3D Vision-2021

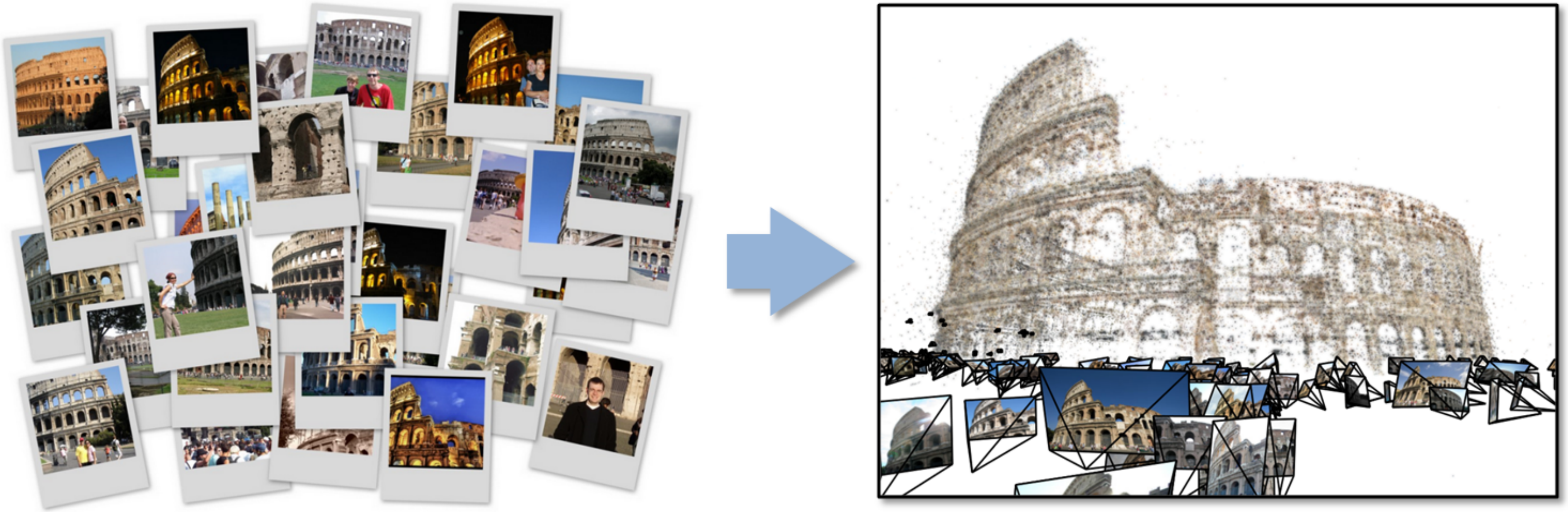
3D Computer Vision

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

[Slide from authors]

CVPR SEATTLE WASHINGTON JUNE 16-18 2020

[1] Australian National University

[2] NVIDIA

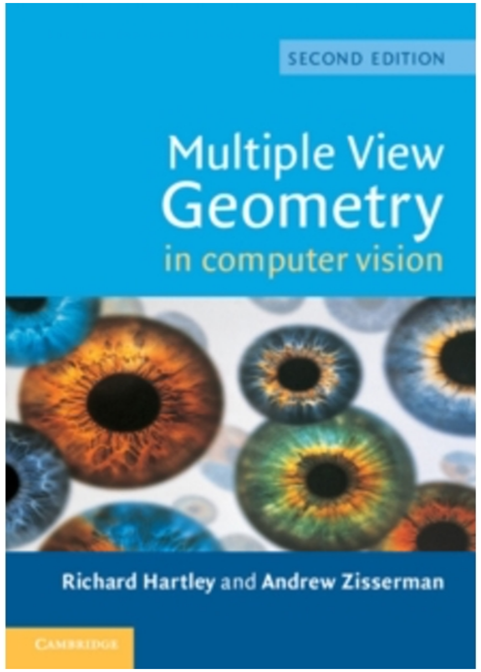
Cost Volume Pyramid Based Depth Inference for Multi-View Stereo

Jiayu Yang[1], Wei Mao[1], Jose M. Alvarez[2], Miaomiao Liu[1]

Fast Compact Flexible Accurate

Reference for 3D Vision

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

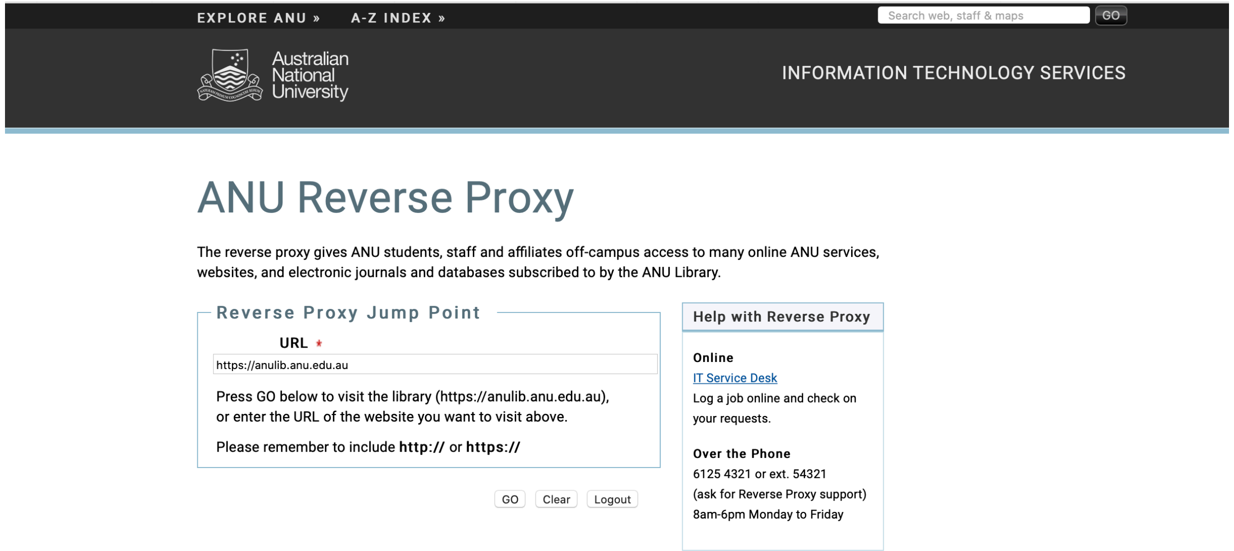


Get the ebook

- Step 1: Go to https://login.virtual.anu.edu.au/menu

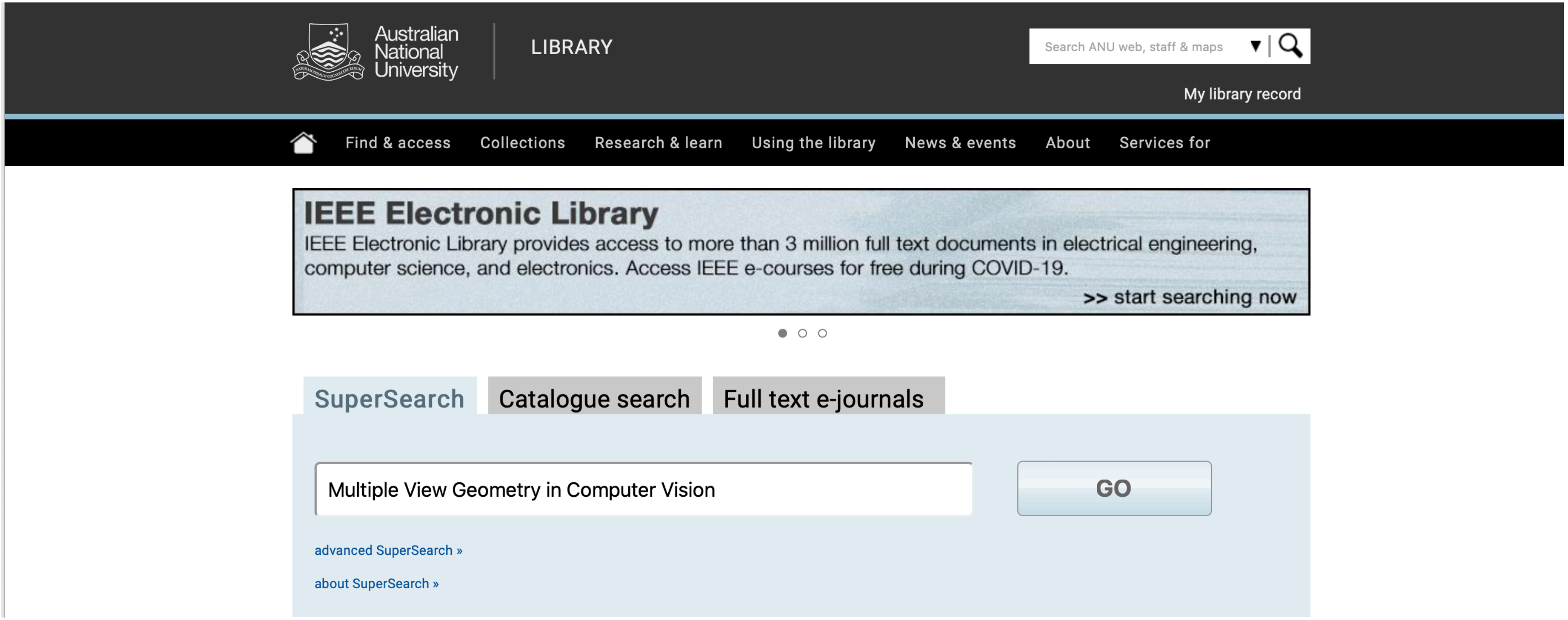
Step 2: Login with your ANU student id and password

Step 3: You will be directed to https://anulib.anu.edu.au



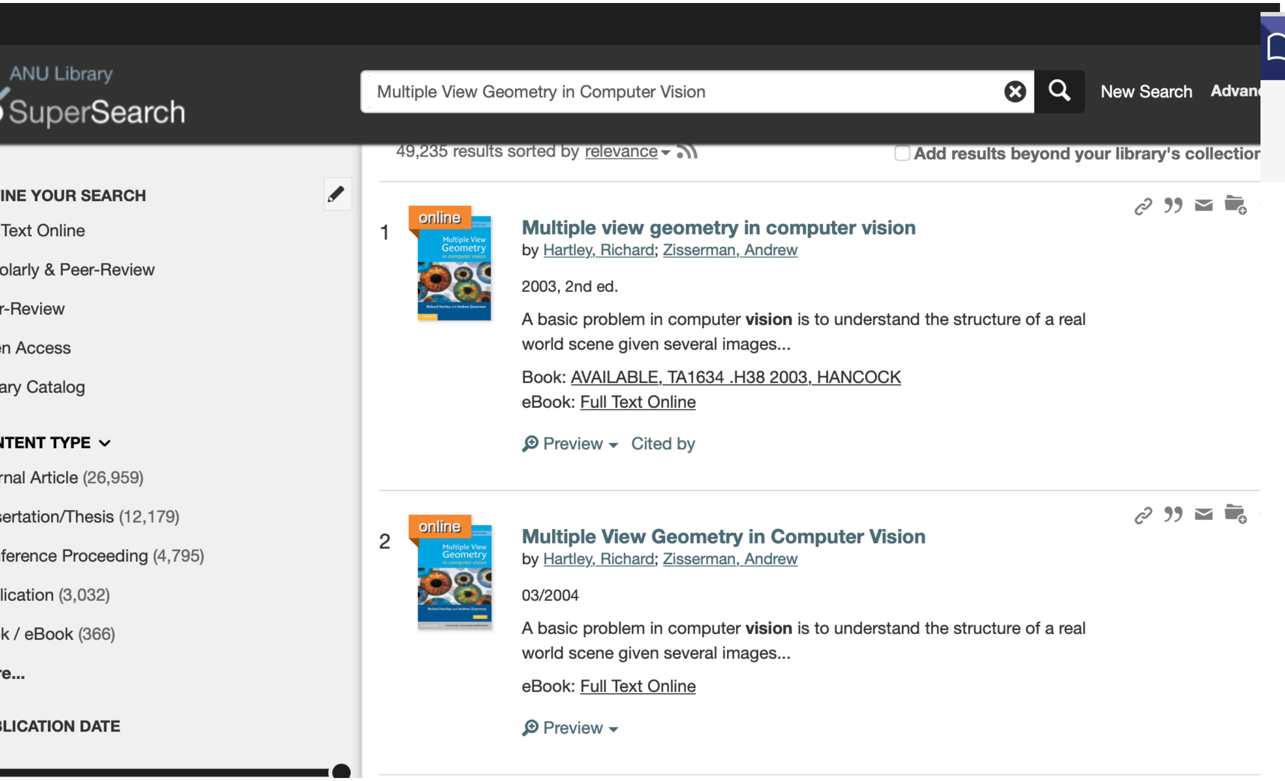
Get the ebook

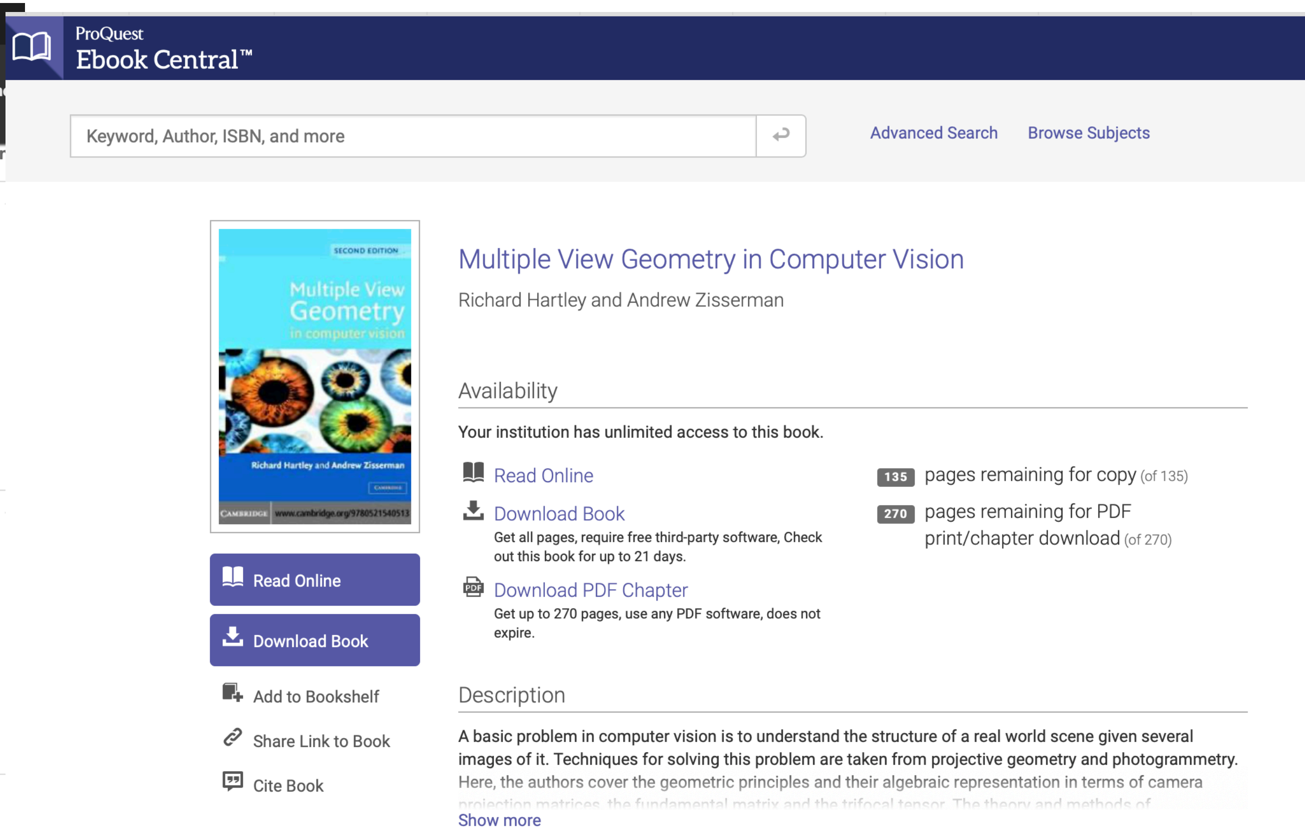
- Step 4: Input "Multiple View Geometry in Computer Vision"



Get the ebook

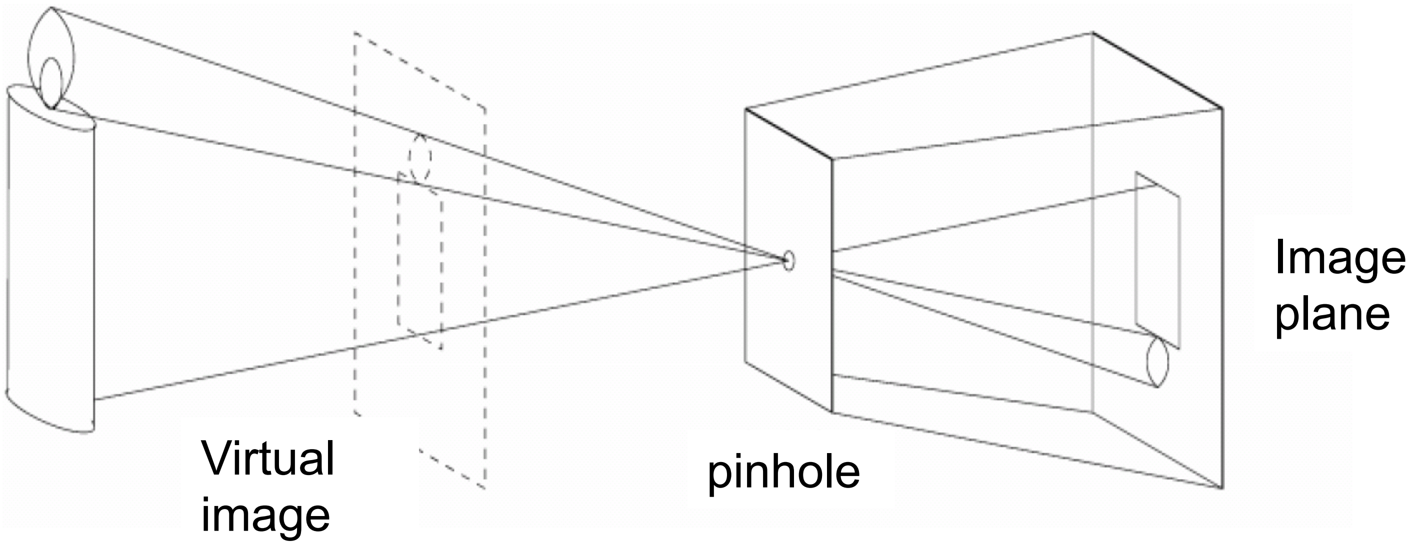
- Step 5: Check the book online





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.

Fig from Forsyth and Ponce

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

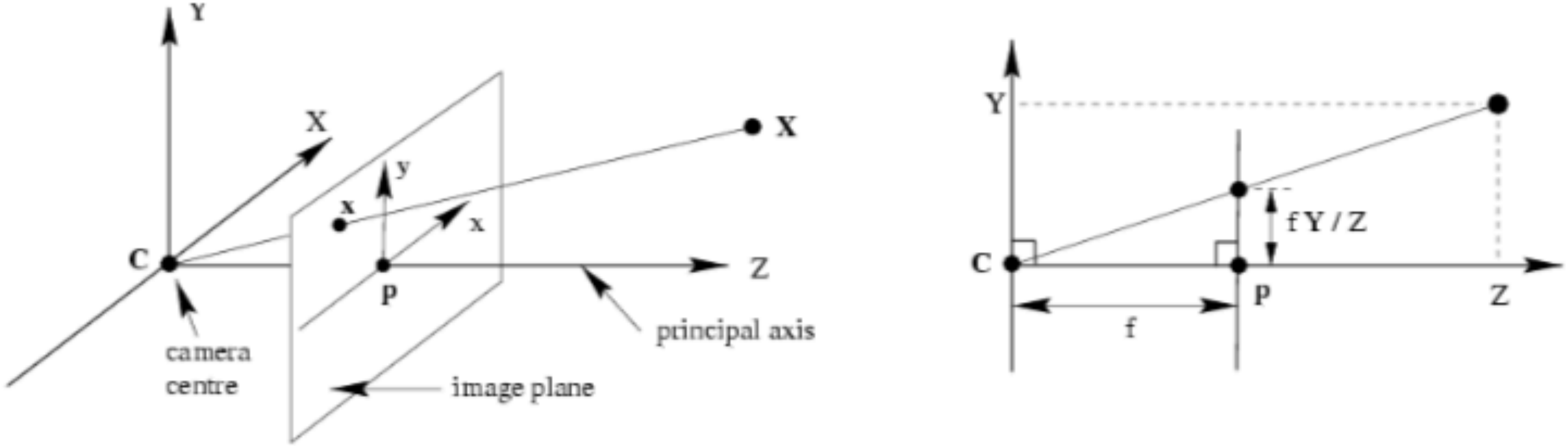
$$k\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} kx \\ ky \\ kw \end{bmatrix} \Rightarrow \begin{bmatrix} \frac{kx}{kw} \\ \frac{ky}{kw} \end{bmatrix} = \begin{bmatrix} \frac{x}{w} \\ \frac{y}{w} \end{bmatrix}$$

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

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

- Using homogeneous coordinates:

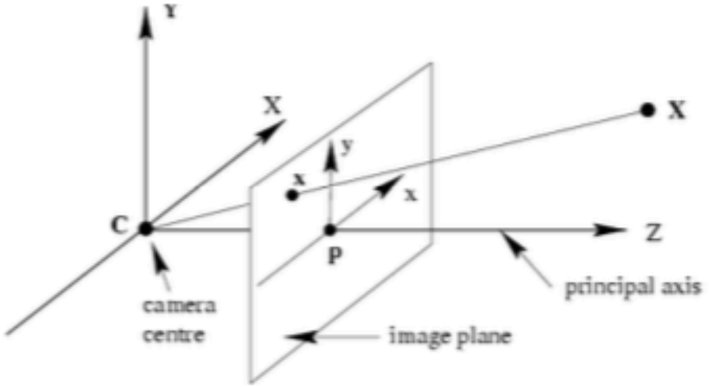
$$\text{2D}: (fX/Z, fY/Z) \mapsto (fX/Z, fY/Z, 1)^T = (fX, fY, Z)^T$$

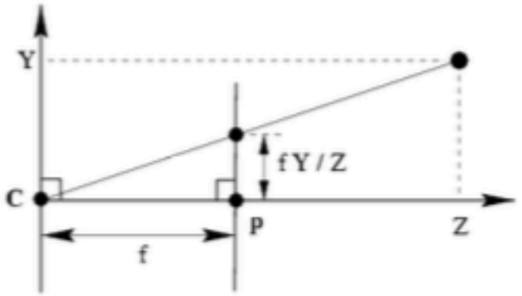
$$\text{3D}: (X,Y,Z) \mapsto (X,Y,Z,1)$$

- 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





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

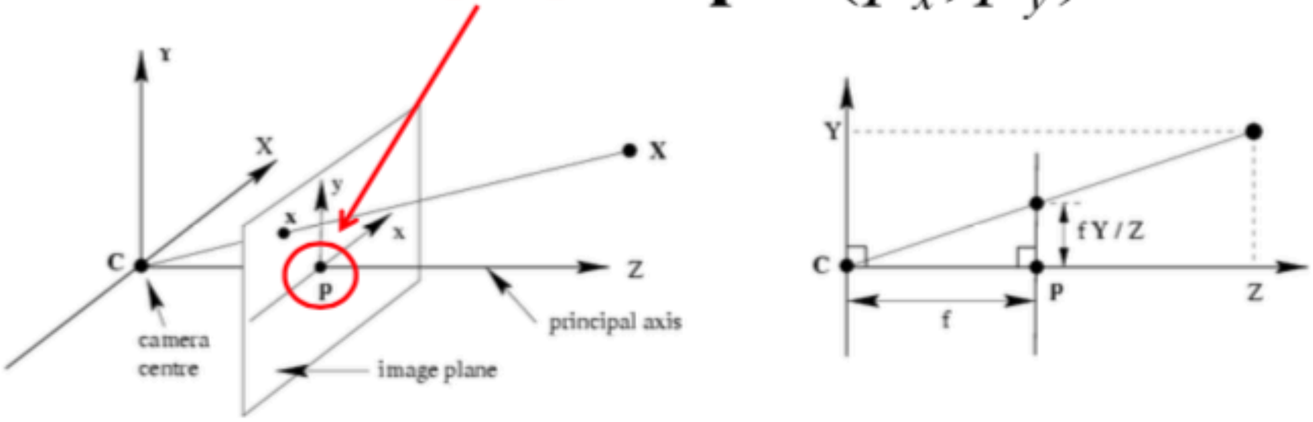
$$\Leftrightarrow \mathbf{x} = \mathbf{PX} \quad \text{with} \quad \mathbf{P} = \mathrm{diag}(f, f, 1)[\mathbf{I} \mid 0]$$

$$= 3 \times 4 \text{ homogeneous camera projection matrix}$$

Principal point

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

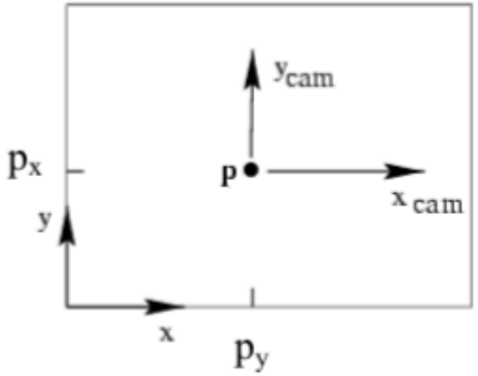
Principal point $\mathbf{p} = (p_x, p_y)^T$



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:

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

Camera Calibration Matrix

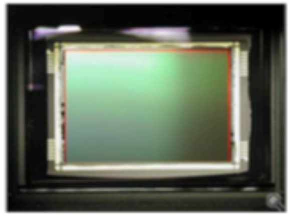
$$\begin{pmatrix} fX + Zp_x \\ fY + Zp_x \\ Z \end{pmatrix} = \begin{bmatrix} f & & p_x & 0 \\ & f & p_y & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

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

$$= \mathbf{K}[\mathbf{I} \mid \mathbf{0}]\mathbf{X}$$

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

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

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

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

- There are four parameters:

  (i) The scaling in the image x and y directions, α_x and α_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 α_y / α_x.

Camera Calibration Matrix

$\mathbf{K}$ is a 3×3 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, α_x and α_y equal to focal-length in pixel units.

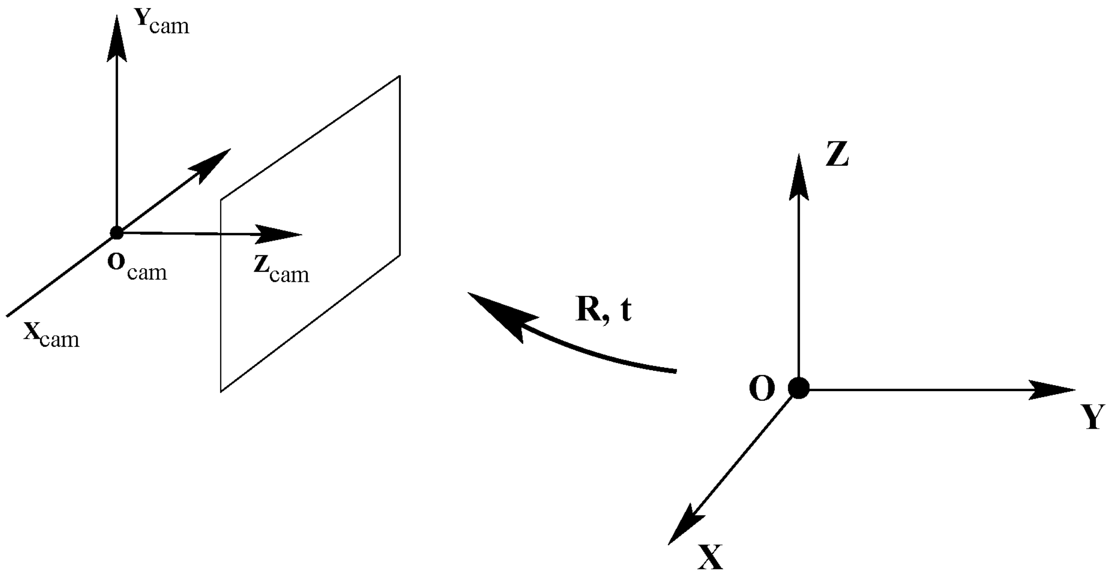
  (ii) The principal point (x_0, y_0), which is the point where the optic axis intersects the image plane.

- The aspect ratio is α_y / α_x.

World Coordinate System

External camera parameters

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

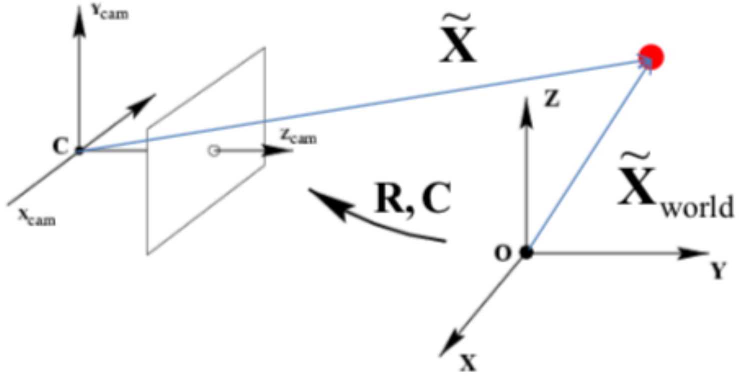


Euclidean transformation between world and camera coordinates

- $\mathtt{R}$ is a 3×3 rotation matrix
- $\mathbf{t}$ is a 3×1 translation vector

Extrinsic camera parameters

- Transform point into camera frame



$$\widetilde{\mathbf{X}} = \mathbf{R}\left(\widetilde{\mathbf{X}}_{world} - \mathbf{C}\right)$$

- Homogeneous:

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

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

Concatenating the three matrices,

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

which defines the 3×4 projection matrix from Euclidean 3-space to an image as

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

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

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

Concatenating the three matrices,

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

which defines the 3×4 projection matrix from Euclidean 3-space to an image as

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

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

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

Concatenating the three matrices,

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

which defines the 3×4 projection matrix from Euclidean 3-space to an image as

$$\boxed{\begin{aligned} \mathbf{x} &\approx \mathbf{P}\mathbf{X}_{\text{world}} \\ \mathbf{P} &= \mathbf{K}\,[\mathbf{R}|\ \mathbf{t}] \end{aligned}}$$

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

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

General Projective Camera

$$\mathbf{x} = \mathbf{K}\left[\mathbf{I} \mid 0\right]\mathbf{X}$$

Skew parameter (mostly 0)

$$= \begin{bmatrix} \alpha_x & s & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix} \left[\mathbf{I} \mid 0\right] \begin{bmatrix} \mathbf{R} & -\mathbf{RC} \\ 0 & 1 \end{bmatrix} \mathbf{X}_{\text{world}}$$

$$= \underbrace{\mathbf{KR}}_{\text{Camera intrinsics}}\underbrace{\left[\mathbf{I} \mid -\mathbf{C}\right]}_{\text{Camera extrinsics}}\mathbf{X}_{\text{world}}$$

5 + 3 + 3 DoF

$$= \mathbf{P}\mathbf{X}_{\text{world}}$$

3x4 projective camera matrix

11 DoF (defined up to scale)

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

$$\alpha = \alpha_x$$

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

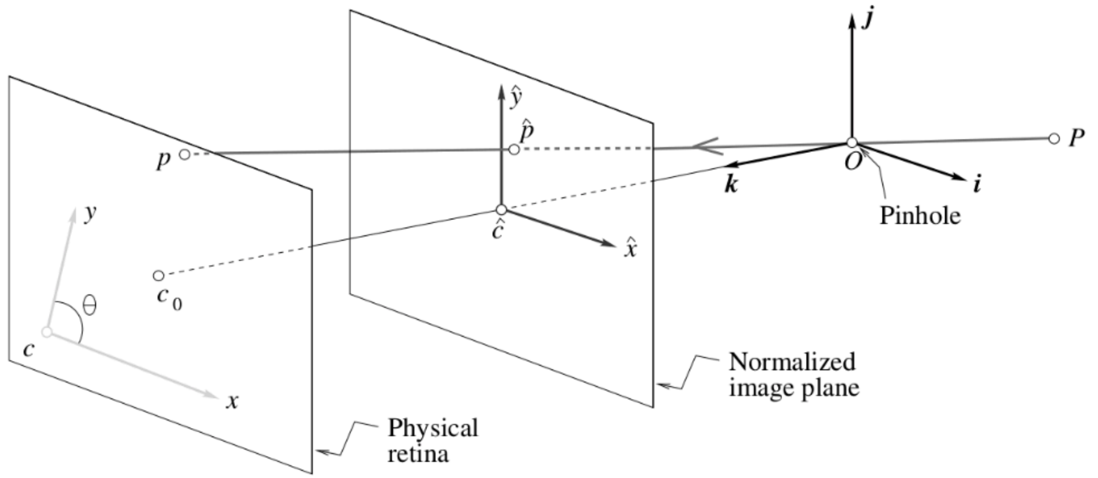
$$\gamma = -\alpha_x \cot\theta$$

θ: skew (angle b/w horizontal and vertical axes , pi/2 for most real cameras

Internal calibration parameters

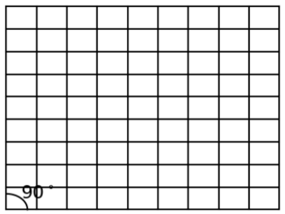
Physical and Normalized Image Coordinate Systems

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

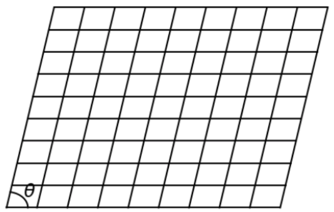


Case: Skew is not zero

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

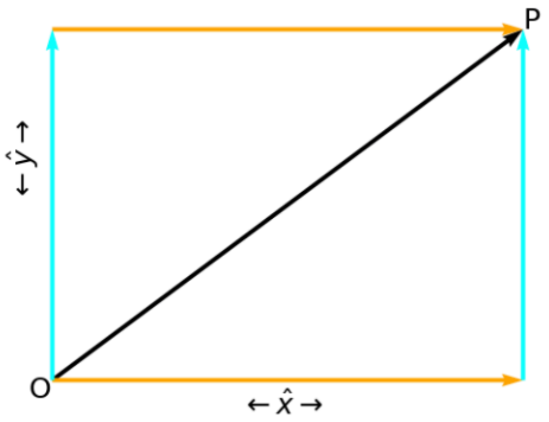


Skewed pixel grid ($Skew$)
Figure 1(b)

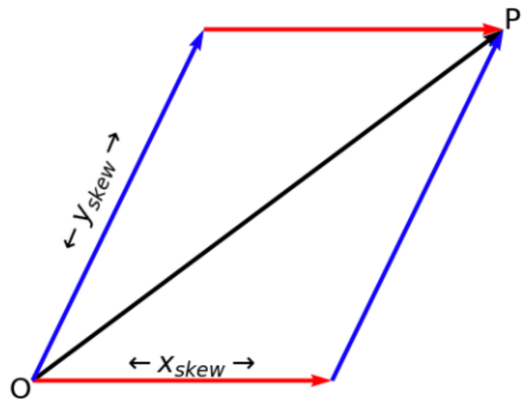


Internal calibration parameters

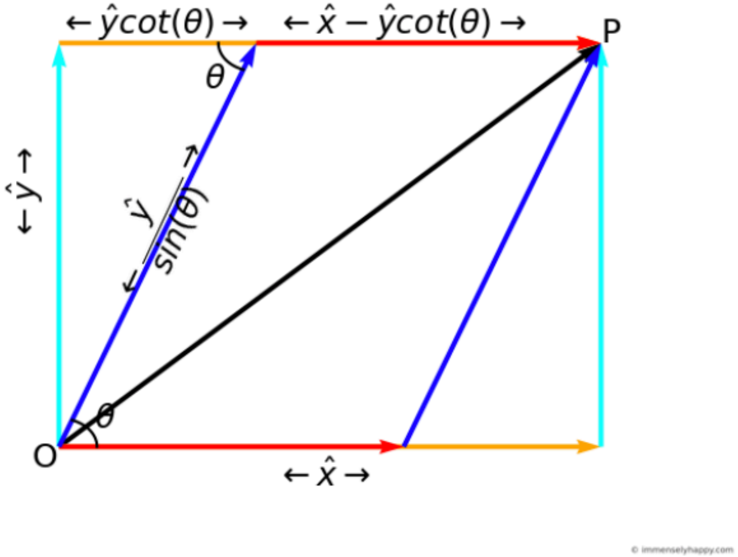
P as a linear combination of the cartesian axes vectors
Figure 2(a)



P as a linear combination of the skewed axes vectors
Figure 2(b)



$Skew$ coordinates in terms of $Norm$ coordinates
Figure 3



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

$$\lambda \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \left[\quad \mathrm{P} \;\; (3 \times 4) \quad \right] \begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \\ 1 \end{pmatrix}$$

Image Point Scene Point

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

- 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

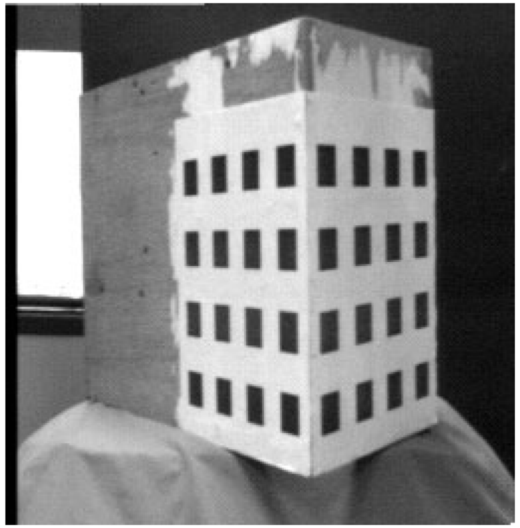
$\mathtt{P} = \mathtt{K}\,[\mathtt{R}|\ \mathbf{t}]$ such that $\mathbf{x}_i = \mathtt{P}\mathbf{X}_i$.

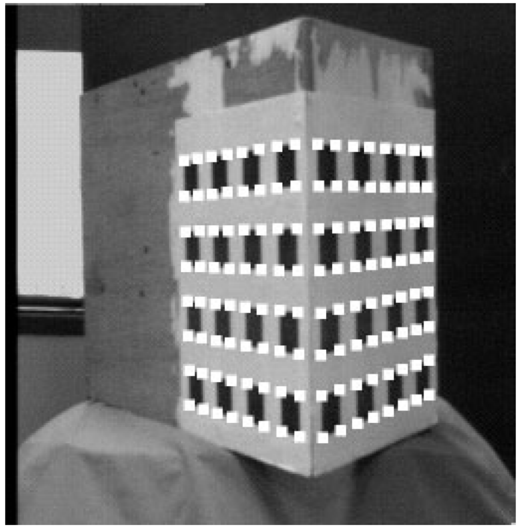
The algorithm for camera calibration has two parts:

(i) Compute the matrix $\mathtt{P}$ from a set of point correspondences.

(ii) Decompose $\mathtt{P}$ into $\mathtt{K}, \mathtt{R}$ and $\mathtt{t}$ via the $\mathtt{QR}$ 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

$$x_i = \frac{p_{11}\mathrm{X}_i + p_{12}\mathrm{Y}_i + p_{13}\mathrm{Z}_i + p_{14}}{p_{31}\mathrm{X}_i + p_{32}\mathrm{Y}_i + p_{33}\mathrm{Z}_i + p_{34}} \qquad y_i = \frac{p_{21}\mathrm{X}_i + p_{22}\mathrm{Y}_i + p_{23}\mathrm{Z}_i + p_{24}}{p_{31}\mathrm{X}_i + p_{32}\mathrm{Y}_i + p_{33}\mathrm{Z}_i + p_{34}}$$

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

$$\begin{aligned} x_i(p_{31}\mathrm{X}_i + p_{32}\mathrm{Y}_i + p_{33}\mathrm{Z}_i + p_{34}) &= p_{11}\mathrm{X}_i + p_{12}\mathrm{Y}_i + p_{13}\mathrm{Z}_i + p_{14} \\ y_i(p_{31}\mathrm{X}_i + p_{32}\mathrm{Y}_i + p_{33}\mathrm{Z}_i + p_{34}) &= p_{21}\mathrm{X}_i + p_{22}\mathrm{Y}_i + p_{23}\mathrm{Z}_i + p_{24} \end{aligned}$$

These equations can be rearranged as

$$\begin{pmatrix} \mathrm{X} & \mathrm{Y} & \mathrm{Z} & 1 & 0 & 0 & 0 & 0 & -x\mathrm{X} & -x\mathrm{Y} & -x\mathrm{Z} & -x \\ 0 & 0 & 0 & 0 & \mathrm{X} & \mathrm{Y} & \mathrm{Z} & 1 & -y\mathrm{X} & -y\mathrm{Y} & -y\mathrm{Z} & -y \end{pmatrix} \mathbf{p} = \mathbf{0}$$

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

$$\min_{\mathbf{P}} \sum_i \left((x_i, y_i) - \mathrm{dehom}(\mathbf{P}(\mathrm{X}_i, \mathrm{Y}_i, \mathrm{Z}_i, 1))\right)^2$$

Direct Linear Transformation (DLT)

Objective

Given $n \geq 6$ 2D-to-3D point correspondences $\{\mathbf{x}_i \leftrightarrow \mathbf{X}_i\}$, determine the 3x4 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 2x12 matrices $\mathbf{A}_i$ into a single 2nx12 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 θ_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 θ_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 θ_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

$$\mathbf{x} = \mathtt{P}\mathbf{X} \qquad \mathtt{P} = \mathtt{K}\,[\mathtt{R}|\ \mathbf{t}] = \mathtt{KR}[\mathtt{I}|\ \mathtt{R}^\top\mathbf{t}]$$

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

Issues with nonlinear minimization / optimization

- Initialization
- Parametrization

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

$$\boxed{\begin{aligned} \mathrm{P} &= \mathrm{K}\,[\mathrm{R}|\ \mathbf{t}] = \mathrm{KR}[\mathrm{I}|\ \mathrm{R}^\top \mathbf{t}] \\ \mathbf{K} &= \begin{bmatrix} \alpha_x & & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix} \end{aligned}}$$

Issues with nonlinear minimization / optimization

- Initialization
- Parametrization

$$\mathbf{x} = \mathrm{P}\mathbf{X} \qquad \mathrm{P} = \mathrm{K}\,[\mathrm{R}|\ \mathbf{t}] = \mathrm{KR}[\mathrm{I}|\ \mathrm{R}^{\top}\mathbf{t}]$$

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

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 $\mathbf{P}$ into $\mathbf{K}$, $\mathbf{R}$ and $\mathbf{t}$

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

$\boxed{\text{RQ}}$

(i) Factor $\mathbf{M}$ into $\mathbf{K}\mathbf{R}$ using the $\mathbf{Q}\mathbf{R}$ 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

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