CS4670 / 5670: Computer Vision

Kavita Bala

Lecture 21: Panoramas

Announcements

- **Prelim on Thu**
  - Everything till Lecture 17
  - Closed book
  - Bring your calculator
  - **7:30 pm, Location**
    - **Kennedy Hall, 116**

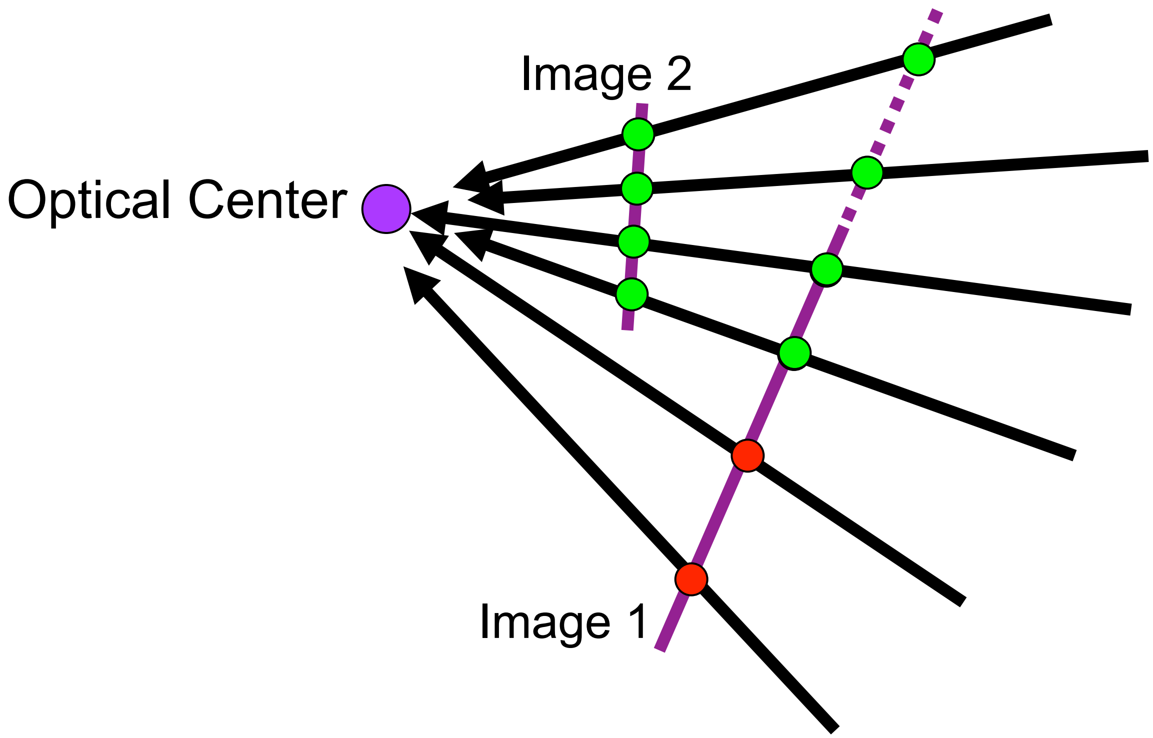
Mosaics

- How do we align the images?

Creating a panorama

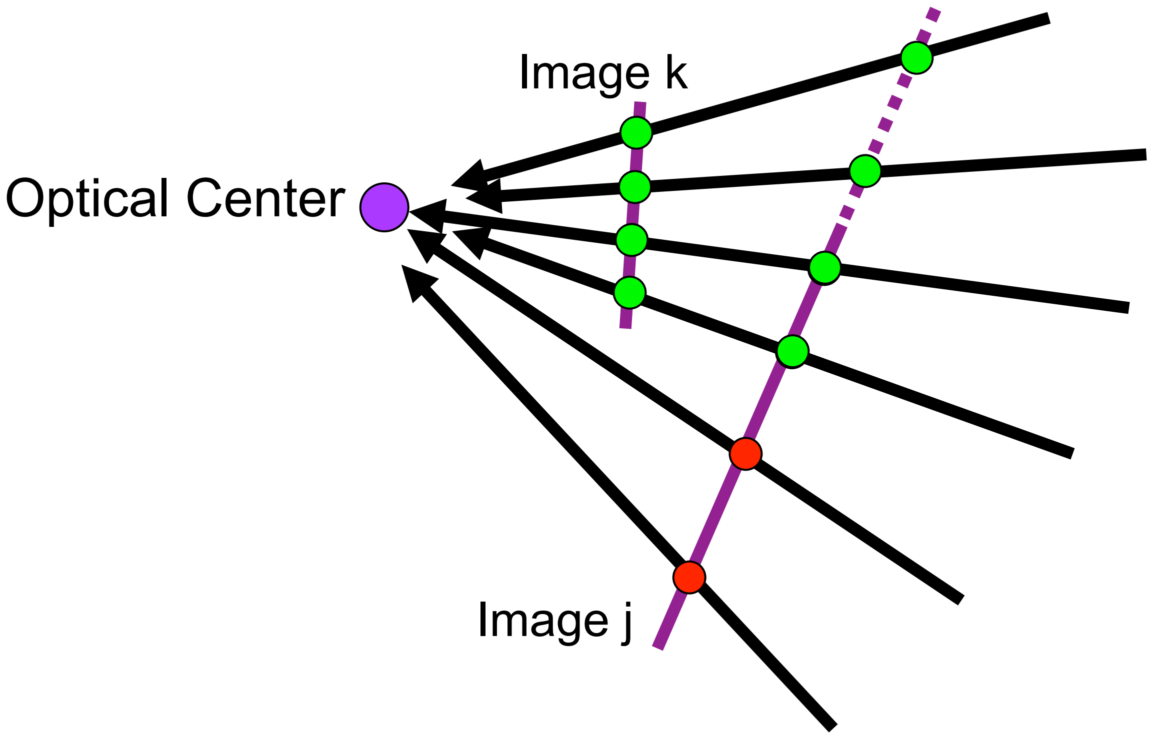
- Basic Procedure
  - Take a sequence of images from the same position
    - Rotate the camera about its optical center
  - Compute transformation between second image and first
  - Transform the second image to overlap with the first
  - Blend the two together to create a mosaic
  - If there are more images, repeat

Geometric Interpretation of Mosaics



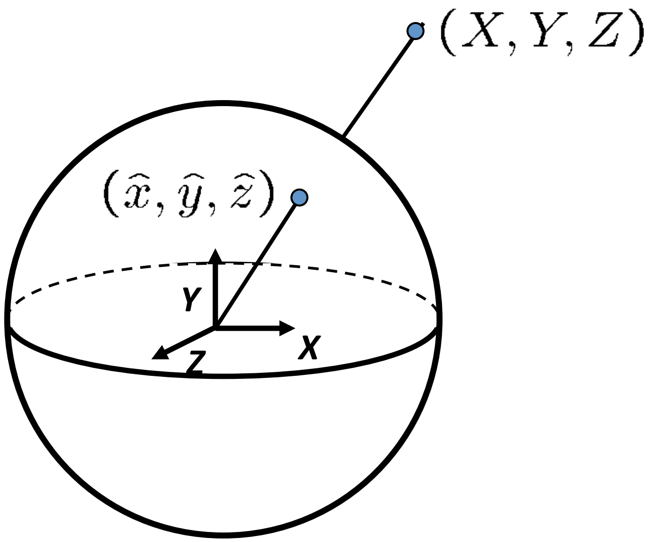
- If we capture all 360º of rays, we can create a 360º panorama
- The basic operation is *projecting* an image from one plane to another
- The projective transformation is scene-INDEPENDENT
  - This depends on all the images having the same optical center

What is the transformation?



$$\tilde{\boldsymbol{x}}_{ik} \sim \tilde{\boldsymbol{H}}_{kj} \tilde{\boldsymbol{x}}_{ij} = \boldsymbol{K}_k \boldsymbol{R}_k \boldsymbol{R}_j^{-1} \boldsymbol{K}_j^{-1} \tilde{\boldsymbol{x}}_{ij}.$$

Spherical projection



unit sphere

- Map 3D point (X,Y,Z) onto sphere

$$(\hat{x}, \hat{y}, \hat{z}) = \frac{1}{\sqrt{X^2+Y^2+Z^2}}(X, Y, Z)$$

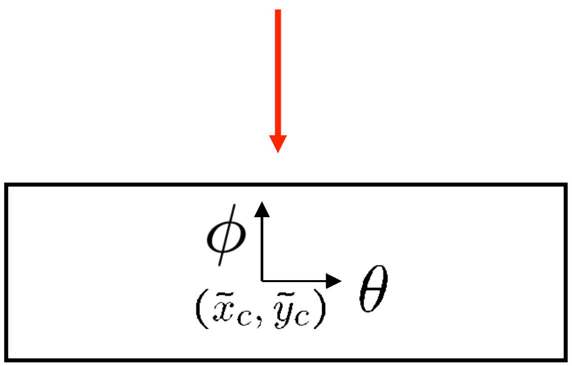
- Convert to spherical coordinates

$$(sin\theta cos\phi, sin\phi, cos\theta cos\phi) = (\hat{x}, \hat{y}, \hat{z})$$

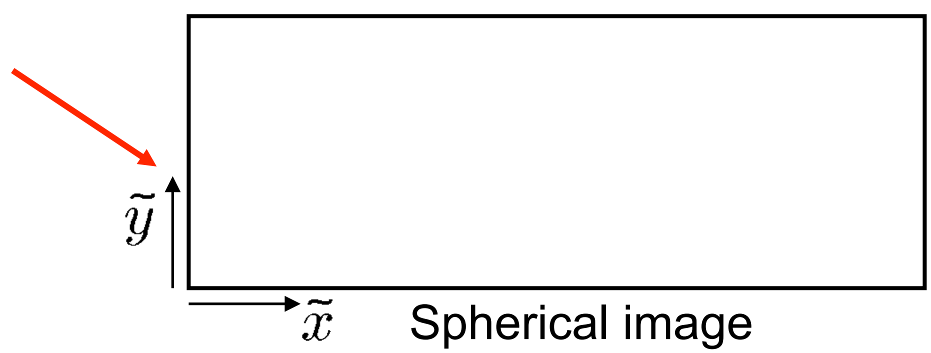
- Convert to spherical image coordinates

$$(\tilde{x}, \tilde{y}) = (s\theta, s\phi) + (\tilde{x}_c, \tilde{y}_c)$$

  - s defines size of the final image
    - » often convenient to set s = camera focal length in pixels



unwrapped sphere



Spherical image

Spherical reprojection

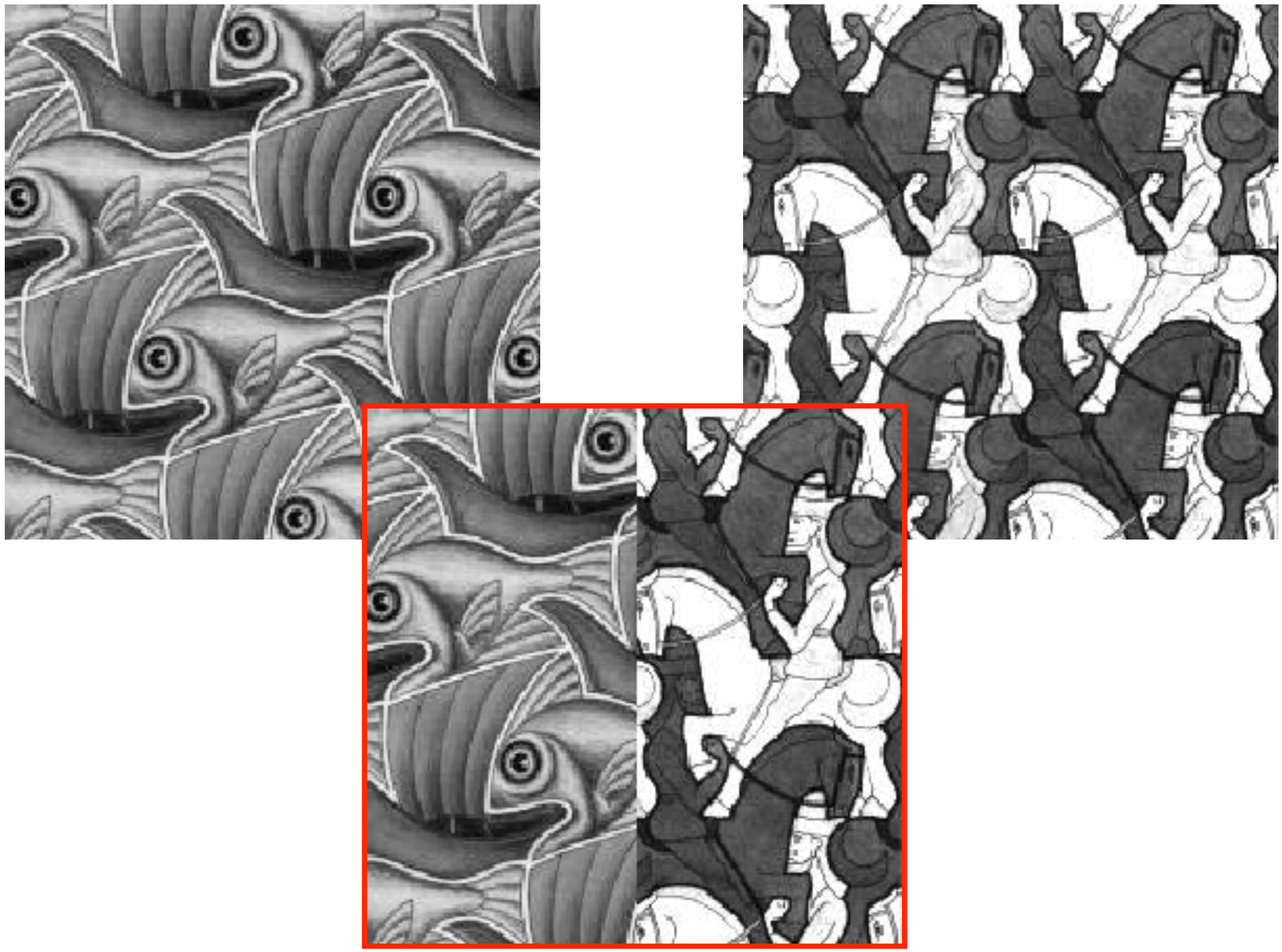
input **f = 200 (pixels)** **f = 400** **f = 800**

- Map image to spherical coordinates
  - need to know the focal length

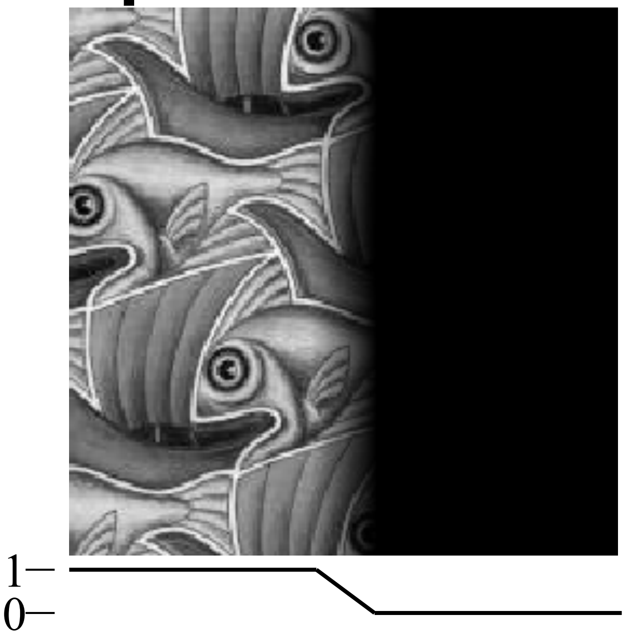
Aligning spherical images

- Suppose we rotate the camera by θ about the vertical axis
  - How does this change the spherical image?
    - Translation by θ
  - This means that we can align spherical images by translation

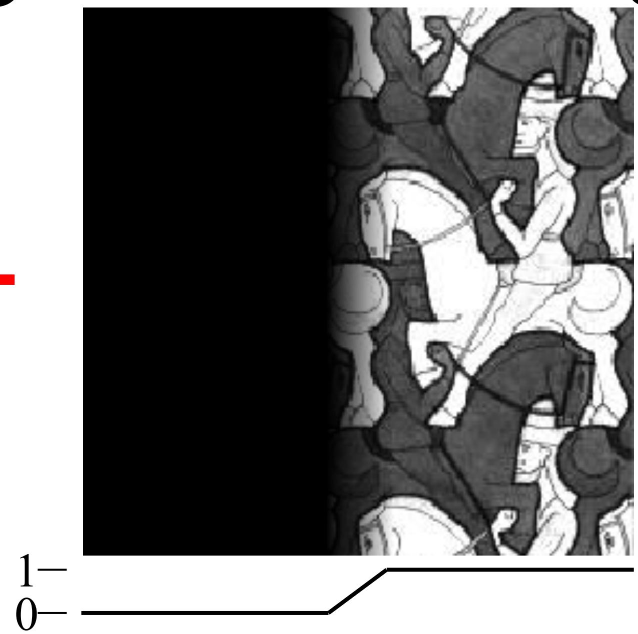
Need blending



Alpha Blending / Feathering



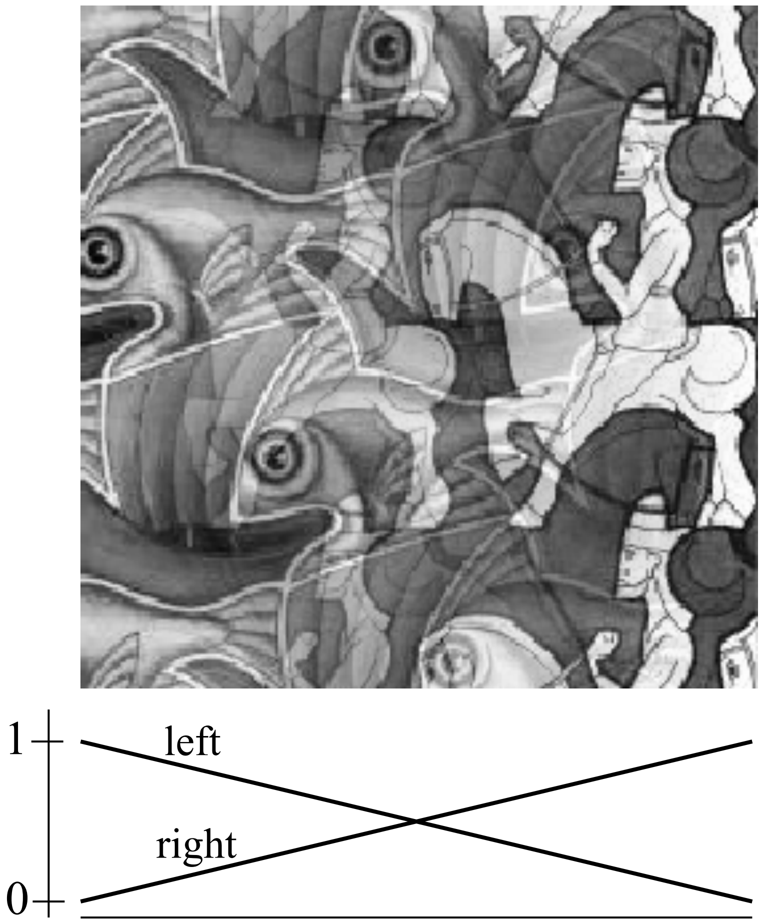
+

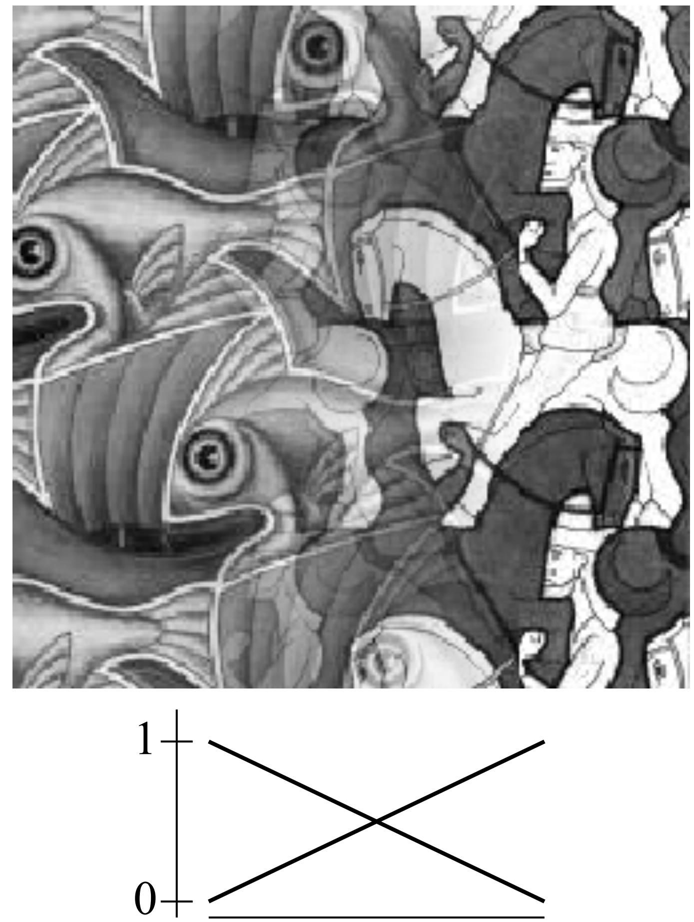


=

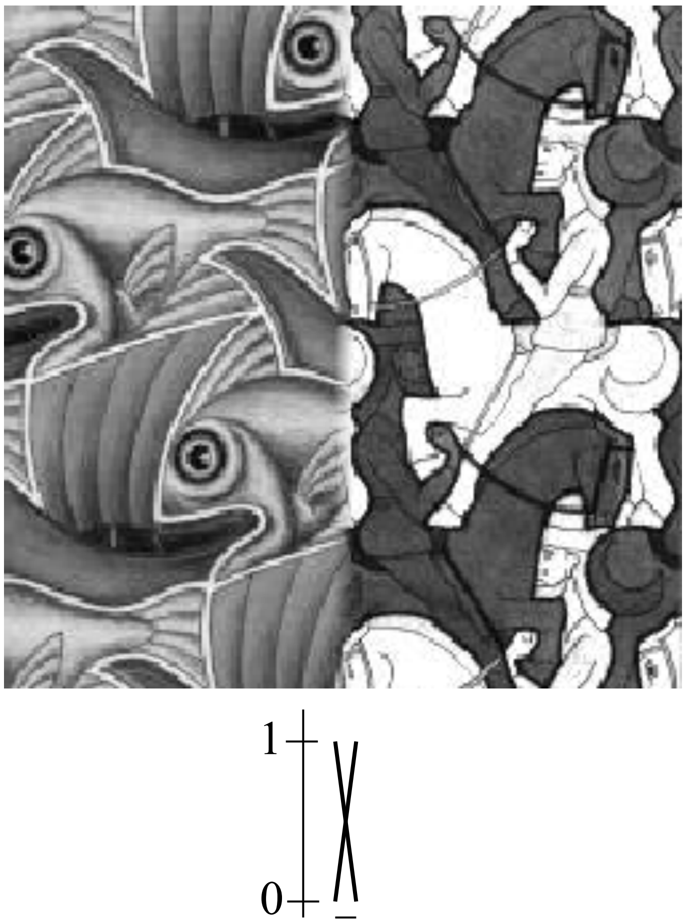
$$I_{blend} = \alpha I_{left} + (1-\alpha) I_{right}$$

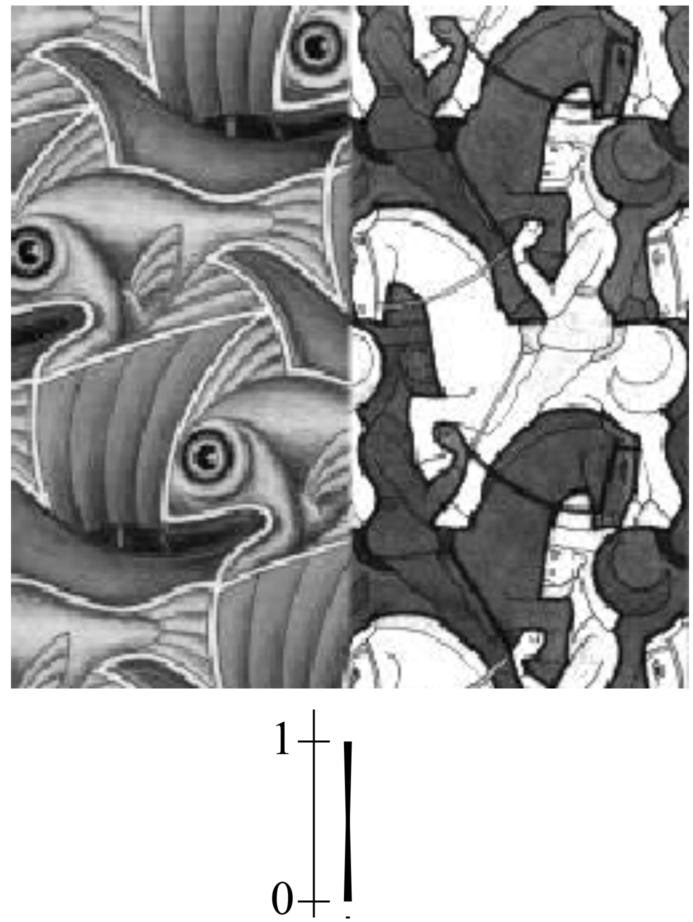
Effect of Window Size



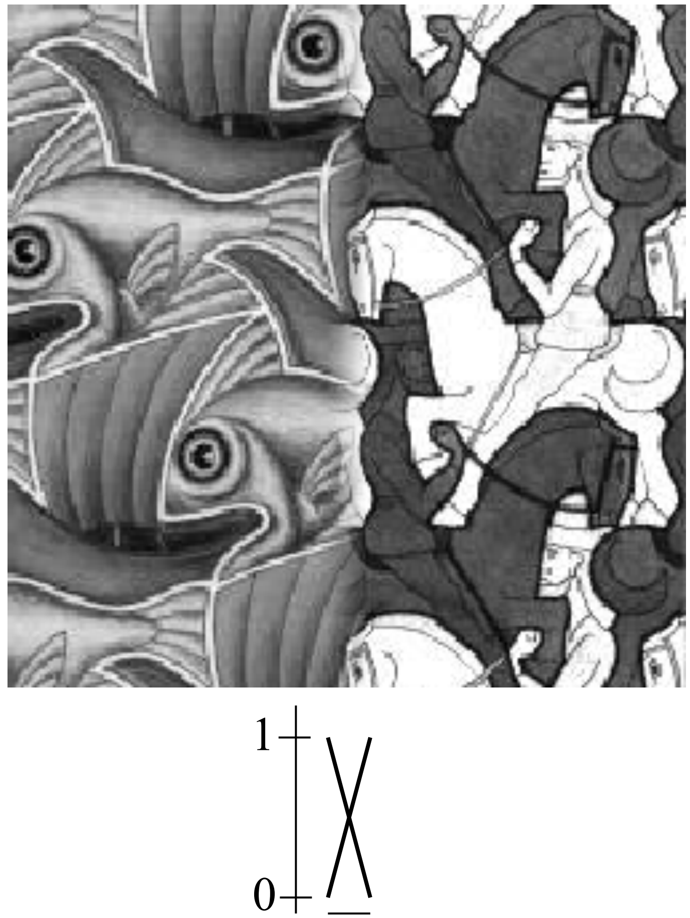


Effect of Window Size





Good Window Size



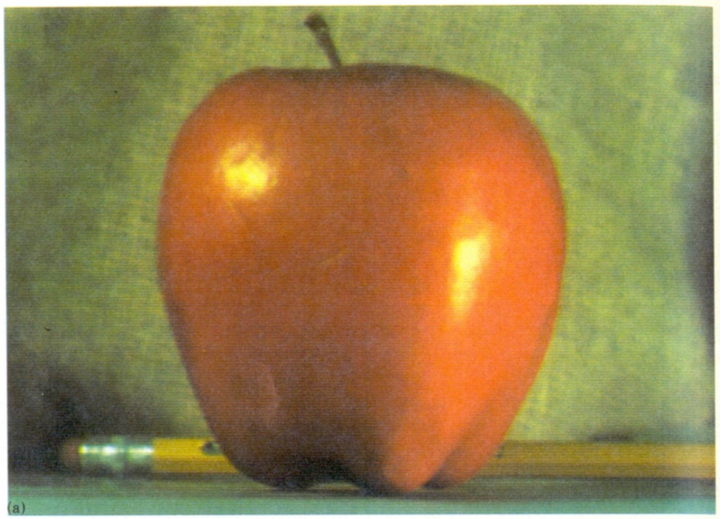
“Optimal” Window: smooth but not ghosted

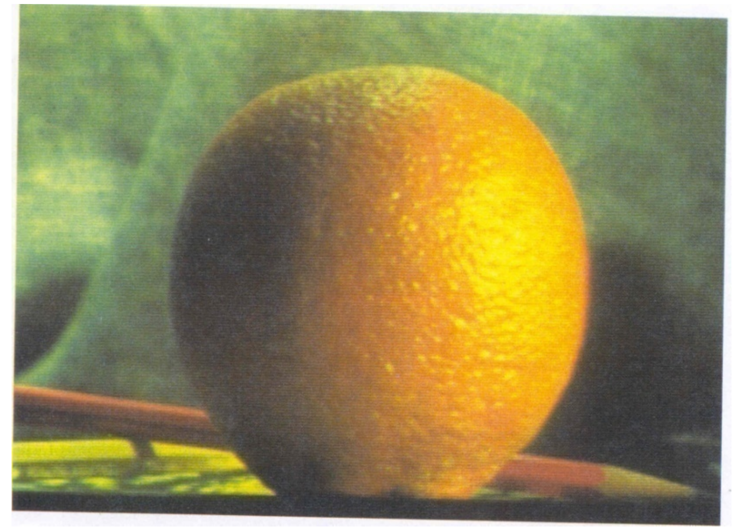
What is the optimal size?

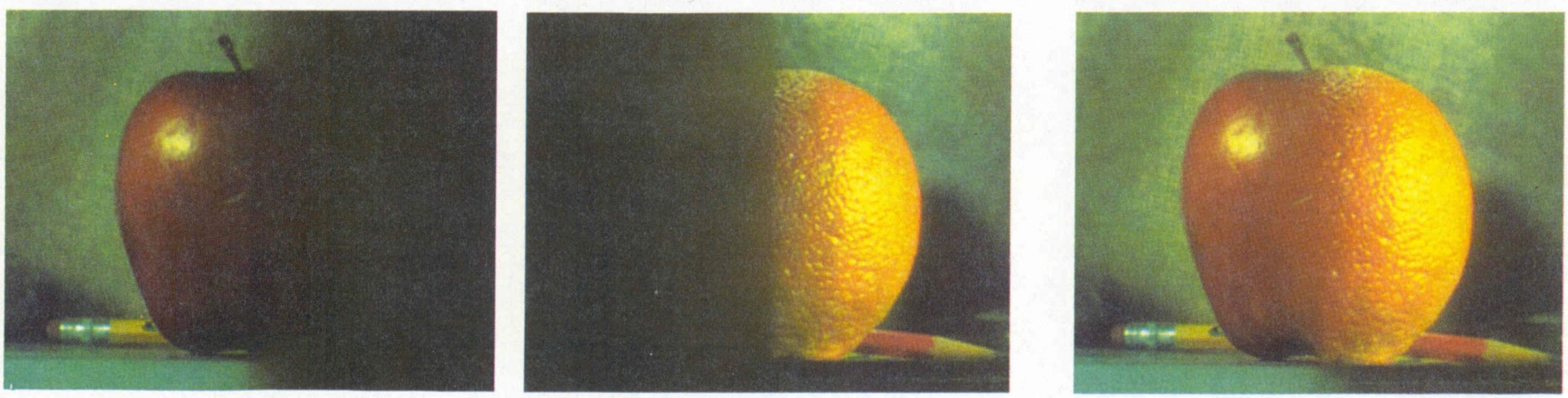
- To avoid seams
  - Window >= size of largest prominent feature

- To avoid ghosting
  - Window/2 <= size of smallest prominent feature

Pyramid blending





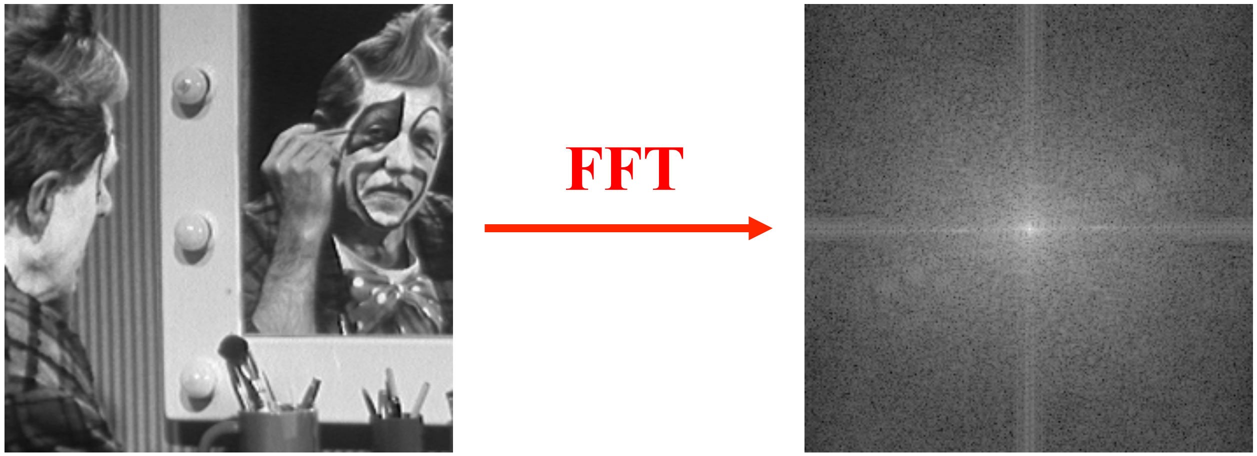


(d) (h) (l)

Create a Laplacian pyramid, blend each level (octave)

- Burt, P. J. and Adelson, E. H., A multiresolution spline with applications to image mosaics, ACM Transactions on Graphics, 42(4), October 1983, 217-236.

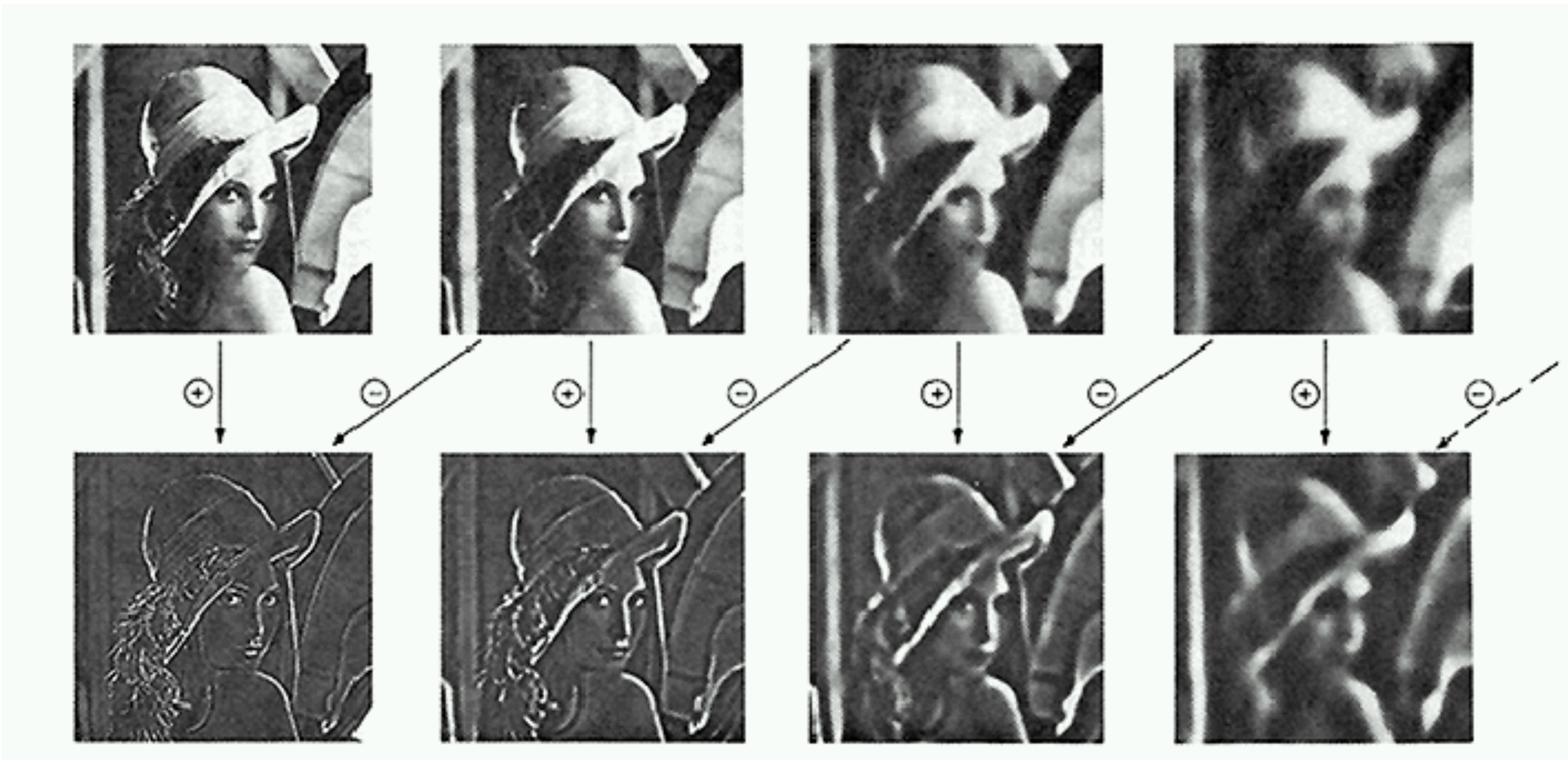
What if the Frequency Spread is Wide



- 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*

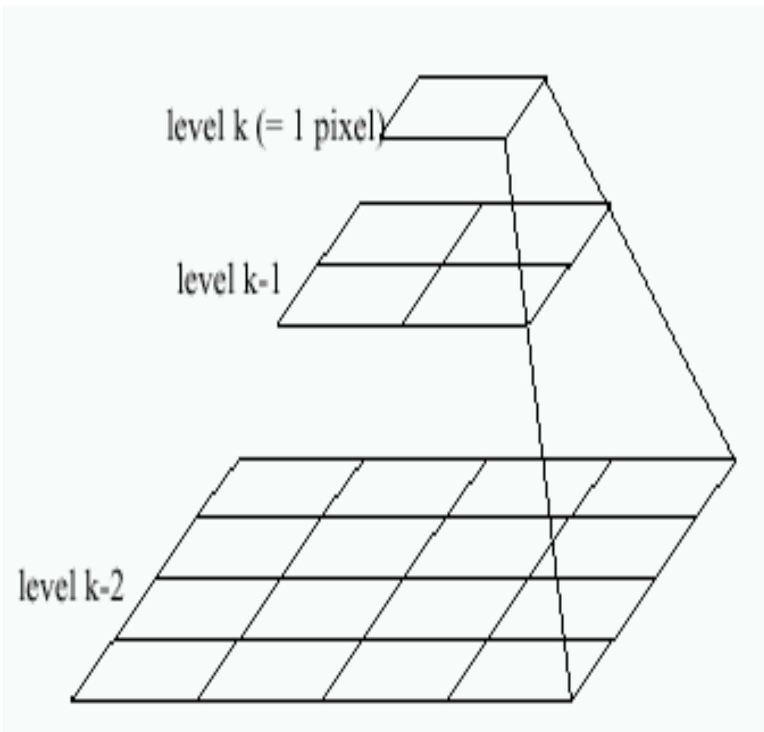
Octaves in the Spatial Domain

Lowpass Images

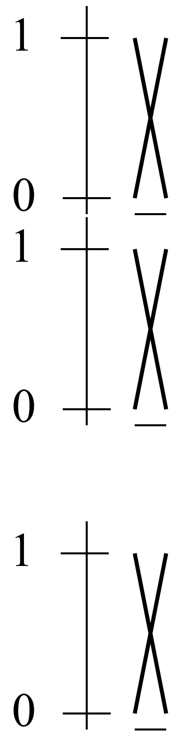


- Bandpass Images

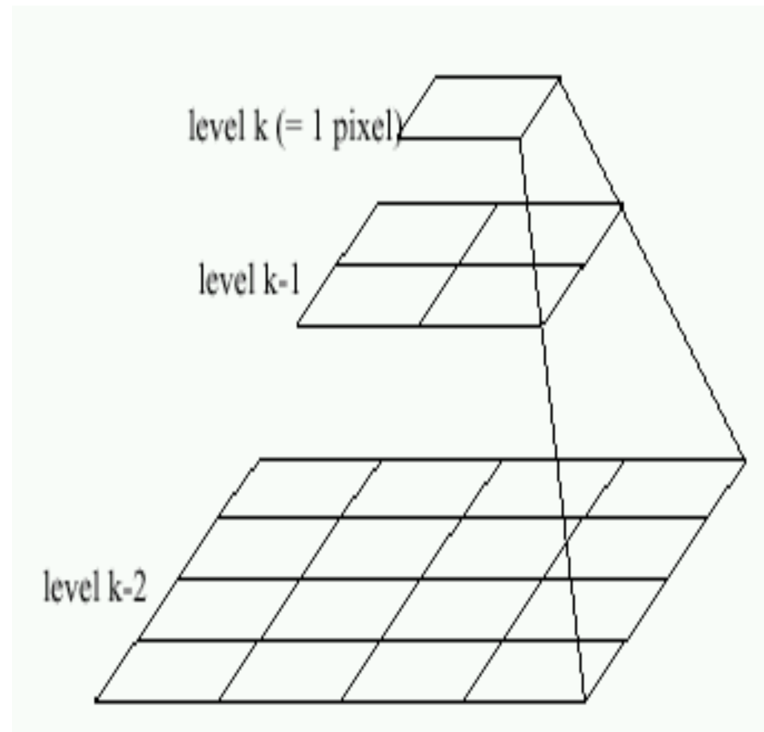
Pyramid Blending



Left pyramid

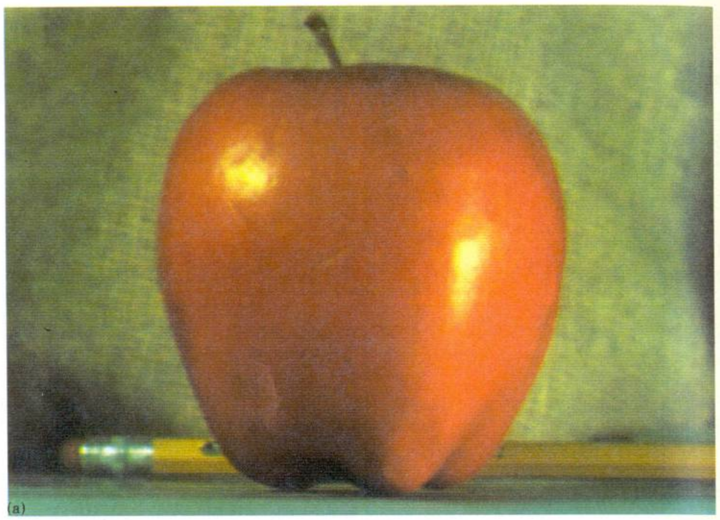


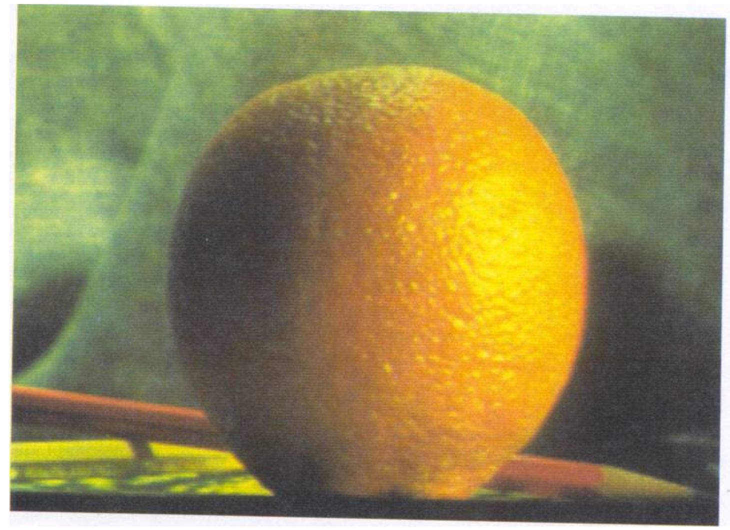
blend

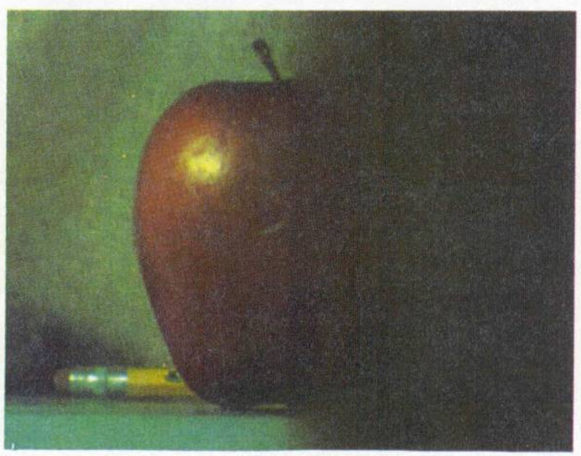


Right pyramid

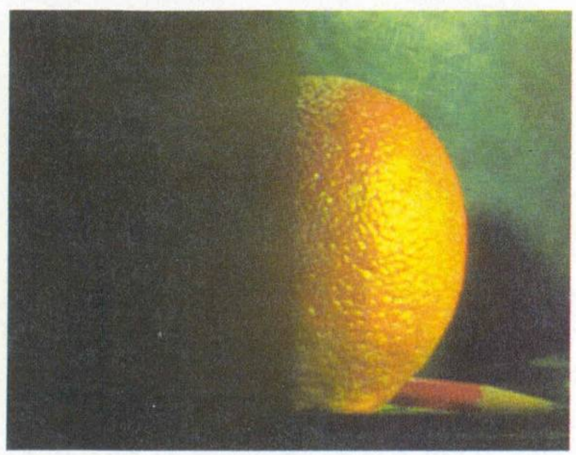
Pyramid Blending



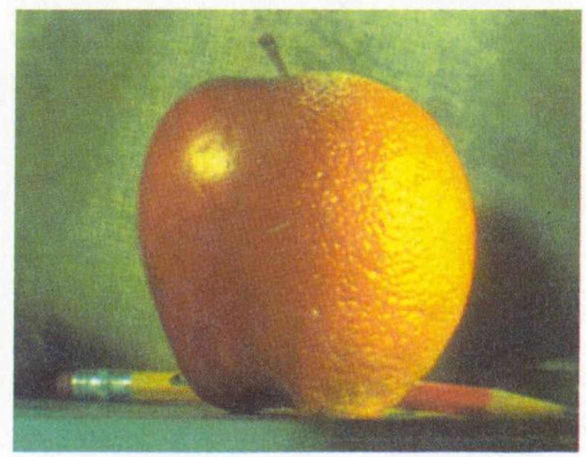




(d)



(h)

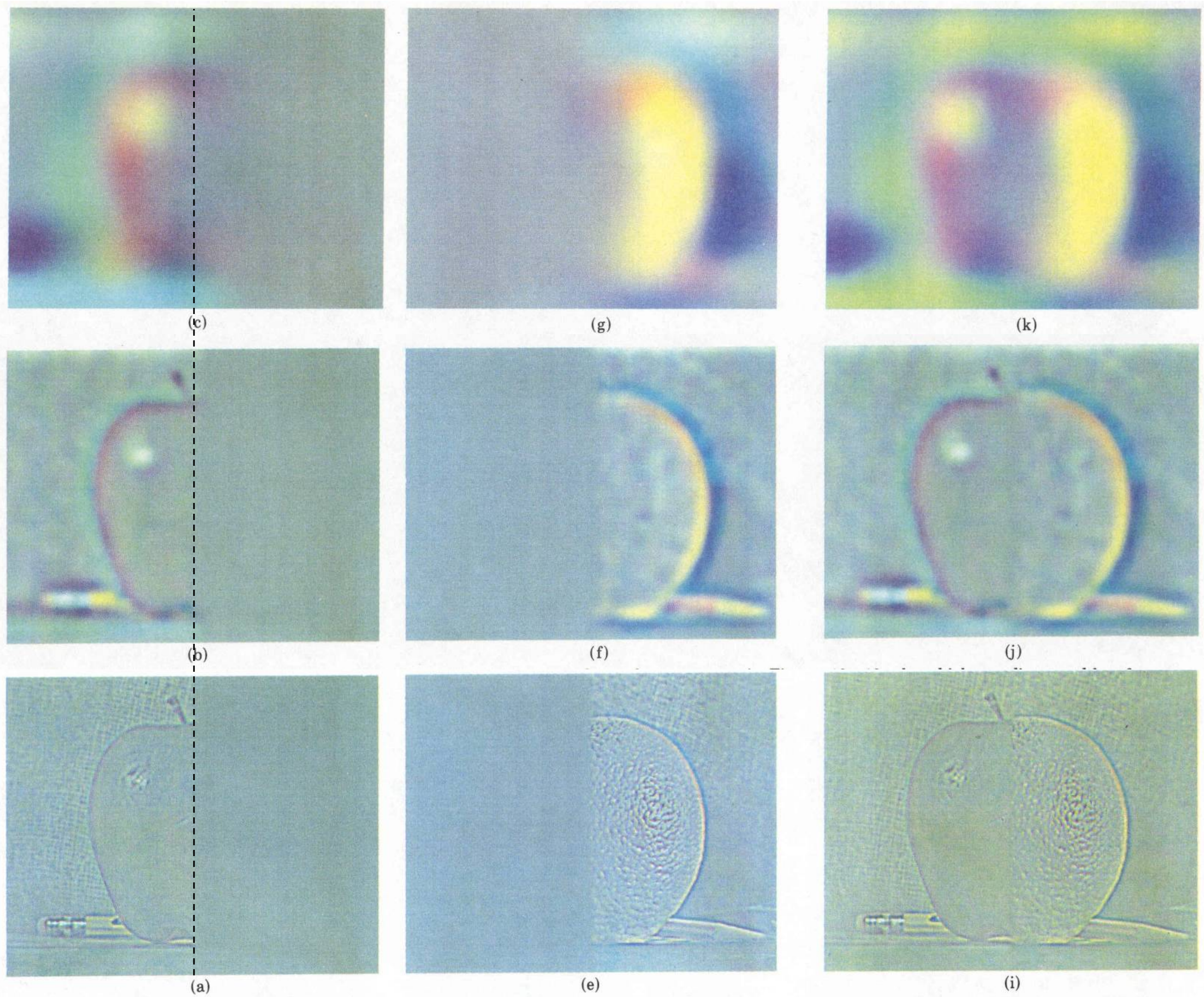


(l)

laplacian level 4

laplacian level 2

laplacian level 0

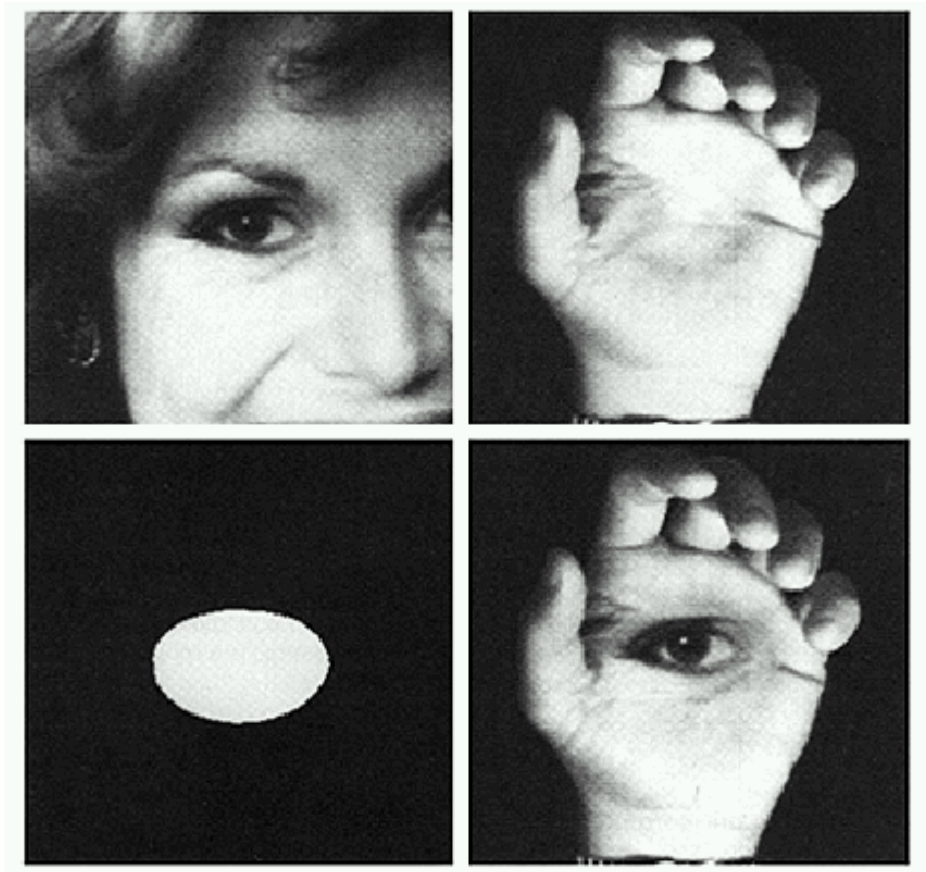


left pyramid

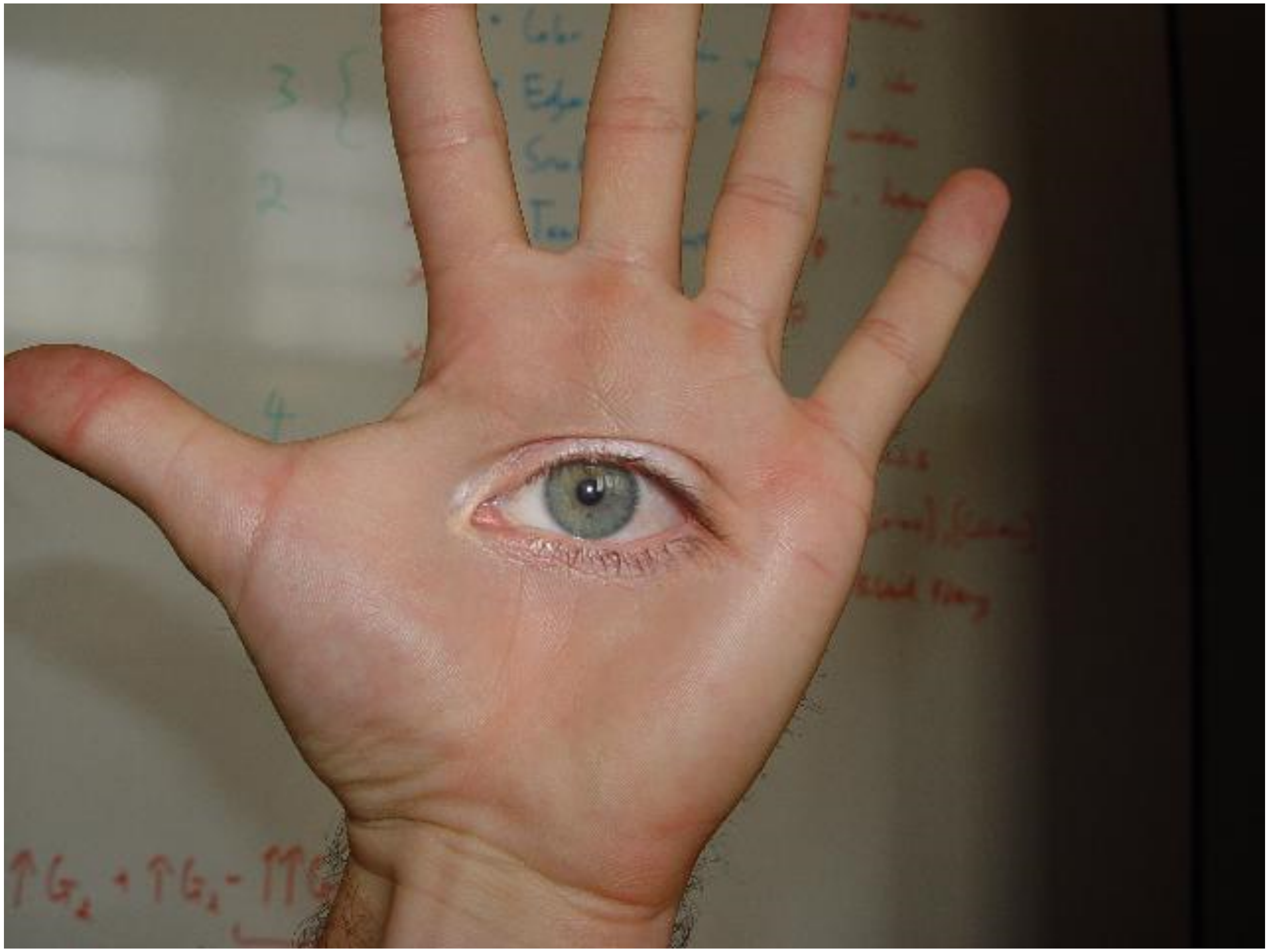
right pyramid

blended pyramid

Blending Regions



Horror Photo



© david dmartin (Boston College)

Simplification: Two-band Blending

- Brown & Lowe, 2003
  - Only use two bands: high freq. and low freq.

2-band Blending

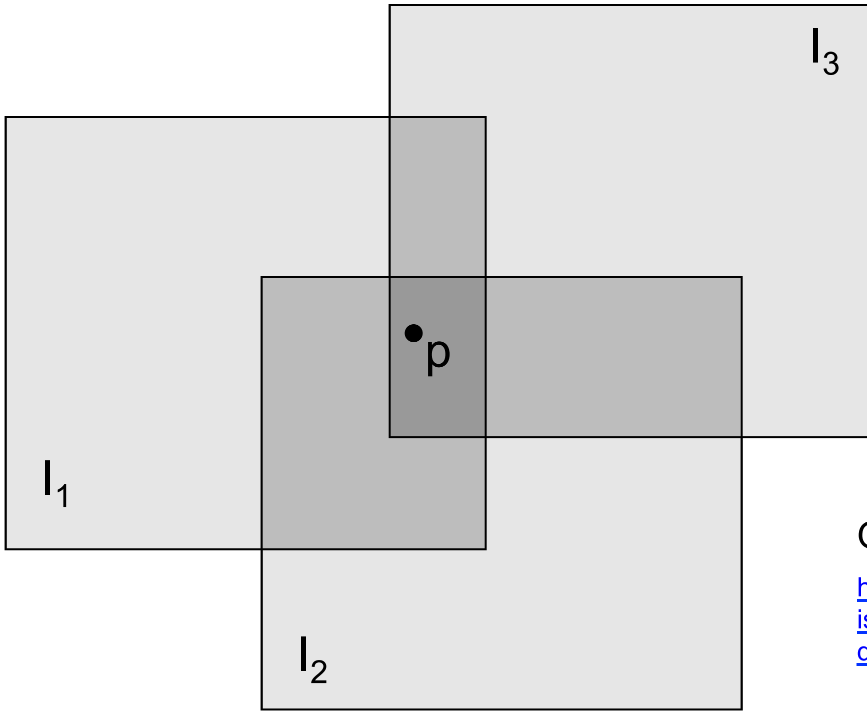
Low frequency (λ > 2 pixels)

High frequency (λ < 2 pixels)

Linear Blending

2-band Blending

Alpha Blending



Optional: see Blinn (CGA, 1994) for details:

http://ieeexplore.ieee.org/iel1/38/7531/00310740.pdf?isNumber=7531&prod=JNL&arnumber=310740&arSt=83&ared=87&arAuthor=Blinn%2C+J.F.

Encoding blend weights: $I(x,y) = (\alpha R, \alpha G, \alpha B, \alpha)$

color at p = $$\frac{(\alpha_1 R_1,\ \alpha_1 G_1,\ \alpha_1 B_1) + (\alpha_2 R_2,\ \alpha_2 G_2,\ \alpha_2 B_2) + (\alpha_3 R_3,\ \alpha_3 G_3,\ \alpha_3 B_3)}{\alpha_1 + \alpha_2 + \alpha_3}$$

Implement this in two steps:

1. accumulate: add up the (α premultiplied) RGBα values at each pixel
2. normalize: divide each pixel's accumulated RGB by its α value

   Q: what if $\alpha = 0$?

Project 3

- Take pictures on a tripod (or handheld)
- Warp to spherical coordinates (optional if using homographies to align images)
- Extract features
- Align neighboring pairs using RANSAC
- Write out list of neighboring translations
- Blend the images
- Correct for drift
- Now enjoy your masterpiece!

Some panorama examples

- Every image on Google Streetview

Other types of mosaics

- Can mosaic onto *any* surface if you know the geometry
  - See NASA's Visible Earth project for some stunning earth mosaics
    - http://earthobservatory.nasa.gov/Newsroom/BlueMarble/
    - Click for images…

- https://t.co/qean7Alb7p

Don’t blend, CUT!

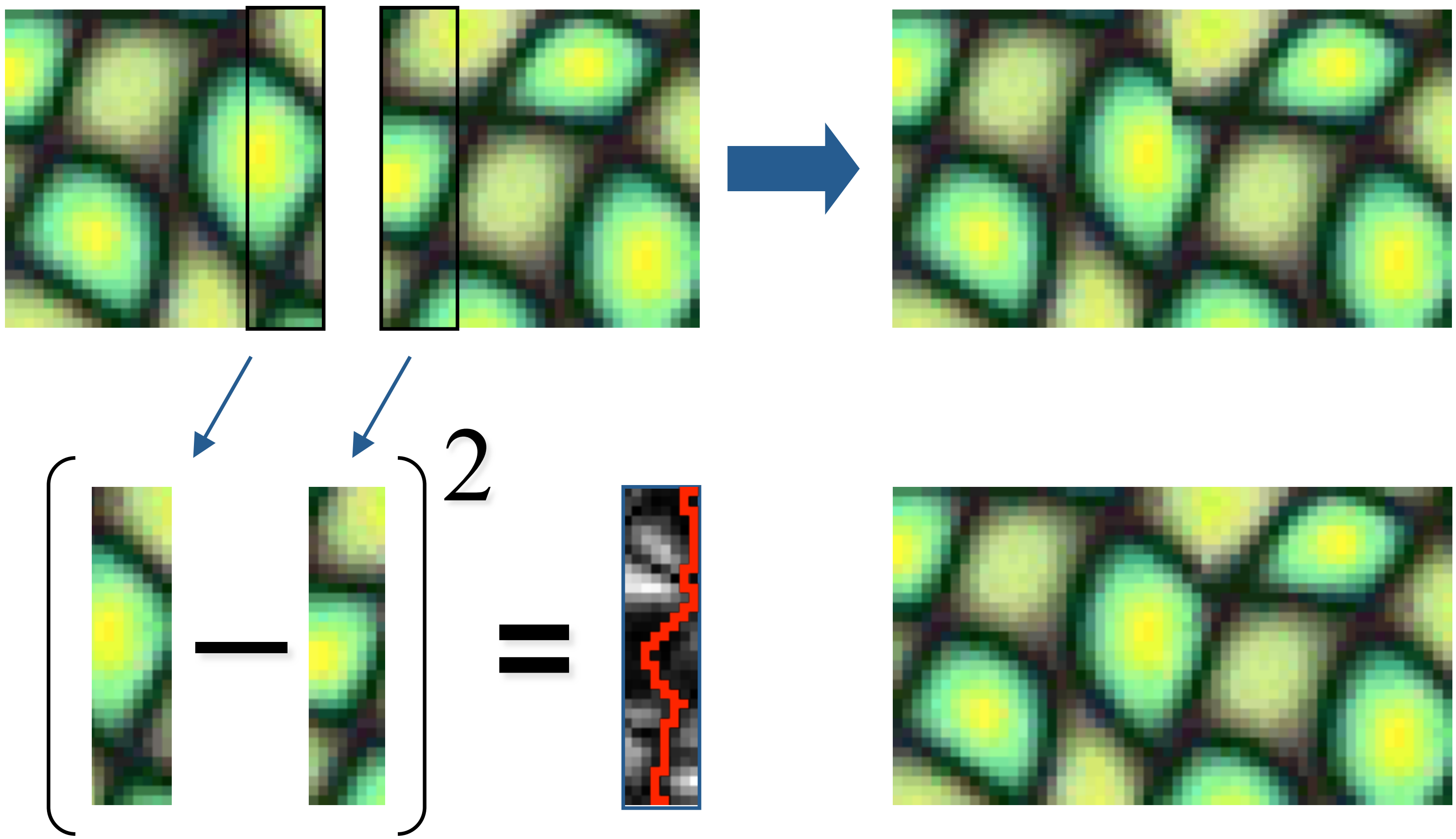
Moving objects become ghosts

- 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

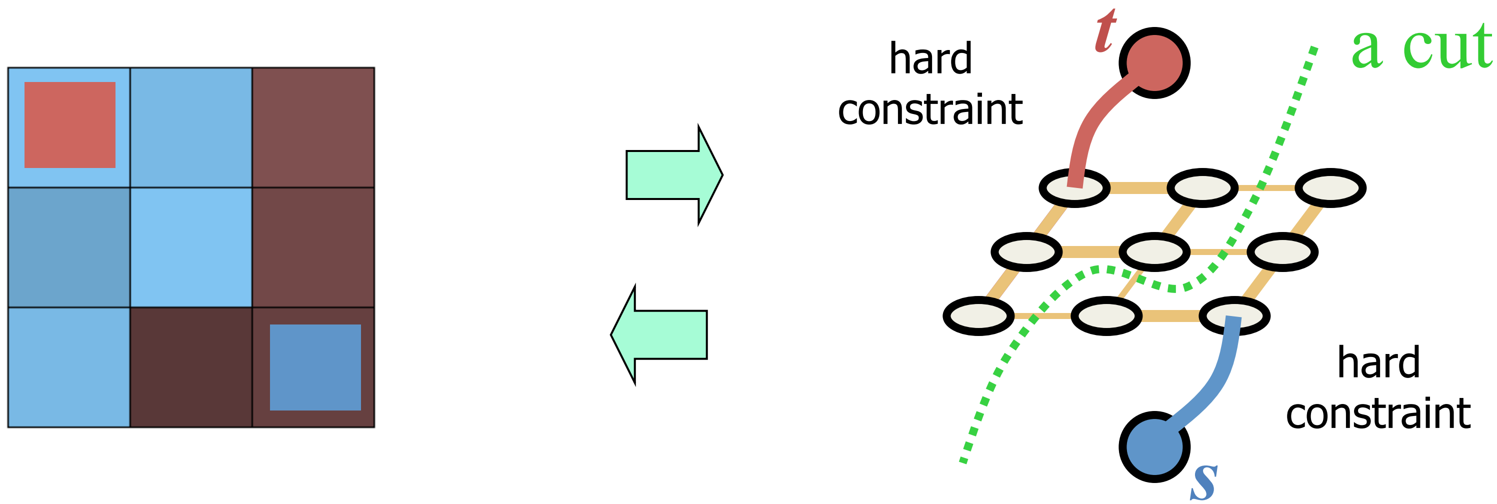
Shai Avidan
Mitsubishi Electric Research Labs

Ariel Shamir
The Interdisciplinary Center & MERL

Graphcuts

- What if we want similar "cut-where-things-agree" idea, but for closed regions?
  – Dynamic programming can't handle loops

Graph cuts
(simple example à la Boykov&Jolly, ICCV’ 01)



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)

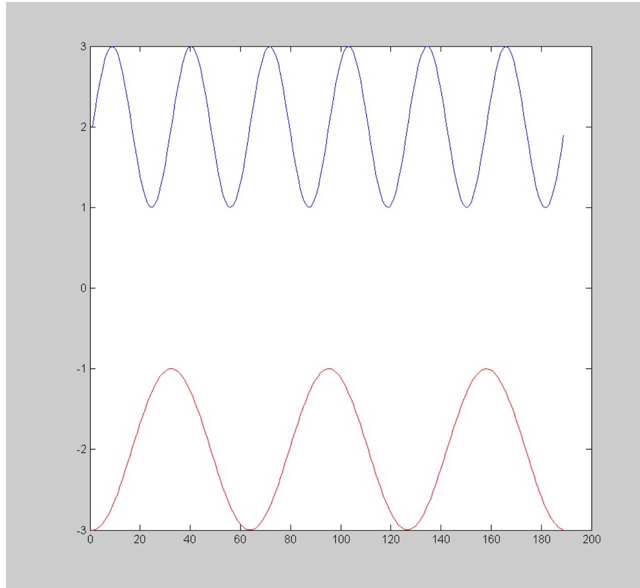
Interactive segmentation using graphcuts

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
    - Reintegrate

Gradient Domain blending (1D)

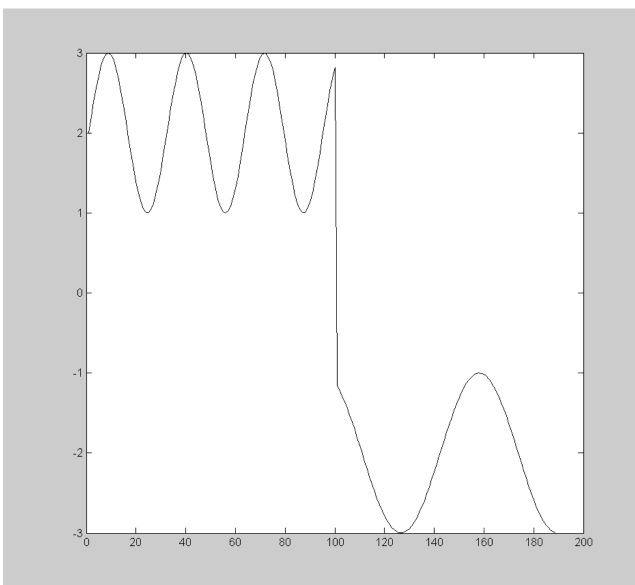
Two signals

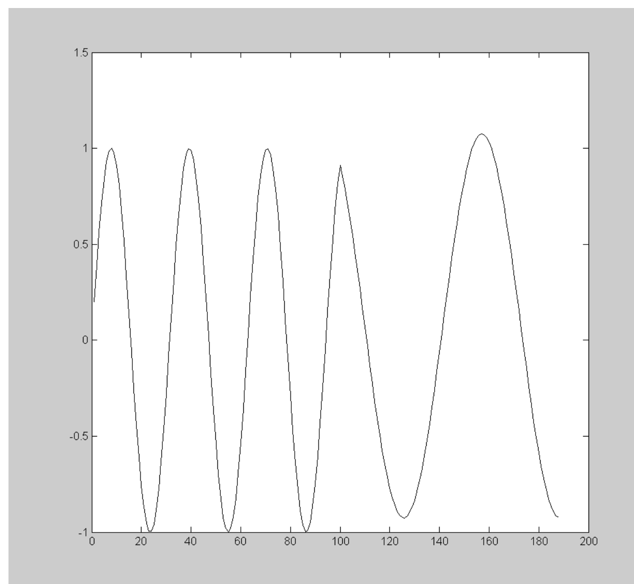


bright

dark

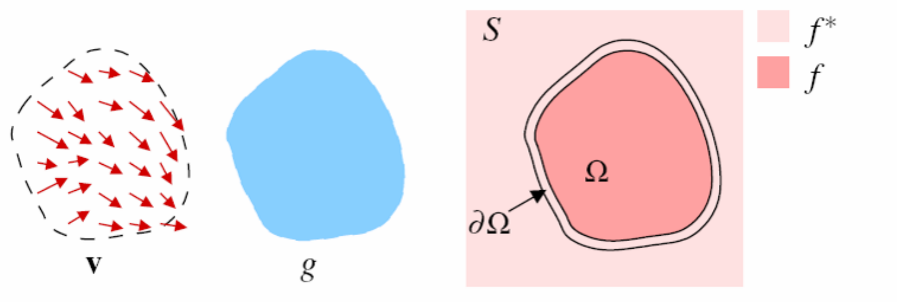
Regular blending





Blending derivatives

Gradient Domain Blending (2D)



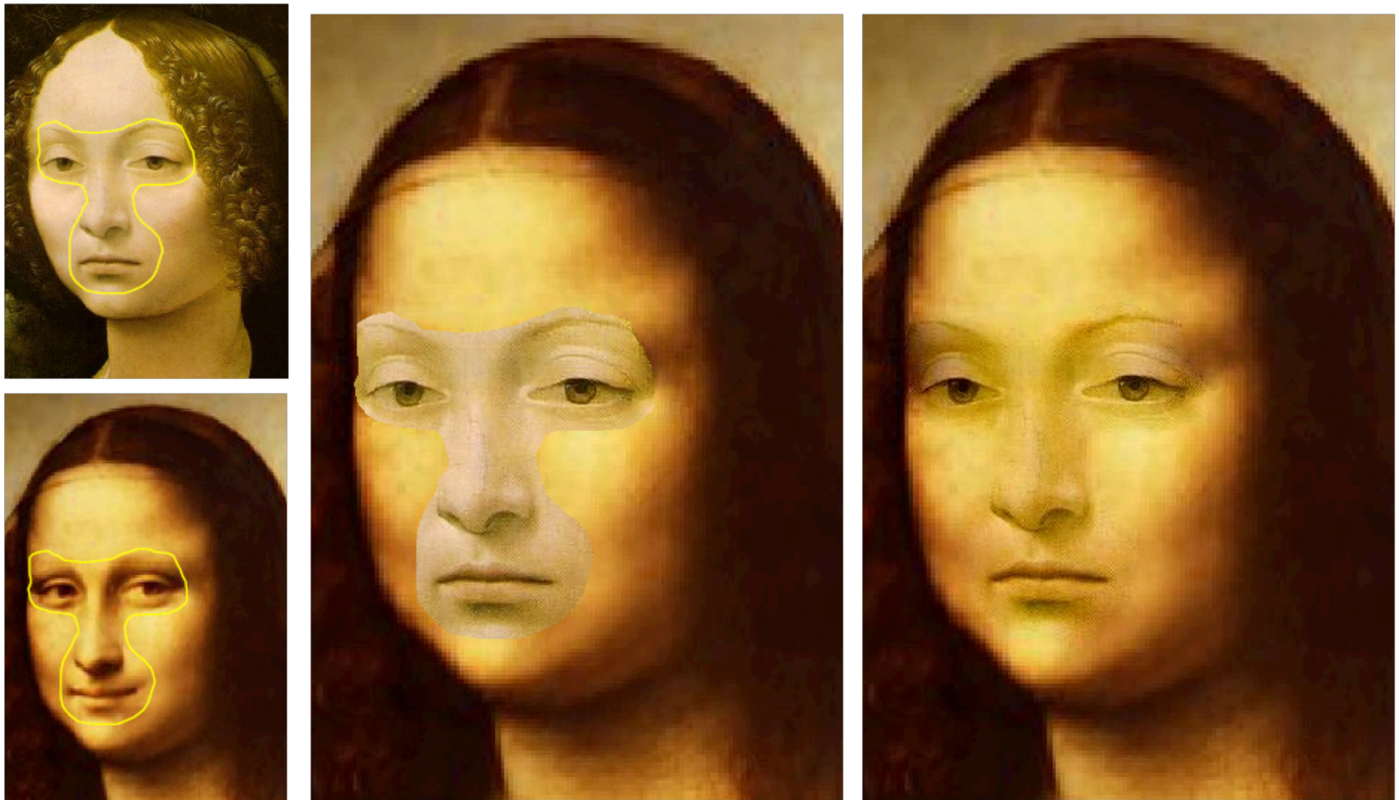
- Trickier in 2D:
  - Take partial derivatives dx and dy (the gradient field)
  - Edit (smooth, blend, feather, etc)
  - Reintegrate
  - Find the most agreeable solution
    - Equivalent to solving Poisson equation
    - Can use FFT, deconvolution, multigrid solvers, etc.

Perez et al., 2003

sources destinations cloning seamless cloning

sources/destinations cloning seamless cloning

Perez et al, 2003



source/destination cloning seamless cloning

- Limitations:
  - Images need to be very well aligned

Putting it all together

- Compositing images
  - Have a clever blending function
    - Feathering
    - 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