Computer Vision I

Image Formation

Credits:

Slides adapted from Konrad Schindler including many slides from:

Svetlana Lazebnik
Steve Seitz
Fredo Durand
Alyosha Efros

Stefan Roth

Announcements

- Moodle
  - … is up and running
  - You should be signed up automatically (when registered in TUCaN)

- First exercise class today
  - Will clarify details on homework assignments
  - Including hand in and groups
  - MATLAB tutorial

- First homework assignment
  - out soon

Computer Vision

TECHNISCHE UNIVERSITÄT DARMSTADT

- We want to devise computer algorithms to understand the visual world much like we do.

- This means
  - we have to use a lot of *different cues*...
  - we have to deal with *ambiguity*...
  - we have to exploit what is a *plausible and meaningful interpretation*...
  - ... in order to extract information about the 3D visual world from a small amount of data.

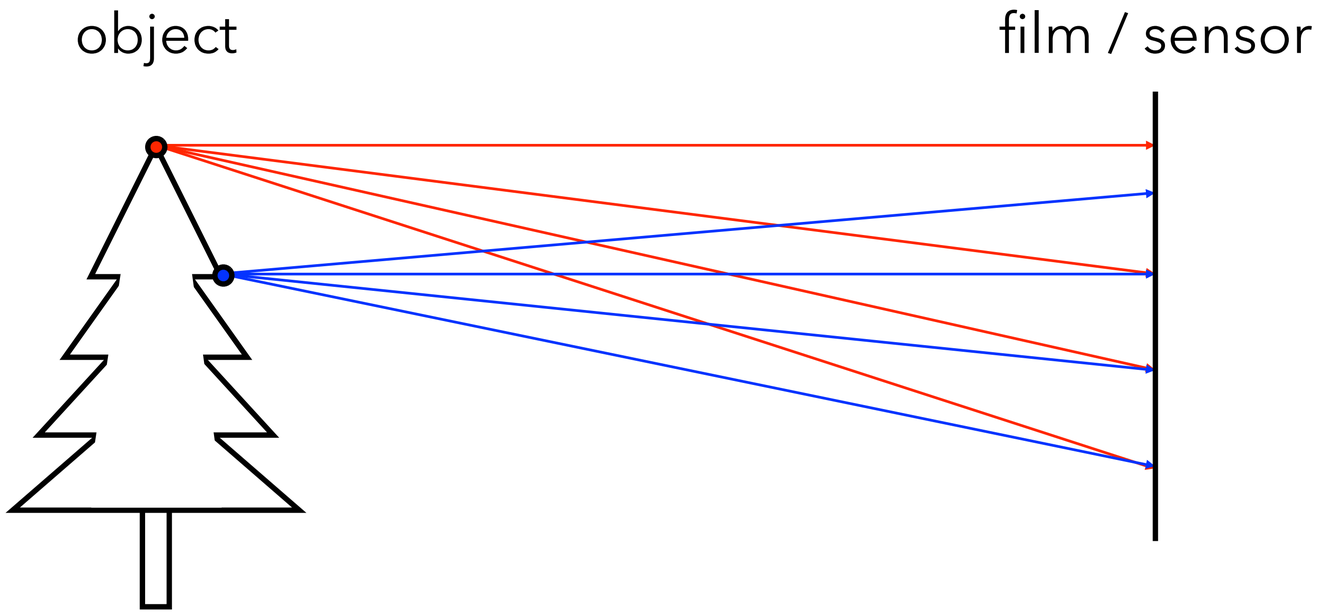
- CV is an inverse problem.

Overview

- The pinhole projection model
  - Properties of pinhole cameras
  - Algebraic model: Perspective projection matrix
- Single view geometry
  - Projection from world to image coordinates
  - Camera calibration
- Cameras with lenses
  - Depth of focus, field of view, lens aberrations
- Digital cameras
  - Types of sensors, color

How do we see the world?

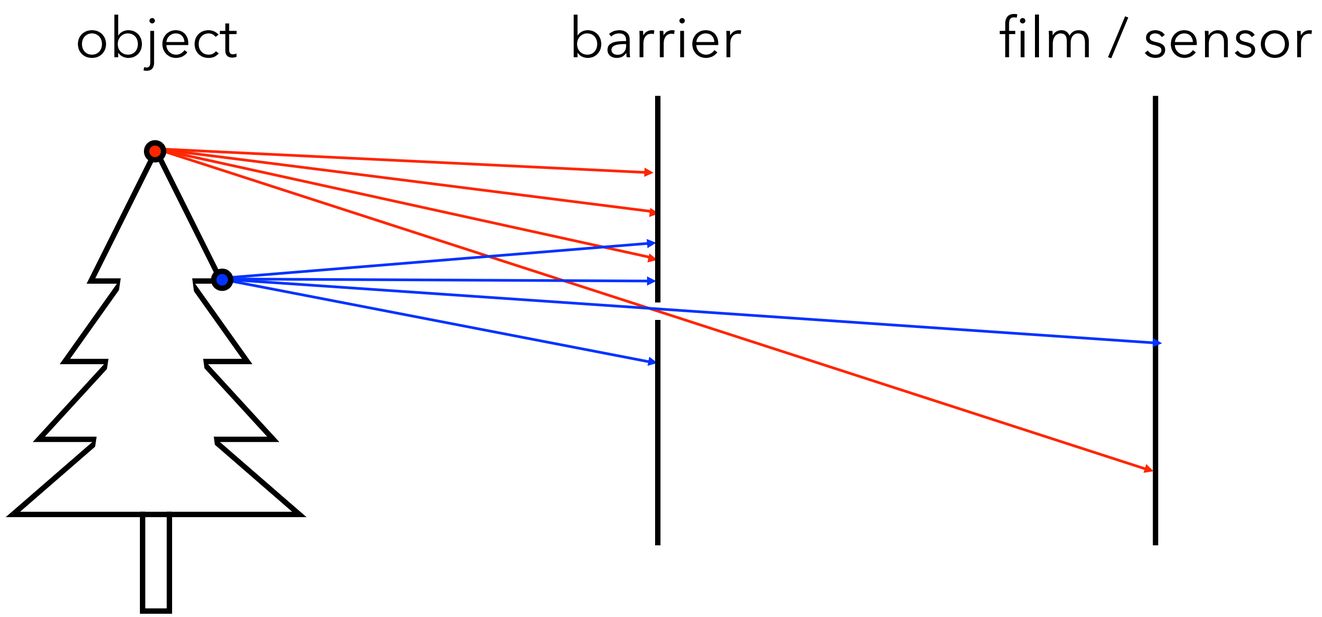
- Let's design a camera
  - *Idea 1:* Put a piece of film (or a CCD) in front of an object
  - Do we get a reasonable image?



[from Steve Seitz]

Pinhole camera

- Add a barrier to block off most of the rays *(idea 2)*
  - This reduces blurring
  - The opening is known as the aperture

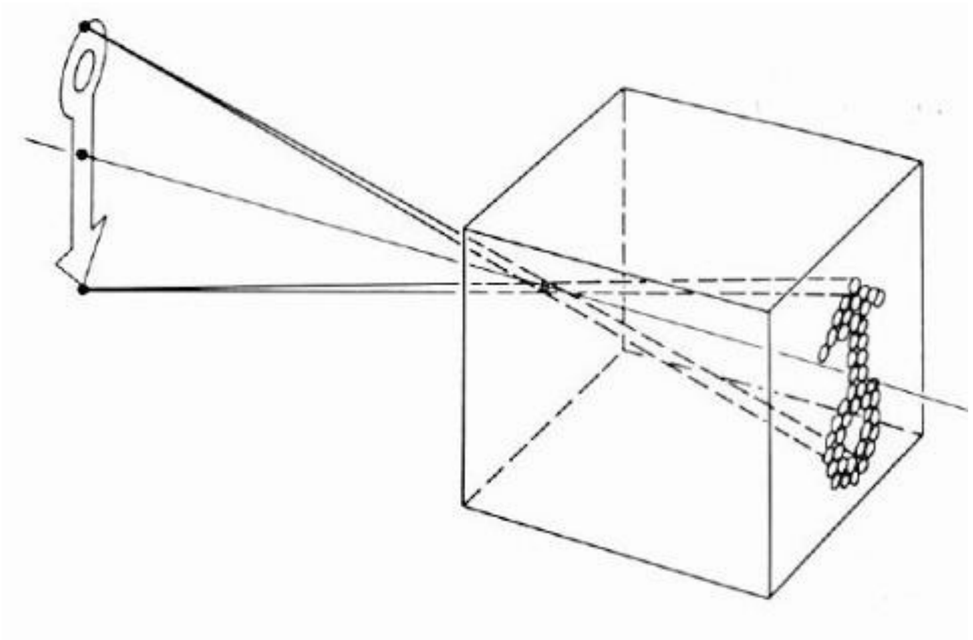


[from Steve Seitz]

Pinhole camera model

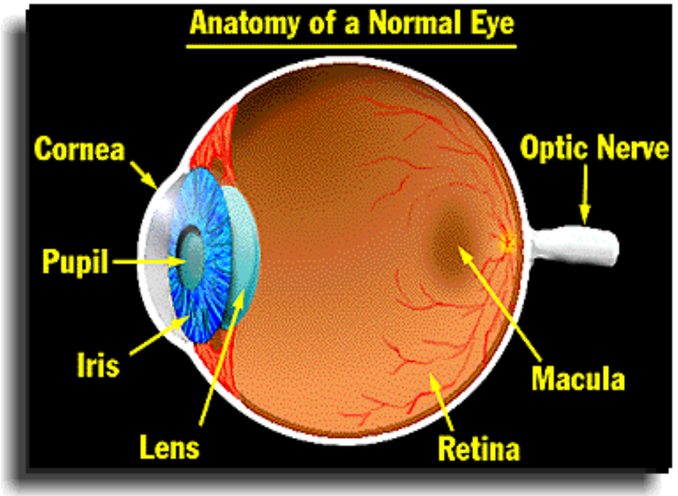
- Pinhole model
  - Captures pencil of rays – all rays through a single point
  - Projection rays are straight lines
  - The point is called *center of projection* (focal point)

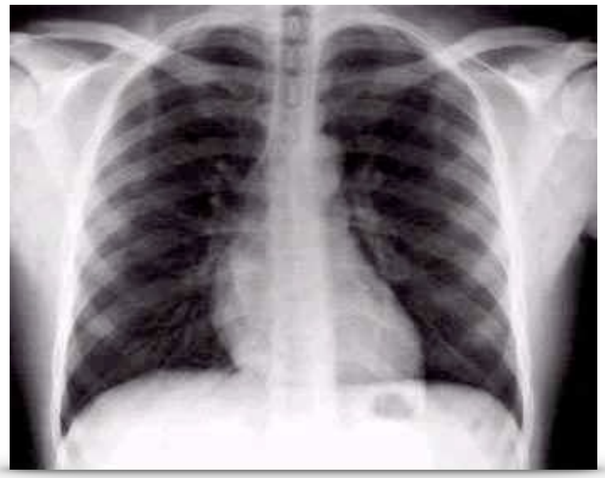
  - The image is formed on the image plane
  - Two equivalent projections: positive, negative



Pinhole camera

- A model for many common sensors
  - human eye
  - photographic cameras
  - X-ray machines

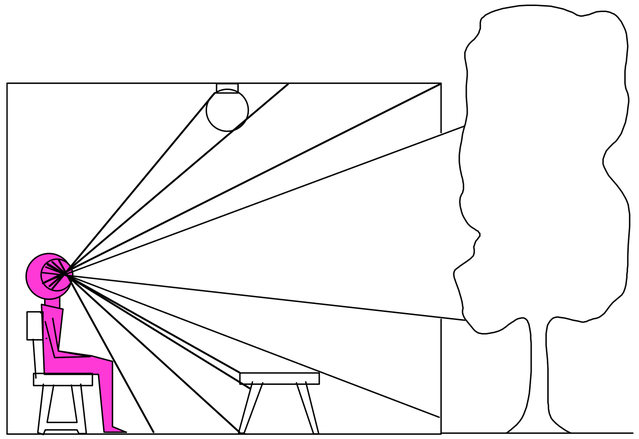




Dimensionality Reduction Machine (3D to 2D)

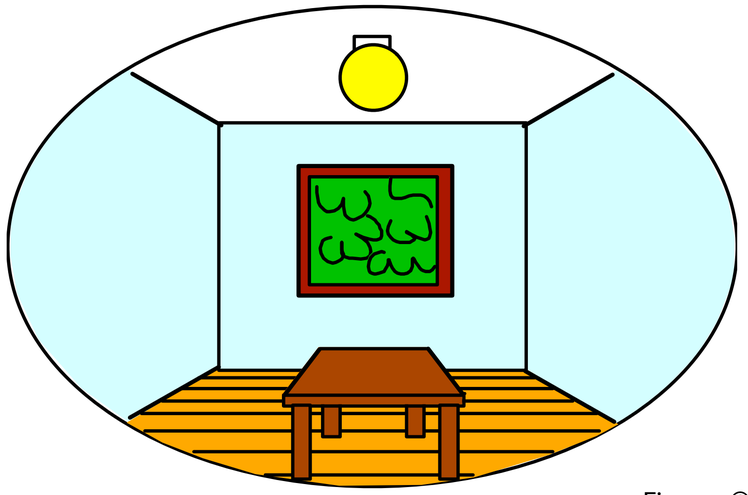
- What have we lost?
  - Angles
  - Distances (lengths)

3D world



Point of observation

2D image



Figures © Stephen E. Palmer, 2002

[from A. Efros]

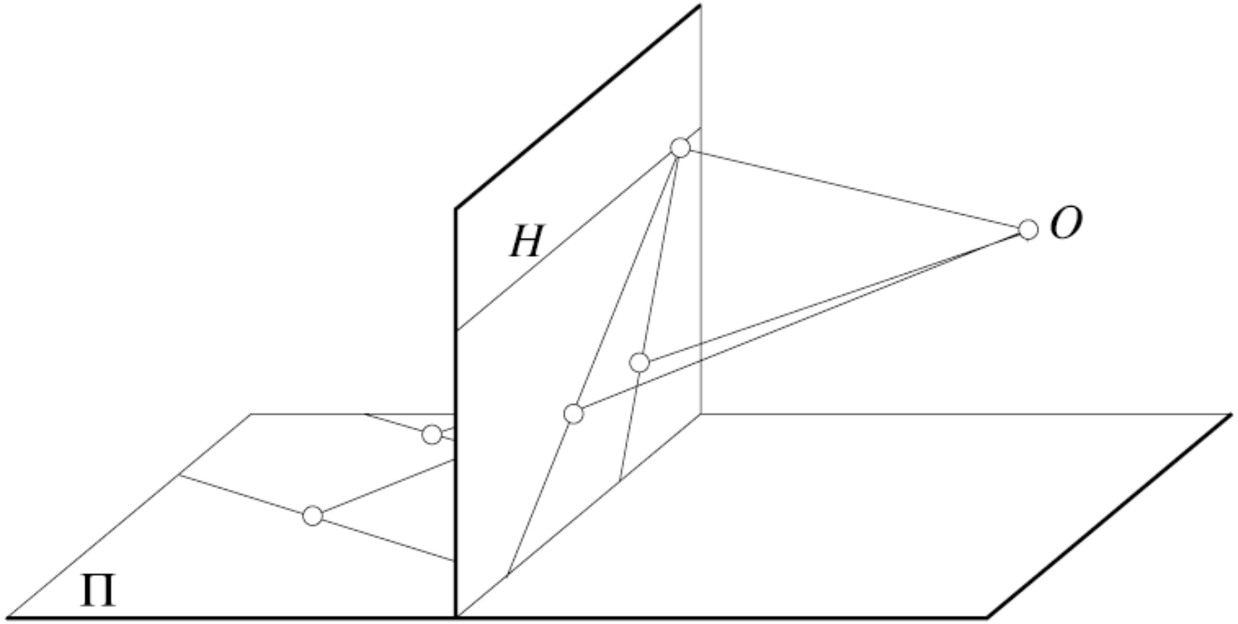
Projection properties

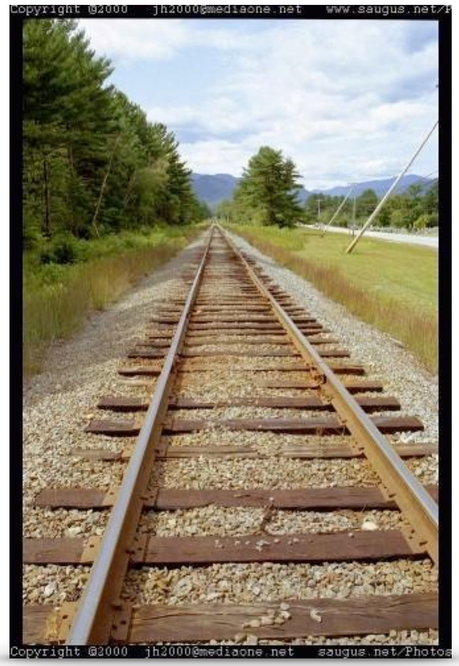
- Many-to-one
  - all points along the same ray map to the same point in image
- Points → points
  - But projection of points on focal plane is undefined
- Lines → lines (collinearity is preserved)
  - But line through focal point projects to a point
- Planes → planes (or half-planes)
  - But plane through focal point projects to line

[from S. Lazebnik]

Projection properties

- Parallel lines converge at a *vanishing point*
  - Each direction in space has its own vanishing point
  - But parallels also parallel to the image plane remain parallel
  - All directions in the same plane have vanishing points on the same line

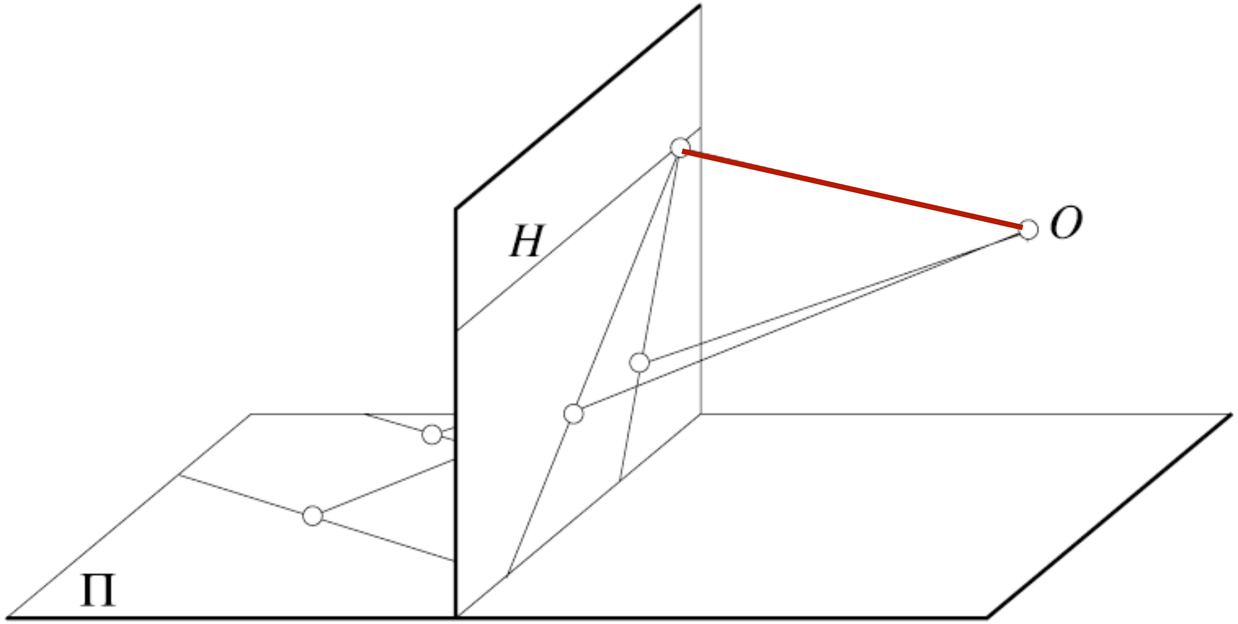


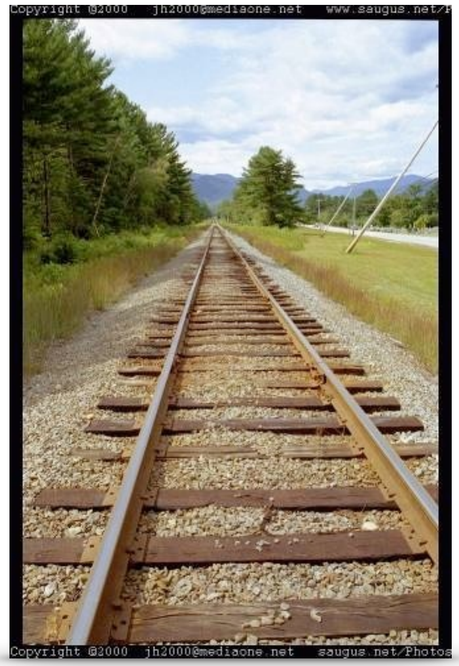


How do we construct the vanishing point/line?

Projection properties

- Parallel lines converge at a *vanishing point*
  - Each direction in space has its own vanishing point
  - But parallels also parallel to the image plane remain parallel
  - All directions in the same plane have vanishing points on the same line



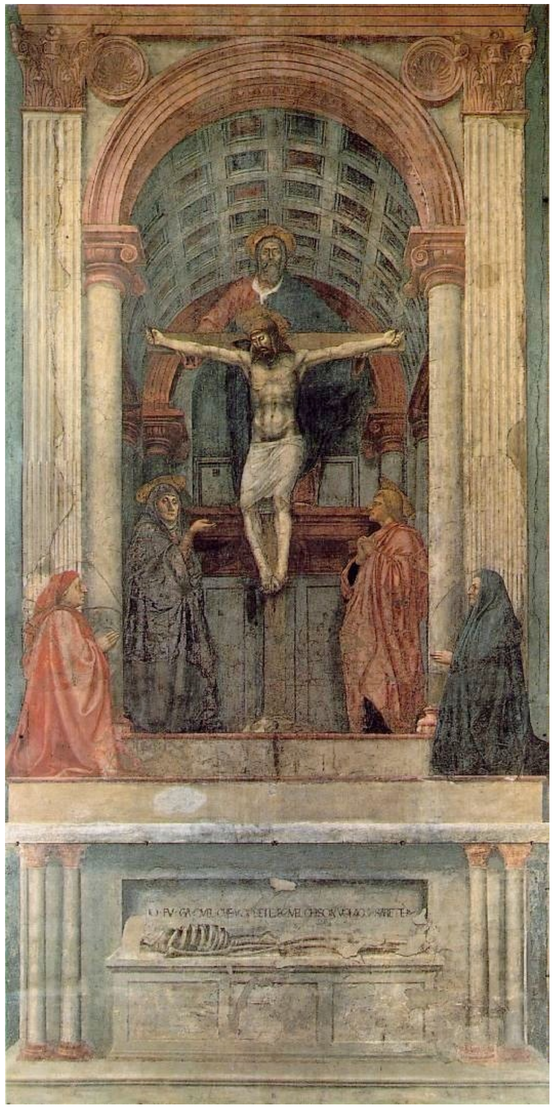


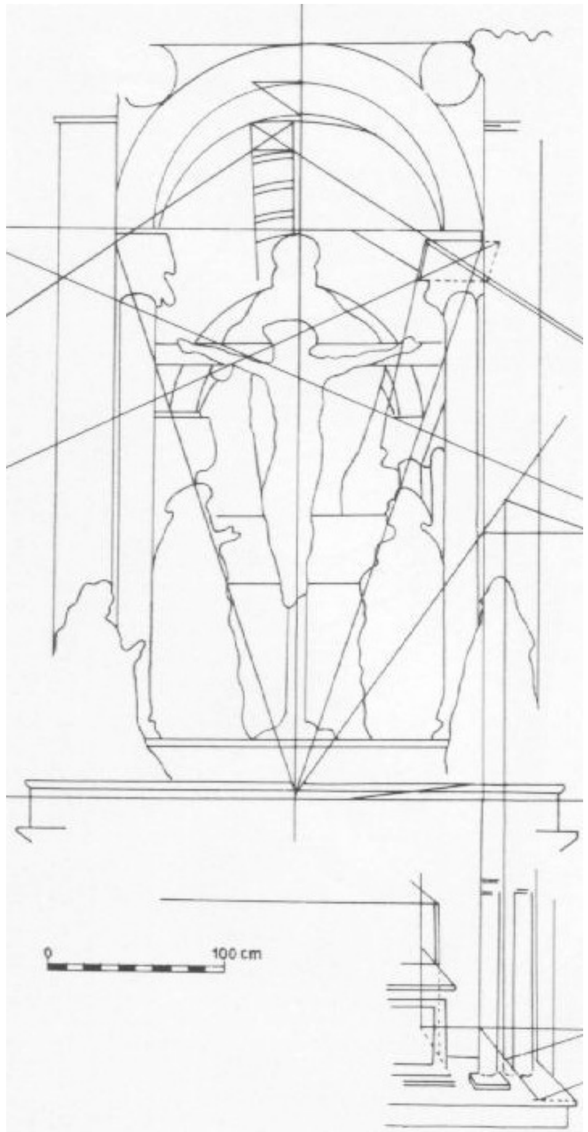
How do we construct the vanishing point/line?

One-point perspective

- Masaccio, Trinity
  Santa Maria Novella, *Florence*, 1425-28

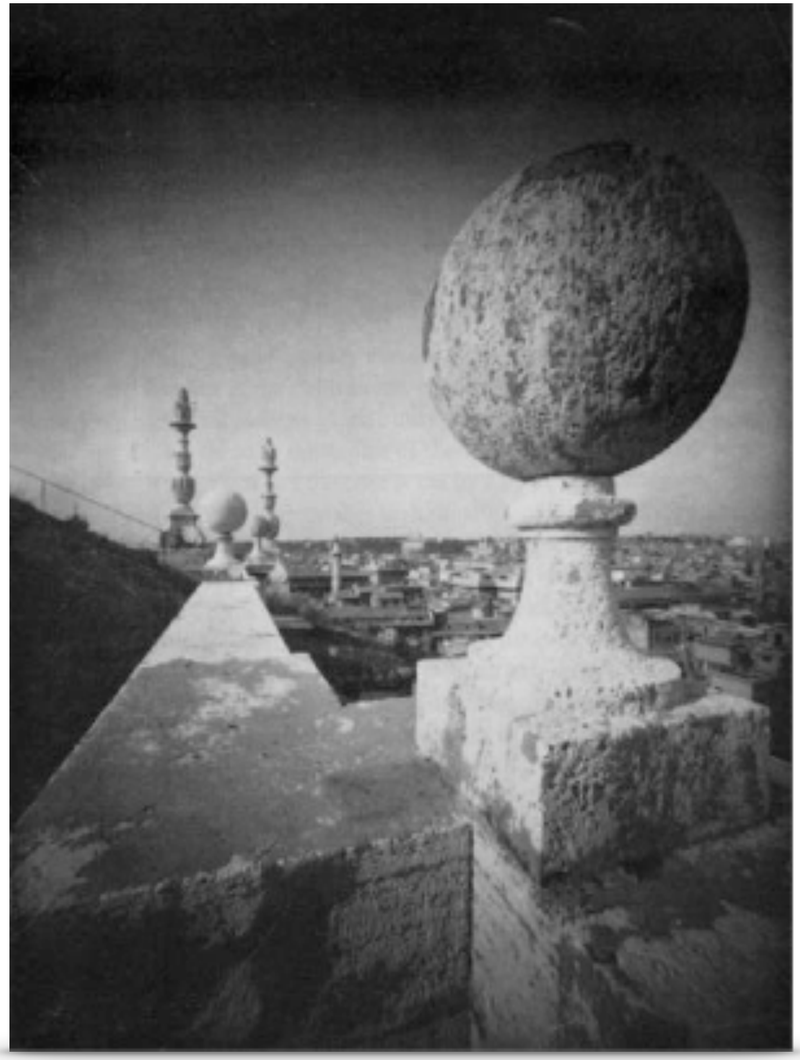
- First consistent use of perspective in Western art?





Perspective distortion

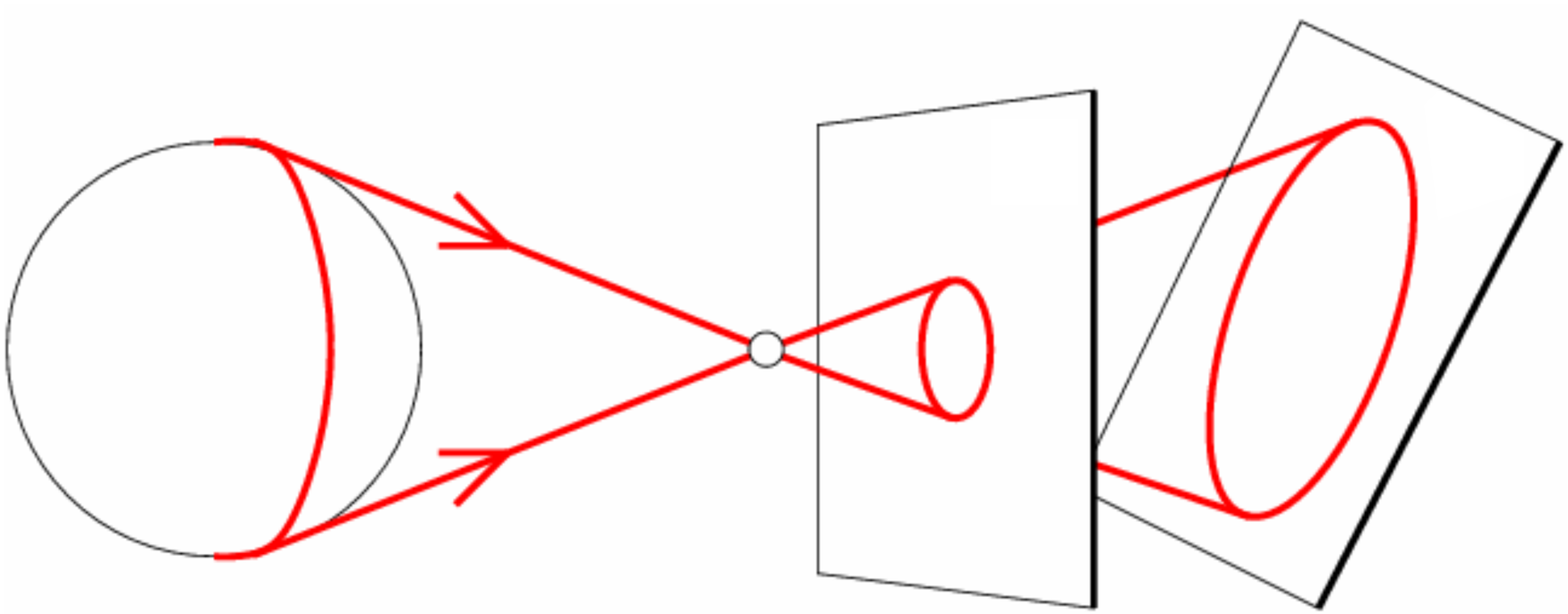
- What does a sphere project to?



[from F. Durand]

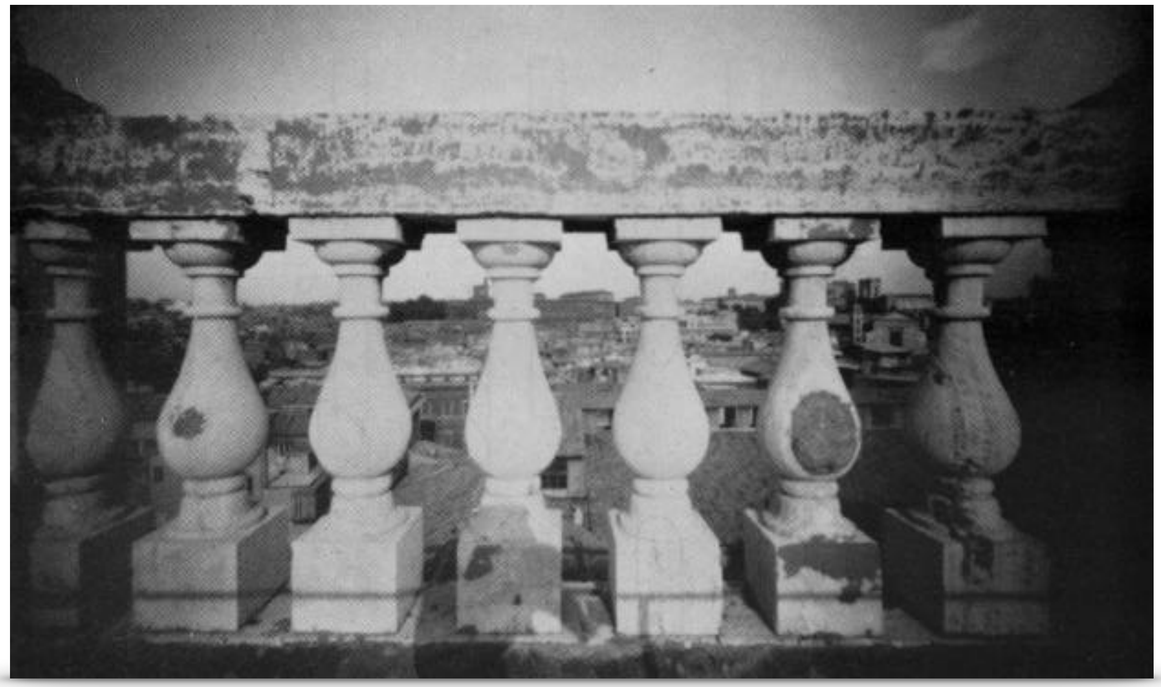
Perspective distortion

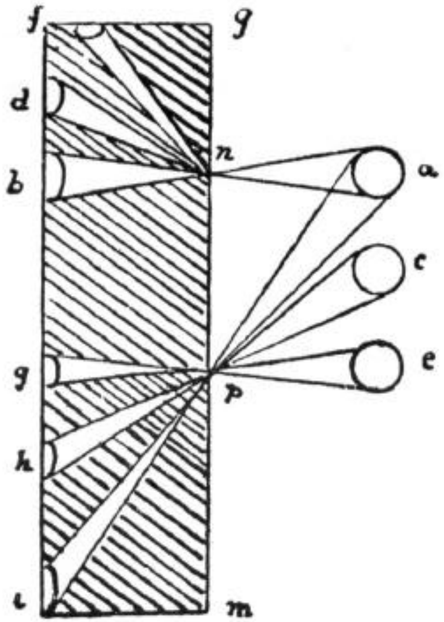
- What does a sphere project to?



Perspective distortion

- The exterior columns appear bigger
- The distortion is *not* due to lens flaws
- Problem pointed out by Da Vinci



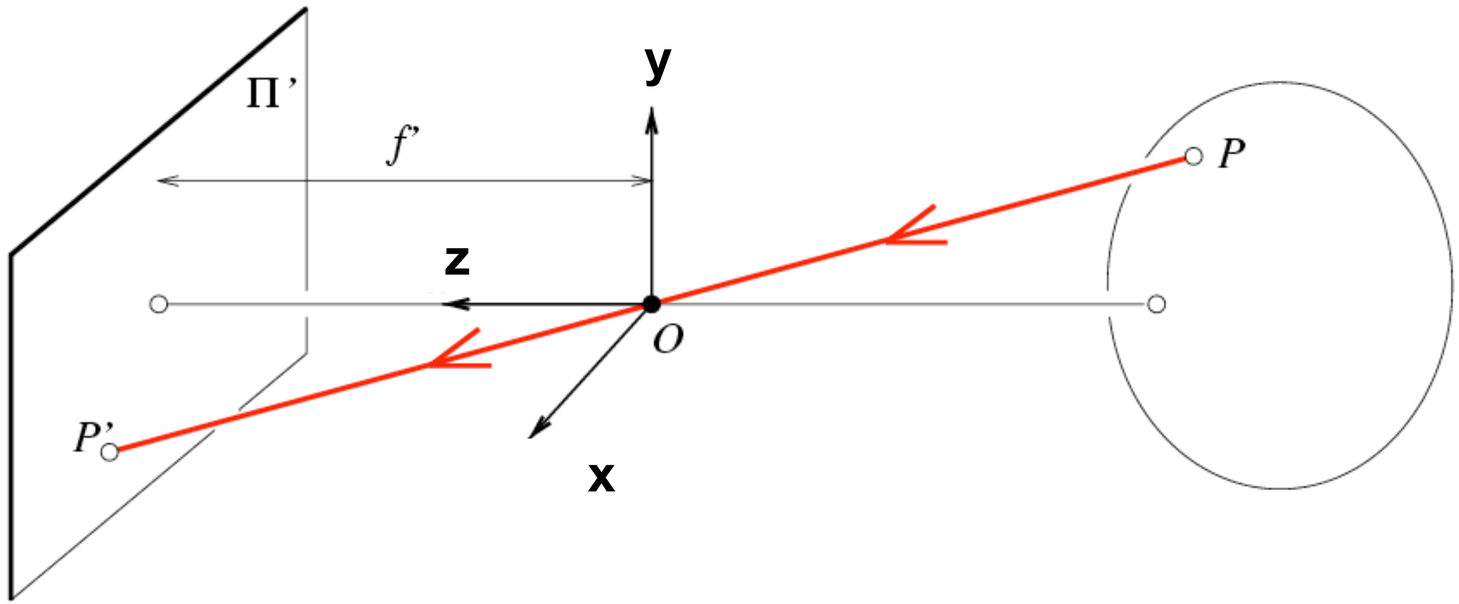


[from F. Durand]

Perspective distortion: People

Modeling projection

- The coordinate system
  - We will use the pinhole model as an approximation
  - Put the optical center (O) at the origin
  - Put the image plane (Π') in front of O

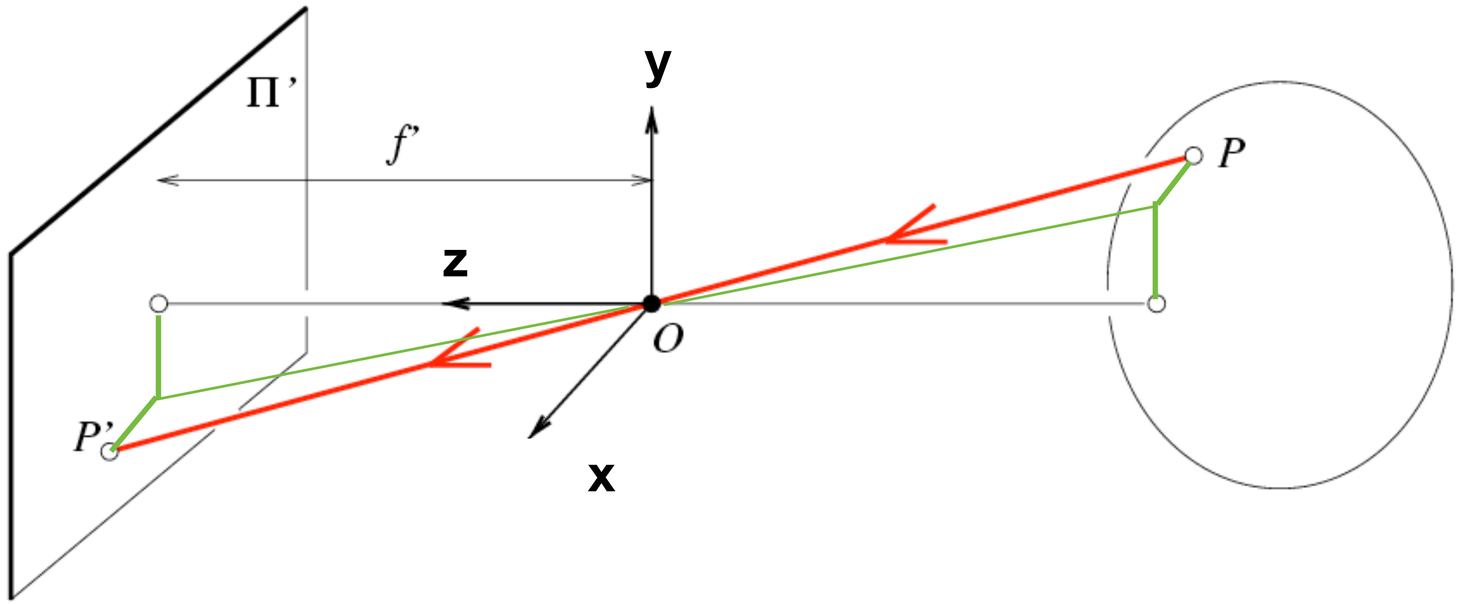


[from J. Ponce & S. Seitz]

Modeling projection

- Projection equations
  - Compute intersection with Π' of ray from $P = (x,y,z)$ to O
  - Derived using similar triangles
  - To get the projection, throw away last coordinate

$$(x, y, z) \rightarrow (f'\frac{x}{z}, f'\frac{y}{z}, f')$$
$$(x, y, z) \rightarrow (f'\frac{x}{z}, f'\frac{y}{z})$$



[from J. Ponce & S. Seitz]

Homogeneous coordinates

- Is this a linear transformation? $(x, y, z) \rightarrow (f' \frac{x}{z}, f' \frac{y}{z})$

[from Steve Seitz]

Homogeneous coordinates

- Is this a linear transformation? $(x, y, z) \rightarrow (f'\frac{x}{z}, f'\frac{y}{z})$
  - no – division by z is nonlinear
- Trick: add *one more coordinate*

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

homogeneous image coordinates

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

homogeneous scene coordinates

  - Converting from homogeneous coordinates

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

[from Steve Seitz]

Perspective Projection Matrix

- Projection is a matrix multiplication in homogeneous coordinates:

$$\begin{pmatrix} f' & 0 & 0 & 0 \\ 0 & f' & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} f'x \\ f'y \\ z \end{pmatrix} \qquad \Rightarrow \quad (f'\frac{x}{z}, f'\frac{y}{z})$$

divide by the third coordinate

[from S. Lazebnik]

Perspective Projection Matrix

- Projection is a matrix multiplication in homogeneous coordinates:

$$\begin{pmatrix} f' & 0 & 0 & 0 \\ 0 & f' & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} f'x \\ f'y \\ z \end{pmatrix} \quad \Rightarrow \quad (f'\frac{x}{z}, f'\frac{y}{z})$$

divide by the third coordinate

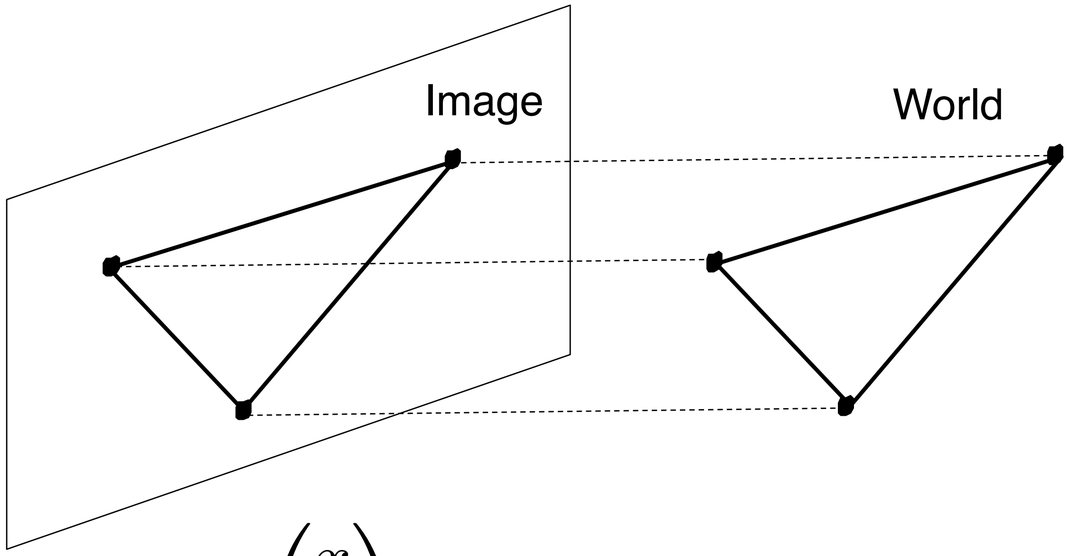
- By changing the algebraic representation all relevant transformations become linear

$$\begin{bmatrix} \text{2D point (3x1)} \end{bmatrix} = \begin{pmatrix} \text{Camera to pixel coord. trans. matrix (3x3)} \end{pmatrix} \begin{pmatrix} \text{Perspective projection matrix (3x4)} \end{pmatrix} \begin{pmatrix} \text{World to camera coord. trans. matrix (4x4)} \end{pmatrix} \begin{bmatrix} \text{3D point (4x1)} \end{bmatrix}$$

[from S. Lazebnik]

Orthographic Projection

- Special case of perspective projection
  - Distance from center of projection to image plane is infinite
  - Also called *"parallel projection"*

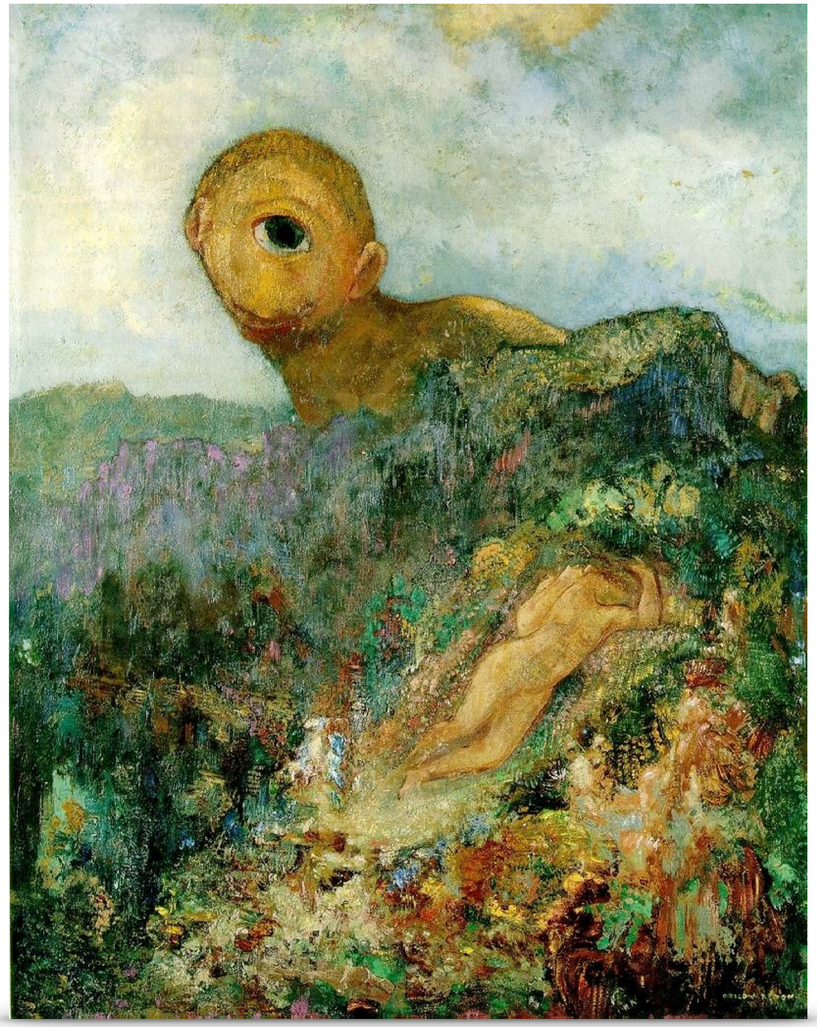


  - What's the projection matrix?

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad \Rightarrow \quad (x, y)$$

[from Steve Seitz]

Single-view geometry

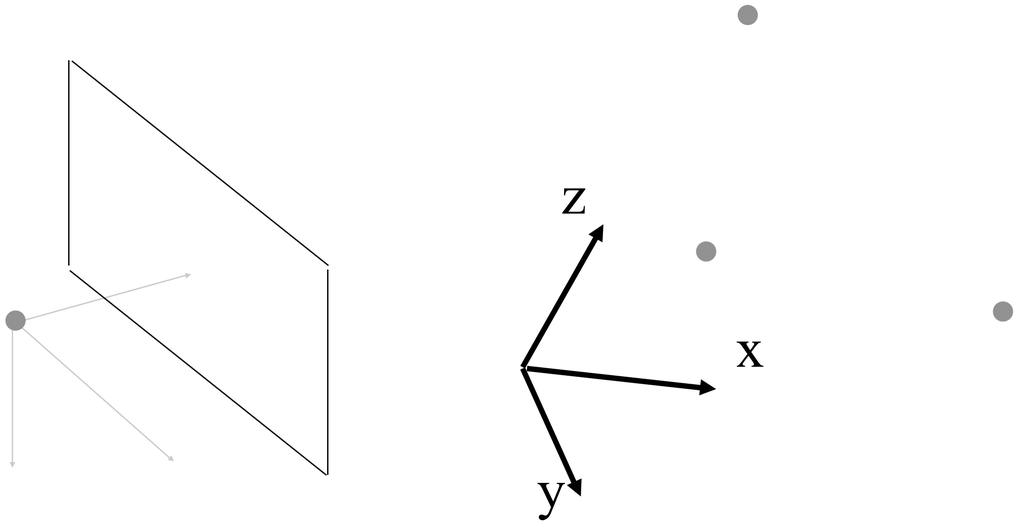


Odilon Redon, Cyclops, 1914

[from S. Lazebnik]

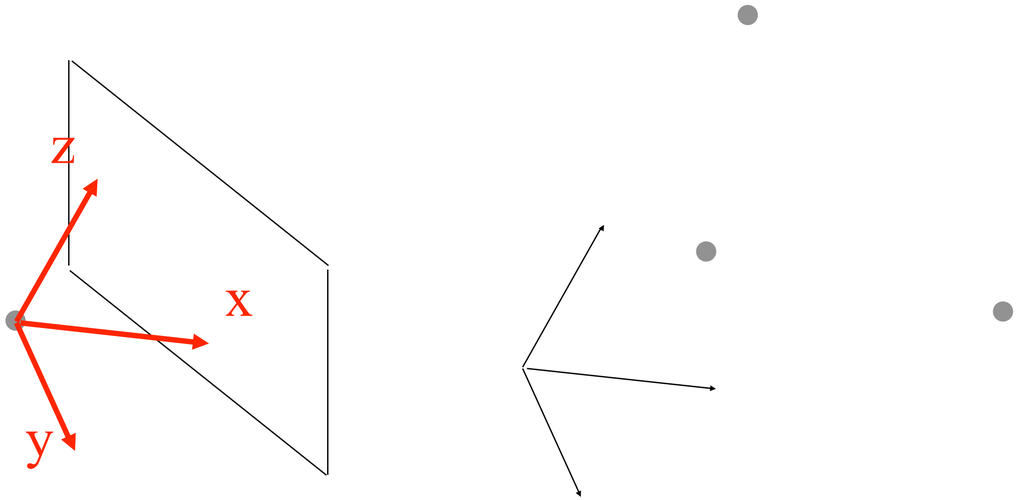
Camera calibration

- Let's place the camera in the world
  - Translation
  - Rotation
  - Pinhole projection



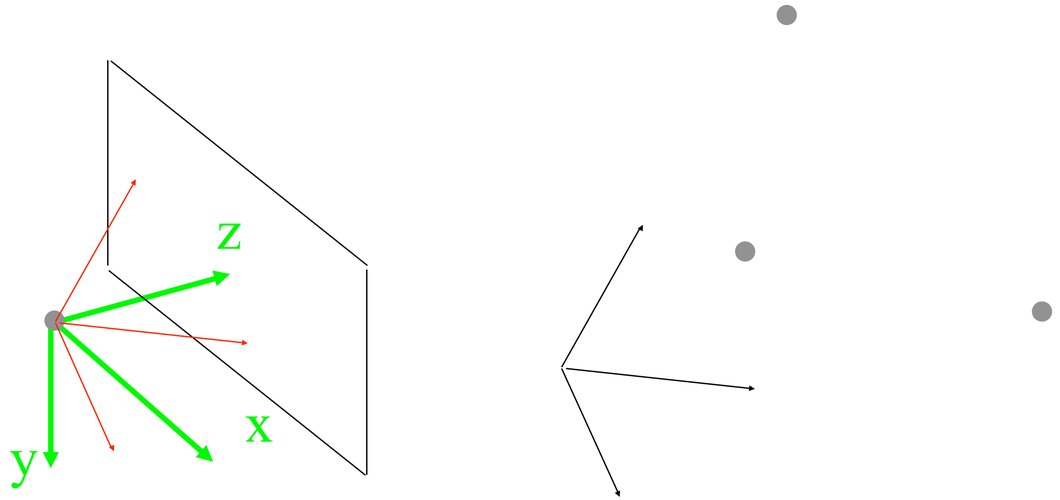
Camera calibration

- Let's place the camera in the world
  - Translation
  - Rotation
  - Pinhole projection



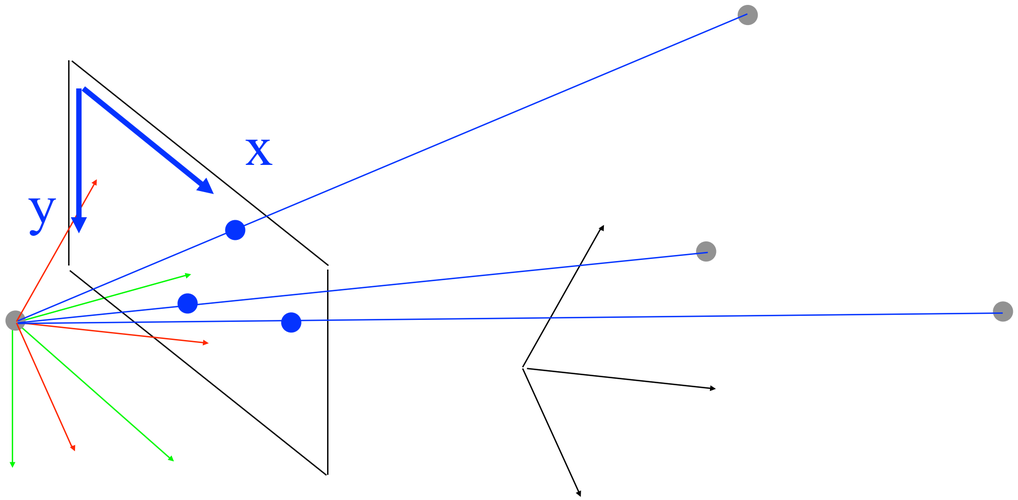
Camera calibration

- Let's place the camera in the world
  - Translation
  - Rotation
  - Pinhole projection

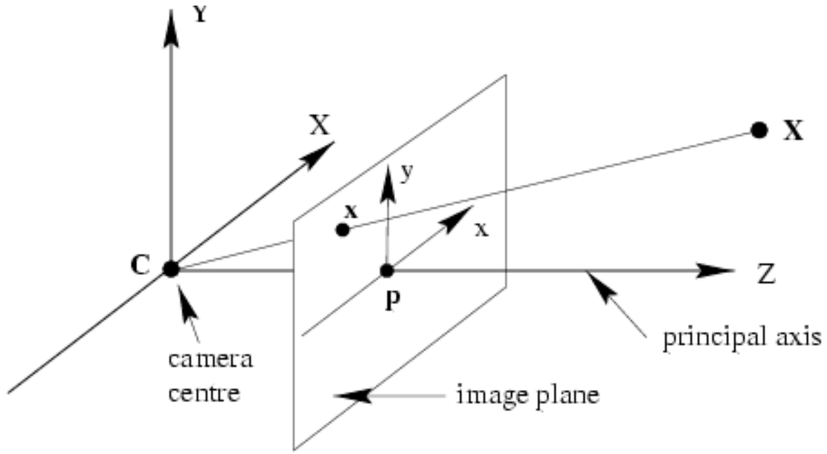


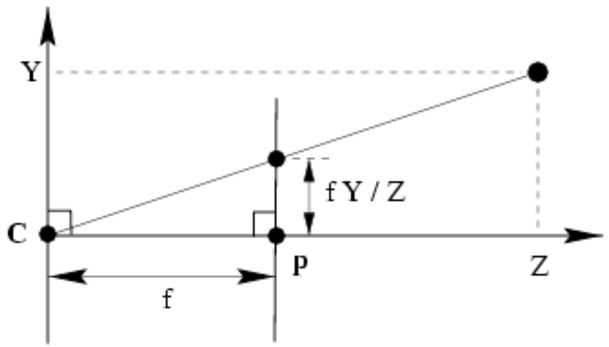
Camera calibration

- Let's place the camera in the world
  - Translation
  - Rotation
  - Pinhole projection



Recall: Pinhole camera model



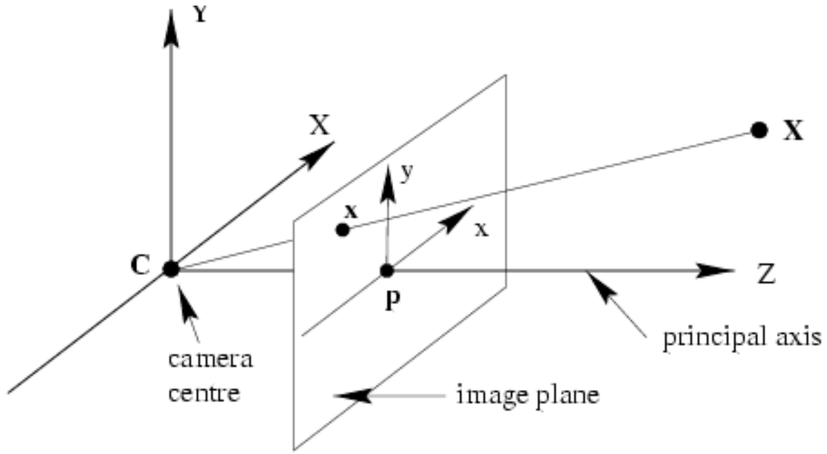


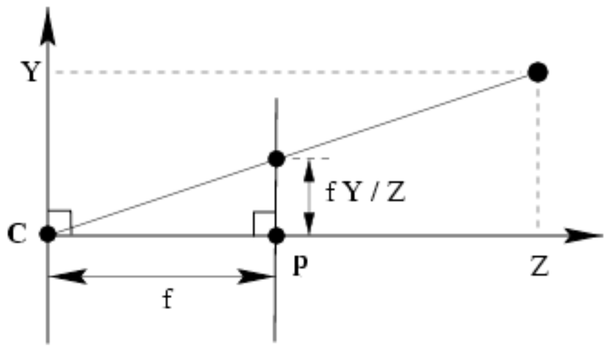
$$(X, Y, Z) \rightarrow (fX/Z, fY/Z)$$

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

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

Recall: Pinhole camera model





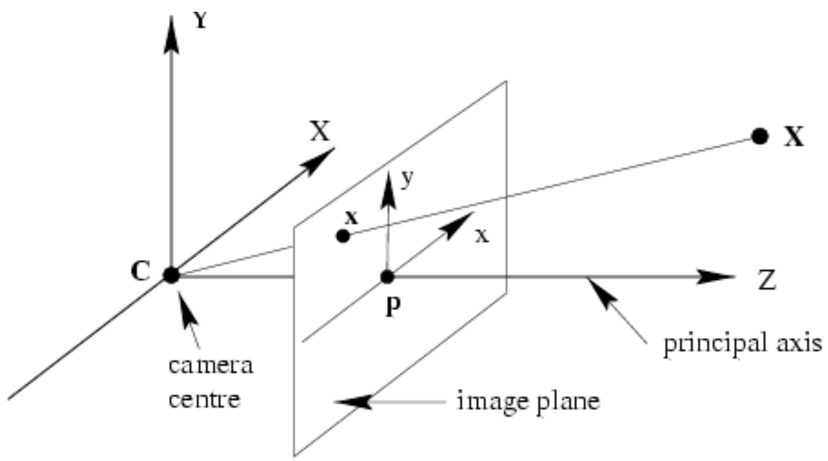
$$(X, Y, Z) \rightarrow (fX/Z, fY/Z)$$

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

$$\mathbf{x} = P\mathbf{X} \qquad P = \mathrm{diag}(f, f, 1)[I|0]$$

Camera coordinate system

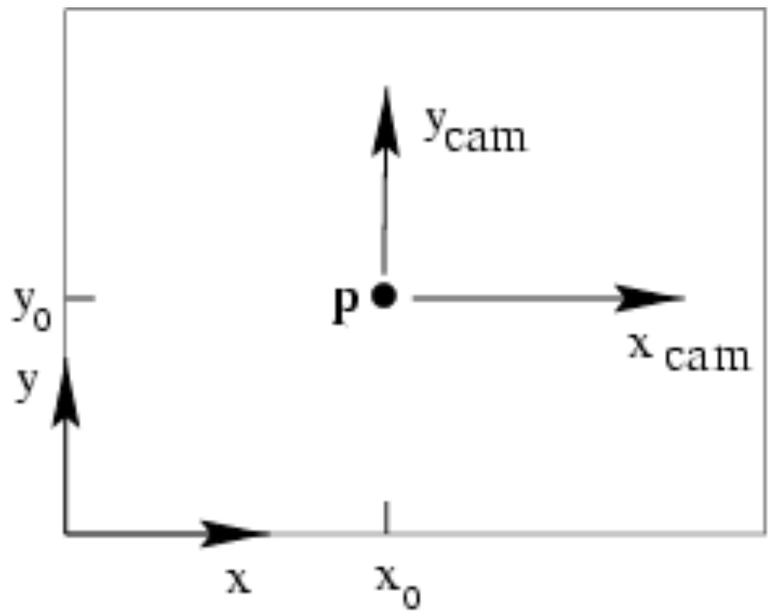
- *Principal axis*: line from the camera center perpendicular to the image plane
- *Normalized (camera) coordinate system*: camera center is at the origin and the principal axis is the z-axis
- *Principal point* (p): point where principal axis intersects the image plane (origin of normalized coordinate system)



[from S. Lazebnik]

Principal point offset

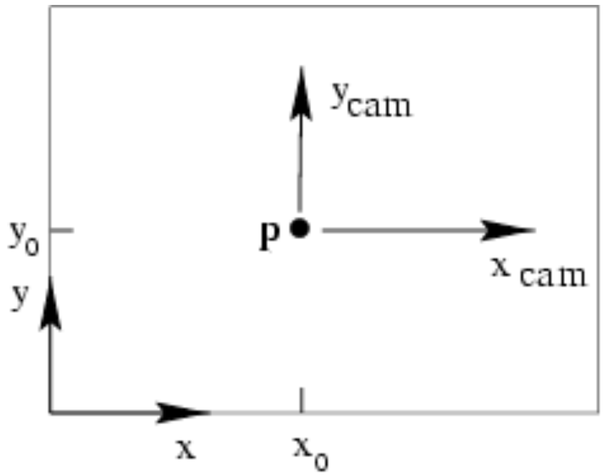
- *Camera coordinate system*: origin is at the principal point
- *Image coordinate system*: origin is in the corner



principal point: (p_x, p_y)

[from S. Lazebnik]

Principal point offset

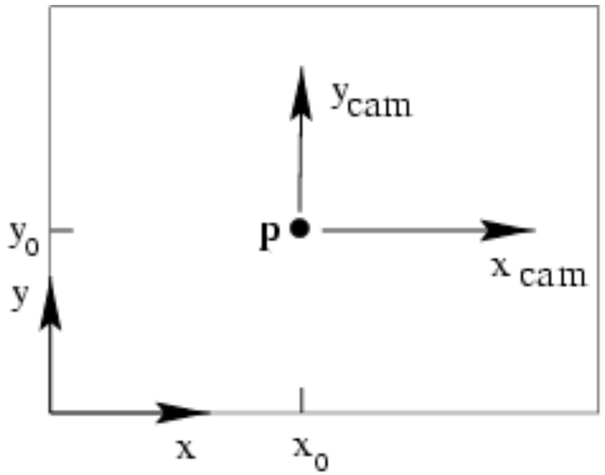


principal point: (p_x, p_y)

$$(X, Y, Z) \rightarrow (fX/Z + p_x, fY/Z + p_y)$$

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} fX + Zp_x \\ fY + Zp_y \\ Z \end{pmatrix} = \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

Principal point offset



principal point: (p_x, p_y)

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \rightarrow \begin{pmatrix} fX + Zp_x \\ fY + Zp_y \\ Z \end{pmatrix} = \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

$$K = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix}$$

calibration matrix

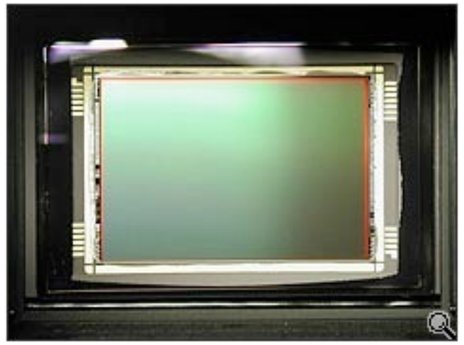
$$P = K[I|0]$$

Pixel coordinates

m_x pixels per unit (m,mm,inch,...) in horizontal direction,

m_y pixels per unit in vertical direction

Pixel size: $\frac{1}{m_x} \times \frac{1}{m_y}$

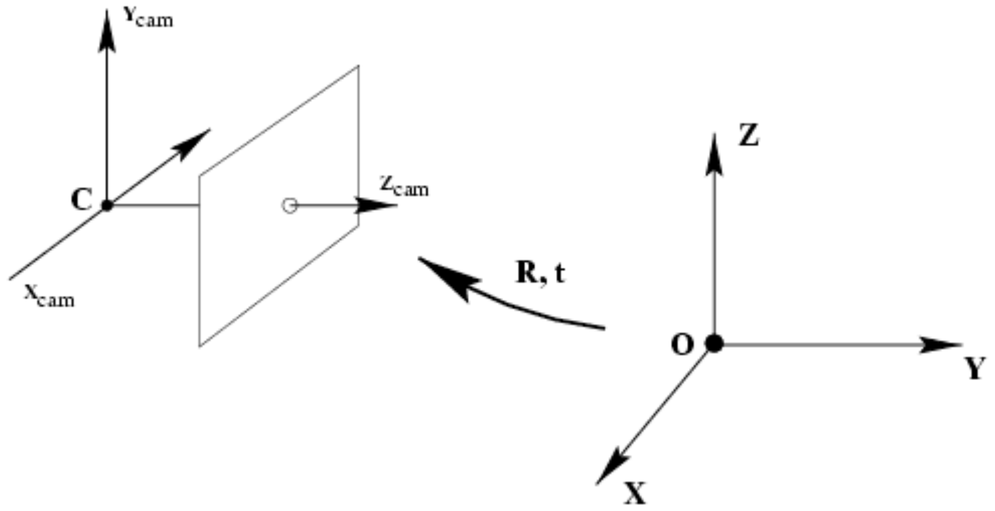


$$K = \underbrace{\begin{bmatrix} m_x & & \\ & m_y & \\ & & 1 \end{bmatrix}}_{\text{pixels/unit}} \underbrace{\begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix}}_{\text{units}} = \underbrace{\begin{bmatrix} \alpha_x & & \beta_x \\ & \alpha_y & \beta_y \\ & & 1 \end{bmatrix}}_{\text{pixels}}$$

[from S. Lazebnik]

Camera rotation and translation

- The camera coordinate frame is related to the world coordinate frame by a rotation and a translation



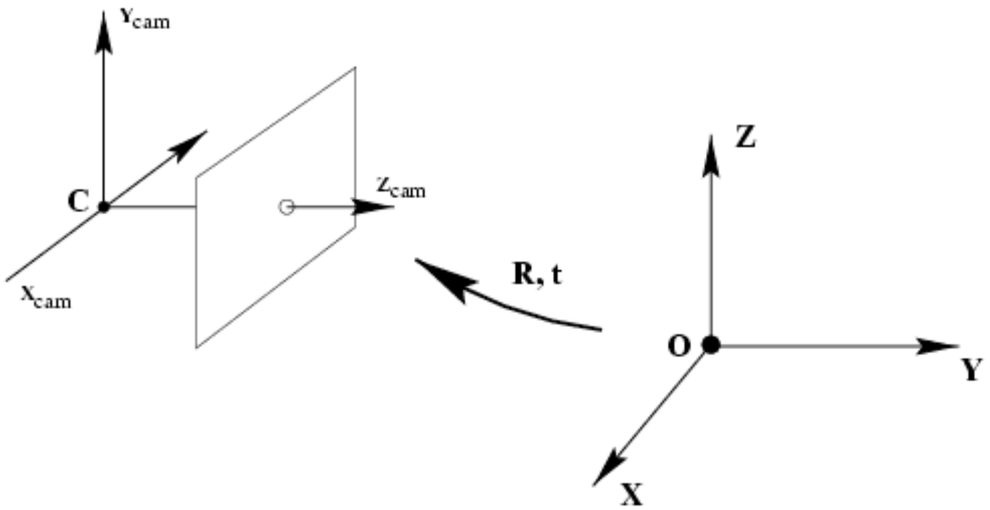
$$\tilde{X}_{cam} = R(\tilde{X} - \tilde{C})$$

coords. of point in camera frame

coords. of a point in world frame (non-homogeneous)

coords. of camera center in world frame

Camera rotation and translation



In non-homogeneous coordinates:

$$\tilde{X}_{cam} = R(\tilde{X} - \tilde{C})$$

$$X_{cam} = \begin{bmatrix} R & -R\tilde{C} \\ 0 & 1 \end{bmatrix} \begin{pmatrix} \tilde{X} \\ 1 \end{pmatrix} = \begin{bmatrix} R & -R\tilde{C} \\ 0 & 1 \end{bmatrix} X$$

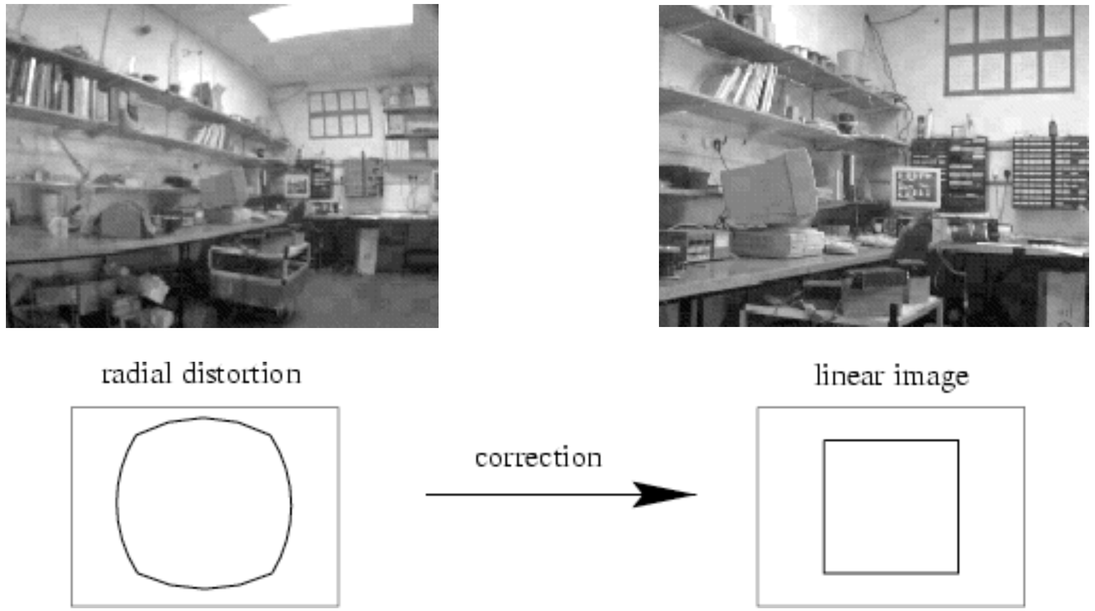
$$x = K[I|0]X_{cam} = K[R| - R\tilde{C}]X \qquad P = K[R|t] \qquad t = -R\tilde{C}$$

Note: C is the null space of the camera projection matrix (P C = 0)

Camera parameters

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

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



[from S. Lazebnik]

Camera parameters

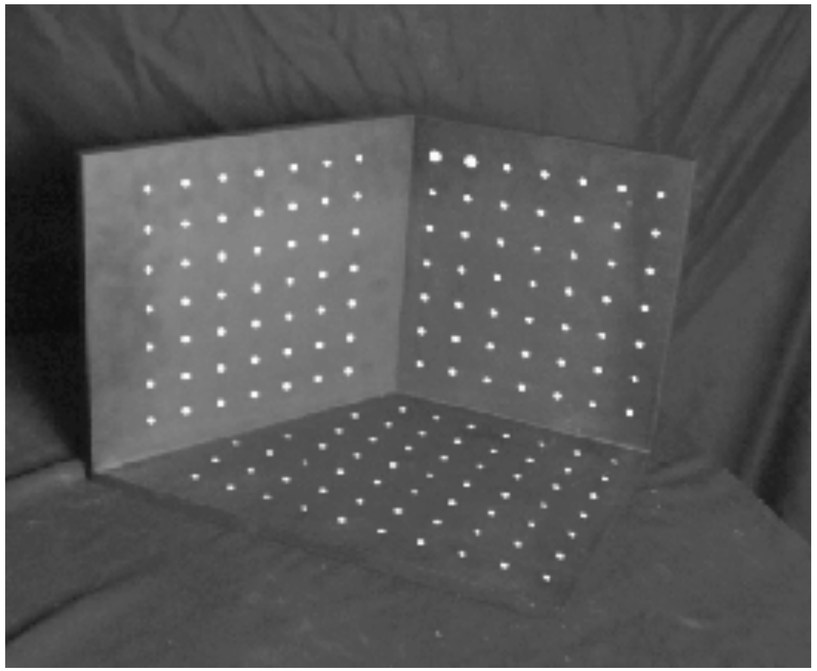
- Intrinsic parameters
  - Principal point coordinates
  - Focal length
  - Pixel magnification factors
  - Skew (non-rectangular pixels)
  - Radial distortion

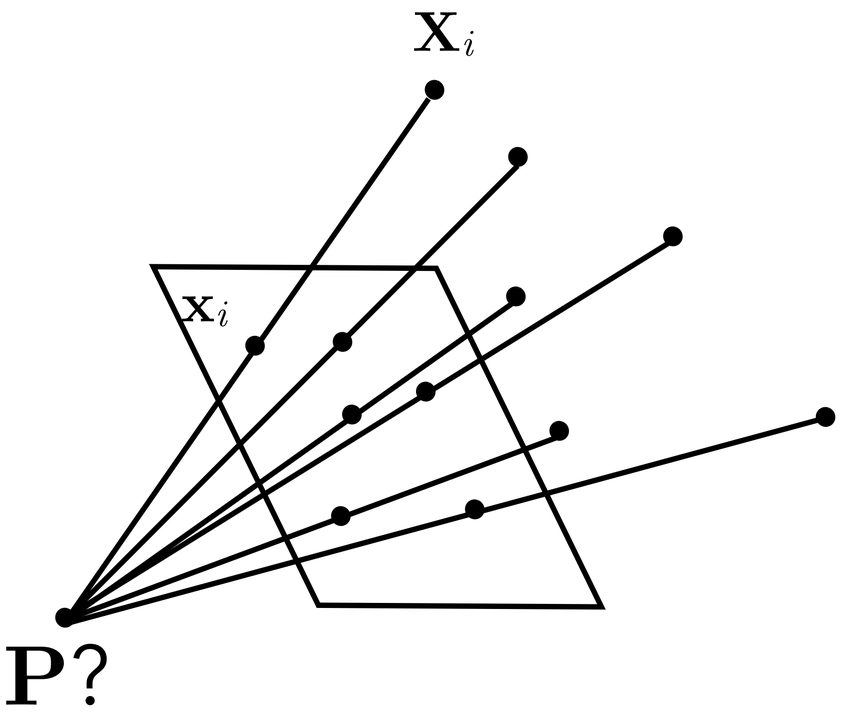
- Extrinsic parameters
  - Rotation and translation relative to world coordinate system

[from S. Lazebnik]

Camera calibration

- Given n points with known 3D coordinates $\mathbf{X}_i$ and known image projections $\mathbf{x}_i$, estimate the camera parameters





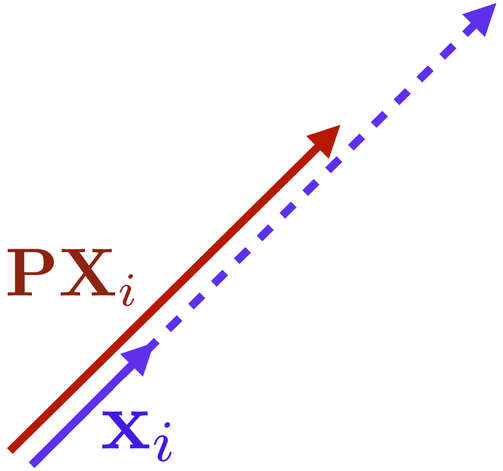
[from S. Lazebnik]

Camera calibration

$\mathbf{p}_j^{\mathrm{T}}$ are the rows of $\mathbf{P}$

- Use *collinearity*

$$\lambda \mathbf{x}_i = \mathbf{P}\mathbf{X}_i \qquad \lambda \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} = \begin{bmatrix} \mathbf{p}_1^{\mathrm{T}} \\ \mathbf{p}_2^{\mathrm{T}} \\ \mathbf{p}_3^{\mathrm{T}} \end{bmatrix} \mathbf{X}_i$$



- Write using cross-product $\mathbf{x}_i \times \mathbf{P}\mathbf{X}_i = 0$

- Expand cross-product

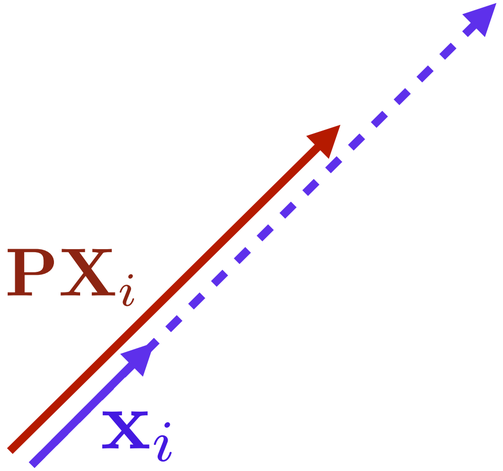
$$\begin{pmatrix} y_i \mathbf{p}_3^{\mathrm{T}} \mathbf{X}_i - \mathbf{p}_2^{\mathrm{T}} \mathbf{X}_i \\ \mathbf{p}_1^{\mathrm{T}} \mathbf{X}_i - x_i \mathbf{p}_3^{\mathrm{T}} \mathbf{X}_i \\ x_i \mathbf{p}_2^{\mathrm{T}} \mathbf{X}_i - y_i \mathbf{p}_1^{\mathrm{T}} \mathbf{X}_i \end{pmatrix} = 0$$

Camera calibration

$\mathbf{p}_j^{\mathrm{T}}$ are the rows of $\mathbf{P}$

- Use collinearity

$$\lambda \mathbf{x}_i = \mathbf{P}\mathbf{X}_i \qquad \lambda \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} = \begin{bmatrix} \mathbf{p}_1^{\mathrm{T}} \\ \mathbf{p}_2^{\mathrm{T}} \\ \mathbf{p}_3^{\mathrm{T}} \end{bmatrix} \mathbf{X}_i$$



- Write using cross-product

$$\begin{pmatrix} y_i \mathbf{p}_3^{\mathrm{T}} \mathbf{X}_i - \mathbf{p}_2^{\mathrm{T}} \mathbf{X}_i \\ \mathbf{p}_1^{\mathrm{T}} \mathbf{X}_i - x_i \mathbf{p}_3^{\mathrm{T}} \mathbf{X}_i \\ x_i \mathbf{p}_2^{\mathrm{T}} \mathbf{X}_i - y_i \mathbf{p}_1^{\mathrm{T}} \mathbf{X}_i \end{pmatrix} = 0$$

entries of $\mathbf{P}$ stacked up in a vector

- Write as equation system

$$\begin{bmatrix} 0 & -\mathbf{X}_i^{\mathrm{T}} & y_i \mathbf{X}_i^{\mathrm{T}} \\ \mathbf{X}_i^{\mathrm{T}} & 0 & -x_i \mathbf{X}_i^{\mathrm{T}} \\ -y_i \mathbf{X}_i^{\mathrm{T}} & x_i \mathbf{X}_i^{\mathrm{T}} & 0 \end{bmatrix} \begin{pmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{pmatrix} = \mathbf{0}$$

Two linearly independent equations!

Camera calibration

- $\mathbf{P}$ has 11 degrees of freedom (12 parameters, but scale is arbitrary)
- One 2D/3D correspondence gives us two linearly independent equations
- 6 correspondences needed for a minimal solution
- more points: least-squares estimation

$$\begin{bmatrix} 0 & -\mathbf{X}_1^{\mathrm{T}} & y_1\mathbf{X}_1^{\mathrm{T}} \\ \mathbf{X}_1^{\mathrm{T}} & 0 & -x_1\mathbf{X}_1^{\mathrm{T}} \\ \vdots & \vdots & \vdots \\ 0 & -\mathbf{X}_n^{\mathrm{T}} & y_n\mathbf{X}_n^{\mathrm{T}} \\ \mathbf{X}_n^{\mathrm{T}} & 0 & -x_n\mathbf{X}_n^{\mathrm{T}} \end{bmatrix} \begin{pmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{pmatrix} = \mathbf{0} \quad \Leftrightarrow \quad \mathbf{A}\mathbf{p} = \mathbf{0}$$

equation system matrix

projection matrix as vector

Camera calibration

- Problem: *trivial solution* $\mathbf{p} = 0$

- Solution: add constraint on norm of $\mathbf{p}$ (value can be arbitrary)

$$\mathbf{A}\mathbf{p} = 0 \quad \text{s.t.} \quad \|\mathbf{p}\| = 1$$

homogeneous least squares

- In practice: Find the right nullspace of $\mathbf{A}$ using *singular value decomposition* (SVD)
  - Decompose equation system matrix $\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^{\mathrm{T}}$
  - Assume that singular values are sorted, i.e. $\mathbf{S} = \mathrm{diag}(s_1, \ldots, s_{12}), \quad s_{i+1} \geq s_i$
  - Take last right singular vector $\mathbf{p} = \mathbf{v}_{12}$

- Why does this work?

Homogeneous least squares: Introduction

- We would like to find the vector $\mathbf{p} \in \mathbb{R}^n$ that satisfies

$$\mathbf{A}\mathbf{p} = \mathbf{0},$$

  with $\mathbf{A} \in \mathbb{R}^{m \times n}$ matrix and $\mathbf{0} \in \mathbb{R}^m$. We further assume that $m \geq n$ and that rank$(\mathbf{A}) = n$.
- Since the entries of $\mathbf{A}$ are measured and thus affected by noise, we rather minimize the (squared) error

$$\|\mathbf{A}\mathbf{p}\|^2$$

  that we are making.
- To exclude the trivial solution $\mathbf{p} = \mathbf{0}$, we additionally constrain the norm of $\mathbf{p}$ as

$$\|\mathbf{p}\| = 1.$$

Constrained optimization

Constrained least squares

Find $\mathbf{p}$ that minimizes $\|\mathbf{A}\mathbf{p}\|^2$ subject to $\|\mathbf{p}\| = 1$.

- Rewriting the constraint as $1 - \mathbf{p}^\mathsf{T}\mathbf{p} = 0$ allows us to write the problem using a Lagrange multiplier and derive the necessary condition

$$\begin{aligned} \frac{\partial}{\partial \mathbf{p}} \left(\mathbf{p}^\mathsf{T}\mathbf{A}^\mathsf{T}\mathbf{A}\mathbf{p} + \lambda(1 - \mathbf{p}^\mathsf{T}\mathbf{p}) \right) &= 0. \\ \Leftrightarrow \qquad 2\mathbf{A}^\mathsf{T}\mathbf{A}\mathbf{p} - 2\lambda\mathbf{p} &= 0. \end{aligned}$$

- Eigenvalue problem $(\mathbf{A}^\mathsf{T}\mathbf{A})\mathbf{p} = \lambda\mathbf{p}$ implies that $\mathbf{p}$ is an eigenvector of $\mathbf{A}^\mathsf{T}\mathbf{A}$ and λ an eigenvalue.
- Which eigenvector/eigenvalue pair to pick?

Constrained optimization

- Recall that we are minimizing $\|\mathbf{A}\mathbf{p}\|^2 = \mathbf{p}^\mathsf{T}\mathbf{A}^\mathsf{T}\mathbf{A}\mathbf{p}$.
- This means that we can rewrite the squared error as

$$\|\mathbf{A}\mathbf{p}\|^2 = \mathbf{p}^\mathsf{T}\lambda\mathbf{p} = \lambda,$$

where the last step used the constraint $\|\mathbf{p}\| = 1$.

Homogeneous least squares

To minimize $\|\mathbf{A}\mathbf{p}\|^2$ subject to $\|\mathbf{p}\| = 1$, we thus choose $\mathbf{p}$ as the eigenvector of $\mathbf{A}^\mathsf{T}\mathbf{A}$ with the smallest eigenvalue λ.

Homogeneous least squares: SVD

- Perform singular value decomposition (SVD), *i. e.* $\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^\mathsf{T}$, where $\mathbf{U} \in \mathbb{R}^{m \times n}$ and $\mathbf{V} \in \mathbb{R}^{n \times n}$ are orthonormal and $\mathbf{S} \in \mathbb{R}^{n \times n}$ is a diagonal matrix with descending singular values along the diagonal.
- Now we have that

$$\mathbf{A}^\mathsf{T}\mathbf{A} = (\mathbf{U}\mathbf{S}\mathbf{V}^\mathsf{T})^\mathsf{T}\mathbf{U}\mathbf{S}\mathbf{V}^\mathsf{T} = \mathbf{V}\mathbf{S}^\mathsf{T}\mathbf{U}^\mathsf{T}\mathbf{U}\mathbf{S}\mathbf{V}^\mathsf{T} = \mathbf{V}\mathbf{S}^2\mathbf{V}^\mathsf{T}.$$

- The eigenvectors of $\mathbf{A}^\mathsf{T}\mathbf{A}$ are the right singular vectors of $\mathbf{A}$ (*i. e.* columns of $\mathbf{V}$) .
- The eigenvalues of $\mathbf{A}^\mathsf{T}\mathbf{A}$ are the square of the singular values of $\mathbf{A}$.
- To find the eigenvector of $\mathbf{A}^\mathsf{T}\mathbf{A}$ with the smallest eigenvalue, we compute the last right-singular vector of $\mathbf{A}$.

Camera calibration

- Note: for *coplanar points* that satisfy $\mathbf{\Pi}^{\mathrm{T}}\mathbf{X_i} = 0$
  we will get degenerate solutions $(\mathbf{\Pi}, 0, 0)$, $(0, \mathbf{\Pi}, 0)$ or $(0, 0, \mathbf{\Pi})$

$$\begin{bmatrix} 0 & -\mathbf{X}_1^{\mathrm{T}} & y_1\mathbf{X}_1^{\mathrm{T}} \\ \mathbf{X}_1^{\mathrm{T}} & 0 & -x_1\mathbf{X}_1^{\mathrm{T}} \\ \vdots & \vdots & \vdots \\ 0 & -\mathbf{X}_n^{\mathrm{T}} & y_n\mathbf{X}_n^{\mathrm{T}} \\ \mathbf{X}_n^{\mathrm{T}} & 0 & -x_n\mathbf{X}_n^{\mathrm{T}} \end{bmatrix} \begin{pmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{pmatrix} = \mathbf{0} \qquad \Leftrightarrow \qquad \mathbf{A}\mathbf{p} = \mathbf{0}$$

Camera calibration

- Once we've recovered the values of the camera matrix, we still have to figure out the *intrinsic* and *extrinsic* parameters
  - First split the projection matrix into a 3x3 matrix and a 3x1 vector

$$\mathbf{P} = [\mathbf{M}|\mathbf{m}] = [\mathbf{KR}| - \mathbf{KR}\tilde{\mathbf{C}}]$$

  - Next, decompose $\mathbf{M}$ into upper triangular part $\mathbf{K}$ (calibration) and orthonormal part $\mathbf{R}$ (rotation) using RQ-decomposition (see e.g. Golub & Van Loan, *Matrix Computations*)
    - if required, extract 3 rotation axis and angle from $\mathbf{R}$ - its columns are the axes of the rotated coordinate system)
  - Finally, find $\mathbf{C}$ as the nullspace of $\mathbf{P}$ by means of SVD.

Camera calibration

- What have we got now?
  - We have a projection matrix $\mathbf{P}$, which maps any point in the 3D world coordinate frame onto the (infinite) image plane of the camera
  - Given a 3D point in the world, we can find out where its projection is in the image
  - One goal of vision: given a point in the images, find out where there pre-image is in the world

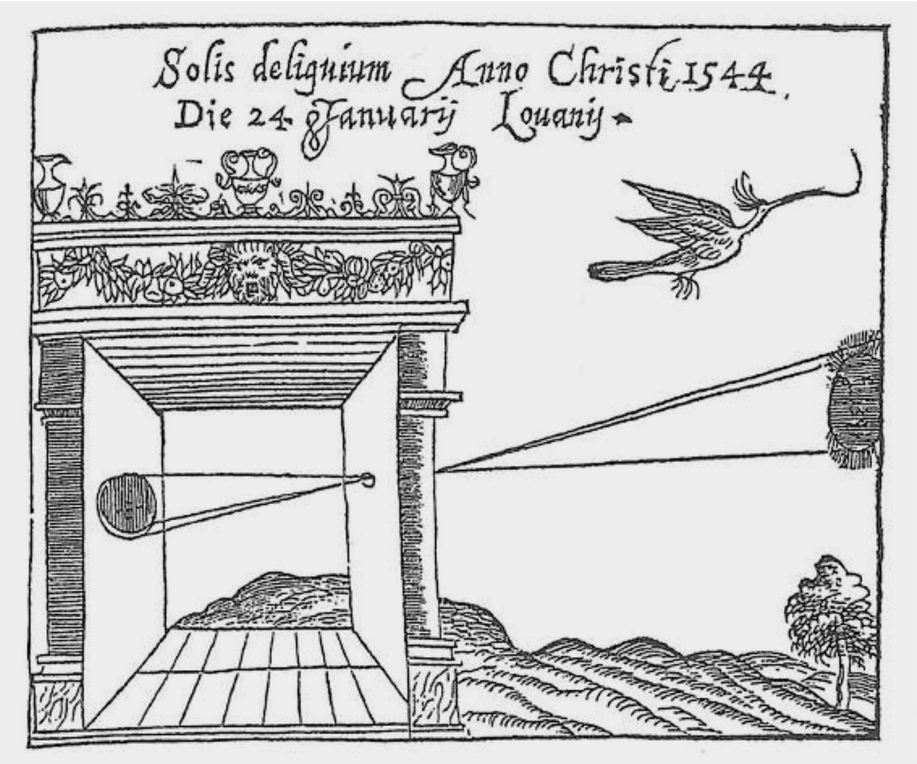
- We cannot do this from one image
  - The depth is lost, the projection matrix can only tell us the ray through an image point $\mathbf{x}$

Building a real camera

[from S. Lazebnik]

Camera Obscura

- Basic principle known to Mozi (470-391 BC) and Aristoteles (384-322 BC)
- Described by Ibn al-Haytham (Alhazen) in his "Book of optics" (around 1000AD).
- Drawing aid for artists: described by Leonardo da Vinci (1452-1519)

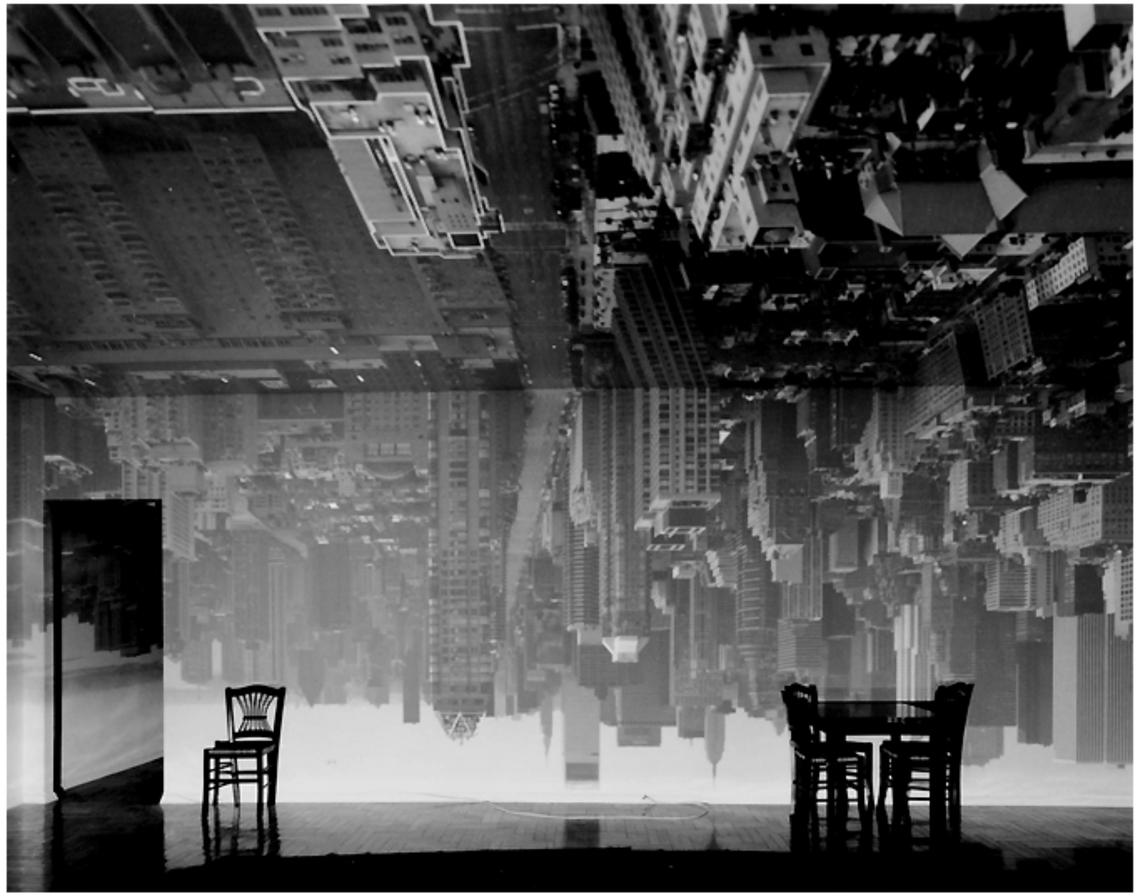


Gemma Frisius, 1558

[from A. Efros]

Abelardo Morell

- Camera Obscura Image of Manhattan View Looking South in Large Room, 1996



After scouting rooms and reserving one for at least a day, Morell masks the windows except for the aperture. He controls three elements: the size of the hole, with a smaller one yielding a sharper but dimmer image; the length of the exposure, usually eight hours; and the distance from the hole to the surface on which the outside image falls and which he will photograph. He used 4 x 5 and 8 x 10 view cameras and lenses ranging from 75 to 150 mm.

After he's done inside, it gets harder. "I leave the room and I am constantly checking the weather, I'm hoping the maid reads my note not to come in, I'm worrying that the sun will hit the plastic masking and it will fall down, or that I didn't trigger the lens."

From Grand Images Through a Tiny Opening, Photo District News, February 2005

http://www.abelardomorell.net/camera_obscura1.html

[from S. Lazebnik]

Home-made pinhole camera

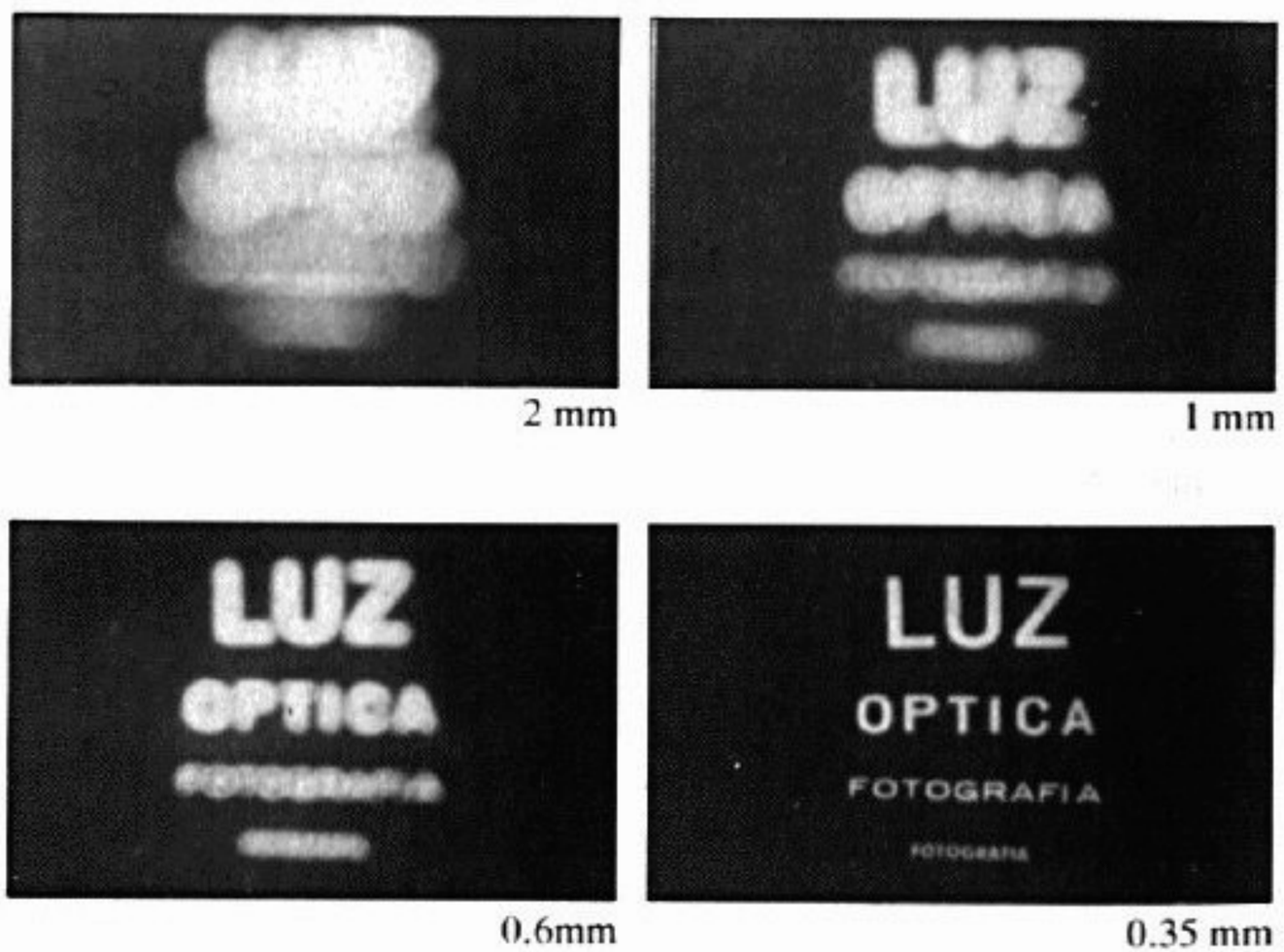
Why so blurry?

http://www.debevec.org/Pinhole/

[from A. Efros]

Shrinking the aperture

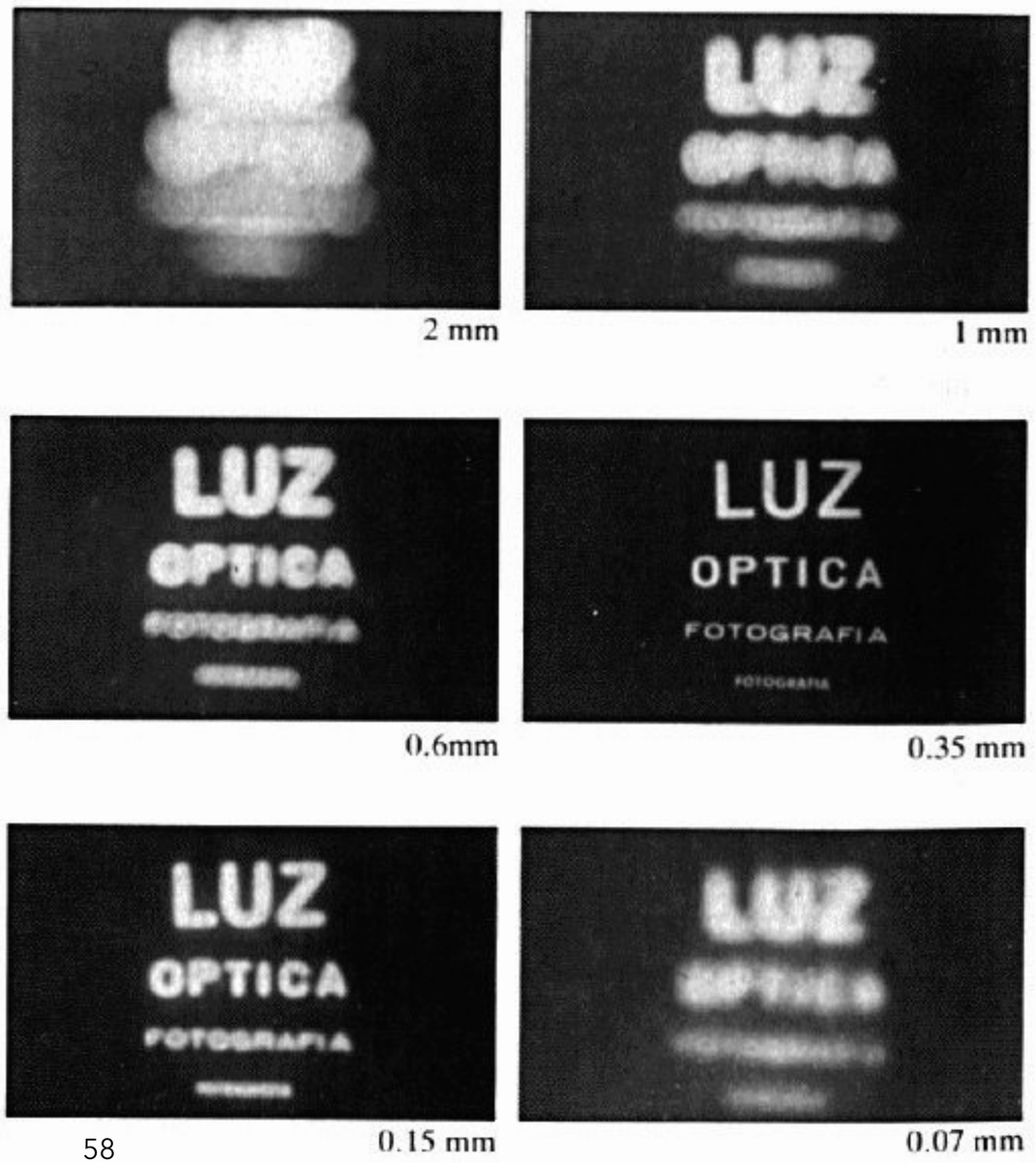
- Why not make the aperture as small as possible?
  - Less light gets through



[from Steve Seitz]

Shrinking the aperture

- Why not make the aperture as small as possible?
  - Less light gets through
  - Diffraction effects…



Readings for next week

- Image processing (Ch. 3)
  - Point operations - 3.1
  - Linear filters - 3.2
  - Neighborhood operators - 3.3
  - Fourier transform - 3.4