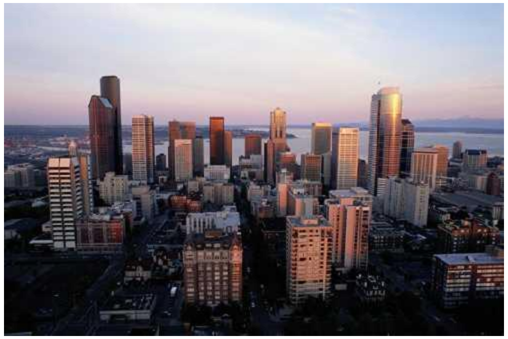
Image Blending and Compositing

15-463: Computational Photography
Alexei Efros, CMU, Fall 2010

Image Compositing

Compositing Procedure

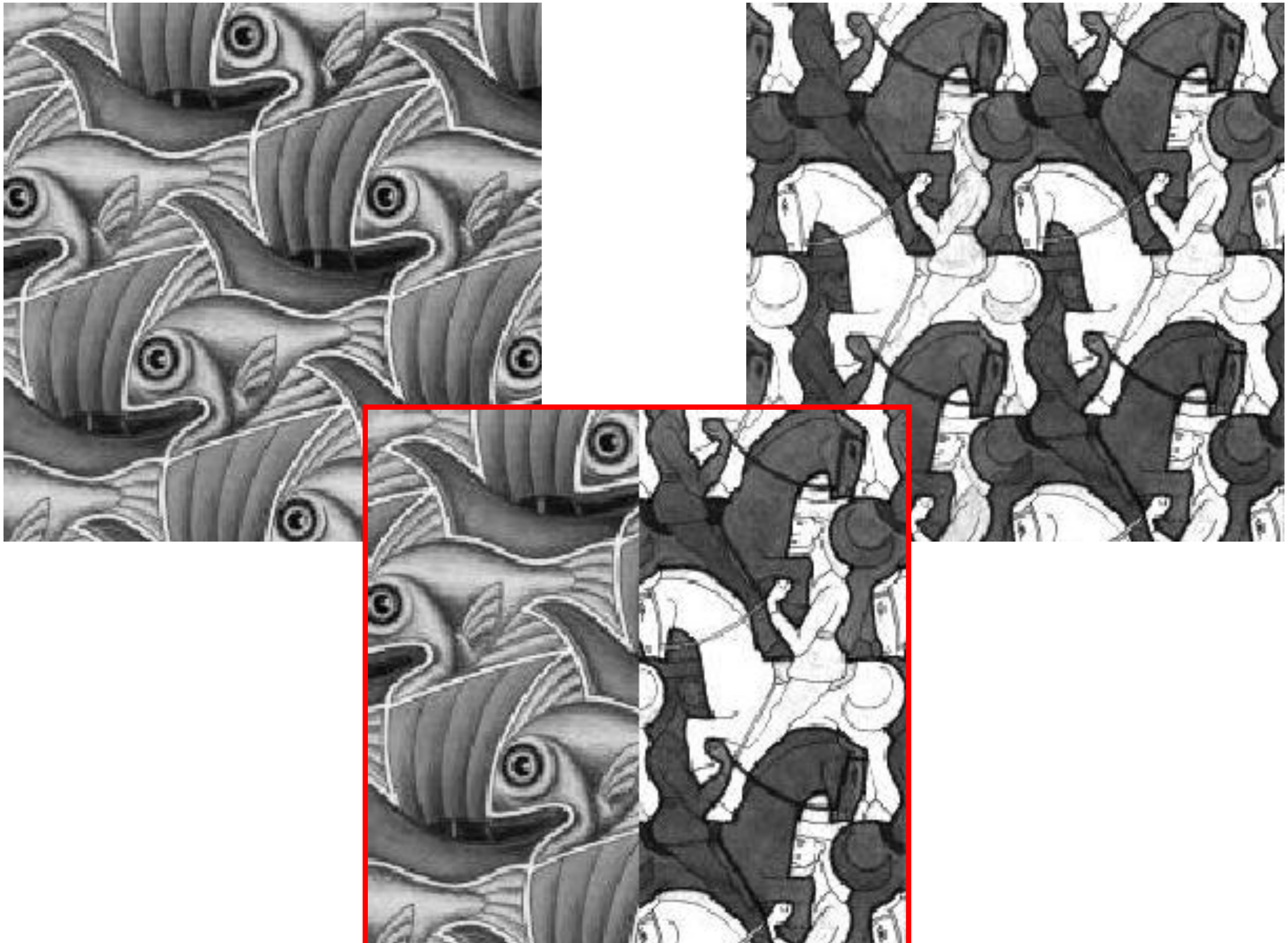
1. Extract Sprites (e.g using *Intelligent Scissors* in Photoshop)



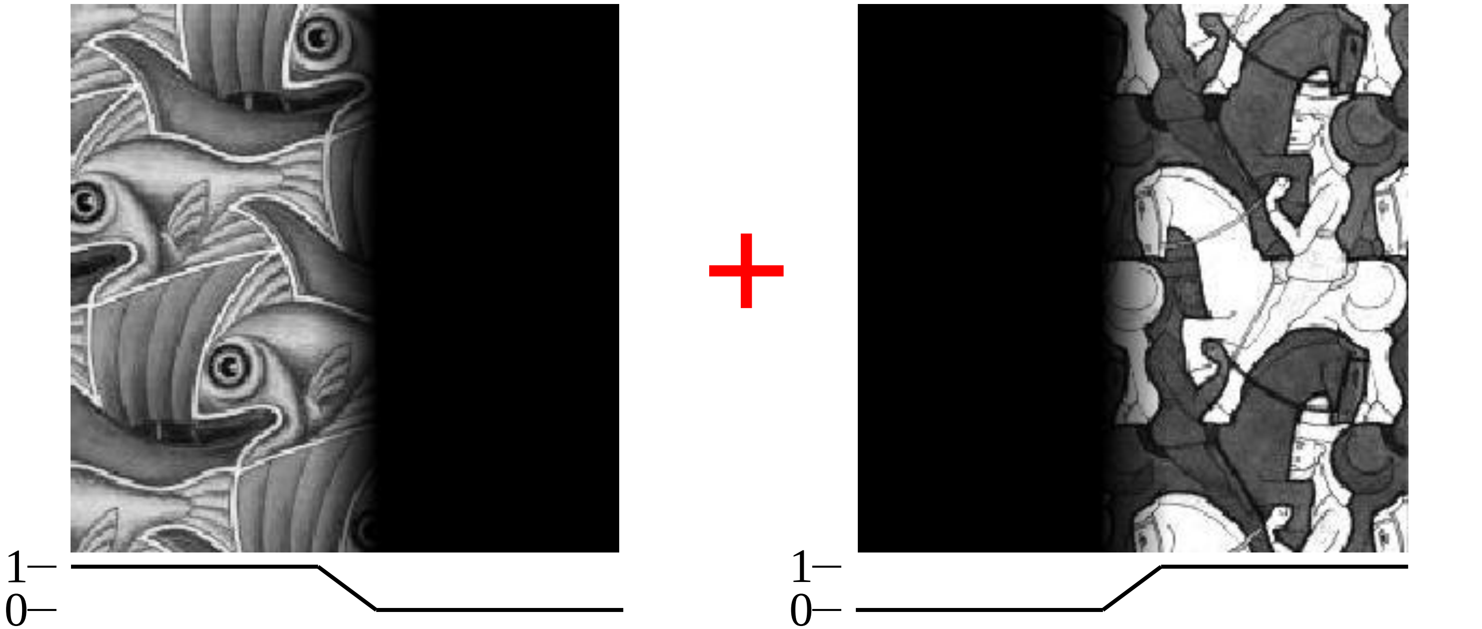

2. Blend them into the composite (in the right order)

Composite by
David Dewey

Need blending



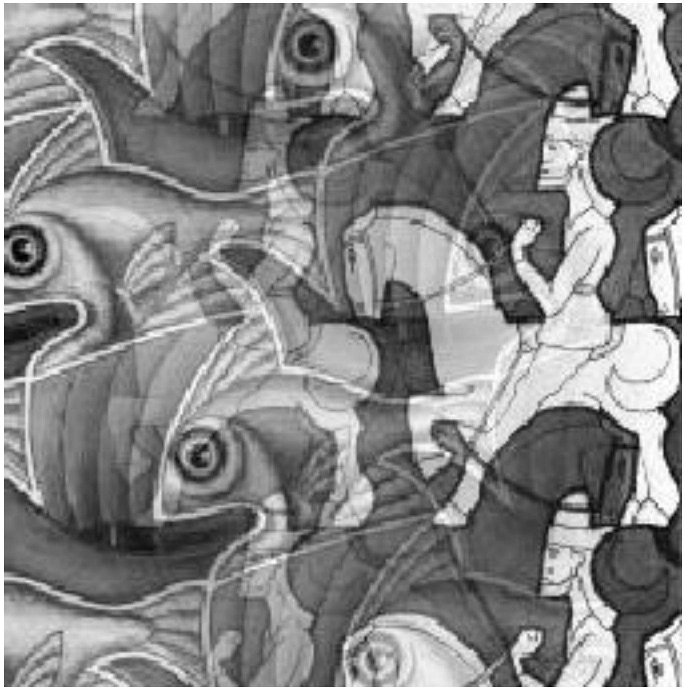
Alpha Blending / Feathering

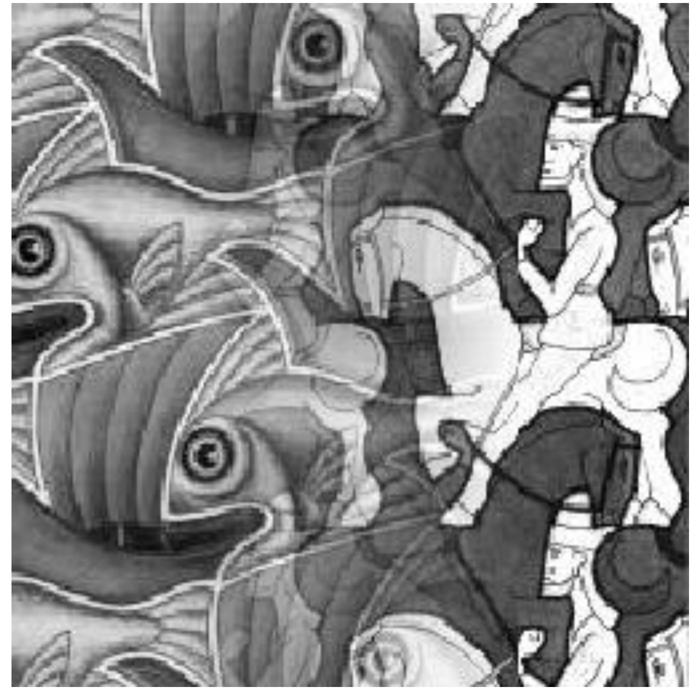


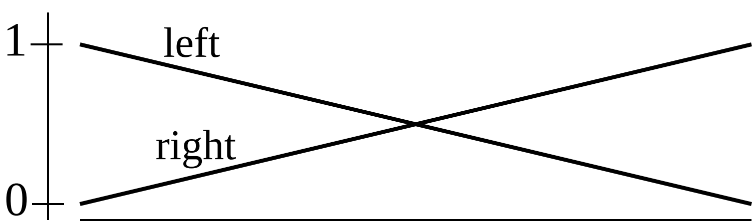
=

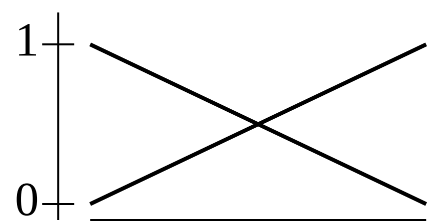
$$I_{blend} = \alpha I_{left} + (1\text{-}\alpha) I_{right}$$

Affect of Window Size

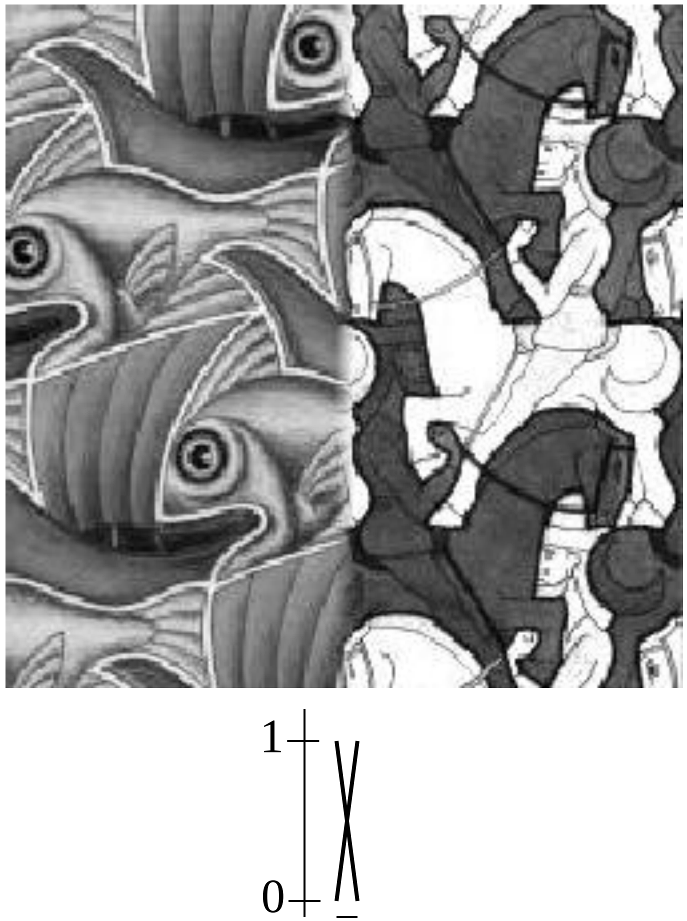


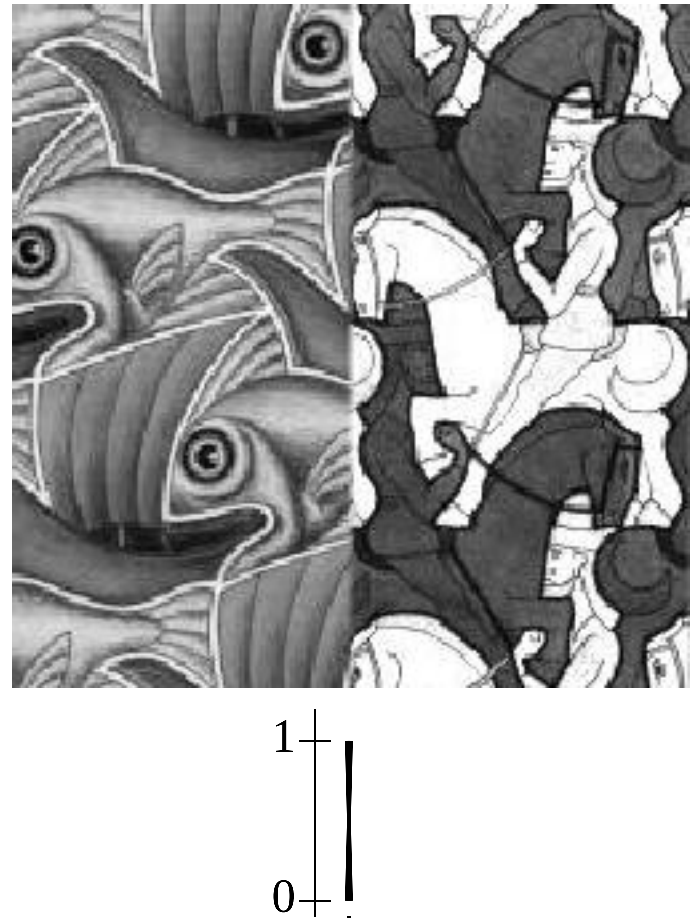




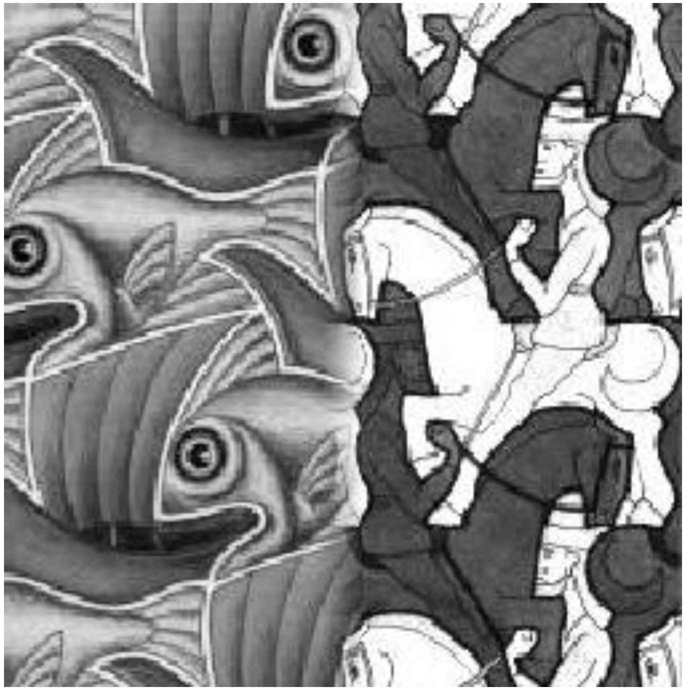


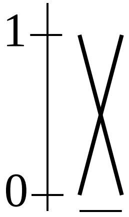
Affect of Window Size





Good Window Size





“Optimal” Window: smooth but not ghosted

What is the Optimal Window?

To avoid seams

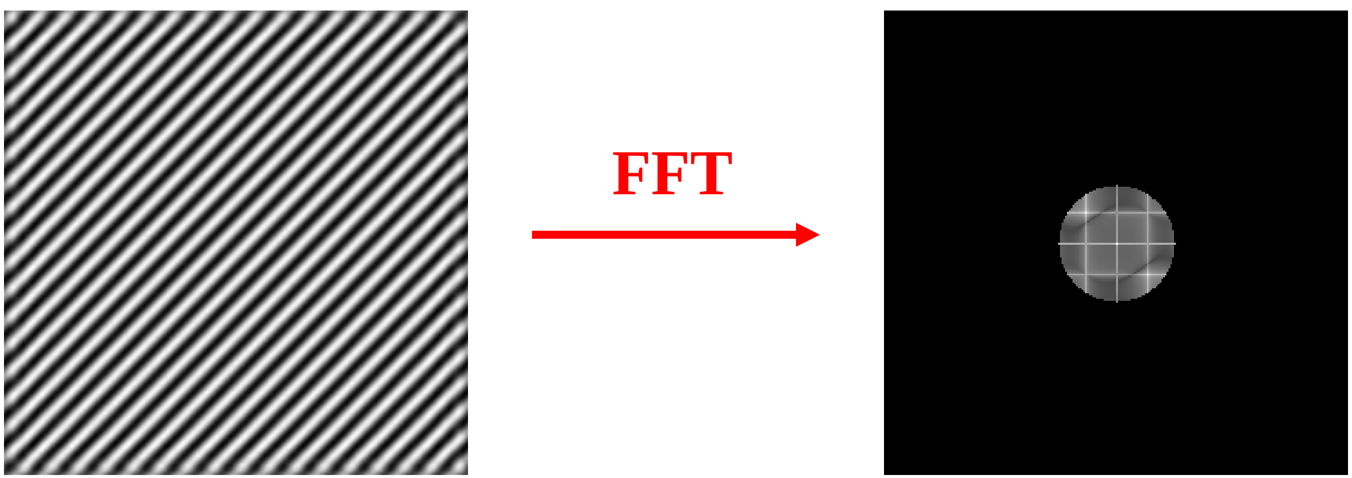
- window = size of largest prominent feature

To avoid ghosting

- window <= 2*size of smallest prominent feature

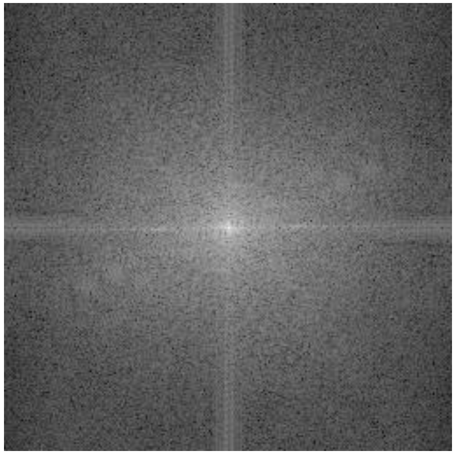
Natural to cast this in the *Fourier domain*

- largest frequency <= 2*size of smallest frequency
- image frequency content should occupy one “octave” (power of two)



What if the Frequency Spread is Wide

FFT



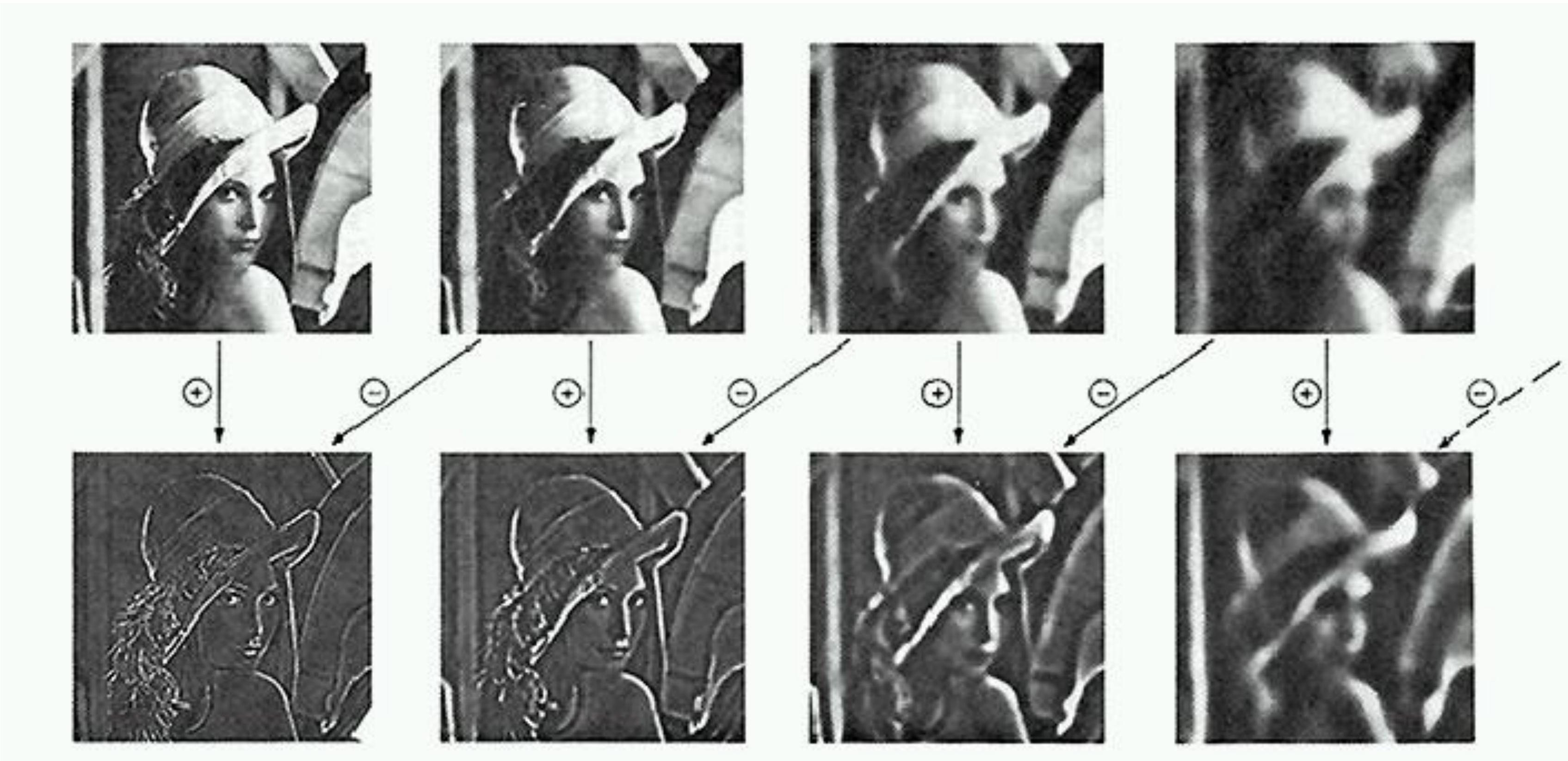
Idea (Burt and Adelson)

- Compute F_{left} = FFT(I_{left}), F_{right} = FFT(I_{right})
- Decompose Fourier image into octaves (bands)
  - $F_{left} = F_{left}^1 + F_{left}^2 + \ldots$
- Feather corresponding octaves F_{left}^i with F_{right}^i
  - Can compute inverse FFT and feather in spatial domain
- Sum feathered octave images in frequency domain

Better implemented in *spatial domain*

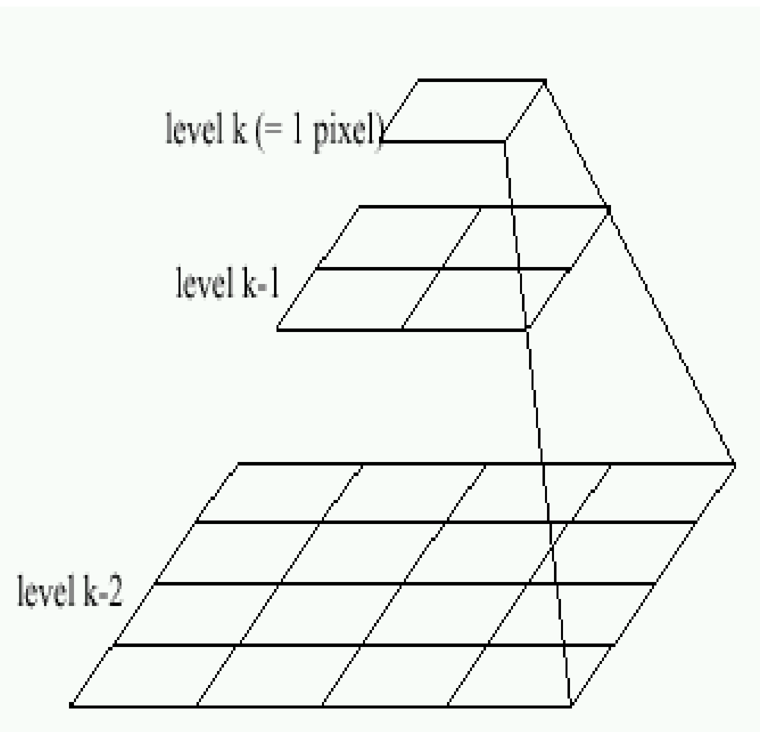
Octaves in the Spatial Domain

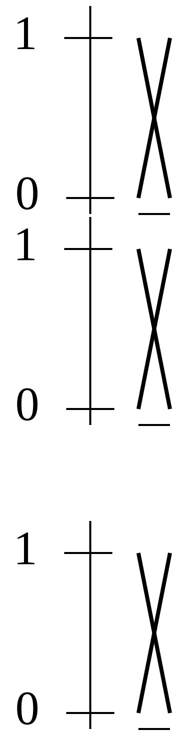
Lowpass Images

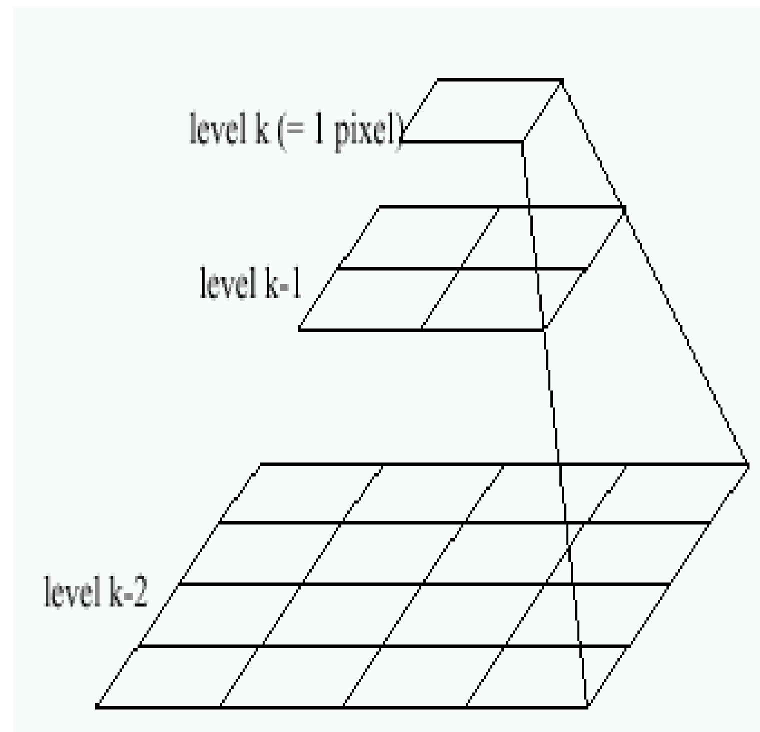


Bandpass Images

Pyramid Blending

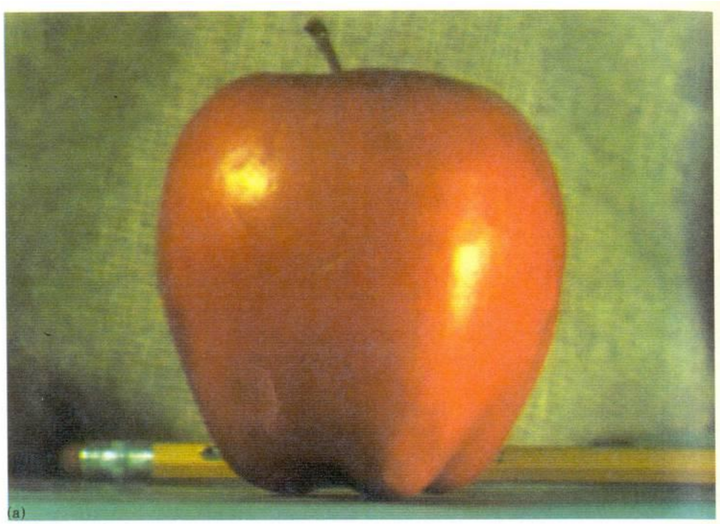


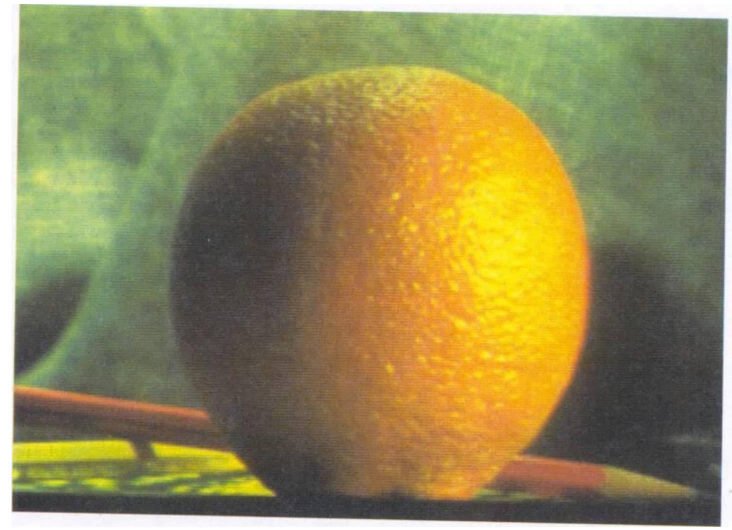


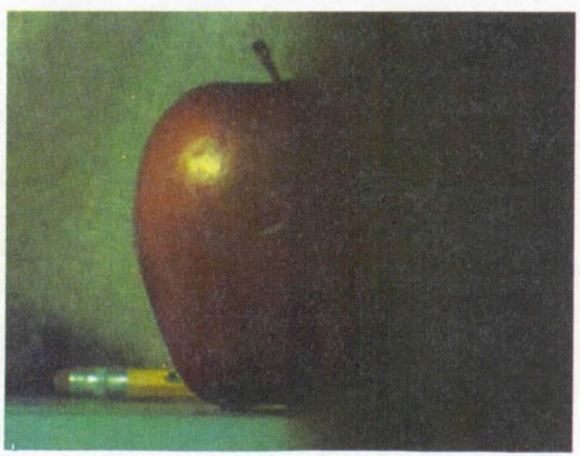


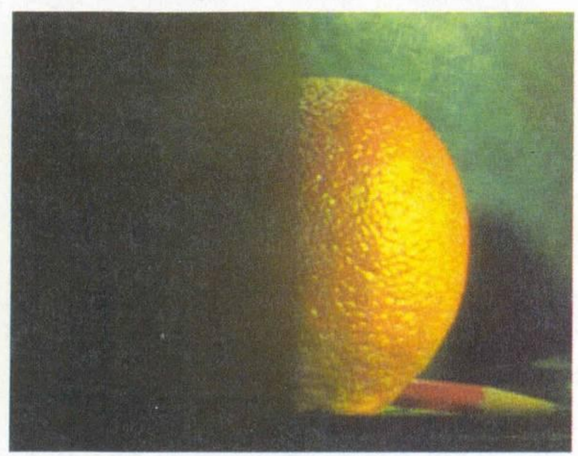
Left pyramid blend Right pyramid

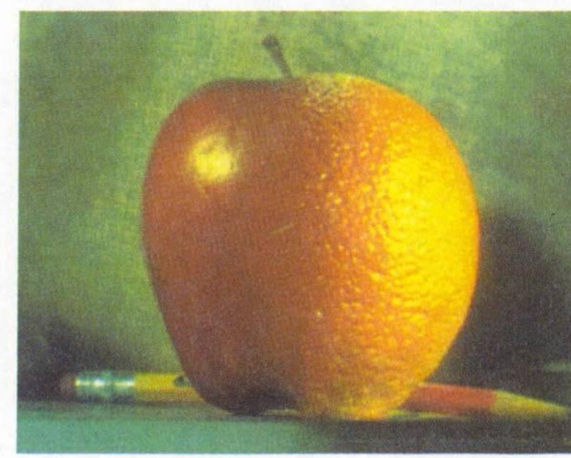
Pyramid Blending

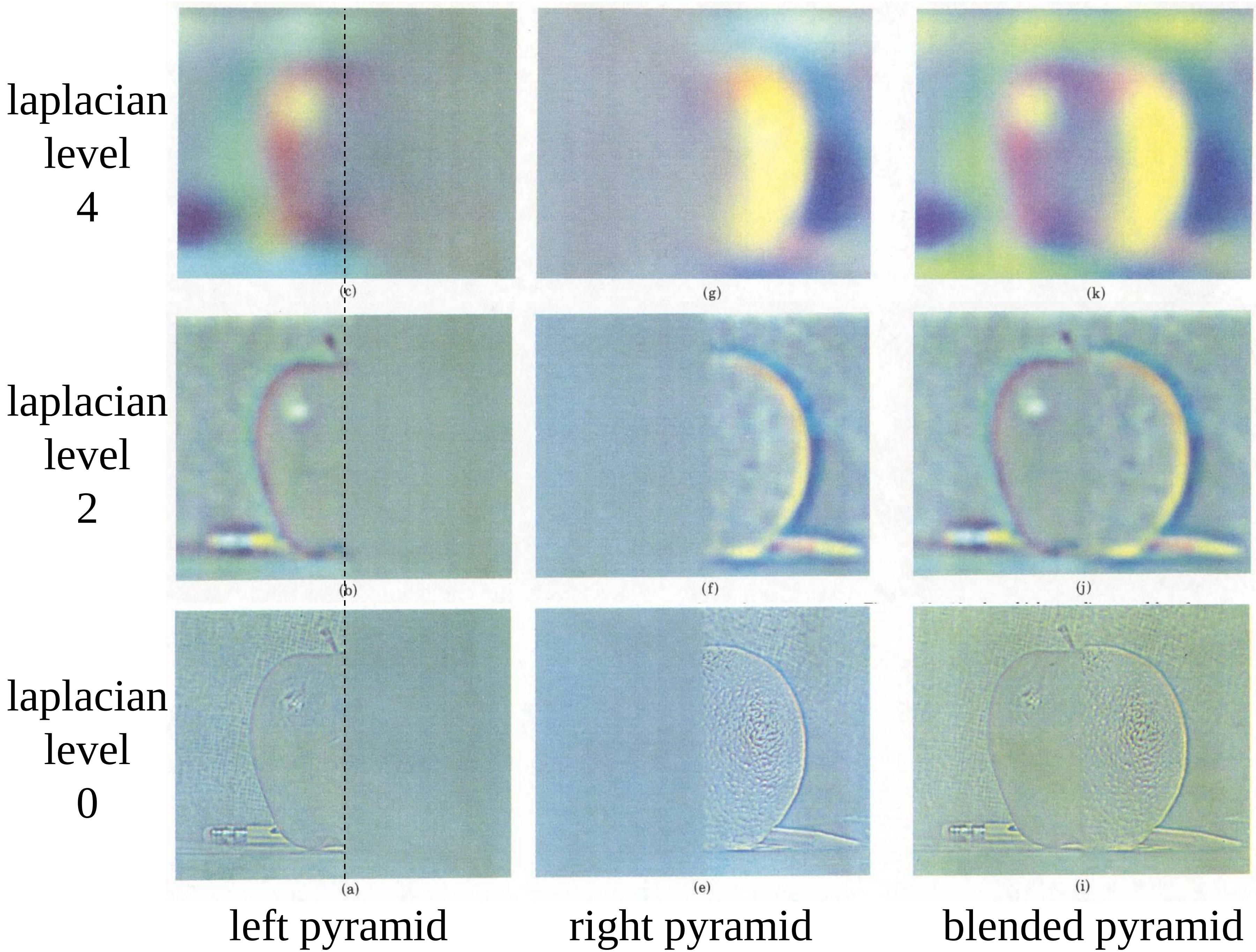

(a)




(d)


(h)


(l)

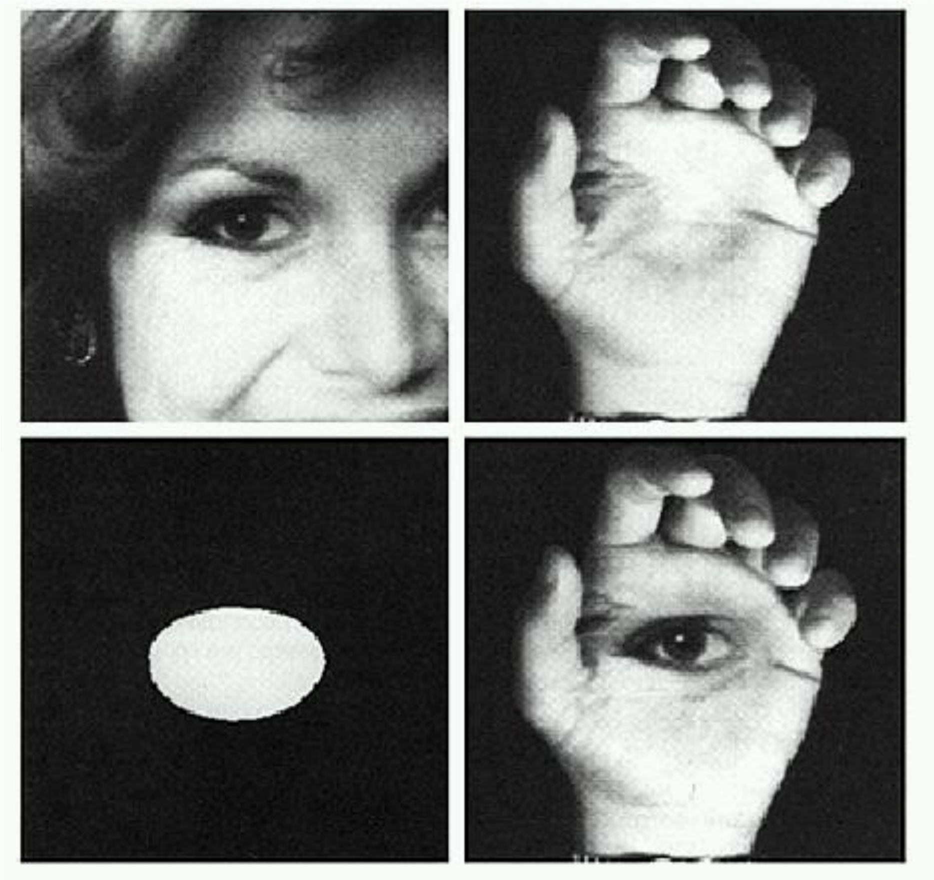

laplacian level 4
laplacian level 2
laplacian level 0
(c)
(g)
(k)
(b)
(f)
(j)
(a)
(e)
(i)
left pyramid
right pyramid
blended pyramid

Laplacian Pyramid: Blending

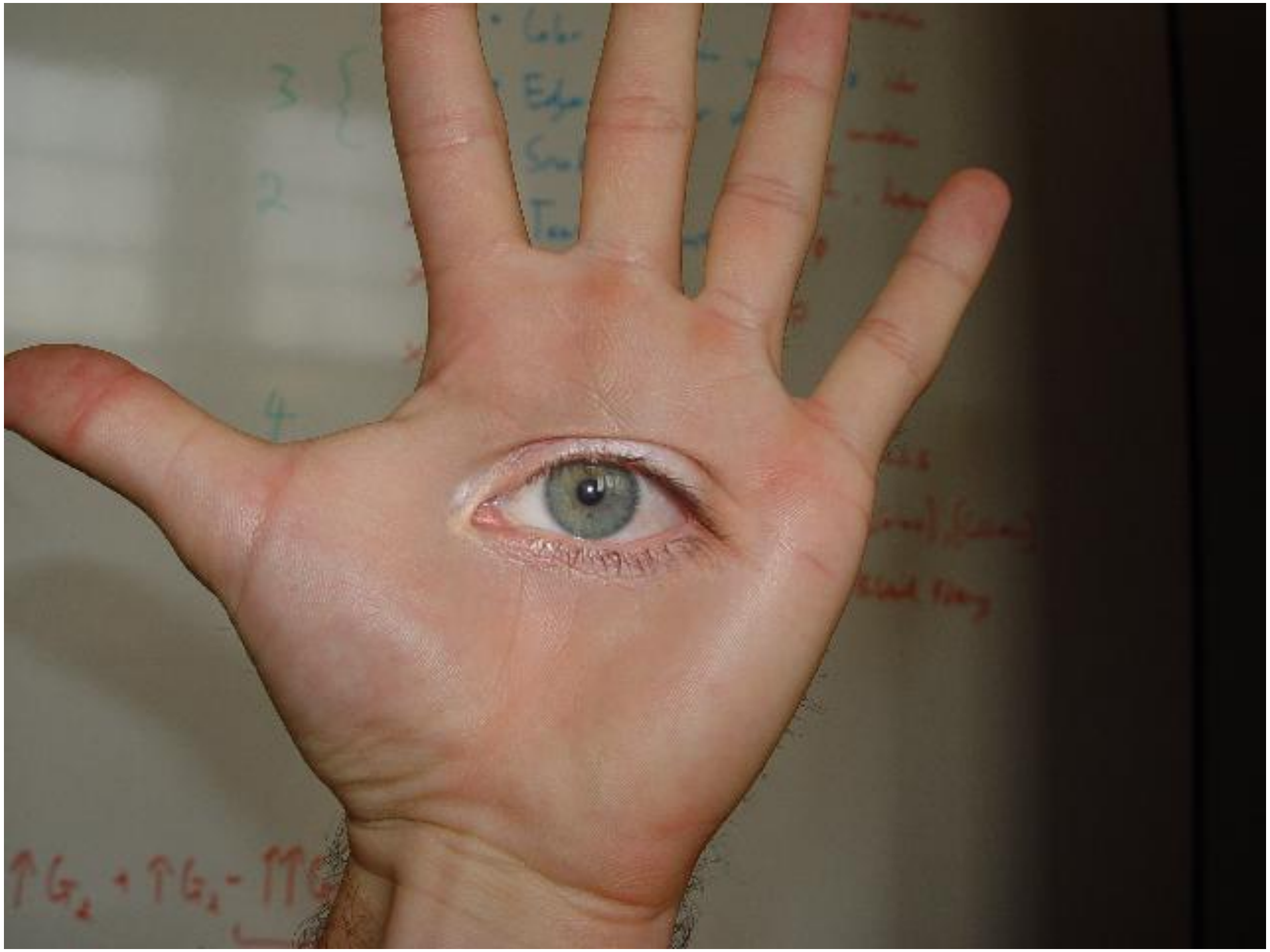
General Approach:

1. Build Laplacian pyramids *LA* and *LB* from images *A* and *B*
2. Build a Gaussian pyramid *GR* from selected region *R*
3. Form a combined pyramid *LS* from *LA* and *LB* using nodes of *GR* as weights:
   - $LS(i,j) = GR(I,j,)*LA(I,j) + (1-GR(I,j))*LB(I,j)$
4. Collapse the *LS* pyramid to get the final blended image

Blending Regions



Horror Photo



© david dmartin (Boston College)

Results from this class (fall 2005)

© Chris Cameron

Season Blending (St. Petersburg)

Season Blending (St. Petersburg)

Simplification: Two-band Blending

Brown & Lowe, 2003

- Only use two bands: high freq. and low freq.
- Blends low freq. smoothly
- Blend high freq. with no smoothing: use binary alpha

2-band Blending

Low frequency (λ > 2 pixels)

High frequency (λ < 2 pixels)

Linear Blending

2-band Blending

Don’t blend, CUT!

Moving objects become ghosts

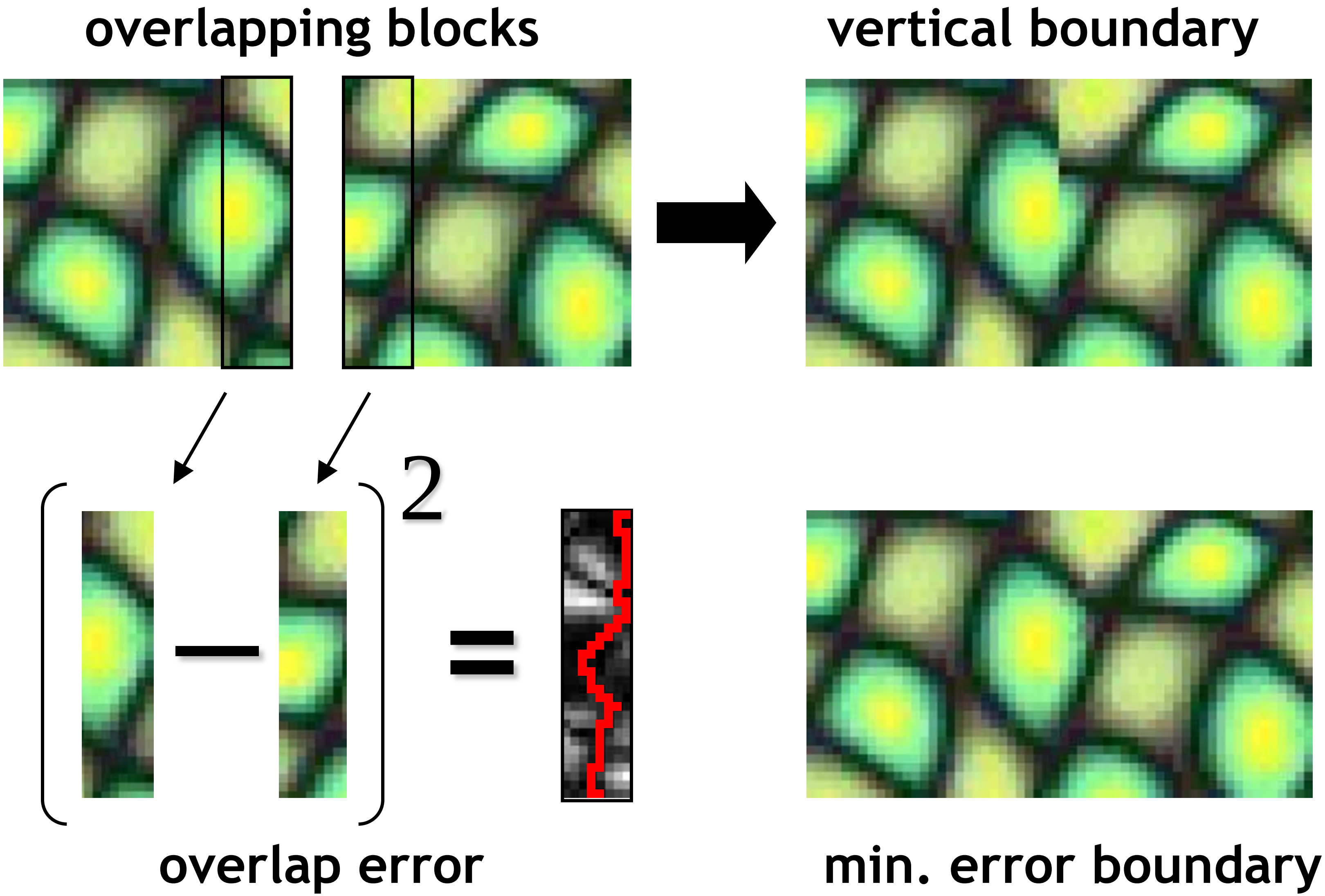
So far we only tried to blend between two images.
What about finding an optimal seam?

Davis, 1998

Segment the mosaic

- Single source image per segment
- Avoid artifacts along boundries
  - Dijkstra’s algorithm

Minimal error boundary



Seam Carving

Seam Carving for Content-Aware Image Resizing

Shai Avidan
Mitsubishi Electric Research Labs

Ariel Shamir
The Interdisciplinary Center & MERL

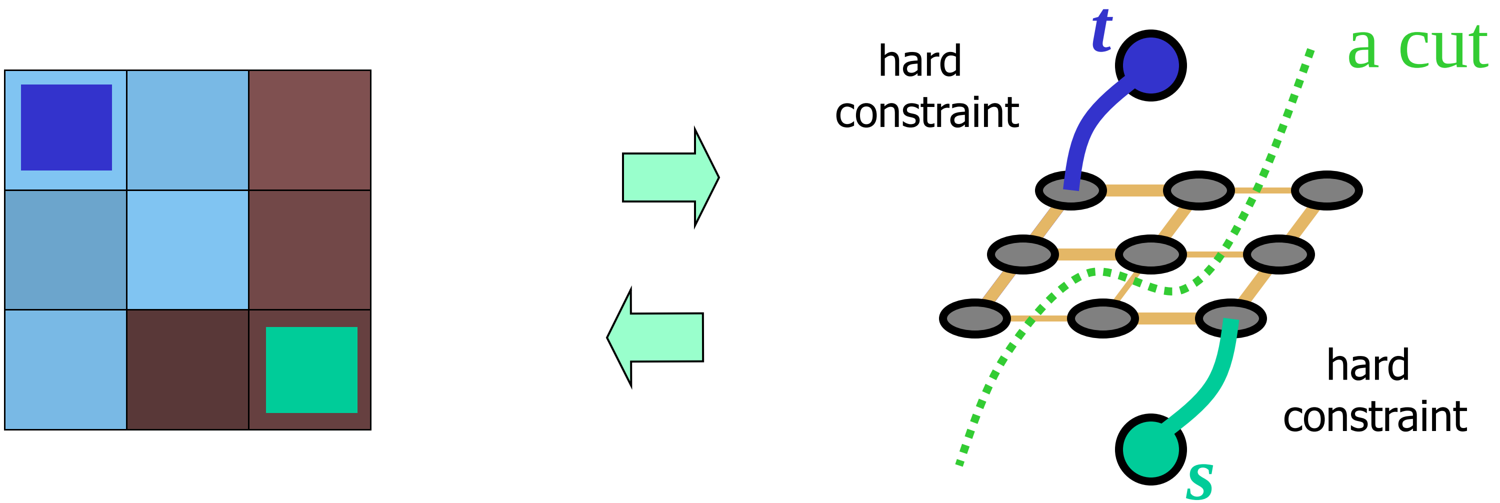
http://www.youtube.com/watch?v=6NcIJXTlugc

Graphcuts

What if we want similar "cut-where-things-agree" idea, but for closed regions?

- Dynamic programming can't handle loops

Graph cuts – a more general solution



Minimum cost cut can be computed in polynomial time

(max-flow/min-cut algorithms)

Kwatra et al, 2003

Actually, for this example, DP will work just as well…

Lazy Snapping

(a) Girl (4/2/12) (b) Ballet (4/7/14) (c) Boy (6/2/13)

(c) Grandpa (4/2/11) (d) Twins (4/4/12)

Interactive segmentation using graphcuts

Gradient Domain

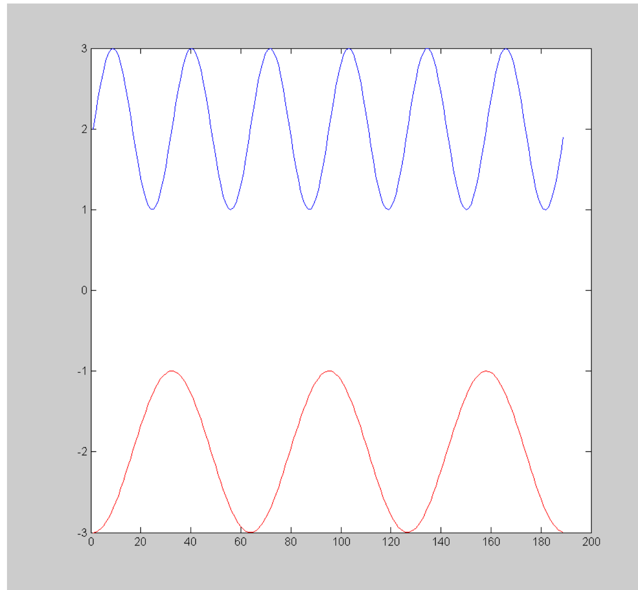
In Pyramid Blending, we decomposed our image into 2nd derivatives (Laplacian) and a low-res image

Let us now look at 1st derivatives (gradients):

- No need for low-res image
  - captures everything (up to a constant)
- Idea:
  - Differentiate
  - Blend / edit / whatever
  - Reintegrate

Gradient Domain blending (1D)

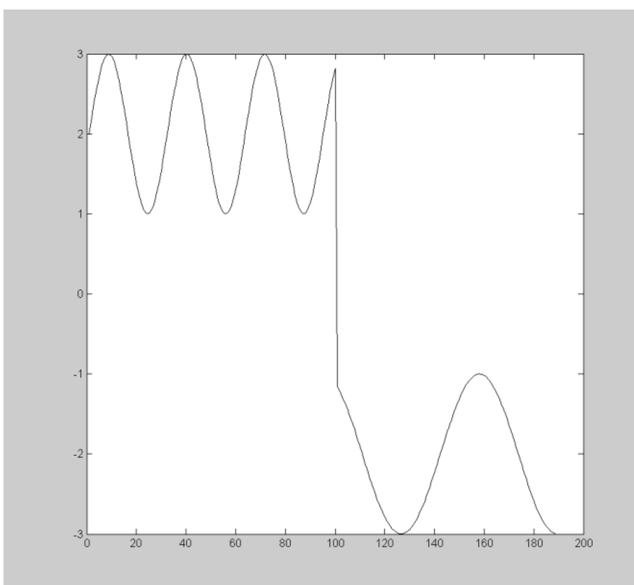
Two signals

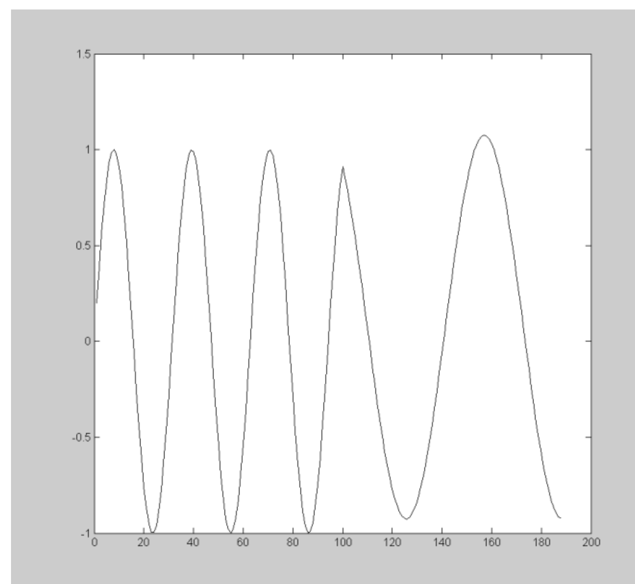


bright

dark

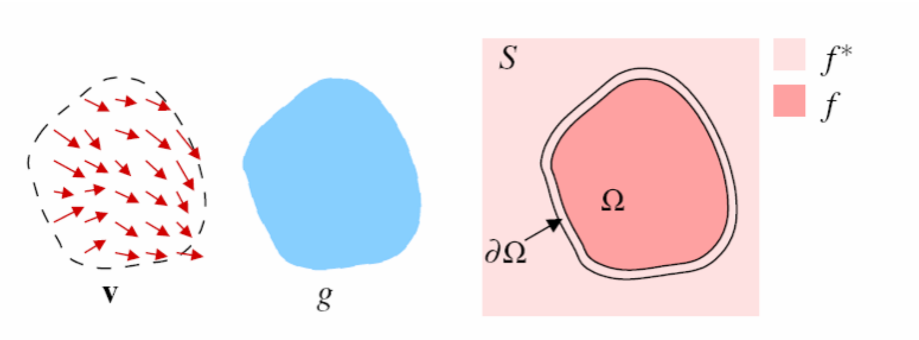
Regular blending





Blending derivatives

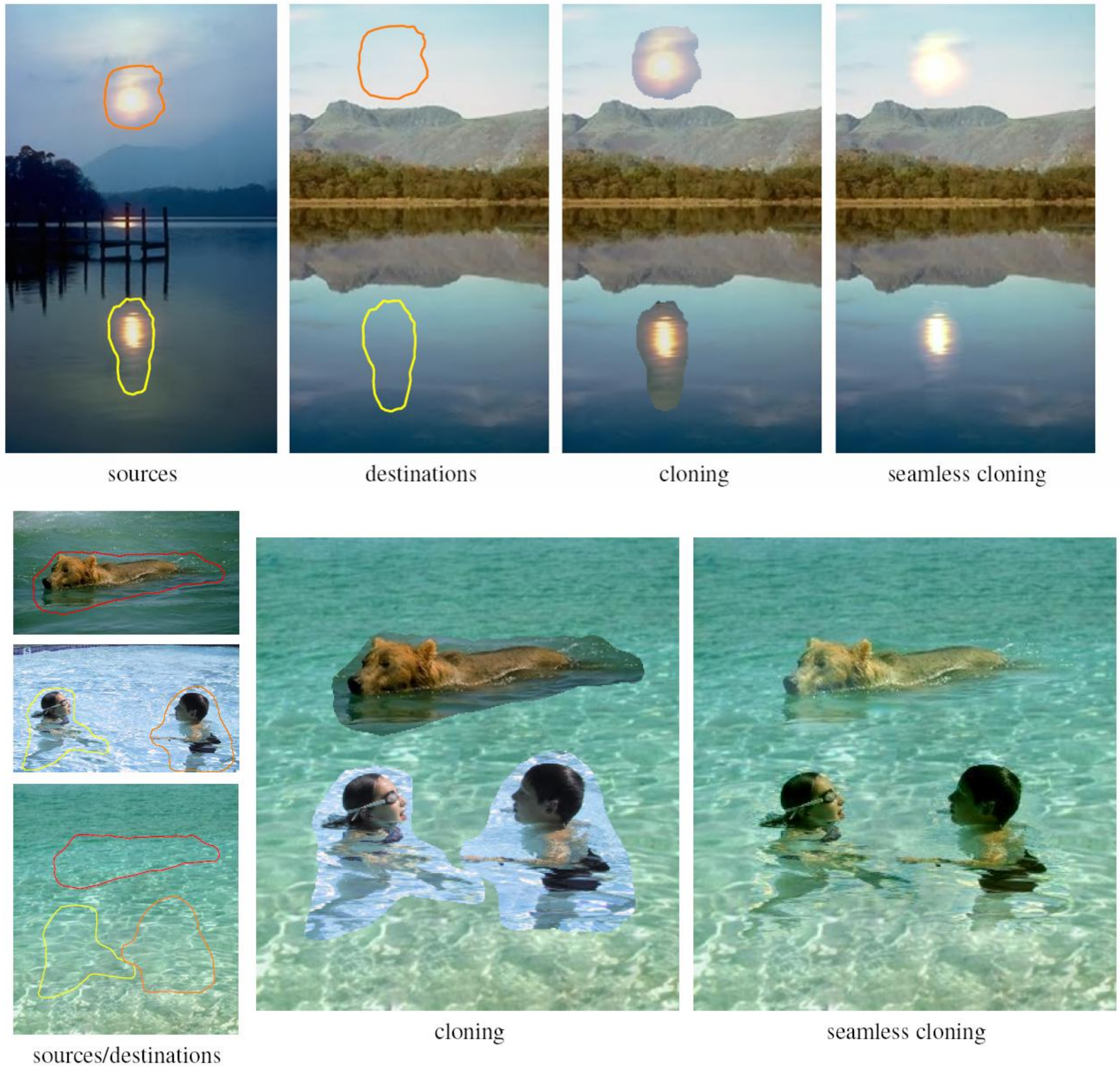
Gradient Domain Blending (2D)



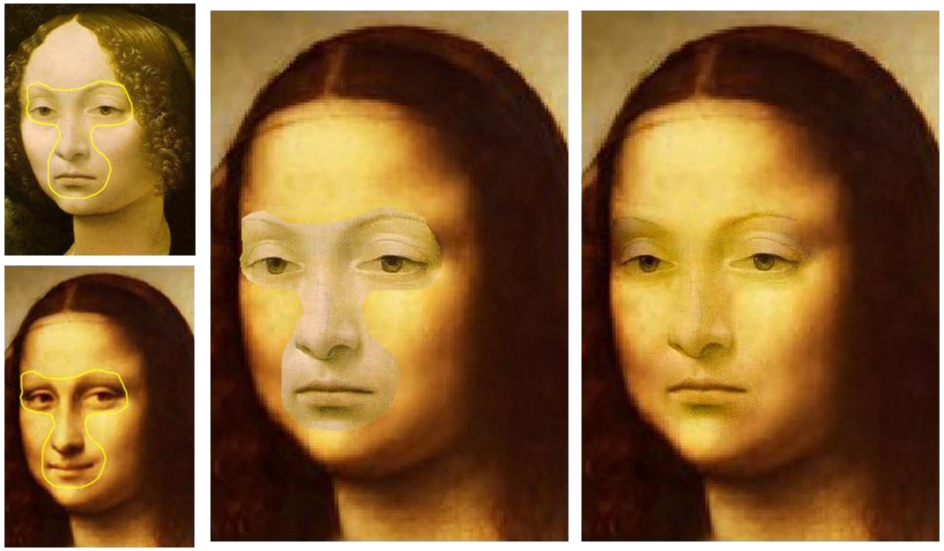
Trickier in 2D:

- Take partial derivatives dx and dy (the gradient field)
- Fidle around with them (smooth, blend, feather, etc)
- Reintegrate
  - But now integral(dx) might not equal integral(dy)
- Find the most agreeable solution
  - Equivalent to solving Poisson equation
  - Can be done using least-squares

Perez et al., 2003



Perez et al, 2003



source/destination cloning seamless cloning

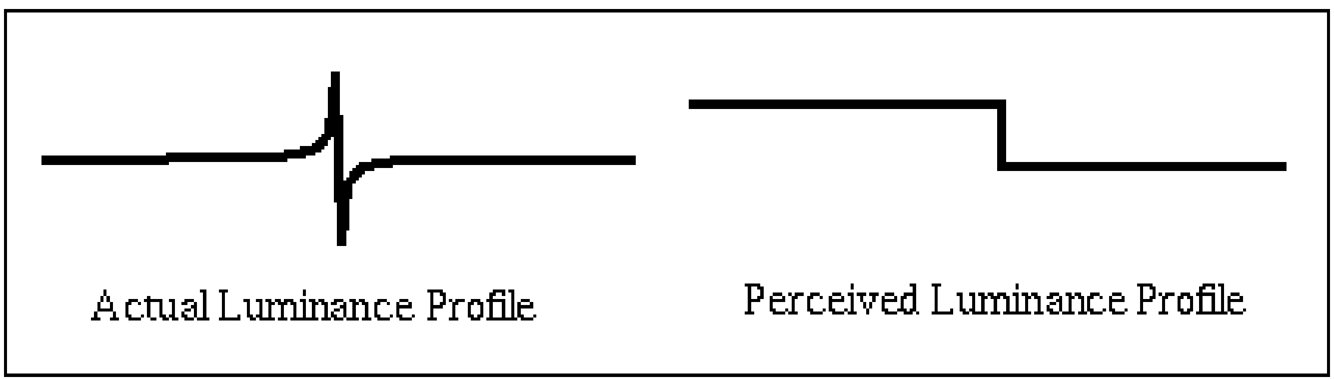
editing

Limitations:

- Can’t do contrast reversal (gray on black -> gray on white)
- Colored backgrounds “bleed through”
- Images need to be very well aligned

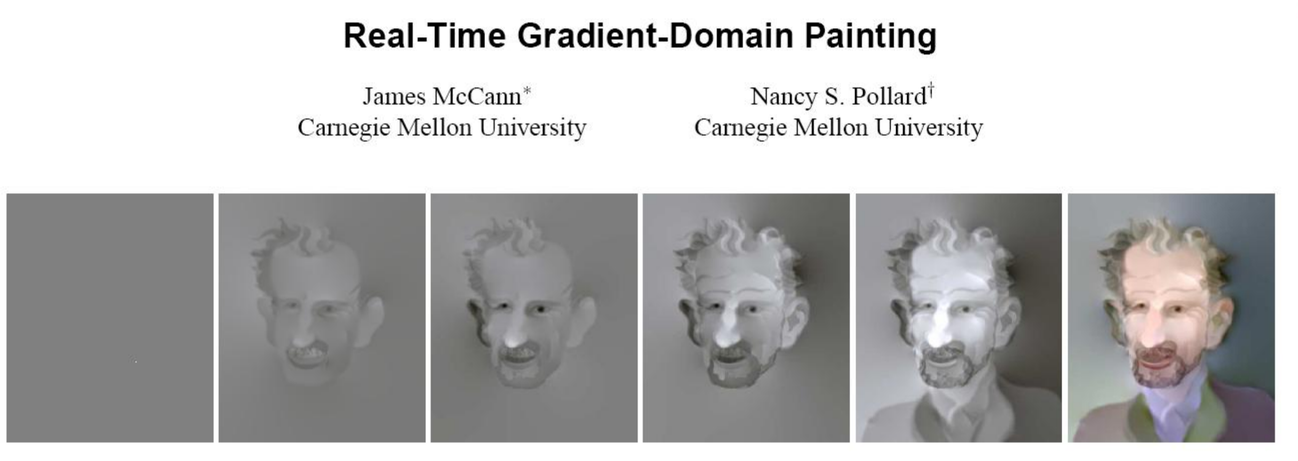
Gradients vs. Pixels

Craik-O'Brien Cornsweet Effect



Can we use this for range compression?

Thinking in Gradient Domain



Real-Time Gradient-Domain Painting

James McCann[*]
Carnegie Mellon University

Nancy S. Pollard[†]
Carnegie Mellon University

Our very own Jim McCann::

James McCann
Real-Time Gradient-Domain Painting, SIGGRAPH 2009

Gradient Domain as Image Representation

See GradientShop paper as good example:

GradientShop: A Gradient-Domain Optimization Framework for Image and Video Filtering

Pravin Bhat[1] C. Lawrence Zitnick[2] Michael Cohen[1,2] Brian Curless[1]
[1]University of Washington [2]Microsoft Research

http://www.gradientshop.com/

Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images

Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – low level image-features

Motivation for gradient-domain filtering?

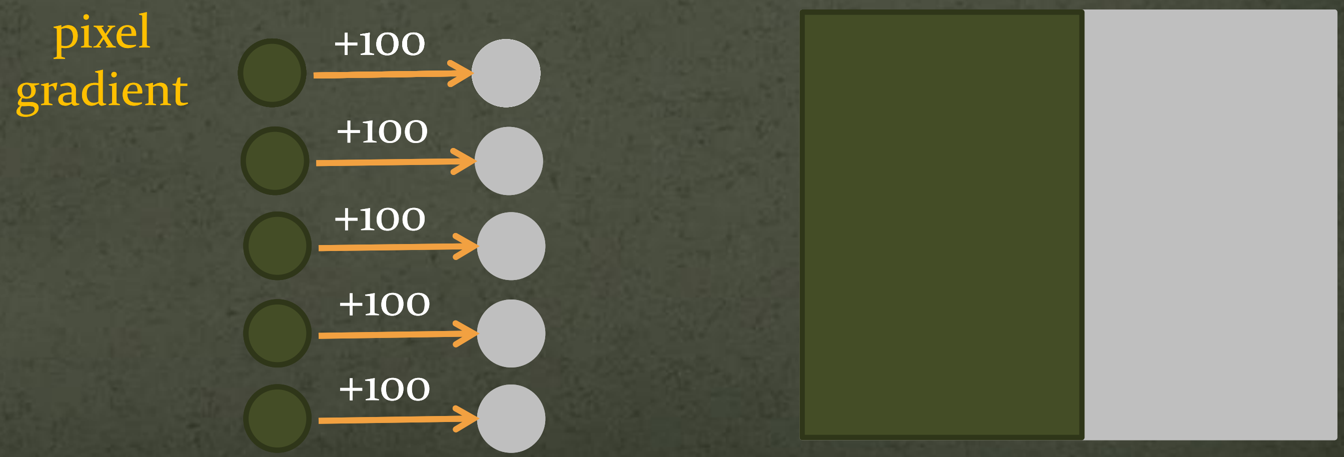
- Can be used to exert high-level control over images
  - gradients – low level image-features

Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – low level image-features
  - gradients – give rise to high level image-features

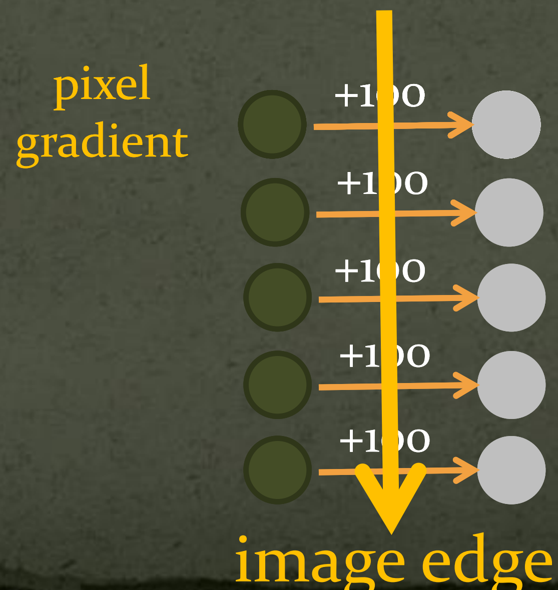
Motivation for gradient-domain filtering?

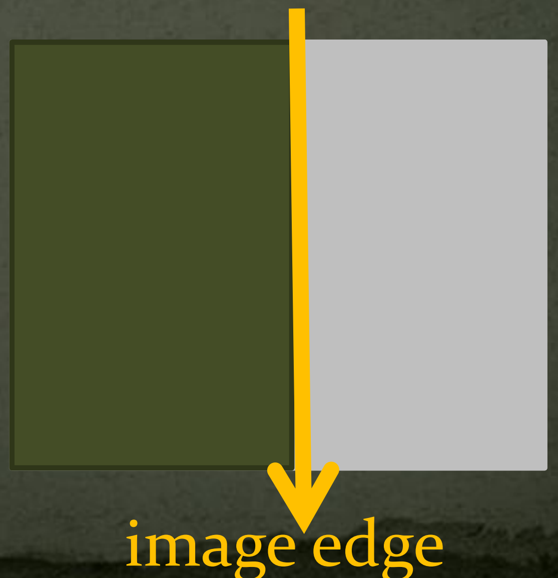
- Can be used to exert high-level control over images
  - gradients – low level image-features
  - gradients – give rise to high level image-features



Motivation for gradient-domain filtering?

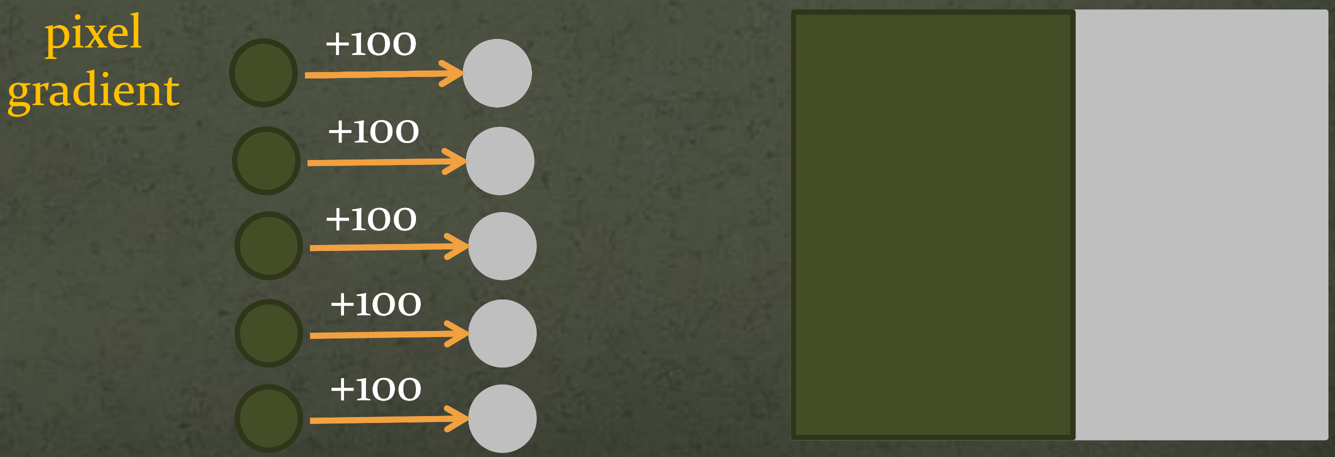
- Can be used to exert high-level control over images
  - gradients – low level image-features
  - gradients – give rise to high level image-features





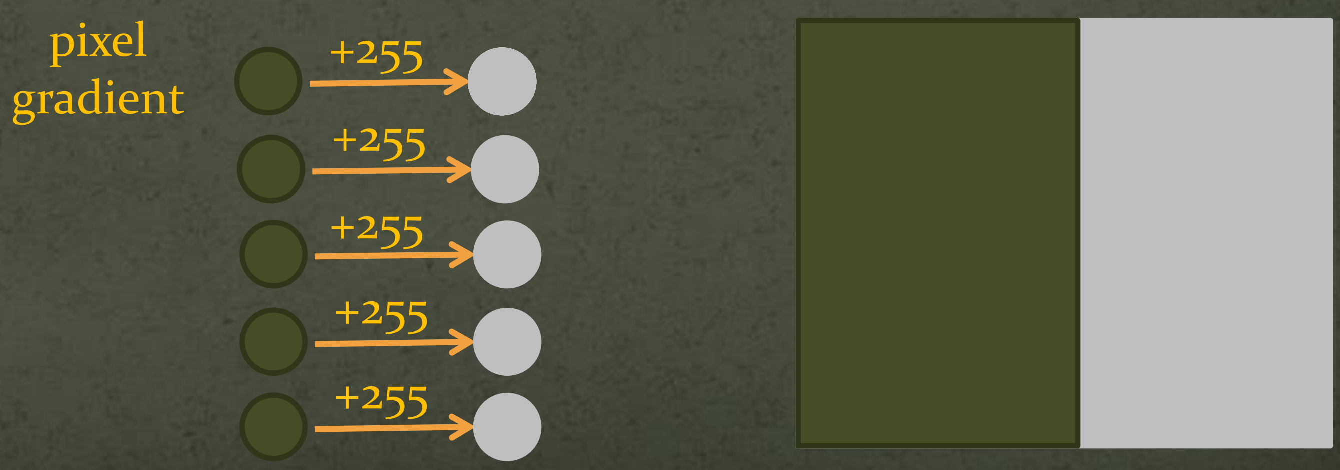
Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – low level image-features
  - gradients – give rise to high level image-features
  - manipulate local gradients to manipulate global image interpretation



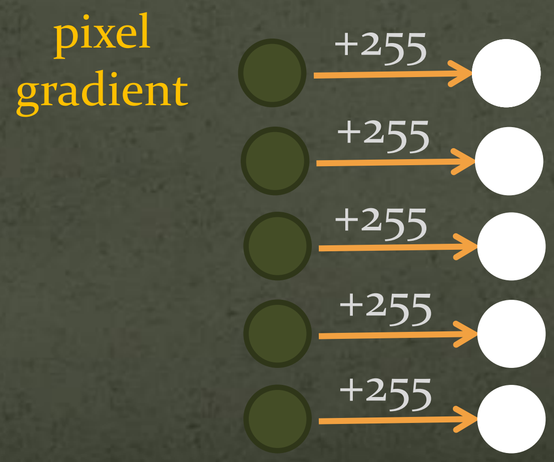
Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – low level image-features
  - gradients – give rise to high level image-features
  - manipulate local gradients to manipulate global image interpretation



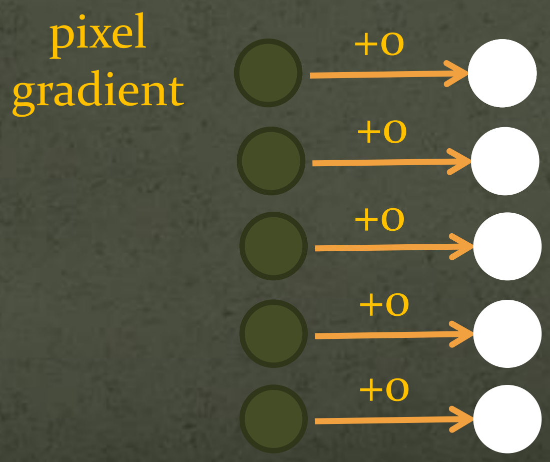
Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – low level image-features
  - gradients – give rise to high level image-features
  - manipulate local gradients to manipulate global image interpretation



Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – low level image-features
  - gradients – give rise to high level image-features
  - manipulate local gradients to manipulate global image interpretation



Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – low level image-features
  - gradients – give rise to high level image-features
  - manipulate local gradients to manipulate global image interpretation

Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – give rise to high level image-features

Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – give rise to high level image-features
    - Edges

Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – give rise to high level image-features
    - Edges
      - object boundaries
      - depth discontinuities
      - shadows
      - …

Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – give rise to high level image-features
    - Edges
    - Texture

Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – give rise to high level imag
    - Edges
    - Texture
      - visual richness
      - surface properties

Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – give rise to high level image-features
    - Edges
    - Texture
    - Shading

Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – give rise to high level image-features
    - Edges
    - Texture
    - Shading
      - lighting

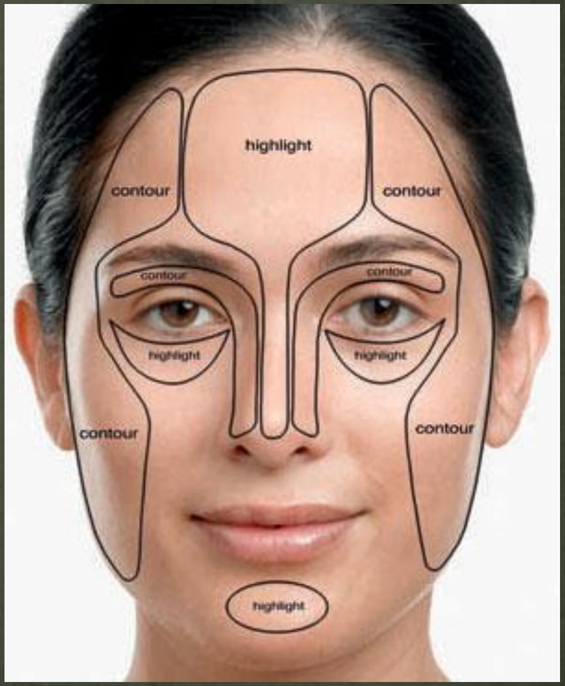
Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – give rise to high level image-features
    - Edges
    - Texture
    - Shading
      - lighting
      - shape

sculpting the face
using shading (makeup)

Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
    - gradients – give rise to high level image-features
        - Edges
        - Texture
        - Shading
            - lighting
            - shape



sculpting the face
using shading (makeup)

Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – give rise to high level image-features
    - Edges
    - Texture
    - Shading
      - lighting
      - shape

sculpting the face
using shading (makeup)

Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – give rise to high level image-features
    - Edges
    - Texture
    - Shading
      - lighting
      - shape

sculpting the face
using shading (makeup)

Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – give rise to high level image-features
    - Edges
    - Texture
    - Shading

Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – give rise to high level image-features
    - Edges
    - Texture
    - Shading
    - Artifacts

Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – give rise to high level image-features
    - Edges
    - Texture
    - Shading
    - Artifacts
      - noise

sensor noise

Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – give rise to high level image-features
    - Edges
    - Texture
    - Shading
    - Artifacts
      - noise
      - seams

seams in composite images

Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – give rise to high level image-features
    - Edges
    - Texture
    - Shading
    - Artifacts
      - noise
      - seams
      - compression artifacts

blocking in compressed images

Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – give rise to high level image-features
    - Edges
    - Texture
    - Shading
    - Artifacts
      - noise
      - seams
      - compression artifacts

ringing in compressed images

Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images
  - gradients – give rise to high level image-features
    - Edges
    - Texture
    - Shading
    - Artifacts
      - noise
      - seams
      - compression artifacts
      - flicker

flicker from exposure changes & film degradation

Motivation for gradient-domain filtering?

- Can be used to exert high-level control over images

GradientShop

- Optimization framework

Pravin Bhat et al

GradientShop

- Optimization framework
  - Input unfiltered image – ***u***

GradientShop

- Optimization framework
  - Input unfiltered image – ***u***
  - Output filtered image – ***f***

GradientShop

- Optimization framework
  - Input unfiltered image – ***u***
  - Output filtered image – ***f***
  - Specify desired pixel-differences – (g^x, g^y)

Energy function

$$\boxed{\min_{f} \quad (f_x - g^x)^2 + \quad (f_y - g^y)^2}$$

GradientShop

- Optimization framework
  - Input unfiltered image – ***u***
  - Output filtered image – ***f***
  - Specify desired pixel-differences – $(\boldsymbol{g}^x, \boldsymbol{g}^y)$
  - Specify desired pixel-values – ***d***

Energy function

$$\boxed{\min_{f} \quad (\boldsymbol{f}_x - \boldsymbol{g}^x)^2 + \quad (\boldsymbol{f}_y - \boldsymbol{g}^y)^2 + \quad (\boldsymbol{f} - \boldsymbol{d})^2}$$

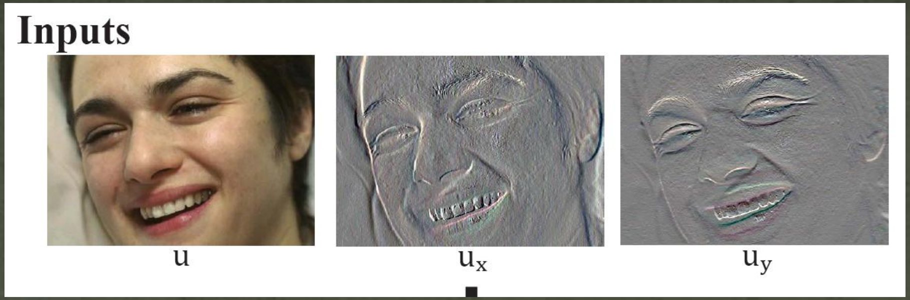
GradientShop

- Optimization framework
  - Input unfiltered image – ***u***
  - Output filtered image – ***f***
  - Specify desired pixel-differences – $(\boldsymbol{g}^x, \boldsymbol{g}^y)$
  - Specify desired pixel-values – ***d***
  - Specify constraints weights – $(\boldsymbol{w}^x, \boldsymbol{w}^y, \boldsymbol{w}^d)$

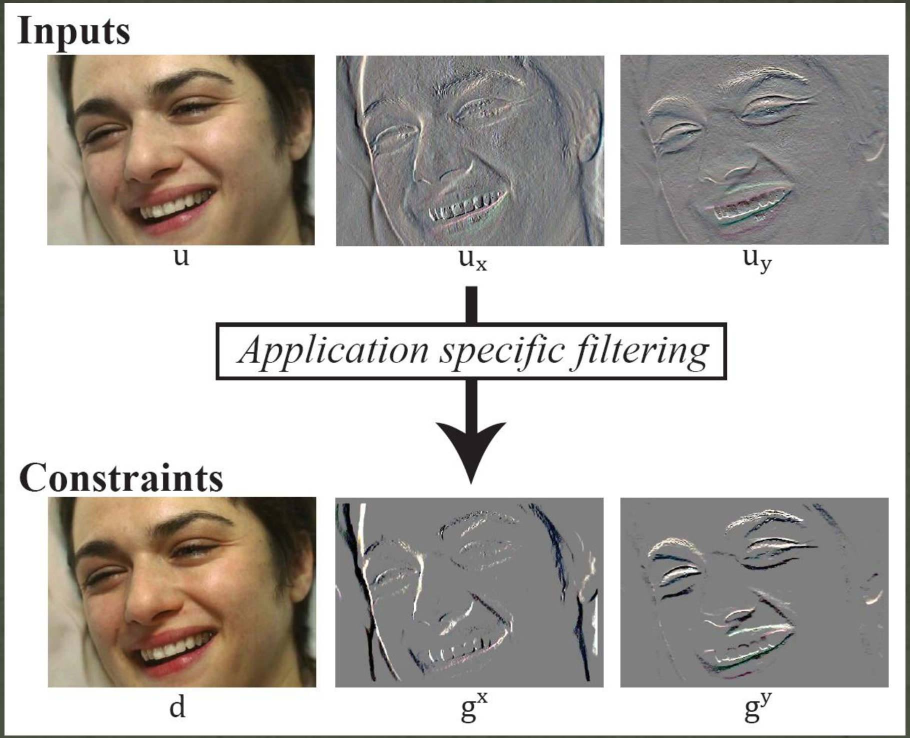
Energy function

$$\boxed{\min_{f} \quad \boldsymbol{w}^x(\boldsymbol{f}_x - \boldsymbol{g}^x)^2 + \boldsymbol{w}^y(\boldsymbol{f}_y - \boldsymbol{g}^y)^2 + \boldsymbol{w}^d(\boldsymbol{f} - \boldsymbol{d})^2}$$

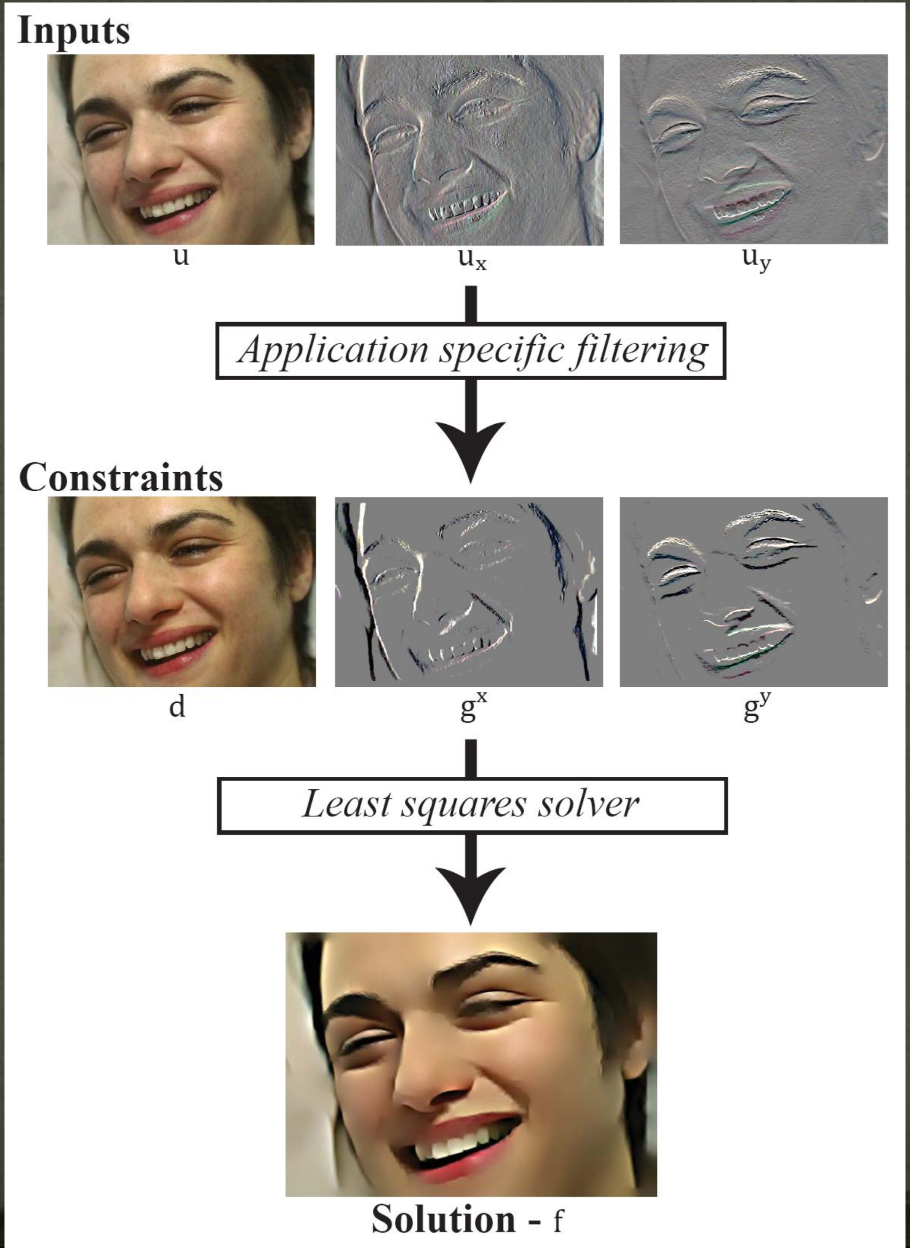
GradientShop



GradientShop



GradientShop



Least Squares Example

Say we have a set of data points (X1,X1'), (X2,X2'), (X3,X3'), etc. (e.g. person's height vs. weight)

We want a nice compact formula (a line) to predict X's from Xs: Xa + b = X'

We want to find a and b

How many (X,X') pairs do we need?

$$X_1 a + b = X_1'$$
$$X_2 a + b = X_2'$$

$$\begin{bmatrix} X_1 & 1 \\ X_2 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} X_1' \\ X_2' \end{bmatrix}$$

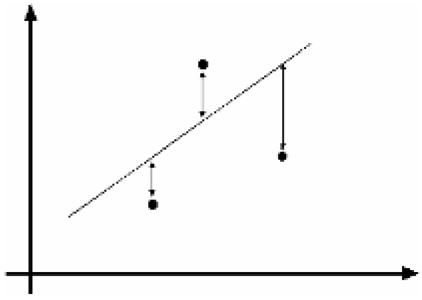
Ax=B

What if the data is noisy?

$$\begin{bmatrix} X_1 & 1 \\ X_2 & 1 \\ X_3 & 1 \\ \dots & \dots \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} X_1' \\ X_2' \\ X_3' \\ \dots \end{bmatrix}$$

overconstrained

$$\min \|Ax - B\|^2$$



Putting it all together

Compositing images

- Have a clever blending function
  - Feathering
  - Center-weighted
  - blend different frequencies differently
  - Gradient based blending
- Choose the right pixels from each image
  - Dynamic programming – optimal seams
  - Graph-cuts

Now, let's put it all together:

- Interactive Digital Photomontage, 2004 (video)

Interactive Digital Photomontage

Aseem Agarwala, Mira Dontcheva
Maneesh Agrawala, Steven Drucker, Alex Colburn
Brian Curless, David Salesin, Michael Cohen