

Homographies and Panoramas



*with a lot of slides stolen from
Steve Seitz and Rick Szeliski*

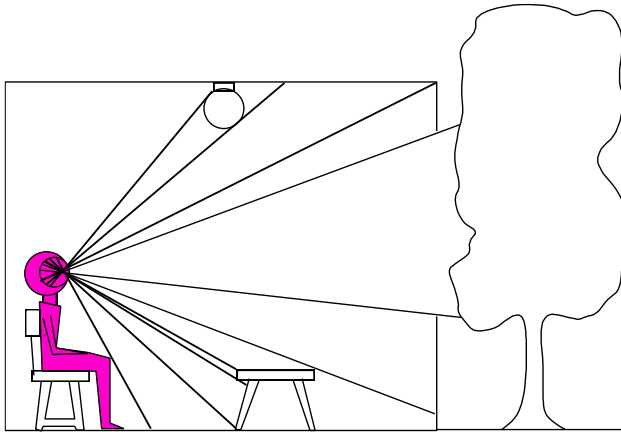
© Andrew Campbell

CS180: Intro to Computer Vision and Comp. Photo
Alexei Efros & Ren Ng, UC Berkeley, Fall 2026

Our visual world as a set of light rays

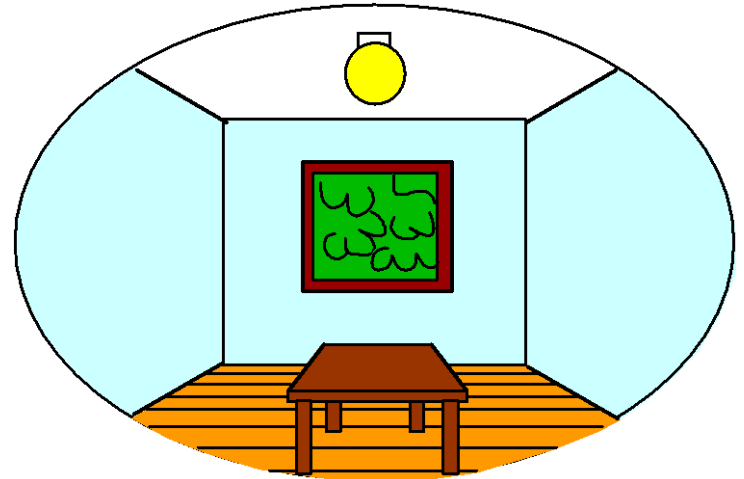
What do we see?

3D world



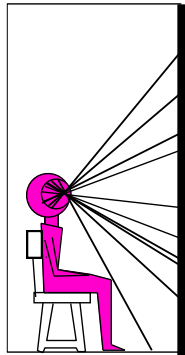
Point of observation

2D image



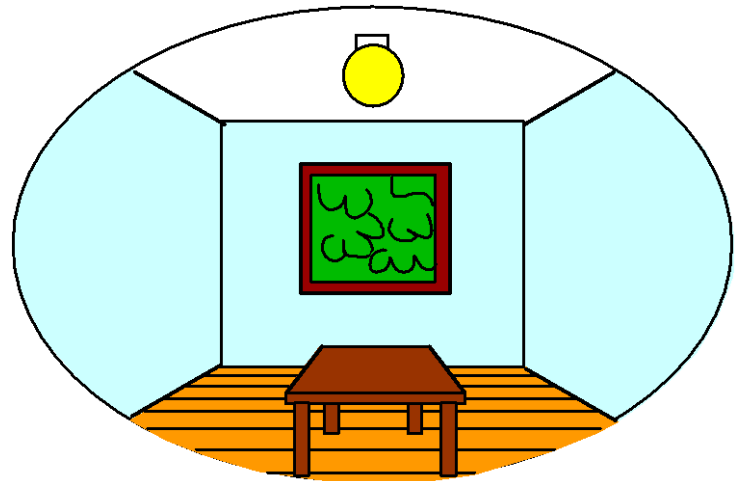
What do we see?

3D world



Painted
backdrop

2D image



On Simulating the Visual Experience

Just feed the eyes the right data

- No one will know the difference!

Philosophy:

- Ancient question: “Does the world really exist?”

Science fiction:

- Many, many, many books on the subject, e.g. *slowglass* from [“Light of Other Days”](#)
- 1990s take: *The Matrix*

Physics:

- *Slowglass* might be possible?

Computer Science:

- Virtual Reality

To simulate we need to know:

What does a person see?

Thought Exp: Record Every Possible Image

- Imagine you want to record every possible image that a person could ever see while exploring this room.
- You have unlimited cameras, storage, and time.
- What would you need to record? How would you organize all those images so that someone could retrieve any view they wanted?
- Think about what variables determine which image you see. Can you make a complete list of variables?

Think (30 sec): think, individually list the variables

Pair (90 sec): compare lists of variables and discuss

The Plenoptic Function

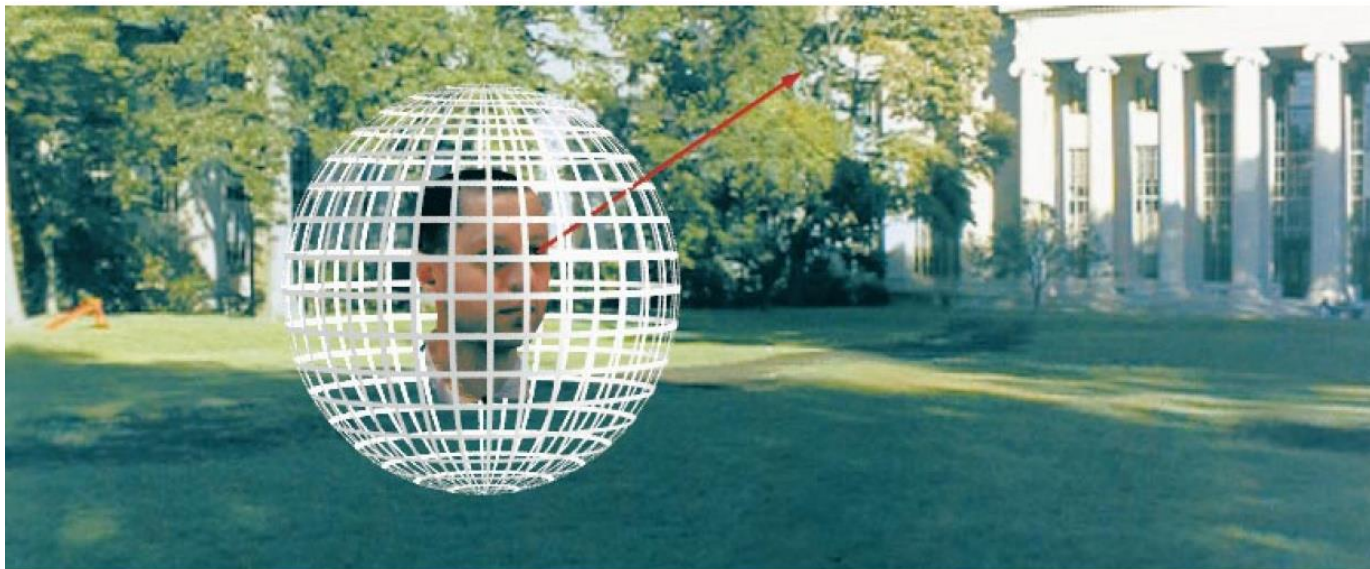


Figure by Leonard McMillan

Q: What is the set of all things that we can ever see?

A: The Plenoptic Function (Adelson & Bergen)

Let's start with a stationary person and try to parameterize everything that this person can see...

Grayscale snapshot



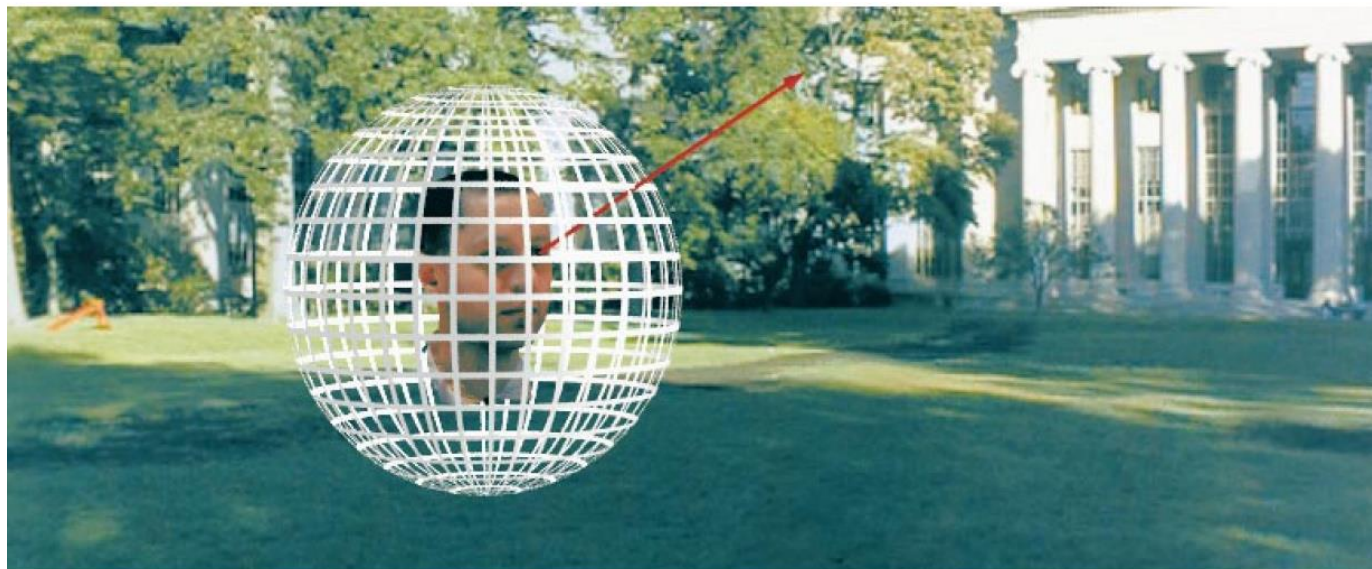
$$P(\theta, \phi)$$

is intensity of light

- Seen from a single view point
- At a single time
- Averaged over the wavelengths of the visible spectrum

(can also do $P(x,y)$, but spherical coordinate are nicer)

Color snapshot

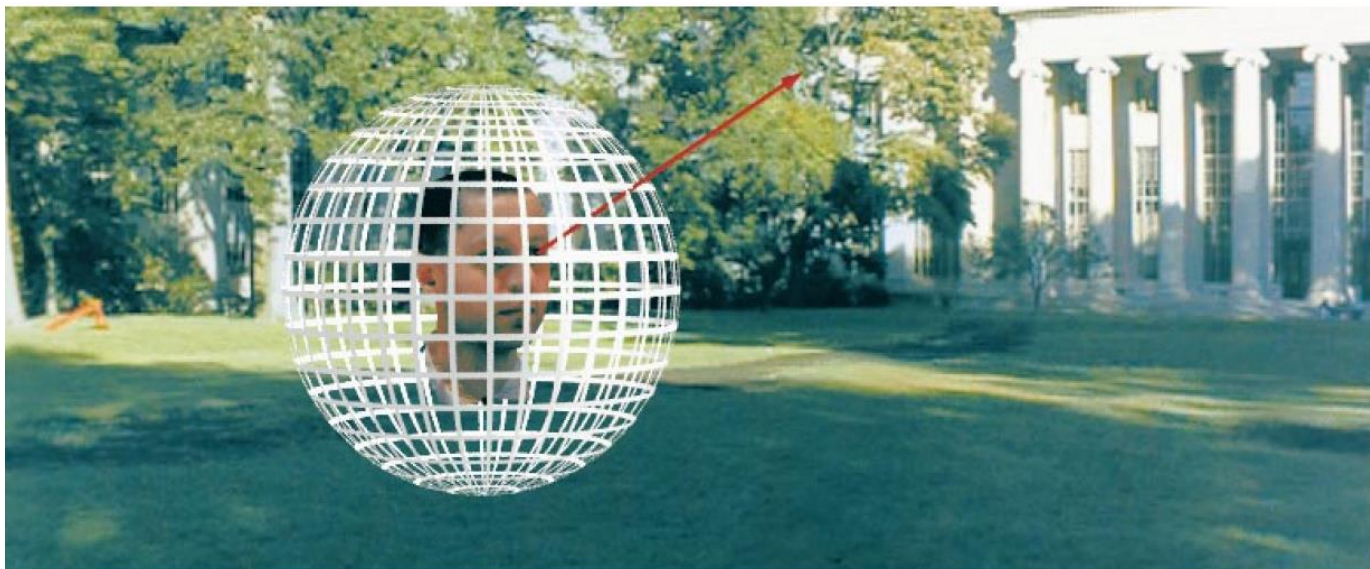


$$P(\theta, \phi, \lambda)$$

is intensity of light

- Seen from a single view point
- At a single time
- As a function of wavelength

A movie

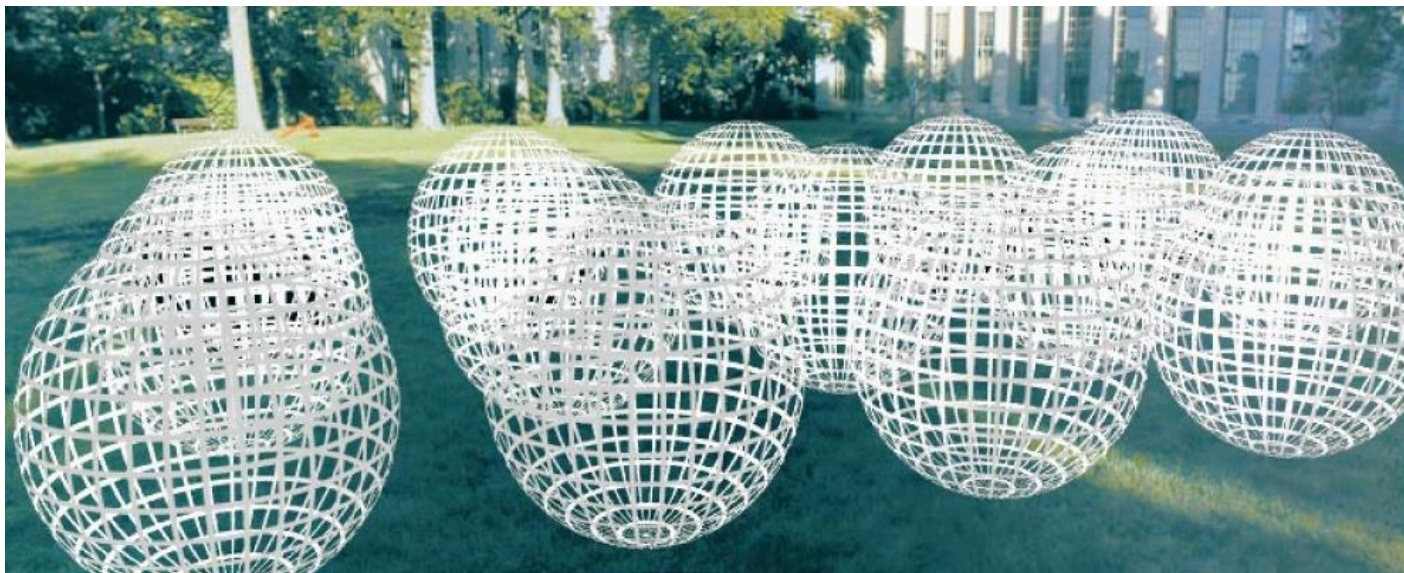


$$P(\theta, \phi, \lambda, t)$$

is intensity of light

- Seen from a single view point
- Over time
- As a function of wavelength

Holographic movie

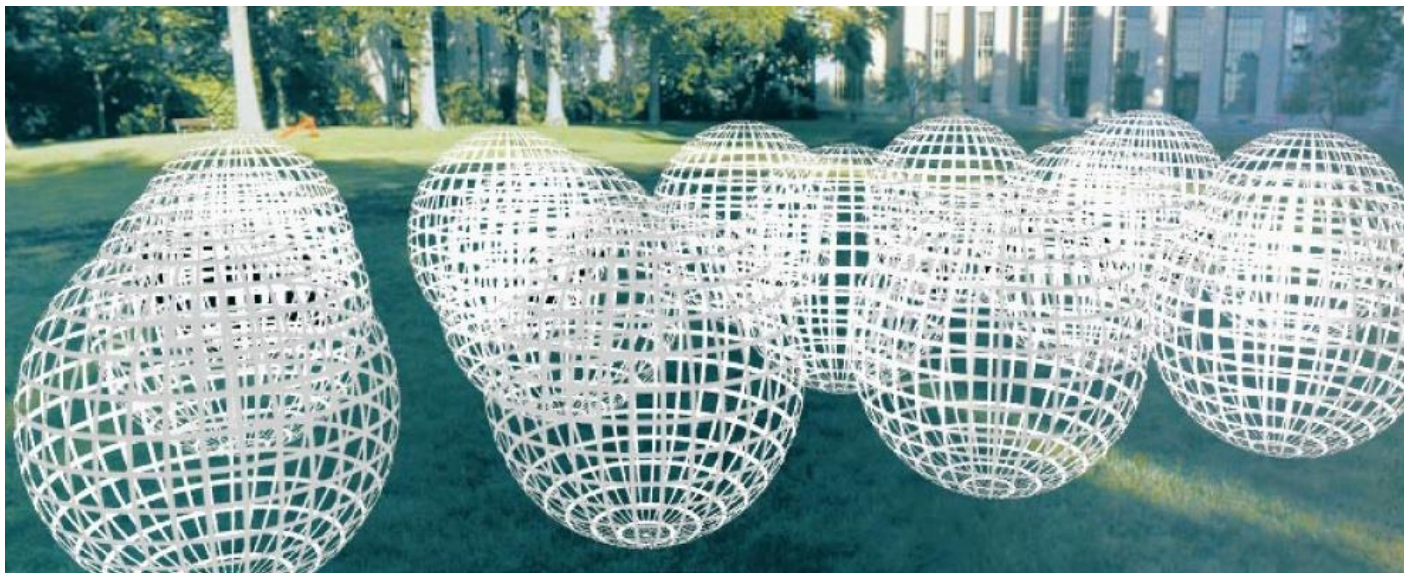


$$P(\theta, \phi, \lambda, t, V_x, V_y, V_z)$$

is intensity of light

- Seen from ANY viewpoint
- Over time
- As a function of wavelength

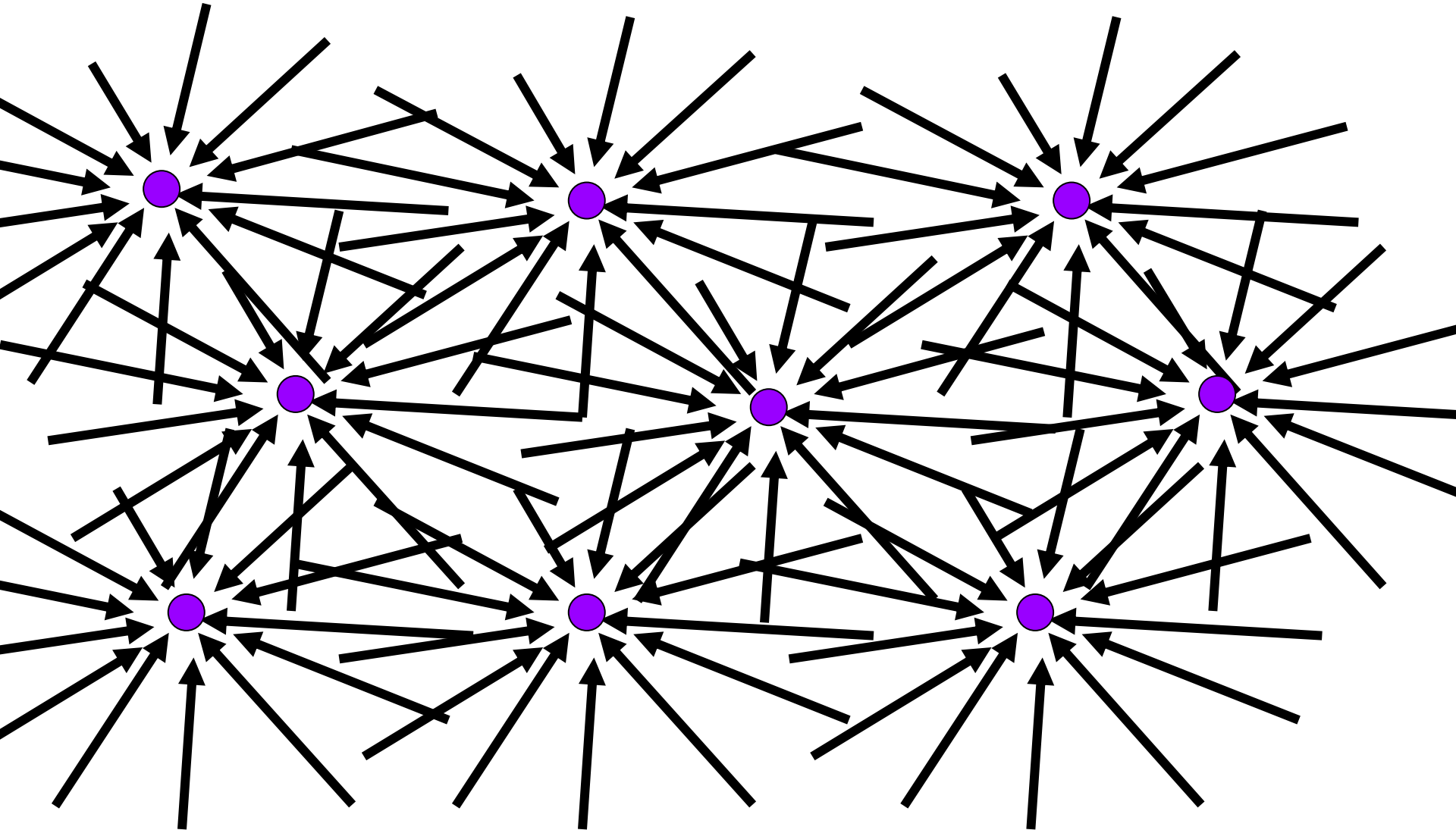
The Plenoptic Function



$$P(\theta, \phi, \lambda, t, V_x, V_y, V_z)$$

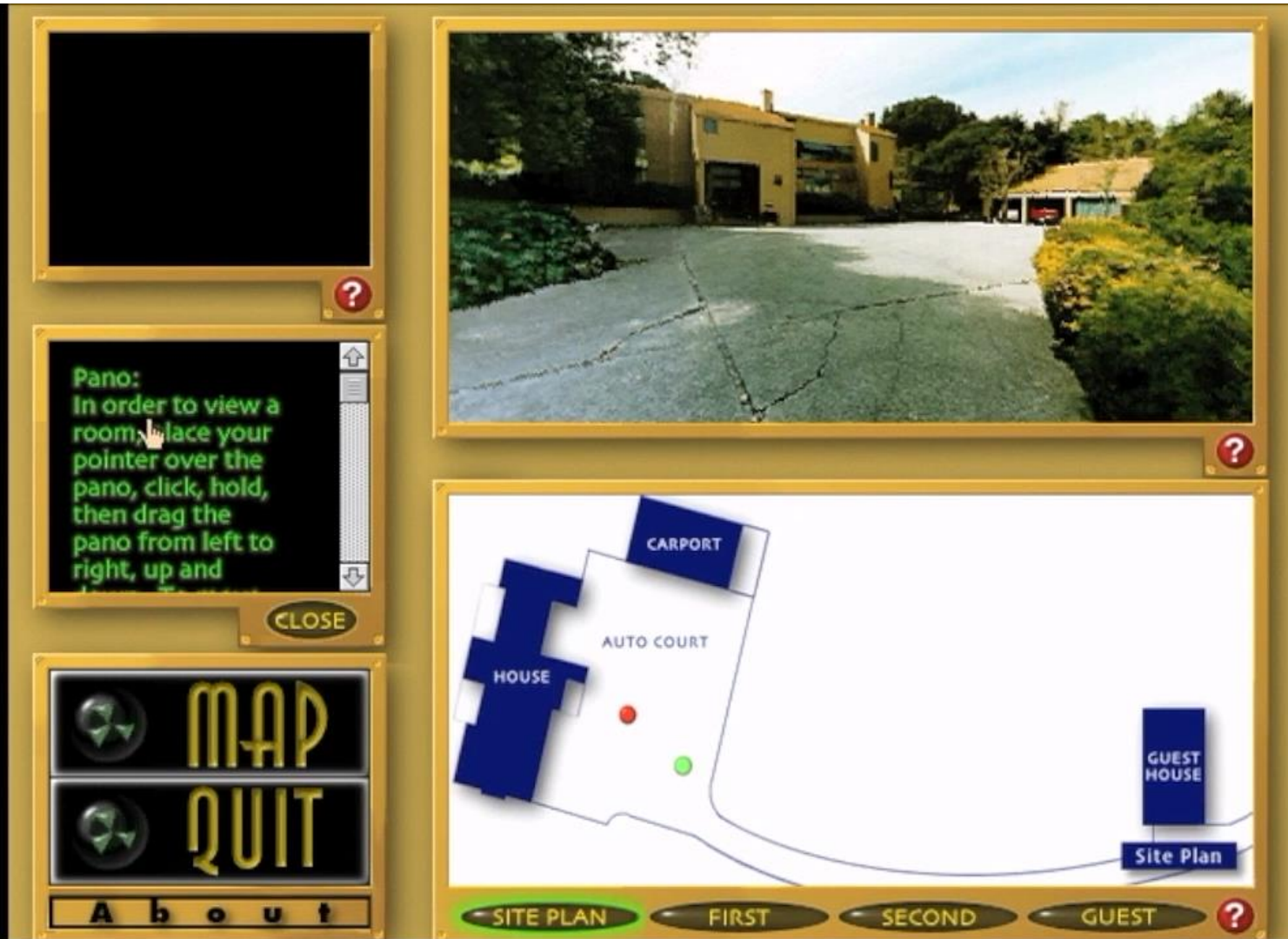
- Can reconstruct every possible view, at every moment, from every position, at every wavelength
- Contains every photograph, every movie, everything that anyone has ever seen! it completely captures our visual reality! Not bad for a function...

Sampling Plenoptic Function (top view)



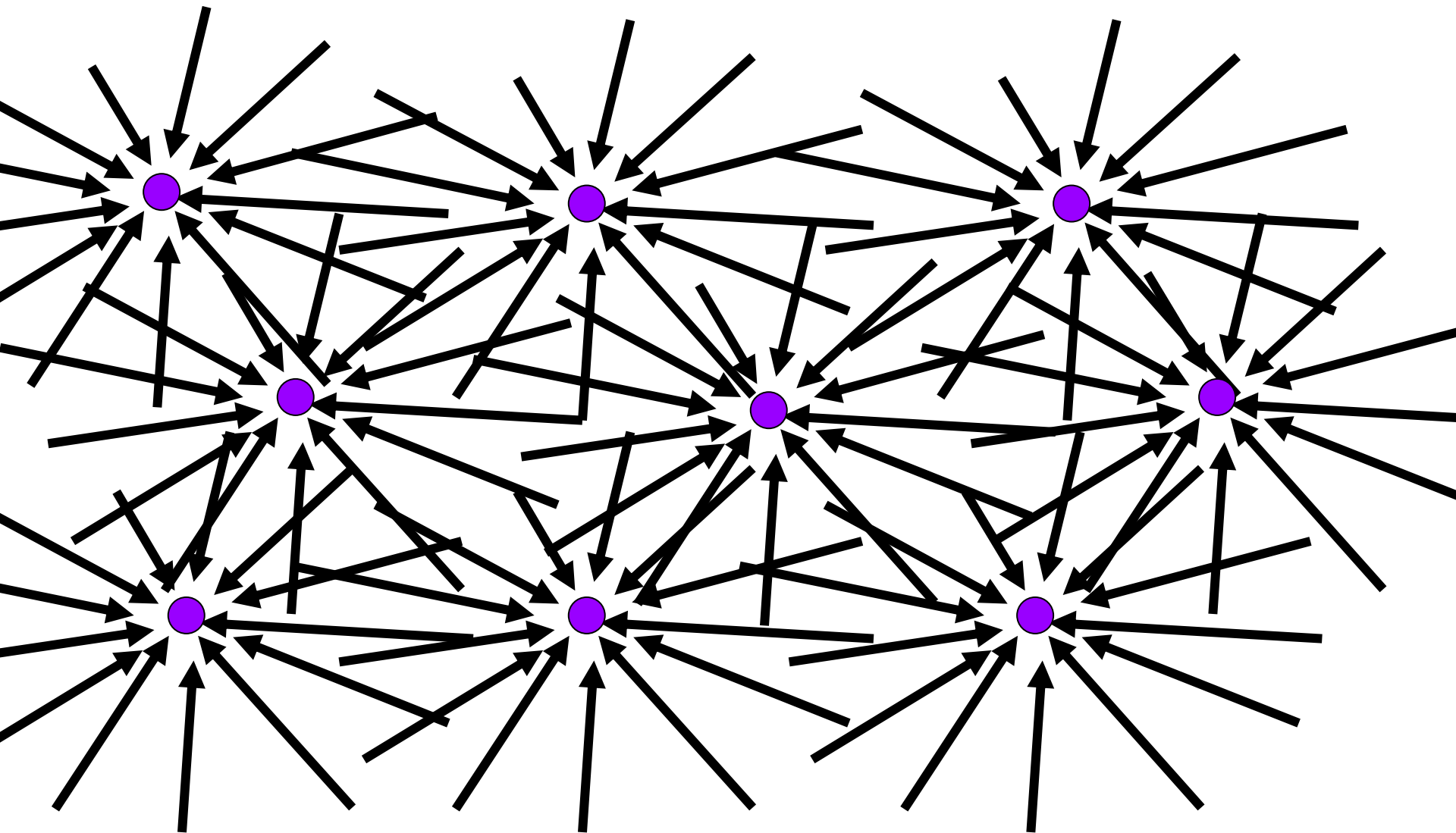
Just lookup -- Quicktime VR

Apple Quicktime VR (1995)

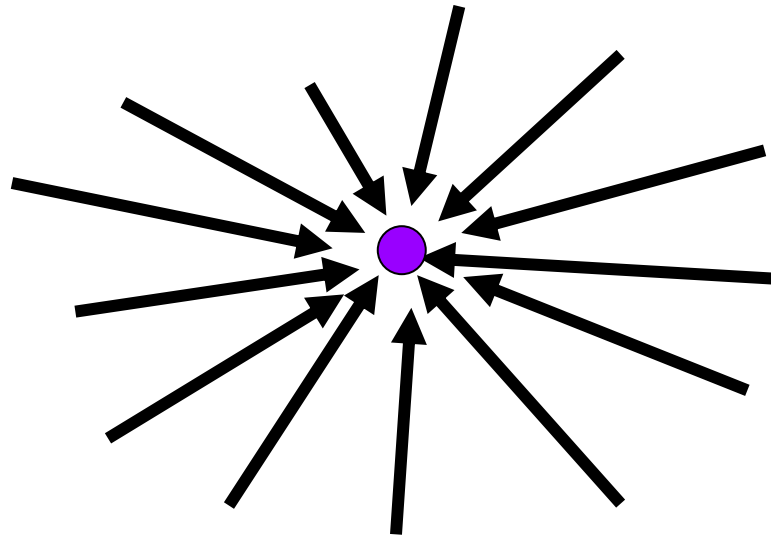


Today: images from a single viewpoint

Plenoptic Function



Images from a single viewpoint



Spherical (“360”) Panorama



All light rays through a point form a panorama

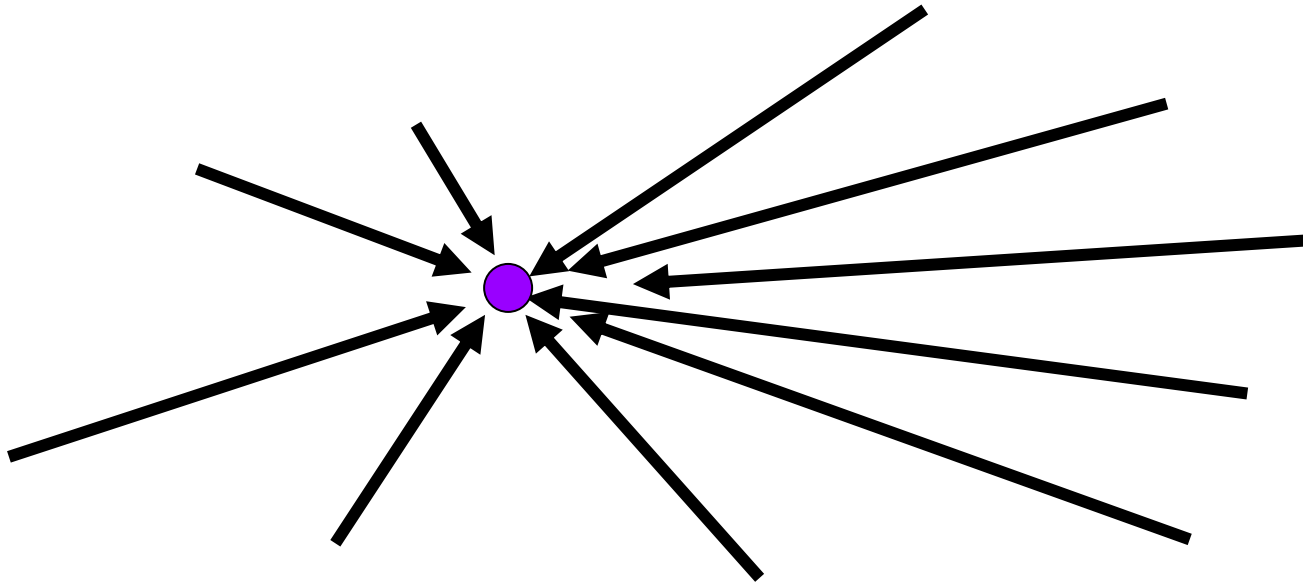
Totally captured in a 2D array -- $P(\theta, \phi)$

Where is the geometry???

Spherical Imaging – e.g. Dual Fisheye Camera

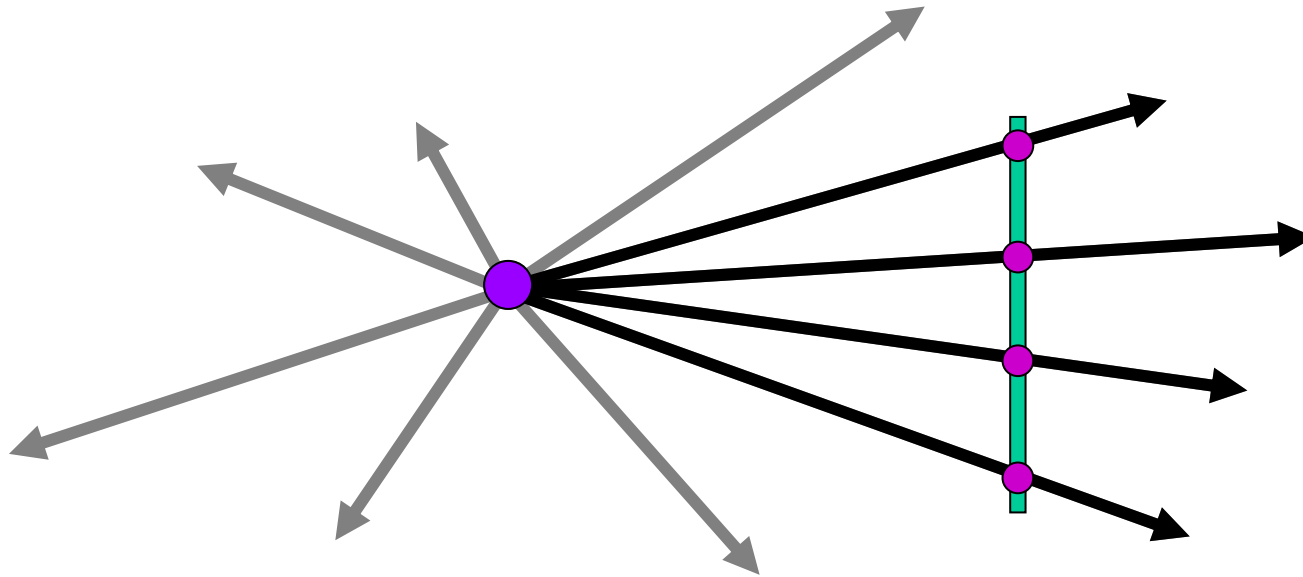


Spherical Panoramas vs Photographs



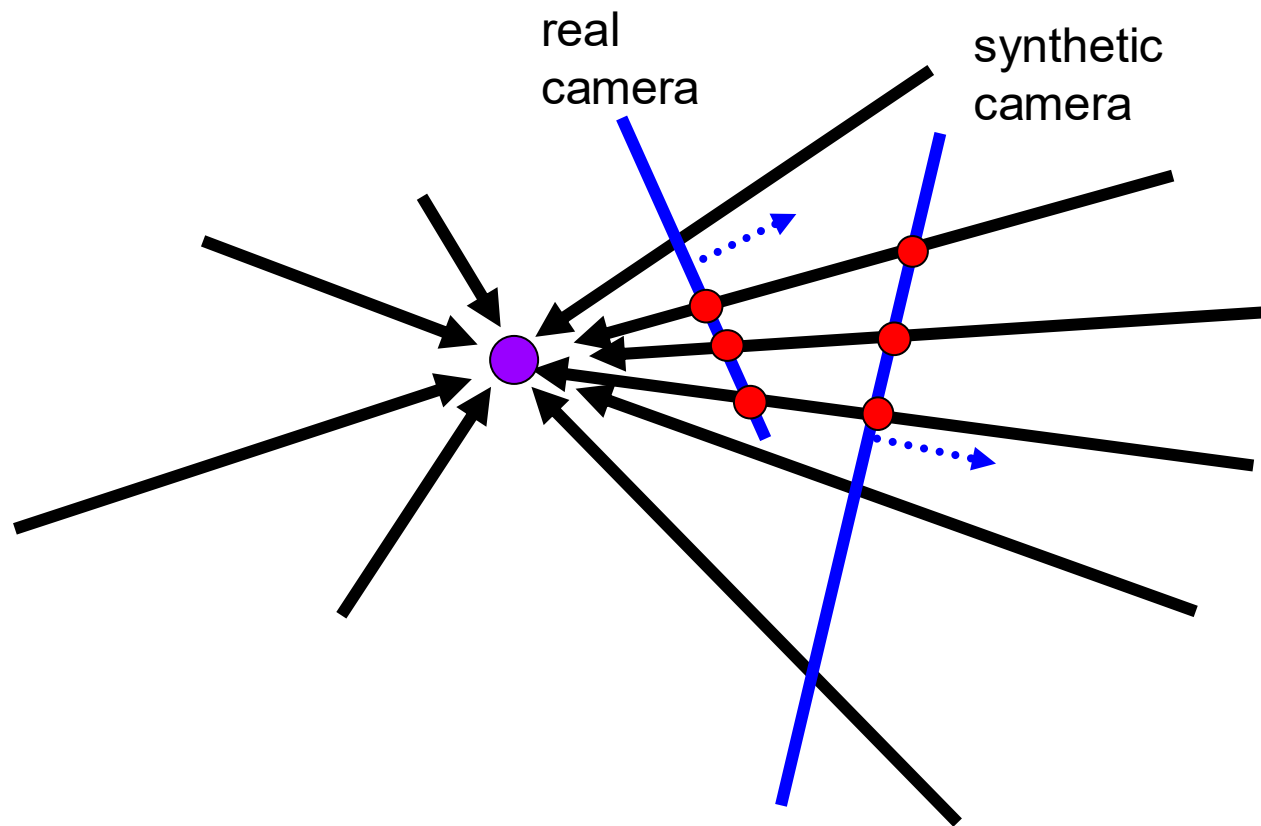
A spherical panorama contains all images from a single viewpoint

Spherical Panoramas vs Photographs



One photograph from this viewpoint

Spherical Panoramas vs Photographs



Can generate any synthetic camera view
as long as it has **the same center of projection!**

Image reprojection

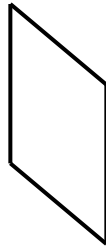
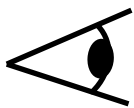


image plane in front

Image reprojection

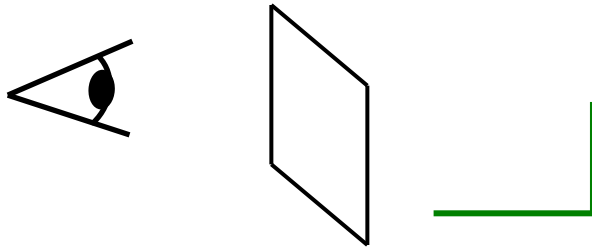
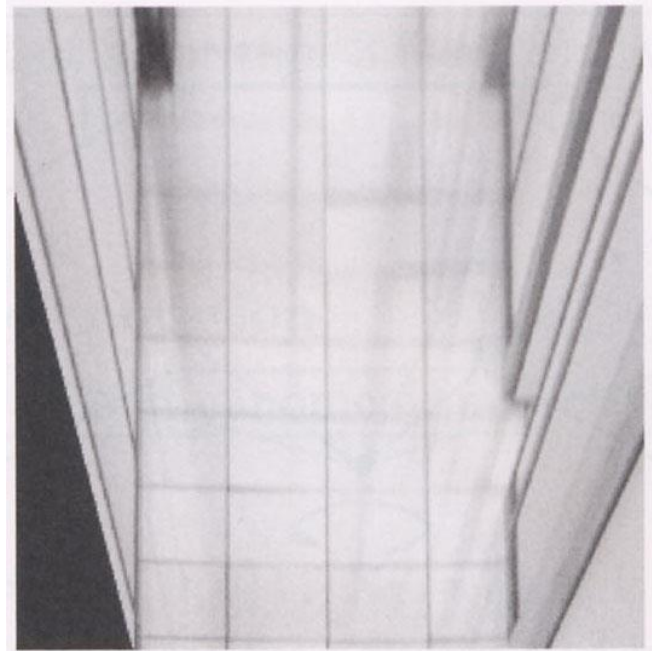


image plane in front

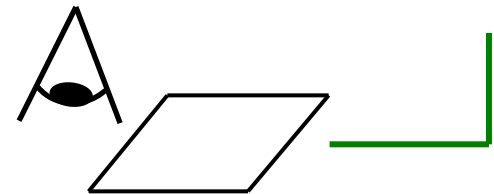


image plane below

Image reprojection

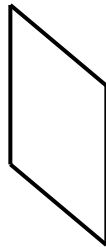
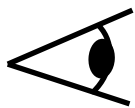
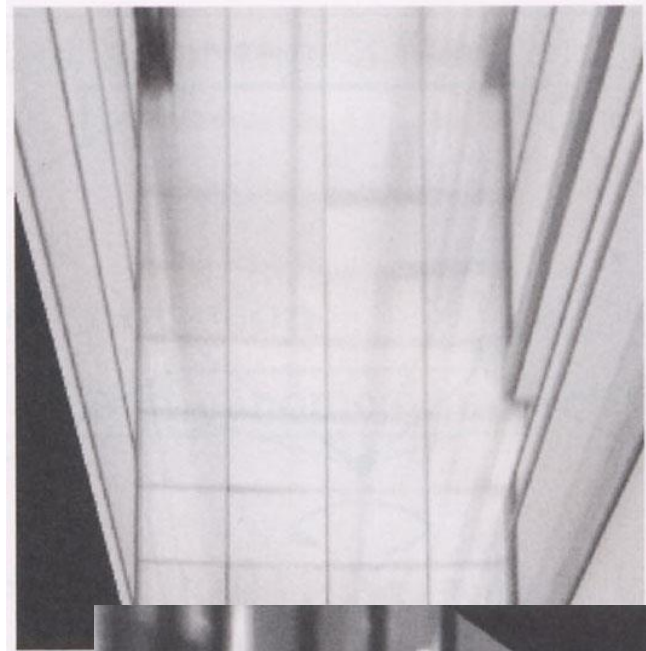


image plane in front

Reverse Engineer: Image Reprojection



image plane in front

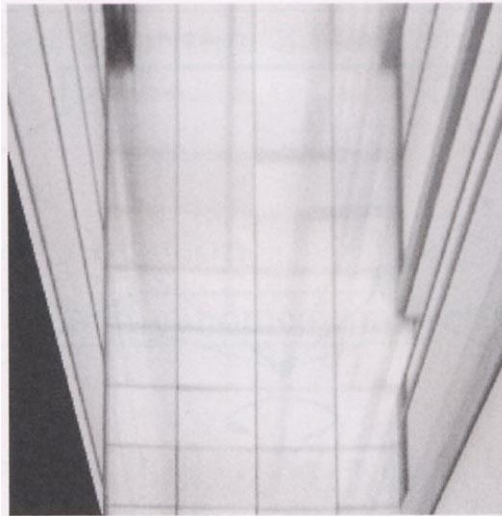


image plane down



image plane to right

- How many input images were needed here?
- How would you compute this reprojection?

Think (30 sec): think individually

Pair (90 min): discuss and merge ideas

Image reprojection

Basic question

- How to relate two images from the same camera center?
 - how to map a pixel from PP1 to PP2

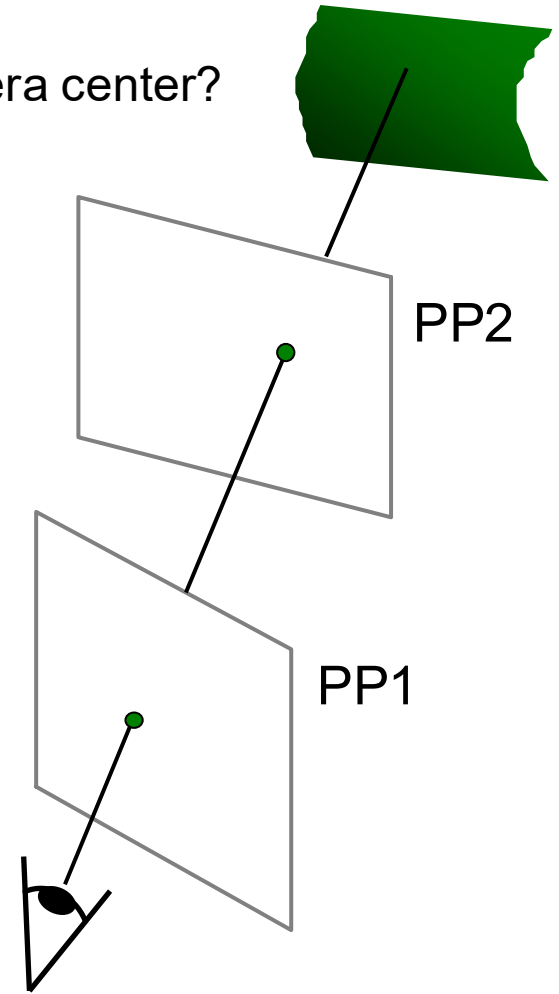
Answer

- Cast a ray through each pixel in PP1
- Draw the pixel where that ray intersects PP2

But don't we need to know the geometry of the two planes in respect to the eye?

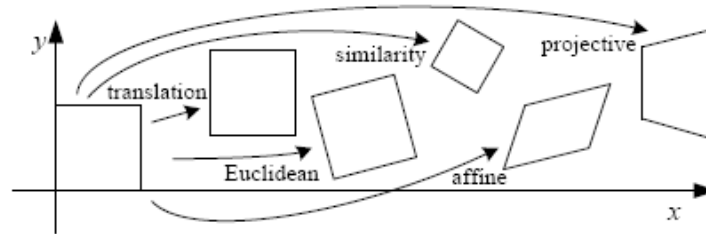
Observation:

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



Back to Image Warping

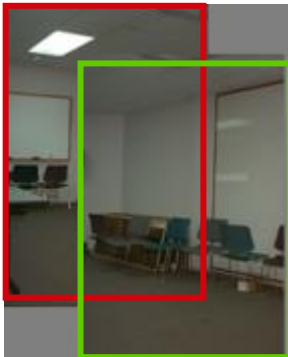
Which t-form is the right one for warping PP1 into PP2?
e.g. translation, Euclidean, affine, projective



Translation

Affine

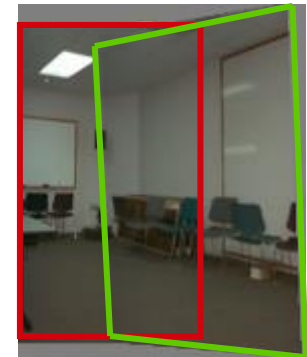
Projective



2 unknowns



6 unknowns



8 unknowns

Homography

A: Projective – mapping between any two PPs with the same center of projection

- rectangle should map to arbitrary quadrilateral
- parallel lines aren't
- but must preserve straight lines
- same as: unproject, rotate, reproject

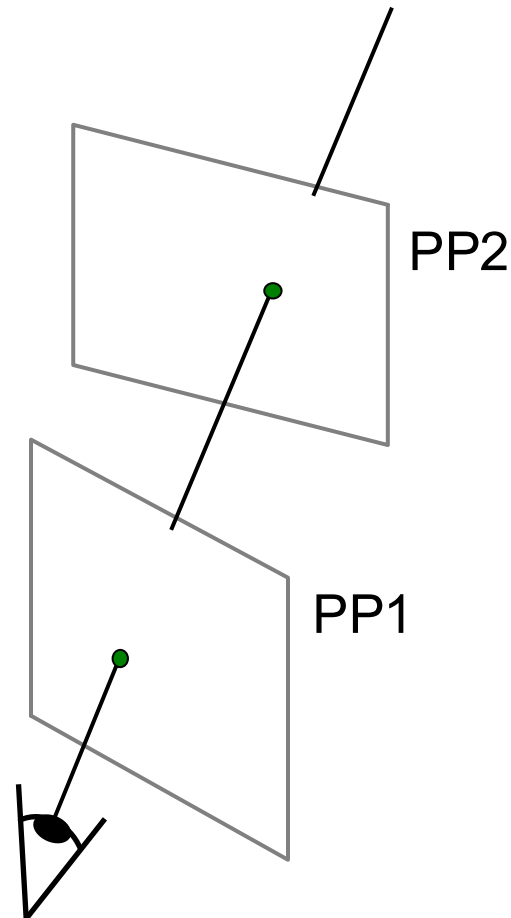
called Homography

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

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

To apply a homography $\mathbf{H}$

- Compute $\mathbf{p}' = \mathbf{H}\mathbf{p}$ (regular matrix multiply)
- Convert $\mathbf{p}'$ from homogeneous to image coordinates



This was warping one image with a homography

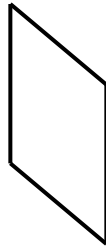
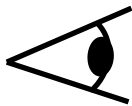


image plane in front

This was warping one image with a homography

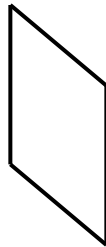
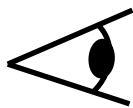
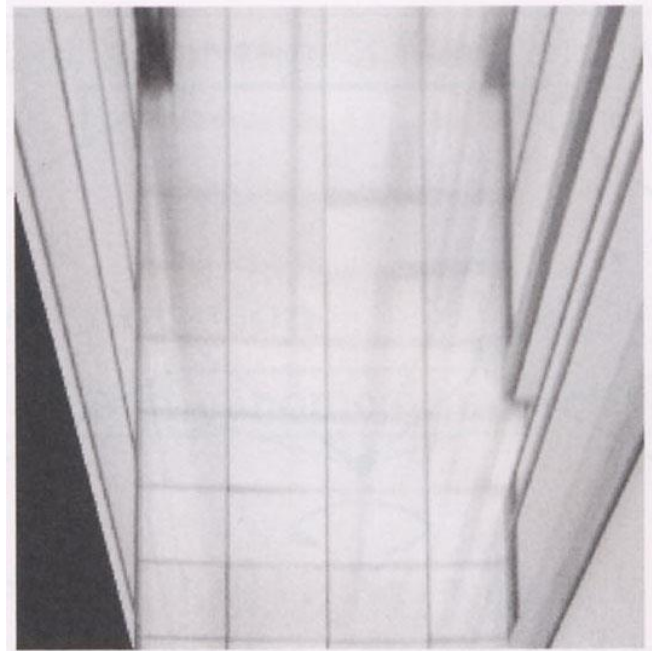


image plane in front

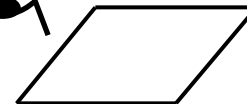


image plane below

black area
where no pixel
maps to

This was warping one image with a homography

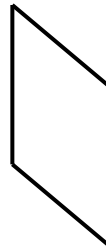
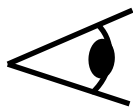
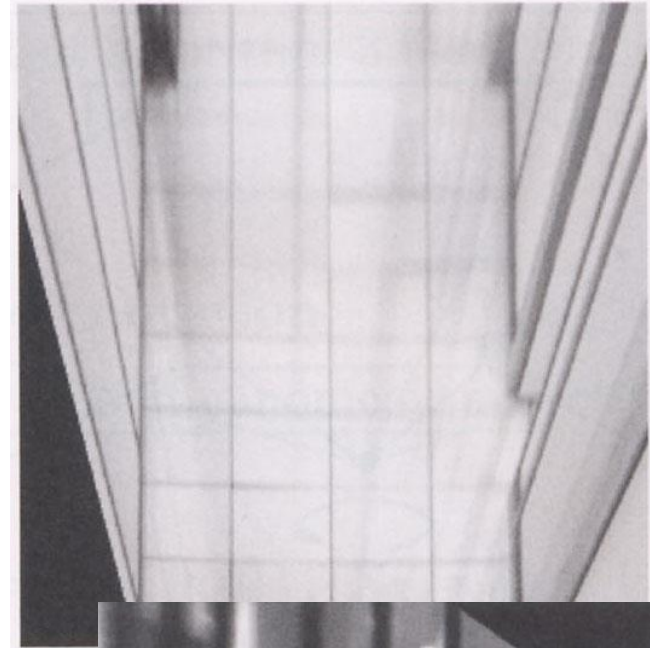


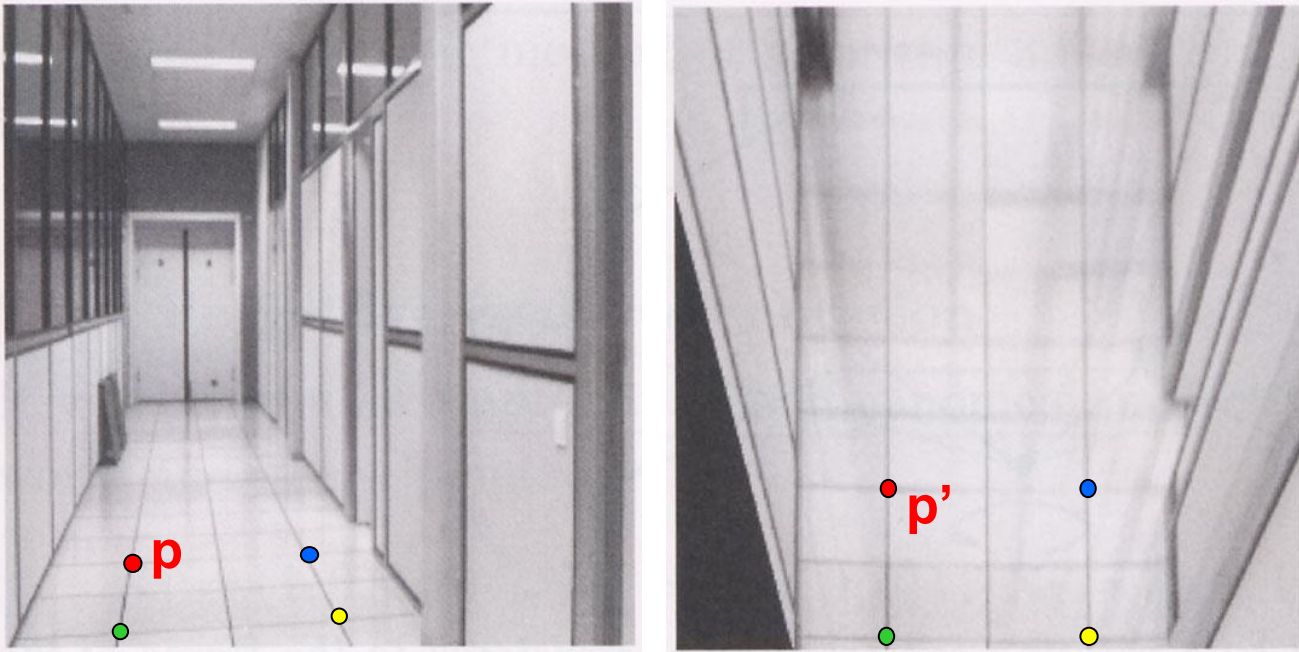
image plane in front



black area
where no pixel
maps to



Implementing image rectification



To unwarp (rectify) an image

- Find the homography $\mathbf{H}$ given a set of $\mathbf{p}$ and $\mathbf{p}'$ pairs
- How many correspondences are needed?
- Tricky to write $\mathbf{H}$ analytically, but we can solve for it!
 - Find such $\mathbf{H}$ that “best” transforms points $\mathbf{p}$ into $\mathbf{p}'$
 - Use least-squares!

Least Squares Example

Say we have a set of data points (p_1, p_1') , (p_2, p_2') , (p_3, p_3') , etc. (e.g. person's height vs. weight)

We want a nice compact formula (a line) to predict p' from p :

$$px_1 + x_2 = p'$$

We want to find x_1 and x_2

How many (p, p') pairs do we need?

$$\begin{aligned} p_1 x_1 + x_2 &= p_1' \\ p_2 x_1 + x_2 &= p_2' \end{aligned} \quad \begin{bmatrix} p_1 & 1 \\ p_2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} p_1' \\ p_2' \end{bmatrix} \quad Ax = b$$

Least Squares Example

Say we have a set of data points (p_1, p_1') , (p_2, p_2') , (p_3, p_3') , etc. (e.g. person's height vs. weight)

We want a nice compact formula (a line) to predict p' from p :

$$px_1 + x_2 = p'$$

We want to find x_1 and x_2

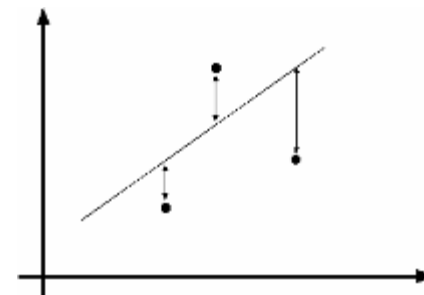
How many (p, p') pairs do we need?

$$\begin{aligned} p_1 x_1 + x_2 &= p_1' \\ p_2 x_1 + x_2 &= p_2' \end{aligned} \quad \begin{bmatrix} p_1 & 1 \\ p_2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} p_1' \\ p_2' \end{bmatrix} \quad Ax = b$$

What if the data is noisy?

$$\begin{bmatrix} p_1 & 1 \\ p_2 & 1 \\ p_3 & 1 \\ \dots & \dots \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} p_1' \\ p_2' \\ p_3' \\ \dots \end{bmatrix}$$

$$\min \|Ax - b\|^2$$



overconstrained

Least-Squares

- Solve:

$$A \mathbf{x} = \mathbf{b}$$

$$(N,d)(d,1) = (N,1)$$

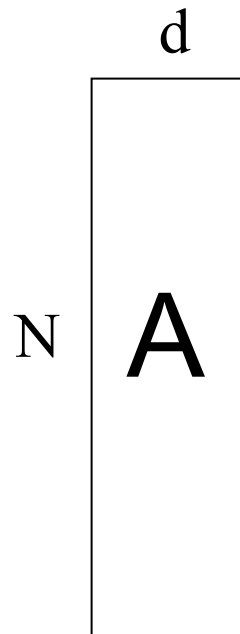
- Normal equations

$$A^T A \mathbf{x} = A^T \mathbf{b}$$

$$(d,N)(N,d)(d,1) = (d,N)(N,1)$$

- Solution:

$$\mathbf{x} = (A^T A)^{-1} A^T \mathbf{b}$$



$\text{rank}(A) \leq \min(d, N)$
assume $\text{rank}(A) = d$
implies $\text{rank}(A^T A) = d$
 $A^T A$ is invertible

Solving for homographies

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

Can set scale factor $i=1$. So, there are 8 unknowns.

Set up a system of linear equations:

$$\mathbf{A}\mathbf{h} = \mathbf{b}$$

where vector of unknowns $\mathbf{h} = [a, b, c, d, e, f, g, h]^T$

Need at least 8 eqs, but the more the better...

Solve for $\mathbf{h}$. If overconstrained, solve using least-squares:

$$\min \|\mathbf{A}\mathbf{h} - \mathbf{b}\|^2$$

Can be done in Matlab using “\” command

- see “help lmdivide”

Image rectification

Original image



St.Petersburg
photo by A. Tikhonov

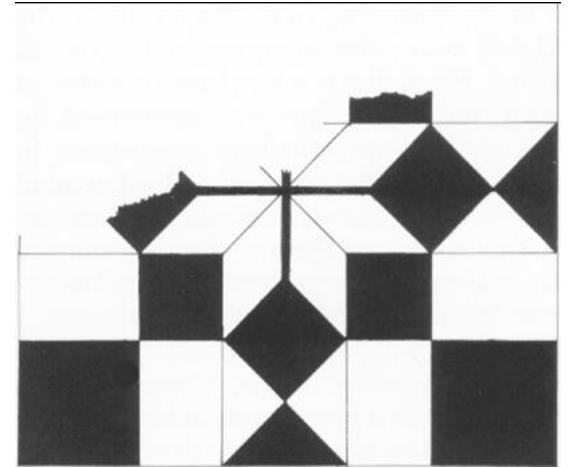
Virtual camera rotation



Inverse rectification



Image Rectification for Art Analysis

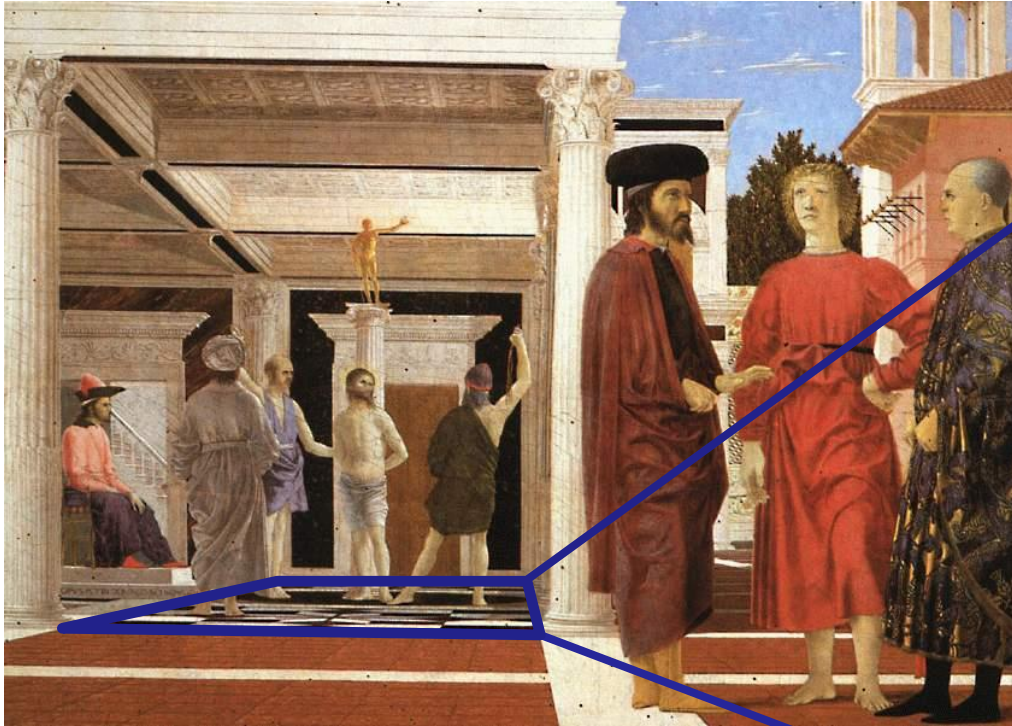


From Martin Kemp
The Science of Art
(manual reconstruction)

The floor (enlarged).
What is the shape of the floor pattern?

Slide adapted from Criminisi

Image Rectification for Art Analysis



The floor (enlarged).
What is the shape of the floor pattern?



Rectified floor
(homography warp)

Slide adapted from Criminisi

Analysing patterns and shapes



What is the (complicated)
shape of the floor pattern?



Automatically rectified floor

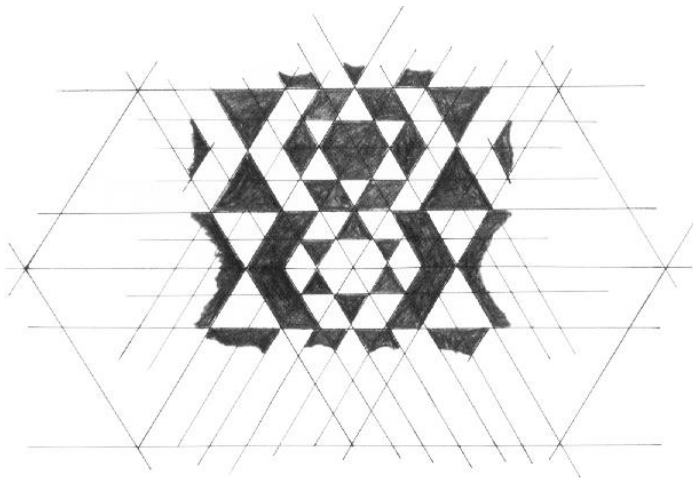
St. Lucy Altarpiece, D. Veneziano

Slide from Criminisi

Analysing patterns and shapes



Automatic
rectification



From Martin Kemp, *The Science of Art*
(*manual reconstruction*)

Interactive Spherical Panorama



https://www.360cities.net/curated_sets/90-new-year's-eve-celebrations

© Aleksandr Reznik, New Year in Prague

Perspective Distortion

Not due to lens flaws

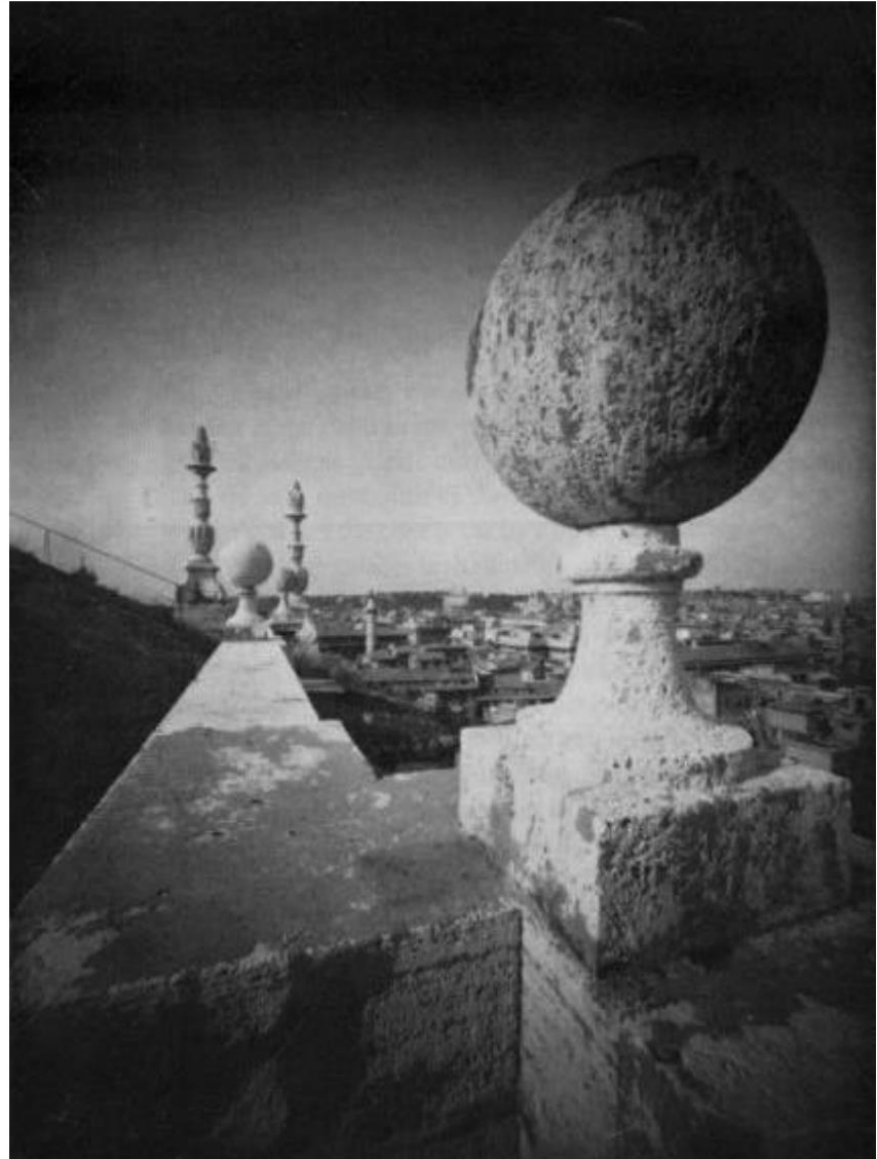
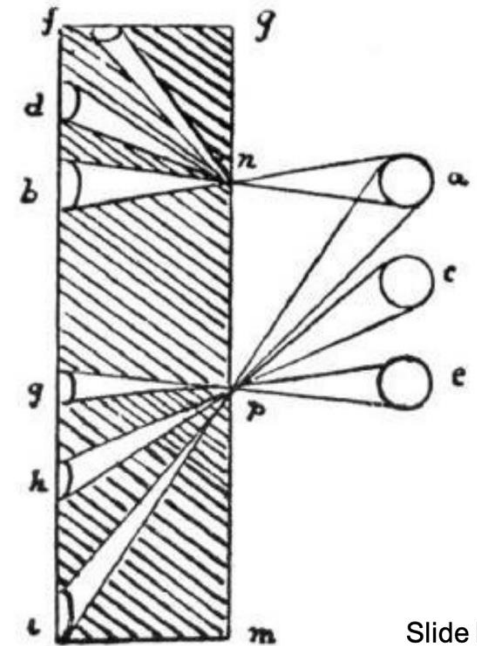
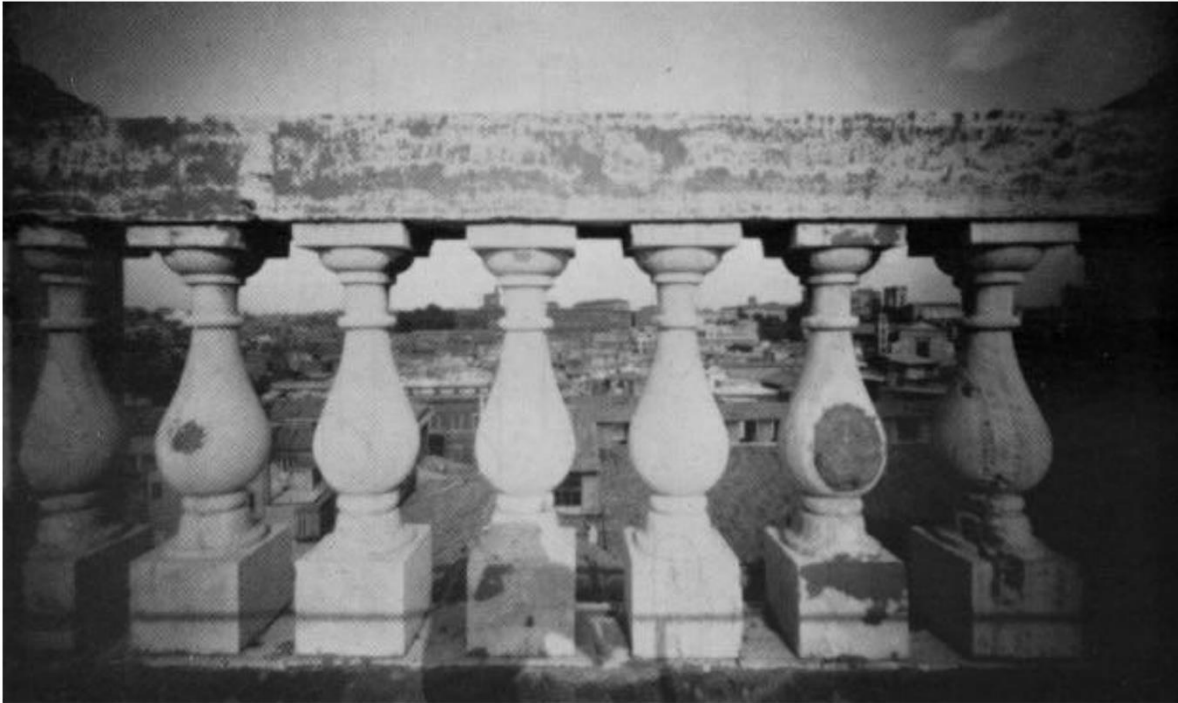


Image source: F. Durand

Problem pointed out by Da Vinci

The exterior columns appear bigger

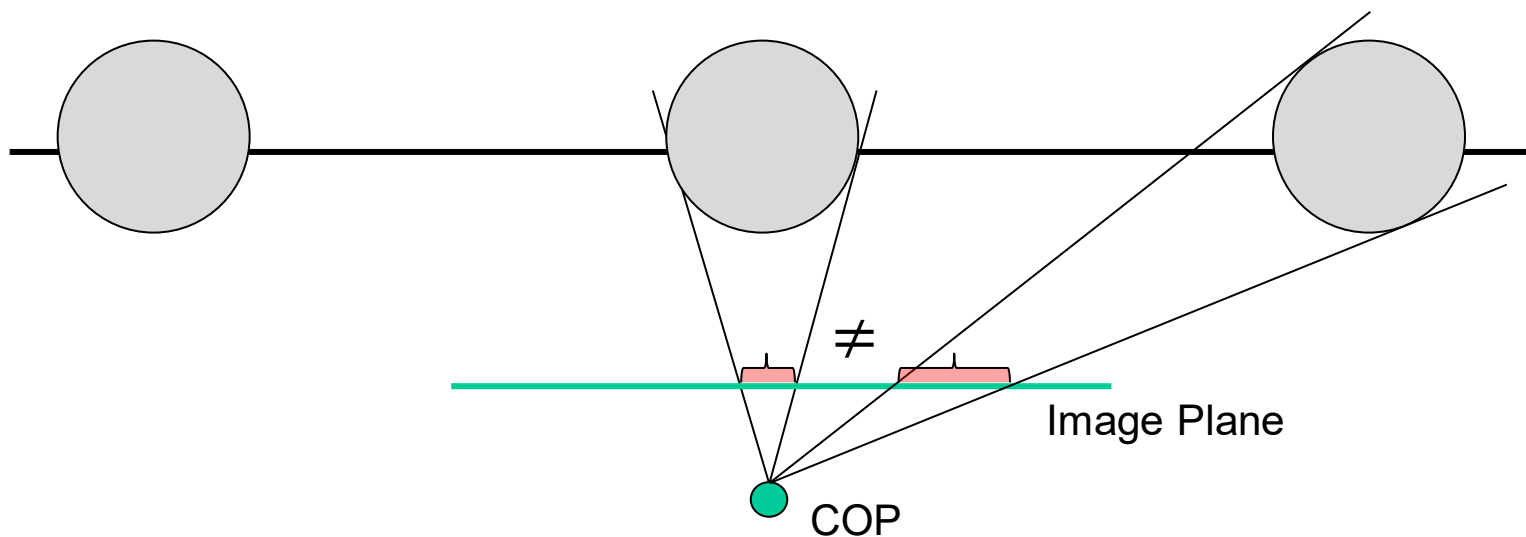


Slide by F. Durand

Image Plane

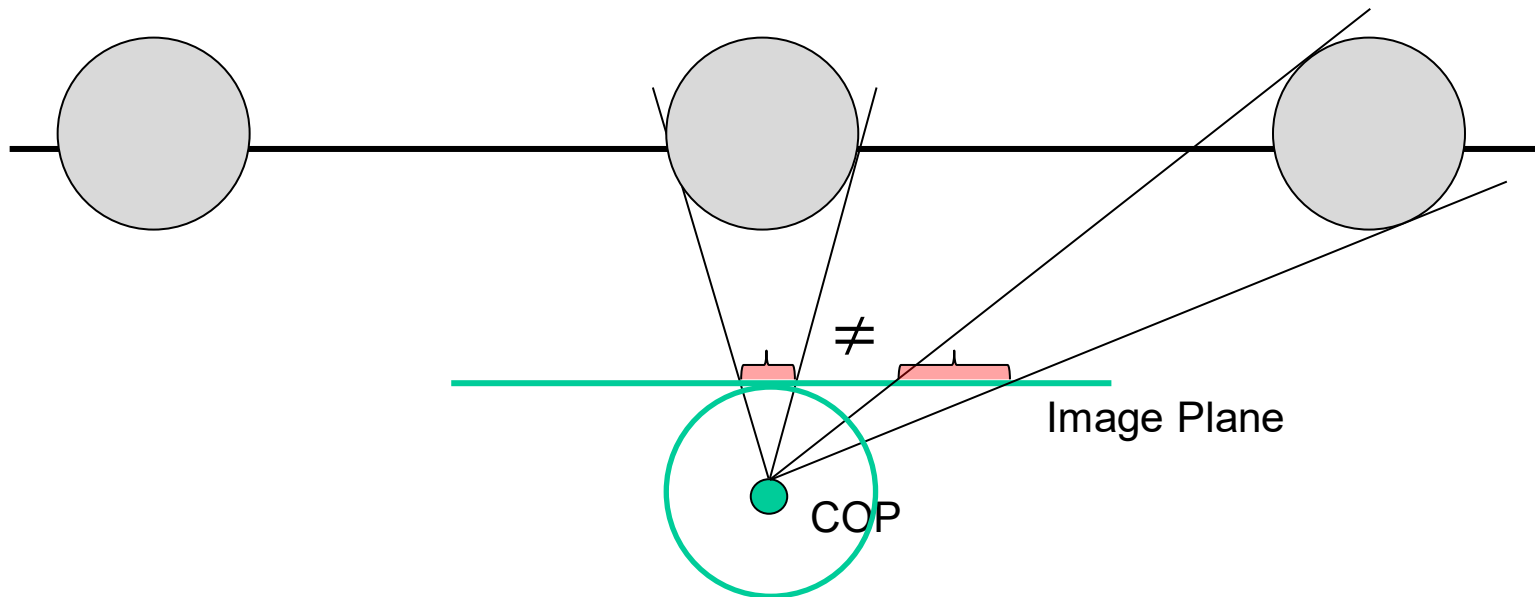
Perspective Distortion

Recall Perspective Projection: $x' = f \frac{x}{z}$ $y' = f \frac{y}{z}$



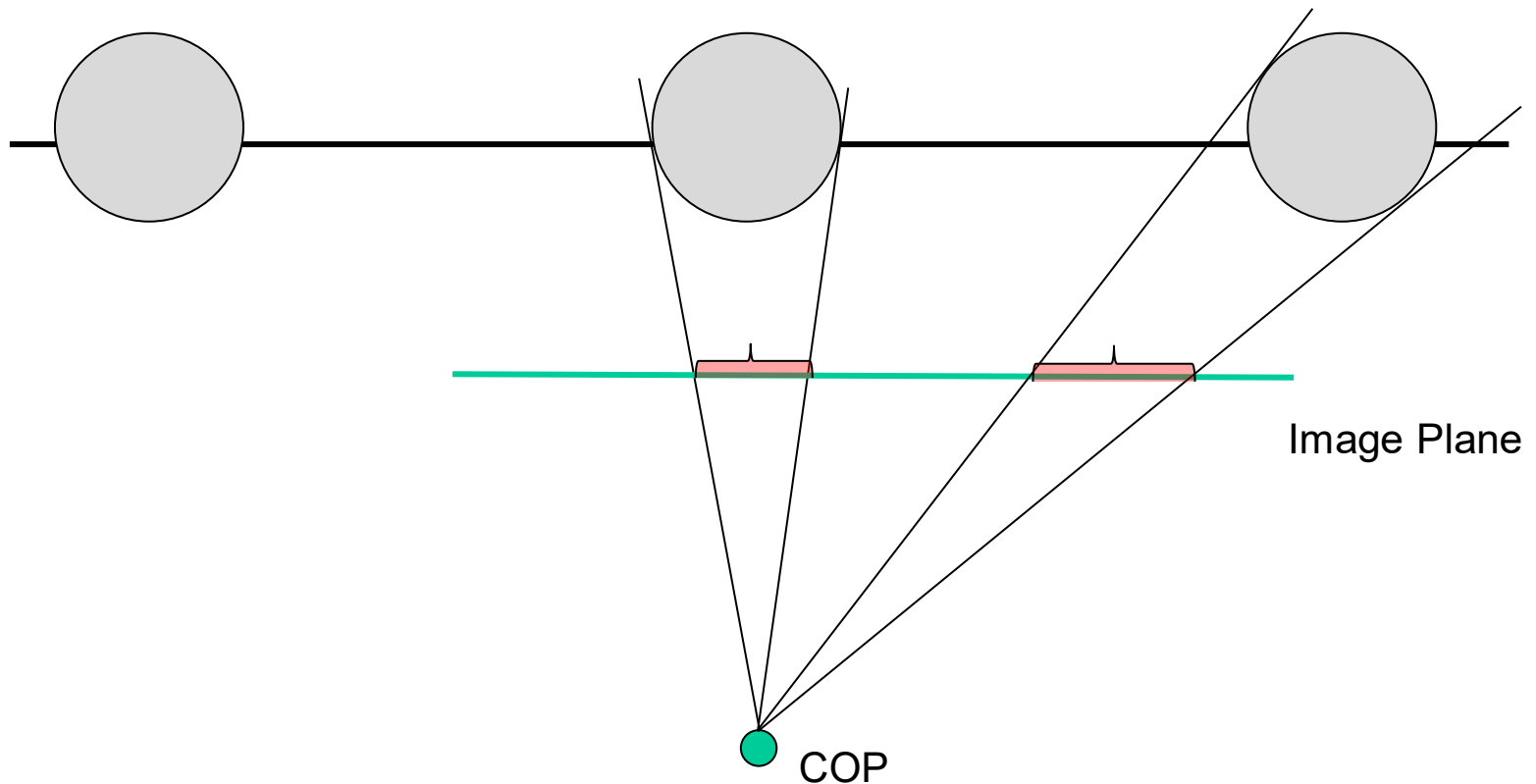
Perspective Distortion

With a spherical projection plane



Perspective Distortion

Less noticeable with long focal length (i.e. you see distortion more with wide-angle camera)



Renaissance Painters' solution



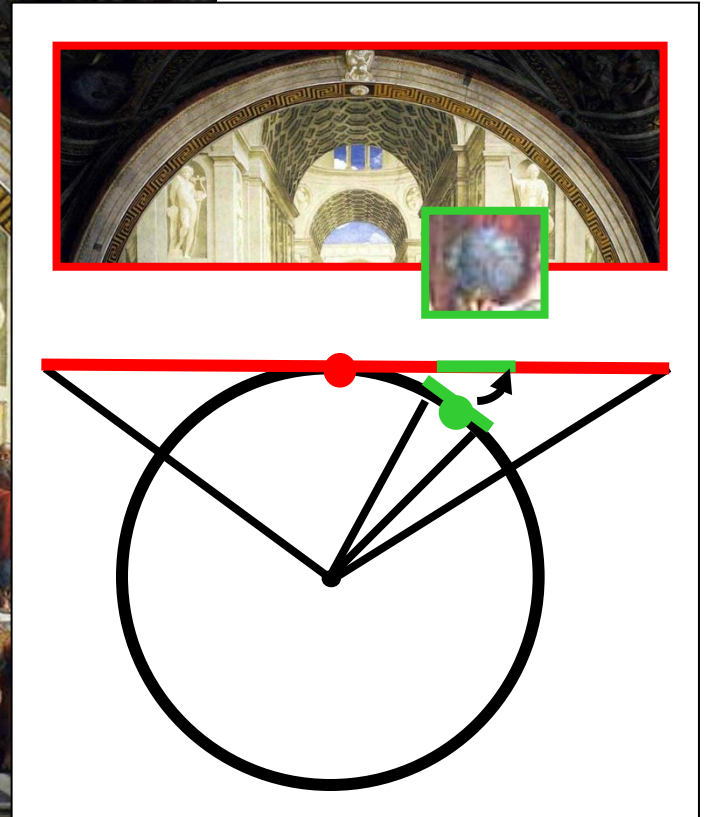
“School of Athens”, Raffaello Sanzio ~1510

Give a separate treatment to different parts of the scene!!

Personalized projections



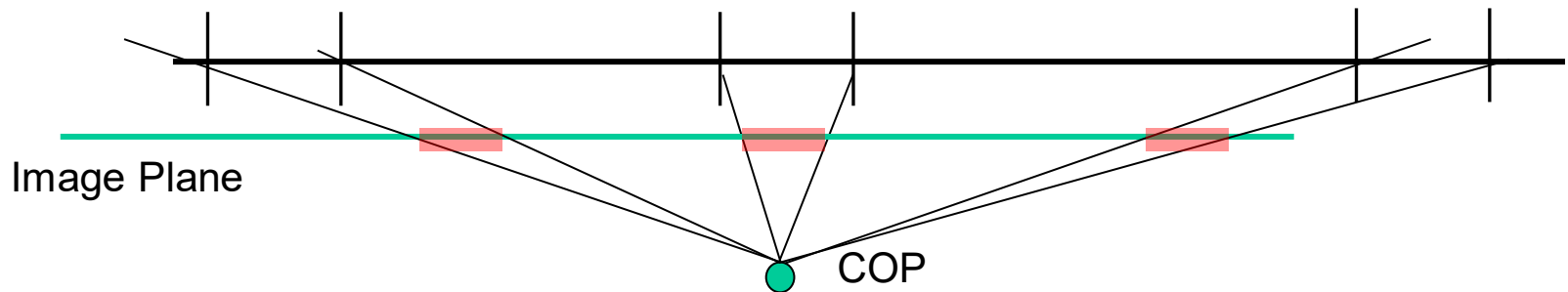
“School of Athens”, Raffaello Sanzio ~1510



Give a separate treatment to different parts of the scene!!

Perspective Distortion

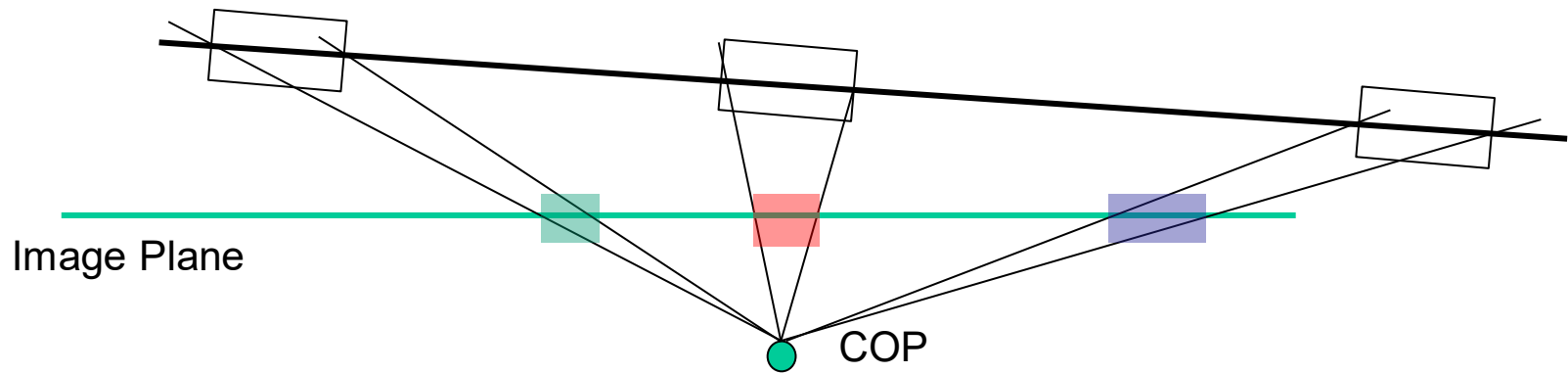
How about if the scene is planar?



But this is a very special case..

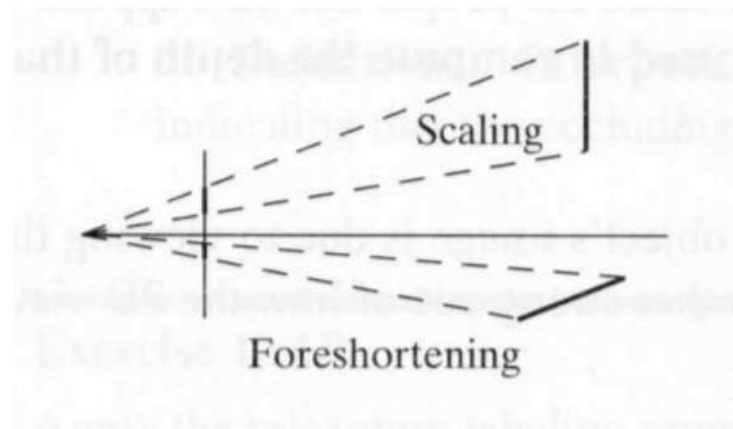
Perspective Distortion

More likely..



Foreshortening

- When a line (or surface) is parallel to the image plane, the effect of perspective projection is *scaling*.
- When an line (or surface) is not parallel to the image plane, we use the term *foreshortening* to describe the projective distortion (i.e., the dimension parallel to the optical axis is compressed relative to the frontal dimension).



Mosaics: stitching images together



virtual wide-angle camera

Why Mosaic?

Are you getting the whole picture?

- Compact Camera FOV = $50 \times 35^\circ$



Why Mosaic?

Are you getting the whole picture?

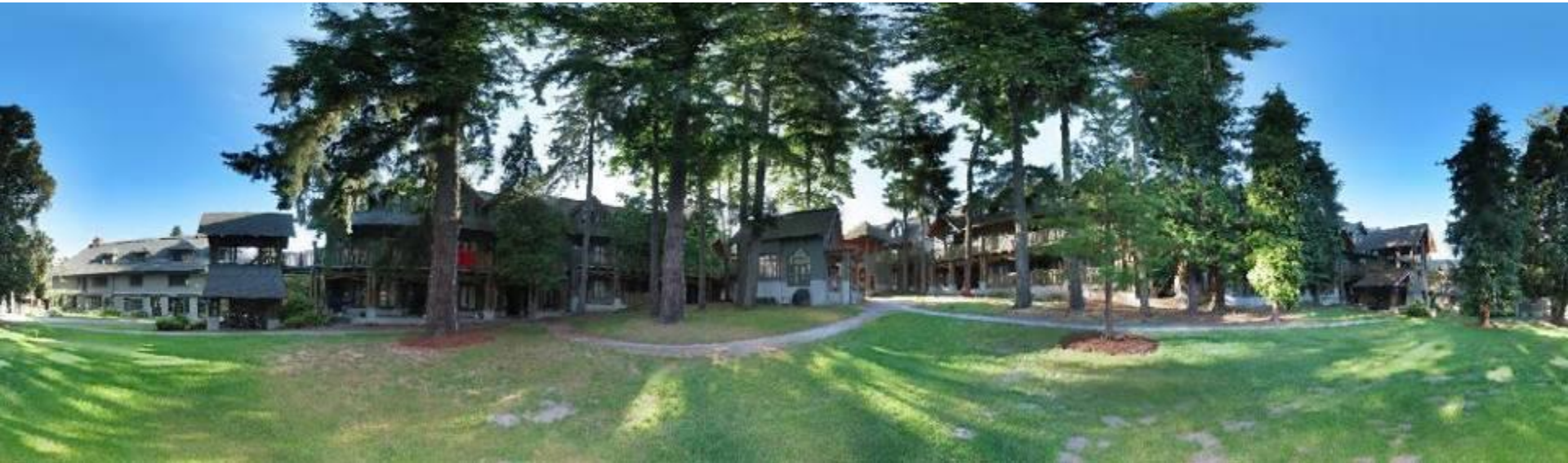
- Compact Camera FOV = $50 \times 35^\circ$
- Human FOV = $200 \times 135^\circ$



Why Mosaic?

Are you getting the whole picture?

- Compact Camera FOV = $50 \times 35^\circ$
- Human FOV = $200 \times 135^\circ$
- Panoramic Mosaic = $360 \times 180^\circ$



Naïve Stitching



left on top



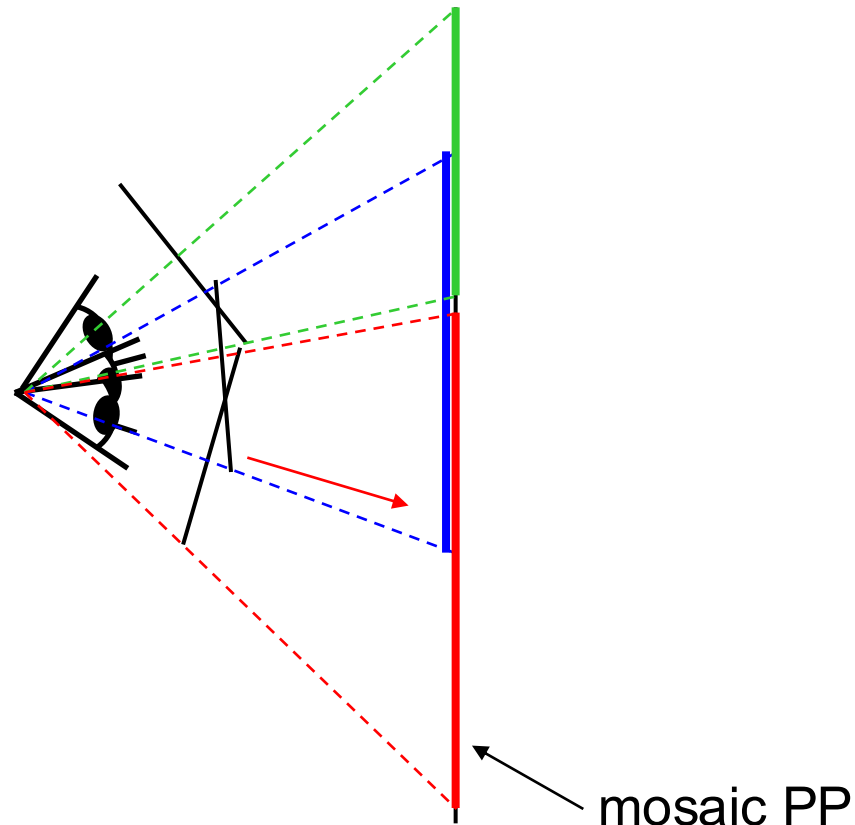
right on top



Translations are not enough to align the images



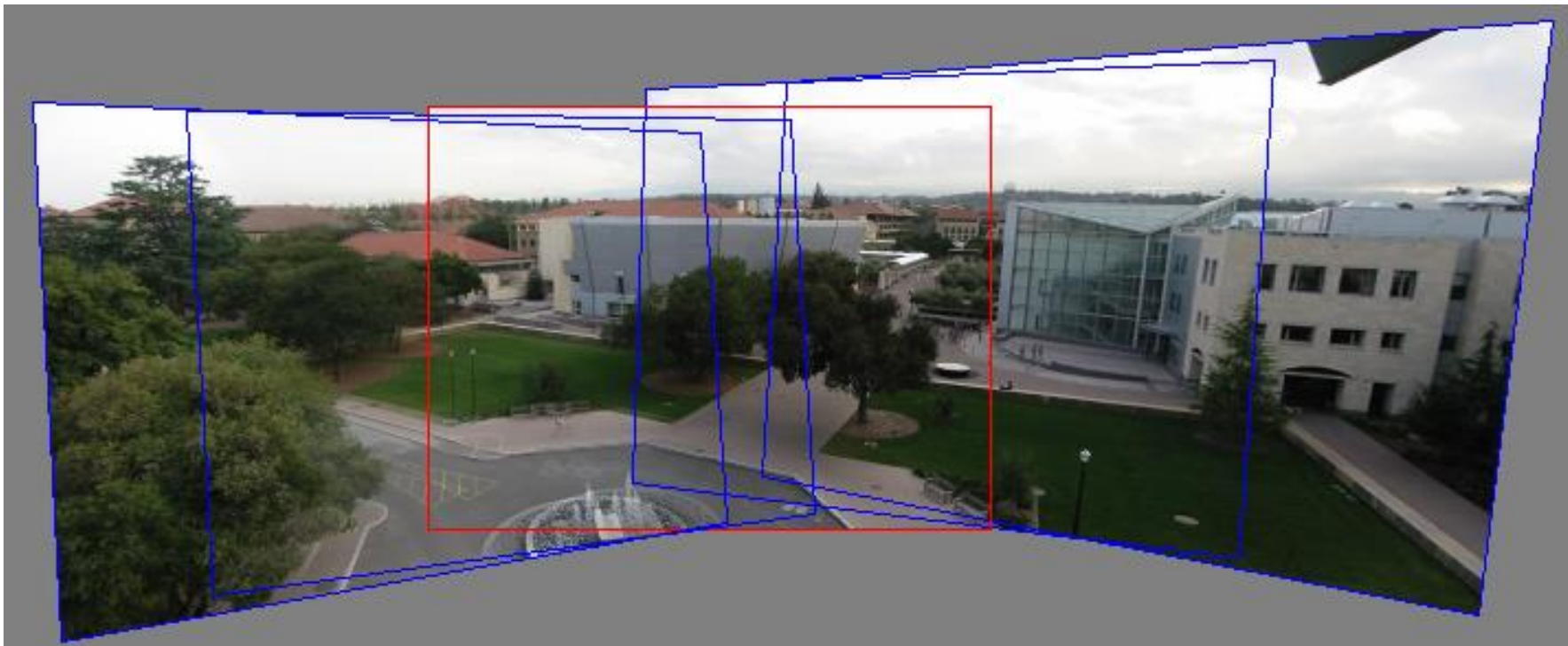
Image reprojection



The mosaic has a natural interpretation in 3D

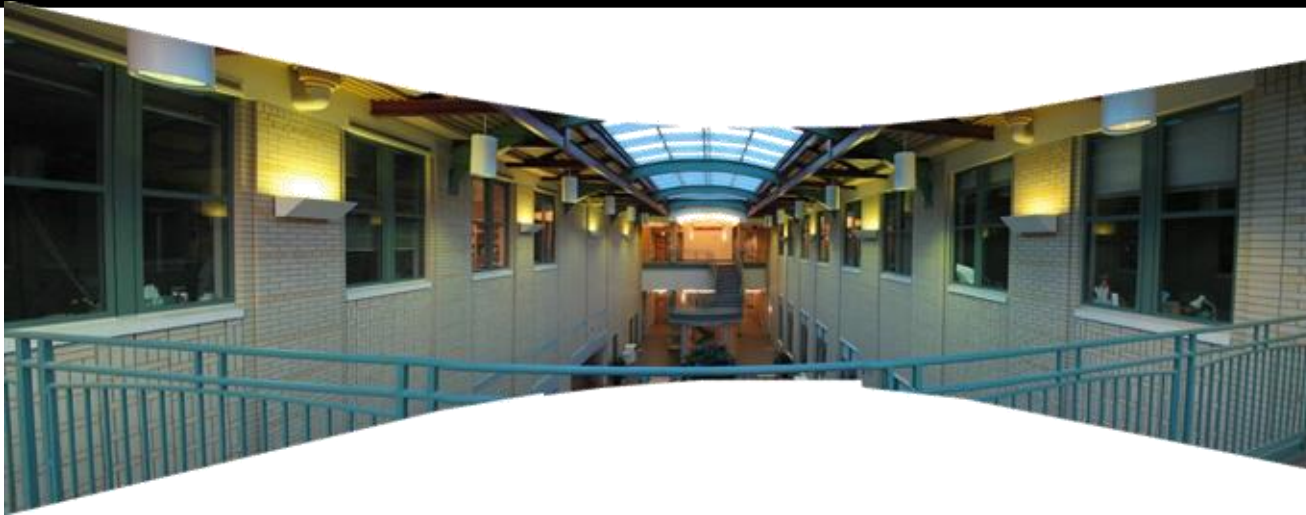
- The images are reprojected onto a common plane
- The mosaic is formed on this plane
- Mosaic is a *synthetic wide-angle camera*

Panoramas



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

Programming Project #4 (part 1)



Homographies and Panoramic Mosaics

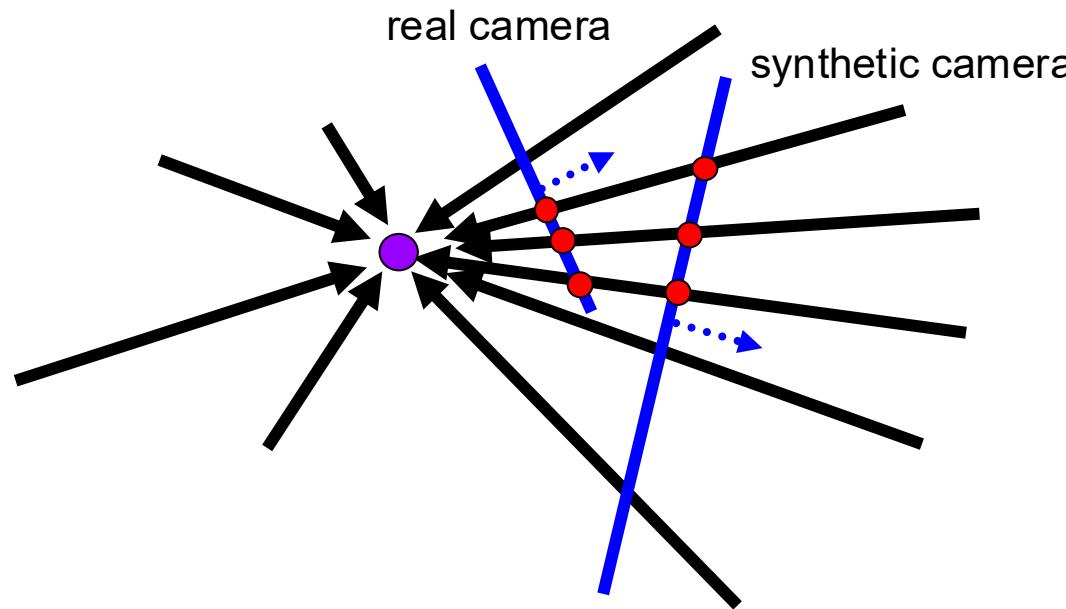
- Capture photographs (and possibly video)
 - Might want to use tripod
- Compute homographies (define correspondences)
 - will need to figure out how to setup system of eqs.
- (un)warp an image (undo perspective distortion)
- Produce panoramic mosaics (with blending)
- Do some of the Bells and Whistles

Example homography final project



Think about this: When is this not true?

We can generate any synthetic camera view as long as it has **the same center of projection**!



What happens if there are two center of projection?
(you move your head)