

## Report Progress

Division of Applied Mathematics, Department of Mathematics

Faculty of Science, Silpakorn University

Date : 23 พฤศจิกายน 2561

Advisor : ผู้ช่วยศาสตราจารย์ ดร.นพดล ชุ่มชอบ

Student : นายภคพล 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

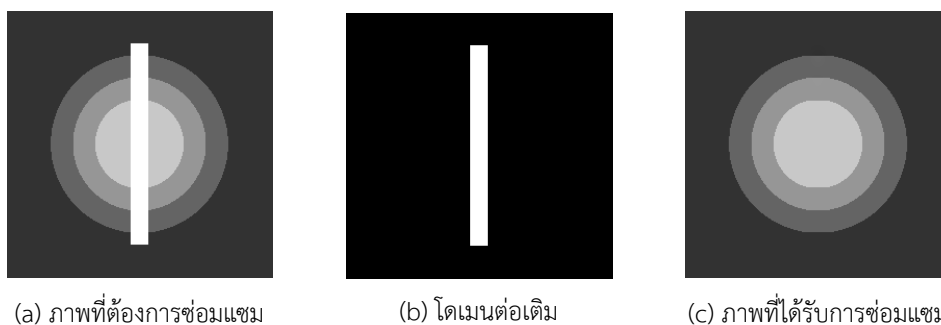
ในปัจจุบันการใช้ภาพดิจิทัล (digital images) ในสังคมเครือข่ายได้รับความนิยมอย่างแพร่หลาย เนื่องจากโทรศัพท์เคลื่อนที่มีราคาถูกลงแต่มีความสามารถที่ชาญฉลาด สามารถทำหน้าที่ได้ตั้งแต่การเป็นกล้องดิจิทัลคอมแพค (compact digital camera) คุณภาพดีให้ภาพดิจิทัลที่มีความคมชัดสูงจนไปถึงการทำหน้าที่ตั้งเช่นเครื่องคอมพิวเตอร์ส่วนบุคคลที่สามารถเชื่อมต่อกับระบบเครือข่ายไร้สายเพื่อรับส่งภาพดิจิทัลในสังคมเครือข่ายด้วยความสะดวกและรวดเร็ว

นอกจากภาพดิจิทัลจะได้รับจากการถ่ายภาพด้วยโทรศัพท์เคลื่อนที่แล้ว ภาพดิจิทัลยังได้รับการถ่ายภาพด้วยกล้องดีเอสแอลอาร์ หรือ กล้องสะท้อนเลนส์เดี่ยวแบบดิจิทัล (digital single lens reflex camera) กล้องโทรทรรศน์ (หรือ กล้องดูดาว) หรือ เครื่องมือสร้างภาพถ่ายทางการแพทย์ (medical imaging device)

โดยทั่วไปภาพดิจิทัลจะได้รับการประมวลผลภาพก่อนนำไปใช้งานเพื่อให้สามารถใช้ข้อมูลที่ปรากฏบนภาพได้ตรงวัตถุประสงค์ของการใช้งานมากที่สุด ตัวอย่างเช่น ภาพบุคคล (portrait) อาจจำเป็นต้องได้รับการกำจัดสัญญาณรบกวนออกจากภาพและ/หรือปรับเพิ่มความละเอียดข้อมูลของความเข้มของสีและความสว่างของสีบนบริเวณใบหน้าก่อนนำภาพไปใช้งานเพื่อจัดทำต้นฉบับวารสารหรือหนังสือของสำนักพิมพ์ เป็นต้น

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

ดังกล่าวจะเริ่มด้วยการกำหนดให้บริเวณแท่งวัตถุรูปร่างสี่เหลี่ยมสีขาวเป็นโดเมนการต่อเติมดังรูปที่ 1.1 (b) จากนั้นภาพที่ได้รับการซ่อมแซมหรือภาพที่ได้รับการต่อเติม (restored or inpainted image) ซึ่งแสดงในรูปที่ 1.1 (c) ได้มาจากขั้นตอนวิธีการต่อเติมภาพ (inpainting algorithm) ซึ่งได้รับการออกแบบเพื่อนำข้อมูลที่ปรากฏบนภาพในบริเวณใกล้เคียงกับขอบของโดเมนต่อเติมมาซ่อมแซมภาพ



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

เท่าที่ผู้วิจัยศึกษาและค้นคว้ามาจนถึงขณะนี้ ผู้วิจัยพบว่าการต่อเติมภาพมักนิยมนำไปใช้งานสำหรับการปรับแต่งความสวยงามของภาพบุคคลที่ถ่ายจากโทรศัพท์เคลื่อนที่ เช่น การลบร่องรอยของรอยตีนกา การลบร่องรอยแผลเป็นที่เกิดจากสิวเสี้ยน การลบร่องรอยของความขรา หรือ การเพิ่มความใสและความเนียนของสีผิวบนบริเวณใบหน้าผ่านโปรแกรมแอปพลิเคชันแต่งรูปภาพที่มีอยู่ในแอปสโตร์ (App Store) หรือ กูเกิลเพลย์ (Google Play) เป็นต้น

## 1.1 การซ่อมแซมภาพจิตรกรรมไทยโบราณ

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

อย่างไรก็ตาม ภาพจิตรกรรมไทยจำนวนไม่น้อยที่เสื่อมสลายตามกาลเวลา และรอคอยการซ่อมแซมจากช่างในสมัยปัจจุบันที่ต้องไม่สร้างความเสียหายให้กับภาพเขียนเพิ่มขึ้นมากกว่าเดิม ที่ผ่านมามีภาพที่ผ่านการซ่อมแซมมาแล้วจำนวนไม่น้อยได้รับความเสียหายหลังจากการซ่อมแซม ถึงแม้ภาพโดยรวมของภาพจิตรกรรมเดิมยังคงอยู่ แต่รายละเอียดในตัวภาพเขียนได้เปลี่ยนไป ก่อให้เกิดความเสียหายที่ประเมินค่าไม่ได้

การซ่อมแซมภาพจิตรกรรมไทยโบราณโดยใช้การต่อเติมภาพเป็นขั้นตอนของการซ่อมแซมแบบหนึ่งซึ่งไม่ก่อให้เกิดความเสียหายใด ๆ กับภาพเดิม เนื่องจากการซ่อมแซมโดยใช้ขั้นตอนวิธีเชิงตัวเลขบนภาพดิจิทัลซึ่งเป็นสำเนาของภาพเดิม ด้วยเหตุผลดังกล่าว ผู้วิจัยได้เล็งเห็นว่าการซ่อมแซมภาพจิตรกรรมไทยโบราณมีความจำเป็นเร่งด่วน เนื่องจากภาพที่ได้รับการซ่อมแซมด้วยการต่อเติมภาพสามารถนำไปใช้ประกอบการตัดสินใจเพื่อวางแผนก่อนการลงมือซ่อมแซมภาพเขียนจริงได้ นอกจากนี้ ขั้นตอนวิธีการต่อเติมภาพสามารถนำไปใช้สร้างแอปพลิเคชันบนโทรศัพท์เคลื่อนที่เพื่อนำไปใช้เป็นข้อมูลในการเข้าชมภาพเขียนเดิมที่ยัง

ไม่ได้รับการซ่อมแซมและภาพเขียนที่ได้รับการซ่อมแซมโดยวิธีการทางคณิตศาสตร์จากแอปพลิเคชันที่พัฒนาขึ้น

รูปที่ 1.2 แสดงตัวอย่างภาพจิตรกรรมไทย<sup>1</sup> ที่ต้องได้รับการซ่อมแซมบนบริเวณแขนเสื้อของรูปวาดผู้ชายที่มีส่วนของสีแดงเดิมหลุดหายไป ทั้งนี้ในการซ่อมแซมภาพโดยการต่อเติมภาพ เราจะเริ่มด้วยการสร้างโดเมนต่อเติมบนบริเวณสีพื้นผิวปูนที่แขนเสื้อ จากนั้นจึงนำขั้นตอนวิธีการต่อเติมภาพเพื่อซ่อมแซมภาพบริเวณนั้นให้เป็นสีแดง



รูปที่ 1.2: ภาพจิตรกรรมไทยที่วัดภูมินทร์ อำเภอเมือง จังหวัดน่าน

## 1.2 การลบบทบรรยายบนอนิเมะ

อนิเมะคือวิดีโอภาพวาดการ์ตูนสไตล์ญี่ปุ่นซึ่งเป็นที่นิยมของเยาวชนไทย ในการรับชมอนิเมะ แม้ว่าเยาวชนไทยสามารถรับชมด้วยบทพากย์เสียงภาษาไทย แต่ก็สูญเสียอรรถรสของการรับชมจากบทบรรยายแบบแข็ง<sup>2</sup> (hardsub) ที่เป็นภาษาต่างประเทศในบริเวณด้านล่างของจอภาพ ในการซ่อมแซมอนิเมะด้วยการลบบทบรรยายภาษาต่างประเทศจึงเป็นงานที่ยุ่งยากและท้าทายมาก เนื่องจาก

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

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

<sup>2</sup> บทบรรยายที่ไม่สามารถปิดหรือเปิดได้

ด้วยความท้าทายข้างต้น การพัฒนาขั้นตอนวิธีการต่อเติมภาพที่สามารถกำหนดโดเมนต่อเติมเชิงอัตโนมัติให้กับแต่ละเฟรมและประมวลผลได้แม่นยำจนการลบทบรอยสามารถทำงานได้แบบเรียลไทม์จึงเป็นสิ่งจำเป็นที่หลีกเลี่ยงไม่ได้

รูปที่ 1.3 แสดงตัวอย่าง 1 เฟรมของอนิเมะที่มีบทบรอยแบบแข็ง<sup>3</sup> ที่ต้องซ่อมแซมด้วยการลบทบรอยออก ทั้งนี้ในการลบทบรอยออกจากเฟรมโดยใช้การต่อเติมภาพ เราจะเริ่มด้วยการสร้างโดเมนต่อเติมแบบอัตโนมัติในบริเวณบทบรอย จากนั้นจึงนำขั้นตอนวิธีการต่อเติมภาพแบบเร็วเพื่อลบทบรอยออกจากเฟรม



รูปที่ 1.3: 1 เฟรมของอนิเมะที่มีบทบรอยแบบแข็ง

โครงการวิจัยนี้ ผู้วิจัยมีเป้าหมายสำคัญคือการพัฒนาขั้นตอนวิธีการต่อเติมภาพแบบเร็วและแม่นยำชนิดใหม่เพื่อนำไปใช้สำหรับซ่อมแซมภาพจิตรกรรมไทยและการลบทบรอยออกจากอนิเมะ

## 2 วรรณกรรมและทฤษฎีบทที่เกี่ยวข้อง

ในการกล่าวถึงขั้นตอนวิธีการต่อเติมภาพ จะเริ่มต้นด้วยการกล่าวทบทวนเกี่ยวกับการต่อเติมภาพเฉดสีเทา (grayscale image) ก่อน ดังนี้

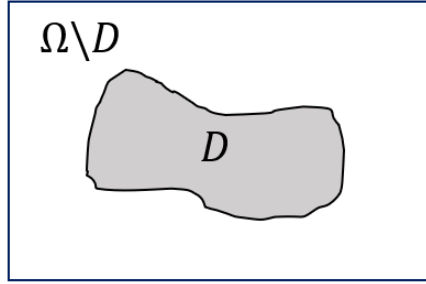
ให้  $\Omega \subset \mathbb{R}^2$  แทนโดเมนภาพ (image domain)  $D \subset \mathbb{R}^2$  แทนโดเมนต่อเติม (ดูรูปที่ 2.1) และ  $V \subset [0, \infty)$

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

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

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

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



รูปที่ 2.1:  $D$  แทนโดเมนต่อเติม

## 2.1 ตัวแบบการต่อเติมภาพเจดสีเทาที่ใช้การแปรผันรวม

ในการต่อเติมภาพเจดสีเทา Chan และ Shen [1] ได้นำเสนอตัวแบบเชิงการแปรผัน (variational model) ที่ใช้เร็กกิวลารีไรซ์เซชันแบบการแปรผันรวม (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

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

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

โดยแคลคูลัสของการแปรผัน (Calculus of variations) จะได้สมการออยเลอร์-ลากรางจ์ที่เกี่ยวข้องกับ (2.1) เป็น

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

เมื่อ  $\mathbf{n}$  แทนเวกเตอร์หน่วยที่ตั้งฉากกับขอบของภาพ

ต่อไปจะกล่าวทบทวนวิธีการเชิงตัวเลขสำหรับแก้สมการเชิงอนุพันธ์ย่อยใน (2.3)

### (1) วิธีการเดินเวลาแบบชัดแจ้ง (explicit time marching method)

คณะวิจัย [2] ได้แนะนำวิธีการเชิงตัวเลขสำหรับการกำจัดสัญญาณรบกวนโดยใช้วิธีการเดินเวลาแบบชัดแจ้ง ซึ่งสามารถประยุกต์เป็นวิธีเชิงตัวเลขสำหรับการต่อเติมภาพได้ดังนี้

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

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

(2) วิธีการทำซ้ำแบบจุดตรึง (fixed-point iteration method)

คณะวิจัย [3] ได้แนะนำวิธีการเชิงตัวเลขสำหรับการกำจัดสัญญาณรบกวนโดยใช้วิธีการทำซ้ำแบบจุดตรึง ซึ่งสามารถประยุกต์เป็นวิธีเชิงตัวเลขสำหรับการต่อเติมภาพได้ดังนี้

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

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

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

เพื่อหลีกเลี่ยงปัญหาเชิงตัวเลขจะเกิดขึ้นใน (2.4) และ (2.5) เราจะใช้

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

จาก (2.4) และ (2.5) เราพบว่ายิ่ง  $\beta$  มีค่าน้อยลงมากขึ้นเท่าไร ความแม่นยำของตัวแบบ (2.1) ยิ่งมีมากขึ้นเท่านั้น นอกจากนี้ เรายังพบอีกว่า การแก้สมการ (2.4) และ (2.5) ยิ่งมีความยุ่งยากมากขึ้นสำหรับ  $\beta$  ที่มีค่าน้อยๆ

เพื่อเอาชนะความยากเชิงตัวเลขนี้ คณะวิจัยโดย [4] ได้แนะนำวิธีการสปริทเบรกแมนซึ่งสามารถกล่าวถึงพอสังเขป ดังนี้

(3) วิธีการสปริทเบรกแมน (Split Bregman method)

เริ่มจากการแนะนำเวกเตอร์เสริม  $\mathbf{w}$  พารามิเตอร์เบรกแมน (Bregman parameter)  $\mathbf{b}$  และพารามิเตอร์เพนัลตี (penalty parameter)  $\theta > 0$  และเขียน (2.1) ใหม่ ดังนี้

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

สำหรับการหาคำตอบของ (2.6) เราจะใช้วิธีการหาค่าต่ำสุดแบบสลับ (alternating minimization method) โดยเริ่มจากการตรึง  $\mathbf{w}^{\text{old}}$  และ  $\mathbf{b}^{\text{old}}$  จากนั้นแก้ปัญหาย่อย

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

จากนั้นใช้  $u^{\text{New}}$  ที่ได้จากการแก้ปัญหาย่อยใน (2.7) เพื่อแก้ปัญหาย่อย

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

สุดท้ายจึงปรับปรุงพารามิเตอร์เบรกแมน

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

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

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

ในทำนองเดียวกันกับตัวแบบการต่อเติมภาพเฉดสีเทาที่ใช้การแปรผันรวม 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

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

เพื่อหลีกเลี่ยงปัญหาที่มาจากเทอม  $\frac{1}{|\nabla u_l|}$  ( $l = 1, 2, 3$ ) โครงการวิจัยนี้จะพัฒนาขั้นตอนวิธีเชิงตัวเลขสำหรับต่อเติมภาพจากวิธีการสปริทเบรกแมน โดยแก้ปัญหาค่าต่ำที่สุดต่อไปนี้

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

ด้วยวิธีการหาค่าต่ำสุดแบบสลับดังเช่น (2.6) - (2.9)

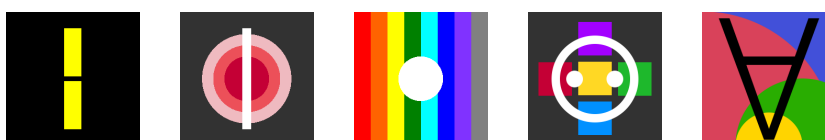
### 3 ผลการดำเนินงานเบื้องต้น

#### 3.1 การซ่อมแซมภาพจิตรกรรมไทยโบราณ

สำหรับการซ่อมแซมจิตรกรรมไทยโบราณ ก่อนอื่นจะทำการปรับปรุงขั้นตอนวิธีเชิงตัวเลขที่มีอยู่แต่เดิมก่อน โดยระหว่างการปรับปรุงวิธีเชิงตัวเลข จะใช้ภาพที่ได้สร้างขึ้น 5 ภาพ แต่ละภาพมีขนาด 256 x 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

วิธีการ	เวลาประมวล (วินาที)	PSNR (dB)	SSIM
การเดินเวลา	146.2108	16.19756	0.9954786
การทำซ้ำจุดตรง	72.32298	40.12234	0.999905
การสปริทเบร็กแมน	11.438246	39.39536	0.999894

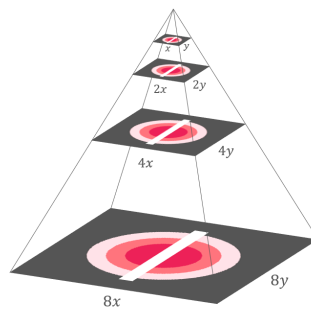
ตารางที่ 1: แสดงผลลัพธ์เฉลี่ยของวิธีการเชิงตัวเลขสำหรับการต่อเติมภาพ

จะเห็นว่าจากตาราง พบว่าแม้วิธีการทำซ้ำจุดตรง และวิธีการสปริทเบร็กแมน จะมีคุณภาพใกล้เคียงกัน แต่ว่าวิธีการสปริทเบร็กแมน ใช้เวลาในการประมวลผลน้อยกว่ามาก ผู้วิจัย จึงนำวิธีการสปริทเบร็กแมนไปใช้พัฒนาวิธีการต่อเติมภาพชนิดใหม่ลำดับถัดไป



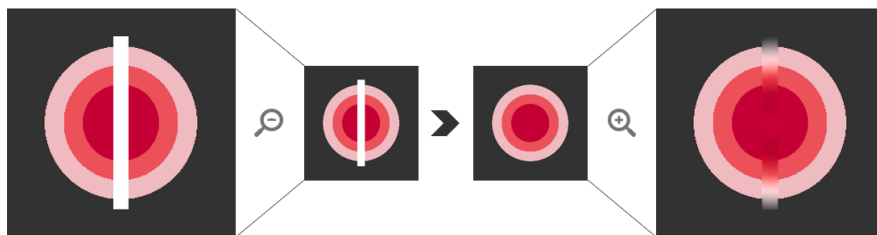
### 3.1.2 ขั้นตอนวิธีการสำหรับต่อเติมภาพชนิดใหม่

จากวิธีการสปริทเบรกแมนนั้นจะใช้วิธีการหาคำตอบโดยวิธีการทำซ้ำจนกระทั่งลู่เข้า ทางผู้ศึกษาจึงสนใจที่หาคำตอบเริ่มต้นสำหรับการทำซ้ำที่ดีขึ้น เพื่อให้การทำซ้ำลู่เข้าสู่คำตอบได้เร็วขึ้น โดยการทำงานกับรูปภาพที่เล็กกว่า จากนั้นจึงทำการขยายผลลัพธ์ที่ได้ขึ้นมาทำกับภาพใหญ่ ซึ่งวิธีนี้เรียกว่าวิธีพีระมิดรูปภาพ (pyramid methods) [5] โดยผู้วิจัยจะทำการย่อขนาดรูปลงครึ่งหนึ่งโดยใช้วิธี Bilinear Interpolation ทั้งสิ้น 4 ครั้ง จากนั้นเริ่มทำการต่อเติมภาพขนาดเล็ก จากนั้นนำผลลัพธ์ที่ได้จากภาพขนาดเล็ก ทำการขยายภาพขึ้นสองเท่าโดยใช้ Bilinear Interpolation ก่อนจะนำเฉพาะส่วนที่อยู่ในโดเมนต่อเติมของภาพที่ถูกขยายมาทำการต่อเติมเพื่อให้ส่วนที่ถูกขยายขึ้นมาเป็นคำตอบเริ่มต้นสำหรับการต่อเติมภาพในขั้นที่สูงขึ้น



รูปที่ 3.3: วิธีการพีระมิดรูปภาพ

อย่างเช่นภาพที่ 3.4 จากภาพที่ต้องการซ่อมแซมทางด้านซ้าย จะทำการย่อภาพให้เหลือขนาดเล็กเพื่อจะได้ทำการต่อเติมภาพได้อย่างรวดเร็ว หลังจากนั้นเมื่อทำการต่อเติมภาพขนาดเล็กเสร็จแล้ว จะทำการขยายภาพขึ้นแล้วทำการนำพื้นที่จากภาพที่ขยายขึ้นไปแทนในโดเมนต่อเติมแล้ว จะได้ภาพนั้น ในส่วนที่เป็นโดเมนต่อเติมจะมีลักษณะเบลอ แต่มีสีใกล้เคียงกับคำตอบที่ต้องการ ซึ่งหลังจากนี้จะนำภาพทางด้านขวาไปทำการต่อเติมจนได้คำตอบที่ต้องการ โดยเมื่อใช้สีที่เบลอแต่มีสีใกล้เคียงเป็นคำตอบเริ่มต้นที่ใช้เวลาการประมวลผลน้อยกว่าการนำภาพไปต่อเติมโดยไม่ผ่านการใช้พีระมิด



รูปที่ 3.4: การเตรียมคำตอบเริ่มต้นจากภาพขนาดเล็ก

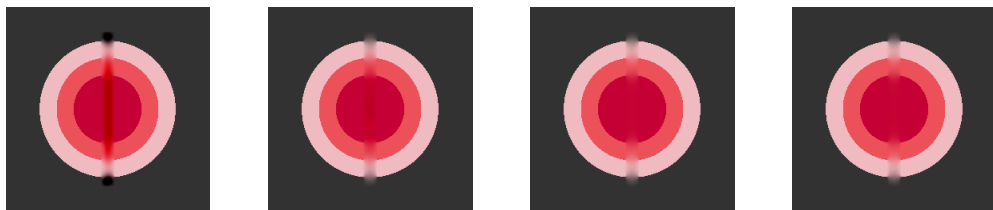
โดยจะทำการเปรียบเทียบจำนวนครั้งในขั้นที่รูปภาพมีขนาดเล็ก จนไปถึงขั้นที่มีขนาดใหญ่ ตัวอย่างเช่น 10/3/3/10000 หมายถึงในขั้นเล็กสุดซึ่งขนาดเป็น 32x32 พิกเซลจะทำซ้ำ 10 ครั้งหรือจนค่าความคลาดเคลื่อนสัมพัทธ์ต่างกันไม่เกิน 0.0001 ขั้นถัดมาขนาดเป็น 64x64 จะทำซ้ำ 3 ครั้งหรือจนค่าความคลาดเคลื่อน

สัมพัทธ์ต่างกันไม่เกิน 0.0001 และชั้นถัดมา ชั้นถัดมาขนาดเป็น 128x128 จะทำซ้ำ 3 ครั้งหรือจนค่าความคลาดเคลื่อนสัมพัทธ์ต่างกันไม่เกิน 0.0001 และสุดท้ายขนาด 256x256 จะทำซ้ำ 3 ครั้งหรือจนค่าความคลาดเคลื่อนสัมพัทธ์ต่างกันไม่เกิน 0.0001 ซึ่งเมื่อทำการทดสอบแล้วได้ผลลัพธ์ดังตารางนี้

รูปแบบการทำซ้ำ	เวลาประมวล (วินาที)	PSNR (dB)	SSIM
ไม่ใช้พีระมิดรูปภาพ	13.690378	30.72924	0.9997676
10/1/1/10000	8.719006	30.5632	0.9997186
10/3/3/10000	5.8064316	30.62492	0.9997298
10/10/10/10000	2.6100824	30.64522	0.9997388
100/1/1/10000	5.839534	30.68802	0.9997514
100/3/3/10000	4.4923898	30.68914	0.9997498
100/10/10/10000	2.4151272	30.67366	0.9997476

ตารางที่ 2: แสดงผลลัพธ์เฉลี่ยของการใช้พีระมิดภาพในการต่อเติมในชั้นที่ต่างกัน

จากตารางจะสังเกตว่า ยิ่งจำนวนการทำซ้ำในชั้นที่รูปภาพมีขนาดเล็กจำนวนมาก ยิ่งประมวลผลได้เร็วขึ้น นอกจากนี้แล้ว ผู้วิจัยยังได้สังเกตอีกว่า การทำซ้ำนั้น จะลู่เข้าเร็วในช่วงแรก จากนั้นความเร็วในการลู่เข้าจะลดลง ซึ่งทำให้การทำซ้ำเพียงไม่กี่ครั้งในรูปภาพขนาดใหญ่สุด มีผลลัพธ์ใกล้เคียงกับภาพต้นฉบับได้



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

รูปที่ 3.5: พีระมิดที่ลำดับการทำซ้ำเป็น 10/10/10 และชั้นใหญ่สุดทำในจำนวนครั้งที่ต่างกัน

โดยผู้วิจัยจึงกำหนดให้การทำซ้ำในรูปภาพขนาดใหญ่สุดมีการทำซ้ำเพียง 10 ครั้ง และพบว่าได้ผลลัพธ์ดังนี้

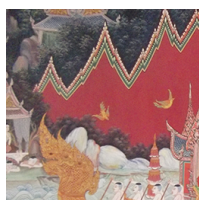
รูปแบบการทำซ้ำ	เวลาประมวล (วินาที)	PSNR (dB)	SSIM
ไม่ใช้พีระมิดรูปภาพ	0.394235	25.22906	0.9989972
10/1/1/10	0.3839476	31.31306	0.9997732
10/3/3/10	0.3672704	30.95474	0.9997682
10/10/10/10	0.4077878	30.78114	0.9997406
100/1/1/10	0.3789076	31.25708	0.9997518
100/3/3/10	0.3939378	30.8896	0.999746
100/10/10/10	0.4507862	30.81302	0.9997504

ตารางที่ 3: แสดงผลลัพธ์เฉลี่ยของการใช้พีระมิดภาพในการต่อเติมโดยกำหนดให้ชั้นรูปภาพขนาดใหญ่ที่สุดทำซ้ำเพียง 10 ครั้ง

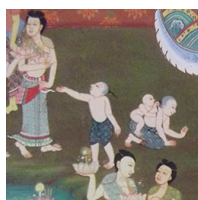
จากตารางจะเห็นว่า การทำซ้ำในชั้นที่รูปภาพมีขนาดเล็กมากจำนวนมาก ไม่ช่วยให้การประมวลผลได้เร็วขึ้น ผู้วิจัยจึงเลือกใช้การทำซ้ำแบบ 10/3/3/10 ในการต่อเติมภาพ

### 3.1.3 การทดสอบประสิทธิภาพในการซ่อมแซมภาพจิตรกรรมไทยโบราณ

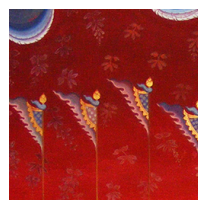
ซึ่งภาพจิตรกรรมที่ใช้ทดสอบ มีทั้งสิ้น 5 ภาพได้แก่ ภาพที่ 1 และภาพที่ 2 คือ จิตรกรรมฝาผนังวัดแก้วไพฑูรย์ ภาพที่ 3 คือ จิตรกรรมฝาผนังวัดพระยืนพุทธบาทยุคล ภาพที่ 4 คือ จิตรกรรมฝาผนังวัดคงคาราม และภาพที่ 5 คือ จิตรกรรมฝาผนังวัดท่าถนน โดยจะทำให้ข้อมูลของทั้ง 5 ภาพเกิดความเสียหาย โดยใช้รอยความเสียหายจากภาพพระเจ้าสร้างอ้อม



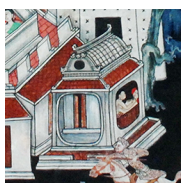
(a) วัดแก้วไพฑูรย์



(b) วัดแก้วไพฑูรย์



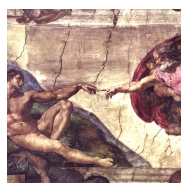
(c) วัดพระยืนพุทธบาทยุคล



(d) วัดคงคาราม



(e) วัดท่าถนน



(f) พระเจ้าสร้างอ้อม



(g) รอยความเสียหาย

จากนั้นทำการทดสอบการต่อเติมภาพทั้ง 5 โดยทดสอบวิธีสปริทเบรกแมน วิธีที่พัฒนาขึ้นโดยใช้วิธีการสปริทเบรกแมนพร้อมทั้งการใช้พีระมิดรูปภาพที่มีการทำซ้ำแต่ละชั้นเป็น 10/3/3/10 และวิธี Fast Marching Method ซึ่งถูกเขียนไว้ในชุดคำสั่ง OpenCV ได้ผลลัพธ์ออกเป็นดังตารางนี้

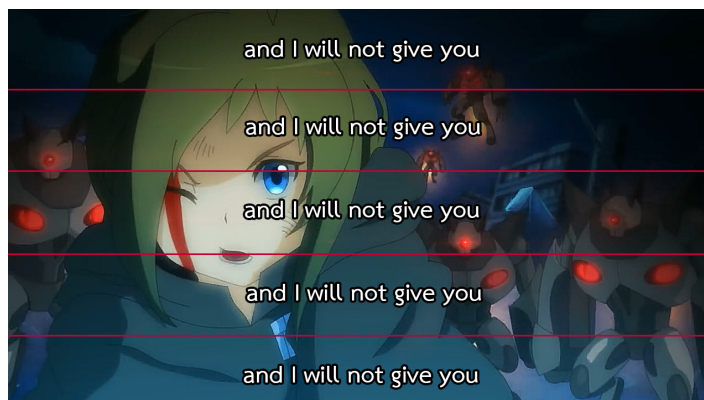
วิธีการ	เวลาประมวล (วินาที)	PSNR (dB)	SSIM
สปิริทเบรกแมน	0.9518894	36.8499	0.9999882
วิธีการที่พัฒนาขึ้น	0.2826256	36.79616	0.999988
OpenCV - Fast Marching Method	0.0198226	35.56518	0.9968536

ตารางที่ 4: แสดงผลลัพธ์เฉลี่ยของการซ่อมแซมภาพศิลปะไทย

ซึ่งจากตารางจะเห็นได้ว่า วิธีสปิริทเบรกแมน นั้นแม้จะมี PSNR ที่ดีกว่าเพียงเล็กน้อย แต่ SSIM กลับแย่กว่า และยังใช้เวลามากกว่าถึง 3 เท่า ส่วนวิธีการ Fast Marching Method แม้ว่าจะมีคุณภาพที่ด้อยกว่าทั้งค่า PSNR และ SSIM แต่กลับมีเวลาที่รวดเร็วกว่ามาก

### 3.2 การลบบทบรรยายบนอนิเมะ

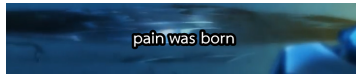
สำหรับการลบบทบรรยายอนิเมะ จะใช้วิดีโอ Anime Festival Asia Special Video - feat. Inori Aizawa ซึ่งผลิตโดย Collateral Damage Studios โดยจะตัดวิดีโอ 1 นาทีแรกสำหรับการทดลอง โดยวิดีโอดังกล่าวขนาด 1280 x 720 พิกเซล แต่เนื่องจากโดยปกติแล้ว อนิเมะมักมีบรรทัดเพียง 1 ถึง 2 บรรทัด จึงทำการแบ่งวิดีโอออกเป็น 5 ส่วนได้ขนาดเป็น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



รูปที่ 3.7: การแบ่งไฟล์วิดีโอเป็น 5 ส่วนสำหรับใช้เป็น 5 ชุดทดสอบ

#### 3.2.1 การหาบทบรรยายบนอนิเมะ

ก่อนจะลบบทบรรยายนั้น จำเป็นต้องหาบทบรรยายในภาพให้ได้เสียก่อน โดยบทบรรยายของอนิเมะนั้น มักจะขึ้นบริเวณด้านล่างของหน้าจอ และนอกจากนี้ บทบรรยายอนิเมะมักจะใช้ขอบของตัวอักษรเป็นสีดำอีกด้วย ด้วยสมบัตินี้เองทำให้เราสามารถหาบริเวณบนเฟรมที่เป็นบทบรรยายได้โดยจะมีวิธีหาพื้นที่ซึ่งเป็นบทบรรยายดังนี้



(a) ภาพเฟรมอนิเมะบริเวณที่มีบทบรรยาย

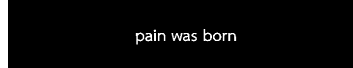


(b) ภาพหลังทำการ thresholding

ตัดเฟรมมาเฉพาะส่วนกลางของเฟรมที่น่าจะมีบทบรรยายปรากฏอยู่ จากนั้นทำการ thresholding เพื่อหาบริเวณที่เป็นสีดำเนื่องจากบทบรรยายจะถูกล้อมรอบด้วยสีดำเสมอ

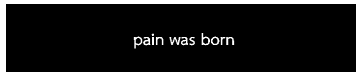


(a) ภาพหลังทำการสลับสี



(b) ภาพหลังทำการเปลี่ยนพื้นที่สีขาว

ทำการสลับสีระหว่างสีดำกับสีขาวของภาพที่ทำการ thresholding หลังจากนั้นทำการเปลี่ยนพื้นที่สีขาวซึ่งติดกับขอบของเฟรมทั้งหมดให้เป็นสีดำ เพราะว่า บทบรรยายไม่อยู่ติดกับหน้าจอ เราจะถือว่าสิ่งที่อยู่ติดกับหน้าจอไม่ใช่บทบรรยาย



(a) ภาพหลังการ erode และ opening



(b) ภาพหลังการ dilate

จากนั้นนำวัตถุที่มีขนาดเล็กเกินไป หรือใหญ่เกินไปออกจากภาพด้วยวิธีการ erode และ opening จะได้ว่าส่วนที่เหลือเป็นสีขาวในภาพคือบทบรรยาย แต่ว่าขอบของบทบรรยายก็ต้องถูกลบออกไปด้วย จึงทำการ dilate เพื่อขยายขอบของบทบรรยายให้เท่ากับบทบรรยายที่อยู่ในเฟรมวิดีโอ และสิ่งที่เหลืออยู่คือโดเมนต่อเติมที่จะนำไปใช้ในการซ่อมแซมภาพต่อไป

ซึ่งวิธีการหาบทบรรยายที่กล่าวไปข้างต้น จะทำการทดสอบกับบทความ lorem ipsum ที่ถูกแปลเป็นภาษาไทย ภาษาอังกฤษ และภาษาญี่ปุ่น โดยมีความสามารถในการหาโดเมนต่อเติมในบทบรรยายภาษาต่างๆ ดังนี้

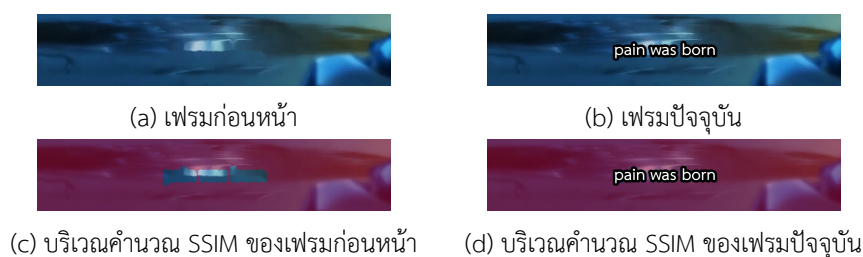
ภาษา	จำนวนพิกเซลในโดเมน	จำนวนพิกเซลที่ตรวจพบ	จำนวนพิกเซลที่ผิดพลาด	ร้อยละการผิดพลาด
ไทย	23222219.8	24083124.8	2141201	9.220398882
อังกฤษ	27278744.8	28598424.2	3714320.8	13.61613769
ญี่ปุ่น	28544173	30103466.2	3740970.6	13.10584872

ตารางที่ 5: แสดงความคาดเคลื่อนเฉลี่ยของการหาโดเมนต่อเติม ในบทบรรยายภาษาต่างๆ

จากการทดลองทั้ง 3 ภาษาพบว่าวิธีการหาคำบรรยายนี้ มีร้อยละการผิดพลาดเฉลี่ยอยู่ที่ 11.98 ซึ่งการทดลองจากนี้ไปจะใช้วิธีการหาคำบรรยายนี้ในการหาโดเมนต่อเติมแบบอัตโนมัติ

### 3.2.2 การลดค่าบรรยายจากบโทนิเมะ

สำหรับอนิเมะนั้น แต่ละเฟรมจะเป็นรูปภาพ เราจึงสามารถประยุกต์ใช้วิธีการซ่อมแซมภาพจิตรกรรมไทย มาใช้ในการลดค่าบรรยายได้ แต่ผู้วิจัยก็ได้สังเกตว่า สำหรับอนิเมะที่เป็นวิดีโอแล้ว ในขณะที่ประมวลผลวิดีโอ เราสามารถใช้ผลการต่อเติมภาพจากภาพที่แล้ว มาใช้เป็นค่าตอบเริ่มต้น โดยจะขอเรียกวิธีทั้งสองวิธีที่คิดขึ้นว่า วิธียืมเฟรม และวิธีข้ามเฟรม สำหรับวิธียืมเฟรม หากเฟรมก่อนหน้า กับเฟรมปัจจุบัน ในส่วนที่อยู่นอกโดเมนต่อเติม มีค่า SSIM มากกว่า 0.9 จะทำการใช้เฟรมก่อนหน้าเป็นค่าเริ่มต้นในการทำซ้ำแทนที่จะใช้ภาพที่มีค่าบรรยายในการต่อเติม โดยจะนำส่วนที่อยู่ในโดเมนต่อเติมมาจากภาพก่อนหน้า ก่อนจะเริ่มการทำซ้ำเพื่อต่อเติมภาพ และวิธีข้ามเฟรมนั้น เมื่อบริเวณนอกโดเมนต่อเติมของเฟรมปัจจุบันกับเฟรมก่อนหน้า มีค่า SSIM มากกว่า 0.95 จะทำการคัดลอกบริเวณในโดเมนต่อเติมของภาพก่อนหน้ามาและข้ามไปประมวลผลเฟรมถัดไปทันที เพื่อลดเวลาการทำงาน



รูปที่ 3.11: บริเวณที่ค่านวน SSIM สำหรับการยืมเฟรมและข้ามเฟรม

จากภาพ 3.11a เป็นเฟรมก่อนหน้าซึ่งได้ถูกการต่อเติมภาพไปแล้ว และภาพ 3.11b คือเฟรมปัจจุบันที่กำลังจะต่อเติม โดยจะทำการพิจารณาค่า SSIM ในระหว่าง 2 พื้นที่แสดงในภาพ 3.11c และ 3.11d โดยสำหรับการข้ามเฟรม หากค่า SSIM มากกว่า 0.95 จะทำการคัดลอกภาพที่อยู่นอกพื้นที่สีแดงของภาพ 3.11c มาแทนส่วนที่อยู่นอกสีแดงของภาพ 3.11d แล้วทำการข้ามเฟรมนั้นไป และสำหรับวิธียืมเฟรม หากค่า SSIM มากกว่า 0.95 จะทำการคัดลอกภาพที่อยู่นอกพื้นที่สีแดงของภาพ 3.11c มาแทนที่โดเมนต่อเติมของเฟรมปัจจุบันแล้วจึงเริ่มการต่อเติมภาพ

ทั้งนี้ทางผู้วิจัยยังได้ทดสอบทั้ง 2 วิธีรวมกันด้วย ซึ่งจะเรียกว่าวิธีข้ามและยืม โดยเมื่อ เมื่อบริเวณนอกโดเมนต่อเติมของเฟรมปัจจุบันกับเฟรมก่อนหน้า มีค่า SSIM มากกว่า 0.95 จะให้ทำการข้ามเฟรม แต่ถ้าน้อยกว่า 0.95 แต่ยังคงมากกว่า 0.9 จะทำวิธีการยืมเฟรมแทน ซึ่งผลลัพธ์เฉลี่ยได้ดังตาราง

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

ตารางที่ 6: แสดงผลลัพธ์เฉลี่ยของการยืมเฟรมและข้ามเฟรม

จากนั้นทำการทดสอบการต่อเติมวิดีโอทั้ง 5 โดยวิธีที่คิดค้นขึ้นใช้วิธีการสปริตเบรกแมนพร้อมทั้งการใช้พีระมิดรูปภาพที่มีการทำซ้ำแต่ละชั้นเป็น 10/3/3/10 พร้อมทั้งใช้การข้ามเฟรมและยืมเฟรม ได้ผลลัพธ์ออกเป็นดังตารางนี้

วิธีการ	เวลาประมวล (วินาที)	PSNR (dB)	SSIM
สปริตเบรกแมน	*	*	*
ที่คิดค้นขึ้น	75.75894242	29.4711168	0.947267
OpenCV - Fast Marching Method	48.64641114	33.0566406	0.9621476

ตารางที่ 7: แสดงผลลัพธ์เฉลี่ยของการซ่อมแซมภาพศิลปะไทย

สำหรับวิธีสปริตเบรกแมน เนื่องจากใช้เวลา 1 ชั่วโมงแล้วยังประมวลผลวิดีโอชุดทดสอบแรกไม่เสร็จทางผู้พัฒนาจึงตัดสินใจยุติการทดลอง เนื่องจากอาจต้องใช้เวลาการประมวลผลเป็นเวลาหลายชั่วโมงสำหรับวิดีโอความยาว 1 นาที ส่วนวิธีที่คิดค้นขึ้น พบว่าเวลาที่ใช้ทำงานช้ากว่าวิธี Fast Marching Method และในด้านคุณภาพ ทั้ง PSNR และ SSIM มีค่าน้อยกว่า แต่ทั้งนี้การปรับค่าพารามิเตอร์เร็กกิวลาไรต์เซชันอยู่นอกเหนือขอบเขตการศึกษา ซึ่งหาทำการปรับพารามิเตอร์ดังกล่าวแล้ว อาจทำให้คุณภาพที่ได้ ดีกว่าวิธีการ Fast Marching Method ได้

## 4 แผนการดำเนินงานวิจัย

แผนการดำเนินงานตลอดทั้งโครงการสามารถสรุปได้โดยย่อจากตารางต่อไปนี้

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