Video 8.1
Jianbo Shi

Gradient Blending

How to blend two images?

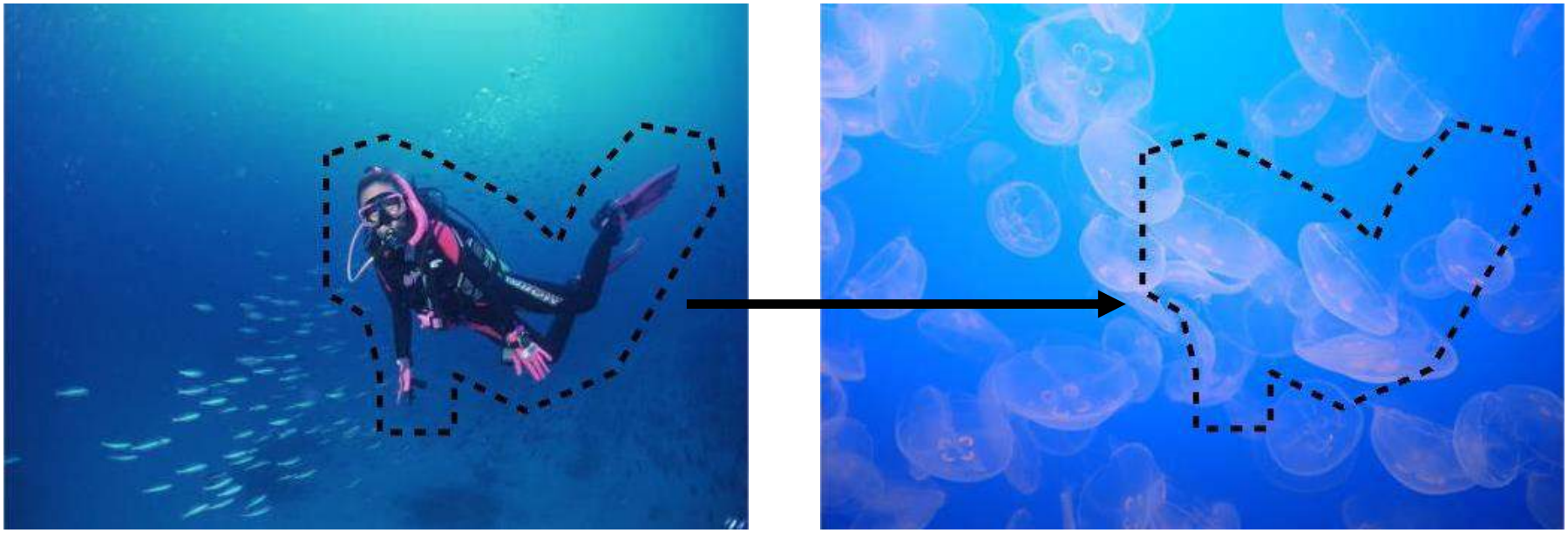
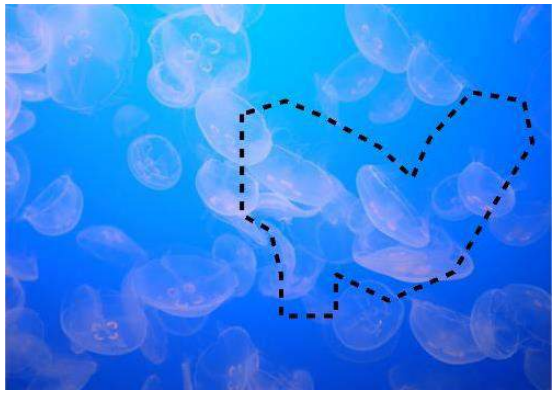


image blending: image surgery…

- cutting from one image (which we will cover in details on segmentation)
- reconstructing onto the new image

Blend = Cut and Paste images



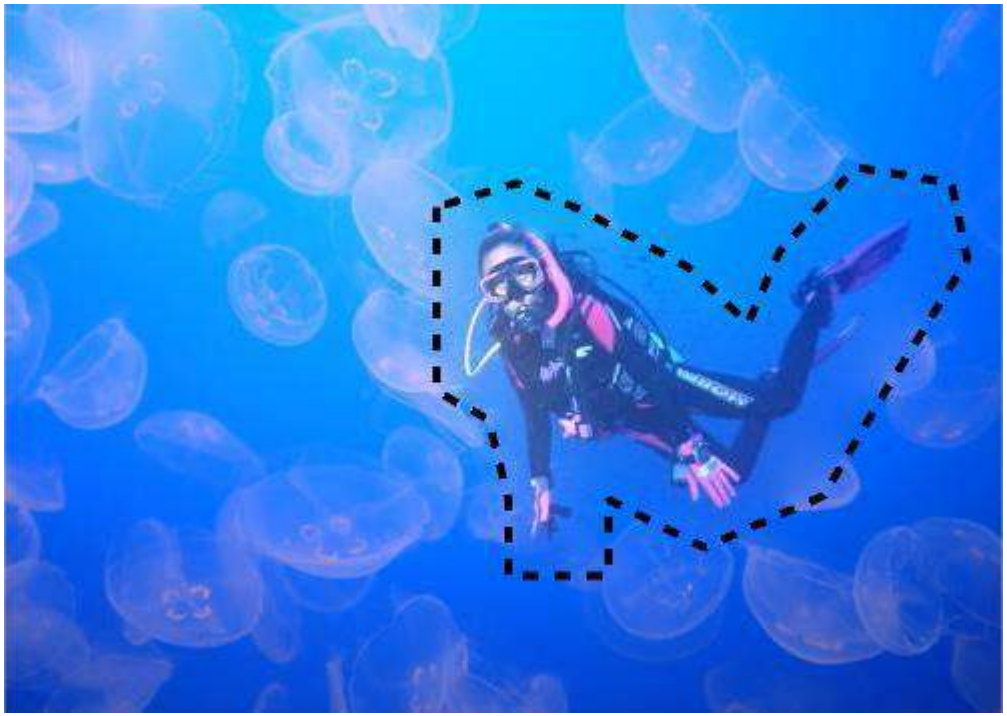
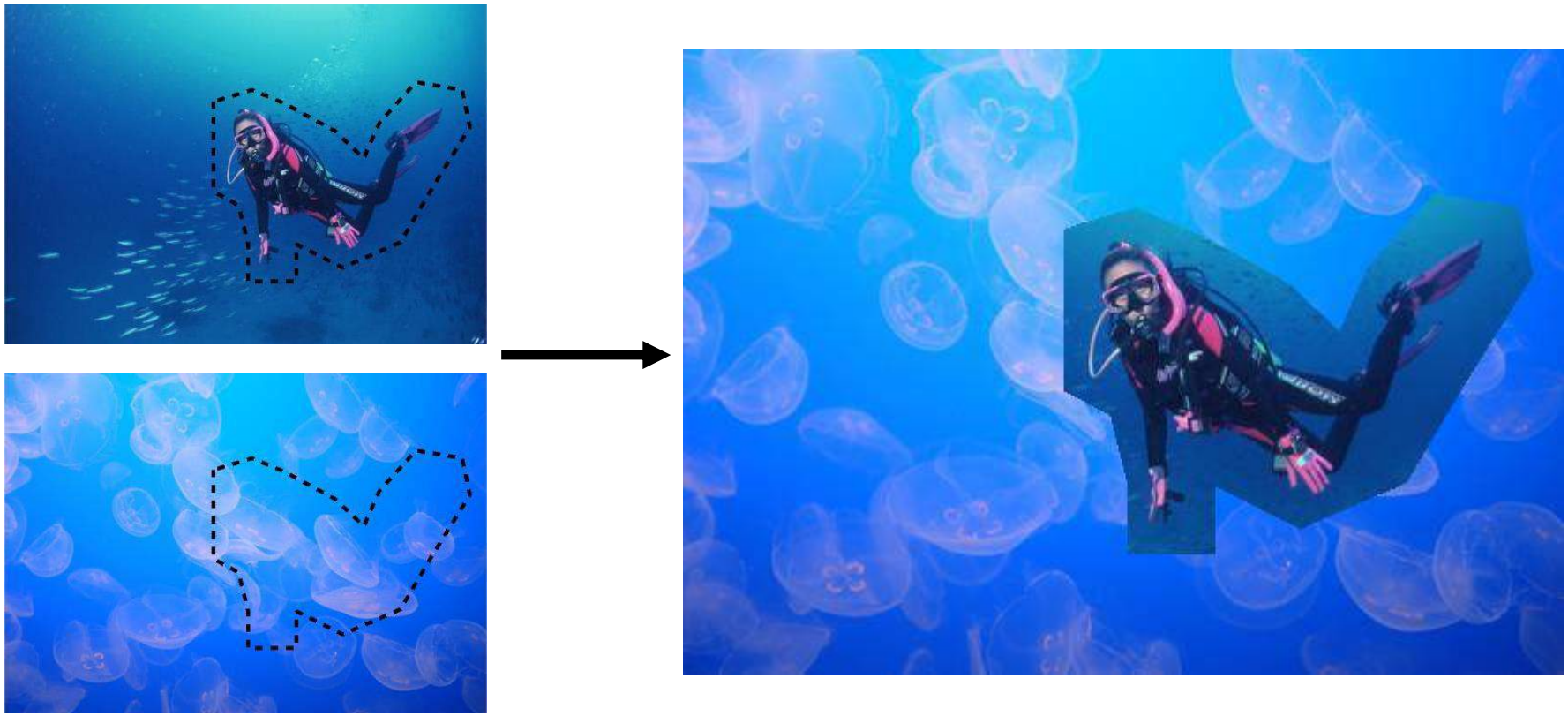


Image blending is an art of …

faking images, hiding evidence of image surgery, making it looks natural

Direct Copy and Paste

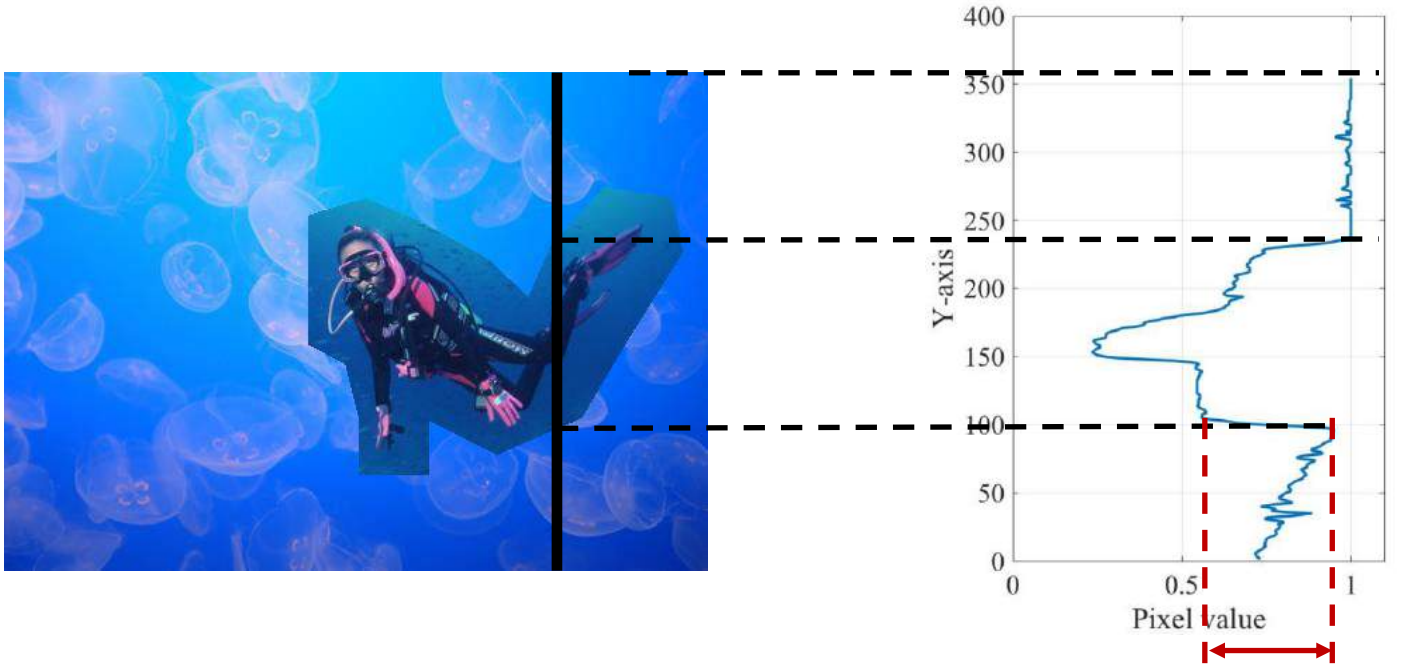


Direct attempt: not so good!

-- we created an artificial boundary between the pasted region

Challenge: color and brightness mismatch

Blue channel value of the vertical line

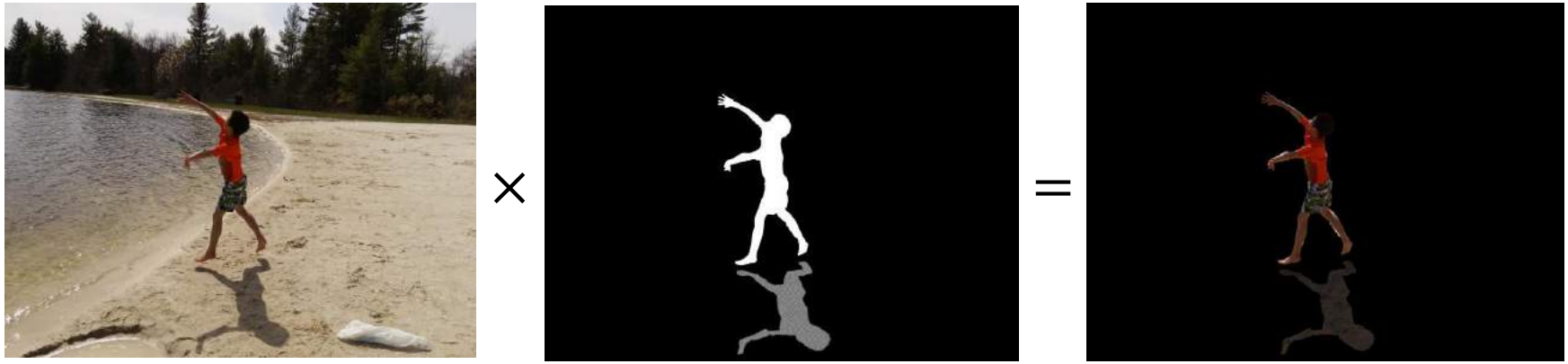


Big change in intensity

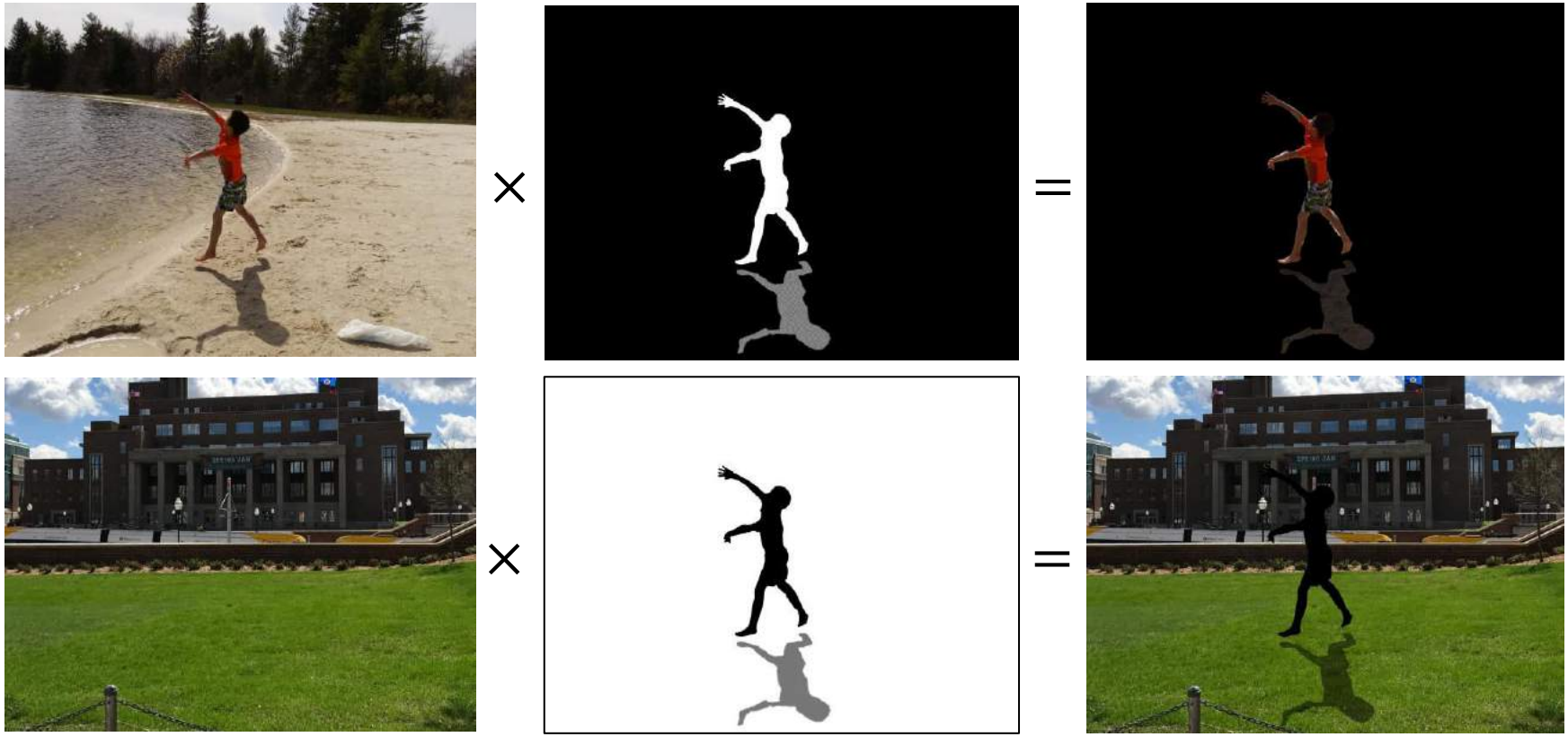
Big change in intensity creates new image boundary…
- any ideas on how to remove that boundary?

Alpha blending

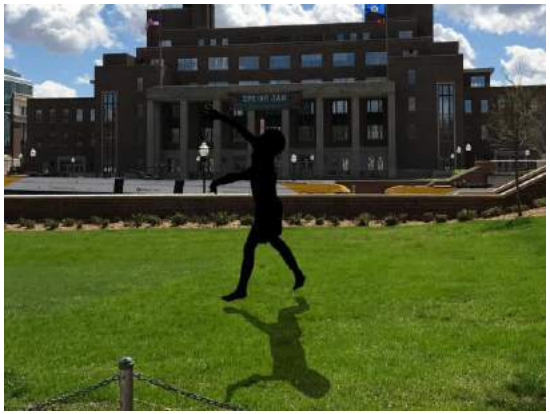
Alpha channel encodes the transparency of the object



Alpha blending

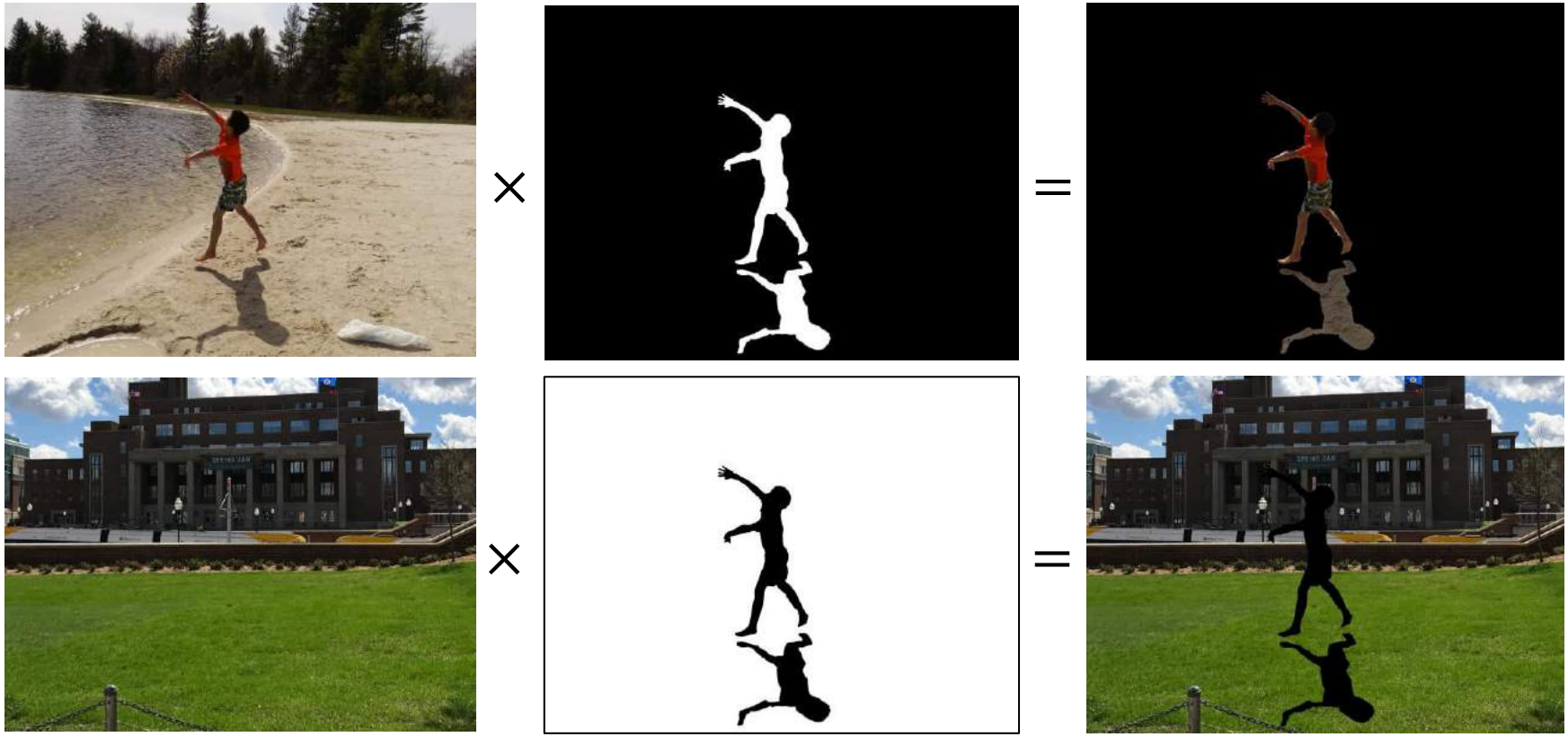


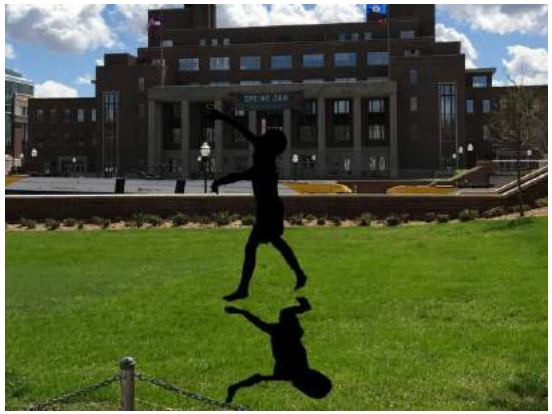
Alpha blending

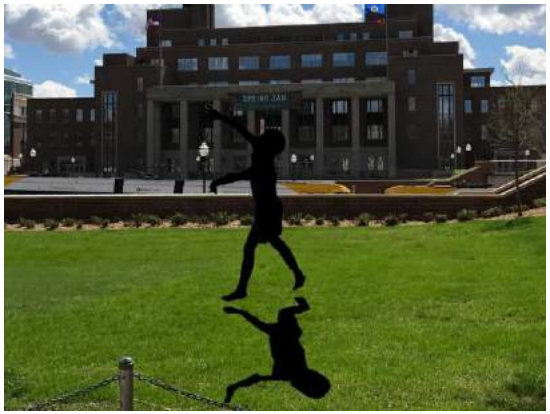


Alpha blending

Copy - paste is a special kind of alpha blending – binary mask







copy - paste

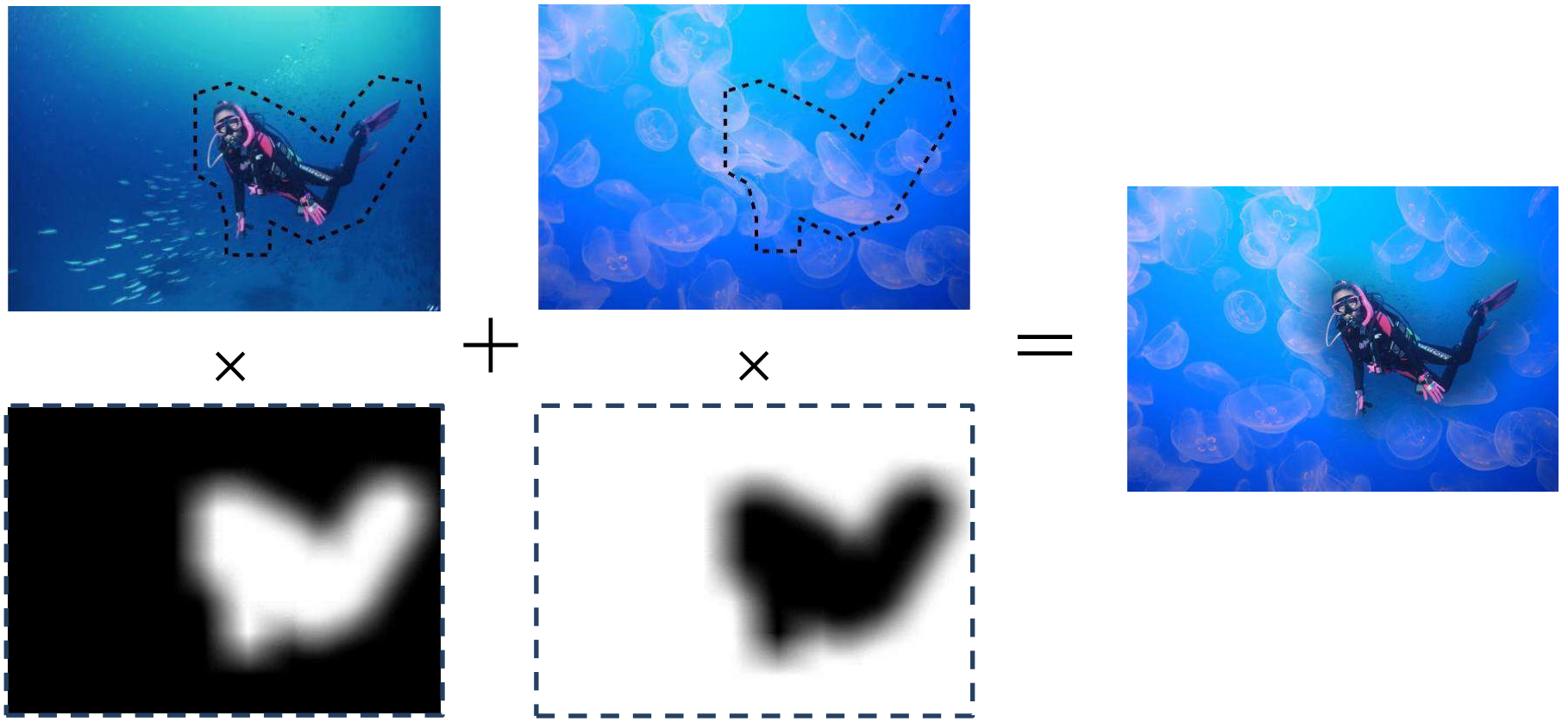
alpha blending

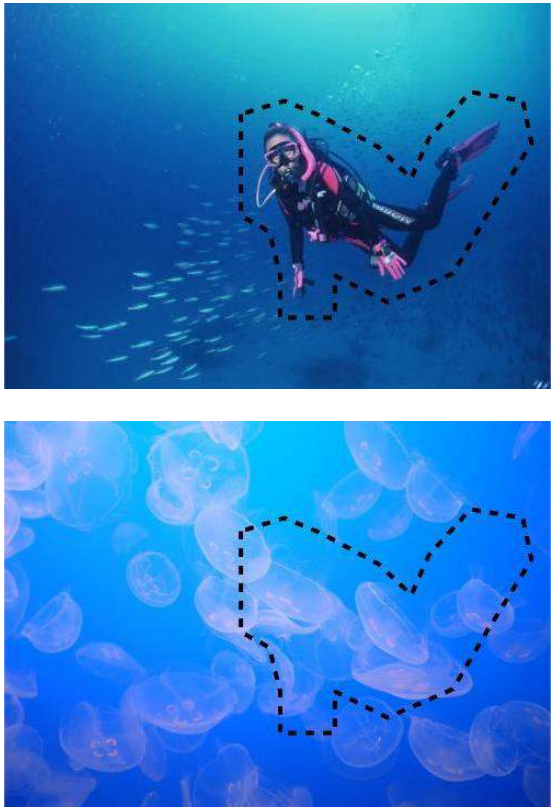
Alpha blending can deal with transparent objects

copy - paste

alpha blending

Alpha blending hacking



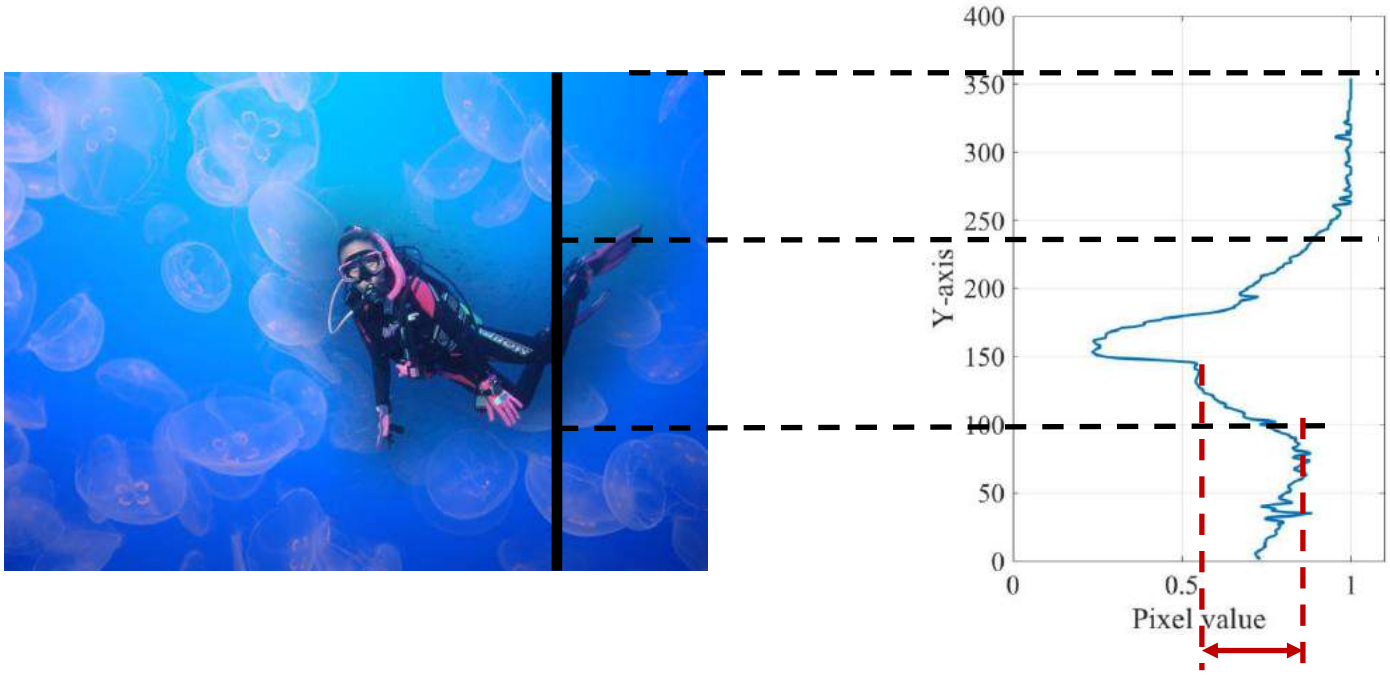


copy - paste

Alpha blending

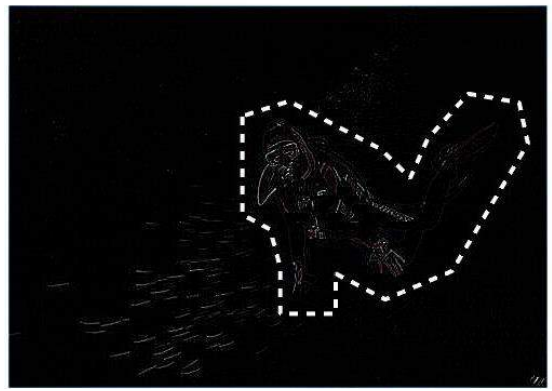
Alpha Blending

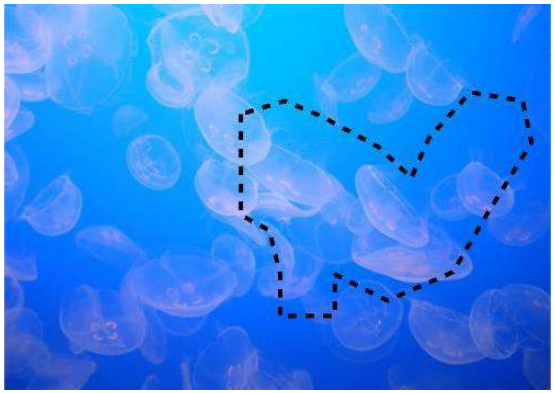
Blue channel value of the vertical line



Continuous change,
but still a visually un-natural pattern in intensity

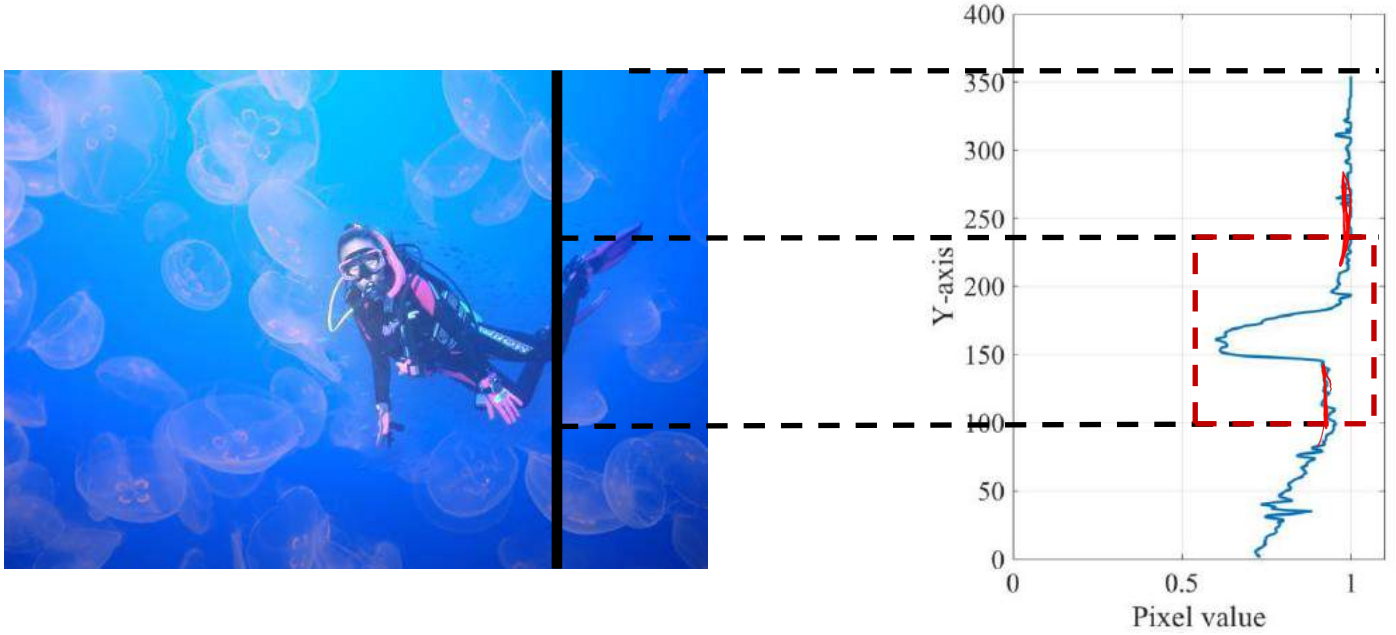
Solution: Copy only gradient + Recreate





Gradient Blending:
No more intensity change!

Blue channel value of the vertical line



Smooth transition

Image Blending

copy - paste

alpha blending

Gradient blending

Results of gradient blending

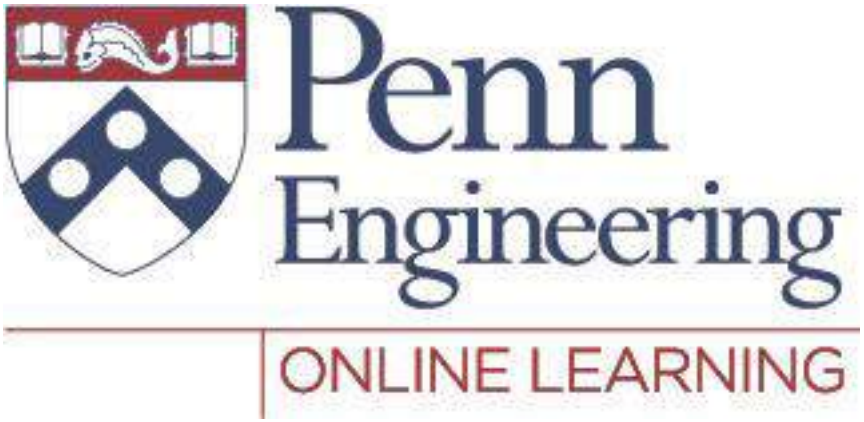
Source (figure) Target (background) Result

Color of the hat blends into the background
-- successful image surgery… with interesting side-affect


Gradient Blending

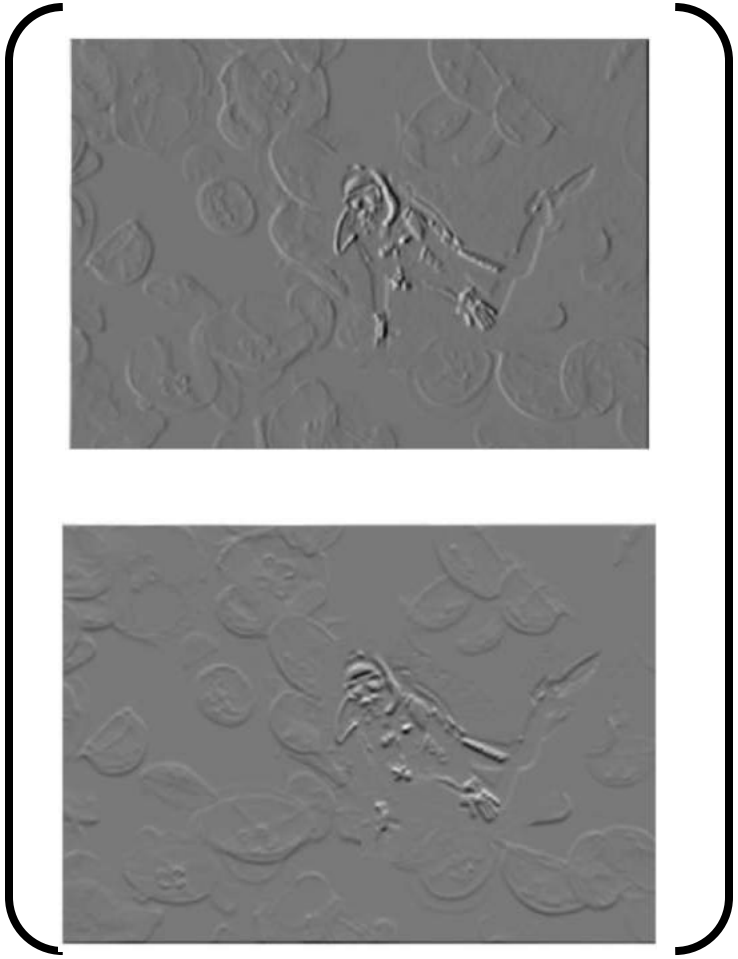

Gradient Blending


Gradient Blending



Video 8.2
Jianbo Shi

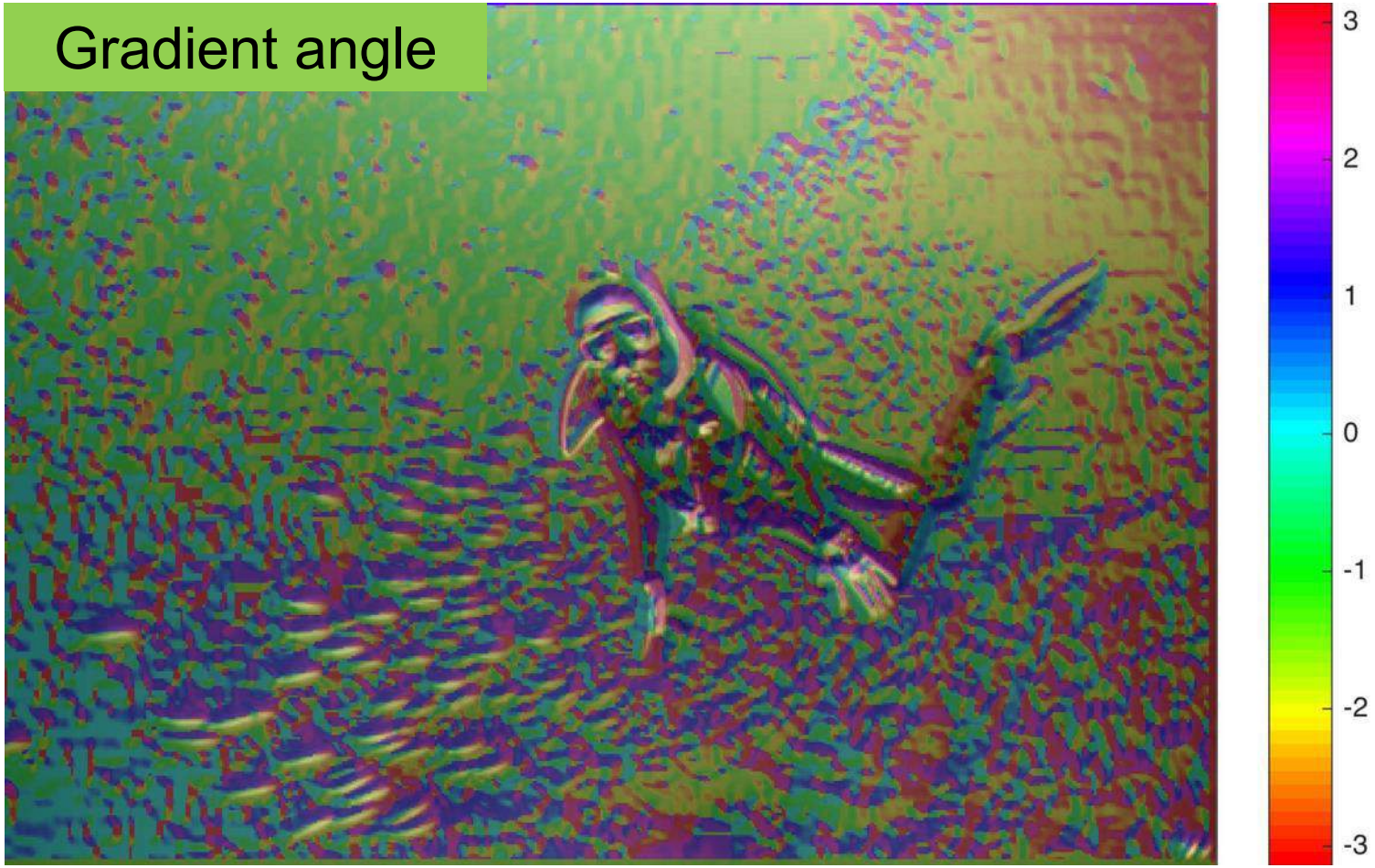
Why use the gradients to recreate images



Source Image

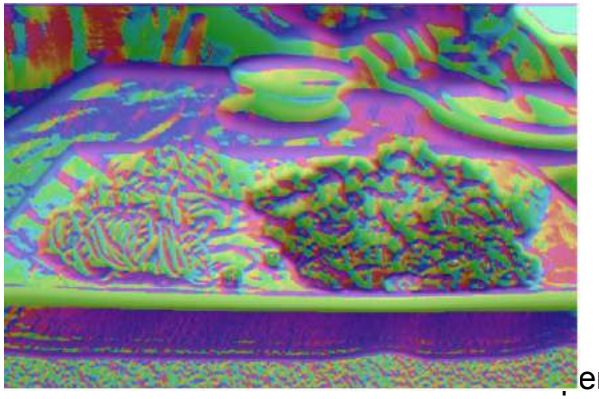
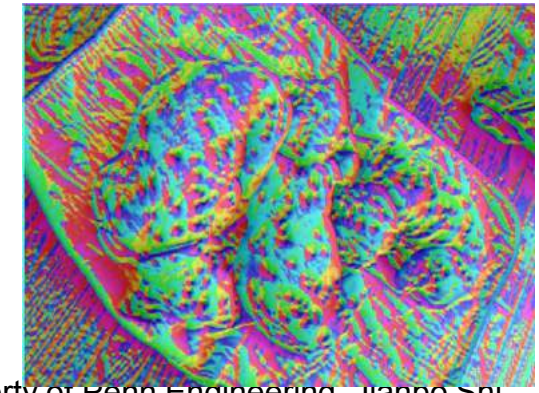
Gradient magnitude

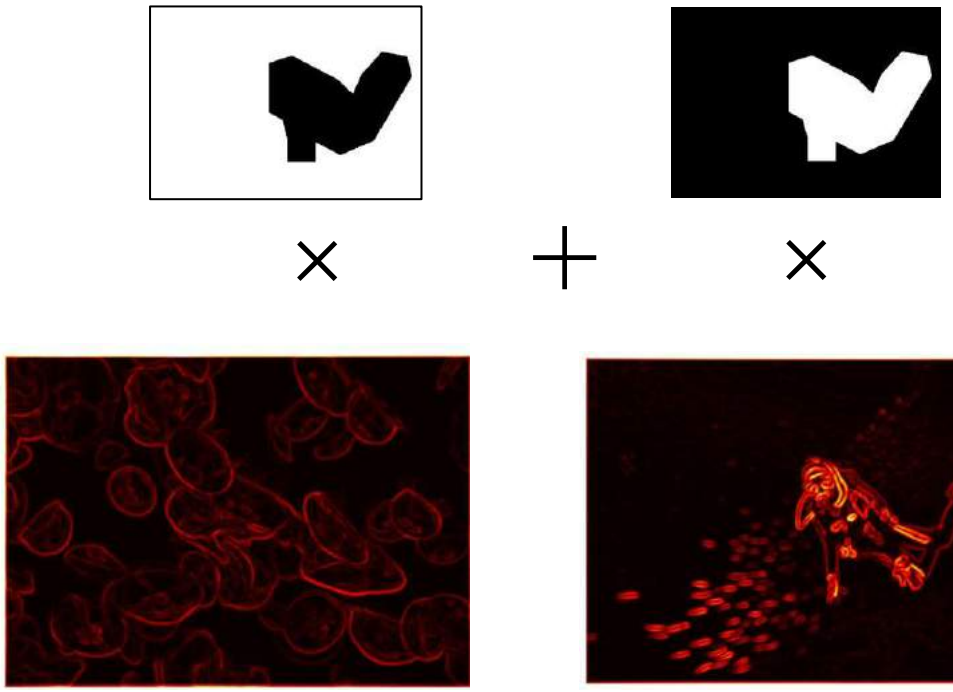
Gradient angle



Gradients domain

- Gradient captures everything important about shape and shading
- It contains the microscope texture of the object.
- It encodes subtle changes of illumination.
- In Pyramid Blending, we decomposed our image into 2nd derivatives (Laplacian) which encodes the shape perception.

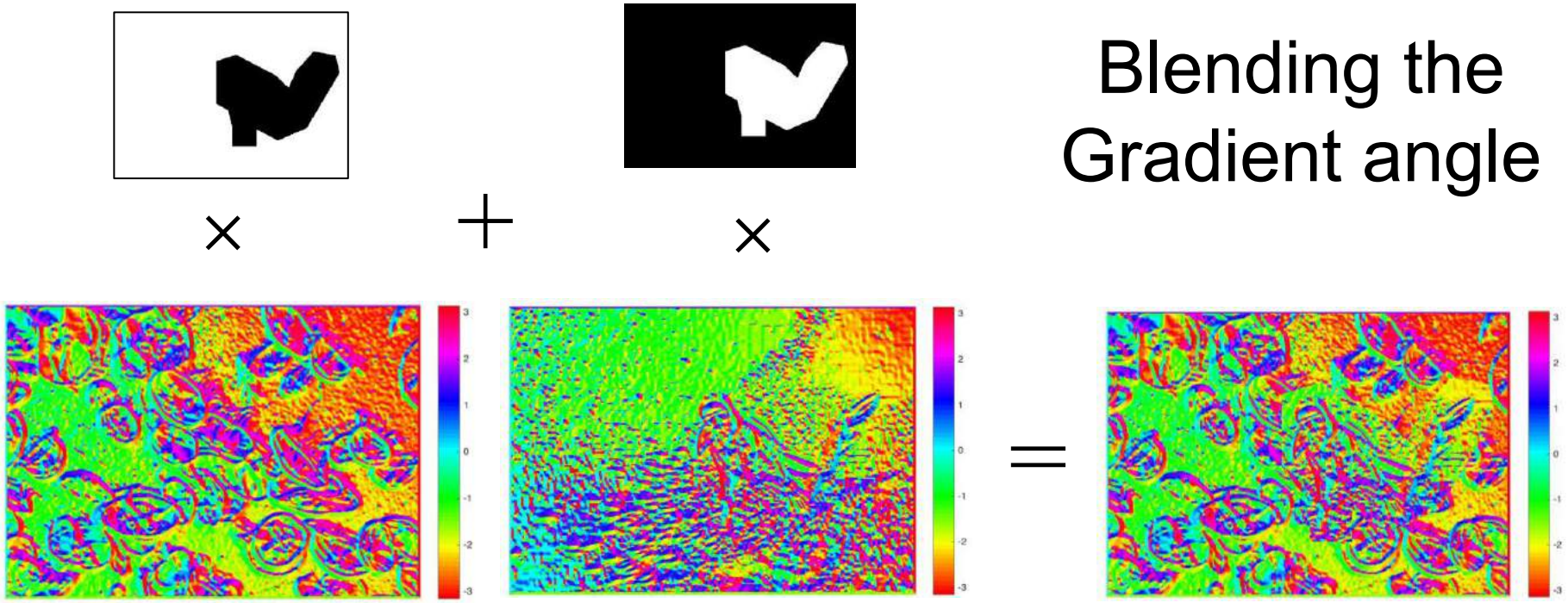



Blending Magnitude

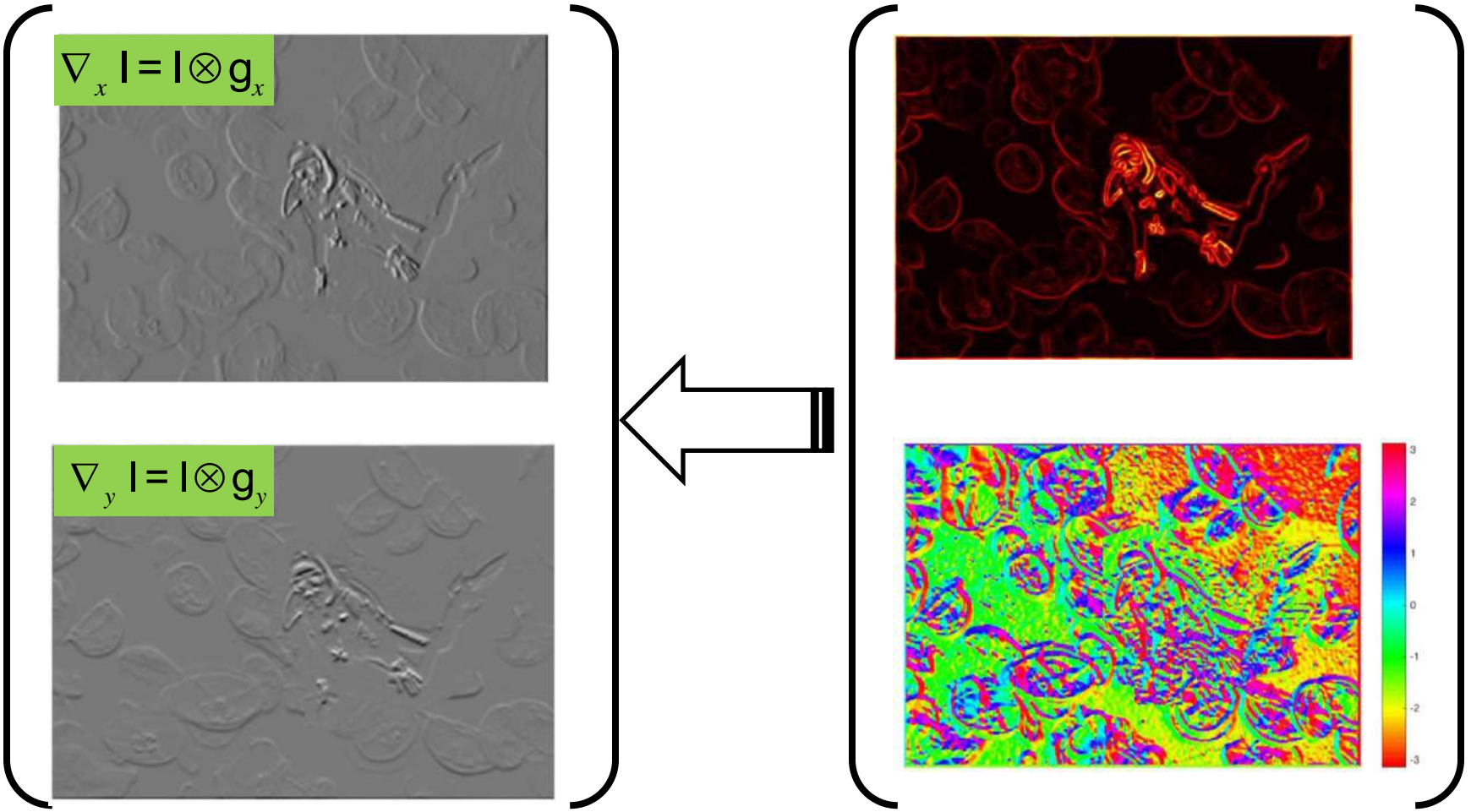
- We blend the gradient magnitude, to create a seamless ‘edge’ image

Blending the Gradient angle

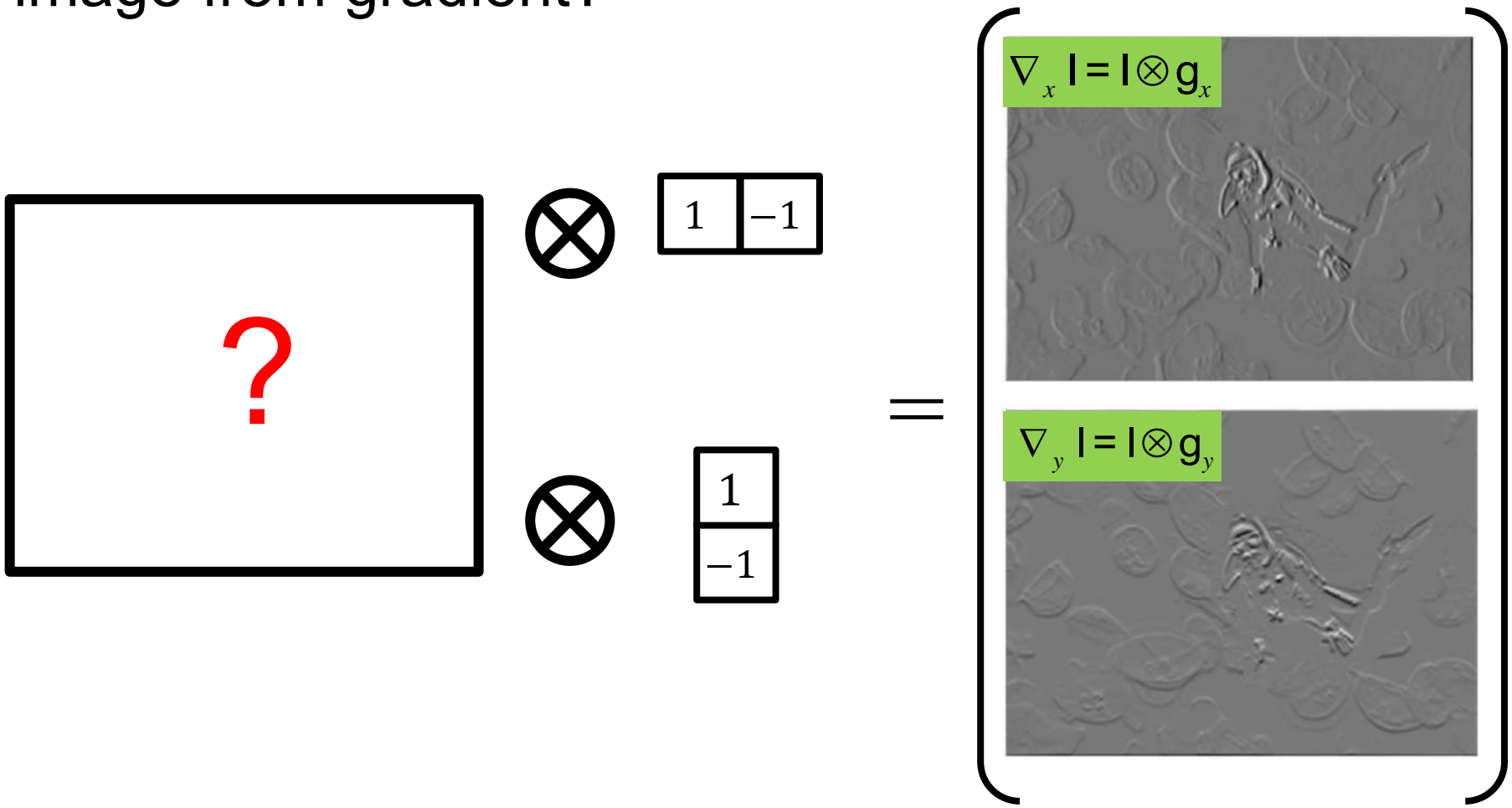


- We blend the gradient angle images, to make sure both the microscopic texture, shape of object boundary, and the illumination changes are smoothly integrated.

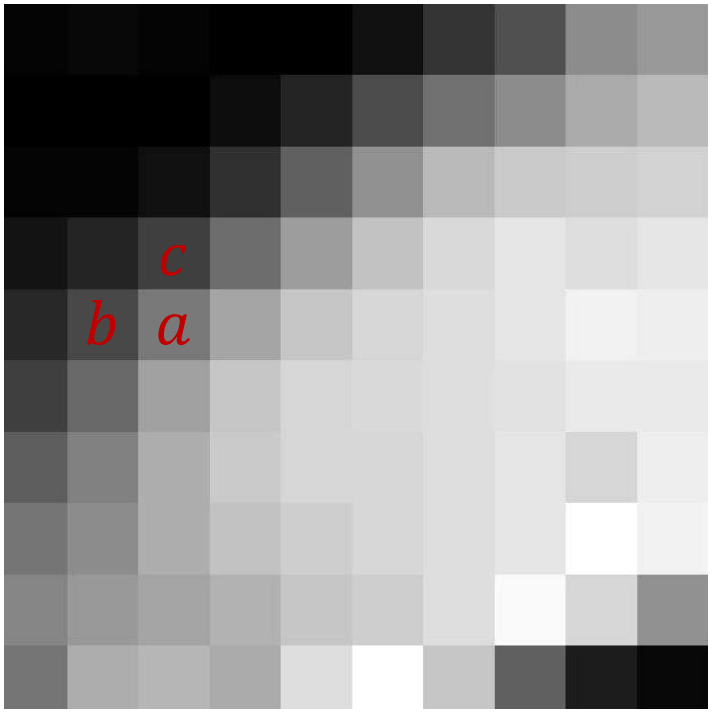
Image gradients blending



How to recreate the original image from gradient?



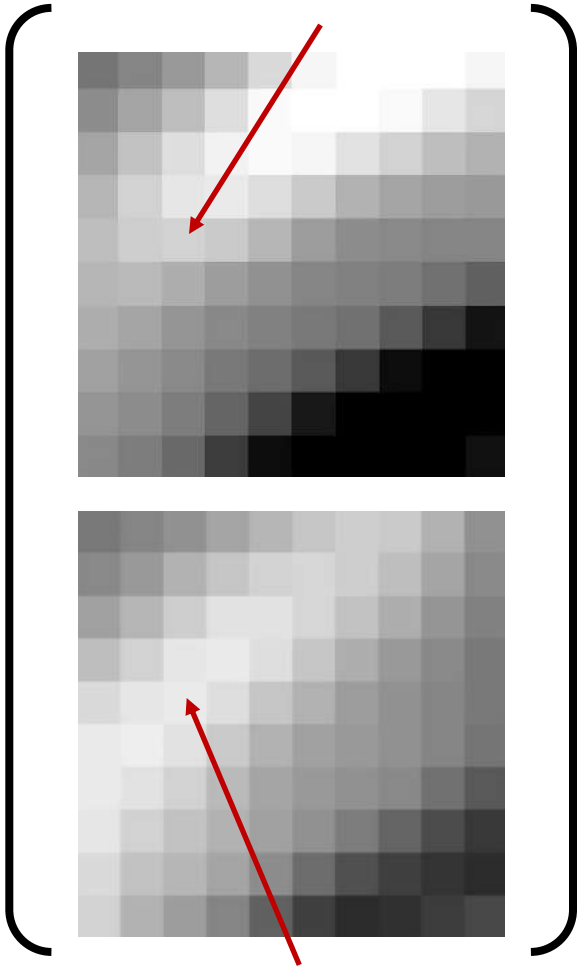
zoom in a small patch



$\otimes$ $\begin{array}{|c|c|}\hline 1 & -1 \\ \hline \end{array}$

$\otimes$ $\begin{array}{|c|}\hline 1 \\ \hline -1 \\ \hline \end{array}$

$\nabla_x I(a)$



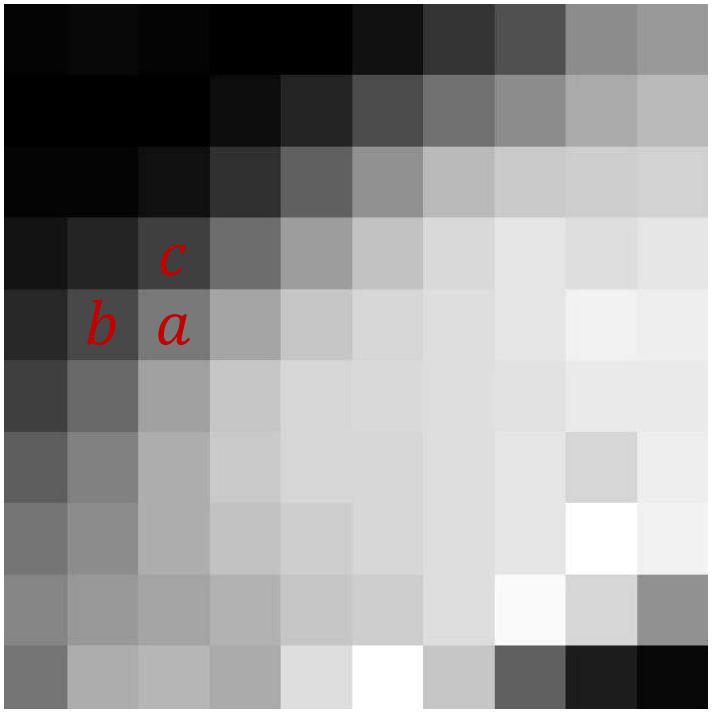
$\nabla_y I(a)$

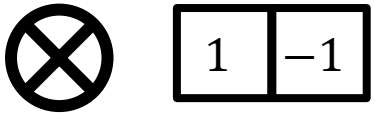
g

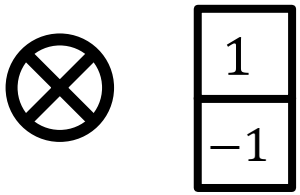
$a - b =$ $\nabla_x I(a)$

$a - c =$ $\nabla_y I(a)$

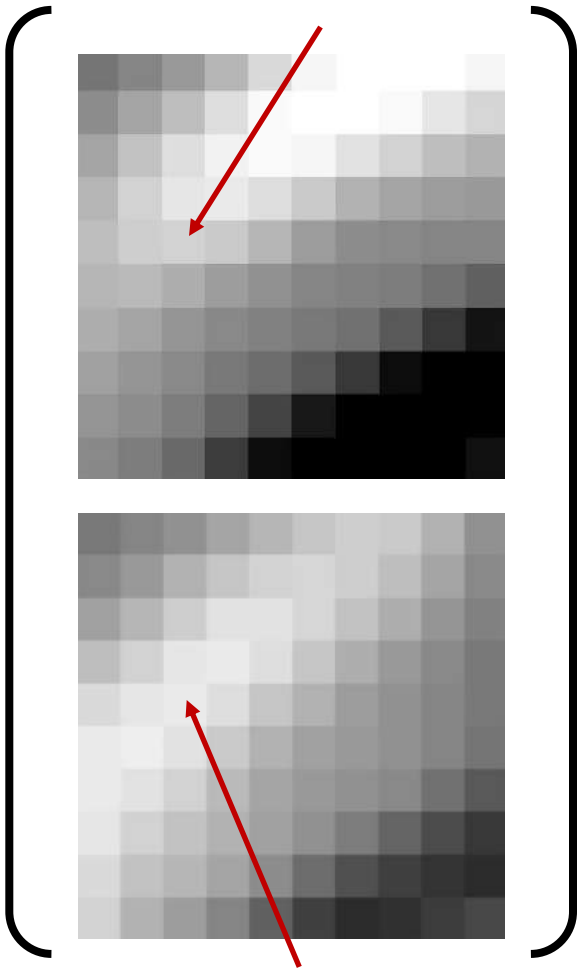
2 equations with 3 unknowns, need constraints





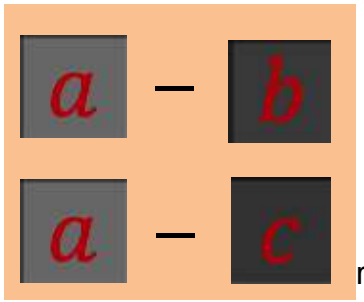


$\nabla_x I(a)$



$\nabla_y I(a)$

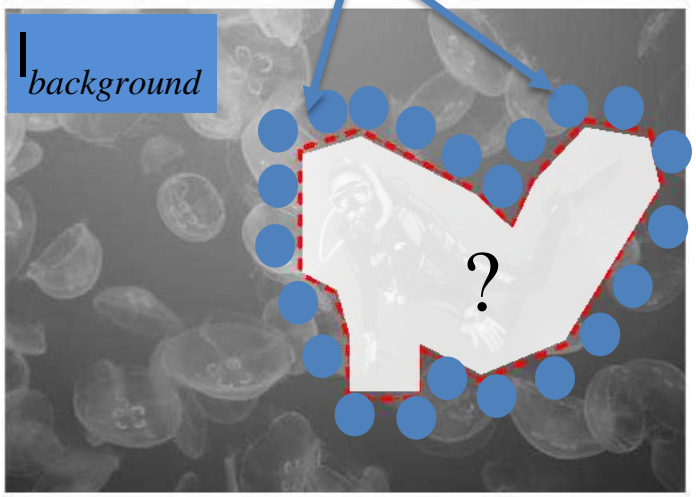
g



$= \nabla_x I(a)$

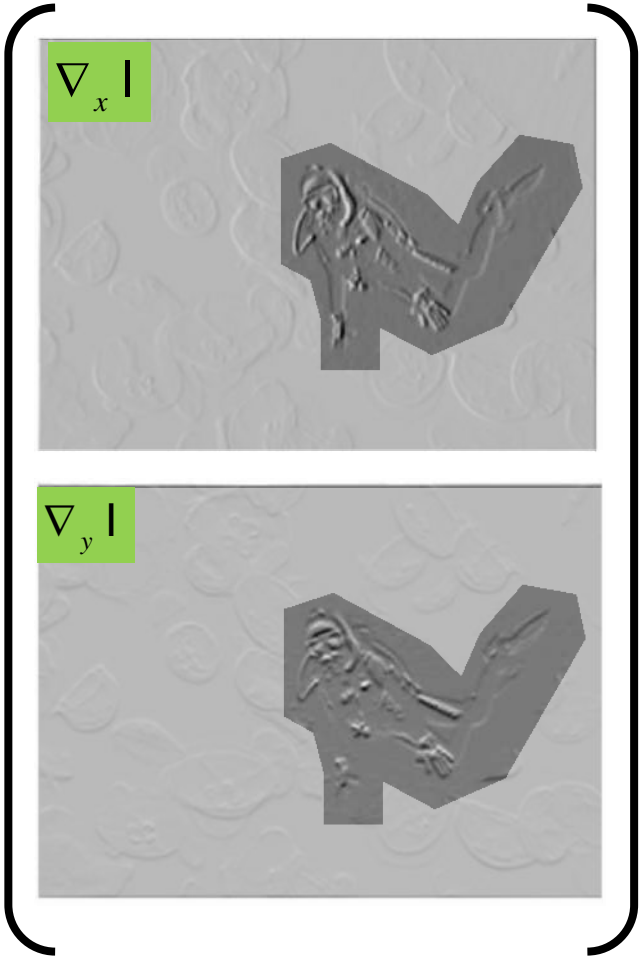
$= \nabla_y I(a)$

Boundary condition



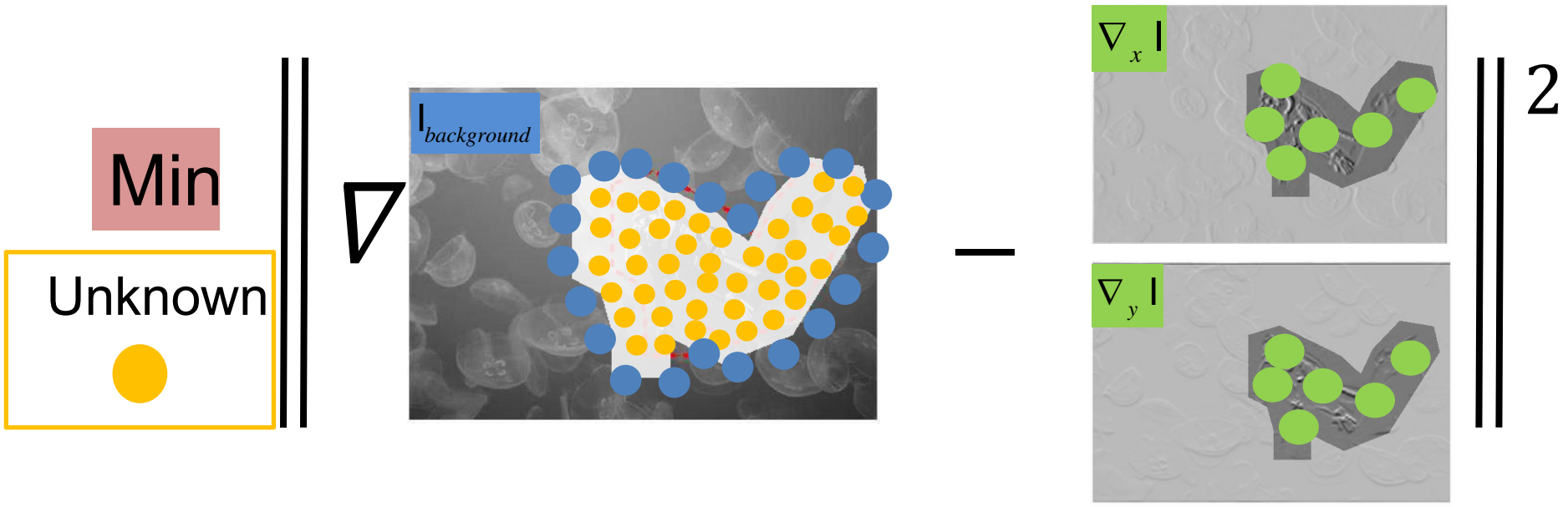
$\otimes$ $\begin{bmatrix} 1 & -1 \end{bmatrix}$ $=$

$\otimes$ $\begin{bmatrix} 1 \\ -1 \end{bmatrix}$

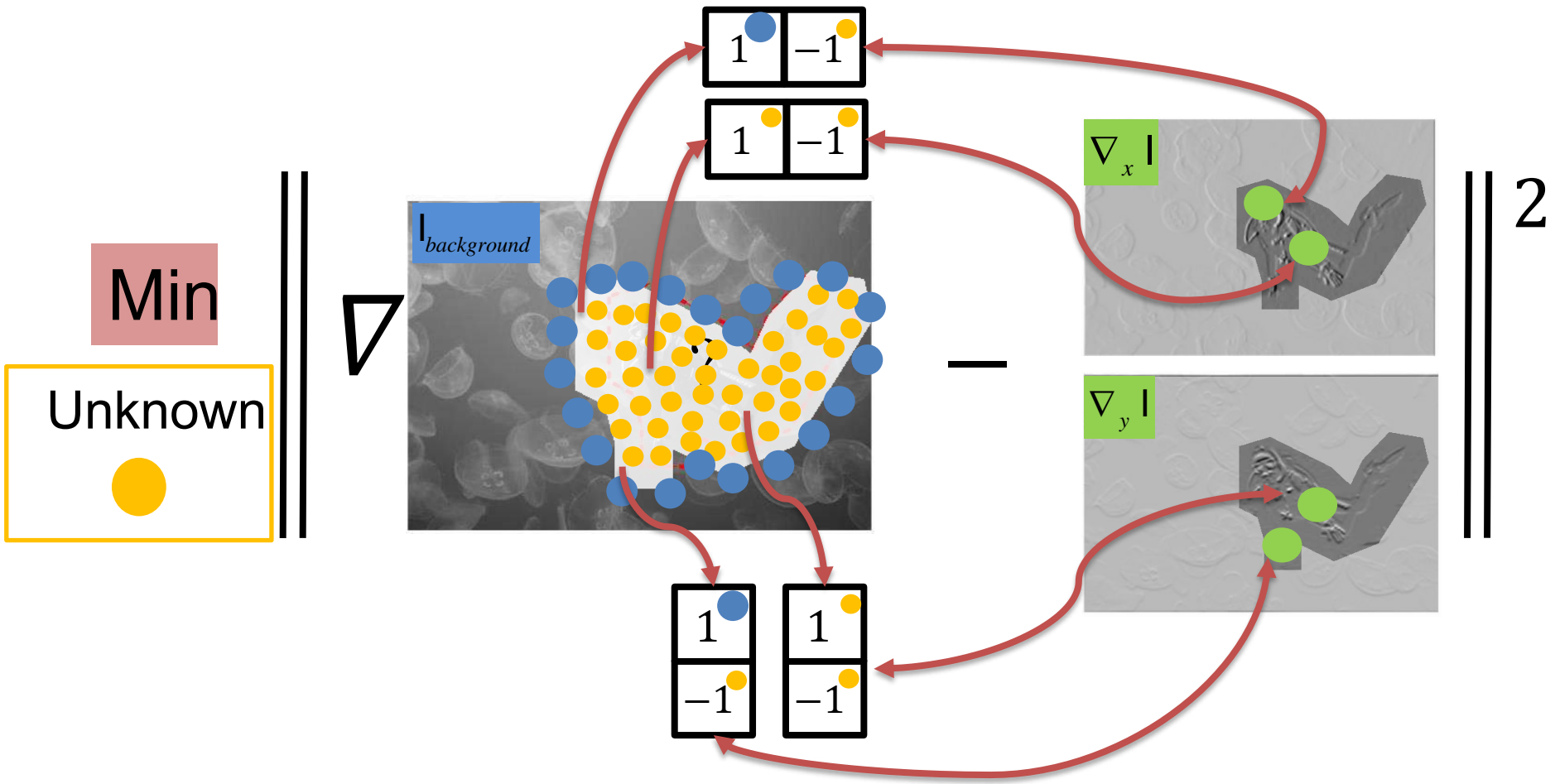


- Keep the value on boundary $\partial\Omega$ the same

Least Square Problem

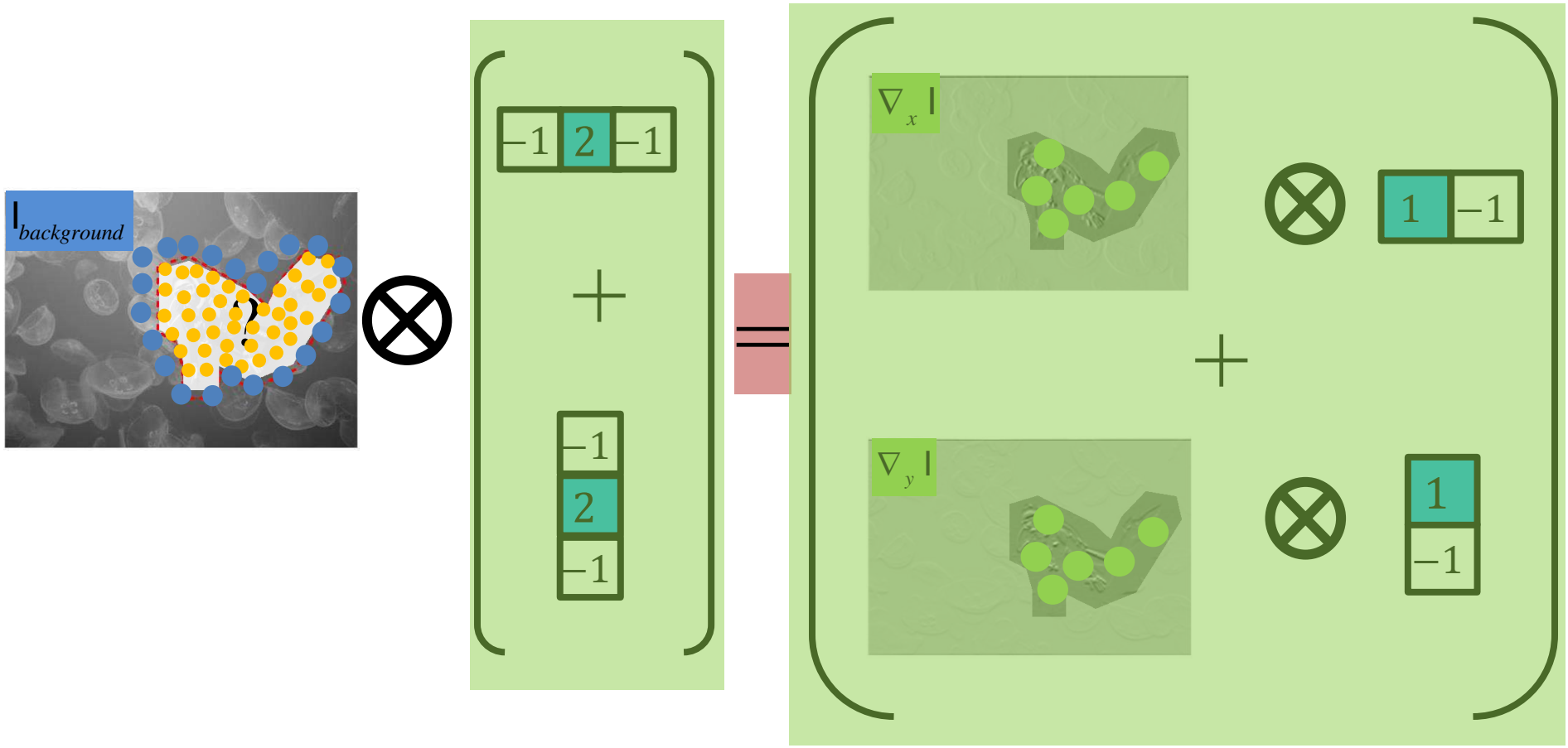


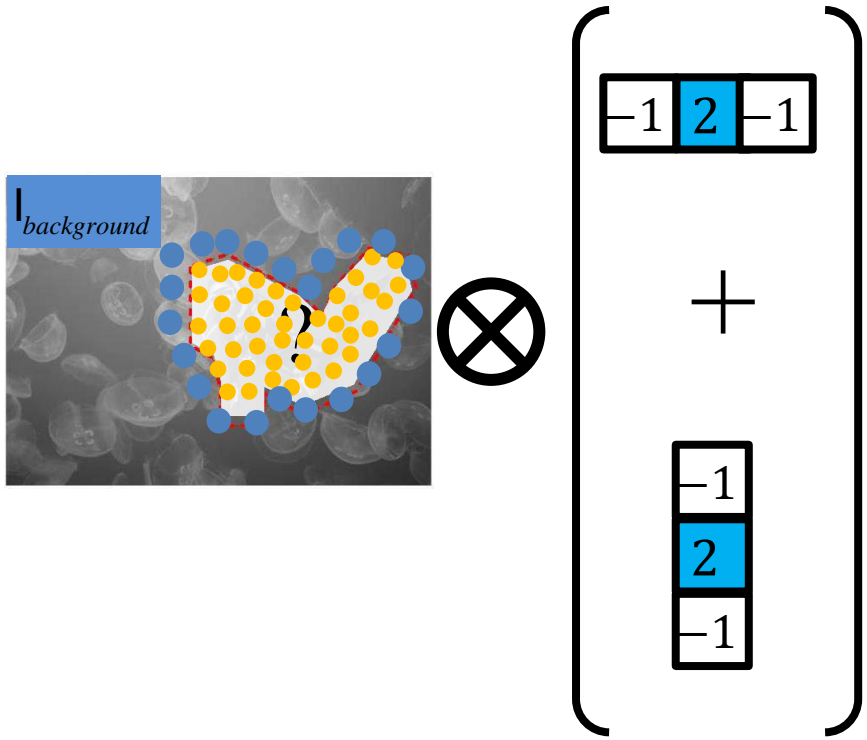
- Minimize the loss with respect to all pixels in the region Ω
- Keep the boundary $\partial\Omega$ the same

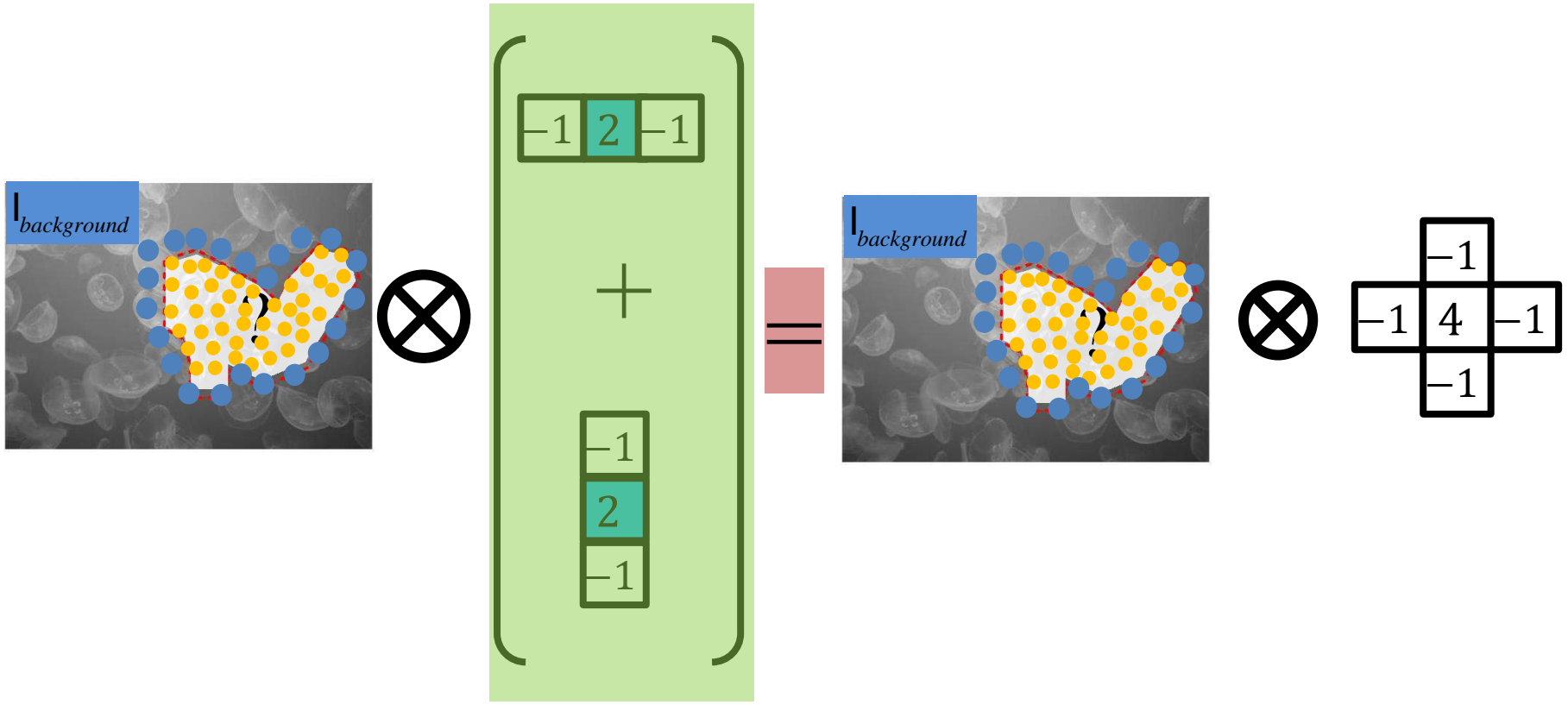


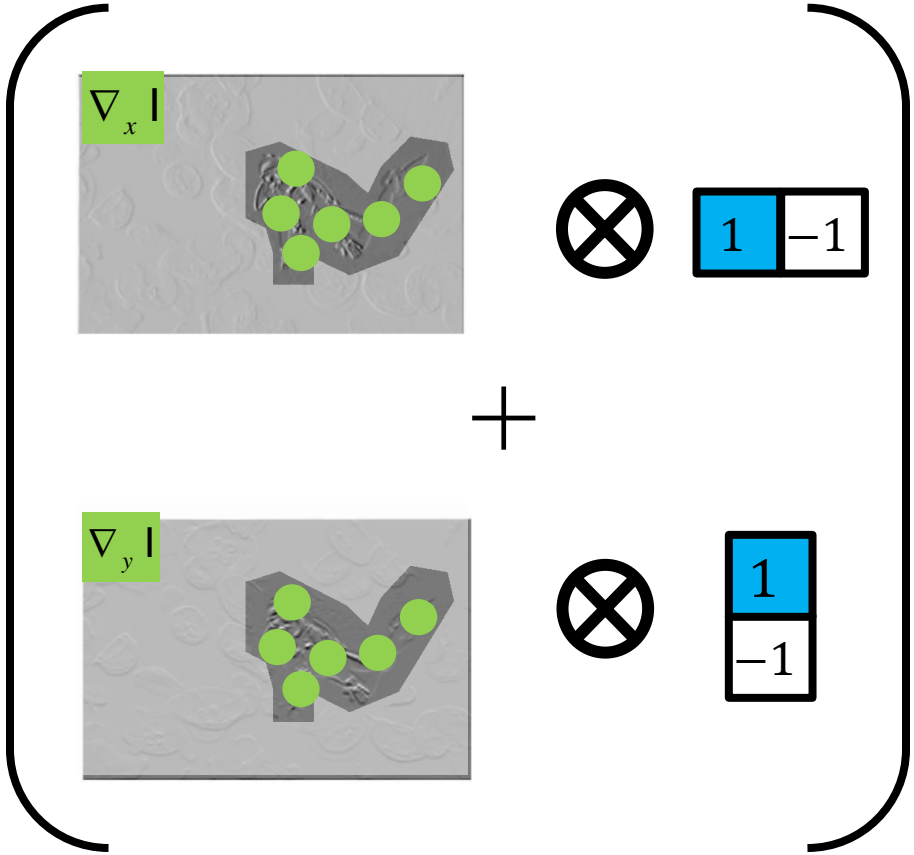
Video 8.3
Jianbo Shi

Least square solution

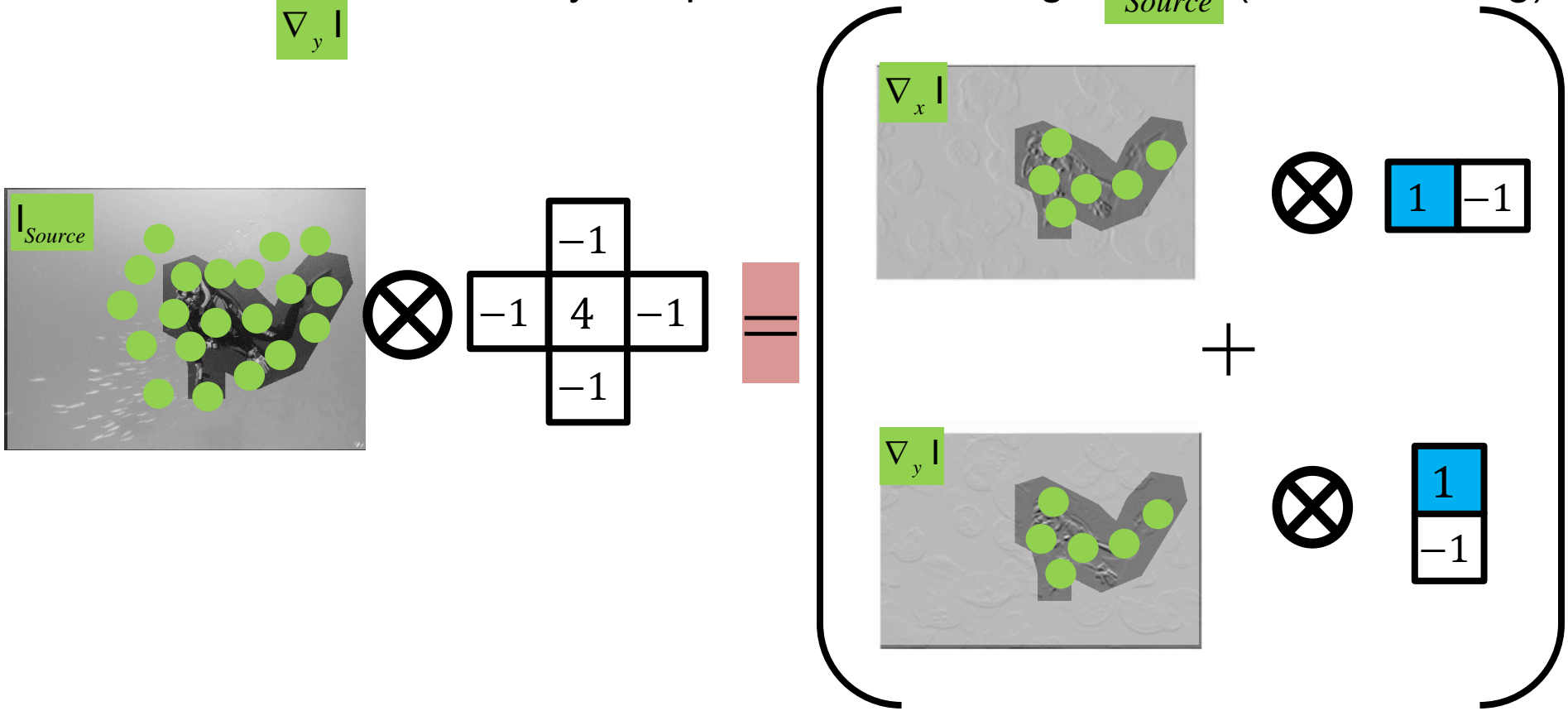




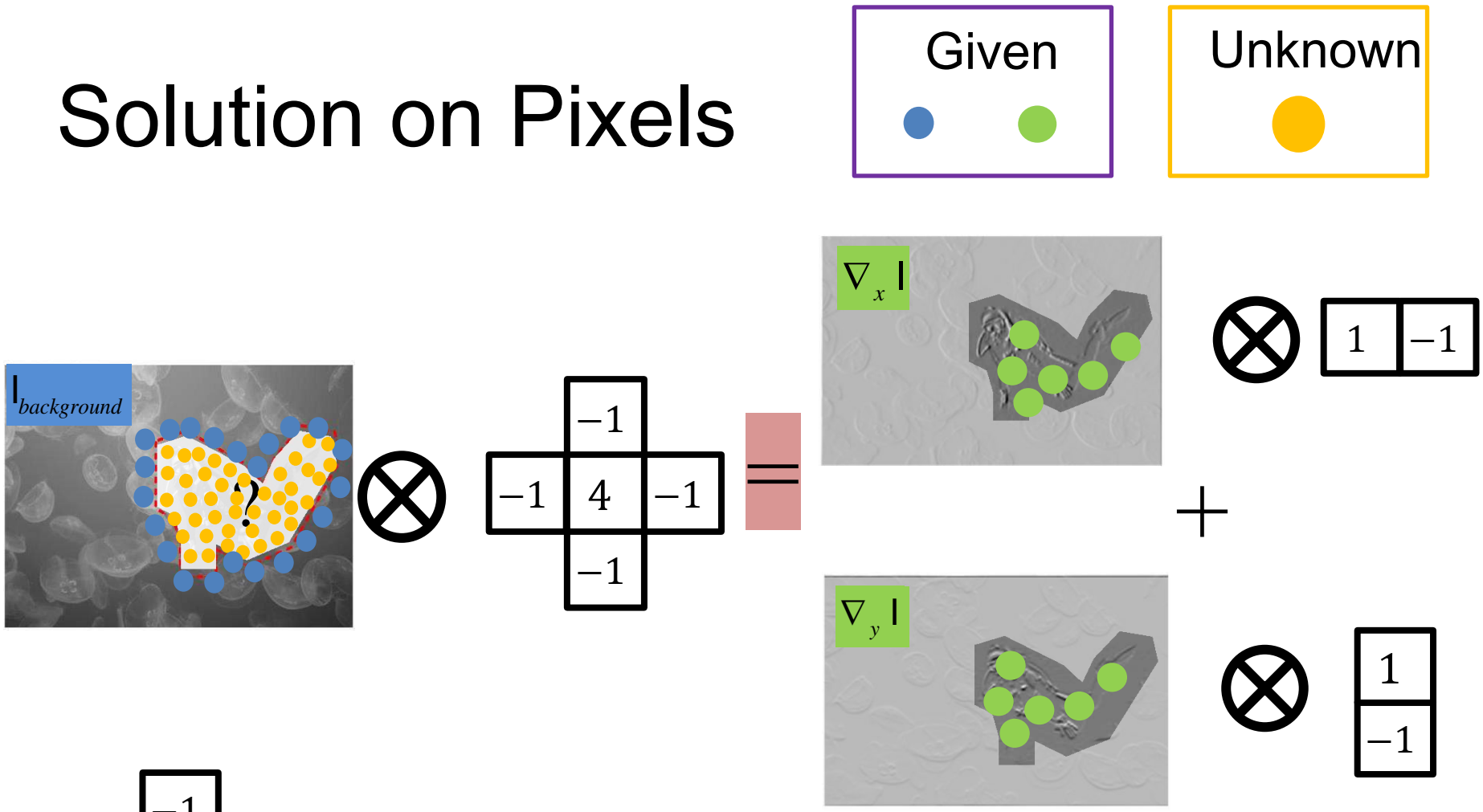




- When the $\nabla_x I$ / $\nabla_y I$ are directly computed from an image I_{Source} (without editing)



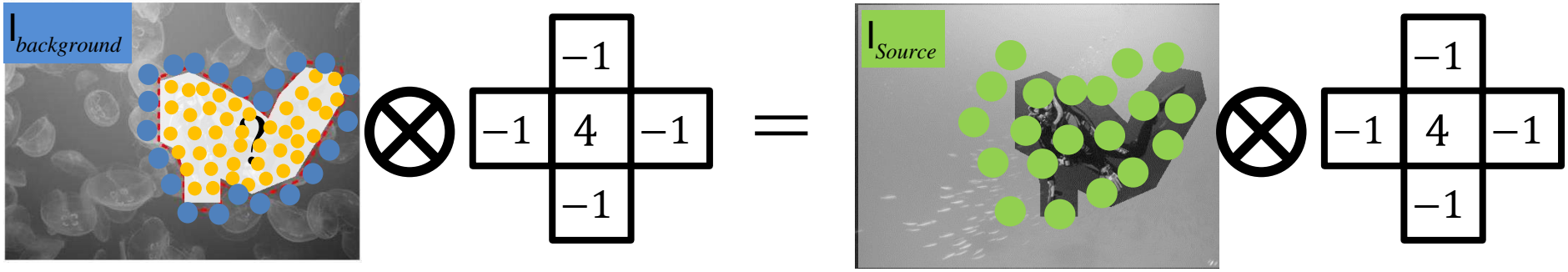
Solution on Pixels



- The convolutional kernel for Laplacian operator

Special case

- When the $\nabla_x I$ / $\nabla_y I$ are directly computed from an image I_{Source} (without editing)

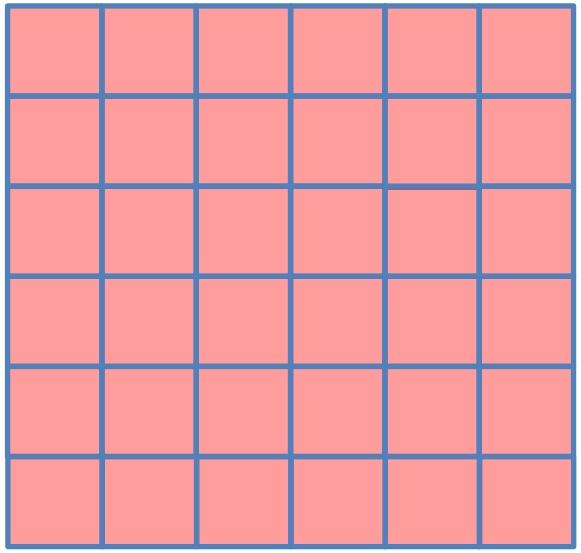


- Keep the value on the boundary $\partial\Omega$ the same
- Solve the equation for each channel (RGB) separately

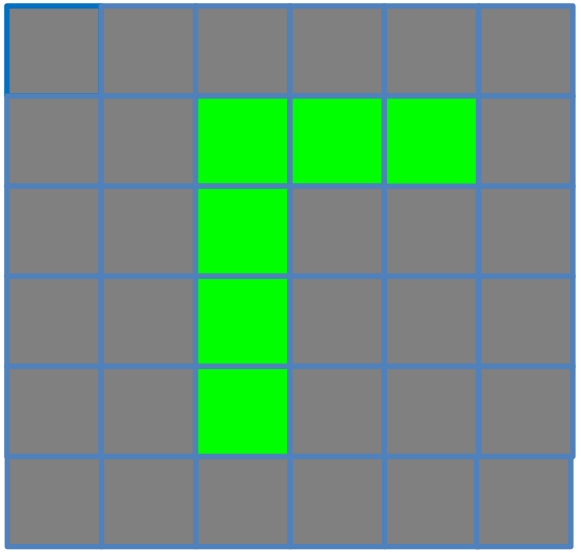
- There is one equation matching the Laplacian values from the source to the target

Video 8.4
Jianbo Shi

An example for gradient blending



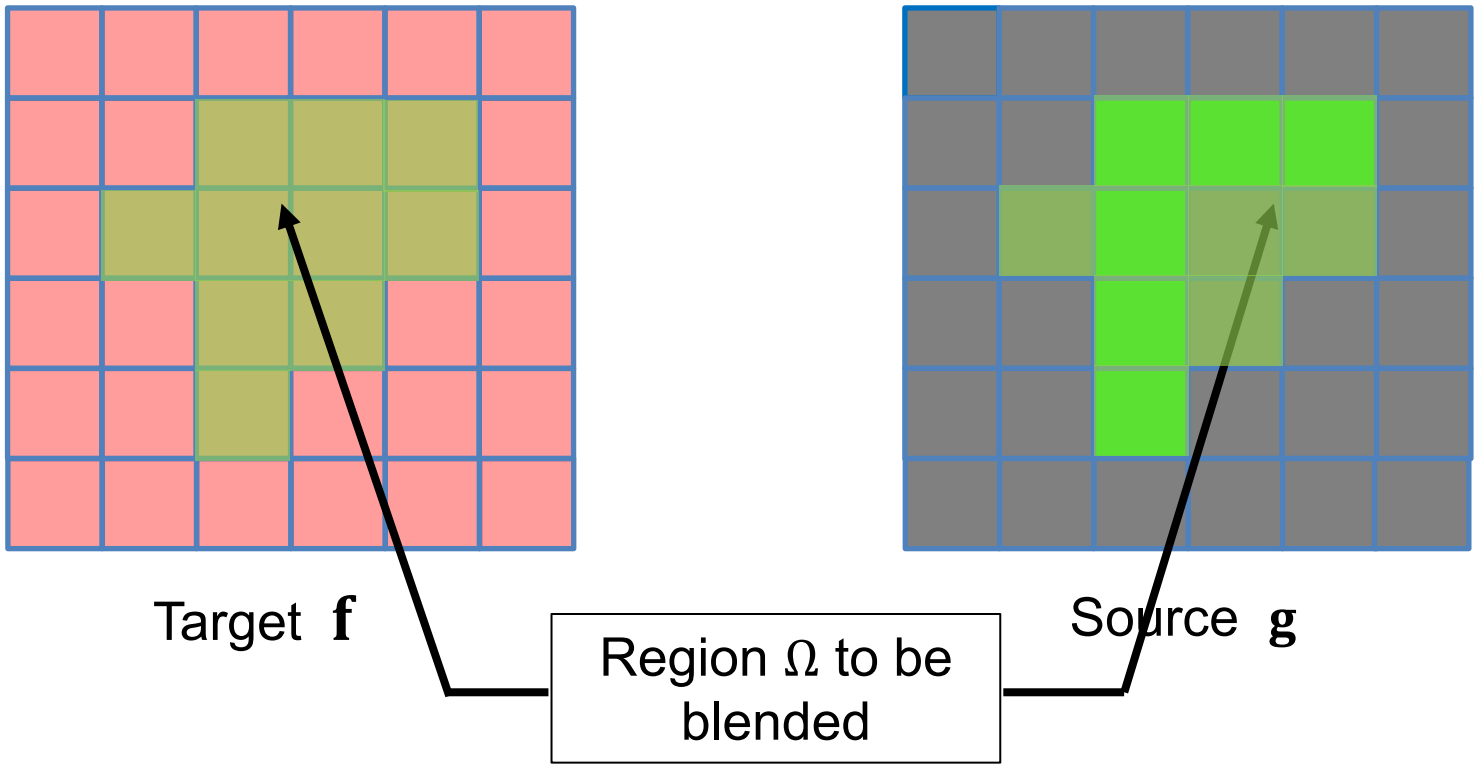
Target $\mathbf{f}$



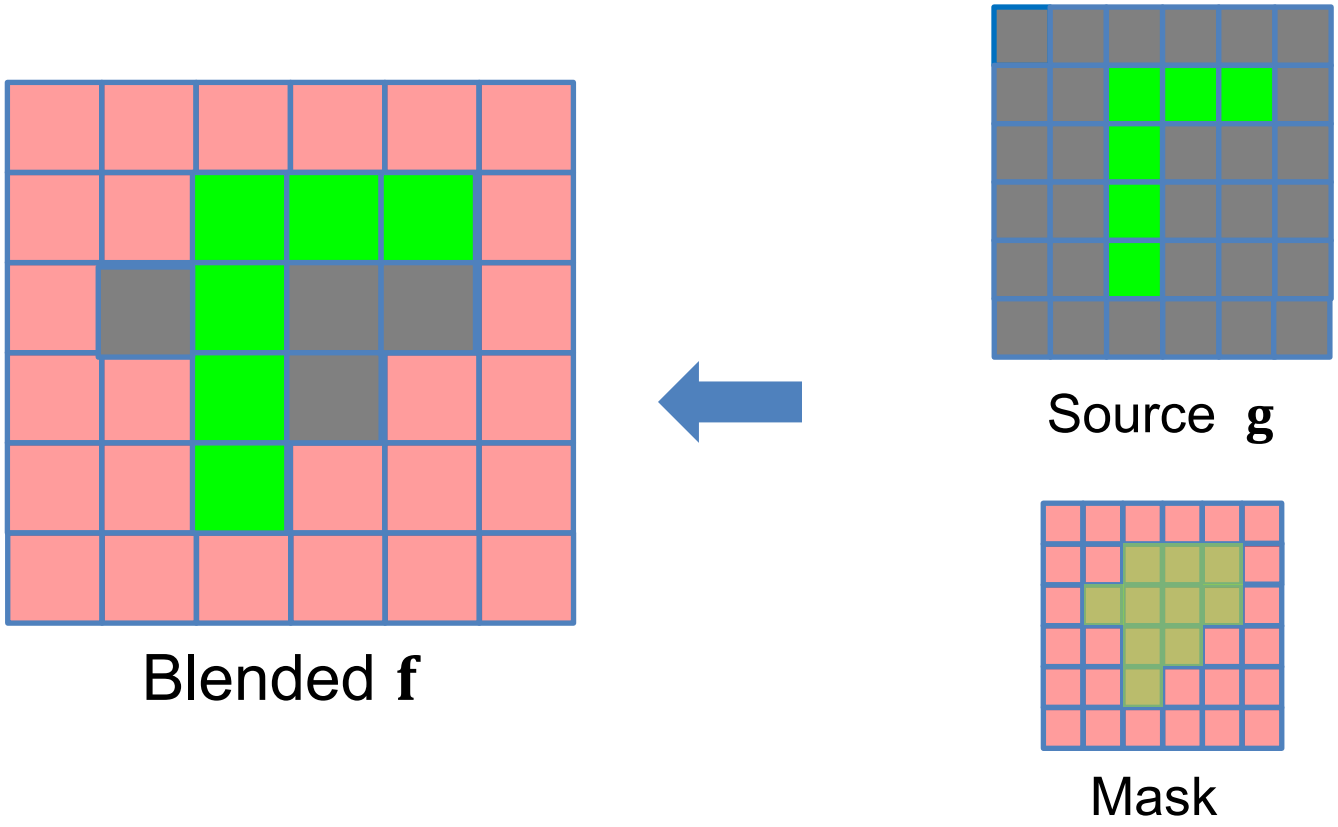
Source $\mathbf{g}$

The RGB value for f is $(255,0,0)$, background of g is $(0,0,0)$

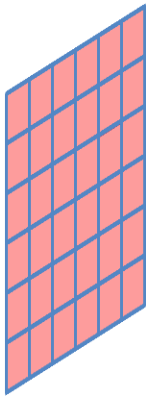
Mask for blending



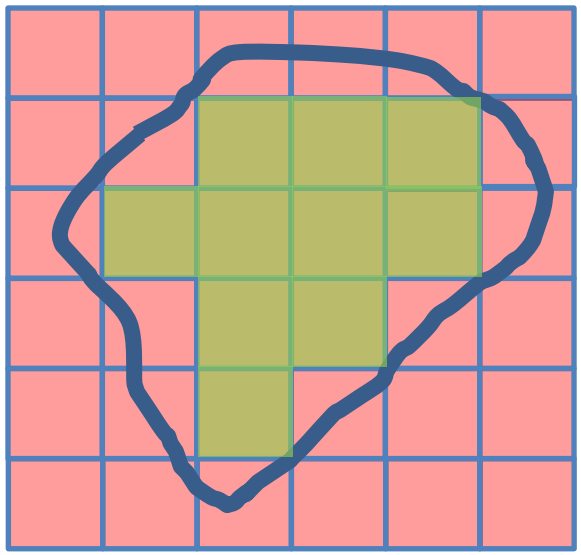
Direct copy and paste



Goal: recreate values in the mask with the two constraints

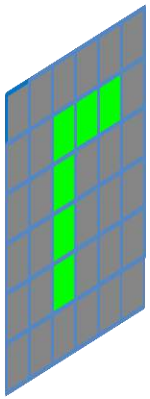


Target $\mathbf{f}$

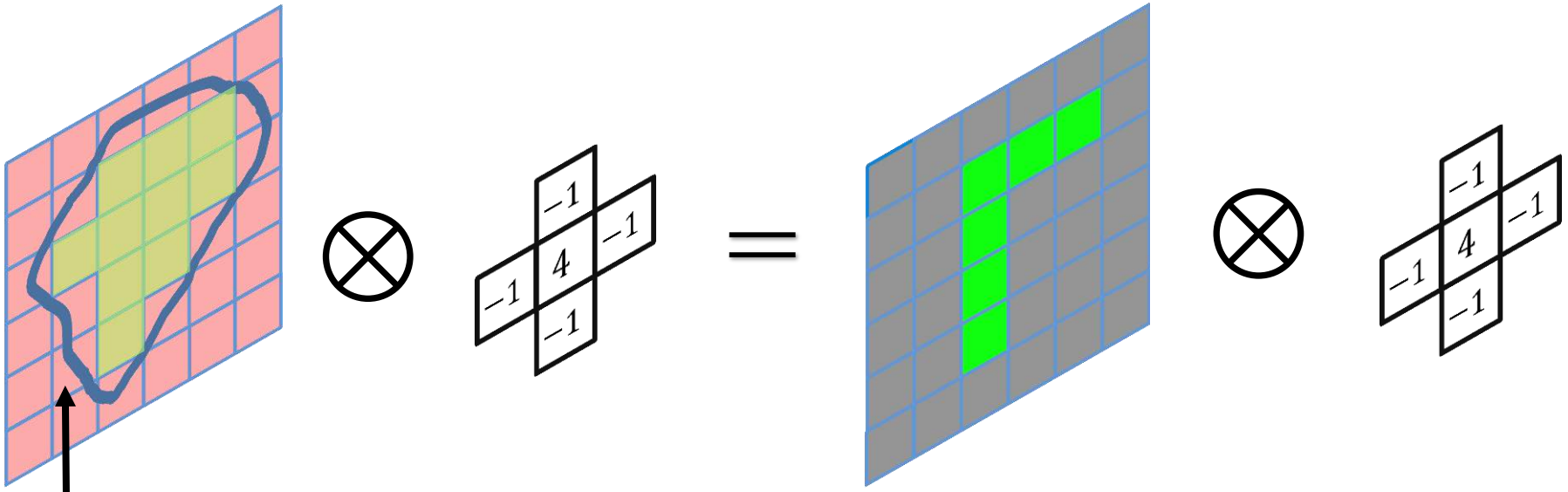


Mask Ω

$$\begin{cases} \Delta \mathbf{f} = \mathbf{b} \text{ in } \Omega \\ \mathrm{f}|_{\partial\Omega} \text{ keeps same} \end{cases}$$



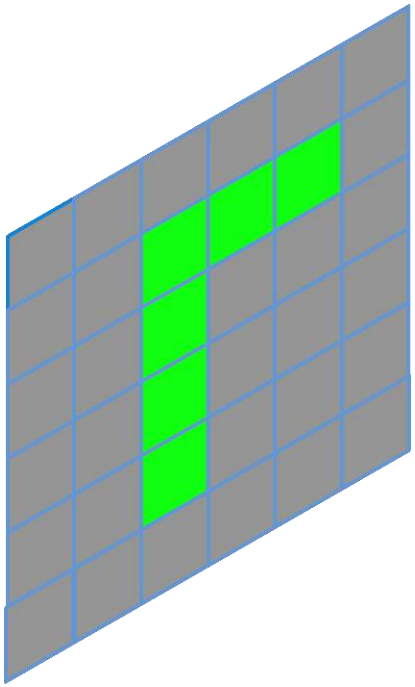
Source $\mathbf{g}$



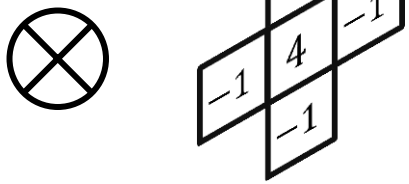
Region boundary $\partial\Omega$

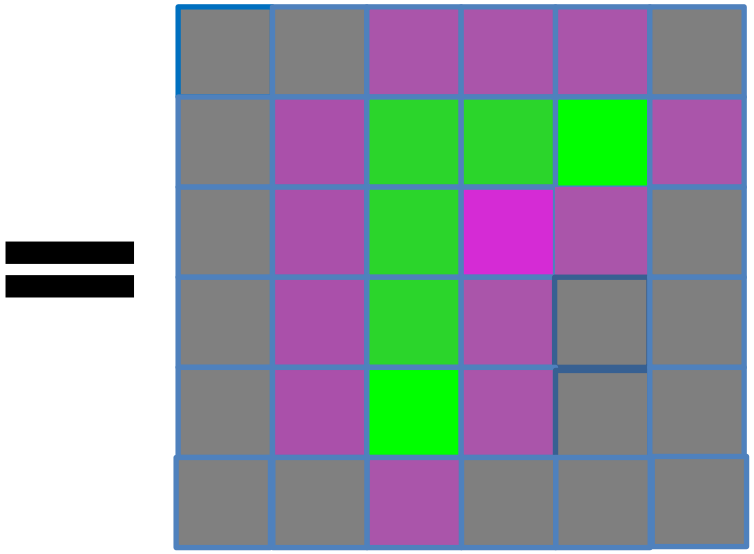
Solution: recreate values in the mask +
Keep f the same on the boundary $\partial\Omega$

Laplacian of the source



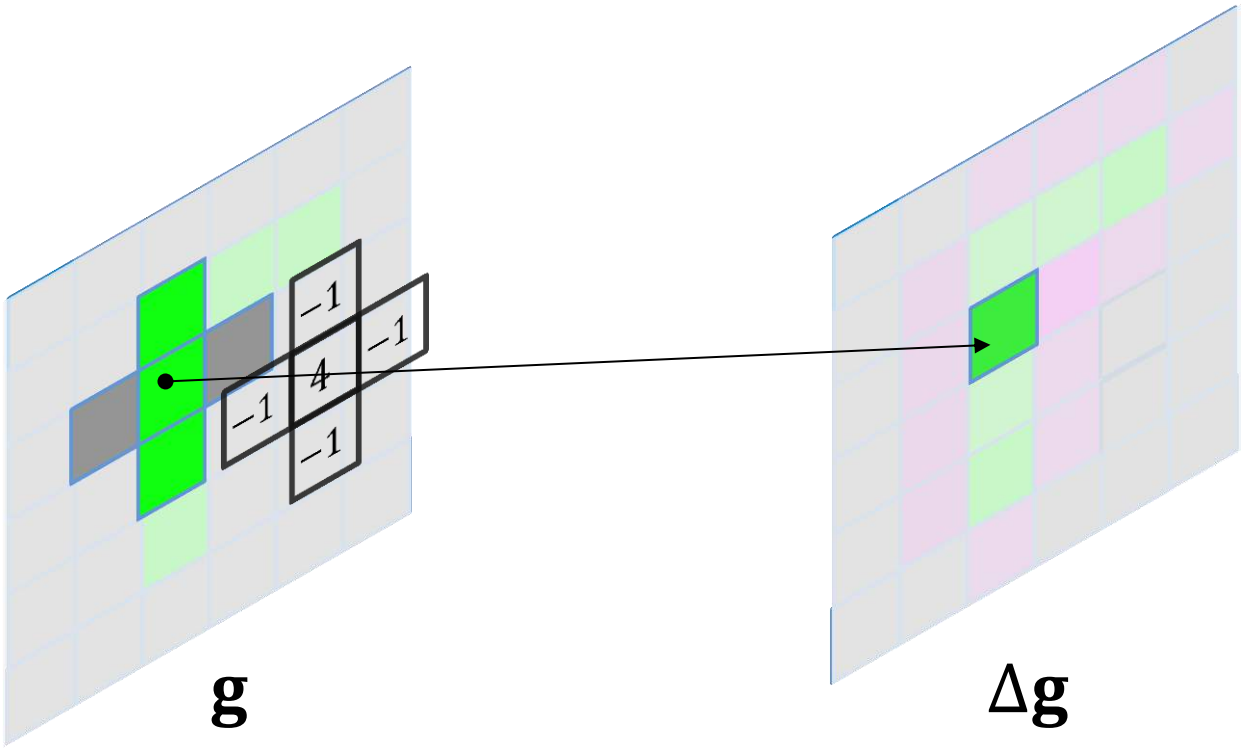
Source $\mathbf{g}$



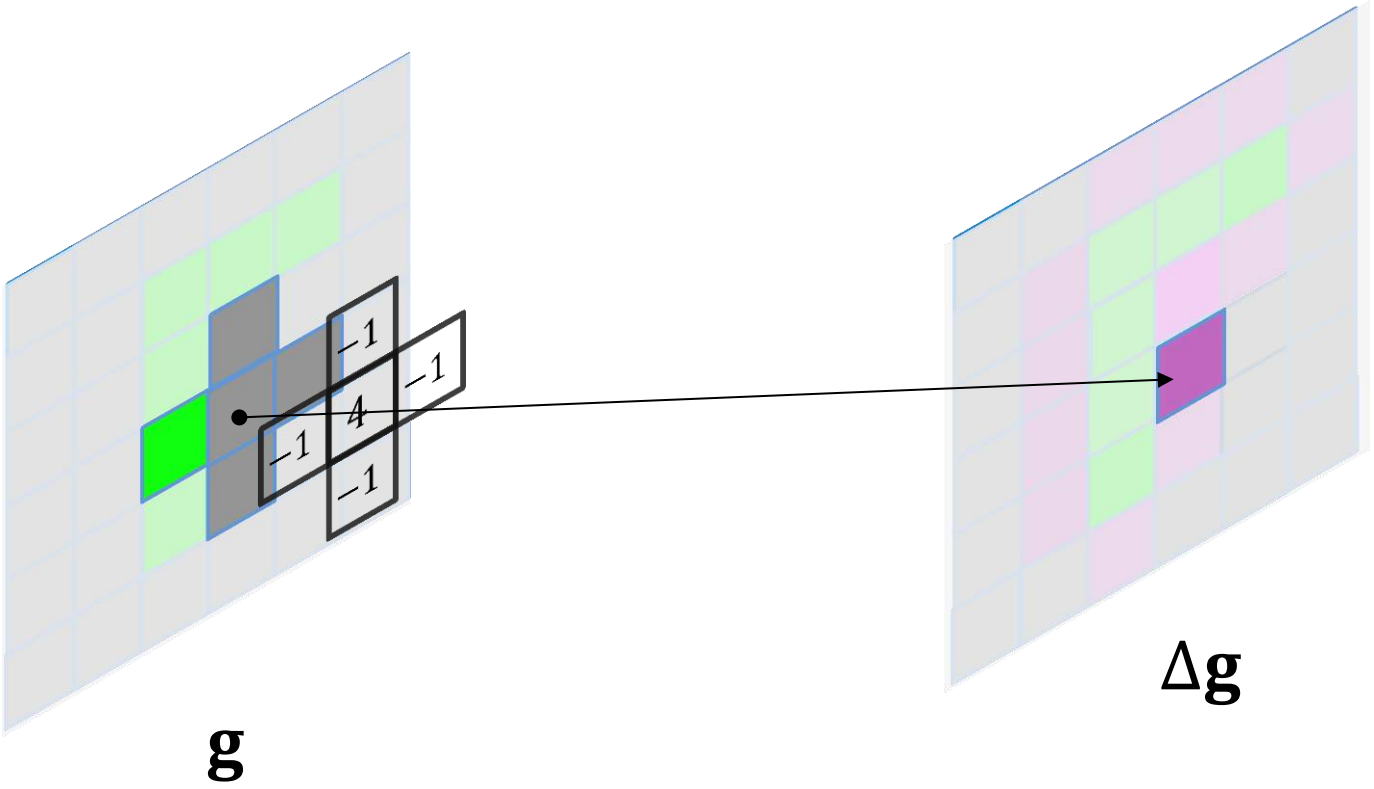


$\Delta \mathbf{g}$

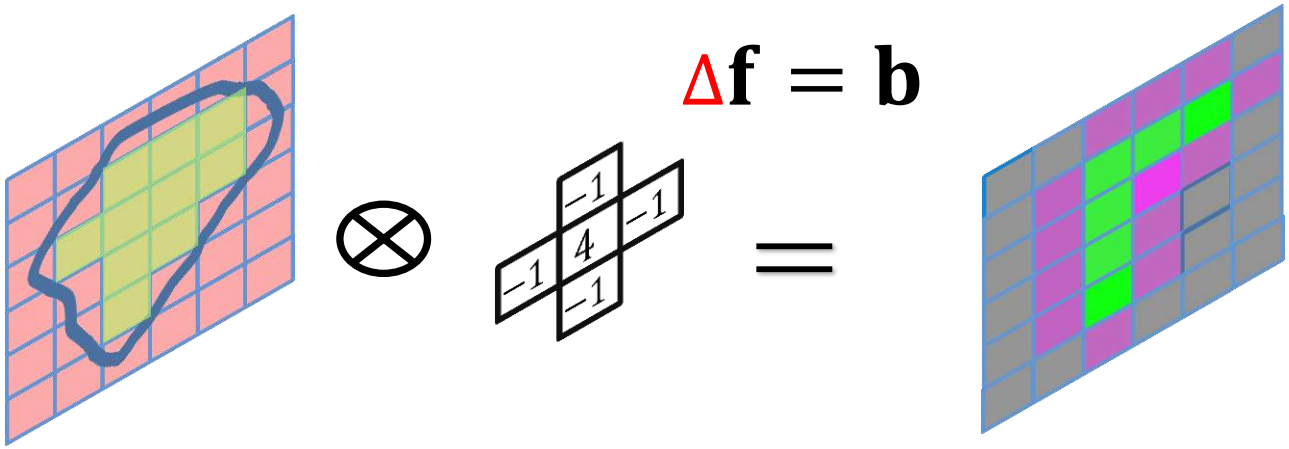
Laplacian of the source



Laplacian of the source

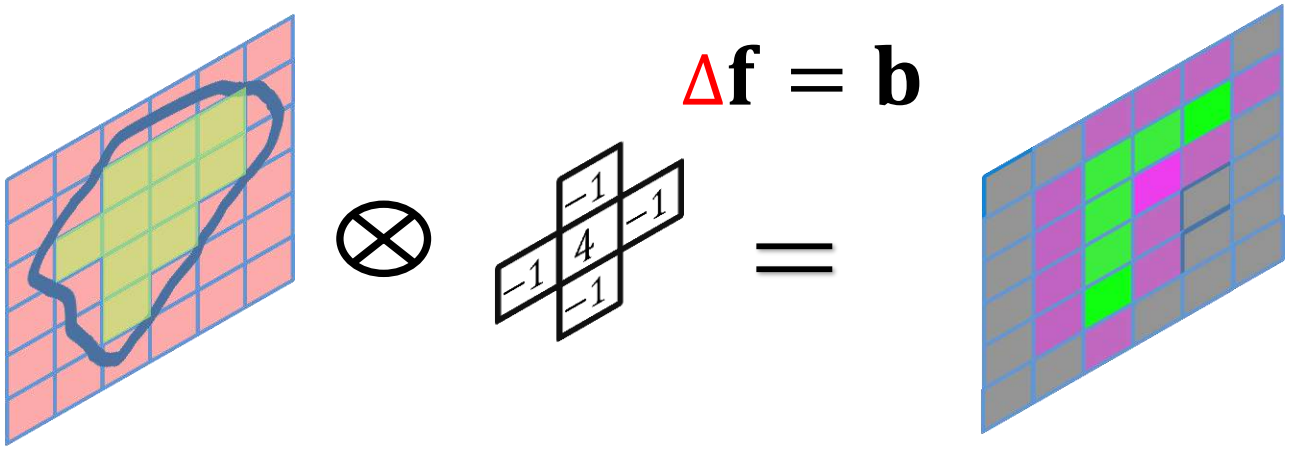


$$\Delta \mathbf{f} = \mathbf{b}$$



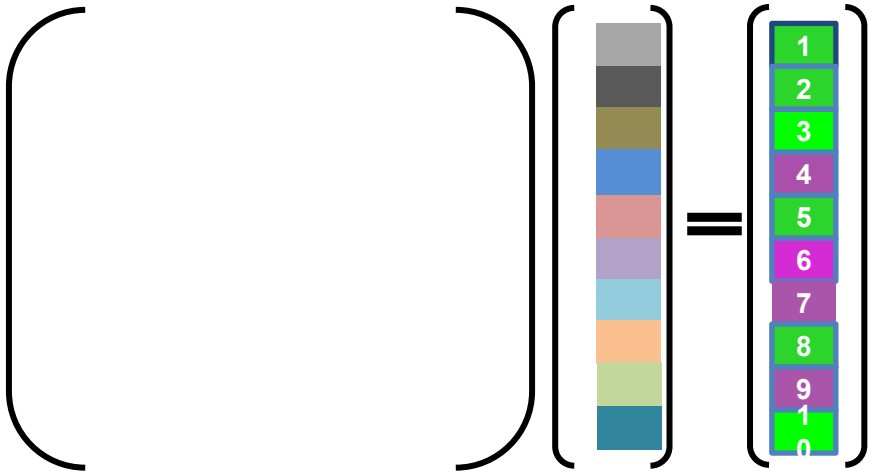
we have per pixel equation for the blended 2D image

$$\Delta \mathbf{f} = \mathbf{b}$$



we have per pixel equation for the blended 2D image
…will rewrite them as Matrix-Vector equation

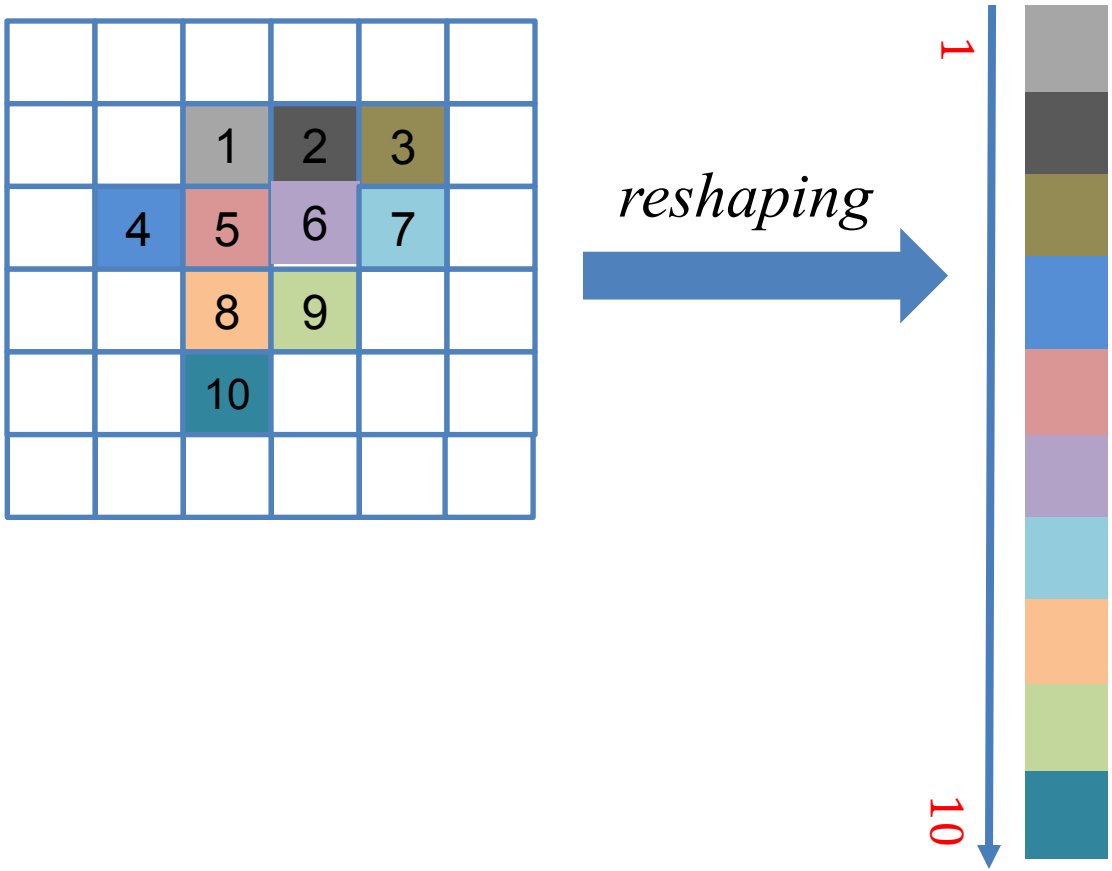
$$\mathbf{A}\mathbf{f} = \mathbf{b}$$



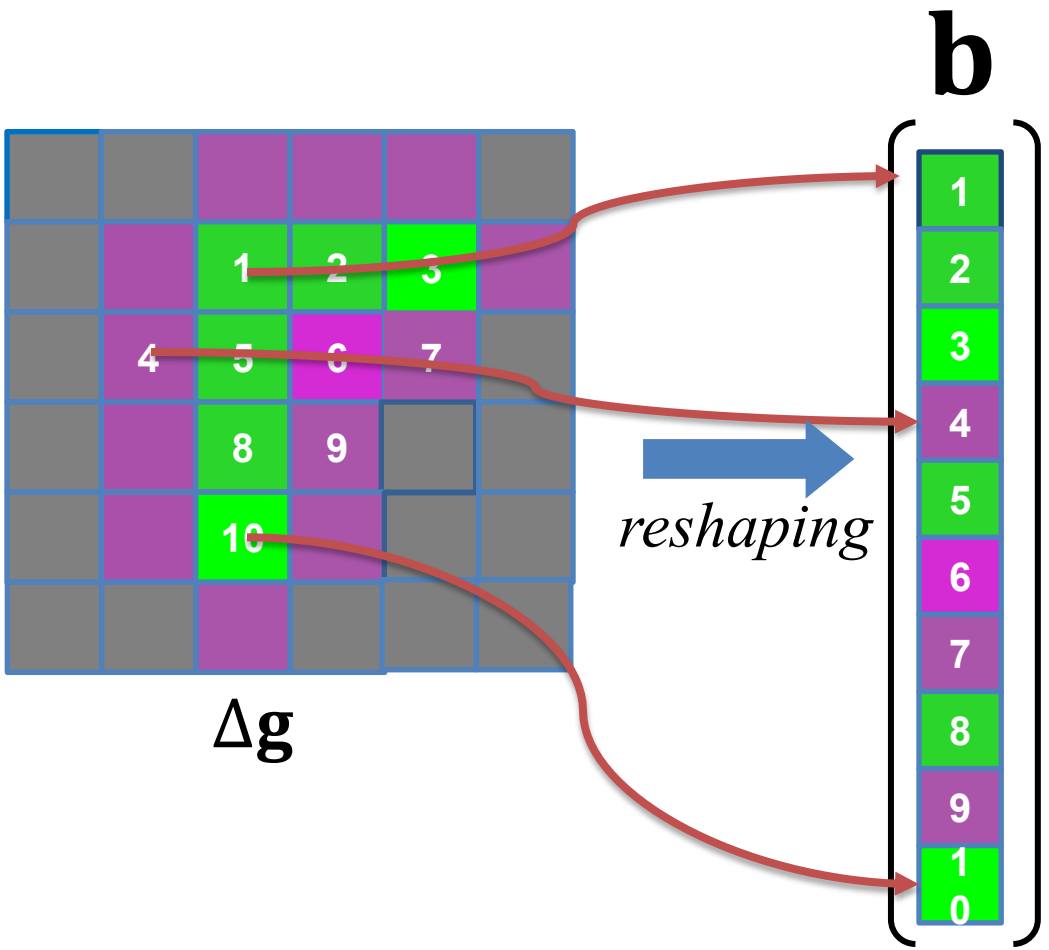
$$\mathbf{A}\mathbf{f} = \mathbf{b}$$

Think 2D image as a 1D vector

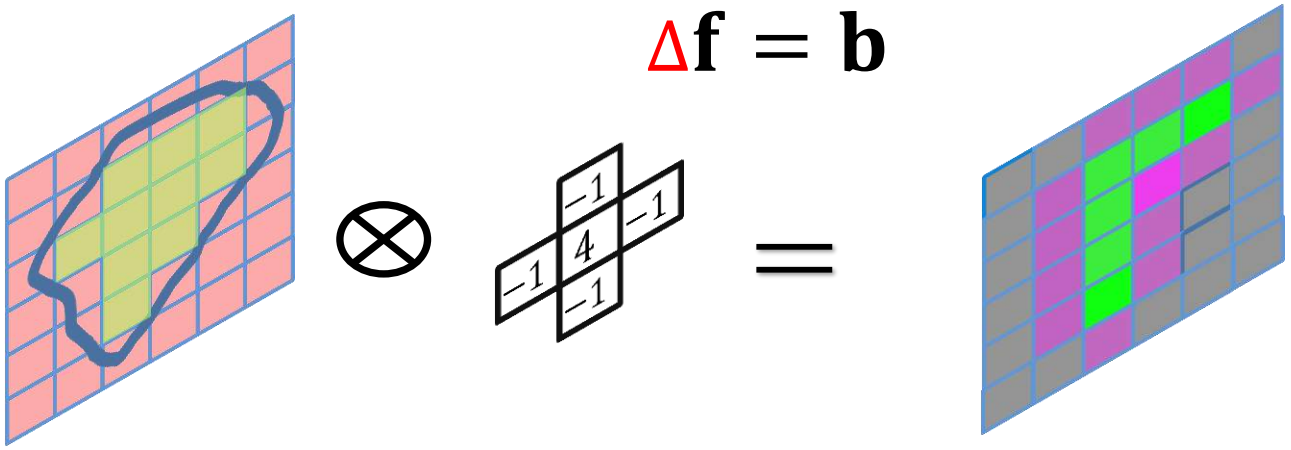
For each pixel in the mask,
we first create a 1D indexing vector



Copying and *reshaping* the Laplacian of source

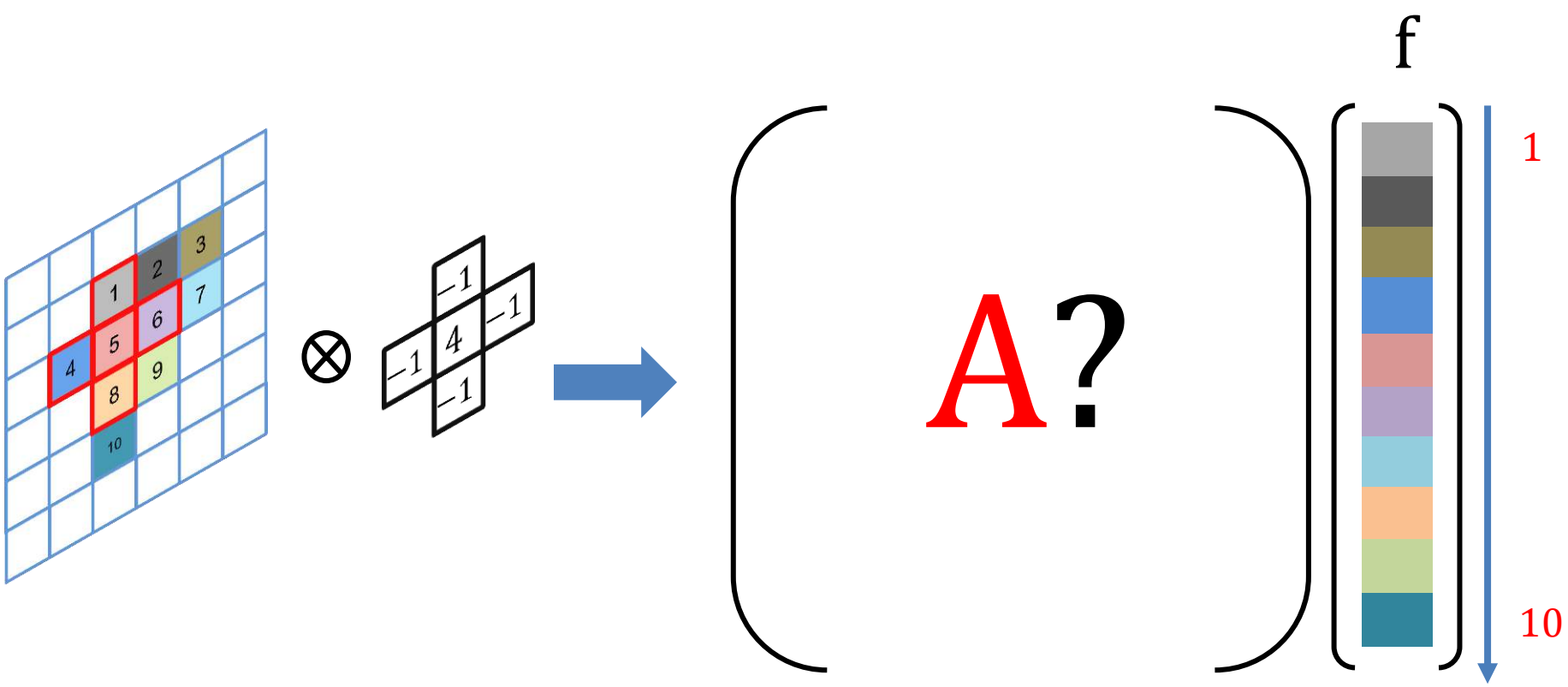


$$\Delta \mathbf{f} = \mathbf{b}$$

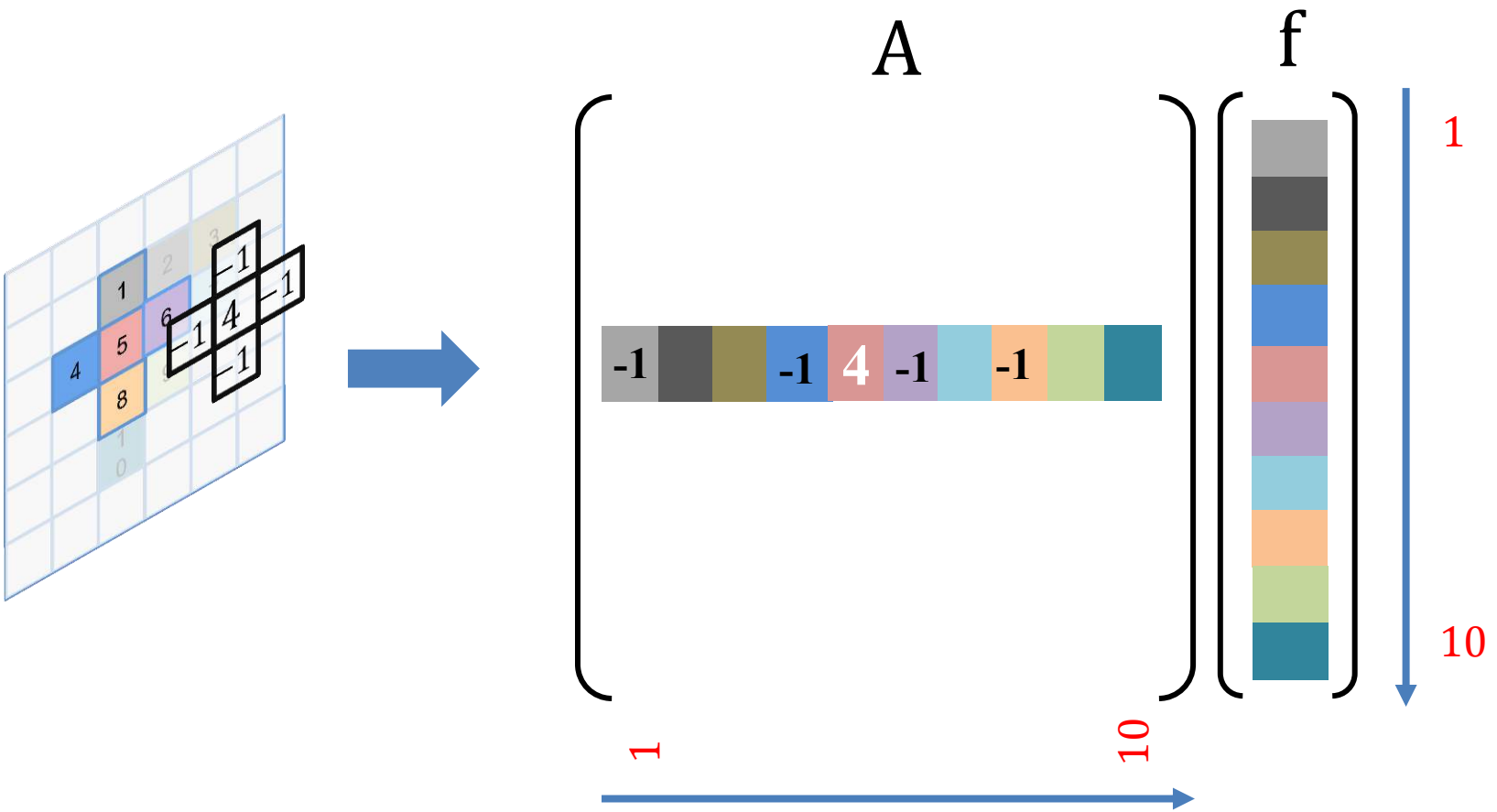


We can use a matrix $\mathbf{A}$ to encode 2D Convolution!

$$\mathbf{A}\mathbf{f} = \mathbf{b}$$

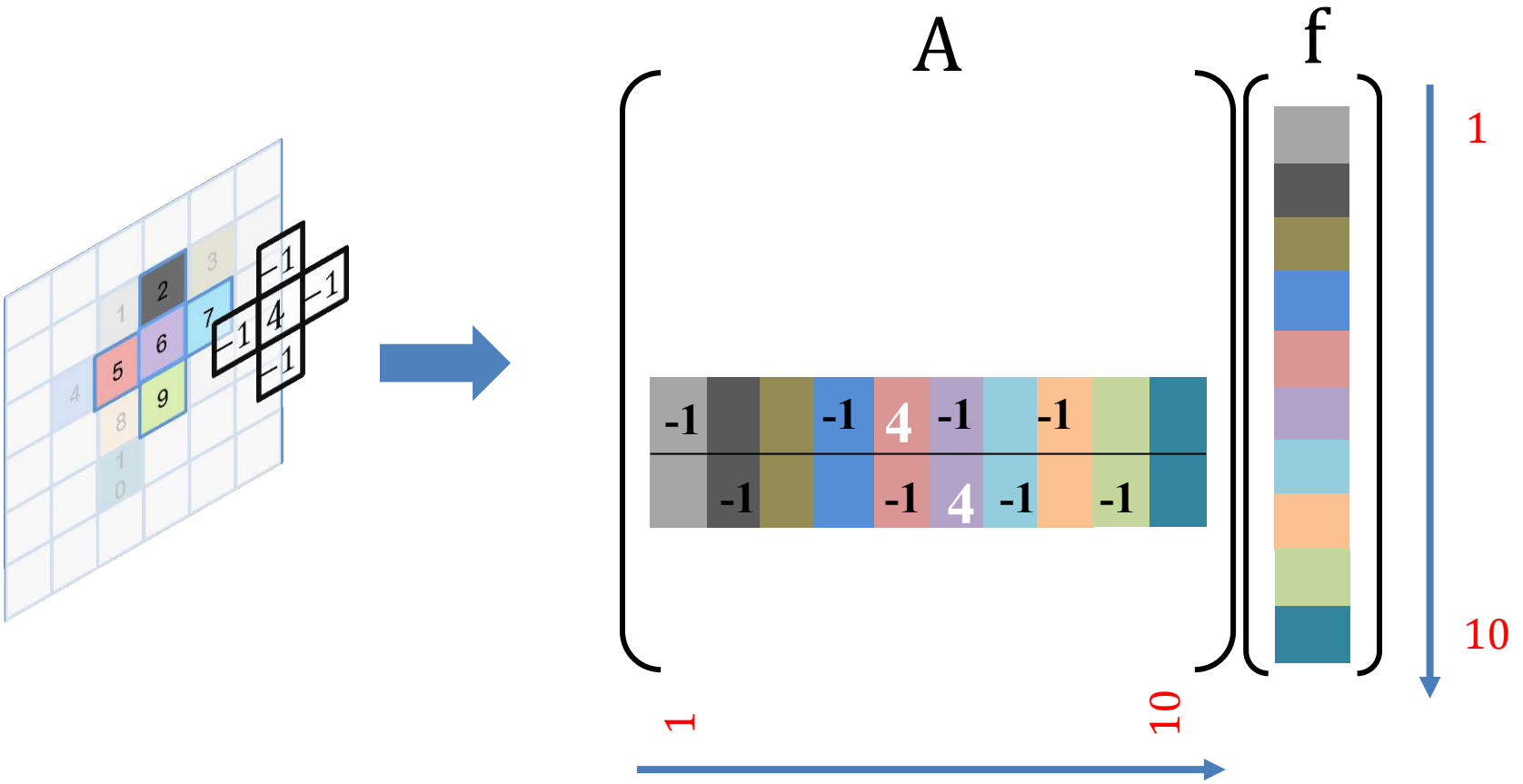


Matrix **A** encoding the Laplacian Convolution

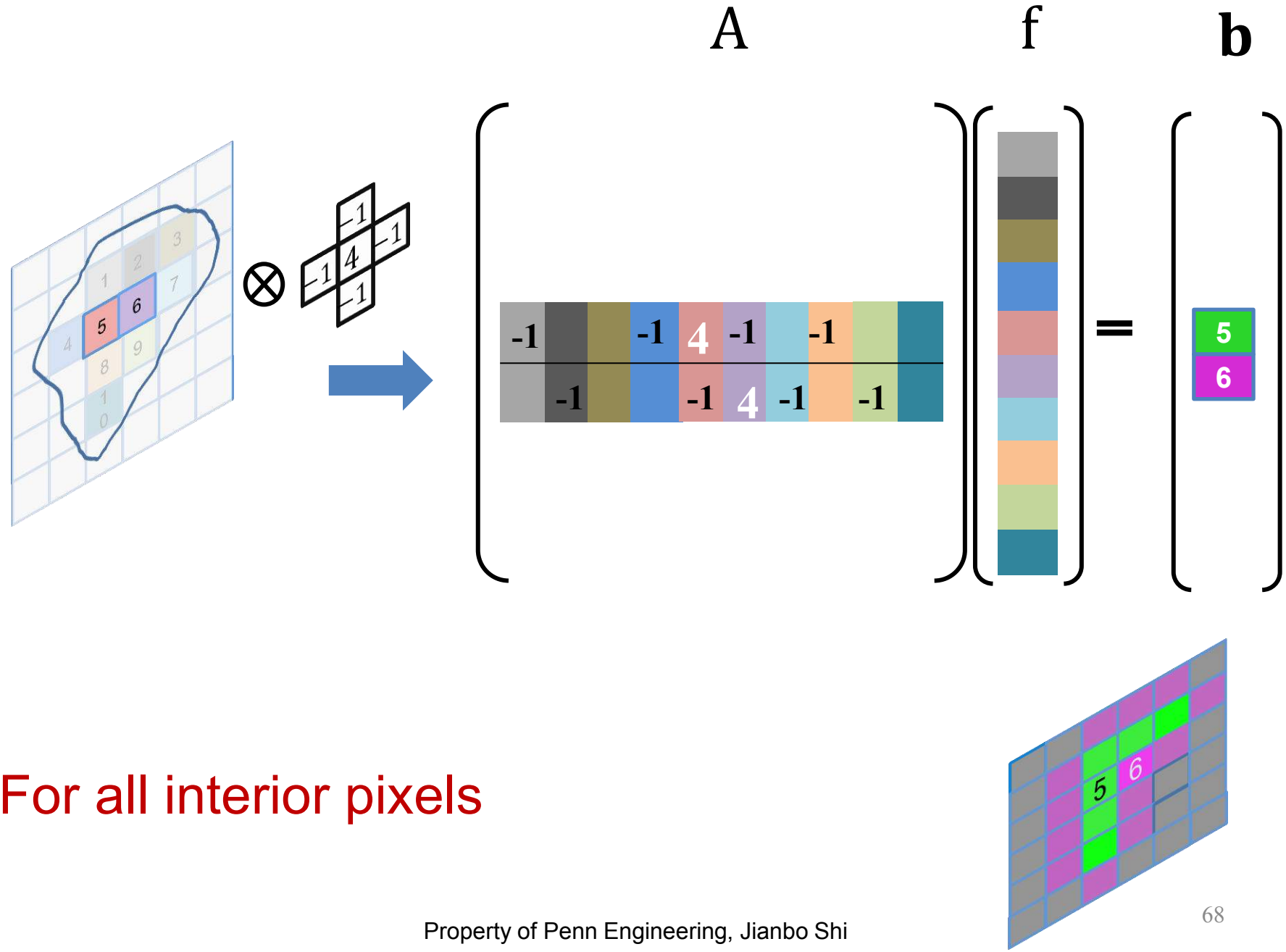


For interior pixels

Matrix **A** encoding the Laplacian Convolution

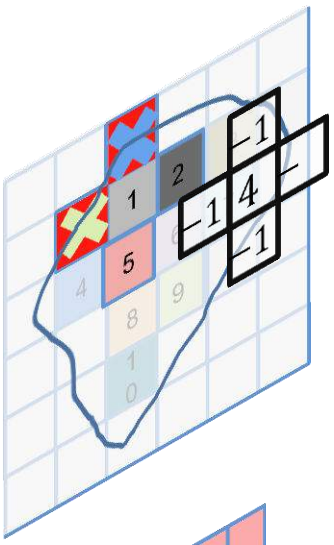


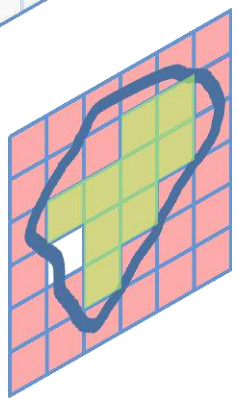
For interior pixels



For all interior pixels

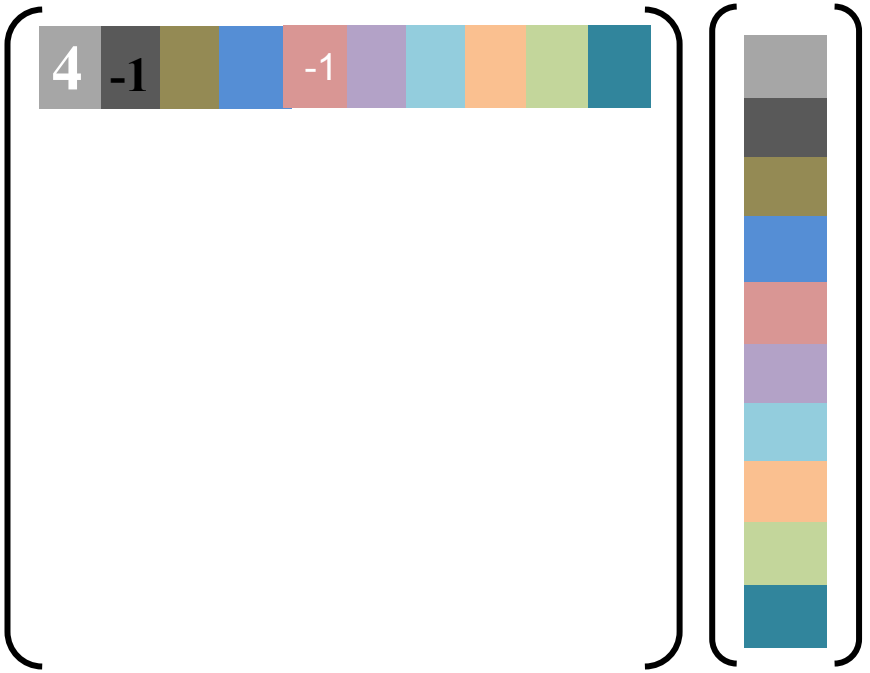
How to deal with the knowns Boundary Values?





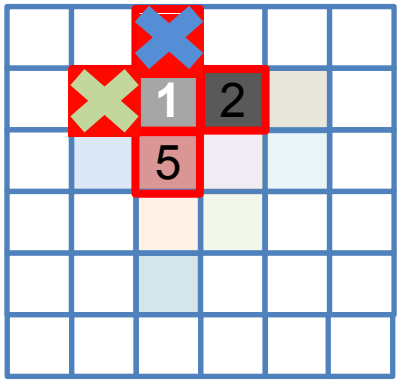
A

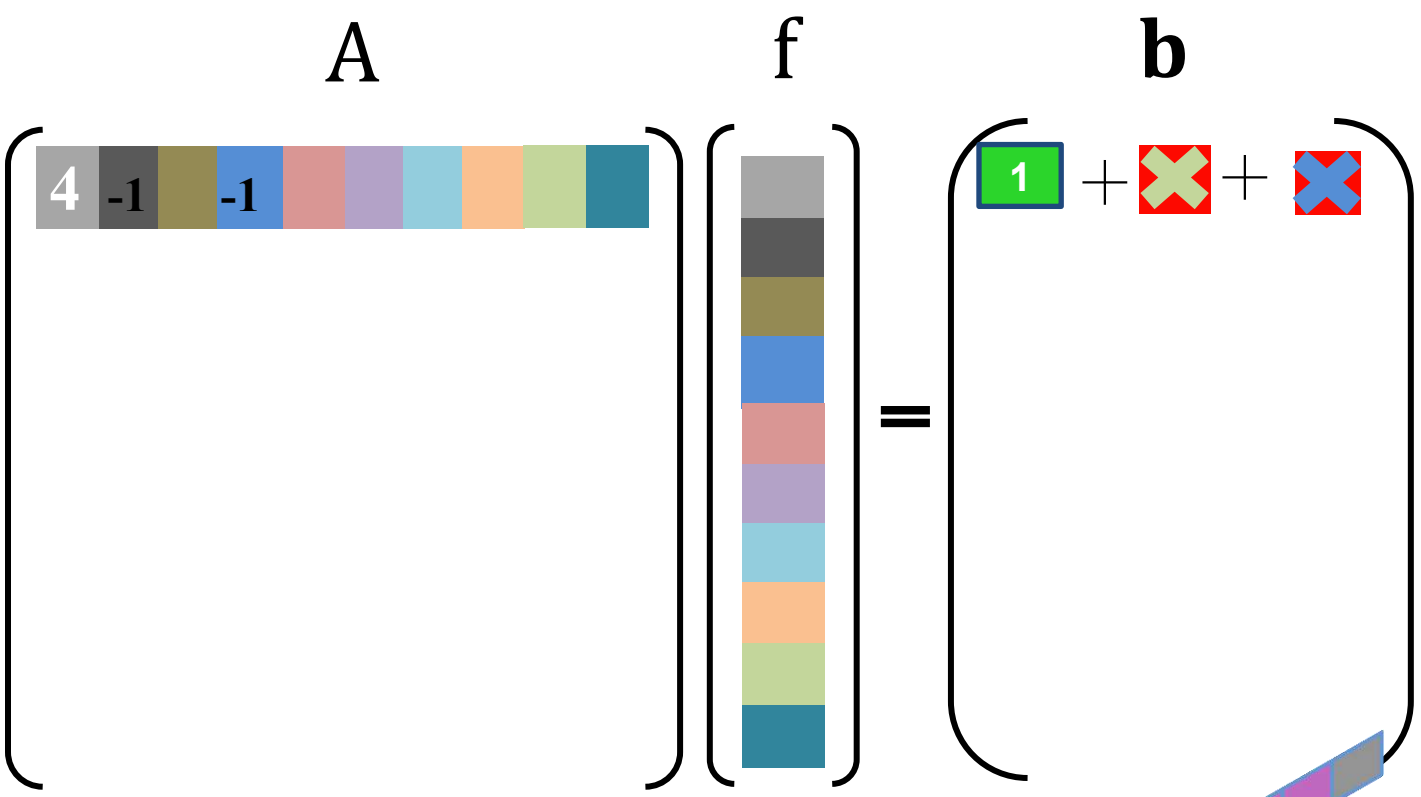
f



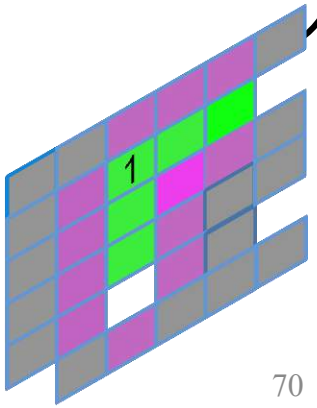
For pixels just inside of the boundary

Move the boundary value of knowns to **b** side!

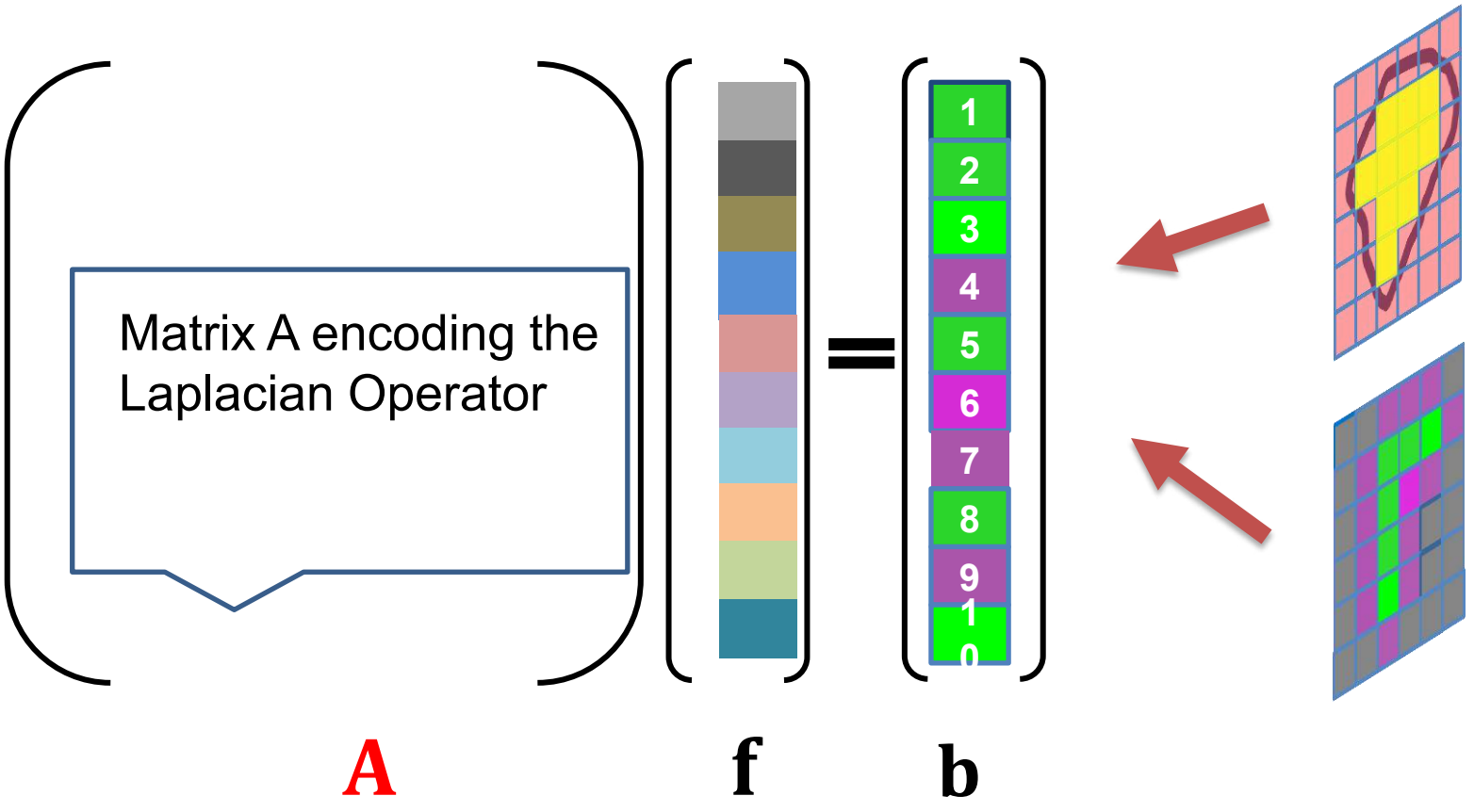




$$f(1) \times 4 - f(2) - f(5) - \boxtimes - \boxtimes = b(1)$$



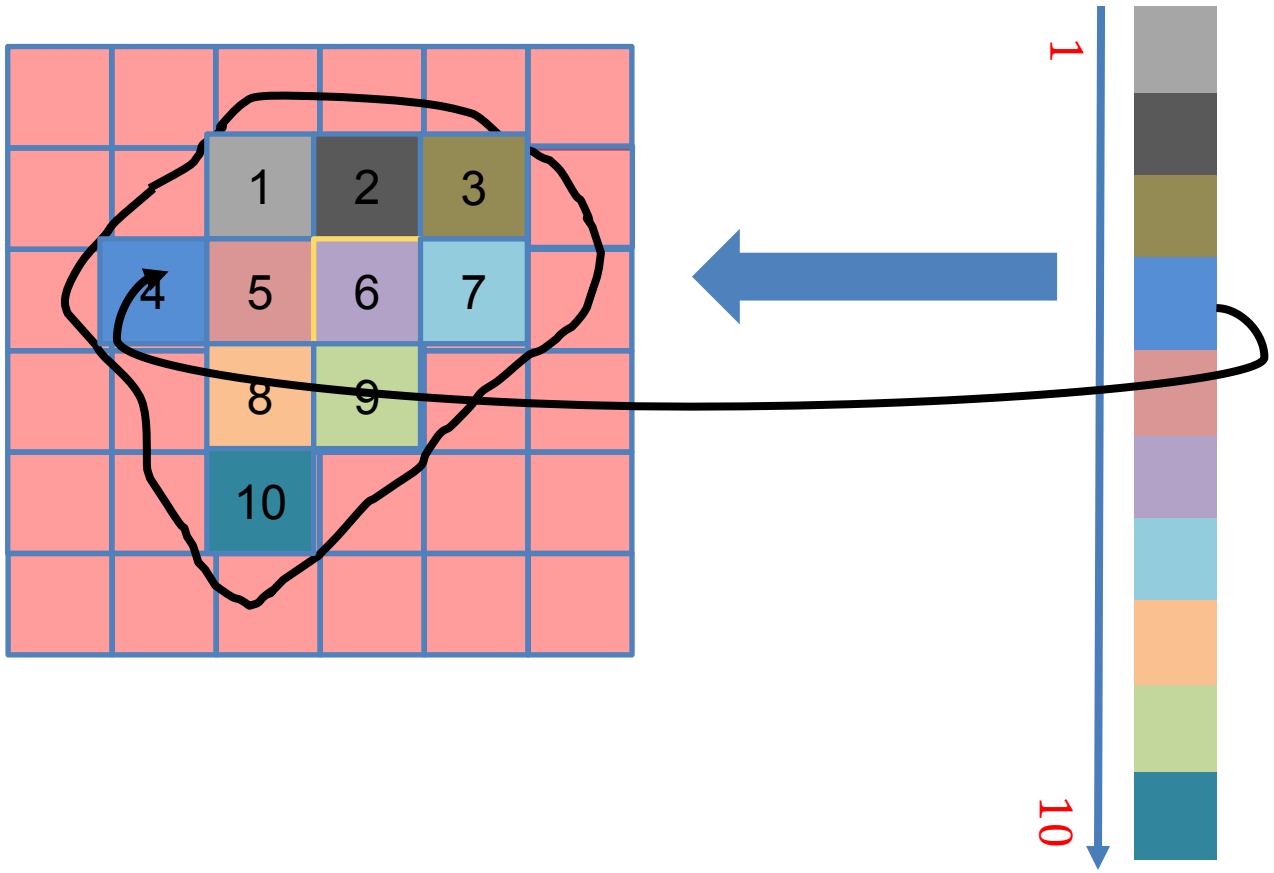
$\mathbf{Af = b}$

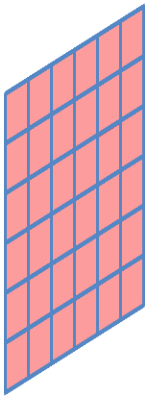


$\mathbf{A}$ $\mathbf{f}$ $\mathbf{b}$

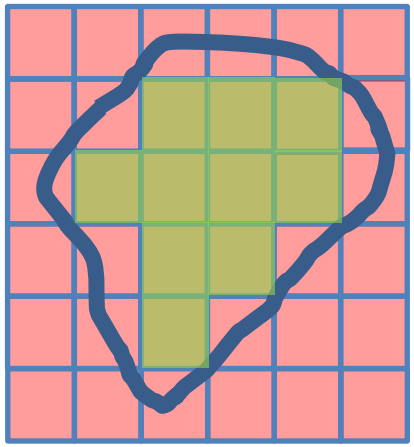
For each pixel in the mask,
put f back to the blended image

$$\mathbf{f} = \mathbf{A} \backslash \mathbf{b}$$

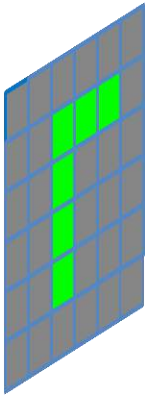




Target $\mathbf{f}$



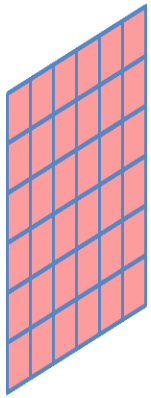
Mask



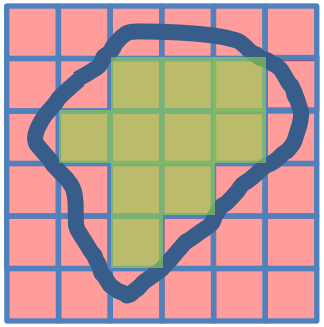
Source $\mathbf{g}$

Guess what's our blended image?

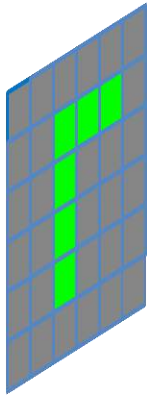
Gradient Blending



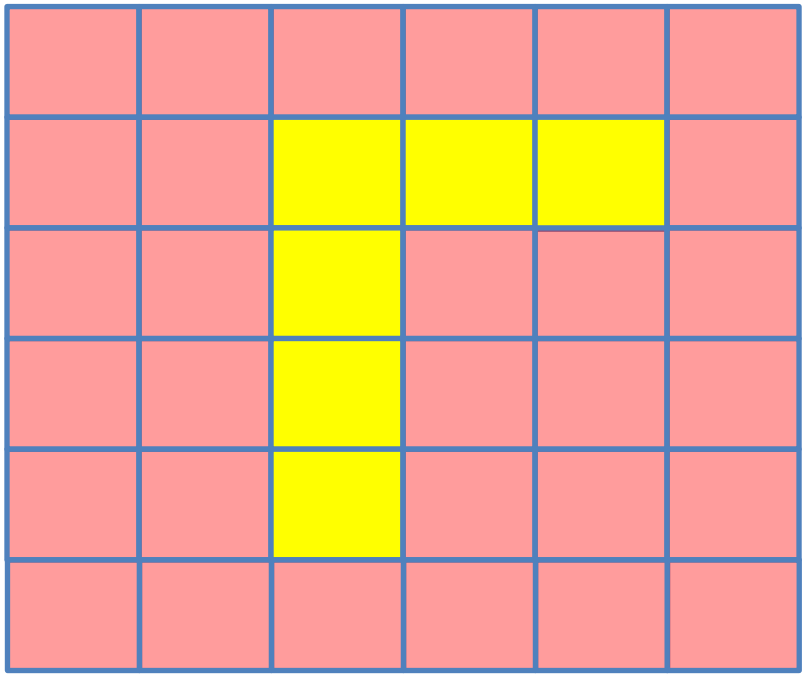
Target $\mathbf{f}$

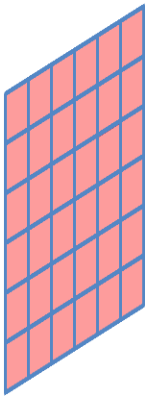


Mask

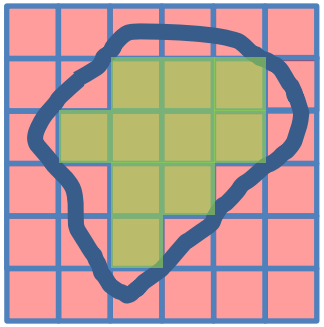


Source $\mathbf{g}$



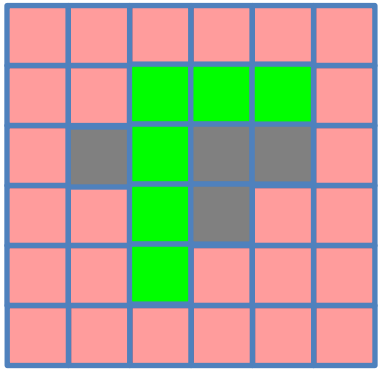


Target $\mathbf{f}$

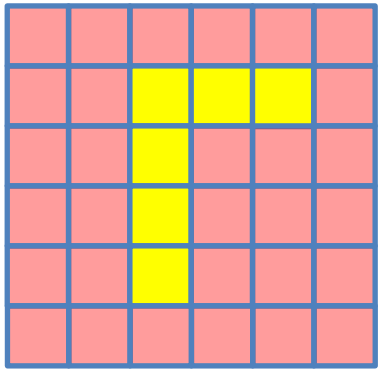


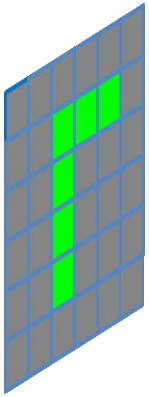
Mask

Direct Copy and Paste

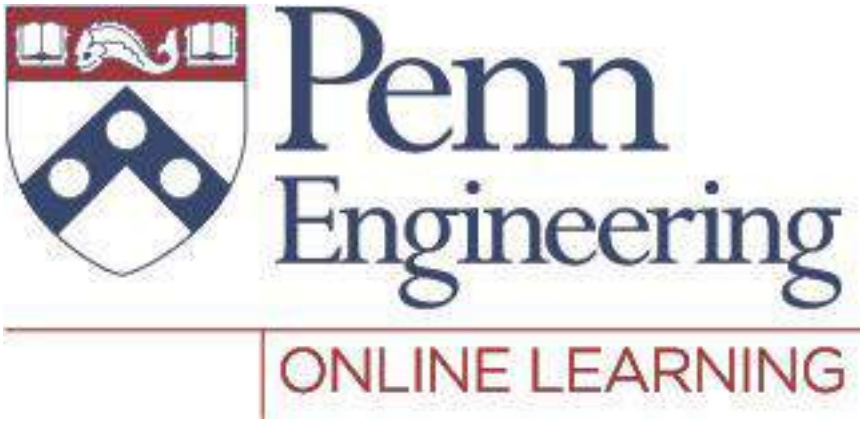


Gradient Blending



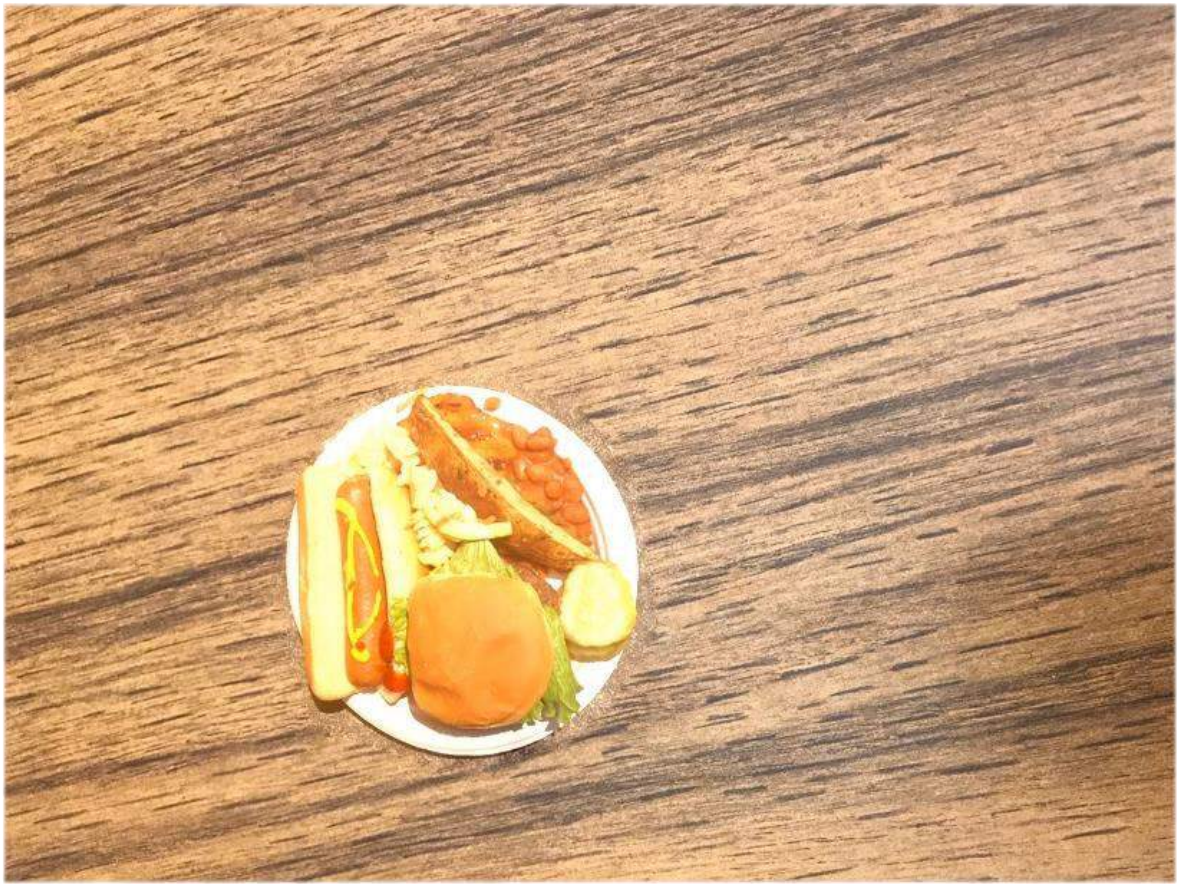


Source $\mathbf{g}$

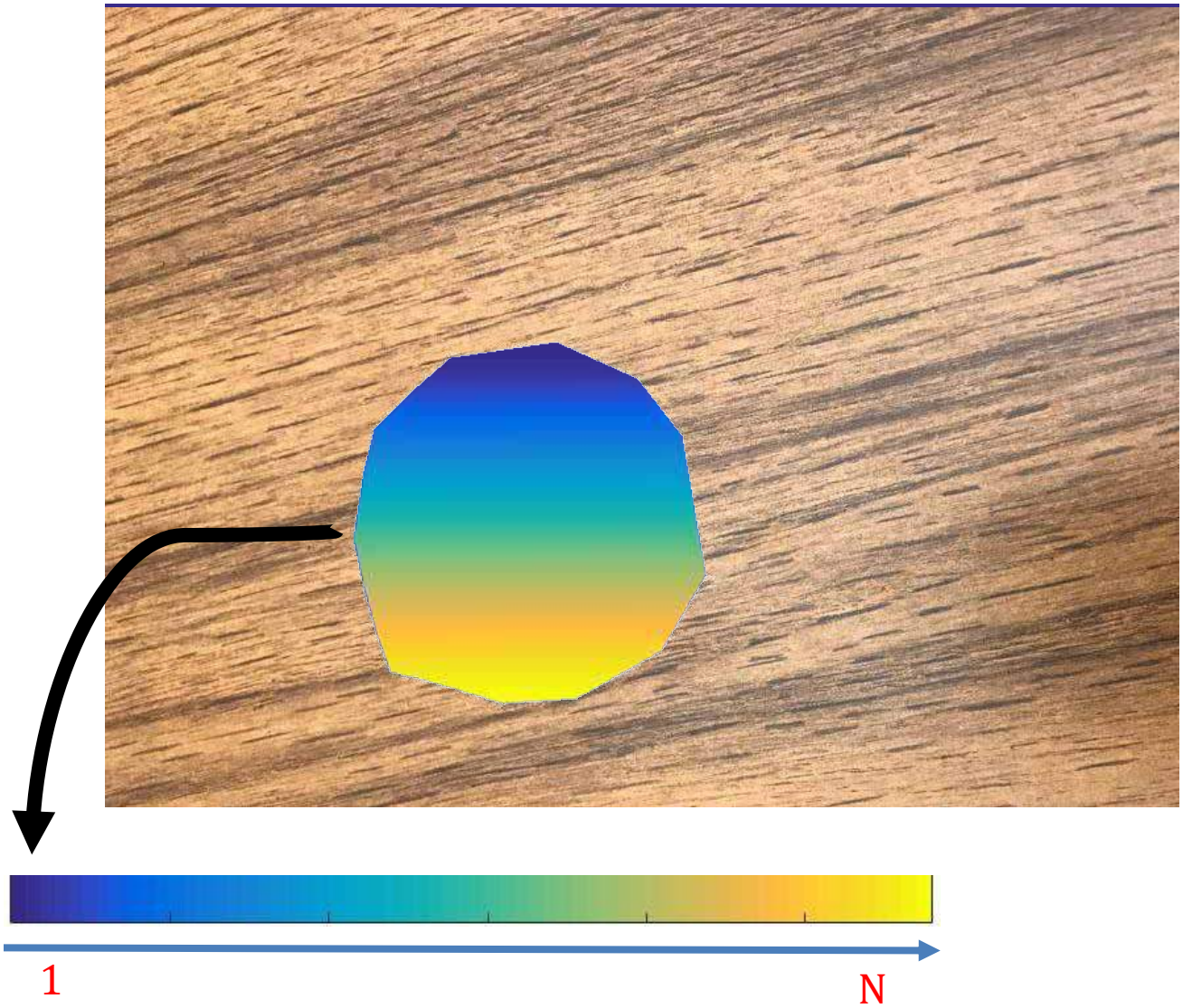


Video 8.5
Jianbo Shi

Let’s summarize with a real image

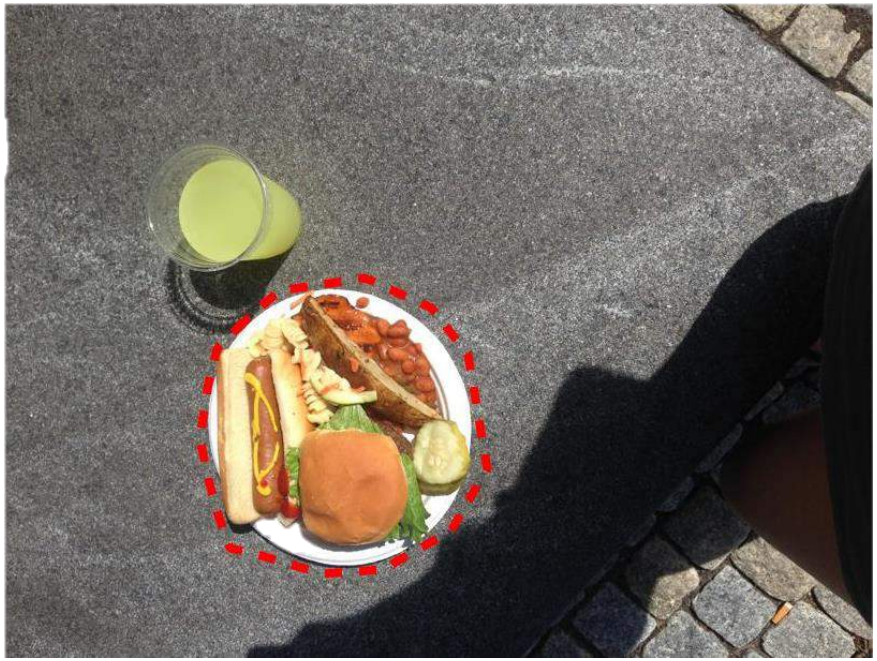


Step1: Indexing the unknowns in masked pixels

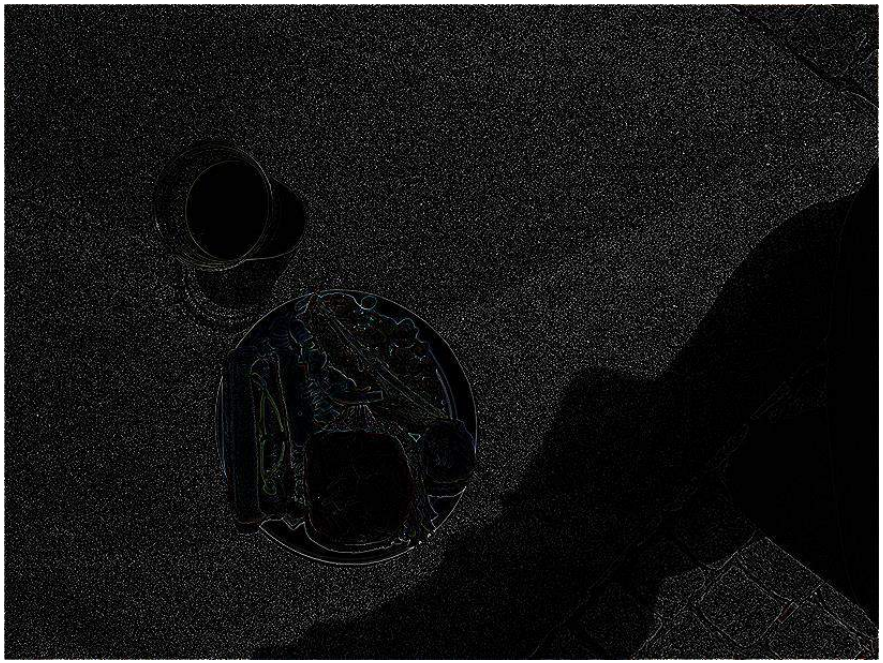


N is the number of unknowns

Step2: Compute the Laplacian from source

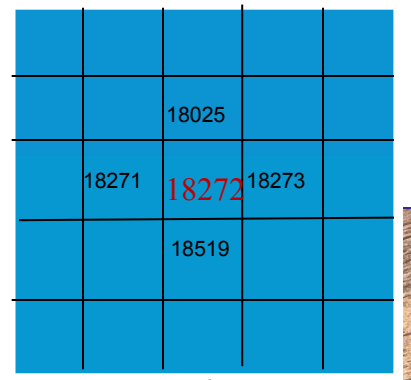


Source image



Laplacian

Step3: Constructing matrix A for the Laplacian Operator



For pixel in the region

$A(18272,18272 = 4)$

$A(18272,18025 = -1)$

$A(18272,18519 = -1)$

$A(18272,18271 = -1)$

$A(18272,18273 = -1)$

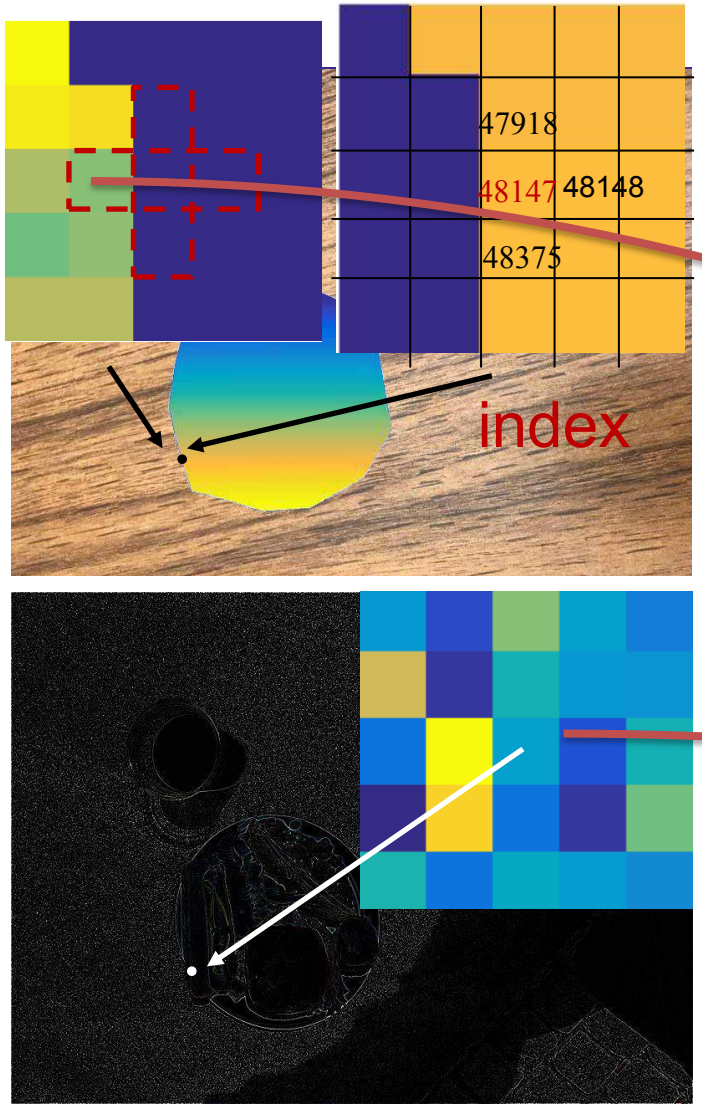
$?$ $=$

$\mathbf{A}$ $\mathbf{f}$ $\mathbf{b}$

$\mathbf{b}(18272) =$

Step3. Constructing matrix A for the Laplacian Operator

boundary value



For pixel on the boundary

$$A(48147,48147) = 4$$

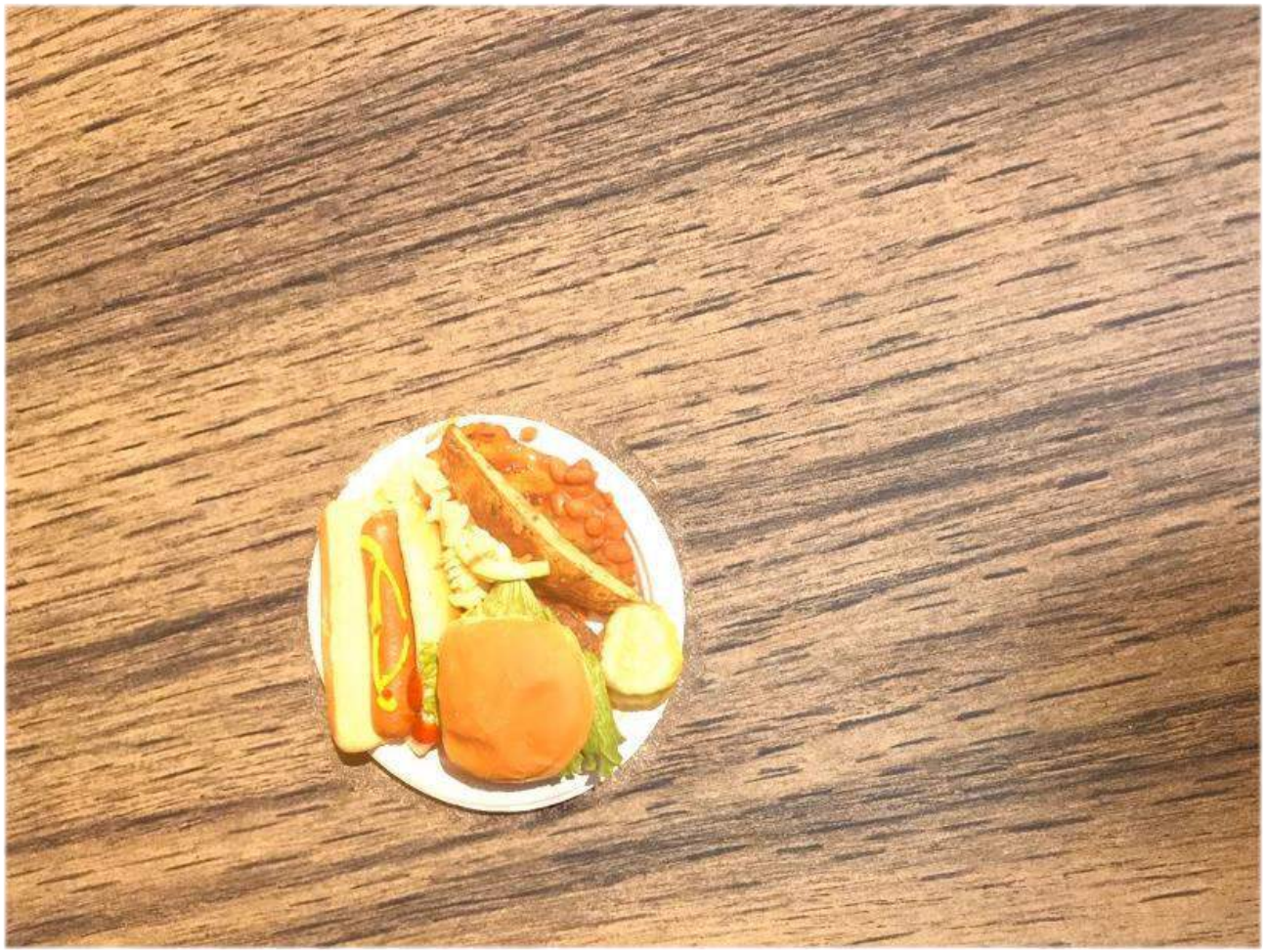
$$A(48147,47918) = -1 \quad A(48147,48375) = -1$$

$$A(48147,48148 = -1)$$

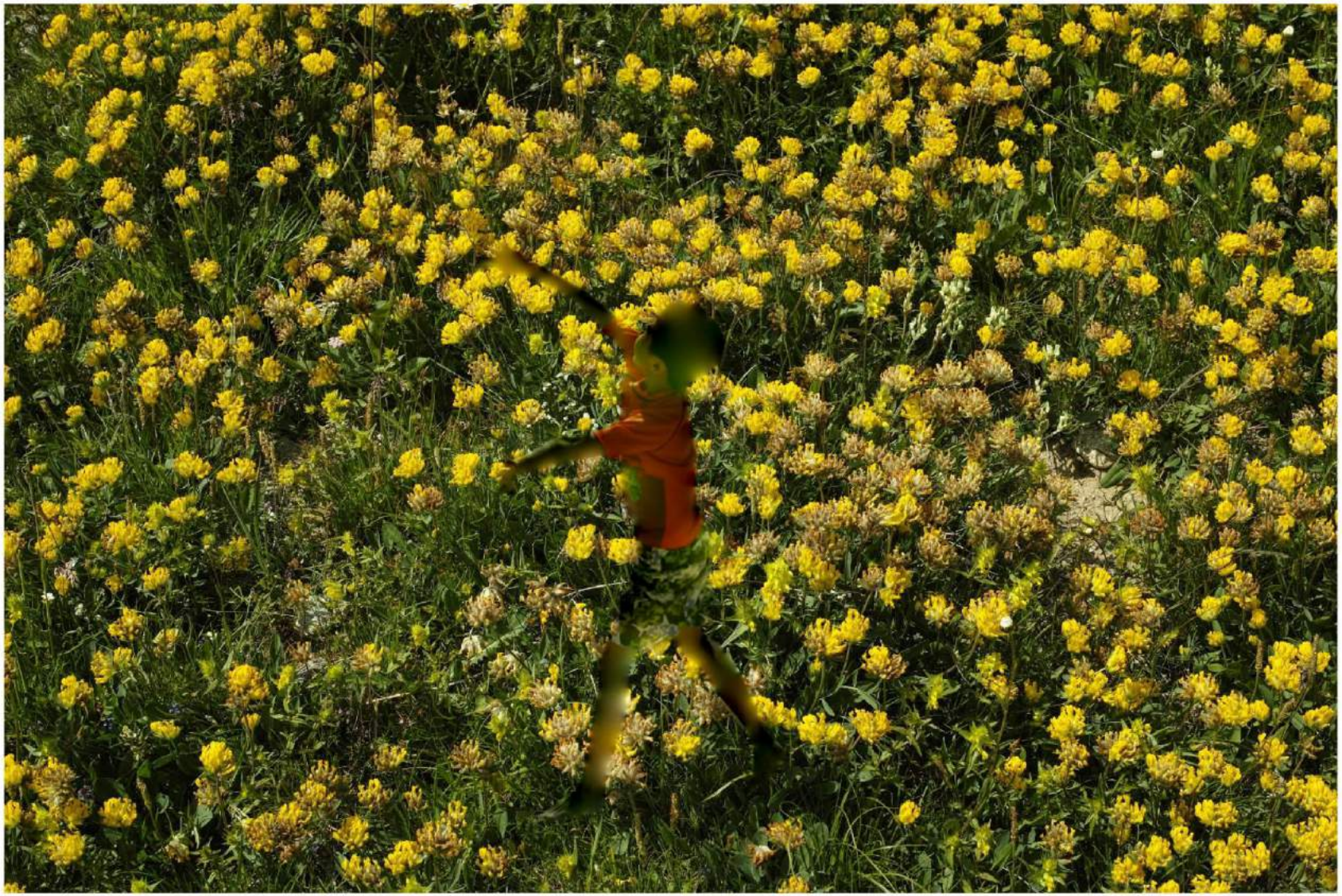
$$A \quad f \quad = \quad b$$

$$\mathbf{b}(48147) =$$

Solving the linear system and copy the values back to blended image



Where is waldo



Where is waldo

Where is waldo

Where is waldo