Lecture 15-1-Image Blending

Yuyao Zhang, Xiran Cai PhD

zhangyy8@shanghaitech.edu.cn caixr@shanghaitech.edu.cn

SIST Building 2 302-F/302-C

Course piazza link:
piazza.com/shanghaitech.edu.cn/spring2021/cs270spring2021

上海科技大学
ShanghaiTech University

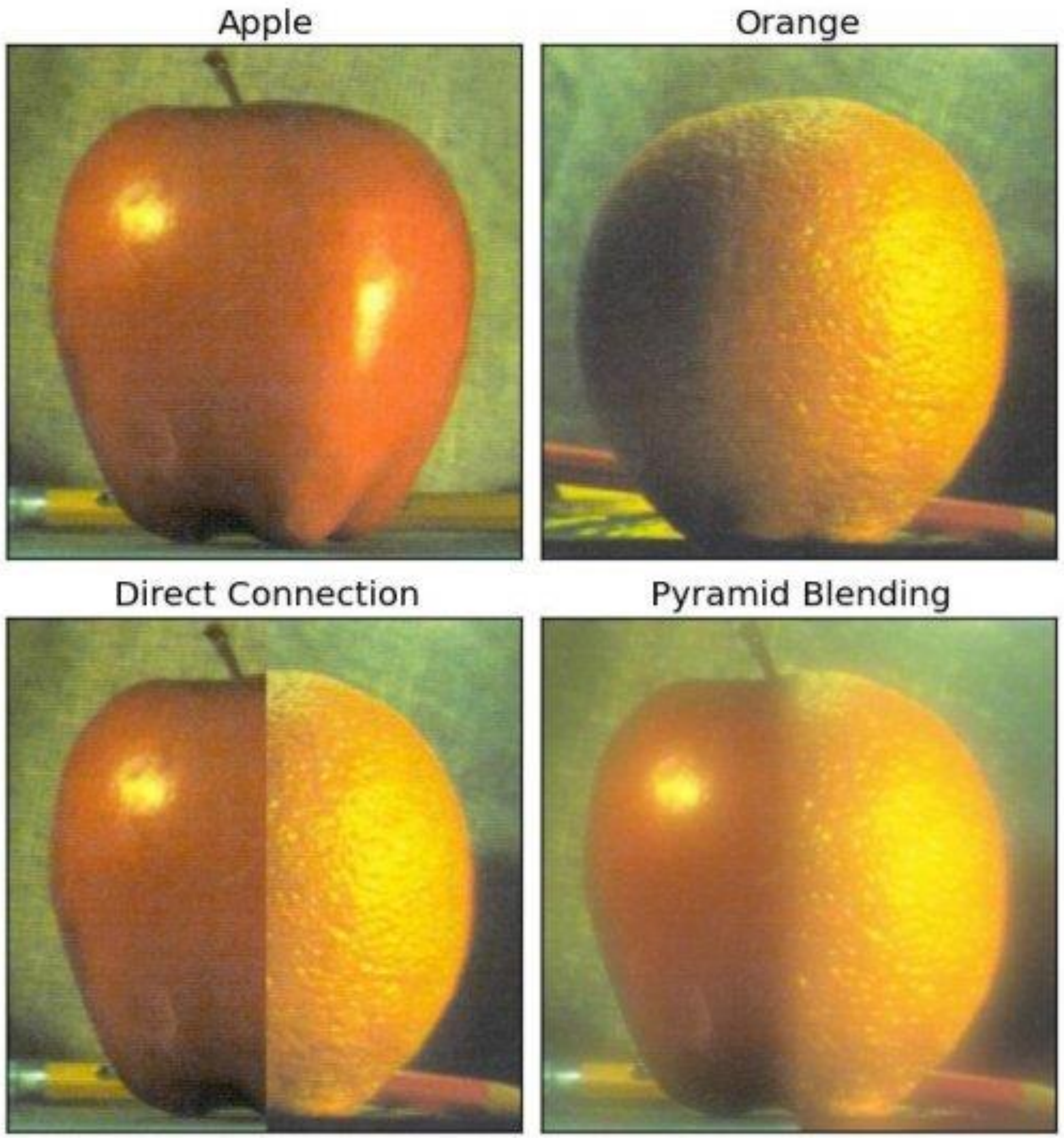
Long history of fake images

Long history of fake images

上海科技大学 ShanghaiTech University

Long history of fake images

Hard edge composition vs Pyramid Blending



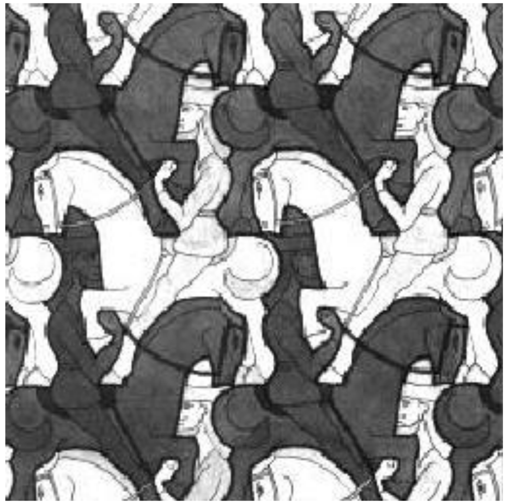
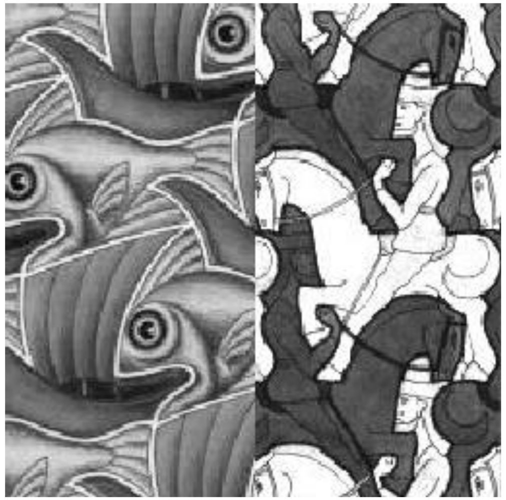
Hard compositing

- Hard compositing:

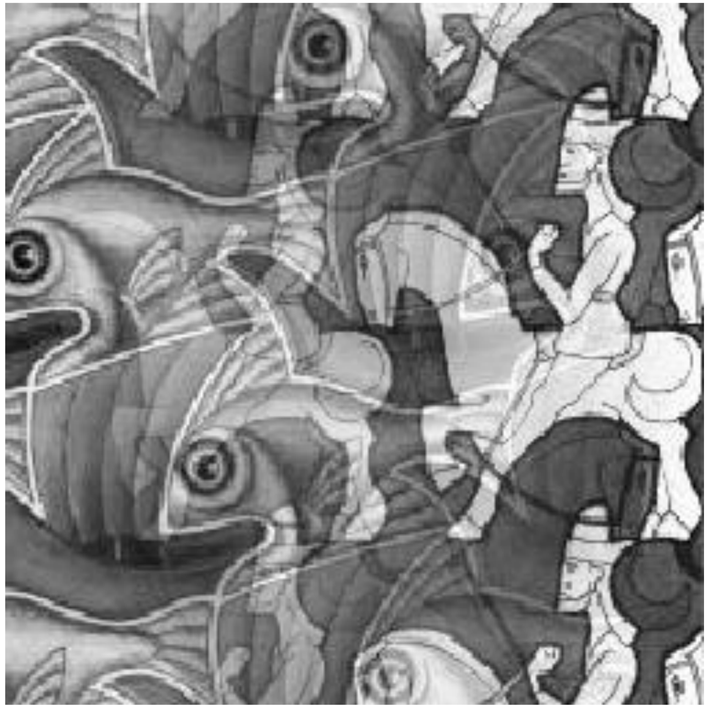
$$I(x,y) = M(x,y)S(x,y) + \left(1 - M(x,y)\right)T(x,y)$$

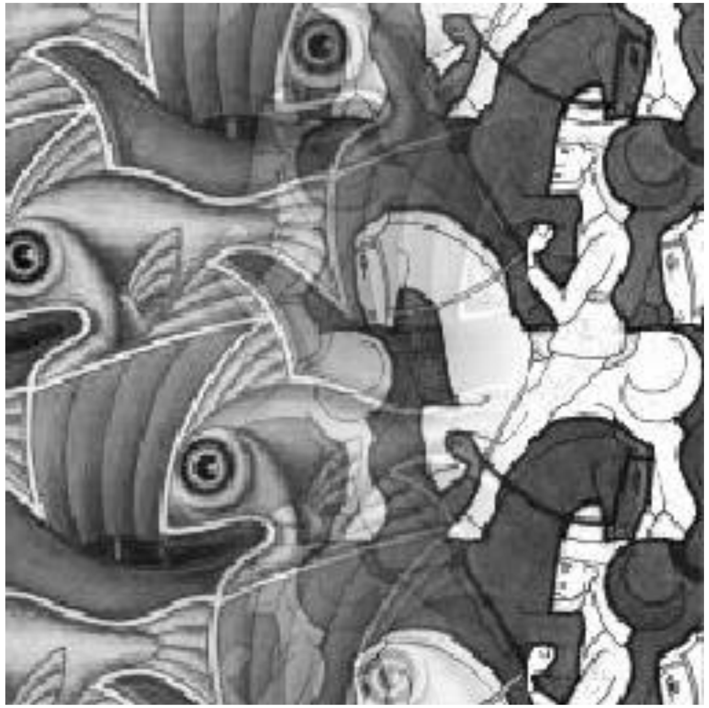
$$= \begin{cases} S(x,y) & M(x,y) = 1 \\ T(x,y) & M(x,y) = 0 \end{cases}$$

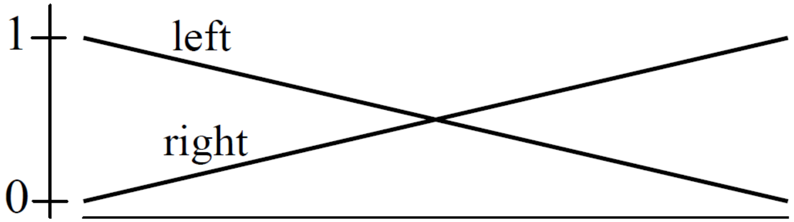
- Generally bad: seam/matte line is visible

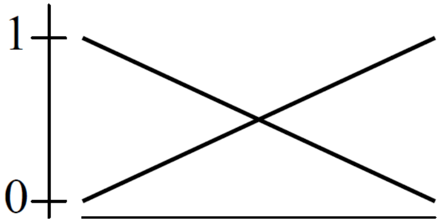




Weighted transition region:

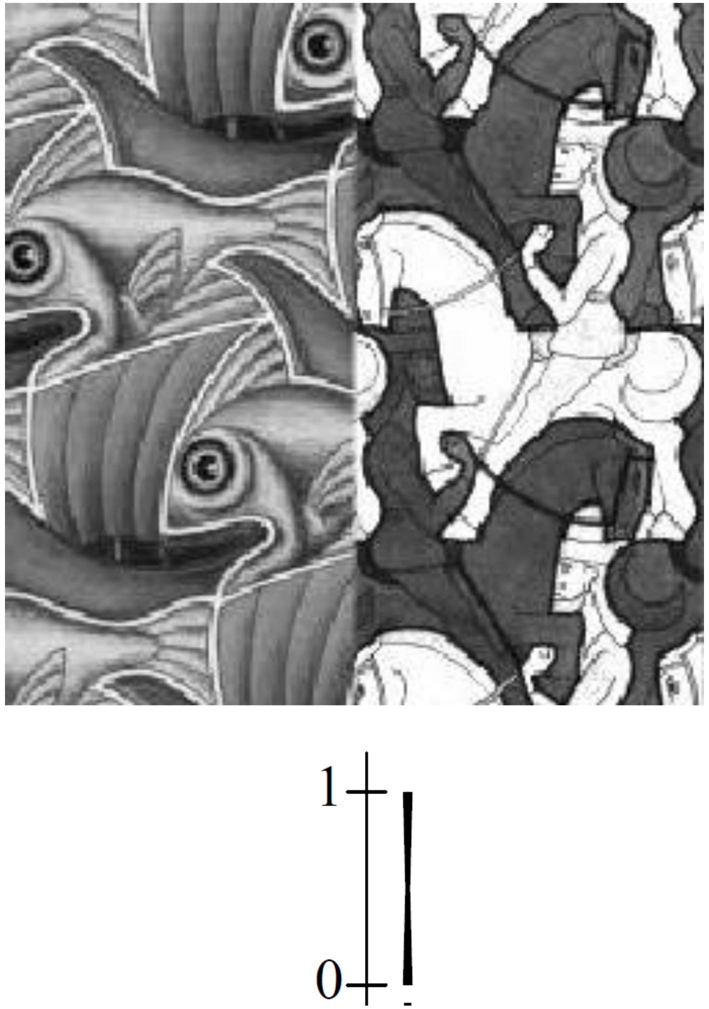




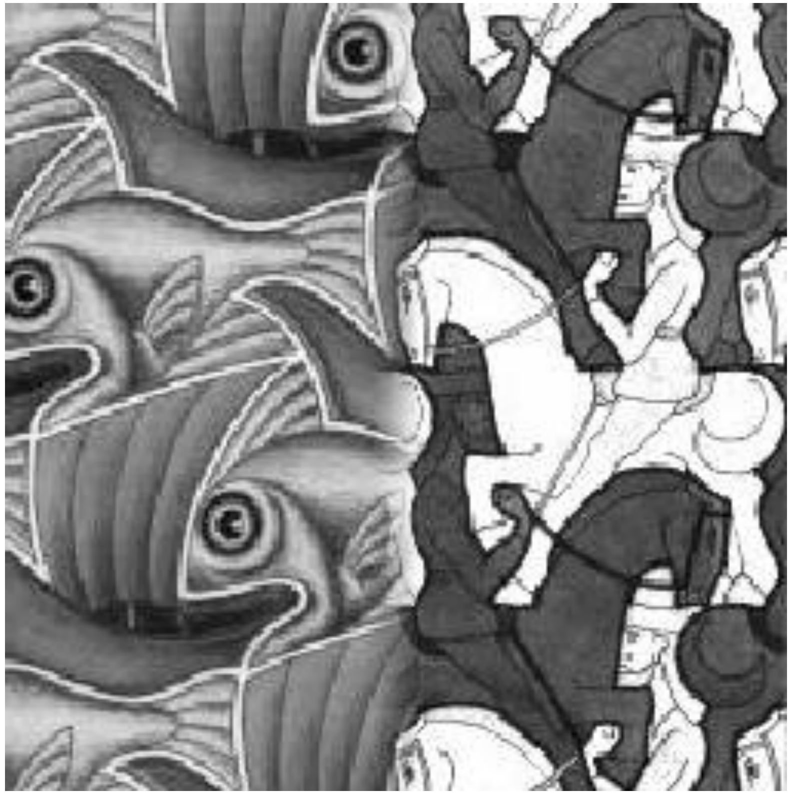


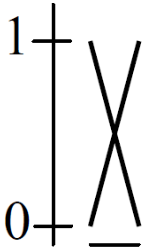


Weighted transition region:



Good window size





Pyramid Blending

➢ **Better idea: Multi-resolution blending with a Laplacian pyramid.**

- Idea: wide transition regions for low-frequency component, narrow transition regions for high-frequency component (edges).
- Gaussian pyramid:

  G = 5x5 Gaussian filter

  $\boldsymbol{I_0}$ = original image (full resolution)

  - $\boldsymbol{I}_{\mathrm{i}} = (\boldsymbol{G} * \boldsymbol{I}_{\mathrm{i}-1}) \downarrow 2$ ← Down-sample twice

    convolution

- Get a series of smaller and blurry images.

What does blurring take away?

What does blurring take away?

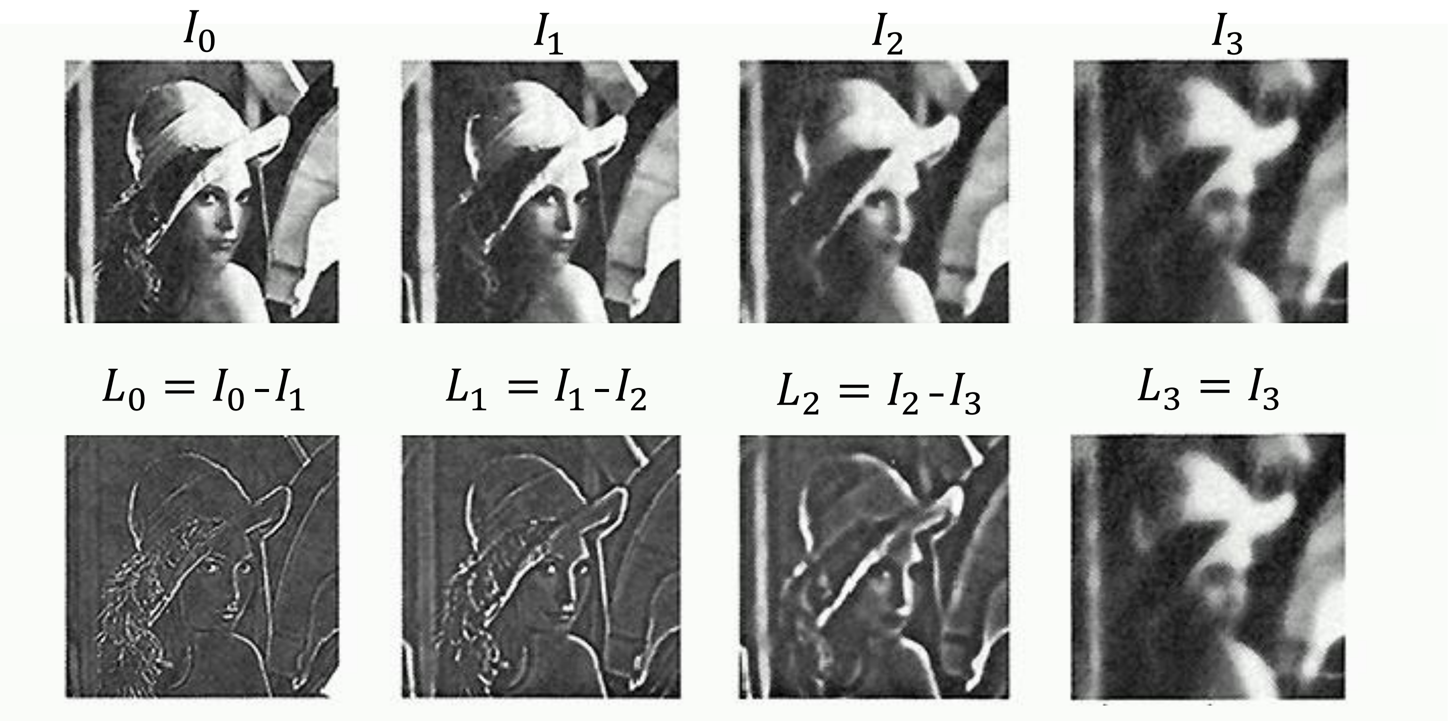
What does blurring take away?

Pyramid Blending

- **Difference of Gaussian at each scale:**

High-pass image at scale i → $\boldsymbol{L_i} = \boldsymbol{I}_{\mathrm{i}} - \boxed{(\boldsymbol{G} * \boldsymbol{I}_{\mathrm{i}}) \downarrow 2}$ ← Blurred version of level i

Gaussian pyramid image at scale i

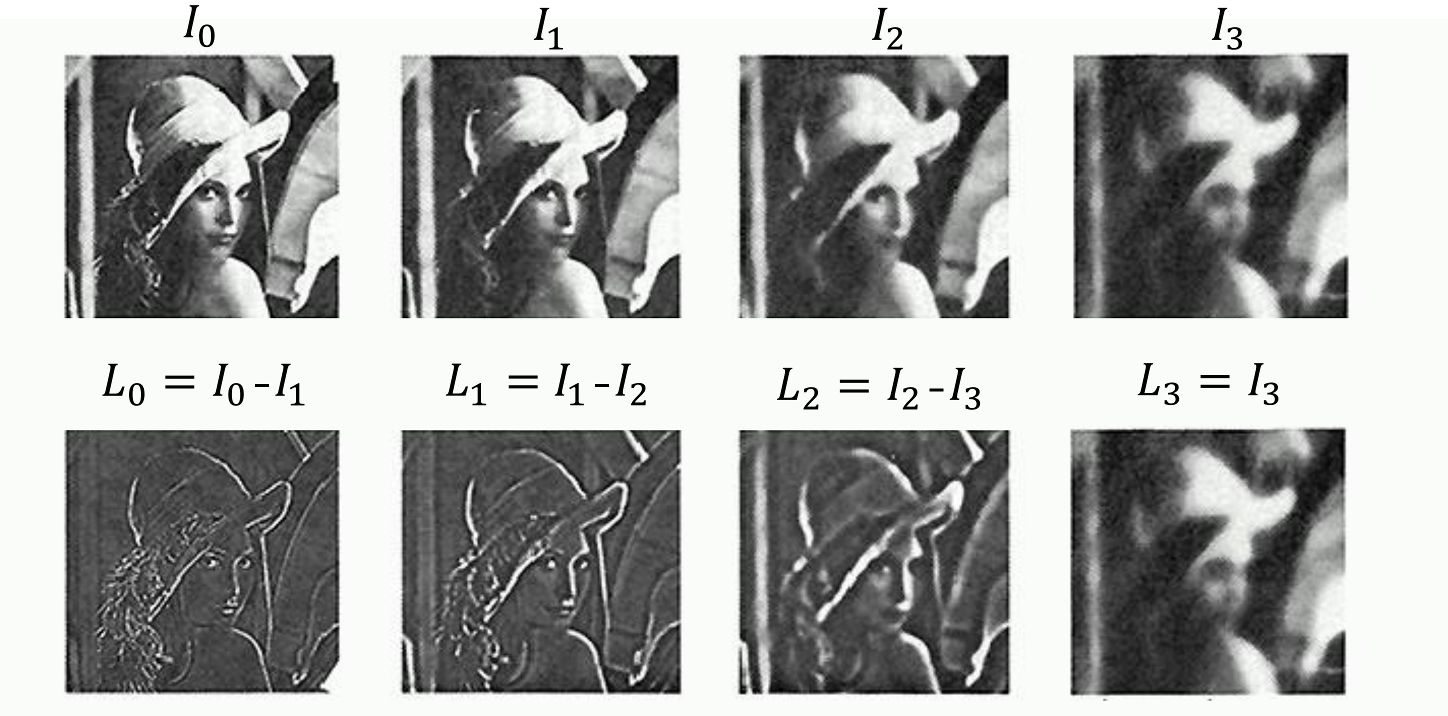


$\{L_i\}$ = the set of L_i form. A Laplacian pyramid $L_1, L_2, L_3 \cdots, L_n$

Pyramid Blending

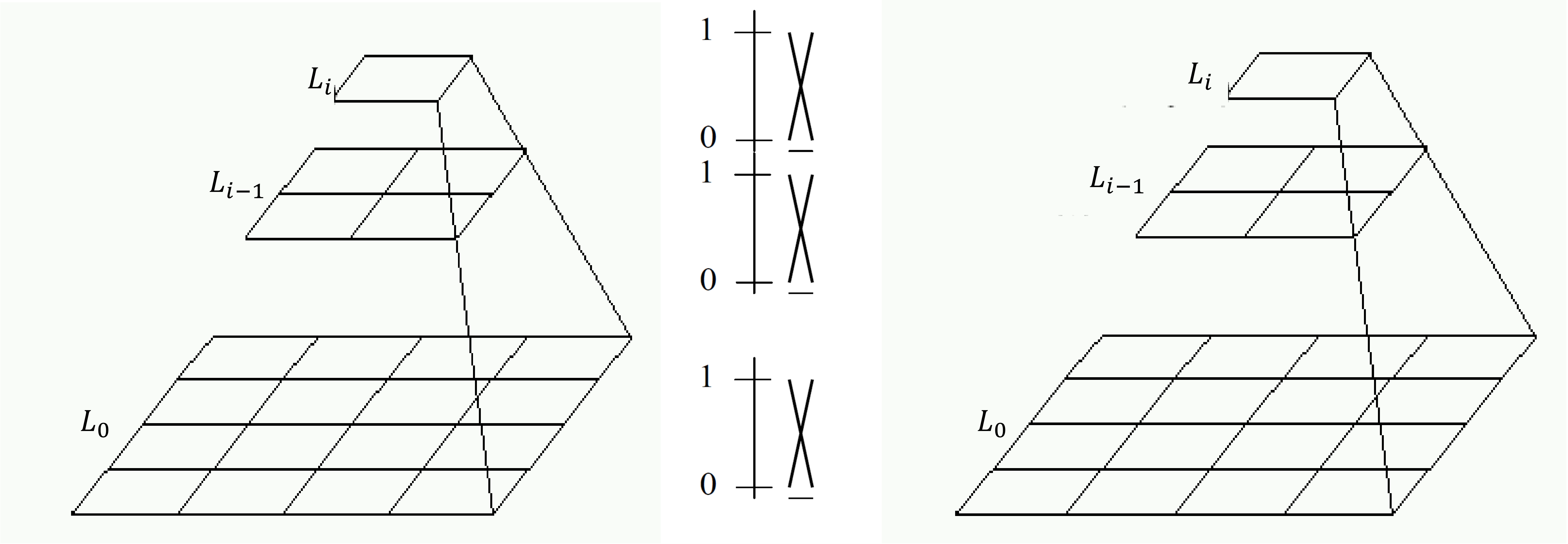
- **We can recover the original as:**

$$\boldsymbol{I} = \sum_{\boldsymbol{i}=\boldsymbol{0}}^{\boldsymbol{N}} (\boldsymbol{L_i}) \uparrow$$



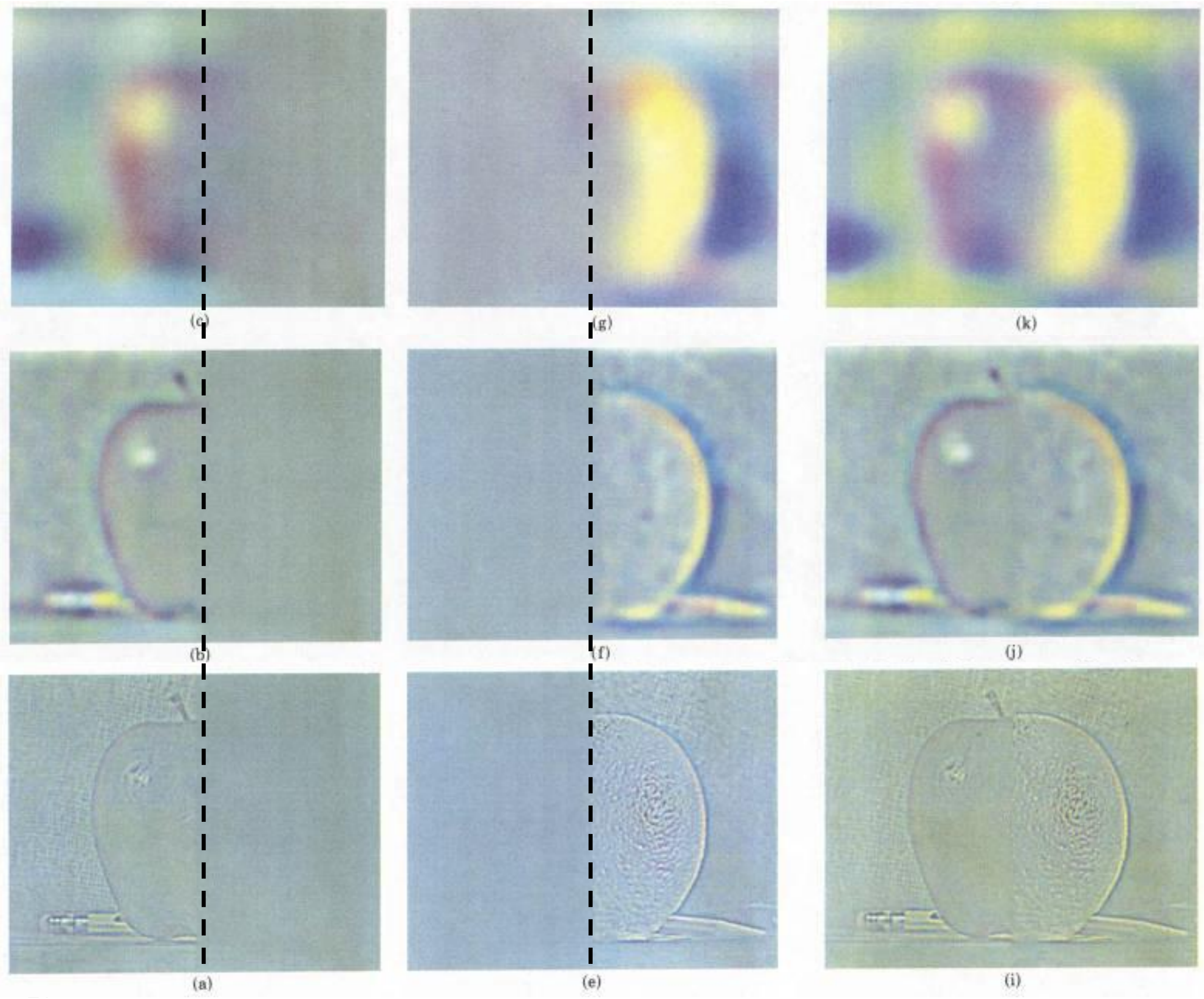
$\{L_i\}$ = the set of L_i form. A Laplacian pyramid L_1, L_2, $L_3 \cdots$, L_n

Pyramid Blending



Left pyramid　　　blend　　　Right pyramid

Pyramid Blending



Season Blending

上海科技大学
ShanghaiTech University

Season Blending

Target image

Target image with editing region

Source image

Result of pyramid blending

Poisson image editing

- A even better idea: to reduce the color mismatch between source and target, create composite in gradient domain.

- We want the gradient of the composite inside $\boldsymbol{\Omega}$ to look as close as possible to the source image gradient. The composite must match target image on the boundary $\boldsymbol{\partial\Omega}$.

Ω

$S(x, y)$

$$\min_{I(x,y)\epsilon\Omega} \sum \|\nabla I(x,y) - \nabla S(x,y)\|^2$$

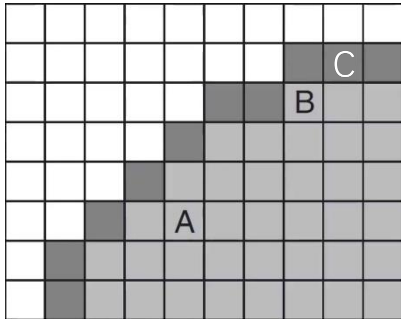
$T(x, y)$

$I(x, y)$

$$s.t. I(x,y) = T(x,y) \text{ on } \partial\Omega$$

- We want the gradient of the composite inside $\boldsymbol{\Omega}$ to look as close as possible to the source image gradient. The composite must match target image on the boundary $\boldsymbol{\partial\Omega}$.

Poisson image editing



- **Solution for this Pb:**

$$\nabla^2 I(x,y) = \nabla^2 S(x,y) \text{ in } \Omega$$

$$I(x,y) = T(x,y) \text{ on } \partial\Omega$$

- **Poisson equation**
- **Discretizing and solving the problem:**
- **1) For a pixel A inside Ω,**

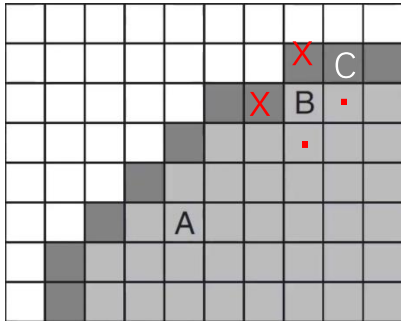
$$\nabla^2 I(x,y) = \nabla^2 S(x,y)$$

	1	
1	-4	1
	1	

$$I(x+1,y)+I(x,y+1)+ I(x-1,y)+I(x,y-1)- 4*I(x,y)$$

$$S(x+1,y)+S(x,y+1)+ S(x-1,y)+S(x,y-1)- 4*S(x,y)$$

Poisson image editing



➢ For a pixel B not inside Ω (whose neighbor is Ω).

$$\nabla^2 I(x,y) = \nabla^2 S(x,y)$$

$$\begin{matrix} I(x+1,y)+I(x,y+1)+ \text{ (xx)} \\ T(x-1,y)+T(x,y-1)- \text{ (..)} \\ 4*I(x,y) \end{matrix} \qquad \begin{matrix} S(x+1,y)+S(x,y+1)+ \\ S(x-1,y)+S(x,y-1)- \\ 4*S(x,y) \end{matrix}$$

- Big linear system: so in all there will be N unknowns and N equations that can be divided into 3 different groups

5 non-zeros values in a row

Less than 5 non-zeros values in a row

1 non-zeros value in a row

$$\begin{bmatrix} -4 & 1 & 0 & \dots & 0 & 1 & 0 & \dots & 0 \\ 1 & -4 & 1 & 0 & \dots & 0 & 1 & \dots & 0 \\ & & & & \vdots & & & & \\ \dots & 1 & -4 & 1 & 0 & \dots & 0 & 0 & 0 \\ & & & & \vdots & & & & \\ 0 & 0 & \dots & 1 & 0 & 0 & 0 & 0 & 0 \\ & & & & \vdots & & & & \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_N \end{bmatrix} = \begin{bmatrix} \nabla^2 S^1 \\ \nabla^2 S^2 \\ \vdots \\ \nabla^2 S^n - \sum T_n \\ \vdots \\ T_N \end{bmatrix}$$

Group 1: $A \in \Omega$

Group 2: $B \in \partial\Omega \cap \Omega$

Group 2: $C \in \partial\Omega$

Source image

Target image

Poisson image editing result

Take home message

- Pyramid image blending is able to merge two images with similar background, however is not robust for color mismatch.
- Poisson image edit is more powerful on image blending Pbs with variations on background color.