Unit 3

Camera Calibration

Ref: Szeliski, Sec. 6.2, 6.3, 7.1, 7.2

Outline

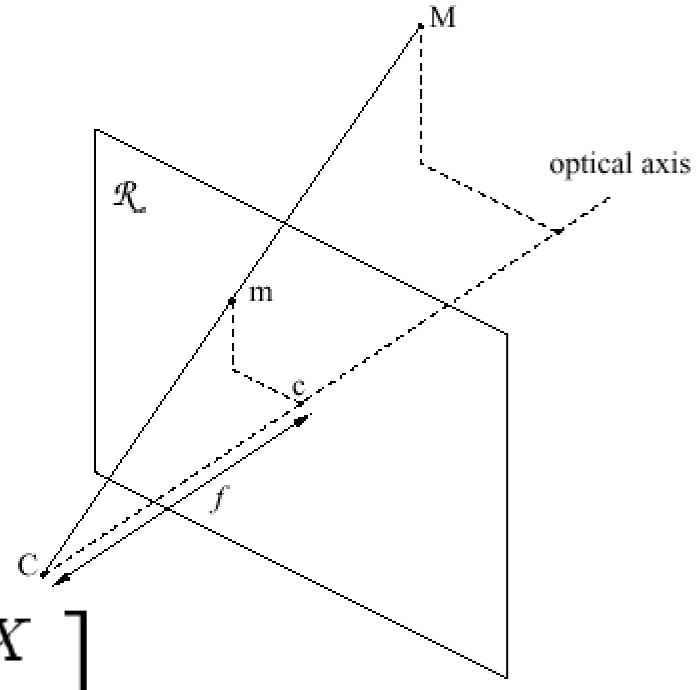
- Geometric camera projection model
- Camera calibration
- Plane projective transformation
- Vanishing points
- Cross-ratio – projective invariant

Camera Model

- Simple Model: Pinhole
- Image plane at Z=1 (f=1)
- Projection process:

$$x = \frac{X}{Z} \quad y = \frac{Y}{Z}$$

- Homogeneous Coord, representation



$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

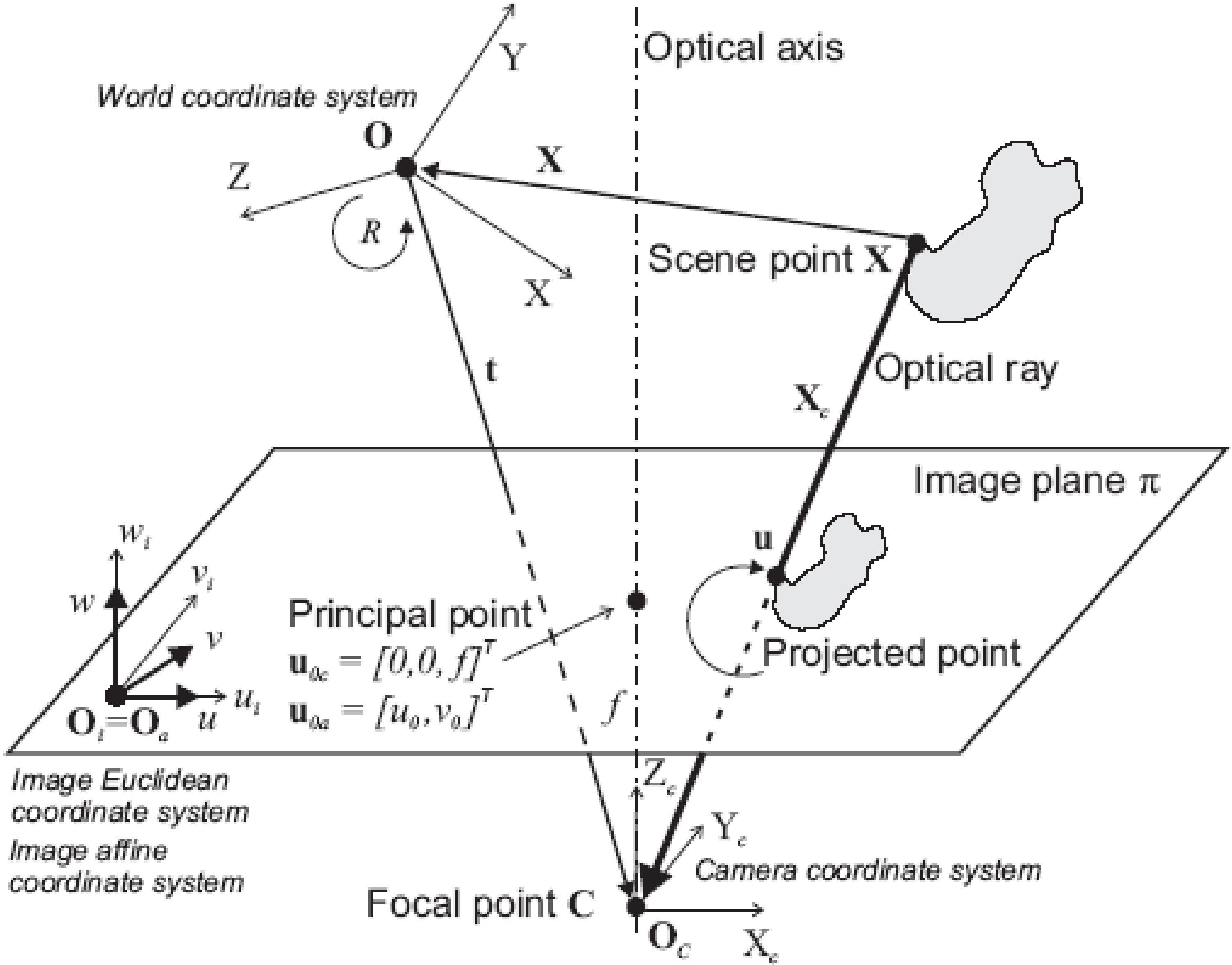
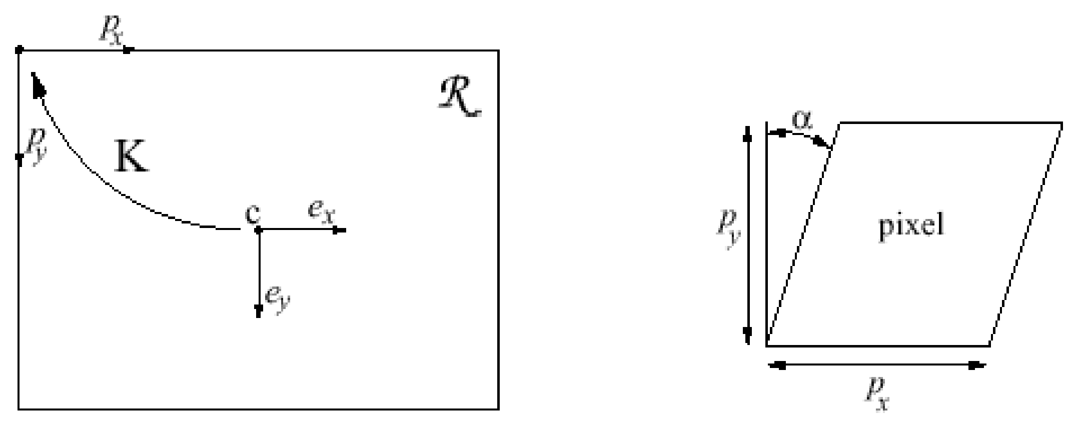


Figure 11.7: The geometry of a linear perspective camera.

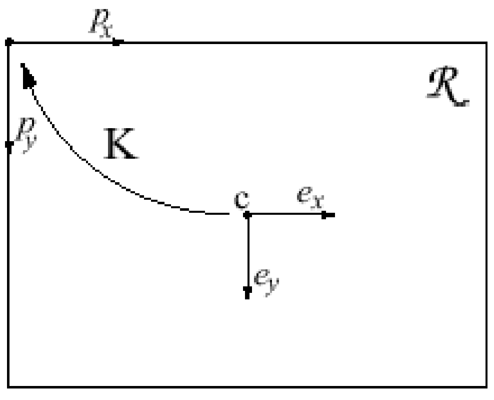
Intrinsic Parameters

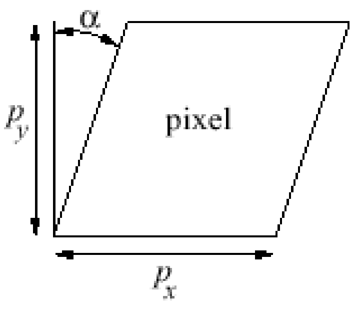
- Perspective projection, focal length f.
- Camera frame (pixel coordinates, pixel size and shape).
- Geometric distortion introduced by optics.



From retinal coordinates to image coordinates

Intrinsic Parameters ctd.





From retinal coordinates to image coordinates

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{f}{p_x} & (\tan\alpha)\frac{f}{p_y} & c_x \\ & \frac{f}{p_y} & c_y \\ & & 1 \end{bmatrix} \begin{bmatrix} x_{\mathcal{R}} \\ y_{\mathcal{R}} \\ 1 \end{bmatrix}$$

- f: focal length
- p_x,p_y width and height of pixels
- $[cx,cy]^T$ principal point (retinal coordinates)
- α: skew angle

Simplified Notation: Calibration Matrix of Camera

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{f}{p_x} & (\tan\alpha)\frac{f}{p_y} & c_x \\ & \frac{f}{p_y} & c_y \\ & & 1 \end{bmatrix} \begin{bmatrix} x_{\mathcal{R}} \\ y_{\mathcal{R}} \\ 1 \end{bmatrix}$$

Simplified notation

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & s & c_x \\ & f_y & c_y \\ & & 1 \end{bmatrix} \begin{bmatrix} x_{\mathcal{R}} \\ y_{\mathcal{R}} \\ 1 \end{bmatrix}$$

$$\boxed{\mathrm{m} = \mathbf{K}\mathrm{m}_{\mathcal{R}}}$$

Extrinsic Parameters: Transformation of Scene Points

$$\mathtt{M}' = \begin{bmatrix} \mathbf{R} & \mathtt{t} \\ 0_3^\top & 1 \end{bmatrix} \mathtt{M}$$

R: Rotation Matrix (3x3)

t: translation vector $[t_x, t_y, t_z]^T$

Combined Projection Matrix

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \underbrace{\begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}}_{\text{Intrinsic K}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{projection}} \underbrace{\begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0_3^T & 1 \end{bmatrix}}_{\text{extrinsic}} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$\mathbf{m} \sim \mathbf{K}\,[\mathbf{R}\,|\,\mathbf{t}]\;\mathbf{M}$$

$$\mathbf{m} \sim \mathbf{P}\;\mathbf{M}$$

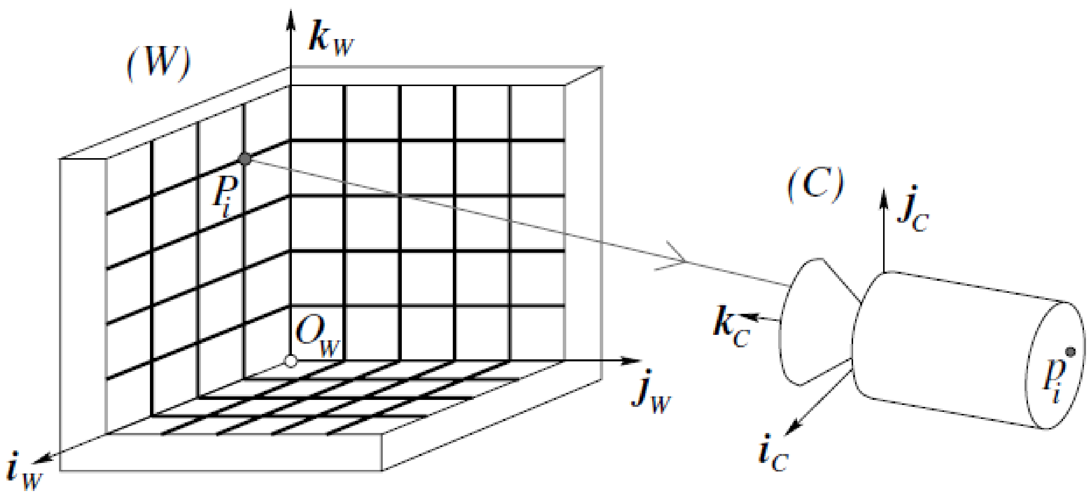
P: 3x4 matrix, camera projection matrix.

Projection Matrix: #Parameters

- Intrinsic: 5 (f_x, f_y, c_x, c_y, {s})
- Extrinsic: 6 (R, T)
- Total: 10-11 DOF
- Simplification often used for initialization:
  – (c_x, c_y) $\approx$ center of image
  – s $\approx$ 0 (rectangular pixels)
- Attention: Intrinsic parameters fixed for fixed optics camera $\neq$ not true for zoom lens!

The Calibration Problem

- Given
  - Calibration pattern with *N* corners
  - *M* views of this calibration pattern
- Recover the intrinsic and extrinsic parameters
  - Sometimes, we are only interested in calibrating intrinsic or extrinsic parameters.

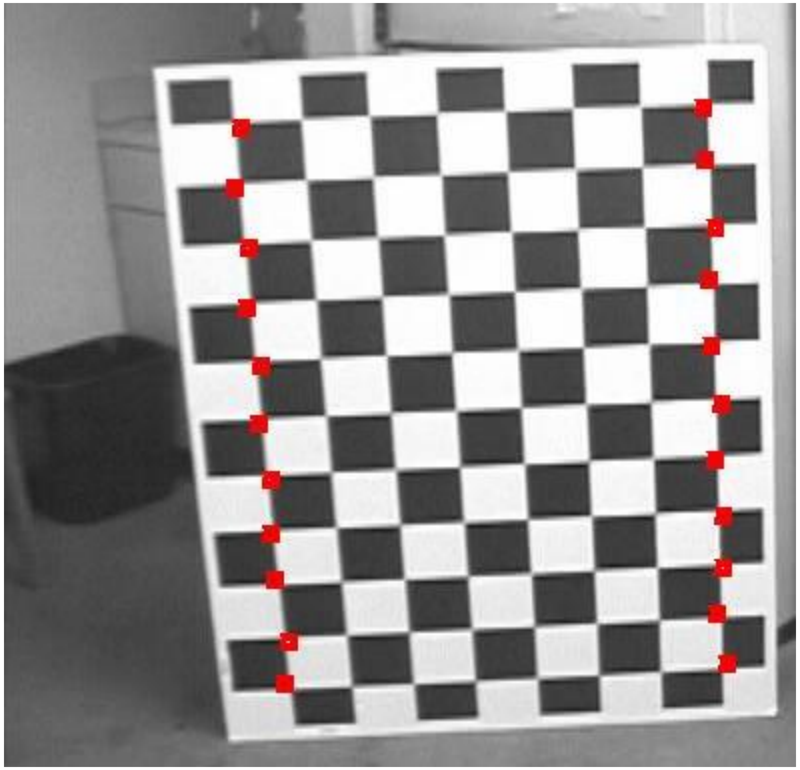


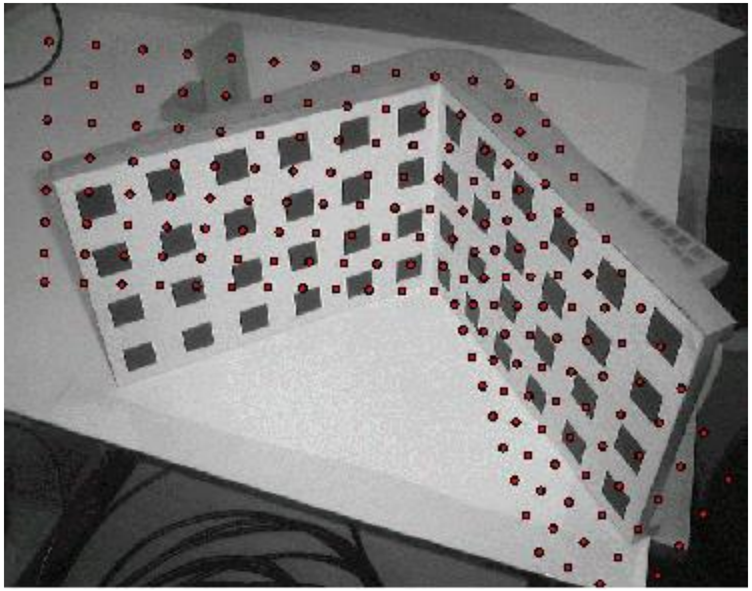
Camera Calibration

- Issues:
  - what are intrinsic parameters of the camera?
  - what is the camera matrix? (intrinsic+extrinsic)
- General strategy:
  - view calibration object
  - identify image points
  - obtain camera matrix by minimizing error
  - obtain intrinsic parameters from camera matrix

- Error minimization:
  - Linear least squares
    - easy problem numerically
    - solution can be rather bad
  - Minimize image distance
    - more difficult numerical problem
    - solution usually rather good,
    - start with linear least squares

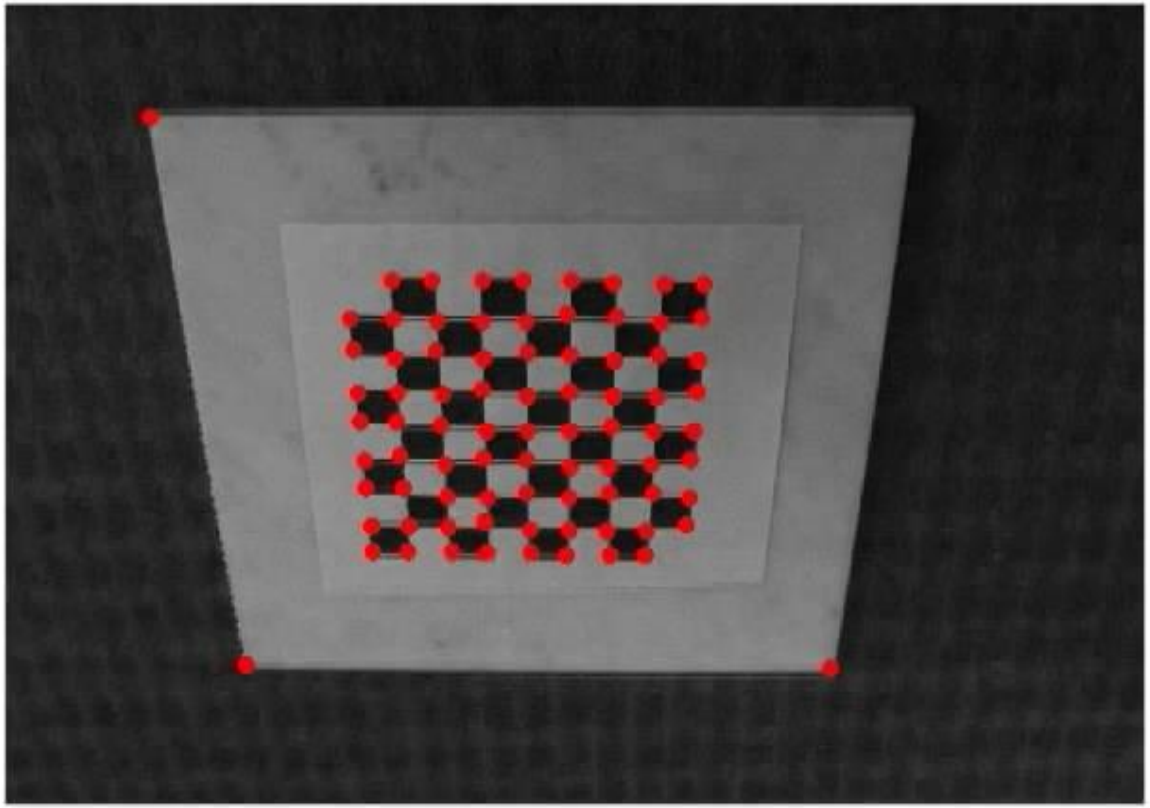
Example Calibration Pattern





Calibration Pattern: Object with features of known size/geometry

Harris Corner Detector



Camera Calibration (DLT)

Problem Statement:

Given n correspondences $\mathbf{x}_i \leftrightarrow \mathbf{X}_i$, where $\mathbf{X}_i$ is a scene point and $\mathbf{x}_i$ its image:

Compute

$\mathtt{P} = \mathtt{K}\left[\mathtt{R} \middle| \, \mathbf{t}\right]$ such that $\mathbf{x}_i = \mathtt{P}\mathbf{X}_i$.

The algorithm for camera calibration has two parts:

(i) Compute the matrix $\mathtt{P}$ from a set of point correspondences.

(ii) Decompose $\mathtt{P}$ into $\mathtt{K}, \mathtt{R}$ and $\mathbf{t}$ via the $\mathtt{QR}$ decomposition.

Algorithm step 1: Compute the matrix P (DLT)

$$\mathbf{x}_i = \mathrm{P}\mathbf{X}_i.$$

Each correspondence generates two equations

$$x_i = \frac{p_{11}X_i + p_{12}Y_i + p_{13}Z_i + p_{14}}{p_{31}X_i + p_{32}Y_i + p_{33}Z_i + p_{34}} \qquad y_i = \frac{p_{21}X_i + p_{22}Y_i + p_{23}Z_i + p_{24}}{p_{31}X_i + p_{32}Y_i + p_{33}Z_i + p_{34}}$$

Multiplying out gives equations linear in the matrix elements of P

$$x_i(p_{31}X_i + p_{32}Y_i + p_{33}Z_i + p_{34}) = p_{11}X_i + p_{12}Y_i + p_{13}Z_i + p_{14}$$
$$y_i(p_{31}X_i + p_{32}Y_i + p_{33}Z_i + p_{34}) = p_{21}X_i + p_{22}Y_i + p_{23}Z_i + p_{24}$$

These equations can be rearranged as

$$\begin{bmatrix} X & Y & Z & 1 & 0 & 0 & 0 & 0 & -xX & -xY & -xZ & -x \\ 0 & 0 & 0 & 0 & X & Y & Z & 1 & -yX & -yY & -yZ & -y \end{bmatrix} \mathbf{p} = \mathbf{0}$$

with $\mathbf{p} = (p_{11}, p_{12}, p_{13}, p_{14}, p_{21}, p_{22}, p_{23}, p_{24}, p_{31}, p_{32}, p_{33}, p_{34})^\top$ a 12-vector.

Algorithm step 1 continued

Solving for $\mathtt{P}$

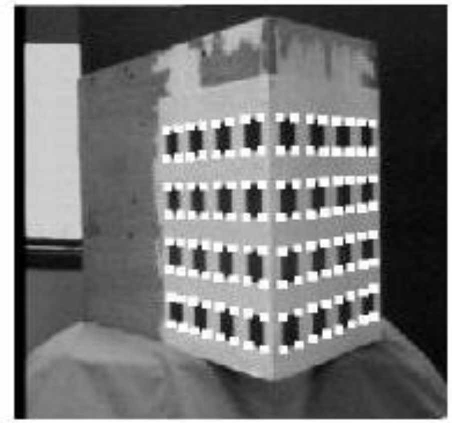
(i) Concatenate the equations from ($n \geq 6$) correspondences to generate $2n$ simultaneous equations, which can be written: $\mathtt{A}\mathbf{p} = \mathbf{0}$, where $\mathtt{A}$ is a $2n \times 12$ matrix.

(ii) In general this will not have an exact solution, but a (linear) solution which minimises $\|\mathtt{A}\mathbf{p}\|$, subject to $\|\mathbf{p}\| = 1$ is obtained from the eigenvector with least eigenvalue of $\mathtt{A}^\top\mathtt{A}$. Or equivalently from the vector corresponding to the smallest singular value of the SVD of $\mathtt{A}$.

(iii) This linear solution is then used as the starting point for a non-linear minimisation of the difference between the measured and projected point:

$$\min_{\mathtt{P}} \sum_i \left((x_i, y_i) - P(\mathrm{X}_i, \mathrm{Y}_i, \mathrm{Z}_i, 1)\right)^2$$

Example – Calibration Object

Determine accurate corner positions by

(i) Extract and link edges using Canny edge operator.

(ii) Fit lines to edges using orthogonal regression.

(iii) Intersect lines to obtain corners to sub-pixel accuracy.

The final error between measured and projected points is typically less than 0.02 pixels.

Algorithm step 2: Decompose P into K,R and t

The first 3×3 submatrix, $\mathtt{M}$, of $\mathtt{P}$ is the product ($\mathtt{M} = \mathtt{KR}$) of an upper triangular and rotation matrix.

(i) Factor $\mathtt{M}$ into $\mathtt{KR}$ using the $\mathtt{QR}$ matrix decomposition. This determines $\mathtt{K}$ and $\mathtt{R}$.

(ii) Then

$$\mathbf{t} = \mathrm{K}^{-1}(p_{14}, p_{24}, p_{34})^{\top}$$

Note, this produces a matrix with an extra skew parameter s

$$\mathrm{K} = \begin{bmatrix} \alpha_x & s & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix}$$

with $s = \tan\theta$, and θ the angle between the image axes.

Another Solution

- Camera projection matrix P = [A b], R=

$$\rho(\mathcal{A} \quad b) = \mathcal{K}(\mathcal{R} \quad t) \Longleftrightarrow \rho \begin{pmatrix} a_1^T \\ a_2^T \\ a_3^T \end{pmatrix} = \begin{pmatrix} \alpha r_1^T - \alpha \cot\theta r_2^T + x_0 r_3^T \\ \dfrac{\beta}{\sin\theta} r_2^T + y_0 r_3^T \\ r_3^T \end{pmatrix}$$

$$\mathcal{K} = \begin{pmatrix} \mathcal{K}_2 & p_0 \\ 0^T & 1 \end{pmatrix}, \quad \text{where} \quad \mathcal{K}_2 \stackrel{\text{def}}{=} \begin{pmatrix} \alpha & -\alpha\cot\theta \\ 0 & \dfrac{\beta}{\sin\theta} \end{pmatrix} \quad \text{and} \quad p_0 \stackrel{\text{def}}{=} \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$$

- Using the fact that the rows of a rotation matrix have unit length and are perpendicular to each other yields

$$\begin{cases} \rho = \varepsilon / ||a_3||, \\ r_3 = \rho a_3, \\ x_0 = \rho^2 (a_1 \cdot a_3), \\ y_0 = \rho^2 (a_2 \cdot a_3), \end{cases} \qquad \text{where } \varepsilon = \mp 1.$$

$$\begin{cases} \rho^2(a_1 \times a_3) = -\alpha r_2 - \alpha \cot\theta r_1, \\ \rho^2(a_2 \times a_3) = \dfrac{\beta}{\sin\theta} r_1, \end{cases} \quad \text{and} \quad \begin{cases} \rho^2 \|a_1 \times a_3\| = \dfrac{|\alpha|}{\sin\theta}, \\ \rho^2 \|a_2 \times a_3\| = \dfrac{|\beta|}{\sin\theta}, \end{cases}$$

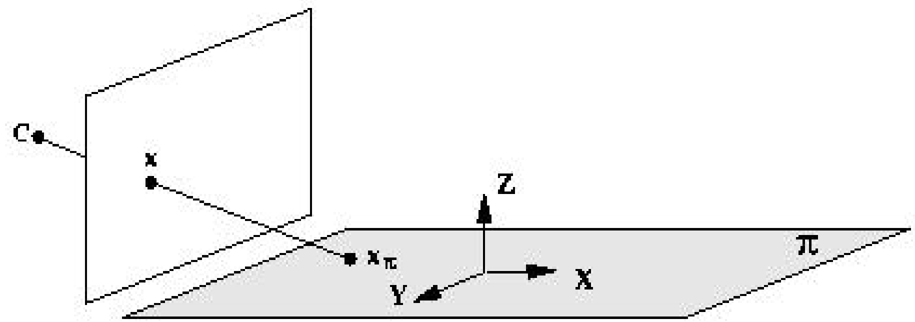
thus:

$$\begin{cases} \cos\theta = -\dfrac{(a_1 \times a_3)\cdot(a_2 \times a_3)}{\|a_1 \times a_3\| \, \|a_2 \times a_3\|}, \\ \alpha = \rho^2 \|a_1 \times a_3\| \sin\theta, \\ \beta = \rho^2 \|a_2 \times a_3\| \sin\theta, \end{cases}$$

Finally, we have

$$\begin{cases} r_1 = \dfrac{\rho^2 \sin\theta}{\beta}(a_2 \times a_3) = \dfrac{1}{\|a_2 \times a_3\|}(a_2 \times a_3) \\ r_2 = r_3 \times r_1. \end{cases}$$

Plane projective transformations



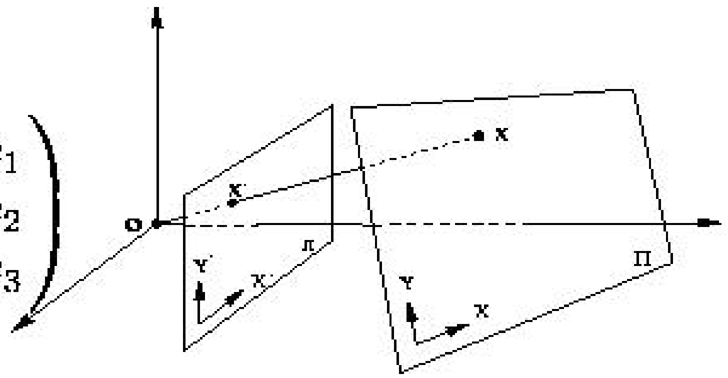
Choose the world coordinate system such that the plane of the points has zero Z coordinate. Then the 3×4 matrix P reduces to

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{pmatrix} X \\ Y \\ 0 \\ 1 \end{pmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{14} \\ p_{21} & p_{22} & p_{24} \\ p_{31} & p_{32} & p_{34} \end{bmatrix} \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}$$

which is a 3×3 matrix representing a general plane to plane projective transformation.

Projective transformations continued

$$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$



or $\mathbf{x}' = \mathtt{H}\mathbf{x}$, where $\mathtt{H}$ is a 3×3 non-singular homogeneous matrix.

- This is the most general transformation between the world and image plane under imaging by a perspective camera.
- It is often only the 3×3 form of the matrix that is important in establishing properties of this transformation.
- A projective transformation is also called a "homography" and a "collineation".
- $\mathtt{H}$ has 8 degrees of freedom.

Four points define a projective transformation

Given n point correspondences $(x, y) \leftrightarrow (x', y')$

Compute $\mathtt{H}$ such that $\mathbf{x}'_i = \mathtt{H}\mathbf{x}_i$

- Each point correspondence gives two constraints

$$x' = \frac{x'_1}{x'_3} = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}}, \qquad y' = \frac{x'_2}{x'_3} = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}}$$

and multiplying out generates two linear equations for the elements of $\mathtt{H}$

$$x'\left(h_{31}x + h_{32}y + h_{33}\right) = h_{11}x + h_{12}y + h_{13}$$
$$y'\left(h_{31}x + h_{32}y + h_{33}\right) = h_{21}x + h_{22}y + h_{23}$$

- If $n \geq 4$ (no three points collinear), then $\mathtt{H}$ is determined uniquely.
- The converse of this is that it is possible to transform any four points in general position to any other four points in general position by a projectivity.

Example 1: Removing Perspective Distortion

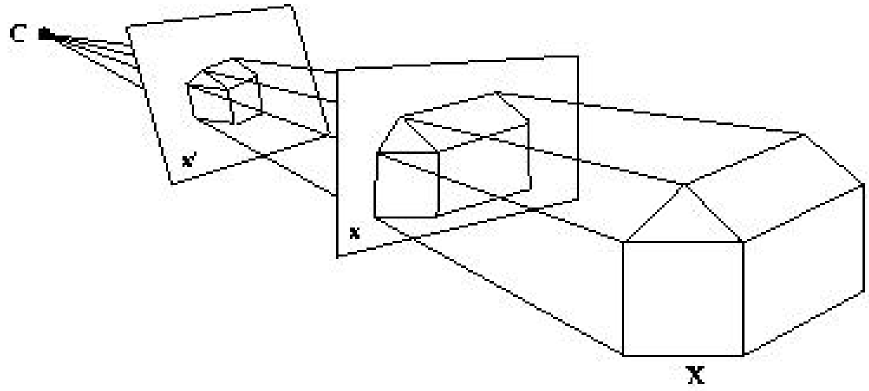
Given: the coordinates of four points on the scene plane

Find: a projective rectification of the plane

- This rectification does not require knowledge of any of the camera's parameters or the pose of the plane.
- It is not always necessary to know coordinates for four points.

The Cone of Rays

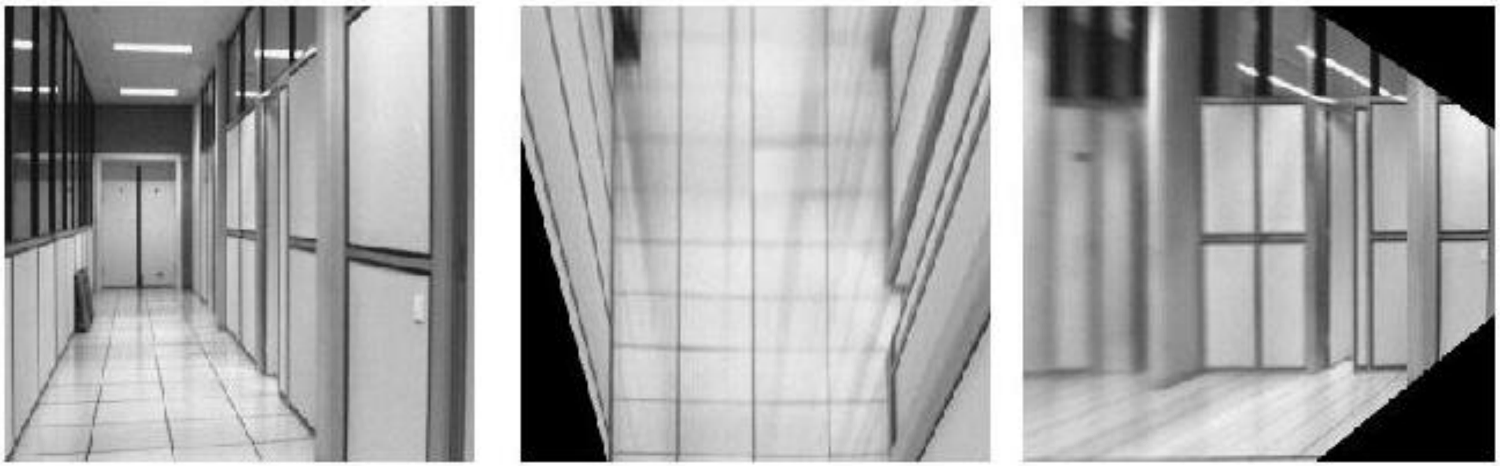
An image is the intersection of a plane with the cone of rays between points in 3-space and the optical centre. Any two such “images” (with the same camera centre) are related by a planar projective transformation.



$$\mathbf{x}' = \mathrm{H}\mathbf{x}$$

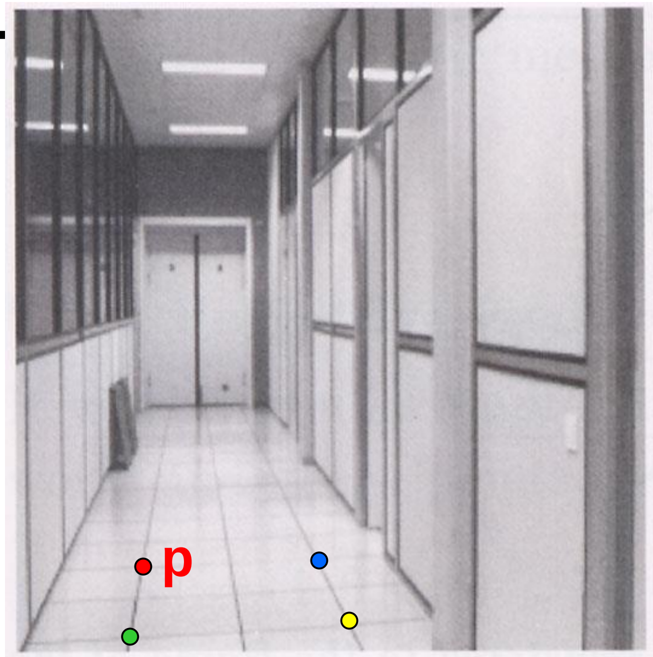
e.g. rotation about the camera centre

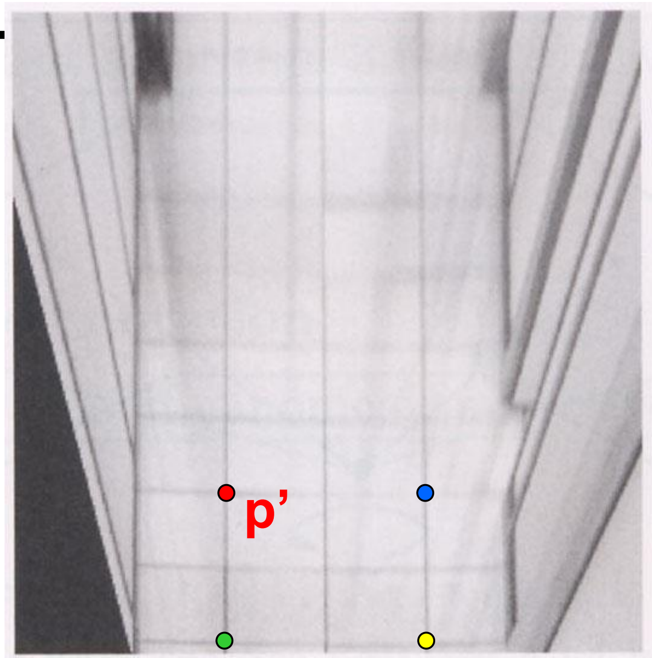
Example 2: Synthetic Rotations



The synthetic images are produced by projectively warping the original image so that four corners of an imaged rectangle map to the corners of a rectangle. Both warpings correspond to a synthetic rotation of the camera about the (fixed) camera centre.

Image rectification





To unwarp (rectify) an image

- solve for homography **H** given **p** and **p’**
- solve equations of the form: w**p’** = **Hp**
  - linear in unknowns: w and coefficients of **H**
  - H is defined up to an arbitrary scale factor
  - how many points are necessary to solve for **H**?

Solving for homographies

$$\begin{bmatrix} x'_i \\ y'_i \\ 1 \end{bmatrix} \cong \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}$$

$$\begin{aligned} x'_i &= \frac{h_{00}x_i + h_{01}y_i + h_{02}}{h_{20}x_i + h_{21}y_i + h_{22}} \\ y'_i &= \frac{h_{10}x_i + h_{11}y_i + h_{12}}{h_{20}x_i + h_{21}y_i + h_{22}} \end{aligned}$$

$$\begin{aligned} x'_i(h_{20}x_i + h_{21}y_i + h_{22}) &= h_{00}x_i + h_{01}y_i + h_{02} \\ y'_i(h_{20}x_i + h_{21}y_i + h_{22}) &= h_{10}x_i + h_{11}y_i + h_{12} \end{aligned}$$

$$\begin{bmatrix} x_i & y_i & 1 & 0 & 0 & 0 & -x'_i x_i & -x'_i y_i & -x'_i \\ 0 & 0 & 0 & x_i & y_i & 1 & -y'_i x_i & -y'_i y_i & -y'_i \end{bmatrix} \begin{bmatrix} h_{00} \\ h_{01} \\ h_{02} \\ h_{10} \\ h_{11} \\ h_{12} \\ h_{20} \\ h_{21} \\ h_{22} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Solving for homographies

$$
\begin{bmatrix}
x_1 & y_1 & 1 & 0 & 0 & 0 & -x'_1 x_1 & -x'_1 y_1 & -x'_1 \\
0 & 0 & 0 & x_1 & y_1 & 1 & -y'_1 x_1 & -y'_1 y_1 & -y'_1 \\
 & & & & \vdots & & & & \\
x_n & y_n & 1 & 0 & 0 & 0 & -x'_n x_n & -x'_n y_n & -x'_n \\
0 & 0 & 0 & x_n & y_n & 1 & -y'_n x_n & -y'_n y_n & -y'_n
\end{bmatrix}
\begin{bmatrix} h_{00} \\ h_{01} \\ h_{02} \\ h_{10} \\ h_{11} \\ h_{12} \\ h_{20} \\ h_{21} \\ h_{22} \end{bmatrix}
=
\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}
$$

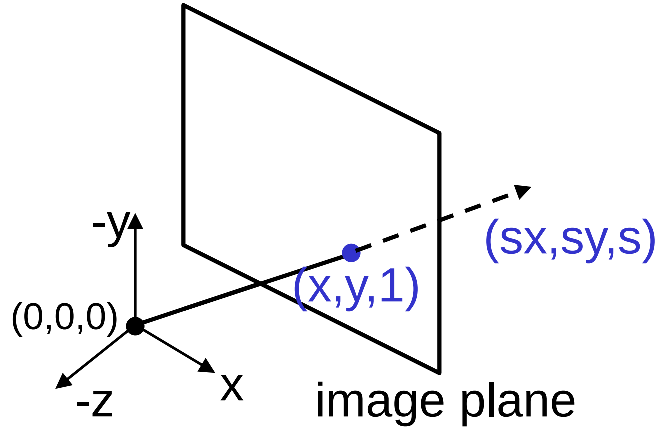
$$\underset{2n \times 9}{\mathbf{A}} \qquad \underset{9}{\mathbf{h}} \qquad \underset{2n}{\mathbf{0}}$$

Defines a least squares problem: $\text{minimize } \|\mathbf{A}\mathbf{h} - \mathbf{0}\|^2$

- Since **h** is only defined up to scale, solve for unit vector $\hat{\mathbf{h}}$
- Solution: $\hat{\mathbf{h}}$ = eigenvector of $\mathbf{A}^T\mathbf{A}$ with smallest eigenvalue
- Works with 4 or more points

The projective plane

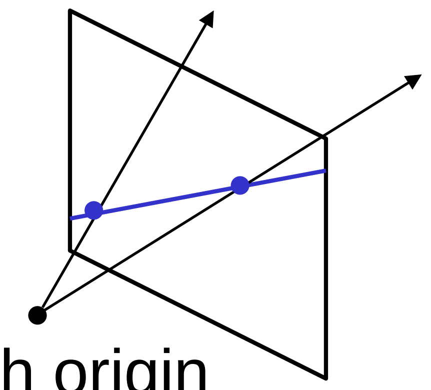
- Why do we need homogeneous coordinates?
  - represent points at infinity, homographies, perspective projection, multi-view relationships
- What is the geometric intuition?
  - a point in the image is a *ray* in projective space



- 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)

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

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

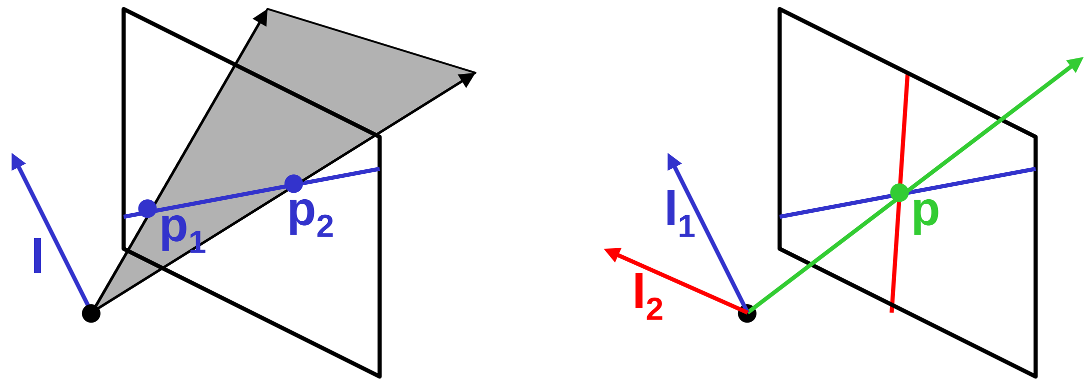
$$\mathbf{l} \qquad \mathbf{p}$$

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

Point and line duality

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

– It is $\perp$ 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 $\perp$ to $\mathbf{p_1}$ and $\mathbf{p_2}$ $\Rightarrow$ $\mathbf{l} = \mathbf{p_1} \times \mathbf{p_2}$
- **l** is the plane normal

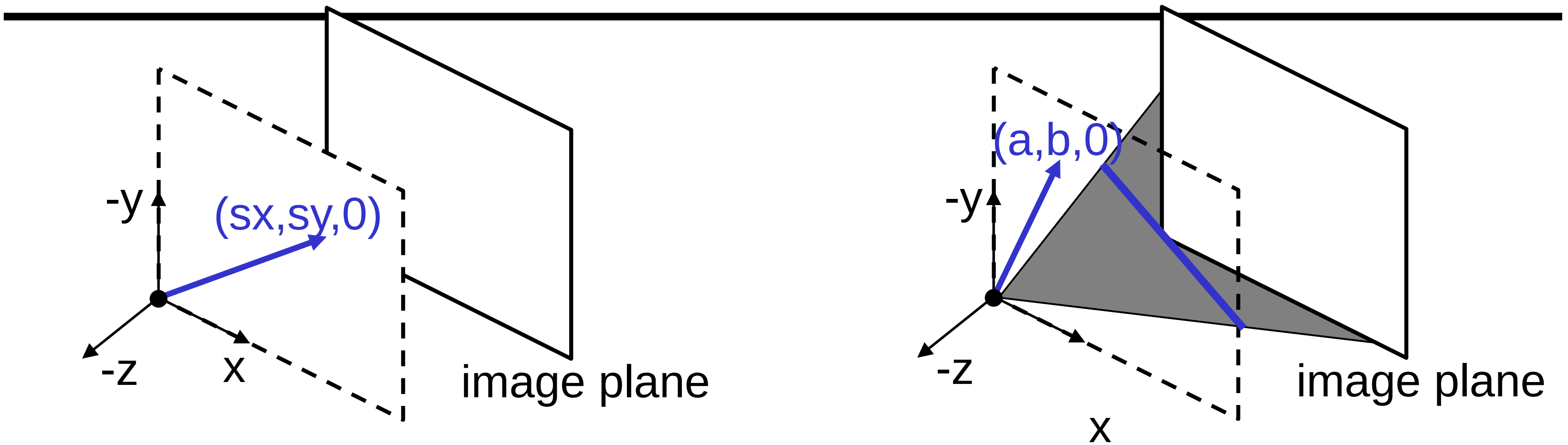
What is the intersection of two lines $\mathbf{l_1}$ and $\mathbf{l_2}$?

- **p** is $\perp$ 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

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)

Homographies of points and lines

- Computed by 3x3 matrix multiplication
  - To transform a point: $\mathbf{p'} = \mathbf{Hp}$
  - To transform a line: $\mathbf{lp}=0 \rightarrow \mathbf{l'p'}=0$
  - $0 = \mathbf{lp} = \mathbf{lH^{-1}Hp} = \mathbf{lH^{-1}p'} \Rightarrow \mathbf{l' = lH^{-1}}$
  - lines are transformed by post-multiplication of $\mathbf{H^{-1}}$

3D projective geometry

- These concepts generalize naturally to 3D
  - Homogeneous coordinates
    - Projective 3D points have four coords: **P** = (X,Y,Z,W)
  - Duality
    - A plane **N** is also represented by a 4-vector
    - Points and planes are dual in 3D: **N P**=0
  - Projective transformations
    - Represented by 4x4 matrices **T**: **P' = TP,** **N' = N** $\mathbf{T}^{-1}$

3D to 2D: “perspective” projection

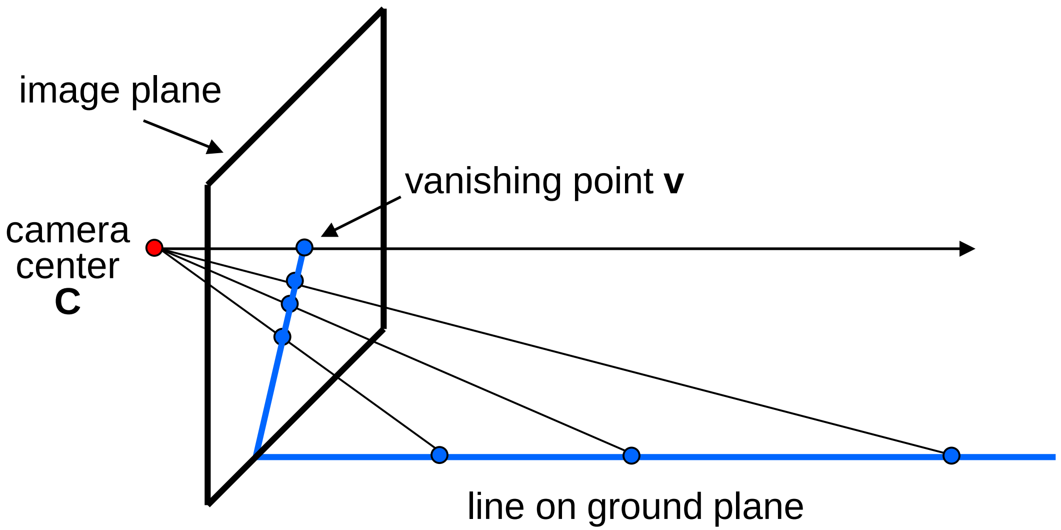
- Matrix Projection:

$$\mathbf{p} = \begin{bmatrix} wx \\ wy \\ w \end{bmatrix} = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \mathbf{\Pi P}$$

What is *not* preserved under perspective projection?

What is preserved?

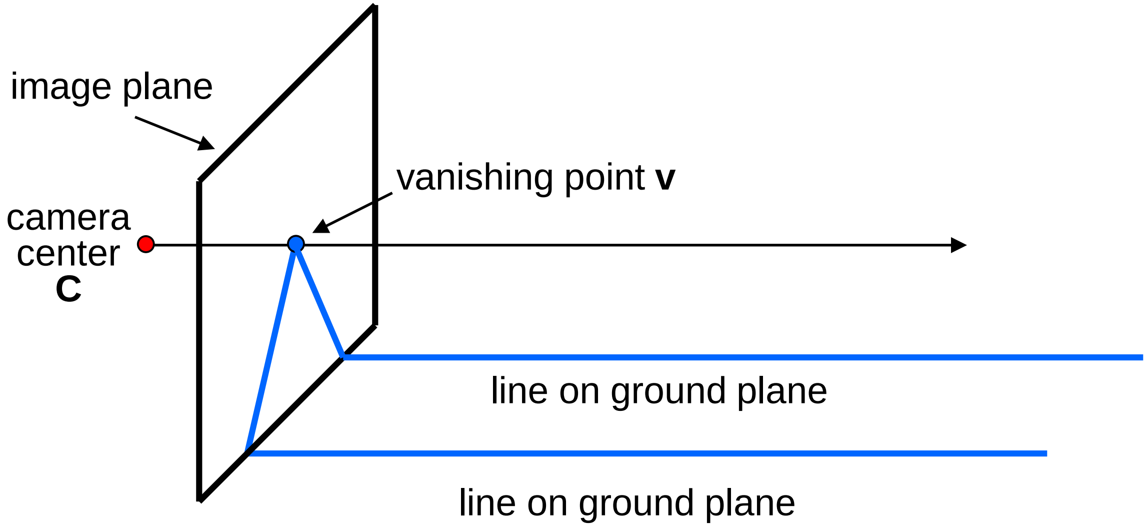
Vanishing points



Vanishing point

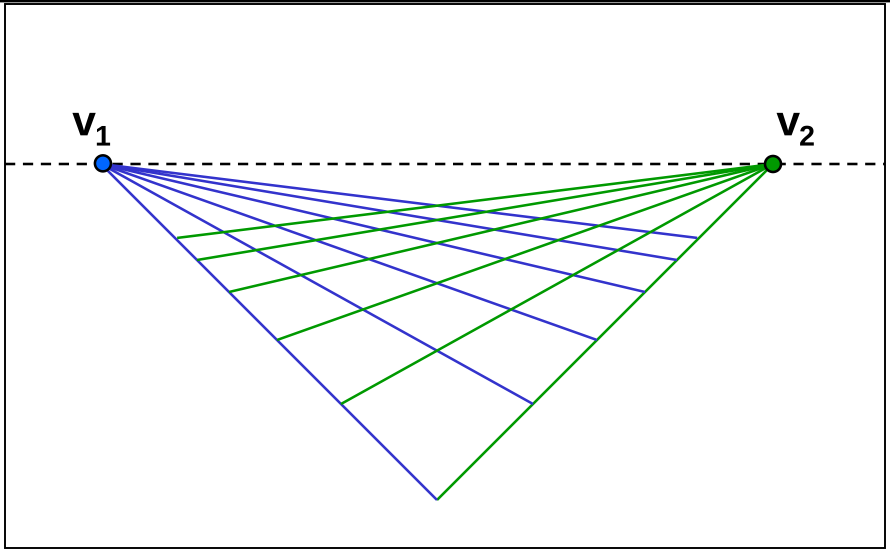
- projection of a point at infinity

Vanishing points



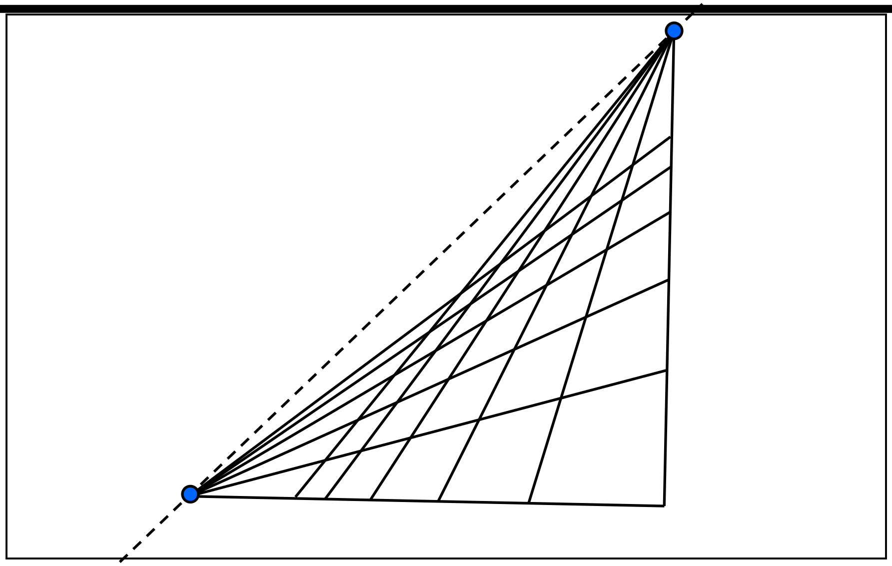
- Properties
  - Any two parallel lines have the same vanishing point **v**
  - The ray from **C** through **v** is parallel to the lines
  - An image may have more than one vanishing point
    - in fact every pixel is a potential vanishing point

Vanishing lines



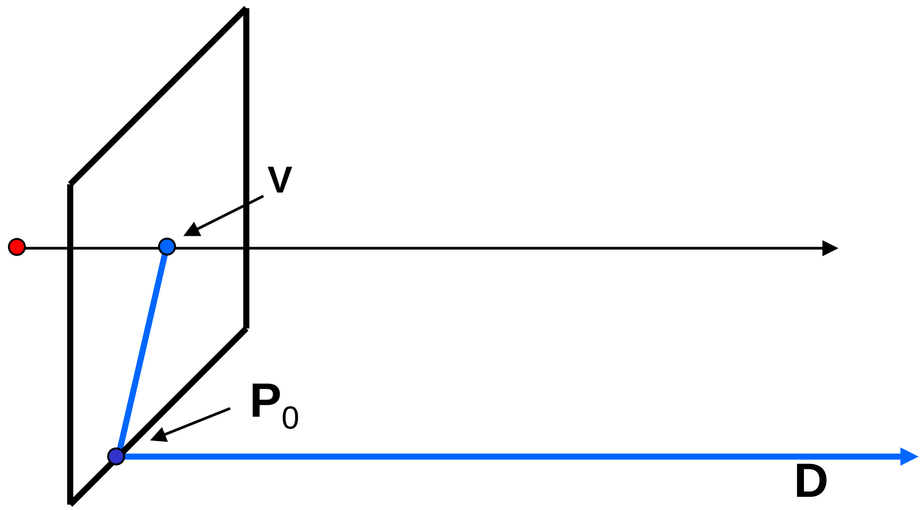
- Multiple Vanishing Points
  - Any set of parallel lines on the plane define a vanishing point
  - The union of all of vanishing points from lines on the same plane is the *vanishing line*
    - For the ground plane, this is called the *horizon*

Vanishing lines



- Multiple Vanishing Points
  - Different planes define different vanishing lines

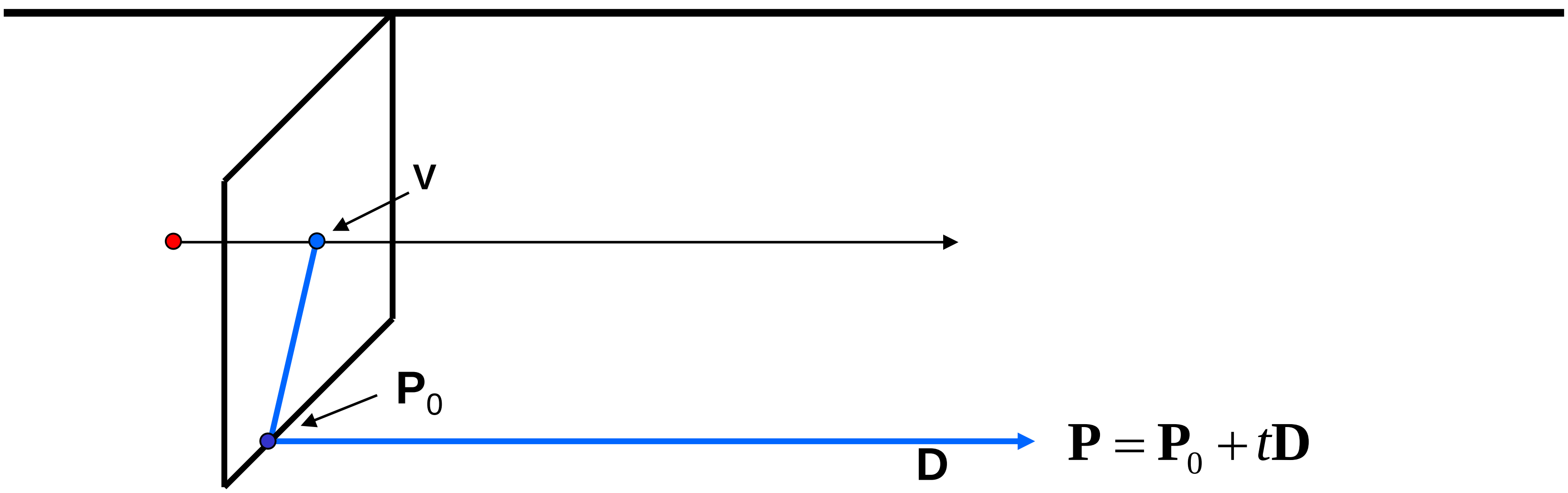
Computing vanishing points



$$\mathbf{P} = \mathbf{P}_0 + t\mathbf{D}$$

$$\mathbf{P}_t = \begin{bmatrix} P_X + tD_X \\ P_Y + tD_Y \\ P_Z + tD_Z \\ 1 \end{bmatrix}$$

Computing vanishing points

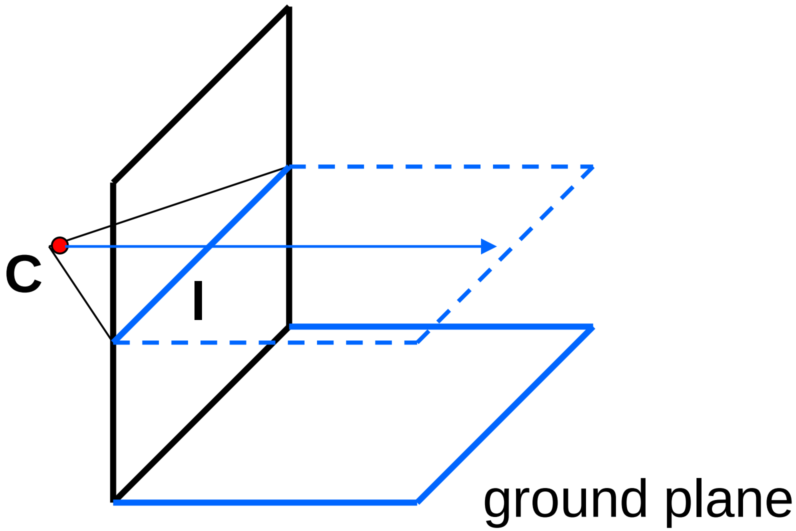


$$\mathbf{P} = \mathbf{P}_0 + t\mathbf{D}$$

$$\mathbf{P}_t = \begin{bmatrix} P_X + tD_X \\ P_Y + tD_Y \\ P_Z + tD_Z \\ 1 \end{bmatrix} \cong \begin{bmatrix} P_X / t + D_X \\ P_Y / t + D_Y \\ P_Z / t + D_Z \\ 1/t \end{bmatrix} \qquad t \to \infty \qquad \mathbf{P}_\infty \cong \begin{bmatrix} D_X \\ D_Y \\ D_Z \\ 0 \end{bmatrix} \qquad \mathbf{v} = \mathbf{\Pi P}_\infty$$

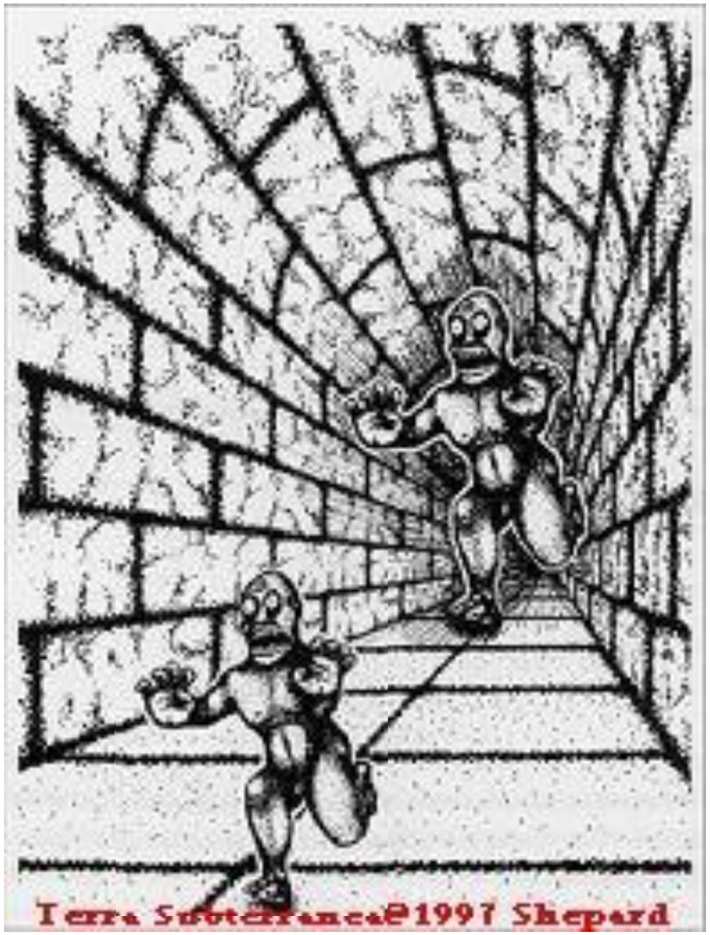
- Properties
  - $\mathbf{P}_\infty$ is a point at *infinity*, **v** is its projection
  - They depend only on line *direction*
  - Parallel lines $\mathbf{P}_0$ + t**D**, $\mathbf{P}_1$ + t**D** intersect at $\mathbf{P}_\infty$

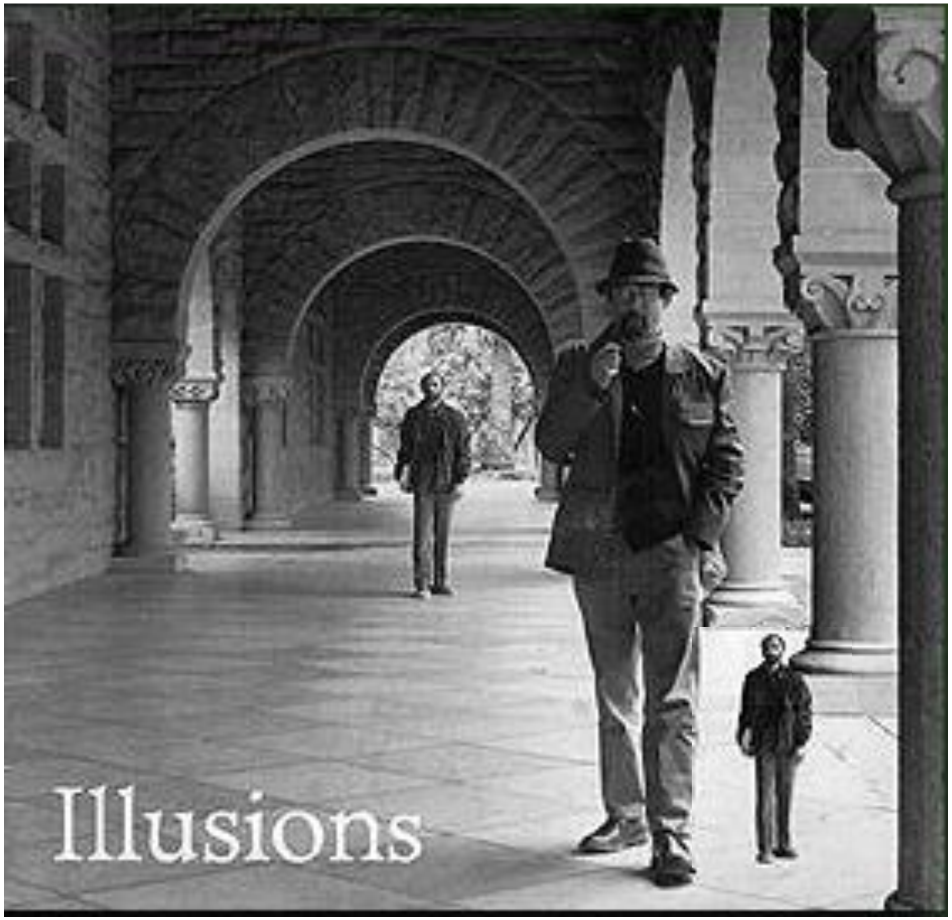
Computing the horizon



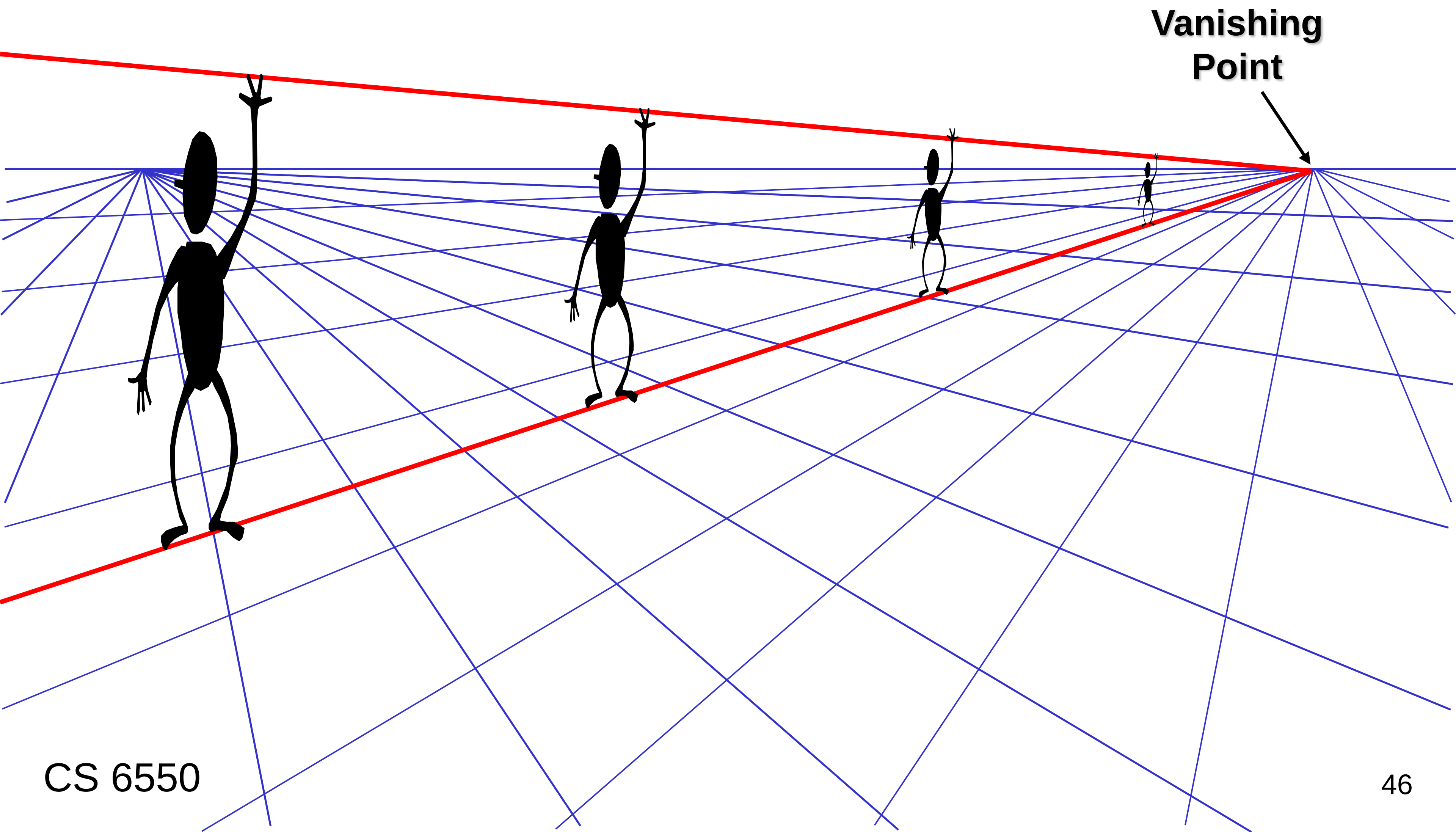
- Properties
  - **I** is intersection of horizontal plane through **C** with image plane
  - Compute **I** from two sets of parallel lines on ground plane
  - All points at same height as **C** project to **I**
    - points higher than **C** project above **I**
  - Provides way of comparing height of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

Fun with vanishing points

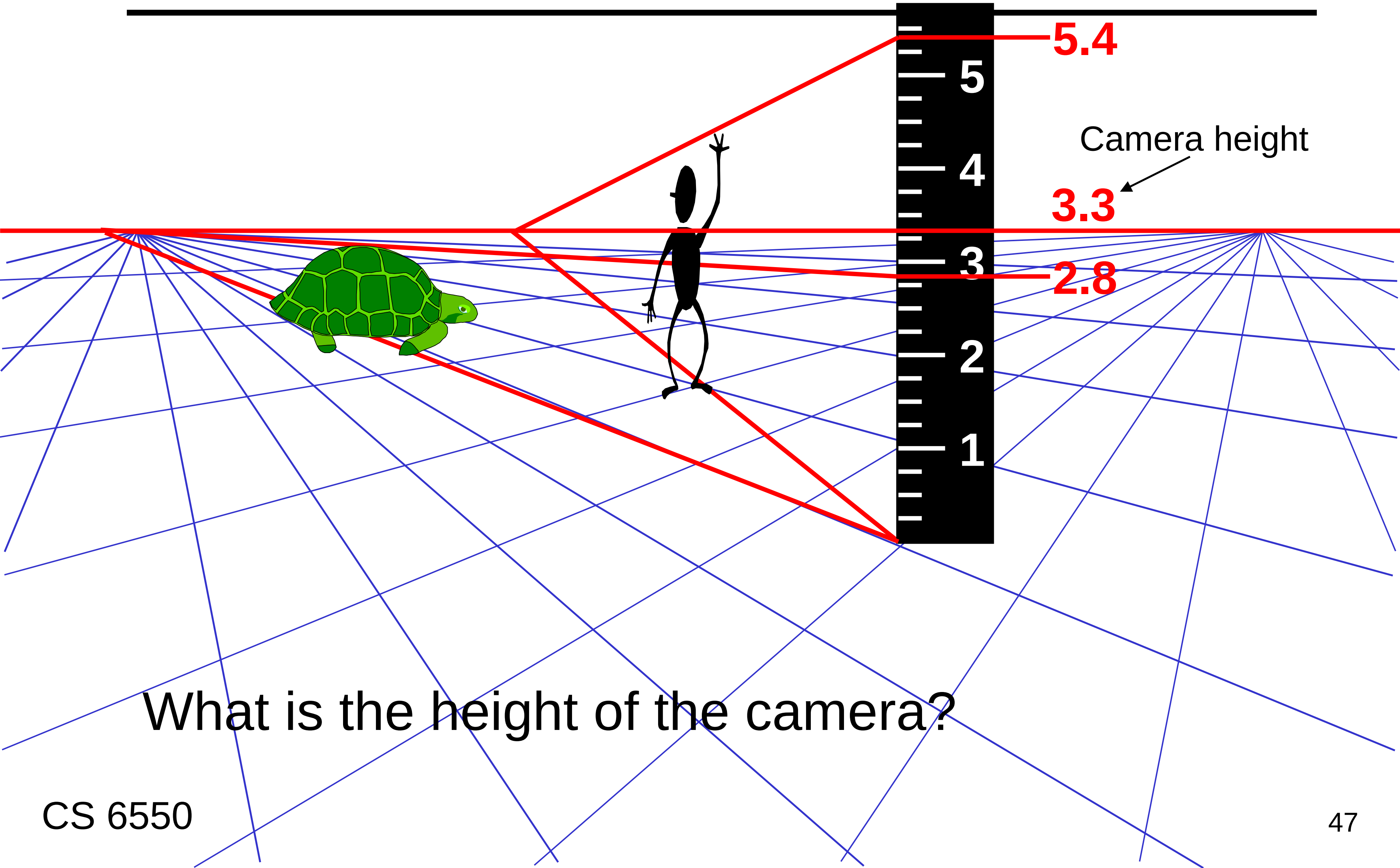




Comparing heights

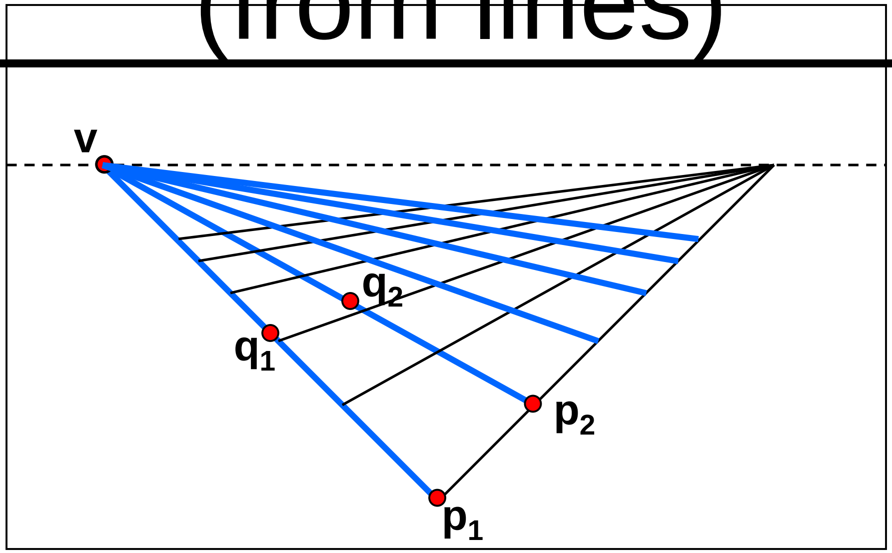


Measuring height



What is the height of the camera?

Computing vanishing points (from lines)



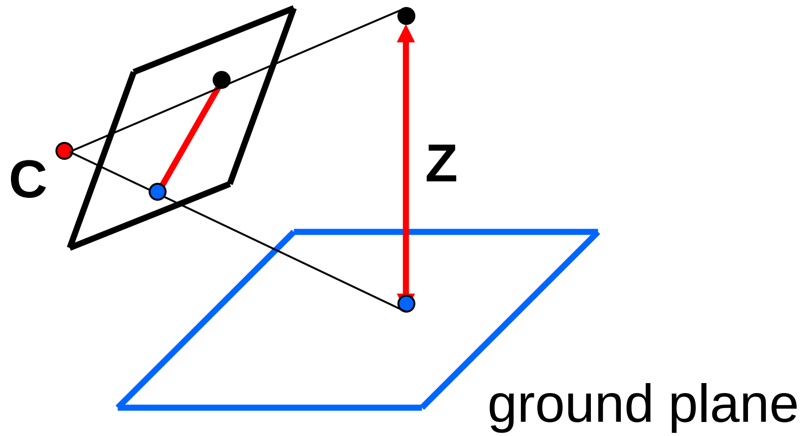
- Intersect p_1q_1 with p_2q_2

$$v = (p_1 \times q_1) \times (p_2 \times q_2)$$

Least squares version

- Better to use more than two lines and compute the "closest" point of intersection

Measuring height from image



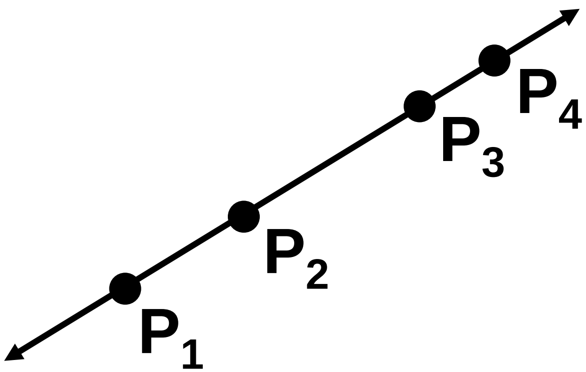
Compute Z from image measurements

- Need more than vanishing points to do this

Cross ratio

- A Projective Invariant
  - Something that does not change under projective transformations (including perspective projection)

The cross-ratio of 4 collinear points



$$\frac{\|\mathbf{P}_3 - \mathbf{P}_1\| \, \|\mathbf{P}_4 - \mathbf{P}_2\|}{\|\mathbf{P}_3 - \mathbf{P}_2\| \, \|\mathbf{P}_4 - \mathbf{P}_1\|}$$

$$\mathbf{P}_i = \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

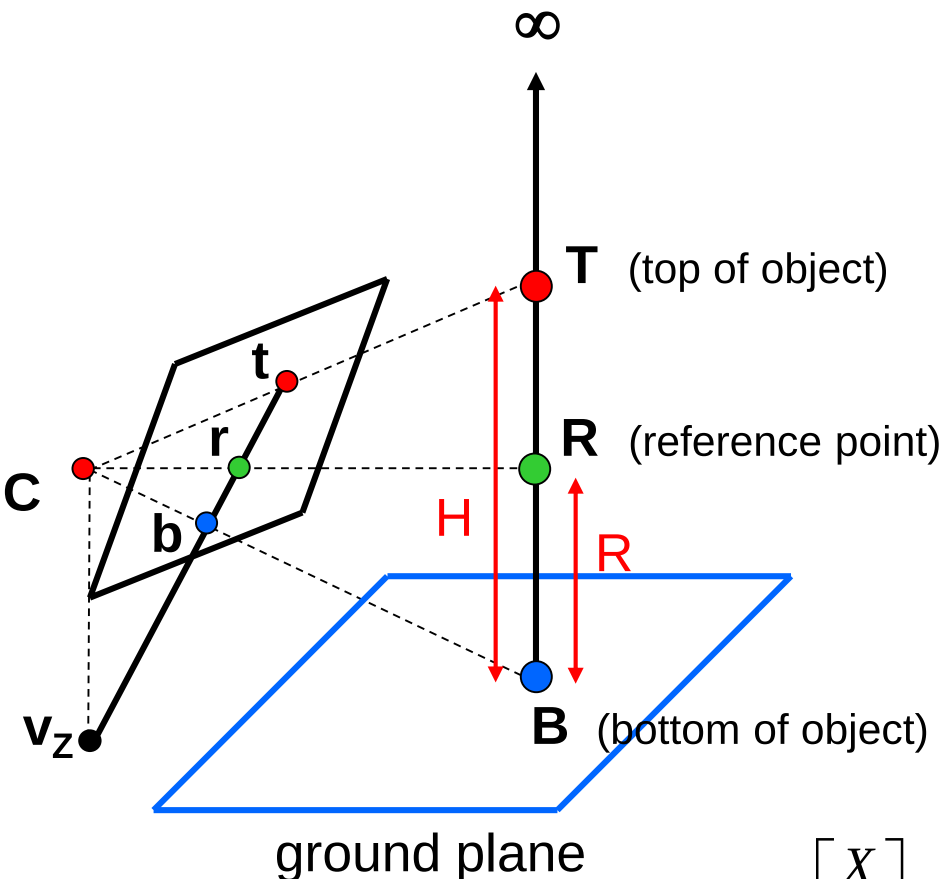
$$\frac{\|\mathbf{P}_1 - \mathbf{P}_3\| \, \|\mathbf{P}_4 - \mathbf{P}_2\|}{\|\mathbf{P}_1 - \mathbf{P}_2\| \, \|\mathbf{P}_4 - \mathbf{P}_3\|}$$

Can permute the point ordering

- 4! = 24 different orders (but only 6 distinct values)

This is the fundamental invariant of projective geometry

Measuring height



$$\frac{\|\mathbf{T}-\mathbf{B}\|\,\|\infty-\mathbf{R}\|}{\|\mathbf{R}-\mathbf{B}\|\,\|\infty-\mathbf{T}\|}=\frac{H}{R}$$

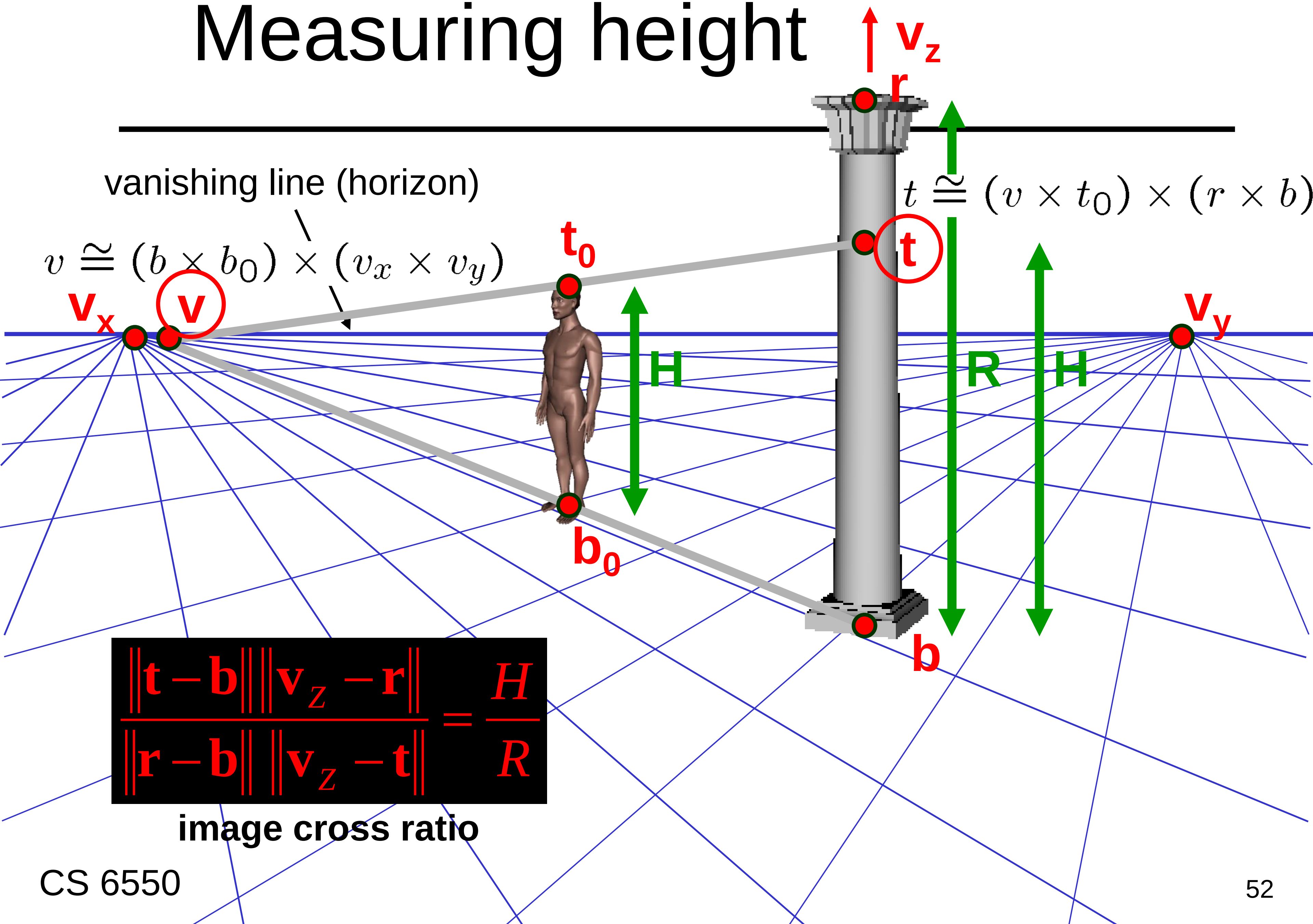
scene cross ratio

$$\frac{\|\mathbf{t}-\mathbf{b}\|\|\mathbf{v}_Z-\mathbf{r}\|}{\|\mathbf{r}-\mathbf{b}\|\,\|\mathbf{v}_Z-\mathbf{t}\|}=\frac{H}{R}$$

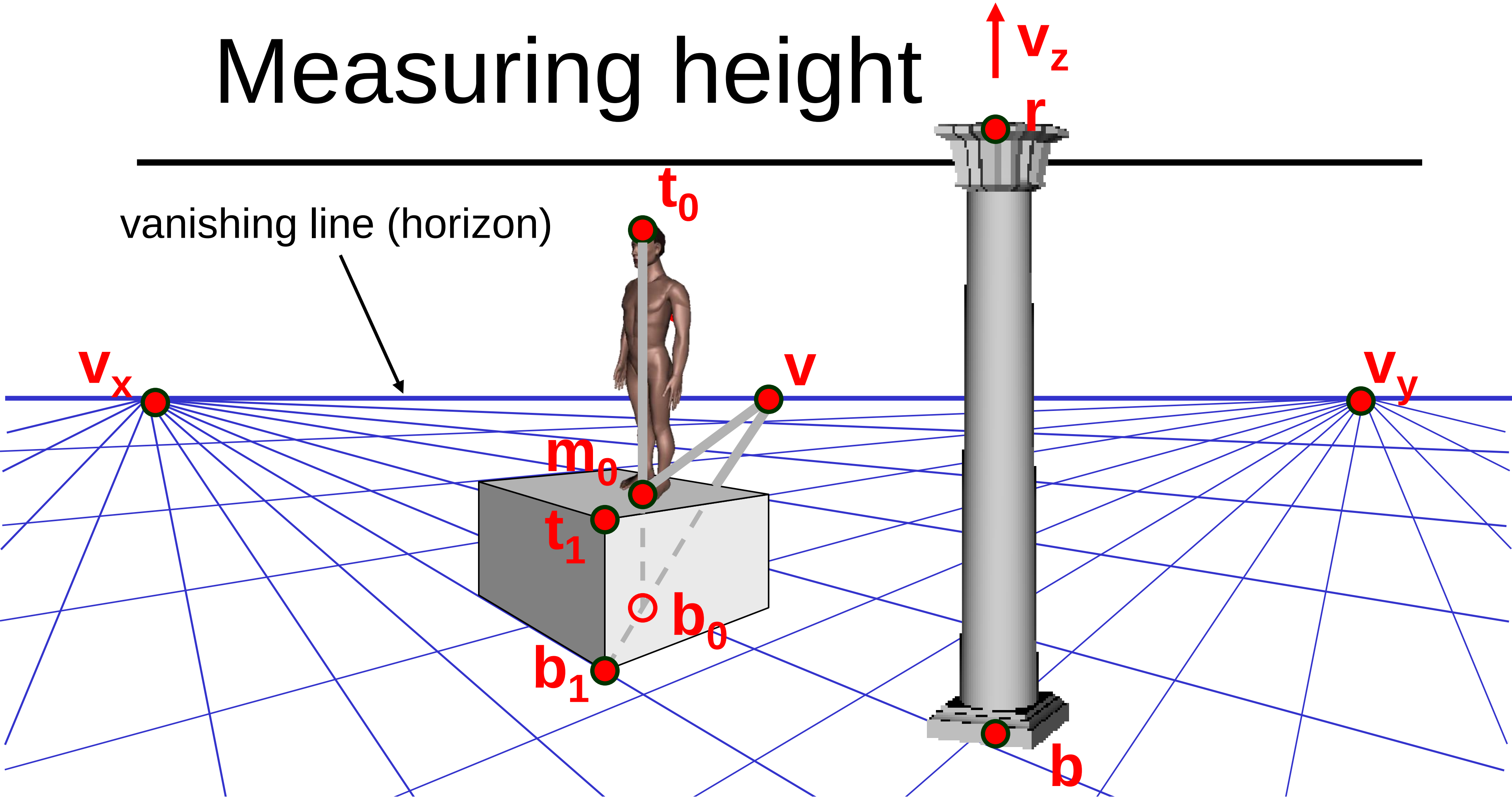
image cross ratio

scene points represented as $\mathbf{P}=\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix}$ image points as $\mathbf{p}=\begin{bmatrix}x\\y\\1\end{bmatrix}$

Measuring height



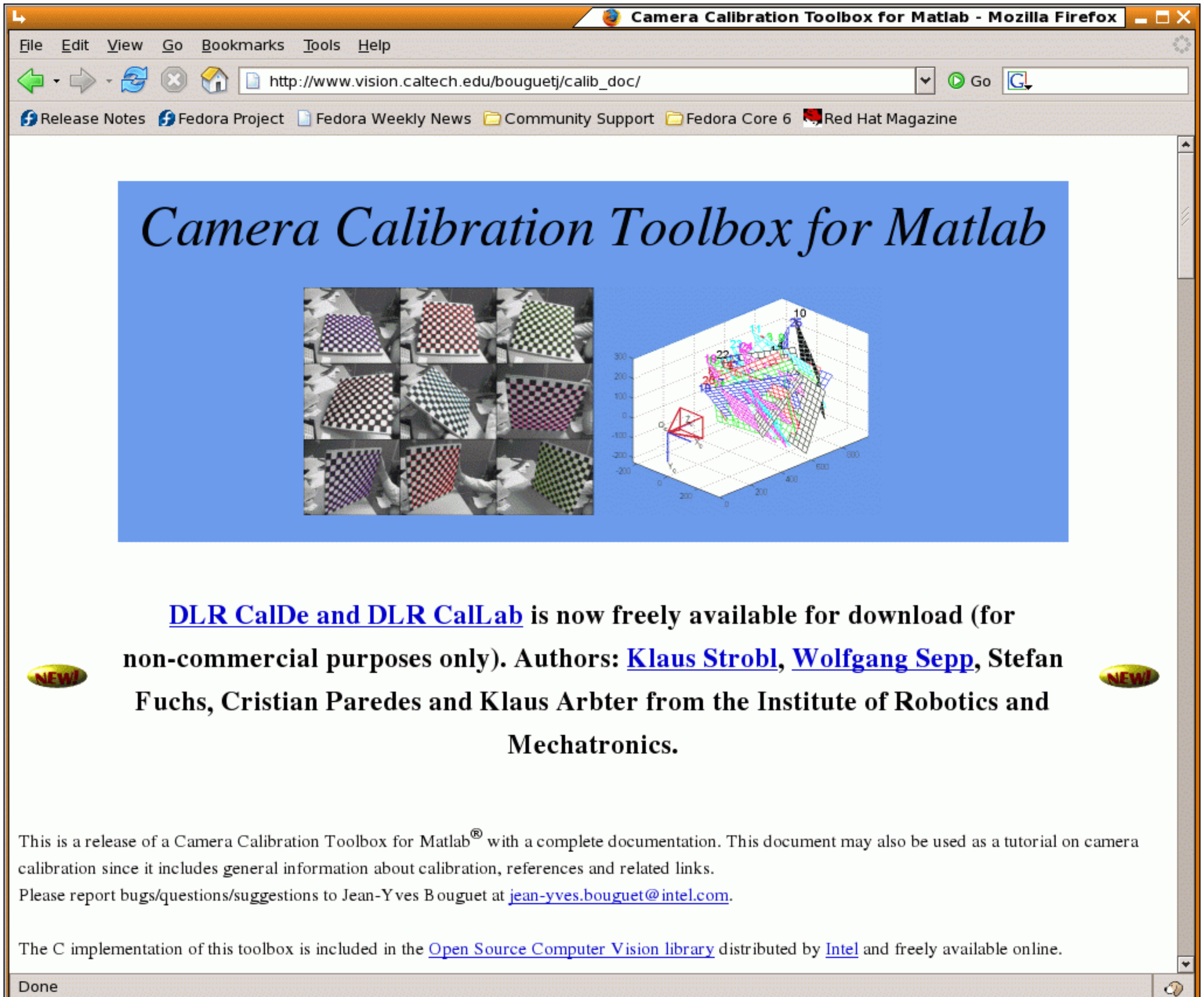
Measuring height



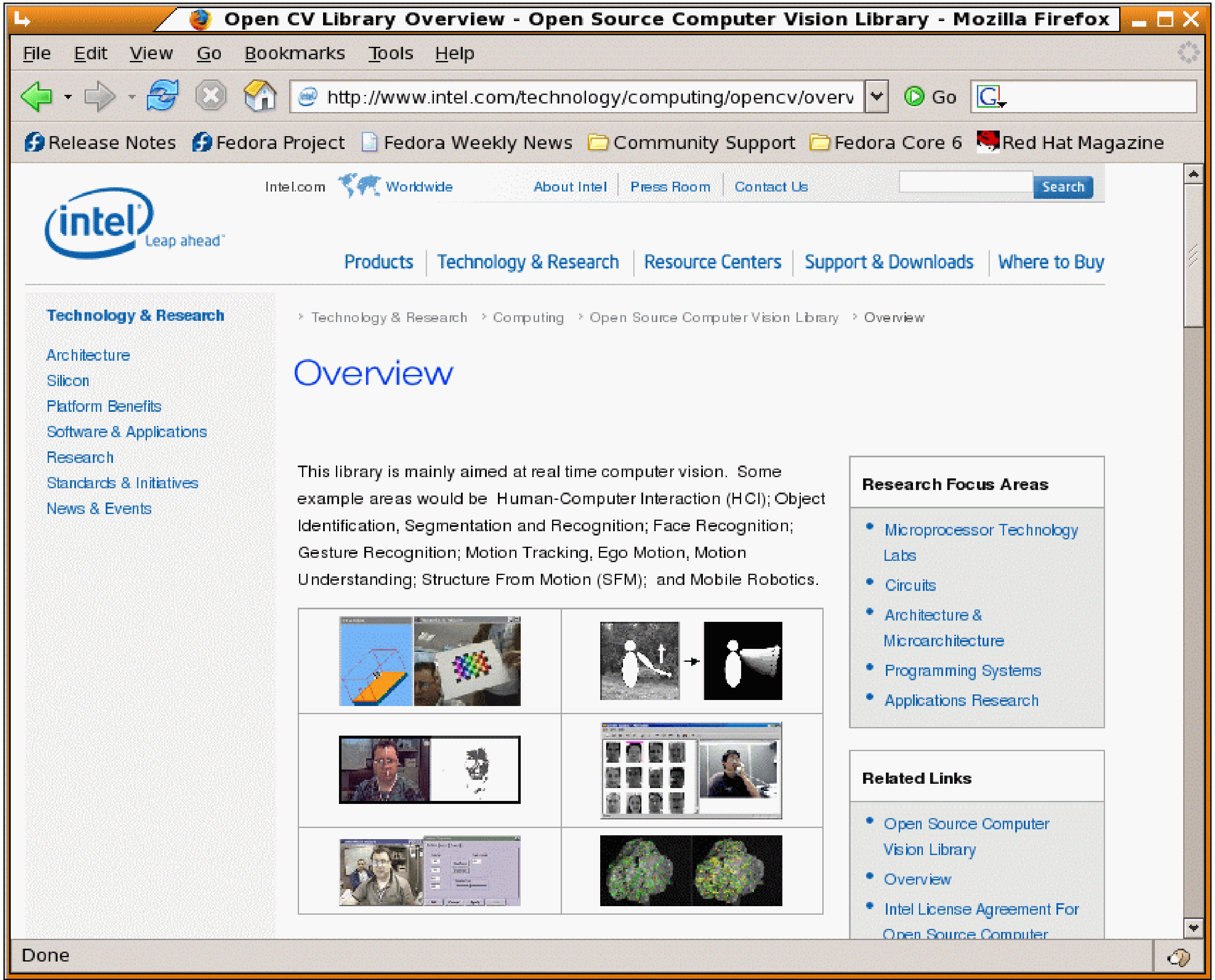
What if the point on the ground plane $\mathbf{b_0}$ is not known?

- Here the guy is standing on the box, height of box is known
- Use one side of the box to help find $\mathbf{b_0}$ as shown above

Calibration Software: Matlab



Calibration Software: OpenCV



Summary

- Camera calibration
  - DLT
- Homography transformation
- Homogeneous coordinates
  - Vanishing points, vanishing lines
- Cross ratio – projective invariant