Dartmouth

Gradient Domain Image Processing

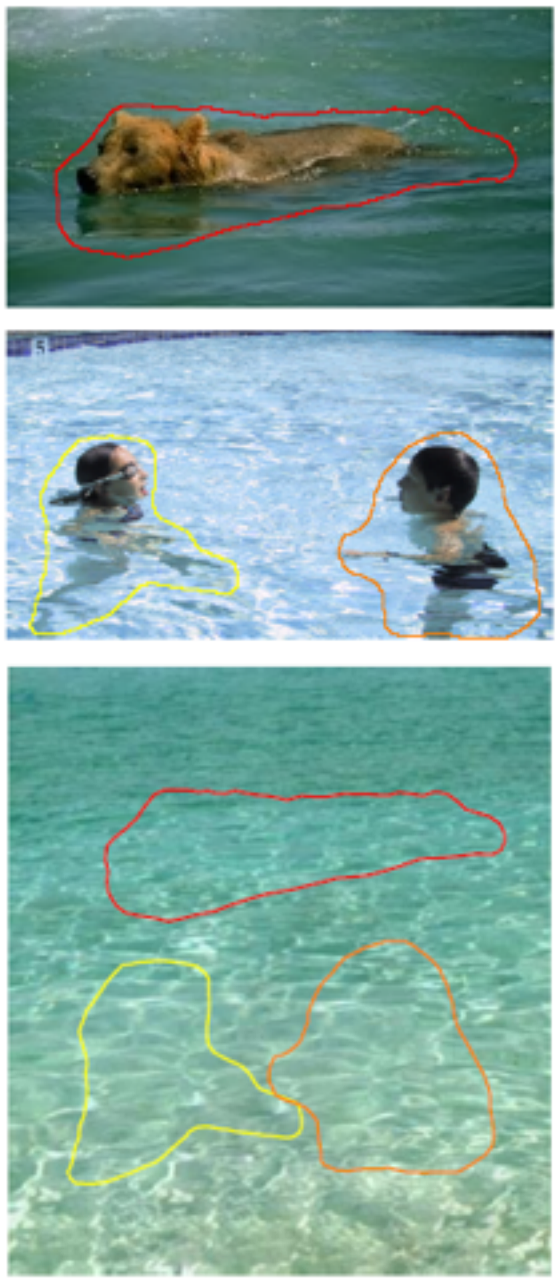
CS 89.15/189.5, Fall 2015

Wojciech Jarosz

wojciech.k.jarosz@dartmouth.edu

Problems with direct copy/paste

http://www.irisa.fr/vista/Papers/2003_siggraph_perez.pdf

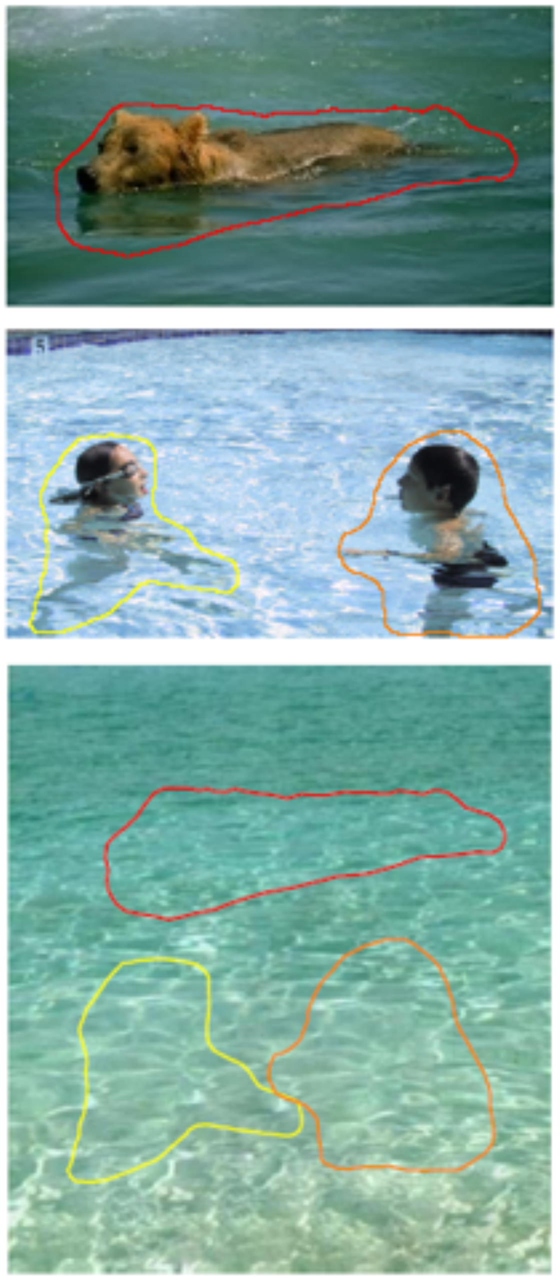


sources/destinations

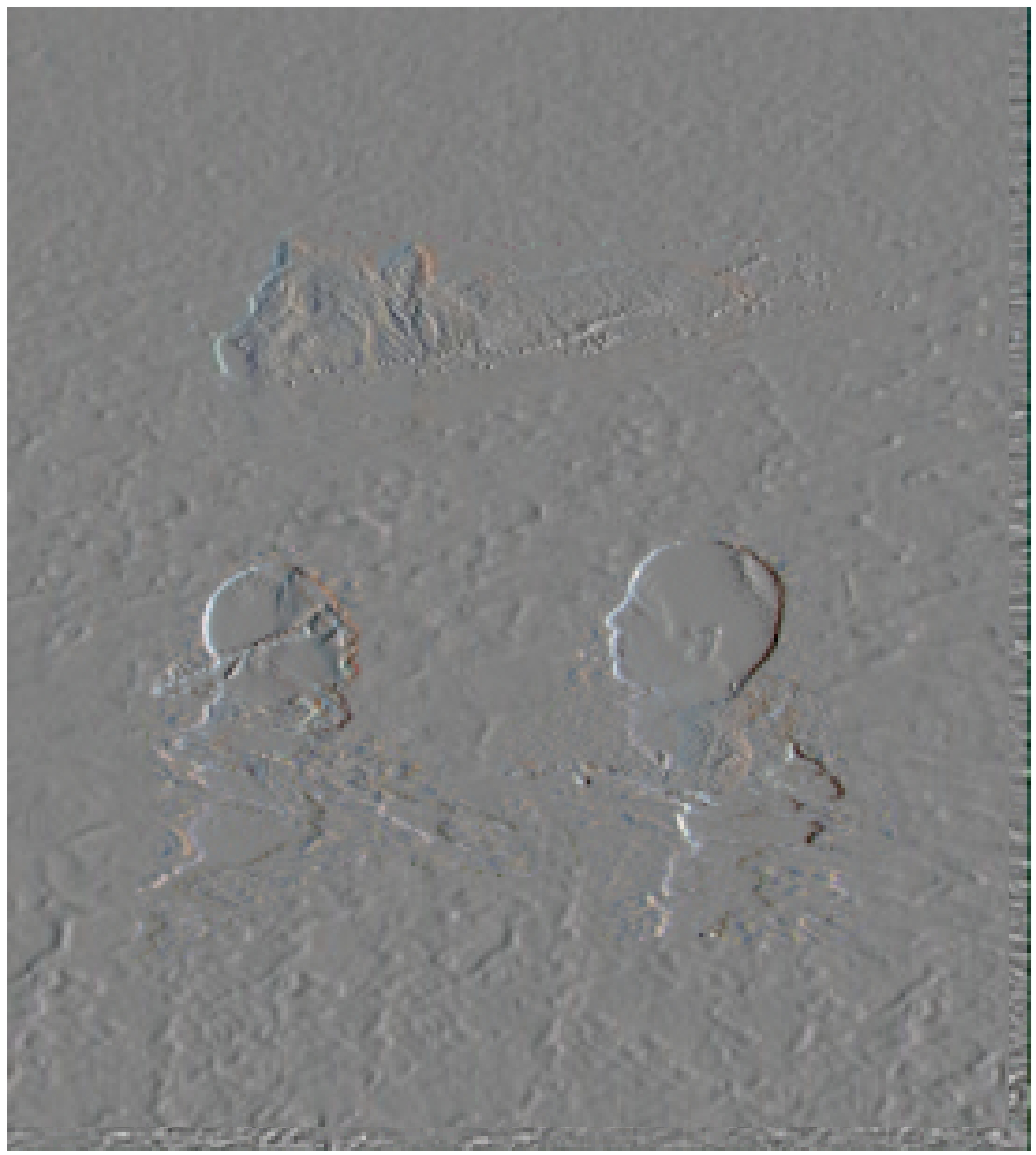
cloning

Solution: paste gradient

http://www.irisa.fr/vista/Papers/2003_siggraph_perez.pdf



sources/destinations



hacky visualization of gradient

seamless cloning

http://fstoppers.com/proof-viral-hurricane-shark-photo-in-street-is-fake

Photoshop healing brush

Slightly smarter version of what we learn today

- higher-order derivative in particular
- See also http://www.petapixel.com/2011/03/02/how-to-use-the-healing-brush-and-patch-tool-in-photoshop/

What is a gradient?

Derivative of a multivariate function; for example, for *f(x,y)*

$$\nabla f = \left(\frac{df}{dx}, \frac{df}{dy} \right)$$

For a discrete image, can be approximated with finite differences

$$\frac{df}{dx} \approx f(x+1, y) - f(x, y)$$
$$\frac{df}{dy} \approx f(x, y+1) - f(x, y)$$

Gradient: intuition

Gradients and grayscale images

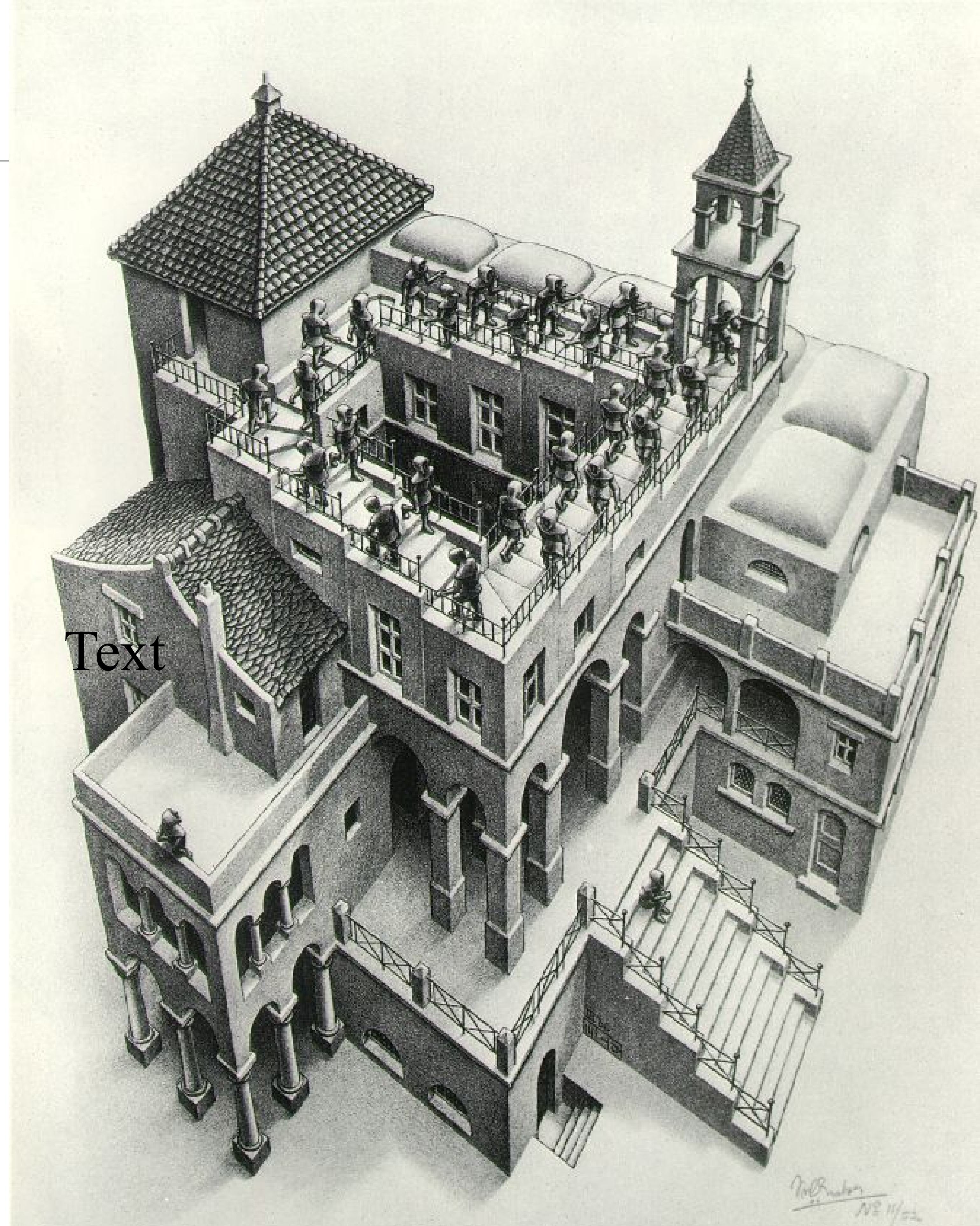
Grayscale image: n×n scalars

Gradient: n×n *2D vectors*

Two many numbers!

What's up with this?

- Not all vector fields are the gradient of an image!
- Only if they are curl-free (a.k.a. conservative)
- But we'll see it does not matter for us



Escher, Maurits Cornelis
Ascending and Descending
1960
Lithograph
35.5 x 28.5 cm (14 x 11 1/4 in.)

Color images

3 gradients, one for each channel

We'll sweep this under the rug for this lecture

In practice, treat each channel independently

Questions?

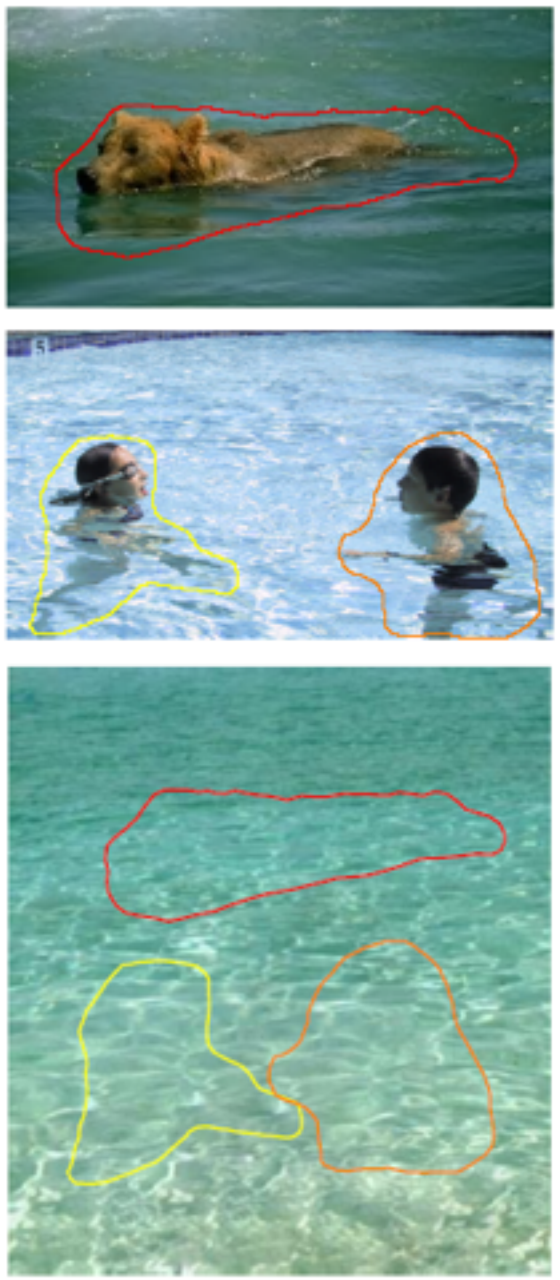
Key gradient domain idea

1. Construct a vector field that we wish was the gradient of our output image
2. Look for an image that has that gradient
3. That won’t work, so look for an image that has *approximately* the desired gradient

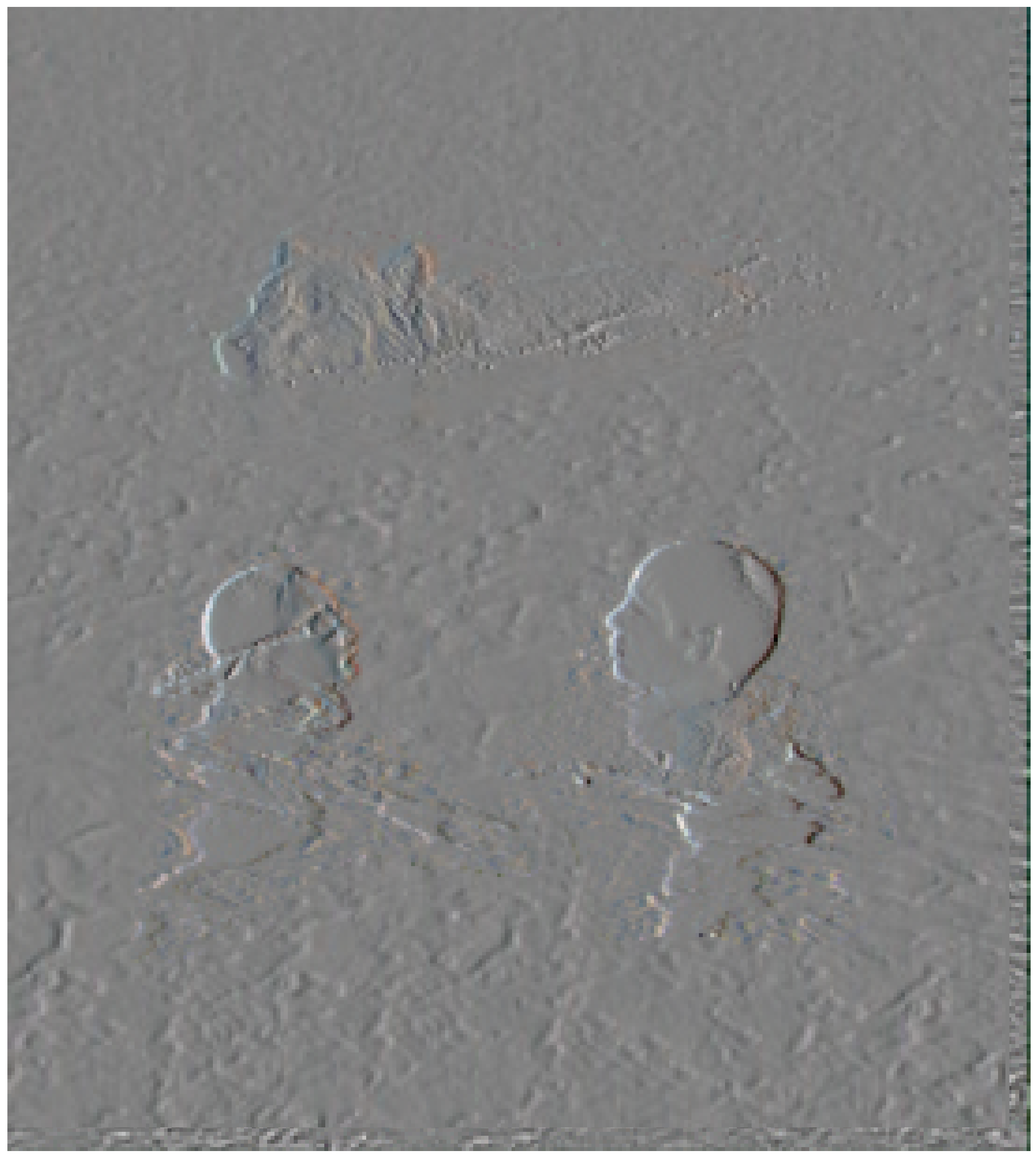
Gradient domain image processing is all about clever choices for (1) and efficient algorithms for (3)

Solution: paste gradient

http://www.irisa.fr/vista/Papers/2003_siggraph_perez.pdf



sources/destinations



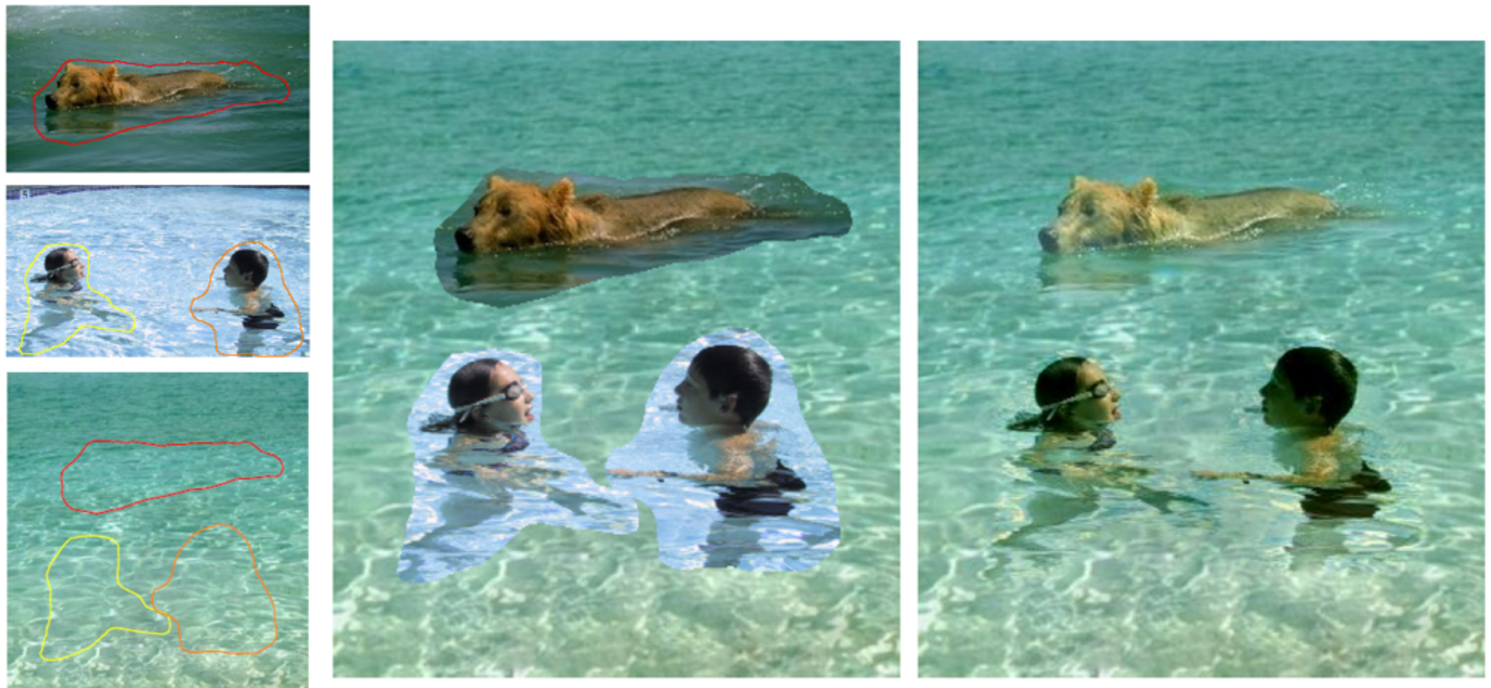
hacky visualization of gradient

seamless cloning

Seamless Poisson cloning

Paste source gradient into target image inside a selected region

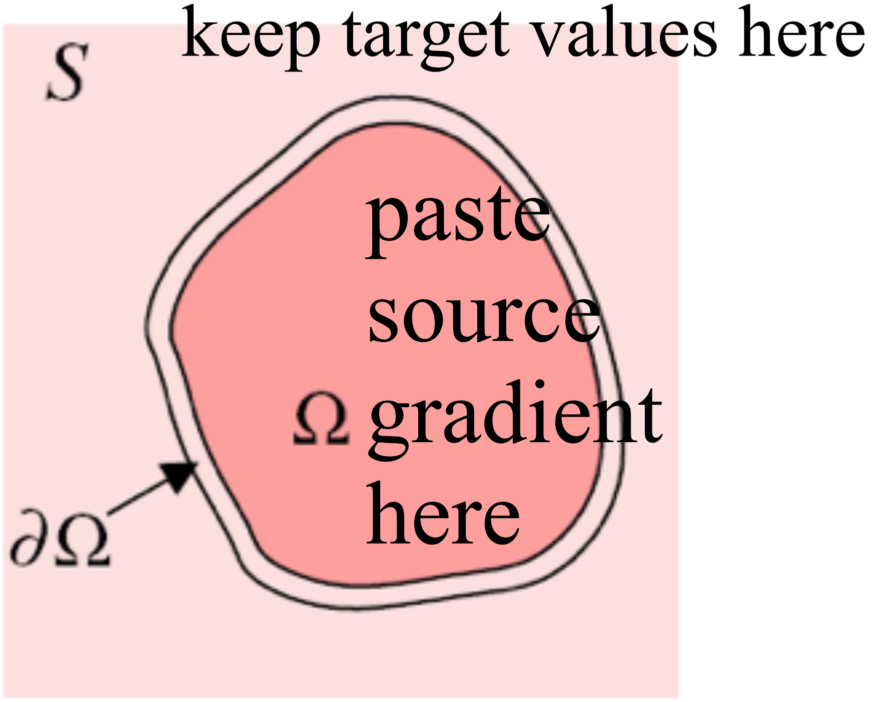
Make the new gradient as close as possible to the source gradient while respecting pixel values at the boundary



sources/destinations

cloning

seamless cloning



Seamless Poisson cloning

Given vector field *v* (pasted gradient), find the value of *f* in unknown region that optimizes:

$$\min_{f} \iint_{\Omega} |\nabla f - \mathbf{v}|^2 \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

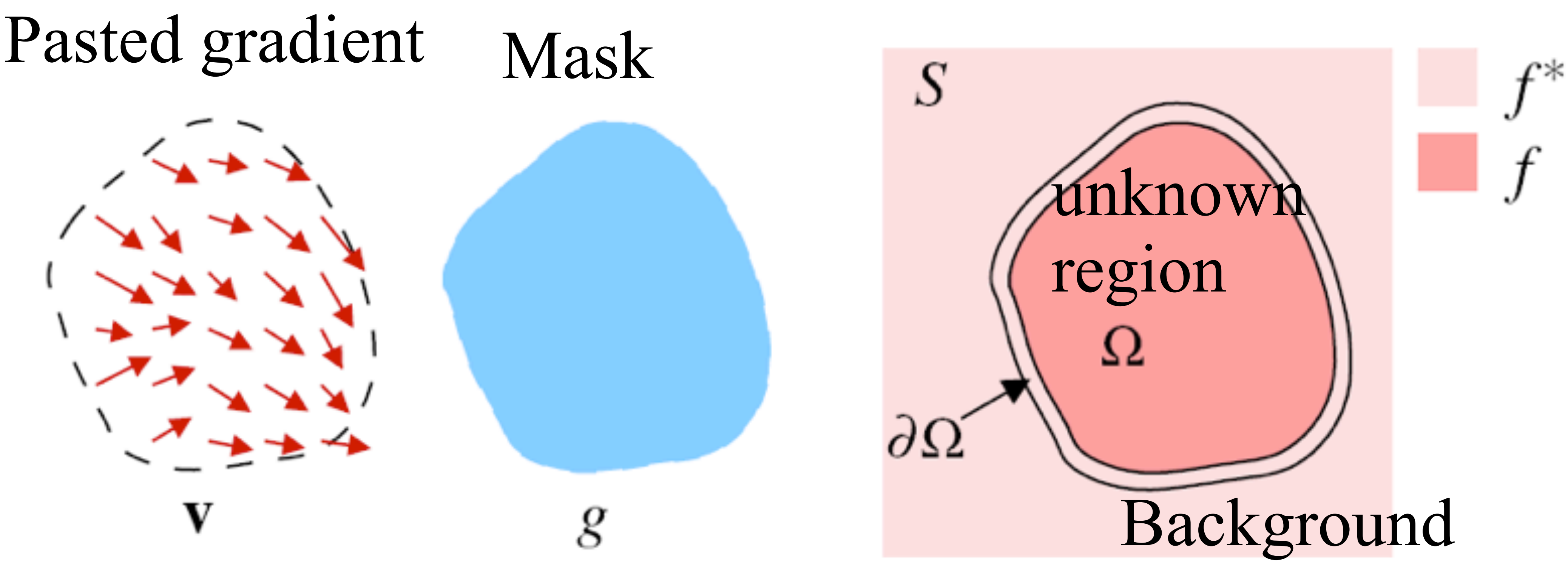
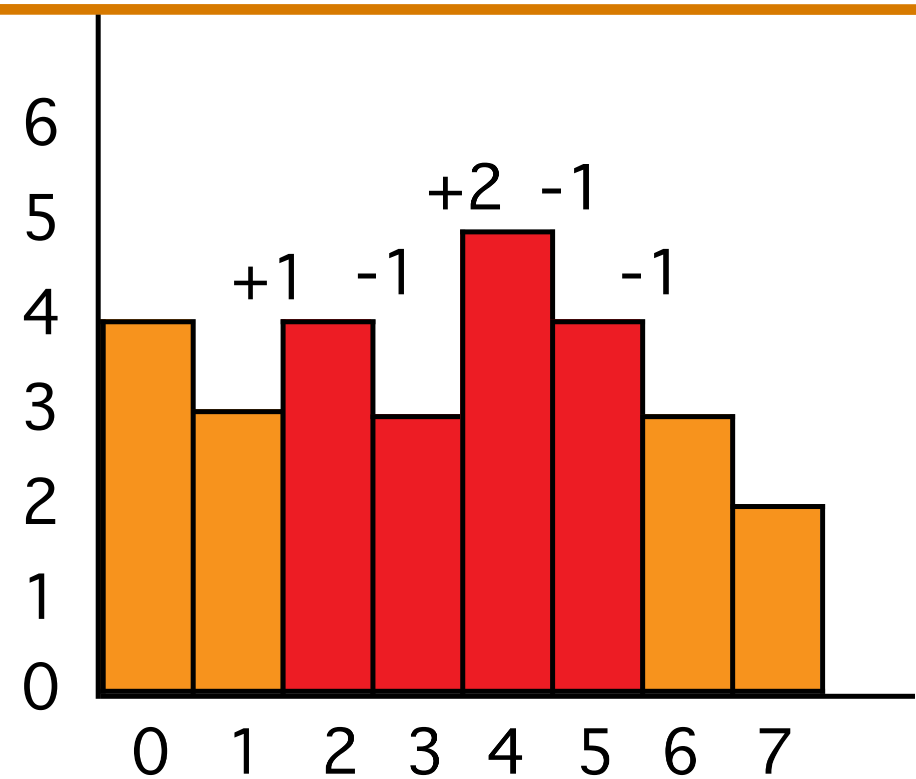
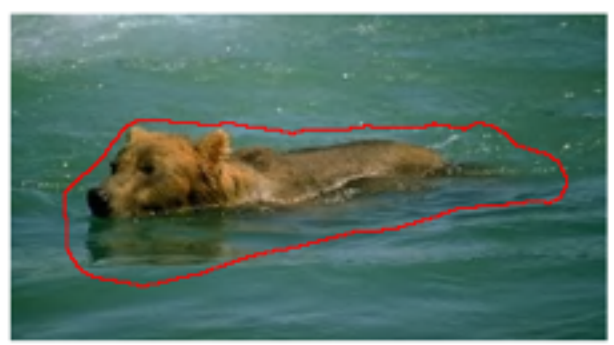


Figure 1: **Guided interpolation notations.** Unknown function f interpolates in domain Ω the destination function f^*, under guidance of vector field $\mathbf{v}$, which might be or not the gradient field of a source function g.

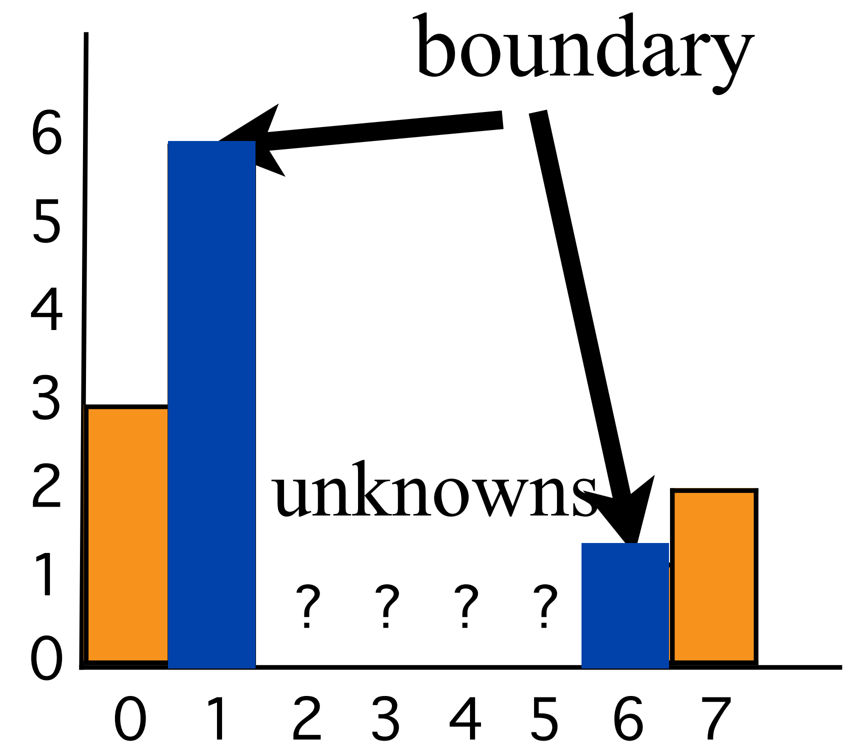
Discrete 1D example: minimization

- **Copy**





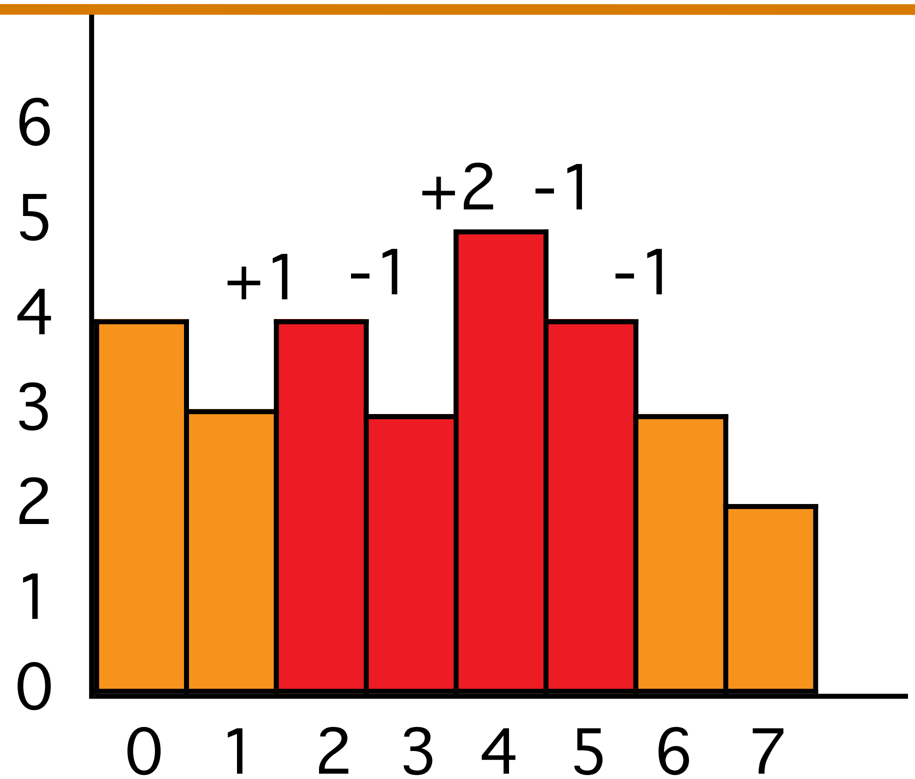
to



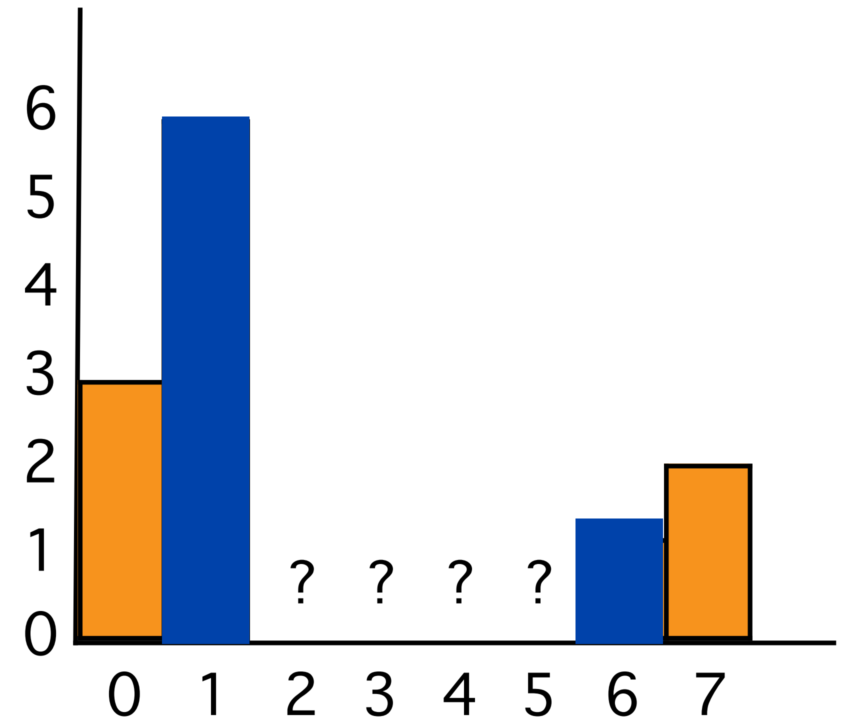
orange: pixel outside the mask
red: source pixel to be pasted
blue: boundary conditions (in background)

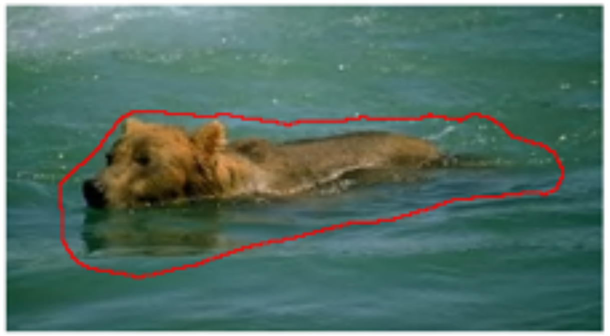
Discrete 1D example: minimization

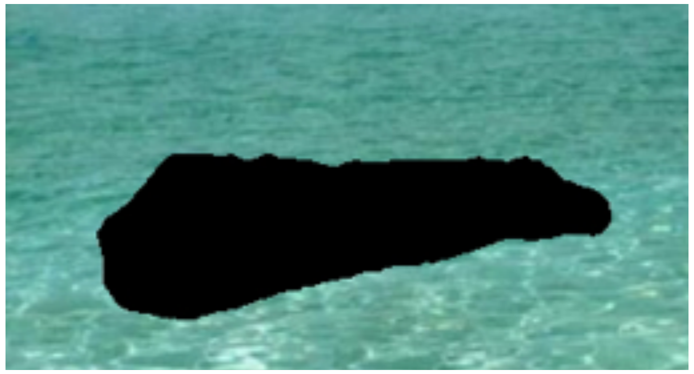
- Copy



to







Min $[(f_2-f_1)-1]^2$

$+ [(f_3-f_2)-(-1)]^2$

$+ [(f_4-f_3)-2]^2$

$+ [(f_5-f_4)-(-1)]^2$

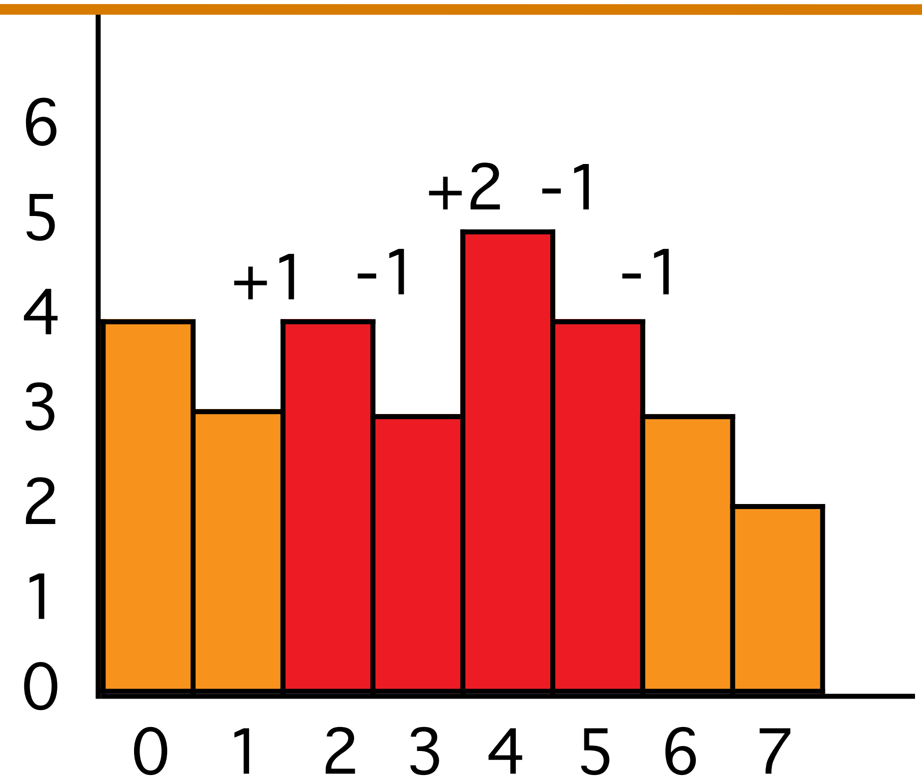
$+ [(f_6-f_5)-(-1)]^2$

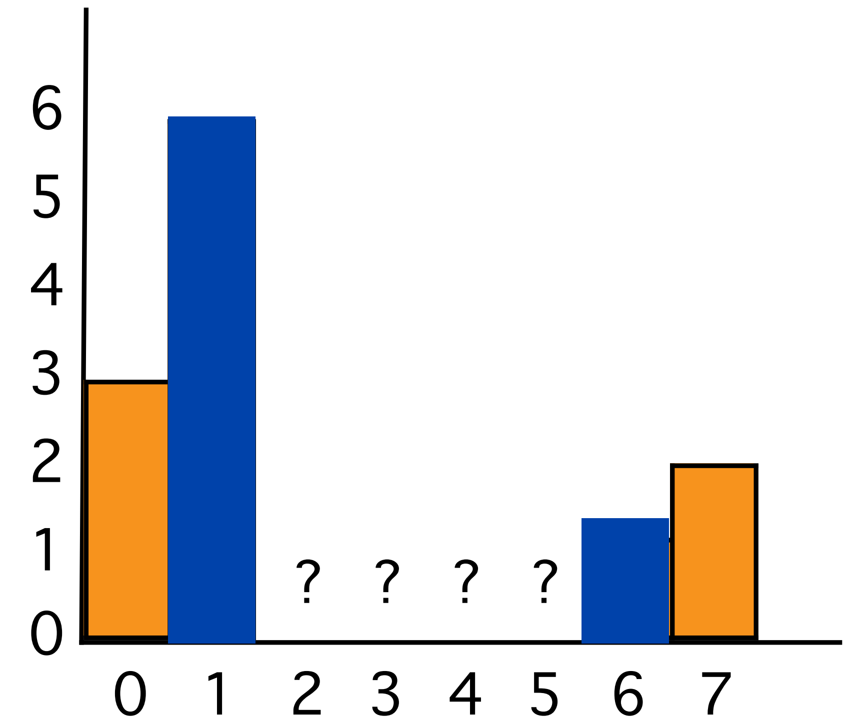
With

$f_1=6$

$f_6=1$

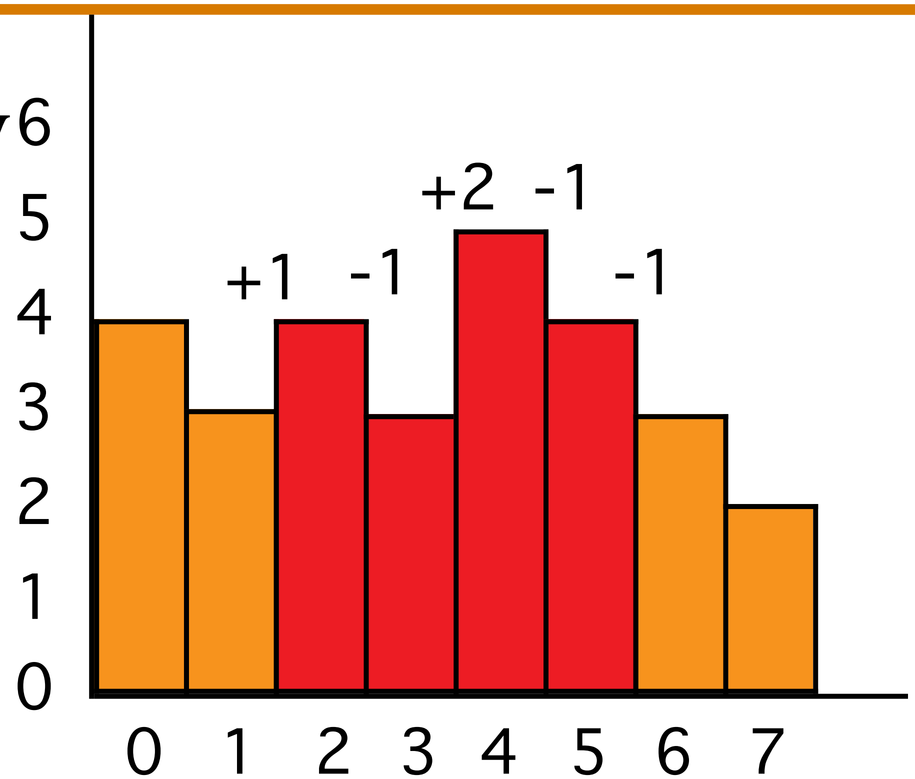
1D example: minimization

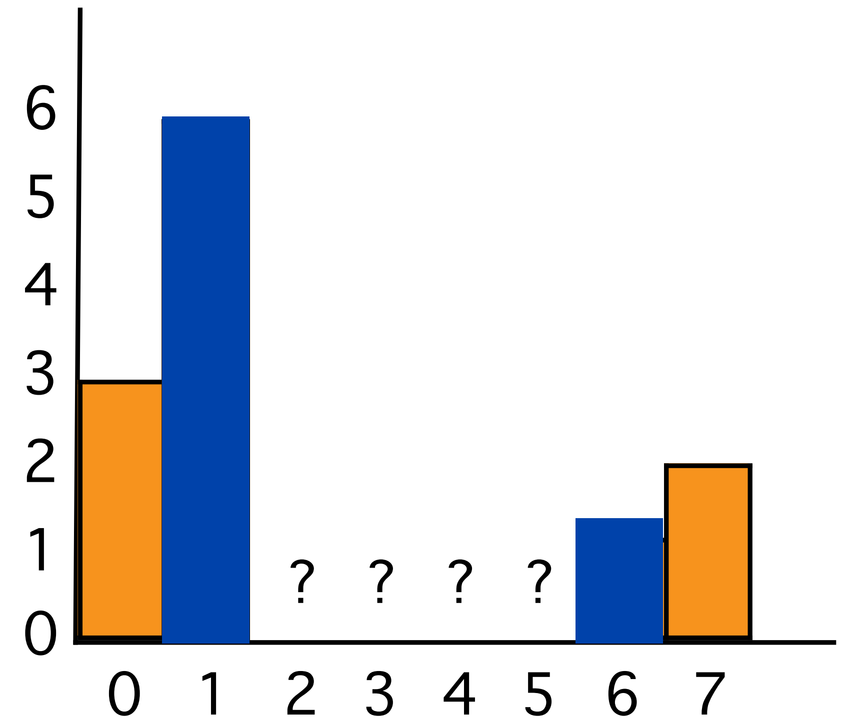
- **Copy** 

to 

Min $[(f_2-f_1)-1]^2$	$\Longrightarrow f_2^2+49-14f_2$
$+ [(f_3-f_2)-(-1)]^2$	$\Longrightarrow f_3^2+f_2^2+1-2f_3f_2 +2f_3-2f_2$
$+ [(f_4-f_3)-2]^2$	$\Longrightarrow f_4^2+f_3^2+4-2f_3f_4 -4f_4+4f_3$
$+ [(f_5-f_4)-(-1)]^2$	$\Longrightarrow f_5^2+f_4^2+1-2f_5f_4 +2f_5-2f_4$
$+ [(f_6-f_5)-(-1)]^2$	$\Longrightarrow f_5^2+4-4f_5$

1D example: big quadratic

- Copy 

to 

- Min $(f_2^2+49-14f_2$

$+ f_3^2+f_2^2+1-2f_3f_2 +2f_3-2f_2$

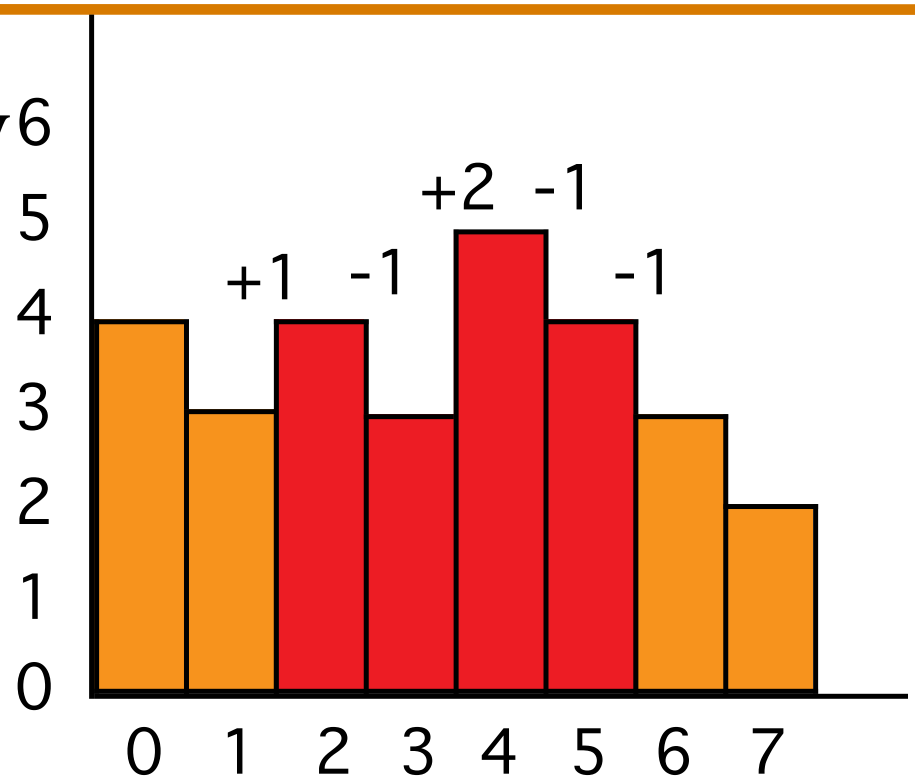
$+ f_4^2+f_3^2+4-2f_3f_4 -4f_4+4f_3$

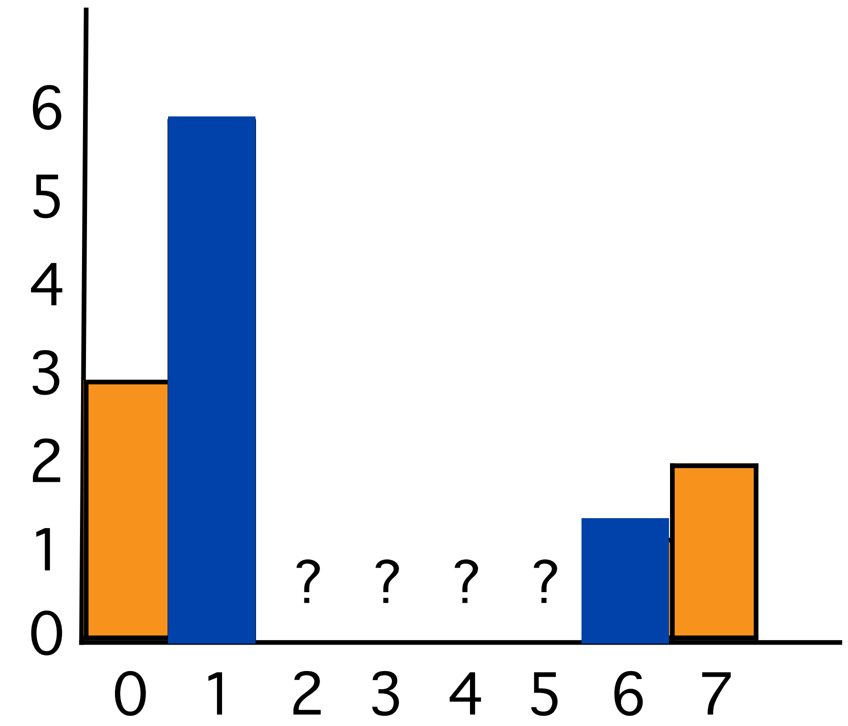
$+ f_5^2+f_4^2+1-2f_5f_4 +2f_5-2f_4$

$+ f_5^2+4-4f_5)$

Denote it Q

1D example: derivatives

- **Copy** 

to 

Min $(f_2^2+49-14f_2$

$+ f_3^2+f_2^2+1-2f_3f_2 +2f_3-2f_2$

$+ f_4^2+f_3^2+4-2f_3f_4 -4f_4+4f_3$

$+ f_5^2+f_4^2+1-2f_5f_4 +2f_5-2f_4$

$+ f_5^2+4-4f_5)$

Denote it Q

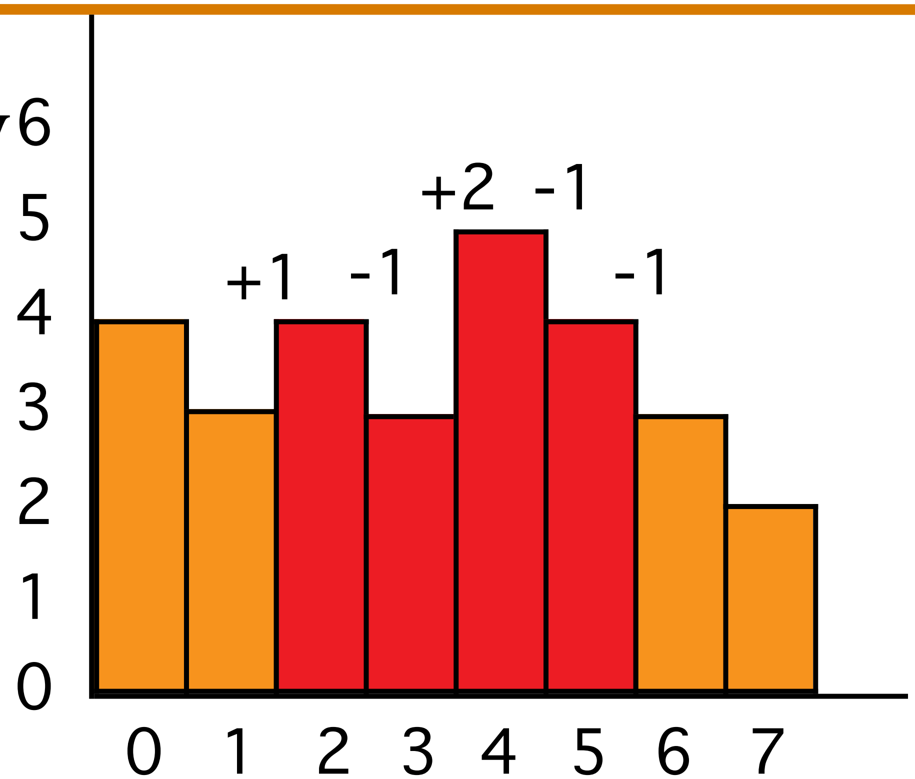
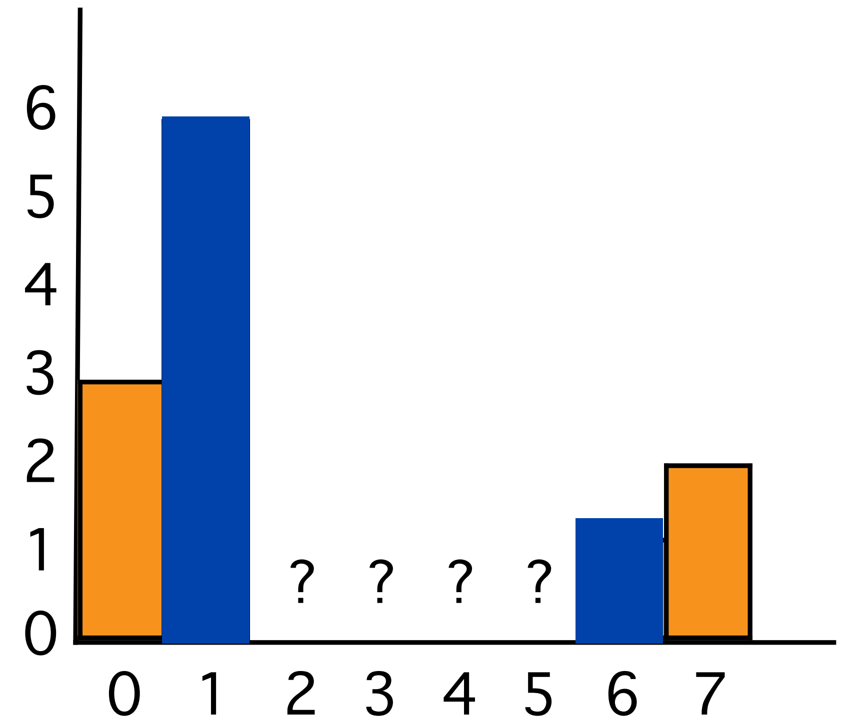
$$\frac{dQ}{df_2} = 2f_2 + 2f_2 - 2f_3 - 16$$

$$\frac{dQ}{df_3} = 2f_3 - 2f_2 + 2 + 2f_3 - 2f_4 + 4$$

$$\frac{dQ}{df_4} = 2f_4 - 2f_3 - 4 + 2f_4 - 2f_5 - 2$$

$$\frac{dQ}{df_5} = 2f_5 - 2f_4 + 2 + 2f_5 - 4$$

1D example: set derivatives to zero

- **Copy**  **to** 

$$\frac{dQ}{df_2} = 2f_2 + 2f_2 - 2f_3 - 16 = 0$$

$$\frac{dQ}{df_3} = 2f_3 - 2f_2 + 2 + 2f_3 - 2f_4 + 4 = 0$$

$$\frac{dQ}{df_4} = 2f_4 - 2f_3 - 4 + 2f_4 - 2f_5 - 2 = 0$$

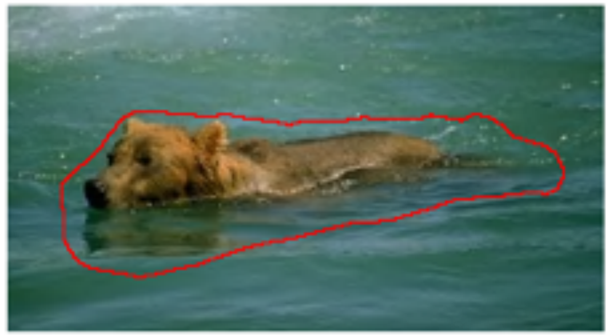
$$\frac{dQ}{df_5} = 2f_5 - 2f_4 + 2 + 2f_5 - 4 = 0$$

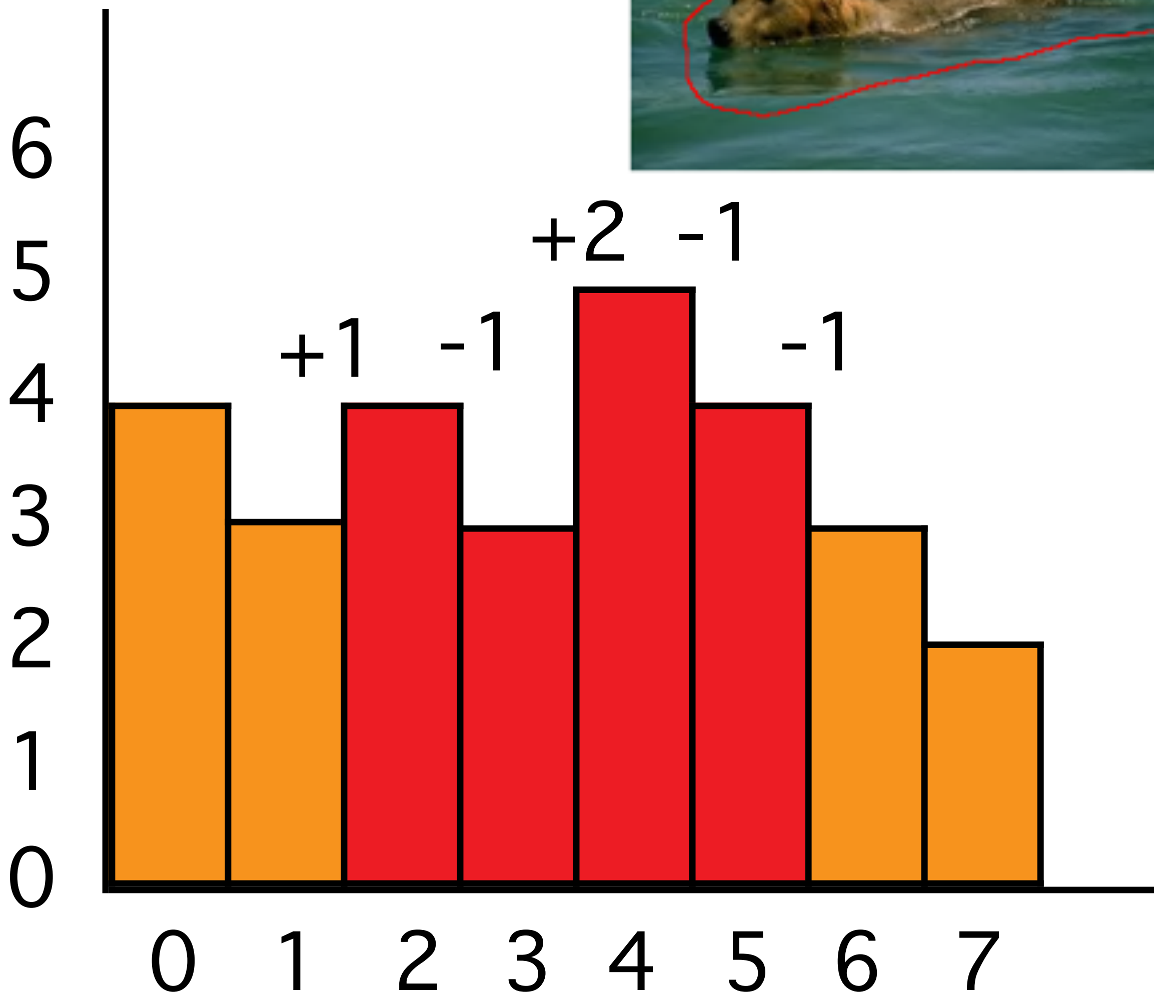
==>

$$\begin{pmatrix} 4 & -2 & 0 & 0 \\ -2 & 4 & -2 & 0 \\ 0 & -2 & 4 & -2 \\ 0 & 0 & -2 & 4 \end{pmatrix} \begin{pmatrix} f_2 \\ f_3 \\ f_4 \\ f_5 \end{pmatrix} = \begin{pmatrix} 16 \\ -6 \\ 6 \\ 2 \end{pmatrix}$$

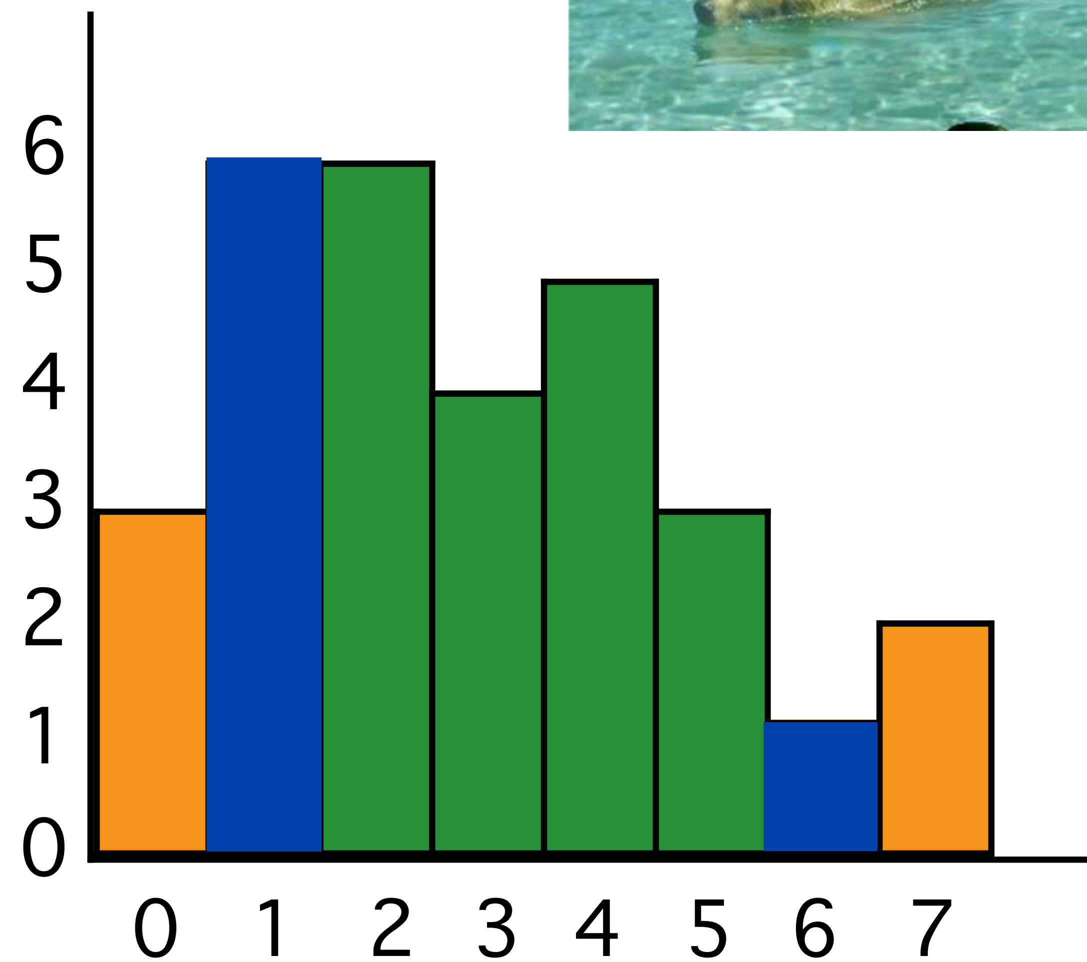
1D example recap

- **Copy**





to



==>

$$\begin{pmatrix} 4 & -2 & 0 & 0 \\ -2 & 4 & -2 & 0 \\ 0 & -2 & 4 & -2 \\ 0 & 0 & -2 & 4 \end{pmatrix} \begin{pmatrix} f_2 \\ f_3 \\ f_4 \\ f_5 \end{pmatrix} = \begin{pmatrix} 16 \\ -6 \\ 6 \\ 2 \end{pmatrix}$$

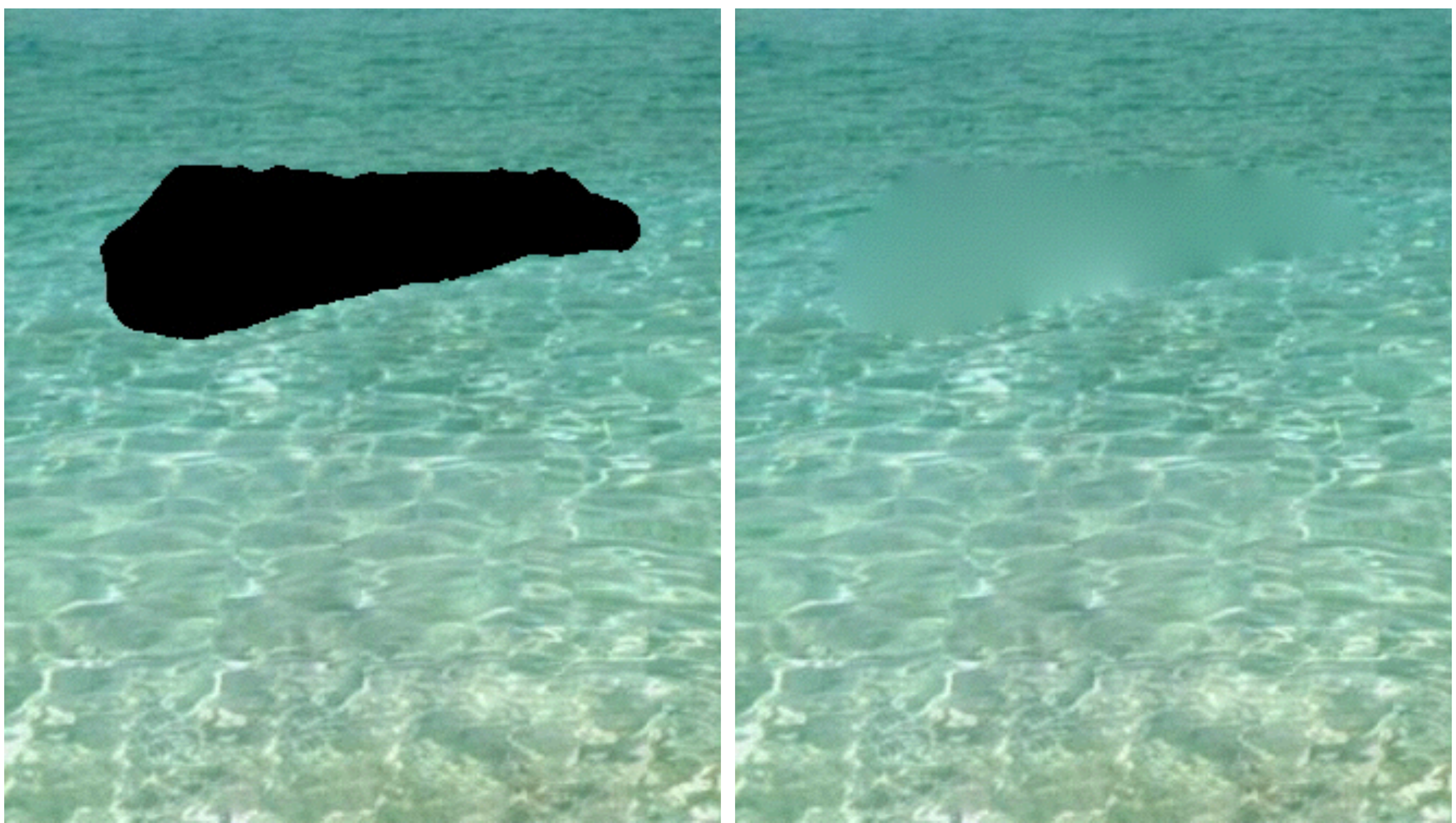
$$\begin{pmatrix} f_2 \\ f_3 \\ f_4 \\ f_5 \end{pmatrix} = \begin{pmatrix} 6 \\ 4 \\ 5 \\ 3 \end{pmatrix}$$

Questions?

Membrane interpolation

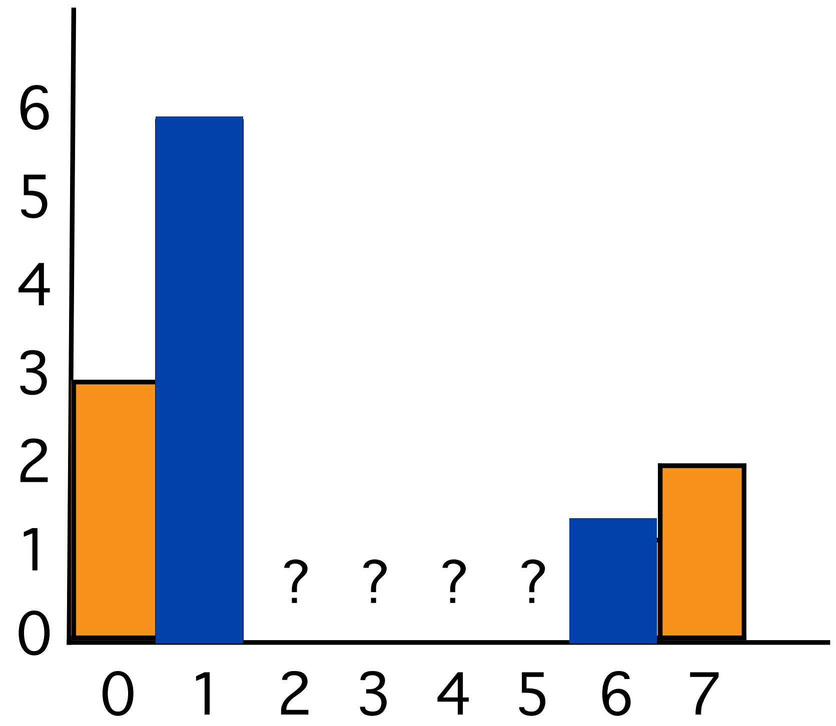
- **What if *v* is null?**
- **Laplace equation (a.k.a. membrane equation)**

$$\min_{f} \iint_{\Omega} |\nabla f|^2 \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$



1D example: minimization

- **Minimize derivatives to interpolate**



Min $(f_2-f_1)^2$

$+ (f_3-f_2)^2$

$+ (f_4-f_3)^2$

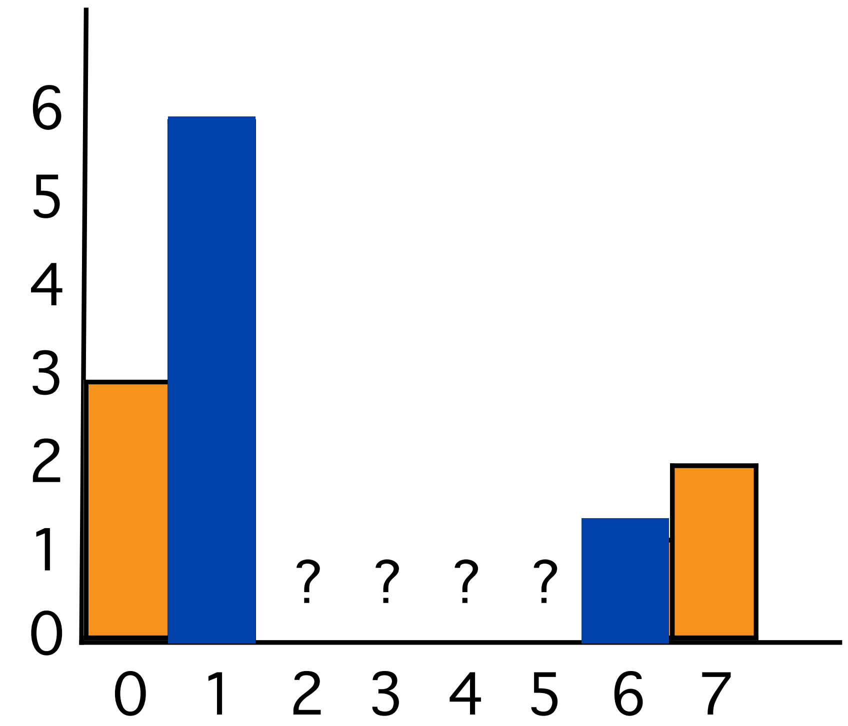
$+ (f_5-f_4)^2$

$+ (f_6-f_5)^2$

With
$f_1=6$
$f_6=1$

1D example: derivatives

- **Minimize derivatives to interpolate**



Min (f_2^2+36-12f_2

+ f_3^2+f_2^2-2f_3f_2

+ f_4^2+f_3^2-2f_3f_4

+ f_5^2+f_4^2-2f_5f_4

+ f_5^2+1-2f_5)

Denote it Q

$$\frac{dQ}{df_2} = 2f_2 + 2f_2 - 2f_3 - 12$$

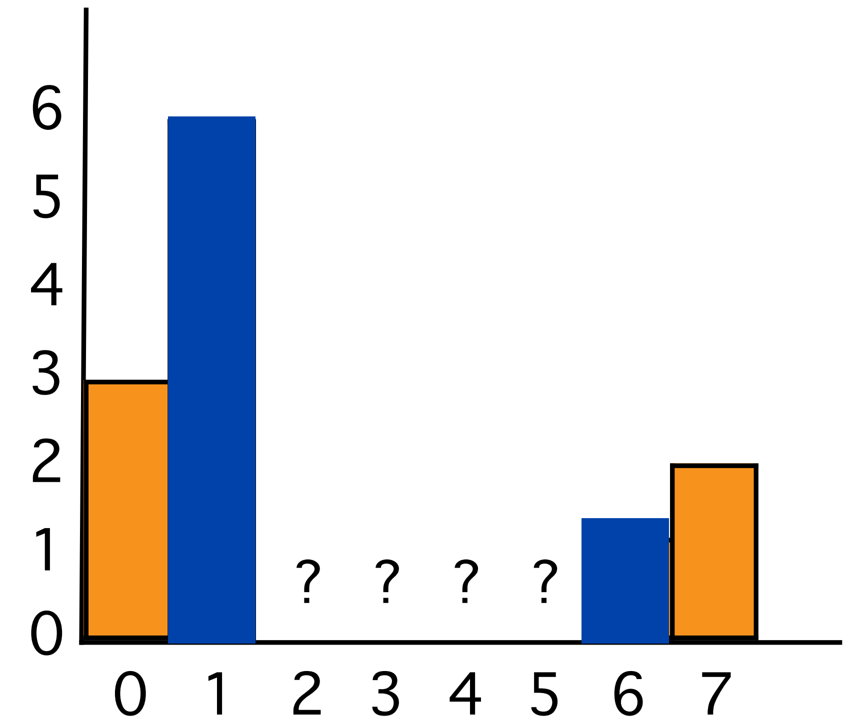
$$\frac{dQ}{df_3} = 2f_3 - 2f_2 + 2f_3 - 2f_4$$

$$\frac{dQ}{df_4} = 2f_4 - 2f_3 + 2f_4 - 2f_5$$

$$\frac{dQ}{df_5} = 2f_5 - 2f_4 + 2f_5 - 2$$

1D example: set derivatives to zero

- **Minimize derivatives to interpolate**



$$\frac{dQ}{df_2} = 2f_2 + 2f_2 - 2f_3 - 12$$

$$\frac{dQ}{df_3} = 2f_3 - 2f_2 + 2f_3 - 2f_4$$

$$\frac{dQ}{df_4} = 2f_4 - 2f_3 + 2f_4 - 2f_5$$

$$\frac{dQ}{df_5} = 2f_5 - 2f_4 + 2f_5 - 2$$

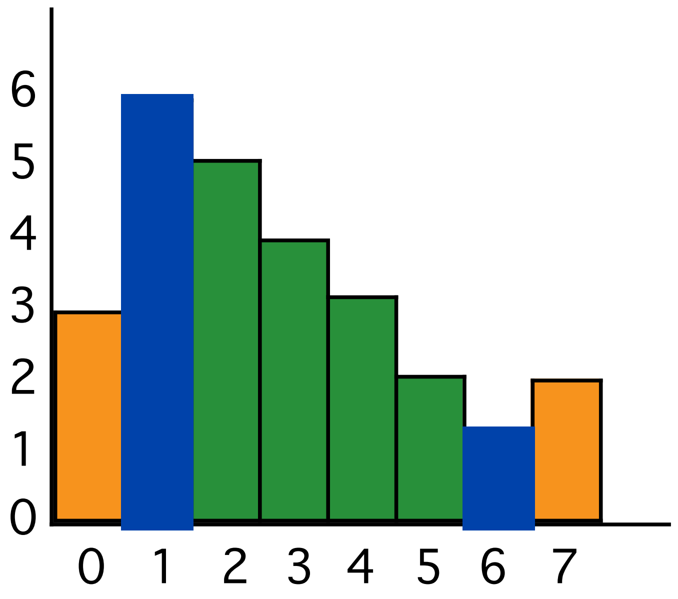
$$\Longrightarrow \begin{pmatrix} 4 & -2 & 0 & 0 \\ -2 & 4 & -2 & 0 \\ 0 & -2 & 4 & -2 \\ 0 & 0 & -2 & 4 \end{pmatrix} \begin{pmatrix} f_2 \\ f_3 \\ f_4 \\ f_5 \end{pmatrix} = \begin{pmatrix} 12 \\ 0 \\ 0 \\ 2 \end{pmatrix}$$

1D example

- **Minimize derivatives to interpolate**

- **Pretty much says that second derivative should be zero**

(-1 2 -1)
is a second derivative filter

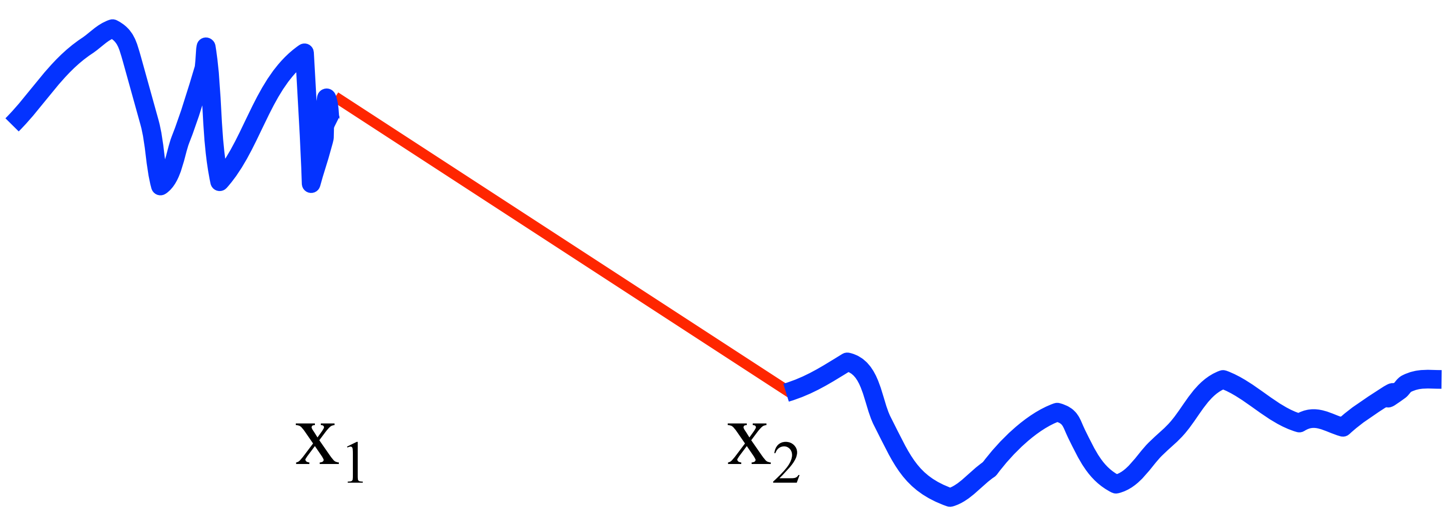


$$\begin{pmatrix} 4 & -2 & 0 & 0 \\ -2 & 4 & -2 & 0 \\ 0 & -2 & 4 & -2 \\ 0 & 0 & -2 & 4 \end{pmatrix} \begin{pmatrix} f_2 \\ f_3 \\ f_4 \\ f_5 \end{pmatrix} = \begin{pmatrix} 12 \\ 0 \\ 0 \\ 2 \end{pmatrix}$$

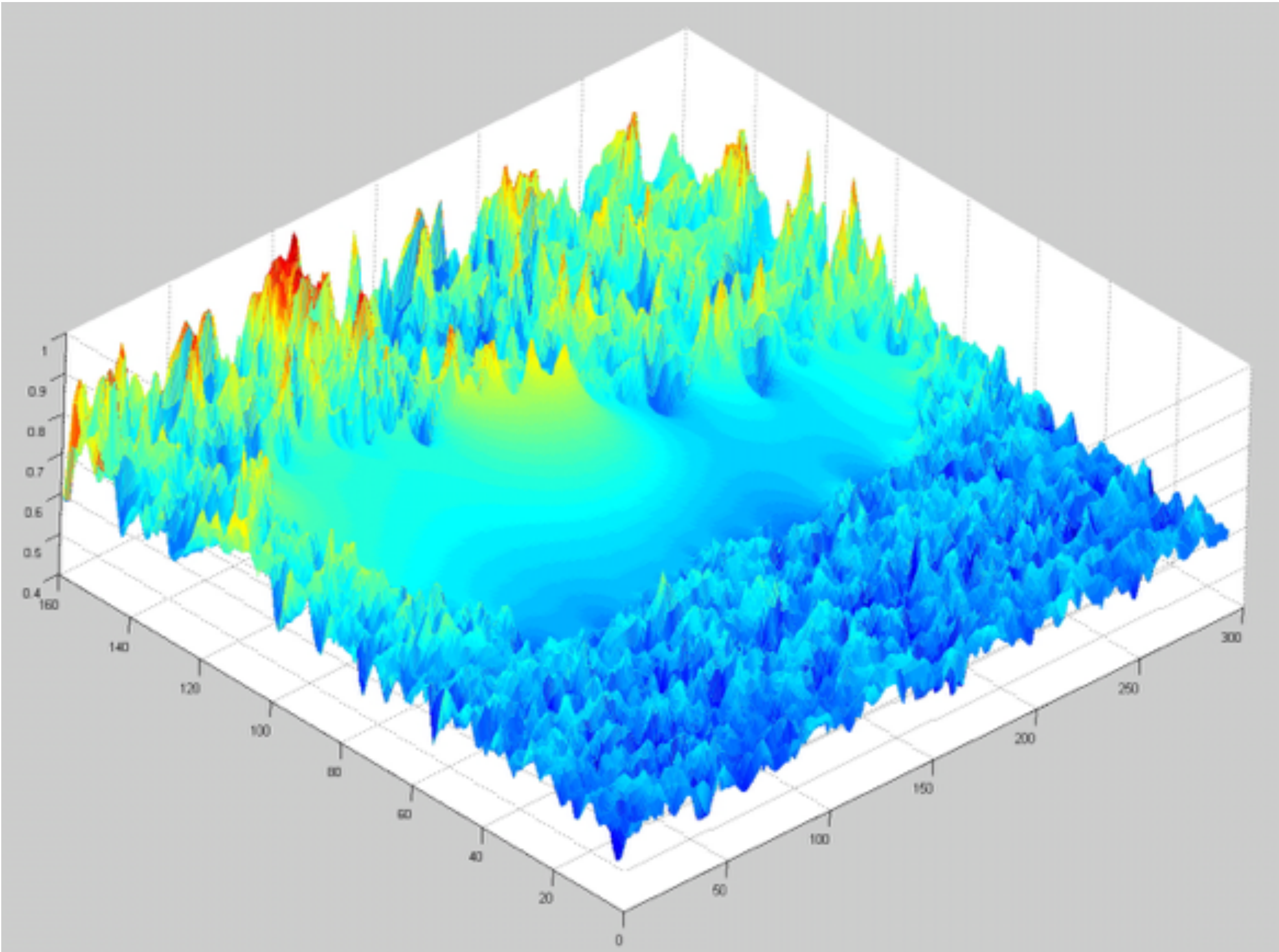
$$\begin{pmatrix} f_2 \\ f_3 \\ f_4 \\ f_5 \end{pmatrix} = \begin{pmatrix} 5 \\ 4 \\ 3 \\ 2 \end{pmatrix}$$

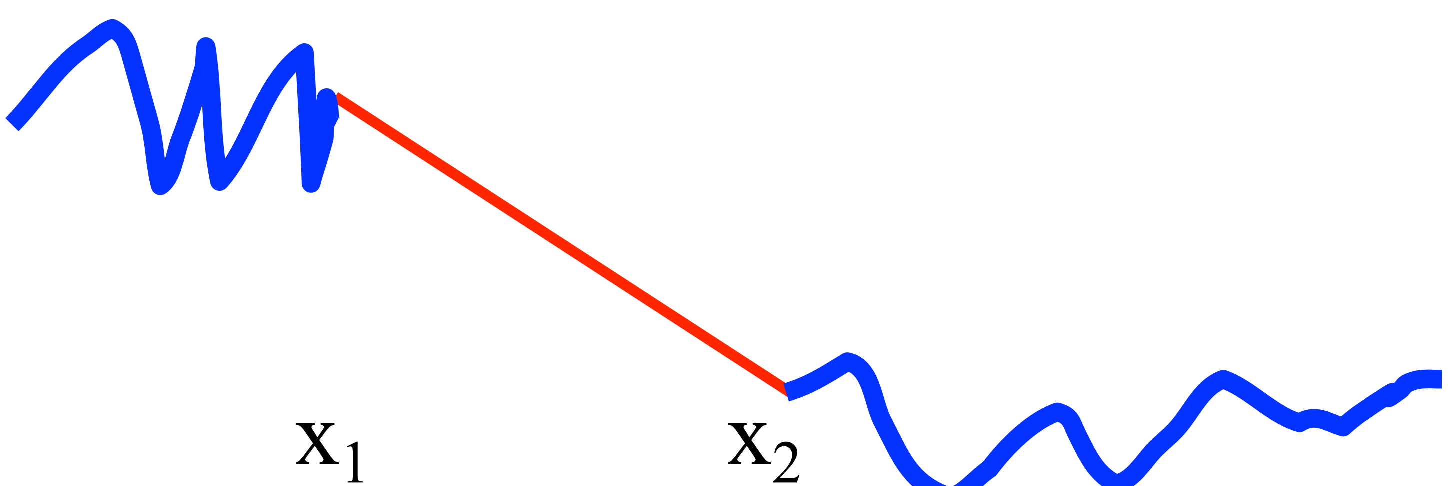
Intuition

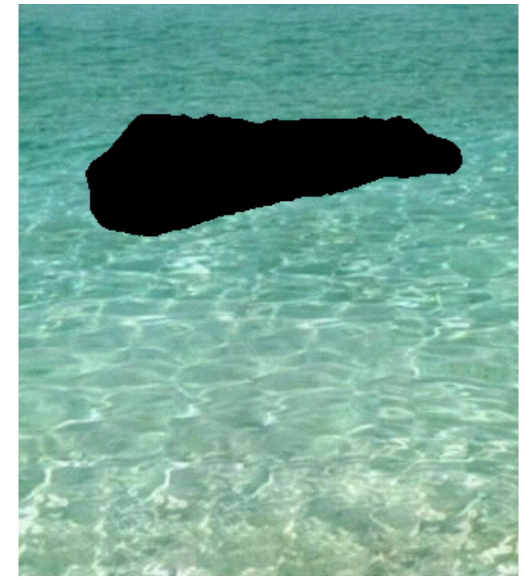
- **In 1D; just linear interpolation!**
- **Locally, if the second derivative was not zero, this would mean that the first derivative is varying, which is bad since we want $(\nabla f)^2$ to be minimized**
- **Note that, in 1D: by setting f'', we leave two degrees of freedom. This is exactly what we need to control the boundary condition at x_1 and x_2**



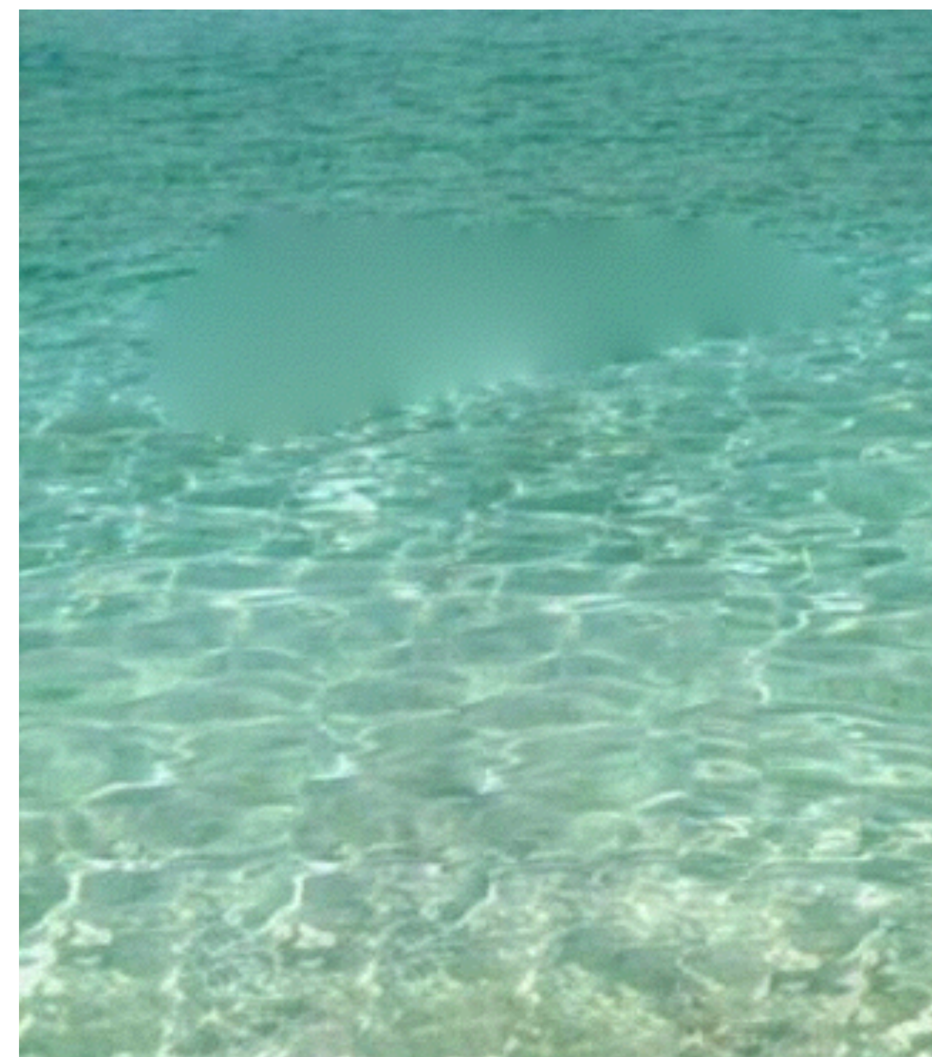
In 2D: membrane interpolation







Not as simple



Questions?

What if *v* is not null?

sources/destinations

seamless cloning

What if *v* is not null?

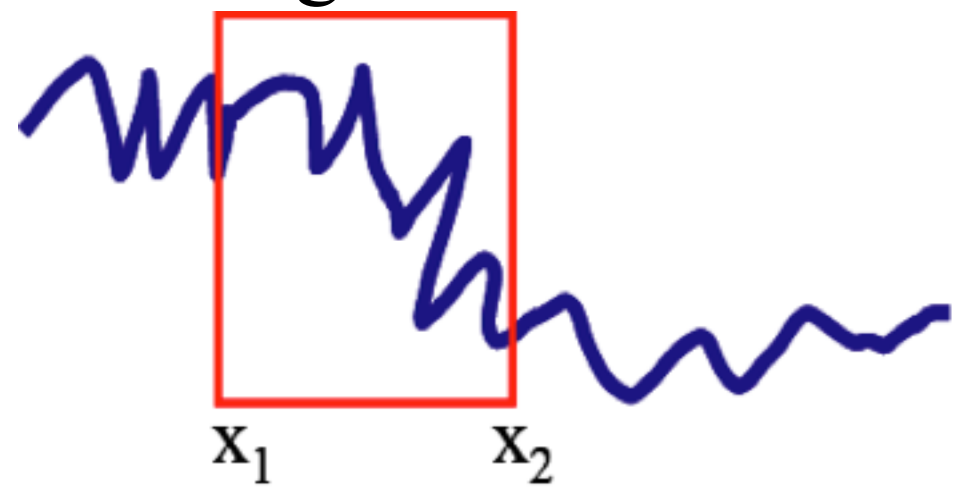
- **1D case**

Seamlessly paste

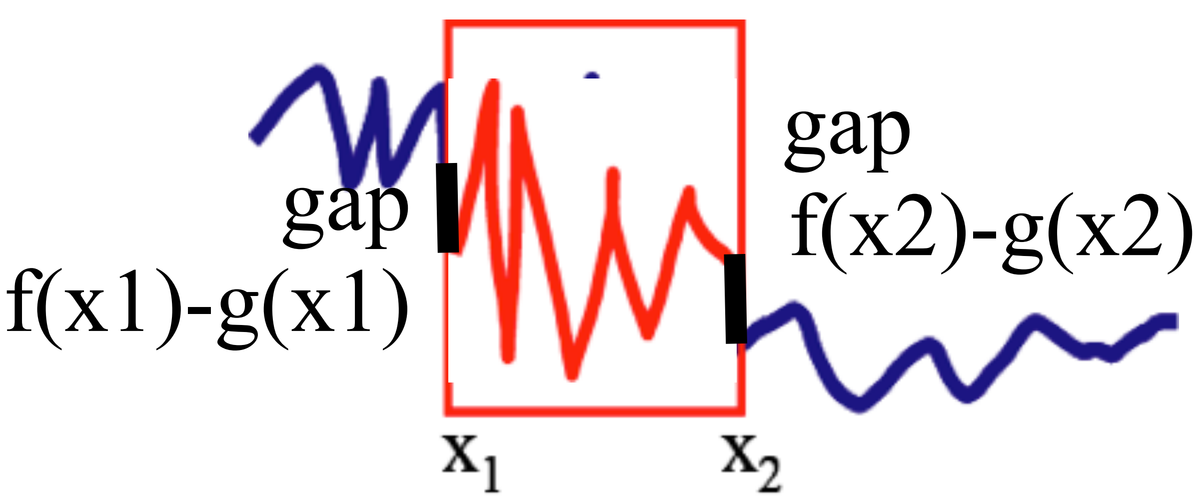
source g

onto

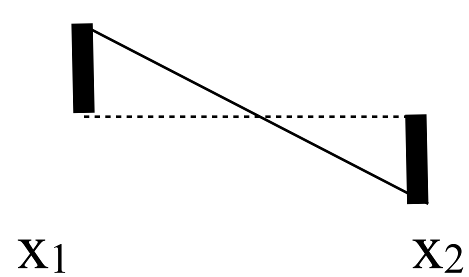
target f*



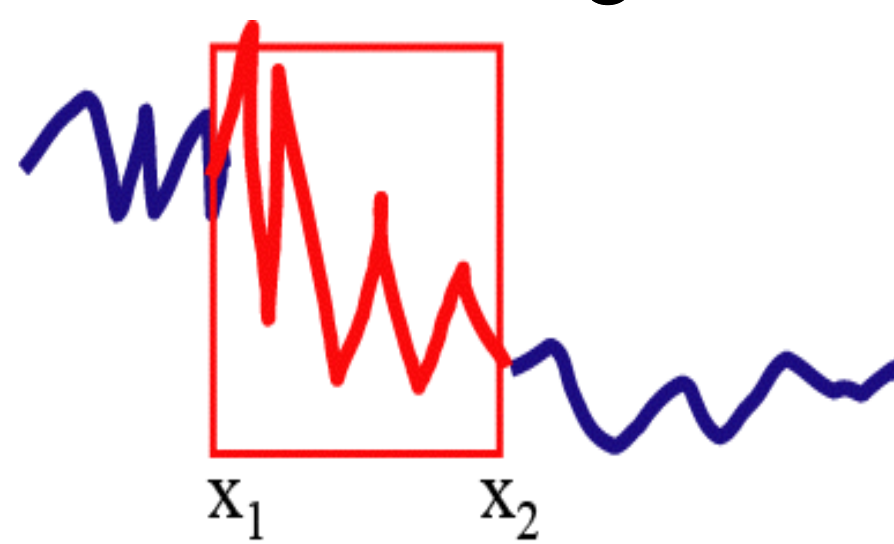
Just add a linear function so that the boundary condition is respected



correction $\hat{f}$



solution $f=\hat{f}+g$



Matrix structure

$$\begin{bmatrix} 4 & -2 & 0 & 0 \\ -2 & 4 & -2 & 0 \\ 0 & -2 & 4 & -2 \\ 0 & 0 & -2 & 4 \end{bmatrix} \begin{bmatrix} f_2 \\ f_3 \\ f_4 \\ f_5 \end{bmatrix} = \begin{bmatrix} 16 \\ -6 \\ 6 \\ 2 \end{bmatrix}$$

denote this matrix A

A is large!

- (# cols = num pixels) x (# rows = num pixels)

but system is sparse!

- most coefficients will be zero

Solution methods

Direct solve (pseudoinverse)

- can be numerically unstable and inefficient for large systems

Orthogonal decomposition methods

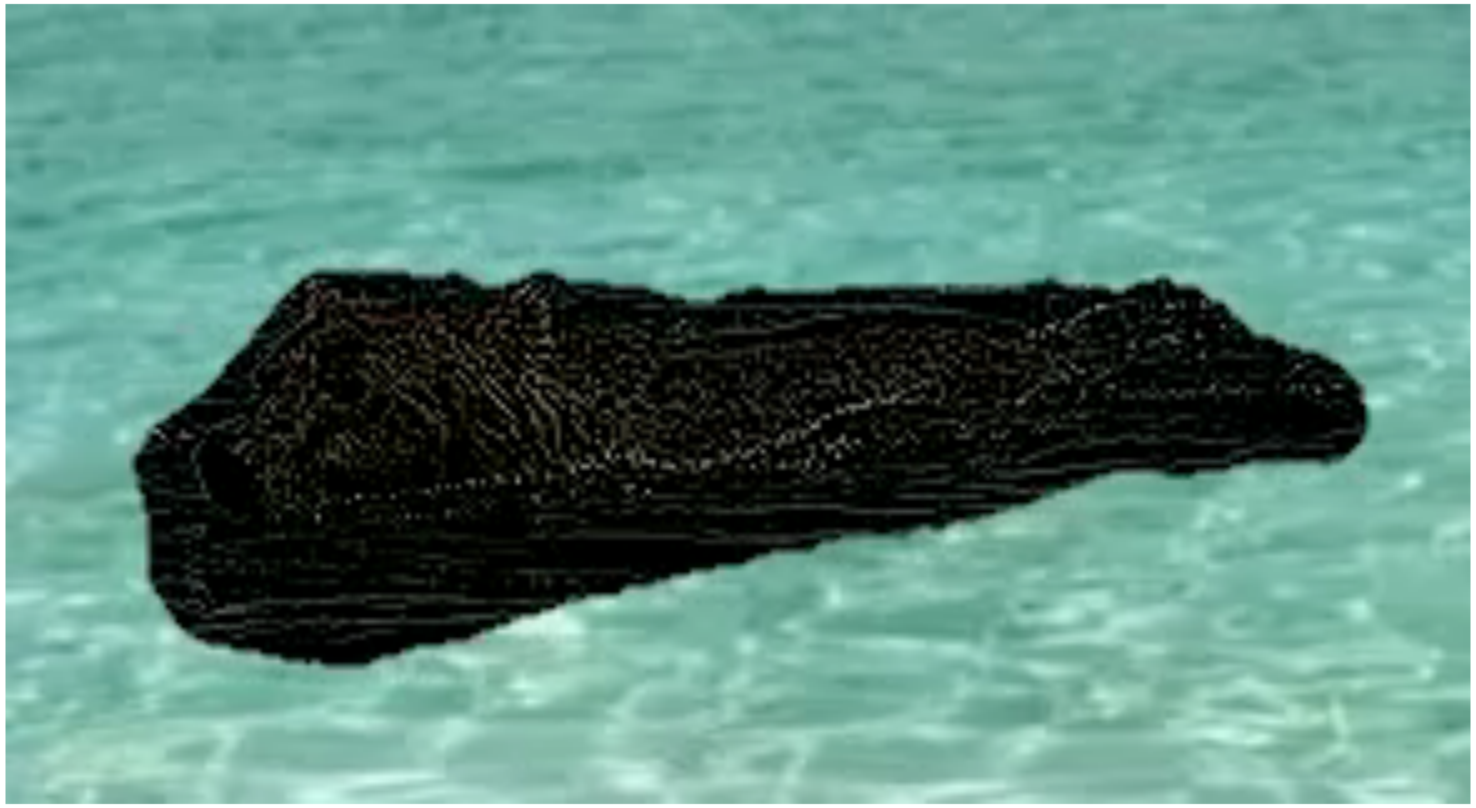
- more stable, but can be slower. e.g. QR decomp.

Iterative methods

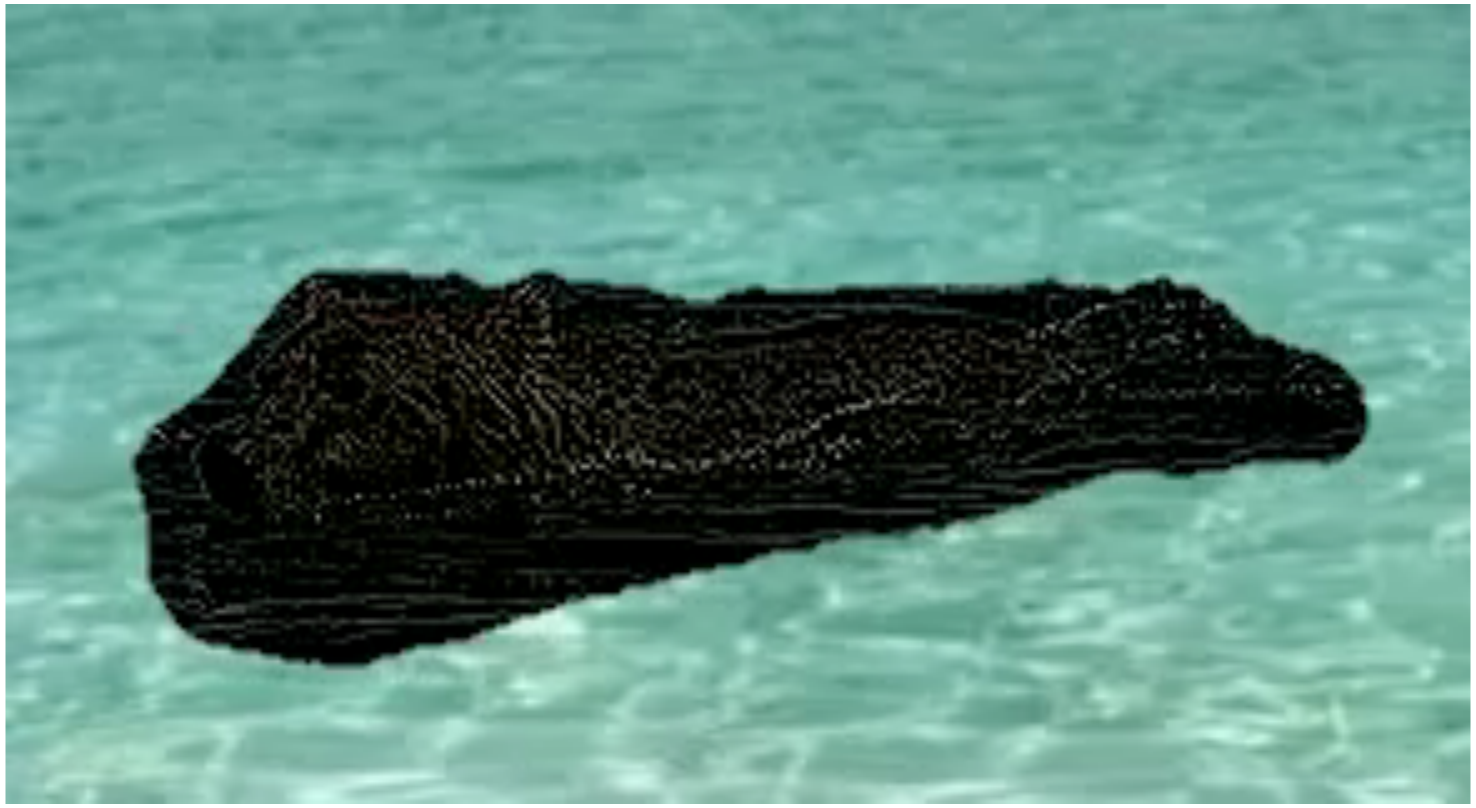
- e.g. steepest descent, conjugate gradients
- efficient for sparse matrices
- needs to be symmetric, positive-definite

Convergence

gradient descent



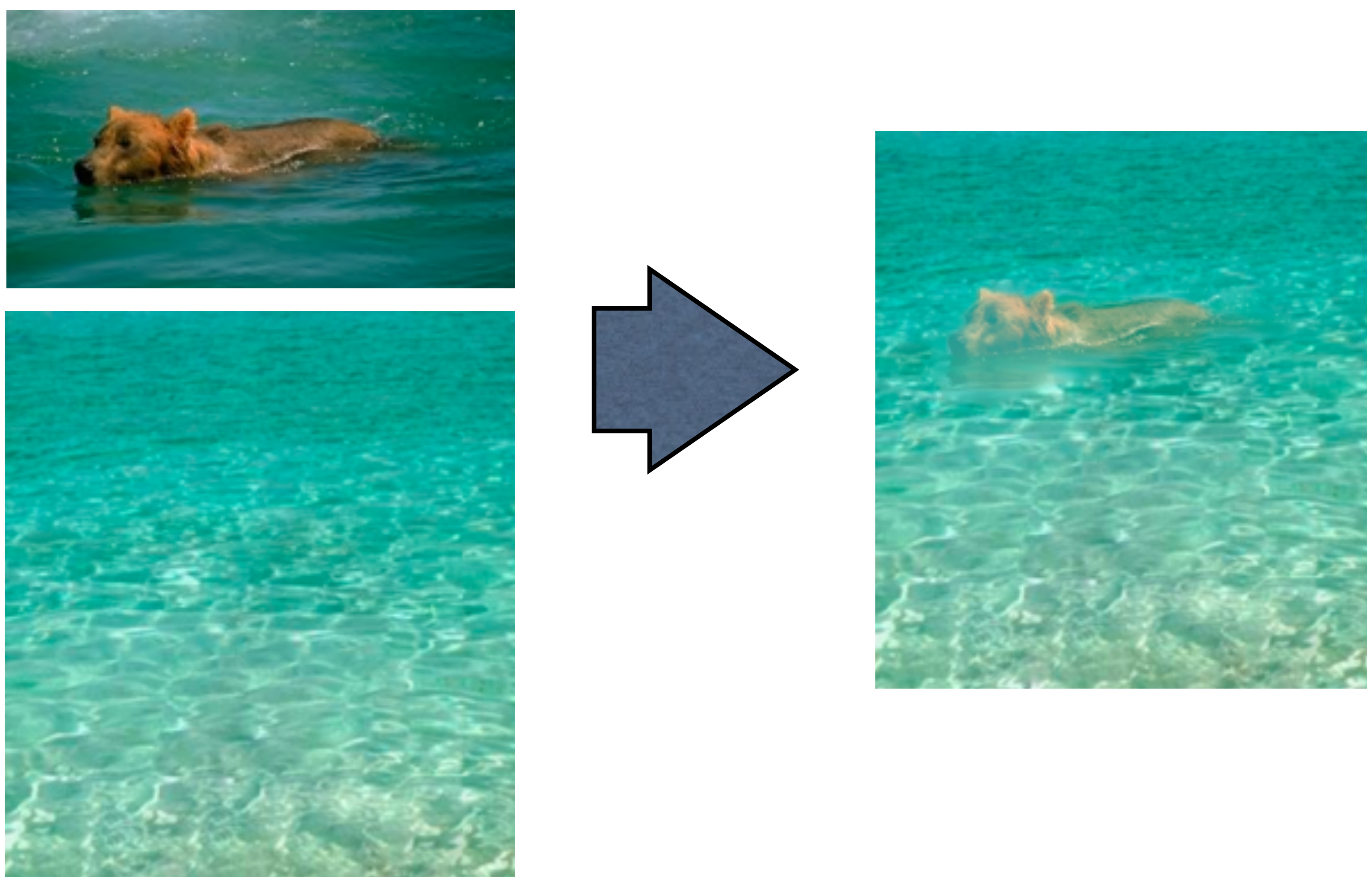
conjugate gradients



Bells and whistles

Contrast problem

- Contrast is a multiplicative quantity
- With Poisson, we try to reproduce linear differences
- Loss of contrast if pasting from dark to bright



Contrast preservation: use the log

Poisson in linear color space

Poisson in log color space

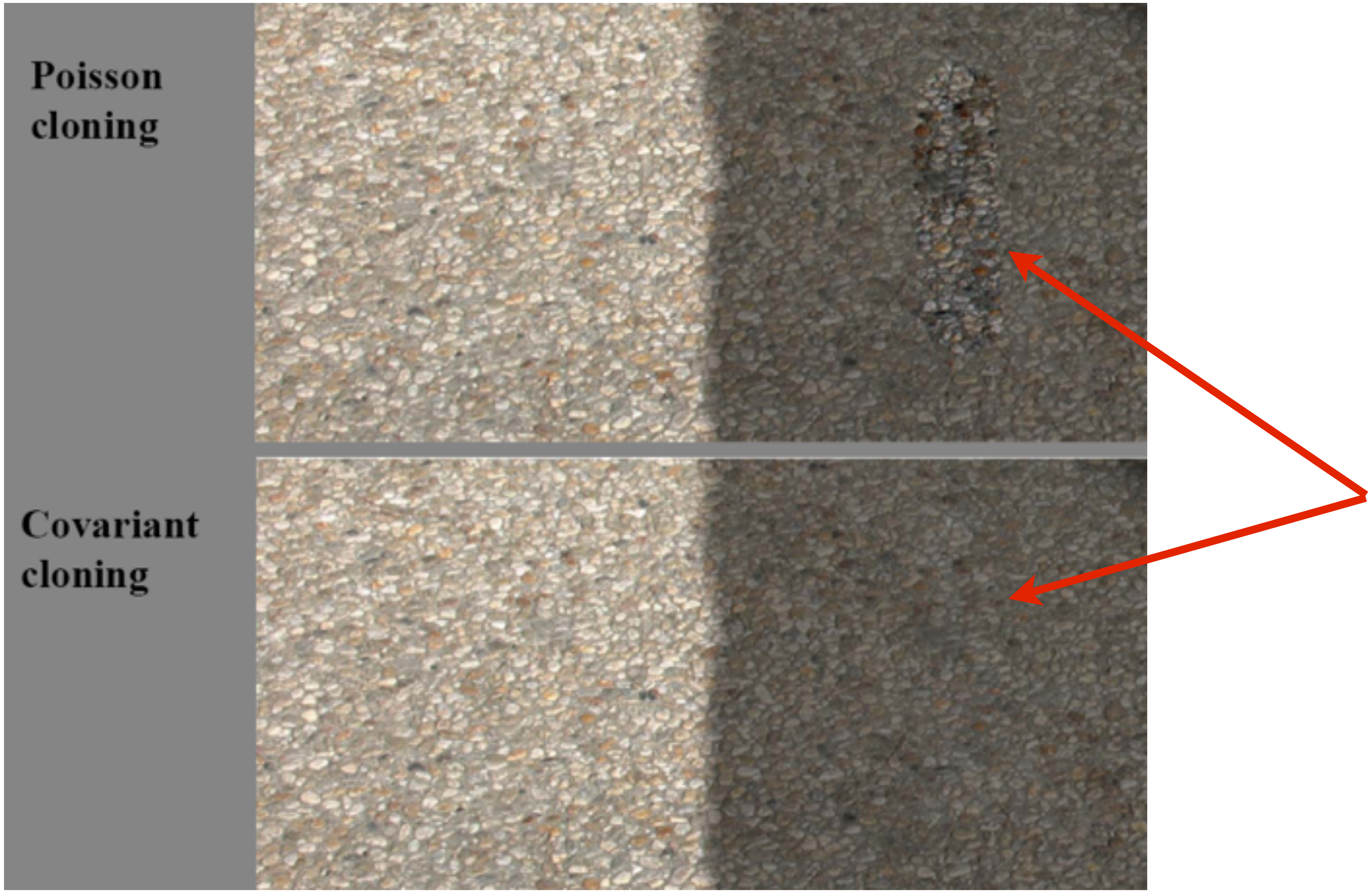
- see A Perception-based Color Space for Illumination-invariant Image Processing

  http://www.eecs.harvard.edu/~hchong/thesis/color_siggraph08.pdf

- Or use covariant derivatives (next slides)

Covariant derivatives & Photoshop

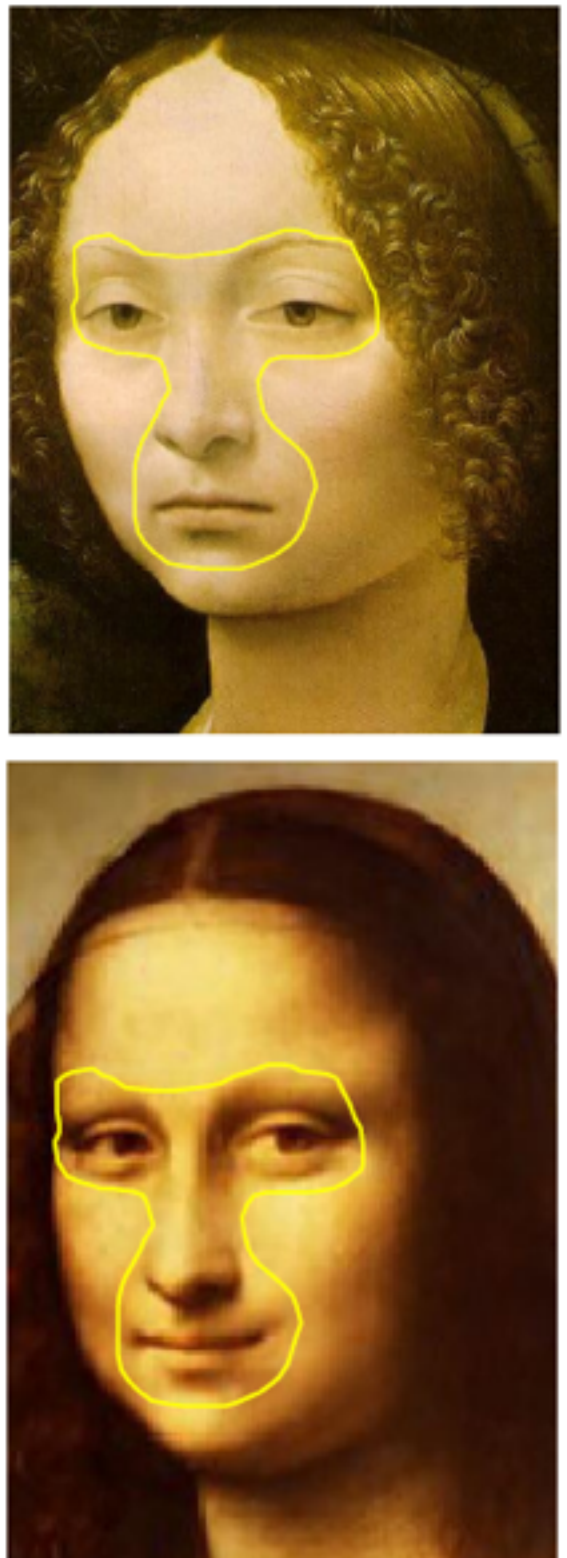
- **Photoshop Healing brush**
- **Developed independently from Poisson editing by Todor Georgiev (Adobe)**



From Todor Georgiev's slides http://photo.csail.mit.edu/posters/todor_slides.pdf

Eye candy

Result (eye candy)



source/destination

cloning

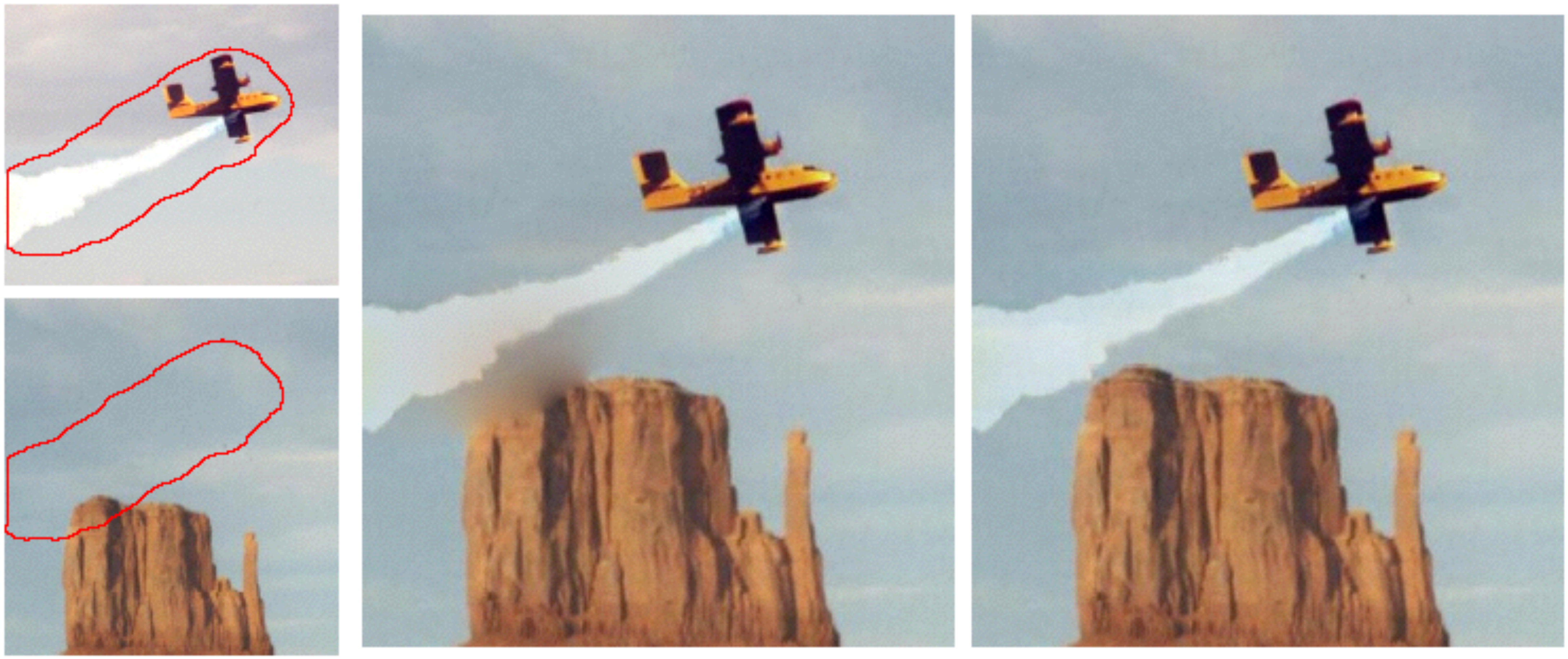
seamless cloning

sources | destinations | cloning | seamless cloning

Figure 2: **Concealment**. By importing seamlessly a piece of the background, complete objects, parts of objects, and undesirable artifacts can easily be hidden. In both examples, multiple strokes (not shown) were used.

Manipulate the gradient

- **Mix gradients of g & f: take the max**



source/destination | seamless cloning | mixed seamless cloning

Figure 8: **Inserting one object close to another**. With seamless cloning, an object in the destination image touching the selected region Ω bleeds into it. Bleeding is inhibited by using mixed gradients as the guidance field.

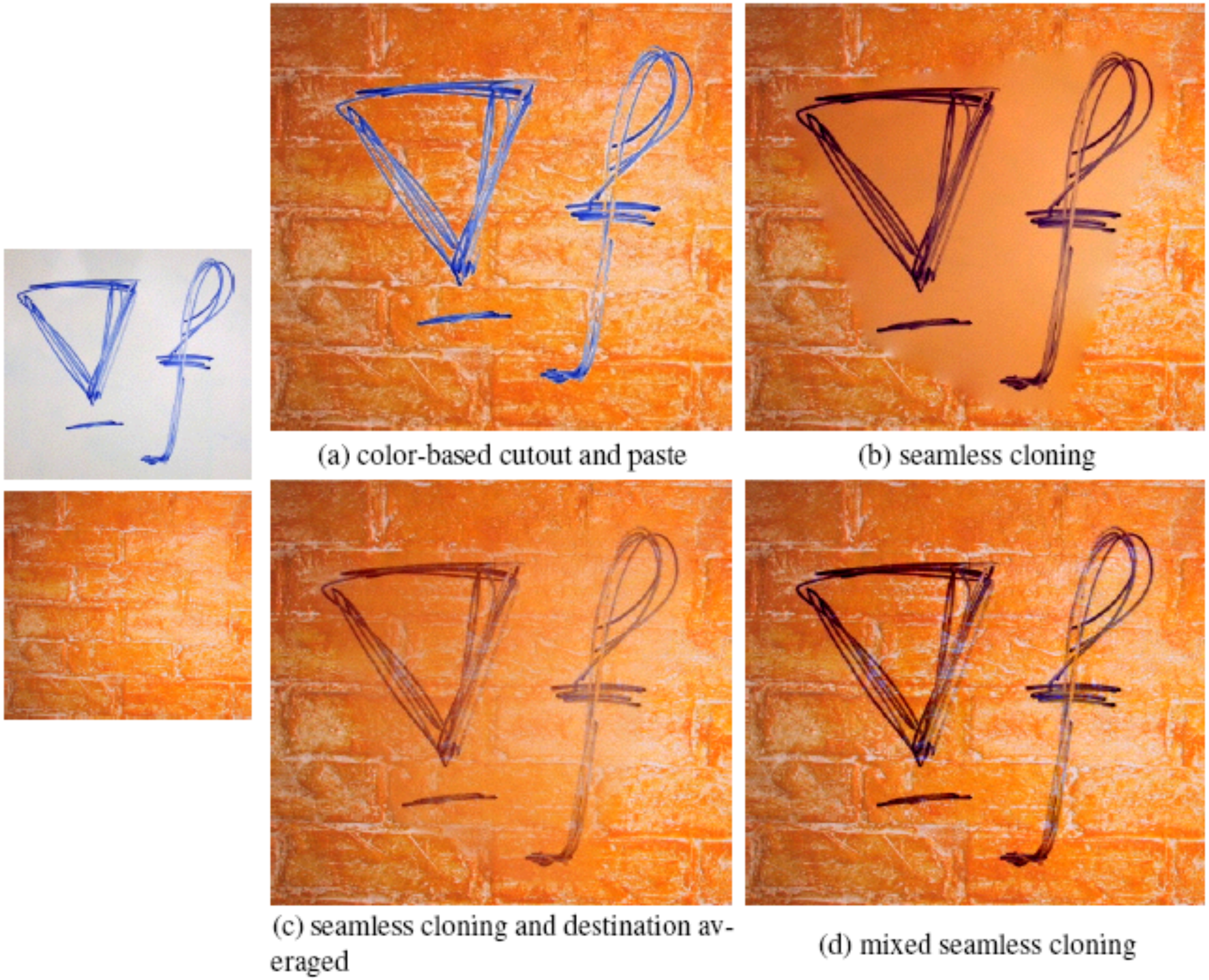
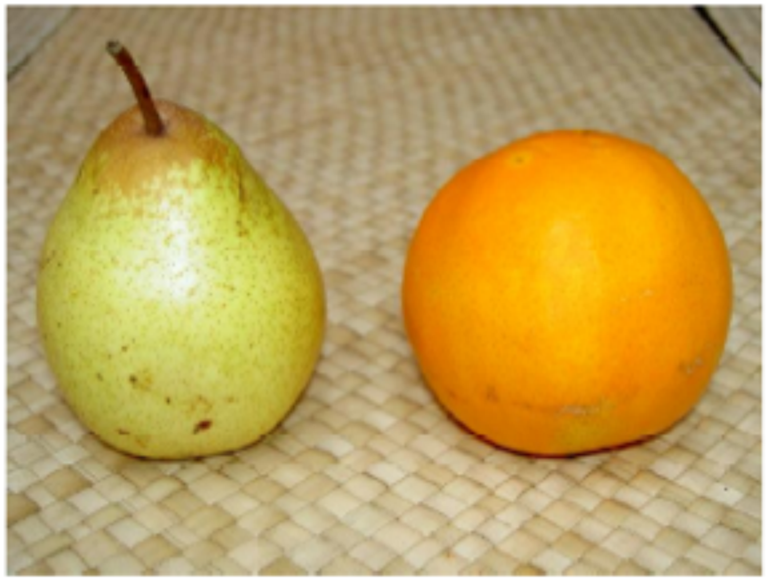


Figure 6: **Inserting objects with holes**. (a) The classic method, color-based selection and alpha masking might be time consuming and often leaves an undesirable halo; (b-c) seamless cloning, even averaged with the original image, is not effective; (d) mixed seamless cloning based on a loose selection proves effective.



swapped textures

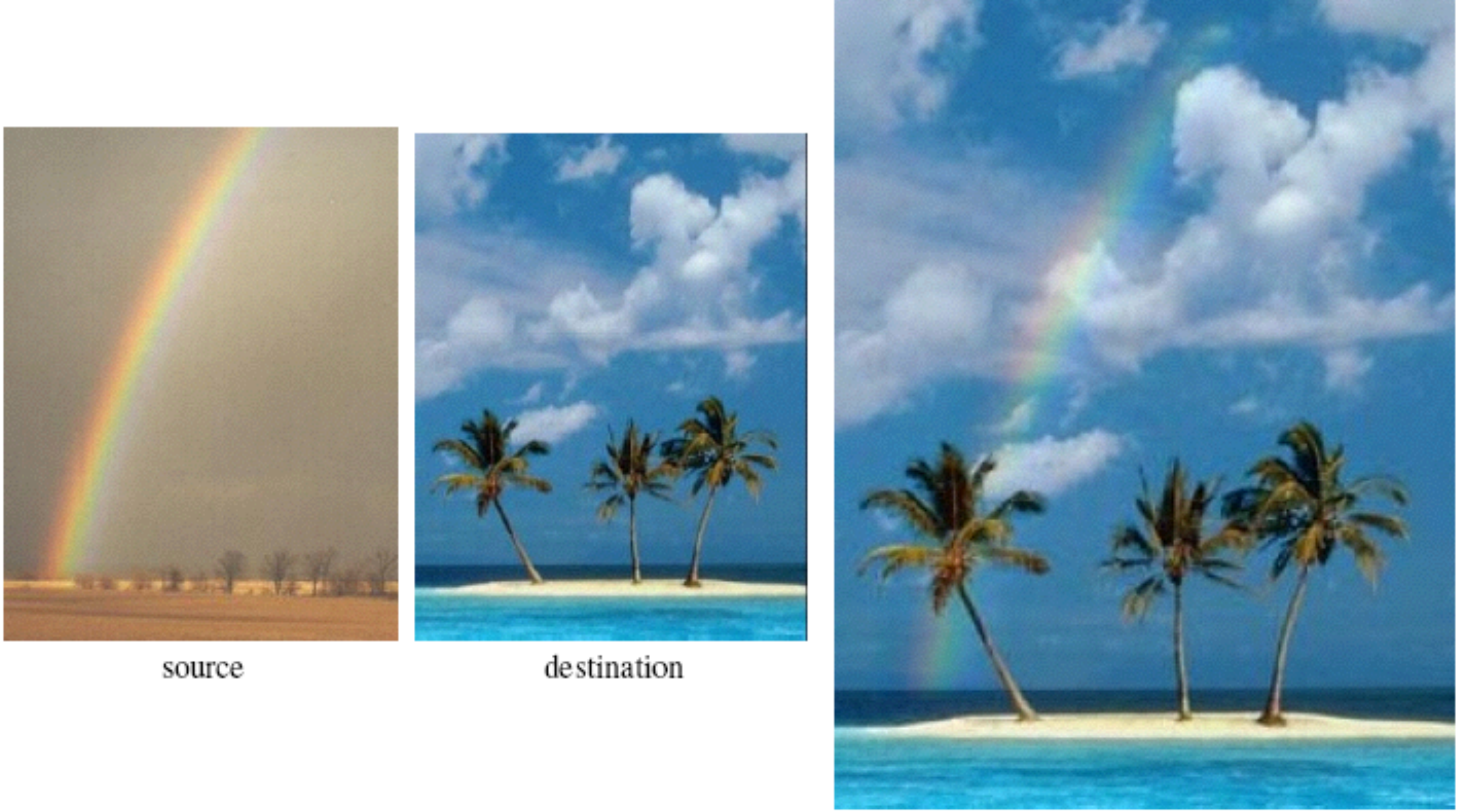


Figure 7: **Inserting transparent objects**. Mixed seamless cloning facilitates the transfer of partly transparent objects, such as the rainbow in this example. The non-linear mixing of gradient fields picks out whichever of source or destination structure is the more salient at each location.

Questions?

Slide credits

Frédo Durand

Steve Marschner

Matthias Zwicker