Perception: Camera Modeling

Kostas Daniilidis

Can our aerial robots control their flight like gannets?

How can gannets' eyes and brain estimate distance to water!

Nature Vol. 293 24 September 1981

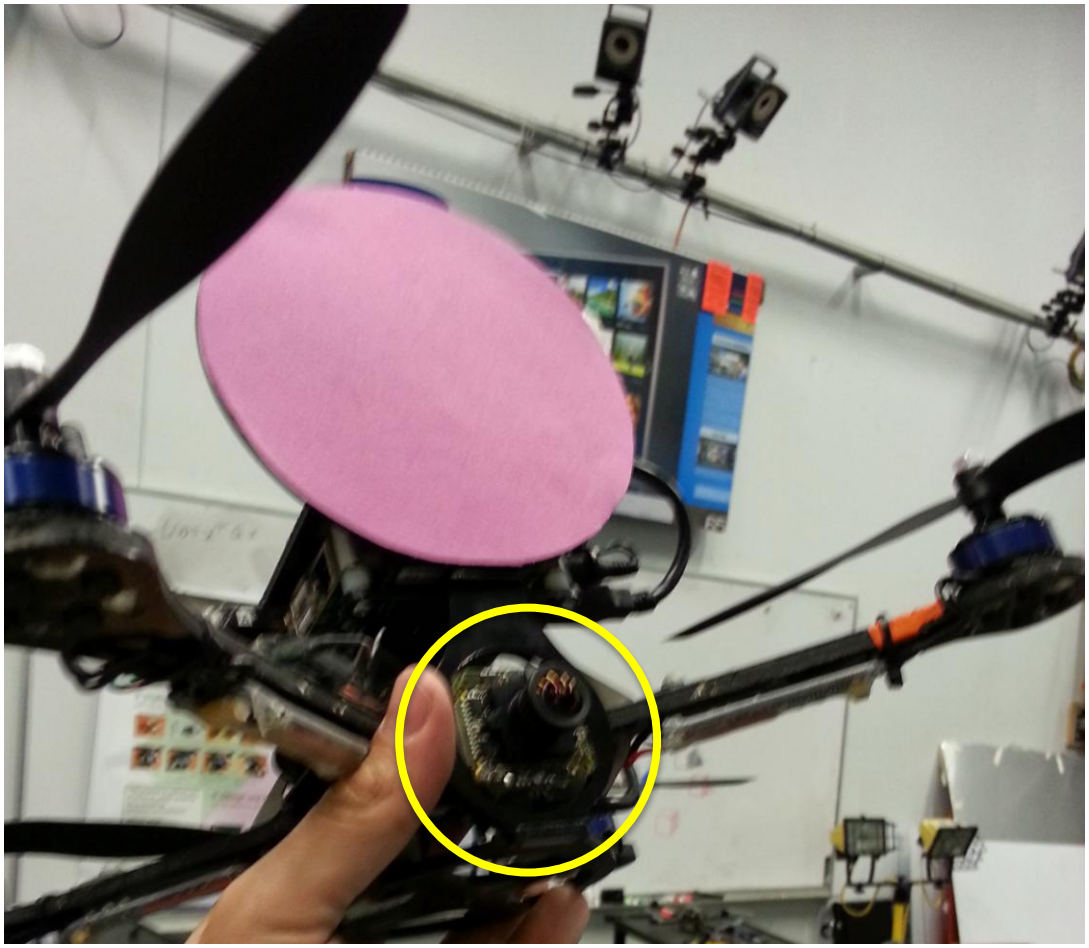
5. Spiess, F. N. *et al. Science* **207,** 1421–1433 (1980).
6. Corliss, J. B. *Science* **203,** 1073–1083 (1979).
7. Edmond, J. M. *et al. Earth planet. Sci. Lett.* **46,** 19–30 (1979).
8. Ruby, E. G., Wirsen, C. O. & Jannasch, H. W. *Appl. envir. Microbiol.* (in the press).
9. Cavanaugh, C. M., Gardiner, S. L., Jones, M. L., Jannasch, H. W. & Waterbury, J. B. *Science* **213,** 340–342 (1981).
10. Felbeck, H. *Science* **213,** 336–338 (1981).
11. Jones, M. L. *Science* **213,** 333–336 (1981).
12. Peck, H. D. Jr *Enzymes* **10,** 651–669 (1974).
13. Latzko, E. & Gibbs, M. *Pl. Physiol.* **44,** 295–300 (1969).
14. Reid, R. G. B. *Can. J. Zool.* **58,** 386–393 (1980).
15. Los Angeles County (California) Sanitation District files.
16. Rau, G. H. *Science* **213,** 338–340 (1981).
17. Rau, G. H. *Nature* **289,** 484–485 (1981).
18. Rau, G. H. & Hedges, J. I. *Science* **203,** 648–649 (1979).
19. Emery, K. O. & Hulsemann, J. *Deep-Sea Res.* **8,** 165–180 (1962).
20. Hartman, O. & Barnard, J. L. *Allan Hancock Pacific Exped.* **22,** (1958).
21. Nicholas, D. J. D., Ferrante, J. V. & Clarke, G. R. *Analyt. Biochem.* **95,** 24–31 (1979).
22. Lonsdale, P. *Nature* **281,** 531–534 (1979).

Plummeting gannets: a paradigm of ecological optics

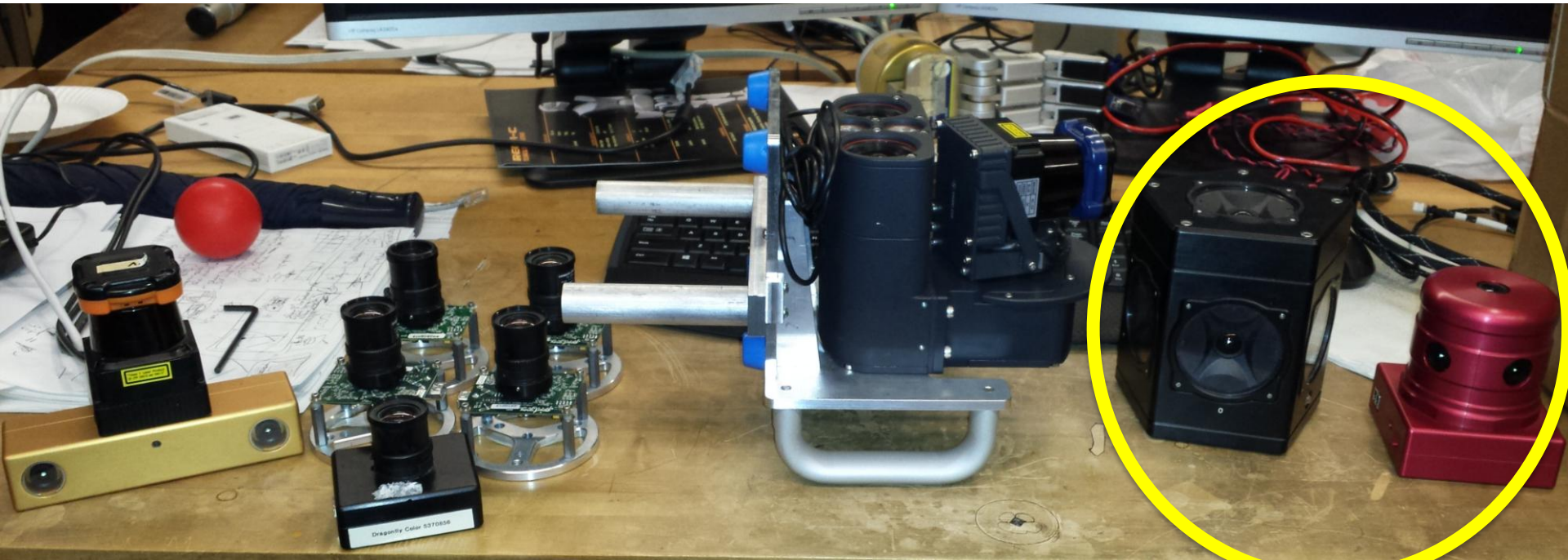
David N. Lee & Paul E. Reddish

Department of Psychology, University of Edinburgh, Edinburgh EH8 9JZ, UK

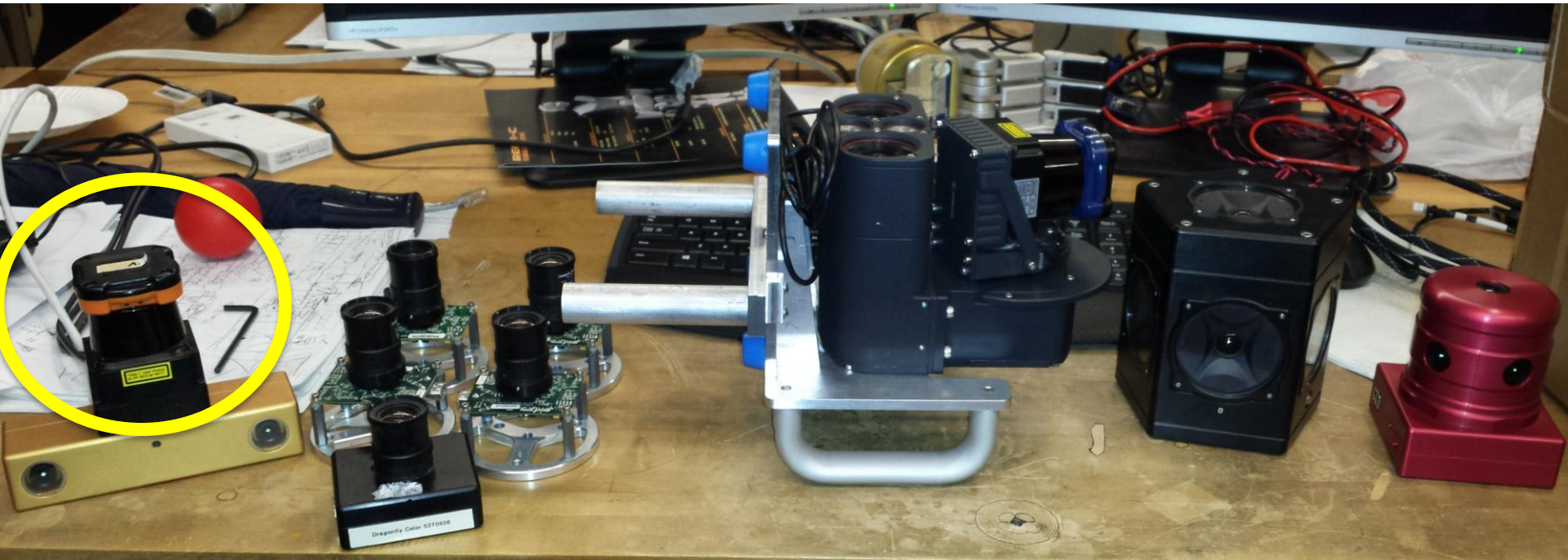
Quadrotor “sees” with a camera



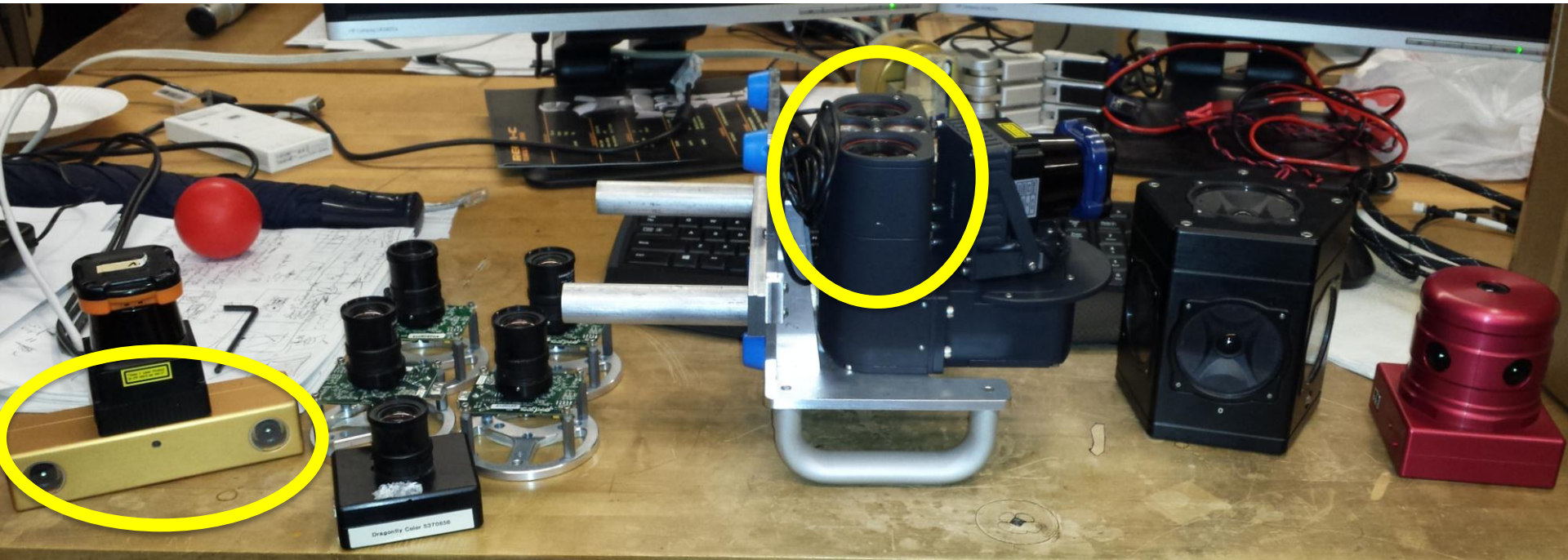
In robotics we use all kinds of cameras!



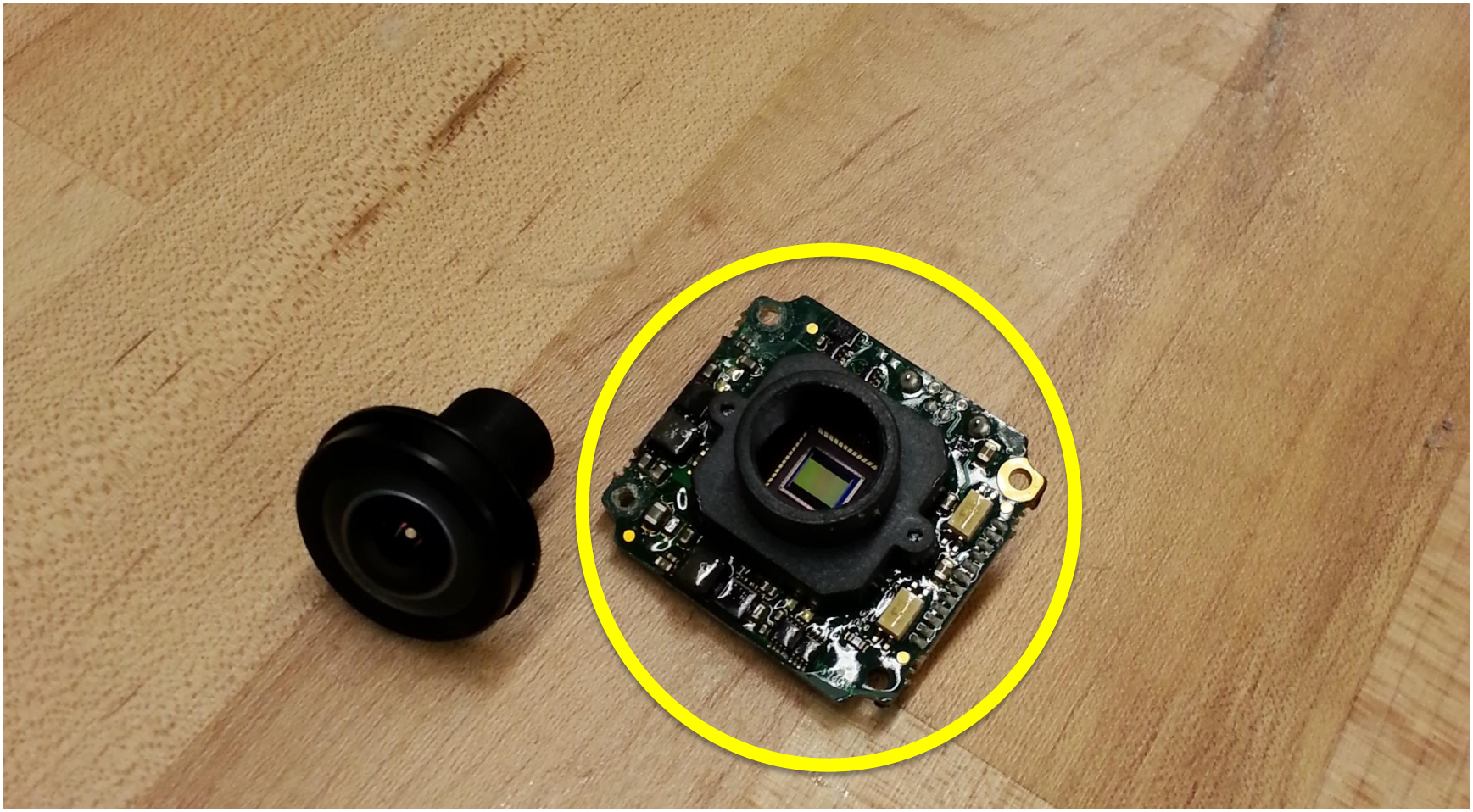
In robotics we use all kinds of cameras!



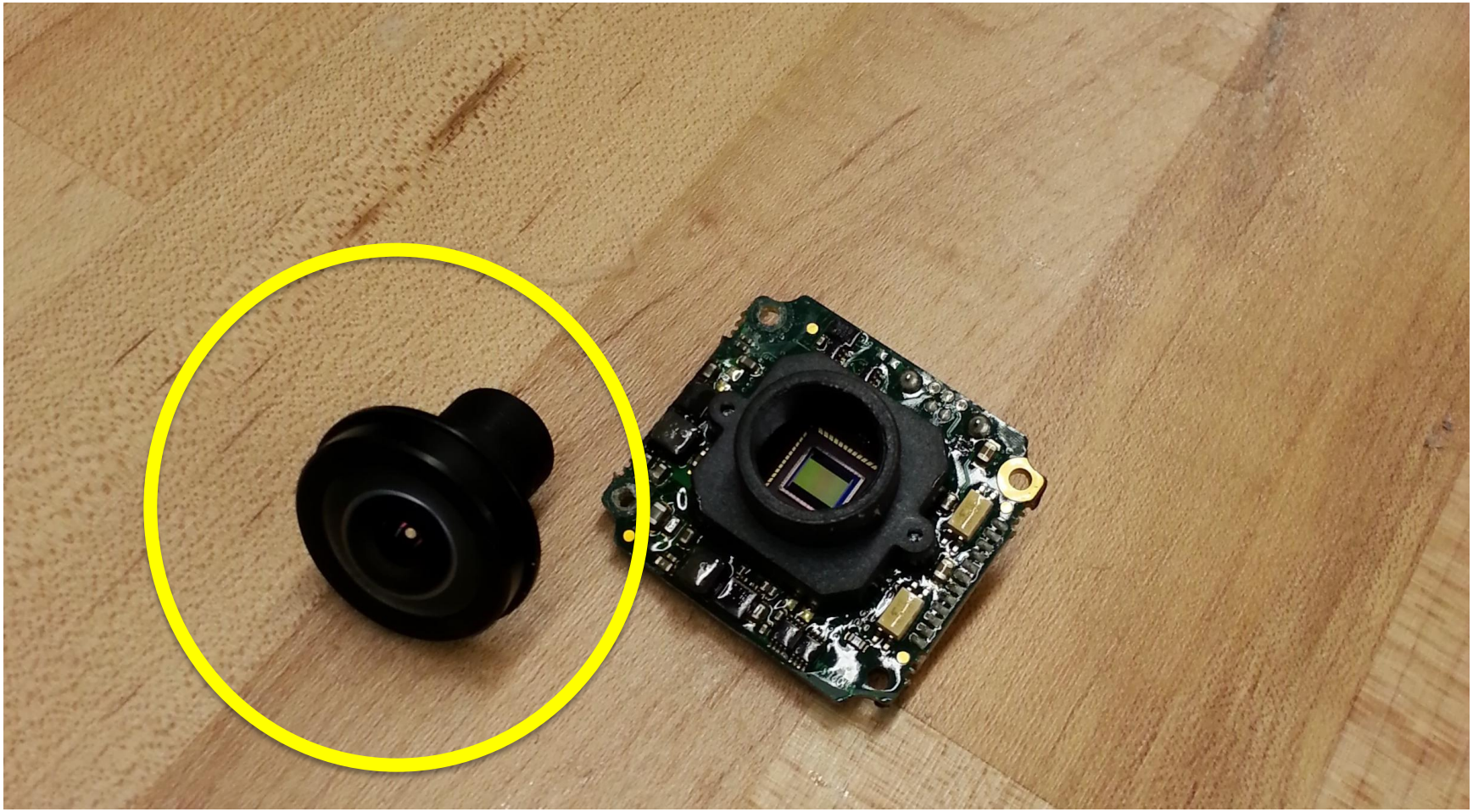
In robotics we use all kinds of cameras!



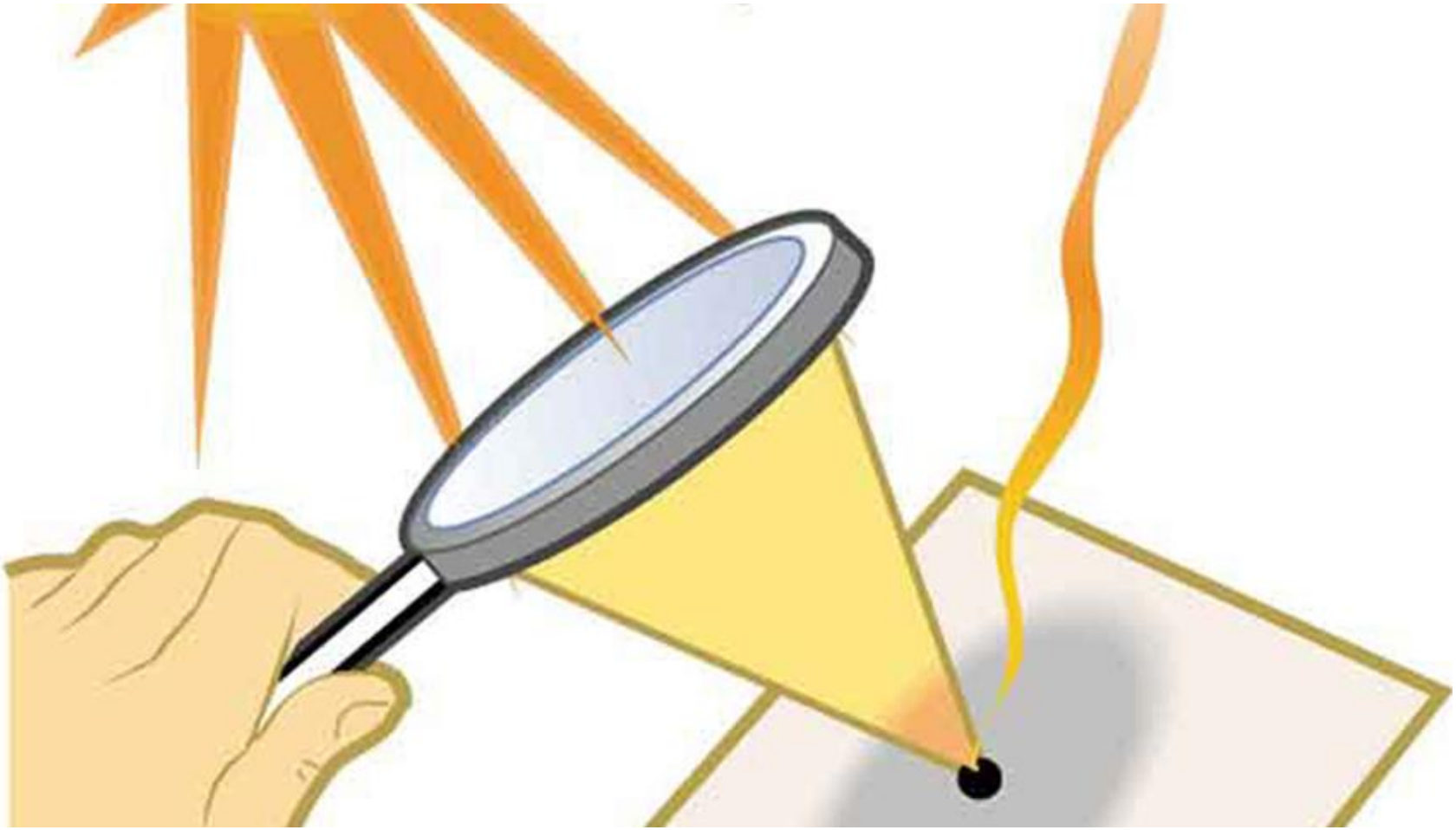
A camera is an imaging chip and a lens



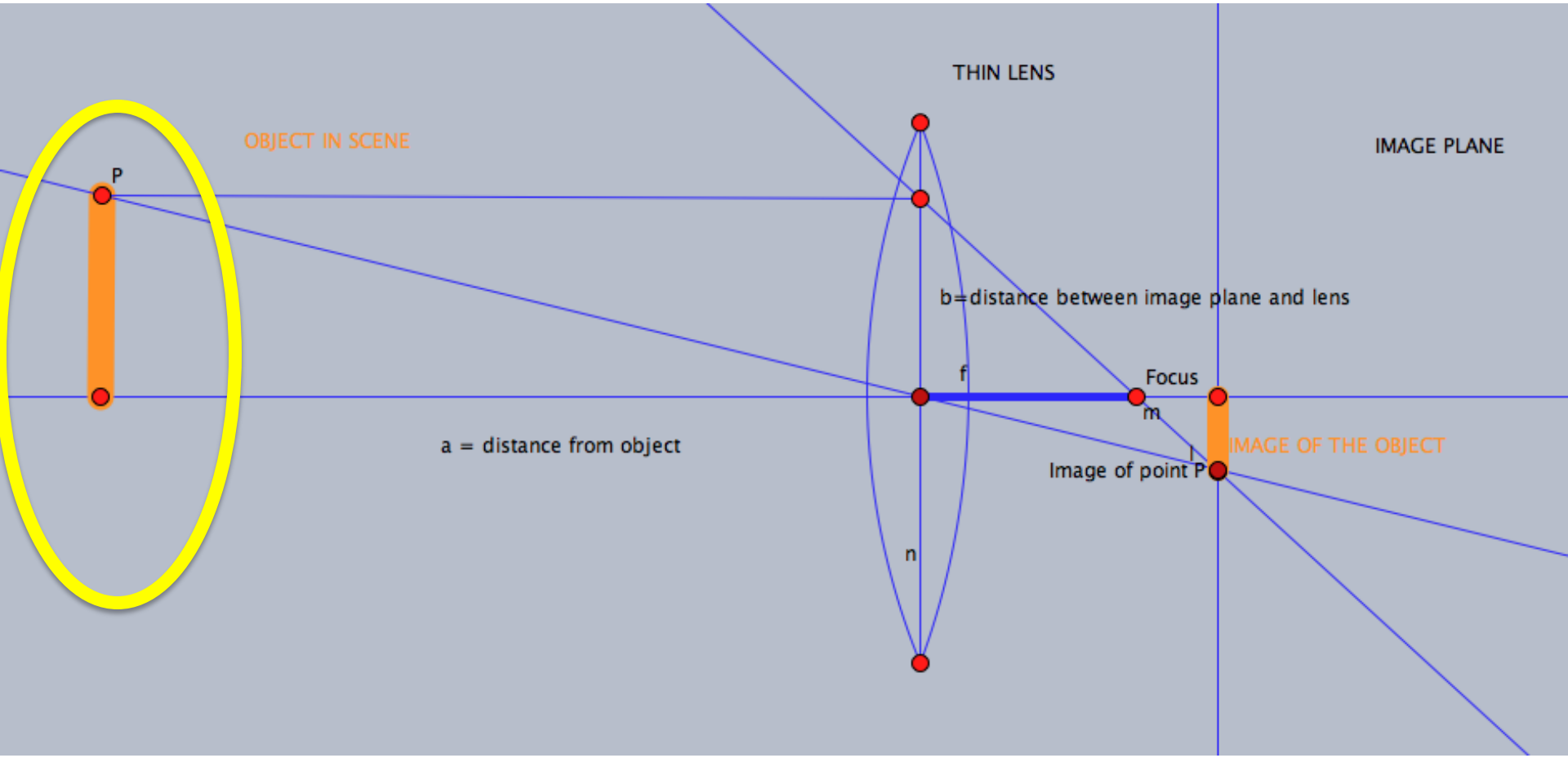
A camera is an imaging chip and a lens



Magnifying glass

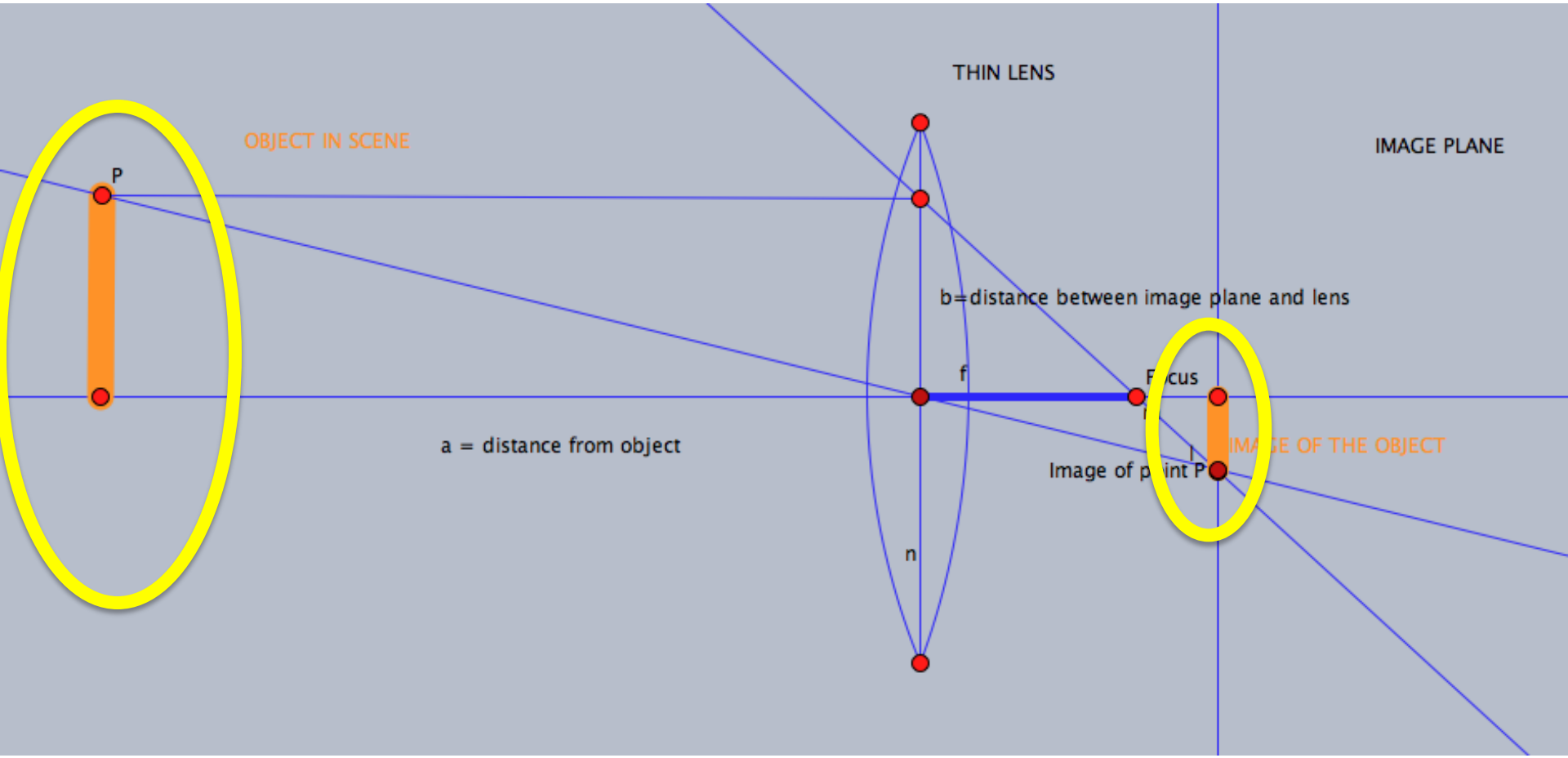


How does a thin lens work



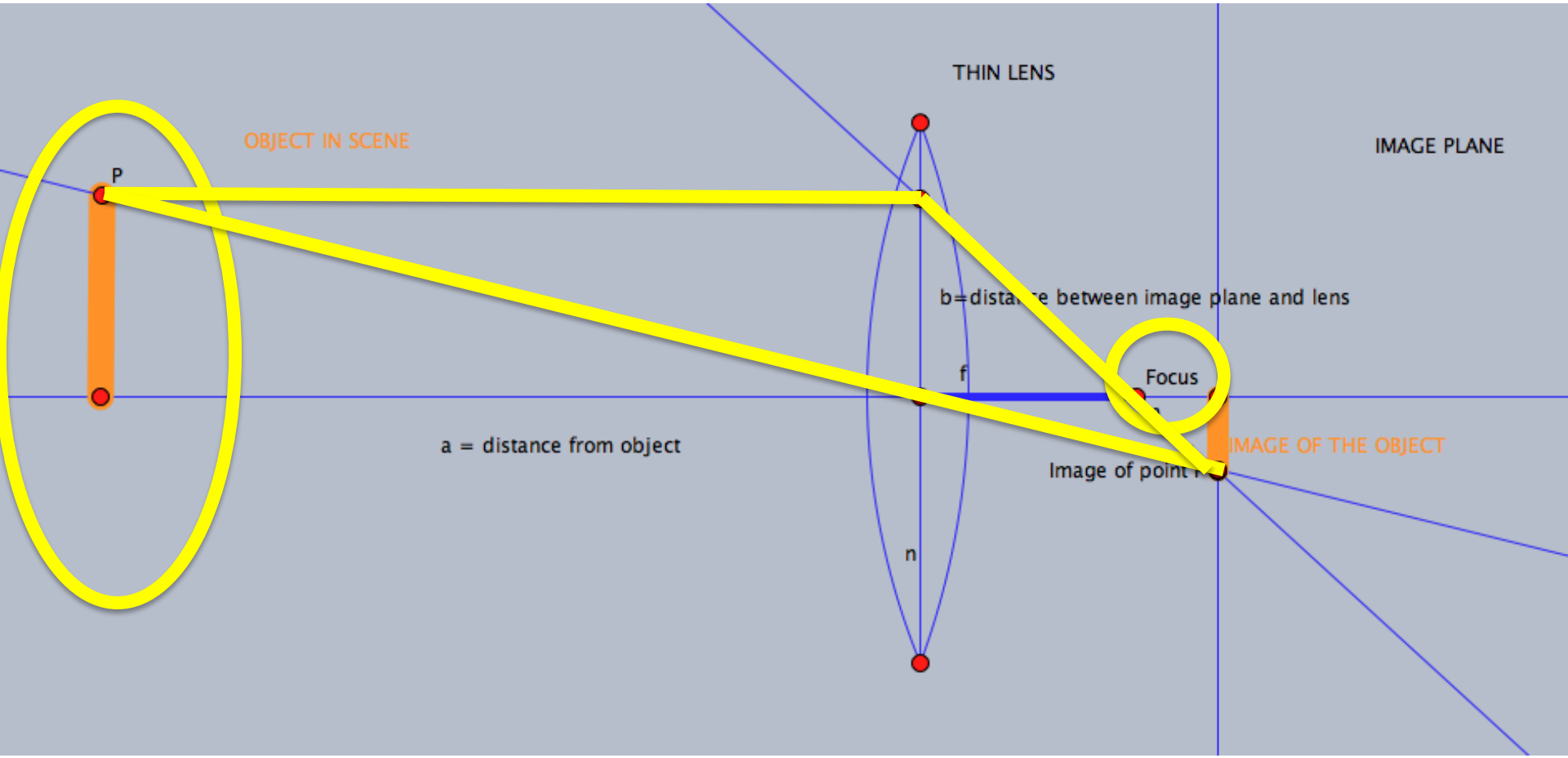
Rays from on object point P converge on a point p on the image plane

How does a thin lens work



Rays from a point P in the scene converge into a point in the image plane

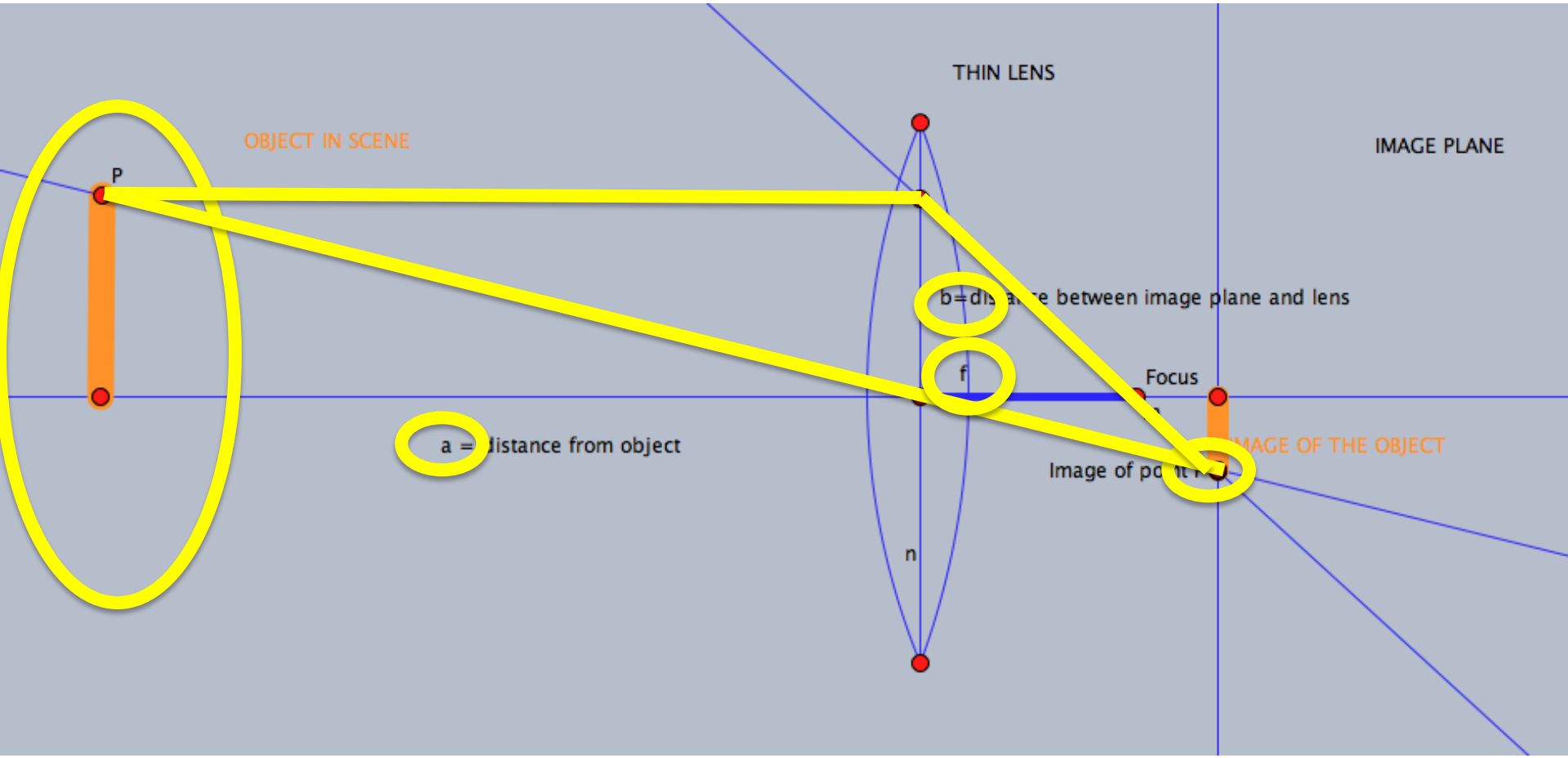
How does a thin lens work



Rays parallel to the optical axis meet the focus after leaving the lens.
Rays through center of the lens do not change direction.

$$\frac{1}{f} = \frac{1}{a} + \frac{1}{b}$$

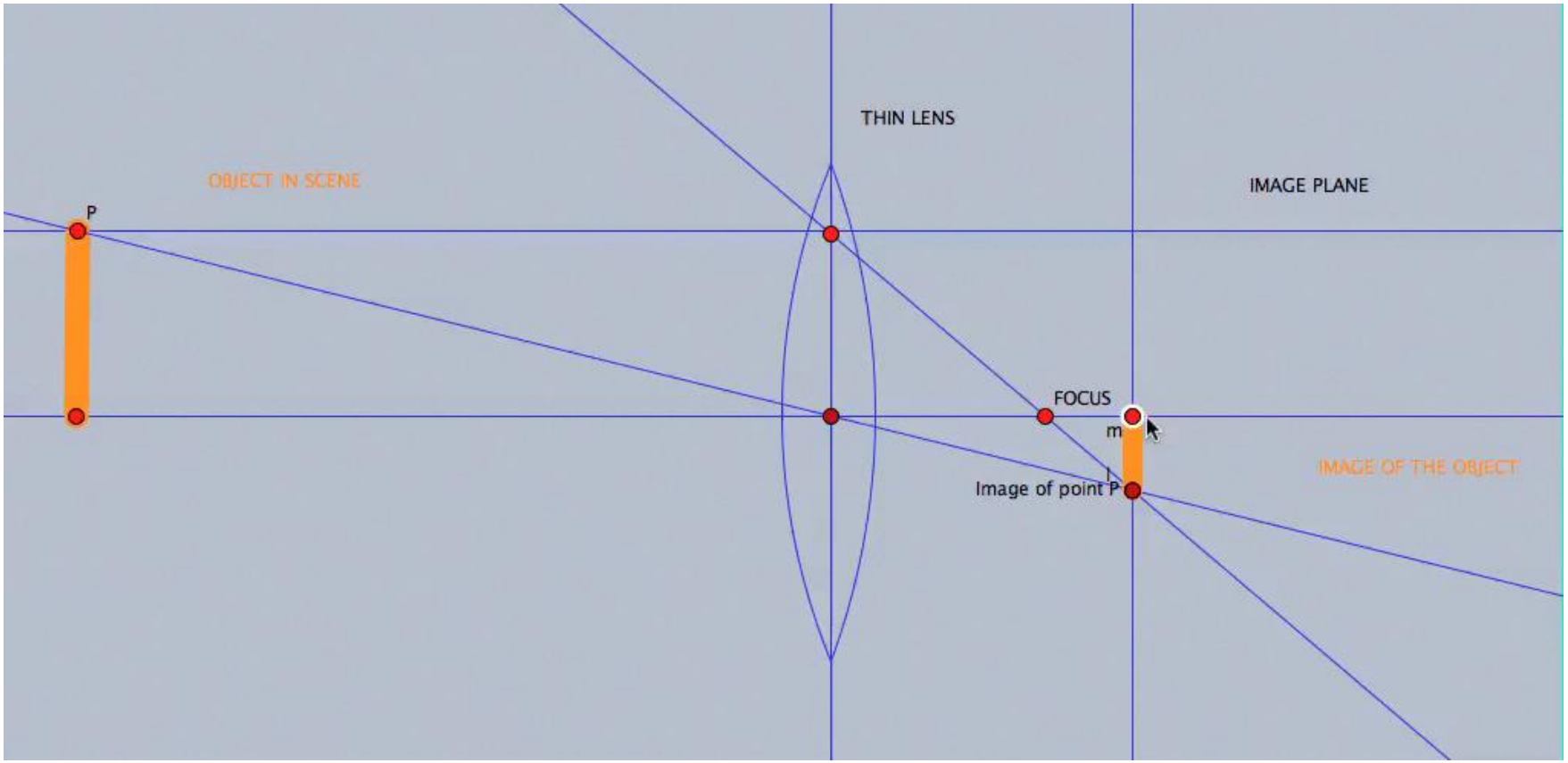
How does a thin lens work



These rays meet at one point if

$$\frac{1}{f} = \frac{1}{a} + \frac{1}{b}$$

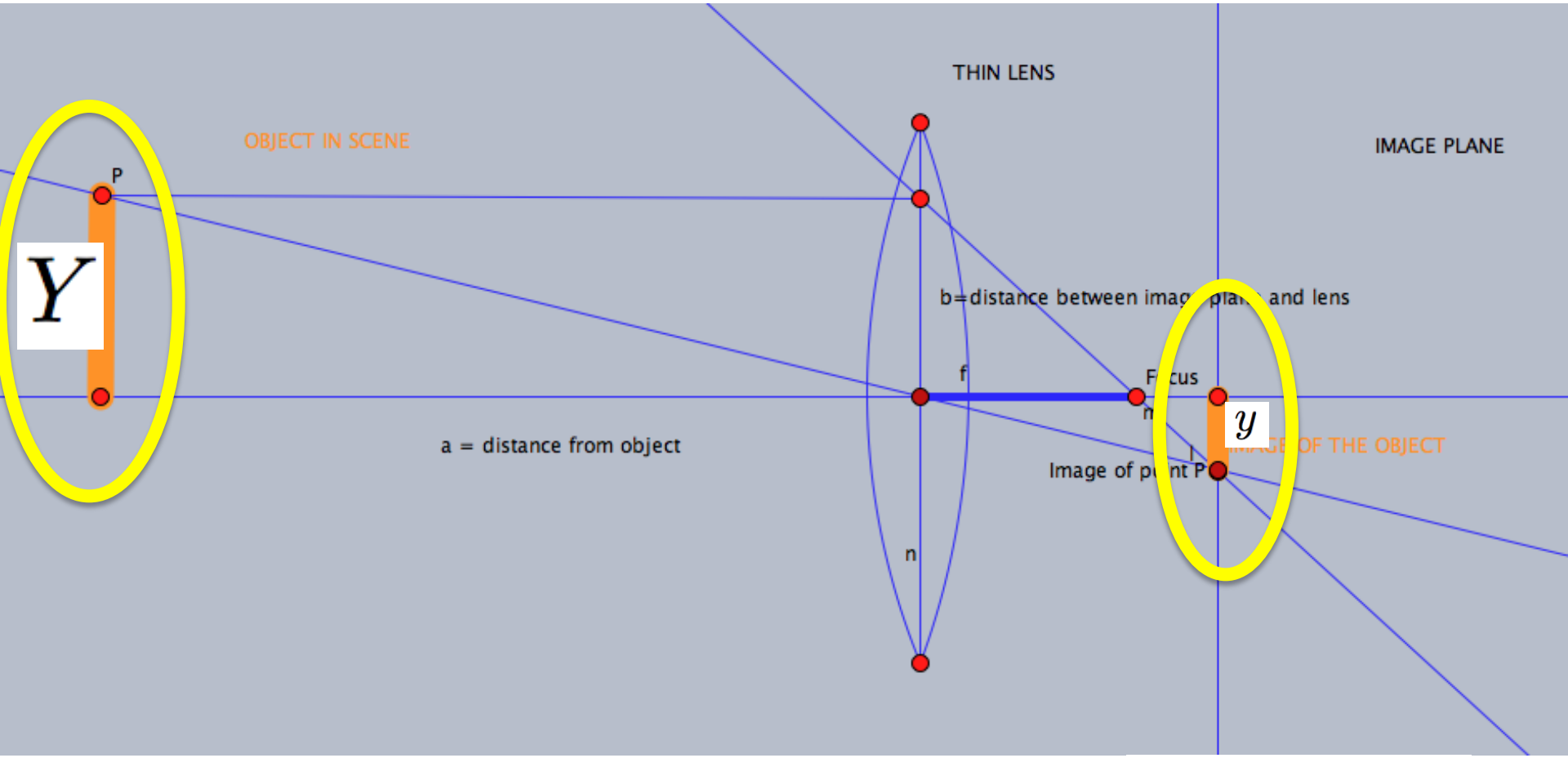
What happens when we move b, the image plane



Moving the image plane is what we call (de-) focusing !
Image starts blurring!

$$\frac{1}{f} \neq \frac{1}{a} + \frac{1}{b}$$

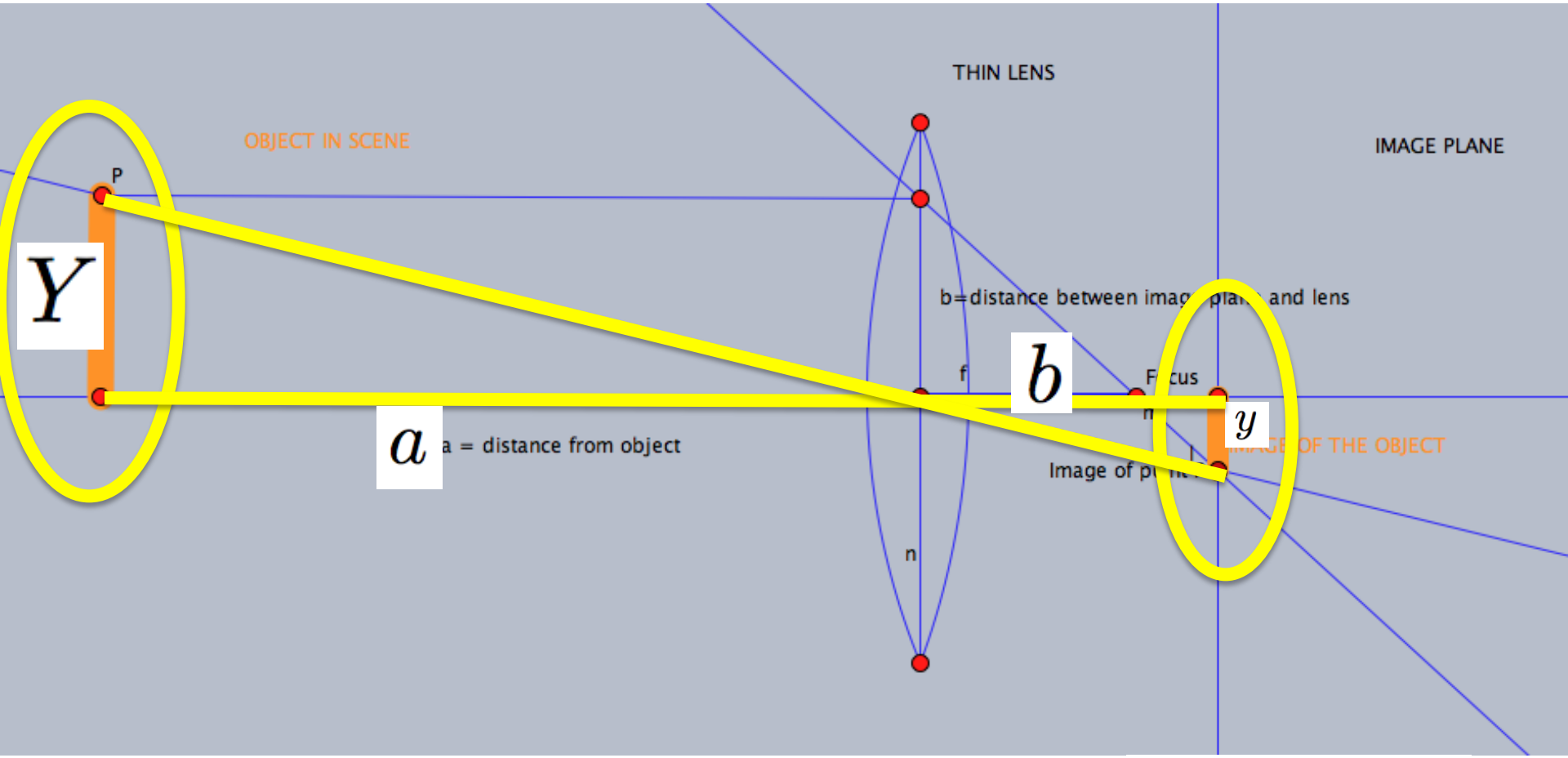
Perspective projection: size of object image



If you look only at the ray going through the center of the lens

$$\frac{Y}{a} = \frac{y}{b}$$

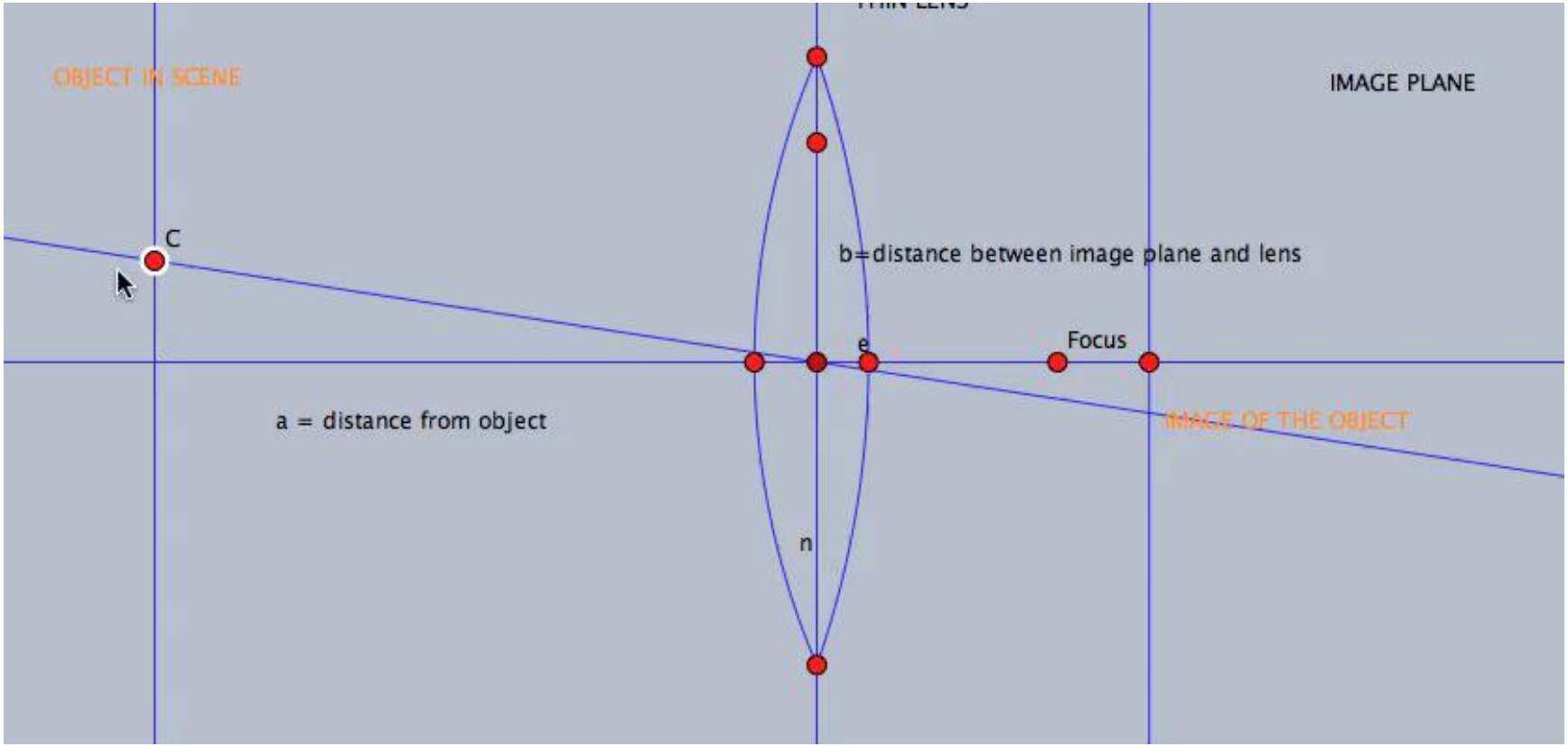
Perspective projection: size of object image



If you look only at the ray going through the center of the lens

$$\frac{Y}{a} = \frac{y}{b}$$

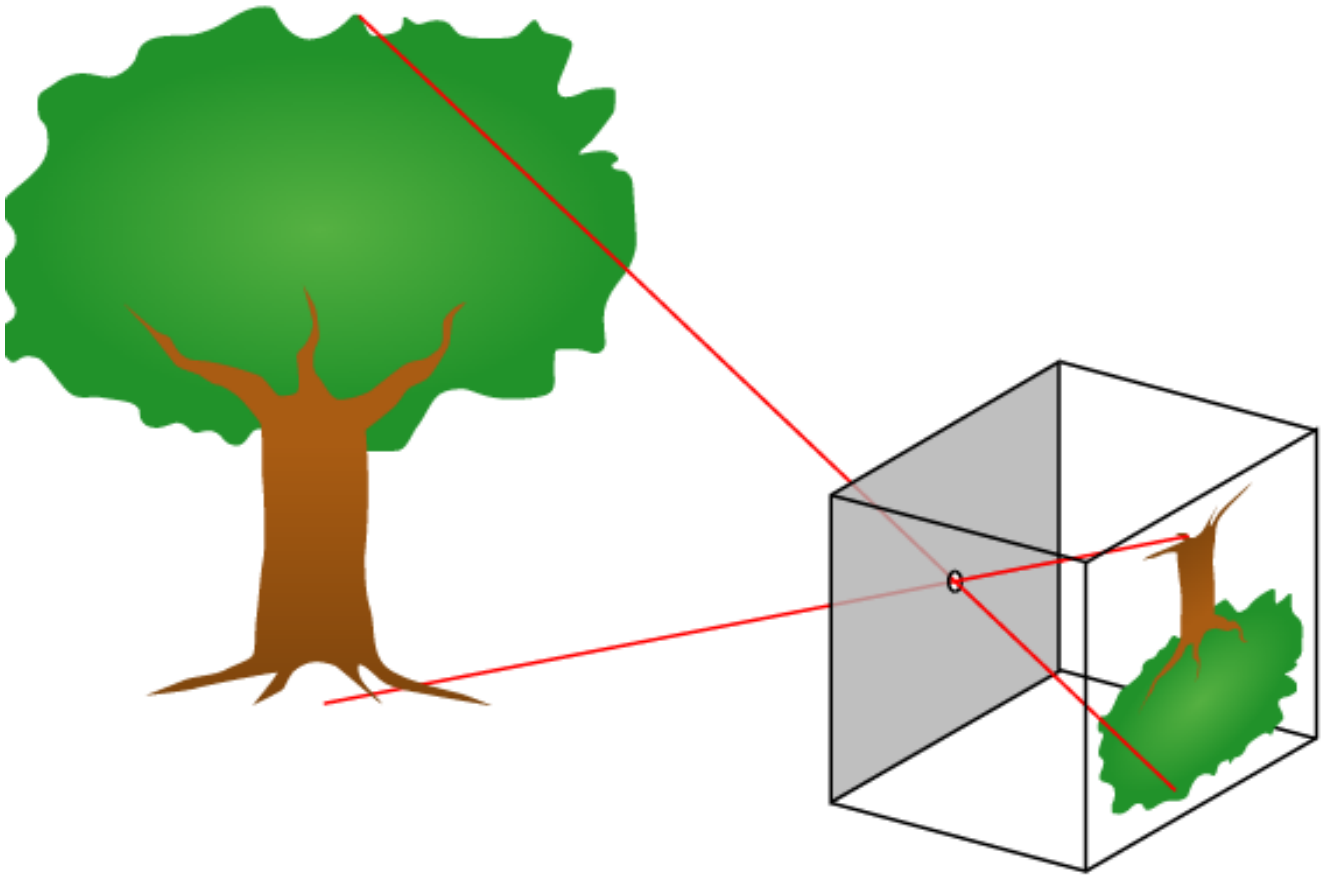
Perspective projection



- An object of the same size coming closer results on a larger image
- A point moving on the same ray does not change its image

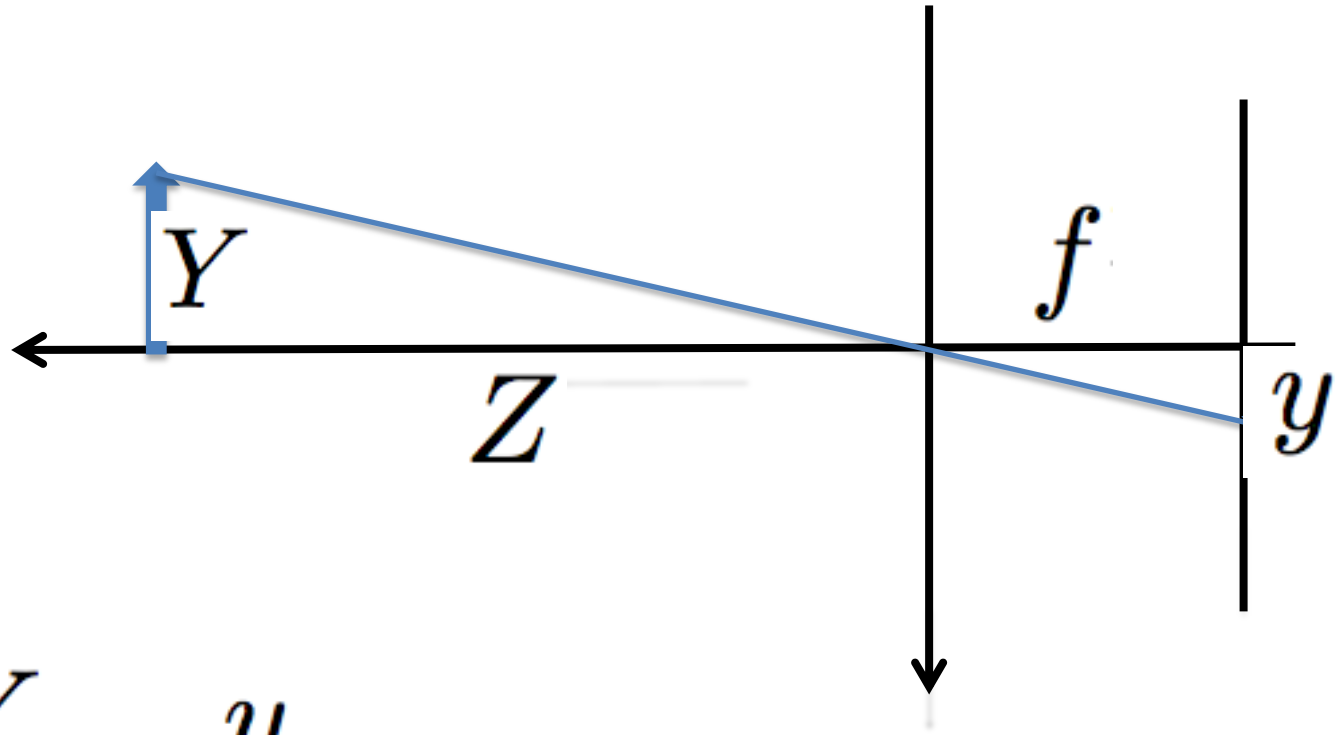
$$\frac{Y}{a} = \frac{y}{b}$$

Perspective projection = Pinhole Model



WIKIPEDIA
The Free Encyclopedia

Pinhole model



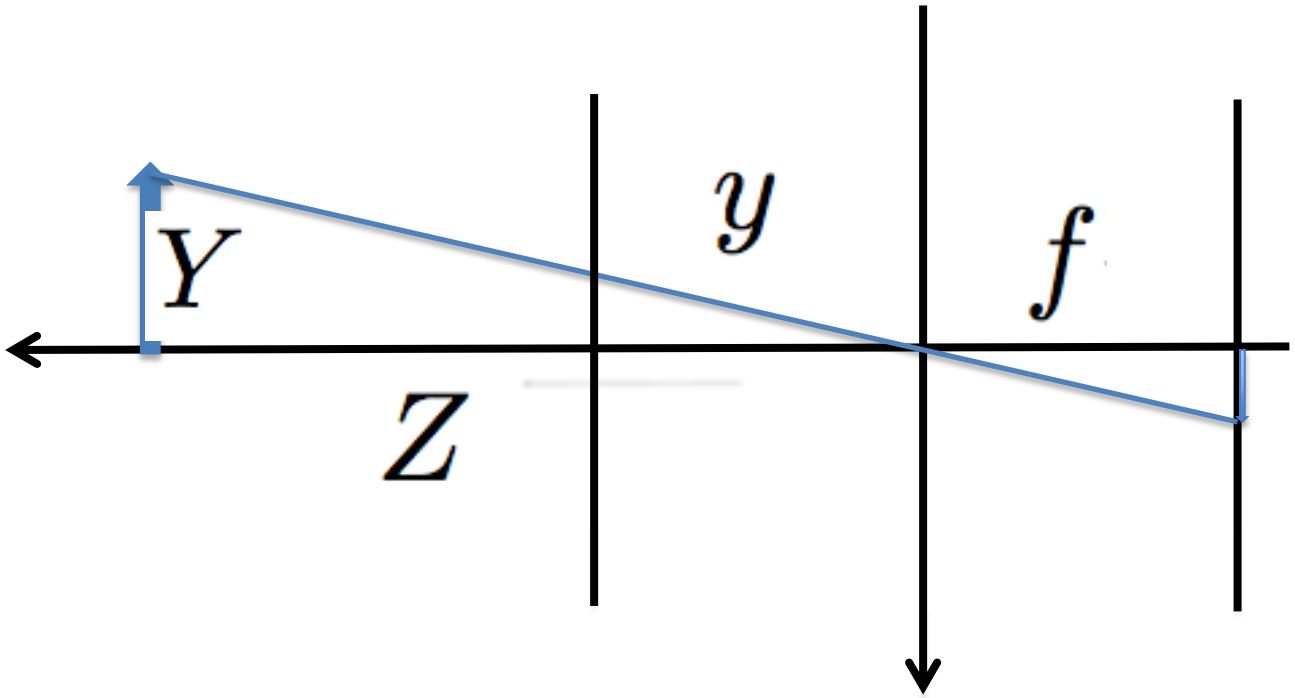
$$\frac{Y}{Z} = \frac{y}{b}$$

$$Z = a$$

If we replace b with f and include a minus because object image is upside down….

$$y = -f\frac{Y}{Z}$$

Pinhole model used in computer vision and robotics



… amd assume that image plane is in front of the lens

$$y = f\frac{Y}{Z}$$

What is the effect of f=b ?

- Theoretically, we expect an offset in the x and y coordinates caused by the error (f-b).
- If the object is on focus:

$$\frac{b-f}{f} = \frac{b}{Z}$$

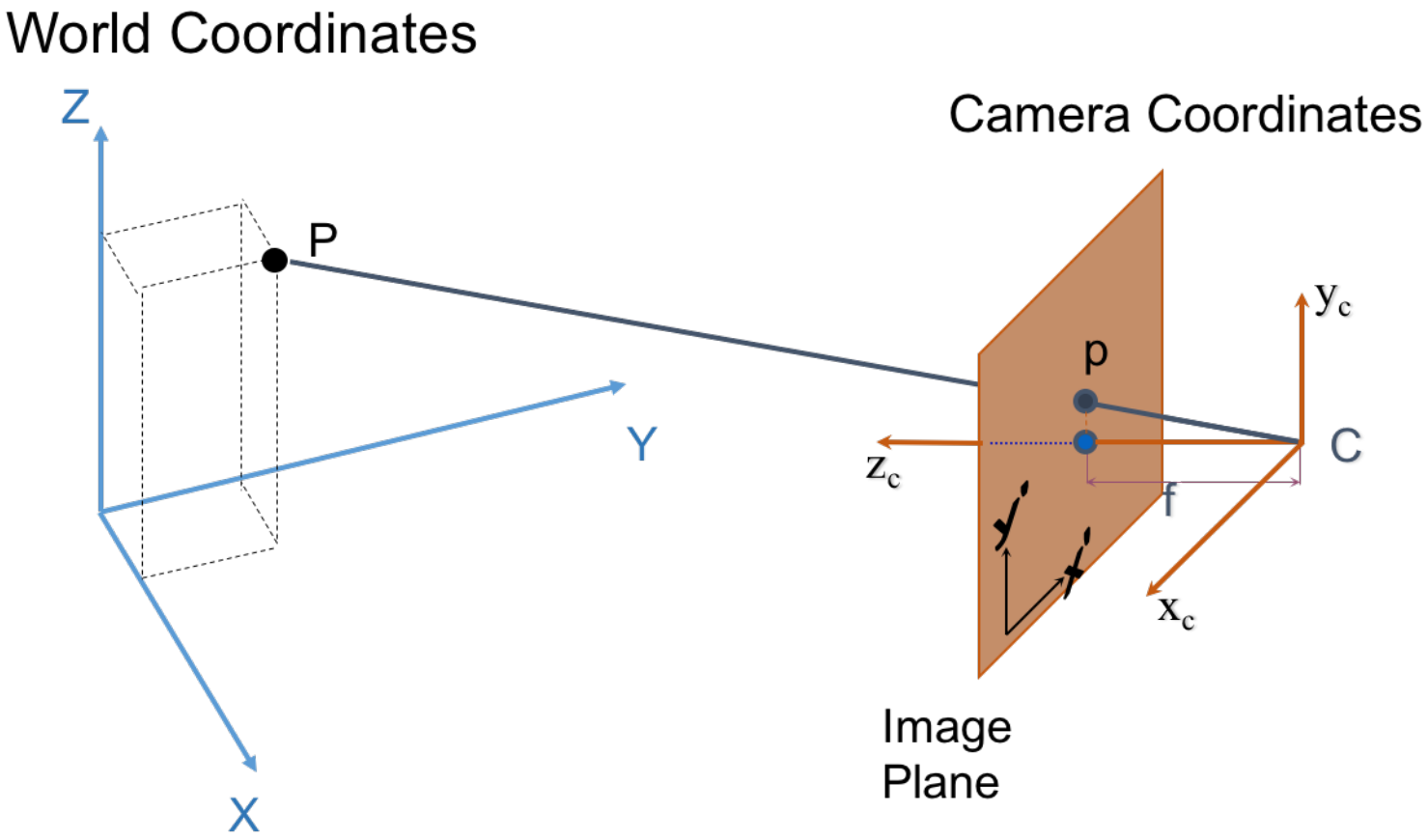
Relative error depends on the ratio of focal length to depth !
This would matter if we would actually use the f from specs of the camer

In practice we use a process called calibration, yielding the f that best satisfies

$$y = f\frac{Y}{Z}$$

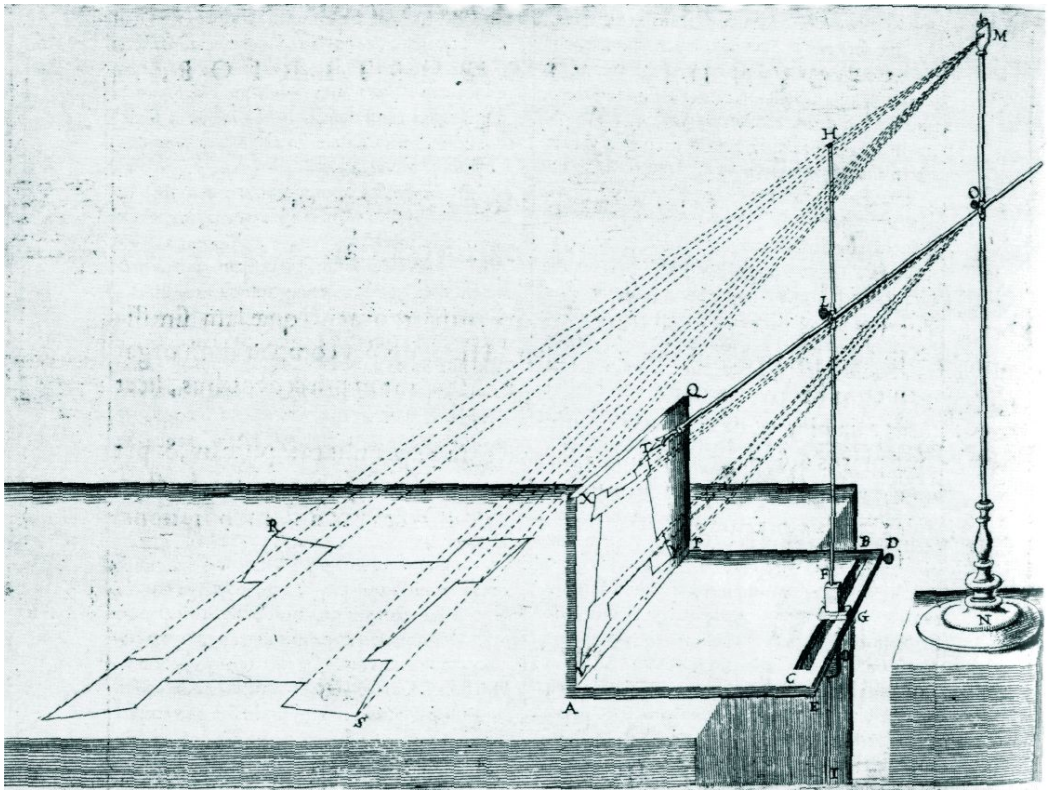
Single View Geometry

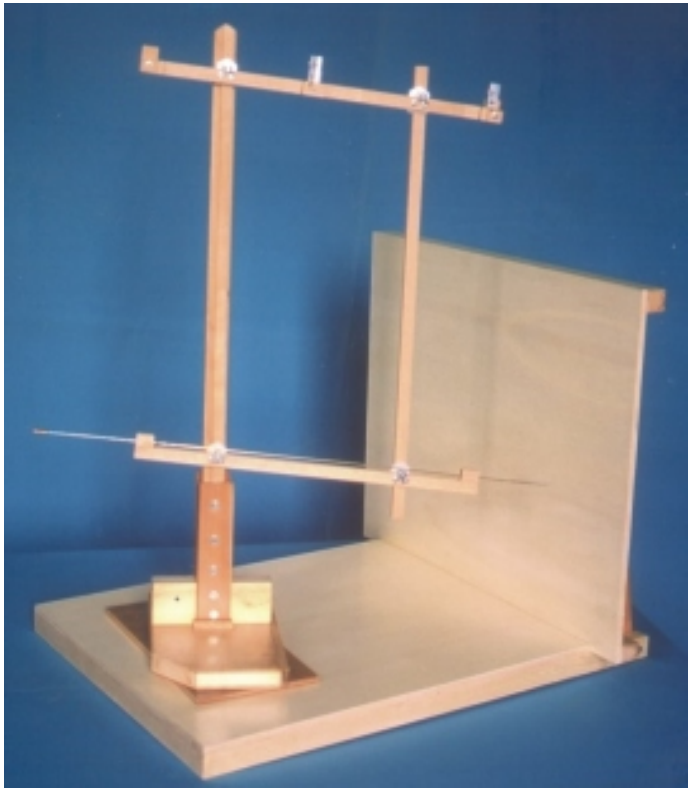

死亡事故発生
横断者あり 速度
株式会社 松金商事

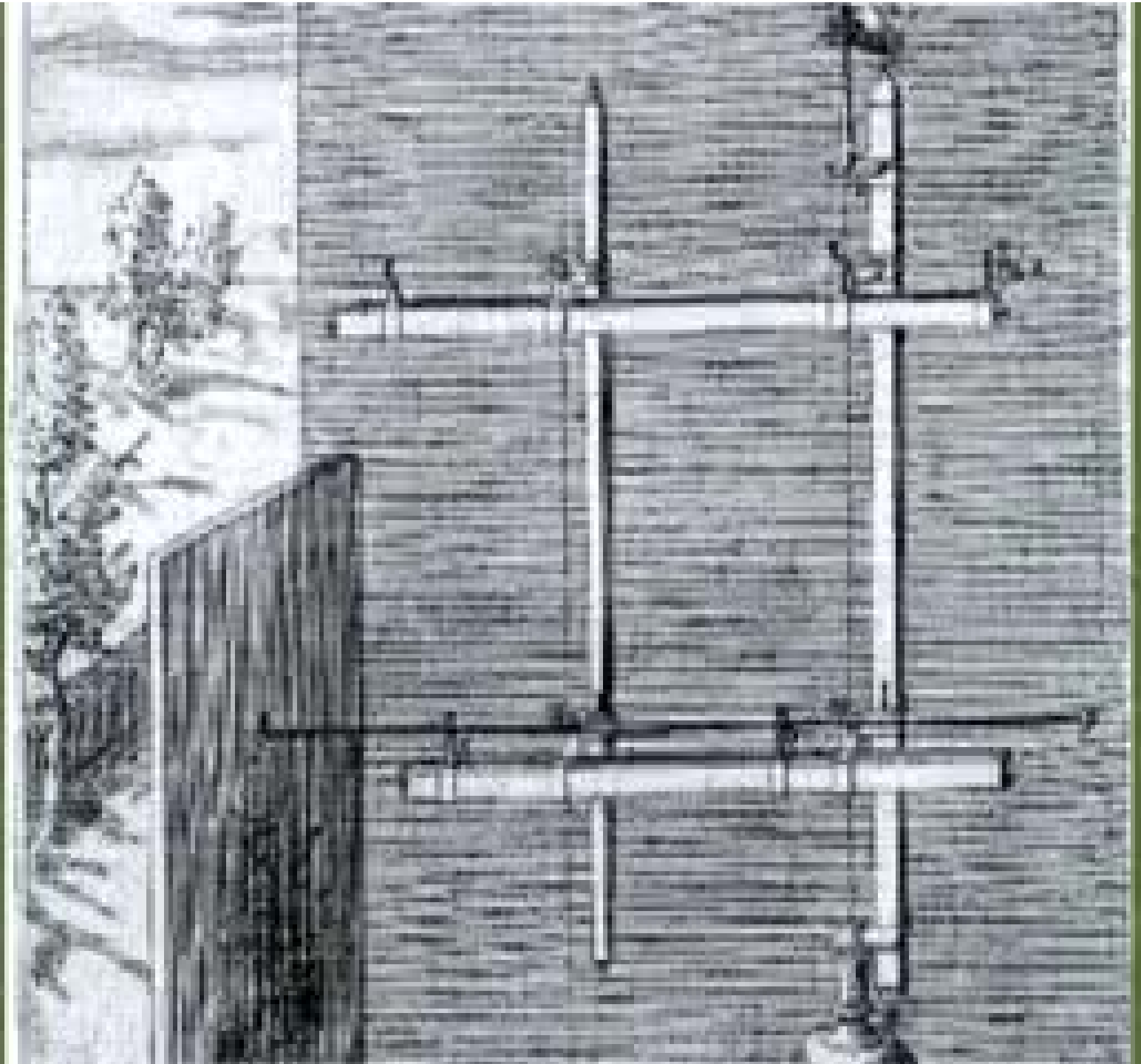

World Coordinates
Z
Y
X
P
Camera Coordinates
p
C
y_c
z_c
x_c
f
Image
Plane

J. Barozzi's Perspectograph

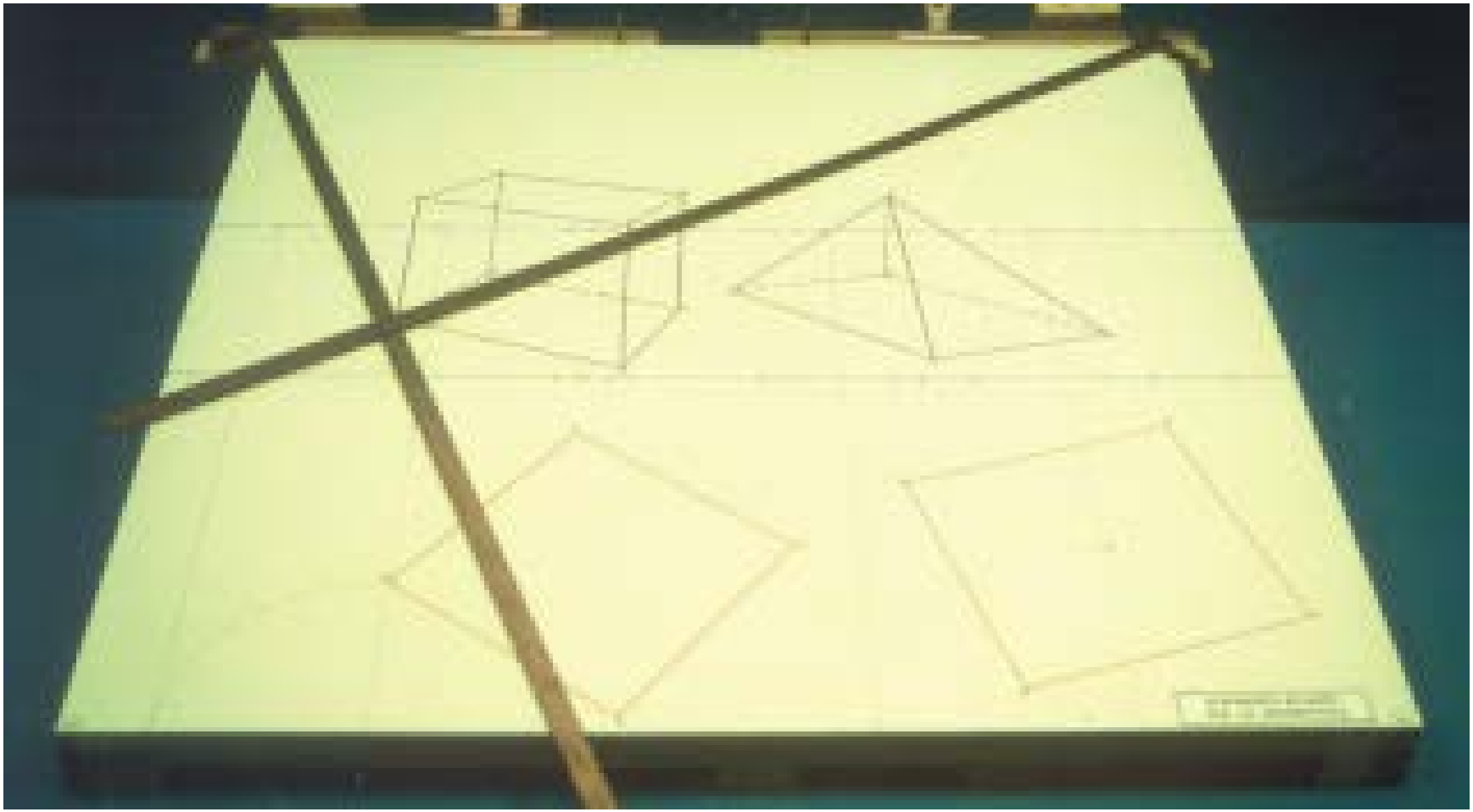
Bettini

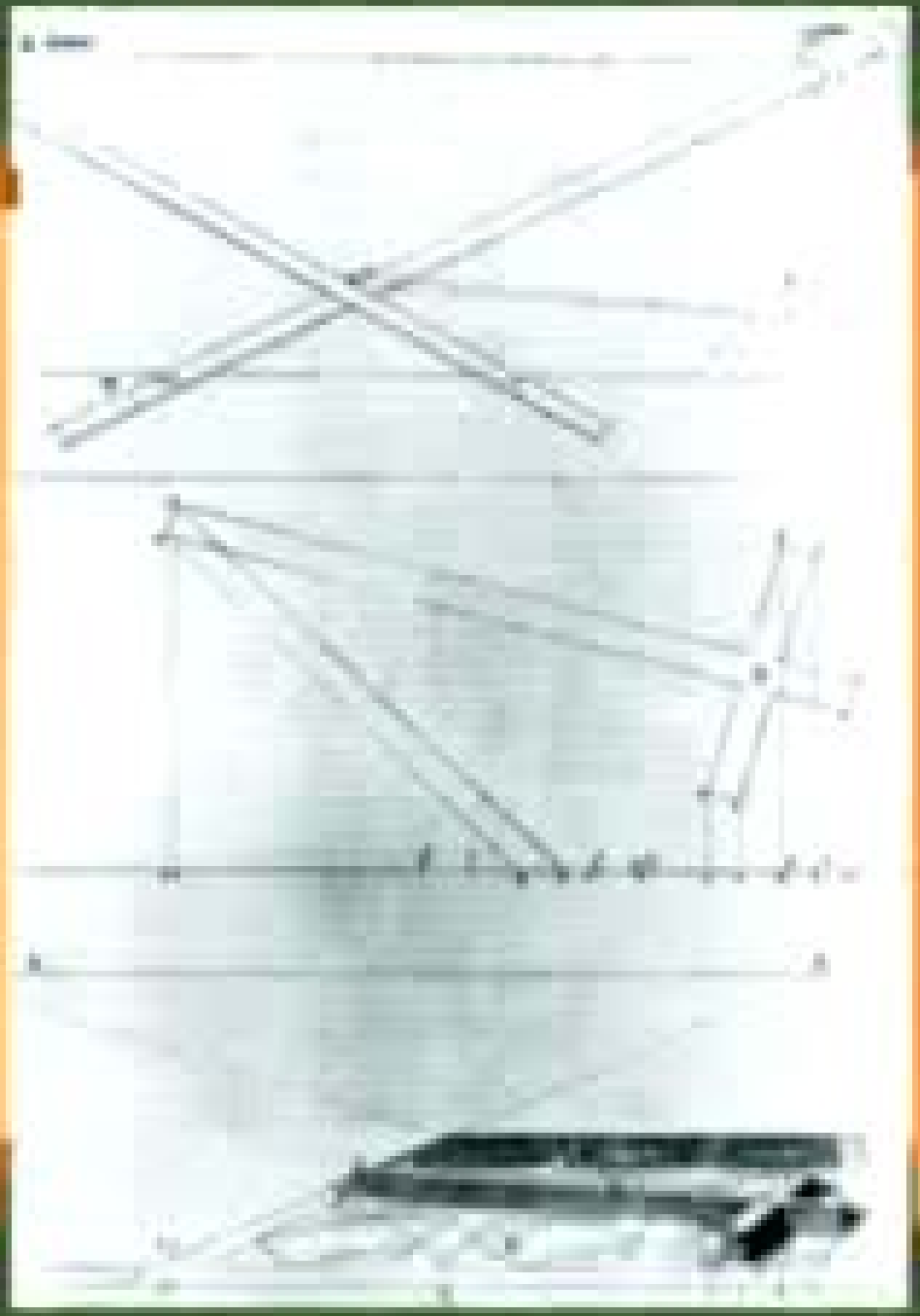






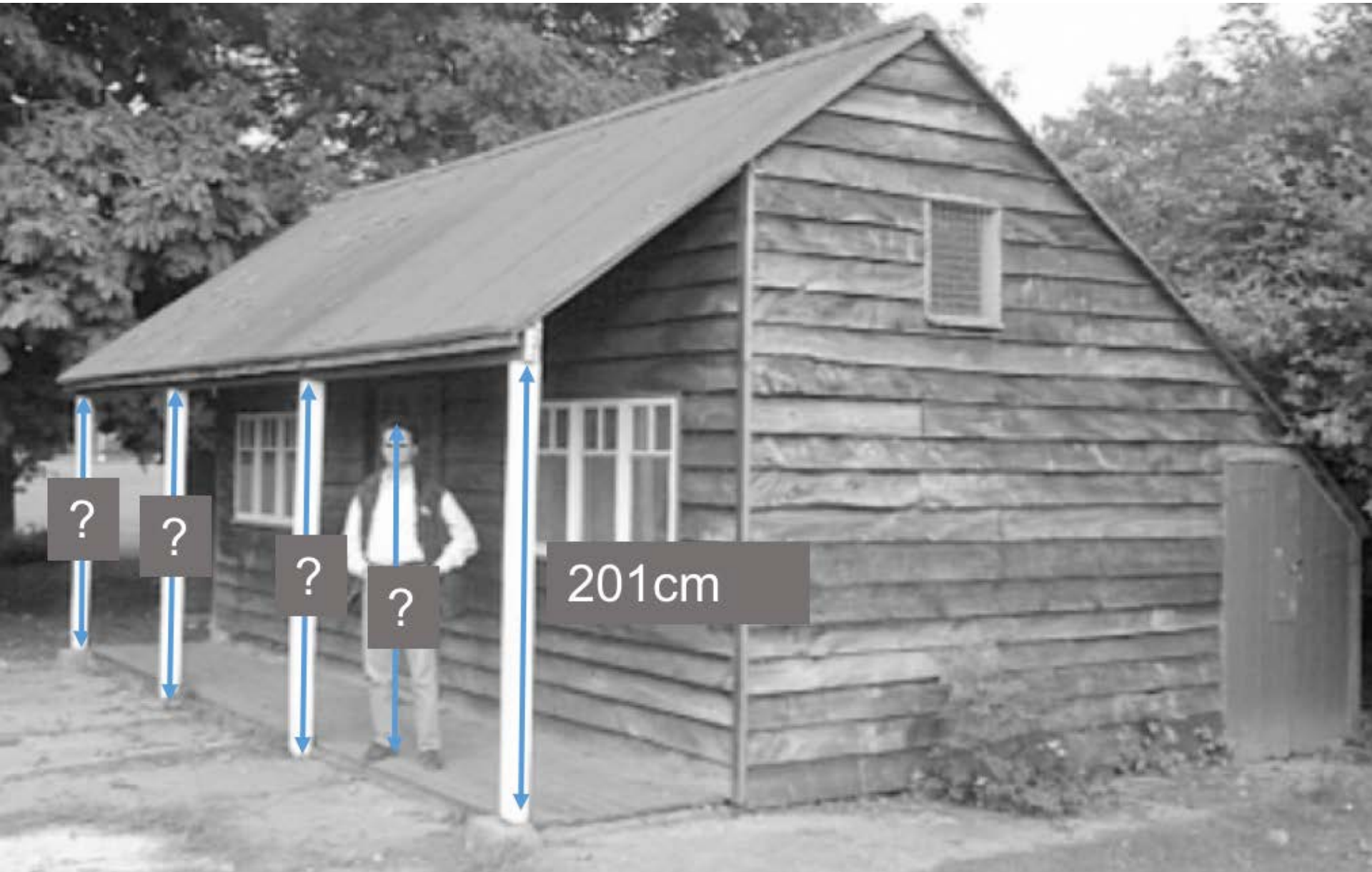
J. Barozzi's Bi-Dimensional Perspectograph





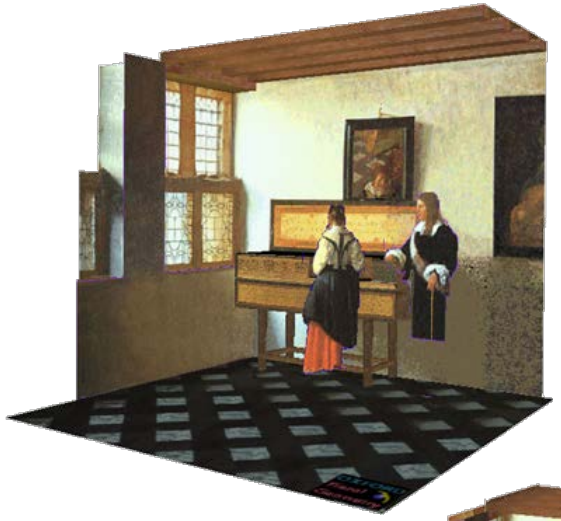
Single View Geometry

Single View Geometry



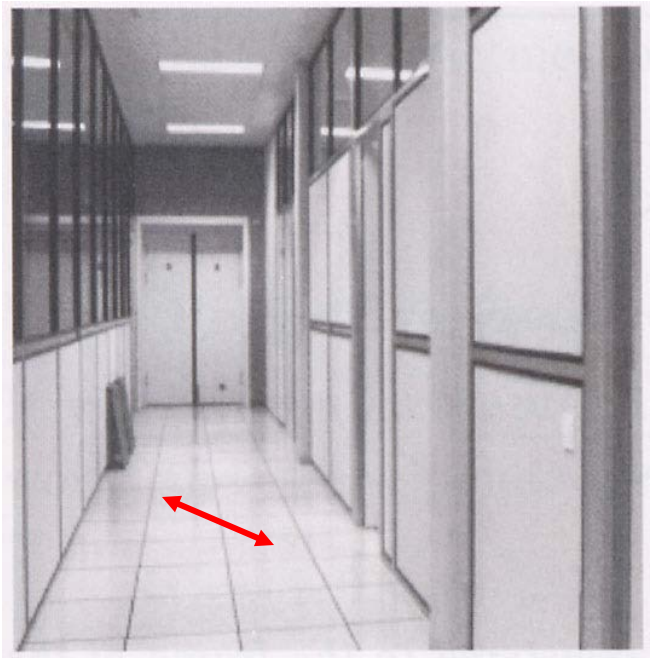
Single View Geometry

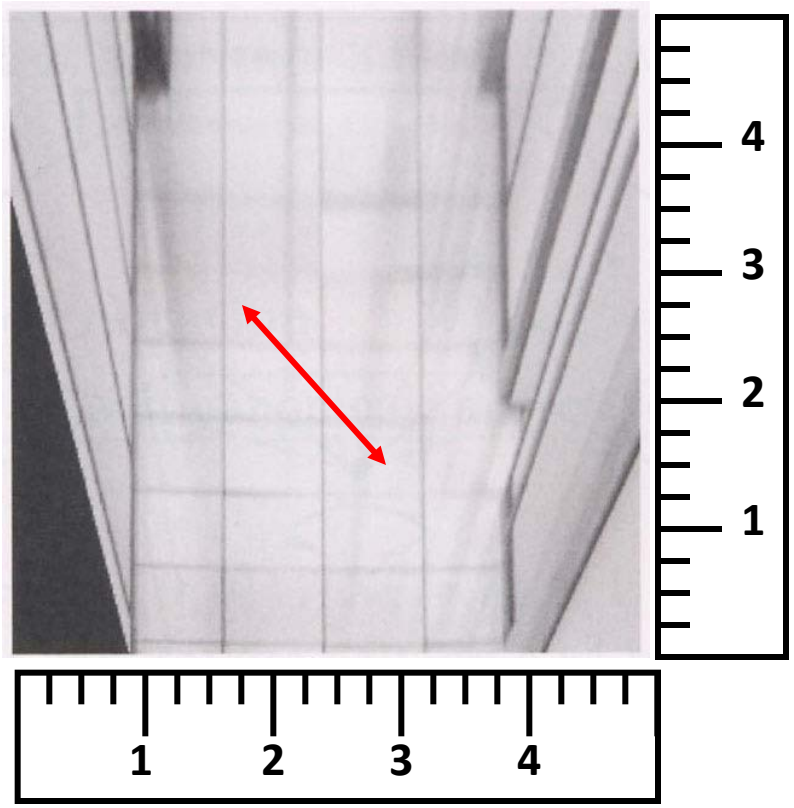
Vermeer's *Music Lesson*



- Criminisi et al., "Single View Metrology", ICCV 1999
- Other methods
  - Horry et al., "Tour Into the Picture", SIGGRAPH 96
  - Shum et al., CVPR 98
  - …

Ideas 1: Measurements on planes



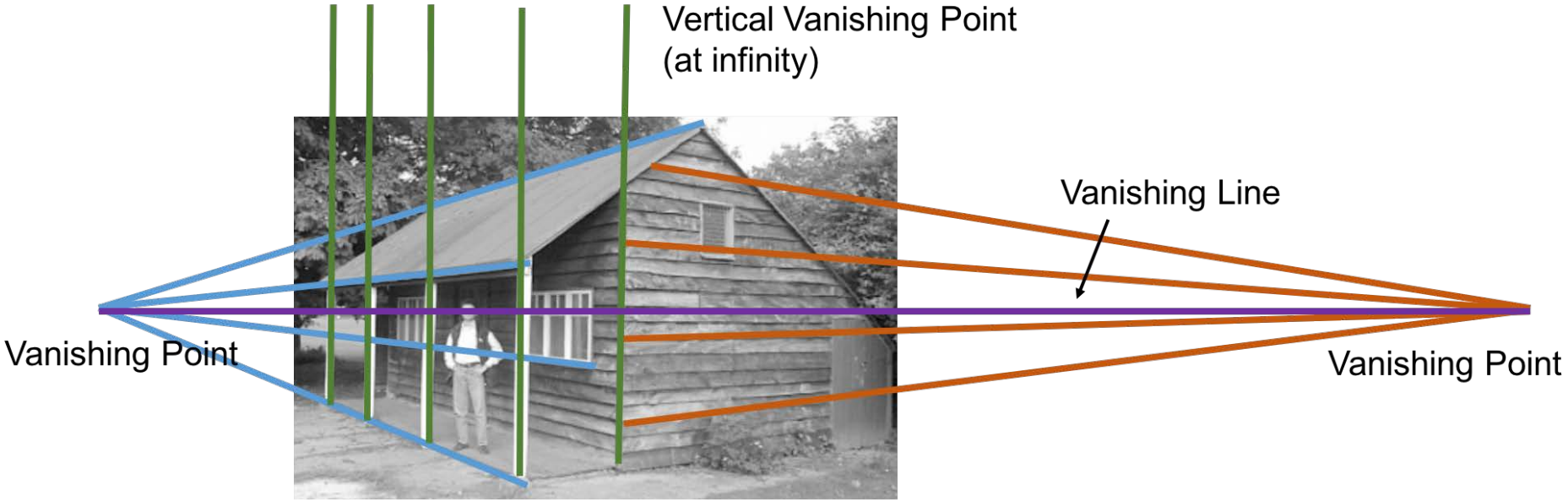


Approach: unwarp then measure

How to unwarp?

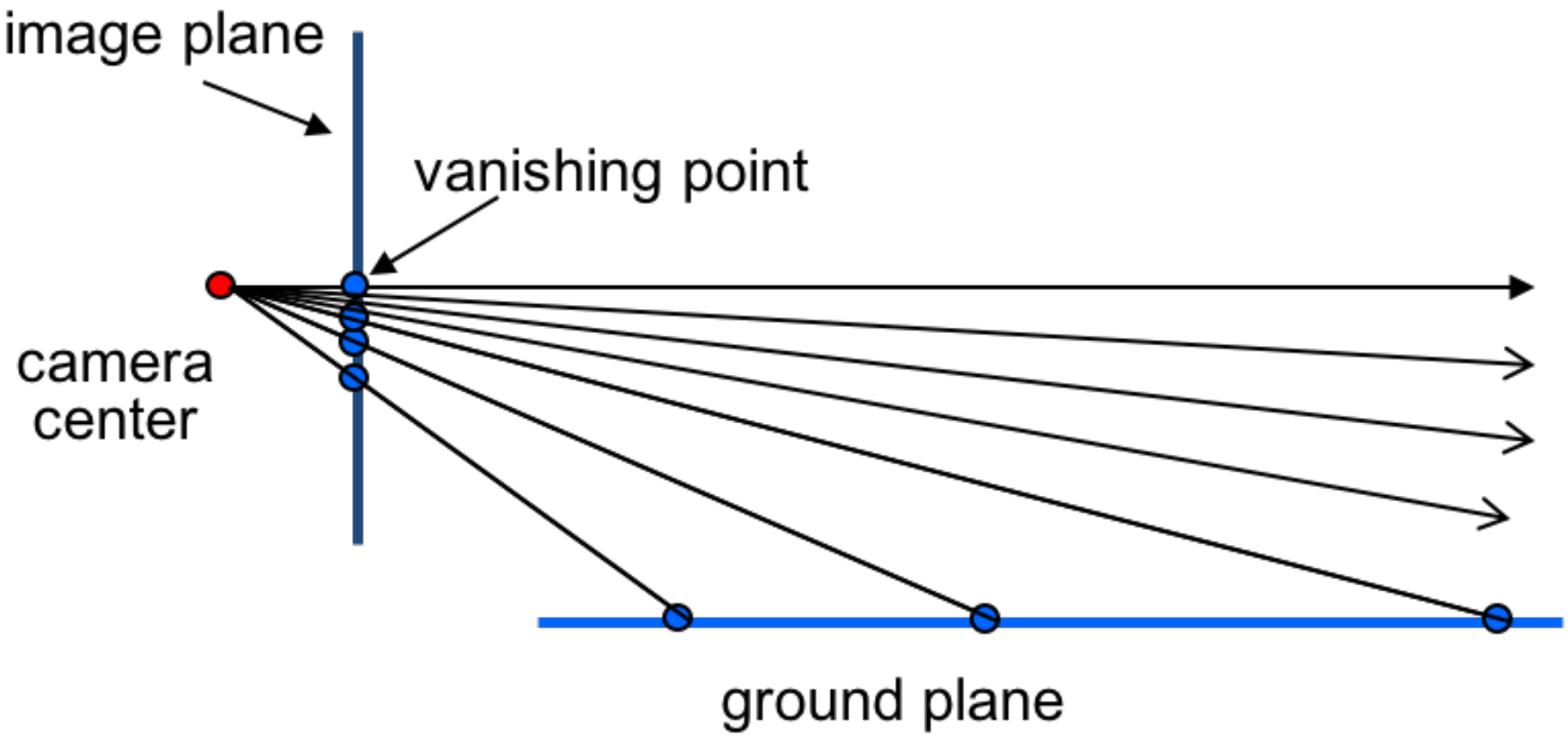
Make parallel lines stay parallel

Ideas of single view measurement



How tall is the photographer comparing to the person in the picture?

Vanishing point



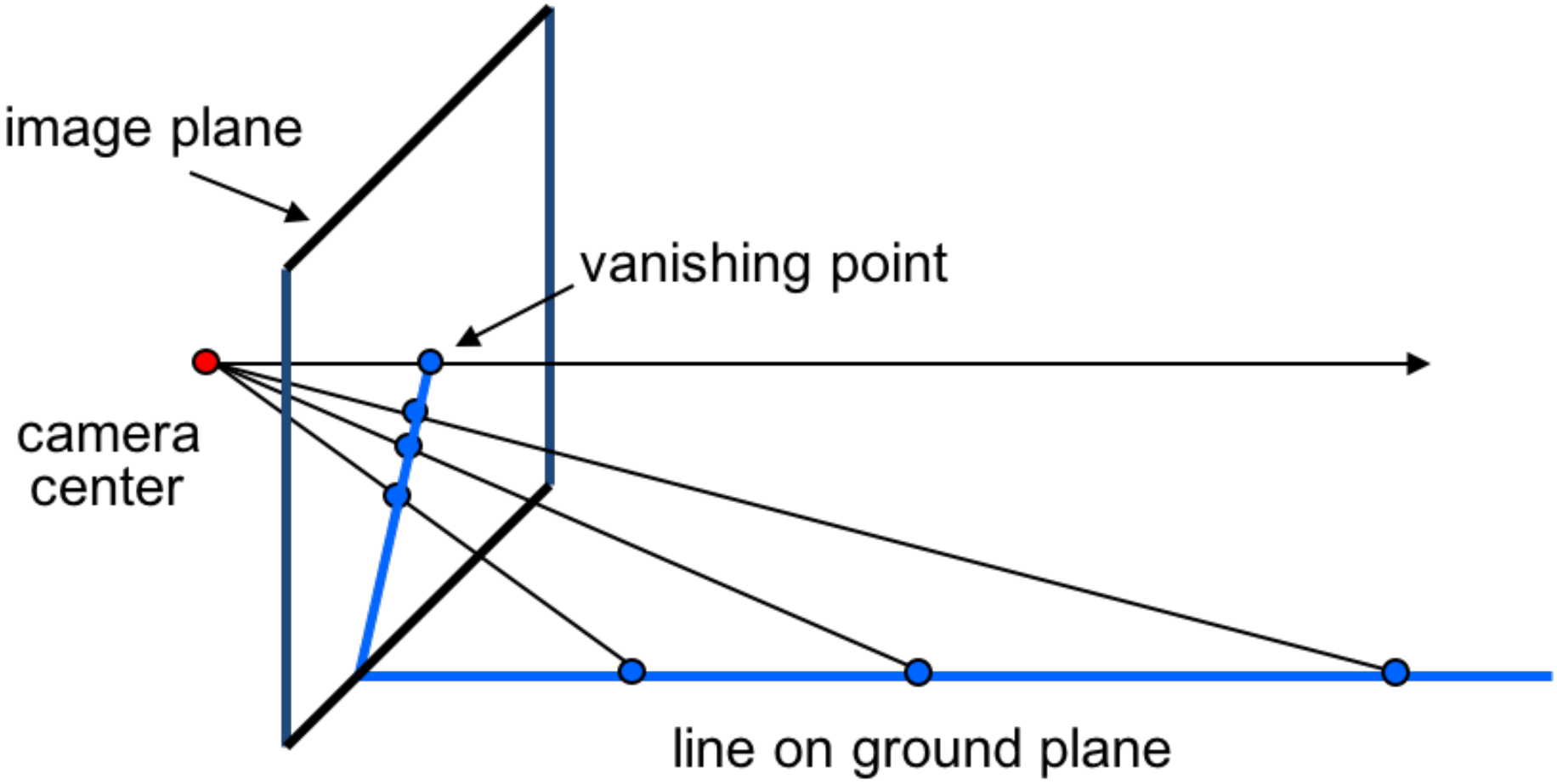
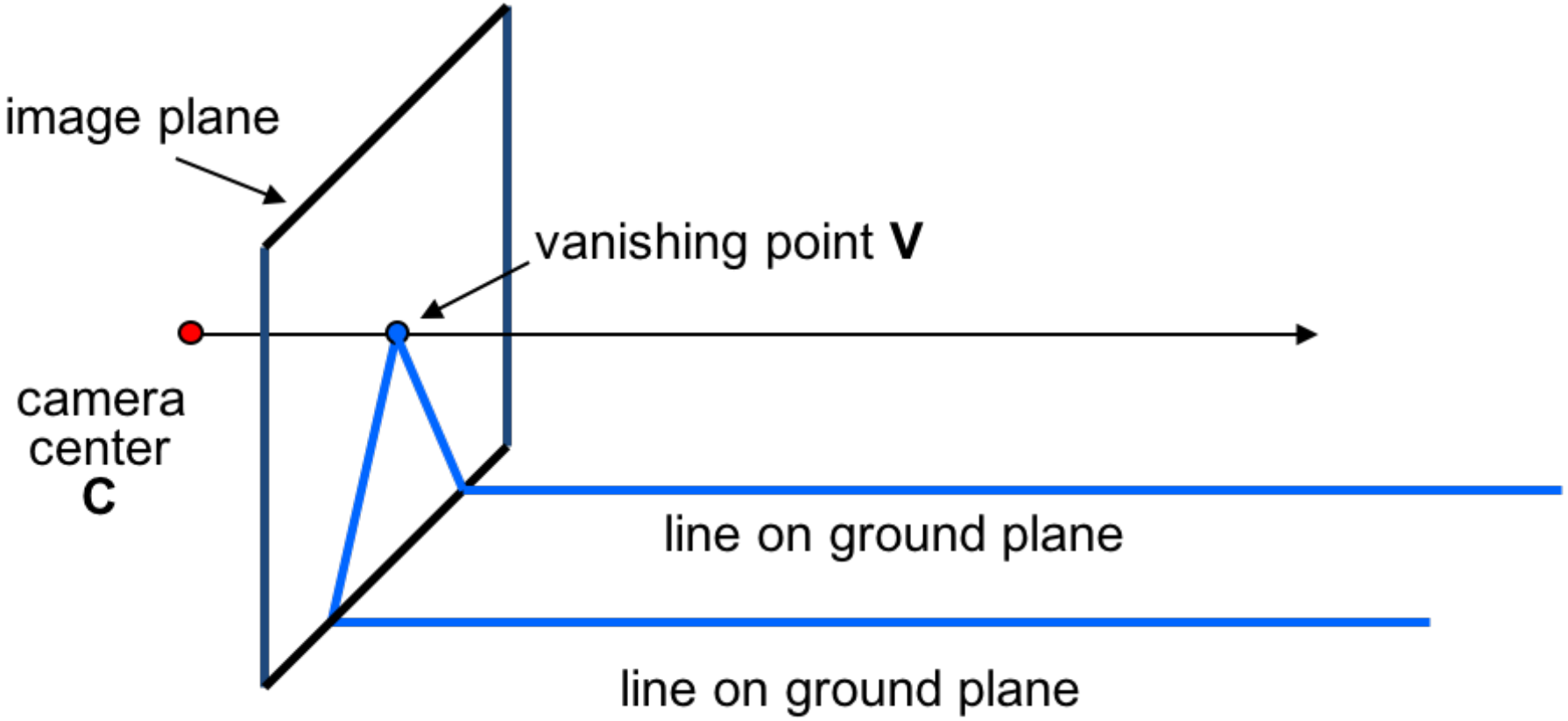


image plane
vanishing point
camera
center
line on ground plane

Imaging Vanishing Point



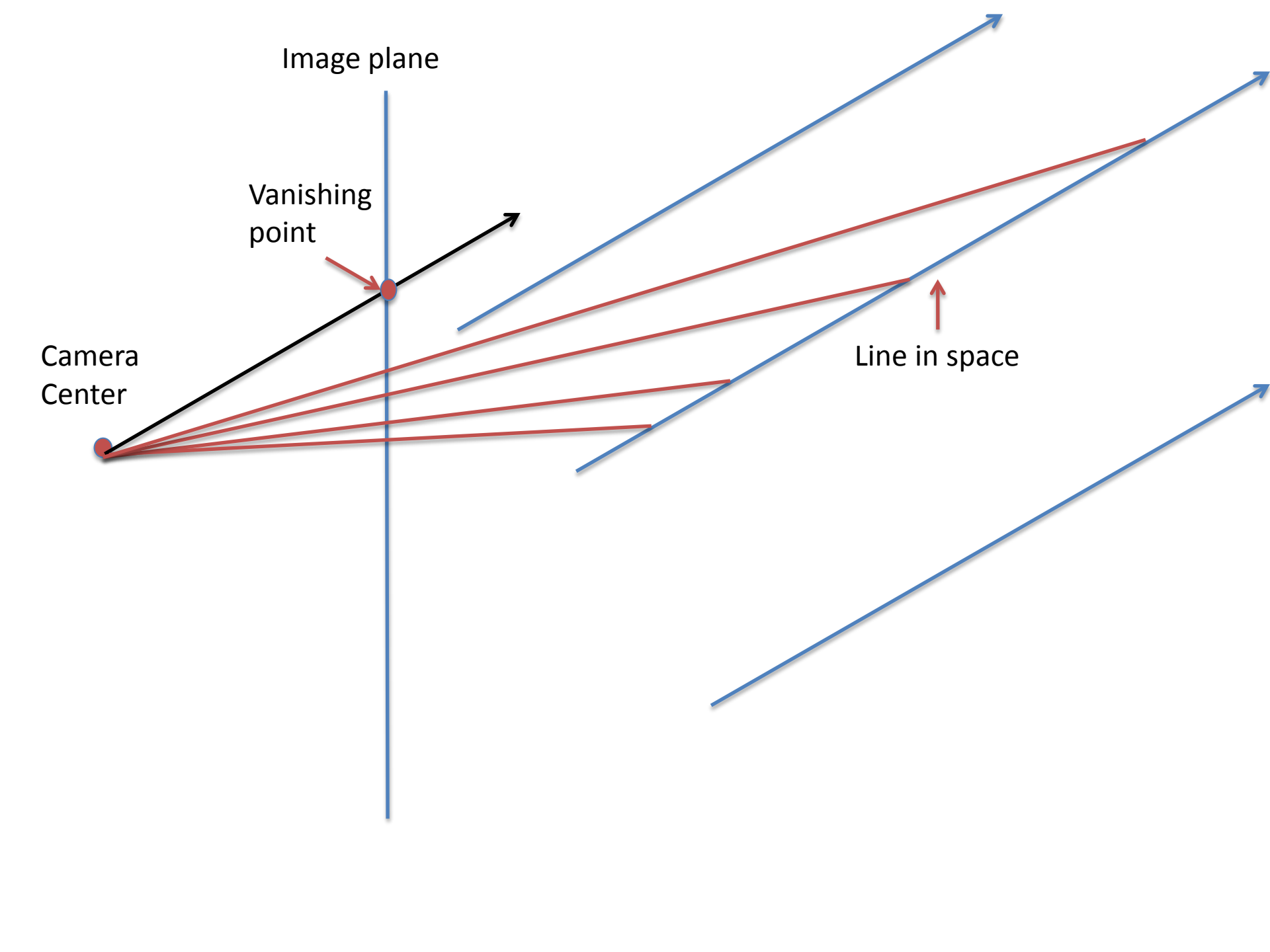
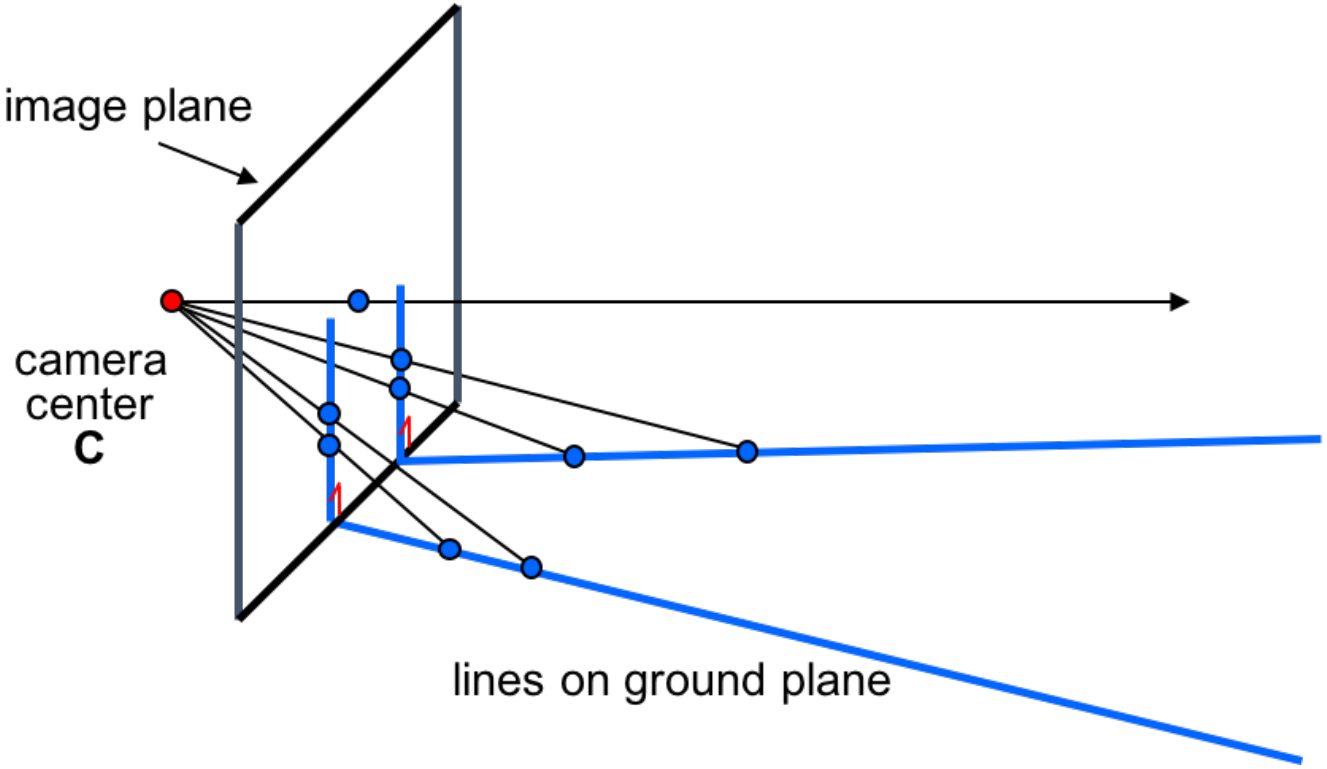
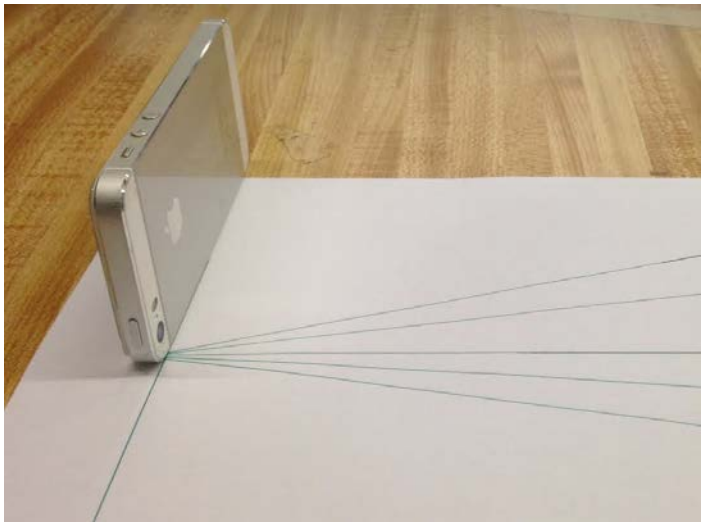
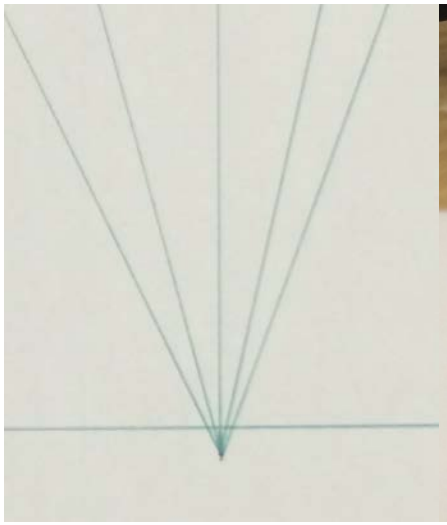
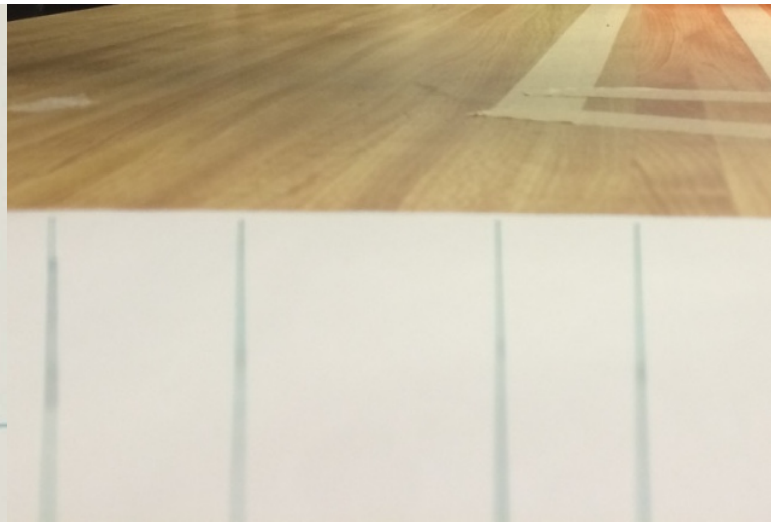


Image plane
Vanishing point
Camera Center
Line in space

Locating Center of Projection



Locating Center of Projection

More on Perspective Projection

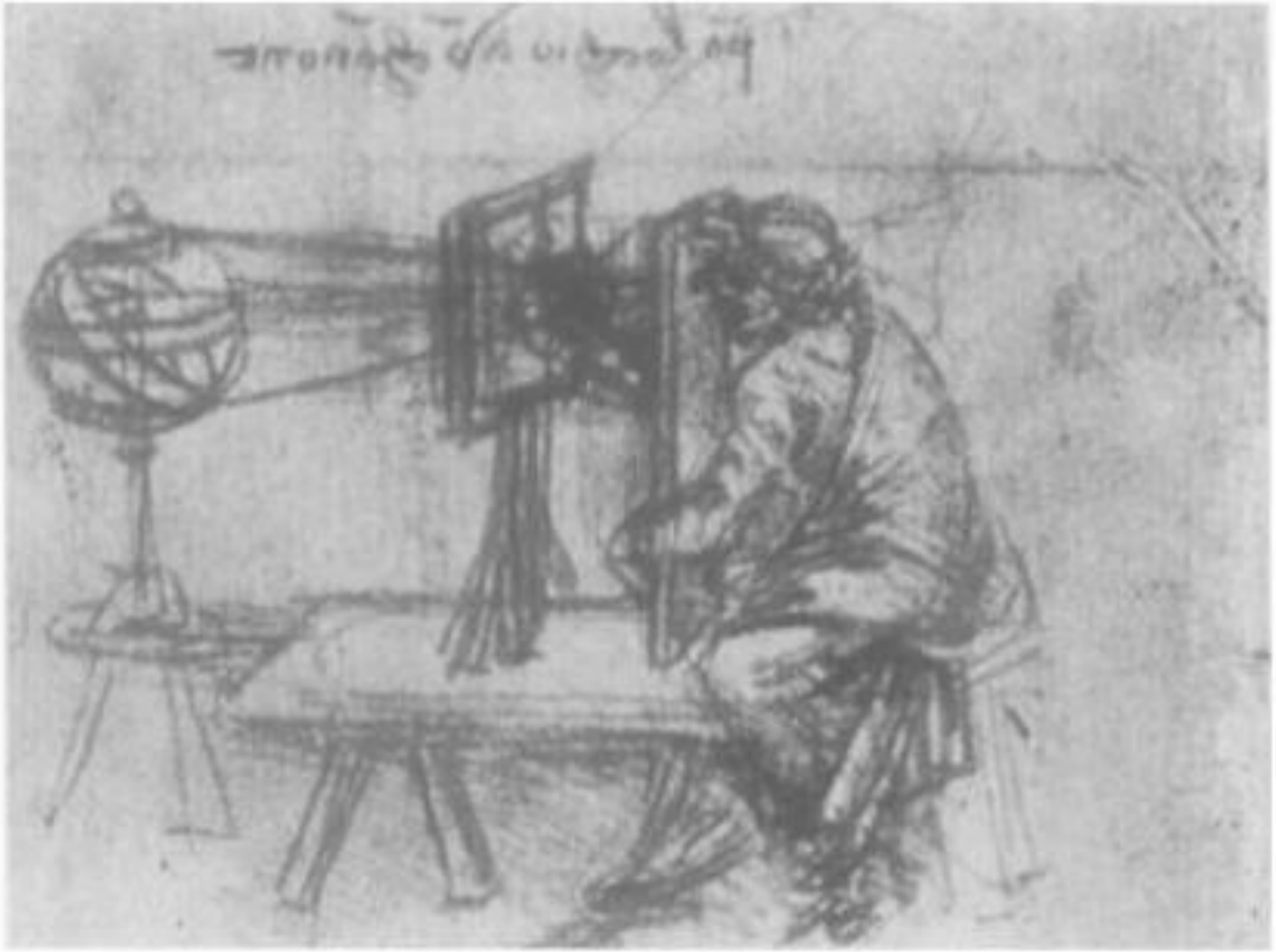


Figure 2.1. Leonardo's technique for making a perspectival drawing of the sphere of the macrocosm (CA 1 ra bis).

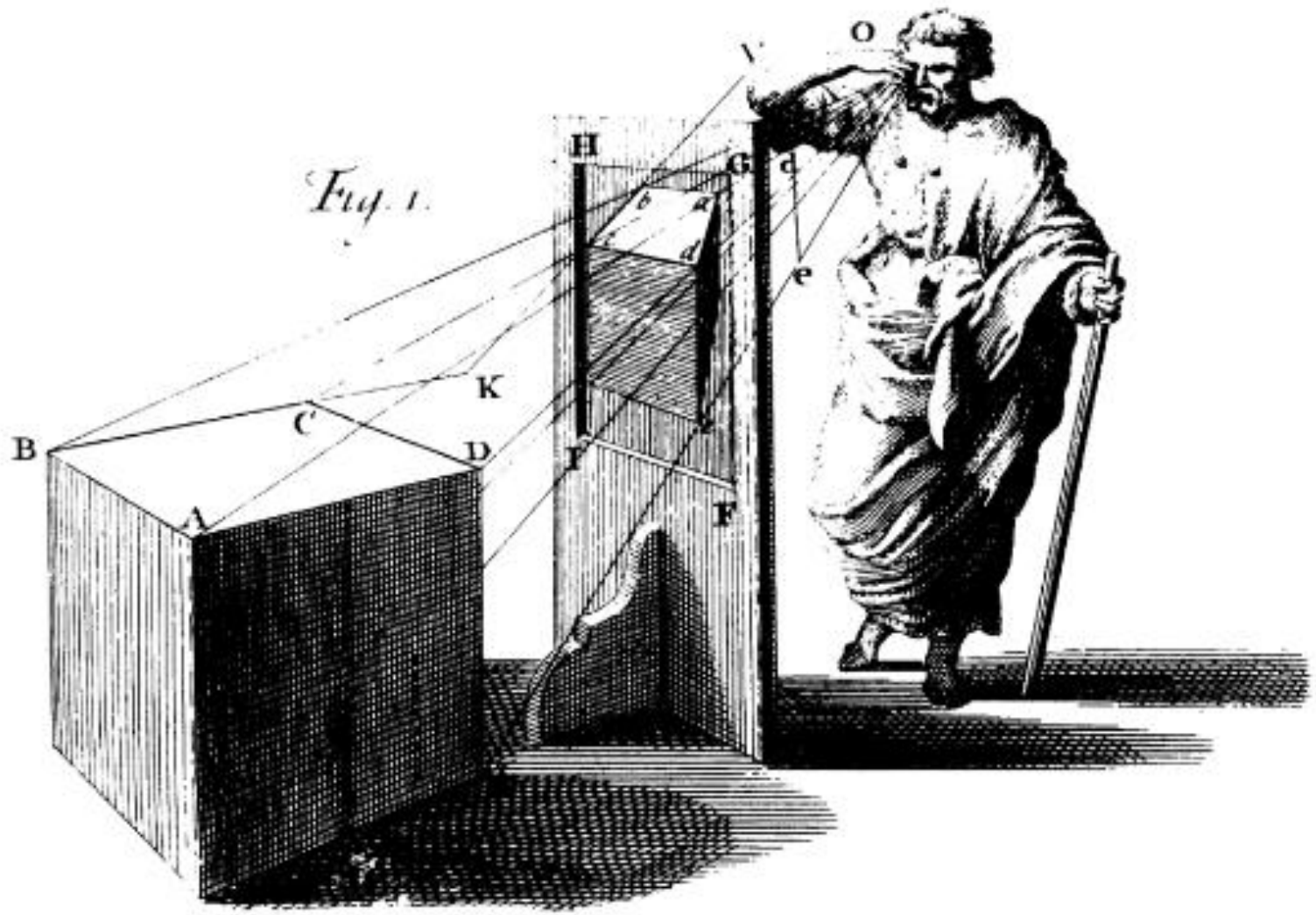


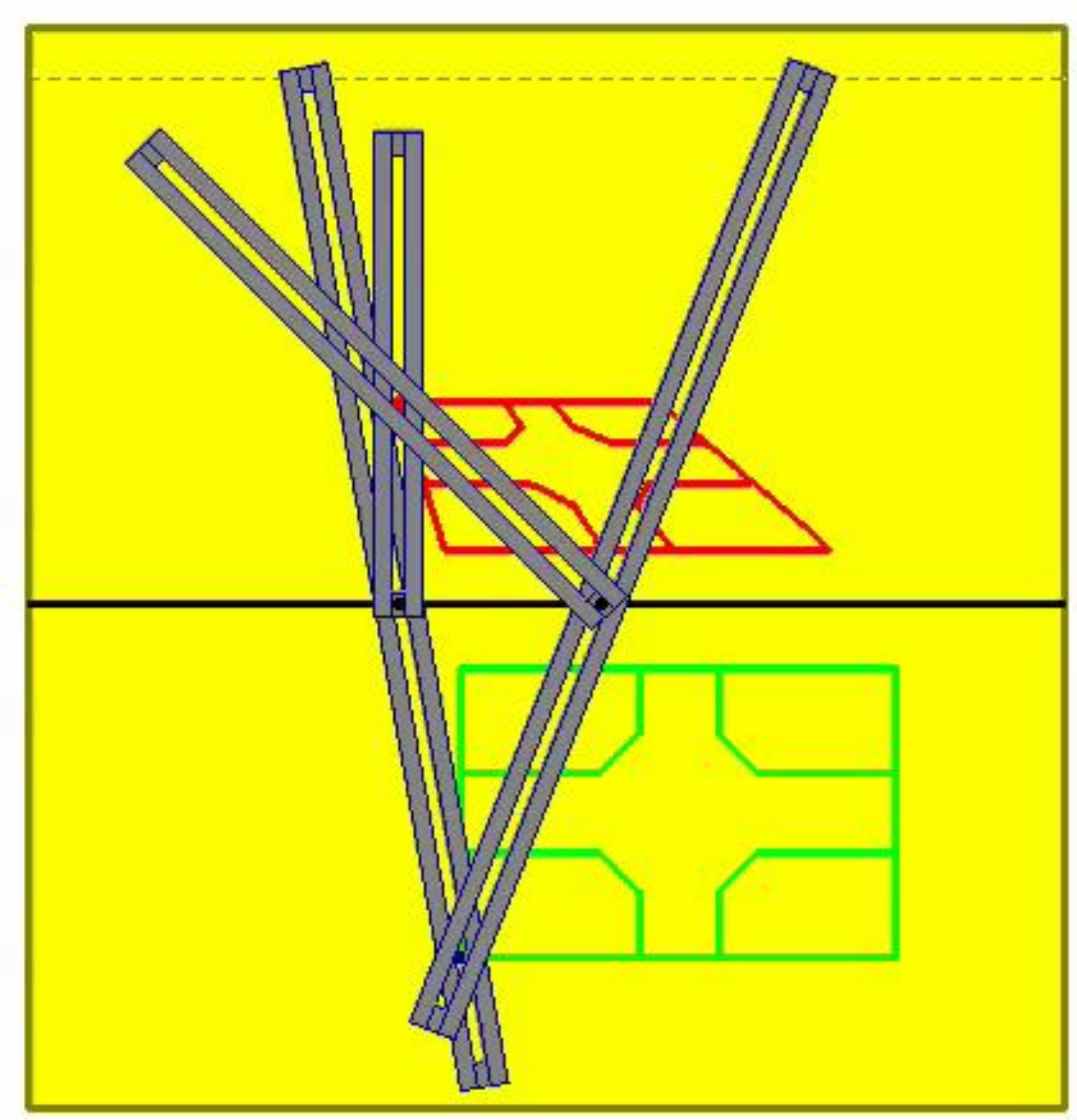
Fig. 1.
O
V
H
G
b
a
d
c
e
K
C
D
B
A
I
F

Bi-perspectograph 1752

LAMBERT'S TWO-DIMENSIONAL PERSPECTOGRAPH (1).

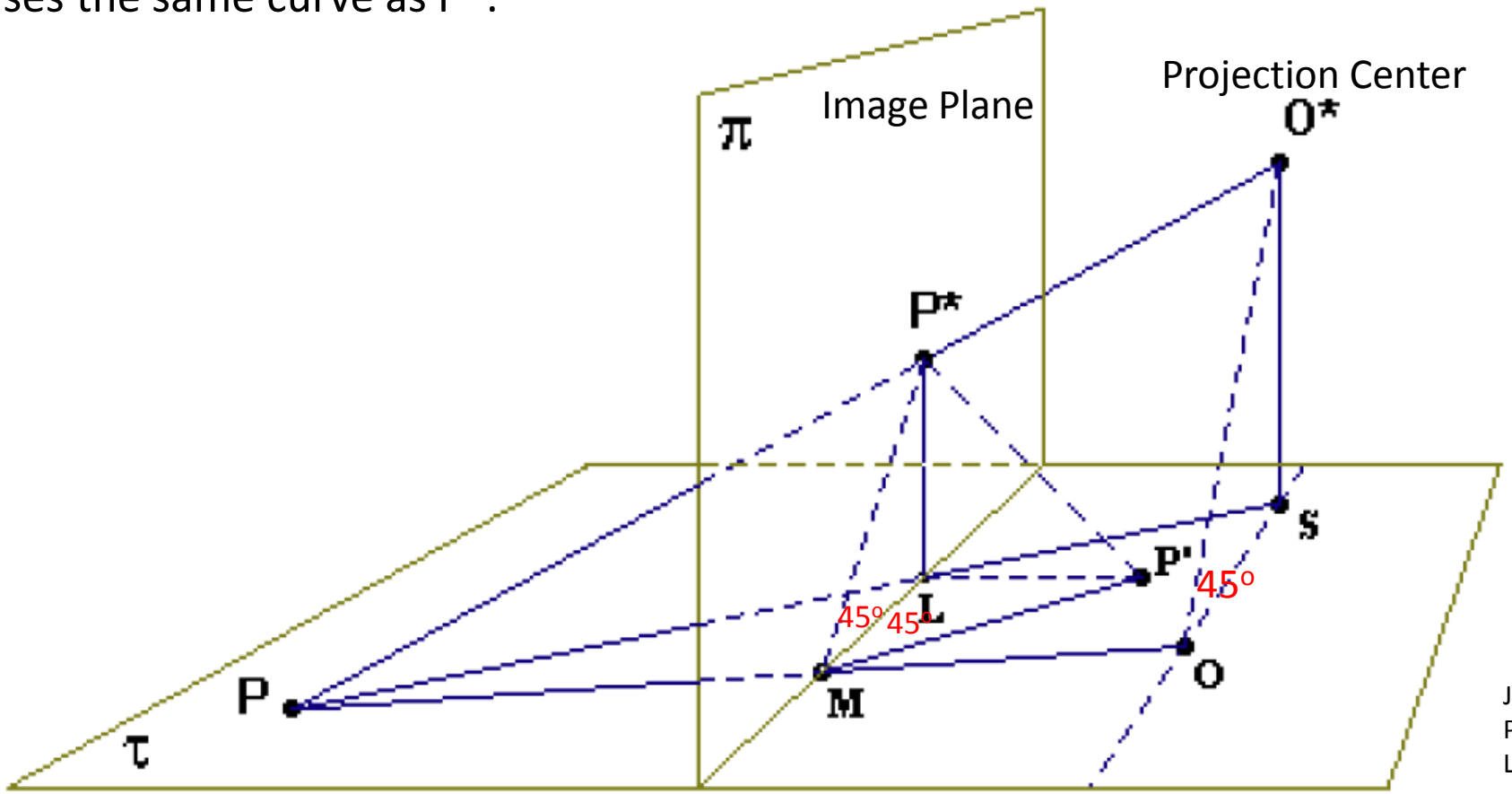
(From: J. H. Lambert, "*Anlage zur Perspektive*", manuscript, August 1752; "*Essai sur la Perspective*", edizione Peiffer – Laurent, 1981)

Mechanical realization



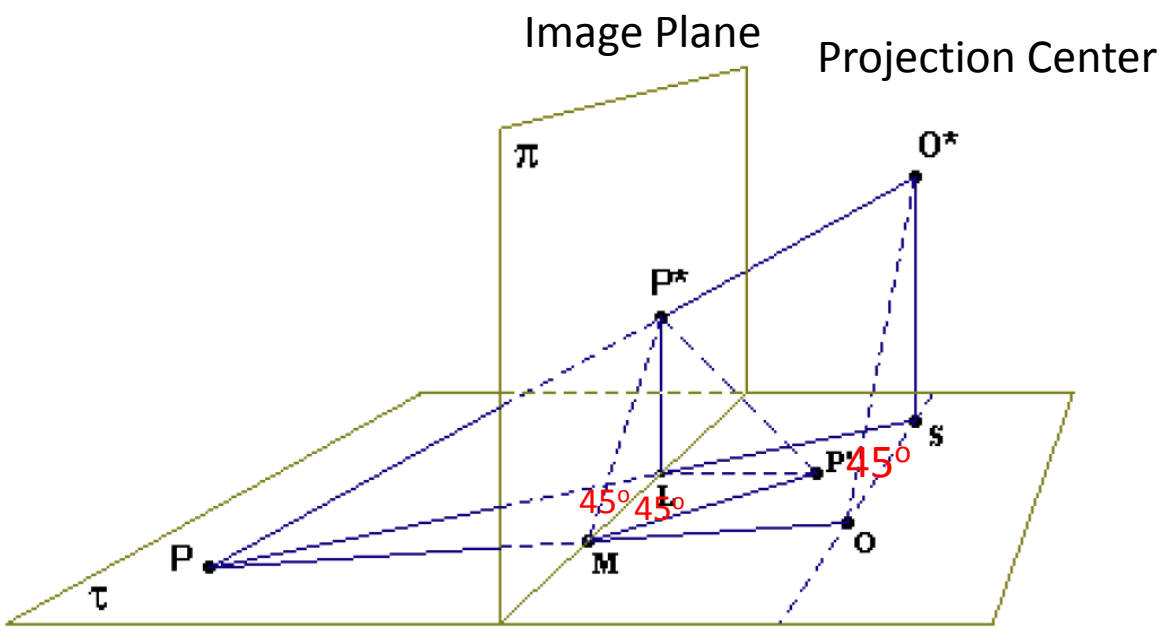
Geometric explanation of bi-perspectograph:

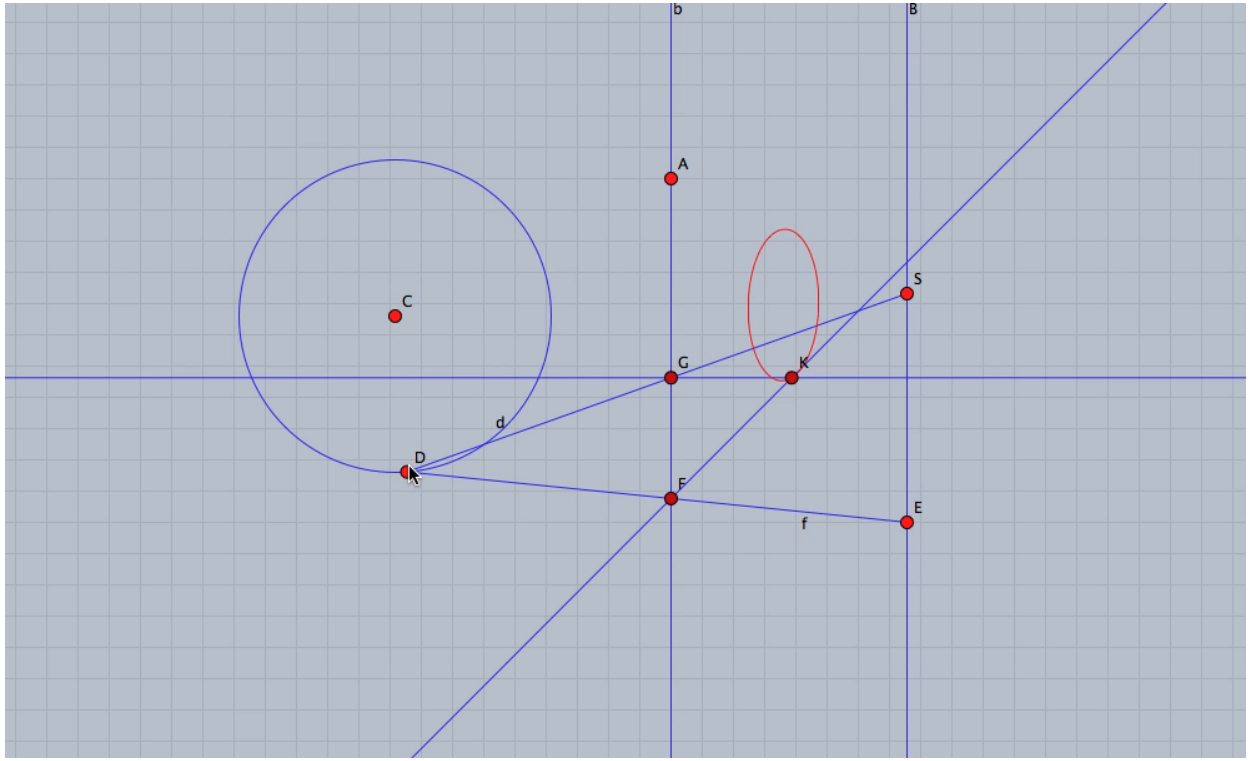
How can we draw a congruent copy of the image plane on the ground plane?
Select O such that OS=SO*. This means angle SOO*=45° and hence angle LMP* is 45° as well.
Draw line at L perpendicular to LM. Draw line at M with fixed angle 45°. Call their intersection P'. Then triangle P*LM is congruent to P'LM.
Hence, P' traverses the same curve as P*.



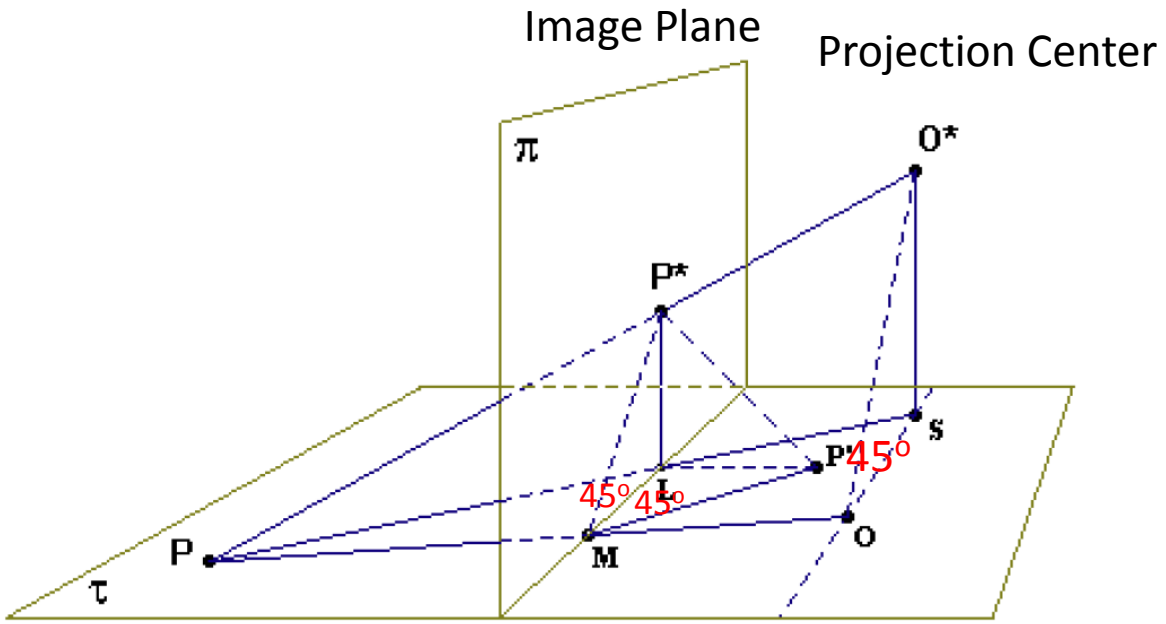
J. H. Lambert, Anlage zur Perspektive, edizione Peiffer - Laurent, 1981

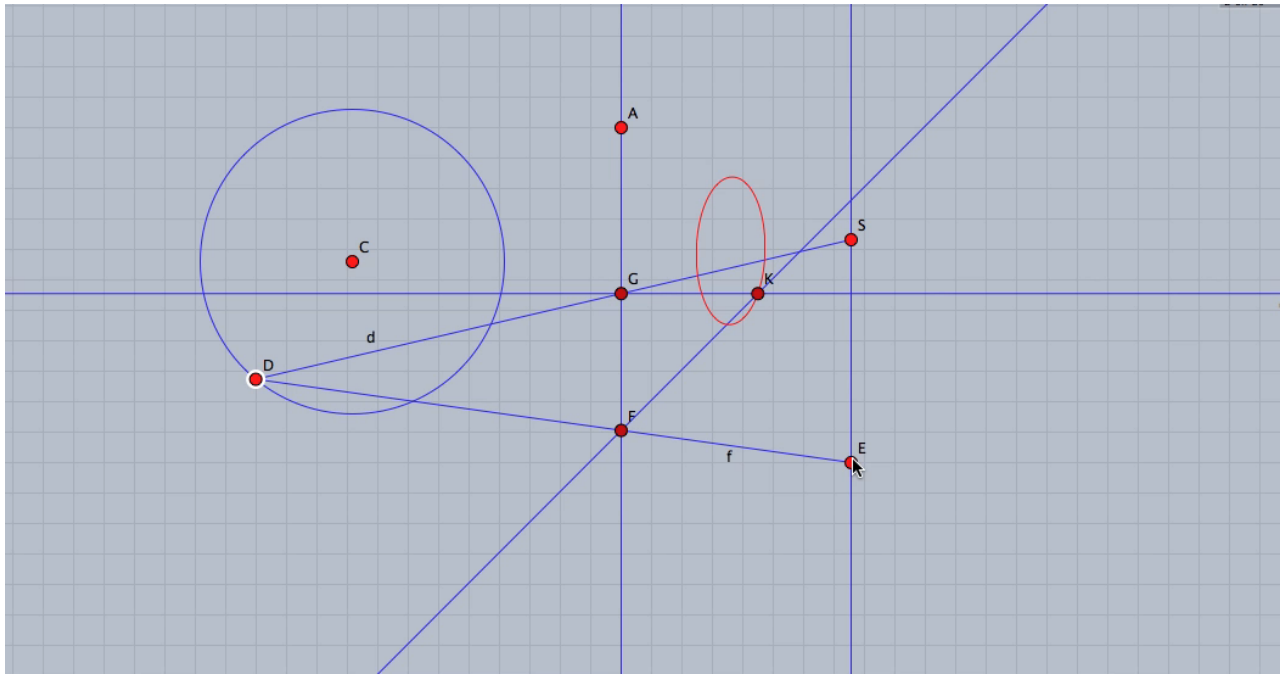
A circle is projected into an ellipse



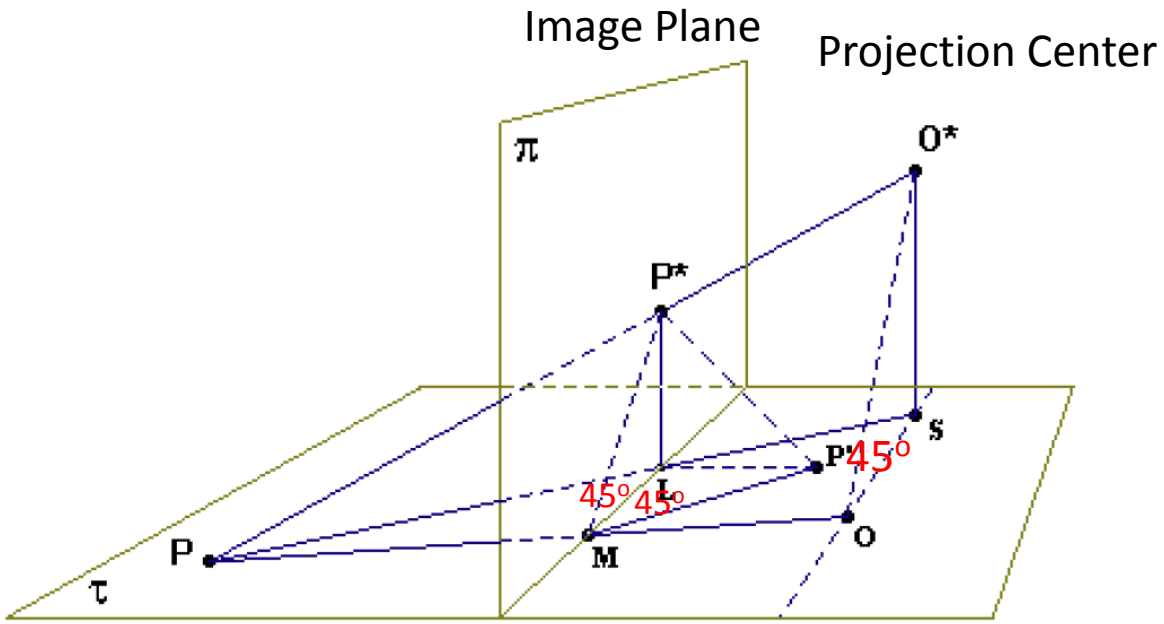


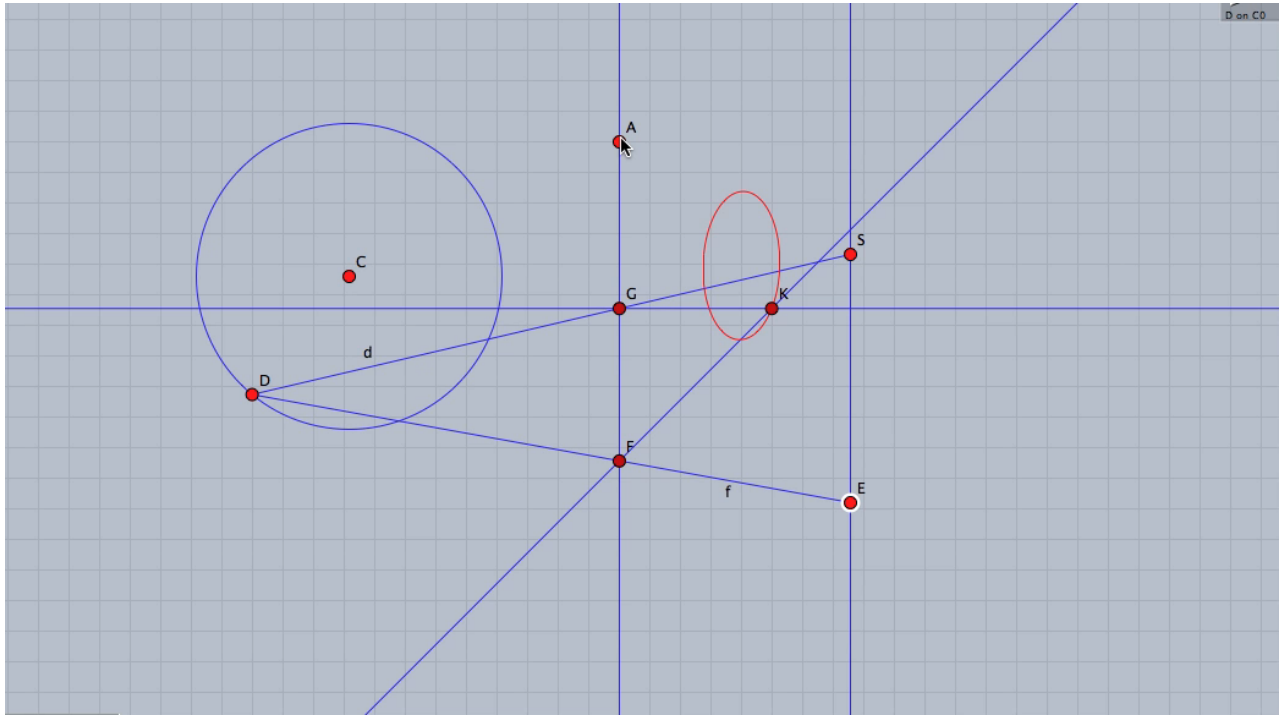
Effect of height of camera on ellipse shape: the lower the camera, the more squeezed is the ellipse.



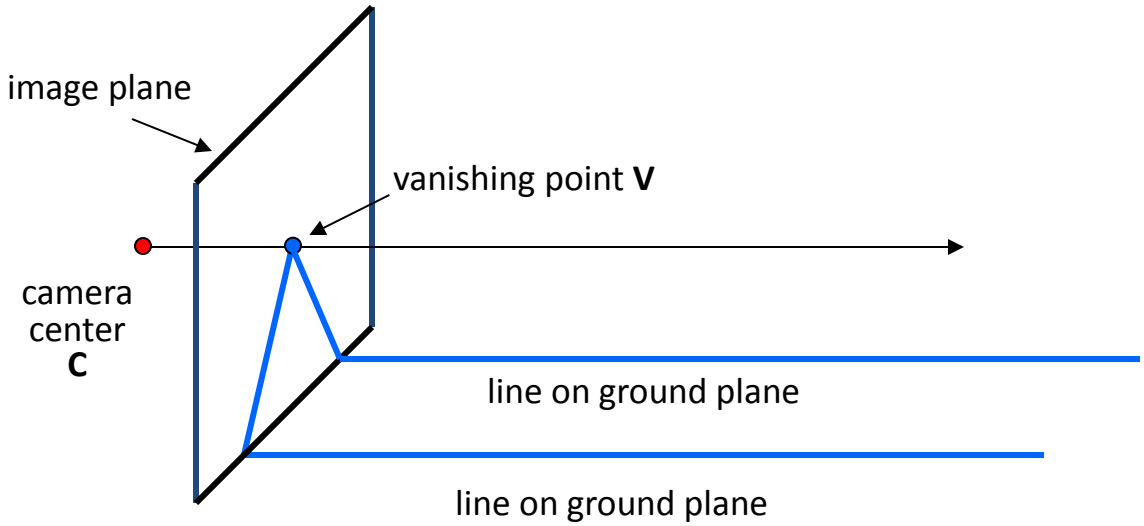


Effect of distance of projection center from image plane: Only the size not the shape of the ellipse changes!



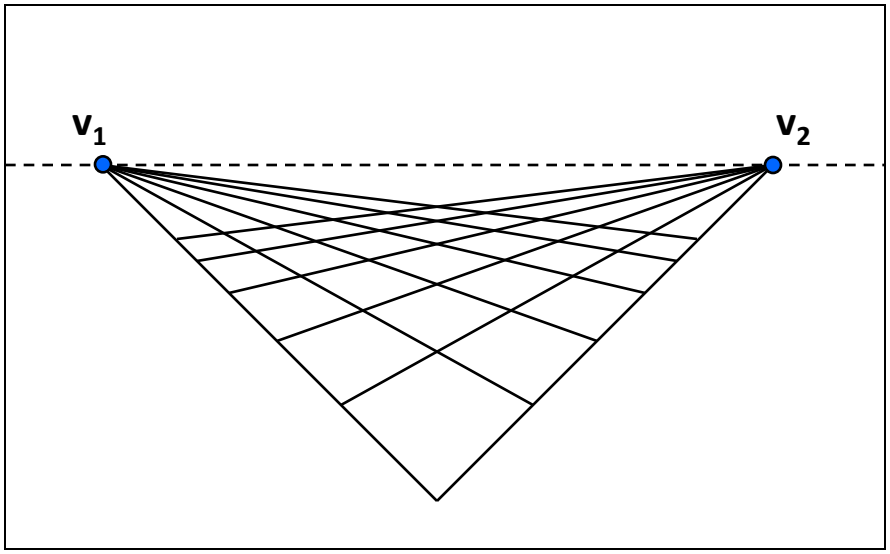


Vanishing points



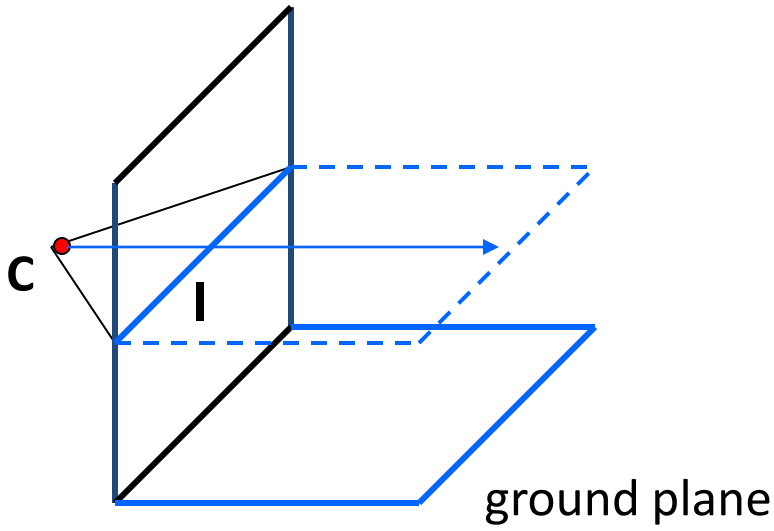
- Properties
  - Any two parallel lines have the same vanishing point
  - The ray from **C** through **v** point is parallel to the lines
  - An image may have more than one vanishing point

Vanishing lines



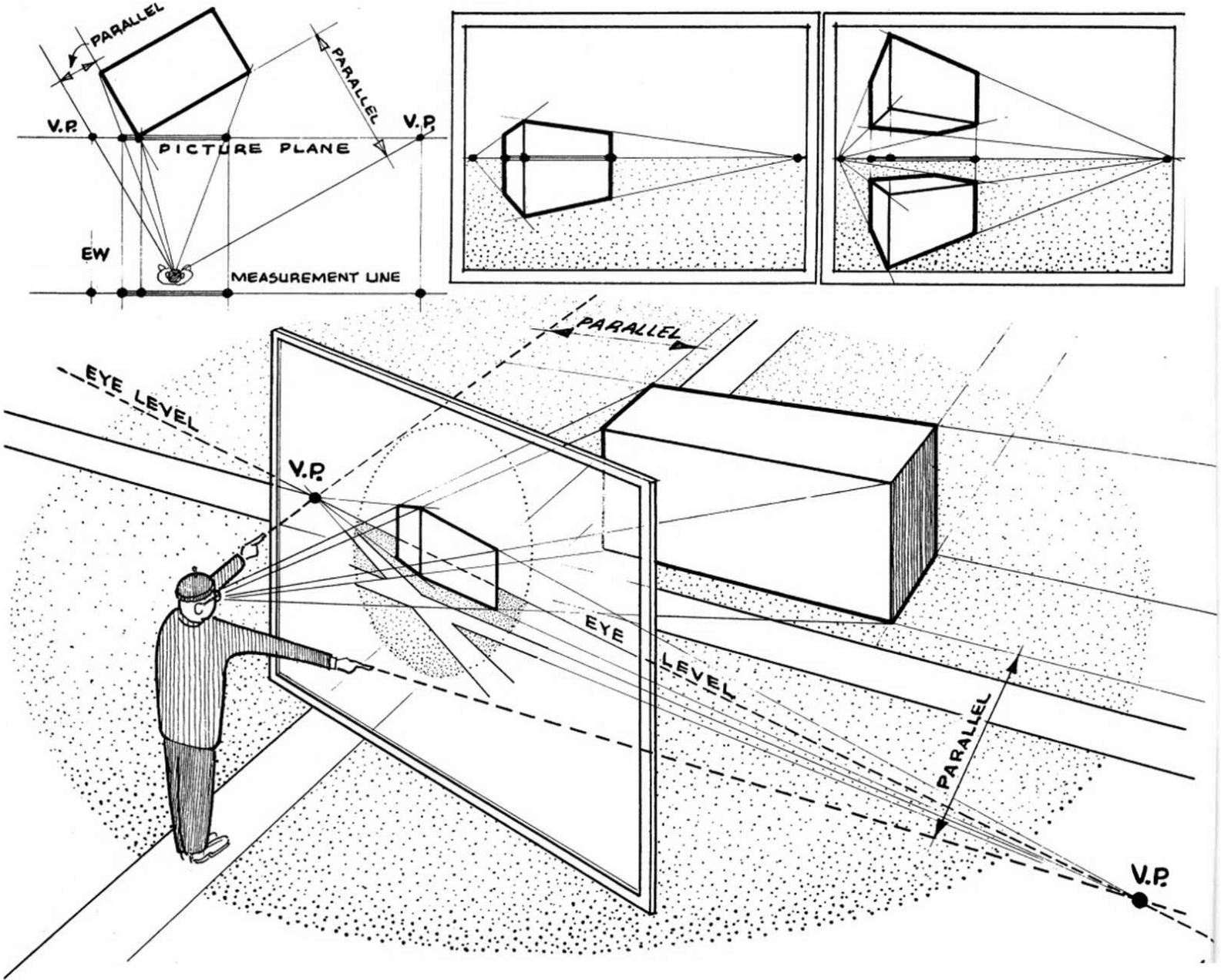
- Multiple Vanishing Points
  - Any set of parallel lines on the plane define a vanishing point
  - The union of all of these vanishing points is the *horizon line*
    - also called *vanishing line*
  - Note that different planes define different vanishing lines

Computing vanishing lines

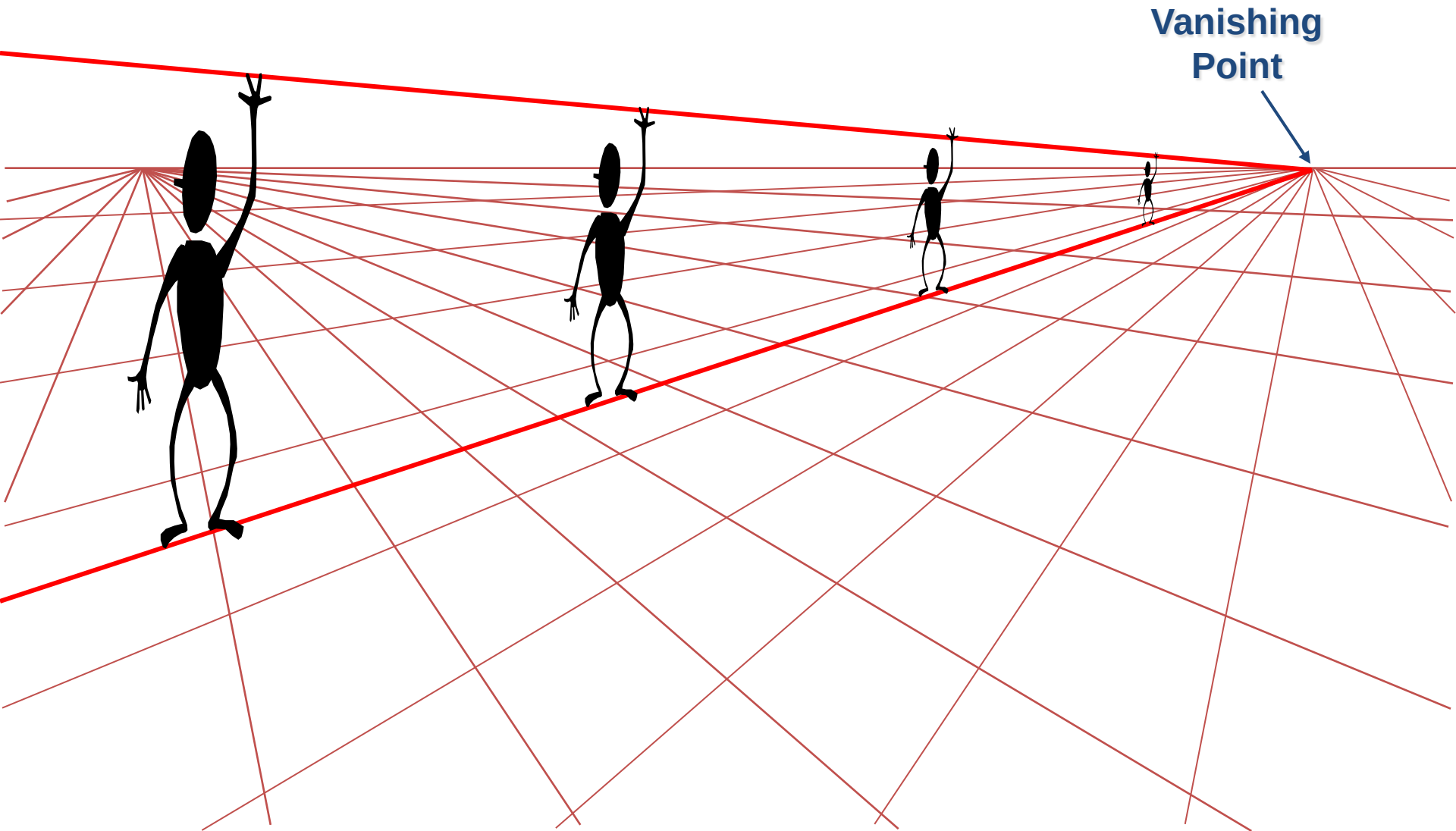


- Properties
  - **l** is intersection of horizontal plane through **C** with image plane
  - Compute **l** from two sets of parallel lines on ground plane
  - All points at same height as **C** project to **l**
  - Provides way of comparing height of objects in the scene

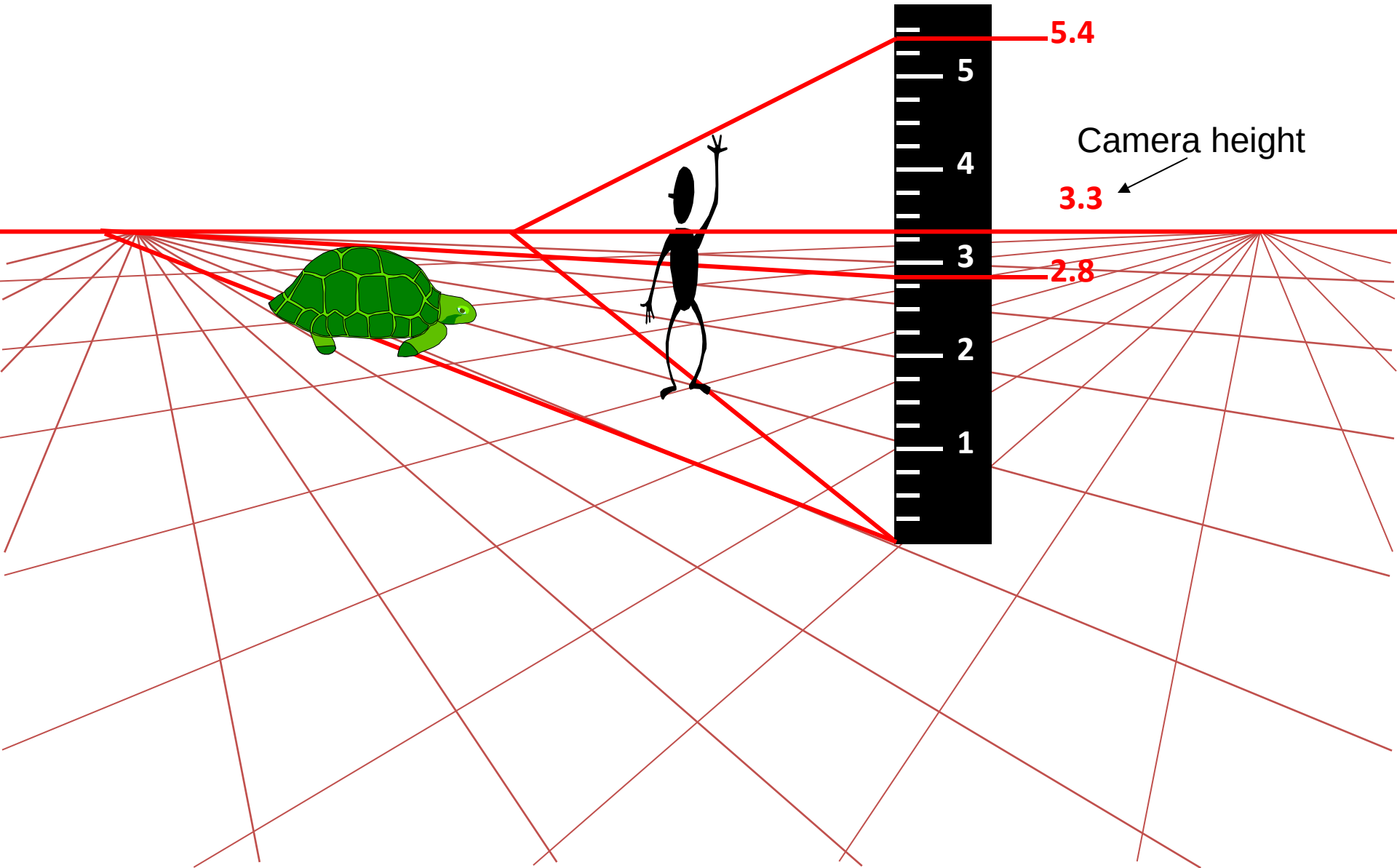
http://www.joshuanava.biz/perspective/in-other-words-the-observer-simply-points-in-the-same-direction-as-the-lines-in-order-to-find-their-vanishing-point.html



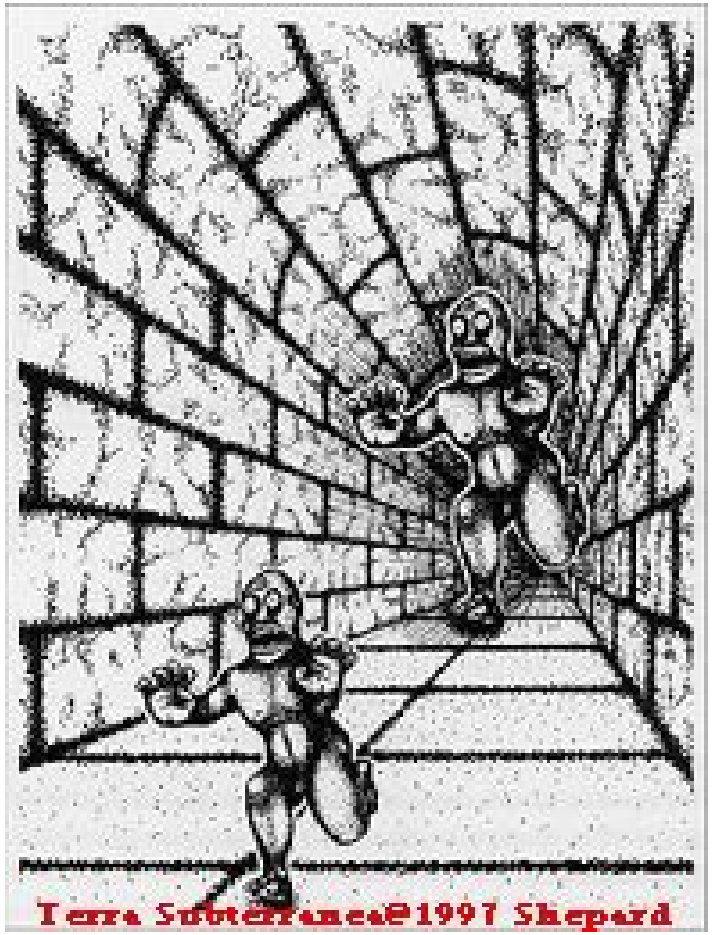
Comparing heights

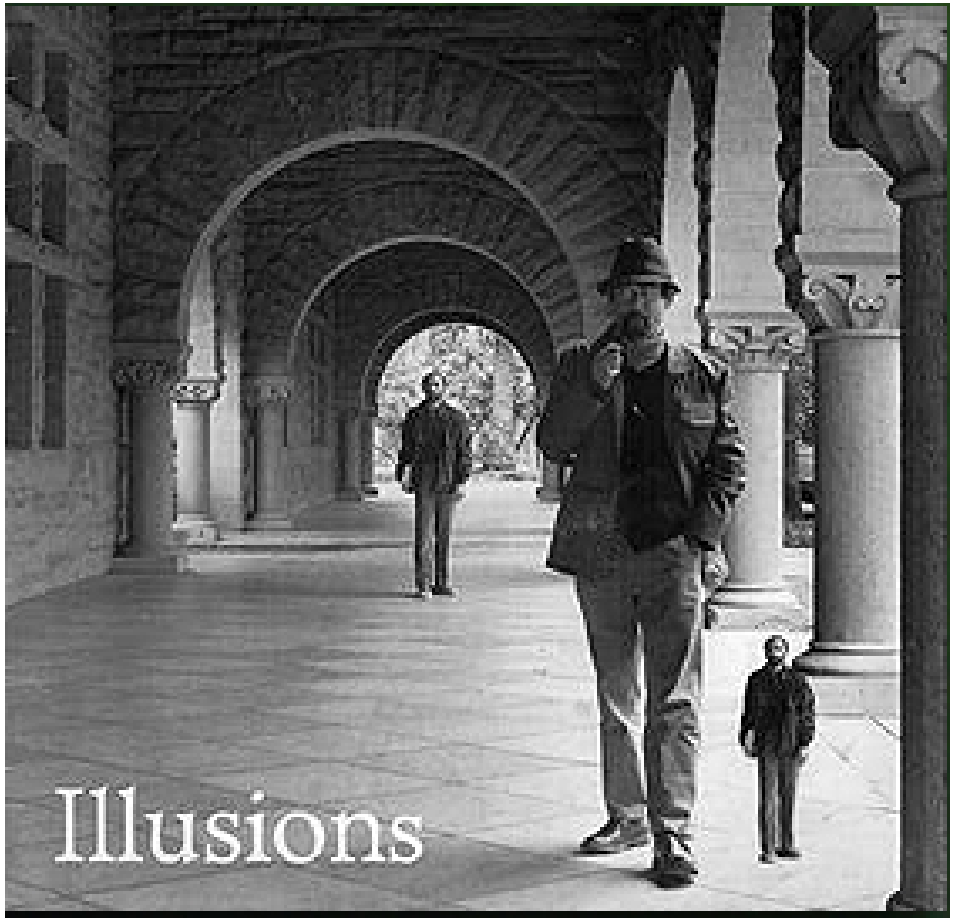


Measuring height

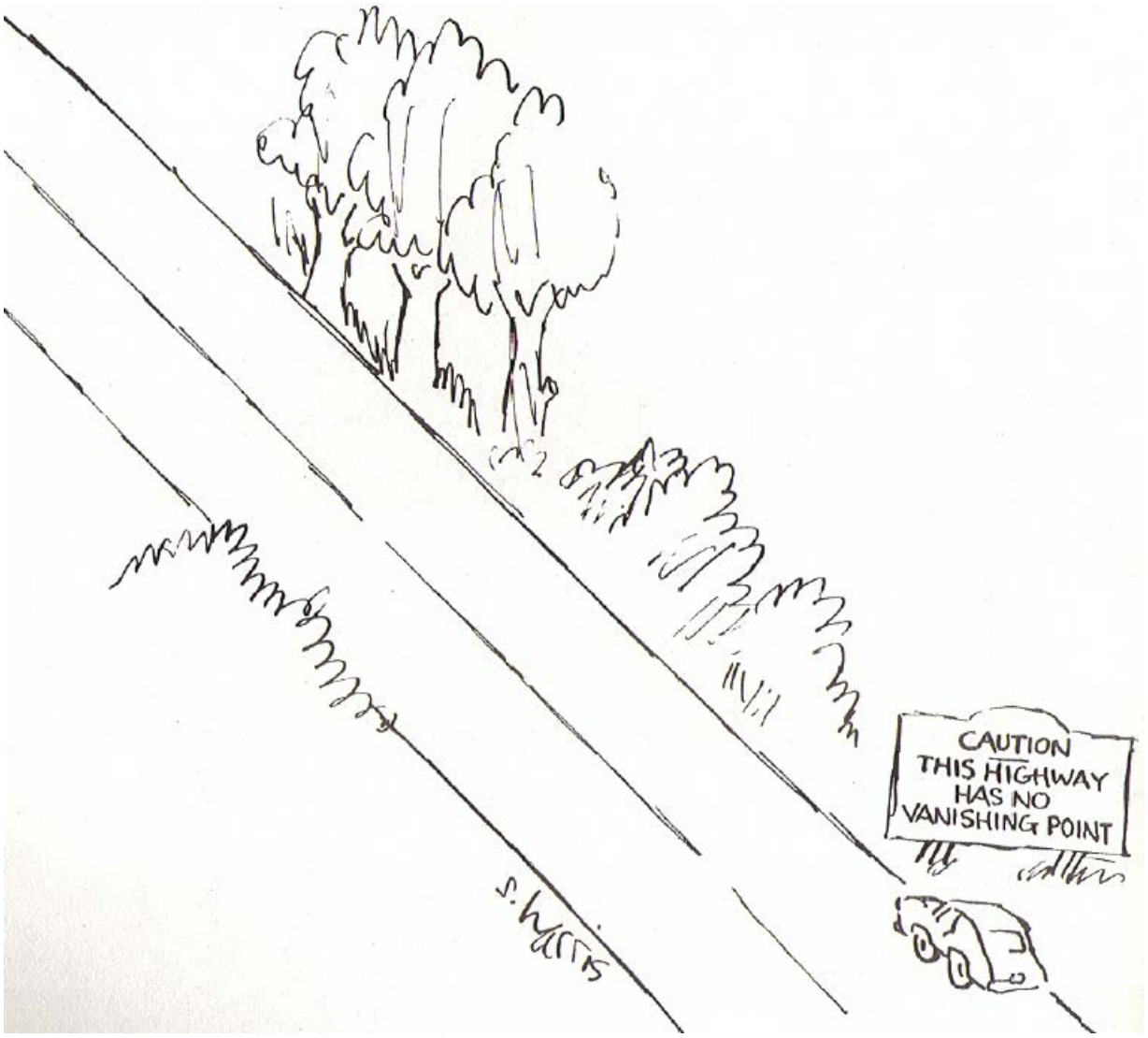


Fun with vanishing points

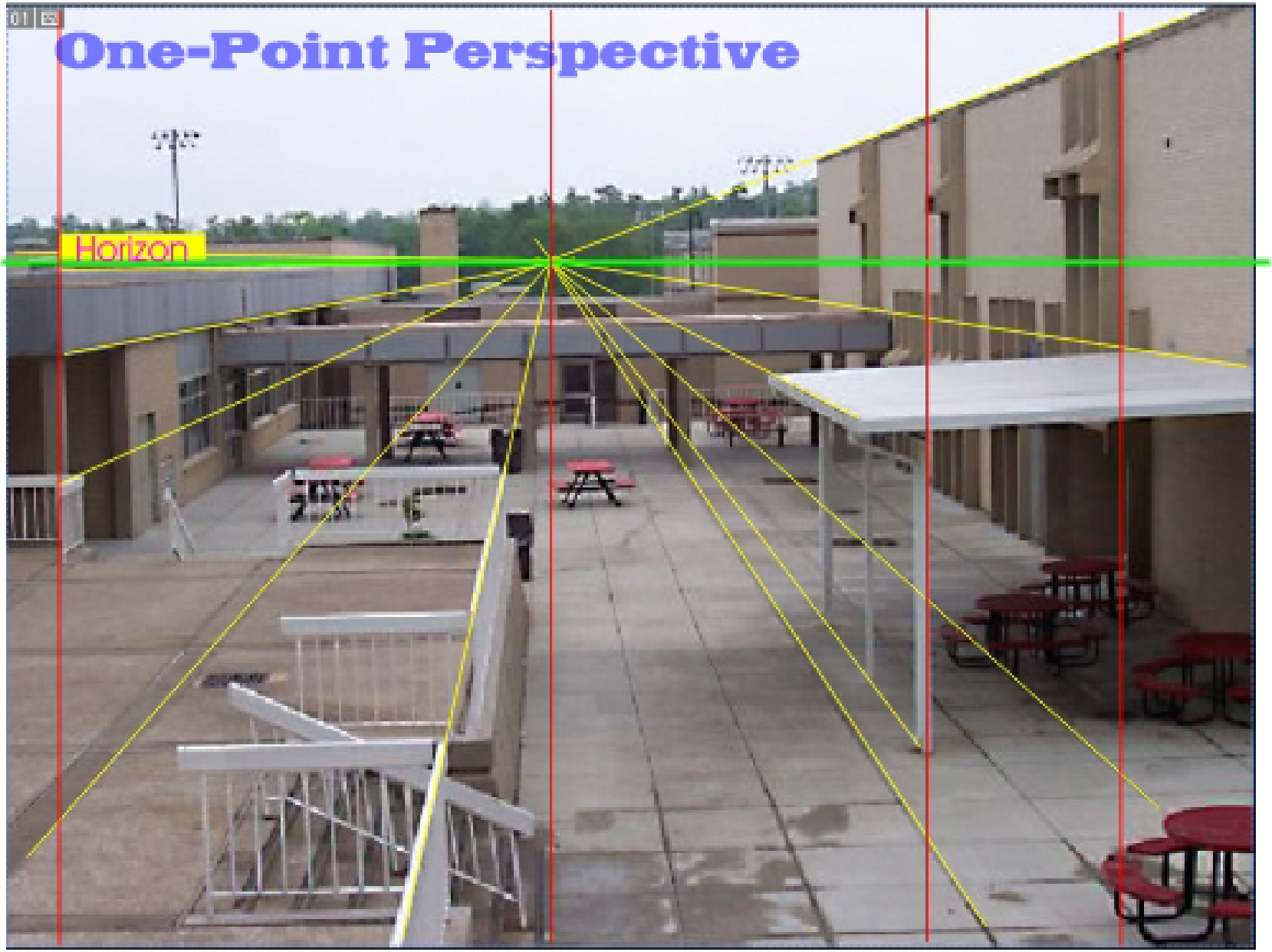




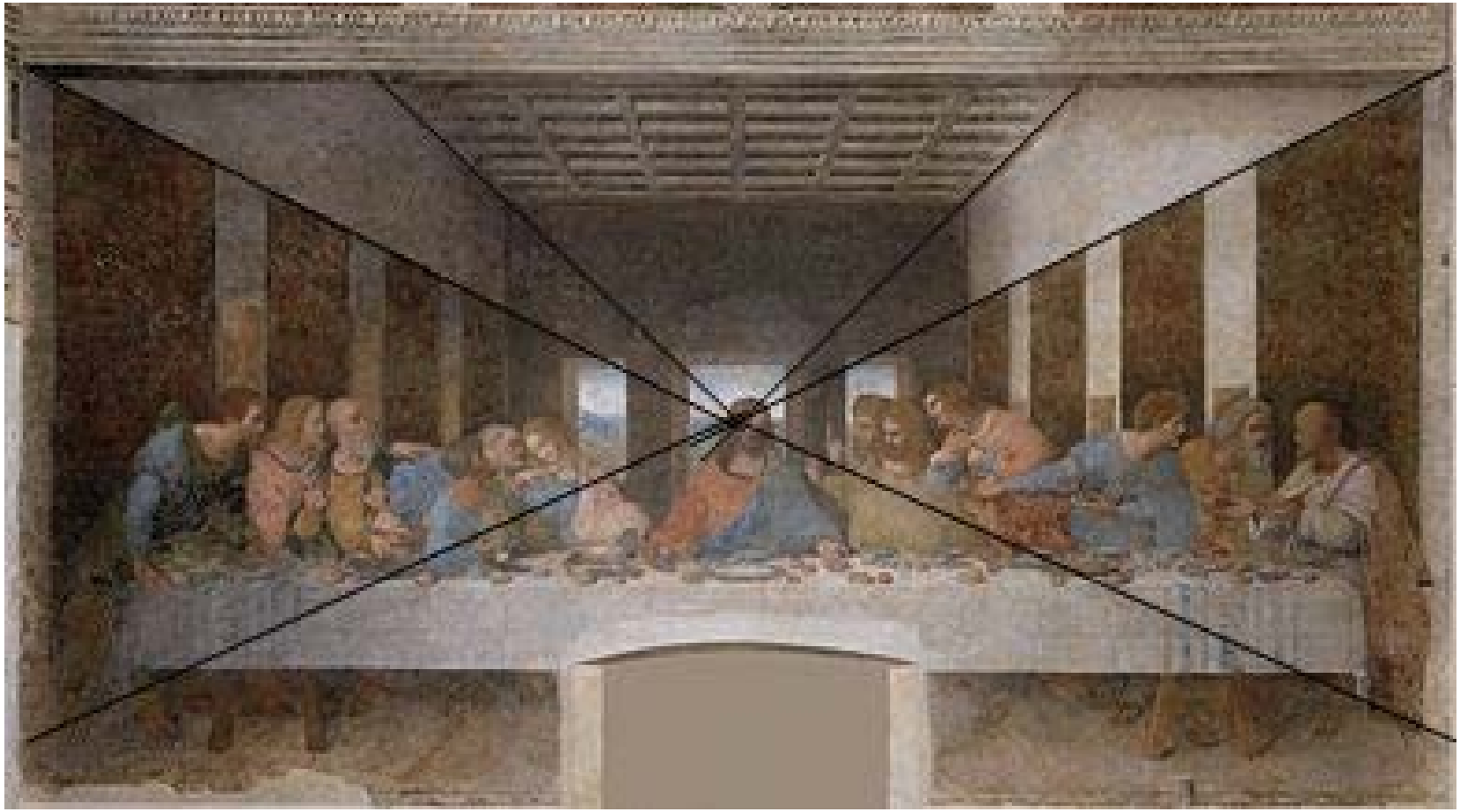
Vanishing point

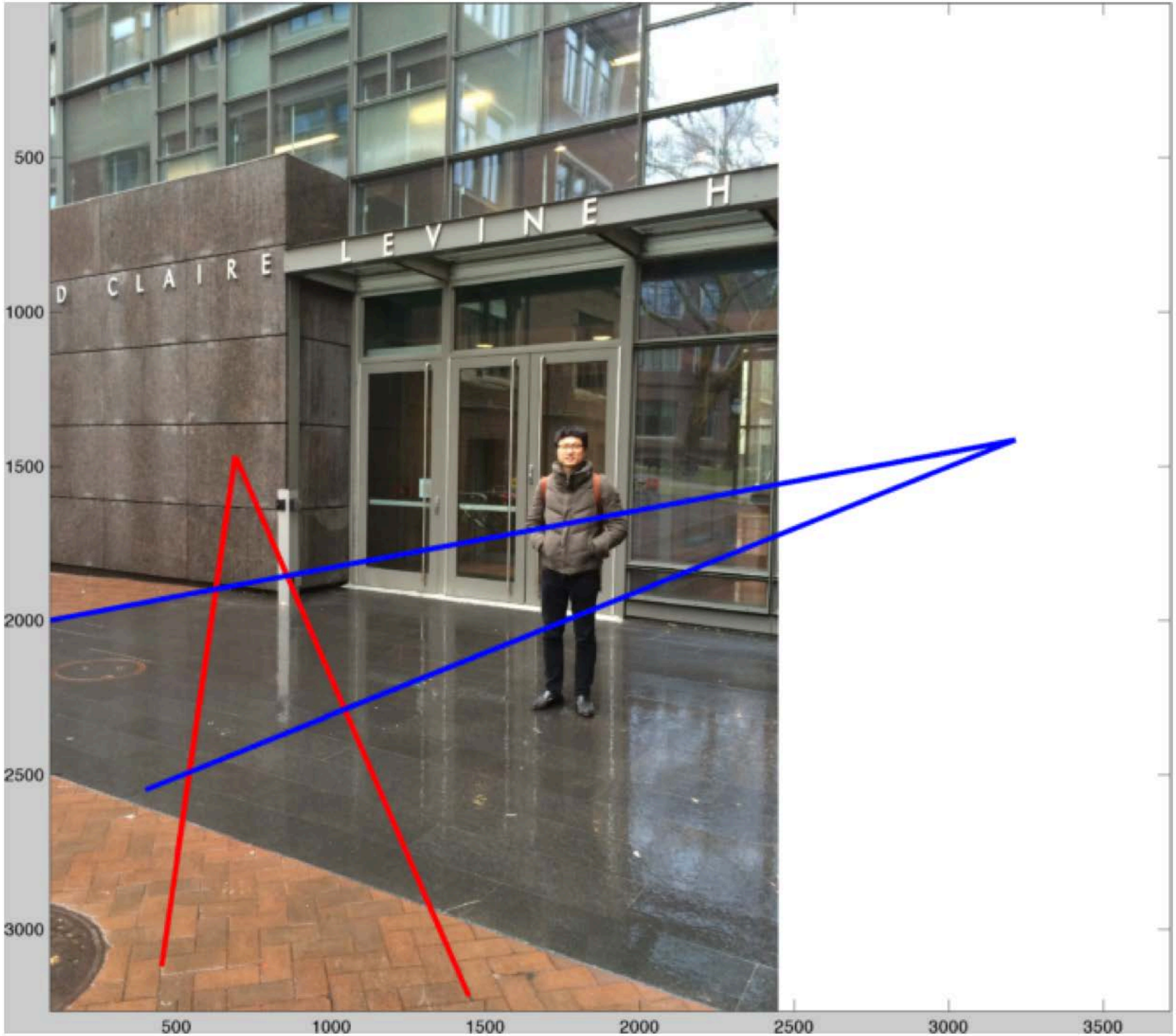


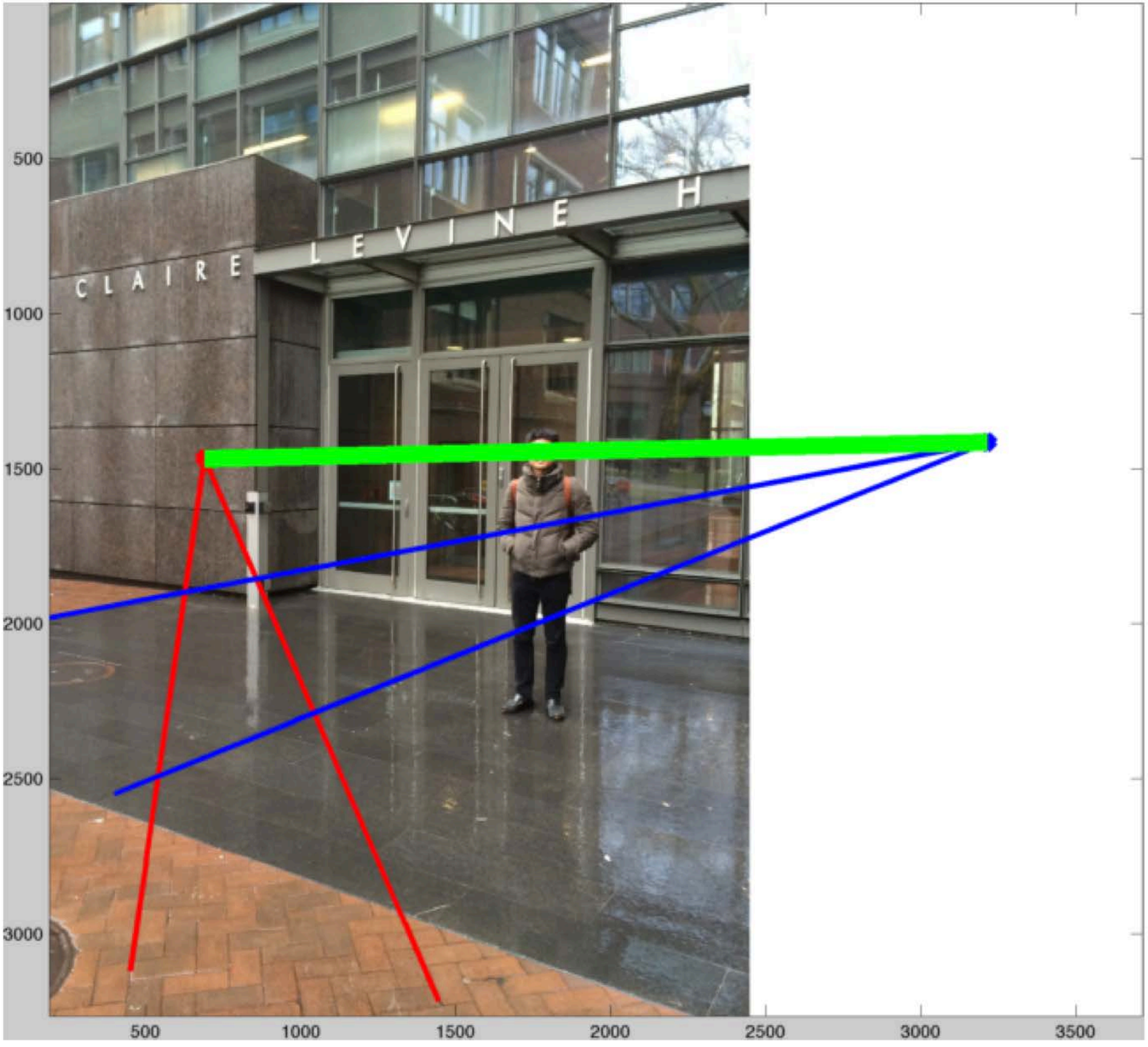
http://www.wetcanvas.com/



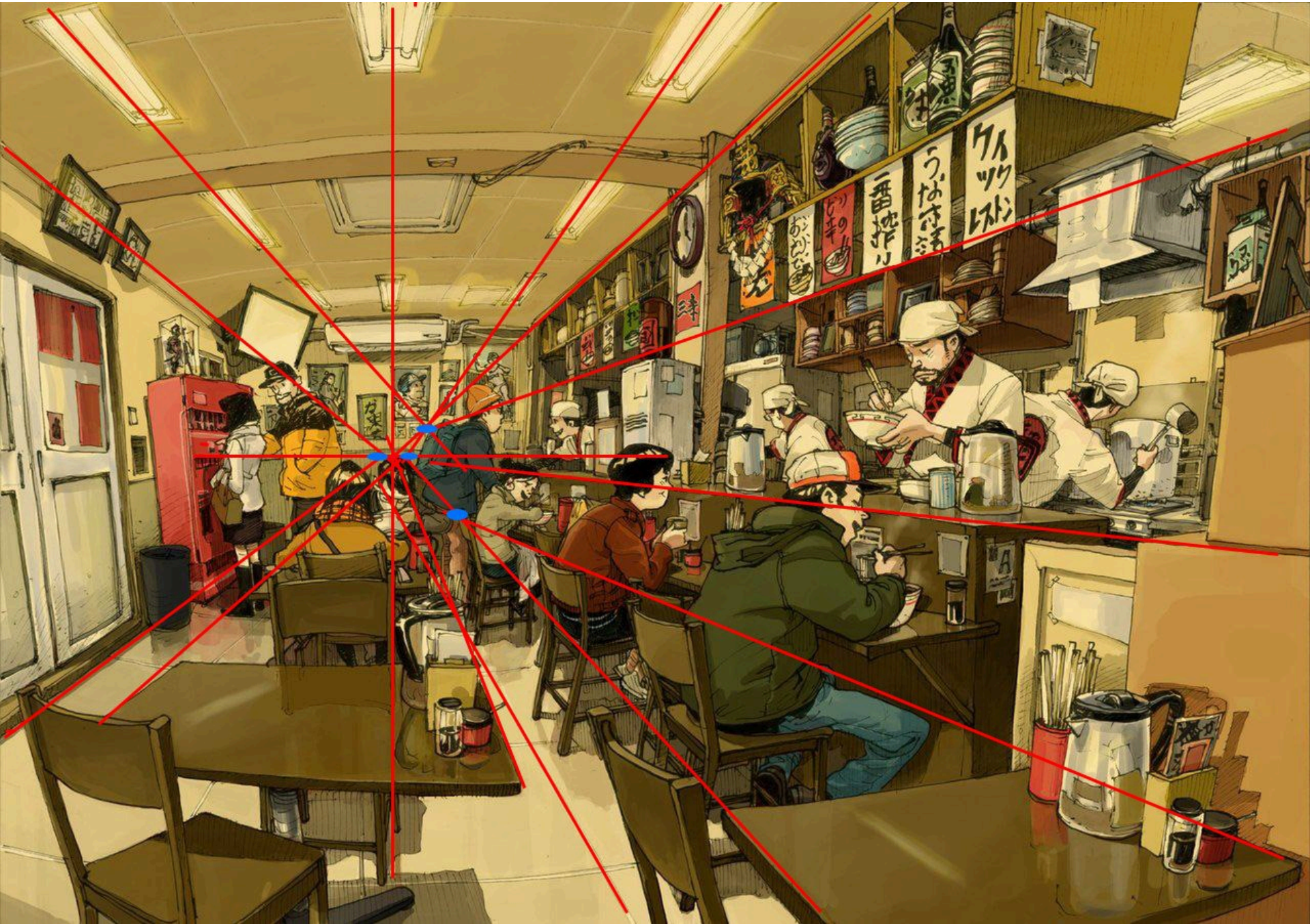
http://pennpaint.blogspot.com/



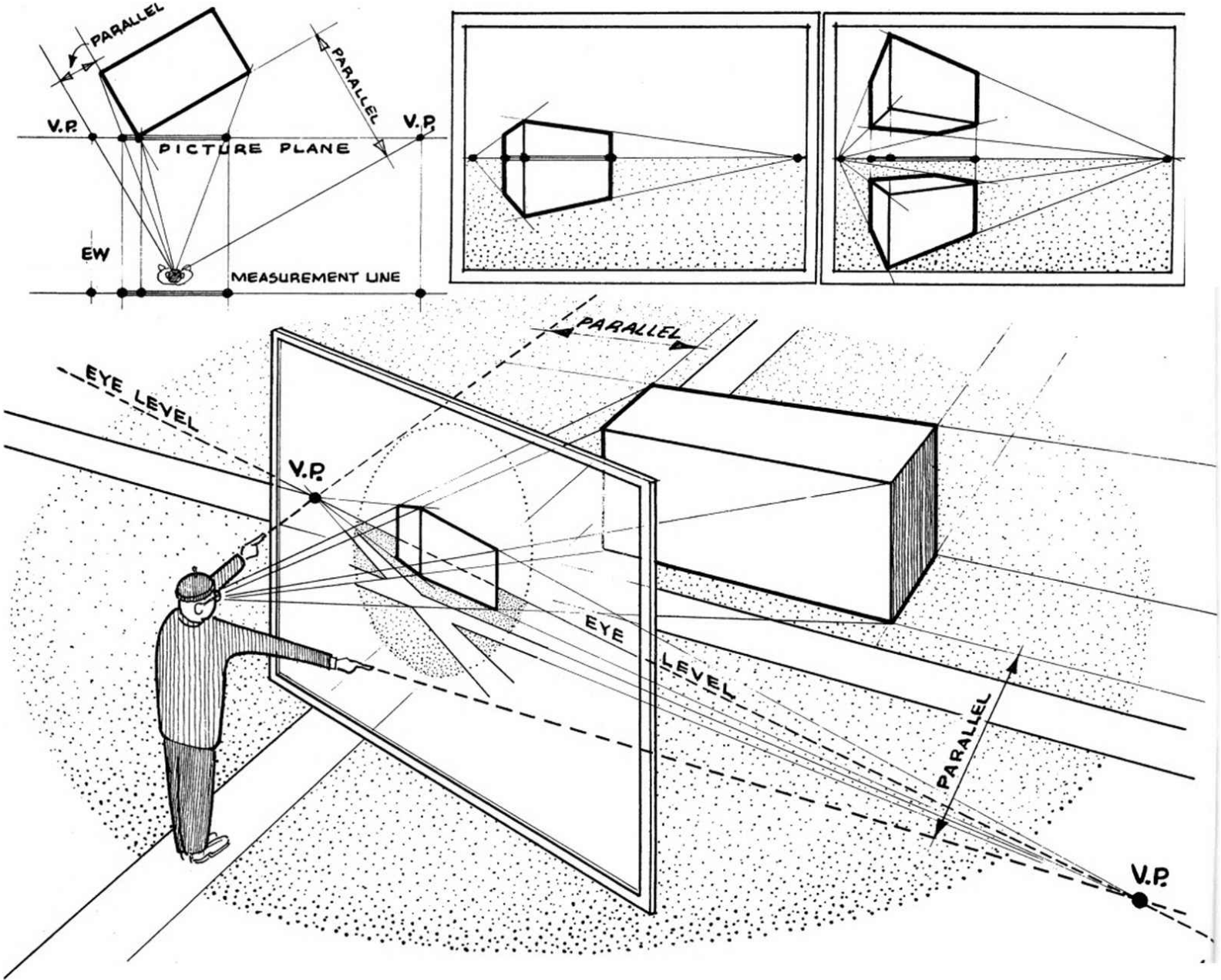

LEVINE H
D CLAIRE
500
1000
1500
2000
2500
3000
500
1000
1500
2000
2500
3000
3500


CLAIRE LEVINE H
500
1000
1500
2000
2500
3000
500
1000
1500
2000
2500
3000
3500

http://dd.salgoodsam.com/

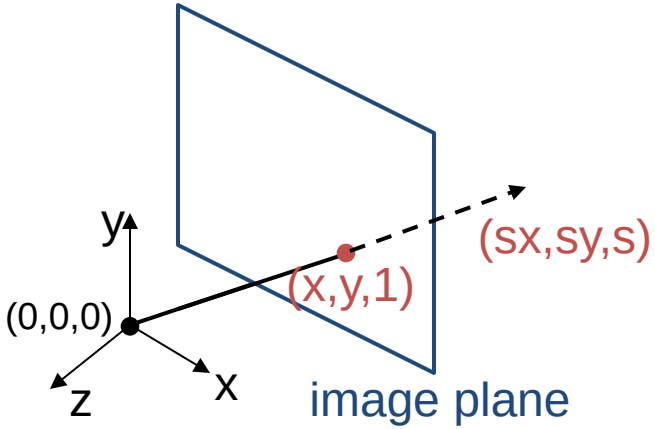


http://www.joshuanava.biz/perspective/in-other-words-the-observer-simply-points-in-the-same-direction-as-the-lines-in-order-to-find-their-vanishing-point.html



The projective plane

- Why do we need homogeneous coordinates?
  - represent points at infinity, homographies, perspective projection, multi-view relationships
- What is the geometric intuition?
  - a point in the image is a *ray* in projective space

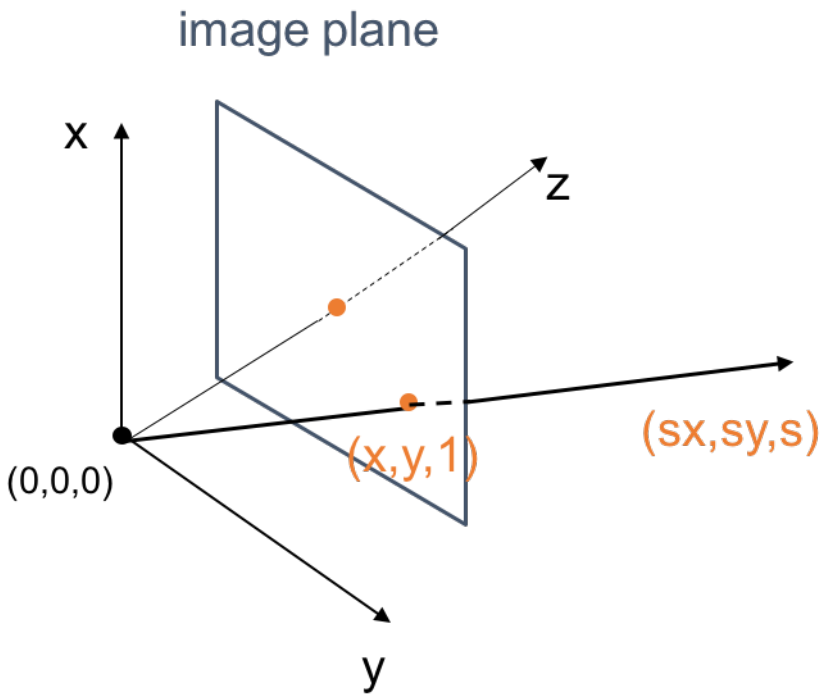


- Each *point* (x,y) on the plane is represented by a *ray* (sx,sy,s)
  - all points on the ray are equivalent: $(x, y, 1) \cong (sx, sy, s)$

Point

Homogeneous coordinates

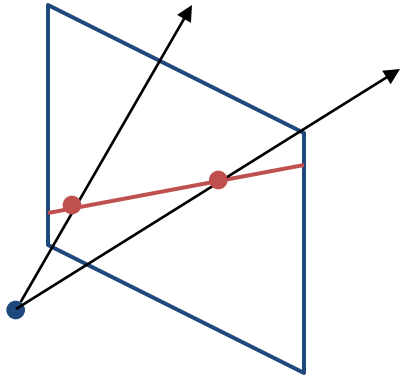
- represent coordinates in 2 dimensions with a 3-vector



$$\begin{bmatrix} x \\ y \end{bmatrix} \xrightarrow{\text{homogeneous coords}} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Projective lines

- What does a line in the image correspond to in projective space?



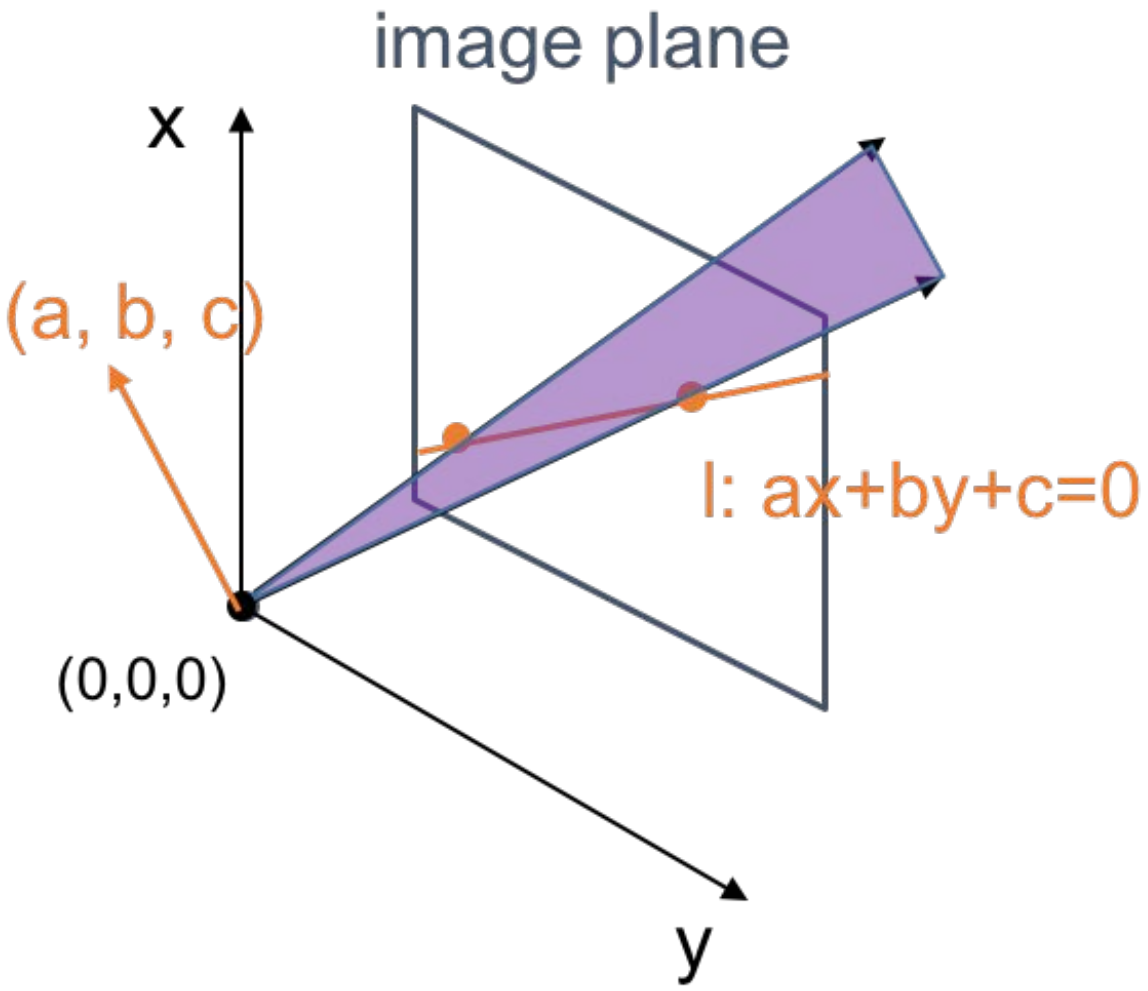
  - A line is a *plane* of rays through origin
    - all rays (x,y,z) satisfying: ax + by + cz = 0

$$\text{in vector notation}: \quad 0 = \begin{bmatrix} a & b & c \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$\mathbf{l} \qquad \mathbf{p}$$

  - A line is also represented as a homogeneous 3-vector **l**

Projective Lines



Line Representation

- a line is $\rho = x\cos\theta + y\sin\theta$

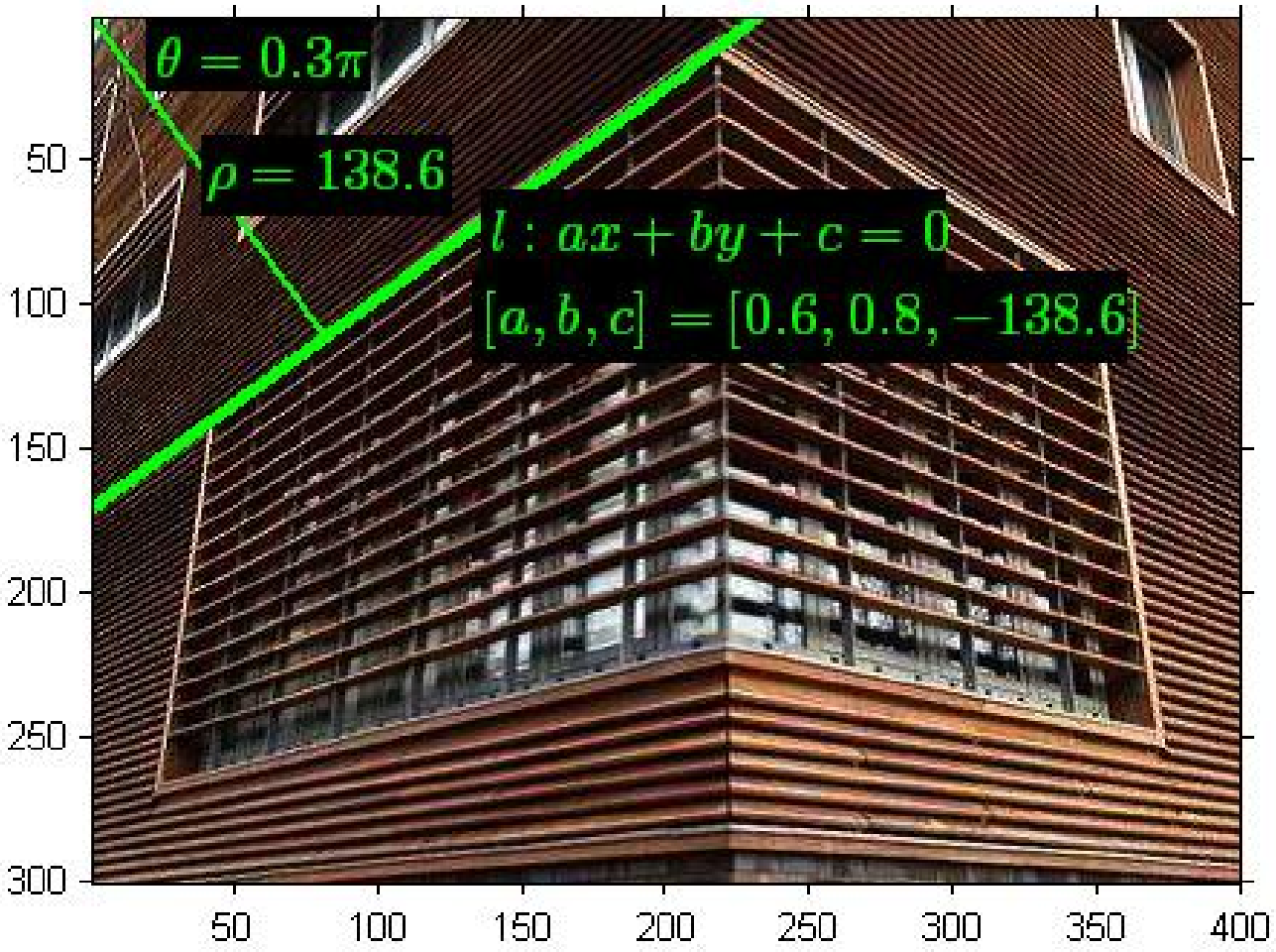
- ρ is the distance from the origin to the line
- θ is the norm direction of the line

- It can also be written as

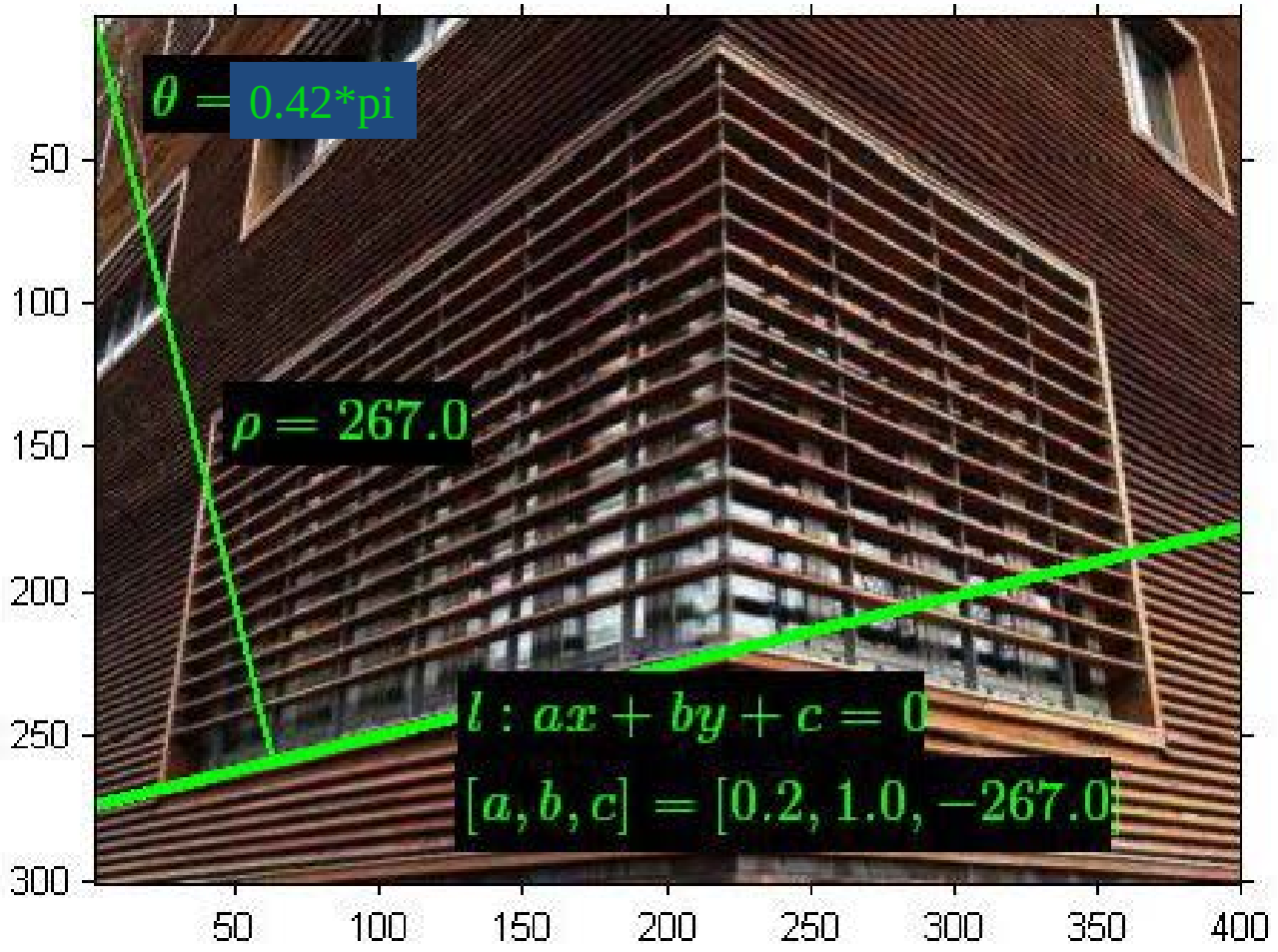
$$ax + by + c = 0;$$

$$\begin{aligned}\cos\theta &= \frac{a}{\sqrt{a^2+b^2}} \\ \sin\theta &= \frac{b}{\sqrt{a^2+b^2}} \\ \rho &= -\frac{c}{\sqrt{a^2+b^2}}\end{aligned}$$

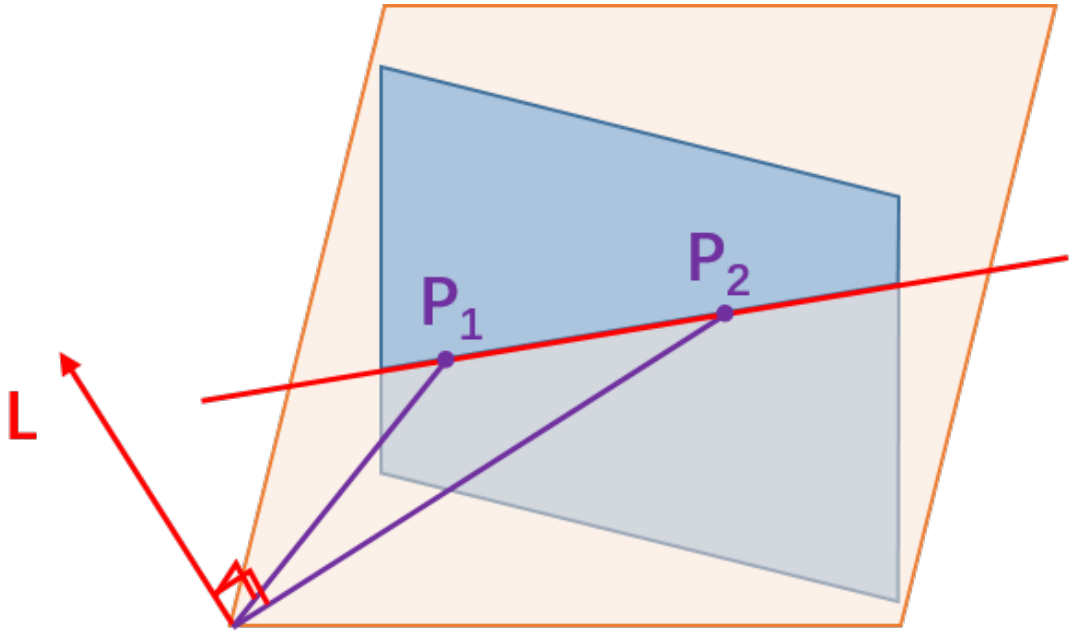
Example of Line



Example of Line (2)



Projective lines from two points



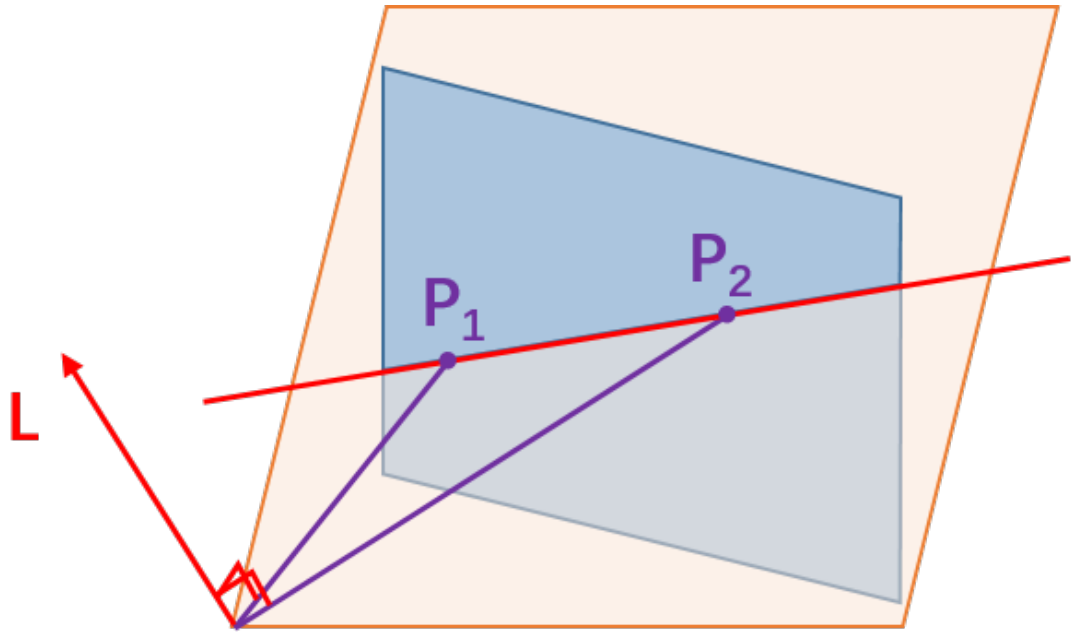
Line passing through two points

Two points: x x'

Define a line

l is the line passing two points

$$l = x \times x'$$



Line passing through two points

Two points: $x \quad x'$

Define a line

$$l = x \times x'$$

l is the line passing two points

Proof:

$$x \cdot (x \times x') = 0$$
$$x' \cdot (x \times x') = 0$$

$$x \cdot l = 0$$
$$x' \cdot l = 0$$

Line passing through two points

- N

$$
\begin{aligned}
l &= x \times x' \\
&= \begin{vmatrix} i & j & k \\ x_1 & x_2 & x_3 \\ x'_1 & x'_2 & x'_3 \end{vmatrix} \\
&= (x_2x'_3 - x_3x'_2)i \\
&\quad +(x_3x'_1 - x_1x'_3)j \\
&\quad +(x_1x'_2 - x_2x'_1)k \\
&= (x_2x'_3 - x_3x'_2, x_3x'_1 - x_1x'_3, x_1x'_2 - x_2x'_1)^T
\end{aligned}
$$

Matlab codes

- function l = get_line_by_two_points(x, y)

- x1 = [x(1), y(1), 1]';
- x2 = [x(2), y(2), 1]';
- l = cross(x1, x2);
- l = l / sqrt(l(1)*l(1)+l(2)*l(2));

Example of Line

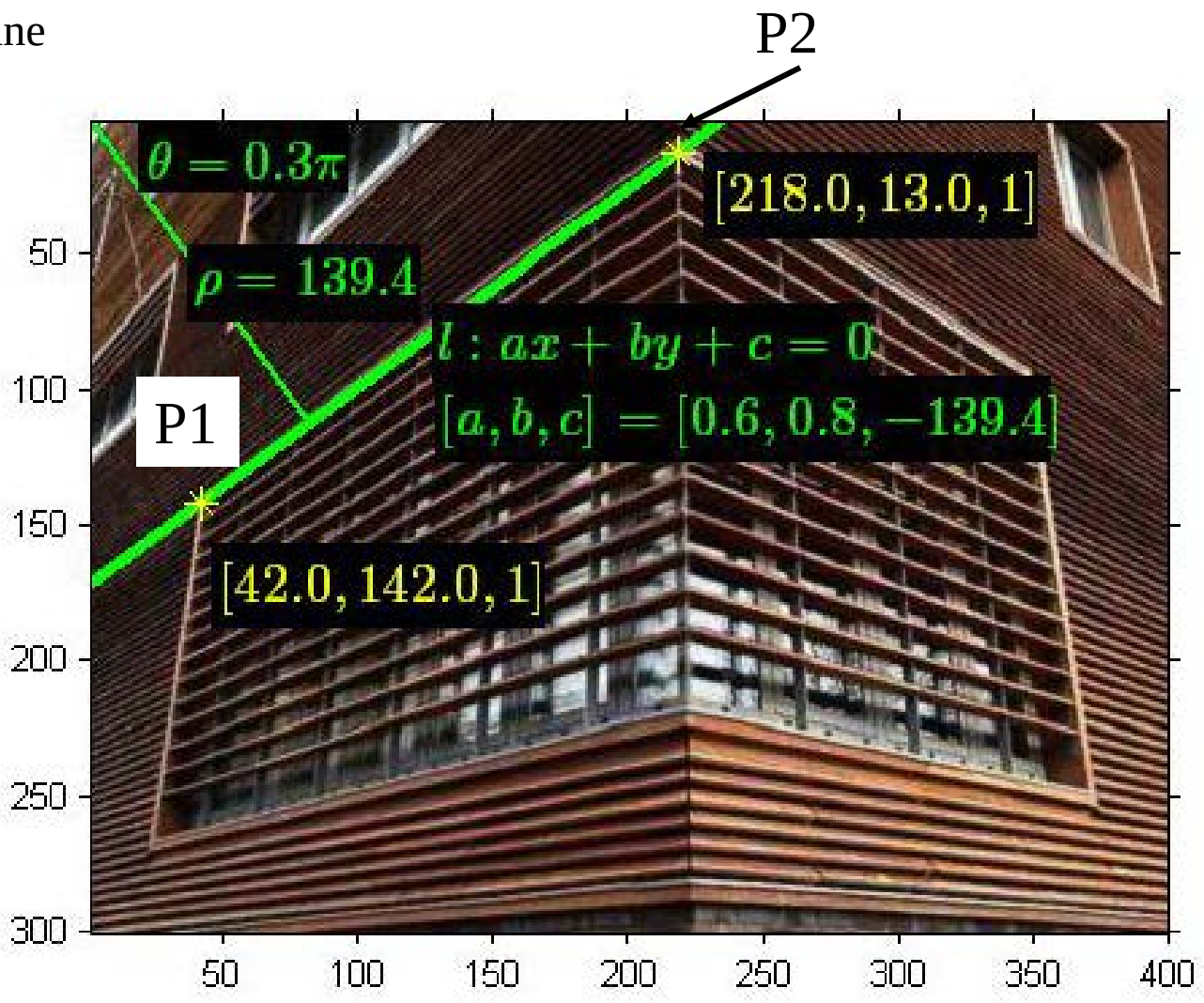
Verify p2 lies on the line

```
>> 218*0.6+13*0.8

ans =

  141.2000
```

(close)

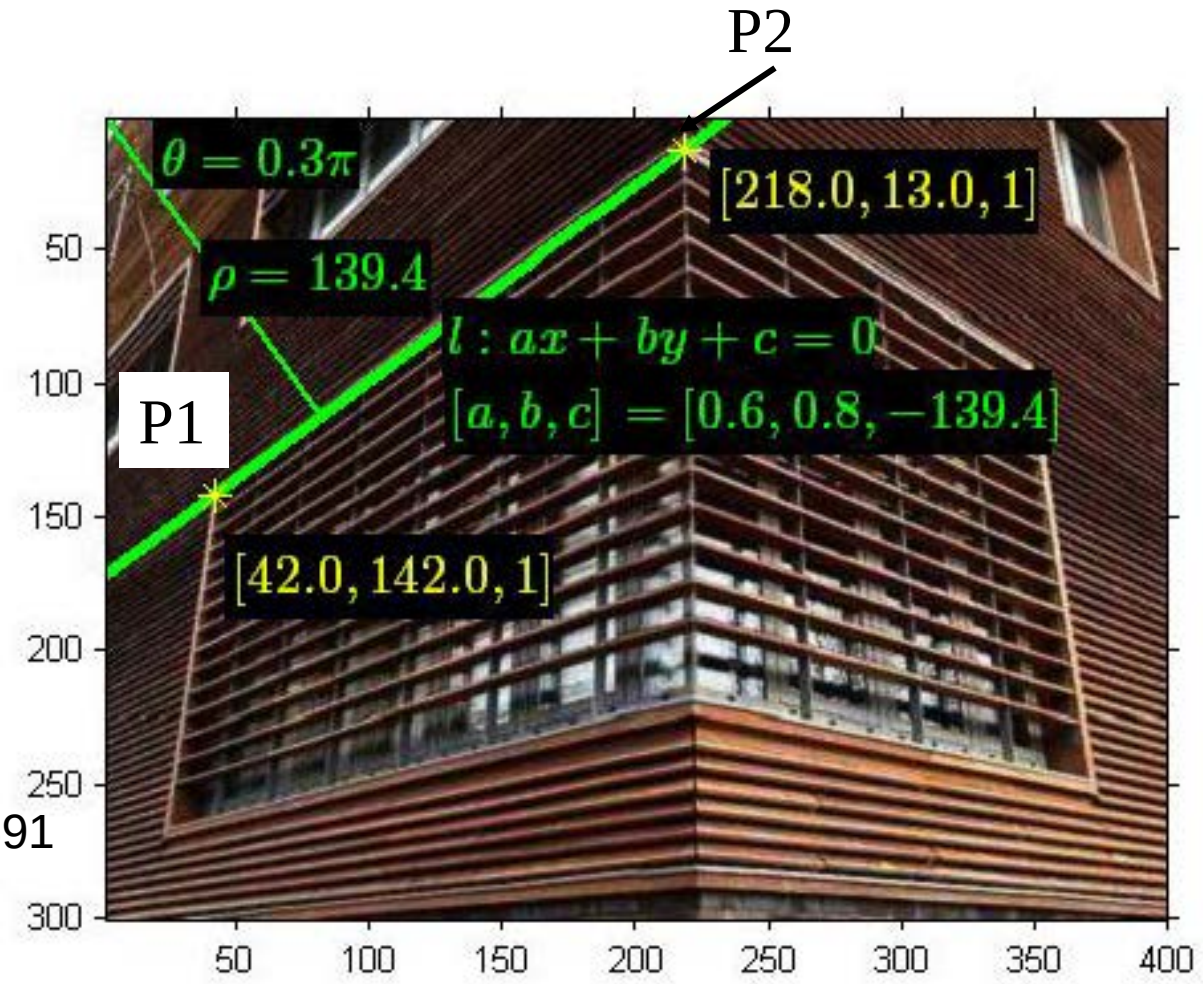


Example of Line

Test line:

```
>> p1 = [42,142,1];
>> p2=[218,13,1];
>> l = cross(p1,p2)
l =
129   176   -30410
```

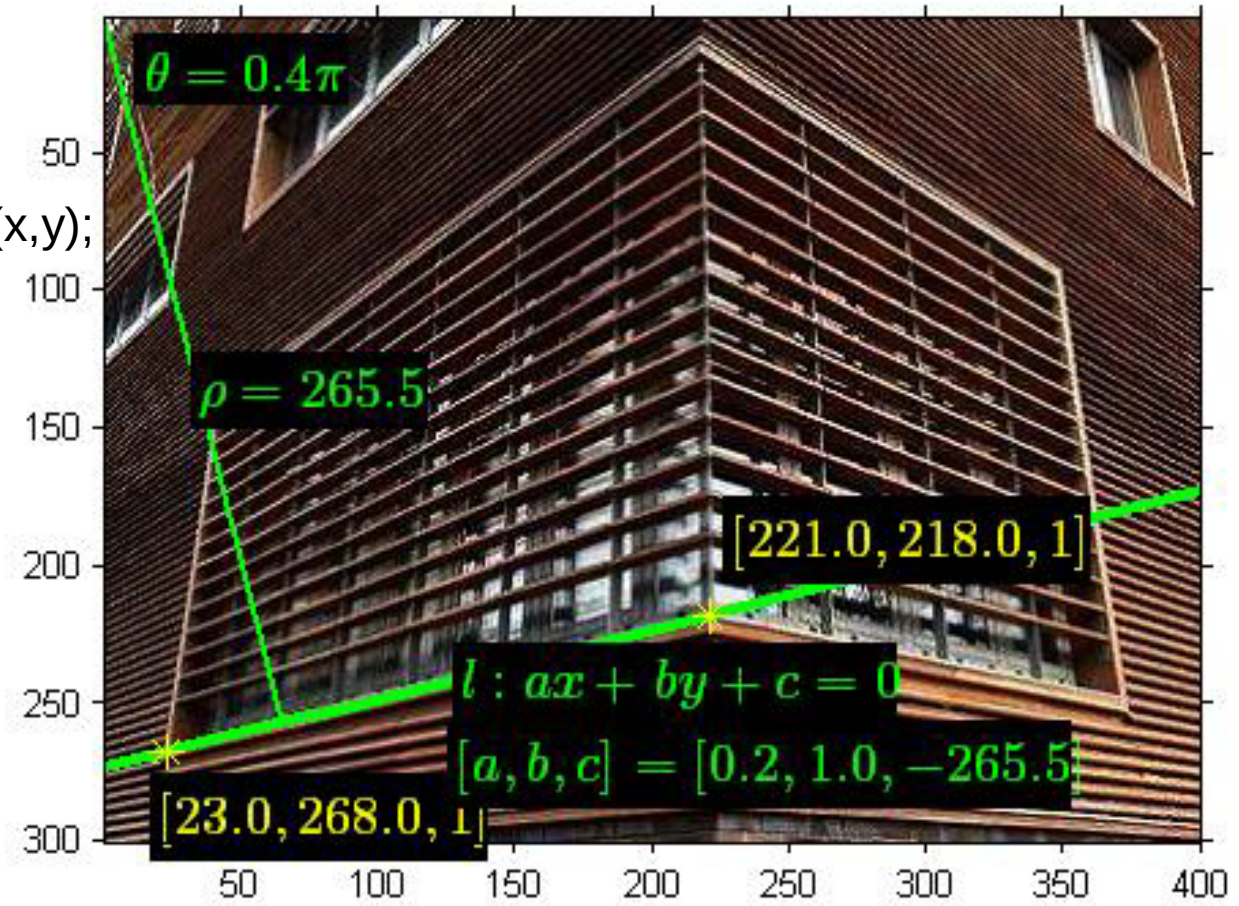
```
>> l = l/sqrt(l(1)^2+l(2)^2)
l =
 0.5912    0.8066 -139.3591
```



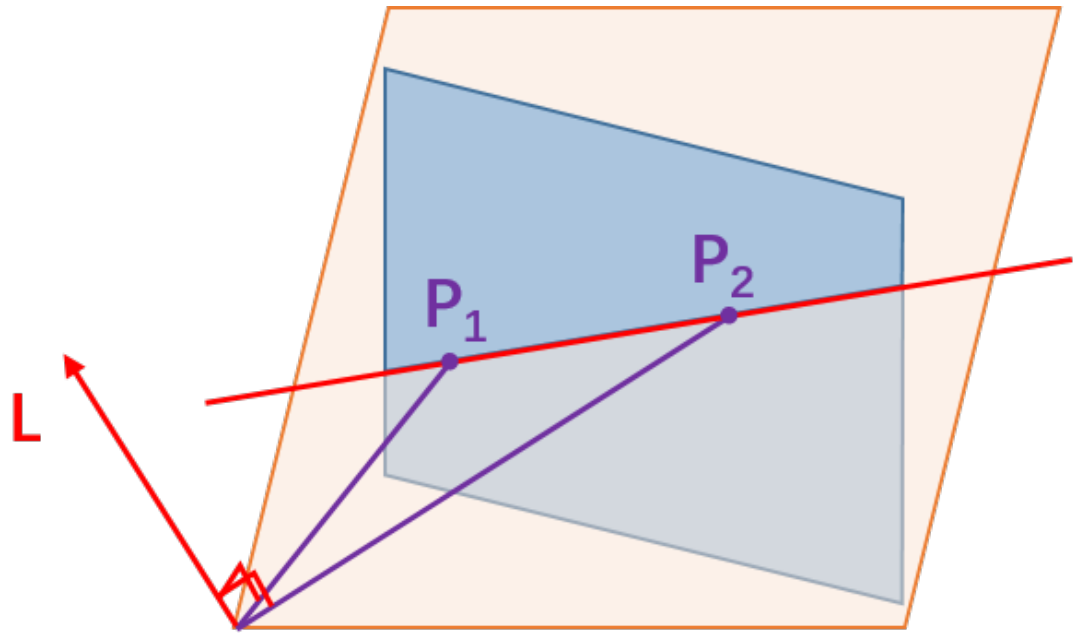
```
>> x = [221;23];
>> y = [218;268];
>> l = get_line_by_two_points(x,y);
>> l

l =


   -0.2448
   -0.9696
  265.4744
```

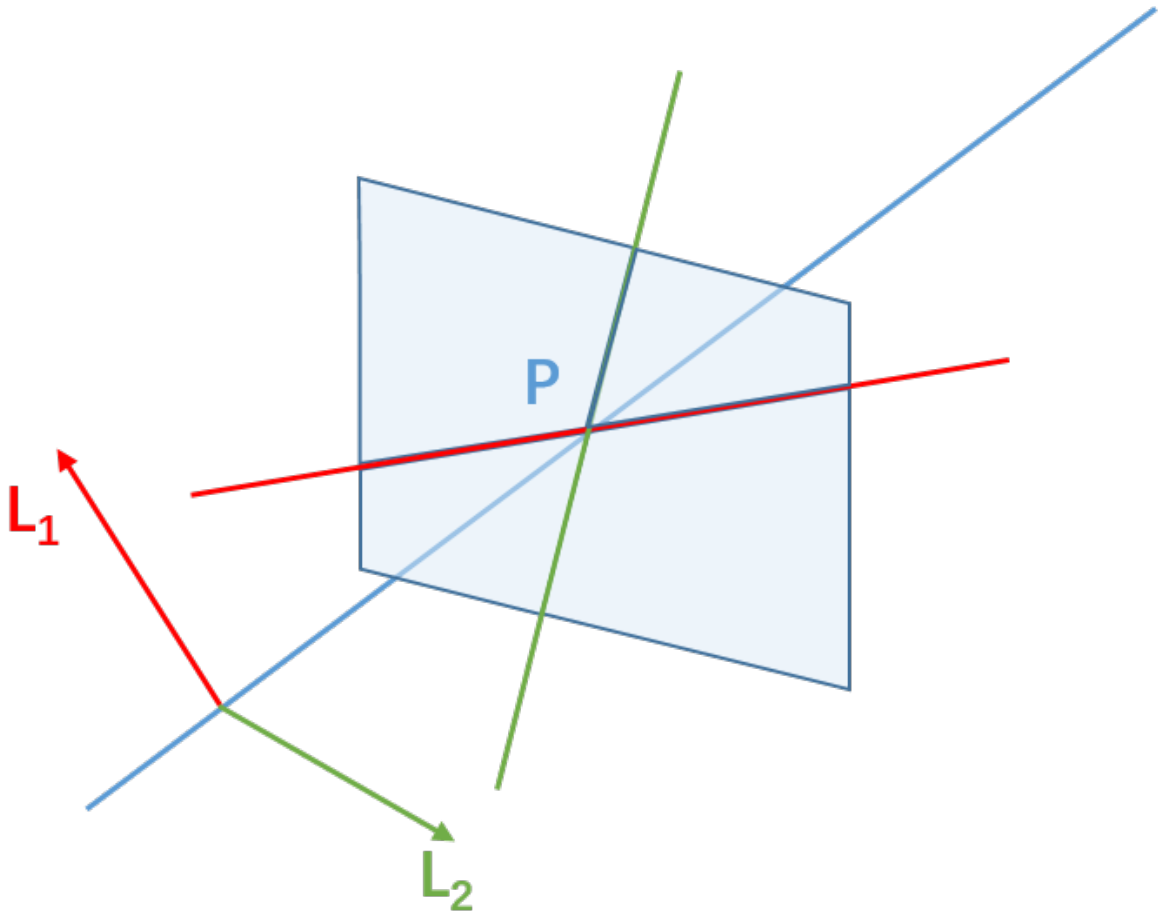


Projective lines from two points



When does the line has the form (a, b, 0)?
When does the line has the form (0, 0, 1)?

Points from two lines

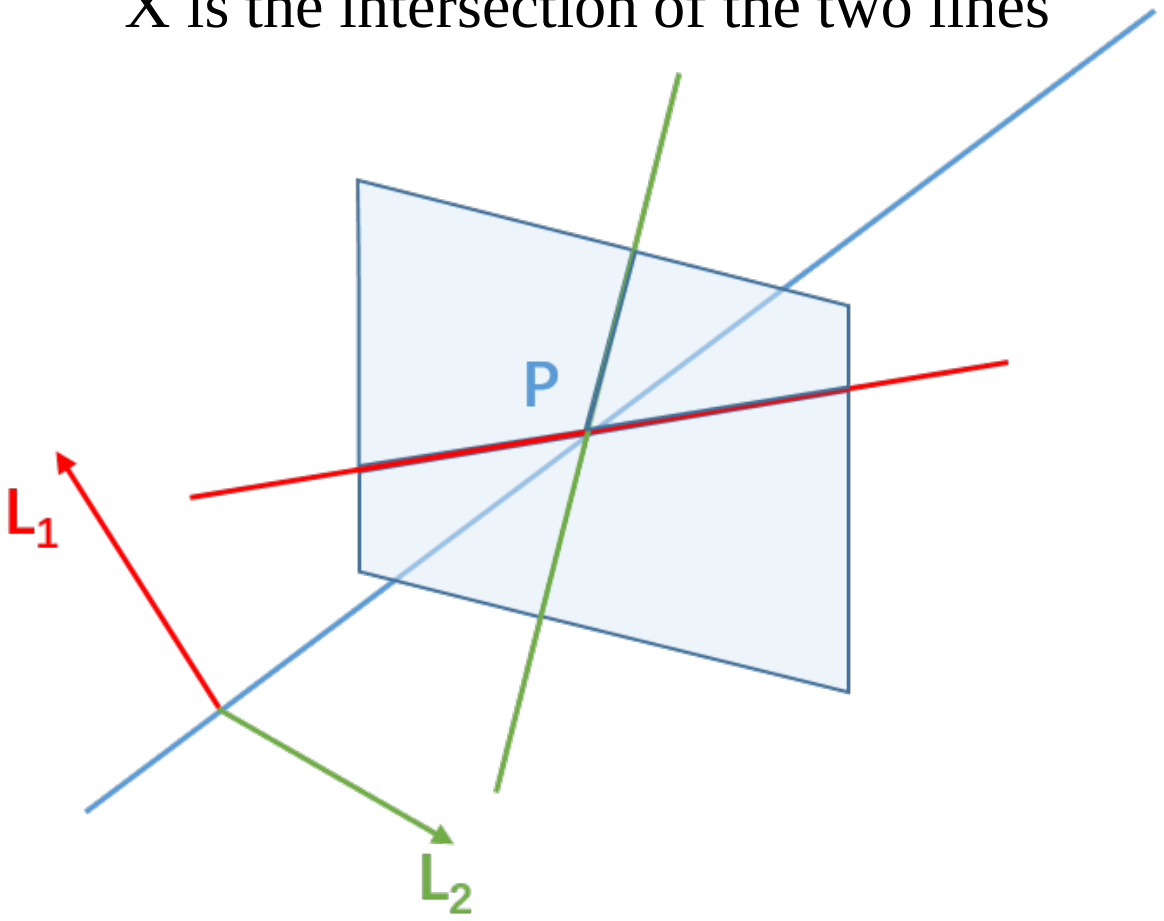


Intersection of lines

Given two lines: l , l'

Define a point $x = l \times l'$

X is the intersection of the two lines

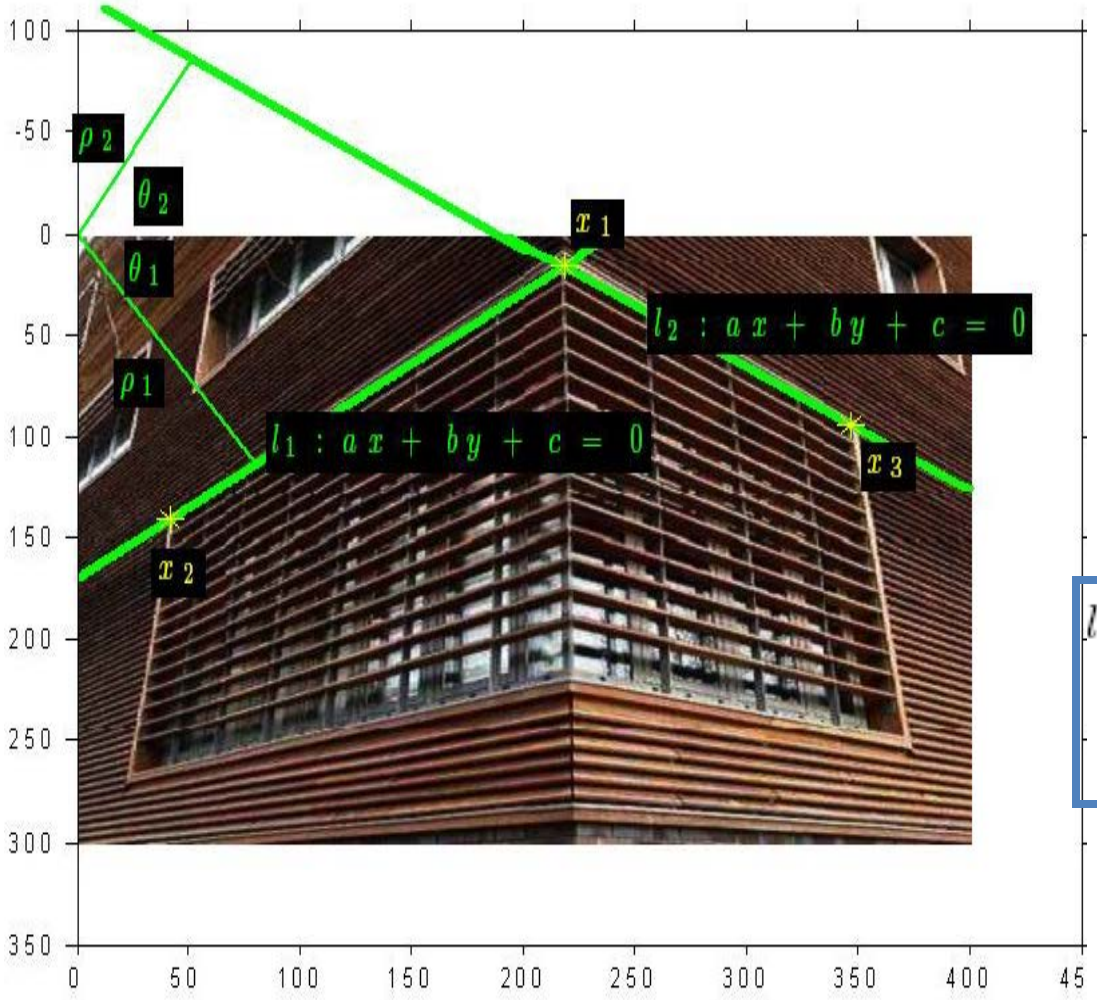


Matlab codes

- function x0 = get_point_by_two_line(l, l1)

- x0 = cross(l, l1);
- x0 = [x0(1)/x0(3); x0(2)/x0(3)];

Example of Lines Intersection



$$
\begin{aligned}
x_1 &= (218, 16, 1)^T \\
x_2 &= (42, 141, 1)^T \\
x_3 &= (347, 94, 1)^T
\end{aligned}
$$

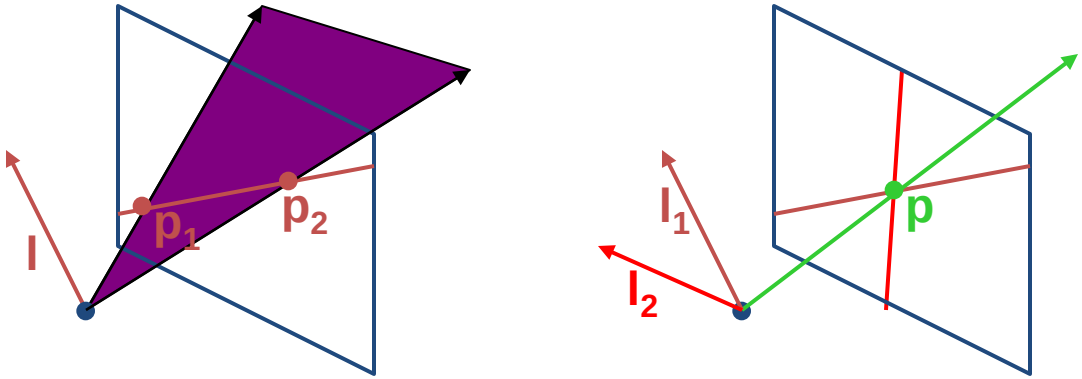
$$
\begin{aligned}
l_1 &= x_1 \times x_2 \\
&= k(-0.58, -0.82, 139.28)^T \\
l_2 &= x_1 \times x_3 \\
&= k(0.52, -0.86, -99.11)^T
\end{aligned}
$$

$$
\begin{aligned}
l_1 \times l_2 &= (-199.98, -14.68, -0.92)^T \\
&= -0.92(218.0, 16.0, 1.0)^T \\
&= kx_1
\end{aligned}
$$

Point and line duality

– A line **l** is a homogeneous 3-vector

– It is ⊥ to every point (ray) **p** on the line: **l p**=0



What is the line **l** spanned by rays $\mathbf{p}_1$ and $\mathbf{p}_2$?

- **l** is ⊥ to $\mathbf{p}_1$ and $\mathbf{p}_2$ $\Rightarrow$ $\mathbf{l} = \mathbf{p}_1 \times \mathbf{p}_2$
- **l** is the plane normal

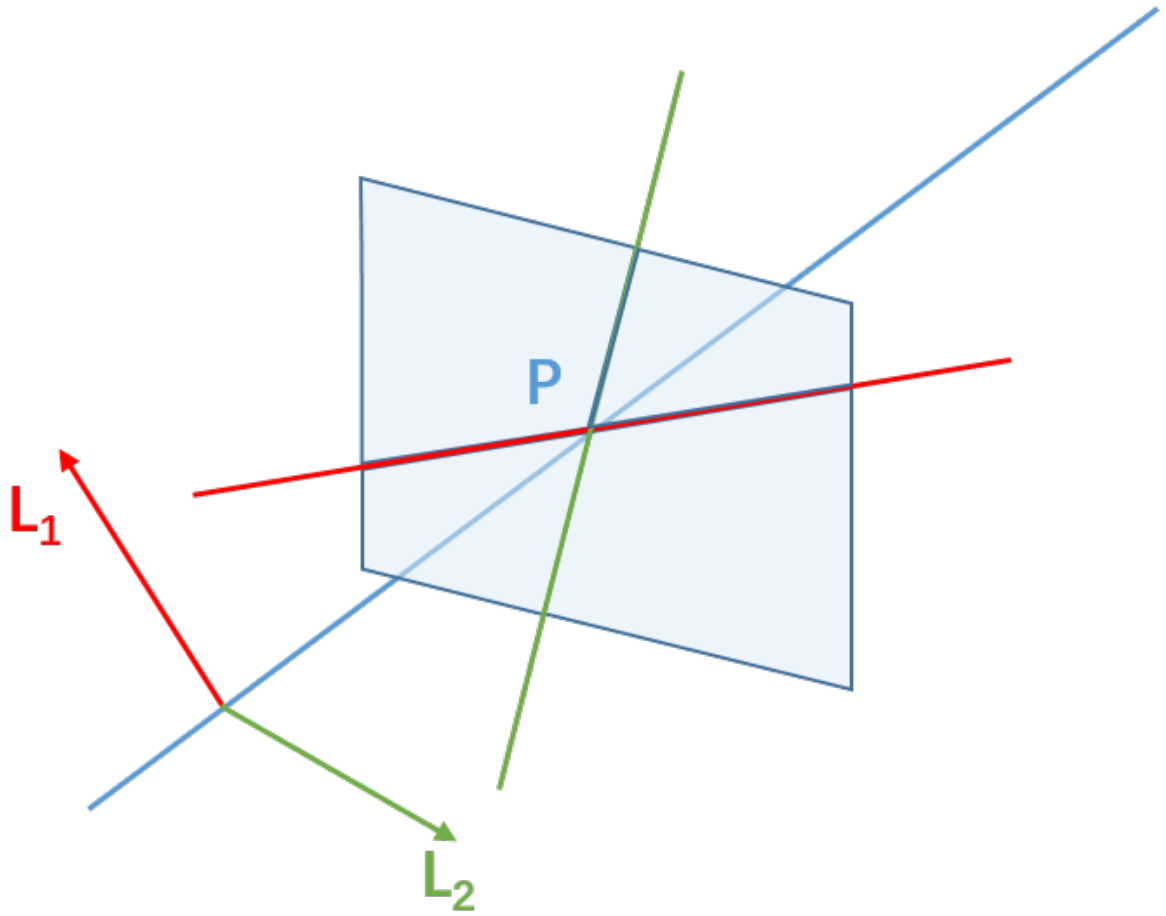
What is the intersection of two lines $\mathbf{l}_1$ and $\mathbf{l}_2$?

- **p** is ⊥ to $\mathbf{l}_1$ and $\mathbf{l}_2$ $\Rightarrow$ $\mathbf{p} = \mathbf{l}_1 \times \mathbf{l}_2$

Points and lines are *dual* in projective space

- given any formula, can switch the meanings of points and lines to get another formula

Points from two lines



When P has the form (x,y,0)?

Point at infinity

Example: Consider two *parallel* horizontal lines:

$$x = 1; x = 2;$$

Intersection =

$$det[(i, j, k); (-1, 0, 1); (-1, 0, 2)]$$

$$= (0, 1, 0)$$

Point at infinity in the direction of y

Point at infinity, Ideal points

$$l = (a, b, c) \qquad l' = (a, b, c')$$

Intersection:

$$\begin{aligned} l \times l' &= l \times l' \\ &= \begin{vmatrix} i & j & k \\ a & b & c \\ a & b & c' \end{vmatrix} \\ &= (bc' - bc, ca - c'a, ab - ab)^T \\ &= (c' - c)(b, -a, 0)^T \end{aligned}$$

Any point (x_1,x_2,0) is intersection of lines at infinity

Points at infinity

- Under projective transformation,
  - All parallel lines intersects at the point at infinity

$$\text{line } l = (a, b, c)^T \text{ intersects at } (b, -a, 0)^T$$

  - One point at infinity ⇔ one parallel line direction

- Where are the points at infinity in the image plane?

Line at infinity

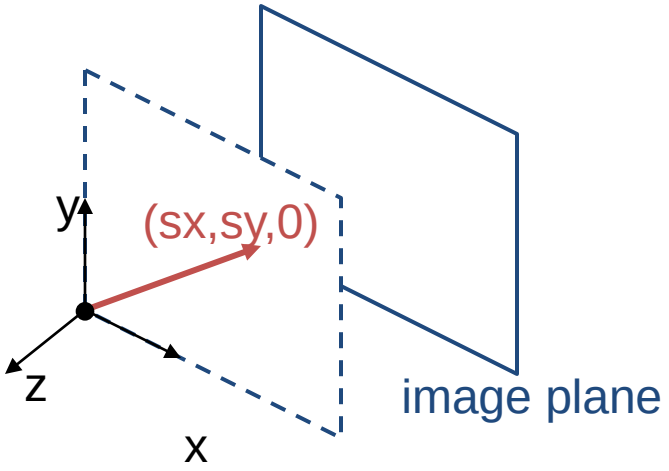
- A line passing all points at infinity:

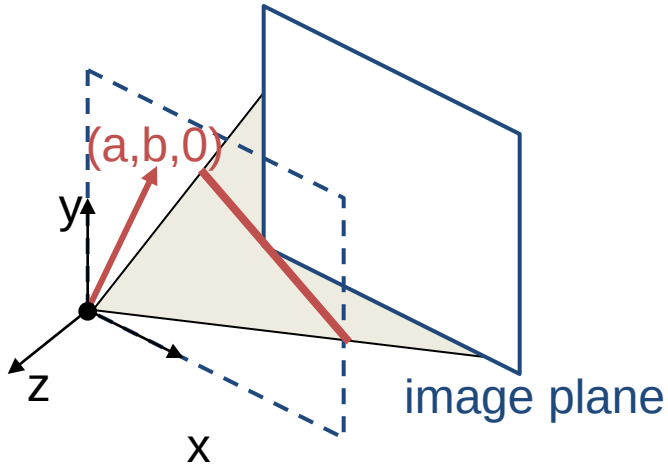
$$l_\infty = (0, 0, 1)^T$$

  - Because :

$$\begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ 0 \end{bmatrix} = 0$$

Ideal points and lines





- Ideal point (“point at infinity”)
  - $p \cong (x, y, 0)$ – parallel to image plane
  - It has infinite image coordinates

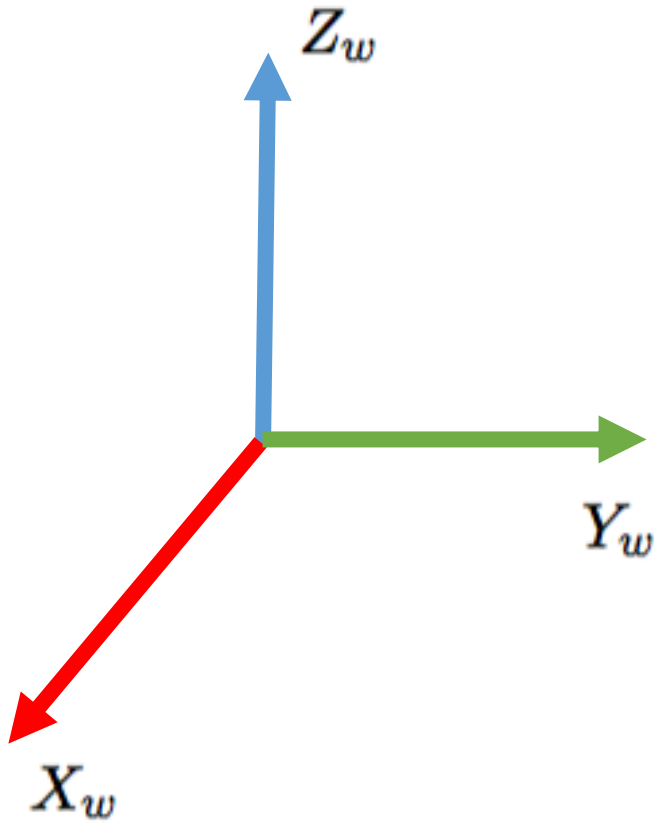
Ideal line

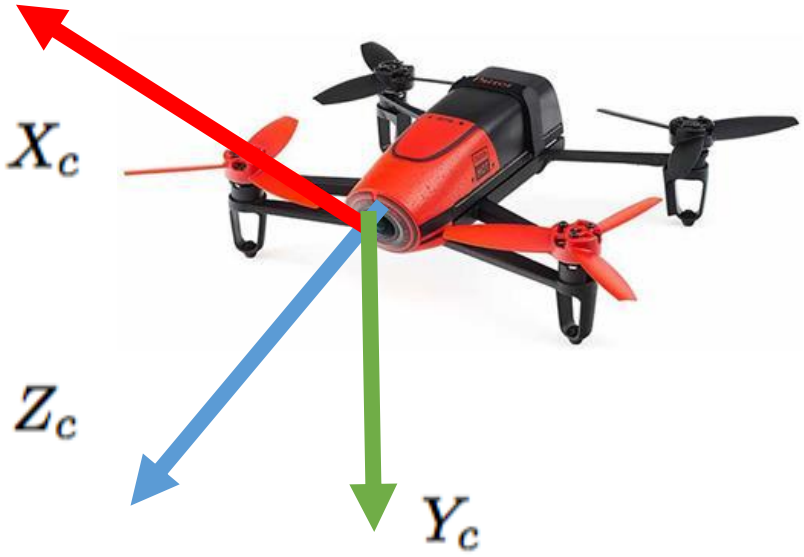
- $l \cong (a, b, 0)$ – parallel to image plane
- Corresponds to a line in the image (finite coordinates)

Perception: Rotations and Translations

Kostas Daniilidis

Transformation between camera and world coordinate systems



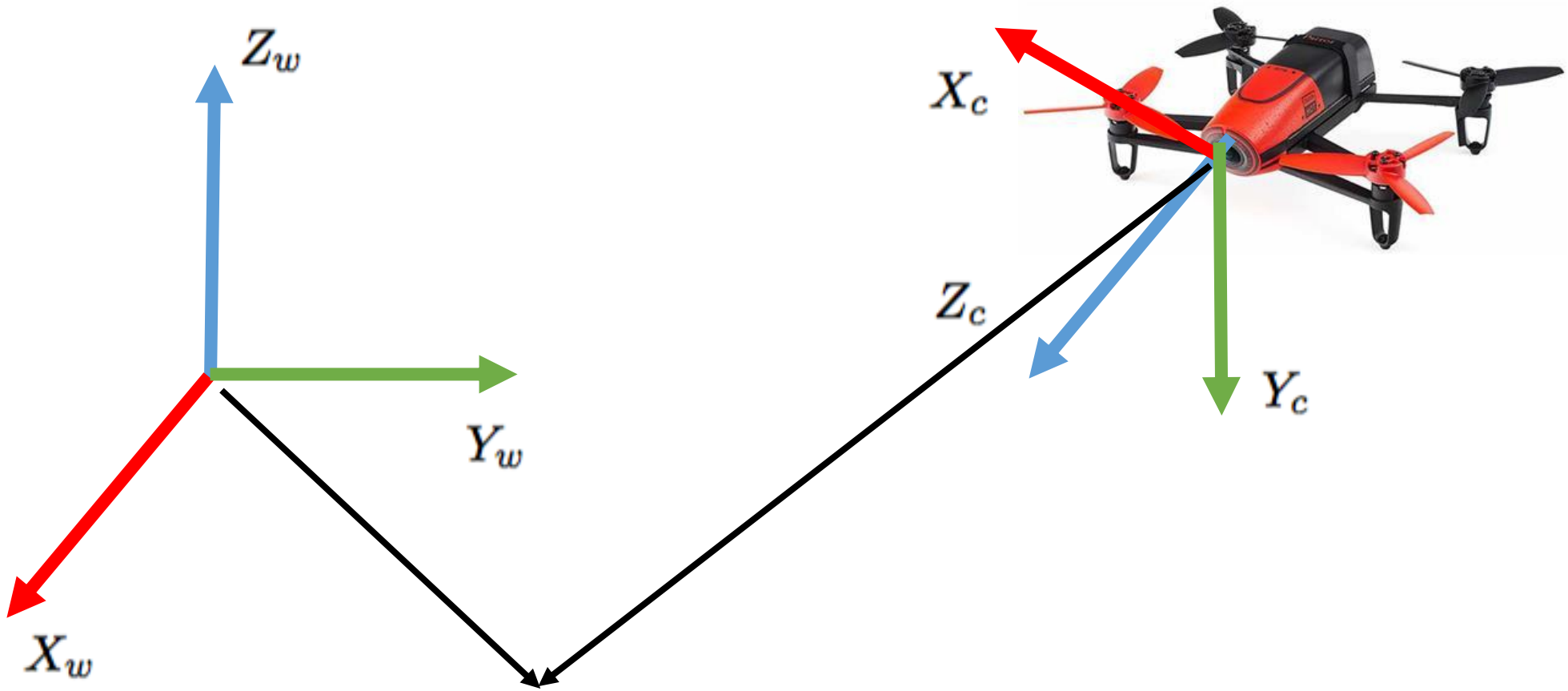


Red for X-Axis
Green for Y-Axis
Blue for Z-Axis

Remember
RGB is XYZ

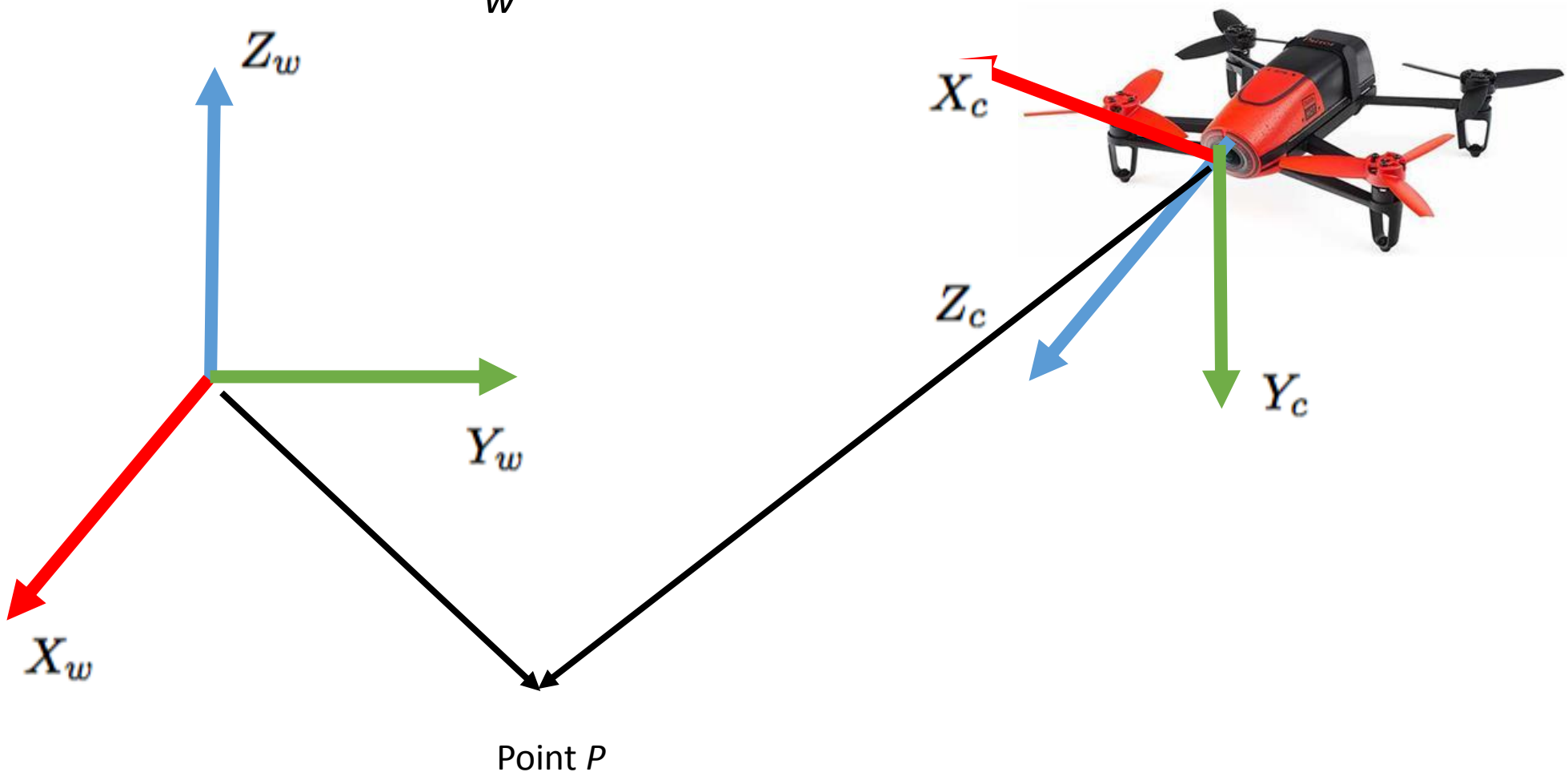
$$^{c}P = {}^{c}R_{w}\,{}^{w}P + {}^{c}T_{w}$$

Point *P* can be expressed with respect to “w” or “c” coordinate frames



$$^{c}P = {}^{c}R_{w}\,{}^{w}P + {}^{c}T_{w}$$

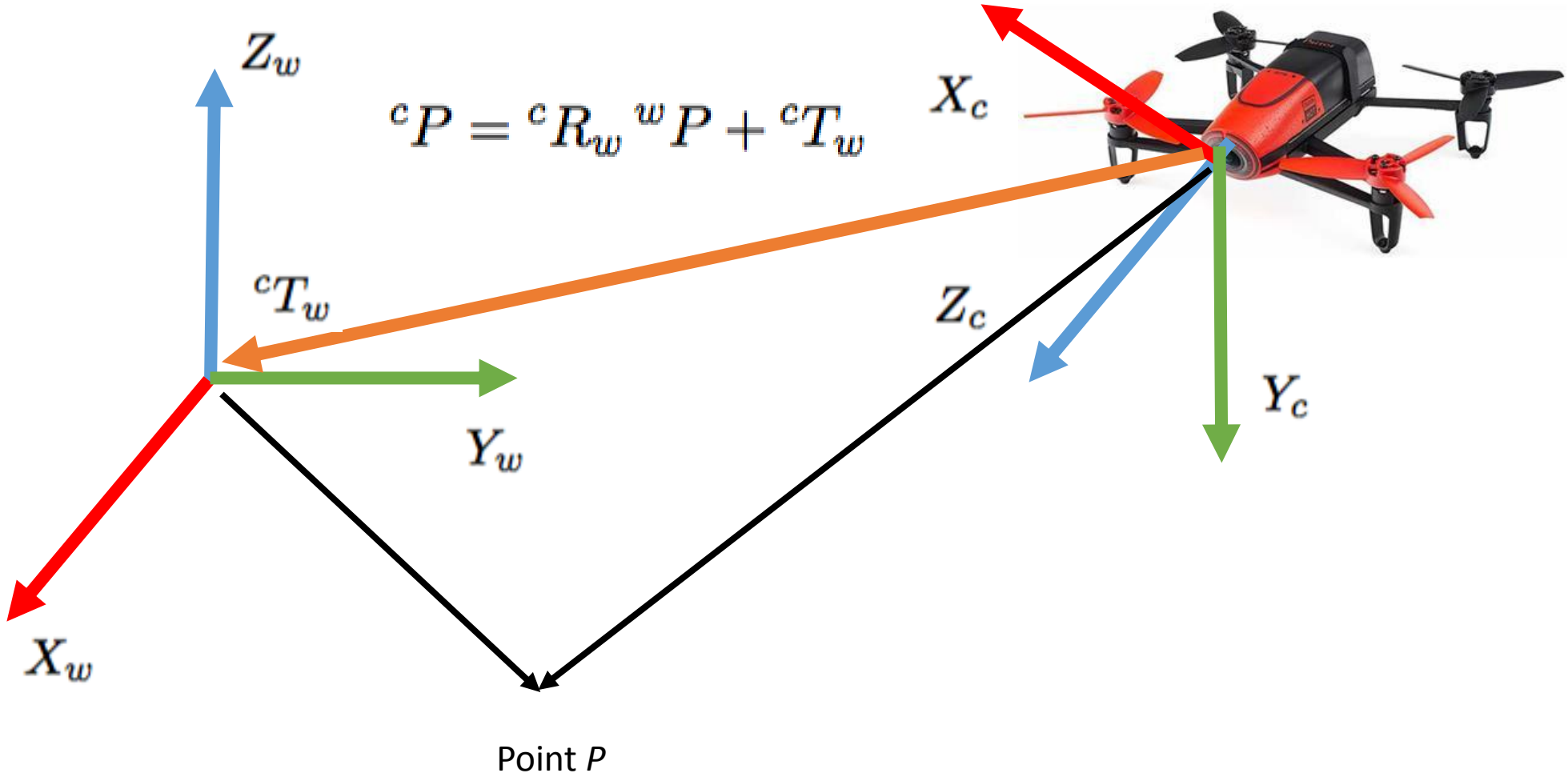
What is the geometric meaning of the rotation ${}^{c}R_{w}$ and the translation ${}^{c}T_{w}$?



What is the geometric meaning of the translation ${}^{c}T_{w}$?

This is easy to see if we set ${}^{w}P$ to zero.

Then, ${}^{c}P = {}^{c}R_{w}\,0 + {}^{c}T_{w}$ is the vector from camera origin to world origin:

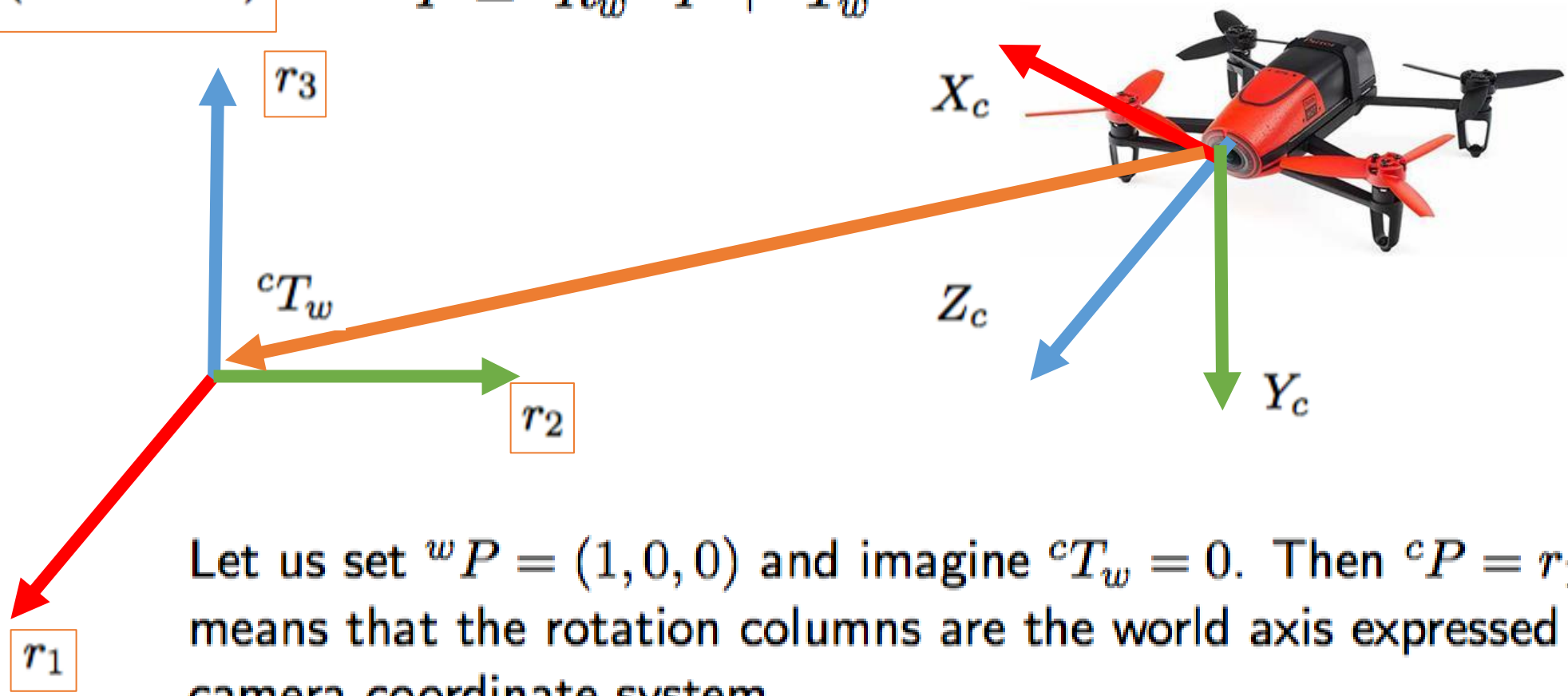


What is the geometric meaning of the rotation ${}^{c}R_{w}$?

Let the rotation matrix be written as 3 orthogonal column vectors:

$$ {}^{c}R_{w} = \begin{pmatrix} r_1 & r_2 & r_3 \end{pmatrix} $$

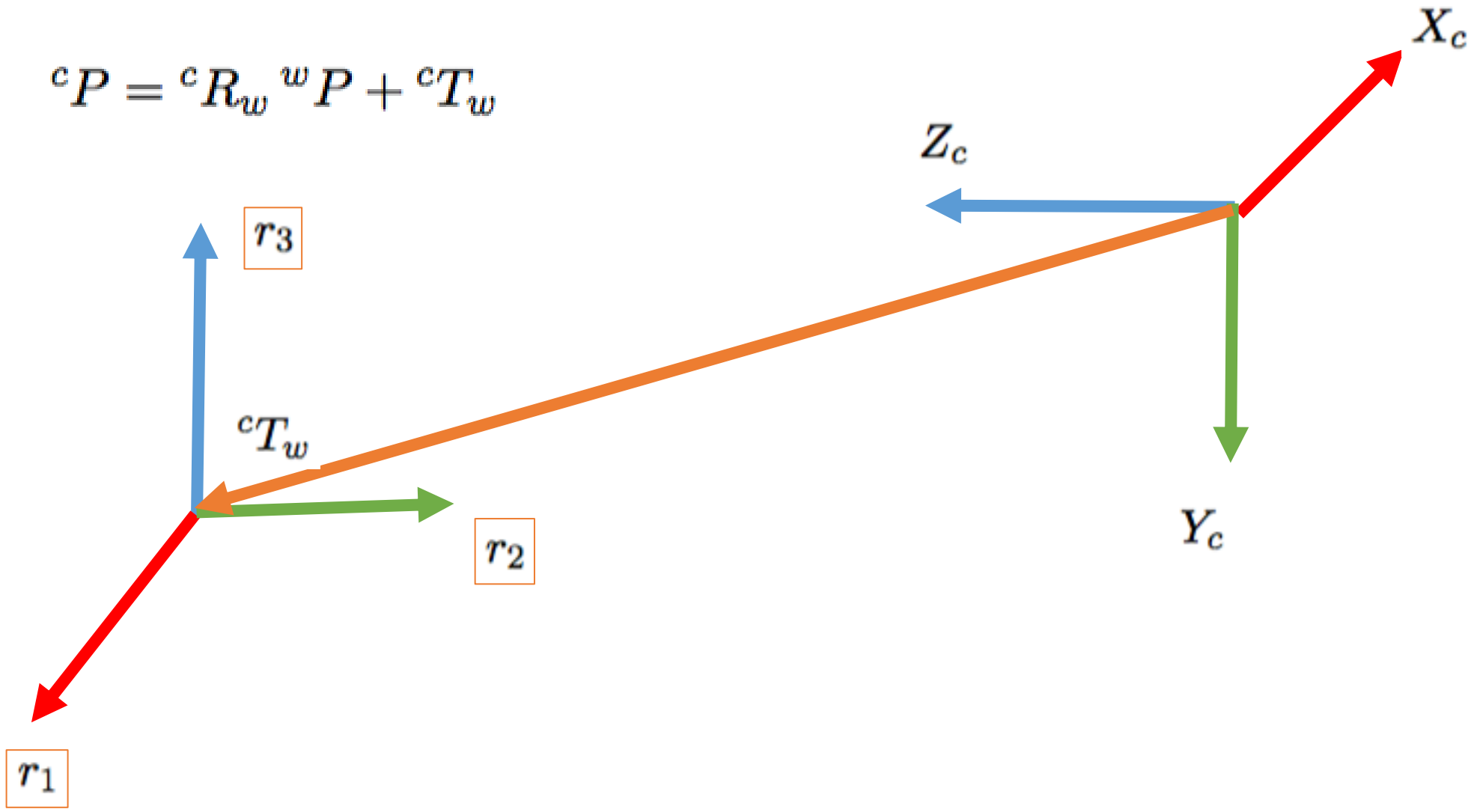
$$ {}^{c}P = {}^{c}R_{w}\,{}^{w}P + {}^{c}T_{w} $$



Let us set ${}^{w}P = (1, 0, 0)$ and imagine ${}^{c}T_{w} = 0$. Then ${}^{c}P = r_1$ which means that the rotation columns are the world axis expressed in the camera coordinate system.

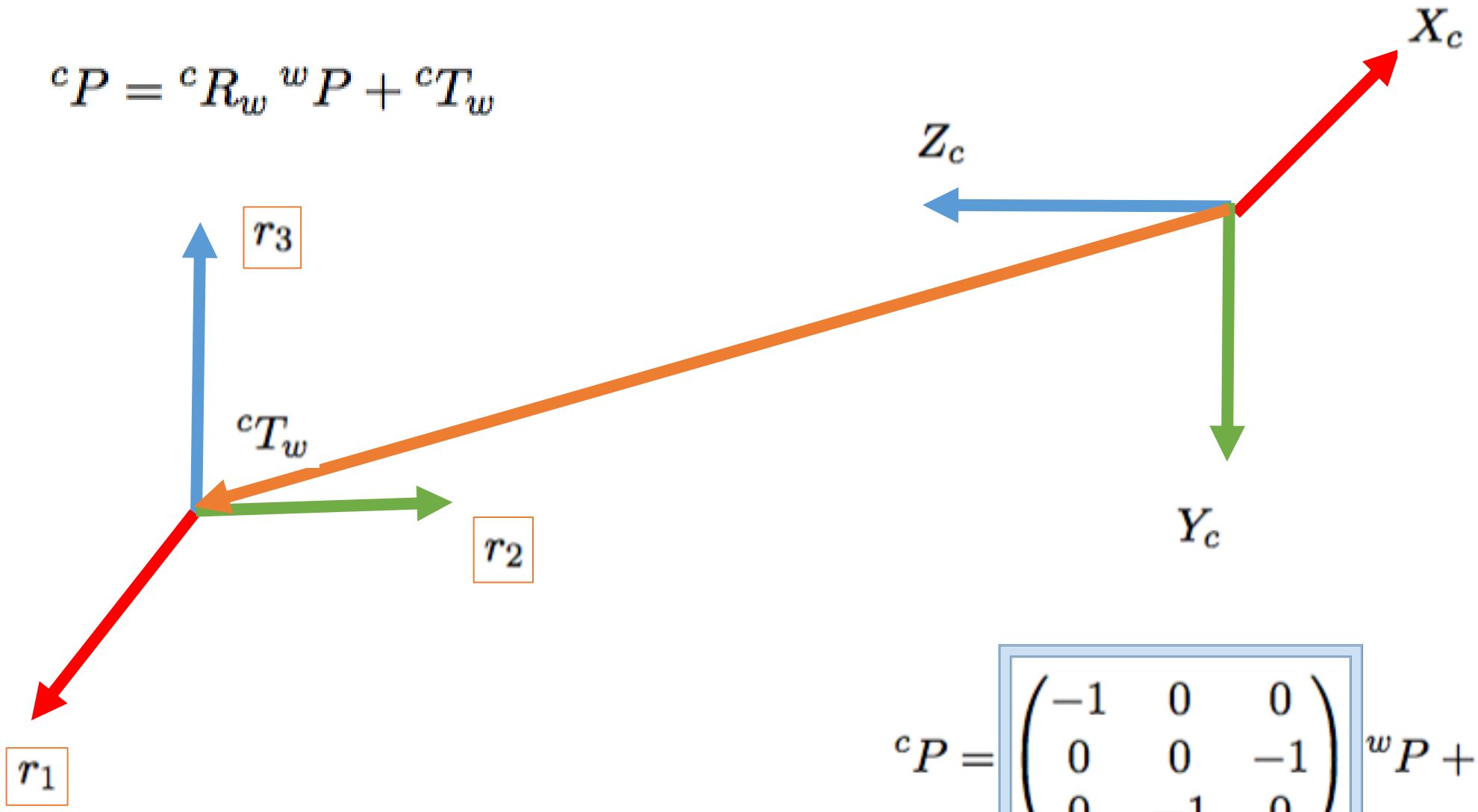
Let us look at the simple example:

$$^{c}P = {}^{c}R_{w}\,{}^{w}P + {}^{c}T_{w}$$



How does the rotation matrix read?

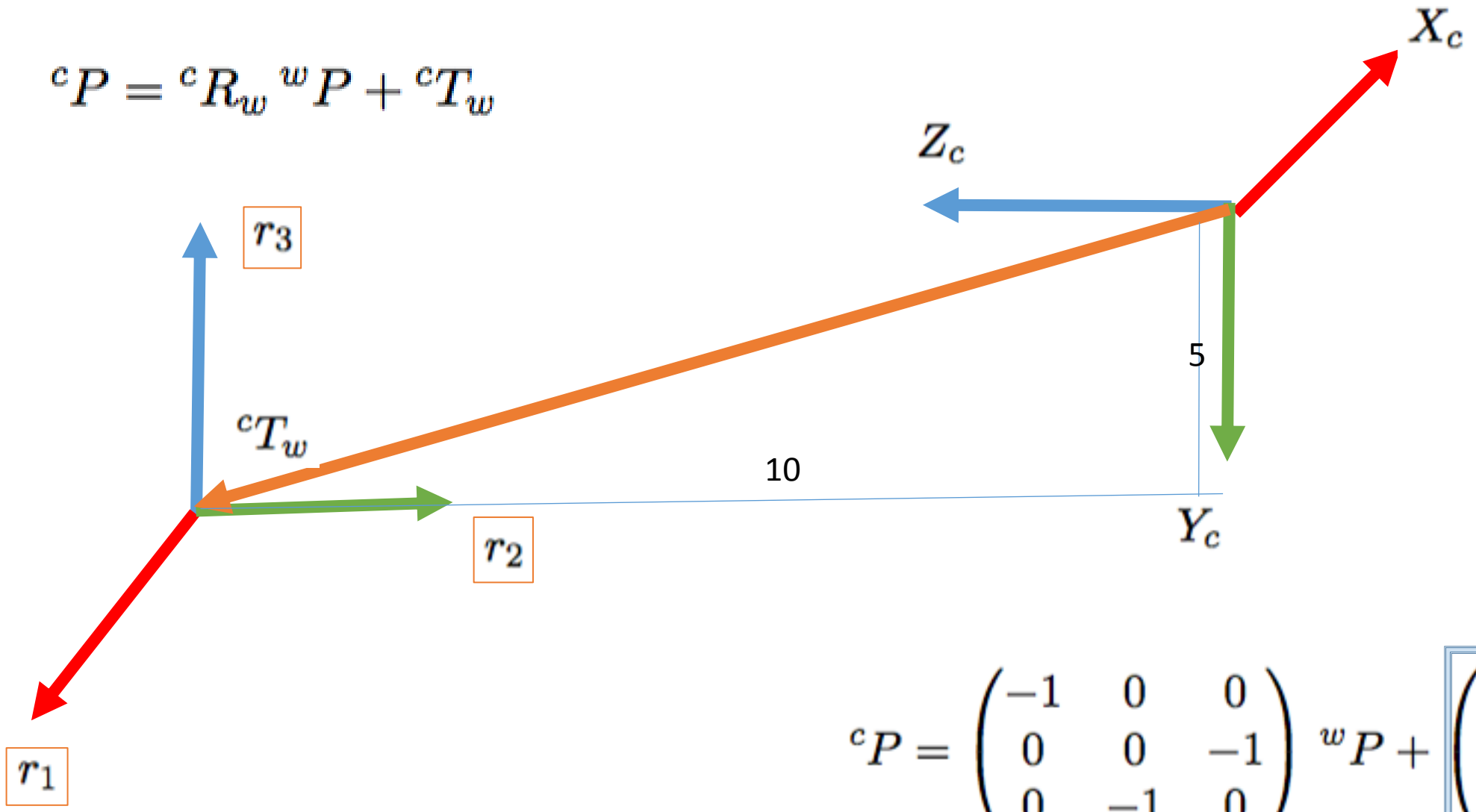
$$^{c}P = {}^{c}R_{w}\,{}^{w}P + {}^{c}T_{w}$$



$$^{c}P = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{pmatrix} {}^{w}P + \begin{pmatrix} ? \\ ? \\ ? \end{pmatrix}$$

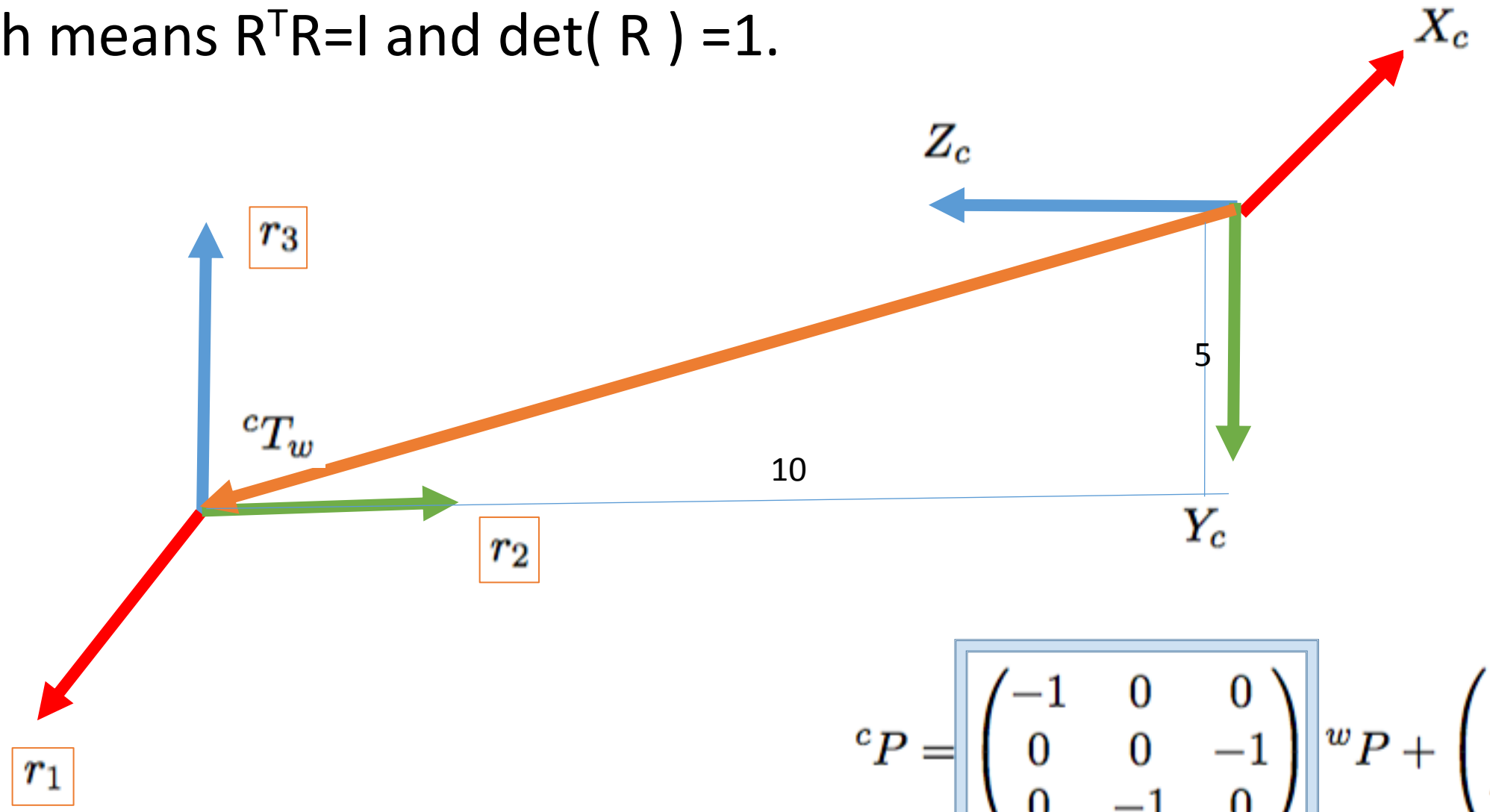
What about the translation:

$$^{c}P = {}^{c}R_{w}\,{}^{w}P + {}^{c}T_{w}$$



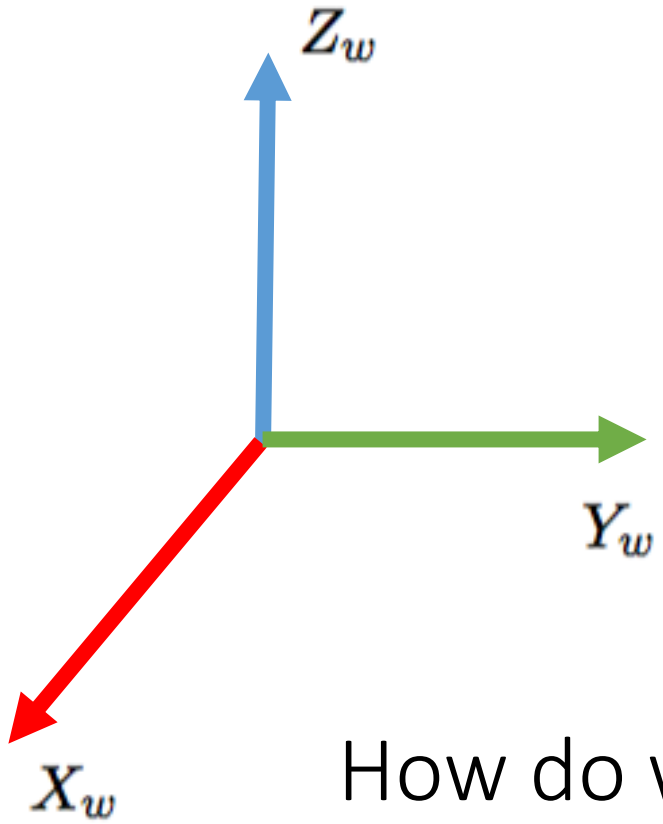
$$^{c}P = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{pmatrix} {}^{w}P + \boxed{\begin{pmatrix} 0 \\ 5 \\ 10 \end{pmatrix}}$$

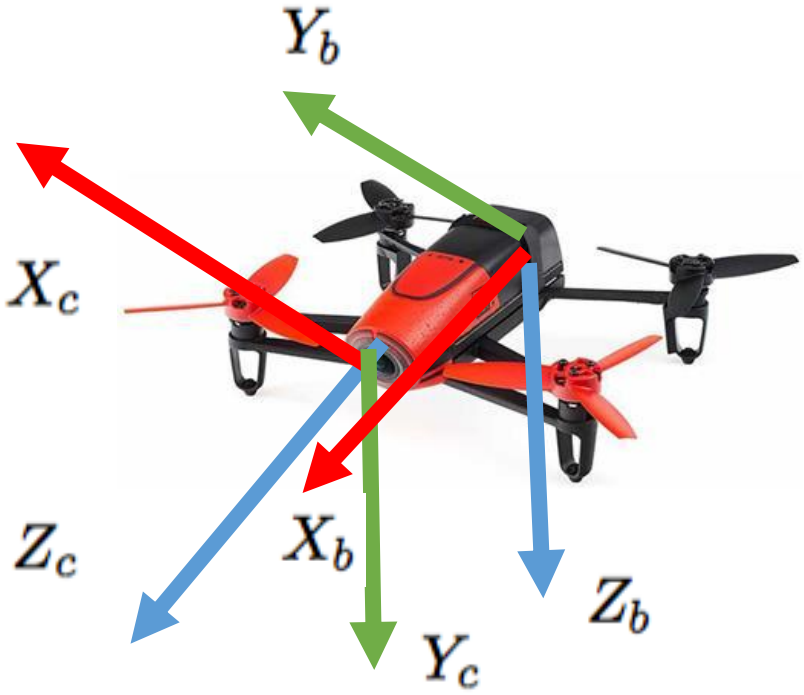
We have to make sure that the 3x3 matrix is a rotation matrix,
Which means $R^TR=I$ and $\det(R)=1$.



$$ {}^cP = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{pmatrix} {}^wP + \begin{pmatrix} 0 \\ 5 \\ 10 \end{pmatrix} $$

Now imagine one more coordinate frame: a body frame with axes corresponding to roll (X_b), pitch (Y_b), yaw (Z_b) angles.

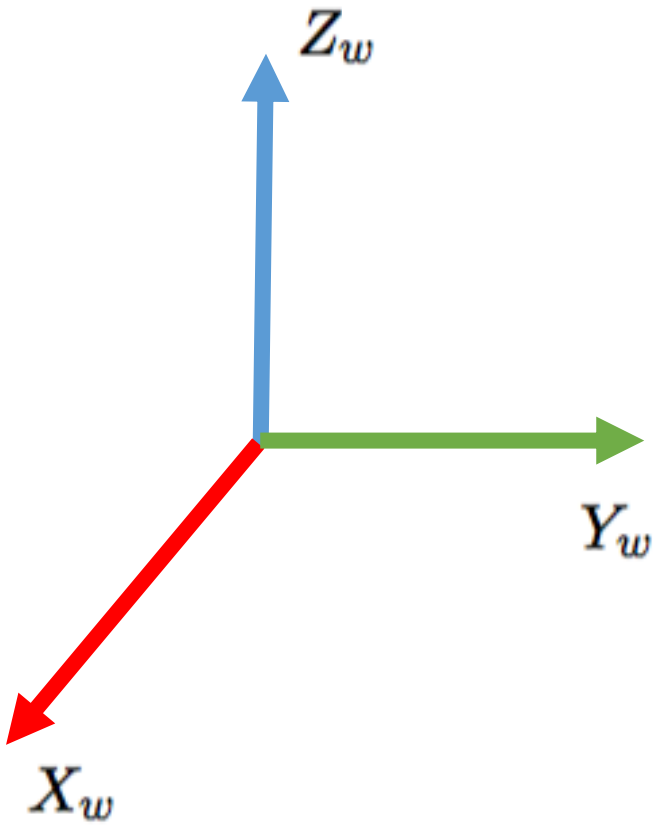


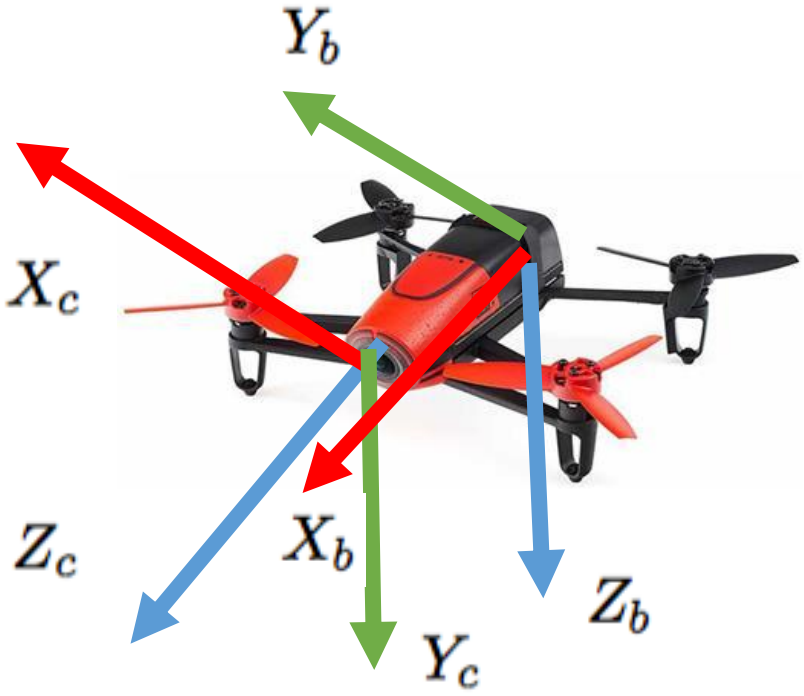


How do we compose transformations?

The easiest way to transform between coordinate systems is to use 4x4 matrices:

$$ {}^{c}M_{w} = \begin{pmatrix} & {}^{c}R_{w} & & {}^{c}T_{w} \\ 0 & 0 & 0 & 1 \end{pmatrix} $$



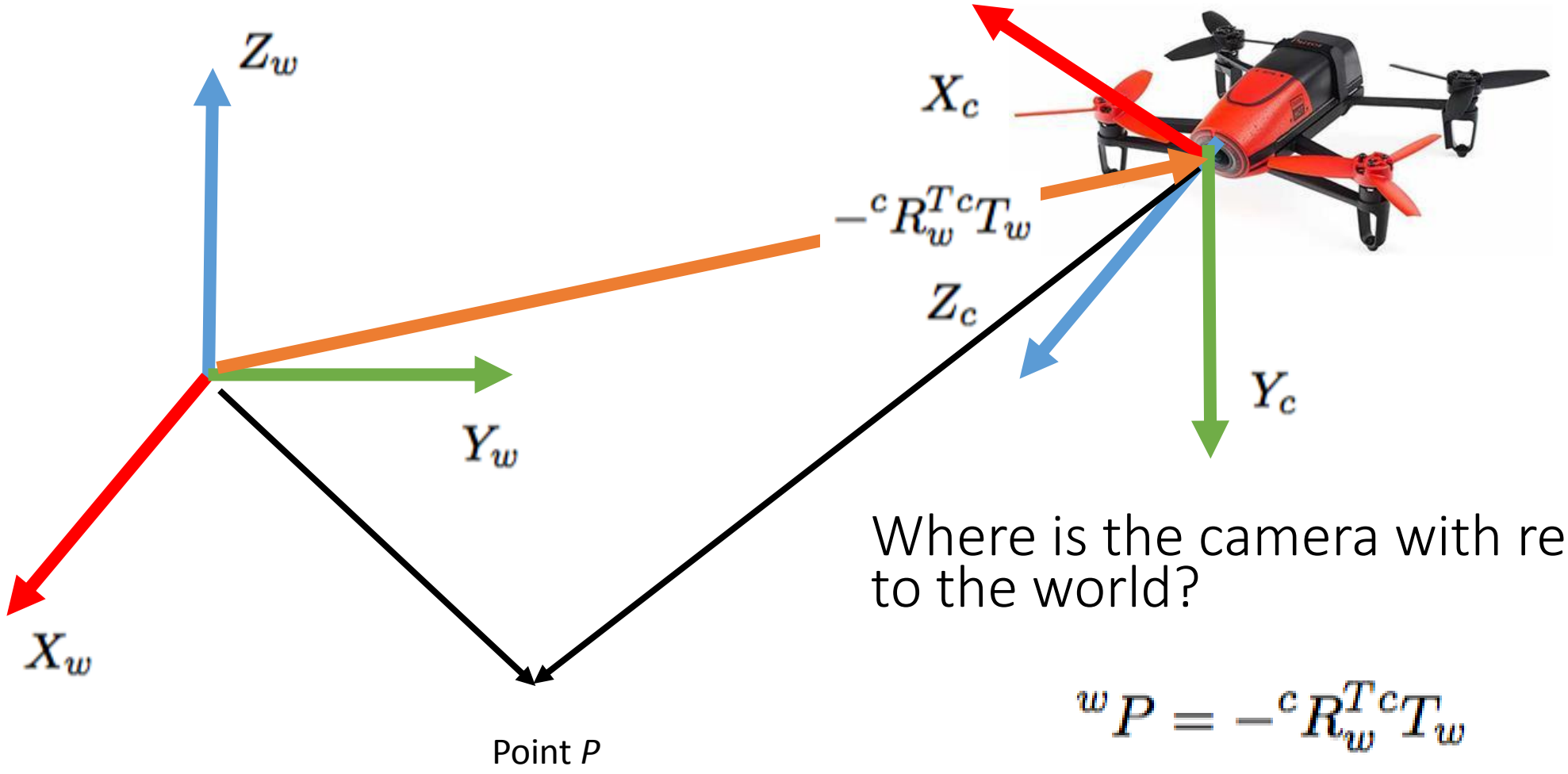


Then we just concatenate the 4x4 matrices

$$ {}^{w}M_{b} = {}^{w}M_{c}\,{}^{c}M_{b} $$

What about the inverse transformation?

$$ {}^{w}M_{c} = \begin{pmatrix} & {}^{c}R_{w}^{T} & & -{}^{c}R_{w}^{T}{}^{c}T_{w} \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

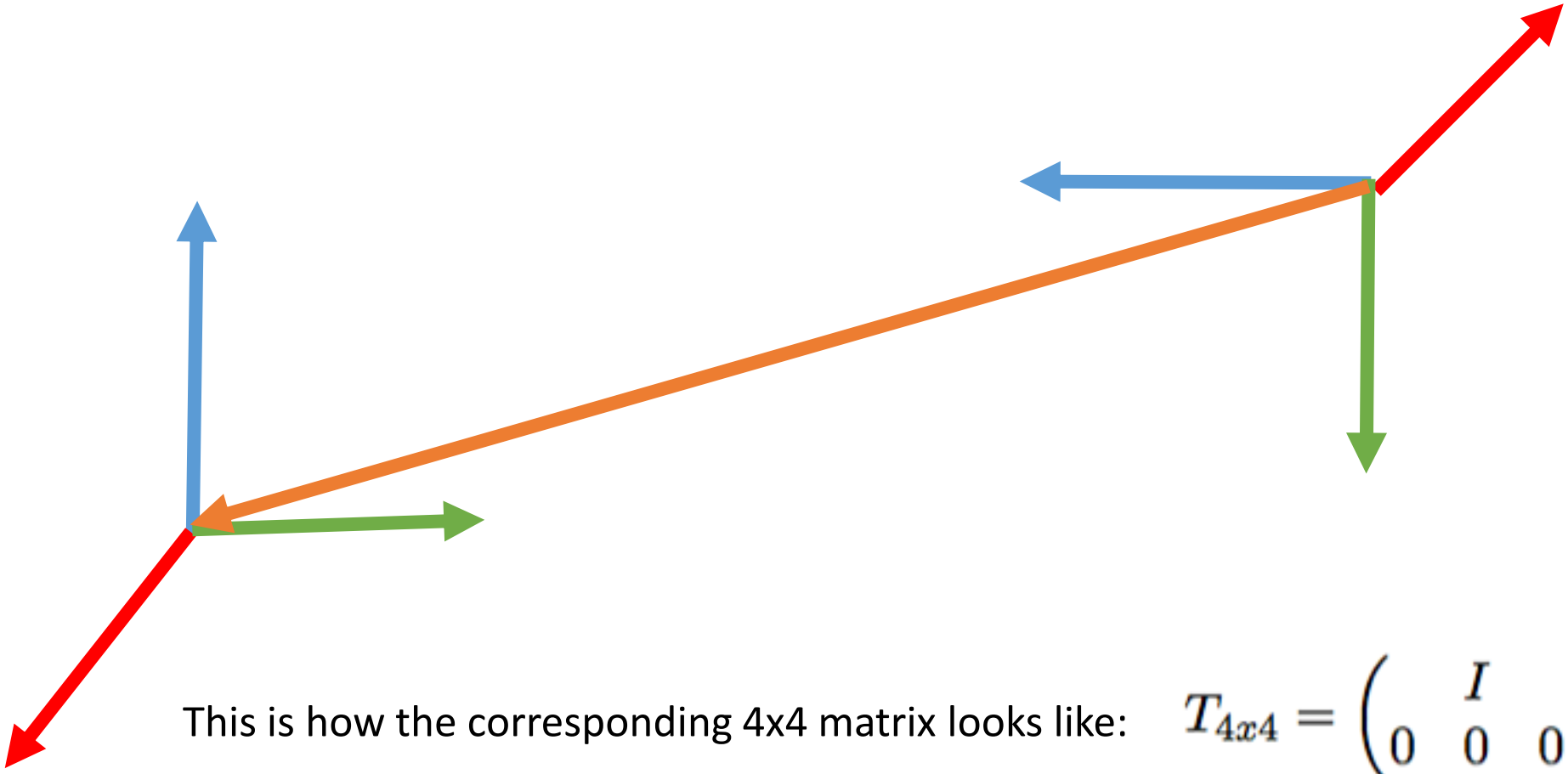


Where is the camera with respect to the world?

$$ {}^{w}P = -{}^{c}R_{w}^{T}{}^{c}T_{w} $$

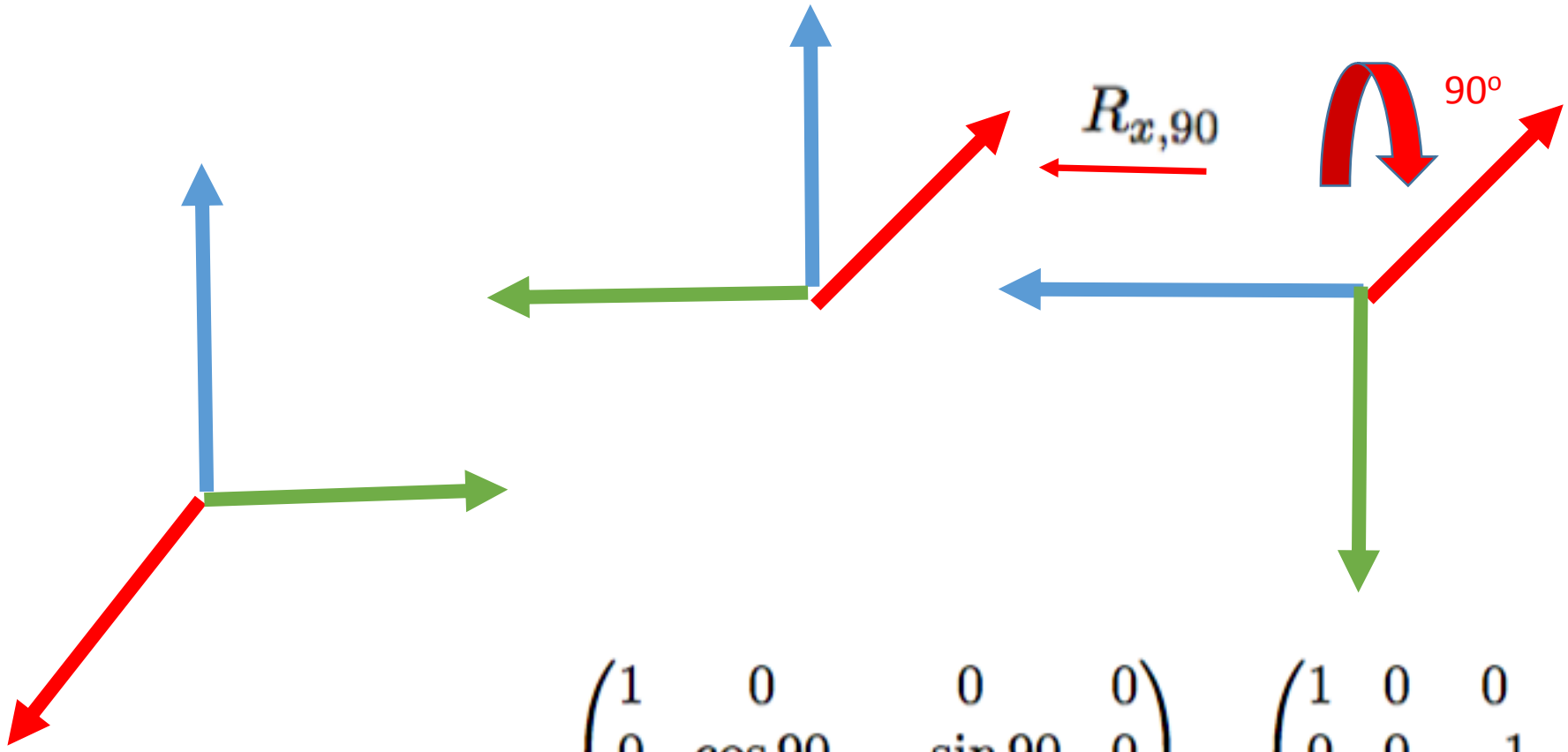
Alternative interpretation as a sequence of motions:

1. The camera frame first translates to the world



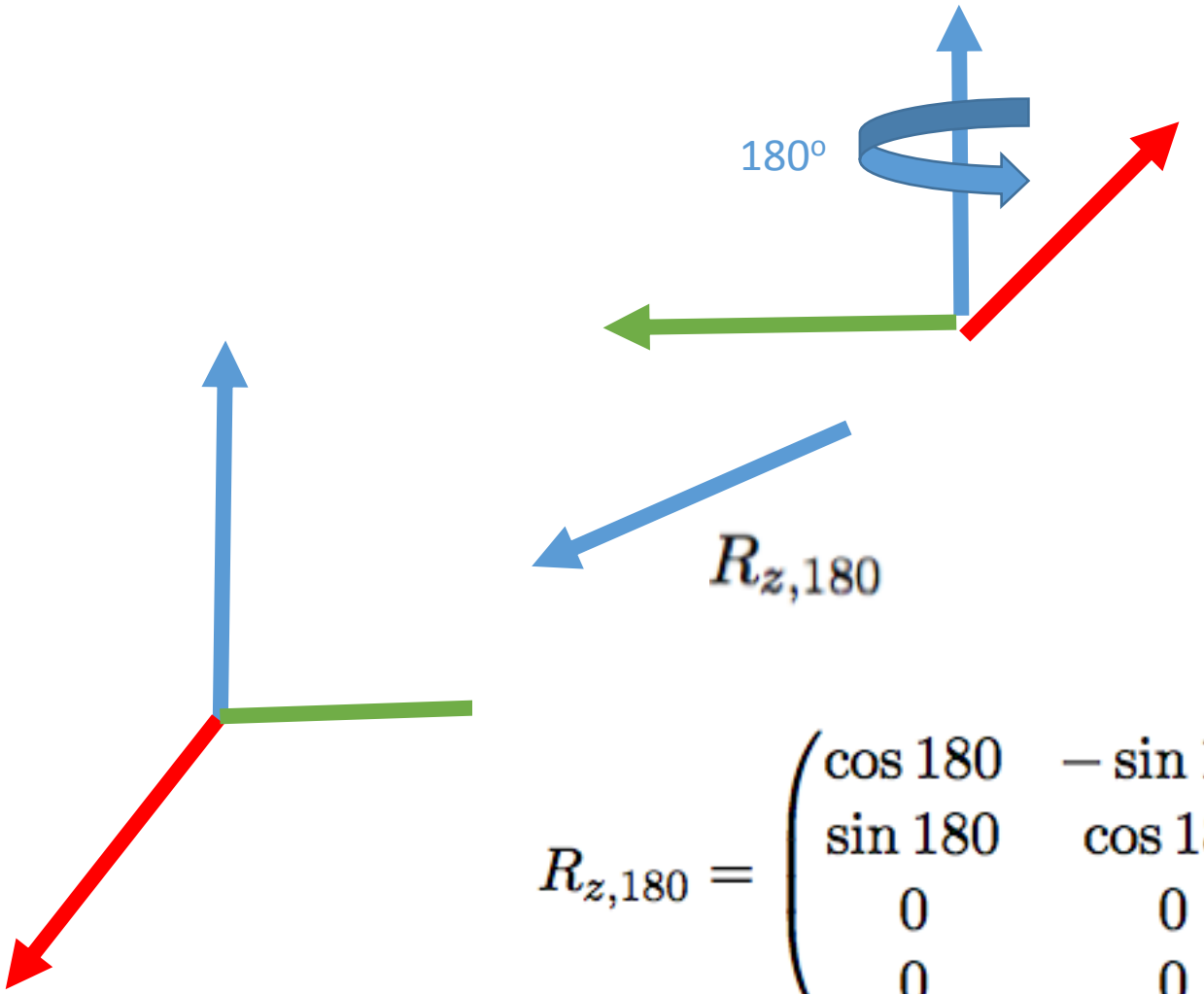
This is how the corresponding 4x4 matrix looks like: $T_{4x4} = \begin{pmatrix} & I & & {}^{c}T_{w} \\ 0 & 0 & 0 & 1 \end{pmatrix}$

2. The camera frame rotates 90 degrees around x



$$R_{x,90} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos 90 & -\sin 90 & 0 \\ 0 & \sin 90 & \cos 90 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

A3. The camera frame rotates 180 degrees around z



$$R_{z,180} = \begin{pmatrix} \cos 180 & -\sin 180 & 0 & 0 \\ \sin 180 & \cos 180 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

How do we compose these motions? Golden rule: when we move coordinate frames and we refer to the most recent coordinate frame
we always postmultiply!

$$
\begin{aligned}
{}^{c}M_{w} &= TR_{x,90}R_{z,180} \\
&= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 5 \\ 0 & 0 & 1 & 10 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 5 \\ 0 & -1 & 0 & 10 \\ 0 & 0 & 0 & 1 \end{pmatrix}
\end{aligned}
$$

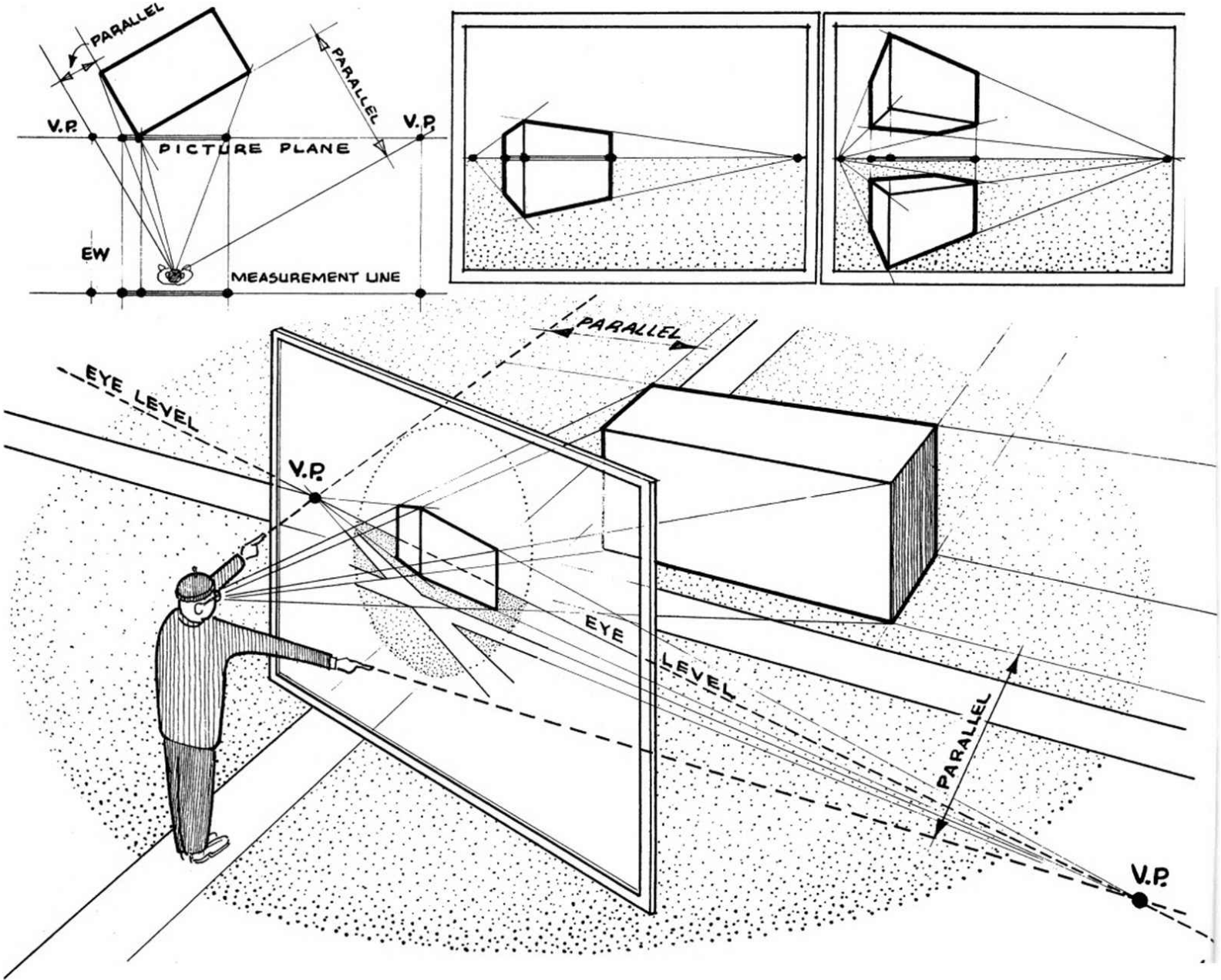
Lens configuration (internal parameter)

$$\begin{bmatrix} x \\ 1 \end{bmatrix} = L\left(K \begin{bmatrix} R & t \end{bmatrix} \begin{bmatrix} X \\ 1 \end{bmatrix} \right)$$

Spatial relationship between sensor and pinhole (internal parameter)

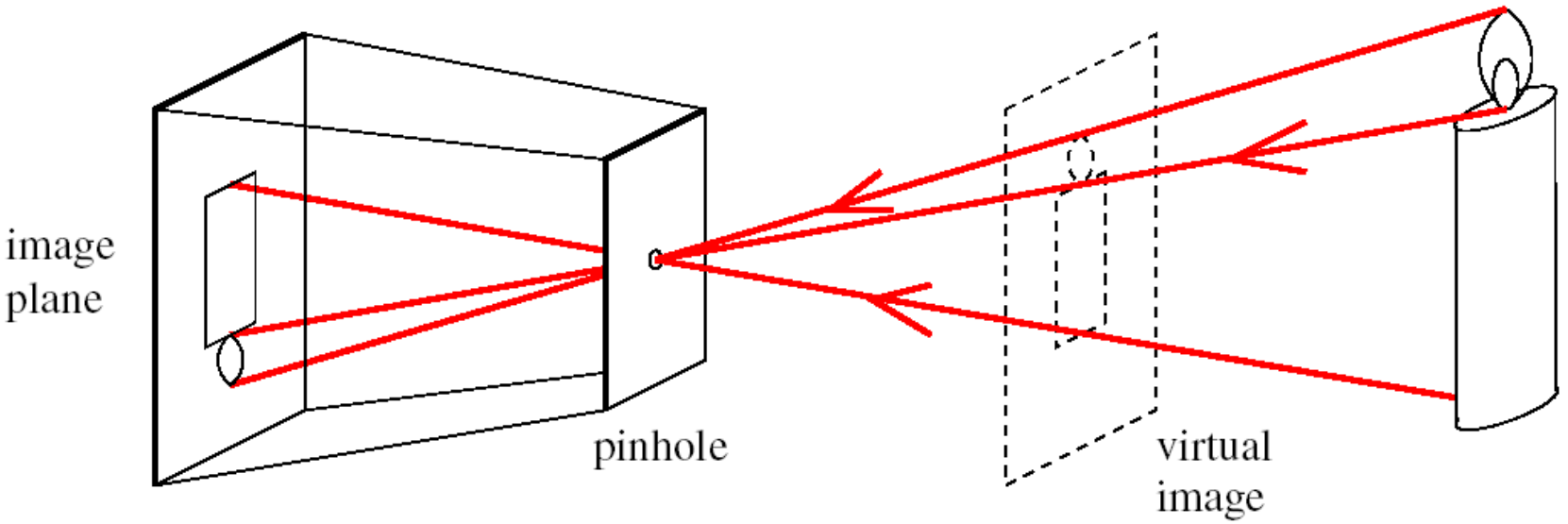
Camera body configuration (extrinsic parameter)

http://www.joshuanava.biz/perspective/in-other-words-the-observer-simply-points-in-the-same-direction-as-the-lines-in-order-to-find-their-vanishing-point.html

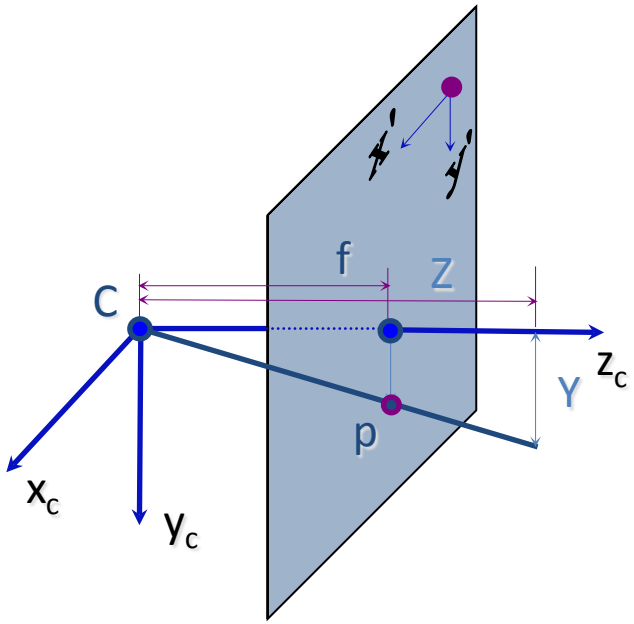


The Pinhole Camera

– Light enters a darkened chamber through a pinhole opening and forms an image on the further surface



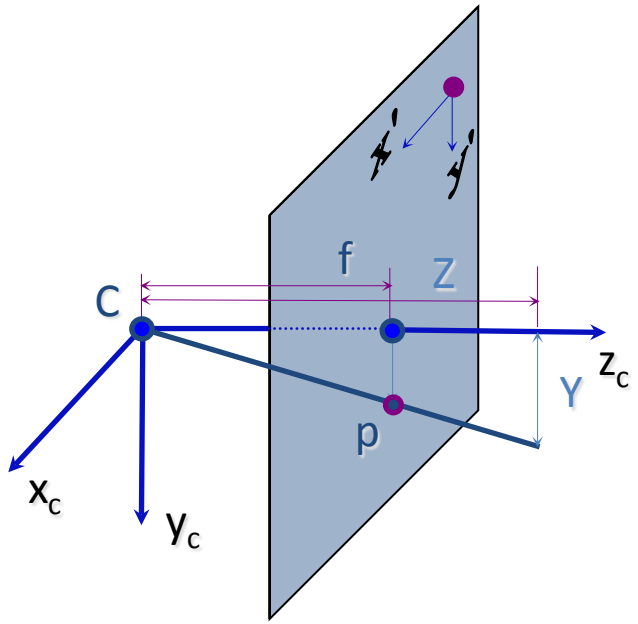
1st Person Camera world



Projection equation:

3D to 2D image: $$x' = f\frac{X}{Z} \qquad y' = f\frac{Y}{Z}$$

1st Person Camera world



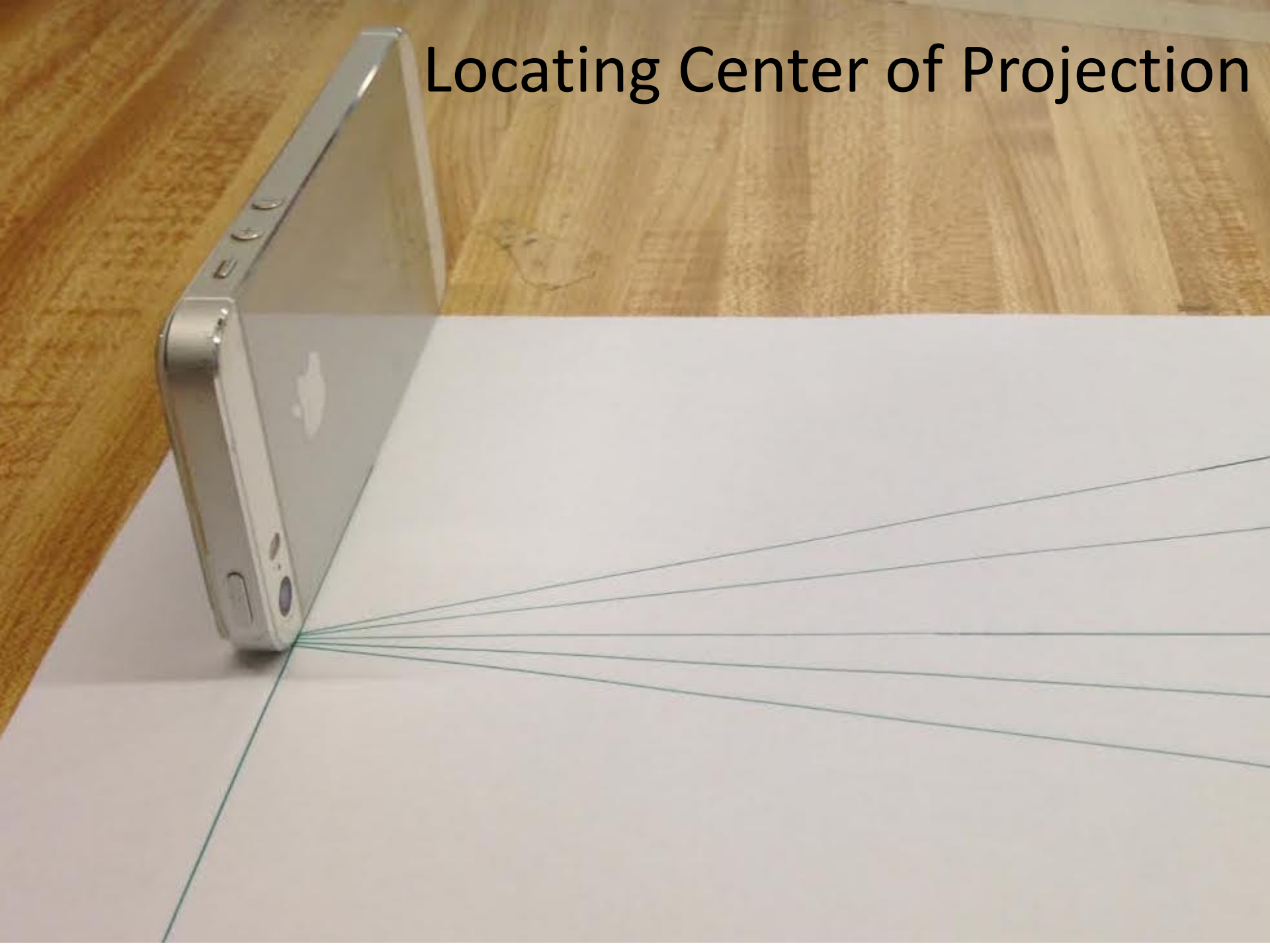
Projection equation:

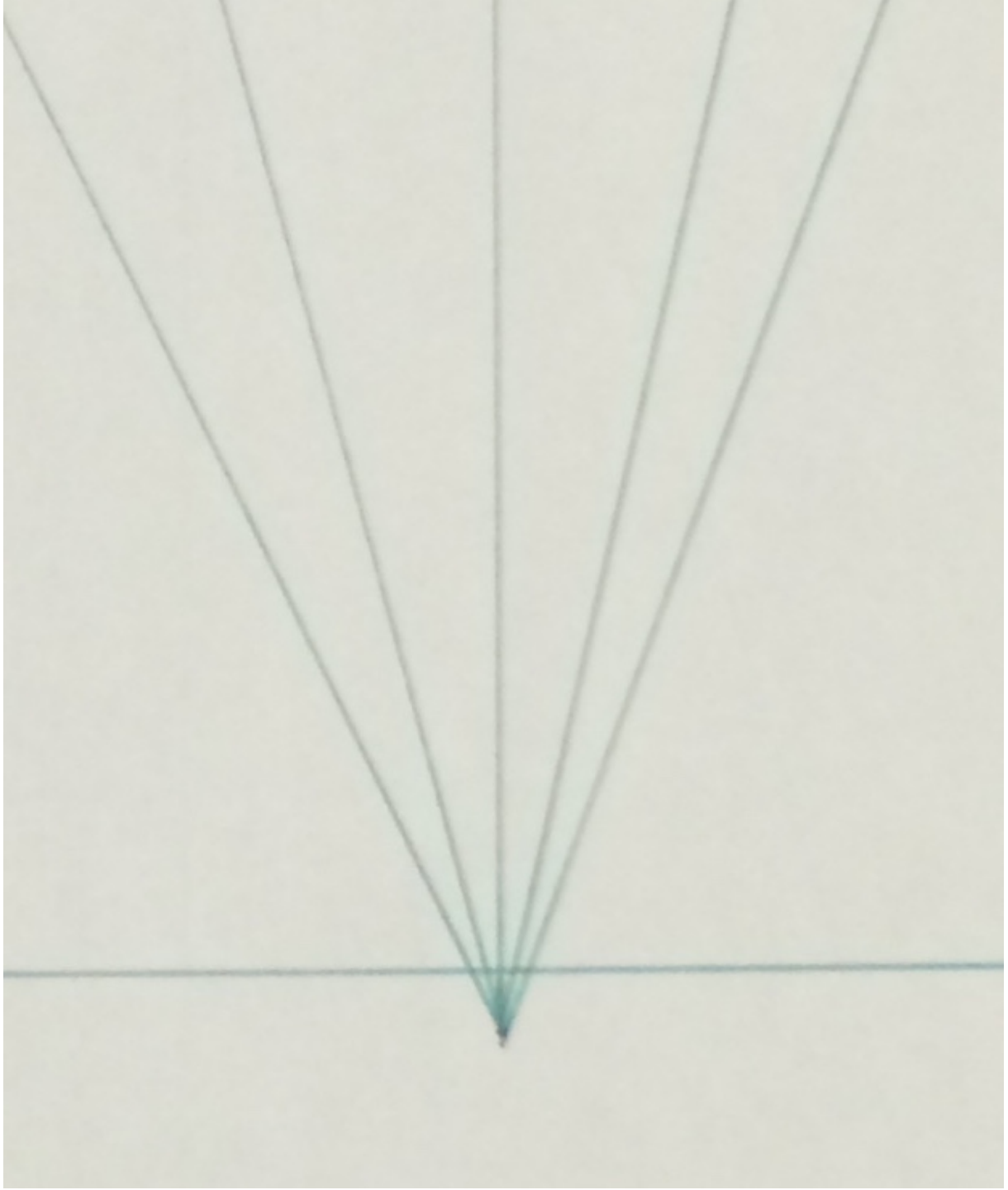
3D to 2D image: $$x' = f\frac{X}{Z} \qquad y' = f\frac{Y}{Z}$$

Where is the Center of Projection?

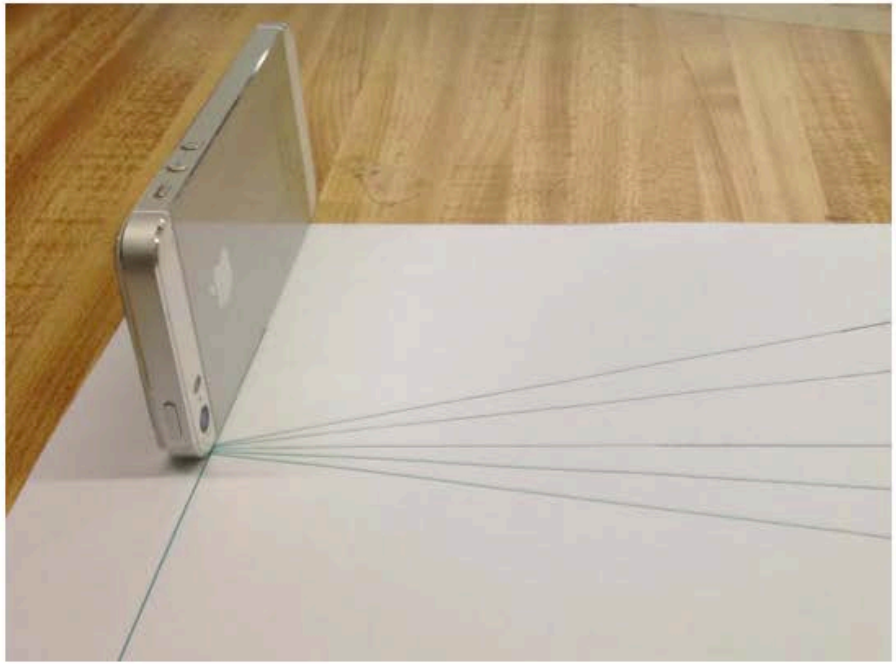
What is the Focal length?

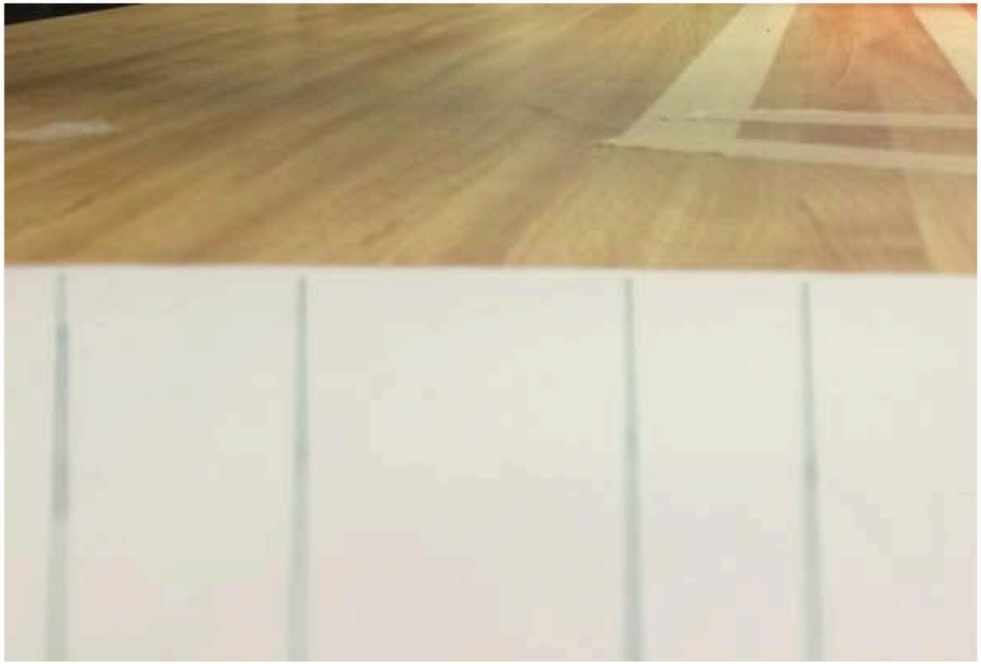
Locating Center of Projection



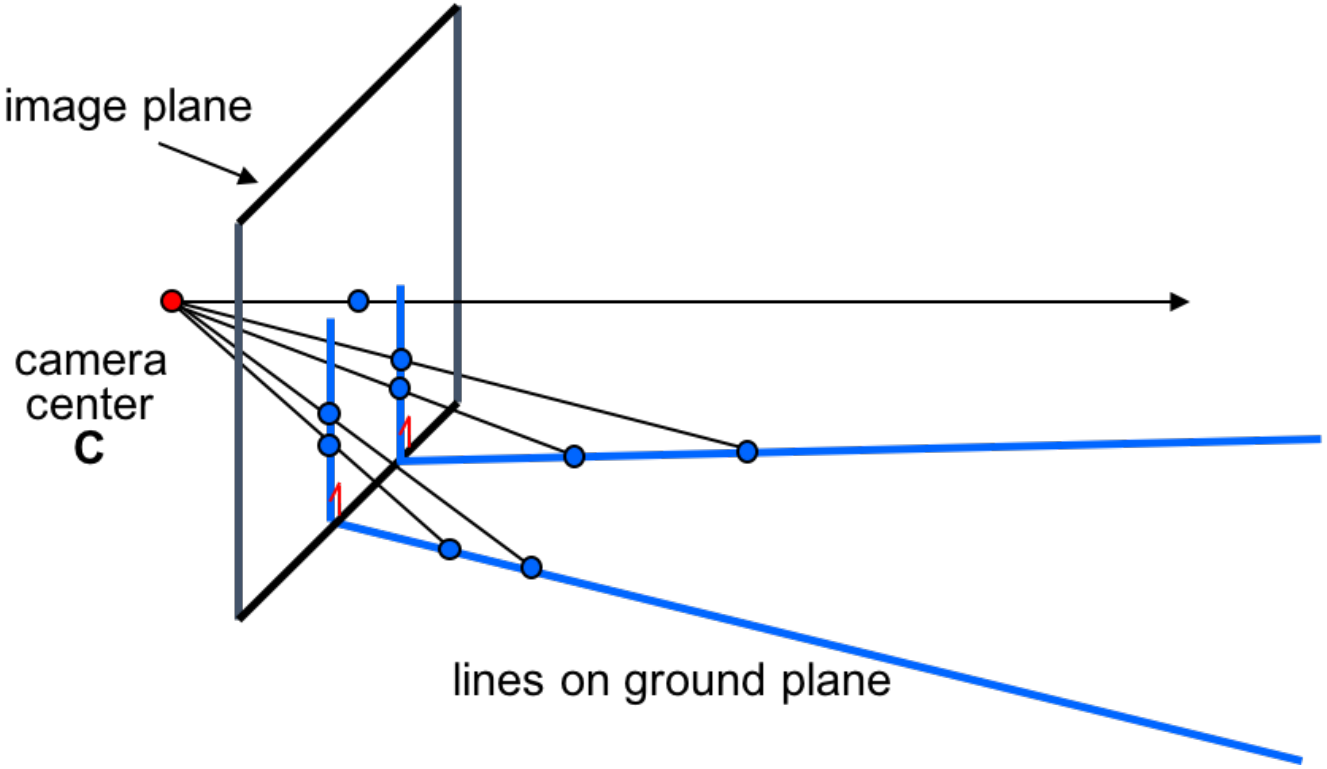


Locating Center of Projection

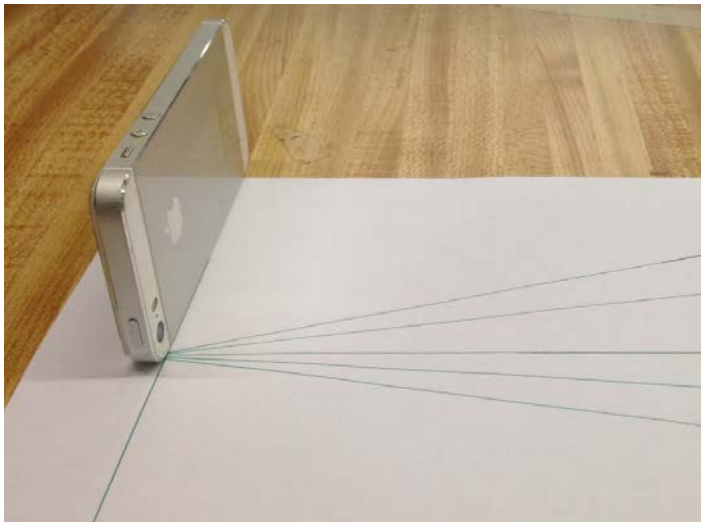
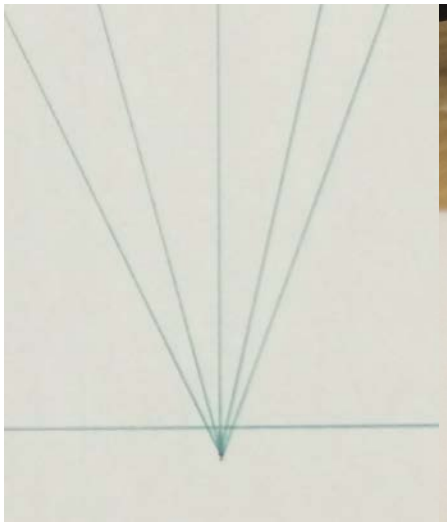
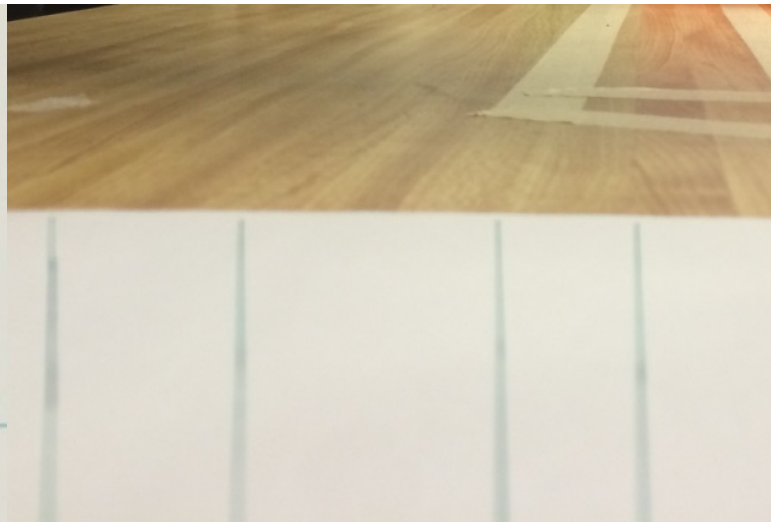




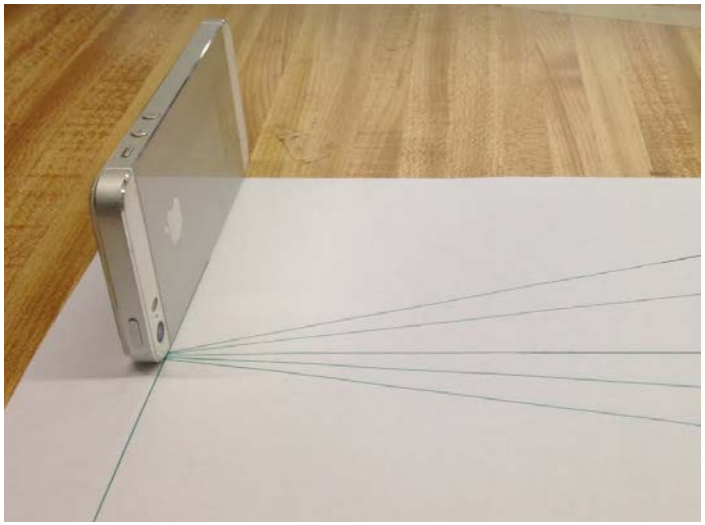
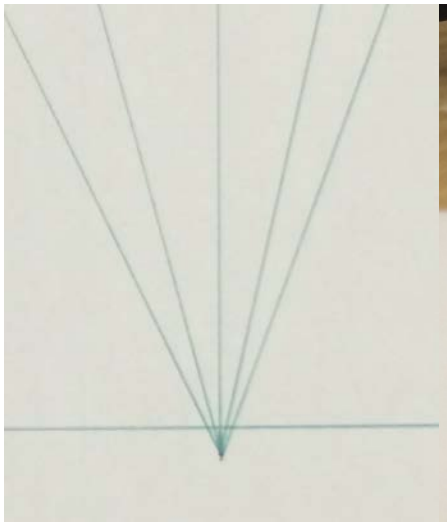
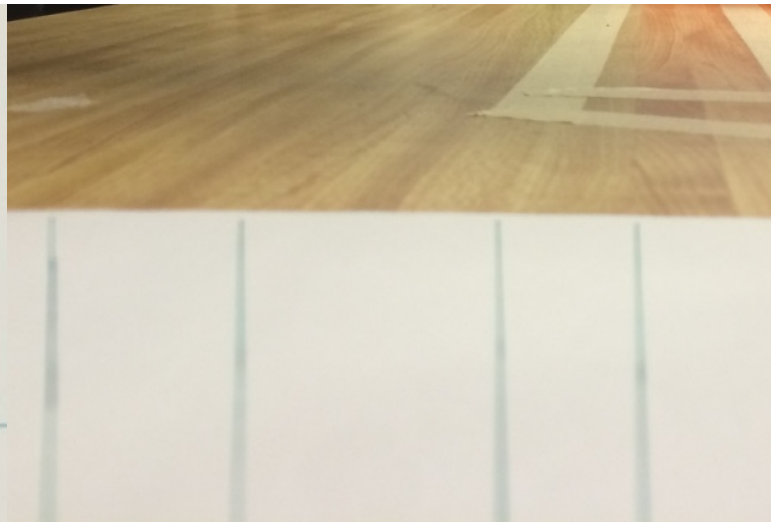
Locating Center of Projection



Locating Center of Projection

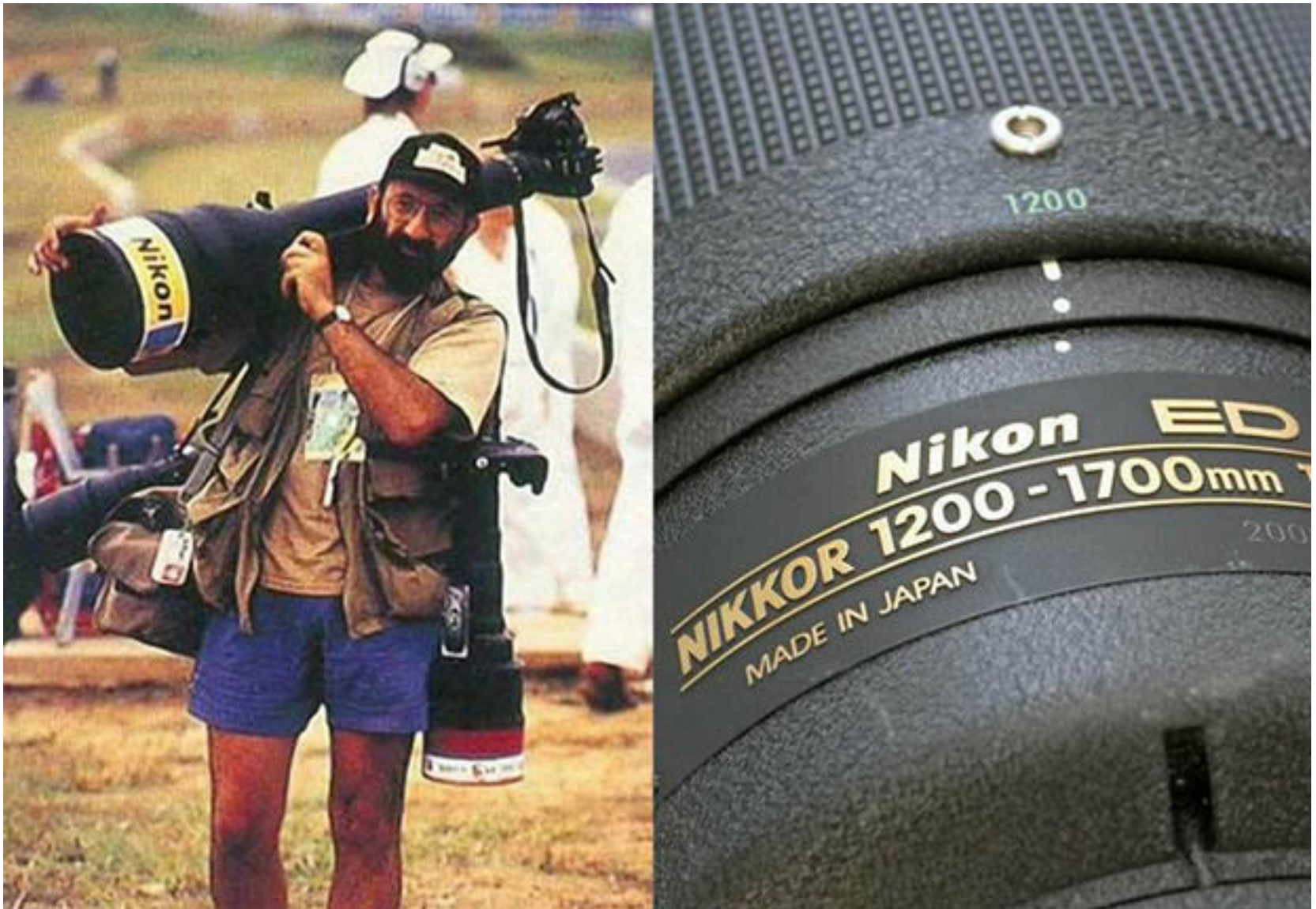




Locating Center of Projection

Focal Length

Nikon’s 1200-1700mm f/5.6-8P lens.



Nikon’s 1200-1700mm f/5.6-8P lens.

Large Focal Length compresses depth

400 mm

200 mm

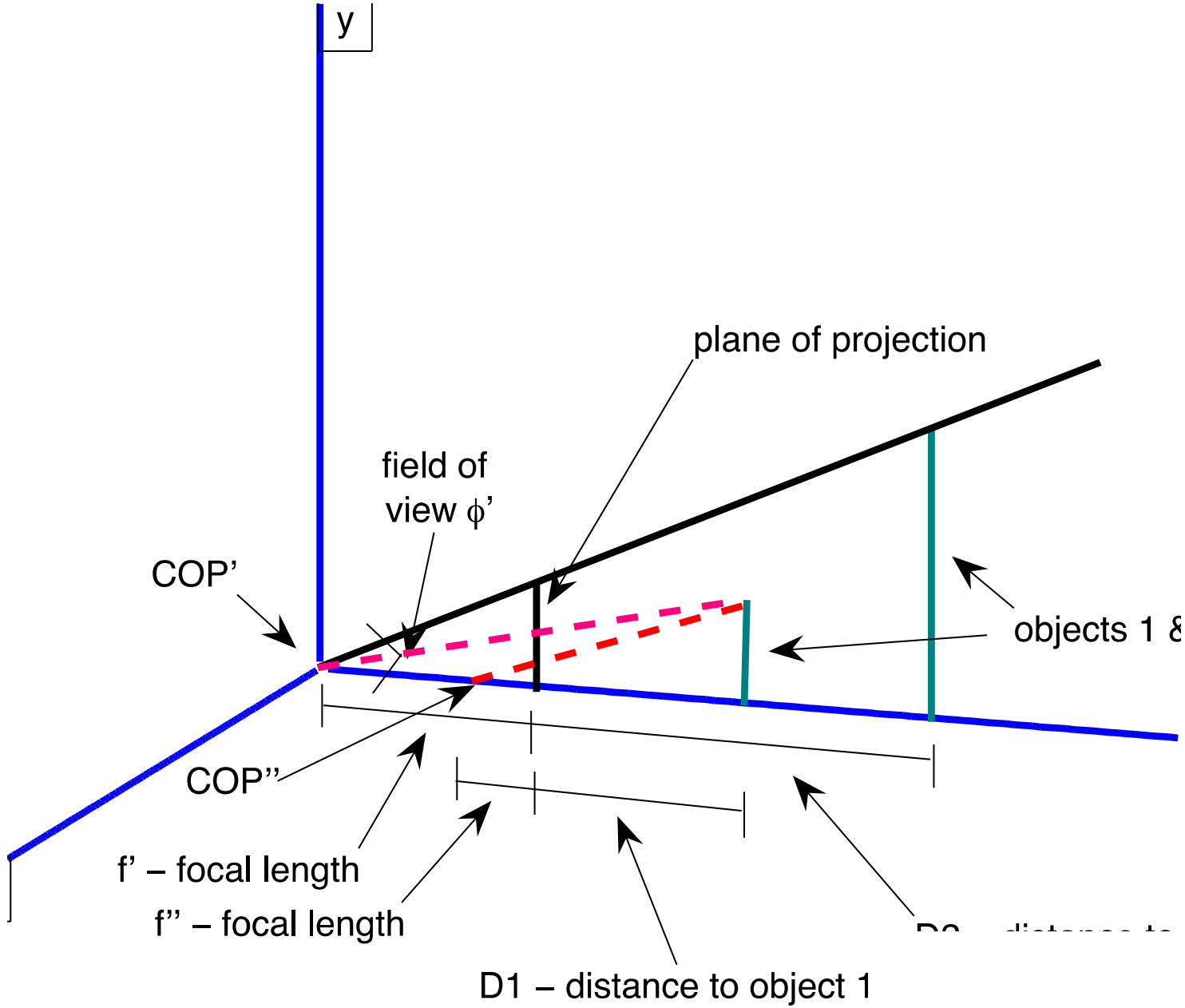
100 mm

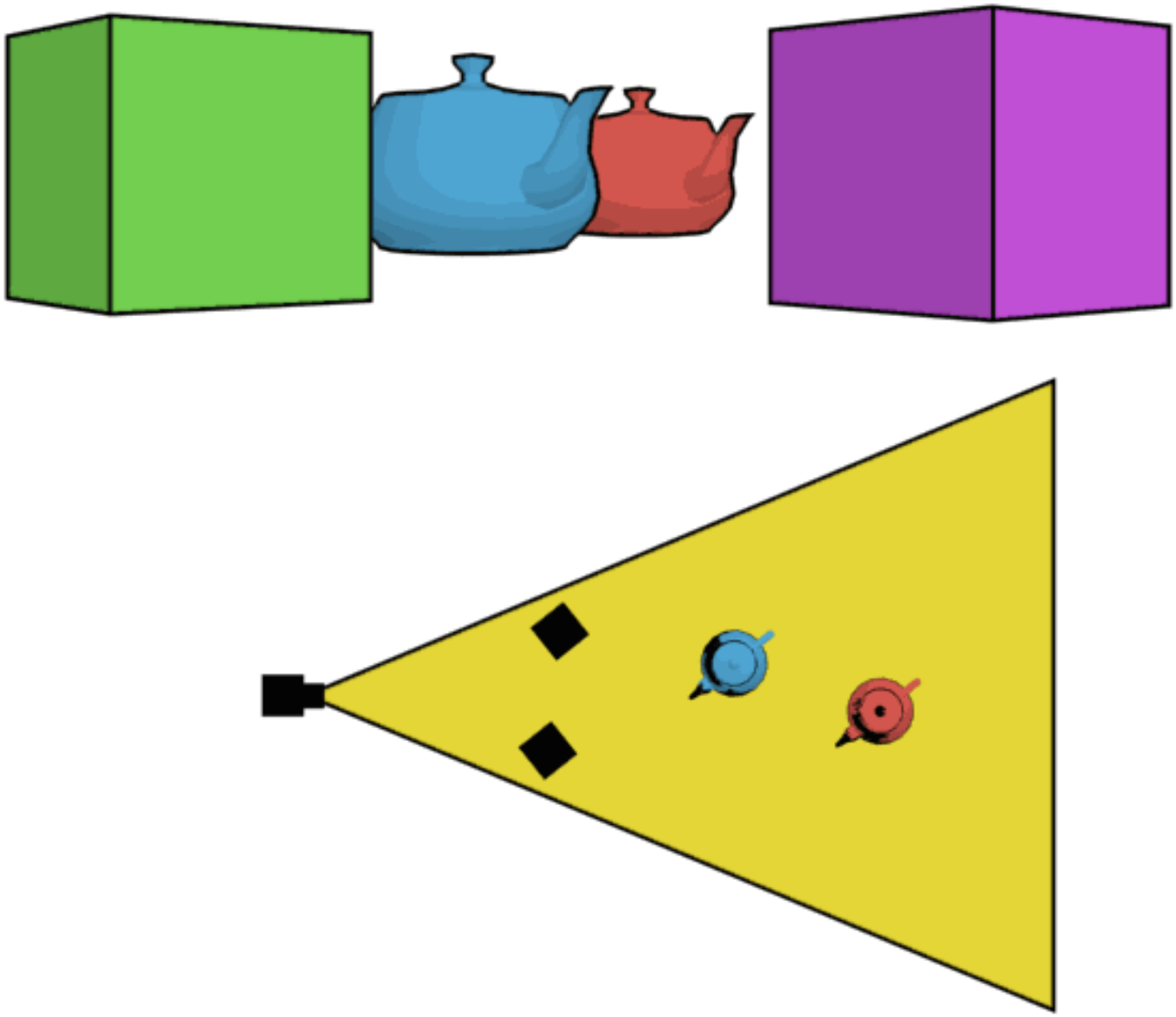
50 mm

28 mm

17 mm

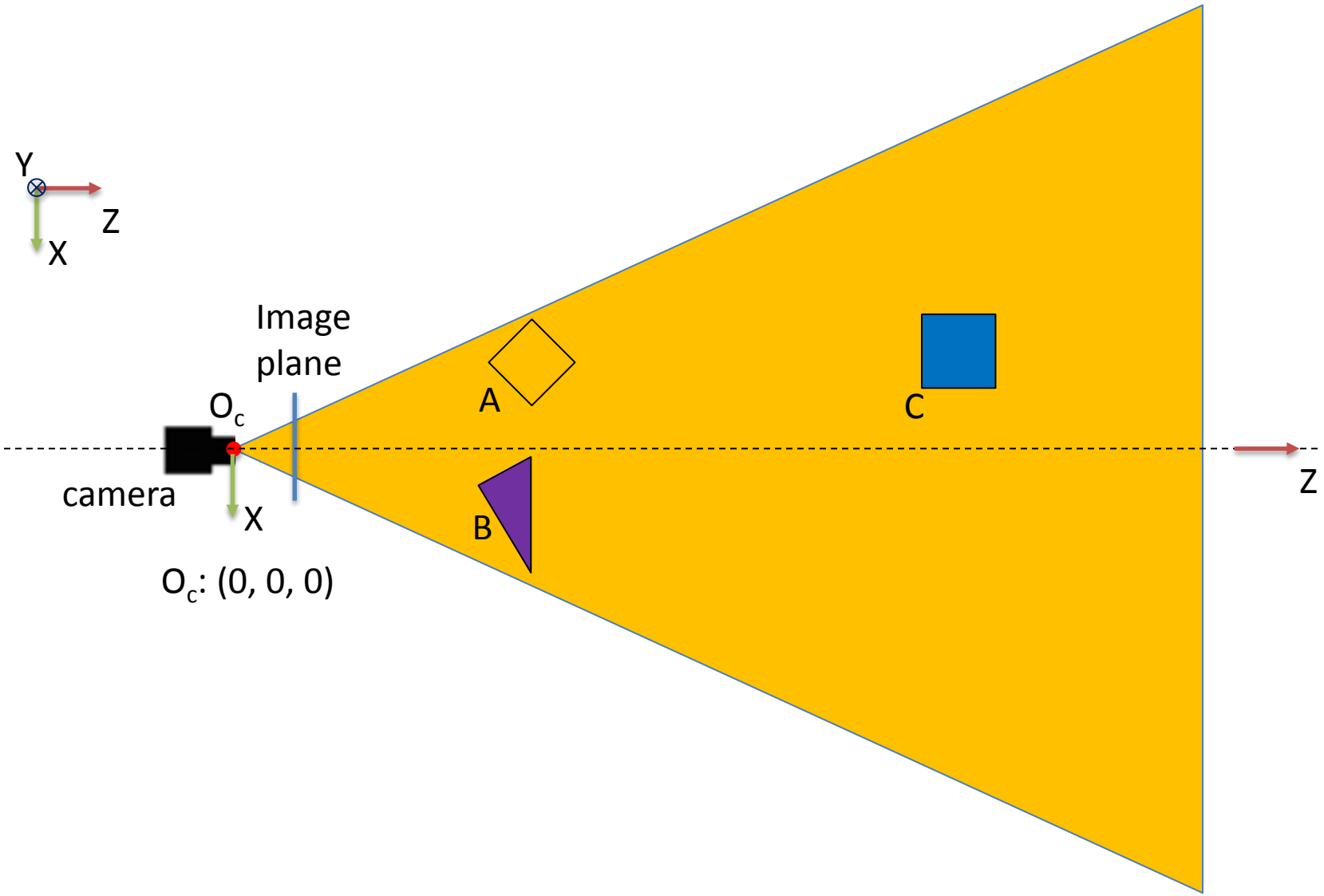
Optical center and focal change

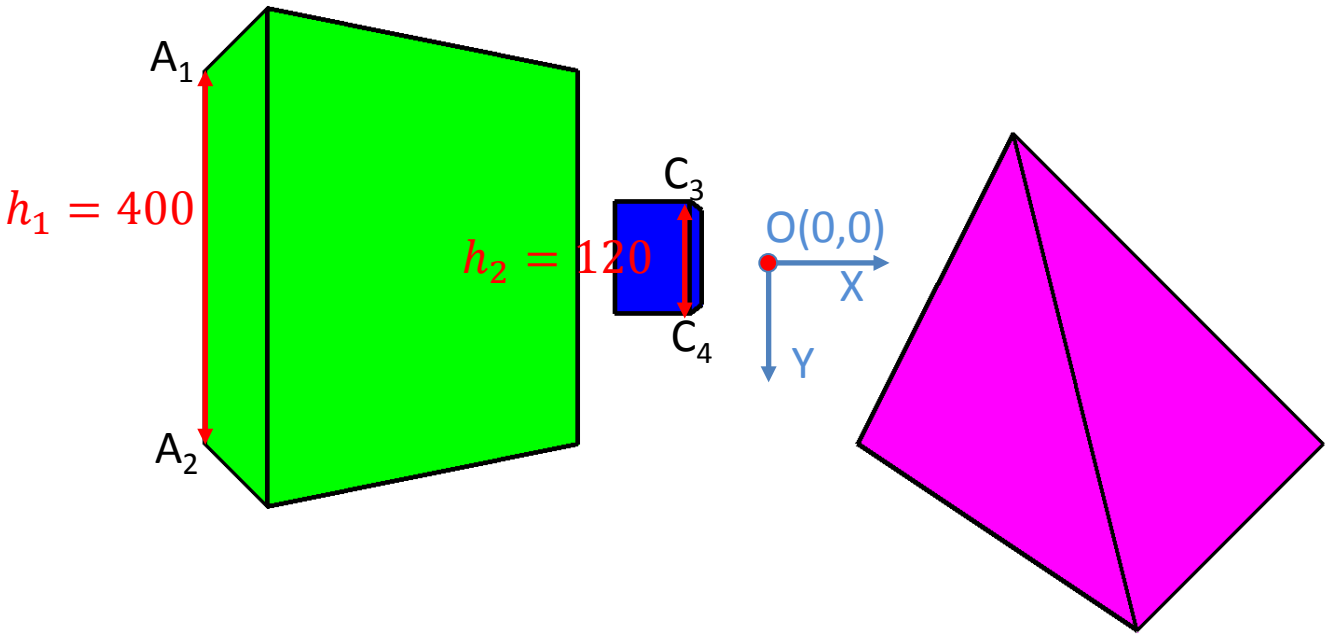

y
plane of projection
field of
view ϕ'
COP'
objects 1 &
COP''
f' – focal length
f'' – focal length
D1 – distance to object 1



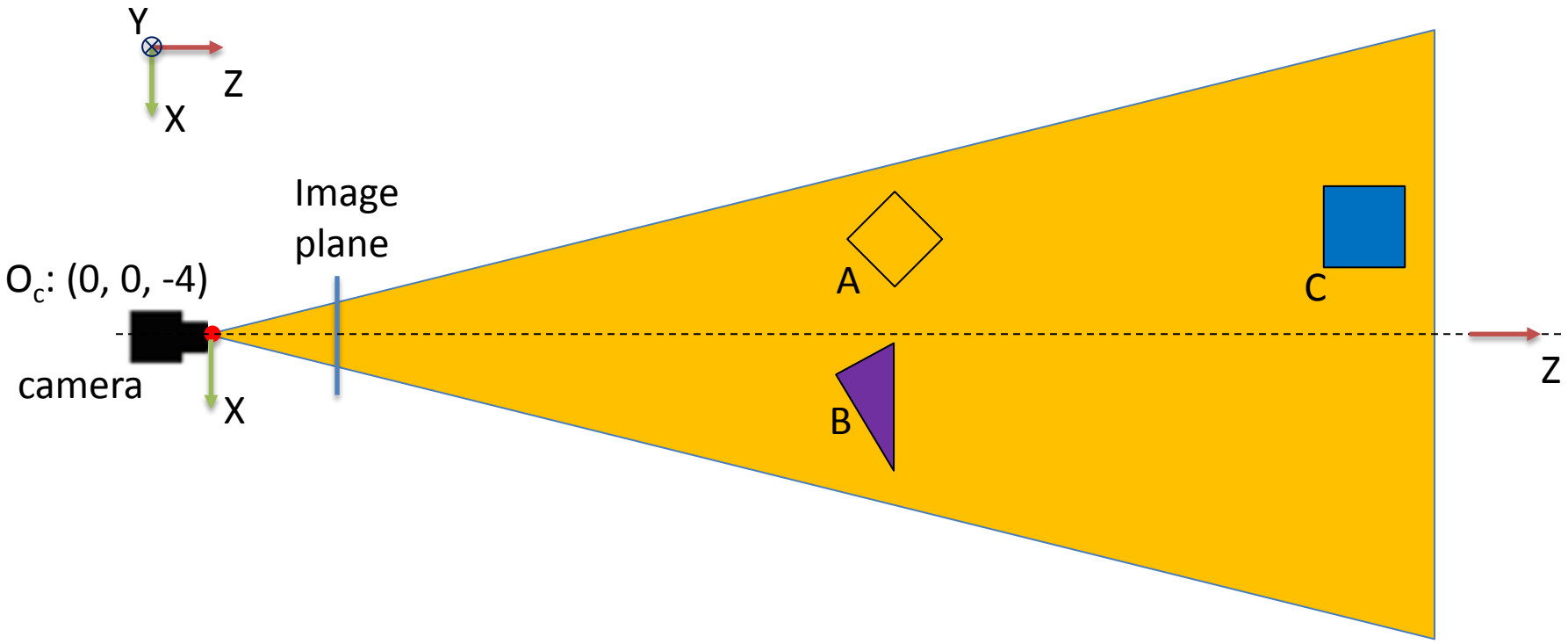
Dolly Zoom Example

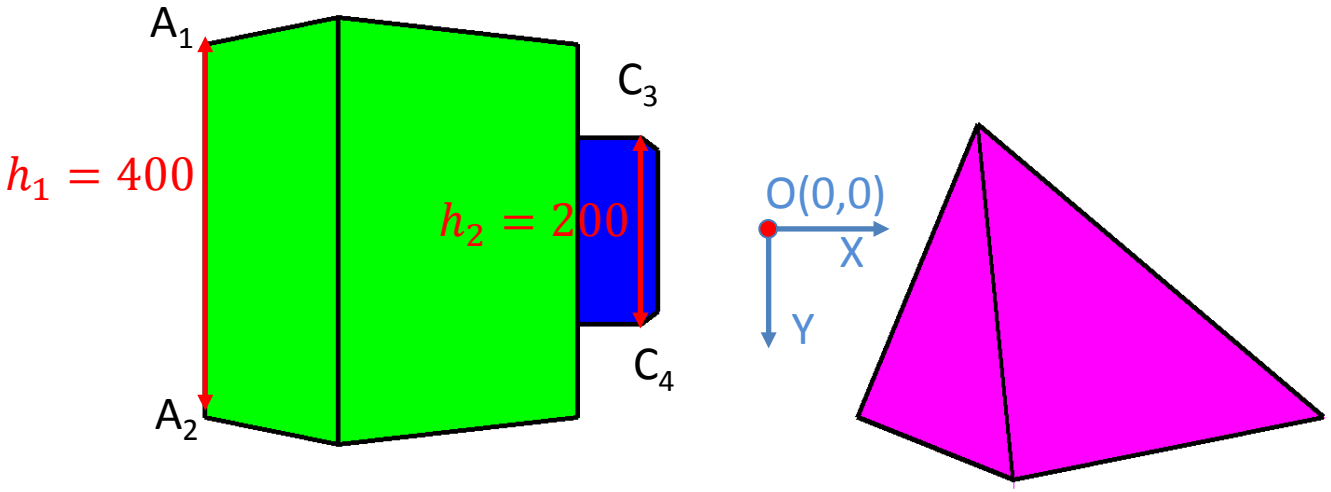
Example of Dolly zoom




A1
$h_1 = 400$
A2
$h_2 = 120$
C3
C4
O(0,0)
X
Y

Example of Dolly zoom

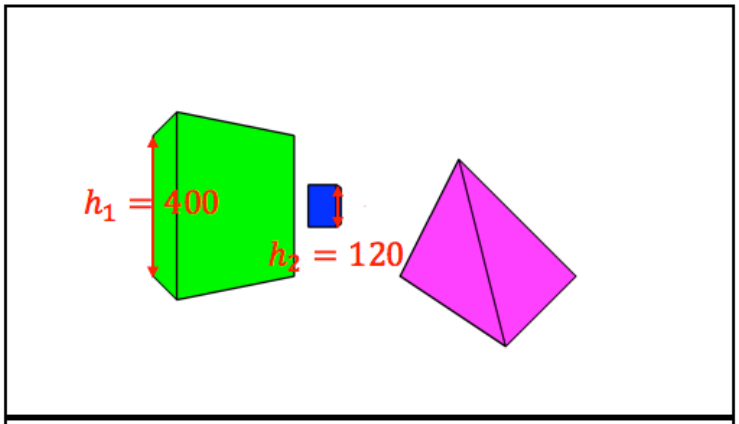


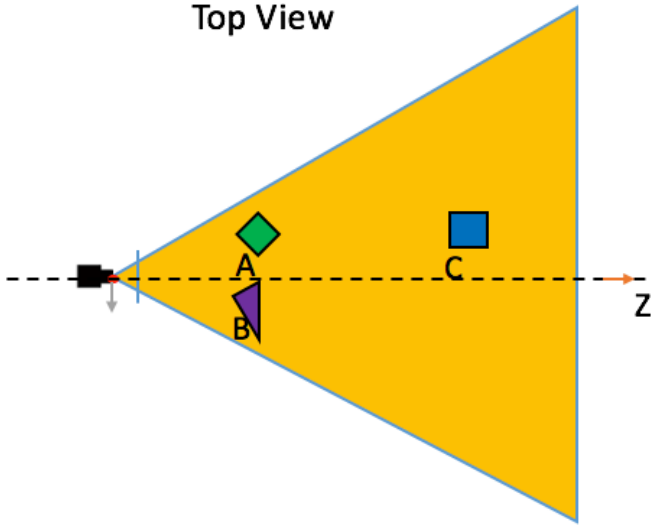

A1
C3
$h_1 = 400$
$h_2 = 200$
C4
A2
O(0,0)
X
Y

First View

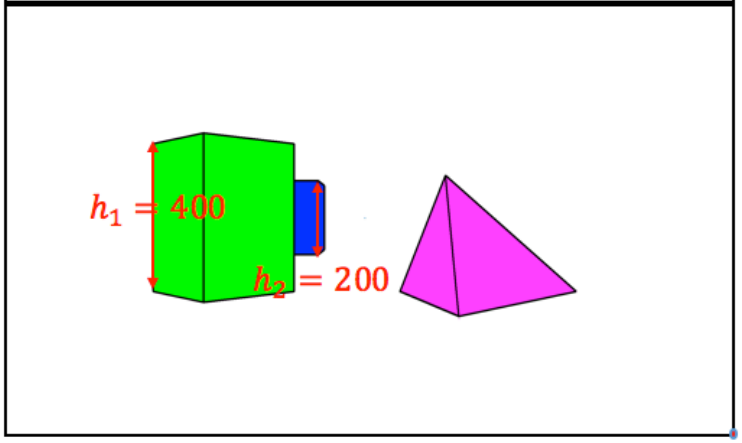
Top View

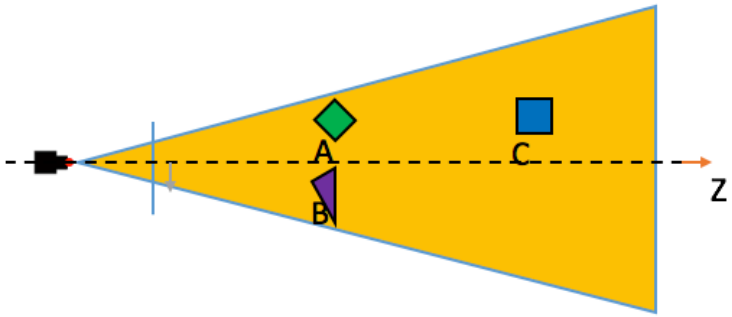
Original

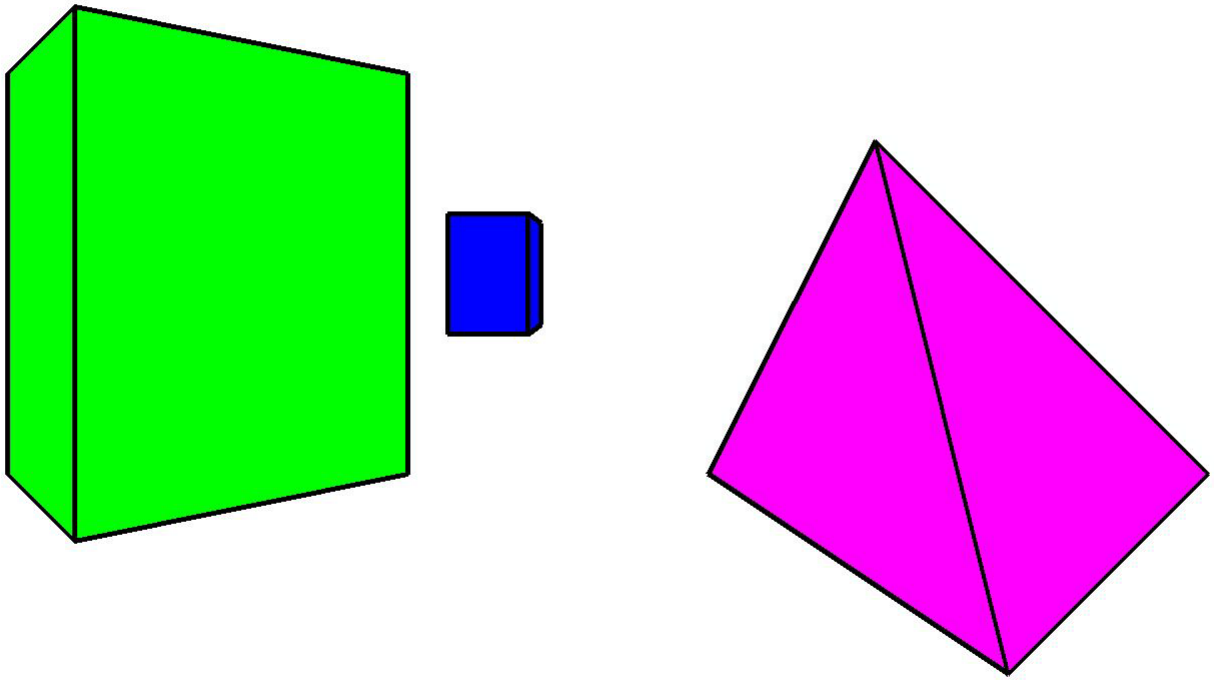




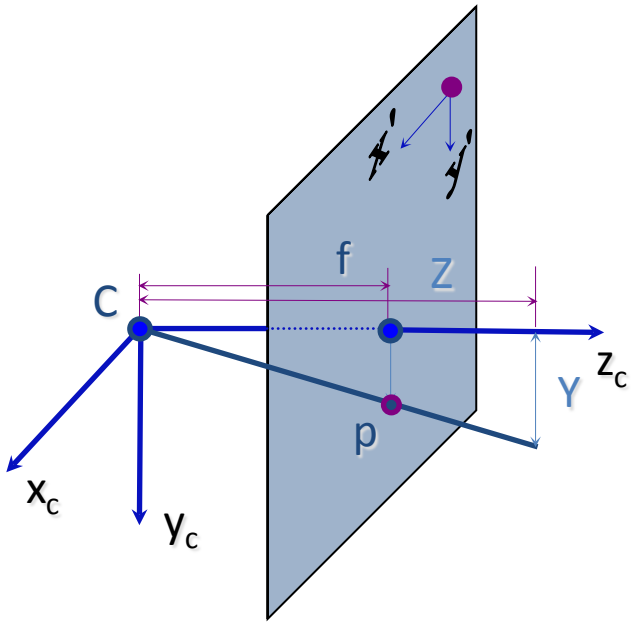
After changing camera position and Focal length







1st Person Camera world



3D to 2D image: $x' = f\frac{X}{Z}$ $y' = f\frac{Y}{Z}$

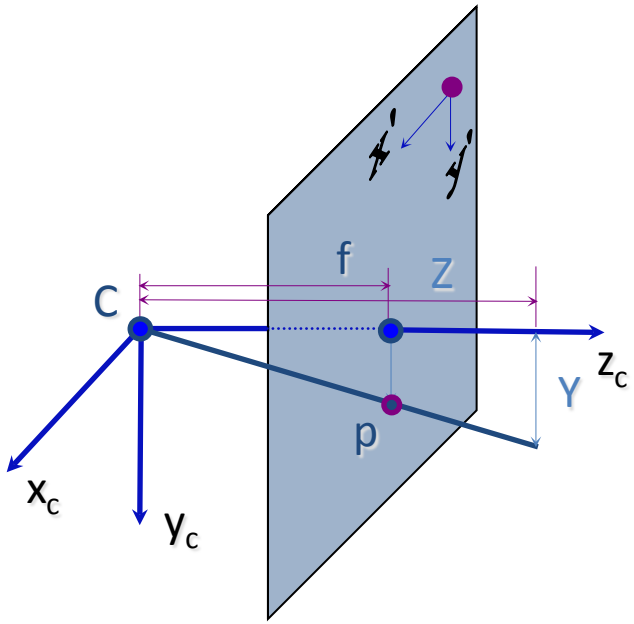
Projection equation:

Zx' = f X
Zy' = f Y
Z = Z

Step 1: Camera projection matrix

$$Zx' = fX$$
$$Zy' = fY$$
$$Z = Z$$

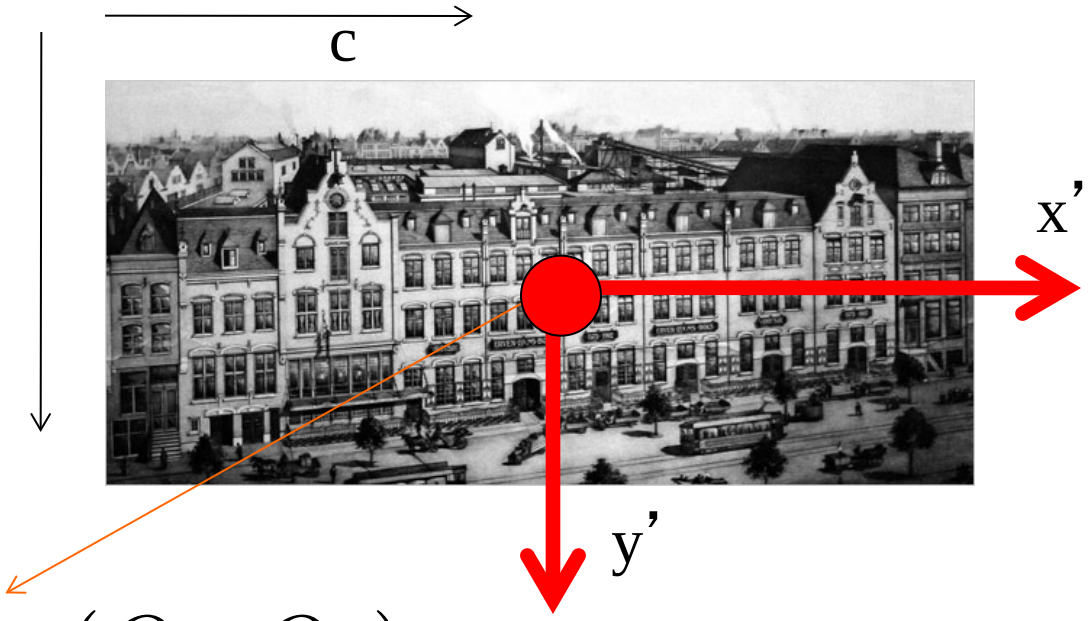
$$z_c \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \left[\begin{array}{ccc|c} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{array}\right] \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ \hline 1 \end{bmatrix}$$



$$\mathbf{x} = \mathbf{P}_0 \mathbf{X}$$

Conversion from mm to pixels

2D image to 2D pixel:



Optical Center= (O_c, O_r)

$$(c - O_c) = \frac{x'}{s_x} \qquad (r - O_r) = \frac{y'}{s_y}$$

$s_x =$ size of pixel width $\qquad s_y =$ size of pixel height

Step 2: Intrinsic camera parameters: map <u>camera</u> coordinate to *<u>pixel</u>* coordinate

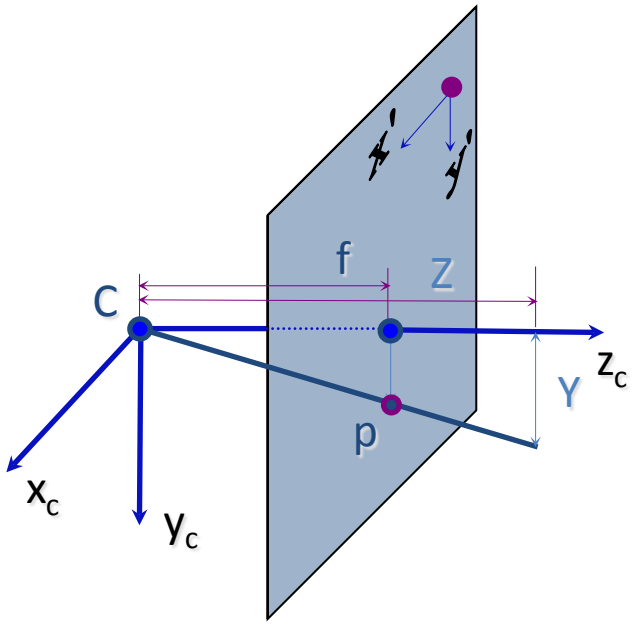
$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \alpha_x & s & p_x \\ 0 & \alpha_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

Pixel world

K

(3x3 submatrix)

Optical world



α_x, α_y is pixel scaling factor

p_x, p_y is the principle point (where optical axis hits image plane)

s is the slant factor, when the image plane is not normal to the optical axis

Combine the Intrinsic camera parameters

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \alpha_x & s & p_x \\ 0 & \alpha_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \left[\begin{array}{ccc|c} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{array}\right] \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \left[\begin{array}{ccc|c} \alpha_x f & sf & p_x & 0 \\ 0 & \alpha_y f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{array}\right] \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}$$

K

(Calibration matrix)

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \alpha_x f & sf & p_x \\ 0 & \alpha_y f & p_y \\ 0 & 0 & 1 \end{bmatrix} \left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{array}\right] \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}$$

1st Person Camera projection

$$Z\begin{bmatrix} u_{\text{img}} \\ v_{\text{img}} \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & s & p_x \\ & f_y & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I}_{3\times 3} & \mathbf{0} \end{bmatrix} \begin{bmatrix} X_{\text{cam}} \\ Y_{\text{cam}} \\ Z_{\text{cam}} \\ 1 \end{bmatrix}$$

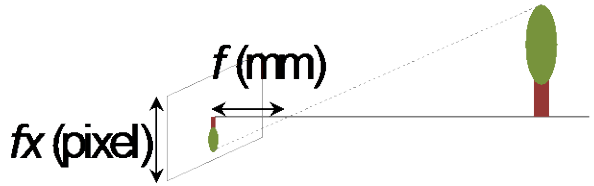
2D pixel in 1st person view

3D World in 1st person view

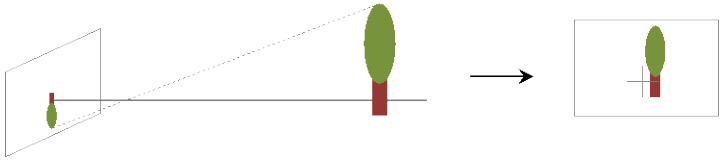
guration (internal parameter)

$$z\begin{bmatrix} u_{\text{img}} \\ v_{\text{img}} \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & s & p_x \\ & f_y & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I}_{3\times 3} & \mathbf{0} \end{bmatrix} \begin{bmatrix} X_{\text{cam}} \\ Y_{\text{cam}} \\ Z_{\text{cam}} \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & s & p_x \\ & f_y & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} X_{\text{cam}} \\ Y_{\text{cam}} \\ Z_{\text{cam}} \end{bmatrix}$$

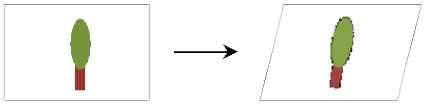
- A scale factor that converts physical focal length to pixel unit, i.e., f (mm) → f_x (pixel).



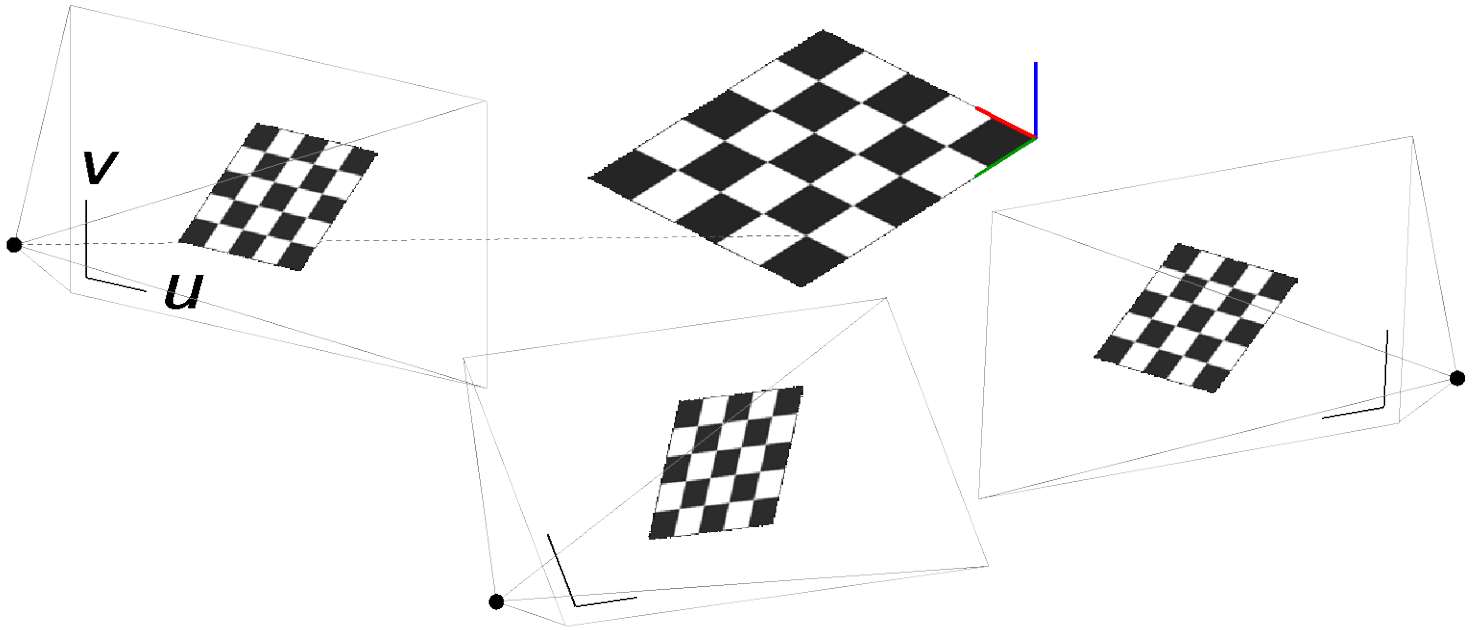
- Position of image center (principal point), i.e., p_x, p_y



- A skew factor between x and y axis of the image, i.e., $u_{img} = f_x x + sy + p_x$.



Multiple View Geometry: 3rd person view measurement

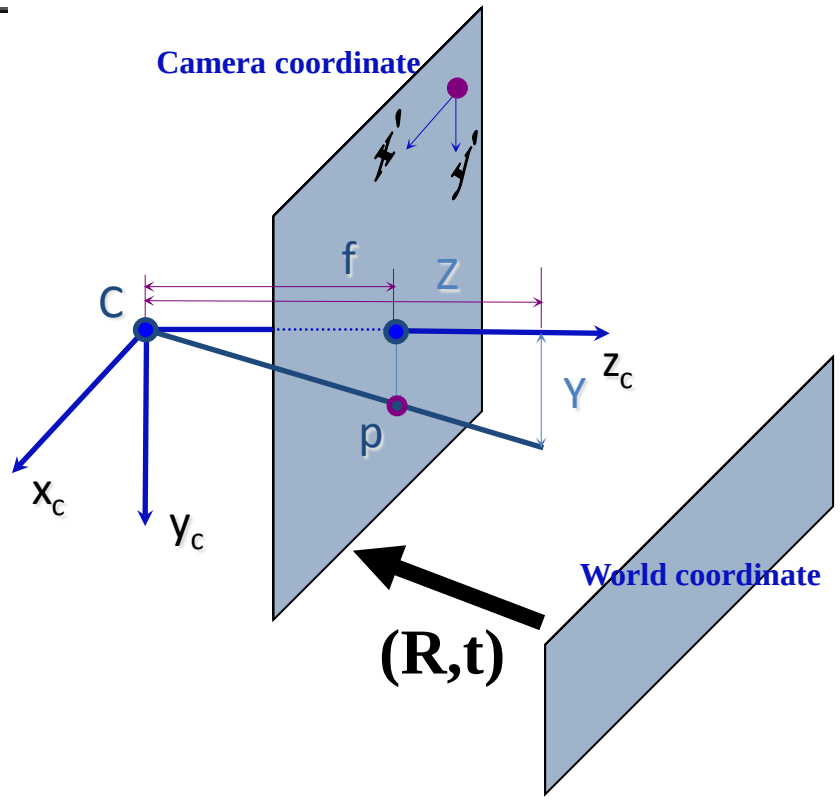


Step 3: 3rd person to 1st person 3D mapping: the world to camera coordinates

$$\mathbf{X_c} = \begin{bmatrix} R_{3x3} & t_{3x1} \\ \mathbf{0} & 1 \end{bmatrix} \mathbf{X}$$

3rd person 3D point

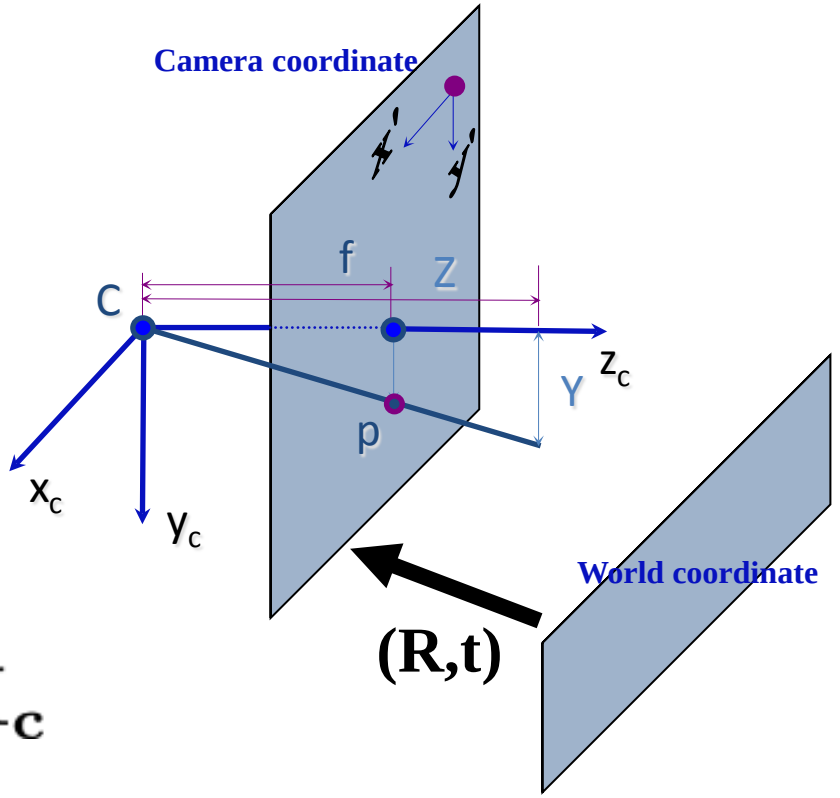
1st person 3D point



Combining Internal and External parameters

1) $$\mathbf{Xc} = \begin{bmatrix} \mathrm{R} & \mathrm{t} \\ \mathbf{0} & 1 \end{bmatrix} \mathbf{X}$$

2) $$\mathbf{x} = \mathbf{K}_{3\times3}\,[\mathbf{I};0]_{3\times4}\,\mathbf{X_c}$$



1) Translate the world coordinate into the camera coordinate
2) Translate the camera coordinate into the pixel coordinate

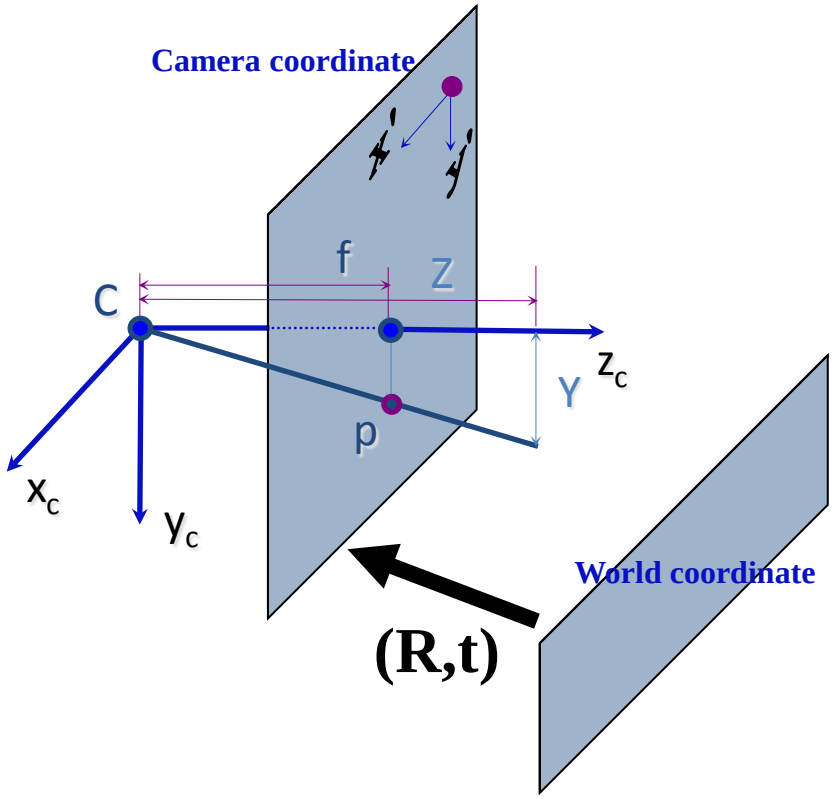
Combining Internal and External parameters

$$\mathbf{Xc} = \begin{bmatrix} \mathrm{R} & \mathrm{t} \\ \mathbf{0} & 1 \end{bmatrix} \mathbf{X}$$

$$\mathbf{x} = \mathbf{K}_{3\times3}\ [\mathbf{I}; 0]_{3\times4}\ \mathbf{X_c}$$

After simplication:

$$x = K\ [R,\ t]\ X$$

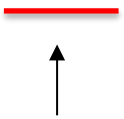


Ideal Perspective Projection

$$z\begin{bmatrix} u_{\text{img}} \\ v_{\text{img}} \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & s & p_x \\ & f_y & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Ideal Perspective Projection

$$z\begin{bmatrix} u_{\text{img}} \\ v_{\text{img}} \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & s & p_x \\ & f_y & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$



3D World in 3rd person view

Ideal Perspective Projection

$$z\begin{bmatrix} u_{\text{img}} \\ v_{\text{img}} \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & s & p_x \\ & f_y & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

3D World in 1st person view

Ideal Perspective Projection

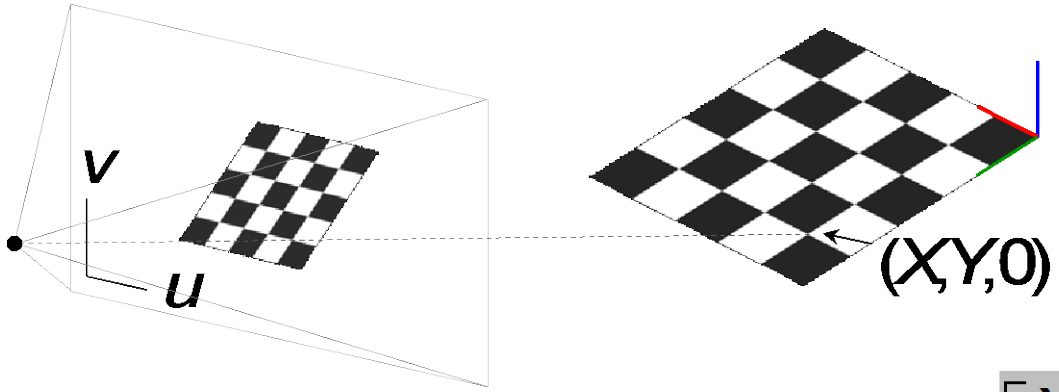
$$z\begin{bmatrix} u_{\text{img}} \\ v_{\text{img}} \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & s & p_x \\ & f_y & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

2D pixels in 1st person view

Camera Projection

$$\begin{bmatrix} x \\ 1 \end{bmatrix} = L \left(K \begin{bmatrix} R & t \end{bmatrix} \begin{bmatrix} X \\ 1 \end{bmatrix} \right)$$

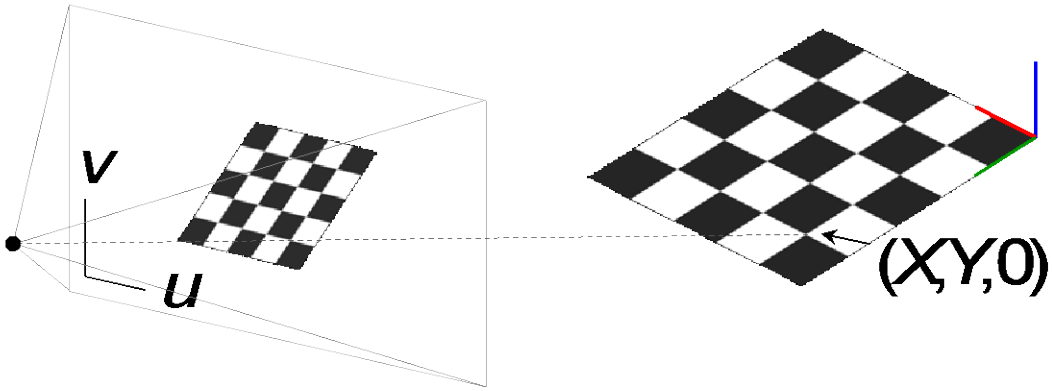
Special case, planar world, homograph



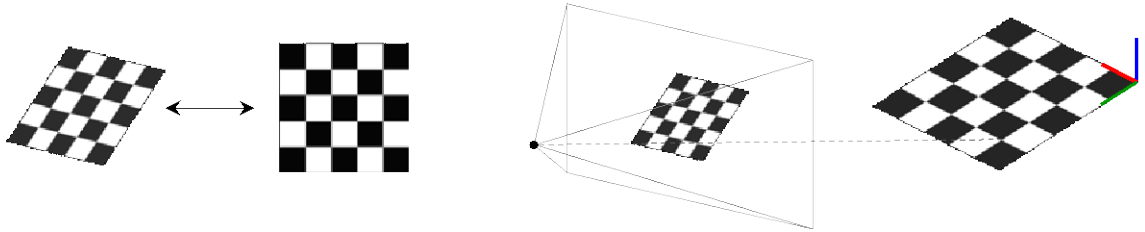
$$z\begin{bmatrix} u_{img} \\ v_{img} \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & s & p_x \\ & f_y & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix}$$

x K $R \in \mathbb{R}^{3\times3}$ t X

Special case, planar world, homograph



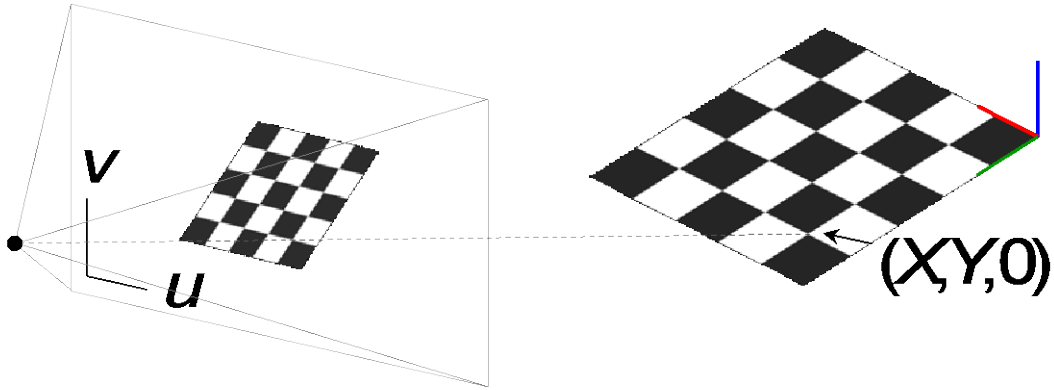
$$z\begin{bmatrix} u_{img} \\ v_{img} \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & s & p_x \\ & f_y & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

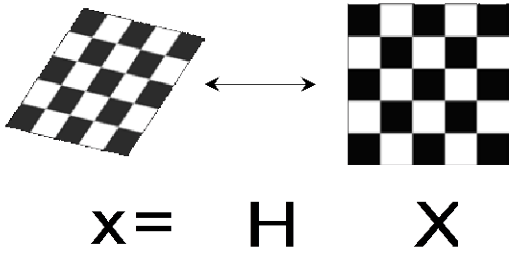


$$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} = z' \begin{bmatrix} f_x & s & p_x \\ & f_y & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & t_1 \\ r_{21} & r_{22} & t_2 \\ r_{31} & r_{32} & t_3 \end{bmatrix}$$

$\mathbf{h}_1 \quad \mathbf{h}_2 \quad \mathbf{h}_3$ $\qquad$ $\mathbf{K}$ $\qquad$ $\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{t}_1$

Special case, planar world, homograph





x = H X

$$z\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

H homography between 3D plane and 2D image plane.

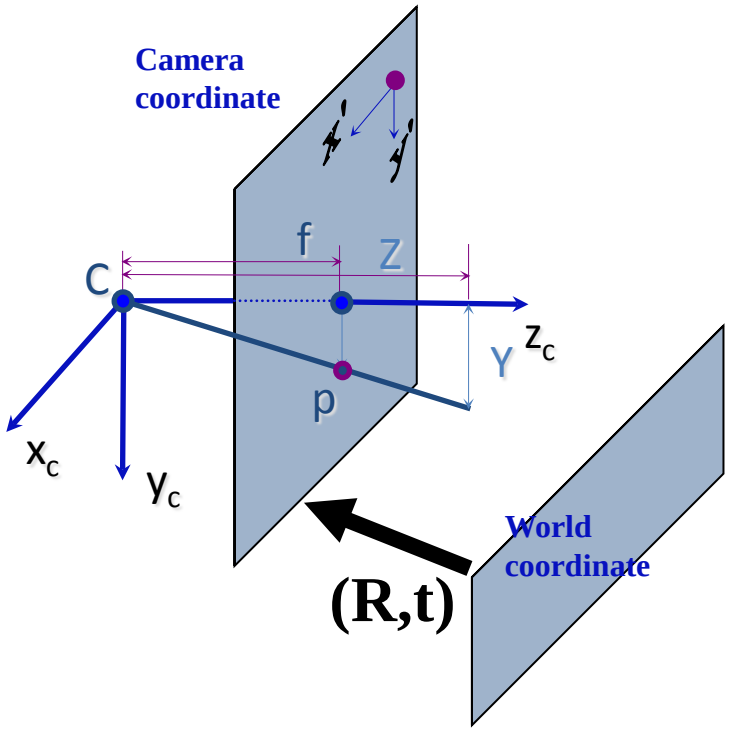
Special case: rotating camera

$$\mathbf{x} = K\,[R,\, t]\,\mathbf{X}$$

with $t = 0$

Expand:

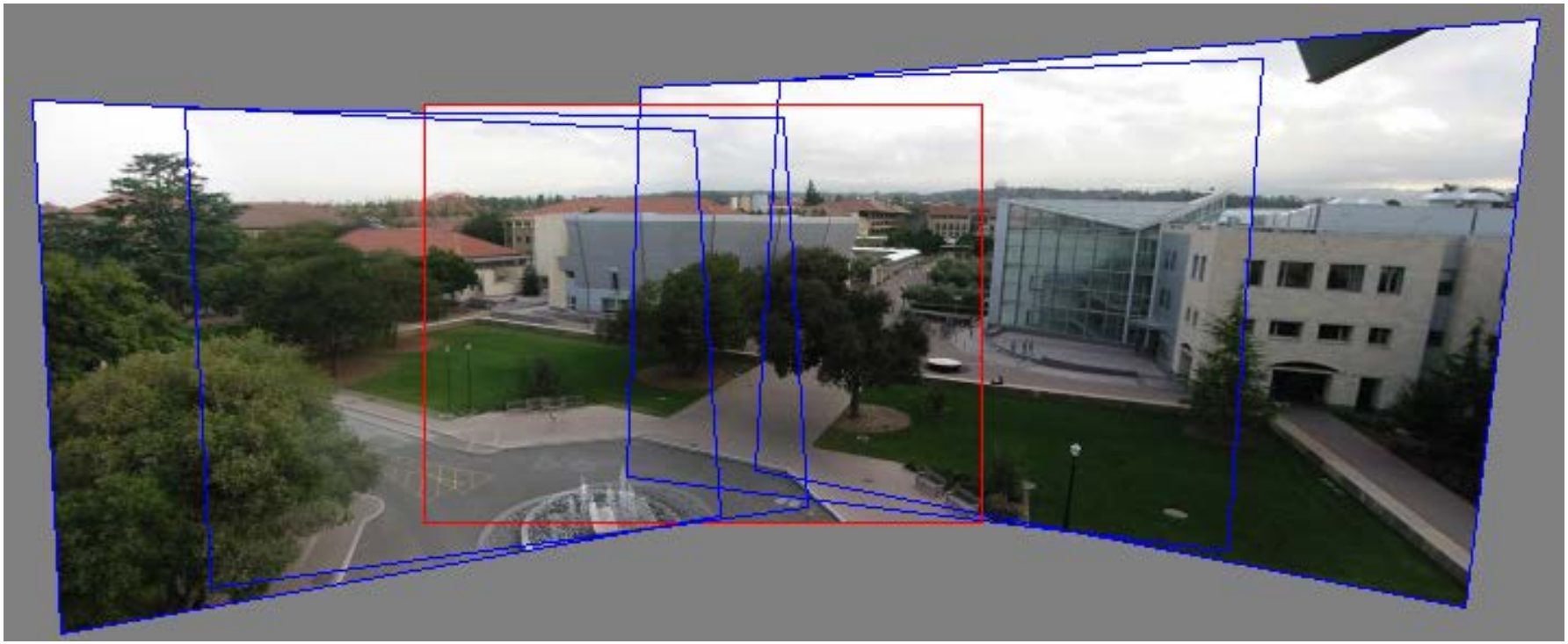
$$\mathbf{x} = K\,[R]\,\mathbf{X}$$



$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_w \\ \mathbf{y}_w \\ \mathbf{z}_w \end{pmatrix}$$

This is also a homography with $H = R$

Panoramas

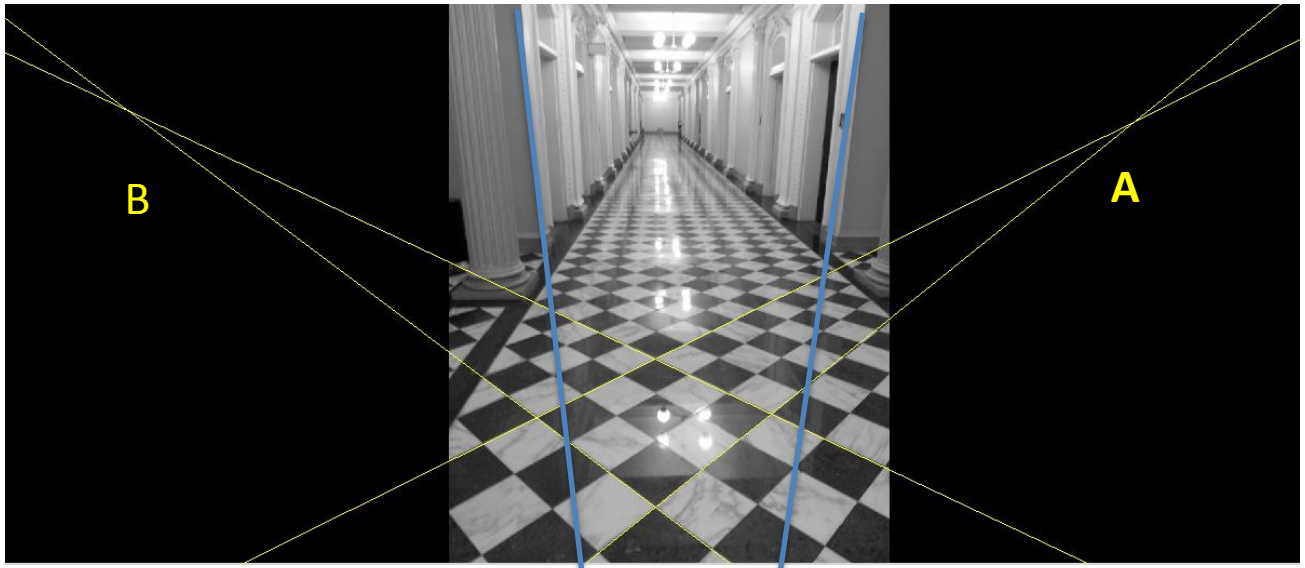


$$\mathbf{x} = K\,[R]\,\mathbf{X}$$

Perception:
How to compute intrinsics from vanishing points

Kostas Daniilidis

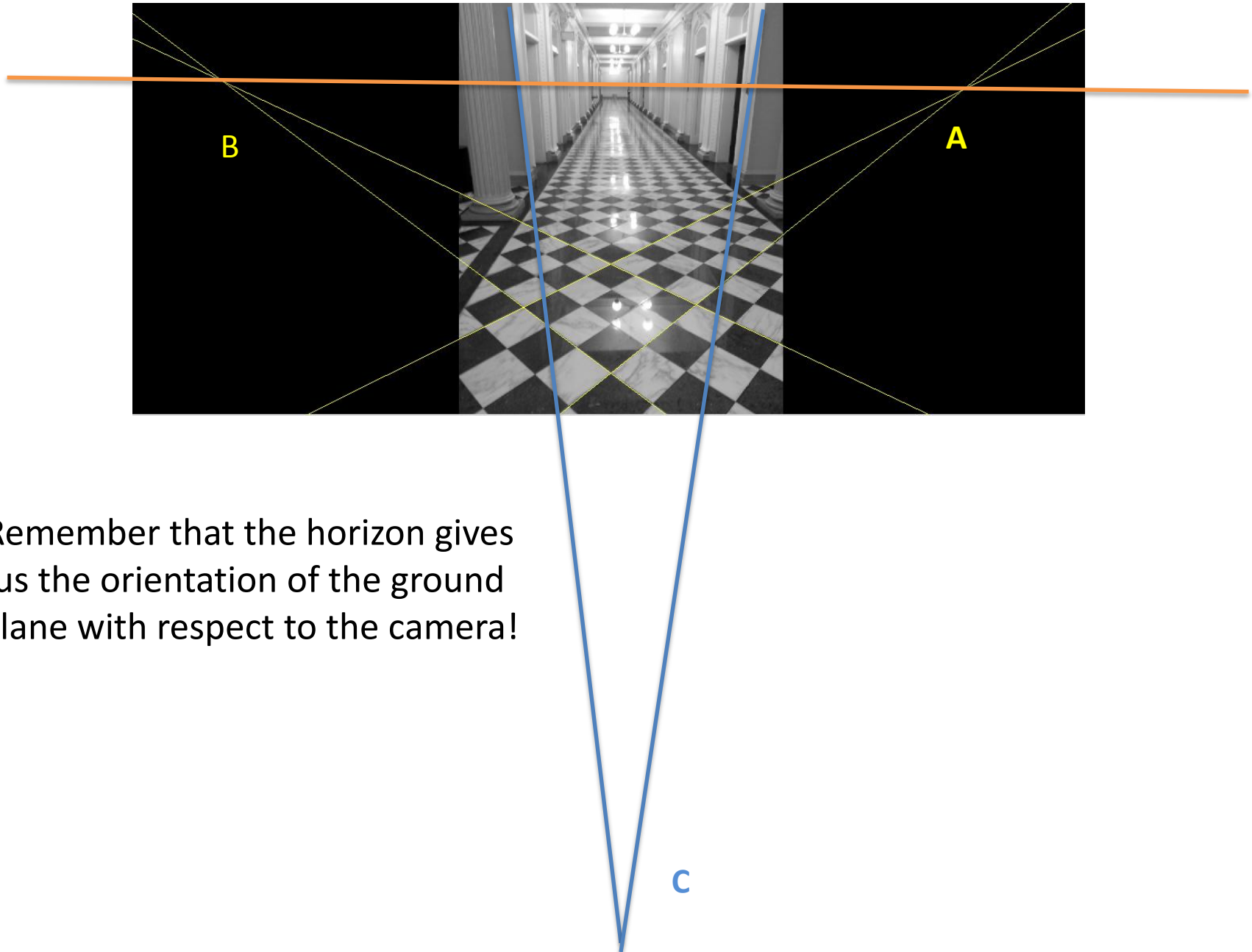
Manhattan world: A scene with three orthogonal sets of parallel lines



Three orthogonal sets of parallel lines create three orthogonal vanishing points

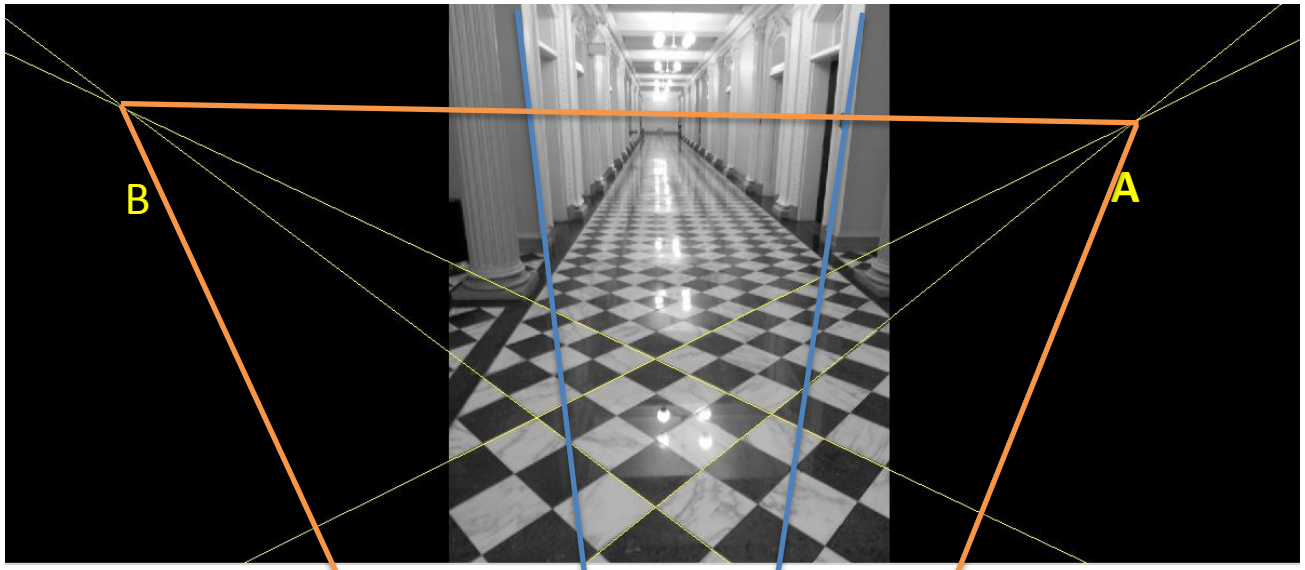
C

Line connecting AB is the horizon!



Remember that the horizon gives us the orientation of the ground plane with respect to the camera!

C is the vertical vanishing point!

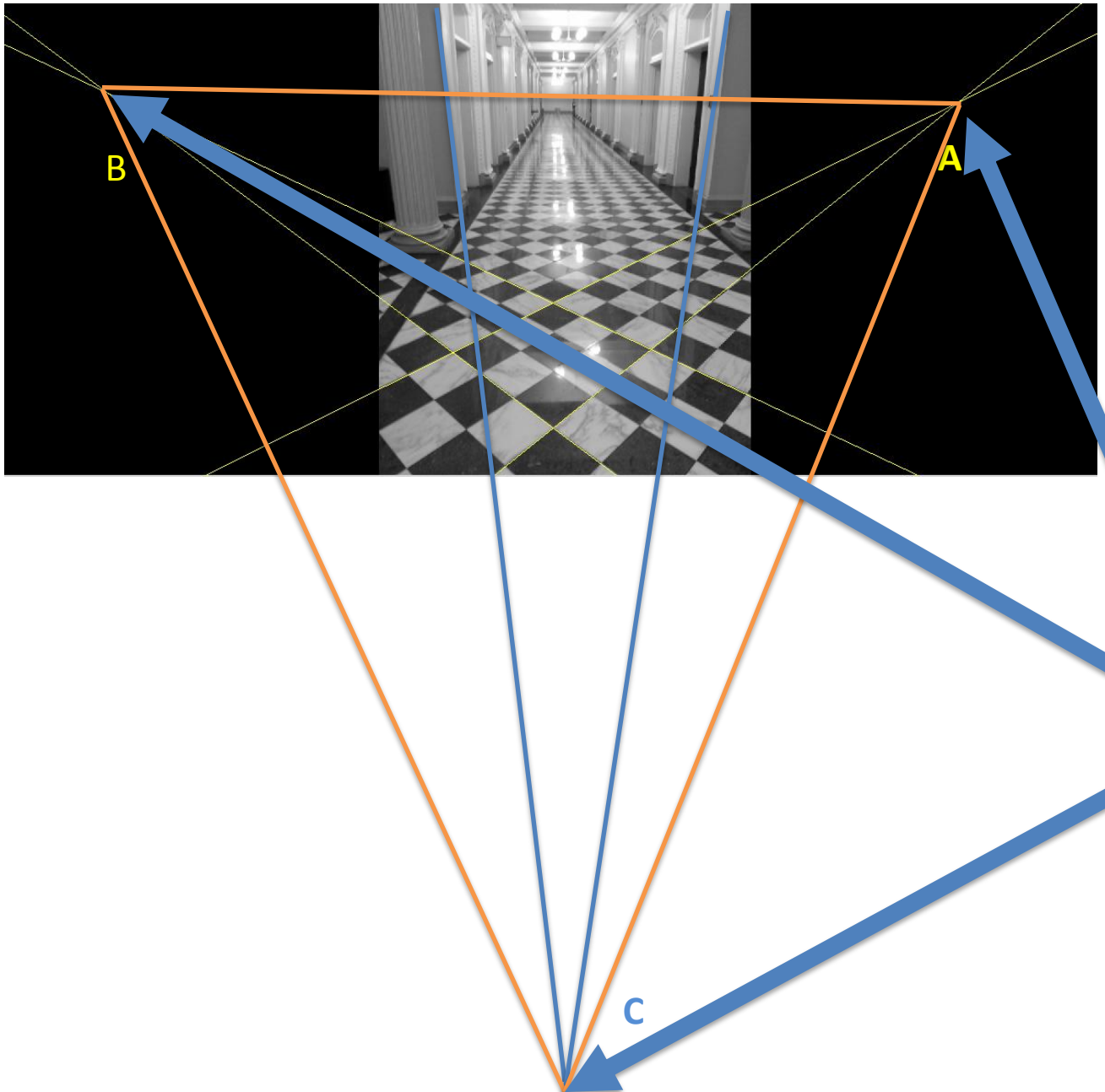


Obvious question: If the horizon AB gives us information about the ground plane and C corresponds to the vertical then shouldn't be C determined by AB ?

The answer is no because we omitted the influence of the focal length and the image center.

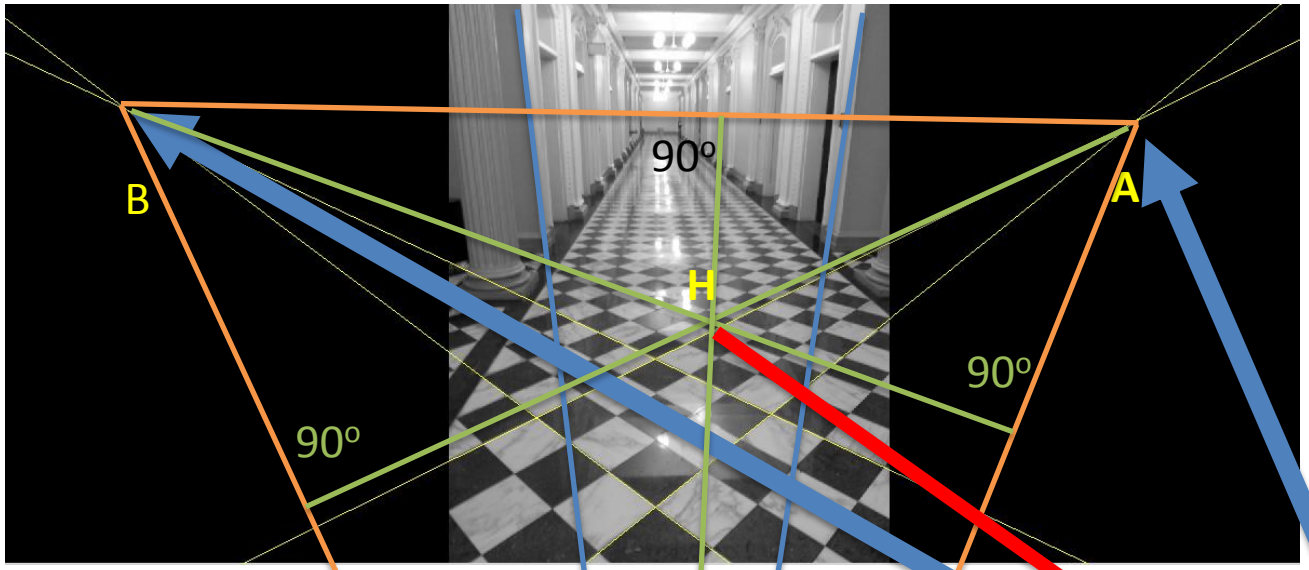
C

Let’s look at ABC as a tetrahedron OABC incl the projection center



O is Projec
center

Let H be the orthocenter of the triangle ABC



90°

90°

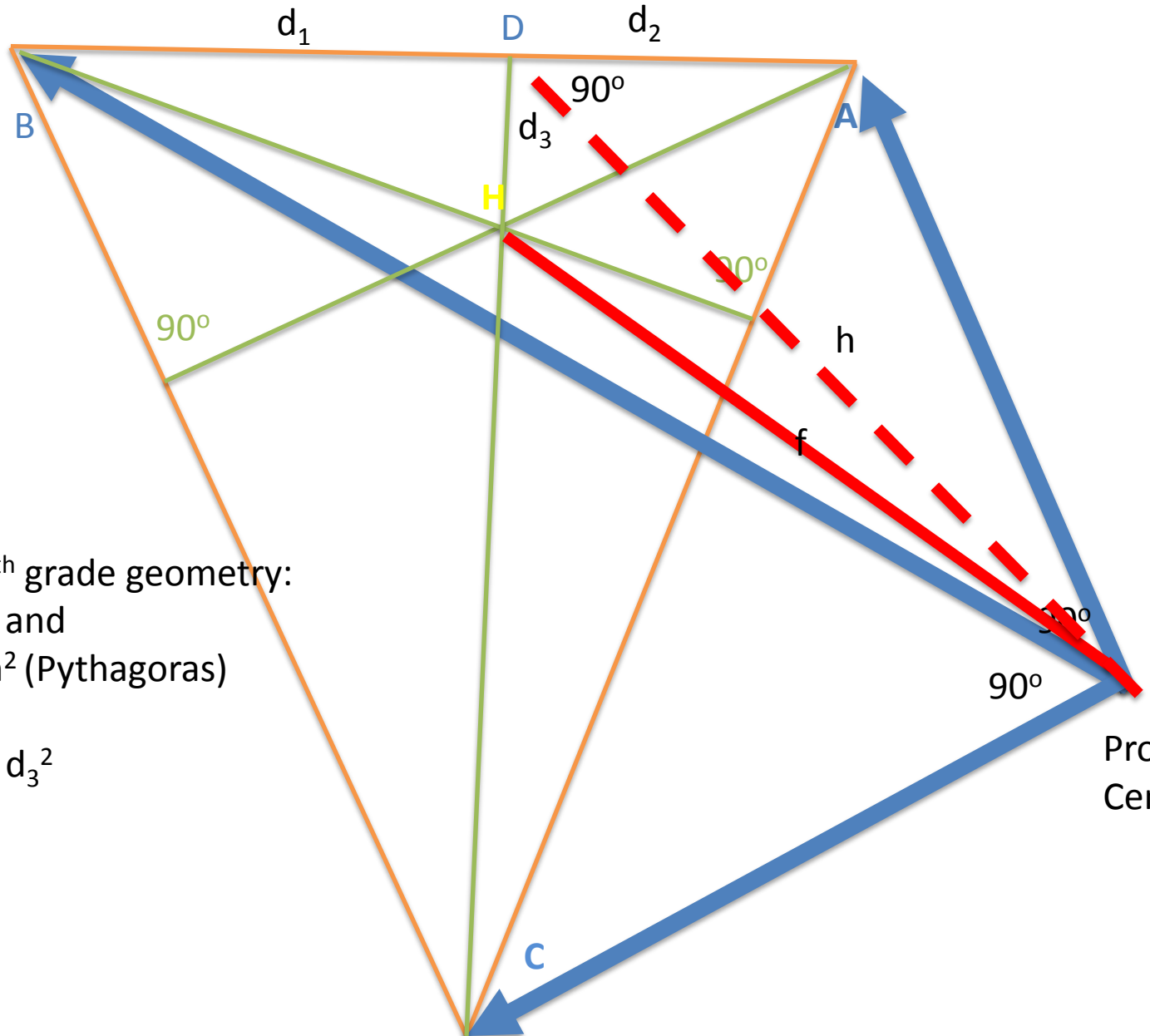
Projection Center O

C

Theorem from Euclidean Geometry:
If H is the orthocenter of ABC and all three angles AOB, BOC, and COA are right angles , the OH is perpendicular to ABC plane!

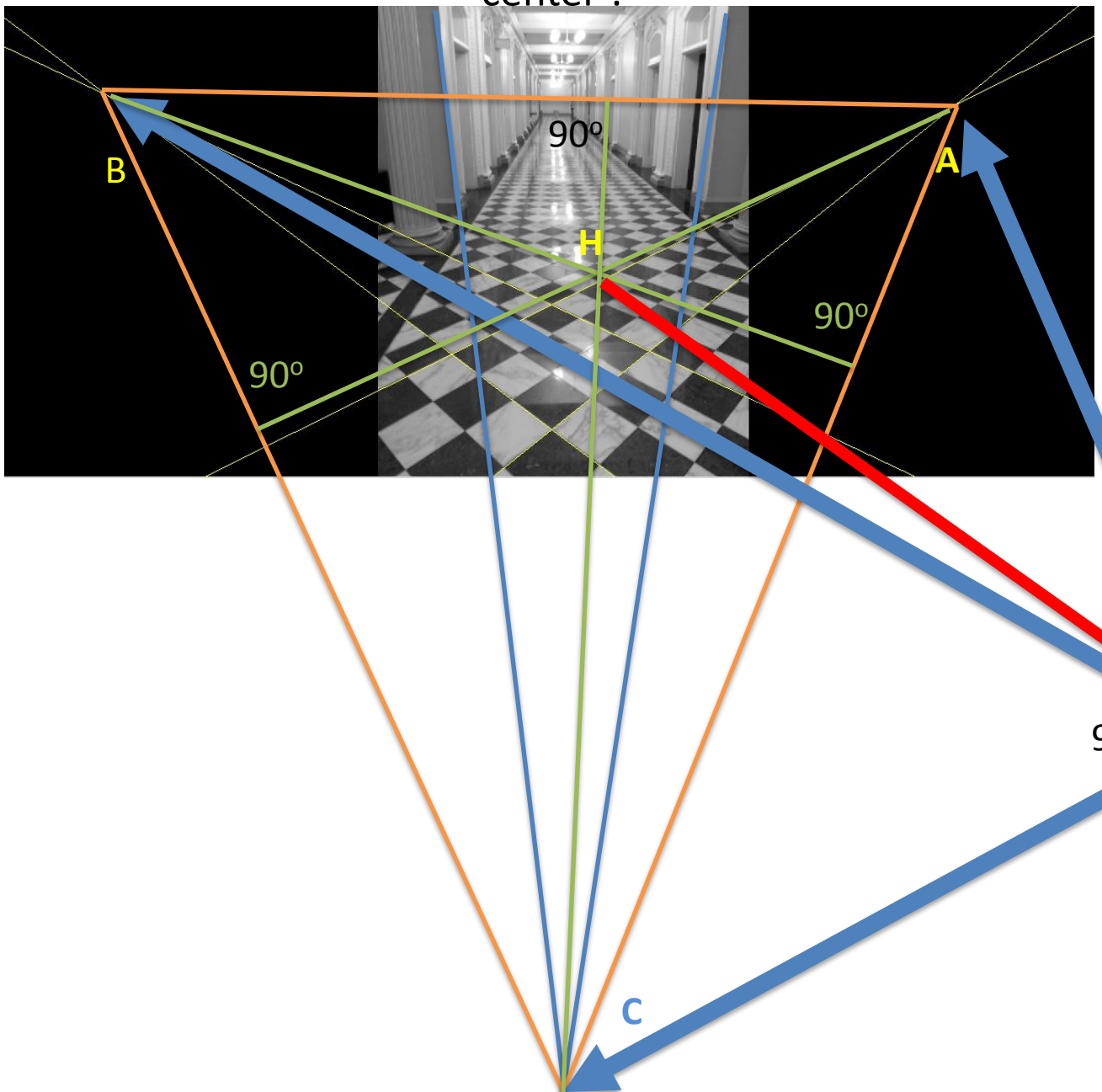
OH is the optical axis and ABC is the image plane, hence, H is the image center

We found the image center! What about the focal length (f=OH)?
Can it be computed from A,B, and C ?



Simple 9th grade geometry:
$h^2 = d_1 d_2$ and
$f^2 + d_3^2 = h^2$ (Pythagoras)
Hence
$f^2 = d_1 d_2 - d_3^2$

Three orthogonal vanishing points allow computation of focal length and image center !



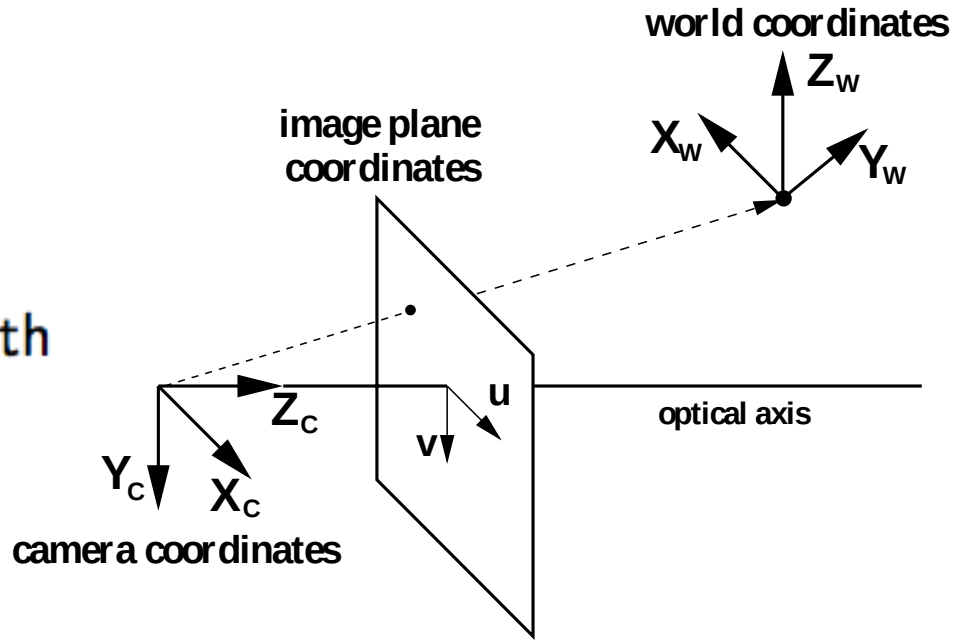
Camera Calibration

Kostas Daniilidis

The 3x4 projection matrix P

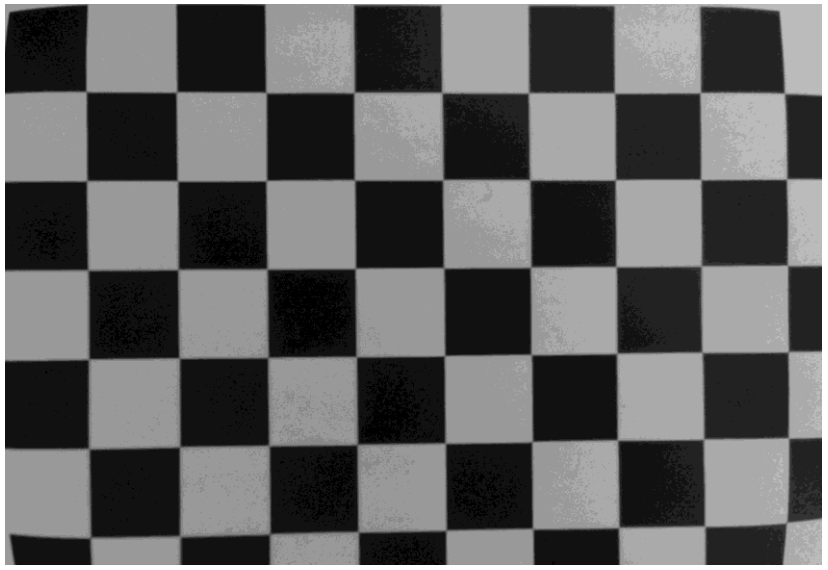
$$\lambda \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f & 0 & u_o \\ 0 & f & v_o \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} R & t \end{pmatrix} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} = P \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix}$$

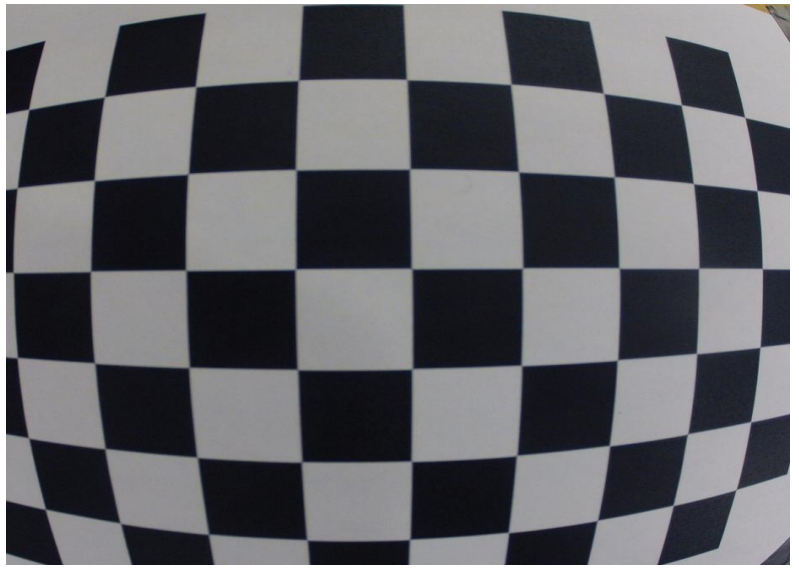
λ is the unknown depth



Cameras with large field of view have radial distortions

$$
\begin{aligned}
u^{dist} &= u(1 + k_1 r + k_2 r^2 + k_3 r^3 + \ldots) \\
v^{dist} &= v(1 + k_1 r + k_2 r^2 + k_3 r^3 + \ldots) \\
\text{where} \quad r^2 &= u^2 + v^2
\end{aligned}
$$

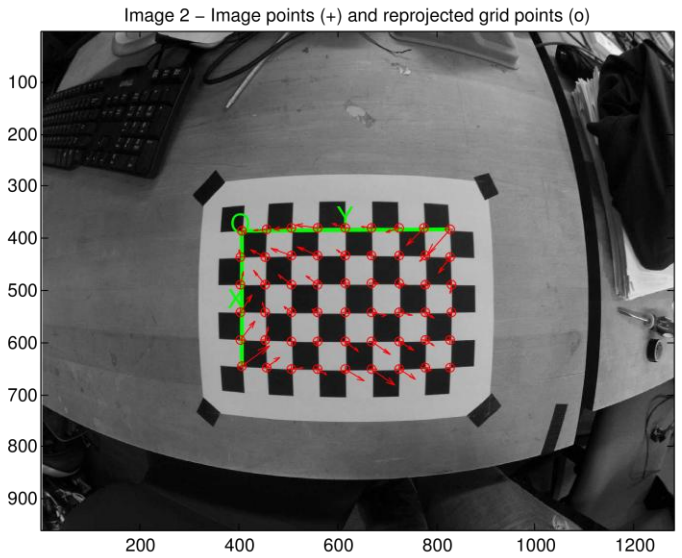


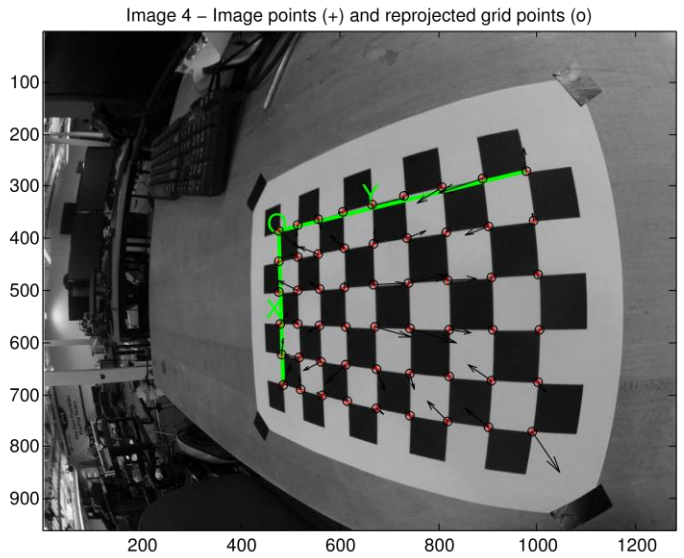


A procedure called calibration

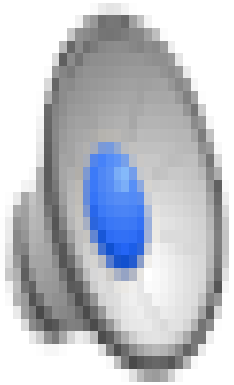
Estimates the *intrinsic parameters*

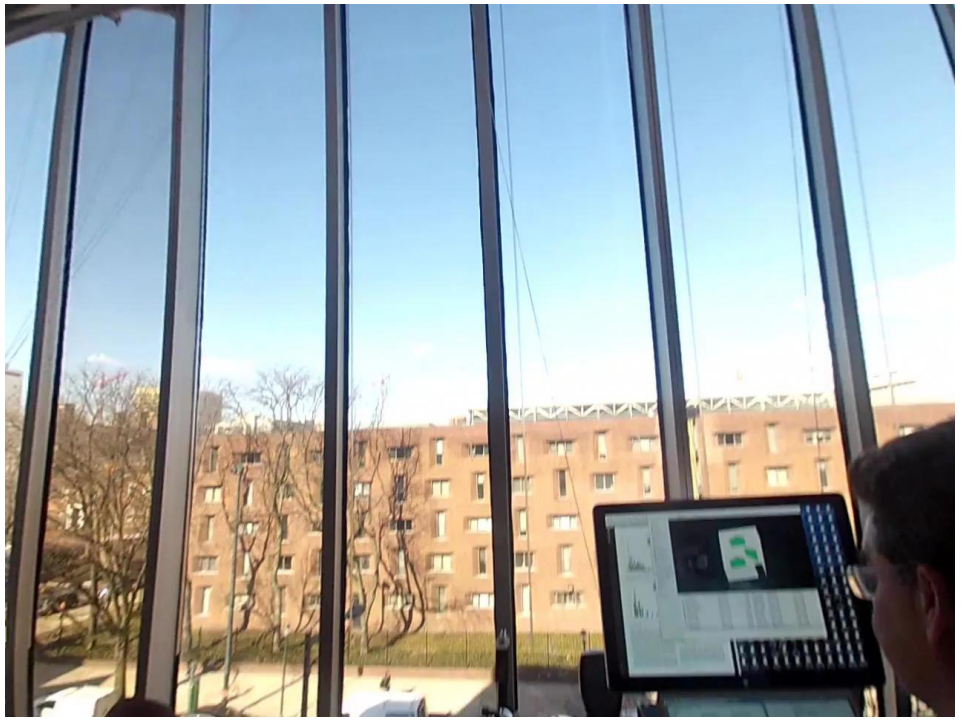
- f focal length
- (u_o, v_o) image center
- $k_1, k_2, \ldots$ radial distortion parameters



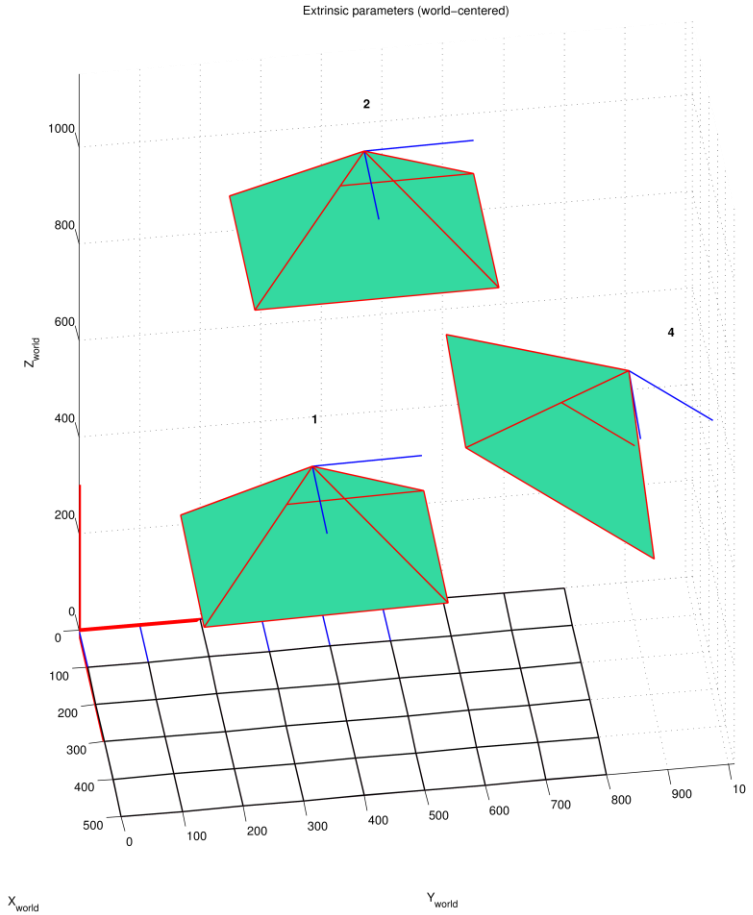


As a result of the calibration we have undistorted images and video





..as well as the poses of the camera and the projection rays in world coordinates

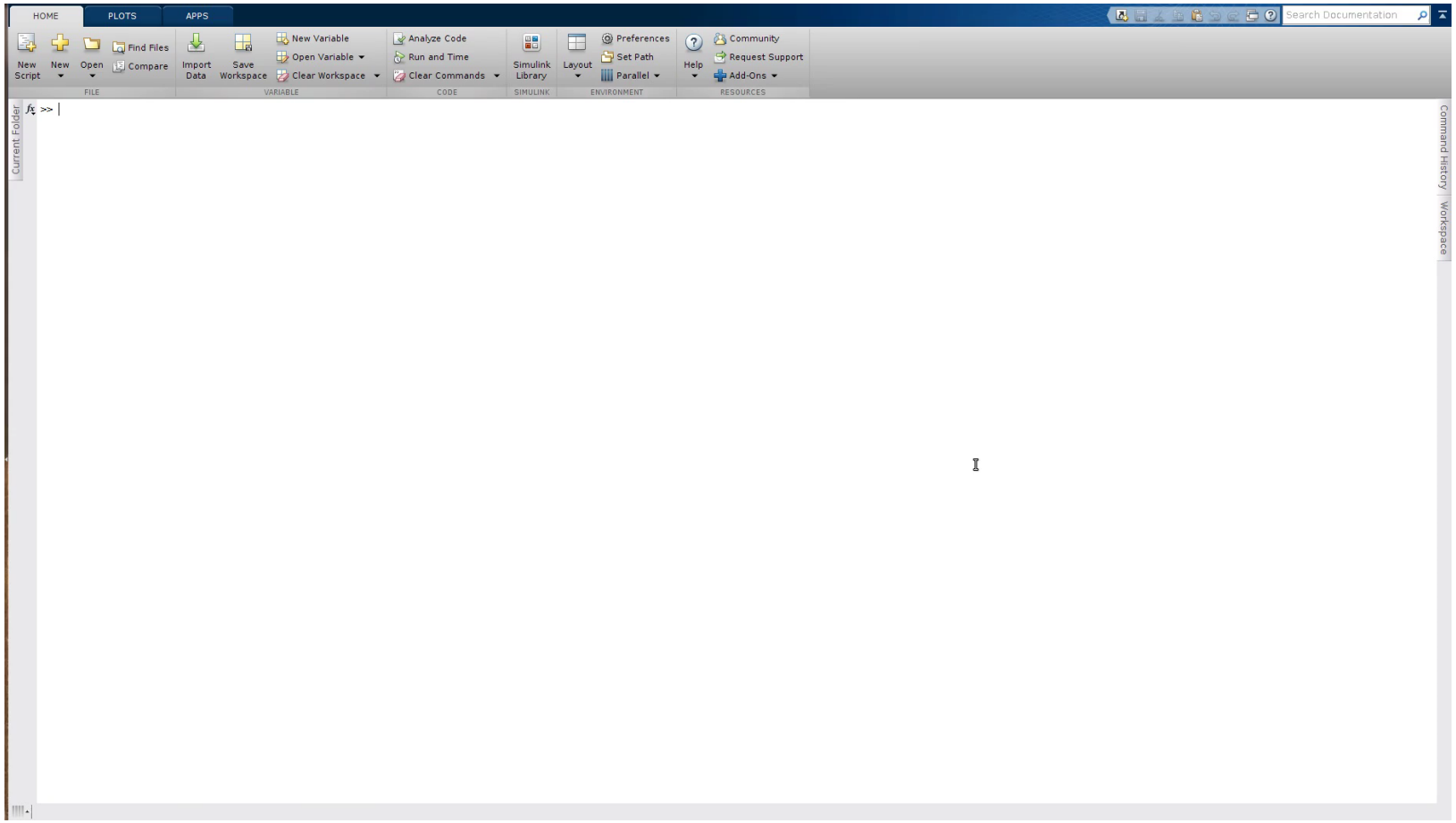


$$\begin{pmatrix} X_w \\ Y_w \\ Z_w \end{pmatrix} = -R^T T + \lambda R^T K^{-1} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}$$

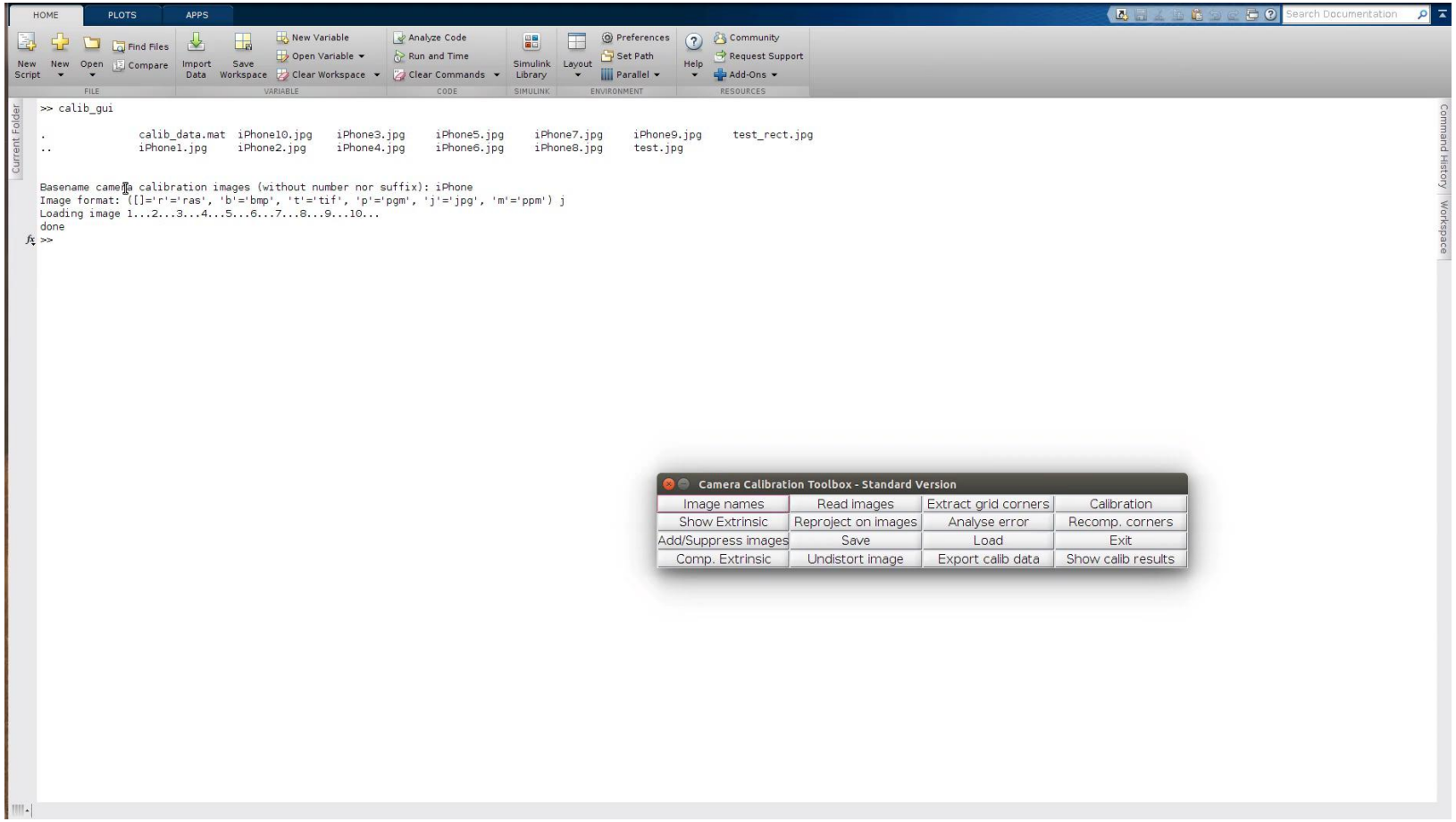
known

We will return later on the specifics of calibration….

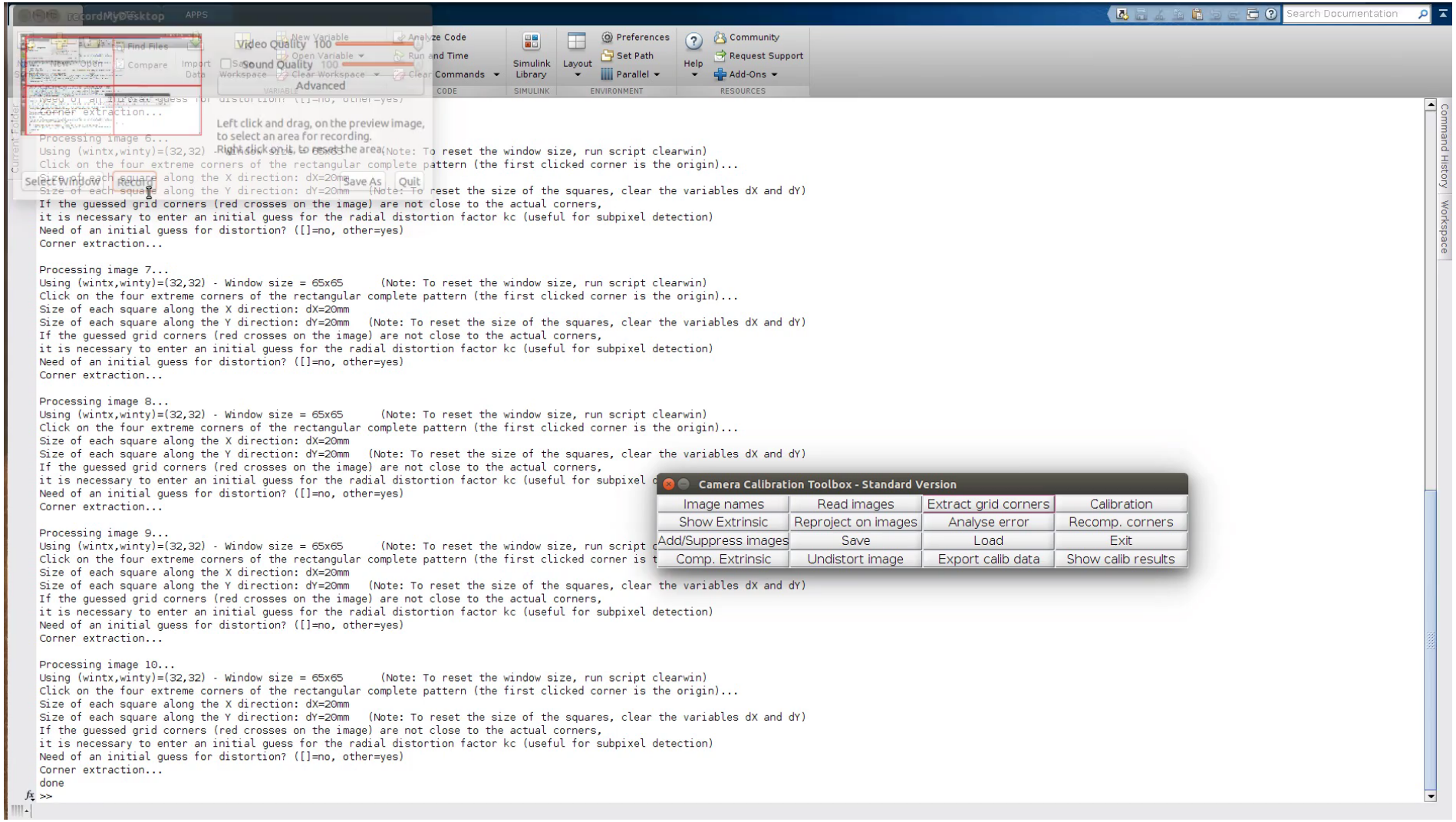
How we calibrate in Matlab?



Extracting corners of the checkerboard



Extrinsic parameter results (R,T)



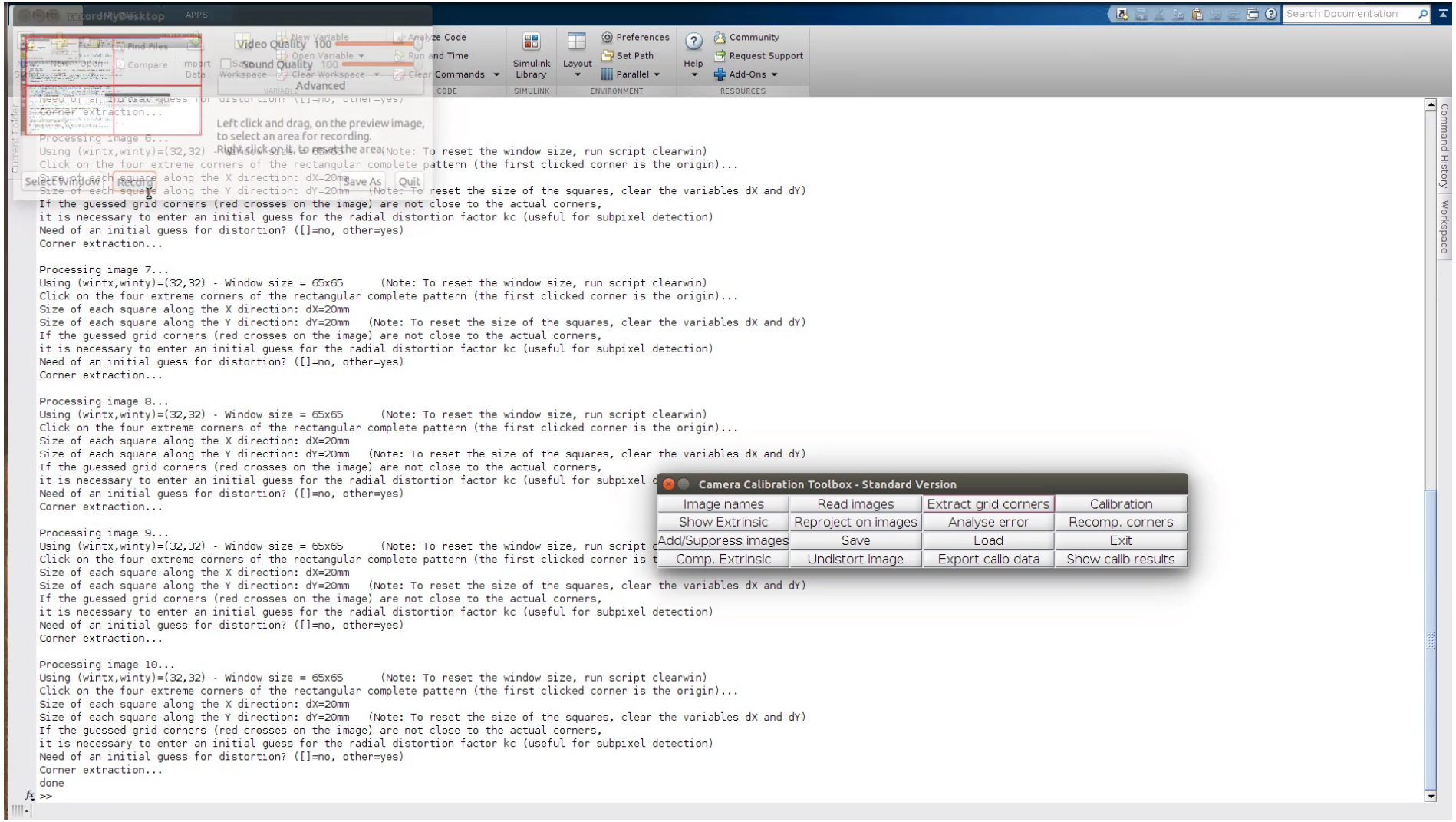
Extrinsic parameter results (R,T)

```
Calibration results after optimization (with uncertainties):

Focal Length:          fc = [ 3468.99207   3458.27097 ] +/- [ 10.59701   10.40724 ]
Principal point:       cc = [ 2029.63006   1511.12039 ] +/- [ 7.55158   7.25687 ]
Skew:             alpha_c = [ 0.00000 ] +/- [ 0.00000  ]   => angle of pixel axes = 90.00000 +/- 0.00000 degrees
Distortion:            kc = [ 0.07878   -0.07501   -0.00072   0.00094  0.00000 ] +/- [ 0.00520   0.01581   0.00073   0.00085  0.00000 ]
Pixel error:          err = [ 0.54275   0.61021 ]

Note: The numerical errors are approximately three times the standard deviations (for reference).
```

Extrinsic parameter results (R,T)



Reproject undistorted coordinates