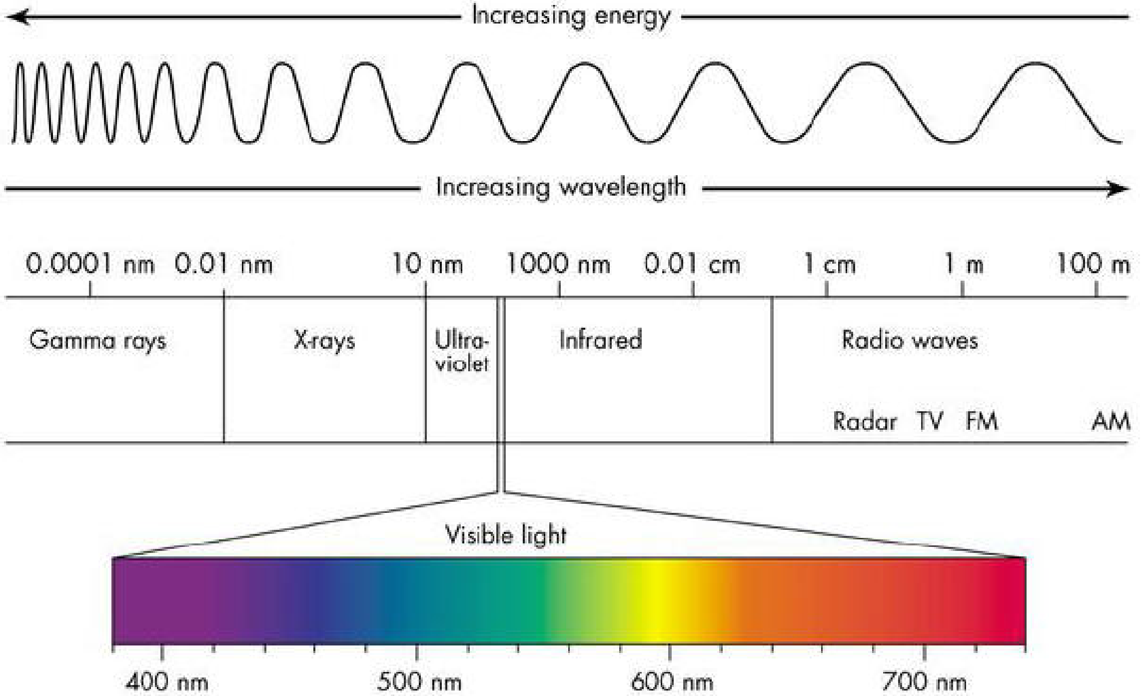
Lights, Cameras, Calibration

Slides adapted from J. Hays, R. Pless and G. Hager

What is Light?

- We almost never see a “pure” wavelength of light
  - Rather a mixture of wavelengths, each with a different “power”
  - Only some colors occur as pure wavelengths; many are mixtures of pure colors (e.g. white)
  - Humans can detect light in the spectrum of 400 nm (violet) to 700 nm (red)



Light

- Light is **generated** by some source
  - point source
  - extended source
  - white/colored ...
- Light is **reflected** from some surface
  - matte,
  - mirrorlike
  - colored/light/dark ....
- Light is **sensed** by some instrument
  - sensitivity
  - field of view
  - gray scale/color/....

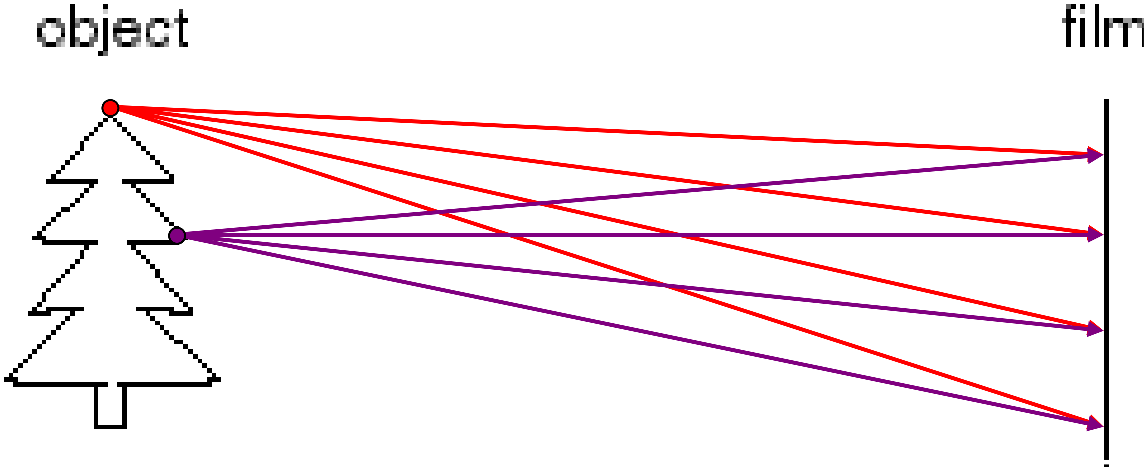
Light Sources

- What is A Source?
  - Anything that generates light
  - Characterized by *spectrum* & *power*

- Common sources:
  - "point source at inifinity"
  - a completely uniform illuminant
  - Both not found in reality

- Reflectance Models
  - Important topic in graphics.
  - Given a light source and an object (and its properties), determine the pattern of radiation.
  - In this course, we're mainly concerned with the light *arriving at the camera.*

Image formation

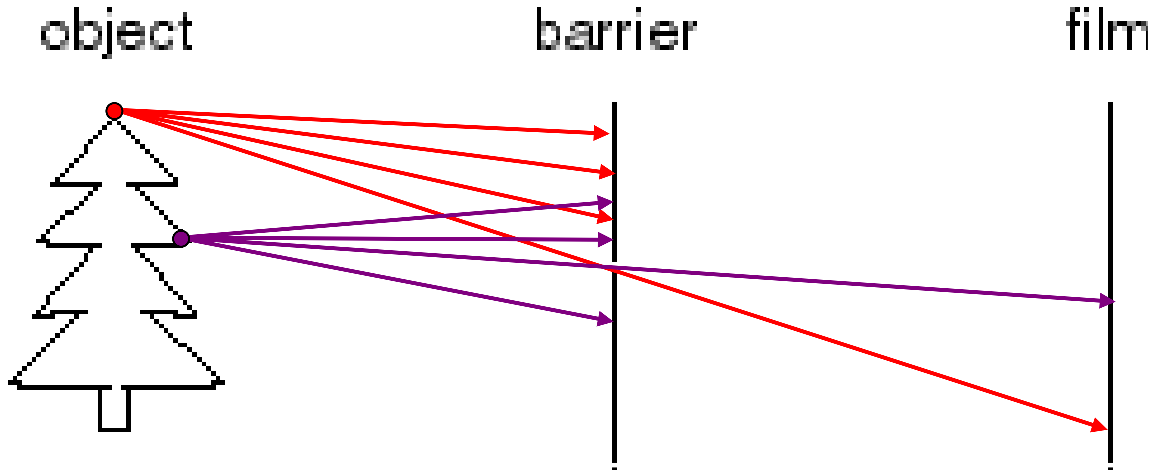


Let's design a camera

- Idea 1: put a piece of film in front of an object
- Do we get a reasonable image?

Slide source: Seitz

Pinhole camera



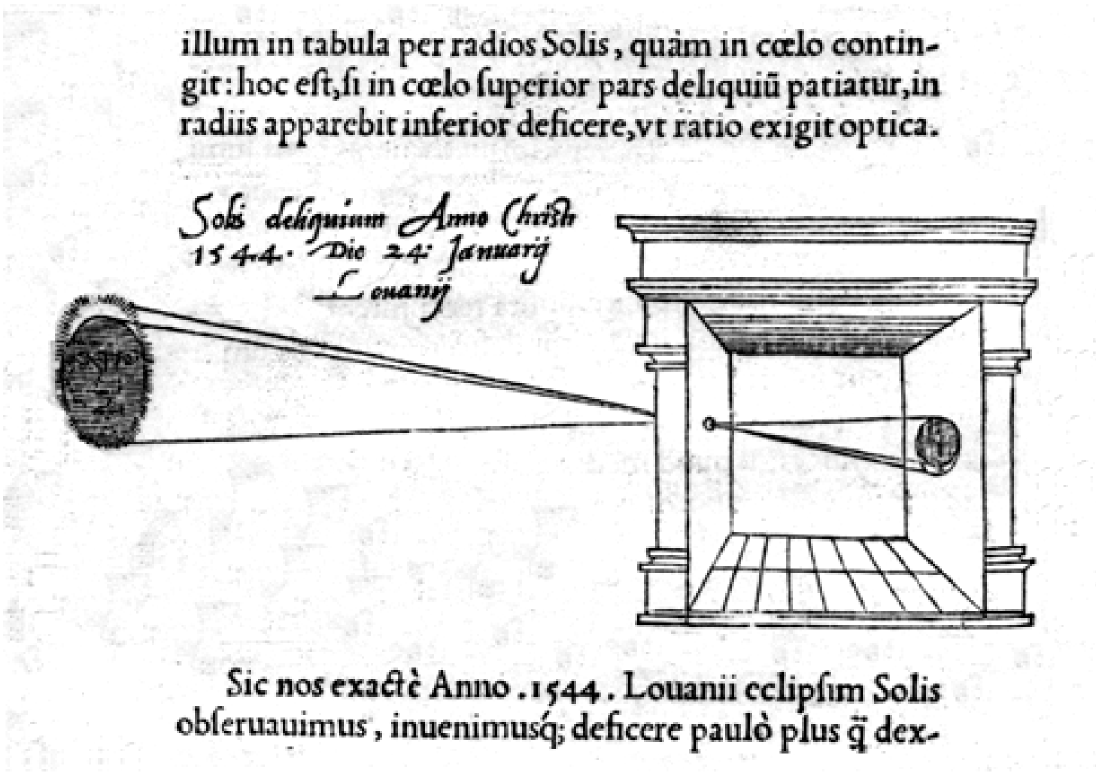
Idea 2: add a barrier to block off most of the rays

- This reduces blurring
- The opening known as the **aperture**

Slide source: Seitz

Basic Camera Model

- Camera Obscura
  - Camera - Latin for "room"
  - Obscura - Latin for "dark“
- "When images of illuminated objects ... penetrate through a small hole into a very dark room ... you will see [on the opposite wall] these objects in their proper form and color, reduced in size ... in a reversed position, owing to the intersection of the rays". *Da Vinci*

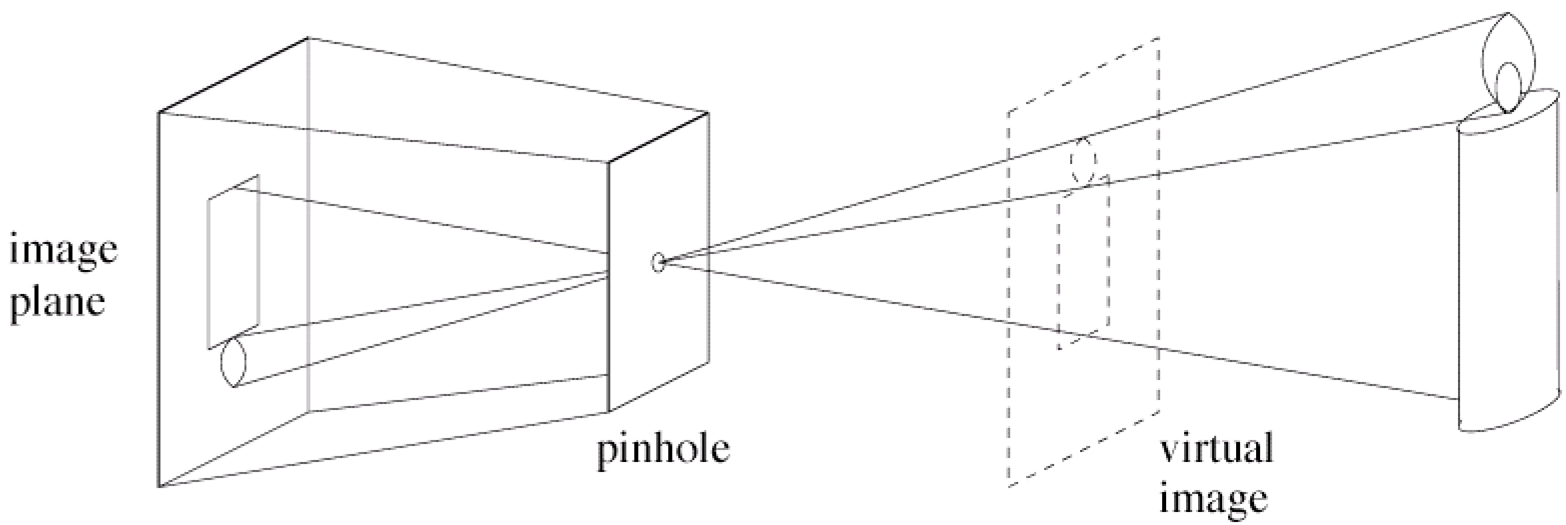

illum in tabula per radios Solis, quàm in cœlo contin-
git: hoc eſt, ſi in cœlo ſuperior pars deliquiū patiatur, in
radiis apparebit inferior deficere, vt ratio exigit optica.

Solis deliquium Anno Christi
1544. Die 24 Januarij
Louanij

Sic nos exactè Anno .1544. Louanii eclipſim Solis
obſeruauimus, inuenimusq; deficere paulò plus q̃ dex-

Our Model: Pinhole Camera

- Abstract camera model
  - Box w/ a small hole in it
- Pinhole cameras actually work in practice



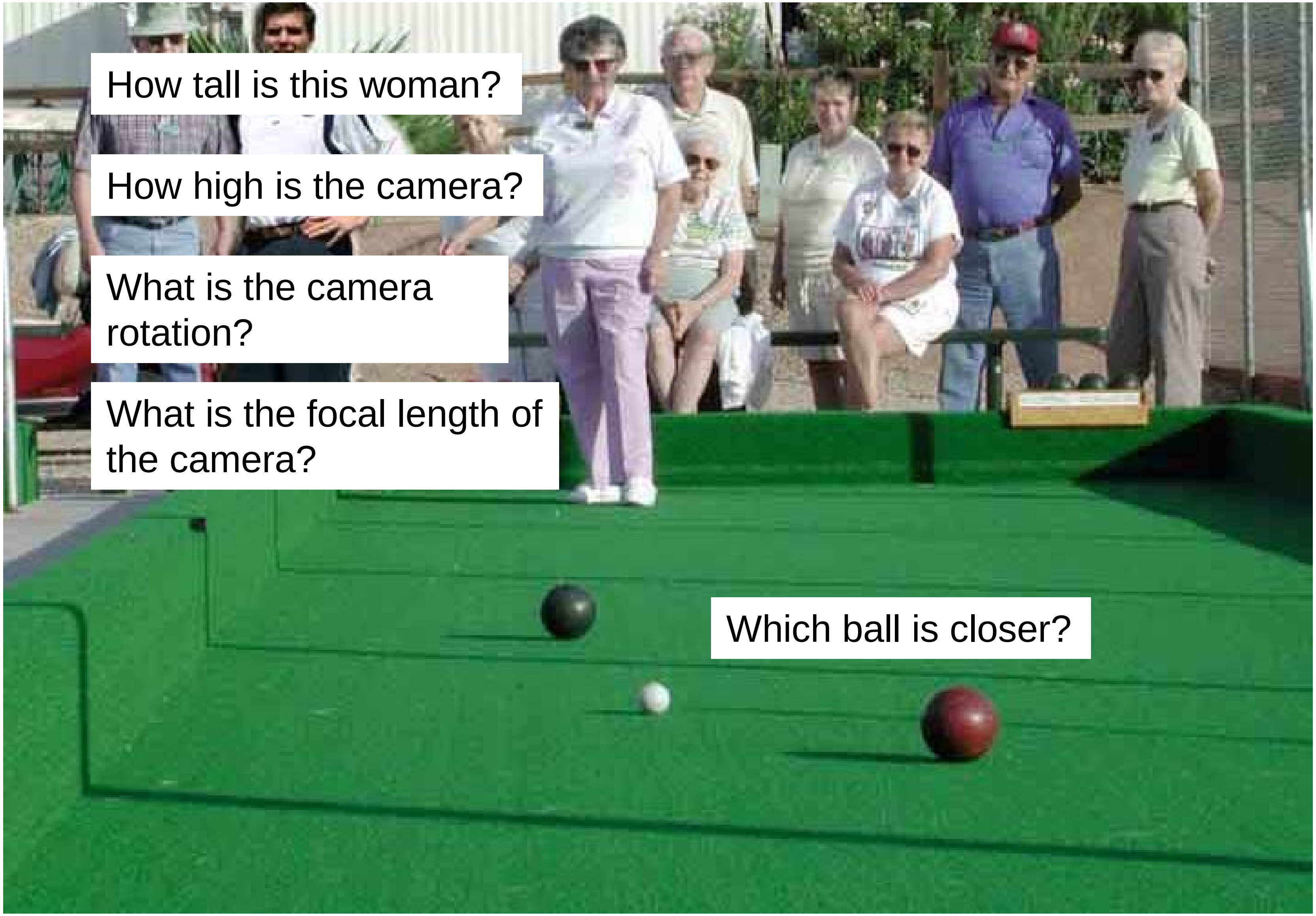
Slide source: Seitz

Projection can be tricky…

Slide source: Seitz

Projection can be tricky…

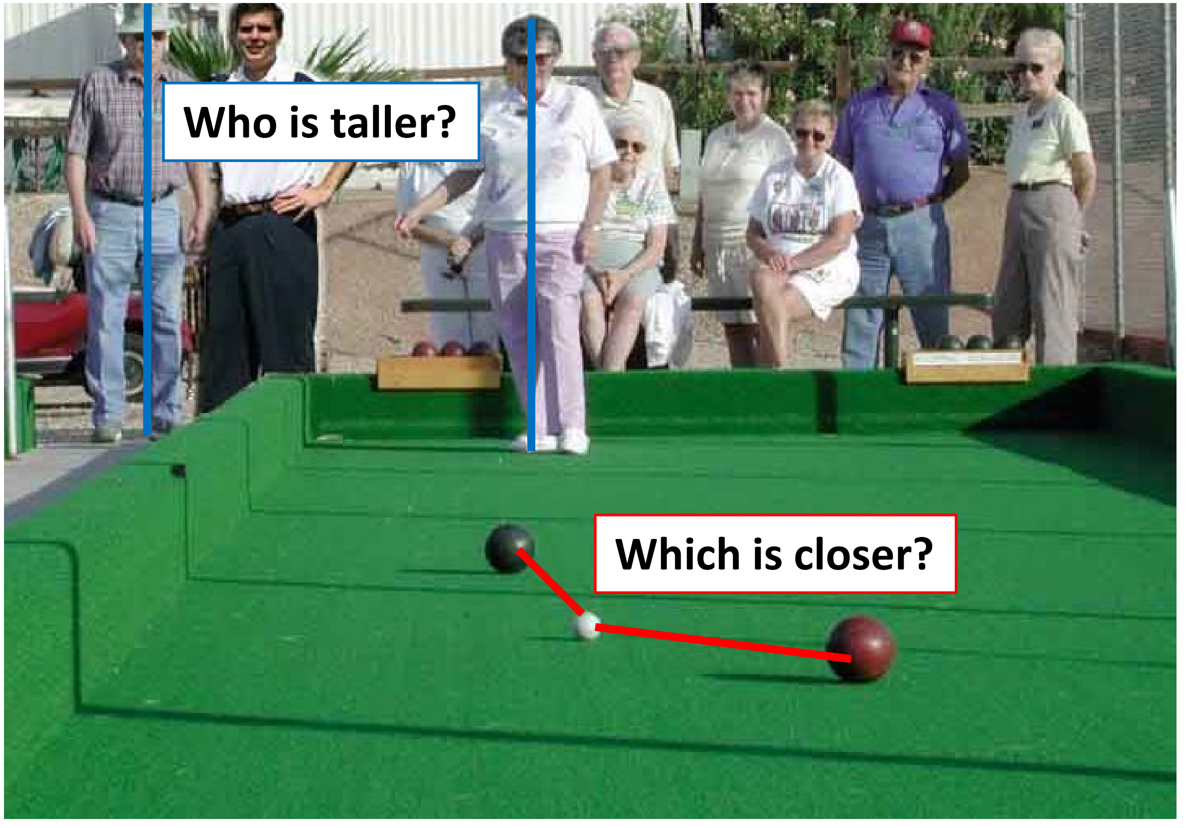
Computer Vision Questions?



Projective Geometry

What is lost?

- Length



Length is not preserved

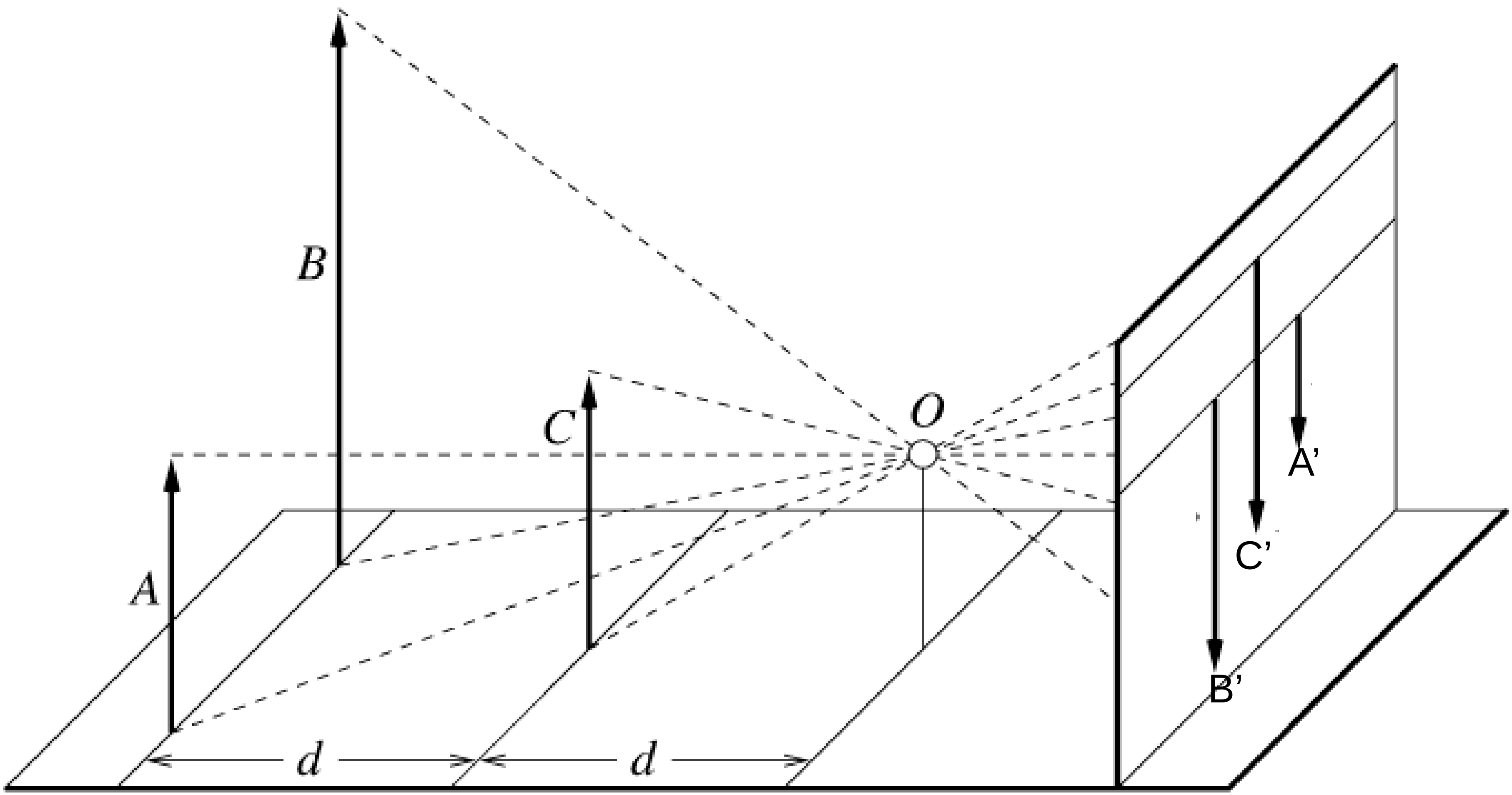
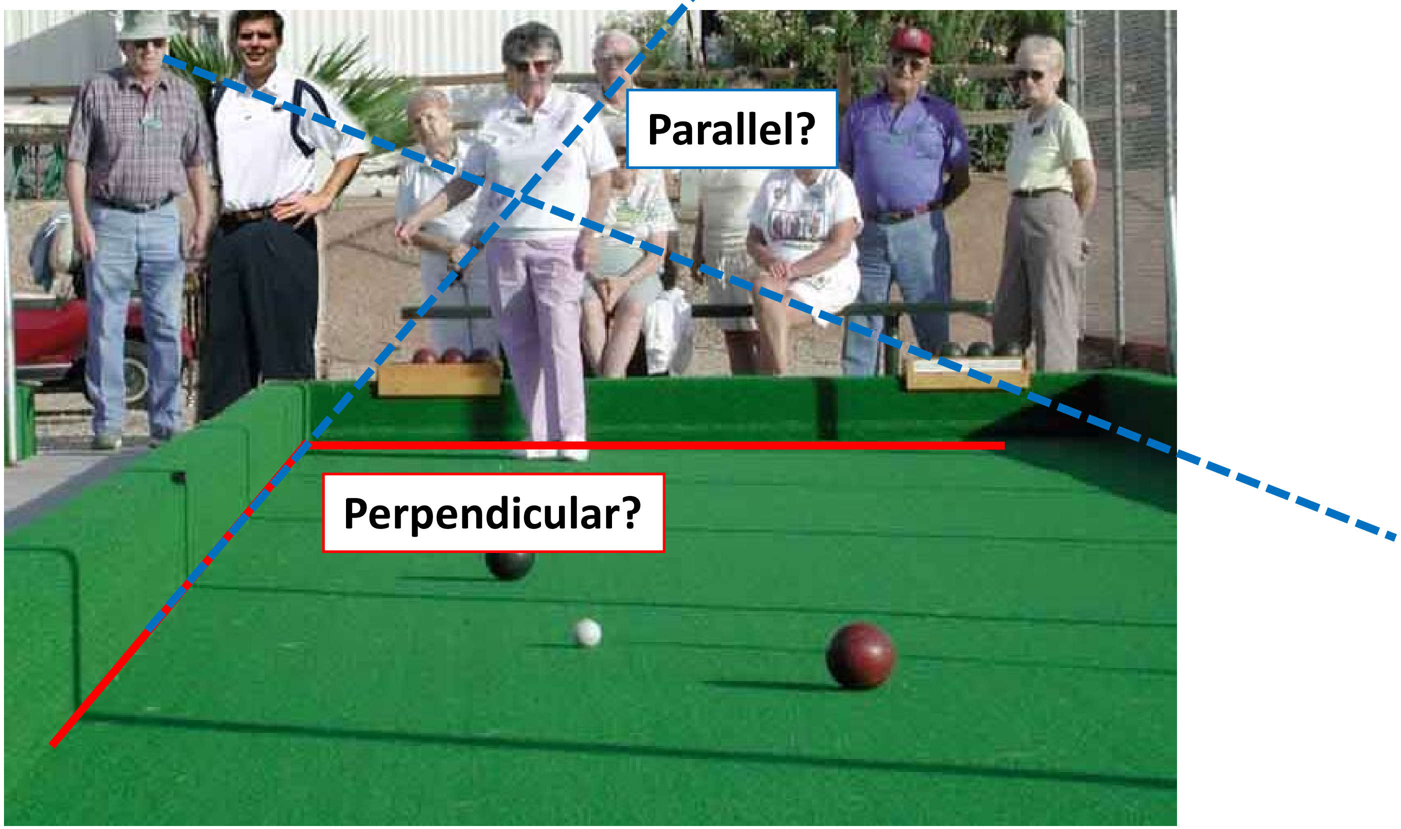


Figure by David Forsyth

Projective Geometry

What is lost?

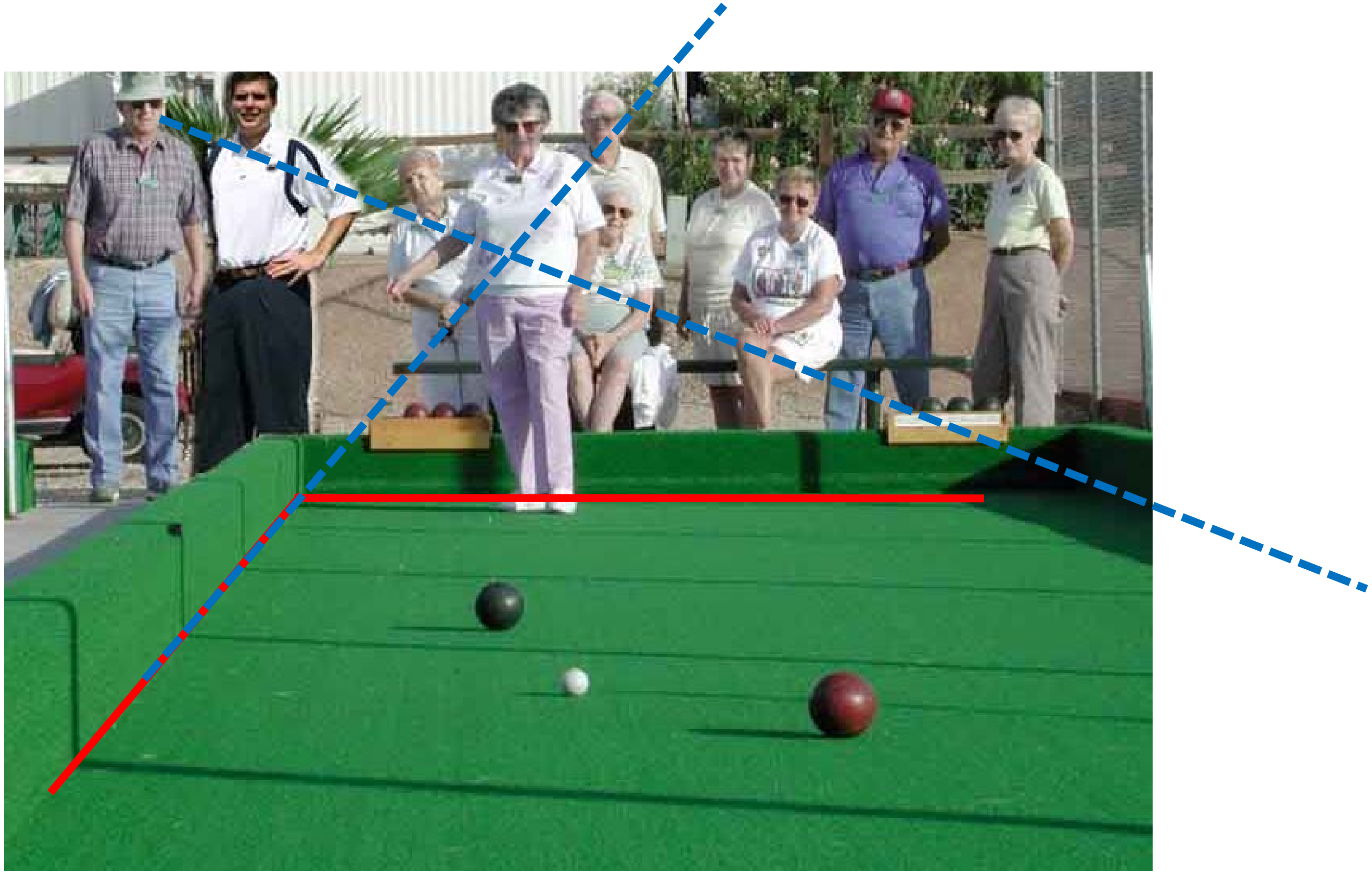
- Length
- Angles



Projective Geometry

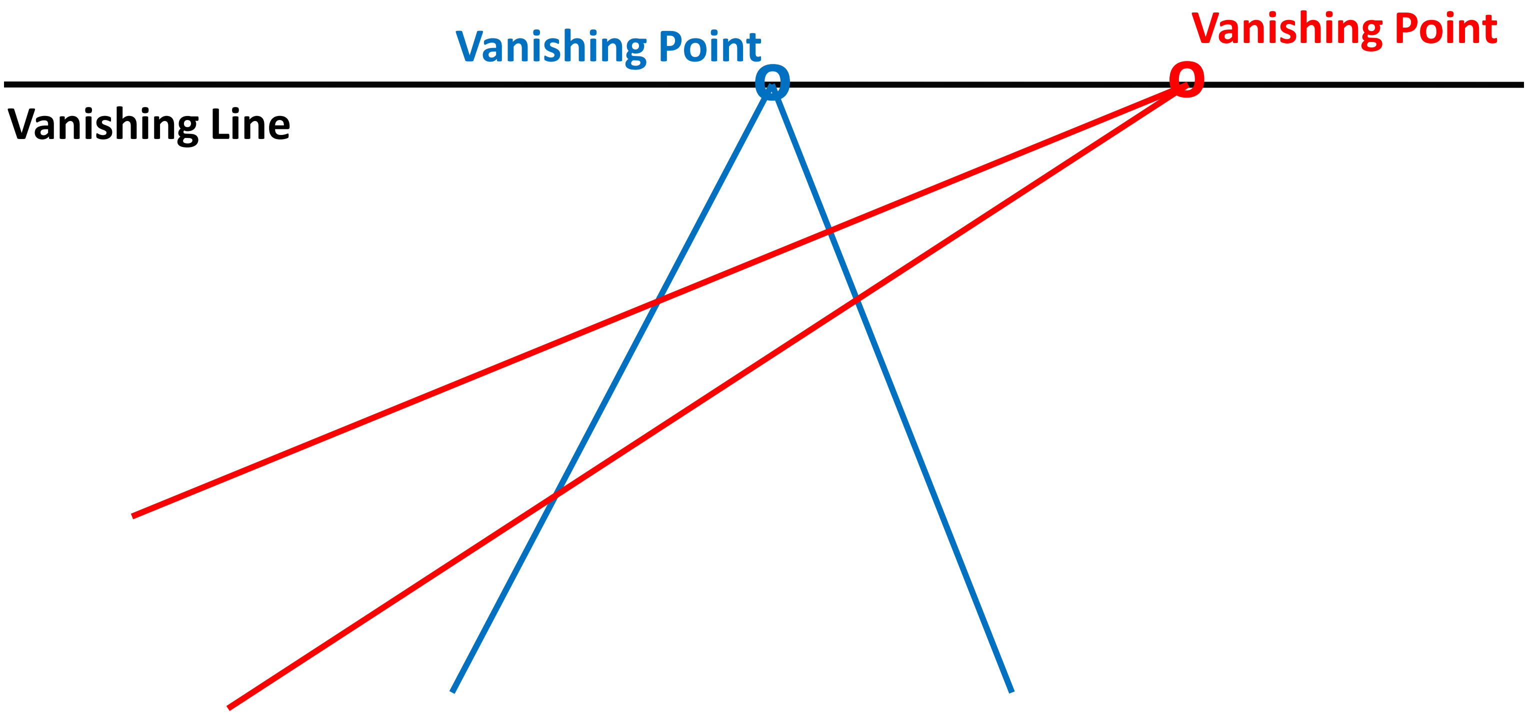
What is preserved?

- Straight lines are still straight

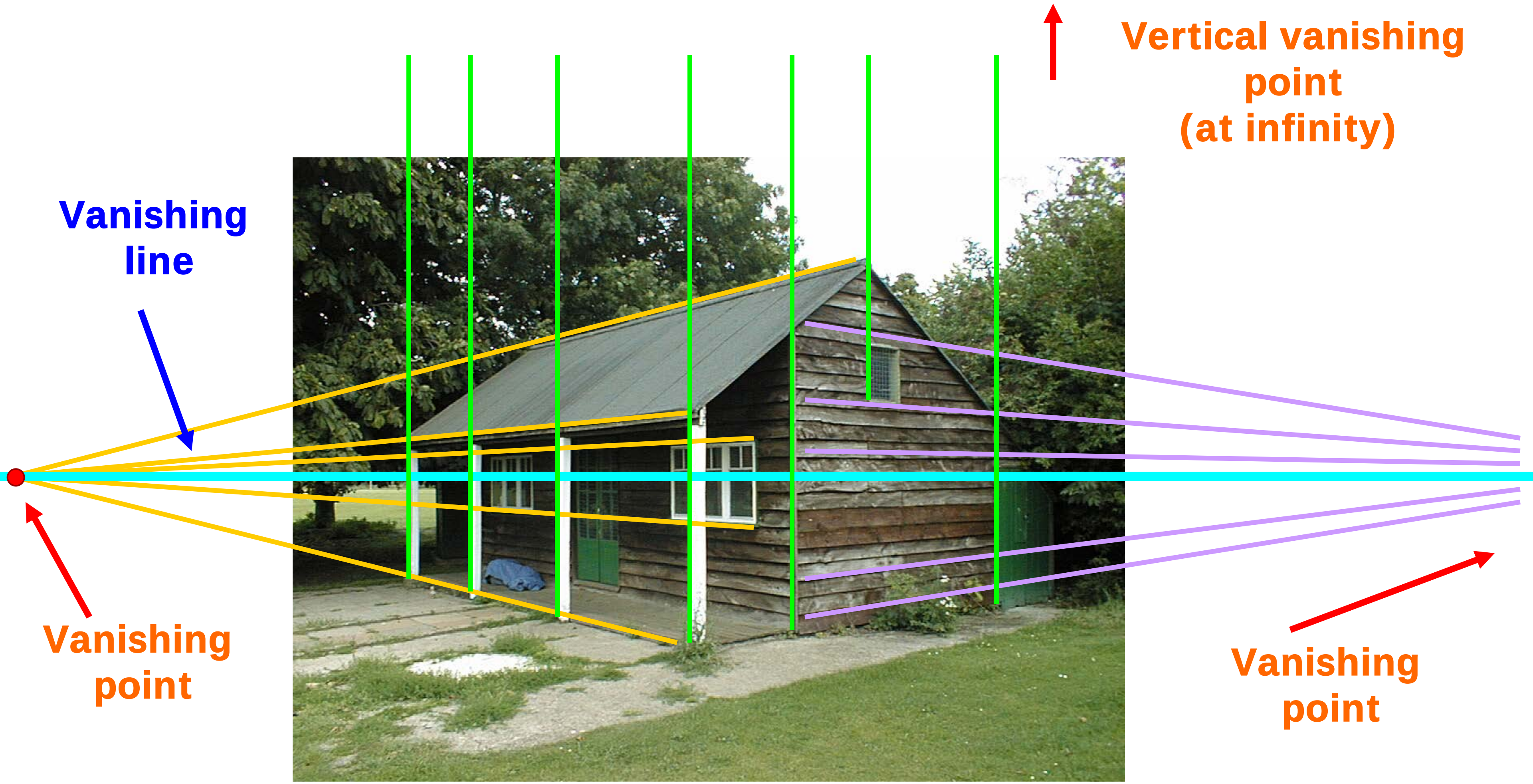


Vanishing points and lines

- Parallel lines in the world intersect in the image at a "vanishing point"

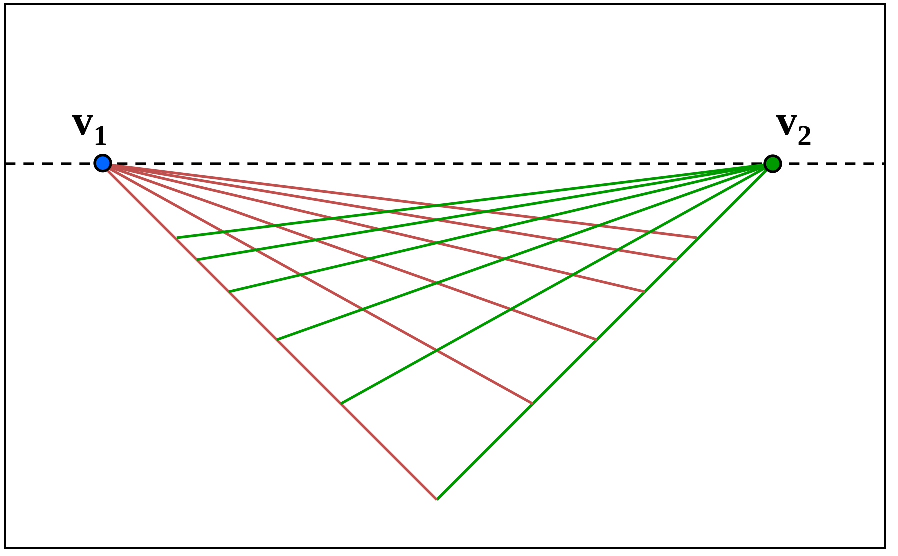


Vanishing points and lines



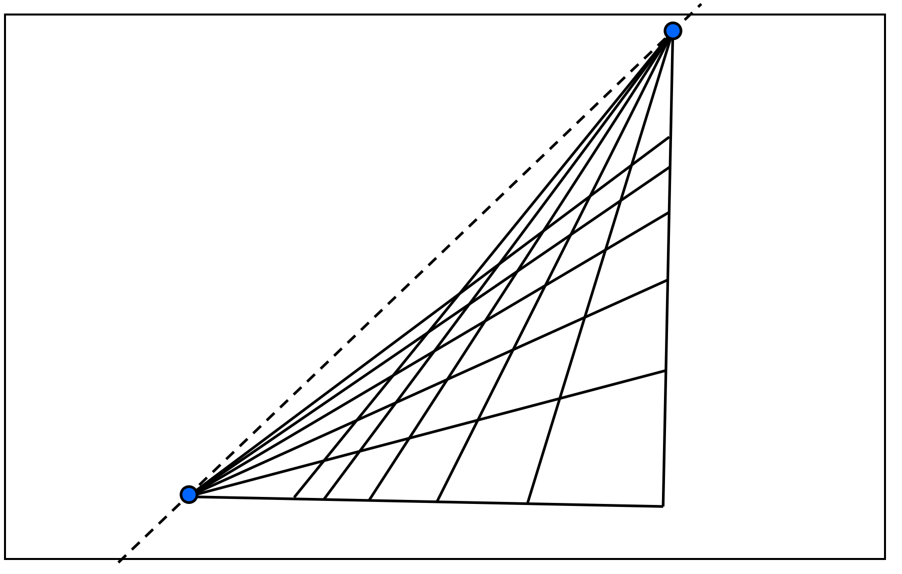
Slide from Efros, Photo from Criminisi

Vanishing lines



- Multiple Vanishing Points
  - Any set of parallel lines on the plane define a vanishing point
  - The union of all of vanishing points from lines on the same plane is the *vanishing line*
    - For the ground plane, this is called the *horizon*

Vanishing lines



- Multiple Vanishing Points
  - Different (sets of) planes define different vanishing lines

Properties of Pinhole Projection

- Invariants (Things that are always true)
  - Points project to points
  - Lines project to lines
  - Planes project to the whole image or half-image
- Degenerate Cases?
  - Line through the focal point → point
  - Plane through focal point → line
- Angles are not preserved
- Parallel lines may intersect

Coordinate Systems

- Lots of coordinate systems to keep track of
  - W (world), O (object)
  - C (camera), I (image)

- If we assume the object is given in W coords, we have 2 transformations to learn:
  - How C & I are related (intrinsic)
  - How C & W are related (extrinsic)
  - This is called *camera calibration*

Homogeneous Coordinates

- Trick. Represent 2D point using 3 numbers.

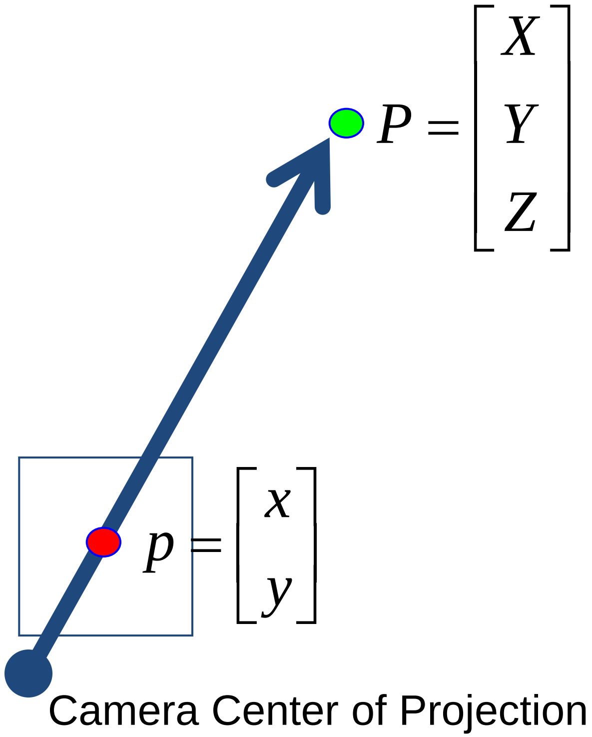
- Converting *to* homogeneous coordinates:
  - (x,y) → (x,y,1)
  - (x,y) → (sx, sy, s)
  - (x,y,z) → (x,y,z,1) *or* (sx,sy,sz,s)

- Mapping homogeneous to Cartesian:
  - (a,b,w) → (a/w, b/w)

- The coordinates of a 3D point (X,Y,Z) ****already are**** the 2D (but homogeneous) coordinates of its projection.
- Makes some things simpler and “linear-er”

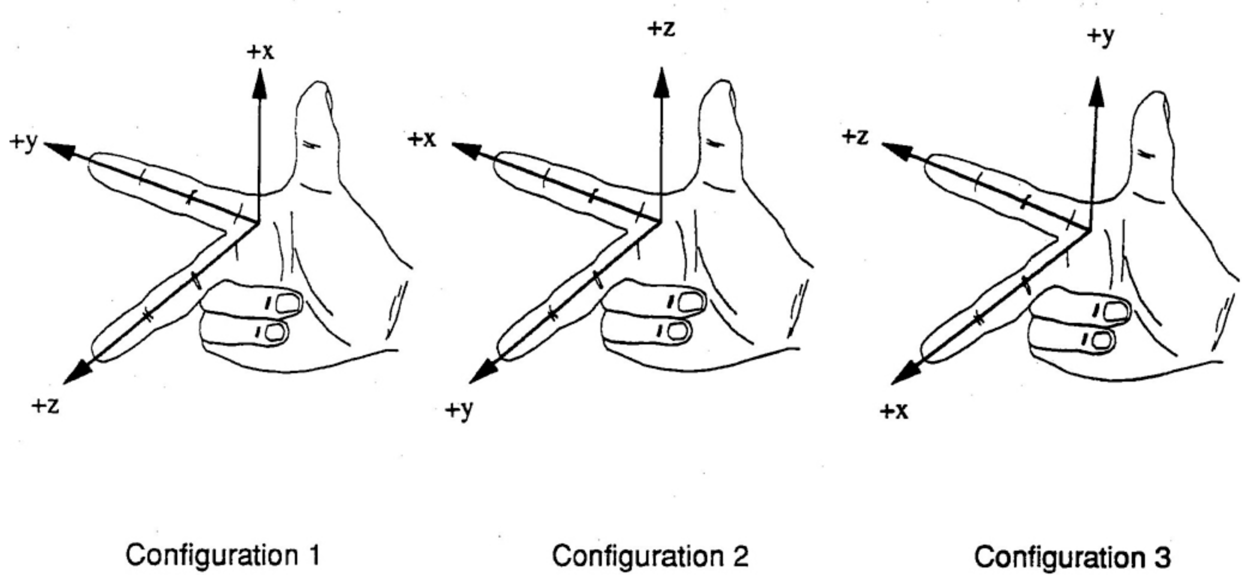
Projection: Normalized Camera

- Pinhole at (0,0,0)
- Virtual film at z = 1 plane
  - *In "front" of camera center*
- Point X,Y,Z is imaged at intersection of:
  - Line from (0,0,0) to (X,Y,Z), and
  - the Z = 1 plane
- So intersection point has coordinates (X/Z, Y/Z, 1)



“Standard” Camera Coords

- The image is in front of the optical center
- Optical axis (viewing direction) is z-axis pointing outward
- X-axis is parallel to the scanlines pointing to the right
- By the right hand rule, the y-axis points downward
  - This corresponds with Matlab indexing



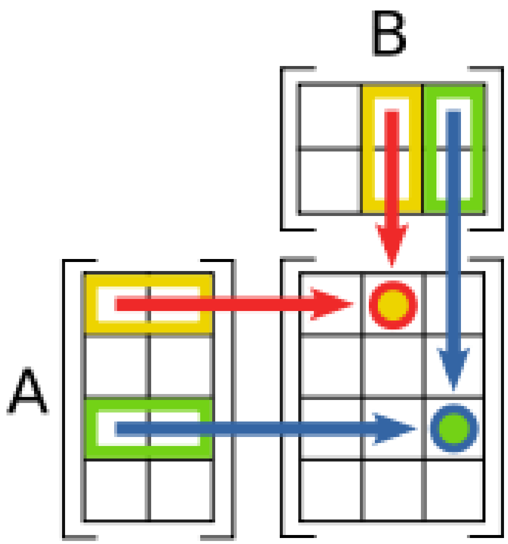
Real Cameras

- Camera may not be at the origin, looking down the z-axis
  - extrinsic parameters
- One unit in camera coordinates may not be the same as one unit in world coordinates
  - intrinsic parameters

$$\begin{pmatrix} U \\ V \\ W \end{pmatrix} = \begin{pmatrix} \text{Transformation} \\ \text{representing} \\ \text{intrinsic parameters} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \text{Transformation} \\ \text{representing} \\ \text{extrinsic parameters} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ T \end{pmatrix}$$

Matrix Multiplication Recap

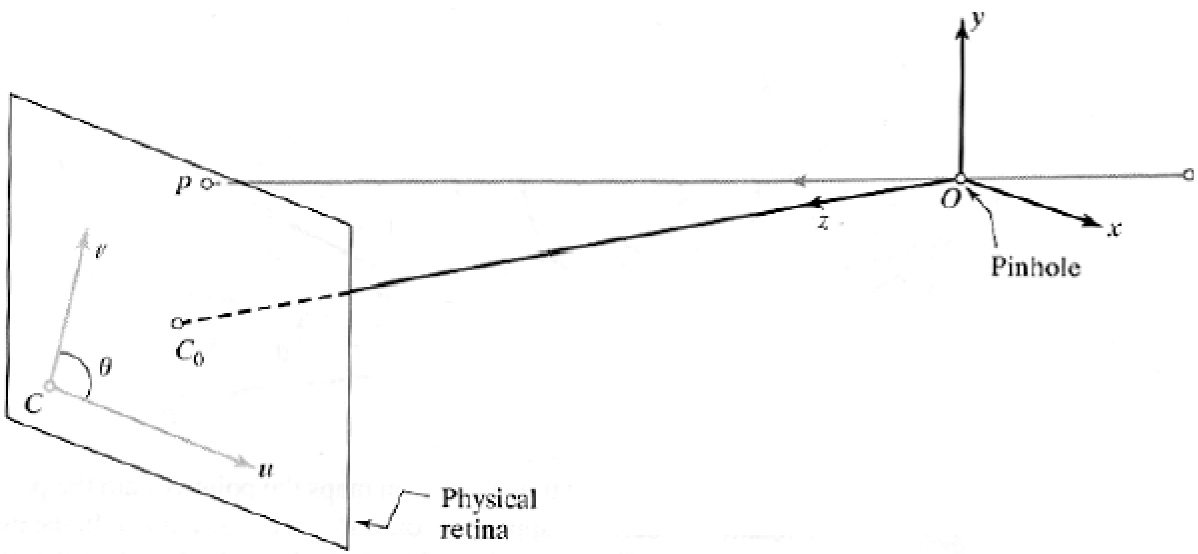
- If we want to multiply matrices A & B
  - A is m*n, B is n*p
  - Result is m*p

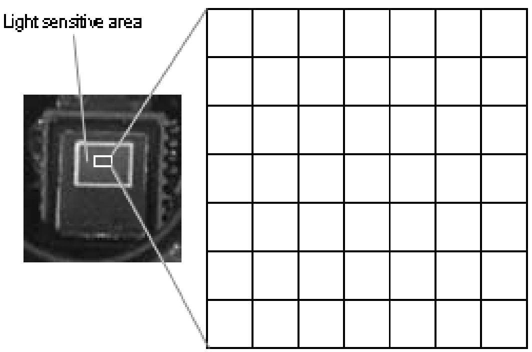


$$(AB)_{1,2} = \sum_{r=1}^{2} a_{1,r} b_{r,2} = a_{1,1} b_{1,2} + a_{1,2} b_{2,2}$$

$$(AB)_{3,3} = \sum_{r=1}^{2} a_{3,r} b_{r,3} = a_{3,1} b_{1,3} + a_{3,2} b_{2,3}$$

Measure in “pixel widths”





The units of measuring X,Y,Z don’t matter (because we are going to divide them anyway), so lets make the unit be the size of the pixel on the ccd (the photon detector) chip

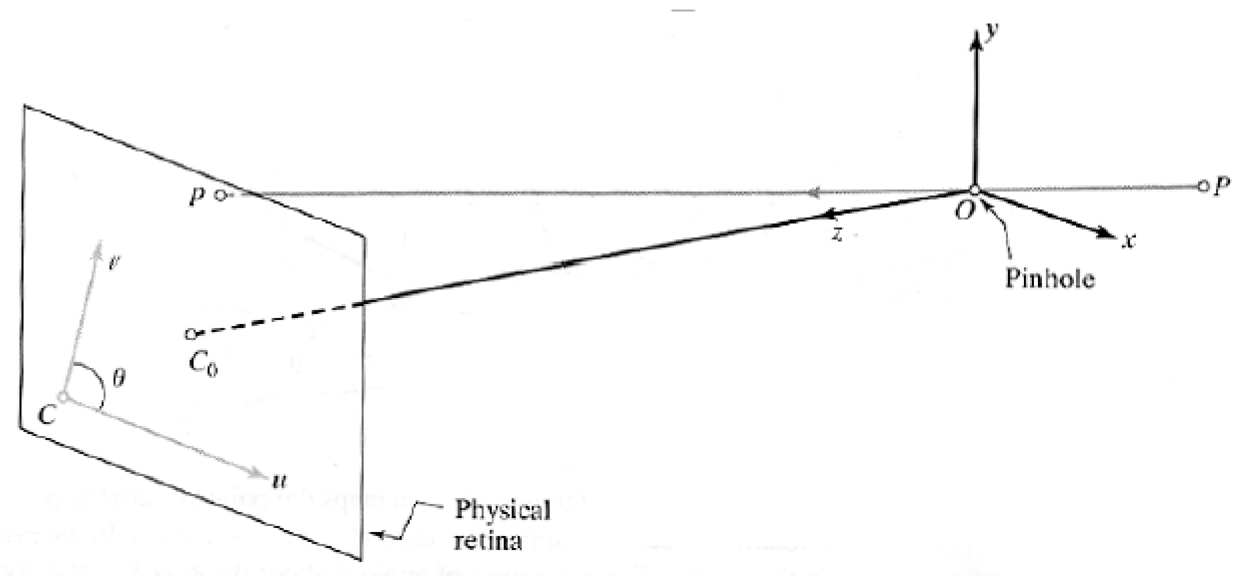
(5 mm /1000 pixels) = 5 μm / pixel = 1 “unit”

Express the focal length in these “units”, so:

x = fX/Z

Y = fY/Z

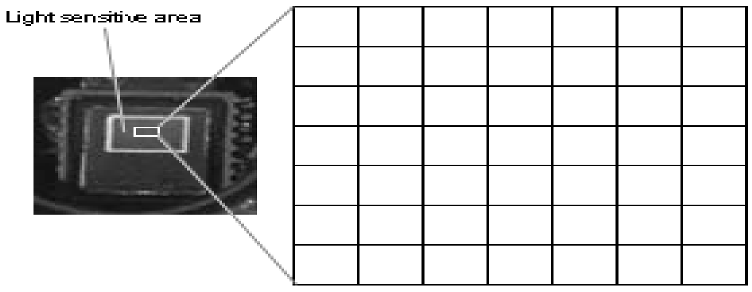
Intrinsic Parameters: Non-square pixels



The pixels may be rectangular:

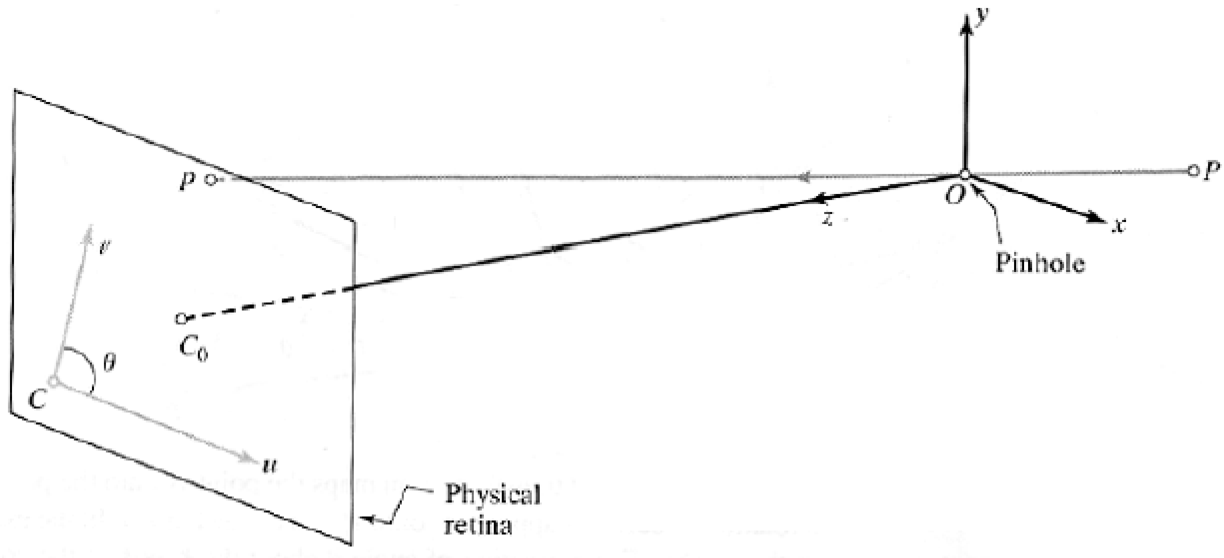
x = αX/Z

Y = βY/Z



CCD sensor array

Intrinsic Parameters: Shifted optical center

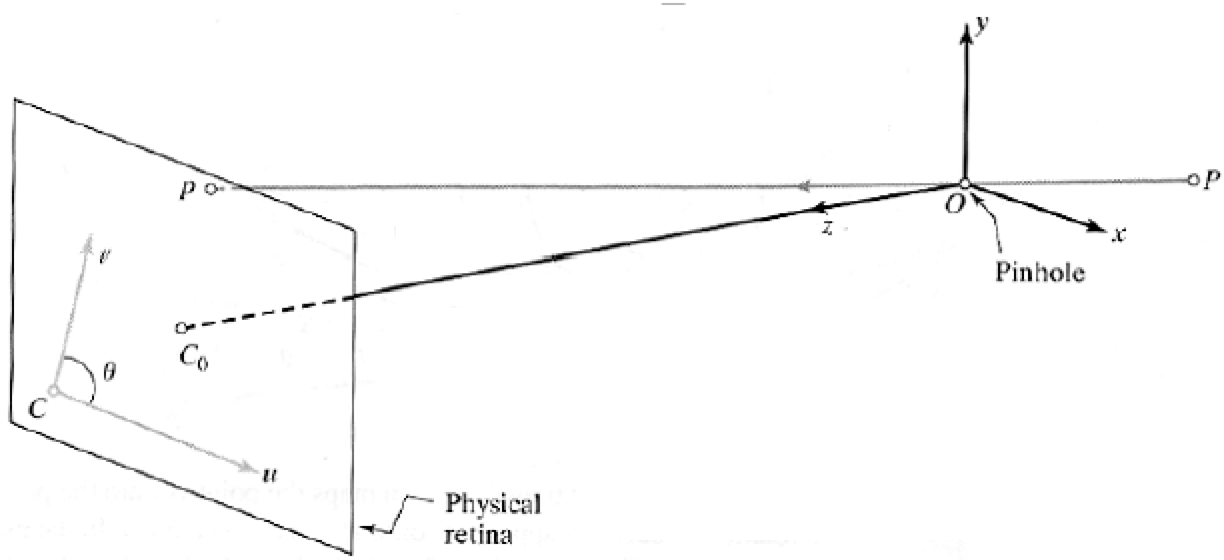


And the origin of the chip sensor may not coincide with the pixel center (the chip may not be centered, or the origin of the chip may not be in the middle of the chip.)

$x = \alpha X/Z + x_0$

$Y = \beta Y/Z + y_0$

Intrinsic Camera Parameters



In homogeneous coordinates

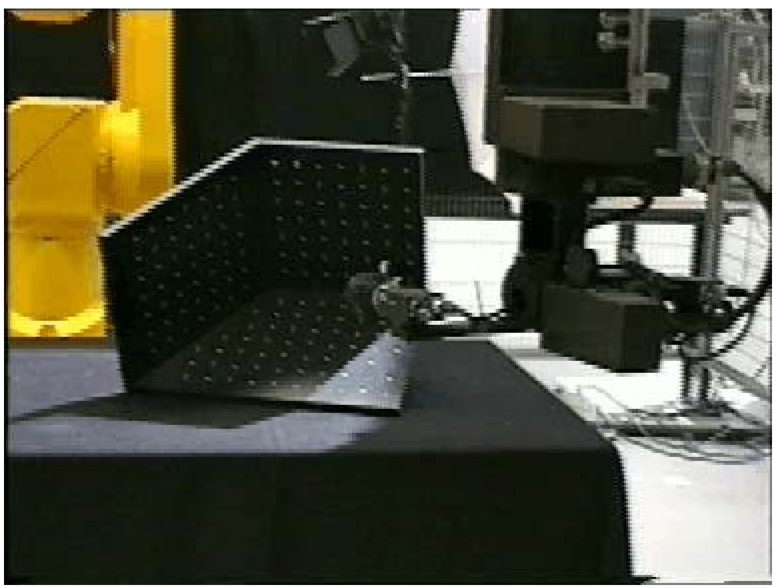
$$w\begin{bmatrix}u\\v\\1\end{bmatrix}=\begin{bmatrix}\alpha & s & u_0 & 0\\0 & \beta & v_0 & 0\\0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}x\\y\\z\\1\end{bmatrix}$$

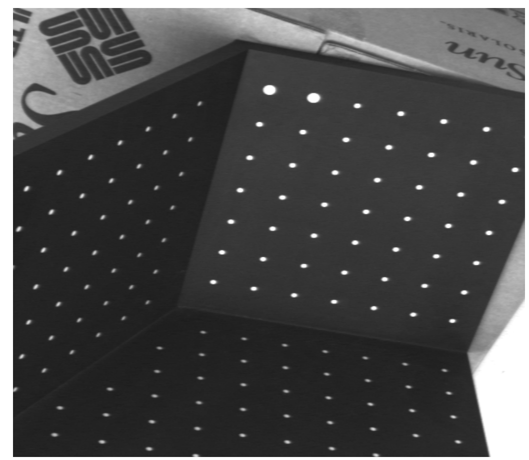
s represents skew. (Not always included. E.g., OpenCV.)

These five parameters are known as ***intrinsic parameters***

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \cong \begin{pmatrix} f_x & s & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

Given enough examples of 3D points and their 2D projections, we can solve for the 5 intrinsic parameters…

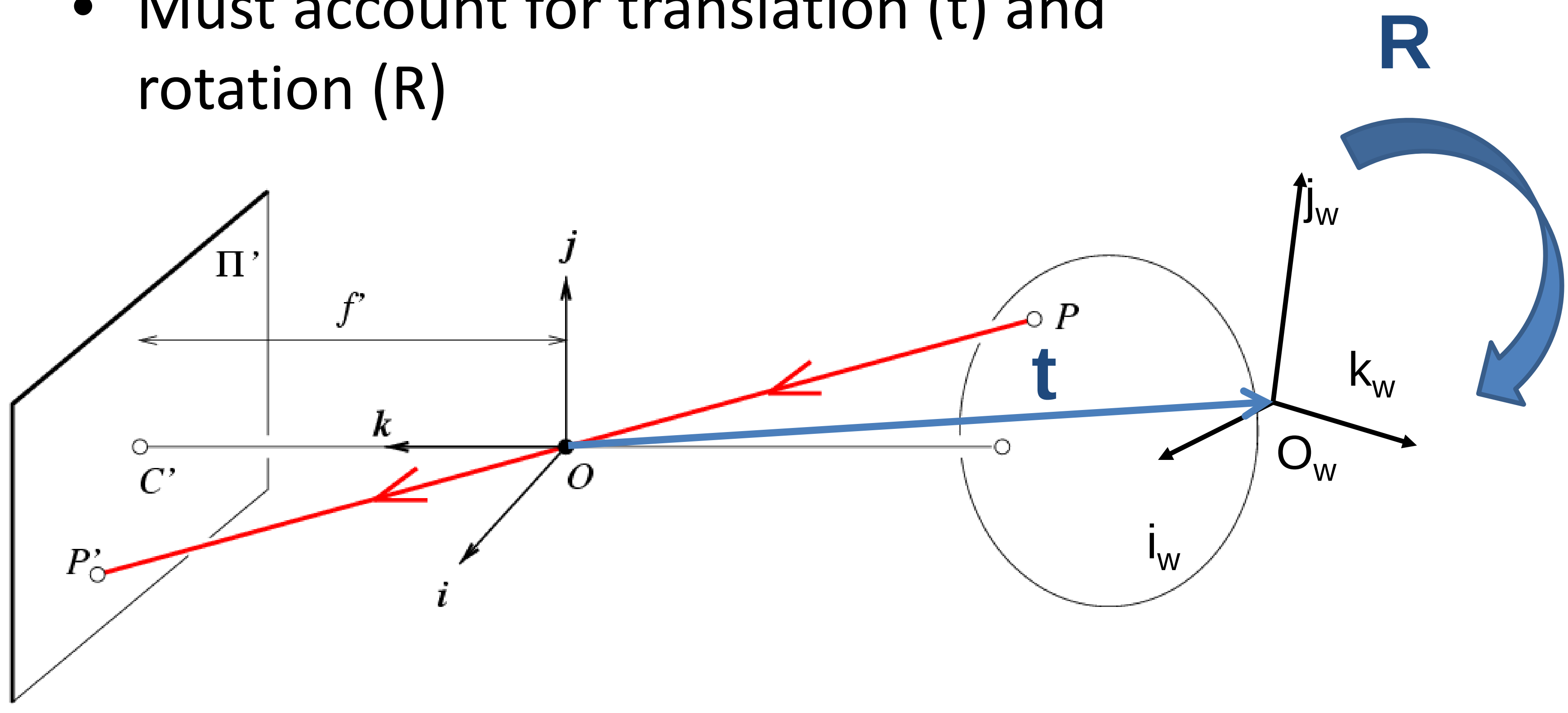




Example calibration object…

Extrinsic Parameters

- Recall: Camera may not be at the origin, looking down the z-axis
  - Must account for translation (t) and rotation (R)



Is translation a linear transform?

- T(x,y,z) → (x', y', z')
- Not in Cartesian coordinates

$$x' = x + t_x$$

$$y' = y + t_y$$

$$z' = z + t_z$$

- *Another win for homogeneous coordinates!*

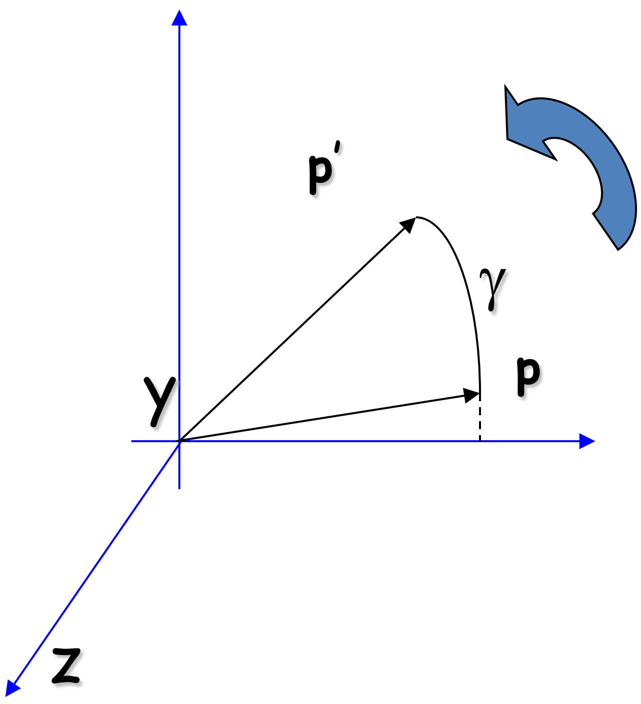
Translations with homogeneous coordinates

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\begin{cases} x' = x + t_x \\ y' = y + t_y \\ z' = z + t_z \end{cases}$$

3D Rotation of Points

Rotation around the coordinate axes, counter-clockwise:



$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}$$

$$R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}$$

$$R_z(\gamma) = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Putting it all together…

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \cong \begin{pmatrix} f_x & s & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}$$

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \cong \begin{pmatrix} . & . & .T_x \\ . & R & .T_y \\ . & . & .T_z \end{pmatrix} \begin{pmatrix} X_{grid} \\ Y_{grid} \\ Z_{grid} \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \cong \begin{pmatrix} f_x & s & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} . & . & .T_x \\ . & R & .T_y \\ . & . & .T_z \end{pmatrix} \begin{pmatrix} X_{grid} \\ Y_{grid} \\ Z_{grid} \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \cong P \begin{pmatrix} X_{grid} \\ Y_{grid} \\ Z_{grid} \\ 1 \end{pmatrix}$$

P is a 3x4 matrix that defines the projection of world points onto images. P has 11 degrees of freedom: 5 intrinsic parameters, 3 rotation, 3 translation

How do we learn P?

- Given many examples of:
  - World points (X,Y,Z), and
  - Their image points (x,y)
- Solve for P.

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \cong P \begin{pmatrix} X_{world} \\ Y_{world} \\ Z_{world} \\ 1 \end{pmatrix}$$

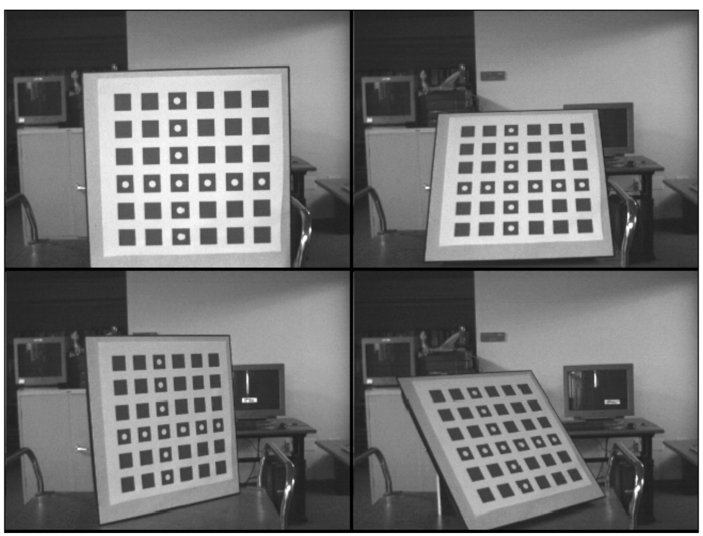
- Then $P = \begin{pmatrix} . & . & .T_x \\ . & KR & .T_y \\ . & . & .T_z \end{pmatrix}$

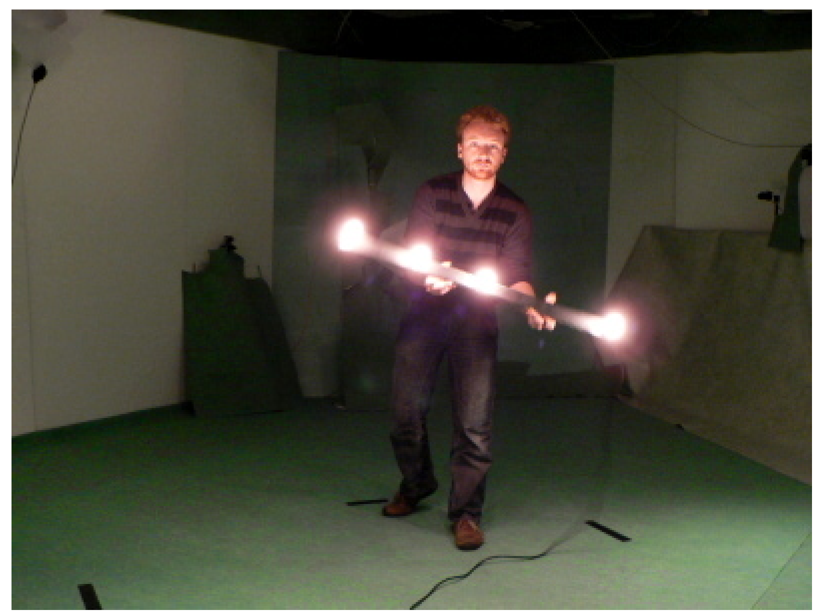
Rotation + intrinsic, all mixed up.

translation

Camera Calibration

- Process of learning the projection matrix P
- Typically requires a calibration object
  - With known dimensions





One more (optional) step…

$$P = \begin{pmatrix} . & . & .T_x \\ . & KR & .T_y \\ . & . & .T_z \end{pmatrix}$$

Then take the QR decomposition of this part of the matrix to get the rotation and the intrinsic parameters.

Given a matrix A, its QR-decomposition is a matrix decomposition of the form

$$A = Q R,$$

where R is an upper triangular matrix and Q is an orthogonal matrix, i.e., one satisfying

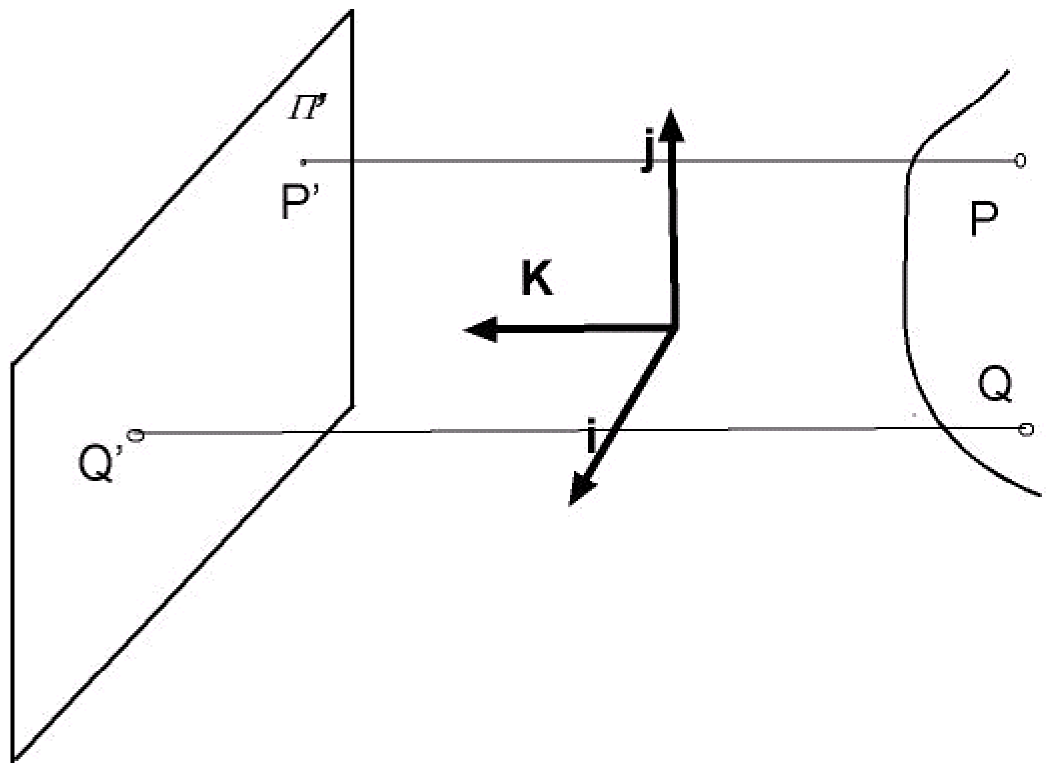
$$Q^T Q = I,$$

where Q^T is the transpose of Q and I is the identity matrix. This matrix decomposition can be used to solve linear systems of equations.

from mathworld

Different Projection Model: Orthographic Projection

- x’ = x; y’ = y



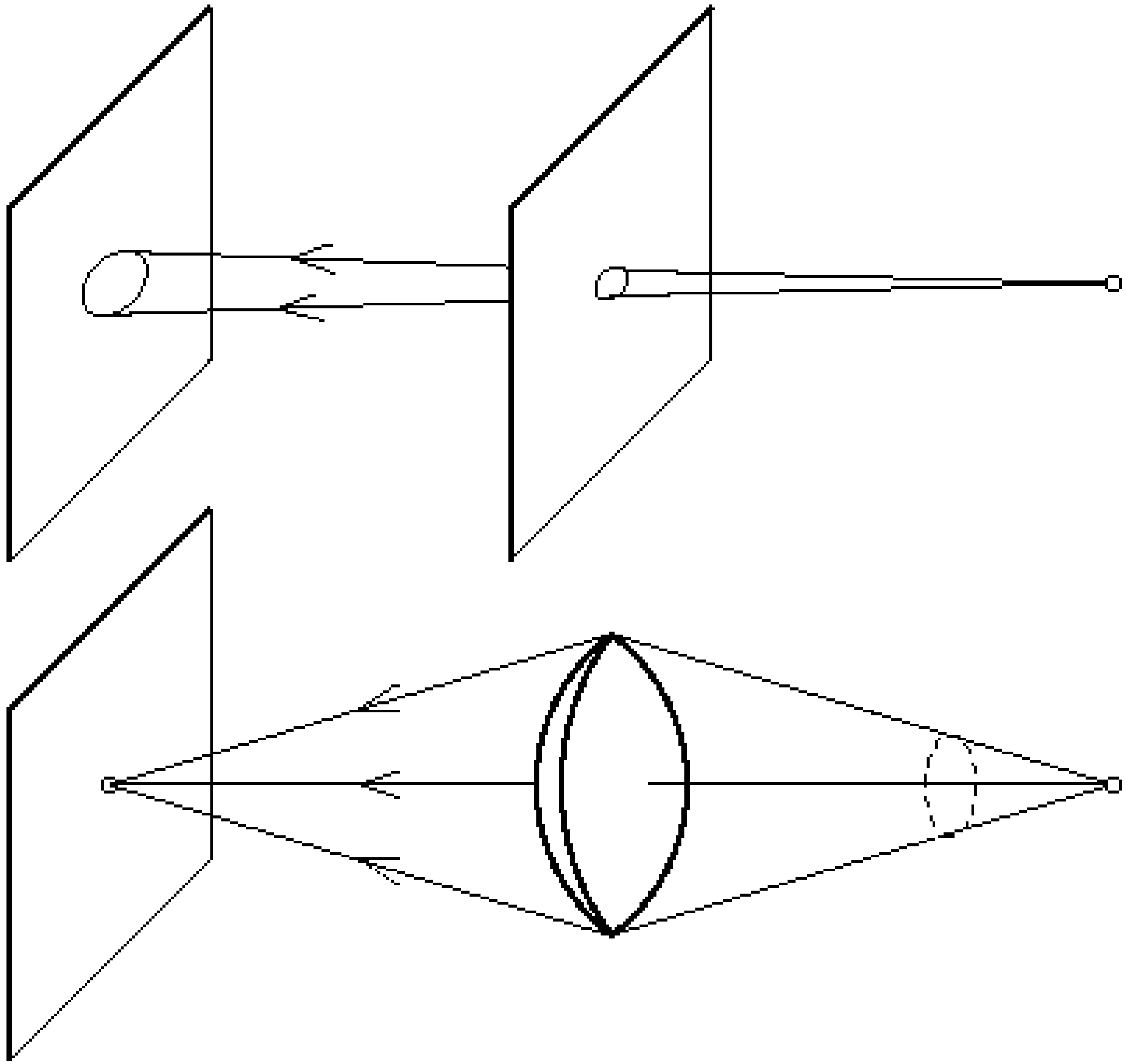
Scaled Orthographic Projection

- $(x, y, z) \rightarrow s(x, y)$
  - s is constant for all points
- Parallel lines no longer converge
  - They remain parallel

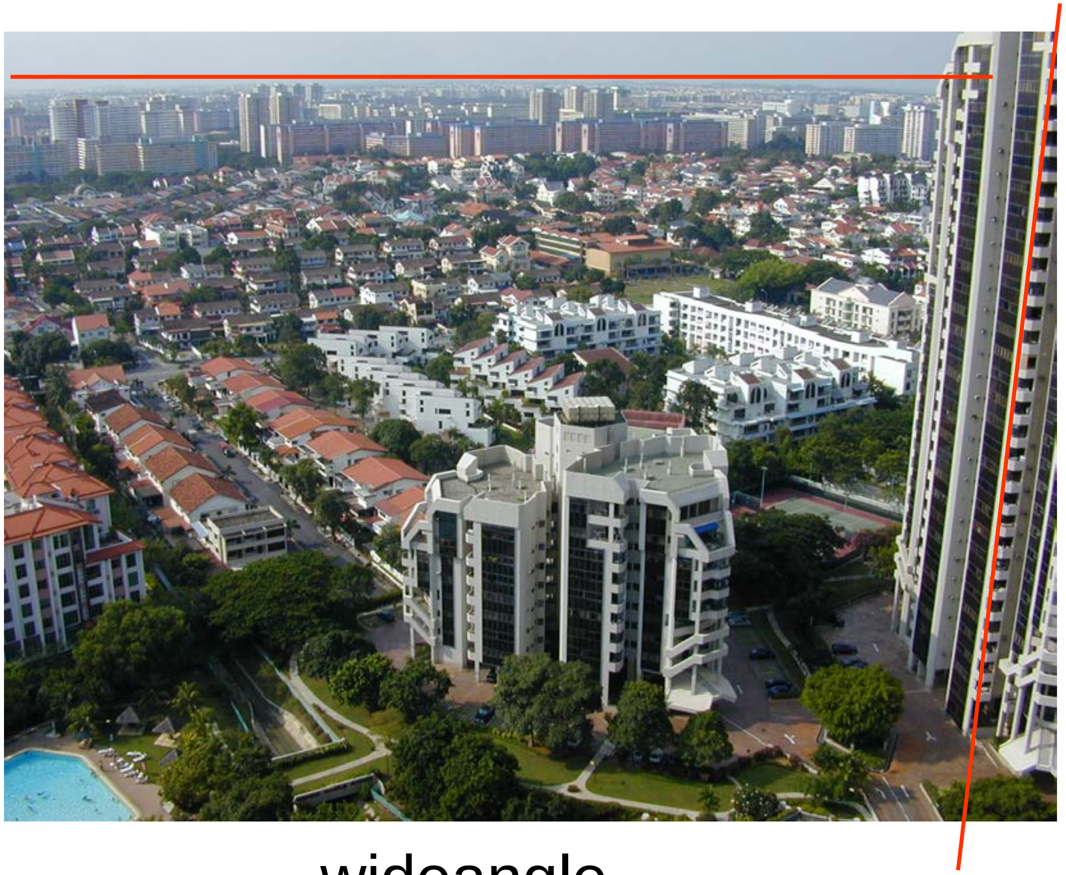
Scaled Orthographic Projection

- Why consider this?
  - Simpler math
  - Accurate when object is small & distant
  - Most useful for recognition
- Pinhole perspective is much more accurate for 3D scenes
  - Used in structure from motion (we'll talk about later)
- When accuracy really matters, we must model "real" cameras

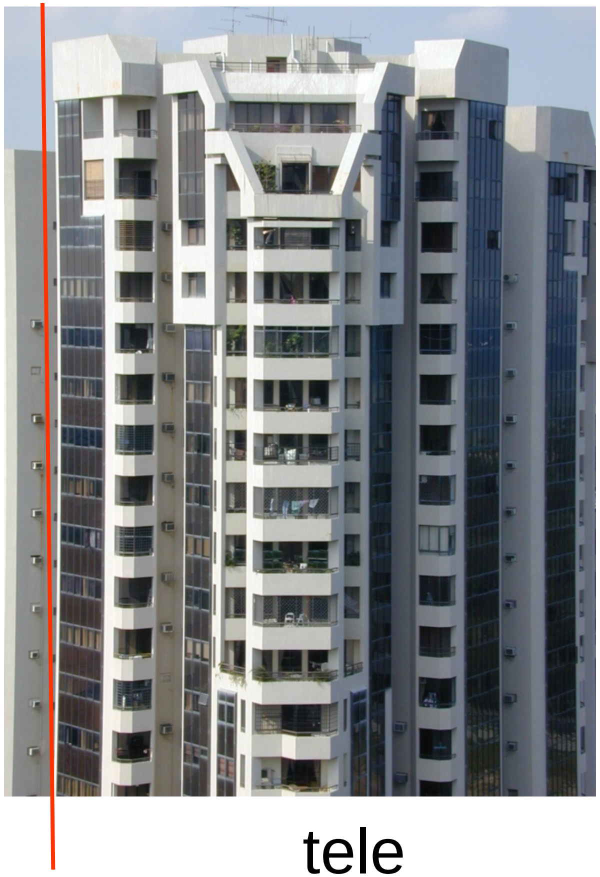
Cameras w/ Lenses



What is left?
Barrel and Pincushion Distortion



wideangle



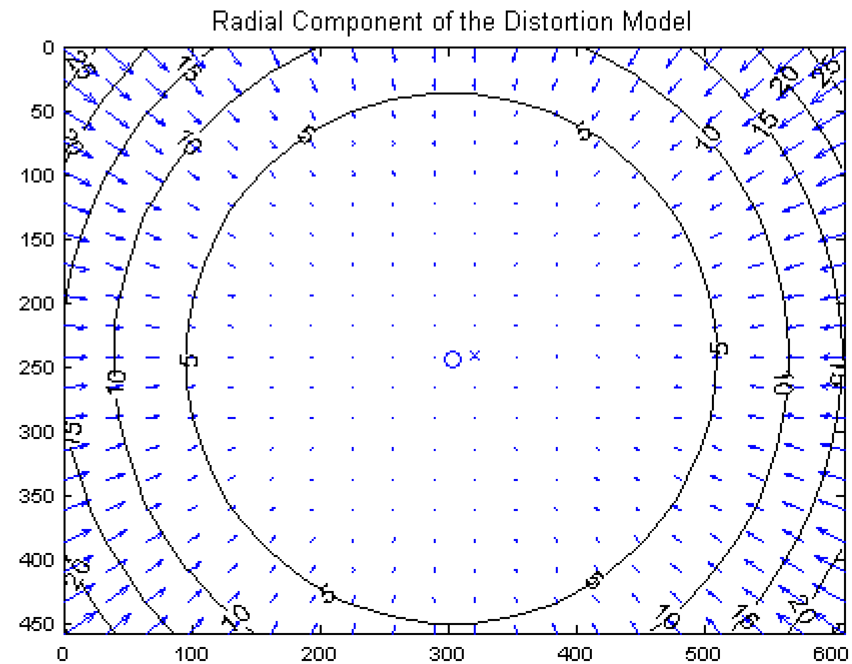
tele

Models of Radial Distortion

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x_d \\ y_d \end{pmatrix}(1 + k_1 r^2 + k_2 r^4)$$

distance from center

k1 and k2 are additional calibration parameters. 4 is common number of radial distortion parameters



Pixel error	= [0.1174, 0.1159]	
Focal Length	= (657.303, 657.744)	+/- [0.2849, 0.2894]
Principal Point	= (302.717, 242.334)	+/- [0.5912, 0.5571]
Skew	= 0.0004198	+/- 0.0001905
Radial coefficients	= (-0.2535, 0.1187, 0)	+/- [0.00231, 0.009418, 0]
Tangential coefficients	= (-0.0002789, 5.174e-005)	+/- [0.0001217, 0.0001208]

Distortion Corrected…

Applications

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \cong \begin{pmatrix} f_x & s & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} . & . & .T_x \\ . & R & .T_y \\ . & . & .T_z \end{pmatrix} \begin{pmatrix} X_{grid} \\ Y_{grid} \\ Z_{grid} \\ 1 \end{pmatrix} \qquad \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \cong P \begin{pmatrix} X_{grid} \\ Y_{grid} \\ Z_{grid} \\ 1 \end{pmatrix}$$

- Can you use this? If you know the 3D coordinates of a virtual point, then you can draw it on your image…
- Often hard to know the 3D coordinate of a point; but there are some (profitable) special cases…

Applications

First-down line

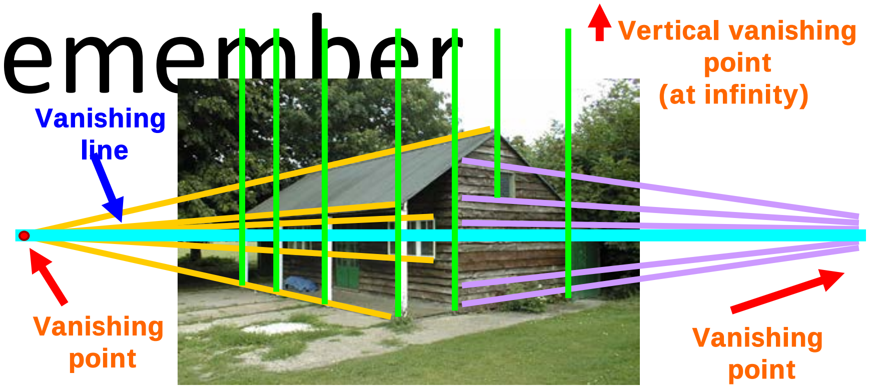
courtesy of Sportvision

Applications

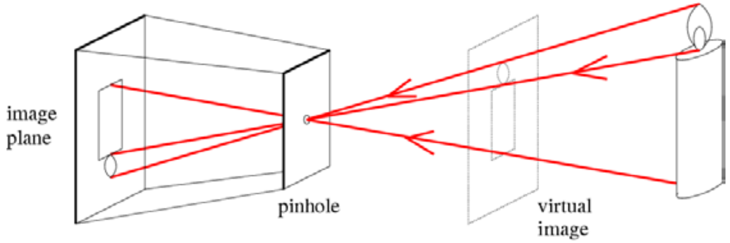
Virtual advertising

courtesy of Princeton Video Image

Things to remember



- Vanishing points and vanishing lines



- Pinhole camera model and camera projection matrix

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\mathbf{X}$$

- Homogeneous coordinates

$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$