

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 เนื่องจากสมบัติการอนุรักษ์เส้นของเร็กกิวลาไรซ์เซชันแบบการแปรผันรวม อย่างไรก็ตาม การพัฒนาวิธีการเชิงตัวเลขที่มีประสิทธิภาพสำหรับสร้างคำตอบซึ่งสอดคล้องกับสมบัติดังกล่าวเป็นงานที่ท้าทาย เนื่องจากเร็กกิวลาไรซ์เซชันแบบการแปรผันรวมมีสมบัติซึ่งทางนุพนธ์ไม่ได้และไม่เป็นเชิงเส้น ในงานวิจัยนี้ เราพิจารณาปัญหาการแก้ตัวแบบดังกล่าว จากความสำเร็จของขั้นตอนวิธีการเชิงตัวเลขสำหรับปัญหาการกำหนดสัญญาณรบกวนออกจากภาพ เราได้นำเสนอวิธีการสนับสนุน สำหรับแต่ละรอบของการทำซ้ำ การคำนวณของวิธีการนี้ต้องการแก้ 2 ปัญหาย่อย ในปัญหาอย่างแรก เราพบความยุ่งยากในการหาคำตอบแม่นตรง สำหรับปัญหาย่อยที่ 2 คำตอบได้นำเสนออยู่ในรูปแบบปิด ทั้งนี้ เราได้นำเสนอขั้นตอนวิธีเชิงตัวเลขแบบใหม่ (ขั้นตอนวิธีแรก) ที่ใช้การหาค่าต่ำสุดแบบลับในกรอบความคิดหลายระดับเพื่อสร้างคำตอบเชิงตัวเลขที่รวดเร็วและแม่นยำ หลังจากนั้น เราได้ศึกษาปัญหาการลบบทบรรยายออกจากอนิเมะ เราพบว่าตัวแบบการต่อเติมภาพดังกล่าวสามารถนำมาปรับปรุงเพื่อให้ผลลัพธ์ที่น่าพอใจ ใน การแก้ตัวแบบที่ปรับปรุง เรานำเสนอด้วยขั้นตอนวิธีการข้ามและการยืด (ขั้นตอนวิธีที่สอง) ซึ่งใช้ขั้นตอนวิธีแรกเพื่อกำจัดบทบรรยายอย่างมีประสิทธิภาพ การทดสอบบนภาพสังเคราะห์และภาพศิลปะไทยโบราณยืนยันว่า ขั้นตอนวิธีแรกมีประสิทธิภาพสูงกว่าวิธีการพื้นฐาน นอกจากนี้เราพบว่า ขั้นตอนวิธีที่สองได้นำไปสู่ผลการลบบทบรรยายจากอนิเมะที่มีคุณภาพสูงอย่างรวดเร็ว

คำสำคัญ: ขั้นตอนวิธีเชิงตัวเลข, ตัวแบบเชิงการแปรผัน, การต่อเติมภาพ

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 and 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

กิตติกรรมประกาศ

การจัดทำโครงการงานวิจัยนี้สำเร็จลุล่วงไปได้ด้วยดีเนื่องจากคณะกรรมการวิจัยได้รับความกรุณาจากผู้ช่วยศาสตราจารย์ ดร.นพดล ชุมชอบ อาจารย์ที่ปรึกษาโครงการงานวิจัยผู้ให้ความรู้ ให้คำแนะนำและแก้ไขข้อบกพร่องต่างๆ ตลอดช่วงของการทำโครงการงานวิจัย คณะกรรมการวิจัยจึงขอกราบขอบพระคุณเป็นอย่างสูง ณ โอกาสนี้

ขอขอบพระคุณอาจารย์ ดร.นัยนรัตน์ กันยะมี และอาจารย์ผู้ช่วยศาสตราจารย์ ดร.พรทวัพย์ พรสวัสดิ์ กรรมการสอบโครงการงานวิจัยสำหรับคำแนะนำที่มีประโยชน์ รวมถึงอาจารย์วิทยากรทุกท่านที่สอนใช้งานโปรแกรมเลเท็กซ์ (LaTeX)

ขอขอบพระคุณอาจารย์ประจำภาควิชาคณิตศาสตร์ทุกท่านที่ได้อบรมสั่งสอนมอบวิชาความรู้อันมีค่าแก่ คณะกรรมการวิจัย และขอขอบพระคุณเจ้าหน้าที่ประจำภาควิชาคณิตศาสตร์ที่ได้อำนวยความสะดวกด้านต่างๆ ในการจัดทำโครงการงานวิจัยฉบับนี้

ขอขอบพระคุณบิดา มารดา และเพื่อนๆ ผู้ให้กำลังใจ รวมไปถึงการให้ความช่วยเหลือในทุกๆ 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	พัฟ์ชันของการแปรผันที่มีข้อบกพร่อง	11
2.4	วิธีการเร็กกิวลารีไลร์เซชัน	13

iv

2.4.1	ปัญหา Well-posed และปัญหา Ill-posed	13
2.4.2	ปัญหาย้อนกลับ	13
2.4.3	เร็กกิวลาดีลีซ์เซชัน	14
2.5	วิธีการไฟน์ติดฟเฟอเรนซ์เบื้องต้น	15
2.6	วิธีการทำซ้ำสำหรับระบบสมการเชิงเส้น	17
2.6.1	วิธีการจัโคบี	18
2.6.2	วิธีการเกาส์-ไซเดล	19
2.7	การรัดคุณภาพของภาพที่ผ่านกระบวนการต่อเติม	20
2.7.1	Peak Signal Noise Ratio	20
2.7.2	Structral Similarity	20
3	ขั้นตอนวิธีเชิงตัวเลขชนิดใหม่	22
3.1	การนำเสนօภาพ	22
3.1.1	การนำเสนօภาพเฉดเทา	22
3.1.2	การต่อเติมภาพเฉดเทา	23
3.1.3	การนำเสนօภาพสี	24
3.2	ตัวแบบเชิงแปรผันสำหรับต่อเติมภาพเฉดเทา	24
3.3	ตัวแบบเชิงแปรผันสำหรับการต่อเติมภาพสี	25
3.4	ขั้นตอนวิธีเชิงตัวเลขพื้นฐาน	25
3.4.1	ขั้นตอนวิธีเดินเวลา (explicit time marching method)	25
3.4.2	วิธีการทำซ้ำแบบจุดตึง (fixed-point iteration method)	27
3.4.3	วิธีการสปริทเบรกแมน (Split Bregman method)	29
3.5	ขั้นตอนวิธีเชิงตัวเลขที่นำเสนօ	32
3.5.1	ขั้นตอนวิธีเชิงตัวเลขสำหรับต่อเติมภาพศิลปะ	32
3.5.2	ขั้นตอนวิธีเชิงตัวเลขสำหรับซ่อมแซมภาพวิดีโอ	35
3.5.3	การทำบทบรรยายบนอนิเมะ	36
3.5.4	การลบคำบรรยายจากบทอนิเมะ	37

4 การทดลองเชิงตัวเลข	40
4.1 การซ่อมแซมภาพจิตรกรรมไทยโบราณ	40
4.1.1 การเปรียบเทียบประสิทธิภาพขันตอนวิธีเชิงตัวเลขที่มีอยู่แล้ว	41
4.1.2 ขันตอนวิธีเชิงตัวเลขสำหรับต่อเติมภาพชนิดใหม่	45
4.1.3 การทดสอบประสิทธิภาพในการซ่อมแซมภาพจิตรกรรมไทยโบราณ	49
4.2 การลับแบบบรรยายบนอนิเมะ	53
4.2.1 การหาบทบรรยายบนอนิเมะ	53
4.2.2 การลับคำบรรยายจากบทอนิเมะ	54
5 สรุป	57
บรรณานุกรม	59
ภาคผนวก A	61
A โปรแกรมที่พัฒนาขึ้น	61
A.1 โปรแกรมสำหรับทดสอบ	61
A.1.1 โปรแกรมสำหรับทดสอบการซ่อมแซมภาพศิลป์ไทย	61
A.1.2 โปรแกรมทดสอบการลับคำบรรยายอนิเมะ	62
A.2 โปรแกรมตัวอย่างการซ่อมแซมภาพศิลป์ไทย	63
A.3 โปรแกรมตัวอย่างการลับคำบรรยายอนิเมะ	64

สารบัญ

1.1.1	ตัวอย่างการซ่อมแซมภาพ	2
1.2.1	ภาพจิตรกรรมไทยที่วัดภูมิทรัตน์ อำเภอเมือง จังหวัดป่าน	3
1.2.2	1 เฟรมของอนิเมะที่มีบทบรรยายแบบแข็ง	4
2.3.1	พังก์ชันแปรผันมีขอบเขตทั้งสามพังก์ชันที่มีการแปรผันรวม เมื่ອนกันเท่ากับ 1 และพังก์ชันที่มีการแปรผันไม่จำกัด	12
2.4.1	a	14
2.5.1	ตัวอย่างการทำอนุพันธ์บนภาพเดา	16
3.1.1	ตัวอย่างภาพเดาที่แสดงระดับความเข้มของภาพในแต่ละระดับ	23
3.1.2	D แทนโคลเมนตอเติม	23
3.5.1	วิธีการพิริมิตรรูปภาพ	32
3.5.2	พิริมิทที่ลำดับการทำข้าเป็น 10/10/10 และที่ระดับความคมชัดจะลดลงเรื่อยๆ จำนวนการทำข้าที่ต่างกัน	35
4.1.1	ภาพต้นฉบับ	41
4.1.2	ภาพที่จะทำการซ่อมแซม	41
4.1.3	ผลการซ่อมแซมจากวิธีการเดินเวลา	42
4.1.4	ผลการซ่อมแซมจากการทำข้าแบบจุดตึง	43
4.1.5	ผลการซ่อมแซมจากการสเปรย์เบรกแม่น	44
4.1.6	ภาพต้นฉบับสำหรับใช้ในการทดสอบ	50
4.1.7	ภาพที่ทำให้เสียหาย	50

4.1.8	ผลการซ่อมแซมโดยวิธีการสปริทเบรกเมน	51
4.1.9	ผลการซ่อมแซมภาพโดยวิธีการใช้ตัวเลขที่พัฒนาขึ้น	52
4.2.1	การแบ่งไฟล์วิดีโอเป็น 5 ส่วนสำหรับใช้เป็น 5 ชุดทดสอบ	53
A.1.1	ตัวอย่างโปรแกรมสำหรับทดสอบการซ่อมแซมภาพศิลปะไทยที่พัฒนาขึ้น	62
A.1.2	ตัวอย่างโปรแกรมสำหรับทดสอบการลบคำบรรยายที่พัฒนาขึ้น	62
A.1.3	ตัวอย่างโปรแกรมสำหรับทดสอบการหาคำบรรยายที่พัฒนาขึ้น	63
A.2.1	ตัวอย่างการเลือกรูปภาพสำหรับทำการทดสอบ	63
A.2.2	ตัวอย่างการปรับค่าพารามิเตอร์ที่ใช้ในโครงงานวิจัยนี้	64
A.2.3	ตัวอย่างภาพผลลัพธ์จาก Google Colab	64
A.3.1	ไฟล์ตัวอย่างเมื่อทำการแตกไฟล์ออกมาแล้ว test.mp4 เป็นวิดีโอมีคำบรรยาย และ SubtitleRemove.avs เป็นโปรแกรมตัวอย่างสำหรับลบคำบรรยาย	65
A.3.2	test.mp4 (ซ้าย) และ SubtitleRemove.avs (ขวา) เมื่อเปิดด้วย MPC-HC	65
A.3.3	SubtitleRemove.avs สามารถแก้พารามิเตอร์เพื่อใช้กับวิดีโອอนิเมชันได้	65

สารบัญตาราง

4.1	ผลการซ่อมแซมวิธีการเดินเวลา	42
4.2	ผลการซ่อมแซมของวิธีการทำข้า้แบบจุดตึง	43
4.3	ผลการซ่อมแซมของวิธีสปริทเบรกแม่น	44
4.4	แสดงการซ่อมแซมเฉลี่ยของวิธีการเขิงตัวเลข	44
4.5	ผลการซ่อมแซมภาพโดยวิธีการเขิงตัวเลขที่นำเสนอน	46
4.6	ผลการซ่อมแซมภาพโดยวิธีการเขิงตัวเลขที่นำเสนอในรูปของค่าเฉลี่ยของผลที่ได้จากตารางที่ 4.5	47
4.7	ผลการซ่อมแซมภาพโดยวิธีการเขิงตัวเลขที่นำเสนอยื่นใช้การทำข้า้ในระดับความคมชัดละเอียดสุด 10 ครั้ง	48
4.8	ผลการซ่อมแซมภาพโดยวิธีการเขิงตัวเลขที่นำเสนอในรูปของค่าเฉลี่ยของผลที่ได้จากตารางที่ 4.7	49
4.9	ผลการซ่อมแซมภาพศิลปะไทยจากวิธีการสปิทเบรกแม่น	51
4.10	ผลการซ่อมแซมภาพศิลปะไทยโดยวิธีการเขิงตัวเลขที่พัฒนาขึ้น	52
4.11	แสดงผลการซ่อมแซมภาพศิลปะไทยในรูปค่าเฉลี่ยจากตารางที่ 4.9 และตารางที่ 4.10 .	52
4.12	ความคลาดเคลื่อนของการหาโดยเมนต์ต่อเติม ในบทบรรยายภาษาต่างๆ	54
4.13	ความคลาดเคลื่อนเฉลี่ยของการหาโดยเมนต์ต่อเติม ในบทบรรยายภาษาต่างๆ	54
4.14	ผลการลับบทบรรยายออกจากอนิเมะด้วยวิธีการเขิงตัวเลขขั้นตอนวิธี 6, 11, 12 และ 13	55
4.15	ผลการซ่อมแซมภาพโดยวิธีการเขิงตัวเลขที่นำเสนอในรูปของค่าเฉลี่ยของผลที่ได้จากตารางที่ 4.14	56
4.16	ผลการลับบทบรรยายออกจากอนิเมะโดยวิธีการสปริทเบรกแม่นและวิธีการที่พัฒนาขึ้น .	56

บทที่ 1

บทนำ

1.1 บทนำของการต่อเติมภาพ

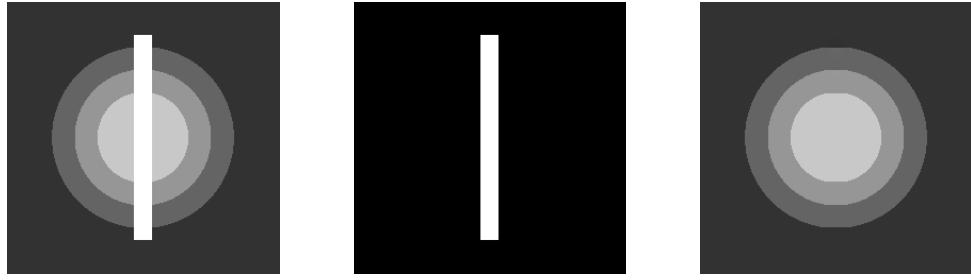
ในปัจจุบันการใช้ภาพดิจิตัล (digital images) ในสังคมเครือข่ายได้รับความนิยมอย่างแพร่หลาย เนื่องจากโทรศัพท์เคลื่อนที่มีราคาถูกลงแต่มีความสามารถที่ชาญฉลาด สามารถทำหน้าที่ได้ตั้งแต่การเป็นกล้องดิจิตัลคอมแพค (compact digital camera) คุณภาพดีให้ภาพดิจิตัลที่มีความคมชัดสูงจนไปถึงการทำหน้าที่ดังเช่น เครื่องคอมพิวเตอร์ส่วนบุคคลที่สามารถเข้มต่อ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

นอกจากภาพดิจิตัลจะได้รับจากการถ่ายภาพด้วยโทรศัพท์เคลื่อนที่แล้ว ภาพดิจิตัลยังได้รับการถ่ายภาพด้วยกล้องดีเอสแอลอาร์ หรือ กล้องสะท้อนเลนส์เดี่ยวแบบดิจิตัล (digital single lens reflex camera) กล้องโทรทรรศน์ (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) ซึ่งแสดงในรูปที่ 1.1.1 (c) ได้มาจากการ -toner วิธีการต่อเติมภาพ (inpainting algorithm) ซึ่งได้รับการออกแบบเพื่อนำข้อมูลที่ปรากฏบนภาพในบริเวณใกล้เคียงกับขอบของโดเมนต่อเติมมาซ่อมแซมภาพ



(a) ภาพที่ต้องการซ่อมแซม

(b) โดเมนต่อเติม

(c) ภาพที่ได้รับการซ่อมแซม

รูปที่ 1.1.1: ตัวอย่างการซ่อมแซมภาพ

หากที่ผู้วิจัยศึกษาและค้นคว้ามาจนถึงขณะนี้ ผู้วิจัยพบว่าการต่อเติมภาพมักนิยมนำไปใช้งานสำหรับการปรับแต่งความสวยงามของภาพบุคคลที่ถ่ายจากโทรศัพท์เคลื่อนที่ เช่น การลบร่องรอยของรอยตีนกา การลบร่องรอยแพลเป็นที่เกิดจากสิ่งเสี้ยน การลดร่องรอยของความชรา หรือ การเพิ่มความใสและความเนียนของสีผิวนะ บริเวณใบหน้าผ่านโปรแกรมแอปพลิเคชันแต่งรูปภาพที่มีอยู่ในแอปสโตร์ (App Store) หรือ ぐุเก็ลเพลย์ (Google Play) เป็นต้น

1.2 การประยุกต์ของการต่อเติมภาพ

1.2.1 การซ่อมแซมภาพจิตรกรรมไทยโบราณ

ภาพจิตรกรรมไทย คือ ภาพเขียนที่มีเอกลักษณ์ความเป็นศิลปะไทยซึ่งโดดเด่นและแตกต่างจากภาพเขียนของชนชาติอื่น ในอดีต ช่างไทยได้สร้างสรรค์ลวดลายและสีสันบนภาพวาดเพื่อสะท้อนประเพณีและวัฒนธรรมในสังคมไทยที่เกี่ยวกับศาสนา ประวัติศาสตร์ โบราณคดี ชีวิตความเป็นอยู่ วัฒนธรรมการแต่งกาย ตลอดจนการแสดงการเล่นพื้นเมืองต่าง ๆ ของแต่ละยุคสมัย

อย่างไรก็ตาม ภาพจิตรกรรมไทยจำนวนไม่น้อยที่เลื่อมลายตามกาลเวลา และรอค่อยการซ่อมแซมจากช่างในสมัยปัจจุบันที่ต้องไม่สร้างความเสียหายให้กับภาพเขียนเพิ่มขึ้นมากกว่าเดิม ที่ผ่านมาภาพที่ผ่านการ

ซ่อมแซมมาแล้วจำนวนไม่น้อยได้รับความเสียหายหลังจากการซ่อมแซม ถึงแม้สภาพโดยรวมของภาพจิตรกรรมเดิมยังคงอยู่ แต่รายละเอียดในตัวภาพเขียนได้เปลี่ยนไป ก่อให้เกิดความเสียหายที่ประเมินค่าไม่ได้

การซ่อมแซมภาพจิตรกรรมไทยโบราณโดยใช้การต่อเติมภาพเป็นขั้นตอนของการซ่อมแซมแบบหนึ่งซึ่งไม่ก่อให้เกิดความเสียหายใด ๆ กับภาพเดิม เนื่องจากเป็นการซ่อมแซมโดยการใช้ขั้นตอนวิธีเชิงตัวเลขบนภาพดิจิตัลซึ่งเป็นสำเนาของภาพเดิม ด้วยเหตุผลดังกล่าว ผู้วิจัยได้เลือกเห็นว่าการซ่อมแซมภาพจิตรกรรมไทยโบราณมีความจำเป็นเร่งด่วน เนื่องจากภาพที่ได้รับการซ่อมแซมด้วยการต่อเติมภาพสามารถนำไปใช้ประกอบการตัดสินใจเพื่อวางแผนก่อนการลงมือซ่อมแซมภาพเขียนจริงได้ นอกจากนี้ ขั้นตอนวิธีการต่อเติมภาพสามารถนำไปใช้สร้างแอปพลิเคชันบนโทรศัพท์เคลื่อนที่เพื่อใบใช้เป็นข้อมูลในการเข้าชมภาพเขียนเดิมที่ยังไม่ได้รับการซ่อมแซม และภาพเขียนที่ได้รับการซ่อมแซมโดยวิธีการทางคณิตศาสตร์จากแอปพลิเคชันที่พัฒนาขึ้น

รูปที่ 1.2.1 แสดงตัวอย่างภาพจิตรกรรมไทย¹ ที่ต้องได้รับการซ่อมแซมบนบริเวณแขนเสื้อของรูปวด ผู้ชายที่มีส่วนของสีแดงเดิมหลุดหายไป ทั้งนี้ในการซ่อมแซมภาพโดยการต่อเติมภาพ เราจะเริ่มด้วยการสร้างโฉมเนตต่อเติมบนบริเวณสีพื้นผิวปูนที่แขนเสื้อ จากนั้นจึงนำขั้นตอนวิธีการต่อเติมภาพเพื่อซ่อมแซมภาพบริเวณนั้นให้เป็นสีแดง



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 การลับบทบรรยายบนอนิเมะ

อนิเมะคือวิดีโอภาพวิดีโอการ์ตูนสแตลล์ญี่ปุ่นซึ่งเป็นที่นิยมของเยาวชนไทย ในการรับชมอนิเมะ แม้ว่า เยาวชนไทยสามารถรับชมด้วยบทพากย์เสียงภาษาไทย แต่ก็สูญเสียอรรถรสของการรับชมจากบทบรรยายแบบ เชิญ² (hardsub) ที่เป็นภาษาต่างประเทศในบริเวณด้านล่างของจอดภาพ ในการซ้อมแซม อนิเมะด้วยการลับบทบรรยายภาษาต่างประเทศจึงเป็นงานที่ยุ่งยากและท้าทายมาก เนื่องจาก

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

ด้วยความท้าทายข้างต้น การพัฒนาขั้นตอนวิธีการต่อเติมภาพที่สามารถกำหนดโดยเมนต์อัตโนมัติ ให้กับแต่ละเฟรมและประมวลผลได้แม่นยำจนการลับบทบรรยายสามารถทำงานได้แบบเรียลไทม์จึงเป็นสิ่ง จำเป็นที่หลีกเลี่ยงไม่ได้

รูปที่ 1.2.2 แสดงตัวอย่าง 1 เฟรมของอนิเมะที่มีบทบรรยายแบบ เชิญ³ ที่ต้องซ้อมแซมด้วยการลับบทบรรยายออก ทั้งนี้ในการลับบทบรรยายออกจากเฟรมโดยใช้การต่อเติมภาพ เราจะเริ่มด้วยการสร้างโดยเมนต์อัตโนมัติในบริเวณบทบรรยาย จากนั้นจึงนำขั้นตอนวิธีการต่อเติมภาพแบบเร็วเพื่อลับบทบรรยายออก จากเฟรม



รูปที่ 1.2.2: 1 เฟรมของอนิเมะที่มีบทบรรยายแบบ เชิญ

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) ศึกษาวิธีการแปรผันและขั้นตอนวิธีการเชิงตัวเลขสำหรับปัญหาการต่อเติมภาพเขตสีเทาและภาพสีในระบบ RGB
- (2) พัฒนาขั้นตอนวิธีการต่อเติมภาพชนิดใหม่สำหรับซ่อมแซมภาพจิตรกรรมไทยและลับบทบรรยายจากอนิเมะ
- (3) นำขั้นตอนวิธีที่พัฒนาขึ้นไปใช้ในการซ่อมแซมภาพจิตรกรรมไทยและลับบทบรรยายในอนิเมะ

บทที่ 2

ความรู้พื้นฐานทางคณิตศาสตร์

2.1 ปริภูมิที่มีค่าประจำ

ค่าประจำ (norm) เป็นเครื่องมือที่ใช้สำหรับบอกขนาดของเวคเตอร์หนึ่งในปริภูมิเวคเตอร์ ซึ่งค่าประจำนั้น มีนิยามดังนี้

บทนิยาม 1. (ค่าประจำ) ค่าประจำบนปริภูมิเวคเตอร์ V คือฟังก์ชันค่าจริง $|| \cdot ||$ ซึ่งนิยามบน V โดยที่

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

โครงการนวัตกรรมนี้รูปภาพแสดงเท่าเป็นฟังก์ชันที่อยู่ในปริภูมิแบบยูคลิด (Euclidean space) นั่นคือค่าประจำทั้งหมดที่พูดถึงในโครงการนวัตกรรมนี้จะเป็นค่าประจำแบบยูคลิด ซึ่งมีนิยามดังนี้

บทนิยาม 2. (ค่าประจำแบบยูคลิด) ปริภูมิยูคลิด n มิติ สามารถเขียนเวคเตอร์ในปริภูมิยูคลิดได้ว่า $\mathbf{x} = (x_1, x_2, \dots, x_n)$

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

2.2 แคลคูลัสของการแปรผันเบื้องต้น

แคลคูลัสของการแปรผัน (Calculus of variations) คือสาขาวิชาในวิชา คณิตศาสตร์วิเคราะห์ เพื่อใช้สำหรับแก้ปัญหาค่าเหมาะสม โดยจะสนใจที่จะหาฟังก์ชันที่เหมาะสมสมแพนที่จะหาค่าของตัวแปรที่เหมาะสม แคลคูลัสของการแปรผันนั้นมักจะเกี่ยวข้องกับปัญหาที่ต้องการปริมาณน้อยสุดหรือมากสุดซึ่งปรากฏอยู่ในรูปของอนุพันธ์หรือบริพันธ์ที่ไม่ทราบค่าฟังก์ชัน

เช่นฟังก์ชันหาค่าต่ำสุด มักจะมีรูปทั่วไปดังสมการ 2.2.1

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

โดยที่ $\mathcal{J} : \mathcal{U} \rightarrow \mathbb{R}$ เป็นฟังก์ชันที่ส่งจากฟังก์ชันไปยังจำนวนจริง เรียกว่า ฟังก์ชันนัล (functional) พร้อมทั้งกำหนด \mathcal{U} เป็นปริภูมิของคำตอบซึ่งประกอบด้วยฟังก์ชันค่าต่ำสุดของ \mathcal{J} และ \mathcal{V} เป็นปริภูมิทดสอบซึ่งปริภูมินี้สามารถเขียนเป็นผลต่างระหว่าง 2 ฟังก์ชันได้ นั่นคือ

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

บทนิยาม 3. (ย่านใกล้เคียง) ให้ \mathcal{U} เป็นปริภูมิคำตอบ ฟังก์ชัน $\hat{u} \in \mathcal{U}$ และ $\epsilon > 0$ แล้ว \mathcal{B}_ϵ จะเป็นย่านใกล้เคียงของ \hat{u} เมื่อ

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

จาก 2.2.1 โอลคอมินีไมเซอร์ (local minimizer) จะกำหนดโดย

บทนิยาม 4. (โอลคอมินีไมเซอร์) ให้ \mathcal{U} เป็นปริภูมิคำตอบ และฟังก์ชันนัล $\mathcal{J} : \mathcal{U} \rightarrow \mathbb{R}$ จะเรียก $\hat{u} \in \mathcal{U}$ ว่า โอลคอมินีไมเซอร์ของ \mathcal{J} ถ้าทุก $\epsilon > 0$ มี $\delta > 0$ โดยที่ $\mathcal{J}(\hat{u}) \leq \mathcal{J}(u)$ สำหรับทุก $u \in \mathcal{B}_\epsilon(\hat{u})$

ในการนิยามเงื่อนไขที่จำเป็นสำหรับโอลคอมินีไมเซอร์ของ \mathcal{J} จำเป็นต้องมีการกำหนดการหาอนุพันธ์แบบมีทิศทาง

บทนิยาม 5. (*Gâteaus-differentiable*) \mathcal{U} เป็นปริภูมิคำตอบ \mathcal{V} เป็นปริภูมิทดสอบ และฟังก์ชันนัล $\mathcal{J} : \mathcal{U} \rightarrow \mathbb{R}$ และ \mathcal{J} เป็น *Gâteaus-differentiable* เมื่อทุก $u \in \mathcal{U}$ อยู่ในทิศทางของ $v \in \mathcal{V}$ ถ้า

- มีจำนวน $\hat{\epsilon} > 0$ ซึ่งทำให้ $u_\epsilon = u + \epsilon v \in \mathcal{U}$ สำหรับทุก $|v| \leq \hat{\epsilon}$

2. พังค์ชัน $J(\epsilon) = \mathcal{J}(u_\epsilon)$

โดยอนุพันธ์อันดับแรกของ Gateaus หรือการแปรผันอันดับแรก (first variation) ของ \mathcal{J} สำหรับ u ที่อยู่ในทิศทางของ v กำหนดโดย

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

และนิยามจุดคงตัว (stationary point) โดย

บทนิยาม 6. (จุดคงตัว) U เป็นปริภูมิคำตอบ V เป็นปริภูมิทดสอบ และพังค์ชันนัล $\mathcal{J} : U \rightarrow \mathbb{R}$ สมมติให้ $u \in U$ และ \mathcal{J} เป็น Gateaus-differentiable สำหรับทุกพังค์ชันทดสอบ $v \in V$ และ u จะเรียกว่าจุดคงตัวของ \mathcal{J} ก็ต่อเมื่อ $\delta \mathcal{J}(u; v) = 0$ สำหรับทุก $v \in V$

เงื่อนไขที่จำเป็นสำหรับมินิไมเซอร์นั้นสามารถสร้างได้จากการใช้จุดคงตัว

ทฤษฎีบท 2.1. (เงื่อนไขที่จำเป็นสำหรับโลกอลมินิไมเซอร์) ให้ U เป็นปริภูมิคำตอบ ซึ่ง $u \in U$, พังค์ชันนัล $\mathcal{J} : U \rightarrow \mathbb{R}$, V เป็นปริภูมิทดสอบโดยที่ v เป็นพังค์ชันทดสอบซึ่ง $v \in V$ และ \mathcal{J} เป็น Gateaus-differentiable สำหรับทุก u

ถ้า u เป็นโลกอลมินิไมเซอร์ของ \mathcal{J} และ u เป็นจุดคงตัวของ \mathcal{J}

สำหรับทพิสูจน์จะพบได้ใน [9]

ด้วยทฤษฎีบทนี้เราจะศึกษาเงื่อนไขของจุดคงตัวเพิ่มเติมโดยเราเลือกพังค์ชันนัลทั่วไป \mathcal{J} ซึ่งนิยามโดย

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

โดยที่ $\Omega \subset \mathbb{R}^d$, $d > 1$ เป็นเซ็ตเปิดมีขอบเขตและ F เป็นพังค์ชันนัลที่ขึ้นอยู่กับ $x = (x_1, x_2, \dots, X_d)^\top$ สมมติให้ \mathcal{J} เป็น Gateaus-differentiable ในทุกทิศทางของปริภูมิทดสอบ ดังนั้นจึงสมมุติได้ว่า F เป็นอนุพันธ์ย่อยที่ต่อเนื่อง

ก่อนที่จะแนะนำเงื่อนไขสำหรับจุดคงตัวของ \mathcal{J} จะแนะนำ

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

สำหรับเกรเดียนซ์ของ F ซึ่งขึ้นอยู่กับ u จะกำหนดโดย

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

ในทำนองเดียวกันเกรเดียนซ์ของ F ซึ่งขึ้นอยู่กับ ∇u กำหนดโดย

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

โดยในขั้นนี้เราจะเลือกคำตอบที่จะใช้ในการเพิ่มเงื่อนไขค่าขอบเข้าไป ตัวอย่างเช่น

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

และเช่นเดียวกันปริภูมิทดสอบจะถูกกำหนดโดย

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

อีกทั้งขั้นตอนที่กล่าวมาไม่เพียงครอบคลุมปริภูมิทั่วไป \mathcal{U} และ \mathcal{V} แต่ยังครอบคลุมไปถึงกรณีที่เป็นเวคเตอร์เมื่อ $\mathbf{u} = (u_1, u_2, \dots, u_d)^{\top} : \mathbb{R}^d \rightarrow \mathbb{R}^d$

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

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

บทพิสูจน์ของทฤษฎีบทประกอบนี้สามารถหาได้จาก [10]

เห็นได้ชัดว่า 2.2.9 เป็นจริงสำหรับพังก์ชันทดสอบคงตัวที่ค่าไม่เจาะจง เพราะถ้า $\nabla_u F - \nabla \cdot \nabla_{\nabla u} F$ การยืนยันทฤษฎีบทนี้ เป็นทฤษฎีบทที่รู้กันดีอยู่แล้ว ดังนั้น $u \in \tilde{\mathcal{U}}$ เป็นจุดคงตัวของพังก์ชันนัล \mathcal{J} ซึ่งเป็น Gâteaus-differentiable ถ้า

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

โดยการใช้ทฤษฎีบท 2.1 กับ 2.2.10 จะได้เงื่อนไขที่จำเป็นสำหรับโอลิมินีไซเซอร์ของ 2.2.1 จาก $d > 1$ และ 2.2.10 จะนำไปสู่สมการเชิงอนุพันธ์อย่าง ซึ่งรู้จักกันในชื่อของสมการอยเลอร์-ลากราจ ที่มีเงื่อนไขค่าขอบ โดยจะเรียก 2.2.1 ที่มีเงื่อนไขค่าขอบว่า รูปแบบการแปรผัน (variational formulation) และถ้าเงื่อนไขค่าขอบนั้นถูกกำหนดไว้ชัดเจนตามทฤษฎีบทประกอบที่ 2.2.1 จะเรียกเงื่อนไขนี้ว่า เงื่อนไขจำเป็น (essential condition) และในทางกลับกันหากค่าขอบไม่ถูกกำหนดไว้ชัดเจนจะเรียกว่า เงื่อนไขธรรมชาติ (natural condition)

โดยสรุปแล้วทุกคำตอบ $u* \in \mathcal{U}$ ในปัญหาค่าหมายสมทั่วไปดังเช่น 2.2.1 ที่มีพังก์ชันนัล \mathcal{J} ซึ่งเป็น Gâteaus-differentiable ที่ถูกกำหนดโดย 2.2.3 จะเป็นคำตอบของปัญหาค่าขอบซึ่งเป็นส่วนประกอบของสมการอยเลอร์-ลากราจ

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

ส่วนเงื่อนไขค่าขอบนั้นสามารถเป็นแบบเงื่อนไขจำเป็น หรือเงื่อนไขธรรมชาติก็ได้

$$\langle \nabla_{\nabla u} F, n \rangle_{\mathbb{R}^d} = 0 \text{ บน } \partial\Omega$$

เมื่อ $n = (n_1, \dots, n_d)^\top$ หมายถึงเวกเตอร์หนึ่งหน่วยปกติภายนอก (outer normal vector unit) ของ $\partial\Omega$

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

$$u_{min} \int_{\Omega} |\nabla u| dx$$

กำหนดให้

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

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

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

ดังนี้

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

โดยเอกสารนี้อันดับหนึ่งของกรีน จะได้ว่า

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

เพราะฉะนั้น

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

เมื่อ \mathbf{n} แทนเวกเตอร์หนึ่งหน่วยที่ดึงออกจากขอบของภาพในทิศทางซึ่งออก
จะได้สมการอย่างเดอร์ที่สมนัยกับปัญหานี้คือ

$$\begin{aligned} -\Delta u &= 0 \text{ ใน } \Omega \\ \frac{\partial u}{\partial \mathbf{n}} &= 0 \text{ ใน } \partial\Omega \end{aligned}$$

2.3 พังค์ชันของการแปรผันที่มีขอบเขต

ให้ Ω เป็นเซตเปิดมีขอบเขตของ \mathbb{R}^d และให้ $u \in L^1(\Omega)$ กำหนดให้การแปรผันรวม u เป็น

$$\int_{\Omega} |Du| = \sup \left\{ \int_{\Omega} u \nabla \cdot \varphi \right\} \quad (2.3.1)$$

เมื่อ เป็น (Lebesgue measure) และ $C_0^1(\Omega, \mathbb{R}^d)$ คือปริภูมิของฟังค์ชันต่อเนื่องที่หาอนุพันธ์ได้และกระซับใน Ω

ตามที่ได้ถูกกล่าวถึงใน [11] สำหรับกรณีเฉพาะซึ่งเป็นที่น่าสนใจ $u \in C^1(\Omega, \mathbb{R}^d)$ โดยการใช้ปริพันธ์แบบแยกส่วน

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

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

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

ฟังค์ชัน $u \in L^1(\Omega)$ เรียกว่ามีขอบเขตการแปรผันใน Ω ถ้า $\int_{\Omega} |Du| < \infty$ โดยเรากำหนดให้ $BV(\Omega)$ เป็น
ปริภูมิของฟังค์ชันที่จะหมดใน $L^1(\Omega)$ การแปรผันที่มีขอบเขต

ตัวอย่าง 2.3.1. ฟังค์ชัน f_1, f_2 และ f_3 ต่อไปนี้กำหนดโดย

$$f_1(x) = \sin x, \quad (2.3.4)$$

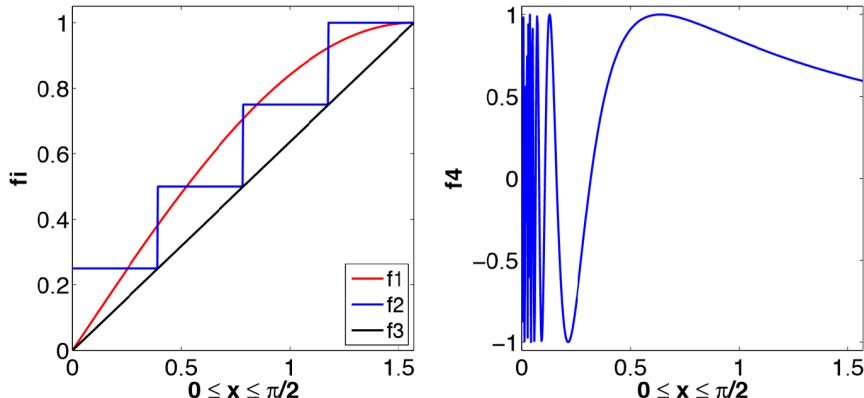
$$f2(x) = \begin{cases} 1/4, & x \in [0, \Pi/8] \\ 1/2, & x \in [\Pi/8, \Pi/4] \\ 3/4, & x \in [\Pi/4, 3\Pi/8] \\ 1, & x \in [3\Pi/8, \Pi/2] \end{cases} \quad (2.3.5)$$

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

จาก $BV(\Omega)$ ซึ่ง $\Omega = [0, \Pi/2]$ และมีการแปรผันรวมมีค่าเป็น 1 ให้ฟังก์ชัน $f4$ กำหนดโดย

$$f4(x) = \begin{cases} 0, & x = 0 \\ \sin 1/x, & x \in (0, a) \text{ และ } a > 0 \end{cases} \quad (2.3.7)$$

มีการแปรผันไม่จำกัดและไม่ออยู่ใน $BV(\Omega)$ ซึ่ง $\Omega = [0, a]$ สำหรับทุก $a > 0$



รูปที่ 2.3.1: ฟังก์ชันแปรผันมีขอบเขตทั้งสามฟังก์ชันที่มีการแปรผันรวมเหมือนกันเท่ากับ 1 และฟังก์ชันที่มีการแปรผันไม่จำกัด

ซึ่งสำหรับในหัวข้อนี้เรารสามารถสรุปได้เป็นสูตรของคลอเลีย (Coarea formula)

ทฤษฎีบท 2.2. (*Coarea formula*) ให้ $\Omega \subset \mathbb{R}^d$ เป็นเซตเปิดและให้ $u \in BV(\Omega)$ และ $L_\lambda = \{x \in \Omega | u(x) < \lambda\}$ เป็นระดับโดเมน (*level domain*) และ

$$\int_{\Omega} |Du| = \int_{-\infty}^{\infty} Per(L_\lambda, \Omega) d\lambda$$

เมื่อ $Per(L_\lambda, \Omega) = \int_{\Omega} |D_{x^{L_\lambda}}|$ คือ perimeter ของ L_λ ใน Ω และ χ^{L_λ} คือลักษณะเฉพาะ (characteristic) ของฟังก์ชัน L_λ

โดยบนพิสูจน์สามารถดูได้ใน [11]

2.4 วิธีการเร็กกิวลาร์เลช์เซ็น

ในโครงการวิจัยเรื่องนี้ จะพบปัญหา Ill-posed ซึ่งทำให้การแก้ปัญหานั้นเกิดความยากลำบากขึ้น จึงจำเป็นต้องหาวิธีแก้ปัญหา Ill-posed ให้เป็นปัญหา Well-posed ก่อนนำไปแก้ปัญหาต่อไป

2.4.1 ปัญหา Well-posed และปัญหา Ill-posed

บทนิยาม 7. (ปัญหา Well-posed) จะเรียกปัญหาต่อไปนี้ว่าเป็นปัญหา Well-posed เมื่อปัญหามีคุณสมบัติดังนี้

1. มีคำตอบ
2. มีเพียงคำตอบเดียว
3. คำตอบขึ้นอยู่กับความต่อเนื่อง

หากปัญหาไม่มีคุณสมบัติจากทั้ง 3 ข้อ จะเรียกปัญหาดังกล่าวว่า ปัญหา Ill-posed

2.4.2 ปัญหาย้อนกลับ

ปัญหาย้อนกลับ (Inverse problem) คือปัญหาสำหรับการถูคืนข้อมูลพารามิเตอร์จากตัวแบบทางคณิตศาสตร์ โดยใช้ข้อมูลบางพารามิเตอร์ที่ทราบค่าอยู่ ซึ่งโดยทั่วไปแล้วปัญหาย้อนกลับนี้มักจะเป็นปัญหา Ill-posed

ตัวอย่าง 2.4.1. สำหรับตัวอย่างปัญหาย้อนกลับ เช่น จงหาค่า x และ y ที่ทำให้ $x + y = 5$ จะเห็นว่ามีชุดของคำตอบ $x + y = 5$ อยู่มากมาย ซึ่งทำให้เป็นปัญหา Ill-posed เนื่องจากคำตอบของปัญหาไม่ได้มีเพียงชุดเดียว

ในส่วนของการต่อเติมภาพที่เสียหายนั้นเป็นปัญหาย้อนกลับและเป็นปัญหา Ill-posed ด้วยเนื่องจากคำตอบในบริเวณที่จะต่อเติมไม่ได้มีเพียงคำตอบเดียว ตัวอย่างเช่น ภาพที่ 2.4.1 ภาพช้างที่เสียหาย¹ ในบริเวณภาพสีแดงซึ่งภาพเกิดความเสียหายขึ้น อาจจะมีคำตอบเป็นขาของช้าง หรือมีคำตอบเป็นสิงโตดังในภาพก็ได้

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



รูปที่ 2.4.1: ตัวอย่างการต่อเติมภาพที่ไม่มีคำตอบเฉพาะเจาะจง

ตัวอย่าง 2.4.2. ตัวอย่างปัญหาข้อนกลับ เมื่อ z เป็นภาพซึ่ง นิยามอยู่ใน $\Omega \in \mathbb{R}^2$ η คือสัญญาณรบกวนแบบ เกาล์เชียนที่มีส่วนเบี่ยงเบนมาตรฐานเป็น σ^2 และ n คือภาพที่มีสัญญาณรบกวน โดยที่ $u = z + \eta$ เรา สามารถนำสัญญาณรบกวนออกได้โดยหา n ที่เหมาะสมจาก

$$u_{\min} \left\{ \left| \int_{\Omega} |u - z|^2 d\Omega - \sigma^2 \right| \right\} \quad (2.4.1)$$

ซึ่ง n ที่เหมาะสมมีหลายคำตอบจึงได้ว่า n นี้เป็นปัญหา *ill-posed*

2.4.3 เร็กกิวลาไลซ์เซชัน

วิธีเร็กกิวลาไลซ์เซชัน (Regularization) เป็นวิธีการทำให้ปัญหาข้อนกลับกลายเป็นปัญหา Well-posed ได้โดยคุณ Tikhonov และคุณ Arsenin [8] ได้นำเสนอวิธีการสำหรับจัดการปัญหาค่าเหมาะสมสมโดยใช้การแนะนำ วิธีการแก้ปัญหานี้โดยการทำให้ปัญหามีคำตอบอยู่ในชุดของคำตอบใด คำตอบหนึ่ง หรือทำให้มีคุณลักษณะที่เฉพาะเจาะจง

จากตัวอย่าง 2.4.1 สามารถทำให้คำตอบเฉพาะเจาะจงขึ้นได้ โดยการเพิ่มเงื่อนไขเข้าไปว่า $x + y = 5$ เมื่อ $\sqrt{x^2 + y^2}$ มีค่าน้อยที่สุด

จากตัวอย่าง 2.4.2 เราสามารถทำให้คำตอบเฉพาะเจาะจงขึ้นได้โดยการเพิ่มพจน์เข้าไปดังนี้

$$u_{\min} \left\{ \left| \int_{\Omega} |u - z|^2 d\Omega - \sigma^2 \right| + \alpha \int_{\Omega} |\nabla u|^2 \right\} \quad (2.4.2)$$

โดยจะเรียกพจน์แรกว่าพจน์ปรับค่าข้อมูล (Data fitting term) และพจน์ที่ 2 ว่าพจน์เริกกิวลาไลซ์เชชัน (Regularization term) โดยเมื่อคำตอบ n มีค่าเกรดียนต์ที่น้อยแล้วจะได้ผลลัพธ์ออกแบบเป็นภาพที่ไม่มีสัญญาณรบกวน

2.5 วิธีการไฟไนต์ดิฟเฟอเรนซ์เบื้องต้น

วิธีการไฟไนต์ดิฟเฟอเรนซ์ (finite difference method) ถูกคิดค้นโดย เป็นวิธีการเชิงตัวเลขที่พัฒนาขึ้นเพื่อแก้ไขปัญหาค่าขอบ ซึ่งทั่วไปแล้วขั้นตอนของวิธีการไฟไนต์ดิฟเฟอเรนซ์สำหรับการแก้ปัญหาค่าขอบประกอบด้วยสามขั้นตอนสำคัญดังนี้

1. ตัดสครีตไทช์ (discretize) โดเมนของผลเฉลย (solution domain) ออกเป็นช่องตาราง (mesh) ของจุดกริด (grid point) ที่ต้องการหาผลเฉลยเชิงตัวเลข
2. ประมาณอนุพันธ์ที่ปรากฏในปัญหาค่าขอบด้วยการประมาณแบบไฟไนต์ดิฟเฟอเรนซ์ (finite difference approximation) ในขั้นตอนนี้ การประมาณดังกล่าวจะนำไปสู่ระบบสมการเชิงเส้น หรือระบบสมการไม่เป็นเชิงเส้นขนาดใหญ่ที่สมนัยกับปัญหาค่าขอบตั้งต้น
3. แก้ระบบสมการเชิงเส้นหรือระบบสมการไม่เป็นเชิงเส้นขนาดใหญ่ที่เกิดขึ้นในขั้นตอนที่ 2 เพื่อกำหนดผลเฉลยเชิงตัวเลข

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

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

1. สูตรฟอร์เวิร์ดดิฟเฟอเรนซ์ (forward-difference formula)

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

2. สูตรແບ່ນວິເວີຣດິຟເພື່ອເຮັນ (backward-difference formula)

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

3. สูตรເຂົ້າທັລດິຟເພື່ອເຮັນ (forward-difference formula)

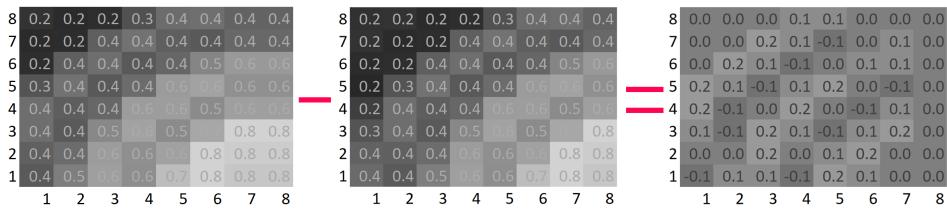
$$D_0u(\bar{x}) = \frac{u(\bar{x} + h) - u(\bar{x} - h)}{2h}$$

โดยที่ h เป็นຈຳນວນຈົງທີ່ມີຄ່ານ້ອຍໆ ຈຶ່ງ $h > 0$

ສໍາຫຼັບໂຄງງານວິຈິຍນີ້ ຈະໃຊ້ວິທີກາຣຫາອນຸພັນຮູໂດຍປະມານໄຟໄຟຕົວຢ່າງເພື່ອເຮັນຈົ່ງໄດ້ວ່າກາຣຫາອນຸພັນຮູຂອງຄ່າຄວາມເຂັ້ມທີ່ພິກັດທາງກາຍກາພເປັນ (i, j) ສາມາດທາໄດ້ຕັ້ງນີ້

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

ເນື່ອຮະບບກຣິດທີ່ໃໝ່ນກາພມີຄວາມທ່າງເພີ່ມທີ່ນີ້ຫຸ້ນໄວ່ ຈຶ່ງໄດ້ວ່າ $h = 1$ ທັງນີ້ຮະຍະທ່າງ h ຈາກເປີ່ມຢັນໄປຕາມຫັ້ນຂອງພື້ນມືດຽບກາພ



ຮູບທີ 2.5.1: ຕ້າວຍ່າງກາຣຫາອນຸພັນຮູບກາພເຂດເທາ

ຈາກກາພ 3.1.1 ເນື່ອຕ້ອງກາຣຫາອນຸພັນຮູເຖິບແກນ \times ຈະທຳຕາມກາພທີ່ 2.5.1 ໂດຍທຳກາຣສ້າງກາພຈຶ່ງທຳກາຣຕັດຂອບທາງໜ້າຍອອກໜີ່ຄົກລົມມົງແລ້ວເພີ່ມຂອບທາງຂວາໜີ່ຄົກລົມມົງໄດ້ໃໝ່ເຈື່ອນໄຂຄ່າຂອບແບບນິວແມນ ຈາກນັ້ນກາພທີ່ສ້າງຂຶ້ນໄປລົບກັບກາພເດີມຈະໄດ້ອນຸພັນຮູຂອງກາພນັ້ນດັ່ງທີ່ປ່ຽກງູ້ທາງຂວາ ທັງນີ້ກ່າວ $h \neq 1$ ສາມາດທຳກາຣກາພຜລລັບຮູດວ່າ h ໄດ້ເພື່ອໃຫ້ໄດ້ຄ່າທີ່ຕ້ອງກາຣ

ສໍາຫຼັບກາຣຫາເກຣເດີຍນີ້ (Gradient) ຈະໃຊ້ກາຣຫາອນຸພັນຮູໂດຍວິທີກົດຝົດິຟເພື່ອເຮັນຈົ່ງທີ່ກ່າວໄປໃນໜ້ານແນວແກນ x ແລະ ແນວແກນ y ດຳຕອບທີ່ໄດ້ຈະເປັນເວັບເຕອຮ່ອຂອງອນຸພັນຮູແນວແກນ x ແລະ ອນຸພັນຮູແນວແກນ y ໄດ້ເວັບເຕອຮ່ອດັ່ງນີ້

$$\nabla \vec{v}_{u_i} = \left(\frac{\partial}{\partial x} u_{i,j}, \frac{\partial}{\partial y} u_{i,j} \right)^{\top}$$

สำหรับไดเวอร์เจน (Divergence) จะเป็นการหาผลรวมของอนุพันธ์ในแต่ละแกนของเวกเตอร์ด้วยวิธีฟอร์เวิร์ดดิฟเพอร์เซนต์ นั่นคือ

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

สำหรับลาป้าเชียน (Lapacian) นั่นคือการทำหาไดเวอร์เจนบนเวกเตอร์ที่หากรเดย์นแล้ว แต่ทั้งนี้สามารถหาลาป้าเชียนได้จาก

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

2.6 วิธีการทำขั้นตอนสำหรับระบบสมการเชิงเส้น

ในหัวข้อนี้จะแนะนำถึงวิธีการทำขั้นตอนสำหรับแก้ปัญหาระบบสมการเชิงเส้น สำหรับโครงงานวิจัยเรื่องนี้มีการแก้ปัญหาของระบบสมการเชิงเส้น เพื่อทำการแก้สมการเชิงอนุพันธ์ที่ถูกเปลี่ยนให้เป็นระบบสมการเชิงเส้นแล้ว โดยระบบสมการเชิงเส้นที่จะทำการแก้ก็คือในรูปของ

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

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

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

เมตริกซ์ความสัมพันธ์ T และเวกเตอร์ c มาจากการแบ่ง $A = M - N$ ของเมตริกซ์ A เมื่อ M เป็นเมตริกซ์ไม่เอกฐาน โดยแยกระบบเดิม 2.6.1 ออกเป็น

$$Ax = (M - N)x = b \quad (2.6.3)$$

นั่นคือ

$$x = (M^{-1}N)x + M^{-1}b = Tx + c \quad (2.6.4)$$

เมื่อ $T = M^{-1}N$ และ $c = M^{-1}b$ โครงงานวิจัยนี้ทางผู้จัดได้เลือกใช้วิธีเกาส์-ไซเดลซึ่งเป็นวิธีการที่พัฒนาต่อมาจากวิธีการจาโคปี จึงขอนำเสนอทั้งสองวิธีการ ดังนี้

2.6.1 วิธีการ Jacobi

วิธีการ Jacobi จะแก้สมการที่ i ของ $Ax = b$ โดยหา x_i ซึ่งกำหนดโดย

$$x_i = \sum_{\substack{j=1 \\ j \neq i}}^N \left(\frac{-a_{ij}x_j}{a_{ii}} \right) + \frac{b_i}{a_{ii}} \quad \text{เมื่อ } i = 1, \dots, N \quad (2.6.5)$$

ให้ $x^{(k-1)}$ สำหรับทุก $k \geq 1$ ซึ่งก่อทำนิดโดย

$$x_i^{(k)} = \sum_{j=1}^N \left(\frac{-a_{ij}x_j^{(k-1)}}{a_{ii}} \right) + \frac{b_i}{a_{ii}} \quad \text{เมื่อ } i = 1, \dots, N \quad (2.6.6)$$

ซึ่งจำเป็นที่ $a_{ii} \neq 0$ สำหรับ $i = 1, \dots, N$ แต่ถ้ามีอย่างน้อยหนึ่ง $a_{ii} = 0$ และระบบไม่เอกฐาน ก็สามารถสับเปลี่ยนลำดับเพื่อให้มี a_{ii} ที่เป็น 0 ได้ และการเขียน $Ax = b$ เป็น $x = Tx + c$ จะทำการเปลี่ยน A เป็น $A = D - L - U$ เมื่อ D เป็นเมตริกซ์泰ยงมุขของ A , $-L$ เป็นสามเหลี่ยมส่วนล่างของ A และ $-U$ เป็นสามเหลี่ยมส่วนบนของ A จึงได้ว่า

$$Ax = (D - L - U)x = b \quad (2.6.7)$$

หรือ

$$x = D^{-1}(L + U)x + D^{-1}b \quad (2.6.8)$$

เมื่อทำการแบ่งเมตริกซ์เป็น $A = M - N$ โดยที่ $M = D$ และ $N = L + U$ แล้วจะได้ว่าเมตริกซ์สำหรับวิธีการ Jacobi คือ

$$x^{(k)} = T_J x^{(k-1)} + c_J \quad (2.6.9)$$

เมื่อ $T_J = D^{-1}(L + U)$ และ $c_J = D^{-1}b$

ขั้นตอนวิธี Jacobi

ขั้นตอนวิธีของ Jacobi เพื่อหาค่าไกล์เคียงของคำตอบ $Ax = b$ จะให้ค่าไกล์เคียงของคำตอบเริ่มต้นเป็น $x^{(0)}$ ให้

จำนวนรอบการทำซ้ำสูงสุดเป็น $IMAX$ และให้ค่าความคลาดเคลื่อนเป็น $\epsilon > 0$

Algorithm 1: ขั้นตอนวิธีจากोบี

```

 $[x] \leftarrow Jacobi(A, b, x^{(0)}, IMAX, \epsilon)$ 
1. ให้  $k = 1, N = size(x^{(0)})$ , done = False
2. ถ้า done = False ทำซ้ำขั้นตอนที่ 3 และ 4
3.  $x_i^{(k)} = \sum_{j=1}^{k-1} \left( \frac{-a_{ij}x_j^{(k-1)}}{a_{ii}} + \frac{b_i}{a_{ii}} \right)$ 
4. if  $\|b - Ax^{(k)}\| < \epsilon$  หรือ  $\|x^{(k)} - x^{(k-1)}\| < \epsilon$  หรือ  $k \geq \epsilon$  then
   ให้ done = True และ  $x = x^{(k)}$ 
else
   ให้  $k = k + 1$ 
end

```

2.6.2 วิธีการเกาส์-ไซเดล

จากวิธีการจากอบีมีคำนวนค่า $x_1^{(k)}, x_2^{(k)}, \dots, x_{i-1}^{(k)}$ ซึ่งสามารถเพิ่มประสิทธิภาพได้ด้วยการเปลี่ยนสมการของ $x_i^{(k)}$ เป็น

$$x_i^{(k)} = \frac{-\sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^N a_{ij}x_j^{(k-1)} + b_i}{a_{ii}} \quad (2.6.10)$$

วิธีการนี้เรียกว่า วิธีการเกาส์-ไซเดล ซึ่งสามารถเขียนสมการใหม่ได้เป็น

$$a_{ii}x_i^{(k)} + \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} = -\sum_{j=i+1}^N a_{ij}x_j^{(k-1)} + b_i \quad (2.6.11)$$

จะได้รูปแบบเมทริกซ์ของวิธีเกาส์-ไซเดลเป็น

$$(D - L)x^{(k)} = Ux^{(k-1)} + b \quad (2.6.12)$$

ซึ่งสมมูลกับ

$$x^{(k)} = T_{GS}x^{(k-1)} + c_{GS} \quad (2.6.13)$$

เมื่อ $T_{GS} = (D - L)^{-1}U$ และ $c_{GS} = (D - L)^{-1}b$ นั้นคือเกาส์-ไซเดล มีพื้นฐานมาจากการแยกเมทริกซ์ ด้วย $M = D - L$ และ $N = U$

ขั้นตอนวิธีเกาส์-ไซเดล

ขั้นตอนวิธีเกาส์-ไซเดล เมื่อันกับขั้นตอนวิธีจากอบี แต่เปลี่ยนขั้นที่ 3 เป็นสมการที่ 2.6.10

2.7 การวัดคุณภาพของภาพที่ผ่านกระบวนการต่อเติม

การประเมินคุณภาพของการต่อเติมภาพของวิธีการเชิงตัวเลข โดยการวัดค่าคุณภาพของภาพ ในโครงงานวิจัยนี้จะใช้ 2 ค่าคือ จะใช้ค่า Peak Signal Noise Ratio (PSNR) [6] และ Structural Similarity (SSIM) [7]

2.7.1 Peak Signal Noise Ratio

ค่า PSNR ใช้สำหรับวัดค่าคุณภาพของแต่ละพิกเซลในภาพ มีหน่วยเป็น เดซิเบล (dB) ซึ่งมีช่วงค่าอยู่ในระหว่าง 0 ถึง อนันต์ โดยค่า PSNR ที่ได้ยิ่งเข้าใกล้สูронั่นแปลว่าคุณภาพที่ได้ในแต่ละพิกเซลนั้นยิ่งมีคุณภาพที่ดีซึ่งสามารถคำนวณค่า PSNR ได้ดังนี้

$$\text{PSNR} = 10 \cdot \log_{10}\left(\frac{1}{\sqrt{\text{MSE}}}\right) \quad (2.7.1)$$

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

2.7.2 Structural Similarity

ค่า SSIM ใช้สำหรับวัดคุณภาพของเค้าโครงในภาพผลลัพธ์ว่ามีเค้าโครงคล้ายกับภาพของเดิมมากเพียงใด โดยค่า SSIM จะมีค่าอยู่ระหว่าง 0 ถึง 1 หากค่าที่ได้ยิ่งเข้าใกล้สูง 1 นั้นคือเค้าโครงใกล้เคียงกับภาพเดิมก่อนทำการต่อเติมภาพ ซึ่งสามารถคำนวณ SSIM ได้ดังนี้

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

- u แทนภาพต้นฉบับ
- \tilde{u} แทนภาพต้นฉบับ และภาพที่ได้จากการซ่อมแซมโดยวิธีเชิงตัวเลข
- μ_u คือค่าเฉลี่ยของ u

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

บทที่ 3

ขั้นตอนวิธีเชิงตัวเลขชนิดใหม่

ในโครงงานวิจัยนี้เป็นการพัฒนาขั้นตอนวิธีเชิงตัวเลขชนิดใหม่สำหรับแก้ปัญหาตัวแบบคณิตศาสตร์ โดยมีพื้นฐานมาจากขั้นตอนเชิงตัวเลขที่เคยมีมาก่อนหน้า โดยขั้นตอนวิธีเชิงตัวเลขที่ใช้ในโครงงานวิจัยดังนี้ ทั้งขั้นตอนเชิงตัวเลขที่มีอยู่เดิม และขั้นตอนเชิงตัวเลขที่ได้พัฒนาขึ้นใหม่ สามารถอธิบายโดยสังเขปได้ดังนี้

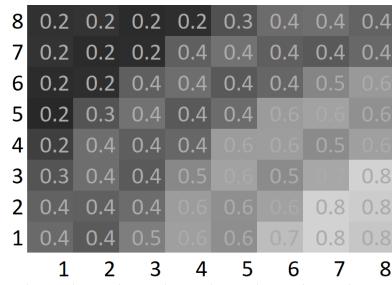
3.1 การนำเสนอภาพ

3.1.1 การนำเสนอภาพเขตเทา

สำหรับภาพถ่ายสามารถพิจารณาภาพเป็นพังก์ชันได้ดังนี้

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

เป็นพังก์ชันต่อเนื่อง 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 u ที่นิยามข้างต้นว่าภาพเขตเทา (grayscale image)



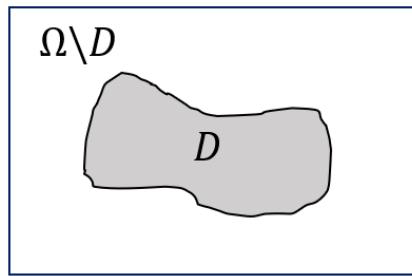
รูปที่ 3.1.1: ตัวอย่างภาพเขตเทาที่แสดงระดับความเข้มของภาพในแต่ละระดับ

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

3.1.2 การต่อเติมภาพเขตเทา

ในโครงการวิจัยชิ้นนี้สำหรับการต่อเติมภาพ จะทำการหาคำตอบทองภาพที่อยู่ในโดเมนต่อเติมภาพ 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 $u : \Omega \rightarrow V$, $z : \Omega \rightarrow V$ แทนภาพที่ได้รับการซ้อมแซมและภาพที่ต้องการซ้อมแซมตามลำดับ

รูปที่ 3.1.2: D แทนโดเมนต่อเติม

การต่อเติมภาพเขตเทาจะเป็นการทำคำตอบทองพื้นที่ได้รับความเสียหายที่อยู่บนภาพ z ซึ่งเป็นบริเวณในโดเมนต่อเติม D โดยใช้ข้อมูลที่มีอยู่ใน $\Omega \setminus D$ เพื่อหาข้อมูลใน D ที่ได้รับความเสียหายเป็นคำตอบที่ในภาพ u

3.1.3 การนำเสนอภาพสี

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

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

เมื่อ $u_1, u_2, u_3 : \Omega \rightarrow V$ แทนภาพในเนడสีแดง สีเขียว และสีน้ำเงินของ u ซึ่งการต่อเติมภาพสีที่พูดถึงในโครงงานวิจัยนี้จะทำการแยกแต่ละเนడสีออกเป็นเนడเทา 3 ระนาบ แล้วจึงใช้การต่อเติมภาพเนಡเทากับทั้ง 3 เนดสีก่อนรวมกลับเป็นภาพสีอีกรั้ง

3.2 ตัวแบบเชิงแปรผันสำหรับต่อเติมภาพเฉดเทา

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

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

୧୮

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

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

โดยแคลคูลัสของการแปรผัน (Calculus of variations) จะได้สมการออยเลอร์-ลากրองจ์ที่เกี่ยวข้องกับ (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 แทนเวลาเตอร์หน่วยที่ตั้งจากกับของของภาพ

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

ในการคำนวณเดียวกันกับตัวแบบการต่อเติมภาพเฉลี่ยวที่ใช้การแปรผันรวม 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 ขั้นตอนวิธีเดินเวลา (explicit time marching method)

คณะวิจัย [2] ได้แนะนำวิธีการเชิงตัวเลขสำหรับการคำนวณโดยใช้วิธีการเดินเวลาแบบชั้ดแจ้ง ซึ่งวิธีการเชิงตัวเลขดังกล่าวสามารถประยุกต์ใช้เพื่อแก้สมการ 3.2.3 สามารถประยุกต์เป็นวิธีเชิงตัวเลขสำหรับการต่อเติมภาพได้ดังนี้

เริ่มจากการแนะนำตัวแปรเวลาสังเคราะห์ (time artificial variable) จากนั้นหาคำตอบแบบสภาวะคงตัว (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

หลังจากใช้การประมาณแบบไฟโนต์ดิฟเฟอร์เรนซ์ดังที่กล่าวในตอนต้นนี้จะได้รูปแบบการคำนวณเป็น

$$(u^{[k+1]})_{i,j} = (u^{[k]})_{i,j} + \tau (\mathcal{K}(u^{[k]})_{i,j} - (\lambda)_{i,j} ((u^{[k]})_{i,j} + (z)_{i,j})) \quad (3.4.2)$$

โดยที่

$$\begin{aligned} \mathcal{K}(u)_{i,j} &= [\partial_x^- (D(u)_{i,j} \partial_x^+(u)_{i,j}) + \partial_y^- (D(u)_{i,j} \partial_y^+(u)_{i,j})] \\ &= ((\Sigma)_{i,j}(u)_{i,j} - (\bar{\Sigma})_{i,j}(u)_{i,j}) \\ (\Sigma)_{i,j}(u)_{i,j} &= (D_1(u)_{i,j} + D_2(u)_{i,j} + 2D_3(u)_{i,j}) (u)_{i,j} \\ (\bar{\Sigma})_{i,j}(u)_{i,j} &= D_1(u)_{i-1,j}(u)_{i-1,j} + D_2(u)_{i,j-1}(u)_{i,j-1} \\ &\quad + D_3(u)_{i,j} ((u)_{i+1,j} + (u)_{i,j+1}) \end{aligned}$$

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

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

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

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

วิธีเดินเวลาแบบชัดแจ้งสำหรับภาพอนาคตเท่านี้ขั้นตอนวิธีดังนี้

Algorithm 2: วิธีการเดินเวลาแบบชัดเจ็บสำหรับการต่อเติมภาพที่ใช้การแปรผันรวม**Input:**

- u คือรูปภาพที่ต้องการต่อเติม
- λ คือพารามิเตอร์เริกกิวลาร์เรซัน 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
```

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

```
 $i = i + 1$ 
```

```
end
```

3.4.2 วิธีการทำซ้ำแบบจุดตรึง (fixed-point iteration method)

คณะวิจัย [3] ได้แนะนำวิธีการเชิงตัวเลขสำหรับการกำจัดสัญญาณรบกวนโดยใช้วิธีการทำซ้ำแบบจุดตรึง ซึ่งสามารถประยุกต์เป็นวิธีเชิงตัวเลขสำหรับการต่อเติมภาพได้ดังนี้

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

$$-\nabla \cdot \left(\frac{\nabla u^{[\nu+1]}}{|\nabla u|^{[v]}} \right) + \lambda(u^{[\nu+1]} - z) = 0, \quad u^{[0]} = z \quad (3.4.3)$$

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 (3.4.3) เราจะใช้

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

วิธีการทำซ้ำแบบจุดตรึงมีขั้นตอนดังนี้

Algorithm 3: วิธีการทำซ้ำจุดตรึงสำหรับการต่อเติมภาพที่ใช้การแปรผันรวม

Input:

- u คือรูปภาพที่ต้องการต่อเติม
- λ คือพารามิเตอร์เร็กกิวลาร์เรเซ็น 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 4: การทำข้ามเกาส์-ไซเดล สำหรับวิธีการจุดตรึง

Input:

- n คือรูปภาพที่ต้องการต่อเติม
- λ ค่าพารามิเตอร์เริกกวิการเรใชชัน 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 (3.4.3) เรายังคงรู้ว่า β มีค่าน้อยลงมากขึ้นเท่าไหร่ ความแม่นยำของตัวแบบ (3.2.1) ยิ่งมีมากขึ้นเท่านั้น นอก จากนี้ รายังพบอีกว่า การแก้สมการ (3.4.1) และ (3.4.3) ยิ่งมีความยุ่งยากมากขึ้นสำหรับ β ที่มีค่าน้อยๆ

เพื่อเอาชนะความยากเชิงตัวเลขนี้ คณะวิจัยโดย [4] ได้แนะนำวิธีการสปริตเบรกแมนซึ่งสามารถถูกล่าวยัง พอกสั้นๆ ดังนี้

3.4.3 วิธีการสปริตเบรกแมน (Split Bregman method)

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

$$\min_{u, \mathbf{w}} \{ \mathcal{J}(u, \mathbf{w}) = \frac{1}{2} \int_{\Omega} \lambda (u - z)^2 d\Omega + \int_{\Omega} |\mathbf{w}| d\Omega + \frac{\theta}{2} \int_{\Omega} (\mathbf{w} - \nabla u + \mathbf{b})^2 d\Omega \} \quad (3.4.4)$$

สำหรับการหาค่าตอบของ (3.4.4) เราจะใช้วิธีการหาค่าต่อที่สุดแบบสลับ (alternating minimization method) โดยเริ่มจากการตึง \mathbf{w}^{old} และ \mathbf{b}^{old} จากนั้นแก้ปัญหาอย่างสำหรับ u

$$u^{\text{New}} = \arg \min_u \{ \mathcal{J}_1(u) = \frac{1}{2} \int_{\Omega} \lambda(u - z)^2 d\Omega + \frac{\theta}{2} \int_{\Omega} (\mathbf{w}^{\text{old}} - \nabla u + \mathbf{b}^{\text{old}})^2 d\Omega \} \quad (3.4.5)$$

ต่อไปใช้ u^{New} ที่ได้จากการแก้ปัญหาอย่าง (3.4.5) เพื่อแก้ปัญหาอย่างสำหรับ \mathbf{w}

$$\mathbf{w}^{\text{New}} = \arg \min_{\mathbf{w}} \{ \mathcal{J}_2(\mathbf{w}) = \int_{\Omega} |\mathbf{w}| d\Omega + \frac{\theta}{2} \int_{\Omega} (\mathbf{w} - \nabla u^{\text{New}} + \mathbf{b}^{\text{old}})^2 d\Omega \} \quad (3.4.6)$$

สุดท้ายจึงปรับปรุงพารามิเตอร์เบรกเมนโดย

$$\mathbf{b}^{\text{New}} = \mathbf{b}^{\text{old}} + \nabla u^{\text{New}} - \mathbf{w}^{\text{New}} \quad (3.4.7)$$

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

วิธีการสบริทเบรกเมนมีขั้นตอนวิธีดังนี้

Algorithm 5: วิธีสปริทเบรกเม่นสำหรับการต่อเติมภาพที่ใช้การแปรผันรวม**Input:**

- u คือรูปภาพที่ต้องการต่อเติม
- λ คือพารามิเตอร์เก็บกิวาวาร์เรชัน ที่ได้ถูกล่างึงในสมการ (3.2.2)
- θ คือพารามิเตอร์เพนล์ที่ ซึ่งเป็นจำนวนจริงบวก
- N_{gs} เป็นจำนวนเต็มบวกสำหรับกำหนดจำนวนรอบที่ทำงานของการทำเกาส์-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
     $u^{old} = u; w^{old} = w; b^{old} = b;$ 
     $u^{new} = \arg \min_u \{J_1(u) = \frac{1}{2} \int_{\Omega} \lambda(u - z)^2 d\Omega + \frac{\theta}{2} \int_{\Omega} (\mathbf{w}^{old} - \nabla u + \mathbf{b}^{old}) d\Omega\}$ 
     $w^{new} = \arg \min_{\mathbf{w}} \{J_2(\mathbf{w}) = \int_{\Omega} |\mathbf{w}| d\Omega + \frac{\theta}{2} \int_{\Omega} (\mathbf{w} - \nabla u^{New} + \mathbf{b}^{old}) d\Omega\}$ 
     $b^{new} = b^{old} + \nabla u^{new} - \mathbf{w}^{new}$ 
     $err = \frac{\|u^{new} - u^{old}\|}{\|u^{new}\|}$ 
     $i = i + 1$ 
end

```

จาก 3.4.5 โดยแคลคูลัสของการแปรผันจะได้สมการอยเลอร์ลAGRAGE ที่เกี่ยวข้องเป็น

$$\begin{cases} -\theta \Delta u + \lambda u = \lambda z - \theta \nabla \cdot (\mathbf{w} - \mathbf{b}) \\ \frac{\partial u}{\partial \mathbf{n}} = 0, \end{cases} \quad x \in \partial \Omega \quad (3.4.8)$$

จึงได้ว่าผลเฉลยของ $u = \arg \min_u J_1(u)$ สามารถหาได้จากการแก้ผลเฉลยของสมการอนุพันธ์ 3.4.8 โดยการใช้วิธีการไฟน์ติดฟเฟอร์น์และวิธีการเกาส์-ไซเดลจำนวน N_{gs} รอบ

และจาก 3.4.3 โดยแคลคูลัสของการแปรผันจะได้สมการอย่างลักษณะที่เกี่ยวข้องเป็น

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

ซึ่งสมการเชิงอนุพันธ์ดังกล่าวมีผลเฉลยแบบแม่นตรงคือ

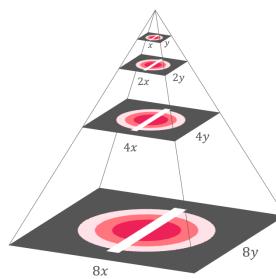
$$\mathbf{w} = \max \left\{ (\nabla u + \mathbf{b}) - \frac{1}{\theta}, 0 \right\} \quad (3.4.10)$$

ซึ่งสามารถใช้สมการผลเฉลยแบบแม่นตรง 3.4.10 เพื่อแก้หาผลเฉลยของ ได้

3.5 ขั้นตอนวิธีเชิงตัวเลขที่นำเสนอ

3.5.1 ขั้นตอนวิธีเชิงตัวเลขสำหรับต่อเติมภาพศิลปะ

สำหรับวิธีการซ่อมแซมภาพศิลปะไทย จะใช้วิธีการสปริทเบรกเมนเพื่อหลีกเลี่ยงปัญหาเชิงตัวเลขที่จะเกิดขึ้น แต่เพื่อให้วิธีการสปริทเบรกเมนประมวลผลภาพได้รวดเร็วขึ้น ผู้วิจัยได้พัฒนากระบวนการกำหนดคำตอบเริ่มต้น โดยวิธีการ มัลติรีโซลูชัน (multi-resolution method) หรือวิธีการพีระมิดรูปภาพ (pyramid method) [5] เริ่มจากการย่อขนาดรูปลงครึ่งหนึ่งโดยใช้วิธี Bilinear Interpolation จนกระทั่งถึงระดับความคมชัดที่ต้องการ จากนั้นทำการต่อเติมภาพขนาดเล็ก และนำผลลัพธ์ที่ได้จากการขยายภาพขึ้นสองเท่าโดยใช้ Bilinear Interpolation เป็นคำตอบเริ่มต้นสำหรับการต่อเติมภาพในชั้นถัดไป



รูปที่ 3.5.1: วิธีการพีระมิดรูปภาพ

ขั้นตอนวิธีสำหรับการทำพีระมิดรูปภาพสำหรับการต่อเติมภาพแบบสปริทเบรกเมนเพื่อให้ประมวลผลได้เร็วขึ้นนั้นสามารถสรุปได้ดังนี้

Algorithm 6: 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 จำนวนรอบการทำสปริทเบรกเมนท์ชั้นละเอียดสุด N_1 จำนวนรอบการทำสปริทเบรกเมนท์ชั้นต่างๆ N_2 จำนวนรอบการทำสปริทเบรกเมนท์ชั้นใหญ่สุด**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 7: Bilinear Interpolation

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

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

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

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

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

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

$$\Delta r = r_f - r$$

$$\Delta c = c_f - c$$

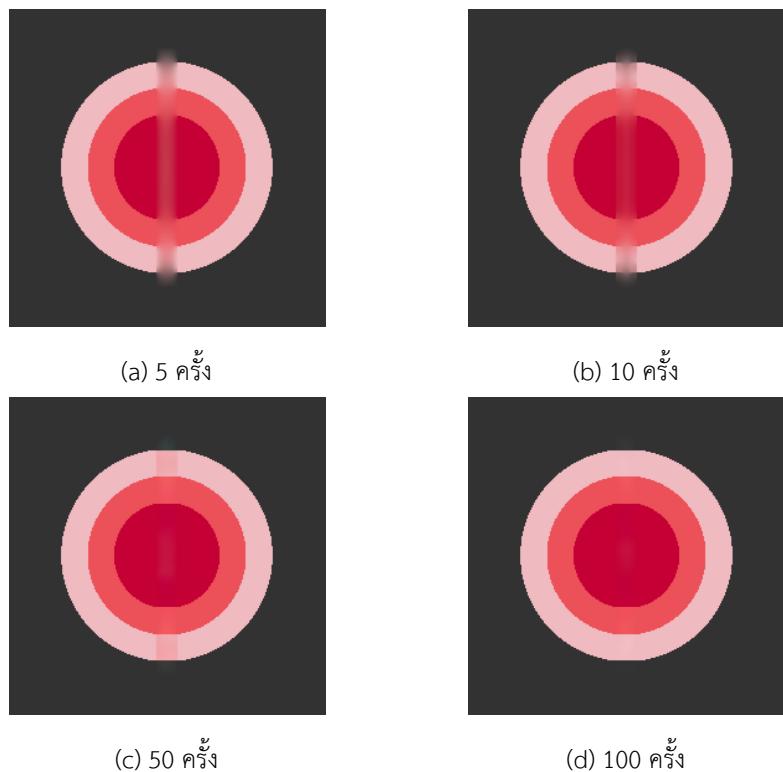
$$J(r', c') = I(r, c) \cdot (1 - \Delta r) \cdot (1 - \Delta c)$$

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

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

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

นอกจากนี้แล้ว ผู้วิจัยยังได้สังเกตอีกว่า การทำขั้นนี้จะสูญเสียเร็วในช่วงแรก จากนั้นความเร็วในการลู่เข้าจะลดลง ซึ่งทำให้การทำขั้นเพียงไม่กี่ครั้งในระดับความคมชัดเดิม มีผลการซ่อมแซมภาพจนแสดงความคล้ายคลึงกับภาพต้นฉบับได้



รูปที่ 3.5.2: พีระมิดที่ลำดับการทำข้าเป็น 10/10/10 และที่ระดับความคุมชัดละเอียดสุดใช้จำนวนการทำข้าที่ต่างกัน

ผู้วิจัยจึงกำหนดให้การทำข้าวในระดับความละเอียดสูงสุดเท่ากับ 10 ครั้ง เพื่อการทำข้าวในระดับความละเอียดสูงสุดทำให้ทำงานได้ช้า อีกทั้งการทำข้าวในขันที่รูปภาพมีขนาดเล็กจำนวนมาก ไม่ช่วยให้การประมวลผลได้เร็วขึ้น ผู้วิจัยจึงเลือกใช้การทำข้าวแบบ 10/3/3/10 ในการต่อเติมภาพ

3.5.2 ขั้นตอนวิธีเชิงตัวเลขสำหรับซ่อมแซมภาพวิดีโอ

เนื่องจากไฟล์วิดีโอนั้นประกอบด้วยชุดของภาพหลายภาพ ก่อให้ว่าคือ $V = \{u_i | i = 1, 2, 3...N_f\}$ ทำให้ขั้นตอนวิธีการลบคำบรรยายออกจากวิดีโอ จะต้องทำการต่อเติมบริเวณที่เป็นบทบรรยายทีละภาพ ดังที่แสดงในขั้นตอนวิธีต่อไปนี้

Algorithm 8: 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 ส่วนการหาโดเมนต่อเติมซึ่งเป็นบทบรรยายอนิเมะ จะกล่าวถึงในหัวข้ออย่างถัดไป

3.5.3 การหาบทบรรยายบนอนิเมะ

ก่อนจะลบทบรรยายนั้น จำเป็นต้องหาบทบรรยายในภาพให้ได้เสียก่อน โดยบทบรรยายของอนิเมะนั้น มักจะใช้ขอบของตัวอักษรเป็นสีดำ อีกทั้งบทบรรยายนั้นจะลอยห่างจากขอบของวิดีโอ และขนาดของคำบรรยายนั้นจะมีขนาดอยู่ประมาณหนึ่งไม่ใหญ่หรือไม่เล็กเกินไป ด้วยสมบัตินี้เองทำให้จึงสามารถหาบริเวณบนเฟรมที่เป็นบทบรรยายได้โดยจะมีวิธีหาพื้นที่ซึ่งเป็นบทบรรยายดังนี้

Algorithm 9: Finding subtitle

```

 $D \leftarrow \text{findsub}(\mathbf{u})$ 
• ทำการเปลี่ยนสีดำในภาพ  $\mathbf{u}$  ให้เป็นสีขาวแล้วเปลี่ยนอื่นๆ ให้เป็นสีดำเพื่อหาขอบของคำ
    บรรยาย
• เปลี่ยนบริเวณสีขาวในภาพให้เป็นสีดำ และเปลี่ยนบริเวณสีดำให้เป็นสีขาว
• ทำการลับบริเวณสีขาวซึ่งติดกับขอบของภาพออกไป เนื่องจากบทบรรยายจะลอยอยู่ไม่ติดกับ
    ขอบเสมอ
• ลับบริเวณที่ใหญ่เกินกว่าจะเป็นบทบรรยาย
• ลับบริเวณที่เล็กเกินกว่าจะเป็นบทบรรยาย
• ทำการขยายพื้นที่ๆ เป็นสีขาวขึ้นด้วยความกว้างของขอบบทบรรยาย
• สีขาวที่เหลืออยู่ในภาพจะเป็นบทบรรยาย

```

3.5.4 การลบคำบรรยายจากบทอนิเมะ

สำหรับอนิเมะนั้น แต่ละเฟรมจะเป็นรูปภาพ เราจึงสามารถประยุกต์ใช้วิธีการซ่อนแซมภาพจิตรกรรมไทยมาใช้ในการลบคำบรรยายได้ แต่ผู้วิจัยก็ได้สังเกตว่า สำหรับอนิเมะที่เป็นวิดีโอล่า ในขณะที่ประมวลผลวิดีโอ เราสามารถใช้ผลการต่อเติมภาพจากภาพที่แล้ว มาใช้เป็นคำตอบเริ่มต้นจึงได้ว่าขั้นตอนการลบบทบรรยายออกจากวิดีโอมีดังนี้

Algorithm 10: วิธีการทำงานบนวิดีโอ เมื่อต้องการผลจากภาพที่แล้วมาใช้เป็นคำตอบเริ่มต้น

```

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

```

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

ขั้นตอนวิธี การยึมเฟรม จะเป็นการนำผลลัพธ์จากเฟรมก่อนหน้ามาเป็นคำตอบในการเริ่มต้นในการประมวลผลเพื่อให้ผลลัพธ์ลุ้นเข้าได้เร็วขึ้น

Algorithm 11: การลับบทบรรยายโดยใช้วิธีการยึมเฟรม

```

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

```

ขั้นตอนวิธี การข้ามเฟรม สำหรับเฟรมใดที่ผลลัพธ์ใกล้เคียงกันมาก จะทำการข้ามการต่อเติมภาพในเฟรมนั้นไปโดยใช้คำตอบจากเฟรมก่อนหน้าแทนเพื่อลดเวลาการประมวลผล

Algorithm 12: การลับบทบรรยายโดยใช้วิธีการข้ามเฟรม

```

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

```

ขั้นตอนวิธี การข้ามและยึมเฟรม คือขั้นตอนวิธี 11 และขั้นตอนวิธี 12 ที่นำมาประยุกต์ใช้งานร่วมกัน

Algorithm 13: การลับบทบรรยายโดยใช้วิธีการข้ามเฟรมและยึมเฟรม

```

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

```

จากนั้นทางผู้พัฒนาจะนำวิธีการแบบทบทวนอยอนิเมะที่นำเสนอในปีก่อนมาลองแบบทบทวนเพื่อเปรียบเทียบประสิทธิภาพกับวิธีสปริตเบรกเมนเดิมที่นำมาใช้กับวิดีโอด้วย

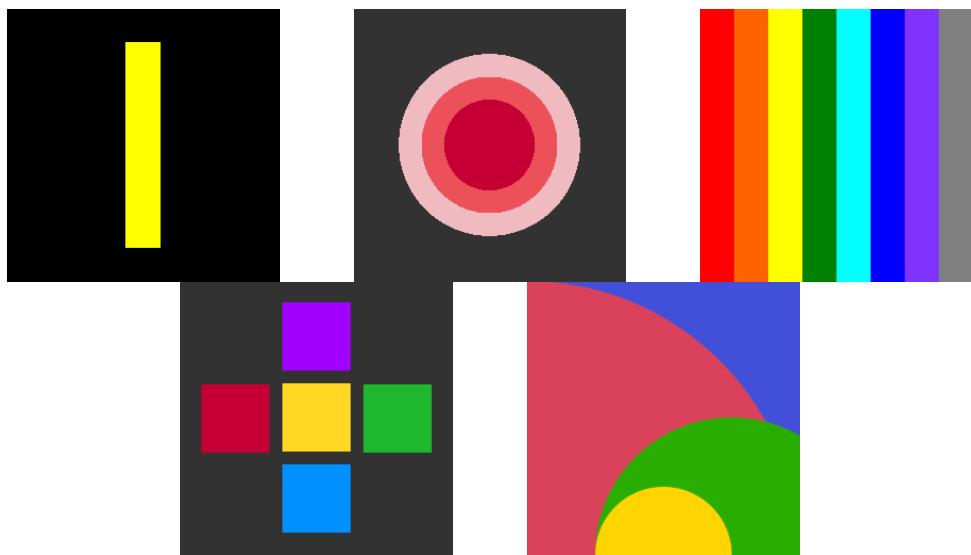
บทที่ 4

การทดลองเชิงตัวเลข

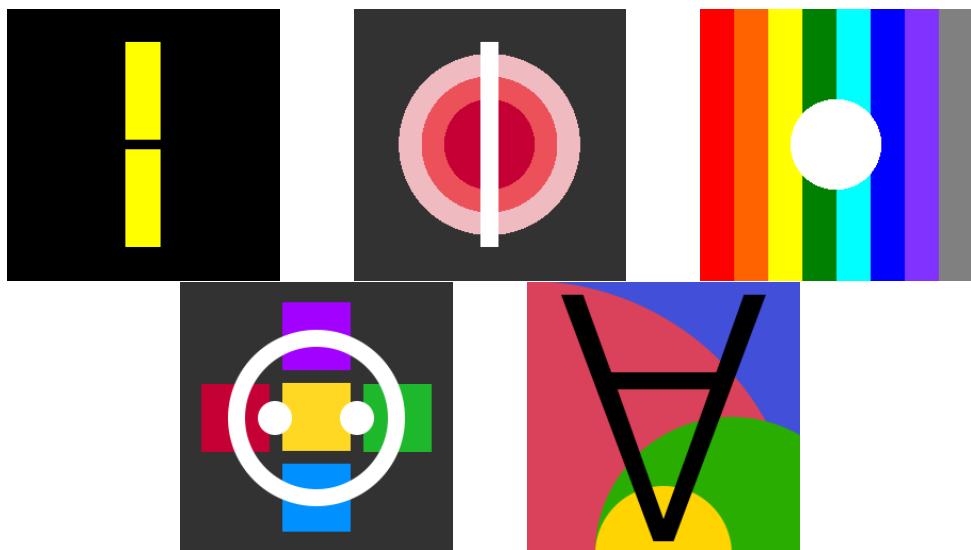
การทดลองเชิงตัวเลขถือเป็นหนึ่งในหัวใจสำคัญของโครงการวิจัยชั้นนี้ โดยจะทำการทดลองเชิงตัวเลข เพื่อเปรียบเทียบวิธีที่พัฒนาขึ้นมา กับวิธีที่มีอยู่เดิม และเปรียบเทียบกันเองระหว่างวิธีที่ได้พัฒนาขึ้น โดยการทดลองเชิงตัวเลข จะแบ่งออกเป็น 2 ส่วน ได้แก่ การทดลองซ่อมแซมภาพศิลปะไทย และการทดลองลบบทบรรยายอนิเมะ

4.1 การซ่อมแซมภาพจิตรกรรมไทยโบราณ

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



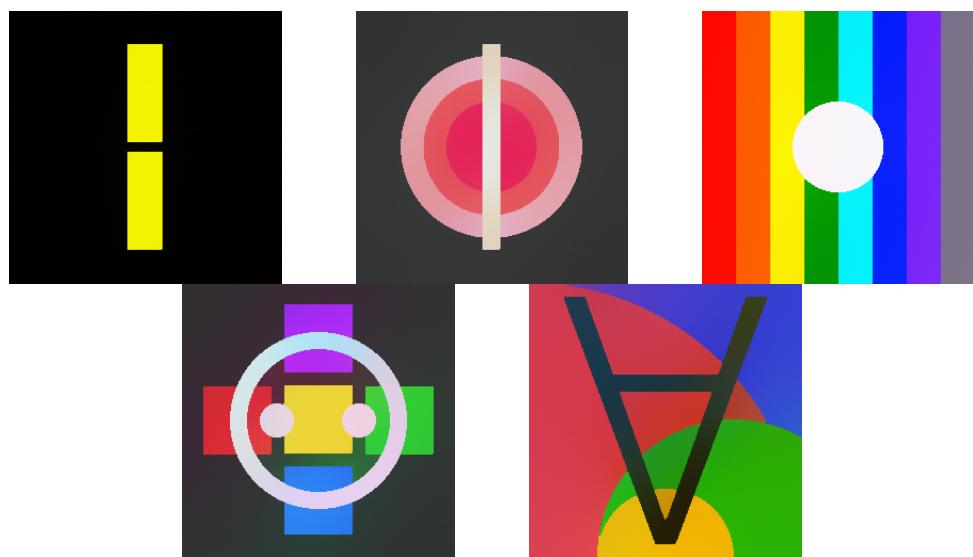
รูปที่ 4.1.1: ภาพต้นฉบับ



รูปที่ 4.1.2: ภาพที่จะทำการซ่อมแซม

4.1.1 การเปรียบเทียบประสิทธิภาพขั้นตอนวิธีเชิงตัวเลขที่มีอยู่แล้ว

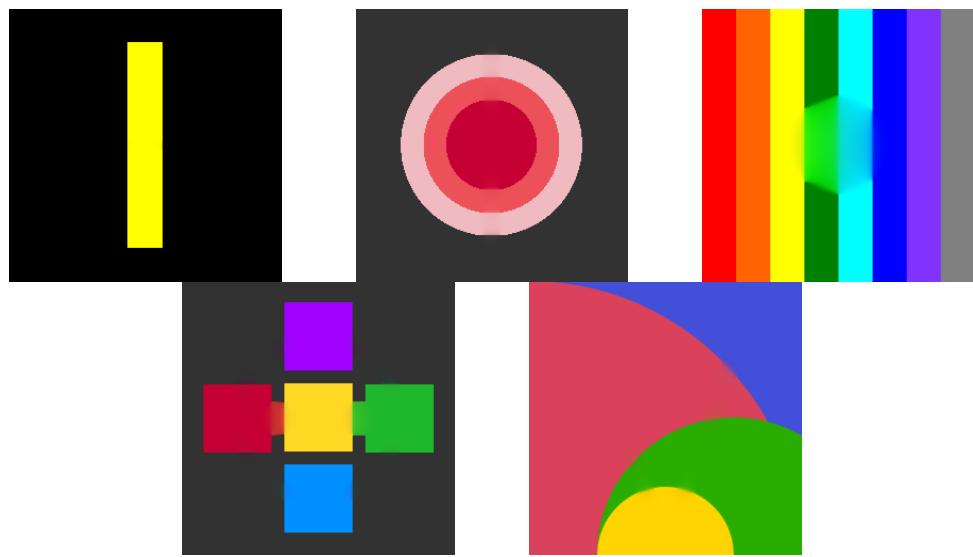
การทดสอบประสิทธิภาพจะใช้ $\varepsilon = 1 \times 10^{-4}$ และ $N = 10,000$ โดยรูปที่ 4.1.3 - 4.1.5 และตารางที่ 4.1 - 4.3 แสดงผลการซ่อมแซมภาพสังเคราะห์ทั้ง 5 ภาพ



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

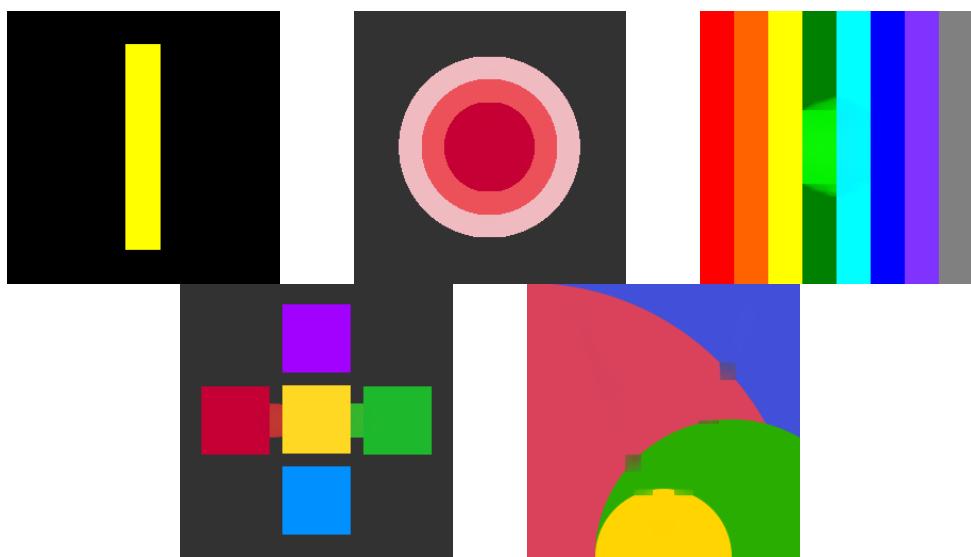
ตารางที่ 4.1: ผลการซ่อมแซมวิธีการเดินเวลา



รูปที่ 4.1.4: ผลการซ่อมแซมจากวิธีการทำข้ามแบบจุดตึง

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: 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

ประสิทธิภาพของวิธีการเชิงตัวเลขทั้ง 3 วิธี สามารถสรุปได้ดังนี้

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

ตารางที่ 4.4: แสดงการซ่อมแซมเฉลี่ยของวิธีการเชิงตัวเลข

จากทั้ง 3 วิธีที่ได้ทดสอบ จะเห็นได้ว่าวิธีการสปริทเบรกแมนใช้เวลาอยกว่าวิธีอื่น และมีคุณภาพซึ่งพิจารณาจาก ค่า PSNR และค่า SSIM มากกว่าวิธีอื่น ผู้จัดจึงสนใจทำการปรับปรุงวิธีสปริทเบรกแมนให้มีประสิทธิภาพสูง

4.1.2 ขั้นตอนวิธีเชิงตัวเลขสำหรับต่อเติมภาพชนิดใหม่

ตารางที่ 4.5 แสดงผลการเปรียบเทียบประสิทธิภาพของขั้นตอนวิธีการที่ 6 ภายใต้การเปลี่ยนแปลงจำนวนรอบของการทำซ้ำของวิธีการสปริทเบรกแมนบนภาพที่มีความคมชัด 256×256 พิกเซล ตัวอย่าง เช่น $10/3/3/10000$ หมายถึงที่ระดับความคมชัดขยายสูดซึ่งมีขนาดเป็น 32×32 พิกเซลจะทำซ้ำไม่เกิน 10 ครั้ง สำหรับที่ความคมชัดละเอียดที่นี่เป็น 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 การทำข้าวันนั้นจะถูกเข้าเร็วในช่วงแรก จนกวนความเร็วในการถูกเข้าจะลดลง 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
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: ผลการซ้อมเช้มภาพโดยวิธีการเชิงตัวเลขที่นำเสนอด้วยรูปของค่าเฉลี่ยของผลที่ได้จากตารางที่ 4.7

จากตารางจะเห็นว่า การทำข้าในชั้นที่รูปภาพมีขนาดเล็กมากจำนวนมาก ไม่ช่วยให้การประมาณผลได้เร็วขึ้น ผู้วิจัยจึงเลือกใช้การทำข้าแบบ 10/3/3/10 ในการต่อเติมภาพ

4.1.3 การทดสอบประสิทธิภาพในการซ้อมเช้มภาพจิตรกรรมไทยโบราณ

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 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

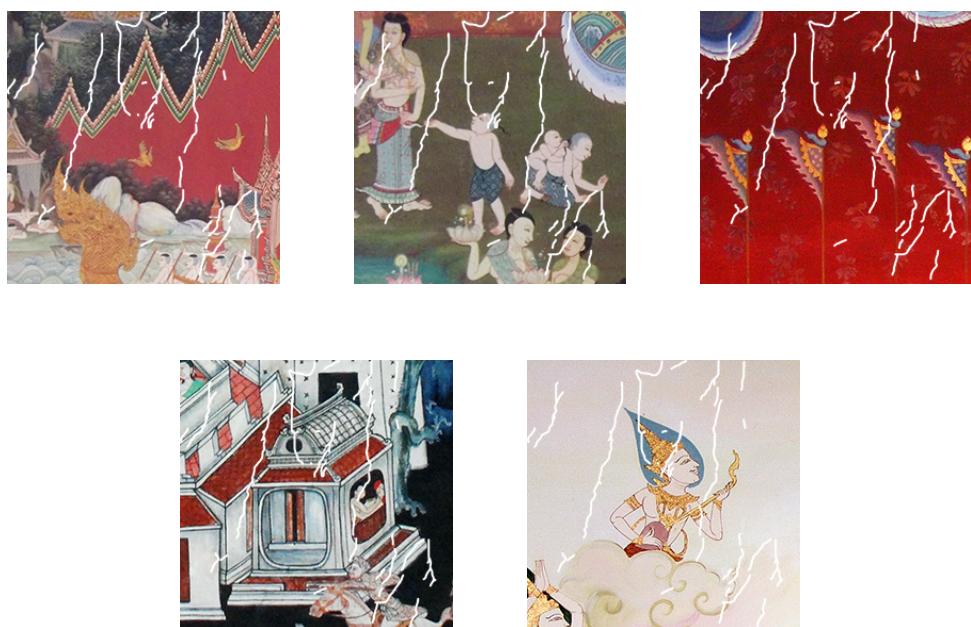
³ภาพถ่ายที่วัดพระยืนพุทธบาทยุคล; ภาพจาก https://commons.wikimedia.org/wiki/File:Wat_Phra_Yuen_Phutthabat_Yukhon_01.jpg สืบคันเมื่อวันที่ 23 กันยายน 2561

⁴ภาพถ่ายที่วัดคงคาราม; ภาพจาก https://commons.wikimedia.org/wiki/File:จิตรกรรม_อุบลสถา瓦ัดคงคาราม.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

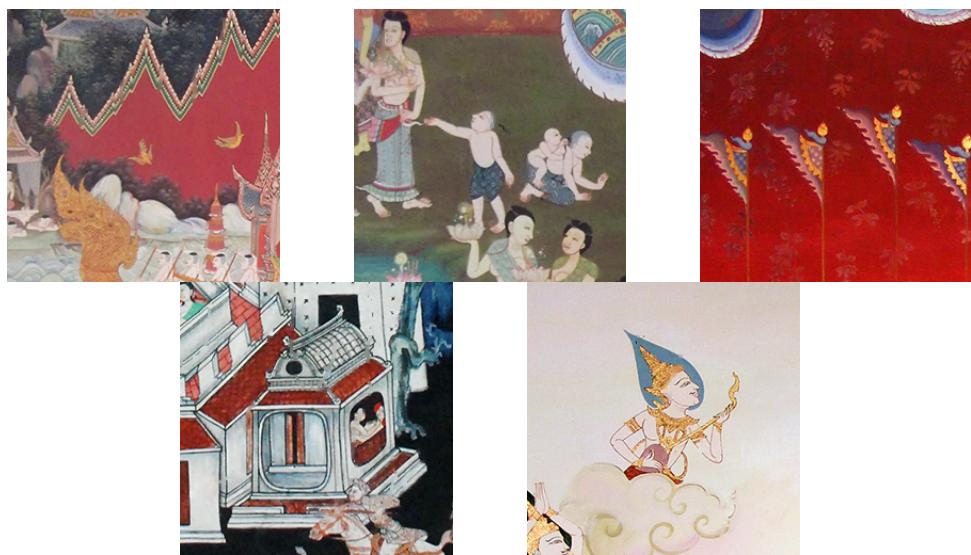
การสบริทเบรกแมนพร้อมทั้งการใช้พีระมิดรูปภาพที่มีการทำซ้ำแต่ละชั้นเป็น 10/3/3/10 ได้ผลลัพธ์ออกมาเป็นดังนี้



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



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

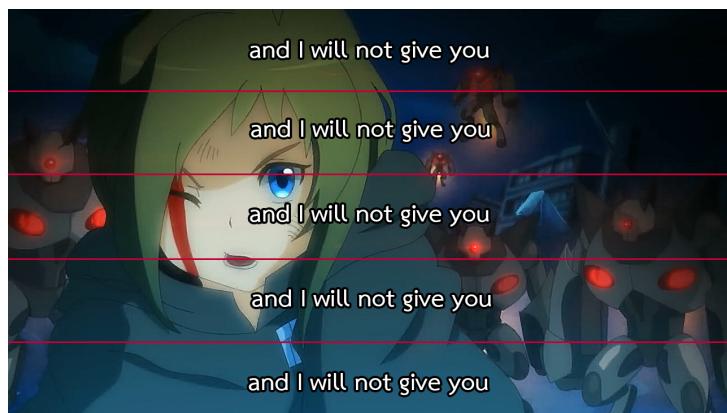
วิธีการ	เวลาประมาณ (วินาที)	PSNR (dB)	SSIM
สปริตเบรกแม่น	2.72	34.89	1.0000
วิธีการที่พัฒนาขึ้น	0.39	35.30	1.0000

ตารางที่ 4.11: แสดงผลการซ่อมแซมภาพศิลปะไทยในรูปค่าเฉลี่ยจากตารางที่ 4.9 และตารางที่ 4.10

จากตารางที่ 4.11 จะเห็นได้ว่า วิธีที่พัฒนาขึ้นนั้นสามารถทำงานได้เร็วกว่าวิธีสปริทเบรกเมนเดิม และบังมีคุณภาพที่ดีขึ้นด้วย

4.2 การลับบทบรรยายบนอนิเมะ

สำหรับการลับบทบรรยายอนิเมะ จะใช้วิดีโอ Anime Festival Asia Special Video - feat. Inori Aizawa ซึ่งผลิตโดย Collateral Damage Studios โดยจะตัดวิดีโอด้วยตัดวิดีโอด้วยปกติแล้ว อนิเมะมีบรรทัดเพียง 1 ถึง 2 บรรทัด จึงทำการแบ่งวิดีโอดอกอีกเป็น 5 ส่วนได้ขนาดเป็น 1280 x 144 พิกเซลก่อนนำไปทดสอบในลำดับถัดไป และสำหรับบทบรรยายที่จะใช้ทดสอบนั้น เป็นจากวิดีโอ Anime Festival Asia Special Video - feat. Inori Aizawa ไม่มีคำพูดใดๆ จึงใช้ทดสอบความ *lorem ipsum* เป็นบทบรรยาย โดยจะทำการแสดงบทบรรยาย 1 บรรทัด ความยาว 3 วินาที ทุก 2 วินาที นั่นคือในวิดีโอดังกล่าวจะมีบทบรรยายทั้งสิ้น 20 บรรทัด



รูปที่ 4.2.1: การแบ่งไฟล์วิดีโอด้วยตัดวิดีโอด้วยปกติ

4.2.1 การหาบทบรรยายบนอนิเมะ

วิธีการหาบทบรรยายที่กล่าวไปข้างต้น จะทำการทดสอบกับความ *lorem ipsum*⁶ ที่ถูกแปลเป็นภาษาไทย ภาษาอังกฤษ และภาษาญี่ปุ่น โดยมีความสามารถในการหาโดเมนต่อเติมในบทบรรยายภาษาต่างๆ ดังนี้

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

ภาษา	วิดีโอ	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 การลบคำบรรยายจากบทอนิเมะ

สำหรับอนิเมะนี้ แต่ละเฟรมจะเป็นรูปภาพ เราจึงสามารถประยุกต์ใช้วิธีการซ้อมแซมภาพจิตกรรมไทยมาใช้ในการลบคำบรรยายได้ แต่ผู้จัดก็ได้สังเกตว่า สำหรับอนิเมะที่เป็นวิดีโอแล้ว ในขณะที่ประมวลผลวิดีโอ จะ

สามารถใช้ผลการต่อเติมภาพจากภาพที่แล้ว มาใช้เป็นคำตอบเรื่มต้นจึงได้ว่าขั้นตอนการระบบทั่วไปของวิดีโอดู ซึ่งผลลัพธ์เปรียบเทียบระหว่างแบบใช้วีรี ยีมเพرم ใช้วีรีข้ามเพرم และวีรีข้ามและยีมเพرم ได้ผลดังตาราง

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

ตารางที่ 4.14: ผลการระบบทั่วไปของวิดีโอดู ซึ่งผลลัพธ์เปรียบเทียบระหว่างแบบใช้วีรี ยีมเพرم ใช้วีรีข้ามเพرم และวีรีข้ามและยีมเพرم ได้ผลดังตาราง

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: ผลการซ่อมแซมภาพโดยวิธีการเชิงตัวเลขที่นำเสนอบนรูปของค่าเฉลี่ยของผลที่ได้จากตารางที่ 4.14

จากนั้นทำการทดสอบการต่อเติมวิดีโอหั้ง 5 โดยวิธีที่คิดค้นขึ้นใช้วิธีการสปริทเบรกเมนพร้อมหั้งการใช้พีระมิดรูปภาพที่มีการทำขา้แต่ละชั้นเป็น 10/3/3/10 พร้อมหั้งใช้การข้ามเฟรมและยีมเฟรม 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 ชั่วโมงแล้วยังประมาณผลวิดีโอชุดทดสอบแรกไม่เสร็จ ทางผู้พัฒนาจึงตัดสินใจยุติการทดลอง เนื่องจากอาจต้องใช้เวลาการประมาณผลเป็นเวลาหลายชั่วโมงสำหรับวิดีโอด้วยความยาว 1 นาที ส่วนวิธีที่คิดค้นขึ้น พบว่าสำหรับวิดีโอด้วยความยาว 1 นาที สามารถทำงานได้เสร็จอย่างรวดเร็วโดยใช้เวลาเพียง 75 วินาที

บทที่ 5

สรุป

สำหรับโครงการวิจัยนี้ได้ทำการพัฒนาขั้นตอนวิธีเชิงตัวเลขสำหรับการซ่อมแซมภาพศิลปะไทยและการลบบทบรรยายอนิเมะได้มีผลการดำเนินงานทั้งสิ้นดังนี้

ขั้นตอนวิธีเชิงตัวเลขทั้ง 3 วิธีสำหรับแก้ตัวแบบการต่อเติมภาพด้วยการแปรผันรวม ได้แก่วิธีเดินเวลาแบบชัดแจ้ง วิธีทำขาจุดตรึง และวิธีการสปริทเบรกแม่น พบว่า วิธีการสปริทเบรกแม่นมีคุณภาพที่ดีกว่าเมื่อวัดด้วยค่า PSNR และ SSIM ส่วนเวลาที่ใช้ประมาณผลพบว่าวิธีสปริทเบรกแม่นใช้เวลาอยกว่าวิธีเดินเวลา 8 เท่า และใช้เวลาอยกว่าวิธีทำขาจุดตรึง 5 เท่า

พิริมิดรูปภาพใช้เพื่อเพิ่มความเร็วในการประมวลผลของวิธีการสปริทเบรกแม่น พบว่าการทำขาบนพิริมิดรูปภาพด้วยการทำขา 10/3/3/10 ใช้เวลาประมวลผลน้อยที่สุดและให้ค่าคุณภาพทั้งในด้าน PSNR และ SSIM ใกล้เคียงกันอีกนึงเลือกใช้พิริมิดรูปภาพนี้ในการซ่อมแซมรูปภาพ

การซ่อมแซมภาพศิลปะไทย การใช้วิธีการสปริทเบรกแม่นพร้อมทั้งการใช้พิริมิดรูปภาพ พบว่าภาพที่ผ่านการต่อเติมให้คุณภาพดีกว่าเมื่อวัดด้วย PSNR และใช้เวลาอยกว่าประมาณ 7 เท่า

ขั้นตอนวิธีค้นหาคำบรรยายในภาพด้วยวิธีการที่คิดค้นขึ้น (ขั้นตอนวิธี 9) พบว่ามีความผิดพลาดในการตรวจหาพิกเซลที่เป็นบทบรรยายอยู่ที่ร้อยละ 11.98 ซึ่งจะใช้ขั้นตอนวิธีนี้ในการหาคำบรรยายเพื่อทำการลบถัดไป

เนื่องจากวิดีโอเป็นชุดของภาพคณผู้วิจัยจึงได้เสนอขั้นตอนวิธีขั้มเฟรม (ขั้นตอนวิธี 11) ขั้นตอนวิธียึดเฟรม (ขั้นตอนวิธี 12) และขั้นตอนวิธียึดเฟรมและขั้มเฟรม (ขั้นตอนวิธี ?? เพื่อช่วยในการลดเวลาการประมวลผล พบว่าวิธียึดเฟรมและขั้มเฟรมใช้เวลาประมวลผลน้อยสุด ซึ่งใช้เวลาอยกว่าวิธีสปริทเบรกแม่นและพิริมิดรูปภาพบันวิดีโอถึง 2 เท่า

สำหรับการตอบบทบรรยายอนิเมะเมื่อใช้วิธีตรวจหาคำบรรยายที่คิดขึ้นขึ้นพร้อมทั้งใช้วิธีการเดียวกับที่ใช้สำหรับซ่อมแซมศิลปะไทย รวมกับการวิธีการขึ้นเฟรมและข้ามเฟรม พบว่าใช้เวลาอยู่กว่าวิธีการสปริทเบรก แนะนำบันวิดีโอ 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.
- [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 โปรแกรมสำหรับทดสอบ

สำหรับโครงงานวิจัยเรื่องนี้เป็นการพัฒนาวิธีเชิงตัวเลข การจะวัดประสิทธิภาพของวิธีการเชิงตัวเลขได้จำเป็นจะต้องใช้โปรแกรมเข้าทดสอบ โดยโครงงานวิจัยนี้ มีโปรแกรมสำหรับทดสอบ ซึ่งสามารถแบ่งได้ออกเป็น 2 ส่วนคือโปรแกรมสำหรับทดสอบขั้นตอนการซ่อมแซมภาพศิลปะไทยและการลบคำบรรยายอนิเมะ

A.1.1 โปรแกรมสำหรับทดสอบการซ่อมแซมภาพศิลปะไทย

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 โปรแกรมทดสอบการลบคำบรรยายอนิเมะ

สำหรับโปรแกรมทดสอบการลบคำบรรยายอนิเมะนี้ เขียนด้วยภาษา 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

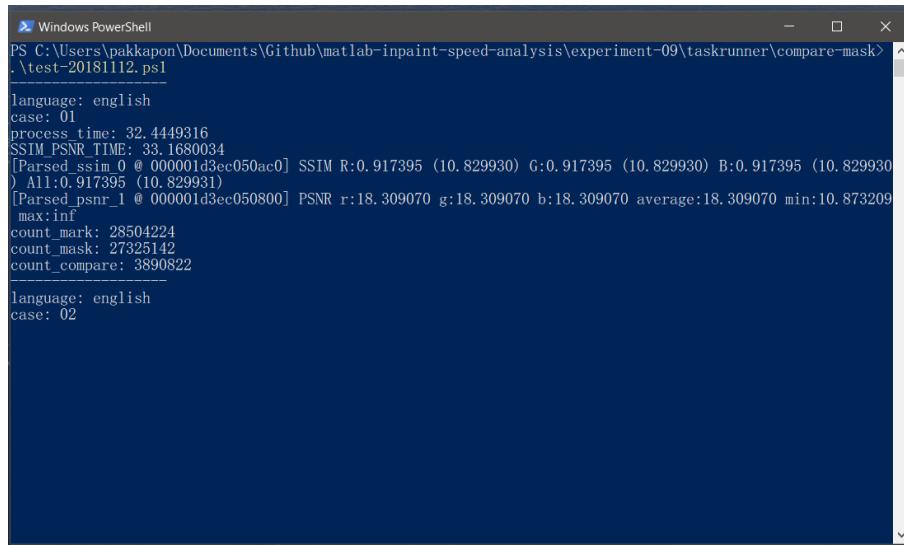
splitbergman
Loop: 10/1/10
Processed in 172.3858612
Comparing Result... splitbergman

```

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

```

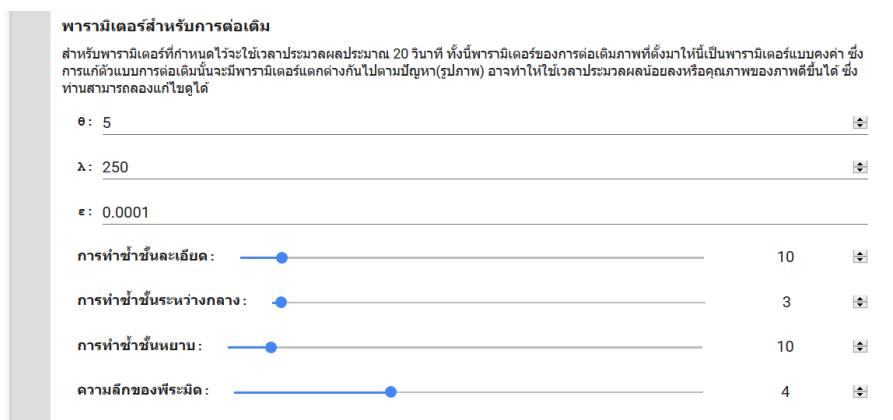
รูปที่ A.1.3: ตัวอย่างโปรแกรมสำหรับทดสอบการทำคำบรรยายที่พัฒนาขึ้น

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



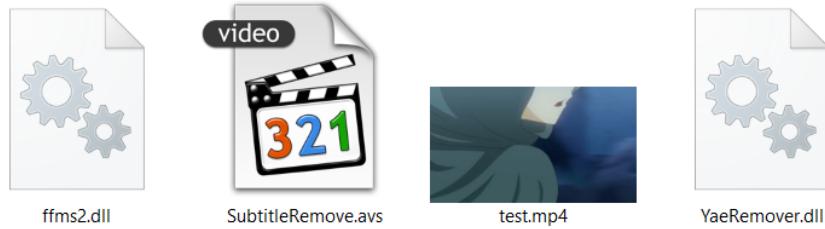
รูปที่ A.2.2: ตัวอย่างการปรับค่าพารามิเตอร์ที่ใช้ในโครงงานวิจัยนี้



รูปที่ A.2.3: ตัวอย่างภาพผลลัพธ์จาก Google Colab

A.3 โปรแกรมตัวอย่างการลบคำบรรยายอนิเมะ

สำหรับการลบคำบรรยายอนิเมะนั้น ขณะนี้ยังรองรับเพียงขอบบรรยายที่เป็นสีดำเท่านั้น 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 สามารถแก้พารามิเตอร์เพื่อใช้กับวิดีโອอนิเมะอื่นได้