

Report Progress

Division of Applied Mathematics, Department of Mathematics

Faculty of Science, Silpakorn University

Date : 23 พฤศจิกายน 2561

Advisor : ผู้ช่วยศาสตราจารย์ ดร.นพดล ชุมชอบ

Student : นายภัคพล พงษ์ทวี รหัส 07580028

Project Title : ขั้นตอนวิธีเชิงตัวเลขชนิดใหม่สำหรับการต่อเติมภาพที่ใช้การแปรผันรวมกับการประยุกต์สำหรับซ่อมแซมภาพจิตรกรรมไทยโบราณและการลบบทบรรยายจากอนิเมะ

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

1 ที่มาและความสำคัญ

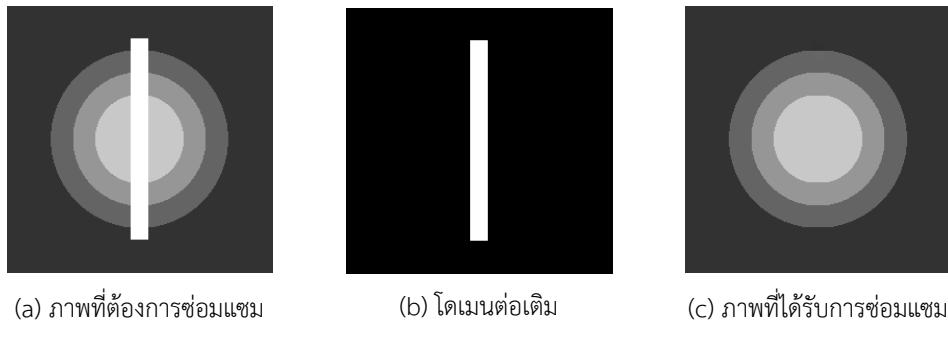
ในปัจจุบันการใช้ภาพดิจิตัล (digital images) ในสังคมเครือข่ายได้รับความนิยมอย่างแพร่หลายเนื่องจากโทรศัพท์เคลื่อนที่มีราคาถูกลงแต่มีความสามารถที่ชาญฉลาด สามารถทำหน้าที่ได้ตั้งแต่การเป็นกล้องดิจิตัลคอมแพค (compact digital camera) คุณภาพดีให้ภาพดิจิตัลที่มีความคมชัดสูงจนไปถึงการทำหน้าที่ดังเช่นเครื่องคอมพิวเตอร์ส่วนบุคคลที่สามารถเชื่อมต่อกับระบบเครือข่ายไร้สายเพื่อรับส่งภาพดิจิตัลในสังคมเครือข่ายด้วยความสะดวกและรวดเร็ว

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

โดยทั่วไปภาพดิจิตัลจะได้รับการประมวลผลภาพก่อนนำไปใช้งานเพื่อให้สามารถใช้ข้อมูลที่ปรากฏบนภาพได้ตรงวัตถุประสงค์ของการใช้งานมากที่สุด ตัวอย่างเช่น ภาพบุคคล (portrait) อาจจำเป็นต้องได้รับการกำจัดสัญญาณรบกวนออกจากภาพและ/หรือปรับเพิ่มความละเอียดข้อมูลของความเข้มของสีและความสว่างของสีบนบริเวณใบหน้าก่อนนำภาพไปใช้งานเพื่อจัดทำต้นฉบับวารสารหรือหนังสือของสำนักพิมพ์ เป็นต้น

การต่อเติมภาพ (image inpainting) เป็นวิธีการประมวลผลภาพชนิดหนึ่งที่มีเป้าหมายเพื่อซ่อมแซมภาพด้วยการต่อเติมข้อมูลของความเข้มของสีบนบริเวณที่กำหนด (ต่อไปจะเรียกบริเวณนี้ว่าโดเมนต่อเติม (inpainting domain)) โดยอาศัยข้อมูลของความเข้มของสีที่ปรากฏในภาพ ตัวอย่างเช่น กำหนดให้รูปที่ 1.1 (a) แสดงภาพที่ต้องการซ่อมแซมระดับความเข้มของสีบนบริเวณแห่งวัตถุรูปร่างสีเหลืองสีขาว การต่อเติมภาพ

ตั้งกล่าวจะเริ่มด้วยการกำหนดให้บริเวณแห่งวัตถุรูปร่างสีเหลี่ยมสีขาวเป็นโหมดเมนการต่อเติมดังรูปที่ 1.1 (b) จากนั้นภาพที่ได้รับการซ่อมแซมหรือภาพที่ได้รับการต่อเติม (restored or inpainted image) ซึ่งแสดงในรูปที่ 1.1 (c) ได้มาจากขั้นตอนวิธีการต่อเติมภาพ (inpainting algorithm) ซึ่งได้รับการออกแบบเพื่อนำข้อมูลที่ปรากฏบนภาพในบริเวณใกล้เคียงกับขอบของโmodeนต่อเติมมาซ่อมแซมภาพ



รูปที่ 1.1: ตัวอย่างการซ่อมแซมภาพ; (a) ภาพที่ต้องการซ่อมแซม; (b) โmodeนต่อเติม; (c) ภาพที่ได้รับการซ่อมแซม

เท่าที่ผู้วิจัยศึกษาและค้นคว้ามาจนถึงขณะนี้ ผู้วิจัยพบว่าการต่อเติมภาพมักนิยมนำไปใช้งานสำหรับการปรับแต่งความสวยงามของภาพบุคคลที่ถ่ายจากโทรศัพท์เคลื่อนที่ เช่น การลบร่องรอยของรอยตีนกา การลบร่องรอยแผลเป็นที่เกิดจากสิ่วเสี้ยน การลดร่องรอยของความชรา หรือ การเพิ่มความใสและความเนียนของสีผิวนบนบริเวณใบหน้าผ่านโปรแกรมแอปพลิเคชันแต่งรูปภาพที่มีอยู่ในแอปสโตร์ (App Store) หรือ กูเกิลเพลย์ (Google Play) เป็นต้น

1.1 การซ่อมแซมภาพจิตรกรรมไทยโบราณ

ภาพจิตรกรรมไทย คือ ภาพเขียนที่มีเอกลักษณ์ความเป็นศิลปะไทยซึ่งได้เด่นและแตกต่างจากภาพเขียนของชนชาติอื่น ในอดีต ช่างไทยได้สร้างสรรค์ลวดลายและสีสันบนภาพวาดเพื่อสะท้อนประเพณีและวัฒนธรรมในสังคมไทยที่เกี่ยวกับศาสนา ประวัติศาสตร์ โบราณคดี ชีวิตความเป็นอยู่ วัฒนธรรมการแต่งกายตลอดจนการแสดงการเล่นพื้นเมืองต่าง ๆ ของแต่ละยุคสมัย

อย่างไรก็ตาม ภาพจิตรกรรมไทยจำนวนไม่น้อยที่เสื่อมสภาพตามกาลเวลา และรอคوعการซ่อมแซมจากช่างในสมัยปัจจุบันที่ต้องไม่สร้างความเสียหายให้กับภาพเขียนเพิ่มขึ้นมากกว่าเดิม ที่ผ่านมาภาพที่ผ่านการซ่อมแซมมาแล้วจำนวนไม่น้อยได้รับความเสียหายหลังจากการซ่อมแซม ถึงแม้สภาพโดยรวมของภาพจิตรกรรมเดิมยังคงอยู่ แต่รายละเอียดในตัวภาพเขียนได้เปลี่ยนไป ก่อให้เกิดความเสียหายที่ประเมินค่าไม่ได้

การซ่อมแซมภาพจิตรกรรมไทยโบราณโดยใช้การต่อเติมภาพเป็นขั้นตอนของการซ่อมแซมแบบหนึ่งซึ่งไม่ก่อให้เกิดความเสียหายได้ ๆ กับภาพเดิม เนื่องจากเป็นการซ่อมแซมโดยการใช้ขั้นตอนวิธีเชิงตัวเลขบนภาพดิจิตัลซึ่งเป็นสำเนาของภาพเดิม ด้วยเหตุผลดังกล่าว ผู้วิจัยได้เลือกใช้วิธีการซ่อมแซมภาพจิตรกรรมไทยโบราณมีความจำเป็นเร่งด่วน เนื่องจากภาพที่ได้รับการซ่อมแซมด้วยการต่อเติมภาพสามารถนำไปใช้ประกอบการตัดสินใจเพื่อวางแผนก่อนการลงมือซ่อมแซมภาพเขียนจริงได้ นอกจากนี้ ขั้นตอนวิธีการต่อเติมภาพสามารถนำไปใช้สร้างแอปพลิเคชันโทรศัพท์เคลื่อนที่เพื่อในไปใช้เป็นข้อมูลในการเข้าชมภาพเขียนเดิมที่ยัง

ไม่ได้รับการซ่อมแซมและภาพเขียนที่ได้รับการซ่อมแซมโดยวิธีการทางคณิตศาสตร์จากแอปพลิเคชันที่พัฒนาขึ้น

รูปที่ 1.2 แสดงตัวอย่างภาพจิตรกรรมไทย¹ ที่ต้องได้รับการซ่อมแซมบนบริเวณแขนสีอ่อนของรูปปานาผู้ชายที่มีส่วนของสีแดงเดิมหลุดหายไป ทั้งนี้ในการซ่อมแซมภาพโดยการต่อเติมภาพ เราจะเริ่มด้วยการสร้างโคลเมนต่อเติมบนบริเวณสีพื้นผิวปูนที่แขนเสื้อ จากนั้นจึงนำขั้นตอนวิธีการต่อเติมภาพเพื่อซ่อมแซมภาพบริเวณนั้นให้เป็นสีแดง



รูปที่ 1.2: ภาพจิตรกรรมไทยที่วัดภูมินทร์ อำเภอเมือง จังหวัดน่าน

1.2 การลับบทบรรยายบนอนิเมะ

อนิเมะคือวิดีโอภาพวดการ์ตูนสต็อกญี่ปุ่นซึ่งเป็นที่นิยมของเยาวชนไทย ใน การรับชมอนิเมะ แม้ว่าเยาวชนไทยสามารถรับชมด้วยบทพากย์เสียงภาษาไทย แต่ก็สูญเสียอรรถรสของการรับชมจากบทบรรยายแบบแข็ง² (hardsub) ที่เป็นภาษาต่างประเทศในบริเวณด้านล่างของจอภาพ ในการซ่อมแซมอนิเมะด้วยการลับบทบรรยายภาษาต่างประเทศจึงเป็นงานที่ยุ่งยากและท้าทายมาก เนื่องจาก

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

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

²บทบรรยายที่ไม่สามารถปิดหรือเปิดได้

ด้วยความท้าทายข้างต้น การพัฒนาขั้นตอนวิธีการต่อเติมภาพที่สามารถดำเนินต่อเติมเขิงอัตโนมัติให้กับแต่ละเฟรมและประมวลผลได้แม่นยำจากการลับบทบรรยายสามารถทำงานได้แบบเรียลไทม์จึงเป็นสิ่งจำเป็นที่หลีกเลี่ยงไม่ได้

รูปที่ 1.3 แสดงตัวอย่าง 1 เฟรมของอนิเมะที่มีบทบรรยายแบบแข็ง³ ที่ต้องซ่อมแซมด้วยการลับบทบรรยายออก ทั้งนี้ในการลับบทบรรยายออกจากเฟรมโดยใช้การต่อเติมภาพ เราจะเริ่มด้วยการสร้างโดเมนต่อเติมแบบอัตโนมัติในบริเวณบทบรรยาย จากนั้นจึงนำขั้นตอนวิธีการต่อเติมภาพแบบเร็วเพื่อลับบทบรรยายออกจากเฟรม



รูปที่ 1.3: 1 เฟรมของอนิเมะที่มีบทบรรยายแบบแข็ง

โครงการวิจัยนี้ ผู้วิจัยมีเป้าหมายสำคัญคือการพัฒนาขั้นตอนวิธีการต่อเติมภาพแบบเร็วและแม่นยำ ชนิดใหม่เพื่อนำไปใช้สำหรับซ่อมแซมภาพจิตรกรรมไทยและการลับบทบรรยายออกจากอนิเมะ

2 วรรณกรรมและทฤษฎีบทที่เกี่ยวข้อง

ในการกล่าวถึงขั้นตอนวิธีการต่อเติมภาพ จะเริ่มต้นด้วยการกล่าวบททวนเกี่ยวกับการต่อเติมภาพเฉดสีเทา (grayscale image) ก่อน ดังนี้

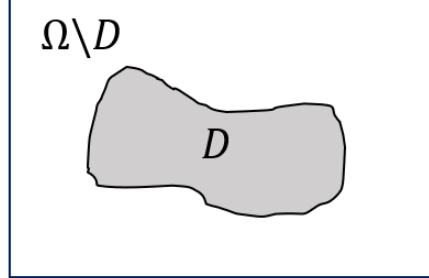
ให้ $\Omega \subset \mathbb{R}^2$ แทนโดเมนภาพ (image domain) $D \subset \mathbb{R}^2$ แทนโดเมนต่อเติม (ดูรูปที่ 2.1) และ $V \subset [0, \infty)$

ให้ $u : \Omega \rightarrow V$, $z : \Omega \rightarrow V$ แทนภาพที่ได้รับการซ่อมแซมและภาพที่ต้องการซ่อมแซม ตามลำดับ

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

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

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



รูปที่ 2.1: D แทนโดเมนต่อเติม

2.1 ตัวแบบการต่อเติมภาพเขตสีเทาที่ใช้การแปรผันรวม

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

$$\min_u \{ \mathcal{J}(u) = \frac{1}{2} \int_{\Omega} \lambda(u - z)^2 d\Omega + \int_{\Omega} |\nabla u| d\Omega \} \quad (2.1)$$

เมื่อ

$$\lambda = \lambda(\mathbf{x}) = \begin{cases} \lambda_0, & x \in \Omega \setminus D \\ 0, & x \in D \end{cases} \quad (2.2)$$

แทนพารามิเตอร์รักษาไว้ในรูปแบบ regularization parameter และ $\lambda_0 > 0$

โดยแคลคูลัสของการแปรผัน (Calculus of variations) จะได้สมการอย่างลักษณะที่เกี่ยวข้องกับ (2.1) เป็น

$$\begin{cases} -\nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) + \lambda(u - z) = 0, & \mathbf{x} \in (1, n)^2 \\ \frac{\partial u}{\partial \mathbf{n}} = 0, & x \in \partial\Omega \end{cases} \quad (2.3)$$

เมื่อ \mathbf{n} แทนเวกเตอร์หน่วยที่ตั้งฉากกับของของภาพ

ต่อไปจะกล่าวทบทวนวิธีการเชิงตัวเลขสำหรับแก้สมการเชิงอนุพันธ์อย่างใน (2.3)

(1) วิธีการเดินเวลาแบบชัดแจ้ง (explicit time marching method)

คณะวิจัย [2] ได้แนะนำวิธีการเชิงตัวเลขสำหรับการกำจัดสัญญาณรบกวนโดยใช้วิธีการเดินเวลาแบบชัดแจ้ง ซึ่งสามารถประยุกต์เป็นวิธีเชิงตัวเลขสำหรับการต่อเติมภาพได้ดังนี้

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

ที่ขึ้นอยู่กับเวลา

$$u(\mathbf{x}, t_{k+1}) = u(\mathbf{x}, t_k) + \tau \left(\nabla \cdot \left(\frac{\nabla u(\mathbf{x}, t_k)}{|\nabla u(\mathbf{x}, t_k)|} \right) + \lambda(\mathbf{x})(u(\mathbf{x}, t_k) - z(\mathbf{x})) \right), \quad u(\mathbf{x}, t_0) = z \quad (2.4)$$

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

จะสามารถเขียน วิธีเดินเวลาแบบชั้ดแจ้งสำหรับภาพเดาเป็นชั้นตอนวิธีได้ดังนี้

Algorithm 1: Explicit time marching gray-scale solver

Input:

u is image which is damaged image

D is image which is image of inpaint domain

λ is positive rational number which is regularization parameter

β is positive rational number which is use to avoid devide by zero

τ is positive rational number which is marching parameter

N is positive interger which is number of maximum loop

ε is positive rational number which is expected relative error

Output: inpainted image

Function `ExplicitTimeMarchingInpaint($u, D, \lambda, \tau, \beta, N, \varepsilon$):`

initialize $\Lambda = \lambda \cdot D$, $z = u$, $i = 0$

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

$$u = u + \tau \left(\nabla \cdot \left(\frac{\nabla u}{\sqrt{u_x^2 + u_y^2 + \beta}} \right) + \Lambda(u - z) \right)$$

$$err = \frac{\|u - z\|}{\|u\|}$$

end

return u

(2) วิธีการทำข้ามแบบจุดตรึง (fixed-point iteration method)

คณะวิจัย [3] ได้แนะนำวิธีการเชิงตัวเลขสำหรับการทำจัดสัญญาณรบกวนโดยใช้วิธีการทำข้ามแบบจุดตรึง ซึ่งสามารถประยุกต์เป็นวิธีเชิงตัวเลขสำหรับการทำภาพได้ดังนี้

เริ่มจากการแนะนำดัชนีการทำข้ามแบบจุดตรึง $\nu = 0, 1, 2, \dots$ และนิยามรูปแบบการทำข้ามโดย

$$-\nabla \cdot \left(\frac{\nabla u^{[\nu+1]}}{|\nabla u|^{[\nu]}} \right) + \lambda(u^{[\nu+1]} - z) = 0, \quad u^{[0]} = z \quad (2.5)$$

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 (2.4) และ (2.5) เราจะใช้

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

จะสามารถเขียน ขั้นตอนวิธีจุดตรึงสำหรับภาพเขตเทาเป็นขั้นตอนวิธีได้ดังนี้

Algorithm 2: Fixed point gray-scale solver

Input:

u is image which is damaged image
 D is image which is image of inpaint domain
 λ is positive rational number which is regularization parameter
 β is positive rational number which is use to avoid devide by zero
 g is positive integer which is number of gauss seidel iteration
 N is positive interger which is number of maximum loop
 ε is positive rational number which is expected relative error

Output: inpainted image

Function `FixedPointInpaint($u, D, \lambda, \beta, g, N, \varepsilon$):`

```

while  $i < N$  and  $err > \varepsilon$  do
     $u = \text{InnerFixedPointInpaint}(u, D, \lambda, \beta, g)$ 
     $u = \left( \nabla \cdot \left( \frac{\nabla u}{\sqrt{u_x^2 + u_y^2 + \beta}} \right) + (z - u) \right)$ 
     $err = \frac{\|u - z\|}{\|u\|}$ 
     $i = i + 1$ 
end
return  $u$ 
Function InnerFixedPointInpaint( $u, D, \lambda, \beta, g, N, \varepsilon$ ):
initialize  $\Lambda = \lambda \cdot D$ ,  $height = \text{height of } u$ ,  $width = \text{width of } u$ ,  $z = u$ ,
 $k = 0$ ,  $d = \frac{1}{\sqrt{u_x^2 + u_y^2 + \beta}}$ ,  $h = 1$ 
for  $k = 0$ ;  $k < g$ ;  $k = k + 1$  do
    for  $i = 0$ ;  $i < height$ ;  $i++$  do
        for  $j=0$ ;  $j < width$ ;  $j++$  do
             $u(i, j) = \frac{\Lambda_{i,j} z_{i,j} + \frac{1}{h^2} (d_{i,j}(u_{i+1,j} + u_{i,j+1}) + d_{i-1,j}u_{i-1,j} + d_{i,j-1}u_{i,j-1})}{\Lambda_{i,j} + \frac{1}{h^2} (2d_{i,j} + d_{i-1,j} + d_{i,j-1})}$ 
        end
    end
end
return  $u$ 

```

จาก (2.4) และ (2.5) เรายกเวลาก่อนว่า β มีค่าน้อยลงมากขึ้นเท่าไหร่ ความแม่นยำของตัวแบบ (2.1) ยิ่งมีมากขึ้นเท่านั้น นอก จากนี้ รายังพบอีกว่า การแก้สมการ (2.4) และ (2.5) ยิ่งมีความยุ่งยากมากขึ้นสำหรับ β ที่มีค่าน้อยๆ

เพื่อเอาชนะความยากเชิงตัวเลขนี้ คณะวิจัยโดย [4] ได้แนะนำวิธีการสปริทเบรกแม่นซึ่งสามารถกล่าวถึงพอกสังเขป ดังนี้

(3) วิธีการสปริทเบรกแม่น (Split Bregman method)

เริ่มจากการແນະນຳເວກເຕອີຣ໌ເສັ່ນ \mathbf{w} ພາຣາມີເຕອີຣ໌ເບຣກແມນ (Bregman parameter) \mathbf{b} ແລະ ພາຣາມີເຕອີຣ໌ພິ້ນໜີ (panalty parameter) $\theta > 0$ ແລະ ເປີຍນ (2.1) ໃໝ່ ດັ່ງນີ້

$$\min_{u, \mathbf{w}} \{ \mathcal{J}(u, \mathbf{w}) = \frac{1}{2} \int_{\Omega} \lambda(u - z)^2 d\Omega + \int_{\Omega} |\nabla \mathbf{w}| d\Omega + \frac{\theta}{2} \int_{\Omega} (\mathbf{w} - \nabla u + \mathbf{b}) d\Omega \} \quad (2.6)$$

ສໍາຫຼັບການหาຄຳຕອບຂອງ (2.6) ເຮັດວຽກທີ່ສຸດແບບສລັບ (alternating minimization method) ໂດຍເຮັດວຽກການຕີ້ງ \mathbf{w}^{old} ແລະ \mathbf{b}^{old} ຈາກນັ້ນແກ້ປຸ່ນຫຍ່ຍ່ອຍ

$$u^{\text{New}} = \arg \min_u \{ \mathcal{J}_1(u) = \frac{1}{2} \int_{\Omega} \lambda(u - z)^2 d\Omega + \frac{\theta}{2} \int_{\Omega} (\mathbf{w}^{\text{old}} - \nabla u + \mathbf{b}^{\text{old}}) d\Omega \} \quad (2.7)$$

ຈາກນັ້ນໃຊ້ u^{New} ທີ່ໄດ້ຈາກການແກ້ປຸ່ນຫຍ່ຍ່ອຍໃນ (2.7) ເພື່ອແກ້ປຸ່ນຫຍ່ຍ່ອຍ

$$\mathbf{w}^{\text{New}} = \arg \min_{\mathbf{w}} \{ \mathcal{J}_2(\mathbf{w}) = \int_{\Omega} |\nabla \mathbf{w}| d\Omega + \frac{\theta}{2} \int_{\Omega} (\mathbf{w} - \nabla u^{\text{New}} + \mathbf{b}^{\text{old}}) d\Omega \} \quad (2.8)$$

ສຸດທ້າຍຈຶ່ງປະບົງພາຣາມີເຕອີຣ໌ເບຣກແມນ

$$\mathbf{b}^{\text{New}} = \mathbf{b}^{\text{old}} + \nabla u^{\text{New}} - \mathbf{w}^{\text{New}} \quad (2.9)$$

ດຳເນີນການເຂັ້ນນີ້ຈະຮ່າງ $\|u^{\text{new}} - u^{\text{old}}\| < \epsilon_1$ ອີ່ວນ $\text{New} > \epsilon_2$ ເນື້ອ $\epsilon_1, \epsilon_2 > 0$ ຈະສາມາດ
ເປີຍນ ຂັ້ນຕອນວິທີສປິທບຣກແມນສໍາຫຼັບກາພເຊີດເທົາເປັນຂັ້ນຕອນວິທີໄດ້ດັ່ງນີ້

Algorithm 3: Split-bergman gray-scale solver

Input:

u is image which is damaged image

D is image which is image of inpaint domain

λ is positive rational number which is regularization parameter

θ is positive rational number which is panelty parameter that shouldn't too big or too small

N is positive interger which is number of maximum loop

g is positive integer which is number of gauss seidel iteration

ε is positive rational number which is expected relative error

Output: inpainted image

Function `SplitBergmanInpaint($u, D, \lambda, \theta, g, N, \varepsilon$)`:

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

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

$v = u$

$w = wSubproblemSolver(u, b, \theta)$

$u = uSubproblemSolver(u, w, v, b, D, \lambda, \theta, g)$

$b = b + -w$

$err = \frac{\|u-v\|}{\|u\|}$

$i = i + 1$

end

return u

Function `uSubproblemSolver($(u, w, z, b, D, \lambda, \theta, gn)$)`:

initialize $b = \vec{0}, w = \vec{0}, v = u, width = \text{width of } u, height = \text{height of } u,$

$d = \left(\nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) + \Lambda(u - z) \right) \Lambda = \lambda \cdot D, h = 1$

for $k = 0; k < g; i++$ **do**

for $i = 0; i < height; i++$ **do**

for $j = 0; j < width; j++$ **do**

$L = \frac{u_{i,j-1} + u_{i,j+1} + u_{i-1,j} + u_{i+1,j}}{h^2}$

$B = \Lambda_{i,j} z_{i,j} - \theta d_{i,j} + \theta L_{i,j}$

$u_{i,j} = \frac{h^2}{\Lambda_{i,j} * h^2 + 4 * \theta} B$

end

end

end

return u

Function `wSubproblemSolver(u, b, θ)`:

$w = \max(|\nabla b| - \frac{1}{\theta}, 0)$

return w

2.2 ตัวแบบการต่อเติมภาพสีที่ใช้การแปรผันรวม

ต่อไปเราจะพิจารณาภาพสีในระบบ 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 และสีฟ้าเงินของ \mathbf{u}, \mathbf{z} ตามลำดับ

ในทำนองเดียวกันกับตัวแบบการต่อเติมภาพเขตสีเทาที่ใช้การแปรผันรวม ตัวแบบการต่อเติมภาพสีที่ใช้การแปรผันรวมสามารถเขียนได้ดังนี้

$$\min_{\mathbf{u}} \{\bar{\mathcal{J}}(\mathbf{u}) = \bar{\mathcal{D}}(\mathbf{u}, \mathbf{z}) + \bar{\mathcal{R}}(\mathbf{u})\} \quad (2.10)$$

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

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

ดังนั้นเพื่อต่อเติมภาพสี จะเป็นการแก้ปัญหาการหาค่าต่าที่สุดต่อไปนี้

$$\begin{aligned} \min_{\mathbf{u}, \mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3} \{ \bar{\mathcal{J}}(\mathbf{u}, \mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3) &= \bar{\mathcal{D}}(\mathbf{u}, \mathbf{z}) + \sum_{l=1}^3 \int_{\Omega} |\mathbf{w}_l| d\Omega \\ &+ \frac{\theta_l}{2} \sum_{l=1}^3 \int_{\Omega} (\mathbf{w}_l - \nabla u_l - \mathbf{b}_l)^2 d\Omega \}, \quad \theta_l > 0 \end{aligned} \quad (2.11)$$

ซึ่งสามารถประยุกต์ใช้ริมภาพเขตเทา มาใช้งานกับภาพสีได้ดังต่อไปนี้

สำหรับวิธีการเดินเวลา สามารถประยุกต์จากขั้นตอนสำหรับภาพเอดเทาเป็นขั้นตอนสำหรับภาพสีได้ดังนี้

Algorithm 4: Explicit time marching color image solver

Input:

u is image which is damaged image
 D is image which is image of inpaint domain
 λ is positive rational number which is regularization parameter
 β is positive rational number which is use to avoid devide by zero
 τ is positive rational number which is marching parameter
 N is positive interger which is number of maximum loop
 ε is positive rational number which is expected relative error

Output: inpainted image

Function *ExplictTimeMarchingColorInpaint*($u, D, \lambda, \tau, \beta, N, \varepsilon$):

```
foreach  $l, l \in \{1, 2, 3\}$  do
    |  $u_l = ExplictTimeMarchingColorInpaint(u_l, D, \lambda, \tau, \beta, N, \varepsilon)$ 
end
return  $u$ 
```

ในทำนองเดียวกัน จะได้ว่าขั้นตอนการต่อเติมภาพสำหรับวิธีการตressing คือ

Algorithm 5: Fixed point color solver

Input:

u is image which is damaged image
 D is image which is image of inpaint domain
 λ is positive rational number which is regularization parameter
 β is positive rational number which is use to avoid devide by zero
 g is positive integer which is number of gauss seidel iteration
 N is positive interger which is number of maximum loop
 ε is positive rational number which is expected relative error

Output: inpainted image

Function *FixedPointColorInpaint*($u, D, \lambda, \beta, g, N, \varepsilon$):

```
foreach  $l, l \in \{1, 2, 3\}$  do
    |  $u_l = FixedPointInpaint(u_l, D, \lambda, \tau, \beta, N, \varepsilon)$ 
end
return  $u$ 
```

และขั้นตอนวิธีสปริทเบรกแม่น สำหรับภาพสี

Algorithm 6: Split-bergman Color solver

Input:

u is image which is damaged image

D is image which is image of inpaint domain

λ is positive rational number which is regularization parameter

θ is positive rational number which is panelty parameter that shouldn't too big or too small

N is positive interger which is number of maximum loop

g is positive integer which is number of gauss seidel iteration

ε is positive rational number which is expected relative error

Output: inpainted image

Function SplitBergmanColorInpaint($u, D, \lambda, \theta, g, N, \varepsilon$):

foreach $l, l \in \{1, 2, 3\}$ **do**

$| u_l = \text{SplitBergmanInpaint}(u_l, D, \lambda, \tau, \beta, N, \varepsilon)$

end

return u

2.3 การวัดประสิทธิภาพของภาพที่ผ่านกระบวนการต่อเติม

หลังจากการต่อเติมภาพแล้วจำเป็นต้องพิจารณาว่าการวัดคุณภาพของภาพที่ผ่านการต่อเติมดีมากน้อยเพียงใด โดยในวิจัยนี้จะสนใจคุณภาพของค่าในแต่ละพิกเซลที่ใกล้เคียงกับภาพต้นฉบับ และโครงสร้างโดยรวมที่ใกล้เคียงกับภาพต้นฉบับ โดยการวัดค่าดังต่อไปนี้

2.3.1 Peak Signal Noise Ratio: PSNR

Peak signal-to-noise ratio (PSNR) [6] ใช้สำหรับวัดคุณภาพของภาพโดยเปรียบเทียบจากพิกเซลแต่ละพิกเซล โดยภาพที่มีความคล้ายต้นฉบับจะมีค่า PSNR เข้าใกล้ล้านนต์ หรือก็คือยิ่งมีค่ามากยิ่งคุณภาพดีซึ่งสามารถคำนวณได้โดย

$$PSNR = 10 \cdot \log_{10} \left(\frac{\text{peak}^2}{\sqrt{MSE}} \right)$$

เมื่อ MSE คือ mean square error และ $peak$ คือค่าสูงสุดโดยประเภทของภาพ ซึ่งสำหรับงานที่จะพูดถึงต่อไปนี้ จะพิจารณาภาพเป็นฟังก์ชันที่มีความเข้มของภาพอยู่ในช่วง $[0, 1]$ จึงได้ว่า $peak$ มีค่าเป็น 1

2.3.2 Structural Similarity: SSIM

Structural similarity (SSIM) [7] ใช้สำหรับว่าวัดคุณภาพของภาพจากโครงสร้างของภาพ โดยพิจารณาว่าภาพนั้นมีโครงสร้างแตกต่างหรือคล้ายคลึงกับภาพต้นฉบับมากน้อยเพียงใด โดยมีค่าอยู่ระหว่าง 0 ถึง 1 หากทั้งสองภาพมีความคล้ายคลึงกันมากค่า SSIM จะเข้าใกล้กับค่า 1 ซึ่ง SSIM นั้นสามารถคำนวณได้

โดย

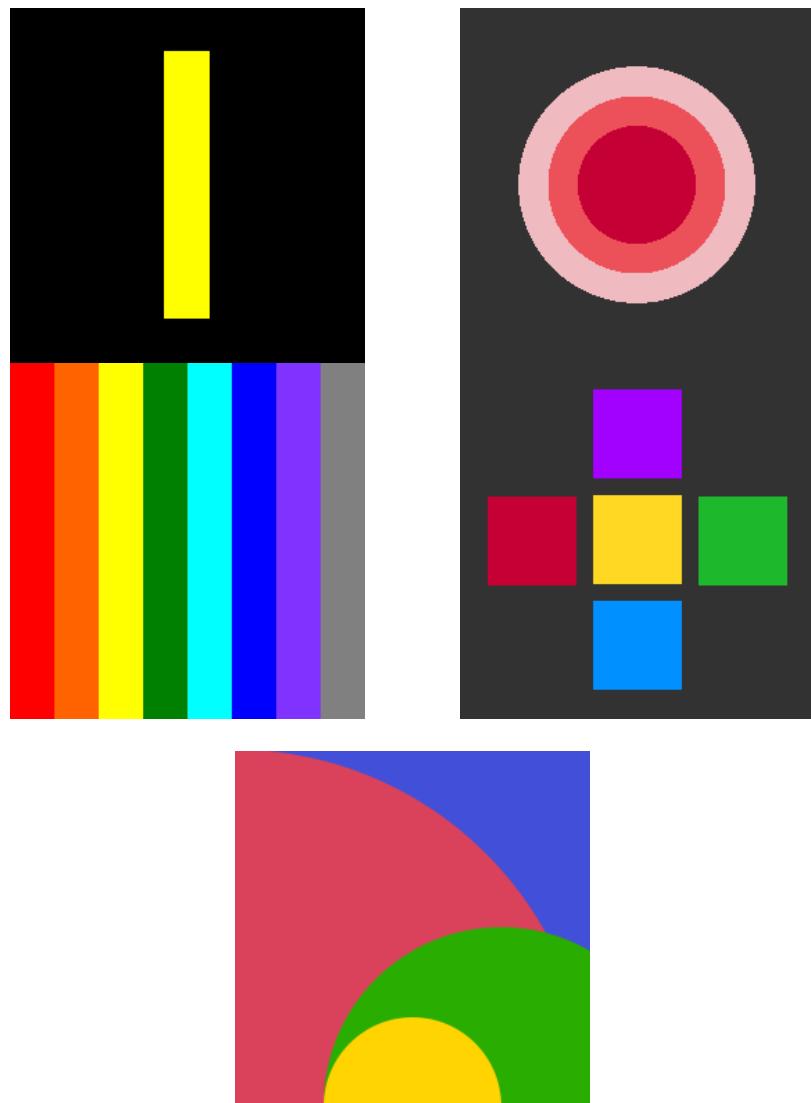
$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

เมื่อ x, y คือภาพที่ต้องการเปรียบเทียบ μ คือค่าเฉลี่ยของภาพ σ^2 คือค่าความแปรปรวนของภาพ σ_{xy} คือความแปรปรวนร่วม $c_1 = (0.01L)^2, c_2 = (0.03L)^2$ และ L คือค่าสูงสุดโดยประเภทของภาพ ซึ่งสำหรับงานที่จะพูดถึงต่อไปนี้ จะพิจารณาภาพเป็นฟังก์ชันที่มีความเข้มของภาพอยู่ในช่วง $[0, 1]$ จึงได้ว่า L มีค่าเป็น 1

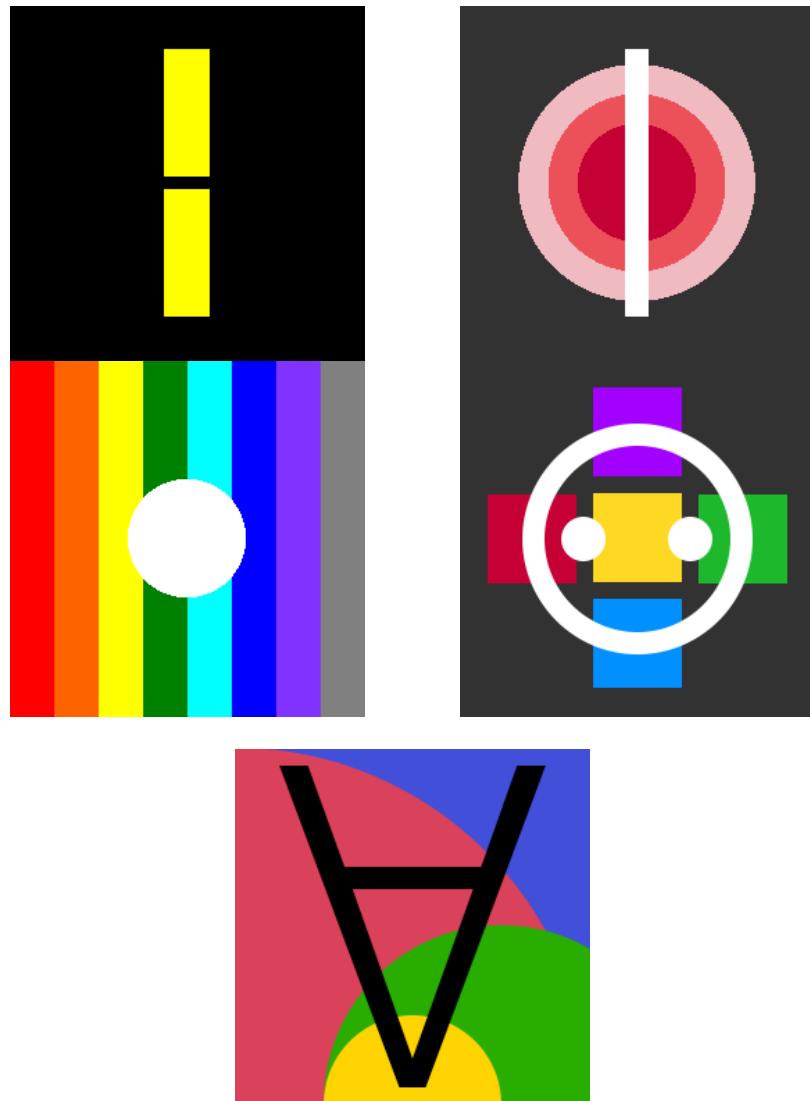
3 ผลการดำเนินงานเบื้องต้น

3.1 การซ้อมแซมภาพจิตรกรรมไทยโบราณ

สำหรับการซ้อมแซมจิตรกรรมไทยโบราณ ก่อนอื่นจะทำการปรับปรุงขั้นตอนวิธีเชิงตัวเลขที่มีอยู่แล้วเดิมก่อน โดยระหว่างการปรับปรุงวิธีเชิงตัวเลข จะใช้ภาพที่ได้สร้างขึ้น 5 ภาพ แต่ละภาพมีขนาด 256×256 พิกเซล ซึ่งมีดังนี้



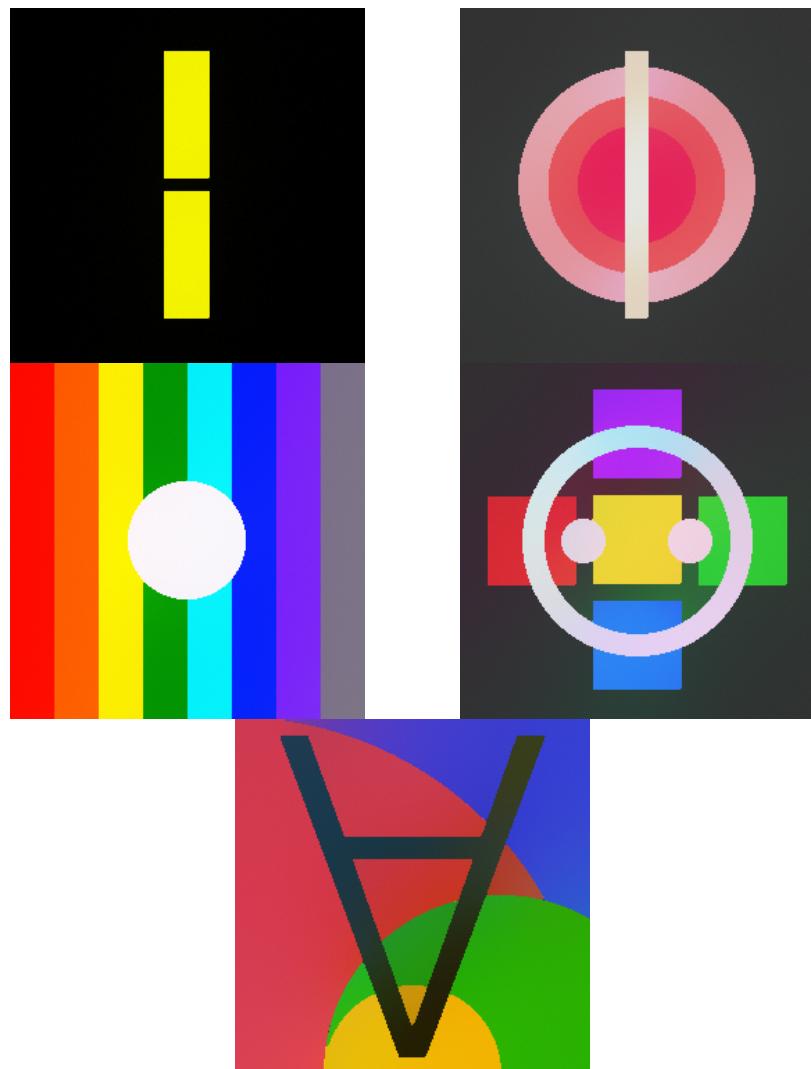
รูปที่ 3.1: ภาพต้นฉบับ



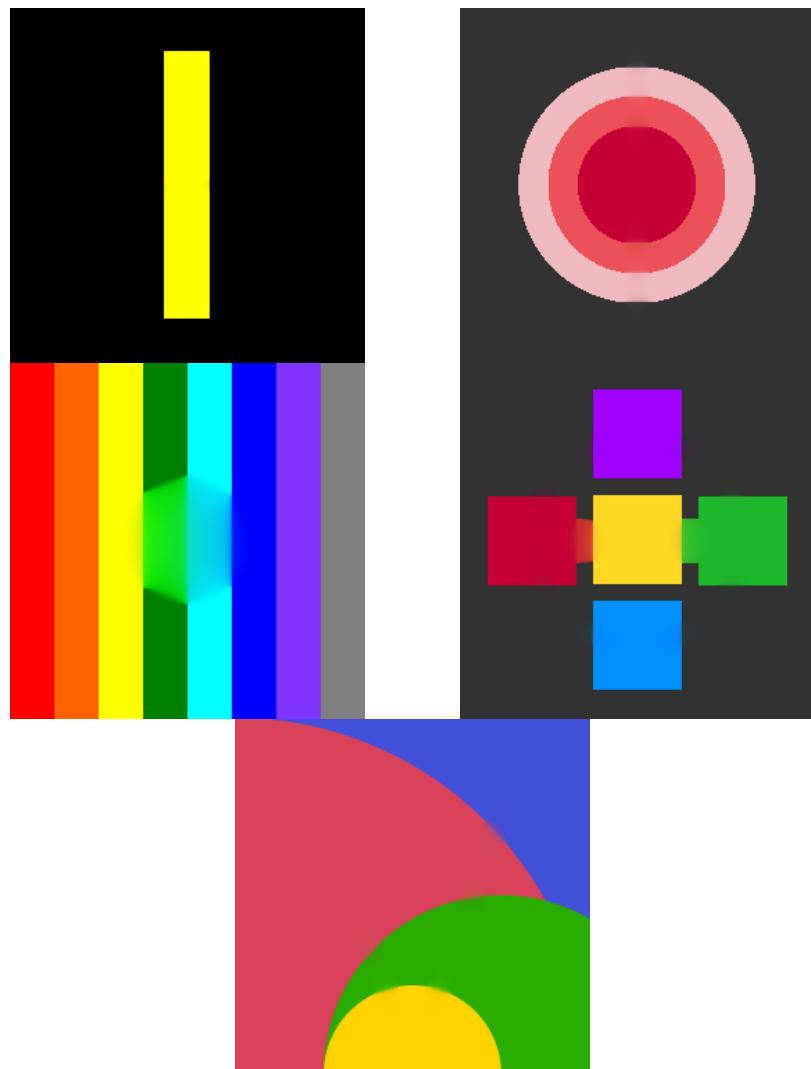
รูปที่ 3.2: ภาพที่จะทำการซ่อมแซม

3.1.1 การเปรียบเทียบประสิทธิภาพขั้นตอนวิธีเชิงตัวเลขที่มีอยู่สำหรับตัวแบบต่อเติมภาพสีที่ใช้การแปรผันรวม

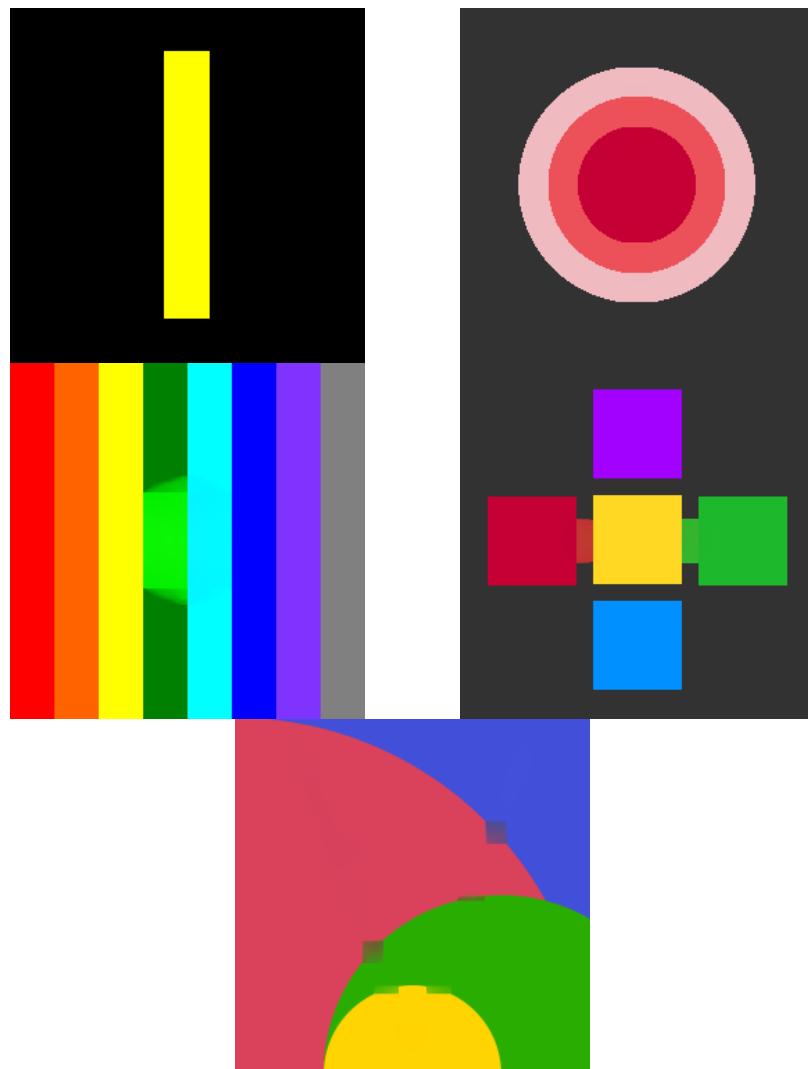
โดยจะทำการเปรียบเทียบความเร็วของวิธีการแก้การแปรผันรวมที่มีอยู่เดิม ดังที่ได้กล่าวไว้ในหัวข้อ 2.1 และหัวข้อ 2.2 โดยวิธีการทั้ง 3 วิธี จะทำการทำซ้ำจนกระทั่งภาพในการทำซ้ำรอบปัจจุบัน กับภาพการทำซ้ำในครั้งก่อนหน้า มีค่าความคลาดเคลื่อนสัมพัทธ์ (relative error) ต่างกันไม่เกิน 0.0001 หรือ ทำซ้ำเกิน 10,000 รอบ ซึ่งได้ผลลัพธ์เฉลี่ยของรูปภาพที่ใช้ทดสอบดังนี้



รูปที่ 3.3: ผลลัพธ์จากการเดินเวลา



รูปที่ 3.4: ผลลัพธ์จากการจุดต้อง



รูปที่ 3.5: ผลลัพธ์จากการสปริทเบรกแม่น

และได้ผลลัพธ์ดังตารางนี้

วิธีการ	รูปภาพ	เวลาประมาณ (วินาที)	PSNR (dB)	SSIM
การเดินเวลา	1			
	2			
	3			
	4			
	5			
การทำข้าจุดตึง	1			
	2			
	3			
	4			
	5			
การสปริทเบรกแม่น	1			
	2			
	3			
	4			
	5			

ตารางที่ 1: แสดงผลลัพธ์ของวิธีการใช้ตัวเลขสำหรับการต่อเติมภาพ

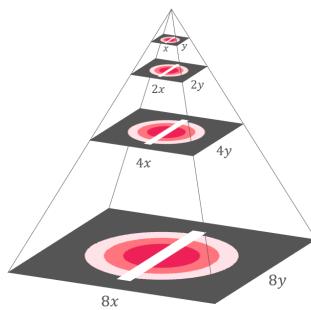
วิธีการ	เวลาประมาณ (วินาที)	PSNR (dB)	SSIM
การเดินเวลา			
การทำข้าจุดตึง			
การสปริทเบรกแม่น			

ตารางที่ 2: แสดงผลลัพธ์เฉลี่ยของวิธีการใช้ตัวเลขสำหรับการต่อเติมภาพ

รอใส่สรุปใหม่

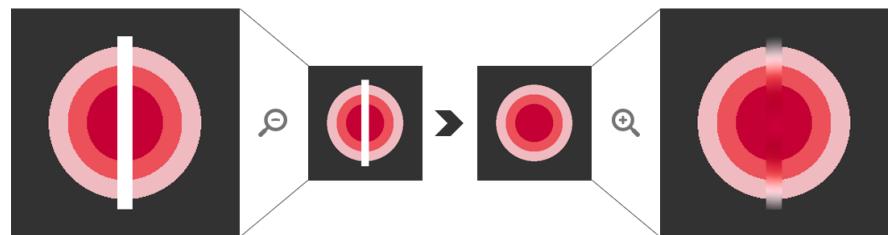
3.1.2 ขั้นตอนวิธีการสำหรับต่อเติมภาพชนิดใหม่

จากวิธีการสปริทเบรกแม่นนั้นจะใช้วิธีการหาคำตอบโดยวิธีการทำข้า江南กระทั้งคู่เข้า ทางผู้ศึกษาจึงสนใจที่หาคำตอบเริ่มต้นสำหรับการทำข้าที่ดีขึ้น เพื่อทำให้การทำข้าลู่เข้าสู่คำตอบได้เร็วขึ้น โดยการทำงานกับรูปภาพที่เล็กกว่า จากนั้นจึงทำการขยายลัพธ์ที่ได้ขึ้นมาทำกับภาพใหญ่ ซึ่งวิธีนี้เรียกว่าวิธีพิรมิตรูปภาพ (pyramid methods) [5] โดยผู้วิจัยจะทำการย่อขนาดรูปลงครึ่งหนึ่งโดยใช้วิธี Bilinear Interpolation ทั้งสิ้น 4 ครั้ง จากนั้นเริ่มทำการต่อเติมภาพขนาดเล็ก จากนั้นนำผลลัพธ์ที่ได้จากการขยายภาพขึ้นสองเท่าโดยใช้ Bilinear Interpolation ก่อนจะนำเฉพาะส่วนที่อยู่ในโดเมนต่อเติมของภาพที่ถูกขยายมาทำการต่อเติมเพื่อให้ส่วนที่ถูกขยายขึ้นมาเป็นคำตอบเริ่มต้นสำหรับการต่อเติมภาพในขั้นที่สูงขึ้น



รูปที่ 3.6: วิธีการพิริมิดรูปภาพ

อย่างเช่นภาพที่ 3.7 จากรูปที่ต้องการซ่อมแซมทางด้านข่าย จะทำการย่อภาพให้เหลือขนาดเล็กเพื่อจะได้ทำการต่อเติมภาพได้อย่างรวดเร็ว หลังจากนั้นเมื่อทำการต่อเติมภาพขนาดเล็กเสร็จแล้ว จะทำการขยายภาพขึ้นแล้วทำการนำพื้นที่จากภาพที่ขยายขึ้นไปแทนในโดเมนต่อเติมแล้ว จะได้ภาพนั้น ในส่วนที่เป็นโดเมนต่อเติมจะมีลักษณะเบลอ แต่มีสีใกล้เคียงกับคำตอบที่ต้องการ ซึ่งหลังจากนี้จะนำภาพทางด้านขวาไปทำการต่อเติมจนได้คำตอบที่ต้องการ โดยเมื่อใช้สีที่เบลอแต่มีสีใกล้เคียงเป็นคำตอบเริ่มต้นที่ใช้เวลาการประมวลผลน้อยกว่าการนำภาพไปต่อเติมโดยไม่ผ่านการใช้พิริมิด



รูปที่ 3.7: การเตรียมคำตอบเริ่มต้นจากภาพขนาดเล็ก

Algorithm 7: Split-bergman Color solver with Image Pyramid (Multi Resolution)

Input:

u is image which is damaged image

D is image which is image of inpaint domain

λ is positive rational number which is regularization parameter

θ is positive rational number which is panelyt parameter that shouldn't too big or too small

N_0 is positive interger which is number of maximum iteration on smallest size image

N_1 is positive interger which is number of maximum iteration on image that isn't the biggest neither smallest

N_2 is positive interger which is number of maximum iteration on biggest size image

g is positive integer which is number of gauss seidel iteration

ε is positive rational number which is expected relative error

Output: inpainted image

Function

MultiSplitBergmanColorInpaint($u, D, \lambda, \theta, g, N_0, N_1, N_2, \varepsilon, c, m$):

Initialize $height =$ height of u , $width =$ width of u

if $c < M$ **then**

$x = \text{BilinearResize}(u, \lfloor width * 0.5 \rfloor, \lfloor height * 0.5 \rfloor)$

$y = \text{BilinearResize}(D, \lfloor width * 0.5 \rfloor, \lfloor height * 0.5 \rfloor)$

$r =$

$\text{MultiSplitBergmanColorInpaint}(x, y, \lambda, \theta, g, N_0, N_1, N_2, \varepsilon, c+1, m)$

$R = \text{BilinearResize}(r, width, height)$

$u = \text{MergeResult}(u, D, R)$

end

if $c \neq 1$ **then**

if $1 = m$ **then**

 | $N_0 = N_1$

else

 | $N_0 = N_2$

end

end

return $\text{SplitBergmanColorInpaint}(u_l, D, \lambda, \tau, \beta, N_0, \varepsilon)$

Function **MergeResult**(u, D, R):

Initialize $height =$ height of u , $width =$ width of u

for $i = 0; i < height; i++$ **do**

for $j = 0; j < i; j++$ **do**

if $D_{i,j} \neq 0$ **then**

 | $u_{i,j} = R_{i,j}$

end

end

end

return u

Algorithm 8: Bilinear Interpolation for Image resizing

```
Function BilinearResize( $I, x, y$ ):  
    Initialize  $J$  is image that height  $x$  and width  $y$ ,  
     $v = \text{height of } I$  and  $w = \text{width of } I$ ,  
     $S_R = \frac{c}{a}, S_C = \frac{d}{b}, r = 1, 2, \dots, v, c = 1, 2, \dots, w,$   
     $r' = 1, 2, \dots, x, c' = 1, 2, \dots, y,$   
     $r_f = \lfloor r' \cdot S_R \rfloor$   
     $c_f = \lfloor c' \cdot S_C \rfloor$   
     $\Delta r = r_f - r$   
     $\Delta c = c_f - c$   
     $J(r', c') = I(r, c) \cdot (1 - \Delta r) \cdot (1 - \Delta c)$   
     $+ I(r + 1, c) \cdot \Delta r \cdot (1 - \Delta c)$   
     $+ I(r, c + 1) \cdot (1 - \Delta r) \cdot \Delta c$   
     $+ I(r + 1, c + 1) \cdot \Delta r \cdot \Delta c$   
    return  $J$ 
```

โดยจะทำการเบรียบเทียบจำนวนครั้งในชั้นที่รูปภาพมีขนาดเล็ก จนไปถึงชั้นที่มีขนาดใหญ่ ตัวอย่าง เช่น 10/3/3/10000 หมายถึงในชั้นเล็กสุดซึ่งขนาดเป็น 32x32 พิกเซลจะทำซ้ำ 10 ครั้งหรือจะค่าความคลาดเคลื่อนสัมพัทธ์ต่างกันไม่เกิน 0.0001 ชั้นถัดมาขนาดเป็น 64x64 จะทำซ้ำ 3 ครั้งหรือจะค่าความคลาดเคลื่อนสัมพัทธ์ต่างกันไม่เกิน 0.0001 และชั้นถัดมา ชั้นถัดมาขนาดเป็น 128x128 จะทำซ้ำ 3 ครั้งหรือจะค่าความคลาดเคลื่อนสัมพัทธ์ต่างกันไม่เกิน 0.0001 และสุดท้ายขนาด 256x256 จะทำซ้ำ 3 ครั้งหรือจะค่าความคลาดเคลื่อนสัมพัทธ์ต่างกันไม่เกิน 0.0001 ซึ่งเมื่อทำการทดสอบแล้วได้ผลลัพธ์ดังตารางนี้

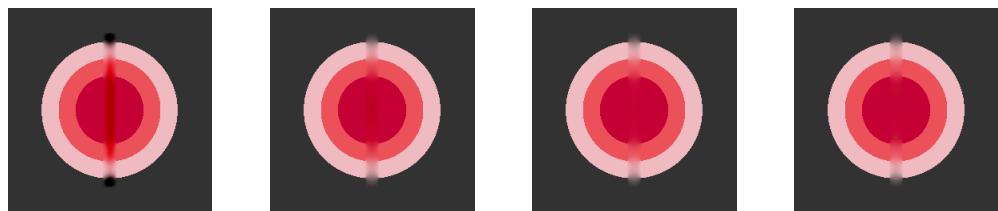
รูปแบบการทำซ้ำ	รูปภาพ	เวลาประมวล (วินาที)	PSNR (dB)	SSIM
ไม่ใช้พิธีกรรมรูปภาพ	1 2 3 4 5			
10/1/1/10000	1 2 3 4 5			
10/3/3/10000	1 2 3 4 5			
10/10/10/10000	1 2 3 4 5			
100/1/1/10000	1 2 3 4 5			
100/3/3/10000	1 2 3 4 5			
100/10/10/10000	1 2 3 4 5			

ตารางที่ 3: แสดงผลลัพธ์ของการใช้พิธีกรรมภาพในการต่อเติมในชั้นที่ต่างกัน

รูปแบบการทำข้าม	เวลาประมาณ (วินาที)	PSNR (dB)	SSIM
ไม่ใช้พิธีร์มิดรูปภาพ			
10/1/1/10000			
10/3/3/10000			
10/10/10/10000			
100/1/1/10000			
100/3/3/10000			
100/10/10/10000			

ตารางที่ 4: แสดงผลลัพธ์เฉลี่ยของการใช้พิธีร์มิดภาพในการต่อเติมในชั้นที่ต่างกัน

จากตารางจะสังเกตว่า ยิ่งจำนวนการทำข้ามในชั้นที่รูปภาพมีขนาดเล็กจำนวนมาก ยิ่งประมาณผลได้เร็วขึ้น นอกเหนือจากนี้แล้ว ผู้วิจัยยังได้สังเกตอีกว่า การทำข้ามนี้ จะถูกเข้าเร็วในช่วงแรก จากนั้นความเร็วในการถูกเข้าจะลดลง ซึ่งทำให้การทำข้ามเพียงไม่กี่ครั้งในรูปภาพขนาดใหญ่สุด มีผลลัพธ์ใกล้เคียงกับภาพต้นฉบับได้



(a) ชั้นใหญ่สุด 10 ครั้ง (b) ชั้นใหญ่สุด 100 ครั้ง (c) ชั้นใหญ่สุด 1,000 ครั้ง (d) ชั้นใหญ่สุด 10,000 ครั้ง

รูปที่ 3.8: พิธีร์มิดที่ลำดับการทำข้ามเป็น 10/10/10 และชั้นใหญ่สุดทำในจำนวนครั้งที่ต่างกัน

โดยผู้วิจัยจึงกำหนดให้การทำข้ามในรูปภาพขนาดใหญ่สุดมีการทำข้ามเพียง 10 ครั้ง และพบว่าได้ผลลัพธ์ดังนี้

รูปแบบการทำซ้ำ	รูปภาพ	เวลาประมวล (วินาที)	PSNR (dB)	SSIM
ไม่ใช้พิธีมิดรูปภาพ	1 2 3 4 5			
10/1/1/10000	1 2 3 4 5			
10/3/3/10000	1 2 3 4 5			
10/10/10/10000	1 2 3 4 5			
100/1/1/10000	1 2 3 4 5			
100/3/3/10000	1 2 3 4 5			
100/10/10/10000	1 2 3 4 5			

ตารางที่ 5: แสดงผลลัพธ์ของการใช้พิธีมิดภาพในการต่อเติมโดยกำหนดให้ขั้นรูปภาพขนาดใหญ่สุดทำซ้ำเพียง 10 ครั้ง

รูปแบบการทำข้าม	เวลาประมาณ (วินาที)	PSNR (dB)	SSIM
ไม่ใช้พิธีมิดรูปภาพ			
10/1/1/10			
10/3/3/10			
10/10/10/10			
100/1/1/10			
100/3/3/10			
100/10/10/10			

ตารางที่ 6: แสดงผลลัพธ์เฉลี่ยของการใช้พิธีมิดภาพในการต่อเติมโดยกำหนดให้ชั้นรูปภาพขนาดใหญ่สุดทำข้ามเพียง 10 ครั้ง

จากตารางจะเห็นว่า การทำข้ามในชั้นที่รูปภาพมีขนาดเล็กมากจำนวนมาก ไม่ช่วยให้การประมวลผลได้เร็วขึ้น ผู้วิจัยจึงเลือกใช้การทำข้ามแบบ 10/3/3/10 ใน การต่อเติมภาพ

3.1.3 การทดสอบประสิทธิภาพในการซ่อมแซมภาพจิตกรรมไทยโบราณ

ซึ่งภาพจิตกรรมที่ใช้ทดสอบ มีทั้งสิ้น 5 ภาพได้แก่ ภาพที่ 3.9a⁴ และภาพที่ 3.9b⁵ คือ จิตกรรมฝาผนังวัดแก้วไพทุรย์ ภาพที่ 3.9c⁶ คือ จิตกรรมฝาผนังวัดพระยืนพุทธบาทยุคล ภาพที่ 3.9d⁷ คือ จิตกรรมฝาผนังวัดคงคาราม และภาพที่ 3.9e⁸ คือ จิตกรรมฝาผนังวัดท่าถนน โดยจะทำให้มูลข้อมูลของทั้ง 5 ภาพเกิดความเสียหาย โดยใช้ร้อยความเสียหายจากภาพพระเจ้าสร้างอดัม

⁴ภาพถ่ายที่วัดแก้วไพทุรย์; ภาพจาก [https://commons.wikimedia.org/wiki/File:จิตกรรมฝาผนัง_วัดแก้วไพทุรย์_\(7\).jpg](https://commons.wikimedia.org/wiki/File:จิตกรรมฝาผนัง_วัดแก้วไพทุรย์_(7).jpg) สืบคันเมื่อวันที่ 23 กันยายน 2561

⁵ภาพถ่ายที่วัดแก้วไพทุรย์; ภาพจาก [https://commons.wikimedia.org/wiki/File:จิตกรรมฝาผนัง_วัดแก้วไพทุรย์_\(2\).jpg](https://commons.wikimedia.org/wiki/File:จิตกรรมฝาผนัง_วัดแก้วไพทุรย์_(2).jpg) สืบคันเมื่อวันที่ 23 กันยายน 2561

⁶ภาพถ่ายที่วัดพระยืนพุทธบาทยุคล; ภาพจาก https://commons.wikimedia.org/wiki/File:Wat_Phra_Yuen_Photthabat_Yukhon_01.jpg สืบคันเมื่อวันที่ 23 กันยายน 2561

⁷ภาพถ่ายที่วัดคงคาราม; ภาพจาก https://commons.wikimedia.org/wiki/File:จิตกรรม_อุโบสถวัดคงคาราม.JPG สืบคันเมื่อวันที่ 23 กันยายน 2561

⁸ภาพถ่ายที่วัดท่าถนน; ภาพจาก https://commons.wikimedia.org/wiki/File:Wat_Tha_Thanon_05.JPG สืบคันเมื่อวันที่ 23 กันยายน 2561



(a) วัดแก้วไพทุรย์



(b) วัดแก้วไพทุรย์



(c) วัดพระยืนพุทธบาทยุคล



(d) วัดคงカラาม



(e) วัดท่าถนน



(f) รอยความเสียหาย



รูปที่ 3.10: ภาพที่ทำให้เสียหาย

จากนั้นทำการทดสอบการต่อเติมภาพทั้ง 5 โดยทดสอบวิธีสปริทเบรกแมน และวิธีที่พัฒนาขึ้นโดยใช้ วิธีการสปริทเบรกแมนพร้อมทั้งการใช้พิระมิดรูปภาพที่มีการทำข้าแต่ละชั้นเป็น 10/3/3/10 ได้ผลลัพธ์ออกมา เป็นดังนี้



รูปที่ 3.11: ผลลัพธ์จากการสปริทเบรกแวน



รูปที่ 3.12: ผลลัพธ์จากการที่พัฒนาขึ้น

โดยได้ผลลัพธ์ออกเป็นดังตารางนี้

วิธีการ	รูปภาพ	เวลาประมาณ (วินาที)	PSNR (dB)	SSIM
สปริทเบรกเมน	1			
	2			
	3			
	4			
	5			
วิธีการที่พัฒนาขึ้น	1			
	2			
	3			
	4			
	5			

ตารางที่ 7: แสดงผลลัพธ์เฉลี่ยของการซ่อมแซมภาพศิลปะไทย

ซึ่งได้ผลลัพธ์เฉลี่ยของมาดังนี้

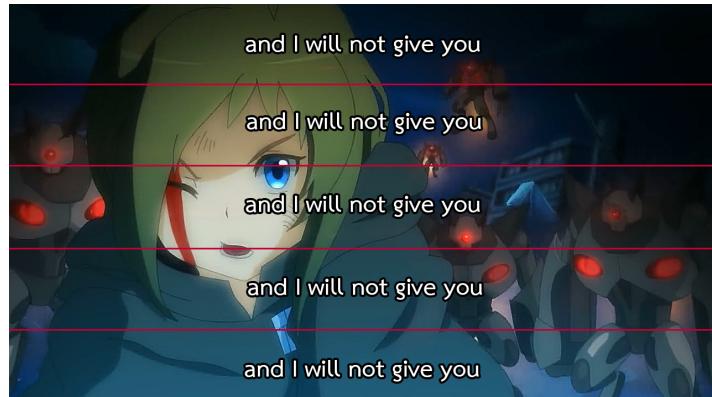
วิธีการ	เวลาประมาณ (วินาที)	PSNR (dB)	SSIM
สปริทเบรกเมน	0.9518894	36.8499	0.9999882
วิธีการที่พัฒนาขึ้น	0.2826256	36.79616	0.999988

ตารางที่ 8: แสดงผลลัพธ์เฉลี่ยของการซ่อมแซมภาพศิลปะไทย

ซึ่งจากตารางจะเห็นได้ว่า วิธีสปริทเบรกเมน นั้นแม้จะมี PSNR ที่ตีกว่าเพียงเดือนน้อย แต่ SSIM กลับแย่กว่า และยังใช้เวลามากกว่าถึง 3 เท่า ส่วนวิธีการ Fast Marching Method แม้ว่าจะมีคุณภาพที่ด้อยกว่า ทั้งค่า PSNR และ SSIM แต่กลับมีเวลาที่รวดเร็วกว่ามาก

3.2 การลับบทบรรยายบนอนิเมะ

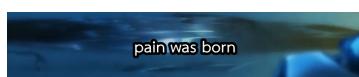
สำหรับการลับบทบรรยายอนิเมะ จะใช้วิดีโอ Anime Festival Asia Special Video - feat. Inori Aizawa ซึ่งผลิตโดย Collateral Damage Studios โดยจะตัดวิดีโอ 1 นาทีแรกสำหรับการทดลอง โดยวิดีโอดังกล่าวขนาด 1280×720 พิกเซล แต่เนื่องจากโดยปกติแล้ว อนิเมะมักมีบรรทัดเพียง 1 ถึง 2 บรรทัด จึงทำการแบ่งวิดีโออกอีกเป็น 5 ส่วนได้ขนาดเป็น 1280×144 พิกเซลก่อนนำไปทดสอบในลำดับถัดไป และสำหรับบทบรรยายที่จะใช้ทดสอบนั้น เป็นจากวิดีโอ Anime Festival Asia Special Video - feat. Inori Aizawa ไม่มีคำพูดใดๆ จึงใช้บล็อกความ lorem ipsum เป็นบทบรรยาย โดยจะทำการแสดงบทบรรยาย 1 บรรทัด ความยาว 3 วินาที ทุก 2 วินาที นั่นคือในวิดีโอดังกล่าวจะมีบทบรรยายทั้งสิ้น 20 บรรทัด



รูปที่ 3.13: การแบ่งไฟล์วิดีโอเป็น 5 ส่วนสำหรับใช้เป็น 5 ชุดทดสอบ

3.2.1 การhabทบทรรยายนอนิเมะ

ก่อนจะระบบทบทรรยายนั้น จะเป็นต้องhabทบทรรยายในภาพให้ได้เสียก่อน โดยบทบทรรยายของอนิเมะนั้น มักจะขึ้นบริเวณด้านล่างของหน้าจอ และนอกจากนี้ บทบทรรยายอนิเมะมักจะใช้ขอบของตัวอักษรเป็นสีดำ อีกด้วย ด้วยสมบัตินี้เองทำให้เราสามารถหาบริเวณบนเฟรมที่เป็นบทบทรรยายได้โดยจะมีวิธีหาพื้นที่ซึ่งเป็นบทบทรรยายดังนี้

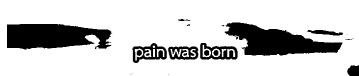


(a) ภาพเฟรมอนิเมะบริเวณที่มีบทบทรรยาย

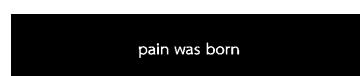


(b) ภาพหลังทำการ thresholding

ตัดเฟรมมาเฉพาะส่วนล่างของเฟรมที่น่าจะมีบทบทรรยายปรากฏอยู่ จากนั้นทำการ thresholding เพื่อหาบริเวณที่เป็นสีดำเนื่องจากบทบทรรยายจะถูกกล้อมรอบด้วยสีดำเสมอ

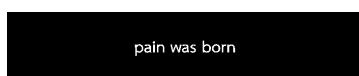


(a) ภาพหลังทำการสลับสี



(b) ภาพหลังทำการเปลี่ยนพื้นที่สีขาว

ทำการสลับสีระหว่างสีดำกับสีขาวของภาพที่ทำการ thresholding หลังจากนั้นทำการเปลี่ยนพื้นที่สีขาวซึ่งติดกับขอบของเฟรมทั้งหมดให้เป็นสีดำ เพราะว่า บทบทรรยายไม่อยู่ติดกับหน้าจอ เราจะถือว่าสิ่งที่อยู่ติดกับหน้าจอไม่ใช่บทบทรรยาย



(a) ภาพหลังการ erode และ opening



(b) ภาพหลังการ dilate

จากนั้นนำวัตถุที่มีขนาดเล็กเกินไป หรือใหญ่เกินไปออกจากภาพด้วยวิธีการ erode และ opening จะได้ว่าส่วนที่เหลือเป็นสีขาวในภาพคือบทบทรรยาย แต่ว่าขอบของบทบทรรยายก็ต้องถูกลบออกไปด้วย จึงทำการ

dilate เพื่อขยายขอบของบทบรรยายให้ท่ากับบทบรรยายที่อยู่ในเฟรมวิดีโอ และสิ่งที่เหลืออยู่คือโดเมนต่อเติมที่จะนำไปใช้ในการซ้อมแซมภาพต่อไป

ซึ่งวิธีการหาบทบรรยายที่กล่าวไปข้างต้น จะทำการทดสอบกับบทความ *lorem ipsum* ที่ถูกแปลเป็นภาษาไทย ภาษาอังกฤษ และภาษาญี่ปุ่น โดยมีความสามารถในการหาโดเมนต่อเติมแบบบทบรรยายภาษาต่างๆ เฉลี่ยดังนี้

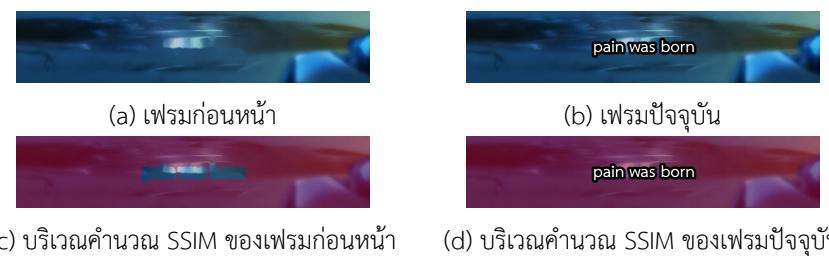
ภาษา	จำนวนพิกเซลในโดเมน	จำนวนพิกเซลที่ตัวจริง	จำนวนพิกเซลที่ผิดพลาด	ร้อยละการผิดพลาด
ไทย	23,222,220	24,083,125	214,1201	9.22
อังกฤษ	27,278,745	28,598,424	371,4321	13.62
ญี่ปุ่น	28,544,173	30,103,466	374,0971	13.11

ตารางที่ 9: แสดงความคาดเคลื่อนเฉลี่ยของการหาโดเมนต่อเติม ในบทบรรยายภาษาต่างๆ

จากการทดลองทั้ง 3 ภาษาพบว่าวิธีการหาคำบรรยายนี้ มีร้อยละการผิดพลาดเฉลี่ยอยู่ที่ 11.98 ซึ่ง การทดลองจากนี้ไปจะใช้วิธีการหาคำบรรยายนี้ในการหาโดเมนต่อเติมแบบอัตโนมัติ

3.2.2 การลบคำบรรยายจากบทอนิเมะ

สำหรับอนิเมะนั้น แต่ละเฟรมจะเป็นรูปภาพ เรายังสามารถประยุกต์ใช้วิธีการซ้อมแซมภาพจิตรกรรมไทย มาใช้ในการลบคำบรรยายได้ แต่ผู้จัดก็ได้สังเกตว่า สำหรับอนิเมะที่เป็นวิดีโอล้วน ในขณะที่ประมวลผลวิดีโอด้วยความสามารถใช้ผลการต่อเติมภาพจากภาพที่แล้ว มาใช้เป็นคำตอบเริ่มต้น โดยจะขอเรียกวิธีทั้งสองวิธีที่คิดขึ้นว่า วิธียึมเฟรม และวิธีข้ามเฟรม สำหรับวิธียึมเฟรม หากเฟรมก่อนหน้า กับเฟรมปัจจุบัน ในส่วนที่อยู่นอกโดเมนต่อเติม มีค่า SSIM มากกว่า 0.9 จะทำการใช้เฟรมก่อนหน้าเป็นค่าเริ่มต้นในการทำซ้ำแทนที่จะใช้ภาพที่มีคำบรรยายในการต่อเติม โดยจะนำส่วนที่อยู่ในโดเมนต่อเติมมาจากภาพก่อนหน้า ก่อนจะเริ่มการทำซ้ำเพื่อต่อเติมภาพ และวิธีข้ามเฟรมนั้น เมื่อบริเวณนอกโดเมนต่อเติมของเฟรมปัจจุบันกับเฟรมก่อนหน้า มีค่า SSIM มากกว่า 0.95 จะทำการคัดลอกบริเวณในโดเมนต่อเติมของภาพก่อนหน้ามาและข้ามไปประมวลผลเฟรมถัดไปทันที เพื่อลดเวลาการทำงาน



รูปที่ 3.17: บริเวณที่คำนวน SSIM สำหรับการยึมเฟรมและข้ามเฟรม

จากภาพ 3.17a เป็นเฟรมก่อนหน้าซึ่งได้ถูกการต่อเติมภาพไปแล้ว และภาพ 3.17b คือเฟรมปัจจุบันที่กำลังจะต่อเติม โดยจะทำการพิจารณาค่า SSIM ในระหว่าง 2 พื้นที่แสดงในภาพ 3.17c และ 3.17d โดย

สำหรับการข้ามเฟรม หากค่า SSIM มากกว่า 0.95 จะทำการคัดลอกภาพที่อยู่นอกพื้นที่สีแดงของภาพ 3.17c มาแทนส่วนที่อยู่นอกสีแดงของภาพ 3.17d แล้วทำการข้ามเฟรมนั้นไป และสำหรับวิธียึดเฟรม หากค่า SSIM มากกว่า 0.95 จะทำการคัดลอกภาพที่อยู่นอกพื้นที่สีแดงของภาพ 3.17c มาแทนที่เดิมต่อเติมของเฟรมปัจจุบันแล้วจึงเริ่มการต่อเติมภาพ

ทั้งนี้ทางผู้วิจัยยังได้ทดสอบทั้ง 2 วิธีรวมกันด้วย ซึ่งจะเรียกว่าวิธีข้ามและยึด โดยเมื่อ เมื่อบริเวณนอก โอดเมนต่อเติมของเฟรมปัจจุบันกับเฟรมก่อนหน้า มีค่า SSIM มากกว่า 0.95 จะให้ทำการข้ามเฟรม แต่ถ้าหากน้อยกว่า 0.95 แต่ยังมากกว่า 0.9 จะทำวิธีการยึดเฟรมแทน ซึ่งผลลัพธ์เฉลี่ยได้ดังตาราง

วิธีการ	เวลาประมาณ (วินาที)	PSNR (dB)	SSIM
ไม่ใช้	141.2850811	31.6037548	0.952397
ยึดเฟรม	132.7787761	32.6481538	0.9658446
ข้ามเฟรม	89.29141836	29.2396524	0.942108
ยึดเฟรมและข้ามเฟรม	75.75894242	29.4711168	0.947267

ตารางที่ 10: แสดงผลลัพธ์เฉลี่ยของการยึดเฟรมและข้ามเฟรม

จากนั้นทำการทดสอบการต่อเติมวิดีโอทั้ง 5 โดยวิธีที่คิดค้นขึ้นใช้วิธีการสปริทเบรกแมนพร้อมทั้งการใช้พิษมิตรูปภาพที่มีการทำข้ามแต่ละชั้นเป็น 10/3/3/10 พร้อมทั้งใช้การข้ามเฟรมและยึดเฟรม ได้ผลลัพธ์ออกเป็นดังตารางนี้

วิธีการ	เวลาประมาณ (วินาที)	PSNR (dB)	SSIM
สปริทเบรกแมน ที่คิดค้นขึ้น	*	*	*

ตารางที่ 11: แสดงผลลัพธ์เฉลี่ยของการลบบทบรรยายจากอนิเมะ

สำหรับวิธีสปริทเบรกแมน เนื่องจากใช้เวลา 1 ชั่วโมงแล้วยังประมาณผลวิดีโอชุดทดสอบแรกไม่เสร็จ ทางผู้พัฒนาจึงตัดสินใจยุติการทดลอง เนื่องจากอาจต้องใช้เวลาการประมาณผลเป็นเวลาหลายชั่วโมงสำหรับวิดีโocommunity 1 นาที ส่วนวิธีที่คิดค้นขึ้น พบว่าเวลาที่ใช้ทำงานข้ากวิธี Fast Marching Method และในด้านคุณภาพ ทั้ง PSNR และ SSIM มีค่าน้อยกว่า แต่ทั้งนี้การปรับค่าพารามิเตอร์เร็กกิวเลาร์ต์เซ็นเซอร์อยู่นอกเหนือขอบเขตการศึกษา ซึ่งหากทำการปรับพารามิเตอร์ตั้งกล่าวแล้ว อาจจะทำให้คุณภาพที่ได้ ดีกว่าวิธีการ Fast Marching Method ได้

4 แผนการดำเนินงานวิจัย

แผนการดำเนินงานตลอดทั้งโครงการสามารถสรุปได้โดยย่อจากตารางต่อไปนี้

แผนการดำเนินงาน	เดือนที่											
	1	2	3	4	5	6	7	8	9	10	11	12
ศึกษาตัวแบบและขั้นตอนวิธีการต่อเติมภาพที่ใช้การแปรผันรวมในเชิงลึก พัฒนาขั้นตอนวิธีสำหรับการต่อเติมภาพที่ใช้การแปรผันรวมชนิดใหม่ ทดสอบขั้นตอนวิธีการต่อเติมภาพที่พัฒนาขึ้นโดยโปรแกรม-คอมพิวเตอร์บนภาพสังเคราะห์และภาพจริง อภิปรายผลที่ได้จากการทดลองเชิงวิเคราะห์ สรุปผลการดำเนินงานวิจัยและจัดทำรูปถ่ายบันทุมูรรณ์	x	x	x	x	x	x				x	x	x

5 บรรณานุกรม

- [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.