

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 มีความเห็นชอบให้โครงการ
วิจัยเรื่องขั้นตอนวิธีเชิงตัวเลขชนิดใหม่สำหรับการต่อเติมภาพที่ใช้การประยุกต์สำหรับซ่อมแซม
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

สำหรับงานวิจัยนี้ได้นำเสนอวิธีการเชิงตัวเลขที่ถูกพัฒนาขึ้นสำหรับการซ่อมแซมภาพจิตรกรรมไทยและการลบบทบรรยายอนิเมะ โดยวิธีการเชิงตัวเลขนี้ใช้สำหรับการแก้ปัญหาด้วยตัวแบบการแปรผันรวมสำหรับการต่อเติมภาพที่ถูกพัฒนาขึ้นโดยคณะวิจัยของคุณ Chan และคุณ Shen [1] โดยการซ่อมแซมภาพศิลปะไทยจะใช้วิธีการสบริทเบรกแม่นที่ถูกพัฒนาขึ้นโดยขณะวิจัยของคุณ Goldstein และคุณ Osher [4] พร้อมทั้งประยุกต์ใช้ร่วมกับวิธีที่เรียกว่า “มิตรูปภาพที่ถูกคิดค้นด้วยคุณ Andelson และคณะ” [5] และลดปริมาณการทำซ้ำในระดับความคมชัดเดิม พบว่าวิธีการที่พัฒนาขึ้นใช้เวลาประมาณ 7 เท่าและมีคุณภาพของภาพหลังการต่อเติมที่ดีขึ้นเมื่อวัดด้วยปริมาณ Peak Signal Noise Ration (PSNR) [6] นอกจากนี้การลบบทบรรยายอนิเมะจะใช้วิธีการที่ใช้ในการซ่อมแซมภาพศิลปะไทย เมื่อนำมาพัฒนาต่อโดยใช้วิธีการข้ามเฟรมและยืมเฟรม พบว่าสามารถทำงานได้เร็วกว่าการใช้วิธีการสบริทเบรกแม่นกับวิดีโอถึง 67 เท่า

คำสำคัญ: การต่อเติมภาพ

Abstract

This work has presented a new numerical algorithm for TV-based image inpainting with its application for restoring ancient Thai painting images and removing subtitles from animes. This numerical algorithm is used for solving Total Variation model for inpainting that was developed by Chan and Shen [1]. To restoring ancient Thai painting. We will use Splitbregman algorithm which has developed by Goldstien and Osher [4] with image pyramid which have developed by Andelson and His research group [5] and reduced iteration number in original pyramid resolution. This algorithm have better quality when measuring with Peak Signal Noise Ratio (PSNR) and 7 times faster than Splitbregman algorithm. For removing subtitle from anime, We have used the algorithm that we have developed for ancient painting restoration with skip and borrow frame technique. This algorithm has 67 times faster than use Splitbregman algorithm on video.

Keywords: Image inpainting

กิตติกรรมประกาศ

การจัดทำโครงการงานวิจัยนี้สำเร็จลุล่วงไปได้ด้วยดีเนื่องจากคณะกรรมการวิจัยได้รับความกรุณาจากผู้ช่วยศาสตราจารย์ ดร.นพดล ชุมขอบ อาจารย์ที่ปรึกษาโครงการงานวิจัยผู้ให้ความรู้ ให้คำแนะนำและแก้ไขข้อบกพร่องต่างๆ ตลอดช่วงของการทำโครงการวิจัย คณะกรรมการวิจัยจึงขอกราบขอบพระคุณเป็นอย่างสูง ณ โอกาสนี้

ขอขอบพระคุณอาจารย์ ดร.นัยนรัตน์ กันยะมี และอาจารย์ผู้ช่วยศาสตราจารย์ ดร.พรทรัพย์ พรสวัสดิ์ กรรมการสอบโครงการงานวิจัยสำหรับคำแนะนำที่มีประโยชน์ รวมถึงอาจารย์วิทยากรทุกท่านที่สอนใช้งานโปรแกรมเล็กซ์ (La-Tex)

ขอขอบพระคุณอาจารย์ประจำภาควิชาคณิตศาสตร์ทุกท่านที่ได้อ้อมสั่งสอนมอบวิชาความรู้อันมีค่าแก่คณิตศาสตร์ และขอขอบพระคุณเจ้าหน้าที่ประจำภาควิชาคณิตศาสตร์ที่ได้อำนวยความสะดวกด้านต่างๆ ในการจัดทำโครงการงานวิจัยฉบับนี้

ขอขอบพระคุณบิดา มารดา และเพื่อนๆ ผู้ให้กำลังใจ รวมไปถึงการให้ความช่วยเหลือในทุกด้าน ด้านเสมอมา และหวังไว้อย่างยิ่งว่าโครงการงานวิจัยครั้งนี้จะเป็นประโยชน์แก่ผู้ที่สนใจและผู้ที่เกี่ยวข้องต่อไป

สารบัญ

บทคัดย่อ

i

บทคัดย่อ

ii

กิตติกรรมประกาศ

iii

สารบัญรูป

viii

สารบัญตาราง

ix

1 บทนำ

1

1.1	บทนำของการต่อเติมภาพ	1
1.2	การประยุกต์ของการต่อเติมภาพ	2
1.2.1	การซ่อมแซมภาพจิตรกรรมไทยโบราณ	2
1.2.2	การลบบทบรรยายบนอนิเมะ	4
1.3	วัตถุประสงค์ของโครงงานวิจัย	5

2 ความรู้พื้นฐานทางคณิตศาสตร์

6

2.1	ปริภูมิที่มีค่าประจำ	6
2.2	แคลคูลัสของการแปรผันเบื้องต้น	7
2.3	ฟังก์ชันของการแปรผันที่มีข้อบกพร่อง	7
2.4	วิธีการเร็กกิวลารีไลซ์เซชัน	8

iv

2.5	วิธีการไฟน์ติดไฟเรนจ์เบื้องต้น	9
2.6	วิธีการทำซ้ำสำหรับระบบสมการเชิงเส้น	9
2.6.1	วิธีการทำซ้ำจุดตรึง	9
2.6.2	วิธีการเกาส์-ไซเดล	10
3	ขั้นตอนวิธีเชิงตัวเลขชนิดใหม่	11
3.1	การนำเสนอภาพ	11
3.1.1	การนำเสนอภาพเอดเทา	11
3.1.2	การต่อเติมภาพเอดเทา	12
3.1.3	การนำเสนอภาพสี	12
3.2	ตัวแบบเชิงแปรผันสำหรับต่อเติมภาพเอดเทา	13
3.3	ตัวแบบเชิงแปรผันสำหรับการต่อเติมภาพสี	13
3.4	ขั้นตอนวิธีเชิงตัวเลขพื้นฐาน	14
3.4.1	การดิสเครท์เซ็นแบบไฟน์ติดไฟเรนจ์	14
3.4.2	ขั้นตอนวิธีเดินเวลา (explicit time marching method)	15
3.4.3	วิธีการทำซ้ำแบบจุดตรึง (fixed-point iteration method)	17
3.4.4	วิธีการสปริทเบรกแมน (Split Bregman method)	18
3.5	ขั้นตอนวิธีเชิงตัวเลขที่นำเสนอ	21
3.5.1	ขั้นตอนวิธีเชิงตัวเลขสำหรับต่อเติมภาพศิลปะ	21
3.5.2	ขั้นตอนวิธีเชิงตัวเลขสำหรับซ่อมแซมภาพวิดีโอ	24
3.5.3	การทำทบทวนรายบันโนนิเมช	25
3.5.4	การลบคำบรรยายจากบทอนิเมช	26
4	การทดลองเชิงตัวเลข	29
4.1	การซ่อมแซมภาพจิตรกรรมไทยโบราณ	29
4.1.1	การเปรียบเทียบประสิทธิภาพขั้นตอนวิธีเชิงตัวเลขที่มีอยู่แล้ว	30
4.1.2	ขั้นตอนวิธีเชิงตัวเลขสำหรับต่อเติมภาพชนิดใหม่	33
4.1.3	การทดสอบประสิทธิภาพในการซ่อมแซมภาพจิตรกรรมไทยโบราณ	37
4.2	การทำทบทวนรายบันโนนิเมช	41

4.2.1	การหาบทบรรยายบนอนิเมะ	41
4.2.2	การลบคำบรรยายจากบทอนิเมะ	42
5 สรุป		45
บรรณานุกรม		47
ภาคผนวก A		48
A โปรแกรมที่พัฒนาขึ้น		48
A.1	โปรแกรมสำหรับทดสอบ	48
A.1.1	โปรแกรมสำหรับทดสอบการซ่อมแซมภาพศิลปะไทย	48
A.1.2	โปรแกรมทดสอบการลบคำบรรยายอนิเมะ	49
A.2	โปรแกรมตัวอย่างการซ่อมแซมภาพศิลปะไทย	50
A.3	โปรแกรมตัวอย่างการลบคำบรรยายอนิเมะ	51

สารบัญ

1.1.1	ตัวอย่างการซ่อมแซมภาพ	2
1.2.1	ภาพจิตรกรรมไทยที่วัดภูมินทร์ อำเภอเมือง จังหวัดป่าน	3
1.2.2	1 เฟรมของอนิเมะที่มีบทบรรยายแบบแข็ง	4
2.3.1	ตัวอย่างภาพขอบเขตแบบนิวแมน	8
3.1.1	ตัวอย่างภาพเนดเทาที่แสดงระดับความเข้มของภาพในแต่ละระดับ	11
3.1.2	D แทนโดเมนต่อเติม	12
3.4.1	ตัวอย่างการหาอนุพันธ์บนภาพเนดเทา	14
3.5.1	วิธีการพิริมิดรูปภาพ	21
3.5.2	พิริมิดที่ลำดับการทำขึ้นเป็น 10/10/10 และที่ระดับความคมชัดละเอียดสุดใช้จำนวนการทำขึ้นที่ต่างกัน	24
4.1.1	ภาพต้นฉบับ	29
4.1.2	ภาพที่จะทำการซ่อมแซม	30
4.1.3	ผลการซ่อมแซมจากวิธีการเดินเวลา	30
4.1.4	ผลการซ่อมแซมจากวิธีการทำข้อแบบจุดตรึง	31
4.1.5	ผลการซ่อมแซมจากวิธีการสปริทเบรกแมน	32
4.1.6	ภาพต้นฉบับสำหรับใช้ในการทดสอบ	38
4.1.7	ภาพที่ทำให้เสียหาย	38
4.1.8	ผลการซ่อมแซมโดยวิธีการสปริทเบรกแมน	39
4.1.9	ผลการซ่อมแซมภาพโดยวิธีการเชิงตัวเลขที่พัฒนาขึ้น	40

4.2.1 การแบ่งไฟล์วิดีโอเป็น 5 ส่วนสำหรับใช้เป็น 5 ชุดทดสอบ	41
A.1.1 ตัวอย่างโปรแกรมสำหรับทดสอบการซ่อมแซมภาพศิลปะไทยที่พัฒนาขึ้น	49
A.1.2 ตัวอย่างโปรแกรมสำหรับทดสอบการลบคำบรรยายที่พัฒนาขึ้น	49
A.1.3 ตัวอย่างโปรแกรมสำหรับทดสอบการหาคำบรรยายที่พัฒนาขึ้น	50
A.2.1 ตัวอย่างการเลือกรูปภาพสำหรับทำการทดสอบ	50
A.2.2 ตัวอย่างการปรับค่าพารามิเตอร์ที่ใช้ในโครงการนิจัยนี้	51
A.2.3 ตัวอย่างภาพผลลัพธ์จาก Google Colab	51
A.3.1 ไฟล์ตัวอย่างเมื่อทำการแตกไฟล์ออกมาแล้ว test.mp4 เป็นวิดีโอมีคำบรรยาย และ SubtitleRemove.avs เป็นโปรแกรมตัวอย่างสำหรับลบคำบรรยาย	52
A.3.2 test.mp4 (ซ้าย) และ SubtitleRemove.avs (ขวา) เมื่อเปิดด้วย MPC-HC	52
A.3.3 SubtitleRemove.avs สามารถแก้พารามิเตอร์เพื่อใช้กับวิดีโອอนิเมーションได้	52

สารบัญตาราง

4.1	ผลการซ่อมแซมวิธีการเดินเวลา	31
4.2	ผลการซ่อมแซมของวิธีการทำข้า้แบบจุดตึง	32
4.3	ผลการซ่อมแซมของวิธีสปริทเบรกแม่น	33
4.4	แสดงการซ่อมแซมเฉลี่ยของวิธีการเชิงตัวเลข	33
4.5	ผลการซ่อมแซมภาพโดยวิธีการเชิงตัวเลขที่นำเสนอน	34
4.6	ผลการซ่อมแซมภาพโดยวิธีการเชิงตัวเลขที่นำเสนอในรูปของค่าเฉลี่ยของผลที่ได้จากการที่ 4.5	35
4.7	ผลการซ่อมแซมภาพโดยวิธีการเชิงตัวเลขที่นำเสนอยื่นใช้การทำข้า้ในระดับความคมชัดละเอียดสุด 10 ครั้ง	36
4.8	ผลการซ่อมแซมภาพโดยวิธีการเชิงตัวเลขที่นำเสนอในรูปของค่าเฉลี่ยของผลที่ได้จากการที่ 4.7	37
4.9	ผลการซ่อมแซมภาพศิลปะไทยจากวิธีการสปิทเบรกแม่น	39
4.10	ผลการซ่อมแซมภาพศิลปะไทยโดยวิธีการเชิงตัวเลขที่พัฒนาขึ้น	40
4.11	แสดงผลการซ่อมแซมภาพศิลปะไทยในรูปค่าเฉลี่ยจากตารางที่ 4.9 และตารางที่ 4.10	40
4.12	ความคลาดเคลื่อนของการหาโดยเมนต์ต่อเติม ในบทบรรยายภาษาต่างๆ	42
4.13	ความคลาดเคลื่อนเฉลี่ยของการหาโดยเมนต์ต่อเติม ในบทบรรยายภาษาต่างๆ	42
4.14	ผลการลับบทบรรยายออกจากอนิเมะด้วยวิธีการเชิงตัวเลขขั้นตอนวิธี 5, 10, 11 และ 12	43
4.15	ผลการซ่อมแซมภาพโดยวิธีการเชิงตัวเลขที่นำเสนอในรูปของค่าเฉลี่ยของผลที่ได้จากการที่ 4.14	44
4.16	ผลการลับบทบรรยายออกจากอนิเมะโดยวิธีการสปริทเบรกแม่นและวิธีการที่พัฒนาขึ้น	44

บทที่ 1

บทนำ

1.1 บทนำของการต่อเติมภาพ

ในปัจจุบันการใช้ภาพดิจิตัล (digital images) ในสังคมเครือข่ายได้รับความนิยมอย่างแพร่หลาย เนื่องจากโทรศัพท์เคลื่อนที่มีราคาถูกลงแต่มีความสามารถที่ชาญฉลาด สามารถทำหน้าที่ได้ตั้งแต่การเป็นกล้องดิจิตัลคอมแพค (compact digital camera) คุณภาพดีให้ภาพดิจิตัลที่มีความคมชัดสูงจนไปถึงการทำหน้าที่ดังเช่นเครื่องคอมพิวเตอร์ ส่วนบุคคลที่สามารถเชื่อมต่อกับระบบเครือข่ายไร้สายเพื่อรับส่งภาพดิจิตัลในสังคมเครือข่ายด้วยความสะดวกและรวดเร็ว

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

โดยทั่วไปภาพดิจิตัลจะได้รับการประมวลผลภาพก่อนนำไปใช้งานเพื่อให้สามารถใช้ข้อมูลที่ปรากฏบนภาพได้ตรงวัตถุประสงค์ของการใช้งานมากที่สุด ตัวอย่างเช่น ภาพบุคคล (portrait) อาจจำเป็นต้องได้รับการทำจัดสัญญาณ รับกวนออกจากรูปและ/หรือปรับเพิ่มความละเอียดข้อมูลของความเข้มของสีและความสว่างของสีบนบริเวณใบหน้า ก่อนนำไปใช้งานเพื่อจัดทำต้นฉบับวารสารหรือหนังสือของสำนักพิมพ์ เป็นต้น

การต่อเติมภาพ (image inpainting) เป็นวิธีการประมวลผลภาพชนิดหนึ่งที่มีเป้าหมายเพื่อซ่อมแซมภาพด้วยการต่อเติมข้อมูลของความเข้มของสีบนบริเวณที่กำหนด (ต่อไปจะเรียกบริเวณนี้ว่าโดเมนต่อเติม (inpainting domain)) โดยอาศัยข้อมูลของความเข้มของสีที่ปรากฏในภาพ ตัวอย่างเช่น กำหนดให้รูปที่ 1.1.1 (a) แสดงภาพที่ต้องการซ่อมแซมระดับความเข้มของสีบนบริเวณแห่งวัตถุปะรังสีเหลืองสีขาว การต่อเติมภาพดังกล่าวจะเริ่มด้วย

การกำหนดให้บริเวณแห่งวัตถุรูปร่างสีเหลี่ยมสีขาวเป็นโดเมนการต่อเติมดังรูปที่ 1.1.1 (b) จากนั้นภาพที่ได้รับการซ่อมแซมหรือภาพที่ได้รับการต่อเติม (restored or inpainted image) ซึ่งแสดงในรูปที่ 1.1.1 (c) ได้มาจากการ -toner วิธีการต่อเติมภาพ (inpainting algorithm) ซึ่งได้รับการออกแบบเพื่อนำข้อมูลที่ปรากฏบนภาพในบริเวณใกล้เคียงกับขอบของโดเมนต่อเติมมาซ่อมแซมภาพ



(a) ภาพที่ต้องการซ่อมแซม

(b) โดเมนต่อเติม

(c) ภาพที่ได้รับการซ่อมแซม

รูปที่ 1.1.1: ตัวอย่างการซ่อมแซมภาพ

เท่าที่ผู้วิจัยศึกษาและค้นคว้ามาจนถึงขณะนี้ ผู้วิจัยพบว่าการต่อเติมภาพมักนิยมนำไปใช้งานสำหรับการปรับแต่งความสวยงามของภาพบุคคลที่ถ่ายจากโทรศัพท์เคลื่อนที่ เช่น การลบร่องรอยของรอยตื้นๆ การลบร่องรอยของความชรา หรือ การเพิ่มความใสและความเนียนของสีผิวนบนบริเวณใบหน้าผ่านโปรแกรมแอปพลิเคชันแต่งรูปภาพที่มีอยู่ในแอปสโตร์ (App Store) หรือ گوگلเพลย์ (Google Play) เป็นต้น

1.2 การประยุกต์ของการต่อเติมภาพ

1.2.1 การซ่อมแซมภาพจิตรกรรมไทยโบราณ

ภาพจิตรกรรมไทย คือ ภาพเขียนที่มีเอกลักษณ์ความเป็นศิลปะไทยซึ่งโดดเด่นและแตกต่างจากภาพเขียนของชนชาติอื่น ในอดีต ช่างไทยได้สร้างสรรค์ลวดลายและสีสันบนภาพวาดเพื่อสะท้อนประเพณีและวัฒนธรรมในสังคมไทยที่เกี่ยวกับศาสนา ประวัติศาสตร์ โบราณคดี ชีวิตความเป็นอยู่ วัฒนธรรมการแต่งกาย ตลอดจนการแสดง การเล่นพื้นเมืองต่าง ๆ ของแต่ละยุคสมัย

อย่างไรก็ตาม ภาพจิตรกรรมไทยจำนวนไม่น้อยที่เสื่อมสภาพตามกาลเวลา และรอคอยการซ่อมแซมจากช่างในสมัยปัจจุบันที่ต้องไม่สร้างความเสียหายให้กับภาพเขียนเพิ่มขึ้นมากกว่าเดิม ที่ผ่านมาภาพที่ผ่านการซ่อมแซม

มาแล้วจำนวนไม่น้อยได้รับความเสียหายหลังจากการซ่อมแซม ถึงแม้สภาพโดยรวมของภาครัฐกรรมเดิมยังคงอยู่แต่รายละเอียดในตัวภาพเขียนได้เปลี่ยนไป ก่อให้เกิดความเสียหายที่ประเมินค่าไม่ได้

การซ่อมแซมภาครัฐกรรมไทยโบราณโดยใช้การต่อเติมภาพเป็นขั้นตอนของการซ่อมแซมแบบหนึ่งซึ่งไม่ก่อให้เกิดความเสียหายใด ๆ กับภาพเดิม เนื่องจากเป็นการซ่อมแซมโดยการใช้ขั้นตอนวิธีเชิงตัวเลขบนภาพดิจิตัลซึ่งเป็นสำเนาของภาพเดิม ด้วยเหตุผลดังกล่าว ผู้จัดได้เลือกเห็นว่าการซ่อมแซมภาครัฐกรรมไทยโบราณมีความจำเป็นเร่งด่วน เนื่องจากภาพที่ได้รับการซ่อมแซมด้วยการต่อเติมภาพสามารถนำไปใช้ประกอบการตัดสินใจเพื่อวางแผนก่อนการลงมือซ่อมแซมภาพเขียนจริงได้ นอกจากนี้ ขั้นตอนวิธีการต่อเติมภาพสามารถนำไปใช้สร้างแอปพลิเคชันบนโทรศัพท์เคลื่อนที่เพื่อในไปใช้เป็นข้อมูลในการเข้าชมภาพเขียนเดิมที่ยังไม่ได้รับการซ่อมแซมและภาพเขียนที่ได้รับการซ่อมแซมโดยวิธีการทางคณิตศาสตร์จากแอปพลิเคชันที่พัฒนาขึ้น

รูปที่ 1.2.1 แสดงตัวอย่างภาครัฐกรรมไทย¹ ที่ต้องได้รับการซ่อมแซมนบนบริเวณแขนเสื้อของรูปป้าดผู้ชายที่มีส่วนของสีแดงเดิมหลุดหายไป ทั้งนี้ในการซ่อมแซมภาพโดยการต่อเติมภาพ เราจะเริ่มด้วยการสร้างโดเมนต่อเติมบนบริเวณสีพื้นผิวปูนที่แขนเสื้อ จากนั้นจึงนำขั้นตอนวิธีการต่อเติมภาพเพื่อซ่อมแซมภาพบริเวณนั้นให้เป็นสีแดง



รูปที่ 1.2.1: ภาครัฐกรรมไทยที่วัดภูมิทร์ อำเภอเมือง จังหวัดน่าน

¹ภาพถ่ายที่วัดภูมิทร์ อำเภอเมือง จังหวัดน่าน; ภาพจาก <http://topicstock.pantip.com/camera/topicstock/2009/02/07514399/O7514399.html> สืบค้นเมื่อวันที่ 23 กันยายน 2561

1.2.2 การลับบทบรรยายบนอนิเมะ

อนิเมะคือวิดีโอภาพวัดการ์ตูนสไตล์ญี่ปุ่นซึ่งเป็นที่นิยมของเยาวชนไทย ในการรับชมอนิเมะ แม้ว่าเยาวชนไทยสามารถรับชมด้วยบทพากย์เสียงภาษาไทย แต่ก็สูญเสียอรรถรสของการรับชมจากบทบรรยายแบบแข็ง² (hard-sub) ที่เป็นภาษาต่างประเทศในบริเวณด้านล่างของภาพ ในการซ่อนแซม

อนิเมะด้วยการลับบทบรรยายภาษาต่างประเทศจึงเป็นงานที่ยุ่งยากและท้าทายมาก เนื่องจาก

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

ด้วยความท้าทายข้างต้น การพัฒนาขั้นตอนวิธีการต่อเติมภาพที่สามารถกำหนดโดยเมนูต่อเติมเชิงอัตโนมัติให้กับแต่ละเฟรมและประมวลผลได้แม่นยำจนการลับบทบรรยายสามารถทำงานได้แบบเรียลไทม์จึงเป็นสิ่งจำเป็นที่หลีกเลี่ยงไม่ได้

รูปที่ 1.2.2 แสดงตัวอย่าง 1 เฟรมของอนิเมะที่มีบทบรรยายแบบแข็ง³ ที่ต้องซ่อนแซมด้วยการลับบทบรรยายออก ทั้งนี้ในการลับบทบรรยายออกจากเฟรมโดยใช้การต่อเติมภาพ เราจะเริ่มด้วยการสร้างโดยเมนูต่อเติมแบบอัตโนมัติ ในบริเวณบทบรรยาย จากนั้นจึงนำขั้นตอนวิธีการต่อเติมภาพแบบเร็วเพื่อลับบทบรรยายออกจากเฟรม



รูปที่ 1.2.2: 1 เฟรมของอนิเมะที่มีบทบรรยายแบบแข็ง

²บทบรรยายที่ไม่สามารถปิดหรือเปิดได้

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

1.3 วัตถุประสงค์ของโครงการวิจัย

โครงการวิจัยนี้ ผู้วิจัยมีวัตถุประสงค์ของโครงการวิจัยดังต่อไปนี้

- (1) ศึกษาวิธีการแปรผันและขั้นตอนวิธีการใช้ตัวเลขสำหรับปัญหาการต่อเติมภาพแนวสีเทาและภาพสีในระบบ RGB
- (2) พัฒนาขั้นตอนวิธีการต่อเติมภาพชนิดใหม่สำหรับช่องแม่ข่ายภาพจิตรกรรมไทยและระบบทบรรยายจากอนิเมะ
- (3) นำขั้นตอนวิธีที่พัฒนาขึ้นไปใช้ในการซ่อมแซมภาพจิตรกรรมไทยและระบบทบรรยายในอนิเมะ

โครงการวิจัยนี้ ผู้วิจัยมีเป้าหมายสำคัญคือการพัฒนาขั้นตอนวิธีการต่อเติมภาพแบบเร็วและแม่นยำชนิดใหม่เพื่อนำไปใช้สำหรับช่องแม่ข่ายภาพจิตรกรรมไทยและการระบบทบรรยายอุกกาภิเษก

บทที่ 2

ความรู้พื้นฐานทางคณิตศาสตร์

2.1 ปริภูมิที่มีค่าประจำ

ค่าประจำ (Norm) คือฟังก์ชันที่มีค่ามากกว่าเท่ากับศูนย์ โดยจะเรียก p ว่าเป็นฟังก์ชันค่าประจำเมื่อ $p : V \rightarrow [0, +\infty)$ สำหรับ ทุก $a \in F$ และทุก $u, v \in V$ เมื่อ V เป็นปริภูมิเวกเตอร์ และ F เป็นฟิล์ด โดยที่มีสมบัติดังต่อไปนี้

1. $p(u + v) \leq p(u) + p(v)$
2. $p(av) = |a|p(v)$
3. ถ้า $p(v) = 0$ และ $v = 0$

สำหรับโครงงานวิจัยนี้รูปภาพเนดเทาจะเขียนเป็นฟังก์ชันที่อยู่ในปริภูมิยุคลิด (Euclidean space) หรือ \mathbb{R}^2 จึงทำให้ค่าประจำทั้งหมดที่พูดถึงในโครงงานวิจัยนี้จะใช้ Norm L_2 หรือระยะทางแบบยุคลิด (Euclidean distance) ทั้งหมด ซึ่งค่าประจำนี้จะกำหนดโดย

$$\|X\| = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2} \quad \forall x_i \in X$$

2.2 แคลคูลัสของการแปรผันเบื้องต้น

แคลคูลัสของการแปรผัน (Calculus of variations) คือสาขาวิชาในวิชา คณิตศาสตร์วิเคราะห์ เพื่อใช้สำหรับแก้ปัญหาค่าเหมาะสม โดยจะสนใจที่จะหาฟังก์ชันที่เหมาะสมแทนที่จะหาค่าของตัวแปรที่เหมาะสม โดยการแก้ปัญหาเหล่านี้จะอยู่ในรูปแบบของฟังก์ชัน (functional) ซึ่งฟังก์ชันเหล่านี้เองจะอยู่ในรูปแบบของปริพันธ์ หรืออนุพันธ์ ซึ่งความสามารถใช้วิธีการแปรผัน (varational methods) เพื่อแปลงปัญหาดังกล่าวให้อยู่ในรูปแบบของสมการอนุพันธ์ย่อย (partial differential equation : PDE) ได้โดยฟังก์ชันที่ได้จากการแก้สมการเชิงอนุพันธ์ นั้นจะถูกแปรผันไปตามคำต่อของสมการอนุพันธ์เชิงย่อย จึงทำให้เรียกถูกเรียกว่าการแปรผัน

ซึ่งฟังก์ชันหาค่าต่ำสุด มักจะมีรูปทั่วไปดังสมการ 2.2.1 โดยที่ \mathcal{J} เป็นปริภูมิของคำต่อซึ่งประกอบด้วย ฟังก์ชันค่าต่ำสุดของ \mathcal{J} และกำหนด $\mathcal{J} : \mathbb{U} \rightarrow \mathbb{R}$

$$\min_u \int \mathcal{J}(u) \quad (2.2.1)$$

ซึ่งปัญหาการหาค่าต่ำสุดนี้สามารถเขียนในรูปแบบของ สมการอนุพันธ์ย่อยได้ ซึ่งสมการอนุพันธ์ย่อยนี้มีอีกชื่อหนึ่งว่า สมการ ออยเลอร์-ลากราจ โดยใช้วิธีการแปรผัน เพื่อหาค่าต่ำสุดได้ดังนี้

$$\frac{\partial}{\partial u} \mathcal{J}(u) = 0$$

2.3 ฟังก์ชันของการแปรผันที่มีขอบเขต

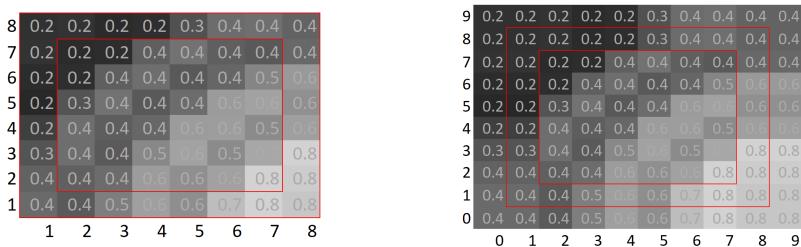
เมื่อใช้วิธีการแปรผันจนได้สมการอนุพันธ์ย่อยแล้ว จะพบว่าสมการอนุพันธ์ย่อยนั้น เป็นปัญหาค่าขอบ (boundary value problem) ซึ่งจะมีเงื่อนไขเพิ่มเติมเป็น ค่าขอบ (boundary value) ตัวอย่างเช่นเมื่อ $\Omega = [0, 1]^2$ และ $u = u(x)$ และสูตรการแปรผัน (varational fomular) เป็น

$$\min_u \int_{\Omega} |\nabla u|^2 dx$$

เมื่อใช้วิธีการแปรผันแล้วจะได้ปัญหาค่าขอบ

$$\begin{cases} -\Delta u = 0, & x \in \Omega \\ \frac{\partial u}{\partial n} = 0, & x \in \partial\Omega \end{cases}$$

ซึ่งปัญหาค่าขอบที่ดังที่ได้ยกตัวอย่างให้เห็นนี้เป็นปัญหาค่าขอบแบบนิวแม่น (neuman) เนื่องจาก $\frac{\partial u}{\partial n} = 0$ ซึ่งค่าขอบแบบนิวแม่นนี้เอง จะเป็นสิ่งที่ใช้ในโครงการวิจัยเรื่องนี้ โดยจะสามารถสร้างค่าขอบนิวแม่นได้โดยการขยายโดยเมโนอก จำนวนนี้ให้ค่าของฟังก์ชันในโดเมนที่ถูกขยายมีค่าเท่ากับค่าที่ใกล้ที่สุดเมื่อวัดระยะทางด้วยระยะทางแบบยุคลิด ดังภาพ ?? โดยภาพซ้ายจะมีโดเมนอยู่ในช่วง $[1, 8]^2$ ซึ่งการคำนวณบางอย่างเช่นการหาเกรดียนตามาเป็นต้องใช้ค่าขอบเพื่อคำนวณ จึงทำการขยายโดยเมโนเป็น $[0, 9]^2$ โดยจะขยายบริเวณสีแดงที่เป็นขอบในภาพออกไป โดยใช้ค่าที่ใกล้ที่สุดที่อยู่ในขอบสีแดง



รูปที่ 2.3.1: ตัวอย่างภาพขอบเขตแบบนิวแม่น

2.4 วิธีการเร็กกิวลารีไลซ์เซชัน

จากโครงการวิจัยนี้จะใช้ตัวแบบ ROF ที่คิดค้นโดยคุณ Chan และคุณ Shen [1] ซึ่งจะมีตัวแบบดังนี้

$$\min_u \{J(u) = \lambda \mathcal{D}(u) + \mathcal{R}(u)\}$$

โดยจะพจน์ \mathcal{D} ว่าพจน์วัดค่าเหมาะสมสมข้อมูล และ \mathcal{R} ว่าพจน์เร็กกิวลารีไลซ์เซชัน ซึ่งพจน์วัดค่าเหมาะสมนี้จะพยายามปรับให้ภาพผลลัพธ์ให้ภาพที่อกรามีค่าใกล้เคียงกับผลลัพธ์ ส่วนพจน์เร็กกิวลารีไลซ์เซชันจะพยายามทำให้ภาพแตกต่างไปจากของเดิม โดยมี λ คือเป็นพารามิเตอร์ที่กำหนดผลกระทบของพจน์เร็กกิวลารีไลซ์เซชัน หาก λ เป็น 0 จะมีแต่พจน์เร็กกิวลารีไลซ์เพียงเท่านั้น ส่งผลให้ภาพที่ได้มีเม็ดโครงของภาพเดิมเหลืออยู่เลย แต่หากกำหนดค่า λ มีค่ามากเกินไปจะทำให้พจน์วัดค่าเหมาะสมส่งผลมาก ซึ่งทำให้ภาพผลลัพธ์ที่ได้มานี้มีอนของเดิมจนแทบไม่เกิดความแตกต่างของการต่อเติมภาพ

2.5 วิธีการไฟไนต์ดิฟเฟอเรนซ์เบื้องต้น

วิธีการไฟไนต์ดิฟเฟอเรนซ์ (finite difference method) เป็นวิธีการหนึ่งที่นิยมใช้ในการประมาณคำตอบของปัญหาค่าขอบ โดยวิธีนี้จะทำการเปลี่ยนอนุพันธ์ในสมการเป็นการประมาณไฟไนต์ดิฟเฟอเรนซ์ (finite difference approximation) และทำการแก้ปัญหาค่าขอบด้วยขั้นตอน 3 ขั้นตอนดังนี้

1. ทำให้คำตอบในโดเมน Ω อยู่ในรูปแบบไม่ต่อเนื่อง โดยแต่ละจุดที่ไม่ต่อเนื่องนี้จะมีชื่อเรียกว่าโนด (node)
2. ประมาณอนุพันธ์หักหมดด้วยไฟไนต์ดิฟเฟอเรนซ์ ซึ่งขั้นตอนนี้จะทำให้สมการเชิงอนุพันธ์ถูกเปลี่ยนเป็นสมการพีชคณิตทำให้สามารถแก้สมการได้ง่ายขึ้น
3. ทำการแก้สมการพีชคณิตด้วยวิธีการต่างๆ

2.6 วิธีการทำซ้ำสำหรับระบบสมการเชิงเส้น

โครงการวิจัยเรื่องนี้จะมีการทำซ้ำด้วยกัน 2 วิธีนั่นคือ วิธีจุดตรึง โดยวิธีจุดตรึงนี้จะใช้ในส่วนวิธีเชิงตัวเลขสำหรับการต่อเติมภาพด้วยวิธีการเดินเวลาแบบชัดแจ้งและการทำซ้ำแบบจุดตรึง และอีกวิธีการคือวิธีการเกาส์-ไซเดล ซึ่งจะใช้ในวิธีเชิงตัวเลขสปริทเบรกเม่น

2.6.1 วิธีการทำซ้ำจุดตรึง

วิธีการทำซ้ำจุดตรึง (fixed-point iteration) นิยมใช้แก้ปัญหาที่อยู่ในรูป $f(x) = 0$ โดยจะทำการจัดรูปด้วยวิธีการทางพีชคณิตให้ได้สมการอยู่ในรูป $x = g(x)$ จากนั้นจึงทำซ้ำ

$$x_{i+1} = g(x_i), \quad i = 0, 1, 2, \dots$$

เมื่อ i มากเกินกว่าจำนวนรอบการทำซ้ำที่กำหนด หรือ $|x_{i+1} - x_i|$ น้อยกว่าค่าที่กำหนดจะหยุดการทำซ้ำ แล้วจะได้ว่า x_i เป็นคำตอบของปัญหา

2.6.2 วิธีการเกาส์-ไซเดล

สำหรับวิธีการเกาส์-ไซเดล นิยมใช้ในการแก้ระบบสมการที่อยู่ในรูป $Ax = b$ เมื่อ A คือเมตริกซ์ขนาด $m \times n$, b และ x เป็นเวกเตอร์ขนาด $m \times 1$

$$x_i^{(n+1)} = \frac{1}{a_{ii}} \left\{ b_i - \sum_{j=1}^{i-1} a_{ij} x_{ij}^{(n)} - \sum_{j=i+1}^m a_{ij} x_j^{(n)} \right\}, \quad n = 1 \dots$$

เมื่อ n มากเกินกว่าจำนวนรอบการทำซ้ำที่กำหนด หรือ $|x_{i+1} - x_i|$ น้อยกว่าค่าที่กำหนดจะหยุดการทำซ้ำ แล้ว จะได้ว่า x_i เป็นคำตอบของปัญหา

บทที่ 3

ขั้นตอนวิธีเชิงตัวเลขชนิดใหม่

3.1 การนำเสนอภาพ

3.1.1 การนำเสนอภาพเดดเทา

สำหรับภาพถ่ายสามารถพิจารณาภาพเป็นฟังก์ชันได้ดังนี้

$$u : \Omega \subset \mathbb{R}^2 \rightarrow V \subset [0, \infty)$$

เป็นฟังก์ชันต่อเนื่อง โดยที่ $\mathbf{x} = (x, y) \in \Omega$ แทนพิกัดทางกายภาพ (physical position) ของภาพ $u(\mathbf{x}) \in V$ แทนระดับความเข้มของภาพ (image intensity) ที่ \mathbf{x} และ Ω แทนโดเมนของภาพ ซึ่งในที่นี้สามารถสมมติได้โดยไม่เสียหลักการสำคัญว่า $\Omega = [1, n]^2$ และ $V = [0, 1]$ เมื่อ $n > 1$ เป็นจำนวนเต็มบวก และโดเมนของภาพเป็นรูปสี่เหลี่ยม ทั้งนี้จะเรียกภาพ n ที่นิยามข้างต้นว่าภาพเดดเทา (grayscale image)

8	0.2	0.2	0.2	0.2	0.3	0.4	0.4	0.4
7	0.2	0.2	0.2	0.4	0.4	0.4	0.4	0.4
6	0.2	0.2	0.4	0.4	0.4	0.4	0.5	0.6
5	0.2	0.3	0.4	0.4	0.4	0.6	0.6	0.6
4	0.2	0.4	0.4	0.4	0.6	0.6	0.5	0.6
3	0.3	0.4	0.4	0.5	0.6	0.5	0.8	
2	0.4	0.4	0.4	0.6	0.6	0.6	0.8	0.8
1	0.4	0.4	0.5	0.6	0.6	0.7	0.8	0.8

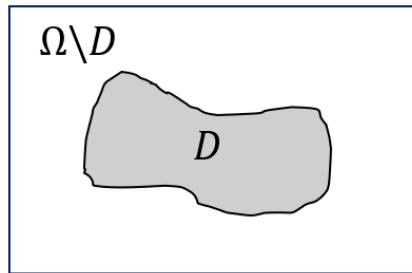
รูปที่ 3.1.1: ตัวอย่างภาพเดดเทาที่แสดงระดับความเข้มของภาพในแต่ละระดับ

จากภาพ 3.1.1 สังเกตว่าที่ค่าความเข้มของภาพเข้าใกล้ 0 จะให้สีเป็นลักษณะสีดำ ดังเช่นบริเวณที่พิกัดทางกายภาพเป็น (4,8) และเมื่อค่าความเข้มของสีเข้าใกล้ 1 จะให้สีที่มีลักษณะเป็นสีขาว ดังเช่นบริเวณที่มีพิกัดทางกายภาพเป็น (7,1)

3.1.2 การต่อเติมภาพเขตเทา

ในโครงงานวิจัยชิ้นนี้สำหรับการต่อเติมภาพ จะทำการหาคำตออบของภาพที่อยู่ในโดเมนต่อเติมภาพ ซึ่งจะของกำหนดตัวแปรต่างๆ ดังนี้

ให้ $\Omega \subset \mathbb{R}^2$ แทนโดเมนภาพ (image domain) $D \subset \mathbb{R}^2$ แทนโดเมนต่อเติม (ดูรูปที่ 3.1.2) และ $V \subset [0, \infty)$ และให้ $u : \Omega \rightarrow V$, $z : \Omega \rightarrow V$ แทนภาพที่ได้รับการซ่อมแซมและภาพที่ต้องการซ่อมแซมตามลำดับ



รูปที่ 3.1.2: D แทนโดเมนต่อเติม

การต่อเติมภาพเขตเทาจะเป็นการหาคำตออบของพื้นที่ที่ได้รับความเสียหายที่อยู่บนภาพ z ซึ่งเป็นบริเวณในโดเมนต่อเติม D โดยใช้ข้อมูลที่มีอยู่ใน $\Omega \setminus D$ เพื่อหาข้อมูลใน D ที่ได้รับความเสียหายเป็นคำตออบในภาพ u

3.1.3 การนำเสนօภาพสี

ต่อไปจะพิจารณาภาพสีในระบบ RGB นั่นคือ จะพิจารณาว่าภาพ u ประกอบด้วยสีด้วยกันทั้งสิ้น 3 คือแดง, เขียว และ น้ำเงิน จึงเขียนภาพ u ในรูปแบบของเวคเตอร์ได้ดังนี้

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

เมื่อ $u_1, u_2, u_3 : \Omega \rightarrow V$ แทนภาพในเนดสีแดง สีเขียว และสีน้ำเงินของ u ซึ่งการต่อเติมภาพสีที่พูดถึงในโครงงานวิจัยนี้จะทำการแยกแต่ละเนดสีออกเป็นเนดเทา 3 ระนาบ แล้วจึงใช้การต่อเติมภาพเขตเทากับทั้ง 3 เนดสีก่อนรวมกลับเป็นภาพสีอีกครั้ง

3.2 ตัวแบบเชิงแปรผันสำหรับต่อเติมภาพเฉพาะ

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

$$\min_u \{ \mathcal{J}(u) = \frac{1}{2} \int_{\Omega} \lambda(u - z)^2 d\Omega + \int_{\Omega} |\nabla u| d\Omega \} \quad (3.2.1)$$

เมื่อ

$$\lambda = \lambda(x) = \begin{cases} \lambda_0, & x \in \Omega \setminus D \\ 0, & x \in D \end{cases} \quad (3.2.2)$$

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

โดยแคลคูลัสของการแปรผัน (Calculus of variations) จะได้สมการอย่างละเอียดลักษณะจึงที่เกี่ยวข้องกับ (3.2.1) เป็น

$$\begin{cases} -\nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) + \lambda(u - z) = 0, & x \in (1, n)^2 \\ \frac{\partial u}{\partial n} = 0, & x \in \partial\Omega \end{cases} \quad (3.2.3)$$

เมื่อ n แทนเวกเตอร์หน่วยที่ตั้งฉากกับของของภาพ

3.3 ตัวแบบเชิงแปรผันสำหรับการต่อเติมภาพสี

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 $z_1, z_2, z_3 : \Omega \rightarrow V$ แทนภาพในเฉดสีแดง สีเขียว และสีน้ำเงินของ \mathbf{u}, \mathbf{z} ตามลำดับ

ในทำงองเดียวกันกับตัวแบบการต่อเติมภาพเฉพาะที่ใช้การแปรผันรวม ตัวแบบการต่อเติมภาพสีที่ใช้การแปรผันรวมสามารถเขียนได้ดังนี้

$$\min_{\mathbf{u}} \{ \bar{\mathcal{J}}(\mathbf{u}) = \bar{\mathcal{D}}(\mathbf{u}, \mathbf{z}) + \bar{\mathcal{R}}(\mathbf{u}) \} \quad (3.3.1)$$

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

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

3.4 ขั้นตอนวิธีเชิงตัวเลขพื้นฐาน

3.4.1 การดิสcretize ชันแบบไฟน์เดิร์ฟเฟอร์เรนซ์

ไฟน์เดิร์ฟเฟอร์เรนซ์ (Finite Difference) คือวิธีการสำหรับการประมาณค่าอนุพันธ์เมื่อใช้วิธีเชิงตัวเลข ซึ่งในโครงงานวิจัยนี้จะมีตัวดำเนินการที่เกี่ยวข้องกับอนุพันธ์ด้วยกัน 3 ตัวได้แก่ แกรเดียนต์ ไดเวอร์เจน และ ลาป้า เชียนซึ่งสามารถทำการหาได้ดังนี้

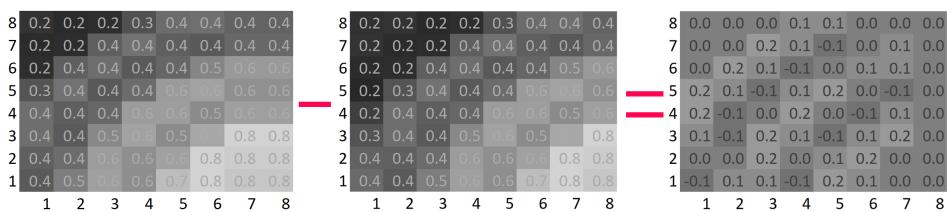
การหาอนุพันธ์

ทั้ง แกรเดียนต์ ไดเวอร์เจน และ ลาป้า เชียน ล้วนมีพื้นฐานมาจาก การหาค่าอนุพันธ์ในโครงงานวิจัยนี้จะใช้วิธีการฟอร์เวิร์ดดิฟเฟอร์เรนซ์ (Forward Difference) และใช้เงื่อนไขค่าขอบแบบนิวmann (neumann boundary condition)

นั่นคือการหาอนุพันธ์ของค่าความเข้มที่พิกัดทางกายภาพเป็น (i, j)

$$\frac{d}{dx} u_{i,j} = \frac{u_{i,j+1} - u_{i,j}}{h}$$

เมื่อระบบกริดที่ใช้มีความท่างเพียงหนึ่งหน่วย จึงได้ว่า $h = 1$ ทั้งนี้ระยะห่าง h อาจเปลี่ยนไปตามชั้นของพีระมิด รูปภาพ



รูปที่ 3.4.1: ตัวอย่างการหาอนุพันธ์บนภาพเนตเทา

จากภาพ 3.1.1 เมื่อต้องการหาอนุพันธ์เทียบแกน x จะทำการสរ้งภาพซึ่งทำการตัดขอบทางซ้ายออกหนึ่งคอลัมน์และเพิ่มขอบทางขวาหนึ่งคอลัมน์โดยใช้เงื่อนไขค่าขอบแบบนิวน์เมน จากนั้นภาพที่สร้างขึ้นไปลบกับภาพเดิมจะได้ออนุพันธ์ของภาพนั้นดังที่ปรากฏทางขวา ทั้งนี้หาก $h \neq 1$ สามารถทำการหารภาพผลลัพธ์ด้วยค่า h ได้เพื่อให้ได้ค่าที่ต้องการ

การหาเกรดรเดียน

สำหรับการหาเกรดรเดียน (Gradient) จะใช้การหาอนุพันธ์โดยวิธีฟอร์เวิร์ดดิฟเฟอร์เรนจ์ดังที่กล่าวไปในหัวข้อก่อนหน้า ซึ่งในแนวแกน x และแนวแกน y คำตอบที่ได้จะเป็นเวคเตอร์ของอนุพันธ์แนวแกน x และอนุพันธ์แนวแกน y ได้เวคเตอร์ดังนี้

$$\nabla \vec{v}_{u_i} = \left(\frac{d}{dx} u_{i,j}, \frac{d}{dy} u_{i,j} \right)^\top$$

การหาไดเวอร์เจน

สำหรับไดเวอร์เจน (Divergence) จะเป็นการหาผลรวมของอนุพันธ์ในแต่ละแกนของเวคเตอร์ด้วยวิธีฟอร์เวิร์ดดิฟเฟอร์เรนจ์ นั่นคือ

$$\nabla \cdot (\vec{v}_{i,j}) = \frac{\partial}{\partial x} v_{i,j,x} + \frac{\partial}{\partial y} v_{i,j,y}$$

การหาลาปาเชียน

สำหรับลาปาเชียน (Lapacian) นั่นคือการทำหาไดเวอร์เจรบแนวเวคเตอร์ที่หาเกรดรเดียนแล้ว แต่ทั้งนี้สามารถหาลาปาเชียนได้จาก

$$\Delta u_{i,j} = u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - 4u_{i,j}$$

3.4.2 ขั้นตอนวิธีเดินเวลา (explicit time marching method)

คณะวิจัย [2] ได้แนะนำวิธีการเชิงตัวเลขสำหรับการคำนวณรบกวนโดยใช้วิธีการเดินเวลาแบบชั้ดเจ้ง ซึ่งสามารถประยุกต์เป็นวิธีเชิงตัวเลขสำหรับการต่อเติมภาพได้ดังนี้

เริ่มจากการแนะนำตัวแปรเวลาสังเคราะห์ (time artificial variable) จากนั้นหาคำตอบแบบสภาพะคงตัว (steady-state solution) ในขณะที่ $t \rightarrow \infty$ ของสมการเชิงอนุพันธ์อย่างไม่เป็นเชิงเส้นที่ขึ้นอยู่กับเวลา

$$u(\mathbf{x}, t_{k+1}) = u(\mathbf{x}, t_k) + \tau \left(\nabla \cdot \left(\frac{\nabla u(\mathbf{x}, t_k)}{|\nabla u(\mathbf{x}, t_k)|} \right) + \lambda(\mathbf{x})(u(\mathbf{x}, t_k) - z(\mathbf{x})) \right), \quad u(\mathbf{x}, t_0) = z \quad (3.4.1)$$

เมื่อ $t_k = t_0 + k\tau$ ($\tau > 0$) แทนขั้นเวลาที่ k และ $t_0 = 0$ แทนขั้นเวลาเริ่มต้น

วิธีเดินเวลาแบบขั้ดแจ้งสำหรับภาพเนดเทามีขั้นตอนวิธีดังนี้

Algorithm 1: วิธีการเดินเวลาแบบขัดแจ้งสำหรับการต่อเติมภาพที่ใช้การแปรผันรวม

Input:

- u คือรูปภาพที่ต้องการต่อเติม
- λ คือพารามิเตอร์เริกกวาร์ฟรีเซนชัน ที่ได้กล่าวถึงในสมการ (3.2.2)
- 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

```

3.4.3 วิธีการทำซ้ำแบบจุดตรึง (fixed-point iteration method)

คณะวิจัย [3] ได้แนะนำวิธีการเชิงตัวเลขสำหรับการกำจัดสัญญาณรบกวนโดยใช้วิธีการทำซ้ำแบบจุดตรึง ซึ่งสามารถประยุกต์เป็นวิธีเชิงตัวเลขสำหรับการต่อเติมภาพได้ดังนี้

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

$$-\nabla \cdot \left(\frac{\nabla u^{[\nu+1]}}{|\nabla u|^{[v]}} \right) + \lambda(u^{[\nu+1]} - z) = 0, \quad u^{[0]} = z \quad (3.4.2)$$

เนื่องจาก $\frac{1}{|\nabla u|} = \frac{1}{\sqrt{u_x^2 + u_y^2}} \rightarrow \infty$ ในบริเวณที่ u มีความเข้มสีเป็นเอกพันธุ์ ($u(\mathbf{x}) = \text{ค่าคงตัว}$) เพื่อหลีกเลี่ยงปัญหาเชิงตัวเลขจะเกิดขึ้นใน (3.4.1) และ (3.4.2) 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 (3.2.2)

β เป็นจำนวนจริงบวกที่ใช้เพื่อหลีกเลี่ยงการหารด้วยศูนย์

N เป็นจำนวนเต็มบวกสำหรับกำหนดจำนวนรอบที่ทำงาน

ε เป็นจำนวนจริงบวกของค่าความคลาดเคลื่อนสัมพัทธ์

Output: รูปภาพที่ผ่านการต่อเติมแล้ว

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

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

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

$u^{old} = u$

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

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

$i = i + 1$

end

Algorithm 3: การทำข้ามเกาส์-ไซเดล สำหรับวิธีการจุดตรึง

Input:

- n คือรูปภาพที่ต้องการต่อเติม
- λ คือพารามิเตอร์เรกูลาร์ไรเซชัน ที่ได้กล่าวถึงในสมการ (3.2.2)
- β เป็นจำนวนจริงบวกที่ใช้เพื่อหลีกเลี่ยงการหารด้วยศูนย์
- N เป็นจำนวนเต็มบวกสำหรับกำหนดจำนวนรอบที่ทำงาน

Output: รูปภาพที่ผ่านการทำเกาส์-ไซเดลแล้ว

$$u \leftarrow GaussSeidel(u, \lambda, \beta, N_{gs})$$

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 (3.4.1) และ (3.4.2) เรายังคงยิ่ง β มีค่าน้อยลงมากขึ้นเท่าไหร่ ความแม่นยำของตัวแบบ (3.2.1) ยิ่งมีมากขึ้นเท่านั้น นอกจากนี้ รายังพบอีกว่า การแก้สมการ (3.4.1) และ (3.4.2) ยิ่งมีความยุ่งยากมากขึ้นสำหรับ β ที่มีค่าน้อยๆ

เพื่ออาจนความยากเชิงตัวเลขนี้ คณะวิจัยโดย [4] ได้แนะนำวิธีการสปริทเบรกแมนซึ่งสามารถถูกล่าவถึง พอกสังเขป ดังนี้

3.4.4 วิธีการสปริทเบรกแมน (Split Bregman method)

เริ่มจากการแนะนำเกตอร์เสริม \mathbf{w} พารามิเตอร์เบรกแมน (Bregman parameter) \mathbf{b} และพารามิเตอร์ เพนลัท (penalty parameter) $\theta > 0$ และเขียน (3.2.1) ใหม่ ดังนี้

$$\min_{u,w} \{ \mathcal{J}(u, \mathbf{w}) = \frac{1}{2} \int_{\Omega} \lambda (u - z)^2 d\Omega + \int_{\Omega} |\mathbf{w}| d\Omega + \frac{\theta}{2} \int_{\Omega} (\mathbf{w} - \nabla u + \mathbf{b}) d\Omega \} \quad (3.4.3)$$

สำหรับการหาค่าตอบของ (3.4.3) เราจะใช้วิธีการหาค่าต่อๆกันแบบสลับ (alternating minimization method) โดยเริ่มจากการตั้ง \mathbf{w}^{old} และ \mathbf{b}^{old} จากนั้นแก้ปัญหาอย่างสำหรับ u

$$u^{\text{New}} = \arg \min_u \{ \mathcal{J}_1(u) = \frac{1}{2} \int_{\Omega} \lambda(u - z)^2 d\Omega + \frac{\theta}{2} \int_{\Omega} (\mathbf{w}^{\text{old}} - \nabla u + \mathbf{b}^{\text{old}}) d\Omega \} \quad (3.4.4)$$

ต่อไปใช้ u^{New} ที่ได้จากการแก้ปัญหาอย่าง (3.4.4) เพื่อแก้ปัญหาอย่างสำหรับ \mathbf{w}

$$\mathbf{w}^{\text{New}} = \arg \min_{\mathbf{w}} \{ \mathcal{J}_2(\mathbf{w}) = \int_{\Omega} |\mathbf{w}| d\Omega + \frac{\theta}{2} \int_{\Omega} (\mathbf{w} - \nabla u^{\text{New}} + \mathbf{b}^{\text{old}}) d\Omega \} \quad (3.4.5)$$

สุดท้ายจึงปรับปรุงพารามิเตอร์เบรกเมนโดย

$$\mathbf{b}^{\text{New}} = \mathbf{b}^{\text{old}} + \nabla u^{\text{New}} - \mathbf{w}^{\text{New}} \quad (3.4.6)$$

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

Algorithm 4: วิธีสปริทเบรกเม่นสำหรับการต่อเติมภาพที่ใช้การแปรผันรวม**Input:**

- u คือรูปภาพที่ต้องการต่อเติม
- λ คือพารามิเตอร์เกกิวาร์เร็กชัน ที่ได้ถูกล่างึงในสมการ (3.2.2)
- θ คือพารามิเตอร์เพนล์ที่ ซึ่งเป็นจำนวนจริงบวก
- N_{gs} เป็นจำนวนเต็มบวกสำหรับกำหนดจำนวนรอบที่ทำงานของการทำกาส์-ไซเดล
- N เป็นจำนวนเต็มบวกสำหรับกำหนดจำนวนรอบที่ทำงานของสปริทเบรกเม่น
- ε เป็นจำนวนจริงบวกของค่าความคลาดเคลื่อนสัมพัทธ์

Output: รูปภาพที่ผ่านการต่อเติมแล้ว

```

 $u \leftarrow SplitBregman(u, \lambda, \theta, N_{gs}, N, \varepsilon)$ 
initialize  $i = 0, \mathbf{b} = \vec{0}, \mathbf{w} = \vec{0}, z = u$ 
while  $i < N$  and  $err > \varepsilon$  do
     $u^{old} = u; w^{old} = w; b^{old} = b;$ 
     $u = \arg \min_u \{ \mathcal{J}_1(u) = \frac{1}{2} \int_{\Omega} \lambda (u - z)^2 d\Omega + \frac{\theta}{2} \int_{\Omega} (\mathbf{w}^{old} - \nabla u + \mathbf{b}^{old}) d\Omega \}$ 
     $w = \arg \min_{\mathbf{w}} \{ \mathcal{J}_2(\mathbf{w}) = \int_{\Omega} |\mathbf{w}| d\Omega + \frac{\theta}{2} \int_{\Omega} (\mathbf{w} - \nabla u^{New} + \mathbf{b}^{old}) d\Omega \}$ 
     $b = b^{old} + \nabla u - \mathbf{w}$ 
     $err = \frac{\|u - u^{old}\|}{\|u\|}$ 
     $i = i + 1$ 
end

```

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

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

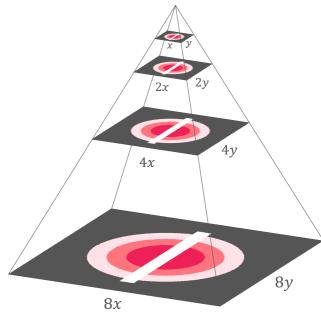
โดยใช้วิธีการไฟน์ติดไฟเรนจ์และวิธีการกาส์-ไซเดลจำนวน N_{gs} รอบ(2) ผลเฉลยของ $\mathbf{w} = \arg \min_{\mathbf{w}} \mathcal{J}_2(\mathbf{w})$ กำหนดโดย

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

3.5 ขั้นตอนวิธีเชิงตัวเลขที่นำเสนอ

3.5.1 ขั้นตอนวิธีเชิงตัวเลขสำหรับต่อเติมภาพศิลปะ

สำหรับวิธีการซ่อมแซมภาพศิลปะไทย จะใช้วิธีการสปริทเบรกแมนเพื่อหลีกเลี่ยงปัญหาเชิงตัวเลขที่จะเกิดขึ้น แต่เพื่อให้วิธีการสปริทเบรกแมนประมวลผลภาพได้รวดเร็วขึ้น ผู้วิจัยได้พัฒนากระบวนการกำหนดค่าตอบรับ ด้วยวิธีการ มัลติรีโซลูชัน (multi-resolution method) หรือวิธีการพีระมิดรูปภาพ (pyramid method) [5] เริ่มจากการย่อขนาดรูปลงครึ่งหนึ่งโดยใช้วิธี Bilinear Interpolation จนกระทั่งถึงระดับความคมชัดที่ต้องการ จากนั้นทำการต่อเติมภาพขนาดเล็ก และนำผลลัพธ์ที่ได้จากการย่อขนาดเล็กทำการขยายภาพขึ้นสองเท่าโดยใช้ Bi-linear Interpolation เป็นค่าตอบรับเริ่มต้นสำหรับการต่อเติมภาพในชั้นถัดไป



รูปที่ 3.5.1: วิธีการพีระมิดรูปภาพ

ขั้นตอนวิธีสำหรับการทำพีระมิดรูปภาพสำหรับการต่อเติมภาพแบบสปริทเบรกแมนเพื่อให้ประมวลผลได้เร็วขึ้นนั้นสามารถสรุปได้ดังนี้

Algorithm 5: วิธีสปริทเบรกเมนท์ใช้พีระมิดรูปภาพ

Input: u คือรูปภาพที่ต้องการต่อเติม λ คือพารามิเตอร์เริกกิวลาร์เรซชัน ที่ได้กล่าวถึงในสมการ (3.2.2) θ คือพารามิเตอร์เพนล์ที่ ซึ่งเป็นจำนวนจริงบวก N_{gs} เป็นจำนวนเต็มบวกสำหรับกำหนดจำนวนรอบที่ทำงานของการทำเกาส์-ไซเดล c ตัวแปรช่วยสำหรับบอกความลึก ให้กำหนดเป็น 1 m คือ ระดับความลึกของพีระมิดรูปภาพ เป็นจำนวนเต็มบวก N_0 จำนวนรอบการทำสปริทเบรกเมนท์ขั้นละเอียดสุด N_1 จำนวนรอบการทำสปริทเบรกเมนท์ขั้นต่างๆ N_2 จำนวนรอบการทำสปริทเบรกเมนท์ขั้นหยาบสุด**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** $x = Bilinear(u, \lfloor width * 0.5 \rfloor, \lfloor height * 0.5 \rfloor)$ $y = Bilinear(\lambda, \lfloor width * 0.5 \rfloor, \lfloor height * 0.5 \rfloor)$ $r = MRSBC(x, z, y, \lambda, \theta,$ $N_{gs}, N_0, N_1, N_2, \varepsilon, c + 1, m)$ $u = Bilinear(r, width, height)$ **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 6: Bilinear Interpolation

$J \leftarrow \text{Bilinear}(I, x, y)$

Initialize $v = \text{ความสูงของภาพ } I, w \text{ คือความกว้างของภาพ } I,$

$$S_R = \frac{c}{a}, S_C = \frac{d}{b}, r = 1, 2, \dots, v, c = 1, 2, \dots, w,$$

$$r' = 1, 2, \dots, x, c' = 1, 2, \dots, y,$$

$$r_f = \lfloor r' \cdot S_R \rfloor$$

$$c_f = \lfloor c' \cdot S_C \rfloor$$

$$\Delta r = r_f - r$$

$$\Delta c = c_f - c$$

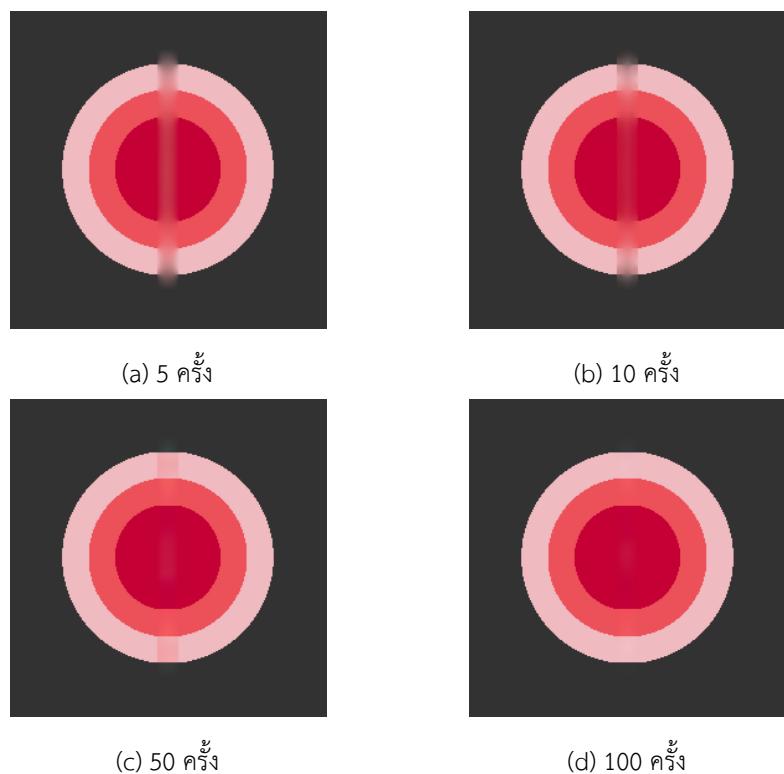
$$J(r', c') = I(r, c) \cdot (1 - \Delta r) \cdot (1 - \Delta c)$$

$$+ I(r + 1, c) \cdot \Delta r \cdot (1 - \Delta c)$$

$$+ I(r, c + 1) \cdot (1 - \Delta r) \cdot \Delta c$$

$$+ I(r + 1, c + 1) \cdot \Delta r \cdot \Delta c$$

นอกจากนี้แล้ว ผู้วิจัยยังได้สังเกตอีกว่า การทำซ้ำนั้นจะลู่เข้าเร็วในช่วงแรก จากนั้นความเร็วในการลู่เข้าจะลดลง ซึ่งทำให้การทำซ้ำเพียงไม่กี่ครั้งในระดับความคมชัดเดิม มีผลการซ่อมแซมภาพจนแสดงความคล้ายคลึงกับภาพต้นฉบับได้



รูปที่ 3.5.2: พีระมิดที่ลำดับการทำข้าเป็น 10/10/10 และที่ระดับความคมชัดละเอียดสุดใช้จำนวนการทำข้าที่ต่างกัน

ผู้วิจัยจึงกำหนดให้การทำข้าในระดับความละเอียดสูงสุดเท่ากับ 10 ครั้ง เพราะการทำข้าในระดับความละเอียดสูงสุดทำให้ทำงานได้ดีข้า อีกทั้งการทำข้าในขันที่รูปภาพมีขนาดเล็กจำนวนมาก ไม่ช่วยให้การประมวลผลได้เร็วขึ้น ผู้วิจัยจึงเลือกใช้การทำข้าแบบ 10/3/3/10 ในการต่อเติมภาพ

3.5.2 ขั้นตอนวิธีเชิงตัวเลขสำหรับซ่อมแซมภาพวิดีโอ

เนื่องจากไฟล์วิดีโอนั้นประกอบด้วยชุดของภาพหลายภาพ กล่าวคือ $V = \{u_i | i = 1, 2, 3 \dots N_f\}$ ทำให้ขั้นตอนวิธีการลับคำบรรยายออกจากวิดีโอ จะต้องทำการต่อเติมบริเวณที่เป็นบทบรรยายที่ลับภาพ ดังที่แสดงในขั้นตอนวิธีต่อไปนี้

Algorithm 7: วิธีการลบบทบรรยายจากวิดีโอ

```

 $V \leftarrow \text{SubtitleRemove}(V)$ 
for  $i = 1, 2, \dots N_f$  do
    • หาโดเมนต่อเติม  $D$  จากเฟรม  $u_i$  ซึ่งเป็นภาพที่  $i$  ของวิดีโอ  $V$ 
    • ต่อเติมเฟรม  $u_i$  โดยใช้โดเมนต่อเติม  $D$ 
end

```

โดยขั้นตอนการต่อเติมภาพ u_i ด้วยโดเมนต่อเติม D นั้นจะสามารถใช้วิธีการเดียวกับการซ่อมแซมภาพศิลปะไทยได้ ส่วนการทำโดเมนต่อเติมซึ่งเป็นบทบรรยายอนิเมะ จะกล่าวถึงในหัวข้ออย่างถัดไป

3.5.3 การหาบทบรรยายบนอนิเมะ

ก่อนจะลบทบรรยายนั้น จำเป็นต้องหาบทบรรยายในภาพให้ได้เสียก่อน โดยบทบรรยายของอนิเมะนั้น มักจะใช้ขอบของตัวอักษรเป็นสีดำ อีกทั้งบทบรรยายนั้นจะลอยห่างออกจากขอบของวิดีโอ และขนาดของคำบรรยายนั้นจะมีขนาดอยู่ประมาณหนึ่งไม่ใหญ่หรือไม่เล็กเกินไป ด้วยสมบัตินี้เองทำให้จึงสามารถหาบริเวณบนเฟรมที่เป็นบทบรรยายได้โดยจะมีวิธีหาพื้นที่ซึ่งเป็นบทบรรยายดังนี้

Algorithm 8: Finding subtitle

```

 $D \leftarrow \text{findsub}(\mathbf{u})$ 
• ทำการเปลี่ยนสีดำในภาพ  $\mathbf{u}$  ให้เป็นสีขาวแล้วเปลี่ยนอีกครั้ง ให้เป็นสีดำเพื่อหาขอบของคำ
    บรรยาย
• เปลี่ยนบริเวณสีขาวในภาพให้เป็นสีดำ และเปลี่ยนบริเวณสีดำให้เป็นสีขาว
• ทำการลบทบริเวณสีขาวซึ่งติดกับขอบของภาพออกไป เนื่องจากบทบรรยายจะลอยอยู่ไม่ติดกับ
    ขอบเสมอ
• ลบบริเวณที่ใหญ่เกินกว่าจะเป็นบทบรรยาย
• ลบบริเวณที่เล็กเกินกว่าจะเป็นบทบรรยาย
• ทำการขยายพื้นที่ๆ เป็นสีขาวขึ้นด้วยความกว้างของขอบบทบรรยาย
• สีขาวที่เหลืออยู่ในภาพจะเป็นบทบรรยาย

```

3.5.4 การลบคำบรรยายจากบทอนิเมะ

สำหรับอนิเมะนั้น แต่ละเฟรมจะเป็นรูปภาพ เราจึงสามารถประยุกต์ใช้วิธีการซ่อนแซมภาพจิตรกรรมไทยมาใช้ในการลบคำบรรยายได้ แต่ผู้วิจัยก็ได้สังเกตว่า สำหรับอนิเมะที่เป็นวิดีโอแล้ว ในขณะที่ประมวลผลวิดีโอ เราสามารถใช้ผลการต่อเติมภาพจากภาพที่แล้ว มาใช้เป็นคำตอบเริ่มต้นจึงได้ว่า ขั้นตอนการลบบทบรรยายออกจากวิดีโอมีดังนี้

Algorithm 9: วิธีการทำงานบนวิดีโอ เมื่อต้องการผลลัพธ์จากการลบบทบรรยายออกจากรูปภาพ

```

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

ขั้นตอนวิธี การยึมเฟรม จะเป็นการนำผลลัพธ์จากเฟรมก่อนหน้ามาเป็นคำตอบในการเริ่มต้นในการประมวลผลเพื่อให้ผลลัพธ์ลู่เข้าได้เร็วขึ้น

Algorithm 10: 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 11: 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 10 และขั้นตอนวิธี 11 ที่นำมาประยุกต์ใช้งานร่วมกัน

Algorithm 12: 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

```

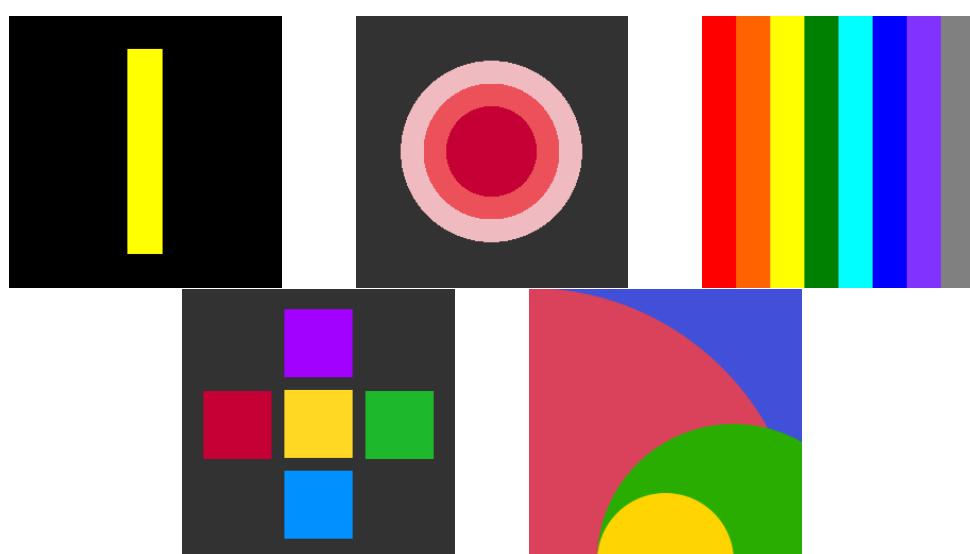
จากนั้นทางผู้พัฒนาจะนำวิธีการแบบบรรยายอนิเมะที่นำเสนอไปทดลองแบบบรรยายเพื่อเปรียบเทียบ
ประสิทธิภาพกับวิธีสปริทเบรกเมนเดิมที่นำมาใช้กับวิดีโอดู

บทที่ 4

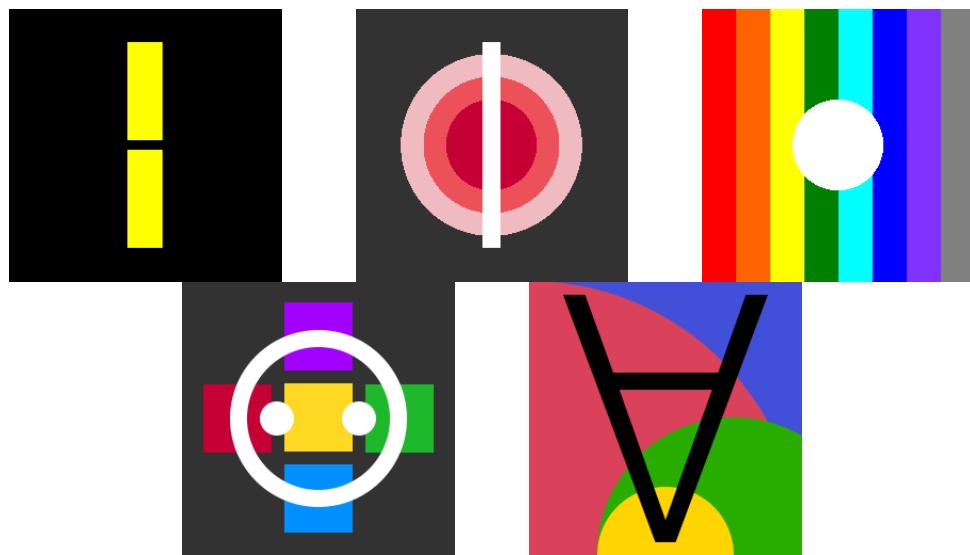
การทดลองเชิงตัวเลข

4.1 การซ่อมแซมภาพจิตรกรรมไทยโบราณ

สำหรับการซ่อมแซมจิตรกรรมไทยโบราณ ผู้วิจัยจะเริ่มจากการทำการปรับปรุงขึ้นตอนวิธีเชิงตัวเลขที่มีอยู่แล้ว โดยใช้ภาพสีที่ได้สังเคราะห์ขึ้นทั้งสิ้น 5 ภาพ โดยแต่ละภาพมีขนาด 256×256 พิกเซล ดังรูปที่ 4.1.1



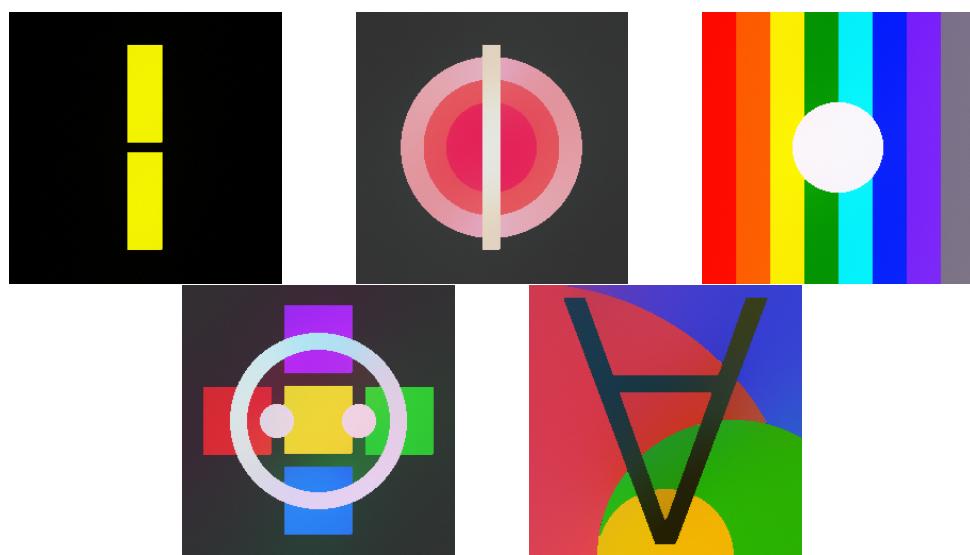
รูปที่ 4.1.1: ภาพต้นฉบับ



รูปที่ 4.1.2: ภาพที่จะทำการซ่อมแซม

4.1.1 การเปรียบเทียบประสิทธิภาพขั้นตอนวิธีเชิงตัวเลขที่มีอยู่แล้ว

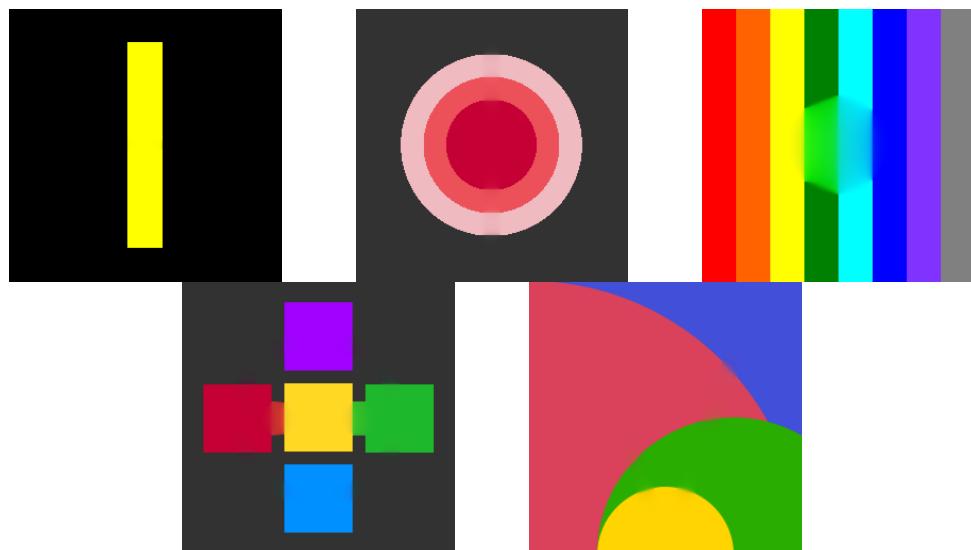
การทดสอบประสิทธิภาพจะใช้ $\varepsilon = 1 \times 10^{-4}$ และ $N = 10,000$ โดยรูปที่ 4.1.3 - 4.1.5 และตารางที่ 4.1 - 4.3 แสดงผลการซ่อมแซมภาพสังเคราะห์ทั้ง 5 ภาพ



รูปที่ 4.1.3: 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

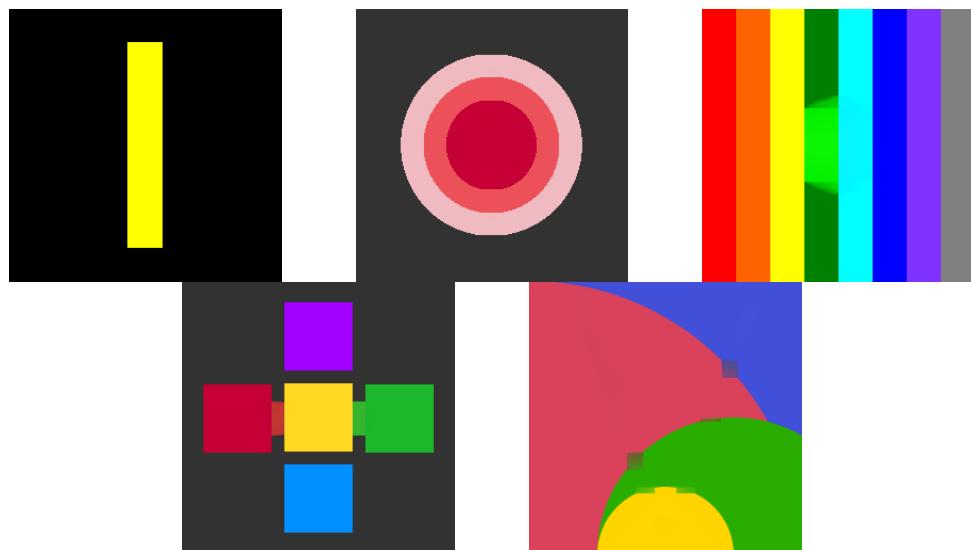
ตารางที่ 4.1: ผลการซ่อมแซมวิธีการเดินเวลา



รูปที่ 4.1.4: ผลการซ่อมแซมจากการวิธีการทำข้ามแบบจุดตึง

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 4.2: ผลการซ่อมแซมของวิธีการทำข้ามแบบจุดตึง



รูปที่ 4.1.5: ผลการซ่อมแซมจากวิธีการสปริทเบรกเมน

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 4.3: ผลการซ่อมแซมของวิธีสปริทเบรกเมน

ประสิทธิภาพของวิธีการเชิงตัวเลขทั้ง 3 วิธี สามารถสรุปได้ดังนี้

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

ตารางที่ 4.4: แสดงการซ่อมแซมเฉลี่ยของวิธีการเชิงตัวเลข

จากทั้ง 3 วิธีที่ได้ทดสอบ จะเห็นได้ว่าวิธีการสปริทเบรกเมนใช้เวลาน้อยกว่าวิธีอื่น และมีคุณภาพ ซึ่งพิจารณาจาก ค่า PSNR และค่า SSIM มากกว่าวิธีอื่น ผู้วิจัยจึงสนใจทำการปรับปรุงวิธีสปริทเบรกเมนให้มีประสิทธิภาพสูง

4.1.2 ขั้นตอนวิธีเชิงตัวเลขสำหรับต่อเติมภาพชนิดใหม่

ตารางที่ 4.5 แสดงผลการเปรียบเทียบประสิทธิภาพของขั้นตอนวิธีการที่ 5 ภายใต้การเปลี่ยนแปลงจำนวนรอบของการทำซ้ำของวิธีการสปริทเบรกเมนบนภาพที่มีความคมชัด 256×256 พิกเซล ตัวอย่าง เช่น $10/3/3/10000$ หมายถึงที่ระดับความคมชัดท้ายบสุดซึ่งมีขนาดเป็น 32×32 พิกเซลจะทำซ้ำไม่เกิน 10 ครั้ง สำหรับที่ความคมชัดละเอียดซึ่นเป็น 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 4.5: 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 4.6: ผลการซ่อมแซมภาพโดยวิธีการเชิงตัวเลขที่นำเสนอด้วยรูปของค่าเฉลี่ยของผลที่ได้จากตารางที่ 4.5

จากตารางที่ 4.6 สังเกตว่า ยิ่งจำนวนการทำข้าวในขั้นที่รูปภาพมีขนาดเล็กจำนวนมากครั้ง จะยิ่งทำให้เวลาประมาณผลที่ใช้ในการต่อเติมภาพใช้เวลาน้อยลง

นอกจากนี้แล้ว ผู้วิจัยยังได้สังเกตอีกว่า การทำข้าวจะถูกเข้าเร็วในช่วงแรก จากนั้นความเร็วในการถูกเข้าจะลดลง ซึ่งทำให้การทำข้าวเพียงไม่กี่ครั้งในระดับความคมชัดเดิม มีผลการซ่อมแซมภาพจนแสดงความคล้ายคลึงกับภาพต้นฉบับได้

ผู้วิจัยจึงกำหนดให้การทำข้าวในระดับความละเอียดสุดเท่ากับ 10 ครั้ง และพบว่าได้ผลการซ่อมแซมดังตารางที่ 4.7

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 4.8: ผลการซ่อมแซมภาพโดยวิธีการเชิงตัวเลขที่นำเสนอด้วยรูปของค่าเฉลี่ยของผลที่ได้จากตารางที่ 4.7

จากตารางจะเห็นว่า การทำข้าในขั้นที่รูปภาพมีขนาดเล็กมากจำนวนมาก ไม่ช่วยให้การประมาณผลได้เร็วขึ้น ผู้วิจัยจึงเลือกใช้การทำข้าแบบ 10/3/3/10 ในการต่อเติมภาพ

4.1.3 การทดสอบประสิทธิภาพในการซ่อมแซมภาพจิตรกรรมไทยโบราณ

ภาพจิตรกรรมที่ใช้ทดสอบ มีทั้งสิ้น 5 ภาพ โดยแต่ละภาพเป็นภาพสีที่มีขนาด 256x256 พิกเซล ซึ่งทั้ง 5 ภาพได้แก่ ภาพที่ 4.1.6a¹ และภาพที่ 4.1.6b² คือ จิตรกรรมฝาผนังวัดแก้วไพฐรย์ ภาพที่ 4.1.6c³ คือ จิตรกรรมฝาผนังวัดพระยืนพุทธบาทยุคล ภาพที่ 4.1.6d⁴ คือ จิตรกรรมฝาผนังวัดคงคaram และภาพที่ 4.1.6e⁵ 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

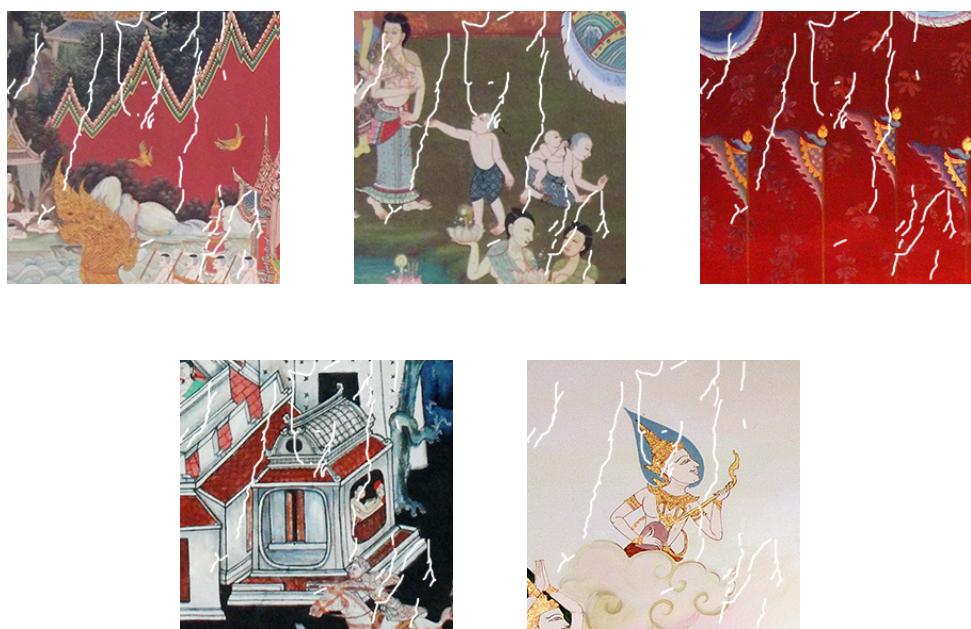
³ภาพถ่ายที่วัดพระยืนพุทธบาทยุคล; ภาพจาก https://commons.wikimedia.org/wiki/File:Wat_Phra_Yuen_Phutthabat_Yukhon_01.jpg สืบคันเมื่อวันที่ 23 กันยายน 2561

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



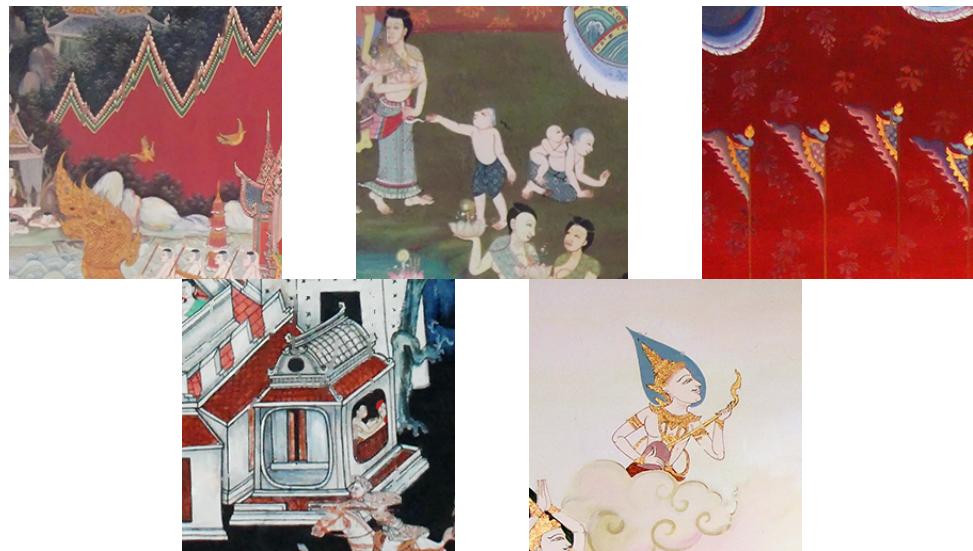
รูปที่ 4.1.6: ภาพต้นฉบับสำหรับใช้ในการทดสอบ



รูปที่ 4.1.7: ภาพที่ทำให้เสียหาย

จากนั้นทำการทดสอบการต่อเติมภาพทั้ง 5 โดยทดสอบวิธีสปริทเบรกเมน และวิธีที่พัฒนาขึ้นโดยใช้วิธี

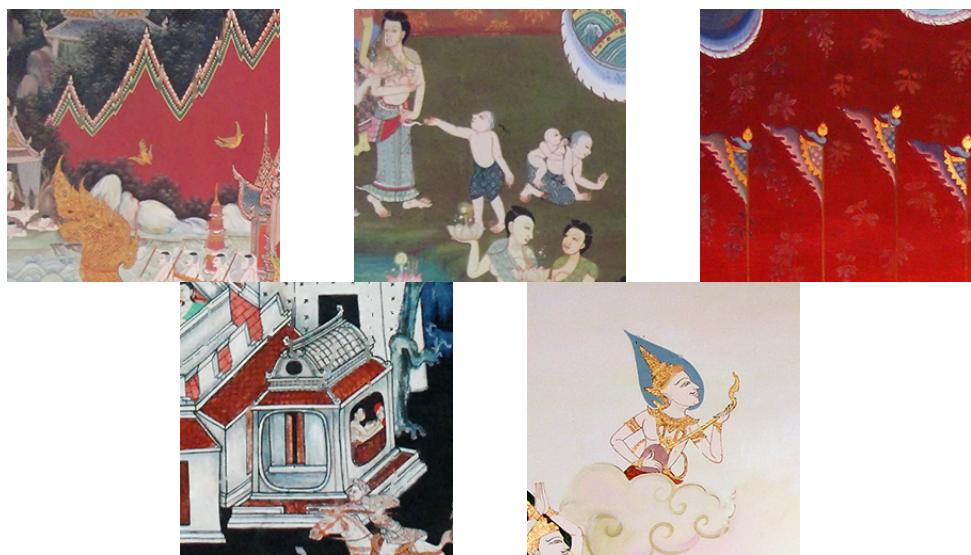
การสปริทเบรกแม่นพร้อมทั้งการใช้พีระมิดรูปภาพที่มีการทำข้าแต่ละชั้นเป็น 10/3/3/10 ได้ผลลัพธ์ออกมาเป็นดังนี้



รูปที่ 4.1.8: 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 4.9: ผลการซ่อมแซมภาพศิลปะไทยจากการสปริทเบรกแม่น



รูปที่ 4.1.9: 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 4.10: ผลการซ่อมแซมภาพศิลปะไทยโดยวิธีการเชิงตัวเลขที่พัฒนาขึ้น

ทั้งสองวิธี ได้ผลลัพธ์การซ่อมแซมภาพศิลปะไทยในรูปค่าเฉลี่ยของมาดังนี้

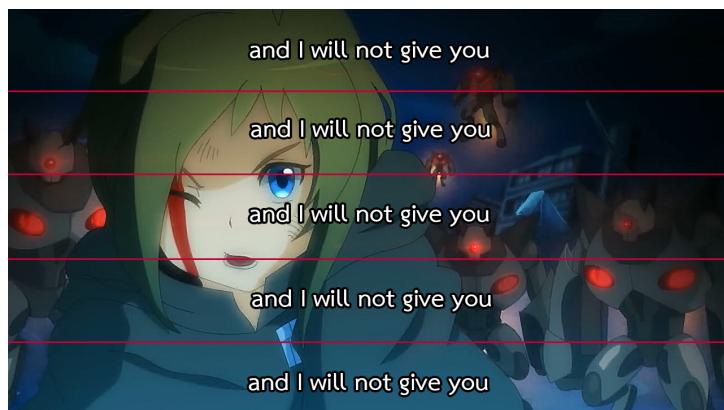
วิธีการ	เวลาประมาณ (วินาที)	PSNR (dB)	SSIM
สปริงเบรกเมน	2.72	34.89	1.0000
วิธีการที่พัฒนาขึ้น	0.39	35.30	1.0000

ตารางที่ 4.11: แสดงผลการซ่อมแซมภาพศิลปะไทยในรูปค่าเฉลี่ยจากตารางที่ 4.9 และตารางที่ 4.10

จากตารางที่ 4.11 จะเห็นได้ว่า วิธีที่พัฒนาขึ้นนั้นสามารถทำงานได้เร็วกว่าวิธีสปริตเบรกเมนเดิม และยังมีคุณภาพที่ดีขึ้นด้วย

4.2 การลับบทบรรยายบนอนิเมะ

สำหรับการลับบทบรรยายอนิเมะ จะใช้วิดีโอ Anime Festival Asia Special Video - feat. Inori Aizawa ซึ่งผลิตโดย Collateral Damage Studios โดยจะตัดวิดีโอด้วยตัวเองจากโดยปกติแล้ว อนิเมะมักมีบรรทัดเพียง 1 ถึง 2 บรรทัด จึงทำการแบ่งวิดีโอออกอีกเป็น 5 ส่วนได้ขนาดเป็น 1280×144 พิกเซลก่อนนำไปทดสอบในลำดับถัดไป และสำหรับบทบรรยายที่จะใช้ทดสอบนั้น เนื่องจากวิดีโอ Anime Festival Asia Special Video - feat. Inori Aizawa ไม่มีคำพูดใดๆ จึงใช้บทความ lorem ipsum เป็นบทบรรยาย โดยจะทำการแสดงบทบรรยาย 1 บรรทัด ความยาว 3 วินาที ทุก 2 วินาที นั่นคือในวิดีโอดังกล่าวจะมีบทบรรยายทั้งสิ้น 20 บรรทัด



รูปที่ 4.2.1: การแบ่งไฟล์วิดีโอด้วยตัวเองเป็น 5 ส่วนสำหรับใช้เป็น 5 ชุดทดสอบ

4.2.1 การหาบทบรรยายบนอนิเมะ

วิธีการหาบทบรรยายที่กล่าวไปข้างต้น จะทำการทดสอบกับบทความ lorem ipsum⁶ ที่ถูกแปลเป็นภาษาไทย ภาษาอังกฤษ และภาษาญี่ปุ่น โดยมีความสามารถในการหาโดยเมนต์เติมใบบทบรรยายภาษาต่างๆ ดังนี้

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

ภาษา	วิดีโอ	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 4.12: 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 4.13: ความคลาดเคลื่อนเฉลี่ยของการหาโดเมนต่อitem ในบทบรรยายภาษาต่างๆ

จากการทดลองทั้ง 3 ภาษาพบว่าวิธีการหาคำบรรยายนี้ มีร้อยละการผิดพลาดเฉลี่ยอยู่ที่ 11.98 ซึ่งการทดลองจากนี้ไปจะใช้วิธีการหาคำบรรยายนี้ในการหาโดเมนต่อitemแบบอัตโนมัติ

4.2.2 การลบคำบรรยายจากบทอนิเมะ

สำหรับอนิเมะนี้ แต่ละเฟรมจะเป็นรูปภาพ เราจึงสามารถประยุกต์ใช้วิธีการซ้อมแซมภาพจิตกรรมไทยมาใช้ในการลบคำบรรยายได้ แต่ผู้จัดก็ได้สังเกตว่า สำหรับอนิเมะที่เป็นวิดีโอแล้ว ในขณะที่ประมวลผลวิดีโอ จะ

สามารถใช้ผลการต่อเติมภาพจากภาพที่แล้ว มาใช้เป็นคำตอบเริ่มต้นจึงได้ว่าขั้นตอนการลับบทบรรยายออกจากวิดีโอ ซึ่งผลลัพธ์เบรียบเทียบระหว่างแบบใช้วิธี ยีมเพرم ใช้วิธีข้ามเพرم และวิธีข้ามและยีมเพرم ได้ผลดังตาราง

วิธีการ	วิดีโอ	เวลาประมาณ (วินาที)	PSNR (dB)	SSIM
สปริทเบรกแม่น และพีระมิดรูปภาพ (ขั้นตอนวิธี 5)	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
ยีมเพرم (ขั้นตอนวิธี 10)	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
ข้ามเพرم (ขั้นตอนวิธี 11)	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
ข้ามและยีมเพرم (ขั้นตอนวิธี 12)	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 4.14: ผลการลับบทบรรยายออกจาكونิเมะด้วยวิธีการเชิงตัวเลขขั้นตอนวิธี 5, 10, 11 และ 12

วิธีการ	เวลาประมาณ (วินาที)	PSNR (dB)	SSIM
สปริทเบรกเมนและพีระมิดรูปภาพ	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 4.15: ผลการซ่อมแซมภาพโดยวิธีการเชิงตัวเลขที่นำเสนอด้วยรูปของค่าเฉลี่ยของผลที่ได้จากตารางที่ 4.14

จากนั้นทำการทดสอบการต่อเติมวิดีโอทั้ง 5 โดยวิธีที่คิดค้นขึ้นใช้วิธีการสปริทเบรกเมนพร้อมทั้งการใช้พีระมิดรูปภาพที่มีการทำข้าแต่ละชั้นเป็น 10/3/3/10 พร้อมทั้งใช้การข้ามเฟรมและยีมเฟรม ได้ผลลัพธ์ออกเป็นดังตารางนี้

วิธีการ	เวลาประมาณ (วินาที)	PSNR (dB)	SSIM
สปริทเบรกเมน	5073.08	32.88	0.9654
วิธีการที่พัฒนาขึ้น	75.76	29.33	0.9454

ตารางที่ 4.16: ผลการลบบทบรรยายออกจากอนิเมะโดยวิธีการสปริทเบรกเมนและวิธีการที่พัฒนาขึ้น

สำหรับวิธีสปริทเบรกเมน เนื่องจากใช้เวลา 1 ชั่วโมงแล้วยังประมาณผลวิดีโอยุ่ดทดสอบแรกไม่เสร็จ ทางผู้พัฒนาจึงตัดสินใจยุติการทดลอง เนื่องจากอาจต้องใช้เวลาการประมาณผลเป็นเวลาหลายชั่วโมงสำหรับวิธีอื่นความยาว 1 นาที ส่วนวิธีที่คิดค้นขึ้น พบว่าสำหรับวิดีโอที่มีความยาว 1 นาที สามารถทำงานได้เสร็จอย่างรวดเร็ว โดยใช้เวลาเพียง 75 วินาที

บทที่ 5

สรุป

สำหรับโครงงานวิจัยนี้ได้ทำการพัฒนาขั้นตอนวิธีเชิงตัวเลขสำหรับการซ่อมแซมภาพศิลปะไทยและการลบบบทรอยยนต์ได้มีผลการดำเนินงานทั้งสิ้นดังนี้

ขั้นตอนวิธีเชิงตัวเลขทั้ง 3 วิธีสำหรับแก้ตัวแบบการต่อเติมภาพด้วยการแปรผันรวม ได้แก่วิธีเดินเวลาแบบชัดแจ้ง วิธีทำข้าจุดตรึง และวิธีการสปอร์ทเบรกแม่น พบร่วม วิธีการสปอร์ทเบรกแม่นมีคุณภาพที่ดีกว่าเมื่อวัดด้วยค่า PSNR และ SSIM ส่วนเวลาที่ใช้ประมวลผลพบว่าวิธีสปอร์ทเบรกแม่นใช้เวลาอยกว่าวิธีเดินเวลา 8 เท่าและใช้เวลาอยกว่าวิธีทำข้าจุดตรึง 5 เท่า

พิริมิดรูปภาพใช้เพื่อเพิ่มความเร็วในการประมวลผลของวิธีการสปอร์ทเบรกแม่น พบร่วมการทำข้าบันพิริมิดรูปภาพด้วยการทำข้า 10/3/3/10 ใช้เวลาประมวลผลน้อยที่สุดและให้ค่าคุณภาพทั้งในด้าน PSNR และ SSIM ใกล้เคียงกวิธีอื่นๆ ถือว่าใช้พิริมิดรูปภาพนี้ในการซ่อมแซมรูปภาพ

การซ่อมแซมภาพศิลปะไทย การใช้วิธีการสปอร์ทเบรกแม่นพร้อมทั้งการใช้พิริมิดรูปภาพ พบร่วมภาพที่ผ่านการต่อเติมให้คุณภาพดีกว่าเมื่อวัดด้วย PSNR และใช้เวลาอยกว่าประมาณ 7 เท่า

ขั้นตอนวิธีค้นหาคำบรรยายในภาพด้วยวิธีการที่คิดค้นขึ้น (ขั้นตอนวิธี 8) พบร่วมมีความผิดพลาดในการตรวจหาพิกเซลที่เป็นบทบรรยายอยู่ที่ร้อยละ 11.98 ซึ่งจะใช้ขั้นตอนวิธีนี้ในการหาคำบรรยายเพื่อทำการลบถัดไป

เนื่องจากวิดีโอเป็นชุดของภาพคณผู้วิจัยจึงได้เสนอขั้นตอนวิธีข้ามเฟรม (ขั้นตอนวิธี 10) ขั้นตอนวิธียึดเฟรม (ขั้นตอนวิธี 11) และขั้นตอนวิธียึดเฟรมและข้ามเฟรม (ขั้นตอนวิธี ?? เพื่อช่วยในการลดเวลาการประมวลผลพบว่าวิธียึดเฟรมและข้ามเฟรมใช้เวลาประมวลผลน้อยสุด ซึ่งใช้เวลาอยกว่าวิธีสปอร์ทเบรกแม่นและพิริมิดรูปภาพบันวิธีอีสิ่ง 2 เท่า

สำหรับการลบบทบรรยายอนิเมะเมื่อใช้วิธีตรวจสอบหาคำบรรยายที่คิดขึ้นพร้อมทั้งใช้วิธีการเดียวกับที่ใช้สำหรับ

ซ่อมแซมศิลปะไทย รวมกับการวิธีการขึ้นเฟรมและข้ามเฟรม พบว่าใช้เวลาหน่อยกว่าวิธีการสปริทเบรคแม่นบนาวีดีโดย 67 เท่า

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.

ภาคผนวก A

โปรแกรมที่พัฒนาขึ้น

A.1 โปรแกรมสำหรับทดสอบ

สำหรับโครงการวิจัยเรื่องนี้เป็นการพัฒนาวิธีเชิงตัวเลข การจะวัดประสิทธิภาพของวิธีการเชิงตัวเลขได้ จำเป็น จะต้องใช้โปรแกรมเข้าทดสอบ โดยโครงการวิจัยนี้ มีโปรแกรมสำหรับทดสอบ ซึ่งสามารถแบ่งได้ออกเป็น 2 ส่วน คือโปรแกรมสำหรับทดสอบขั้นตอนการซ้อมแซมภาพศิลปะไทยและการลบคำบรรยายอนิเมะ

A.1.1 โปรแกรมสำหรับทดสอบการซ้อมแซมภาพศิลปะไทย

โดยโค้ดของโปรแกรมสามารถดาวน์โหลดได้ที่ <https://github.com/pureexe/YaeProgression01-color-image> เป็นโค้ดภาษา C++ โดยถูกพัฒนาบน Visual Studio 2017 และจำเป็นต้องคอมไพล์เดิมก่อนใช้งาน ทั้งนี้ ท่านสามารถดาวน์โหลดไฟล์ที่คอมไпал์เรียบร้อยสำหรับ Windows 64 bit ได้ที่ <https://github.com/pureexe/YaeProgression01-color-image/releases>

โดยการรัน ให้เปิด command prompt โดยการ เปิดปุ่ม Windows+R จากนั้นพิมพ์ cd ”ชื่อโฟลเดอร์” ที่ได้ทำการดาวน์โหลด releases มาแตกไฟล์ไว้ จากนั้นใช้คำสั่ง cd ”application” เพื่อเข้าไปโฟลเดอร์ที่มีไฟล์ โปรแกรมอยู่ แล้วจึงใช้คำสั่ง labarotory.exe เพื่อทำการทดสอบ โดยไฟล์ผลลัพธ์จากการทดสอบจะปรากฏใน โฟลเดอร์ result และจะแสดงเวลาที่ใช้ในการประมวลผลออกทางหน้าจอ

```

C:\Windows\system32\cmd.exe - laboratory.exe
Microsoft Windows [Version 10.0.17763.253]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\pakkapon>cd C:\Users\pakkapon\Desktop\project\
C:\Users\pakkapon\Desktop\project>cd application
C:\Users\pakkapon\Desktop\project\application>laboratory.exe
=====
Experiment 01 - Measurement Exist algorithm
=====
ExplicitTimeMarchingColorInpaint
CASE: 1
TIME: 115.844
PSNR: 25.1665
SSIM: 0.999671
CASE: 2
TIME: 168.871
PSNR: 17.9235
SSIM: 0.997956
CASE: 3
TIME: 171.769
PSNR: 13.3341
SSIM: 0.994067
CASE: 4
TIME: 228.172
PSNR: 12.4009
SSIM: 0.99271
CASE: 5

```

รูปที่ A.1.1: ตัวอย่างโปรแกรมสำหรับทดสอบการซ่อมแซมภาพคลิปไทยที่พัฒนาขึ้น

A.1.2 โปรแกรมทดสอบการลบคำบรรยายอนิเมะ

สำหรับโปรแกรมทดสอบการลบคำบรรยายอนิเมะนี้ เขียนด้วยภาษา power shell เพื่อใช้ในการทดสอบโดยเครื่องที่จะนำไปทดสอบ จำเป็นต้องติดตั้ง ffmpeg, MPC-HC, Avisynth+ และ OpenCV ซึ่งเมื่อติดตั้งแล้วสามารถทำการทดสอบได้โดยการเรียกใช้สคริปต์ <https://github.com/pureexe/matlab-inpaint-speed-analysis/blob/master/experiment-08/taskrunner/test-algorithm/test-20181111.ps1>

```

Windows PowerShell
PS C:\Users\pakkapon\Documents\Github\matlab-inpaint-speed-analysis\experiment-08\taskrunner\test-algorithm> .\test-20181111.ps1
=====
splitbergman
Loop: 10
Processed in 170.6230171
Comparing Result... splitbergman
Compared in 167.0800813
[Parsed_ssims_0 @ 00000159136e40c0] SSIM R:0.700363 (5.234051) G:0.711575 (5.399673) B:0.714408 (5.442540) All:0.708782 (5.357821)
[Parsed_psnsr_1 @ 00000159136e4300] PSNR r:18.908856 g:19.166903 b:18.097365 average:18.700048 min:11.415450 max:inf

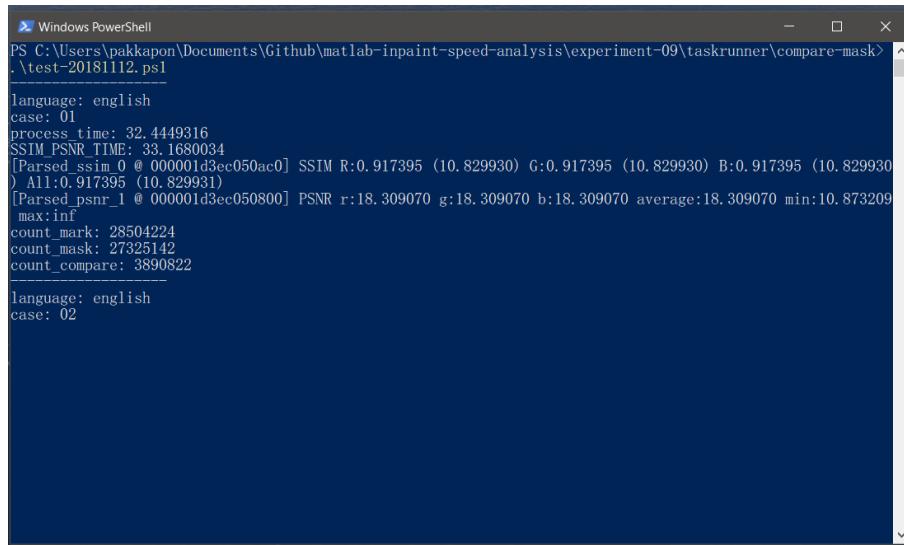
splitbergman
Loop: 10/1/10
Processed in 172.3858612
Comparing Result... splitbergman

```

รูปที่ A.1.2: ตัวอย่างโปรแกรมสำหรับทดสอบการลบคำบรรยายที่พัฒนาขึ้น

นอกจากนี้ในส่วนของการตรวจว่าขั้นตอนการหาคำบรรยายนั้นสามารถทำได้ดีเพียงใด สามารถทดสอบได้

โดยการรันสคริปต์ <https://github.com/pureexe/matlab-inpaint-speed-analysis/blob/master/experiment-09/taskrunner/compare-mask/test-20181112.ps1>



```

Windows PowerShell
PS C:\Users\pakkapon\Documents\Github\matlab-inpaint-speed-analysis\experiment-09\taskrunner\compare-mask> .\test-20181112.ps1
language: english
case: 01
process_time: 32.4449316
SSIM_PSNR_TIME: 33.1680034
[Parsed_ssim_0 @ 000001d3ec050ac0] SSIM R:0.917395 (10.829930) G:0.917395 (10.829930) B:0.917395 (10.829930)
) All:0.917395 (10.829931)
[Parsed_psnr_1 @ 000001d3ec050800] PSNR r:18.309070 g:18.309070 b:18.309070 average:18.309070 min:10.873209
max:inf
count_mark: 28504224
count_mask: 27325142
count_compare: 3890822
language: english
case: 02

```

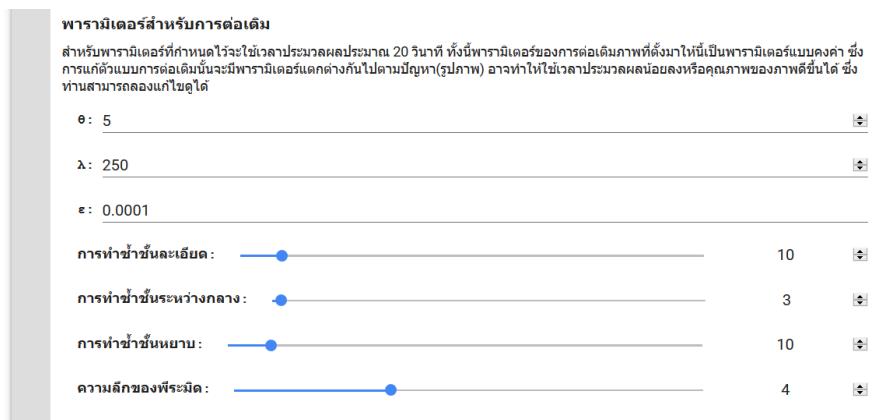
รูปที่ A.1.3: ตัวอย่างโปรแกรมสำหรับทดสอบการหาคำบรรยายที่พัฒนาขึ้น

A.2 โปรแกรมตัวอย่างการซ่อมแซมภาพศิลปะไทย

เนื่องจากโปรแกรมสำหรับทดสอบที่ได้สร้างขึ้นมาบ้านนี้ข้อจำกัดด้านอุปกรณ์ ซึ่งรองรับเพียง Windows 64 bit เท่านั้น ทำให้ไม่สามารถทำงานได้บนอุปกรณ์อื่นๆ ทางผู้วิจัยจึงได้ทำการเขียนขั้นตอนพัฒนาขึ้นใหม่ให้ใช้งานบน Google Colab ได้ ซึ่งสามารถเข้าใช้งานได้ที่ <https://bit.ly/thai-inpaint-colab> ซึ่งนอกจากตัวอย่างที่เตรียมไว้ให้จำนวน 5 ภาพแล้ว ยังสามารถอัปโหลดภาพที่เสียหายพร้อมทั้งโคลเม้นสำหรับการต่อเติมเพื่อทำการซ่อมแซมภาพได้อีกด้วย



รูปที่ A.2.1: ตัวอย่างการเลือกรูปภาพสำหรับทำการทดสอบ



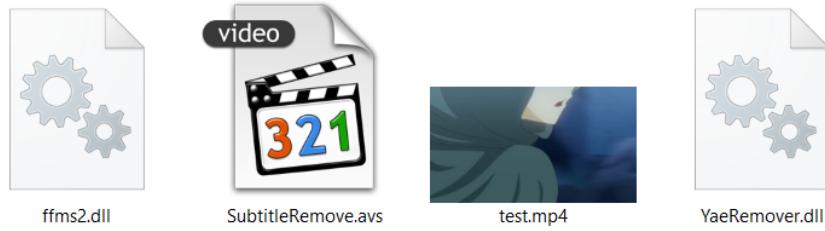
รูปที่ A.2.2: ตัวอย่างการปรับค่าพารามิเตอร์ที่ใช้ในโครงงานวิจัยนี้



รูปที่ A.2.3: ตัวอย่างภาพผลลัพธ์จาก Google Colab

A.3 โปรแกรมตัวอย่างการลบคำบรรยายอนิเมะ

สำหรับการลบคำบรรยายอนิเมะนั้น ขณะนี้ยังรองรับเพียงขอบบทบรรยายที่เป็นสีดำเท่านั้น เครื่องคอมพิวเตอร์ที่จะใช้งาน จะต้องเป็น Windows 64 bit ที่มีการติดตั้ง MPC-HC, Avisynth+ และ OpenCV เสียก่อน จากนั้นดาวน์โหลดตัวอย่างได้ที่ <http://bit.ly/demo-anime-inpaint> เมื่อทำการแตกไฟล์ให้เปิดไฟล์ SubtitleRemove.avs ด้วย MPC-HC เพื่อแสดงตัวอย่าง และสามารถนำไฟล์ตัวอย่างนี้ไปใช้กับวิดีโออนิเมะอื่นได้โดยทำการเปิด SubtitleRemove.avs ด้วยโปรแกรม Text Editor อื่นๆ เช่น Notepad++ เพื่อแก้ไขพารามิเตอร์ Top, Bottom, Left และ Right เพื่อรับที่อยู่ตำแหน่งของคำบรรยายในหน่วยพิกเซล อีกทั้งแก้ไขพารามิเตอร์ StokeWidth เพื่อแก้ไขตัวความหนาของคำบรรยายในหน่วยพิกเซล



รูปที่ A.3.1: ไฟล์ตัวอย่างเมื่อทำการแตกไฟล์ออกมาแล้ว test.mp4 เป็นวิดีโอมีคำบรรยาย และ SubtitleRemove.avs เป็นโปรแกรมตัวอย่างสำหรับลบคำบรรยาย



รูปที่ A.3.2: test.mp4 (ซ้าย) และ SubtitleRemove.avs (ขวา) เมื่อเปิดด้วย MPC-HC

```

1 LoadPlugin("ffms2.dll") #https://github.com/FFMS/ffms2
2 LoadPlugin("YaeRemover.dll") #our method
3 Video = FFMS2("test.mp4", ATrack=-1, fpsnum=24000,
   fpsden=1000, ColorSpace="RGB24", UTF8=True)
4
5 Result = Video.YaeRemover(Left = 280, Right = 1000, Top =
   613, Bottom = 683, StrokeWidth = 6)
6 return Result

```

รูปที่ A.3.3: SubtitleRemove.avs สามารถแก้พารามิเตอร์เพื่อใช้กับวิดีโອนิเมะอื่นได้