

ขั้นตอนวิธีใช้ตัวเลขชนิดใหม่สำหรับการต่อเติมภาพที่ใช้การแปรผันรวมกับ
การประยุกต์สำหรับซ้อมแซมภาพจิตกรรรมไทยโบราณและการลบบท
บรรยายจากอนิเมะ

A new numerical algorithm for TV-based image inpainting with its
applications for restoring ancient Thai painting images and
removing subtitles from animes

จัดทำโดย
ภัคพล พงษ์ทวี รหัส 07580028

อาจารย์ที่ปรึกษา
อาจารย์ ผู้ช่วยศาสตราจารย์ ดร. นพดล ชุมชอบ

เอกสารฉบับนี้เป็นส่วนหนึ่งของรายวิชา 511 493 โครงการวิจัย
ภาคเรียนที่ 2 ปีการศึกษา 2561
สาขาวิชาคณิตศาสตร์ ภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์ มหาวิทยาลัยศิลปากร

สาขาวิชาคณิตศาสตร์ ภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์ มหาวิทยาลัยศิลปากร มีความเห็นชอบให้โครง
งานวิจัยเรื่องขั้นตอนวิธีใช้ตัวเลขชนิดใหม่สำหรับการต่อเติมภาพที่ใช้การแปรผันรวมกับการประยุกต์สำหรับซ่อมแซม
ภาพจิตรกรรมไทยโบราณและการลบบทบรรยายจากอนิเมะ (A new numerical algorithm for TV-based im-
age inpainting with its applications for restoring ancient Thai painting images and removing subti-
tles from animes) ชื่อเสนอโดย นายภัคพล พงษ์ทวี รหัส 07580028 เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
วิทยาศาสตรบัณฑิต สาขาวิชาคณิตศาสตร์ ประจำปีการศึกษา 2561

อาจารย์ ผู้ช่วยศาสตราจารย์ ดร. นพดล ชุมชอบ อาจารย์ที่ปรึกษา

_____ / _____ / _____

อาจารย์ ดร. นัยนรัตน์ กันยะมี กรรมการ

_____ / _____ / _____

อาจารย์ ผู้ช่วยศาสตราจารย์ ดร. พรทรัพย์ พรสวัสดิ์ กรรมการ

_____ / _____ / _____

บทคัดย่อ

(PLACE)

คำสำคัญ: การต่อเติมภาพ

Abstract

Genetic algorithm

Keywords: Image inpainting

กิจกรรมประจำ

โครงการวิจัยเรื่องการจัดตารางสอนและตารางสอบของภาควิชาคณิตศาสตร์ มหาวิทยาลัยศิลปากรด้วยขั้นตอนเชิงพื้นฐาน (Creating Class Schedule and Exam Schedule for Department of Mathematics, Silpakorn University, by using Genetic Algorithm) สำเร็จลุล่วงไปด้วยดีด้วยความกรุณาและคำแนะนำจากอาจารย์ที่ปรึกษาโครงการวิจัยที่เสียสละเวลาให้ความอนุเคราะห์ดูแลช่วยเหลือ แนะนำ ตรวจสอบข้อบกพร่อง

สารบัญ

บทคัดย่อ	i
บทคัดย่อ	ii
กิตติกรรมประกาศ	iii
สารบัญรูป	vi
สารบัญตาราง	vii
1 บทนำ	1
1.1 ที่มาและความสำคัญ	1
1.1.1 การซ้อมแซมภาพจิตรกรรมไทยโบราณ	2
1.1.2 การลับบทบรรยายบนอนิเมะ	3
2 วรรณกรรมและทฤษฎีบทที่เกี่ยวข้อง	5
2.1 การต่อเติมภาพเนดเทา	5
2.1.1 ตัวแบบการต่อเติมภาพเนดสีเทาที่ใช้การแปรผันรวม	6
2.2 วิธีการเชิงตัวเลขสำหรับการต่อเติมภาพเนดเทา	6
2.2.1 วิธีการเดินเวลาแบบบัดจัด (explicit time marching method)	6
2.2.2 วิธีการทำซ้ำแบบจุดตึง (fixed-point iteration method)	8
2.2.3 วิธีการสปริทเบรกแมน (Split Bregman method)	9

2.3	ตัวแบบการต่อเติมภาพสีที่ใช้การแปรผันรวม	11
2.4	การวัดประสิทธิภาพของภาพที่ผ่านกระบวนการการต่อเติม	12
3	การทดลองเชิงตัวเลข	14
3.1	การซ้อมแซมภาพจิตกรรมไทยโบราณ	14
3.1.1	การเปรียบเทียบประสิทธิภาพขั้นตอนวิธีเชิงตัวเลขที่มีอยู่แล้ว	16
3.1.2	ขั้นตอนวิธีเชิงตัวเลขสำหรับต่อเติมภาพชนิดใหม่	22
3.1.3	การทดสอบประสิทธิภาพในการซ้อมแซมภาพจิตกรรมไทยโบราณ	30
3.2	การลับบทบรรยายบนอนิเมะ	36
3.2.1	การทำบทบรรยายบนอนิเมะ	37
3.2.2	การลับคำบรรยายจากบทอนิเมะ	39
4	สรุป	44
	บรรณานุกรม	45

สารบัญ

1.1.1	ตัวอย่างการซ่อมแซมภาพ	2
1.1.2	ภาพจิตรกรรมไทยที่วัดภูมินทร์ อำเภอเมือง จังหวัดป่าน	3
1.1.3	1 เฟรมของอนิเมะที่มีบทบรรยายแบบแข็ง	4
2.1.1	<i>D</i> แทนโดเมนต่อเติม	5
3.1.1	ภาพต้นฉบับ	15
3.1.2	ภาพที่จะทำการซ่อมแซม	16
3.1.3	ผลการซ่อมแซมจากการเดินเวลา	17
3.1.4	ผลการซ่อมแซมจากการทำข้าแบบจุดตรึง	19
3.1.5	ผลการซ่อมแซมจากการสปริทเบรกแม่น	21
3.1.6	วิธีการพิริมิดรูปภาพ	23
3.1.7	พิริมิดที่ดำเนินการทำข้าเป็น 10/10/10 และที่ระดับความคมชัดละเอียดสุดใช้จำนวนการทำข้าที่ต่างกัน	28
3.1.8	ภาพต้นฉบับสำหรับใช้ในการทดสอบ	31
3.1.9	ภาพที่ทำให้เสียหาย	32
3.1.10	ผลการซ่อมแซมโดยวิธีการสปริทเบรกแม่น	33
3.1.11	ผลการซ่อมแซมภาพโดยวิธีการเชิงตัวเลขที่พัฒนาขึ้น	35
3.2.1	การแบ่งไฟล์วิดีโอเป็น 5 ส่วนสำหรับใช้เป็น 5 ชุดทดสอบ	37

สารบัญตาราง

3.1	ผลการซ่อมแซมวิธีการเดินเวลา	18
3.2	ผลการซ่อมแซมของวิธีการทำข้า้แบบจุดตึง	20
3.3	ผลการซ่อมแซมของวิธีสปริทเบรกแม่น	22
3.4	แสดงการซ่อมแซมเฉลี่ยของวิธีการเชิงตัวเลข	22
3.5	ผลการซ่อมแซมภาพโดยวิธีการเชิงตัวเลขที่นำเสนอ	26
3.6	ผลการซ่อมแซมภาพโดยวิธีการเชิงตัวเลขที่นำเสนอในรูปของค่าเฉลี่ยของผลที่ได้จากการ ที่ 3.5	27
3.7	ผลการซ่อมแซมภาพโดยวิธีการเชิงตัวเลขที่นำเสนอยื่นใช้การทำข้า้ในระดับความคมชัดละเอียด สุด 10 ครั้ง	29
3.8	ผลการซ่อมแซมภาพโดยวิธีการเชิงตัวเลขที่นำเสนอในรูปของค่าเฉลี่ยของผลที่ได้จากการ ที่ 3.7	30
3.9	ผลการซ่อมแซมภาพศิลปะไทยจากวิธีการสปิทเบรกแม่น	34
3.10	ผลการซ่อมแซมภาพศิลปะไทยโดยวิธีการเชิงตัวเลขที่พัฒนาขึ้น	36
3.11	แสดงผลการซ่อมแซมภาพศิลปะไทยในรูปค่าเฉลี่ยจากตารางที่ 3.9 และตารางที่ 3.10 . .	36
3.12	ความคลาดเคลื่อนของการหาโดยเมนต์ต่อเติม ในบทบรรยายภาษาต่างๆ	39
3.13	ความคลาดเคลื่อนเฉลี่ยของการหาโดยเมนต์ต่อเติม ในบทบรรยายภาษาต่างๆ	39
3.14	ผลการลับบทบรรยายออกจากอนิเมะด้วยวิธีการเชิงตัวเลขขั้นตอนวิธี 6, 11, 12 และ 13	42
3.15	ผลการซ่อมแซมภาพโดยวิธีการเชิงตัวเลขที่นำเสนอในรูปของค่าเฉลี่ยของผลที่ได้จากการ ที่ 3.14	43
3.16	ผลการลับบทบรรยายออกจากอนิเมะโดยวิธีการสปริทเบรกแม่นและวิธีการที่พัฒนาขึ้น .	43

บทที่ 1

ບໍ່ທຳນົດ

1.1 ที่มาและความสำคัญ

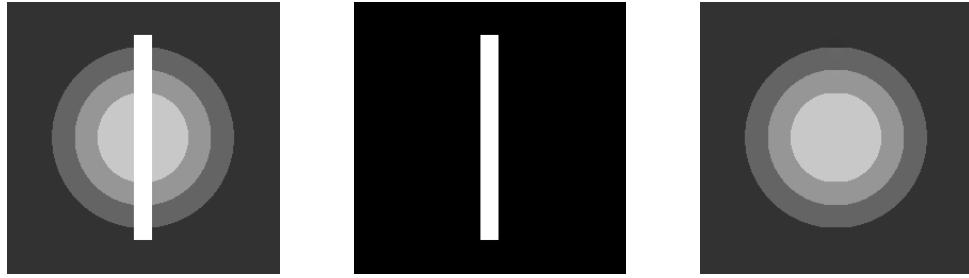
ในปัจจุบันการใช้ภาพดิจิตัล (digital images) ในสังคมเครือข่ายได้รับความนิยมอย่างแพร่หลาย เนื่องจากโทรศัพท์เคลื่อนที่มีราคาถูกลงแต่มีความสามารถที่ชาญฉลาด สามารถทำหน้าที่ได้ตั้งแต่การเป็นกล้องดิจิตัลคอมแพค (compact digital camera) คุณภาพดีให้ภาพดิจิตัลที่มีความคมชัดสูงจนไปถึงการทำงานหน้าที่ดังเช่นเครื่องคอมพิวเตอร์ ส่วนบุคคลที่สามารถเชื่อมต่อ กับระบบเครือข่ายไร้สายเพื่อรับส่งภาพดิจิตัลในสังคมเครือข่ายด้วยความสะดวกและรวดเร็ว

นอกจากภาพดิจิตัลจะได้รับจากการถ่ายภาพด้วยโทรศัพท์เคลื่อนที่แล้ว ภาพดิจิตัลยังได้รับการถ่ายภาพด้วยกล้องดีเอสแอลอาร์ หรือ กล้องสะท้อนเลนส์เดียวแบบดิจิตัล (digital single lens reflex camera) กล้องโทรศัพท์ (หรือ กล้องดูดาว) หรือ เครื่องมือสร้างภาพถ่ายทางการแพทย์ (medical imaging device)

โดยทั่วไปภาพดิจิตัลจะได้รับการประมวลผลภาพก่อนนำไปใช้งานเพื่อให้สามารถใช้ข้อมูลที่ปรากฏบนภาพได้ตรงวัตถุประสงค์ของการใช้งานมากที่สุด ตัวอย่างเช่น ภาพบุคคล (portrait) อาจจำเป็นต้องได้รับการทำจัดสัญญาณรบกวนออกจากภาพและ/หรือปรับเพิ่มความละเอียดข้อมูลของความเข้มของสีและความสว่างของสีบนบริเวณใบหน้าก่อนนำภาพไปใช้งานเพื่อจัดทำต้นฉบับวารสารหรือหนังสือของสำนักพิมพ์ เป็นต้น

การต่อเติมภาพ (image inpainting) เป็นวิธีการประมวลผลภาพชนิดหนึ่ง มีเป้าหมายเพื่อซ่อมแซมภาพด้วยการต่อเติมข้อมูลของความเข้มของสีบนบริเวณที่กำหนด (ต่อไปจะเรียกบริเวณนี้ว่า โดเมนต่อเติม (inpainting domain)) โดยอาศัยข้อมูลของความเข้มของสีที่ปรากฏในภาพ ตัวอย่างเช่น กำหนดให้รูปที่ 1.1.1 (a) แสดงภาพที่ต้องการซ่อมแซมระดับความเข้มของสีบนบริเวณแห่งวัตถุปะร่างสีเหลี่ยมสีขาว การต่อเติมภาพดังกล่าวจะเริ่ม

ด้วยการกำหนดให้บริเวณแห่งวัตถุรูปร่างสีเหลี่ยมสีขาวเป็นโดเมนการต่อเติมดังรูปที่ 1.1.1 (b) จากนั้นภาพที่ได้รับการซ้อมแซมหรือภาพที่ได้รับการต่อเติม (restored or inpainted image) ซึ่งแสดงในรูปที่ 1.1.1 (c) ได้มาจากขั้นตอนวิธีการต่อเติมภาพ (inpainting algorithm) ซึ่งได้รับการออกแบบเพื่อนำข้อมูลที่ปรากฏบนภาพในบริเวณใกล้เคียงกับขอบของโดเมนต่อเติมมาซ้อมแซมภาพ



(a) ภาพที่ต้องการซ้อมแซม

(b) โดเมนต่อเติม

(c) ภาพที่ได้รับการซ้อมแซม

รูปที่ 1.1.1: ตัวอย่างการซ้อมแซมภาพ

เท่าที่ผู้วิจัยศึกษาและค้นคว้ามาจนถึงขณะนี้ ผู้วิจัยพบว่าการต่อเติมภาพมักนิยมนำไปใช้งานสำหรับการปรับแต่งความสวยงามของภาพบุคคลที่ถ่ายจากโทรศัพท์เคลื่อนที่ เช่น การลบร่องรอยของรอยตื้นๆ การลบร่องรอยแผลเป็นที่เกิดจากสิ่งเสี้ยน การลดร่องรอยของความชรา หรือ การเพิ่มความใสและความเนียนของสีผิวนะ บริเวณใบหน้าผ่านโปรแกรมแอปพลิเคชันแต่งรูปภาพที่มีอยู่ในแอปสโตร์ (App Store) หรือ ဂุเก็ลเพลย์ (Google Play) เป็นต้น

1.1.1 การซ้อมแซมภาพจิตรกรรมไทยโบราณ

ภาพจิตรกรรมไทย คือ ภาพเขียนที่มีเอกลักษณ์ความเป็นศิลปะไทยซึ่งโดดเด่นและแตกต่างจากภาพเขียนของชนชาติอื่น ในอดีต ช่างไทยได้สร้างสรรค์ลวดลายและสีสันบนภาพวาดเพื่อสะท้อนประเพณีและวัฒนธรรมในสังคมไทยที่เกี่ยวกับศาสนา ประวัติศาสตร์โบราณคดี ชีวิตความเป็นอยู่ วัฒนธรรมการแต่งกาย ตลอดจนการแสดง การเล่นพื้นเมืองต่าง ๆ ของแต่ละยุคสมัย

อย่างไรก็ตาม ภาพจิตรกรรมไทยจำนวนไม่น้อยที่เสื่อมสภาพตามกาลเวลา และรอค่อยการซ้อมแซม จากช่างในสมัยปัจจุบันที่ต้องไม่สร้างความเสียหายให้กับภาพเขียนเพิ่มขึ้นมากกว่าเดิม ที่ผ่านมาภาพที่ผ่านการซ้อมแซมน้ำแล้วจำนวนไม่น้อยได้รับความเสียหายหลังจากการซ้อมแซม ถึงแม้สภาพโดยรวมของภาพจิตรกรรมเดิมยังคงอยู่แต่รายละเอียดในตัวภาพเขียนได้เปลี่ยนไป ก่อให้เกิดความเสียหายที่ประเมินค่าไม่ได้

การซ่อมแซมภาพจิตรกรรมไทยโบราณโดยใช้การต่อเติมภาพเป็นขั้นตอนของการซ่อมแซมแบบหนึ่ง ซึ่งไม่ก่อให้เกิดความเสียหายใด ๆ กับภาพเดิม เนื่องจากเป็นการซ่อมแซมโดยการใช้ขั้นตอนวิธีเชิงตัวเลขบนภาพดิจิตัลซึ่งเป็นสำเนาของภาพเดิม ด้วยเหตุผลดังกล่าว ผู้วิจัยได้เลือกเห็นว่าการซ่อมแซมภาพจิตรกรรมไทยโบราณมีความจำเป็นเร่งด่วน เนื่องจากภาพที่ได้รับการซ่อมแซมด้วยการต่อเติมภาพสามารถนำไปใช้ประกอบการตัดสินใจเพื่อวางแผนก่อนการลงมือซ่อมแซมภาพเขียนจริงได้ นอกจากนี้ ขั้นตอนวิธีการต่อเติมภาพสามารถนำไปใช้สร้างแอปพลิเคชันบนโทรศัพท์เคลื่อนที่เพื่อในไปใช้เป็นข้อมูลในการเข้าชมภาพเขียนเดิมที่ยังไม่ได้รับการซ่อมแซมและภาพเขียนที่ได้รับการซ่อมแซมโดยวิธีการทางคณิตศาสตร์จากแอปพลิเคชันที่พัฒนาขึ้น

รูปที่ 1.1.2 แสดงตัวอย่างภาพจิตรกรรมไทย¹ ที่ต้องได้รับการซ่อมแซมนบริเวณแขนเสื้อของรูปวด ผู้ชายที่มีส่วนของสีแดงเดิมหลุดหายไป ทั้งนี้ในการซ่อมแซมภาพโดยการต่อเติมภาพ เราจะเริ่มด้วยการสร้างโดเมนต่อเติมบนบริเวณสีพื้นผิวปูนที่แขนเสื้อ จากนั้นจึงนำขั้นตอนวิธีการต่อเติมภาพเพื่อซ่อมแซมภาพบริเวณนั้นให้เป็นสีเดิม



รูปที่ 1.1.2: ภาพจิตรกรรมไทยที่วัดภูมินทร์ อำเภอเมือง จังหวัดน่าน

1.1.2 การลบบทบรรยายบนอนิเมะ

อนิเมะคือวิดีโอภาพวิดีโอภาพการ์ตูนสไตล์ญี่ปุ่นซึ่งเป็นที่นิยมของเยาวชนไทย ในการรับชมอนิเมะ แม้ว่าเยาวชนไทยสามารถรับชมด้วยบทพากย์เสียงภาษาไทย แต่ก็สูญเสียอรรถรสของการรับชมจากบทบรรยายแบบแข็ง² (hard-

¹ภาพถ่ายที่วัดภูมินทร์ อำเภอเมือง จังหวัดน่าน; ภาพจาก <http://topicstock.pantip.com/camera/topicstock/2009/02/O7514399/O7514399.html> สืบค้นเมื่อวันที่ 23 กันยายน 2561

²บทบรรยายที่ไม่สามารถปิดหรือเปิดได้

sub) ที่เป็นภาษาต่างประเทศในบริเวณด้านล่างของภาพ ในการซ่อนแซม
อนิเมะด้วยการลับบทบรรยายภาษาต่างประเทศจะเป็นงานที่ยุ่งยากและห้าห้ายมาก เนื่องจาก

- (1) อนิเมะเป็นวิดีโอซึ่งแสดงผลประมาณ 24 เฟรม(ภาพ)ต่อวินาที
- (2) แต่ละเฟรมอาจมีหรืออาจไม่มีบทบรรยายก็ได้
- (3) แต่ละเฟรมอาจมีหรืออาจไม่มีบทบรรยายเดียวกันก็ได้
- (4) แต่ละเฟรมเป็นการแสดงผลภาพสีที่มีระดับความคมชัดสูง (high definition) ขนาดมากถึง 1920×1080 พิกเซล

ด้วยความท้าทายข้างต้น การพัฒนาขั้นตอนวิธีการต่อเติมภาพที่สามารถกำหนดโดยเมนูต่อเติมเชิงอัตโนมัติให้กับแต่ละเฟรมและประมวลผลได้แม่นยำจากการลับบทบรรยายสามารถทำงานได้แบบเรียลไทม์เป็นสิ่งจำเป็นที่หลีกเลี่ยงไม่ได้

รูปที่ 1.1.3 แสดงตัวอย่าง 1 เฟรมของอนิเมะที่มีบทบรรยายแบบแข็ง³ ที่ต้องซ่อนแซมด้วยการลับบทบรรยายออก ทั้งนี้ในการลับบทบรรยายออกจากเฟรมโดยใช้การต่อเติมภาพ เราจะเริ่มด้วยการสร้างโดยเมนูต่อเติมแบบอัตโนมัติในบริเวณบทบรรยาย จากนั้นจึงนำขั้นตอนวิธีการต่อเติมภาพแบบเร็วเพื่อลับบทบรรยายออกจากเฟรม



รูปที่ 1.1.3: 1 เฟรมของอนิเมะที่มีบทบรรยายแบบแข็ง

โครงการวิจัยนี้ ผู้วิจัยมีเป้าหมายสำคัญคือการพัฒนาขั้นตอนวิธีการต่อเติมภาพแบบเร็วและแม่นยำ ชนิดใหม่เพื่อนำไปใช้สำหรับซ่อนแซมภาพจิตรกรรมไทยและการลับบทบรรยายออกจากอนิเมะ

³ ภาพจาก <https://www.samehadaku.tv/2018/07/grand-blue-episode-1-subtitle-indonesia.html> สืบคันเมื่อวันที่ 23 กันยายน 2561

บทที่ 2

วรรณกรรมและทฤษฎีกับที่เกี่ยวข้อง

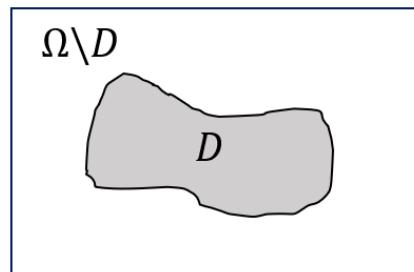
2.1 การต่อเติมภาพขาดเทา

ในการกล่าวถึงขั้นตอนวิธีการต่อเติมภาพ จะเริ่มต้นด้วยการกล่าวทบทวนเกี่ยวกับการต่อเติมภาพขาดสีเทา (grayscale image) ก่อน ดังนี้

ให้ $\Omega \subset \mathbb{R}^2$ แทนโดเมนภาพ (image domain) $D \subset \mathbb{R}^2$ แทนโดเมนต่อเติม (คูรูปที่ 2.1.1) และ $V \subset [0, \infty)$ และให้ $u : \Omega \rightarrow V$, $z : \Omega \rightarrow V$ แทนภาพที่ได้รับการซ่อมแซมและภาพที่ต้องการซ่อมแซมตามลำดับ

ในที่นี้ $\mathbf{x} = (x, y) \in \Omega$ แทนพิกัดทางกายภาพ (physical position) ของภาพ และ $u(\mathbf{x}) \in V$ แทนระดับความเข้มของภาพ (image intensity) ที่ \mathbf{x} และ Ω มีรูปร่างสี่เหลี่ยม

นอกจากนี้เราสามารถสมมติได้โดยไม่เสียหลักการสำคัญว่า $\Omega = [1, n]^2$ และ $V = [0, 1]$ เมื่อ $n > 0$ เป็นจำนวนเต็มบวก ทั้งนี้ เราจะเรียกภาพ u , z ที่นิยามข้างต้นว่าภาพขาดสีเทา



รูปที่ 2.1.1: D แทนโดเมนต่อเติม

2.1.1 ตัวแบบการต่อเติมภาพเขตสีเทาที่ใช้การแปรผันรวม

ในการต่อเติมภาพเขตสีเทา Chan และ Shen [1] ได้นำเสนอตัวแบบเชิงการแปรผัน (variational model) ที่ใช้รากวารีโรซ์เซชันแบบการแปรผันรวม (Total variation based regularization) โดยพัฒนาต่อจากตัวแบบ ROF สำหรับการกำจัดสัญญาณรบกวน [2] ซึ่งตัวแบบเชิงการแปรผันนี้กำหนดโดย

$$\min_u \{ \mathcal{J}(u) = \frac{1}{2} \int_{\Omega} \lambda(u - z)^2 d\Omega + \int_{\Omega} |\nabla u| d\Omega \} \quad (2.1.1)$$

เมื่อ

$$\lambda = \lambda(x) = \begin{cases} \lambda_0, & x \in \Omega \setminus D \\ 0, & x \in D \end{cases} \quad (2.1.2)$$

แทนพารามิเตอร์รากวารีโรซ์เซชัน (regularization parameter) และ $\lambda_0 > 0$

โดยแคลคูลัสของการแปรผัน (Calculus of variations) จะได้สมการอยเลอร์ลากรางจ์ที่เกี่ยวข้องกับ (2.1.1) เป็น

$$\begin{cases} -\nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) + \lambda(u - z) = 0, & x \in (1, n)^2 \\ \frac{\partial u}{\partial n} = 0, & x \in \partial\Omega \end{cases} \quad (2.1.3)$$

เมื่อ n แทนเวกเตอร์หน่วยที่ตั้งฉากกับขอบของภาพ

2.2 วิธีการเชิงตัวเลขสำหรับการต่อเติมภาพเขตสี

ต่อไปจะกล่าวทบทวนวิธีการเชิงตัวเลขสำหรับแก้สมการเชิงอนุพันธ์ย่อยใน (2.1.3)

2.2.1 วิธีการเดินเวลาแบบชัดแจ้ง (explicit time marching method)

คณะวิจัย [2] ได้แนะนำวิธีการเชิงตัวเลขสำหรับการกำจัดสัญญาณรบกวนโดยใช้วิธีการเดินเวลาแบบชัดแจ้ง ซึ่งสามารถประยุกต์เป็นวิธีเชิงตัวเลขสำหรับการต่อเติมภาพได้ดังนี้

เริ่มจากการแนะนำตัวแปรเวลาสังเคราะห์ (time artificial variable) จากนั้นหาค่าตอบแบบสภาวะ

คงตัว (steady-state solution) ในขณะที่ $t \rightarrow \infty$ ของสมการเชิงอนุพันธ์อย่างไม่เป็นเชิงเส้นที่ขึ้นอยู่กับเวลา

$$u(\mathbf{x}, t_{k+1}) = u(\mathbf{x}, t_k) + \tau \left(\nabla \cdot \left(\frac{\nabla u(\mathbf{x}, t_k)}{|\nabla u(\mathbf{x}, t_k)|} \right) + \lambda(\mathbf{x})(u(\mathbf{x}, t_k) - z(\mathbf{x})) \right), \quad u(\mathbf{x}, t_0) = z \quad (2.2.1)$$

เมื่อ $t_k = t_0 + k\tau$ ($\tau > 0$) แทนชั้นเวลาที่ k และ $t_0 = 0$ แทนชั้นเวลาเริ่มต้น

วิธีเดินเวลาแบบชัดเจ็บสำหรับภาพเฉพาะมีขั้นตอนวิธีดังนี้

Algorithm 1: วิธีการเดินเวลาแบบชัดเจ็บสำหรับการต่อเติมภาพที่ใช้การประผันรวม

Input:

- u คือรูปภาพที่ต้องการต่อเติม
- λ คือพารามิเตอร์เริกกวาร์เรเช่น ที่ได้กล่าวถึงในสมการ (2.1.2)
- β เป็นจำนวนจริงบวกที่ใช้เพื่อหลีกเลี่ยงการหารด้วยศูนย์
- τ เป็นจำนวนจริงบวกที่เป็นตัวแปรเดินเวลา
- N เป็นจำนวนเต็มบวกสำหรับกำหนดจำนวนรอบที่ทำงาน
- ε เป็นจำนวนจริงบวกของค่าความคลาดเคลื่อนสัมพัทธ์

Output: รูปภาพที่ผ่านการต่อเติมแล้ว

```

 $u \leftarrow \text{ExplicitTimeMarching}(u, \lambda, \beta, \tau, N, \varepsilon)$ 
initialize  $i = 0; z = u; err = 1$ 
while  $i < N$  and  $err > \varepsilon$  do
     $u^{old} = u$ 
     $u = u + \tau \left( \nabla \cdot \left( \frac{\nabla u}{\sqrt{u_x^2 + u_y^2 + \beta}} \right) + \lambda(u - z) \right)$ 
     $err = \frac{\|u - u^{old}\|}{\|u\|}$ 
     $i = i + 1$ 
end

```

2.2.2 วิธีการทำซ้ำแบบจุดตรึง (fixed-point iteration method)

คณะวิจัย [3] ได้แนะนำวิธีการเชิงตัวเลขสำหรับการกำจัดสัญญาณรบกวนโดยใช้วิธีการทำซ้ำแบบจุดตรึง ซึ่งสามารถประยุกต์เป็นวิธีเชิงตัวเลขสำหรับการต่อเติมภาพได้ดังนี้

เริ่มจากแนะนำดัชนีการทำซ้ำแบบจุดตรึง $\nu = 0, 1, 2, \dots$ และนิยามรูปแบบการทำซ้ำโดย

$$-\nabla \cdot \left(\frac{\nabla u^{[\nu+1]}}{|\nabla u|^{[v]}} \right) + \lambda(u^{[\nu+1]} - z) = 0, \quad u^{[0]} = z \quad (2.2.2)$$

เนื่องจาก $\frac{1}{|\nabla u|} = \frac{1}{\sqrt{u_x^2 + u_y^2}} \rightarrow \infty$ ในบริเวณที่ u มีความเข้มสีเป็นเอกพันธุ์ ($u(x) = \text{ค่าคงตัว}$) เพื่อหลีกเลี่ยงปัญหาเชิงตัวเลขจะเกิดขึ้นใน (2.2.1) และ (2.2.2) เราจะใช้

$$|\nabla u| \approx |\nabla u|_\beta = \sqrt{u_x^2 + u_y^2 + \beta}, \quad 0 < \beta \ll 1$$

วิธีการทำซ้ำแบบจุดตรึงมีขั้นตอนดังนี้

Algorithm 2: วิธีการทำซ้ำจุดตรึงสำหรับการต่อเติมภาพที่ใช้การแปรผันรวม

Input:

u คือรูปภาพที่ต้องการต่อเติม

λ คือพารามิเตอร์เร็กกิวลาร์ไซเรชัน ที่ได้กล่าวถึงในสมการ (2.1.2)

β เป็นจำนวนจริงบวกที่ใช้เพื่อหลีกเลี่ยงการหารด้วยศูนย์

N เป็นจำนวนเต็มบวกสำหรับกำหนดจำนวนรอบที่ทำงาน

ε เป็นจำนวนจริงบวกของค่าความคลาดเคลื่อนสัมพัทธ์

Output: รูปภาพที่ผ่านการต่อเติมแล้ว

$u \leftarrow \text{FixedPoint}(u, z, \lambda, \beta, N, \varepsilon)$

initialize $i = 0; u = z; err = 1$

while $i < N$ and $err > \varepsilon$ do

$u^{old} = u$

$u = \text{GaussSeidel}(u, z, \lambda, \beta, N_{gs})$

$err = \frac{\|u - u^{old}\|}{\|u\|}$

$i = i + 1$

end

Algorithm 3: การทำซ้ำแก๊ส-ไซเดล สำหรับวิธีการจุดตรึง

```

 $u \leftarrow GaussSeidel(u, \lambda, \beta, N_{gs})$ 
initialize  $k = 0$ 
 $D(u)_{i,j} = \frac{1}{\sqrt{u_x^2 + u_y^2 + \beta}}, 1 \leq i \leq n_x, 1 \leq j \leq n_y$ 
while  $k < N_{gs}$  do
 $u_{i,j}^{k+1} = \frac{\lambda_{i,j} z_{i,j} + (D_{i,j}(u_{i+1,j}^k + u_{i,j+1}^k) + D_{i-1,j} u_{i-1,j}^{k+1} + D_{i,j-1} u_{i,j-1}^{k+1})}{\lambda_{i,j} + (2D_{i,j} + D_{i-1,j} + D_{i,j-1})}$ 
 $k = k + 1$ 
end

```

จาก (2.2.1) และ (2.2.2) เราพบว่าเมื่อ β มีค่าน้อยลงมากขึ้นเท่าไหร่ ความแม่นยำของตัวแบบ (2.1.1) ยิ่งมีมากขึ้นเท่านั้น นอกจากนี้ เรายังพบอีกว่า การแก้สมการ (2.2.1) และ (2.2.2) ยิ่งมีความยุ่งยากมากขึ้นสำหรับ β ที่มีค่าน้อยๆ

เพื่อเอาชนะความยากเชิงตัวเลขนี้ คณะวิจัยโดย [4] ได้แนะนำวิธีการสปริทเบรกแมนซึ่งสามารถกล่าวถึงพอกสังเขป ดังนี้

2.2.3 วิธีการสปริทเบรกแมน (Split Bregman method)

เริ่มจากการแนะนำเวกเตอร์เสริม \mathbf{w} พารามิเตอร์เบรกแมน (Bregman parameter) \mathbf{b} และพารามิเตอร์เพนล็อก (penalty parameter) $\theta > 0$ และเขียน (2.1.1) ใหม่ ดังนี้

$$\min_{u,\mathbf{w}} \{ \mathcal{J}(u, \mathbf{w}) = \frac{1}{2} \int_{\Omega} \lambda(u - z)^2 d\Omega + \int_{\Omega} |\mathbf{w}| d\Omega + \frac{\theta}{2} \int_{\Omega} (\mathbf{w} - \nabla u + \mathbf{b}) d\Omega \} \quad (2.2.3)$$

สำหรับการหาคำตอบของ (2.2.3) เราจะใช้วิธีการหาค่าต่อต้าที่สุดแบบสลับ (alternating minimization method) โดยเริ่มจากการตั้ง \mathbf{w}^{old} และ \mathbf{b}^{old} จากนั้นแก้ปัญหาอย่างสำหรับ u

$$u^{\text{New}} = \arg \min_u \{ \mathcal{J}_1(u) = \frac{1}{2} \int_{\Omega} \lambda(u - z)^2 d\Omega + \frac{\theta}{2} \int_{\Omega} (\mathbf{w}^{\text{old}} - \nabla u + \mathbf{b}^{\text{old}}) d\Omega \} \quad (2.2.4)$$

ต่อไปใช้ u^{New} ที่ได้จากการแก้ปัญหาอย่าง (2.2.4) เพื่อแก้ปัญหาอย่างสำหรับ \mathbf{w}

$$\mathbf{w}^{\text{New}} = \arg \min_{\mathbf{w}} \{ \mathcal{J}_2(\mathbf{w}) = \int_{\Omega} |\mathbf{w}| d\Omega + \frac{\theta}{2} \int_{\Omega} (\mathbf{w} - \nabla u^{\text{New}} + \mathbf{b}^{\text{old}}) d\Omega \} \quad (2.2.5)$$

สุดท้ายจึงปรับปรุงพารามิเตอร์เบรกเมนโดย

$$\mathbf{b}^{\text{New}} = \mathbf{b}^{\text{old}} + \nabla u^{\text{New}} - \mathbf{w}^{\text{New}} \quad (2.2.6)$$

ดำเนินการเช่นนี้จนกระทั่ง $\|u^{\text{new}} - u^{\text{old}}\| < \epsilon_1$ หรือ $\text{New} > \epsilon_2$ เมื่อ $\epsilon_1, \epsilon_2 > 0$

วิธีการสปริทเบรกเมนมีขั้นตอนวิธีดังนี้

Algorithm 4: วิธีสปริทเบรกเมนสำหรับการต่อเติมภาพที่ใช้การแปรผันรวม

Input:

u คือรูปภาพที่ต้องการต่อเติม

λ คือพารามิเตอร์เร็กกิวลาร์ไซเรชัน ที่ได้ถูกล่างถึงในสมการ (2.1.2)

θ คือพารามิเตอร์เพนัลตี้ ซึ่งเป็นจำนวนจริงบวก

N_{gs} เป็นจำนวนเต็มบวกสำหรับกำหนดจำนวนรอบที่ทำงานของการทำเกาส์-ไซเดล

N เป็นจำนวนเต็มบวกสำหรับกำหนดจำนวนรอบที่ทำงานของสปริทเบรกเมน

ε เป็นจำนวนจริงบวกของค่าความคลาดเคลื่อนสัมพัทธ์

Output: รูปภาพที่ผ่านการต่อเติมแล้ว

$u \leftarrow \text{SplitBregman}(u, \lambda, \theta, N_{gs}, N, \varepsilon)$

initialize $i = 0$, $\mathbf{b} = \vec{0}$, $\mathbf{w} = \vec{0}$, $z = u$

while $i < N$ and $\text{err} > \varepsilon$ do

$u^{\text{old}} = u$; $w^{\text{old}} = w$; $b^{\text{old}} = b$;

$u = \arg \min_u \{\mathcal{J}_1(u) = \frac{1}{2} \int_{\Omega} \lambda(u - z)^2 d\Omega + \frac{\theta}{2} \int_{\Omega} (\mathbf{w}^{\text{old}} - \nabla u + \mathbf{b}^{\text{old}}) d\Omega\}$

$w = \arg \min_w \{\mathcal{J}_2(\mathbf{w}) = \int_{\Omega} |\mathbf{w}| d\Omega + \frac{\theta}{2} \int_{\Omega} (\mathbf{w} - \nabla u^{\text{New}} + \mathbf{b}^{\text{old}}) d\Omega\}$

$b = b^{\text{old}} + \nabla u - \mathbf{w}$

$\text{err} = \frac{\|u - u^{\text{old}}\|}{\|u\|}$

$i = i + 1$

end

หมายเหตุ:

(1) ผลเฉลยของ $u = \arg \min_u \mathcal{J}_1(u)$ กำหนดโดยการแก้ปัญหาผลเฉลยของ

$$-\theta \Delta u + \lambda u = \lambda z - \theta \nabla \cdot (\mathbf{w} - \mathbf{b})$$

โดยใช้วิธีการไฟน์ติดไฟอเรนจ์และวิธีการเกาส์-ไซเดลจำนวน N_{gs} รอบ

(2) ผลเฉลยของ $\mathbf{w} = \arg \min_{\mathbf{w}} \mathcal{J}_2(\mathbf{w})$ กำหนดโดย

$$\mathbf{w} = \max \left\{ (\nabla u + \mathbf{b}) - \frac{1}{\theta}, 0 \right\}$$

2.3 ตัวแบบการต่อเติมภาพสีที่ใช้การแปรผันรวม

ต่อไปเราจะพิจารณาภาพสีในระบบสี RGB นั่นคือ เราสมมติว่า

$$\mathbf{u} = (u_1, u_2, u_3)^\top, \mathbf{z} = (z_1, z_2, z_3)^\top : \Omega \rightarrow V^3$$

เมื่อ $u_1, u_2, u_3 : \Omega \rightarrow V$ และ $z_1, z_2, z_3 : \Omega \rightarrow V$ แทนภาพในเขตสีแดง สีเขียว และสีน้ำเงินของ \mathbf{u}, \mathbf{z} ตามลำดับ

ในทำนองเดียวกันกับตัวแบบการต่อเติมภาพแฉลสีท่าที่ใช้การแปรผันรวม ตัวแบบการต่อเติมภาพสีที่ใช้การแปรผันรวมสามารถเขียนได้ดังนี้

$$\min_{\mathbf{u}} \{ \bar{\mathcal{J}}(\mathbf{u}) = \bar{\mathcal{D}}(\mathbf{u}, \mathbf{z}) + \bar{\mathcal{R}}(\mathbf{u}) \} \quad (2.3.1)$$

เมื่อ

$$\bar{\mathcal{D}}(\mathbf{u}, \mathbf{z}) = \frac{1}{2} \int_{\Omega} \lambda (u_1 - z_1)^2 d\Omega + \frac{1}{2} \int_{\Omega} \lambda (u_2 - z_2)^2 d\Omega + \frac{1}{2} \int_{\Omega} \lambda (u_3 - z_3)^2 d\Omega$$

และ

$$\bar{\mathcal{R}}(\mathbf{u}) = \int_{\Omega} |\nabla u_1| d\Omega + \int_{\Omega} |\nabla u_2| d\Omega + \int_{\Omega} |\nabla u_3| d\Omega$$

หลังจากใช้วิธีสปีทเบรกแมนกับ (2.3.1) จะได้

$$\begin{aligned} \min_{\mathbf{u}, \mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3} \{ \bar{\mathcal{J}}(\mathbf{u}, \mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3) = \bar{\mathcal{D}}(\mathbf{u}, \mathbf{z}) + \sum_{l=1}^3 \int_{\Omega} |\mathbf{w}_l| d\Omega \\ + \frac{\theta_l}{2} \sum_{l=1}^3 \int_{\Omega} (\mathbf{w}_l - \nabla u_l - \mathbf{b}_l)^2 d\Omega \}, \quad \theta_l > 0 \end{aligned} \quad (2.3.2)$$

ขั้นตอนวิธีสปีทเบรกแมนสำหรับภาพสี แสดงได้ดังนี้

Algorithm 5: วิธีการสปีทเบรกแมนสำหรับการต่อเติมภาพสีด้วยวิธีการแปรผันรวม

```

 $\mathbf{u} \leftarrow SplitBregmanColor(\mathbf{u}, \lambda, \theta, N_{gs}, N, \varepsilon)$ 
for  $l = 1 : 3$  do
     $u_l \leftarrow SplitBregman(u_l \lambda, \theta, N_{gs}, N, \varepsilon)$ 
end

```

2.4 การวัดประสิทธิภาพของภาพที่ผ่านกระบวนการต่อเติม

การประเมินคุณภาพของการต่อเติมภาพของวิธีการเชิงตัวเลข จะใช้ค่า PSNR [6] และ SSIM [7] โดย

$$\begin{aligned} \text{PSNR} &= 10 \cdot \log_{10} \left(\frac{1}{\sqrt{\text{MSE}}} \right) \\ \text{SSIM}(u, \tilde{u}) &= \frac{(2\mu_u \mu_{\tilde{u}} + 0.0001)(2\sigma_{u\tilde{u}} + 0.0009)}{(\mu_u^2 + \mu_{\tilde{u}}^2 + 0.0001)(\sigma_u^2 + \sigma_{\tilde{u}}^2 + 0.0009)} \end{aligned}$$

- MSE คือค่าค่าคาดเคลื่อนกำลังสองเฉลี่ยของภาพ โดยที่ $\text{MSE} = \left(\frac{1}{nx \times ny} \sum (u - \bar{u})^2 \right)$
- u แทนภาพต้นฉบับ
- \tilde{u} แทนภาพต้นฉบับ และภาพที่ได้จากการซ่อมแซมโดยวิธีเชิงตัวเลข

- μ_u คือค่าเฉลี่ยของ u
- $\mu_{\tilde{u}}$ คือค่าเฉลี่ยของ \tilde{u}
- σ_u คือความแปรปรวนของ u
- $\sigma_{\tilde{u}}$ คือความแปรปรวนของ \tilde{u}

หมายเหตุ:

(1) ถ้า $\tilde{u} \rightarrow u$ และ PSNR $\rightarrow \infty$ หมายถึง ภาพที่ซ่อมแซม \tilde{u} มีคุณภาพดี

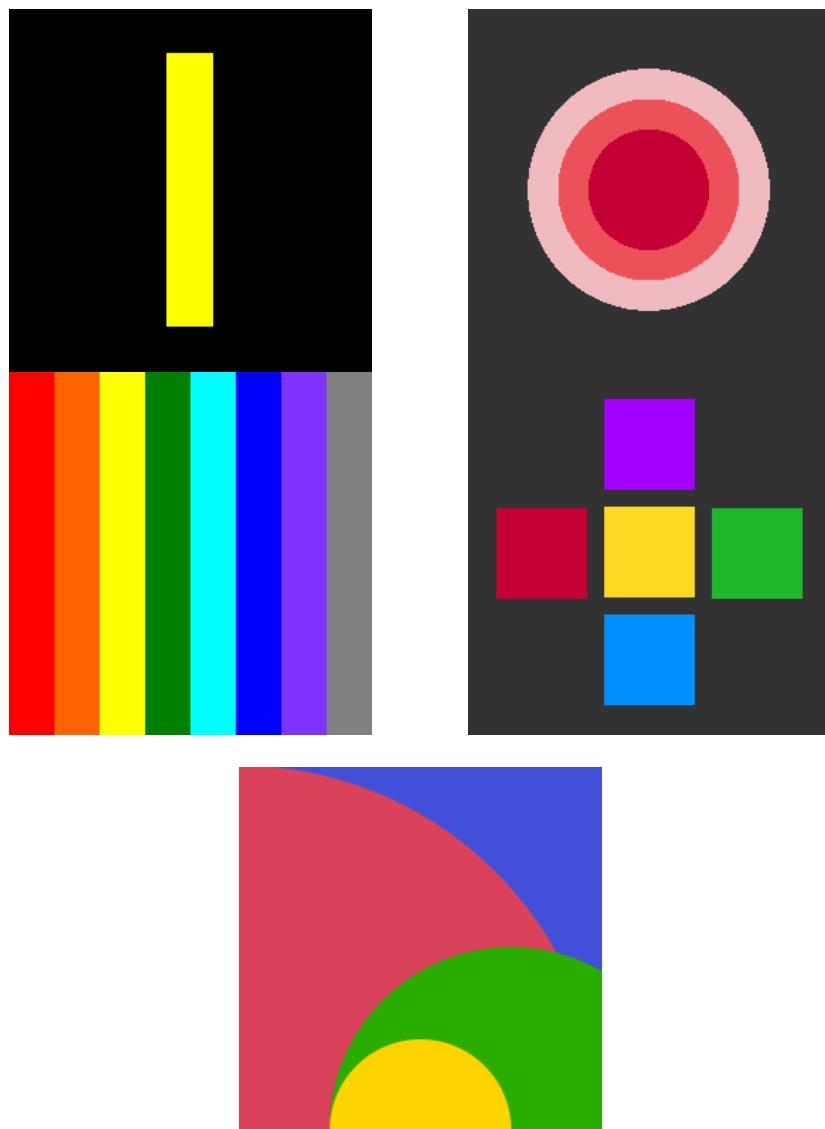
(2) ถ้า $SSIM(u, \tilde{u}) = 1$ หมายถึง ภาพที่ซ่อมแซม \tilde{u} มีคุณภาพดี

บทที่ 3

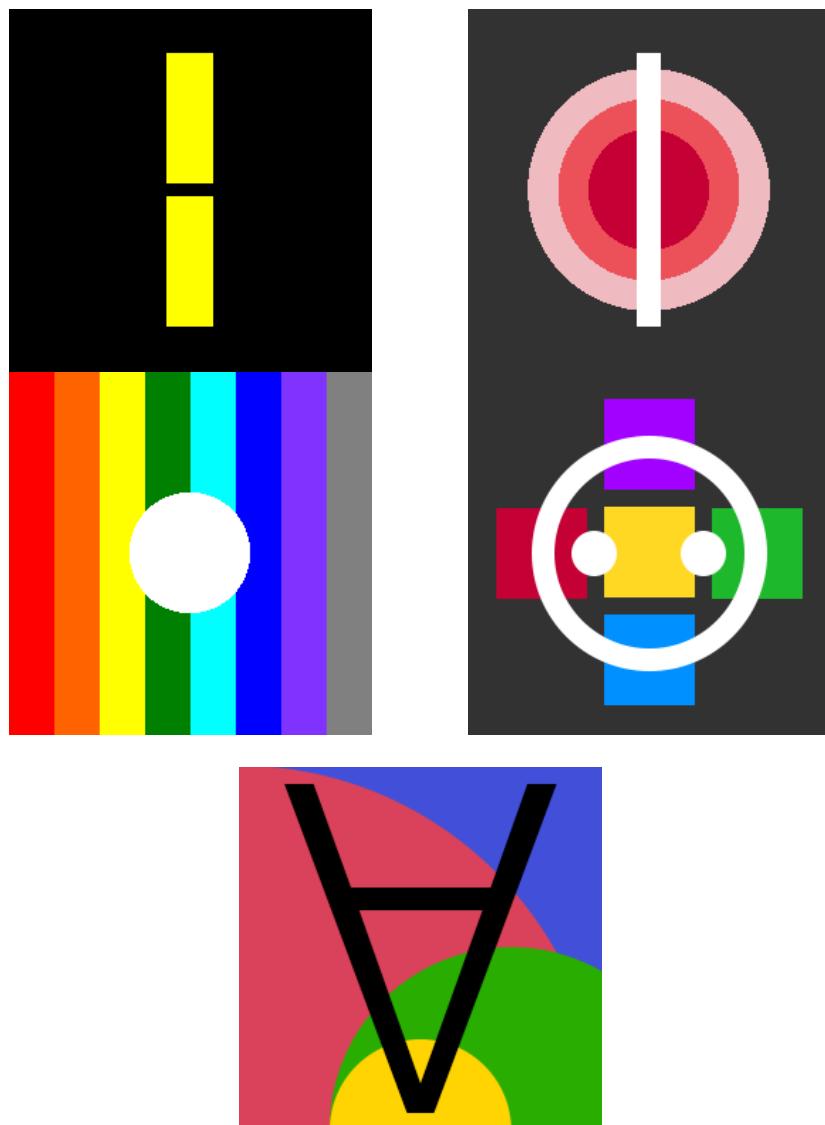
การทดลองเชิงตัวเลข

3.1 การซ่อมแซมภาพจิตรกรรมไทยโบราณ

สำหรับการซ่อมแซมจิตรกรรมไทยโบราณ ผู้วิจัยจะเริ่มจากการทำการปรับปรุงขึ้นตอนวิธีเชิงตัวเลขที่มีอยู่แล้ว โดยใช้ภาพสีที่ได้สังเคราะห์ขึ้นทั้งสิ้น 5 ภาพ โดยแต่ละภาพมีขนาด 256×256 พิกเซล ดังรูปที่ 3.1.1



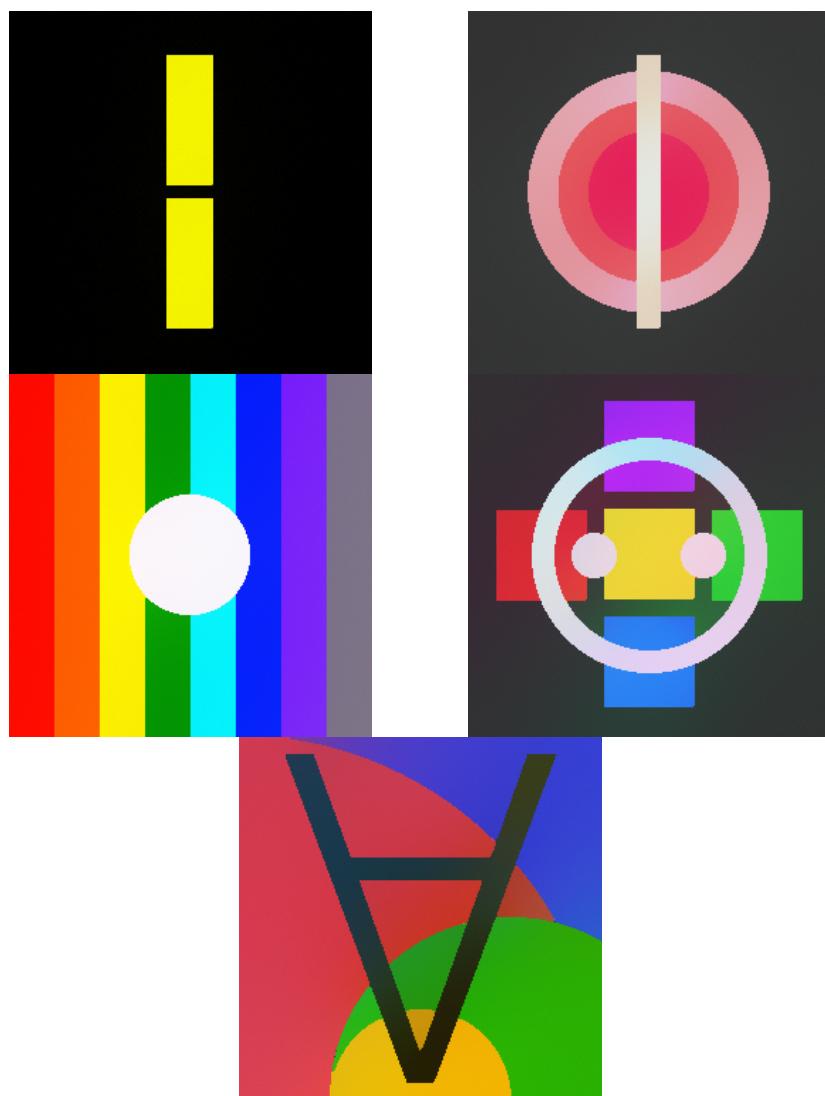
รูปที่ 3.1.1: ภาพต้นฉบับ



รูปที่ 3.1.2: ภาพที่จะทำการซ่อมแซม

3.1.1 การเปรียบเทียบประสิทธิภาพขั้นตอนวิธีเชิงตัวเลขที่มีอยู่แล้ว

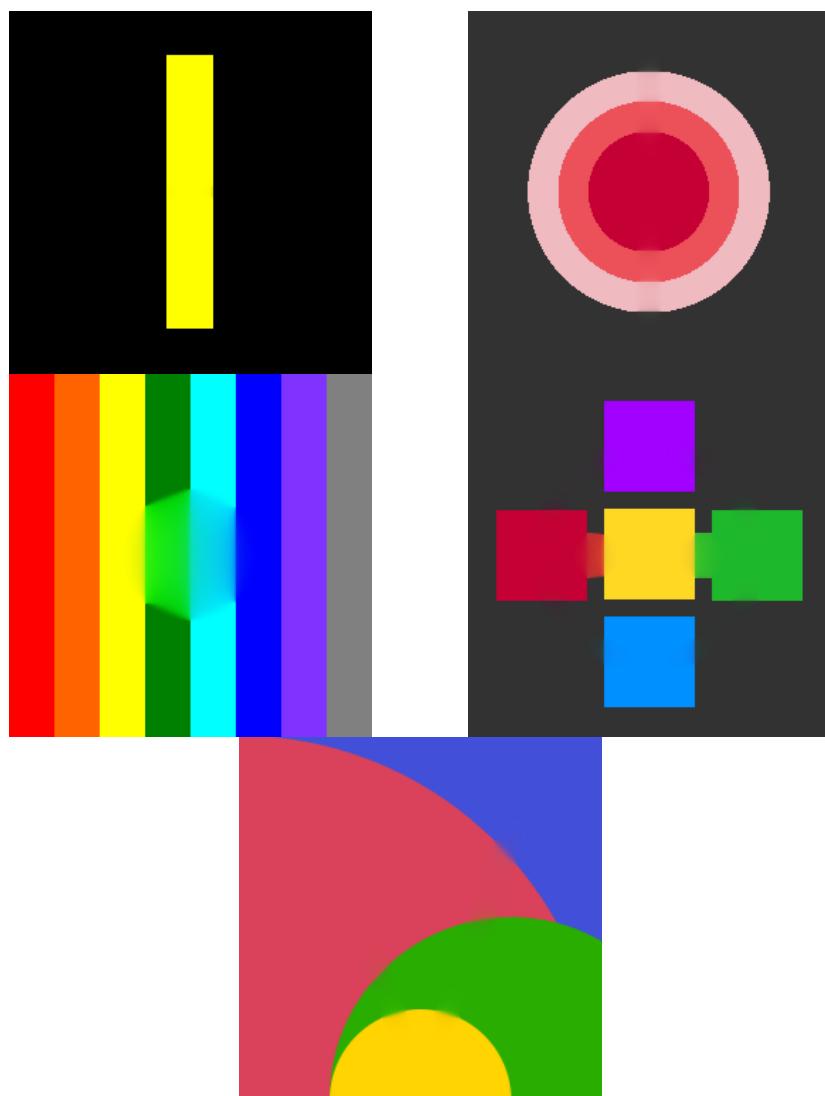
การทดสอบประสิทธิภาพจะใช้ $\varepsilon = 1 \times 10^{-4}$ และ $N = 10,000$ โดยรูปที่ 3.1.3 - 3.1.5 และตารางที่ 3.1 - 3.3 แสดงผลการซ่อมแซมภาพสังเคราะห์ทั้ง 5 ภาพ



รูปที่ 3.1.3: ผลการซ่อมแซมจากการเดินเวลา

รูปภาพ	เวลาประมาณ (วินาที)	PSNR (dB)	SSIM
1	82.40	25.17	0.9997
2	127.36	17.92	0.9980
3	116.39	13.33	0.9941
4	160.59	12.40	0.9927
5	116.66	14.79	0.9958
เฉลี่ย	120.68	16.72	0.9960

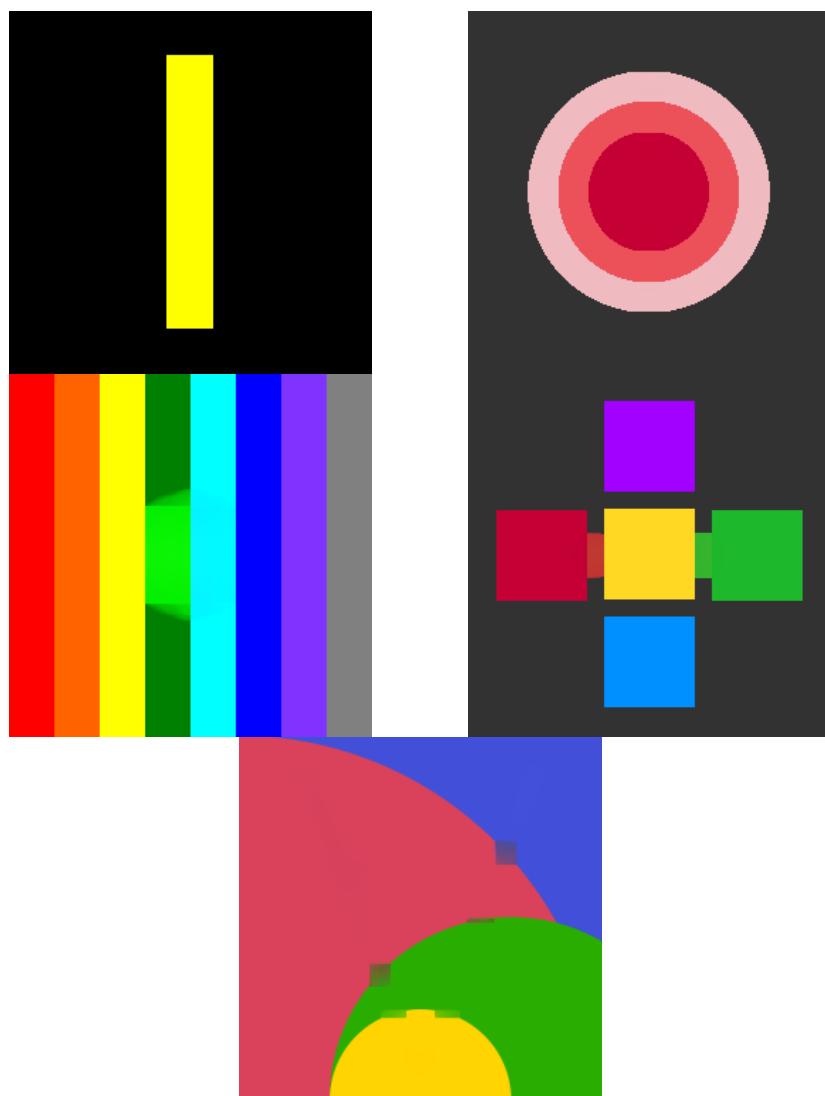
ตารางที่ 3.1: ผลการซ่อมแซมวิธีการเดินเวลา



รูปที่ 3.1.4: ผลการซ่อมแซมจากการทำซ้ำแบบบุกตีรึ่ง

รูปภาพ	เวลาประมาณ (วินาที)	PSNR (dB)	SSIM
1	24.97	60.95	1.0000
2	53.06	37.69	1.0000
3	190.64	25.17	0.9997
4	50.63	28.81	0.9999
5	54.74	40.73	1.0000
เฉลี่ย	74.81	38.67	0.9999

ตารางที่ 3.2: ผลการซ่อมแซมของวิธีการทำข้ามแบบจุดตรึง



รูปที่ 3.1.5: ผลการซ้อมแซมจากวิธีการสปริทเบรกเมน

รูปภาพ	เวลาประมาณ (วินาที)	PSNR (dB)	SSIM
1	3.39	71.54	1.0000
2	10.74	37.08	1.0000
3	24.50	26.08	0.9997
4	15.80	29.61	0.9999
5	15.85	32.78	1.0000
เฉลี่ย	14.06	39.42	0.9999

ตารางที่ 3.3: ผลการซ่อมแซมของวิธีสปริทเบรกเมน

ประสิทธิภาพของวิธีการเชิงตัวเลขทั้ง 3 วิธี สามารถสรุปได้ดังนี้

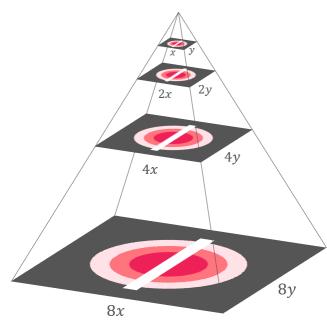
วิธีการ	เวลาประมาณ (วินาที)	PSNR (dB)	SSIM
การเดินเวลา	120.68	16.72	0.9960
การทำชาจุดร่อง	74.81	38.67	0.9999
การสปริทเบรกเมน	14.06	39.42	0.9999

ตารางที่ 3.4: แสดงการซ่อมแซมเฉลี่ยของวิธีการเชิงตัวเลข

จากทั้ง 3 วิธีที่ได้ทดสอบ จะเห็นได้ว่าวิธีการสปริทเบรกเมนใช้เวลาน้อยกว่าวิธีอื่น และมีคุณภาพ ซึ่งพิจารณาจาก ค่า PSNR และค่า SSIM มากกว่าวิธีอื่น ผู้วิจัยจึงสนใจทำการปรับปรุงวิธีสปริทเบรกเมนให้มีประสิทธิภาพ ยิ่ง

3.1.2 ขั้นตอนวิธีเชิงตัวเลขสำหรับต่อเติมภาพชนิดใหม่

เพื่อให้วิธีการสปริทเบรกเมนประมาณภาพได้รวดเร็วขึ้น ผู้วิจัยได้พัฒนากระบวนการกำหนดคำตอบเริ่มต้น โดยวิธีการ มัลติรีโซลูชัน (multi-resolution method) หรือวิธีการพีระมิดรูปภาพ (pyramid method) [5] เริ่มจากการย่อขนาดรูปลงครึ่งหนึ่งโดยใช้วิธี Bilinear Interpolation จนกระทั่งถึงระดับความคมชัดที่ต้องการ จากนั้นทำการต่อเติมภาพขนาดเล็ก และนำผลลัพธ์ที่ได้จากการขยายภาพขึ้นสองเท่าโดยใช้ Bi-linear Interpolation เป็นคำตอบเริ่มต้นสำหรับการต่อเติมภาพในขั้นถัดไป



รูปที่ 3.1.6: วิธีการพีระมิดรูปภาพ

ขั้นตอนวิธีสำหรับการทำพีระมิดรูปภาพสำหรับการต่อเติมภาพแบบสปริตเบรกแม่นเพื่อให้ประมวล

ผลได้เร็วขึ้นนั้นสามารถสรุปได้ดังนี้

Algorithm 6: วิธีสปริตเบรกเม้นท์ใช้พิรมิตรูปภาพ

Input:

- u คือรูปภาพที่ต้องการต่อเติม
- λ คือพารามิเตอร์เริกกวิลาร์เรเซชัน ที่ได้กล่าวถึงในสมการ (2.1.2)
- θ คือพารามิเตอร์เพนด์ที่ ซึ่งเป็นจำนวนจริงบวก
- N_{gs} เป็นจำนวนเต็มบวกสำหรับกำหนดจำนวนรอบที่ทำงานของการทำเกาส์-ไซเดล
- c ตัวแปรช่วยสำหรับบอกความลึก ให้กำหนดเป็น 1
- m คือ ระดับความลึกของพิรมิตรูปภาพ เป็นจำนวนเต็มบวก
- N_0 จำนวนรอบการทำสปริตเบรกเม้นท์ชั้นละเอียดสุด
- N_1 จำนวนรอบการทำสปริตเบรกเม้นท์ชั้นต่างๆ
- N_2 จำนวนรอบการทำสปริตเบรกเม้นท์ชั้นหยาบสุด

Output: รูปภาพที่ผ่านการต่อเติมแล้ว

```

 $u \leftarrow MultiSplitBregmanColor(u, \lambda, \theta, N_{gs}, N_0, N_1, N_2, \varepsilon, c, m)$ 
Initialize  $height$  = ความสูงของภาพ  $u$ ,  $width$  = ความกว้างของภาพ  $u$ 

if  $c < m$  then
     $\mathbf{x} = Bilinear(u, \lfloor width * 0.5 \rfloor, \lfloor height * 0.5 \rfloor)$ 
     $y = Bilinear(\lambda, \lfloor width * 0.5 \rfloor, \lfloor height * 0.5 \rfloor)$ 
     $r = MRSBC(\mathbf{x}, \mathbf{z}, y, \lambda, \theta,$ 
         $N_{gs}, N_0, N_1, N_2, \varepsilon, c + 1, m)$ 
     $u = Bilinear(r, width, height)$ 
end

if  $c = 1$  then
     $N_{SB} = N_0$ 
else if  $c = m$  then
     $N_{SB} = N_2$ 
else
     $N_{SB} = N_1$ 
end

 $u = SplitBregmanColor(u, \lambda, \theta, N_{gs}, N_{SB}, \varepsilon)$ 

```

Algorithm 7: Bilinear Interpolation

```

 $J \leftarrow \text{Bilinear}(I, x, y)$ 
Initialize  $v = \text{ความสูงของภาพ } I, w \text{ คือความกว้างของภาพ } I,$ 
 $S_R = \frac{c}{a}, S_C = \frac{d}{b}, r = 1, 2, \dots, v, c = 1, 2, \dots, w,$ 
 $r' = 1, 2, \dots, x, c' = 1, 2, \dots, y,$ 
 $r_f = \lfloor r' \cdot S_R \rfloor$ 
 $c_f = \lfloor c' \cdot S_C \rfloor$ 
 $\Delta r = r_f - r$ 
 $\Delta c = c_f - c$ 
 $J(r', c') = I(r, c) \cdot (1 - \Delta r) \cdot (1 - \Delta c)$ 
 $+ I(r + 1, c) \cdot \Delta r \cdot (1 - \Delta c)$ 
 $+ I(r, c + 1) \cdot (1 - \Delta r) \cdot \Delta c$ 
 $+ I(r + 1, c + 1) \cdot \Delta r \cdot \Delta c$ 

```

ตารางที่ 3.5 แสดงผลการเปรียบเทียบประสิทธิภาพของขั้นตอนวิธีการที่ 6 ภายใต้การเปลี่ยนแปลงจำนวนรอบของการทำซ้ำของวิธีการสปริทเบรคแมนบนภาพที่มีความคมชัด 256×256 พิกเซล ตัวอย่าง เช่น $10/3/3/10000$ หมายถึงที่ระดับความคมชัดทั้งหมดเป็น 32×32 พิกเซลจะทำซ้ำไม่เกิน 10 ครั้ง สำหรับที่ความคมชัดละเอียดขึ้นเป็น 64×64 พิกเซลจะทำซ้ำไม่เกิน 3 ครั้ง และสำหรับที่ระดับความคมชัดเป็น 128×128 พิกเซลจะทำซ้ำ 3 ครั้ง และที่ระดับความคมชัดเป็น 256×256 จะทำซ้ำไม่เกิน 10,000 ครั้งหรือจนค่าความคลาดเคลื่อนสัมพัทธ์ต่างกันไม่เกิน 0.0001

รูปแบบการทำซ้ำ	รูปภาพ	เวลาประมาณ (วินาที)	PSNR (dB)	SSIM
ไม่ใช้พิธีกรรูปภาพ	1	4.49	71.54	1.0000
	2	13.16	37.08	1.0000
	3	29.46	26.08	0.9997
	4	20.50	29.61	0.9999
	5	19.32	32.78	1.0000
10/1/1/10000	1	2.44	69.59	1.0000
	2	11.31	37.04	1.0000
	3	23.48	27.34	0.9998
	4	16.60	29.42	0.9999
	5	13.75	33.53	1.0000
10/3/3/10000	1	2.24	69.96	1.0000
	2	10.91	37.05	1.0000
	3	21.99	27.66	0.9998
	4	12.70	29.35	0.9999
	5	11.49	33.69	1.0000
10/10/10/10000	1	1.83	71.58	1.0000
	2	7.83	37.05	1.0000
	3	16.75	28.62	0.9998
	4	11.89	29.32	0.9999
	5	8.00	34.26	1.0000
100/1/1/10000	1	1.43	67.63	1.0000
	2	7.17	37.10	1.0000
	3	20.86	27.70	0.9998
	4	12.80	29.64	0.9999
	5	9.17	33.14	1.0000
100/3/3/10000	1	1.68	71.18	1.0000
	2	7.41	37.11	1.0000
	3	21.08	28.00	0.9998
	4	13.28	29.38	0.9999
	5	7.96	33.34	1.0000
100/10/10/10000	1	1.76	71.56	1.0000
	2	7.32	37.04	1.0000
	3	16.62	28.65	0.9998
	4	13.18	29.39	0.9999
	5	7.45	33.94	1.0000

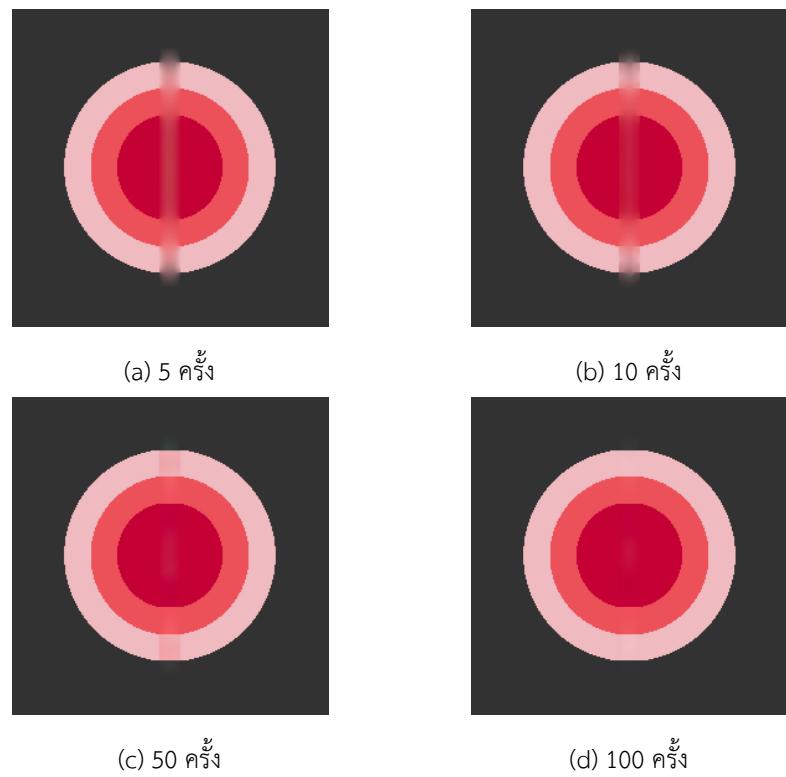
ตารางที่ 3.5: ผลการซ่อมแซมภาพโดยวิธีการเชิงตัวเลขที่นำเสนอ

รูปแบบการทำข้าว	เวลาประมาณ (วินาที)	PSNR (dB)	SSIM
ไม่ใช้พิระมิดรูปภาพ	17.38	39.42	0.9999
10/1/1/10000	13.52	39.38	0.9999
10/3/3/10000	11.86	39.54	0.9999
10/10/10/10000	9.26	40.17	0.9999
100/1/1/10000	10.28	39.04	0.9999
100/3/3/10000	10.28	39.80	0.9999
100/10/10/10000	9.27	40.12	0.9999

ตารางที่ 3.6: ผลการซ่อมแซมภาพโดยวิธีการเชิงตัวเลขที่นำเสนอด้วยรูปของค่าเฉลี่ยของผลที่ได้จากตารางที่ 3.5

จากตารางที่ 3.6 สังเกตว่า ยิ่งจำนวนการทำข้าวในชั้นที่รูปภาพมีขนาดเล็กจำนวนมากครั้ง จะยิ่งทำให้เวลาประมาณผลที่ใช้ในการต่อเติมภาพใช้เวลาน้อยลง

นอกจากนี้แล้ว ผู้วิจัยยังได้สังเกตอีกว่า การทำข้าว จะสูญเสียเร็วในช่วงแรก งานนั้นความเร็วในการถูเข้าจะลดลง ซึ่งทำให้การทำข้าวเพียงไม่กี่ครั้งในระดับความคมชัดเดิม มีผลการซ่อมแซมภาพจนแสดงความคล้ายคลึงกับภาพต้นฉบับได้



รูปที่ 3.1.7: พีระมิดที่ลำดับการทำข้าเป็น 10/10/10 และที่ระดับความคมชัดลงทะเบียดสุดใช้จำนวนการทำข้าที่ต่างกัน

ผู้วิจัยจึงกำหนดให้การทำข้าในระดับความละเอียดสุดเท่ากับ 10 ครั้ง และพบว่าได้ผลการซ่อมแซมดังตารางที่ 3.7

รูปแบบการทำข้าว ไม่ใช้พรมีดรูปภาพ	รูปภาพ	เวลาประมาณ (วินาที)	PSNR (dB)	SSIM
10/1/1/10	1	0.30	26.71	0.9998
	2	0.39	18.39	0.9982
	3	0.38	13.66	0.9944
	4	0.40	12.86	0.9934
	5	0.38	14.69	0.9956
10/3/3/10	1	0.29	40.10	1.0000
	2	0.41	31.28	0.9999
	3	0.46	16.51	0.9970
	4	0.47	26.56	0.9998
	5	0.39	28.25	0.9998
10/10/10/10	1	0.28	42.53	1.0000
	2	0.36	32.91	1.0000
	3	0.35	16.88	0.9972
	4	0.34	27.06	0.9998
	5	0.34	29.76	0.9999
100/1/1/10	1	0.31	50.06	1.0000
	2	0.41	34.01	1.0000
	3	0.38	18.19	0.9980
	4	0.39	27.50	0.9998
	5	0.40	33.05	1.0000
100/3/3/10	1	0.27	43.97	1.0000
	2	0.37	31.28	0.9999
	3	0.36	24.98	0.9997
	4	0.36	28.05	0.9998
	5	0.36	29.24	0.9999
100/10/10/10	1	0.29	45.08	1.0000
	2	0.36	32.36	0.9999
	3	0.40	24.35	0.9996
	4	0.38	27.88	0.9998
	5	0.37	30.28	0.9999

รูปแบบการทำข้า	เวลาประมาณ (วินาที)	PSNR (dB)	SSIM
ไม่ใช้พิธีกรรมรูปภาพ	0.37	17.26	0.9963
10/1/1/10	0.40	28.54	0.9993
10/3/3/10	0.33	29.83	0.9994
10/10/10/10	0.38	32.56	0.9995
100/1/1/10	0.34	31.50	0.9999
100/3/3/10	0.36	31.99	0.9999
100/10/10/10	0.38	33.39	0.9998

ตารางที่ 3.8: ผลการซ่อมแซมภาพโดยวิธีการเชิงตัวเลขที่นำเสนอด้วยรูปของค่าเฉลี่ยของผลที่ได้จากตารางที่ 3.7

จากตารางจะเห็นว่า การทำข้าในขั้นที่รูปภาพมีขนาดเด็กมากจำนวนนัก ไม่ช่วยให้การประมวลผลได้เร็วขึ้น ผู้วิจัยจึงเลือกใช้การทำข้าแบบ 10/3/3/10 ในการต่อเติมภาพ

3.1.3 การทดสอบประสิทธิภาพในการซ่อมแซมภาพจิตรกรรมไทยโบราณ

ภาพจิตรกรรมที่ใช้ทดสอบ มีทั้งสิ้น 5 ภาพ โดยแต่ละภาพเป็นภาพสีที่มีขนาด 256x256 พิกเซล ซึ่งทั้ง 5 ภาพได้แก่ ภาพที่ 3.1.8a¹ และภาพที่ 3.1.8b² คือ จิตรกรรมฝาผนังวัดแก้วไพฐรย์ ภาพที่ 3.1.8c³ คือ จิตรกรรมฝาผนังวัดพระยืนพุทธบาทใหญ่ ภาพที่ 3.1.8d⁴ คือ จิตรกรรมฝาผนังวัดคงคaram และภาพที่ 3.1.8e⁵ คือ จิตรกรรมฝาผนังวัดท่าถนน โดยจะทำให้ข้อมูลข้างทั้ง 5 ภาพเกิดความเสียหาย โดยใช้ร้อยความเสียหายจากภาพพระเจ้าสร้างอดัม

¹ภาพถ่ายที่วัดแก้วไพฐรย์; ภาพจาก [https://commons.wikimedia.org/wiki/File:จิตรกรรมฝาผนัง_วัดแก้วไพฐรย์_\(7\).jpg](https://commons.wikimedia.org/wiki/File:จิตรกรรมฝาผนัง_วัดแก้วไพฐรย์_(7).jpg) สืบคันเมื่อวันที่ 23 กันยายน 2561

²ภาพถ่ายที่วัดแก้วไพฐรย์; ภาพจาก [https://commons.wikimedia.org/wiki/File:จิตรกรรมฝาผนัง_วัดแก้วไพฐรย์_\(2\).jpg](https://commons.wikimedia.org/wiki/File:จิตรกรรมฝาผนัง_วัดแก้วไพฐรย์_(2).jpg) สืบคันเมื่อวันที่ 23 กันยายน 2561

³ภาพถ่ายที่วัดพระยืนพุทธบาทใหญ่; ภาพจาก https://commons.wikimedia.org/wiki/File:Wat_Phra_Yuen_Phutthabat_Yukhon_01.jpg สืบคันเมื่อวันที่ 23 กันยายน 2561

⁴ภาพถ่ายที่วัดคงคaram; ภาพจาก https://commons.wikimedia.org/wiki/File:จิตรกรรม_อุโบสถวัดคงคaram.JPG สืบคันเมื่อวันที่ 23 กันยายน 2561

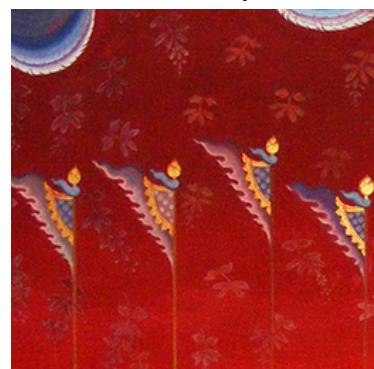
⁵ภาพถ่ายที่วัดท่าถนน; ภาพจาก https://commons.wikimedia.org/wiki/File:Wat_Tha_Thanon_05.JPG สืบคันเมื่อวันที่ 23 กันยายน 2561



(a) วัดแก้วไพพุรย์



(b) วัดแก้วไพพุรย์



(c) วัดพระเย็นพุทธบາท yüktaram



(d) วัดคงカラม



(e) วัดท่าถนน



(f) รอยความเสียหาย

รูปที่ 3.1.8: ภาพต้นฉบับสำหรับใช้ในการทดสอบ



รูปที่ 3.1.9: ภาพที่ทำให้เสียหาย

จากนั้นทำการทดสอบการต่อเติมภาพทั้ง 5 โดยทดสอบวิธีสปริทเบรกแมก และวิธีที่พัฒนาขึ้นโดยใช้วิธีการสปริทเบรกแมกพร้อมทั้งการใช้พีระมิดรูปภาพที่มีการทำซ้ำแต่ละชั้นเป็น 10/3/3/10 ได้ผลลัพธ์ออกมาเป็นดังนี้



รูปที่ 3.1.10: ผลการซ่อมแซมโดยวิธีการสปริทเบรกแม่น

รูปภาพ	เวลาประมาณ (วินาที)	PSNR (dB)	SSIM
1	2.95	33.92	1.0000
2	2.64	37.33	1.0000
3	3.49	37.21	1.0000
4	2.70	29.47	1.0000
5	15.85	32.78	1.0000
เฉลี่ย	2.72	34.89	1.0000

ตารางที่ 3.9: ผลการซ่อมแซมภาพศิลปะไทยจากวิธีการสปีทเบรกเมน



รูปที่ 3.1.11: ผลการซ่อมแซมภาพโดยวิธีการเชิงตัวเลขที่พัฒนาขึ้น

รูปภาพ	เวลาประมาณ (วินาที)	PSNR (dB)	SSIM
1	0.40	34.13	1.0000
2	0.40	38.18	1.0000
3	0.39	37.73	1.0000
4	0.38	29.38	1.0000
5	0.39	37.11	1.0000
เฉลี่ย	0.39	35.30	1.0000

ตารางที่ 3.10: ผลการซ่อมแซมภาพศิลปะไทยโดยวิธีการเชิงตัวเลขที่พัฒนาขึ้น

ทั้งสองวิธี ได้ผลลัพธ์การซ่อมแซมภาพศิลปะไทยในรูปค่าเฉลี่ยอกมาดังนี้

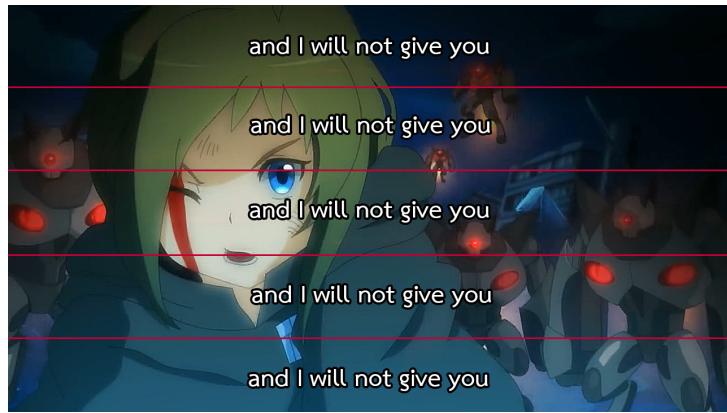
วิธีการ	เวลาประมาณ (วินาที)	PSNR (dB)	SSIM
สปริทเบรกเมน	2.72	34.89	1.0000
วิธีการที่พัฒนาขึ้น	0.39	35.30	1.0000

ตารางที่ 3.11: แสดงผลการซ่อมแซมภาพศิลปะไทยในรูปค่าเฉลี่ยจากตารางที่ 3.9 และตารางที่ 3.10

จากตารางที่ 3.11 จะเห็นได้ว่า วิธีที่พัฒนาขึ้นนั้นสามารถทำงานได้เร็วกว่าวิธีสปริทเบรกเมนเดิม และยังมีคุณภาพที่ดีขึ้นด้วย

3.2 การลบบทบรรยายบนอนิเมะ

สำหรับการลบบทบรรยายอนิเมะ จะใช้วิดีโอ Anime Festival Asia Special Video - feat. Inori Aizawa ซึ่งผลิตโดย Collateral Damage Studios โดยจะตัดวิดีโอด้วยวิดีโอด้วยกันขนาด 1280 x 720 พิกเซล แต่เนื่องจากโดยปกติแล้ว อนิเมะมักมีบรรทัดเพียง 1 ถึง 2 บรรทัด จึงทำการแบ่งวิดีโอดอกอีกเป็น 5 ส่วนได้ขนาดเป็น 1280 x 144 พิกเซลก่อนนำไปทดสอบในลำดับถัดไป และสำหรับบทบรรยายที่จะใช้ทดสอบนั้น เนื่องจากวิดีโอ Anime Festival Asia Special Video - feat. Inori Aizawa ไม่มีคำพูดใดๆ จึงใช้ข้อความ lorem ipsum เป็นบทบรรยาย โดยทำการแสดงบทบรรยาย 1 บรรทัด ความยาว 3 วินาที ทุก 2 วินาที นั่นคือในวิดีโอดังกล่าวจะมีบทบรรยายทั้งสิ้น 20 บรรทัด



รูปที่ 3.2.1: การแบ่งไฟล์วิดีโอเป็น 5 ส่วนสำหรับใช้เป็น 5 ชุดทดสอบ

เนื่องจากไฟล์วิดีโอนั้นประกอบด้วยชุดของภาพหลายภาพ กล่าวคือ $V = \{\mathbf{u}_i | i = 1, 2, 3 \dots N_f\}$ ทำให้ขั้นตอนวิธีการลบคำบรรยายออกจากวิดีโอ จะต้องทำการต่อเติมบริเวณที่เป็นบทบรรยายที่ล่องภาพ ดังที่แสดงในขั้นตอนวิธีต่อไปนี้

Algorithm 8: วิธีการลบบทบรรยายจากวิดีโอ

```

 $V \leftarrow \text{SubtitleRemove}(V)$ 
for  $i = 1, 2, \dots N_f$  do
    • หาโดยเมนต์ต่อเติม  $D$  จากเฟรม  $\mathbf{u}_i$  ซึ่งเป็นภาพที่  $i$  ของวิดีโอ  $V$ 
    • ต่อเติมเฟรม  $\mathbf{u}_i$  โดยใช้โดยเมนต์ต่อเติม  $D$ 
end

```

โดยขั้นตอนการต่อเติมภาพ \mathbf{u}_i ด้วยโดยเมนต์ต่อเติม D นั้นจะสามารถใช้วิธีการเดียวกับการซ่อมแซมภาพศิลปะไทยได้ ส่วนการหาบทบรรยายอนิเมะ จะกล่าวถึงในหัวข้ออยู่อีก

3.2.1 การหาบทบรรยายบนอนิเมะ

ก่อนจะลับบทบรรยายนั้น จำเป็นต้องหาบทบรรยายในภาพให้ได้เสียก่อน โดยบทบรรยายของอนิเมะนั้น มักจะใช้ขอบของตัวอักษรเป็นสีดำ อีกทั้งบทบรรยายนั้นจะลอยห่างออกจากขอบของวิดีโอ และขนาดของคำบรรยายนั้นจะมีขนาดอยู่ประมาณหนึ่งไม่ใหญ่หรือไม่เล็กเกินไป ด้วยสมบัตินี้เองทำให้จึงสามารถหาบริเวณบนเฟรมที่เป็น

บทบรรยายได้โดยจะมีวิธีหาพื้นที่ซึ่งเป็นบทบรรยายดังนี้

Algorithm 9: Finding subtitle

$D \leftarrow findsub(\mathbf{u})$

- ทำการเปลี่ยนสีดำในภาพ \mathbf{u} ให้เป็นสีขาวแล้วเปลี่ยนอีกๆ ให้เป็นสีดำเพื่อหาขอบของคำบรรยาย
 - เปลี่ยนบริเวณสีขาวในภาพให้เป็นสีดำ และเปลี่ยนบริเวณสีดำให้เป็นสีขาว
 - ทำการลบบริเวณสีขาวซึ่งติดกับขอบของภาพออกไป เนื่องจากบทบรรยายจะloyอยู่ไม่ติดกับขอบเสมอ
 - ลบบริเวณที่ใหญ่เกินกว่าจะเป็นบทบรรยาย
 - ลบบริเวณที่เล็กเกินกว่าจะเป็นบทบรรยาย
 - ทำการขยายพื้นที่ๆ เป็นสีขาวขึ้นด้วยความกว้างของขอบบทบรรยาย
 - สีขาวที่เหลืออยู่ในภาพจะเป็นบทบรรยาย
-

วิธีการหาบทบรรยายที่กล่าวไปข้างต้น จะทำการทดสอบกับบทความ *lorem ipsum*⁶ ที่ถูกแปลเป็นภาษาไทย ภาษาอังกฤษ และภาษาญี่ปุ่น โดยมีความสามารถในการหาได้เมนต่อเติมใบบทบรรยายภาษาต่างๆ ดังนี้

⁶Cicero, De finibus bonorum et malorum; เข้าถึงได้ทาง https://en.wikipedia.org/wiki/Lorem_ipsum สืบคันเมื่อวันที่ 23 ตุลาคม 2561

ภาษา	วีดีโอ	จำนวนพิกเซลในโดเมน	จำนวนพิกเซลที่ตรวจพบ	จำนวนพิกเซลที่ผิดพลาด	ร้อยละการผิดพลาด
ไทย	1	23,190,522	24,044,004	2,108,772	9.09
	2	23,232,287	24,026,820	2,204,025	9.49
	3	23,189,082	24,300,589	2,081,340	8.98
	4	23,277,706	23,796,276	2,126,004	9.13
	5	23,221,502	24,247,935	2,185,864	9.41
อังกฤษ	1	27,281,185	28,631,063	3,477,960	12.75
	2	27,269,671	28,513,248	3,514,859	12.89
	3	27,325,148	28,611,300	3,815,082	13.96
	4	27,191,136	28,527,105	3,854,121	14.17
	5	27,326,584	28,709,405	3,909,582	14.31
ญี่ปุ่น	1	28,509,908	30,058,101	3,953,067	13.87
	2	28,534,363	30,023,923	3,565,609	12.50
	3	28,537,968	30,015,047	3,553,128	12.45
	4	28,579,778	30,065,985	3,961,319	13.86
	5	28,558,848	30,354,275	3,671,730	12.86

ตารางที่ 3.12: ความคลาดเคลื่อนของการหาโดเมนต่อitem ในบทบรรยายภาษาต่างๆ

ภาษา	จำนวนพิกเซลในโดเมน	จำนวนพิกเซลที่ตรวจพบ	จำนวนพิกเซลที่ผิดพลาด	ร้อยละการผิดพลาด
ไทย	23,222,220	24,083,125	2,141,201	9.22
อังกฤษ	27,278,745	28,598,424	3,714,321	13.62
ญี่ปุ่น	28,544,173	30,103,466	3,740,971	13.11

ตารางที่ 3.13: ความคลาดเคลื่อนเฉลี่ยของการหาโดเมนต่อitem ในบทบรรยายภาษาต่างๆ

จากการทดลองทั้ง 3 ภาษาพบว่าวิธีการหาคำบรรยายนี้ มีร้อยละการผิดพลาดเฉลี่ยอยู่ที่ 11.98 ซึ่งการทดลองจากนี้ไปจะใช้วิธีการหาคำบรรยายนี้ในการหาโดเมนต่อitemแบบอัตโนมัติ

3.2.2 การลบคำบรรยายจากบทอนิเมะ

สำหรับอนิเมะนี้ แต่ละเฟรมจะเป็นรูปภาพ เราจึงสามารถประยุกต์ใช้วิธีการซ้อมแซมภาพจิตกรรมไทยมาใช้ในการลบคำบรรยายได้ แต่ผู้วิจัยก็ได้สังเกตว่า สำหรับอนิเมะที่เป็นวิดีโอล้วน ในขณะที่ประมวลผลวิดีโอด้วย

สามารถใช้ผลการต่อเติมภาพจากภาพที่แล้ว มาใช้เป็นคำตอบเริ่มต้น จึงได้ว่าขั้นตอนการลบบทบรรยายออกจากวิดีโอ มีดังนี้

Algorithm 10: วิธีการทำงานบนวิดีโอ เมื่อต้องการผลจากภาพที่แล้วมาใช้เป็นคำตอบเริ่มต้น

```

 $V \leftarrow RemoveSubtitle(V)$ 
initialize  $i = 1$ 
while  $i < N_f - 1$  do
     $\mathbf{u}_i$  คือเฟรมที่  $i$  ใน  $V$ 
     $\mathbf{u}_{i+1}$  คือเฟรมที่  $i + 1$  ใน  $V$ 
     $D$  คือโดเมนต่อเติมใน  $\mathbf{u}_{i+1}$ 
     $\mathbf{u}_{i+1} = RemoveByBorrowFrame(\mathbf{u}_i, D, \mathbf{u}_{i+1})$ 
end

```

$RemoveByBorrowFrame(\mathbf{u}_i, D, u_{i+1})$ คือขั้นตอนวิธีที่ 11 ซึ่งในทำนองเดียวกันเราสามารถเปลี่ยน $RemoveByBorrowFrame(u_i, D, \mathbf{u}_{i+1})$ เป็น $RemoveBySkipFrame(\mathbf{u}_i, D, u_{i+1})$ เพื่อใช้กับขั้นตอนวิธี 12 และเปลี่ยนเป็น $RemoveBySkipAndBorrowFrame(\mathbf{u}_i, D, \mathbf{u}_{i+1})$ เพื่อใช้กับขั้นตอนวิธี 13 ได้

ขั้นตอนวิธี การยืมเฟรม จะเป็นการนำผลลัพธ์จากเฟรมก่อนหน้ามาเป็นคำตอบในการเริ่มต้นในการประมวลผลเพื่อให้ผลลัพธ์ลุ่เข้าได้เร็วขึ้น

Algorithm 11: การลับบทบรรยายโดยใช้วิธีการยึมเฟรม

```

 $v \leftarrow RemoveByBorrowFrame(u, D, v)$ 
 $s =$  ค่า SSIM ระหว่าง  $u$  และ  $v$  บริเวณนอกโดเมนต่อเติม
if  $s > 0.9$  then
    คัดลอกบริเวณในโดเมนต่อเติมจาก  $u$  ไปยัง  $v$ 
end
 $v = MultiSplitBregmanColor(v, \lambda, \theta, N_{gs}, N_0, N_1, N_2, \varepsilon, 1, m)$ 

```

ขั้นตอนวิธี การข้ามเฟรม สำหรับเฟรมใดที่ผลลัพธ์ใกล้เคียงกันมาก จะทำการข้ามการต่อเติมภาพในเฟรมนั้นไปโดยใช้คำตอบจากเฟรมก่อนหน้าแทนเพื่อลดเวลาการประมวลผล

Algorithm 12: การลับบทบรรยายโดยใช้วิธีการข้ามเฟรม

```

 $v, \leftarrow RemoveBySkipFrame(u, D, v)$ 
 $s =$  ค่า SSIM ระหว่าง  $u$  และ  $v$  บริเวณนอกโดเมนต่อเติม
if  $s > 0.95$  then
    คัดลอกบริเวณในโดเมนต่อเติมจาก  $u$  ไปยัง  $v$ 
else
     $v = MultiSplitBregmanColor(v, \lambda, \theta, N_{gs}, N_0, N_1, N_2, \varepsilon, 1, m)$ 
end

```

ขั้นตอนวิธี การข้ามและยึมเฟรม คือขั้นตอนวิธี 11 และขั้นตอนวิธี 12 ที่นำมาประยุกต์ใช้งานร่วมกัน

Algorithm 13: การลับบทบรรยายโดยใช้วิธีการข้ามเฟรมและยึมเฟรม

```

 $v \leftarrow RemoveBySkipAndBorrowFrame(u, D, v)$ 
 $s =$  ค่า SSIM ระหว่าง  $u$  และ  $v$  บริเวณนอกโดเมนต่อเติม
if  $s > 0.95$  then
    คัดลอกบริเวณในโดเมนต่อเติมจาก  $u$  ไปยัง  $v$ 
else if  $s > 0.9$  then
    คัดลอกบริเวณในโดเมนต่อเติมจาก  $u$  ไปยัง  $v$ 
     $v = MultiSplitBregmanColor(v, \lambda, \theta, N_{gs}, N_0, N_1, N_2, \varepsilon, 1, m)$ 
else
     $v = MultiSplitBregmanColor(v, \lambda, \theta, N_{gs}, N_0, N_1, N_2, \varepsilon, 1, m)$ 
end

```

ซึ่งผลลัพธ์เปรียบเทียบระหว่างแบบใช้วิธี ยีมเฟรม ใช้วิธีข้ามเฟรม และวิธีข้ามและยีมเฟรม ได้ผลดังตาราง

วิธีการ	วิดีโอ	เวลาประมาณ (วินาที)	PSNR (dB)	SSIM
สปริทเบรกแม่น และพีระมิดรูปภาพ (ขั้นตอนวิธี 6)	1	130.03	32.19	0.9528
	2	135.17	29.98	0.9488
	3	142.11	30.54	0.9485
	4	151.42	30.79	0.9494
	5	147.70	33.48	0.9556
ยีมเฟรม (ขั้นตอนวิธี 11)	1	127.77	33.13	0.9701
	2	137.54	30.21	0.9590
	3	124.71	31.43	0.9620
	4	136.71	31.66	0.9614
	5	137.16	34.56	0.9748
ข้ามเฟรม (ขั้นตอนวิธี 12)	1	104.55	27.10	0.9429
	2	78.07	27.17	0.9351
	3	73.35	29.21	0.9393
	4	116.20	29.91	0.9423
	5	74.28	31.95	0.9442
ข้ามและยีมเฟรม (ขั้นตอนวิธี 13)	1	68.11	27.24	0.9424
	2	73.91	27.22	0.9386
	3	77.34	29.36	0.9437
	4	81.98	30.35	0.9483
	5	77.45	32.46	0.9540

ตารางที่ 3.14: ผลการแบบบรรยายอุกจากอนิเมะด้วยวิธีการเชิงตัวเลขขั้นตอนวิธี 6, 11, 12 และ 13

วิธีการ	เวลาประมาณ (วินาที)	PSNR (dB)	SSIM
สปริทเบรกเมนและพิรษมิตรูปภาพ	141.29	31.39	0.9510
ยีมเฟรม	132.78	32.20	0.9655
ข้ามเฟรม	89.29	29.07	0.9408
ยีมเฟรมและข้ามเฟรม	75.76	29.33	0.9454

ตารางที่ 3.15: ผลการซ่อมแซมภาพโดยวิธีการเชิงตัวเลขที่นำเสนอนี้รูปของค่าเฉลี่ยของผลที่ได้จากตารางที่ 3.14

จากนั้นทำการทดสอบการต่อเติมวิดีโอห้อง 5 โดยวิธีที่คิดค้นขึ้นใช้วิธีการสปริทเบรกเมนพร้อมทั้งการใช้พิรษมิตรูปภาพที่มีการทำซ้ำแต่ละชั้นเป็น 10/3/3/10 พร้อมทั้งใช้การข้ามเฟรมและยีมเฟรม ได้ผลลัพธ์ออกเป็นดังตารางนี้

วิธีการ	เวลาประมาณ (วินาที)	PSNR (dB)	SSIM
สปริทเบรกเมน	*	*	*
วิธีการที่พัฒนาขึ้น	75.76	29.33	0.9454

ตารางที่ 3.16: ผลการลบบทบรรยายออกจากอนิเมะโดยวิธีการสปริทเบรกเมนและวิธีการที่พัฒนาขึ้น

สำหรับวิธีสปริทเบรกเมน เนื่องจากใช้เวลา 1 ชั่วโมงแล้วยังประมาณผลวิดีโอดูทดสอบแรกไม่เสร็จ ทางผู้พัฒนาจึงตัดสินใจยุติการทดลอง เนื่องจากอาจต้องใช้เวลาการประมาณผลเป็นเวลาหลายชั่วโมงสำหรับวิดีโอด้วยความยาว 1 นาที ส่วนวิธีที่คิดค้นขึ้น พบว่าสำหรับวิดีโอด้วยความยาว 1 นาที สามารถทำงานได้เสร็จอย่างรวดเร็วโดยใช้เวลาเพียง 75 วินาที

บทที่ 4

สรุป

បរណ្ណកម្ម

- [1] T.F. Chan and J. Shen , “Mathematical models of local non-texture inpaintings”, SIAM Journal on Applied Mathematics, vol. 62, no. 3, pp. 1019–1043, 2001.
- [2] L. I. Rudin, S. Osher, E. Fatemi, “Nonlinear total variation based noise removal algorithms”, Physica D: Nonlinear Phenomena, vol 60, issues 1–4, pp. 259-268, 1992.
- [3] C.R. Vogel and M.E. Oman,“Iterative methods for total variation denoising”, SIAM Journal on Scientific Computing. vol. 17, pp. 227-238, 1996.
- [4] T. Goldstein and S. Osher,“The Split Bregman Method for L1-Regularized Problems”, SIAM Journal on Imaging Sciences. vol. 2, issue 2, pp. 323-343, 2009.
- [5] E.H. Andelson and C.H. Anderson and J.R. Bergen and P.J. Burt and J.M. Ogden. ”Pyramid methods in image processing”. 1984
- [6] David Salomon. Data Compression: The Complete Reference (4 ed.). Springer. pp. 281. 2007.
- [7] Zhou Wang, Alan Conrad Bovik, Hamid Rahim Sheikh and Eero P. Simoncelli, ”Image quality assessment: from error visibility to structural similarity,” in IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600-612, 2004.