计算机视觉与模式识别

Computer Vision and Pattern Recognition

- Camera Model & Calibration

袁泽剑

人工智能与机器人研究所

Institute of Artificial Intelligence and Robotics

Email: yuan.ze.jian@xjtu.edu.cn

科学馆102室

Syllabus

- Physiology and theories of Vision (1)
- **Image Formation and Camera Model (1)**
  - **Image formation; Camera model;**
  - **Camera parameters; Calibration**
- Stereo Vision & SFM (3)
- Motion / Optical flow / tracking (2)
- Image Filtering & Structure Extraction (2)
- Local Features & Image Matching (2)
- Segmentation (Clustering /Grouping) (2)
- Visual Recognition (4)

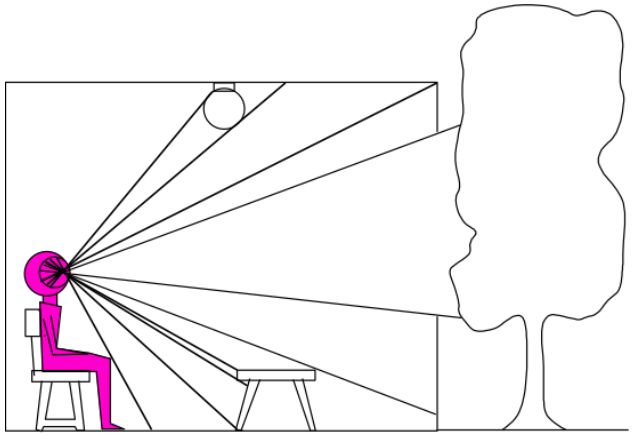
Background

- 感知3D环境中物体的几何信息是计算机视觉的目标之一。92年以来最大的理论成果：基于几何的分层重建(拓扑、射影、仿射、欧氏)。
- 90s年代中期以来，计算机视觉界，将射影几何、仿射几何、欧氏几何的描述方法，系统引入到基于几何的视觉建模中，这三种描述手段完美地对应于由粗到细的三维物体几何描述，降低了对相机参数了解的要求，提高了系统的鲁棒性。目前，基于几何的视觉计算方法与理论体系日趋成熟。
- 近年来，主要关注多摄像机系统几何建模及其计算方法、宽极线或大视点重建问题；多约束条件、弱标定情况下的全局优化方法。

Imaging

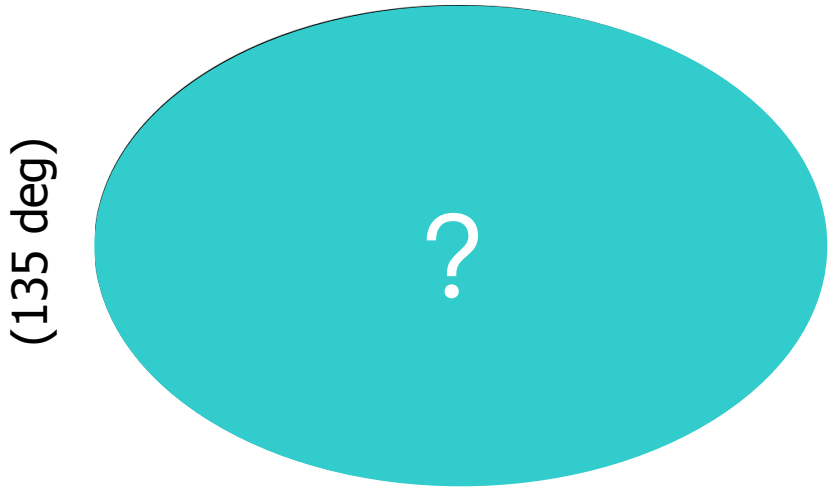
- **What do we see?**

3D world



Point of observation

2D image

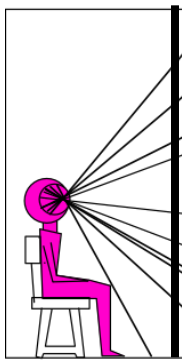


Imaging

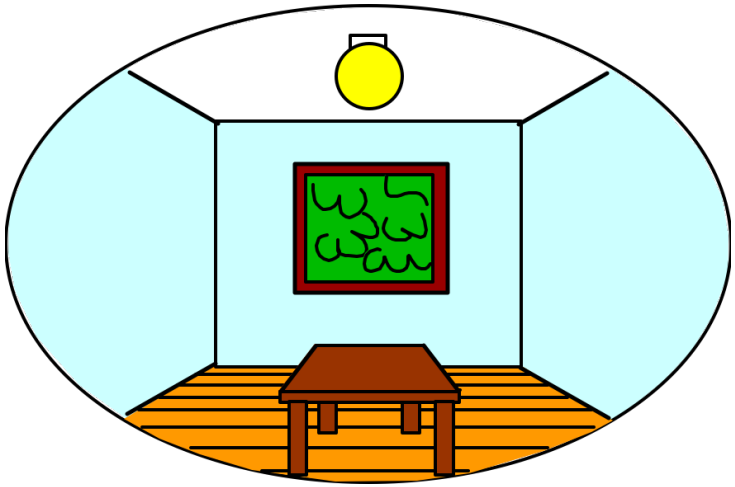
- **What do we see?**

3D world

2D image



Painted backdrop

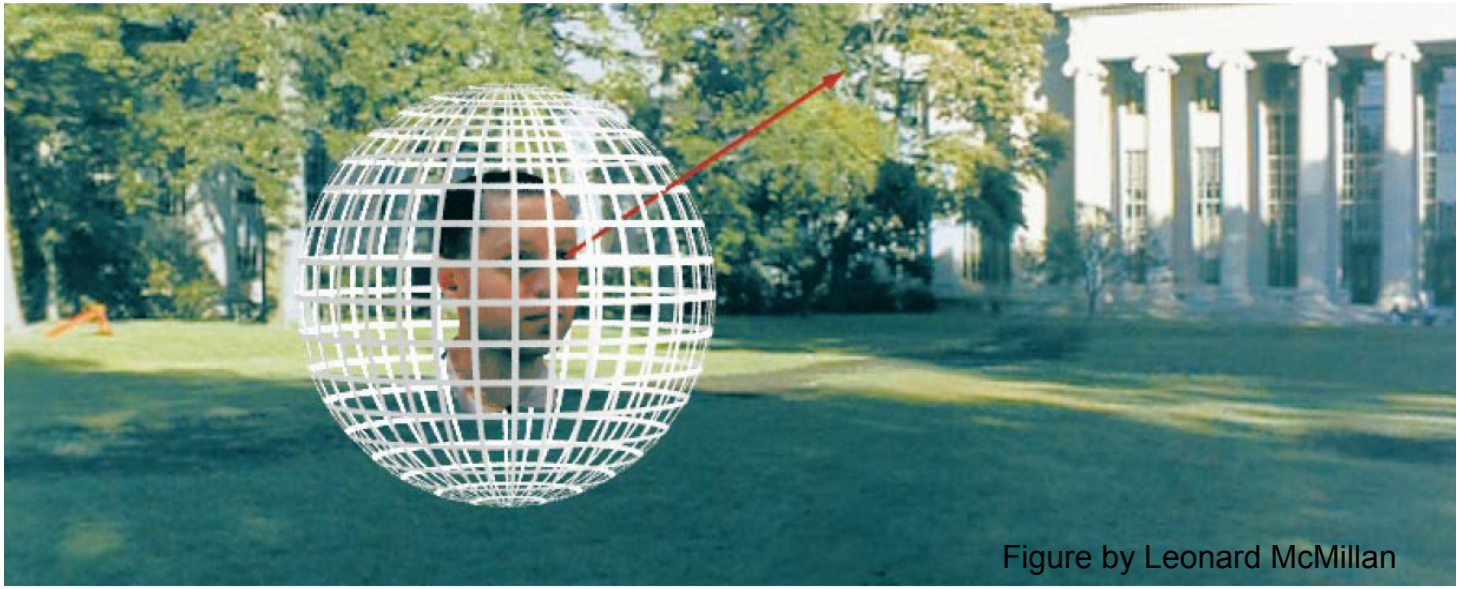


Figures © Stephen E. Palmer, 2002

How to represent ?

Imaging

The Plenoptic Function

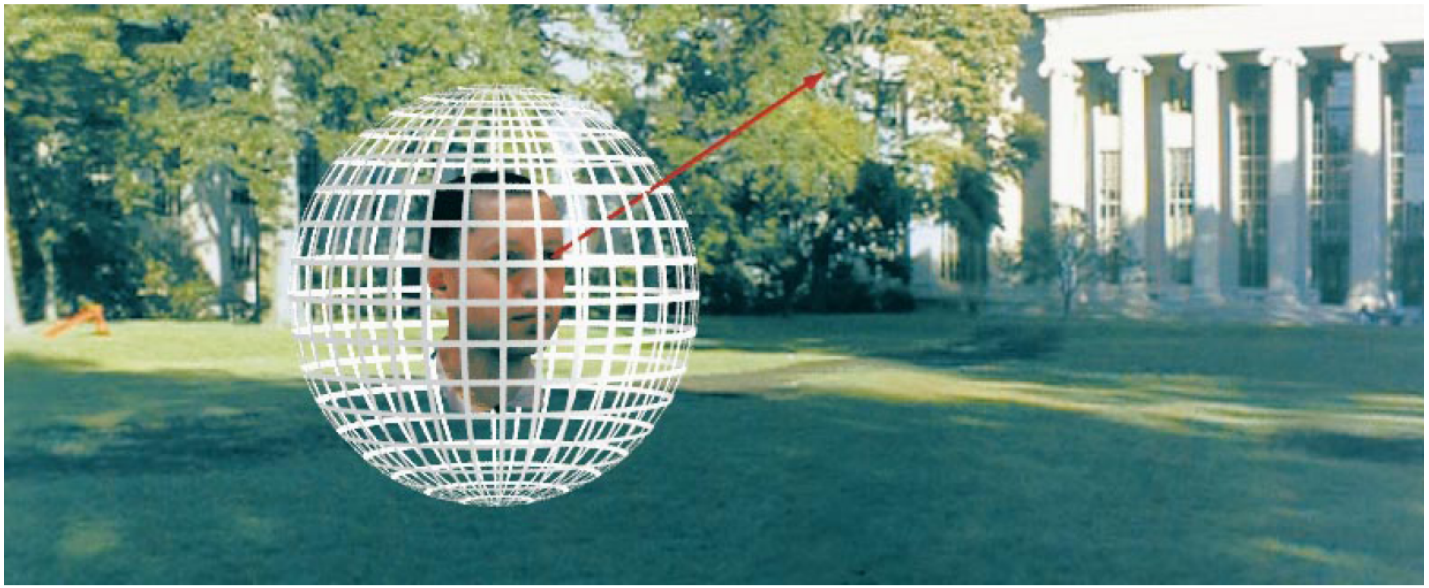


- What is the set of all things that we can ever see?
  - The Plenoptic Function (Adelson & Bergen)

- Parameterize everything that he can see…

Imaging

- **Grayscale snapshot**



- is intensity of light $f(\theta,\phi)$
  - ✓ A single view point & A single time
  - ✓ Averaged over the wavelengths of the visible spectrum
- (can also do f(x,y), but spherical coordinate are nicer)

Imaging

- **Color snapshot**

- is intensity of light $f(\theta,\phi,\lambda)$
  - ✓ A single view point & A single time
  - ✓ As a function of wavelength

Imaging

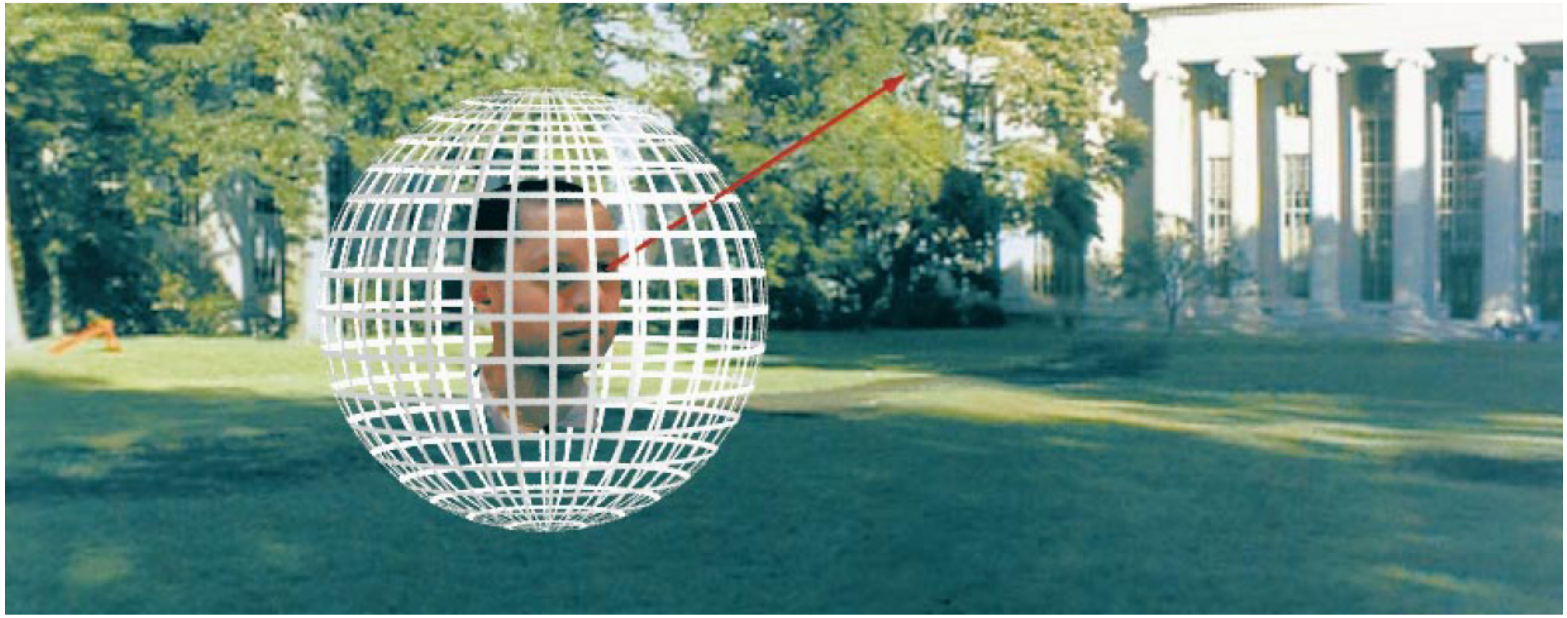
- **Spherical Panorama**

http://www.panoramas.dk/fullscreen3/f1.html

- All light rays through a point form a panorama
- Totally captured in a 2D array -- $f(\theta,\phi)$

Imaging

A Movie

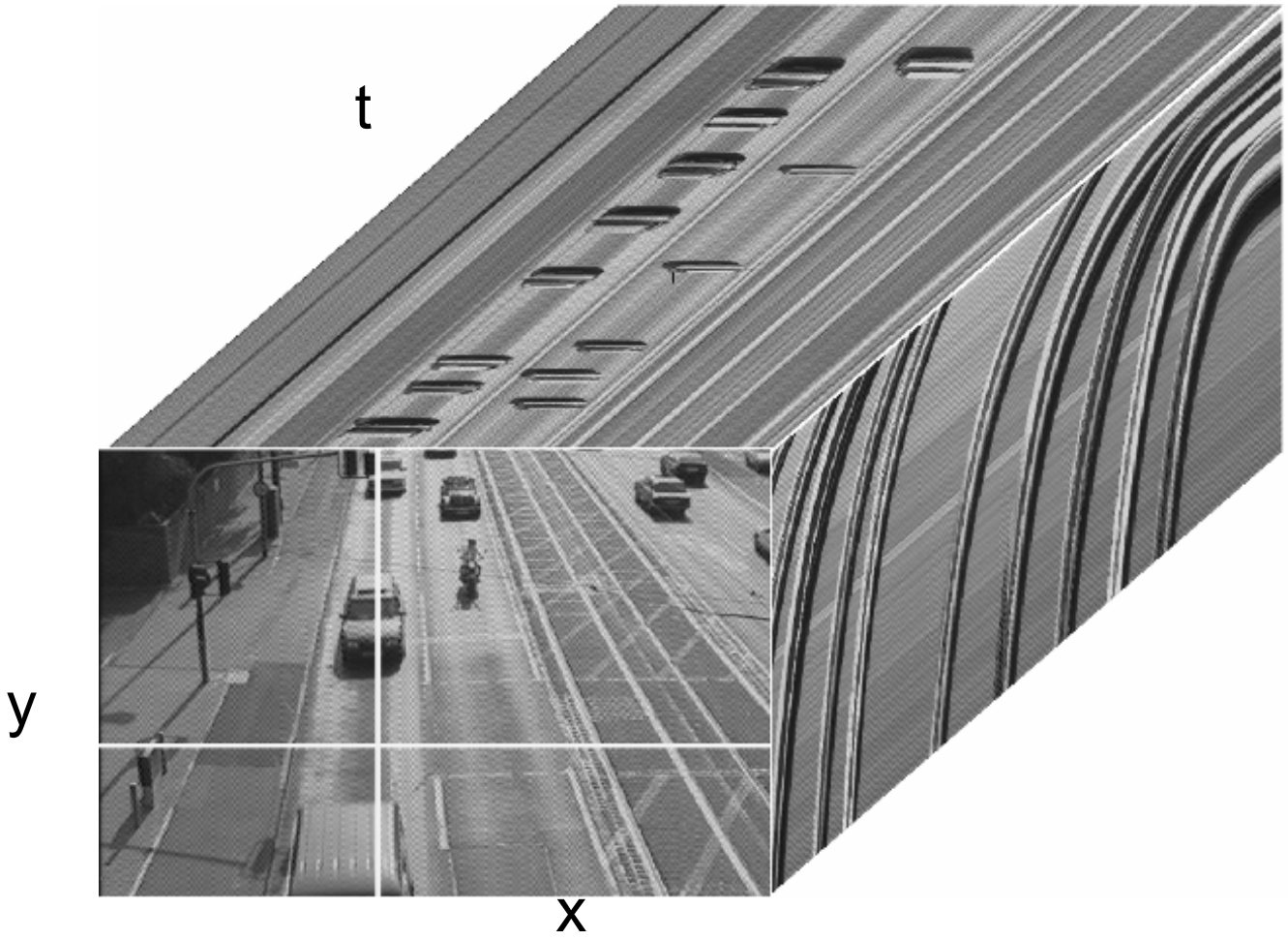


- is intensity of light
  - A single view point & Time
  - As a function of wavelength

$$f(\theta,\phi,\lambda,t)$$

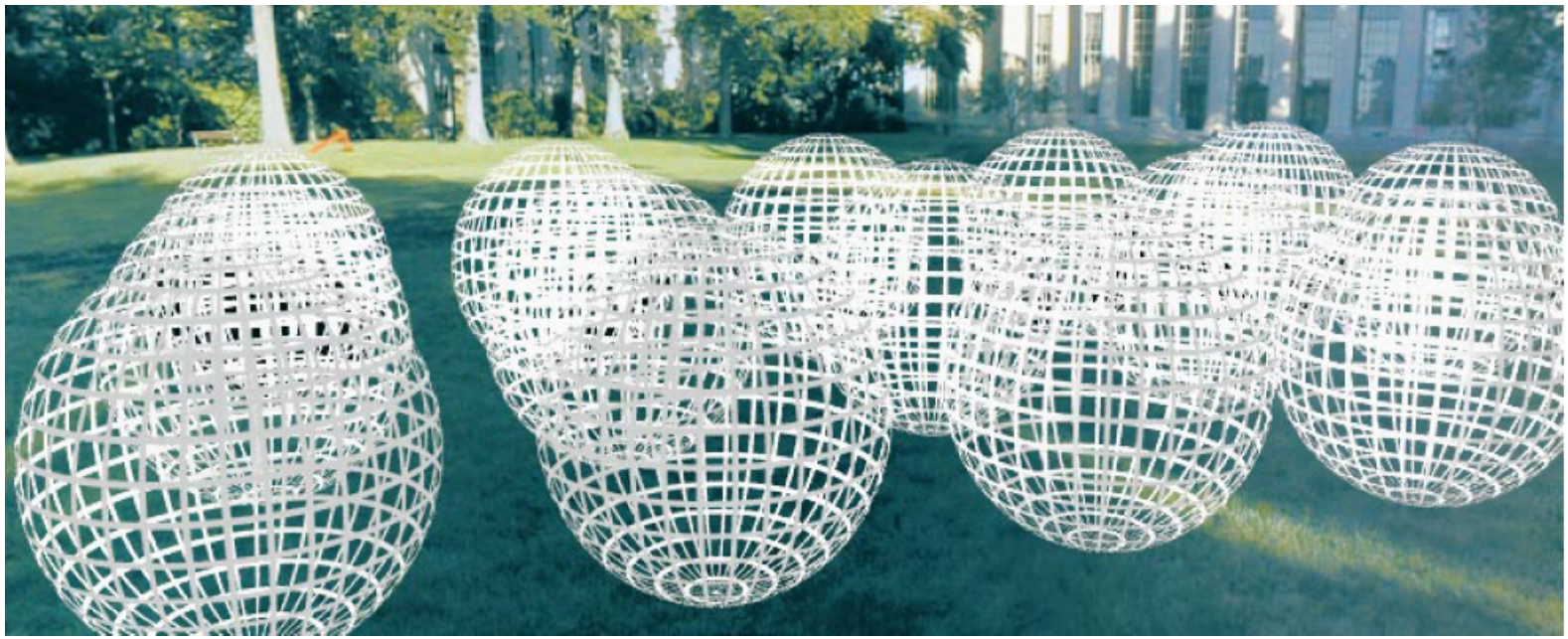
Imaging

- Space-time images



Imaging

- **Holographic movie** （全息电影）

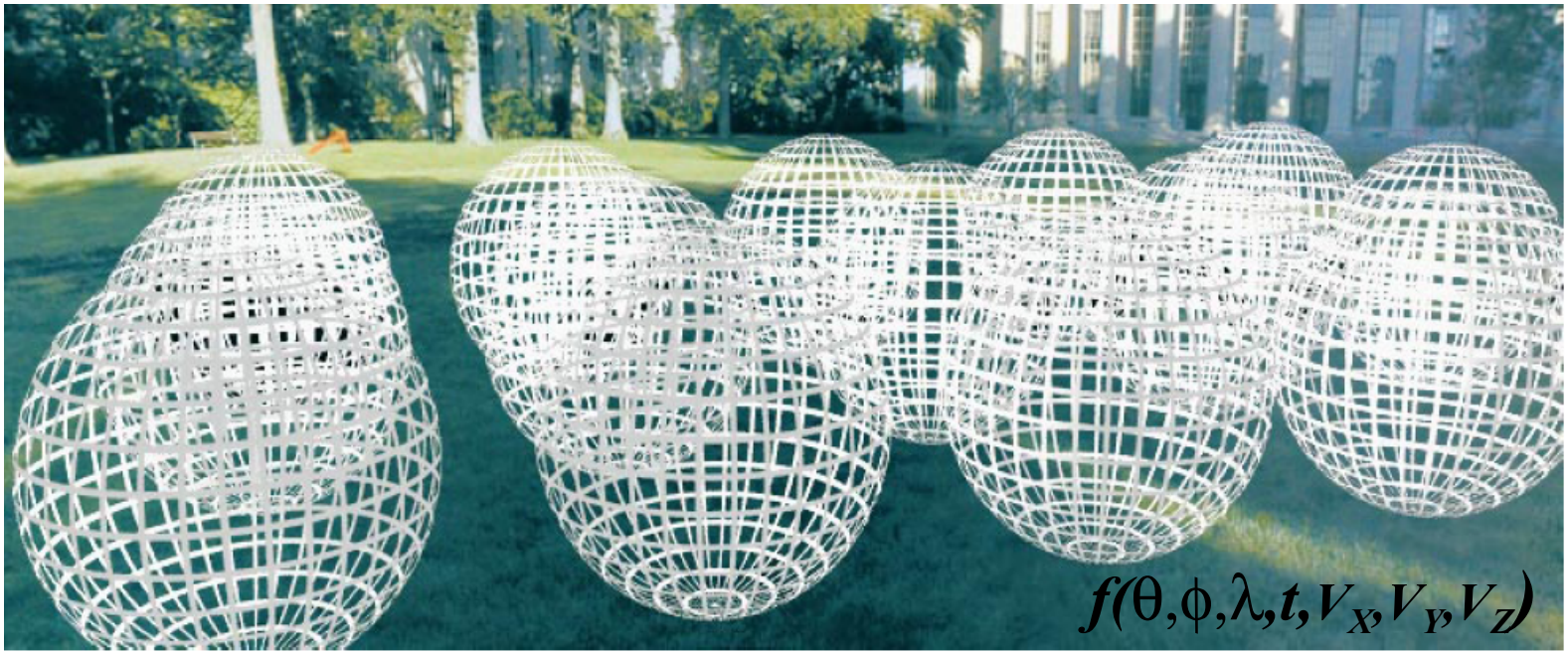


- is intensity of light
  - ✓ ANY viewpoint & Time
  - ✓ As a function of wavelength

$$f(\theta,\phi,\lambda,t,V_X,V_Y,V_Z)$$

Imaging

· The Plenoptic Function



- Can reconstruct every possible view, at every moment, from every position, at every wavelength
- It completely captures our visual reality! Not bad for a function…

Imaging

· Image formation

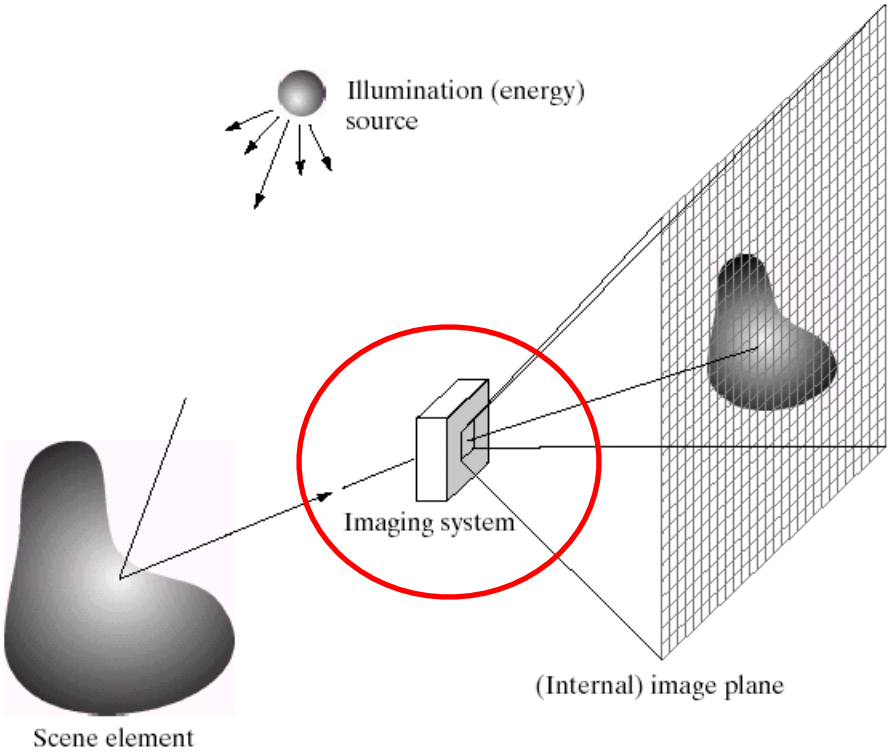
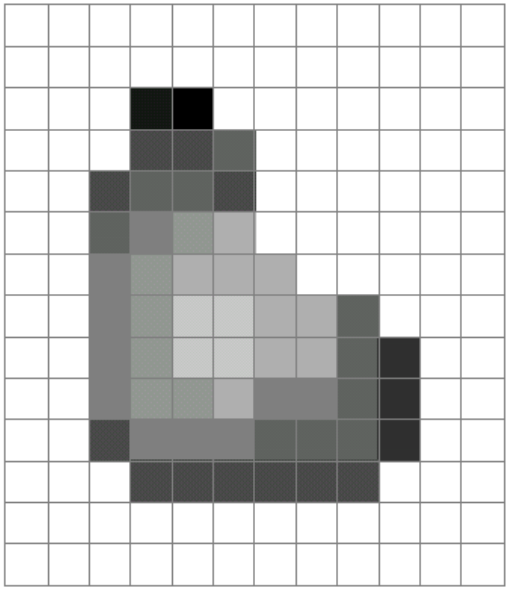


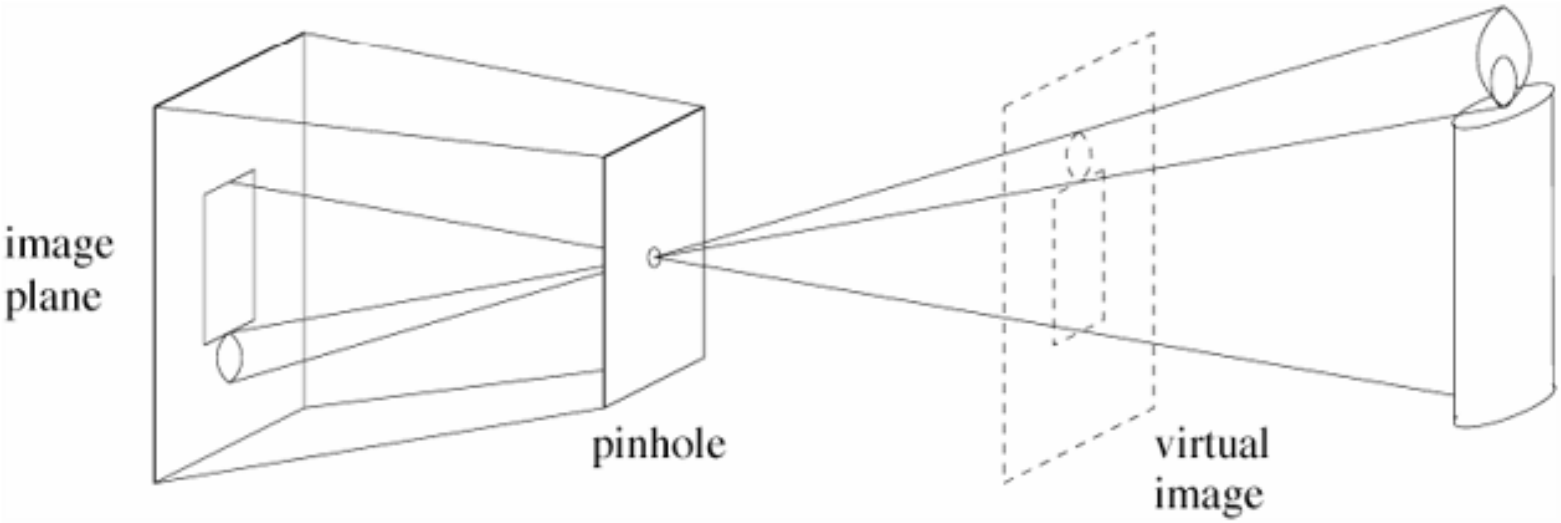
Image plane



Digital Camera

Pinhole Camera

- (Simple) standard and abstract model
  - Box with a small hole in it
  - Works in practice



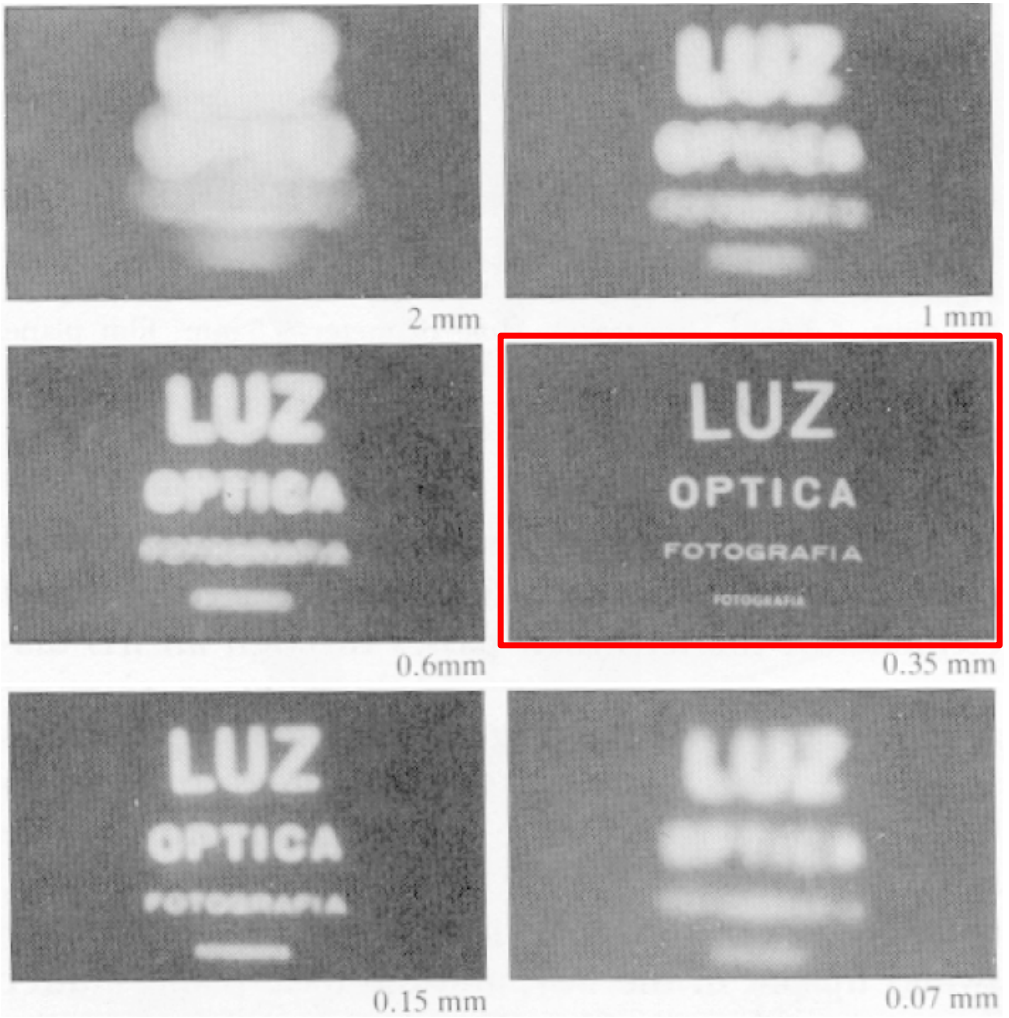
Source: Forsyth & Ponce

Imaging

- **Pinhole Size & Image Quality**

Pinhole too big - many directions are averaged, blurring the image

Pinhole too small- diffraction effects blur the image

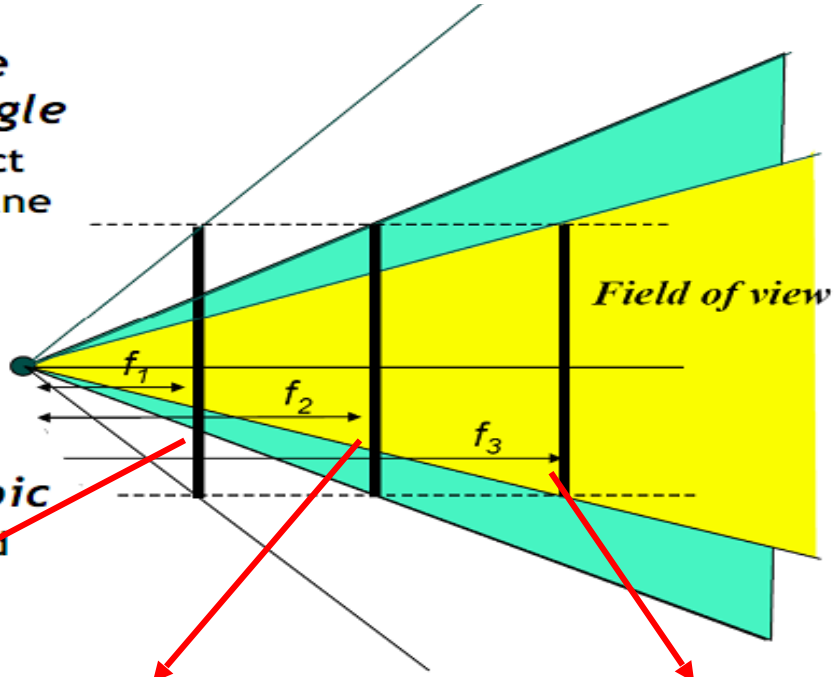


Diffraction: 衍射

Imaging (Pinhole Camera)

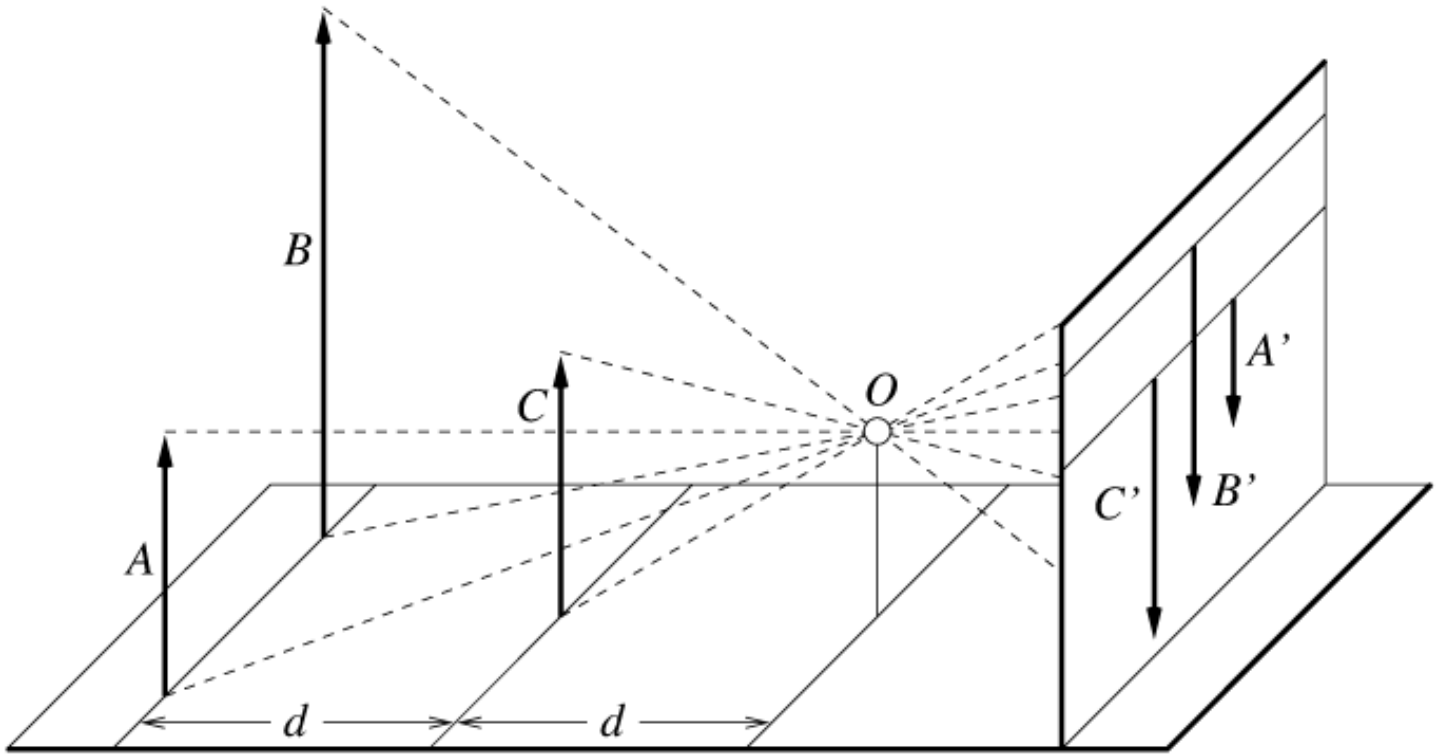
Visual field & Focal length

- As *f* gets smaller, image becomes more *wide angle*
  - More world points project onto the finite image plane
- As *f* gets larger, image becomes more *telescopic*
  - Smaller part of the world projects onto the finite image plane



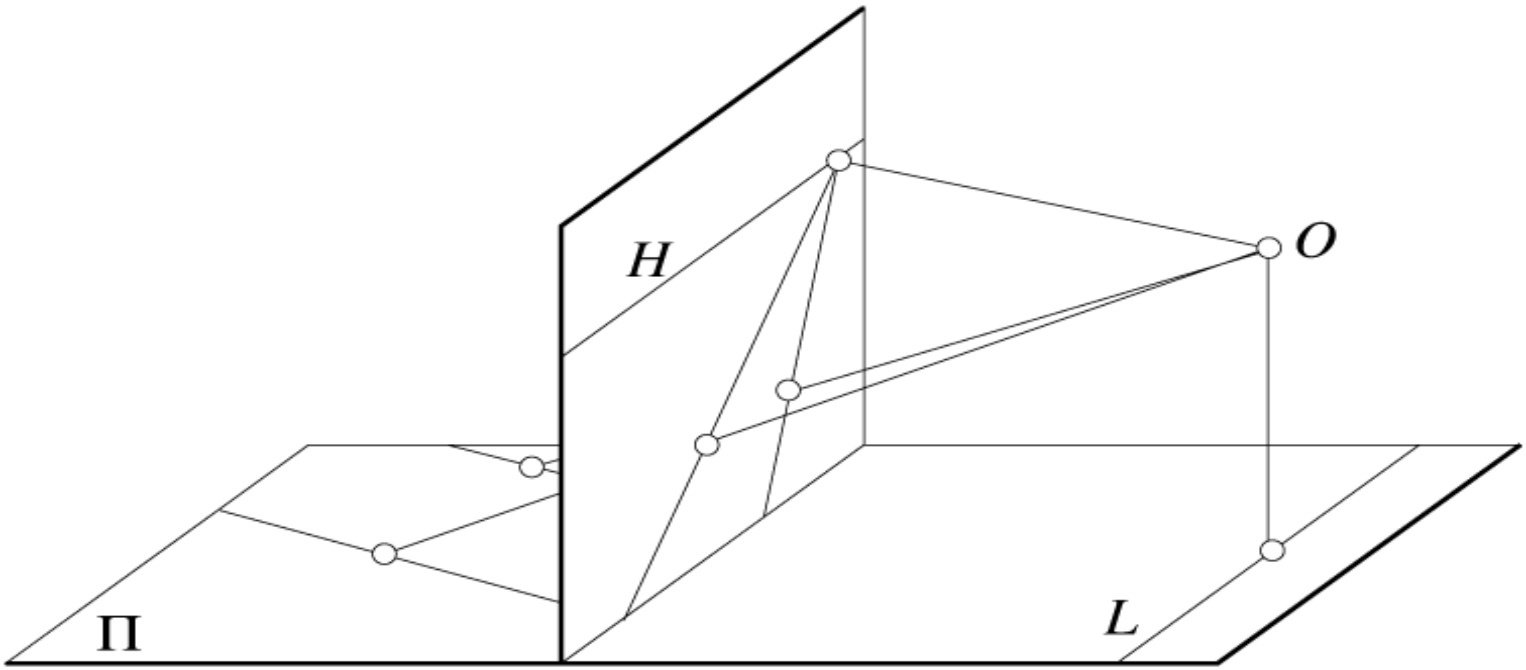
Imaging (Pinhole Camera)

- **Scale relation**
  - **Distance, Size, Zoom**



Imaging (Pinhole Camera)

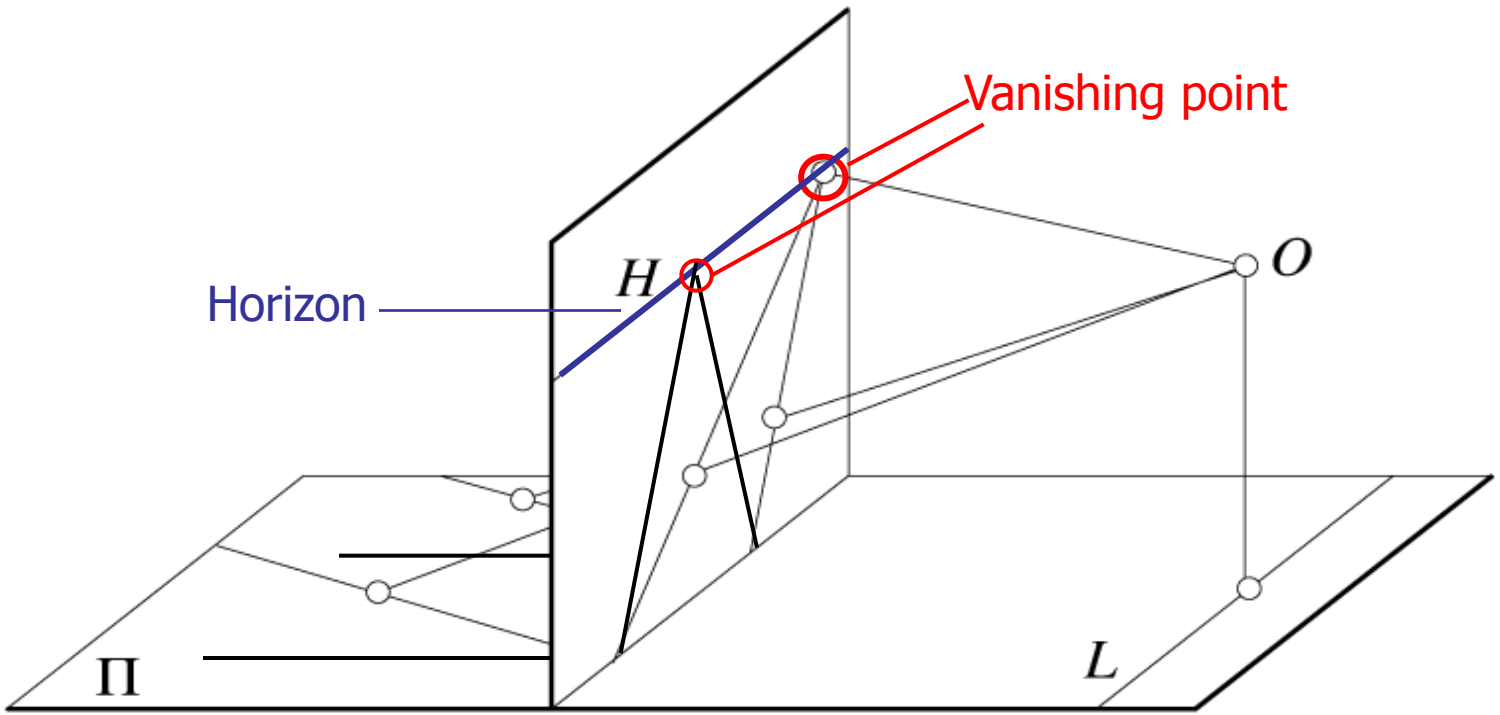
- **Parallel lines meet**
  - ➢ Each set of parallel lines (=direction) meets at a different point.
  - ➢ Sets of parallel lines on the same plane lead to collinear vanishing points.



Imaging (Pinhole Camera)

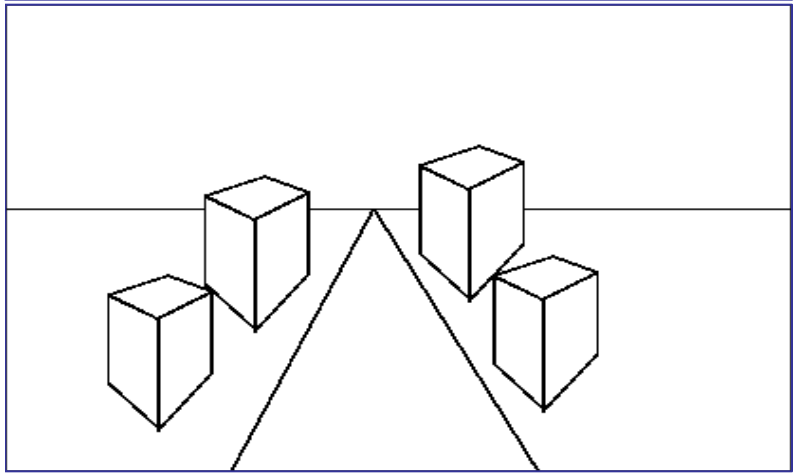
Vanishing point & Horizon

- Each set of parallel lines (=direction) meets at a different point.
  - ✓The vanishing point for this direction
- Sets of parallel lines on the same plane lead to collinear vanishing points.
  - ✓The line is called the horizon for that plane

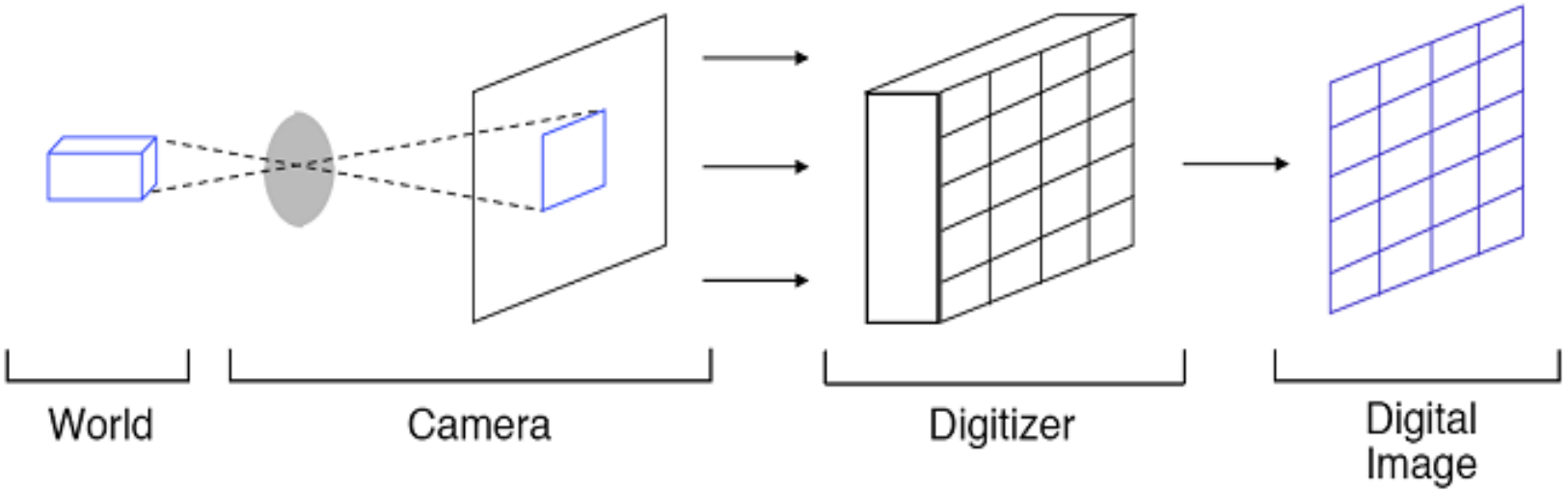


Imaging (Pinhole Camera)

- **Vanishing point & Horizon**



Digital Images

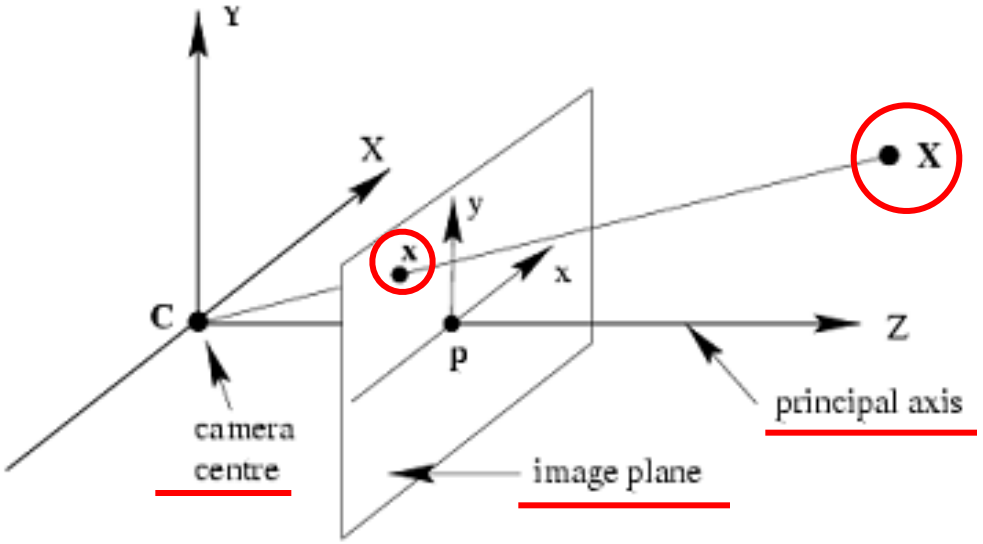


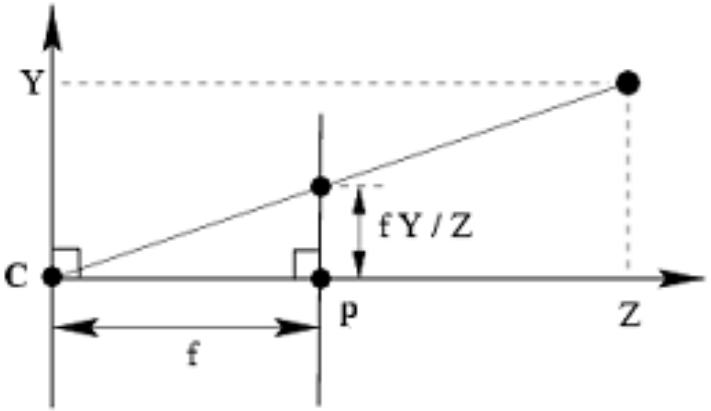
- Film is replaced by a sensor array
- Current technology: arrays of *charge coupled devices* (CCD)
- *Discretize* the image into pixels (space)
- *Quantize* light intensities into pixel values. (intensity)

Slide credit: Michael Black, B.Leibe

Pinhole Camera Model

Pinhole Camera Model





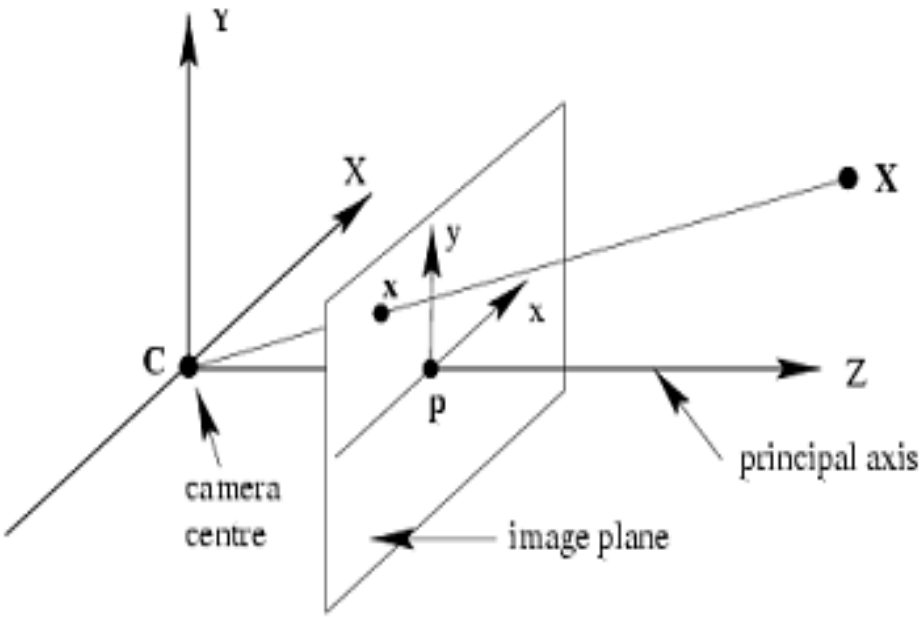
$$(X, Y, Z) \mapsto (fX/Z, fY/Z)$$

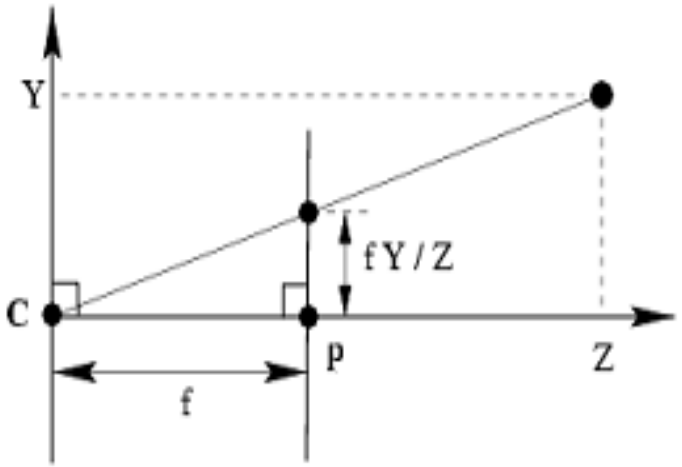
$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & & & 0 \\ & f & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

$$\mathbf{x} = \mathbf{PX}$$

Slide credit: Svetlana Lazebnik

Pinhole Camera Model





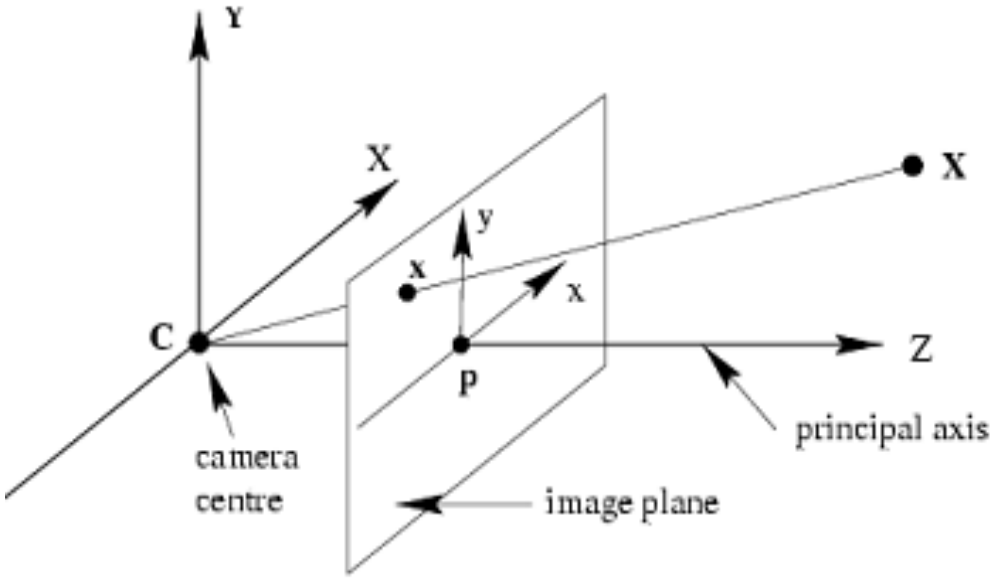
$$\begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & & & \\ & f & & \\ & & 1 & \end{bmatrix} \begin{bmatrix} 1 & & & 0 \\ & 1 & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

$$x = PX \qquad P = \mathrm{diag}(f, f, 1)[I \mid 0]$$

P: 3x4

Slide credit: Svetlana Lazebnik

Camera Coordinate System

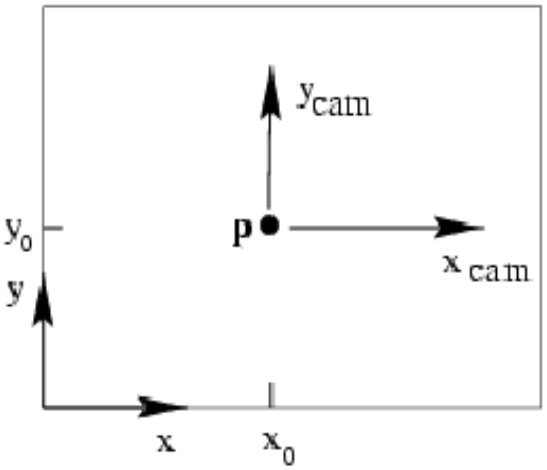


- *Principal axis*: line from the camera center perpendicular to the image plane
- *Normalized (camera) coordinate system*: camera center is at the origin and the principal axis is the z-axis
- *Principal point* (p): point where principal axis intersects the image plane (origin of normalized coordinate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

Slide credit: Svetlana Lazebnik

Pinhole Camera Model

Principal Point Offset

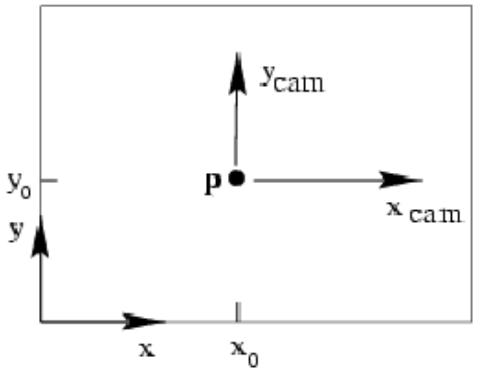


principal point: (p_x, p_y)

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

Slide credit: Svetlana Lazebnik

Pinhole Camera Model



principal point: (p_x, p_y)

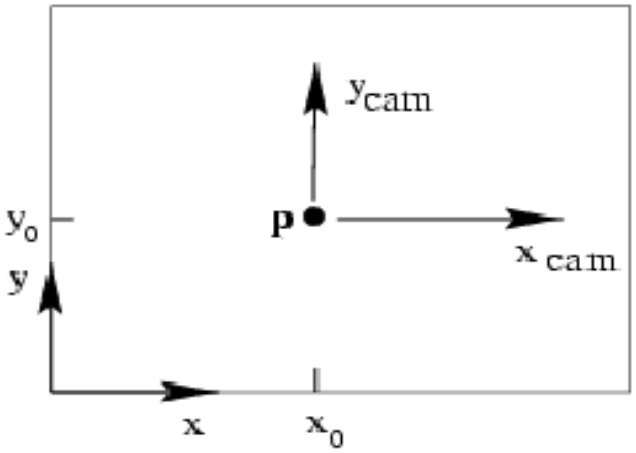
- Camera coordinate system: origin at the prinicipal point
- Image coordinate system: origin is in the corner

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

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

Slide credit: Svetlana Lazebnik

Pinhole Camera Model



principal point: (p_x, p_y)

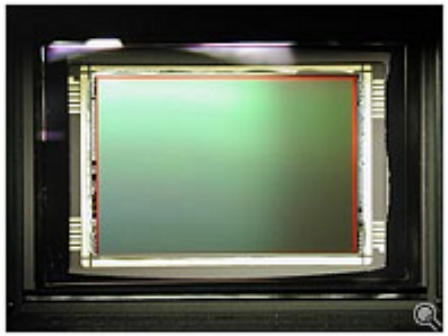
$$\begin{pmatrix} fX + Zp_x \\ fY + Zp_x \\ Z \end{pmatrix} = \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & 0 \\ & 1 & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

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

calibration matrix $\quad P = K[I \mid 0]$

Slide credit: Svetlana Lazebnik

Pixel Coordinates: Non-Square Pixels



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

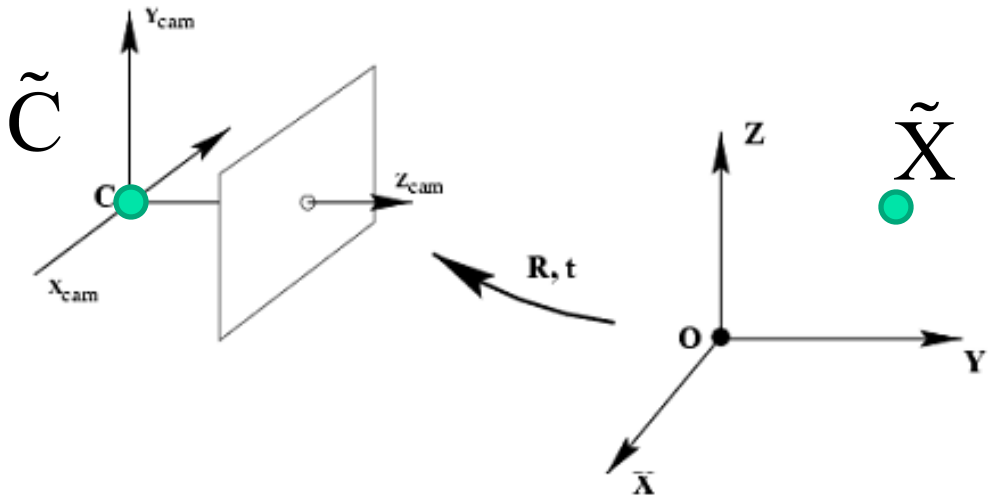
m_x pixels per meter in horizontal direction,
m_y pixels per meter in vertical direction

$$K = \begin{bmatrix} m_x & & \\ & m_y & \\ & & 1 \end{bmatrix} \begin{bmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} = \begin{bmatrix} \alpha_x & & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix}$$

pixels/m　　m　　pixels

Slide credit: Svetlana Lazebnik

Camera **Rotation** and **Translation**



- In general, the camera coordinate frame will be related to the world coordinate frame by a rotation and a translation

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

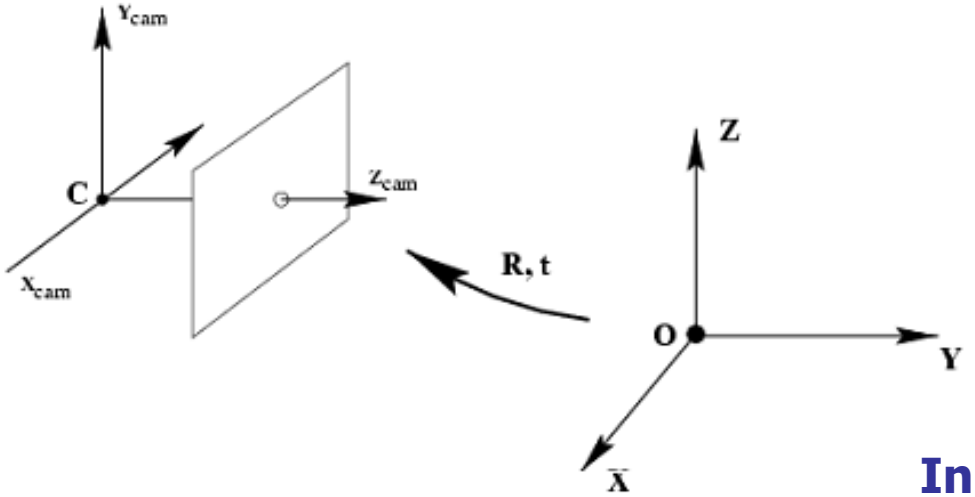
coords. of point in camera frame

coords. of a point in world frame (nonhomogeneous)

coords. of camera center in world frame

Slide credit: Svetlana Lazebnik

Camera Rotation and Translation



In non-homogeneous coordinates:

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

In homogeneous coordinates:

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

$$x = K\left[I \mid 0\right] X_{cam} = K\left[R \mid -R\tilde{C}\right] X \qquad \boxed{P = K\left[R \mid t\right]}, \qquad t = -R\tilde{C}$$

Slide credit: Svetlana Lazebnik

Camera Model

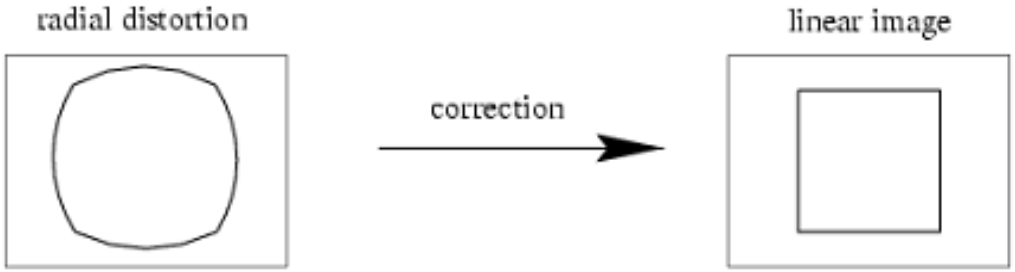
$$x=PX, \quad \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{PX}$$

$$\mathbf{P} = \mathbf{K}\left[\mathbf{R} \mid \mathbf{t}\right]$$

Summary: Camera Parameters

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

$$K = \begin{bmatrix} m_x & & \\ & m_y & \\ & & 1 \end{bmatrix} \begin{bmatrix} f & S & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} = \begin{bmatrix} \alpha_x & S & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix}$$



Slide credit: Svetlana Lazebnik

Summary: Camera Parameters

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

$$K = \begin{bmatrix} m_x & & \\ & m_y & \\ & & 1 \end{bmatrix} \begin{bmatrix} f & S & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} = \begin{bmatrix} \alpha_x & S & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix}$$

- **Extrinsic parameters**
  - Rotation R
  - Translation t
    (both relative to world coordinate system)

- **Camera projection matrix** $P = K[R \mid t] = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix}$

How many degrees of freedom does P have?

Slide credit: Svetlana Lazebnik

Camera Parameters: Degrees of Freedom

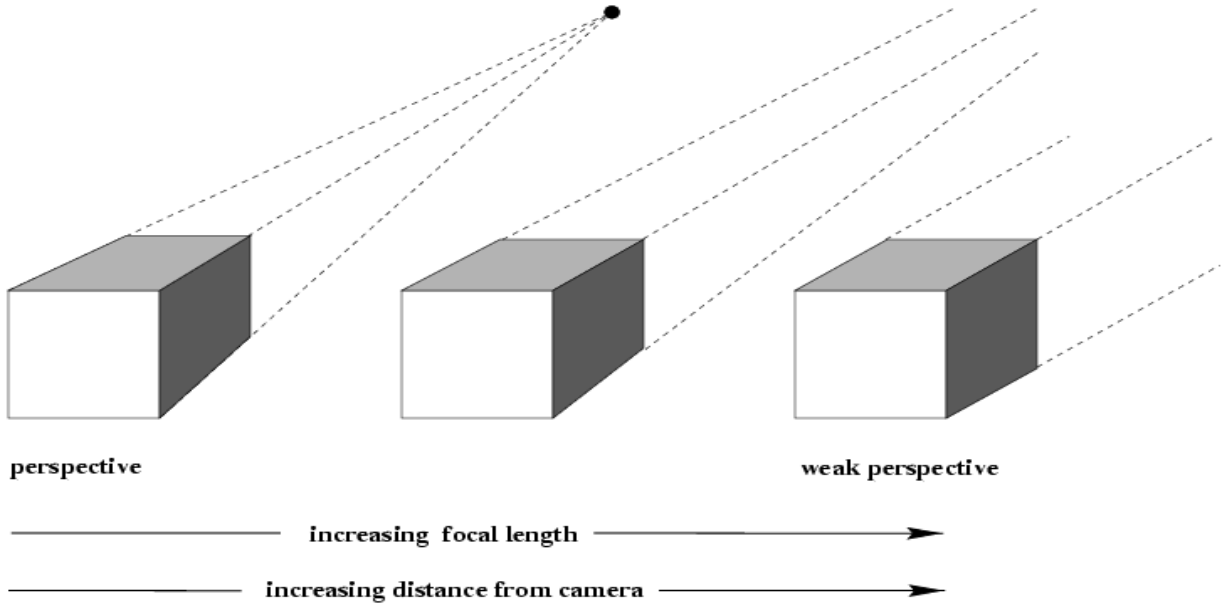
- **Intrinsic parameters** — **DoF**
  - Principal point coordinates — 2
  - Focal length — 1
  - Pixel magnification factors — 1
  - *Skew (non-rectangular pixels)* — 1
  - *Radial distortion*
- **Extrinsic parameters**
  - Rotation R — 3
  - Translation t — 3
    (both relative to world coordinate system)
- **Camera projection matrix**
  - ⇒ General pinhole camera: 9 DoF
  - ⇒ CCD Camera with square pixels: 10 DoF
  - ⇒ General camera: 11 DoF

$$K = \begin{bmatrix} m_x & & \\ & m_y & \\ & & 1 \end{bmatrix} \begin{bmatrix} f & s & p_x \\ & f & p_y \\ & & 1 \end{bmatrix} = \begin{bmatrix} \alpha_x & s & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix}$$

$$P = K[R \mid t]$$

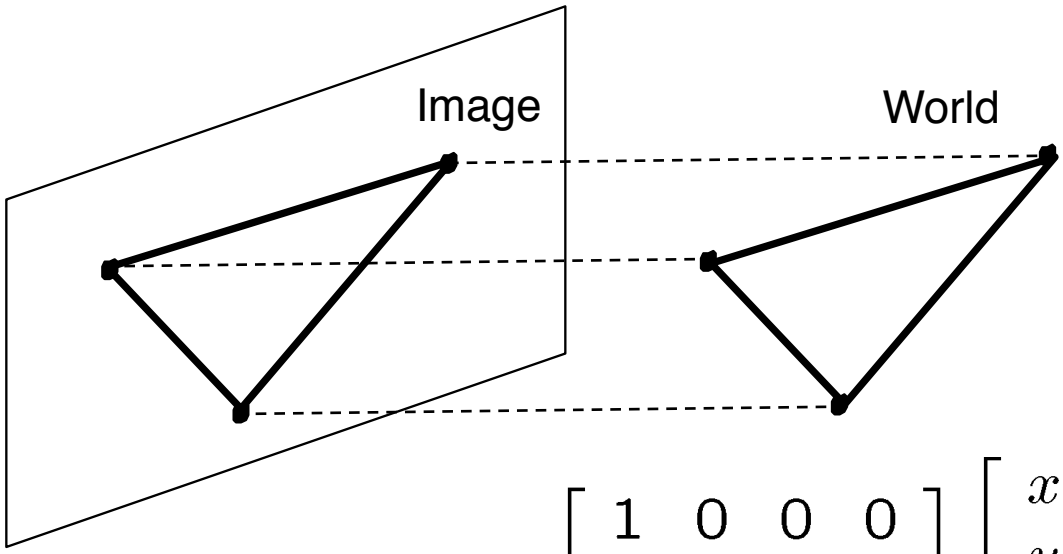
$$\begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix}$$

Other projection models



Orthographic projection

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



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

  - Also called “parallel projection”: (x, y, z) → (x, y)

Other types of projections

- Scaled orthographic
  – Also called “weak perspective”

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1/d \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1/d \end{bmatrix} \Rightarrow (dx, dy)$$

弱透视

- Affine projection
  – Also called “paraperspective”

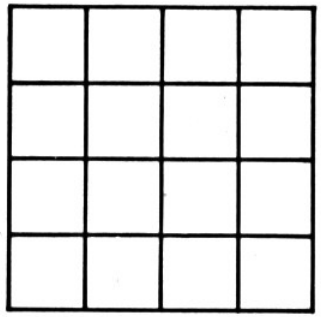
$$\begin{bmatrix} a & b & c & d \\ e & f & g & h \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

平行透视

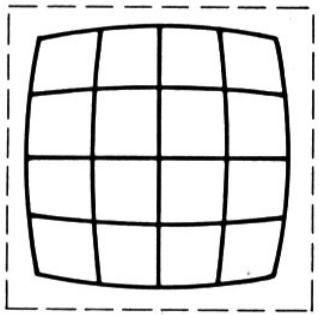
Camera Distortion

- Radial distortion of the image
  - *Caused by imperfect lenses*
  - *Deviations are most noticeable for rays that pass through the edge of the lens*

Barrel Distortion



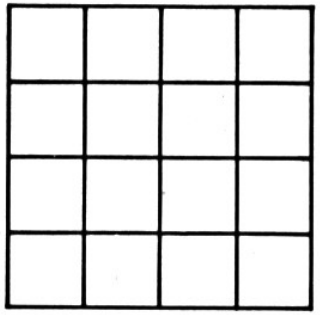
No distortion



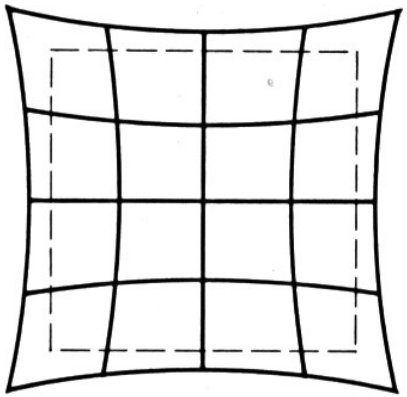
Barrel

Wide Angle Lens

Pin-Cushion Distortion



No distortion



Pin cushion

Telephoto lens

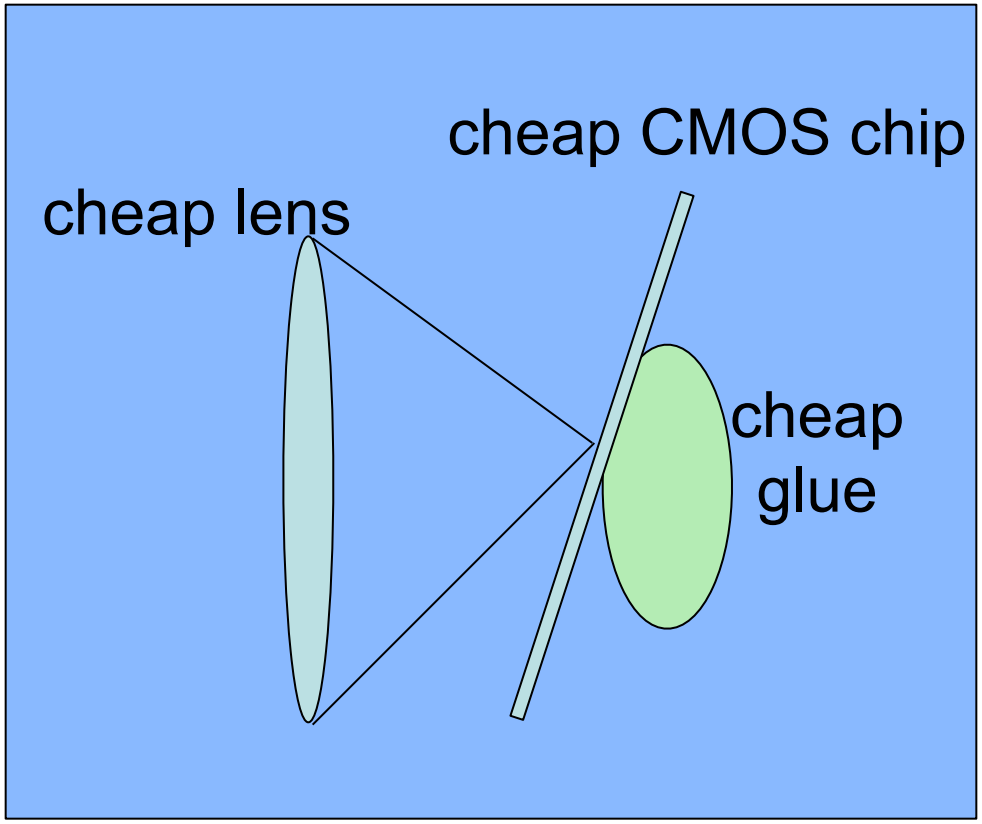
Telephoto lens : 长焦镜头

Correcting Radial Lens Distortions

Before

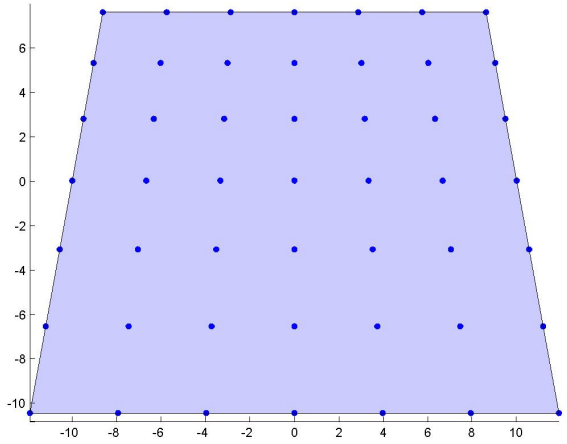
After

Tangential Distortion



cheap camera

image



正切畸变

Brown's distortion model

- **radial distortion**
- **tangential distortion (distortion caused by lens placement errors)**

$$(x_u, y_u) = f(x_d, y_d, x_c, y_c)$$

radial distortion *tangential distortion*

$$x_u = (x_d - x_c)(1 + K_1 r^2 + K_2 r^4 + \cdots) + (P_1(r^2 + 2(x_d - x_c)^2) + 2P_2(x_d - x_c)(y_d - y_c))(1 + P_3 r^2 + \cdots)$$

$$y_u = (y_d - y_c)(1 + K_1 r^2 + K_2 r^4 + \cdots) + (P_2(r^2 + 2(y_d - y_c)^2) + 2P_1(x_d - x_c)(y_d - y_c))(1 + P_3 r^2 + \cdots)$$

(x_u, y_u) — *undistorted image point as in ideal pinhole camera*

(x_d, y_d) — *distorted image point of camera with radial distortion*

(x_c, y_c) — *distortion center*

K_n — *n-th radial distortion coefficient*

P_n — *n-th tangential distortion coefficient*

- *typically K_1 is used or K_1, K_2, K_3, P_1, P_2*

Camera calibration

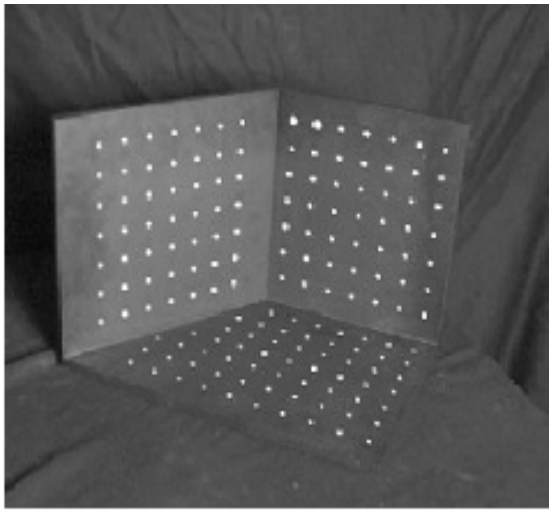
Camera calibration

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \sim \begin{bmatrix} f & 0 & x_c \\ 0 & f & y_c \\ 0 & 0 & 1 \end{bmatrix} [\mathbf{R}|\mathbf{t}] \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

- Estimate both intrinsic and extrinsic parameters
- Mainly, two categories:
  1. Using objects with known geometry as reference
  2. Self calibration (structure from motion)

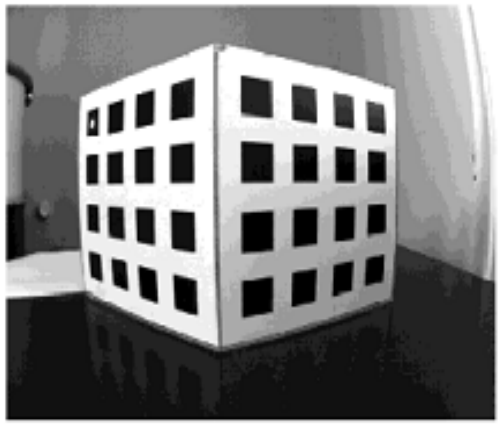
Calibrating a Camera

- Compute intrinsic and extrinsic parameters using observed camera data.



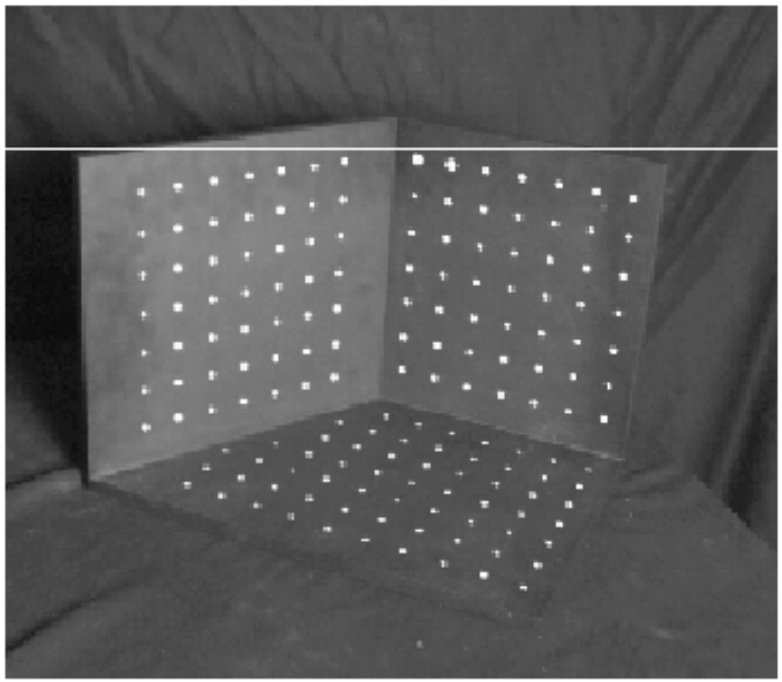
Main idea

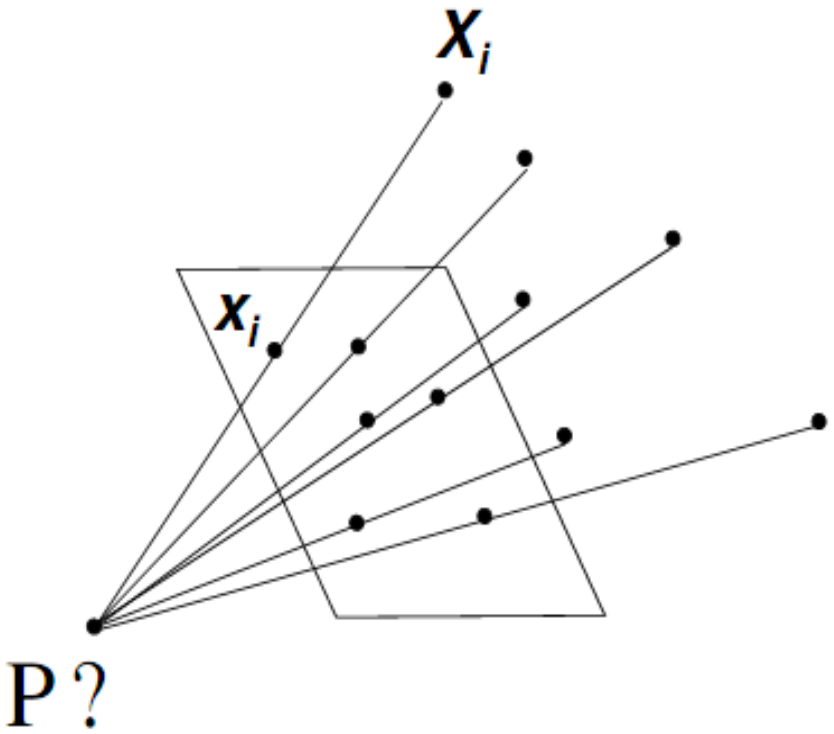
- Place "calibration object" with known geometry in the scene
- Get correspondences
- Solve for mapping from scene to image: estimate $P=P_{int}P_{ext}$



Camera Calibration

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



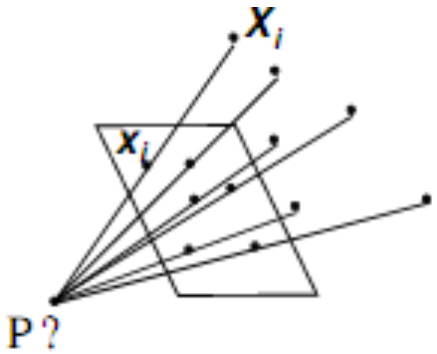


Camera Calibration: DLT Algorithm

$$\lambda \mathrm{x}_i = \mathrm{P}\mathrm{X}_i$$

$$\lambda \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \end{bmatrix} \begin{bmatrix} \mathrm{X}_{i,1} \\ \mathrm{X}_{i,2} \\ \mathrm{X}_{i,3} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathrm{P}_1^T \\ \mathrm{P}_2^T \\ \mathrm{P}_3^T \end{bmatrix} \mathrm{X}_i$$

$$\mathrm{x}_i \times \mathrm{P}\mathrm{X}_i = 0$$

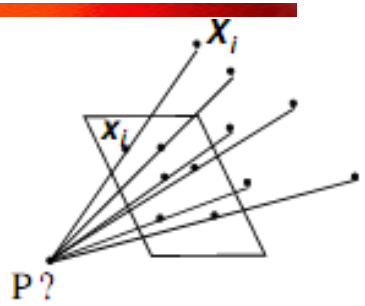


$$\begin{bmatrix} 0 & -\mathrm{X}_i^T & y_i \mathrm{X}_i^T \\ \mathrm{X}_i^T & 0 & -x_i \mathrm{X}_i^T \\ -y_i \mathrm{X}_i^T & x_i \mathrm{X}_i^T & 0 \end{bmatrix} \begin{pmatrix} \mathrm{P}_1 \\ \mathrm{P}_2 \\ \mathrm{P}_3 \end{pmatrix} = 0$$

Only two linearly independent equations

Camera Calibration: DLT Algorithm

$$\begin{bmatrix} 0^T & \mathrm{X}_1^T & -y_1\mathrm{X}_1^T \\ \mathrm{X}_1^T & 0^T & -x_1\mathrm{X}_1^T \\ \cdots & \cdots & \cdots \\ 0^T & \mathrm{X}_n^T & -y_n\mathrm{X}_n^T \\ \mathrm{X}_n^T & 0^T & -x_n\mathrm{X}_n^T \end{bmatrix} \begin{pmatrix} \mathrm{P}_1 \\ \mathrm{P}_2 \\ \mathrm{P}_3 \end{pmatrix} = 0$$



$$\mathrm{A}\mathrm{p} = 0$$

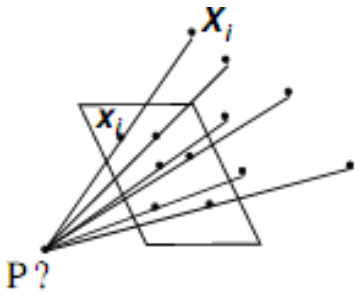
Solve using... SVD!

- **Notes**
  - P has 11 degrees of freedom (12 parameters, but scale is arbitrary).
  - One 2D/3D correspondence gives us two linearly independent equations.
  - Homogeneous least squares (similar to homography est.)

⇒ 5 ½ correspondences needed for a minimal solution.

Camera Calibration: DLT Algorithm

$$\begin{bmatrix} 0^T & X_1^T & -y_1 X_1^T \\ X_1^T & 0^T & -x_1 X_1^T \\ \cdots & \cdots & \cdots \\ 0^T & X_n^T & -y_n X_n^T \\ X_n^T & 0^T & -x_n X_n^T \end{bmatrix} \begin{pmatrix} P_1 \\ P_2 \\ P_3 \end{pmatrix} = 0$$

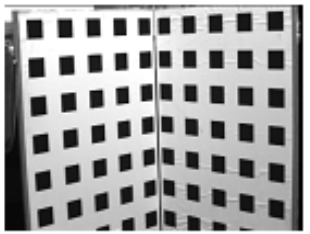


$$Ap = 0$$

Solve using... SVD!

- **Notes**
  - For coplanar points that satisfy $\Pi^T X = 0$, we will get degenerate solutions $(\Pi,0,0)$, $(0,\Pi,0)$, or $(0,0,\Pi)$.

$\Rightarrow$ We need calibration points in more than one plane!



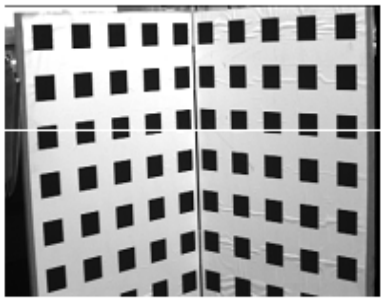
Camera Calibration

- Once we've recovered the numerical form of the camera matrix, we still have to figure out the **intrinsic** and **extrinsic parameters**

- This is a **matrix decomposition problem**, not an estimation problem (see F&P sec. 3.2, 3.3)

Camera Calibration: Some Practical Tips

- For best results, it is important that the calibration points are measured with *subpixel accuracy*.
- How this can be done depends on the exact pattern.
- Algorithm for checkerboard pattern
  1. Perform Canny edge detection.
  2. Fit straight lines to detected linked edges.
  3. Intersect lines to obtain corners.
  - If sufficient care is taken, the points can then be obtained with localization accuracy < 1/10 pixel.



- Rule of thumb
  - Number of constraints should exceed number of unknowns by a factor of five.

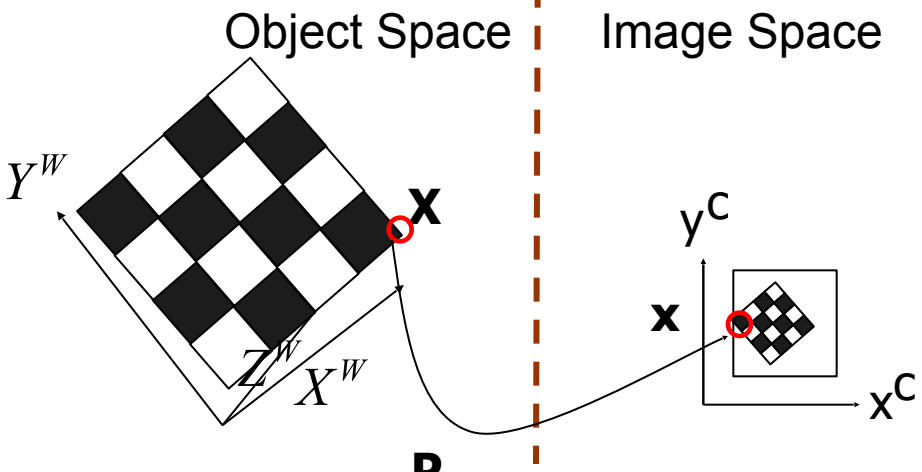
⇒ For 11 parameters of P, at least 28 points should be used.

Camera Calibration: Some Practical Tips

- For numerical reasons, it is important to carry out some data normalization.
  - Translate the image points x_i to the (image) origin and scale them such that their RMS distance to the origin is $\sqrt{2}$.
  - Translate the 3D points X_i to the (world) origin and scale them such that their RMS distance to the origin is $\sqrt{3}$.
  - (This is valid for compact point distributions on calibration objects).
- The DLT algorithm presented here is easy to implement, but there are some more accurate algorithms available (see H&Z sec. 7.2).
- For practical applications, it is also often needed to correct for radial distortion. Algorithms for this can be found in H&Z sec. 7.4, or F&P sec. 3.3.

Camera Calibration from Planar Patterns

- ICCV Zhang'99: "*Flexible Calibration by Viewing a Plane From Unknown Orientations*"



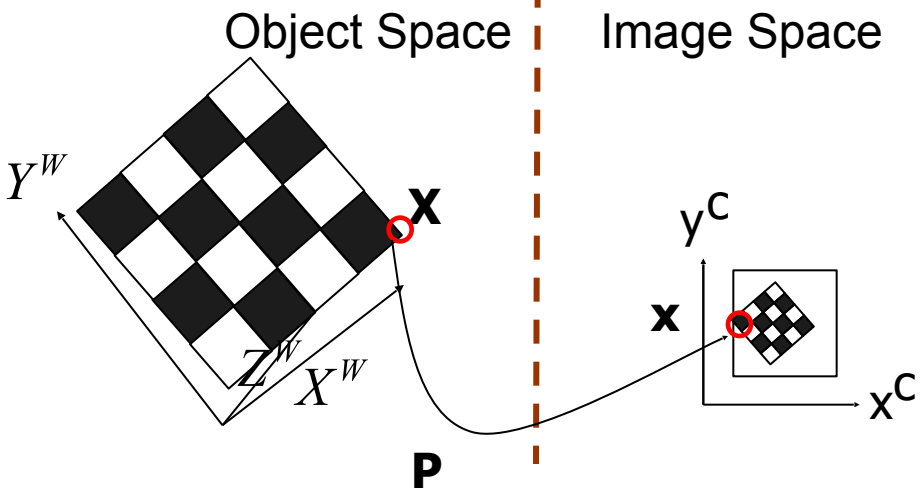
$$\mathbf{x} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \overbrace{K\begin{bmatrix} R & t \end{bmatrix}}^{\mathbf{P}} \mathbf{X}_w = K\begin{bmatrix} r_1 & r_2 & r_3 & t \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ 0 \\ 1 \end{bmatrix}$$

$$= K\begin{bmatrix} r_1 & r_2 & t \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ 1 \end{bmatrix} = H \begin{bmatrix} x_w \\ y_w \\ 1 \end{bmatrix}$$

H — 3×3 homography matrix

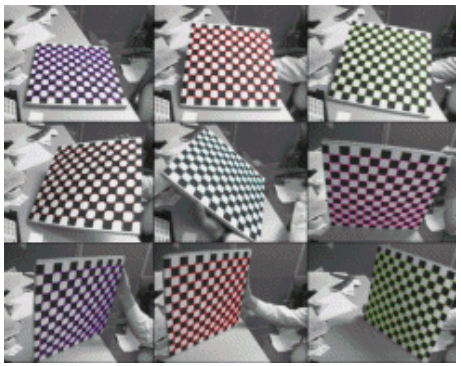
Camera Calibration from Planar Patterns

- ICCV Zhang'99: "*Flexible Calibration by Viewing a Plane From Unknown Orientations*"



➢Estimate the image homography matrix **H** for each image

$$\tilde{\mathbf{x}} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \lambda H \begin{bmatrix} x_w \\ y_w \\ 1 \end{bmatrix}$$



Camera Calibration from Planar Patterns

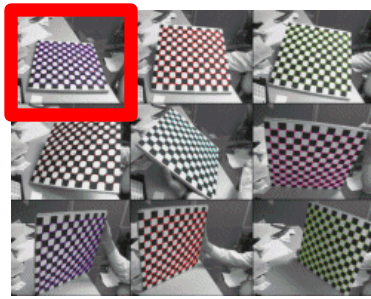
➢ Estimate K from homography matrix **H**

$$K\begin{bmatrix} r_1 & r_2 & t \end{bmatrix} = \frac{1}{\lambda} H$$

$$\Rightarrow H = \begin{bmatrix} h_1 & h_2 & h_3 \end{bmatrix} = \lambda K \begin{bmatrix} r_1 & r_2 & t \end{bmatrix}$$

$$\Rightarrow \begin{cases} r_1 = \frac{1}{\lambda} K^{-1} h_1 \\ r_2 = \frac{1}{\lambda} K^{-1} h_2 \end{cases}$$

$$r_1 \perp r_2 \Rightarrow h_1^T K^{-T} K^{-1} h_2 = 0$$



➢ Solve for **b** in the linear system (multiple H) : **V b = 0**

$$\mathbf{b} = [B_{11}, B_{12}, B_{22}, B_{13}, B_{23}, B_{33}]^T$$

B = K $^{-T}$ K $^{-1}$

b is the eigenvector of V^TV with smallest eigenvalue

b yields the intrinsic parameter matrix K

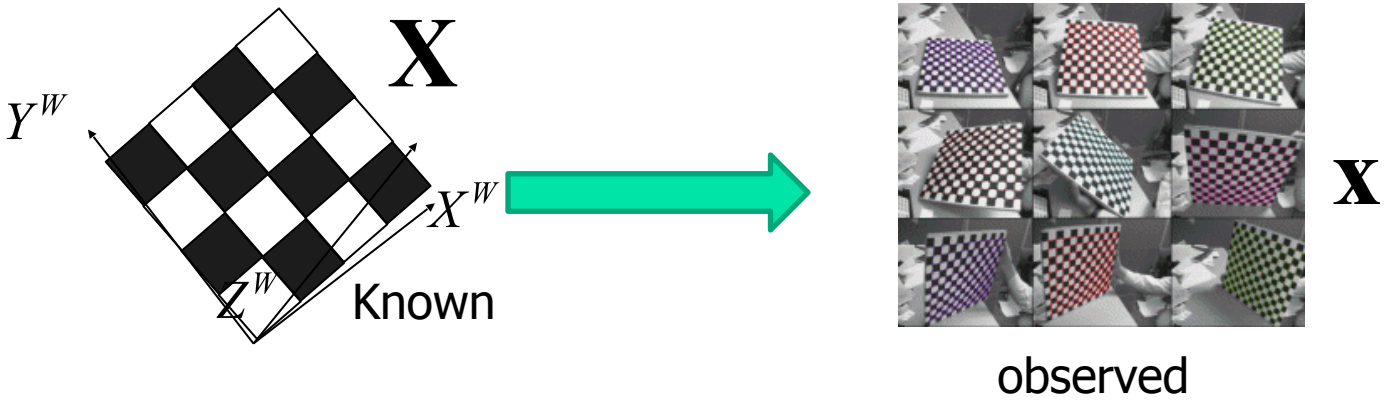
Camera Calibration from Planar Patterns

➢Compute rotation matrix **[r1 r2 r3]** and translation **t** given ***K*** **&** ***H***

$$K^{-1}H = \lambda\begin{bmatrix} r_1 & r_2 & t \end{bmatrix} \qquad H = \begin{bmatrix} h_1 & h_2 & h_3 \end{bmatrix}$$

$$\begin{cases} r_1 = \dfrac{1}{\lambda}K^{-1}h_1 \\ r_2 = \dfrac{1}{\lambda}K^{-1}h_2 \\ r_3 = r_1 \times r_2 \end{cases} \qquad \begin{cases} \mathbf{t} = \dfrac{1}{\lambda}K^{-1}h_2 \\ \lambda = \left\| K^{-1}h_1 \right\| \end{cases}$$

Camera Calibration from Planar Patterns



➢Minimize the objective function

(Nonlinear Least-Squares algorithm:

✓Steepest Descent Method-SDM

✓Newton Method-NM （*2nd-order approximation*）

✓Levenberg-Marquardt-LM（*SDM+GNM*）

$$J(K,R,t,\lambda)=\sum_{i=1}^{n}\sum_{j=1}^{m}\left\|\tilde{\mathbf{x}}_{ij}-f\left(K,R_i,t_i,\lambda_i,\mathbf{X}_j\right)\right\|_2^2$$

Camera Calibration from Planar Patterns

- **Finding an initial solution**
  - First step
    - Estimate the image homography matrix **H** for each image
  - Second step
    - Solve for **b** in the linear system: $\mathbf{V}\,\mathbf{b} = \mathbf{0}$
    - **b** yields the intrinsic parameter matrix **K**.

      Rotation matrix **[r1 r2 r3]** and translation **t**:
    - But the computed rotation matrix does not satisfy the properties of rotation matrix: $\mathbf{R}^T\mathbf{R}=\mathbf{R}\mathbf{R}^T=\mathbf{I}$.

      One can it enforce by: $\min||\mathbf{R}_{new} - \mathbf{R}||$,

      $$[\mathbf{U}\ \mathbf{D}\ \mathbf{V}] = \mathrm{SVD}(\mathbf{R}),$$

      $$\mathbf{R}_{new} = \mathbf{U}\mathbf{V}^T$$
- **Minimize the objective function using the initial solution**

  (use "*lsqnonlin*" in Matlab)

Camera Model & Computation of Camera Matrix

Multiple View Geometry in Computer Vision, Chapter 6~7, pp.153-194

6

Camera Models

A camera is a mapping between the 3D world (object space) and a 2D image. The principal camera of interest in this book is *central projection*. This chapter develops a number of camera *models* which are matrices with particular properties that represent the camera mapping.

It will be seen that all cameras modelling central projection are specializations of the *general projective camera*. The anatomy of this most general camera model is examined using the tools of projective geometry. It will be seen that geometric entities of the camera, such as the projection centre and image plane, can be computed quite simply from its matrix representation. Specializations of the general projective camera inherit its properties, for example their geometry is computed using the same algebraic expressions.

The specialized models fall into two major classes – those that model cameras with a finite centre, and those that model cameras with centre "at infinity". Of the cameras at infinity the *affine camera* is of particular importance because it is the natural generalization of parallel projection.

This chapter is principally concerned with the projection of points. The action of a camera on other geometric entities, such as lines, is deferred until chapter 8.

6.1 Finite cameras

In this section we start with the most specialized and simplest camera model, which is the basic pinhole camera, and then progressively generalize this model through a series of gradations.

The models we develop are principally designed for CCD like sensors, but are also applicable to other cameras, for example X-ray images, scanned photographic negatives, scanned photographs from enlarged negatives, etc.

The basic pinhole model. We consider the central projection of points in space onto a plane. Let the centre of projection be the origin of a Euclidean coordinate system, and consider the plane $Z = f$, which is called the *image plane* or *focal plane*. Under the pinhole camera model, a point in space with coordinates $\mathbf{X} = (X, Y, Z)^\top$ is mapped to the point on the image plane where a line joining the point $\mathbf{X}$ to the centre of projection meets the image plane. This is shown in figure 6.1. By similar triangles, one quickly

153

7

Computation of the Camera Matrix P

This chapter describes numerical methods for estimating the camera projection matrix from corresponding 3-space and image entities. This computation of the camera matrix is known as *resectioning*. The simplest such correspondence is that between a 3D point $\mathbf{X}$ and its image $\mathbf{x}$ under the unknown camera mapping. Given sufficiently many correspondences $\mathbf{X}_i \leftrightarrow \mathbf{x}_i$ the camera matrix P may be determined. Similarly, P may be determined from sufficiently many corresponding world and image lines.

If additional constraints apply to the matrix P, such as that the pixels are square, then a *restricted* camera matrix subject to these constraints may be estimated from world to image correspondences.

Throughout this book it is assumed that the map from 3-space to the image is linear. This assumption is invalid if there is lens distortion. The topic of radial lens distortion correction is dealt with in this chapter.

The internal parameters K of the camera may be extracted from the matrix P by the decomposition of section 6.2.4. Alternatively, the internal parameters can be computed directly, without necessitating estimating P, by the methods of chapter 8.

7.1 Basic equations

We assume a number of point correspondences $\mathbf{X}_i \leftrightarrow \mathbf{x}_i$ between 3D points $\mathbf{X}_i$ and 2D image points $\mathbf{x}_i$ are given. We are required to find a camera matrix P, namely a 3×4 matrix such that $\mathbf{x}_i = \mathrm{P}\mathbf{X}_i$ for all i. The similarity of this problem with that of computing a 2D projective transformation H, treated in chapter 4, is evident. The only difference is the dimension of the problem. In the 2D case the matrix H has dimension 3×3, whereas in the present case, P is a 3×4 matrix. As one may expect, much of the material from chapter 4 applies almost unchanged to the present case.

As in section 4.1(*p88*) for each correspondence $\mathbf{X}_i \leftrightarrow \mathbf{x}_i$ we derive a relationship

$$\begin{bmatrix} \mathbf{0}^\top & -w_i\mathbf{X}_i^\top & y_i\mathbf{X}_i^\top \\ w_i\mathbf{X}_i^\top & \mathbf{0}^\top & -x_i\mathbf{X}_i^\top \\ -y_i\mathbf{X}_i^\top & x_i\mathbf{X}_i^\top & \mathbf{0}^\top \end{bmatrix} \begin{pmatrix} \mathbf{P}^1 \\ \mathbf{P}^2 \\ \mathbf{P}^3 \end{pmatrix} = \mathbf{0}. \qquad (7.1)$$

where each $\mathbf{P}^{i\top}$ is a 4-vector, the i-th row of P. Alternatively, one may choose to use

178

References

[1] Forsyth, and Ponce, **Computer Vision-A Modern Approach**, (英文影印版），清华大学出版社，2004, ISBN:7-302-07795-9。(Chapter 1-3)

[2] R. Hartley, A. Zisserman, **Multiple View Geometry in Computer Vision**, 2nd Ed., Cambridge Univ. Press, 2004. **Chapters 6~7**.

[4] Zhenyou Zhang , **Flexible camera calibration by viewing a plane from unknown orientations**, ICCV99.

[5] Open Source (Code): **OpenCV**(Intel)

http://www.vision.caltech.edu/bouguetj/calib_doc/