

## 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

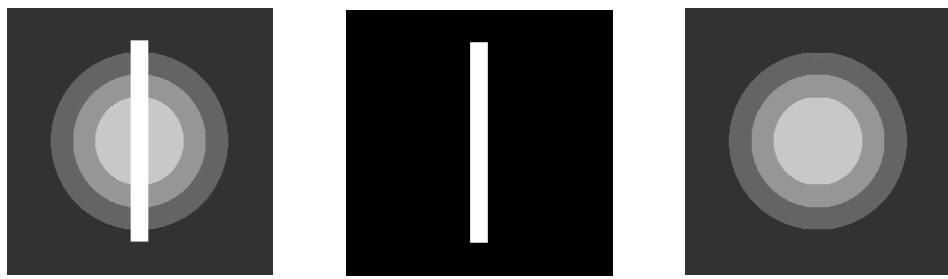
ในปัจจุบันการใช้ภาพดิจิตัล (digital images) ในสังคมเครือข่ายได้รับความนิยมอย่างแพร่หลายเนื่องจากโทรศัพท์เคลื่อนที่มีราคาถูกลงแต่มีความสามารถที่ชาญฉลาด สามารถทำหน้าที่ได้ตั้งแต่การเป็นกล้องดิจิตัลคอมแพค (compact digital camera) คุณภาพดีให้ภาพดิจิตัลที่มีความคมชัดสูงจนไปถึงการทำหน้าที่ดังเช่นเครื่องคอมพิวเตอร์ส่วนบุคคลที่สามารถเชื่อมต่อกับระบบเครือข่ายไร้สายเพื่อรั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

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



(a) ภาพที่ต้องการซ่อมแซม

(b) โmodeนต่อเติม

(c) ภาพที่ได้รับการซ่อมแซม

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

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

## 1.1 การซ่อมแซมภาพจิตรกรรมไทยโบราณ

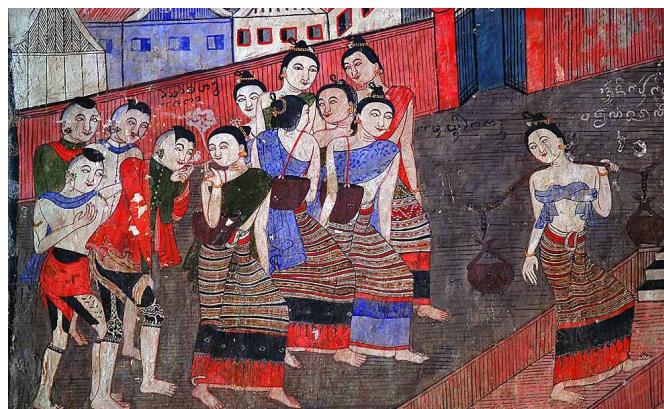
ภาพจิตรกรรมไทย คือ ภาพเขียนที่มีเอกลักษณ์ความเป็นศิลปะไทยซึ่งได้เด่นและแตกต่างจากภาพเขียนของชนชาติอื่น ในอดีต ช่างไทยได้สร้างสรรค์ลวดลายและสีสันบนภาพวาดเพื่อสะท้อนประเพณีและวัฒนธรรมในสังคมไทยที่เกี่ยวกับศาสนา ประวัติศาสตร์ โบราณคดี ชีวิตความเป็นอยู่ วัฒนธรรมการแต่งกายตลอดจนการแสดงการเล่นพื้นเมืองต่าง ๆ ของแต่ละยุคสมัย

อย่างไรก็ตาม ภาพจิตรกรรมไทยจำนวนไม่น้อยที่เสื่อมสภาพตามกาลเวลา และรอคوعการซ่อมแซมจากช่างในสมัยปัจจุบันที่ต้องไม่สร้างความเสียหายให้กับภาพเขียนเพิ่มขึ้นมากกว่าเดิม ที่ผ่านมาภาพที่ผ่านการซ่อมแซมมาแล้วจำนวนไม่น้อยได้รับความเสียหายหลังจากการซ่อมแซม ถึงแม้สภาพโดยรวมของภาพจิตรกรรมเดิมยังคงอยู่ แต่รายละเอียดในตัวภาพเขียนได้เปลี่ยนไป ก่อให้เกิดความเสียหายที่ประเมินค่าไม่ได้

การซ่อมแซมภาพจิตรกรรมไทยโบราณโดยใช้การต่อเติมภาพเป็นขั้นตอนของการซ่อมแซมแบบหนึ่งซึ่งไม่ก่อให้เกิดความเสียหายได้ ๆ กับภาพเดิม เนื่องจากเป็นการซ่อมแซมโดยการใช้ขั้นตอนวิธีเชิงตัวเลขบนภาพดิจิตัลซึ่งเป็นสำเนาของภาพเดิม ด้วยเหตุผลดังกล่าว ผู้วิจัยได้เลือกใช้วิธีการซ่อมแซมภาพจิตรกรรมไทยโบราณมีความจำเป็นเร่งด่วน เนื่องจากภาพที่ได้รับการซ่อมแซมด้วยการต่อเติมภาพสามารถนำไปใช้ประกอบการตัดสินใจเพื่อวางแผนก่อนการลงมือซ่อมแซมภาพเขียนจริงได้ นอกจากนี้ ขั้นตอนวิธีการต่อเติมภาพสามารถนำไปใช้สร้างแอปพลิเคชันโทรศัพท์เคลื่อนที่เพื่อในไปใช้เป็นข้อมูลในการเข้าชมภาพเขียนเดิมที่ยัง

ไม่ได้รับการซ่อมแซมและภาพเขียนที่ได้รับการซ่อมแซมโดยวิธีการทางคณิตศาสตร์จากแอปพลิเคชันที่พัฒนาขึ้น

รูปที่ 1.2 แสดงตัวอย่างภาพจิตรกรรมไทย<sup>1</sup> ที่ต้องได้รับการซ่อมแซมบนบริเวณแขนสีอ่อนของรูปปานาผู้ชายที่มีส่วนของสีแดงเดิมหลุดหายไป ทั้งนี้ในการซ่อมแซมภาพโดยการต่อเติมภาพ เราจะเริ่มด้วยการสร้างโคลเมนต่อเติมบนบริเวณสีพื้นผิวปูนที่แขนเสื้อ จากนั้นจึงนำขั้นตอนวิธีการต่อเติมภาพเพื่อซ่อมแซมภาพบริเวณนั้นให้เป็นสีแดง



รูปที่ 1.2: ภาพจิตรกรรมไทยที่วัดภูมินทร์ อำเภอเมือง จังหวัดน่าน

## 1.2 การลับบทบรรยายบนอนิเมะ

อนิเมะคือวิดีโอภาพวดการ์ตูนสต็อกญี่ปุ่นซึ่งเป็นที่นิยมของเยาวชนไทย ใน การรับชมอนิเมะ แม้ว่าเยาวชนไทยสามารถรับชมด้วยบทพากย์เสียงภาษาไทย แต่ก็สูญเสียอรรถรสของการรับชมจากบทบรรยายแบบแข็ง<sup>2</sup> (hardsub) ที่เป็นภาษาต่างประเทศในบริเวณด้านล่างของจอภาพ ในการซ่อมแซมอนิเมะด้วยการลับบทบรรยายภาษาต่างประเทศจึงเป็นงานที่ยุ่งยากและท้าทายมาก เนื่องจาก

- (1) อนิเมะเป็นวิดีโอซึ่งแสดงผลประมาณ 24 เฟรม(ภาพ)ต่อวินาที
- (2) แต่ละเฟรมอาจมีหรืออาจไม่มีบทบรรยายก็ได้
- (3) แต่ละเฟรมอาจมีหรืออาจไม่มีบทบรรยายเดียวกันก็ได้
- (4) แต่ละเฟรมเป็นการแสดงภาพสีที่มีระดับความคมชัดสูง (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 การพัฒนาขั้นตอนวิธีการต่อเติมภาพที่สามารถกำหนดโดยเมนูต่อเติมเชิงอัตโนมัติให้กับแต่ละเฟรมและประมวลผลได้แม่นยำจากการลับบทบรรยายสามารถทำงานได้แบบเรียลไทม์จึงเป็นสิ่งจำเป็นที่หลีกเลี่ยงไม่ได้

รูปที่ 1.3 แสดงตัวอย่าง 1 เฟรมของอนิเมะที่มีบทบรรยายแบบแข็ง<sup>3</sup> ที่ต้องซ้อมแซมด้วยการลับบทบรรยายออก ทั้งนี้ในการลับบทบรรยายออกจากเฟรมโดยใช้การต่อเติมภาพ เราจะเริ่มด้วยการสร้างโดยเมนูต่อเติมแบบอัตโนมัติในบริเวณบทบรรยาย จากนั้นจึงนำขั้นตอนวิธีการต่อเติมภาพแบบเร็วเพื่อลับบทบรรยายออกจากเฟรม



รูปที่ 1.3: 1 เฟรมของอนิเมะที่มีบทบรรยายแบบแข็ง

โครงการวิจัยนี้ ผู้วิจัยมีเป้าหมายสำคัญคือการพัฒนาขั้นตอนวิธีการต่อเติมภาพแบบเร็วและแม่นยำ ชนิดใหม่เพื่อนำไปใช้สำหรับซ้อมแซมภาพจิตกรรมไทยและการลับบทบรรยายออกจากอนิเมะ

## 2 วรรณกรรมและทฤษฎีบทที่เกี่ยวข้อง

ในการกล่าวถึงขั้นตอนวิธีการต่อเติมภาพ จะเริ่มต้นด้วยการกล่าวบททวนเกี่ยวกับการต่อเติมภาพเฉดสีเทา (grayscale image) ก่อน ดังนี้

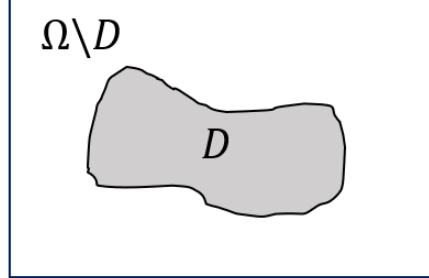
ให้  $\Omega \subset \mathbb{R}^2$  แทนโดเมนภาพ (image domain)  $D \subset \mathbb{R}^2$  แทนโดเมนต่อเติม (ดูรูปที่ 2.1) และ  $V \subset [0, \infty)$

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

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

นอกจากนี้ความสามารถสมมติได้โดยไม่เสียหลักการสำคัญว่า  $\Omega = [1, n]^2$  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> 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

วิธีเดินเวลาแบบชั้ดเจ้งสำหรับภาพเขตเทาเมื่อขั้นตอนวิธีดังนี้

**Algorithm 1:** ETM Method for TV-based Image Inpainting

```

 $u \leftarrow EMT(u, z, \lambda, \beta, \tau, N, \varepsilon)$ 
initialize  $i = 0; z = u; err = 1$ 
while  $i < N$  and  $err > \varepsilon$  do
     $u^{old} = u$ 
     $u = u + \tau \left( \nabla \cdot \left( \frac{\nabla u}{\sqrt{u_x^2 + u_y^2 + \beta}} \right) + \lambda(u - z) \right)$ 
     $err = \frac{\|u - u^{old}\|}{\|u\|}$ 
     $i = i + 1$ 
end
end

```

(2) วิธีการทำจุดตรึง (fixed-point iteration method)

คณะวิจัย [3] ได้แนะนำวิธีการเชิงตัวเลขสำหรับการกำจัดสัญญาณรบกวนโดยใช้วิธีการทำจุดตรึง ซึ่งสามารถประยุกต์เป็นวิธีเชิงตัวเลขสำหรับการต่อเติมภาพได้ดังนี้

เริ่มจากแนะนำด้วยการทำจุดตรึง  $\nu = 0, 1, 2, \dots$  และนิยามรูปแบบการทำจุดโดย

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

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

เพื่อหลีกเลี่ยงปัญหาเชิงตัวเลขจะเกิดขึ้นใน (2.4) และ (2.5) เราจะใช้

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

## วิธีการทำข้ามแบบจุดต้องมีขั้นตอนดังนี้

**Algorithm 2:** FP Method for TV-based Image Inpainting

---

```

FP( $u, z, \lambda, \beta, N, \varepsilon$ )
    initialize  $i = 0; u = z; err = 1$ 
    while  $i < N$  and  $err > \varepsilon$  do
         $u^{old} = u$ 
         $u = GS(u, z, \lambda, \beta, N_{gs})$ 
         $err = \frac{\|u - u^{old}\|}{\|u\|}$ 
         $i = i + 1$ 
    end
    return  $u$ 
end

 $u \leftarrow GS(u, z, \lambda, \beta, N_{gs})(u, z, \lambda, \beta, N_{gs})$ 
    initialize  $k = 0$ 
    compute  $D(u)_{i,j} = \frac{1}{\sqrt{u_x^2 + u_y^2 + \beta}}, 1 \leq i \leq n_x, 1 \leq j \leq n_y$ 
    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
    return  $u$ 
end

```

---

จาก (2.4) และ (2.5) เรายังสามารถพิสูจน์ได้ว่า การแก้สมการ (2.4) และ (2.5) ยังคงมีความแม่นยำของตัวแบบ (2.1) อยู่เมื่อมากขึ้นเท่านั้น นอก จากนี้ รายังพอบอกว่า การแก้สมการ (2.4) และ (2.5) ยังมีความยุ่งยากมากขึ้นสำหรับ  $\beta$  ที่มีค่าน้อยๆ

เพื่อเอาชนะความยากเชิงตัวเลขนี้ คณะวิจัยโดย [4] ได้แนะนำวิธีการสปริทเบรกแวนซ์สามารถถูกล่างถึงพื้นที่ของข้อมูล ดังนี้

(3) วิธีการสปริทเบรกแม่น (Split Bregman method)

เริ่มจากการແນະນຳເວກເຕອບສໍາເລັດ  $\mathbf{w}$  ພາຣາມີເຕອບເບຣກແມນ (Bregman parameter)  $\mathbf{b}$  ແລະ ພາຣາມີເຕອບພັນລື້ (panalty parameter)  $\theta > 0$  ແລະ ເປີຍິນ (2.1) ໃໝ່ ດັ່ງນີ້

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

ສໍາຫຼັບການທາຄຳຕອບຂອງ (2.6) ເຮັດວຽກໃຊ້ວິທີການທາຄ່າຕໍ່ສຸດແບບສລັບ (alternating minimization method) ໂດຍເຮັດວຽກການຕີ້ງ  $\mathbf{w}^{\text{old}}$  ແລະ  $\mathbf{b}^{\text{old}}$  ຈາກນັ້ນແກ້ປຸ່ນຫຍ່ອຍ

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

ຈາກນັ້ນໃຊ້  $u^{\text{New}}$  ທີ່ໄດ້ຈາກການແກ້ປຸ່ນຫຍ່ອຍໃນ (2.7) ເພື່ອແກ້ປຸ່ນຫຍ່ອຍ່ອຍ

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

ສຸດທ້າຍຈຶ່ງປະກຳປຸ່ນຫຍ່ອຍ

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

ດໍາເນີນການເຂັ້ນນີ້ຈະຮັບສູນ  $\|u^{\text{new}} - u^{\text{old}}\| < \epsilon_1$  ຢ່າງ ອີ່ວນ  $\epsilon_1 > \epsilon_2$  ເນື້ອ  $\epsilon_1, \epsilon_2 > 0$  ວິທີການສປິຖິກ  
ເບຣກແມນມີຂັ້ນຕອນວິທີດັ່ງນີ້

---

**Algorithm 3:** SB method for TV-based Image Inpainting

---

```

 $u \leftarrow SB(u, z, \lambda, \theta, N_{gs}, N, \epsilon)$ 
    initialize  $i = 0, \mathbf{b} = \vec{0}, \mathbf{w} = \vec{0}, u = u_0$ 
    while  $i < N$  and  $err > \epsilon$  do
         $u^{old} = u; w^{old} = w; b^{old} = b;$ 
         $u = \arg \min_u \{\mathcal{J}_1(u) = \frac{1}{2} \int_{\Omega} \lambda(u - z)^2 d\Omega + \frac{\theta}{2} \int_{\Omega} (\mathbf{w}^{\text{old}} - \nabla u + \mathbf{b}^{\text{old}}) d\Omega\}$ 
         $w = \arg \min_{\mathbf{w}} \{\mathcal{J}_2(\mathbf{w}) = \int_{\Omega} |\mathbf{w}| d\Omega + \frac{\theta}{2} \int_{\Omega} (\mathbf{w} - \nabla u^{\text{New}} + \mathbf{b}^{\text{old}}) d\Omega\}$ 
         $b = b^{\text{old}} + \nabla u - w$ 
         $err = \frac{\|u - u^{old}\|}{\|u\|}$ 
         $i = i + 1$ 
    end
end

```

---

หมายเหตุ:

- (1) แก้ปัญหา  $u = \arg \min_u \mathcal{J}_1(u)$  ด้วยการคำนวณคำตอบของ  $-\theta \Delta u + \lambda u = \lambda z - \theta \nabla \cdot (\mathbf{w} - \mathbf{b})$  โดยใช้ Finite Difference Method และการทำซ้ำ Gauss-seidel จำนวน  $N_{gs}$  รอบ
- (2) แก้ปัญหา  $\mathbf{w} = \arg \min_w \mathcal{J}_2(\mathbf{w})$  ด้วยการคำนวณคำตอบของ  $\mathbf{w} = \max((\nabla u + \mathbf{b}) - \frac{1}{\theta}, 0)$

## 2.2 ตัวแบบการต่อเติมภาพสีที่ใช้การแปรผันรวม

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

ในการทำงานเดียวกันกับตัวแบบการต่อเติมภาพเอดสีเทาที่ใช้การแปรผันรวม 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 จะเป็นการแก้ปัญหาการหาค่าต่ำที่สุดต่อไปนี้

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

สำหรับขั้นตอนวิธีสปริตเบรกเม้นสำหรับภาพสี แสดงได้ดังนี้

---

**Algorithm 4:** Split-bregman Color solver

---

```
 $\mathbf{u} \leftarrow SBC(\mathbf{u}, \mathbf{z}\lambda, \theta, N_{gs}, N, \varepsilon)$ 
  |   for  $l = 1 : 3$  do
  |   |    $u_l \leftarrow SB(u_l, z_l\lambda, \theta, N_{gs}, N, \varepsilon)$ 
  |   |   end
  |   end
```

---

### 2.3 การวัดประสิทธิภาพของภาพที่ผ่านกระบวนการต่อเติม

การประเมินคุณภาพของการต่อเติมภาพของวิธีการเชิงตัวเลข จะใช้ค่า PSNR [6] และ SSIM [7] โดย

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

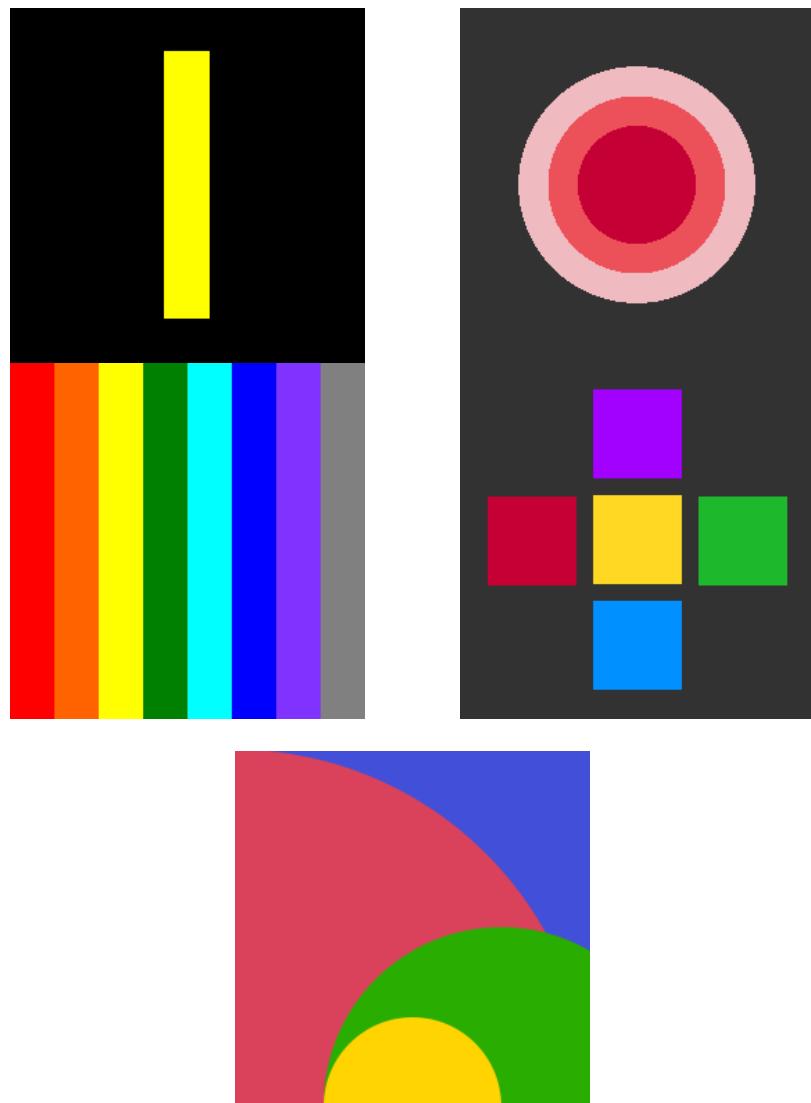
$$SSIM(u, \tilde{u}) = \frac{(2\mu_u\mu_{\tilde{u}} + 0.0001)(2\sigma_{u\tilde{u}} + 0.0009)}{(\mu_u^2 + \mu_{\tilde{u}}^2 + 0.0001)(\sigma_u^2 + \sigma_{\tilde{u}}^2 + 0.0009)}$$

เมื่อ MSE คือ mean square error ของภาพ  $u$  และภาพ  $\tilde{u}$  โดยที่ภาพ  $u$  และภาพ  $\tilde{u}$  แทนภาพต้นฉบับ และภาพที่ได้จากการซ่อมแซมโดยวิธีเชิงตัวเลข

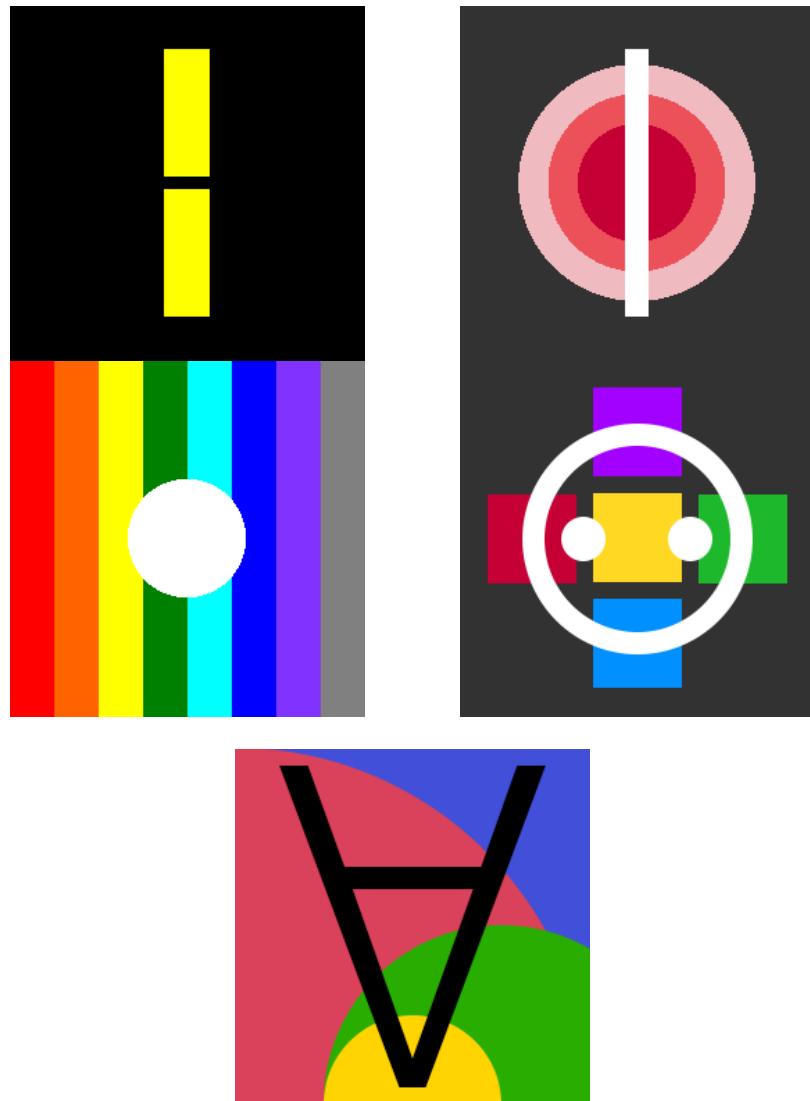
## 3 ผลการดำเนินงานเบื้องต้น

### 3.1 การซ่อมแซมภาพจิตรกรรมไทยโบราณ

สำหรับการซ่อมแซมจิตรกรรมไทยโบราณ จะเริ่มจากการทำการปรับปรุงขั้นตอนวิธีเชิงตัวเลขที่มีอยู่แล้วให้ดีขึ้นเสียก่อน โดยระหว่างการปรับปรุงวิธีเชิงตัวเลข จะใช้ภาพสีที่ได้สังเคราะห์ขึ้นทั้งสิ้น 5 ภาพ โดยแต่ละภาพมีขนาด  $256 \times 256$  พิกเซล ซึ่งมีดังนี้



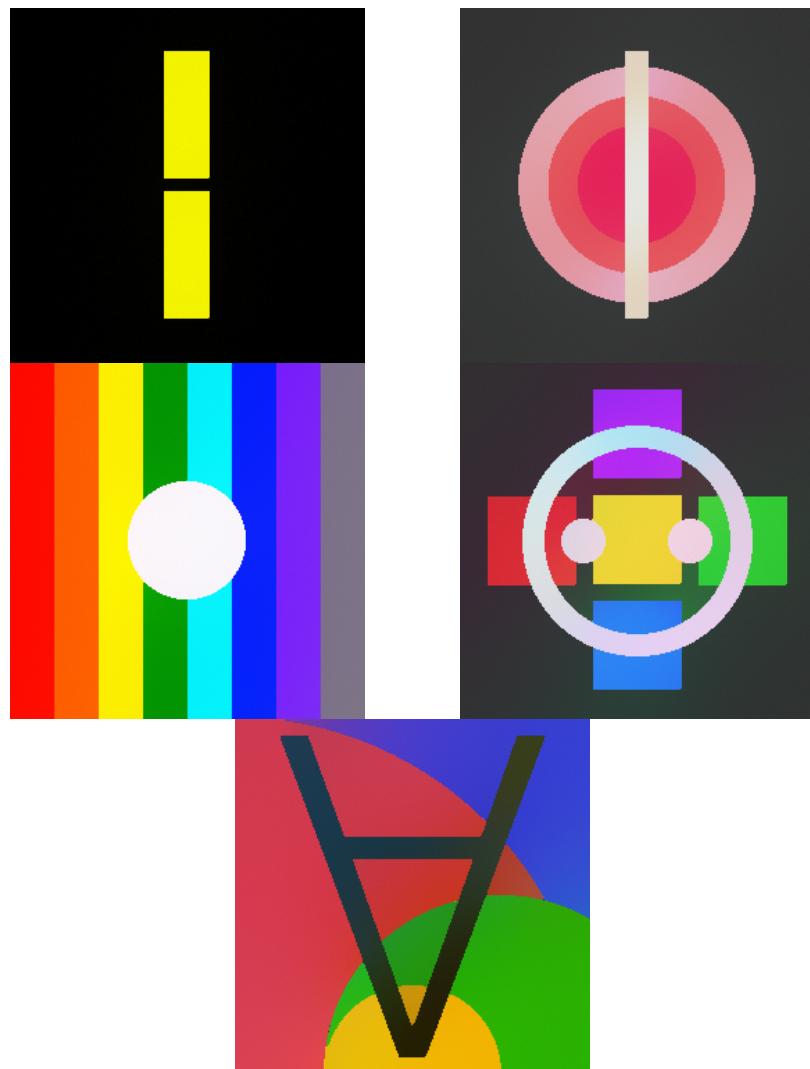
รูปที่ 3.1: ภาพต้นฉบับ



รูปที่ 3.2: ภาพที่จะทำการซ่อมแซม

### 3.1.1 การเปรียบเทียบประสิทธิภาพขั้นตอนวิธีเชิงตัวเลขที่มีอยู่สำหรับตัวแบบต่อเติมภาพสีที่ใช้การแปรผันรวม

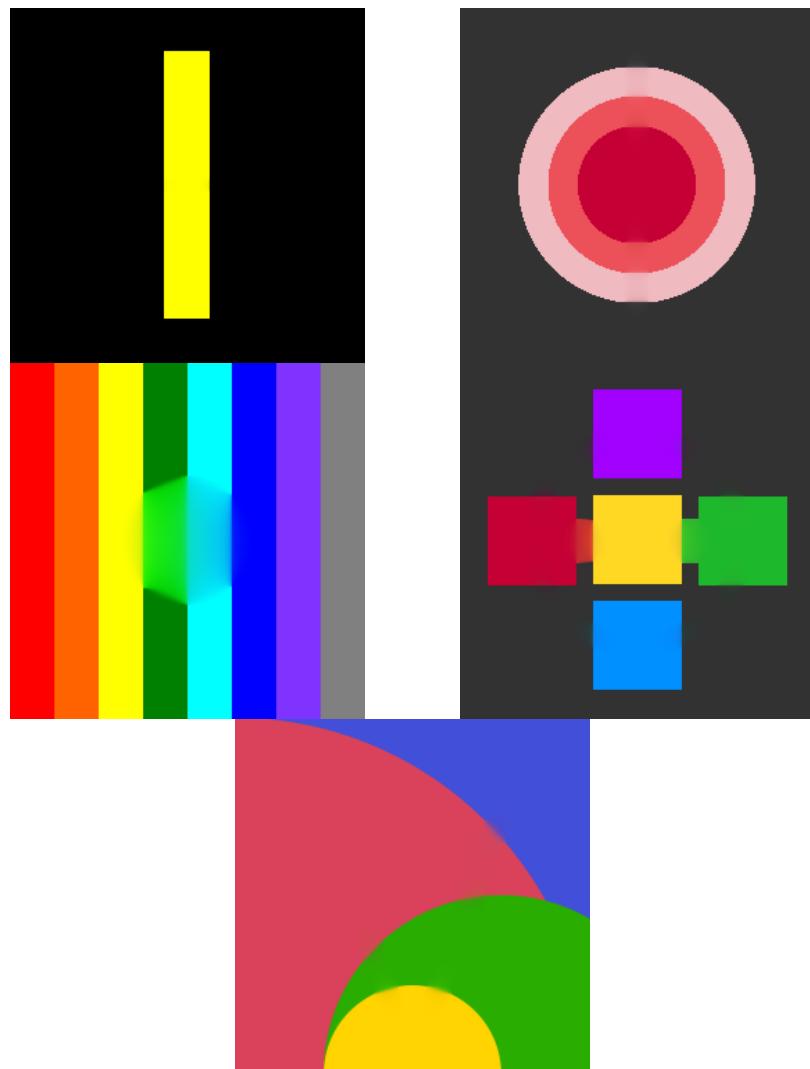
การทดสอบประสิทธิภาพจะใช้  $\varepsilon = 1 \times 10^{-4} N = 10,000$  โดยมีการซ่อมแซมภาพสังเคราะห์ทั้ง 5 ภาพดังนี้



รูปที่ 3.3: 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

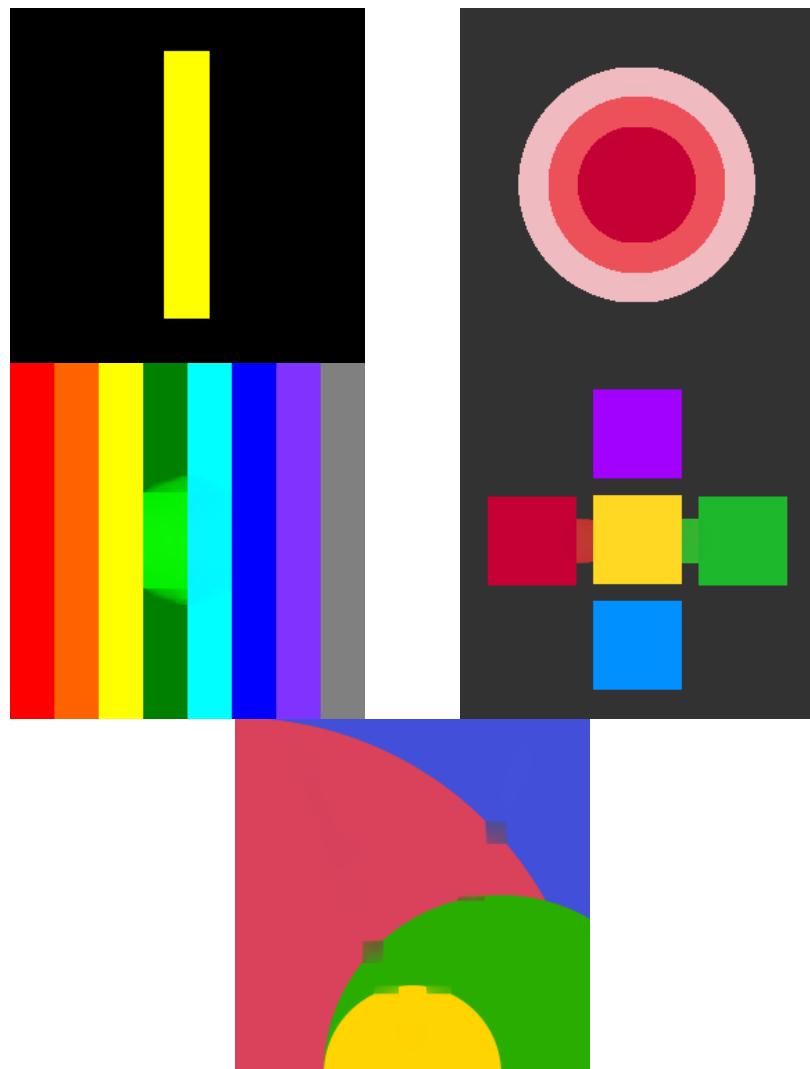
ตารางที่ 1: ผลการซ่อมแซมวิธีการเดินเวลา



รูปที่ 3.4: ผลการซ่อมแซมจากการวิจัยการจุดตรีง

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 2: ผลการซ่อมแซมของวิจัยจุดตรีง



รูปที่ 3.5: ผลการซ่อมแซมจากการสปริทเบรกเมน

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 3: ผลการซ่อมแซมของวิธีสปริทเบรกเมน

ประสิทธิภาพของวิธีการเชิงตัวเลขทั้ง 3 วิธี สามารถสรุปได้ดังนี้

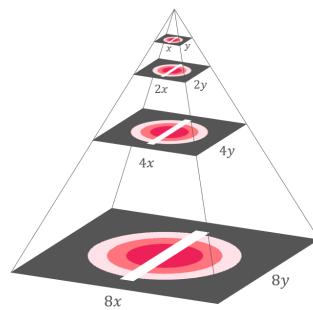
วิธีการ	เวลาประมวล (วินาที)	PSNR (dB)	SSIM
การเดินเวลา	120.68	16.72	0.9960
การทำซ้ำจุดตรึง	74.81	38.67	0.9999
การสปริทเบรกแม่น	14.06	39.42	0.9999

ตารางที่ 4: แสดงการซ่อมแซมเฉลี่ยของวิธีการเชิงตัวเลข

ซึ่งจากทั้ง 3 วิธีที่ได้ทดสอบ จะเห็นได้ว่าวิธีการสปริทเบรกแม่นใช้เวลาอย่างกว่าวิธีอื่น และมีคุณภาพซึ่งพิจารณาจาก ค่า PSNR และค่า SSIM มากกว่าวิธีอื่น ซึ่งผู้วิจัยจะใช้วิธีสปริทเบรกแม่นมาใช้ในการพัฒนาวิธีต่อเดิมภาพเป็นลำดับถัดไป

### 3.1.2 ขั้นตอนวิธีการสำหรับต่อเติมภาพชนิดใหม่

จากวิธีการสปริทเบรกแม่นนั้นจะใช้วิธีการทำห้ามกระทำการทำซ้ำบนกราฟทั้งสู่เข้า ทางผู้ศึกษาจึงสนใจที่หาคำตอบเริ่มต้นสำหรับการทำซ้ำที่ดีขึ้น เพื่อทำให้การทำซ้ำสู่เข้าสู่คำตอบได้เร็วขึ้น โดยการทำงานกับรูปภาพที่เล็กกว่า จากนั้นจึงทำการขยายผลลัพธ์ที่ได้ขึ้นมาทำกับภาพใหญ่ ซึ่งวิธีนี้เรียกว่าวิธีพิริมิตรูปภาพ (pyramid methods) [5] โดยผู้วิจัยจะทำการย่อขนาดรูปลงครึ่งหนึ่งโดยใช้วิธี Bilinear Interpolation ทั้งสิ้น 4 ครั้ง จากนั้นเริ่มทำการต่อเติมภาพขนาดเล็ก จากนั้นนำผลลัพธ์ที่ได้จากการขนาดเล็ก ทำการขยายภาพขึ้นสองเท่าโดยใช้ Bilinear Interpolation ก่อนจะนำเฉพาะส่วนที่อยู่ในโดเมนต่อเติมของภาพที่ถูกขยายมาทำการต่อเติมเพื่อให้ส่วนที่ถูกขยายขึ้นมาเป็นคำตอบเริ่มต้นสำหรับการต่อเติมภาพในขั้นถัดไป



รูปที่ 3.6: วิธีการพิริมิตรูปภาพ

ขั้นตอนวิธีสำหรับการทำพิริมิตรูปภาพสำหรับการต่อเติมภาพแบบสปริทเบรกแม่นเพื่อให้ประมวลผลได้เร็วขึ้นสามารถสรุปได้ดังนี้

---

**Algorithm 5:** Multi-resolution SB method

---

```
u ← MRSBC( $\mathbf{u}, \mathbf{z}, \lambda, \theta, N_{gs}, N_0, N_1, N_2, \varepsilon, c, m$ )
    Initialize height = height of  $\mathbf{u}$ , width = width of  $\mathbf{u}$ 
    if  $c < m$  then
        |  $x = \text{Bilinear}(\mathbf{u}, \lfloor \text{width} * 0.5 \rfloor, \lfloor \text{height} * 0.5 \rfloor)$ 
        |  $y = \text{Bilinear}(\lambda, \lfloor \text{width} * 0.5 \rfloor, \lfloor \text{height} * 0.5 \rfloor)$ 
        |  $r = \text{MRSBC}(x, y, \lambda, \theta,$ 
        |  $N_{gs}, N_0, N_1, N_2, \varepsilon, c + 1, m)$ 
        |  $\mathbf{u} = \text{Bilinear}(r, \text{width}, \text{height})$ 
    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 = SBC( $\mathbf{u}, \mathbf{z}, \lambda, \theta, N_{gs}, N_{SB}, \varepsilon$ )
end
```

---

**Algorithm 6:** Bilinear Interpolation

---

```
J ← Bilinear(I, x, y)
    Initialize J is image that height x and width y,
    v = height of I and w is 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
end
```

---

โดยจะทำการเบรียบเทียบจำนวนครั้งในขั้นที่รูปภาพมีขนาดเล็ก จนไปถึงขั้นที่มีขนาดใหญ่ ตัวอย่าง เช่น 10/3/3/10000 หมายถึงในขั้นเล็กสุดซึ่งขนาดเป็น 32x32 พิกเซลจะทำซ้ำไม่เกิน 10 ครั้ง ขั้นถัดมาขนาด เป็น 64x64 จะทำซ้ำไม่เกิน 3 ครั้ง และขั้นถัดมา ขั้นถัดมาขนาดเป็น 128x128 จะทำซ้ำ 3 ครั้ง และสุดท้าย ขนาด 256x256 จะทำซ้ำไม่เกิน 10,000 ครั้งหรือจนค่าความคลาดเคลื่อนสัมพัทธ์ต่างกันไม่เกิน 0.0001 ซึ่ง เมื่อทำการทดสอบแล้วได้ผลลัพธ์ดังตารางนี้

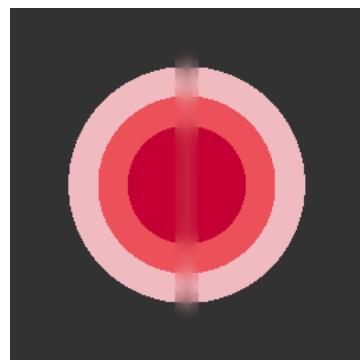
รูปแบบการทำซ้ำ	รูปภาพ	เวลาประมาณ (วินาที)	PSNR (dB)	SSIM
ไม่ใช้พิริมิดรูปภาพ	1	4.49	71.54	1.0000
	2	13.16	37.08	1.0000
	3	29.46	26.08	0.9997
	4	20.50	29.61	0.9999
	5	19.32	32.78	1.0000
10/1/1/10000	1	2.44	69.59	1.0000
	2	11.31	37.04	1.0000
	3	23.48	27.34	0.9998
	4	16.60	29.42	0.9999
	5	13.75	33.53	1.0000
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 5: แสดงผลลัพธ์ของการใช้พิริมิดภาพในการต่อเติมในขั้นที่ต่างกัน

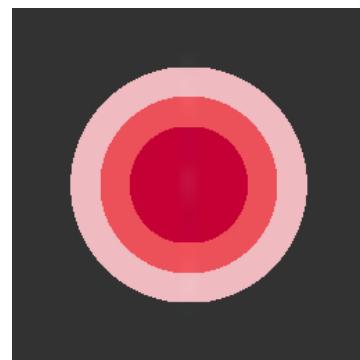
รูปแบบการทำข้า	เวลาประมาณ (วินาที)	PSNR (dB)	SSIM
ไม่ใช้พิริเมติรูปภาพ	17.38	39.42	0.9999
10/1/1/10000	13.52	39.38	0.9999
10/3/3/10000	11.86	39.54	0.9999
10/10/10/10000	9.26	40.17	0.9999
100/1/1/10000	10.28	39.04	0.9999
100/3/3/10000	10.28	39.80	0.9999
100/10/10/10000	9.27	40.12	0.9999

ตารางที่ 6: แสดงผลลัพธ์เฉลี่ยของการใช้พิริเมติภาพในการต่อเติมในชั้นที่ต่างกัน

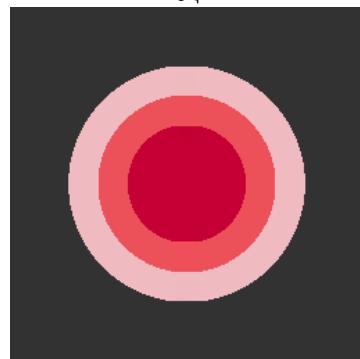
จากตารางจะสังเกตว่า ยิ่งจำนวนการทำข้าในชั้นที่รูปภาพมีขนาดเล็กจำนวนมากครั้ง จะยิ่งทำให้เวลาประมาณลดลง ซึ่งทำให้การทำการต่อเติมภาพ ใช้เวลาอยู่ในช่วงแรก จากนั้นความเร็วในการต่อเติมลดลง ซึ่งทำให้การทำการต่อเติมภาพเพียงไม่กี่ครั้งในรูปภาพขนาดใหญ่สุด มีผลลัพธ์ใกล้เคียงกับภาพต้นฉบับได้



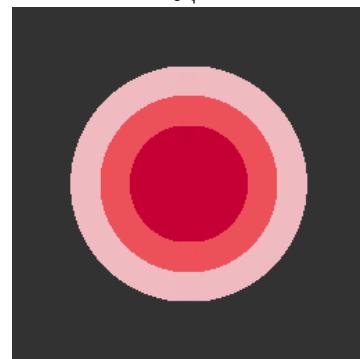
(a) ชั้นใหญ่สุด 10 ครั้ง



(b) ชั้นใหญ่สุด 100 ครั้ง



(c) ชั้นใหญ่สุด 1,000 ครั้ง



(d) ชั้นใหญ่สุด 10,000 ครั้ง

รูปที่ 3.7: พิริเมติที่ลำดับการทำข้าเป็น 10/10/10 และชั้นใหญ่สุดทำในจำนวนครั้งที่ต่างกัน

โดยผู้วิจัยจึงกำหนดให้การทำข้าในรูปภาพขนาดใหญ่สุดมีการทำข้าเพียง 10 ครั้ง และพบว่าได้ผลลัพธ์ดังนี้

รูปแบบการทำข้า	รูปภาพ	เวลาประมาณ (วินาที)	PSNR (dB)	SSIM
ไม่ใช้พีระมิดรูปภาพ	1	0.30	26.71	0.9998
	2	0.39	18.39	0.9982
	3	0.38	13.66	0.9944
	4	0.40	12.86	0.9934
	5	0.38	14.69	0.9956
10/1/1/10	1	0.29	40.10	1.0000
	2	0.41	31.28	0.9999
	3	0.46	16.51	0.9970
	4	0.47	26.56	0.9998
	5	0.39	28.25	0.9998
10/3/3/10	1	0.28	42.53	1.0000
	2	0.36	32.91	1.0000
	3	0.35	16.88	0.9972
	4	0.34	27.06	0.9998
	5	0.34	29.76	0.9999
10/10/10/10	1	0.31	50.06	1.0000
	2	0.41	34.01	1.0000
	3	0.38	18.19	0.9980
	4	0.39	27.50	0.9998
	5	0.40	33.05	1.0000
100/1/1/10	1	0.27	43.97	1.0000
	2	0.37	31.28	0.9999
	3	0.36	24.98	0.9997
	4	0.36	28.05	0.9998
	5	0.36	29.24	0.9999
100/3/3/10	1	0.29	45.08	1.0000
	2	0.36	32.36	0.9999
	3	0.40	24.35	0.9996
	4	0.38	27.88	0.9998
	5	0.37	30.28	0.9999
100/10/10/10	1	0.28	50.05	1.0000
	2	0.41	33.25	1.0000
	3	0.42	23.51	0.9995
	4	0.42	27.78	0.9998
	5	0.39	32.38	0.9999

ตารางที่ 7: แสดงผลลัพธ์ของการใช้พีระมิดภาพในการต่อเติมโดยกำหนดให้ขั้นรูปภาพขนาดใหญ่สุดทำข้าเพียง 10 ครั้ง

รูปแบบการทำซ้ำ	เวลาประมาณ (วินาที)	PSNR (dB)	SSIM
ไม่ใช้พิระมิดรูปภาพ	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 8: แสดงผลลัพธ์เฉลี่ยของการใช้พิระมิดภาพในการต่อเติมโดยกำหนดให้ชั้นรูปภาพขนาดใหญ่สุดทำซ้ำเพียง 10 ครั้ง

จากการจะเห็นว่า การทำซ้ำในชั้นที่รูปภาพมีขนาดเล็กมากจำนวนมาก ไม่ช่วยให้การประมวลผลได้เร็วขึ้น ผู้วิจัยจึงเลือกใช้การทำซ้ำแบบ 10/3/3/10 ในการต่อเติมภาพ

### 3.1.3 การทดสอบประสิทธิภาพในการซ่อมแซมภาพจิตกรรมไทยโบราณ

ชิ้นภาพจิตกรรมที่ใช้ทดสอบ มีทั้งสิ้น 5 ภาพ โดยแต่ละภาพเป็นภาพสีที่มีขนาด  $256 \times 256$  พิกเซล ซึ่งทั้ง 5 ภาพได้แก่ ภาพที่ 3.8a<sup>4</sup> และภาพที่ 3.8b<sup>5</sup> คือ จิตกรรมฝาผนังวัดแก้วไพทุรย์ ภาพที่ 3.8c<sup>6</sup> คือ จิตกรรมฝาผนังวัดพระยืนพุทธบາทყຸคล ภาพที่ 3.8d<sup>7</sup> คือ จิตกรรมฝาผนังวัดคงคaram และภาพที่ 3.8e<sup>8</sup> คือ จิตกรรมฝาผนังวัดท่าถนน โดยจะทำให้ข้อมูลข้างทั้ง 5 ภาพเกิดความเสียหาย โดยใช้ร้อยความเสียหายจากภาพพระเจ้าสร้างอดัม

<sup>4</sup>ภาพถ่ายที่วัดแก้วไพทุรย์; 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

<sup>5</sup>ภาพถ่ายที่วัดแก้วไพทุรย์; 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

<sup>6</sup>ภาพถ่ายที่วัดพระยืนพุทธบາทყຸคล; ภาพจาก [https://commons.wikimedia.org/wiki/File:Wat\\_Phra\\_Yuen\\_Phutthabat\\_Yukhon\\_01.jpg](https://commons.wikimedia.org/wiki/File:Wat_Phra_Yuen_Phutthabat_Yukhon_01.jpg) สืบคันเมื่อวันที่ 23 กันยายน 2561

<sup>7</sup>ภาพถ่ายที่วัดคงคaram; ภาพจาก [https://commons.wikimedia.org/wiki/File:จิตกรรม\\_อุบลสกัดคงคaram.JPG](https://commons.wikimedia.org/wiki/File:จิตกรรม_อุบลสกัดคงคaram.JPG) สืบคันเมื่อวันที่ 23 กันยายน 2561

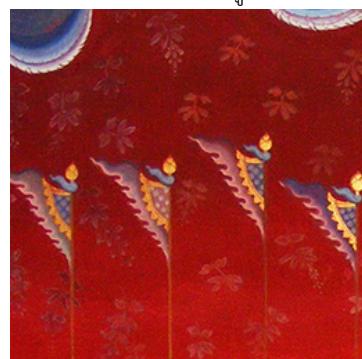
<sup>8</sup>ภาพถ่ายที่วัดท่าถนน; ภาพจาก [https://commons.wikimedia.org/wiki/File:Wat\\_Tha\\_Thanon\\_05.JPG](https://commons.wikimedia.org/wiki/File:Wat_Tha_Thanon_05.JPG) สืบคันเมื่อวันที่ 23 กันยายน 2561



(a) วัดแก้วไพทุรย์



(b) วัดแก้วไพทุรย์



(c) วัดพระยืนพุทธบาทยุคล



(d) วัดคงคาราม



(e) วัดท่าถนน



(f) รอยความเสียหาย

รูปที่ 3.8: ภาพต้นฉบับสำหรับใช้ในการทดสอบ



รูปที่ 3.9: ภาพที่ทำให้เสียหาย

จากนั้นทำการทดสอบการต่อเติมภาพทั้ง 5 โดยทดสอบวิธีสปริทเบรกแมน และวิธีที่พัฒนาขึ้นโดยใช้วิธีการสปริทเบรกแมนพร้อมทั้งการใช้พิระมิดรูปภาพที่มีการทำข้าแต่ละชั้นเป็น 10/3/3/10 ได้ผลลัพธ์ออกมาเป็นดังนี้



รูปที่ 3.10: ผลลัพธ์จากการสปีดเบรกแม่น

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 9: ผลลัพธ์การซ่อมแซมภาพศิลปะไทยจากการสปีดเบรกแม่น



รูปที่ 3.11: ผลลัพธ์จากการที่พัฒนาขึ้น

รูปภาพ	เวลาประมวล (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 10: ผลลัพธ์การซ่อมแซมภาพศิลปะไทยจากวิธีที่พัฒนาขึ้น

ซึ่งทั้งสองวิธี ได้ผลลัพธ์การซ้อมแซมศิลปะไทยเฉลี่ยอกมาดังนี้

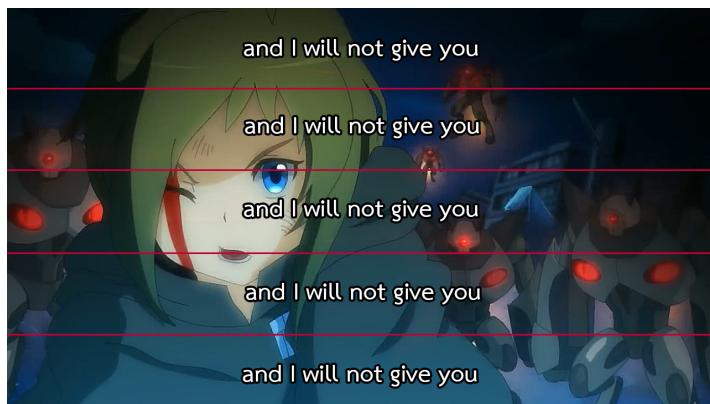
วิธีการ	เวลาประมวล (วินาที)	PSNR (dB)	SSIM
สเปรทเบรกแม่น	2.72	34.89	1.0000
วิธีการที่พัฒนาขึ้น	0.39	35.30	1.0000

ตารางที่ 11: แสดงผลลัพธ์เฉลี่ยของการซ้อมแซมภาคศีลปะไทย

จากตารางจะเห็นได้ว่า วิธีที่พัฒนาขึ้นนั้นสามารถทำงานได้เร็วกว่าวิธีสปริตเบรกเมนเดิม และยังมีคุณภาพที่ดีขึ้นด้วย

### 3.2 การลับบทบรรยายบนอนิเมะ

สำหรับการลับบทบรรยายอนิเมะ จะใช้วิดีโอ Anime Festival Asia Special Video - feat. Inori Aizawa ซึ่งผลิตโดย Collateral Damage Studios โดยจะตัดวิดีโอด้วยตัวเอง 1 นาทีแรกสำหรับการทดลอง โดยวิดีอดังกล่าวขนาด 1280 x 720 พิกเซล แต่เนื่องจากโดยปกติแล้ว อนิเมะมักมีบรรยาย 1 ถึง 2 บรรทัด จึงทำการแบ่งวิดีโອอกกือเป็น 5 ส่วนได้ขนาดเป็น 1280 x 144 พิกเซลก่อนนำไปทดสอบในลำดับถัดไป และสำหรับบทบรรยายที่จะใช้ทดสอบนั้น เป็นจากวิดีโอ Anime Festival Asia Special Video - feat. Inori Aizawa ไม่มีคำพูดใดๆ จึงใช้แทนความ lorem ipsum เป็นบทบรรยาย โดยจะทำการแสดงบทบรรยาย 1 บรรทัด ความยาว 3 วินาที ทุก 2 วินาที นั่นคือในวิดีโอดังกล่าวจะมีบทบรรยายทั้งสิ้น 20 บรรทัด



รูปที่ 3.12: การแบ่งไฟล์วิดีโอเป็น 5 ส่วนสำหรับใช้เป็น 5 ชุดทดสอบ

เนื่องจากไฟล์วิดีโอนั้นประกอบด้วยภาพที่ลักษณะทำให้เข้าใจต่อนิรธีการลับคำบรรยายออกจากวิดีโอ จะต้องทำการต่อเติมบริเวณที่เป็นบทบรรยายที่ลักษณะนั้นคือจะมีข้อความดังนี้

---

**Algorithm 7:** remove subtitle from video

---

```
 $V \leftarrow RemoveS(V)$ 
  foreach  $u_i$  เป็นเฟรมที่  $i$  ในวิดีโอ  $V$  do
    |   • หาโดยเมนต์อเติม  $D$  จาก  $u_i$ 
    |   • ต่อเติมภาพ  $u_i$  โดยใช้โดยเมนต์อเติม  $D$ 
  end
end
```

---

โดยขั้นตอนการต่อเติมภาพ  $u_i$  ด้วยโดยเมนต์อเติม  $D$  นั้นจะสามารถใช้วิธีการเดียวกับการซ่อมแซมภาพศิลปะไทยได้ ส่วนการทำทบทรร้ายอนิเมะ จะกล่าวถึงในขั้นถัดไป

### 3.2.1 การทำบทบรรยายบนอนิเมะ

ก่อนจะระบบบทบรรยายนั้น จำเป็นต้องทำบทบรรยายในภาพให้ได้เสียก่อน โดยบทบรรยายของอนิเมะนั้น มักจะใช้ขอบของตัวอักษรเป็นสีดำ อีกทั้งบทบรรยายนั้นจะลอยห่างจากขอบของวิดีโอ และขนาดของคำบรรยายนั้นจะมีขนาดอยู่ประมาณหนึ่งไม่ใหญ่หรือไม่เล็กเกินไป ด้วยสมบัตินี้เองทำให้จึงสามารถหาบริเวณบนเฟรมที่เป็นบทบรรยายได้โดยจะมีวิธีหาพื้นที่ซึ่งเป็นบทบรรยายดังนี้

---

**Algorithm 8:** finding subtitle

---

```
 $D \leftarrow findsu(\mathbf{u})$ 
  • ทำการเปลี่ยนสีดำในภาพ  $\mathbf{u}$  ให้เป็นสีขาวแล้วเปลี่ยนอื่นๆ ให้เป็นสีดำเพื่อหาขอบของคำ
    บรรยาย
  • เปลี่ยนพื้นที่สีขาวในภาพให้เป็นสีดำ และเปลี่ยนพื้นที่สีดำให้เป็นสีขาว
  • ทำการลบพื้นที่สีขาวซึ่งติดกับขอบของภาพออกไป เนื่องจากบทบรรยายจะลอยอยู่ ไม่ติด
    กับขอบเสมอ
  • ลบพื้นที่ๆ ใหญ่เกินกว่าจะเป็นบทบรรยาย
  • ลบพื้นที่ ซึ่งเล็กเกินกว่าจะเป็นบทบรรยาย
  • ทำการขยายพื้นที่ๆ เป็นสีขาวขึ้นด้วยความกว้างของขอบบทบรรยาย
  • สีขาวที่เหลืออยู่ในภาพจะเป็นบทบรรยาย
```

---

ซึ่งวิธีการหาบทบรรยายที่กล่าวไว้ไปข้างต้น จะทำการทดสอบกับบทความ *lorem ipsum* ที่ถูกแปลเป็นภาษาไทย ภาษาอังกฤษ และภาษาญี่ปุ่น โดยมีความสามารถในการหาโดยเมนต่อเติมในบทบรรยายภาษาต่างๆ ดังนี้

ภาษา	รีติโว	จำนวนพิกเซลในโดยเมน	จำนวนพิกเซลที่ตรวจพบ	จำนวนพิกเซลที่ผิดพลาด	ร้อยละการผิดพลาด
ไทย	1	23,190,522	24,044,004	2,108,772	9.09
	2	23,232,287	24,026,820	2,204,025	9.49
	3	23,189,082	24,300,589	2,081,340	8.98
	4	23,277,706	23,796,276	2,126,004	9.13
	5	23,221,502	24,247,935	2,185,864	9.41
อังกฤษ	1	27,281,185	28,631,063	3,477,960	12.75
	2	27,269,671	28,513,248	3,514,859	12.89
	3	27,325,148	28,611,300	3,815,082	13.96
	4	27,191,136	28,527,105	3,854,121	14.17
	5	27,326,584	28,709,405	3,909,582	14.31
ญี่ปุ่น	1	28,509,908	30,058,101	3,953,067	13.87
	2	28,534,363	30,023,923	3,565,609	12.50
	3	28,537,968	30,015,047	3,553,128	12.45
	4	28,579,778	30,065,985	3,961,319	13.86
	5	28,558,848	30,354,275	3,671,730	12.86

ตารางที่ 12: แสดงความคาดเคลื่อนของการหาโดยเมนต่อเติม ในบทบรรยายภาษาต่างๆ

ภาษา	จำนวนพิกเซลในโดยเมน	จำนวนพิกเซลที่ตรวจพบ	จำนวนพิกเซลที่ผิดพลาด	ร้อยละการผิดพลาด
ไทย	23,222,220	24,083,125	2,141,201	9.22
อังกฤษ	27,278,745	28,598,424	3,714,321	13.62
ญี่ปุ่น	28,544,173	30,103,466	3,740,971	13.11

ตารางที่ 13: แสดงความคาดเคลื่อนเฉลี่ยของการหาโดยเมนต่อเติม ในบทบรรยายภาษาต่างๆ

จากการทดลองทั้ง 3 ภาษาพบว่าวิธีการหาคำบรรยายนี้ มีร้อยละการผิดพลาดเฉลี่ยอยู่ที่ 11.98 ซึ่ง การทดลองจากนี้ไปจะใช้วิธีการหาคำบรรยายนี้ในการหาโดยเมนต่อเติมแบบอัตโนมัติ

### 3.2.2 การลบคำบรรยายจากหน้าเมะ

สำหรับหน้าเมะนั้น แต่ละเฟรมจะเป็นรูปภาพ เราจึงสามารถประยุกต์ใช้วิธีการซ่อมแซมภาพจิตกรรรมไทย มาใช้ในการลบคำบรรยายได้ แต่ผู้จัดทำได้สังเกตว่า สำหรับหน้าเมะที่เป็นวิดีโอแล้ว ในขณะที่ประมวลผลวิดีโอ เราสามารถใช้ผลการต่อเติมภาพจากภาพที่แล้ว มาใช้เป็นคำตอบเริ่มต้นวิธีดังนี้

**Algorithm 9:** subtitle remove with reference from previous frame

---

```

 $V \leftarrow removeS1(V)$ 
initialize  $i = 1, N_f$  เป็นจำนวนเฟรมใน  $V$ 
while  $i < N_f - 1$  do
     $\mathbf{u}_i$  คือเฟรมที่  $i$  ใน  $V$ 
     $u_{i+1}$  คือเฟรมที่  $i + 1$  ใน  $V$ 
     $D$  คือโคล เมนต่อเติมใน  $\mathbf{u}_{i+1}$ 
     $\mathbf{u}_{i+1} = removeS2(\mathbf{u}_i, D, u_{i+1})$ 
end
return  $V$ 
end

```

---

โดย  $removeS2(\mathbf{u}_i, D, u_{i+1})$  คือขั้นตอนวิธีที่ 10 ซึ่งในทำนองเดียวกันเราสามารถเปลี่ยน  $removeS2(u_i, D, \mathbf{u}_{i+1})$  เป็น  $removeS3(\mathbf{u}_i, D, u_{i+1})$  เพื่อใช้กับขั้นตอนวิธี 11 และเปลี่ยนเป็น  $removeS4(\mathbf{u}_i, D, \mathbf{u}_{i+1})$  เพื่อใช้กับขั้นตอนวิธี 12 ได้  
ขั้นตอนวิธี การยืมเฟรม จะเป็นการนำผลลัพธ์จากเฟรมก่อนหน้ามาเป็นคำตอบในการเริ่มต้นในการประมวลผลเพื่อให้ผลลัพธ์ถูกเข้าได้เร็วขึ้น

**Algorithm 10:** borrow frame technique

---

```

 $v \leftarrow removeS2(\mathbf{u}, D, v)$ 
 $s =$  ค่า SSIM ระหว่าง  $\mathbf{u}$  และ  $v$  บริเวณนอกโคล เมนต่อเติม
if  $s > 0.9$  then
    | คัดลอกบริเวณในโคล เมนต่อเติมจาก  $\mathbf{u}$  ไปยัง  $z$ 
end
 $v = MRSBC(\mathbf{u}, z, \lambda, \theta, N_{gs}, N_0, N_1, N_2, \varepsilon, 1, m)$ 
end

```

---

ขั้นตอนวิธี การข้ามเฟรม สำหรับเฟรมใดที่ผลลัพธ์ใกล้เคียงกันมาก จะทำการข้ามการต่อเติมภาพในเฟรมนั้นไปโดยใช้คำตอบจากเฟรมก่อนหน้าแทนเพื่อลดเวลาการประมวลผล

---

**Algorithm 11:** skip frame technique

---

```

 $v \leftarrow removeS3(u, D, v)$ 
    |  $s =$  ค่า SSIM ระหว่าง  $u$  และ  $v$  บริเวณนอกโดเมนต่อเติม
    | if  $s > 0.95$  then
    |   | คัดลอกบริเวณในโดเมนต่อเติมจาก  $u$  ไปยัง  $v$ 
    | else
    |   |  $v = MRSBC(u, z, \lambda, \theta, N_{gs}, N_0, N_1, N_2, \varepsilon, 1, m)$ 
    | end
end

```

---

ขั้นตอนวิธี การข้ามและยืมเฟรม คือขั้นตอนวิธี 10 และขั้นตอนวิธี 11 ที่นำมาประยุกต์ใช้งานร่วมกัน

---

**Algorithm 12:** skip and borrow frame technique

---

```

 $v \leftarrow removeS4(u, D, v)$ 
    |  $s =$  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  $z$ 
    |   |  $v = MRSBC(u, z, \lambda, \theta, N_{gs}, N_0, N_1, N_2, \varepsilon, 1, m)$ 
    | else
    |   |  $v = MRSBC(u, z, \lambda, \theta, N_{gs}, N_0, N_1, N_2, \varepsilon, 1, m)$ 
    | end
end

```

---

ซึ่งผลลัพธ์เปรียบเทียบระหว่างแบบใช้วิธี ยืมเฟรม ใช้วิธีข้ามเฟรม และวิธีข้ามและยืมเฟรม ได้ผลดังตาราง

วิธีการ	วิดีโอ	เวลาประมาณ (วินาที)	PSNR (dB)	SSIM
สปริทเบรกแม่น และพีระมิดรูปภาพ (ขั้นตอนวิธี 5)	1	130.03	32.19	0.9528
	2	135.17	29.98	0.9488
	3	142.11	30.54	0.9485
	4	151.42	30.79	0.9494
	5	147.70	33.48	0.9556
ยีมเฟรม (ขั้นตอนวิธี 10)	1	127.77	33.13	0.9701
	2	137.54	30.21	0.9590
	3	124.71	31.43	0.9620
	4	136.71	31.66	0.9614
	5	137.16	34.56	0.9748
ข้ามเฟรม (ขั้นตอนวิธี 11)	1	104.55	27.10	0.9429
	2	78.07	27.17	0.9351
	3	73.35	29.21	0.9393
	4	116.20	29.91	0.9423
	5	74.28	31.95	0.9442
ข้ามและยีมเฟรม (ขั้นตอนวิธี 12)	1	68.11	27.24	0.9424
	2	73.91	27.22	0.9386
	3	77.34	29.36	0.9437
	4	81.98	30.35	0.9483
	5	77.45	32.46	0.9540

ตารางที่ 14: แสดงผลลัพธ์ของขั้นตอนวิธี 5, 10, 11 และ 12

วิธีการ	เวลาประมาณ (วินาที)	PSNR (dB)	SSIM
สปริทเบรกแม่นและพีระมิดรูปภาพ	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 15: แสดงผลลัพธ์เฉลี่ยของขั้นตอนวิธี 5, 10, 11 และ 12

จากนั้นทำการทดสอบการต่อเดิมวิดีโอทั้ง 5 โดยวิธีที่คิดค้นขึ้นใช้วิธีการสปริทเบรกแม่นพร้อมทั้งการใช้พีระมิดรูปภาพที่มีการทำซ้ำแต่ละชั้นเป็น 10/3/3/10 พร้อมทั้งใช้การข้ามเฟรมและยีมเฟรม ได้ผลลัพธ์ออกเป็นดังตารางนี้

วิธีการ	เวลาประมาณ (วินาที)	PSNR (dB)	SSIM
สปริทเบรกแม่น วิธีการที่พัฒนาขึ้น	*	*	*
	75.76	29.33	0.9454

ตารางที่ 16: แสดงผลลัพธ์เฉลี่ยของการลบบทบรรยายจากอนิเมะ

สำหรับวิธีสปอร์ทเบรกเม็น เนื่องจากใช้เวลา 1 ชั่วโมงแล้วยังประมาณผลวิดีโอชุดทดสอบไม่เสร็จ ทางผู้พัฒนาจึงตัดสินใจยกการทดลอง เนื่องจากอาจต้องใช้เวลาการประมาณผลเป็นเวลาหลายชั่วโมงสำหรับวิดีโอด้วยความยาว 1 นาที สรุนวิธีที่คิดค้นขึ้น พบว่าสำหรับวิดีโอด้วยความยาว 1 นาที สามารถทำงานได้เสร็จอย่างรวดเร็ว โดยใช้เวลาเพียง 75 วินาที

## 4 แผนการดำเนินงานวิจัย

แผนการดำเนินงานตลอดทั้งโครงการสามารถสรุปได้โดยย่อจากตารางต่อไปนี้

แผนการดำเนินงาน	เดือนที่											
	1	2	3	4	5	6	7	8	9	10	11	12
ศึกษาดูแบบและขั้นตอนวิธีการต่อเติมภาพที่ใช้การแปรผันรวมในเชิงลึก พัฒนาขั้นตอนวิธีสำหรับการต่อเติมภาพที่ใช้การแปรผันรวมชนิดใหม่ ทดสอบขั้นตอนวิธีการต่อเติมภาพที่พัฒนาขึ้นโดยโปรแกรมคอมพิวเตอร์บนภาษาสั้นเคราะห์ที่และภาษาจีน อภิปรายผลที่ได้จากการทดลองเชิงตัวเลข สรุปผลการดำเนินงานวิจัยและจัดทำรูปถ่ายบันทึก	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.