

3D Vision-2021

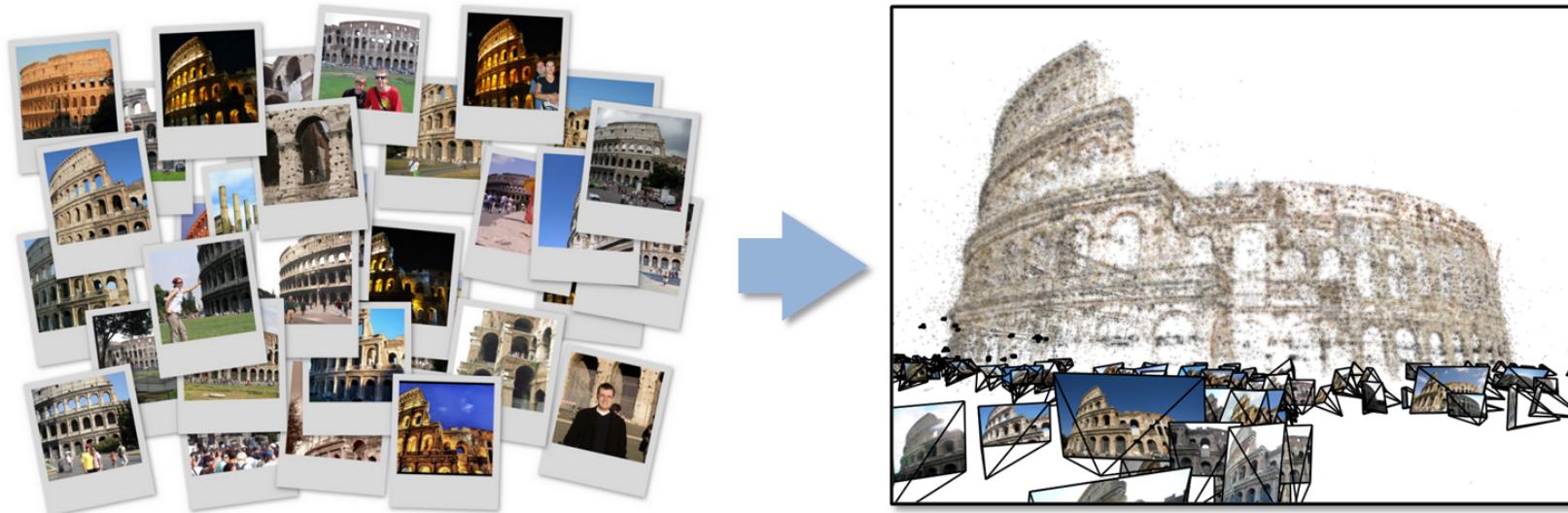
# 3D Computer Vision

A.K.A.

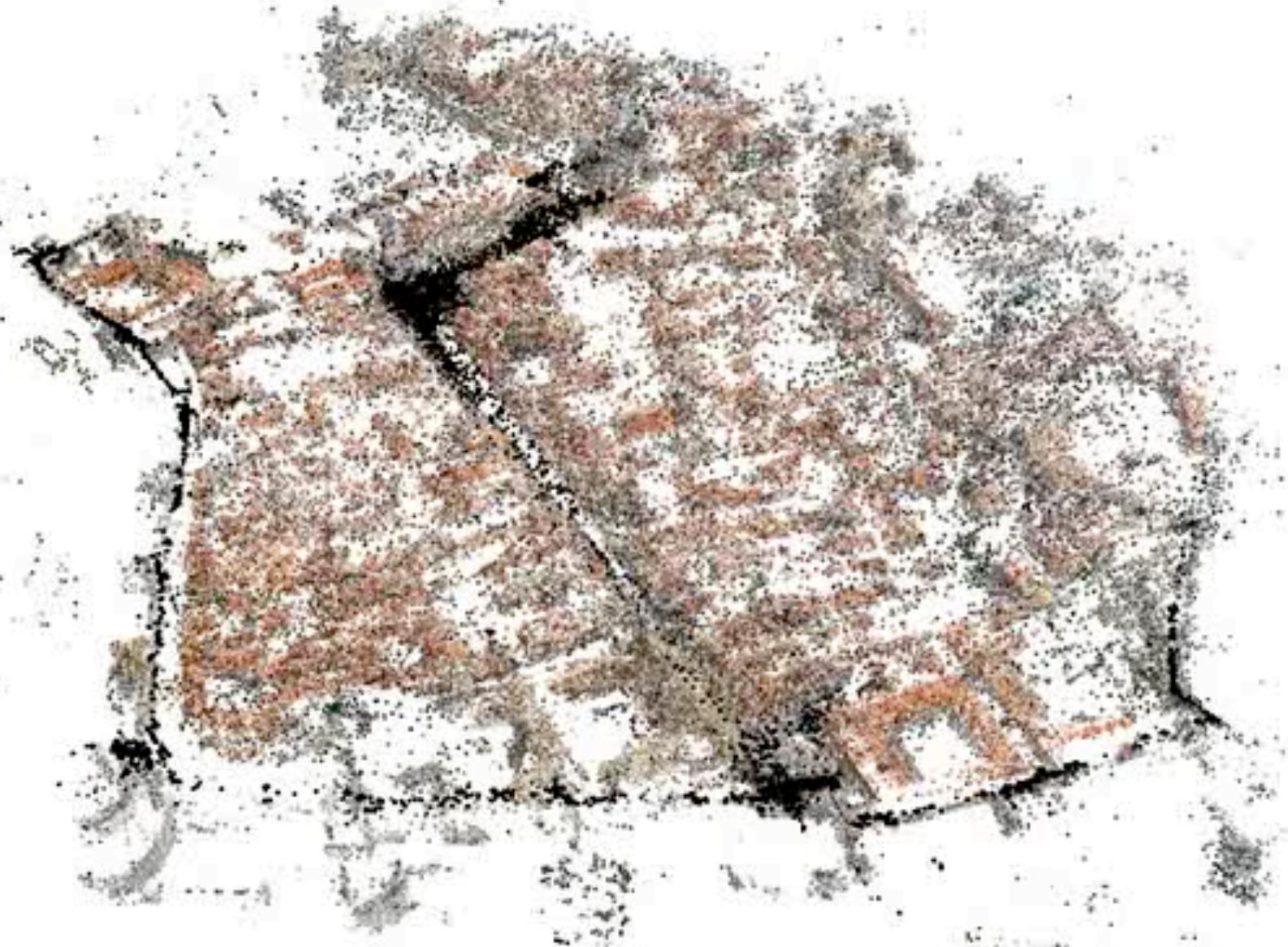
- 3D Reconstruction/3D Visual Modelling
- Multiple View Geometry
- Structure-from-Motion, or structure-and-motion
- Robot Visual-SLAM (partly Visual Odometry)
- Triangulation (in photogrammetry and geodesy communities)

# SfM from Internet Images

- Recent work has built 3D models from large, unstructured online image collections
  - [Snavely06], [Li08], [Agarwal09], [Frahm10], Microsoft's PhotoSynth, ...



- SfM is a key part of these reconstruction pipelines



# Cost Volume Pyramid Based Depth Inference for Multi-View Stereo

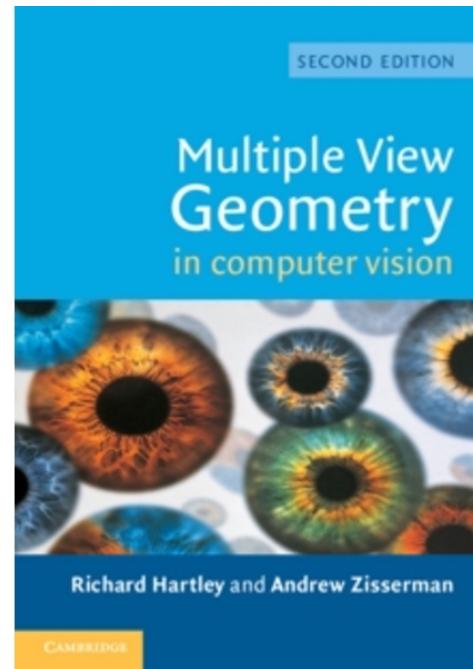
Jiayu Yang<sup>1</sup>, Wei Mao<sup>1</sup>, Jose M. Alvarez<sup>2</sup>, Miaomiao Liu<sup>1</sup>



Fast    Compact    Flexible    Accurate

# Reference for 3D Vision

- Book: Multiple view geometry in computer vision,  
Richard Hartley (ANU) and Andrew Zisserman  
(Oxford)



# Get the ebook

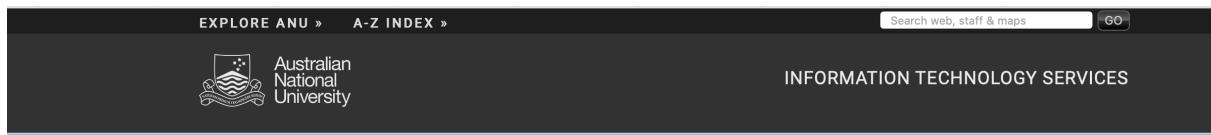
- Step 1: Go to

<https://login.virtual.anu.edu.au/menu>

Step 2: Login with your ANU student id and password

Step 3: You will be directed to

<https://anulib.anu.edu.au>



## ANU Reverse Proxy

The reverse proxy gives ANU students, staff and affiliates off-campus access to many online ANU services, websites, and electronic journals and databases subscribed to by the ANU Library.

Reverse Proxy Jump Point

URL \*

Press GO below to visit the library (<https://anulib.anu.edu.au>), or enter the URL of the website you want to visit above.

Please remember to include **http://** or **https://**

Help with Reverse Proxy

**Online**  
[IT Service Desk](#)  
Log a job online and check on your requests.

**Over the Phone**  
6125 4321 or ext. 54321  
(ask for Reverse Proxy support)  
8am-6pm Monday to Friday

# Get the ebook

- Step 4: Input “Multiple View Geometry in Computer Vision”

The screenshot shows the homepage of the Australian National University Library. At the top, there is a dark header with the university's crest, the text "Australian National University", and "LIBRARY". To the right is a search bar with the placeholder "Search ANU web, staff & maps" and a magnifying glass icon. Below the search bar is a link "My library record". A navigation menu below the header includes links for "Find & access", "Collections", "Research & learn", "Using the library", "News & events", "About", and "Services for".

The main content area features a box titled "IEEE Electronic Library". Inside the box, it says: "IEEE Electronic Library provides access to more than 3 million full text documents in electrical engineering, computer science, and electronics. Access IEEE e-courses for free during COVID-19." Below this text is a link "» start searching now".

At the bottom of the page, there are three tabs: "SuperSearch", "Catalogue search", and "Full text e-journals". The "SuperSearch" tab is currently active. Below these tabs is a search input field containing the text "Multiple View Geometry in Computer Vision". To the right of the input field is a blue "GO" button. At the very bottom of the page, there are links for "advanced SuperSearch »" and "about SuperSearch »".

# Get the ebook

- Step 5: Check the book online

ANU Library  
SuperSearch

Multiple View Geometry in Computer Vision

49,235 results sorted by relevance ▾  Add results beyond your library's collection

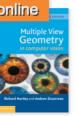
IN YOUR SEARCH

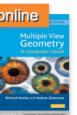
- Text Online
- Scholarly & Peer-Review
- Other-Review
- Open Access
- Library Catalog

ITEM TYPE ▾

- Journal Article (26,959)
- Dissertation/Thesis (12,179)
- Conference Proceeding (4,795)
- Collection (3,032)
- Book / eBook (366)
- e...

PUBLICATION DATE

1  **Multiple view geometry in computer vision**  
by Hartley, Richard; Zisserman, Andrew  
2003, 2nd ed.  
A basic problem in computer **vision** is to understand the structure of a real world scene given several images...  
Book: AVAILABLE, TA1634 .H38 2003, HANCOCK  
eBook: [Full Text Online](#)  
[Preview](#) ▾ Cited by

2  **Multiple View Geometry in Computer Vision**  
by Hartley, Richard; Zisserman, Andrew  
03/2004  
A basic problem in computer **vision** is to understand the structure of a real world scene given several images...  
eBook: [Full Text Online](#)  
[Preview](#) ▾

ProQuest  
Ebook Central™

Keyword, Author, ISBN, and more

Advanced Search Browse Subjects

**Multiple View Geometry in Computer Vision**  
Richard Hartley and Andrew Zisserman

Availability

Your institution has unlimited access to this book.

 [Read Online](#)  
 [Download Book](#)  
Get all pages, require free third-party software, Check out this book for up to 21 days.

 [Download PDF Chapter](#)  
Get up to 270 pages, use any PDF software, does not expire.

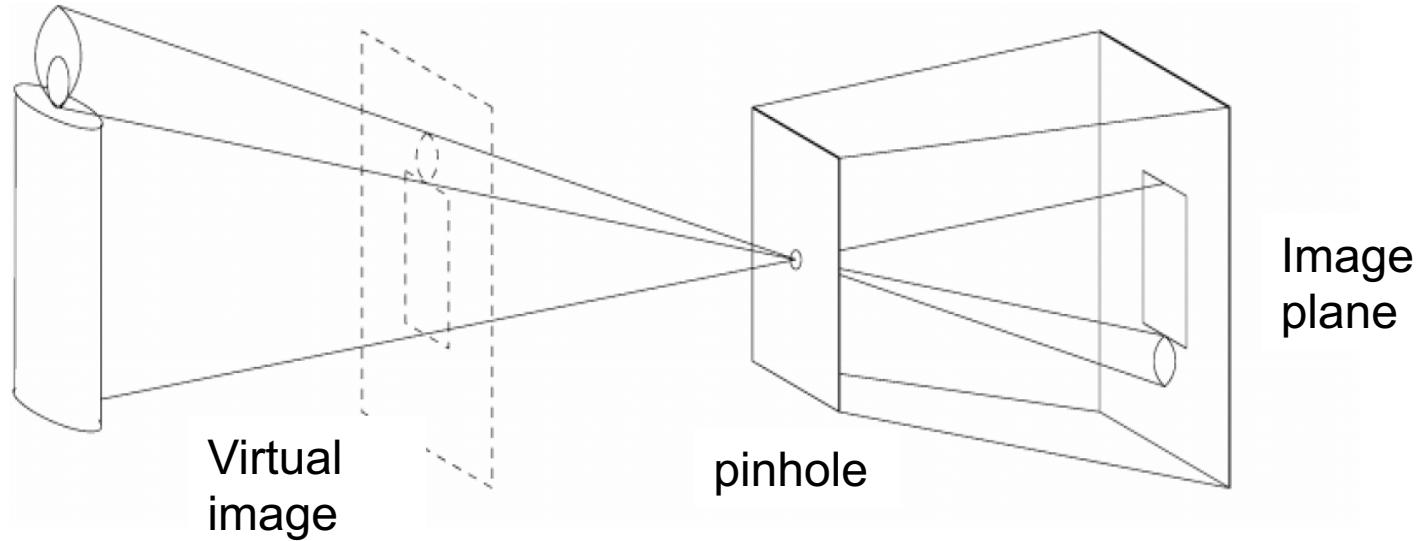
Description

A basic problem in computer vision is to understand the structure of a real world scene given several images of it. Techniques for solving this problem are taken from projective geometry and photogrammetry. Here, the authors cover the geometric principles and their algebraic representation in terms of camera projection matrices, the fundamental matrix and the trifocal tensor. The theory and methods of

Show more

# Review: Pinhole camera model

- Pinhole camera is an abstract model to approximate imaging process: **Perspective projection**.



If we treat pinhole as a point, only one ray from any given point can enter the camera.

# Homogeneous Coordinates

Converting inhomogeneous to *homogeneous* coordinates

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

homogeneous image  
coordinates

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

homogeneous scene  
coordinates

Converting *from* homogeneous coordinates

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

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

# Homogeneous Coordinates

Invariant to scaling

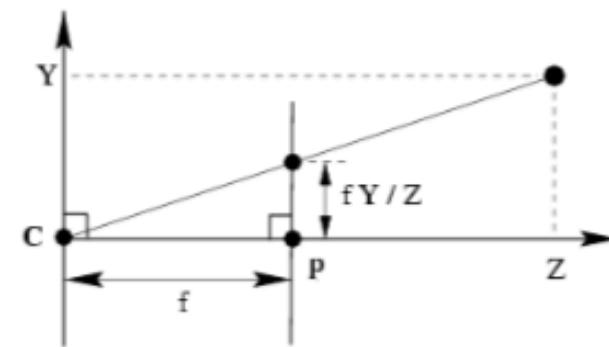
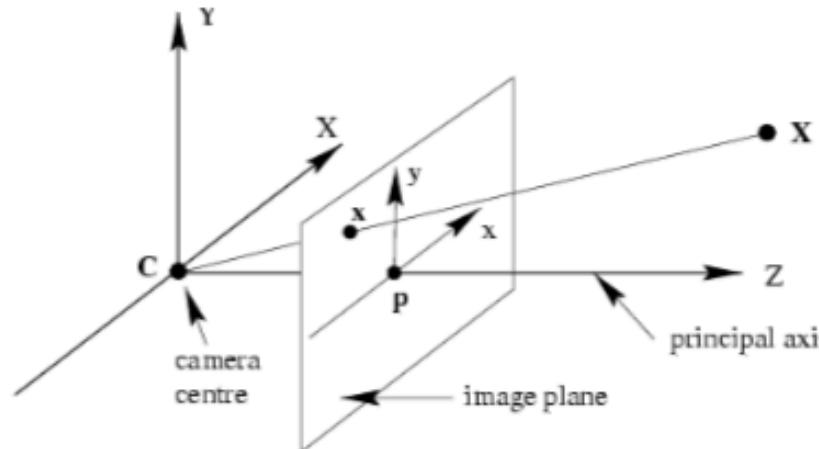
$$k \begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} kx \\ ky \\ kw \end{bmatrix} \Rightarrow \begin{bmatrix} \frac{kx}{kw} \\ \frac{ky}{kw} \end{bmatrix} = \begin{bmatrix} \frac{x}{w} \\ \frac{y}{w} \end{bmatrix}$$

Homogeneous  
Coordinates

Cartesian  
Coordinates

Point in Cartesian is a ray in Homogeneous.

# Pinhole camera model



- By similar triangles

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

- Dropping third coordinate

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

# Pinhole camera model

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

- Using homogeneous coordinates:

$$2D: (fX/Z, fY/Z) \mapsto (fX/Z, fY/Z, 1)^T = (fX, fY, Z)^T$$

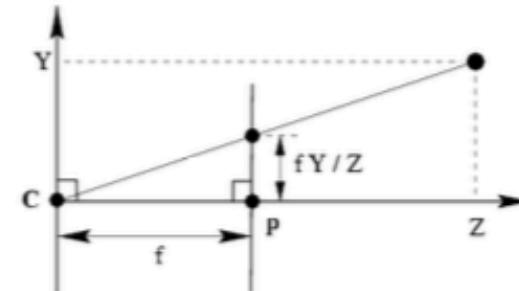
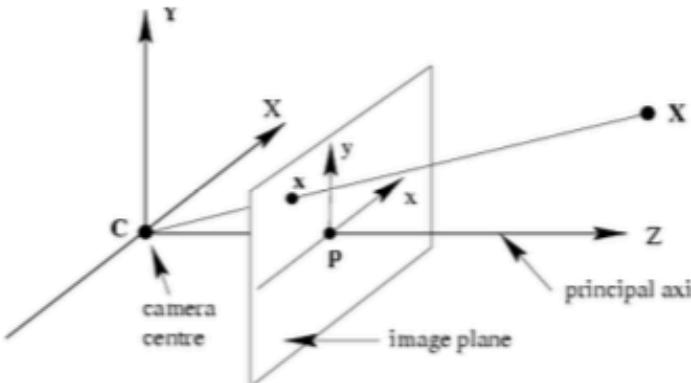
$$3D: (X, Y, Z) \mapsto (X, Y, Z, 1)$$

- Result:

- Linear projection in homogeneous coordinates!

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

# Pinhole camera model



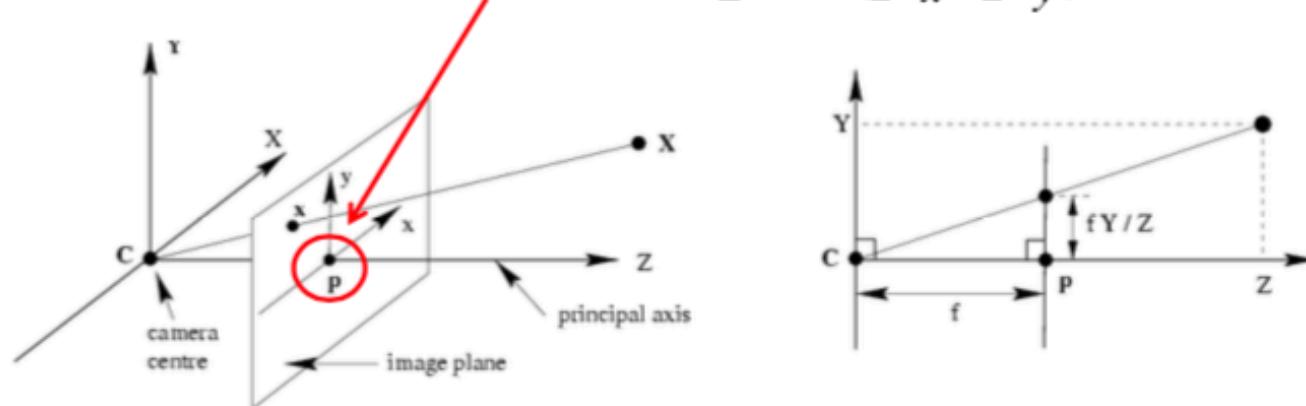
$$\begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{bmatrix} f & & \\ & f & \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ & 1 & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

$\Leftrightarrow \mathbf{x} = \mathbf{P}\mathbf{X}$     with     $\mathbf{P} = \text{diag}(f, f, 1)[\mathbf{I} | 0]$   
=  $3 \times 4$  homogeneous camera projection matrix

## Principal point

- The point where the principal axis intersects with the image plane

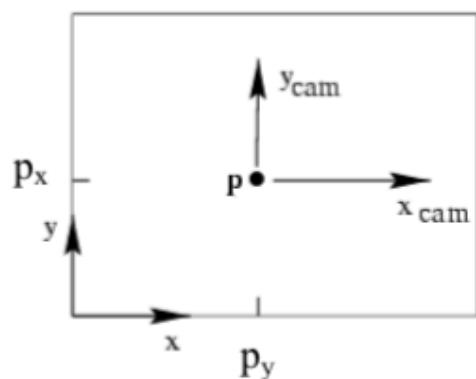
Principal point  $\mathbf{p} = (p_x, p_y)^T$



# Principal point offset

- So far:
  - Assumption that origin of points in image plane is at principal point
- Practically:
  - Origin of points may be somewhere else (e.g. border of image)

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



# Principal point offset

- Inhomogeneous:

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

- Homogeneous:

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

# Camera Calibration Matrix

$$\begin{pmatrix} fX + Zp_x \\ fY + Zp_y \\ Z \end{pmatrix} = \begin{bmatrix} f & p_x & 0 \\ f & p_y & 0 \\ 1 & 0 & \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$
$$= \begin{bmatrix} f & p_x \\ f & p_y \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$
$$= \mathbf{K}[\mathbf{I} | \mathbf{0}] \mathbf{X}$$

$$\boxed{\mathbf{K} = \begin{bmatrix} f & p_x \\ f & p_y \\ 1 \end{bmatrix}} \quad \text{Camera Parameters Only}$$

## CCD cameras



- “From image plane to pixel coordinates”

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

$$\begin{cases} x_0 = m_x p_x \\ y_0 = m_y p_y \\ \alpha_x = m_x f \\ \alpha_y = m_y f \end{cases}, \text{with } \begin{cases} m_x = \# \text{pixels / unit distance along x} \\ m_y = \# \text{pixels / unit distance along y} \end{cases}$$

CCD chip: pixels maybe rectangular!

## Camera Calibration Matrix

---

$K$  is a  $3 \times 3$  upper triangular matrix, called the **camera calibration matrix**:

$$K = \begin{bmatrix} \alpha_x & x_0 \\ \alpha_y & y_0 \\ & 1 \end{bmatrix}$$

- There are four parameters:
  - (i) The **scaling** in the image  $x$  and  $y$  directions,  $\alpha_x$  and  $\alpha_y$ .
  - (ii) The **principal point**  $(x_0, y_0)$ , which is the point where the optic axis intersects the image plane.
- The **aspect ratio** is  $\alpha_y/\alpha_x$ .

# Camera Calibration Matrix

---

$\mathbf{K}$  is a  $3 \times 3$  upper triangular matrix, called the camera calibration matrix:

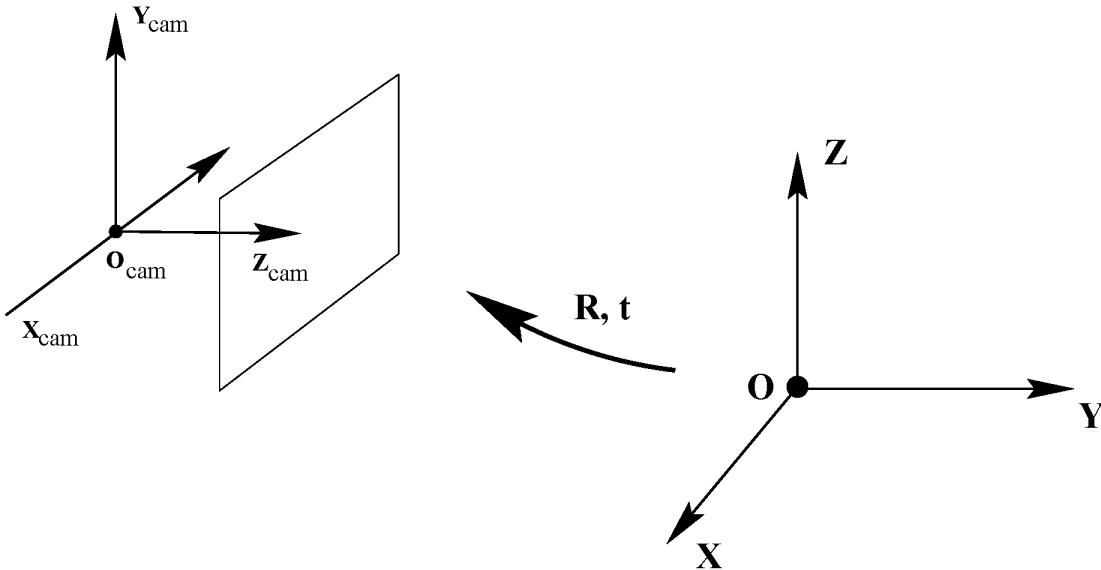
$$\mathbf{K} = \begin{bmatrix} \alpha_x & x_0 \\ \alpha_y & y_0 \\ & 1 \end{bmatrix}$$

- There are four parameters:
  - (i) The scaling in the image  $x$  and  $y$  directions,  $\alpha_x$  and  $\alpha_y$  equal to focal-length in pixel units.
  - (ii) The principal point  $(x_0, y_0)$ , which is the point where the optic axis intersects the image plane.
- The aspect ratio is  $\alpha_y/\alpha_x$ .

# World Coordinate System

---

## External camera parameters

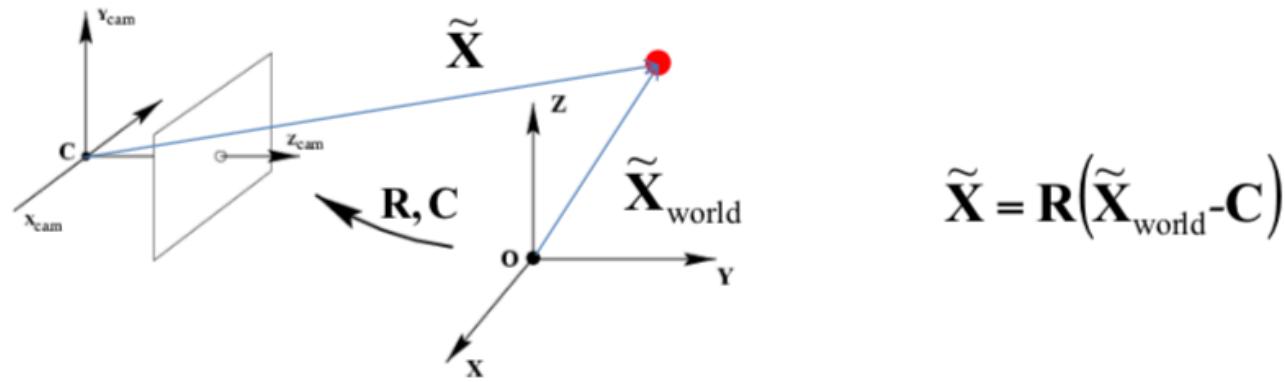
$$\begin{pmatrix} X_{\text{cam}} \\ Y_{\text{cam}} \\ Z_{\text{cam}} \\ 1 \end{pmatrix} = \begin{bmatrix} R & t \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$


Euclidean transformation between world and camera coordinates

- $R$  is a  $3 \times 3$  rotation matrix
- $t$  is a  $3 \times 1$  translation vector

# Extrinsic camera parameters

- Transform point into camera frame



$$\tilde{\mathbf{X}} = \mathbf{R}(\tilde{\mathbf{X}}_{\text{world}} - \mathbf{C})$$

- Homogeneous:

$$\mathbf{X} = \begin{bmatrix} \mathbf{R} & -\mathbf{RC} \\ 0 & 1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{bmatrix} \mathbf{R} & -\mathbf{RC} \\ 0 & 1 \end{bmatrix} \mathbf{X}_{\text{world}}$$

We could define the translation vector as  $\mathbf{t} = -\mathbf{RC}$

Concatenating the three matrices,

$$\mathbf{x} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = K \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & t \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = K[R|t] \mathbf{X}_{\text{world}}$$

which defines the  $3 \times 4$  projection matrix from Euclidean 3-space to an image as

$$\mathbf{x} = P \mathbf{X}_{\text{world}} \quad P = K[R|t] = KR[I|R^\top t]$$

Note, the camera centre is at  $(X, Y, Z)^\top = -R^\top t$ .

In the following it is often only the  $3 \times 4$  form of  $P$  that is important, rather than its decomposition.

Concatenating the three matrices,

$$\mathbf{x} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = K \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} R & t \\ 0^\top & 1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = K[R|t] \mathbf{X}_{\text{world}}$$

A red box highlights the matrix  $\begin{bmatrix} R & t \\ 0^\top & 1 \end{bmatrix}$ . A blue arrow points from the label  $\mathbf{X}_{\text{cam}}$  to the right side of the highlighted matrix.

which defines the  $3 \times 4$  projection matrix from Euclidean 3-space to an image as

$$\mathbf{x} = P \mathbf{X}_{\text{world}} \quad P = K[R|t] = KR[I|R^\top t]$$

Note, the camera centre is at  $(X, Y, Z)^\top = -R^\top t$ .

In the following it is often only the  $3 \times 4$  form of  $P$  that is important, rather than its decomposition.

Concatenating the three matrices,

$$\mathbf{x} \approx \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \mathbf{K} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \approx \mathbf{K} [\mathbf{R} | \mathbf{t}] \mathbf{X}_{\text{world}}$$

which defines the  $3 \times 4$  projection matrix from Euclidean 3-space to an image as

$$\mathbf{x} \approx \mathbf{P} \mathbf{X}_{\text{world}}$$
$$\mathbf{P} = \mathbf{K} [\mathbf{R} | \mathbf{t}]$$

Note, the camera centre is at  $(x, Y, Z)^\top = -\mathbf{R}^\top \mathbf{t}$ .       $\mathbf{t} = -\mathbf{R}\mathbf{C} = \mathbf{R}(-\mathbf{C})$

In the following it is often only the  $3 \times 4$  form of  $\mathbf{P}$  that is important, rather than its decomposition.

# General Projective Camera

$$\begin{aligned}\mathbf{x} &= \mathbf{K}[\mathbf{I} | 0]\mathbf{X} && \text{Skew parameter (mostly 0)} \\ &= \begin{bmatrix} \alpha_x & \mathbf{s} & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix} [\mathbf{I} | 0] \begin{bmatrix} \mathbf{R} & -\mathbf{RC} \\ 0 & 1 \end{bmatrix} \mathbf{X}_{\text{world}} \\ &= \mathbf{KR}[\mathbf{I} | -\mathbf{C}]\mathbf{X}_{\text{world}} && 5 + 3 + 3 \text{ DoF} \\ &= \mathbf{P}\mathbf{X}_{\text{world}} && \begin{array}{l} \text{Camera intrinsics} \\ \text{Camera extrinsics} \end{array} \\ &&& \text{3x4 projective camera matrix} \\ &&& 11 \text{ DoF (defined up to scale)}\end{aligned}$$

$$K = \begin{bmatrix} \alpha & \gamma & x_0 \\ 0 & \beta & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\alpha = \alpha_x$$

$$\beta = \frac{\alpha_y}{\sin \theta}$$

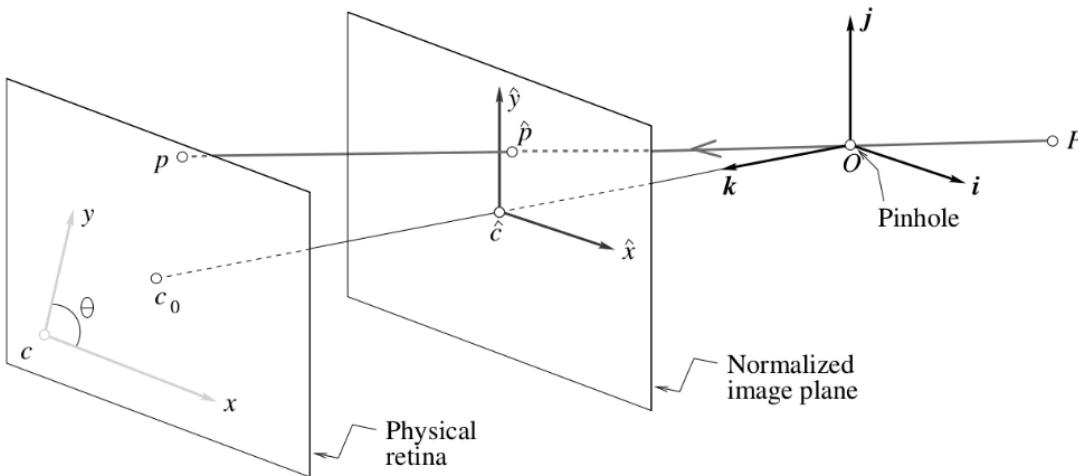
$$\gamma = -\alpha_x \cot \theta$$

$\theta$ : skew (angle b/w horizontal and vertical axes ,  $\pi/2$  for most real cameras)

# Internal calibration parameters

## Physical and Normalized Image Coordinate Systems

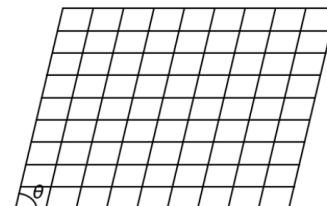
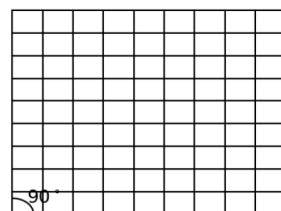
David A. Forsyth and Jean Ponce, Computer Vision A Modern Approach



Ideally aligned pixel grid (*Norm*)  
Figure 1(a)

Skewed pixel grid (*Skew*)  
Figure 1(b)

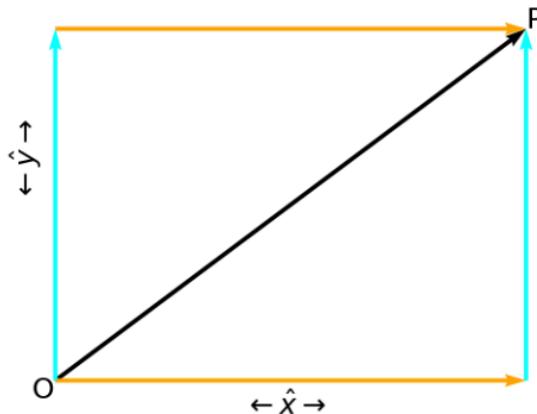
Case: Skew is not zero



# Internal calibration parameters

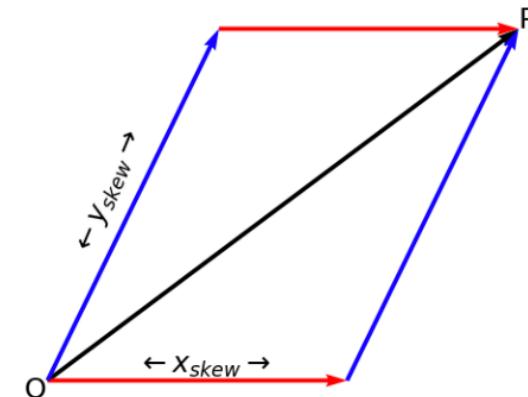
P as a linear combination of the cartesian axes vectors

Figure 2(a)



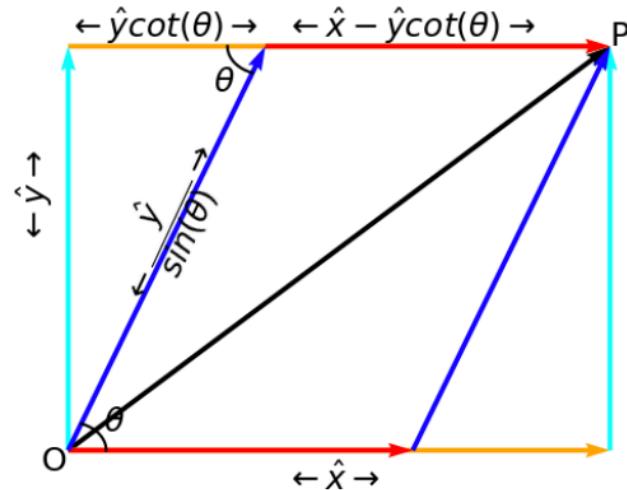
P as a linear combination of the skewed axes vectors

Figure 2(b)



*Skew coordinates in terms of Norm coordinates*

**Figure 3**



© immenselyhappy.com

As you can see

$$x_{skew} = \hat{x} - \hat{y}\cot(\theta)$$

$$y_{skew} = \frac{\hat{y}}{\sin(\theta)}$$

## A Projective Camera

---

The camera model for perspective projection is a linear map between homogeneous point coordinates

$$\lambda \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{bmatrix} & & & \\ & P \ (3 \times 4) & & \\ & & & \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

Image Point

Scene Point

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

- The camera centre is the null-vector of  $\mathbf{P}$   
e.g. if  $\mathbf{P} = [\mathbf{I}|0]$  then the centre is  $\mathbf{X} = (0, 0, 0, 1)^\top$ .
- $\mathbf{P}$  has 11 degrees of freedom (essential parameters).
- $\mathbf{P}$  has rank 3.

# Camera Calibration (Resectioning)

---

## Problem Statement:

Given  $n$  correspondences  $\mathbf{x}_i \leftrightarrow \mathbf{X}_i$ , where  $\mathbf{X}_i$  is a scene point and  $\mathbf{x}_i$  its image:

## Compute

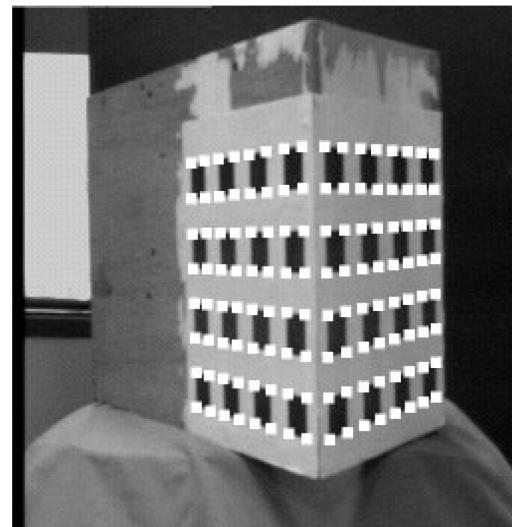
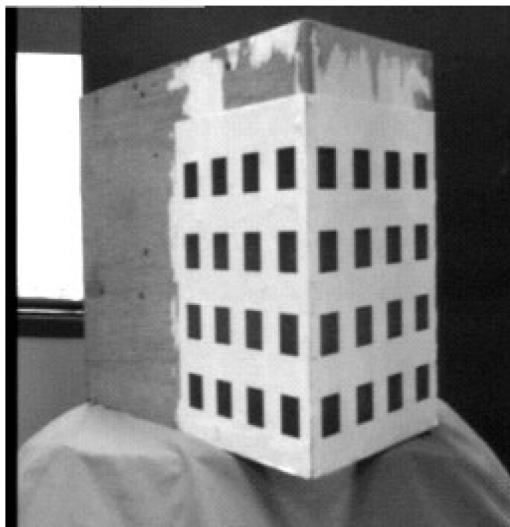
$\mathbf{P} = \mathbf{K} [\mathbf{R} | \mathbf{t}]$  such that  $\mathbf{x}_i = \mathbf{P}\mathbf{X}_i$ .

The algorithm for camera calibration has two parts:

- (i) Compute the matrix  $\mathbf{P}$  from a set of point correspondences.
- (ii) Decompose  $\mathbf{P}$  into  $\mathbf{K}$ ,  $\mathbf{R}$  and  $\mathbf{t}$  via the QR decomposition.

## Example - Calibration Object

---



Determine accurate corner positions by

- (i) Extract and link edges using Canny edge operator.
- (ii) Fit lines to edges using orthogonal regression.
- (iii) Intersect lines to obtain corners to sub-pixel accuracy.

The final error between measured and projected points is typically less than 0.02 pixels.

# The DLT algorithm – camera resection

( Direct Linear Transformation (DLT) )

---

Problem Statement:

Given  $n$  correspondences  $\mathbf{x}_i \leftrightarrow \mathbf{X}_i$ , points in the image and the world.

Compute camera matrix  $\mathbf{P}$  such that  $\mathbf{x}'_i \approx \mathbf{P}\mathbf{X}_i$ .

Each correspondence generates two equations

$$x_i = \frac{p_{11}X_i + p_{12}Y_i + p_{13}Z_i + p_{14}}{p_{31}X_i + p_{32}Y_i + p_{33}Z_i + p_{34}} \quad y_i = \frac{p_{21}X_i + p_{22}Y_i + p_{23}Z_i + p_{24}}{p_{31}X_i + p_{32}Y_i + p_{33}Z_i + p_{34}}$$

Multiplying out gives equations linear in the matrix elements of  $\mathbf{P}$

$$\begin{aligned} x_i(p_{31}X_i + p_{32}Y_i + p_{33}Z_i + p_{34}) &= p_{11}X_i + p_{12}Y_i + p_{13}Z_i + p_{14} \\ y_i(p_{31}X_i + p_{32}Y_i + p_{33}Z_i + p_{34}) &= p_{21}X_i + p_{22}Y_i + p_{23}Z_i + p_{24} \end{aligned}$$

These equations can be rearranged as

$$\begin{pmatrix} X & Y & Z & 1 & 0 & 0 & 0 & 0 & -xX & -xY & -xZ & -x \\ 0 & 0 & 0 & 0 & X & Y & Z & 1 & -yX & -yY & -yZ & -y \end{pmatrix} \mathbf{p} = \mathbf{0}$$

with  $\mathbf{p} = (p_{11}, p_{12}, p_{13}, p_{14}, p_{21}, p_{22}, p_{23}, p_{24}, p_{31}, p_{32}, p_{33}, p_{34})^\top$  a 12-vector.

## Camera resection continued

---

Solving for  $\mathbf{P}$

- (i) Concatenate the equations from ( $n \geq 6$ ) correspondences to generate  $2n$  simultaneous equations, which can be written:  $\mathbf{Ap} = \mathbf{0}$ , where  $\mathbf{A}$  is a  $2n \times 12$  matrix.
- (ii) In general this will not have an exact solution, but a (linear) solution which minimizes  $|\mathbf{Ap}|$ , subject to  $|\mathbf{p}| = 1$  is obtained from the eigenvector with least eigenvalue of  $\mathbf{A}^\top \mathbf{A}$ . Or equivalently from the vector corresponding to the smallest singular value of the SVD of  $\mathbf{A}$ .
- (iii) This linear solution is then used as the starting point for a non-linear minimization of the difference between the measured and projected point:

$$\min_{\mathbf{P}} \sum_i ((x_i, y_i) - \text{dehom}(\mathbf{P}(x_i, Y_i, z_i, 1)))^2$$

# Direct Linear Transformation (DLT)

## Objective

Given  $n \geq 6$  2D-to-3D point correspondences  $\{\mathbf{x}_i \leftrightarrow \mathbf{X}_i\}$ ,  
determine the  $3 \times 4$  projection matrix  $\mathbf{P}$  such that  $\mathbf{x}_i = \mathbf{P}\mathbf{X}_i$

## Algorithm

- (i) For each correspondence  $\{\mathbf{x}_i \leftrightarrow \mathbf{X}_i\}$  compute  $\mathbf{A}_i$ . Usually only the two first rows are needed.
- (ii) Assemble  $n$   $2 \times 12$  matrices  $\mathbf{A}_i$  into a single  $2n \times 12$  matrix  $\mathbf{A}$
- (iii) Compute the SVD of  $\mathbf{A}$ . The solution for  $\mathbf{p}$  is the last column of  $\mathbf{V}$
- (iv) Normalize  $\mathbf{p}$  to 1, and rearrange to obtain  $\mathbf{P}$

## Issues with nonlinear minimization / optimization

- Initialization
- Parametrization

R 矩阵计算：

$$\mathcal{R}_x(\theta_x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_x & -\sin \theta_x \\ 0 & \sin \theta_x & \cos \theta_x \end{bmatrix} = \exp\left(\theta_x \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}\right)$$

这里的  $\theta_x$  是 roll 角，和右手螺旋的方向相同（在yz平面逆时针）

- 绕  $y$ -轴的主动旋转定义为：

$$\mathcal{R}_y(\theta_y) = \begin{bmatrix} \cos \theta_y & 0 & -\sin \theta_y \\ 0 & 1 & 0 \\ \sin \theta_y & 0 & \cos \theta_y \end{bmatrix} = \exp\left(\theta_y \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}\right)$$

这里的  $\theta_y$  是 pitch 角，和右手螺旋的方向相同（在zx平面逆时针）

- 绕  $z$ -轴的主动旋转定义为：

$$\mathcal{R}_z(\theta_z) = \begin{bmatrix} \cos \theta_z & -\sin \theta_z & 0 \\ \sin \theta_z & \cos \theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} = \exp\left(\theta_z \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\right)$$

这里的  $\theta_z$  是 yaw 角，和右手螺旋的方向相同（在xy平面逆时针）

## Issues with nonlinear minimization / optimization

- Initialization
- Parametrization

$$\mathbf{x} = \mathbf{P}\mathbf{X} \quad \mathbf{P} = \mathbf{K} [\mathbf{R} | \mathbf{t}] = \mathbf{K}\mathbf{R}[\mathbf{I} | \mathbf{R}^\top \mathbf{t}]$$

$$\mathbf{K} = \begin{bmatrix} \alpha_x & x_0 \\ \alpha_y & y_0 \\ 1 & \end{bmatrix}$$

## Issues with nonlinear minimization / optimization

- Initialization
- Parametrization

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

$$\mathbf{P} = \mathbf{K} [\mathbf{R} | \mathbf{t}] = \mathbf{K}\mathbf{R}[\mathbf{I} | \mathbf{R}^\top \mathbf{t}]$$

$$\mathbf{K} = \begin{bmatrix} \alpha_x & x_0 \\ \alpha_y & y_0 \\ 1 & \end{bmatrix}$$

## Issues with nonlinear minimization / optimization

- Initialization
- Parametrization

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

$$\mathbf{P} = \mathbf{K} [\mathbf{R} | \mathbf{t}] = \mathbf{K}\mathbf{R}[\mathbf{I} | \mathbf{R}^\top \mathbf{t}]$$

$$\mathbf{K} = \begin{bmatrix} \alpha_x & x_0 \\ \alpha_y & y_0 \\ 1 & \end{bmatrix}$$

Parameters for P:

- 3 parameters for rotation (angle-axis representation) or 4 (quaternions)
- 2-parameters for principal point
- 2 parameters for scale  $\alpha_x = \alpha_y$
- (1 parameter if scale  $\alpha_x = \alpha_y$ )

## Camera resection continued: Decompose P into K, R and t

---

The first  $3 \times 3$  submatrix,  $\mathbf{M}$ , of  $\mathbf{P}$  is the product ( $\mathbf{M} = \mathbf{KR}$ ) of an upper triangular and rotation matrix.

RQ

- (i) Factor  $\mathbf{M}$  into  $\mathbf{KR}$  using the QR matrix decomposition. This determines  $\mathbf{K}$  and  $\mathbf{R}$ .
- (ii) Then

$$\mathbf{t} = \mathbf{K}^{-1}(p_{14}, p_{24}, p_{34})^\top$$

Note, this produces a matrix with an extra skew parameter  $s$

$$\mathbf{K} = \begin{bmatrix} \alpha_x & s & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix}$$

# Reading

- Multiple View Geometry in Computer Vision: Chapter 6 (Camera models)