Mosaics part 3

CSE 455, Winter 2010

February 12, 2010

Announcements

- The Midterm:
  - **Due this NOW**
  - **Late exams not be accepted**

- Project 3:
  - Out today
  - We'll discuss it towards the end of class

Review From Last Time

How to do it?

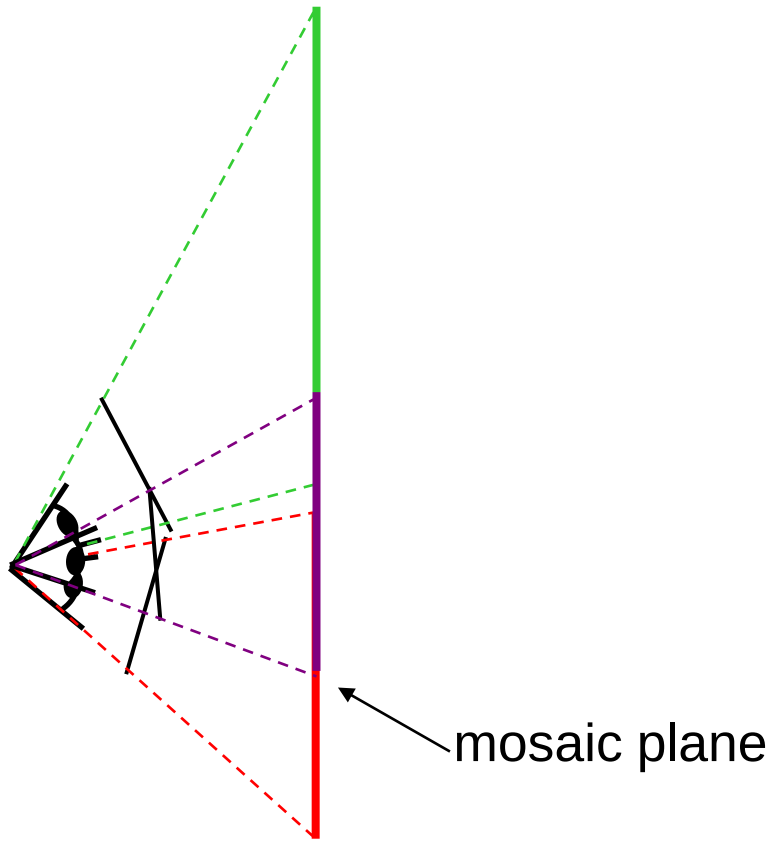
- Similar to Structure from Motion, but easier
- Basic Procedure
  - Take a sequence of images from the same position
    - Rotate the camera about its optical center
  - Compute transformation between second image and first
  - Transform the second image to overlap with the first
  - Blend the two together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

Panoramic Stitching

Input

Goal

Image reprojection

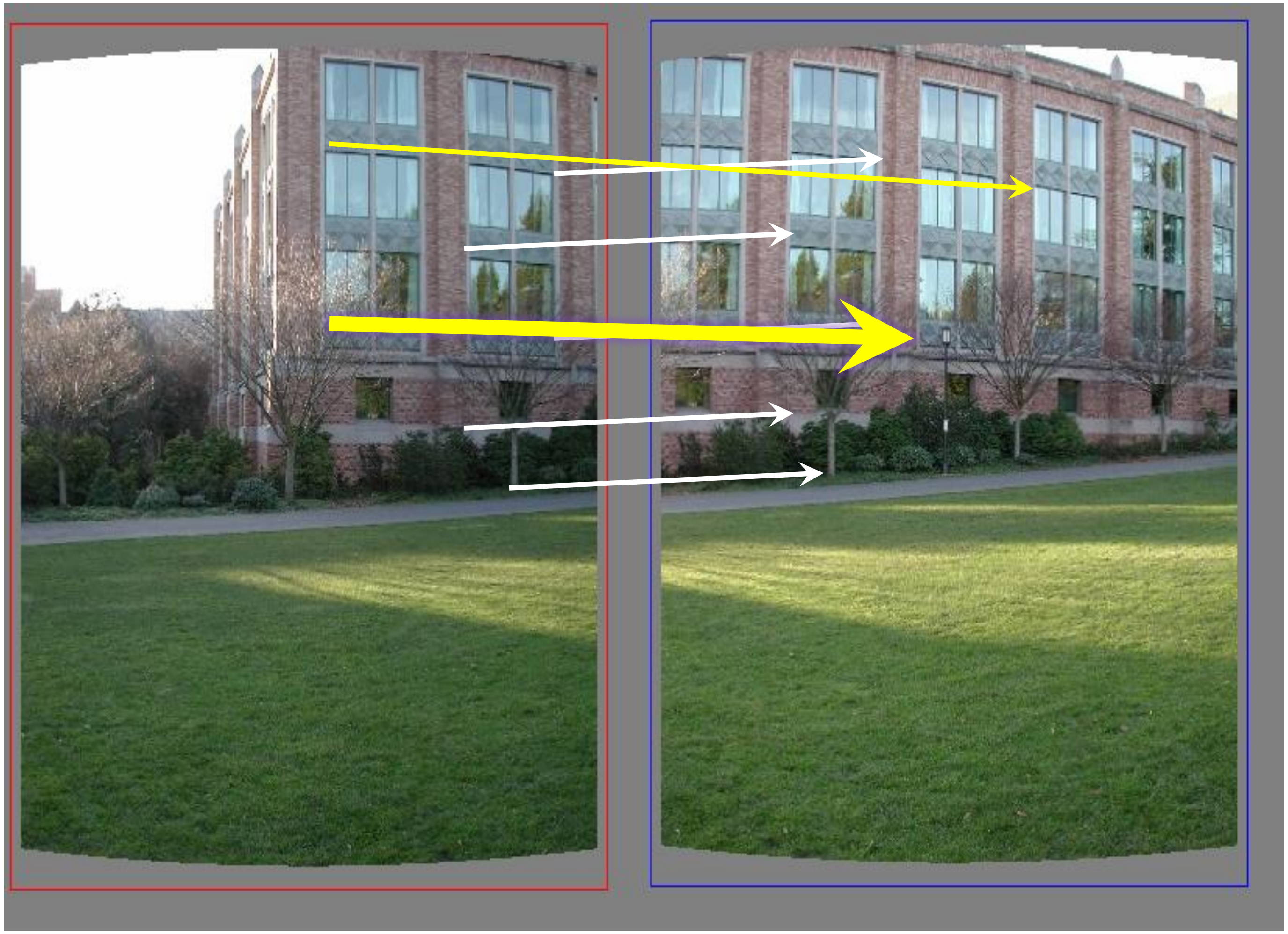


- The mosaic has a natural interpretation in 3D
  - The images are reprojected onto a common plane
  - The mosaic is formed on this plane

Algorihtm for finding correspondences

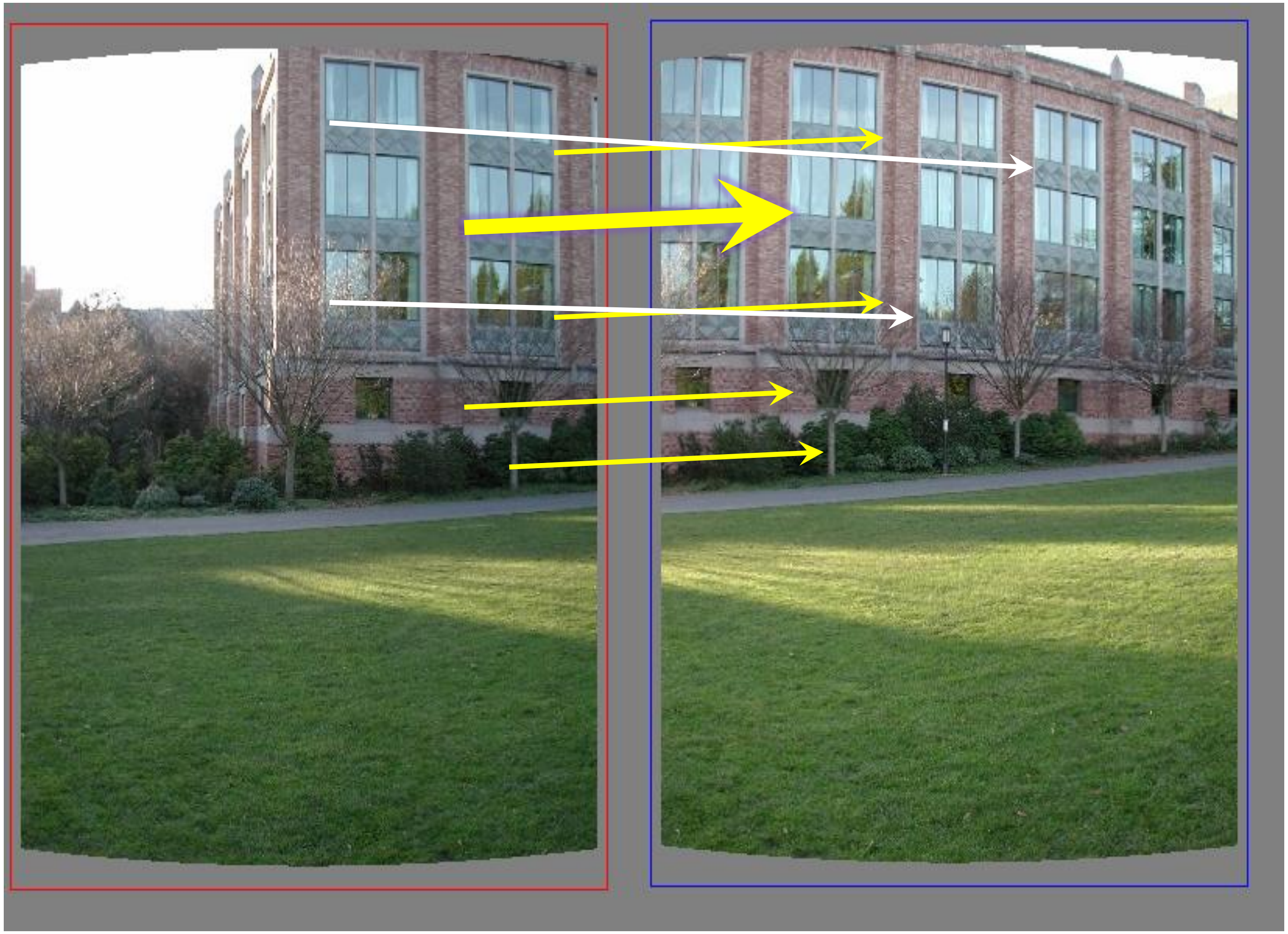
- Guess some matches
- Compute a transformation using those matches
- Check if the transformation is good (count “inliers”)
- Keep the correspondences and transformation the is the best

RAndom SAmple Consensus



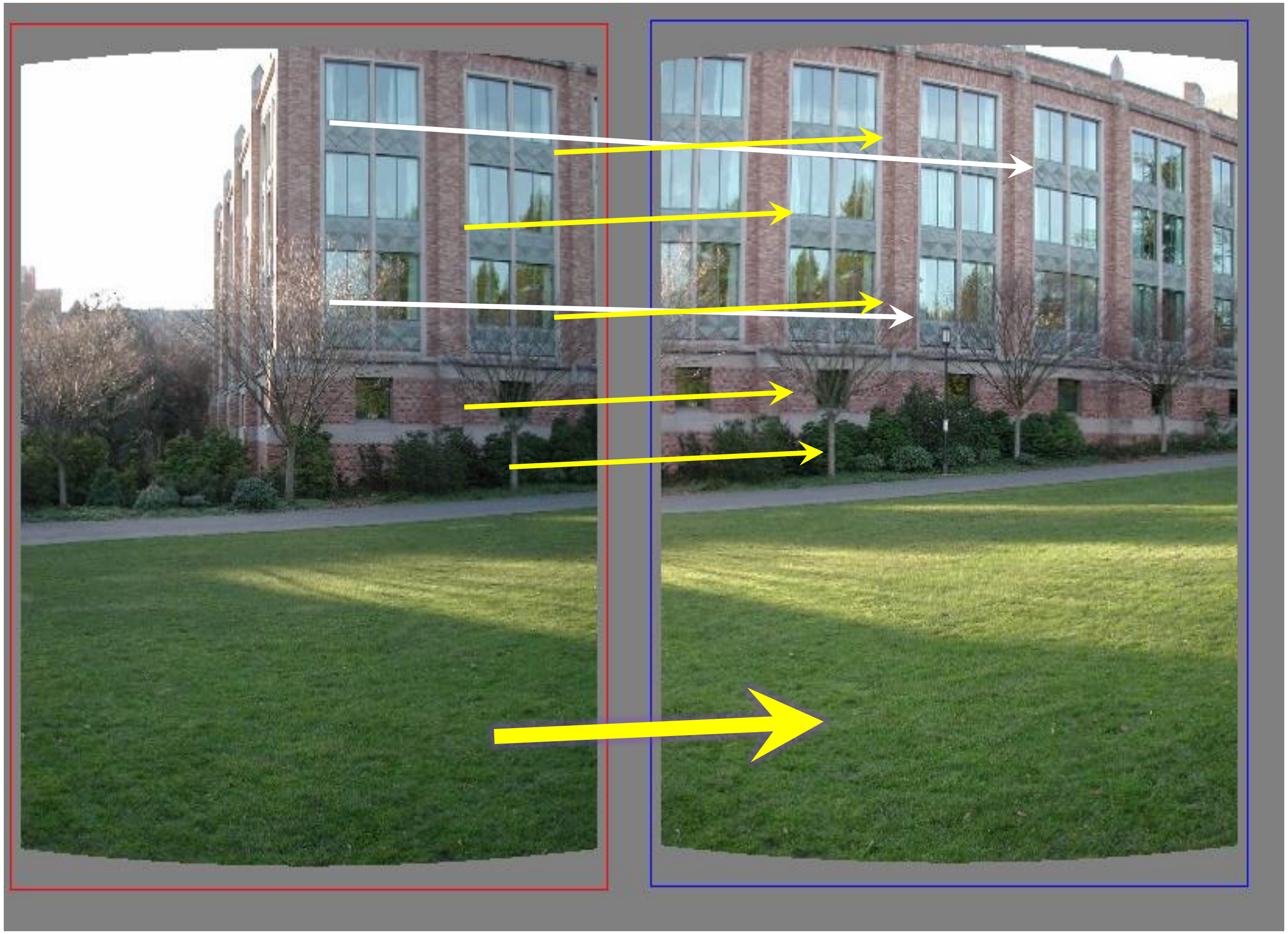
Select *one* match, count *inliers*
(in this case, only one)

RAndom SAmple Consensus



Select *one* match, count *inliers*
(4 inliers)

Least squares fit



Find "average" translation vector
for largest set of inliers

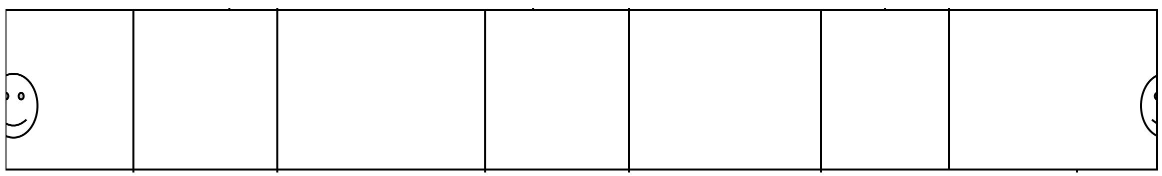
RANSAC

- Same basic approach works for any transformation
  - Translation, rotation, homographies, etc.
  - Very useful tool
- General version
  - Randomly choose a set of K correspondences
    - Typically K is the minimum size that lets you fit a model
  - Fit a model (e.g., homography) to those correspondences
  - Count the number of inliers that “approximately” fit the model
    - Need a threshold on the error
  - Repeat as many times as you can
  - Choose the model that has the largest set of inliers
  - Refine the model by doing a least squares fit using ALL of the inliers

Today

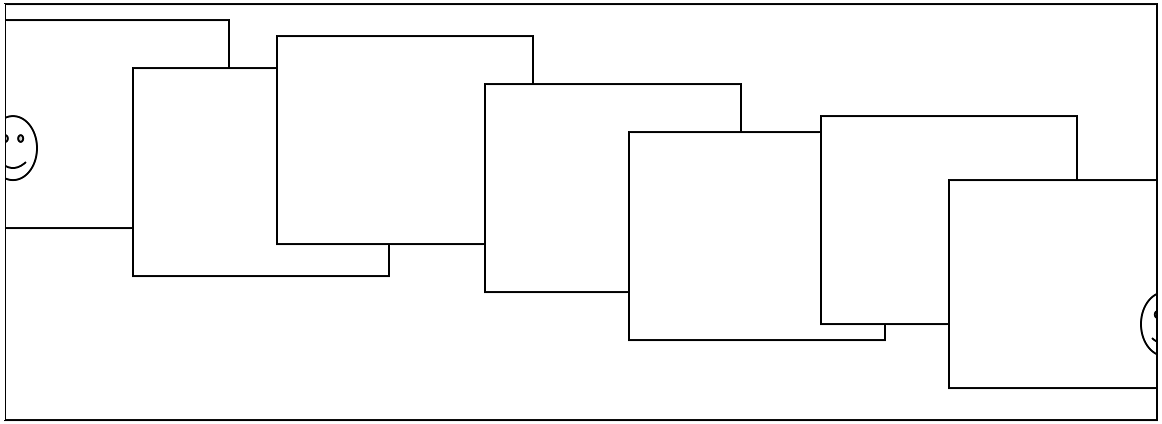
- Stitching Order and Transformation Order
- Blending
- Warping
- Project 3

Assembling the panorama



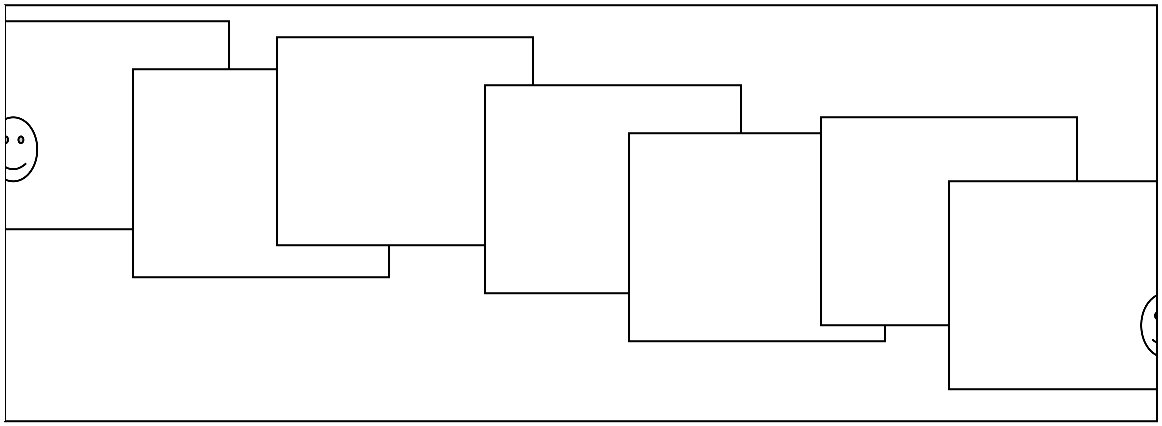
- Stitch pairs together, blend, then crop

Problem: Drift



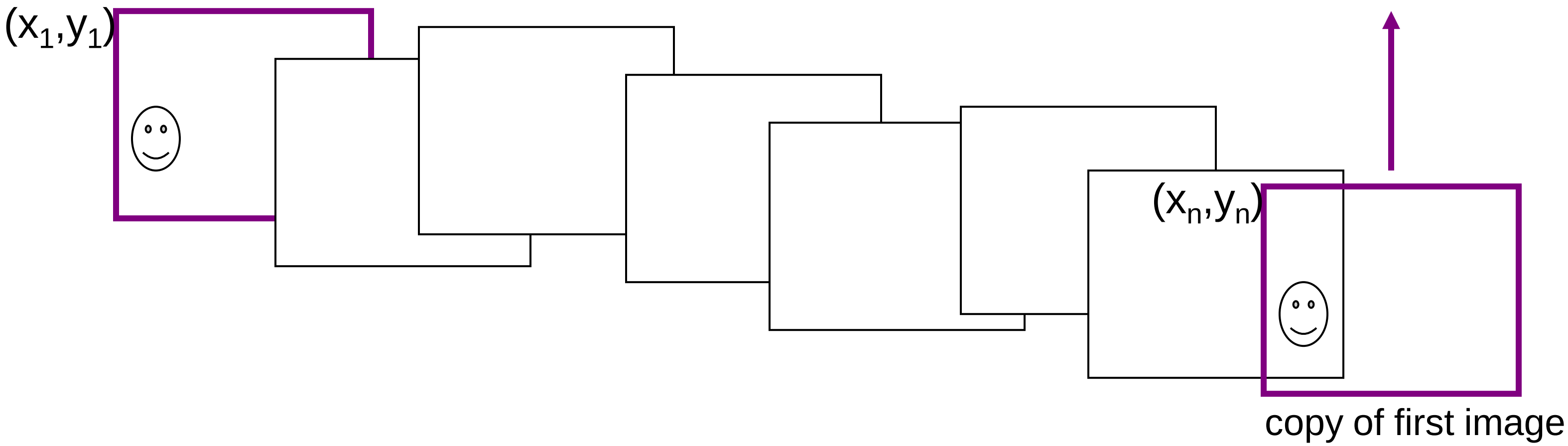
- Error accumulation
  - small errors accumulate over time

Simple Solution



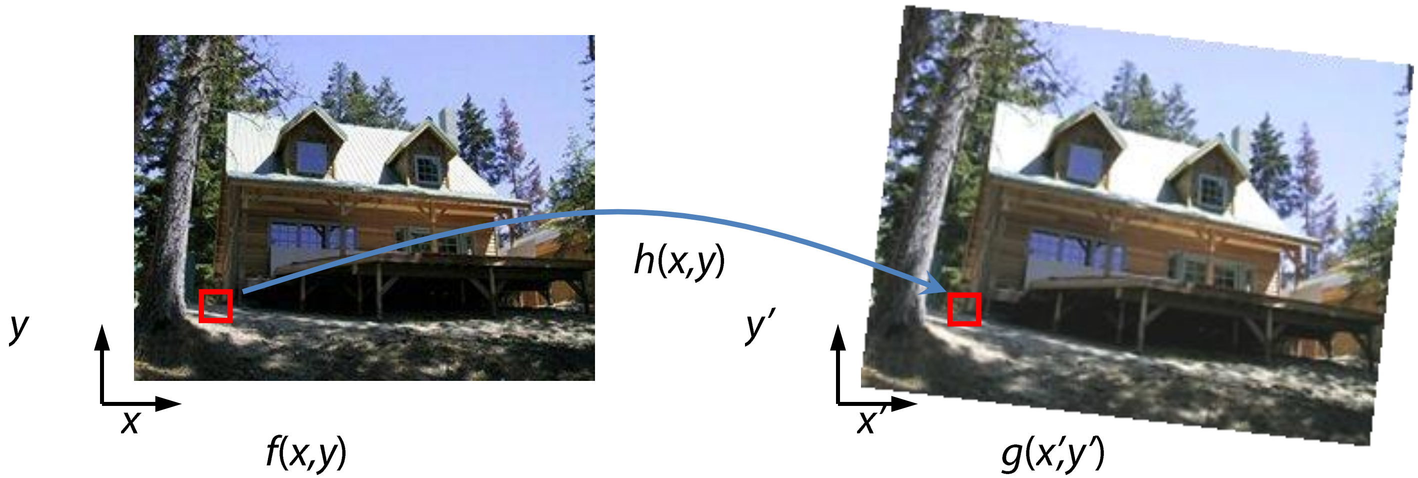
- Build out from the center
- Pick the center after an initial round of pairwise matching

Problem: Drift



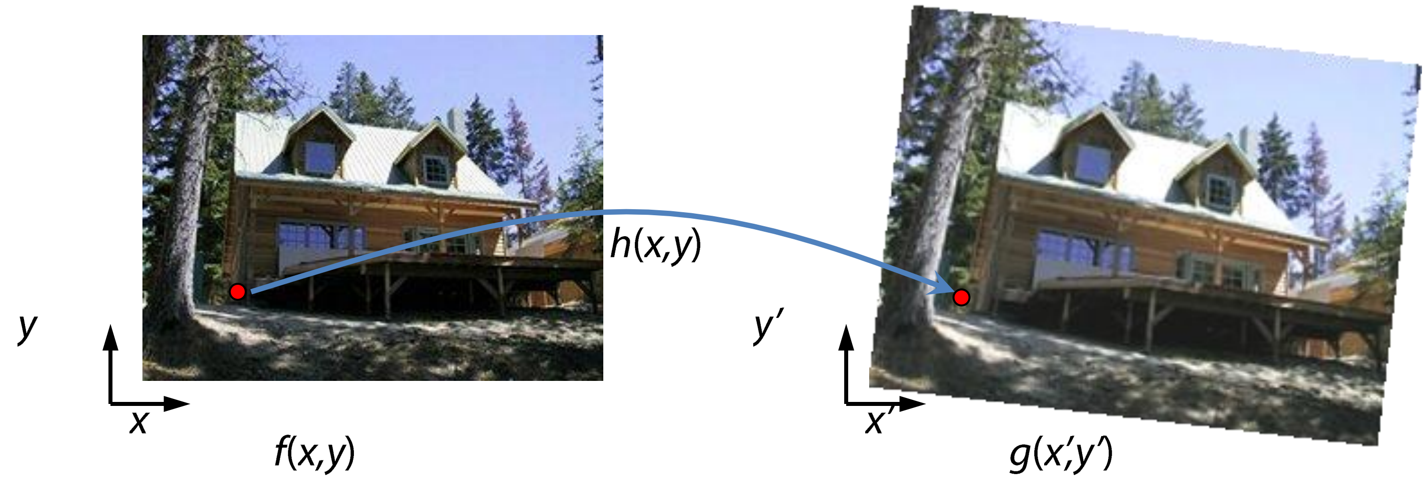
- Solution
  - add another copy of first image at the end
  - this gives a constraint: $y_n = y_1$
  - there are a bunch of ways to solve this problem
    - add displacement of $(y_1 - y_n)/(n - 1)$ to each image after the first
    - compute a global warp: $y' = y + ax$
    - run a big optimization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

Image warping



- Given a coordinate transform $(x',y') = h(x,y)$ and a source image $f(x,y)$, how do we compute a transformed image $g(x',y') = f(h(x,y))$?

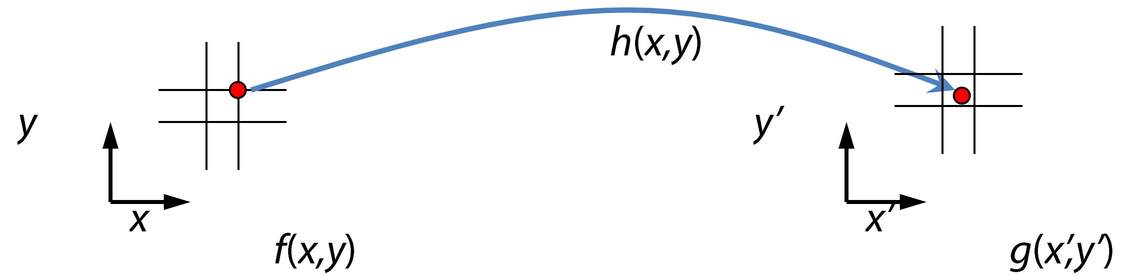
Forward warping



- Send each pixel $f(x,y)$ to its corresponding location
- $(x',y') = h(x,y)$ in the second image

Q: what if pixel lands "between" two pixels?

Forward warping



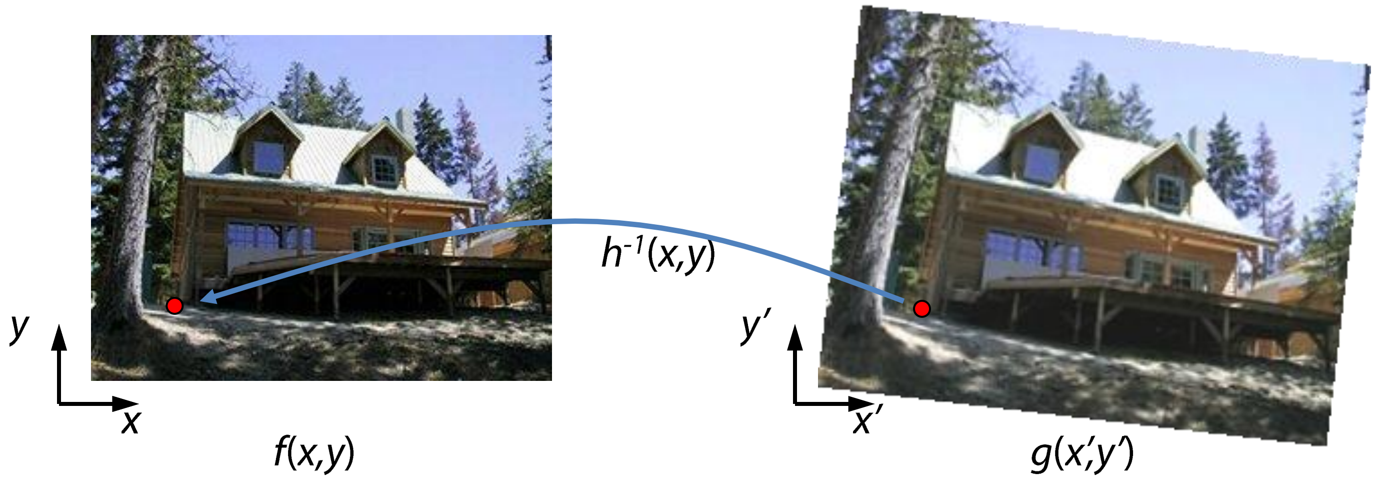
- Send each pixel $f(x,y)$ to its corresponding location
- $(x',y') = h(x,y)$ in the second image

Q: what if pixel lands "between" two pixels?

A: distribute color among neighboring pixels (x',y')

- Known as "splatting"

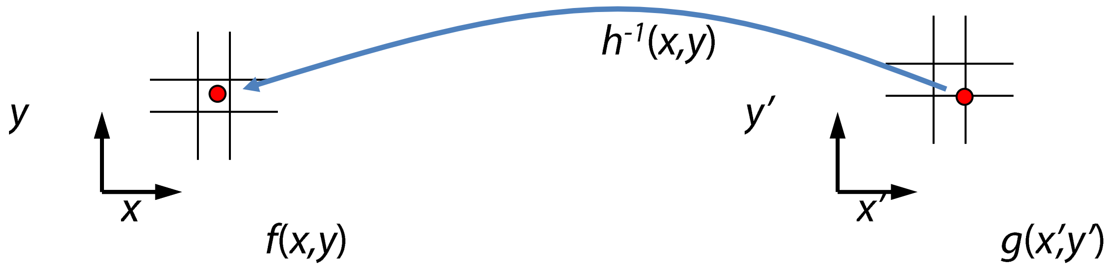
Inverse warping



- Get each pixel $g(x',y')$ from its corresponding location
- $(x,y) = h^{-1}(x',y')$ in the first image

Q: what if pixel comes from "between" two pixels?

Inverse warping



- Get each pixel $g(x',y')$ from its corresponding location
- $(x,y) = h^{-1}(x',y')$ in the first image

Q: what if pixel comes from “between” two pixels?

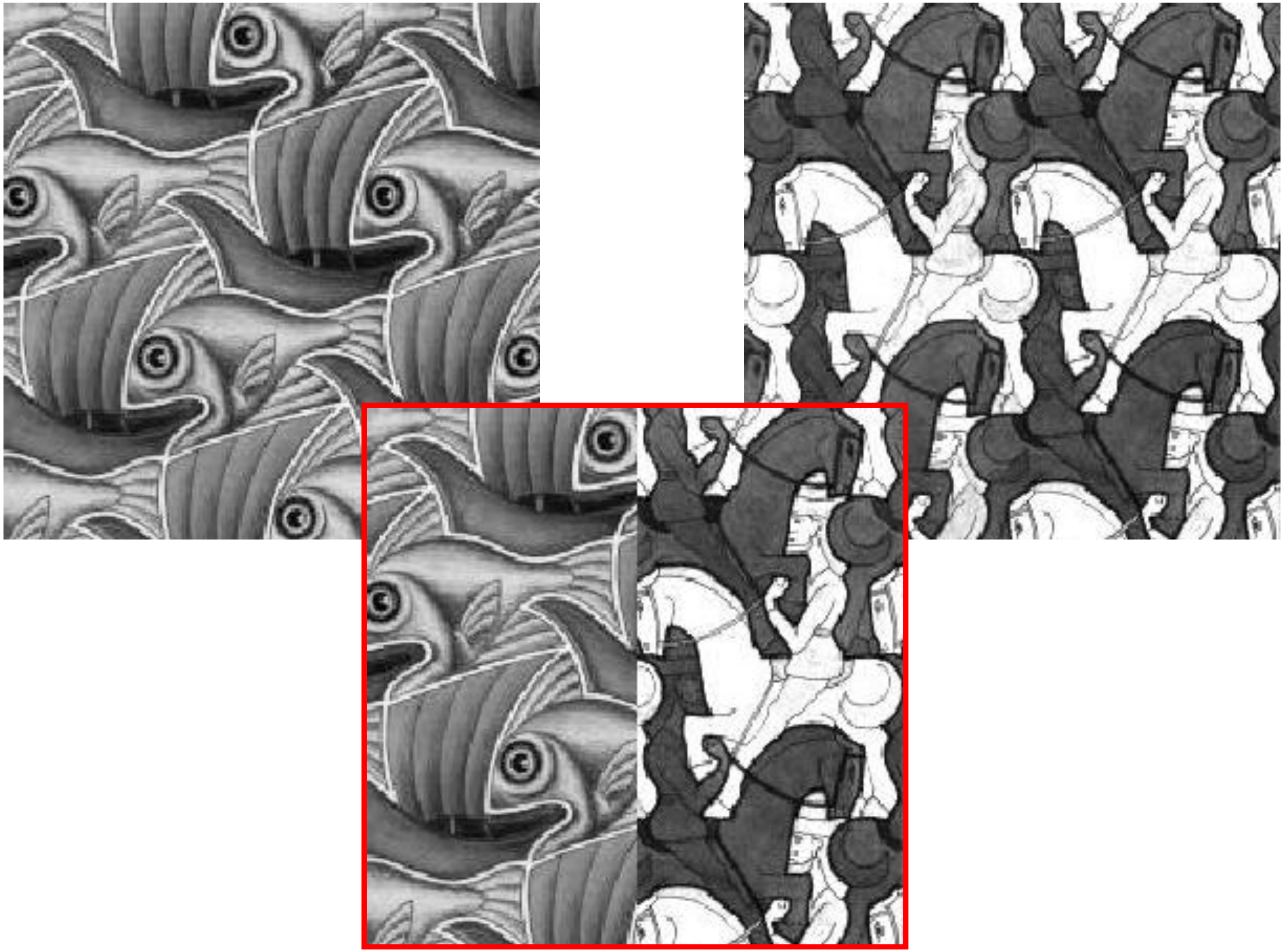
A: *resample* color value
  - We discussed resampling techniques before
    - nearest neighbor, bilinear, Gaussian, bicubic

Forward vs. inverse warping

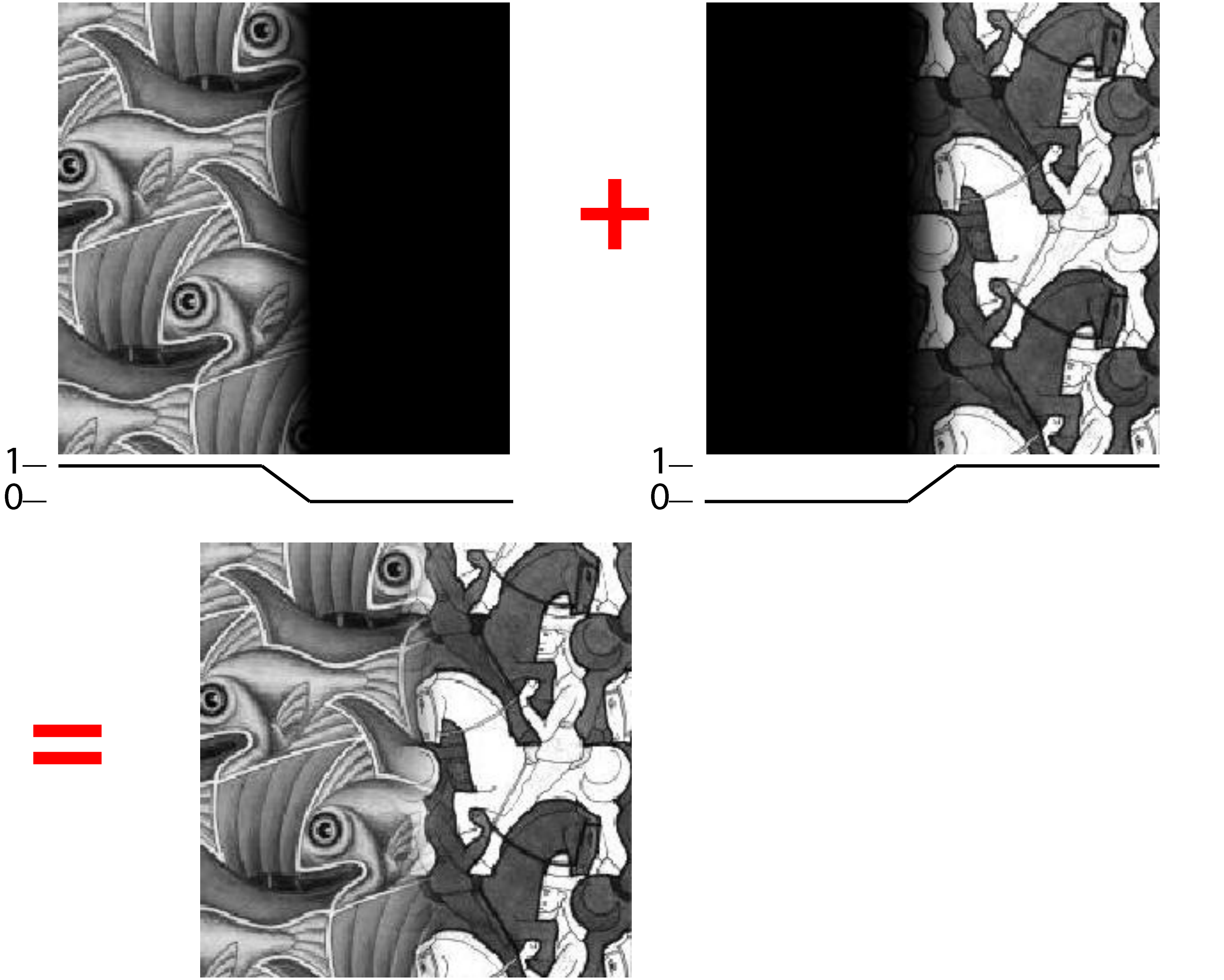
- Q: which is better?

- A: usually inverse—eliminates holes
  - however, it requires an invertible warp function—not always possible...

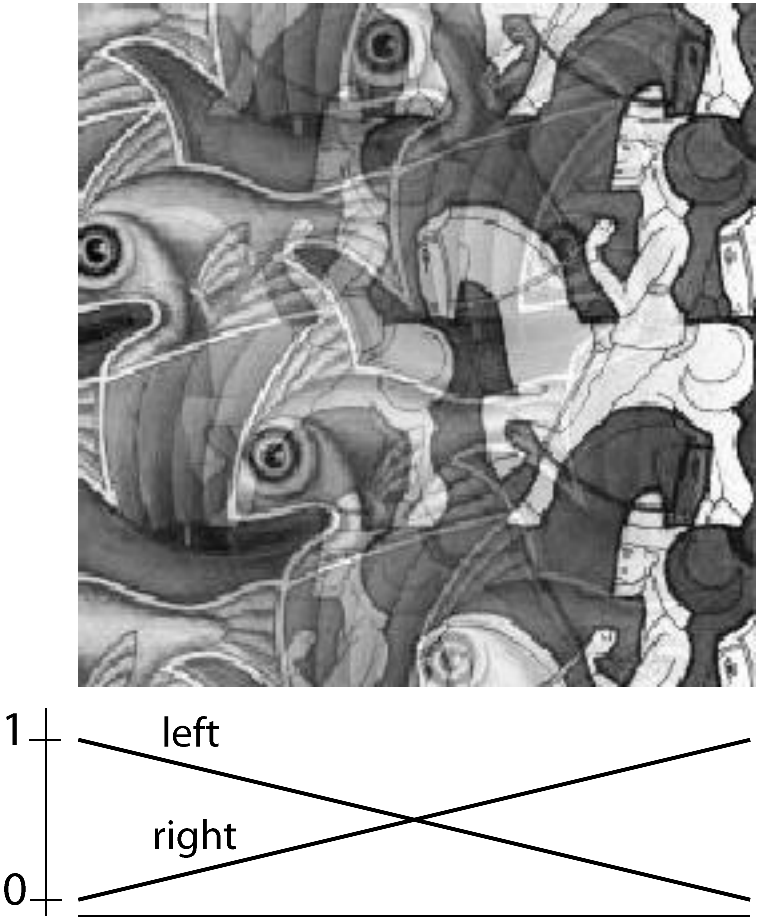
Image Blending

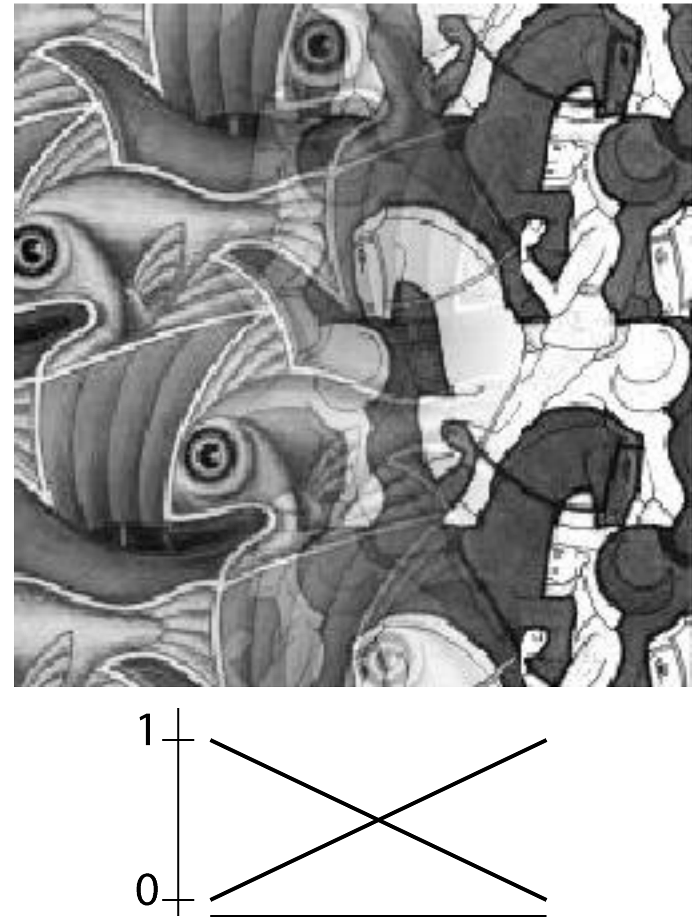


Feathering

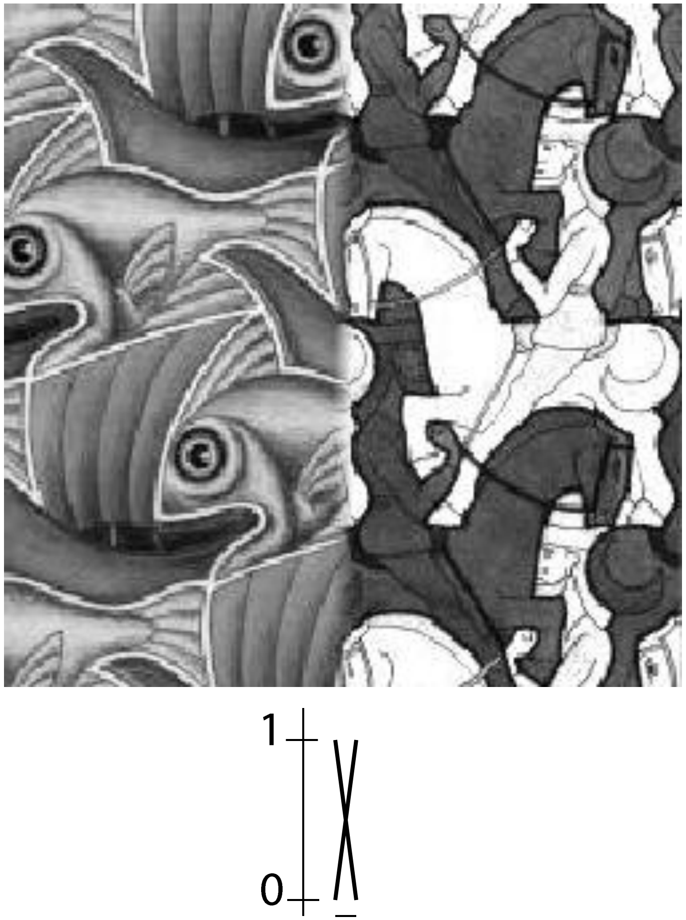


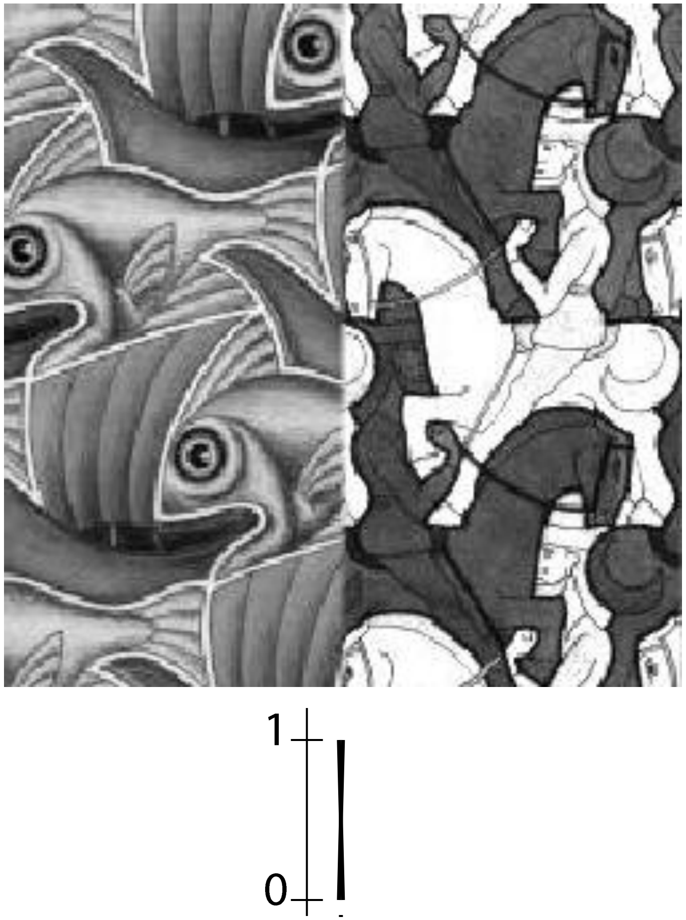
Effect of window size



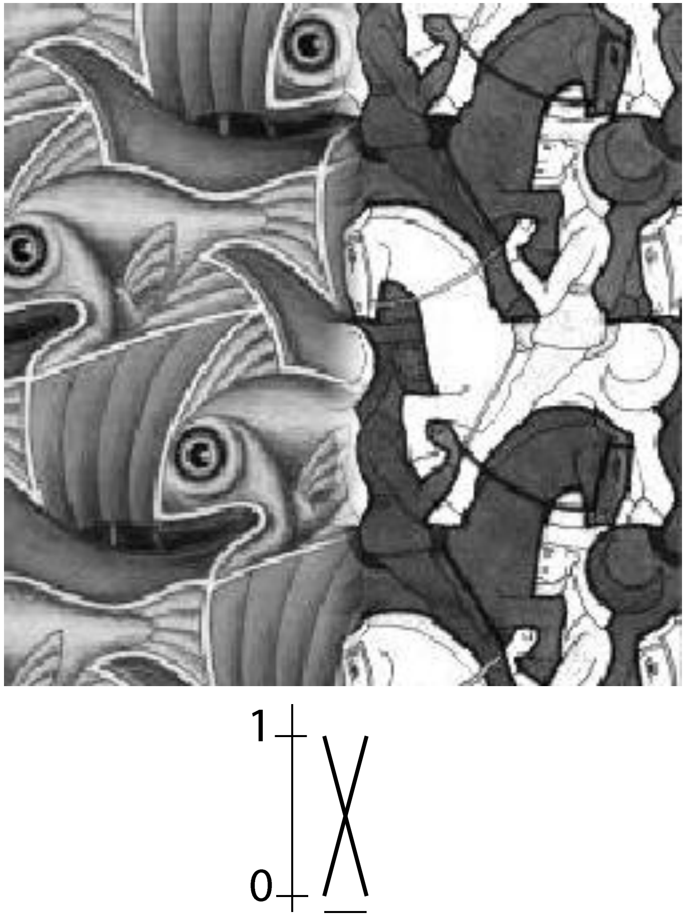


Effect of window size





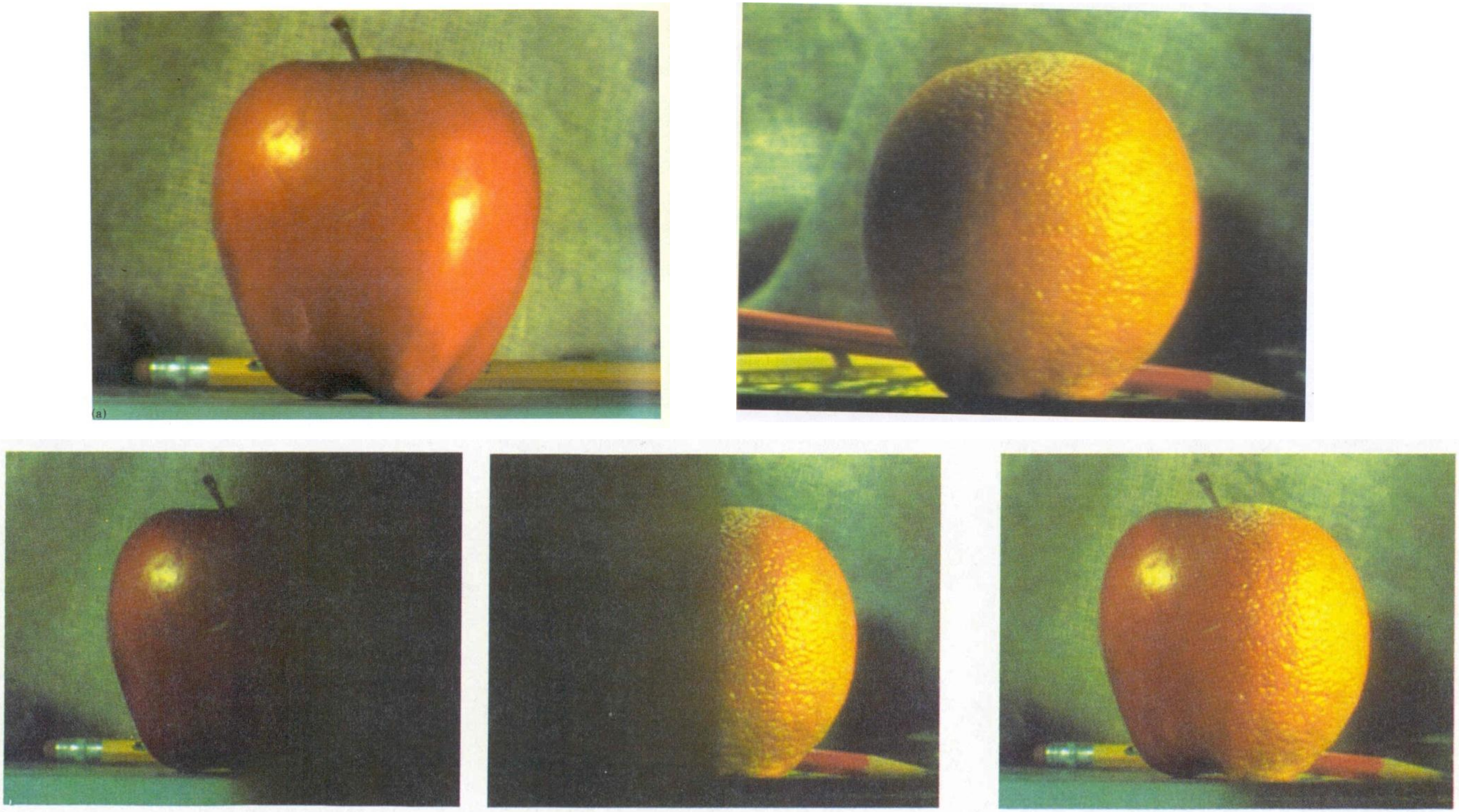
Good window size



"Optimal" window: smooth but not ghosted

- Doesn't always work...

Pyramid blending

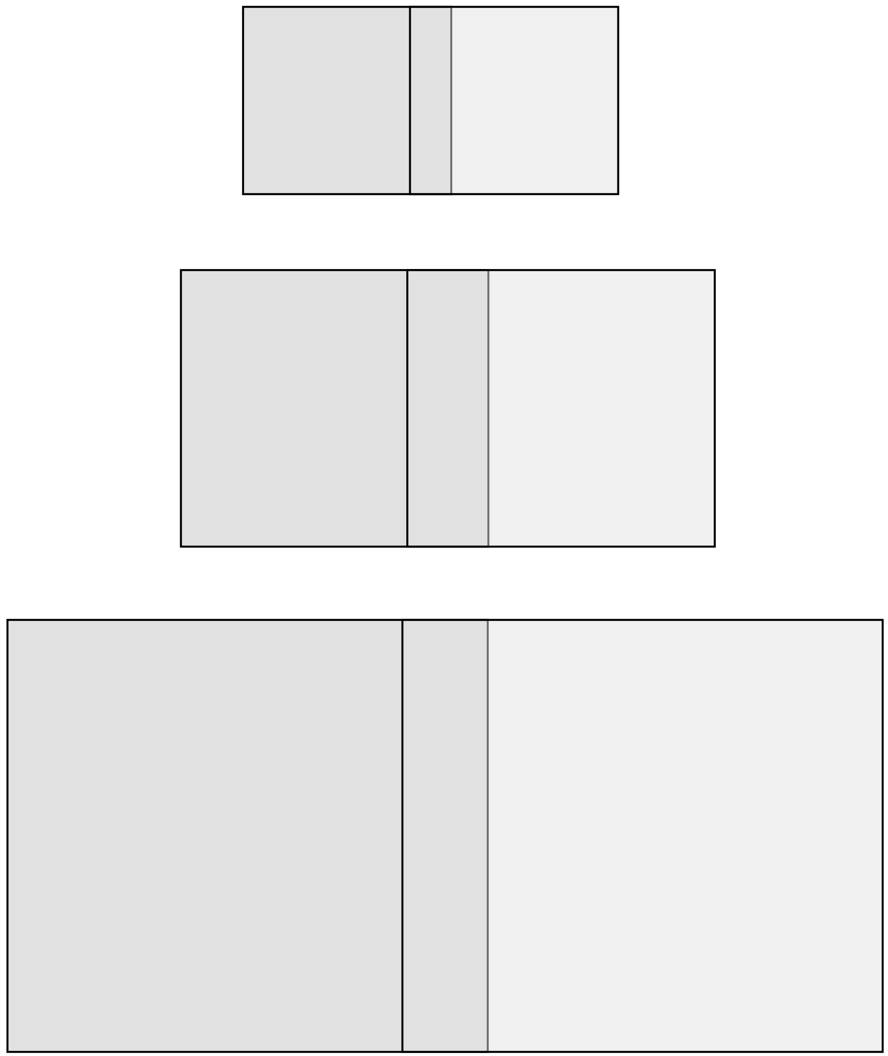


(d) (h) (l)

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.

Gaussian Blending



Poisson Image Editing

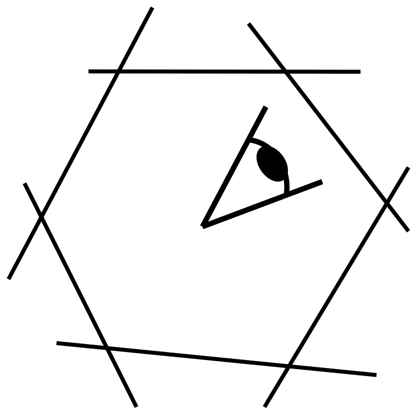
- For more info: Perez et al, SIGGRAPH 2003
  - http://research.microsoft.com/vision/cambridge/papers/perez_siggraph03.pdf

Poisson Image Editing/Blending

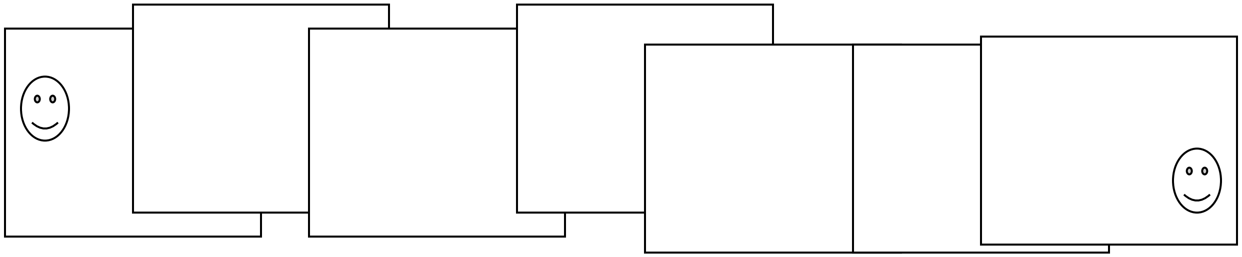
- Gradient Domain
- Smooths out color and intensity changes

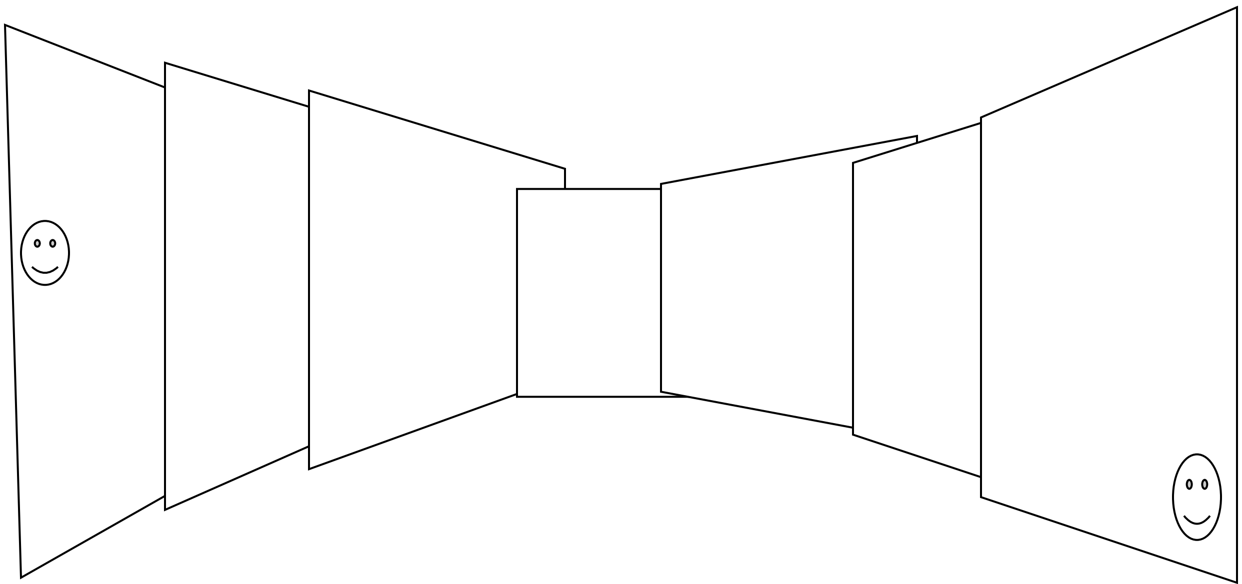
Panoramas

- What if you want a large field of view?



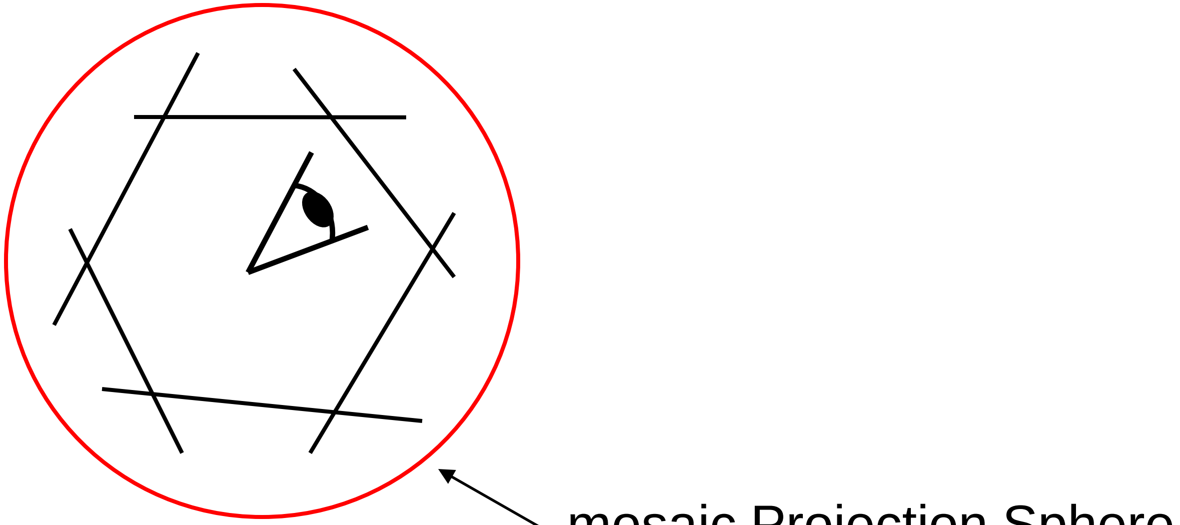
Large Field of View: Planar Projection





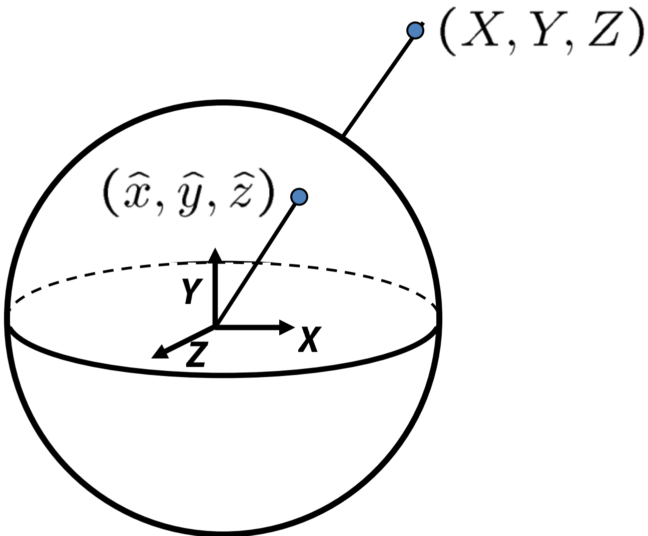
Panoramas

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



Examples

Spherical projection



unit sphere

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

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

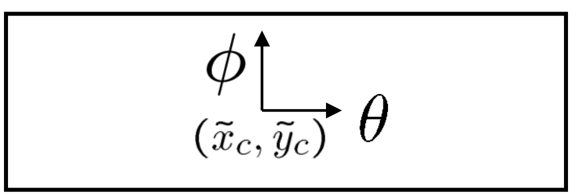
- Convert to spherical coordinates

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

- Convert to spherical image coordinates

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

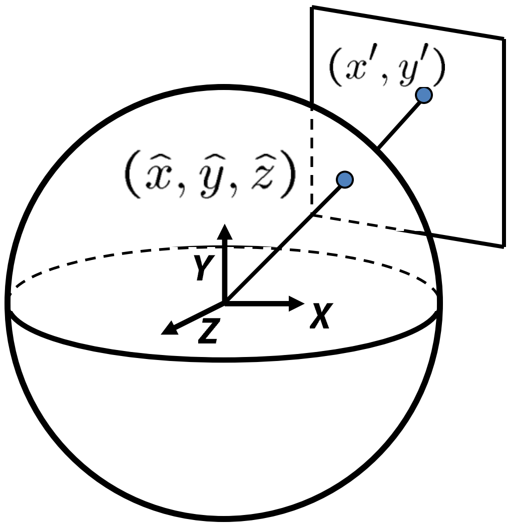
  – s defines size of the final image
    » Often set s = camera focal length



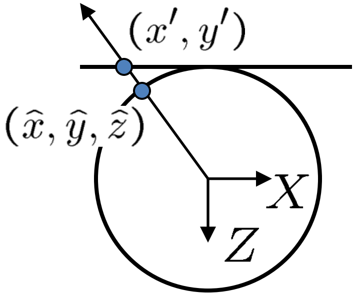
unwrapped sphere

Spherical image

Spherical reprojection



side view

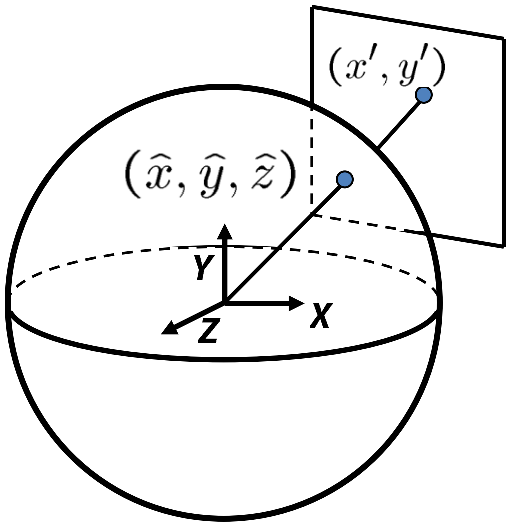


top-down view

How to map sphere onto a flat image?

$(\hat{x}, \hat{y}, \hat{z})$ to (x', y')

Spherical reprojection

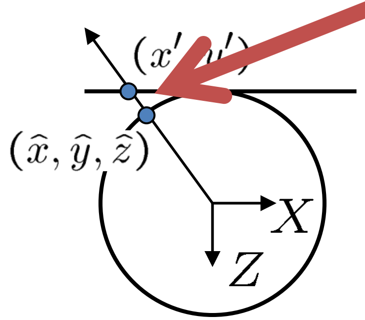


side view

How to map sphere onto a flat image?

- $(\hat{x}, \hat{y}, \hat{z})$ to (x', y')
  - Use image projection matrix!
    - or use the version of projection that properly accounts for radial distortion, as discussed in projection slides. This is what you'll do for project 3.

Project the point along the radius



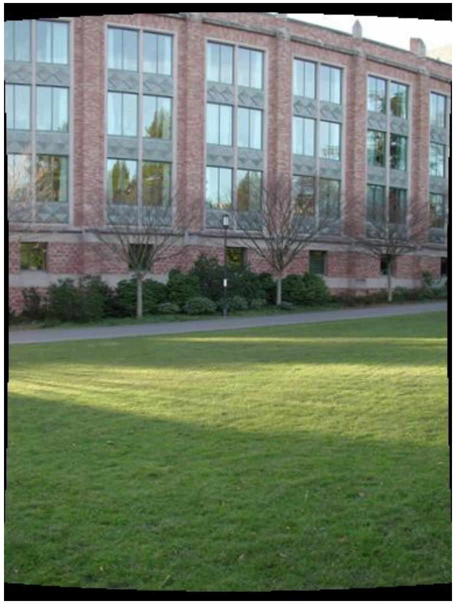
top-down view

Spherical reprojection, before stiching

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?

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

Spherical image stitching

- What if you don't know the camera rotation?
  - Solve for the camera rotations
    - Note that a pan (rotation) of the camera is a **translation** of the sphere!
    - Use feature matching to solve for translations of spherical-warped images

Different projections are possible

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…

Project 3

- In pairs
- Check out a Panorama Kit