55:148 Digital Image Processing

Chapter 11

3D Vision, Geometry

Topics:

- Basics of projective geometry
  - Points and hyperplanes in projective space
  - Homography
  - Estimating homography from point correspondence
- The single perspective camera
  - An overview of single camera calibration
  - Calibration of one camera from the known scene
- Two cameras, stereopsis
  - The geometry of two cameras. The fundamental matrix
  - Relative motion of the camera; the essential matrix
  - Estimation of a fundamental matrix from image point correspondences
  - Applications of the epipolar geometry in vision
- Three and more cameras
  - Stereo correspondence algorithms

Green = To be discussed today

Kronecker product

In mathematics, the **Kronecker product**, denoted by "⊗", is an operation on two matrices of arbitrary size resulting in a block matrix. The Kronecker product should **NOT** be confused with the usual matrix multiplication, which is an entirely different operation.

If A is an m-by-n matrix and B is a p-by-q matrix, then the Kronecker product $A \otimes B$ is the mp-by-nq block matrix

Example

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \otimes \begin{bmatrix} 0 & 5 \\ 6 & 7 \end{bmatrix} = \begin{bmatrix} 1\cdot 0 & 1\cdot 5 & 2\cdot 0 & 2\cdot 5 \\ 1\cdot 6 & 1\cdot 7 & 2\cdot 6 & 2\cdot 7 \\ 3\cdot 0 & 3\cdot 5 & 4\cdot 0 & 4\cdot 5 \\ 3\cdot 6 & 3\cdot 7 & 4\cdot 6 & 4\cdot 7 \end{bmatrix} = \begin{bmatrix} 0 & 5 & 0 & 10 \\ 6 & 7 & 12 & 14 \\ 0 & 15 & 0 & 20 \\ 18 & 21 & 24 & 28 \end{bmatrix}$$

We will use the following identity

$$AB\mathbf{c} = \left(\mathbf{c}^{\mathrm{T}} \otimes A\right)\mathbf{b} \;\mid\; b = [b_{11} \quad \cdots \quad b_{p1}\, b_{12} \quad \cdots \quad b_{qp}]^{\mathrm{T}}$$

Linear estimation

Why linear estimation?

Cost function for maximum likelihood estimation

$$\min_{\mathbf{h},u_i,v_i} \sum_{i=1}^{m} \left[(u_i - \hat{u}_i)^2 + (v_i - \hat{v}_i)^2 + \left(\frac{h_{11}u_i + h_{12}v_i + h_{13}}{h_{31}u_i + h_{32}v_i + h_{33}} - \hat{u}_i' \right)^2 + \left(\frac{h_{21}u_i + h_{22}v_i + h_{23}}{h_{31}u_i + h_{32}v_i + h_{33}} - \hat{v}_i' \right)^2 \right]$$

Maximum likelihood estimation: non convex cost function → multiple optimal solutions (local optimal problem) → a need for initial estimation of the matrix H

Linear estimation approach is used to solve the initialization problem

Linear estimation

Principle: minimize the sum of **algebraic distance** between the estimated (transformed from first projection) and actual points in the second position

Advantage: canonical solution, i.e., no optimization search algorithm is needed → unique optimum

Method:

- Start with the equation $\alpha\mathbf{u}' = H\mathbf{u}$
- Eliminate the term α by multiplying with a matrix $S(\mathbf{u}')$ whose rows are orthogonal to $\mathbf{u}'$ and $\det S(\mathbf{u}') \neq 0$; example: $S(\mathbf{u}') = S([u', v', w']) = \begin{bmatrix} 0 & -w' & v' \\ w' & 0 & -u' \\ -v' & u' & 0 \end{bmatrix}$
- Rearrange the equation $S(\mathbf{u}')H\mathbf{u} = \mathbf{0}$ using the identity of Kronecker product identity

$$S(\mathbf{u}')H\mathbf{u} = [\mathbf{u}^{\mathrm{T}} \otimes S(\mathbf{u}')]\mathbf{h} = \mathbf{0}$$

where $\mathbf{h} = [h_{11}, h_{21}, \ldots, m_{23}, m_{33}]$

Linear estimation Method

- Combine m correspondences into a single matrix form

$$\left(\begin{bmatrix} \mathbf{u}_1^{\mathrm{T}} \otimes S(\mathbf{u}_1') \\ \mathbf{u}_2^{\mathrm{T}} \otimes S(\mathbf{u}_2') \\ \vdots \\ \mathbf{u}_m^{\mathrm{T}} \otimes S(\mathbf{u}_m') \end{bmatrix} = W\right) \mathbf{h} = \mathbf{0}$$

- In real application this equality does not hold good (why??); so, it boils down to a minimization of the term on the l.h.s., i.e., $\|W\mathbf{h}\|$ subject to $\|\mathbf{h}\| = 1$
- Soln: Compute $W^T W = 9 \times 9$ square matrix
  - Compute eigenvectors and eigenvalues of $W^T W$
  - Find $\mathbf{h}$ along the eigenvector associated with the smallest eigenvalue

- Data preconditioning:
  - Use H_{pre} and H'_{pre} to pre the data points such mean and std of points on each project are 0 and 1, respectively

- Finally, $H = H_{\mathrm{pre}}\ \overline{H}\ H'^{-1}_{\mathrm{pre}}$

11.3 A single projection camera

11.3.1 Camera model

- Different coordinate system
- Transformation between every two coordinate systems

Homogeneous coordinates

- Image points: $[u, v, w]^{\mathrm{T}}$
- 3D scene points: $[X, Y, Z, W]^{\mathrm{T}}$

In a homogeneous coordinate, the last element w or W is used to indicate the 'α' parameter of Eq. 11.1. Thus a homogeneous coordinate represents a ray.
All points with $w = 1$ lie on the image plane.
Thus, by performing all transformations on homogeneous coordinates, we do not only transform a set of points onto another set of points; rather, a set of rays is transformed onto another set of rays.

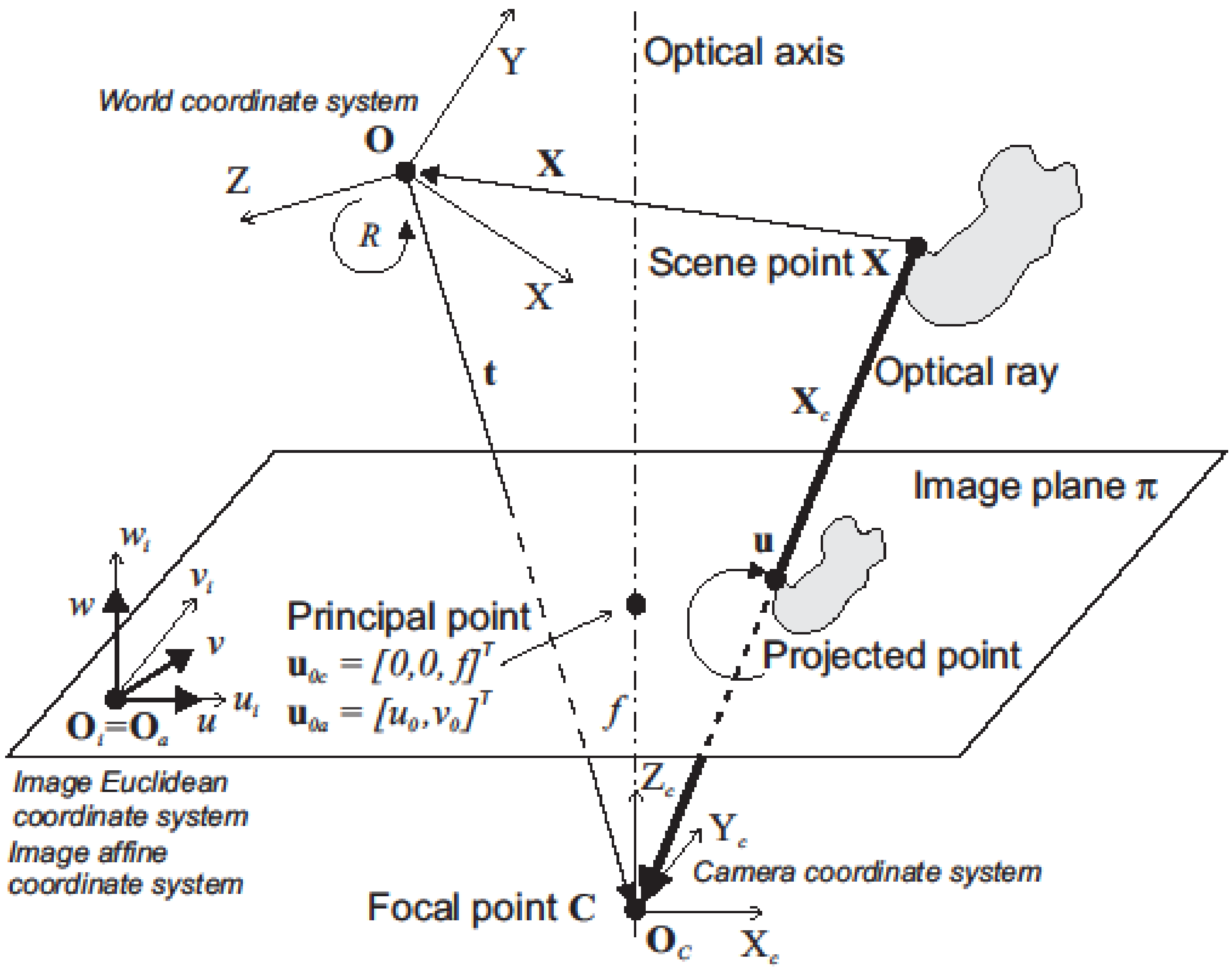
Camera model

Important parameters

- Image plane
- Optical axis
- Focal point/optical center/center of projection

Different coordinate systems

- World/scene coordinate system
- Camera coordinate system
- Image Euclidean coordinate system
- Image affine coordinate system



Transformations between different coordinate systems

Scene → Camera

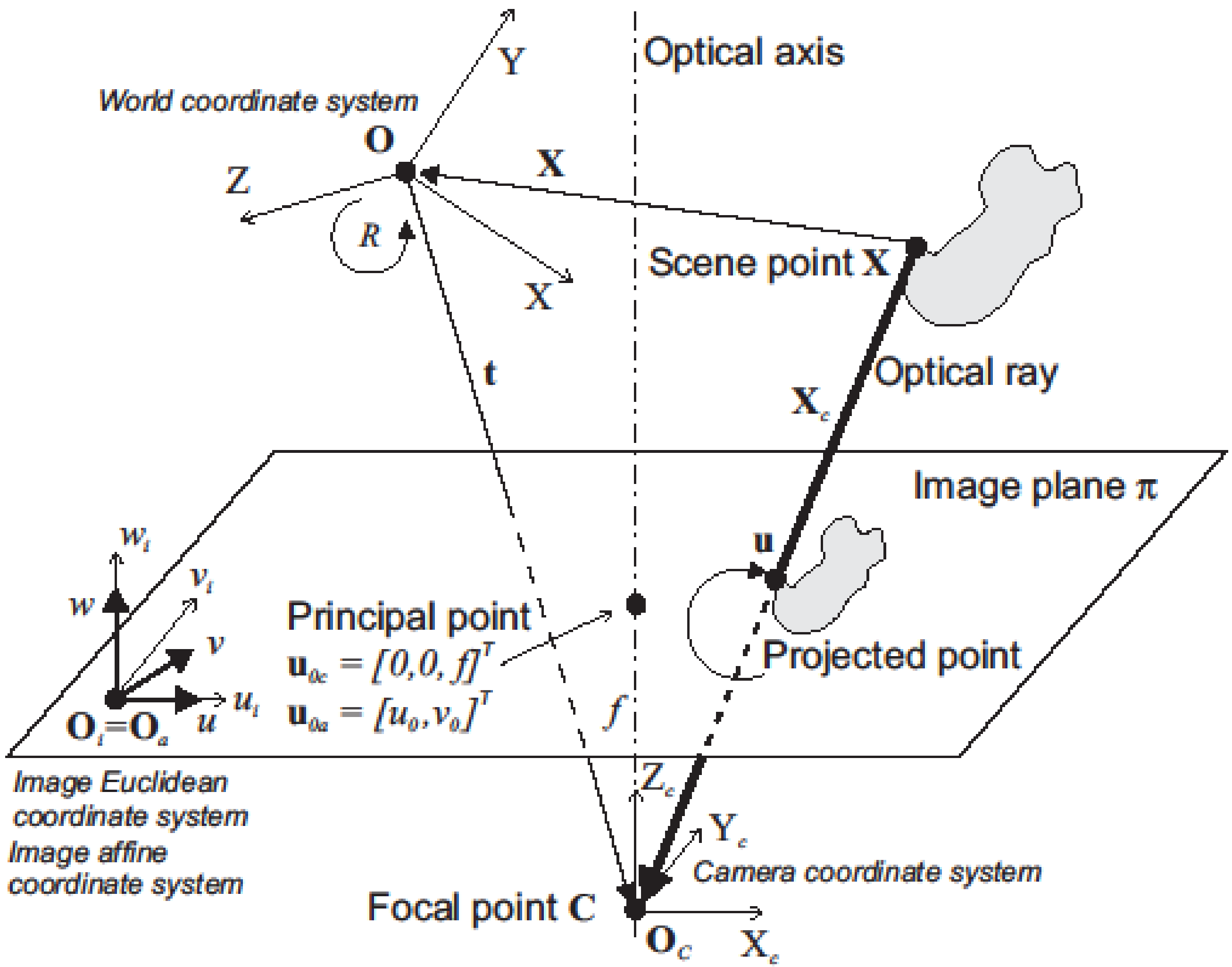
Non-homogeneous coordinates

$$\mathbf{X}_c = R(\mathbf{X} - \mathbf{t})$$

Homogeneous coordinates

$$\mathbf{X}_c = \begin{bmatrix} R & -R\mathbf{t} \\ \mathbf{0}^\mathrm{T} & 1 \end{bmatrix} \mathbf{X}$$

R and $\mathbf{t}$ are called **extrinsic camera calibration parameters**



Camera→ Image Euclidean

Non-homogeneous coordinates

$$u_i = \frac{X_c f}{Z_c}, \qquad v_i = \frac{Y_c f}{Z_c}$$

Homogeneous coordinates

$$\mathbf{u}_i \cong \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{X}_c$$

Explain the relation between the above two equations

Normalized image plane ($f = 1$)

$$\mathbf{u}_i \cong \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{X}_c$$

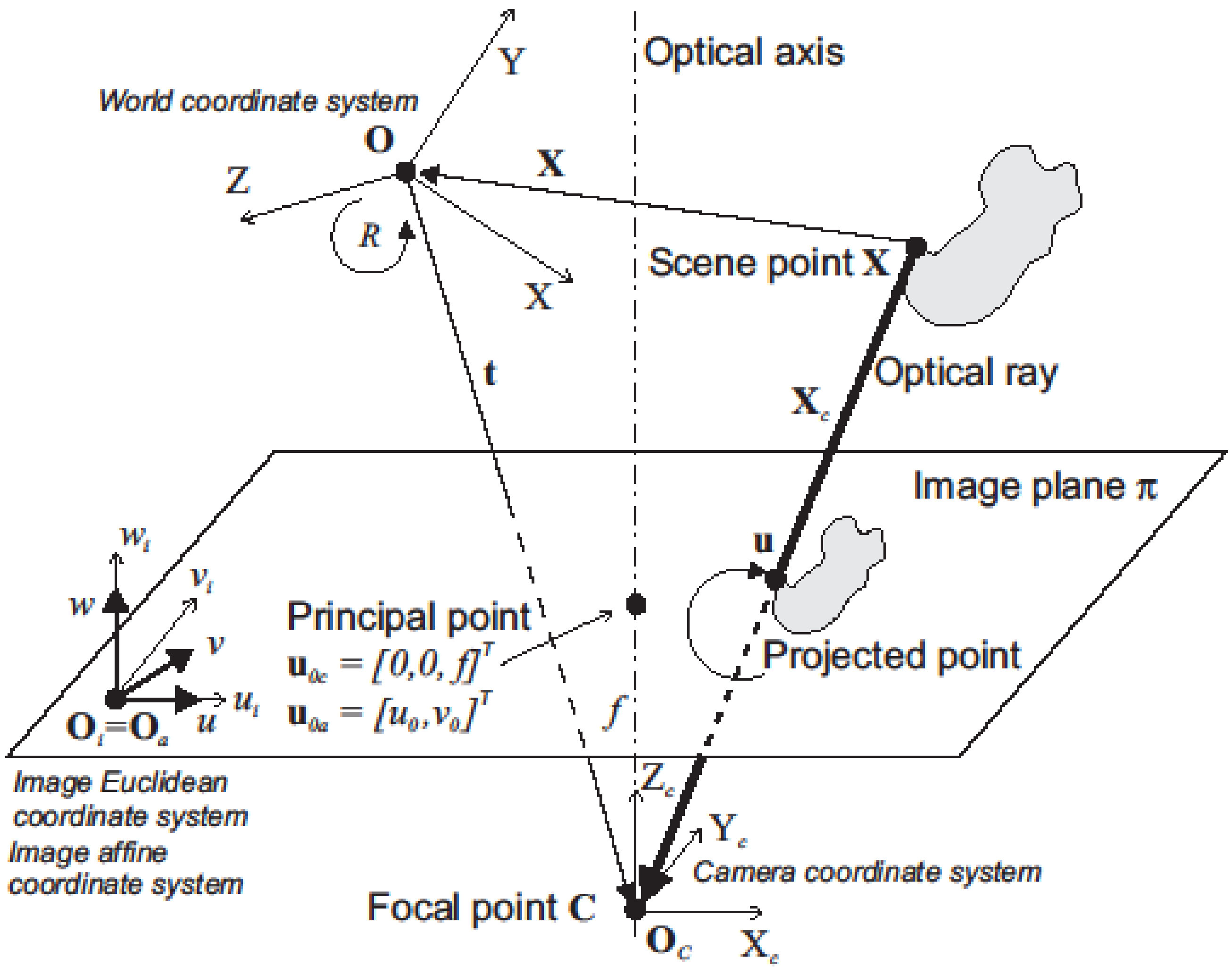


Image Euclidean → Image affine

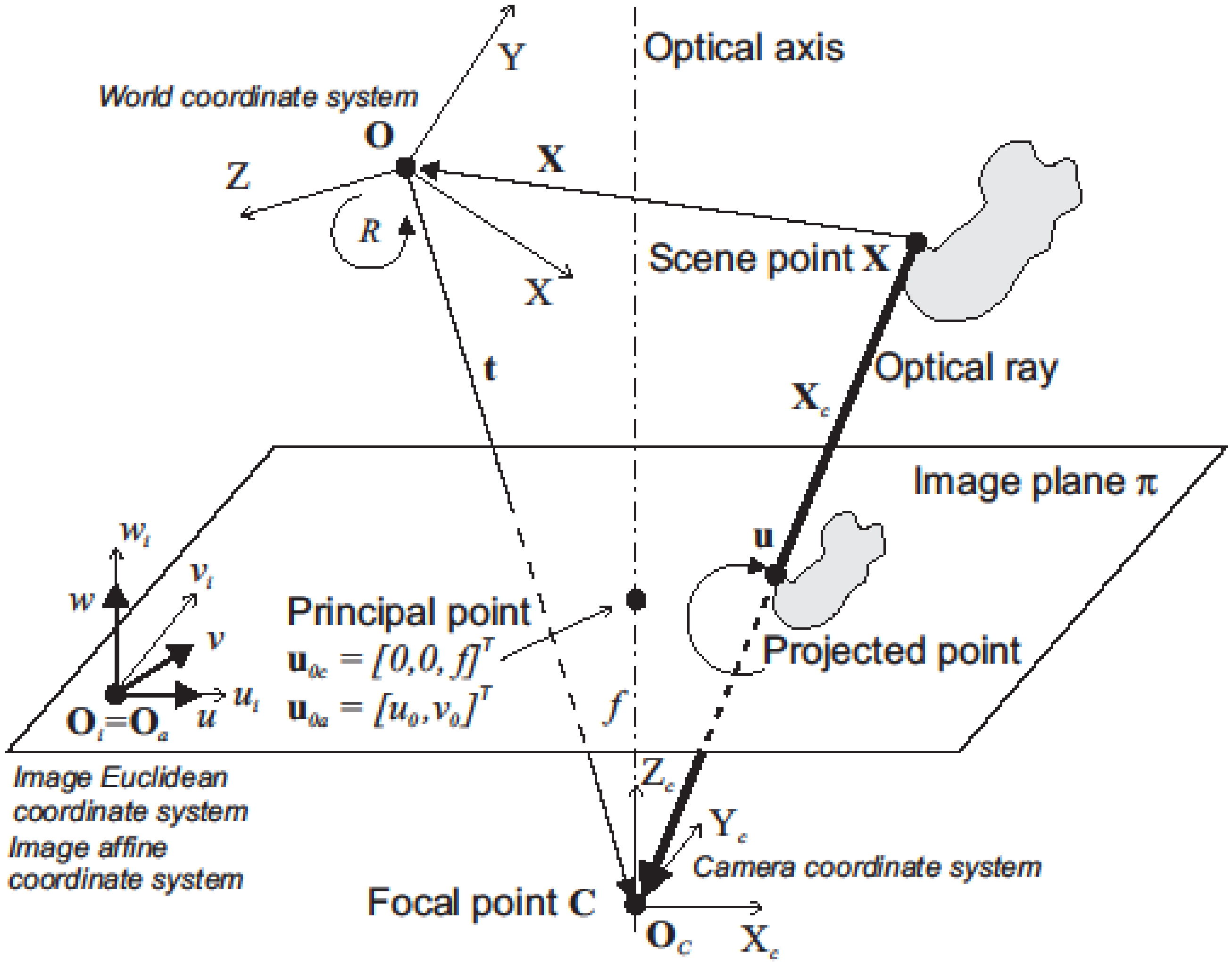
Homogeneous coordinates

$$\mathbf{u} \cong K\mathbf{u}_i$$
$$= \begin{bmatrix} k & s & -u_0 \\ 0 & g & -v_0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{u}_i$$

Explain different parameters in the matrix K

K is called the **intrinsic calibration matrix**

Note that $\mathbf{u}$ represents the image pixel coordinate system

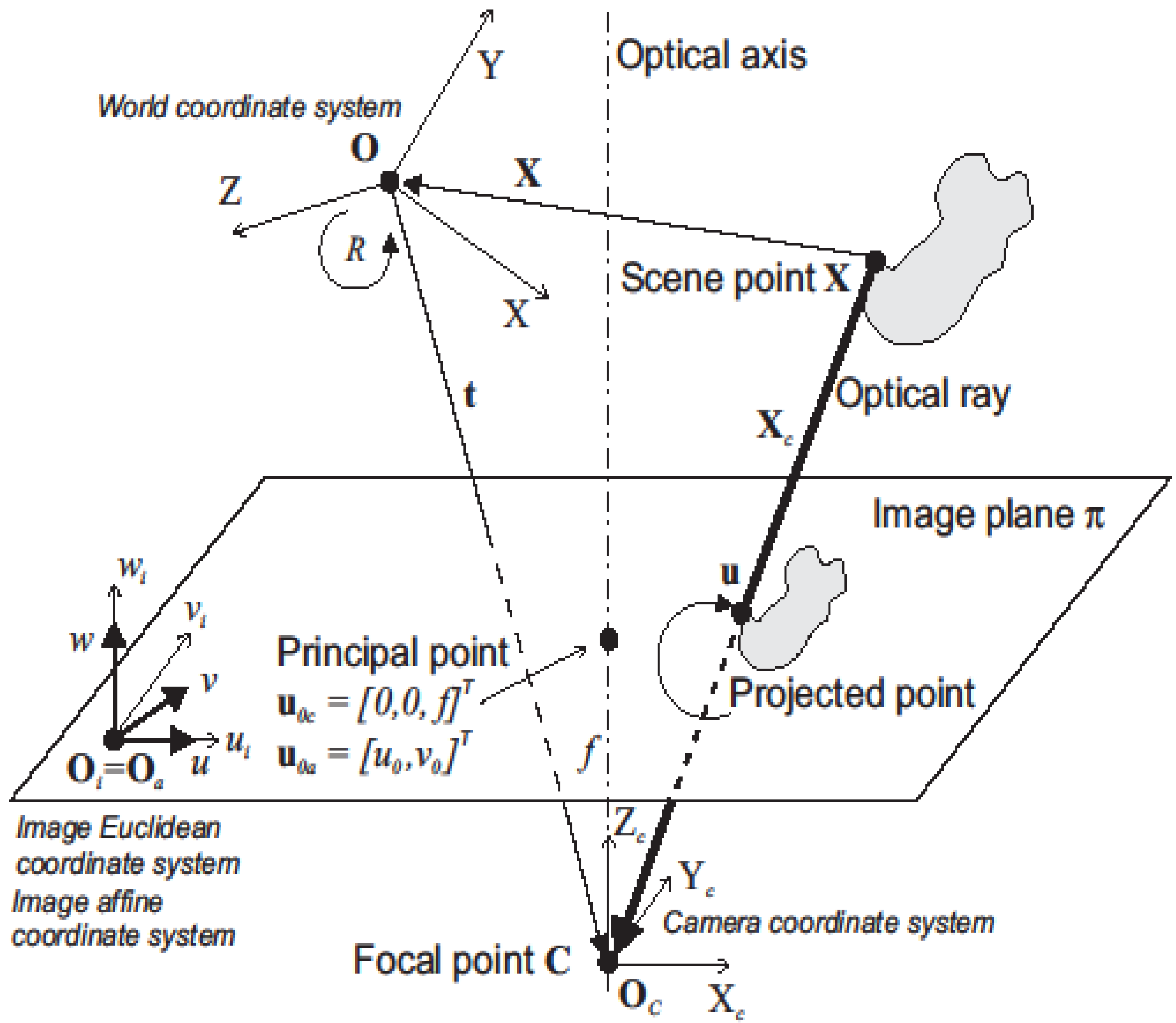


Scene → Image affine

Homogeneous coordinates

$$\mathbf{u} \cong K\mathbf{u}_i$$
$$= K \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{X}_c$$
$$= K \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & -R\mathbf{t} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix} \mathbf{X}$$
$$= M\mathbf{X}$$

M $(3 \times 4$ matrix) is called the **projection matrix**



Projection and back projection

Here, homogeneous coordinates are used

$$\mathbf{u} \cong M\mathbf{X}$$

To compute the inverse map we need to compute the inverse of M
But M is not a square matrix!
How to solve it?
Use **pseudo inverse** $M^{+} = M^{\mathrm{T}}\left(MM^{\mathrm{T}}\right)^{-1}$; Property $MM^{+} = I$

Image → Scene transformation

$$\mathbf{X} = M^{+}\mathbf{u}$$

- Does it mean that we can uniquely map a 2D image point to a 3D scene point?

Projection and back projection

Here, homogeneous coordinates are used

$$\mathbf{u} \cong M\mathbf{X}$$

To compute the inverse map we need to compute the inverse of M
But M is not a square matrix!
How to solve it?
Use **pseudo inverse** $M^{+} = M^{\mathrm{T}}\left(MM^{\mathrm{T}}\right)^{-1}$; Property $MM^{+} = I$

Image → Scene transformation

$$\mathbf{X} = M^{+}\mathbf{u}$$

- Does it mean that we can uniquely map a 2D image point to a 3D scene point?
  NO!
- Note that we are using homogeneous coordinate which represent a ray and not a point
- So the above equation maps a ray from the image affine coordinate system to scene coordinate system
- Here is the beauty of homogenous coordinate system; <u>it does not solve the ambiguity but provide a compact mathematical representation in the presence of ambiguity</u>

Back projection of an image line

Given: an image line $\mathbf{l}$

Note that, in homogeneous coordinates, a image line $\mathbf{l}$ represents a plane, say $\mathbf{a}$

A scene point $\mathbf{X}$ on the plane $\mathbf{a}$ satisfies $\mathbf{a}^{\mathrm{T}}\mathbf{X} = 0$

Now, the projection of $\mathbf{X}$, say $\mathbf{u} = M\mathbf{X}$, must lie on the line $\mathbf{l}$

Thus,

$$\mathbf{l}^{\mathrm{T}}\mathbf{u} = \mathbf{l}^{\mathrm{T}}M\mathbf{X} = 0 \Rightarrow \ \mathbf{a} = \left(\mathbf{l}^{\mathrm{T}}M\right)^{\mathrm{T}} = M^{\mathrm{T}}\mathbf{l}$$

The plane contains the optical center, i.e, $\mathbf{a}^{\mathrm{T}}\mathbf{C} = 0$.

11.3.3 Camera calibration from a known scene

We want to derive the projection matrix M that consists of both extrinsic camera calibration parameters (what are those?) and the intrinsic calibration matrix (again, what is that?)

Given: a set of image-scene point correspondences $\{(\mathbf{u}_i, \mathbf{X}_i)\}_{i=1}^{m}$

Output: the projection matrix M

It sounds like we have solved a similar problem before!!

11.3.3 Camera calibration from a known scene

We want to derive the projection matrix M that consists of both extrinsic camera calibration parameters (what are those?) and the intrinsic calibration matrix (again, what is that?)

Given: a set of image-scene point correspondences $\{(\mathbf{u}_i, \mathbf{X}_i)\}_{i=1}^{m}$

Output: the projection matrix M

It sounds like we have solved a similar problem before!!
Remember the process of homography estimation

What is the comparison?

11.3.3 Camera calibration from a known scene

We want to derive the projection matrix M that consists of both extrinsic camera calibration parameters (what are those?) and the intrinsic calibration matrix (again, what is that?)

Given: a set of image-scene point correspondences $\{(\mathbf{u}_i, \mathbf{X}_i)\}_{i=1}^{m}$

Output: the projection matrix M

It sounds like we have solved a similar problem before!!
Remember the process of homography estimation

What is the comparison?
In homography estimation we solved a 3×3 matrix H using a $\mathcal{P}^2 \to \mathcal{P}^2$ correspondence
Here we have to solve a 3×4 matrix M using a $\mathcal{P}^2 \to \Re^3$ correspondence

Remember that we are using homogeneous coordinates

Steps?

11.3.3 Camera calibration from a known scene

We want to derive the projection matrix M that consists of both extrinsic camera calibration parameters (what are those?) and the intrinsic calibration matrix (again, what is that?)

Given: a set of image-scene point correspondences $\{(\mathbf{u}_i, \mathbf{X}_i)\}_{i=1}^{m}$

Output: the projection matrix M

It sounds like we have solved a similar problem before!!
Remember the process of homography estimation

What is the comparison?
In homography estimation we solved a 3×3 matrix H using a $\mathcal{P}^2 \rightarrow \mathcal{P}^2$ correspondence
Here we have to solve a 3×4 matrix M using a $\mathcal{P}^2 \rightarrow \Re^3$ correspondence

Remember that we are using homogeneous coordinates

Steps?
Initialization using linear estimation
Optimization using maximum likelihood estimation

11.3.3 Camera calibration from a known scene

We want to derive the projection matrix M that consists of both extrinsic camera calibration parameters (what are those?) and the intrinsic calibration matrix (again, what is that?)

Given: a set of image-scene point correspondences $\{(\mathbf{u}_i, \mathbf{X}_i)\}_{i=1}^{m}$

Output: the projection matrix M

It sounds like we have solved a similar problem before!!
Remember the process of homography estimation

What is the comparison?
In homography estimation we solved a 3×3 matrix H using a $\mathcal{P}^2 \to \mathcal{P}^2$ correspondence
Here we have to solve a 3×4 matrix M using a $\mathcal{P}^2 \to \Re^3$ correspondence

Remember that we are using homogeneous coordinates

Steps?
Step 1 Initialization using linear estimation
Step 2 Optimization using maximum likelihood estimation

Step 1 Initialization using linear estimation

We want to derive the projection matrix M that consists of both extrinsic camera calibration parameters (what are those?) and the intrinsic calibration matrix (again, what is that?)

- Multiply the equation $\mathbf{u} = M\mathbf{X}$ by $S(\mathbf{u})$ to get $\mathbf{0} = S(\mathbf{u})M\mathbf{X}$ (Remember how $S(\mathbf{u})$ is formed?)
- Rearrange the expression $S(\mathbf{u})M\mathbf{X}$ to get $[\mathbf{X}^{\mathrm{T}} \otimes S(\mathbf{u})]\mathbf{m} = \mathbf{0}$ where $\mathbf{m} = [m_{11}, m_{21}, \dots, m_{24}, m_{34}]$
- Integrate all m correspondences into a single system of linear equation

$$\left(\begin{bmatrix} \mathbf{X}_1^{\mathrm{T}} \otimes S(\mathbf{u}_1) \\ \mathbf{X}_2^{\mathrm{T}} \otimes S(\mathbf{u}_2) \\ \vdots \\ \mathbf{X}_m^{\mathrm{T}} \otimes S(\mathbf{u}_m) \end{bmatrix} = W \right) \mathbf{m} = W\mathbf{m} = \mathbf{0}$$

- Compute eigenvectors and eigenvalues of $W^T W$
- Find $\mathbf{m}$ along the eigenvector associated with the smallest eigenvalue subject to $\|\mathbf{m}\| = 1$

Step 2 Maximum Likelihood Estimation

Cost function :

$$\min_{\mathbf{m},X_i,Y_i,Z_i} \sum_{i=1}^{m} \left[\left(X_i - \hat{X}_i\right)^2 + \left(Y_i - \hat{Y}_i\right)^2 + \left(Z_i - \hat{Z}_i\right)^2 + (Proj_u(\mathbf{m},\mathbf{X}) \ - \hat{u}_i)^2 + (Proj_v(\mathbf{m},\mathbf{X}) \ - \hat{v}_i)^2 \right]$$