บทที่ 3

ขั้นตอนวิธีเชิงตัวเลขชนิดใหม่

3.1 การนำเสนอภาพ

3.1.1 การนำเสนอภาพเฉดเทา

สำหรับภาพถ่ายสามารถพิจารณาภาพเป็นฟังก์ชันได้ดังนี้

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

เป็นฟังก์ชันต่อเนื่อง โดยที่ $\mathsf{x} = (x, y) \in \Omega$ แทนพิกัดทางกายภาพ (physical position) ของภาพ $u(\mathsf{x}) \in V$ แทนระดับความเข้มของภาพ (image intensity) ที่ x และ Ω แทนโดเมนของภาพ ซึ่งในที่นี้สามารถสมมติได้โดยไม่เสียหลักการสำคัญว่า $\Omega = [1, n]^2$ และ $V = [0, 1]$ เมื่อ $n > 1$ เป็นจำนวนเต็มบวก และโดเมนของภาพเป็นรูปสี่เหลี่ยม ทั้งนี้จะเรียกภาพ u ที่นิยามข้างต้นว่าภาพเฉดเทา (grayscale image)

	1	2	3	4	5	6	7	8
8	0.2	0.2	0.2	0.2	0.3	0.4	0.4	0.4
7	0.2	0.2	0.2	0.4	0.4	0.4	0.4	0.4
6	0.2	0.2	0.4	0.4	0.4	0.4	0.5	0.6
5	0.2	0.3	0.4	0.4	0.4	0.6	0.6	0.6
4	0.2	0.4	0.4	0.4	0.6	0.6	0.5	0.6
3	0.3	0.4	0.4	0.5	0.6	0.5	0.7	0.8
2	0.4	0.4	0.4	0.6	0.6	0.6	0.8	0.8
1	0.4	0.4	0.5	0.6	0.6	0.7	0.8	0.8

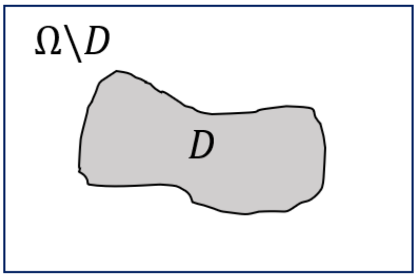
รูปที่ 3.1.1: ตัวอย่างภาพเฉดเทาที่แสดงระดับความเข้มของภาพในแต่ละระดับ

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

3.1.2 การต่อเติมภาพเฉดเทา

ในโครงงานวิจัยชิ้นนี้สำหรับการต่อเติมภาพ จะทำการหาคำตอบของภาพที่อยู่ในโดเมนต่อเติมภาพ ซึ่งจะของกำหนดตัวแปรต่างๆ ดังนี้

ให้ $\Omega \subset \mathbb{R}^2$ แทนโดเมนภาพ (image domain) $D \subset \mathbb{R}^2$ แทนโดเมนต่อเติม (ดูรูปที่ 3.1.2) และ $V \subset [0, \infty)$ และให้ $u : \Omega \to V,\ z : \Omega \to V$ แทนภาพที่ได้รับการซ่อมแซมและภาพที่ต้องการซ่อมแซมตามลำดับ



รูปที่ 3.1.2: D แทนโดเมนต่อเติม

การต่อเติมภาพเฉดเทาจะเป็นการหาคำตอบของพื้นที่ได้รับความเสียหายที่อยู่บนภาพ z ซึ่งเป็นบริเวณในโดเมนต่อเติม D โดยใช้ข้อมูลที่มีอยู่ใน $\Omega \backslash D$ เพื่อหาข้อมูลใน D ที่ได้รับความเสียหายเป็นคำตอบในภาพ u

3.1.3 การนำเสนอภาพสี

ต่อไปจะพิจารณาภาพสีในระบบ RGB นั่นคือ จะพิจารณาว่าภาพ $\boldsymbol{u}$ ประกอบด้วยสีด้วยกันทั้งสิ้น 3 คือ แดง, เขียว และ น้ำเงิน จึงเขียนภาพ $\boldsymbol{u}$ ในรูปแบบของเวคเตอร์ได้ดังนี้

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

เมื่อ $u_1, u_2, u_3 : \Omega \to V$ แทนภาพในเฉดสีแดง สีเขียว และสีน้ำเงินของ $\boldsymbol{u}$ ซึ่งการต่อเติมภาพสีที่พูดถึงในโครงงานวิจัยนี้จะทำการแยกแต่ละเฉดสีออกเป็นเฉดเทา 3 ระนาบ แล้วจึงใช้การต่อเติมภาพเฉดเทากับทั้ง 3 เฉดสีก่อนรวมกลับเป็นภาพสีอีกครั้ง

3.2 ตัวแบบเชิงแปรผันสำหรับต่อเติมภาพเฉดเทา

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

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

เมื่อ

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

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

โดยแคลคูลัสของการแปรผัน (Calculus of variations) จะได้สมการออยเลอร์ลากรางจ์ที่เกี่ยวข้องกับ (3.2.1) เป็น

$$\begin{cases} -\nabla \cdot \left(\dfrac{\nabla u}{|\nabla u|}\right) + \lambda(u-z) = 0, & \mathsf{x} \in (1,n)^2 \\ \dfrac{\partial u}{\partial \boldsymbol{n}} = 0, & x \in \partial\Omega \end{cases} \tag{3.2.3}$$

เมื่อ $\boldsymbol{n}$ แทนเวกเตอร์หน่วยที่ตั้งฉากกับของของภาพ

3.3 ตัวแบบเชิงแปรผันสำหรับการต่อเติมภาพสี

ต่อไปเราจะพิจารณาภาพสีในระบบสี RGB นั่นคือ เราสมมติว่า

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

เมื่อ $u_1, u_2, u_3 : \Omega \to V$ และ $z_1, z_2, z_3 : \Omega \to V$ แทนภาพในเฉดสีแดง สีเขียว และสีน้ำเงินของ $\boldsymbol{u}, \boldsymbol{z}$ ตามลำดับ

ในทำนองเดียวกันกับตัวแบบการต่อเติมภาพเฉดสีเทาที่ใช้การแปรผันรวม ตัวแบบการต่อเติมภาพสีที่ใช้การแปรผันรวมสามารถเขียนได้ดังนี้

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

เมื่อ

$$\bar{\mathcal{D}}(\boldsymbol{u},\boldsymbol{z}) = \frac{1}{2}\int_\Omega \lambda(u_1 - z_1)^2 d\Omega + \frac{1}{2}\int_\Omega \lambda(u_2 - z_2)^2 d\Omega + \frac{1}{2}\int_\Omega \lambda(u_3 - z_3)^2 d\Omega$$

และ

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

3.4 ขั้นตอนวิธีเชิงตัวเลขพื้นฐาน

3.4.1 การดิสครีไทซ์เซชันแบบไฟไนต์ดิฟเฟอเรนจ์

ไฟไนต์ดิฟเฟอร์เรนจ์ (Finite Difference) คือวิธีการสำหรับการประมาณค่าอนุพันธ์เมื่อใช้วิธีเชิงตัวเลข ซึ่งในโครงงานวิจัยชิ้นนี้จะมีตัวดำเนินการที่เกี่ยวข้องกับอนุพันธ์ด้วยกัน 3 ตัวได้แก่ แกรเดียน ไดเวอร์เจน และ ลาปาเซียนซึ่งสามารถทำการหาได้ดังนี้

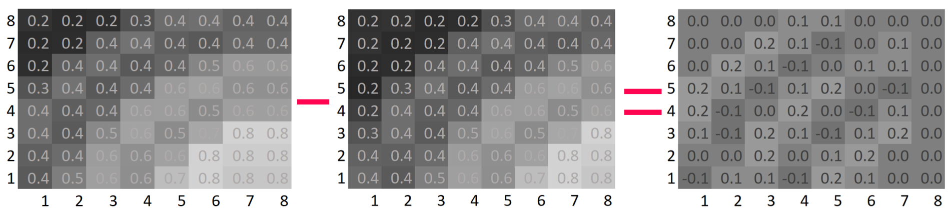
การหาอนุพันธ์

ทั้ง แกรเดียน ไดเวอร์เจน และ ลาปาเซียน ล้วนมีพื้นฐานมาจากการหาค่าอนุพันธ์ในโครงงานวิจัยนี้จะใช้วิธีการฟอร์เวิร์ดดิฟเฟอร์เรนจ์ (Forward Difference) และใช้เงื่อนไขค่าขอบแบบนิวแมน (neumann boundary condition)

นั่นคือการหาอนุพันน์ของค่าความเข้มที่พิกัดทางกายภาพเป็น (i,j)

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

เมื่อระบบกริดที่ใช้มีความห่างเพียงหนึ่งหน่วย จึงได้ว่า $h = 1$ ทั้งนี้ระยะห่าง h อาจเปลี่ยนไปตามชั้นของพีระมิดรูปภาพ



รูปที่ 3.4.1: ตัวอย่างการหาอนุพันธ์บนภาพเฉดเทา

จากภาพ 3.1.1 เมื่อต้องการหาอนุพันธ์เทียบแกน x จะทำตามภาพที่ 3.4.1โดยทำการสร้างภาพซึ่งทำการตัดขอบทางซ้ายออกหนึ่งคอลัมม์และเพิ่มขอบทางขวาหนึ่งคอลัมม์โดยใช้เงื่อนไขค่าขอบแบบนิวแมน จากนั้นภาพที่สร้างขึ้นไปลบกับภาพเดิมจะได้อนุพันธ์ของภาพนั้นดังที่ปรากฏทางขวา ทั้งนี้หาก $h \neq 1$ สามารถทำการหารภาพผลลัพธ์ด้วยค่า h ได้เพื่อให้ได้ค่าที่ต้องการ

การหาแกรเดียน

สำหรับการหาแกรเดียน (Gradient) จะใช้การหาอนุพันธ์โดยวิธีฟอร์เวิร์ดดิฟเฟอร์เรนจ์ดังที่กล่าวไปในหัวข้อก่อนหน้า ั้งในแนวแกน x และแนวแกน y คำตอบที่ได้จะเป็นเวคเตอร์ของอนุพันธ์แนวแกน x และอนุพันธ์แนวแกน y ได้เวคเตอร์ดังนี้

$$\nabla \vec{v_{u_i}} = (\frac{d}{dx}u_{i,j}, \frac{d}{dy}u_{i,j})^\top$$

การหาไดเวอร์เจน

สำหรับไดเวอร์เจน (Divergence) จะเป็นการหาผลรวมของอนุพันธ์ในแต่ละแกนของเวคเตอร์ด้วยวิธีฟอร์เวิร์ดดิฟเฟอร์เรนจ์ นั่นคือ

$$\nabla \cdot (\vec{v_{i,j}}) = \frac{\partial d}{\partial x}\vec{v_{i,j}}_x + \frac{\partial d}{\partial y}\vec{v_{i,j}}_y$$

การหาลาปาเซียน

สำหรับลาปาเซียน (Lapacian) นั่นคือการทำหาไดเวอร์เจรบนเวคเตอร์ที่หาแกรเดียนแล้ว แต่ทั้งนี้สามารถหาลาปาเซียนได้จาก

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

3.4.2 ขั้นตอนวิธีเดินเวลา (explicit time marching method)

คณะวิจัย [2] ได้แนะนำวิธีการเชิงตัวเลขสำหรับการกำจัดสัญญาณรบกวนโดยใช้วิธีการเดินเวลาแบบชัดแจ้ง ซึ่งสามารถประยุกต์เป็นวิธีเชิงตัวเลขสำหรับการต่อเติมภาพได้ดังนี้

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

$$u(\mathsf{x}, t_{k+1}) = u(\mathsf{x}, t_k) + \tau \left(\nabla \cdot \left(\frac{\nabla u(\mathsf{x}, t_k)}{|\nabla u(\mathsf{x}, t_k)|} \right) + \lambda(\mathsf{x})(u(\mathsf{x}, t_k) - z(\mathsf{x})) \right), \; u(\mathsf{x}, t_0) = z \tag{3.4.1}$$

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

วิธีเดินเวลาแบบชัดแจ้งสำหรับภาพเฉดเทามีขั้นตอนวิธีดังนี้

Algorithm 1: วิธีการเดินเวลาแบบชัดแจ้งสำหรับการต่อเติมภาพที่ใช้การแปรผันรวม

Input:

u คือรูปภาพที่ต้องการต่อเติม

λ คือพารามิเตอร์เร็กกิวลาร์ไรเซชัน ที่ได้กล่าวถึงในสมการ (3.2.2)

β เป็นจำนวนจริงบวกที่ใช้เพื่อหลีกเลี่ยงการหารด้วยศูนย์

τ เป็นจำนวนจริงบวกที่เป็นตัวแปรเดินเวลา

N เป็นจำนวนเต็มบวกสำหรับกำหนดจำนวนรอบที่ทำงาน

ε เป็นจำนวนจริงบวกของค่าความคลาดเคลื่อนสัมพัทธ์

Output: รูปภาพที่ผ่านการต่อเติมแล้ว

$u \longleftarrow EplicitTimeMarching(u, \lambda, \beta, \tau, N, \varepsilon)$

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

3.4.3 วิธีการทำซ้ำแบบจุดตรึง (fixed-point iteration method)

คณะวิจัย [3] ได้แนะนำวิธีการเชิงตัวเลขสำหรับการกำจัดสัญญาณรบกวนโดยใช้วิธีการทำซ้ำแบบจุดตรึง ซึ่งสามารถประยุกต์เป็นวิธีเชิงตัวเลขสำหรับการต่อเติมภาพได้ดังนี้

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

$$-\nabla \cdot \left(\frac{\nabla u^{[\nu+1]}}{|\nabla u|^{[v]}} \right) + \lambda(u^{[\nu+1]} - z) = 0,\ u^{[0]} = z \quad (3.4.2)$$

เนื่องจาก $\frac{1}{|\nabla u|} = \frac{1}{\sqrt{u_x^2+u_y^2}} \to \infty$ ในบริเวณที่ u มีความเข้มสีเป็นเอกพันธุ์ ($u(\mathbf{x}) =$ ค่าคงตัว) เพื่อหลีกเลี่ยงปัญหาเชิงตัวเลขจะเกิดขึ้นใน (3.4.1) และ (3.4.2) เราจะใช้

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

วิธีการทำซ้ำแบบจุดตรึงมีขั้นตอนดังนี้

Algorithm 2: วิธีการทำซ้ำจุดตรึงสำหรับการต่อเติมภาพที่ใช้การแปรผันรวม

Input:

u คือรูปภาพที่ต้องการต่อเติม

λ คือพารามิเตอร์เร็กกิวลาร์ไรเซชัน ที่ได้กล่าวถึงในสมการ (3.2.2)

β เป็นจำนวนจริงบวกที่ใช้เพื่อหลีกเลี่ยงการหารด้วยศูนย์

N เป็นจำนวนเต็มบวกสำหรับกำหนดจำนวนรอบที่ทำงาน

ε เป็นจำนวนจริงบวกของค่าความคลาดเคลื่อนสัมพัทธ์

Output: รูปภาพที่ผ่านการต่อเติมแล้ว

$u \longleftarrow FixedPoint(u, z, \lambda, \beta, N, \varepsilon)$

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

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

$u^{old} = u$

$u = GaussSeidel(u, z, \lambda, \beta, N_{gs})$

$err = \frac{||u-u^{old}||}{||u||}$

$i = i + 1$

end

Algorithm 3: การทำซ้ำเกาส์-ไซเดล สำหรับวิธีการจุดตรึง

Input:

u คือรูปภาพที่ต้องการต่อเติม

λ คือพารามิเตอร์เร็กกิวลาร์ไรเซชัน ที่ได้กล่างถึงในสมการ (3.2.2)

β เป็นจำนวนจริงบวกที่ใช้เพื่อหลีกเลี่ยงการหารด้วยศูนย์

N เป็นจำนวนเต็มบวกสำหรับกำหนดจำนวนรอบที่ทำงาน

Output: รูปภาพที่ผ่านการทำเกาส์-ไซเดลแล้ว

$u \longleftarrow GaussSeidel(u, \lambda, \beta, N_{gs})$

initialize $k = 0$

$D(u)_{i,j} = \frac{1}{\sqrt{u_x^2+u_y^2+\beta}}, 1 \leq i \leq n_x, 1 \leq j \leq n_y$

while $k < N_{gs}$ **do**

$u_{i,j}^{k+1} = \frac{\lambda_{i,j}z_{i,j}+(D_{i,j}(u_{i+1,j}^k+u_{i,j+1}^k)+D_{i-1,j}u_{i-1,j}^{k+1}+D_{i,j-1}u_{i,j-1}^{k+1})}{\lambda_{i,j}+(2D_{i,j}+D_{i-1,j}+D_{i,j-1})}$

$k = k + 1$

end

จาก (3.4.1) และ (3.4.2) เราพบว่ายิ่ง β มีค่าน้อยลงมากขึ้นเท่าไหร่ ความแม่นยำของตัวแบบ (3.2.1) ยิ่งมีมากขึ้นเท่านั้น นอกจากนี้ เรายังพบอีกว่า การแก้สมการ (3.4.1) และ (3.4.2) ยิ่งมีความยุ่งยากมากขึ้นสำหรับ β ที่มีค่าน้อยๆ

เพื่อเอาชนะความยากเชิงตัวเลขนี้ คณะวิจัยโดย [4] ได้แนะนำวิธีการสปริทเบรกแมนซึ่งสามารถกล่าวถึงพอสังเขป ดังนี้

3.4.4 วิธีการสปริทเบรกแมน (Split Bregman method)

เริ่มจากการแนะนำเวกเตอร์เสริม $\boldsymbol{w}$ พารามิเตอร์เบรกแมน (Bregman parameter) $\boldsymbol{b}$ และพารามิเตอร์เพนัลที (panalty parameter) $\theta > 0$ และเขียน (3.2.1) ใหม่ ดังนี้

$$\min_{u,\boldsymbol{w}}\{\mathcal{J}(u,\boldsymbol{w}) = \frac{1}{2}\int_\Omega \lambda(u-z)^2 d\Omega + \int_\Omega |\boldsymbol{w}| d\Omega + \frac{\theta}{2}\int_\Omega (\boldsymbol{w} - \nabla u + \boldsymbol{b}) d\Omega\} \qquad (3.4.3)$$

สำหรับการหาคำตอบของ (3.4.3) เราจะใช้วิธีการหาค่าต่ำที่สุดแบบสลับ (alternating minimization method) โดยเริ่มจากการตรึง $\boldsymbol{w}^{\text{old}}$ และ $\boldsymbol{b}^{\text{old}}$ จากนั้นแก้ปัญหาย่อยสำหรับ u

$$u^{\text{New}} = \arg\min_{u}\{\mathcal{J}_1(u) = \frac{1}{2}\int_{\Omega}\lambda(u-z)^2 d\Omega + \frac{\theta}{2}\int_{\Omega}(\boldsymbol{w}^{\text{old}} - \nabla u + \boldsymbol{b}^{\text{old}})d\Omega\} \tag{3.4.4}$$

ต่อไปใช้ u^{New} ที่ได้จากการแก้ปัญหาย่อย (3.4.4) เพื่อแก้ปัญหาย่อยสำหรับ $\boldsymbol{w}$

$$\boldsymbol{w}^{\text{New}} = \arg\min_{\boldsymbol{w}}\{\mathcal{J}_2(\boldsymbol{w}) = \int_{\Omega}|\boldsymbol{w}|d\Omega + \frac{\theta}{2}\int_{\Omega}(\boldsymbol{w} - \nabla u^{\text{New}} + \boldsymbol{b}^{\text{old}})d\Omega\} \tag{3.4.5}$$

สุดท้ายจึงปรับปรุงพารามิเตอร์เบรกแมนโดย

$$\boldsymbol{b}^{\text{New}} = \boldsymbol{b}^{\text{old}} + \nabla u^{\text{New}} - \boldsymbol{w}^{\text{New}} \tag{3.4.6}$$

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

วิธีการสปริทเบรกแมนมีขั้นตอนวิธีดังนี้

Algorithm 4: วิธีสปริทเบรกแมนสำหรับการต่อเติมภาพที่ใช้การแปรผันรวม

Input:

u คือรูปภาพที่ต้องการต่อเติม

λ คือพารามิเตอร์เร็กกิวลาร์ไรเซชัน ที่ได้กล่าวถึงในสมการ (3.2.2)

θ คือพารามิเตอร์เพนัลที ซึ่งเป็นจำนวนจริงบวก

N_{gs} เป็นจำนวนเต็มบวกสำหรับกำหนดจำนวนรอบที่ทำงานของการทำเกาส์-ไซเดล

N เป็นจำนวนเต็มบวกสำหรับกำหนดจำนวนรอบที่ทำงานของสปริทเบรกแมน

ε เป็นจำนวนจริงบวกของค่าความคลาดเคลื่อนสัมพัทธ์

Output: รูปภาพที่ผ่านการต่อเติมแล้ว

$u \longleftarrow SplitBregman(u, \lambda, \theta, N_{gs}, N, \varepsilon)$

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

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

$u^{old} = u$; $w^{old} = w$; $b^{old} = b$;

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

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

$b = b^{old} + \nabla u - \boldsymbol{w}$

$err = \frac{||u-u^{old}||}{||u||}$

$i = i + 1$

end

หมายเหตุ:

(1) ผลเฉลยของ $u = \arg\min_u \mathcal{J}_1(u)$ กำหนดโดยการแก้ปัญหาผลเฉลยของ

$$-\theta\triangle u + \lambda u = \lambda z - \theta\nabla\cdot(\boldsymbol{w} - \boldsymbol{b})$$

โดยใช้วิธีการไฟไนต์ดิฟเฟอเรนจ์และวิธีการเกาส์-ไซเดลจำนวน N_{gs} รอบ

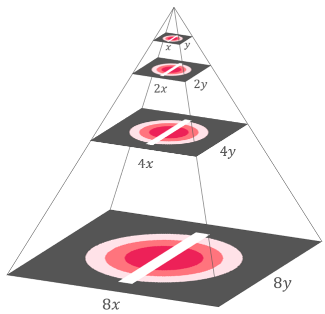
(2) ผลเฉลยของ $\boldsymbol{w} = \arg\min_{\boldsymbol{w}} \mathcal{J}_2(\boldsymbol{w})$ กำหนดโดย

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

3.5 ขั้นตอนวิธีเชิงตัวเลขที่นำเสนอ

3.5.1 ขั้นตอนวิธีเชิงตัวเลขสำหรับต่อเติมภาพศิลปะ

สำหรับวิธีการซ่อมแซมภาพสศิลปะไทย จะใช้วิธีการสปริทเบรกแมนเพื่อหลีกเลี่ยงปัญหาเชิงตัวเลขที่จะเกิดขึ้น แต่เพื่อให้วิธีการสปริทเบรกแมนประมวลผลภาพได้รวดเร็วขึ้น ผู้วิจัยได้พัฒนากระบวนการกำหนดคำตอบเริ่มต้น โดยวิธีการ มัลติรีโซลูชัน (multi-resolution method) หรือวิธีการพีระมิดรูปภาพ (pyramid method) [5] เริ่มจากการย่อขนาดรูปลงครึ่งนึงโดยใช้วิธี Bilinear Interpolation จนกระทั่งถึงระดับความคมชัดที่ต้องการ จากนั้นทำการต่อเติมภาพขนาดเล็ก และนำผลลัพธ์ที่ได้จากภาพขนาดเล็กทำการขยายภาพขึ้นสองเท่าโดยใช้ Bilinear Interpolation เป็นคำตอบเริ่มต้นสำหรับการต่อเติมภาพในชั้นถัดไป



รูปที่ 3.5.1: วิธีการพีระมิดรูปภาพ

ขั้นตอนวิธีสำหรับการทำพีระมิดรูปภาพสำหรับการต่อเติมภาพแบบสปริทเบรกแมนเพื่อให้ประมวลผลได้เร็วขึ้นนั้นสามารถสรุปได้ดังนี้

Algorithm 5: วิธีสปริทเบรกแมนที่ใช้พีระมิดรูปภาพ

Input:

u คือรูปภาพที่ต้องการต่อเติม

λ คือพารามิเตอร์เร็กกิวลาร์ไรเซชัน ที่ได้กล่าวถึงในสมการ (3.2.2)

θ คือพารามิเตอร์เพนัลที ซึ่งเป็นจำนวนจริงบวก

N_{gs} เป็นจำนวนเต็มบวกสำหรับกำหนดจำนวนรอบที่ทำงานของการทำเกาส์-ไซเดล

c ตัวแปรช่วยสำหรับบอกความลึก ให้กำหนดเป็น 1

m คือ ระดับความลึกของพีระมิดรูปภาพ เป็นจำนวนเต็มบวก

N_0 จำนวนรอบการทำสปริทเบรกแมนที่ชั้นละเอียดสุด

N_1 จำนวนรอบการทำสปริทเบรกแมนที่ชั้นต่างๆ

N_2 ตำนวนรอบการทำสปริทเบรกแมนที่ชั้นหยาบสุด

Output: รูปภาพที่ผ่านการต่อเติมแล้ว

$u \longleftarrow MultiSplitBregmanColor(\boldsymbol{u}, \lambda, \theta, N_{gs}, N_0, N_1, N_2, \varepsilon, c, m)$

Initialize $height =$ ความสูงของภาพ $\boldsymbol{u}$, $width =$ ความกว้างของภาพ $\boldsymbol{u}$

if $c < m$ **then**

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

$y = Bilinear(\lambda, \lfloor width * 0.5 \rfloor, \lfloor height * 0.5 \rfloor)$

$r = MRSBC(\boldsymbol{x}, \boldsymbol{z}, y, \lambda, \theta,$

$N_{gs}, N_0, N_1, N_2, \varepsilon, c+1, m)$

$\boldsymbol{u} = Bilinear(r, width, height)$

end

if $c = 1$ **then**

$N_{SB} = N_0$

else if $c = m$ **then**

$N_{SB} = N_2$

else

$N_{SB} = N_1$

end

$u = SplitBregmanColor(\boldsymbol{u}, \lambda, \theta, N_{gs}, N_{SB}, \varepsilon)$

Algorithm 6: Bilinear Interpolation

$J \longleftarrow Bilinear(I, x, y)$

Initialize $v =$ ความสูงของภาพ I, w คือความกว้างของภาพ I,

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

$r' = 1, 2, ...x, c' = 1, 2, ..., y,$

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

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

$\triangle r = r_f - r$

$\triangle c = c_f - c$

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

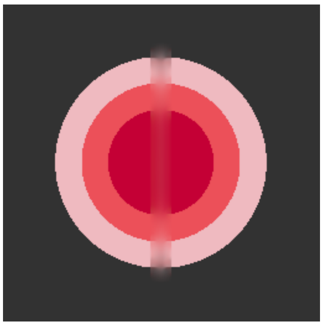
$+I(r + 1, c) \cdot \triangle r \cdot (1 - \triangle c)$

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

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

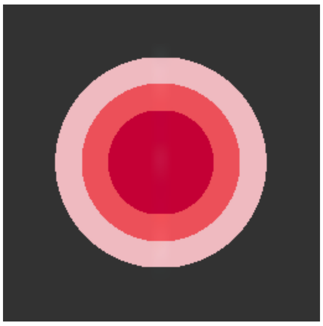
นอกจากนี้แล้ว ผู้วิจัยยังได้สังเกตอีกว่า การทำซ้ำนั้นจะลู่เข้าเร็วในช่วงแรก จากนั้นความเร็วในการลู่เข้าจะลดลง ซึ่งทำให้การทำซ้ำเพียงไม่กี่ครั้งในระดับความคมชัดเดิม มีผลการซ่อมแซมภาพจนแสดงความคล้ายคลึงกับภาพต้นฉบับได้

(a) 5 ครั้ง



(b) 10 ครั้ง

(c) 50 ครั้ง



(d) 100 ครั้ง

รูปที่ 3.5.2: พีระมิดที่ลำดับการทำซ้ำเป็น 10/10/10 และที่ระดับความคมชัดละเอียดสุดใช้จำนวนการทำซ้ำที่ต่างกัน

ผู้วิจัยจึงกำหนดให้การทำซ้ำในระดับความละเอียดสูงสุดเท่ากับ 10 ครั้ง เพราะการทำซ้ำในระดับความละเอียดสูงสุดทำให้ทำงานได้ช้า อีกทั้งการทำซ้ำในชั้นที่รูปภาพมีขนาดเล็กจำนวนมาก ไม่ช่วยให้การประมวลผลได้เร็วขึ้น ผู้วิจัยจึงเลือกใช้การทำซ้ำแบบ 10/3/3/10 ในการต่อเติมภาพ

3.5.2 ขั้นตอนวิธีเชิงตัวเลขสำหรับซ่อมแซมภาพวิดีโอ

เนื่องจากไฟล์วิดีโอนั้นประกอบด้วยชุดของภาพหลายภาพ กล่าวคือ $V = \{\boldsymbol{u}_i | i = 1, 2, 3 \ldots N_f\}$ ทำให้ขั้นตอนวิธีการลบคำบรรยายออกจากวิดีโอ จะต้องทำการต่อเติมบริเวณที่เป็นบทบรรยายทีละภาพ ดังที่แสดงในขั้นตอนวิธีต่อไปนี้

Algorithm 7: วิธีการลบบทบรรยายจากวิดีโอ

$V \longleftarrow SubtitleRemove(V)$

for $i = 1, 2, ...N_f$ **do**

- หาโดเมนต่อเติม D จากเฟรม $\boldsymbol{u}_i$ ซึ่งเป็นภาพที่ i ของวิดีโอ V
- ต่อเติมเฟรม $\boldsymbol{u}_i$ โดยใช้โดเมนต่อเติม D

end

โดยขั้นตอนการต่อเติมภาพ $\boldsymbol{u}_i$ ด้วยโดเมนต่อเติม D นั้นจะสามารถใช้วิธีการเดียวกับการซ่อมแซมภาพศิลปะไทยได้ ส่วนการหาโดเมนต่อเติมซึ่งเป็นบทบรรยายอนิเมะ จะกล่าวถึงในหัวข้อย่อยถัดไป

3.5.3 การหาบทบรรยายบนอนิเมะ

ก่อนจะลบบทบรรยายนั้น จำเป็นต้องหาบทบรรยายในภาพให้ได้เสียก่อน โดยบทบรรยายของอนิเมะนั้น มักจะใช้ขอบของตัวอักษรเป็นสีดำ อีกทั้งบทบรรยายนั้นจะลอยห่างออกมาจากขอบของวิดีโอ และขนาดของคำบรรยายนั้นจะมีขนาดอยู่ประมาณหนึ่งไม่ใหญ่หรือไม่เล็กเกินไป ด้วยสมบัตินี้เองทำให้จึงสามารถหาบริเวณบนเฟรมที่เป็นบทบรรยายได้โดยจะมีวิธีหาพื้นที่ซึ่งเป็นบทบรรยายดังนี้

Algorithm 8: Finding subtitle

$D \longleftarrow findsub(\boldsymbol{u})$

- ทำการเปลี่ยนสีดำในภาพ $\boldsymbol{u}$ ให้เป็นสีขาวแล้วเปลี่ยนอื่นๆ ให้เป็นสีดำเพื่อหาขอบของคำบรรยาย
- เปลี่ยนบริเวณสีขาวในภาพให้เป็นสีดำ และเปลี่ยนบริเวณสีดำให้เป็นสีขาว
- ทำการลบบริเวณสีขาวซึ่งติดกับขอบของภาพออกไป เนื่องจากบทบรรยายจะลอยอยู่ ไม่ติดกับขอบเสมอ
- ลบบริเวณที่ใหญ่เกินกว่าจะเป็นบทบรรยาย
- ลบบริเวณที่เล็กเกินกว่าจะเป็นบทบรรยาย
- ทำการขยายพื้นที่ๆ เป็นสีขาวขึ้นด้วยความกว้างของขอบบทบรรยาย
- สีขาวที่เหลืออยู่ในภาพจะเป็นบทบรรยาย

3.5.4 การลบคำบรรยายจากบทอนิเมะ

สำหรับอนิเมะนั้น แต่ละเฟรมจะเป็นรูปภาพ เราจึงสามารถประยุกต์ใช้วิธีการซ่อมแซมภาพจิตรกรรมไทยมาใช้ในการลบคำบรรยายได้ แต่ผู้วิจัยก็ได้สังเกตว่า สำหรับอนิเมะที่เป็นวิดีโอแล้ว ในขณะที่ประมวลผลวิดีโอ เราสามารถใช้ผลการต่อเติมภาพจากภาพที่แล้ว มาใช้เป็นคำตอบเริ่มต้นจึงได้ว่าขั้นตอนการลบบทบรรยายออกจากวิดีโอมีดังนี้

Algorithm 9: วิธีการทำงานบนวิดีโอ เมื่อต้องการผลจากภาพที่แล้วมาใช้เป็นคำตอบเริ่มต้น

$V \longleftarrow RemoveSubtitle(V)$

initialize $i = 1$

while $i < N_f - 1$ do

$\boldsymbol{u}_i$ คือเฟรมที่ i ใน V

$\boldsymbol{u}_{i+1}$ คือเฟรมที่ $i + 1$ ใน V

D คือโดเมนต่อเติมใน $\boldsymbol{u}_{i+1}$

$\boldsymbol{u}_{i+1} = RemoveByBorrowFrame(\boldsymbol{u}_i, D, \boldsymbol{u}_{i+1})$

end

$RemoveByBorrowFrame(\boldsymbol{u}_i, D, u_{i+1})$ คือขั้นตอนวิธีที่ 10 ซึ่งในทำนองเดียวกันเราสามารถเปลี่ยน $RemoveByBorrowFrame(u_i, D, \boldsymbol{u}_{i+1})$ เป็น $RemoveBySkipFrame(\boldsymbol{u}_i, D, u_{i+1})$ เพื่อใช้กับขั้นตอนวิธี 11 และเปลี่ยนเป็น $RemoveBySkipAndBorrowFrame(\boldsymbol{u}_i, D, \boldsymbol{u}_{i+1})$ เพื่อใช้กับขั้นตอนวิธี 12 ได้

ขั้นตอนวิธี การยืมเฟรม จะเป็นการนำผลลัพธ์จากเฟรมก่อนหน้ามาเป็นคำตอบในการเริ่มต้นในการประมวลผลเพื่อให้ผลลัพธ์ลู่เข้าได้เร็วขึ้น

Algorithm 10: การลบบทบรรยายโดยใช้วิธีการยืมเฟรม

$\boldsymbol{v} \longleftarrow RemoveByBorrowFrame(\boldsymbol{u}, D, \boldsymbol{v})$

$s =$ ค่า SSIM ระหว่าง $\boldsymbol{u}$ และ $\boldsymbol{v}$ บริเวณนอกโดเมนต่อเติม

if *s > 0.9* **then**

คัดลอกบริเวณในโดเมนต่อเติมจาก $\boldsymbol{u}$ ไปยัง $\boldsymbol{v}$

end

$\boldsymbol{v} = MultiSplitBregmanColor(\boldsymbol{v}, \lambda, \theta, N_{gs}, N_0, N_1, N_2, \varepsilon, 1, m)$

ขั้นตอนวิธี การข้ามเฟรม สำหรับเฟรมใดที่ผลลัพธ์ใกล้เคียงกันมาก จะทำการข้ามการต่อเติมภาพในเฟรมนั้นไปโดยใช้คำตอบจากเฟรมก่อนหน้าแทนเพื่อลดเวลาการประมวลผล

Algorithm 11: การลบบทบรรยายโดยใช้วิธีการข้ามเฟรม

$\boldsymbol{v}, \longleftarrow RemoveBySkipFrame(\boldsymbol{u}, D, \boldsymbol{v})$

$s =$ ค่า SSIM ระหว่าง $\boldsymbol{u}$ และ $\boldsymbol{v}$ บริเวณนอกโดเมนต่อเติม

if *s > 0.95* **then**

คัดลอกบริเวณในโดเมนต่อเติมจาก $\boldsymbol{u}$ ไปยัง $\boldsymbol{v}$

else

$\boldsymbol{v} = MultiSplitBregmanColor(\boldsymbol{v}, \lambda, \theta, N_{gs}, N_0, N_1, N_2, \varepsilon, 1, m)$

end

ขั้นตอนวิธี การข้ามและยืมเฟรม คือขั้นตอนวิธี 10 และขั้นตอนวิธี 11 ที่นำมาประยุกต์ใช้งานร่วมกัน

Algorithm 12: การลบบทบรรยายโดยใชวิธีการข้ามเฟรมและยืมเฟรม

$\boldsymbol{v} \longleftarrow RemoveBySkipAndBorrowFrame(\boldsymbol{u}, D, \boldsymbol{v})$

$s =$ ค่า SSIM ระหว่าง u และ v บริเวณนอกโดเมนต่อเติม

if *s > 0.95* **then**

คัดลอกบริเวณในโดเมนต่อเติมจาก $\boldsymbol{u}$ ไปยัง $\boldsymbol{v}$

else if *s > 0.9* **then**

คัดลอกบริเวณในโดเมนต่อเติมจาก $\boldsymbol{u}$ ไปยัง $\boldsymbol{v}$

$\boldsymbol{v} = MultiSplitBregmanColor(\boldsymbol{v}, \lambda, \theta, N_{gs}, N_0, N_1, N_2, \varepsilon, 1, m)$

else

$\boldsymbol{v} = MultiSplitBregmanColor(\boldsymbol{v}, \lambda, \theta, N_{gs}, N_0, N_1, N_2, \varepsilon, 1, m)$

end

จากนั้นทางผู้พัฒนาจะนำวิธีการลบบทบรรยายอนิเมะที่นำเสนอนี้ไปทดลองลบบทบรรยายเพื่อเปรียบเทียบประสิทธิภาพกับวิธีสปริทเบรกแมนเดิมที่นำมาใช้กับวิดีโอ

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.