

ขั้นตอนวิธีใช้ตัวเลขชนิดใหม่สำหรับการต่อเติมภาพที่ใช้การแปรผันรวมกับ[†]
การประยุกต์สำหรับซ่อมแซมภาพจิตรกรรมไทยโบราณ
และการลบบทบรรยายจากอนิเมะ

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
ผู้ช่วยศาสตราจารย์ ดร. นพดล ชุมชอบ

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

สาขาวิชาคณิตศาสตร์ ภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์ มหาวิทยาลัยศิลปากร มีความเห็นชอบให้โครงการวิจัยเรื่องขั้นตอนวิธีใช้ตัวเลขชนิดใหม่สำหรับการต่อเติมภาพที่ใช้การแปรผันรวมกับการประยุกต์สำหรับซ่อมแซมภาพจิตรกรรมไทยโบราณและการลบบทบรรยายจากอนิเมะ (A new numerical algorithm for TV-based image inpainting with its applications for restoring ancient Thai painting images and removing subtitles from animes) ชื่อเสนอโดย นายภัคพล พงษ์ทวี รหัส 07580028 เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาคณิตศาสตร์ ประจำปีการศึกษา 2561

ผู้ช่วยศาสตราจารย์ ดร. นพดล ชุมชوب อาจารย์ที่ปรึกษา

_____ / _____ / _____

ผู้ช่วยศาสตราจารย์ ดร. นัยนรัตน์ กันยะมี กรรมการ

_____ / _____ / _____

ผู้ช่วยศาสตราจารย์ ดร. พรทรัพย์ พรสวัสดิ์ กรรมการ

_____ / _____ / _____

บทคัดย่อ

ตัวแบบการแปรผันได้รับการยอมรับอย่างกว้างขวางเพื่อนำมาใช้ต่อเติมภาพ เนื่องจากสมบัติการอนุรักษ์เส้นของเร็กกิวลาร์ไฮซ์เซชันแบบการแปรผันรวม อย่างไรก็ตาม การพัฒนาวิธีการเชิงตัวเลขที่มีประสิทธิภาพสำหรับสร้างคำตอบซึ่งสอดคล้องกับสมบัติดังกล่าวเป็นงานที่ท้าทาย เนื่องจากเร็กกิวลาร์ไฮซ์เซชันแบบการแปรผันรวมมีสมบัติซึ่งหอนุพันธ์ไม่ได้และไม่เป็นเชิงเส้น ในงานวิจัยนี้ เรายังคงดำเนินการแก้ตัวแบบดังกล่าวจากความสำเร็จของขั้นตอนวิธีการเชิงตัวเลขสำหรับปัญหาการจำจัดสัญญาณรบกวนออกจากภาพ เราได้นำเสนอ

วิธีการสเปรย์เบรกเม่น สำหรับแต่ละรอบของการทำซ้ำ การคำนวณของวิธีการนี้ต้องการแก้ 2 ปัญหาย่อย ในปัญหาย่อยแรก เราพบความยุ่งยากในการหาคำตอบแม่นตรง สำหรับปัญหาย่อยที่ 2 คำตอบได้นำเสนออยู่ในรูปแบบปิด ทั้งนี้ เราได้นำเสนอขั้นตอนวิธีเชิงตัวเลขแบบใหม่ (ขั้นตอนวิธีแรก) ที่ใช้การหาค่าต่ำสุดแบบสลับในกรอบความคุณภาพที่ต้องการ เพื่อสร้างคำตอบเชิงตัวเลขที่รวดเร็วและแม่นยำ หลังจากนั้น เราได้ศึกษาปัญหาการลบบทบรรยายออกจากอนิเมะ เราพบว่าตัวแบบการต่อเติมภาพดังกล่าวสามารถนำมารับประทานเพื่อให้ผลลัพธ์ที่น่าพอใจ ใน การแก้ตัวแบบที่ปรับปรุง เรานำเสนอขั้นตอนวิธีการข้ามและการยืด (ขั้นตอนวิธีที่สอง) ซึ่งใช้ขั้นตอนวิธีแรกเพื่อกำจัดบทบรรยายอย่างมีประสิทธิภาพ การทดสอบบนภาพสังเคราะห์และภาพศิลปะไทยโดยรวมยืนยันว่า ขั้นตอนวิธีแรกมีประสิทธิภาพสูงกว่าวิธีการพื้นฐาน นอกจากนี้เราพบว่า ขั้นตอนวิธีที่สองได้นำไปสู่ผลการลบบทบรรยายจากอนิเมะที่มีคุณภาพสูงอย่างรวดเร็ว

คำสำคัญ: ขั้นตอนวิธีเชิงตัวเลข, การแปรผันรวม, การต่อเติมภาพ, ภาพศิลปะไทย, อนิเมะ

Abstract

The classical total variation (TV) model has made great successes in image inpainting due to the edge-preserving property of the TV regularization. However, it is difficult in developing an efficient numerical method to ensure that numerical solutions satisfy this requirement because of the non-differentiability and non-linearity of the TV regularization. In this work we focus on computational challenges arising in approximately solving TV-based image inpainting model. Motivated by many efficient numerical algorithms in image denoising, we propose to use the so-called split Bregman method (SBM) in this work. At each iteration, the computation of our proposed SBM requires to solve two subproblems. On one hand for the first subproblem, it is difficult to obtain exact solution. On the other hand for the second subproblem, it has a closed-form solution. To this end, we propose a new numerical algorithm (our first algorithm) based on an alternating minimization method in a multi-resolution framework to obtain a fast and accurate numerical solution for TV-based image inpainting model. We further study the problem of removing subtitles from animes. It is found that TV-based image inpainting model can be improved to deliver visually pleasing results. In order to solve the modified model, we propose the skipping and borrowing algorithm (our second algorithm) including the first algorithm to efficiently eliminate subtitles from the animes. Numerical tests on synthetic and real ancient Thai painting images which confirm first that our first algorithm is more computationally efficient than some traditional methods in producing the high quality results. Second, the numerical tests show that the second algorithm is fast in delivering high quality of the restored animes.

Keywords: numerical method, total Variation, image inpainting, thai painting images, anime

กิตติกรรมประกาศ

การจัดทำโครงการวิจัยครั้นี้สามารถสำเร็จลุล่วงไปได้ด้วยดีเนื่องจากผู้วิจัยได้รับความกรุณาช่วยเหลือจาก ผู้ช่วยศาสตราจารย์ ดร.นพดล ชุมชอบ อาจารย์ที่ปรึกษาโครงการวิจัยผู้ให้ความรู้ ให้คำแนะนำและแก้ไขข้อบกพร่องต่างๆ ตลอดช่วงของการทำโครงการวิจัย คณะผู้วิจัยจึงขอกราบขอบพระคุณเป็นอย่างสูง ณ โอกาสนี้

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.นันดรัตน์ กันยะมี และผู้ช่วยศาสตราจารย์ ดร.พรทรัพย์ พรสวัสดิ์ กรรมการสอบโครงการวิจัยสำหรับคำแนะนำที่มีประโยชน์ รวมถึงอาจารย์วิทยากรทุกท่านที่สอนใช้งานโปรแกรมเลเท็กซ์ (LaTeX)

ขอขอบพระคุณอาจารย์ประจำภาควิชาคณิตศาสตร์ทุกท่านที่ได้อบรมสั่งสอนมอบวิชาความรู้อันมีค่าแก่ คณะผู้วิจัย และขอขอบพระคุณเจ้าหน้าที่ประจำภาควิชาคณิตศาสตร์ที่ได้อำนวยความสะดวกด้านต่างๆ ในการจัดทำโครงการวิจัยฉบับนี้

ขอขอบพระคุณบิดา มารดา และเพื่อนๆ ผู้ให้กำลังใจ รวมไปถึงการให้ความช่วยเหลือในทุกๆ ด้านเสมอมา และหวังไว้อย่างยิ่งว่าโครงการวิจัยครั้นี้จะเป็นประโยชน์แก่ผู้ที่สนใจและผู้ที่เกี่ยวข้องต่อไป

สารบัญ

บทคัดย่อ

i

บทคัดย่อ

ii

กิตติกรรมประกาศ

iii

สารบัญรูป

viii

สารบัญตาราง

ix

1 บทนำ

1

1.1	ปัญหาของการต่อเติมภาพ	1
1.2	การประยุกต์ของการต่อเติมภาพ	2
1.2.1	การซ่อมแซมภาพจิตรกรรมไทยโบราณ	2
1.2.2	การลบบทบรรยายจากอนิเมช	4
1.3	วัตถุประสงค์ของโครงงานวิจัย	5

2 ความรู้พื้นฐานทางคณิตศาสตร์

6

2.1	ปริภูมิที่มีค่าประจำ	6
2.2	แคลคูลัสของการแปรผันเบื้องต้น	7
2.3	ฟังก์ชันของการแปรผันที่มีข้อบกเขต	11
2.4	ปัญหาแบบอิลล์โพสและวิธีการเร็กกิวลาร์ไลซ์เซชัน	12

iv

2.4.1	วิธีการเร็กเกิลาร์เรซเซชัน	13
2.5	วิธีการไฟน์ดิฟเฟอเรนซ์เบื้องต้น	14
2.6	วิธีการเกาส์ไซเดลสำหรับระบบสมการเชิงเส้น	15
2.7	การวัดคุณภาพของภาพที่ผ่านกระบวนการต่อเติม	16
2.7.1	Peak signal-to-noise ratio (PSNR)	16
2.7.2	Structral Similarity index (SSIM)	16
3	ขั้นตอนวิธีเชิงตัวเลขชนิดใหม่	17
3.1	การนำเสนอดูภาพ	17
3.1.1	การนำเสนอภาพแสดงเทา	17
3.1.2	การนำเสนอภาพสี	18
3.2	ตัวแบบเชิงแปรผันสำหรับต่อเติมภาพแสดงเทา	18
3.3	ตัวแบบเชิงแปรผันสำหรับการต่อเติมภาพสี	19
3.4	ขั้นตอนวิธีเชิงตัวเลขพื้นฐาน	19
3.4.1	การเดินเวลาแบบชัดแจ้ง	19
3.4.2	วิธีการทำตัวแบบจุดต้อง	21
3.4.3	วิธีการสปริทเบรกแม่น	23
3.5	ขั้นตอนวิธีเชิงตัวเลขที่นำเสนอดู	26
3.5.1	ขั้นตอนวิธีเชิงตัวเลขสำหรับต่อเติมภาพศิลปะ	26
3.5.2	ขั้นตอนวิธีเชิงตัวเลขสำหรับซ่อมแซมภาพวิดีโอ	30
3.5.3	การทำทบทวนรายบันฉบับนี้	31
3.5.4	การลบบทบรรยายจากบทฉบับนี้	32
4	การทดลองเชิงตัวเลข	34
4.1	การซ่อมแซมภาพจิตรกรรมไทยโบราณ	34
4.1.1	การเปรียบเทียบประสิทธิภาพขั้นตอนวิธีเชิงตัวเลขที่มีอยู่แล้ว	35
4.1.2	การทดสอบการซ่อมแซมภาพตามวิเตอร์	39
4.1.3	การทดสอบประสิทธิภาพในการซ่อมแซมภาพจิตรกรรมไทยโบราณ	43
4.2	การลบบทบรรยายบทฉบับนี้	47

4.2.1	การหาบทบรรยายบนอนิเมะ	47
4.2.2	การลับบทบรรยายจากบทอนิเมะ	48
5 สรุป		51
บรรณานุกรม		53
ภาคผนวก A		55
A โปรแกรมที่พัฒนาขึ้น		55
A.1	โปรแกรมทดสอบ	55
A.1.1	โปรแกรมทดสอบการซ่อมแซมภาพศิลปะไทย	55
A.1.2	โปรแกรมทดสอบการลับบทบรรยายจากอนิเมะ	56
A.2	โปรแกรมตัวอย่างการซ่อมแซมภาพศิลปะไทย	57
A.3	โปรแกรมตัวอย่างการลับบทบรรยายจากอนิเมะ	58

สารบัญ

1.1.1	ตัวอย่างการซ่อมแซมภาพ	2
1.2.1	ภาพจิตรกรรมไทยที่วัดภูมิทรัตน์ อำเภอเมือง จังหวัดป่า恩	3
1.2.2	1 เฟรมของอนิเมะที่มีบทบรรยายแบบแข็ง	4
2.3.1	(ข้าย) พังก์ชันที่มีขอบเขตกับการแปรผันรวมที่มีค่าเดียวกัน (ขวา) พังก์ชันที่มีขอบเขต กับการแปรผันรวมอนันต์	12
2.4.1	a	13
3.1.1	ตัวอย่างภาพเดดเทาที่แสดงระดับความเข้มของภาพ	17
3.2.1	D แทนโดเมนต่อเติมของ z	18
3.5.1	วิธีการพิริมิดรูปภาพ	27
3.5.2	ผลการซ่อมแซมภาพวิเคราะห์เมื่อใช้จำนวนรอบในการทำข้าวที่ระดับความคมชัดสูงสุด ซึ่งมีค่าต่างกัน	30
4.1.1	ภาพต้นฉบับ	35
4.1.2	ภาพที่จะทำการซ่อมแซม	35
4.1.3	ผลการซ่อมแซมจากวิธีการเดินเวลา	36
4.1.4	ผลการซ่อมแซมจากวิธีการทำข้าวแบบจุดตึง	37
4.1.5	ผลการซ่อมแซมจากวิธีการสปริทเบรกแม่น	38
4.1.6	ภาพต้นฉบับสำหรับใช้ในการทดสอบ	44
4.1.7	ภาพที่ทำให้เสียหาย	44
4.1.8	ผลการซ่อมแซมโดยวิธีการสปริทเบรกแม่น	45

4.1.9	ผลการซ่อมแซมภาพโดยวิธีการเชิงตัวเลขที่พัฒนาขึ้น	46
4.2.1	การแบ่งไฟล์วิดีโอเป็น 5 ส่วนสำหรับใช้เป็น 5 ชุดทดสอบ	47
A.1.1	ตัวอย่างโปรแกรมสำหรับทดสอบการซ่อมแซมภาพศิลปะไทยที่พัฒนาขึ้น	56
A.1.2	ตัวอย่างโปรแกรมสำหรับทดสอบการลบคำบรรยายที่พัฒนาขึ้น	56
A.1.3	ตัวอย่างโปรแกรมสำหรับทดสอบการหาคำบรรยายที่พัฒนาขึ้น	57
A.2.1	ตัวอย่างการเลือกรูปภาพสำหรับทำการทดสอบ	57
A.2.2	ตัวอย่างการปรับค่าพารามิเตอร์ที่ใช้ในโครงงานวิจัยนี้	58
A.2.3	ตัวอย่างภาพผลลัพธ์จาก Google Colab	58
A.3.1	ไฟล์ตัวอย่างเมื่อทำการแตกไฟล์ออกมาแล้ว test.mp4 เป็นวิดีโอมีคำบรรยาย และ SubtitleRemove.avs เป็นโปรแกรมตัวอย่างสำหรับลบคำบรรยาย	59
A.3.2	test.mp4 (ซ้าย) และ SubtitleRemove.avs (ขวา) เมื่อเปิดด้วย MPC-HC	59
A.3.3	SubtitleRemove.avs สามารถแก้พารามิเตอร์เพื่อใช้กับวิดีโອอนิเมชันได้	59

สารบัญตาราง

4.1	ผลการซ่อมแซมวิธีการเดินเวลา (Algorithm 1)	36
4.2	ผลการซ่อมแซมของวิธีการทำข้ามแบบจุดตึง (Algorithm 2)	37
4.3	ผลการซ่อมแซมของวิธีสปริตเบรกแม่น (Algorithm 4)	38
4.4	แสดงการซ่อมแซมเฉลี่ยของวิธีการเชิงตัวเลข	38
4.5	ผลการซ่อมแซมภาพโดยวิธีการเชิงตัวเลขที่นำเสนอน	40
4.6	ผลการซ่อมแซมภาพโดยวิธีการเชิงตัวเลขที่นำเสนอในรูปของค่าเฉลี่ยของผลที่ได้จากตารางที่ 4.5	41
4.7	ผลการซ่อมแซมภาพโดยวิธีการเชิงตัวเลขที่นำเสนอยื่นใช้การทำข้ามระดับความคมชัดละเอียดสุด 10 ครั้ง	42
4.8	ผลการซ่อมแซมภาพโดยวิธีการเชิงตัวเลขที่นำเสนอในรูปของค่าเฉลี่ยของผลที่ได้จากตารางที่ 4.7	43
4.9	ผลการซ่อมแซมภาพศิลปะไทยจากวิธีการสปิตเบรกแม่น (Algorithm 4)	45
4.10	ผลการซ่อมแซมภาพศิลปะไทยโดยวิธีการเชิงตัวเลขที่พัฒนาขึ้น (Algorithm 5)	46
4.11	แสดงผลการซ่อมแซมภาพศิลปะไทยในรูปค่าเฉลี่ยจากตารางที่ 4.9 และตารางที่ 4.10	46
4.12	ความคลาดเคลื่อนของการหาโดยเมนต์ต่อเติม ในบทบรรยายภาษาต่างๆ	48
4.13	ความคลาดเคลื่อนเฉลี่ยของการหาโดยเมนต์ต่อเติม ในบทบรรยายภาษาต่างๆ	48
4.14	ผลการลับบทบรรยายออกจากอนิเมะด้วย Algorithm 5, 10, 11 และ 12	49
4.15	ผลการซ่อมแซมภาพโดยวิธีการเชิงตัวเลขที่นำเสนอในรูปของค่าเฉลี่ยของผลที่ได้จากตารางที่ 4.14	50
4.16	ผลการลับบทบรรยายออกจากอนิเมะโดยวิธีการสปริตเบรกแม่นและวิธีการที่พัฒนาขึ้น	50

บทที่ 1

บทนำ

1.1 ปัญหาของการต่อเติมภาพ

ในปัจจุบันการใช้ภาพดิจิตัล (digital images) ในสังคมเครือข่ายได้รับความนิยมอย่างแพร่หลาย เนื่องจากโทรศัพท์เคลื่อนที่มีราคาถูกลงแต่มีความสามารถที่ชาญฉลาด สามารถทำหน้าที่ได้ตั้งแต่การเป็นกล้องดิจิตัลคอมแพค (compact digital camera) คุณภาพดีให้ภาพดิจิตัลที่มีความคมชัดสูงจนไปถึงการทำหน้าที่ดังเช่น เครื่องคอมพิวเตอร์ส่วนบุคคลที่สามารถเข้ามายื่นต่อ กับระบบเครือข่ายไร้สายเพื่อรับส่งภาพดิจิตัลในสังคมเครือข่าย ด้วยความสะดวกและรวดเร็ว

นอกจากภาพดิจิตัลจะได้รับจากการถ่ายภาพด้วยโทรศัพท์เคลื่อนที่แล้ว ภาพดิจิตัลยังได้รับการถ่ายภาพด้วยกล้องดิจิตอล (หรือ กล้องสะท้อนเลนส์เดี่ยวแบบดิจิตัล (digital single lens reflex camera)) กล้องโทรทรศน์ (หรือ กล้องดูดาว) หรือ เครื่องมือสร้างภาพถ่ายทางการแพทย์ (medical imaging device)

โดยทั่วไปภาพดิจิตัลจะได้รับการประมวลผลภาพก่อนนำไปใช้งานเพื่อให้สามารถใช้ข้อมูลที่ปรากฏบนภาพได้ตรงวัตถุประสงค์ของการใช้งานมากที่สุด ตัวอย่างเช่น ภาพบุคคล (portrait) จะจำเป็นต้องได้รับการกำจัดสัญญาณรบกวนออกจากภาพและ/หรือปรับเพิ่มความละเอียดข้อมูลของความเข้มของสีและความสว่างของสีบนบริเวณใบหน้าก่อนนำภาพไปใช้งานเพื่อจัดทำต้นฉบับวารสารหรือหนังสือของสำนักพิมพ์ เป็นต้น

การต่อเติมภาพ (image inpainting) เป็นวิธีการประมวลผลภาพชนิดหนึ่งมีเป้าหมายเพื่อซ่อมแซมภาพด้วยการต่อเติมข้อมูลของความเข้มของสีบนบริเวณที่กำหนด (ต่อไปจะเรียกบริเวณนี้ว่าโดเมนต่อเติม (inpainting domain)) โดยอาศัยข้อมูลของความเข้มของสีที่ปรากฏในภาพ ตัวอย่างเช่น รูปที่ 1.1.1 (a) แสดงภาพที่ต้องการซ่อมแซมระดับความเข้มของสีบนบริเวณแห่งวัตถุปั่นสีเหลี่ยมสีขาว การต่อเติมภาพดังกล่าวจะเริ่มด้วยการ

กำหนดให้บริเวณแห่งวัตถุรูปร่างสีเหลี่ยมสีขาวเป็นโดเมนการต่อเติมดังรูปที่ 1.1.1 (b) จากนั้นภาพที่ได้รับการซ่อมแซมหรือภาพที่ได้รับการต่อเติม (restored or inpainted image) และในรูปที่ 1.1.1 (c) ได้มาจากขั้นตอนวิธีการต่อเติมภาพ (inpainting algorithm) ซึ่งได้รับการออกแบบเพื่อนำข้อมูลที่ปรากฏบนภาพในบริเวณใกล้เคียงกับขอบของโดเมนต่อเติมมาซ่อมแซมภาพ



(a) ภาพที่ต้องการซ่อมแซม

(b) โดเมนต่อเติม

(c) ภาพที่ได้รับการซ่อมแซม

รูปที่ 1.1.1: ตัวอย่างการซ่อมแซมภาพ

หากที่ผู้วิจัยศึกษาและค้นคว้ามาจนถึงขณะนี้ ผู้วิจัยพบว่าการต่อเติมภาพมักนิยมนำไปใช้งานสำหรับการปรับแต่งความสวยงามของภาพบุคคลที่ถ่ายจากโทรศัพท์เคลื่อนที่ เช่น การลบร่องรอยของรอยตีนกา การลบร่องรอยแพลเป็นที่เกิดจากสิ่งเสี้ยน การลดร่องรอยของความชรา หรือ การเพิ่มความใสและความเนียนของสีผิวนบนบริเวณใบหน้าผ่านโปรแกรมแอปพลิเคชันแต่งรูปภาพที่มีอยู่ในแอปสโตร์ (App Store) หรือ ぐุเกิลเพลย์ (Google Play) เป็นต้น

1.2 การประยุกต์ของการต่อเติมภาพ

1.2.1 การซ่อมแซมภาพจิตรกรรมไทยโบราณ

ภาพจิตรกรรมไทย คือ ภาพเขียนที่มีเอกลักษณ์ความเป็นศิลปะไทยซึ่งโดดเด่นและแตกต่างจากภาพเขียนของชนชาติอื่น ซึ่งไทยในอดีตได้สร้างสรรค์ลวดลายและสีสันบนภาพวาดเพื่อสะท้อนประเพณีและวัฒนธรรมในสังคมไทยที่เกี่ยวกับศาสนา ประวัติศาสตร์ โบราณคดี ชีวิตความเป็นอยู่ วัฒนธรรมการแต่งกาย ตลอดจนการแสดงการเล่นพื้นเมืองต่าง ๆ ของแต่ละยุคสมัย

อย่างไรก็ตาม ภาพจิตรกรรมไทยโบราณจำนวนไม่น้อยได้เสื่อมสภาพตามกาลเวลา และรอคوعการซ่อมแซมจากช่างในสมัยปัจจุบันที่ต้องไม่สร้างความเสียหายให้กับภาพเขียนเพิ่มขึ้นมากกว่าเดิม ที่ผ่านมาภาพ

ที่ซึ่งรับซ่อมแซมมาแล้วจำนวนไม่น้อยได้รับความเสียหายหลังจากการซ่อมแซม ถึงแม้สภาพโดยรวมของภาพจิตรกรรมเดิมยังคงอยู่ แต่รายละเอียดในตัวภาพเขียนได้เปลี่ยนไป ก่อให้เกิดความเสียหายที่ประเมินค่าไม่ได้

การซ่อมแซมภาพจิตรกรรมไทยโบราณโดยใช้ขันตอนวิธีการต่อเติมภาพเป็นขันตอนของการซ่อมแซมแบบหนึ่งซึ่งไม่ก่อให้เกิดความเสียหายใด ๆ กับภาพเดิม เนื่องจากเป็นการซ่อมแซมโดยการใช้ขันตอนวิธีเชิงตัวเลขบนภาพ ดิจิตอลซึ่งเป็นสำเนาของภาพเดิม ด้วยเหตุผลดังกล่าว ผู้วิจัยได้เลือกเห็นว่าการซ่อมแซมภาพจิตรกรรมไทยโบราณมีความจำเป็นเร่งด่วน เนื่องจากภาพที่ได้รับการซ่อมแซมด้วยการต่อเติมภาพสามารถนำไปใช้ประกอบการตัดสินใจเพื่อวางแผนก่อนการลงมือซ่อมแซมภาพเขียนจริงได้ นอกจากนี้ ขันตอนวิธีการต่อเติมภาพสามารถนำไปใช้สร้างแอปพลิเคชันบนโทรศัพท์เคลื่อนที่เพื่อในการใช้เป็นข้อมูลในการเข้าชมภาพเขียนเดิมที่ยังไม่ได้รับการซ่อมแซมและภาพเขียนที่ได้รับการซ่อมแซมโดยวิธีการทางคณิตศาสตร์จากแอปพลิเคชันที่พัฒนาขึ้น

รูปที่ 1.2.1 แสดงตัวอย่างภาพจิตรกรรมไทย¹ ที่ต้องได้รับการซ่อมแซมนบบวณแขนเสือของรูปวด ผู้ชายที่มีส่วนของสีแดงเดิมหลุดหายไป ทั้งนี้ในการซ่อมแซมภาพโดยการต่อเติมภาพ เราจะเริ่มด้วยการสร้างโฉเมนต่อเติมนบบวณเสือพิวปุนที่แขนเสือ จากนั้นจึงนำขันตอนวิธีการต่อเติมภาพเพื่อซ่อมแซมภาพบวณแขนเสือให้เป็นสีแดง



รูปที่ 1.2.1: ภาพจิตรกรรมไทยที่วัดภูมินทร์ อำเภอเมือง จังหวัดน่าน

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

1.2.2 การลับบทบรรยายจากอนิเมะ

อนิเมะคือวิดีโอภาพวิดีโอการ์ตูนสแตลล์ญี่ปุ่นซึ่งเป็นที่นิยมของเยาวชนไทย ในการรับชมอนิเมะ แม้ว่า เยาวชนไทยสามารถรับชมอนิเมะด้วยบทพากย์เสียงภาษาไทยอย่างถูกต้องตามลิขสิทธิ์การเผยแพร่ แต่ก็สูญเสียอรรถรสของการรับชมจากการลับบทบรรยายแบบแข็ง² (hardsub) ที่เป็นภาษาต่างประเทศในบริเวณด้านล่างของจอภาพ อย่างไรก็ตาม ในการซ้อมแซมอนิเมะด้วยการลับบทบรรยายภาษาต่างประเทศจะเป็นงานที่ยุ่งยากและท้าทายมาก เนื่องจาก

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

ด้วยความท้าทายข้างต้น การพัฒนาขั้นตอนวิธีการต่อเติมภาพที่สามารถกำหนดโดยเมนต์อัตโนมัติ ให้กับแต่ละเฟรมและประมวลผลได้แม่นยำจนการลับบทบรรยายสามารถทำงานได้แบบเรียลไทม์จึงเป็นสิ่งจำเป็นที่หลีกเลี่ยงไม่ได้

รูปที่ 1.2.2 แสดงตัวอย่าง 1 เฟรมของอนิเมะ³ที่มีบทบรรยายแบบแข็ง ที่ต้องซ้อมแซมด้วยการลับบทบรรยายออก ทั้งนี้ในการลับบทบรรยายออกจากเฟรมโดยใช้การต่อเติมภาพ เราจะเริ่มด้วยการสร้างโดยเมนต์อัตโนมัติในบริเวณที่มีบทบรรยาย 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) ศึกษาวิธีการแปรผันและขั้นตอนวิธีการเชิงตัวเลขสำหรับปัญหาการต่อเติมภาพสีเทาและภาพสีในระบบ RGB
- (2) พัฒนาขั้นตอนวิธีต่อเติมภาพชนิดใหม่สำหรับซ่อมแซมภาพจิตรกรรมไทยและลับบทบรรยายจากอนิเมะ
- (3) นำขั้นตอนวิธีที่พัฒนาขึ้นไปใช้ในการซ่อมแซมภาพจิตรกรรมไทยและลับบทบรรยายในอนิเมะ

บทที่ 2

ความรู้พื้นฐานทางคณิตศาสตร์

2.1 ปริภูมิที่มีค่าประจำ

บทนิยาม 1. (ค่าประจำ) ค่าประจำบนปริภูมิเวกเตอร์เชิงเส้น V คือฟังก์ชันค่าจริง $\|\cdot\|$ ซึ่งนิยามบน V และสอดคล้องสมบัติต่อไปนี้

1. $\|u\| > 0$ เมื่อ $u \neq 0 \in V$
2. $\|\lambda u\| = |\lambda| \|u\|$ สำหรับทุกสเกลาร์ λ และทุกเวกเตอร์ u
3. $\|u + v\| \leq \|u\| + \|v\|$ สำหรับทุก $u, v \in V$

บทนิยาม 2. ปริภูมิที่มีค่าประจำ (Norm space) คือปริภูมิเวกเตอร์เชิงเส้น V ซึ่งมีค่าประจำ $\|\cdot\|$

ตัวอย่าง 2.1.1. ถ้า $V = \mathbb{R}^n$ ได้ว่า V เป็นปริภูมิเวกเตอร์เชิงเส้น ซึ่งค่าประจำ $\|\cdot\|$ สามารถนิยามได้โดย

$$\|\mathbf{x}\| = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2} \quad (2.1.1)$$

และค่าประจำนี้เรียกว่า ค่าประจำแบบยุคลิด (Euclidean norm)

หมายเหตุ: $\|\cdot\|$ สัญลักษณ์นี้ต่อไปในเอกสารนี้จะใช้แทนค่าประจำ

2.2 แคลคูลัสของการแปรผันเบื้องต้น

ในหัวข้ออย่างนี้ เราจะกล่าวถึงปัญหาการหาค่าเหมาะสมสมที่สุดประเภทหนึ่งสำหรับหาฟังก์ชันที่เหมาะสมสม (แทนที่จะเป็นการหาค่าเหมาะสมสมของตัวแปรดังเช่นปัญหาการหาค่าเหมาะสมสมที่สุดทั่วไป) ซึ่งทำให้ปริมาณที่กำหนด (มักนำเสนอปริมาณนี้ในรูปของอินทิกรัลจำกัดเขต) มีภาวะหยุดนิ่ง (stationary) เนื่องจากค่าของฟังก์ชันเปลี่ยนแปลงตามตัวแปรอิสระ ปัญหาการหาค่าเหมาะสมสมที่สุดประเภทนี้จึงถูกแก้ด้วยแคลคูลัสของการแปรผัน ทั้งนี้แคลคูลัสของการแปรผันสามารถนำมาใช้หาค่าขีดสุดของปริมาณที่เขียนในรูปอินทิกรัลจำกัดเขตที่ประกอบด้วยฟังก์ชันไม่ทราบค่าและ/หรืออนุพันธ์ของฟังก์ชันไม่ทราบค่า

พิจารณาปัญหาค่าต่ำที่สุดต่อไปนี้

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

โดยที่ $\mathcal{J} : \mathcal{U} \rightarrow \mathbb{R}$ เป็นฟังก์ชันที่ส่งจากเซตของฟังก์ชันไปยังจำนวนจริง (เราจะเรียก \mathcal{J} ว่า ฟังก์ชันนัล (functional)) \mathcal{U} เป็นปริภูมิของคำตอบซึ่งประกอบด้วยฟังก์ชันที่สามารถทำให้ \mathcal{J} ต่ำที่สุด และ \mathcal{V} เป็นปริภูมิทดสอบซึ่งกำหนดโดย

$$\mathcal{V} = \{v | v = u - \hat{u} \text{ และ } u, \hat{u} \in \mathcal{U}\} \quad (2.2.2)$$

บทนิยาม 3. (ย่านไกล์เดียง) ให้ \mathcal{U} เป็นปริภูมิคำตอบ $\hat{u} \in \mathcal{U}$ และ $\epsilon > 0$ แล้วย่านไกล์เดียงของ \hat{u} เขียนแทนด้วย \mathcal{B}_ϵ นิยามโดย

$$\mathcal{B}_\epsilon = \{u \in \mathcal{U} | \|u - \hat{u}\| < \epsilon\}$$

บทนิยาม 4. (โอลคอมบินีไมเชอร์) ให้ \mathcal{U} เป็นปริภูมิคำตอบ และ $\mathcal{J} : \mathcal{U} \rightarrow \mathbb{R}$ เป็นฟังก์ชันนัล จะเรียก $\hat{u} \in \mathcal{U}$ ว่า โอลคอมบินีไมเชอร์ของ \mathcal{J} ถ้าสำหรับทุก $\epsilon > 0$ จะมี $\delta > 0$ ซึ่งทำให้ $\mathcal{J}(\hat{u}) \leq \mathcal{J}(u)$ สำหรับทุกๆ $u \in \mathcal{B}_\epsilon(\hat{u})$

บทนิยาม 5. (*Gateaux-differentiable*) ให้ \mathcal{U} เป็นปริภูมิคำตอบ \mathcal{V} เป็นปริภูมิทดสอบ และ $\mathcal{J} : \mathcal{U} \rightarrow \mathbb{R}$ เป็นฟังก์ชันนัล แล้ว เรากล่าวว่า \mathcal{J} หาอนุพันธ์แบบก้าโตว์ได้สำหรับทุก $u \in \mathcal{U}$ อยู่ในทิศทางของ $v \in \mathcal{V}$ เมื่อ

1. มีจำนวน $\hat{\epsilon} > 0$ ซึ่งทำให้ $u_\epsilon = u + \epsilon v \in \mathcal{U}$ สำหรับทุก $|\epsilon| \leq \hat{\epsilon}$
2. ฟังก์ชัน $J(\epsilon) = \mathcal{J}(u_\epsilon)$ หาอนุพันธ์ได้ที่ $\epsilon = 0$

อนุพันธ์ก้าวต่ออันดับหนึ่งหรือการแปรผันอันดับแรก (first variation) ของ \mathcal{J} สำหรับ u ในทิศทางของ v กำหนดโดย

$$\delta \mathcal{J}(u; v) = J'(0) = \frac{d\mathcal{J}(u + \epsilon v)}{d\epsilon} \Big|_{\epsilon=0} = \lim_{\epsilon \rightarrow 0} \frac{\mathcal{J}(u + \epsilon v) - \mathcal{J}(u)}{\epsilon}$$

บทนิยาม 6. (จุดคงตัว) ให้ \mathcal{U} เป็นปริภูมิคำตอบ \mathcal{V} เป็นปริภูมิทดสอบ และ $\mathcal{J} : \mathcal{U} \rightarrow \mathbb{R}$ เป็นฟังก์ชันลัลสมมติให้ \mathcal{J} หาอนุพันธ์แบบก้าวต่อได้ที่บาง $\hat{u} \in \mathcal{U}$ สำหรับทุกฟังก์ชันทดสอบ $v \in \mathcal{V}$ และ \hat{u} จะเรียกว่าจุดคงตัวของ \mathcal{J} ก็ต่อเมื่อ $\delta \mathcal{J}(\hat{u}; v) = 0$ สำหรับทุก $v \in \mathcal{V}$

ทฤษฎีบท 2.1. ให้ \mathcal{U} เป็นปริภูมิคำตอบ, $\hat{u} \in \mathcal{U}$, $\mathcal{J} : \mathcal{U} \rightarrow \mathbb{R}$ เป็นฟังก์ชันลัล และ \mathcal{V} เป็นปริภูมิทดสอบ สมมติว่า \mathcal{J} หาอนุพันธ์แบบก้าวต่อได้สำหรับ \hat{u} ในทุกๆ ทิศทางที่ $v \in \mathcal{V}$ ดังนั้น

ถ้า \hat{u} เป็นผลลัพธ์ไม่ซ้ำของ \mathcal{J} และ \hat{u} เป็นจุดคงตัวของ \mathcal{J}

ด้วยทฤษฎีบทนี้เราสามารถสำรวจเส้นทางสำหรับจุดคงตัวของฟังก์ชันลัล \mathcal{J} ซึ่งนิยามโดย

$$\mathcal{J}(u) = \int_{\Omega} F[\mathbf{x}, u(\mathbf{x}), \nabla u(\mathbf{x})] dx \quad (2.2.3)$$

โดยที่ $\Omega \subset \mathbb{R}^d (d > 1)$ เป็นเซตเปิดที่มีขอบเขตและ F เป็นฟังก์ชันลัลที่ต่อเนื่องอยู่กับ $\mathbf{x} = (x_1, x_2, \dots, x_d)^\top$, $u : \mathbb{R}^d \rightarrow \mathbb{R}$, $\nabla u(x) = (\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \dots, \frac{\partial}{\partial x_d})^\top$ สมมติให้ \mathcal{J} หาอนุพันธ์แบบก้าวต่อได้ ดังนั้นเราจึงสามารถพิจารณาได้ว่า F เป็นอนุพันธ์ย่อยที่ต่อเนื่องภายใต้ตัวแปรของมัน

เราจะใช้

$$\nabla_u F = \partial F / \partial u = F_u \quad (2.2.4)$$

แทนเกรเดียนต์ของ F ภายใต้ u เพื่อสร้างความแตกต่างกับเกรเดียนต์ของ F ภายใต้ \mathbf{x} ซึ่งเป็นสัญลักษณ์ของ

$$\nabla F = (\partial F / \partial x_1, \dots, \partial F / \partial x_d)^\top \quad (2.2.5)$$

ในทำนองเดียวกันสัญลักษณ์เกรเดียนต์ของ F ภายใต้ ∇u หมายถึง

$$\nabla_{\nabla u} F = (\partial F / \partial u_{x_1}, \dots, \partial F / \partial u_{x_d})^\top \in \mathbb{R}^d \quad (2.2.6)$$

ต่อไปเราจะพิจารณาปริภูมิของฟังก์ชันที่มีเส้นทางของขอบต่อไปนี้

$$\tilde{\mathcal{U}} = \{u \in \mathcal{U} | u = c \text{ บน } \partial\Omega\} \quad (2.2.7)$$

พร้อมกับปริภูมิของฟังก์ชันทดสอบกำหนดโดย

$$\tilde{\mathcal{V}} = \{v \in \mathcal{V} | v = 0 \text{ บน } \partial\Omega\} \quad (2.2.8)$$

ทฤษฎีบทประกอบ 2.2.1. (จุดคงตัวของ \mathcal{J}) พังก์ชัน $u \in \mathcal{U}$ เป็นจุดคงตัวของพังก์ชันนัล \mathcal{J} (2.2.3) ถ้า
เงื่อนไข

$$\int_{\Omega} \left\langle \nabla_u F - \nabla \cdot \nabla_{\nabla u} F, v \right\rangle_{\mathbb{R}^d} dx = 0 \quad (2.2.9)$$

เป็นจริงสำหรับทุกๆ พังก์ชันทดสอบ $v \in \mathcal{V}$

เห็นได้ชัดว่า (2.2.9) เป็นจริงสำหรับพังก์ชันใดๆ เมื่อ $\nabla_u F - \nabla \cdot \nabla_{\nabla u} F = 0$ ดังนั้น $u \in \hat{\mathcal{U}}$ เป็นจุด
คงตัวของพังก์ชันนัล \mathcal{J} (2.2.3) เมื่อ

$$\nabla_u F - \nabla \cdot \nabla_{\nabla u} F = 0 \text{ บน } \Omega \quad (2.2.10)$$

โดยการใช้ผลของทฤษฎีบท 2.1 กับ (2.2.10) เป็นเงื่อนไขที่จำเป็นสำหรับโอลิมินไมเซอร์ของ (2.2.1) ซึ่งถ้า
 $d > 1$ จะได้ว่า (2.2.10) จะนำไปสู่สมการเชิงอนุพันธ์ย่อย (เชิงรุ้จักกันในชื่อของสมการอยเลอร์-ลากรา
น) ที่มีเงื่อนไขค่าขอบ โดยจะเรียก (2.2.1) ว่า รูปแบบการแปรผัน (variational formulation) ของปัญหาค่า
ขอบ (2.2.10) และถ้าเงื่อนไขค่าขอบนั้นถูกกำหนดไว้อย่างชัดเจนจะเรียกเงื่อนไขนี้ว่า เงื่อนไขจำเป็น (essential
condition) และในทางกลับกันหากค่าขอบไม่ถูกกำหนดไว้อย่างชัดเจนจะเรียกว่า เงื่อนไขธรรมชาติ (natural
condition)

โดยสรุปทุกคำตوب $u \in \mathcal{U}$ ของปัญหาค่าหมายตามที่สุด ดังเช่น (2.2.1) \mathcal{J} ใน (2.2.3) เป็นคำตوبของ
ปัญหาค่าขอบที่ประกอบด้วยสมการ

$$\nabla_u F - \nabla \cdot \nabla_{\nabla u} F = 0 \text{ บน } \Omega$$

และเงื่อนไขขอบธรรมชาติบน $\partial\Omega$

ตัวอย่าง 2.2.1. ให้ $d = 2$, $\Omega = [0, 1]^2$, $F = |\nabla u|$ เมื่อ $u = u(\mathbf{x})$ จะได้ว่ารูปแบบเชิงแปรผันต่อไปนี้

$$\min_u \int_{\Omega} |\nabla u| d\Omega$$

เทียบเท่ากับปัญหาค่าขอบที่กำหนดโดย

$$\begin{cases} -\nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) = 0, & \mathbf{x} \in \Omega = (1, n)^2 \\ \frac{\partial u}{\partial \mathbf{n}} = 0, & \mathbf{x} \in \partial\Omega \end{cases} \quad (2.2.11)$$

กำหนดให้

$$\mathcal{R}(u) = \int_{\Omega} |\nabla u| d\Omega = \int_{\Omega} \sqrt{u_x^2 + u_y^2} d\Omega$$

เพื่อสะดวกในการคำนวณการแปรผันอันดับหนึ่งของ \mathcal{R} กำหนดให้ $\Phi(s) = s$ จะได้

$$\frac{\delta}{\delta u} \mathcal{R}(u; v) = \left. \frac{d}{d\varepsilon} \mathcal{R}(u + \varepsilon v) \right|_{\varepsilon=0} = \left. \frac{d}{d\varepsilon} \int_{\Omega} \Phi(|\nabla(u + \varepsilon v)|) d\Omega \right|_{\varepsilon=0}$$

ดังนั้น

$$\begin{aligned} \frac{\delta}{\delta u} \mathcal{R}(u; v) &= \int_{\Omega} \left. \frac{d}{d\varepsilon} \Phi(\sqrt{(u_x + \varepsilon v_x)^2 + (u_y + \varepsilon v_y)^2}) \right|_{\varepsilon=0} d\Omega \\ &= \int_{\Omega} \left[\Phi'(\sqrt{(u_x + \varepsilon v_x)^2 + (u_y + \varepsilon v_y)^2}) \frac{(u_x + \varepsilon v_x)v_x}{\sqrt{(u_x + \varepsilon v_x)^2 + (u_y + \varepsilon v_y)^2}} \right. \\ &\quad \left. + \Phi'(\sqrt{(u_x + \varepsilon v_x)^2 + (u_y + \varepsilon v_y)^2}) \frac{(u_y + \varepsilon v_y)v_y}{\sqrt{(u_x + \varepsilon v_x)^2 + (u_y + \varepsilon v_y)^2}} \right]_{\varepsilon=0} d\Omega \\ &= \int_{\Omega} \Phi'(|\nabla u|) \left(\frac{u_x v_x}{|\nabla u|} + \frac{u_y v_y}{|\nabla u|} \right) d\Omega \\ &= \int_{\Omega} \frac{\Phi'(|\nabla u|)}{|\nabla u|} (\nabla u \cdot \nabla v) d\Omega \\ &= \int_{\Omega} \Phi'(|\nabla u|) \frac{\nabla u}{|\nabla u|} \cdot \nabla v d\Omega \end{aligned}$$

โดยเอกสารลักษณ์อันดับหนึ่งของกรีน จะได้ว่า

$$\begin{aligned} \frac{\delta}{\delta u} \mathcal{R}(u; v) &= - \int_{\Omega} v \nabla \cdot \left(\Phi'(|\nabla u|) \frac{\nabla u}{|\nabla u|} \right) d\Omega + \int_{\partial\Omega} v \left(\Phi'(|\nabla u|) \frac{\nabla u}{|\nabla u|} \cdot \mathbf{n} \right) dS \\ &= - \int_{\Omega} v \nabla \cdot \left(\Phi'(|\nabla u|) \frac{\nabla u}{|\nabla u|} \right) d\Omega + \int_{\partial\Omega} v \left(\Phi'(|\nabla u|) \frac{\partial u}{\partial \mathbf{n}} \right) dS \end{aligned}$$

เพราะนั้น

$$\frac{\delta}{\delta u} \mathcal{R}(u; v) = - \int_{\Omega} v \nabla \cdot \left(\Phi'(|\nabla u|) \frac{\nabla u}{|\nabla u|} \right) d\Omega + \int_{\partial\Omega} v \left(\Phi'(|\nabla u|) \frac{\partial u}{\partial \mathbf{n}} \right) dS$$

เมื่อ \mathbf{n} แทนเวกเตอร์หนึ่งที่ตั้งฉากกับขอบของภาพในทิศทางซึ่อออก

ดังนั้นเมื่อกำหนดให้ $\frac{\partial u}{\partial \mathbf{n}} = 0$ จะได้สมการอย่างที่สมมัยกับปัญหานี้คือ

$$\begin{aligned} -\nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) &= 0 \text{ บน } \Omega \\ \frac{\partial u}{\partial \mathbf{n}} &= 0 \text{ บน } \partial\Omega \end{aligned}$$

2.3 พังก์ชันของการแปรผันที่มีขอบเขต

ให้ Ω เป็นเซตเปิดที่มีขอบเขตของ \mathbb{R}^d และให้ $u \in L^1(\Omega)$ เราอนิยามการแปรผันรวม (total variation) ของ u เป็น

$$\int_{\Omega} |Du| = \sup \left\{ \int_{\Omega} u \nabla \cdot \varphi d\mathbf{x} \mid \varphi = (\varphi_1, \varphi_2, \dots, \varphi_d) \in C_0^1(\Omega, \mathbb{R}^d)^d \text{ และ } \|\varphi_i\|_{L^\infty} \leq 1 \text{ สำหรับ } i = 1, \dots, d \right\} \quad (2.3.1)$$

เมื่อ $\nabla \cdot \varphi = \sum_{i=1}^d \frac{\partial \varphi_i}{\partial x_i}(\mathbf{x})$, $d\mathbf{x}$ เป็น (Lebesgue measure) และ $C_0^1(\Omega, \mathbb{R}^d)$ คือปริภูมิของพังก์ชันต่อเนื่องที่หาอนุพันธ์ได้และกระซับใน Ω

ถ้า $u \in C_0^1(\Omega, \mathbb{R}^d)$ จะได้

$$\int_{\Omega} u \nabla \cdot \varphi dx = - \int_{\Omega} \sum_{i=1}^d \frac{\partial u}{\partial x_i} \varphi_i d\Omega \quad (2.3.2)$$

สำหรับทุก $\varphi \in C_0^1(\Omega, \mathbb{R}^d)^d$ ดังนั้น

$$\int_{\Omega} |Du| = \int_{\Omega} |\nabla u| dx \quad (2.3.3)$$

พังก์ชัน $u \in L^1(\Omega)$ เรียกว่าพังก์ชันที่มีการแปรผันแบบมีขอบเขตใน Ω และจะใช้สัญลักษณ์ $BV(\Omega)$ แทนปริภูมิของทุกๆ พังก์ชันใน $L^1(\Omega)$ ที่มีการแปรผันแบบมีขอบเขต

ตัวอย่าง 2.3.1. พังก์ชัน $f1, f2$ และ $f3$ ซึ่งกำหนดโดย

$$f1(x) = \sin(x), \quad (2.3.4)$$

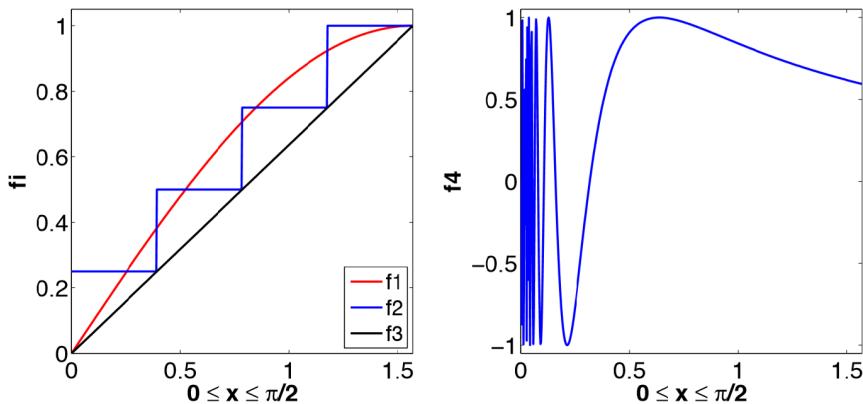
$$f2(x) = \begin{cases} 1/4, & x \in [0, \pi/8] \\ 1/2, & x \in [\pi/8, \pi/4] \\ 3/4, & x \in [\pi/4, 3\pi/8] \\ 1, & x \in [3\pi/8, \pi/2] \end{cases} \quad (2.3.5)$$

$$f3(x) = \frac{2x}{\pi}, \quad (2.3.6)$$

เป็นฟังก์ชันในบริภูมิ $BV(\Omega)$ ถ้า $\Omega = [0, \Pi/2]$ และทั้ง 3 ฟังก์ชันด้านบนการแปรผันรวมเดียวกันคือ 1 สำหรับฟังก์ชัน f_4 กำหนดโดย

$$f_4(x) = \begin{cases} 0, & x = 0 \\ \sin(1/x), & x \in (0, a) \text{ และ } a > 0 \end{cases} \quad (2.3.7)$$

มีการแปรผันรวมอนันต์ (*infinite total variation*) และไม่มีอยู่ในบริภูมิ $BV(\Omega)$ ซึ่ง $\Omega = [0, a]$ สำหรับทุก $a > 0$



รูปที่ 2.3.1: (ซ้าย) ฟังก์ชันที่มีขอบเขตกับการแปรผันรวมที่มีค่าเดียวกัน (ขวา) ฟังก์ชันที่มีขอบเขตกับการแปรผันรวมอนันต์

2.4 ปัญหาแบบอิลล์โพสและวิธีการเริกกวิเคราะห์เลเซชัน

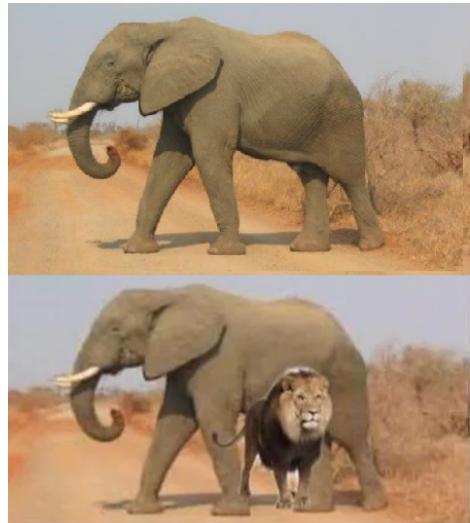
บทนิยาม 7. (ปัญหาแบบเวลล์โพส) เราจะเรียกปัญหาที่กำหนดว่าเป็นปัญหาแบบเวลล์โพส เมื่อเงื่อนไขทุกข้อต่อไปนี้เป็นจริง

1. ปัญหามีคำตอบ
2. ปัญหามีเพียงคำตอบไม่หนึ่งเดียว
3. คำตอบทองปัญหาขึ้นอยู่กับความต้องเนื่องของข้อมูล

หากข้อใดข้อหนึ่งไม่จริง จะเรียกปัญหาที่กำหนดว่า ปัญหาอิลล์โพส

ตัวอย่าง 2.4.1. ปัญหาการหาค่า x และ y ที่ทำให้ $x + y = 5$ เป็นปัญหาที่มีหลายคำตอบ ดังนั้นปัญหานี้เป็นปัญหาแบบอิลลิปส์

ปัญหาการต่อเติมภาพเป็นปัญหาแบบอิลลิปส์เนื่องจากคำตอบในบริเวณโดเมนต่อเติมไม่ได้มีเพียงคำตอบเดียว ตัวอย่างเช่น รูปที่ 2.4.1 ภาพซึ่งที่เสียหาย¹ ในบริเวณส่วนสีแดงซึ่งภาพเกิดความเสียหายขึ้น อาจจะมีคำตอบเป็นขาของช้าง หรือมีคำตอบเป็นสิงโตตั้งในภาพก็ได้



รูปที่ 2.4.1: (ซ้าย) ปัญหาการต่อเติมภาพ (ขวา) ตัวอย่างคำตอบที่เป็นไปได้

2.4.1 วิธีการเร็กกิวลาร์ไรซ์เซชัน

วิธีเร็กกิวลาร์ไรซ์เซชัน (Regularization) เป็นวิธีการทางคณิตศาสตร์ที่นิยมอย่างแพร่หลายในการแก้ปัญหาแบบอิลลิปส์คิดค้นโดย Tikhonov [8]

โดยวิธีการนี้จะทำการแก้ปัญหาแบบอิลลิปส์โดยการเพิ่มเงื่อนไขเข้าไปในปัญหาเพื่อให้คำตอบที่ได้เป็นเขตคำตอบที่อยู่ในเขตของคำตอบที่เป็นไปได้

ตัวอย่าง 2.4.2. กำหนดให้ $x + y = 5$ จงหา x และ y ที่ทำให้ล้มการเป็นจริง จะพบว่าปัญหานี้เป็นปัญหาแบบอิลลิปส์เนื่องจากมีคำตอบได้หลายคำตอบ

หลังจากใช้แนวคิดของเร็กกิวลาร์ไรซ์เซชันโดยการเพิ่มเงื่อนไขว่า $\sqrt{x^2 + y^2}$ มีค่าน้อยที่สุด จึงได้ว่า $x = 2.5$ และ $y = 2.5$ เป็นคำตอบของปัญหาที่ลอดคล้องกับเงื่อนไขทางคณิตศาสตร์ที่ต้องการ

¹ภาพจาก <https://9gag.com/gag/aer4VwB> สืบค้นเมื่อ 10 มีนาคม 2562

2.5 วิธีการไฟนิตดิฟเฟอเรนซ์เบื้องต้น

วิธีการไฟนิตดิฟเฟอเรนซ์ (finite difference method) เป็นวิธีการเชิงตัวเลขที่พัฒนาขึ้นเพื่อแก้ไขปัญหาค่าขอบ ขั้นตอนของวิธีการไฟนิตดิฟเฟอเรนซ์สำหรับการแก้ปัญหาค่าขอบประกอบด้วยสามขั้นตอนสำคัญดังนี้

1. ดิสcretize (discretize) โดยแบ่งพื้นที่ของผลเฉลย (solution domain) ออกเป็นช่องตาราง (mesh) ของจุดกริด (grid point) ที่ต้องการหาผลเฉลยเชิงตัวเลข
2. ประมาณอนุพันธ์ที่ปรากฏในปัญหาค่าขอบด้วยการประมาณแบบไฟนิตดิฟเฟอเรนซ์ (finite difference approximation) ในขั้นตอนนี้ การประมาณดังกล่าวจะนำไปสู่ระบบสมการเชิงเส้น หรือระบบสมการไม่เป็นเชิงเส้นขนาดใหญ่ที่สมนัยกับปัญหาค่าขอบตั้งต้น
3. แก้ระบบสมการเชิงเส้นหรือระบบสมการไม่เป็นเชิงเส้นขนาดใหญ่ที่เกิดขึ้นในขั้นตอนที่ 2 เพื่อกำหนดผลเฉลยเชิงตัวเลข

กำหนดให้ $u(x)$ แทนฟังก์ชันค่าจริงและเป็นฟังก์ชันราบรื่น (smooth function) นั่นคือ u สามารถหาอนุพันธ์ได้หลายครั้ง โดยแต่ละครั้ง อนุพันธ์ที่หาได้เป็นฟังก์ชันที่ถูกนิยามอย่างดี (well-defined) และมีขอบเขตหนึ่งซึ่งที่มีจุดที่สนใจ \bar{x}

ในการประมาณ $u'(\bar{x})$ โดยใช้ค่าของ u ที่เกิดจากจุดที่อยู่ในบริเวณใกล้เคียงกับ \bar{x} สามารถใช้สูตรการประมาณแบบไฟนิตดิฟเฟอเรนซ์ที่ถูกกำหนดได้ดังต่อไปนี้

1. สูตรฟอร์เวิร์ดดิฟเฟอเรนซ์ (forward-difference formula)

$$D_+ u(\bar{x}) = \frac{u(\bar{x} + h) - u(\bar{x})}{h}$$

2. สูตรแบ็คเวย์ดดิฟเฟอเรนซ์ (backward-difference formula)

$$D_- u(\bar{x}) = \frac{u(\bar{x}) - u(\bar{x} - h)}{h}$$

3. สูตรเช็นทรัลดิฟเฟอเรนซ์ (central-difference formula)

$$D_0 u(\bar{x}) = \frac{u(\bar{x} + h) - u(\bar{x} - h)}{2h}$$

โดยที่ h เป็นจำนวนจริงที่มีค่าน้อยๆ ซึ่ง $h > 0$

ตัวอย่าง 2.5.1. พิจารณาปัญหาค่าขอบดังต่อไปนี้

$$\begin{cases} -\Delta u = f & x \in \Omega = (0, 1)^2 \\ u = 0 & x \in \partial\Omega \end{cases} \quad (2.5.1)$$

กำหนดให้ m_x, m_y เป็นจำนวนนับแทนจำนวนจุดกริดในทาง x และทาง y ตามลำดับ และ

$$\Omega_h = \{\mathbf{x} \in \Omega | \mathbf{x} = (x_i, y_j)^\top = (ih_x, jh_y), 1 \leq i \leq m_x, 1 \leq j \leq m_y\}$$

แทนโดยเมนของผลเฉลยที่ถูกติดต่อตัวที่ $(m_x + 1) \times (m_y + 1)$ เขลล์ (cell) แต่ละเซลล์มีขนาด $h_x \times h_y$ โดยที่

$$h = (h_x, h_y)^\top = \left(\frac{1}{m_x + 1}, \frac{1}{m_y + 1} \right)^\top$$

กำหนดให้ $u(x, y)$ แทนค่าตอบของปัญหา (2.5.1), $(f)_{i,j}$ แทน $f(x_i, y_j)$ และ $(u)_{i,j}$ แทนฟังก์ชันกริด (grid function) ซึ่งล้มพั้นธ์กับค่าตอบแบบนtru ที่จุดกริด $(x_i, y_j)^\top$ เราจะทำการแทนที่อนพั้นธ์อยอันดับสองใน (2.5.1) ด้วยสูตรเซนทรัล diff เฟอร์เรนซ์ได้ดังนี้

$$-\left(\frac{1}{h_x^2} ((u)_{i-1,j} - 2(u)_{i,j} + (u)_{i+1,j}) + \frac{1}{h_y^2} ((u)_{i,j-1} - 2(u)_{i,j} + (u)_{i,j+1}) \right) = (f)_{i,j} \quad (2.5.2)$$

ภายใต้เงื่อนไขขอบ $(u)_{i,j} = 0$ เมื่อ $i = 0$ หรือ m_x , $j = 0$ หรือ m_y

2.6 วิธีการแก้สีใชเดลสำหรับระบบสมการเชิงเส้น

ในหัวข้อนี้จะแนะนำถึงวิธีการแก้สีใชเดล (Gauss-seidel) เพื่อแก้ปัญหาระบบสมการเชิงเส้น

$$Ax = b \quad (2.6.1)$$

เมื่อ $x \in \mathbb{R}^n$ และ A เป็นเมตริกซ์ขนาด $n \times n$ โดยการทำซ้ำนี้จะเริ่มจากค่าประมาณเริ่มต้น (intial approximation) $x^{(0)}$ และทำการสร้างลำดับ $\{x^{(k)}\}_{k=1}^{\infty}$ จากความสัมพันธ์

$$x^{(k)} = Tx^{(k-1)} + c \quad (2.6.2)$$

โดยที่ เมทริกซ์ความสัมพันธ์ T และเวกเตอร์ c มาจากการแยก $A = D - L - U$ เมื่อ D เป็นเมทริกซ์ที่ແບงมุ่งของ A , $-L$ เป็นสามเหลี่ยมส่วนล่างของ A และ $-U$ เป็นสามเหลี่ยมส่วนบนของ A จึงได้ว่า $T = (D - L)^{-1}U$ และ $c = (D - L)^{-1}b$

2.7 การวัดคุณภาพของภาพที่ผ่านกระบวนการต่อเติม

การประเมินคุณภาพของขั้นตอนวิธีการต่อเติมภาพ จะใช้ค่า Peak signal-to-noise ratio (PSNR) [6] และ Structral Similarity index (SSIM) [7]

2.7.1 Peak signal-to-noise ratio (PSNR)

PSNR นิยามโดย

$$\text{PSNR} = 10 \cdot \log_{10}\left(\frac{1}{\sqrt{\frac{1}{N} \sum(u - \bar{u})^2}}\right) \quad (2.7.1)$$

เมื่อ N แทนจำนวนพิกเซลในภาพ, u แทนภาพต้นฉบับและ \tilde{u} แทนภาพที่ได้จากการซ่อมแซมโดยวิธีเชิงตัวเลข PSNR ใช้สำหรับวัดคุณภาพของภาพ มีหน่วยเป็น เดซิเบล (dB) และมีค่าอยู่ระหว่าง 0 ถึง ∞ โดยค่า PSNR ที่ได้ยิ่งเข้าใกล้สูงสุดเท่าไหร่ แปลว่าภาพที่ได้จากการซ่อมแซมมีคุณภาพดี

2.7.2 Structral Similarity index (SSIM)

SSIM นิยามโดย

$$\text{SSIM}(u, \tilde{u}) = \frac{(2\mu_u\mu_{\tilde{u}} + 0.0001)(2\sigma_{u\tilde{u}} + 0.0009)}{(\mu_u^2 + \mu_{\tilde{u}}^2 + 0.0001)(\sigma_u^2 + \sigma_{\tilde{u}}^2 + 0.0009)}$$

u แทนภาพต้นฉบับ, \tilde{u} แทนภาพที่ได้จากการซ่อมแซมโดยวิธีเชิงตัวเลข, μ_u คือค่าเฉลี่ยของ u , $\mu_{\tilde{u}}$ คือค่าเฉลี่ยของ \tilde{u} , σ_u คือความแปรปรวนของ u และ $\sigma_{\tilde{u}}$ คือความแปรปรวนของ \tilde{u}

SSIM ใช้สำหรับวัดคุณภาพของเค้าโครงในภาพผลลัพธ์ว่ามีเค้าโครงคล้ายกับภาพต้นฉบับมากเพียงใด โดยค่า SSIM จะมีค่าอยู่ระหว่าง 0 ถึง 1 หากค่าที่ได้ยิ่งเข้าใกล้สูง 1 หมายถึงภาพที่ได้รับการซ่อมแซมมีเค้าโครงใกล้เคียงกับภาพต้นฉบับ

บทที่ 3

ขั้นตอนวิธีเชิงตัวเลขชนิดใหม่

3.1 การนำเสนอภาพ

ก่อนจะนำเสนอขั้นตอนวิธีเชิงตัวเลขชนิดใหม่ จะขอกล่าวถึงการนำเสนอภาพเชิงคณิตศาสตร์ ดังนี้

3.1.1 การนำเสนอภาพเขตเทา

กำหนดให้ $I : \Omega \subset \mathbb{R}^2 \rightarrow V \subset [0, \infty)$ แทนภาพเขตเทา (grayscale image) เป็นฟังก์ชันต่อเนื่อง โดยที่ $\mathbf{x} = (x, y) \in \Omega$ แทนพิกัดทางกายภาพ (physical position) ของภาพ $I(\mathbf{x}) \in V$ แทนระดับความเข้มของภาพ (image intensity) ที่ \mathbf{x} และ Ω แทนโดเมนของภาพซึ่งเป็นรูปร่างสี่เหลี่ยม ซึ่งในที่นี้สามารถสมมติได้โดยไม่เสียหลักการสำคัญว่า $\Omega = [1, n]^2$ และ $V = [0, 1]$ เมื่อ $n > 1$ เป็นจำนวนเต็มบวก

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	0.6	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: ตัวอย่างภาพเขตเทาที่แสดงระดับความเข้มของภาพ

จากภาพ 3.1.1 สังเกตว่าที่ค่าความเข้มของภาพเข้าใกล้ 0 จะให้สีเป็นลักษณะสีดำ ดังเช่นบริเวณที่พิกัด

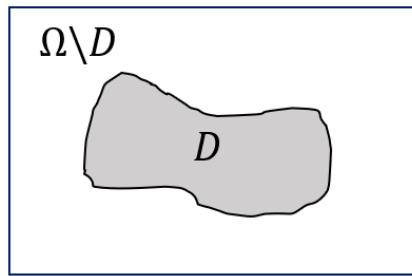
ทางกายภาพเป็น $(4,8)$ และเมื่อค่าความเข้มของสีเข้าใกล้ 1 จะให้สีที่มีลักษณะเป็นสีขาว ดังเช่นบริเวณที่มีพิกัดทางกายภาพเป็น $(7,1)$

3.1.2 การนำเสนօภาพลี

กำหนดให้ $\mathbf{I} = (I_1, I_2, I_3)^\top : \Omega \rightarrow V^3$ แทนภาพสีในระบบ RGB เมื่อ $I_1, I_2, I_3 : \Omega \rightarrow V$ แทนส่วนประกอบสีแดง สีเขียว และสีน้ำเงินตามลำดับ

3.2 ตัวแบบเชิงแปรผันสำหรับต่อเติมภาพขาดเทา

กำหนดให้ $u : \Omega \rightarrow V$, $z : \Omega \rightarrow V$ แทนภาพที่ได้รับการต่อเติมและภาพที่ต้องการต่อเติม ตามลำดับ



รูปที่ 3.2.1: D แทนโดเมนต่อเติมของ z

การต่อเติมภาพขาดเทาคือการหาค่าหมายรวมที่สุดของค่าความเข้มของภาพในบริเวณโดเมนต่อเติม D โดยใช้ข้อมูลที่มีอยู่ใน $\Omega \setminus D$

Chan และ Shen [1] ได้นำเสนอตัวแบบเชิงการแปรผัน (variational model) ที่ใช้เร็กวิวาร์เรชเชชันแบบการแปรผันรวม (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

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

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

3.3 ตัวแบบเชิงแปรผันสำหรับการต่อเติมภาพสี

กำหนดให้

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

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

ในทำนองเดียวกันกับตัวแบบการต่อเติมภาพเดสีเทาที่ใช้การแปรผันรวม ตัวแบบการต่อเติมภาพสีที่ใช้การแปรผันรวมสามารถนำเสนอได้ดังนี้

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

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 การเดินเวลาแบบชัดแจ้ง

คณะวิจัย [2] ได้แนะนำวิธีการเชิงตัวเลขสำหรับการกำจัดสัญญาณรบกวนโดยใช้วิธีการเดินเวลาแบบชัดแจ้ง เพื่อแก้สมการอยเลอร์-ลากරองที่สมนัยกับปัญหาการแปรผัน (3.2.1) ต่อไปนี้

$$\begin{cases} -\nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) + \lambda(u - z) = 0, & \mathbf{x} \in \Omega = (1, n)^2 \\ \frac{\partial u}{\partial \mathbf{n}} = 0, & x \in \partial\Omega \end{cases} \quad (3.4.1)$$

เมื่อ n แทนเวลาเตอร์หน่วยที่ตั้งจากกับของของภาพ

วิธีการนี้เราจะเริ่มจากการแนะนำตัวแปรเวลาสังเคราะห์ (time artificial variable) จากนั้นหาคำตอบแบบสภาวะคงตัว (steady-state solution) ในขณะที่ $t \rightarrow \infty$ ของสมการเชิงอนุพันธ์อยู่ไม่เป็นเชิงเส้นที่ขึ้นอยู่กับเวลาต่อไปนี้

$$u(\mathbf{x}, t_{k+1}) = u(\mathbf{x}, t_k) + \tau \left(\nabla \cdot \left(\frac{\nabla u(\mathbf{x}, t_k)}{|\nabla u(\mathbf{x}, t_k)|} \right) + \lambda(\mathbf{x})(u(\mathbf{x}, t_k) - z(\mathbf{x})) \right), \quad u(\mathbf{x}, t_0) = z \quad (3.4.2)$$

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

หลังจากใช้การประมาณแบบไฟโนติดฟเฟอร์เรนจ์ดังที่กล่าวในบทที่ 2 จะได้รูปแบบการทำซ้ำเป็น

$$(u^{[k+1]})_{i,j} = (u^{[k]})_{i,j} + \tau \left(\mathcal{K}(u^{[k]})_{i,j} - (\lambda)_{i,j} ((u^{[k]})_{i,j} + (z)_{i,j}) \right) \quad (3.4.3)$$

โดยที่

$$\mathcal{K}(u)_{i,j} = [\partial_x^- (D(u)_{i,j} \partial_x^+(u)_{i,j}) + \partial_y^- (D(u)_{i,j} \partial_y^+(u)_{i,j})]$$

$$= ((\Sigma)_{i,j}(u)_{i,j} - (\bar{\Sigma})_{i,j}(u)_{i,j})$$

$$(\Sigma)_{i,j}(u)_{i,j} = (D_1(u)_{i,j} + D_2(u)_{i,j} + 2D_3(u)_{i,j}) (u)_{i,j}$$

$$(\bar{\Sigma})_{i,j}(u)_{i,j} = D_1(u)_{i-1,j}(u)_{i-1,j} + D_2(u)_{i,j-1}(u)_{i,j-1}$$

$$+ D_3(u)_{i,j} ((u)_{i+1,j} + (u)_{i,j+1})$$

$$D(u)_{i,j} = \frac{1}{\sqrt{(\partial_x^+(u)_{i,j})^2 + (\partial_y^+(u)_{i,j})^2 + \beta}}, \quad 0 < \beta \ll 1$$

$$D_1(u)_{i,j} = D(u)_{i-1,j}, \quad D_2(u)_{i,j} = D(u)_{i,j-1}, \quad D_3(u)_{i,j} = D(u)_{i,j}$$

$$\partial_x^\pm(u)_{i,j} = \pm ((u)_{i\pm 1,j} - (u)_{i,j})$$

$$\partial_y^\pm(u)_{i,j} = \pm ((u)_{i,j\pm 1} - (u)_{i,j})$$

Algorithm 1 นำเสนอบริการเดินเวลาแบบชั้ดแจ้งสำหรับการต่อเติมภาพแนวเทาที่ใช้การประผันรวม

Algorithm 1: วิธีการเดินเวลาแบบชัดเจ็บสำหรับการต่อเติมภาพเขตที่ใช้การแปรผันรวม

Input:

- u คือภาพที่ได้รับการต่อเติม
- z คือภาพที่ต้องการต่อเติม
- λ คือพารามิเตอร์เริกกิวลาร์เรซเซชันในสมการ (3.2.2)
- β เป็นจำนวนจริงบวก
- τ เป็นจำนวนจริงบวก
- N เป็นจำนวนเต็มบวก
- ε เป็นจำนวนจริงบวก

Output: ภาพที่ผ่านการต่อเติมแล้ว

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

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

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

$$\begin{aligned} u^{old} &= u \\ u &= u + \tau \left(\nabla \cdot \left(\frac{\nabla u}{\sqrt{u_x^2 + u_y^2 + \beta}} \right) + \lambda(u - z) \right) \\ err &= \frac{\|u - u^{old}\|}{\|u\|} \end{aligned}$$

$i = i + 1$

end

3.4.2 วิธีการทำซ้ำแบบจุดตรีง

คณะวิจัย [3] ได้แนะนำวิธีการเชิงตัวเลขสำหรับการกำจัดสัญญาณรบกวนโดยใช้วิธีการทำซ้ำแบบจุดตรีง กับสมการอยเลอร์-ลากรองที่สมนัยกับปัญหาเชิงแปรผัน (3.2.1) ซึ่งสามารถกล่าวถึงได้โดยสังเขปดังนี้ เริ่มจากแนะนำดัชนีการทำซ้ำแบบจุดตรีง $\nu = 0, 1, 2, \dots$ และนิยามรูปแบบการทำซ้ำโดย

$$-\nabla \cdot \left(\frac{\nabla u^{[\nu+1]}}{|\nabla u|^{[v]}} \right) + \lambda(u^{[\nu+1]} - z) = 0, \quad u^{[0]} = z \quad (3.4.4)$$

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

หลักเลี้ยงปัญหาเชิงตัวเลขจะเกิดขึ้นใน (3.4.2) และ (3.4.4) เราจะใช้

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

Algorithm 2 สรุปวิธีการทำขั้นตอนแบบจุดตรึงสำหรับต่อเติมภาพเขตเทาที่ใช้การแปรผันรวม

Algorithm 2: วิธีการทำขั้นตอนแบบจุดตรึงสำหรับการต่อเติมภาพเขตเทาที่ใช้การแปรผันรวม

Input:

- u คือภาพที่ได้รับการต่อเติม
- z คือภาพที่ต้องการต่อเติม
- λ คือพารามิเตอร์เร็กกิวลาร์เรซซ์เซชันในสมการ (3.2.2)
- β เป็นจำนวนจริงบวก

N_{FP} เป็นจำนวนเต็มบวกสำหรับกำหนดจำนวนรอบการทำขั้นตอน

N_{GS} เป็นจำนวนเต็มบวกสำหรับกำหนดจำนวนรอบการทำขั้นตอน Gauss-Seidel

ε เป็นจำนวนจริงบวก

Output: ภาพที่ผ่านการต่อเติมแล้ว

```

 $u \leftarrow FixedPoint(u, z, \lambda, \beta, N_{FP}, N_{GS}, \varepsilon)$ 
initialize  $i = 0; u = z; err = 1$ 

while  $i < N_{FP}$  and  $err > \varepsilon$  do
     $u^{old} = u$ 
     $D = [D(u)]_{i,j} = \frac{1}{\sqrt{(u_x^2)_{i,j} + (u_y^2)_{i,j} + \beta}}, 1 \leq i \leq n_x, 1 \leq j \leq n_y$ 
     $u = GaussSeidel(u, z, D, \lambda, \beta, N_{GS})$ 
     $err = \frac{\|u - u^{old}\|}{\|u\|}$ 
     $i = i + 1$ 
end

```

Algorithm 3: การทำซ้ำเกาส์-ไซเดล ภายใต้สำหรับวิธีการทำซ้ำจุดตรึง

Input:

- u คือภาพที่ได้รับการต่อเติม
- z คือภาพที่ต้องการต่อเติม
- D คือสัมประสิทธิ์การแพร่กระจายไม่เป็นเชิงเส้น
- λ คือพารามิเตอร์เริกกิวลาเรซเซชันในสมการ (3.2.2)
- β เป็นจำนวนจริงบวก
- N_{GS} เป็นจำนวนเต็มบวก

Output: รูปภาพที่ผ่านการทำเกาส์-ไซเดลแล้ว $u \leftarrow GaussSeidel(u, z, D, \lambda, \beta, N_{GS})$ **initialize** $k = 0$ **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 (3.4.2) และ (3.4.4) เรายกเว้น β มีค่าน้อยลงมากขึ้นเท่าไหร่ ความแม่นยำของตัวแบบ (3.2.1) ยิ่งมีมากขึ้นเท่านั้น นอกเหนือจากนี้ เรา秧งพบอีกว่า การแก้สมการ (3.4.2) และ (3.4.4) ยิ่งมีความยุ่งยากมากขึ้นสำหรับ β ที่มีค่าน้อยๆ

เพื่อเข้าใจความยากเชิงตัวเลขนี้ คณวิจัยโดย [4] ได้แนะนำวิธีการสปริทเบรกเมนซึ่งสามารถกล่าวถึง พอกสังเขป ดังนี้

3.4.3 วิธีการสปริทเบรกเมน

เริ่มจากการแนะนำเกตอร์เสริม w พารามิเตอร์เบรกเมน (Bregman parameter) b และพารามิเตอร์ เพนลัที (panalty parameter) $\theta > 0$ และเขียน (3.2.1) ใหม่ ดังนี้

$$\min_{u,w} \{ \mathcal{J}(u, w) = \frac{1}{2} \int_{\Omega} \lambda(u - z)^2 d\Omega + \int_{\Omega} |w| d\Omega + \frac{\theta}{2} \int_{\Omega} (w - \nabla u + b)^2 d\Omega \} \quad (3.4.5)$$

สำหรับการหาค่าตอบของ (3.4.5) เราจะใช้วิธีการหาค่าต่อที่สุดแบบสลับ (alternating minimization method) โดยเริ่มจากการตรึง \mathbf{w}^{old} และ \mathbf{b}^{old} จากนั้นแก้ปัญหาอย่างสำหรับ u

$$u^{\text{New}} = \arg \min_u \{ \mathcal{J}_1(u) = \frac{1}{2} \int_{\Omega} \lambda(u - z)^2 d\Omega + \frac{\theta}{2} \int_{\Omega} (\mathbf{w}^{\text{old}} - \nabla u + \mathbf{b}^{\text{old}})^2 d\Omega \} \quad (3.4.6)$$

ต่อไปใช้ u^{New} ที่ได้จากการแก้ปัญหาอย่าง (3.4.6) เพื่อแก้ปัญหาอย่างสำหรับ \mathbf{w}

$$\mathbf{w}^{\text{New}} = \arg \min_{\mathbf{w}} \{ \mathcal{J}_2(\mathbf{w}) = \int_{\Omega} |\mathbf{w}| d\Omega + \frac{\theta}{2} \int_{\Omega} (\mathbf{w} - \nabla u^{\text{New}} + \mathbf{b}^{\text{old}})^2 d\Omega \} \quad (3.4.7)$$

สุดท้ายจึงปรับปรุงพารามิเตอร์เบรกเมนโดย

$$\mathbf{b}^{\text{New}} = \mathbf{b}^{\text{old}} + \nabla u^{\text{New}} - \mathbf{w}^{\text{New}} \quad (3.4.8)$$

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

Algorithm 4 แสดงขั้นตอนต่างๆ ของวิธีการสปริทเบรกเมน

Algorithm 4: วิธีสปริตเบรกเม่นสำหรับการต่อเติมภาพคาดเทาที่ใช้การแปรผันรวม**Input:**

- u คือภาพที่ต้องการต่อเติม/ภาพที่ได้รับการต่อเติม
- λ คือพารามิเตอร์เร็กกิวลาร์เรช์เซชัน ที่ได้ถูกง่ายในสมการ (3.2.2)
- θ คือพารามิเตอร์เพนล์ที่ซึ่งเป็นจำนวนจริงบวก
- N_{GS} เป็นจำนวนเต็มบวก
- N_{SB} เป็นจำนวนเต็มบวก
- ε เป็นจำนวนจริงบวก

Output: ภาพที่ผ่านการต่อเติมแล้ว

```

 $u \leftarrow SplitBregman(u, \lambda, \theta, N_{GS}, N_{SB}, \varepsilon)$ 
initialize  $i = 0, \mathbf{b} = \vec{0}, \mathbf{w} = \vec{0}, z = u$ 
while  $i < N_{SB}$  and  $err > \varepsilon$  do
     $u^{old} = u; w^{old} = w; b^{old} = b;$ 
     $u^{new} = \arg \min_u \{J_1(u) = \frac{1}{2} \int_{\Omega} \lambda(u-z)^2 d\Omega + \frac{\theta}{2} \int_{\Omega} (\mathbf{w}^{old} - \nabla u + \mathbf{b}^{old}) d\Omega\}$ 
     $w^{new} = \arg \min_{\mathbf{w}} \{J_2(\mathbf{w}) = \int_{\Omega} |\mathbf{w}| d\Omega + \frac{\theta}{2} \int_{\Omega} (\mathbf{w} - \nabla u^{New} + \mathbf{b}^{old}) d\Omega\}$ 
     $b^{new} = b^{old} + \nabla u^{new} - \mathbf{w}^{new}$ 
     $err = \frac{\|u^{new} - u^{old}\|}{\|u^{new}\|}$ 
     $i = i + 1$ 
end

```

โดย accolclss ของการแปรผันจะได้สมการอยเลอร์ลาการ์สำหรับ (3.4.6) เทียบกับตัวแปร u กำหนดโดย

$$\begin{cases} -\theta \Delta u + \lambda u = \lambda z - \theta \nabla \cdot (\mathbf{w} - \mathbf{b}) & x \in \Omega \\ \frac{\partial u}{\partial \mathbf{n}} = 0, & x \in \partial \Omega \end{cases} \quad (3.4.9)$$

ในการแก้สมการ (3.4.9) โดยวิธีการเชิงตัวเลขเราจะใช้วิธีการไฟน์ต์ดิฟเฟอเรนซ์ดังนี้
กำหนดให้ m_x, m_y เป็นจำนวนนับแทนจำนวนจุดกริด(พิกเซล)ในทาง x และทาง y ตามลำดับ และ

$$\Omega_h = \{\mathbf{x} \in \Omega | \mathbf{x} = (x_i, y_j)^\top = (ih_x, jh_y), 1 \leq i \leq m_x, 1 \leq j \leq m_y\}$$

แทนโดยเมนของผลเฉลยที่ถูกดิสcretize ให้ $u(x, y)$ แทนค่าตอบของปัญหา (3.4.9), ให้ $(\lambda)_{i,j}$ และ $\lambda(x_i, y_j)$, ให้ $(u)_{i,j}$ และ $u(x_i, y_j)$, ให้ $(z)_{i,j}$ และ $z(x_i, y_j)$, $u(x_i, y_j)$, ให้ $(w)_{i,j}$ และ $w(x_i, y_j)$, ให้ $(b)_{i,j}$ และ $b(x_i, y_j)$ และ h แทนระยะห่างของแต่ละจุดกริด เราจะทำการใช้สูตรพ่อร์เวิร์ดดิฟเพอร์ร์เรนจ์ได้ดังนี้

$$\begin{aligned} & -\theta \frac{1}{h^2} \left((u)_{i-1,j} + (u)_{i+1,j} + (u)_{i,j-1} + (u)_{i,j+1} - 4(u)_{i,j} \right) + (\lambda)_{i,j}(u)_{i,j} \\ & = (\lambda)_{i,j}(z)_{i,j} - \theta \left(\left(\frac{1}{h} \left((w)_{i,j} - (b)_{i,j} \right) - \frac{1}{h} \left((w)_{i-1,j} - (b)_{i-1,j} \right) \right) \right. \\ & \quad \left. + \left(\frac{1}{h} \left((w)_{i,j} - (b)_{i,j} \right) - \frac{1}{h} \left((w)_{i,j-1} - (b)_{i,j-1} \right) \right) \right) \end{aligned} \quad (3.4.10)$$

ภายใต้เงื่อนไขขอบ $(u)_{i,j} = (u)_{1,j}$ เมื่อ $i < 0$, $(u)_{i,j} = (u)_{m_x,j}$ เมื่อ $i > m_x$, $(u)_{i,j} = (u)_{i,0}$ เมื่อ $j < 0$ และ $(u)_{i,m_y} = (u)_{i,m_y}$ เมื่อ $j > m_y$

ในทำนองเดียวกันโดยแคลคูลัสของการประพันจะได้สมการอยเลอร์ลากราจ์สำหรับ \mathcal{J}_2 ใน (3.4.7) เทียบกับตัวแปร w กำหนดโดย

$$\begin{cases} \frac{w}{|w|} + \theta(w - \nabla u - b) = 0 & x \in \Omega \\ \frac{\partial u}{\partial n} = 0, & x \in \partial\Omega \end{cases} \quad (3.4.11)$$

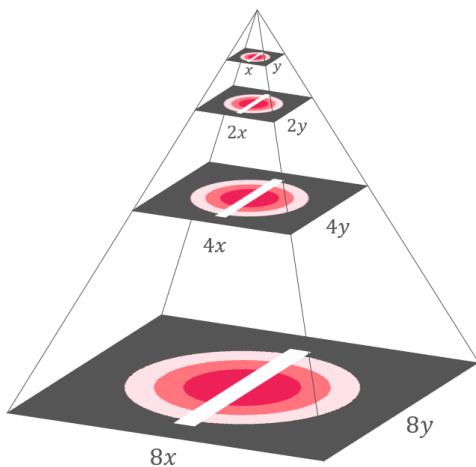
ซึ่งสมการเชิงอนุพันธ์ดังกล่าวมีผลเฉลยแบบแม่นตรง [4] คือ

$$w = \max \left\{ (\nabla u + b) - \frac{1}{\theta}, 0 \right\} \quad (3.4.12)$$

3.5 ขั้นตอนวิธีเชิงตัวเลขที่นำเสนอด้วย

3.5.1 ขั้นตอนวิธีเชิงตัวเลขสำหรับต่อเติมภาพศิลปะ

สำหรับวิธีการซ่อมแซมภาพศิลปะไทย จะใช้วิธีการสบริทเบรกเมนเพื่อหลิกเลี่ยงปัญหาเชิงตัวเลขที่จะเกิดขึ้น แต่เพื่อให้วิธีการสบริทเบรกเมนประมวลผลภาพได้รวดเร็วขึ้น ผู้วิจัยได้พัฒนากระบวนการกำหนดค่าตอบเริ่มต้นโดยวิธีการมัลติเรโซลูชัน (multi-resolution method) หรือวิธีการพิริมิติรูปภาพ (pyramid method) [5] เริ่มจากการย่อขนาดรูปลงครึ่งหนึ่งโดยใช้วิธี Bilinear Interpolation จนกระทั่งถึงระดับความคมชัดที่ต้องการ จากนั้นทำการต่อเติมภาพขนาดเล็ก และนำผลลัพธ์ที่ได้จากภาพขนาดเล็กทำการขยายภาพขึ้นสองเท่าโดยใช้ Bilinear Interpolation เป็นค่าตอบเริ่มต้นสำหรับการต่อเติมภาพในชั้นถัดไป



รูปที่ 3.5.1: วิธีการพิระมิดรูปภาพ

ขั้นตอนวิธีสำหรับการทำพิระมิดรูปภาพสำหรับการต่อเติมภาพแบบสปริตเบรกแม่นเพื่อให้ประมวลผลได้เร็วขึ้นนั้นสามารถสรุปได้ดังนี้

Algorithm 5: วิธีสปริทเบรกเมนท์ใช้พีระมิดรูปภาพ

Input:

- u คือภาพที่ต้องการต่อเติม/ภาพที่ได้รับการต่อเติม
- λ คือพารามิเตอร์เริกกิวลาร์เรซชัน ที่ได้กล่าวถึงในสมการ (3.2.2)
- θ คือพารามิเตอร์เพนล์ที่ซึ่งเป็นจำนวนเต็มบวก
- N_{GS} เป็นจำนวนเต็มบวก
- c ตัวแปรช่วยสำหรับบอกความลึก ให้กำหนดเป็น 1
- m คือ ระดับความลึกของพีระมิดรูปภาพ เป็นจำนวนเต็มบวก
- N_0 จำนวนรอบการทำสปริทเบรกเมนท์ขั้นความคมชัดสูงสุด
- N_1 จำนวนรอบการทำสปริทเบรกเมนท์ขั้นระหว่างกลาง
- N_2 จำนวนรอบการทำสปริทเบรกเมนท์ขั้นความคมชัดหมายบสุด

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

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

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

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

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

$$\Delta r = r_f - r$$

$$\Delta c = c_f - c$$

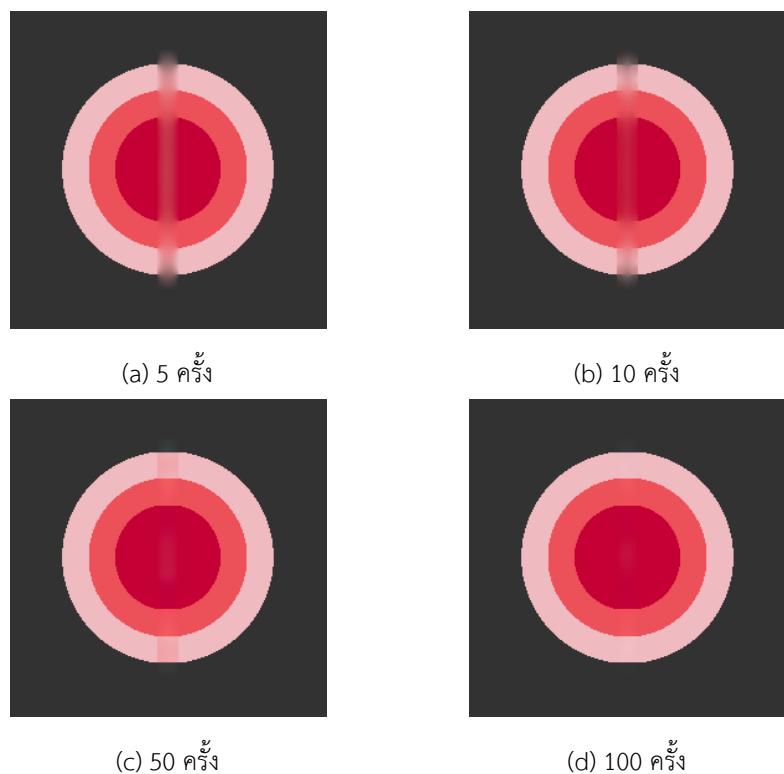
$$J(r', c') = I(r, c) \cdot (1 - \Delta r) \cdot (1 - \Delta c)$$

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

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

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

นอกจากนี้แล้ว ผู้วิจัยได้สังเกตว่า การทำซ้ำจะถูกเข้าเร็วในช่วงแรก จากนั้นความเร็วในการถูกเข้าจะลดลง ซึ่งทำให้เราสามารถใช้จำนวนรอบของการทำซ้ำเพียงไม่กี่ครั้งในระดับความคมชัดเดิมเพื่อซ่อมแซมภาพ



รูปที่ 3.5.2: ผลการซ้อมแซมภาพวิเคราะห์เมื่อใช้จำนวนรอบในการทำข้อที่ระดับความคุณชั้นสูงสุดซึ่งมีค่าต่างกัน

จากรูปที่ 3.5.2 แสดงให้เห็นถึงจำนวนรอบการทำซ้ำที่ความคุณภาพสูงสุด 10 ครั้งเพียงพอต่อการซ้อมแซม ภาพจนมีผลการซ้อมแซมภาพที่ดี นอกจากนี้เรายังพบจากการตรวจสอบว่าการใช้จำนวนรอบของการทำซ้ำในระดับความคุณภาพต่ำยังทำให้ผลการทำงานในภาพรวมเร็วขึ้นอย่างมีนัยสำคัญ

3.5.2 ขั้นตอนวิธีเชิงตัวเลขสำหรับซ่อมแซมภาพวิดีโอ

เนื่องจากไฟล์วิดีโอนั้นประกอบด้วยชุดของภาพหล่ายภาพ กล่าวคือ $V = \{u_i | i = 1, 2, 3...N_f\}$ ทำให้ขั้นตอนการระบบทบทรรยายนอกจากวิดีโอ จะต้องทำการต่อเติมภาพในบริเวณที่เป็นบทบรรยายที่ลงทะเบียนไว้ในขั้นตอนนี้

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

```

ขั้นตอนการต่อเติมภาพ u_i ในโดเมนต่อเติม D สามารถใช้วิธีการเดียวกับการซ่อมแซมภาพคลิปไปได้ ส่วนการหาโดเมนต่อเติมซึ่งเป็นบทบรรยายอนิเมะจะกล่าวถึงในหัวข้ออย่างถัดไป

3.5.3 การหาบทบรรยายบนอนิเมะ

ก่อนจะลบทบรรยายนั้น จำเป็นต้องหาบทบรรยายในภาพให้ได้เสียก่อน โดยบทบรรยายของอนิเมะนั้น มักจะใช้ขอบของตัวอักษรสำคัญ อีกทั้งบทบรรยายนั้นมีตำแหน่งที่ทางออกมากจากขอบของวิดีโอ และขนาดของบทบรรยายนั้นจะมีขนาดเล็กเมื่อเทียบกับเฟรมภาพ ด้วยสมบัตินี้เองทำให้เจ้าสามารถหาบริเวณบนเฟรมที่เป็นบทบรรยายได้โดยใช้ขั้นตอนวิธีที่สรุปดัง Algorithm 8 ดังนี้

Algorithm 8: Finding subtitle

```

 $D \leftarrow \text{findsub}(\mathbf{u})$ 
• ทำการเปลี่ยนสีดำในภาพ  $\mathbf{u}$  ให้เป็นสีขาวแล้วเปลี่ยนอื่นๆ ให้เป็นสีดำเพื่อหาขอบของคำบรรยาย
• เปลี่ยนบริเวณสีขาวในภาพให้เป็นสีดำ และเปลี่ยนบริเวณสีดำให้เป็นสีขาว
• ทำการลับบริเวณสีขาวซึ่งติดกับขอบของภาพออกไป เนื่องจากบทบรรยายจะลอยอยู่ไม่ติดกับขอบเสมอ
• ลับบริเวณที่ใหญ่เกินกว่าจะเป็นบทบรรยาย
• ลับบริเวณที่เล็กเกินกว่าจะเป็นบทบรรยาย
• ทำการขยายพื้นที่ เป็นสีขาวขึ้นด้วยความกว้างของขอบบทบรรยาย
• สีขาวที่เหลืออยู่ในภาพจะเป็นบทบรรยาย

```

3.5.4 การลบบทบรรยายจากบทอนิเมะ

เพื่อเร่งการลบบทบรรยาย เราสามารถใช้ผลการต่อเติมภาพจากเฟรมก่อนหน้า มาใช้เป็นคำตอบเริ่มต้นของเฟรมปัจจุบัน จึงได้ว่า

Algorithm 9: วิธีการทำงานบนวิดีโอ เมื่อต้องการผลลัพธ์ที่แล้วมาใช้เป็นคำตอบเริ่มต้น

```

 $V \leftarrow RemoveSubtitle(V)$ 
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

```

นอกจากนี้เราสามารถใช้แนวคิดของการยึดมุลจากเฟรมที่เกี่ยวข้องเพื่อเป็นข้อมูลเริ่มต้นในการเร่งการประมวลผลได้ดังนี้

Algorithm 10: การลบบทบรรยายโดยใช้วิธีการยึดมุล

```

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

```

Algorithm 11: การลบบทบรรยายโดยใช้วิธีการข้ามเฟรม

```

 $v, \leftarrow RemoveBySkipFrame(u, D, v)$ 
 $s =$  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

```

Algorithm 12: การลบบทบรรยายโดยใช้วิธีการข้ามเฟรมและยืมเฟรม

```

 $v \leftarrow RemoveBySkipAndBorrowFrame(u, D, v)$ 
 $s =$  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

```

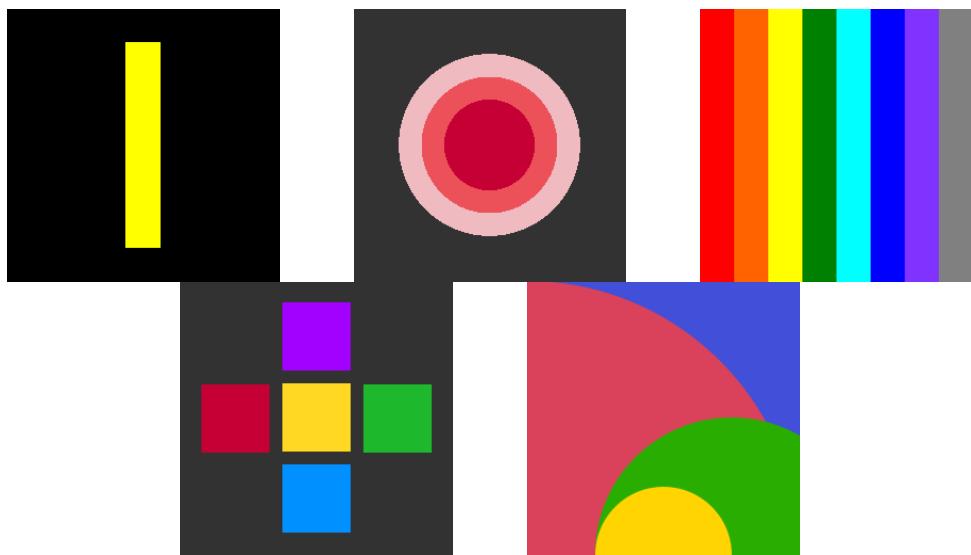
บทที่ 4

การทดลองเชิงตัวเลข

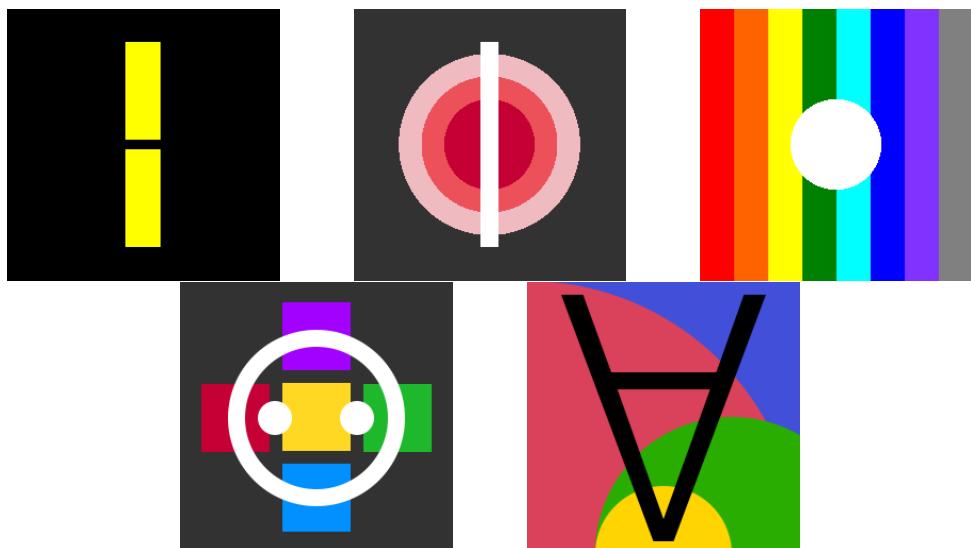
การทดลองเชิงตัวเลขถือเป็นสิ่งสำคัญของโครงการนิจัยชิ้นนี้ ผู้วิจัยจะทำการทดลองเชิงตัวเลขเพื่อเปรียบเทียบวิธีที่พัฒนาขึ้นมากับวิธีที่มีอยู่เดิมและเปรียบเทียบกันเองระหว่างวิธีที่ได้พัฒนาขึ้น โดยการทดลองเชิงตัวเลข จะแบ่งออกเป็น 2 ส่วนได้แก่ การทดลองซ่อมแซมภาพศิลปะไทย และการทดลองระบบบรรยายอุกกาโนนเมะ

4.1 การซ่อมแซมภาพจิตกรรมไทยโบราณ

สำหรับการซ่อมแซมจิตกรรมไทยโบราณ ผู้วิจัยจะเริ่มจากการทำการปรับปรุงขั้นตอนวิธีเชิงตัวเลขที่มีอยู่แล้ว โดยใช้ภาพสีที่ได้สংเคราะห์ขึ้นทั้งสิ้น 5 ภาพ โดยแต่ละภาพมีขนาด 256×256 พิกเซล ดังรูปที่ 4.1.1



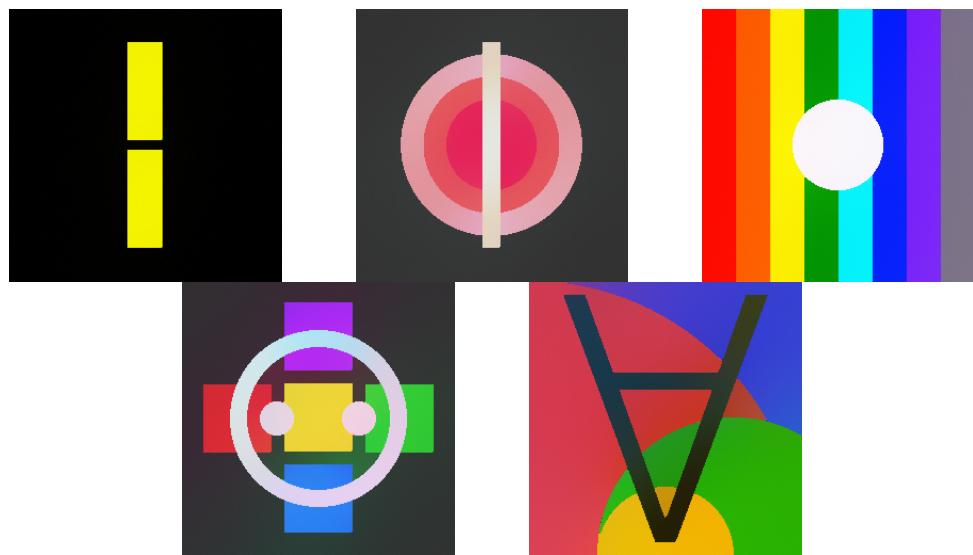
รูปที่ 4.1.1: ภาพต้นฉบับ



รูปที่ 4.1.2: ภาพที่จะทำการซ่อมแซม

4.1.1 การเปรียบเทียบประสิทธิภาพขั้นตอนวิธีเชิงตัวเลขที่มีอยู่แล้ว

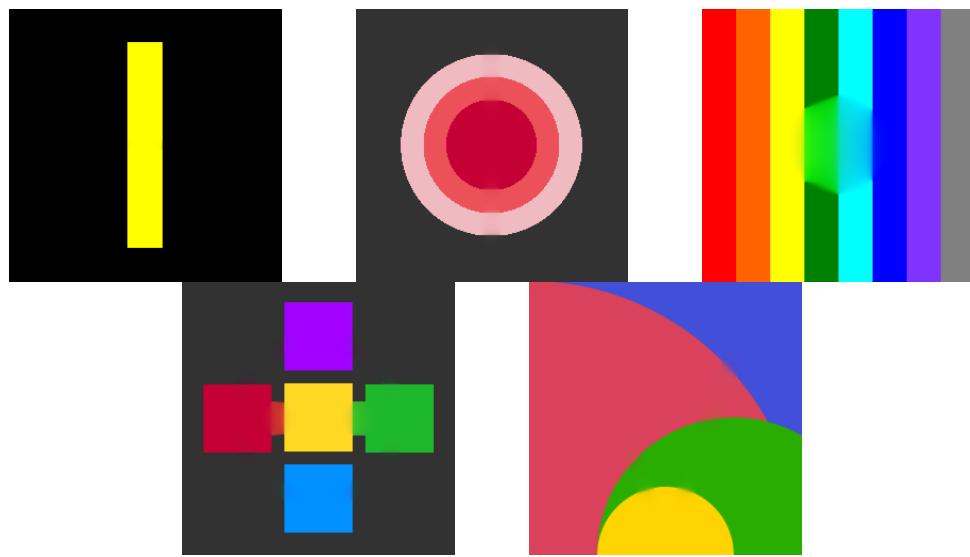
การทดสอบประสิทธิภาพจะใช้ $\varepsilon = 1 \times 10^{-4}$ และจำนวนการทำซ้ำ (N สำหรับวิธีเดินเวลา, N_{FP} สำหรับวิธีทำซ้ำจุดตرجิร์ง และ N_{SB} สำหรับวิธีสปริทเบรกแม่น) 10,000 รอบ โดยรูปที่ 4.1.3 - 4.1.5 และตารางที่ 4.1 - 4.3 แสดงผลการซ่อมแซมภาพสังเคราะห์ทั้ง 5 ภาพ



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

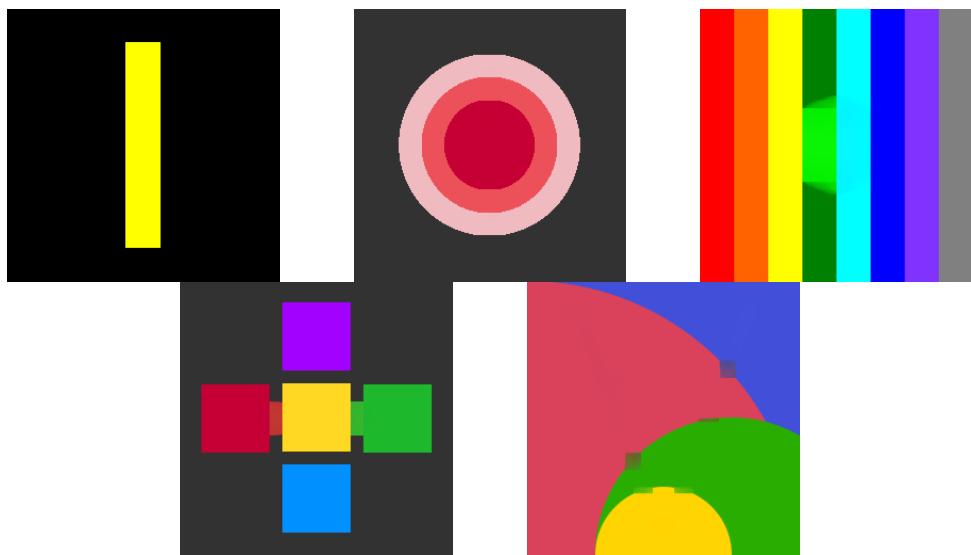
ตารางที่ 4.1: ผลการซ่อมแซมวิธีการเดินเวลา (Algorithm 1)



รูปที่ 4.1.4: ผลการซ่อมแซมจากวิธีการทำซ้ำแบบจุดตรีง

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: ผลการซ่อมแซมของวิธีการทำซ้ำแบบจุดตรีง (Algorithm 2)



รูปที่ 4.1.5: ผลการซ่อมแซมจากวิธีการสปริทเบรกแม่น

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: ผลการซ่อมแซมของวิธีสปริทเบรกแม่น (Algorithm 4)

ประสิทธิภาพของวิธีการเชิงตัวเลขทั้ง 3 วิธี สามารถสรุปได้ดังนี้

วิธีการ	เวลาประมาณ (วินาที)	PSNR (dB)	SSIM
วิธีเดินเวลา (Algorithm 1)	120.68	16.72	0.9960
วิธีทำซ้ำจุดตรึง (Algorithm 2)	74.81	38.67	0.9999
วิธีสปริทเบรกแม่น (Algorithm 4)	14.06	39.42	0.9999

ตารางที่ 4.4: แสดงการซ่อมแซมเฉลี่ยของวิธีการเชิงตัวเลข

จากทั้ง 3 วิธีที่ได้ทดสอบ เรายกบว่าวิธีการสปาร์ทเบรกแม่นใช้เวลาอ้อยกว่าวิธีอื่น และมีคุณภาพ (จีงพิจารณาจาก ค่า PSNR และค่า SSIM) สูงกว่าวิธีอื่น ผู้วิจัยจึงสนใจทำการปรับปรุงวิธีสปาร์ทเบรกแม่นให้มีประสิทธิภาพสูง

4.1.2 การทดสอบการขึ้นอยู่กับพารามิเตอร์

ตารางที่ 4.5 แสดงผลการเปรียบเทียบประสิทธิภาพของ Algorithm 5 ภายใต้การเปลี่ยนแปลงจำนวนรอบของการทำซ้ำของวิธีการสปาร์ทเบรกแม่นบนภาพที่มีความคมชัด 256×256 พิกเซล ตัวอย่าง เช่น $10/3/3/10000$ หมายถึงที่ระดับความคมชัดที่สุดซึ่งมีขนาดเป็น 32×32 พิกเซลจะทำซ้ำไม่เกิน 10 ครั้ง สำหรับที่ความคมชัดลดลงเป็น 64×64 พิกเซลจะทำซ้ำไม่เกิน 3 ครั้ง และสำหรับที่ระดับความคมชัดเป็น 128×128 พิกเซลจะทำซ้ำ 3 ครั้ง และที่ระดับความคมชัดเป็น 256×256 จะทำซ้ำไม่เกิน 10,000 ครั้งหรือจนค่าความคลาดเคลื่อนสัมพัทธ์ต่างกันไม่เกิน 0.0001

รูปแบบการทำข้าว	รูปภาพ	เวลาประมวล (วินาที)	PSNR (dB)	SSIM
ไม่ใช้พิระมิดรูปภาพ	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: ผลการซ้อมแซมภาพโดยวิธีการเชิงตัวเลขที่นำเสนอ

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 การทำข้าวันนั้นจะถูกเข้าเร็วในช่วงแรก จนกวนความเร็วในการถูกเข้าจะลดลง ซึ่งทำให้การทำข้าวเพียงไม่กี่ครั้งในระดับความคมชัดเดิม มีผลการซ่อมแซมภาพจนแสดงความคล้ายคลึงกับภาพต้นฉบับได้

ผู้วิจัยจึงกำหนดให้การทำข้าวในระดับความละเอียดสุดเท่ากับ 10 ครั้ง และพบว่าได้ผลการซ่อมแซมดังตารางที่ 4.7

รูปแบบการทำข้าว	รูปภาพ	เวลาประมาณ (วินาที)	PSNR (dB)	SSIM
'ไม่มีพิระมิดรูปภาพ	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
10/1/1/10	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
10/3/3/10	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
10/10/10/10	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
100/1/1/10	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
100/3/3/10	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
100/10/10/10	1	0.28	50.05	1.0000
	2	0.41	33.25	1.0000
	3	0.42	23.51	0.9995
	4	0.42	27.78	0.9998
	5	0.39	32.38	0.9999

ตารางที่ 4.7: ผลการซ่อมแซมภาพโดยวิธีการเชิงตัวเลขที่นำเสนอเมื่อใช้การทำข้าวในระดับความคมชัดละเอียดสุด 10 ครั้ง

รูปแบบการทำข้า	เวลาประมาณ (วินาที)	PSNR (dB)	SSIM
ไม่ใช้พิธีมิตรภาพ	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 การทำข้าในชั้นที่รูปภาพมีขนาดเล็กมากจำนวนมาก 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 จิตรกรรมฝาผนังวัดคงคาราม และภาพที่ 4.1.6e⁵ คือ จิตรกรรมฝาผนังวัดท่าถนน โดยจะทำให้ข้อมูลข้างทั้ง 5 ภาพเกิดความเสียหาย โดยใช้ร้อยความเสียหาย จากภาพพระเจ้าสร้างอดัม

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

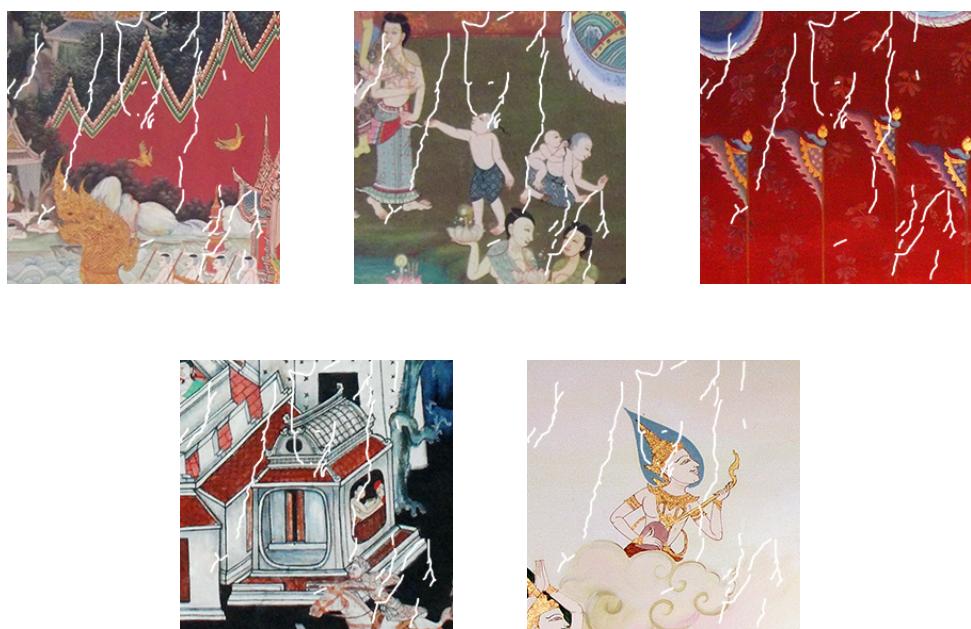
³ภาพถ่ายที่วัดพระยืนพุทธบาทยุคล; ภาพจาก https://commons.wikimedia.org/wiki/File:Wat_Phra_Yuen_Phutthabat_Yukhon_01.jpg สืบคันเมื่อวันที่ 23 กันยายน 2561

⁴ภาพถ่ายที่วัดคงคาราม; ภาพจาก https://commons.wikimedia.org/wiki/File:จิตรกรรม_อุโบสถวัดคงคาราม.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

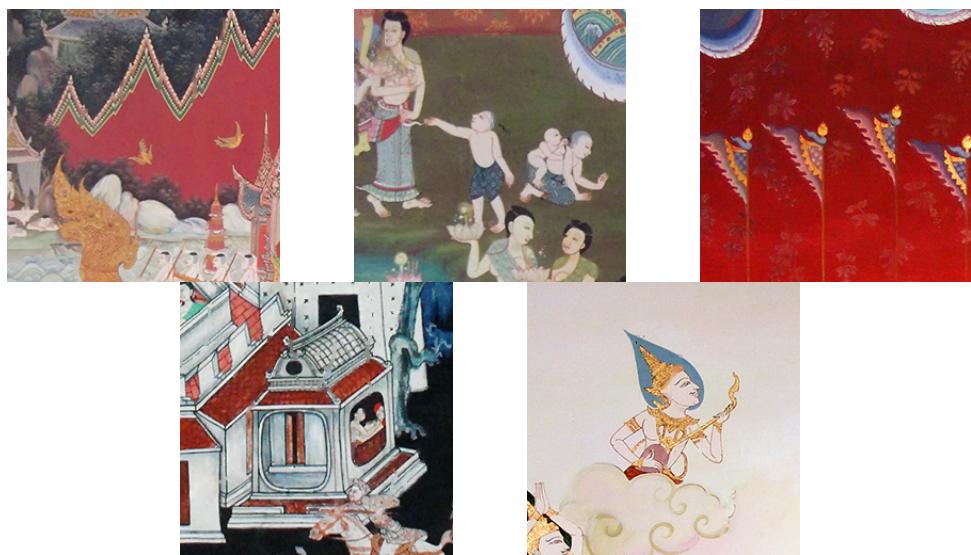
การสบริทเบรกแมนพร้อมทั้งการใช้พีระมิดรูปภาพที่มีการทำซ้ำแต่ละชั้นเป็น 10/3/3/10 ได้ผลลัพธ์ออกมาเป็นดังนี้



รูปที่ 4.1.8: ผลการซ่อมแซมโดยวิธีการสบริทเบรกแมน

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: ผลการซ่อมแซมภาพศิลปะไทยจากวิธีการสบริทเบรกแมน (Algorithm 4)



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 (Algorithm 5)

ทั้งสองวิธี ได้ผลลัพธ์การซ่อมแซมภาพศิลปะไทยในรูปค่าเฉลี่ยของมาดังนี้

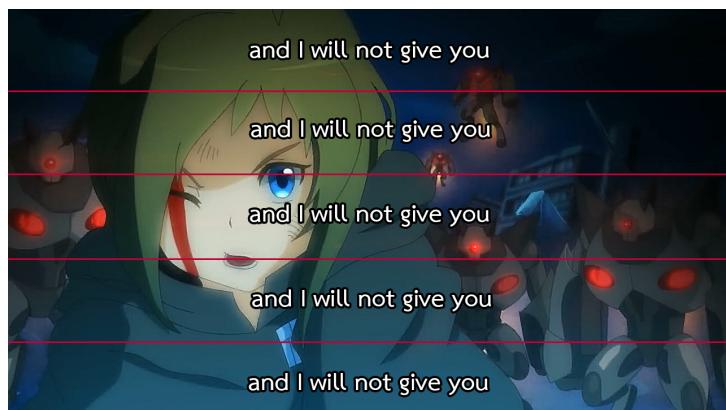
วิธีการ	เวลาประมาณ (วินาที)	PSNR (dB)	SSIM
วิธีสปริตเบรกแม่น (Algorithm 4)	2.72	34.89	1.0000
วิธีการที่พัฒนาขึ้น (Algorithm 5)	0.39	35.30	1.0000

ตารางที่ 4.11: แสดงผลการซ่อมแซมภาพศิลปะไทยในรูปค่าเฉลี่ยจากตารางที่ 4.9 และตารางที่ 4.10

จากตารางที่ 4.11 จะเห็นได้ว่า วิธีที่พัฒนาขึ้นนั้นสามารถทำงานได้เร็วกว่าวิธีสเปรทเบรกเมนเดิม และบังมีคุณภาพที่ดีขึ้นด้วย

4.2 การลับบทบรรยายบนอนิเมะ

สำหรับการลับบทบรรยายอนิเมะ จะใช้วิดีโอ Anime Festival Asia Special Video - feat. Inori Aizawa ซึ่งผลิตโดย Collateral Damage Studios โดยจะตัดวิดีโอด้วยตัดวิดีโอด้วยปกติแล้ว 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: การแบ่งไฟล์วิดีโอเป็น 5 ส่วนสำหรับใช้เป็น 5 ชุดทดสอบ

4.2.1 การหาบทบรรยายบนอนิเมะ

วิธีการหาบทบรรยายที่กล่าวไปข้างต้น จะทำการทดสอบกับบทความ lorem ipsum⁶ ที่ถูกแปลเป็นภาษาไทย ภาษาอังกฤษ และภาษาญี่ปุ่น โดยมีความสามารถในการหาโดยมีแต่ตัวอักษรในบทบรรยายภาษาต่างๆ ดังนี้

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

ภาษา	วีดีโอ	จำนวนพิกเซลในโดเมน	จำนวนพิกเซลที่ตรวจสอบ	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	จำนวนพิกเซลที่ตรวจสอบ	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 การลับบทบรรยายจากบทอนิเมะ

การลับบทบรรยายอนิเมะจะทำการทดลองเบรี่ยบเทียบระหว่าง วิธีสปริทเบรกแม่นและพ์รอมดูรูปภาพ (Algorithm 5) วิธียิ่มเฟรม (Algorithm 10) วิธีข้ามเฟรม (Algorithm 11) และวิธีข้ามและยิ่มเฟรมวิธีข้ามและ

ยึมเฟรม (12) โดยใช้ค่าพารามิเตอร์ชุดเดียวกับการทดลองภาพศิลปะไทย ได้ผลลัพธ์ดังตาราง

วิธีการ	วิดีโอ	เวลาประมาณ (วินาที)	PSNR (dB)	SSIM
วิธีสปริทเบรกแม่น และพีระมิดรูปภาพ (Algorithm 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
วิธียึมเฟรม (Algorithm 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
วิธีข้ามเฟรม (Algorithm 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
วิธีข้ามและยึมเฟรม (Algorithm 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: ผลการลับบทบรรยายออกจากอนิเมะด้วย Algorithm 5, 10, 11 และ 12

วิธีการ	เวลาประมาณ (วินาที)	PSNR (dB)	SSIM
วิธีสปริตเบรกเมนและพีระมิดรูปภาพ (Algorithm 5)	141.29	31.39	0.9510
วิธียีมเฟรม (Algorithm 10)	132.78	32.20	0.9655
วิธีข้ามเฟรม (Algorithm 11)	89.29	29.07	0.9408
วิธียีมเฟรมและข้ามเฟรม (Algorithm 12)	75.76	29.33	0.9454

ตารางที่ 4.15: ผลการซ่อมแซมภาพโดยวิธีการเชิงตัวเลขที่นำเสนอด้วยรูปของค่าเฉลี่ยของผลที่ได้จากตารางที่ 4.14

จากนั้นทำการทดสอบการต่อเติมวิดีโอห้อง 5 พบร่วมกับวิธีข้ามและยีมเฟรม (Algorithm 12) ใช้เวลาประมาณ 75.76 วินาที ผลน้อยที่สุดจึงให้วิธีนี้เป็นวิธีการที่ผู้วิจัยได้พัฒนาขึ้นสำหรับลบคำบรรยายอนิเมะ เมื่อเปรียบเทียบกับวิธีสปริตเบรกเมน (Algorithm 4) ที่มีอยุ่เดิมได้ผลลัพธ์ออกเป็นดังตารางนี้

วิธีการ	เวลาประมาณ (วินาที)	PSNR (dB)	SSIM
วิธีสปริตเบรกเมน (Algorithm 5)	5073.08	32.88	0.9654
วิธีการที่พัฒนาขึ้น (Algorithm 12)	75.76	29.33	0.9454

ตารางที่ 4.16: ผลการลบบทบรรยายออกจากอนิเมะโดยวิธีการสปริตเบรกเมนและวิธีการที่พัฒนาขึ้น

สำหรับวิธีสปริตเบรกเมน (Algorithm 4) เนื่องจากใช้เวลา 1 ชั่วโมงแล้วยังประมาณผลวิดีโอยุ่ดทดสอบ แรกไม่เสร็จ ทางผู้พัฒนาจึงตัดสินใจยุติการทดลอง เนื่องจากอาจต้องใช้เวลาการประมาณผลเป็นเวลาหลายชั่วโมง สำหรับวิดีโocomความยาว 1 นาที ส่วนวิธีที่คิดค้นขึ้น พบร่วมกับวิดีโอที่มีความยาว 1 นาที สามารถทำงานได้เร็ว อย่างรวดเร็ว โดยใช้เวลาเพียง 75 วินาที

บทที่ 5

สรุป

สำหรับโครงการวิจัยนี้ได้ทำการพัฒนาขั้นตอนวิธีเชิงตัวเลขสำหรับการซ่อมแซมภาพศิลปะไทยและการลบบทบรรยายอนิเมะได้มีผลการดำเนินงานทั้งสิ้นดังนี้

ขั้นตอนวิธีเชิงตัวเลขทั้ง 3 วิธีสำหรับแก้ตัวแบบการต่อเติมภาพด้วยการแปรผันรวม ได้แก่วิธีเดินเวลาแบบชัดแจ้ง (Algorithm 1) วิธีทำซ้ำจุดตรึง (Algorithm 2) และวิธีการสปริทเบรกแม่น (Algorithm ??) พบว่า วิธีการสปริทเบรกแม่นมีคุณภาพที่ดีกว่าเมื่อวัดด้วยค่า PSNR และ SSIM ส่วนเวลาที่ใช้ประมาณผลพบว่าวิธีสปริทเบรกแม่นใช้เวลาน้อยกว่าวิธีเดินเวลา 8 เท่าและใช้เวลาห้อยกว่าวิธีทำซ้ำจุดตรึง 5 เท่า

พิริมิดรูปภาพใช้เพื่อเพิ่มความเร็วในการประมาณผลของวิธีการสปริทเบรกแม่น พบร่วมกับการทำซ้ำบนพิริมิดรูปภาพด้วยการทำซ้ำ 10/3/3/10 ใช้เวลาประมาณผลน้อยที่สุดและให้ค่าคุณภาพทั้งในด้าน PSNR และ SSIM ใกล้เคียงกันอีกด้วยเลือกใช้พิริมิดรูปภาพนี้ในการซ่อมแซมรูปภาพ

การซ่อมแซมภาพศิลปะไทย การใช้วิธีการสปริทเบรกแม่นพร้อมทั้งการใช้พิริมิดรูปภาพ (Algorithm 5) พบร่วมกับที่ผ่านการต่อเติมให้คุณภาพดีกว่าเมื่อวัดด้วย PSNR และใช้เวลาห้อยกว่าประมาณ 7 เท่า

ขั้นตอนวิธีค้นหาคำบรรยายในภาพด้วยวิธีการที่คิดค้นขึ้น (Algorithm 8) พบร่วมกับมีความผิดพลาดในการตรวจหาพิกเซลที่เป็นบทบรรยายอยู่ที่ร้อยละ 11.98 ซึ่งจะใช้ขั้นตอนวิธีนี้ในการหาคำบรรยายเพื่อทำการลบถัดไป

เนื่องจากวิดีโอด้วยภาพค่อนข้างวิจัยจึงได้เสนอขั้นตอนวิธีข้ามเฟรม (Algorithm 10) ขั้นตอนวิธียึดเฟรม (Algorithm 11) และขั้นตอนวิธียึดเฟรมและข้ามเฟรม (Algorithm 12) เพื่อช่วยในการลดเวลาการประมาณผล พบร่วมกับวิธียึดเฟรมและข้ามเฟรมใช้เวลาประมาณผลน้อยสุด ซึ่งใช้เวลาห้อยกว่าวิธีสปริทเบรกแม่นและพิริมิดรูปภาพบันวิดีโอถึง 2 เท่า

สำหรับการตอบบทบรรยายอนิเมะเมื่อใช้วิธีตรวจหาคำบรรยายที่คิดขึ้นขึ้นพร้อมทั้งใช้วิธีการเดียวกับที่ใช้สำหรับซ่อมแซมศิลปะไทย รวมกับการวิธีการขึ้นเฟรมและข้ามเฟรม (Algorithm 12) พบว่าใช้เวลาอยู่กว่าวิธีการสปริทเบรคแม่น (Algorithm 4) บนวิดีโอ 67 เท่า

បរណ្ណការណ៍

- [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.
- [8] A. N. Tikhonov and V. Y. Arsenin. Solutions of Ill-posed Problems. Wiston and Sons, Washington, D.C., 1977
- [9] B. Dacorogna. Introduction to the Calculus of Variations. Introduction to the Calculus of Variations, 2004.

- [10] N. Chumchob. A study of effective variational models and efficient numerical methods for image registration. University of Liverpool, UK 2010
- [11] E. Giusti. Minimal Surfaces and Functions of Bounded Variation. Monographs in Mathematics, Vol. 80. Birkhauser, 1984

ภาคผนวก A

โปรแกรมที่พัฒนาขึ้น

A.1 โปรแกรมทดสอบ

สำหรับโครงการวิจัยเรื่องนี้เป็นการพัฒนาวิธีเชิงตัวเลข การจะวัดประสิทธิภาพของวิธีการเชิงตัวเลขได้จำเป็นจะต้องใช้โปรแกรมเข้าทดสอบ โดยโครงการวิจัยนี้ มีโปรแกรมสำหรับทดสอบ ซึ่งสามารถแบ่งได้ออกเป็น 2 ส่วนคือโปรแกรมสำหรับทดสอบขั้นตอนการซ่อมแซมภาคศิลปะไทยและการลบคำบรรยายอนิเมะ

A.1.1 โปรแกรมทดสอบการซ่อมแซมภาคศิลปะไทย

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

โดยการรันให้เปิด command prompt โดยการ เปิดปุ่ม Windows+R จากนั้นพิมพ์ cd ”ชื่อโฟลเดอร์” ที่ได้ทำการดาวน์โหลด releases มาแตกไฟล์ไว้ จากนั้นใช้คำสั่ง cd ”application” เพื่อเข้าไปโฟลเดอร์ที่มีไฟล์โปรแกรมอยู่ แล้วจึงใช้คำสั่ง labarotory.exe เพื่อทำการทดสอบ โดยไฟล์ผลลัพธ์จากการทดสอบจะปรากฏในโฟลเดอร์ 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 โปรแกรมทดสอบการลบบทบรรยายจากอนิเมะ

สำหรับโปรแกรมทดสอบการลบคำบรรยายอนิเมะนั้น เจียนด้วยภาษา power shell เพื่อใช้ในการทดสอบ โดยเครื่องที่จะนำไปทดสอบ 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-20181111.ps1
=====
splitbergman
Loop: 10
Processed in 170.6230171
Comparing Result... splitbergman
Compared in 167.0800813
[Parse_ssim_0 @ 00000159136e40c0] SSIM R:0.700363 (5.234051) G:0.711575 (5.399673) B:0.714408 (5.442540) All:0.708782 (5.357821)
[Parse_psnr_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 สามารถทดสอบ

ได้โดยการรันสคริปต์ <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

```

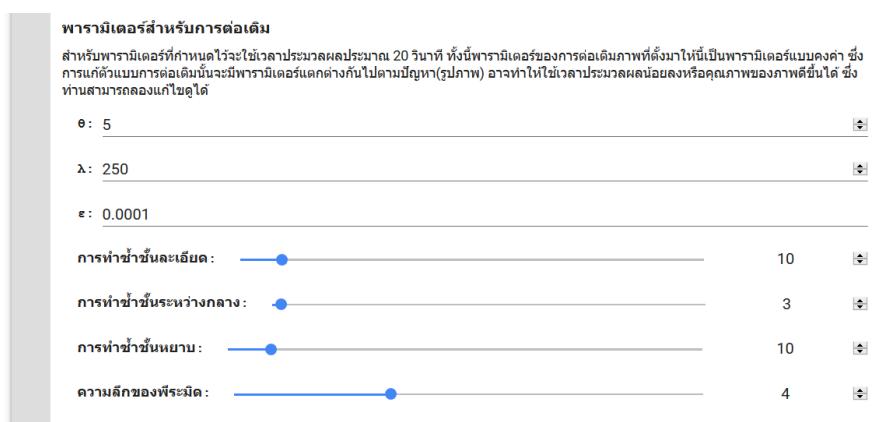
รูปที่ A.1.3: ตัวอย่างโปรแกรมสำหรับทดสอบการทำงานของภาษาที่พัฒนาขึ้น

A.2 โปรแกรมตัวอย่างการซ่อมแซมภาพศิลปะไทย

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



รูปที่ A.2.1: ตัวอย่างการเลือกรูปภาพสำหรับทำการทดสอบ



รูปที่ A.2.2: ตัวอย่างการปรับค่าพารามิเตอร์ที่ใช้ในโครงงานวิจัยนี้



รูปที่ A.2.3: ตัวอย่างภาพผลลัพธ์จาก Google Colab

A.3 โปรแกรมตัวอย่างการลบบทบรรยายจากอนิเมะ

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 เพื่อระบุที่อยู่ตำแหน่งของคำบรรยายในหน่วยพิกเซล 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