Camera Geometry II

COS 429

Princeton University

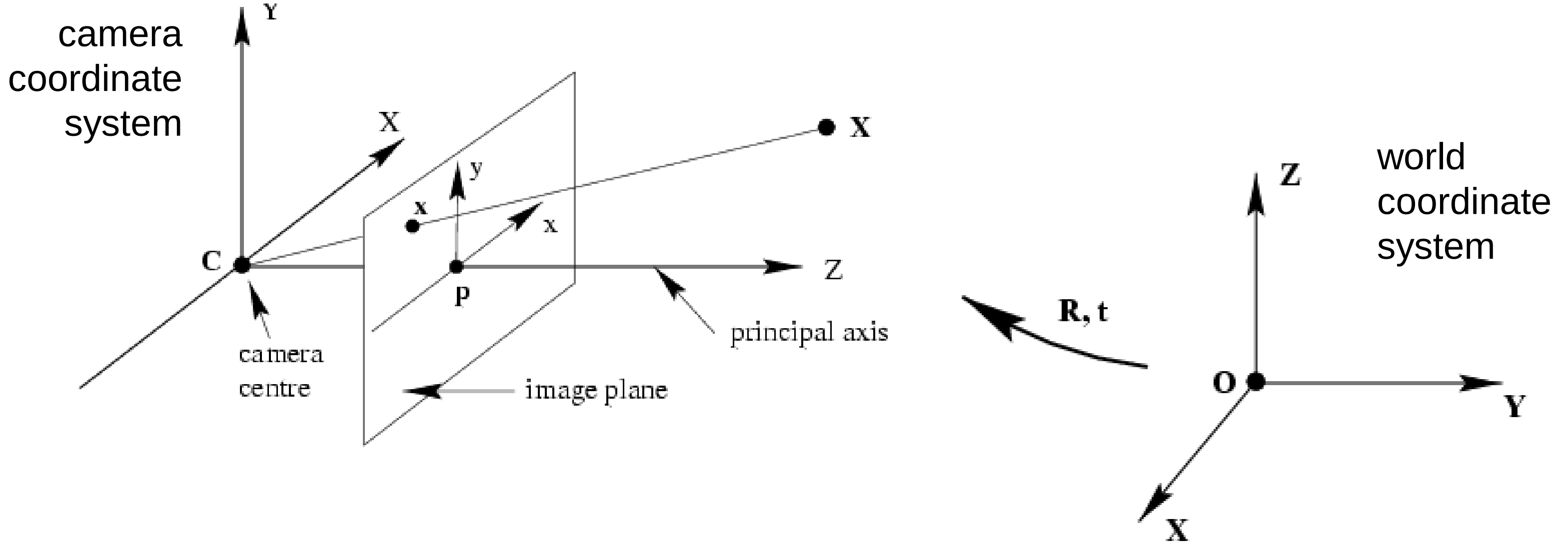
Outline

- Projective geometry
- Vanishing points
  - Application: camera calibration
  - Application: single-view metrology
- Epipolar geometry
  - Application: stereo correspondence
  - Application: structure from motion revisited

Outline

- Projective geometry
- Vanishing points
  - Application: camera calibration
  - Application: single-view metrology
- Epipolar geometry
  - Application: stereo correspondence
  - Application: structure from motion revisited

Review: Camera projection matrix



$$\lambda \mathrm{x} = \mathrm{PX} \qquad \lambda \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{bmatrix} \alpha_x & & \beta_x \\ & \alpha_y & \beta_y \\ & & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

intrinsic parameters

extrinsic parameters

Review: Camera parameters

- Intrinsic parameters
  - Image center (p_x,p_y)
  - Focal length (f)
  - Pixel magnification (m_x,m_y)
  - *Skew (non-rectangular pixels)*
  - *Radial distortion*

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

$$\mathbf{K} = \begin{bmatrix} m_x & & \\ & m_y & \\ & & 1 \end{bmatrix}\begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} = \begin{bmatrix} \alpha_x & & \beta_x \\ & \alpha_y & \beta_y \\ & & 1 \end{bmatrix}$$

Review: Camera parameters

- Intrinsic parameters
  - Principal point coordinates
  - Focal length
  - Pixel magnification
  - *Skew (non-rectangular pixels)*
  - *Radial distortion*

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

- Extrinsic parameters
  - Rotation (R) and translation (t) relative to world coordinate system

Review: Camera calibration

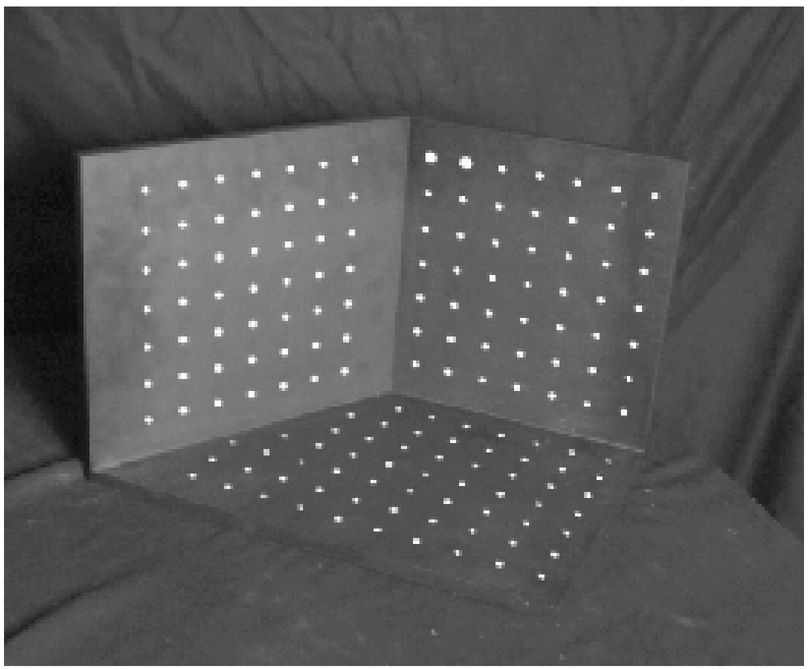
$$\lambda \mathbf{x} = \mathbf{K}\begin{bmatrix}\mathbf{R} & \mathbf{t}\end{bmatrix}\mathbf{X}$$

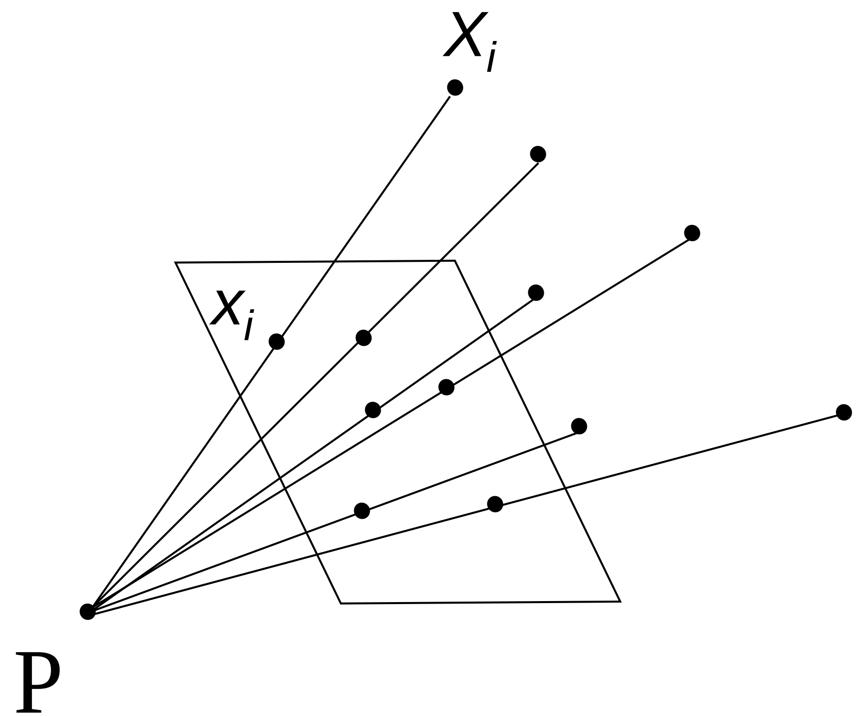
$$\lambda \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Source: D. Hoiem

Review: Camera calibration

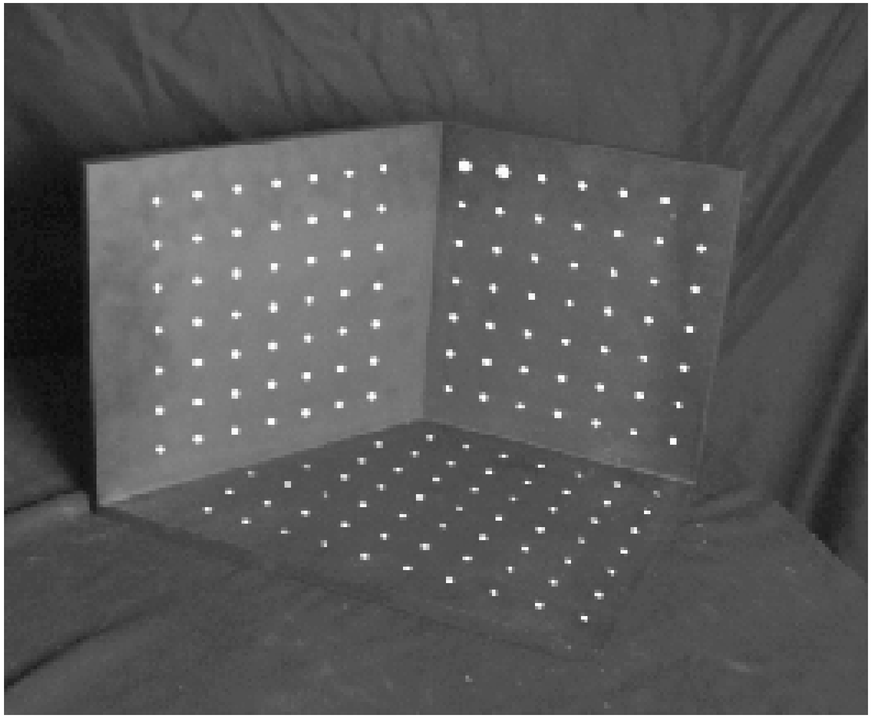
- Given n points with known 3D coordinates X_i and known image projections x_i, estimate the camera parameters





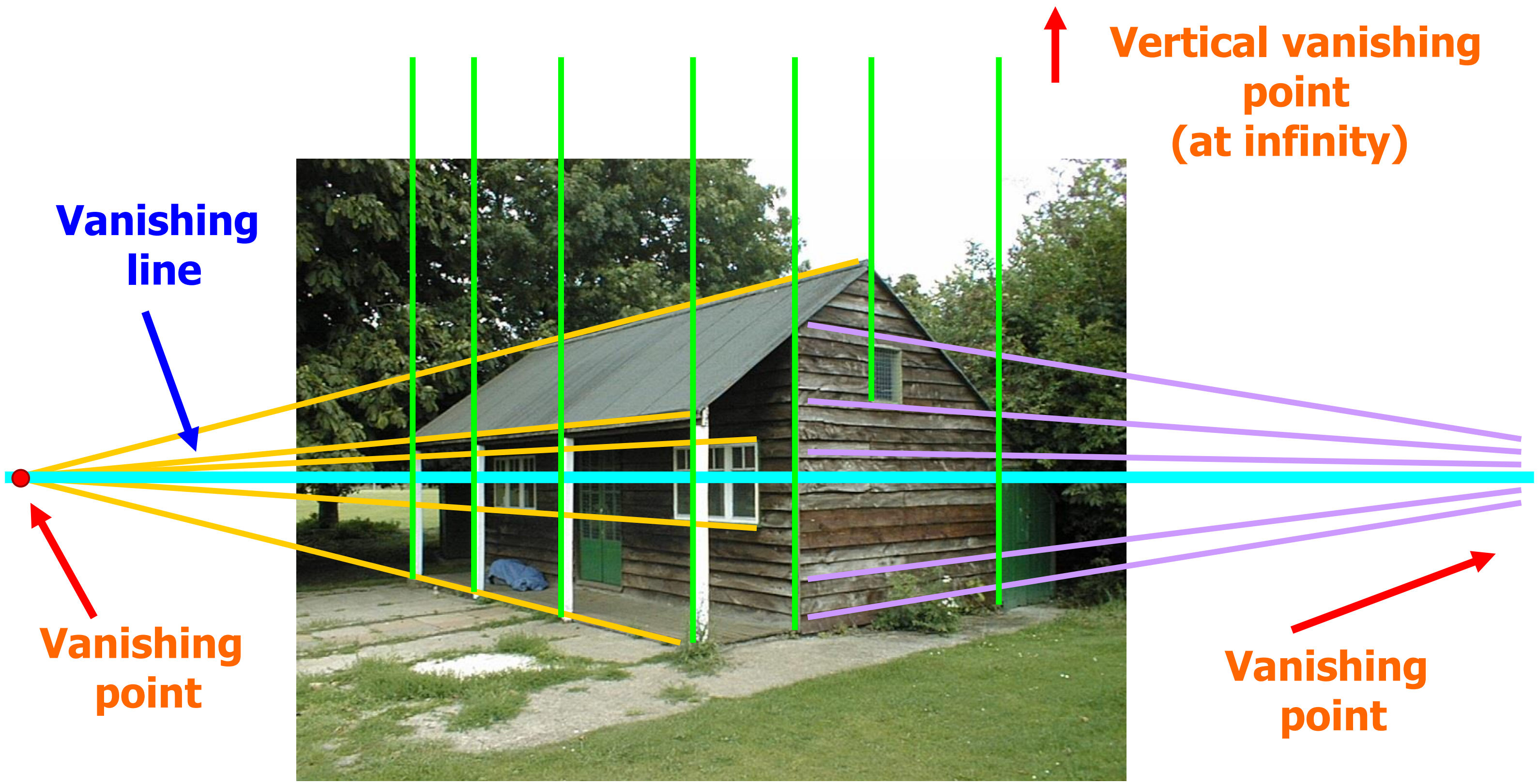
Camera calibration

- What if don’t know correspondences?
- Can you determine the camera intrinsic parameters (focal length, center) or extrinsic parameters (rotation, translation)?



Camera calibration

- Let's see what we can get from vanishing points



Slide from Efros, Photo from Criminisi

Outline

Projective geometry

Vanishing points

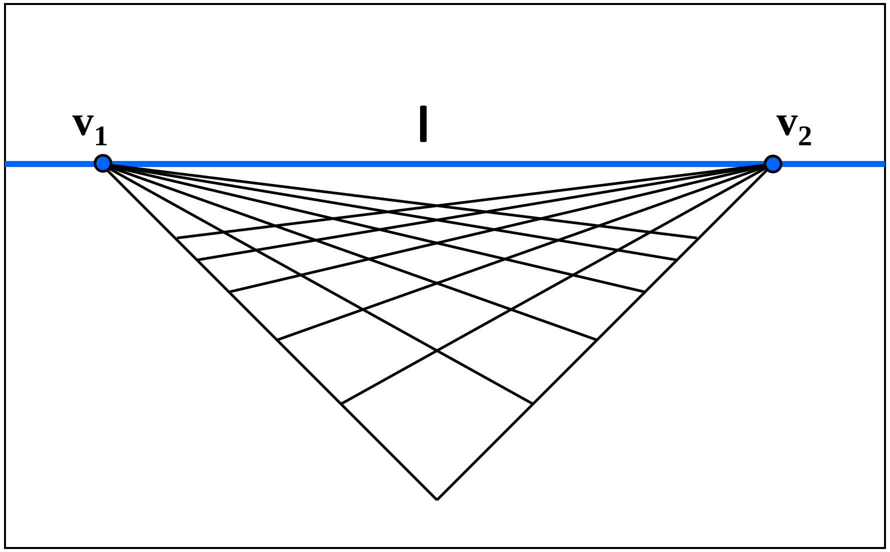
- Application: camera calibration
- Application: single-view metrology

Epipolar geometry

- Application: stereo correspondence
- Application: structure from motion revisited

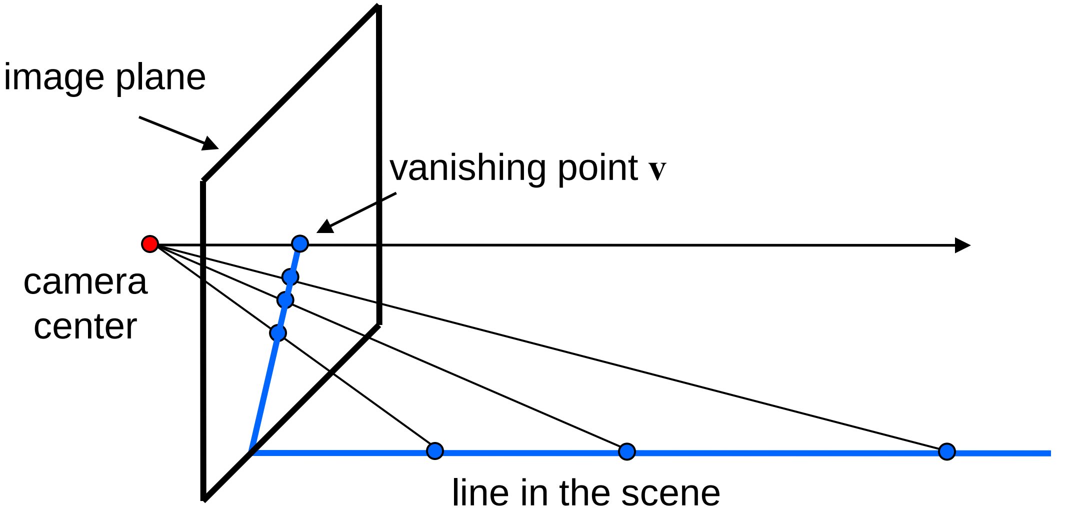
Review: Vanishing Points

- Any set of parallel lines on a plane define a vanishing point
- The union of all of these vanishing points is the *horizon line*
  - also called *vanishing line*
- Different planes (can) define different vanishing lines



Noah Snavely

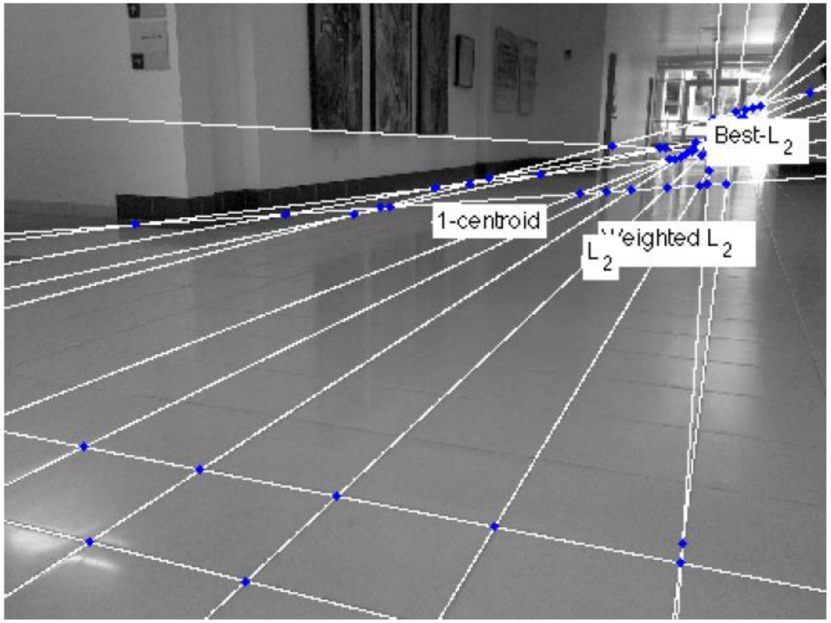
Review: Vanishing points



- All lines having the same direction share the same vanishing point

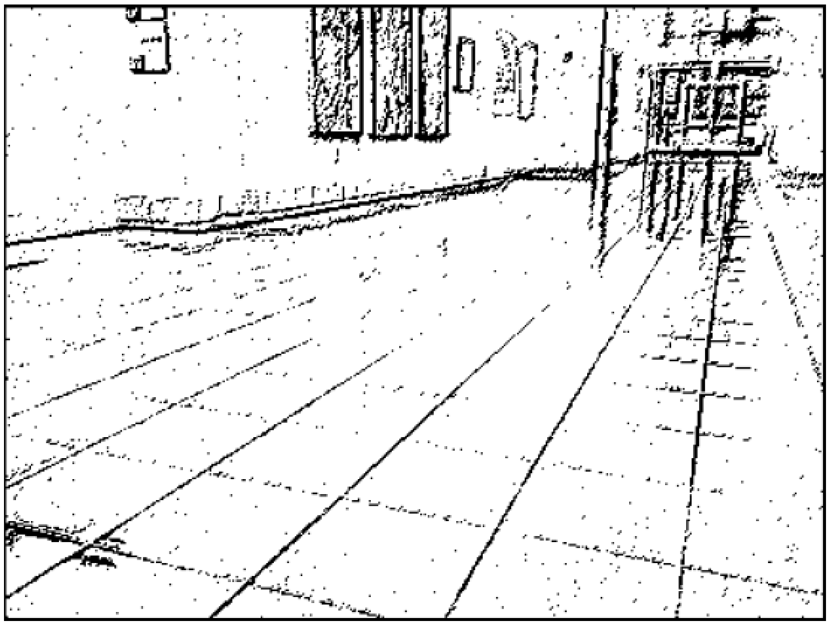
Computing Vanishing Points

How can we find lines in an image?



Computing Vanishing Points

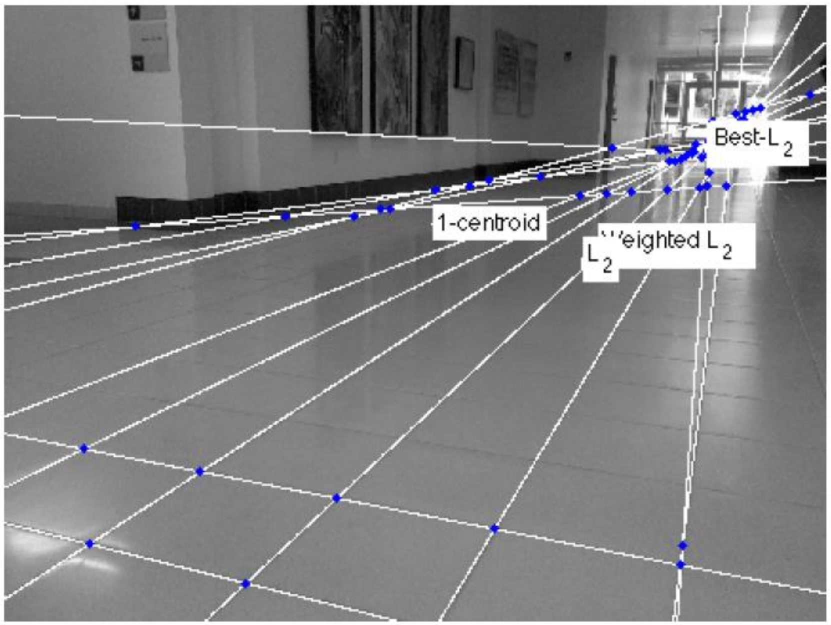
Edge detection



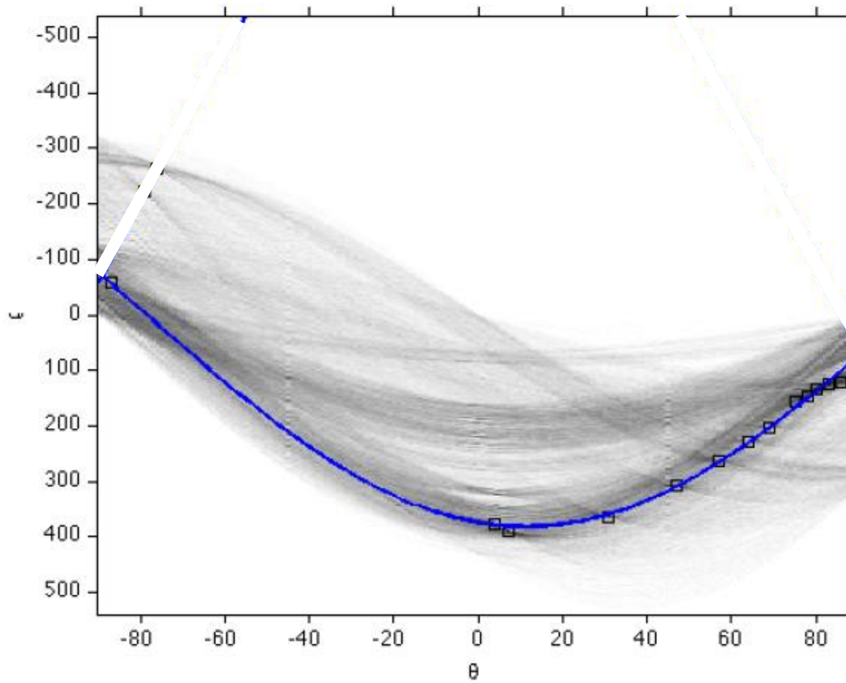
Edges

Computing Vanishing Points

Edge detection + Hough transform



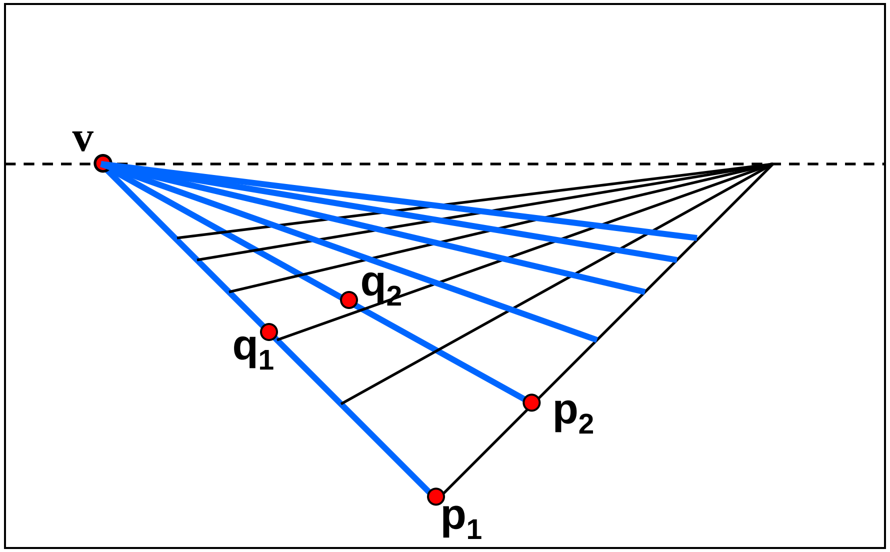
Strong Lines



Hough Transform

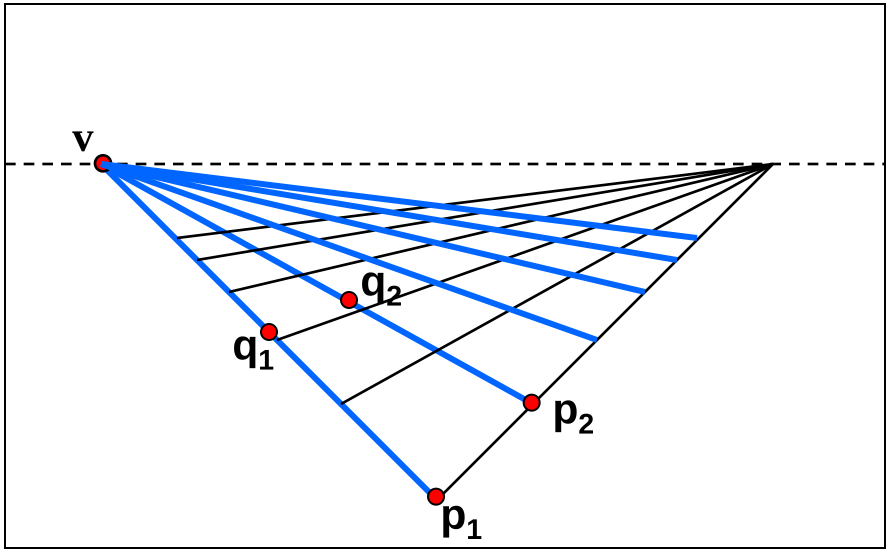
Papusha & Ho

Computing Vanishing Points



For a set of parallel lines, how can we find where they intersect?

Computing Vanishing Points

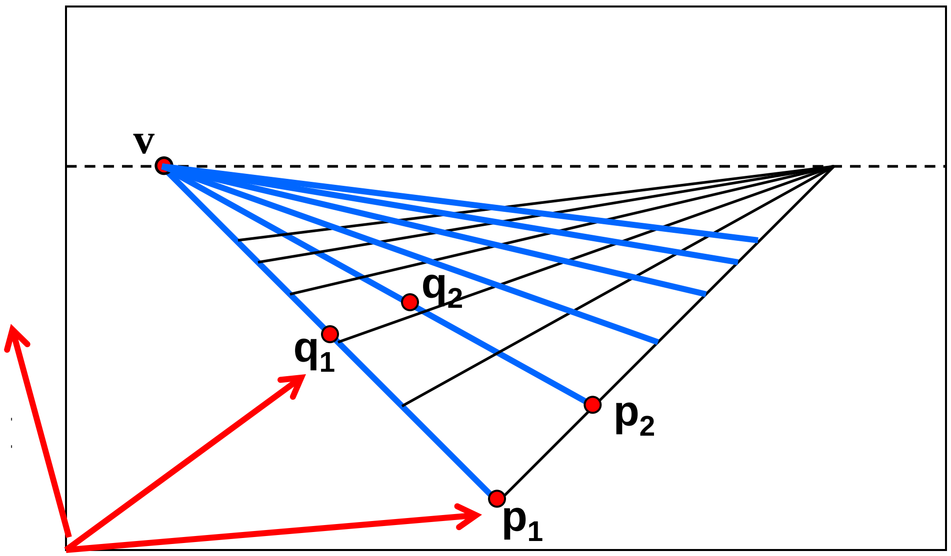


Intersect p_1q_1 with p_2q_2

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

Computing Vanishing Points

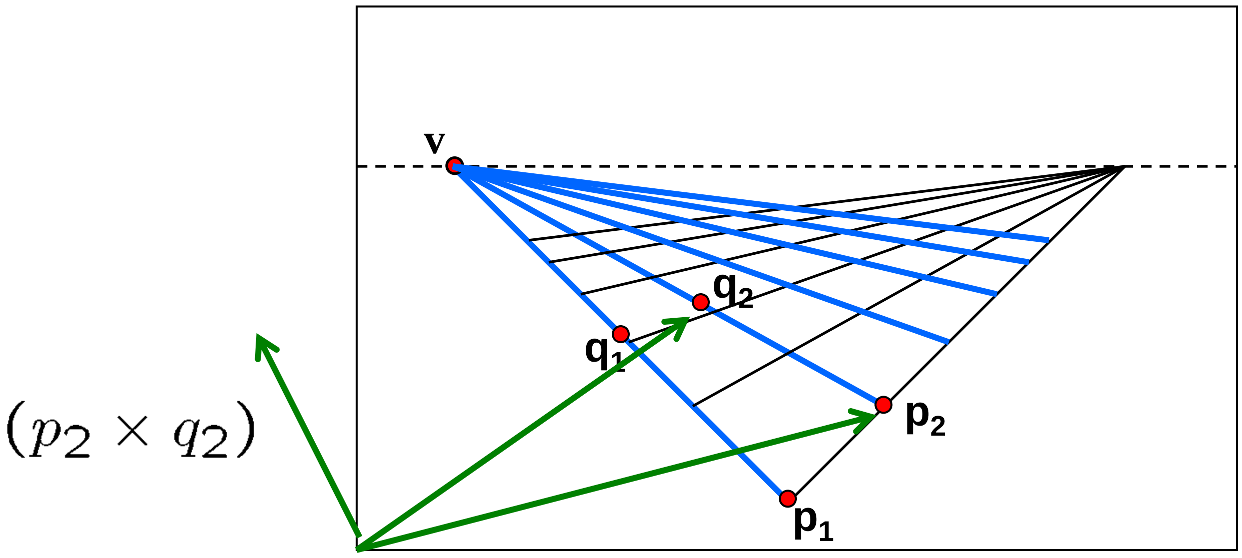
$(p_1 \times q_1)$:



Intersect p_1q_1 with p_2q_2

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

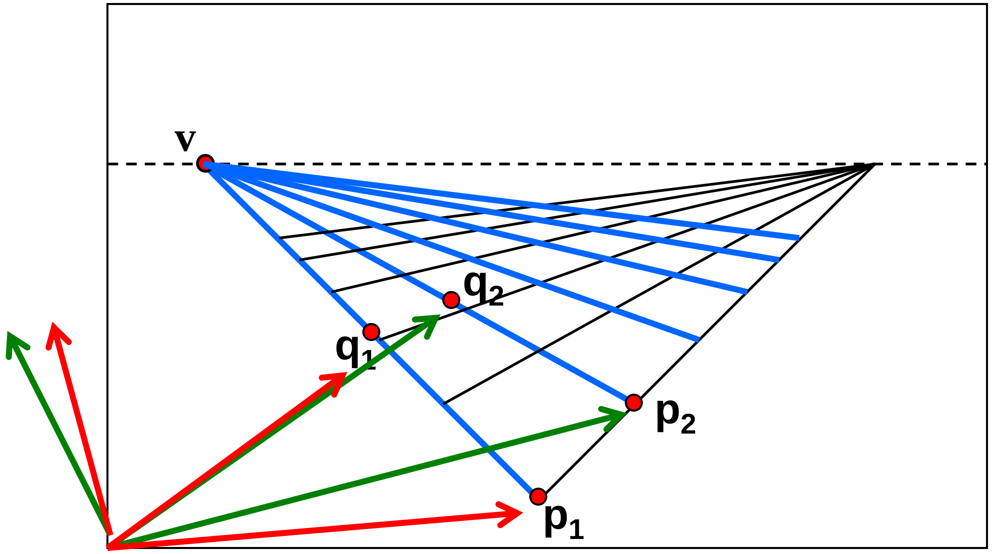
Computing Vanishing Points



Intersect p_1q_1 with p_2q_2

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

Computing Vanishing Points



Intersect p_1q_1 with p_2q_2

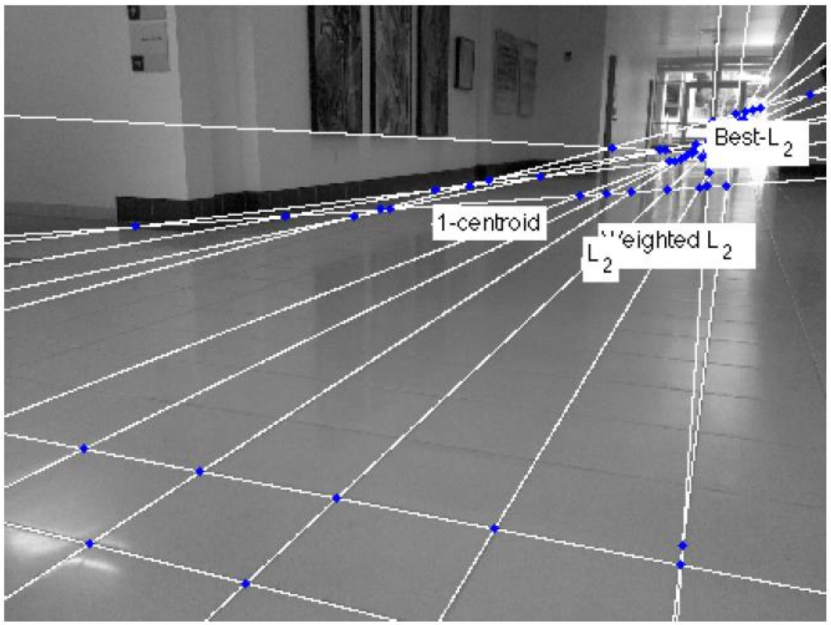
$$v = (p_1 \times q_1) \times (p_2 \times q_2)$$

Least squares version

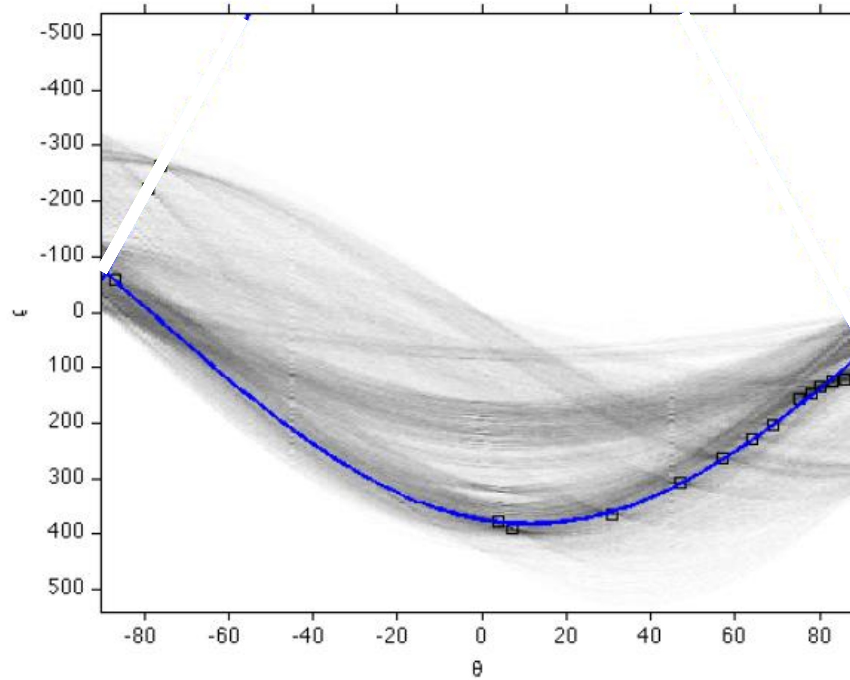
- Better to use more than two lines and compute the "closest" point of intersection

Computing Vanishing Points

Alternative: vanishing points can be extracted directly from Hough transform (fit sine curves)



Strong Lines

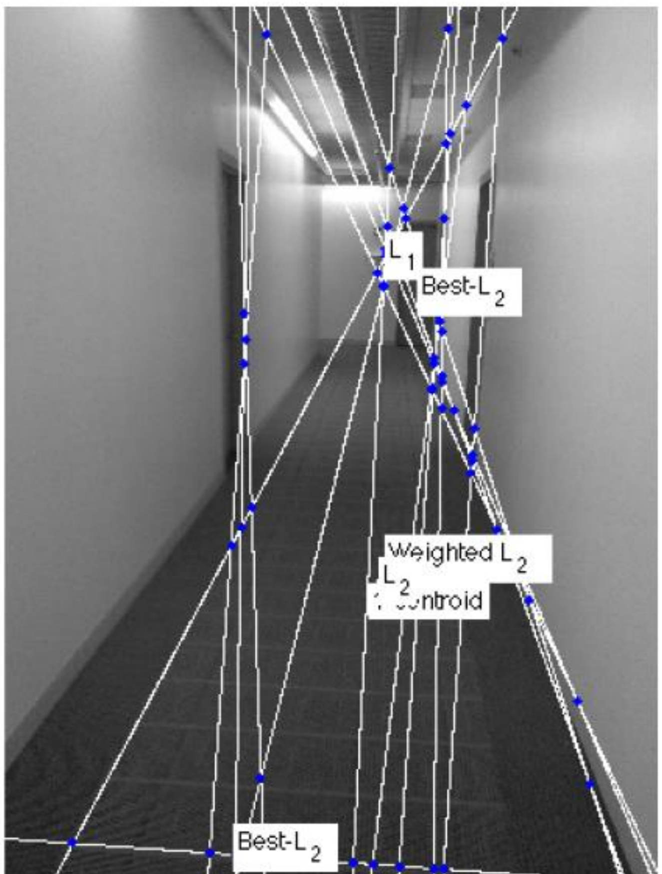


Hough Transform

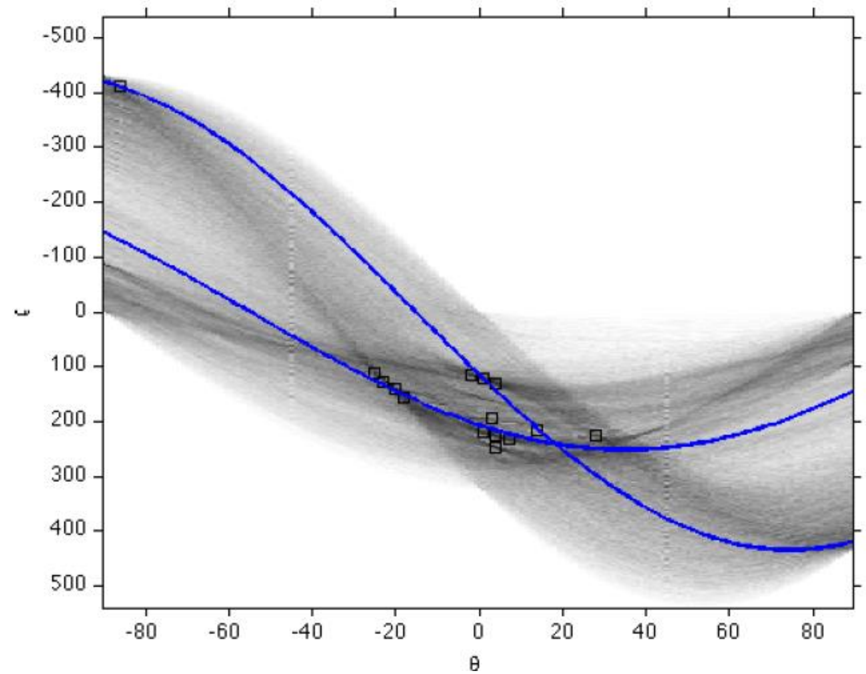
Papusha & Ho

Computing Vanishing Points

Vanishing points can be extracted directly from Hough transform (fit sine curves)



Strong Lines



Hough Transform

Papusha & Ho

Outline

Projective geometry

Vanishing points

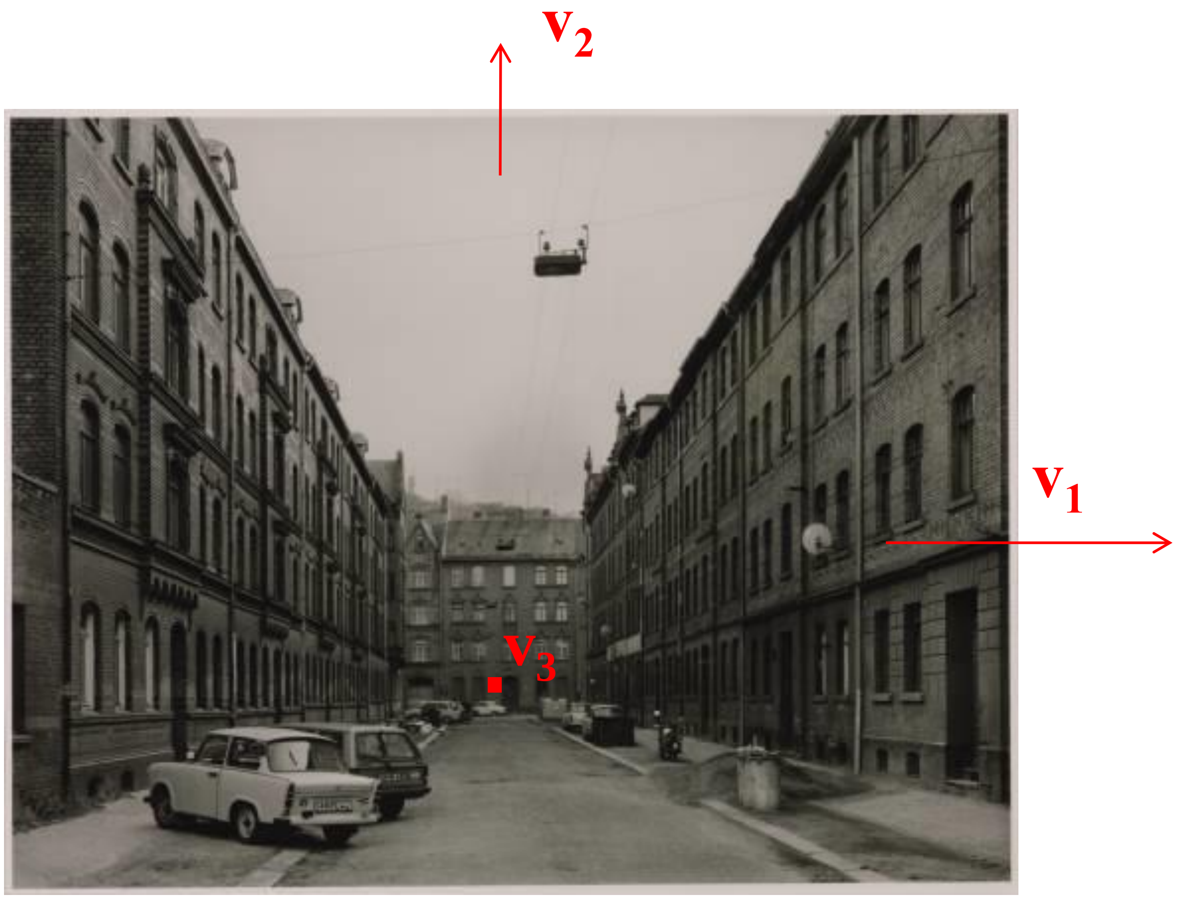
- Application: camera calibration
- Application: single-view metrology

Epipolar geometry

- Application: stereo correspondence
- Application: structure from motion revisited

Calibration from vanishing points

- What camera parameters can we calibrate using three orthogonal vanishing directions (points)?



Calibration from vanishing points

- Let us align the world coordinate system with three orthogonal vanishing directions in the scene:

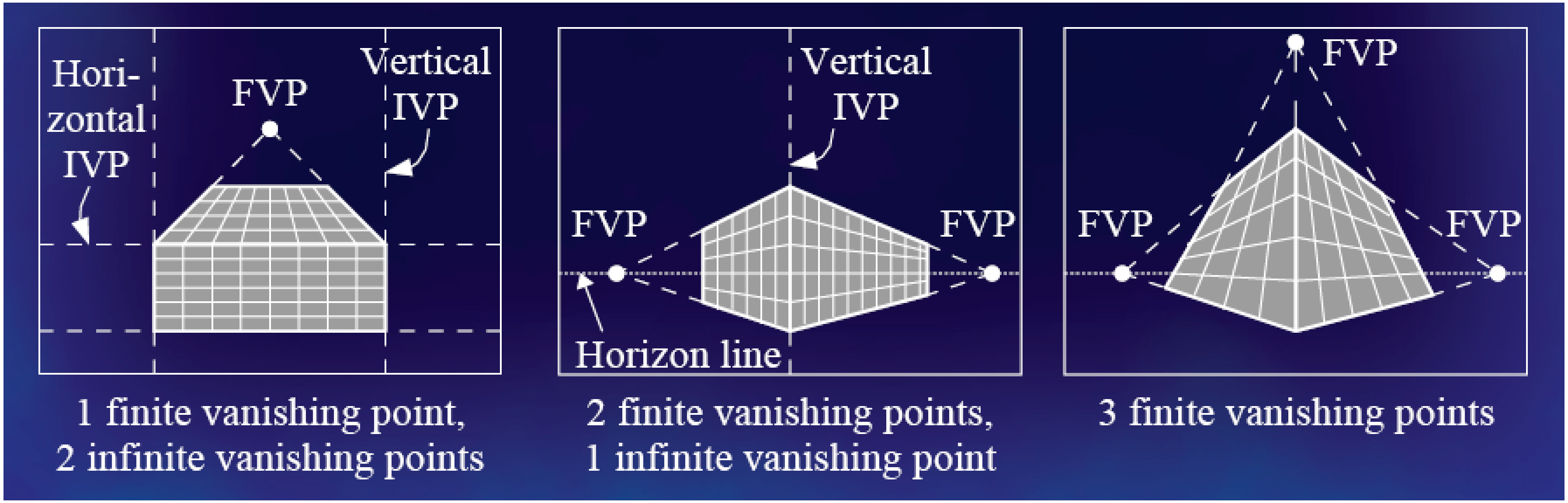
$$\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \mathbf{e}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad \lambda \mathbf{v}_i = \mathbf{K}\left[\mathbf{R} \mid \mathbf{t}\right]\begin{bmatrix} \mathbf{e}_i \\ 0 \end{bmatrix} = \mathbf{K}\mathbf{R}\mathbf{e}_i$$

$$\mathbf{e}_i = \lambda \mathbf{R}^T \mathbf{K}^{-1} \mathbf{v}_i, \quad \mathbf{e}_i^T \mathbf{e}_j = 0$$

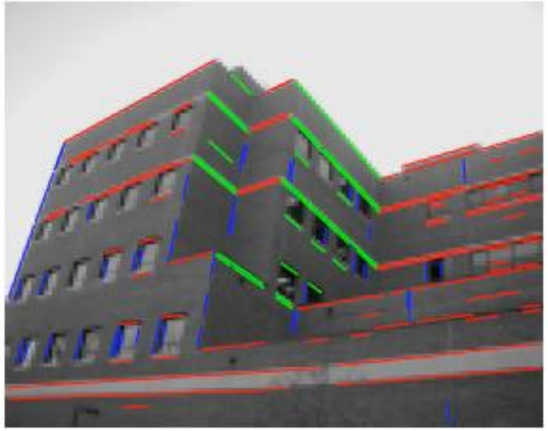
$$\mathbf{v}_i^T \mathbf{K}^{-T} \mathbf{R}\mathbf{R}^T \mathbf{K}^{-1} \mathbf{v}_j = \mathbf{v}_i^T \mathbf{K}^{-T} \mathbf{K}^{-1} \mathbf{v}_j = 0$$

- Each pair of vanishing points gives us a constraint on the focal length and principal point

Intrinsic calibration from vanishing points



Cannot recover focal length, image center is the finite vanishing point



Can solve for focal length, image center

Rotation from vanishing points

$$\lambda \mathbf{v}_i = \mathbf{K}\left[\mathbf{R} \mid \mathbf{t}\right]\begin{bmatrix}\mathbf{e}_i \\ 0\end{bmatrix} = \mathbf{K}\mathbf{R}\mathbf{e}_i$$

$$\lambda \mathbf{K}^{-1}\mathbf{v}_1 = \mathbf{R}\mathbf{e}_1 = [\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{r}_3]\begin{bmatrix}1 \\ 0 \\ 0\end{bmatrix} = \mathbf{r}_1$$

Thus, $\lambda \mathbf{K}^{-1}\mathbf{v}_i = \mathbf{r}_i$.

Get λ by using the constraint $\|\mathbf{r}_i\|^2=1$.

Calibration from vanishing points: Summary

- Solve for K (focal length, principal point) using three orthogonal vanishing points
- Get rotation directly from vanishing points once K is known

- Advantages
  - No need for calibration chart (2D-3D correspondences)
  - Could be completely automatic
- Disadvantages
  - Only applies to certain kinds of scenes
  - Inaccuracies in computation of vanishing points
  - Problems due to infinite vanishing points

Outline

Projective geometry

Vanishing points

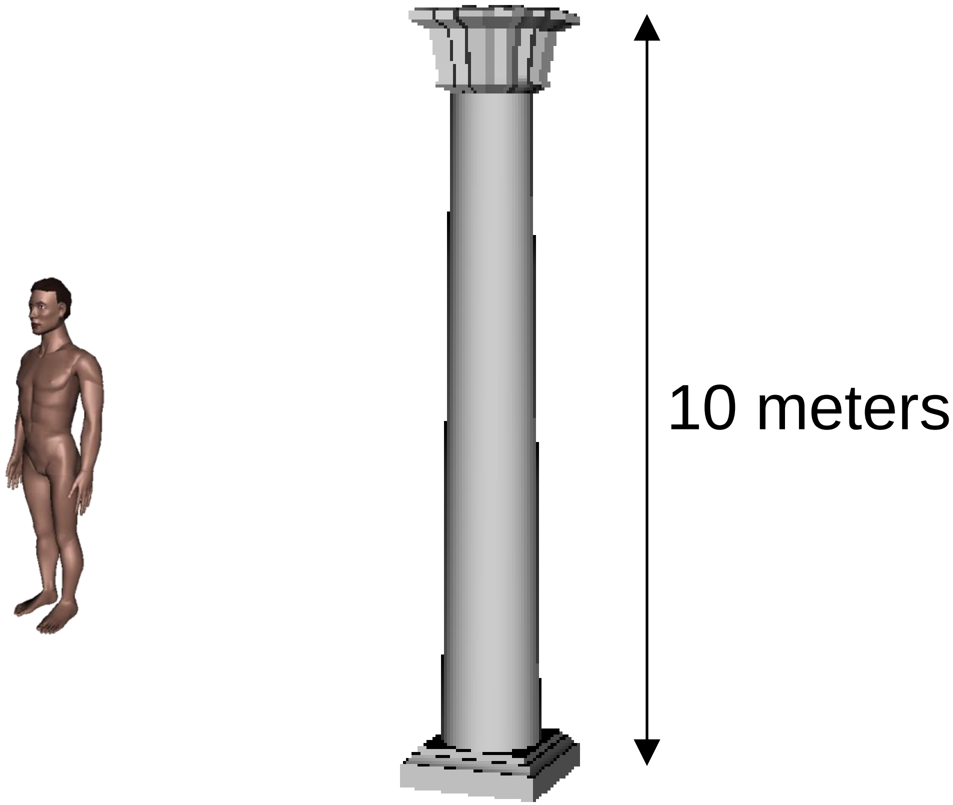
- Application: camera calibration
- Application: single-view metrology

Epipolar geometry

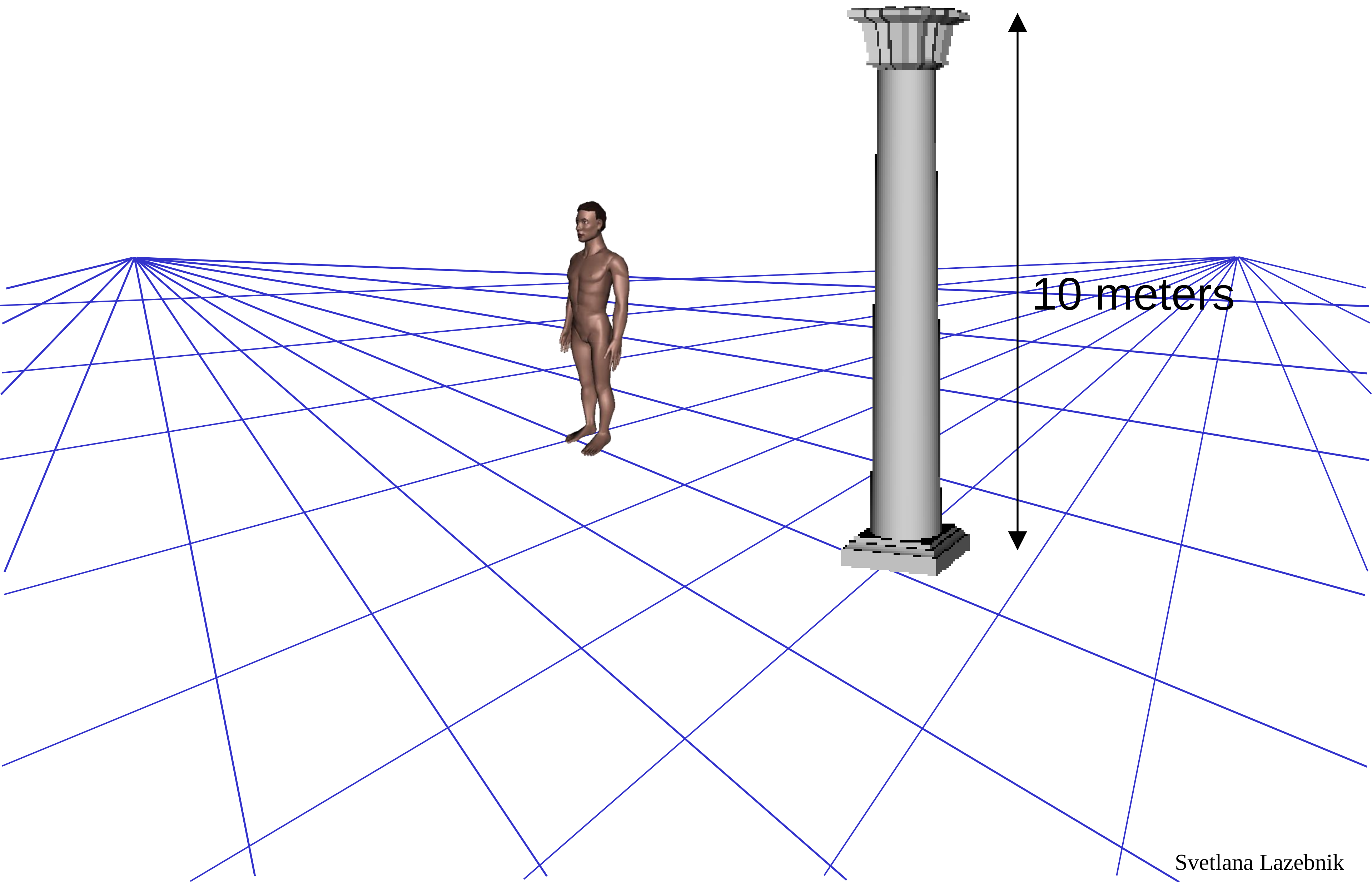
- Application: stereo correspondence
- Application: structure from motion revisited

How Tall is the Man in this Image?

How Tall is the Man in this Image?

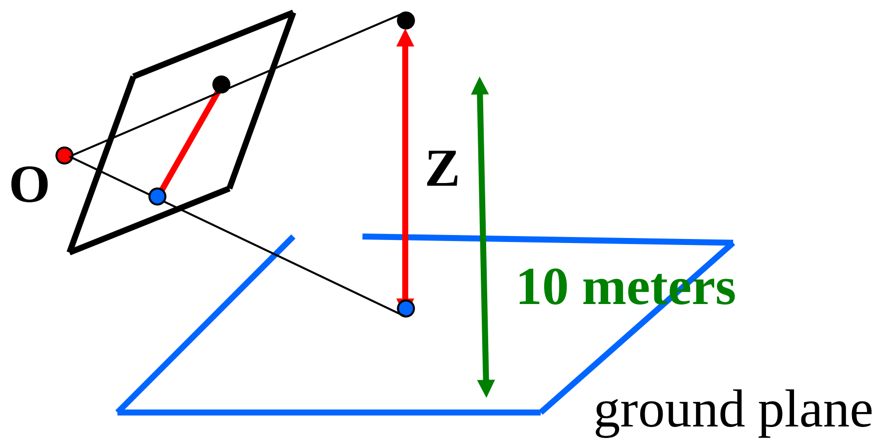


How Tall is the Man in this Image?



Svetlana Lazebnik

How Tall is the Man in this Image?

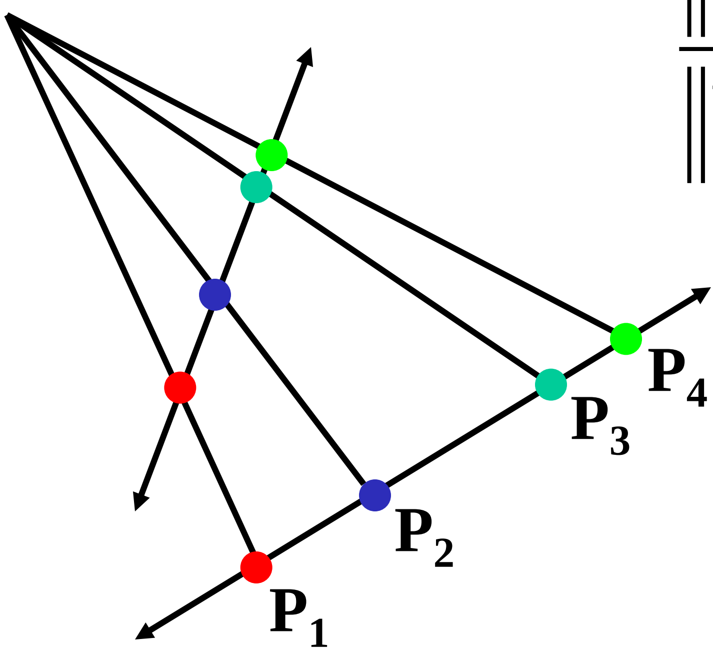


Goal: compute Z

Problem: depends on camera angle and distance

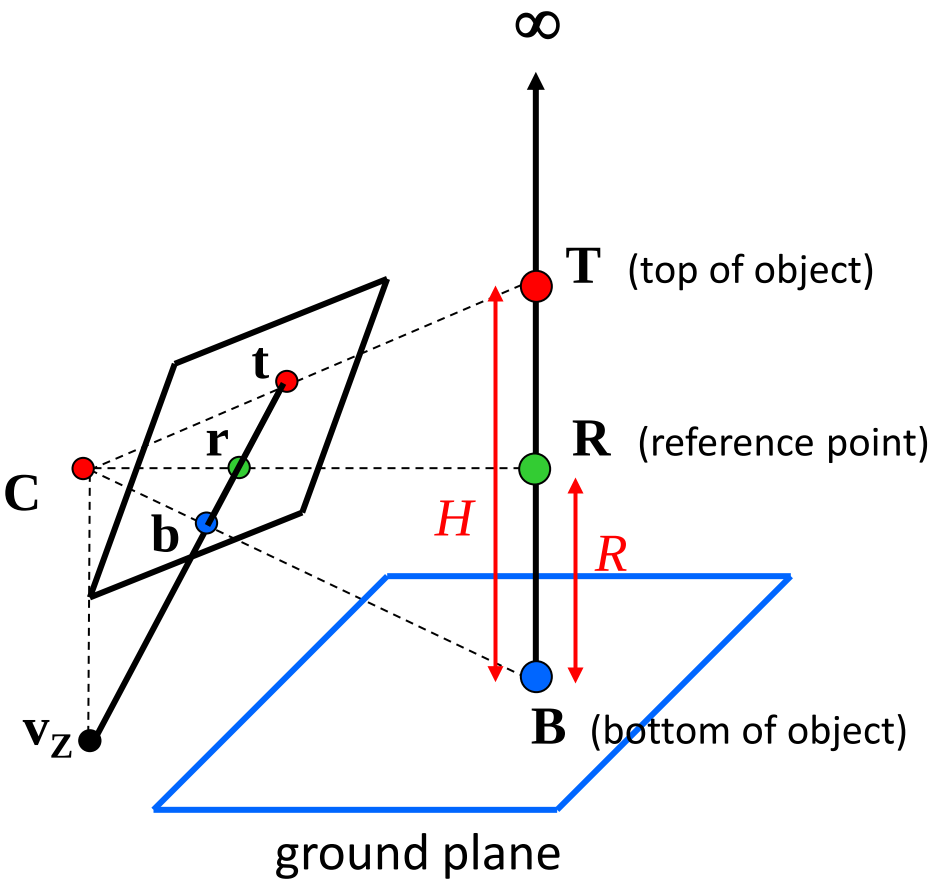
The cross-ratio

- A *projective invariant*: quantity that does not change under projective transformations (including perspective projection)
- The cross-ratio of four points:



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

Measuring height



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

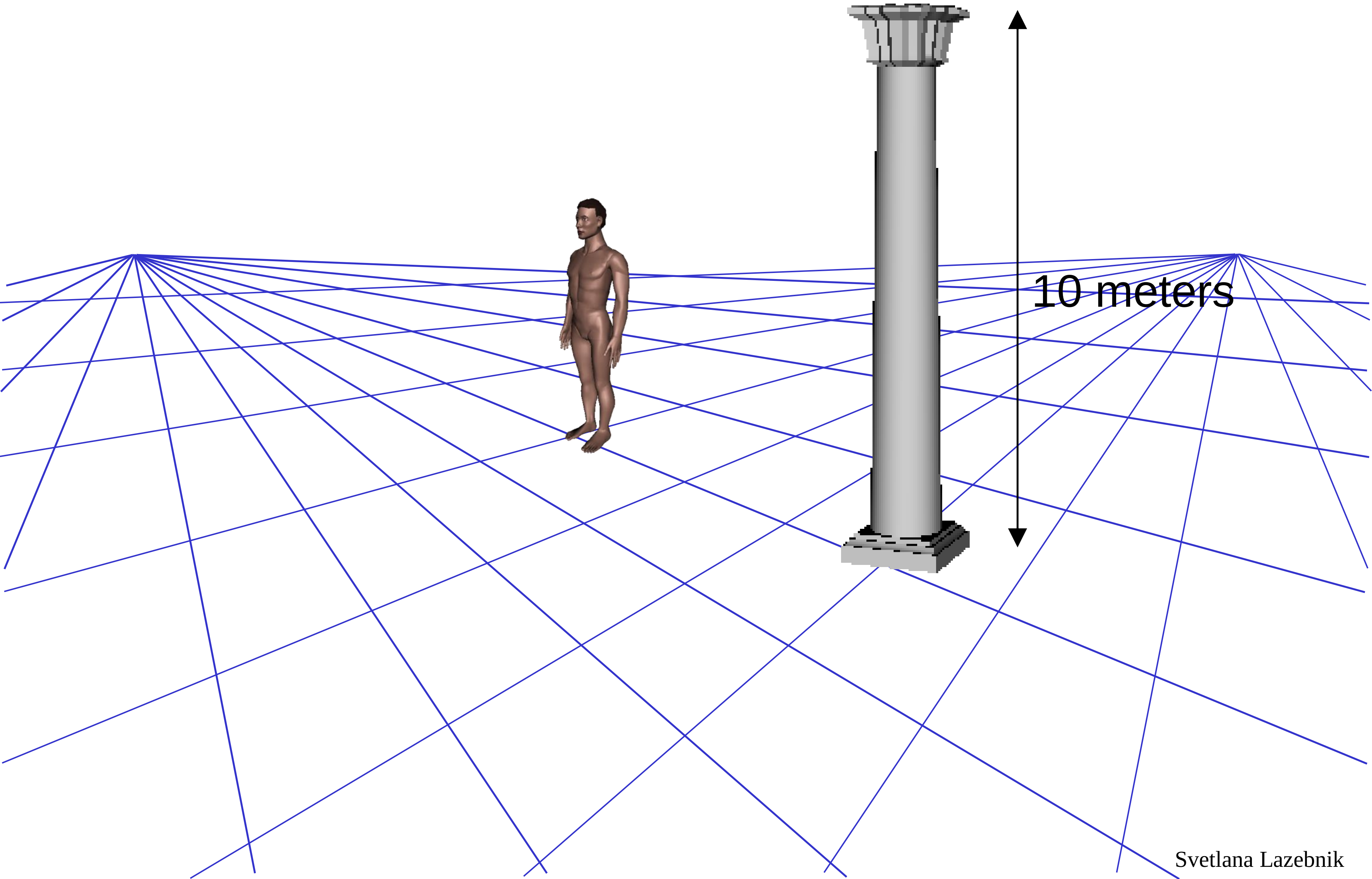
scene cross ratio

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

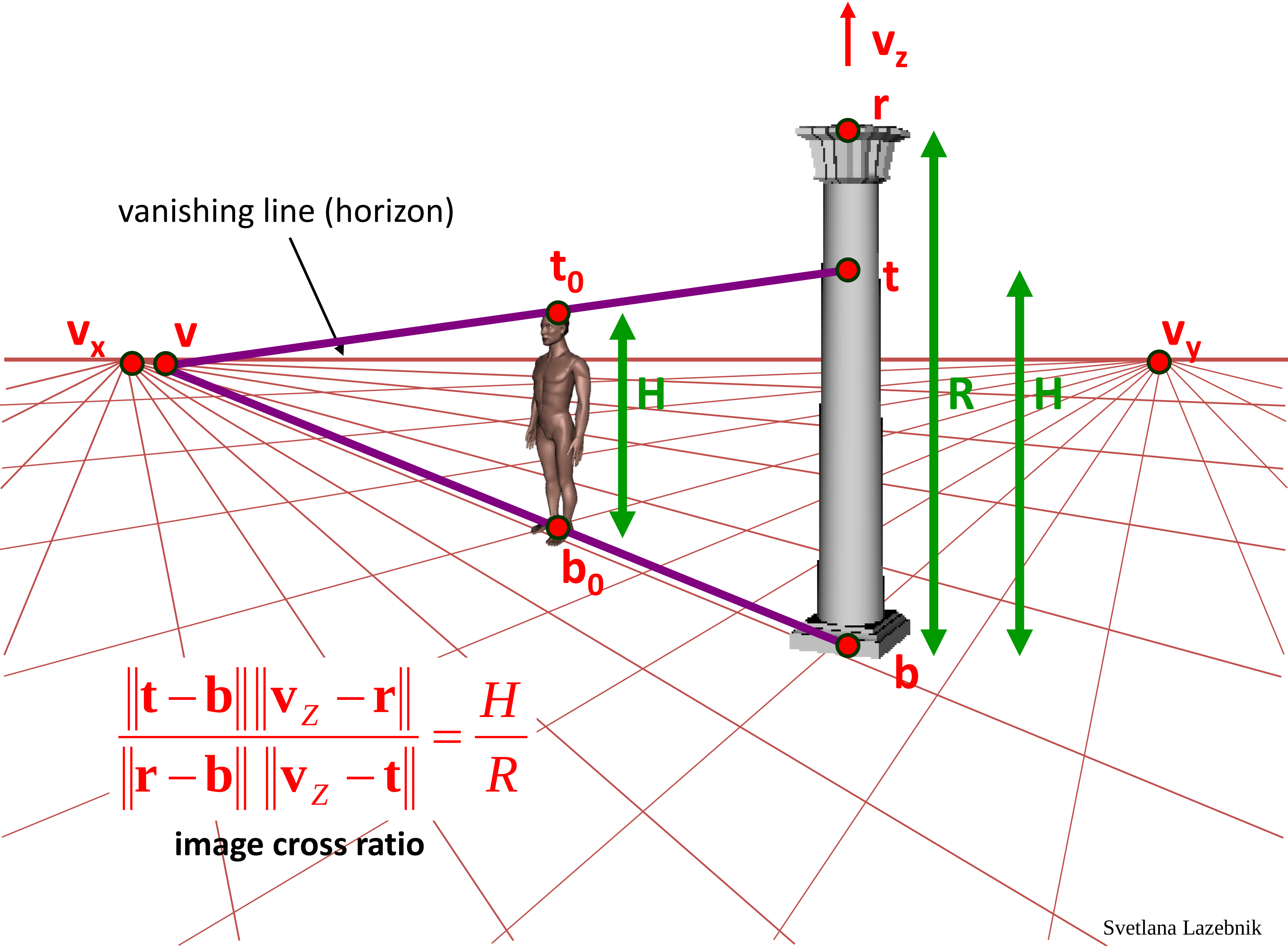
image cross ratio

Svetlana Lazebnik

How Tall is the Man in this Image?



Svetlana Lazebnik



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

image cross ratio

Svetlana Lazebnik

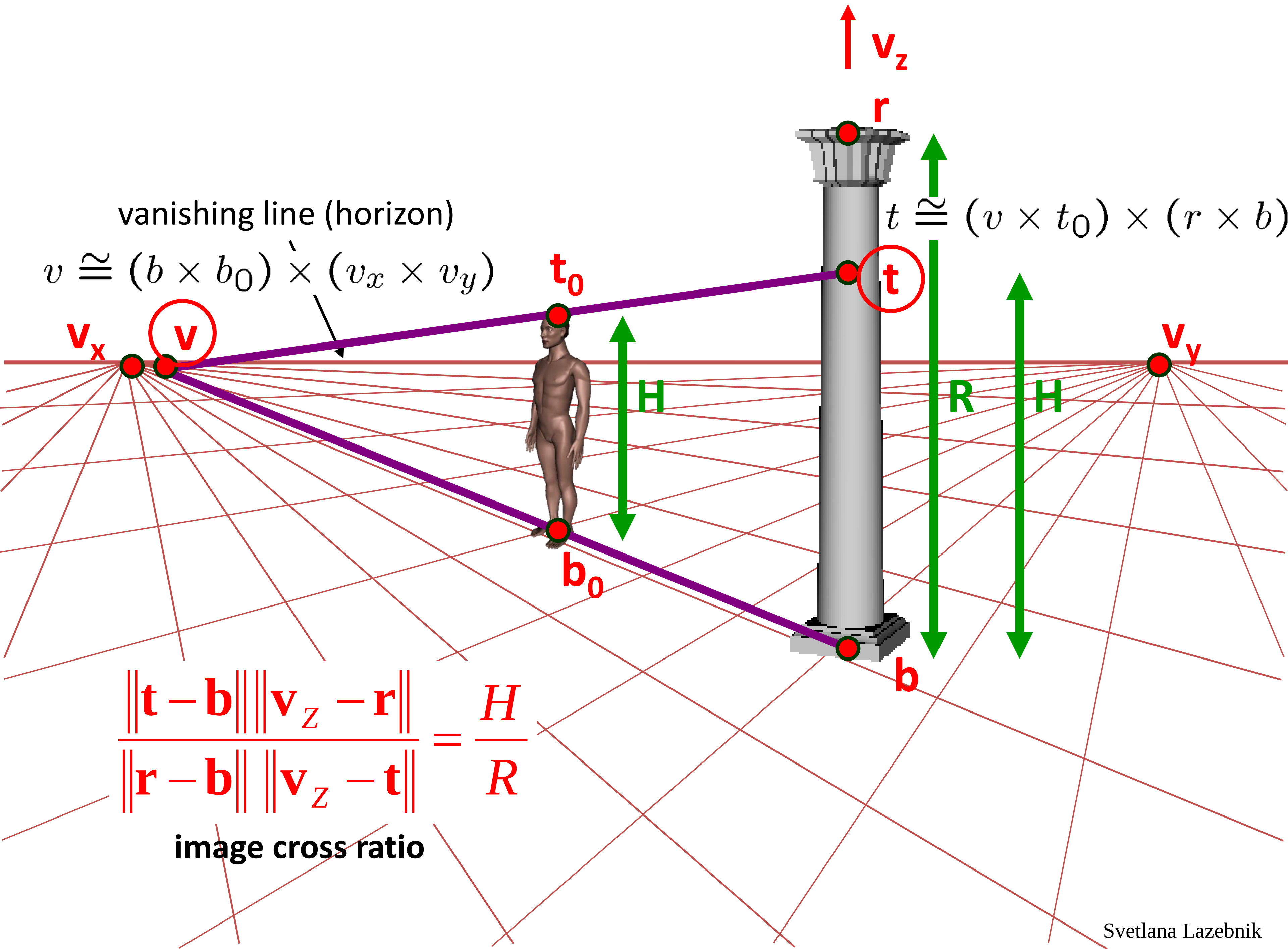
2D lines in homogeneous coordinates

- Line equation: $ax + by + c = 0$

$$\mathbf{l}^T\mathbf{x} = 0 \quad \text{where} \quad \mathbf{l} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

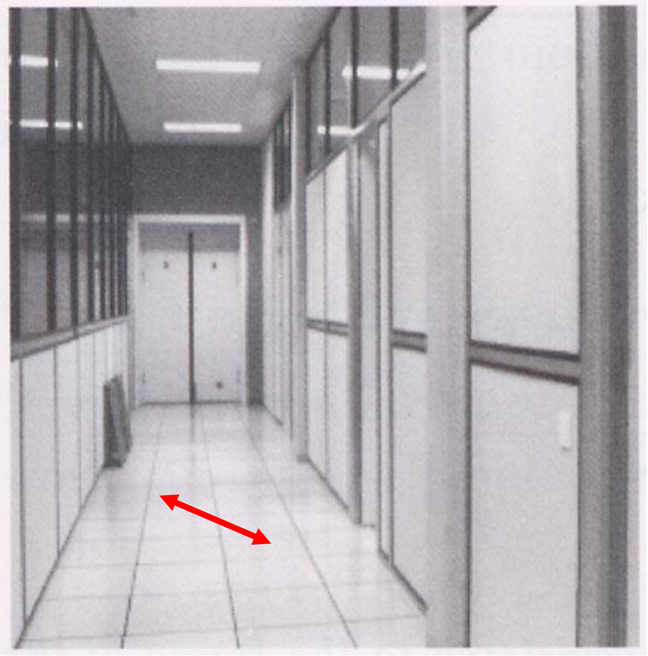
- Line passing through two points: $\mathbf{l} = \mathbf{x}_1 \times \mathbf{x}_2$

- Intersection of two lines: $\mathbf{x} = \mathbf{l}_1 \times \mathbf{l}_2$



v_z
r
vanishing line (horizon)
$t \cong (v \times t_0) \times (r \times b)$
$v \cong (b \times b_0) \times (v_x \times v_y)$
t_0
t
v_x
v
v_y
H
R
H
b_0
b
$$\frac{\|\mathbf{t}-\mathbf{b}\|\,\|\mathbf{v}_Z-\mathbf{r}\|}{\|\mathbf{r}-\mathbf{b}\|\,\|\mathbf{v}_Z-\mathbf{t}\|}=\frac{H}{R}$$
image cross ratio
Svetlana Lazebnik

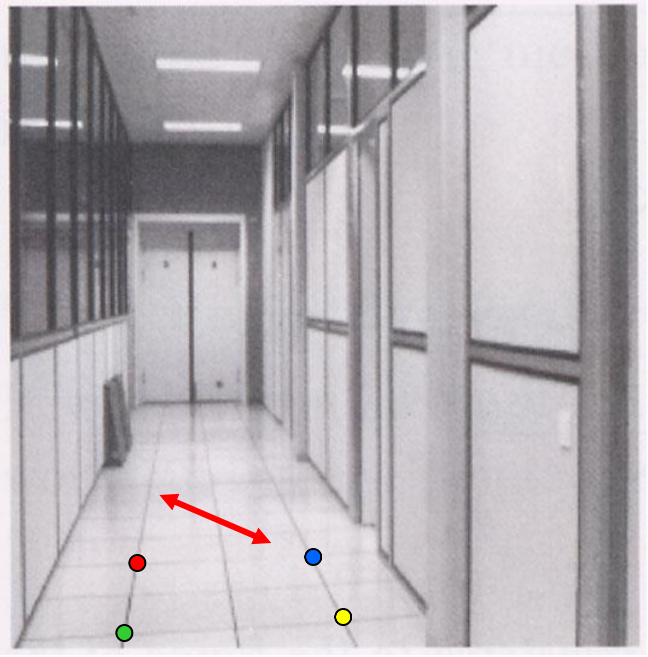
How Long is this Line Segment?



Can we measure distances between two points on same plane?

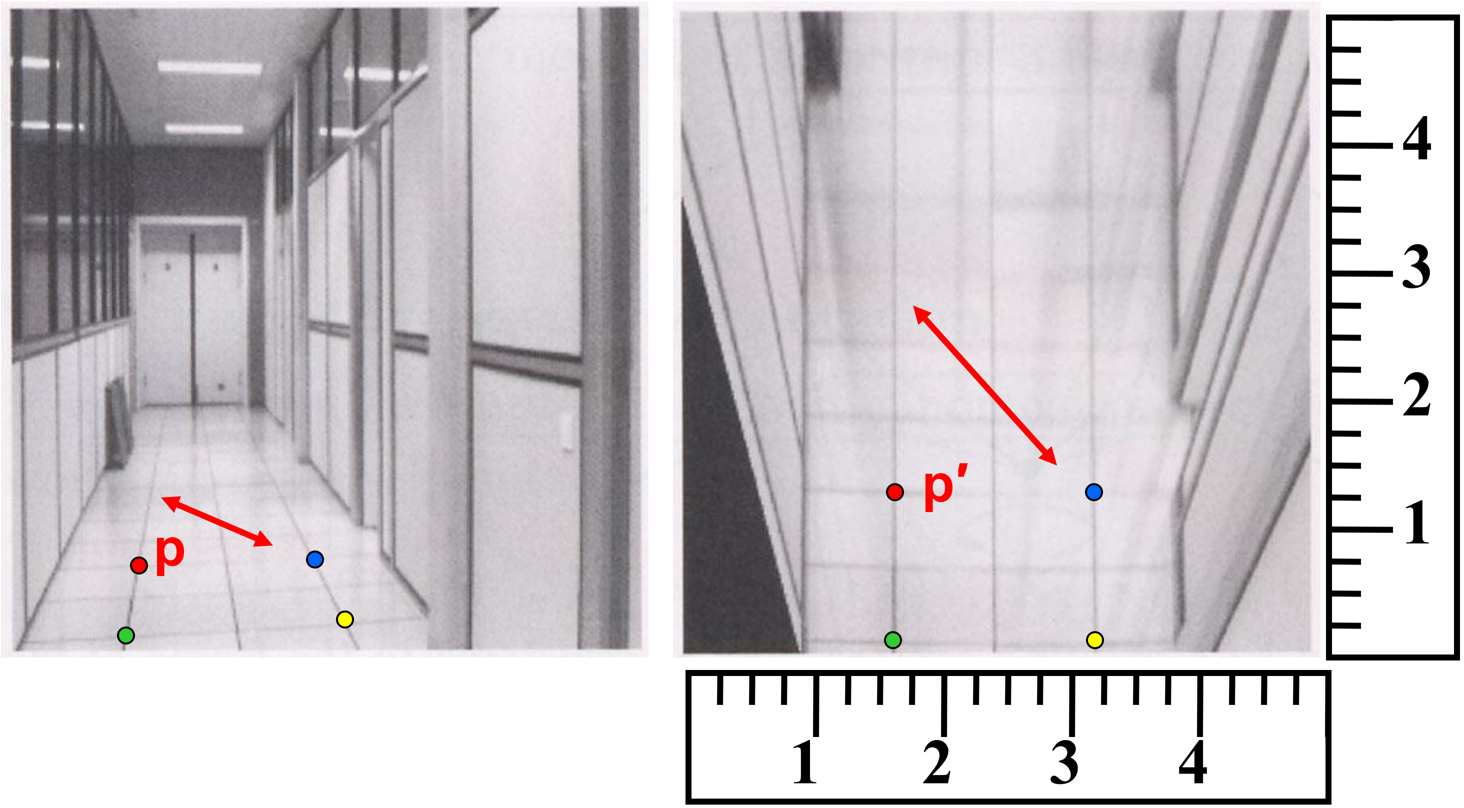
Svetlana Lazebnik

How Long is this Line Segment?



What if we know distances between some pairs of points on the same plane?

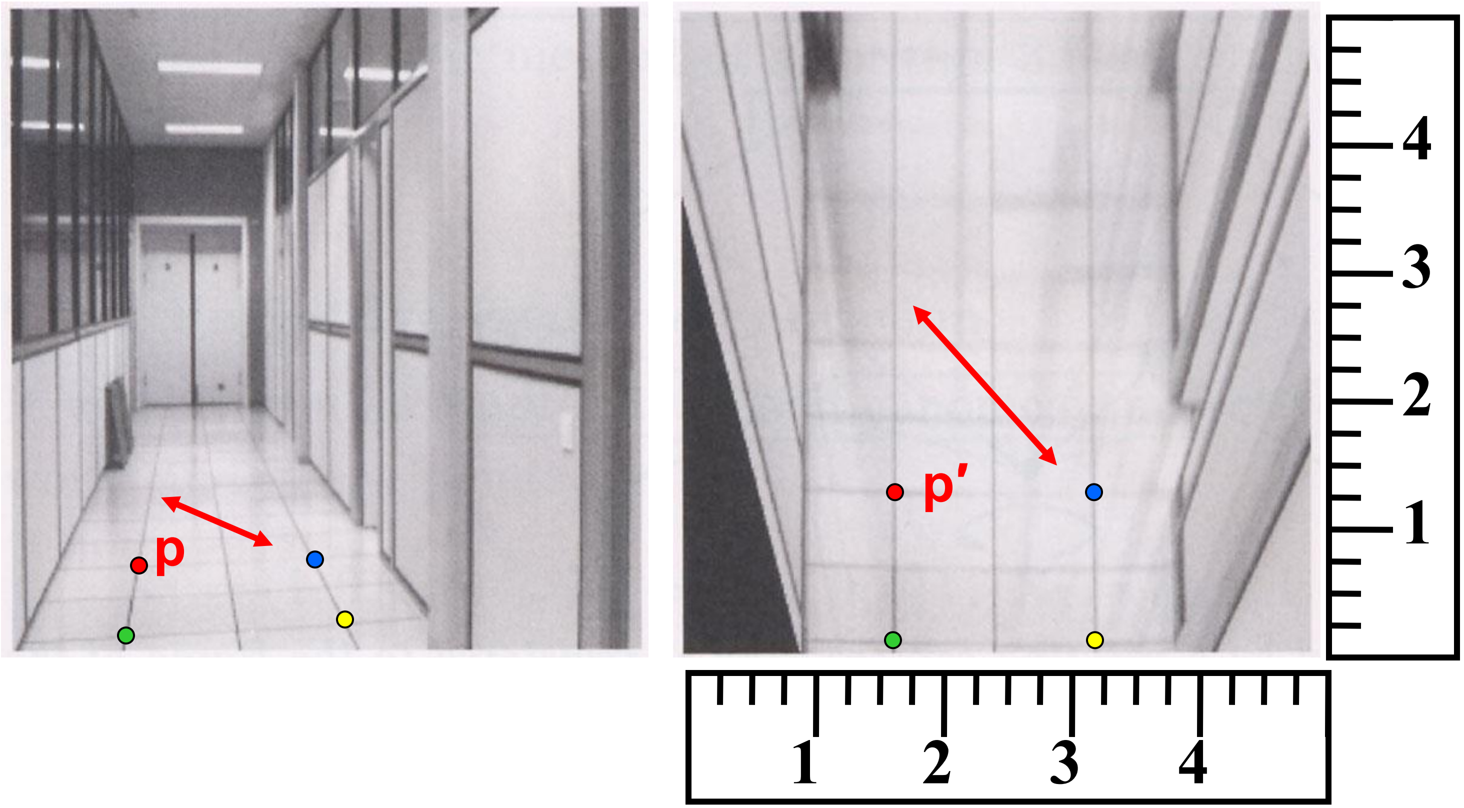
Measurements on planes



Approach: unwarp then measure

What kind of warp is this?

Measurements on planes

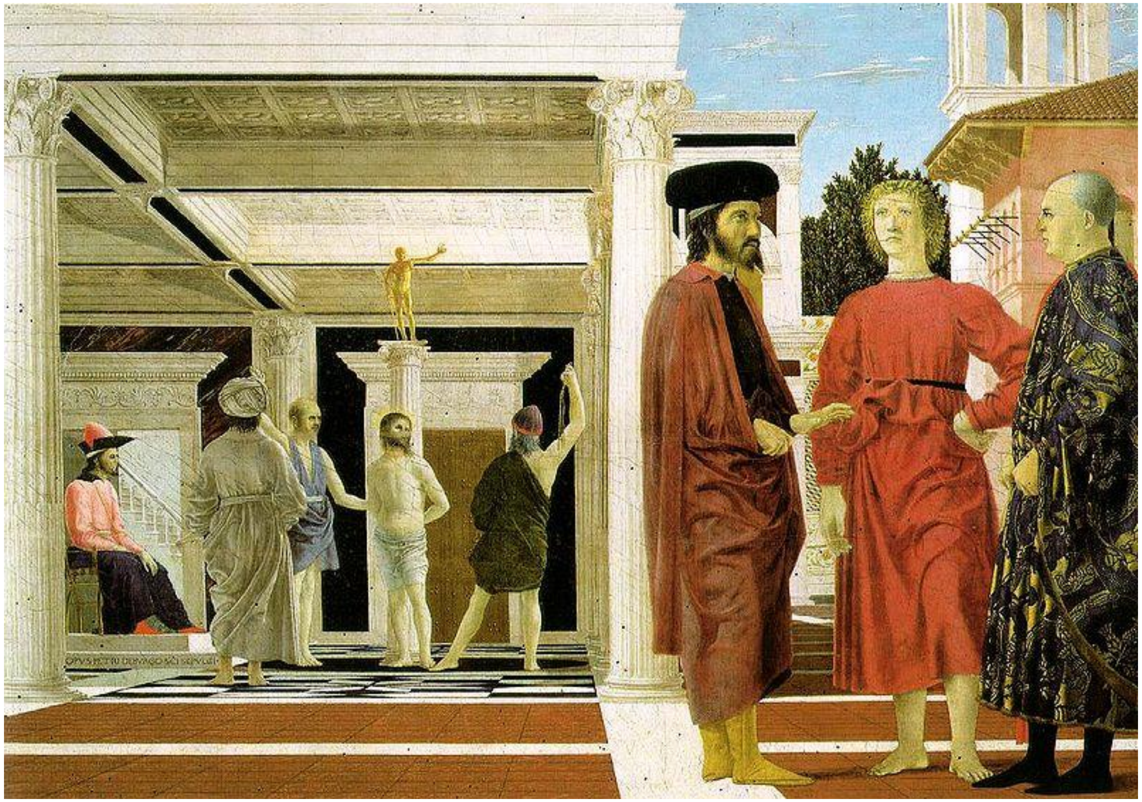


Approach: unwarp then measure

Homography!

Svetlana Lazebnik

Application: Image rectification



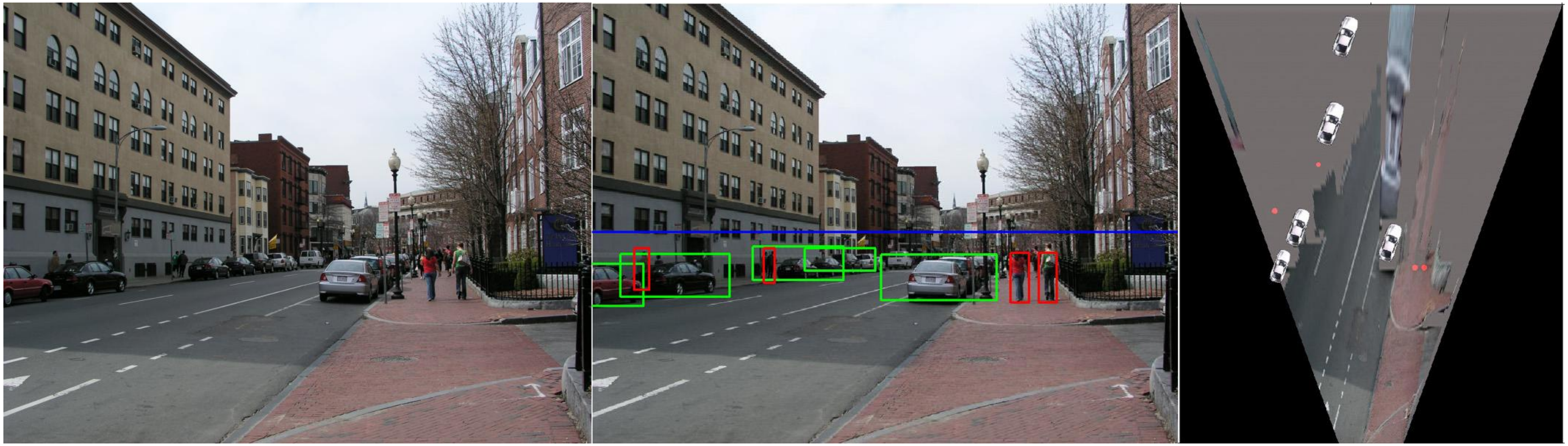
Piero della Francesca*, Flagellation,* ca. 1455

Application: Image editing

Inserting synthetic objects into images:
http://vimeo.com/28962540

K. Karsch and V. Hedau and D. Forsyth and D. Hoiem, “Rendering Synthetic Objects into Legacy Photographs,” *SIGGRAPH Asia* 2011

Application: Object recognition



D. Hoiem, A.A. Efros, and M. Hebert, "Putting Objects in Perspective", CVPR 2006

Svetlana Lazebnik

Outline

Projective geometry

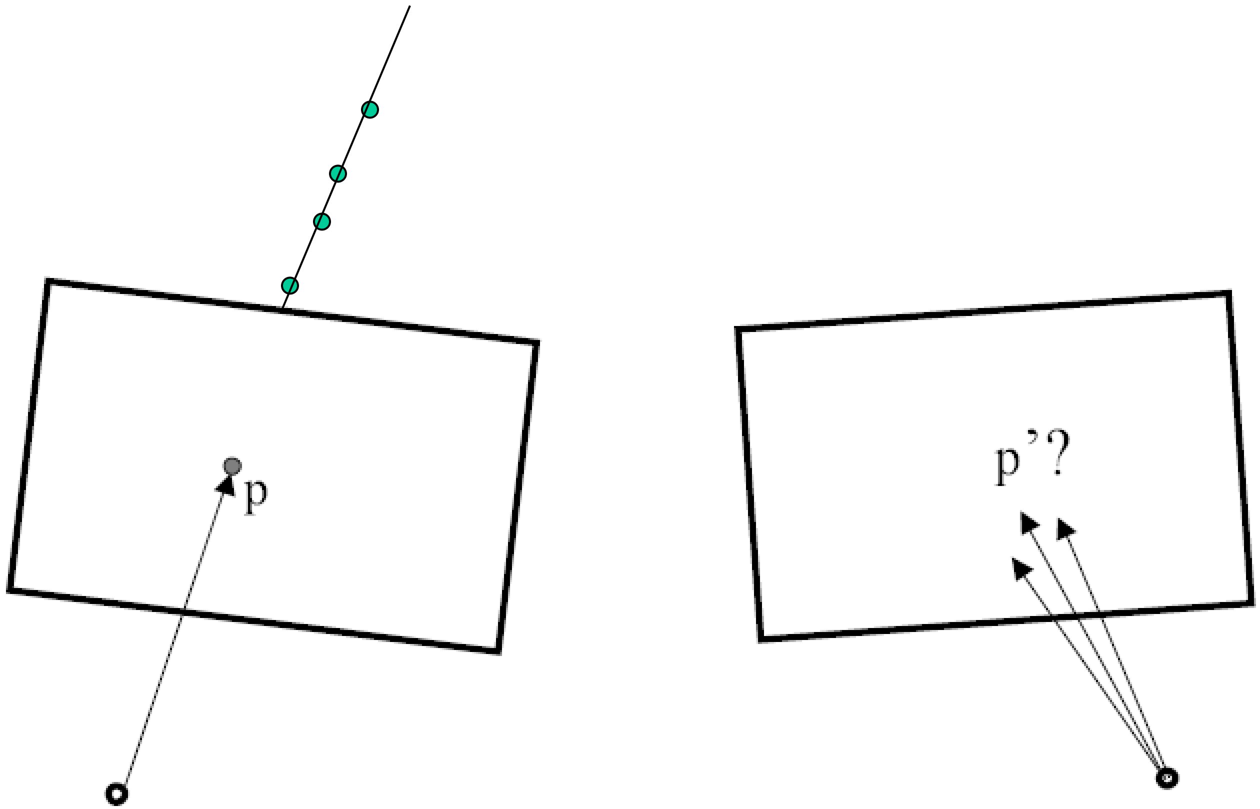
Vanishing points

- Application: camera calibration
- Application: single-view metrology

Epipolar geometry

- Application: stereo correspondence
- Application: structure from motion revisited

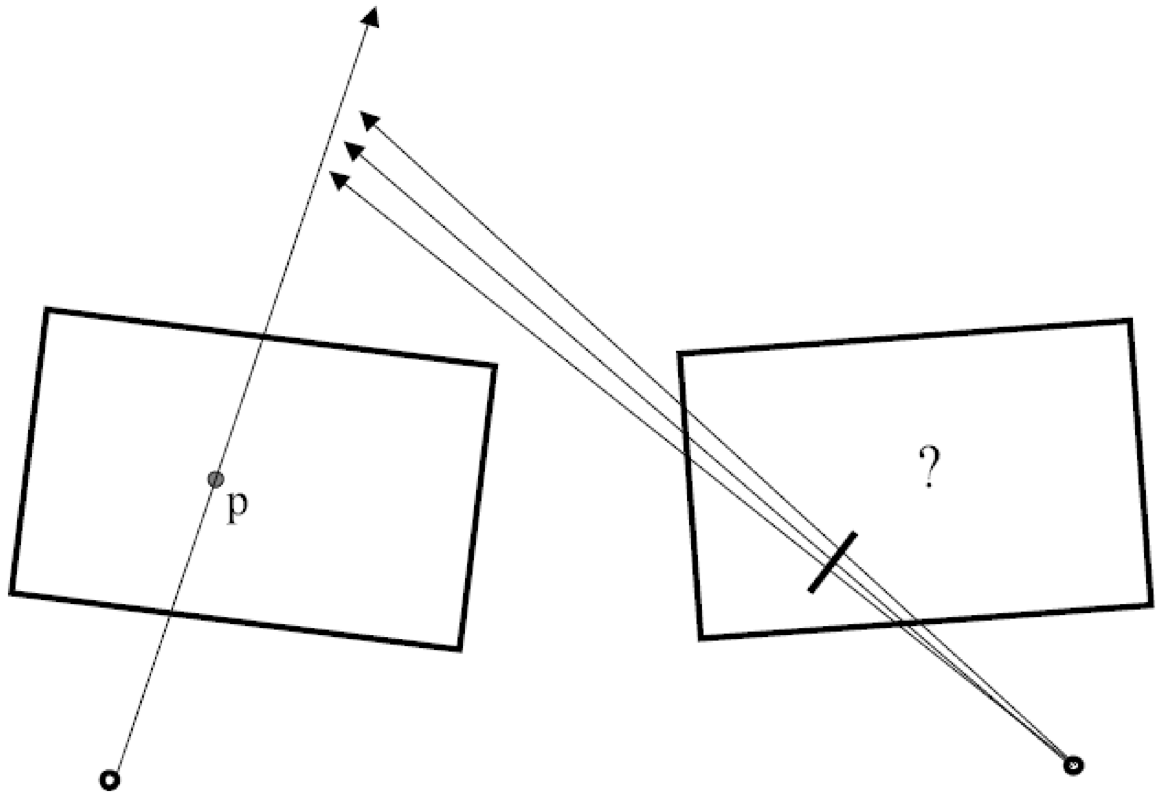
Correspondence constraints?



- Given p in left image, where can corresponding point p' be?

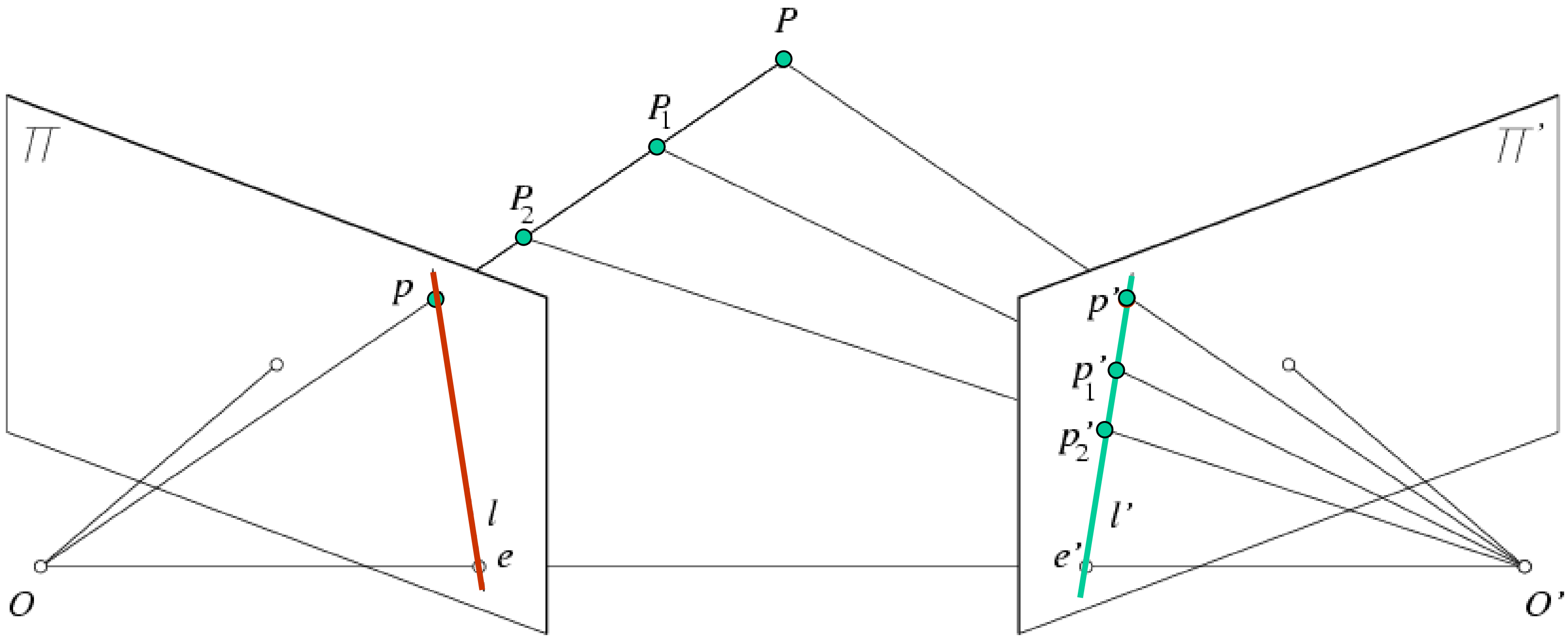
Kristen Grauman

Correspondence constraints?



Kristen Grauman

Epipolar constraint

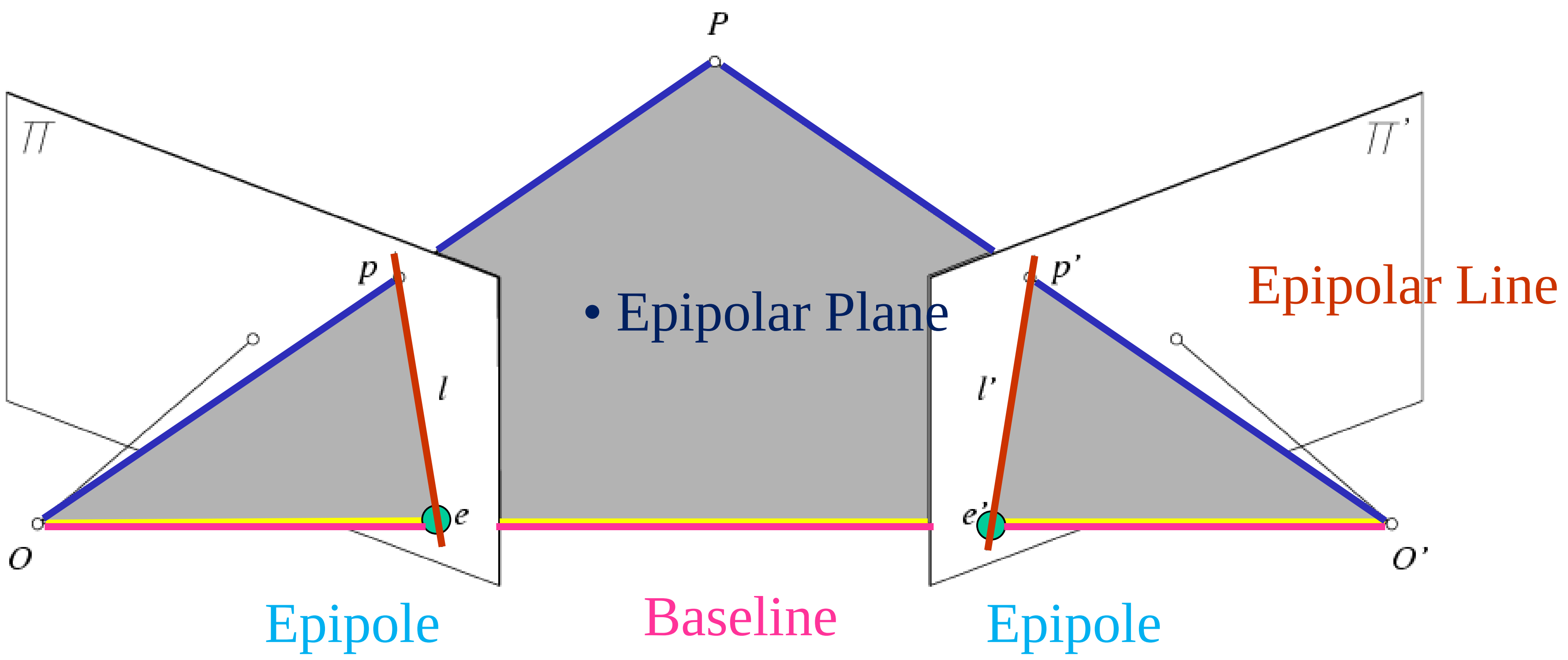


Geometry of two views constrains where the corresponding pixel for some image point in the first view must occur in the second view.

It must be on the line carved out by a plane connecting the world point and optical centers.

Kristen Grauman

Epipolar geometry



http://www.ai.sri.com/~luong/research/Meta3DViewer/EpipolarGeo.html

Kristen Grauman

Epipolar geometry: terms

Baseline: line joining the camera centers

Epipole: point of intersection of baseline with image plane

Epipolar plane: plane containing baseline and world point

Epipolar line: intersection of epipolar plane with the image plane

All epipolar lines intersect at the epipole

An epipolar plane intersects the left and right image planes in epipolar lines

Why is the epipolar constraint useful?

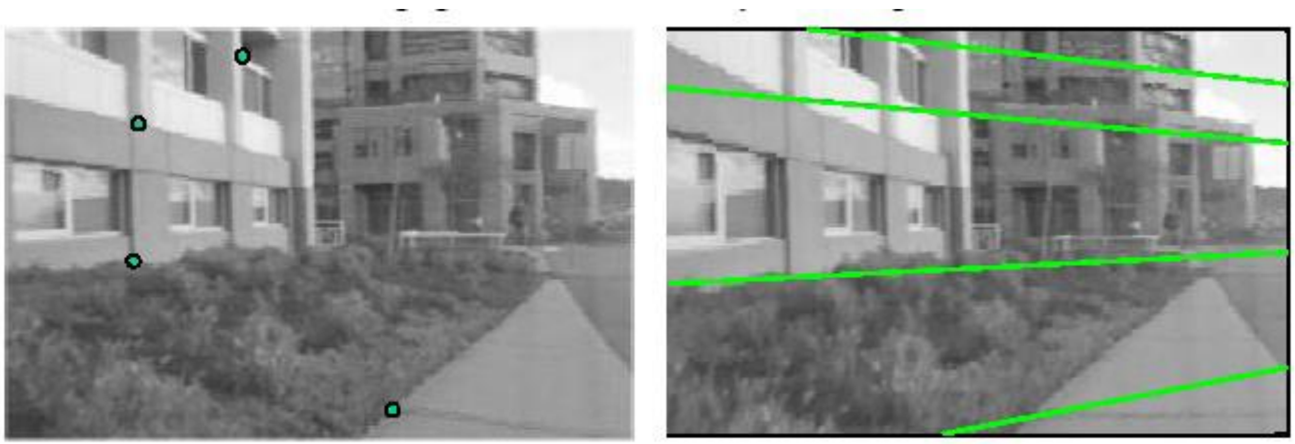
Kristen Grauman

Epipolar constraint

This is useful because it reduces the correspondence problem to a 1D search along an epipolar line.

Image from Andrew Zisserman

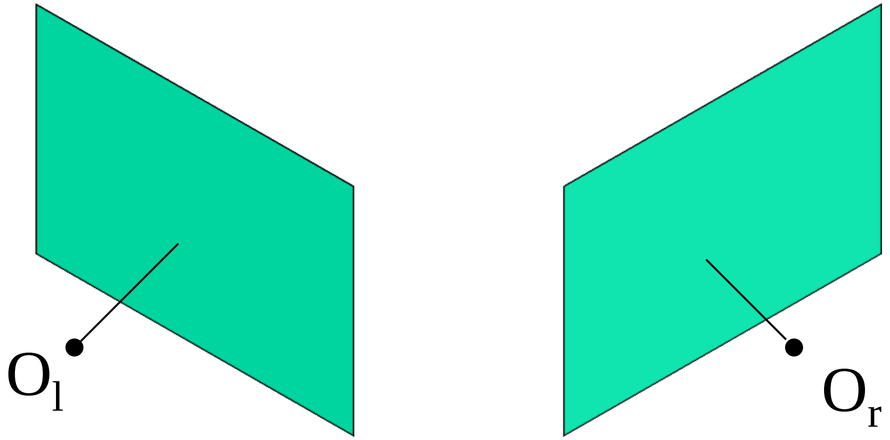
Example



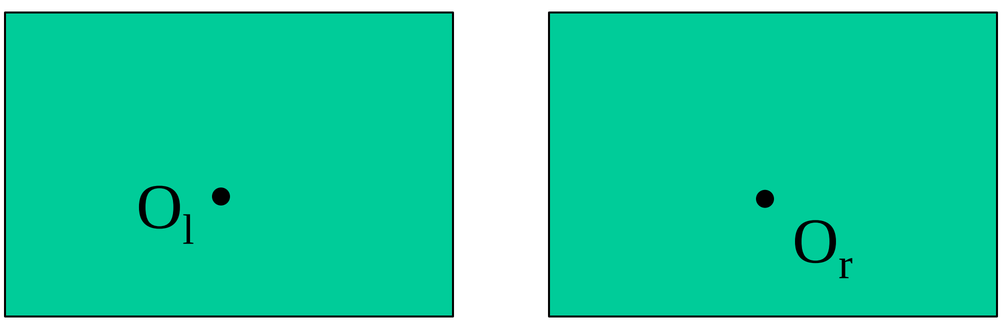
Kristen Grauman

What do the epipolar lines look like?

1.



2.



Kristen Grauman

Epipolar lines: converging cameras

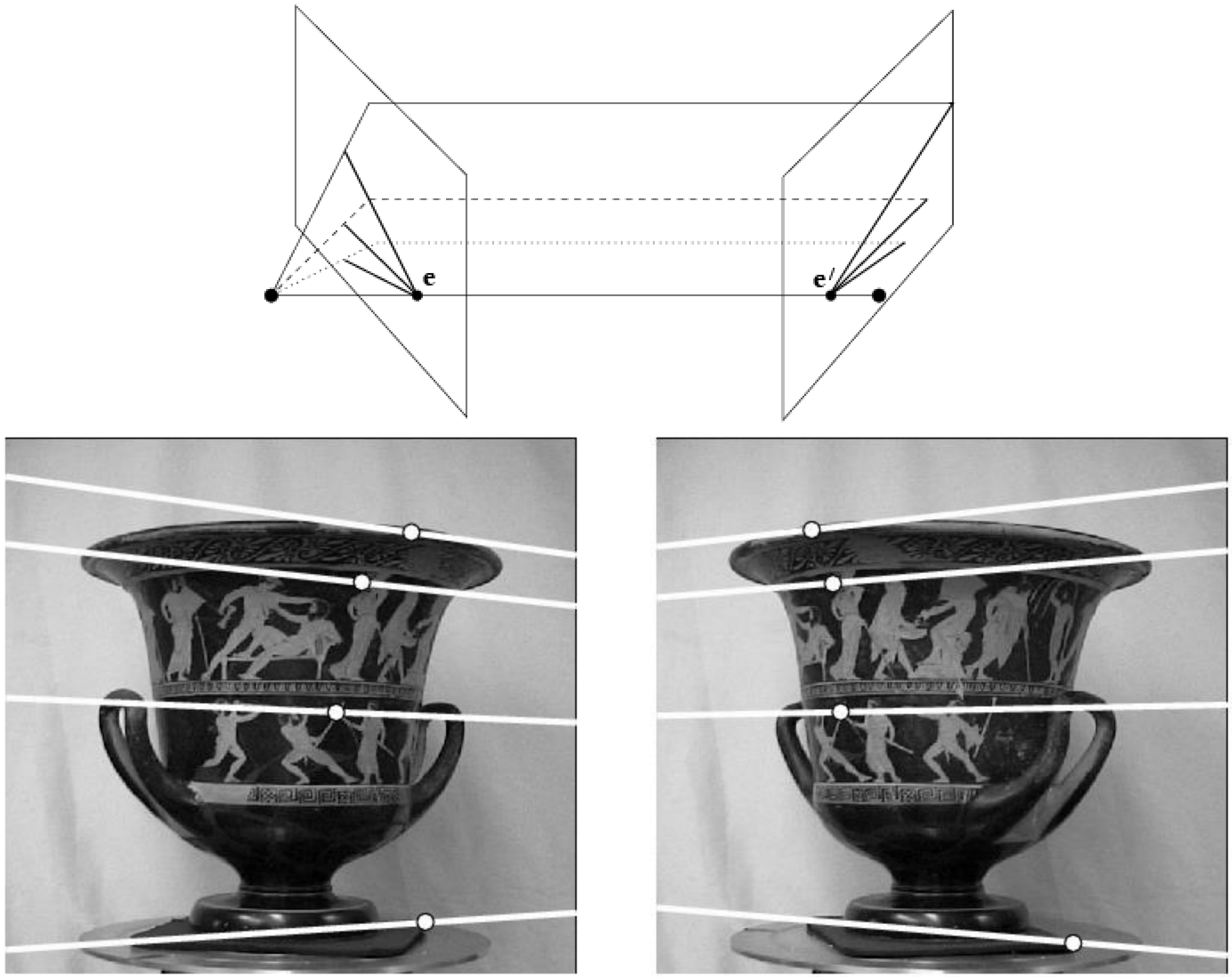
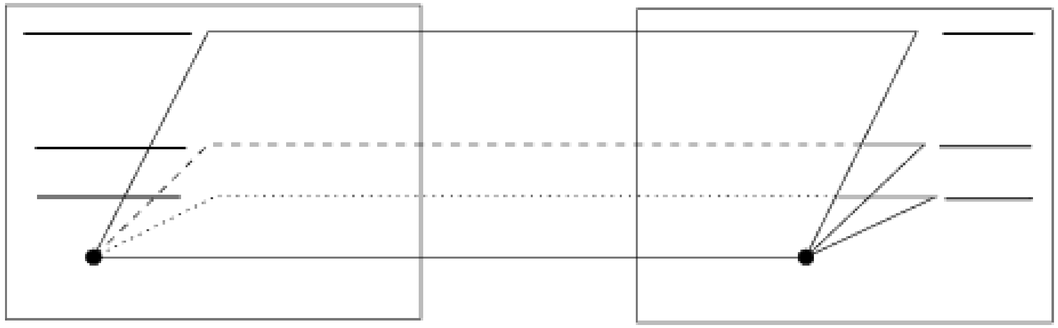


Figure from Hartley & Zisserman

Epipolar lines: parallel cameras



Where are the epipoles?

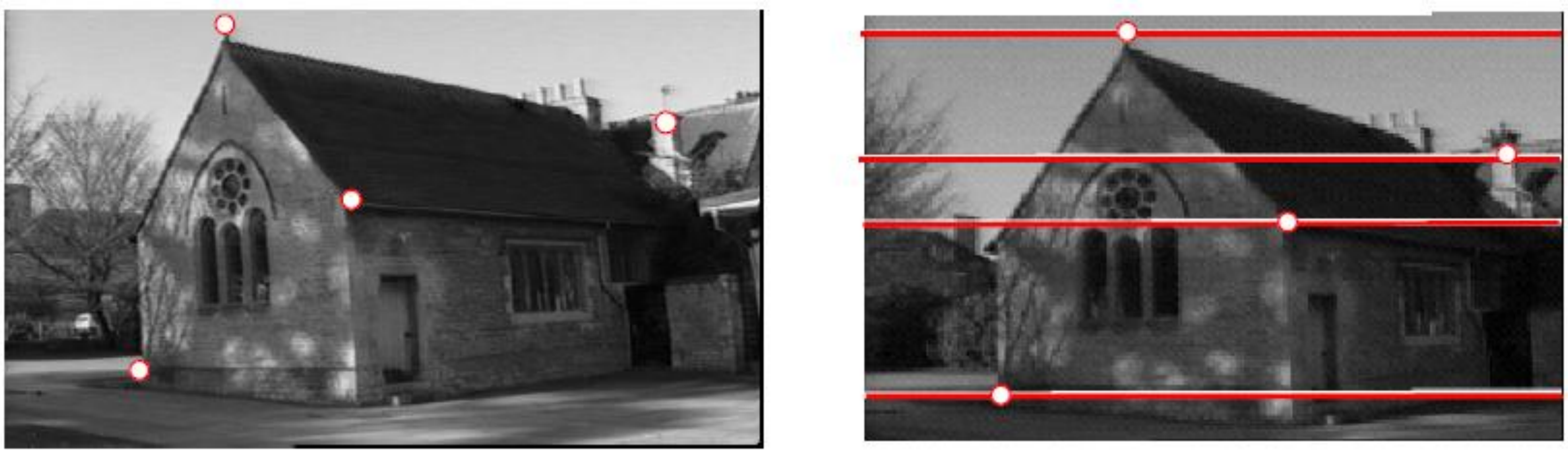


Figure from Hartley & Zisserman

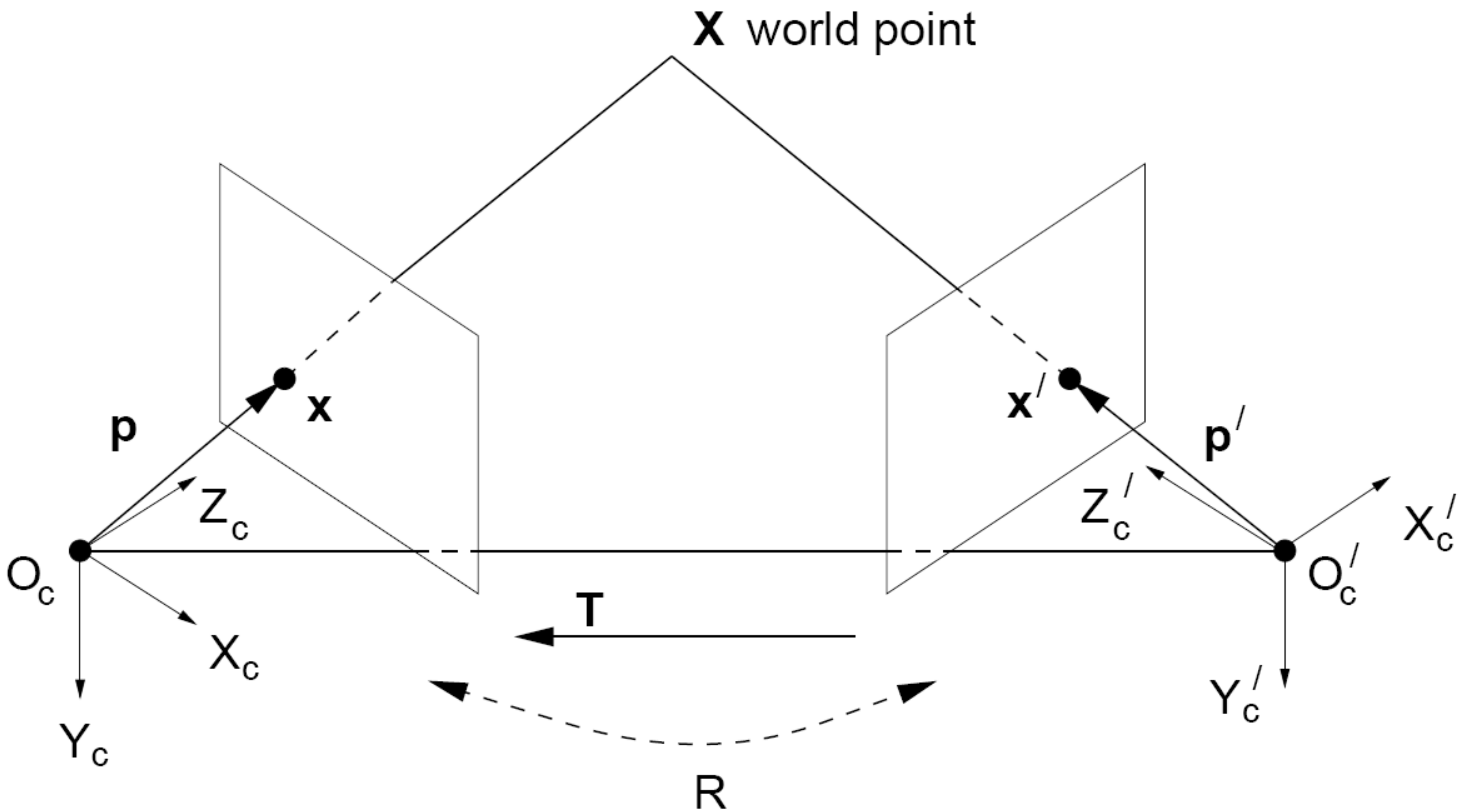
Epipolar lines

So far, we have the explanation in terms of geometry.

Now, how to express the epipolar constraints algebraically?

Kristen Grauman

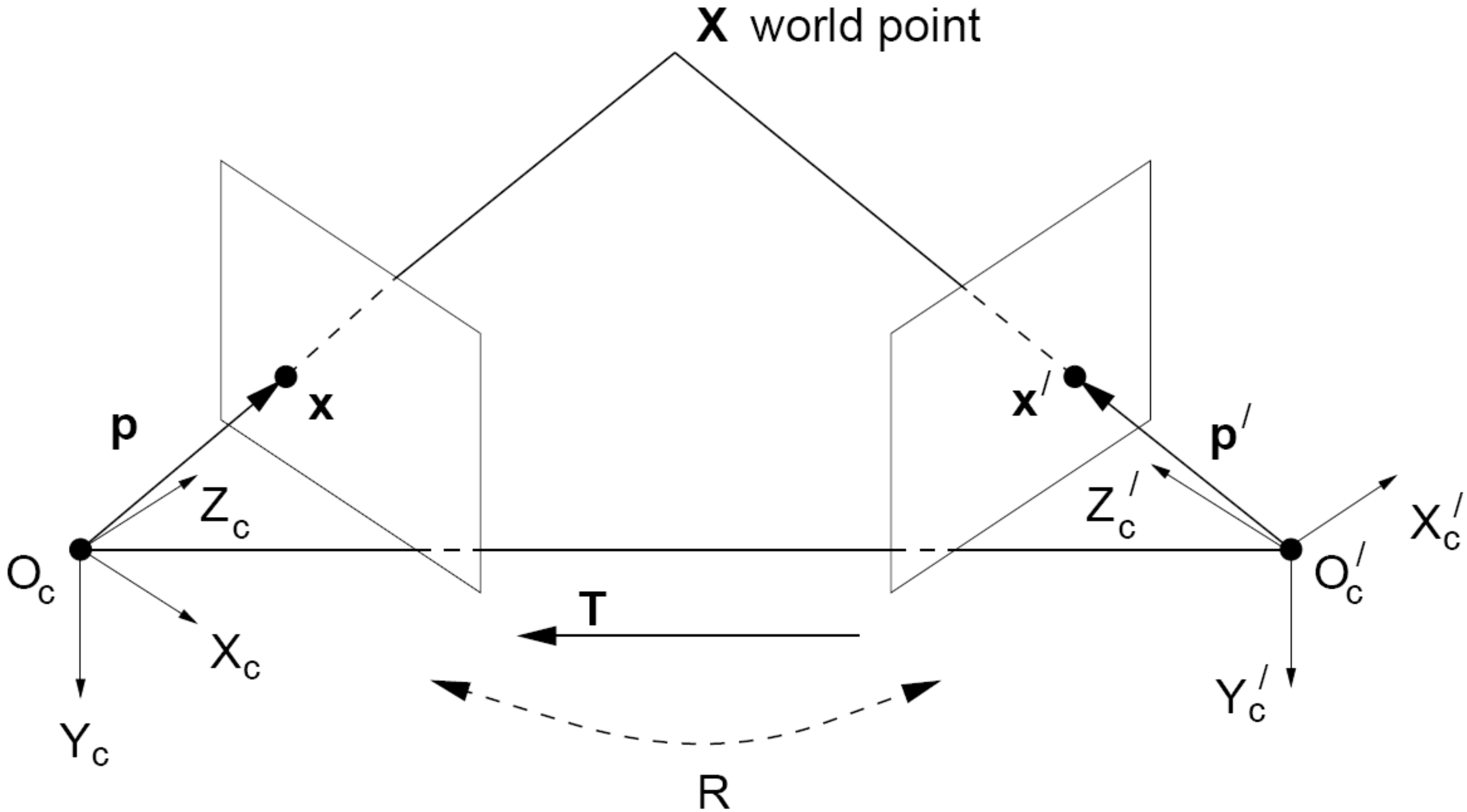
Stereo geometry



Main idea

Kristen Grauman

Stereo geometry



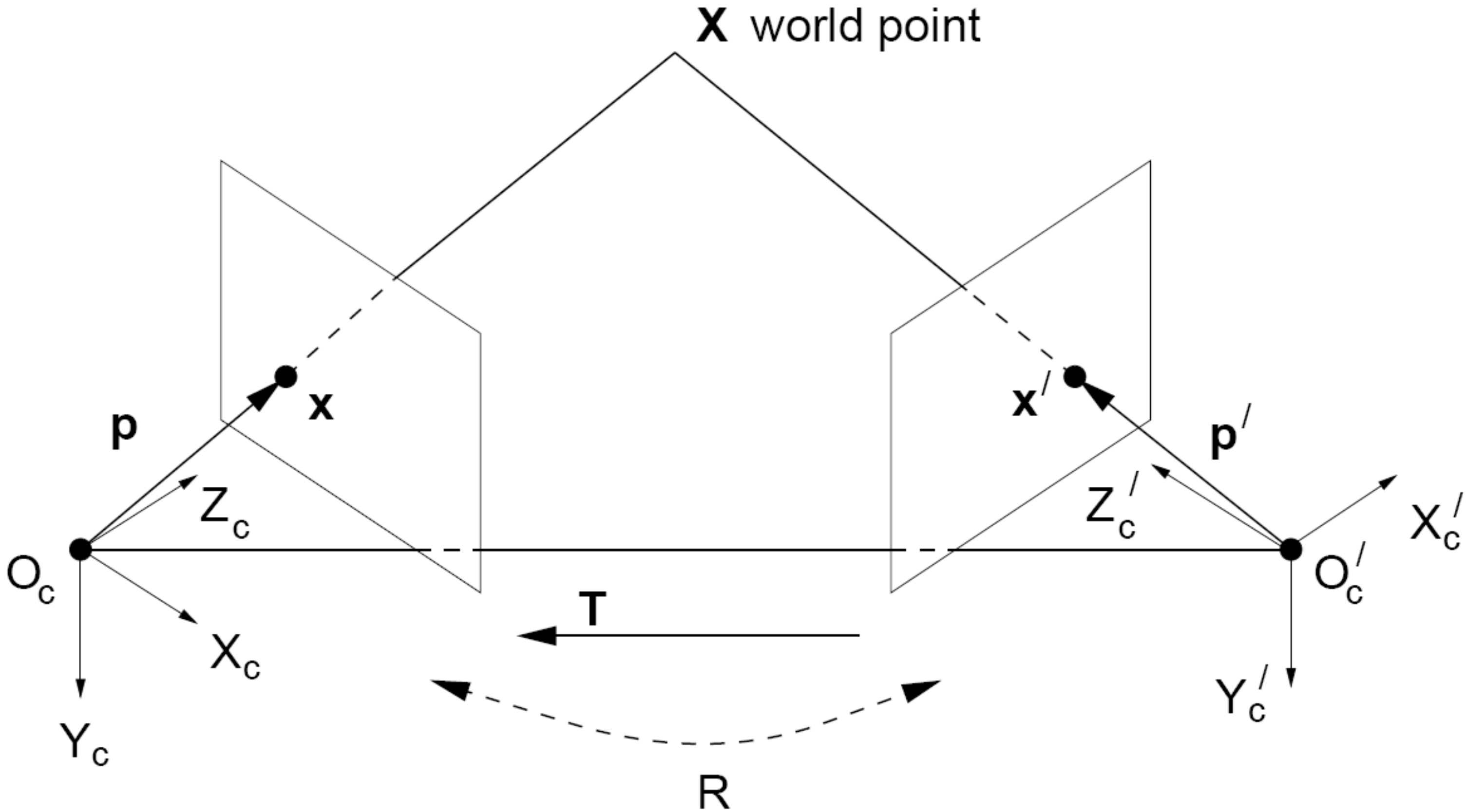
If the stereo rig is calibrated, we know :

how to **rotate** and **translate** camera reference frame 1 to get to camera reference frame 2.

Rotation: 3 x 3 matrix **R**; translation: 3 vector **T**.

Kristen Grauman

Stereo geometry



If the stereo rig is calibrated, we know :

how to **rotate** and **translate** camera reference frame 1 to get to camera reference frame 2.

$$\mathbf{X}'_c = \mathbf{R}\mathbf{X}_c + \mathbf{T}$$

Kristen Grauman

An aside: cross product

$$\vec{a} \times \vec{b} = \vec{c}$$

$$\vec{a} \cdot \vec{c} = 0$$

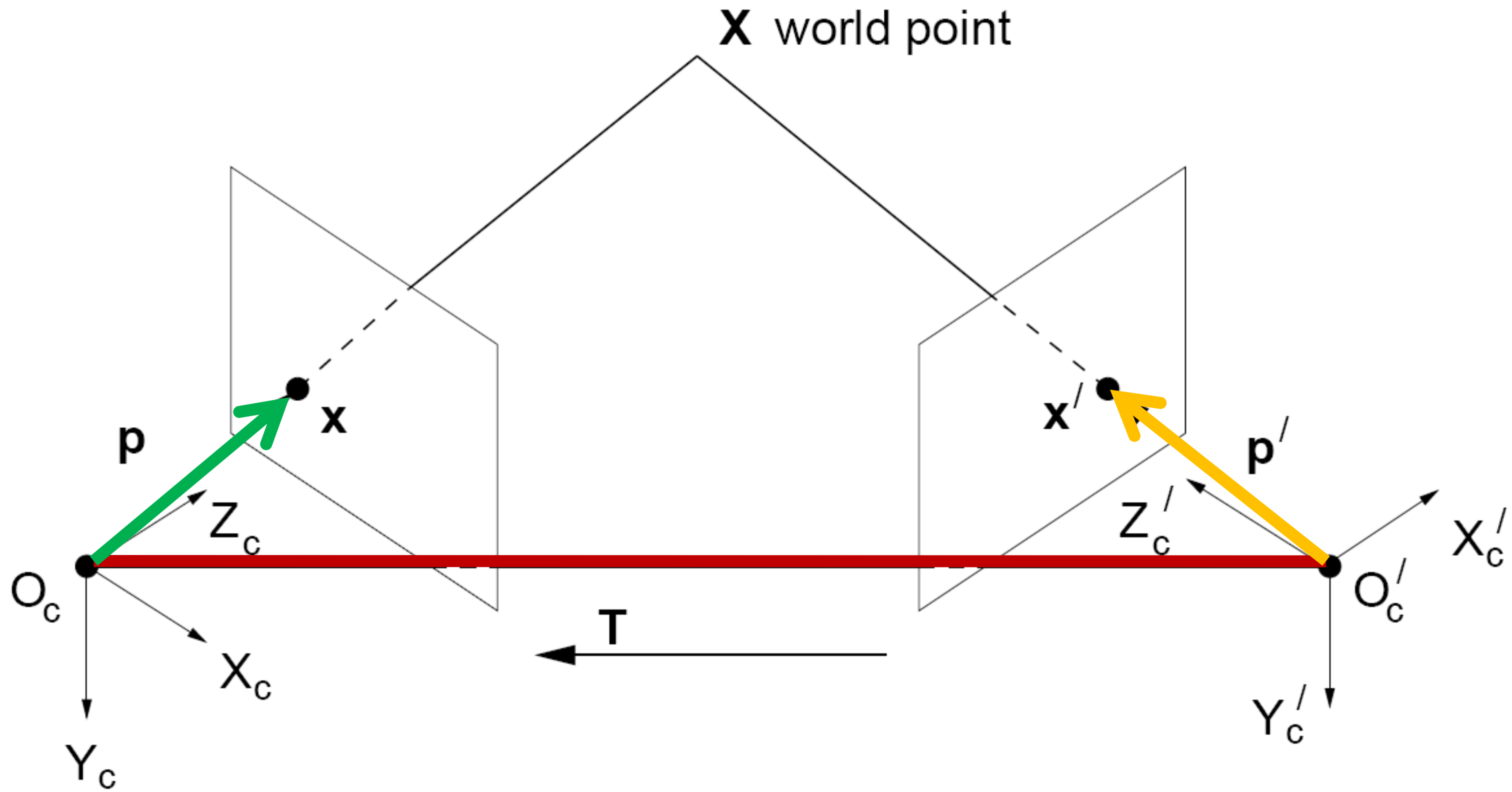
$$\vec{b} \cdot \vec{c} = 0$$

Vector cross product takes two vectors and returns a third vector that's perpendicular to both inputs.

So here, c is perpendicular to both a and b, which means the dot product = 0.

Kristen Grauman

From geometry to algebra



$$\boxed{\mathbf{X'}} = \mathbf{R}\boxed{\mathbf{X}} + \boxed{\mathbf{T}}$$

$$\underbrace{\mathbf{T} \times \mathbf{X'}}_{\text{Normal to the plane}} =$$

$$= \mathbf{T} \times \mathbf{R}\mathbf{X}$$

$$\mathbf{X'} \cdot (\mathbf{T} \times \mathbf{X'}) = \mathbf{X'} \cdot (\mathbf{T} \times \mathbf{R}\mathbf{X})$$

$$= 0$$

Kristen Grauman

Another aside: Matrix form of cross product

$$\vec{a} \times \vec{b} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \vec{c}$$

$$\vec{a} \cdot \vec{c} = 0$$

$$\vec{b} \cdot \vec{c} = 0$$

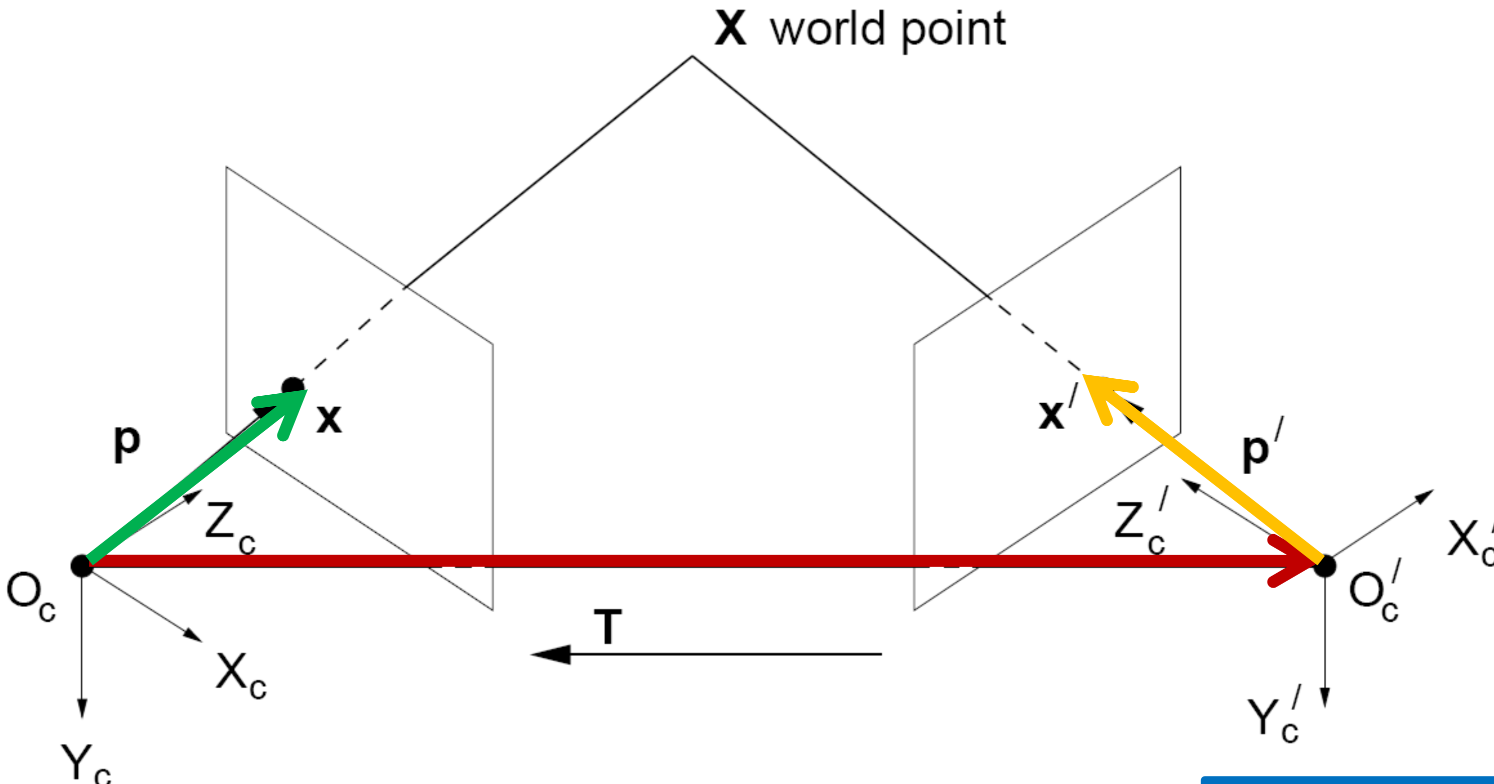
Can be expressed as a matrix multiplication.

$$[a_x] = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$$

$$\boxed{\vec{a} \times \vec{b} = [a_x]\vec{b}}$$

Kristen Grauman

From geometry to algebra



$$\mathbf{X'} = \mathbf{RX} + \mathbf{T}$$

$$\underbrace{\mathbf{T} \times \mathbf{X}'}_{\text{Normal to the plane}} = \mathbf{T} \times \mathbf{RX} + \mathbf{T} \times \mathbf{T}$$

$$= \mathbf{T} \times \mathbf{RX}$$

$$\mathbf{X}' \cdot (\mathbf{T} \times \mathbf{X}') \boxed{= \mathbf{X}' \cdot (\mathbf{T} \times \mathbf{RX}) \\ = 0}$$

Kristen Grauman

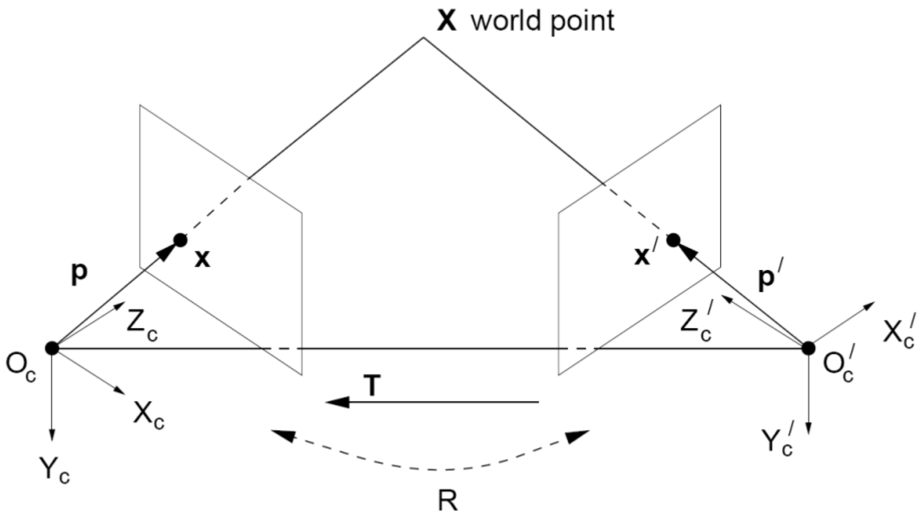
Essential matrix

$$\boxed{\mathbf{X}' \cdot \left(\mathbf{T} \times \mathbf{R}\mathbf{X}\right) = 0}$$

$$\mathbf{X}' \cdot \left([\mathrm{T}_x]\mathbf{R}\mathbf{X}\right) = 0$$

Let $\mathbf{E} = [\mathrm{T}_x]\mathbf{R}$

$$\mathbf{X}'^T \mathbf{E}\mathbf{X} = 0$$



E is called the **essential matrix**, and it relates corresponding image points between both cameras, given the rotation and translation.

If we observe a point in one image, its position in other image is constrained to lie on line defined by above.

Note: these points are in **camera coordinate systems**.

Kristen Grauman

Fundamental matrix

Relates **pixel coordinates** in the two views

$$\boxed{\mathbf{p}_{im,right}^{\mathrm{T}}\mathbf{F}\mathbf{p}_{im,left} = 0}$$

More general form than essential matrix: we remove need to know intrinsic parameters

If we estimate fundamental matrix from correspondences in *pixel coordinates*, can reconstruct epipolar geometry without intrinsic or extrinsic parameters.

Fundamental Matrix

$$\boxed{\mathbf{p}_{c,right}{}^{\mathrm{T}}\mathbf{E}\mathbf{p}_{c,left} = 0}$$

From before, the **essential** matrix **E.**

$$\left(\mathbf{M}_{int,right}^{-1}\mathbf{p}_{im,right}\right)^{\mathrm{T}}\mathbf{E}\left(\mathbf{M}_{int,left}^{-1}\mathbf{p}_{im,left}\right) = 0$$

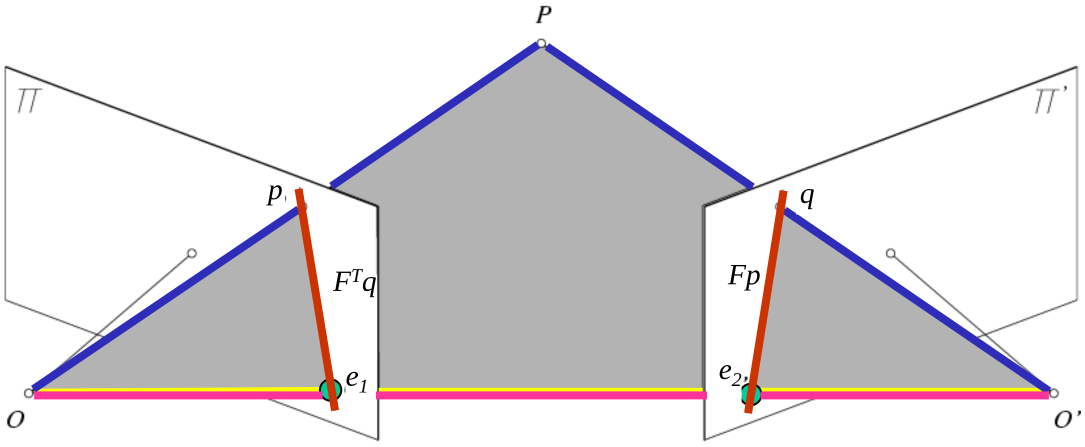
$$\mathbf{p}_{im,right}^{\mathrm{T}}\underbrace{\left(\mathbf{M}_{int,right}^{-\mathrm{T}}\mathbf{E}\mathbf{M}_{int,left}^{-1}\right)}_{\mathbf{F}}\mathbf{p}_{im,left} = 0$$

“Fundamental matrix”

$$\boxed{\mathbf{p}_{im,right}^{\mathrm{T}}\mathbf{F}\mathbf{p}_{im,left} = 0}$$

Properties of the Fundamental Matrix

- $\mathbf{F}\mathbf{p}$ is the epipolar line associated with $\mathbf{p}$
- $\mathbf{F}^T\mathbf{q}$ is the epipolar line associated with $\mathbf{q}$
- $\mathbf{F}\mathbf{e}_1 = \mathbf{0}$ and $\mathbf{F}^T\mathbf{e}_2 = \mathbf{0}$
- $\mathbf{F}$ is rank 2



Estimating the Fundamental Matrix

If we don't know $\mathbf{K}_1$, $\mathbf{K}_2$, $\mathbf{R}$, or $\mathbf{t}$, can we estimate $\mathbf{F}$ for two images?

Yes, given enough correspondences

Estimating F – 8-point algorithm

The fundamental matrix F is defined by

$$\mathbf{x}'^{\mathrm{T}}\mathbf{F}\mathbf{x} = 0$$

for any pair of matches x and x' in two images.

- Let x=($u,v,1$)$^{\mathrm{T}}$ and x'=($u',v',1$)$^{\mathrm{T}}$,

$$\mathbf{F} = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix}$$

each match gives a linear equation

$$uu' f_{11} + vu' f_{12} + u' f_{13} + uv' f_{21} + vv' f_{22} + v' f_{23} + uf_{31} + vf_{32} + f_{33} = 0$$

Estimating F – 8-point algorithm

$$\begin{bmatrix} u_1u_1' & v_1u_1' & u_1' & u_1v_1' & v_1v_1' & v_1' & u_1 & v_1 & 1 \\ u_2u_2' & v_2u_2' & u_2' & u_2v_2' & v_2v_2' & v_2' & u_2 & v_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_nu_n' & v_nu_n' & u_n' & u_nv_n' & v_nv_n' & v_n' & u_n & v_n & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0$$

In reality, instead of solving $\mathbf{Af} = 0$, we seek $\mathbf{f}$ to minimize $\|\mathbf{Af}\|$, least eigenvector of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$.

Estimating F – 8-point algorithm

F should have rank 2

To enforce that **F** is of rank 2, F is replaced by F' that minimizes $\left\|\mathbf{F}-\mathbf{F}'\right\|$ subject to the rank constraint.

- This is achieved by SVD. Let $\mathbf{F}=\mathbf{U}\Sigma\mathbf{V}^{\mathrm{T}}$, where

$$\Sigma=\begin{bmatrix}\sigma_1 & 0 & 0\\ 0 & \sigma_2 & 0\\ 0 & 0 & \sigma_3\end{bmatrix}, \text{ let } \quad \Sigma'=\begin{bmatrix}\sigma_1 & 0 & 0\\ 0 & \sigma_2 & 0\\ 0 & 0 & 0\end{bmatrix}$$

then $\mathbf{F}'=\mathbf{U}\Sigma'\mathbf{V}^{\mathrm{T}}$ is the solution.

Estimating F – 8-point algorithm

Pros: it is linear, easy to implement and fast

Cons: susceptible to noise

Outline

Projective geometry

Vanishing points

- Application: camera calibration
- Application: single-view metrology

Epipolar geometry

- Application: stereo correspondence
- Application: structure from motion revisited

Stereo Correspondence

Goal:

- Find correspondences (pairs of points (u’,v’) $\leftrightarrow$ (u,v)).

1) Find interest points in image
2) Compute correspondences
3) Compute epipolar geometry
4) Refine

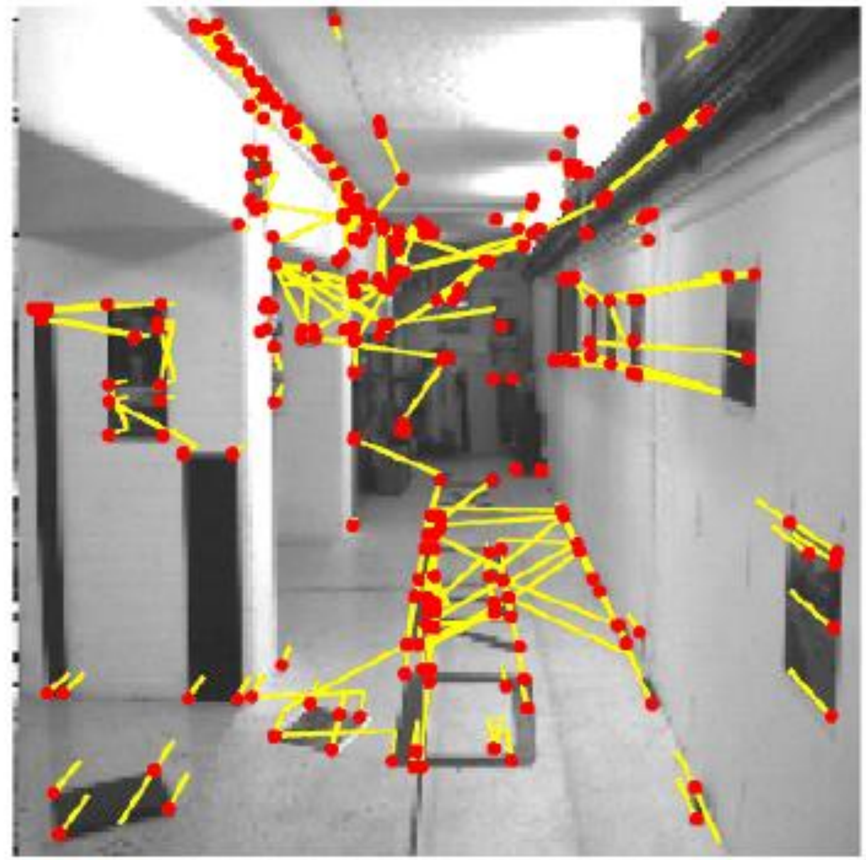
Example from Andrew Zisserman

Stereo Correspondence

1) Find interest points


Stereo Correspondence

2) Match points within proximity to get putative matches



Stereo Correspondence

3) Compute epipolar geometry -- robustly with RANSAC

Select random sample of putative correspondences

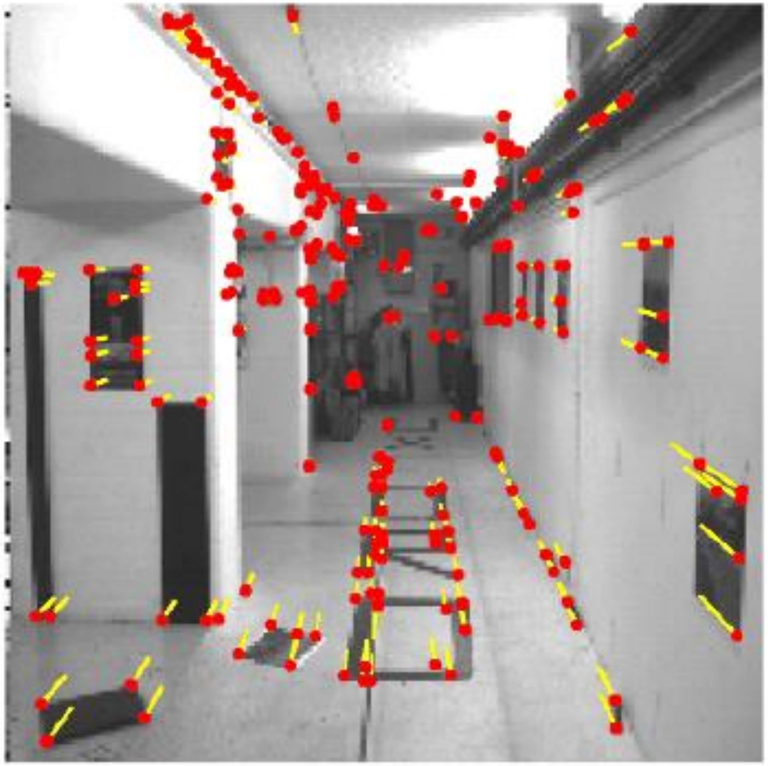
Compute **F** using them

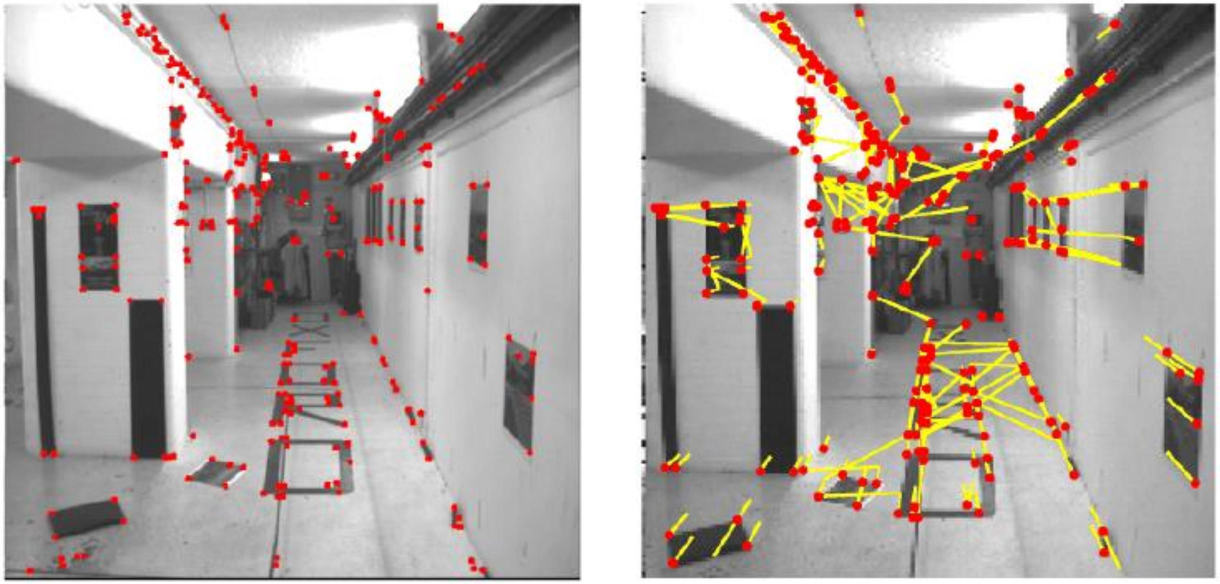
- determines epipolar constraint

Evaluate amount of support

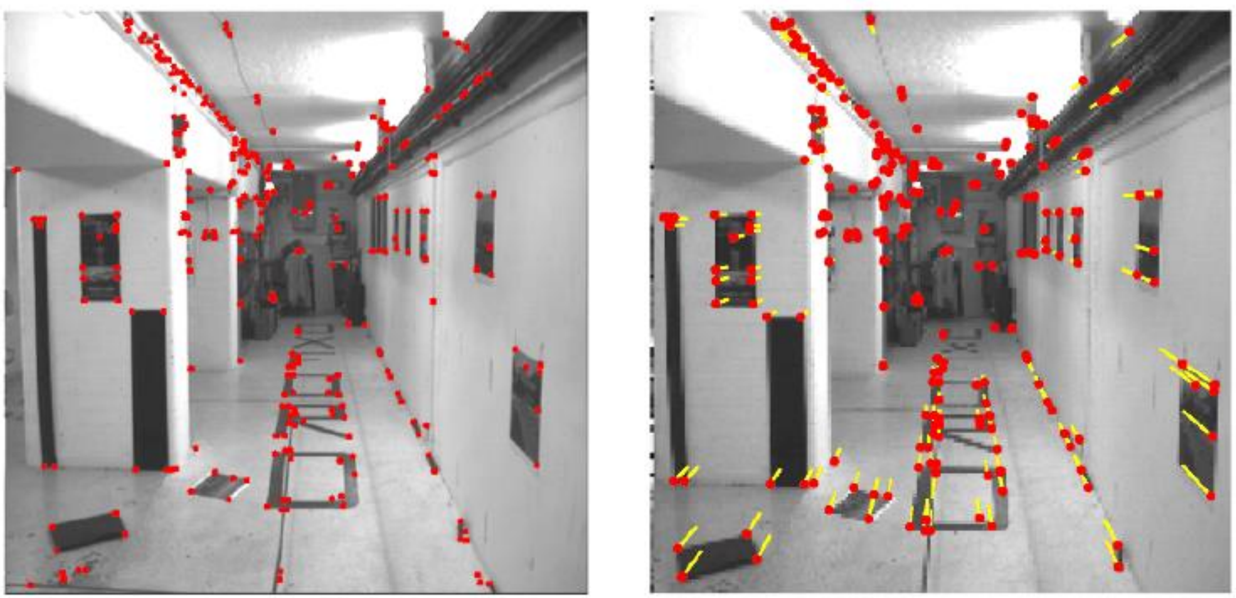
- inliers within threshold distance of epipolar line

Choose F with most support (inliers)





Using window search to get putative matches: noisy, but enough to compute F using RANSAC



Pruned matches: those consistent with epipolar geometry

Outline

Projective geometry

Vanishing points

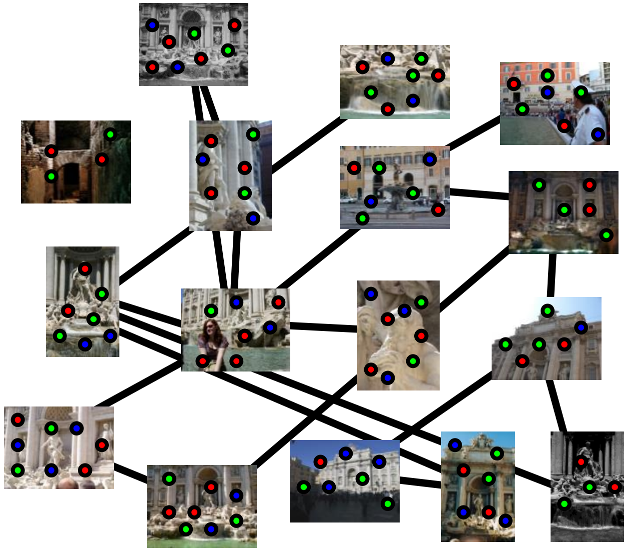
- Application: camera calibration
- Application: single-view metrology

Epipolar geometry

- Application: stereo correspondence
- Application: structure from motion revisited

Structure from Motion

1. Detect features using SIFT
2. Match features between pairs of images
3. Refine matching using RANSAC to find correspondences between each pair
4. Massive bundle adjustment

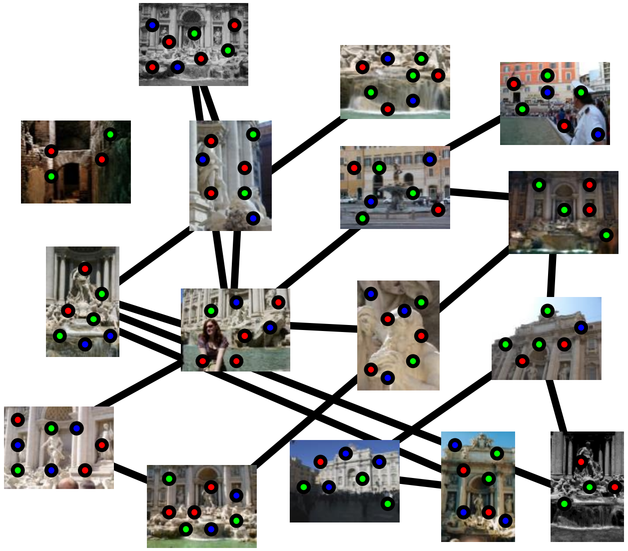


Noah Snavely

Structure from Motion

1. Detect features using SIFT
2. Match features between pairs of images
3. Refine matching using RANSAC to find correspondences between each pair
4. Massive bundle adjustment

Use fundamental matrix to detect inliers during RANSAC!



Noah Snavely

Structure from Motion Results

Noah Snavely

Structure from Motion Results

Noah Snavely

Recap

- Projective geometry
- Vanishing points
  - Application: camera calibration
  - Application: single-view metrology
- Epipolar geometry
  - Application: stereo correspondence
  - Application: structure from motion revisited