CS6670: Computer Vision

Noah Snavely

Lecture 7: Panoramas and stitching

What's inside your fridge?
http://www.cs.washington.edu/education/courses/cse590ss/01wi/

Announcements

- Project 1 due tonight by 11:59pm
  - Upload to CMS. If you haven't been added to CMS, let me know right away
- Next week: guest lecturer, Prof. Pedro Felzenszwalb, U. Chicago

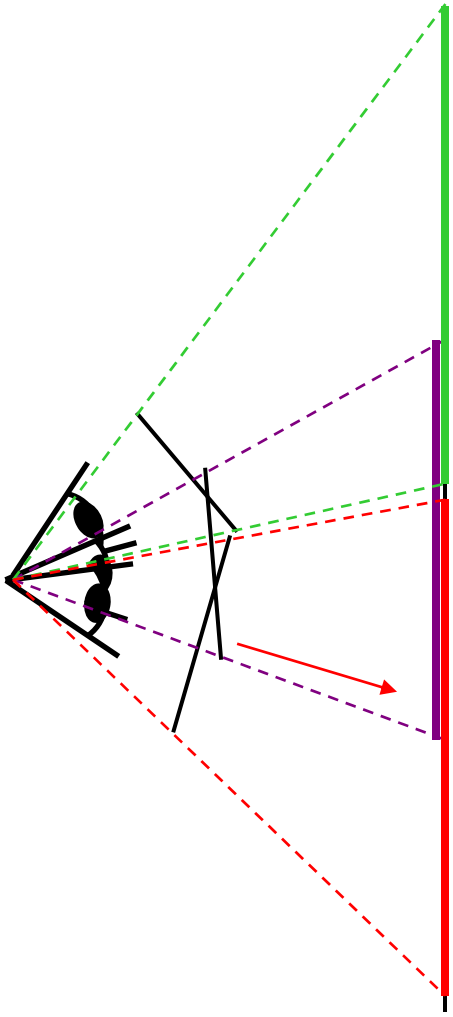
Announcements

- Project 2 will be released on Tuesday
- You can work in groups of two
  - Send me your groups by Friday evening
- Adarsh will demo the project sometime next week (TBA)

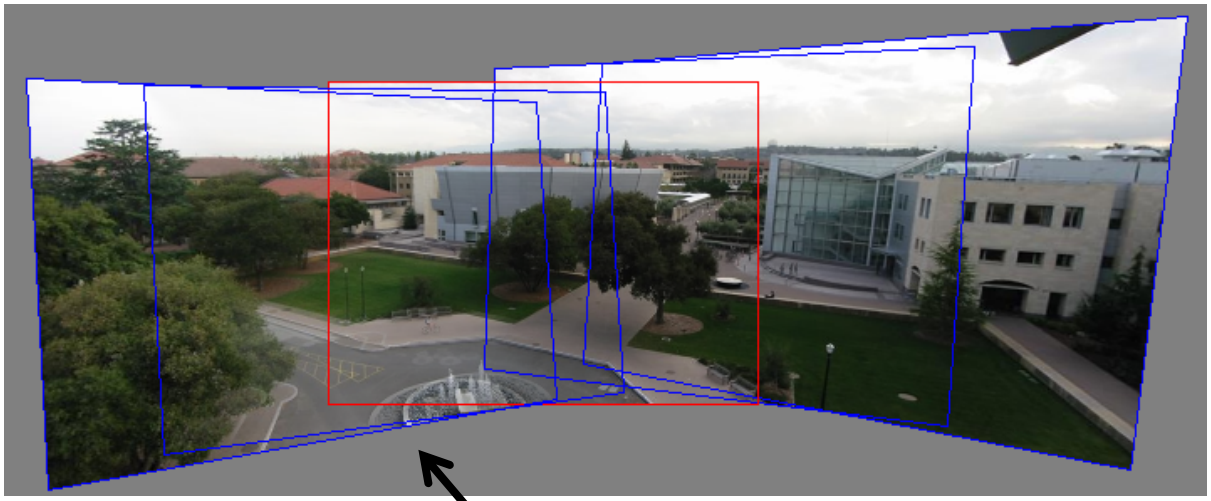
Readings

- Szeliski, Chapter 9.3

Last time: projecting images onto a common plane



mosaic PP

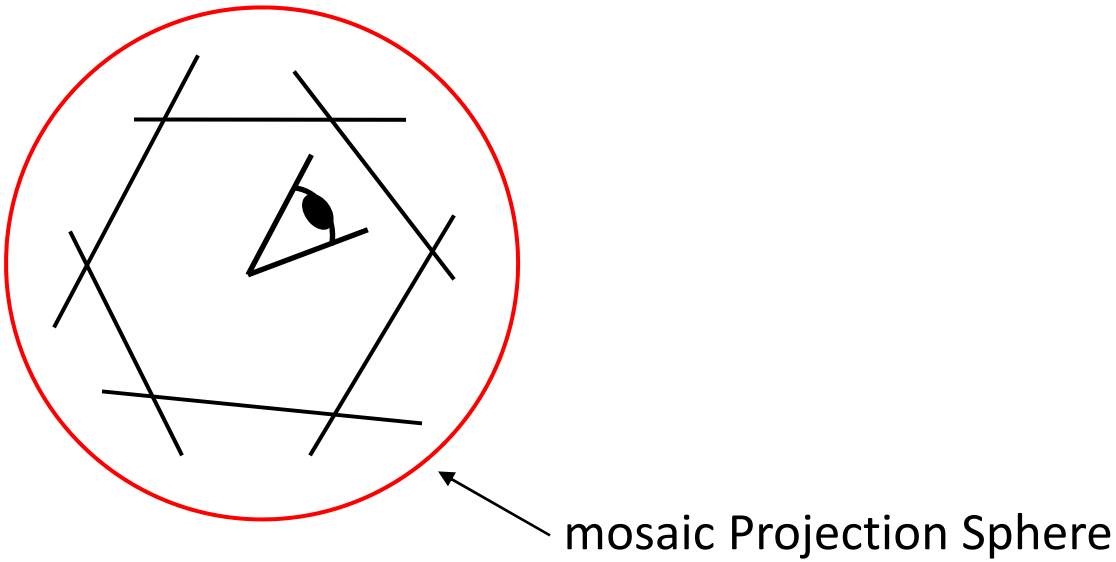


each image is warped with a homography $\mathbf{H}$

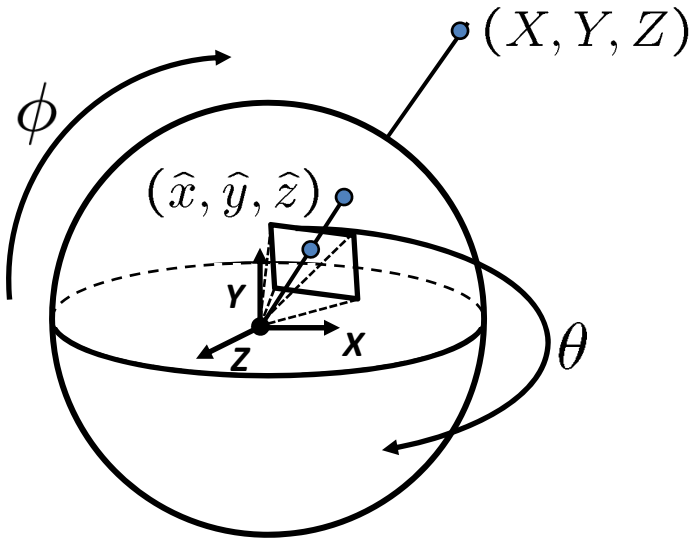
Can’t create a 360 panorama this way…

Panoramas

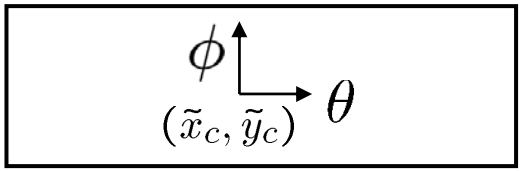
- What if you want a 360° field of view?



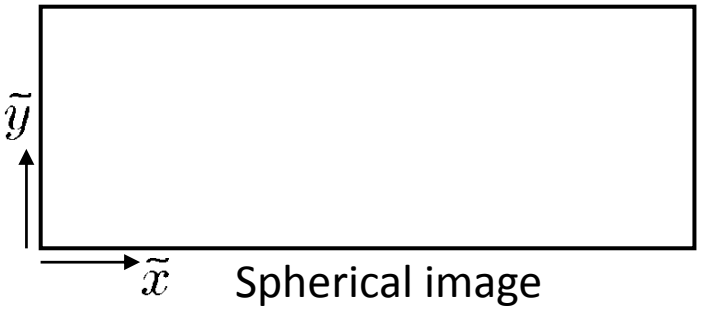
Spherical projection



unit sphere



unwrapped sphere



Spherical image

- Map image point (x, y) to a ray in the world

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \mathbf{R}^T \mathbf{K}^{-1} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

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

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

- Convert to spherical coordinates

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

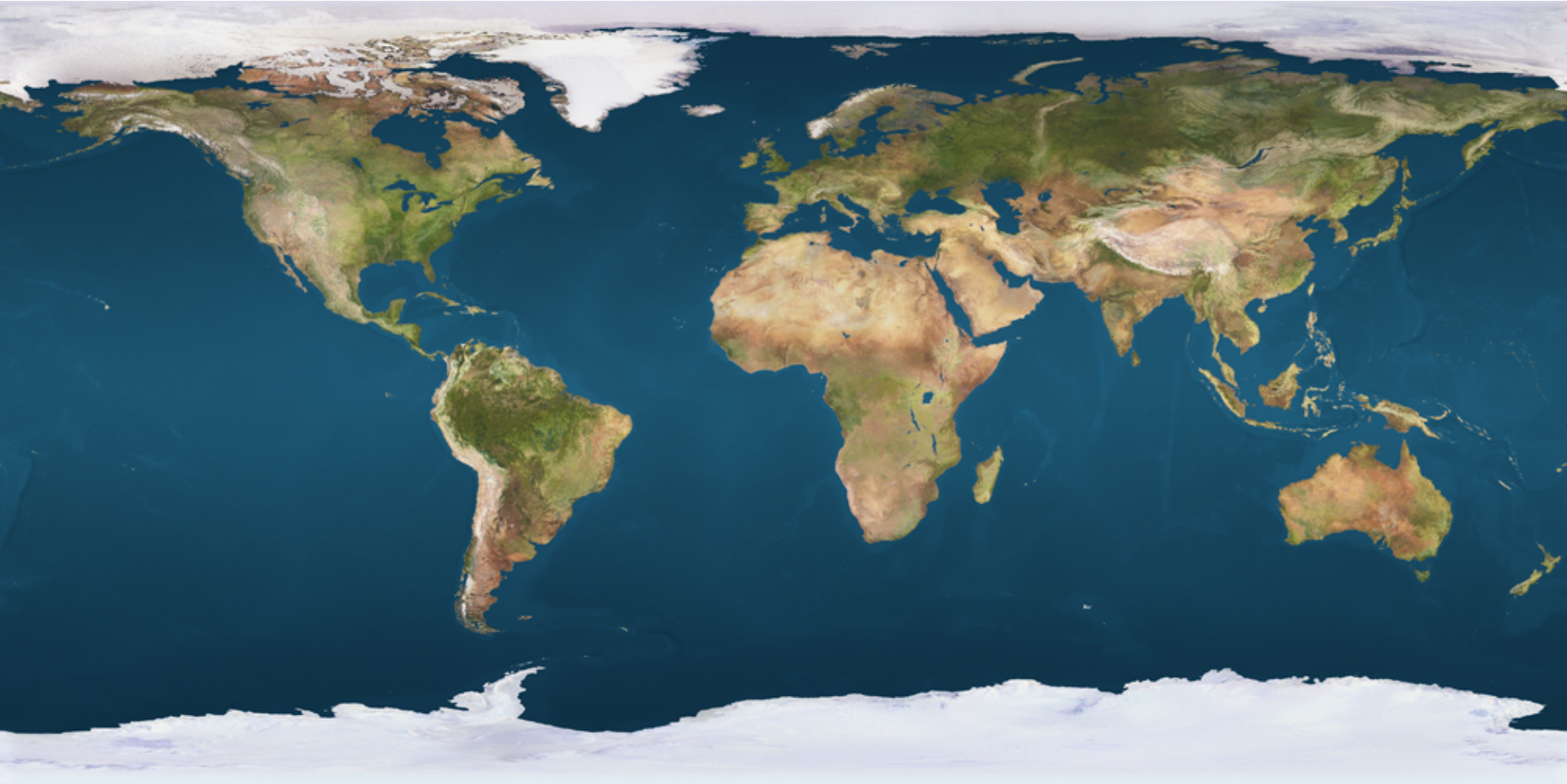
- Convert to spherical image coordinates

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

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

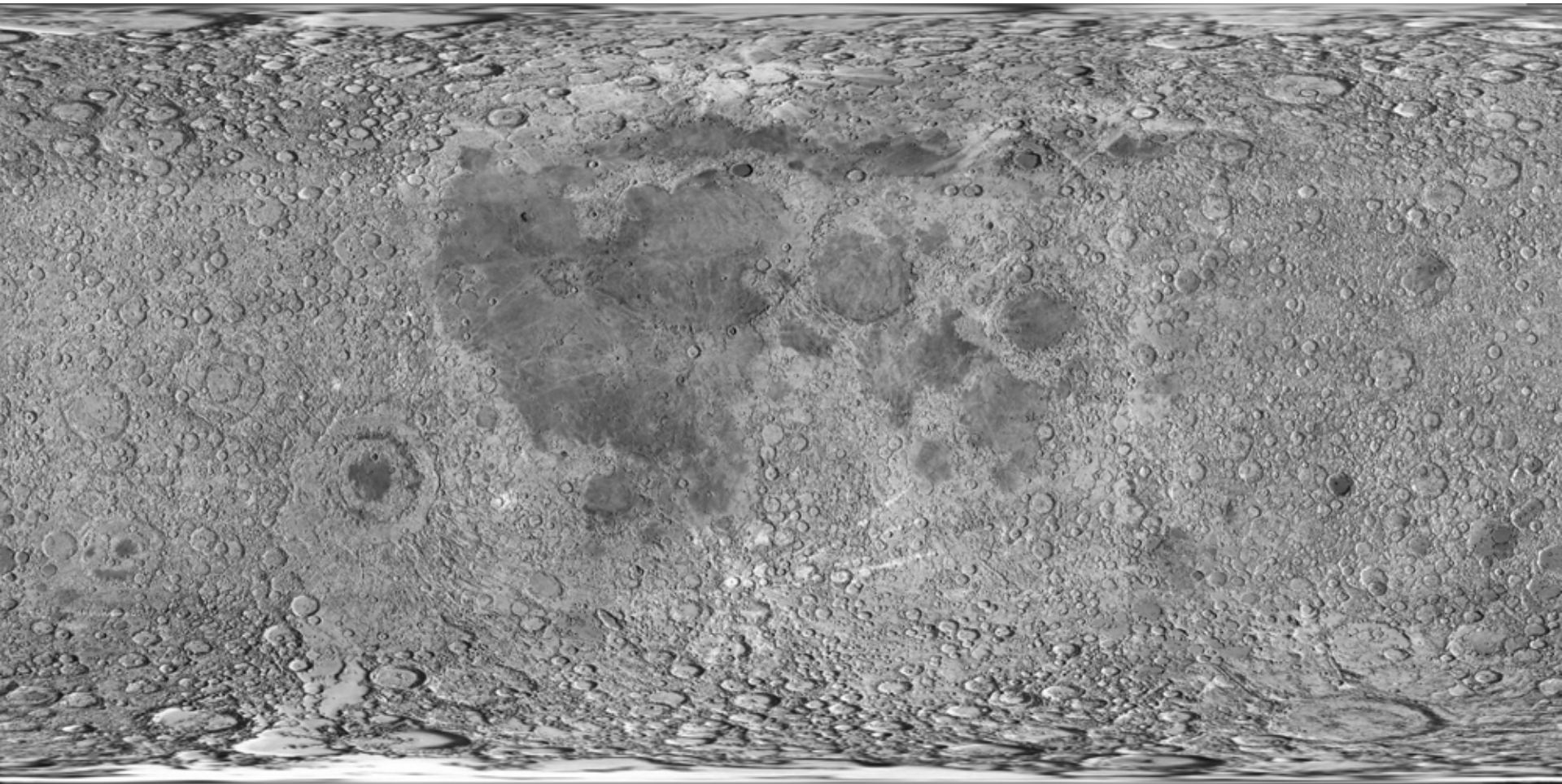
Unwrapping a sphere

Credit: JHT's Planetary Pixel Emporium



Unwrapping a sphere

Credit: JHT's Planetary Pixel Emporium



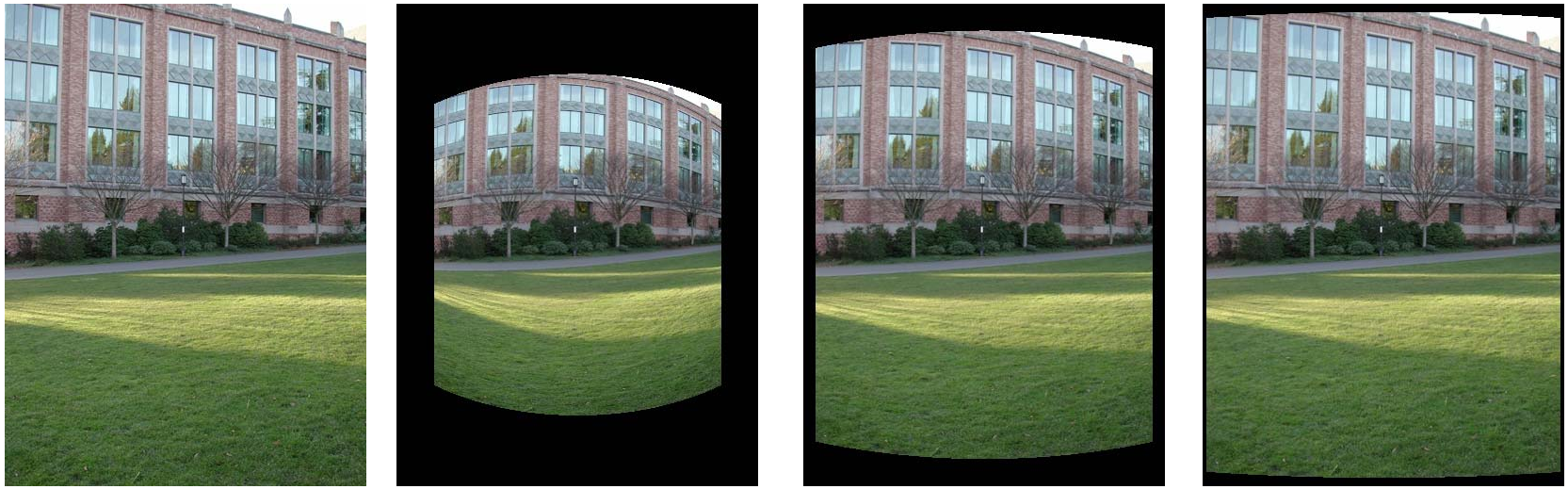
Spherical panoramas

Microsoft Lobby: http://www.acm.org/pubs/citations/proceedings/graph/258734/p251-szeliski

Different projections are possible

Spherical panoramas

Spherical reprojection



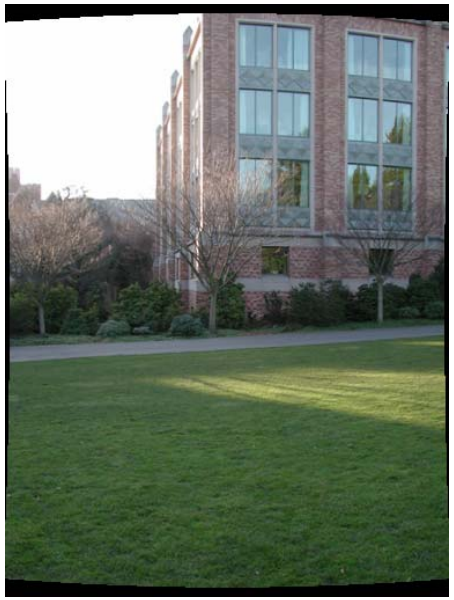
input | f = 200 (pixels) | f = 400 | f = 800

- Map image to spherical coordinates*
  - using inverse warping
  - need to know the focal length

*assuming the horizon is in the middle of the image, i.e. ϕ=0

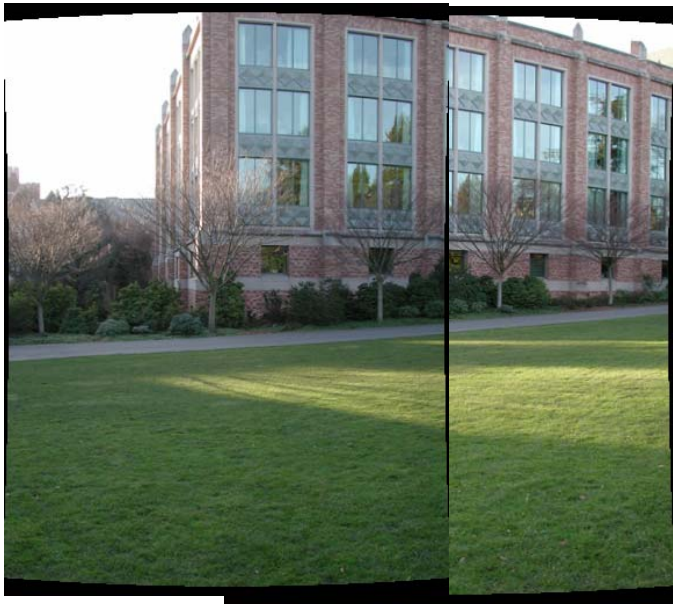
Aligning spherical images

What if we don't know **R**?



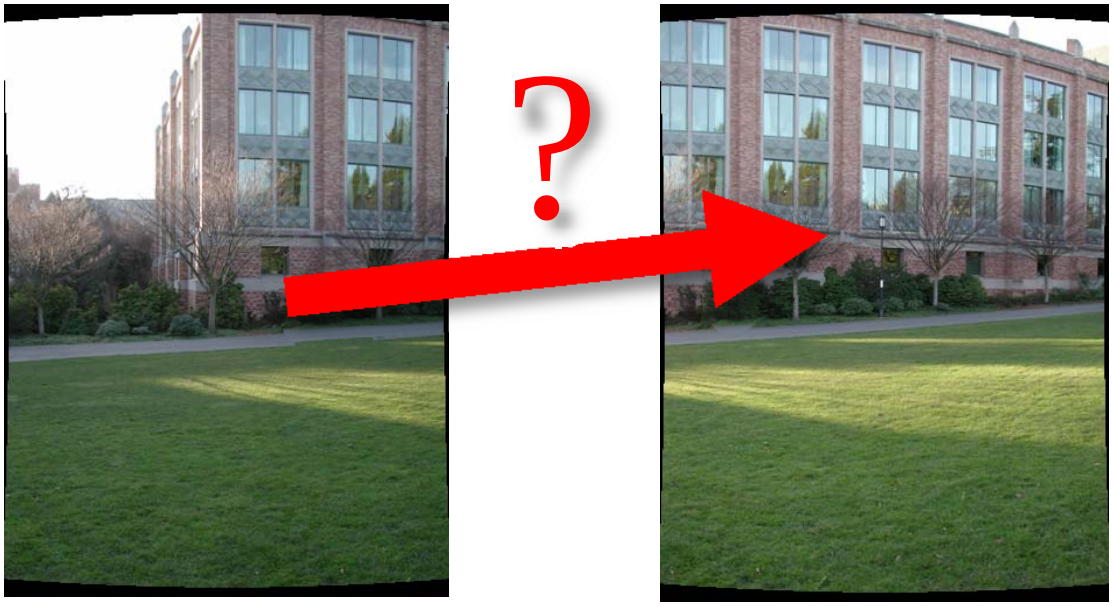
- Suppose we rotate the camera by θ about the vertical axis
  - How does this change the spherical image?

Aligning spherical images



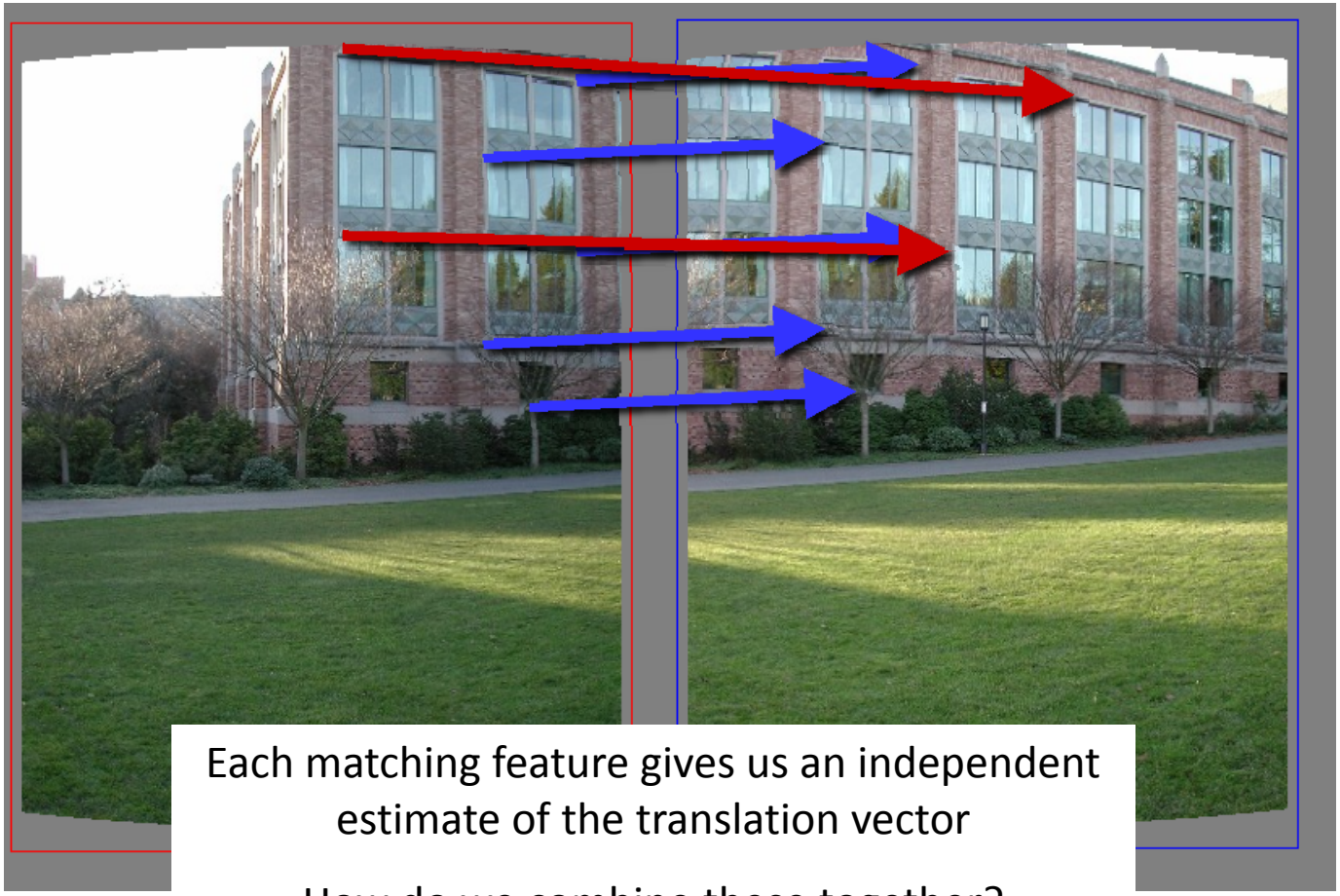
- Suppose we rotate the camera by θ about the vertical axis
  - How does this change the spherical image?
    - Translation by θ
  - This means that we can align spherical images by translation
  - (This doesn't quite work out if we rotate by ϕ about a horizontal axis)

Image alignment



- Given two images in spherical coordinates, how do we compute the translation that aligns them?
  - Answer: use feature matching

Feature-based alignment

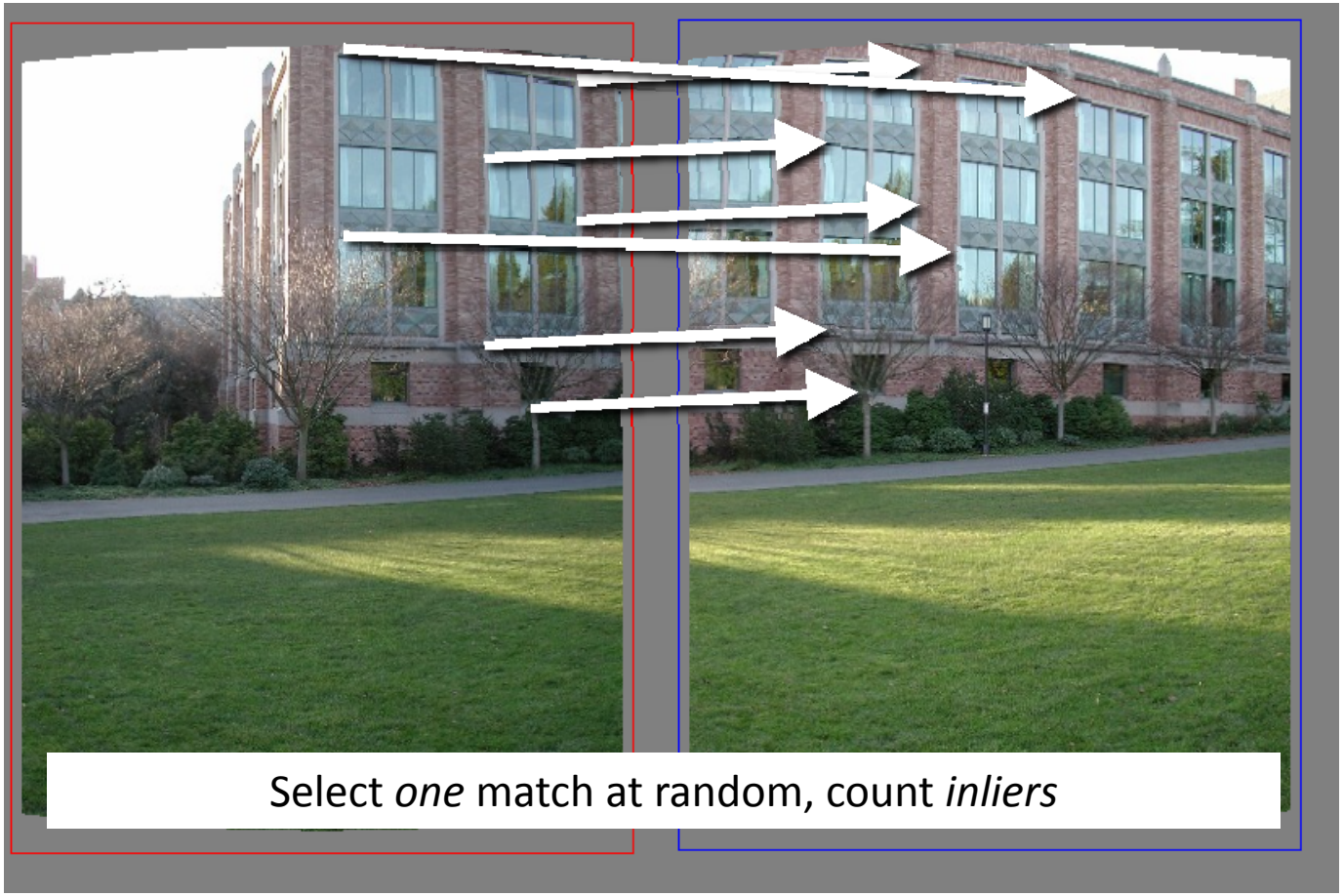


Each matching feature gives us an independent estimate of the translation vector

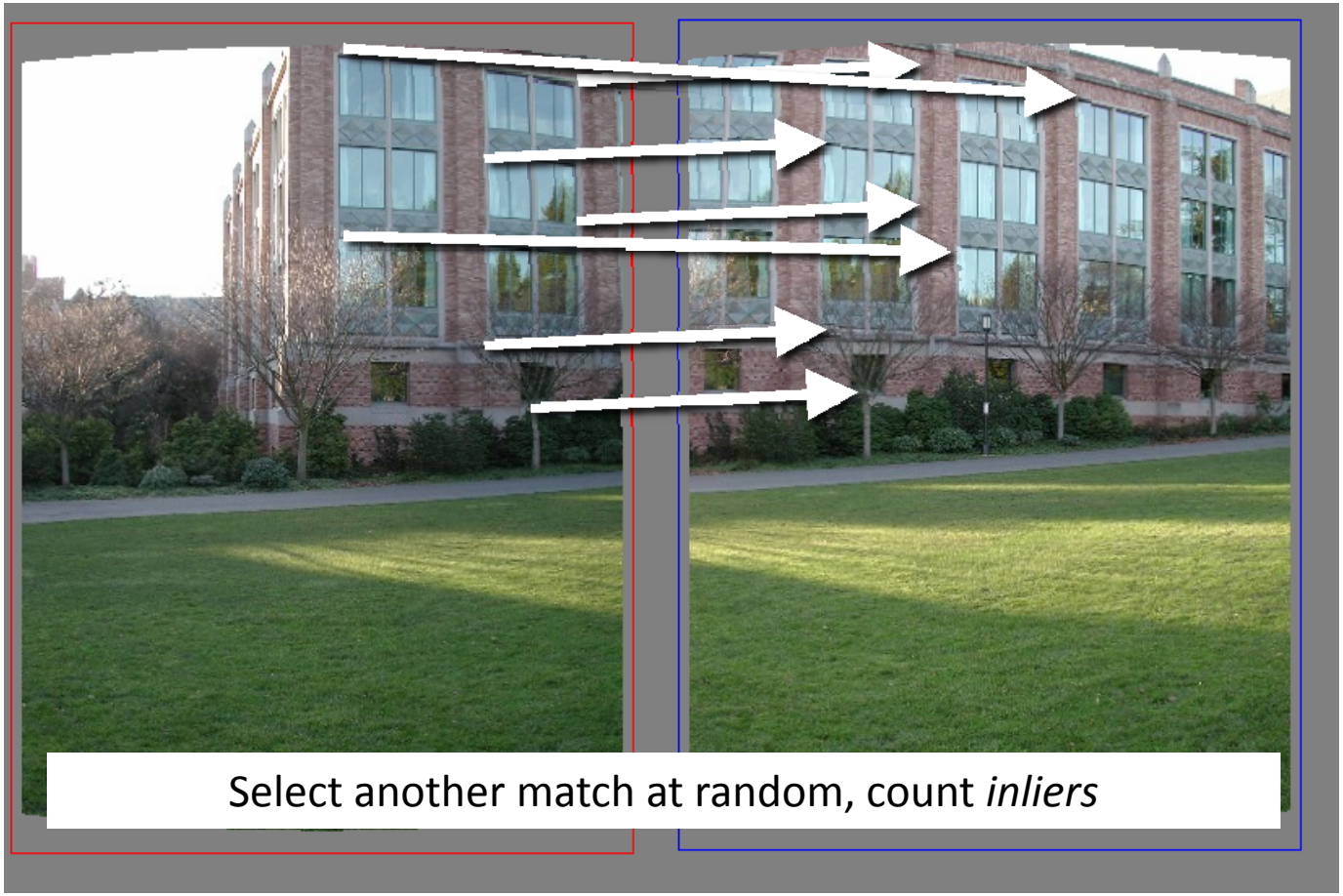
How do we combine these together?

What about the bad matches?

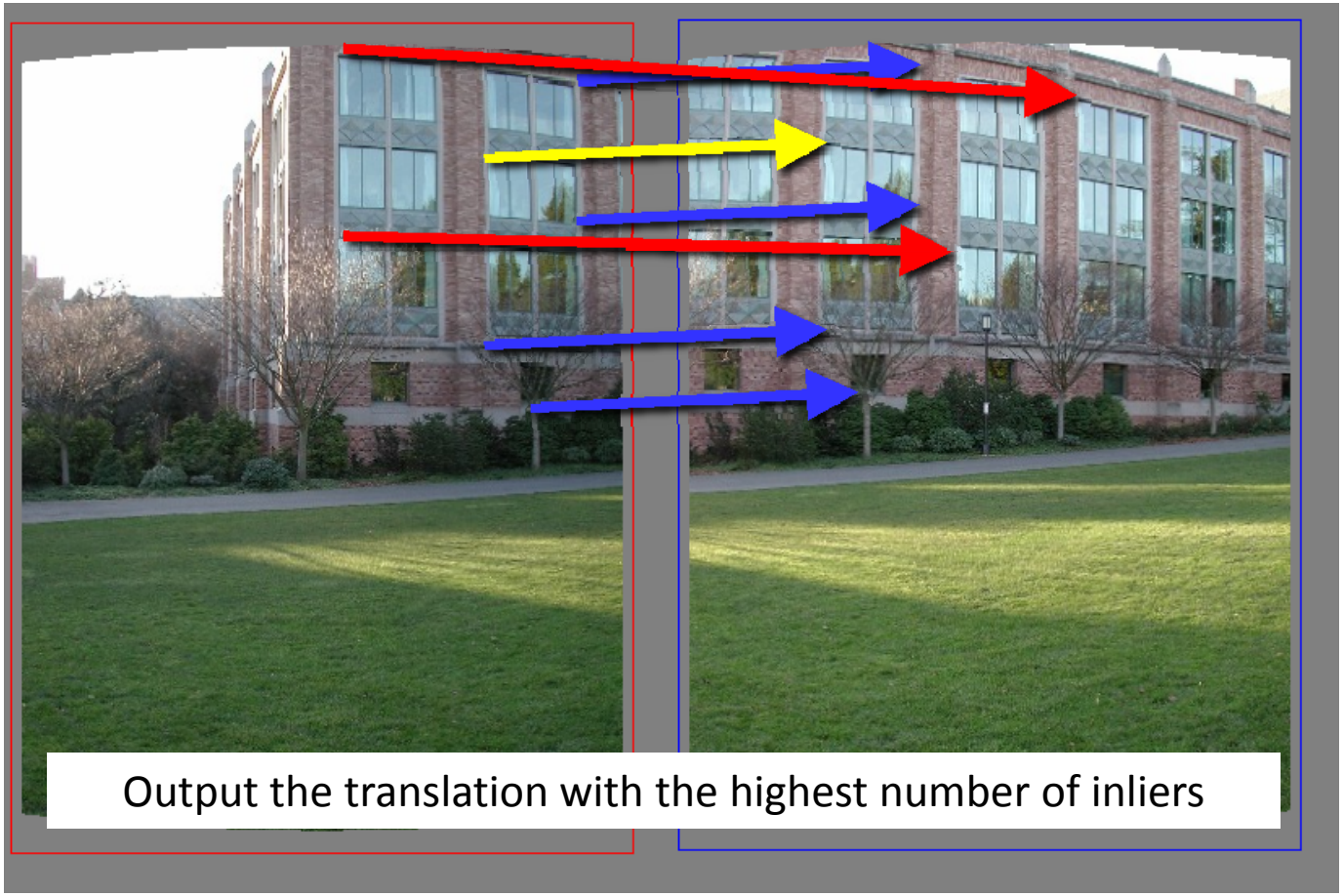
RAndom SAmple Consensus



RAndom SAmple Consensus



RAndom SAmple Consensus



RANSAC

- Idea:
  - All the inliers will agree with each other on the translation vector; the (hopefully small) number of outliers will (hopefully) disagree with each other
    - RANSAC only has guarantees if there are < 50% outliers
  - "All good matches are alike; every bad match is bad in its own way."

    – Tolstoy via Alyosha Efros

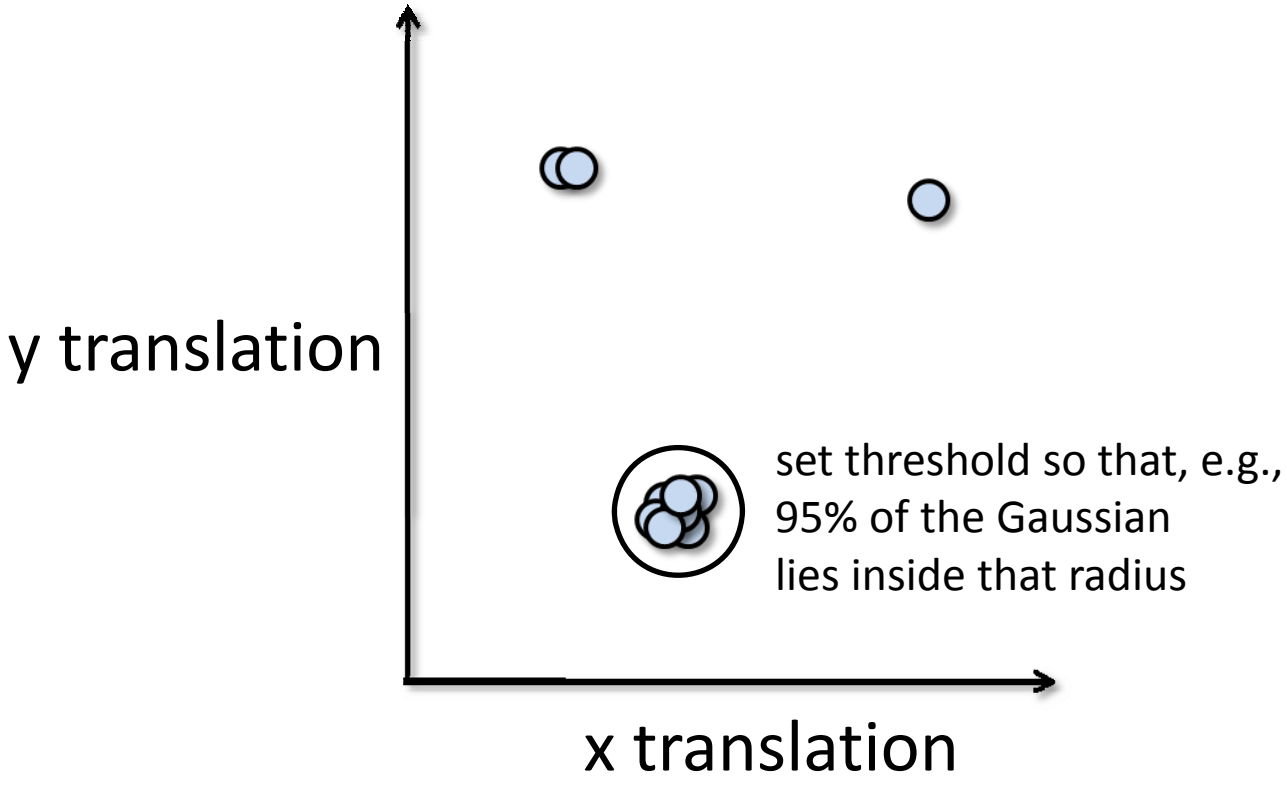
RANSAC

1. Pick a match at random
2. Compute the translation for that match
   - also called the “hypothesis”
3. Count the number of inlier matches (matches that agree with that translation)
   - What does it mean to agree?
4. Repeat some number of times
   - How many?
5. The winning hypothesis is one with highest vote

RANSAC

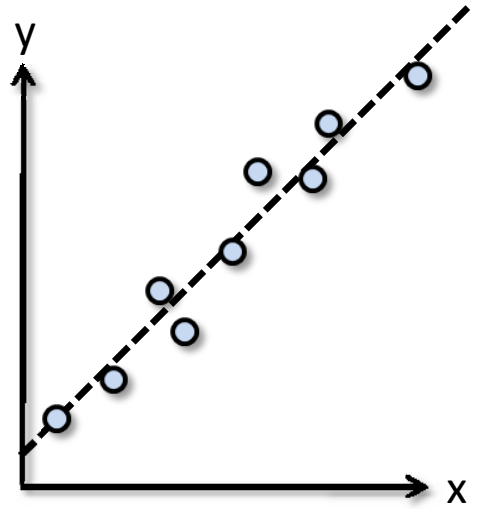
- **Inlier threshold** related to the amount of noise we expect in inliers
  - Often model noise as Gaussian with some standard deviation (e.g., 3 pixels)
- **Number of rounds** related to the percentage of outliers we expect, and the probability of success we'd like to guarantee
  - Suppose there are 20% outliers, and we want to find the correct answer with 99% probability
  - How many rounds do we need?

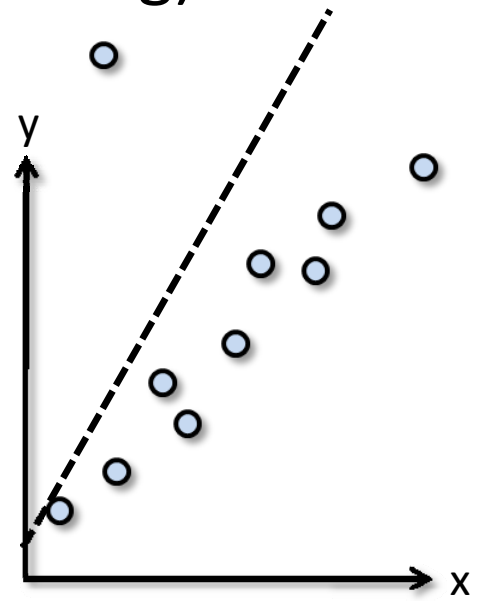
RANSAC


y translation
x translation
set threshold so that, e.g.,
95% of the Gaussian
lies inside that radius

RANSAC

- RANSAC can be used to fit any parametric model in the presence of outliers
  - e.g., linear regression (line fitting)

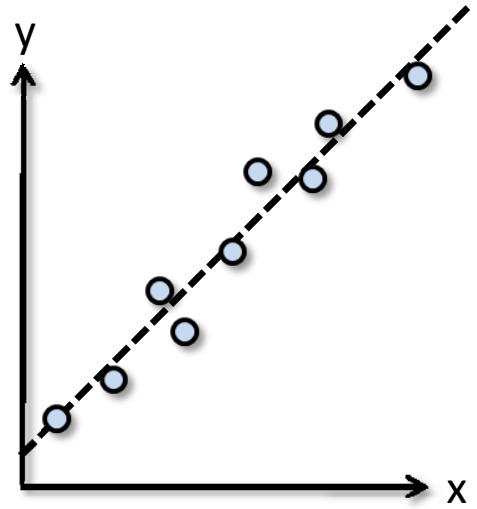




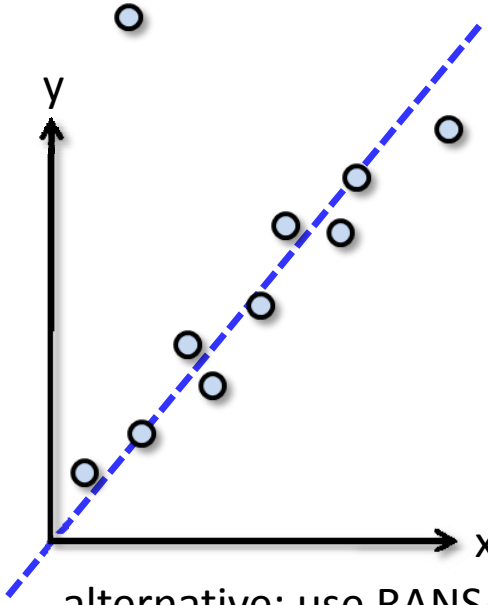
"fit a line $y = ax + b$ to these data points"
one solution: solve for a, b that minimize the sum of squared vertical distance to the line

RANSAC

- RANSAC can be used to fit any parametric model in the presence of outliers
  - e.g., linear regression (line fitting)



"fit a line $y = ax + b$ to these data points"
one solution: solve for a, b that minimize the sum of squared vertical distance to the line



alternative: use RANSAC
(need 2 points to generate a hypothesis)

RANSAC

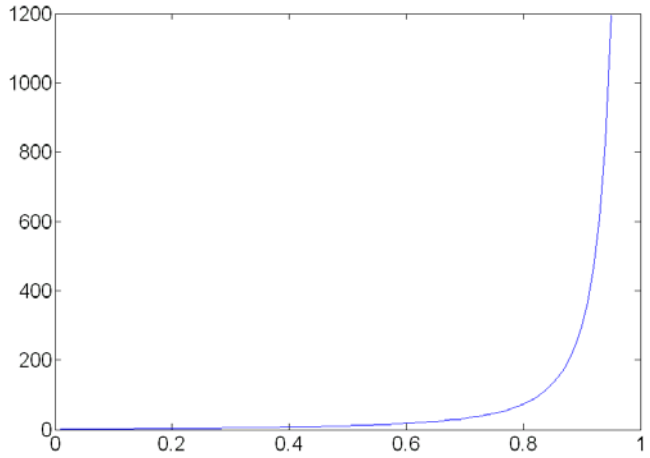
- General version:
  1. Randomly choose *s* samples
     - Typically *s* = minimum sample size that lets you fit a model
  2. Fit a model (e.g., line) to those samples
  3. Count the number of inliers that approximately fit the model
  4. Repeat *N* times
  5. Choose the model that has the largest set of inliers

How many rounds?

- If we have to choose *s* samples each time
  - with an outlier ratio *e*
  - and we want the right answer with probability *p*

	proportion of outliers *e*						
s	5%	10%	20%	25%	30%	40%	50%
2	2	3	5	6	7	11	17
3	3	4	7	9	11	19	35
4	3	5	9	13	17	34	72
5	4	6	12	17	26	57	146
6	4	7	16	24	37	97	293
7	4	8	20	33	54	163	588
8	5	9	26	44	78	272	1177

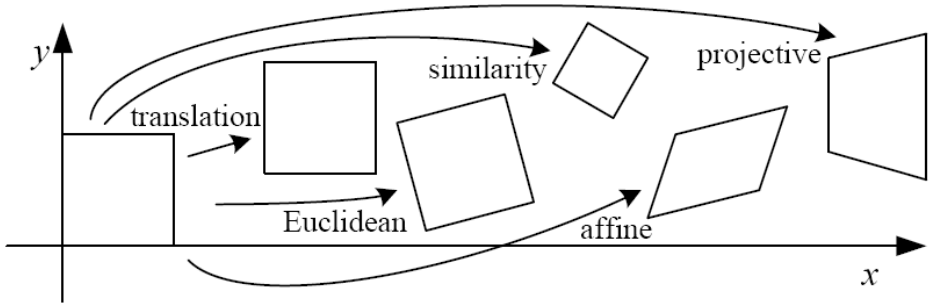
$p = 0.99$



Source: M. Pollefeys

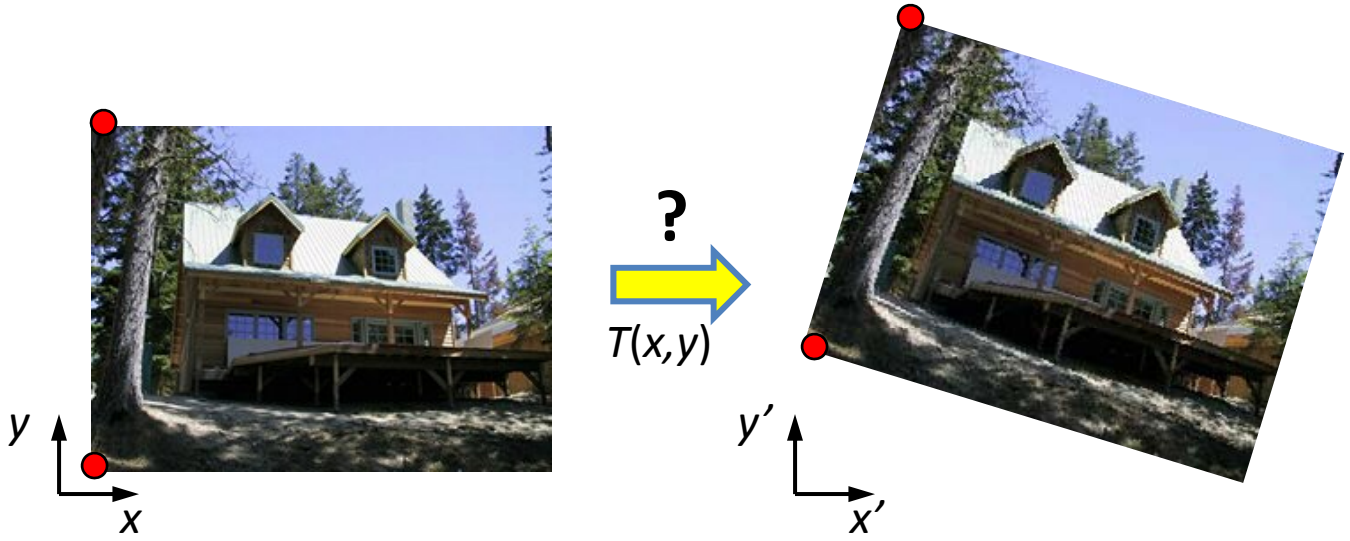
How big is *s*?

- For alignment, depends on the motion model
  - Here, each sample is a correspondence (pair of matching points)



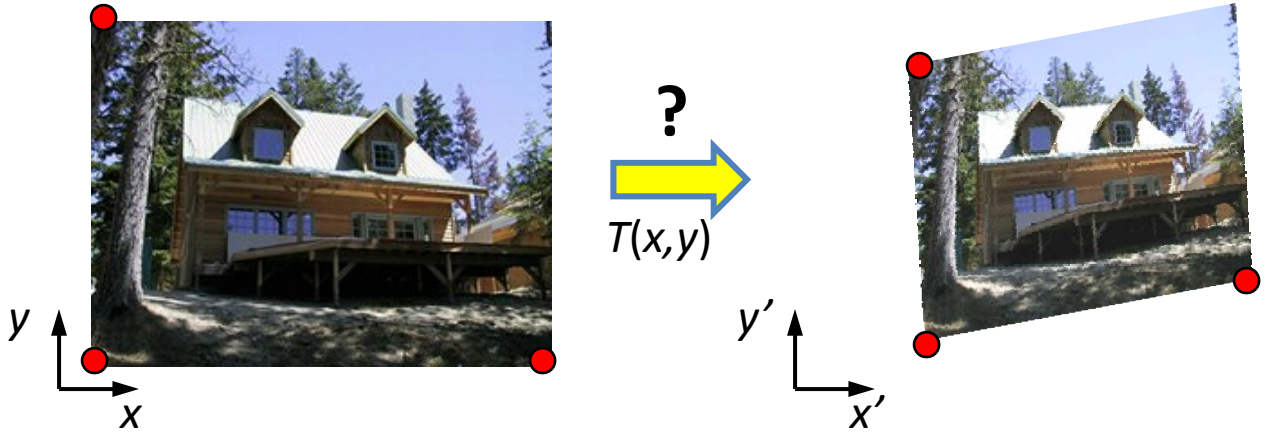
Name	Matrix	# D.O.F.	Preserves:	Icon
translation	$\left[\begin{array}{c\|c} \boldsymbol{I} & \boldsymbol{t} \end{array}\right]_{2\times 3}$	2	orientation $+\cdots$	
rigid (Euclidean)	$\left[\begin{array}{c\|c} \boldsymbol{R} & \boldsymbol{t} \end{array}\right]_{2\times 3}$	3	lengths $+\cdots$	
similarity	$\left[\begin{array}{c\|c} s\boldsymbol{R} & \boldsymbol{t} \end{array}\right]_{2\times 3}$	4	angles $+\cdots$	
affine	$\left[\begin{array}{c} \boldsymbol{A} \end{array}\right]_{2\times 3}$	6	parallelism $+\cdots$	
projective	$\left[\begin{array}{c} \tilde{\boldsymbol{H}} \end{array}\right]_{3\times 3}$	8	straight lines	

Euclidean: # correspondences?



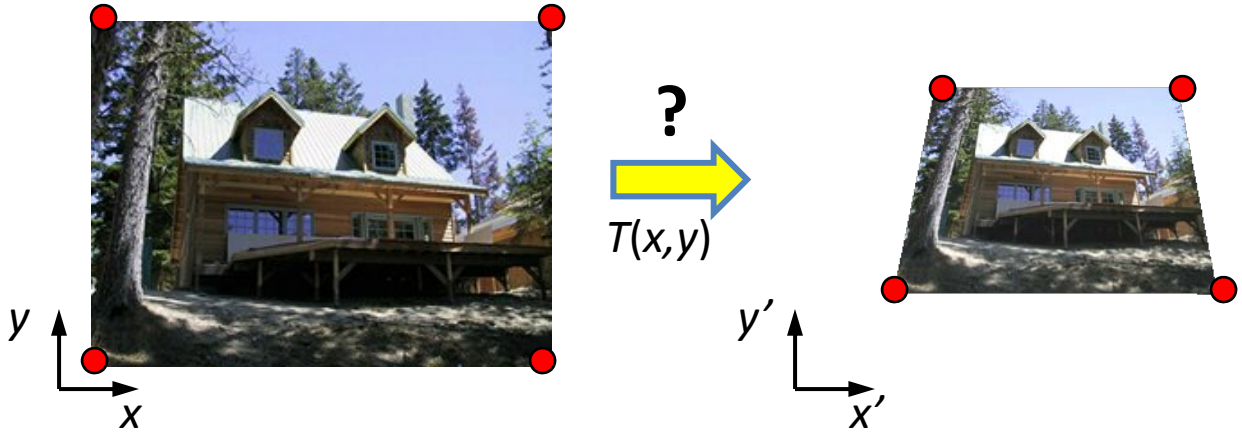
- How many DOF?
- How many correspondences needed for translation+rotation?

Affine: # correspondences?



- How many DOF?
- How many correspondences?

Projective: # correspondences?

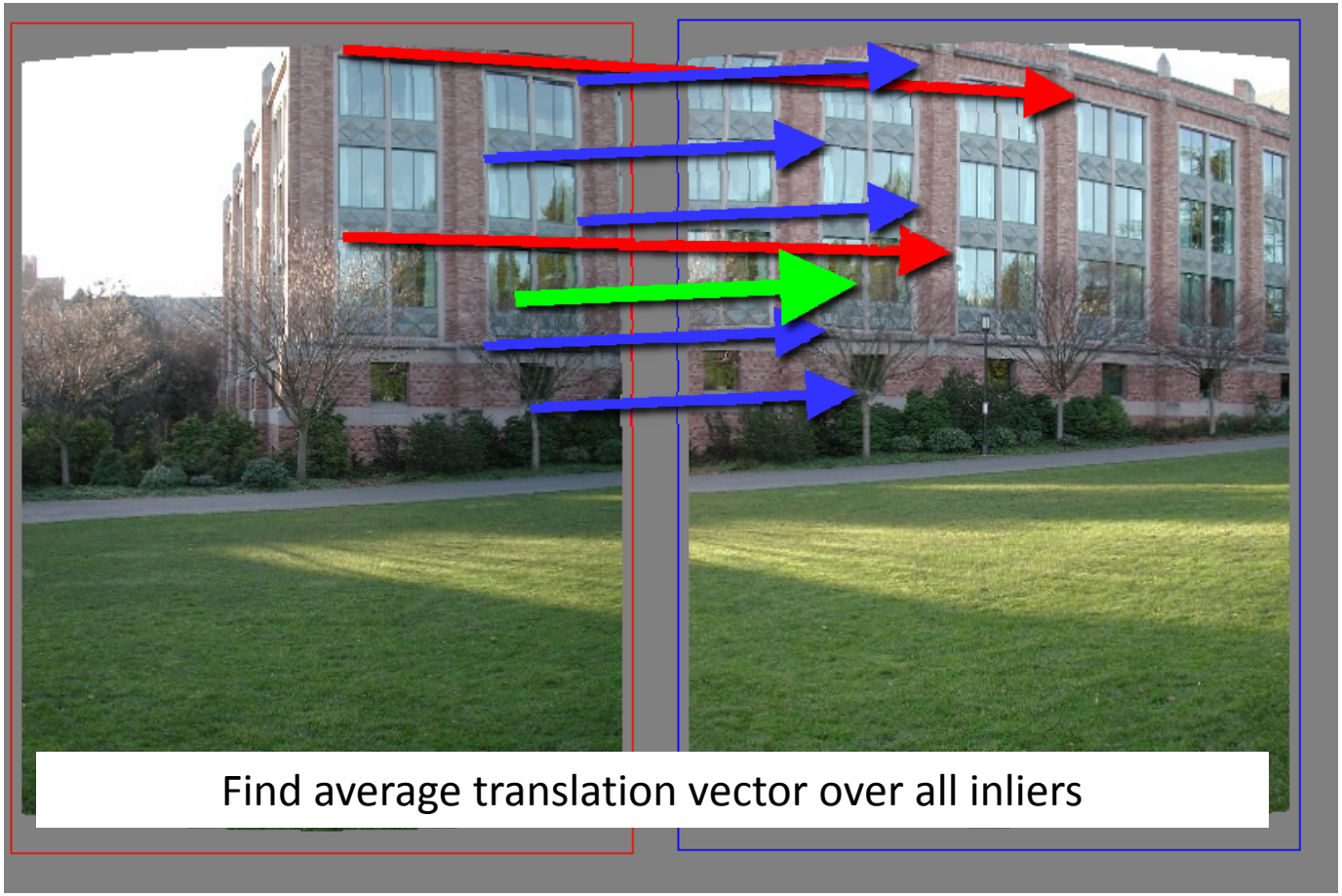


- How many DOF?
- How many correspondences?

RANSAC pros and cons

- Pros
  - Simple and general
  - Applicable to many different problems
  - Often works well in practice
- Cons
  - Lots of parameters to tune
  - Can’t always get a good initialization of the model based on the minimum number of samples
  - Sometimes too many iterations are required
  - Can fail for extremely low inlier ratios
  - We can often do better than brute-force sampling

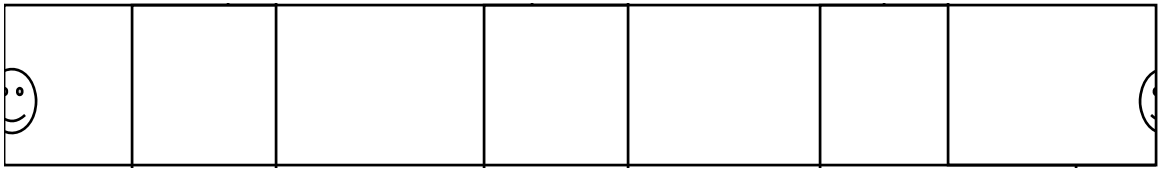
Final step: least squares fit



More on this later…

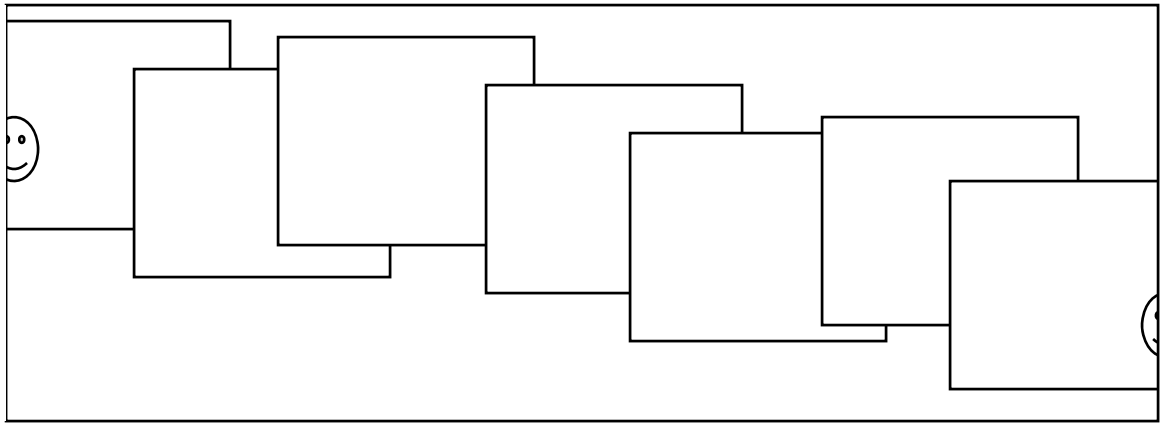
Questions?

Assembling the panorama



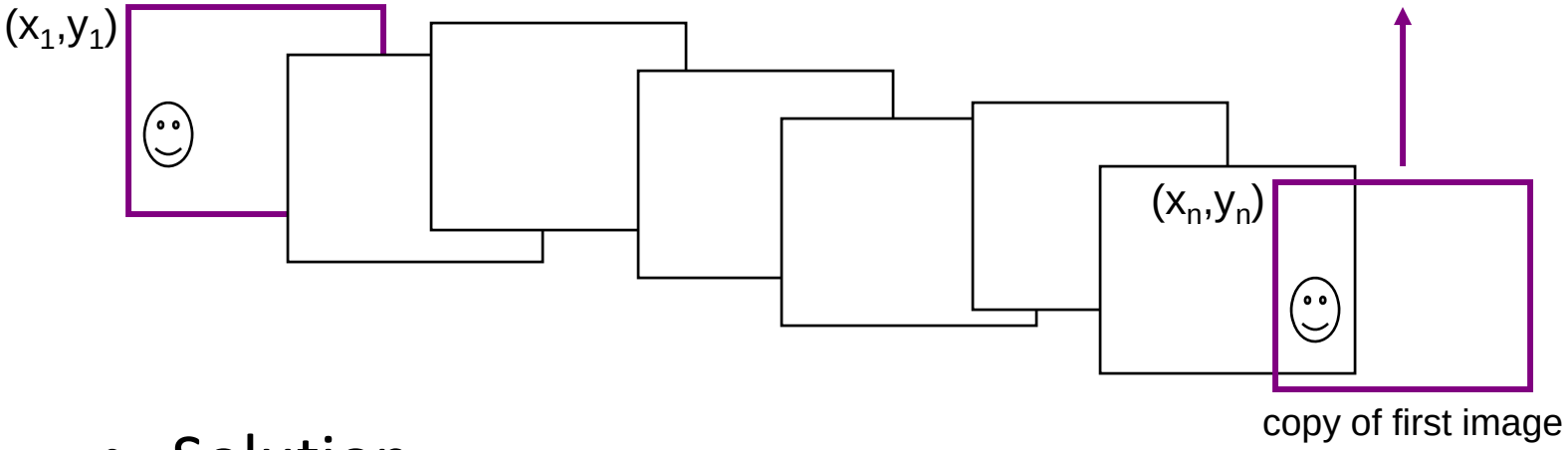
- Stitch pairs together, blend, then crop

Problem: Drift



- Error accumulation
  - small errors accumulate over time

Problem: Drift



- Solution
  - add another copy of first image at the end
  - this gives a constraint: $y_n = y_1$
  - there are a bunch of ways to solve this problem
    - add displacement of $(y_1 - y_n)/(n - 1)$ to each image after the first
    - compute a global warp: $y' = y + ax$
    - run a big optimization problem, incorporating this constraint
      - best solution, but more complicated
      - known as “bundle adjustment”

End-to-end alignment and crop

Full-view Panorama

+

+

+

+

Project 2

1. Take pictures on a tripod (or handheld)
2. Warp to spherical coordinates
3. Extract features
4. Align neighboring pairs using RANSAC
5. Write out list of neighboring translations
6. Correct for drift
7. Read in warped images and blend them
8. Crop the result and import into a viewer

- Roughly based on **Autostitch**
  - By Matthew Brown and David Lowe
  - http://www.cs.ubc.ca/~mbrown/autostitch/autostitch.html

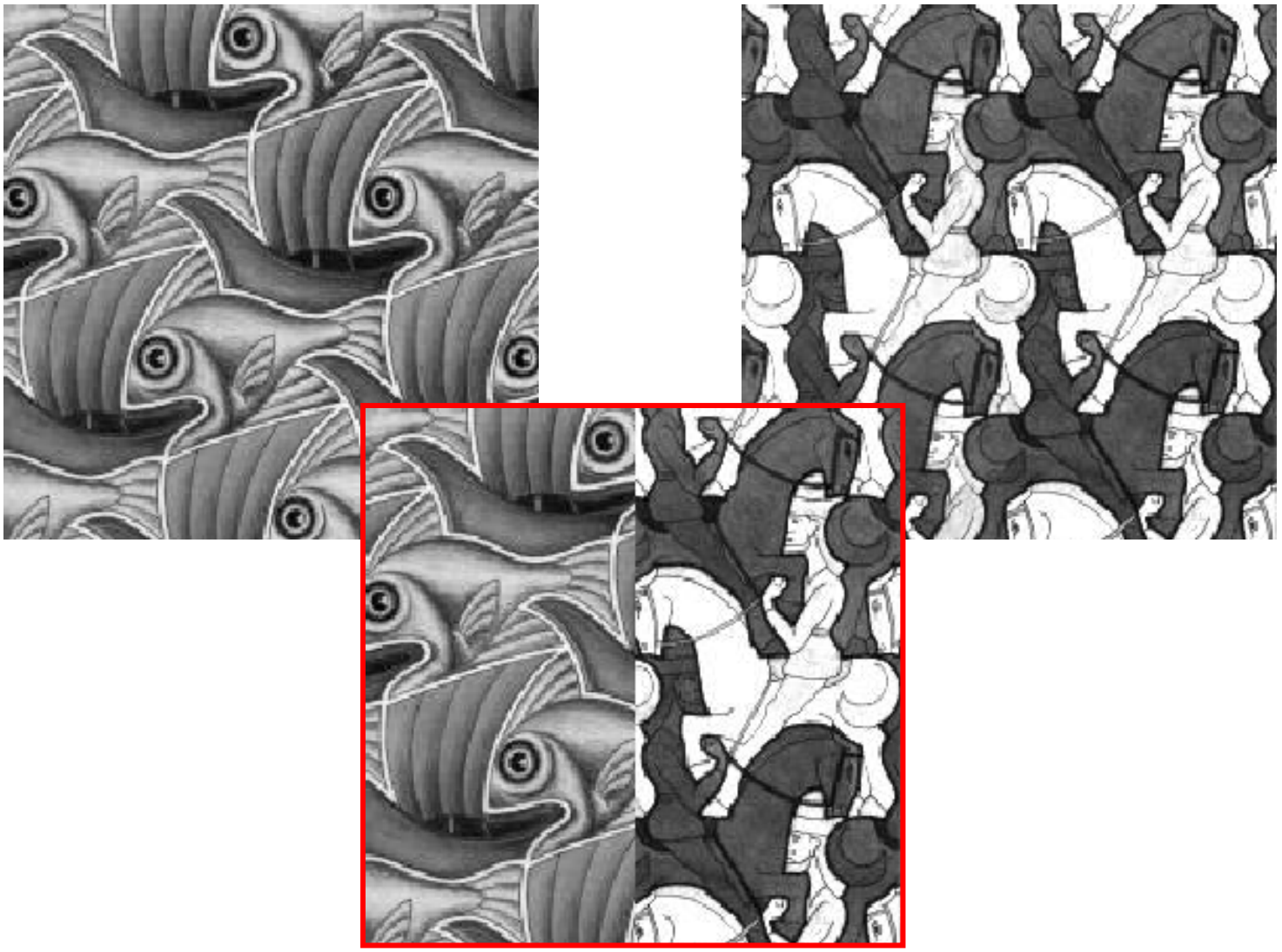
Blending

- We’ve aligned the images – now what?

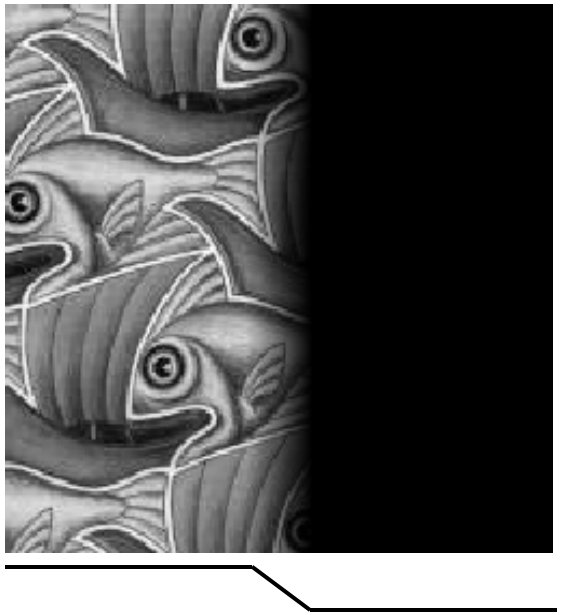
Blending

- Want to seamlessly blend them together

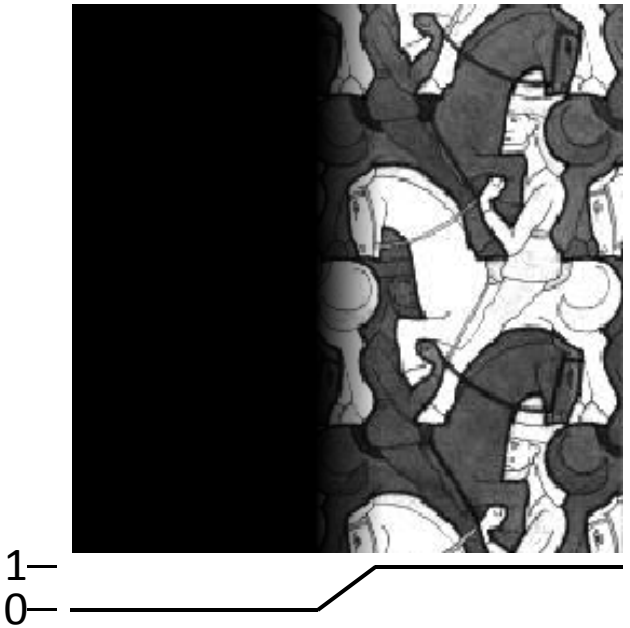
Image Blending



Feathering

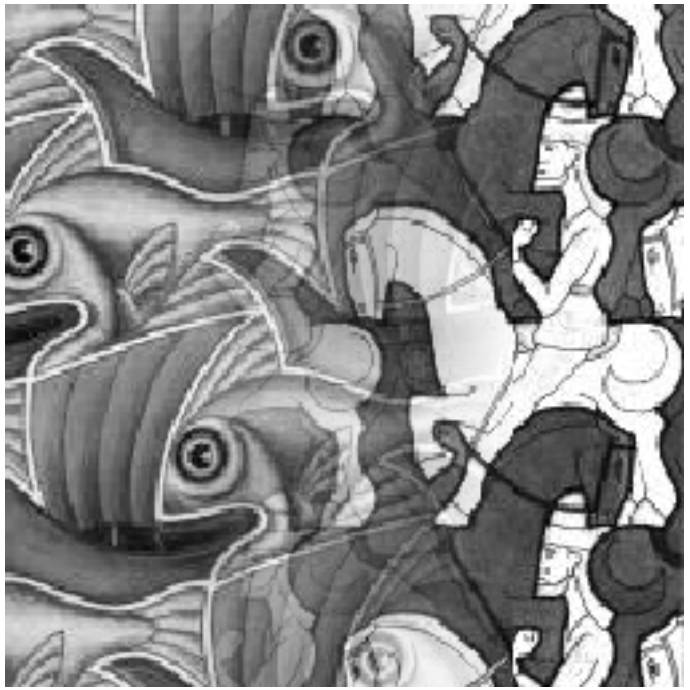


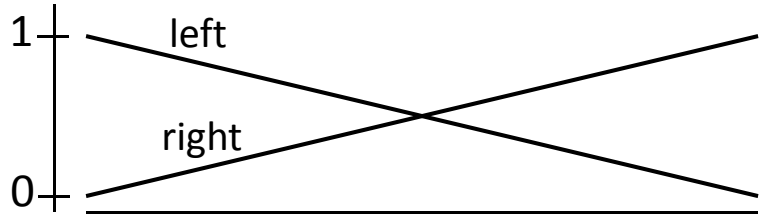
+

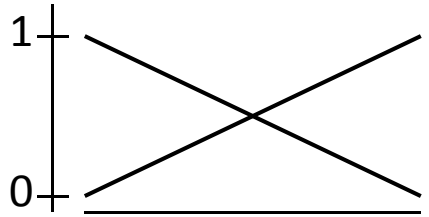


=

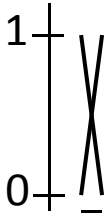
Effect of window size

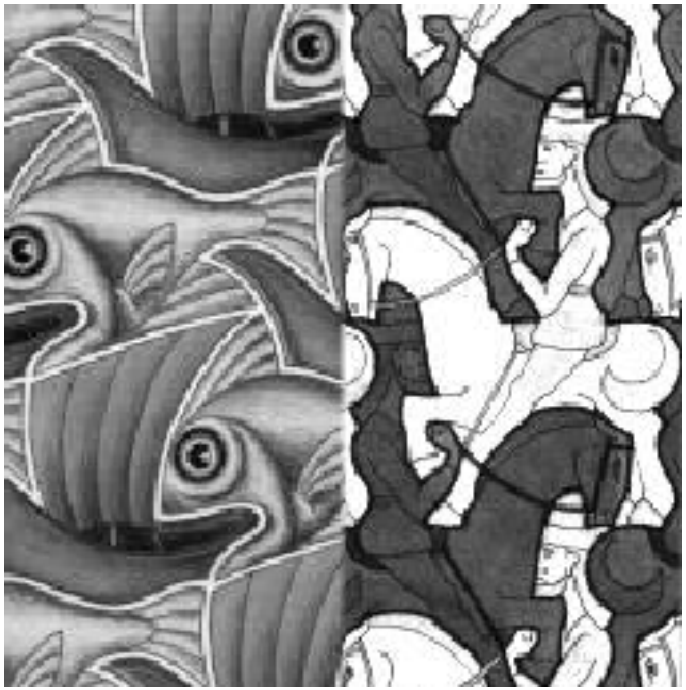


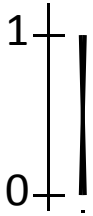




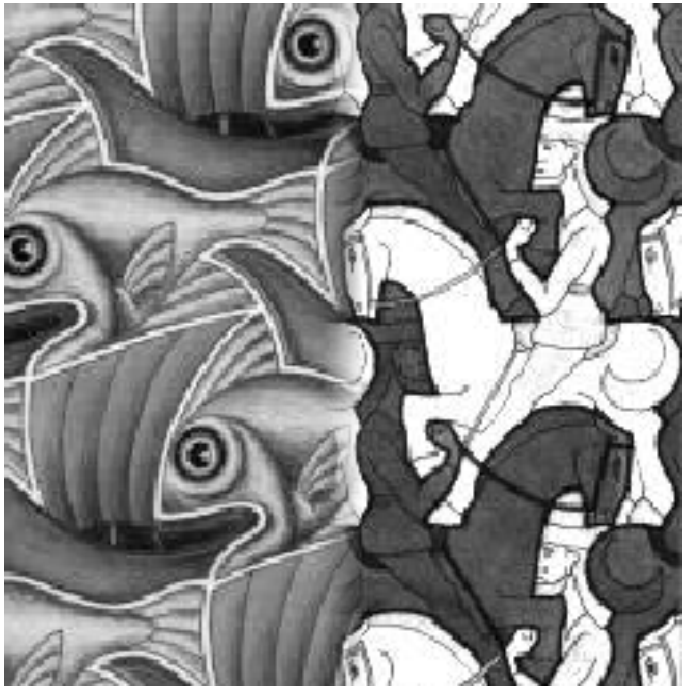
Effect of window size







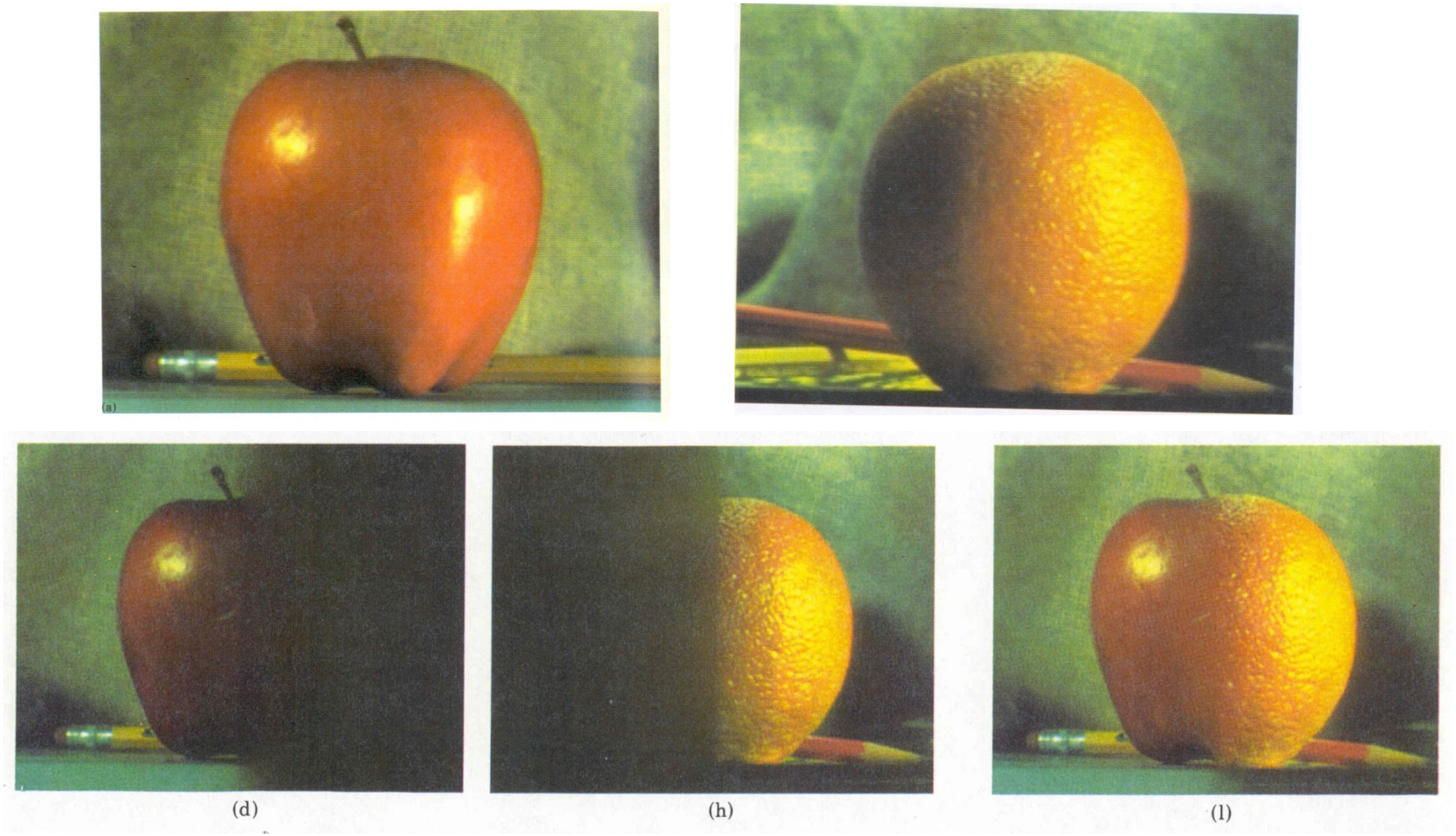
Good window size



"Optimal" window: smooth but not ghosted

- Doesn't always work...

Pyramid blending



Create a Laplacian pyramid, blend each level

- 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.

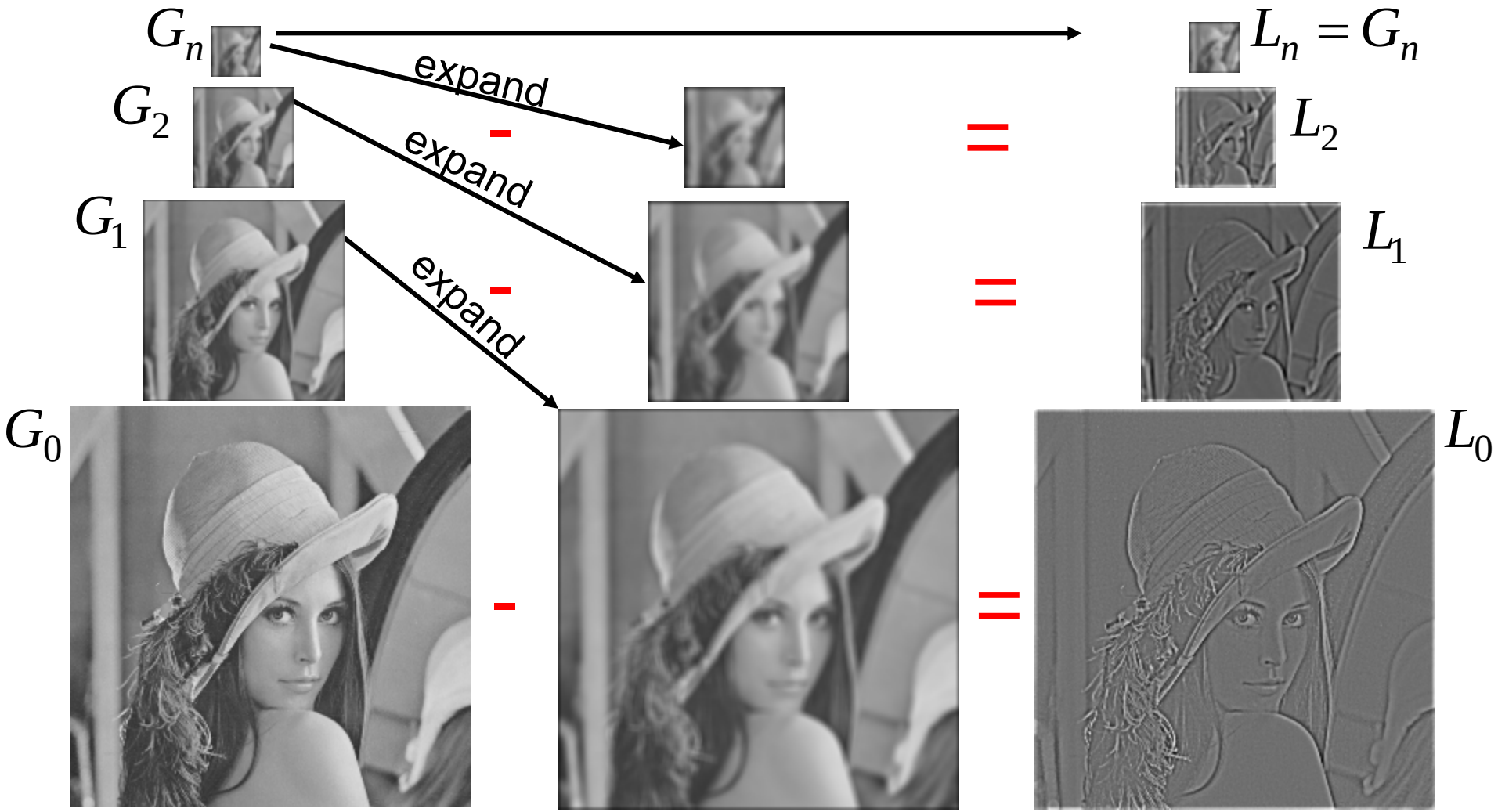
The Laplacian Pyramid

$$L_i = G_i - \text{expand}(G_{i+1})$$

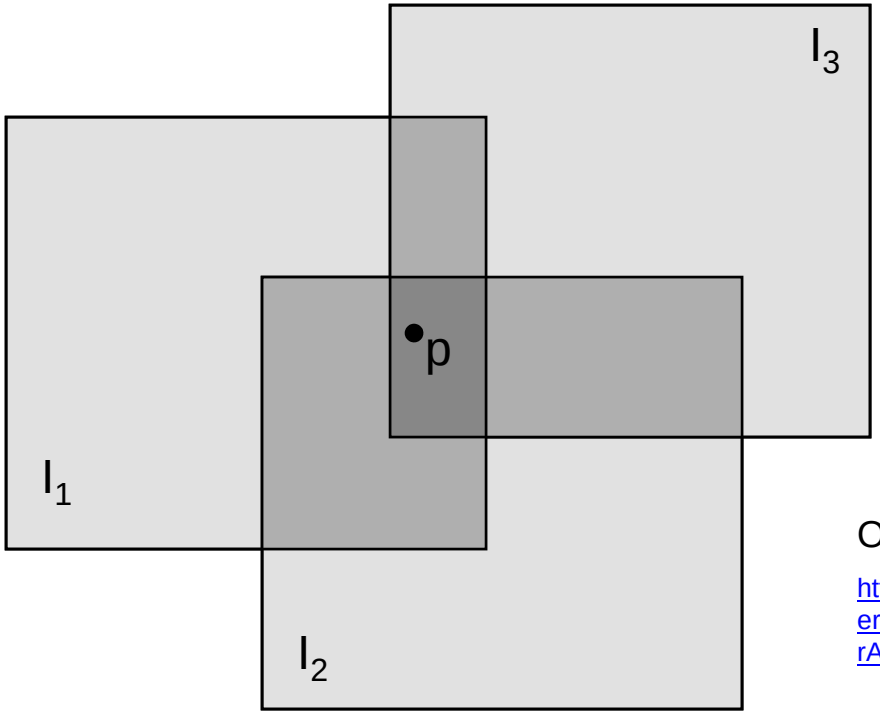
Gaussian Pyramid

$$G_i = L_i + \text{expand}(G_{i+1})$$

Laplacian Pyramid



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

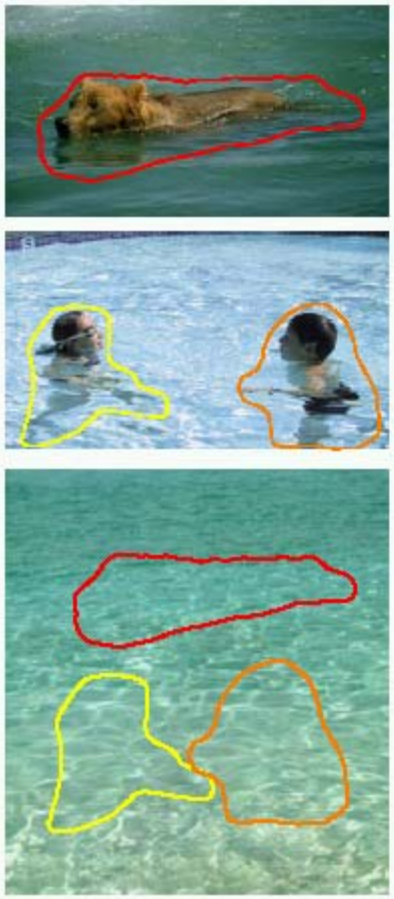
color at p = $$\frac{(\alpha_1 R_1,\ \alpha_1 G_1,\ \alpha_1 B_1) + (\alpha_2 R_2,\ \alpha_2 G_2,\ \alpha_2 B_2) + (\alpha_3 R_3,\ \alpha_3 G_3,\ \alpha_3 B_3)}{\alpha_1 + \alpha_2 + \alpha_3}$$

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 $\alpha = 0$?

Poisson Image Editing



sources/destinations

cloning

seamless cloning

- For more info: Perez et al, SIGGRAPH 2003
  - http://research.microsoft.com/vision/cambridge/papers/perez_siggraph03.pdf

Some panorama examples

Before Siggraph Deadline:
http://www.cs.washington.edu/education/courses/cse590ss/01wi/projects/project1/students/dougz/siggraph-hires.html

Some panorama examples

- Every image on Google Streetview

Magic: ghost removal

M. Uyttendaele, A. Eden, and R. Szeliski.
***Eliminating ghosting and exposure artifacts in image mosaics*.**
In Proceedings of the Interational Conference on Computer Vision and Pattern Recognition, volume 2, pages 509--516, Kauai, Hawaii, December 2001.

Magic: ghost removal

M. Uyttendaele, A. Eden, and R. Szeliski.
***Eliminating ghosting and exposure artifacts in image mosaics*.**
In Proceedings of the Interational Conference on Computer Vision and Pattern Recognition, volume 2, pages 509--516, Kauai, Hawaii, December 2001.

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…

Questions?