Notation

(i) From now on, we assume that all points are given in homogenous coordinates

(ii) Points (in homogeneous coordinates) are denoted simply by $\mathbf{x}$, and not by $\hat{\mathbf{x}}$.

(iii) The symbol $\approx$ means *equal up to scale*, or *equivalent as homogeneous vectors*.

Homographies

(i) Affine (infinity-preserving) mappings are sometimes not enough to capture the transformations that apply to images.

(a) Views taken from different viewpoints.

(b) Correspondences between world-plane and image.

(ii) We need a transformation:

$$\begin{aligned}\mathbf{x}' &\approx \mathbf{H}\mathbf{x}\\ &= \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}\end{aligned}$$

(iii) In non-homogeneous coordinates:

$$x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{13}}$$

$$y' = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{13}}$$

(iv) They are transformations of P^2 (the projective plane).

The DLT algorithm – computing a homography

Problem Statement:

Given n correspondences $\mathbf{x}'_i \leftrightarrow \mathbf{x}_i$, points in two images.

Compute homography $\mathbf{H}$ such that $\mathbf{x}'_i \approx \mathbf{H}\mathbf{x}_i$.

Each correspondence generates two equations

$$x'_i = \frac{h_{11}x_i + h_{12}y_i + h_{13}}{h_{31}x_i + h_{32}y_i + h_{33}} \qquad y'_i = \frac{h_{21}x_i + h_{22}y_i + h_{23}}{h_{31}x_i + h_{32}y_i + h_{33}}$$

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

$$\begin{aligned} x'_i(h_{31}x_i + h_{32}y_i + h_{33}) &= h_{11}x_i + h_{12}y_i + h_{13} \\ y'_i(h_{31}x_i + h_{32}y_i + h_{33}) &= h_{21}x_i + h_{22}y_i + h_{23} \end{aligned}$$

These equations can be rearranged as

$$\begin{pmatrix} x & y & 1 & 0 & 0 & 0 & -x'x & -x'y & -x' \\ 0 & 0 & 0 & x & y & 1 & -y'x & -y'y & -y' \end{pmatrix} \mathbf{h} = \mathbf{0}$$

with $\mathbf{h} = (h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, h_{31}, h_{32}, h_{33})^\top$ a 9-vector.

Computing homography continued ...

Solving for $\mathbf{H}$

(i) Concatenate the equations from $(n \geq 4)$ correspondences to generate $2n$ simultaneous equations, which can be written: $\mathbf{A}\mathbf{h} = \mathbf{0}$, 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 $|\mathbf{A}\mathbf{h}|$, subject to $|\mathbf{h}| = 1$ 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:

$$\min_{\mathbf{H}} \sum_i \big((x_i', y_i') - \mathrm{dehom}(\mathbf{H}(x_i, y_i, 1))\big)^2$$

where

(i) hom is the homogenizing operation $(x, y) \mapsto (x, y, 1)$.

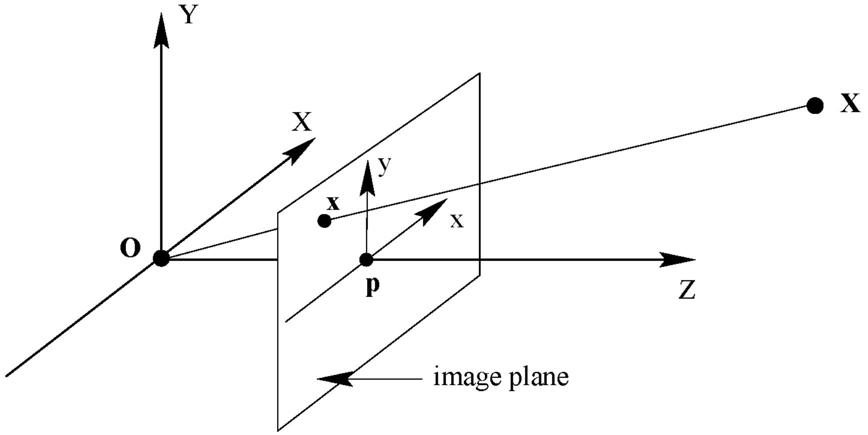
(ii) dehom is the dehomogenizing operation $(x, y, w) \mapsto (x/w, y/w)$.

Imaging Geometry

Perspective projection

$$\lambda \begin{pmatrix} x \\ y \\ f \end{pmatrix} = \begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \end{pmatrix}$$

where $\lambda = \mathrm{Z}/f$.



This can be written as a linear mapping between homogeneous coordinates (the equation is only up to a scale factor):

$$\begin{pmatrix} x \\ y \\ f \end{pmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \\ 1 \end{pmatrix}$$

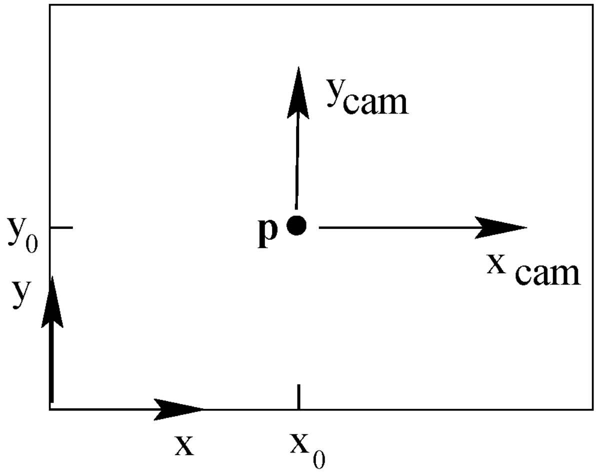
where a 3×4 projection matrix represents a map from 3D to 2D.

Image Coordinate System

Internal camera parameters

$$k_x \, x_{\text{cam}} = x - x_0$$
$$k_y \, y_{\text{cam}} = y - y_0$$

where the units of k_x, k_y are $[\text{pixels}/\text{length}]$.



$$\mathbf{x} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \frac{1}{f} \begin{bmatrix} \alpha_x & & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix} \begin{pmatrix} x_{\text{cam}} \\ y_{\text{cam}} \\ f \end{pmatrix} = \mathtt{K} \begin{pmatrix} x_{\text{cam}} \\ y_{\text{cam}} \\ f \end{pmatrix}$$

where $\alpha_x = f k_x, \; \alpha_y = f k_y$.

Image Coordinate System

Internal camera parameters

This gives

$$\mathbf{x} \approx \begin{bmatrix} \alpha_x & & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix} \begin{pmatrix} x_{\text{cam}} \\ y_{\text{cam}} \\ f \end{pmatrix} = \mathbf{K} \begin{pmatrix} x_{\text{cam}} \\ y_{\text{cam}} \\ f \end{pmatrix}$$

where $\alpha_x = fk_x,\ \alpha_y = fk_y$.

Units of α_x and α_y.

$$\begin{aligned} \alpha_x &= fk_x \\ &= \text{(focal length in mm)} \times \text{(pixels / mm)} \\ &= \text{focal length in pixels} \end{aligned}$$

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.

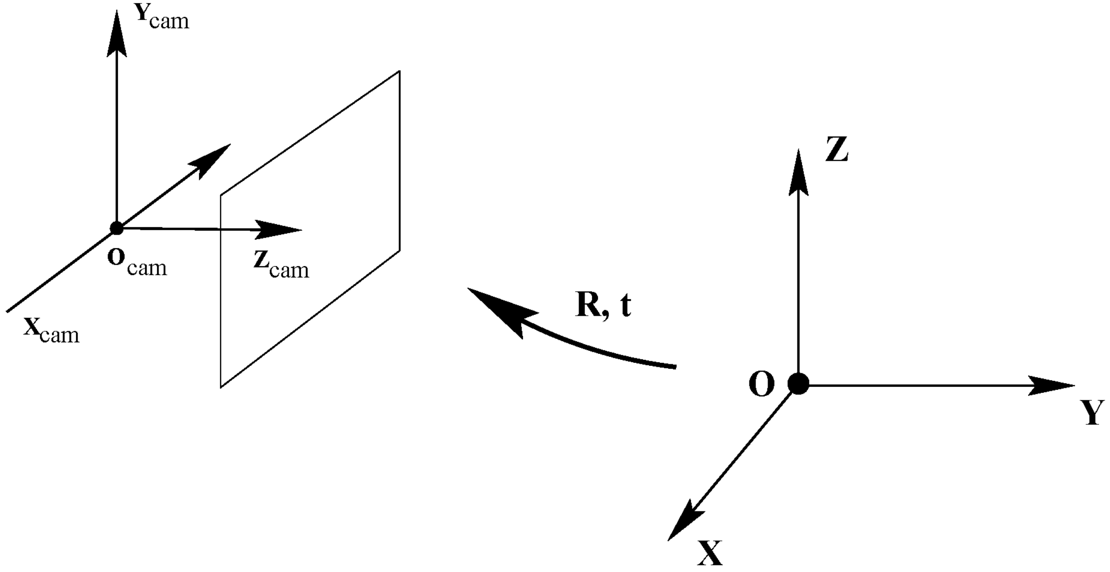
  (ii) The principal point (x_0, y_0), which is the point where the optic axis intersects the image plane.

- The aspect ratio is α_y / α_x.

World Coordinate System

External camera parameters

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



Euclidean transformation between world and camera coordinates

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

Concatenating the three matrices,

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

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

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

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,

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

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

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

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

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

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

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

- 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

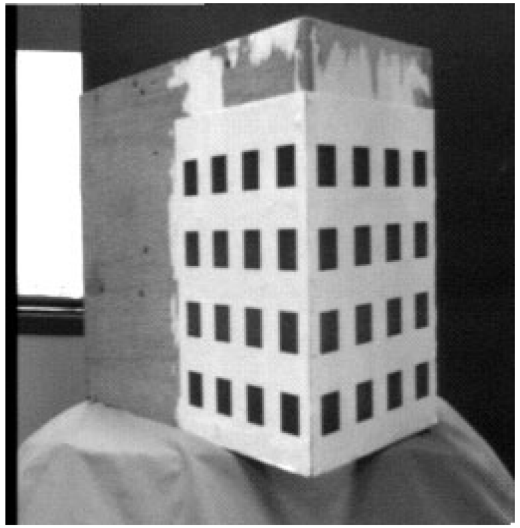
$\mathtt{P} = \mathtt{K}\left[\mathtt{R} \middle| \ \mathbf{t}\right]$ such that $\mathbf{x}_i = \mathtt{P}\mathbf{X}_i$.

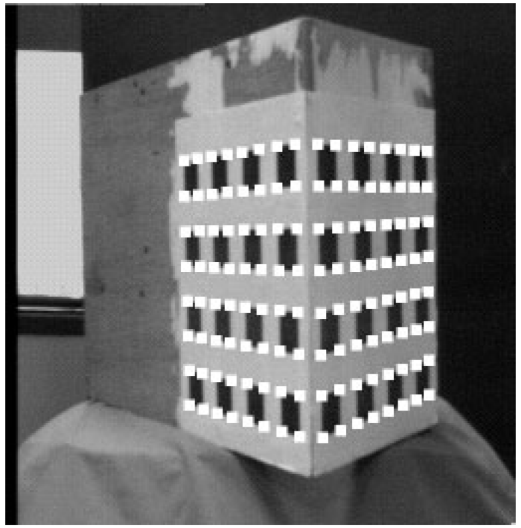
The algorithm for camera calibration has two parts:

(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

Problem Statement:

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

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

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:

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

Issues with nonlinear minimization / optimization

- Initialization
- Parametrization

Issues with nonlinear minimization / optimization

- Initialization
- Parametrization

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

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

Issues with nonlinear minimization / optimization

- Initialization
- Parametrization

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

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

Issues with nonlinear minimization / optimization

- Initialization
- Parametrization

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

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

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

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 $(\mathbf{M} = \mathbf{KR})$ of an upper triangular and rotation matrix.

(i) Factor $\mathbf{M}$ into $\mathbf{KR}$ using the $\mathbf{QR}$ 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

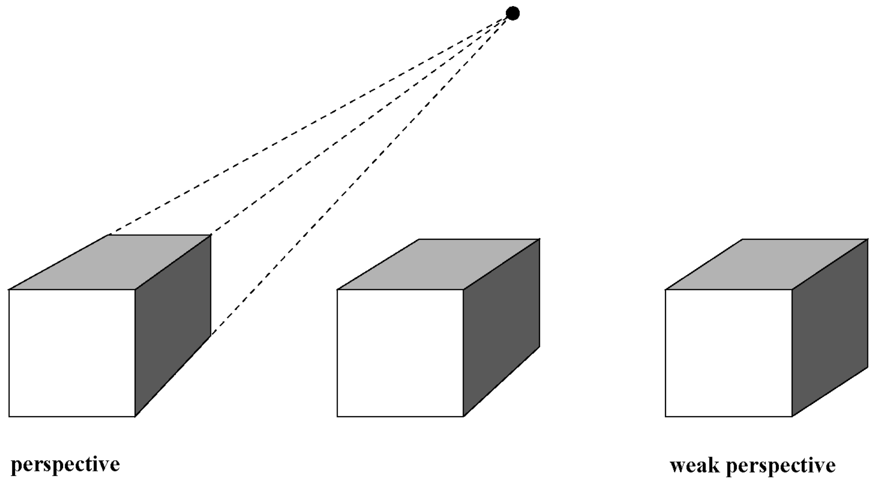
$$\mathbf{K} = \begin{bmatrix} \alpha_x & s & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{bmatrix}$$

with $s = \tan\theta$, and θ the angle between the image axes.

Simpler camera models

Weak Perspective

Track back, whilst zooming to keep image size fixed



The imaging rays become parallel, and the result is:

$$\mathtt{P} = \mathtt{K} \begin{bmatrix} r_{11} & r_{12} & r_{13} & * \\ r_{21} & r_{22} & r_{23} & * \\ 0 & 0 & 0 & * \end{bmatrix}$$

A generalization is the *affine camera*

The Affine Camera

$$\mathtt{P} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & t_1 \\ m_{21} & m_{22} & m_{23} & t_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

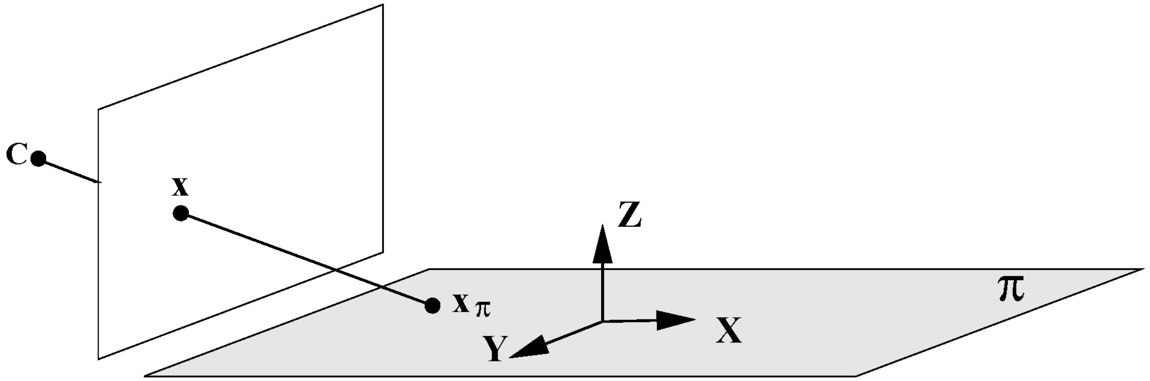
The matrix $\mathtt{M}_{2\times 3}$ has rank two.

Projection under an affine camera is a linear mapping on non-homogeneous coordinates composed with a translation:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \end{bmatrix} \begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ \mathrm{Z} \end{pmatrix} + \begin{pmatrix} t_1 \\ t_2 \end{pmatrix}$$

- The point $(t_1, t_2)^\top$ is the image of the world origin.
- The centre of the affine camera is at infinity.
- An affine camera has 8 degrees of freedom.
- It models weak-perspective and para-perspective.

Plane projective transformations



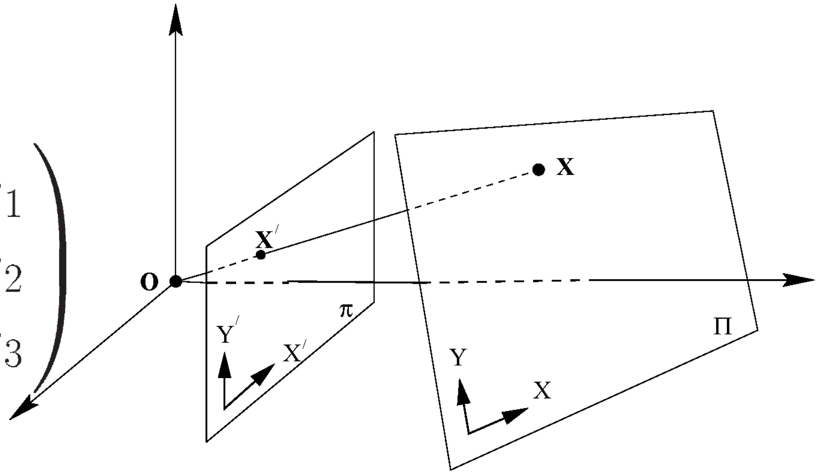
Choose the world coordinate system such that the plane of the points has zero $\mathtt{Z}$ coordinate. Then the 3×4 matrix $\mathtt{P}$ reduces to

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ 0 \\ 1 \end{pmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{14} \\ p_{21} & p_{22} & p_{24} \\ p_{31} & p_{32} & p_{34} \end{bmatrix} \begin{pmatrix} \mathrm{X} \\ \mathrm{Y} \\ 1 \end{pmatrix}$$

which is a 3×3 matrix representing a general plane to plane projective transformation.

Projective transformations continued

$$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$



or $\mathbf{x}' = \mathtt{H}\mathbf{x}$, where $\mathtt{H}$ is a 3×3 non-singular homogeneous matrix.

- This is the most general transformation between the world and image plane under imaging by a perspective camera.
- It is often only the 3×3 form of the matrix that is important in establishing properties of this transformation.
- A projective transformation is also called a "homography" and a "collineation".
- $\mathtt{H}$ has 8 degrees of freedom.

Four points define a projective transformation

Given n point correspondences $(x, y) \leftrightarrow (x', y')$

Compute $\mathtt{H}$ such that $\mathbf{x}'_i = \mathtt{H}\mathbf{x}_i$

- Each point correspondence gives two constraints

$$x' = \frac{x'_1}{x'_3} = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}}, \qquad y' = \frac{x'_2}{x'_3} = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}}$$

and multiplying out generates two linear equations for the elements of $\mathtt{H}$

$$x'\left(h_{31}x + h_{32}y + h_{33}\right) = h_{11}x + h_{12}y + h_{13}$$
$$y'\left(h_{31}x + h_{32}y + h_{33}\right) = h_{21}x + h_{22}y + h_{23}$$

- If $n \geq 4$ (no three points collinear), then $\mathtt{H}$ is determined uniquely.
- The converse of this is that it is possible to transform any four points in general position to any other four points in general position by a projectivity.

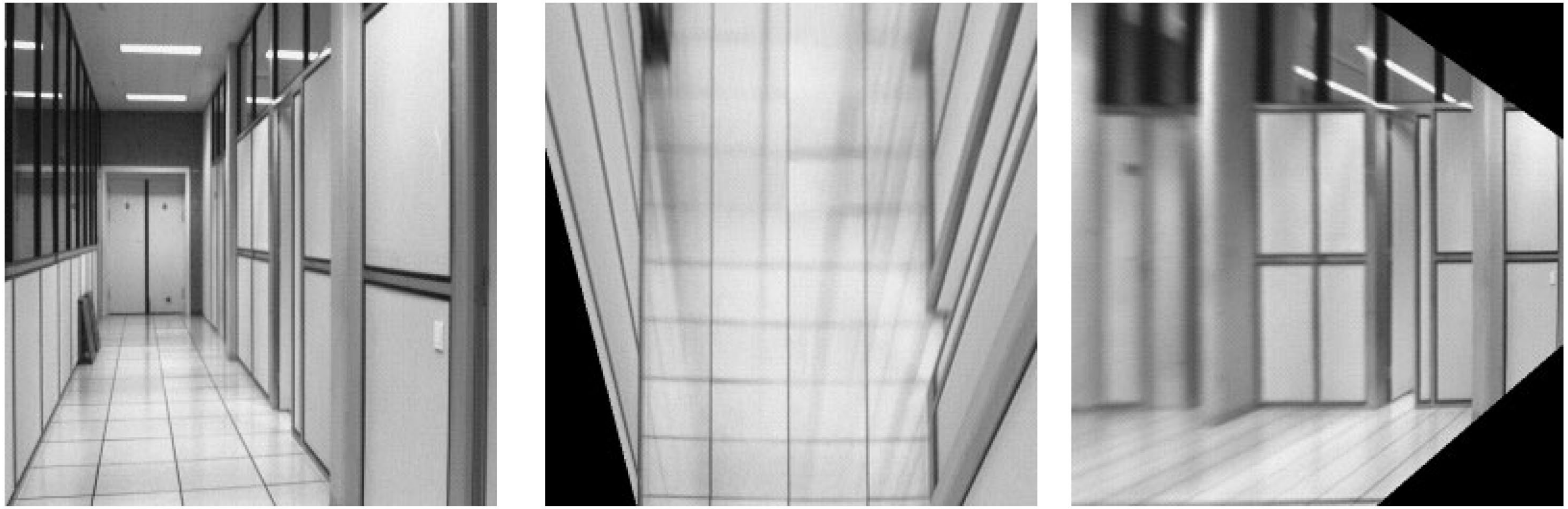
Example 1: Removing Perspective Distortion

Given: the coordinates of four points on the scene plane

Find: a projective rectification of the plane

- This rectification does not require knowledge of any of the camera's parameters or the pose of the plane.
- It is not always necessary to know coordinates for four points.

Example 2: Synthetic Rotations



The synthetic images are produced by projectively warping the original image so that four corners of an imaged rectangle map to the corners of a rectangle. Both warpings correspond to a synthetic rotation of the camera about the (fixed) camera centre.

Two View Geometry

- Cameras $\mathtt{P}$ and $\mathtt{P}'$ such that

$$\mathbf{x} = \mathtt{P}\mathbf{X} \qquad \mathbf{x}' = \mathtt{P}'\mathbf{X}$$

- Baseline between the cameras is non-zero.

Given an image point in the first view, where is the corresponding point in the second view?

What is the relative position of the cameras?

What is the 3D geometry of the scene?

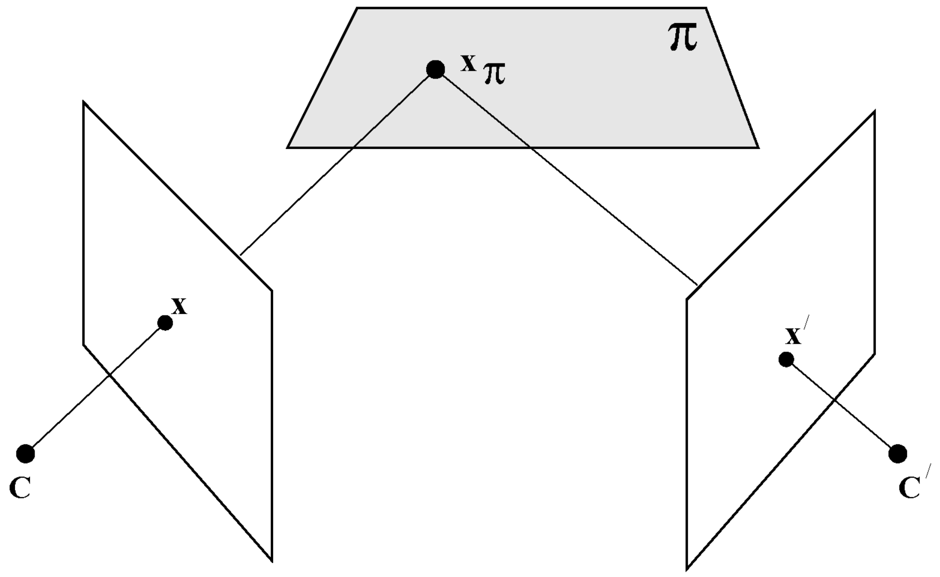
Images of Planes

Projective transformation between images induced by a plane

$$\mathbf{x} = \mathtt{H}_{1\pi}\mathbf{x}_\pi \qquad \mathbf{x}' = \mathtt{H}_{2\pi}\mathbf{x}_\pi$$

$$\mathbf{x}' = \mathtt{H}_{2\pi}\mathbf{x}_\pi$$

$$= \mathtt{H}_{2\pi}\mathtt{H}_{1\pi}^{-1}\mathbf{x} = \mathtt{H}\mathbf{x}$$



- $\mathtt{H}$ can be computed from the correspondence of four points on the plane.