Today

Homographies, Mosaics, Panorama (some slides from Snavely)

- Today's Readings
  - Szeliski and Shum paper
    http://www.acm.org/pubs/citations/proceedings/graph/258734/p251-szeliski/
  - Hartley Zisserman (Ch 3—4)

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

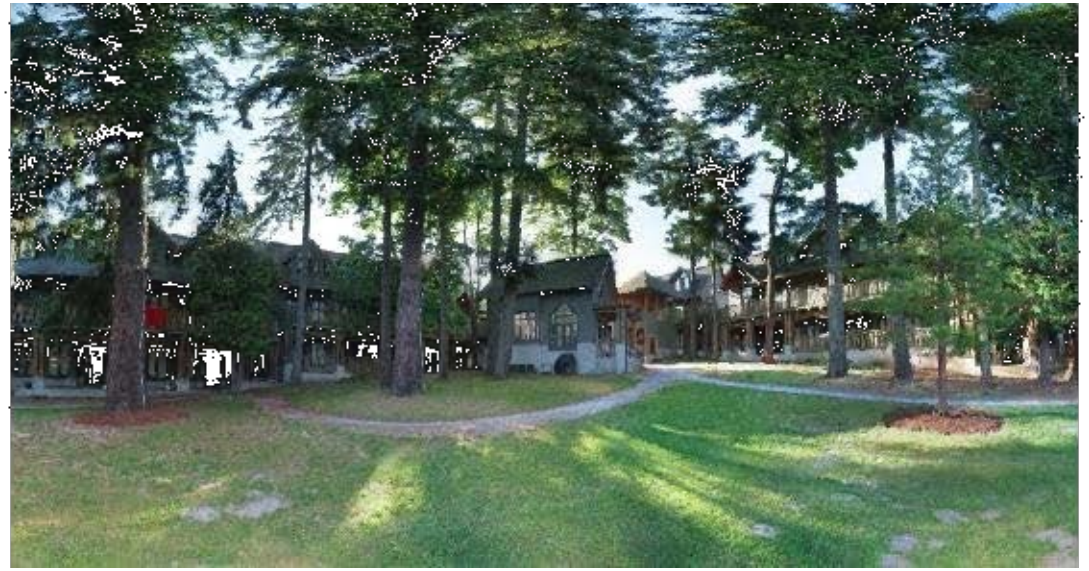
Why Mosaic?

- 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

Madison Panoramas

Hans Werner, 2011

Madison Panoramas

Jean Forde, 2011

Madison Panoramas

George Wanant, 2010

How to do it?

- Basic Procedure
  - Take a sequence of images from the same position
    - Rotate the camera about its optical center

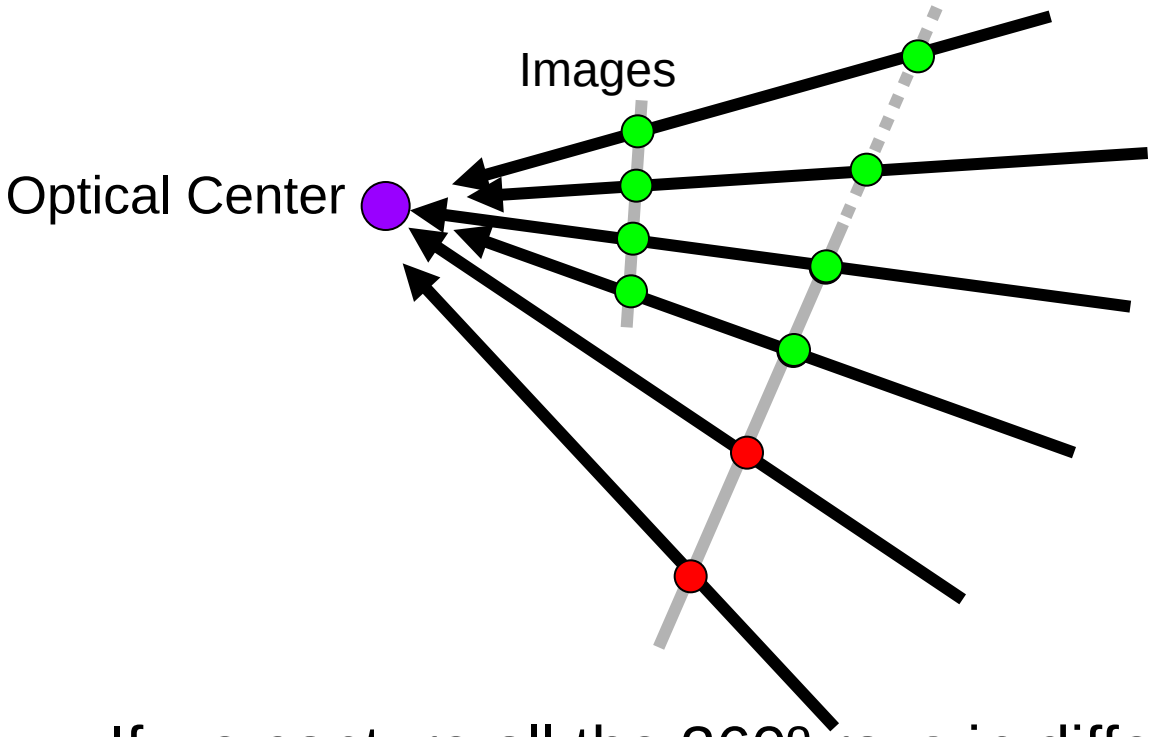
How to do it?

- 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 (somehow)
  - Transform the second image to overlap with first

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 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

What is the geometric relationship between these two images?

?

What is the transformation?

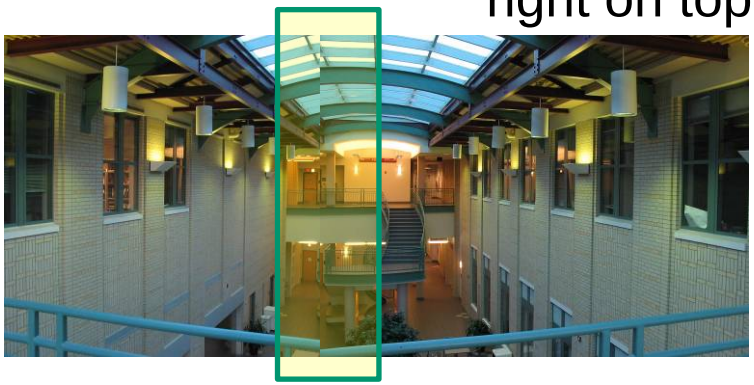
left on top

right on top

What is the transformation?

left on top

right on top



What is the transformation?

left on top

right on top

Translations are not enough to align the images

Where to go from affine transformations?

$$\begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix}$$

affine transformation

Where to go from affine transformations?

$$\begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix}$$

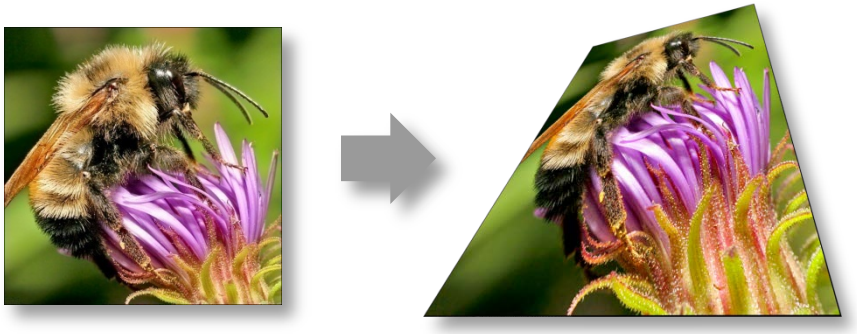
This is what happens when we change this row…

affine transformation

Projective Transformations aka Homographies aka Planar Perspective Maps

$$\mathbf{H} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix}$$

Called a *homography*
(or *planar perspective map*)



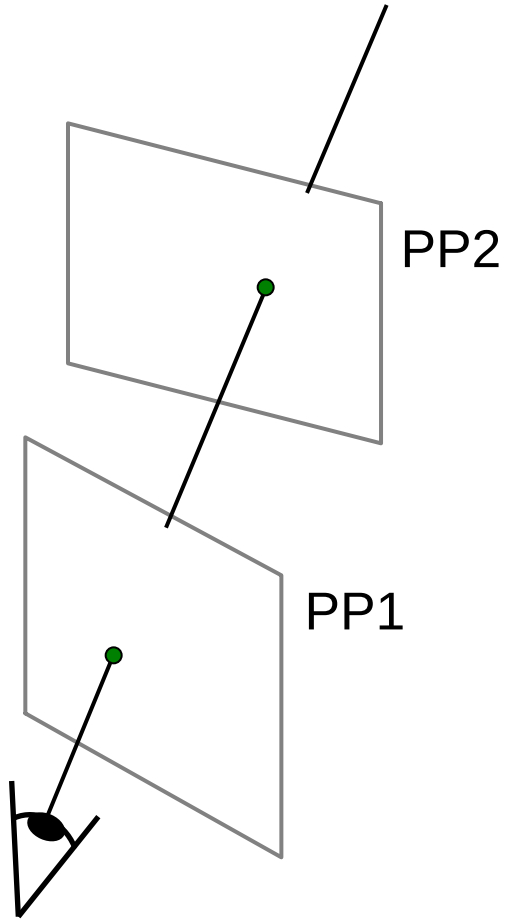
projection of 3D plane

Homography

- Mapping between any two PPs with the same center of projection
  - rectangle should map to arbitrary quadrilateral

$$\begin{bmatrix} wx' \\ wy' \\ w \end{bmatrix} = \begin{bmatrix} * & * & * \\ * & * & * \\ * & * & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$\mathbf{p'}$ $\quad$ $\mathbf{H}$ $\quad$ $\mathbf{p}$



To apply a homography **H**

- Compute **p'** = **Hp** (regular matrix multiply)
- Convert **p'** from homogeneous to image coordinates

Image warping with homographies

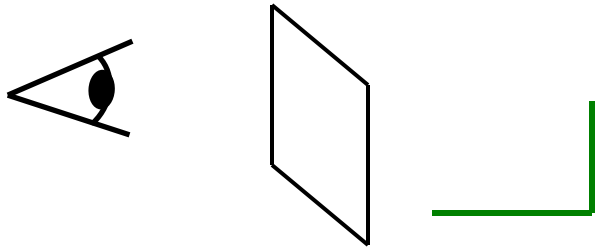
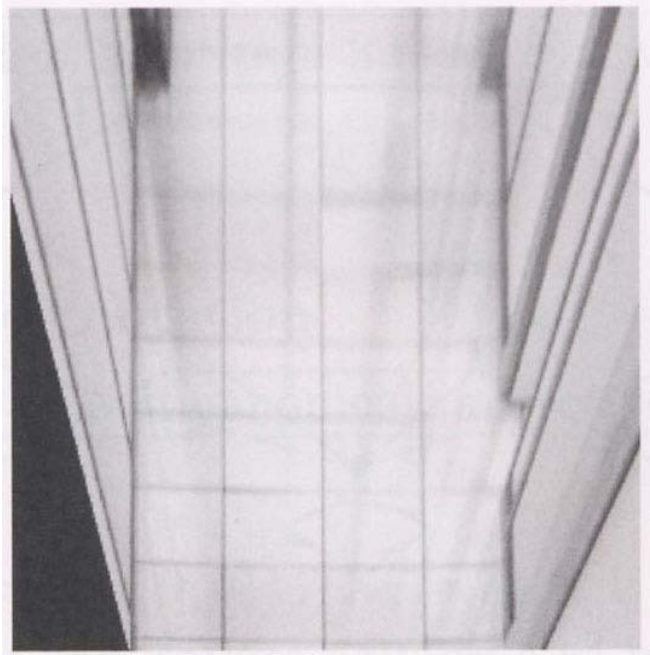


image plane in front

Image warping with homographies

$\mathbf{H}_1$



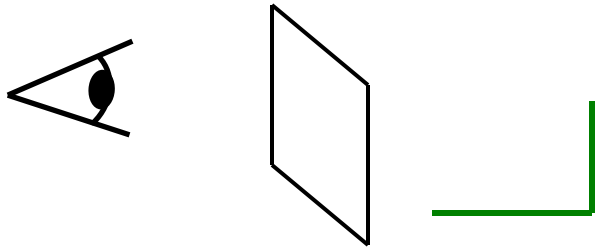


image plane in front

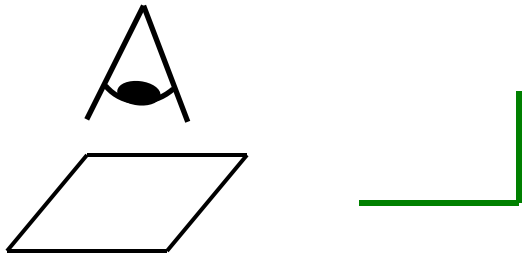
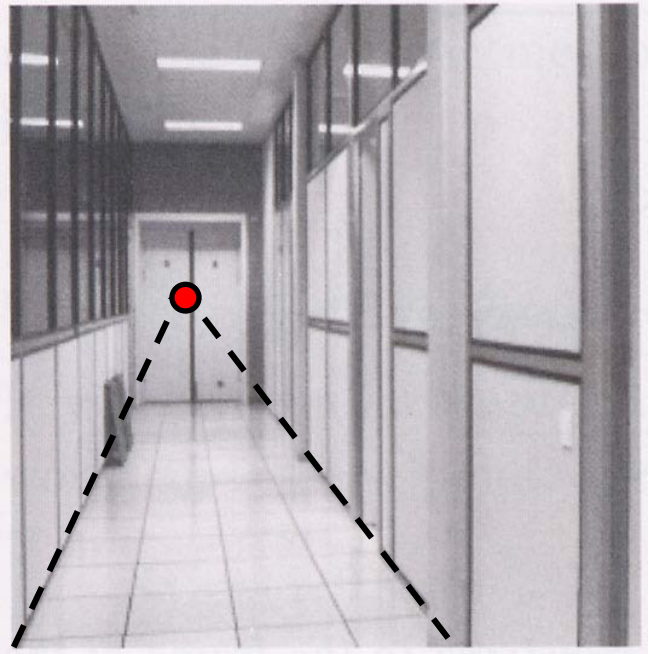
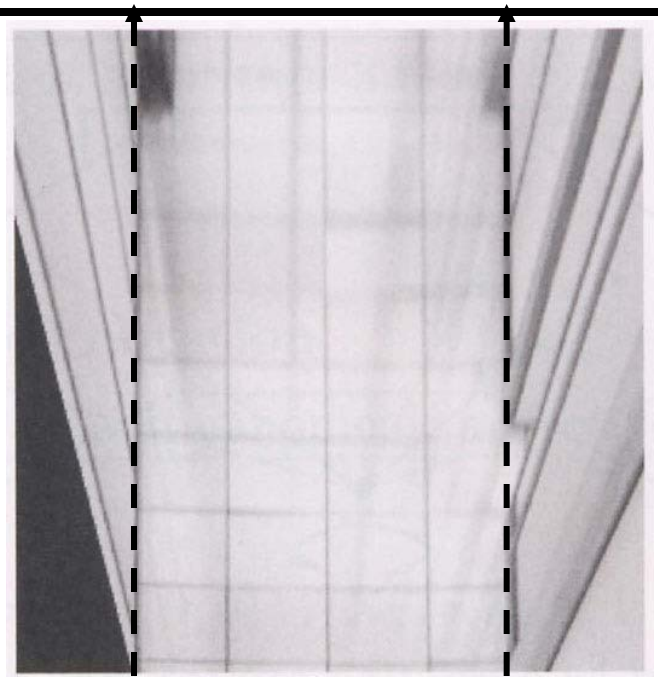


image plane below

Image warping with homographies



$\mathbf{H}_1$



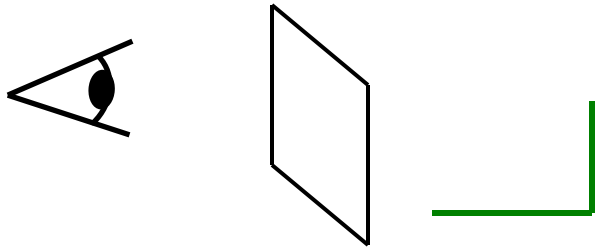


image plane in front

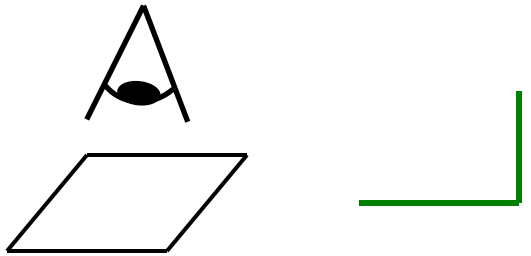
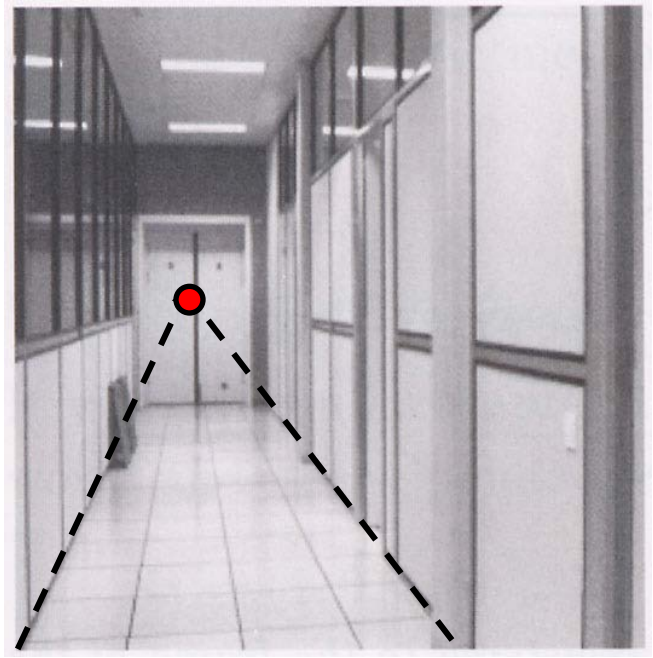
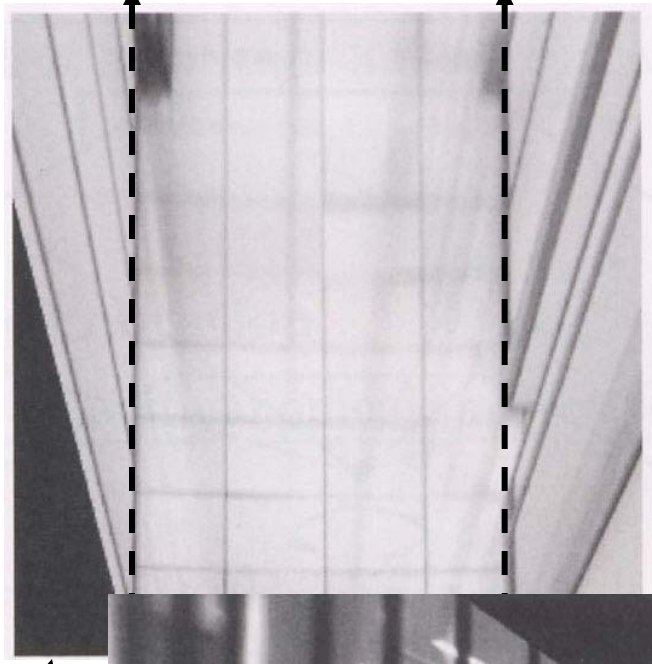


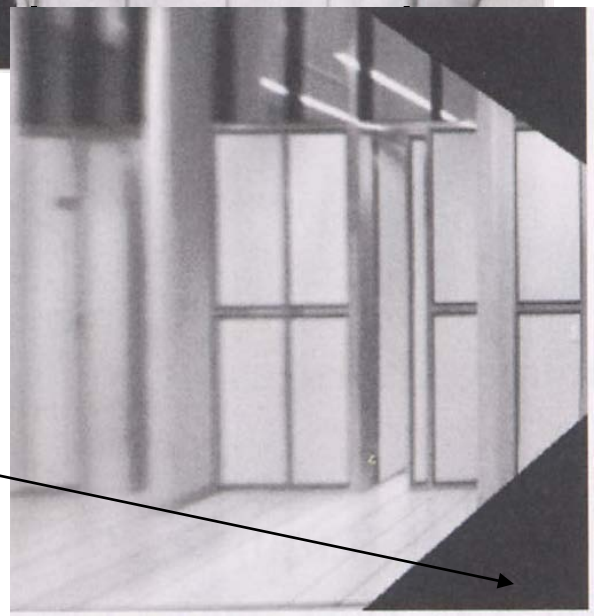
image plane below

Image warping with homographies



$\mathbf{H}_1$





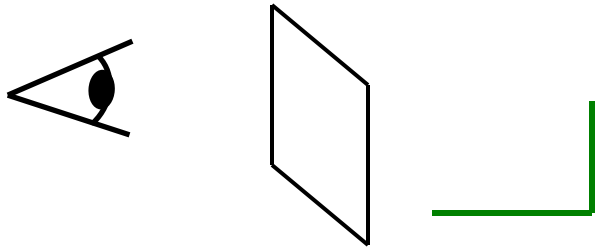


image plane in front

black area
where no pixel
maps to

Homographies

Observation:
Rather than thinking of this as a 3D reprojection,
think of it as a 2D **image warp** from one image to another

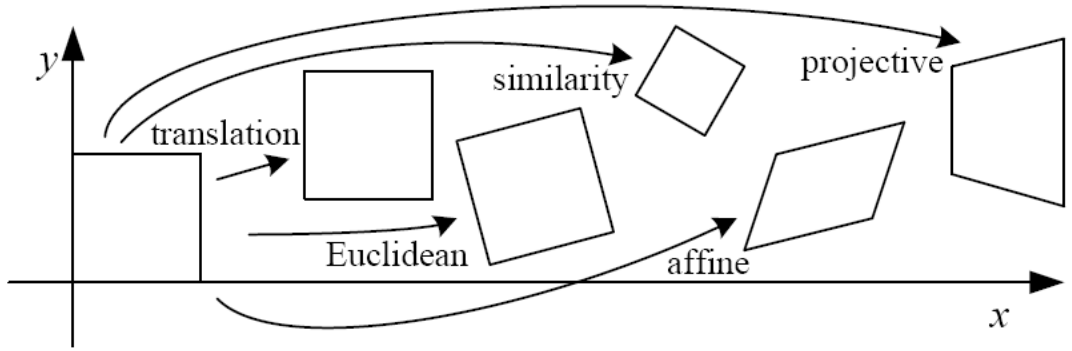
Homographies

- Homographies …
  - Affine transformations, and
  - Projective warps

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

- Properties of projective transformations:
  - Origin does not necessarily map to origin
  - Lines map to lines
  - Parallel lines do not necessarily remain parallel
  - Ratios are not preserved
  - Closed under composition

2D image transformations



Name	Matrix	# D.O.F.	Preserves:	Icon
translation	$\left[\begin{array}{c\|c} \boldsymbol{I} & \boldsymbol{t} \end{array}\right]_{2\times3}$	2	orientation $+\cdots$	
rigid (Euclidean)	$\left[\begin{array}{c\|c} \boldsymbol{R} & \boldsymbol{t} \end{array}\right]_{2\times3}$	3	lengths $+\cdots$	
similarity	$\left[\begin{array}{c\|c} s\boldsymbol{R} & \boldsymbol{t} \end{array}\right]_{2\times3}$	4	angles $+\cdots$	
affine	$\left[\begin{array}{c} \boldsymbol{A} \end{array}\right]_{2\times3}$	6	parallelism $+\cdots$	
projective	$\left[\begin{array}{c} \tilde{\boldsymbol{H}} \end{array}\right]_{3\times3}$	8	straight lines	

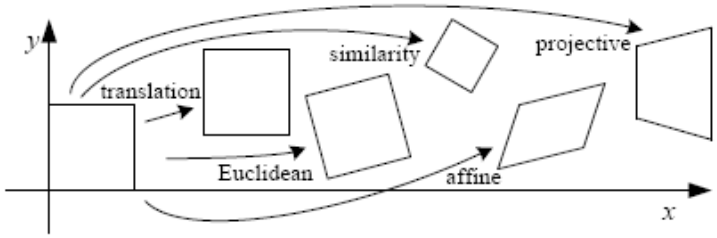
These transformations are a nested set of groups

- Closed under composition and inverse is a member

2D image transformations

Which transform is the right one for warping PP1 into PP2?

e.g. translation, Euclidean, affine, projective

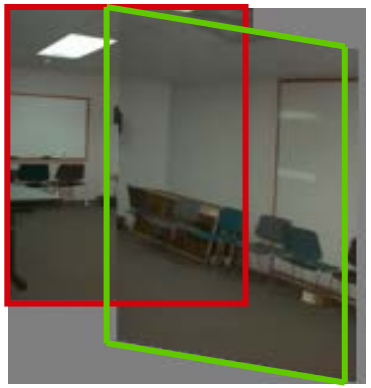


Translation

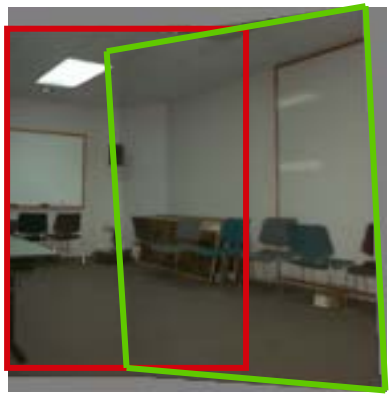
Affine

Perspective

2 unknowns

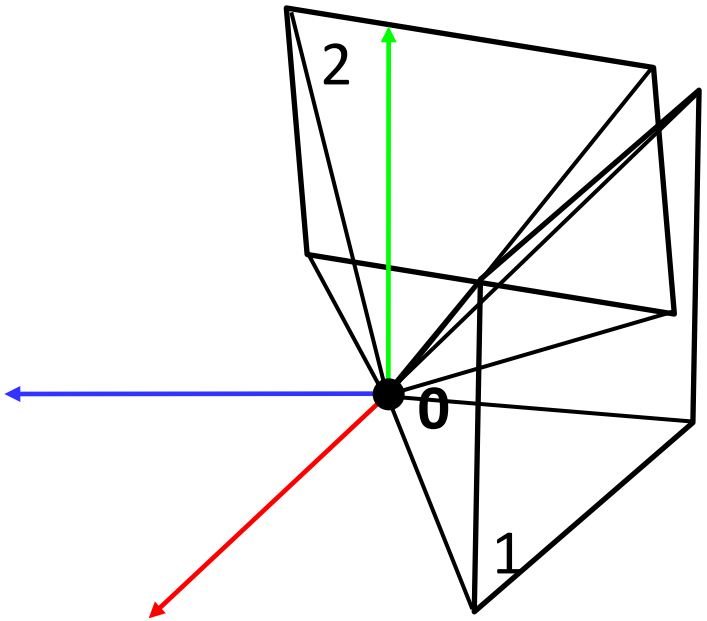


6 unknowns

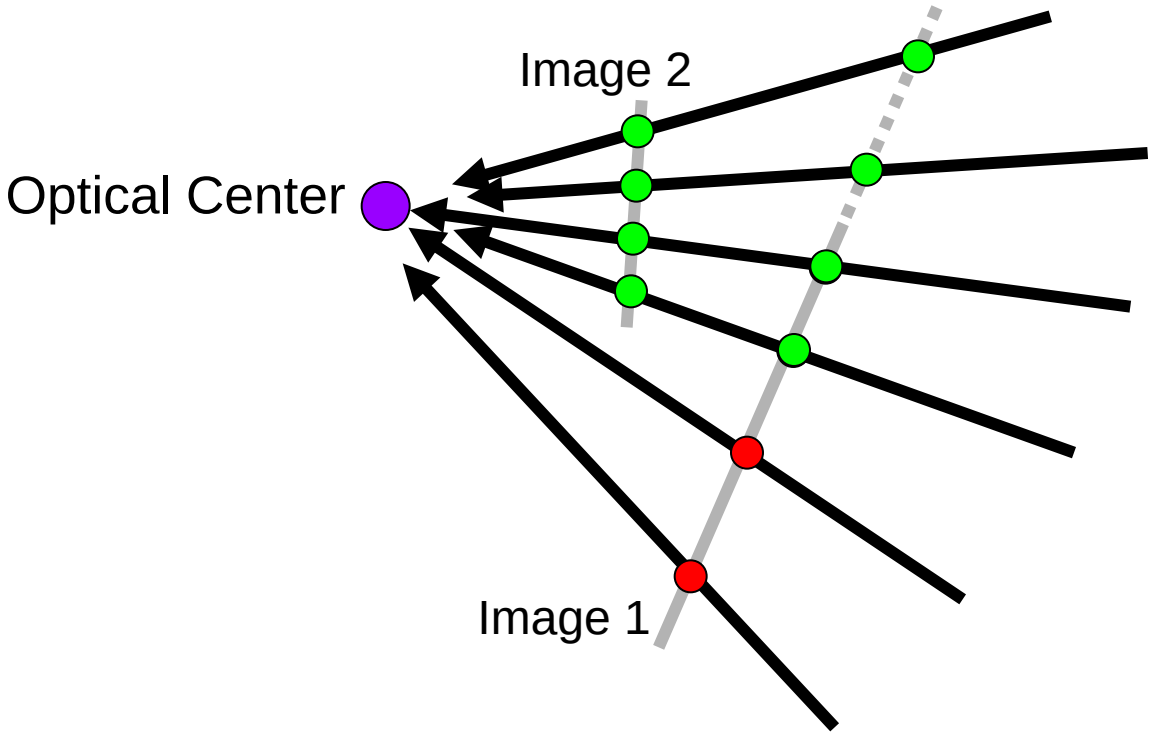


8 unknowns

Geometric setup of mosaics

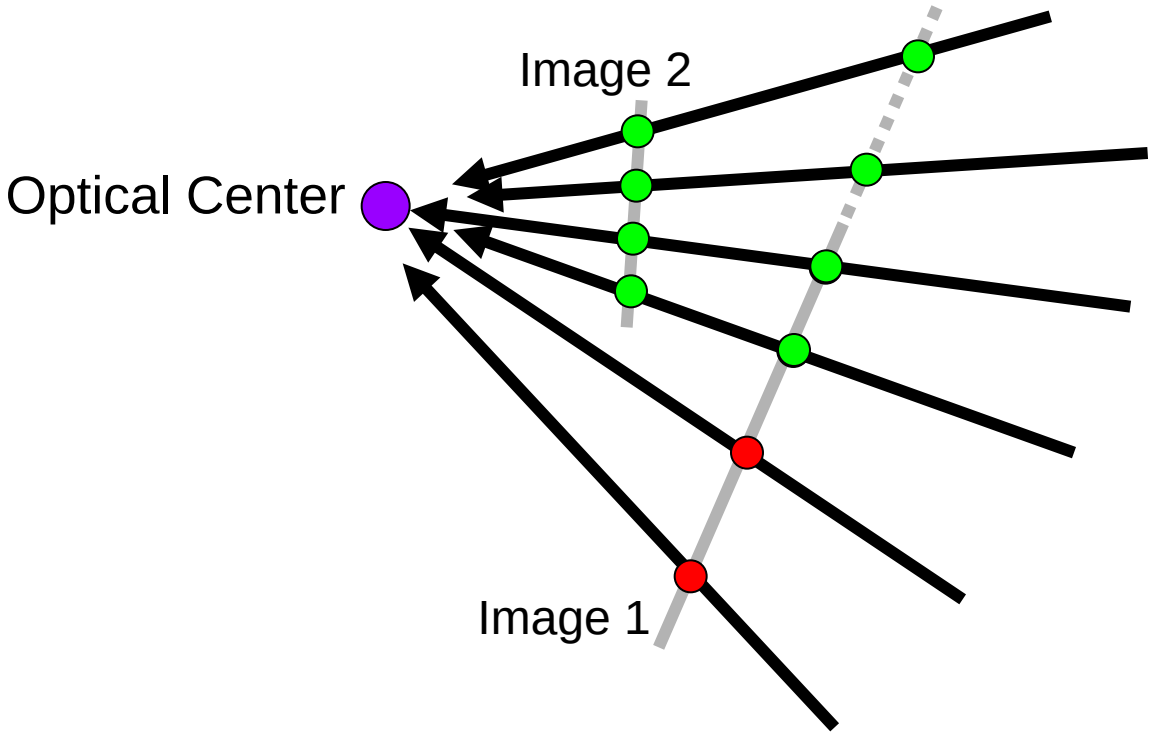


What is the transformation?



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

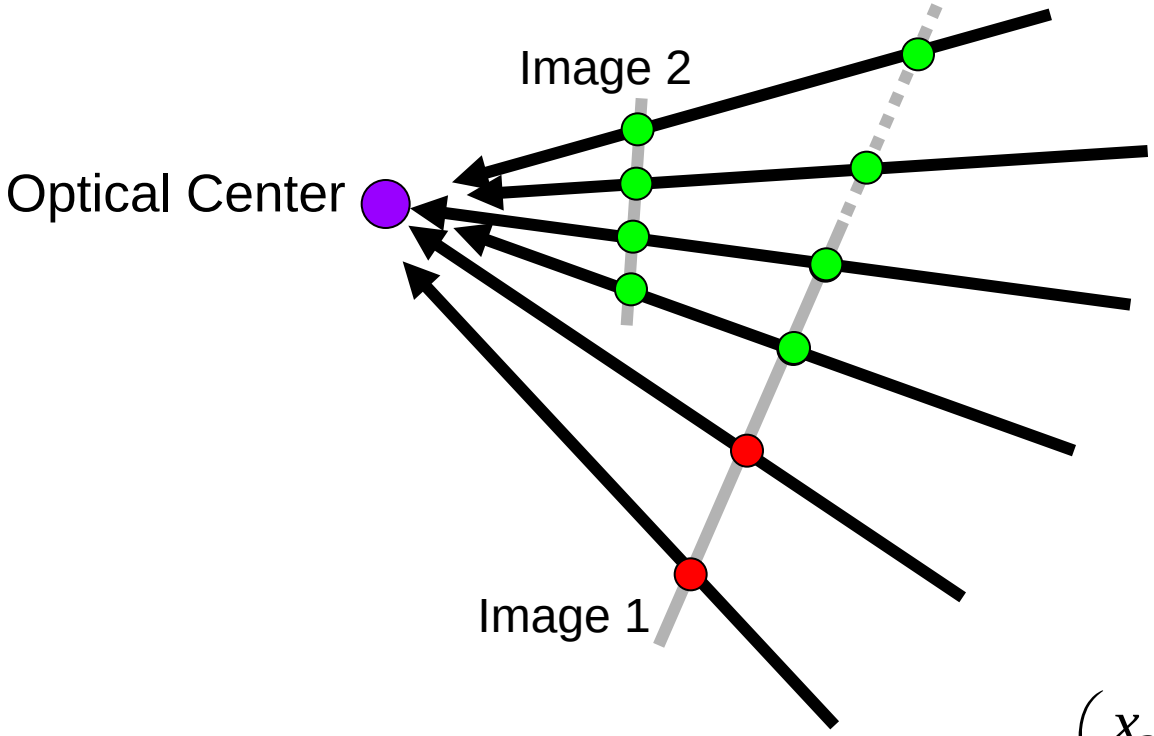
What is the transformation?



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

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

What is the transformation?



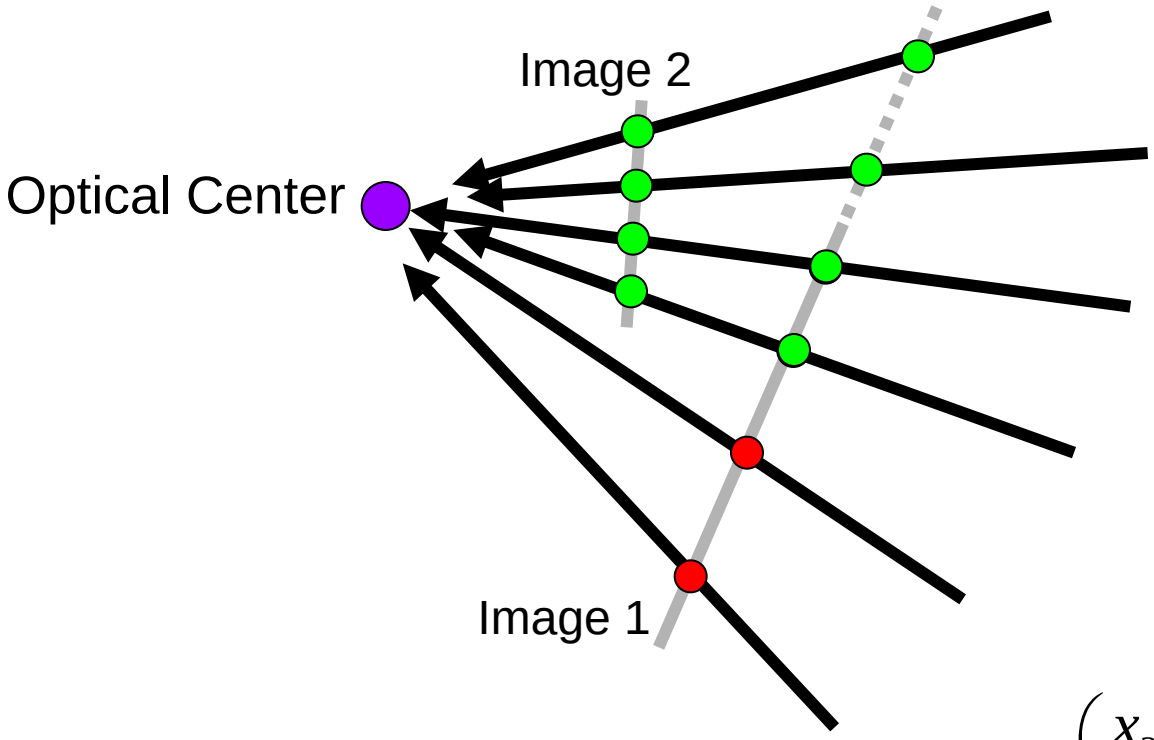
$$\begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix} \sim \mathbf{K}_1 \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

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

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

3x3 matrix
also called Homography

What is the transformation?



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

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

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

Project from image to 3D ray

Rotate the ray by camera motion

Project back into new (source) image

3x3 matrix

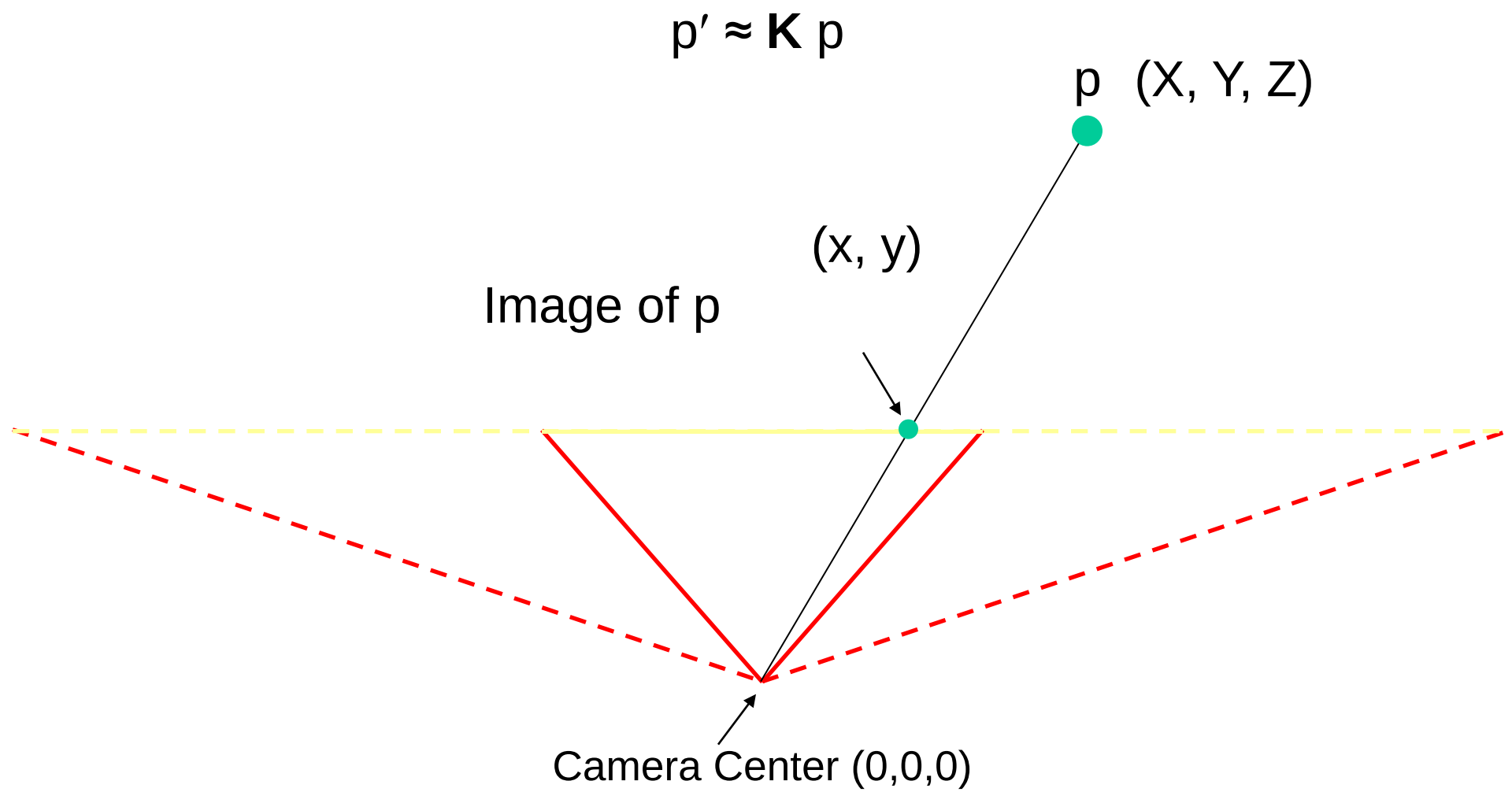
also called Homography

Combining Images into Panoramas

- Theorem: Any 2 images of an arbitrary scene taken with *same* camera center are related by $p_2 \approx \mathbf{K}\,\mathbf{R}\,\mathbf{K}^{-1}\,p_1$ where p_1 and p_2 are homogeneous coords of 2 corresponding points, **K** is 3 x 3 camera calibration matrix, and **R** is 3 x 3 rotation matrix

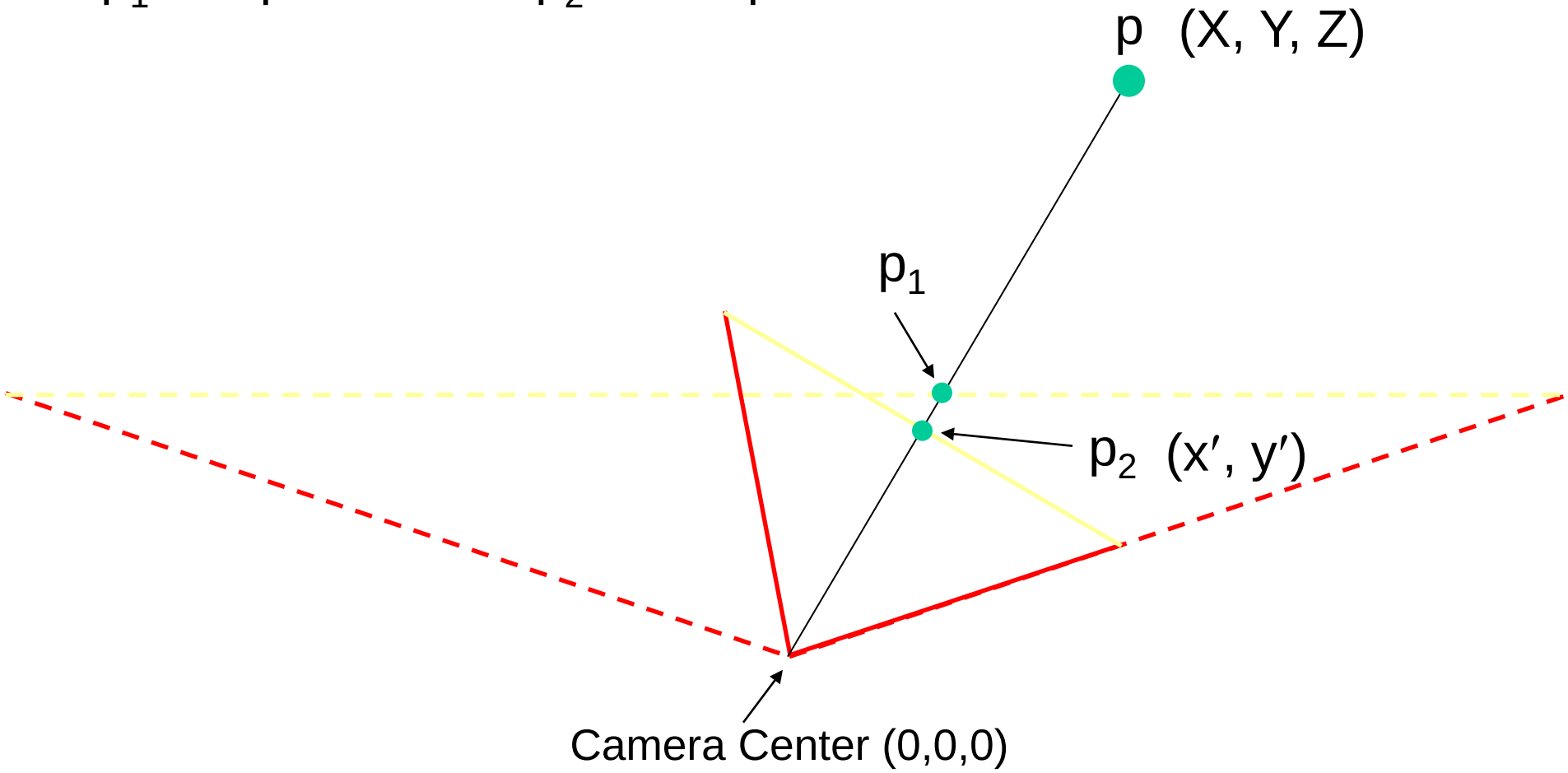
- $\mathbf{K}\,\mathbf{R}\,\mathbf{K}^{-1}$ is 3 x 3 matrix called the "homography induced by the plane at infinity"

Another example



Another example

$p_1 \approx \mathbf{K}\, p$ $p_2 \approx \mathbf{K}\, \mathbf{R}\, p$



Another example

$p_1 \approx \mathbf{K}\, p$ $\qquad$ $p_2 \approx \mathbf{K}\, \mathbf{R}\, p$

$\mathbf{K}^{-1}\, p_1 \approx p$

$p_2 \approx \mathbf{K}\, \mathbf{R}\, \mathbf{K}^{-1}\, p_1$

homography

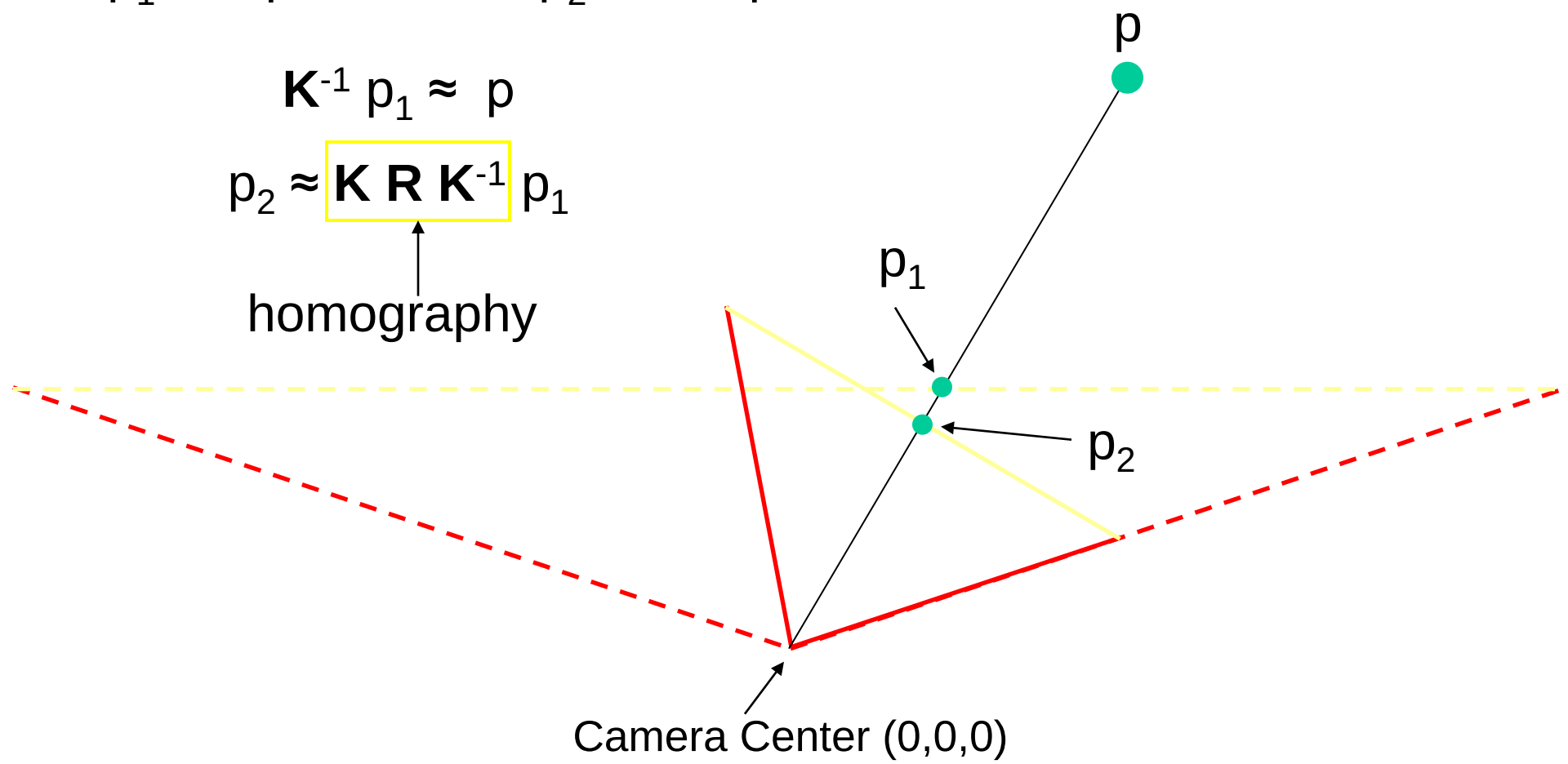
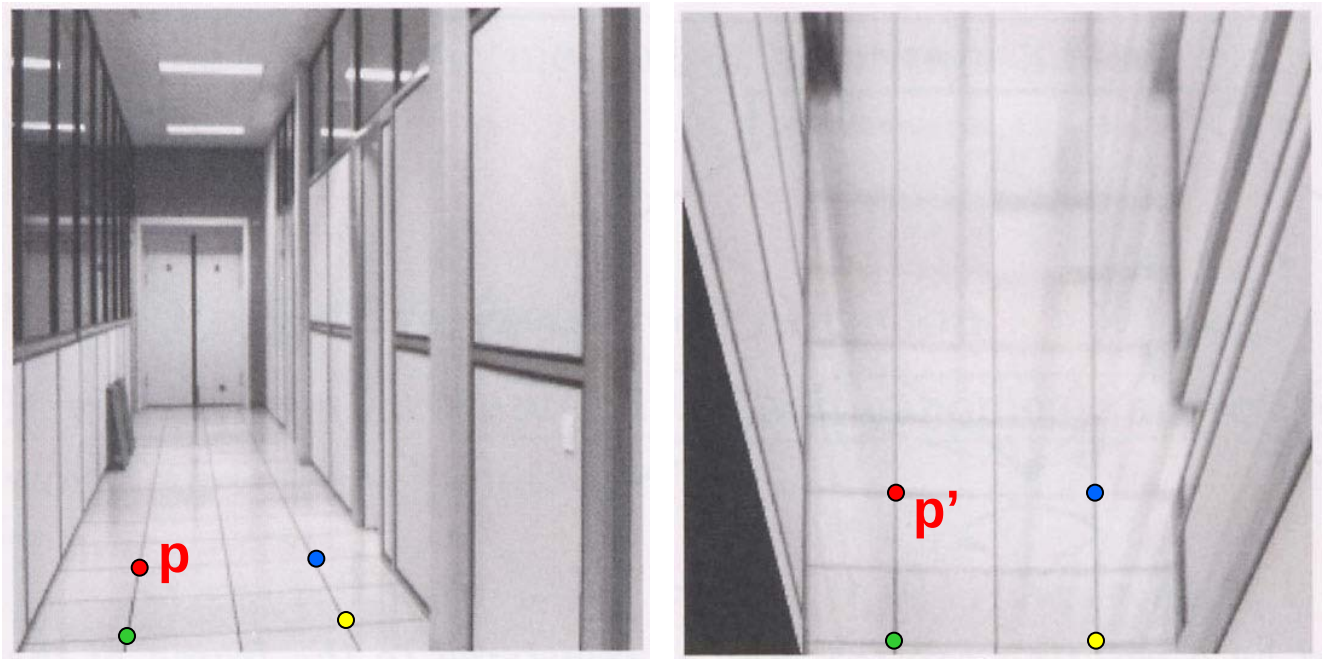


Image rectification



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

$$\mathbf{p'} = \mathbf{Hp}$$

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

Can set scale factor *i*=1 and *w=1*. So, there are 8 unknowns.
Set up a system of linear equations:

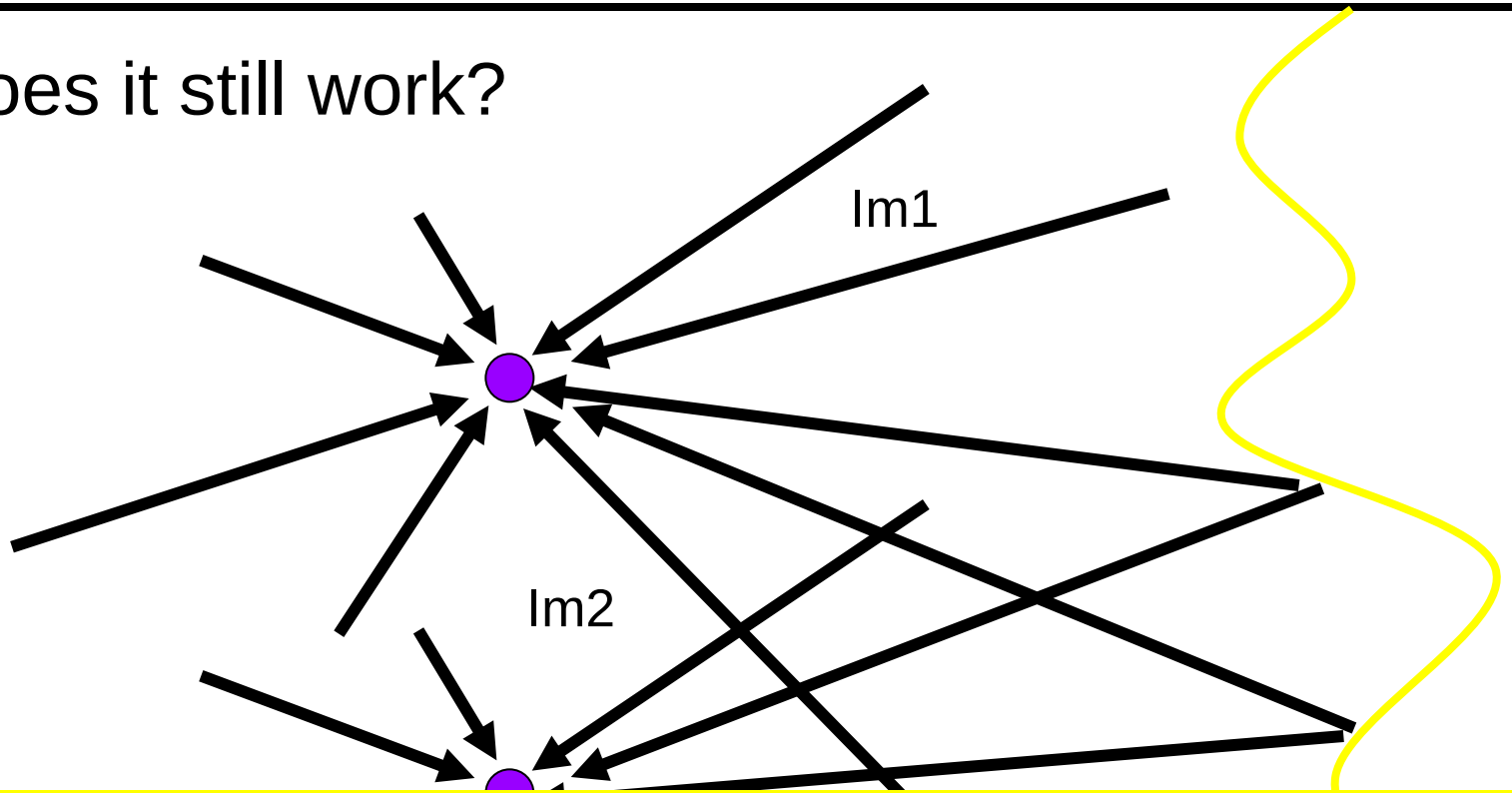
$$\mathbf{Ah = b}$$

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:

Can be done in Matlab $\min \|Ah - b\|^2$

changing camera center

- Does it still work?

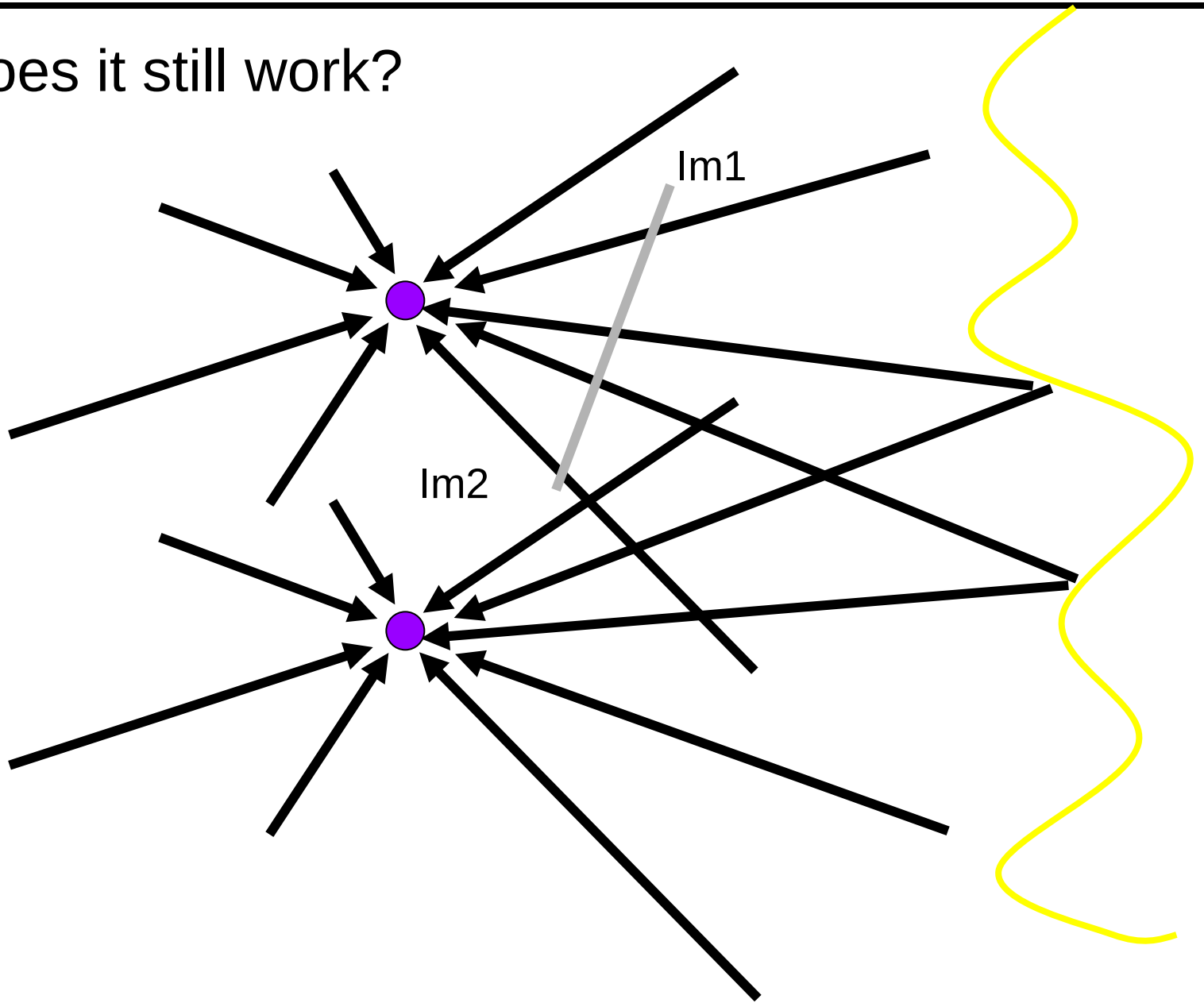


The mosaic has a natural interpretation in 3D

- The images are reprojected onto a common plane
- The mosaic is formed on this plane
- Mosaic is a *synthetic wide-angle camera*

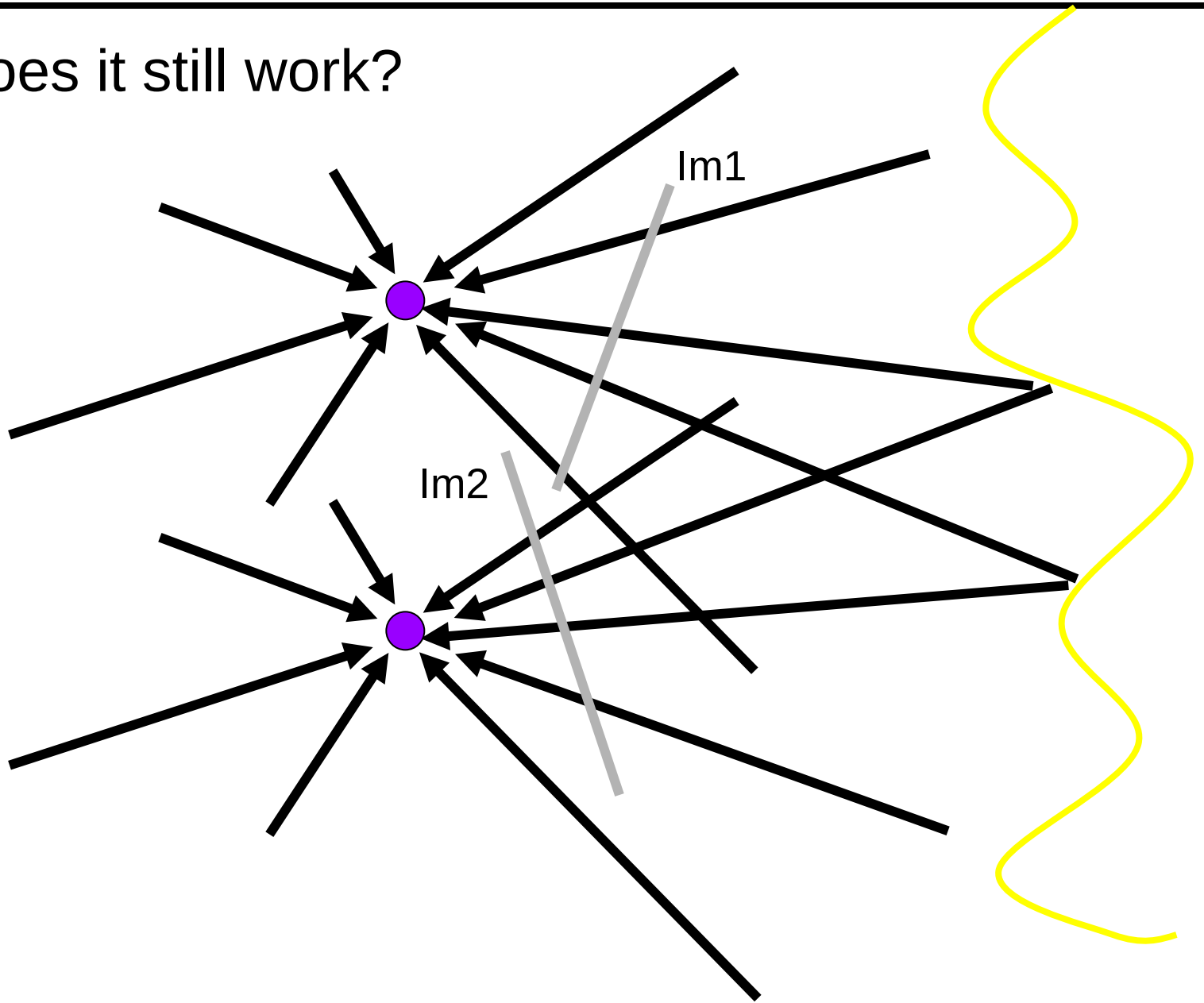
changing camera center

- Does it still work?



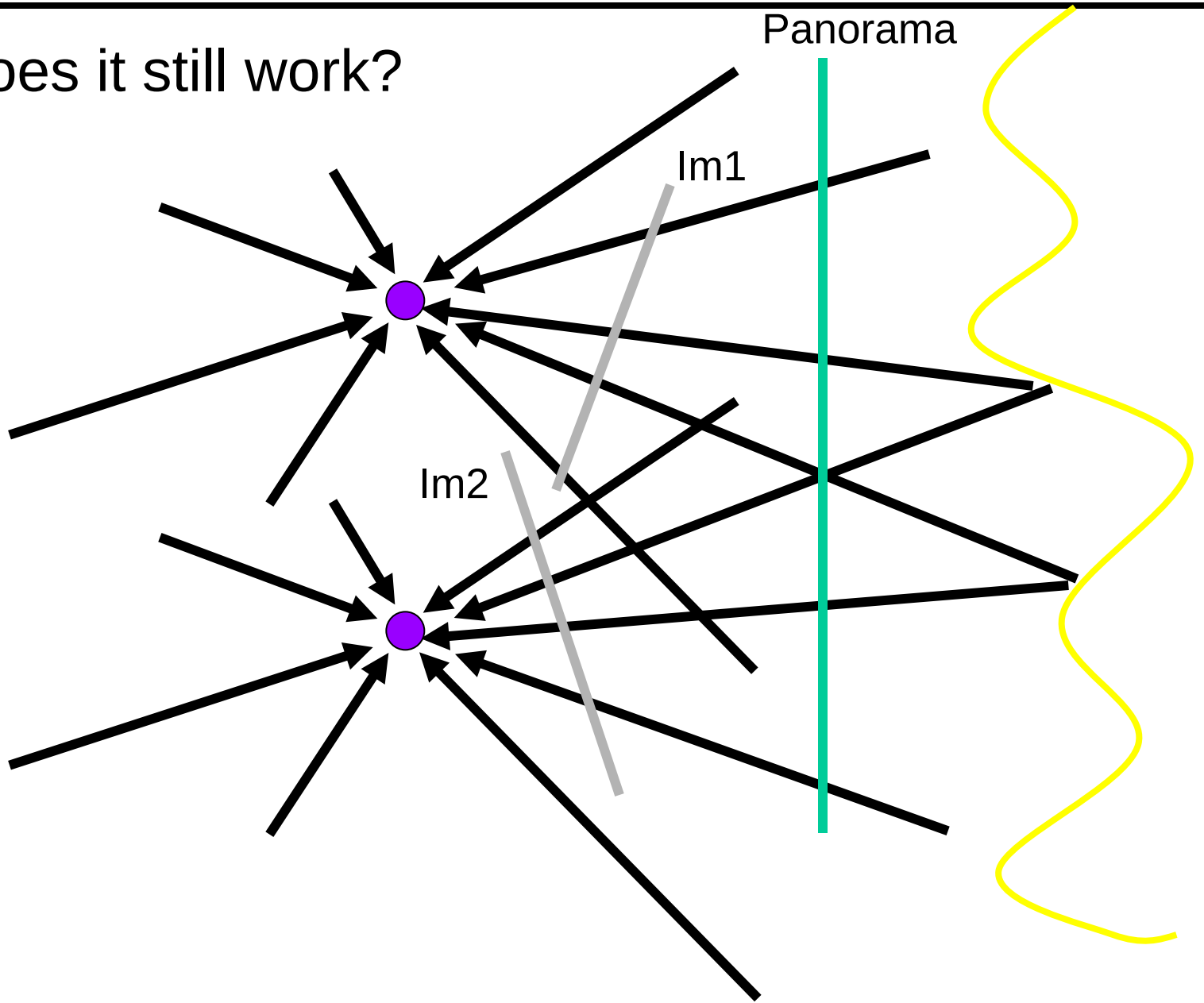
changing camera center

- Does it still work?



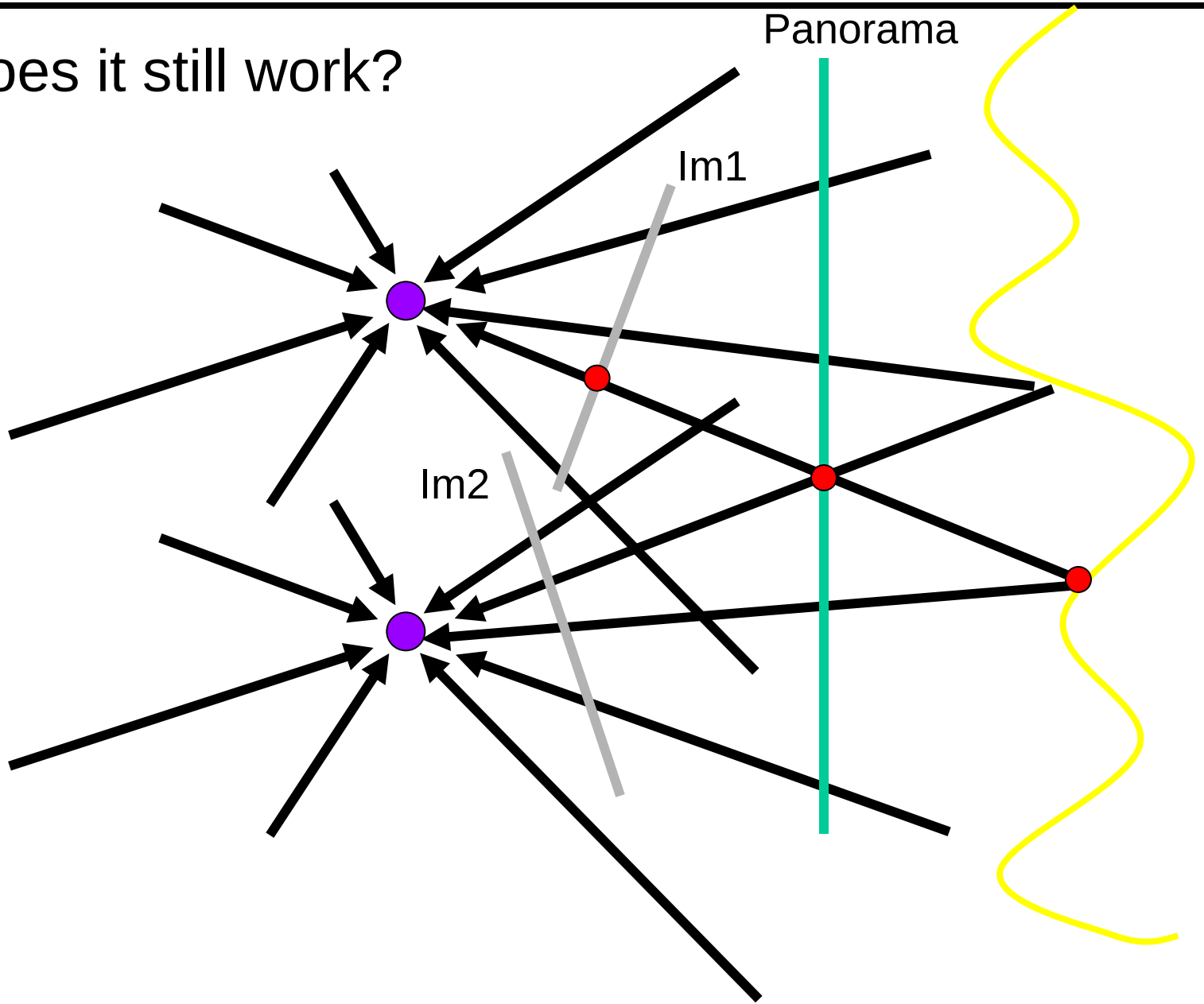
changing camera center

- Does it still work?



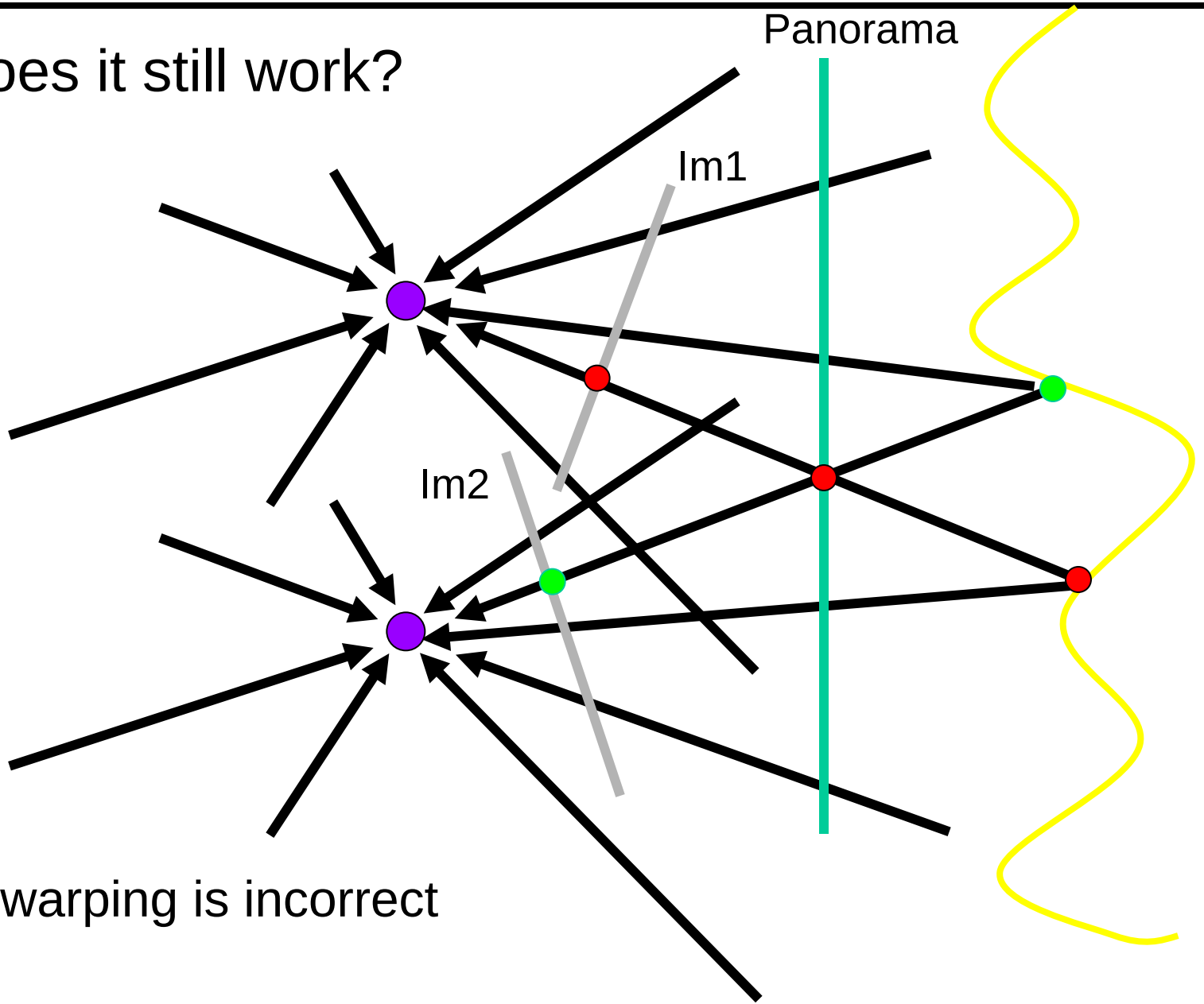
changing camera center

- Does it still work?



changing camera center

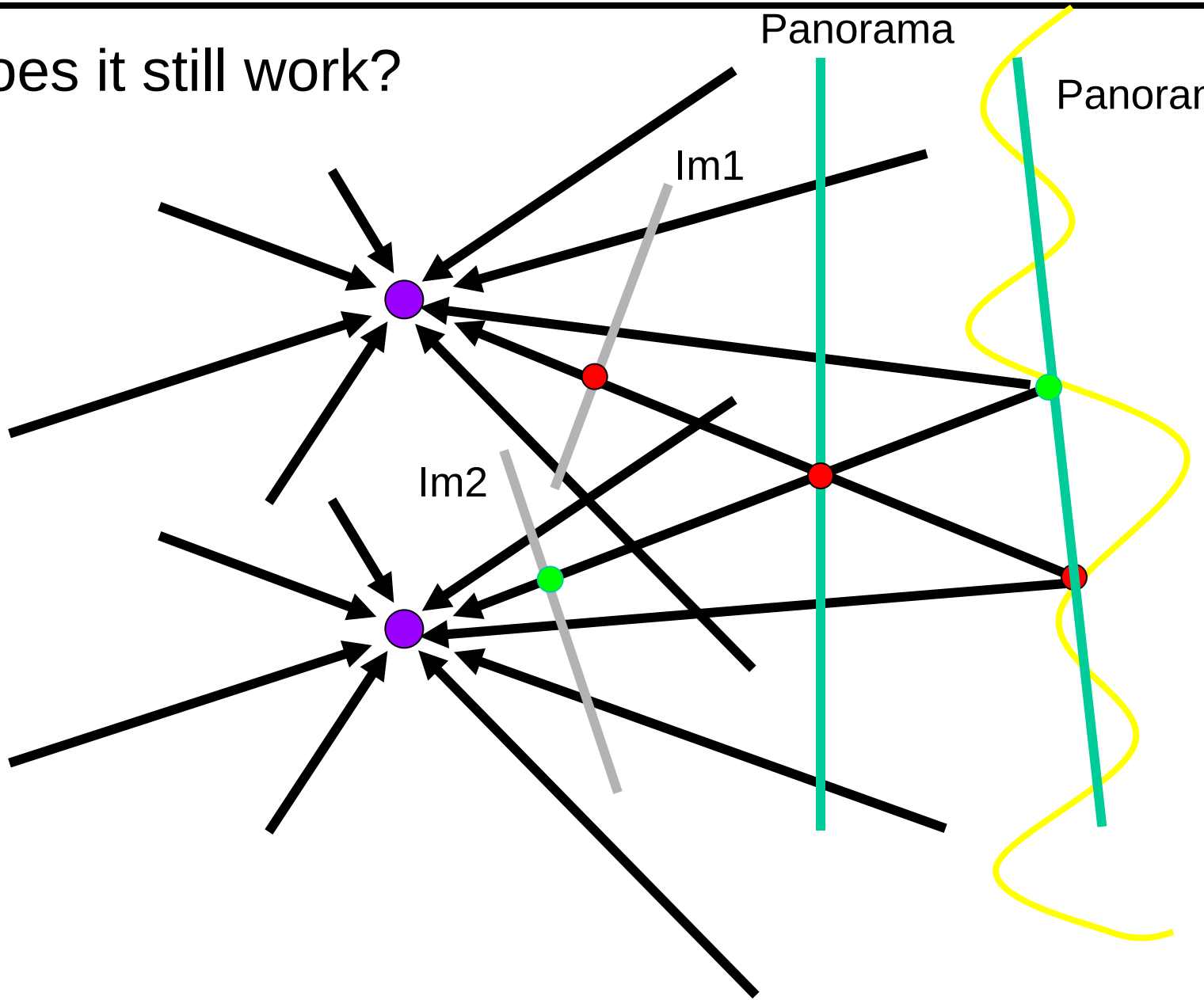
- Does it still work?



Your warping is incorrect

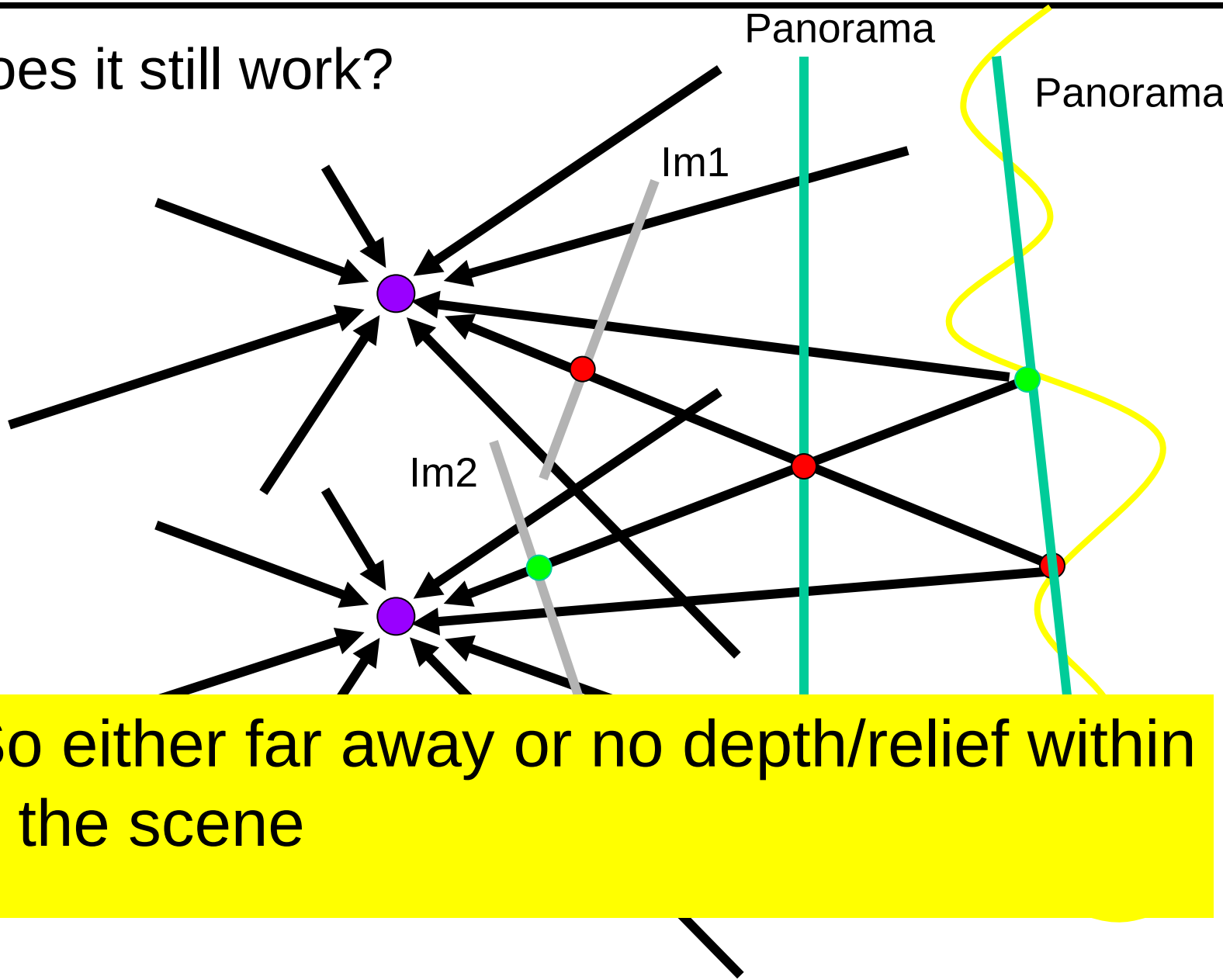
changing camera center

- Does it still work?



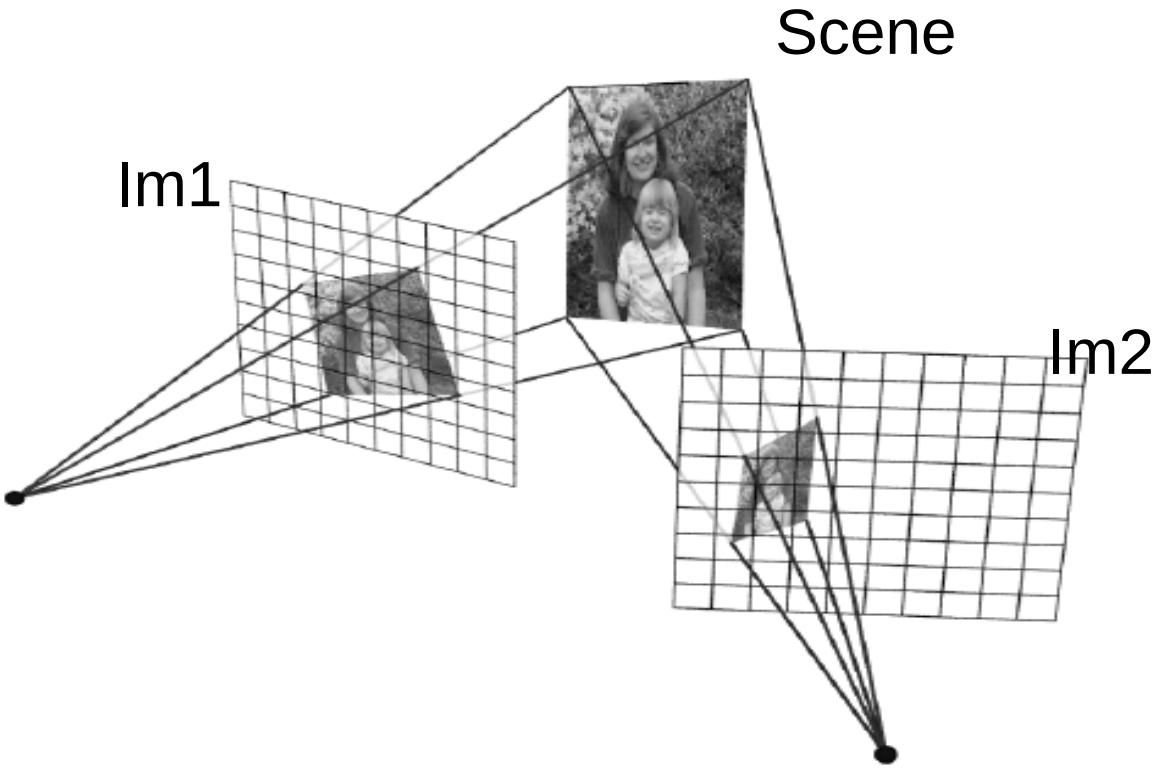
changing camera center

- Does it still work?



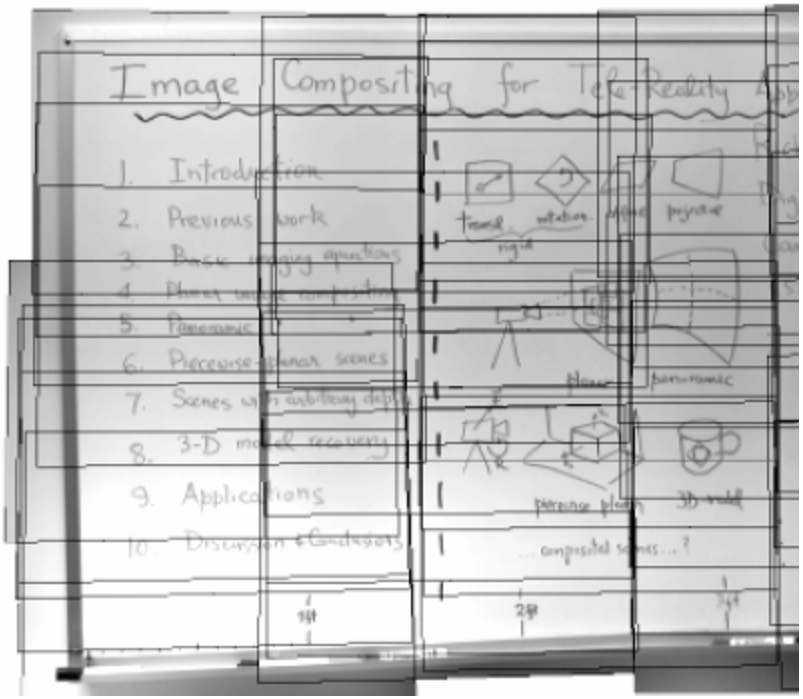
So either far away or no depth/relief within the scene

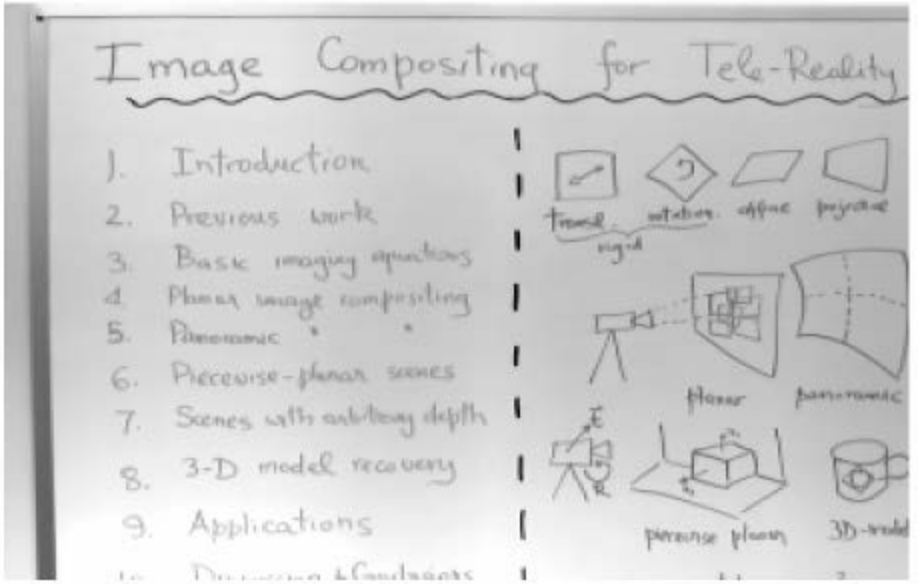
Planar scene (far away)



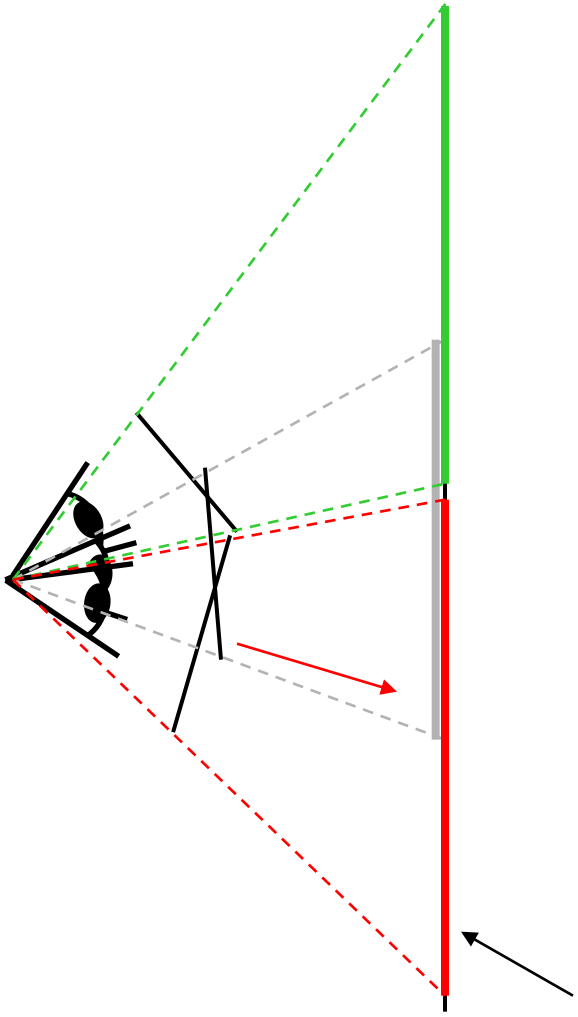
- If scene is planar, we are OK!
- This is how big aerial photographs are made

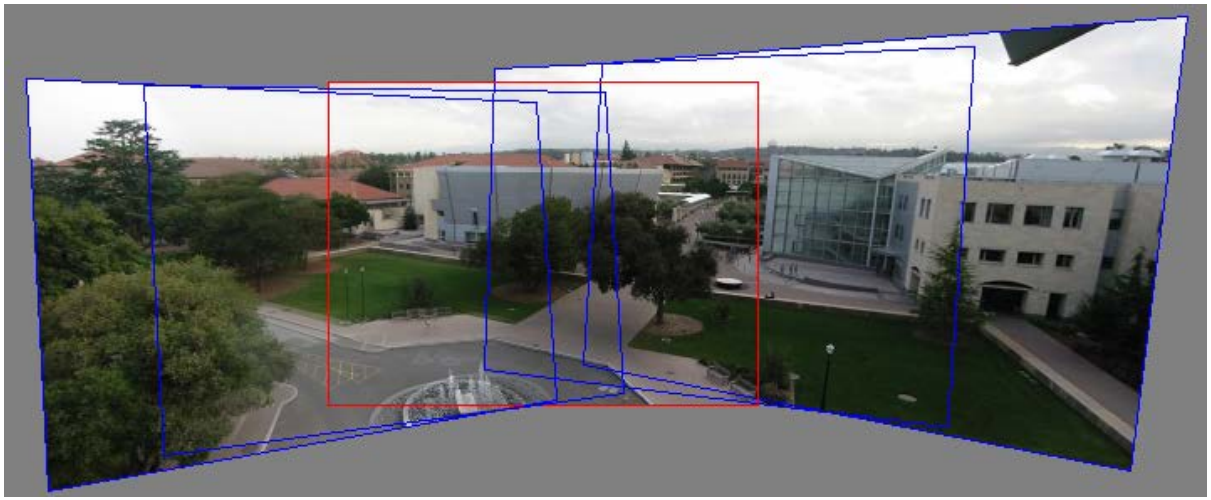
Planar mosaic Examples



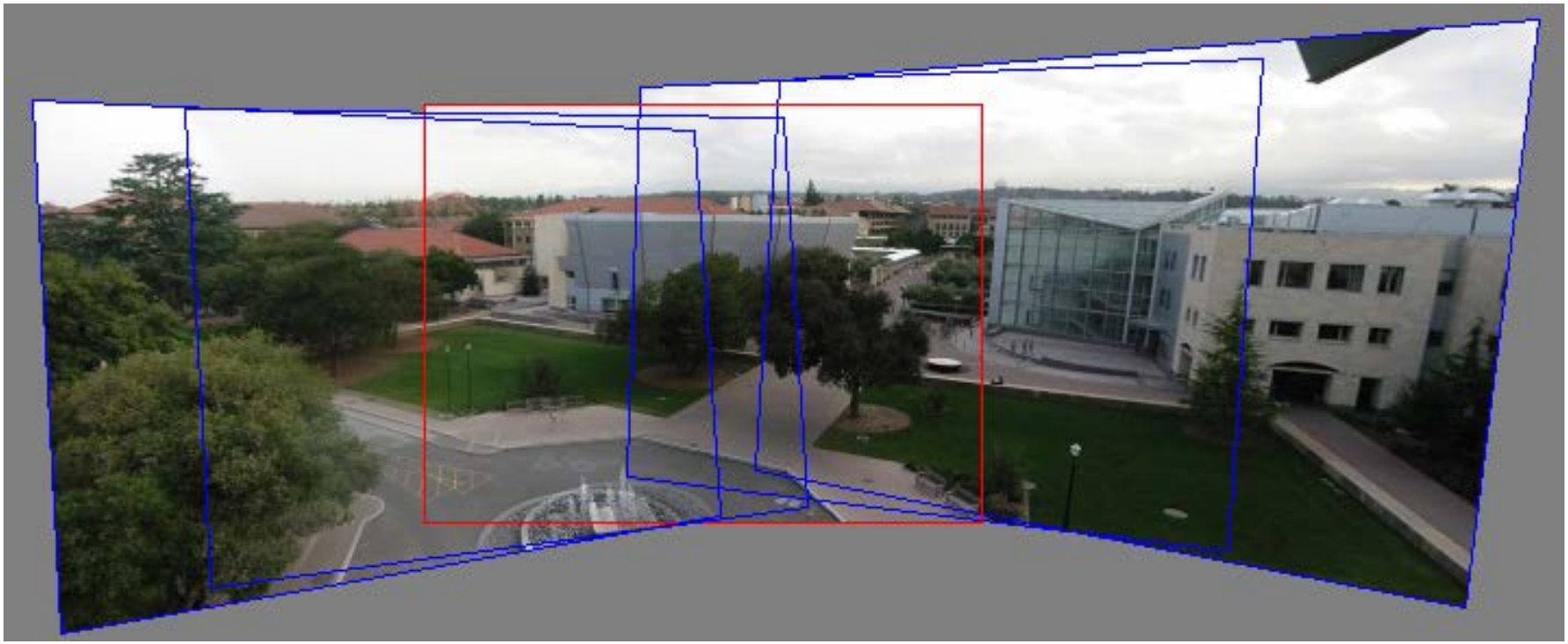


Projecting images onto a common plane





Can use homography to create a 360 panorama

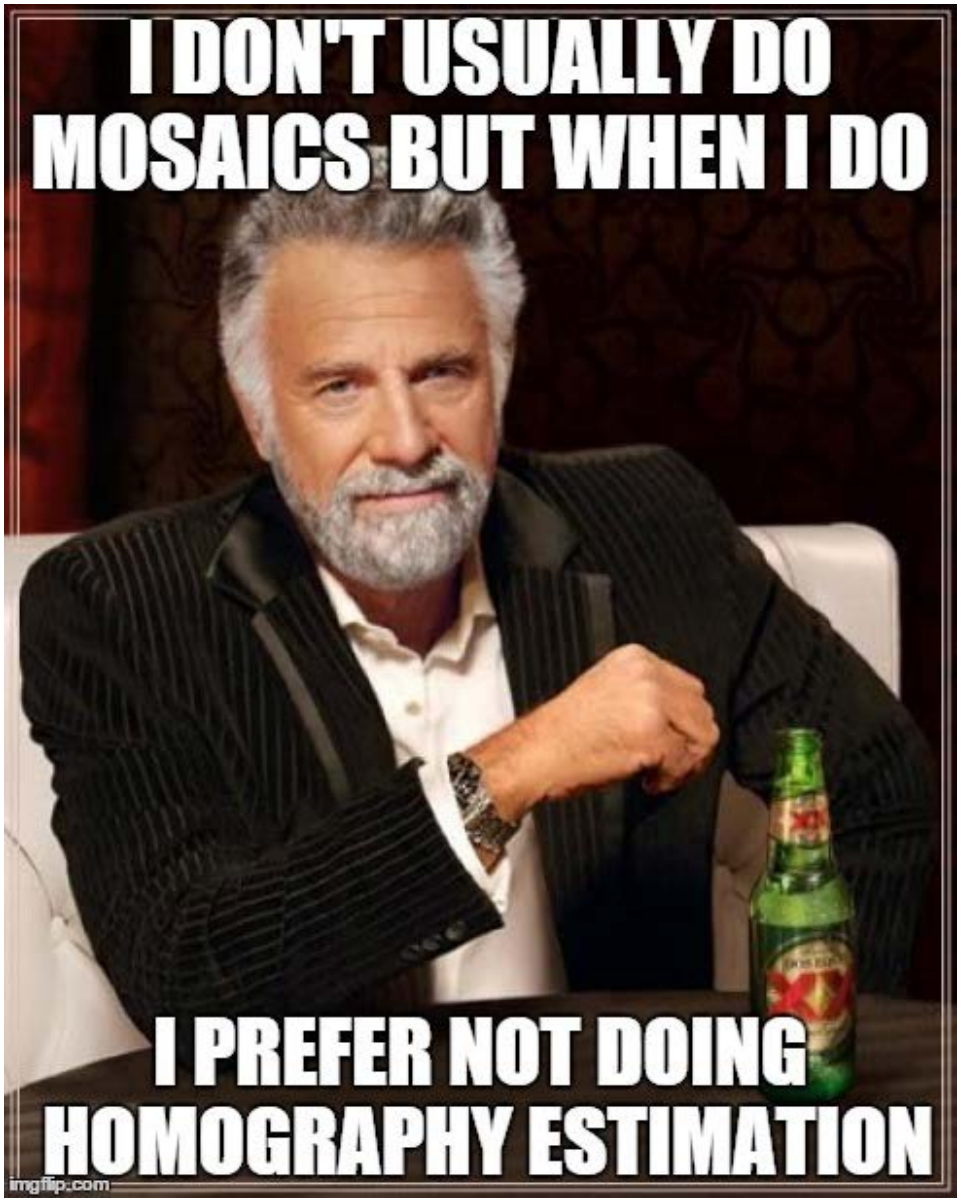


1. Pick one image (red)
2. Warp the other images towards it (usually, one by one)
3. blend

Recap:

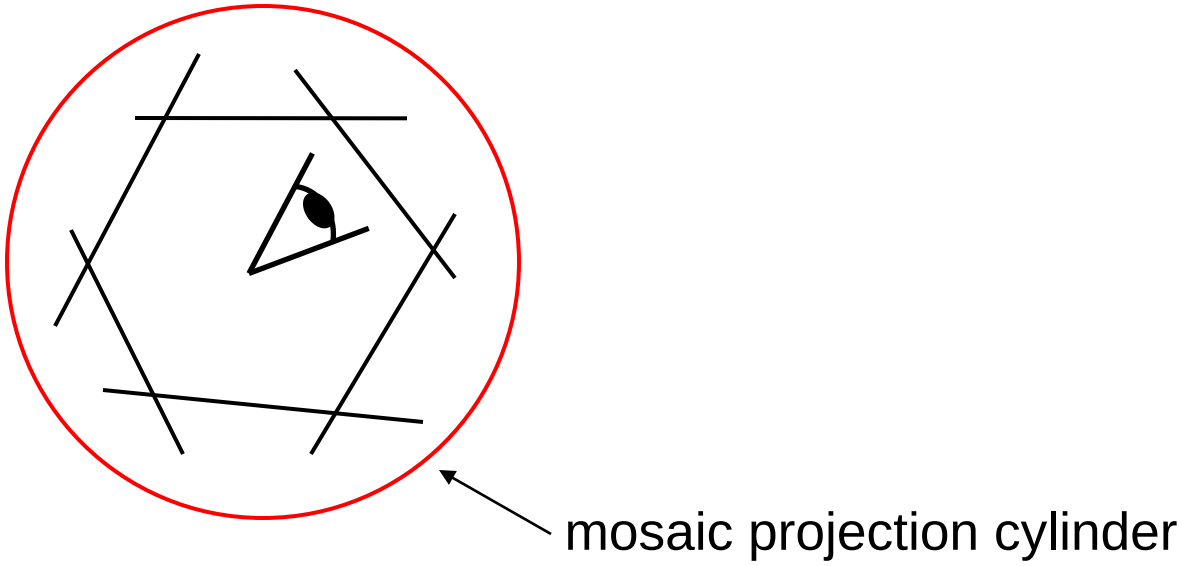
- With enough images from the same optical center, we can create panorama.
- If the camera moves, we can't in general
- If the scene is planar or faraway, we are OK.

Moving on…

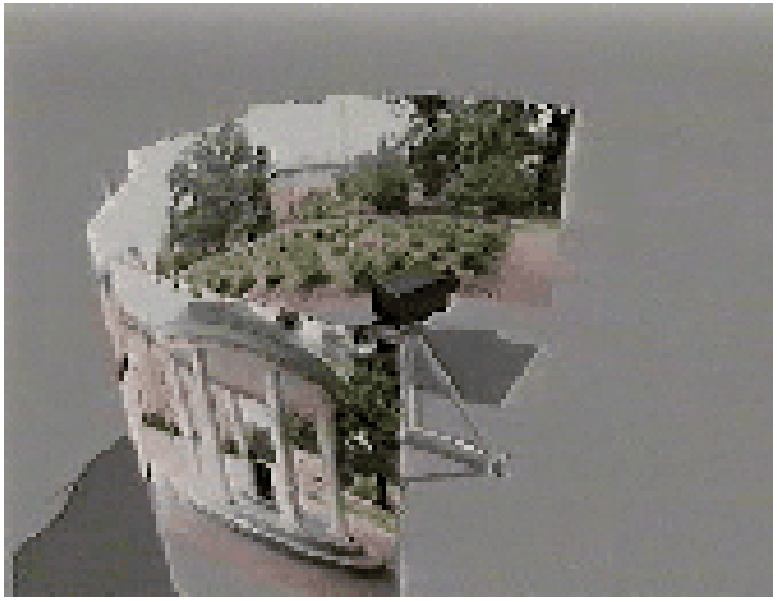

I DON'T USUALLY DO
MOSAICS BUT WHEN I DO
I PREFER NOT DOING
HOMOGRAPHY ESTIMATION
imgflip.com

Should use Cylindrical Projection

When we cannot choose a “reference” image (the red one) to map everything else to…

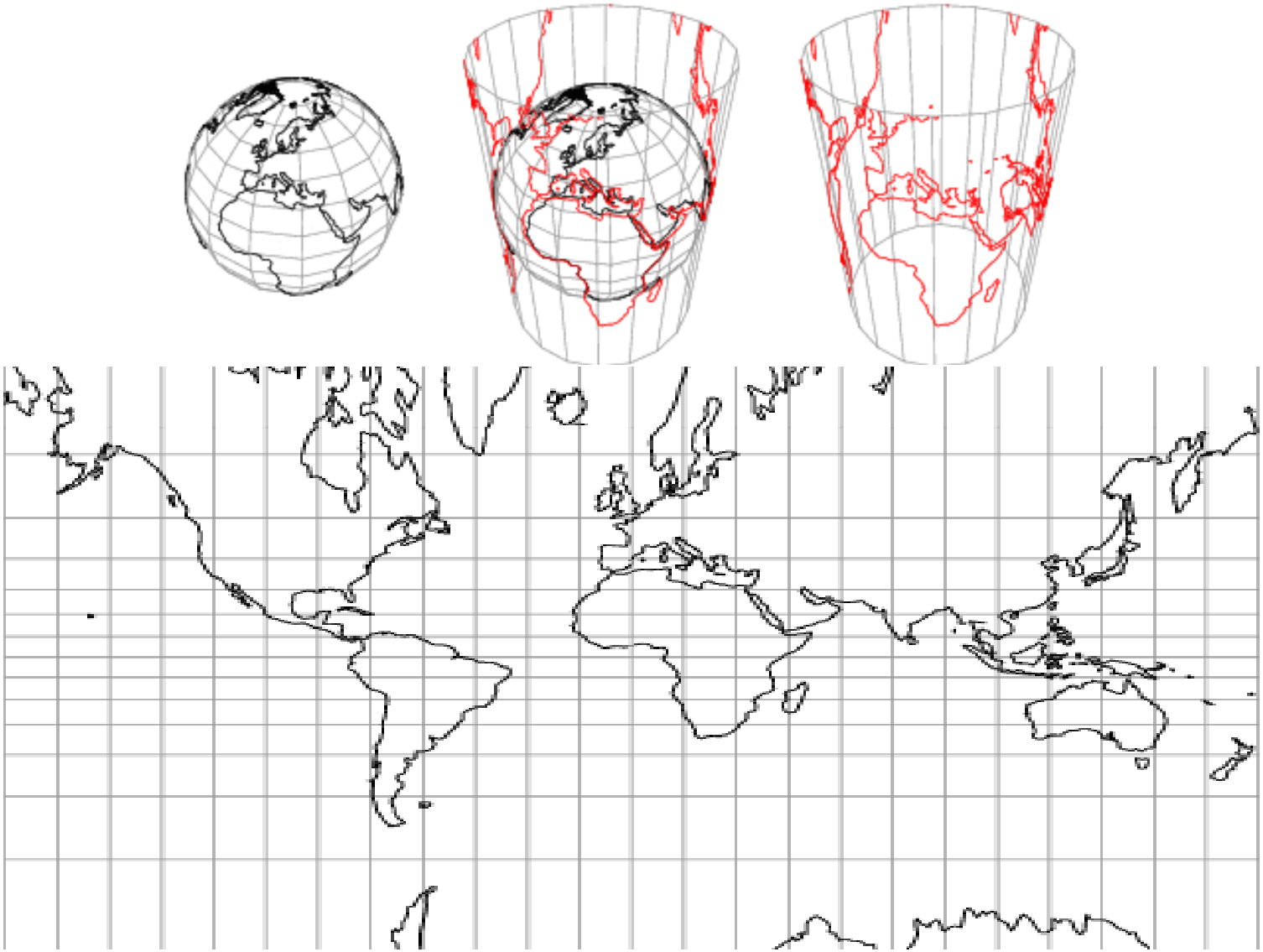


Cylindrical panoramas:



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

Cylindrical projection (An Example)



Taking pictures

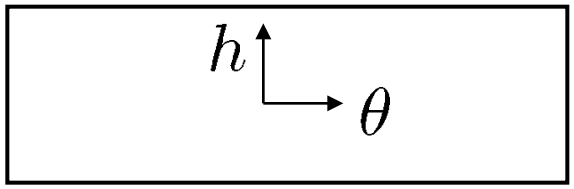



Kaidan panoramic tripod head

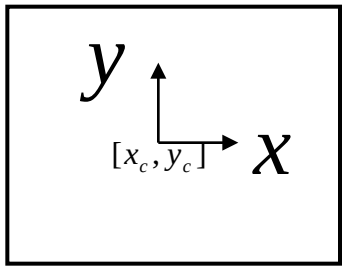
Assume principal axis is parallel to horizontal plane

Kogeto Dot 360 Camera for iPhone

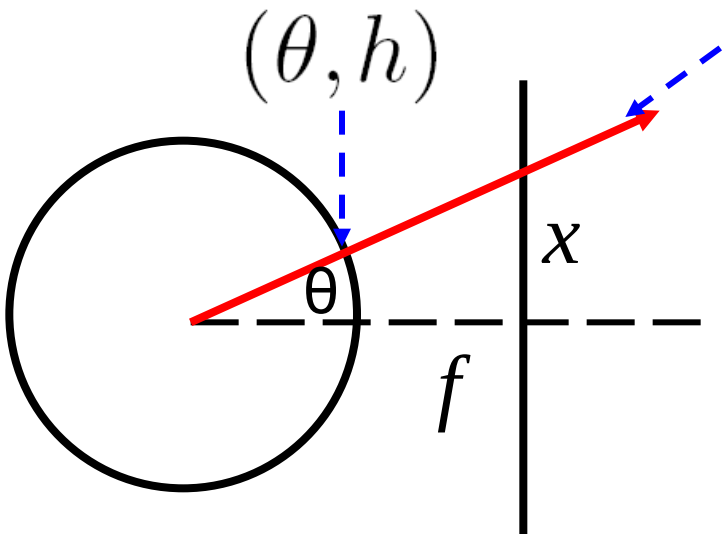
Cylindrical projection



unwrapped cylinder

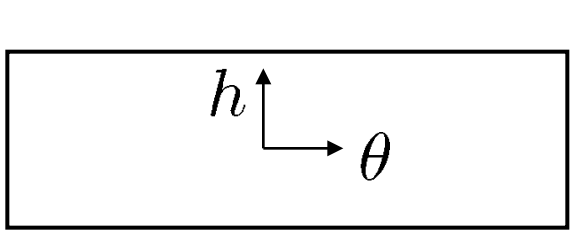


this is raised a bit by h

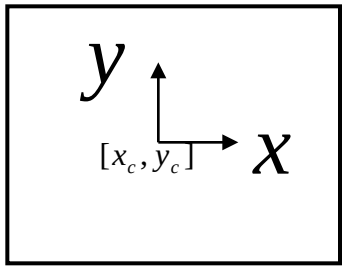


$$\theta = \tan^{-1}\frac{x}{f}$$

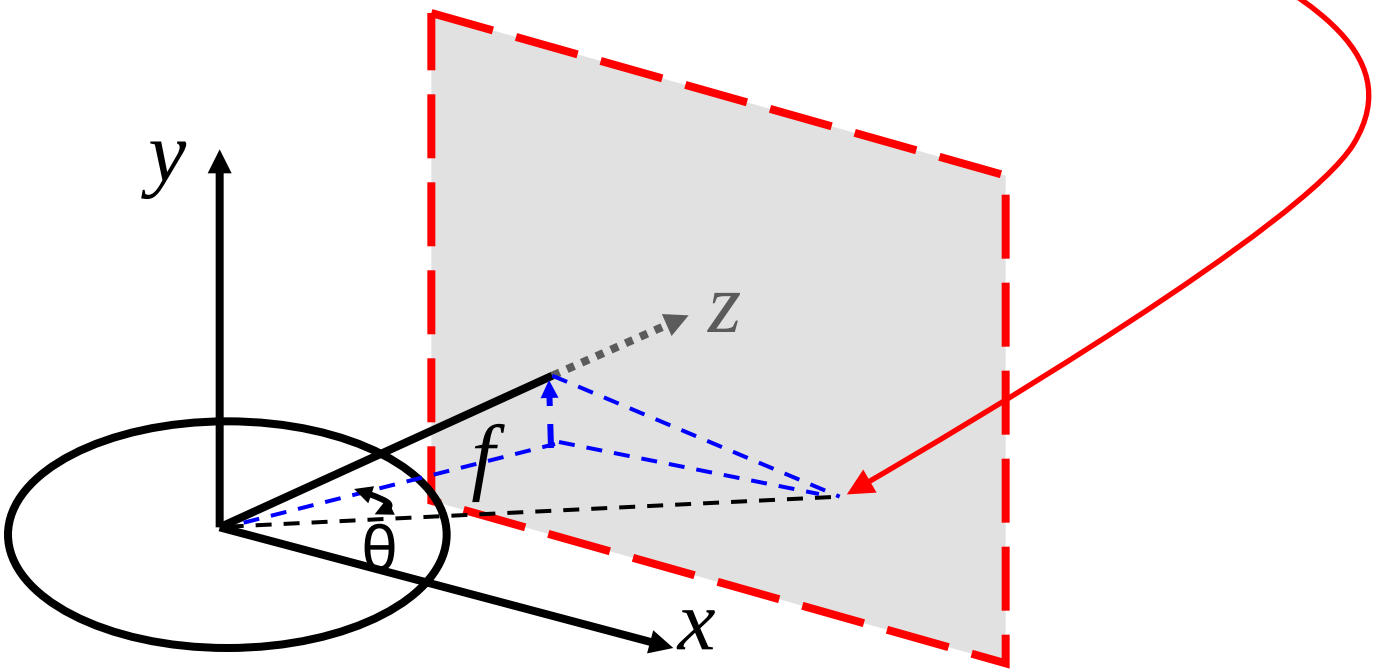
Cylindrical projection



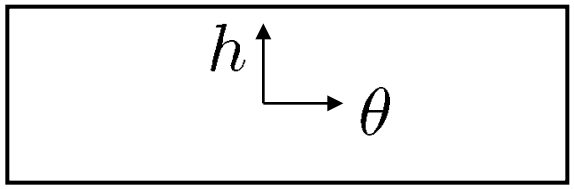
unwrapped cylinder



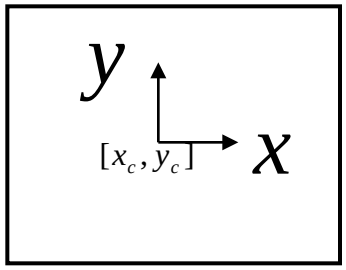
$$\propto \quad (x, y, f)$$



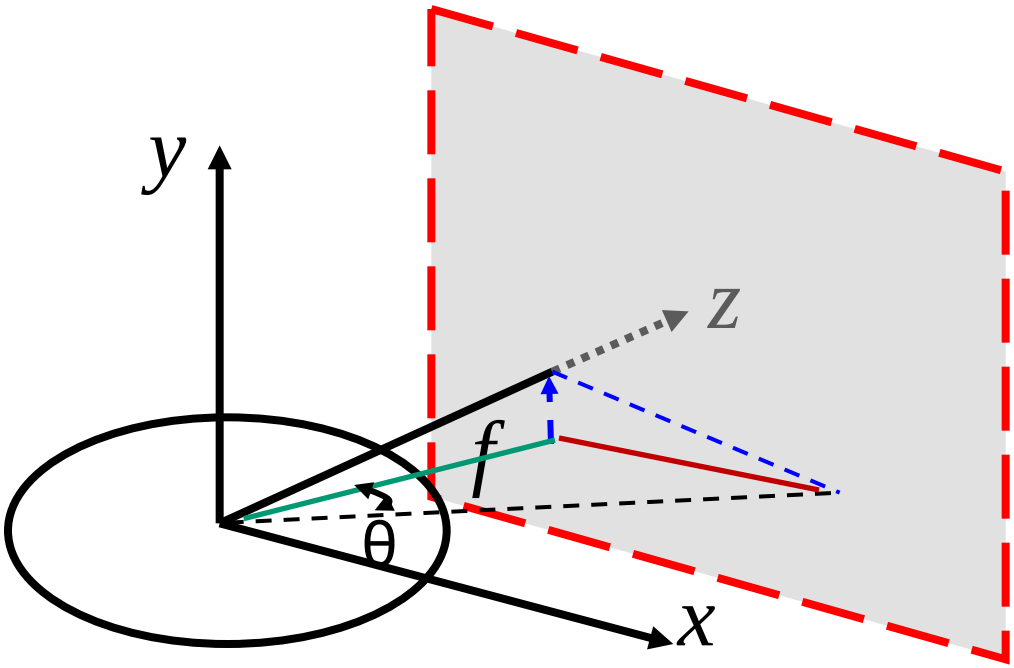
Cylindrical projection



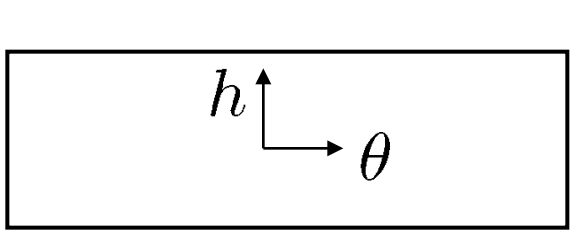
unwrapped cylinder



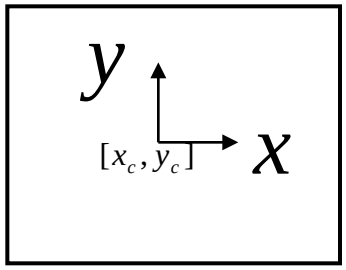
$$(\sin\theta, h, \cos\theta) \propto (x, y, f)$$



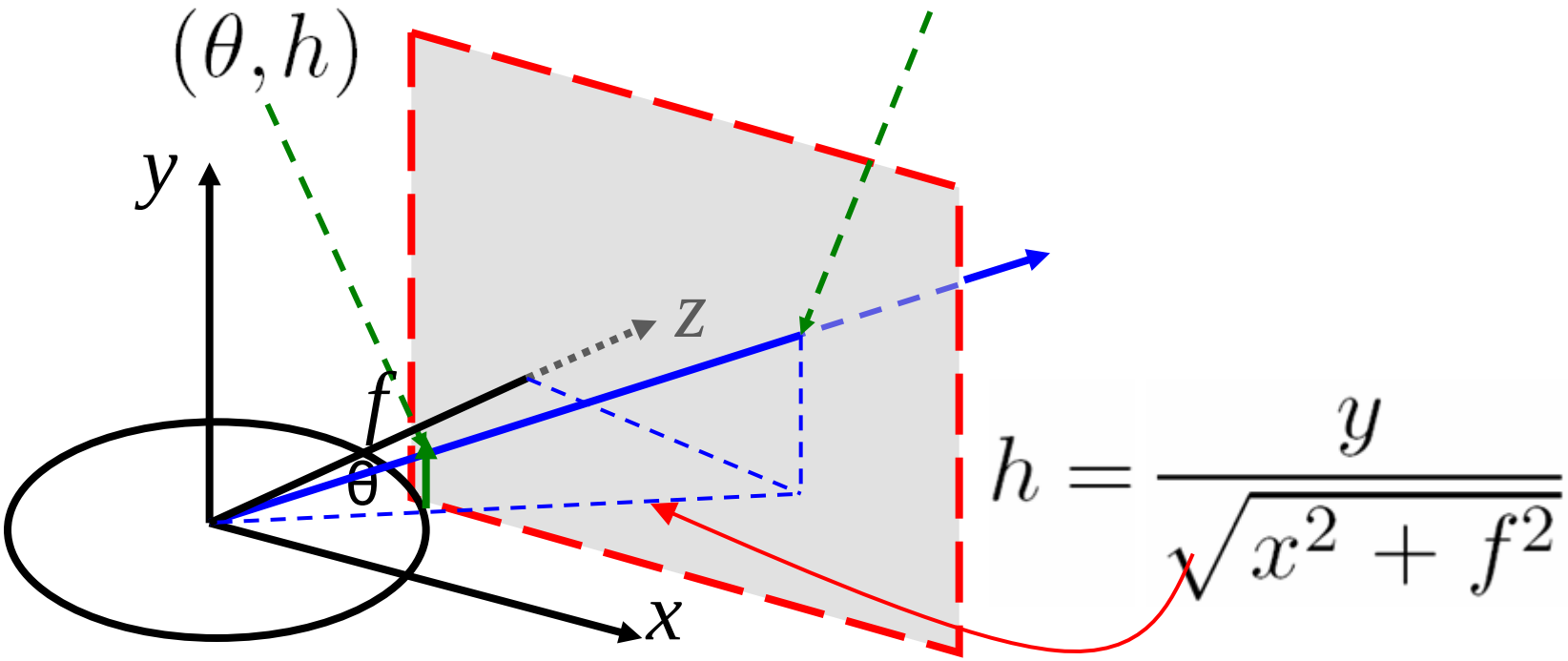
Cylindrical projection



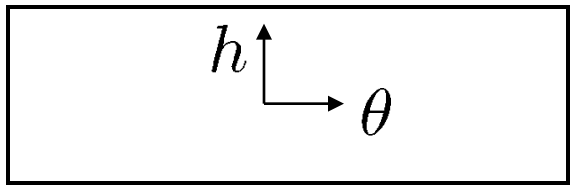
unwrapped cylinder



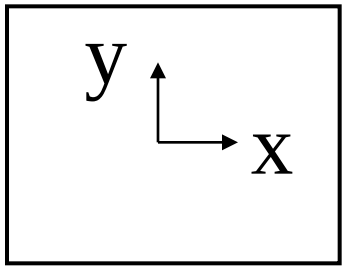
$$(\sin\theta, h, \cos\theta) \propto (x, y, f)$$



Cylindrical Projection



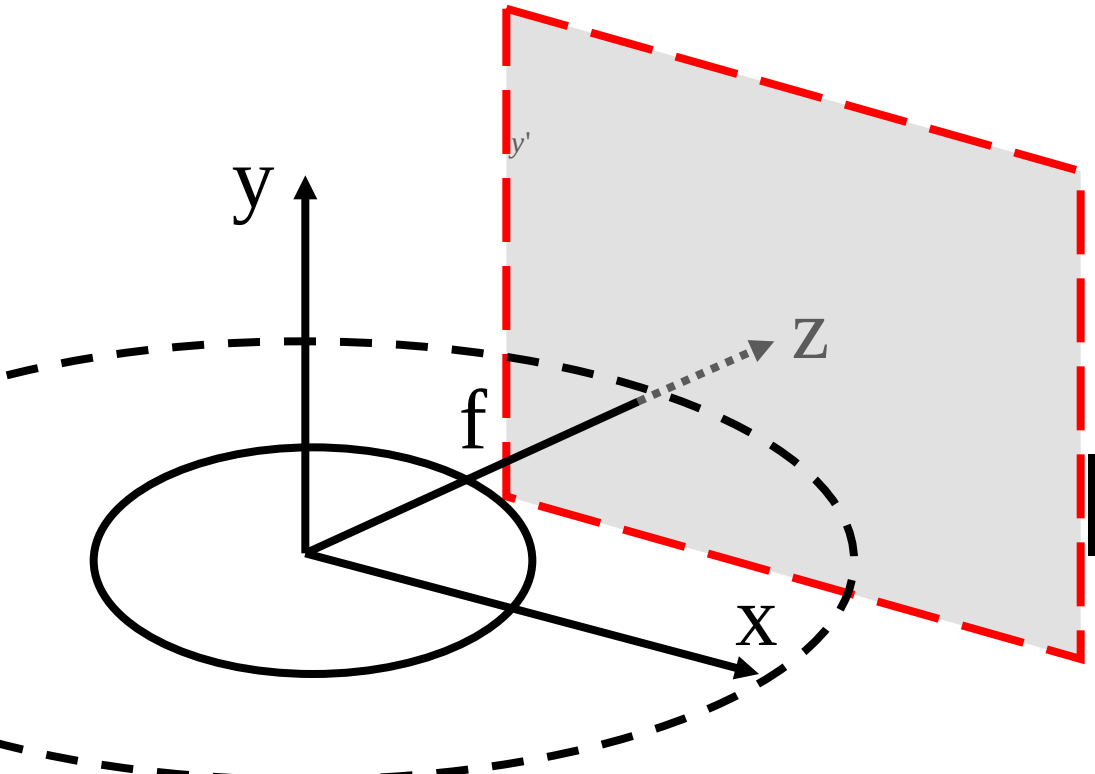
unwrapped cylinder



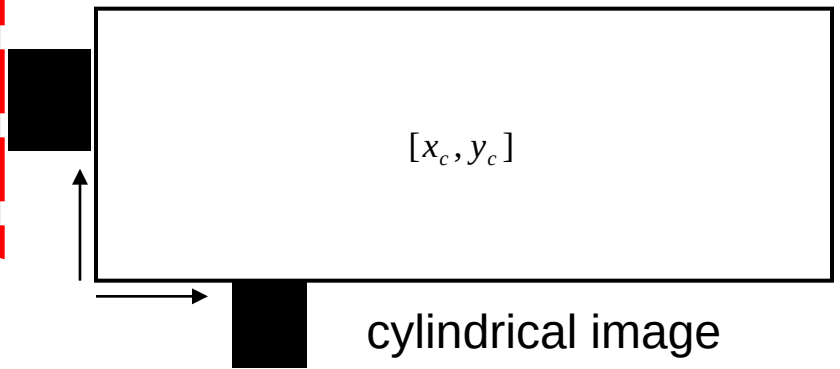
$[x_c, y_c]$

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

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

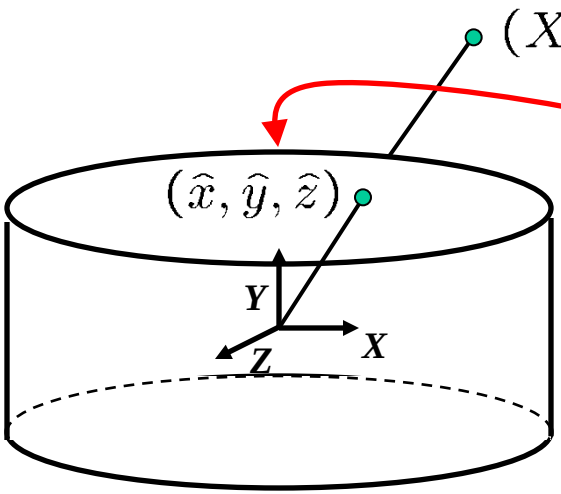


s defines size of the final image, often convenient to set s = f



cylindrical image

Pipeline



unit cylinder

Map 3D point/line (X,Y,Z) or Z=f onto cylinder

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

Convert to cylindrical coordinates

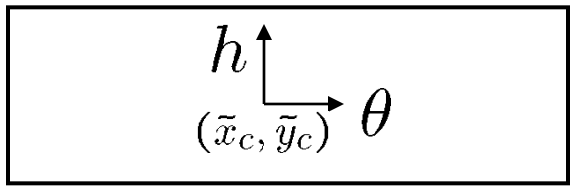
$$(sin\theta, h, cos\theta) = (\hat{x}, \hat{y}, \hat{z})$$

To get an “image”

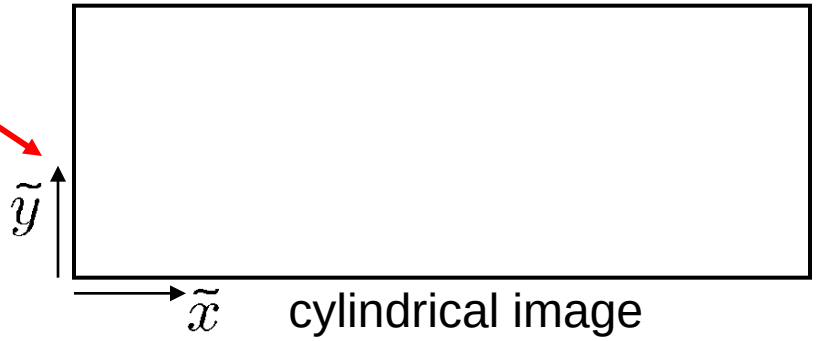
Convert to cylindrical image coordinates

$$(\tilde{x}, \tilde{y}) = (f\theta, fh) + (\tilde{x}_c, \tilde{y}_c)$$

s (or f) defines size of the final image often convenient to set s = camera focal length



unwrapped cylinder



cylindrical image

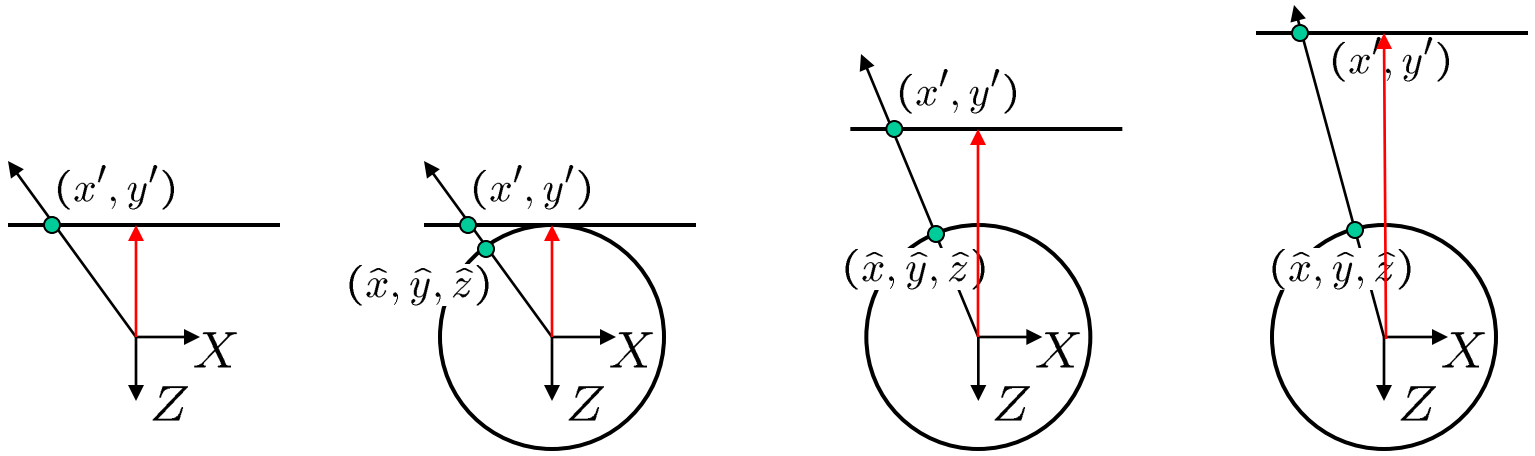
Cylindrical image stitching

- What if you don’t know the camera rotation?
  - Solve for the camera rotations
    - Note that a rotation of the camera is a **translation** of the cylinder

Cylindrical image stitching

- What if you don’t know the camera rotation?
  - Solve for the camera rotations
    - Remember we are not doing any perspective transformations!! Only translations

Cylindrical reprojection

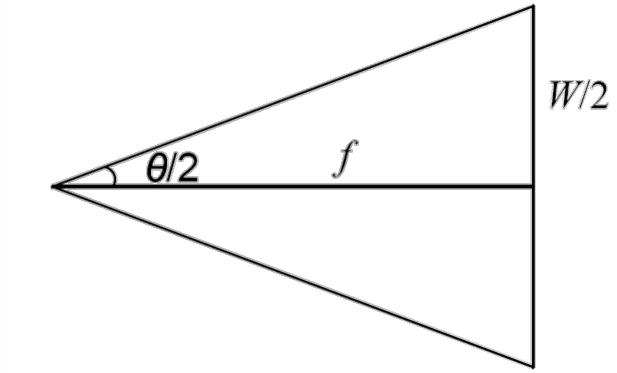


top-down view

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

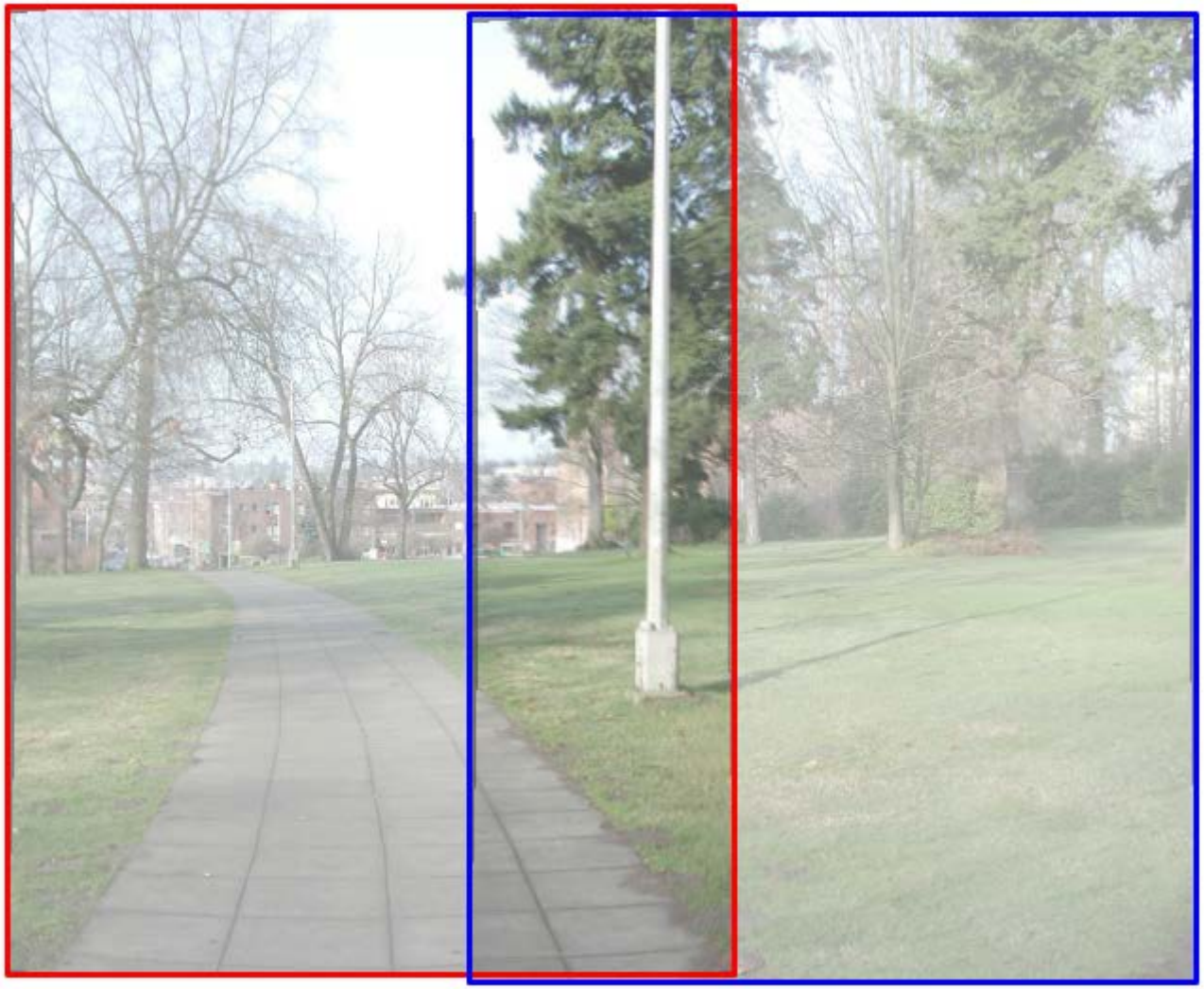
What's your focal length?

- Focal length is (highly!) camera dependent
  - Can get a rough estimate by measuring FOV:

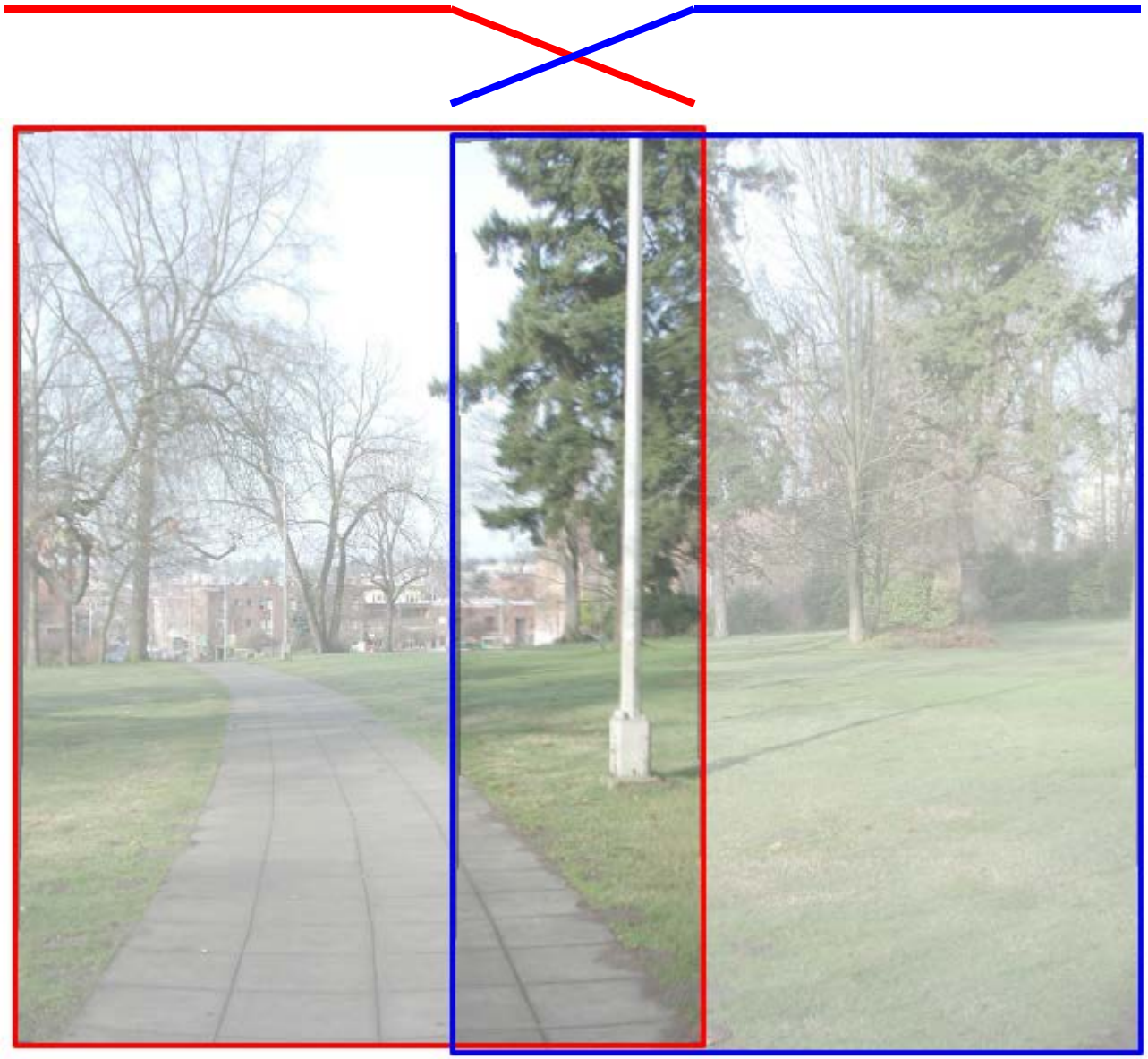


  - Can use the EXIF data tag (might not give the right info)
  - Can use several images together and try to find f that would make them match
  - Can use a known 3D object and its projection to solve for f

Blending the seams

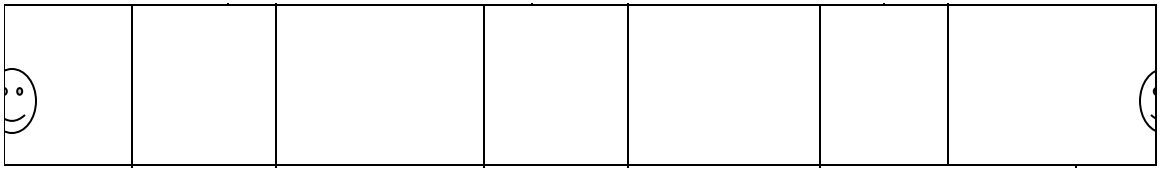


Blending the seams



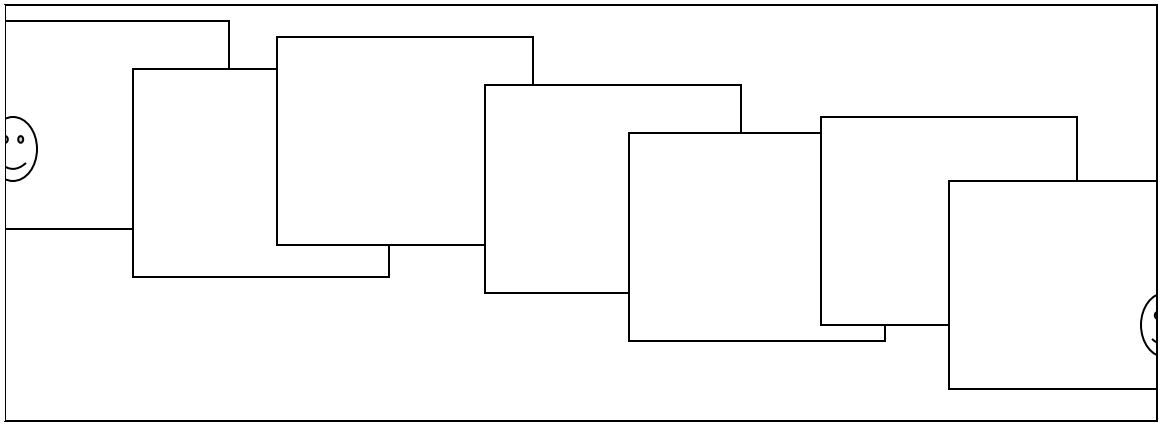
Blending the seams

Assembling the panorama



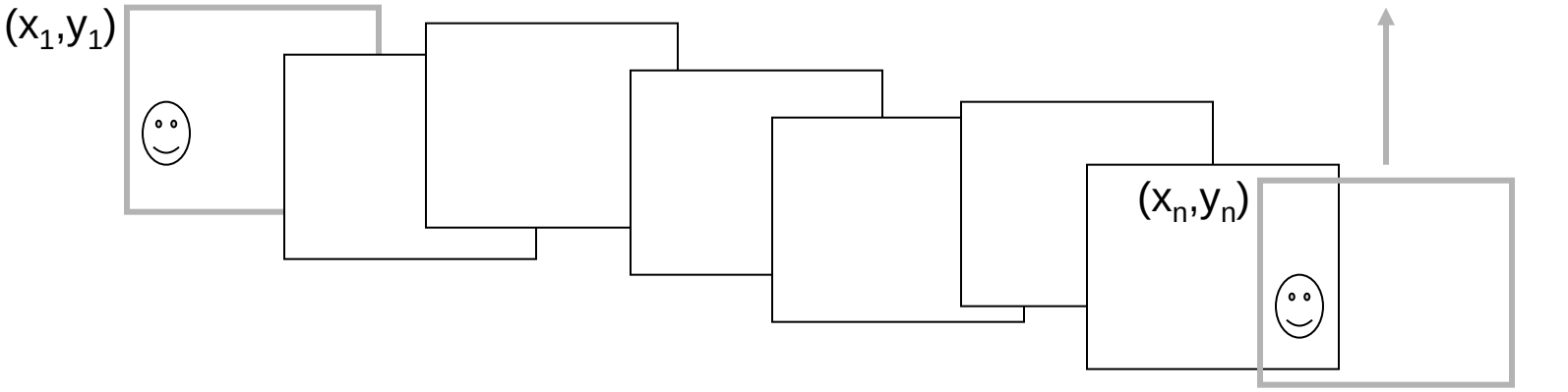
- Stitch pairs together, blend, then **crop**

Problem: Drift



- Error accumulation
  - small errors accumulate over time

Problem: Drift



- copy of first image

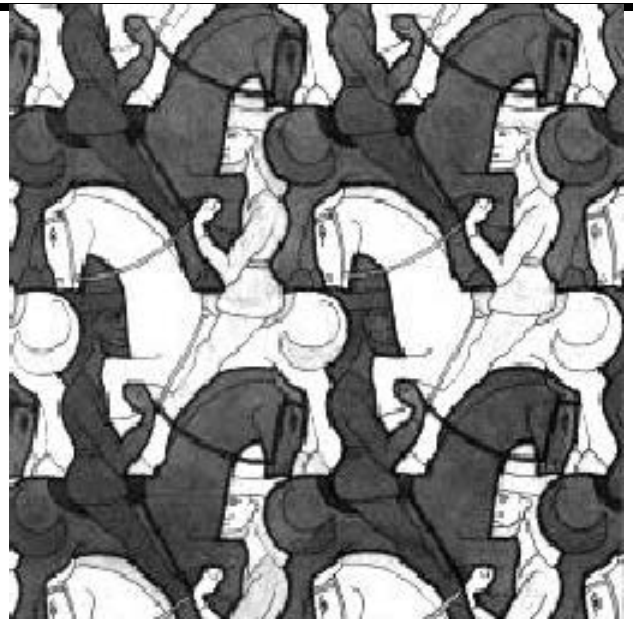
- Error accumulation
  - small (vertical) errors accumulate over time
  - apply correction so that sum = 0

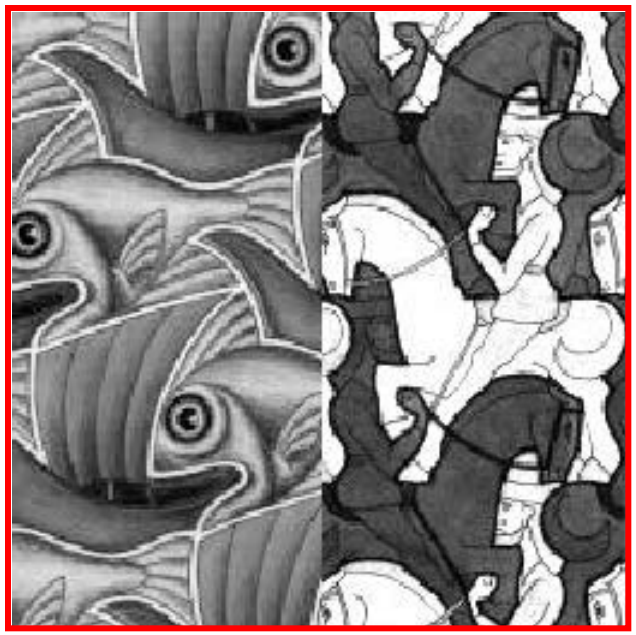
End-to-end alignment and crop

Cylindrical panorama

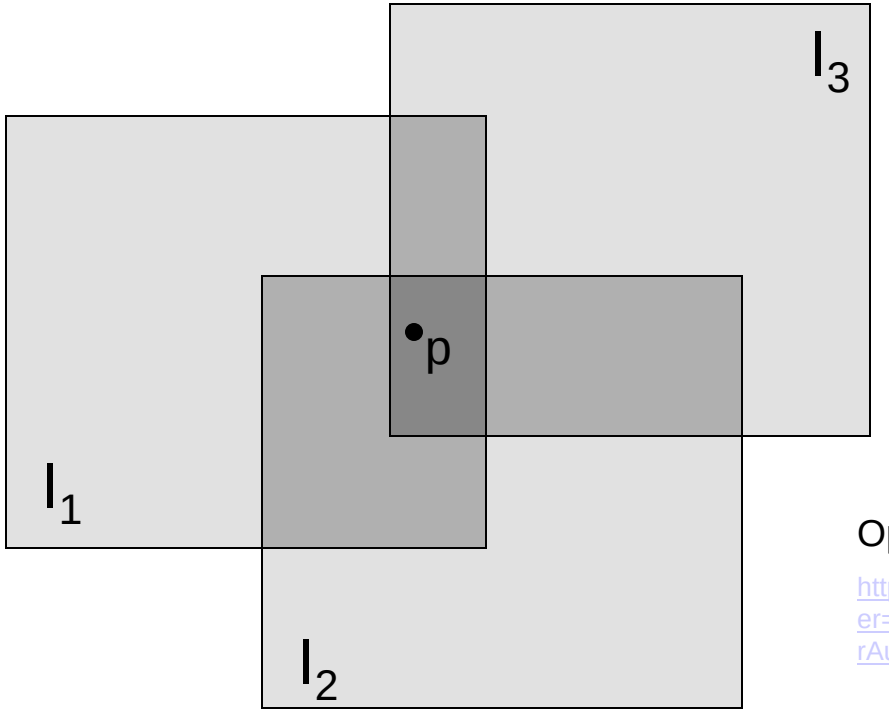
1. Take pictures on a tripod (or handheld)
2. Warp to cylindrical coordinate
3. Compute pairwise alignments
4. Fix up the end-to-end alignment if needed
5. Blending
6. Crop the result

Image 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.

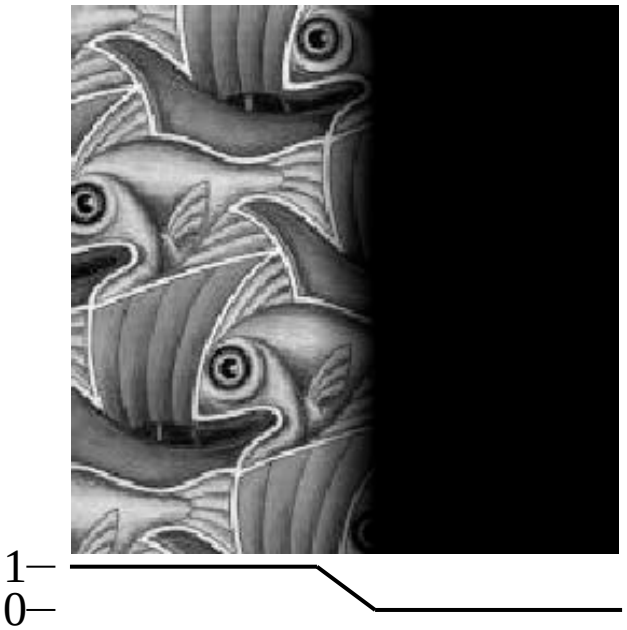
Encoding blend weights

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

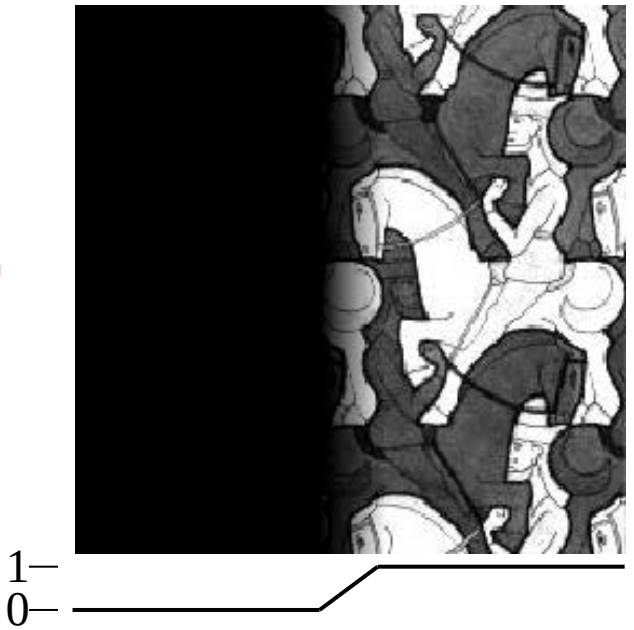
Implement this in two steps:

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

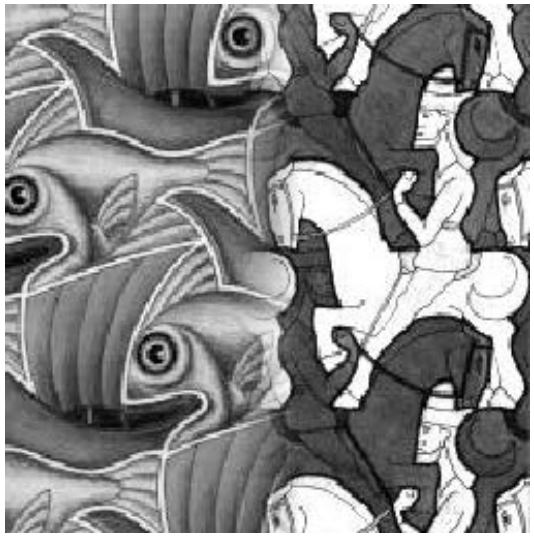
Alpha Blending / Feathering



+

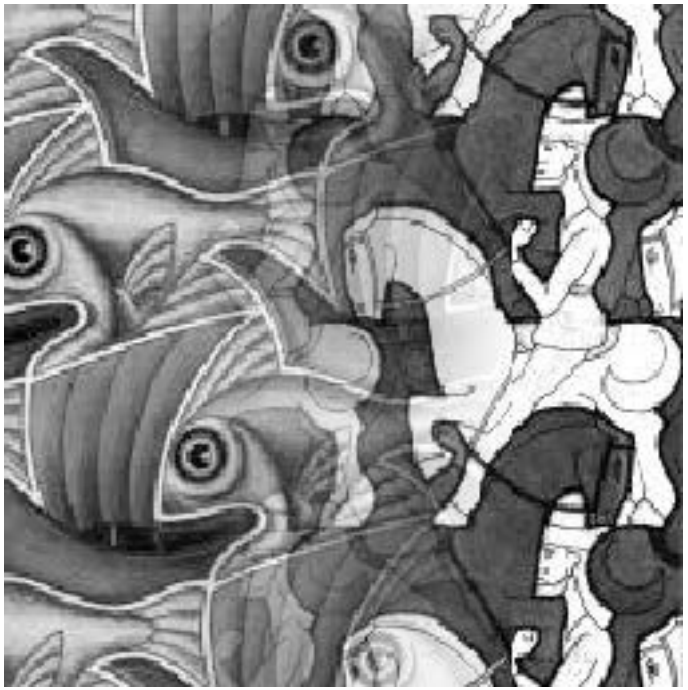


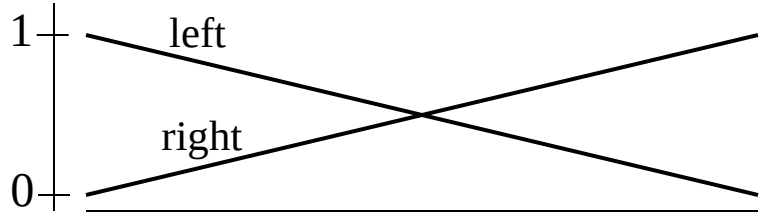
=

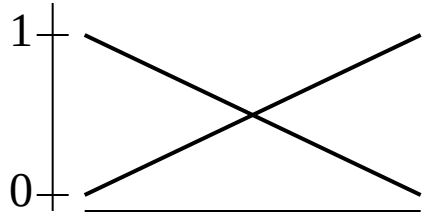


$$I_{blend} = \alpha I_{left} + (1-\alpha) I_{right}$$

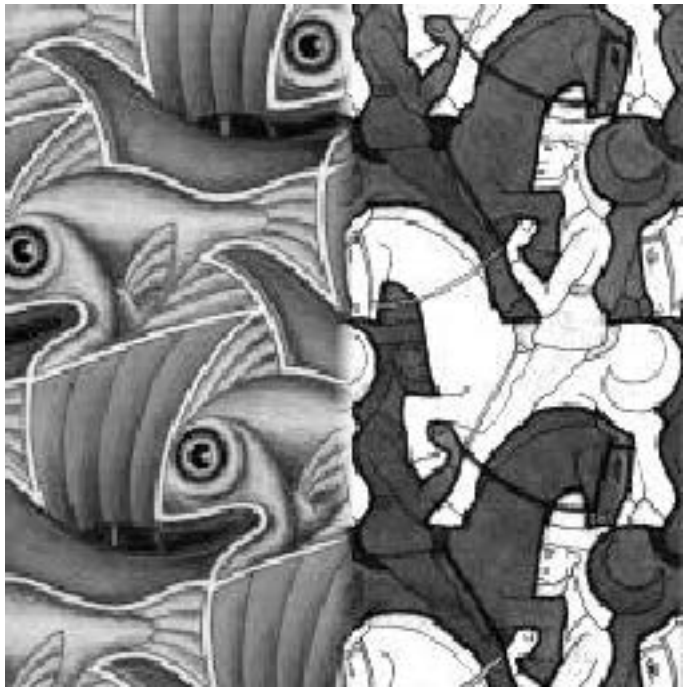
Effect of window size

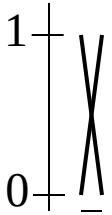


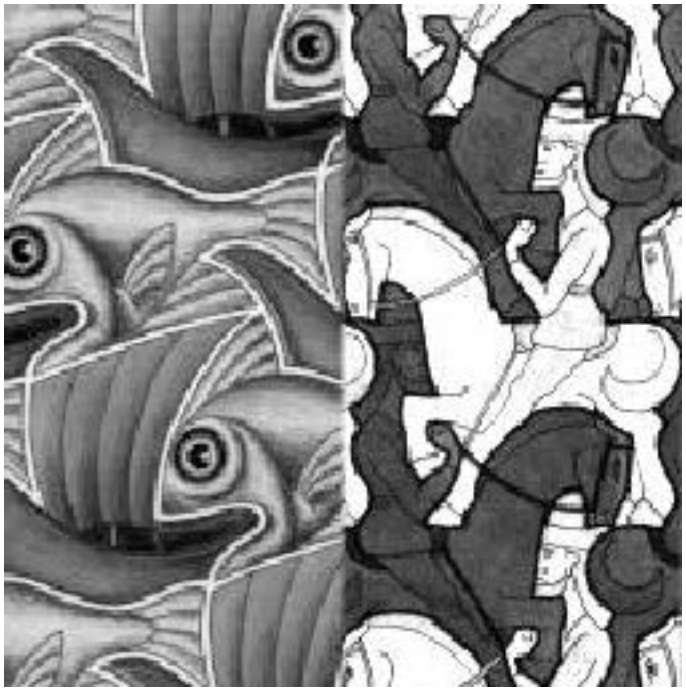




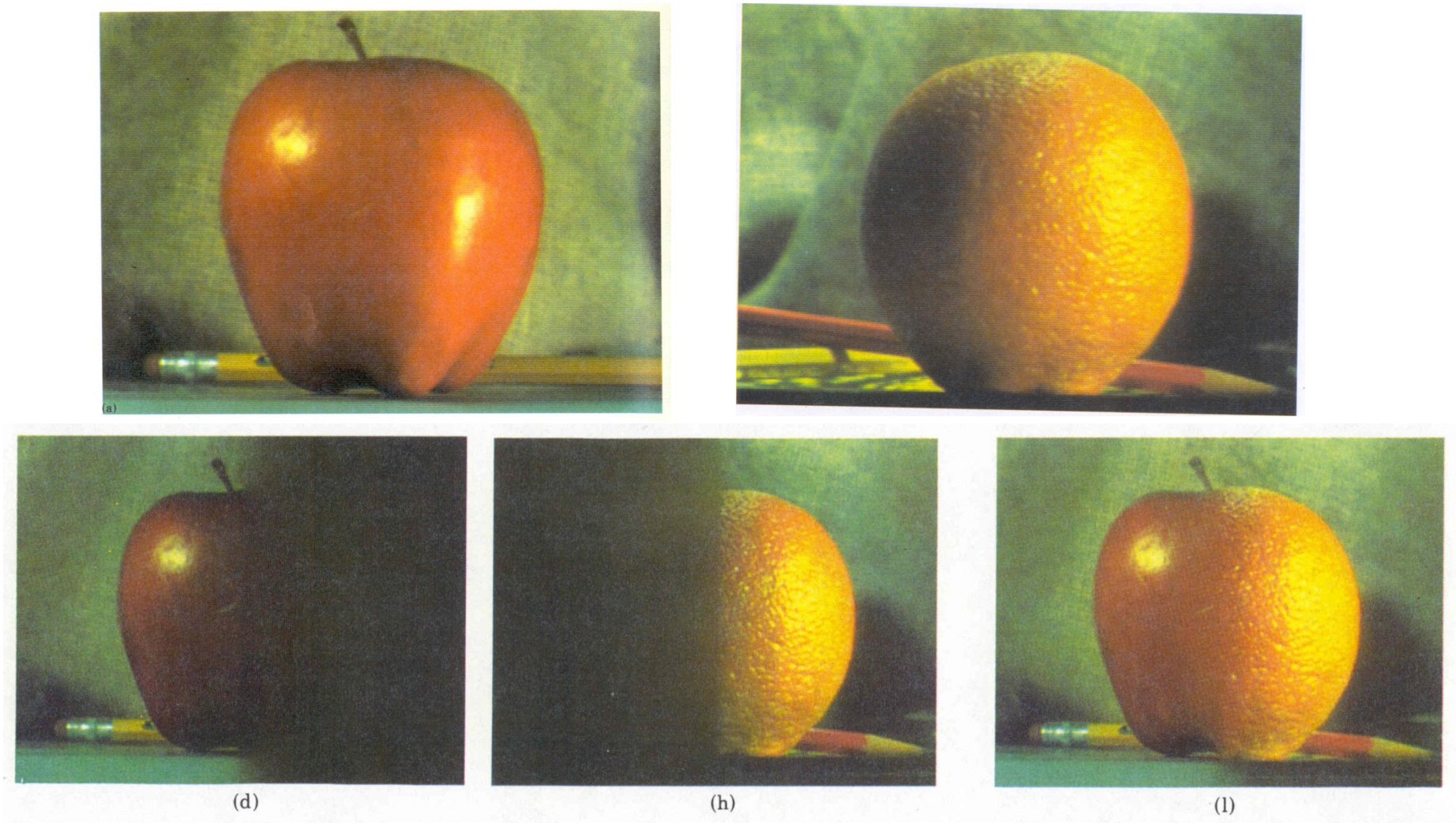
Effect of window size







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.

Don’t blend, CUT when there is motion!

Moving objects become ghosts

- So far we only tried to blend between two images. What about finding an optimal seam?

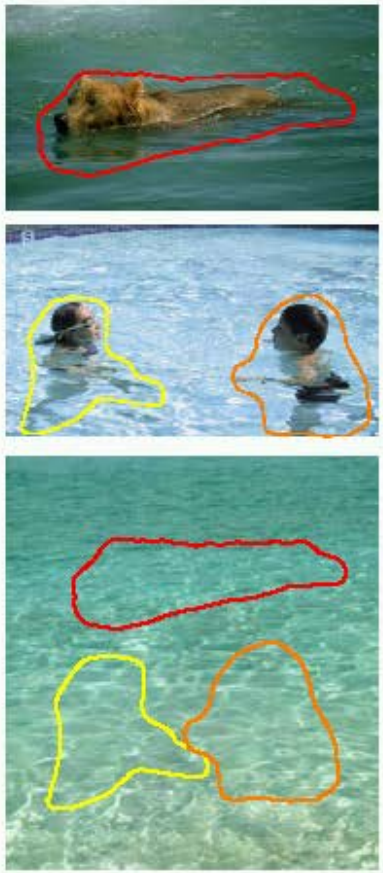
Davis, 1998

- Segment the mosaic
  - Single source image per segment!!!
  - Avoids artifacts along boundries
    - Dijkstra's algorithm

Davis, 1998

- Segment the mosaic
  - Single source image per segment
  - Avoid artifacts along boundries
    - Dijkstra's algorithm

Poisson Image Editing (not panoramas)



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

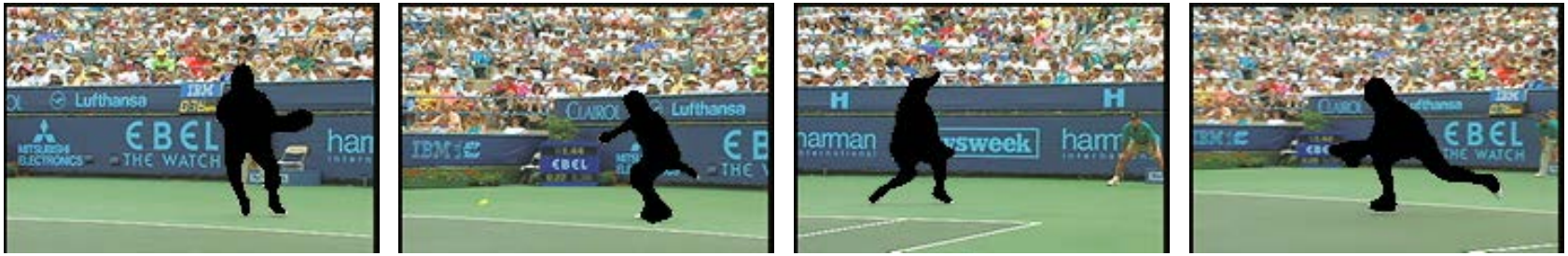
Microsoft Lobby: http://www.acm.org/pubs/citations/proceedings/graph/258734/p251-szeliski

Some panorama examples

What's inside your refrig?

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

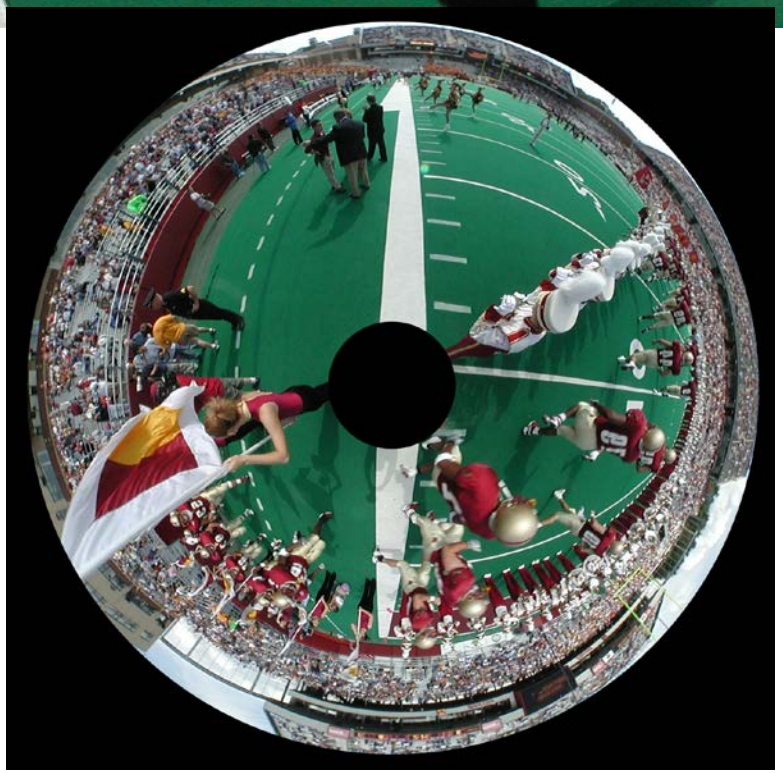
Video compression?

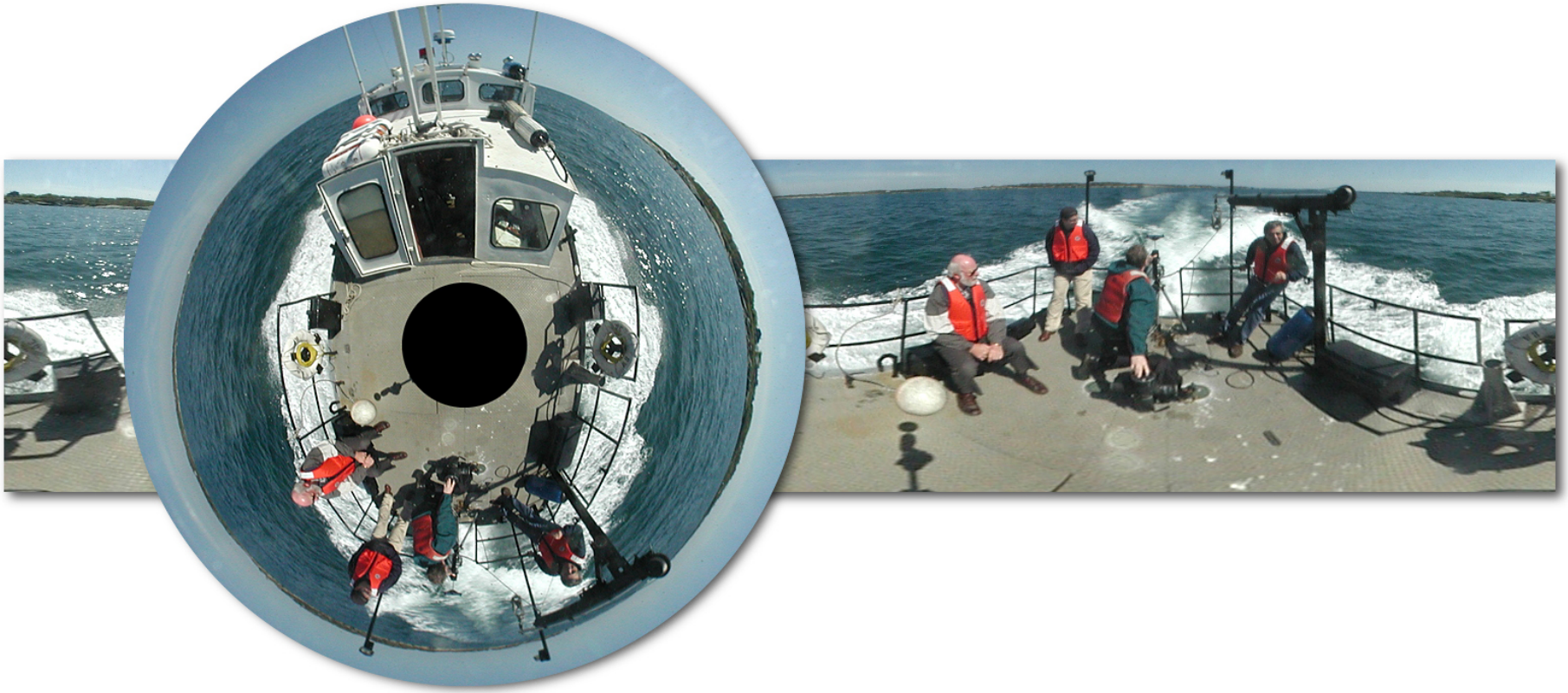


Magic: ghost removal

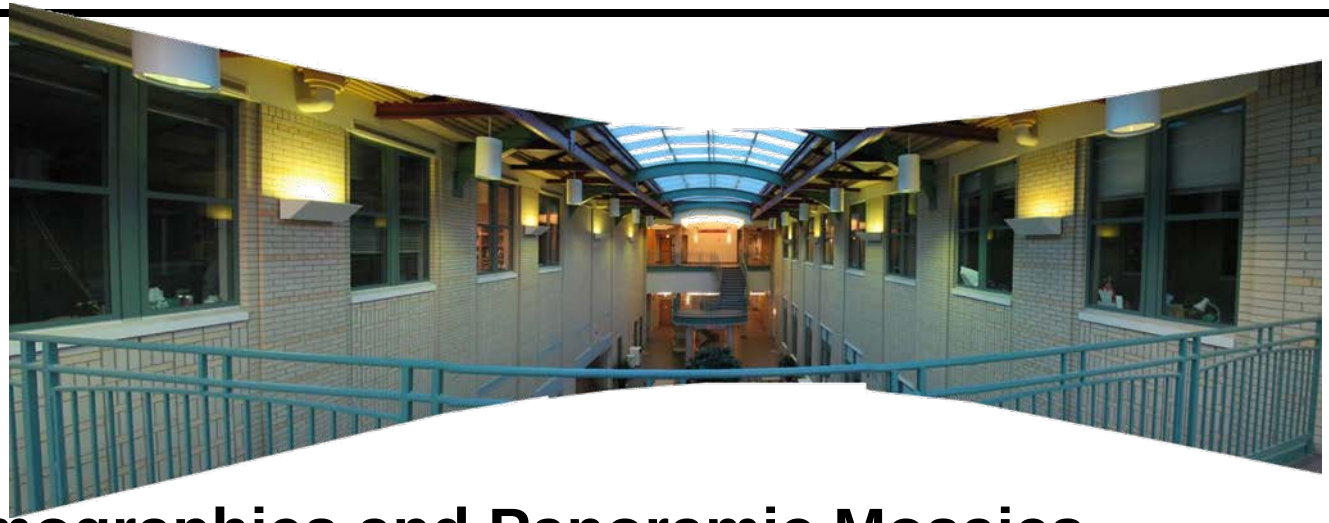
Magic: ghost removal

M. Uyttendaele, A. Eden, and R. Szeliski.
***Eliminating ghosting and exposure artifacts in image mosaics*.**
CVPR 2001





Programming Project #2



- **Homographies and Panoramic Mosaics**
- Capture photographs (and possibly video)
  - can check out cameras and/or tripods
- Compute homographies (define correspondences)
  - will need to figure out how to setup system of eqs.
- (un)warp an image (undo perspective distortion)
- Produce a few panoramic mosaics (with blending)
- Do some of the Bells and Whistles

When does a Video lead to a Good Panorama?

- Good homographies between frames
- Individual images have good quality
- Result has a wide field of view

Competing interests:

- More frames give wider field of view
- More frames give more accumulated error

In defense of 8 point algorithm (Hartley)

Observation: Linear estimation of projective transformation parameters from point correspondences often suffer from poor “conditioning” of the matrices involves. This means the solution is sensitive to noise in the points (even if there are no outliers).

Solution: To get better answers, precondition the matrices by performing a normalization of each point set by:

- translating center of mass to the origin
- scaling so that average distance of points from origin is sqrt(2).