Geometric Models & Camera Calibration

Reading: Chap. 2 & 3

Introduction

- We have seen that a camera captures both **geometric** and **photometric** information of a 3D scene.
  - Geometric: shape, e.g. lines, angles, curves
  - Photometric: color, intensity
- What is the geometric and photometric relationship between a 3D scene and its 2D image?
- We will understand these in terms of **models**.

Models

- Models are **approximations** of reality.
- “Reality” is often too complex, or computationally intractable, to handle.
- Examples:
  - Newton’s Laws of Motion vs. Einstein’s Theory of Relativity
  - Light: waves or particles?
- No model is perfect.
  - Need to understand its strengths & limitations.

Camera Projection Models

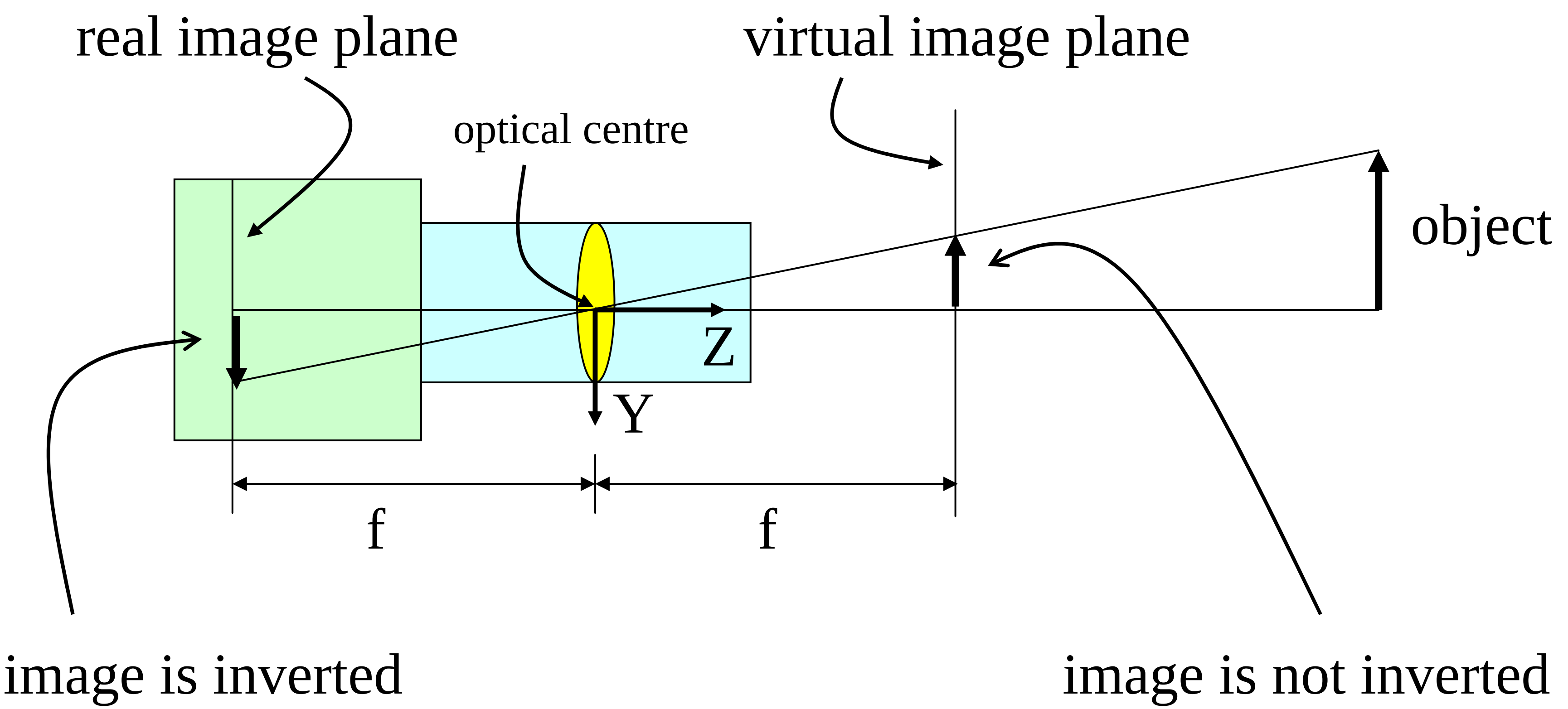
- 3D scenes project to 2D images
- Most common model: **pinhole camera model**

- From this we may derive several types of projections:
  - orthographic
  - weak-perspective
  - para-perspective
  - perspective

least accurate

most accurate

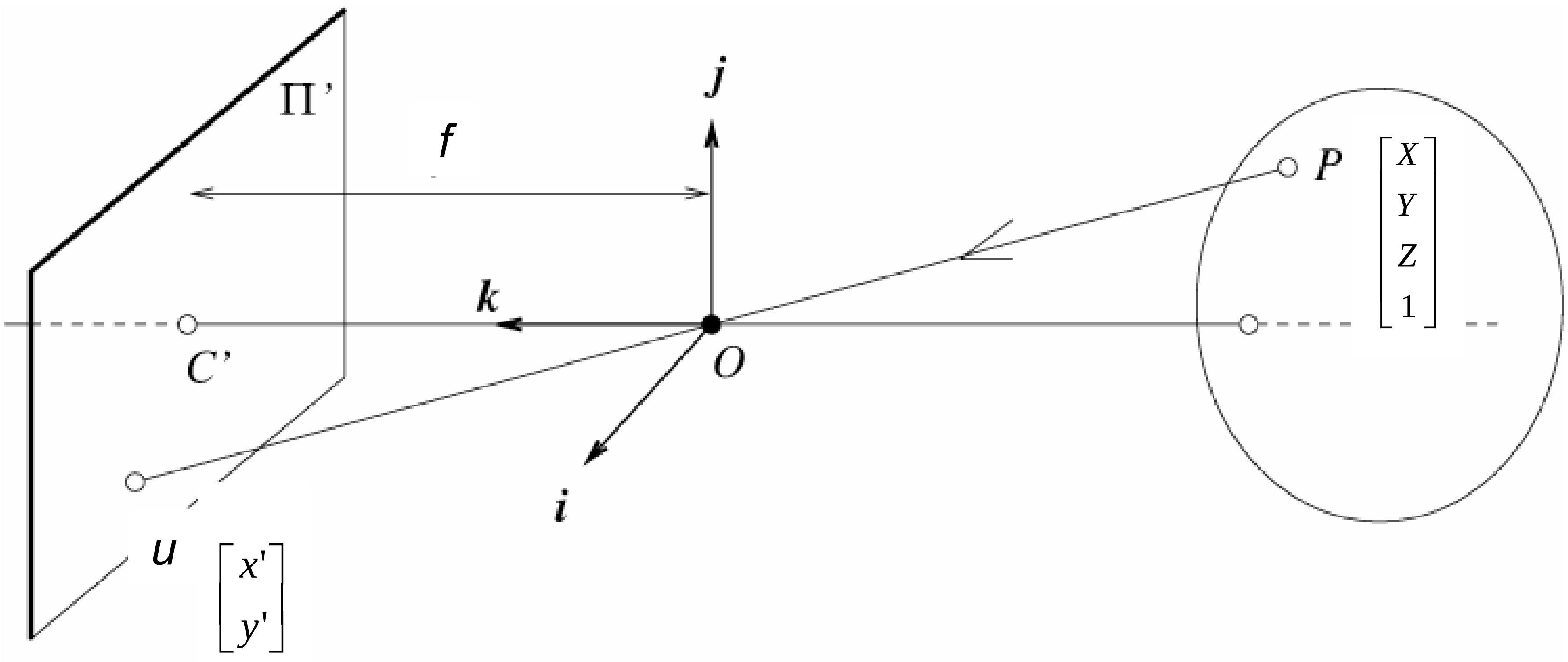
Pinhole Camera Model



Real vs. Virtual Image Plane

- The image is physically formed on the real image plane (**retina**).
- The image is vertically and laterally inverted.
- We can imagine a virtual image plane at a distance of ***f*** in front of the camera optical center, where ***f*** is the focal length.
- The image on this virtual image plane is not inverted.
  - This is actually more convenient.
- Henceforth, when we say “image plane” we will mean the virtual image plane.

Perspective Projection



Perspective Projection

- The imaging process is a many-to-one mapping
  - all points on the 3D ray map to a single image point.

- Therefore, depth information is lost

- From similar triangles, can write down the perspective equation:

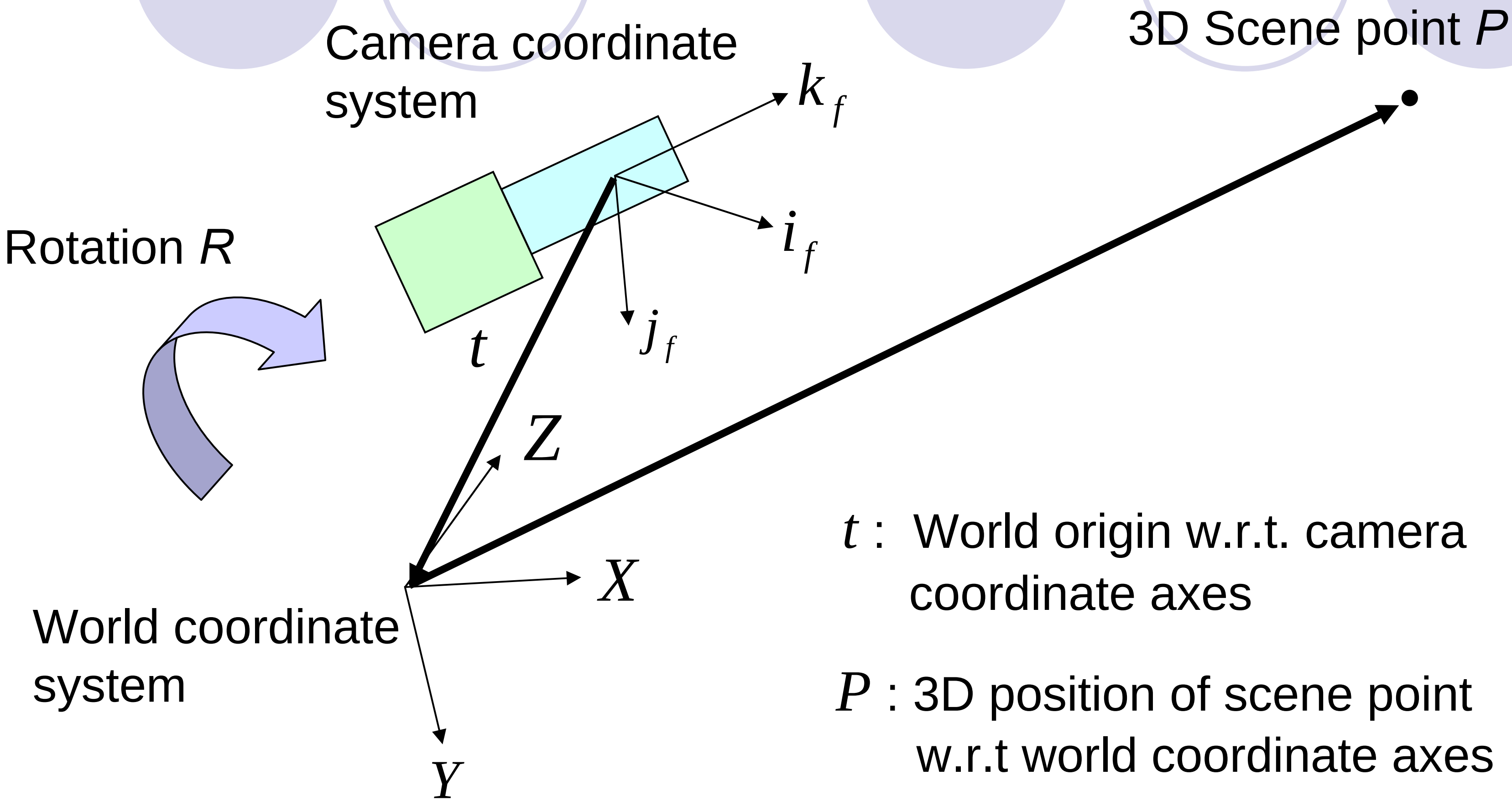
$$\frac{x'}{f} = \frac{X}{Z}, \quad \frac{y'}{f} = \frac{Y}{Z},$$

Perspective Projection

$$\begin{bmatrix} x' \\ y' \end{bmatrix} \Leftrightarrow \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

or $\quad u = MP$

This assumes coordinate axes is at the pinhole.



t : World origin w.r.t. camera coordinate axes

P : 3D position of scene point w.r.t world coordinate axes

R : Rotation matrix to align world coordinate axes to camera axes

Perspective Projection

$$u = MP$$

$$M: \quad 3 \times 4 \quad \text{projection matrix}$$

$$= K[R \quad t]$$

$$K: \quad 3 \times 3 \quad \text{intrinsic parameter matrix}$$

$$= \begin{bmatrix} \alpha & -\alpha \cot\theta & u_0 \\ 0 & \beta / \sin\theta & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

α, β : scaling in image u, v axes, respectively

Θ : skew angle, that is, angle between u, v axes

u_0, v_0 : origin offset

Note: *α=kf*, *β=lf* where *k, l* are the magnification factors

Intrinsic vs. Extrinsic parameters

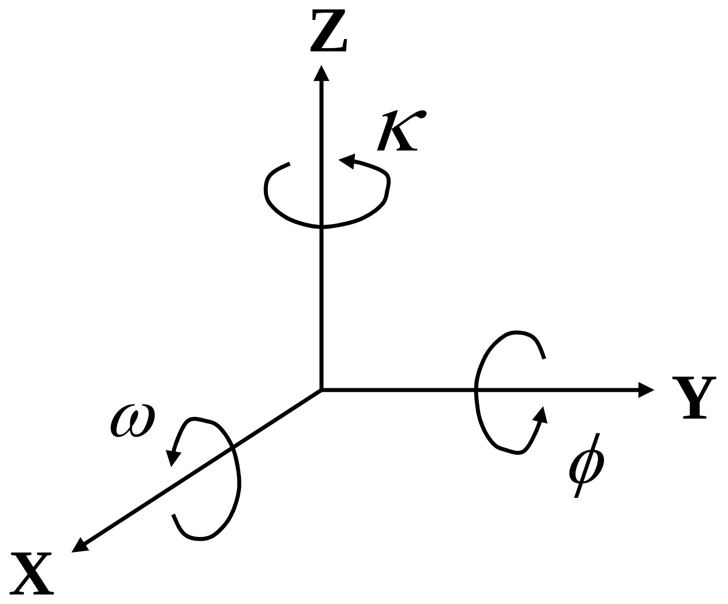
- Intrinsic: internal camera parameters.
  - 6 of them (5 if you don't care about focal length)
  - Focal length, horiz. & vert. magnification, horiz. & vert. offset, skew angle

- Extrinsic: external parameters
  - 6 of them
  - 3 rotation angles, 3 translation param

- Imposing assumptions will reduce # params.
- Estimating params is called **camera calibration**.

Representing Rotations

- Euler angles
  - pitch: rotation about x axis :
  - yaw: rotation about y axis:
  - roll: rotation about z axis:



Rotation Matrix

$$R = R_z R_y R_x$$

$$= \begin{pmatrix} \cos(\kappa)\cos\phi & \cos(\kappa)\sin(\phi)\sin(\omega) - \sin(\kappa)\cos(\omega) & \cos(\kappa)\sin(\phi)\cos(\omega) + \sin(\kappa)\sin(\omega) \\ \sin(\kappa)\cos(\phi) & \sin(\kappa)\sin(\phi)\sin(\omega) + \cos(\kappa)\cos(\omega) & \sin(\kappa)\sin(\phi)\cos(\omega) - \cos(\kappa)\sin(\omega) \\ -\sin(\phi) & \cos(\phi)\sin(\omega) & \cos(\phi)\cos(\omega) \end{pmatrix}$$

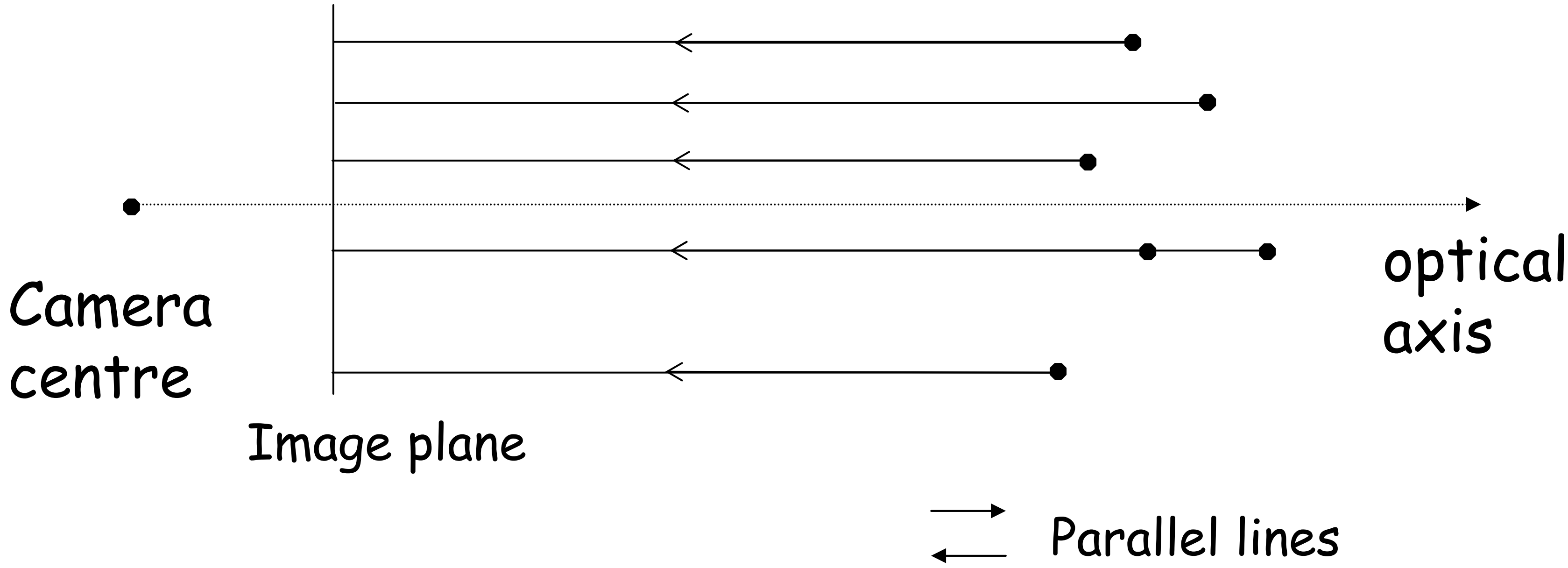
Two properties of rotation matrix:

R is orthogonal: $R^T R = I$

det(R) = 1

Orthographic Projection

Projection rays are parallel



Orthographic Projection Equations

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix}$$

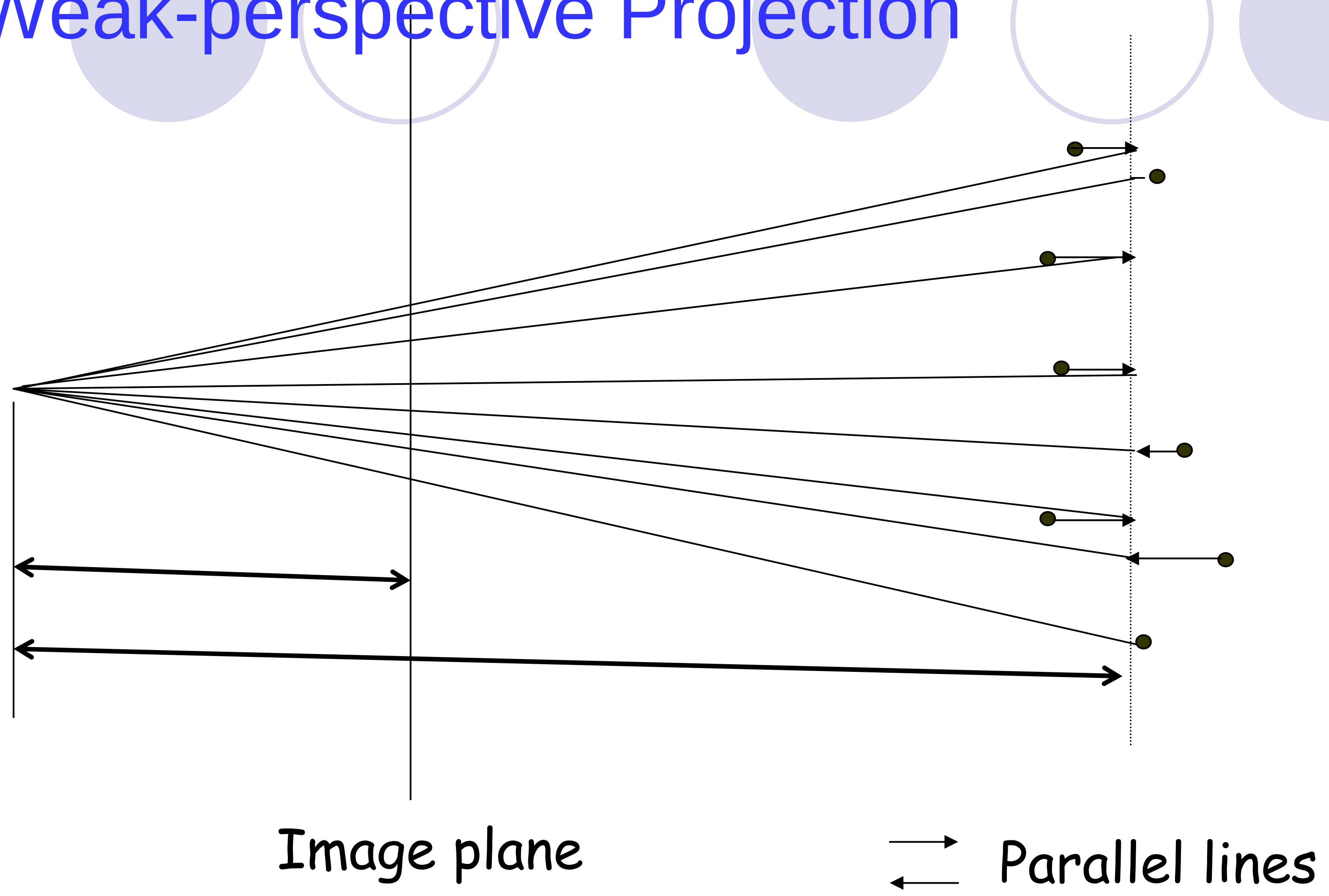
$$= \begin{bmatrix} r_1^T & t_x \\ r_2^T & t_y \\ 0^T & 1 \end{bmatrix}$$

r_1^T, r_2^T : first two rows of *R*

t_x, t_y : first two elements of *t*

This projection has 5 degrees of freedom.

Weak-perspective Projection

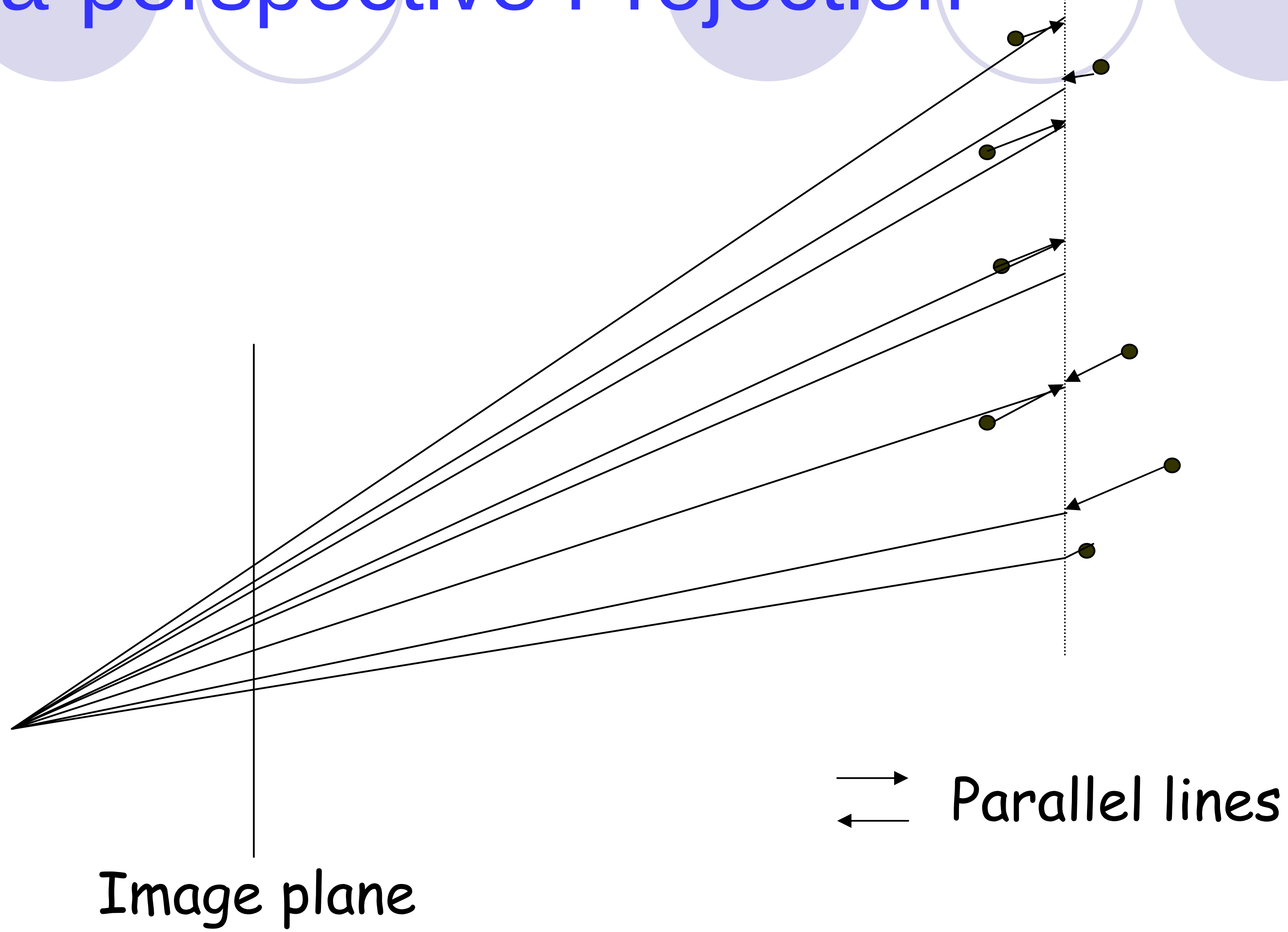


Weak-perspective Projection

$$M = \begin{bmatrix} \alpha & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_1^T & t_x \\ r_2^T & t_y \\ 0^T & 1 \end{bmatrix}$$

This projection has 7 degrees of freedom.

Para-perspective Projection



Para-perspective Projection

- Camera matrix given in textbook Table 2.1, page 36.
- Note: Written a bit differently.
- This has 9 d.o.f.

Real cameras

- Real cameras use lenses
- Lens distortion causes distortion in image
  - Lines may project into curves
  - See Chap 1.2, 3.3 for details
- Change of focal length (zooming) scales the image
  - not true if assuming orthographic projection)
- Color distortions too

Color Aberration

- Bad White Balance

Color Aberration

- Purple Fringing

Color Aberration

- Vignetting

Geometric Camera Calibration

Recap

- A general projective matrix:

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

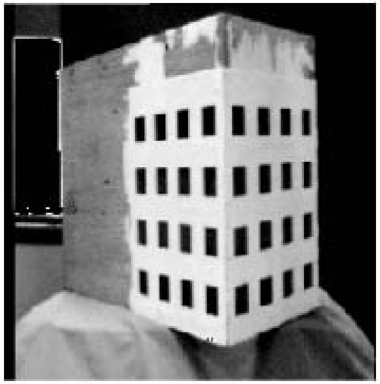
- This has 11 d.o.f. , i.e. 11 parameters
  - 5 intrinsic, 6 extrinsic
- How to estimate all of these?
  - Geometric camera calibration

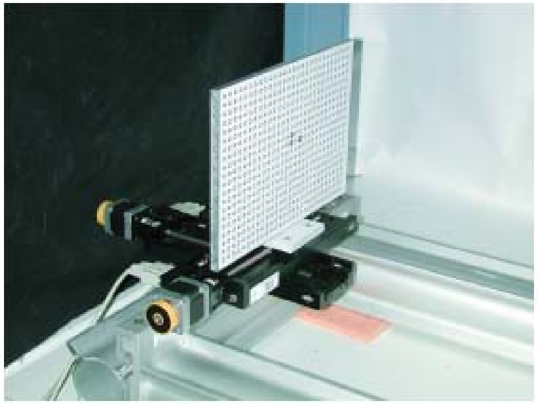
Key Idea

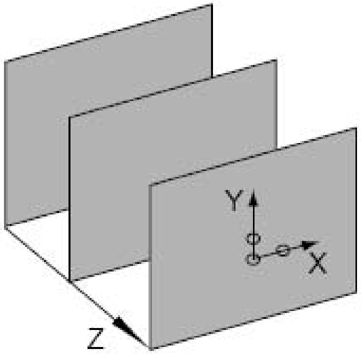
- Capture image of a known 3D object (calibration object).
- Establish correspondences between 3D points and their 2D image projections.
- Estimate *M*
- Estimate *K*, *R*, and *t* from *M*
- Estimation can be done using linear or non-linear methods.
  - We will study linear methods first.

Setting things up

- Calibration object:
  - 3D object, or
  - 2D planar object captured at different locations







Setting it up

- Suppose we have N point correspondences:
  - $P_1, P_2, \ldots P_N$ are 3D scene points
  - $u_1, u_2, \ldots u_N$ are corresponding image points

- Let $m^T{}_1$, $m^T{}_{2,}$ $m^T{}_3$ be the 3 rows of M
- Let (u_i, v_i) be the (non-homogeneous) coords of the i th image point.
- Let P_i be the homogeneous coords of i th scene point.

Math

- For the *i* th point

$$(m_1 - u_i m_3)^T P_i = 0$$

$$(m_2 - v_i m_3)^T P_i = 0$$

- Each point gives 2 equations.
- Using all *N* points gives 2*N* equations:

Math

$$\begin{bmatrix} P_1^T & 0^T & -u_1 P_1^T \\ 0^T & P_1^T & -v_1 P_1^T \\ \cdots & \cdots & \cdots \\ P_N^T & 0^T & -u_N P_N^T \\ 0^T & P_N^T & -v_N P_N^T \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \end{bmatrix} = 0$$

$$\mathbf{A} \qquad \mathbf{m} = 0$$

A : 2*N* x 12 matrix, rank is 11
m : 12 x 1 column vector
Note that **m** has only 11 d.o.f.

Math

- Solution?
- **m** is in the Nullspace of **A** !

- Use SVD: $\mathbf{A} = \mathbf{U} \Sigma \mathbf{V}^T$

- **m** = last column of **V**
  - **m** is only up to an unknown scale
  - In this case, SVD solution makes $|| \mathbf{m} || = 1$

Estimating *K*, *R*, *t*

- Now that we have *M* (3x4 matrix)

$$M = K[R \quad t] = [B \quad p]$$
$$t = -Rc$$

- *c* is the 3D coords of camera center wrt World axes
- Let $\hat{c}$ be homogeneous coords of c
- It can be shown that $M\hat{c} = 0$
  - So $\hat{c}$ is in the Nullspace of *M*
- And *t* is computed from $-Rc$, once *R* is known.

Estimating *K*, *R*, *t*

- *B*, the left 3x3 submatrix of *M* is *KR*
- Perform "RQ" decomposition on *B*.
- The "Q" is the required rotation matrix *R*
- And the upper triangular "R" is our *K* matrix !
  - Impose the condition that diagonals of *K* are positive

- We are done!

RQ factorization

- Any n x n matrix A can be factored into A = RQ
  - Where both R, Q are n x n
  - R is upper triangular, Q is orthogonal
- Not the same as QR factorization
- Trick: post multiply A by Givens rotation matrices: Q_x ,Q_y, Q_z

$$Q_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c & -s \\ 0 & s & c \end{bmatrix}, \quad Q_y = \begin{bmatrix} c & 0 & s \\ 0 & 1 & 0 \\ -s & 0 & c \end{bmatrix}, \quad Q_z = \begin{bmatrix} c & -s & 0 \\ s & c & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\text{where} \quad c = \cos\theta, \quad s = \sin\theta$$

RQ factorization

- *c, s* chosen to make a particular entry of A zero
- For example, to make $a_{21} = 0$, we solve

$$ca_{21} + sa_{22} = 0$$

$$\Rightarrow \quad c = \frac{-a_{22}}{\sqrt{a_{21}^2 + a_{22}^2}}, \quad s = \frac{a_{21}}{\sqrt{a_{21}^2 + a_{22}^2}}$$

- Choose Q_x ,Q_y, Q_z such that

$$AQ_xQ_yQ_z = R \quad \text{upper triangular}$$

$$\Rightarrow A = RQ_z^TQ_y^TQ_x^T = RQ$$

Degeneracy

- Caution: The 3D scene points should not all lie on a plane.
  - Otherwise no solution
- In practice, choose N >= 6 points in "general position"

- Note: the above assumes no lens distortion.
  - See Chap 3.3 for how to deal with lens distortion.

Summary

- Presented pinhole camera model
  - From this we get hierarchy of projection types
  - Perspective, Para-perspective, Weak-perspective, Orthographic
- Showed how to calibrate camera to estimate intrinsic + extrinsic parameters.