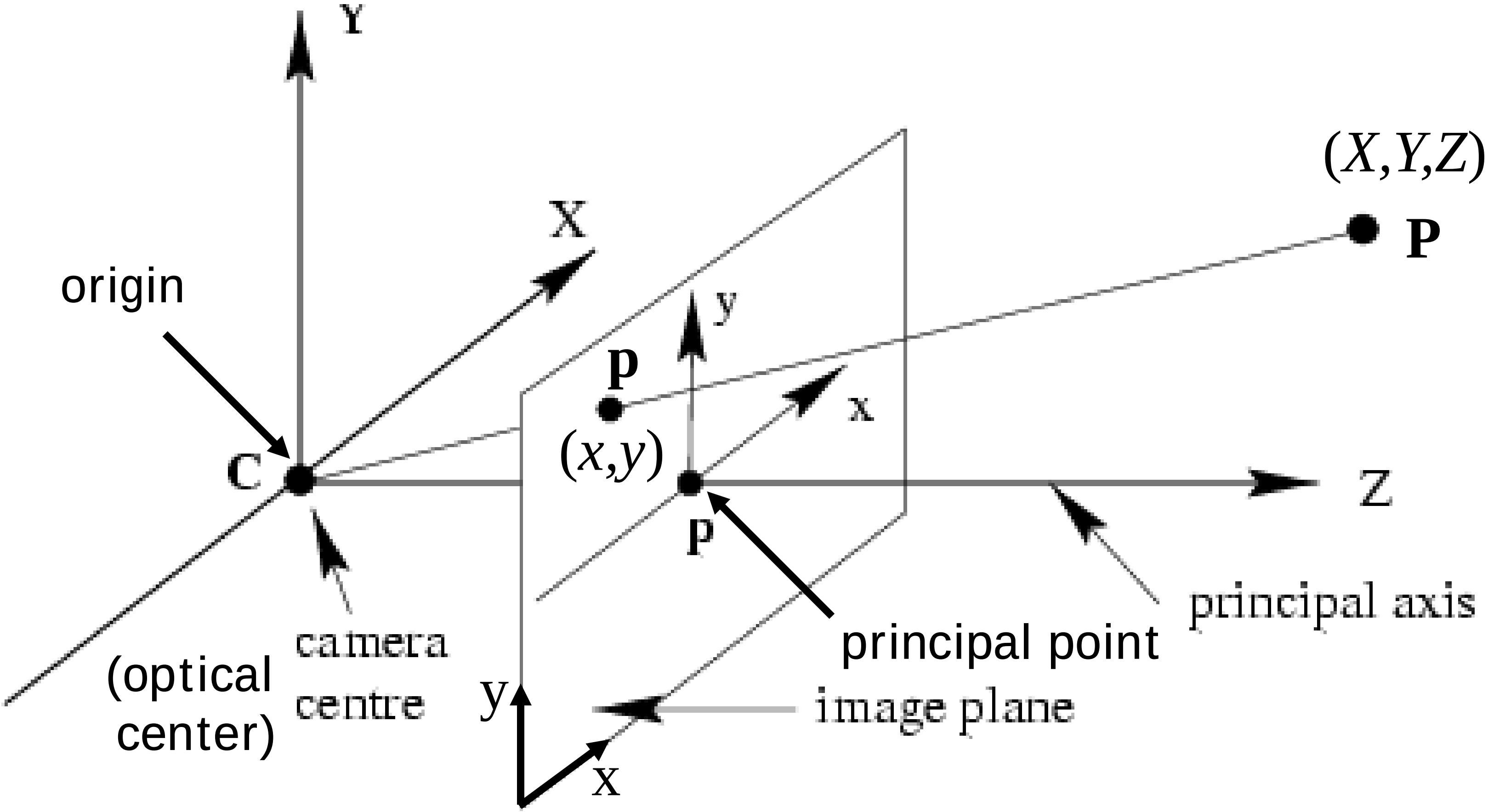
Last Lecture



$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \sim \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

Today

Image Mosaics and Panorama

- Today's Readings
  - Szeliski and Shum paper
    http://www.acm.org/pubs/citations/proceedings/graph/258734/p251-szeliski/

Full screen panoramas (cubic): http://www.panoramas.dk/
Mars: http://www.panoramas.dk/fullscreen3/f2_mars97.html
2003 New Years Eve: http://www.panoramas.dk/fullscreen3/f1.html

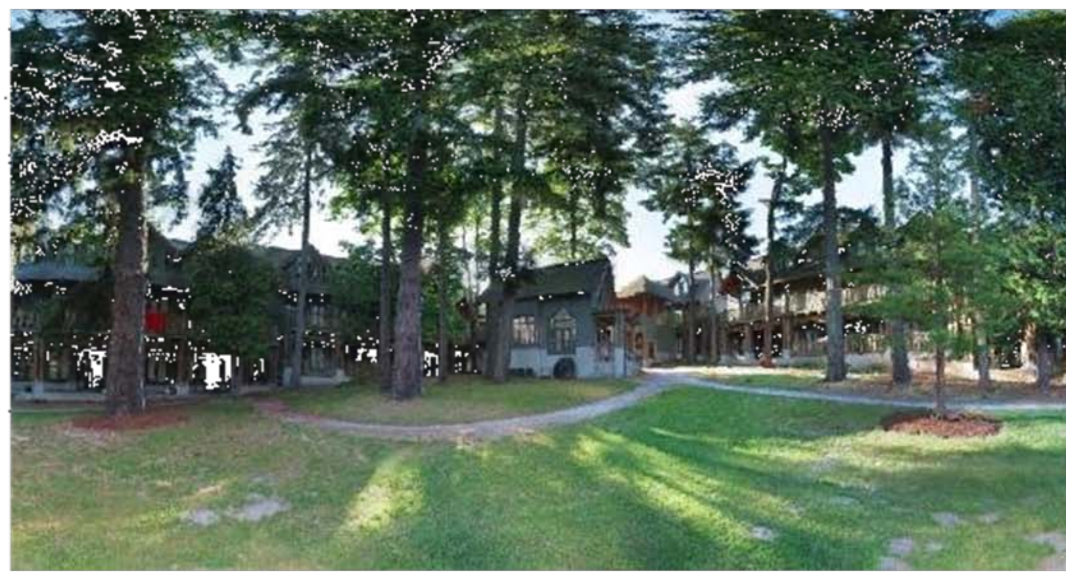
Why Mosaic?

- Are you getting the whole picture?
  - Compact Camera FOV = 50 x 35°

Slide from Brown & Lowe

Why Mosaic?

- Are you getting the whole picture?
  - Compact Camera FOV = 50 x 35°
  - Human FOV = 200 x 135°



Slide from Brown & Lowe

Why Mosaic?

- Are you getting the whole picture?
  - Compact Camera FOV = 50 x 35°
  - Human FOV = 200 x 135°
  - Panoramic Mosaic = 360 x 180°

Slide from Brown & Lowe

Mosaics: stitching images together

Creating virtual wide-angle camera

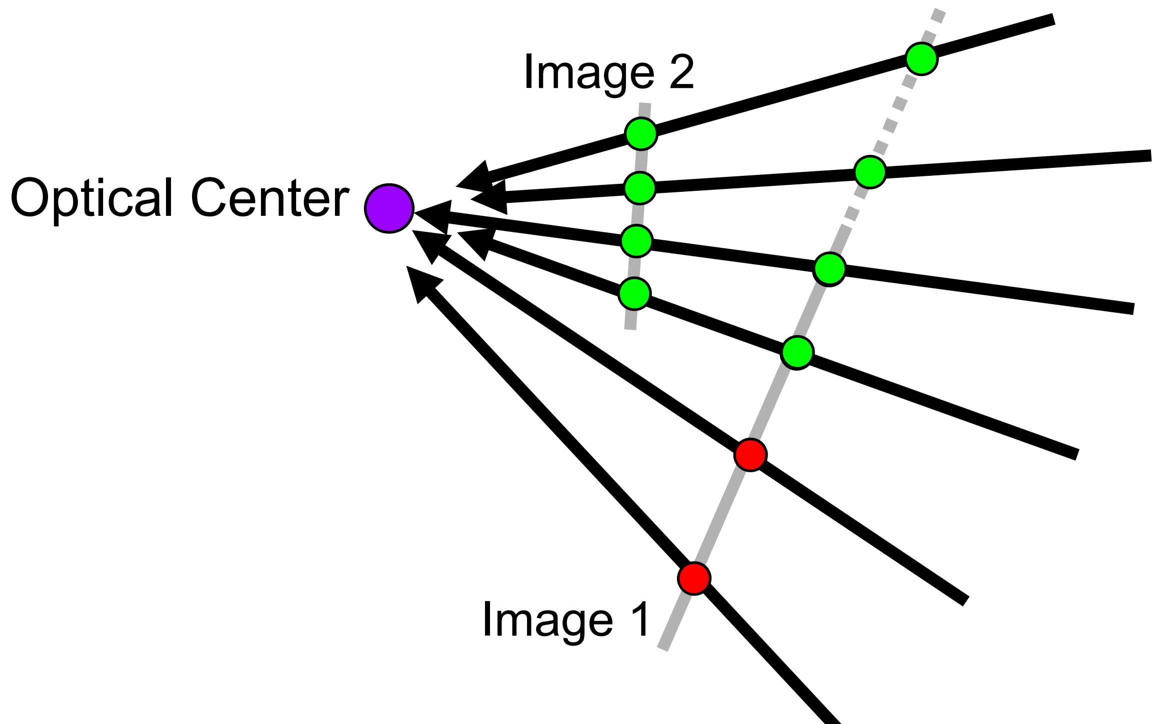
Auto Stitch: the State of Art Method

- Demo
- Project 2 is a striped-down AutoStitch

How to do it?

- 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 the 360º rays in different images, we can assemble them into a panorama.
- The basic operation is projecting an image from one plane to another
- The projective transformation is scene-INDEPENDENT

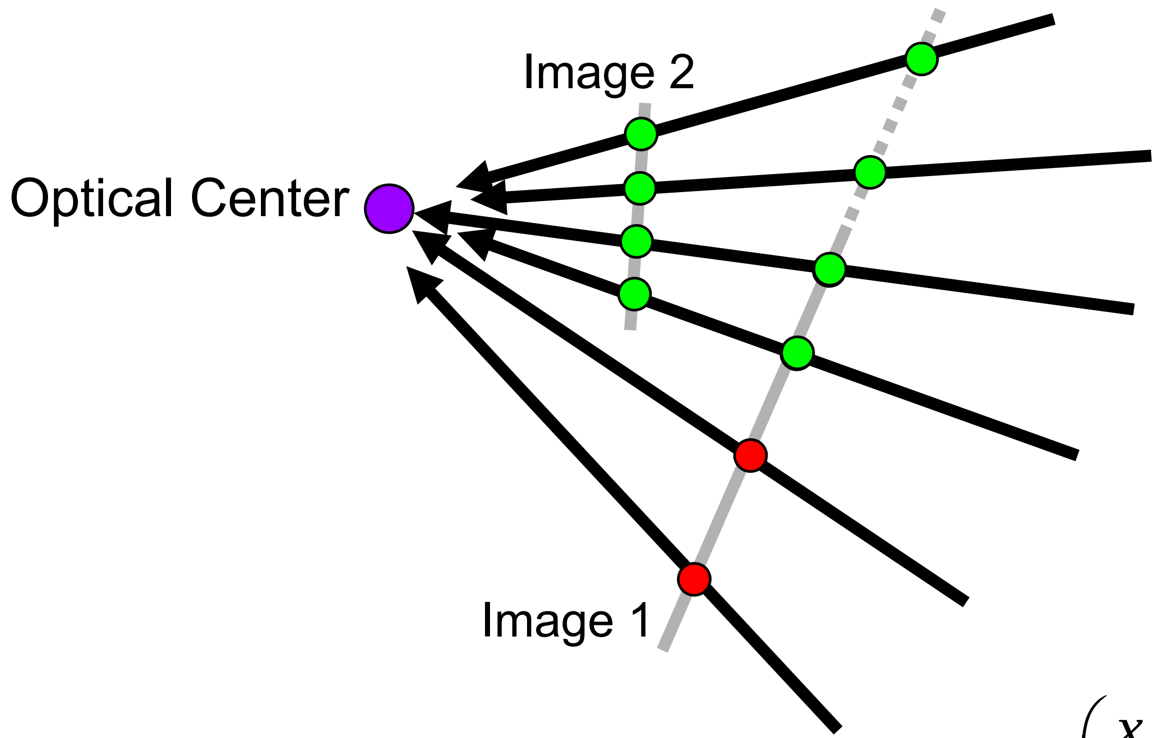
What is the transformation?

left on top

right on top

Translations are not enough to align the images

What is the transformation?



$$\begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix} \sim \mathbf{K}_1 \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

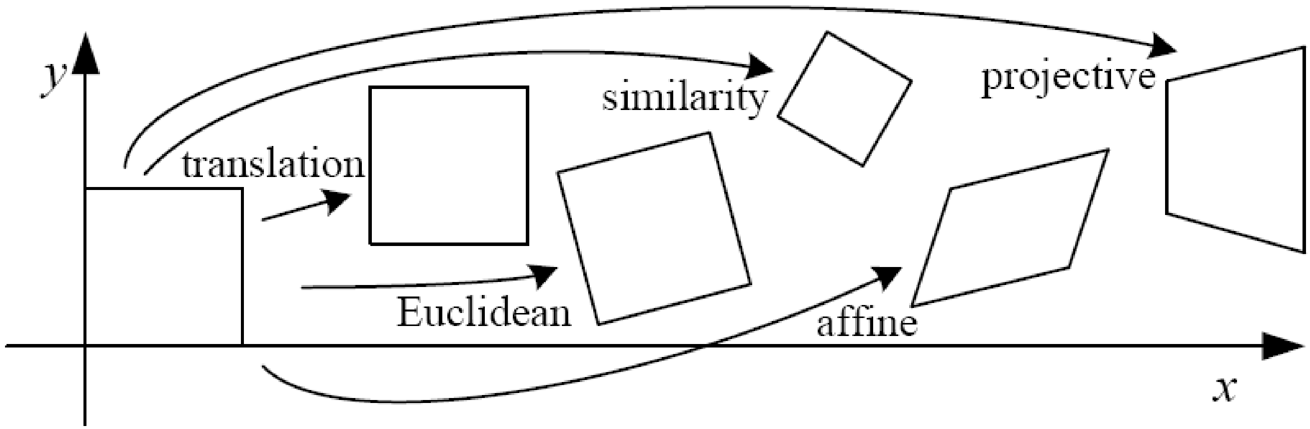
$$\begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} \sim \mathbf{K}_2\mathbf{R} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

$$\begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} \sim \mathbf{K}_2\mathbf{R}\mathbf{K}_1^{-1} \begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix}$$

3x3 matrix
also called Homography

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \sim \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & 0 & x_0 \\ 0 & f & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

Recall in the Image Warping Lecture:



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

Image warping with homographies

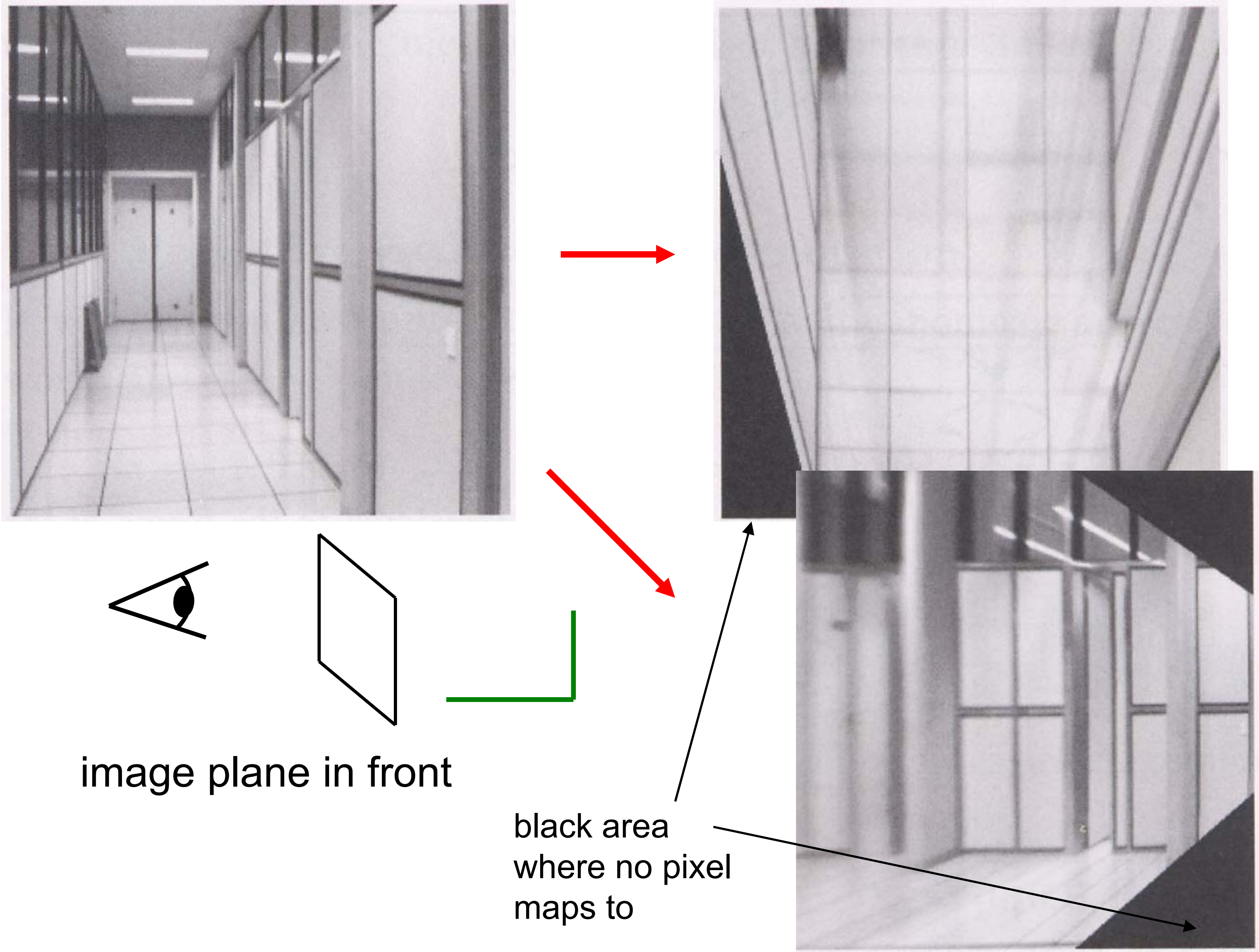
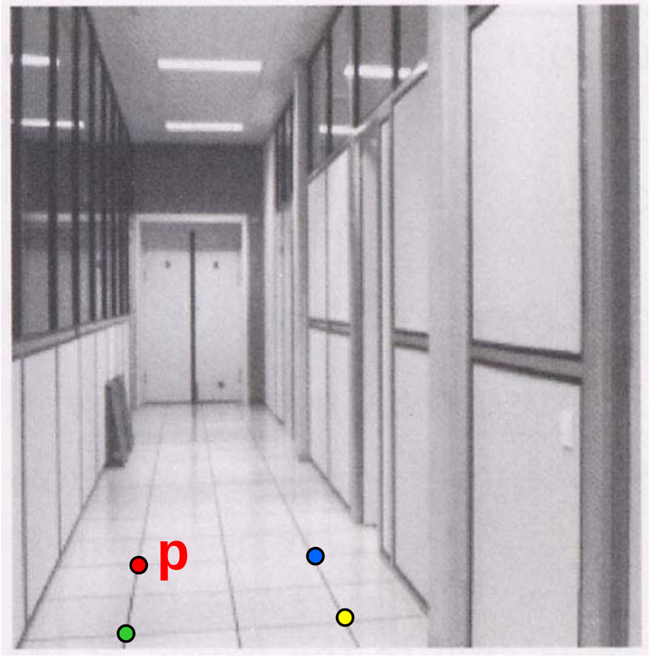
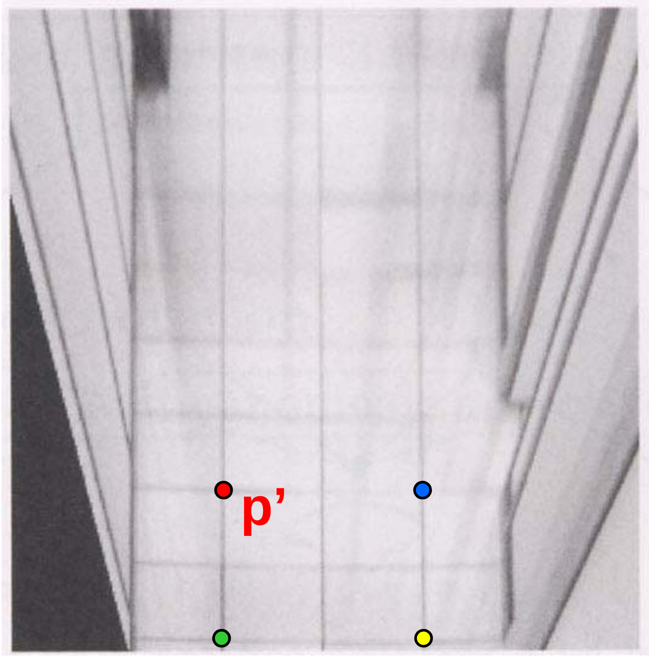


Image rectification





To unwarp (rectify) an image

- Find the homography **H** given a set of **p** and **p'** pairs
- How many correspondences are needed?

Solving for homographies

p’ = Hp

$$\begin{bmatrix} wx' \\ wy' \\ w \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

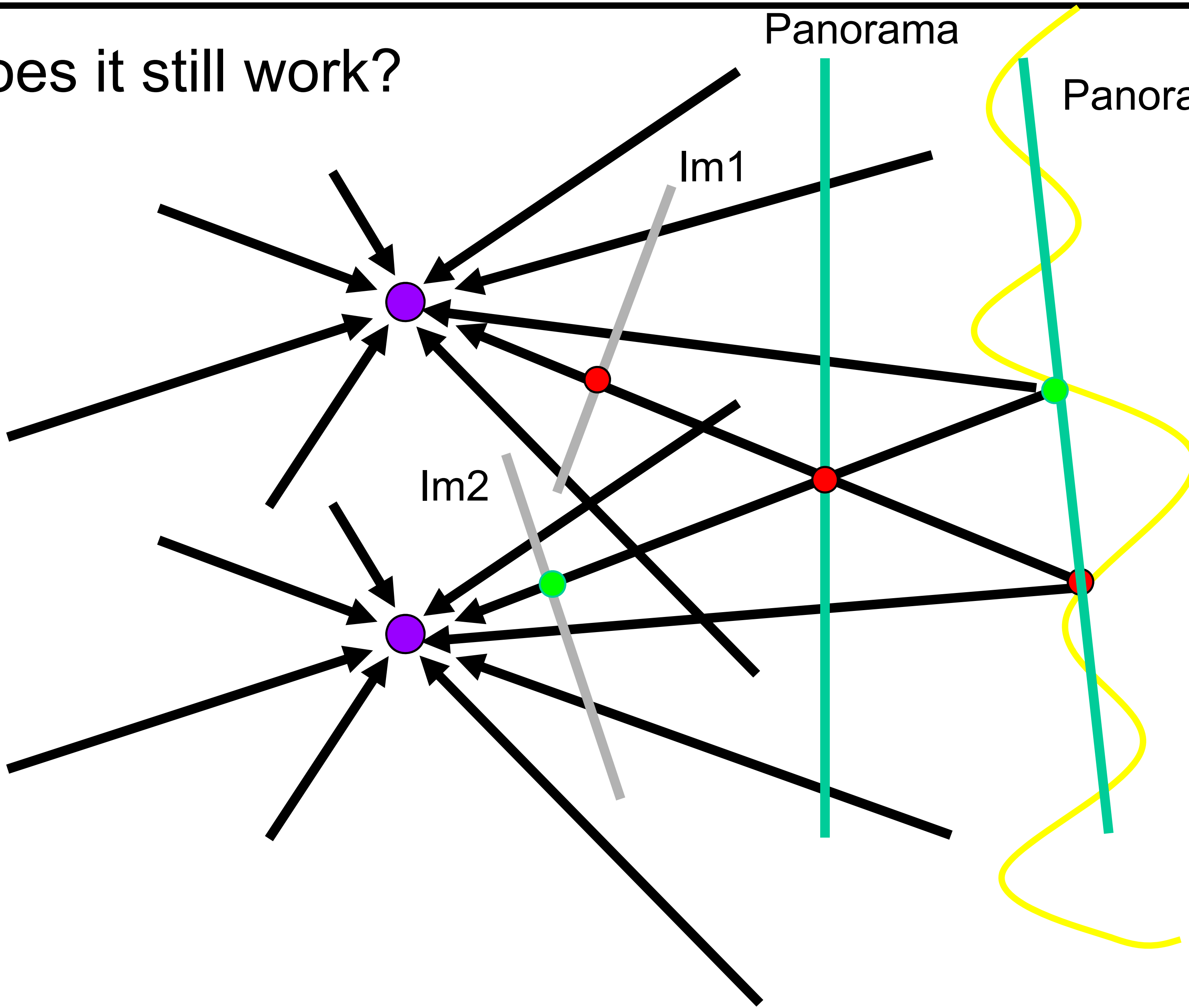
- Can set scale factor *i*=1. So, there are 8 unkowns.
- Set up a system of linear equations:
  - **Ah = b**
- where vector of unknowns h = $[a,b,c,d,e,f,g,h]^T$
- Need at least 8 eqs, but the more the better…
- Solve for h. If overconstrained, solve using least-squares:

$$\min \left\| Ah - b \right\|^2$$

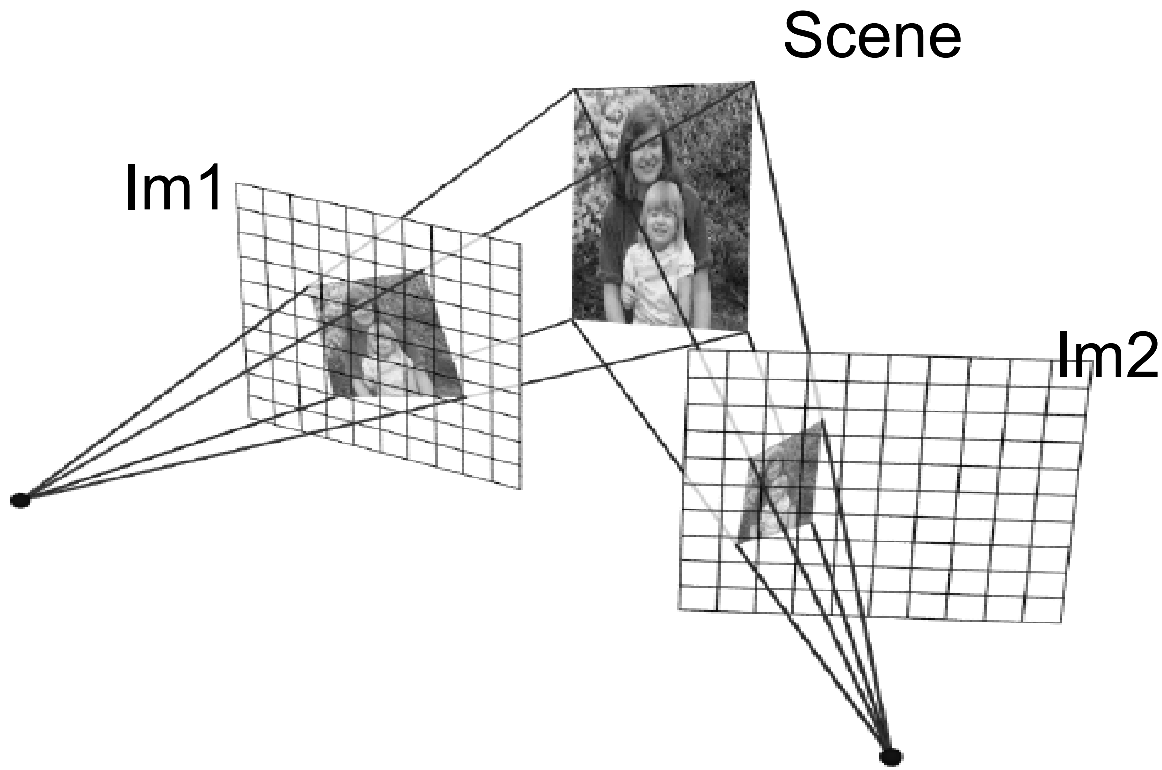
- Can be done in Matlab using “\” command
  - see “help lmdivide”

changing camera center

- Does it still work?

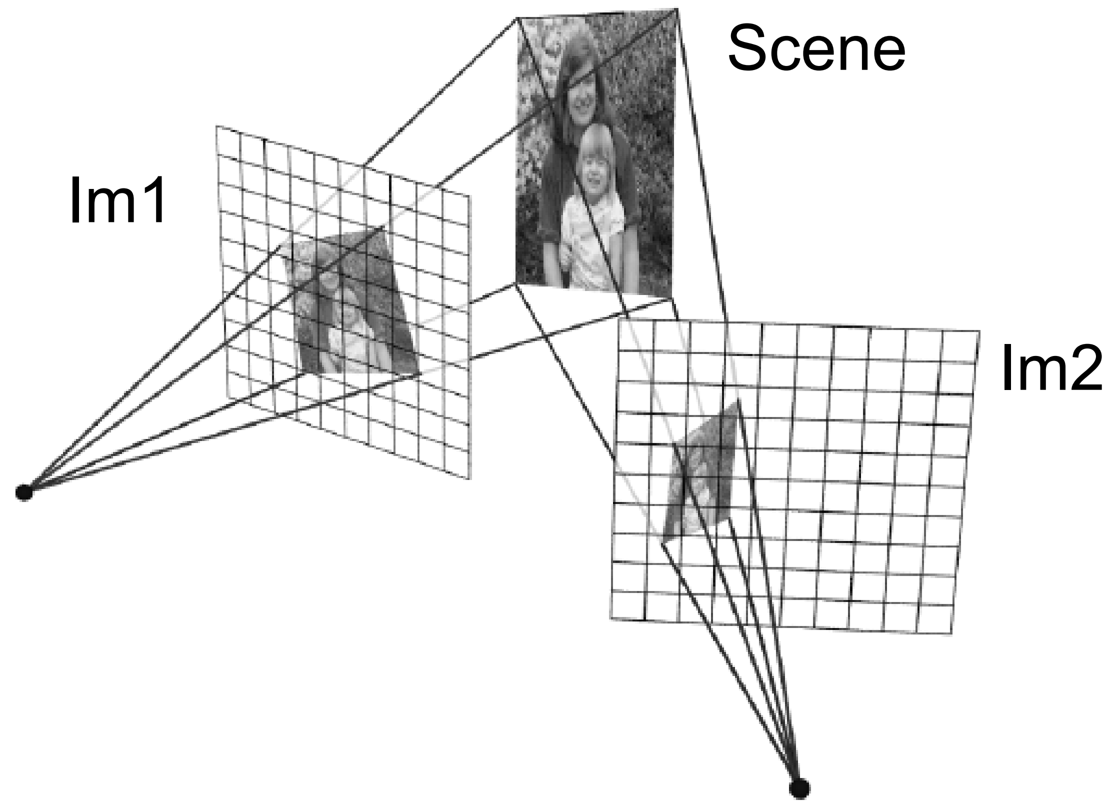


Planar scene (or far away)

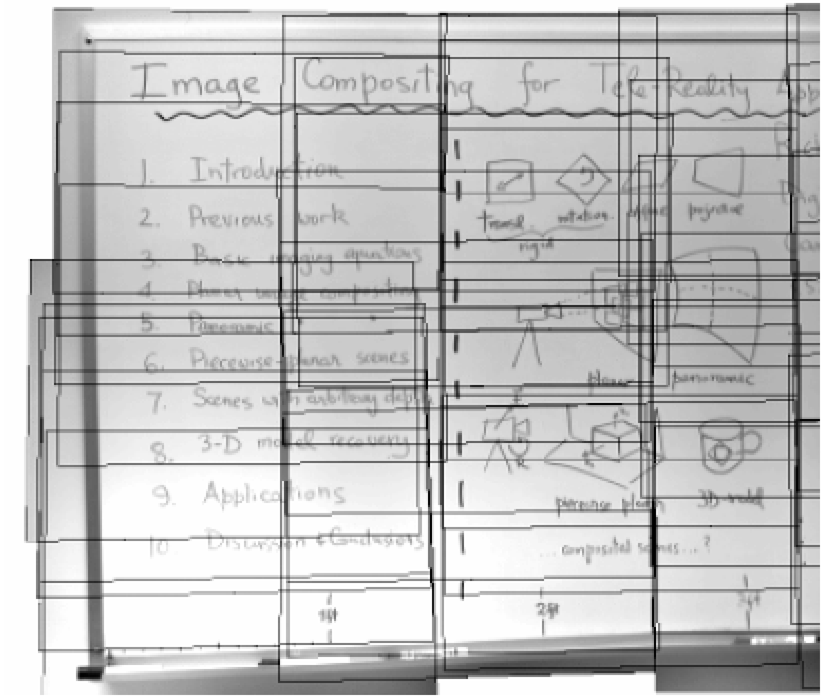


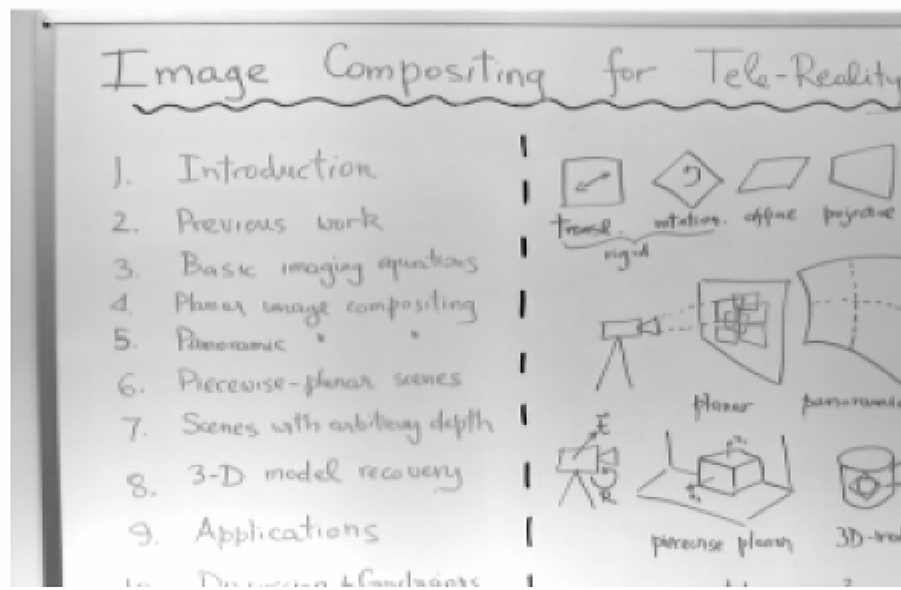
- If scene is planar, we are OK!
- This is how big aerial photographs are made

Why is so?



Planar mosaic Examples

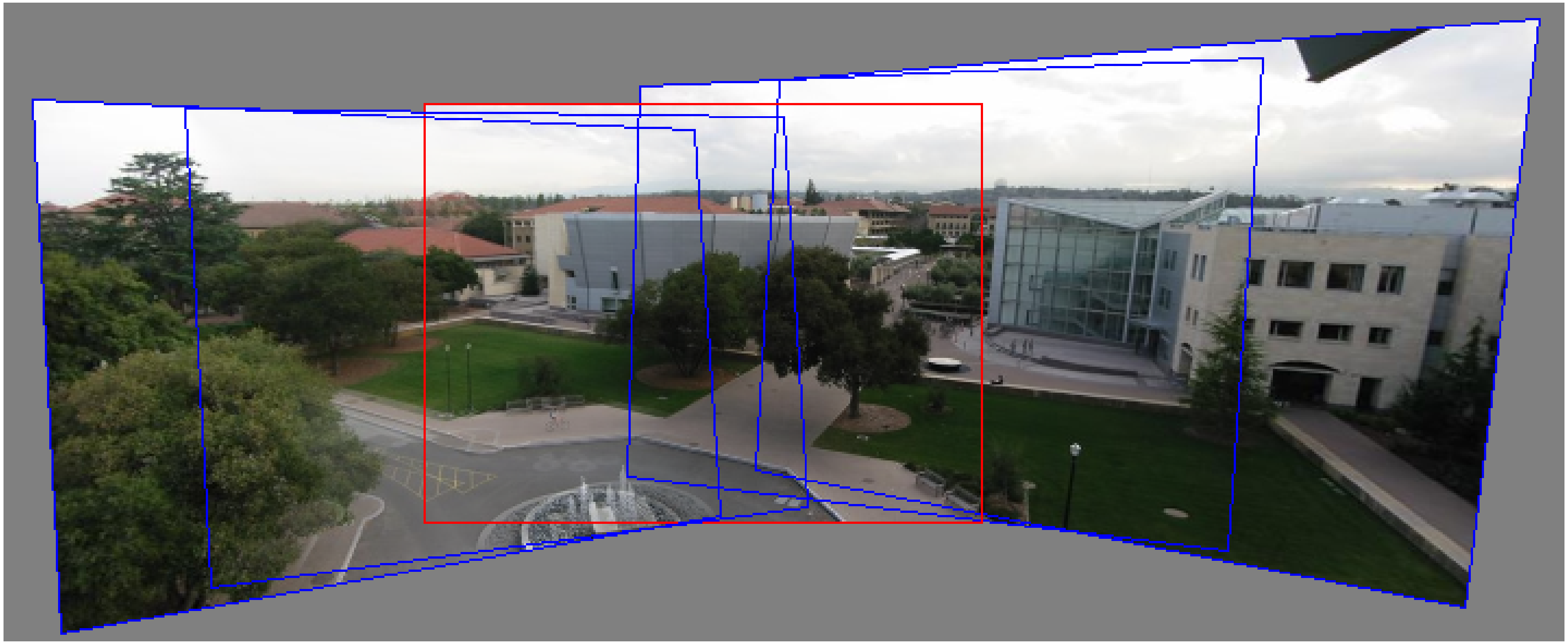




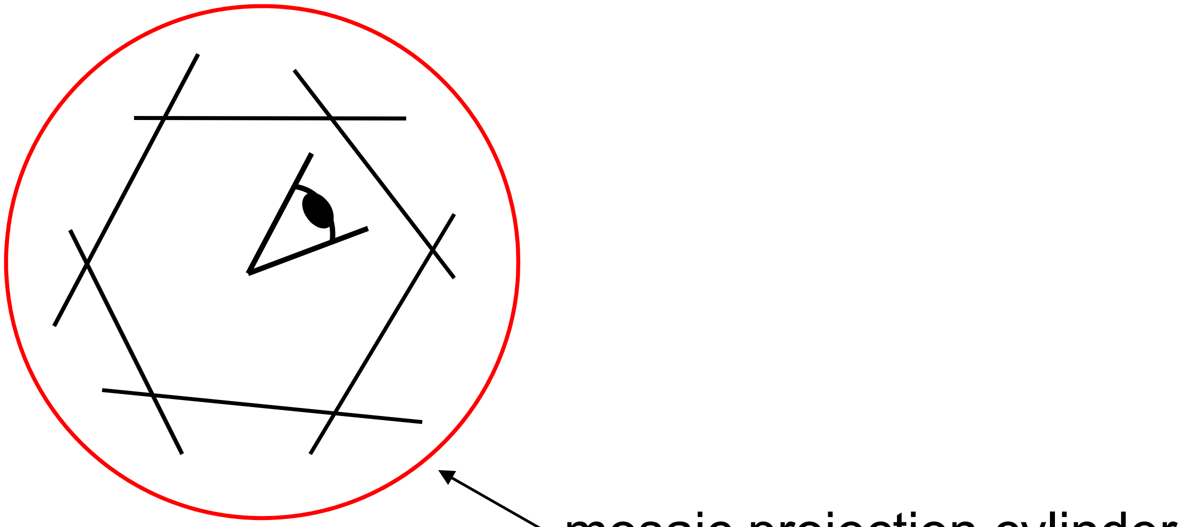
Recap:

- With enough images from the same optical center, we can create panorama.
- If the camera moves, we can’t in general
- If the scene is planar or faraway, we are OK.

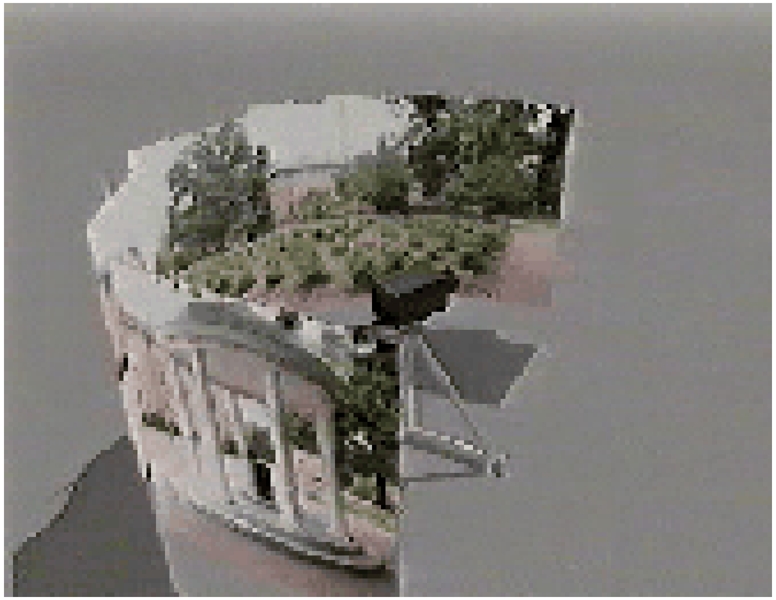
Can we use homography to create a 360 panorama?



Should use Cylindrical Projection



Cylindrical panoramas



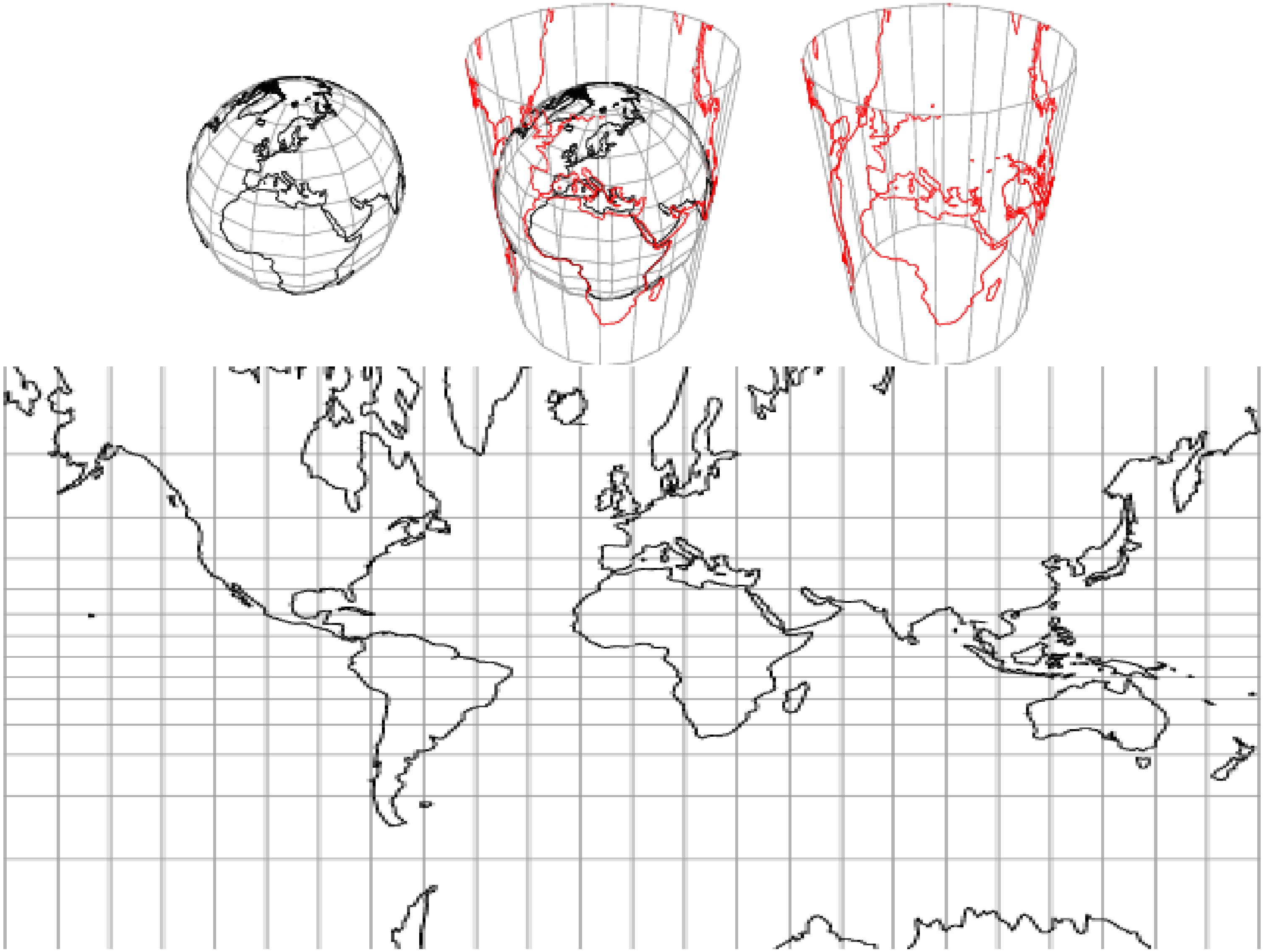
- Steps
  - Reproject each image onto a cylinder
  - Align and Blend
  - Output the resulting mosaic

Taking pictures

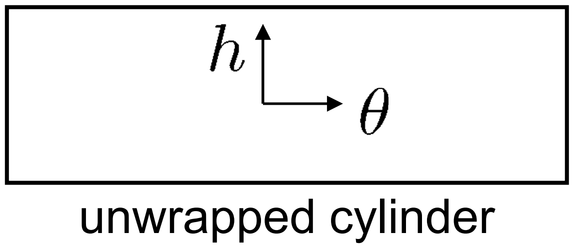
Kaidan panoramic tripod head

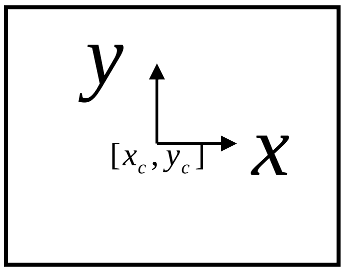
Warped Images

Cylindrical projection (An Example)

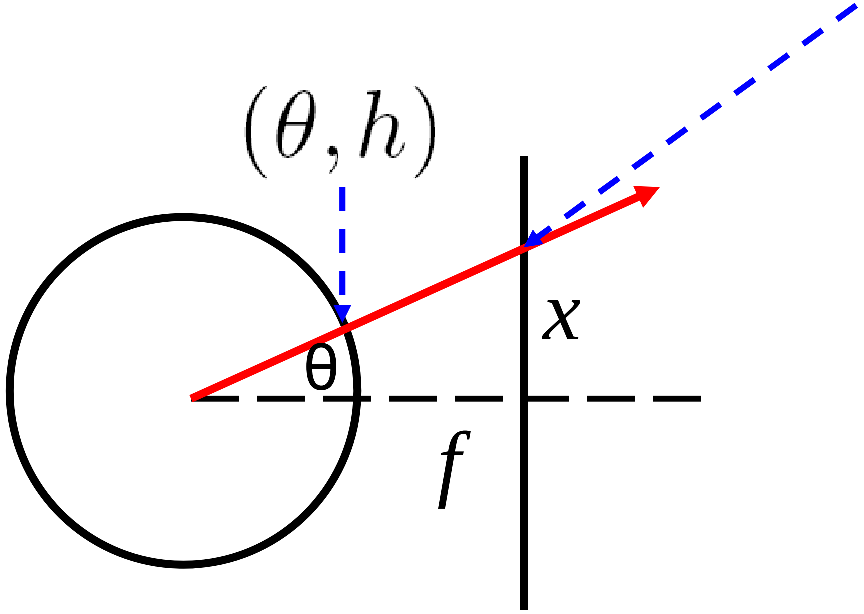


Cylindrical projection



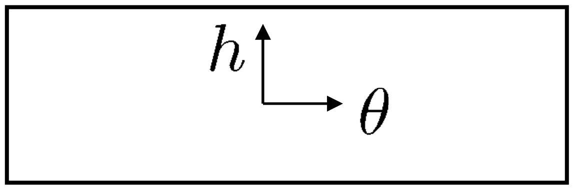


$$(\sin\theta, h, \cos\theta) \propto (x, y, f)$$

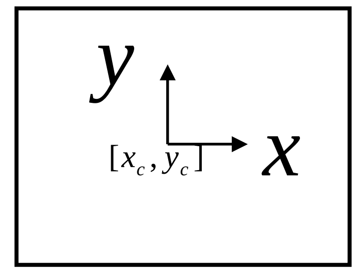


$$\theta = \tan^{-1}\frac{x}{f}$$

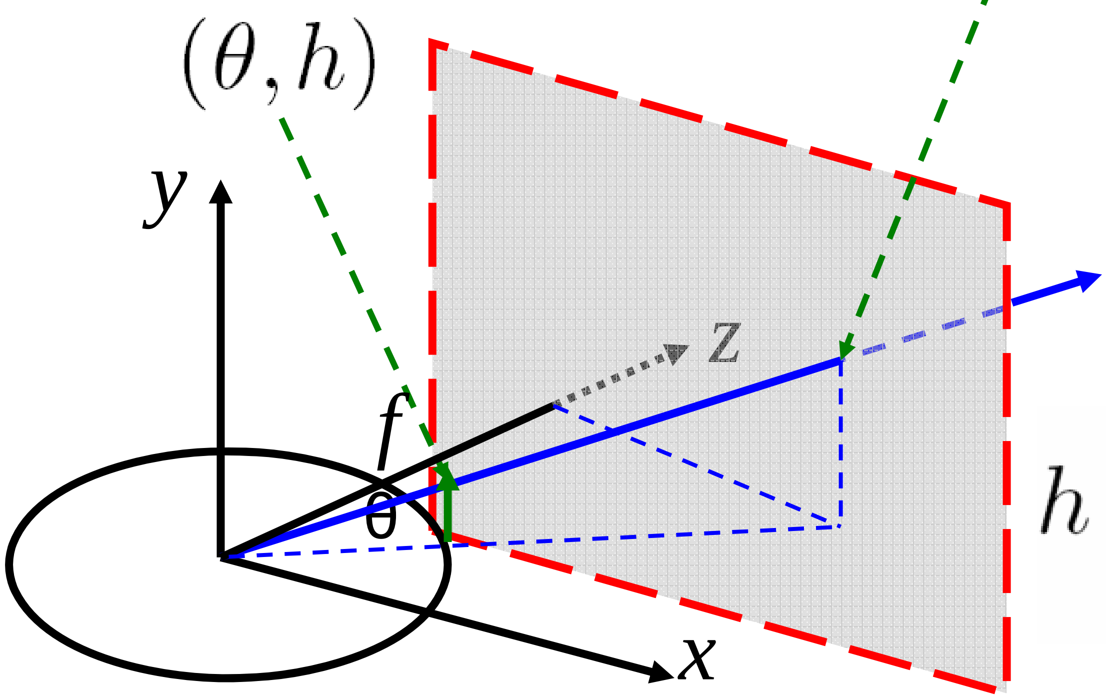
Cylindrical projection



unwrapped cylinder

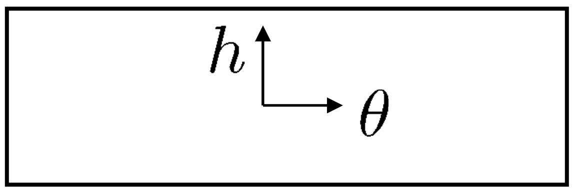


$$(\sin\theta, h, \cos\theta) \propto (x, y, f)$$

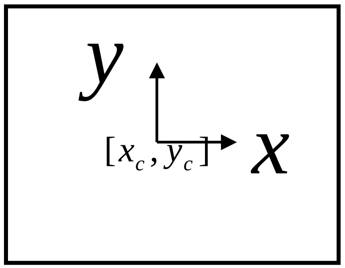


$$h = \frac{y}{\sqrt{x^2 + f^2}}$$

Cylindrical projection

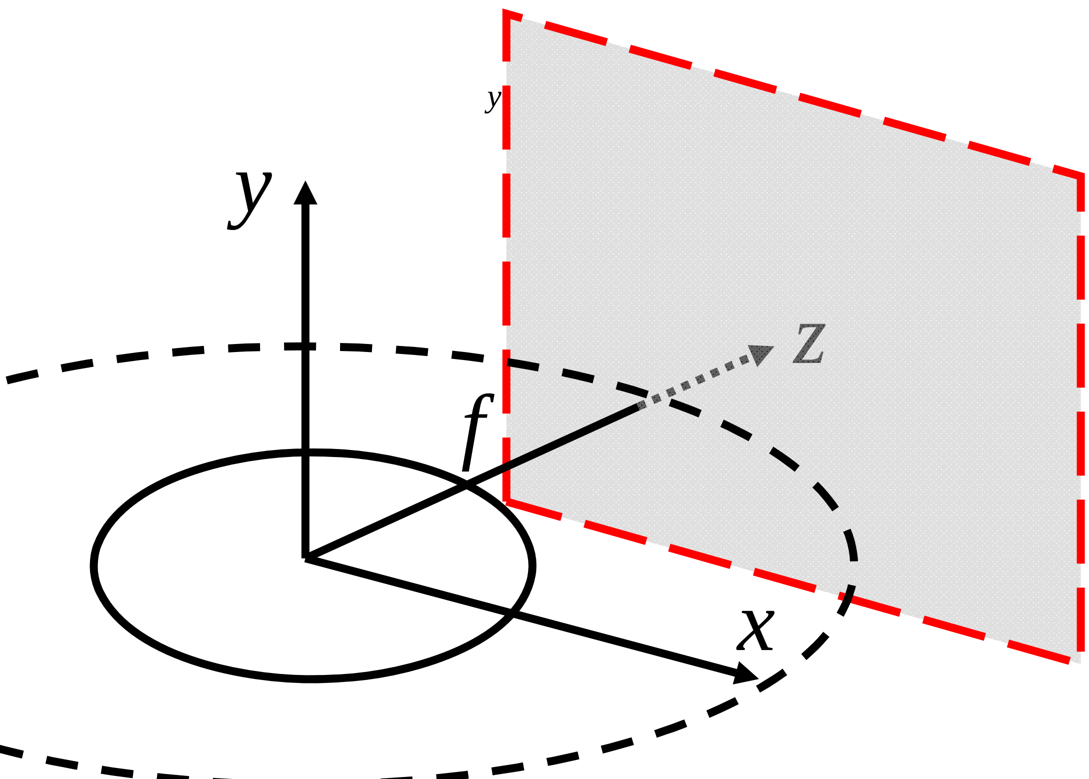


unwrapped cylinder

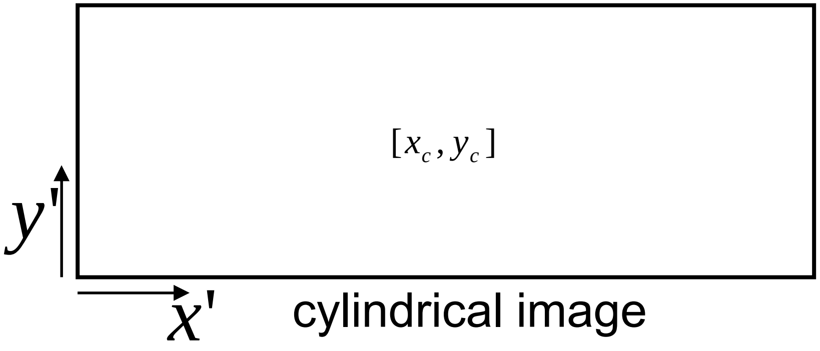


$$x' = s\theta + x_c \;= s\tan^{-1}\frac{x}{f} + x_c$$

$$y' = sh + y_c \;= s\frac{y}{\sqrt{x^2 + f^2}} + y_c$$

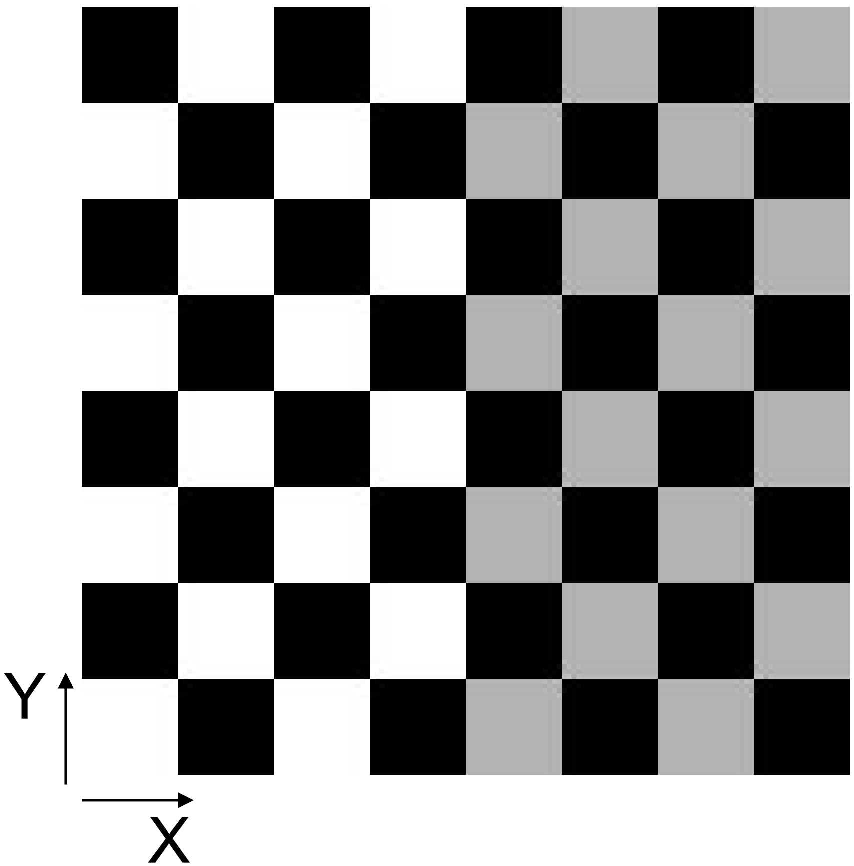


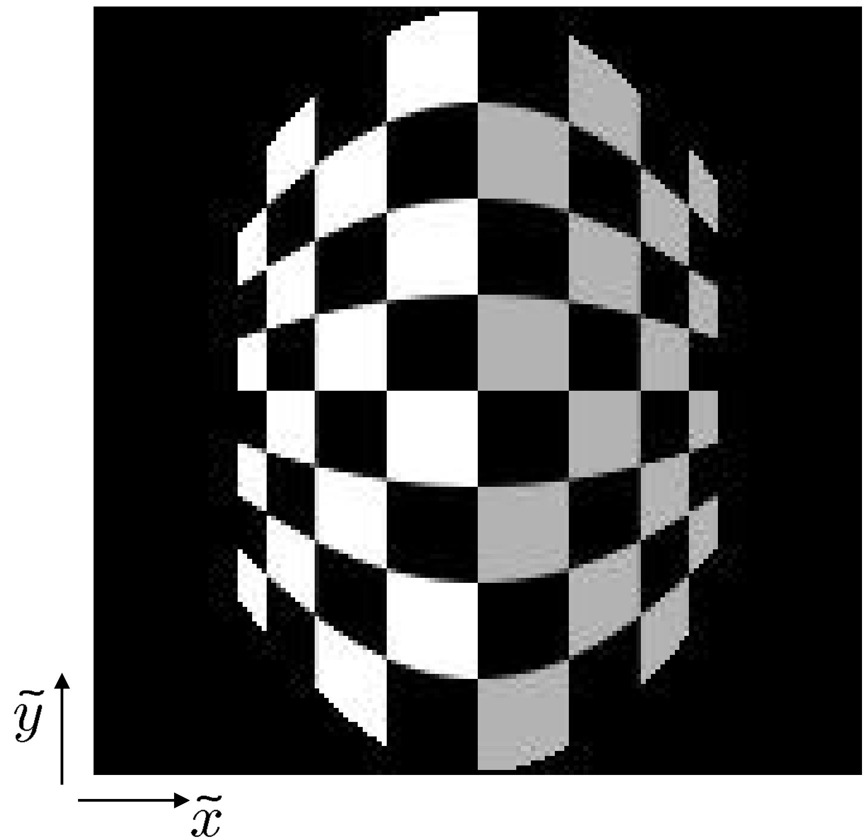
s defines size of the final image, often convenient to set s = f



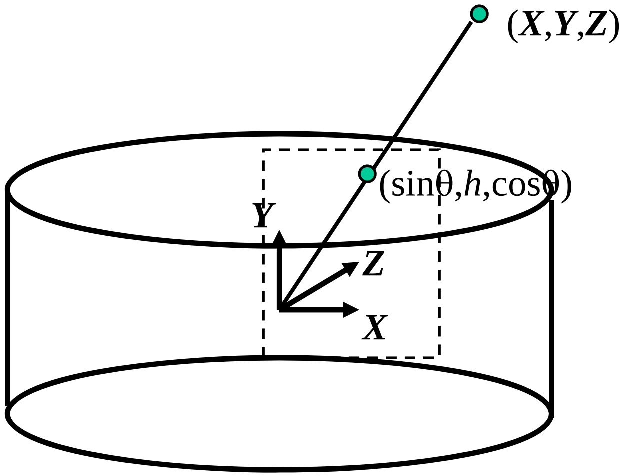
cylindrical image

Cylindrical Projection



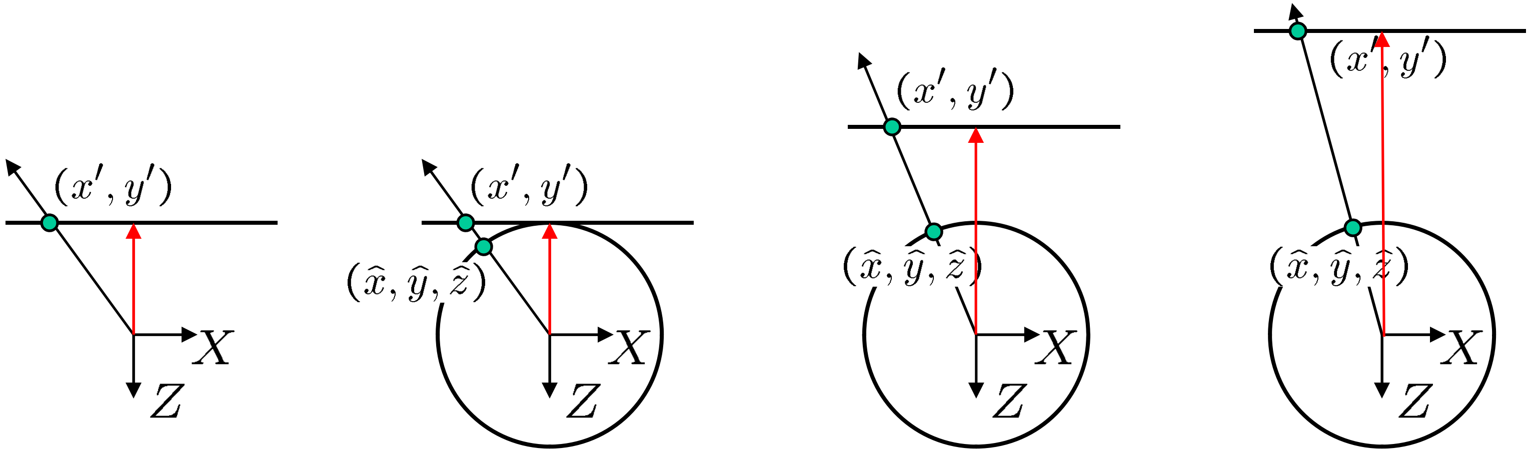


Inverse Cylindrical projection



$$
\begin{aligned}
\theta &= (x_{cyl} - x_c)/f \\
h &= (y_{cyl} - y_c)/f \\
\hat{x} &= \sin\theta \\
\hat{y} &= h \\
\hat{z} &= \cos\theta \\
x &= f\hat{x}/\hat{z} + x_c \\
y &= f\hat{y}/\hat{z} + y_c
\end{aligned}
$$

Need to know the focal length

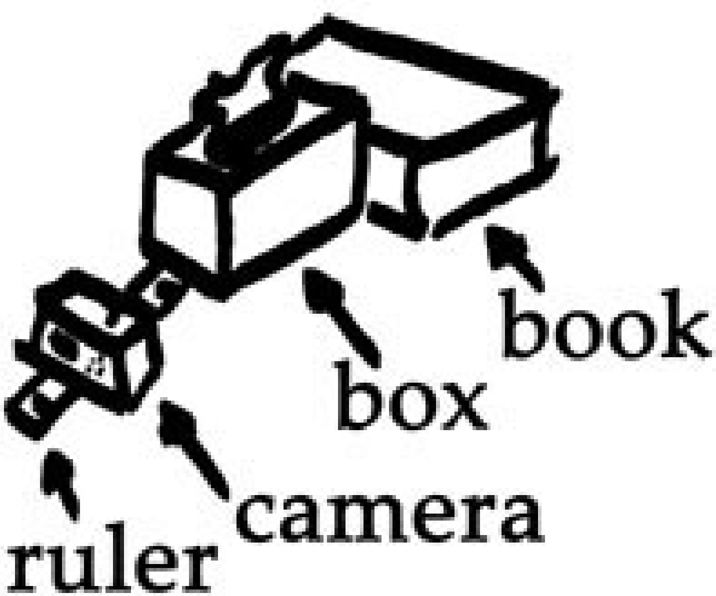


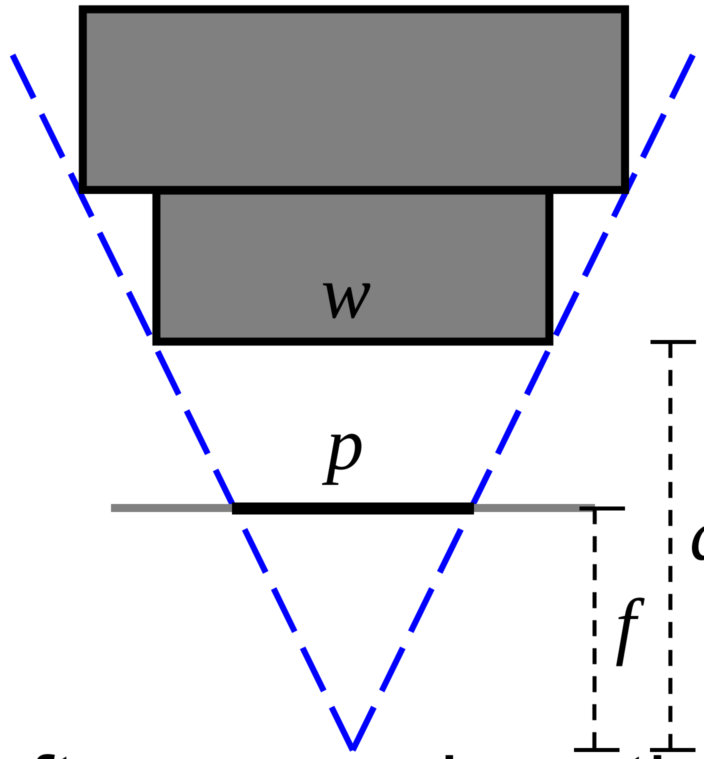
top-down view

Focal length – the dirty secret…

Image 384x300 **f = 180 (pixels)** **f = 280** **f = 380**

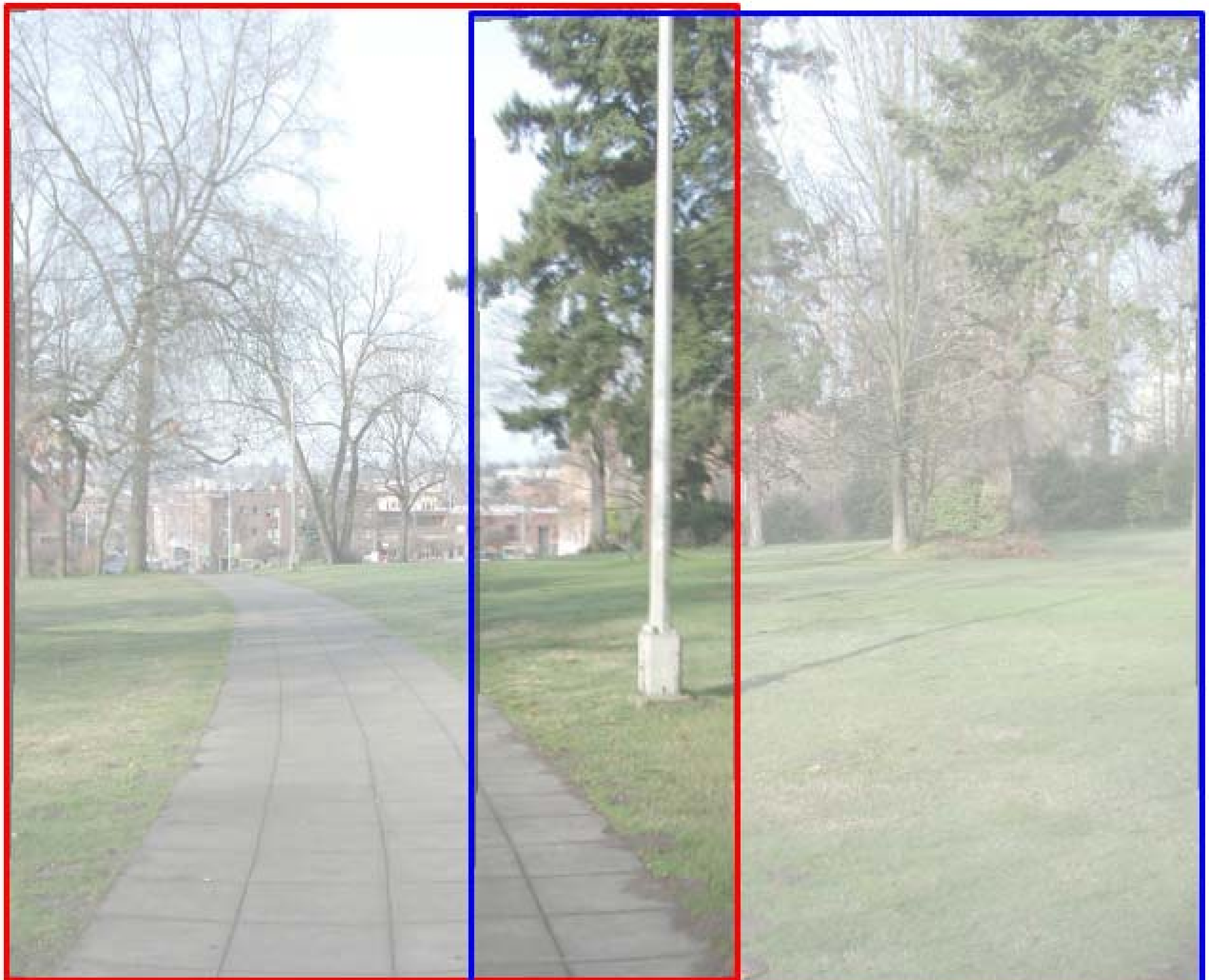
A simple method for estimating f



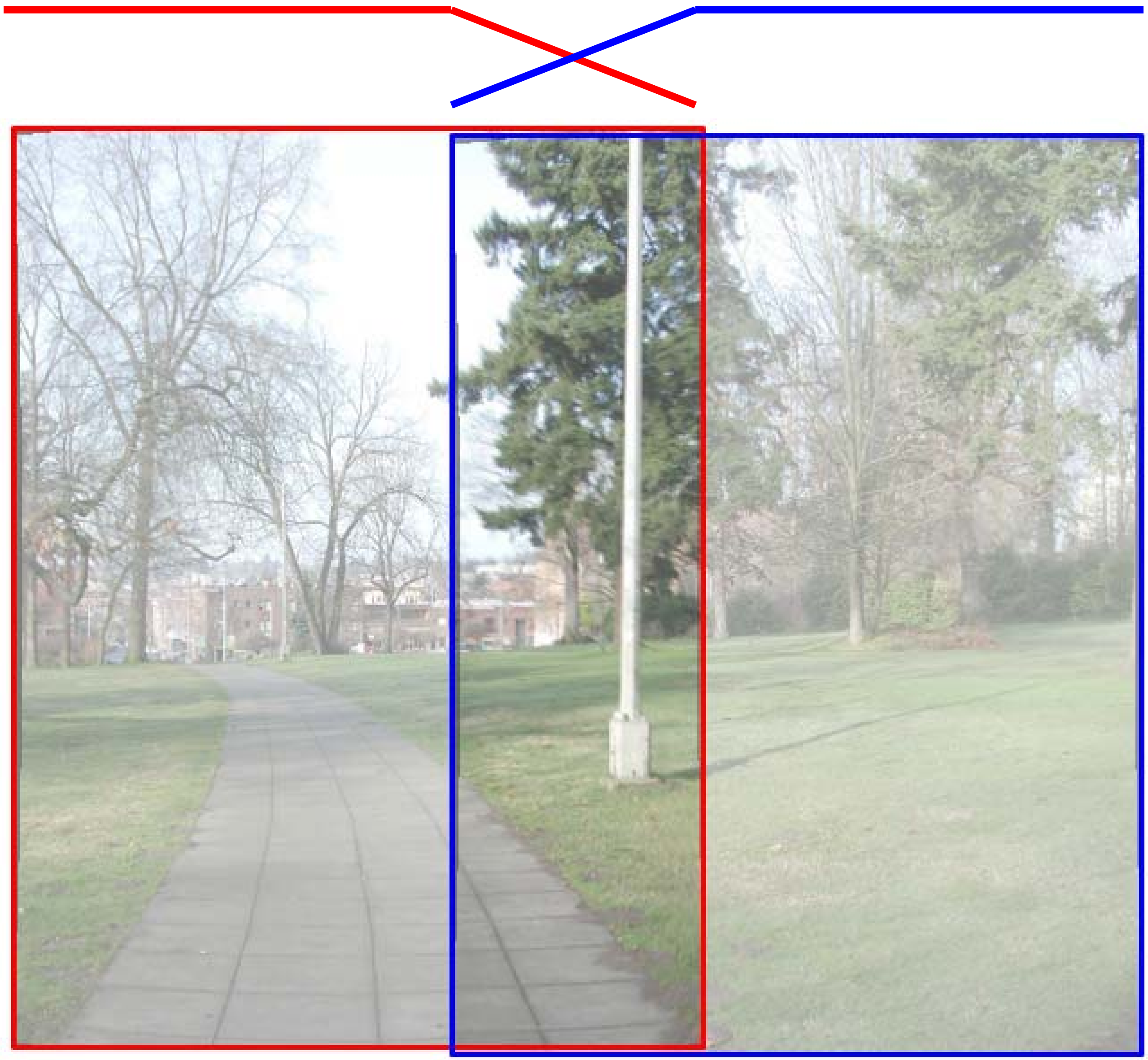


Or, you can use other software, such as the Caltech Camera Calibration Toolkit, to help.

Blending

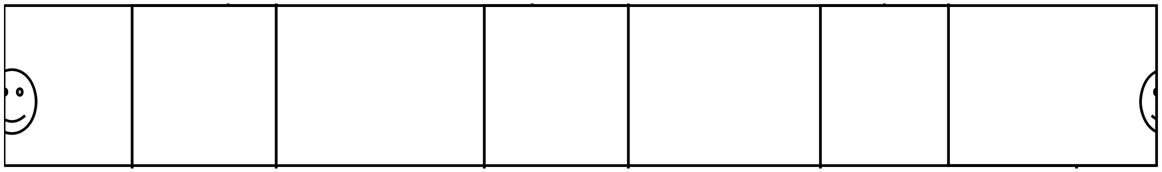


Blending



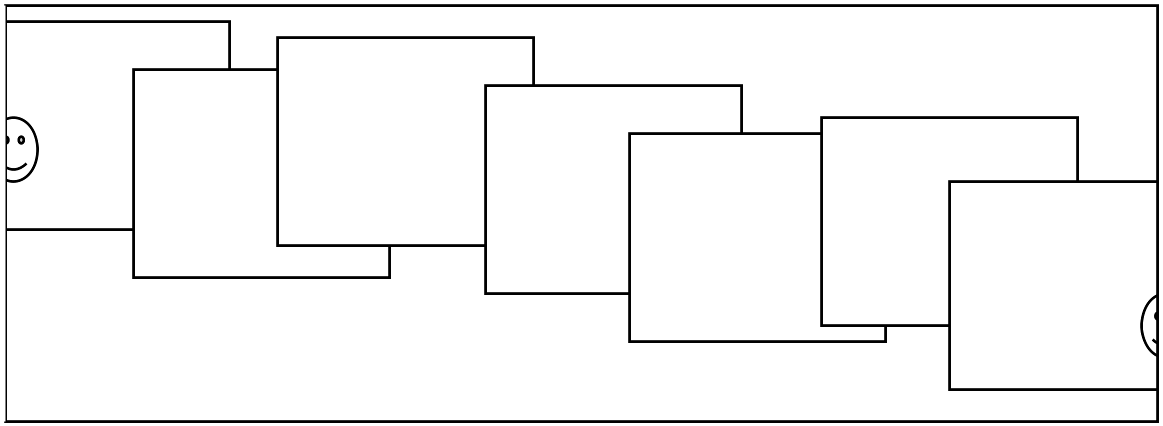
Blending

Assembling the panorama



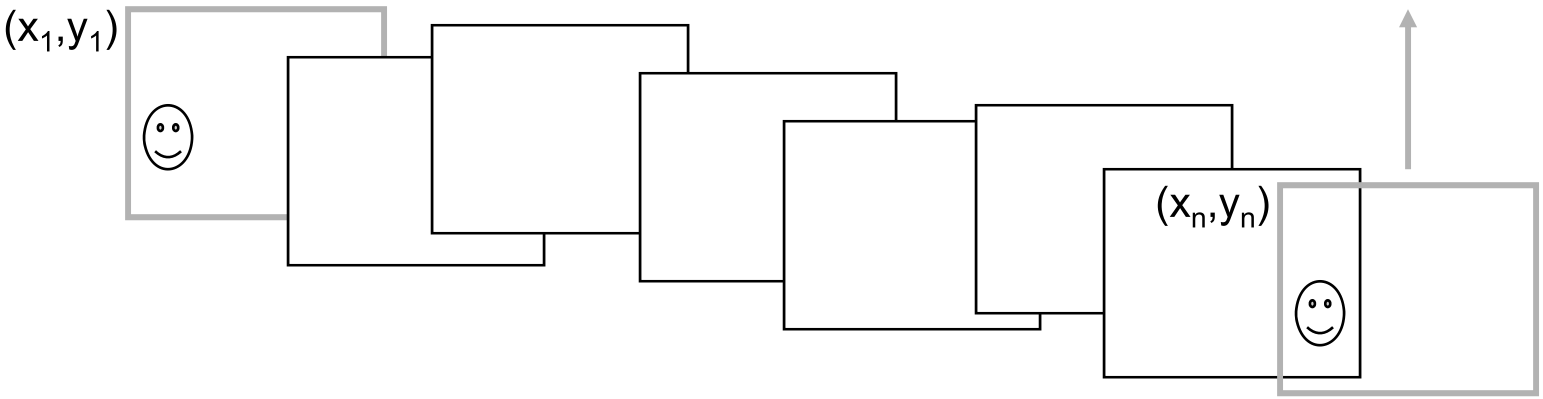
- Stitch pairs together, blend, then crop

Problem: Drift



- Error accumulation
  - small errors accumulate over time

Problem: Drift



- copy of first image

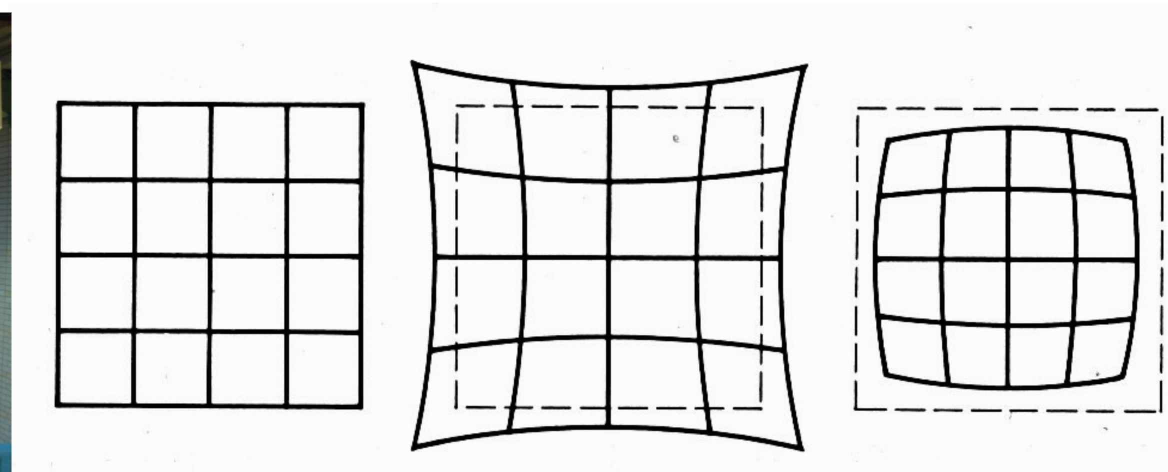
- Solution
  - add another copy of first image at the end
  - there are a bunch of ways to solve this problem
    - add displacement of $(y_1 – y_n)/(n -1)$ to each image after the first
    - compute a global warp: $y' = y + ax$
    - run a big optimization problem, incorporating this constraint
  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

End-to-end alignment and crop

Cylindrical panorama

1. Take pictures on a tripod (or handheld)
2. Warp to cylindrical coordinate
3. Compute pairwise alignments
4. Fix up the end-to-end alignment
5. Blending
6. Crop the result and import into a viewer

Distortion



No distortion Pin cushion Barrel

- Radial distortion of the image
  - Caused by imperfect lenses
  - Deviations are most noticeable for rays that pass through the edge of the lens

Removing distortion

Distortion-Free:

$$x = \frac{fX}{Z} + x_c$$

$$y = \frac{fY}{Z} + y_c$$

Distortion Model:

1. Project (X, Y, Z) to “normalized” image coordinates

$$x_n = \frac{X}{Z}$$

$$y_n = \frac{Y}{Z}$$

$$r^2 = x_n^2 + y_n^2$$

2. Apply radial distortion

$$x_d = x_n\left(1 + \kappa_1 r^2 + \kappa_2 r^4\right)$$

$$y_d = y_n\left(1 + \kappa_1 r^2 + \kappa_2 r^4\right)$$

3. Apply focal length translate image center

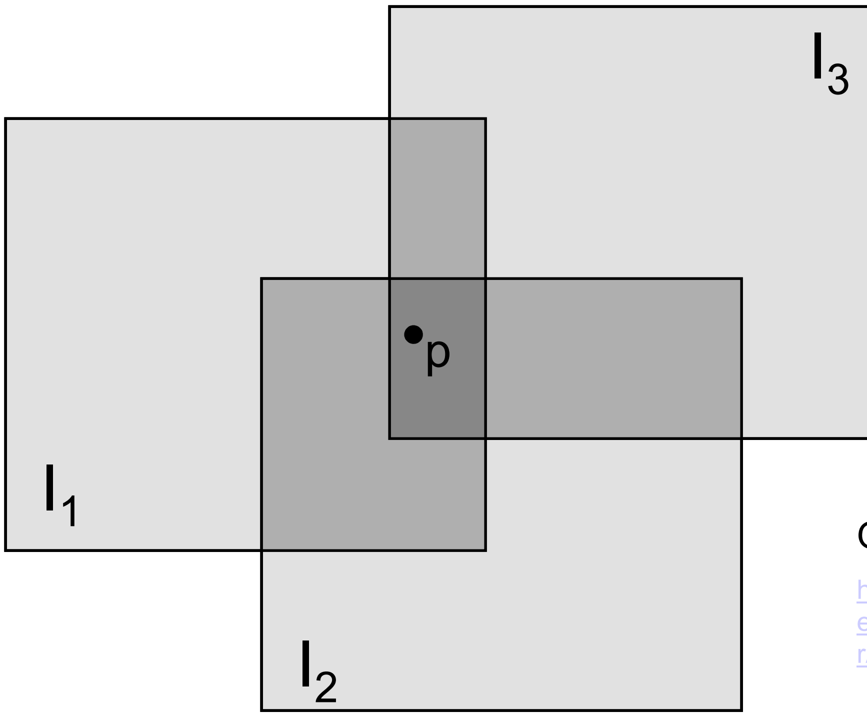
$$x' = f x_d + x_c$$

$$y' = f y_d + y_c$$

- How can we undo radial distortion if we know k1, k2, and f?
  - Inverse warping

Removing Radial Distortion

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

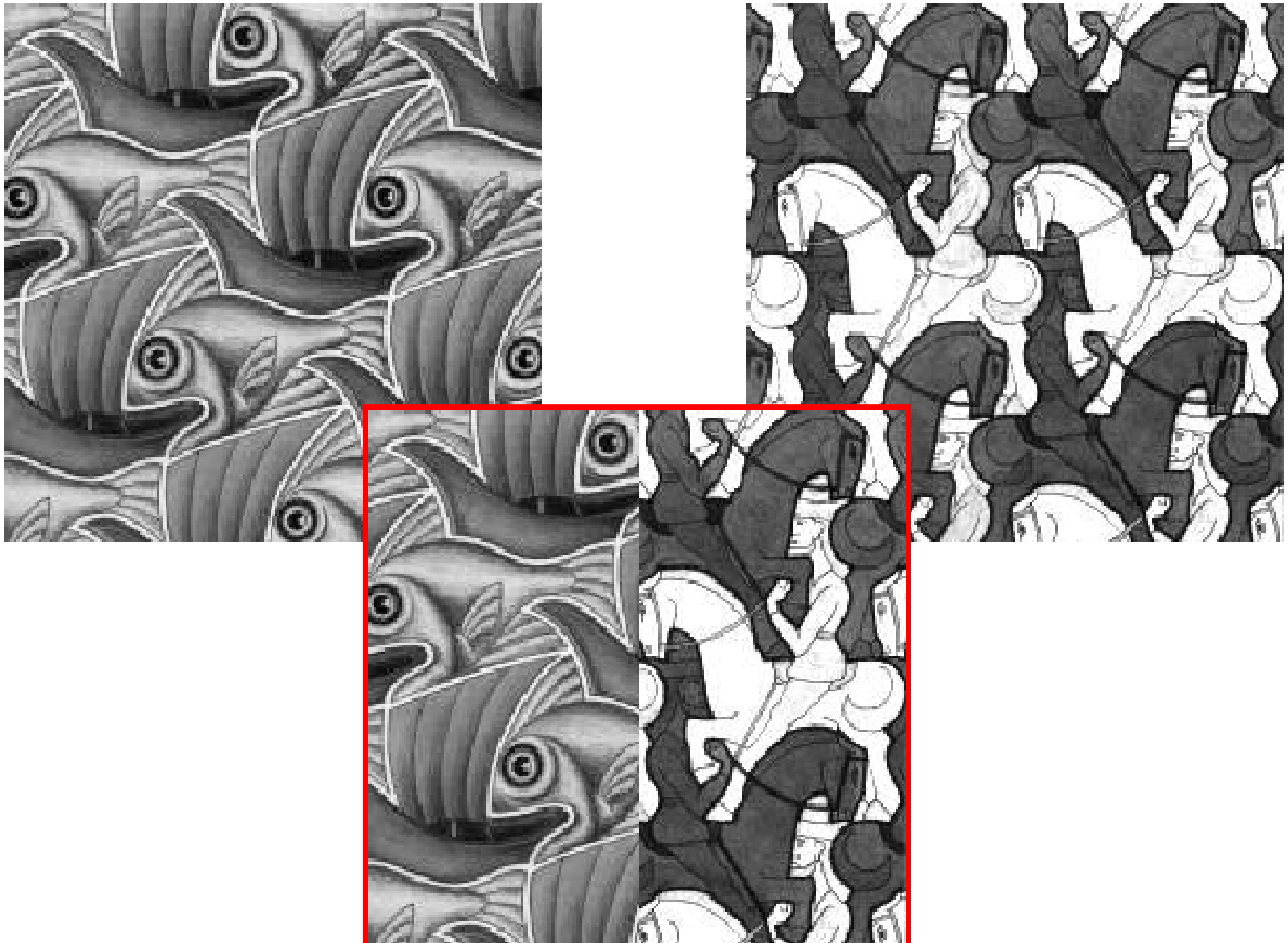
color at p = $$\frac{(\alpha_1 R_1,\ \alpha_1 G_1,\ \alpha_1 B_1) + (\alpha_2 R_2,\ \alpha_2 G_2,\ \alpha_2 B_2) + (\alpha_3 R_3,\ \alpha_3 G_3,\ \alpha_3 B_3)}{\alpha_1 + \alpha_2 + \alpha_3}$$

Implement this in two steps:

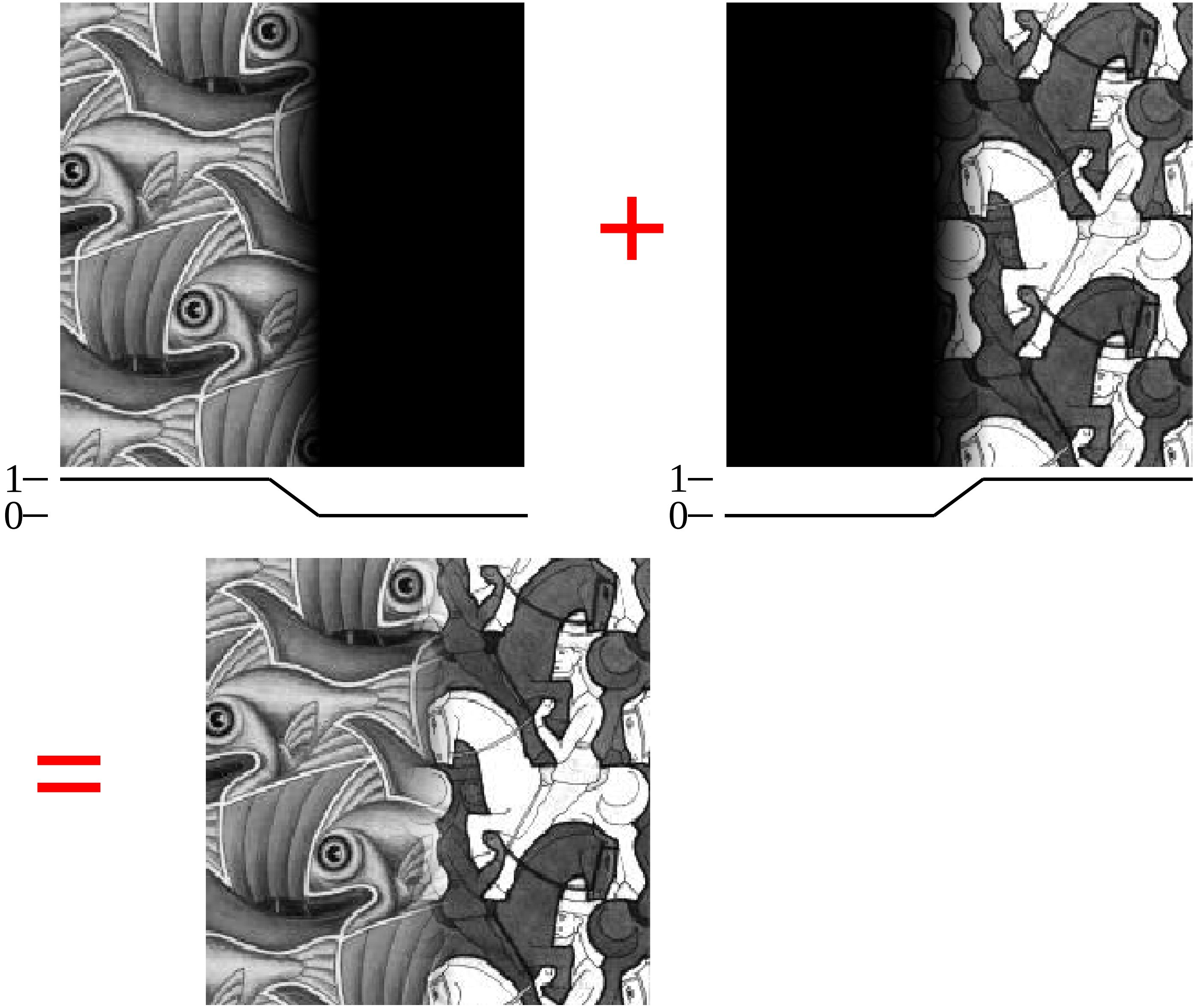
1. accumulate: add up the (α premultiplied) RGBα values at each pixel
2. normalize: divide each pixel's accumulated RGB by its α value

   Q: what if $\alpha = 0$?

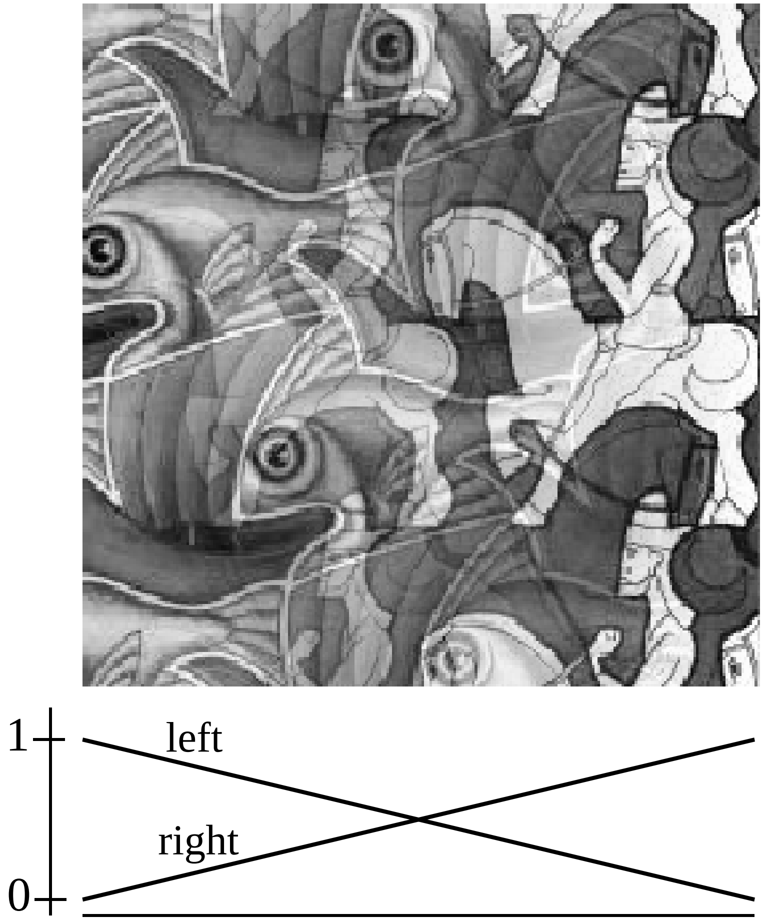
Image Blending

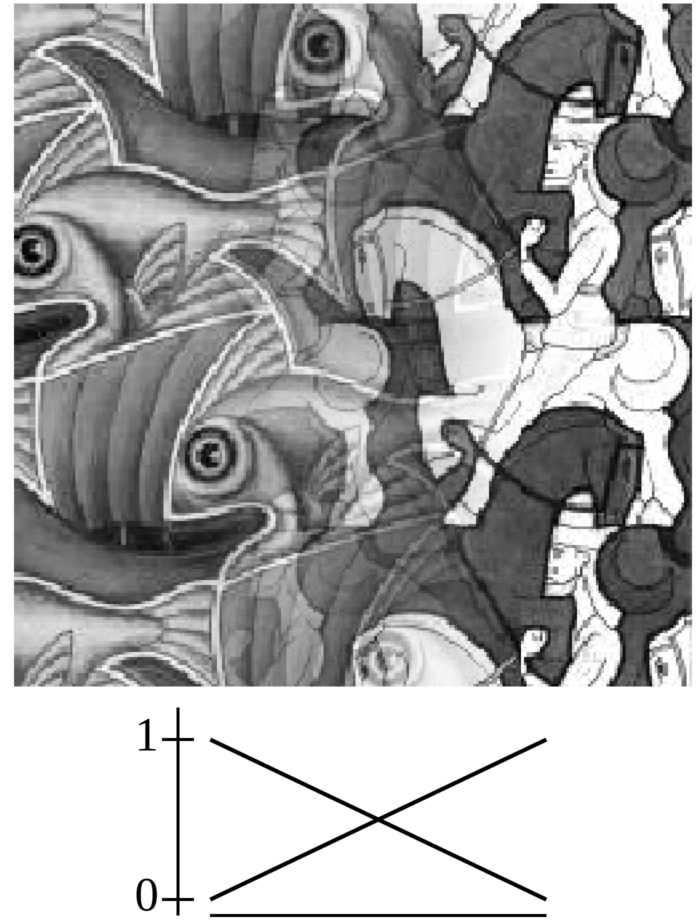


Feathering

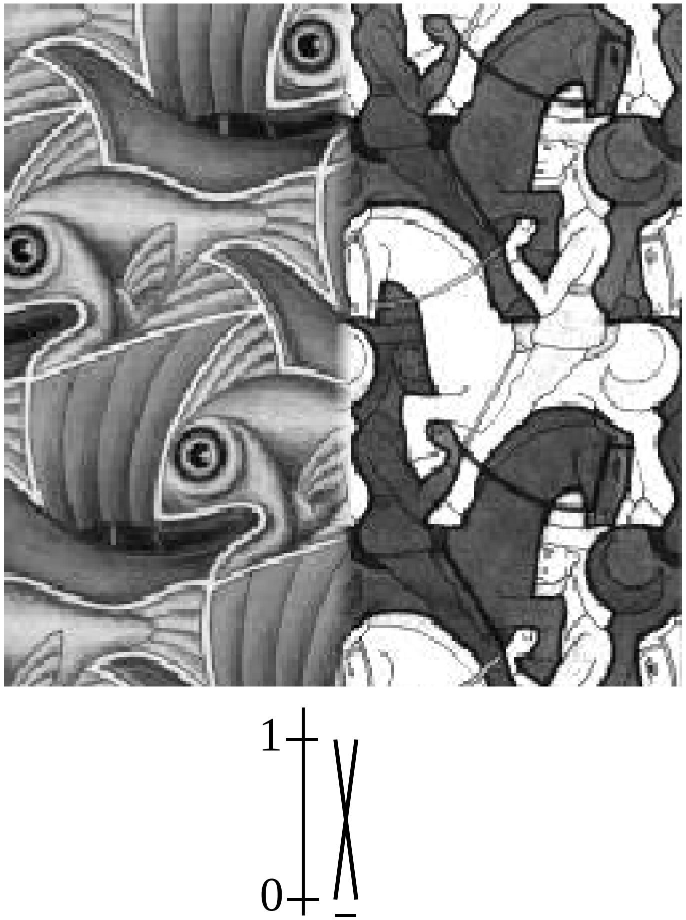


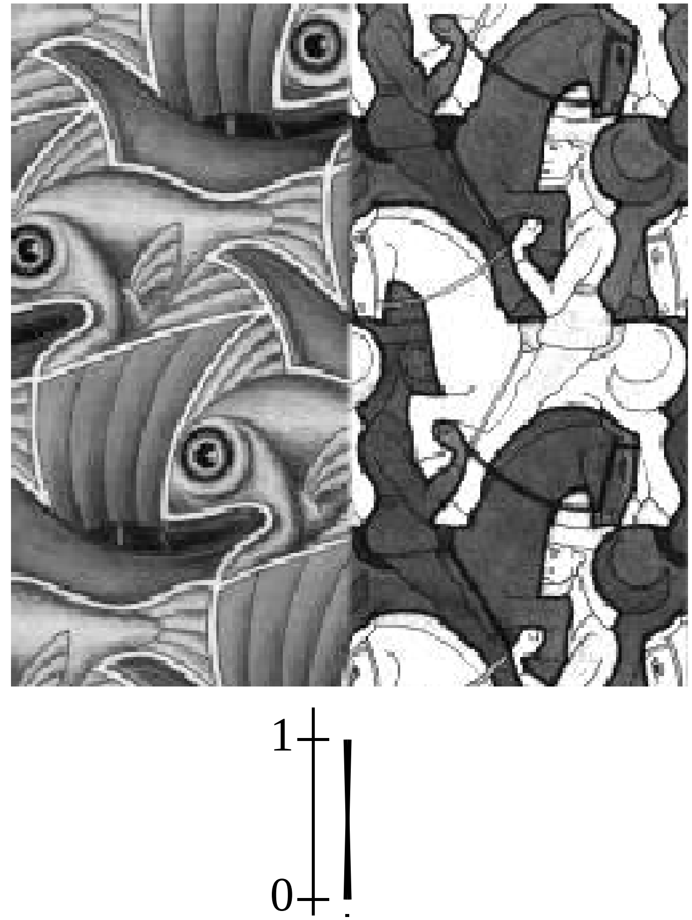
Effect of window size



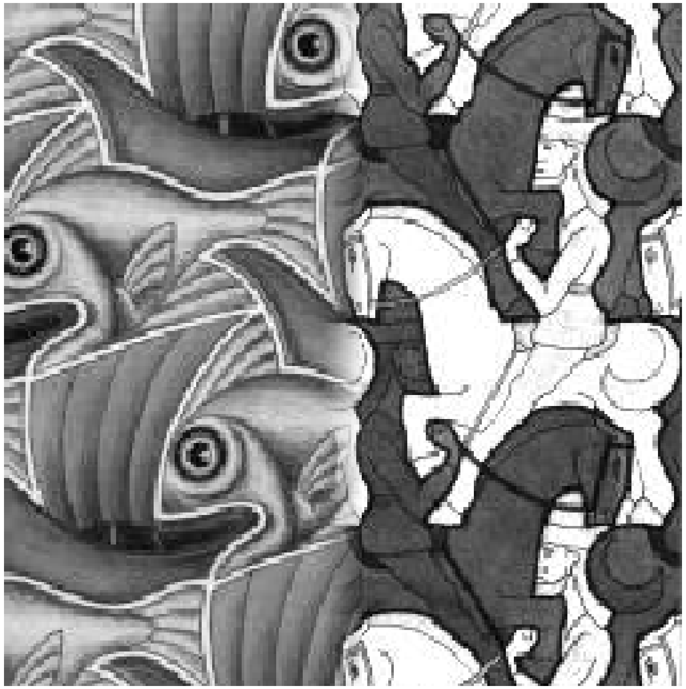


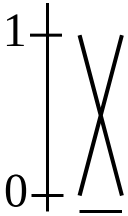
Effect of window size





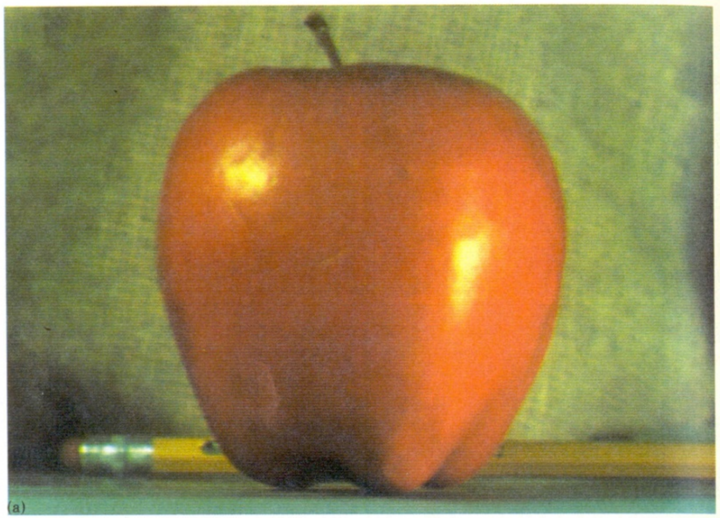
Good window size

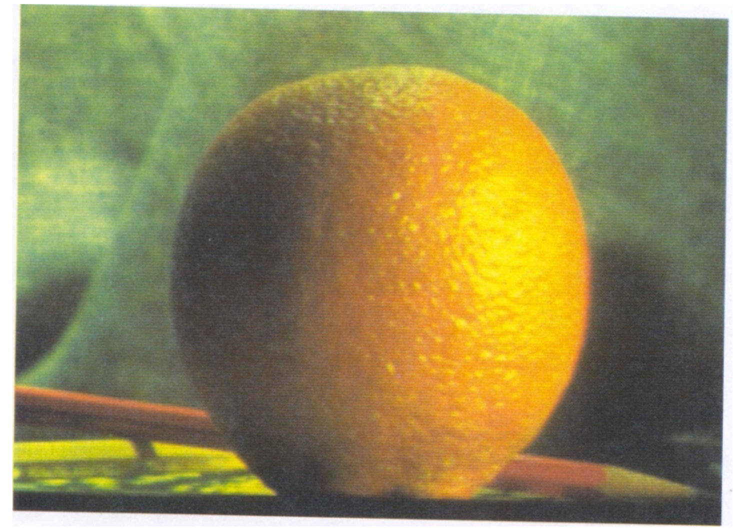


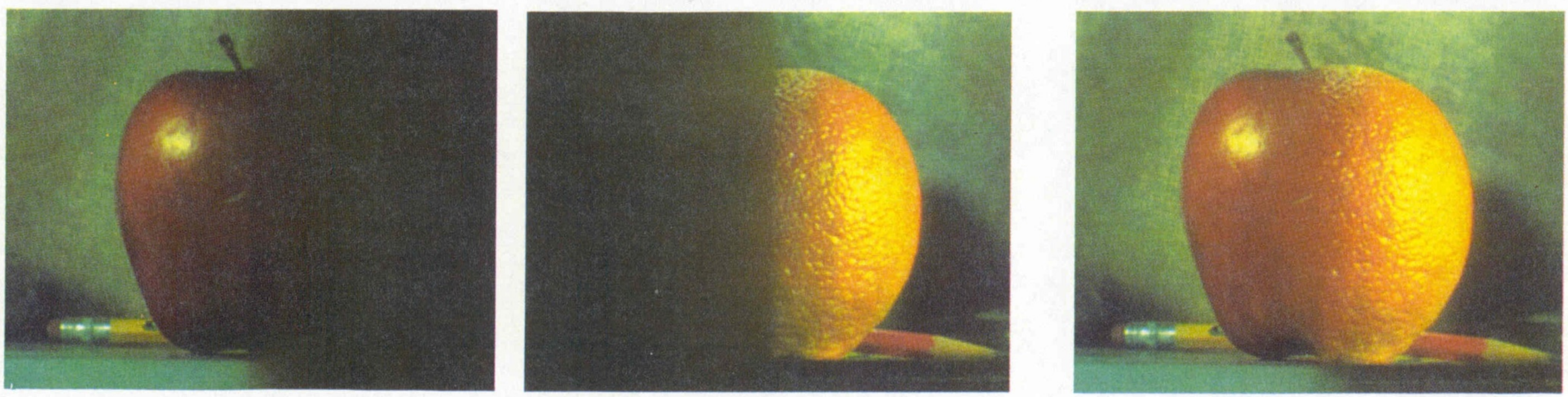


- “Optimal” window: smooth but not ghosted
  - Doesn’t always work...

Pyramid blending







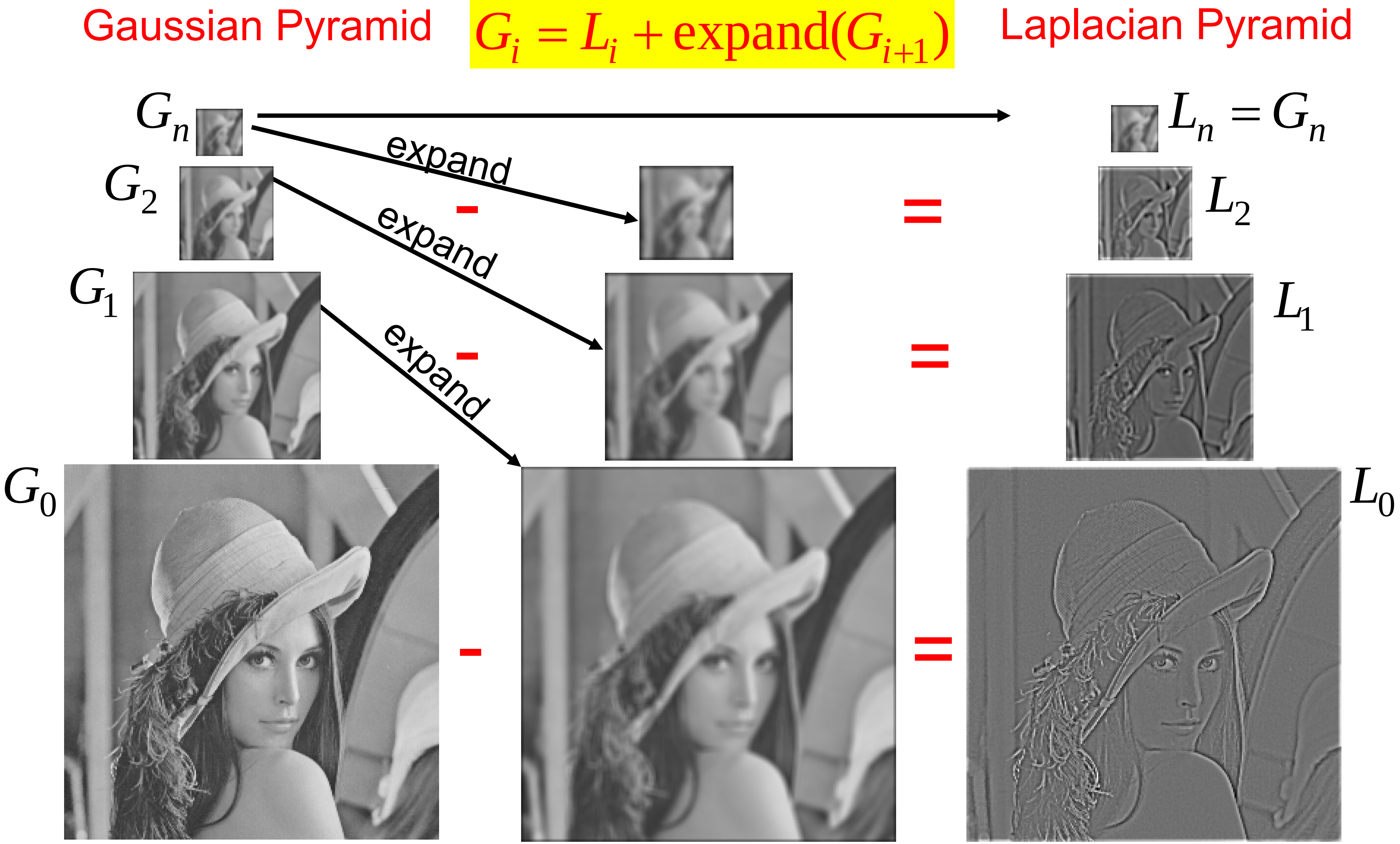
(d) (h) (l)

- 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.

The Laplacian Pyramid

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

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



Multi-band Blending

Multi-band Blending

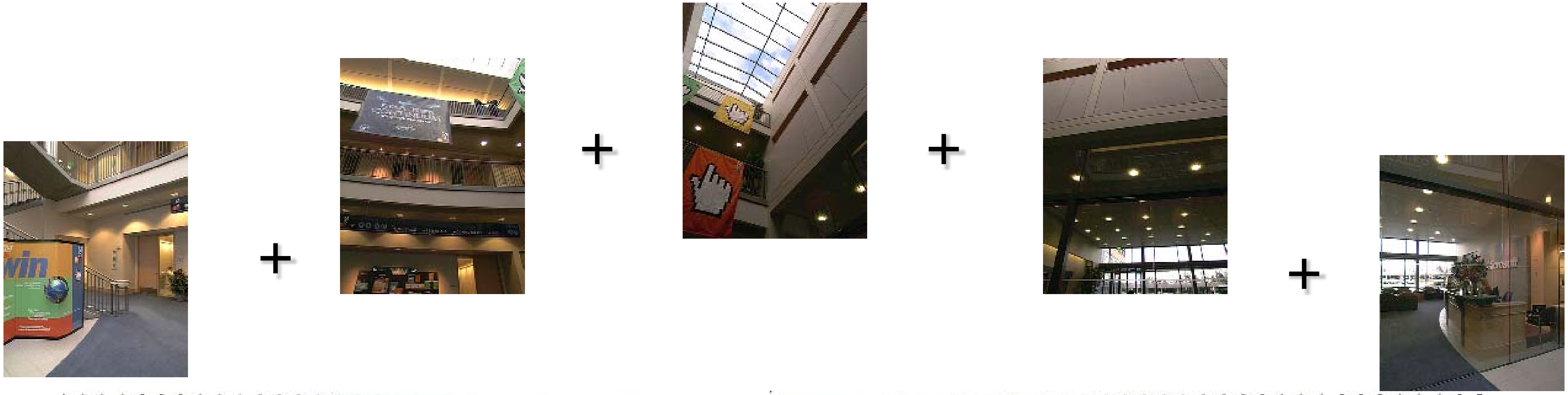
- Burt & Adelson 1983

Multi-band Blending

Poisson Image Editing

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

Some panorama examples



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

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

What’s inside your refrig?

http://www.cs.washington.edu/education/courses/cse590ss/01wi/

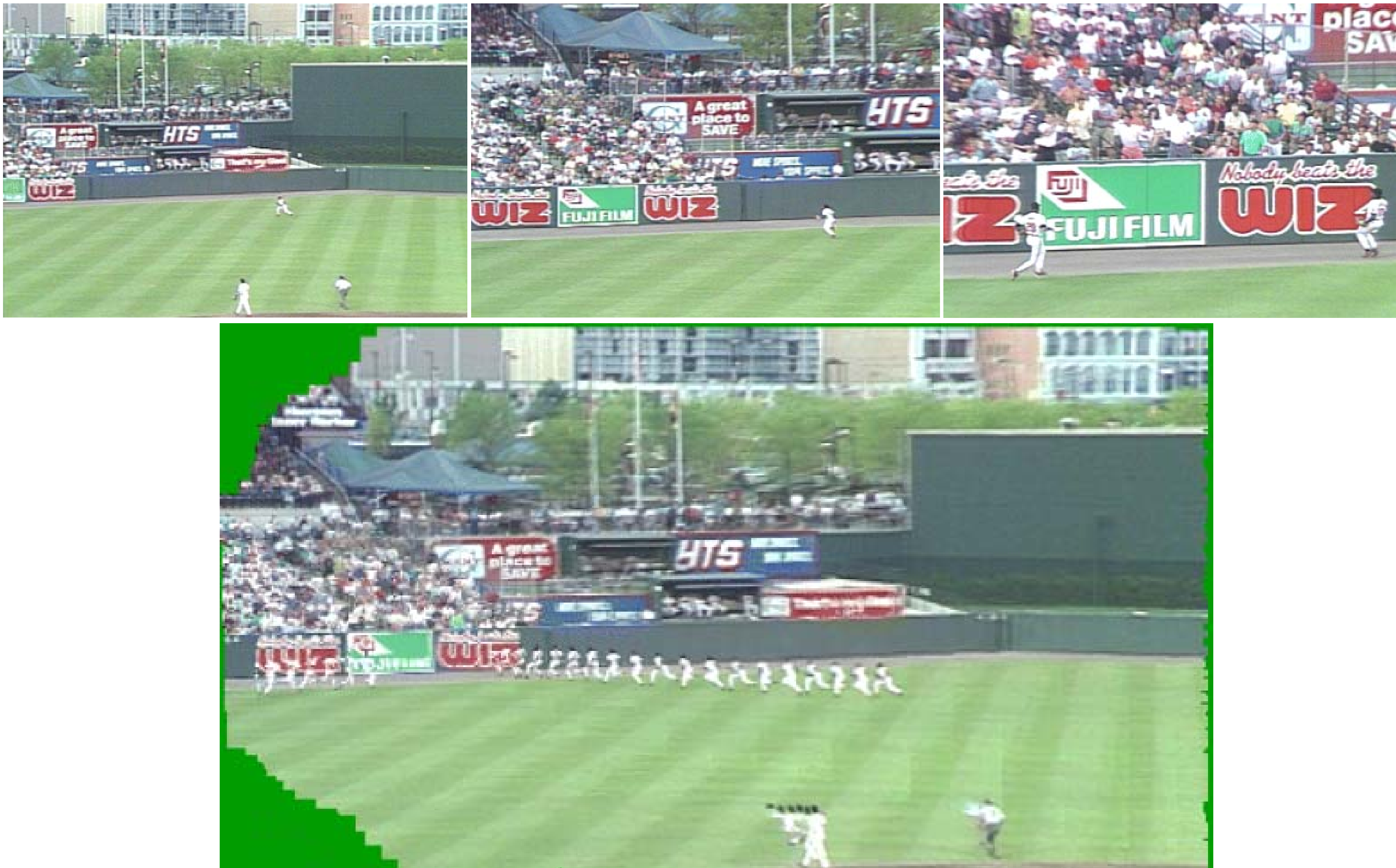
Some panorama examples

Mars: http://www.panoramas.dk/fullscreen3/f2_mars97.html

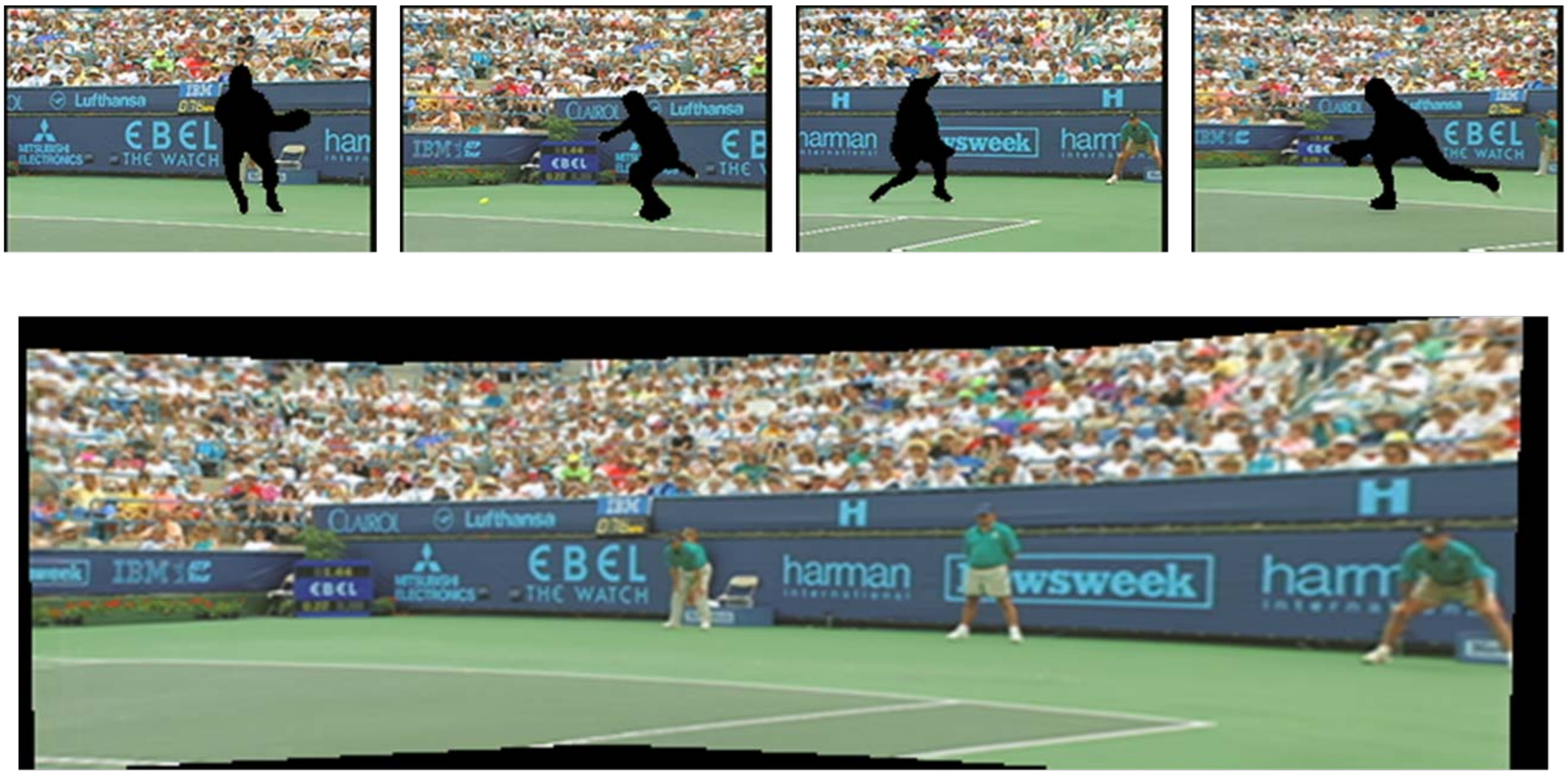
2003 New Years Eve: http://www.panoramas.dk/fullscreen3/f1.html

Video Summarization: http://www.vision.huji.ac.il/video-synopsis/

Video Summarization



Video compression



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.

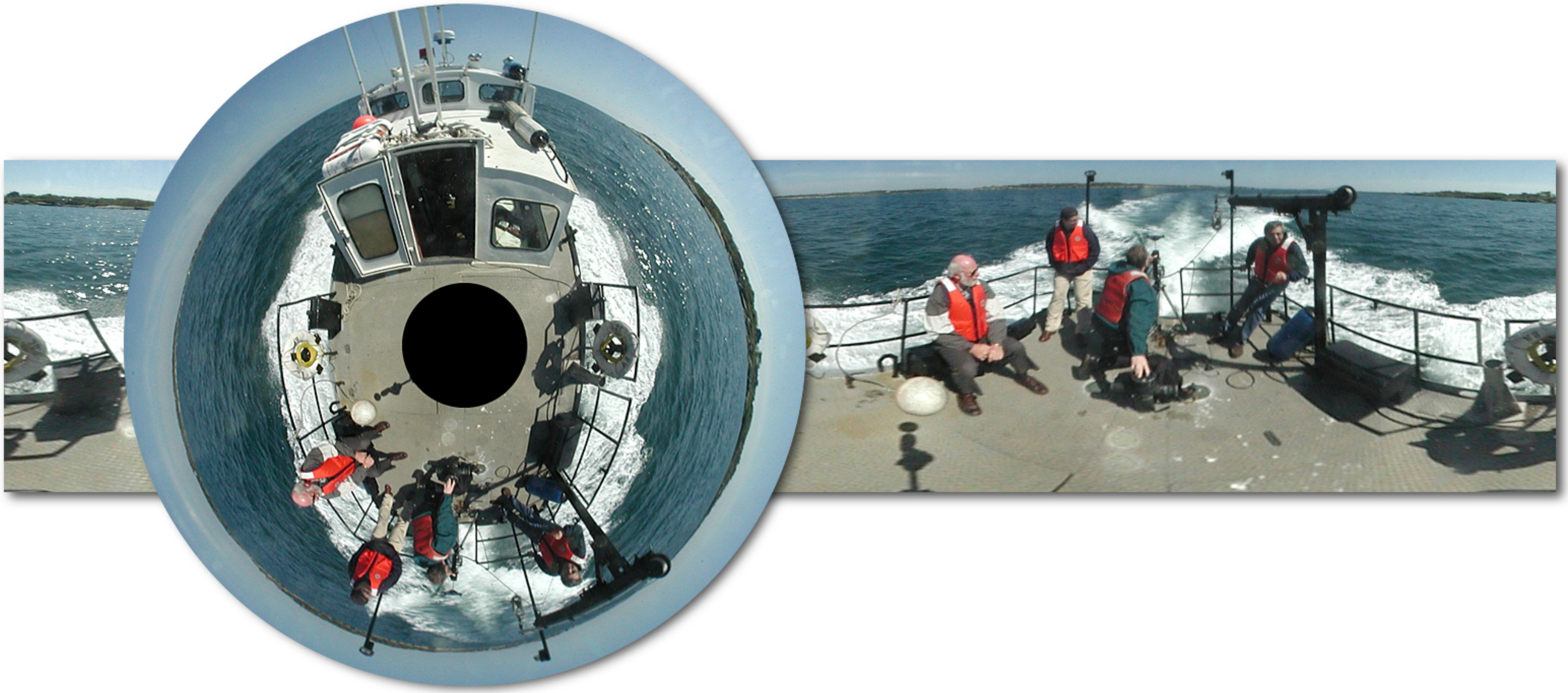
For dynamic Scenes

Point Grey Ladybug2

http://www.ptgrey.com/products/ladybug2/samples.asp

For dynamic scenes

http://www1.cs.columbia.edu/CAVE/projects/cat_cam_360/cat_cam_360.php




Video Conferencing

More and Blending

Cylindrical panorama

1. Take pictures on a tripod (or handheld)
2. Warp to cylindrical coordinate
3. Compute pairwise alignments
4. Fix up the end-to-end alignment
5. Blending
6. Crop the result and import into a viewer

Cylindrical panorama

1. Take pictures on a tripod (or handheld)
2. Warp to cylindrical coordinate
3. Compute pairwise alignments
4. Fix up the end-to-end alignment
5. Blending
6. Crop the result and import into a viewer