลือกภาพระดับสีเทาที่จะใช้ในการทดลอง ประกอบด้วยภาพ Lenna และ Peppers ซึ่งเป็นภาพถ่ายที่มีลักษณะของโทนีสตอเนื่อง , Frabric และ Rock ซึ่งเป็นภาพพื้นผิว , Colonel ซึ่งเป็นภาพลายเส้น และ France ซึ่งเป็นภาพกราฟฟิคสังเคราะห์



Lenna.pgm (512×512)



Peppers.pgm (512×512)



Fabric.pgm (512×512)



Rock.pgm (512×512)



Colonel.pgm (512×512)



France.pgm (512×512)

ภาพที่ 36 ภาพที่ใช้ในการทดลอง

ข. ทดลองบีบอัดภาพแต่ละชนิดด้วยเงื่อนไขต่างๆกัน

1) การทดลองที่ 1 : ทดสอบผลของขนาดของกลุ่มโหนดใกล้เคียงที่สุด ที่มีต่อคุณภาพของภาพ เพื่อหาค่าของขนาดกลุ่มที่เหมาะสมที่จะนำไปใช้งานต่อไป

2) การทดลองที่ 2 : เพื่อทดสอบผลของความละเอียดของการหมุนบล็อก ที่มีต่อคุณภาพของภาพ และอัตราส่วนการบีบอัด ใช้ควอดทรี 4 ระดับ 32-16-8-4 และตั้งระยะเวลาเลื่อนกรอบโดเมนเป็น 4 รันค่าความผิดพลาดกำลังสองเฉลี่ยต่อจุดที่ยอมรับได้เป็น 4, 7, 10, 14, 20, 30, 43, 60 เลือกขนาดของกลุ่มโหนดใกล้เคียงที่สุดตามการทดลองที่ 1 และเลือกใช้โหมดการหมุนดังต่อไปนี้

ตารางที่ 4 แสดงความหมายของแต่ละโหมดการหมุน

ชื่อโหมด	จำนวนบิตที่ใช้อ้างอิง / รูปแบบการหมุนทั้งหมด / ชั้น การหมุน (องศา)			
	พาร์ติชัน 32×32	พาร์ติชัน 16×16	พาร์ติชัน 8×8	พาร์ติชัน 4×4
1111*	1 / 2 / 180	1 / 2 / 180	1 / 2 / 180	1 / 2 / 180
2222*	2 / 4 / 90	2 / 4 / 90	2 / 4 / 90	2 / 4 / 90
3332	3 / 8 / 45	3 / 8 / 45	3 / 8 / 45	2 / 4 / 90
3333	3 / 8 / 45	3 / 8 / 45	3 / 8 / 45	3 / 8 / 45
4443	4 / 16 / 22.5	4 / 16 / 22.5	4 / 16 / 22.5	3 / 8 / 45
4444	4 / 16 / 22.5	4 / 16 / 22.5	4 / 16 / 22.5	4 / 16 / 22.5
5432	5 / 32 / 11.25	4 / 16 / 22.5	3 / 8 / 45	2 / 4 / 90
5555	5 / 32 / 11.25	5 / 32 / 11.25	5 / 32 / 11.25	5 / 32 / 11.25
6542	6 / 64 / 5.625	5 / 32 / 11.25	4 / 16 / 22.5	2 / 4 / 90
6543	6 / 64 / 5.625	5 / 32 / 11.25	4 / 16 / 22.5	3 / 8 / 45
6666	6 / 64 / 5.625	6 / 64 / 5.625	6 / 64 / 5.625	6 / 64 / 5.625

หมายเหตุ * : โหมด 1111 และ 2222 เป็นโหมดของขั้นตอนวิธีเก่า ส่วนโหมดที่เหลือพัฒนาขึ้นด้วยขั้นตอนวิธีที่เสนอในงานวิจัยนี้

3) การทดลองที่ 3 เพื่อวัดเวลาที่ใช้ในการบีบอัดภาพสำหรับแต่ละโหมดการหมุน ใช้ควอดทรี 3 ระดับ (16-8-4) และ 4 ระดับ (32-16-8-4) และตั้งระยะการเลื่อนกรอบโดเมนเป็น 4 ค่าความผิดพลาดกำลังสองเฉลี่ยต่อจุดที่ยอมรับได้เป็น 30 และเลือกโหมดการหมุนตามการทดลองที่ 2

4) การทดลองที่ 4 : วัดคุณภาพของภาพ และอัตราส่วนการบีบอัดเปรียบเทียบกับวิธีการบีบอัดแบบเจบีค โดยเลือกโหมดการหมุนให้แนวโน้มโดยทั่วไปที่ดีที่สุด และใช้ตัวเลือกอื่นตามการทดลองที่ 2

ก. การวัดผลใช้เกณฑ์ต่อไปนี้

1) PSNR (Peak Signal-To-Noise Ratio) ของภาพที่ถอดรหัส เปรียบเทียบกับภาพต้นฉบับการที่ค่า PSNR ยิ่งมีค่ามากแสดงว่าภาพยิ่งใกล้เคียงภาพต้นฉบับ ค่า PSNR กำหนดได้จากสูตร

$$\text{PSNR} = 20 \log_{10} \left(\frac{255}{\text{RMS}} \right)$$

2) อัตราส่วนการบีบอัด ซึ่งมีค่าเท่ากับ ขนาดของไฟล์ที่ไม่บีบอัด / ขนาดของไฟล์ที่บีบอัด

ง. วิเคราะห์ความสัมพันธ์ระหว่าง PSNR กับ อัตราส่วนการบีบอัดของแต่ละชุดพารามิเตอร์สำหรับแต่ละรูปแบบของภาพ

ผลการทดลอง

1. การทดลองที่ 1 ทดสอบผลของขนาดของกลุ่มโหนดใกล้เคียงที่สุด ที่มีต่อคุณภาพของภาพ เพื่อหาค่าของขนาดกลุ่มที่เหมาะสมที่สุด

ตารางที่ 5 ผลการทดลองของการทดลองที่ 1

โหนด การ หมุน	ค่าความ ผิดพลาด ที่รับได้	ค่า PSNR เมื่อกำหนดขนาดของกลุ่มโหนดใกล้เคียงที่สุดที่ทำการทดสอบเป็น									
		1	2	4	8	12	16	20	24	28	32
2222	4	30.58	32.32	34.17	35.31	35.52	35.78	35.87	36.03	36.06	36.12
	7	30.54	32.25	34.06	35.16	35.35	35.59	35.68	35.82	35.85	35.91
	10	30.48	32.16	33.91	34.94	35.13	35.36	35.44	35.58	35.60	35.65
	14	30.40	32.02	33.70	34.69	34.86	35.08	35.16	35.28	35.31	35.35
4444	4	35.47	35.66	36.04	36.24	36.41	36.50	36.55	36.60	36.65	36.68
	7	35.31	35.48	35.84	36.02	36.18	36.26	36.31	36.35	36.40	36.42
	10	35.10	35.26	35.59	35.75	35.90	35.98	36.03	36.07	36.11	36.13
	14	34.83	34.97	35.30	35.46	35.59	35.66	35.69	35.73	35.78	35.80
6666	4	36.41	36.51	36.68	36.80	36.87	36.94	36.96	37.00	37.05	37.07
	7	36.18	36.27	36.42	36.53	36.59	36.65	36.67	36.70	36.75	36.77
	10	35.92	36.01	36.15	36.23	36.30	36.36	36.37	36.40	36.44	36.47
	14	35.59	35.67	35.81	35.90	35.96	36.00	36.03	36.05	36.09	36.11

การทดลองนี้แสดงให้เห็นว่า ขั้นตอนวิธีการค้นหาด้วยเคดิทรีในลักษณะนี้ จะไม่ได้ให้ค่าโดเมนที่ดีที่สุดออกมาในการค้นหาครั้งเดียว ทั้งนี้เนื่องจากเหตุผลสองอย่าง ประการแรก การค้นหาในเคดิทรีจะพบโดเมนที่ดีที่สุดภายใต้ค่าพารามิเตอร์ที่เป็นจำนวนจริง แต่เนื่องจากการเข้ารหัส จะต้องมีการควอนไทซ์ค่าพารามิเตอร์เพื่อเพิ่มอัตราการบีบอัด โดเมนที่เลือกขึ้นมาจึงอาจจะไม่ใช่โดเมนที่ดีที่สุดภายใต้พารามิเตอร์ที่ควอนไทซ์แล้ว และประการที่สอง การหมุนภาพบิดเบือนจะมีการเกิดความผิดพลาดขึ้นอย่างไม่อาจหลีกเลี่ยงได้ และด้วยลักษณะของการจับคู่โดเมนและเรนจ์นั้น ในขั้นตอนของการเข้ารหัสที่เป็นการนำเรนจ์บล็อกไปเปรียบเทียบกับโดเมน แต่ในขั้นตอนการถอดรหัส จะกระทำกลับทิศทาง คือเป็นการนำโดเมนบล็อกไปเปรียบเทียบกับเรนจ์ ด้วยเหตุนี้จึงทำให้โดเมนที่ค้นพบในครั้งแรกอาจเป็นเพียงโดเมนที่ใกล้เคียงแต่อาจไม่ใช่โดเมนที่ดีที่สุดเสมอไป

ด้วยเหตุนี้ จึงต้องการทำการค้นหาแบบ m โหนดใกล้เคียงที่สุดเพื่อหากลุ่มโหนดที่ดีที่สุดออกมาจำนวนหนึ่ง และทำการทดสอบอีกครั้งในสภาวะที่เหมือนกับการถอดรหัส เพื่อตรวจสอบความเหมาะสมของโดเมน ผลการทดลองพบว่าเมื่อเพิ่มขนาดของ m ความผิดพลาดของการจับคู่โดเมนและเรนจ์จะยิ่งน้อยลง และส่งผลสะท้อนไปถึง PSNR ของภาพที่ถอดรหัสที่สูงขึ้น

2. การทดลองที่ 2 ทดสอบผลของความละเอียดของการหมุนบล็อก ที่มีต่อคุณภาพของภาพ และ อัตราส่วนการบีบอัด

ตารางที่ 6 ผลการทดลองของการทดลองที่ 2 ในการบีบอัดภาพ Lenna

โหมดการ หมุน		ค่าพารามิเตอร์ความผิดพลาดที่ยอมรับได้							
		4	7	10	14	20	30	43	60
1111	Ratio	5.11	6.32	7.44	8.67	9.96	11.81	13.89	16.43
	PSNR	34.66	34.53	34.36	34.14	33.86	33.4	32.85	32.20
	#Func.	15334	12394	10501	8992	7816	6577	5572	4687
2222	Ratio	5.05	6.32	7.50	8.73	10.04	12.01	14.27	16.77
	PSNR	35.78	35.59	35.36	35.08	34.74	34.15	33.5	32.79
	#Func.	15190	12163	10252	8797	7654	6388	5359	4549
3332	Ratio	5.13	6.47	7.68	8.85	10.19	12.17	14.63	17.26
	PSNR	35.77	35.59	35.35	35.09	34.77	34.17	33.49	32.76
	#Func.	14941	11860	9988	8665	7510	6277	5206	4396
3333	Ratio	5.03	6.36	7.55	8.70	10.03	11.98	14.4	17.02
	PSNR	36.22	36.01	35.74	35.45	35.10	34.44	33.72	32.96
	#Func.	14941	11860	9988	8665	7510	6277	5206	4396
4442	Ratio	5.19	6.56	7.77	8.95	10.35	12.38	15.03	17.75
	PSNR	35.77	35.59	35.36	35.08	34.78	34.19	33.44	32.60
	#Func.	14782	11665	9835	8521	7360	6130	5023	4234
4444	Ratio	4.98	6.33	7.51	8.66	10.02	12.02	14.61	17.29
	PSNR	36.50	36.26	35.98	35.66	35.30	34.62	33.81	32.89
	#Func.	14782	11665	9835	8521	7360	6130	5023	4234
5432	Ratio	5.14	6.50	7.71	8.88	10.24	12.23	14.78	17.61
	PSNR	35.77	35.59	35.35	35.09	34.78	34.17	33.45	32.56
	#Func.	14929	11791	9934	8623	7468	6241	5140	4294
5555	Ratio	4.76	6.07	7.17	8.33	9.66	11.59	14.13	16.76
	PSNR	36.74	36.48	36.19	35.86	35.46	34.75	33.92	32.92
	#Func.	14593	11476	9700	8353	7195	5983	4885	4102
6542	Ratio	5.19	6.58	7.81	8.96	10.39	12.4	15.06	17.79
	PSNR	35.77	35.58	35.36	35.11	34.78	34.20	33.49	32.61
	#Func.	14782	11617	9787	8503	7312	6097	4990	4192
6543	Ratio	5.08	6.47	7.68	8.81	10.22	12.2	14.81	17.53
	PSNR	36.22	36.00	35.75	35.47	35.11	34.48	33.73	32.80
	#Func.	14782	11617	9787	8503	7312	6097	4990	4192
6666	Ratio	4.61	5.91	6.96	8.11	9.40	11.37	13.96	16.53
	PSNR	36.94	36.65	36.36	36.00	35.58	34.83	34.00	33.00
	#Func.	14500	11293	958	8224	7081	5848	4750	3997

หมายเหตุ Ratio = อัตราส่วนการบีบอัด

PSNR = Peak Signal-To-Noise Ratio

#Func. = จำนวนฟังก์ชันที่ใช้เข้ารหัสภาพ ซึ่งมีจำนวนเท่ากับพาริตีชัน

ผลการทดลองแสดงให้เห็นว่า PSNR และอัตราส่วนการบีบอัดจะมีทิศทางที่สวนทางกันเสมอ กล่าวคือเมื่อ PSNR มีค่ามากขึ้นอัตราส่วนการบีบอัดจะน้อยลง ในทำนองเดียวกันเมื่ออัตราส่วนการบีบอัดเพิ่มมากขึ้น PSNR ก็จะน้อยลง เมื่อกำหนดค่าความผิดพลาดที่ยอมรับได้คงที่ การบีบอัดที่มีรูปแบบการหมุนมากขึ้นในทุกระดับพาร์ติชันจะให้ค่า PSNR ที่สูงขึ้นเสมอ ในขณะที่อัตราส่วนการบีบอัดมีแนวโน้มที่ไม่แน่นอน อาจจะมีมากขึ้นหรือน้อยลงก็ได้ นอกจากนี้ การเพิ่มค่าความผิดพลาดที่ยอมรับได้จะส่งผลให้ PSNR ต่ำลง ในขณะที่อัตราส่วนการบีบอัดเพิ่มขึ้น

ตารางที่ 7 ผลการทดลองของการทดลองที่ 2 ในการบีบอัดภาพ Peppers

โหมดการ หมุน		ค่าพารามิเตอร์ความผิดพลาดที่ยอมรับได้							
		4	7	10	14	20	30	43	60
1111	Ratio	5.51	6.72	7.76	9.03	10.69	12.97	15.48	18.05
	PSNR	32.56	32.46	32.35	32.21	31.98	31.64	31.21	30.70
	IFS	14056	11530	9976	8572	7222	5926	4945	4228
2222	Ratio	5.48	6.77	7.84	9.16	10.99	13.34	15.82	18.56
	PSNR	33.83	33.70	33.56	33.35	33.08	32.68	32.25	31.56
	IFS	13810	11200	9661	8263	6877	5638	4729	4009
3332	Ratio	5.59	6.94	8.01	9.37	11.14	13.58	16.03	19.01
	PSNR	33.82	33.70	33.56	33.36	33.09	32.69	32.27	31.51
	IFS	13561	10921	9466	8071	6766	5518	4657	3898
3333	Ratio	5.51	6.85	7.91	9.27	11.04	13.51	15.94	18.97
	PSNR	33.93	33.80	33.65	33.44	33.17	32.75	32.32	31.56
	IFS	13561	10921	9466	8071	6766	5518	4657	3898
4442	Ratio	5.66	7.04	8.07	9.50	11.29	13.75	16.19	19.40
	PSNR	33.82	33.69	33.56	33.38	33.10	32.67	32.22	31.51
	IFS	13387	10729	9340	7903	6616	5410	4567	3784
4444	Ratio	5.49	6.86	7.89	9.31	11.10	13.55	15.98	19.21
	PSNR	34.00	33.85	33.69	33.51	33.22	32.76	32.30	31.56
	IFS	13387	10729	9340	7903	6616	5410	4567	3784
5432	Ratio	5.59	6.96	8.02	9.41	11.19	13.73	16.08	19.15
	PSNR	33.82	33.70	33.56	33.36	33.09	32.65	32.29	31.53
	IFS	13546	10873	9436	8023	6724	5446	4621	3853
5555	Ratio	5.26	6.59	7.56	8.99	10.80	13.16	15.65	18.75
	PSNR	34.14	34.00	33.87	33.63	33.32	32.84	32.37	31.61
	IFS	13168	10537	9199	7723	6424	5254	4405	3652
6542	Ratio	5.66	7.06	8.09	9.55	11.34	13.84	16.30	19.42
	PSNR	33.82	33.69	33.56	33.37	33.09	32.68	32.22	31.53
	IFS	13363	10687	9307	7843	6568	5356	4510	3754
6543	Ratio	5.58	6.97	8.00	9.46	11.26	13.75	16.24	19.38
	PSNR	33.93	33.80	33.65	33.45	33.17	32.74	32.27	31.57
	IFS	13363	10687	9307	7843	6568	5356	4510	3754
6666	Ratio	5.12	6.41	7.39	8.82	10.61	13.00	15.46	18.40
	PSNR	34.46	34.29	34.13	33.85	33.56	33.05	32.55	31.78
	IFS	12997	10396	9037	7573	6283	5125	4288	3577

ตารางที่ 8 ผลการทดลองของการทดลองที่ 2 ในการบีบอัดภาพ Rock

โหมดการ หมุน		ค่าพารามิเตอร์ความผิดพลาดที่ยอมรับได้							
		4	7	10	14	20	30	43	60
1111	Ratio	4.74	4.76	4.77	4.79	4.82	4.86	4.92	5.00
	PSNR	34.19	34.19	34.18	34.17	34.15	34.09	33.98	33.81
	IFS	16291	16219	16174	16087	16009	15850	15631	15400
2222	Ratio	4.60	4.62	4.63	4.66	4.68	4.73	4.78	4.86
	PSNR	37.96	37.94	37.93	37.89	37.85	37.69	37.42	36.99
	IFS	16285	16213	16162	16072	16000	15823	15610	15364
3332	Ratio	4.61	4.62	4.64	4.66	4.68	4.73	4.79	4.87
	PSNR	37.96	37.94	37.93	37.89	37.84	37.67	37.40	36.97
	IFS	16279	16207	16153	16060	15979	15781	15577	15328
3333	Ratio	4.52	4.53	4.55	4.57	4.59	4.64	4.70	4.77
	PSNR	41.56	41.53	41.49	41.41	41.30	40.91	40.35	39.53
	IFS	16279	16207	16153	16060	15979	15781	15577	15328
4442	Ratio	4.61	4.63	4.64	4.67	4.69	4.74	4.81	4.88
	PSNR	37.96	37.95	37.93	37.89	37.83	37.65	37.37	36.93
	IFS	16273	16201	16138	16042	15952	15745	15523	15268
4444	Ratio	4.42	4.44	4.46	4.48	4.50	4.56	4.62	4.69
	PSNR	42.05	42.02	41.97	41.87	41.73	41.28	40.65	39.73
	IFS	16273	16201	16138	16042	15952	15745	15523	15268
5432	Ratio	4.61	4.62	4.64	4.66	4.68	4.74	4.79	4.87
	PSNR	37.96	37.94	37.93	37.89	37.84	37.66	37.41	36.99
	IFS	16279	16207	16153	16060	15979	15775	15577	15328
5555	Ratio	4.31	4.33	4.34	4.37	4.40	4.45	4.51	4.59
	PSNR	42.51	42.47	42.40	42.30	42.10	41.61	40.91	39.78
	IFS	16270	16198	16126	16033	15913	15706	15478	15175
6542	Ratio	4.61	4.63	4.64	4.67	4.69	4.74	4.81	4.88
	PSNR	37.96	37.95	37.93	37.89	37.83	37.65	37.37	36.93
	IFS	16273	16201	16138	16042	15952	15745	15523	15268
6543	Ratio	4.52	4.54	4.55	4.58	4.60	4.65	4.71	4.79
	PSNR	41.56	41.53	41.49	41.40	41.28	40.87	40.29	39.45
	IFS	16273	16201	16138	16042	15952	15745	15523	15268
6666	Ratio	4.19	4.21	4.22	4.25	4.28	4.33	4.38	4.48
	PSNR	43.28	43.24	43.15	43.01	42.77	42.19	41.46	40.03
	IFS	16264	16192	16114	16006	15883	15673	15463	15106

ตารางที่ 9 ผลการทดลองของการทดลองที่ 2 ในการบีบอัดภาพ Fabric

โหมดการ หมุน		ค่าพารามิเตอร์ความผิดพลาดที่ยอมรับได้							
		4	7	10	14	20	30	43	60
1111	Ratio	4.71	4.71	4.71	4.71	4.71	4.71	4.71	4.72
	PSNR	33.09	33.09	33.09	33.09	33.09	33.09	33.08	33.07
	IFS	16384	16384	16384	16384	16384	16384	16375	16345
2222	Ratio	4.58	4.58	4.58	4.58	4.58	4.58	4.59	4.60
	PSNR	37.07	37.07	37.07	37.07	37.07	37.07	37.05	36.95
	IFS	16384	16384	16384	16384	16384	16384	16372	16306
3332	Ratio	4.58	4.58	4.58	4.58	4.58	4.58	4.59	4.61
	PSNR	37.07	37.07	37.07	37.07	37.07	37.07	37.05	36.92
	IFS	16384	16384	16384	16384	16384	16384	16369	16285
3333	Ratio	4.58	4.58	4.58	4.58	4.58	4.58	4.59	4.60
	PSNR	37.07	37.07	37.07	37.07	37.07	37.07	37.05	36.95
	IFS	16384	16384	16384	16384	16384	16384	16372	16306
4442	Ratio	4.58	4.58	4.58	4.58	4.58	4.58	4.59	4.61
	PSNR	37.07	37.07	37.07	37.07	37.07	37.07	37.05	36.92
	IFS	16384	16384	16384	16384	16384	16384	16369	16285
4444	Ratio	4.49	4.49	4.49	4.49	4.49	4.49	4.49	4.52
	PSNR	39.33	39.33	39.33	39.33	39.33	39.33	39.31	39.07
	IFS	16384	16384	16384	16384	16384	16384	16369	16285
5432	Ratio	4.58	4.58	4.58	4.58	4.58	4.58	4.59	4.60
	PSNR	37.07	37.07	37.07	37.07	37.07	37.07	37.06	37
	IFS	16384	16384	16384	16384	16384	16384	16375	16342
5555	Ratio	4.49	4.49	4.49	4.49	4.49	4.49	4.49	4.51
	PSNR	39.33	39.33	39.33	39.33	39.33	39.33	39.31	39.12
	IFS	16384	16384	16384	16384	16384	16384	16372	16306
6542	Ratio	4.40	4.40	4.40	4.40	4.40	4.40	4.40	4.42
	PSNR	39.76	39.76	39.76	39.76	39.76	39.76	39.73	39.46
	IFS	16384	16384	16384	16384	16384	16384	16369	16285
6543	Ratio	4.26	4.26	4.26	4.26	4.26	4.26	4.26	4.29
	PSNR	40.18	40.18	40.18	40.18	40.18	40.18	40.12	39.72
	IFS	16384	16384	16384	16384	16384	16384	16360	16237
6666	Ratio	4.13	4.13	4.13	4.13	4.13	4.13	4.14	4.17
	PSNR	40.86	40.86	40.86	40.86	40.86	40.86	40.8	40.22
	IFS	16384	16384	16384	16384	16384	16384	16357	16201

ตารางที่ 10 ผลการทดลองของการทดลองที่ 2 ในการบีบอัดภาพ Colonel

โหมดการ หมุน		ค่าพารามิเตอร์ความผิดพลาดที่ยอมรับได้							
		4	7	10	14	20	30	43	60
1111	Ratio	20.33	21.73	22.86	24.07	25.57	28.20	32.05	36.77
	PSNR	40.99	40.86	40.66	40.23	39.92	39.23	38.05	36.85
	IFS	4297	4003	3796	3592	3376	3049	2662	2308
2222	Ratio	20.48	21.73	22.80	24.01	25.71	28.21	32.53	37.62
	PSNR	41.94	41.80	41.55	41.05	40.62	39.88	38.58	37.25
	IFS	4150	3880	3694	3481	3232	2920	2512	2158
3332	Ratio	20.79	22.15	23.12	24.42	26.01	28.58	32.82	38.02
	PSNR	41.95	41.76	41.55	41.09	40.66	39.86	38.61	37.07
	IFS	4123	3835	3667	3448	3220	2899	2500	2137
3333	Ratio	21.27	22.71	23.61	24.96	26.52	29.15	33.15	38.31
	PSNR	42.04	41.84	41.63	41.16	40.72	39.91	38.65	37.09
	IFS	4123	3835	3667	3448	3220	2899	2500	2137
4442	Ratio	20.94	22.32	23.33	24.45	26.07	28.83	33.12	38.53
	PSNR	41.97	41.77	41.53	41.11	40.69	39.72	38.39	36.86
	IFS	4069	3790	3619	3427	3199	2857	2443	2080
4444	Ratio	20.78	22.17	23.16	24.29	25.88	28.63	32.85	37.91
	PSNR	42.22	42.01	41.75	41.31	40.88	39.88	38.52	36.95
	IFS	4069	3790	3619	3427	3199	2857	2443	2080
5432	Ratio	20.88	22.34	23.35	24.48	26.09	28.80	33.23	38.63
	PSNR	41.97	41.78	41.51	41.13	40.69	39.74	38.27	36.79
	IFS	4099	3805	3625	3439	3205	2875	2458	2092
5555	Ratio	20.15	21.58	22.66	23.85	25.66	28.23	33.03	38.54
	PSNR	42.30	42.08	41.78	41.35	40.89	39.89	38.40	36.89
	IFS	4036	3739	3553	3358	3097	2785	2320	1957
6542	Ratio	20.98	22.38	23.35	24.41	26.07	29.06	33.60	39.24
	PSNR	41.98	41.80	41.53	41.14	40.72	39.59	38.31	36.38
	IFS	4051	3763	3598	3415	3184	2821	2395	2029
6543	Ratio	21.51	22.96	23.83	24.96	26.56	29.53	33.92	39.46
	PSNR	42.07	41.88	41.61	41.22	40.78	39.64	38.34	36.40
	IFS	4051	3763	3598	3415	3184	2821	2395	2029
6666	Ratio	19.74	21.25	22.19	23.59	25.52	28.45	33.20	39.86
	PSNR	42.32	42.11	41.83	41.37	40.92	39.74	38.50	36.30
	IFS	3991	3697	3526	3295	3025	2677	2230	1837

ตารางที่ 11 ผลการทดลองของการทดลองที่ 2 ในการบีบอัดภาพ France

โหมดการ หมุน		ค่าพารามิเตอร์ความผิดพลาดที่ยอมรับได้							
		4	7	10	14	20	30	43	60
1111	Ratio	12.24	12.41	12.50	12.62	12.77	12.99	13.27	13.80
	PSNR	25.40	25.39	25.38	25.38	25.38	25.37	25.36	25.31
	IFS	7279	7153	7090	7015	6919	6787	6628	6313
2222	Ratio	11.04	11.24	11.40	11.53	11.67	11.95	12.28	12.76
	PSNR	26.38	26.37	26.37	26.36	26.35	26.33	26.31	26.26
	IFS	7195	7036	6925	6847	6760	6601	6415	6172
3332	Ratio	11.05	11.24	11.40	11.52	11.69	11.95	12.30	12.79
	PSNR	26.38	26.37	26.37	26.36	26.35	26.33	26.31	26.26
	IFS	7180	7027	6910	6832	6736	6580	6382	6142
3333	Ratio	11.05	11.24	11.39	11.52	11.66	11.94	12.29	12.76
	PSNR	26.38	26.37	26.37	26.36	26.35	26.33	26.31	26.26
	IFS	7195	7033	6922	6847	6754	6592	6400	6163
4442	Ratio	11.06	11.24	11.40	11.52	11.67	11.96	12.32	12.78
	PSNR	26.38	26.37	26.37	26.36	26.35	26.33	26.30	26.26
	IFS	7171	7027	6910	6829	6736	6568	6367	6136
4444	Ratio	11.48	11.65	11.81	11.93	12.09	12.37	12.70	13.16
	PSNR	26.56	26.55	26.55	26.54	26.53	26.51	26.47	26.43
	IFS	7171	7027	6910	6829	6736	6568	6367	6136
5432	Ratio	11.32	11.51	11.66	11.79	11.95	12.19	12.55	13.02
	PSNR	26.38	26.37	26.37	26.36	26.35	26.33	26.31	26.26
	IFS	7198	7048	6949	6865	6772	6625	6436	6196
5555	Ratio	11.47	11.66	11.81	11.94	12.06	12.34	12.67	13.12
	PSNR	26.56	26.55	26.55	26.54	26.53	26.52	26.49	26.44
	IFS	7195	7036	6925	6847	6760	6601	6415	6172
6542	Ratio	10.98	11.15	11.30	11.42	11.58	11.82	12.19	12.59
	PSNR	27.25	27.25	27.24	27.24	27.23	27.21	27.18	27.14
	IFS	7180	7027	6910	6832	6736	6580	6382	6142
6543	Ratio	10.55	10.69	10.86	10.99	11.18	11.45	11.87	12.16
	PSNR	28.50	28.50	28.49	28.50	28.48	28.44	28.40	28.37
	IFS	7156	7018	6901	6814	6700	6538	6256	6115
6666	Ratio	10.17	10.32	10.47	10.61	10.79	11.21	11.55	11.80
	PSNR	28.53	28.53	28.52	28.52	28.50	28.46	28.42	28.38
	IFS	7120	6985	6883	6793	6664	6418	6226	6088

3. การทดลองที่ 3 วัดเวลาที่ใช้ในการบีบอัดภาพสำหรับแต่ละโหมดการหมุน

ตารางที่ 10 แสดงให้เห็นว่าจำนวนของรูปแบบการหมุนที่ต้องทำการค้นหา มีผลต่อเวลาในการประมวลผล กล่าวคือยังมีรูปแบบการหมุนหลากหลายมากก็ยิ่งใช้เวลามากขึ้นตามไปด้วย และภาพต่างชนิดกันเวลาที่ใช้ในการบีบอัดก็อาจต่างกัน

ตารางที่ 12 ผลการทดลองของการทดลองที่ 3 เปรียบเทียบเวลาที่ใช้ในการบีบอัดของโหมดต่างๆ เมื่อใช้ควอดทรี 3 ระดับ

ภาพ	เวลาที่ใช้สำหรับแต่ละโหมด (วินาที)										
	111	222	332	333	432	442	444	542	543	555	666
Colonel	3.9	4.4	8.4	9.0	14.6	16.3	17.8	28.6	28.7	35.4	69.2
Fabric	6.5	7.9	14.6	18.3	21.0	27.8	38.9	41.0	44.6	79.6	160.0
France	5.3	6.3	11.0	12.3	17.2	20.6	24.5	33.8	35.1	49.6	100.0
Lenna	5.5	6.3	11.8	12.9	17.8	21.6	25.1	35.7	35.4	49.8	97.5
Peppers	5.5	6.3	11.5	12.4	17.8	21.5	24.5	34.0	34.9	47.3	92.0
Rock	6.5	7.9	14.6	18.2	21.0	27.9	38.6	41.0	44.3	78.4	157.2

ตารางที่ 13 ผลการทดลองของการทดลองที่ 3 เปรียบเทียบเวลาที่ใช้ในการบีบอัดของโหมดต่างๆ เมื่อใช้ควอดทรี 4 ระดับ

ภาพ	เวลาที่ใช้สำหรับแต่ละโหมด (วินาที)										
	1111	2222	3332	3333	4442	5432	4444	6542	6543	5555	6666
Colonel	14.7	15.3	22.1	22.7	35.3	46.8	37.1	84.6	85.2	65.4	120.5
Fabric	18.6	20.5	32.2	36.7	55.6	62.6	68.6	116.2	120.4	132.5	257.9
France	16.5	17.8	25.8	27.4	42.1	52.1	46.8	95.9	97.4	86.6	166.7
Lenna	17.7	18.6	27.8	29.2	45.6	56.2	49.6	102.1	105.8	89.9	169.3
Peppers	17.3	18.5	27.9	29.0	46.0	56.6	49.4	103.0	104.0	89.5	167.9
Rock	18.5	20.4	32.2	36.5	55.8	63.2	68.4	117.4	121.3	131.7	256.3

4. การทดลองที่ 4 วัดคุณภาพของภาพ และอัตราส่วนการบีบอัด เปรียบเทียบกับวิธีการบีบอัดภาพแบบเจบีค

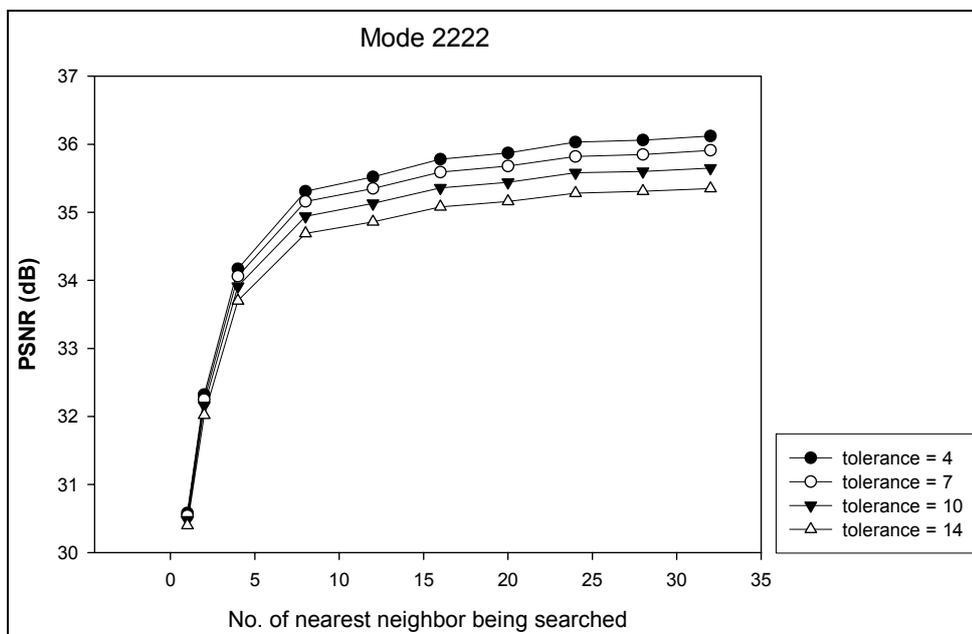
ผลการทดลองแสดงให้เห็นว่า PSNR และอัตราส่วนการบีบอัดจะมีทิศทางที่สวนทางกันเสมอทั้งสองรูปแบบการบีบอัด กล่าวคือเมื่อ PSNR มีค่ามากขึ้นอัตราส่วนการบีบอัดจะน้อยลง ในทำนองเดียวกันเมื่อ ถ้าอัตราส่วนการบีบอัดเพิ่มมากขึ้น PSNR ก็จะน้อยลง เห็นได้ชัดว่าไฟล์ Colonel, Fabric และ Rock การบีบอัดแบบแฟร็กทัลให้ PSNR ที่สูงกว่า ในขณะที่ไฟล์ France, Lenna และ Peppers ในระดับของอัตราส่วนการบีบอัดที่ใกล้เคียงกัน การบีบอัดแบบเจบีคจะให้ PSNR ที่สูงกว่า

ตารางที่ 14 ผลการทดลองของการทดลองที่ 4 เปรียบเทียบการบีบอัดแบบแฟร็กทัลกับเจบีค

ภาพ	ผลการวัด									
Colonel	Jpeg	Ratio	12.33	14.72	16.86	18.7	20.81	23.52	28.18	36.77
		PSNR	42.33	40.40	39.00	38.13	37.37	36.34	34.41	32.03
	fractal	Ratio	19.74	21.25	22.19	23.59	25.52	28.45	33.20	39.86
		PSNR	42.32	42.11	41.83	41.37	40.92	39.74	38.50	36.30
Fabric	Jpeg	Ratio	3.69	4.72	5.67	6.47	7.42	8.78	11.50	19.43
		PSNR	33.88	32.38	31.47	30.86	30.26	29.52	28.33	26.13
	fractal	Ratio	4.13	4.13	4.13	4.13	4.13	4.13	4.14	4.17
		PSNR	40.86	40.86	40.86	40.86	40.86	40.86	40.80	40.22
France	Jpeg	Ratio	6.77	8.13	9.31	10.30	11.51	13.26	16.16	24.17
		PSNR	37.65	35.00	33.38	32.29	31.27	30.09	28.49	26.06
	fractal	Ratio	10.17	10.32	10.47	10.61	10.79	11.21	11.55	11.80
		PSNR	28.53	28.53	28.52	28.52	28.5	28.46	28.42	28.38
Lenna	Jpeg	Ratio	6.91	8.94	10.88	12.54	14.52	17.30	22.06	32.72
		PSNR	38.54	37.33	36.45	35.81	35.13	34.28	32.96	30.41
	fractal	Ratio	4.61	5.91	6.96	8.11	9.40	11.37	13.96	16.53
		PSNR	36.94	36.65	36.36	36.00	35.58	34.83	34.00	33.00
Peppers	Jpeg	Ratio	7.07	8.95	10.75	12.29	14.17	16.76	21.04	30.4
		PSNR	40.14	38.88	37.87	37.09	36.23	35.21	33.59	30.46
	fractal	Ratio	5.12	6.41	7.39	8.82	10.61	13	15.46	18.4
		PSNR	34.46	34.29	34.13	33.85	33.56	33.05	32.55	31.78
Rock	Jpeg	Ratio	3.38	4.2	4.93	5.53	6.23	7.18	9.01	13.75
		PSNR	34.22	32.79	31.91	31.28	30.63	29.76	28.36	25.64
	fractal	Ratio	4.19	4.21	4.22	4.25	4.28	4.33	4.38	4.48
		PSNR	43.28	43.24	43.15	43.01	42.77	42.19	41.46	40.03

วิจารณ์ผลการทดลอง

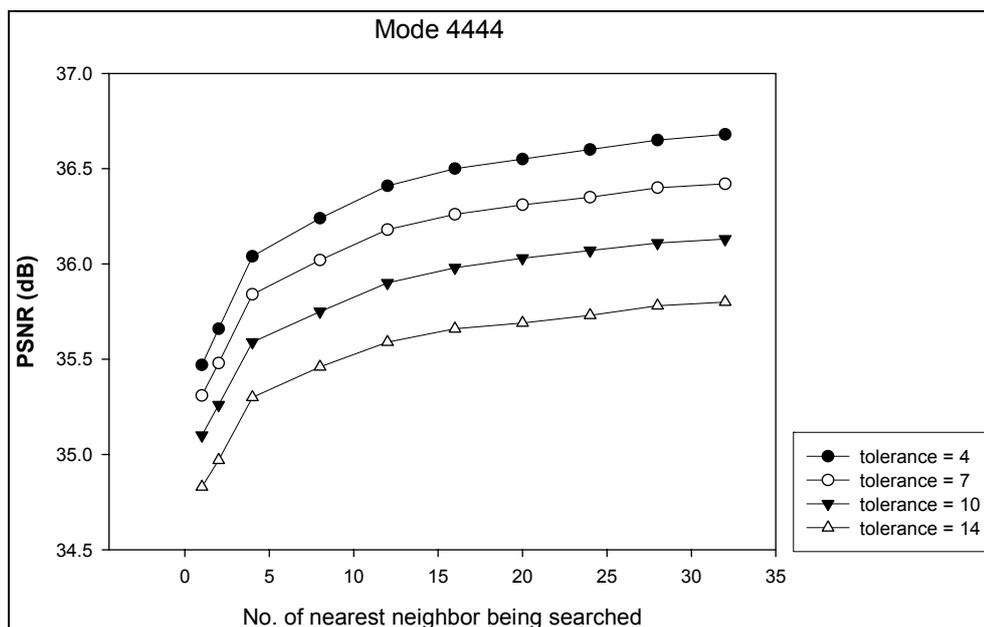
1. การทดลองที่ 1



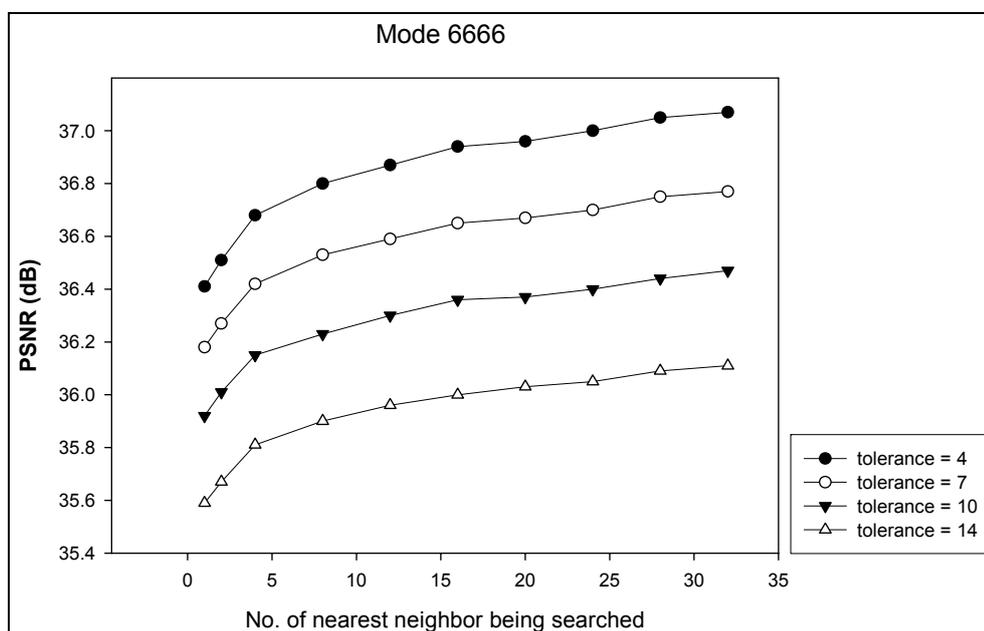
ภาพที่ 37 ความสัมพันธ์ระหว่างค่า m กับค่า PSNR ของโหมด 2222

จากภาพ 37 พบว่าเมื่อเพิ่มค่า m ค่า PSNR จะเพิ่มขึ้นอย่างรวดเร็วในช่วงแรก และจะค่อยๆ ลดระดับความชันลงเมื่อค่า m สูงถึงระดับหนึ่ง เป็นเพราะการเพิ่มจำนวนของกลุ่มโหนดใกล้เคียงที่สุดที่จะทำการเปรียบเทียบโดยละเอียดนั้น ทำให้มีโอกาสพบโหนดที่ยิ่งใกล้เคียงมากขึ้น ทั้งนี้เป็นไปตามเหตุผล 2 ประการที่ได้กล่าวมาแล้ว ประการแรกคือ การที่พารามิเตอร์ที่นำไปใช้เข้ารหัสไม่ใช่ค่าพารามิเตอร์ที่ดีที่สุดที่ได้จากการค้นพบ แต่เป็นพารามิเตอร์ที่ผ่านการควอนไทซ์จากค่าพารามิเตอร์ของโหนดที่ค้นได้ และ ประการที่สองคือ การที่การเข้ารหัสและถอดรหัส เป็นการดำเนินการที่กลับทิศทางกัน

เมื่อพิจารณารูปของโหมด 4444 และ 6666 ก็จะพบแนวโน้มในลักษณะเดียวกัน ในความเป็นจริงแล้ว ค่า m ยิ่งมากก็จะยิ่งเพิ่มคุณภาพของภาพ แต่การที่มีจำนวนโหนดให้เปรียบเทียบเป็นจำนวนมากนั้น จะยังต้องใช้เวลาดังนั้นค่า m ที่เหมาะสม จึงควรเป็นค่าที่ไม่มากจนเกินไป แต่อยู่ในระดับพอดีที่ค่าความชันของกราฟไม่ชันมากนัก ซึ่งหมายความว่า จุดนั้น ถึงจะเพิ่มค่า m ขึ้นอีกหน่อยก็ไม่ได้ส่งผลที่ดีขึ้นมากนัก เมื่อพิจารณาแล้ว จึงเลือกใช้ค่า $m = 16$ สำหรับการทดลองในขั้นต่อไป

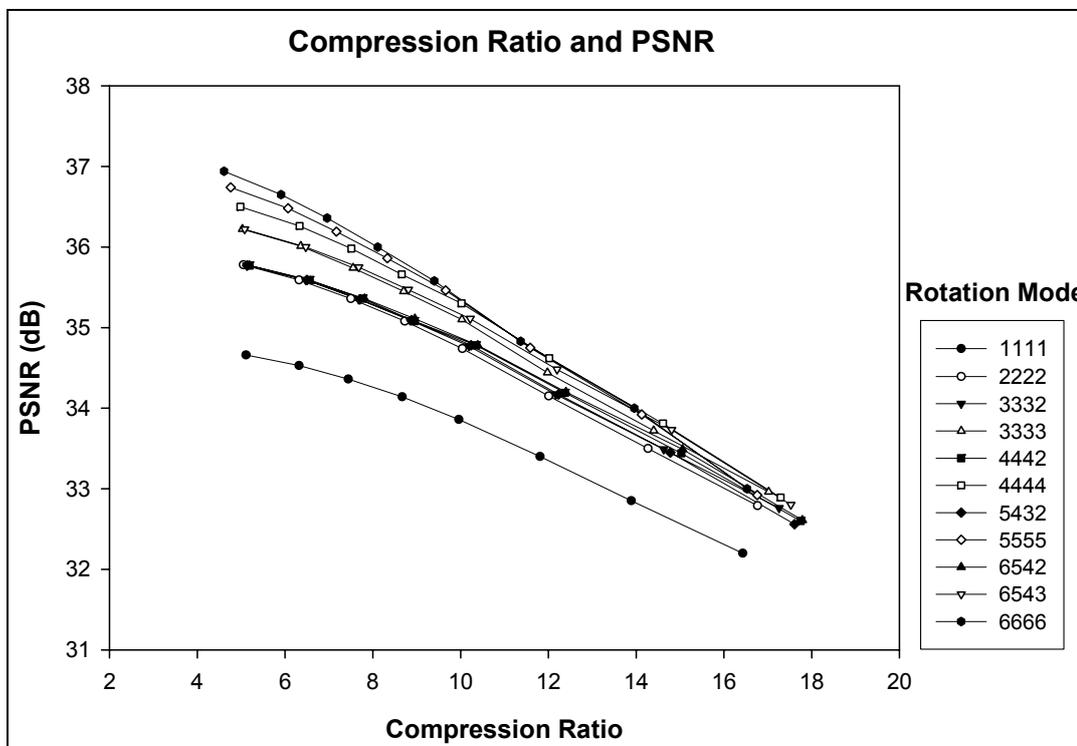


ภาพที่ 38 ความสัมพันธ์ระหว่างค่า m กับค่า PSNR ของโหมด 4444



ภาพที่ 39 ความสัมพันธ์ระหว่างค่า m กับค่า PSNR ของโหมด 6666

2. การทดลองที่ 2



ภาพที่ 40 กราฟแสดงอัตราส่วนการบีบอัดและ PSNR ของภาพ Lenna

ในขั้นตอนวิธีต้นแบบ (Jaquin, 1992) การหมุนบนสี่เหลี่ยมจัตุรัสจะทำในทิศทาง 0, 90, 180 และ 270 องศาเท่านั้น โหมดการหมุนที่ขั้นตอนวิธีต้นแบบทำได้จึงมีเพียงโหมด 1111 และ 2222 ซึ่งจากการทดลองพบว่าโหมดที่เหลือซึ่งพัฒนาขึ้นในงานวิจัยนี้เกือบทุกรูปแบบจะให้คุณภาพของภาพที่ผ่านการบีบอัดดีกว่าโหมดของขั้นตอนวิธีเดิม ในขณะที่อัตราส่วนการบีบอัดก็สูงกว่าด้วย ดังจะเห็นได้จากการที่เส้นกราฟในภาพที่ 40 ซึ่งแสดงความสัมพันธ์ระหว่างอัตราส่วนการบีบอัดและ PSNR วางตัวอยู่ถัดไปทางด้านขวาบนมากกว่า ซึ่งแสดงว่าวิธีนั้นให้ค่าของ PSNR สูงกว่า ในขณะที่อัตราส่วนการบีบอัดก็สูงกว่าด้วย

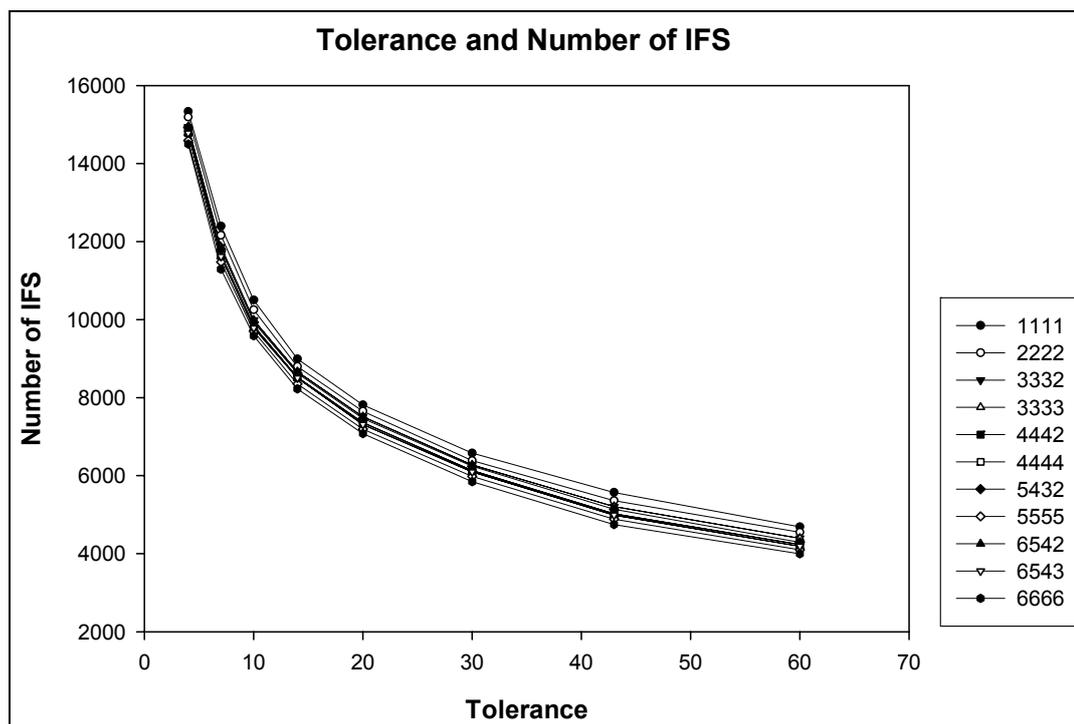
จริงอยู่ที่การหมุนในมุมที่ละเอียดขึ้นทำให้มีโอกาสพบความคล้ายในองศาอื่นได้ดีกว่า แต่เมื่อพิจารณาโครงสร้างของไฟล์ข้อมูลภาพที่บีบอัดแบบแฟร็กทัลดังในแสดงตารางที่ 3 จะเห็นได้ว่าเราสามารถคำนวณขนาดของไฟล์ข้อมูลที่บีบอัดโดยประมาณได้จากสูตร

$$\begin{aligned} \text{ขนาดของไฟล์ที่บีบอัด} &\cong \text{ขนาดของเซกเตอร์} + \text{จำนวนพาร์ติชัน} \times \\ &(\text{ceiling}(\log_2(\text{จำนวนแถวของโดเมน} \times \text{จำนวนหลักของโดเมน})) \\ &+ 13 + \text{จำนวนบิตที่ใช้บ่งบอกมุมของการหมุน}) \end{aligned}$$

สำหรับแต่ละโหมด ค่าตัวแปรต่างๆ ในสมการข้างต้นจะมีค่าเท่ากันหมด ยกเว้นเพียงจำนวนบิตที่ใช้บ่งบอกมุมของการหมุนเท่านั้น ซึ่งโหมดที่มีการหมุนในมุมที่ละเอียดกว่าย่อมต้องการจำนวนบิตในส่วนนี้มากกว่า ดังนั้นการใช้โหมดการหมุนที่ละเอียดขึ้น จะทำให้ขนาดของพารามิเตอร์ของฟังก์ชันใหญ่ขึ้นตามไปด้วย แต่การทดลองพบว่า การหมุนในโหมดที่ละเอียดขึ้นนั้น จะส่งผลให้พาร์ติชันที่ต้องใช้มีจำนวนลดลง เนื่องจากมีการพบความคล้ายในระดับพาร์ติชันที่ใหญ่ได้มากขึ้น จึงสามารถลดปริมาณการแตกควอดทริลงได้บางส่วน เกิดเป็นการแลกเปลี่ยนกันระหว่างบิตที่เพิ่มขึ้นกับจำนวนพาร์ติชันที่ลดลง จากกราฟผลการทดลองภาพที่ 40 พิสูจน์ให้เห็นว่าจำนวนของพาร์ติชันที่ลดลงนั้น มีปริมาณมากพอที่จะชดเชยในส่วนของบิตพิเศษที่เพิ่มขึ้นมา ซึ่งส่งผลให้คุณภาพของภาพดีขึ้น ในขณะที่อัตราส่วนการบีบอัดก็สูงขึ้นด้วย โดยเฉพาะในช่วงของอัตราส่วนการบีบอัดต่ำที่เน้นคุณภาพภาพมากกว่าขนาดไฟล์

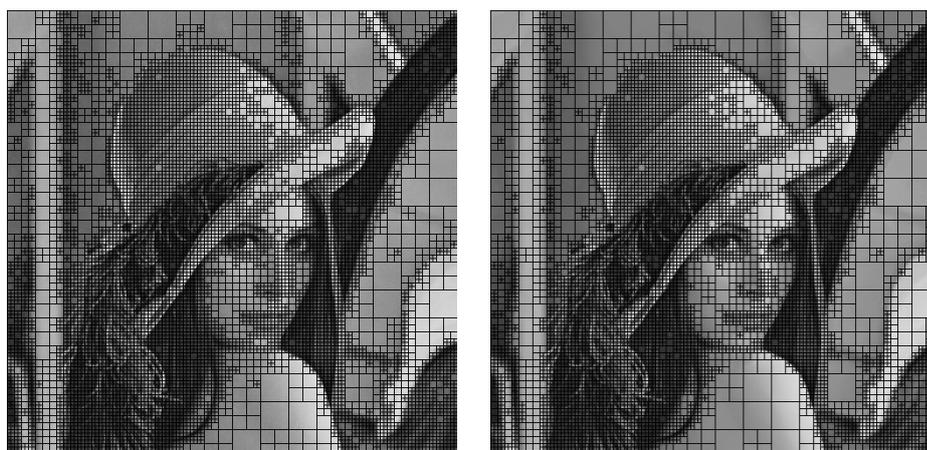
และผลการทดลองนี้ยังพบว่าเมื่อเพิ่มรูปแบบการหมุนมากขึ้น จะมีแนวโน้มที่คุณภาพของภาพจะยิ่งสูงขึ้นตามไปด้วย เป็นเพราะว่าภาพนี้เป็นภาพถ่ายของบุคคล ซึ่งมีระดับของความคล้ายตัวเอง (self-similarity) ในภาพค่อนข้างสูง กระจายอยู่ในหลายระดับขนาด และหลายองศาการหมุน การหมุนด้วยความละเอียดต่ำจึงมีโอกาสมองข้ามรูปแบบที่เหมาะสมเหล่านี้ไป เมื่อพิจารณาแนวโน้มของอัตราส่วนการบีบอัดจะพบว่ามีความไม่แน่นอน คือในช่วงของอัตราส่วนการบีบอัดต่ำ โหมดการหมุนที่ให้อัตราส่วนคุณภาพของภาพต่อขนาดไฟล์ดีที่สุด คือ 6666 แต่เมื่อพิจารณาช่วงอัตราส่วนการบีบอัดสูง โหมด 4444 จะให้ผลในแง่ของคุณภาพต่อขนาดไฟล์ที่ดีกว่า แสดงให้เห็นว่าจุดคุ้มค่าของการแลกเปลี่ยนต่างกันไปตามเงื่อนไขของค่าความผิดพลาดที่ยอมรับได้ การหมุนที่ละเอียดเกินไปจะทำให้พารามิเตอร์ฟังก์ชันมีขนาดใหญ่ขึ้น ในขณะที่จำนวนพาร์ติชันที่ลดลงได้อาจมีไม่มากพอที่จะชดเชยค่าใช้จ่ายส่วนนี้

อย่างไรก็ตามทั้งโหมด 4444 และ 6666 ต่างก็เป็นโหมดที่พัฒนาขึ้นจากการสังเคราะห์ด้วยวิธีที่เสนอในงานวิจัยนี้ ซึ่งให้ผลลัพธ์ที่ดีกว่าโหมด 1111 และ 2222 ที่เป็นโหมดที่ใช้ในขั้นตอนวิธีต้นแบบ

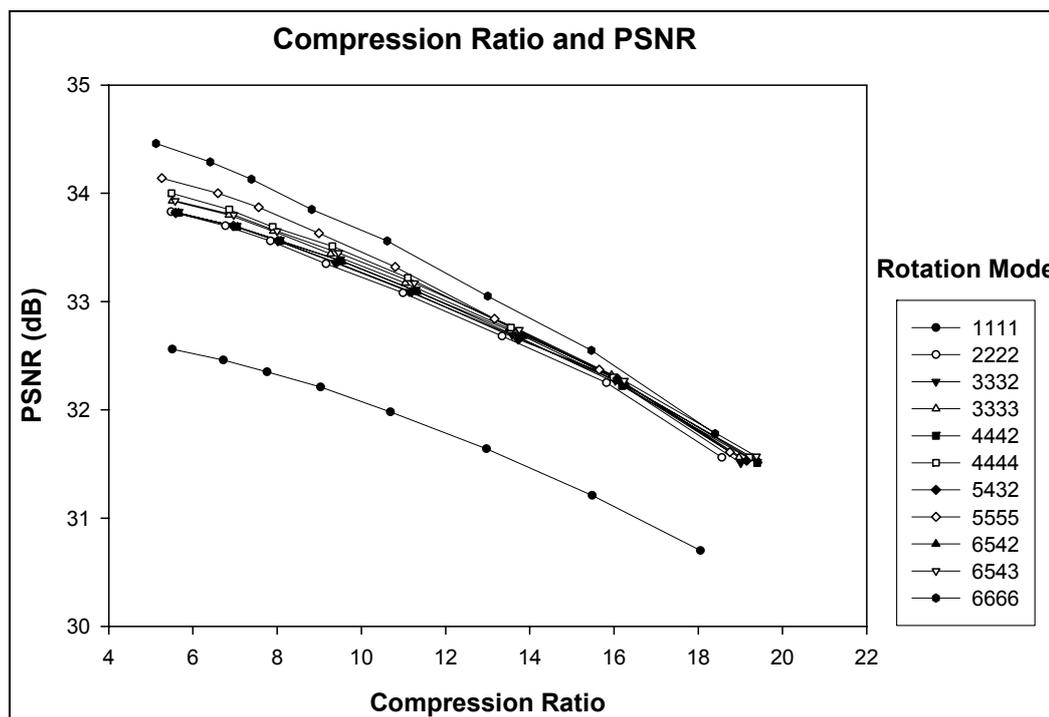


ภาพที่ 41 กราฟแสดงค่าระดับความผิดพลาดที่ยอมรับได้กับจำนวนของฟังก์ชันของภาพ Lenna

จากภาพที่ 41 พบว่าในขณะที่ระดับของความผิดพลาดเฉลี่ยที่ยอมรับได้ยังคงตั้งไว้ที่ค่าเดิม การหมุนที่ละเอียดขึ้นส่งผลให้จำนวนของพาร์ติชันลดลงอย่างเห็นได้ชัด จำนวนฟังก์ชันที่ต้องใช้ในการเข้ารหัสเพื่อแทนข้อมูลภาพจึงมีจำนวนลดลง แสดงให้เห็นว่าการหมุนที่ละเอียดขึ้นด้วยการส่งคงแบบ ส่งผลให้ให้เกิดการค้นพบความคล้ายในองศาใหม่นอกเหนือจากในมุมมอง

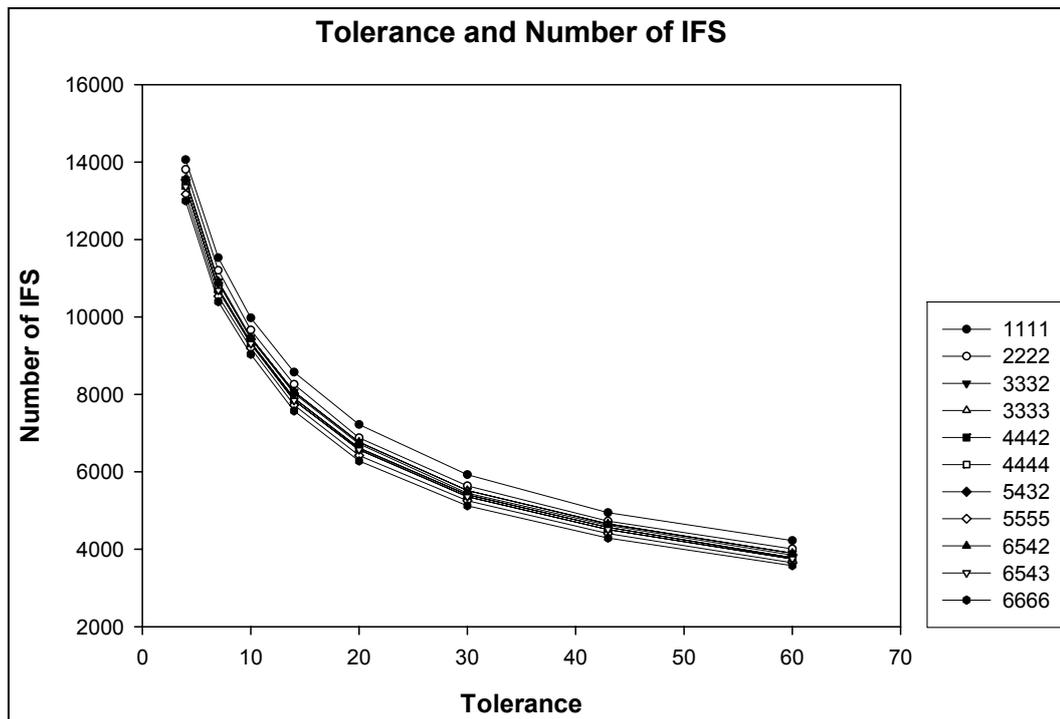


ภาพที่ 42 แสดงการแบ่งพาร์ติชันของโหมด 2222 (ซ้าย) เปรียบเทียบกับโหมด 6666 (ขวา) แสดงให้เห็นว่าการหมุนละเอียดขึ้นทำให้เกิดพาร์ติชันที่มีขนาดใหญ่เป็นจำนวนมากกว่า



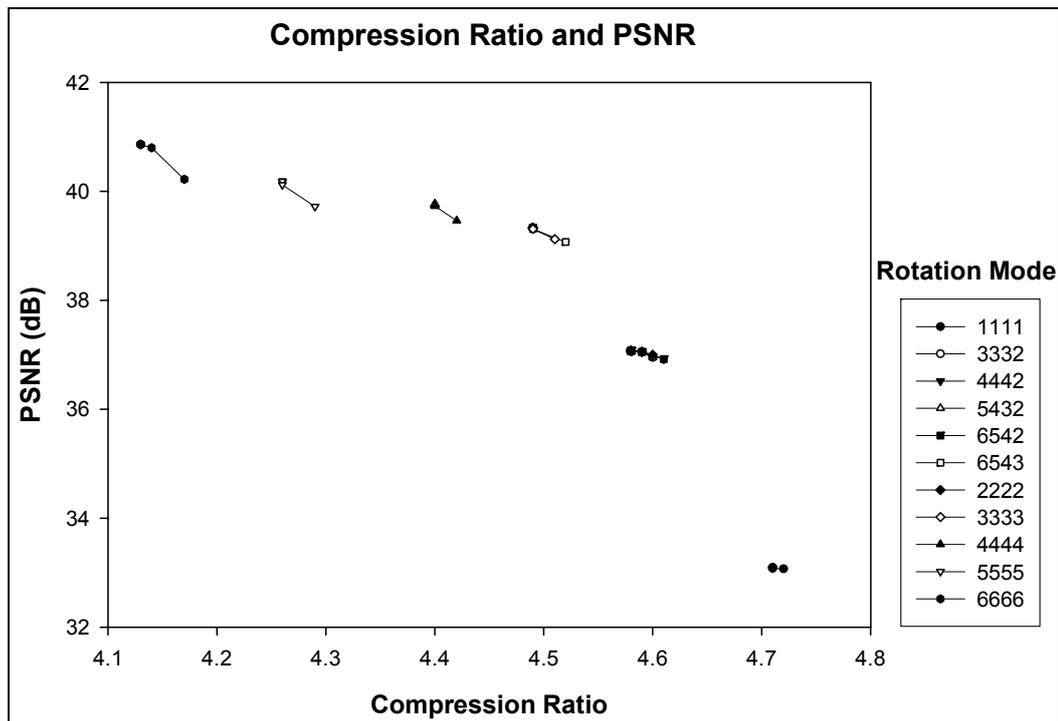
ภาพที่ 43 กราฟแสดงอัตราส่วนการบีบอัดและ PSNR ของภาพ Peppers

กราฟข้างต้นมีแนวโน้มใกล้เคียงกับภาพ Lenna เนื่องจากเป็นภาพถ่ายที่มีโทนระดับสีเทาค่อนข้างต่อเนื่องคล้ายกัน และมีลักษณะของความคล้ายตัวเองภายในภาพเช่นเดียวกัน โหมด 1111 ให้ผลที่ด้อยที่สุดอย่างเห็นได้ชัด ในขณะที่โหมด 2222 ให้ผลที่ใกล้เคียง โหมด 3332, 4442, 5432 และ 6542 แต่โหมด 3333, 4444, 5555 และ 6666 ให้ผลที่ดีกว่า โดยเฉพาะโหมด 6666 ดีกว่าอย่างเห็นได้ชัดในช่วง PSNR มากกว่า 32 แสดงให้เห็นว่าการหมุนอย่างละเอียดในพาร์ติชันขนาดเล็กมีผลต่อคุณภาพรวมของการบีบอัดเป็นอย่างมาก



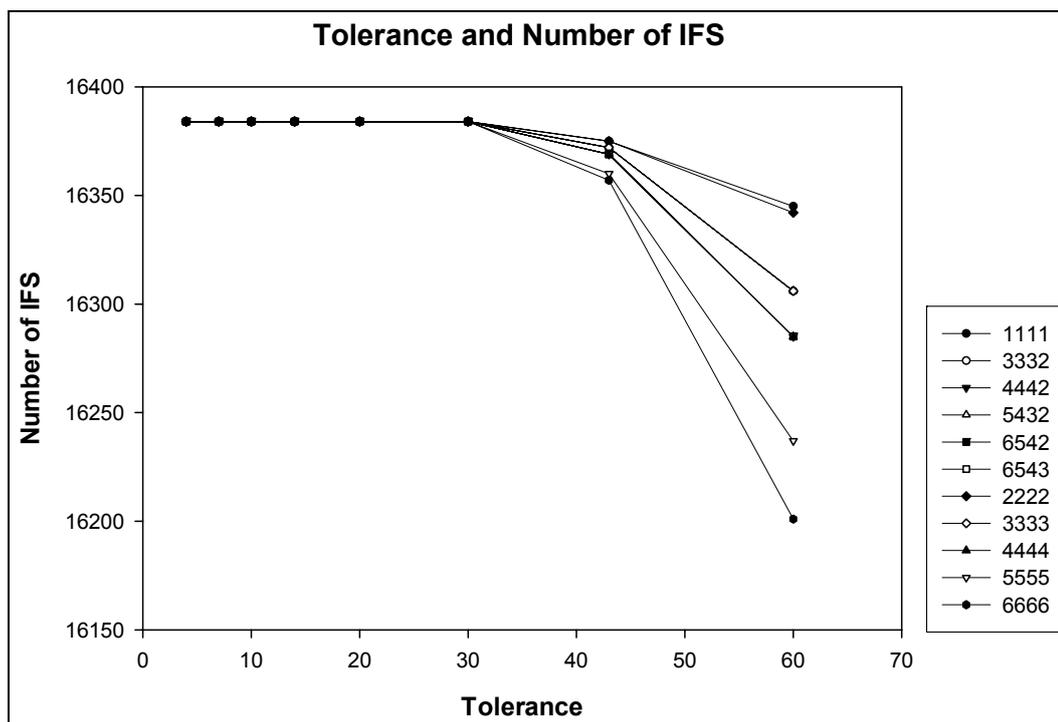
ภาพที่ 44 กราฟแสดงค่าระดับความผิดพลาดที่ยอมรับได้กับจำนวนของฟังก์ชันของภาพ Peppers

กราฟข้างต้นพบว่า แนวโน้มของความสัมพันธ์ระหว่างค่าระดับความผิดพลาดที่ยอมรับได้กับจำนวนของฟังก์ชันที่ใช้ยังคงคล้ายคลึงกับภาพ Lena กล่าวคือ การเพิ่มรูปแบบการหมุนในองศาที่ละเอียดขึ้น ทำให้จำนวนฟังก์ชันลดลงในระดับที่มากเพียงพอที่จะชดเชยกับบิดพิเศษที่เพิ่มขึ้นมา เนื่องจากรูปแบบการหมุนที่หลากหลายขึ้น



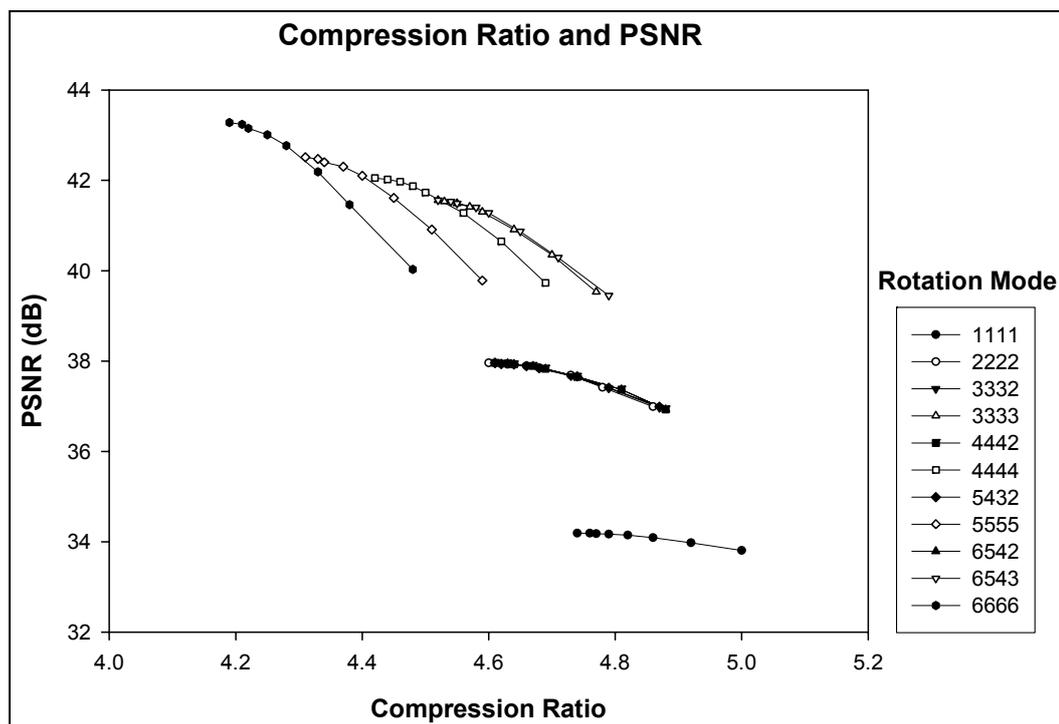
ภาพที่ 45 กราฟแสดงอัตราส่วนการบีบอัดและ PSNR ของภาพ Fabric

จากกราฟข้างบนพบว่าสำหรับภาพ Fabric ถึงแม้จะเพิ่มระดับความผิดพลาดที่ยอมรับได้มากขึ้นเพียงใด อัตราส่วนการบีบอัดก็ไม่แตกต่างจากเดิมมากนัก สังเกตได้ว่าโหมดที่มีจำนวนรูปแบบการหมุนมากกว่ามีแนวโน้มจะทำให้คุณภาพของภาพที่สูงกว่า แต่ปัจจัยที่มีผลต่อคุณภาพของภาพจริงๆคือจำนวนรูปแบบการหมุนในพาร์ติชันขนาด 4×4 ซึ่งเป็นขนาดเล็กที่สุดเท่านั้น ทั้งนี้เป็นเพราะว่าภาพนี้มีระดับของความคล้ายตัวเองสูงมาก แต่ความคล้ายเหล่านั้นอยู่ในระดับขนาดเล็กมาก ต้องทำการแตกพาร์ติชันลึกลงไปในระดับที่เล็กที่สุดจึงจะพบความคล้ายเหล่านั้น การหมุนในระดับพาร์ติชันใหญ่จึงไม่มีผลต่อ PSNR เท่าไรนัก และเนื่องจากพาร์ติชันเกือบทั้งหมดมีขนาดเล็กมาก จำนวนของพารามิเตอร์ของฟังก์ชันที่ต้องบันทึกของไฟล์จึงมีจำนวนมากตามไปด้วย อัตราส่วนการบีบอัดจึงต่ำ



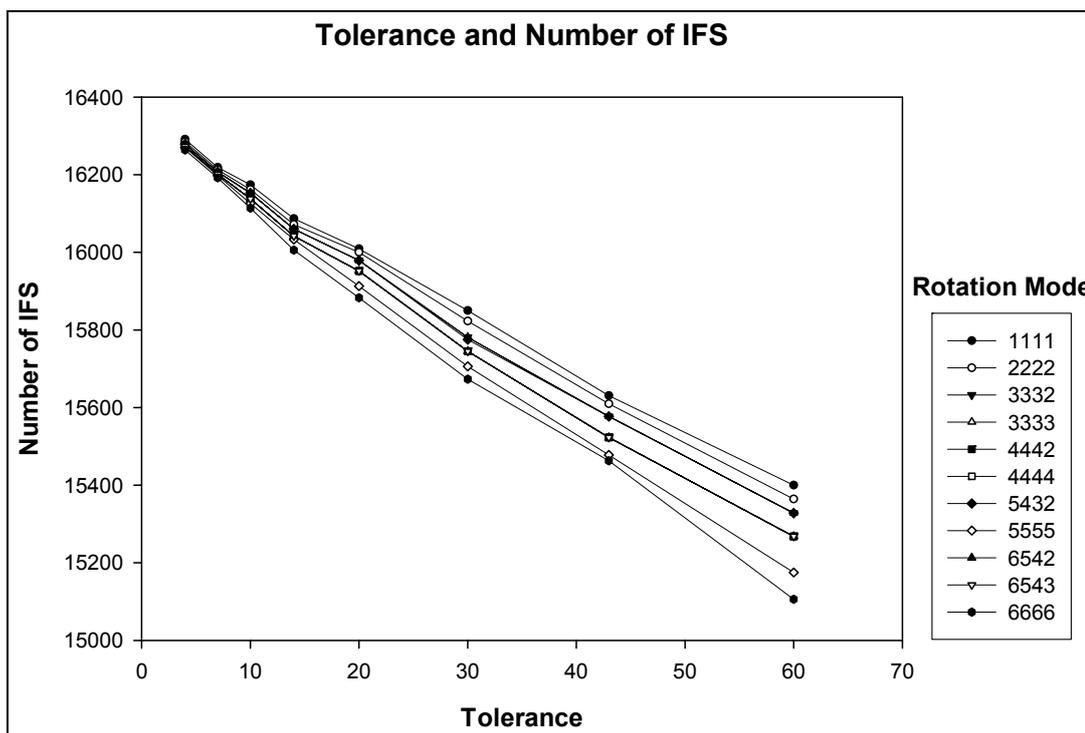
ภาพที่ 46 กราฟแสดงค่าระดับความผิดพลาดที่ยอมรับได้กับจำนวนของฟังก์ชันของภาพ Fabric

จากกราฟจะพบว่าถึงแม้จะยอมให้มีความผิดพลาดสูงขึ้น แต่จำนวนของพาร์ติชันก็ยังคงไม่แตกต่างจากเดิมมากนัก ทั้งนี้เป็นเพราะพาร์ติชันส่วนใหญ่เป็นพาร์ติชันระดับที่เล็กที่สุด ทั้งนี้เนื่องจากภาพมีรายละเอียดที่เล็กมากและมีความแปรปรวนของจุดสีสูง จึงเป็นการยากที่จะพบความคล้ายในระดับพาร์ติชันที่ใหญ่ ความคล้ายเกือบทั้งหมดจึงเกิดในระดับของพาร์ติชันขนาดเล็กที่สุด ส่งผลให้การลดจำนวนฟังก์ชันลงทำได้ไม่คืนกแม้จะใช้โหมดที่มีการหมุนละเอียดก็ตาม การหมุนในมุมที่ละเอียดจึงมีผลต่อคุณภาพของภาพเป็นหลัก ในขณะที่อัตราส่วนการบีบอัดกลับลดต่ำลง เนื่องจากไม่สามารถลดจำนวนพาร์ติชันลงได้เพียงพอที่จะชดเชยการเพิ่มบิตเพื่อนิยามรูปแบบการหมุนที่หลากหลายขึ้น



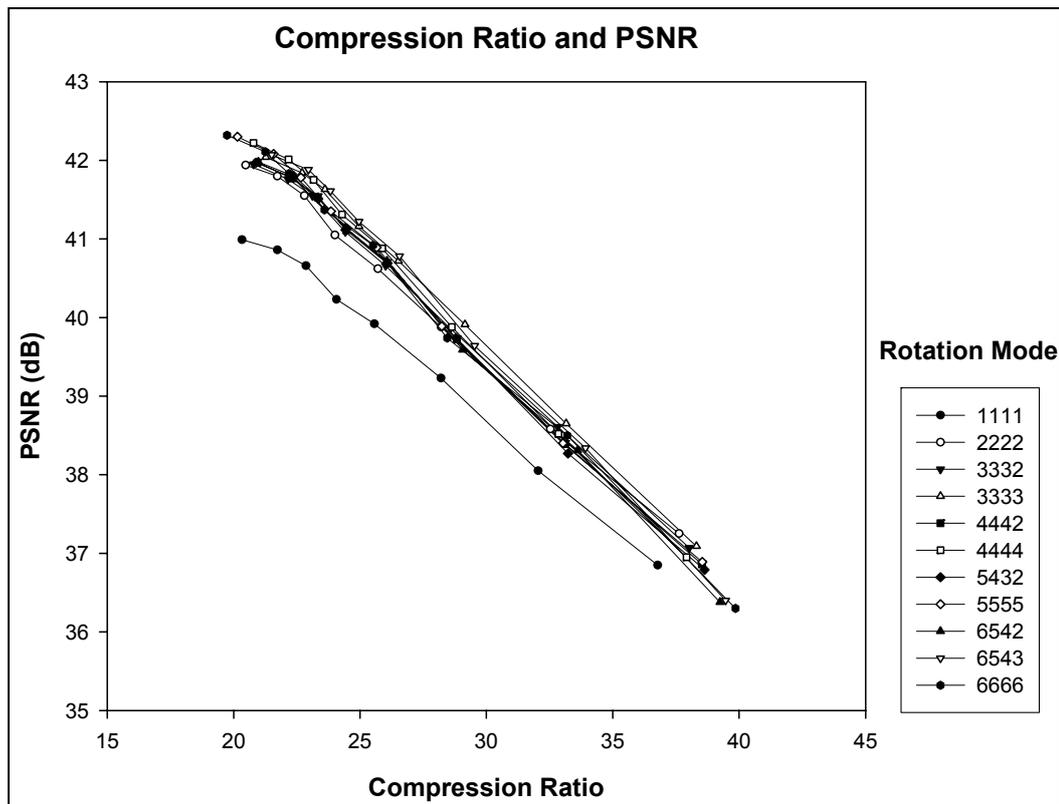
ภาพที่ 47 กราฟแสดงอัตราส่วนการบีบอัดและ PSNR ของภาพ Rock

จากกราฟของการทดลองนี้ พบว่าโหมด 6666 สามารถให้คุณภาพการบีบอัดสูงสุด เช่นเดียวกับกับภาพอื่นๆ เพียงแต่ในบางช่วงของอัตราส่วนการบีบอัด ความคุ้มค่าของการแลกเปลี่ยนจะดีกว่าโหมด 5555 และ 4444 เนื่องจากภาพนี้เป็นภาพพื้นผิวของภูเขาที่มีความแปรปรวนของจุดสีสูง ภาพมีรายละเอียดเล็กๆเป็นจำนวนมาก ความคล้ายส่วนใหญ่จึงเกิดในระดับที่เล็กมาก และไม่อาจลดจำนวนพิกเซลลงได้มากพอ การเพิ่มจำนวนรูปแบบการหมุนมากเกินไป จึงเป็นความสิ้นเปลือง



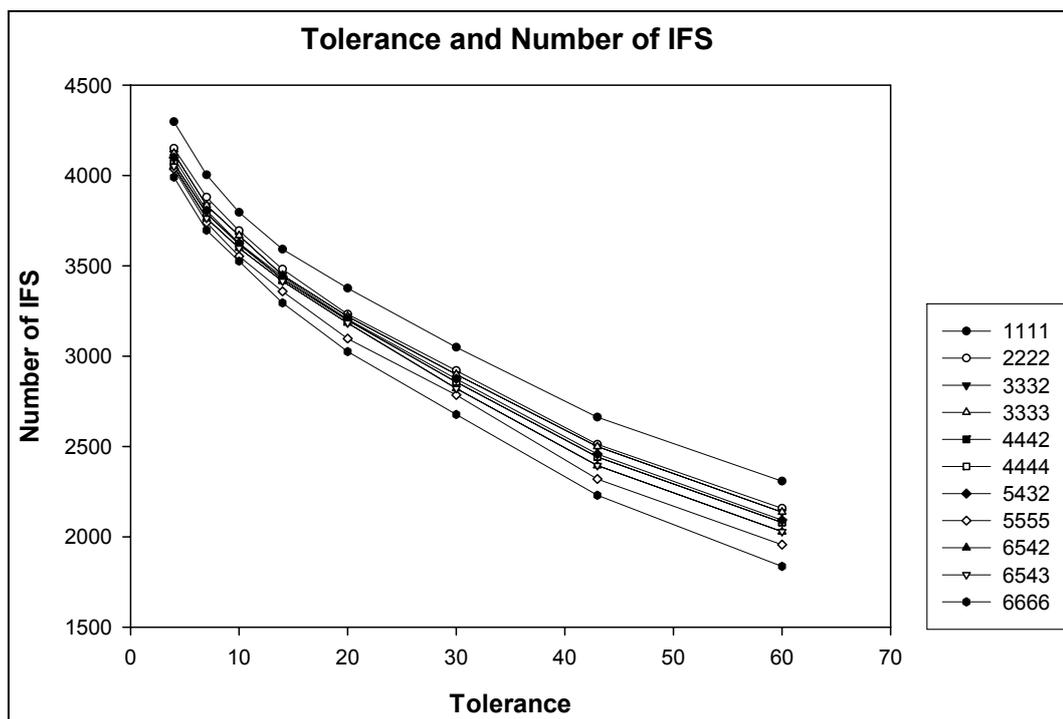
ภาพที่ 48 กราฟแสดงค่าระดับความผิดพลาดที่ยอมรับได้กับจำนวนของฟังก์ชันของภาพ Rock

จากกราฟพบว่าเมื่อกำหนดระดับความผิดพลาดที่ยอมรับได้ต่ำ จำนวนของพาร์ติชันก็ยังคงไม่แตกต่างจากเดิมมากนัก การหมุนในมุมที่ละเอียดจึงมีผลต่อคุณภาพของภาพเป็นหลัก แต่เมื่อกำหนดระดับความผิดพลาดที่ยอมรับได้ไว้สูง จะเห็นผลของความหลากหลายของรูปแบบการหมุนได้ชัดเจนขึ้น เนื่องจากการหมุนในมุมละเอียดมีโอกาสนำให้ได้ฟังก์ชันที่ให้ความผิดพลาดต่ำในระดับที่ยอมรับได้ จำนวนฟังก์ชันจึงน้อยกว่า



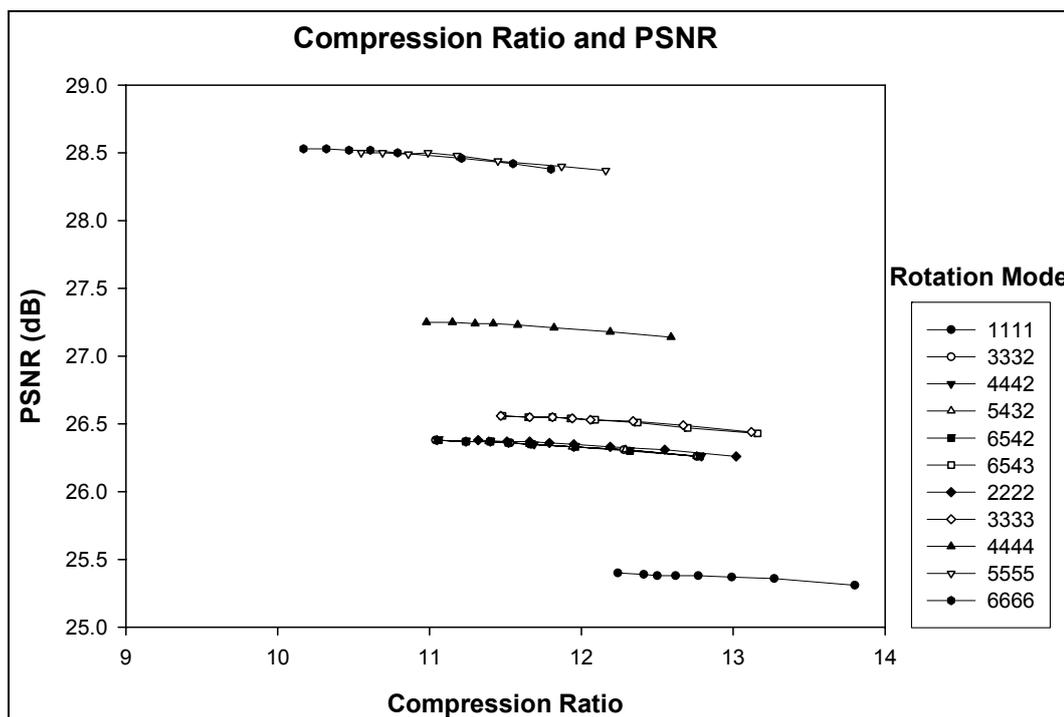
ภาพที่ 49 กราฟแสดงอัตราส่วนการบีบอัดและ PSNR ของภาพ Colonel

จากกราฟการทดลอง พบว่าอัตราส่วนการบีบอัดของภาพนี้อยู่ในเกณฑ์ที่สูงมาก ในขณะที่ PSNR ก็อยู่ในระดับที่สูงมากเช่นกัน เนื่องจากภาพนี้เป็นภาพที่มีลักษณะของความแปรปรวนของจุดสีต่ำ และมีความคล้ายตัวเองในทุกครั้งที่สูงมาก ซึ่งเป็นผลดีต่อขั้นตอนวิธีการบีบอัดแบบแฟร็กทัล เนื่องจากการพบความคล้ายในระดับพาร์ติชันที่ใหญ่เป็นจำนวนมากนั้นหมายถึงการประหยัดจำนวนฟังก์ชันที่ใช้ในการเข้ารหัสภาพ ซึ่งจะส่งผลให้อัตราส่วนของการบีบอัดสูงขึ้นตามไปด้วย และจะเห็นได้ว่าการหมุนในมุมที่ละเอียดขึ้นนั้นทำให้อัตราส่วนการบีบอัดและคุณภาพของภาพดีขึ้น เป็นเพราะว่าการหมุนในมุมที่ละเอียดขึ้นส่งผลให้พบความคล้ายในรูปแบบใหม่ ทำให้สามารถลดจำนวนการแตกพาร์ติชันลงได้มากพอที่จะชดเชยบิตพิเศษที่เพิ่มขึ้นมา



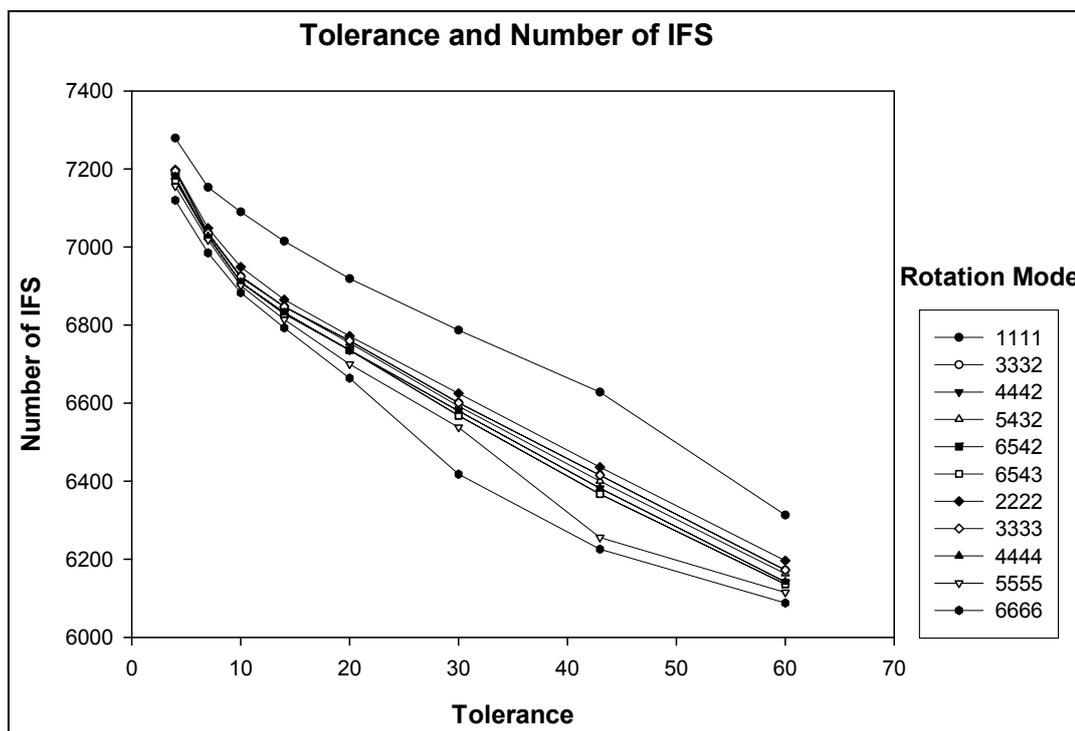
ภาพที่ 50 กราฟแสดงค่าระดับความผิดพลาดที่ยอมรับได้กับจำนวนของฟังก์ชันของภาพ Colonel

จากกราฟพบว่าการเพิ่มรูปแบบการหมุน ส่งผลให้จำนวนของพาร์ติชันลดลงอย่าง
 สม่ำเสมอ เนื่องจากความคล้ายกระจายอยู่ในทุกระดับขนาดพาร์ติชัน การหมุนที่ละเอียดจึงสามารถ
 สร้างโอกาสที่จะพบความคล้ายได้มากขึ้นในทุกขนาด



ภาพที่ 51 กราฟแสดงอัตราส่วนการบีบอัดและ PSNR ของภาพ France

จากกราฟจะเห็นว่าอัตราส่วนการบีบอัดค่อนข้างสูง แต่คุณภาพของภาพอยู่ในเกณฑ์ที่ไม่ดีนัก เป็นเพราะว่าภาพชนิดนี้ไม่เหมาะกับขั้นตอนวิธีการบีบอัดแบบแฟร็กทัล เนื่องจากลักษณะของภาพมีการผสมระหว่างส่วนที่เป็นภาพกราฟิก ซึ่งมีระดับของความคล้ายตัวเองสูง และส่วนที่เป็นตัวอักษรขนาดเล็ก ซึ่งเป็นส่วนที่มีคุณสมบัติความคล้ายตัวเองต่ำ ในบริเวณนี้จึงเกิดการแบ่งพาร์ติชันที่เล็กมาก แต่เนื่องจากตัวอักษรเหล่านี้ไม่มีส่วนคล้ายคลึงบริเวณใดในภาพ ฟังก์ชันที่นิยามความคล้ายในส่วนของบริเวณนี้จึงมีความผิดพลาดสูง ส่งผลให้ค่า PSNR โดยรวมต่ำ จากกราฟจะเห็นว่า การเพิ่มการใช้โหมดที่มีรูปแบบการหมุนมากขึ้นสามารถเพิ่มคุณภาพของภาพได้อย่างมาก แสดงให้เห็นว่าขั้นตอนวิธีที่นำเสนอให้ผลดีกว่าขั้นตอนวิธีเดิม ถึงแม้คุณภาพของภาพอยู่ในเกณฑ์ที่ไม่ดีนัก

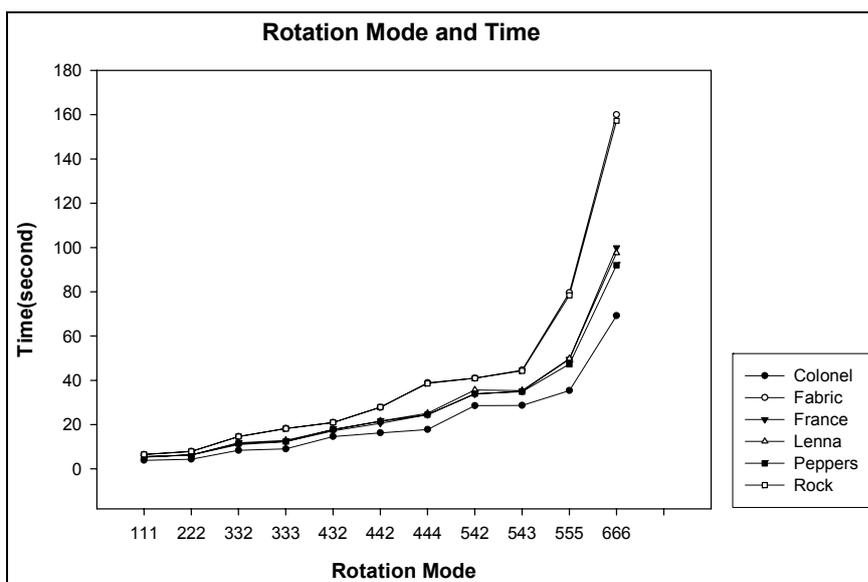


ภาพที่ 52 กราฟแสดงค่าระดับความผิดพลาดที่ยอมรับได้กับจำนวนของฟังก์ชันของภาพ France

จากกราฟผลการทดลองพบว่าเมื่อกำหนดระดับความผิดพลาดที่ยอมรับได้ต่ำ จำนวนพาร์ติชันของโหมด 1111 จะสูงกว่าโหมดอื่นมาก ส่วนโหมดอื่นจำนวนพาร์ติชันก็ยังคงไม่แตกต่างจากเดิมมากนัก แต่เมื่อกำหนดระดับความผิดพลาดที่ยอมรับได้ไว้สูง จะเห็นผลของความหลากหลายของรูปแบบการหมุนได้ชัดเจนขึ้น เนื่องจากการหมุนในมุมละเอียดมีโอกาสทำให้ได้ฟังก์ชันที่ให้ความผิดพลาดต่ำในระดับที่ยอมรับได้ จำนวนฟังก์ชันจึงน้อยกว่า

3. การทดลองที่ 3

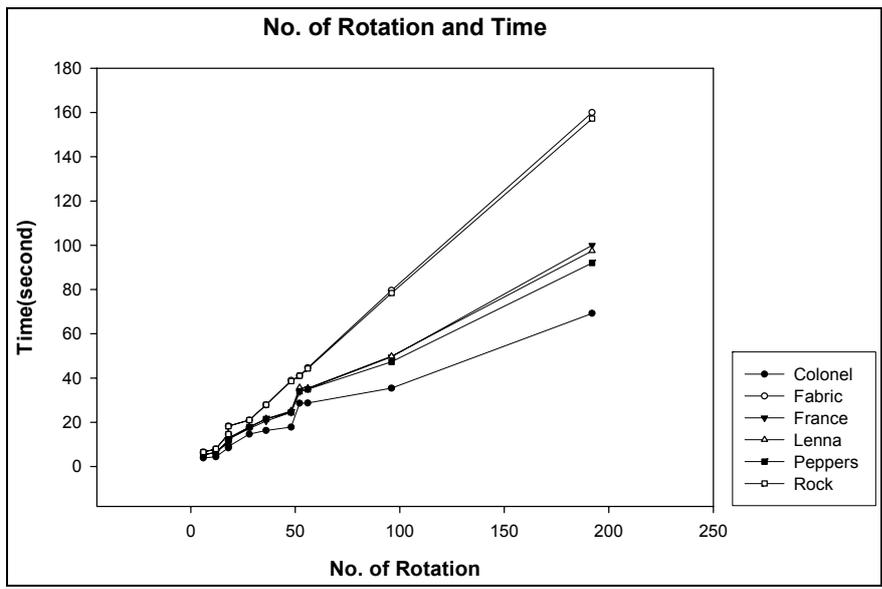
นำผลการทดลองที่ 2 ของการย่อข้อมูลแบบแฟร็กทัลด้วยควอดตรี 3 ระดับ (ตารางที่ 11) มาพล็อตแสดงความสัมพันธ์ระหว่างโหมดต่างๆ และเวลาที่ใช้ในการประมวลผล จะได้กราฟดังแสดงในภาพที่ 53



ภาพที่ 53 กราฟแสดงโหมดการหมุนและเวลาที่ใช้ เมื่อใช้ควอดตรี 3 ระดับ

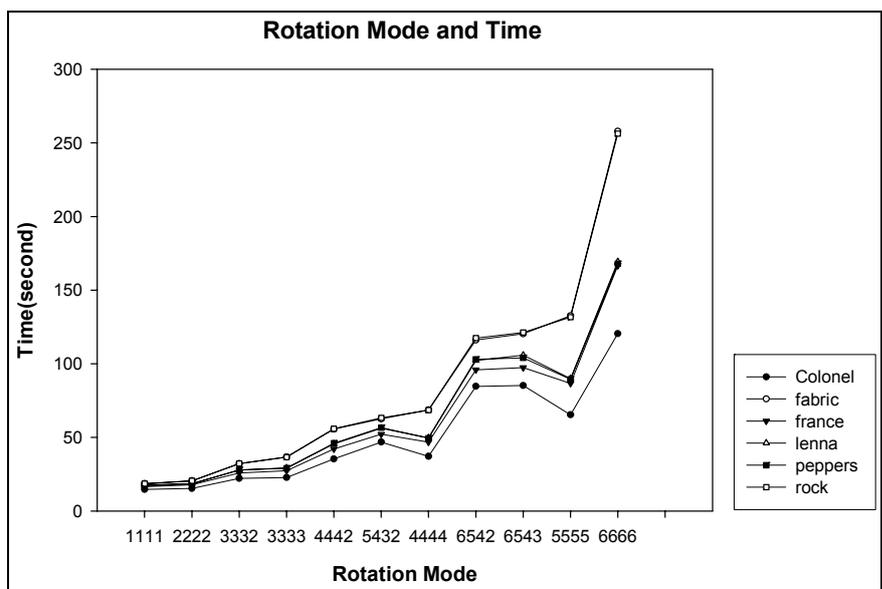
จะเห็นได้ว่าเมื่อมีการหมุนหลากหลายรูปแบบขึ้น ก็จะมีการใช้เวลามากขึ้น ซึ่งเวลาที่ใช้ในการประมวลผลจะประกอบด้วย 2 ส่วนคือ เวลาในการสร้างทรี ซึ่งจะมีค่าเท่ากันในทุกโหมด และเวลาในการค้นหาโดเมนบล็อก ซึ่งจะมากขึ้นตามรูปแบบการหมุนที่ต้องตรวจสอบ

เมื่อนำเวลาที่ใช้ มาพล็อตกับจำนวนของรูปแบบการหมุนรวมในแต่ละระดับควอดตรี จะได้กราฟตามภาพที่ 54 จะพบว่ากราฟมีแนวโน้มที่เกือบจะเป็นเส้นตรง นั่นคือโดยประมาณแล้ว เวลาที่ใช้ในการประมวลผลเกือบจะเป็นสัดส่วนโดยตรงกับจำนวนรูปแบบการหมุน โดยความแตกต่างของความชันของกราฟแต่ละเส้น จะขึ้นอยู่กับจำนวนของพาร์ติชันที่เกิดขึ้น ซึ่งผลที่ได้นั้นก็สอดคล้องกับผลการทดลองที่ 2



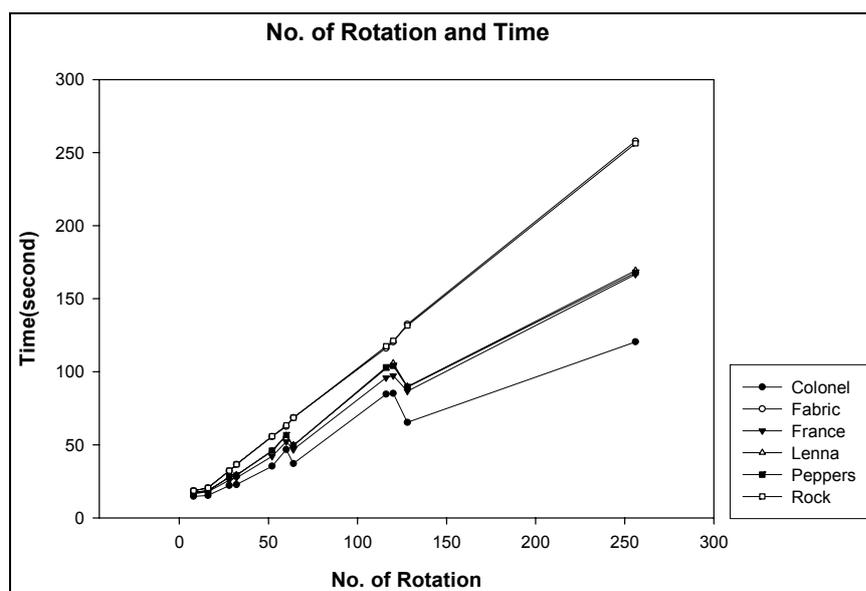
ภาพที่ 54 กราฟแสดงรูปแบบรวมของการหมุนและเวลาที่ใช้ เมื่อใช้ควอดตริ 3 ระดับ

สำหรับควอดตริ 4 ระดับ เมื่อนำผลการทดลองในตารางที่ 12 มาพล็อตแสดงความสัมพันธ์ระหว่างโหมดต่างๆ และเวลาที่ใช้ในการประมวลผล ได้กราฟดังรูปที่ 55 จะพบแนวโน้มเช่นเดิม กล่าวคือ เมื่อมีการหมุนหลากหลายรูปแบบขึ้น ก็จะใช้เวลามากขึ้น



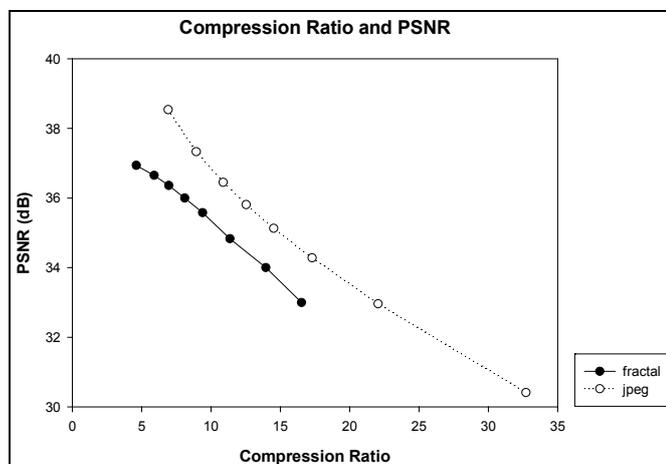
ภาพที่ 55 กราฟแสดงโหมดการหมุนและเวลาที่ใช้ เมื่อใช้ควอดตริ 4 ระดับ

แตเมื่อนำผลข้างต้นมาพล็อตกราฟ ระหว่างจำนวนของรูปแบบของการหมุนรวมในแต่ละระดับควอดทรี กับเวลาที่ใช้ จะได้กราฟตามภาพที่ 56 ซึ่งพบว่ามีบางจุดของกราฟที่กระโดดออก จากแนวเส้นตรงไปมาก ใน 3 โหมด ได้แก่ 5432, 6542 และ 6543 ซึ่งเป็น 3 โหมดที่ทำให้ความสำคัญ กับพาร์ติชันขนาดใหญ่สุด (32×32) มากกว่าพาร์ติชันขนาดรองลงมา ซึ่งจะพบว่าโหมด ดังกล่าวนี้นี้มีการใช้เวลามากกว่า เมื่อเทียบกับจำนวนรูปแบบของการหมุนที่อาจจะไม่ได้มากไปกว่า โหมดใกล้เคียงกันที่ทำให้ความสำคัญของการหมุนในระดับที่เปลี่ยนใกล้เคียงกันในทุกขนาดพาร์ติชัน ทั้งนี้อาจเป็นเพราะว่าการหมุนละเอียดในพาร์ติชันขนาด 32×32 นั้นเกิดความสูญเปล่ามากกว่า ซึ่ง หมายความว่า จะตรวจสอบด้วยของละเอียดหรือไม่ขึ้นอยู่กับว่าจำเป็นต้องแตกพาร์ติชันอยู่ดี แสดงให้เห็นว่าการหมุนในลักษณะที่ทำให้ความสำคัญกับพาร์ติชันใดพาร์ติชันหนึ่งมากเกินไปจะทำให้ผลไม่คุ้มค่าในแง่ของเวลา กล่าวคือ อาจจะช่วยให้พบพาร์ติชันในระดับใหญ่ได้เพิ่มขึ้น และลดการแตกควอดทรีได้บ้าง แต่เวลาที่หมดไปกับการทดสอบการหมุนในรูปแบบที่เพิ่มขึ้นมานี้มีปริมาณมากกว่า ซึ่งคิดกับโหมดที่เฉลี่ยความสำคัญไปตามแต่ละขนาดพาร์ติชัน นอกจากจะไม่ต้องเช็คพาร์ติชันขนาดใหญ่สุดอย่างละเอียดแล้ว ยังมีแนวโน้มที่จะสามารถลดการแตกพาร์ติชันลง และทำให้ปริมาณของการหมุนโดยเฉลี่ยต่อพาร์ติชันลดลงอีกด้วย แต่จากกราฟ ของ Colonel และ Fabric ที่ผลออกมาเป็นเส้นตรง เมื่ออ้างอิงจากการผลทดลองที่ 2 ที่พบว่าภาพนี้มีระดับความคล้ายตัวเอง ในระดับขนาดพาร์ติชันที่เล็กมาก ดังนั้นการหมุนละเอียดในระดับพาร์ติชันขนาดใหญ่ ส่วนใหญ่จึงเป็นการหมุนที่ไม่ทำให้เกิดประโยชน์ กล่าวคือไม่พบมุมมองชาติที่ดีพอในระดับที่จะนำไปใช้ได้ ต้องทำการแตกพาร์ติชันจนสุดอยู่ดีดังนั้นเวลาที่ใช้จึงแปรผันตรงกับกับจำนวนรูปแบบการหมุนรวม



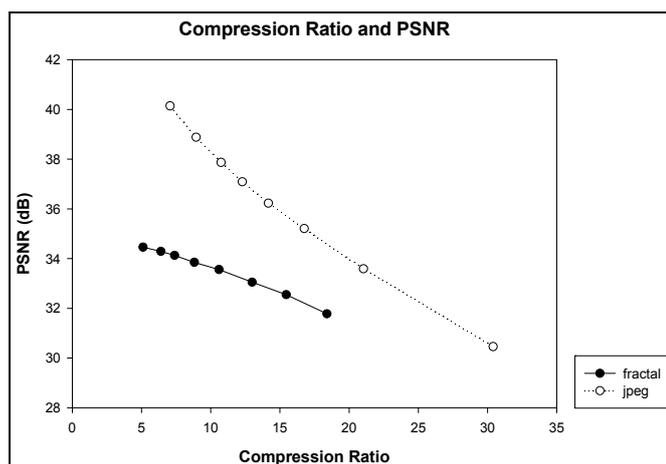
ภาพที่ 56 กราฟแสดงรูปแบบรวมของการหมุนและเวลาที่ใช้ เมื่อใช้ควอดทรี 4 ระดับ

4. การทดลองที่ 4



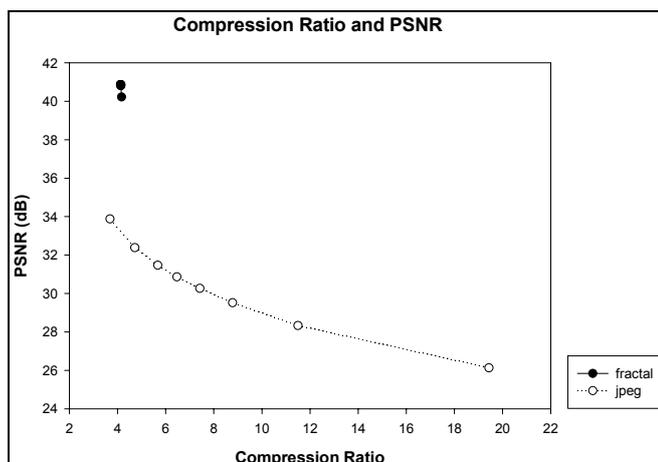
ภาพที่ 57 กราฟเปรียบเทียบการบีบอัดแบบแฟร็กทัลกับเจบีคของภาพ Lenna

จากกราฟจะพบว่าถึงแม้การบีบอัดแบบแฟร็กทัลที่มีหมุนในมุมที่ละเอียดจะให้คุณภาพและอัตราส่วนการบีบอัดในเกณฑ์ที่ดีขึ้นกว่าแบบเดิม แต่ก็ยังด้อยกว่าการบีบอัดแบบเจบีค ทั้งนี้เนื่องจากลักษณะของภาพเป็นภาพถ่ายที่มีโทนสีต่อเนื่องซึ่งเป็นลักษณะที่เหมาะสมกับขั้นตอนวิธีแบบเจบีคอยู่แล้ว การที่จะให้ขั้นตอนวิธีแบบแฟร็กทัลบีบอัดได้ดีกว่า จำเป็นต้องอาศัยการแบ่งพาร์ติชันที่มีประสิทธิภาพกว่าควอดทรี



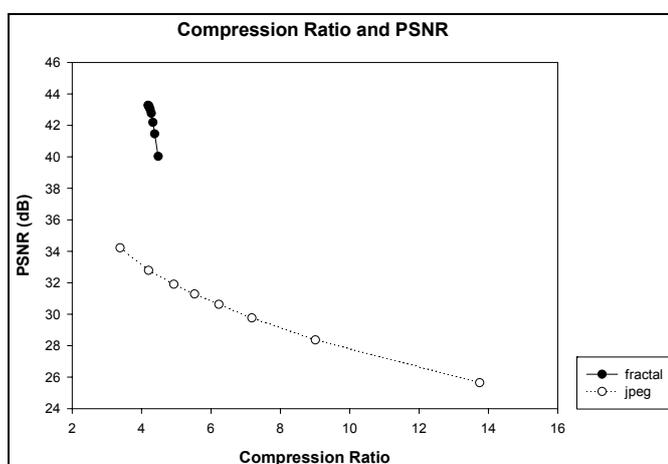
ภาพที่ 58 กราฟเปรียบเทียบการบีบอัดแบบแฟร็กทัลกับเจบีคของภาพ Peppers

เช่นเดียวกันกับภาพ Lenna จะพบว่าการบีบอัดแบบแฟร็กทัลยังให้ผลลัพธ์ที่ด้อยกว่าการบีบอัดแบบเจบีค ทั้งนี้เนื่องจากลักษณะของภาพเป็นภาพ โทนสีต่อเนื่องเช่นเดียวกัน



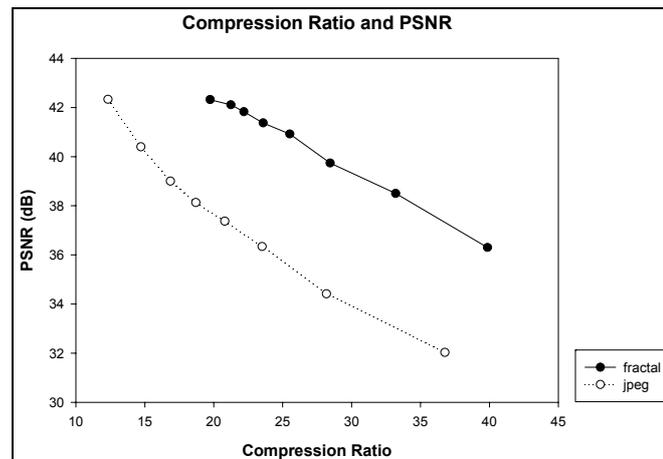
ภาพที่ 59 กราฟเปรียบเทียบการบีบอัดแบบแฟร็กทัลกับเจเป็กของภาพ Fabric

จากกราฟพบว่าที่ระดับการบีบอัดเท่าๆกัน การบีบอัดแบบแฟร็กทัลให้คุณภาพที่ดีกว่า ทั้งนี้เนื่องจากลักษณะของภาพมีการเปลี่ยนแปลงค่าสีขึ้นลงในช่วงแคบๆสูงมาก ซึ่งโดยปกติเจเป็กจะบีบอัดภาพลักษณะนี้ได้ไม่ดีนัก



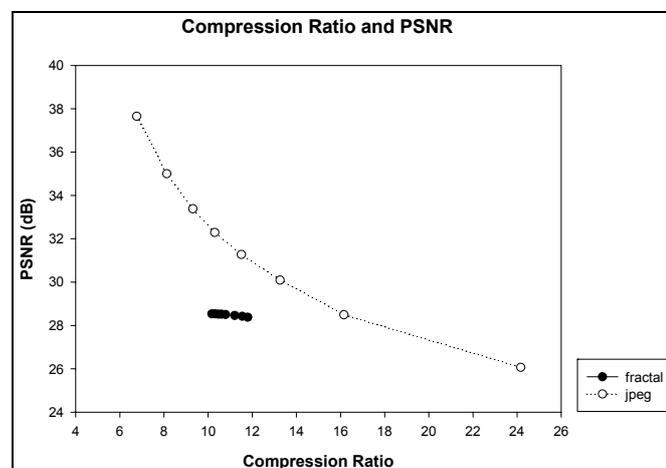
ภาพที่ 60 กราฟเปรียบเทียบการบีบอัดแบบแฟร็กทัลกับเจเป็กของภาพ Rock

เช่นเดียวกับกับภาพ Fabric ที่ระดับการบีบอัดเท่าๆกัน การบีบอัดแบบแฟร็กทัลให้คุณภาพที่ดีกว่า ทั้งนี้เนื่องจากลักษณะของภาพมีความแปรปรวนของค่าสีในบริเวณใกล้เคียงกันสูงมาก



ภาพที่ 61 กราฟเปรียบเทียบการบีบอัดแบบแฟร็กทัลกับเจบีคของภาพ Colonel

จากกราฟข้างต้น จะเห็นได้ชัดว่าขั้นตอนวิธีการบีบอัดแบบแฟร็กทัลมีประสิทธิภาพสูงกว่ามากเนื่องจากภาพนี้เป็นภาพโลโก้ที่มีลักษณะของความแปรปรวนของจุดสีดำ และมีความคล้ายตัวเองในทุกระดับขนาดที่สูงมาก



ภาพที่ 62 กราฟเปรียบเทียบการบีบอัดแบบแฟร็กทัลกับเจบีคของภาพ France

จากกราฟพบว่าการบีบอัดแบบเจบีคมีประสิทธิภาพดีกว่า เนื่องจากภาพนี้มีบริเวณของภาพที่ไม่มีลักษณะของการคล้ายตัวเอง จึงทำให้ขั้นตอนวิธีการบีบอัดแบบแฟร็กทัลทำงานได้ไม่ดีนัก



ภาพที่ 63 ภาพขั้นตอนการถอดรหัสแบบดั้งเดิม แสดงให้เห็นลักษณะการก่อดัวของภาพซึ่งเกิดจาก ไอเอฟเอสที่นิยามโดยฟังก์ชันที่หมุนในแนวฉาก ภาพจึงมีลักษณะเป็นบล็อกสีเหลี่ยม ที่จะค่อยๆลดขนาดลงเมื่อทำซ้ำไปเรื่อยๆ



ภาพที่ 64 ภาพขั้นตอนการถอดรหัสแบบมีการหมุนด้วยการส่งคงแบบ ซึ่งจะเห็นได้ว่าสำหรับในแต่ละพาร์ติชัน บล็อกจะถูกหมุนไปในทิศทางที่เหมาะสม สอดรับกับส่วนประกอบของภาพในบริเวณนั้น

สรุป

1. การนำการส่งคงแบบมาสร้างฟังก์ชันการหมุนในองศาอิสระที่เสนอในงานวิจัยชิ้นนี้ สามารถเพิ่มโอกาสในการค้นพบความคล้ายระหว่างโดเมนและเรนจ์ในมุมมองศาอื่นๆที่ดีกว่าค่าฟังก์ชันการค้นในทิส 0, 90, 180 และ 270 องศา และสามารถเพิ่มคุณภาพของภาพที่บีบอัดได้เสมอ
2. การหมุนด้วยมุมที่ละเอียดขึ้น สามารถลดจำนวนของพารามิเตอร์ที่ต้องใช้ในการเข้ารหัสได้เสมอ
3. โดยทั่วไปแล้ว ในโหมดที่มีการหมุนด้วยมุมที่ละเอียดกว่า 90 องศา จำนวนพารามิเตอร์ที่สามารถลดลงได้ มีปริมาณมากพอที่จะชดเชยการเพิ่มของบิตพิเศษที่ต้องใช้บ่งบอกมุมของการหมุน ส่งผลให้คุณภาพของภาพดีขึ้น ในขณะที่อัตราส่วนการบีบอัดก็สูงขึ้นด้วย โดยเฉพาะในช่วงของอัตราการบีบอัดต่ำที่เน้นคุณภาพมากกว่าขนาดไฟล์ จุดคุ้มค่าของการแลกเปลี่ยนระหว่างบิตพิเศษที่เพิ่มขึ้นกับจำนวนพารามิเตอร์ที่ลดลงได้นั้น แตกต่างกันไปในแต่ละภาพ
4. คุณลักษณะของภาพมีส่วนสำคัญต่อประสิทธิภาพของการบีบอัด ภาพที่มีความคล้ายตัวเองสูงอย่างภาพถ่ายที่มีโทนสีต่อเนื่อง และภาพโลโก้ มีแนวโน้มที่จะบีบอัดด้วยขั้นตอนวิธีการบีบอัดแบบแฟร็กทัลได้ดี ในขณะที่ภาพที่มีรายละเอียดเล็กมากอย่างภาพลวดลาย (texture) และภาพที่มีตัวหนังสือ อัตราส่วนการบีบอัดจะไม่ดีนัก

เอกสารและสิ่งอ้างอิง

Au, O. C., M. L. Liou and L. K. Ma. 1997. Fast Fractal Encoding in Frequency Domain. **IEEE International Conference on Image Processing**, California.

Barnsley, M. F. 1988. **Fractal Everywhere**. Academic Press, San Diego.

_____ and A.D. Sloan. 1988. A better way to compress images. **BYTE**, 13 (1): 215 - 223.

_____ and L.P. Hurd. 1992, **Fractal Image Compression**. AK Peters Ltd., Wellesley, Massachusetts.

Bently, J.L. 1975. Multidimensional binary search trees used for associative searching. **Comm. ACM** 18 (9): 509 - 517.

Boss, R. D. and E. W. Jacobs. 1994. Archetype classification in an iterated transformation image compression algorithms". In Y. Fisher. **Fractal Image Compression -- Theory and Applications**. Springer-Verlag, New York.

Cardinal J. 1998. Fractal compression using the discrete Karhunen-Loeve transform. **Brussels Free University Internal Report**.

_____, 2001. Fast fractal compression of greyscale images. **IEEE Trans. Image Processing** 10 (1): 159 - 164.

Churchill, R. V. and J. W. Brown. 1990. **Complex Variables and Applications**. 5th edition. McGraw-Hill.

Driscoll, T. A. 1996. Algorithm 756: A MATLAB toolbox for Schwarz-Christoffel mapping, **ACM Trans. Math. Software** 22 (2): 239 - 251.

- Falconer, K. 1995. **Fractal Geometry, Mathematical Foundations and Applications**. Wiley, New York.
- Fisher, S. D. 1999. **Complex Variables**. 2nd edition. Dover.
- Fisher, Y. 1994. Fractal Image Compression with Quadrees. In Y. Fisher. **Fractal Image Compression -- Theory and Applications**. Springer-Verlag, New York.
- _____ and S. Menlove. 1994 Fractal encoding with HV partitions. In Y. Fisher. **Fractal Image Compression -- Theory and Applications**. Springer-Verlag, New York.
- _____, E. W. Jacobs and R. D. Boss. 1994. Fractal image compression using iterated transforms, In J. Storer and Ed. Norwell. **Image and Text Compression**, Massachusetts.
- Friedman, J. H., J. L. Bentley and R. A. Finkel. 1997, An algorithm for finding best matches in logarithmic expected time. **ACM Transactions on Mathematical Software** 3 (3): 209 - 226
- Gonzalez, M. O. 1992 **Complex Analysis Selected Topics**. Dekker.
- Gonzalez, R. C. and R. E. Woods. 2002. **Digital Image Processing. 2nd edition**. Prentice Hall, New Jersey.
- Jacquin, A. 1992, Image coding based on a fractal theory of iterated contractive image transformations. **IEEE Trans. Image Processing** 1 (1): 18 - 30.
- _____, 1993. Fractal image coding: A review. **Proceedings of the IEEE** 81 (10): 1451 - 1465.
- Lai, C. M., K. M. Lan and W. C. Siu. 2003, A fast fractal image coding based on kick-out and zero contrast conditions. **IEEE Trans. on Image Processing** 12 (11): 1398 - 1403.

MandelBrot, B. 1982. **Fractal Geometry of Nature**. Freeman, New York.

Moon, Y. H., H. S. Kim and J. H. Kim. 2000. A fast fractal decoding algorithm based on the selection of an initial image. **IEEE Trans. Image Processing** 9 (5): 941 - 945.

Paliouras, J. D. and D.S. Meadows. 1990. **Complex Variables for Scientists and Engineers**. 2nd Edition. Macmillan.

Polvere, M. and M. Nappi. 2000. Speed-Up In Fractal Image Coding: Comparison of Methods, **IEEE Trans. on Image Processing** 9 (6): 1002 - 1009.

Reusens, E. 1994. Partitioning complexity issue for iterated function systems based image coding, pp. 171-174. **Proc. VIIth Eur. Signal Processing Conf.** Edinburgh, U.K.

Saupe, D., R. Hamzaoui and H. Hartenstein. 1996. Fractal image compression: an introductory overview. *Fractal Models for Image Synthesis, Encoding, and Analysis*. **SIGGRAPH '96 Course Notes XX**.

_____ and M. Ruhl. 1996. Evolutionary fractal image compression. **Proc. ICIP-96 IEEE International Conference on Image Processing** 1, Lausanne.

_____, _____ and Hartenstein. 1997. Adaptive partitionings for fractal image compression. **Proc. ICIP-97 IEEE International Conference on Image Processing**, Santa Barbara, California.

_____, _____ and _____. 2000. Region-based fractal image compression. **IEEE Transactions on Image Processing** 9 (7): 1171 - 1184.

Sproull, R. F. 1991. Refinements to Nearest-Neighbor Searching in k-Dimensional Trees.

Algorithmica, 6: 579 - 589.

Thomas, L. and F. Deravi. 1995. Region-based fractal image Compression using heuristic

search. **IEEE Transactions on Image Processing**, 4 (6): 832- 838.

Truong, T. K. et al. 2000, A Fast Encoding Algorithm for Fractal Image Compression Using the

DCT Inner Product. **IEEE Trans. on Image Processing** 9 (4): 529 - 535.

Tong, C. S. and M. Wong. 2002, Adaptive Approximate Nearest Neighbor Search for Fractal

Image Compression. **IEEE Transactions on Image Processing** 11 (6): 605 - 615.

Wein, C. J. and I. F. Blake I. F. 1996. On the performance of fractal compression with clustering.

IEEE.Trans on Image Processing 5 (3): 522 - 526.

ประวัติการศึกษา และการทำงาน

ชื่อ นายชาวีร์ อิศริยภัทร์
เกิดวันที่ 7 กุมภาพันธ์ พ.ศ. 2519
สถานที่เกิด อำเภอเมือง จังหวัดยะลา
ประวัติการศึกษา วท.บ. (คณิตศาสตร์) มหาวิทยาลัยสงขลานครินทร์ (พ.ศ. 2541)