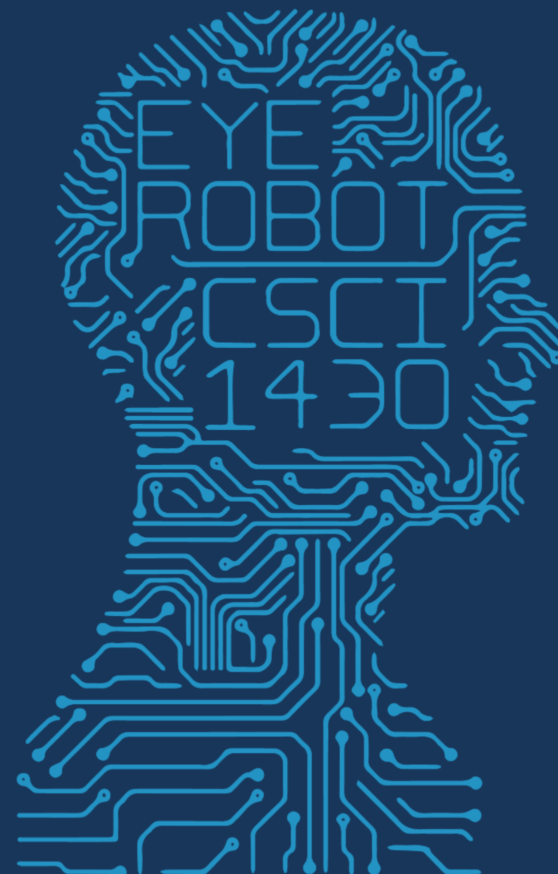
1950

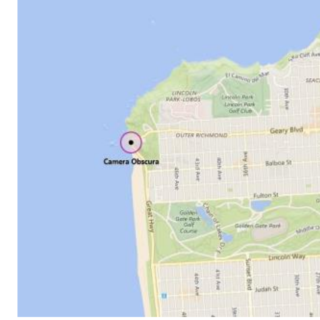
Future Vision



2017 MWF 1pm 368

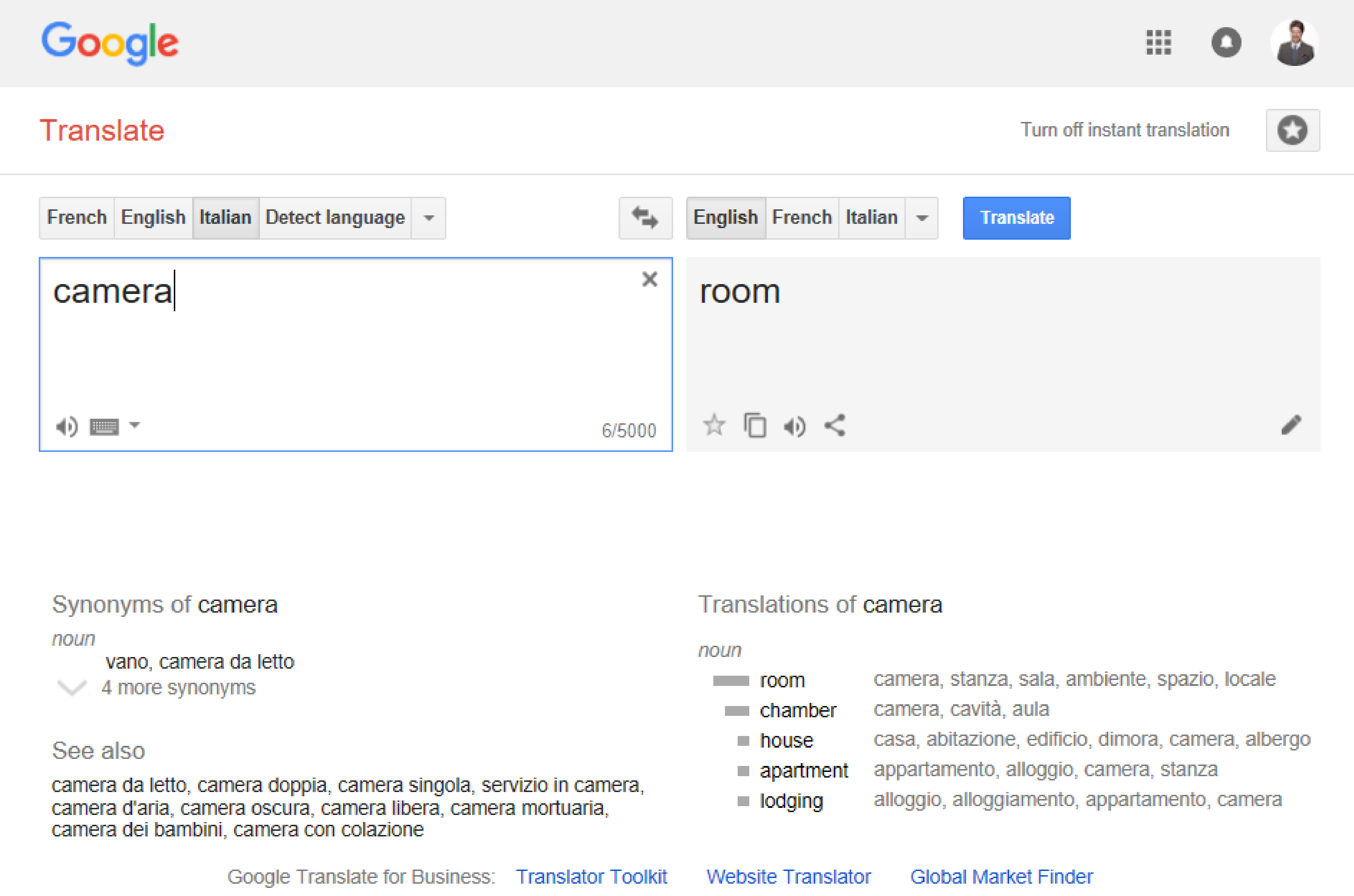
Computer Vision

James, San Francisco, Aug. 2017



CAMERAS, MULTIPLE VIEWS, AND MOTION

What is a camera?


Google
Translate
Turn off instant translation
French English Italian Detect language
English French Italian
Translate
camera
6/5000
room
Synonyms of camera
noun
vano, camera da letto
4 more synonyms
See also
camera da letto, camera doppia, camera singola, servizio in camera, camera d'aria, camera oscura, camera libera, camera mortuaria, camera dei bambini, camera con colazione
Translations of camera
noun
room camera, stanza, sala, ambiente, spazio, locale
chamber camera, cavità, aula
house casa, abitazione, edificio, dimora, camera, albergo
apartment appartamento, alloggio, camera, stanza
lodging alloggio, alloggiamento, appartamento, camera
Google Translate for Business: Translator Toolkit Website Translator Global Market Finder

Camera obscura: dark room

- Known during classical period in China and Greece (e.g., Mo-Ti, China, 470BC to 390BC)

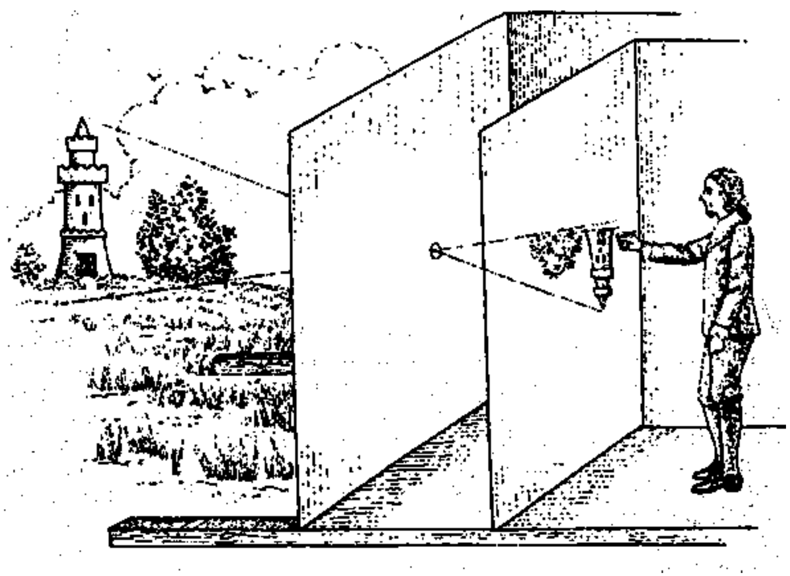


Illustration of Camera Obscura

Freestanding camera obscura at UNC Chapel Hill

Photo by Seth Ilys

James Hays

Camera obscura / lucida used for tracing

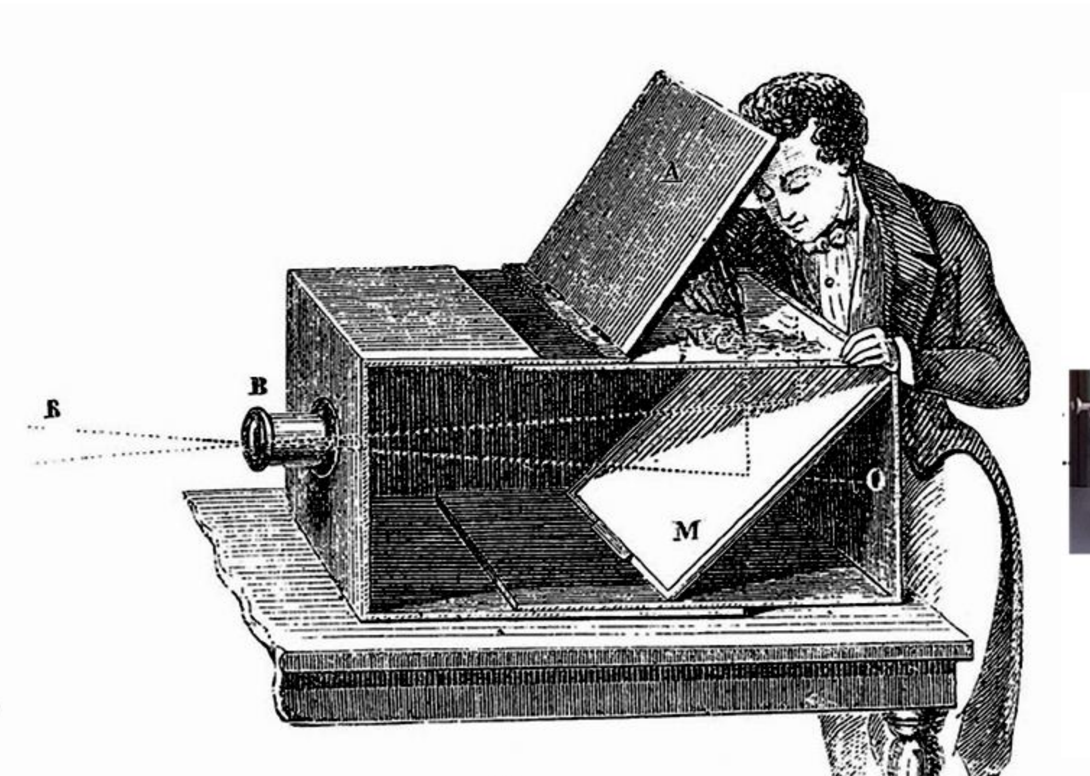
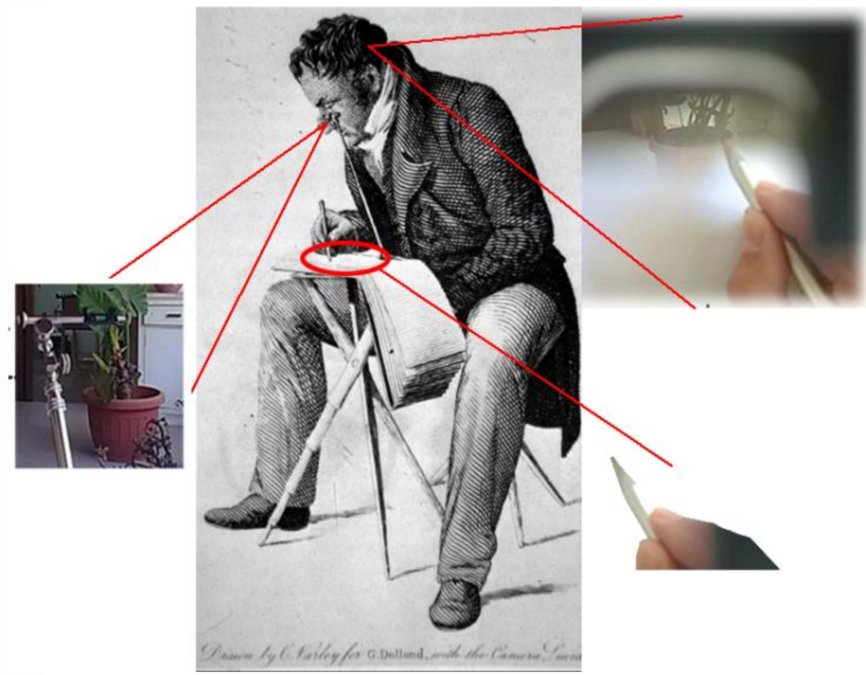


Fig. 434.

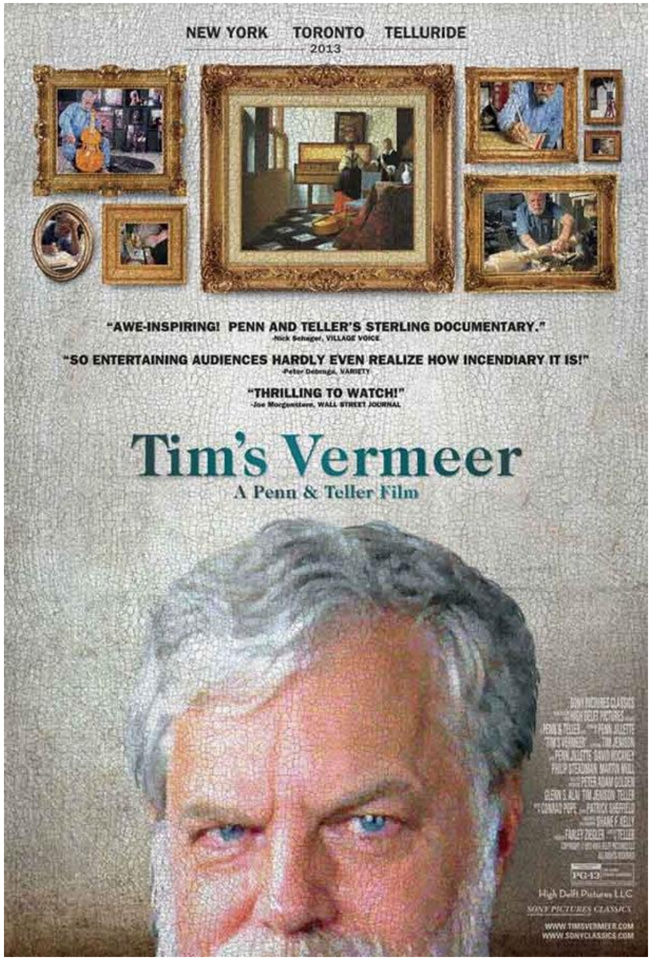


Lens Based Camera Obscura, 1568

Camera lucida

Tim’s Vermeer

Vermeer, The Music Lesson, 1665



Tim Jenison (Lightwave 3D, Video Toaster)

Tim’s Vermeer – video still

First Photograph

Oldest surviving photograph

- Took 8 hours on pewter plate

Photograph of the first photograph

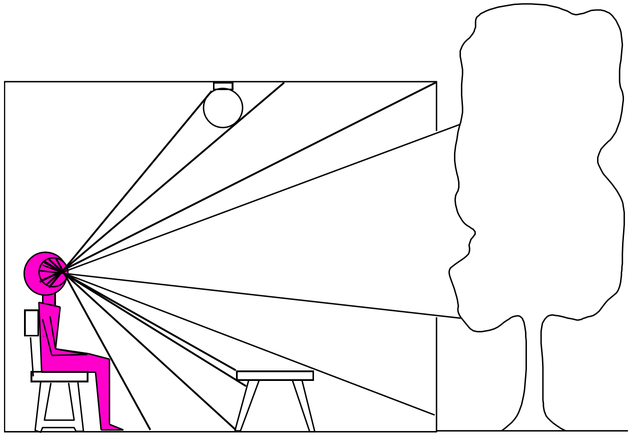
Joseph Niepce, 1826

Stored at UT Austin

Niepce later teamed up with Daguerre, who eventually created Daguerrotypes

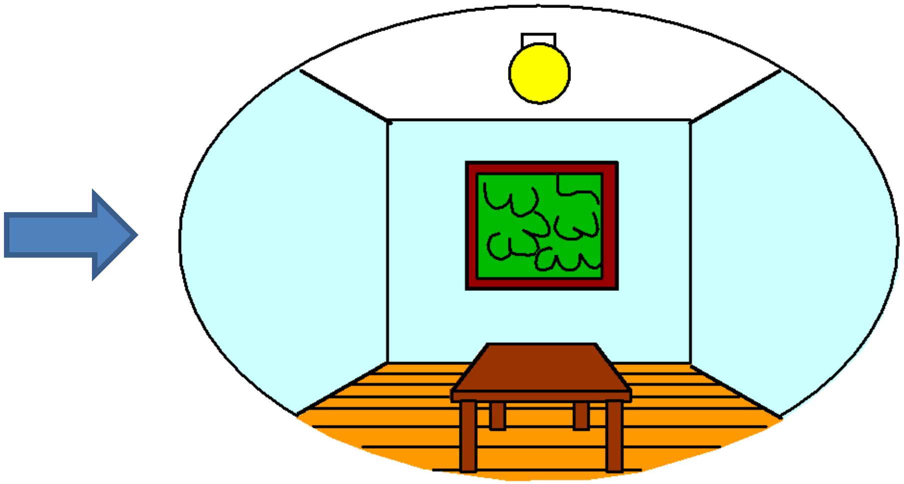
Dimensionality Reduction Machine (3D to 2D)

3D world



Point of observation

2D image



Lake Sørvágsvatn in Faroe Islands

100 metres above sea level

Lake Sørvágsvatn in Faroe Islands

~~100~~ 30 metres above sea level

amusingplanet.com, thanks to Aaron Gokaslan

Holbein's The Ambassadors - 1533

Holbein's The Ambassadors – Memento Mori

Cameras and World Geometry

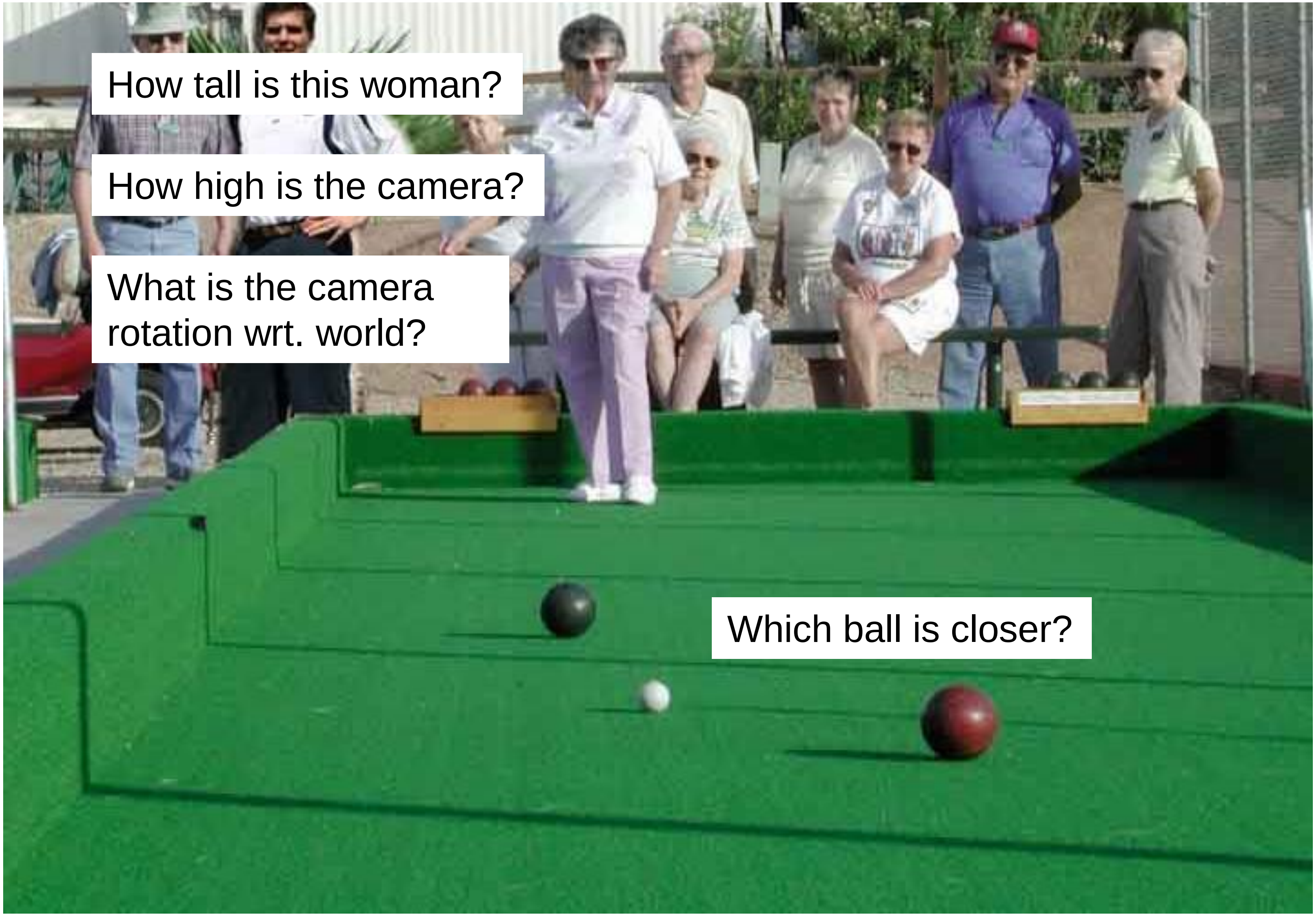


Photo Tourism

Exploring photo collections in 3D

Noah Snavely   Steven M. Seitz   Richard Szeliski

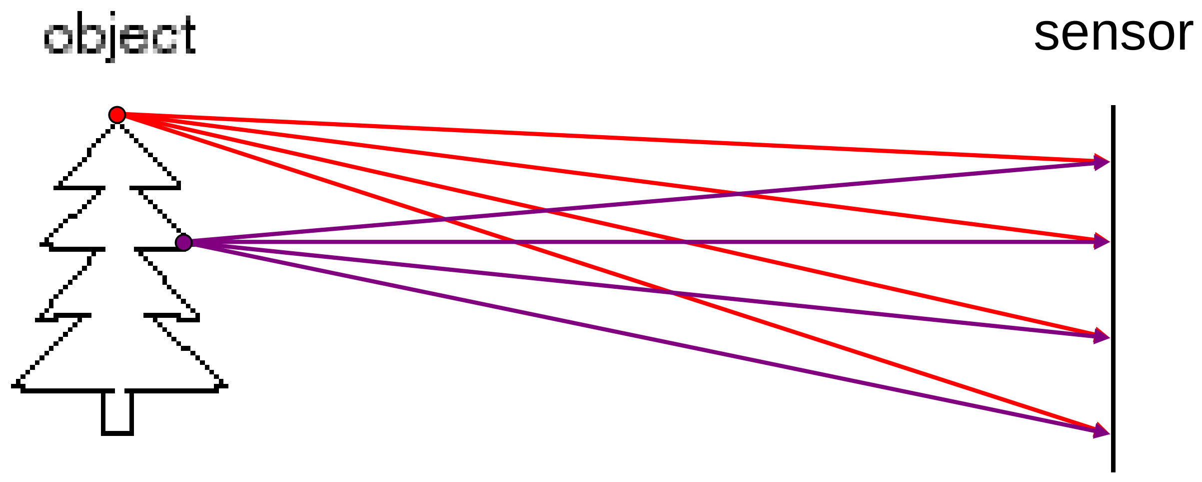
University of Washington   *Microsoft Research*

SIGGRAPH 2006

Let’s design a camera

Idea 1: Put a sensor in front of an object

Do we get a reasonable image?

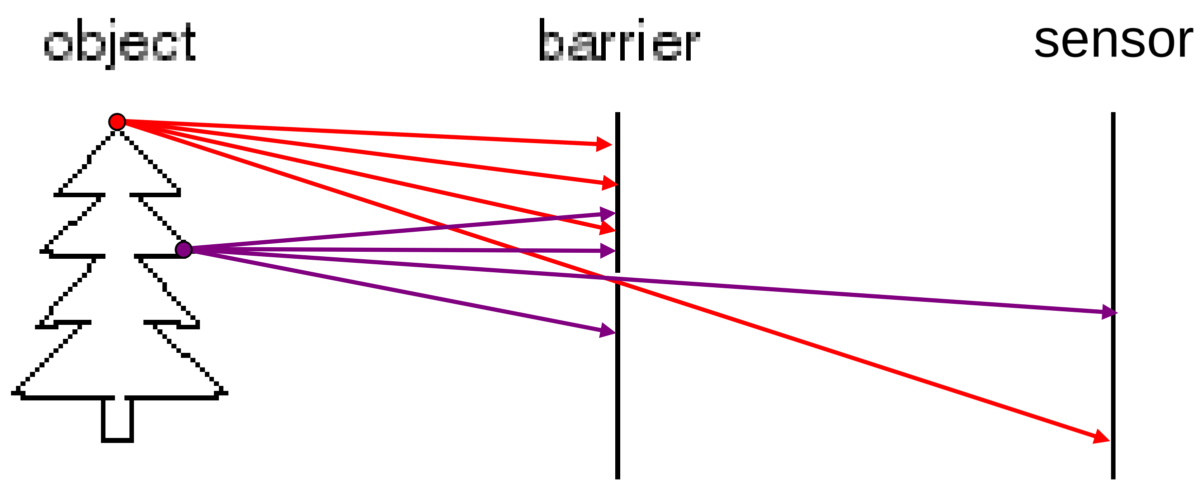


Slide source: Seitz

Let’s design a camera

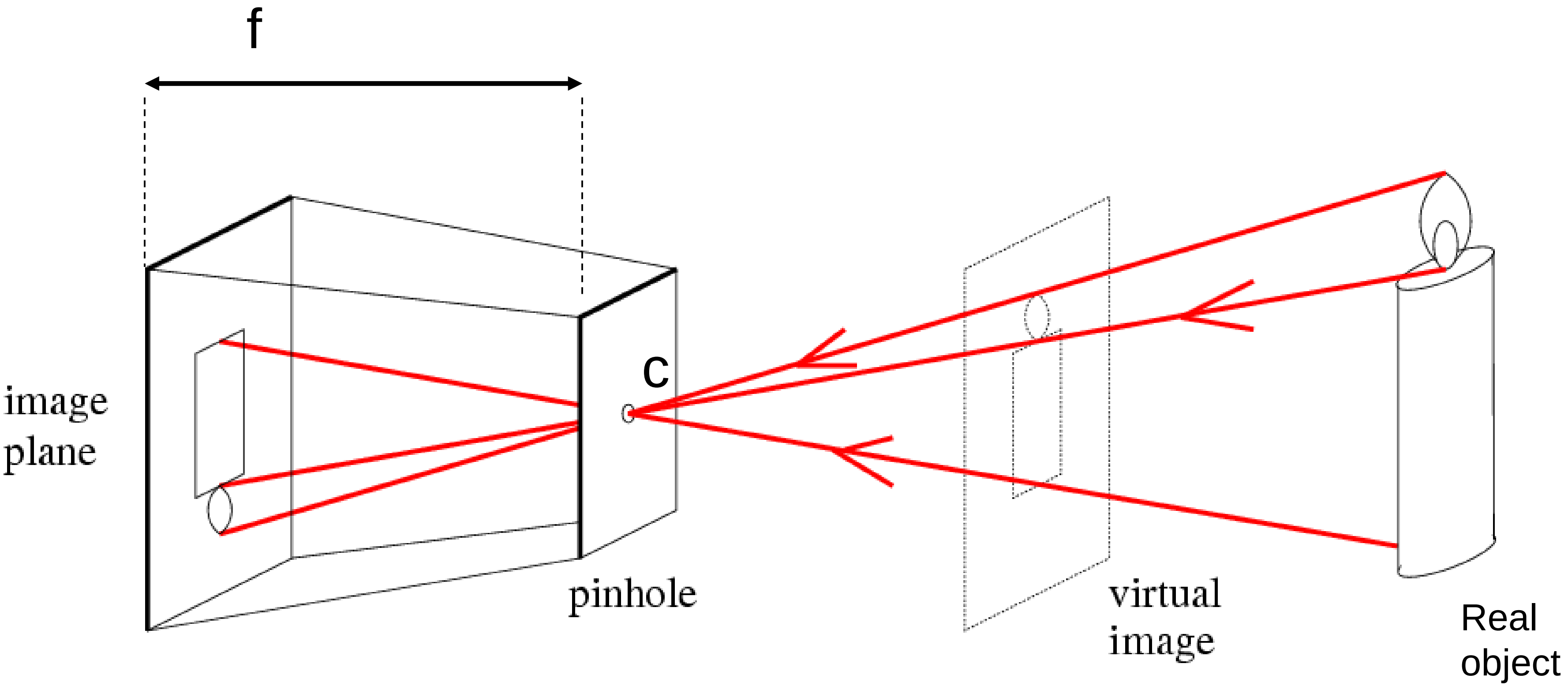
Idea 2: Add a barrier to block most rays

- Pinhole in barrier
- Only sense light from one direction.
  - Reduces blurring.
- In most cameras, this **aperture** can vary in size.



Slide source: Seitz

Pinhole camera model

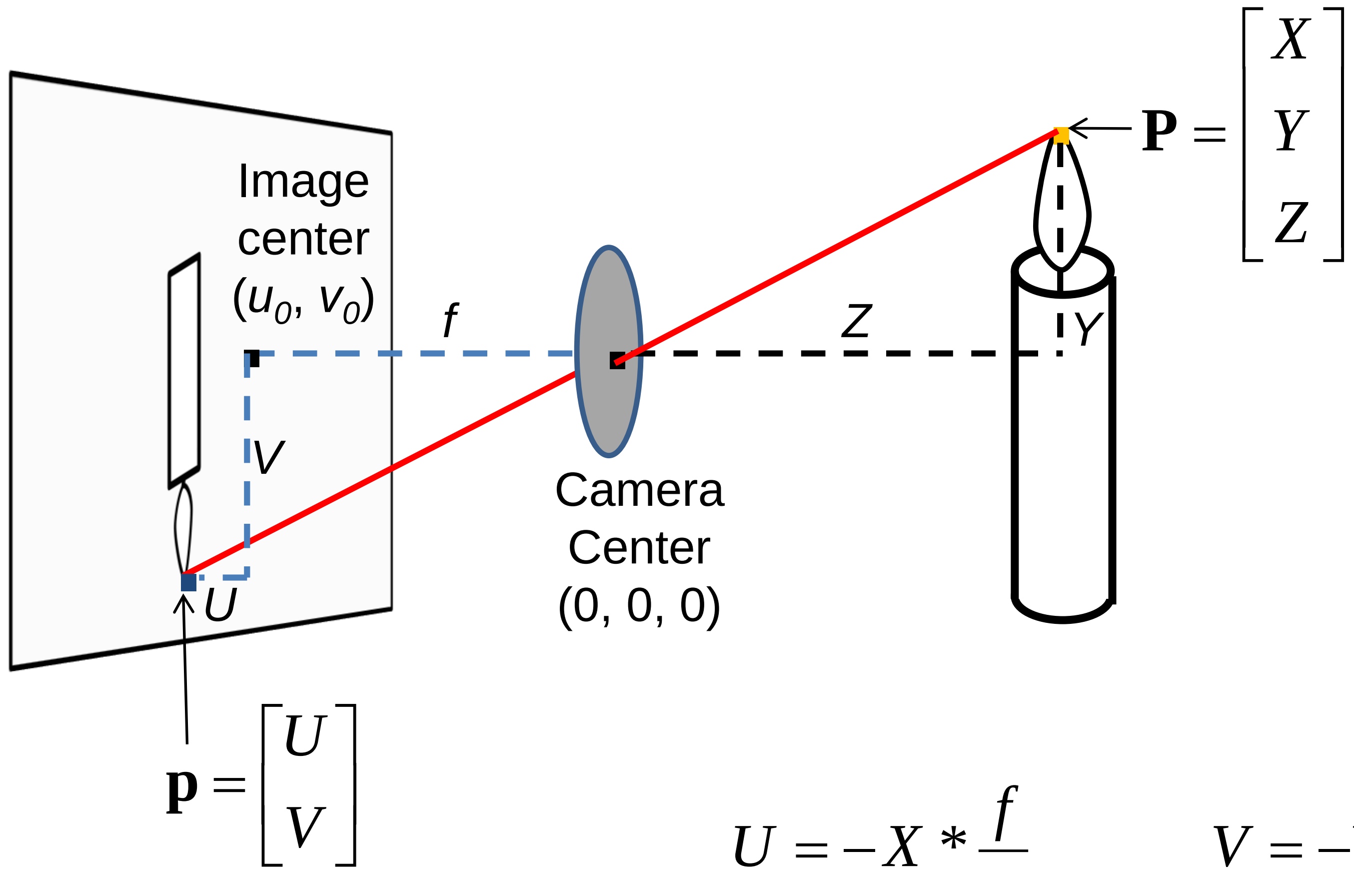


f = Focal length

c = Optical center of the camera

Figure from Forsyth

Projection: world coordinates→image coordinates



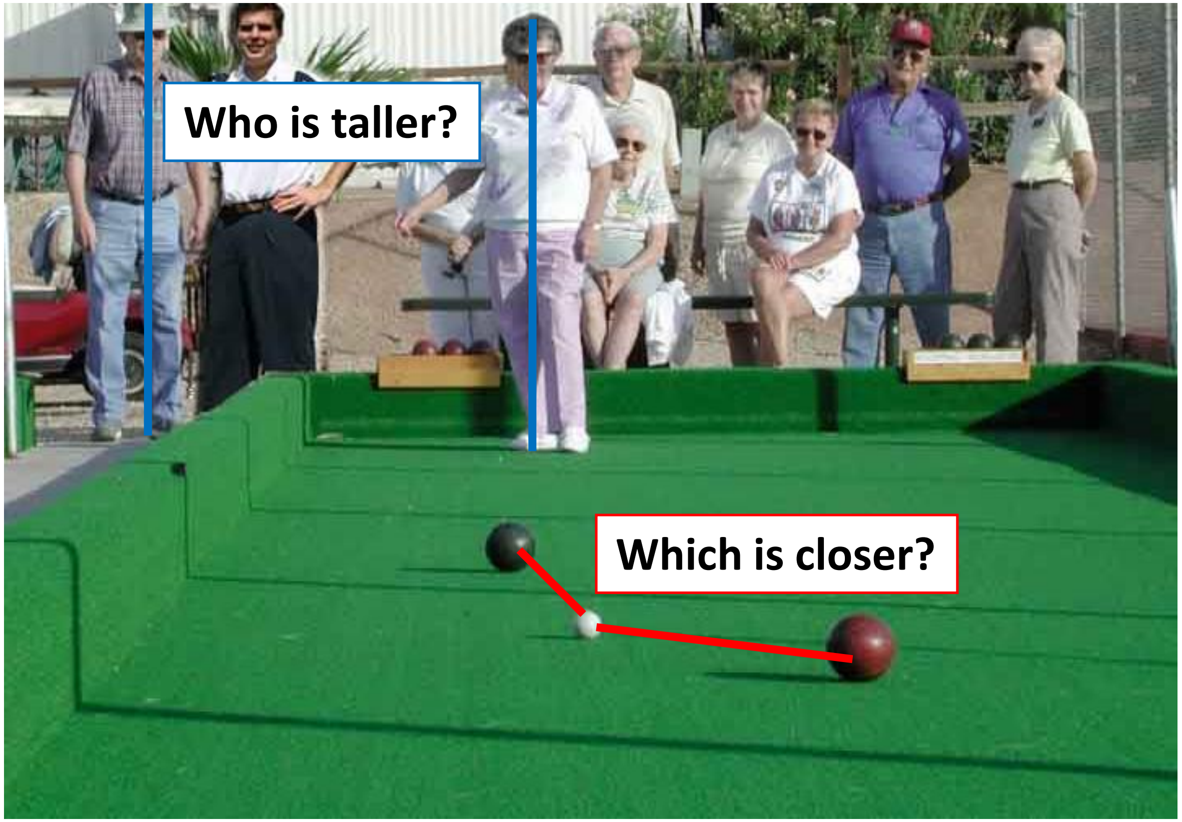
p = distance from image center

$$U = -X * \frac{f}{Z} \qquad V = -Y * \frac{f}{Z}$$

What is the effect if f and Z are equal?

Projective Geometry

Length (and so area) is lost.



Length and area are not preserved

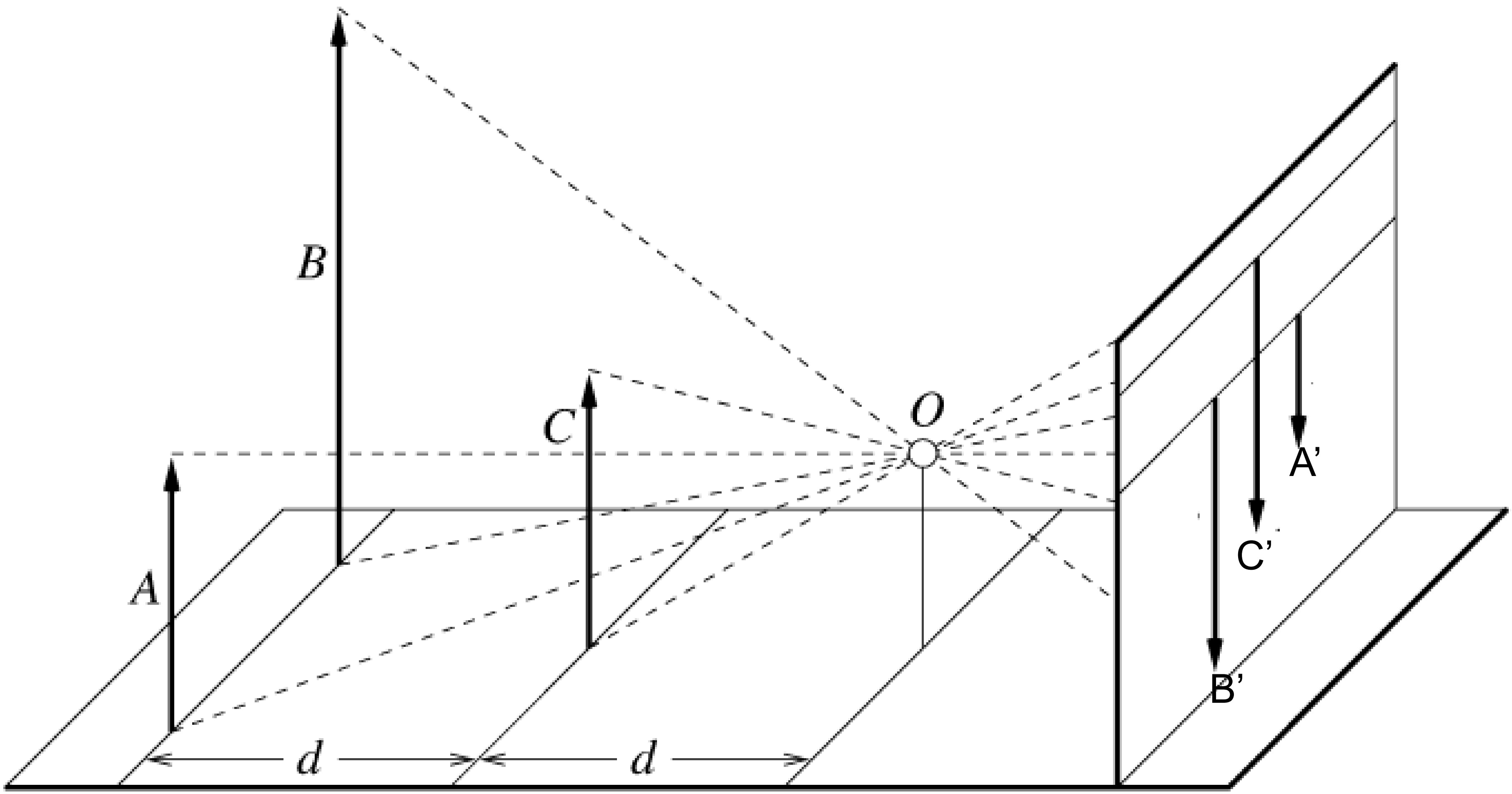
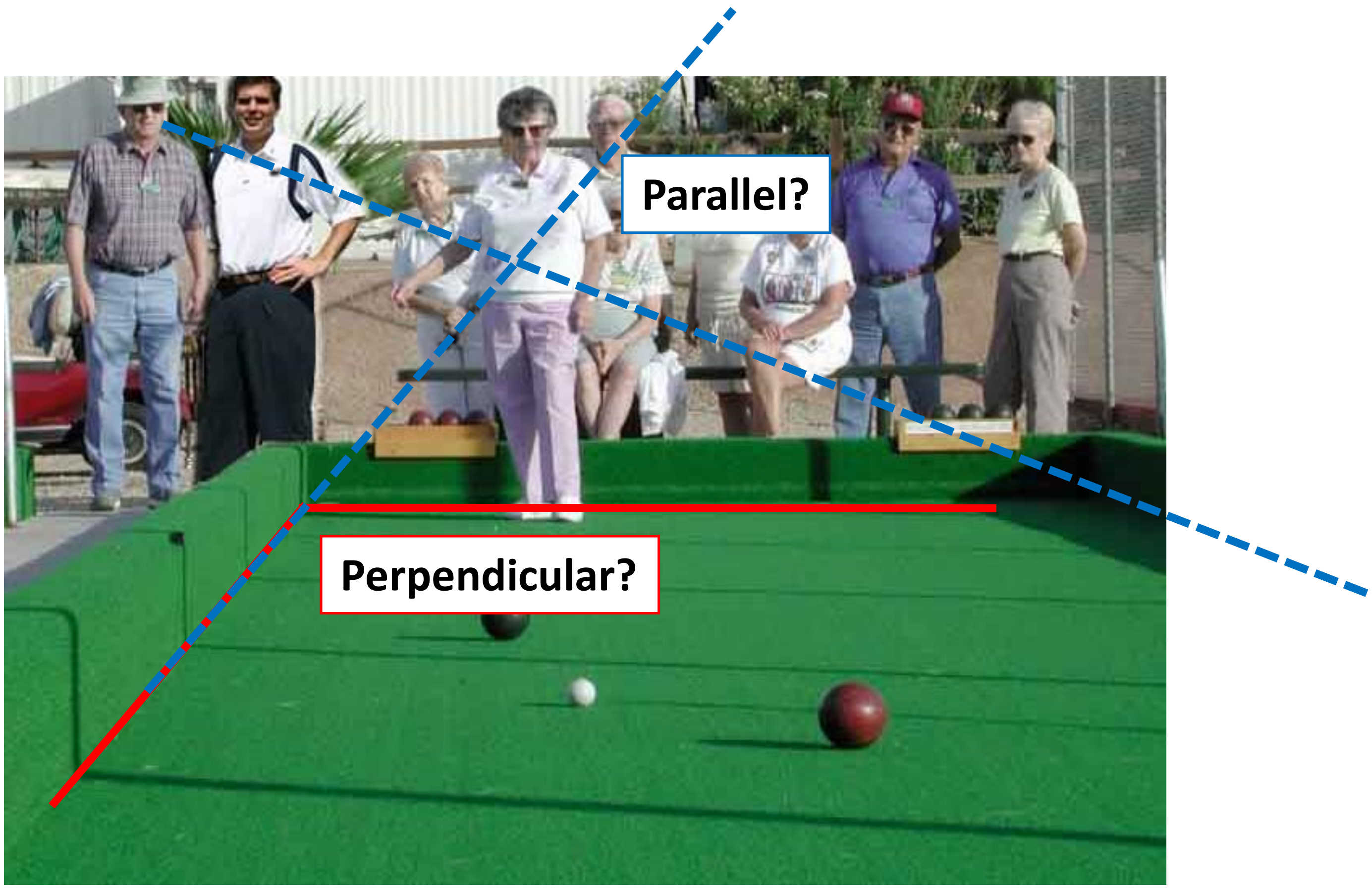


Figure by David Forsyth

Projective Geometry

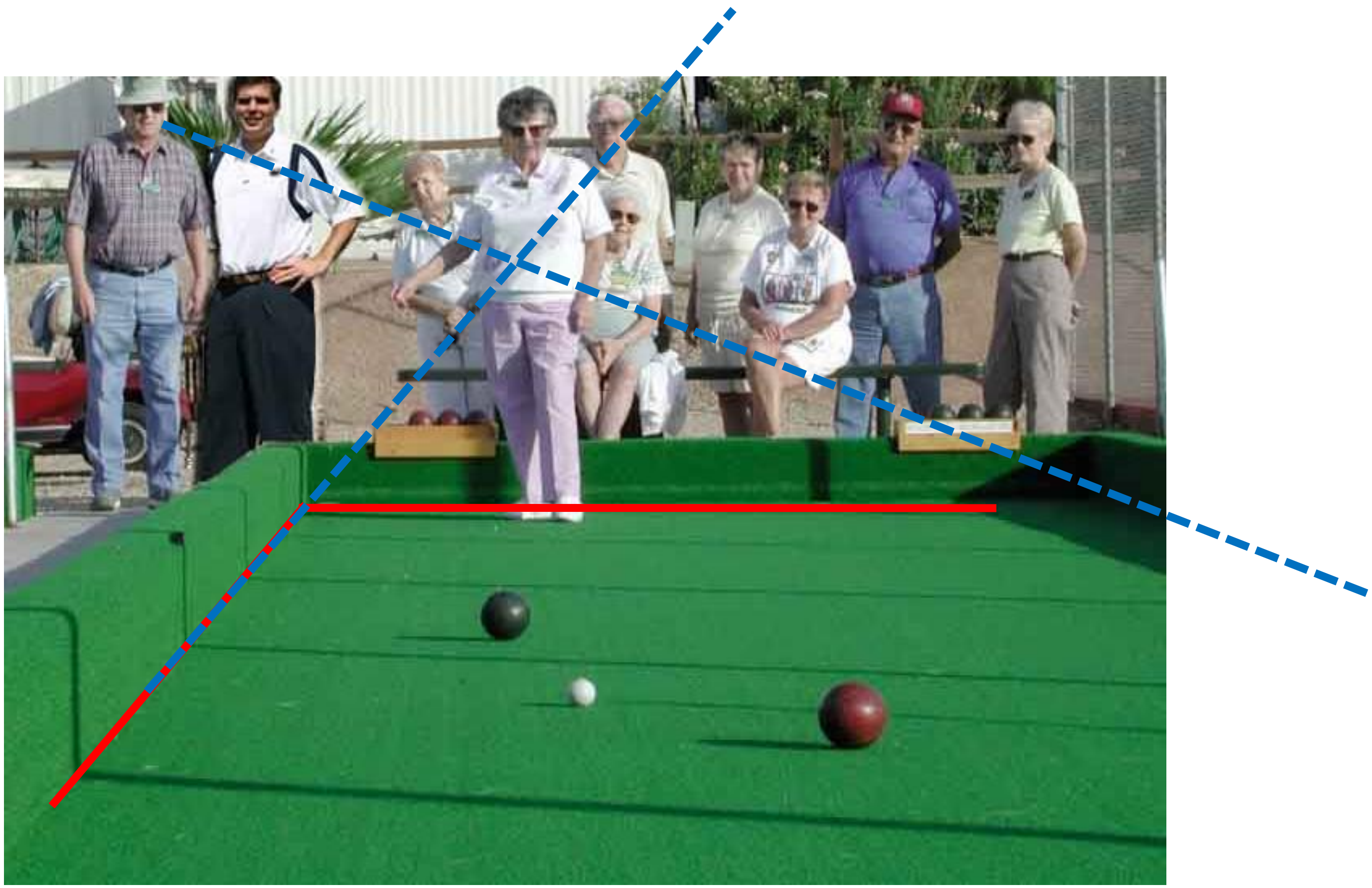
Angles are lost.



Projective Geometry

What is preserved?

- Straight lines are still straight.

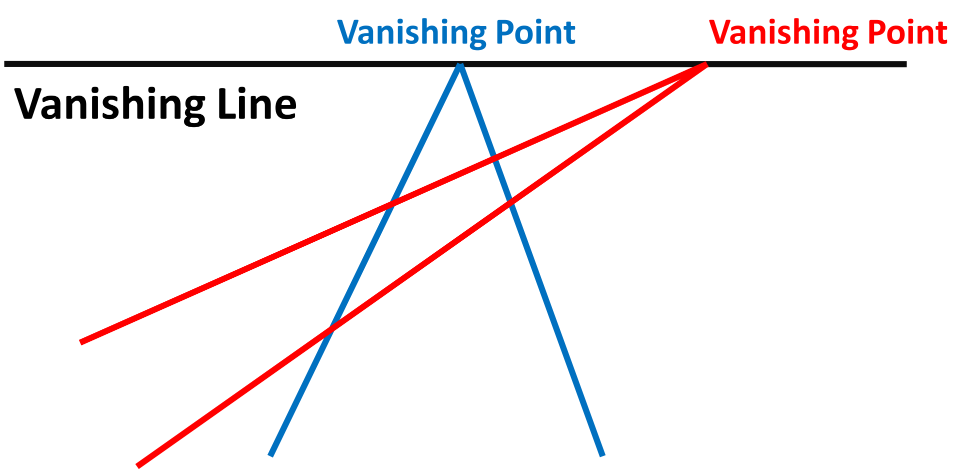


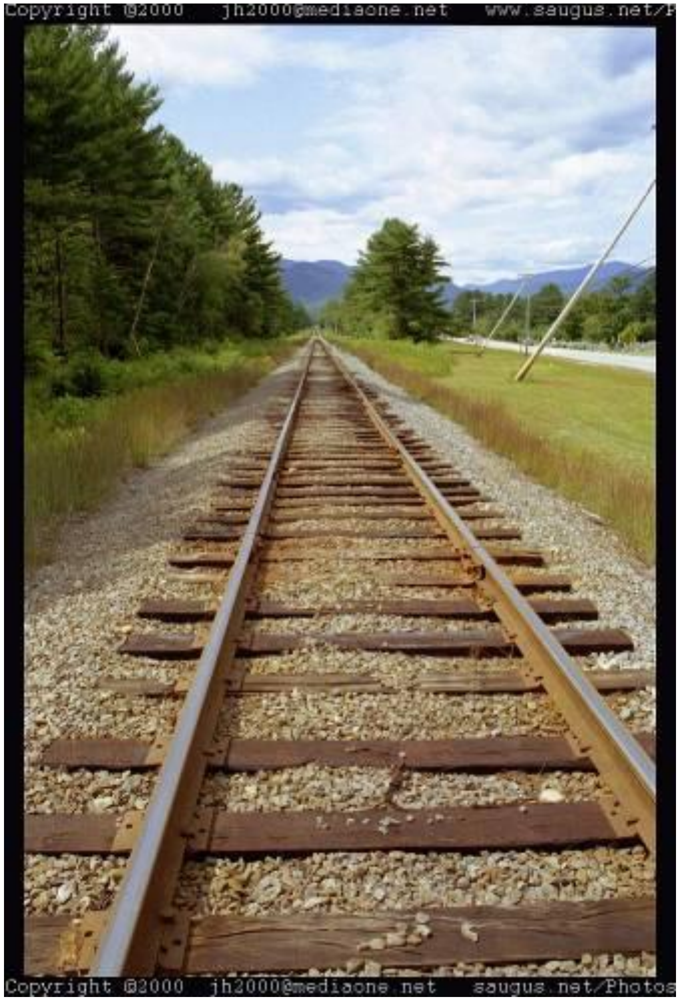
Vanishing points and lines

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

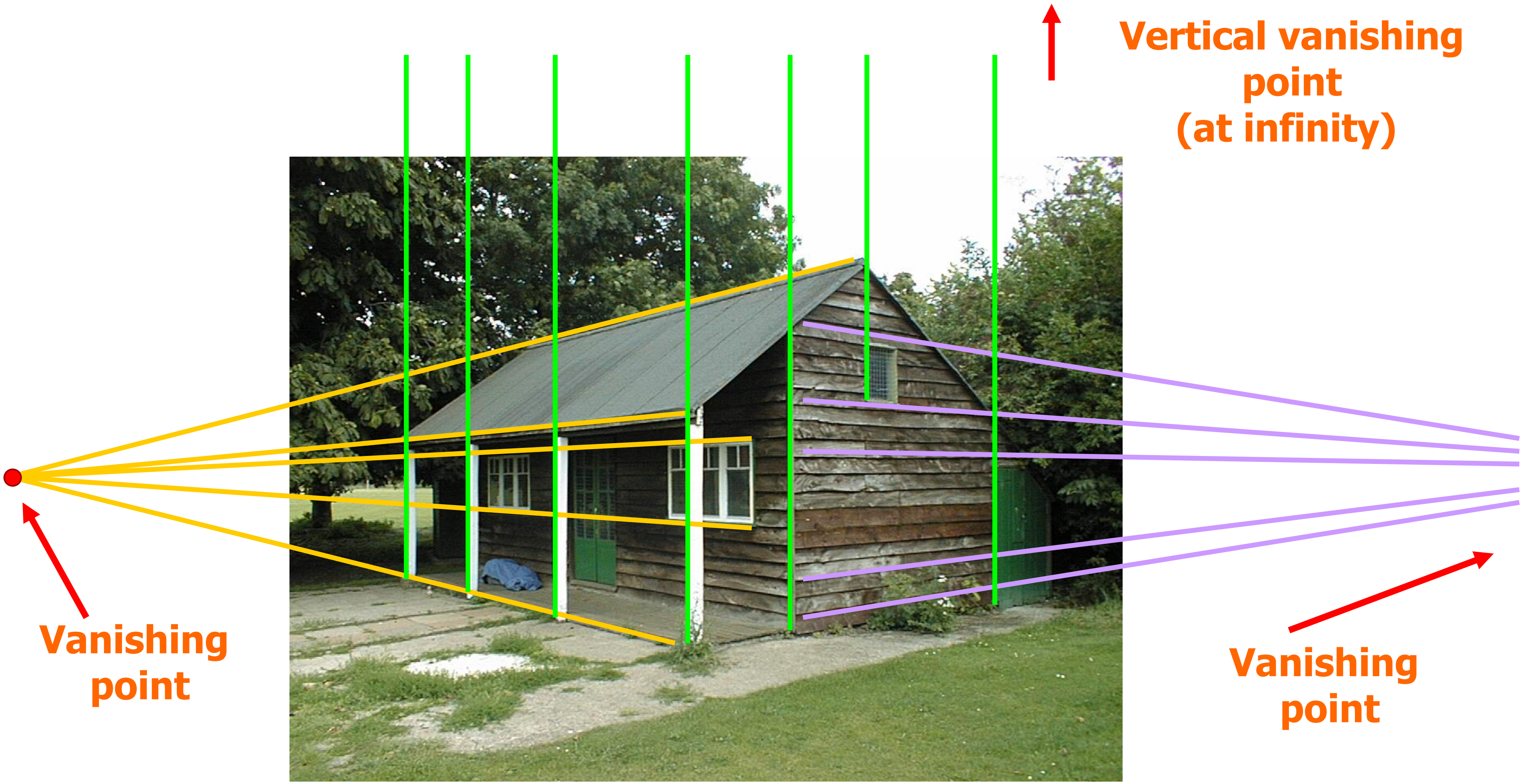
Parallel lines on the same plane in the world converge to vanishing points on a "vanishing line".

E.G., the horizon.



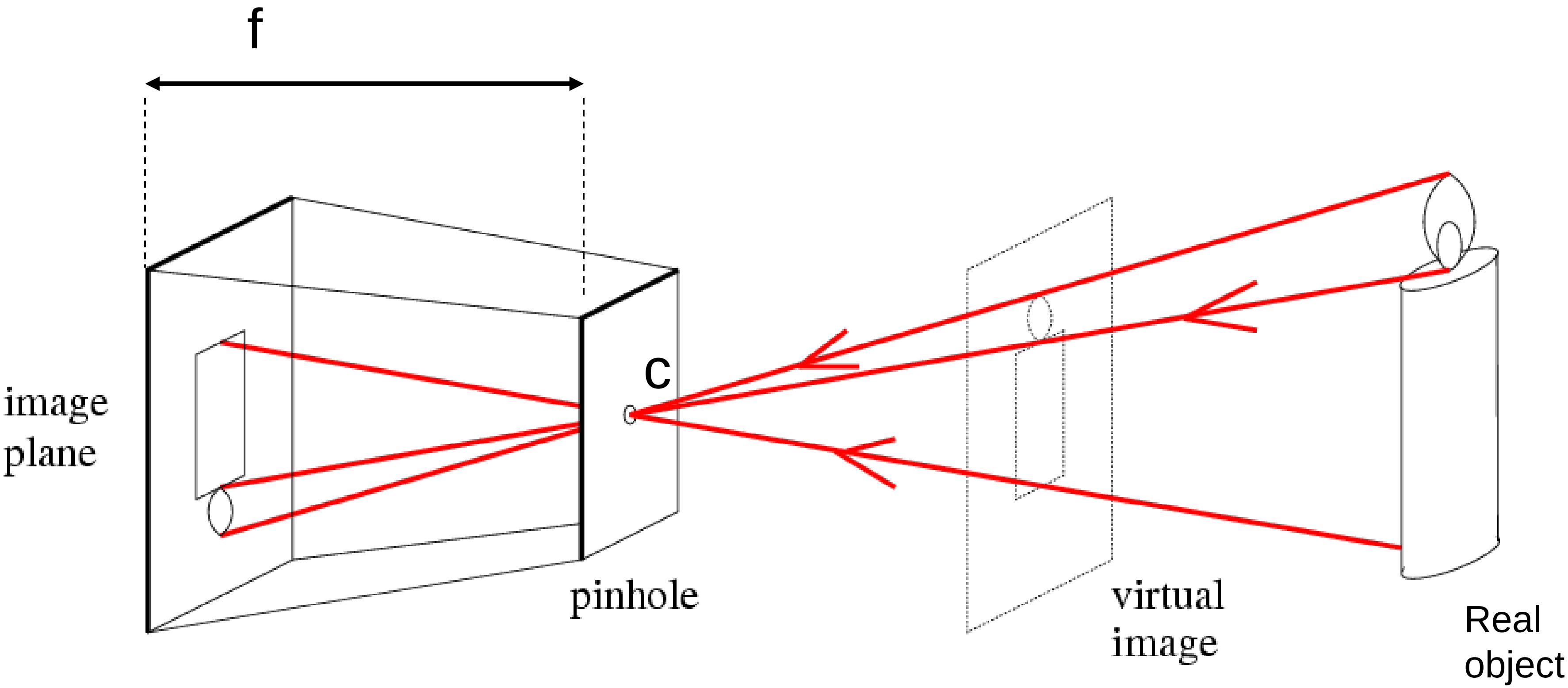


Vanishing points and lines



Slide from Efros, Photo from Criminisi

Pinhole camera model

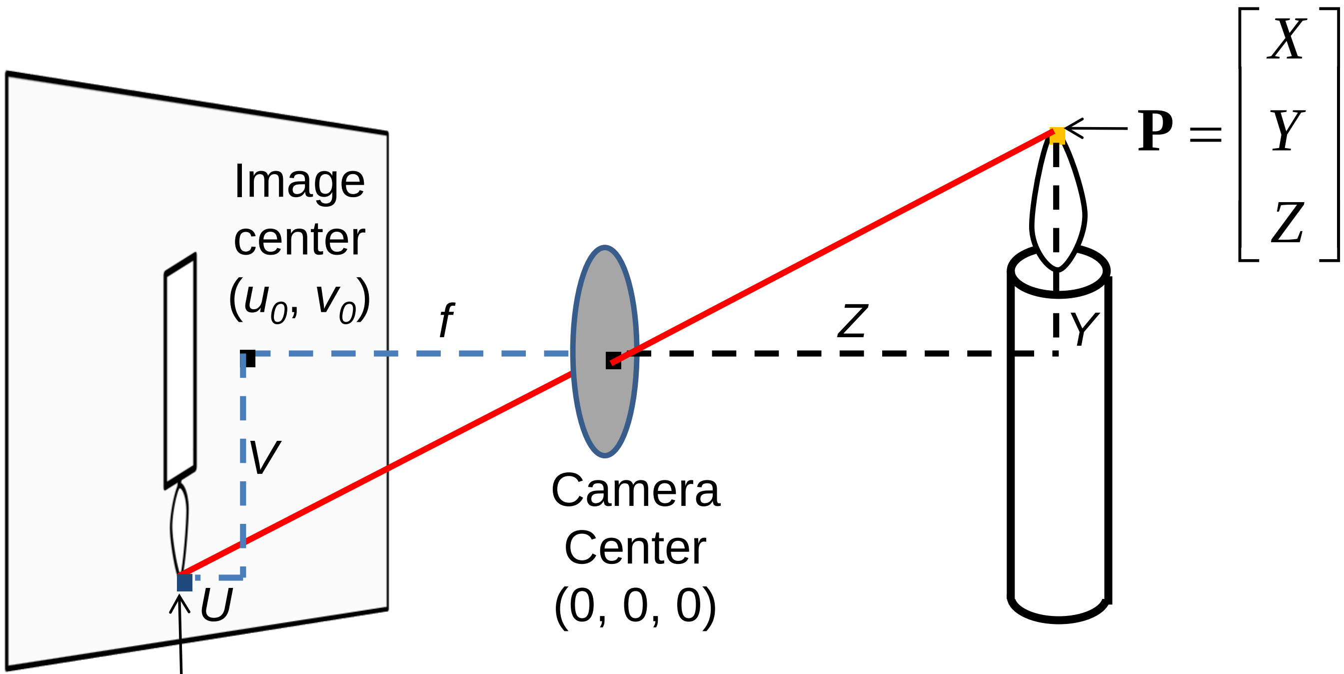


f = Focal length

c = Optical center of the camera

Forsyth

Projection: world coordinates→image coordinates



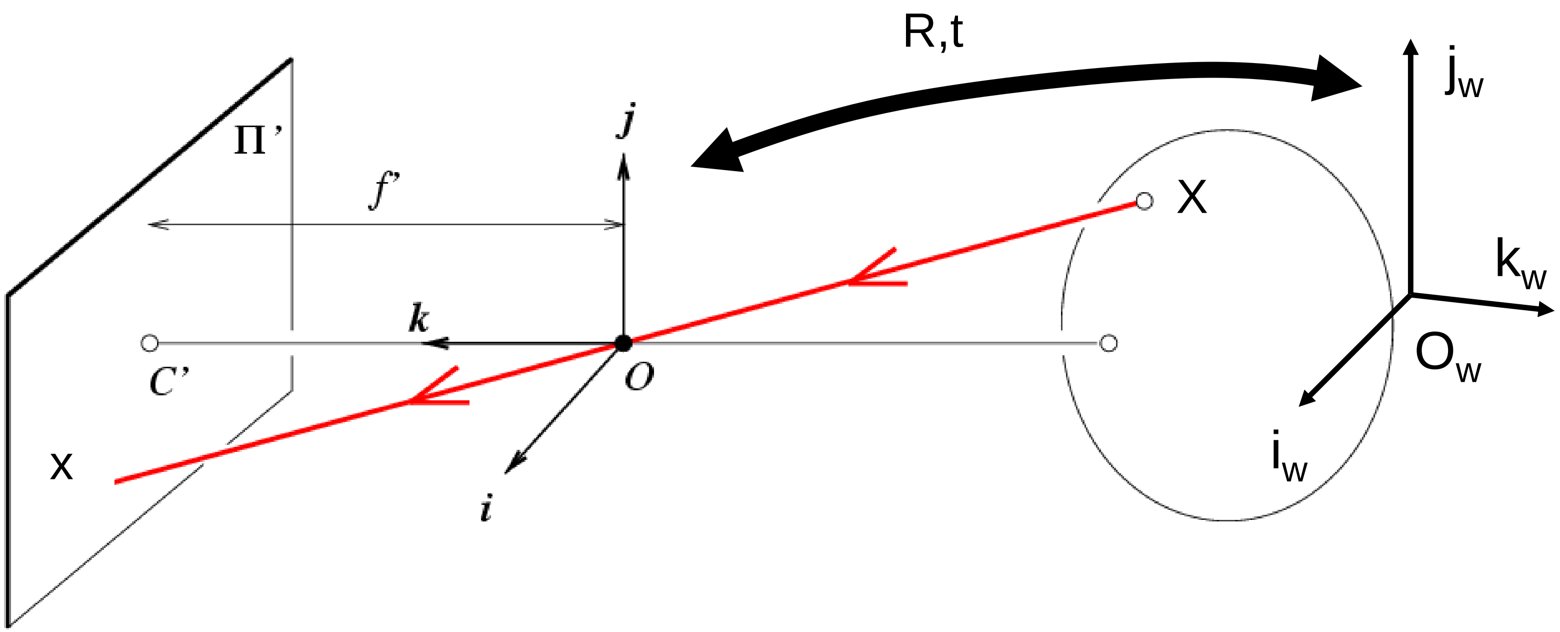
$$\mathbf{p} = \begin{bmatrix} U \\ V \end{bmatrix}$$

p = distance from image center

$$U = -X * \frac{f}{Z} \qquad V = -Y * \frac{f}{Z}$$

What is the effect if f and Z are equal?

Camera (projection) matrix



$$\mathbf{x} = \mathbf{K}\underbrace{\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}}_{\text{Extrinsic Matrix}}\mathbf{X}$$

x: Image Coordinates: (u,v,1)
K: Intrinsic Matrix (3x3)
R: Rotation (3x3)
t: Translation (3x1)
X: World Coordinates: (X,Y,Z,1)

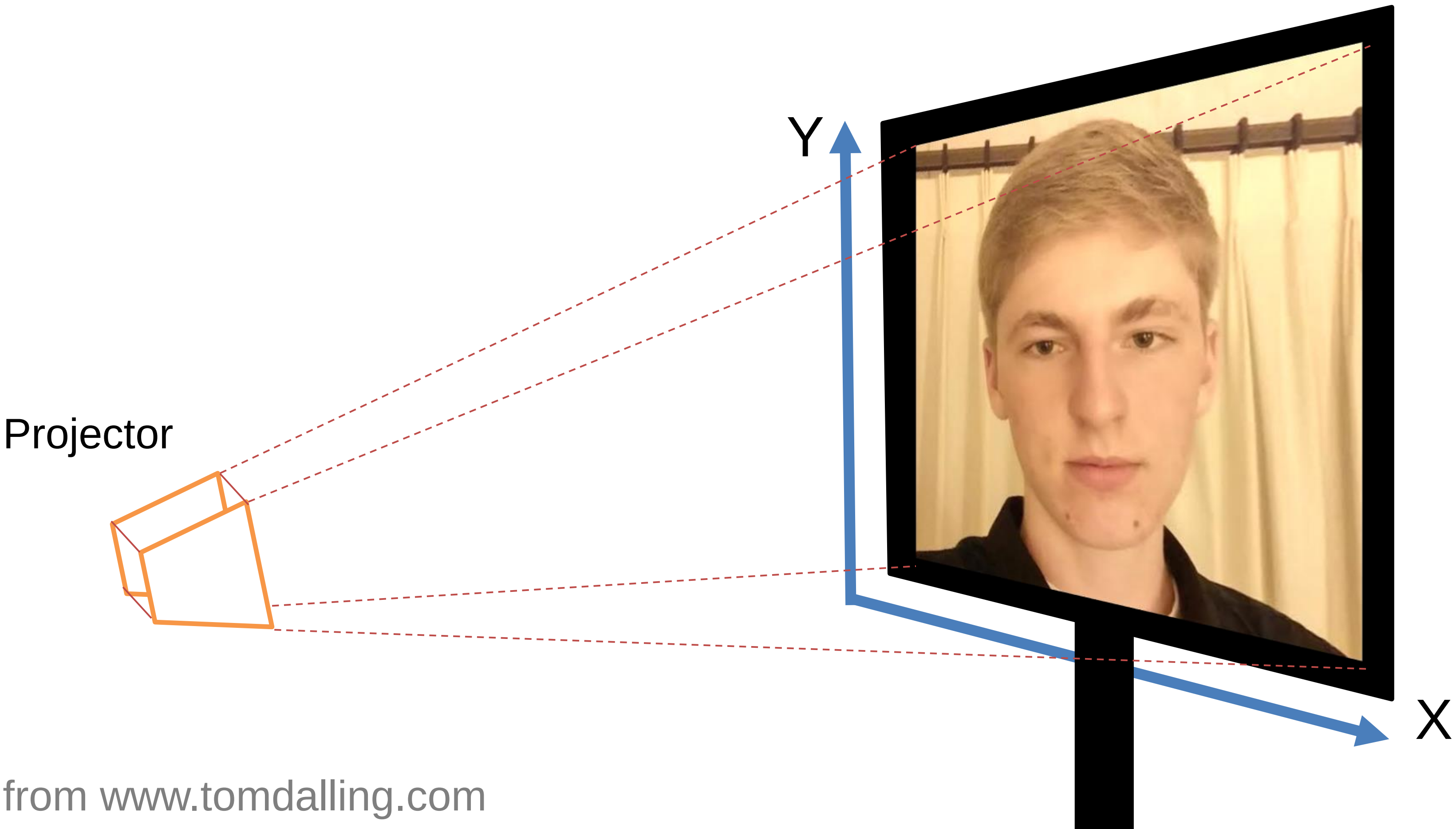
Demo – Kyle Simek

- "Dissecting the Camera Matrix"
- Three-part blog series
- http://ksimek.github.io/2012/08/14/decompose/
- http://ksimek.github.io/2012/08/22/extrinsic/
- http://ksimek.github.io/2013/08/13/intrinsic/

- "Perspective toy"
- http://ksimek.github.io/perspective_camera_toy.html

Projective geometry

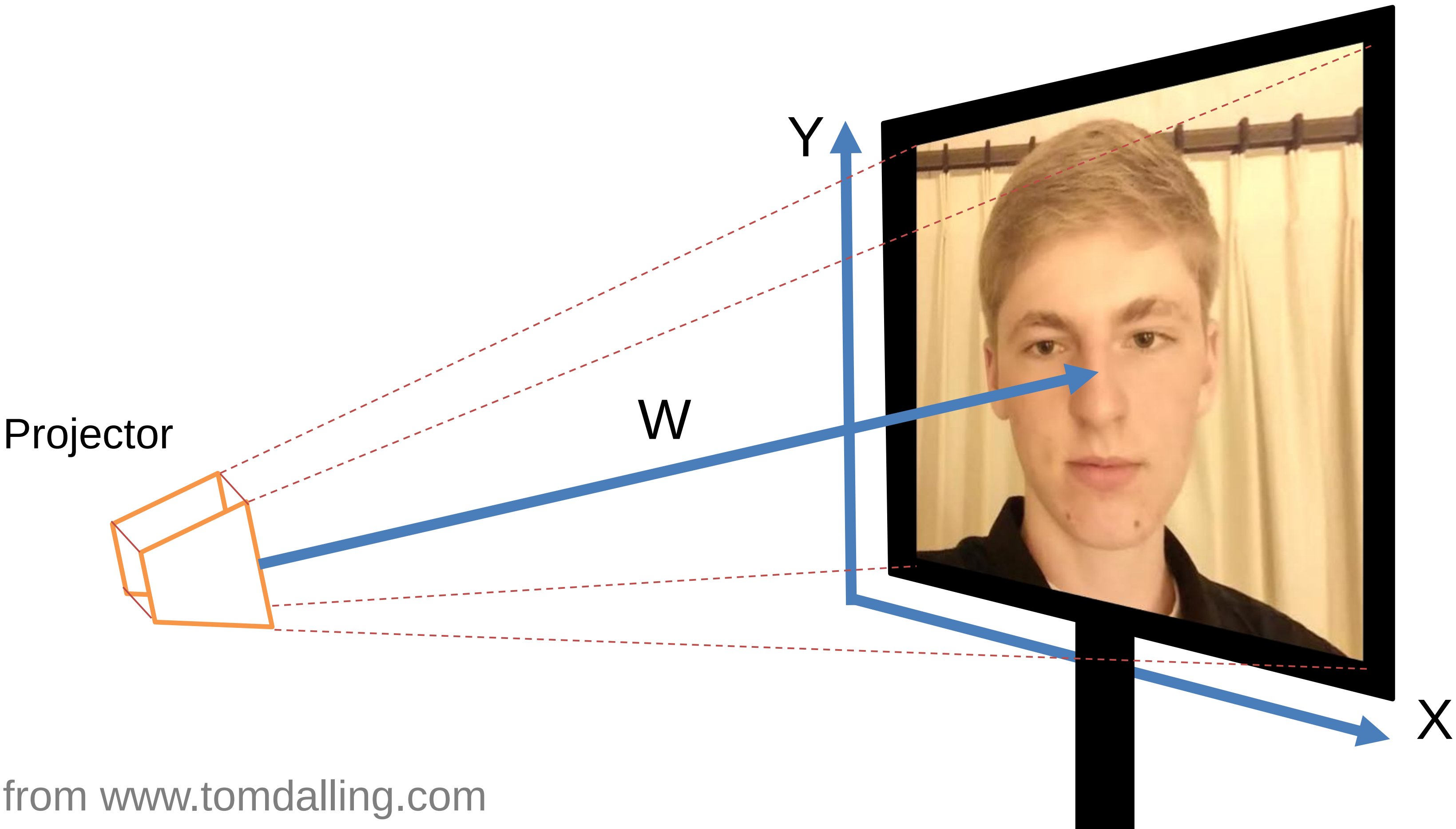
- 2D point in cartesian = (x,y) coordinates
- 2D point in projective = (x,y,w) coordinates



Idea from www.tomdalling.com

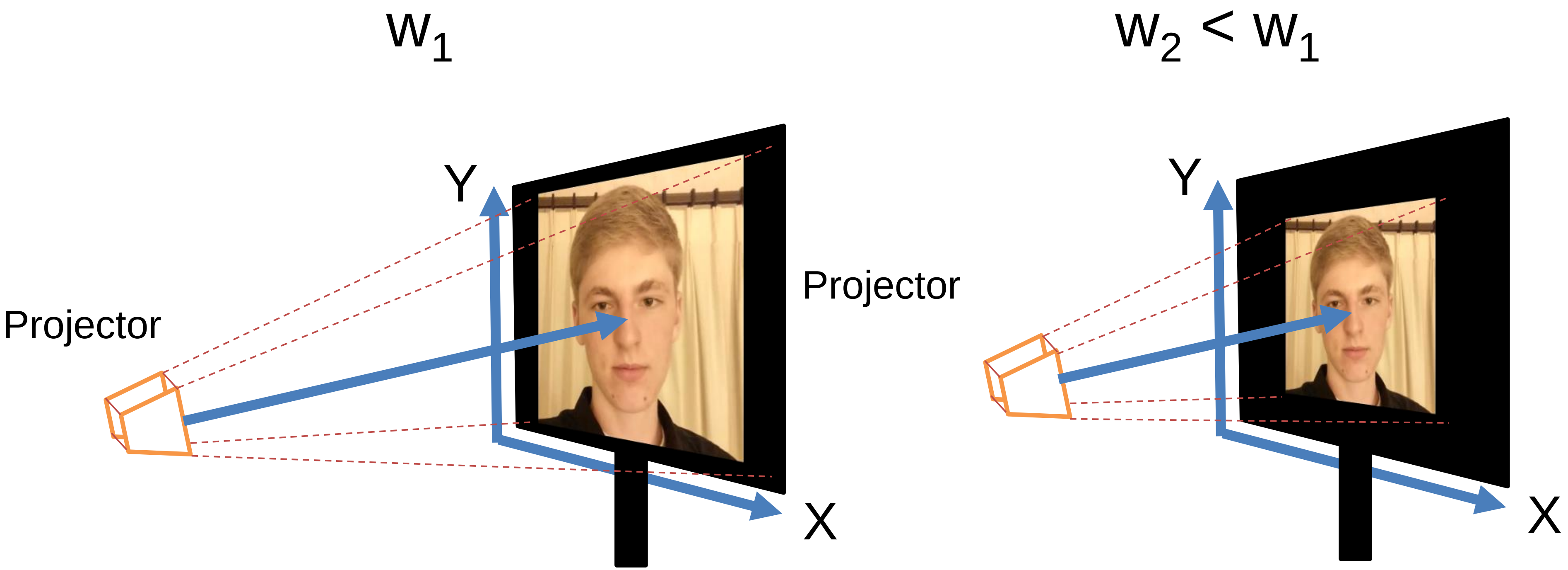
Projective geometry

- 2D point in cartesian = (x,y) coordinates
- 2D point in projective = (x,y,w) coordinates



Idea from www.tomdalling.com

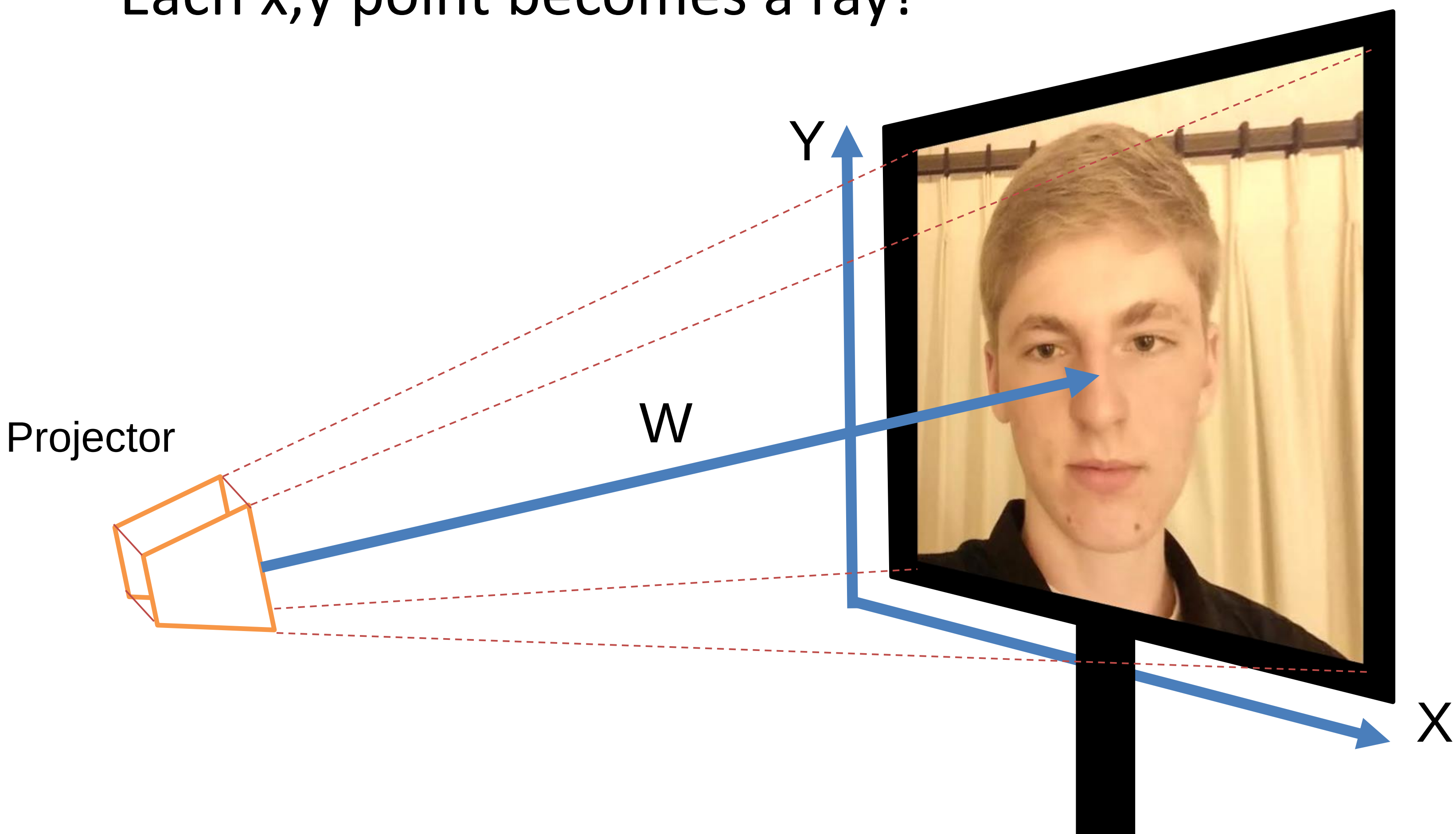
Varying w



Projected image becomes smaller.

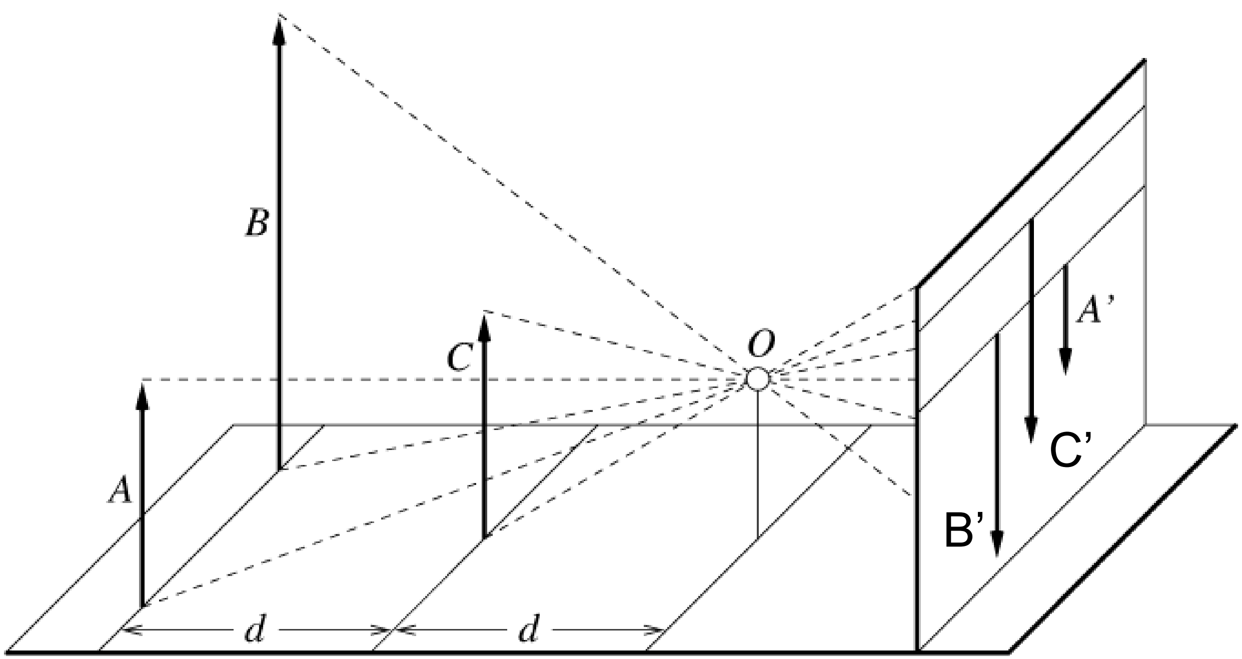
Projective geometry

- 2D point in projective = (x,y,w) coordinates
  - w defines the scale of the projected image.
  - Each x,y point becomes a ray!



Projective geometry

- In 3D, point (x,y,z) becomes (x,y,z,w)
- Perspective is w varying with z:
  - Objects far away are appear smaller



Homogeneous coordinates

Converting *to* homogeneous coordinates

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

2D (image) coordinates

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

3D (scene) coordinates

Converting *from* homogeneous coordinates

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

2D (image) coordinates

$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \Rightarrow (x/w, y/w, z/w)$$

3D (scene) coordinates

Homogeneous coordinates

Scale invariance in projection space

$$k\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} kx \\ ky \\ kw \end{bmatrix} \Rightarrow \begin{bmatrix} \frac{kx}{kw} \\ \frac{ky}{kw} \end{bmatrix} = \begin{bmatrix} \frac{x}{w} \\ \frac{y}{w} \end{bmatrix}$$

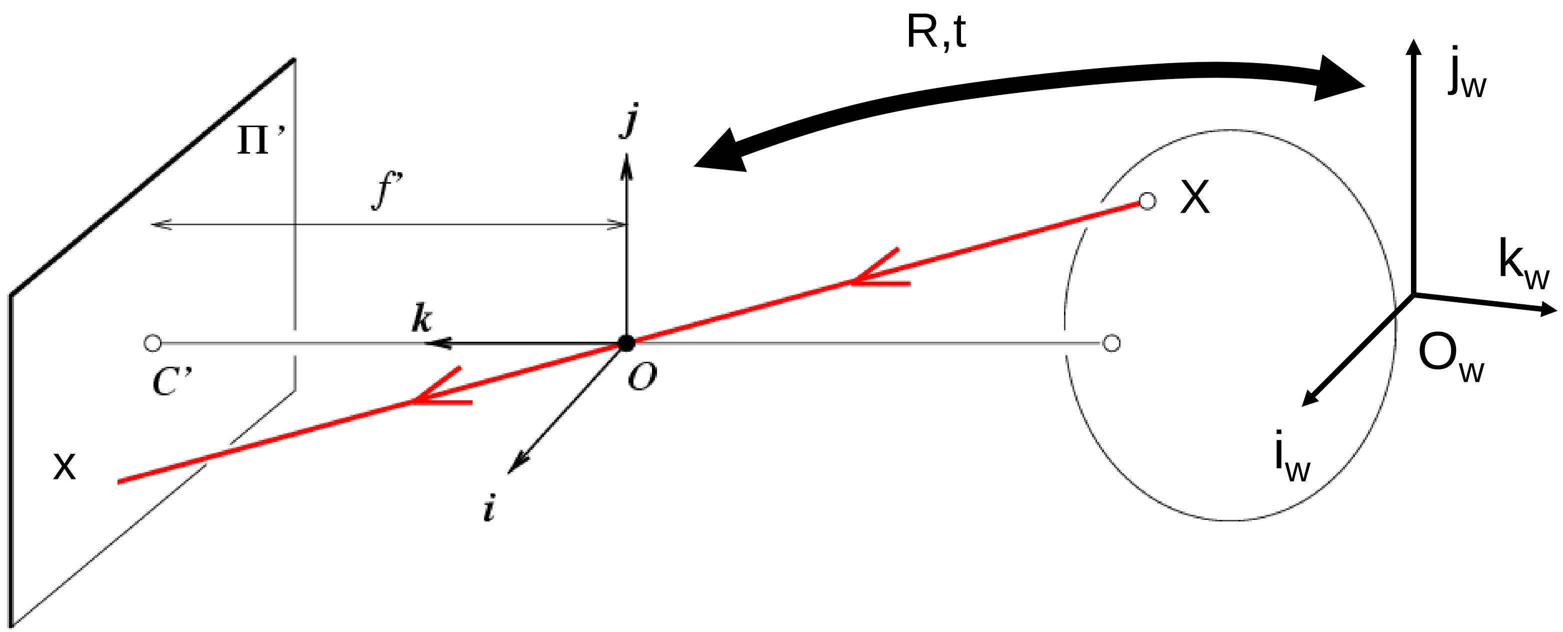
Homogeneous Coordinates

Cartesian Coordinates

E.G., we can uniformly scale the projective space, and it will still produce the same image -> *scale ambiguity*

Slide Credit: Savarese

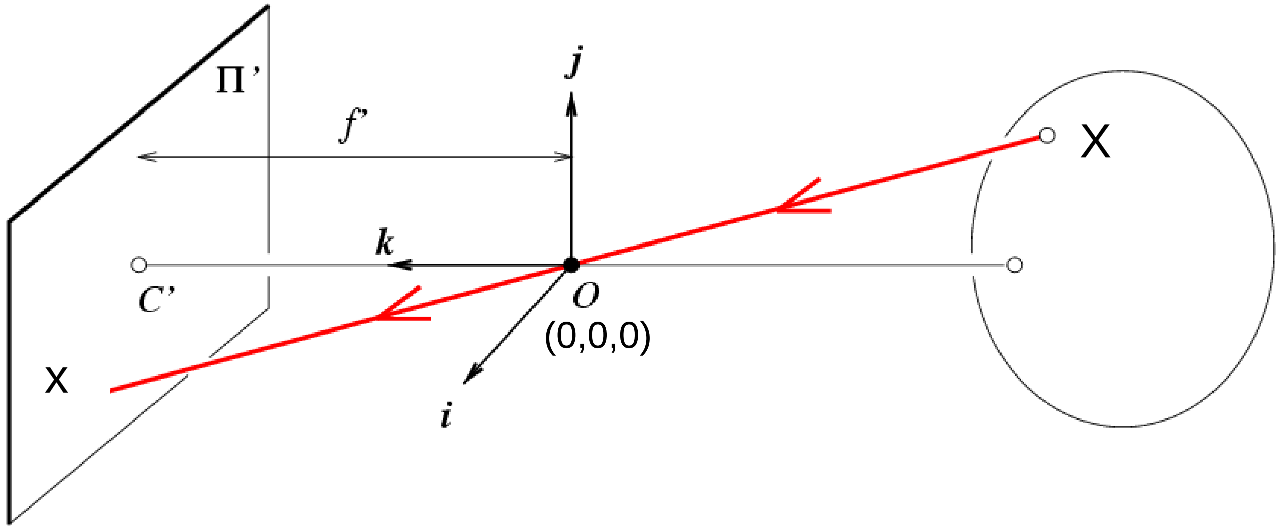
Camera (projection) matrix



$$\mathbf{x} = \mathbf{K}\underbrace{\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}}_{\text{Extrinsic Matrix}}\mathbf{X}$$

x: Image Coordinates: (u,v,1)
K: Intrinsic Matrix (3x3)
R: Rotation (3x3)
t: Translation (3x1)
X: World Coordinates: (X,Y,Z,1)

Projection matrix



Intrinsic Assumptions
- Unit aspect ratio
- Optical center at (0,0)
- No skew

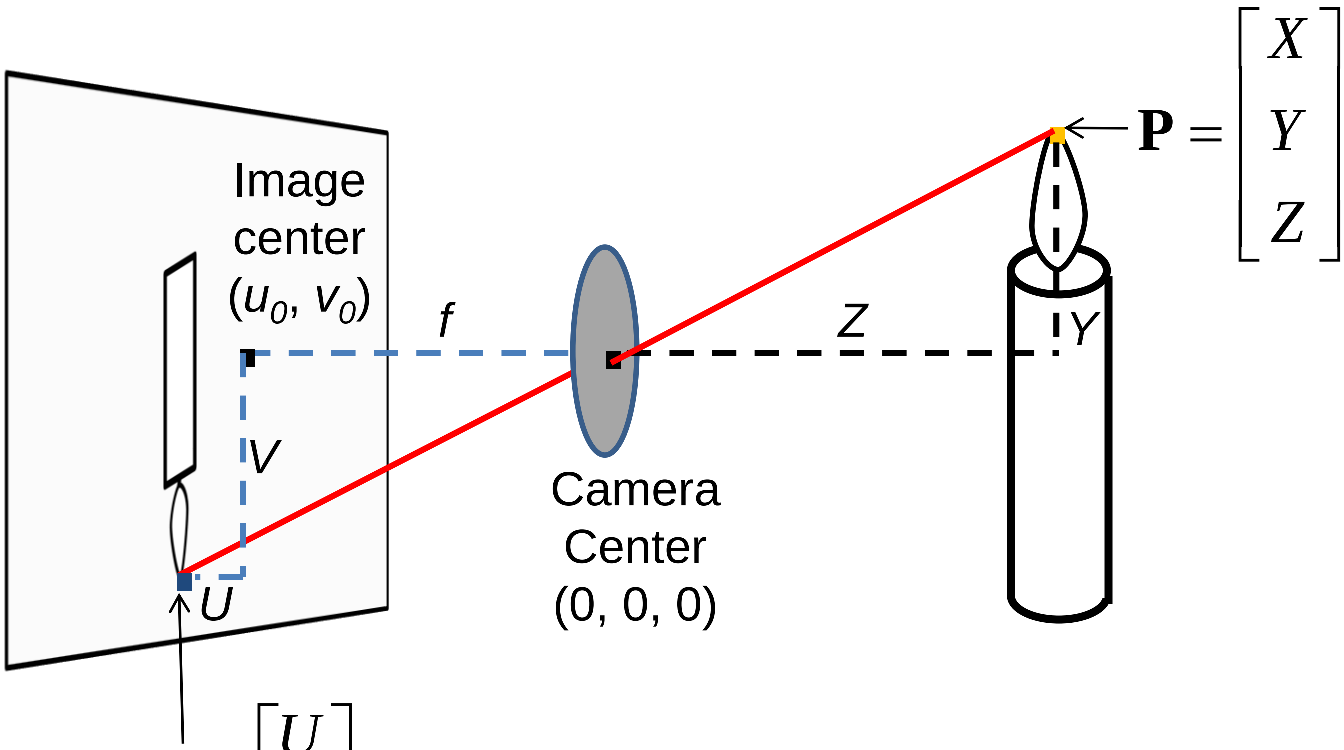
Extrinsic Assumptions
- No rotation
- Camera at (0,0,0)

K

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}\mathbf{X} \Rightarrow w\begin{bmatrix}u \\ v \\ 1\end{bmatrix} = \begin{bmatrix}f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}x \\ y \\ z \\ 1\end{bmatrix}$$

Slide Credit: Savarese

Projection: world coordinates→image coordinates



$$\mathbf{p} = \begin{bmatrix} U \\ V \end{bmatrix}$$

p = distance from
image center

$$U = -X * \frac{f}{Z} \qquad V = -Y * \frac{f}{Z}$$

Remove assumption: known optical center

Intrinsic Assumptions

- Unit aspect ratio
- No skew

Extrinsic Assumptions

- No rotation
- Camera at (0,0,0)

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}\mathbf{X} \Rightarrow w\begin{bmatrix}u\\v\\1\end{bmatrix} = \begin{bmatrix}\overbrace{\boxed{\begin{matrix}f & 0 & u_0\\0 & f & v_0\\0 & 0 & 1\end{matrix}}}^{\mathbf{K}} & \begin{matrix}0\\0\\0\end{matrix}\end{bmatrix}\begin{bmatrix}x\\y\\z\\1\end{bmatrix}$$

Remove assumption: equal aspect ratio

Intrinsic Assumptions
- No skew

Extrinsic Assumptions
- No rotation
- Camera at (0,0,0)

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}\mathbf{X} \Rightarrow w\begin{bmatrix}u\\v\\1\end{bmatrix} = \begin{bmatrix}f_x & 0 & u_0 & 0\\0 & f_y & v_0 & 0\\0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}x\\y\\z\\1\end{bmatrix}$$

Remove assumption: non-skewed pixels

Intrinsic Assumptions

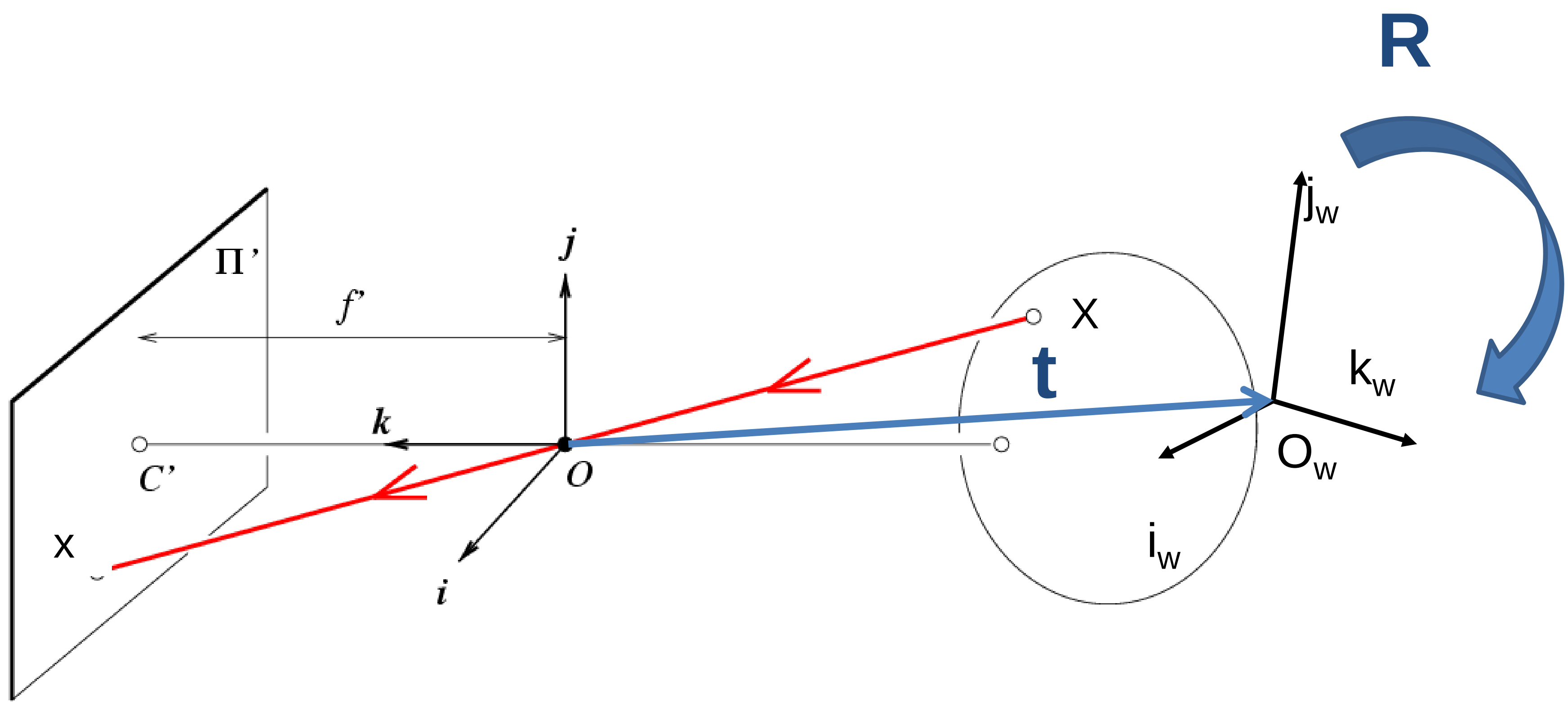
Extrinsic Assumptions
- No rotation
- Camera at (0,0,0)

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}\mathbf{X} \Rightarrow w\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & s & u_0 & 0 \\ 0 & f_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Note: different books use different notation for parameters

James Hays

Oriented and Translated Camera



James Hays

Allow camera translation

Intrinsic Assumptions

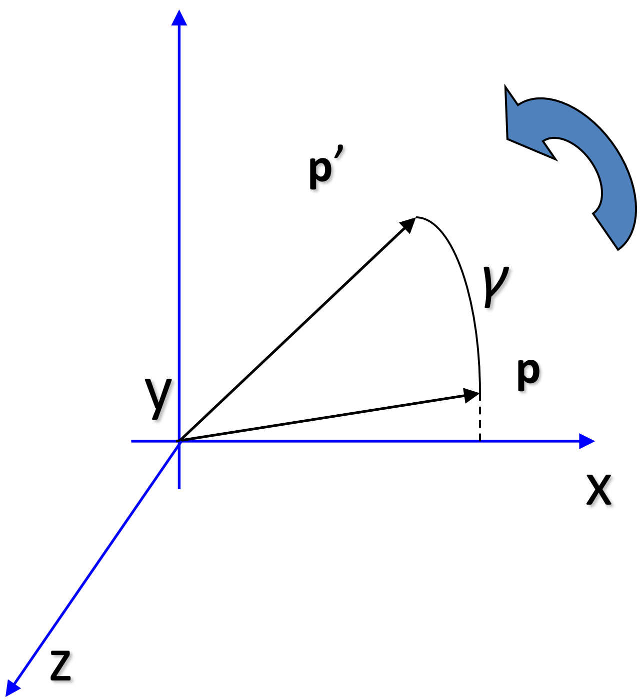
Extrinsic Assumptions
- No rotation

$$\mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{I} & \mathbf{t}\end{bmatrix}\mathbf{X} \Rightarrow w\begin{bmatrix}u\\v\\1\end{bmatrix} = \begin{bmatrix}f_x & s & u_0\\0 & f_y & v_0\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & 0 & t_x\\0 & 1 & 0 & t_y\\0 & 0 & 1 & t_z\end{bmatrix}\begin{bmatrix}x\\y\\z\\1\end{bmatrix}$$

James Hays

3D Rotation of Points

Rotation around the coordinate axes, counter-clockwise:



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

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

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

Allow camera rotation

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

$$w\begin{bmatrix}u\\v\\1\end{bmatrix} = \begin{bmatrix}f_x & s & u_0\\0 & f_y & v_0\\0 & 0 & 1\end{bmatrix}\begin{bmatrix}r_{11} & r_{12} & r_{13} & t_x\\r_{21} & r_{22} & r_{23} & t_y\\r_{31} & r_{32} & r_{33} & t_z\end{bmatrix}\begin{bmatrix}x\\y\\z\\1\end{bmatrix}$$

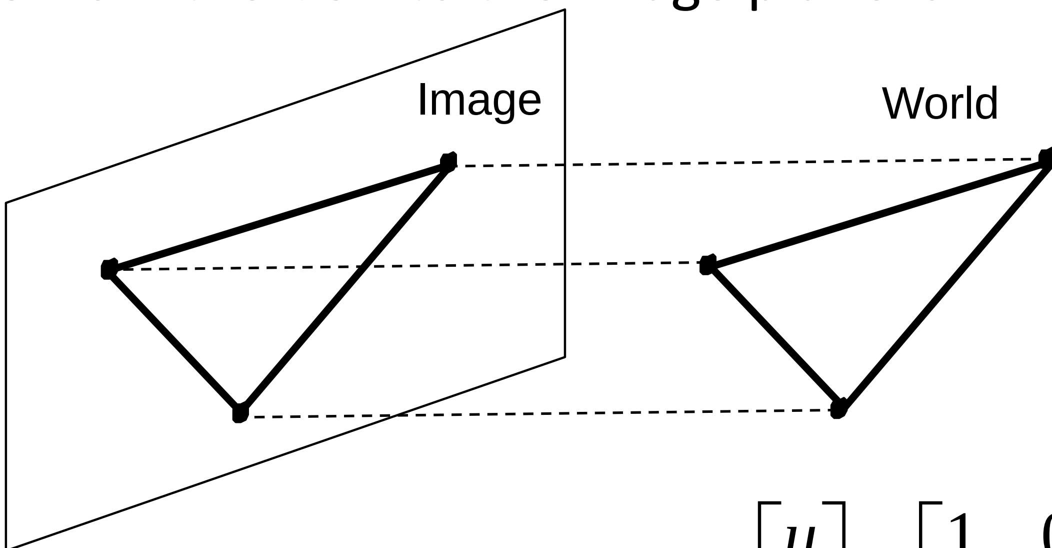
Demo – Kyle Simek

- “Dissecting the Camera Matrix”
- Three-part blog series
- http://ksimek.github.io/2012/08/14/decompose/
- http://ksimek.github.io/2012/08/22/extrinsic/
- http://ksimek.github.io/2013/08/13/intrinsic/

- “Perspective toy”
- http://ksimek.github.io/perspective_camera_toy.html

Orthographic Projection

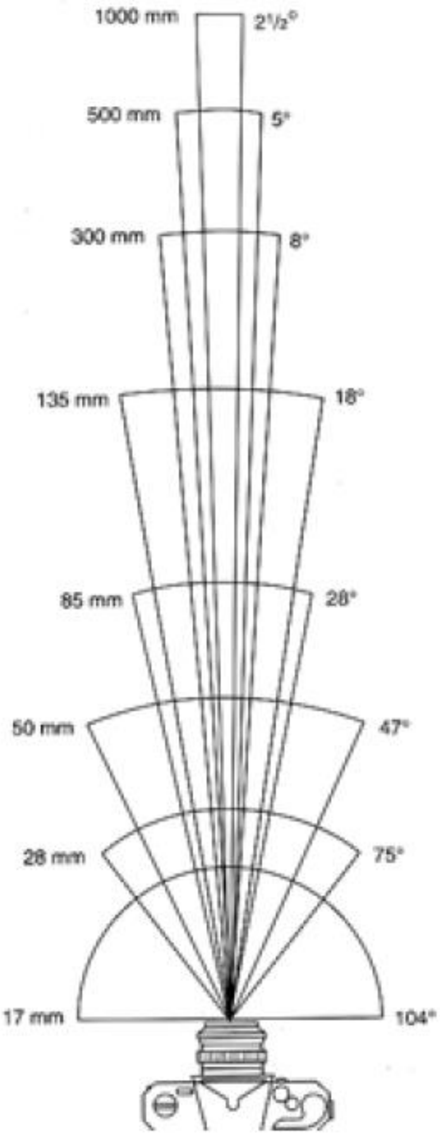
- Special case of perspective projection
  - Distance from the COP to the image plane is infinite

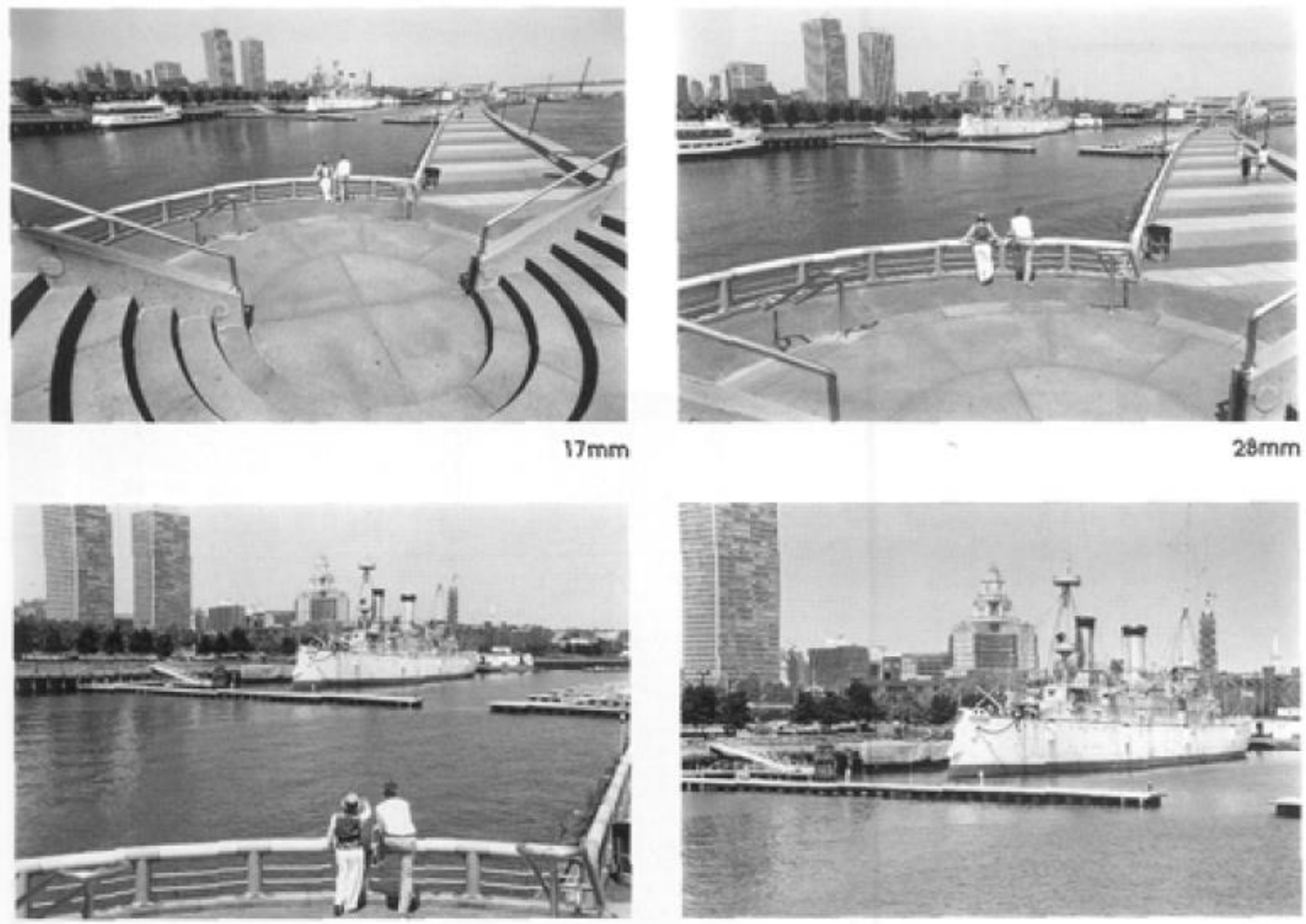


  - Also called “parallel projection”
  - What’s the projection matrix?

$$w\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Field of View (Zoom, focal length)




17mm

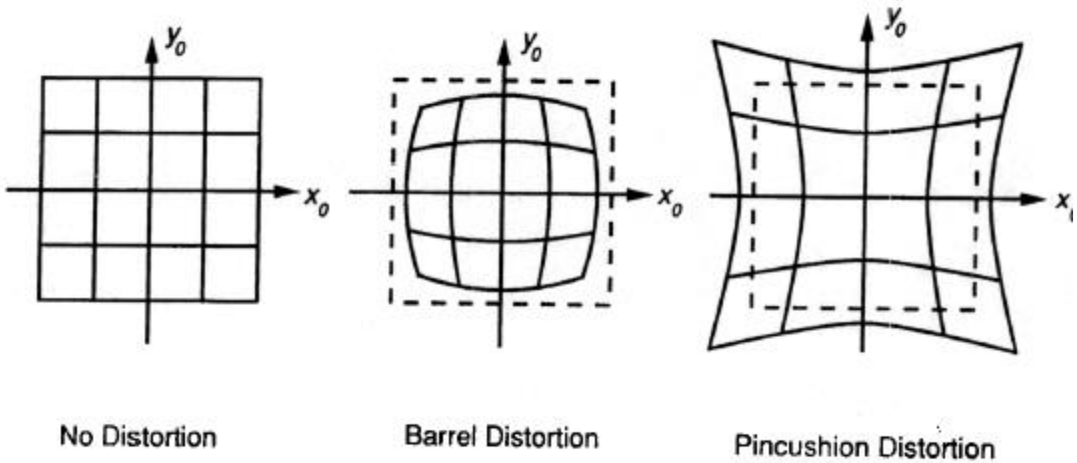
28mm

50mm

85mm

From London and Upton

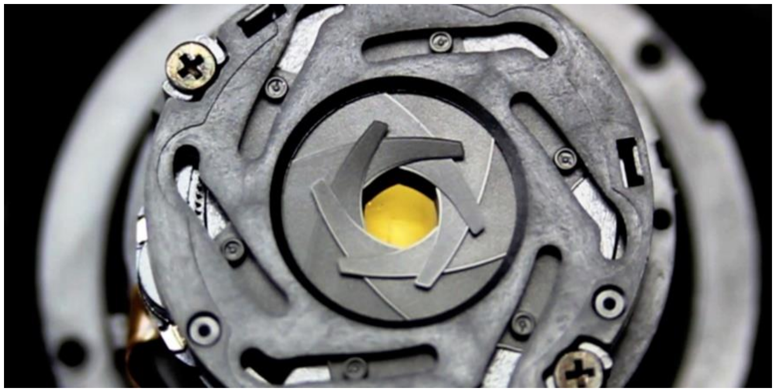
Beyond Pinholes: Radial Distortion

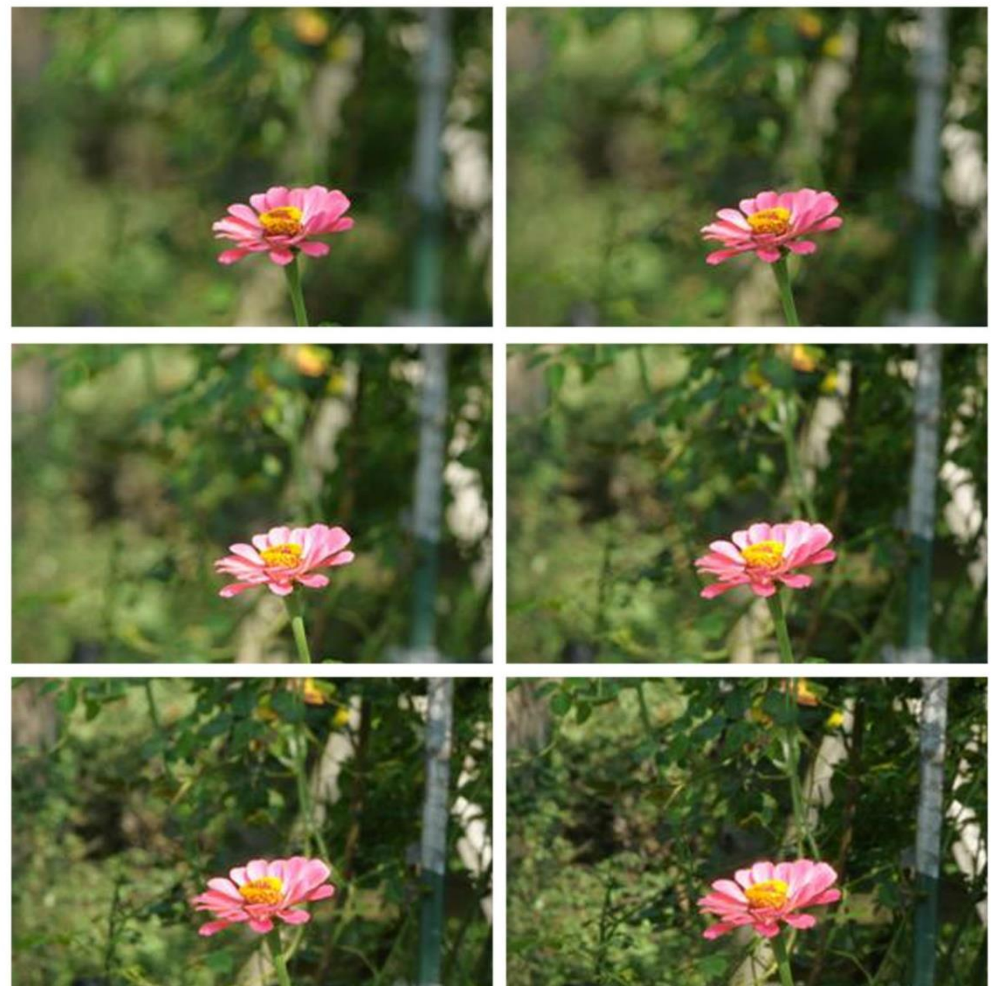


Corrected Barrel Distortion

Image from Martin Habbecke

Beyond Pinholes: Real apertures

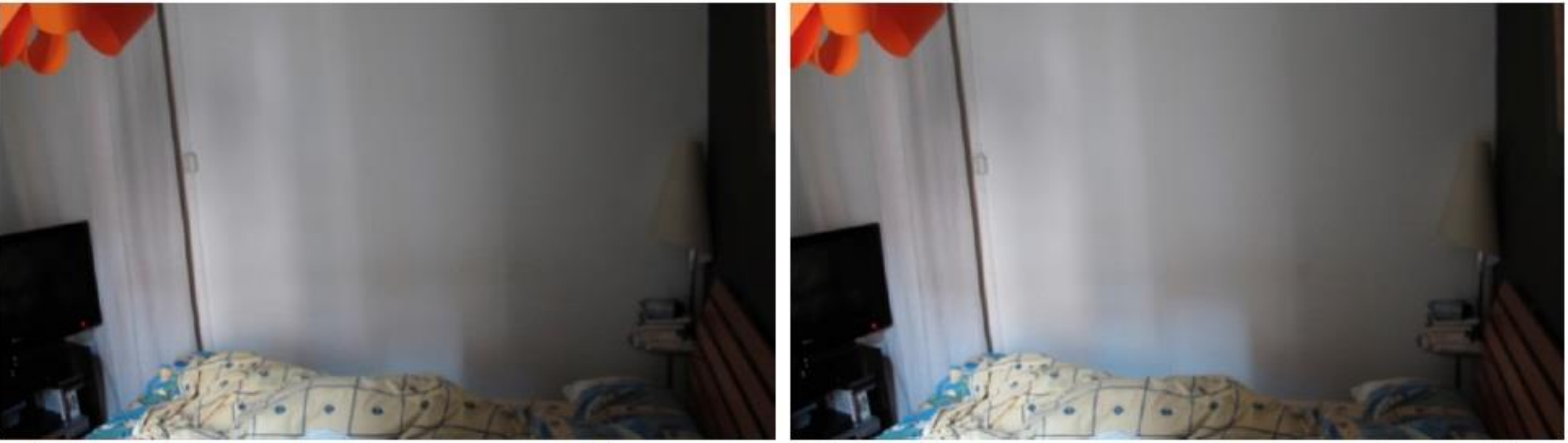




Accidental Cameras

Accidental Pinhole and Pinspeck Cameras
Revealing the scene outside the picture.
Antonio Torralba, William T. Freeman

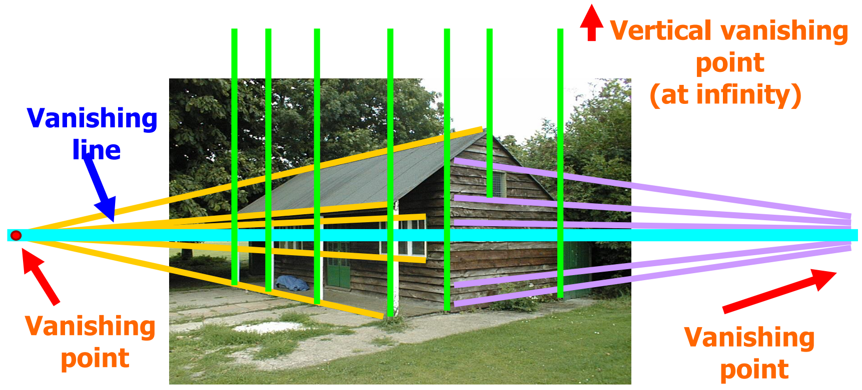
Accidental Cameras



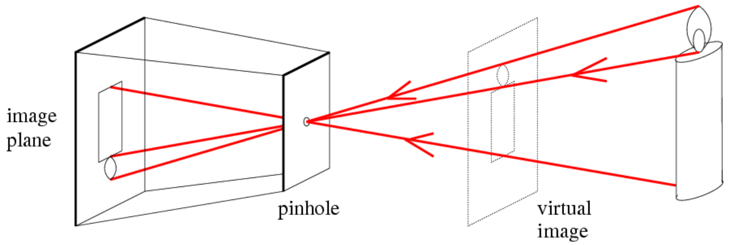
a) Input (occluder present) b) Reference (occluder absent)

c) Difference image (b-a) d) Crop upside down e) True view

Things to remember



- Vanishing points and vanishing lines



- Pinhole camera model and camera projection matrix

$$\mathbf{x} = \mathbf{K}\left[\mathbf{R} \quad \mathbf{t}\right]\mathbf{X}$$

- Homogeneous coordinates

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

James Hays

IS THIS ENOUGH?


Erik Johansson – The Architect