Homographies and Panoramas

© Andrew Campbell

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

CS180: Intro to Computer Vision and Comp. Photo
Angjoo Kanazawa and Alexei Efros, UC Berkeley, Fall 2023

Logistics

Project 3 due today!

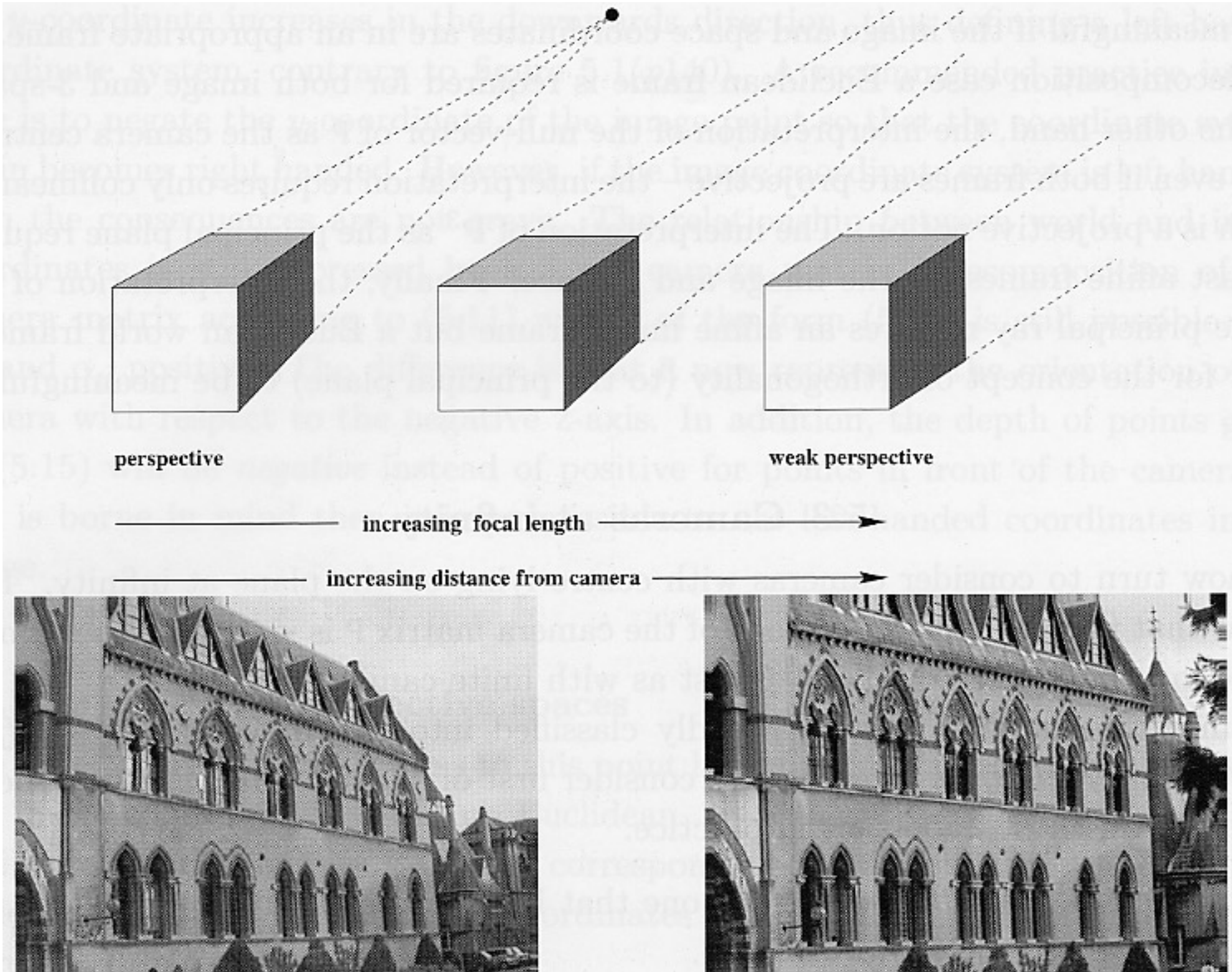
Project 4 released this Wednesday

All is due 10/25 (with some midpoint checks)!! It is quite challenging!
Make sure to start early!

Try to preserve slip days for emergencies.

Project 2 voting is out on Ed!!

Recap



From Zisserman & Hartley

Focal length / distance in portraiture

Perspective Distortion

Not due to lens flaws

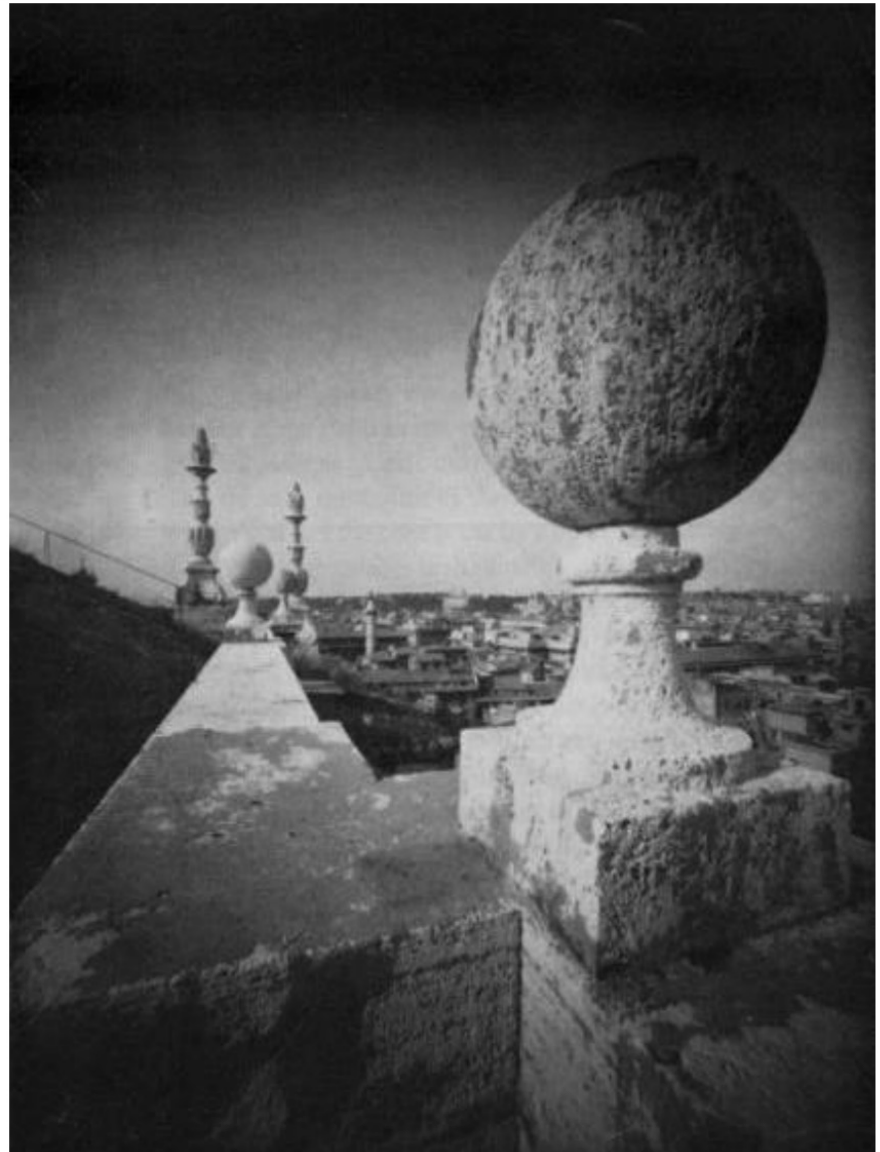
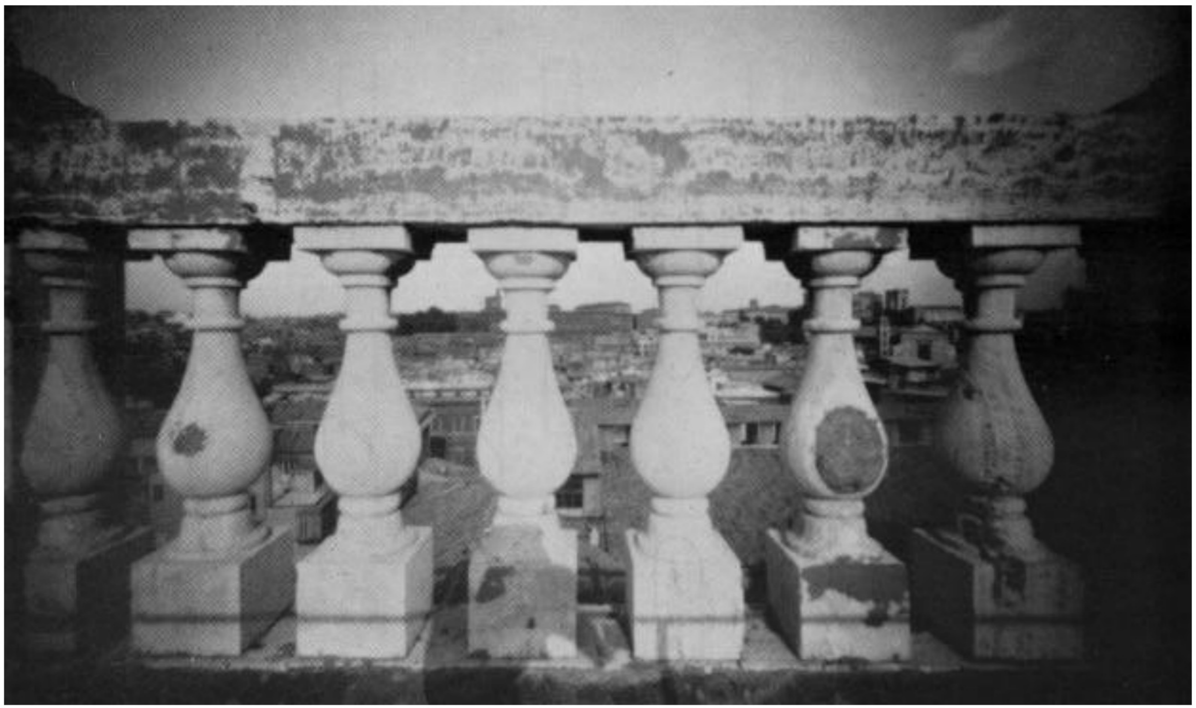


Image source: F. Durand

Problem pointed out by Da Vinci

The exterior columns appear bigger



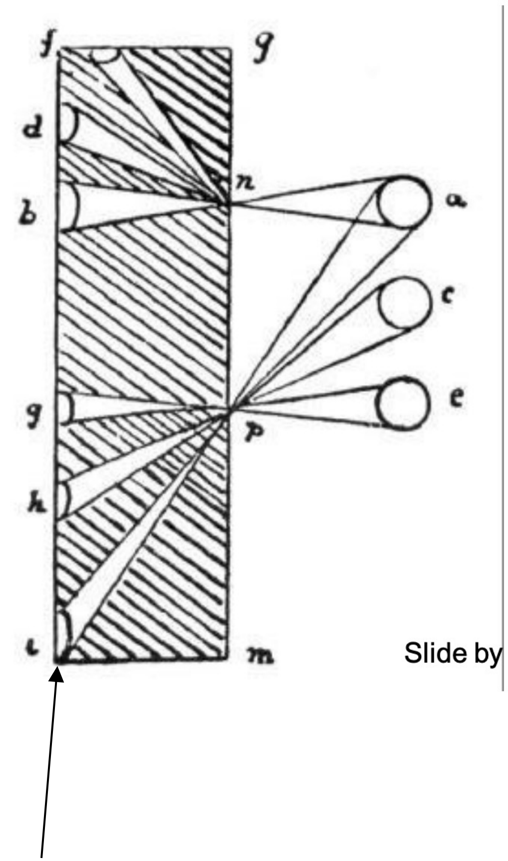
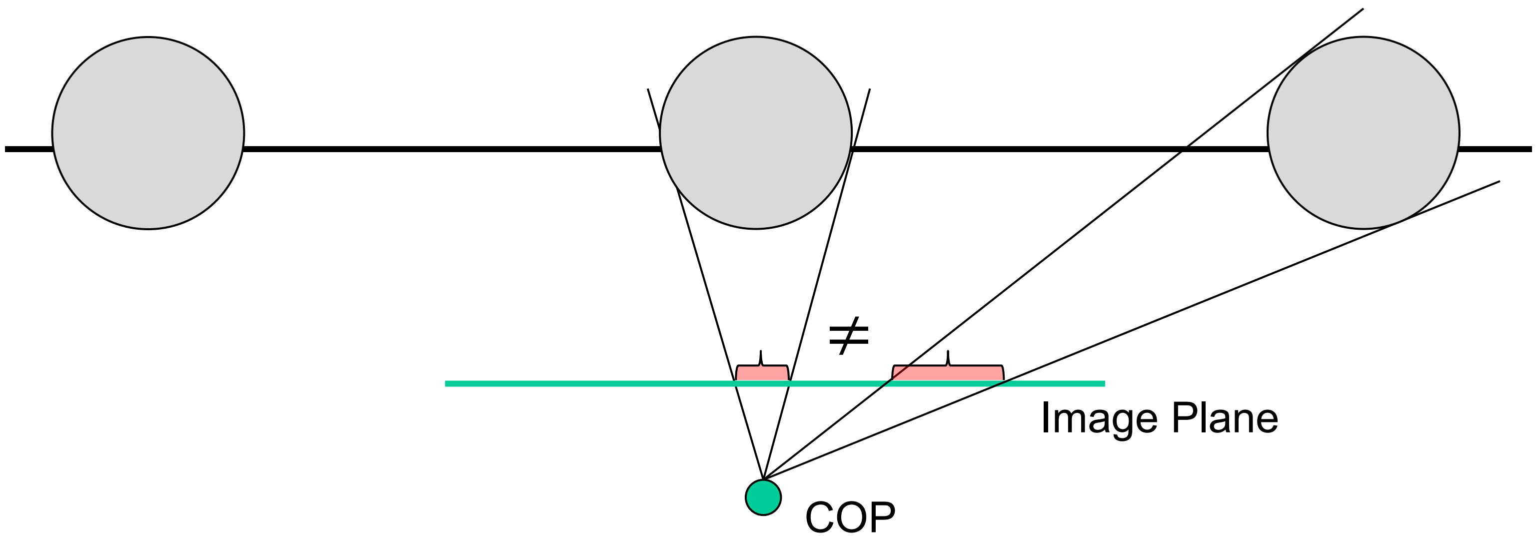


Image Plane

Slide by F. Durand

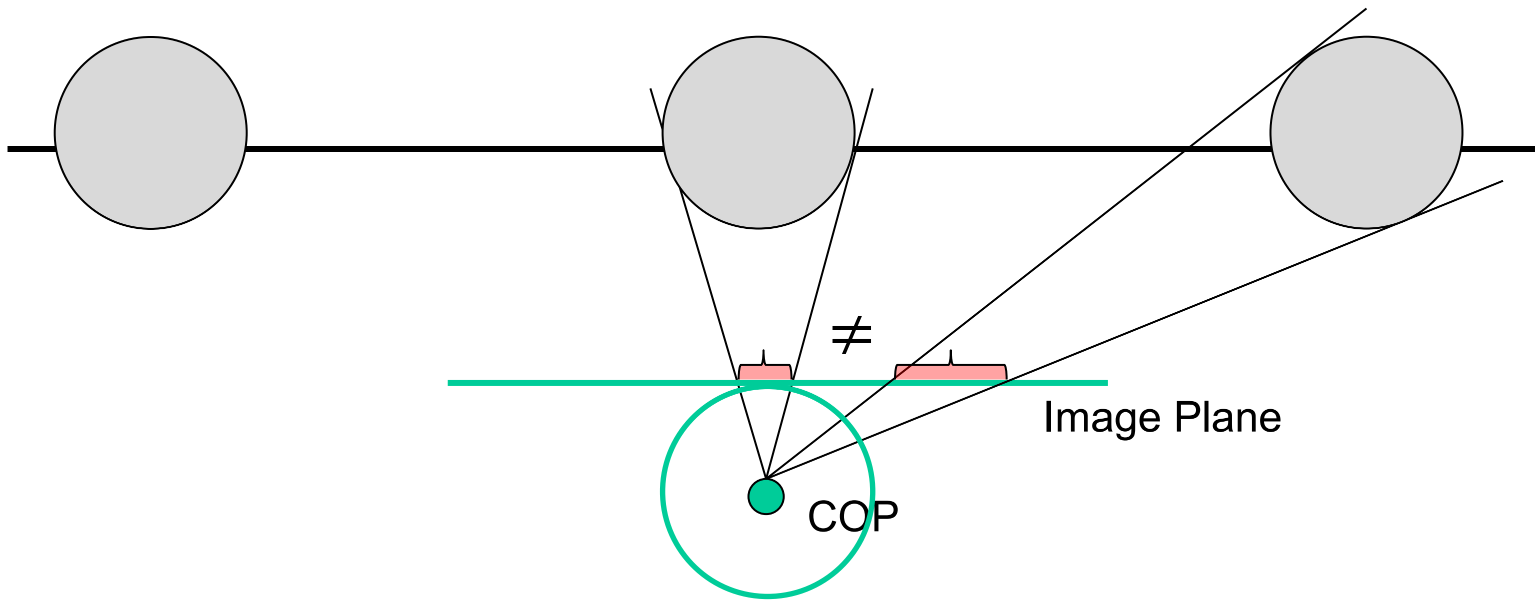
Perspective Distortion

Recall Perspective Projection: $x' = f\frac{X}{Z}$ $y' = f\frac{Y}{Z}$



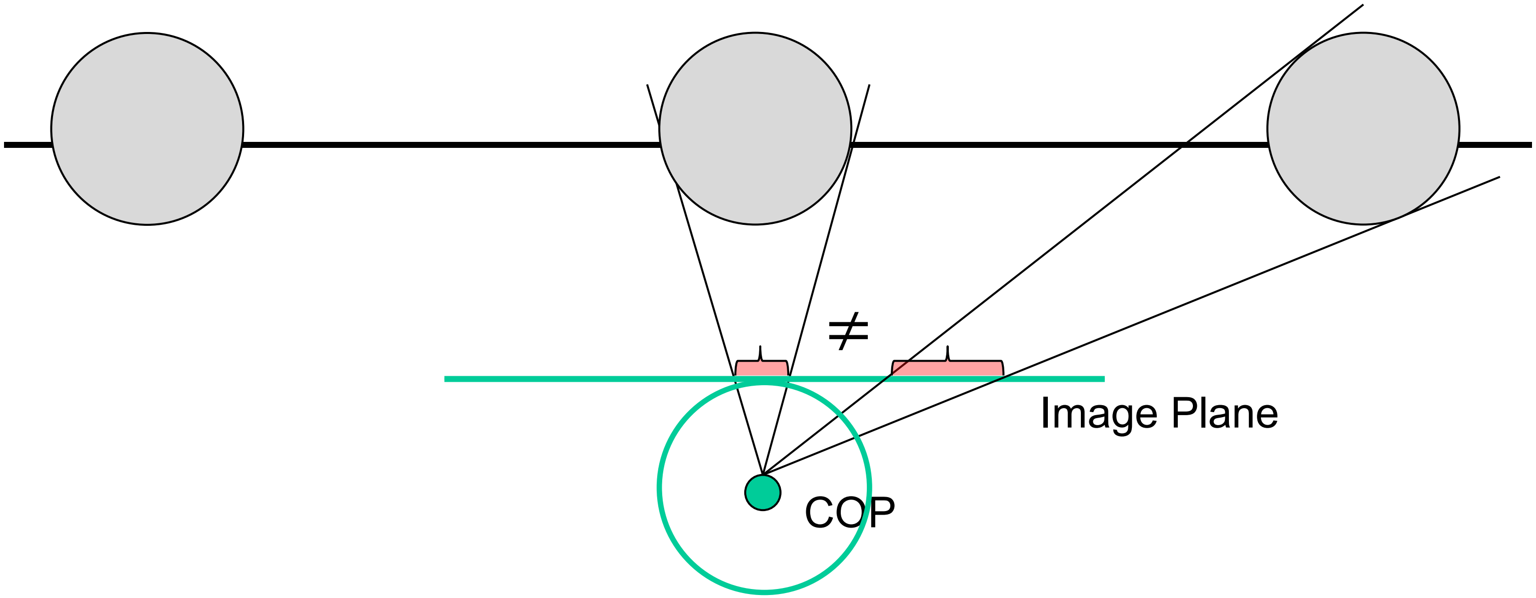
Perspective Distortion

With a spherical projection plane



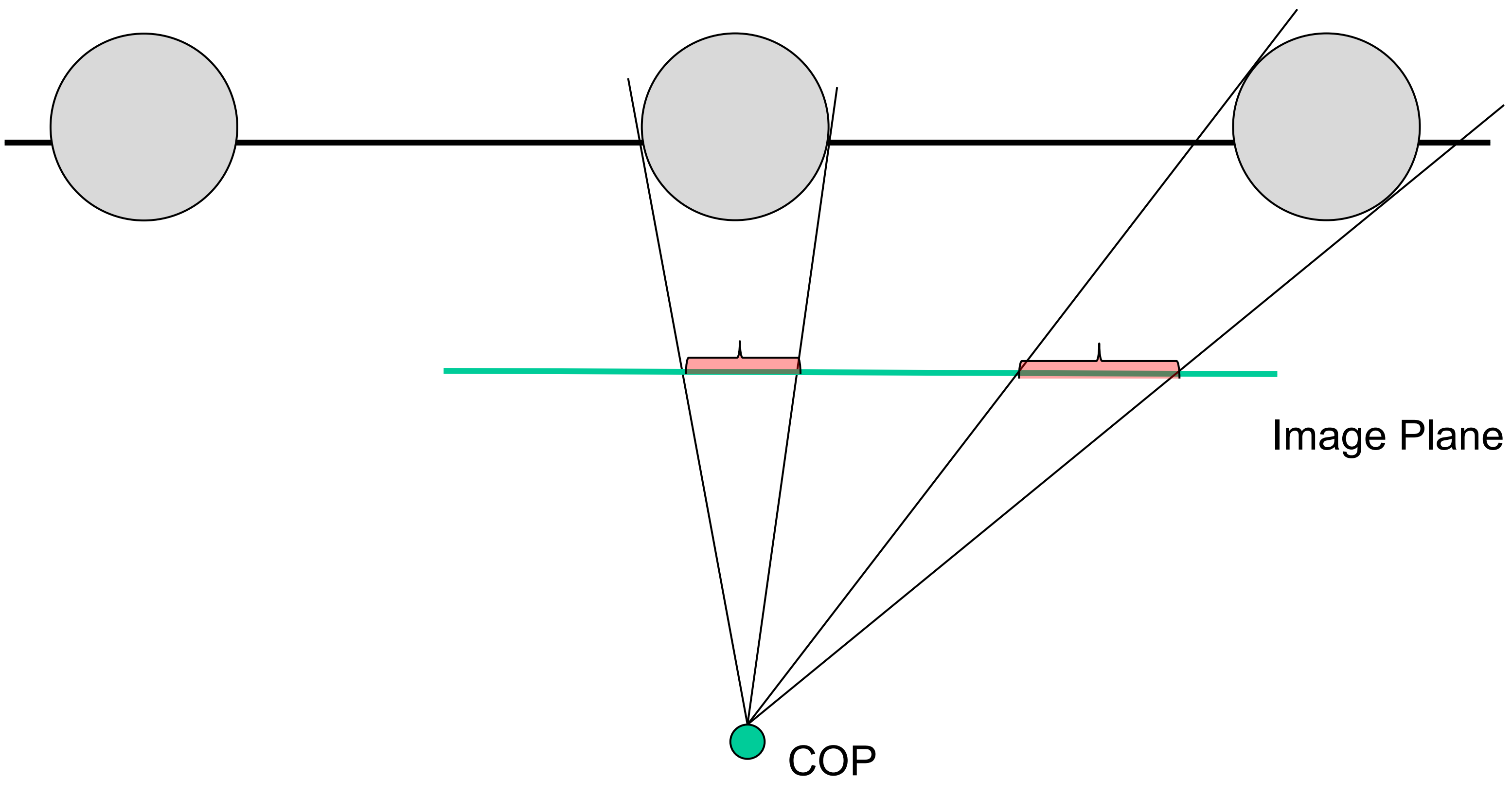
Perspective Distortion

With a spherical projection plane



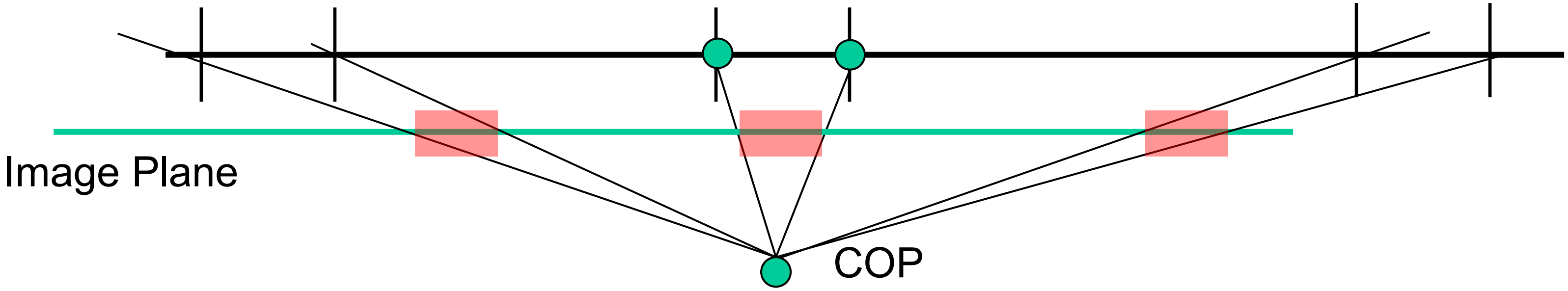
Perspective Distortion

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



Perspective Distortion

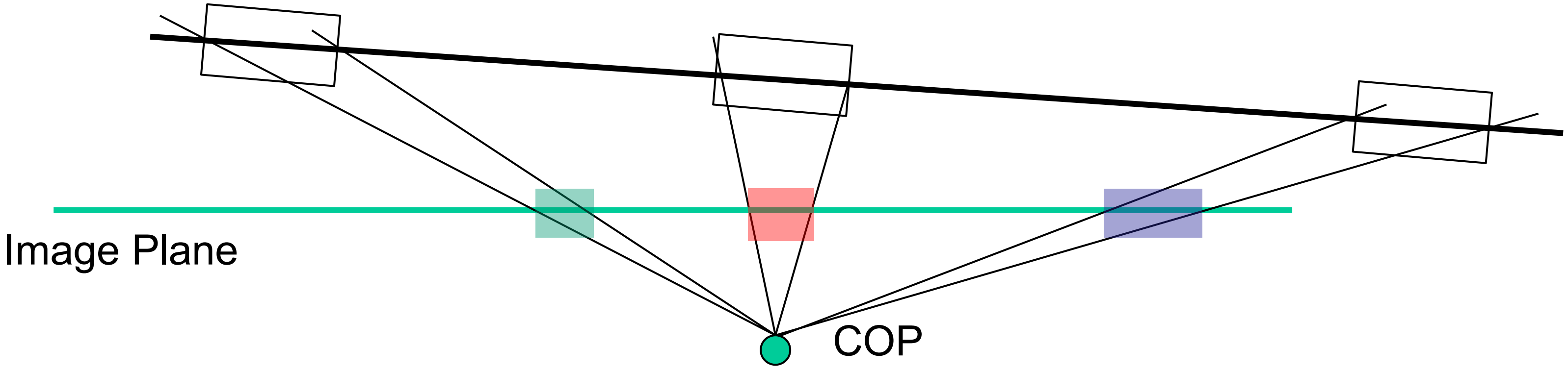
It’s about the change in depths



But this is a very special case..

Perspective Distortion

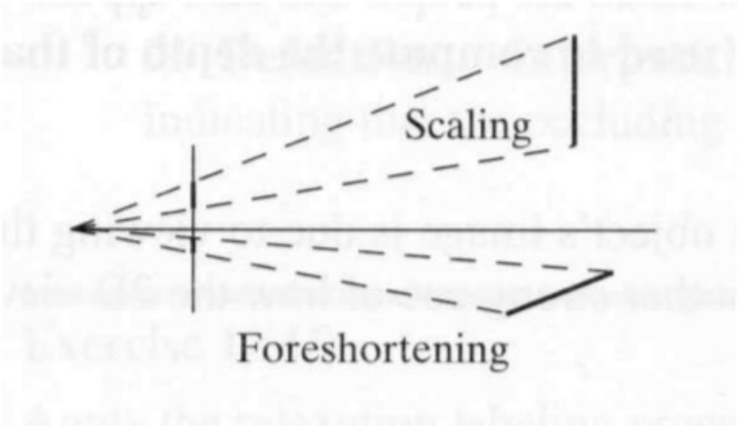
More likely..



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).



Fixing Perspective Distortion

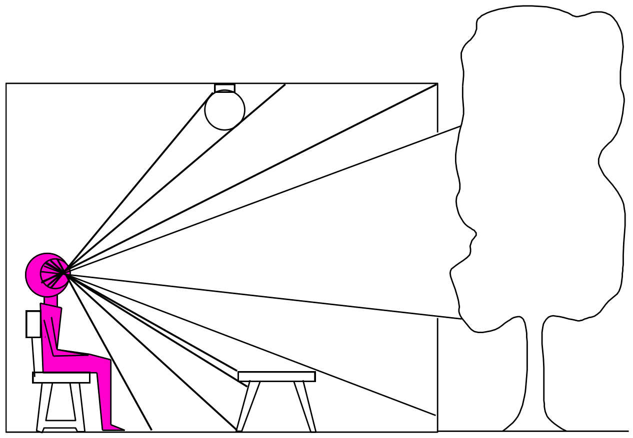
(a) A wide-angle photo with distortions on subjects' faces.

(b) Distortion-free photo by our method.

Distortion-Free Wide-Angle Portraits on Camera Phones
Shih et al. SIGGRAPH 2019

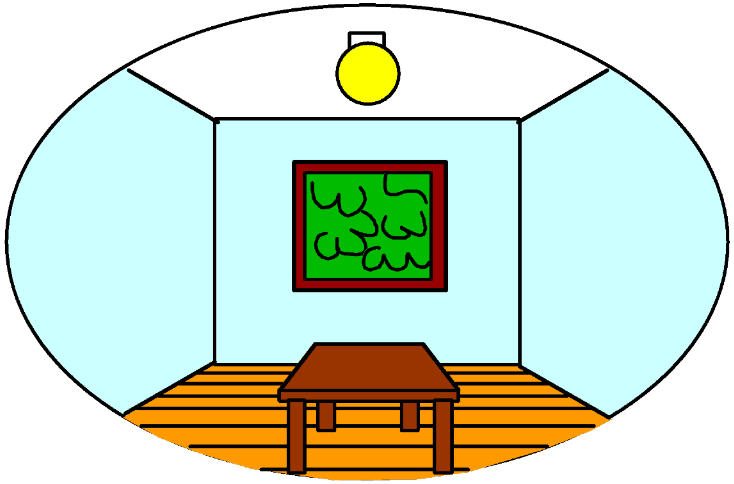
What do we see?

3D world



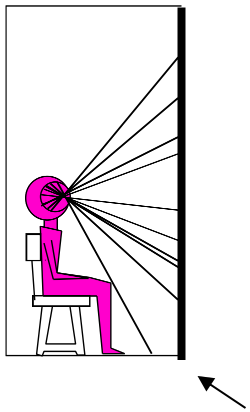
Point of observation

2D image



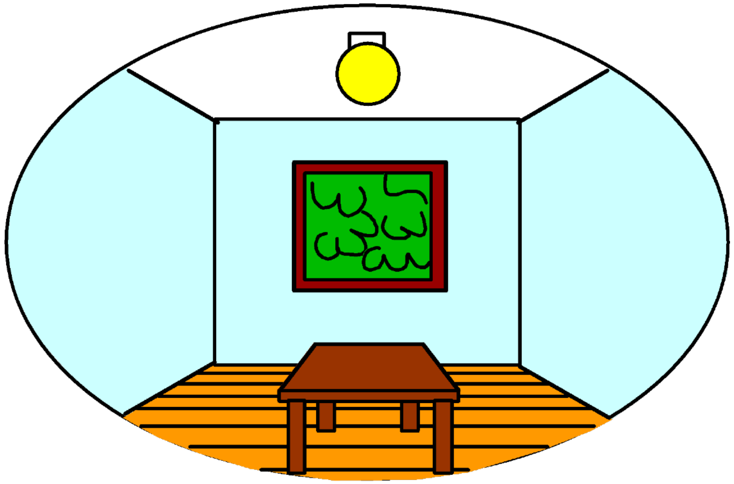
What do we see?

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”
- “Latest” take: *The Matrix*

Physics:

- *Slowglass* might be possible?

Computer Science:

- Virtual Reality

To simulate we need to know:

What does a person see?

The Plenoptic Function

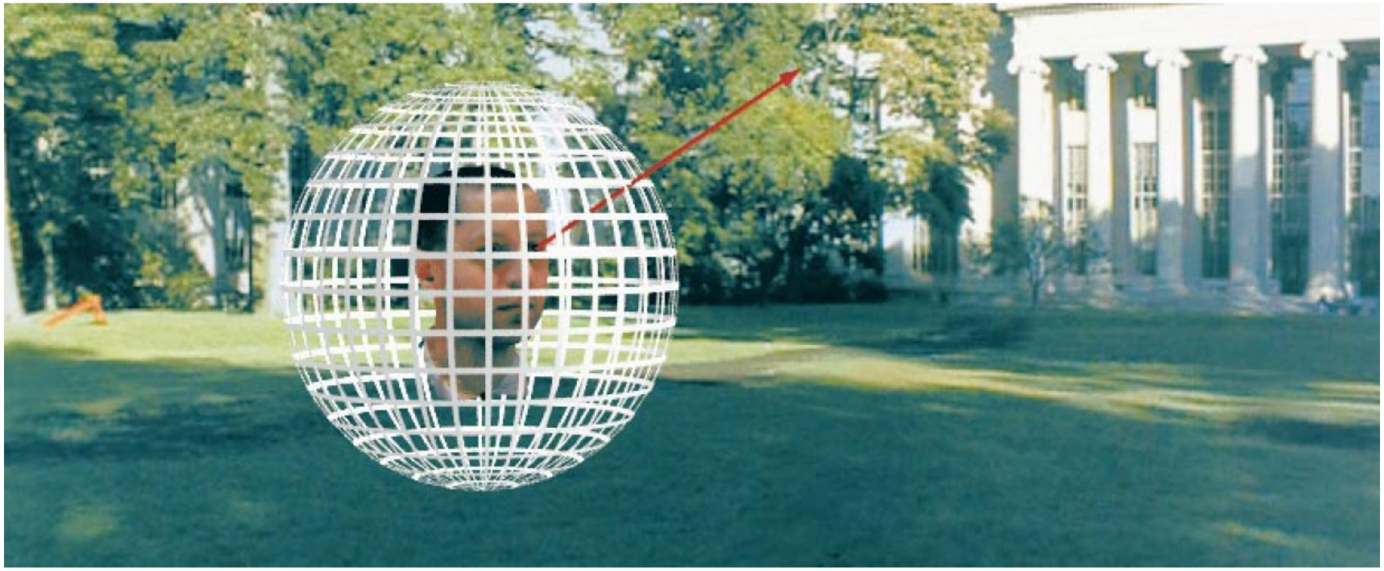


Figure by Leonard McMillan

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 he can see…

Grayscale snapshot

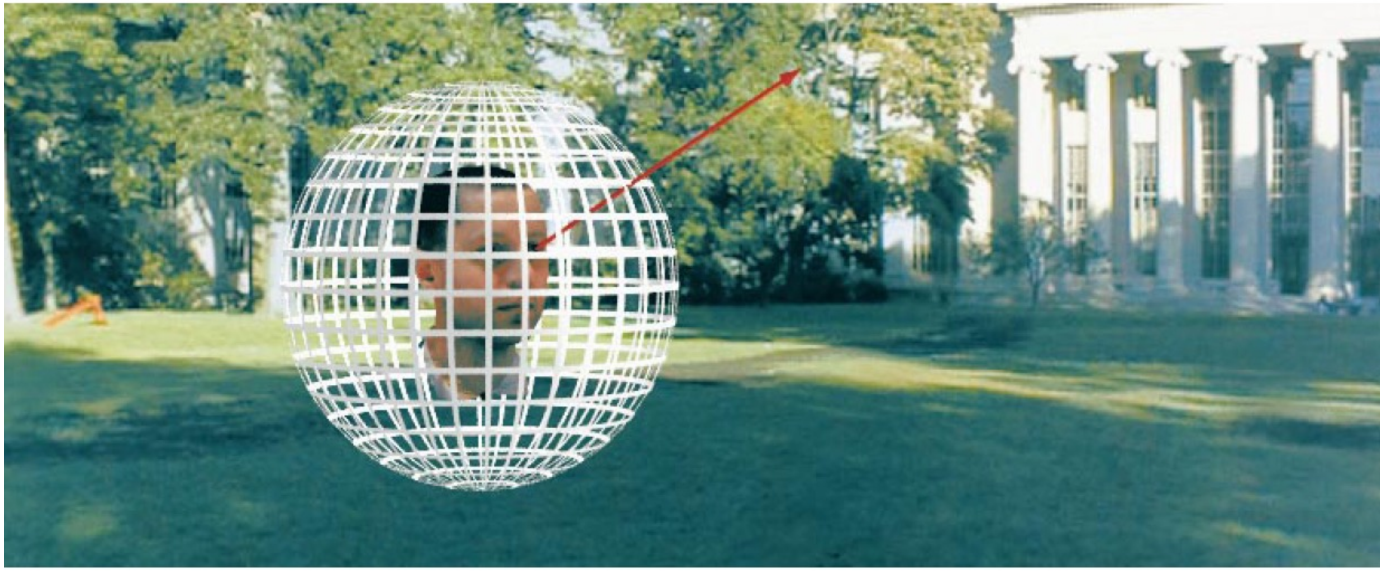
$$P(\theta,\phi)$$

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

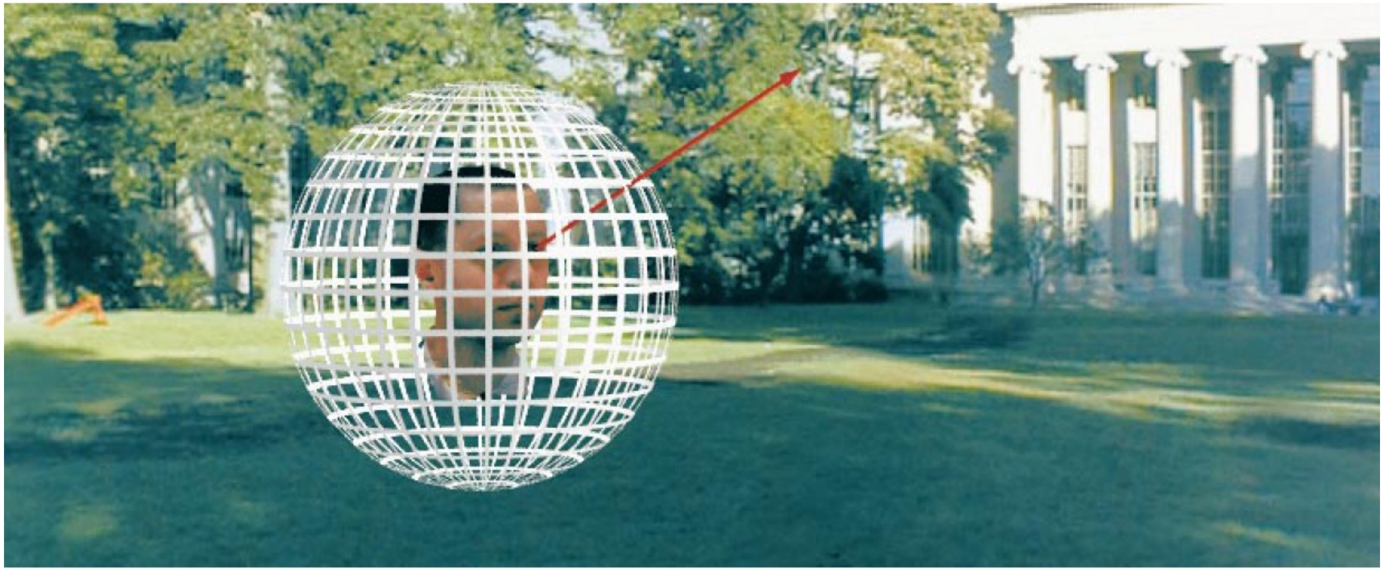


$$P(\theta,\phi,\lambda)$$

is intensity of light

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

A movie

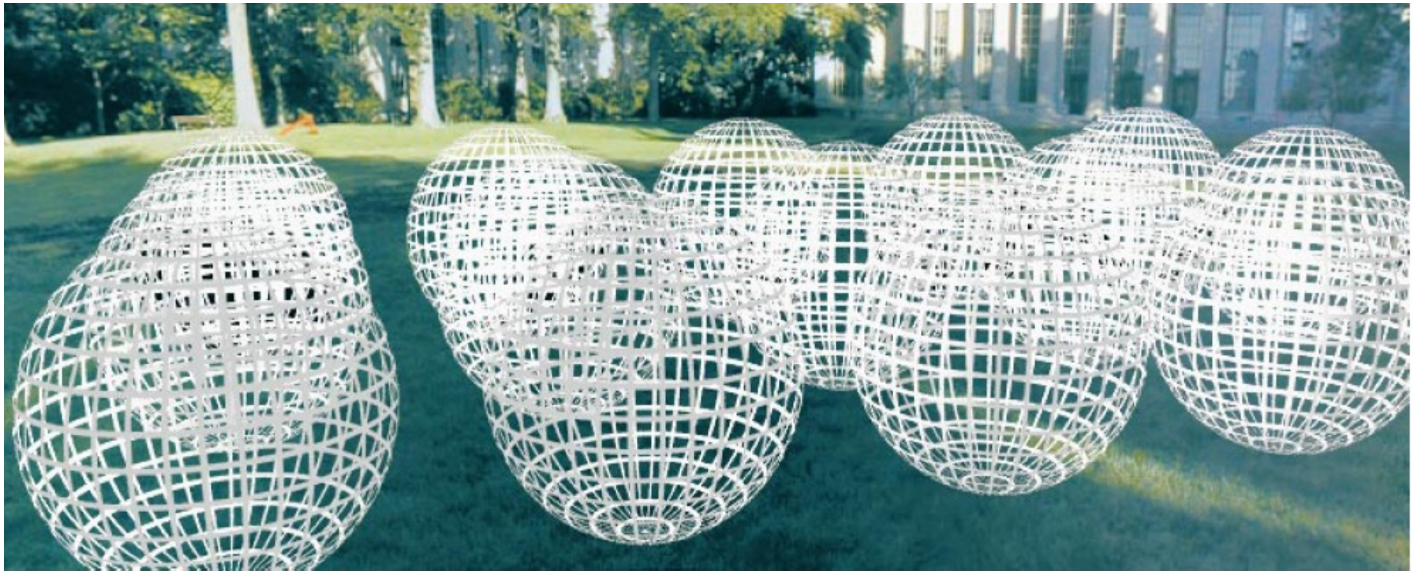


$$P(\theta,\phi,\lambda,t)$$

is intensity of light

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

Holographic movie

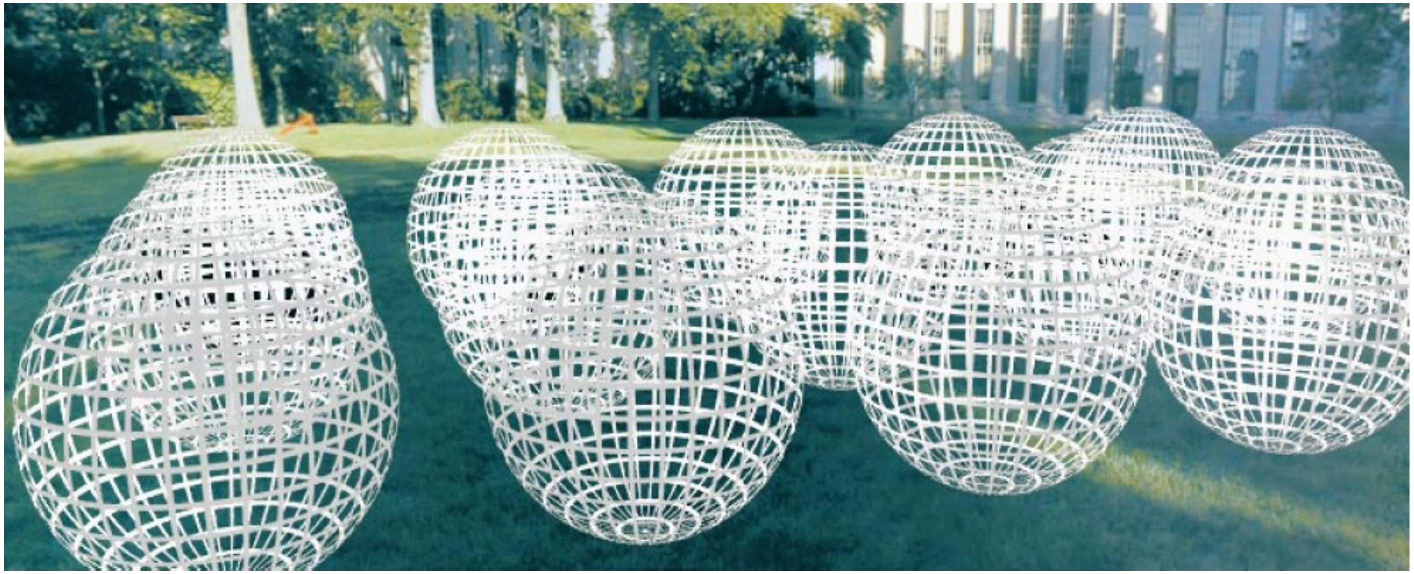


$$P(\theta,\phi,\lambda,t,V_X,V_Y,V_Z)$$

is intensity of light

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

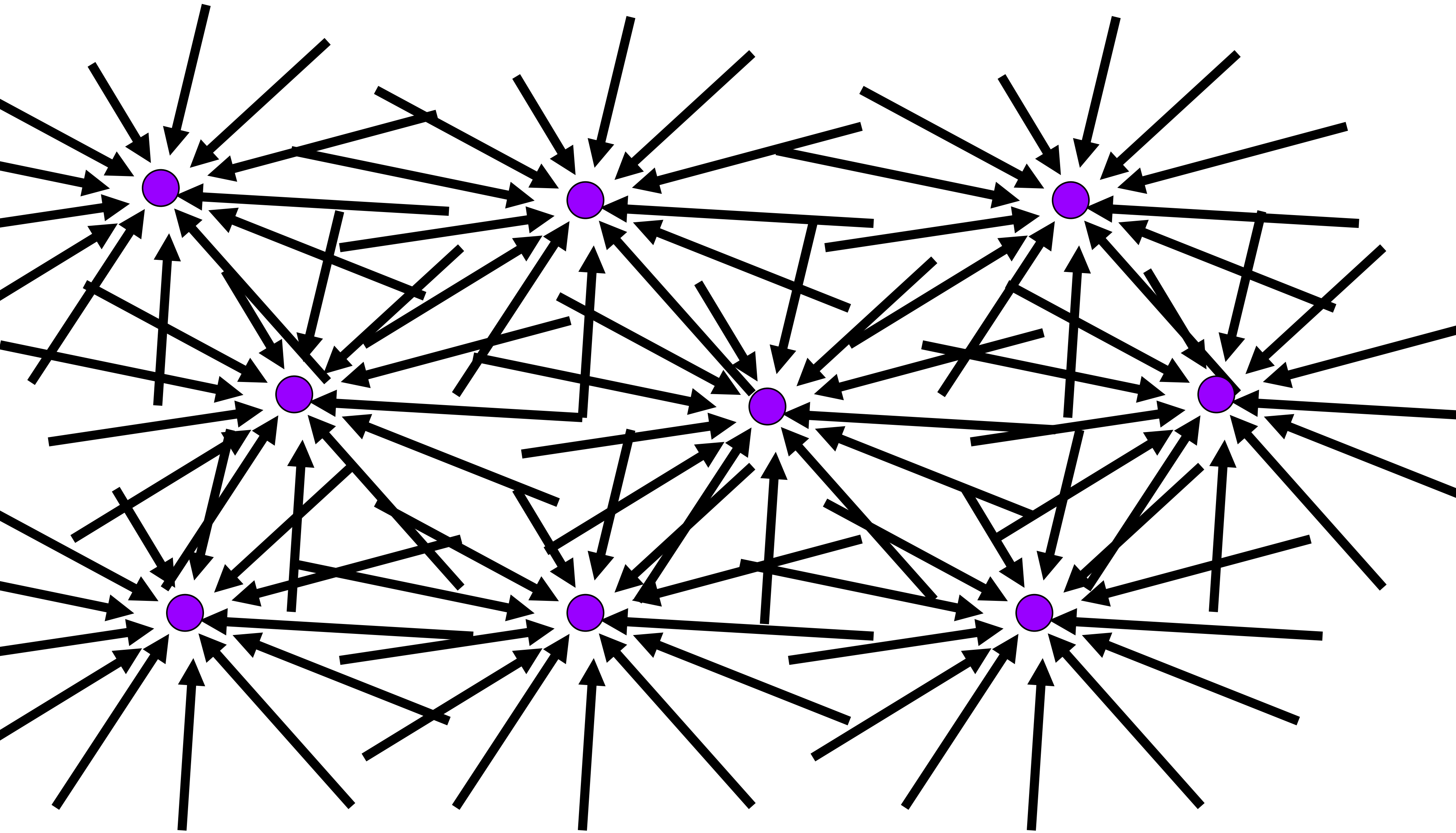
The Plenoptic Function



$$P(\theta,\phi,\lambda,t,V_X,V_Y,V_Z)$$

- 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…

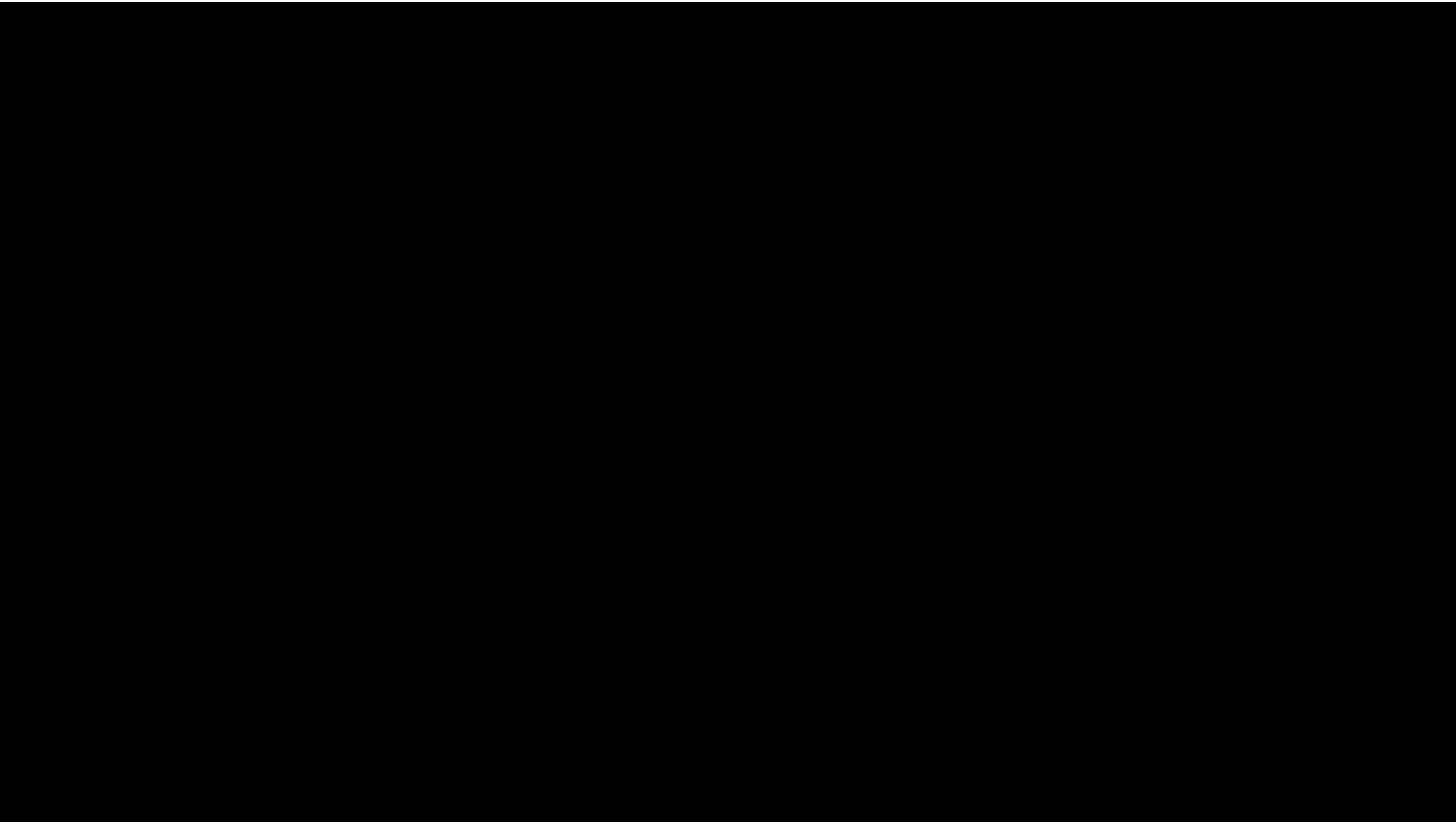
Sampling Plenoptic Function (top view)



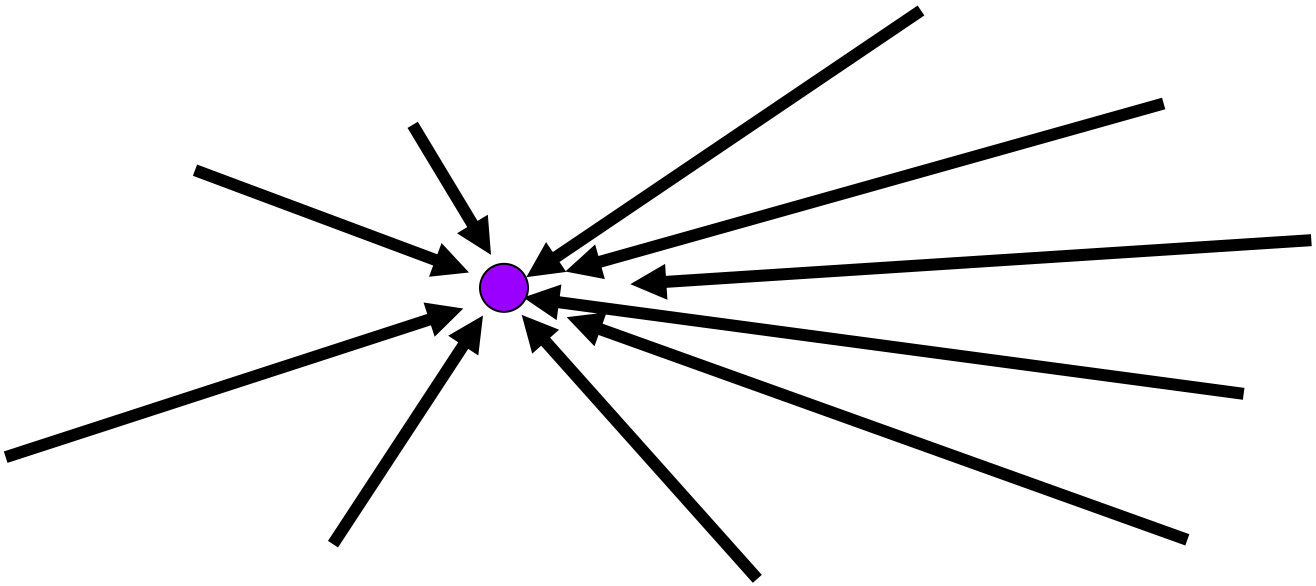
Just lookup -- Quicktime VR

QuickTime VR 1995

Apple Quicktime VR



What is an image?



Slide by Rick Szeliski and Michael Cohen

Spherical Panorama

See also: 2003 New Years Eve
http://www.panoramas.dk/New-Year/times-square.html

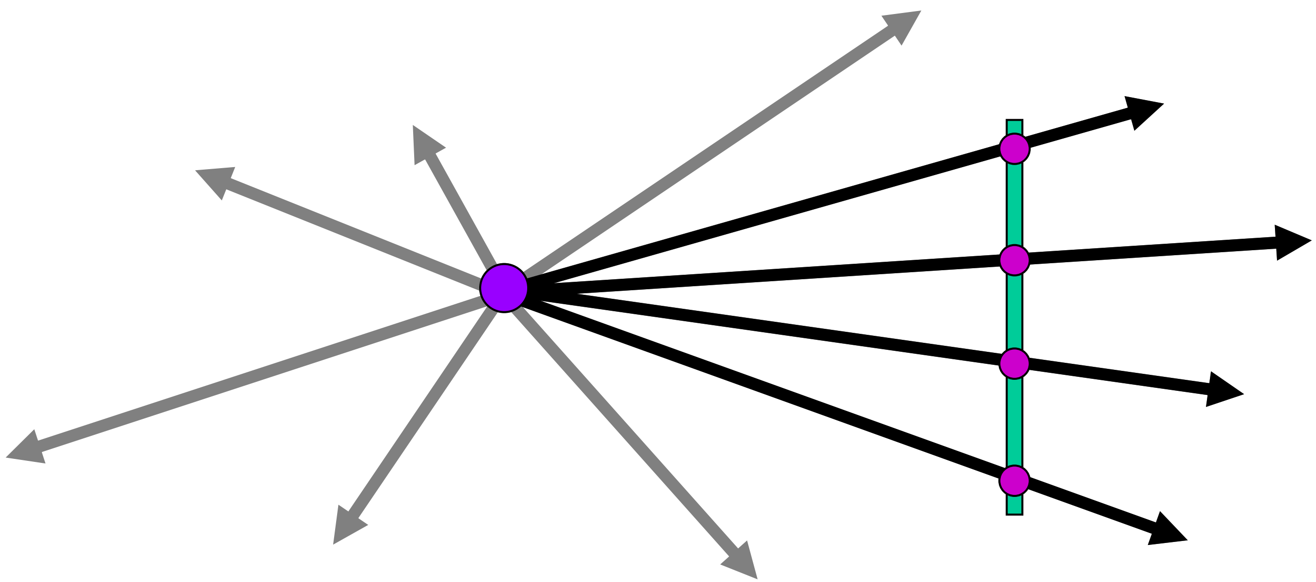
All light rays through a point form a ponorama

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

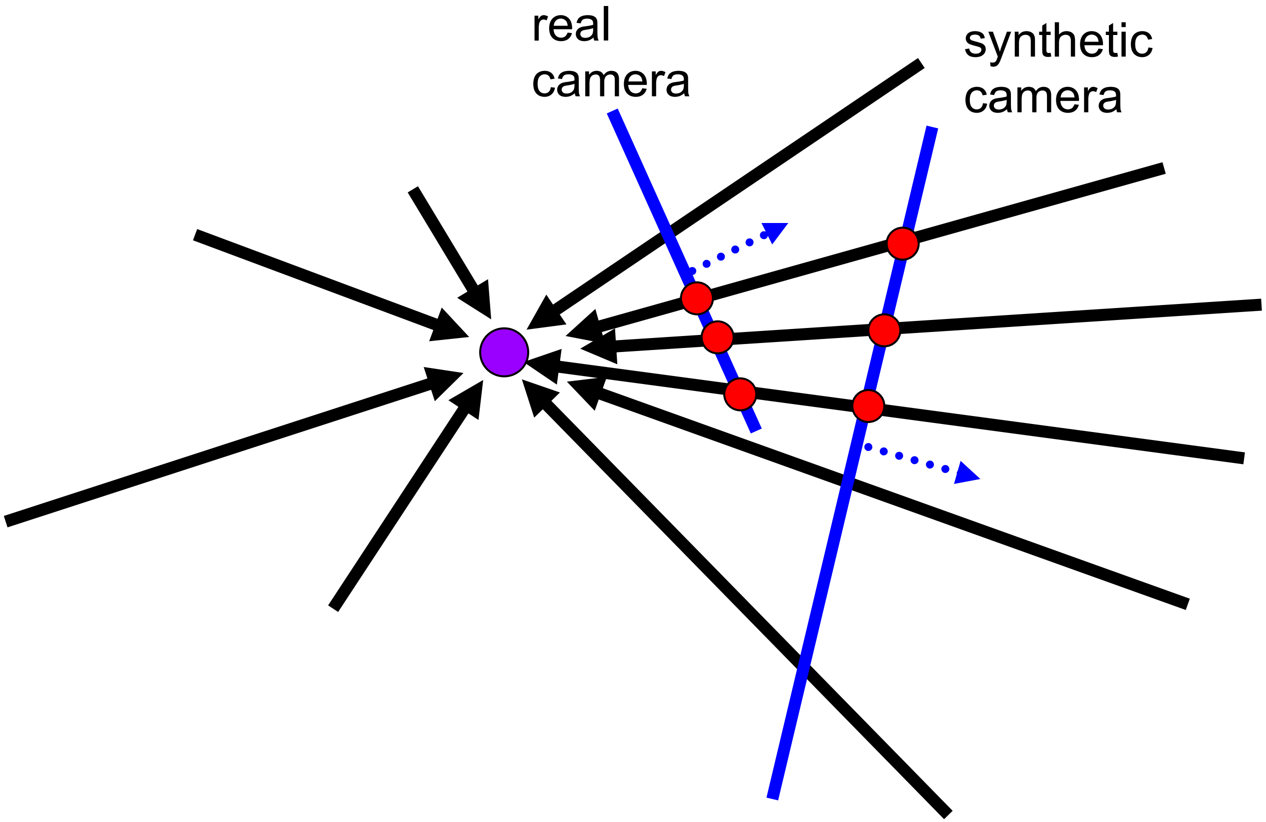
Where is the geometry???

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

What is an Image?



A pencil of rays contains all views

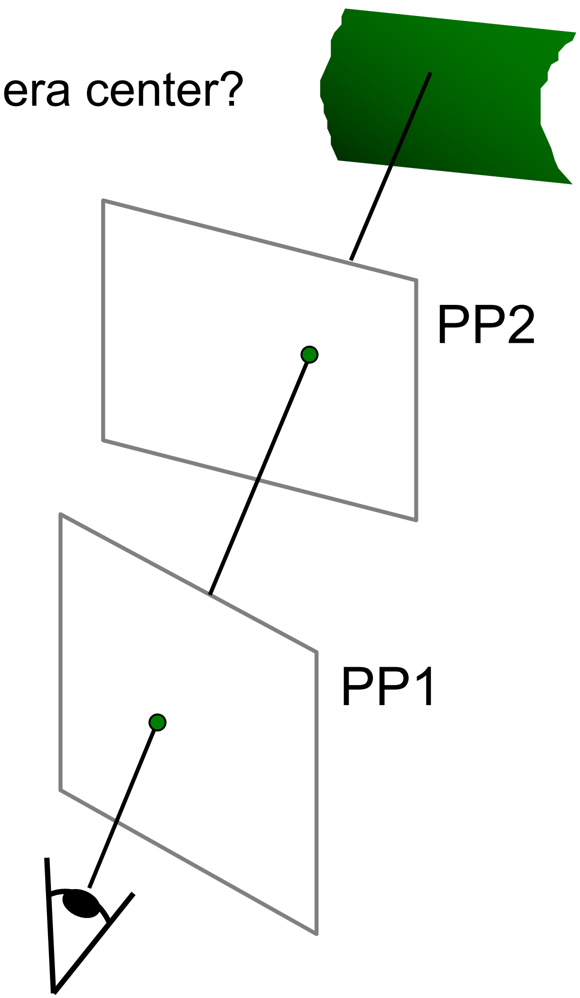


Can generate any synthetic camera view
as long as it has **the same center of projection**!

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

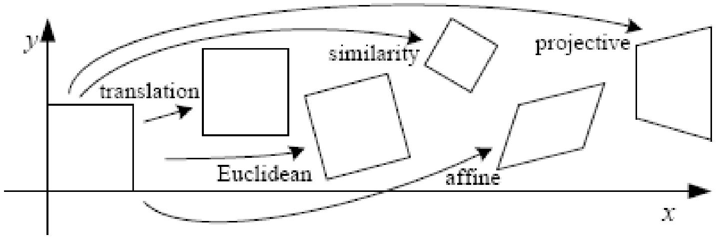
But don't we need to know the geometry of the two planes in respect to the eye?

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

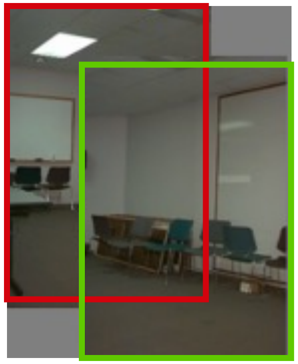
Back to Image Warping

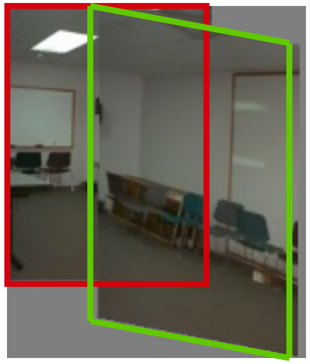
Which t-form is the right one for warping PP1 into PP2?

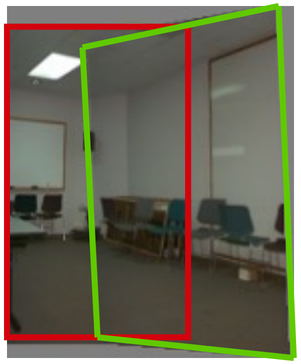
e.g. translation, Euclidean, affine, projective



Translation | **Affine** | **Perspective**







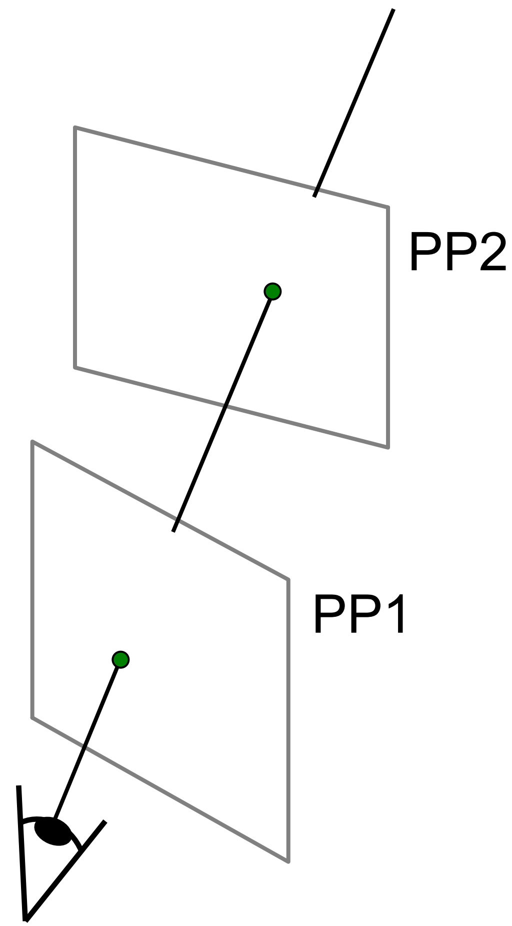
2 unknowns | **6 unknowns** | **8 unknowns**

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



$$\underbrace{\begin{bmatrix} wx' \\ wy' \\ w \end{bmatrix}}_{\mathbf{p'}} = \underbrace{\begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}}_{\mathbf{H}} \underbrace{\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}_{\mathbf{p}}$$

To apply a homography **H**

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

Image warping with homographies

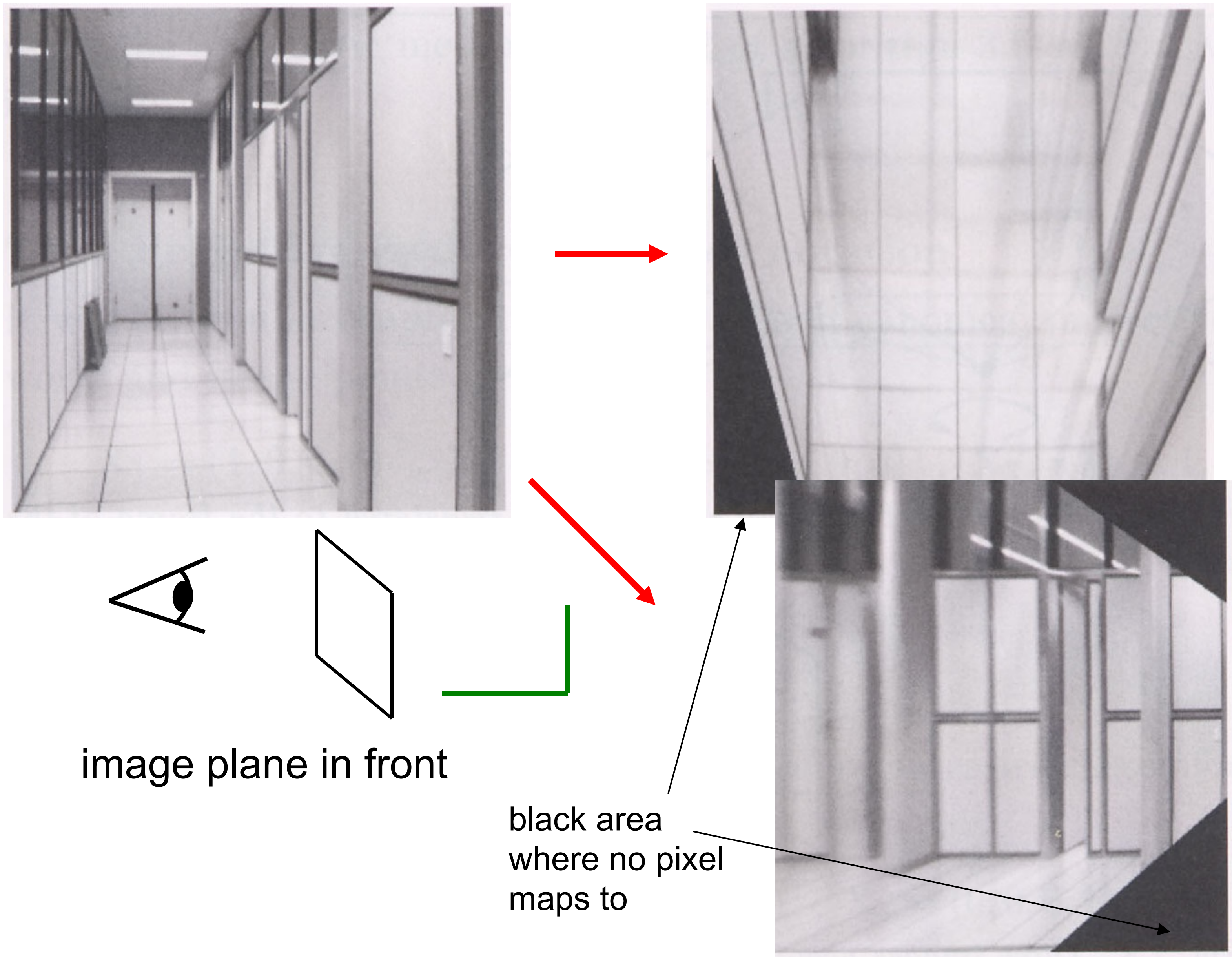
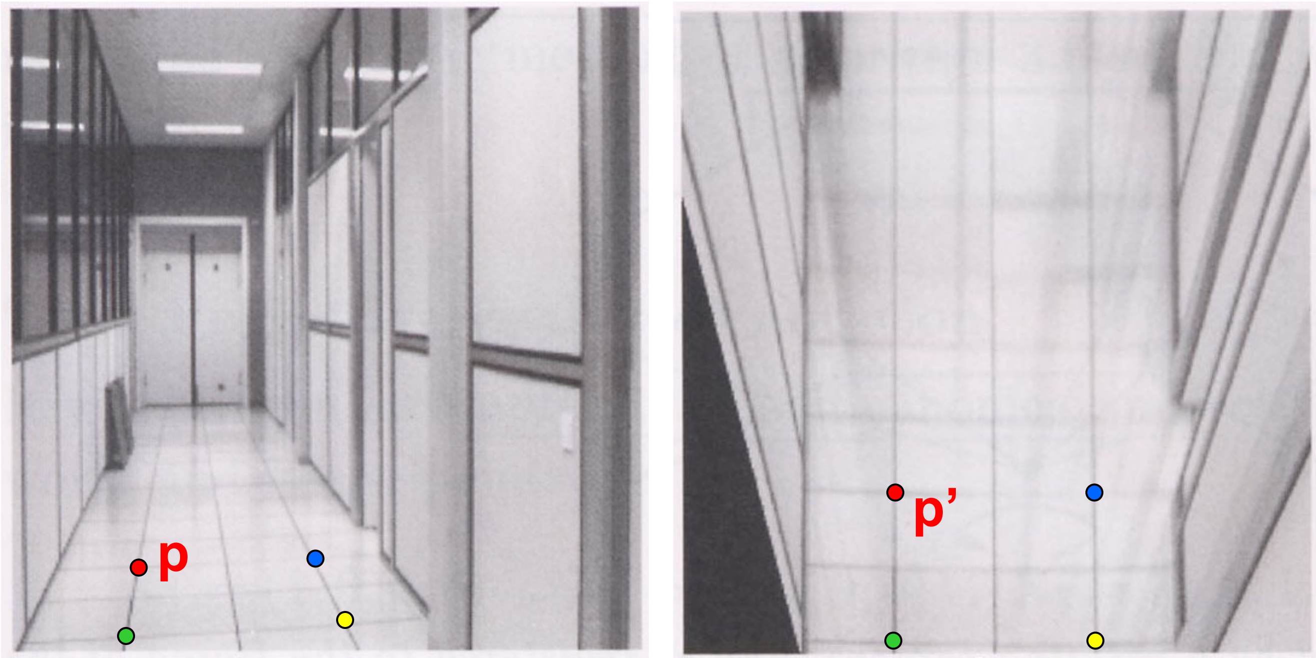


Image rectification



To unwarp (rectify) an image

- Find the homography **H** given a set of **p** and **p’** pairs
- How many correspondences are needed?
- Tricky to write H analytically, but we can <u>solve</u> for it!
    - Find such H that “best” transforms points p into p’
    - Use least-squares!

Least Squares Example

Say we have a set of data points (p1,p1'), (p2,p2'), (p3,p3'), 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?

$$p_1 x_1 + x_2 = p_1^{'}$$
$$p_2 x_1 + x_2 = p_2^{'}$$

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

$$\mathrm{Ax} = b$$

Least Squares Example

Say we have a set of data points (p1,p1'), (p2,p2'), (p3,p3'), 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?

$$p_1 x_1 + x_2 = p_1'$$
$$p_2 x_1 + x_2 = p_2'$$

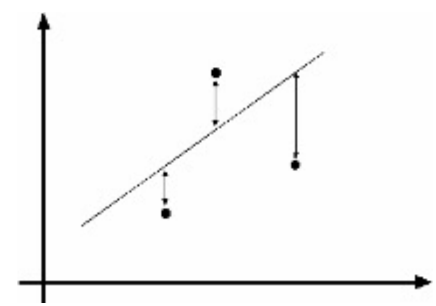
$$\begin{bmatrix} p_1 & 1 \\ p_2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} p_1' \\ p_2' \end{bmatrix}$$

$$\mathrm{Ax} = b$$

What if the data is noisy?

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

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



overconstrained

Least-Squares

- Solve:

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

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

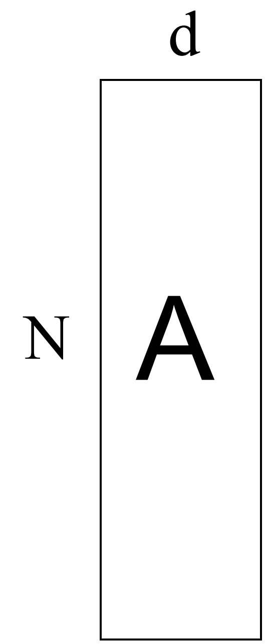
- Normal equations

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

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

- Solution:

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



rank(A)≤min(d,N)
assume rank(A)=d
implies rank(A^TA)=d
A^TA is invertible

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:

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

Can be done in Matlab using "\" command

- see "help lmdivide"

Fun with homographies

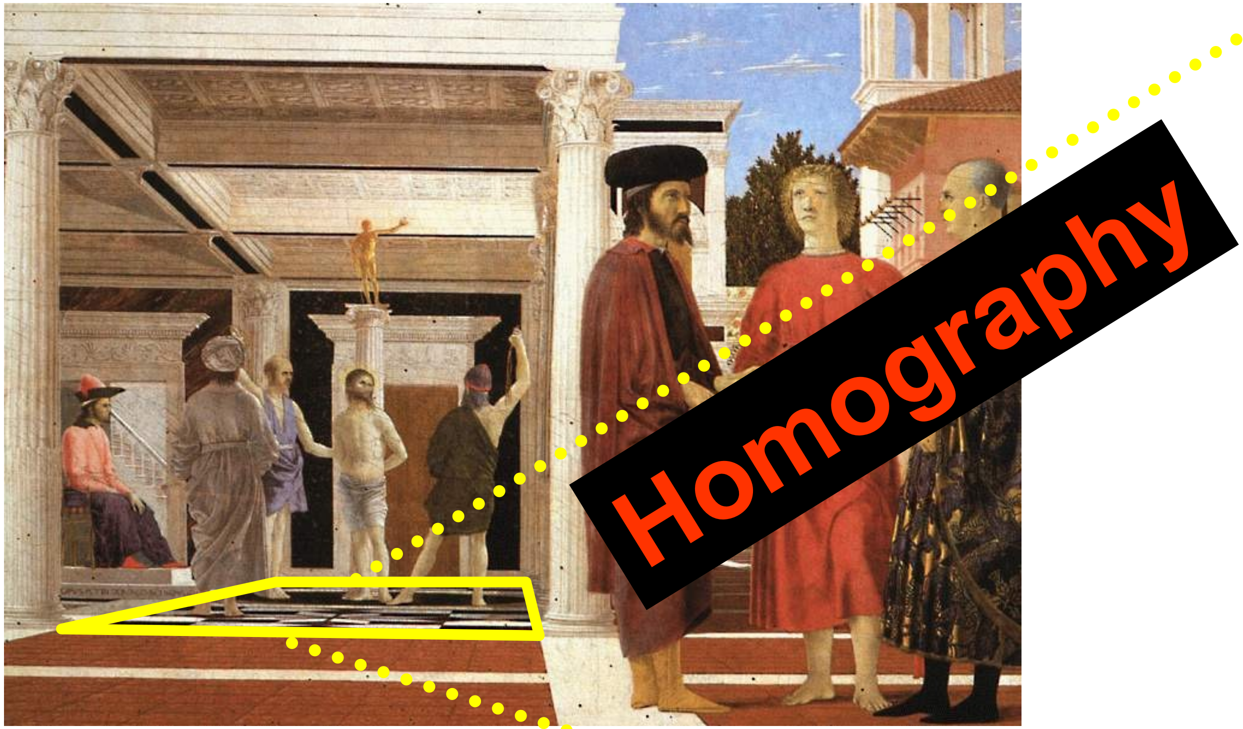
Original image

St.Petersburg
photo by A. Tikhonov

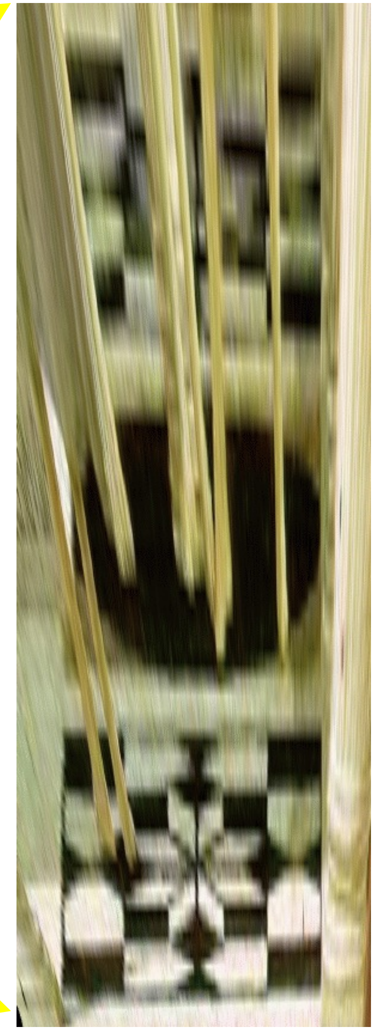
Virtual camera rotations

Analysing patterns and shapes

What is the shape of the b/w floor pattern?



The floor (enlarged)

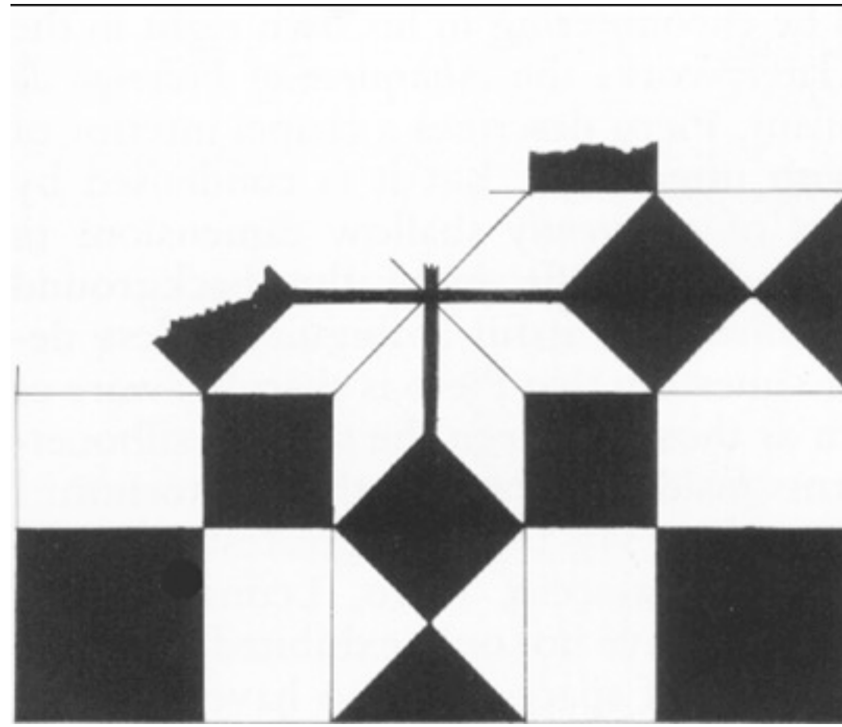


Automatically rectified floor

Slide from Criminisi

Analysing patterns and shapes

Automatic rectification



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

2 patterns have been discovered !

Slide from Criminisi

Analysing patterns and shapes

St. Lucy Altarpiece, **D. Veneziano**

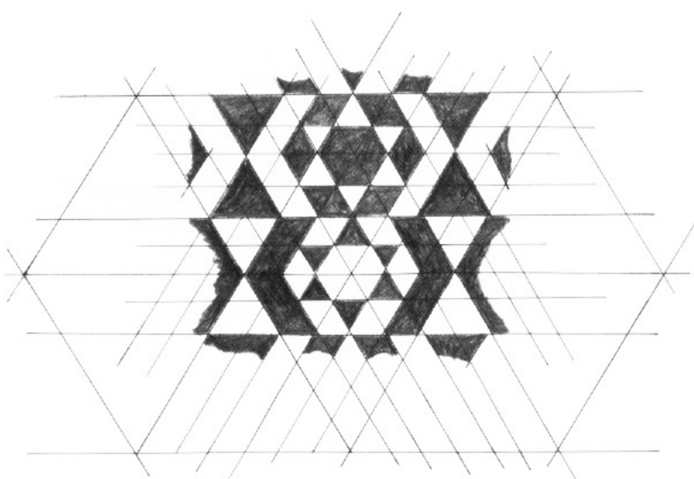
What is the (complicated) shape of the floor pattern?

Automatically rectified floor

Slide from Criminisi

Analysing patterns and shapes

Automatic rectification



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

Slide from Criminisi

Mosaics: stitching images together

virtual wide-angle camera

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

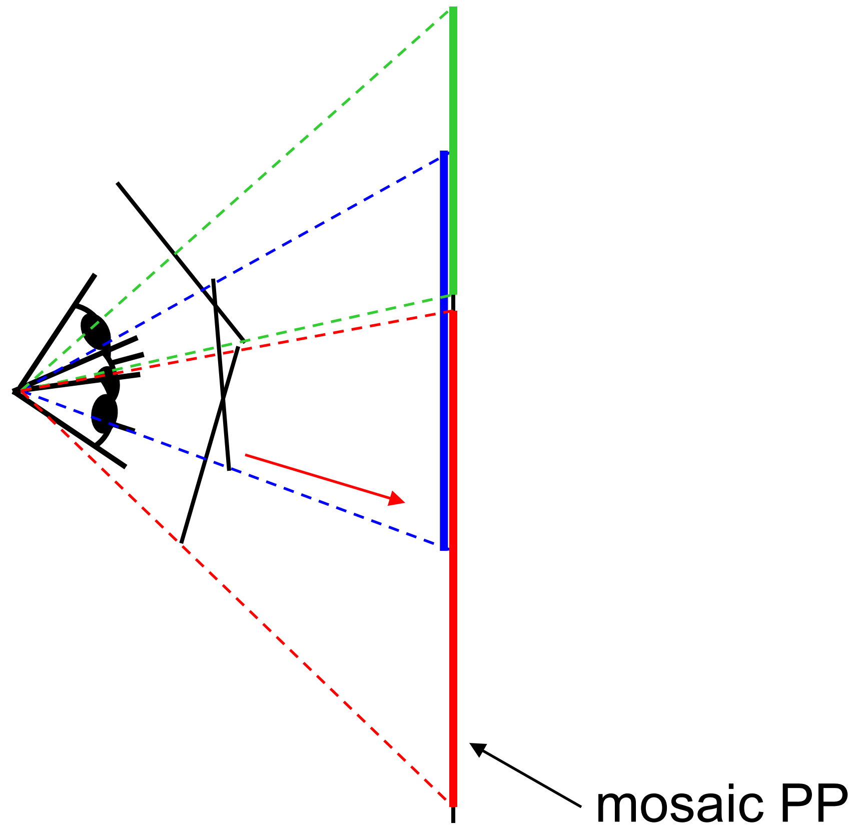
Naïve Stitching

left on top

right on top

Translations are not enough to align the images

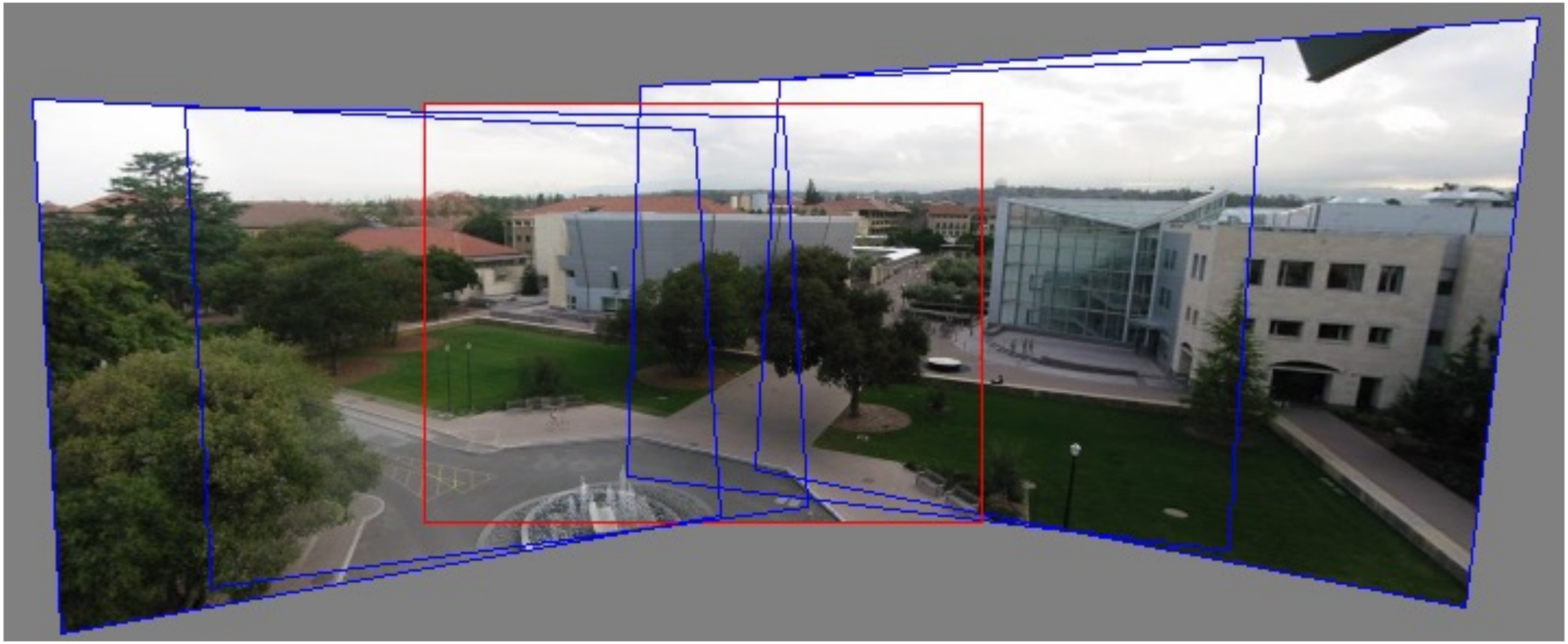
Image reprojection



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*

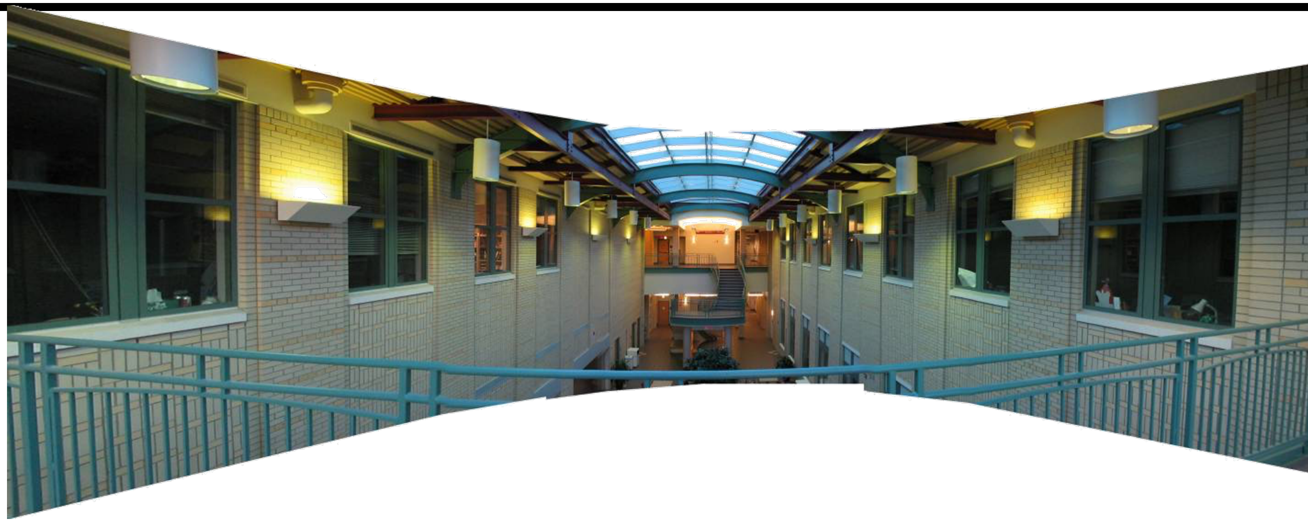
Panoramas



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

Holbein, *The Ambassadors*

Programming Project #4 (part 1)



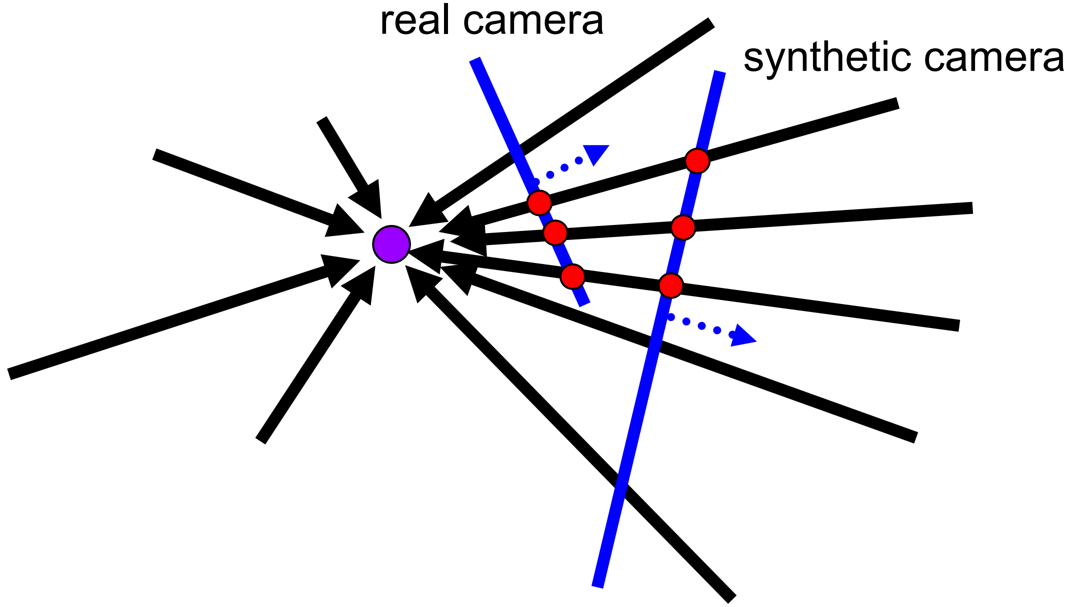
Homographies and Panoramic Mosaics

- 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)