Image compositing

15-463, 15-663, 15-862
Computational Photography
Fall 2018, Lecture 8

http://graphics.cs.cmu.edu/courses/15-463

Course announcements

- If you haven’t started Homework 2 yet, you should do that ASAP.
- Today we will post more details about the final project.
- Guest lecture on Monday: Ravi Teja Mullapudi, “High-performance image processing.”
- RI Seminar on Friday: Anat Levin, “Light-sensitive displays.”

Overview of today’s lecture

- Leftover from color lecture.
- Some motivational examples.
- Cut-and-paste.
- Alpha (linear) blending.
- Multi-band blending.
- Seam stitching.

Slide credits

Most of these slides were adapted from:

- Kris Kitani (15-463, Fall 2016).

Some slides were inspired or taken from:

- Fredo Durand (MIT).
- James Hays (Georgia Tech).

Some motivational examples

Gangster, Frankie Yale, killed by a drive-by in Brooklyn in 1928.

A tragic photo from 1959 after three-year-old Martha Cartagena was killed while riding her tricycle in Brooklyn

In 1958 there was a fatal fire at the Elkins Paper & Twine Co. on Wooster Street in SoHo. The building burned to the ground.


MEDIMAX RINGCENTER
MEDIMAX
(c)Sergey Larenkov

Berlin 65 years later

Berlin, 1945/2010, Mehringdamm

Forrest Gump (1994)

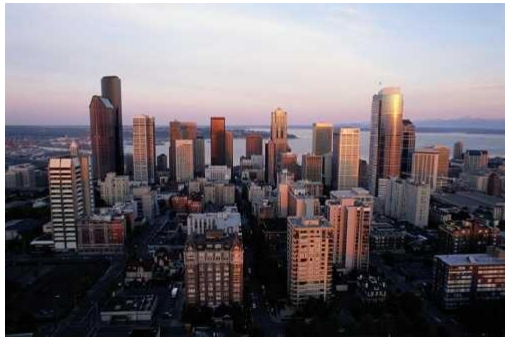
Techniques for compositing

- Cut-and-paste.
- Alpha (linear) blending.
- Multi-band blending.
- Poisson blending.
- Seam stitching.

Cut-and-paste

Cut and paste procedure

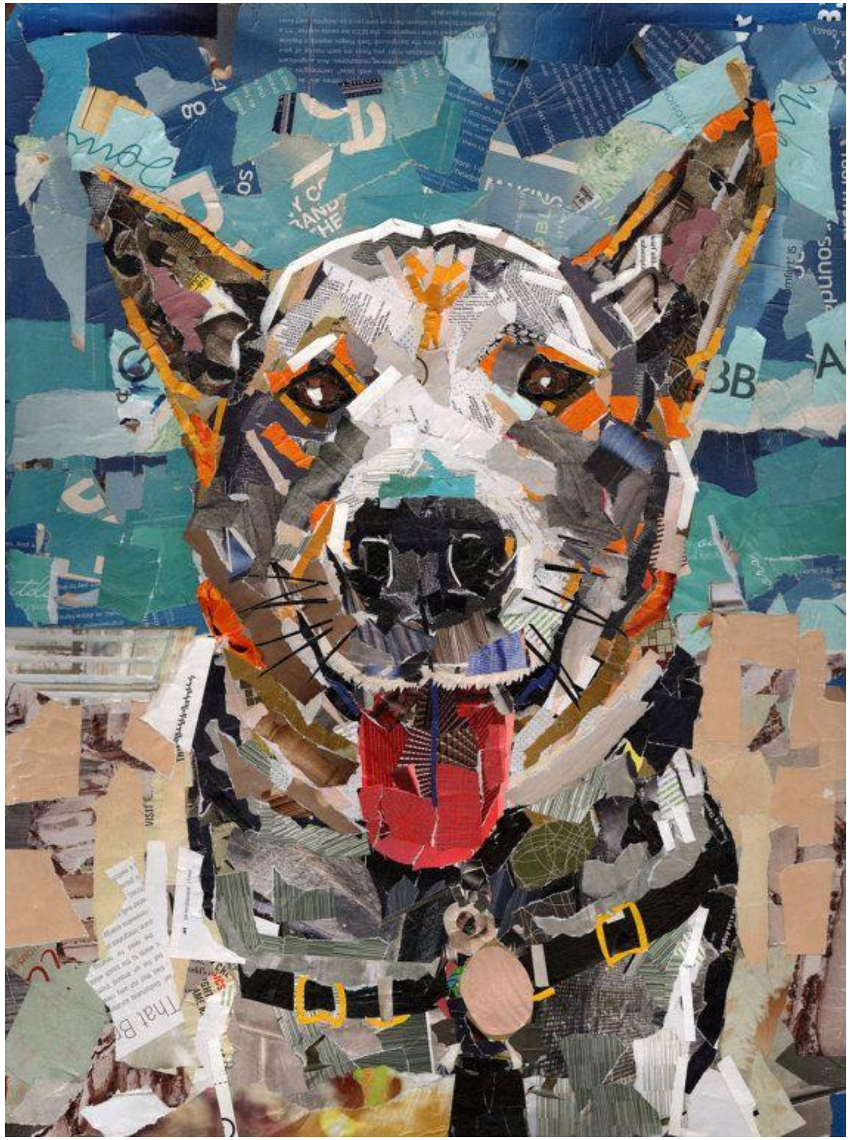
1. Extract Sprites (e.g., using *Intelligent Scissors* in Photoshop)



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

You may have also heard it as collaging

Cut and paste



Sometimes it produces visually compelling results.

Cut and paste

Other times, not so much.

What is wrong with this composite?

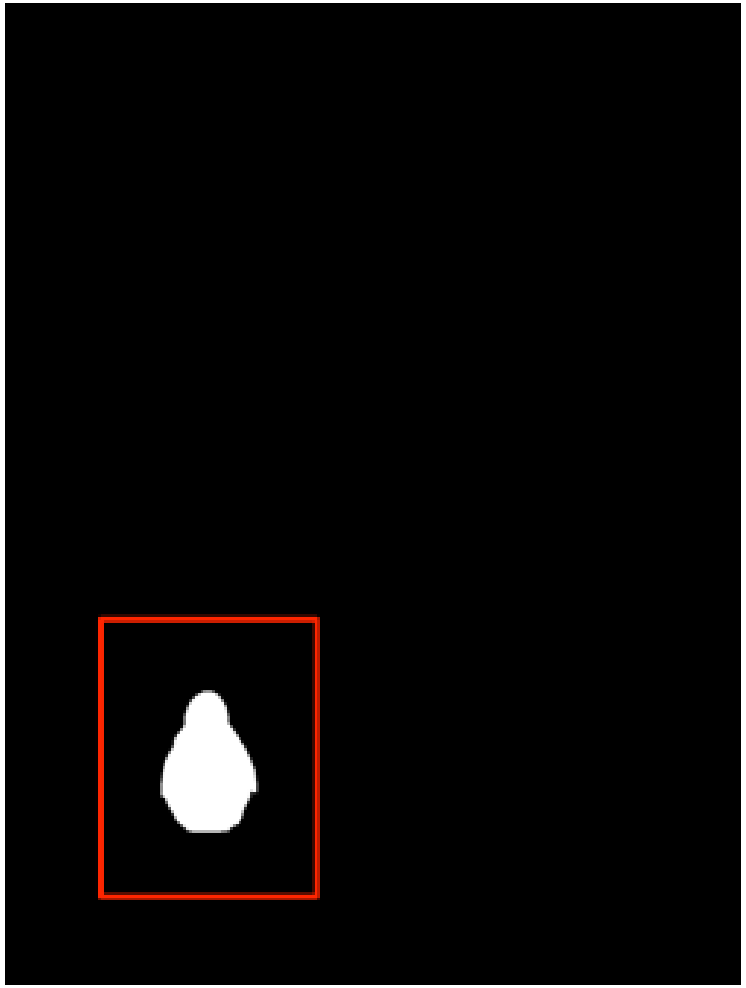
Alpha (linear) blending

Alpha blending

foreground

background

mask

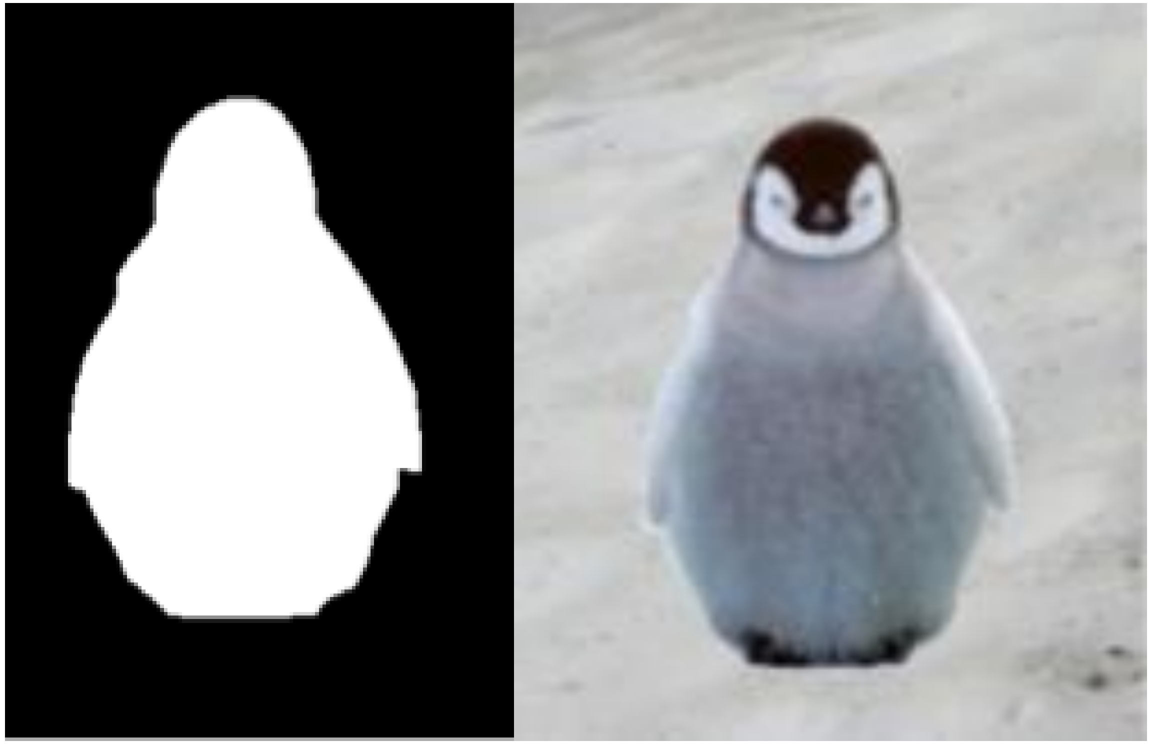


output

a.k.a. alpha matte
or alpha composite

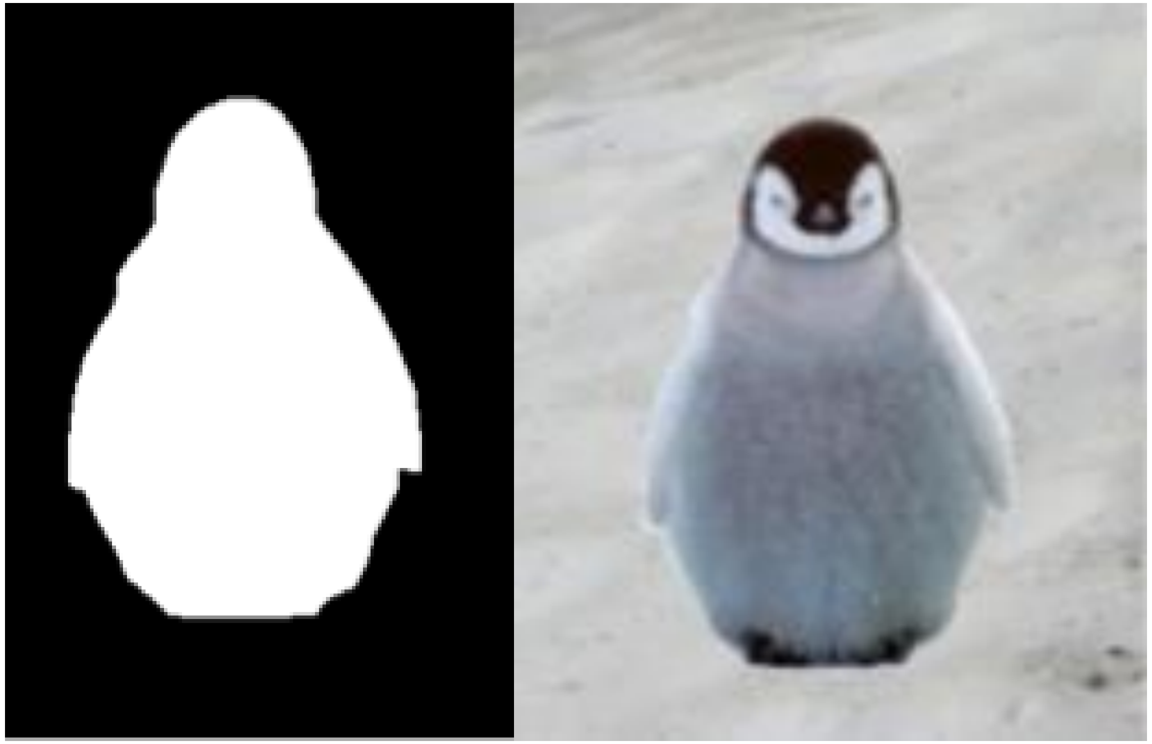
output = foreground * mask + background * (1-mask)

Binary alpha mask



Does this look unnatural?

Binary alpha mask

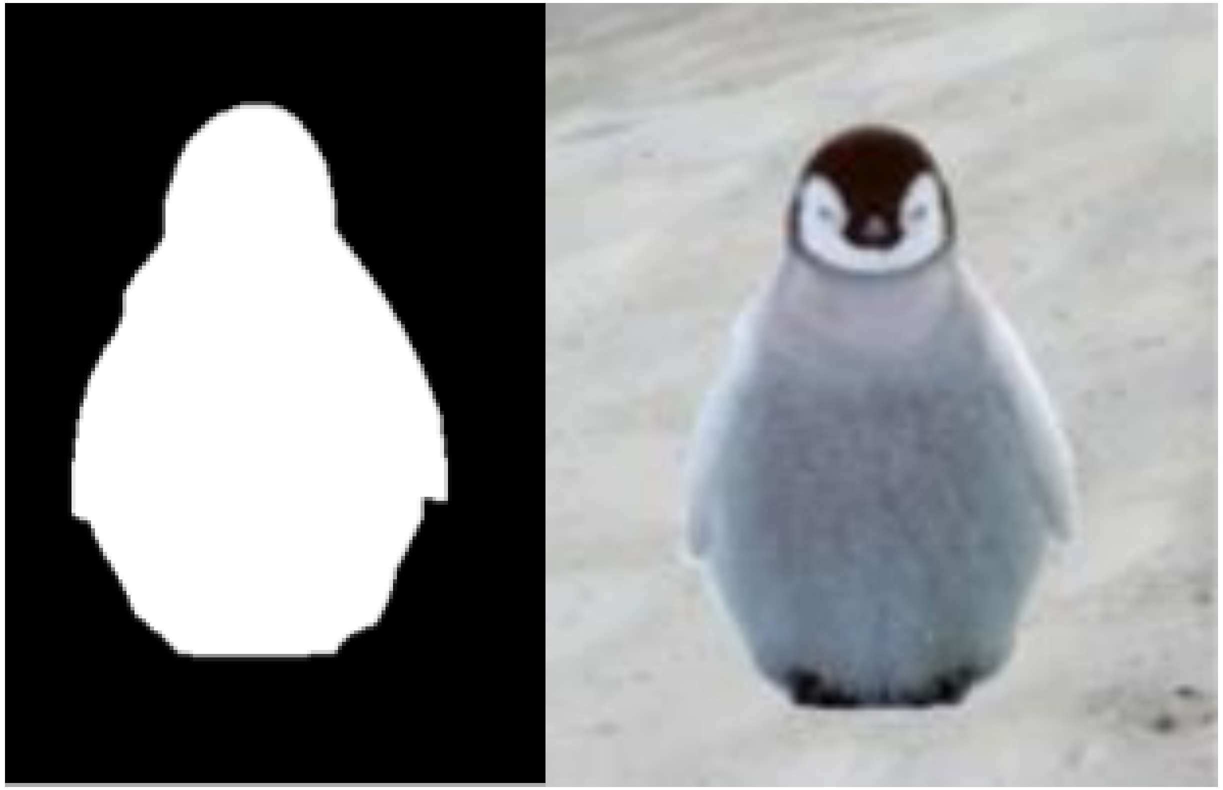


Does this look unnatural?

How can we fix it?

Non-binary alpha mask

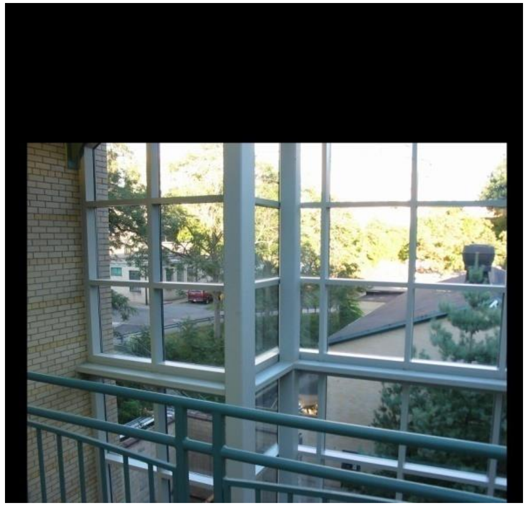
binary alpha mask



feathering (smoothed alpha)

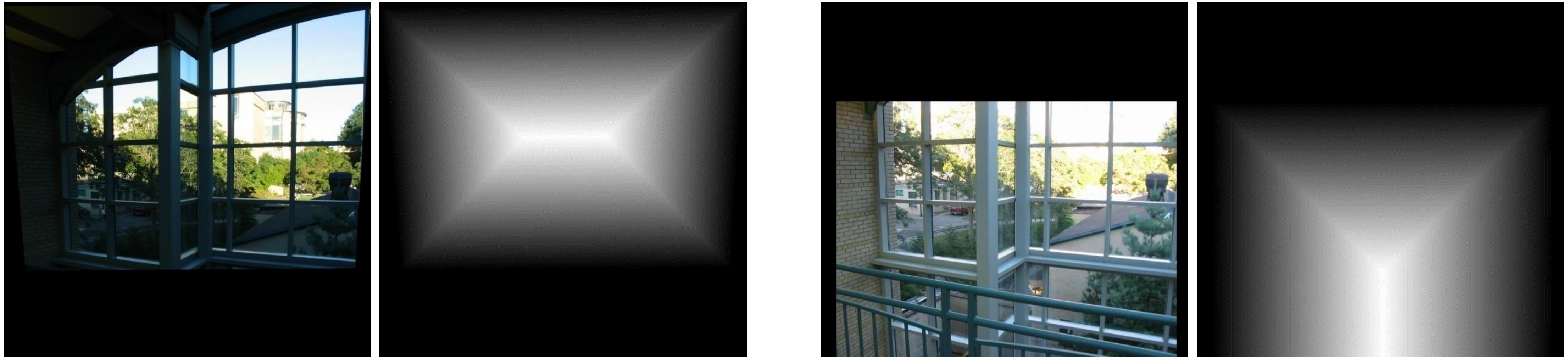
How would you implement feathering?

Setting the alpha mask: center seam



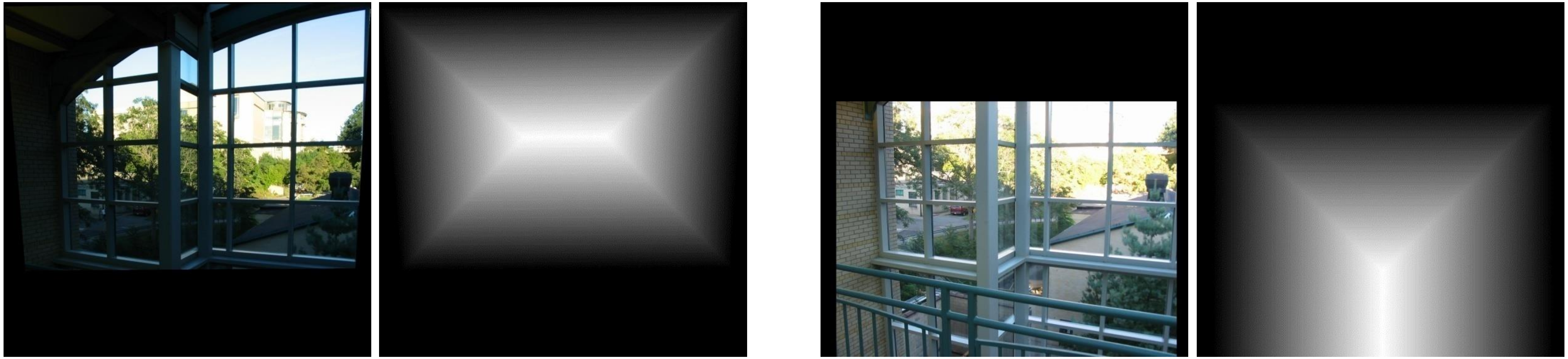
How would you create a binary alpha mask for these two images?

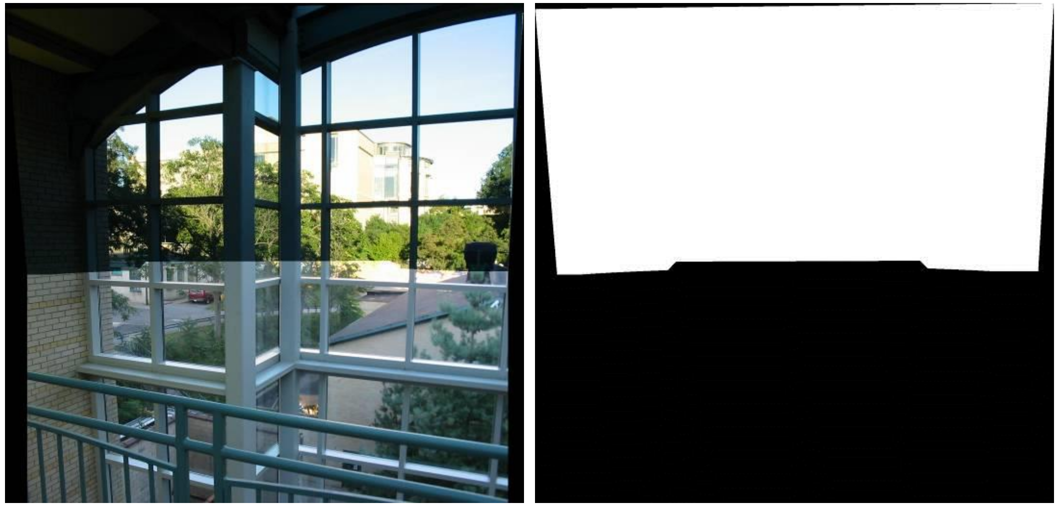
Setting the alpha mask: center seam



Step 1: Compute their distance transform (`bwdist`)

Setting the alpha mask: center seam





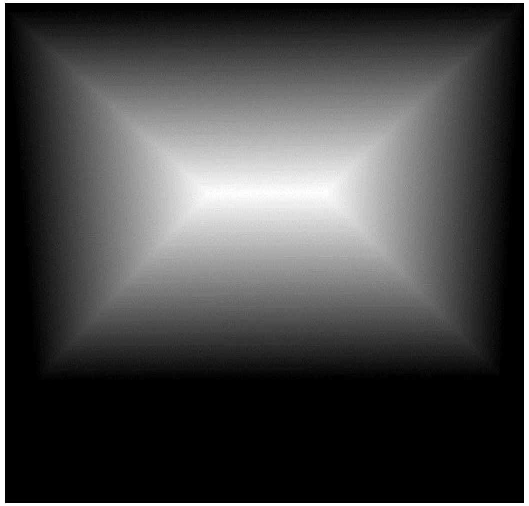
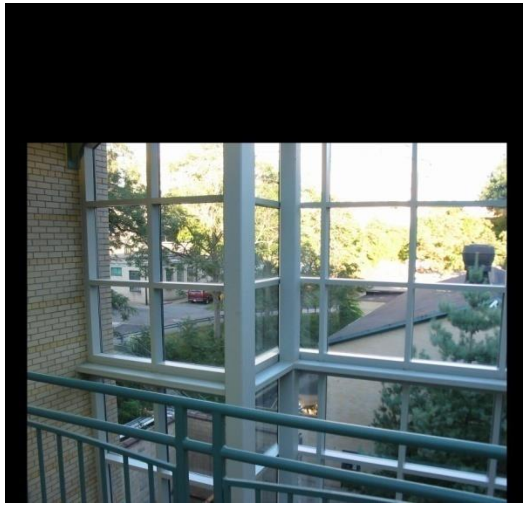
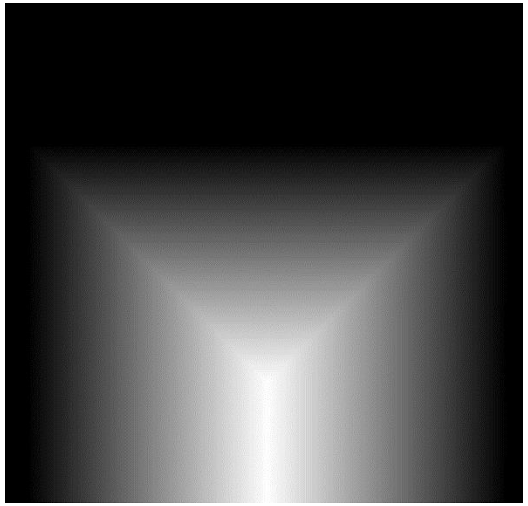
Step 2: set mask

```
alpha = logical(dtrans1>dtrans2)
```

Setting the alpha mask: center seam

Anything wrong with this alpha matte?

Setting the alpha mask: center seam

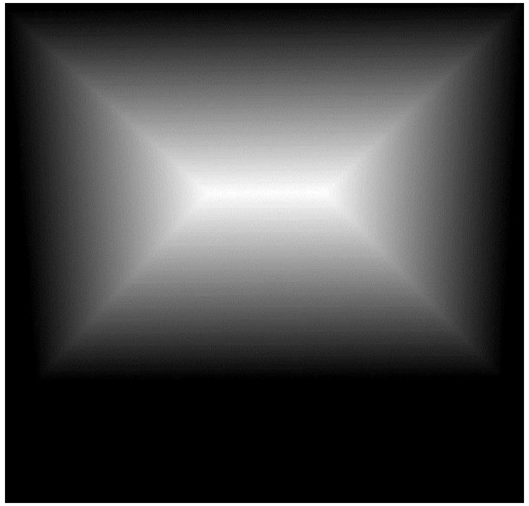
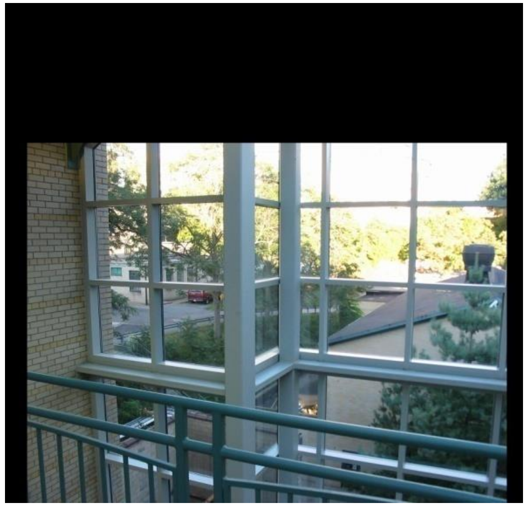
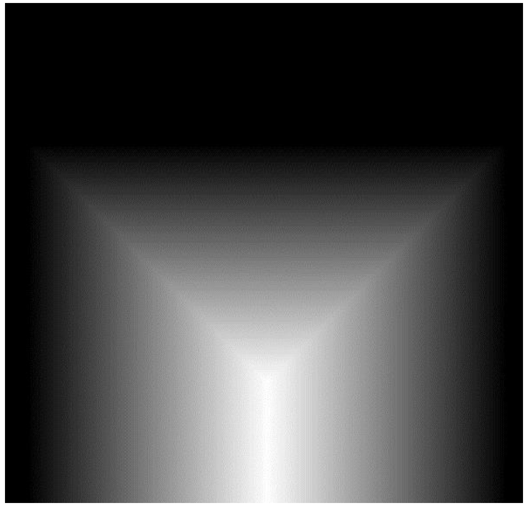

Step 3: blur the mask

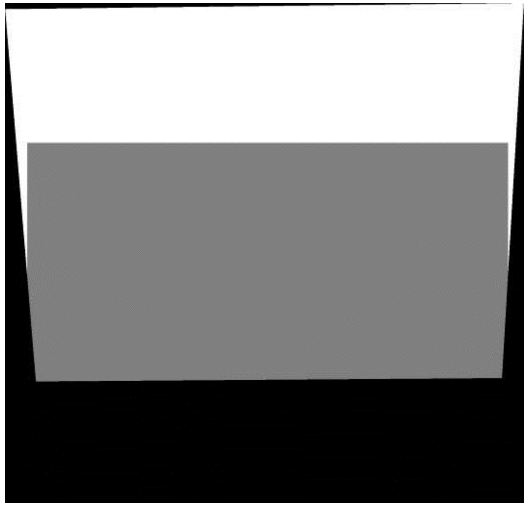
```
alpha = blur(alpha)
```

Setting the alpha mask: center seam

Still doesn't look terribly good

Setting the alpha mask: center seam

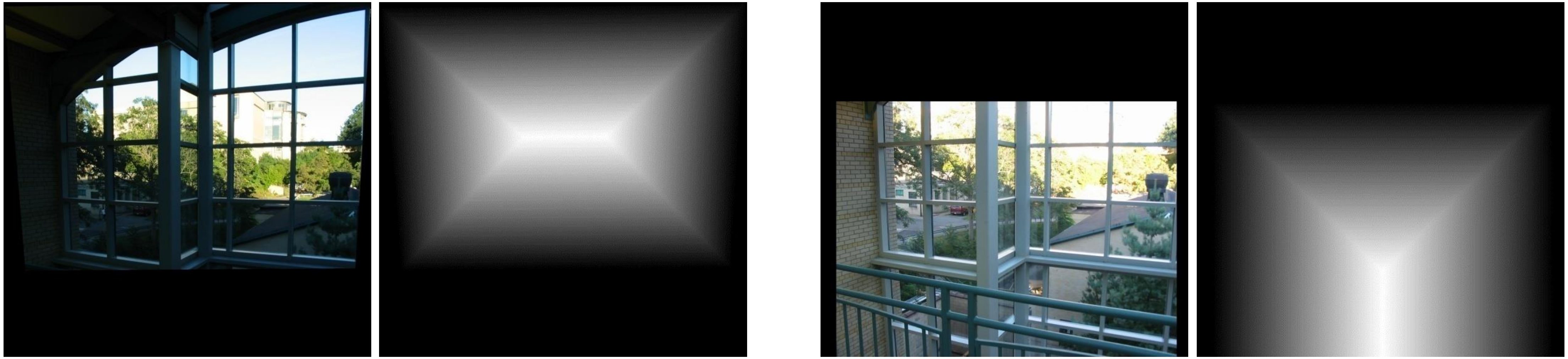
Step 4: go beyond blurring for non-binary

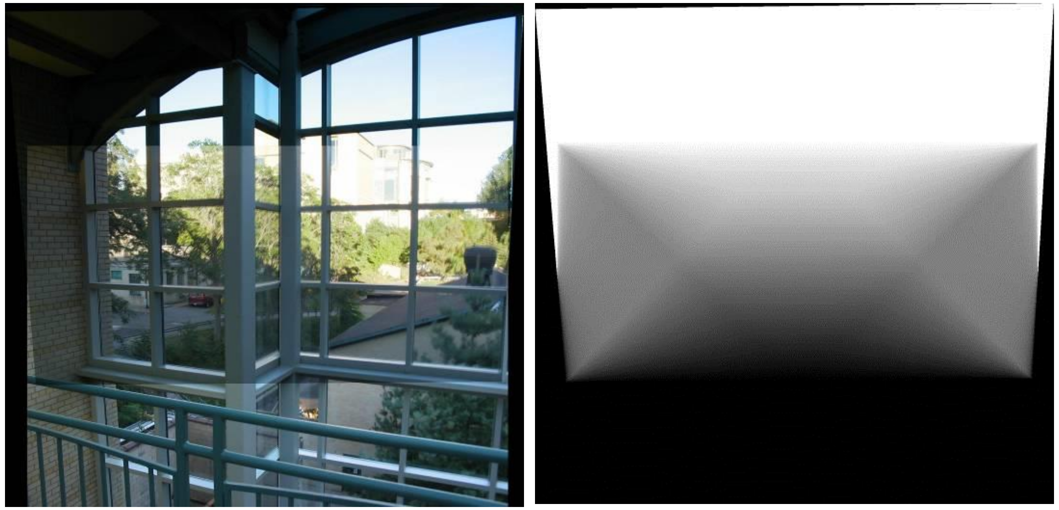
`alpha = 0.5` in overlap region

Setting the alpha mask: center seam

Still not OK

Setting the alpha mask: center seam





Step 5: more elaborate non-binary

```
alpha = dtrans1 / (dtrans1+dtrans2)
```

Setting the alpha mask: center seam

Looks better but some dangers remain.

Another blending example

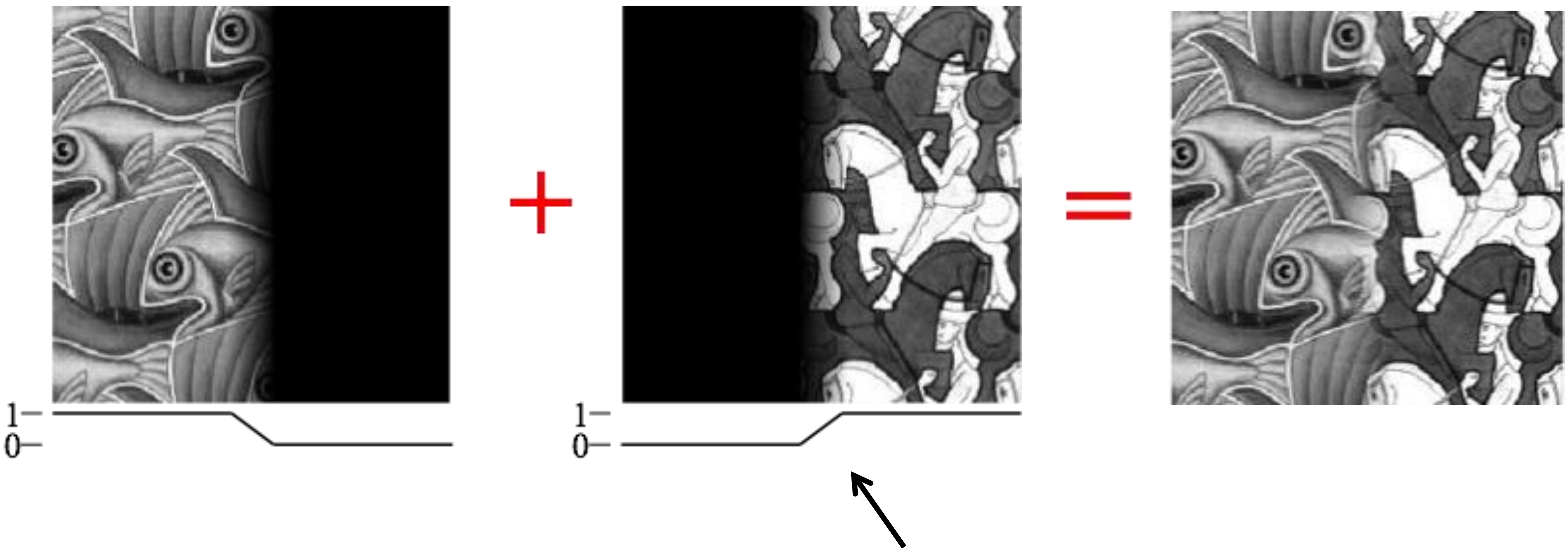
Let's blend these two images...

left side

right side

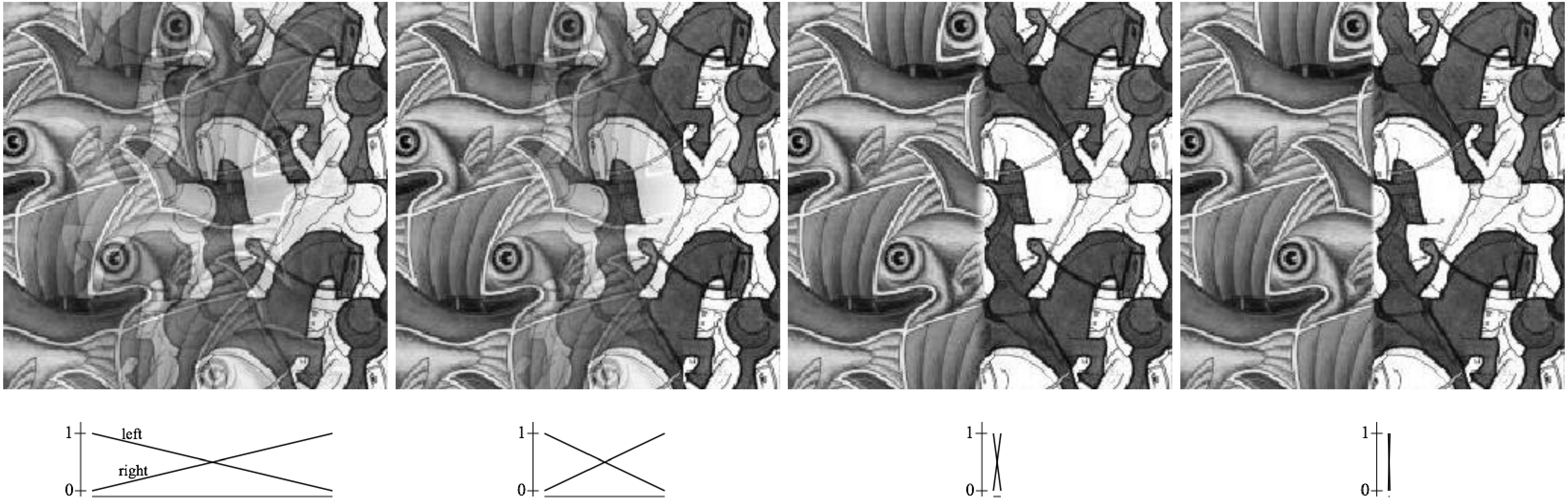
What kind of mask would you use?

Another blending example



How would you select this window?

Effects of different windows

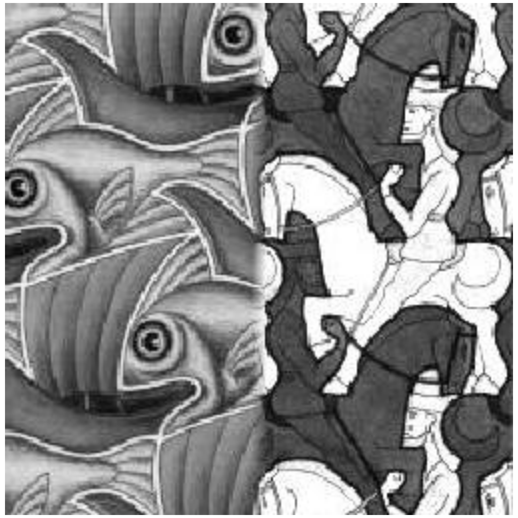


Bad windows: ghosting.

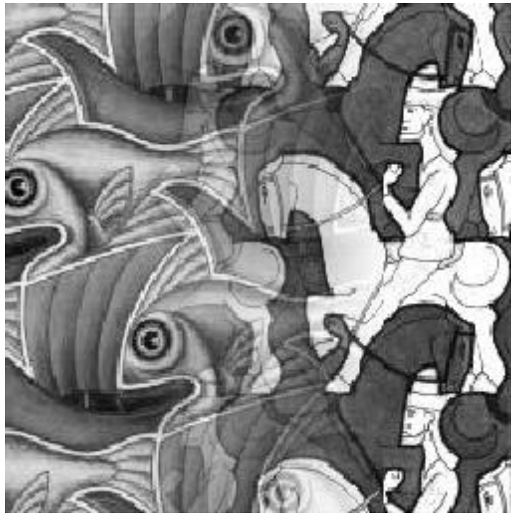
Good window: smooth but no ghosting.

Bad window: non-smooth seam.

What is a good window size?

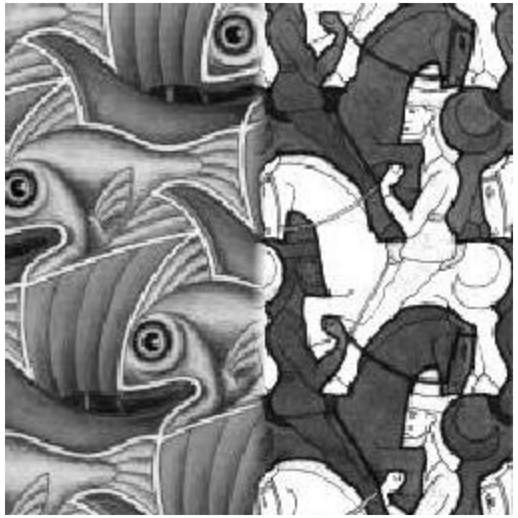


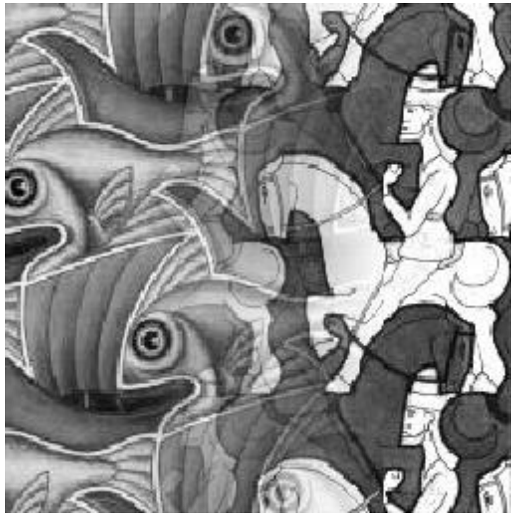
To avoid discontinuities:
window = size of largest prominent feature



To avoid ghosting:
window <= 2*size of smallest prominent feature

What is a good window size?





To avoid discontinuities:
window = size of largest prominent feature

To avoid ghosting:
window <= 2*size of smallest prominent feature

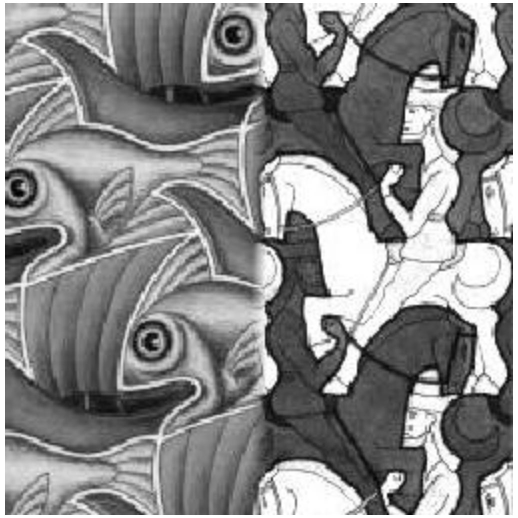
Fourier domain interpretation:

linear blending should work when:
image frequency content occupies roughly one “octave” (power of two)

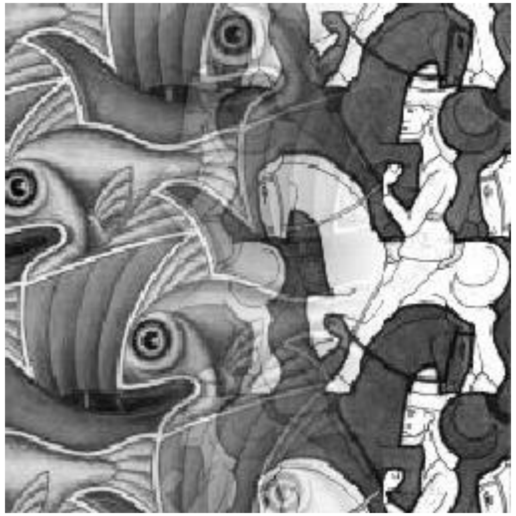
linear blending should work when:
largest frequency <= 2*size of smallest frequency

What if the frequency spread is too wide?

What is a good window size?



To avoid discontinuities:
window = size of largest prominent feature



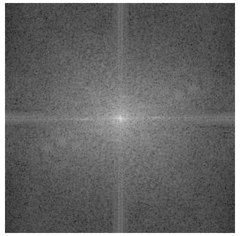
To avoid ghosting:
window <= 2*size of smallest prominent feature

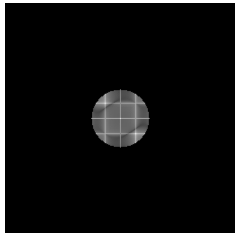
Fourier domain interpretation:

linear blending should work when: image frequency content occupies roughly one "octave" (power of two)

linear blending should work when: largest frequency <= 2*size of smallest frequency

Most natural images have a very wide frequency spread. What do we do then?

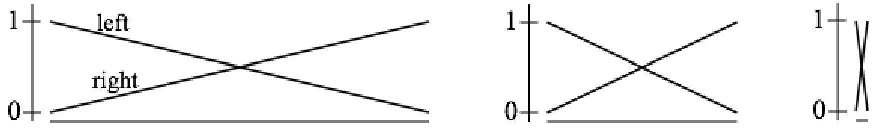




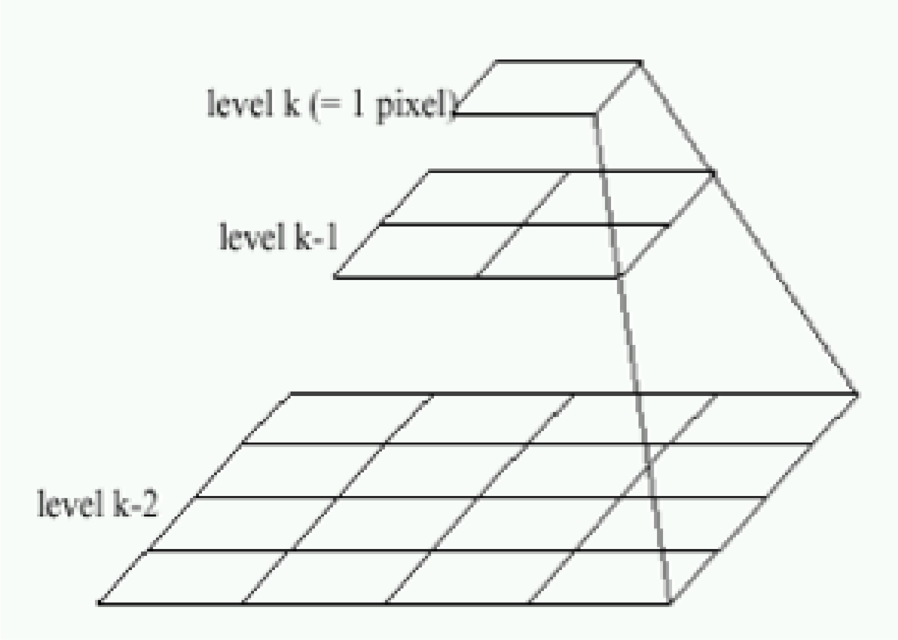
Multi-band blending

Time to use pyramids again

At low frequencies, blend slowly to avoid seams
At high frequencies, blend quickly to avoid ghosts



Which mask goes where?



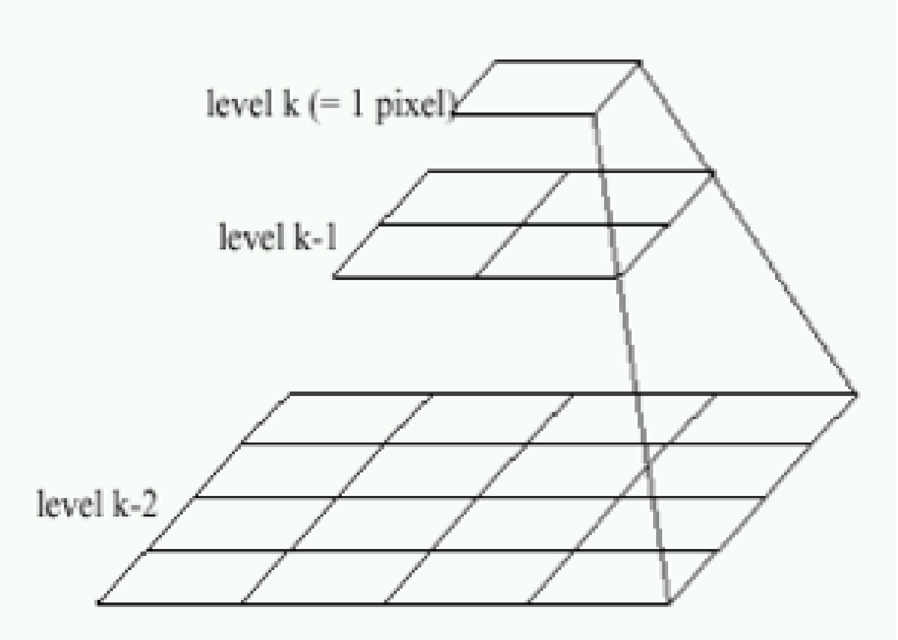
left image

?

?

?

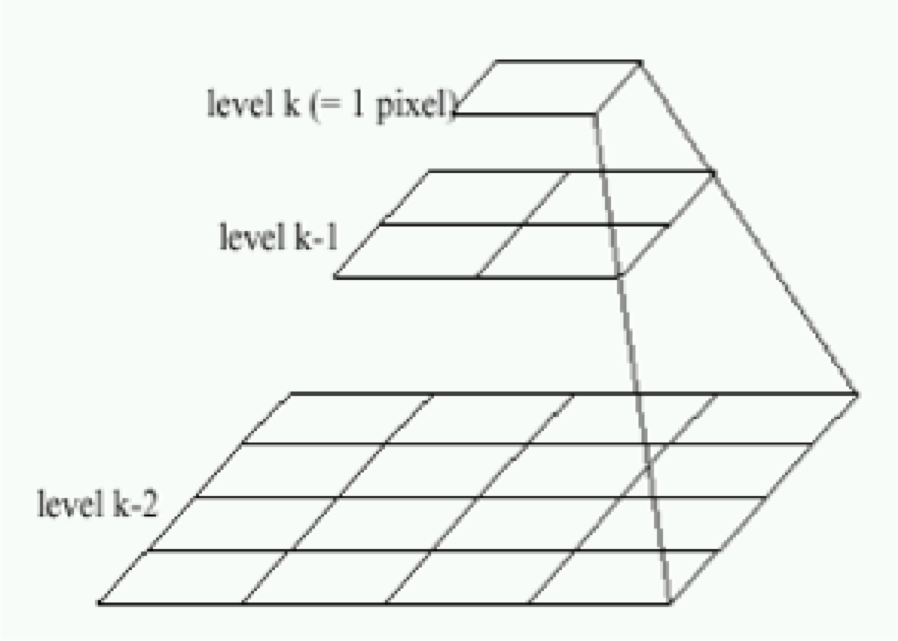
alpha mask



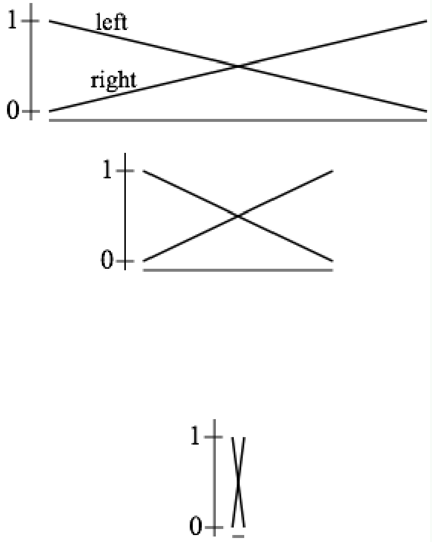
right image

Time to use pyramids again

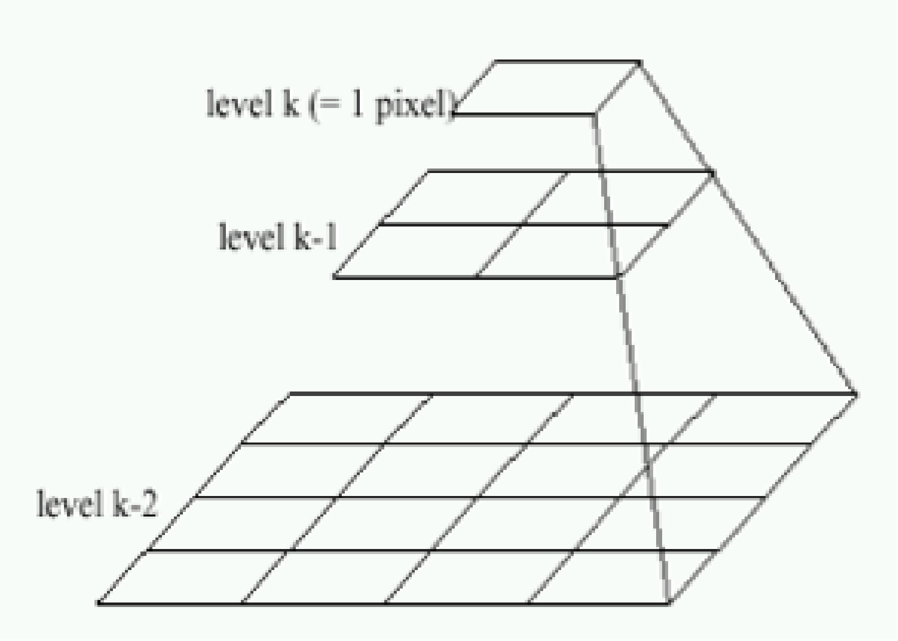
At low frequencies, blend slowly to avoid seams
At high frequencies, blend quickly to avoid ghosts



left image

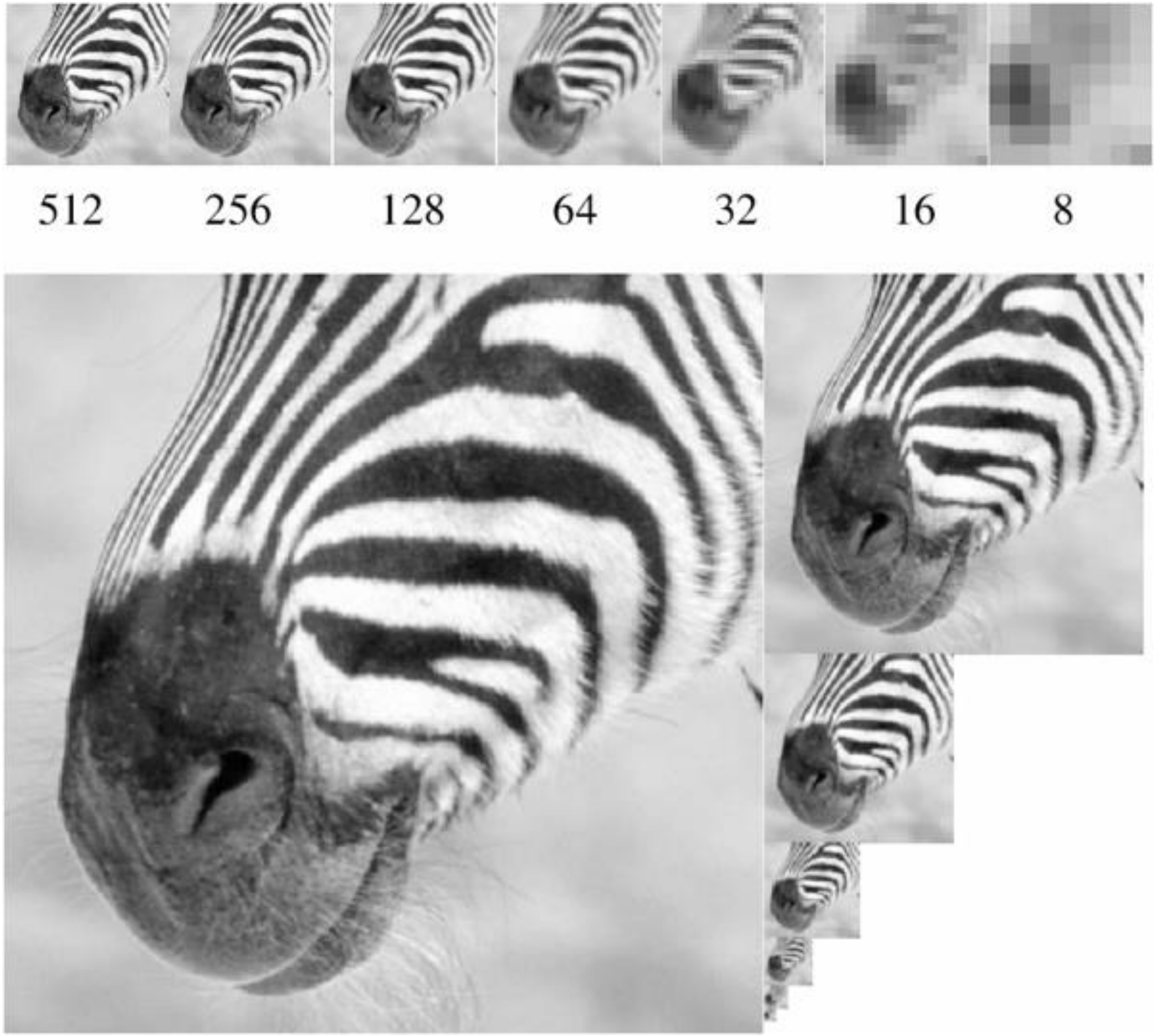


alpha mask

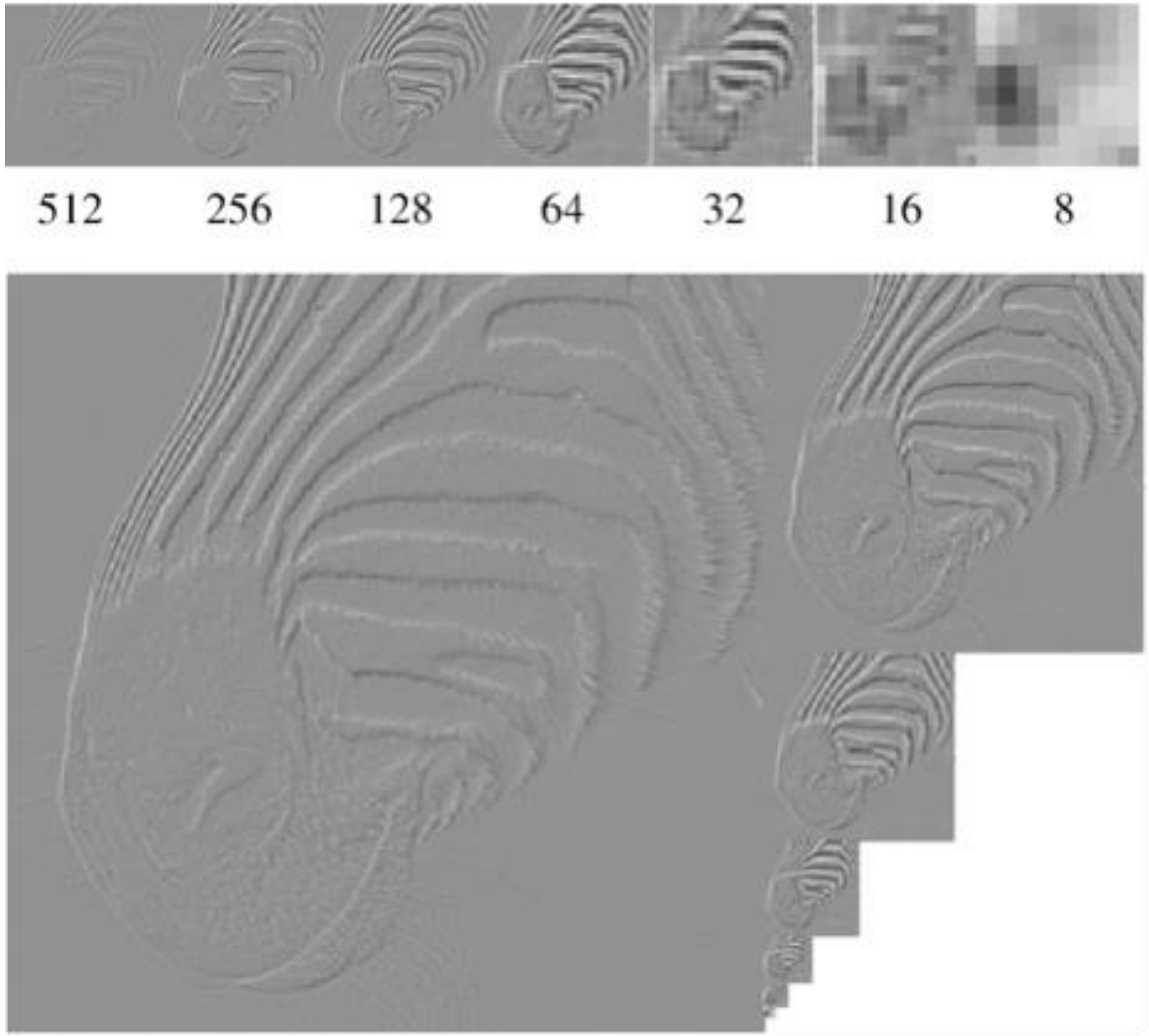


right image

Remember our two types of pyramids



Gaussian pyramid



Laplacian pyramid

Remember our two types of pyramids

1. Build Laplacian pyramids for each image

2. Blend each level of pyramid using region mask

$$L_{12}^{i} = L_{1}^{i} \cdot R^{i} + L_{2}^{i} \cdot (1 - R^{i})$$

image 1 at level i — image 2 at level i — region mask at level i

How large should the blending region be at each level?

3. Collapse the pyramid to get the final blended image

Remember our two types of pyramids

1. Build Laplacian pyramids for each image

2. Blend each level of pyramid using region mask

$$L_{12}^{i} = L_{1}^{i} \cdot R^{i} + L_{2}^{i} \cdot (1 - R^{i})$$

image 1 at level i

image 2 at level i

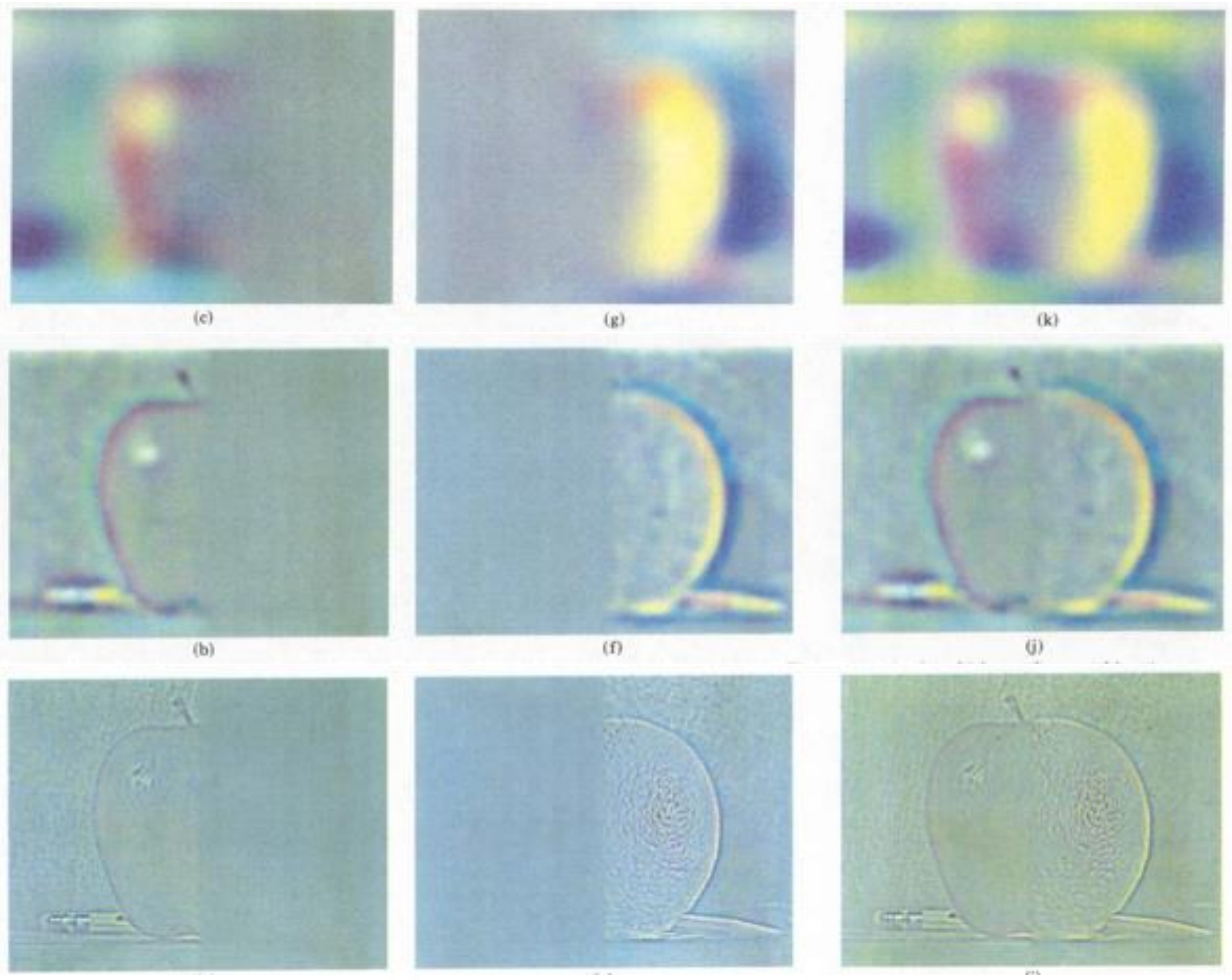
region mask at level i

3. Collapse the pyramid to get the final blended image

How large should the blending region be at each level?

About the size of that level's blur

Multi-band blending using the Laplacian pyramid



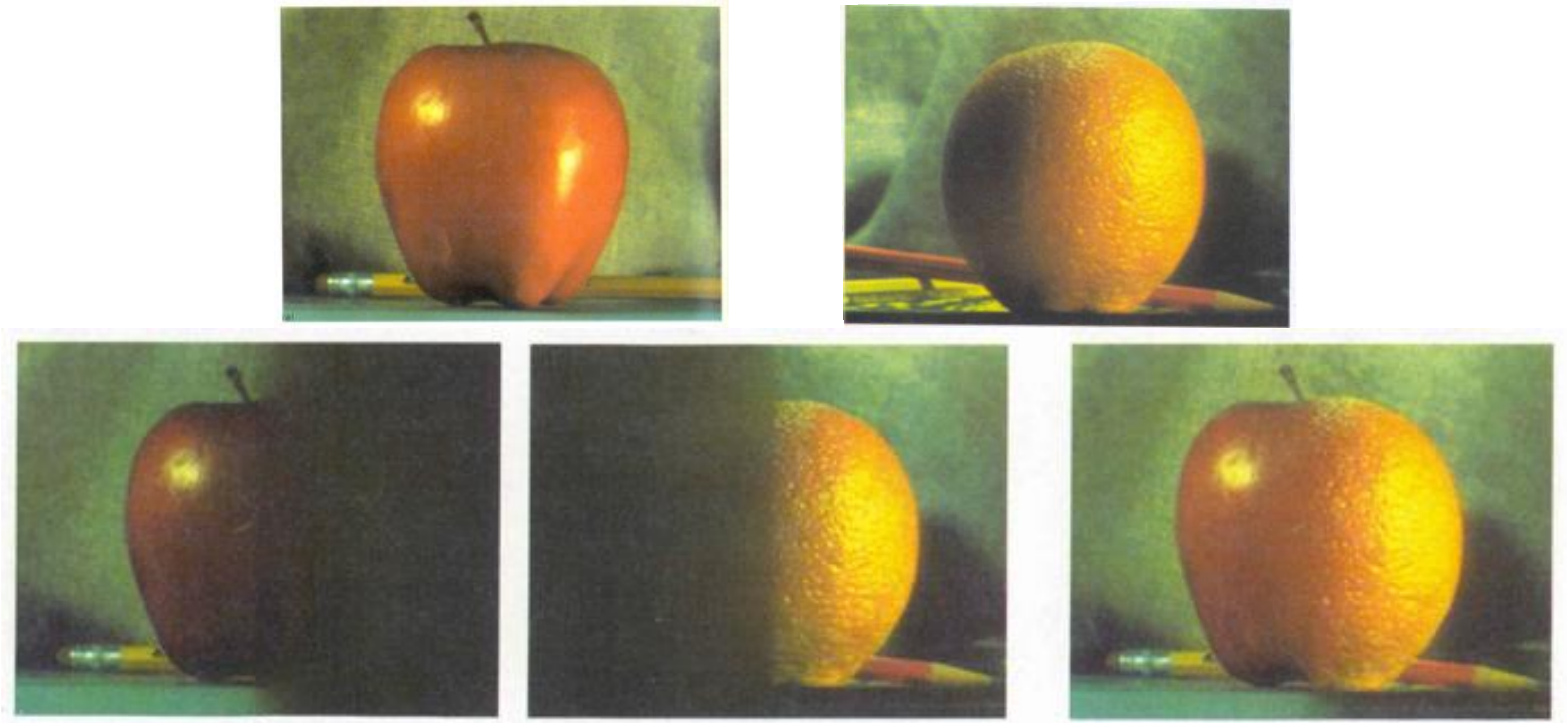
Laplacian level 4

Laplacian level 2

Laplacian level 0

left pyramid right pyramid blended pyramid

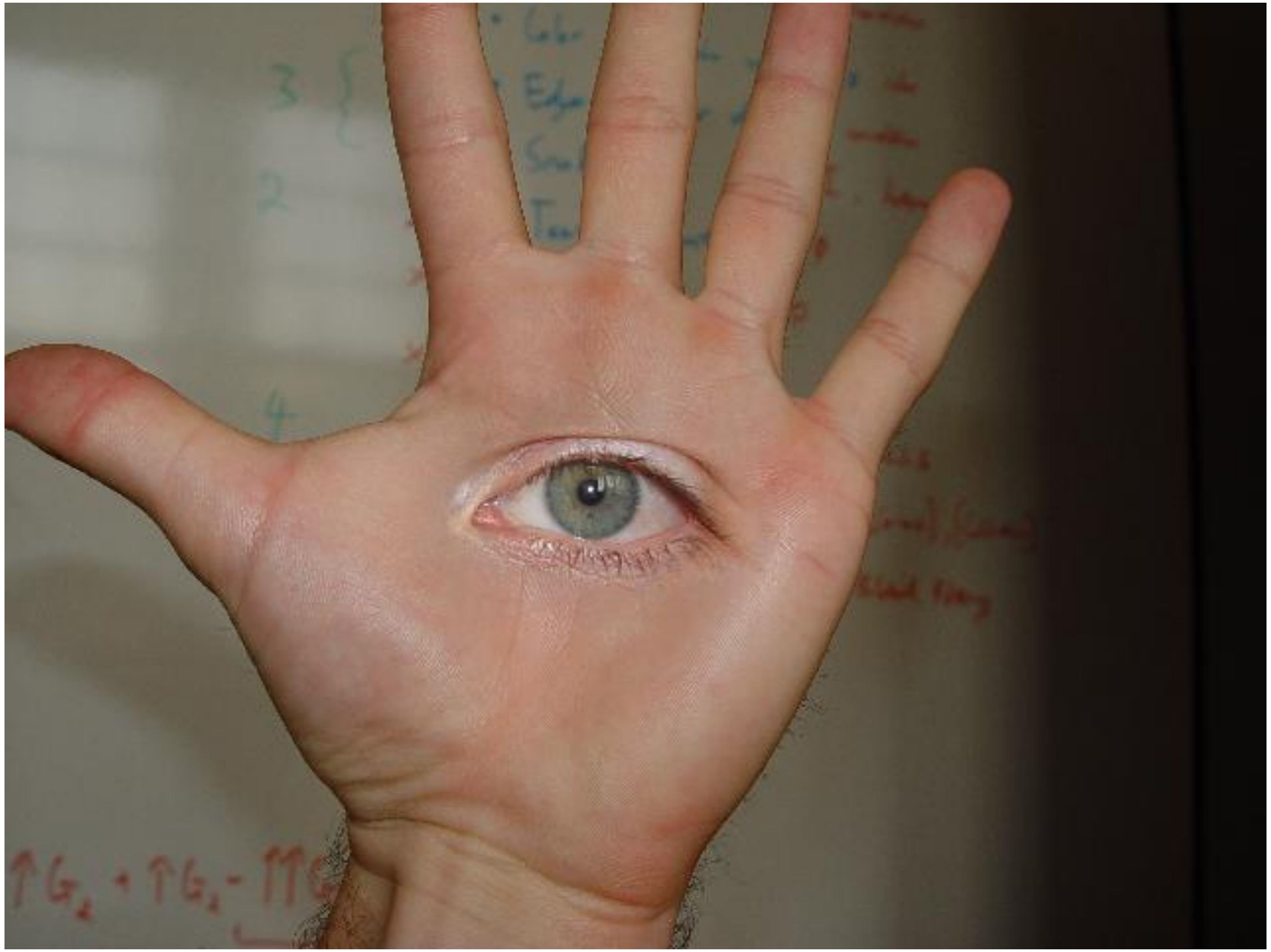
A famous result (for its time)



(d) (h) (l)

A famous result (for its time)

A creepier result

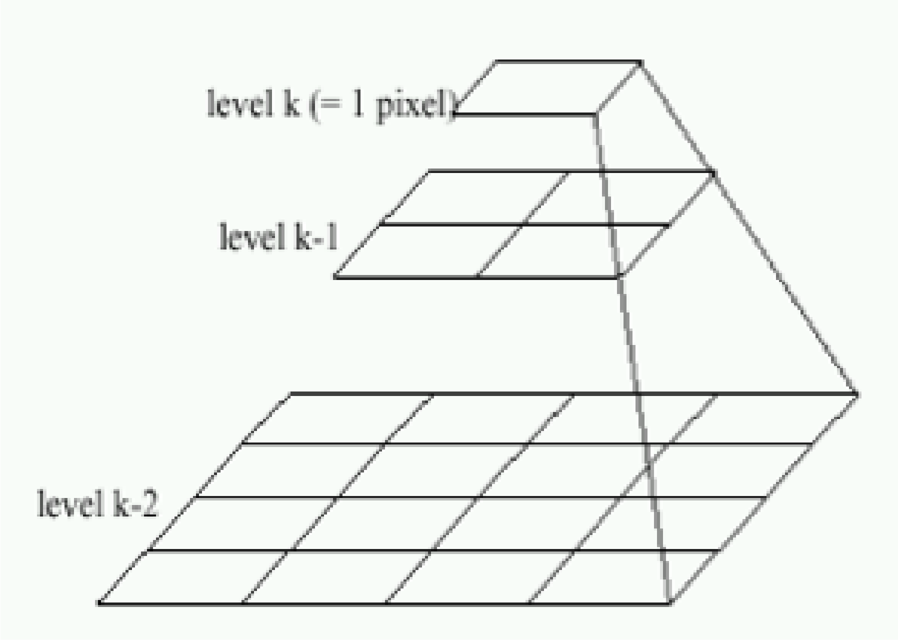


Can we get the same result with less computation?

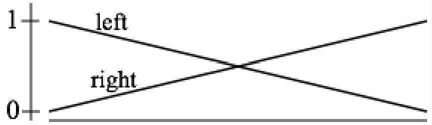
Two-band blending

Only use two bands: high frequency and low frequency

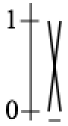
- Blend low frequency with smooth alpha
- Blend high frequency with binary alpha



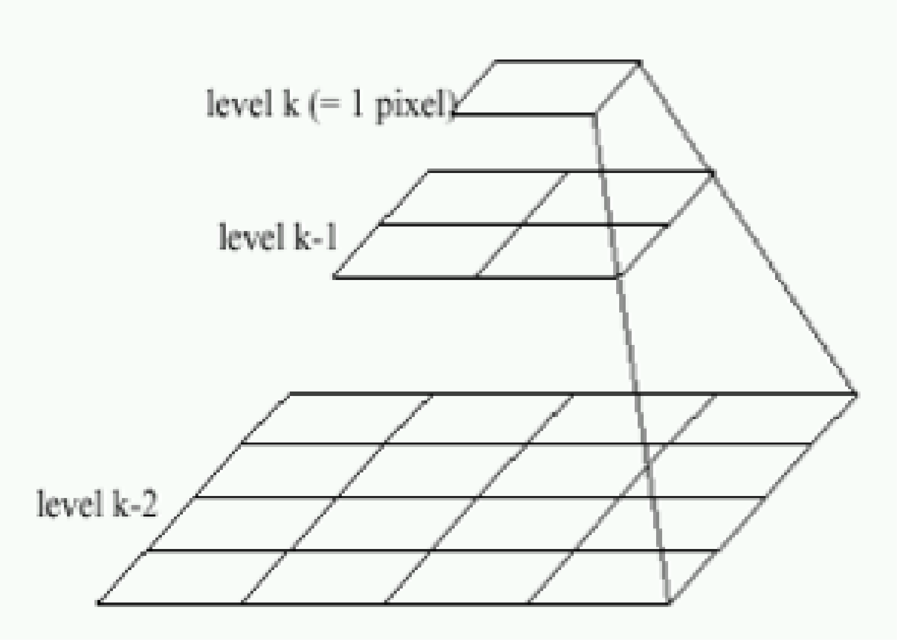
left image



do not use



alpha mask



right image

Example: blending panoramas

original collage

blended collage

Example: blending panoramas

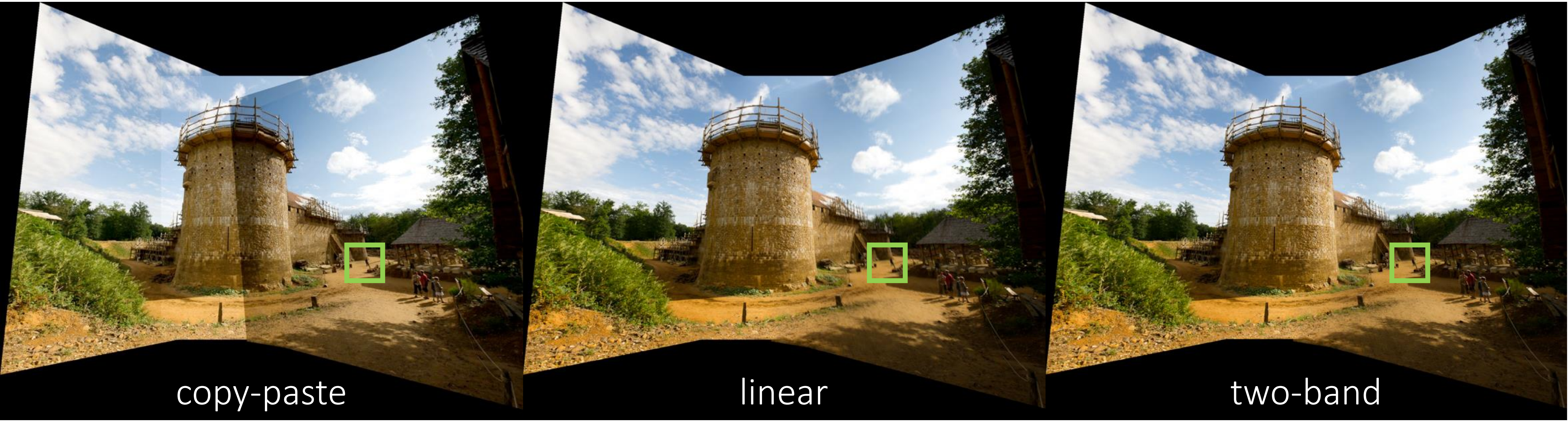
low frequency blend

high frequency blend

Linear blending

Two-band blending

One more comparison



Why do these images look weirdly cropped?

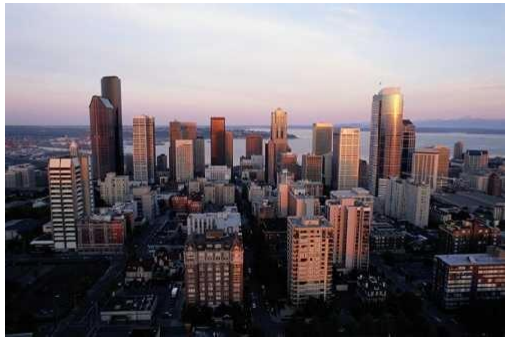
They were warped using homographies before being aligned.

Homework 6: autostitching

Seam stitching

Cut and paste procedure

1. Extract Sprites




How do we do this?

Two different ways to think about the same thing:

- Finding seams (i.e., finding the pixels where to cut an image)
- Segmentation (i.e., splitting the image into “foreground” and “background”)

I will be using the two terms interchangeable

Question about blending

When blending multiple images of the same scene, moving objects become ghosts!

What can we do instead of blending?

Question about blending

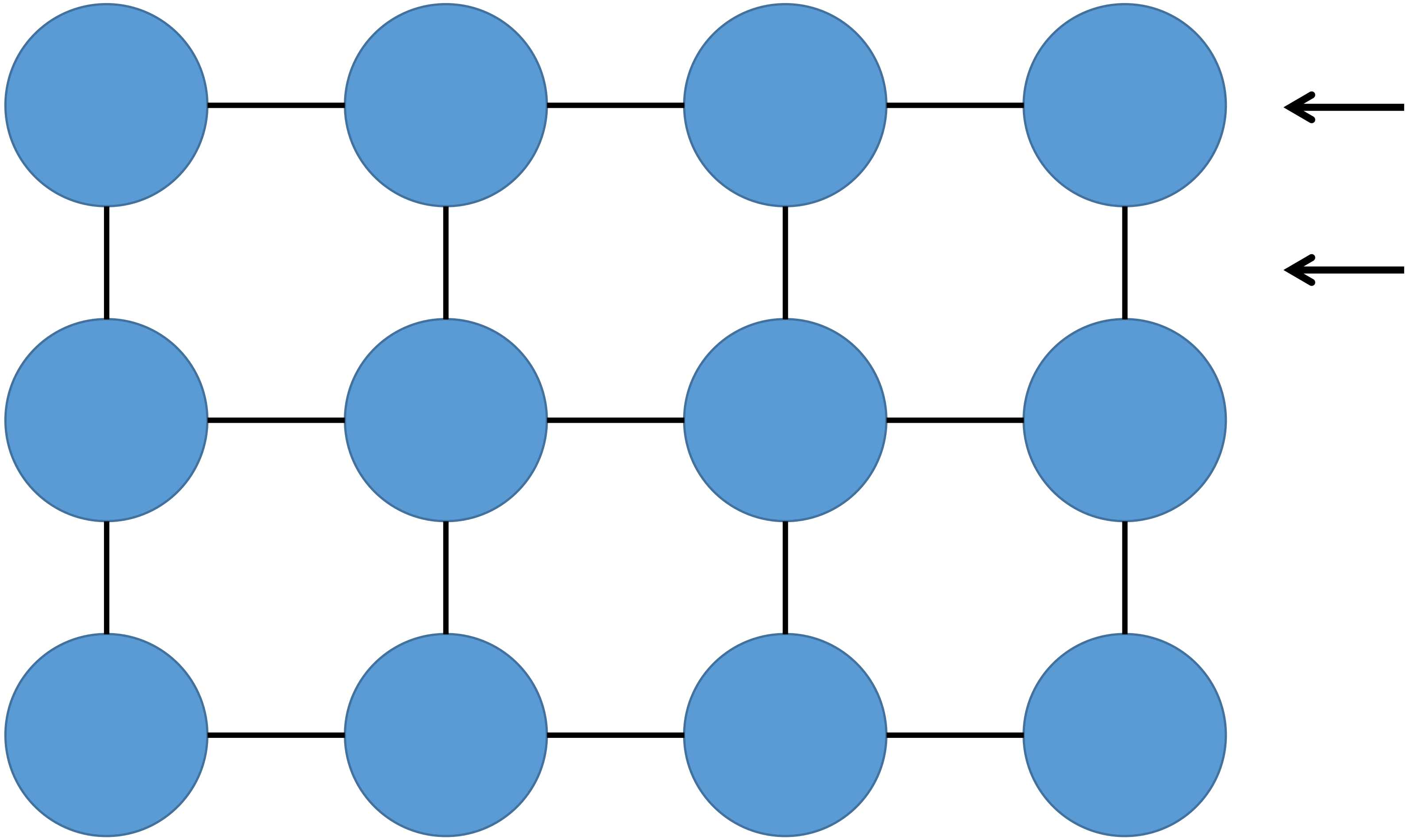
When blending multiple images of the same scene, moving objects become ghosts!

Instead of blending the images, cut them and stitch them together!

Image as a graph

Fundamental theme of today's lecture

Images can be viewed as graphs

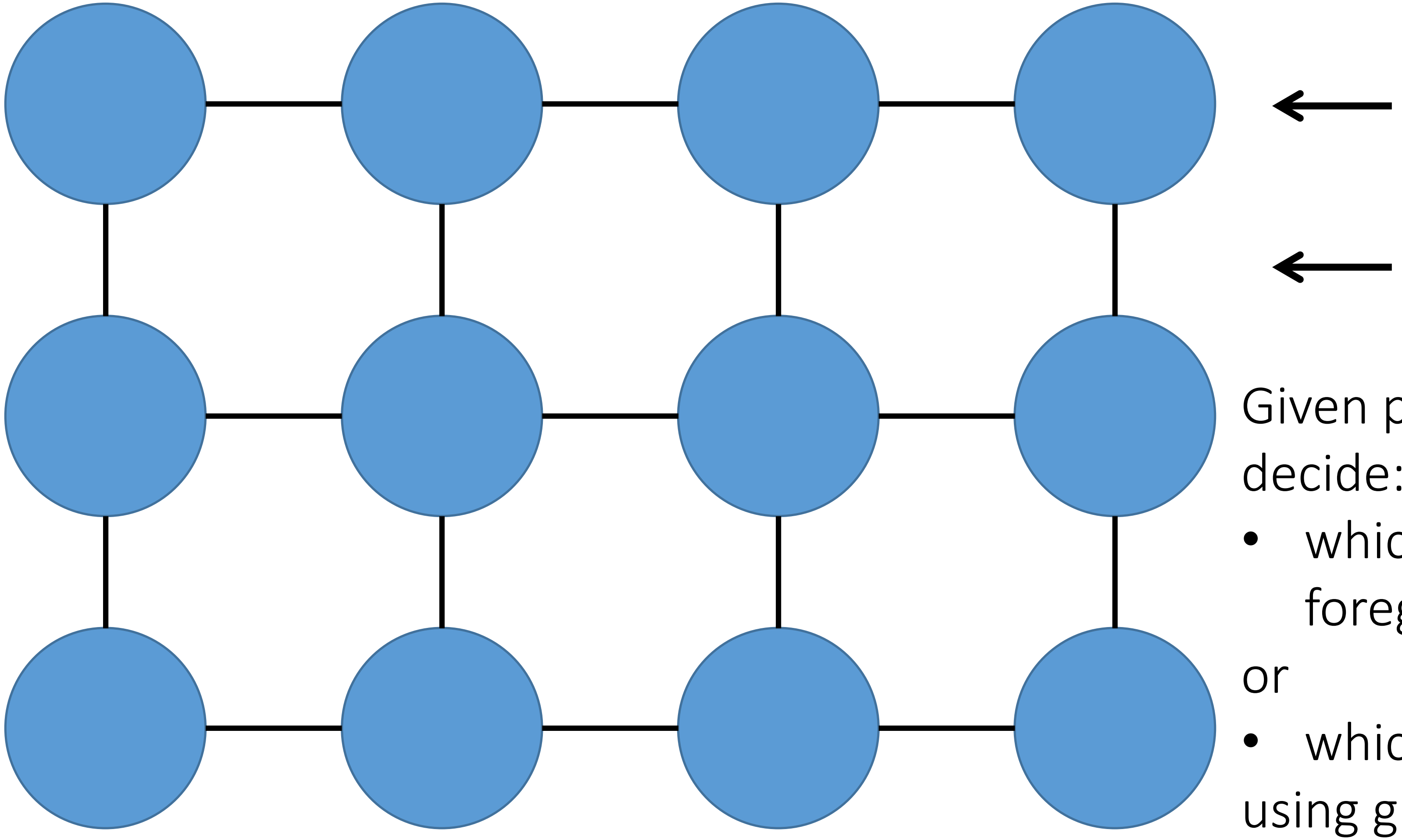


Nodes: pixels

Edges: Constraints between neighboring pixels

Graph-view of segmentation problem

Segmentation is node-labeling



Nodes: pixels

Edges: Constraints between neighboring pixels

Given pixel values and neighborhoods, decide:

- which nodes to label as foreground/background

or

- which nodes to label as seams

using graph algorithms

Graph-view of segmentation problem

Today we will cover:

Method	Labeling problem	Algorithm	Intuition
Intelligent scissors	label pixels as seams	Dijkstra’s shortest path (dynamic programming)	short path is a good **boundary**
GrabCut	label pixels as foreground/background	max-flow/min-cut (graph cutting)	good **region** has low cutting cost

Shortest graph paths and intelligent scissors

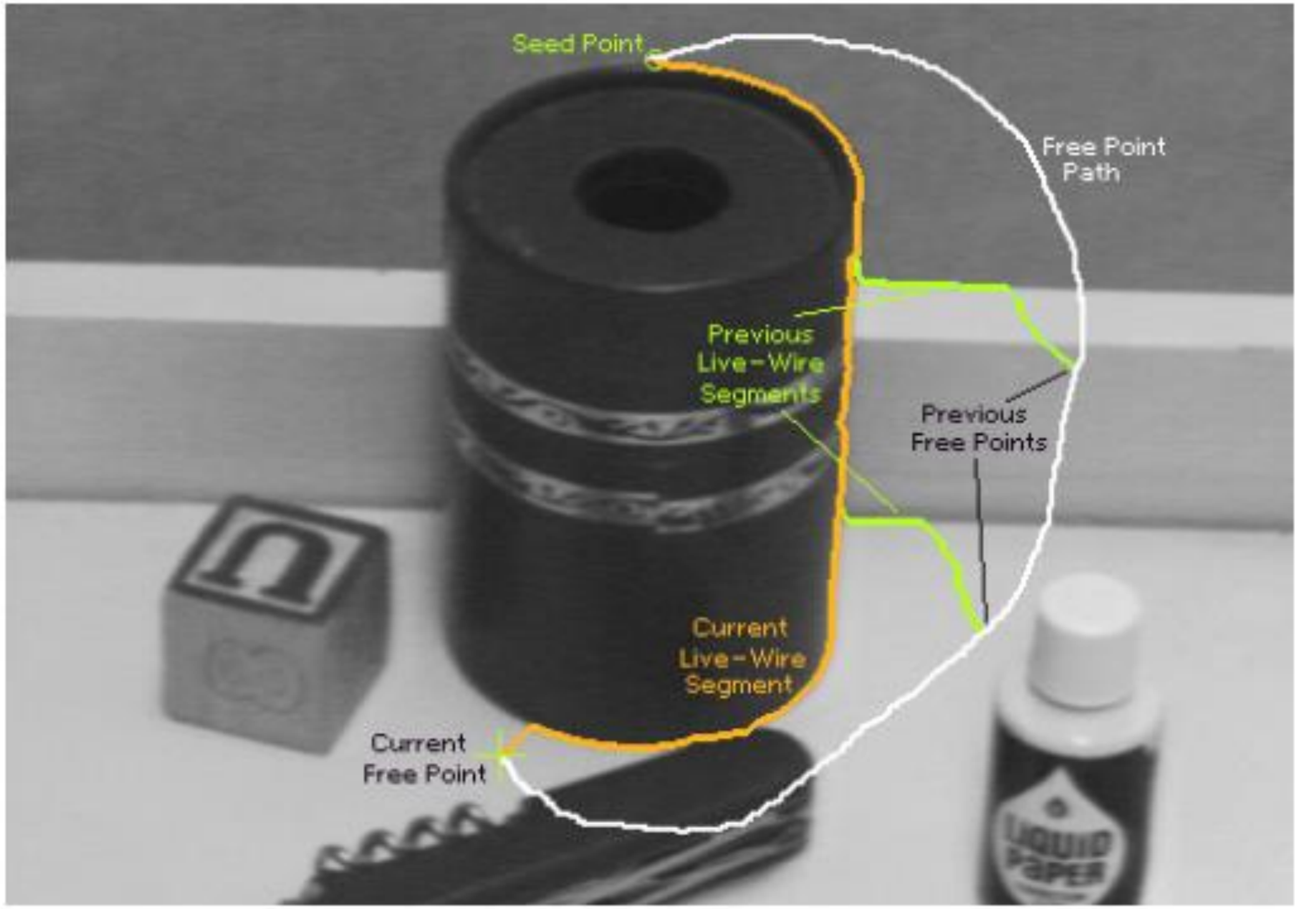
Intelligent scissors

Problem statement:

Given two seed points, find a good boundary connecting them

Challenges:

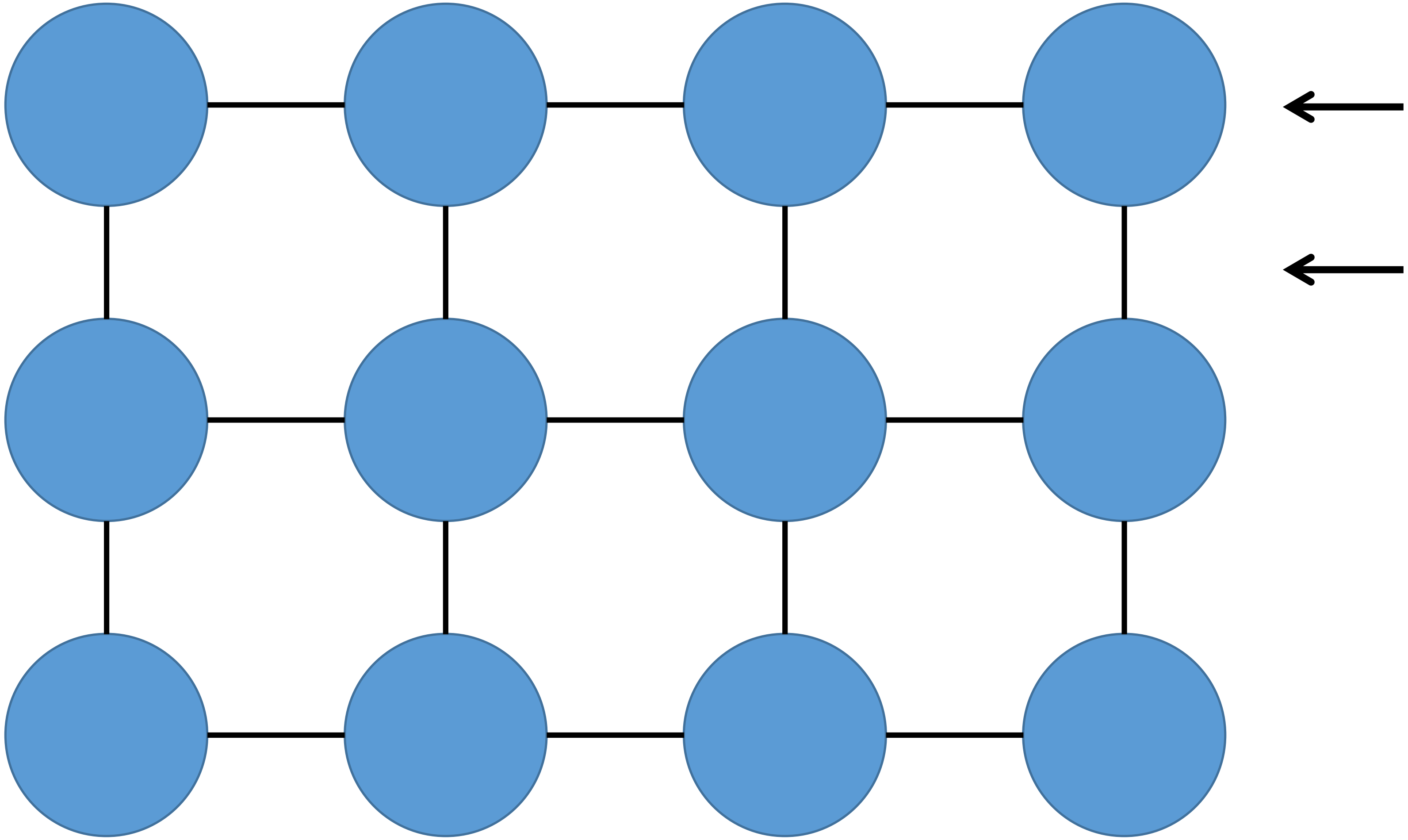
- Make this real-time for interaction
- Define what makes a good boundary



Mortenson and Barrett (SIGGRAPH 1995)
(you can tell it's old from the paper's low quality teaser figure)

Graph-view of this problem

Images can be viewed as graphs

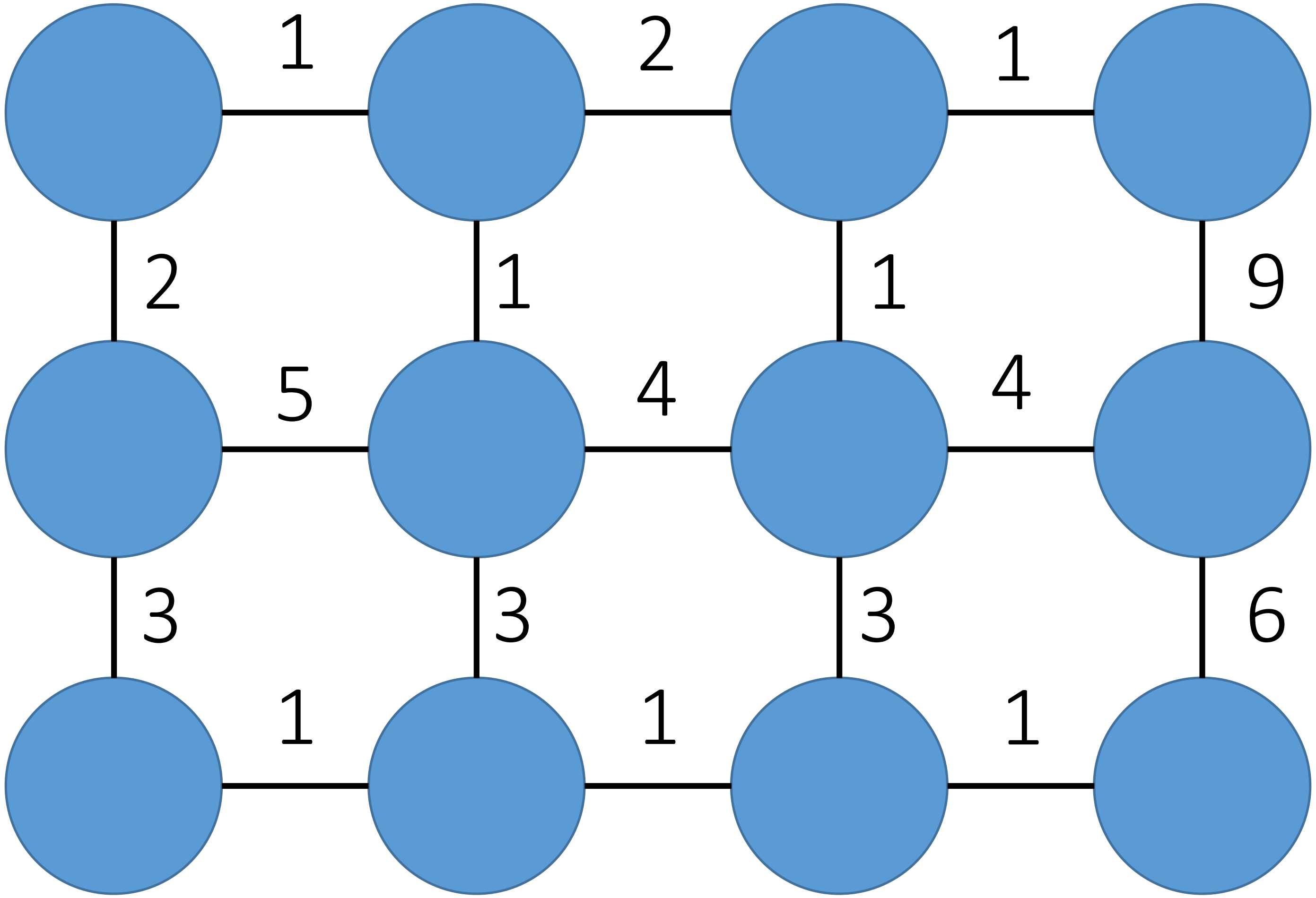


Nodes: pixels

Edges: Constraints between neighboring pixels

Graph-view of this problem

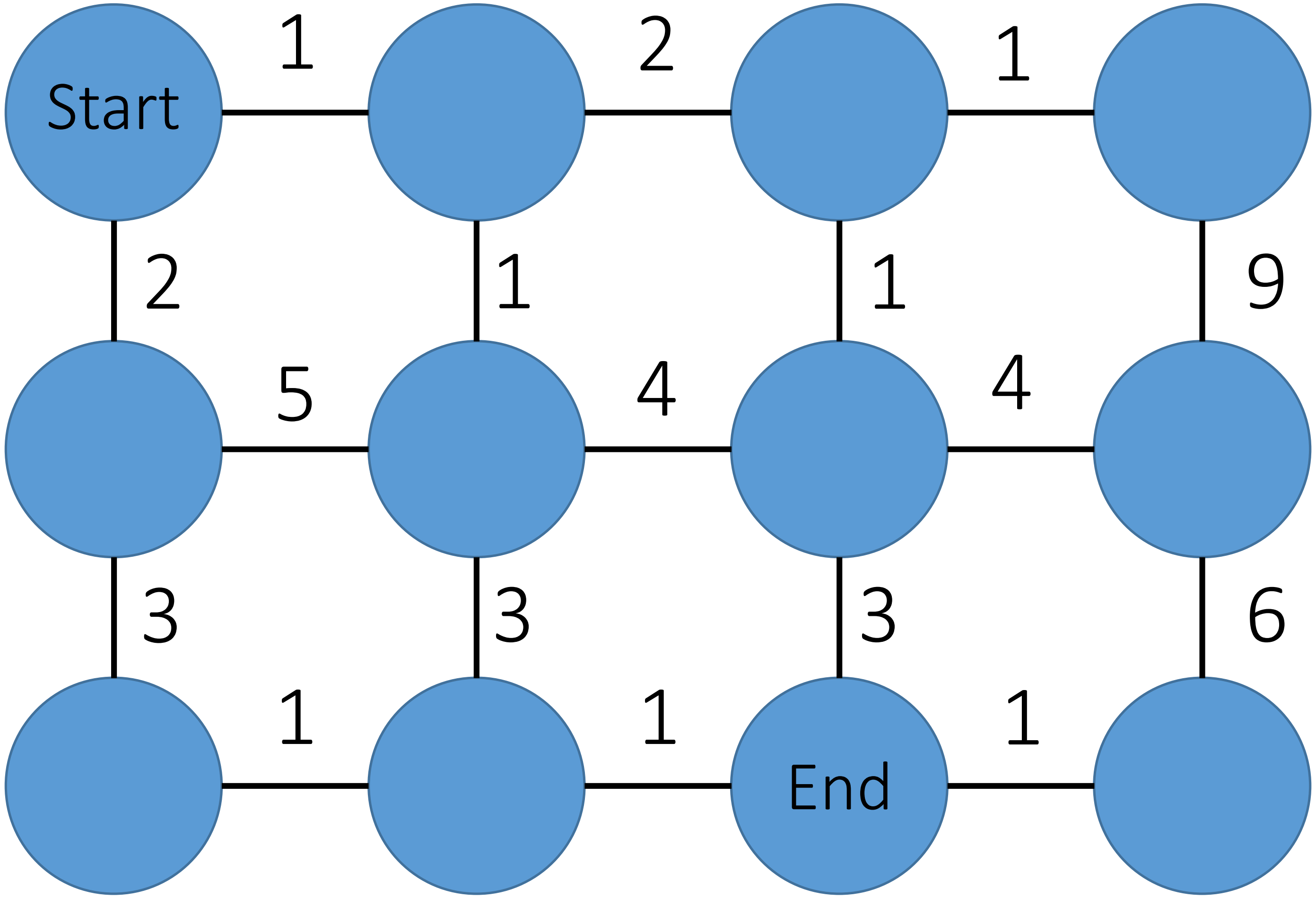
Graph-view of intelligent scissors:



1. Assign weights (costs) to edges

Graph-view of this problem

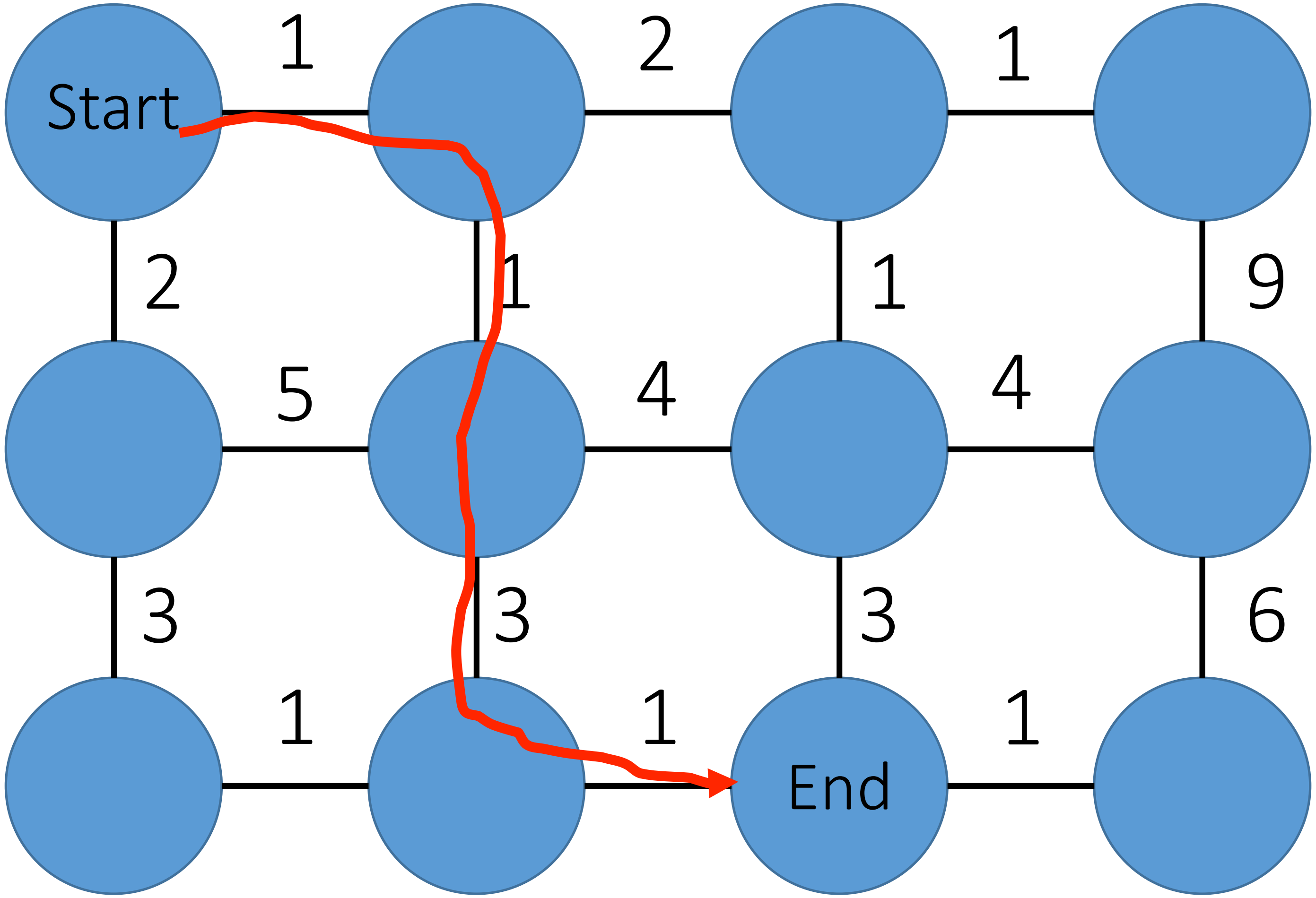
Graph-view of intelligent scissors:



1. Assign weights (costs) to edges
2. Select the seed nodes

Graph-view of this problem

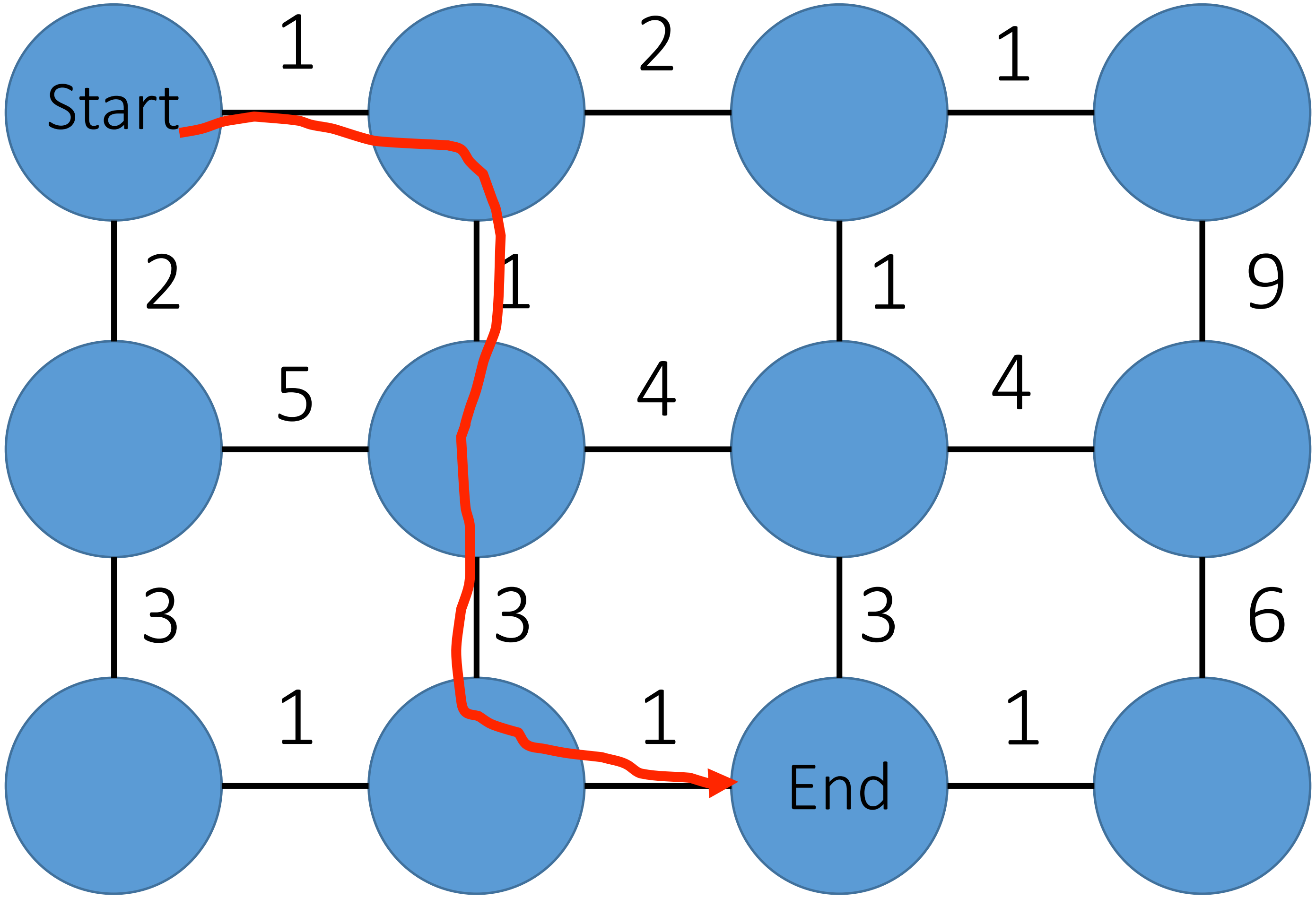
Graph-view of intelligent scissors:



1. Assign weights (costs) to edges
2. Select the seed nodes
3. Find shortest path between them

Graph-view of this problem

Graph-view of intelligent scissors:

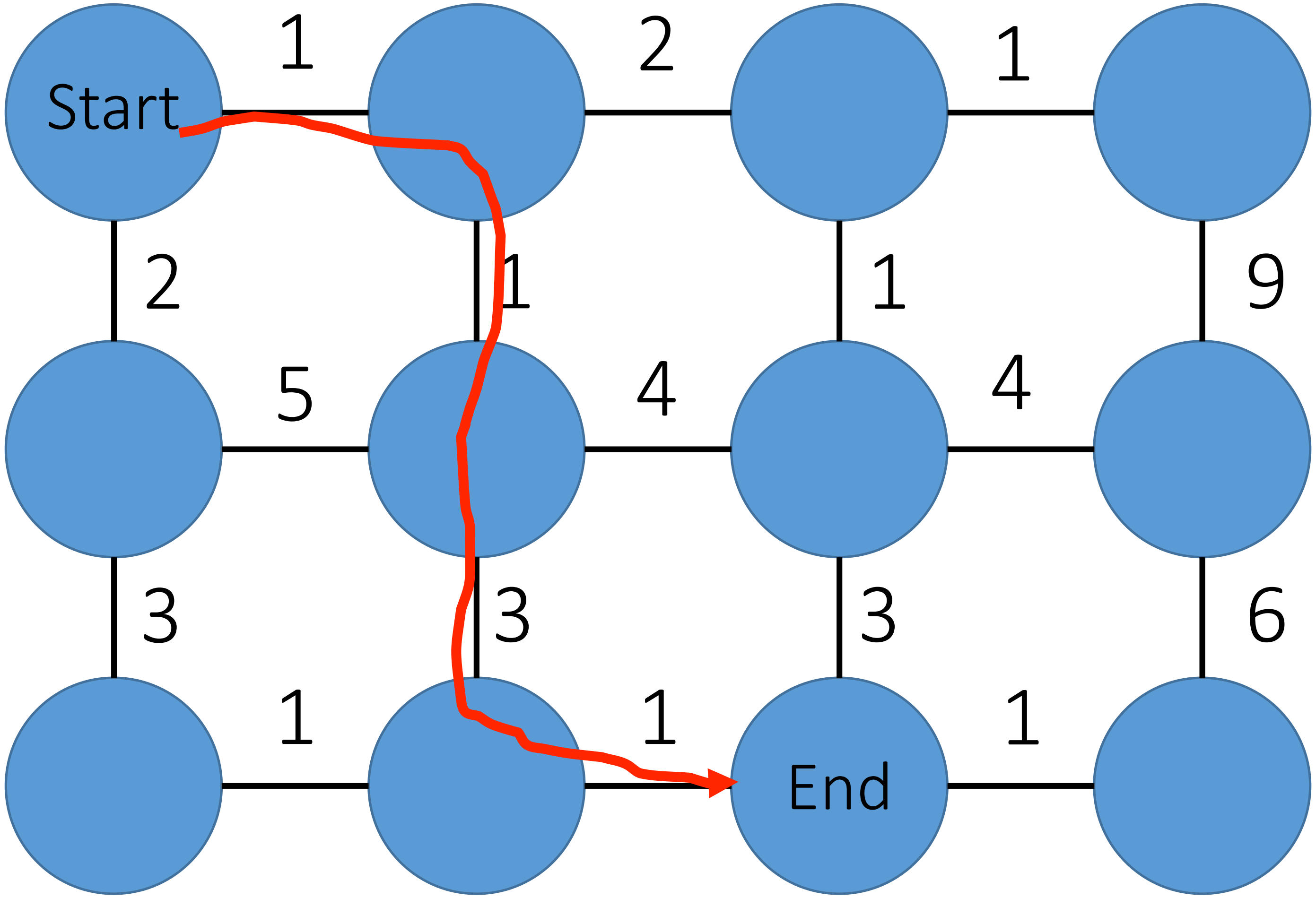


1. Assign weights (costs) to edges
2. Select the seed nodes
3. Find shortest path between them

What algorithm can we use to find the shortest path?

Graph-view of this problem

Graph-view of intelligent scissors:



1. Assign weights (costs) to edges
2. Select the seed nodes
3. Find shortest path between them

What algorithm can we use to find the shortest path?

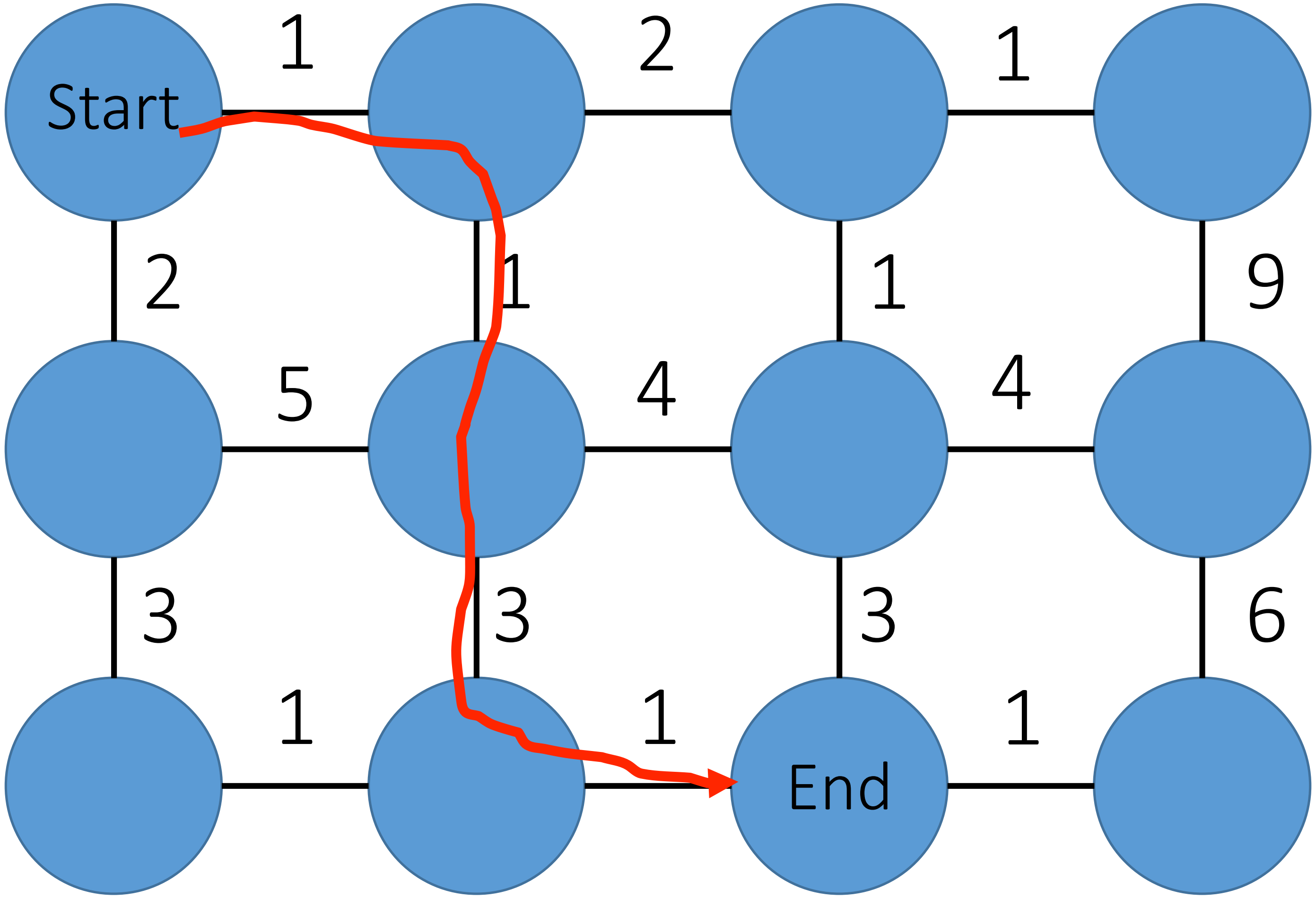
- Dijkstra’s algorithm (dynamic programming)

Dijkstra's shortest path algorithm

```
Initialize, given seed s (pixel ID):
 • cost(s) = 0            % total cost from seed to this point
 • cost(!s) = big
 • A = {all pixels}       % set to be expanded
 • prev(s)=undefined      % pointer to pixel that leads to q=s

Precompute cost2(q, r)  % cost between q to neighboring pixel r

Loop while A is not empty

1.q = pixel in A with lowest cost

2.Remove q from A

3.For each pixel r in neighborhood of q that is in A

  a)cost_tmp = cost(q) + cost2(q,r) %this updates the costs

  b)if (cost_tmp < cost(r))
     i.cost(r) = cost_tmp
     ii. prev(r) = q
```

Graph-view of this problem

Graph-view of intelligent scissors:



1. Assign weights (costs) to edges
2. Select the seed nodes
3. Find shortest path between them

What algorithm can we use to find the shortest path?

- Dijkstra's algorithm (dynamic programming)

How should we select the edge weights to get good boundaries?

Selecting edge weights

Define boundary cost between neighboring pixels:

1. Lower if an image edge is present (e.g., as found by Sobel filtering).

2. Lower if the gradient magnitude at that point is strong.

3. Lower if gradient is similar in boundary direction.

Selecting edge weights

Gradient magnitude

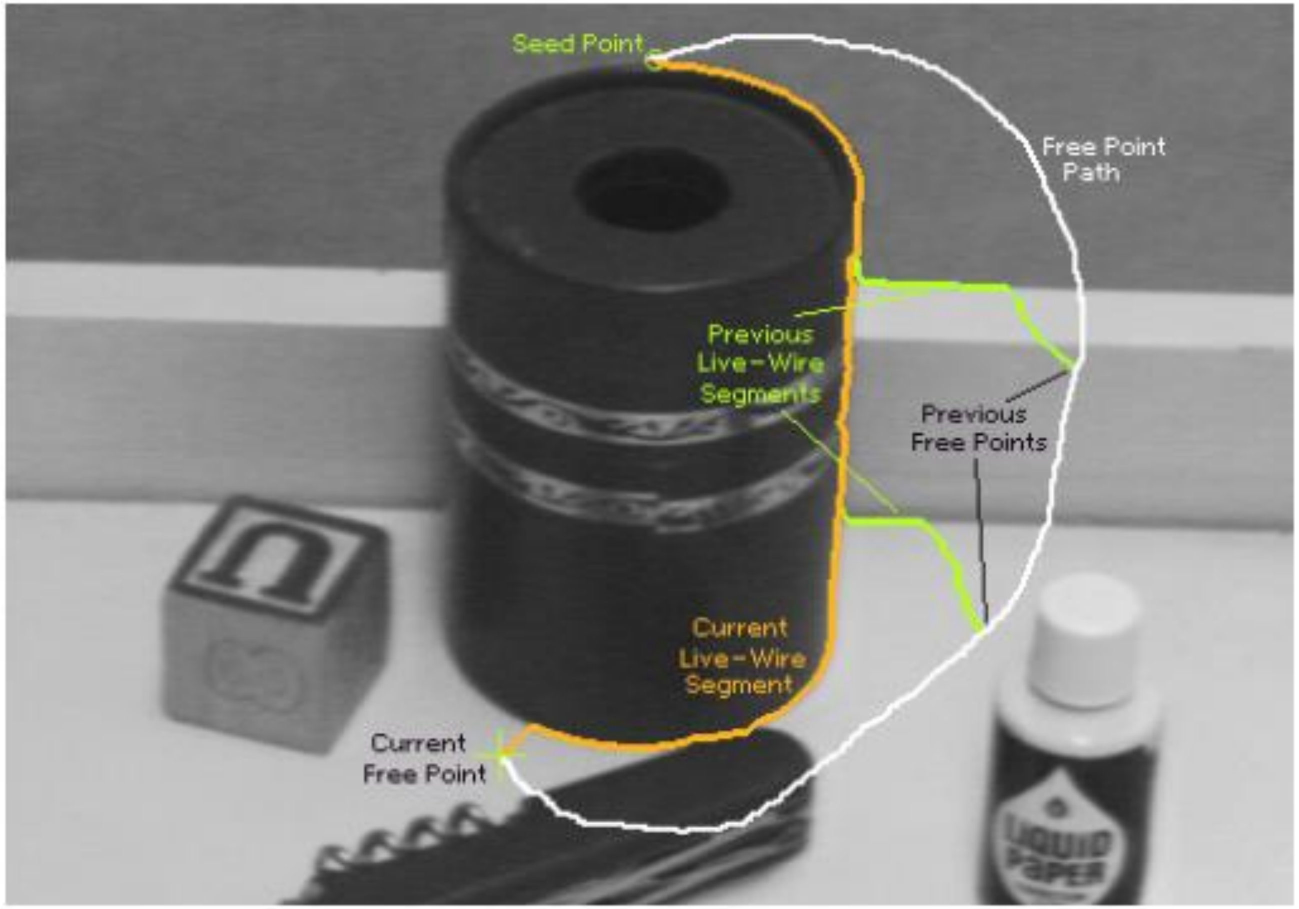
Pixel-wise cost

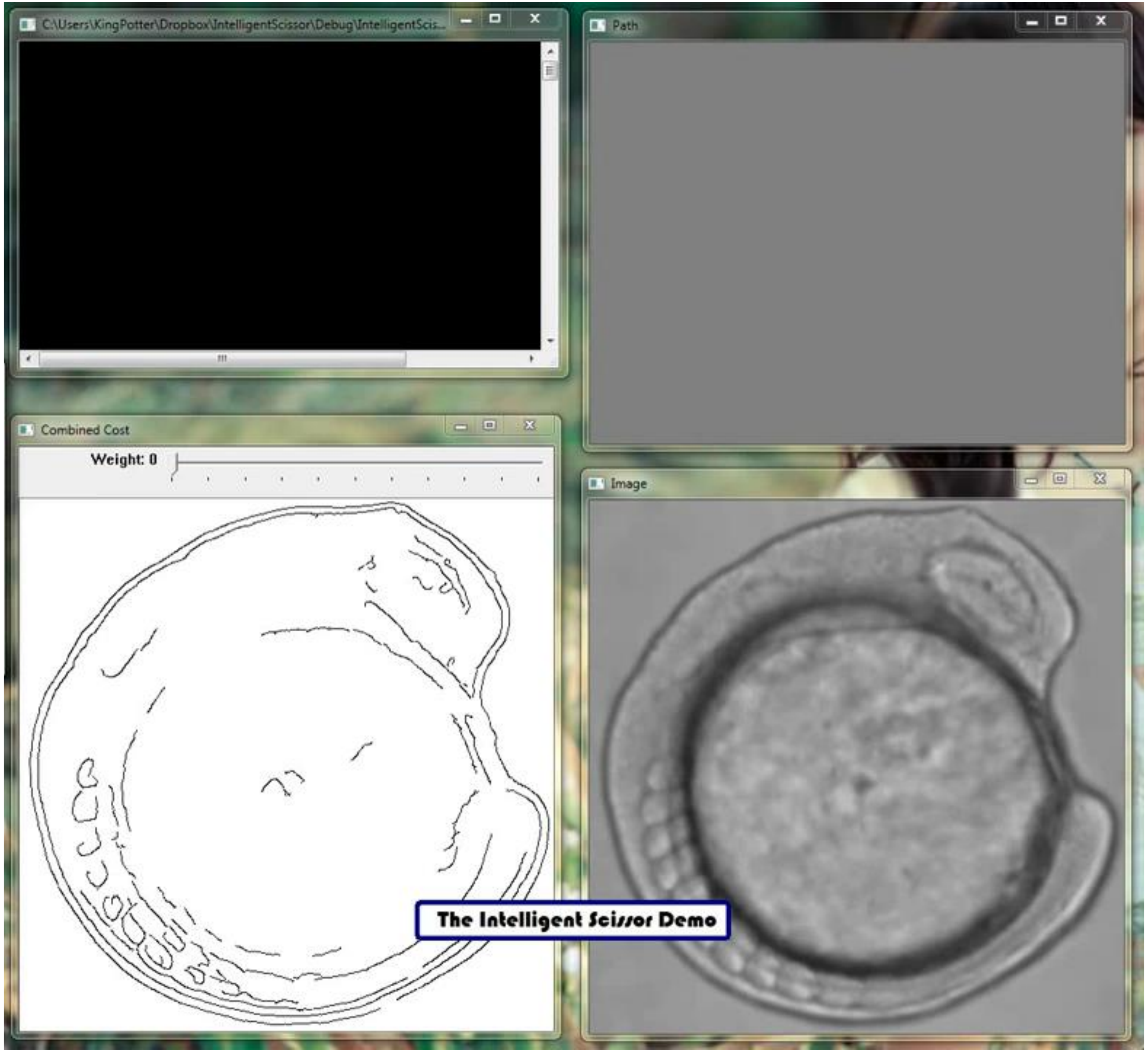
Edge image

Making it more interactive

1. Use cursor as the “end” seed, and always connect start seed to that

2. Every time the user clicks, use that point as a new starting seed and repeat

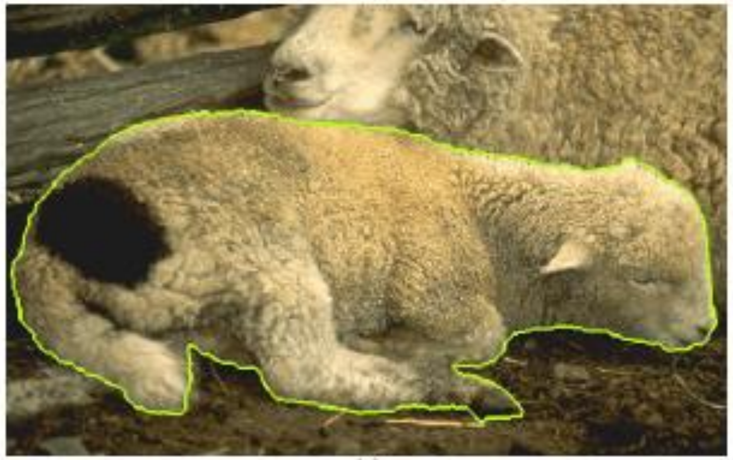



C:\Users\KingPotter\Dropbox\IntelligentScissor\Debug\IntelligentScis...
Path
Combined Cost
Weight: 0
Image
The Intelligent Scissor Demo

Examples

(a)

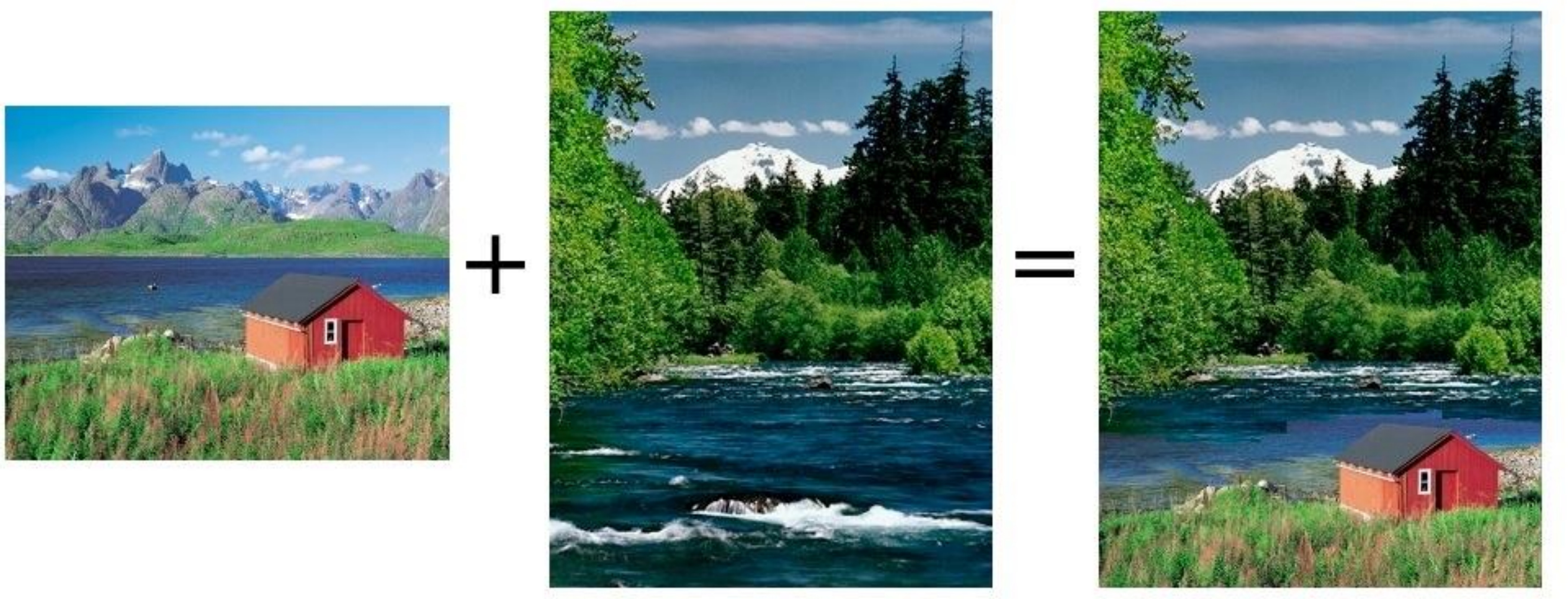
(b)



(c)

Seam collaging

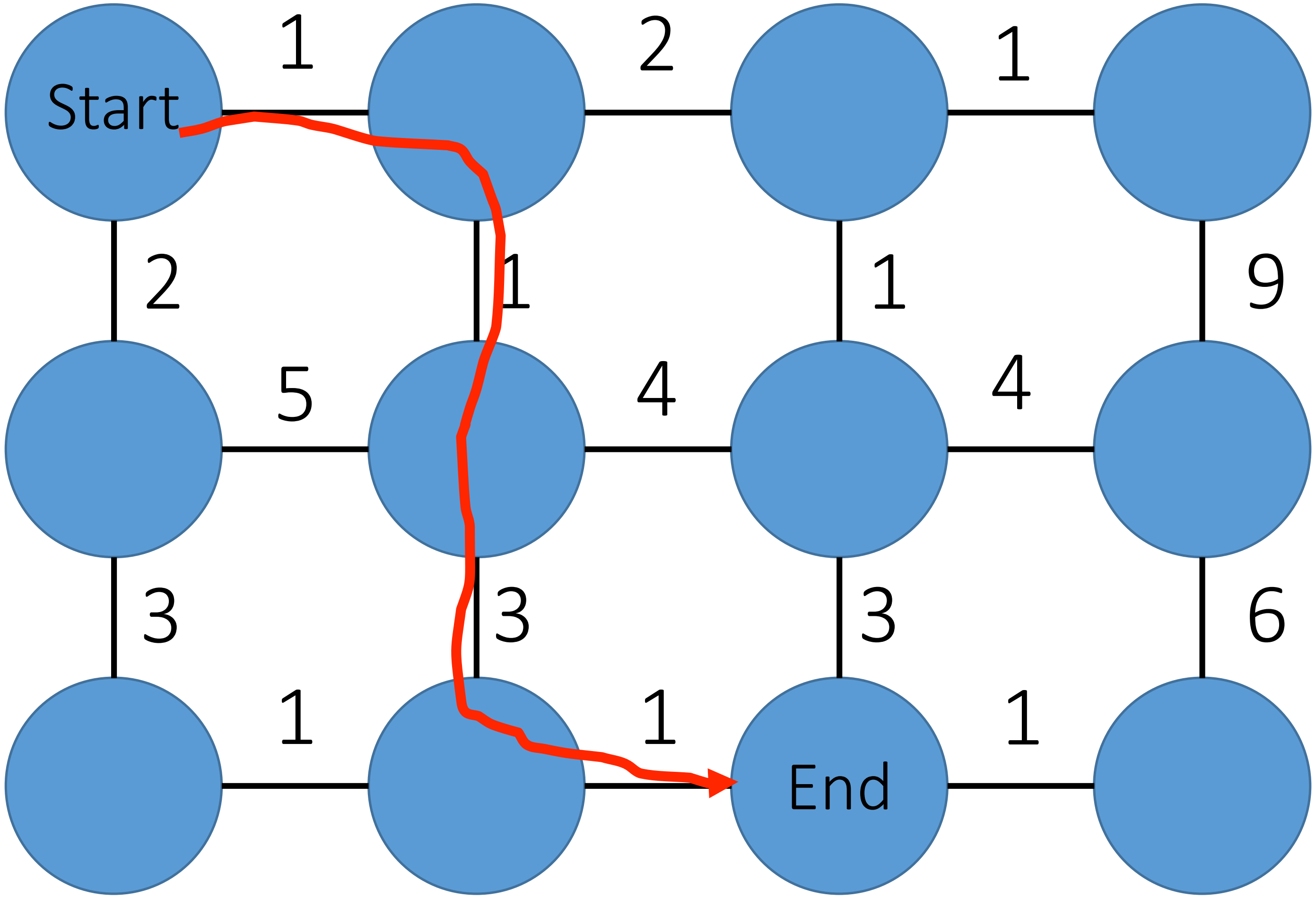
Another use for image seam selection



Kwatra et al., Graphcut Textures: Image and Video Synthesis using Graph Cuts, SIGGRAPH 2003

Graph-view of this problem

Graph-view of image collaging:



1. Assign weights (costs) to edges
2. Select the seed nodes
3. Find shortest path between them

What edge weights would you use for collaging?

Selecting edge weights for seam collaging

Good places to cut:

- similar color in both images
- high gradient in both images

Seam carving

Another use for image seam selection

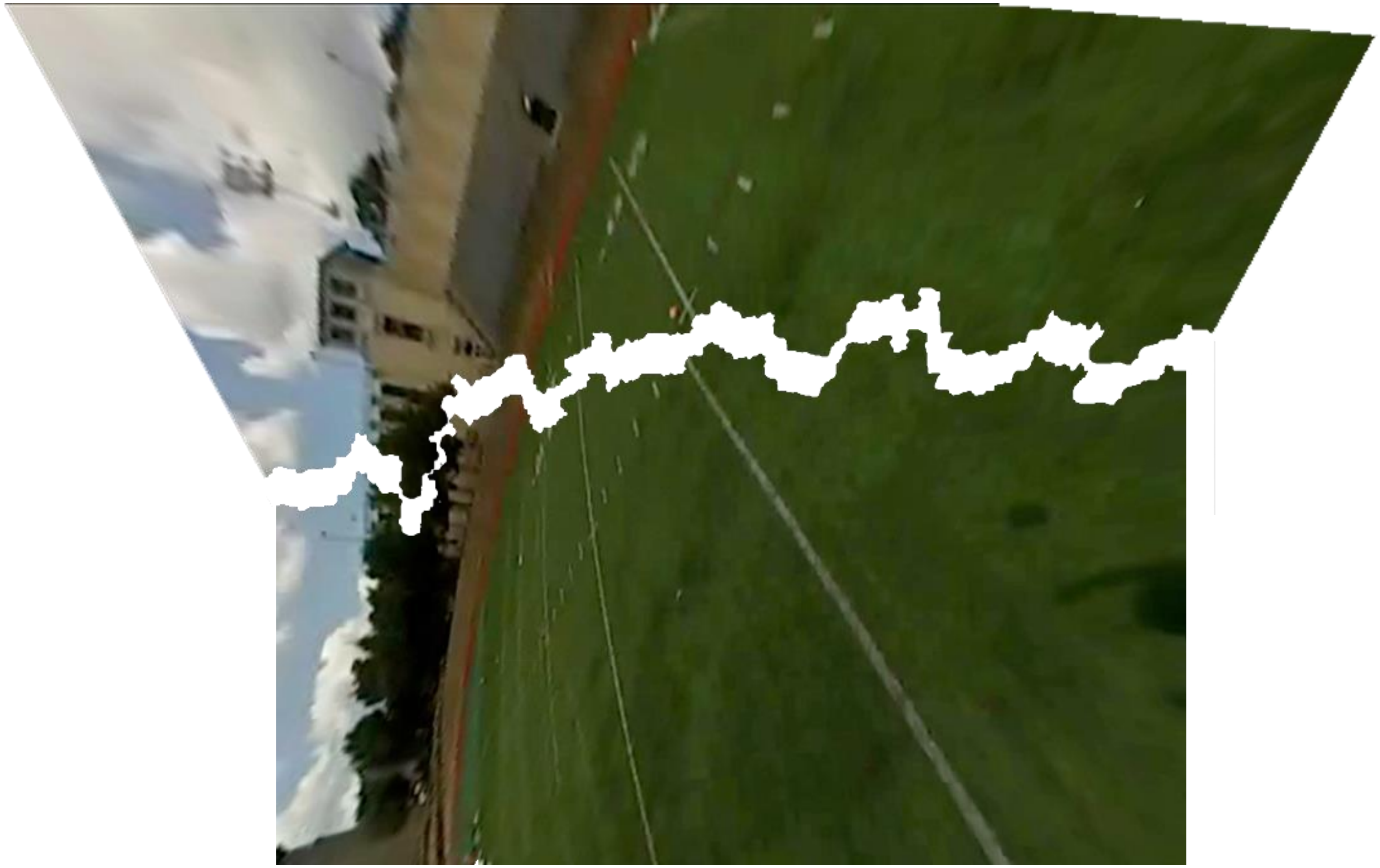
Avidan and Shamir, Seam Carving for Content-Aware Image Resizing, SIGGRAPH 2007

Shai Avidan
Mitsubishi Electric Research Lab
Ariel Shamir
The interdisciplinary Center & MERL

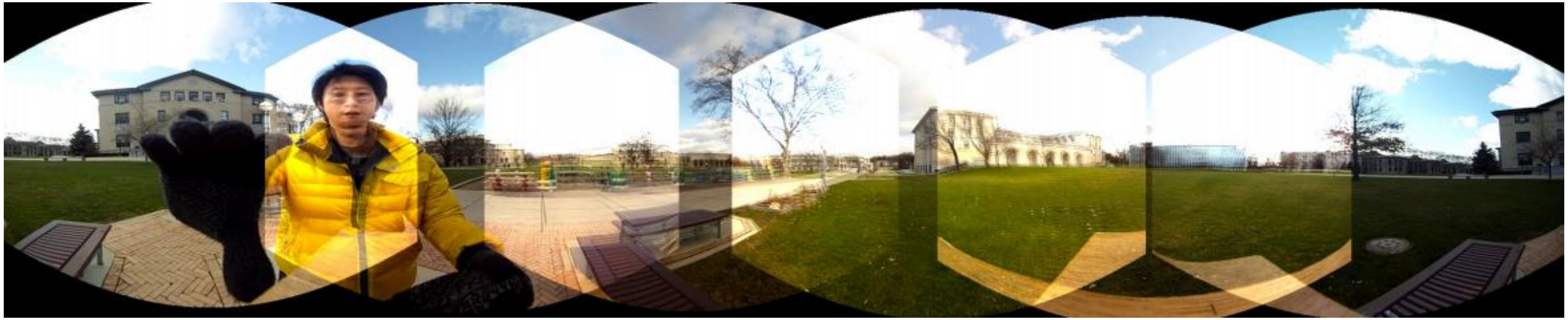
Seam stitching

Another use for image seam selection:

- instead of blending the images, cut them and stitch them together



Seam stitching

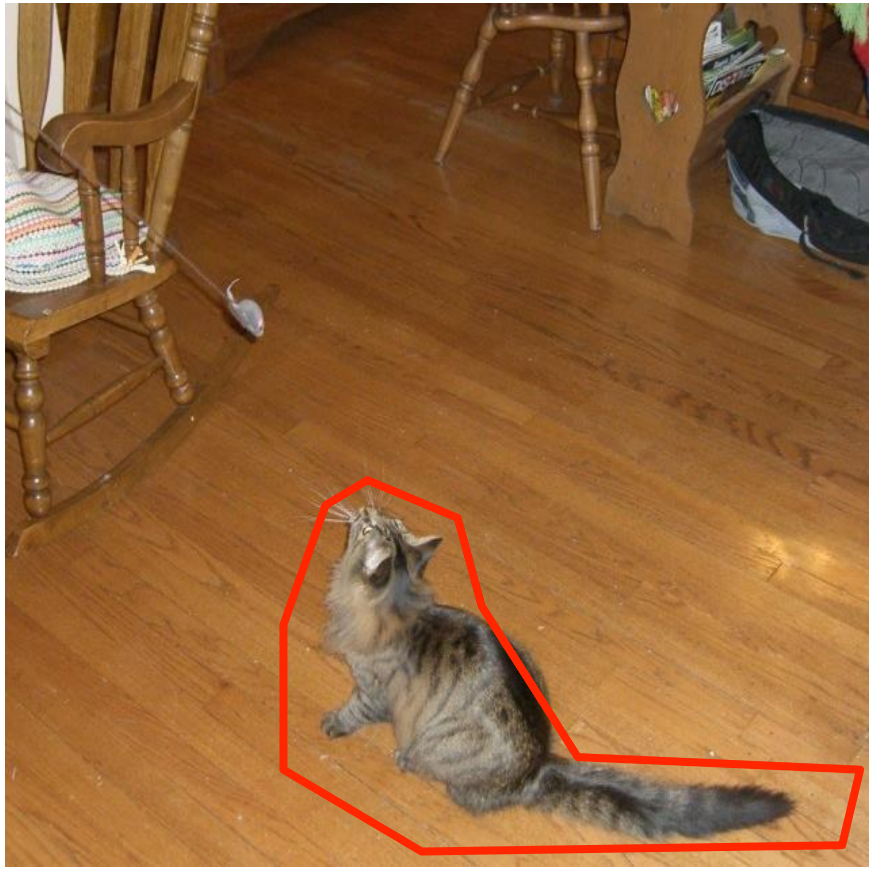


alpha blending

AutoStitch

Examples

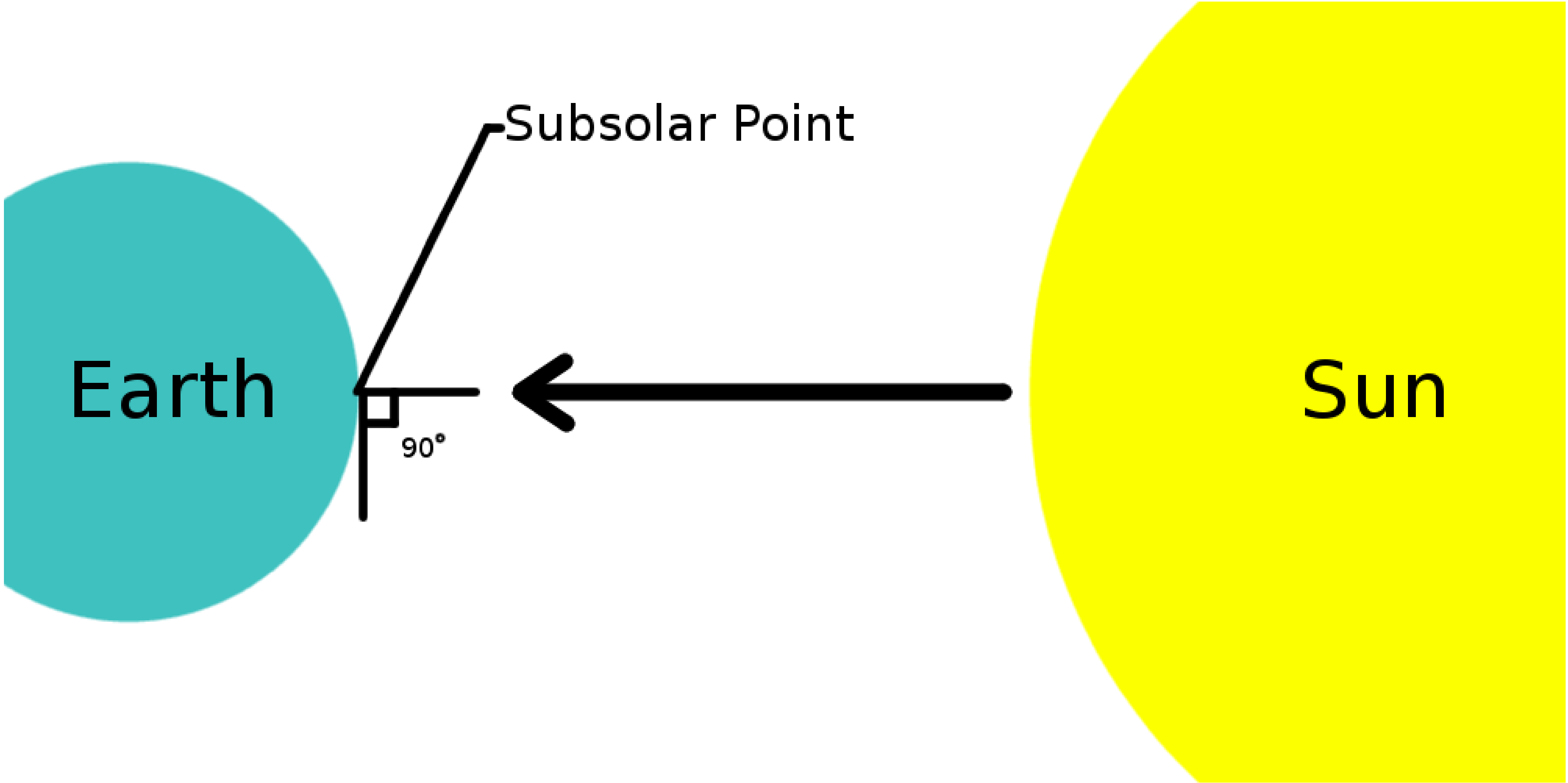
Where will intelligent scissors work well, or have problems?


Some notes about cutting-and-pasting

Real or composite, and why?

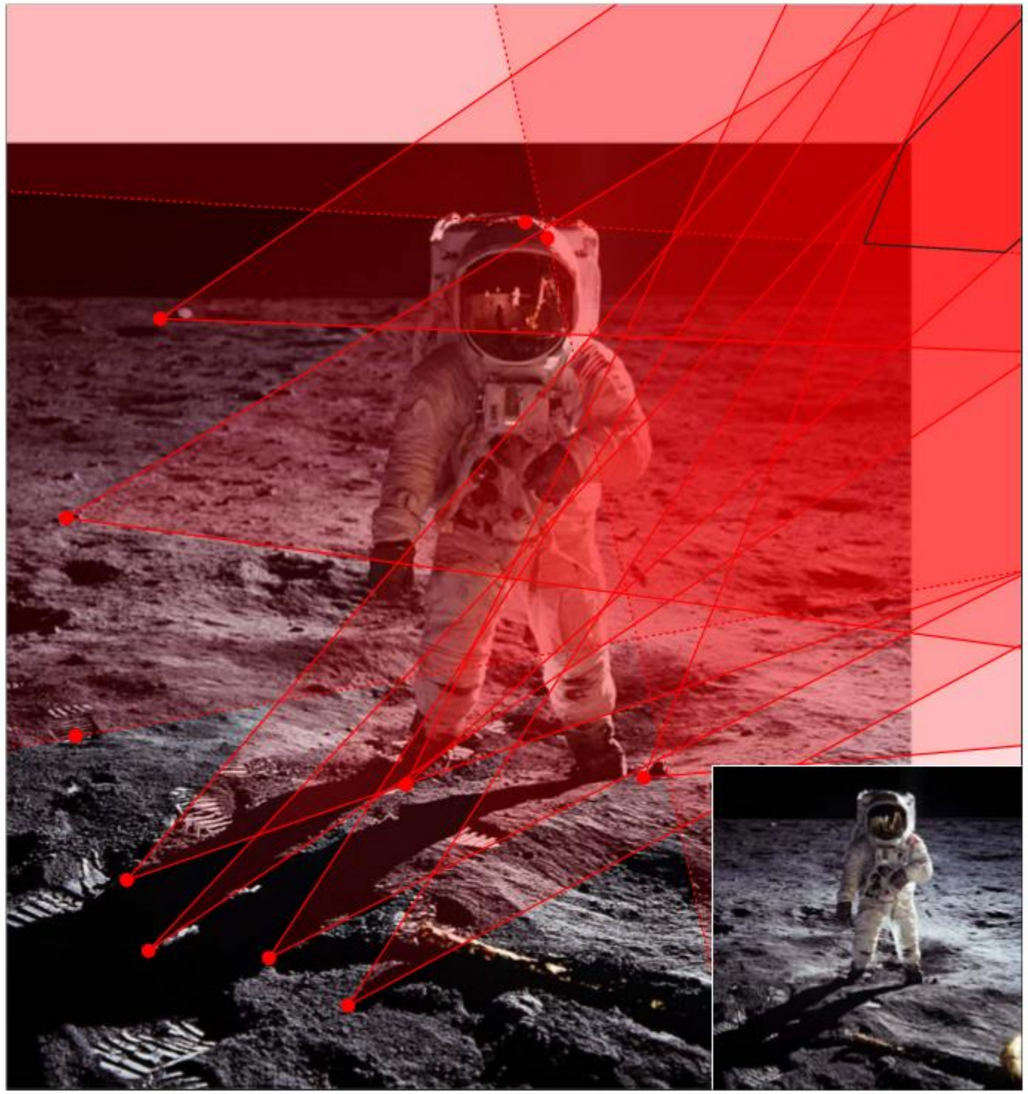
Real: Lahaina noon (or noon at subsolar point)



Real or composite, and why?

Composite: Inconsistent shadows

Composite: Inconsistent shadows



Original image copyright 1969, NASA

Fig. 1. Our algorithm finds that the shadows in this 1969 moon landing photo are physically consistent with a single light source. The solid lines correspond to constraints from cast shadows and dashed lines correspond to constraints from attached shadows. The region outlined in black, which extends beyond the figure boundary, contains the projected light locations that satisfy all of these constraints.

Kee et al., “Exposing Photo Manipulation with Inconsistent Shadows,” ToG 2014

Photorealistic compositing

Karsch et al., "Rendering Synthetic Objects into Legacy Photographs," SIGGRAPH Asia 2011

Photorealistic compositing

Karsch et al., “Rendering Synthetic Objects into Legacy Photographs,” SIGGRAPH Asia 2011

Photorealistic compositing

Karsch et al., “Rendering Synthetic Objects into Legacy Photographs,” SIGGRAPH Asia 2011

Photorealistic compositing

Karsch et al., “Rendering Synthetic Objects into Legacy Photographs,” SIGGRAPH Asia 2011

Photorealistic compositing

Karsch et al., “Rendering Synthetic Objects into Legacy Photographs,” SIGGRAPH Asia 2011

Photorealistic compositing

Karsch et al., “Rendering Synthetic Objects into Legacy Photographs,” SIGGRAPH Asia 2011

Photorealistic compositing

How would you do this?

References

Basic reading:
- Szeliski textbook, Sections 3.13, 3.5.5, 9.3.4, 10.4.3.

Additional reading:
- Pérez et al., "Poisson Image Editing," SIGGRAPH 2003.
    the original Poisson blending paper.
- Georgiev, "Covariant Derivatives and Vision," ECCV 2006.
    a paper from Adobe describing the version of Poisson blending implemented in Photoshop's "healing brush".
- Elder and Goldberg, "Image editing in the contour domain", PAMI 2001.
- Bhat et al., "GradientShop: A Gradient-Domain Optimization Framework for Image and Video Filtering," ToG 2010.
- Agrawal and Raskar, "Gradient Domain Manipulation Techniques in Vision and Graphics," ICCV 2007 course, http://www.amitkagrawal.com/ICCV2007Course/
    the above references provide an overview of gradient-domain processing as a general image processing paradigm, which can be used for a broad set of applications beyond blending, including tone-mapping, colorization, converting to grayscale, edge enhancement, image abstraction and non-photorealistic rendering.

References

Basic reading:

- Szeliski textbook, Sections 3.13, 3.5.5, 5.1.3, 5.3.1, 9.3.2, 9.3.3, 9.3.4, 10.4.3.
- Mortensen and Barrett, "Intelligent scissors for image composition," SIGGRAPH 1995.
  The intelligent scissors paper.
- Kwatra et al., Graphcut Textures: Image and Video Synthesis using Graph Cuts, SIGGRAPH 2003.
  The seam collaging paper.
- Avidan and Shamir, "Seam Carving for Content-aware Image Resizing," SIGGRAPH 2007.
  The seam carving paper.

Additional reading:

- Felzenszwalb and Zabih, "Dynamic Programming and Graph Algorithms in Computer Vision," PAMI 2010.
  A great review of graph-based techniques, including shortest path and graph-cut, in computer vision.
- Kee et al., "Exposing photo manipulation with inconsistent shadows," ToG 2013.
  The paper demonstrating how image forgeries can be detecting by reasoning about the physical accuracy of shadows in an image.
- Karsch et al., "Rendering synthetic objects into legacy photographs", SIGGRAPH 2011.
  The paper where the photorealistic compositing examples came from.