

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

Pakkapon Phongthawee

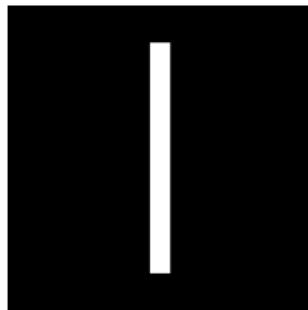
Department of Mathematics  
Silpakorn University

DPST Student Conference on Science and Technology  
June 21-23,2019

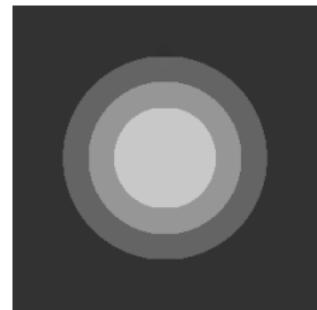
# Image inpainting



(a) Damaged image



(b) Inpainted domain



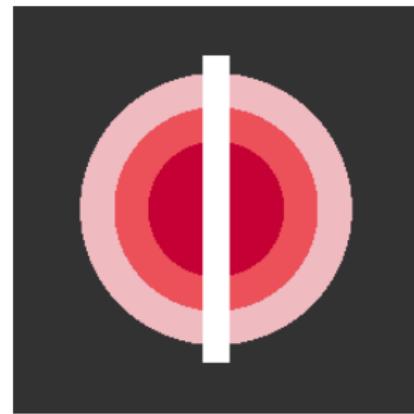
(c) Restored image

Figure 1: Example of image inpainting

# Image inpainting problem



grayscale image



color image

## grayscale image

- image domain  $\Omega \subset \mathbb{R}^2$
- inpainting domain  $D \subset \mathbb{R}^2$
- physical position  $\mathbf{x} = (x, y) \in \Omega$
- image intensity  $V \subset [0, \infty)$
- grayscale image  $u : \Omega \rightarrow V$ ,  $z : \Omega \rightarrow V$
- without loss of generality  $\Omega = [1, n]^2$  and  $V = [0, 1]$  which  $n > 0$  is positive integer

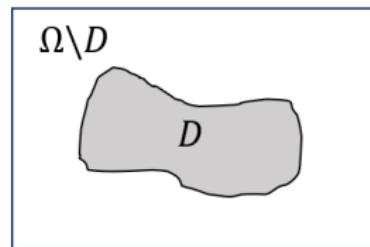


Figure 3:  $D$  is an inpainting domain

# Total variation model for grayscale image inpainting

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

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

---

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.

# Numerical algorithm

- (1) Explicit time marching
- (2) Fixed point iteration
- (3) Split Bregman

# Explicit time marching

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



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

# Explicit time marching

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

$$u(\mathbf{x}, t_0) = z \quad t_k = t_0 + k\tau \ (\tau > 0) \quad t_0 = 0$$

$$u(\mathbf{x}, t_0), u(\mathbf{x}, t_1), u(\mathbf{x}, t_2), u(\mathbf{x}, t_3), \dots, \color{red}{u(\mathbf{x}, t^*)}$$

## Limitation of explicit time marching

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

$$\tau < 1$$

# Fixed-point iteration

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

$u^{[0]}, u^{[1]}, u^{[2]}, u^{[3]}, \dots, u^*$

# Numerical problem

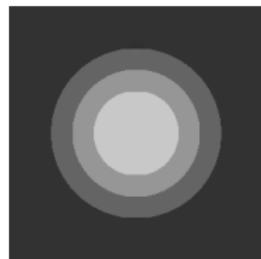


Figure 4: Example of image which has numerical problem

$$\frac{1}{|\nabla u|} = \frac{1}{\sqrt{u_x^2 + u_y^2}} \rightarrow \infty$$

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

# Split Bregman

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



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



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

---

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.

# Split Bregman

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



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

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

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

## Color image

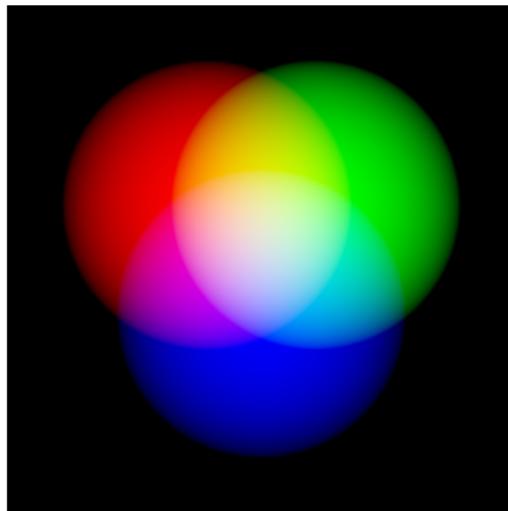


Figure 5: Color image create from red,green and blue color<sup>1</sup>

<sup>1</sup> ภาพจาก [https://commons.wikimedia.org/wiki/File:Additive\\_RGB\\_Circles+48bpp.png](https://commons.wikimedia.org/wiki/File:Additive_RGB_Circles+48bpp.png) สืบคันมืออวันที่ 25 กันยายน 2561

# Color image

$$u, z : \Omega \rightarrow V$$



$$u = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}, z = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} : \Omega \rightarrow V^3$$

# Total variation model for color image inpainting

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



$$\min_u \{ \mathcal{J}(u) = \sum_{l=1}^3 \left( \frac{1}{2} \int_{\Omega} \lambda(u_l - z_l)^2 d\Omega + \int_{\Omega} |\nabla u_l| d\Omega \right) \}$$

# Split Bregman for Color image

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



$$\begin{aligned} \min_{u, w_1, w_2, w_3} \{ \mathcal{J}(u, w_1, w_2, w_3) &= \sum_{l=1}^3 \left( \frac{1}{2} \int_{\Omega} \lambda(u_l - z_l)^2 d\Omega + \int_{\Omega} |\nabla w_l| d\Omega \right. \\ &\quad \left. + \frac{\theta}{2} \int_{\Omega} (w_l - \nabla u_l + b_l) d\Omega \right) \end{aligned}$$

# Quality measurement

- (1) Peak Signal Noise Ratio (PSNR)
- (2) Structural Similarity (SSIM)

# Peak Signal Noise Ratio (PSNR)

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

- MSE is mean square error which is  $\text{MSE} = \frac{1}{nx \times ny} \sum (u - \tilde{u})^2$
- $u$  is original image.
- $\tilde{u}$  is restored image from numerical method.
- The unit of PSNR is the decibel (dB).

# Structural Similarity (SSIM)

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

- $u$  is original image.
- $\tilde{u}$  is restored image from numerical method.
- $\mu_u$  is average of  $u$
- $\mu_{\tilde{u}}$  is average of  $\tilde{u}$
- $\sigma_u$  is variation of  $u$
- $\sigma_{\tilde{u}}$  is variation of  $\tilde{u}$
- SSIM has range between 0 to 1.

# Synthetic image



Figure 6: Original image



Figure 7: Damaged image

iteration  $\leq 10,000$  round

$$\frac{\|u_{new} - u_{old}\|}{\|u_{new}\|} \geq 10^{-4}$$

## Restoration result



Figure 8: Explicit time marching

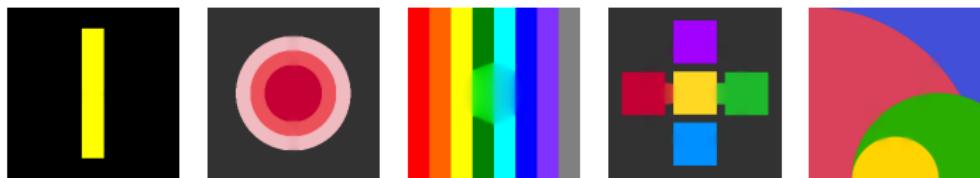


Figure 9: Fixed point iteration

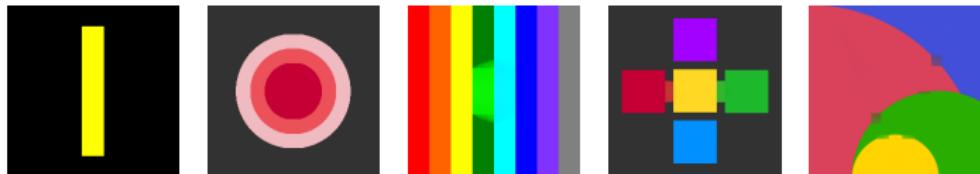


Figure 10: Split Bregman

# Performance of numerical method

Method	Processed time (Second)	PSNR (dB)	SSIM
Explicit time marching	120.68	16.72	0.9960
Fixed point iteration	74.81	38.67	0.9999
Split Bregman	14.06	39.42	0.9999

Table 1: Average image restoration result of numerical method  
 $\lambda = 250, \beta = 10^{-5}, \tau = 10^{-5}, \theta = 5$

## Intial solution

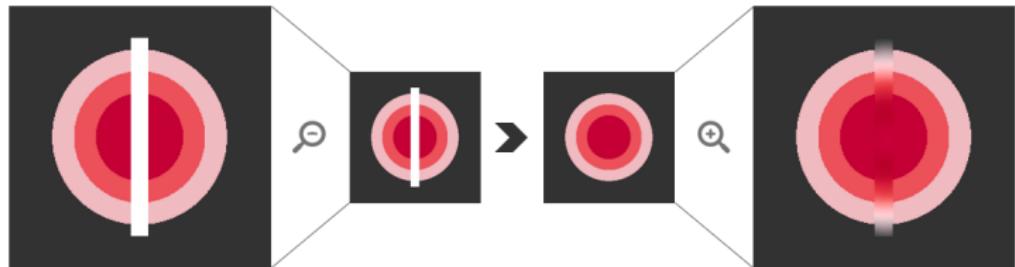


Figure 11: Find intial solution by using image pyramid

# Image pyramid

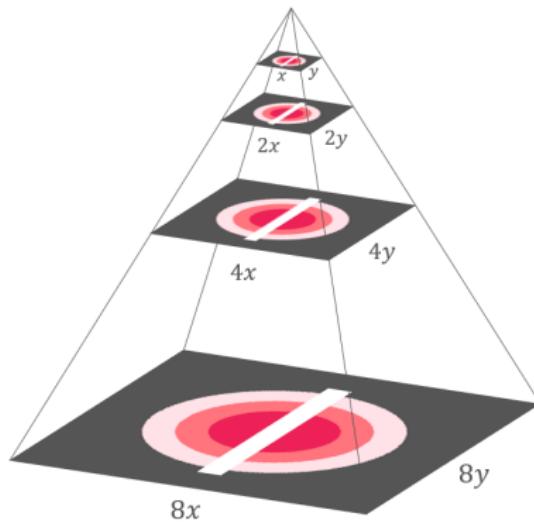


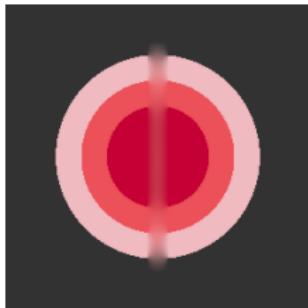
Figure 12: Image pyramid method

## Restoration result with image pyramid

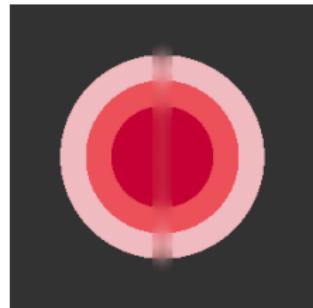
Iteration step	Processed time (Second)	PSNR (dB)	SSIM
Without Image pyramid	17.38	39.42	0.9999
10/1/1/10000	13.52	39.38	0.9999
10/3/3/10000	11.86	39.54	0.9999
10/10/10/10000	9.26	40.17	0.9999
100/1/1/10000	10.28	39.04	0.9999
100/3/3/10000	10.28	39.80	0.9999
100/10/10/10000	9.27	40.12	0.9999

Table 2: Average result of Split Bregman method with Image pyramid

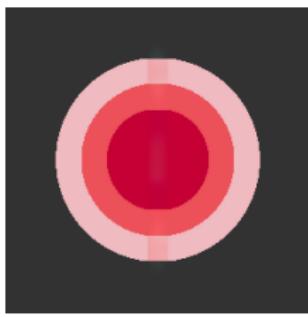
## Iteration on Finest level



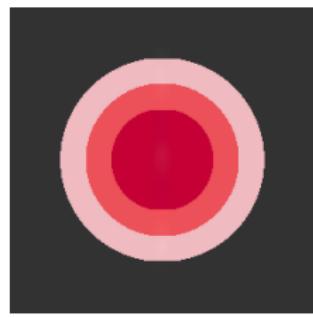
(a) 5 times



(b) 10 times



(c) 50 times



(d) 100 times

Figure 13: image pyramid with iteration 10/10/10 but have difference iteration on finest step

## Restoration result with only 10 iteration on finest step

Iteration step	Processed time (Second)	PSNR (dB)	SSIM
Without Image pyramid	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

Table 3: Average result of Resotration with only 10 itearation on finest step

# Algorithm for Thai painting restoration

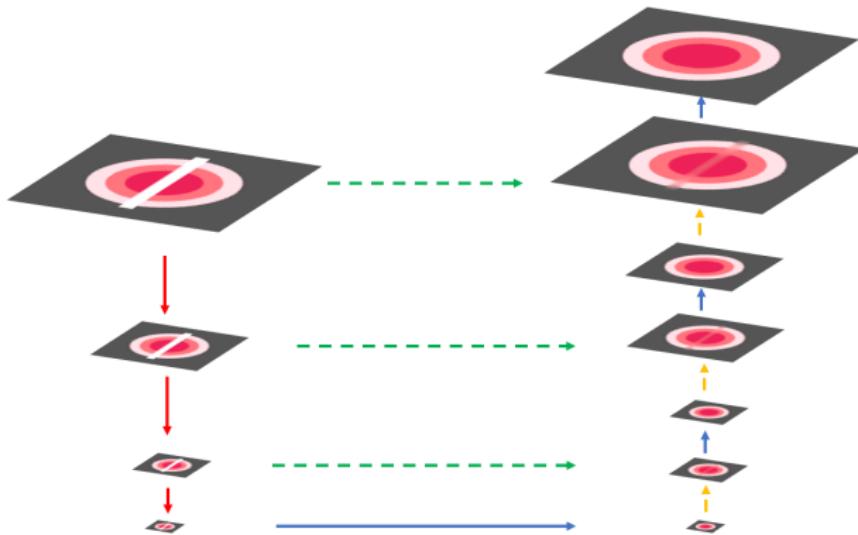


Figure 14: Our algorithm for Thai painting restoration

# Algorithm for Thai painting restoration

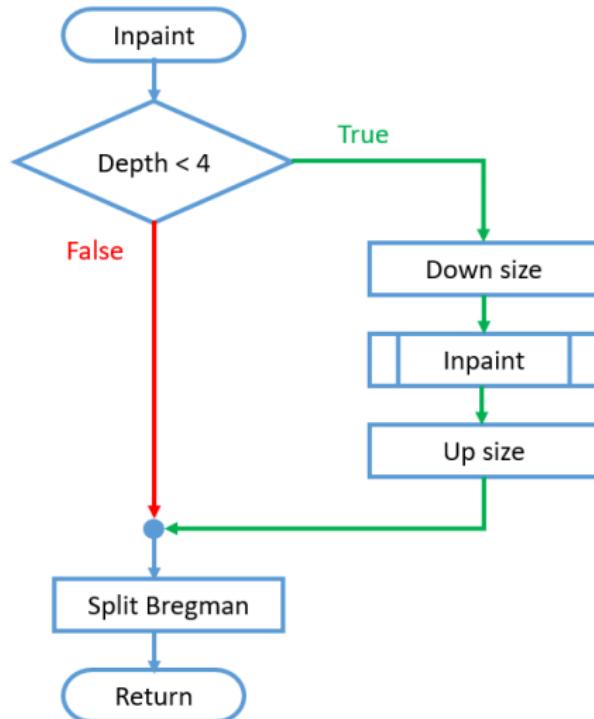


Figure 15: Our algorithm's flow chart

# Thai painting restoration



Figure 16: Original Thai painting

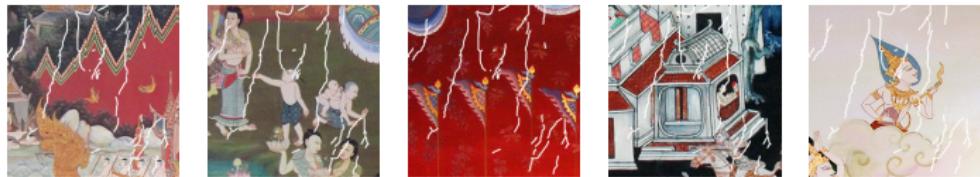


Figure 17: Damage Thai painting

# Thai painting restoration result



Figure 18: Split Breman



Figure 19: Our algorithm

## Comparison between 2 algorithms



(a) Original image



(b) Damaged image



(c) Split Bremgan



(d) Our algorithm

Figure 20: Comparison of images that have been expanded 4 times

# Thai painting restoration result

Method	Processing time (Second)	PSNR (dB)	SSIM
Split Bregman	2.72	34.89	1.0000
Our algorithm	0.39	35.30	1.0000

Table 4: Average Thai painting restoration

# Anime subtitle removal



Figure 21: Festival Asia Special Video - feat. Inori Aizawa

# Lorem Ipsum

Lore ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

But I must explain to you how all this mistaken idea of denouncing of a pleasure and praising pain was born and I will not give you a complete account of the system, and expound the actual teachings of the great explorer of the truth, the master-builder of human happiness. No one rejects, dislikes, or avoids pleasure itself, because it is pleasure

しかし私は、喜びを非難して苦痛を賞賛するという誤ったこの考えがすべてどのようにして誕生したかをあなたに説明しなければならないから、私はあなたにその体系を完璧に説明し、真実を求める偉大な探究家、人間の喜びを築く建築家の実践的な教えを詳しく説明しよう

หากแต่ข้าพเจ้าต้องขอ匕ายต่อท่านในเหตุที่มิโนดีอันความซึ้งอันหลงผิดทั้งหลาย ในการไฟห้าแลบสุดีบากแผล  
ร่างบ่ำเพาะขึ้นได้เยี่ยงไว้แล้ว ข้าพเจ้าจักให้สามารถแห่งระบบอันครบถ้วนแต่ท่าน และประดิษฐ์คำสอนที่แท้แห่งการแสวงหาอัน  
ยิ่งใหญ่ในความจริง ซึ่งเป็นบูรพคณานารย์ผู้สร้างความพำสุขแห่งมวลมนุษย์ ทุกวีผู้ได้มิรับ มิชอบ หรือเลียงหลีก  
ความสุข ด้วยเพราะมันเป็นสุขก็ทำไม่

Figure 22: Lorem Ipsum ทั้ง 4 ภาษา<sup>2</sup>

<sup>2</sup> Lorem ipsum in difference language [https://en.wikipedia.org/wiki/Lorem\\_ipsum](https://en.wikipedia.org/wiki/Lorem_ipsum), [https://ja.wikipedia.org/wiki/Lorem\\_ipsum](https://ja.wikipedia.org/wiki/Lorem_ipsum) and <https://th.wikipedia.org/wiki/ລອມເນື້ອປັບປຸງ>, 25 November 2018

# Split video

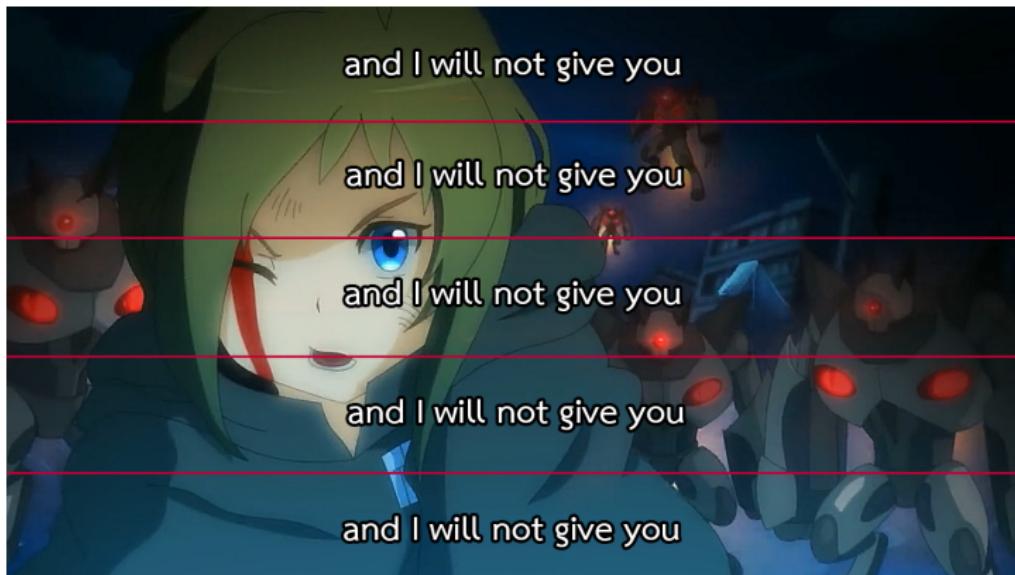


Figure 23: Split video into 5 parts

# Video and image

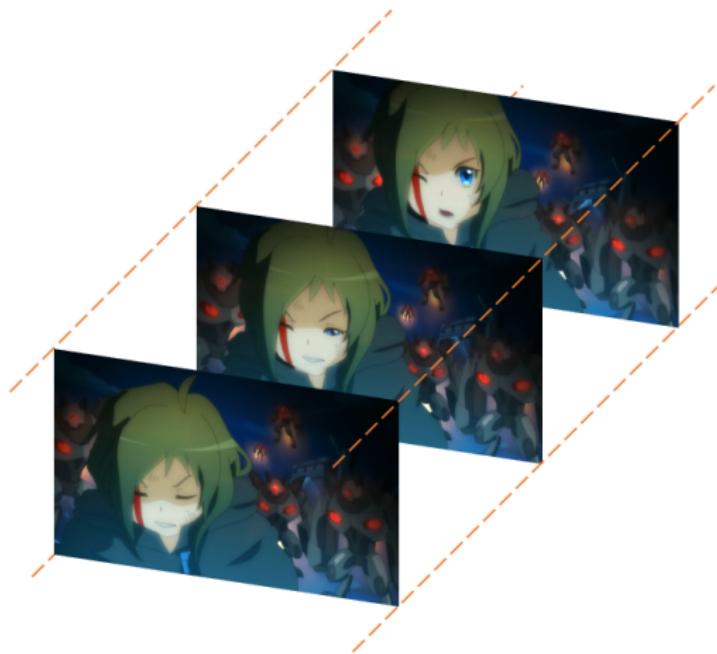


Figure 24: Video is sequences of image

## Finding subtitle

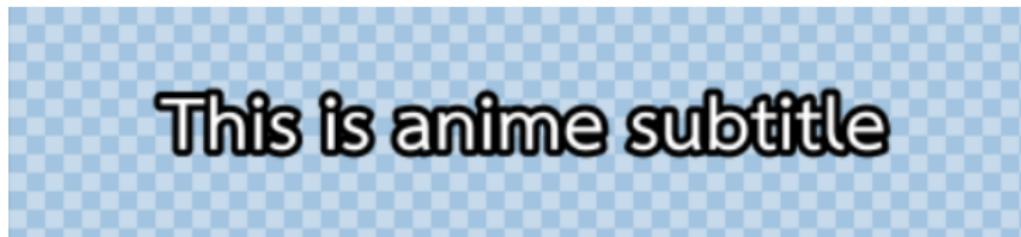
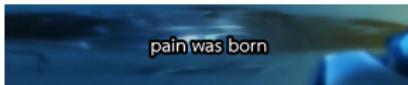
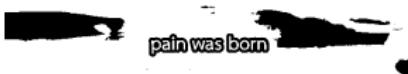


Figure 25: Anime subtitle always has black border

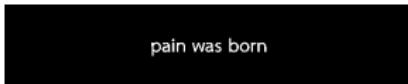
# Finding subtitle



(a) Anime with subtitle



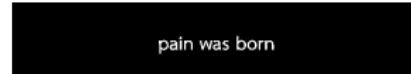
(c) inverse color



(e) remove smaller or larger object



(b) mark black as white



(d) remove white which attach to edge



(f) expand inpaint domain

Figure 26: our method to finding subtitle

## Finding subtitle result

Language	Pixel in inpaint domain	Pixel which detected	Pixel which error	Error (percent)
Thai	23,222,220	24,083,125	2,141,201	9.22
English	27,278,745	28,598,424	3,714,321	13.62
Japanese	28,544,173	30,103,466	3,740,971	13.11

Table 5: Average error of finding inpainting domain in various language subtitles

## Sequence of image and intial solution

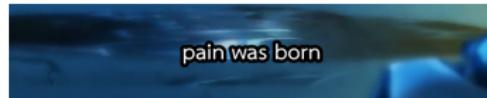


Figure 27: Image sequence in video which has subtitle

## Consider similarities with SSIM



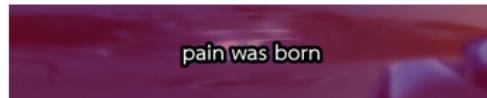
(a) Previous frame



(b) Current frame



(c) Area to find SSIM in previous frame



(d) Area to find SSIM in current frame

Figure 28: Area to calculate SSIM for skip frame and borrow frame method

# Skip frame

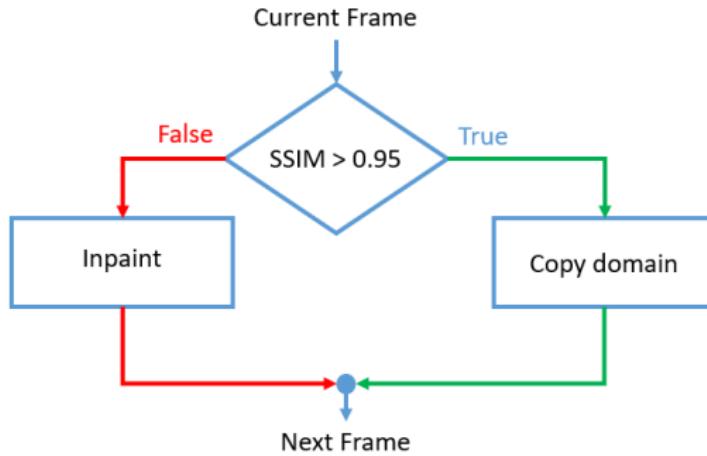


Figure 29: Skip frame method

# Borrow frame

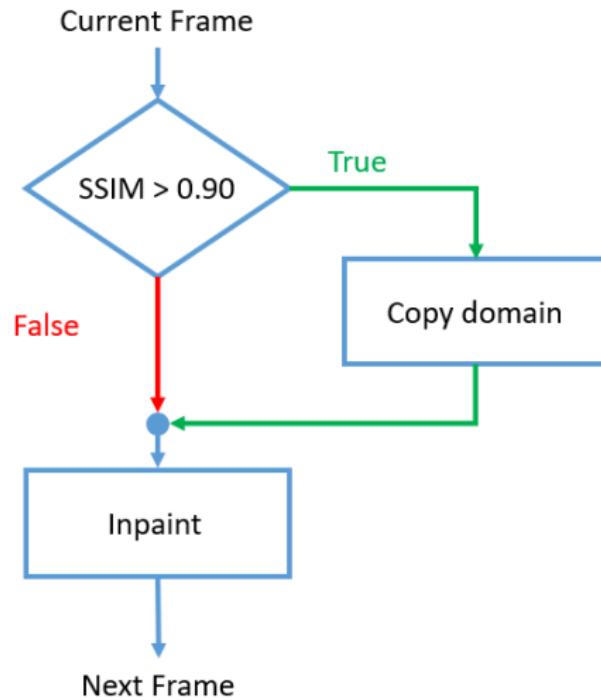


Figure 30: Borrow frame method

# Skip and borrow frame

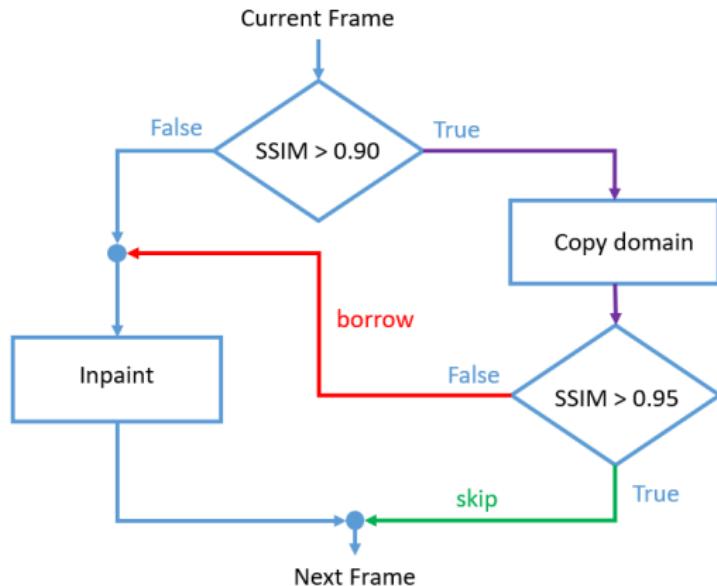


Figure 31: Skip and borrow frame method

## Skip/Borrow frame result

Method	Processing time (Second)	PSNR (dB)	SSIM
Our first algorithm	141.29	31.39	0.9510
Skip frame	89.29	29.07	0.9408
Borrow frame	132.78	32.20	0.9655
Skip and borrow frame	75.76	29.33	0.9454

Table 6: Average result of removing subtitle from anime

# Algorithm for removing subtitle from anime

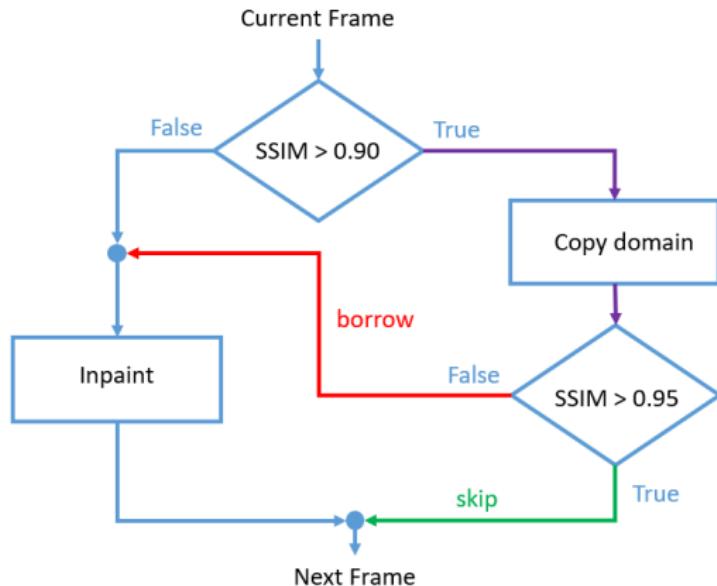


Figure 32: Skip and borrow frame method

# Removing subtitle result



(a) Before



(b) After

# removing subtitle result

Method	Processed time (Second)	PSNR (dB)	SSIM
Split Bregman	5073.08	32.88	0.9654
Our algorithm	75.76	29.33	0.9454

Table 7: Average result of removing subtitle with Split Bregman and our algorithm

# Thai painting restoration

ก้าวเดียวเปลี่ยนรูปภาพเดิม

ก้าวเดียว: รีแมปใบใหญ่ๆ (1)

เมื่อต้องการที่จะบันทึกภาพไว้ต่อไปน้ำยาเพื่อกำลังปืนออกตาม หรือต้องรักษาใบไม้ให้คงทน

พารามิเตอร์สำหรับการตัดต่อ

สำหรับการตัดต่อที่ต้องการใช้เวลาประมาณ 20 วินาที ที่นี่ค่าเริ่มต้นของค่าตัดต่อที่ดีที่สุดที่สามารถได้มาโดยใช้เวลาตัดต่อที่ต่ำกว่า 1 วินาที แต่ก็ต้องยอมรับว่าคุณภาพของภาพที่ตัดต่อไว้ ซึ่งก้าวตามากก็จะดีกว่า

θ: 5

λ: 250

ε: 0.0001

ค่าตัดต่อที่แนบมา: 10

ค่าตัดต่อที่บันทึกไว้ครั้ง: 3

ค่าตัดต่อที่บันทึกไว้: 10

ความถี่ของตัดต่อ: 4

ก้าวเดียวที่ต้องการซ่อมแซม



Figure 34: Try by yourself at <https://bit.ly/demothai>

# Program for removing subtitle from anime

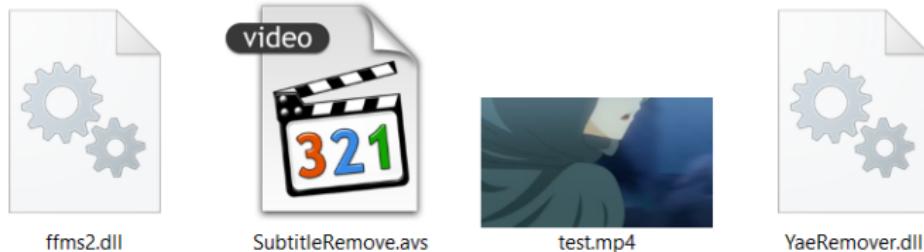


Figure 35: file for removing subtitle (<https://bit.ly/demo-anime-inpaint>)

```
1 LoadPlugin("ffms2.dll") #https://github.com/FFMS/ffms2
2 LoadPlugin("YaeRemover.dll") #our method
3 Video = FFMS2("test.mp4", ATrack=-1, fpsnum=24000,
   fpsden=1000, ColorSpace="RGB24", UTF8=True)
4
5 Result = Video.YaeRemover(Left = 280, Right = 1000, Top =
   613, Bottom = 683, StokeWidth = 6)
6 return Result
```

Figure 36: source code in SubtitleRemove.avs

# Thank you