Two View Geometry: Camera Rotation and Imaging a Plane

Computer Vision II

CSE 252B

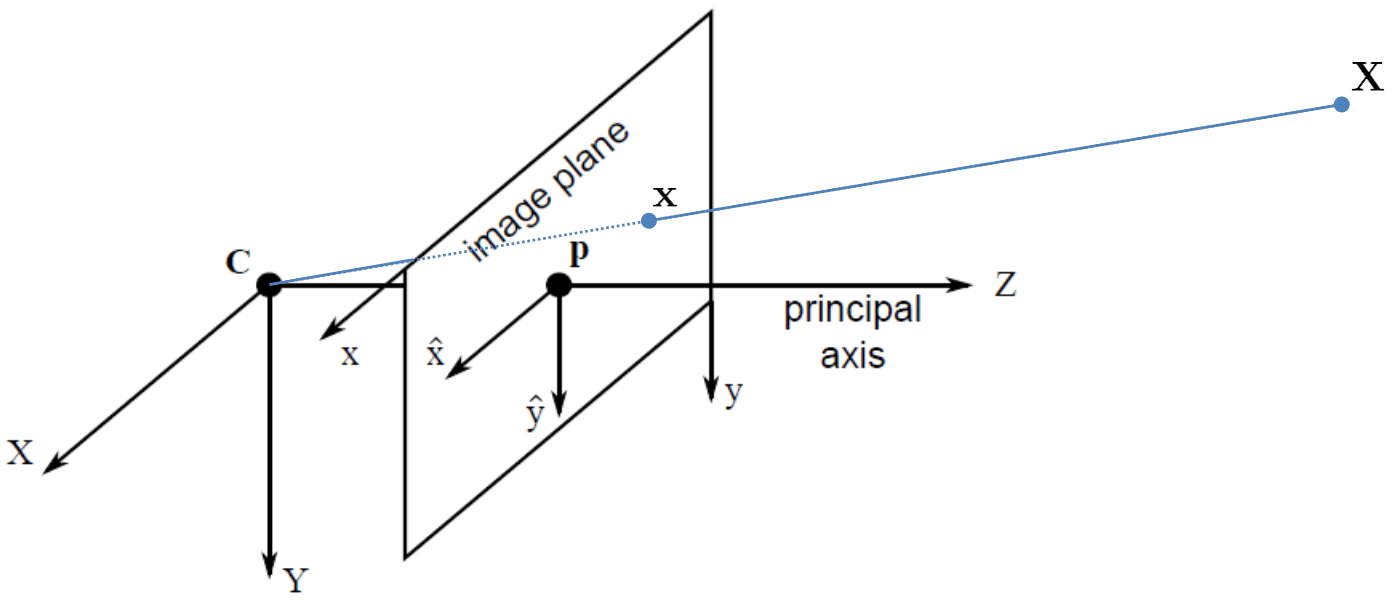
Lecture 10

Announcements

- Assignment 3 is due Feb 18, 11:59 PM
- Reading
  - Section 8.4
  - Section 13.1

Single view geometry

Camera projection



Camera projection matrix

- Projective camera

When camera projection matrix is known up to projective transformation
3×4 homogeneous (i.e., defined up to nonzero scale) matrix (11 degrees of freedom)

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

When parameters of camera projection matrix are known or known up to nonzero scale (11 degrees of freedom)

$$\mathtt{P} = \mathtt{K}[\mathtt{R} \,|\, \mathbf{t}]$$
$$\mathtt{P} = \mathtt{K}\mathtt{R}[\mathtt{I} \,|\, -\tilde{\mathbf{C}}]$$

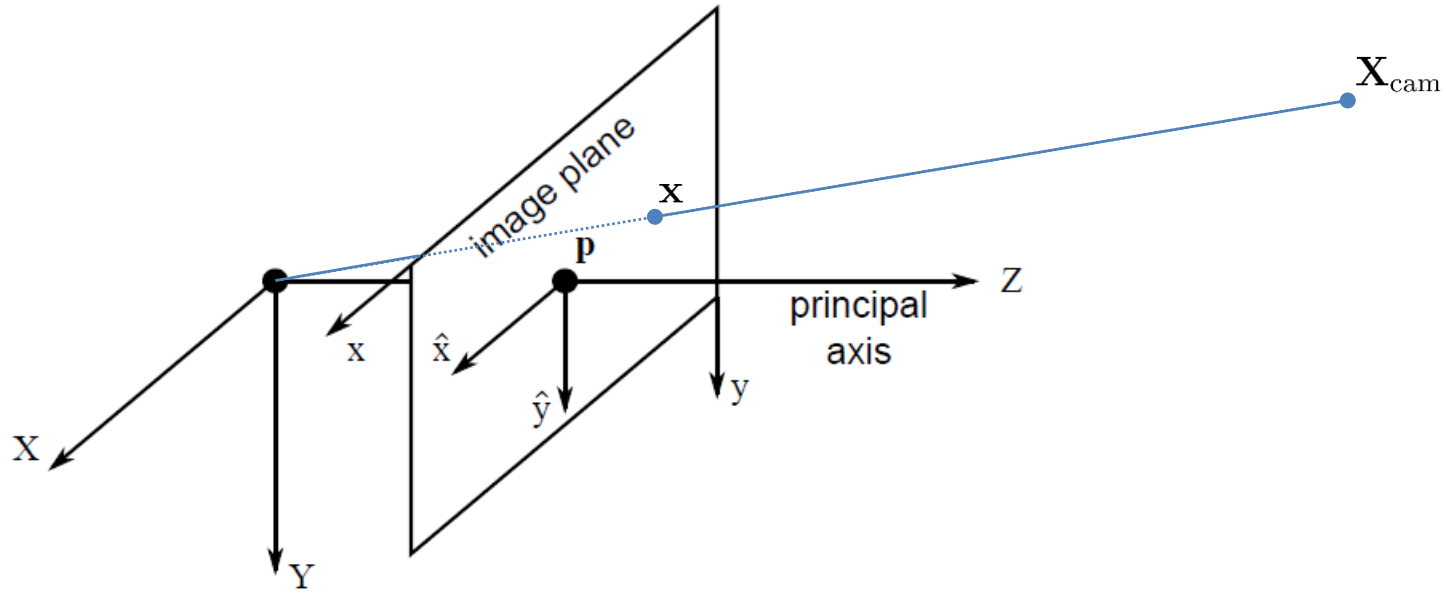
where

$\mathtt{K}$	Camera calibration matrix
$\mathtt{R}$	Camera rotation matrix
$\mathbf{t}$	Camera translation vector
$\mathbf{C}$	Camera center

Camera projection

Transform points from world coordinate frame to camera coordinate frame

$$\tilde{\mathbf{X}}_{\text{cam}} = \mathtt{R}\tilde{\mathbf{X}} + \mathbf{t}$$



Calibrated camera

$$\mathtt{K} = \begin{bmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

where

$\alpha_x = f m_x$	Focal length in x direction in terms of pixel dimensions
$\alpha_y = f m_y$	Focal length in y direction in terms of pixel dimensions
s	Skew
$(x_0, y_0)^\top = (p_x m_x, p_y m_y)^\top$	Coordinates of principal point in terms of pixel dimensions

where m_x and m_y are number of pixels per unit disance in x and y directions, respectively

- If camera calibration parameters are known, then use normalized camera projection matrix and image points in normalized coordinates

Normalized camera projection matrix

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

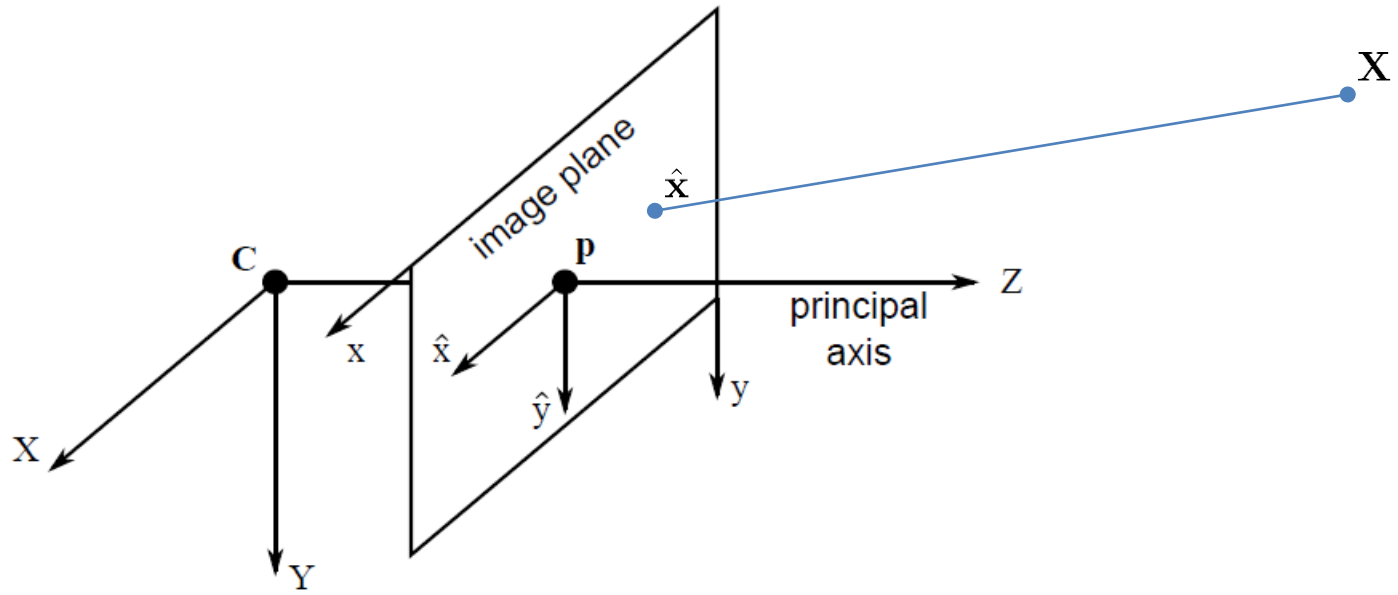
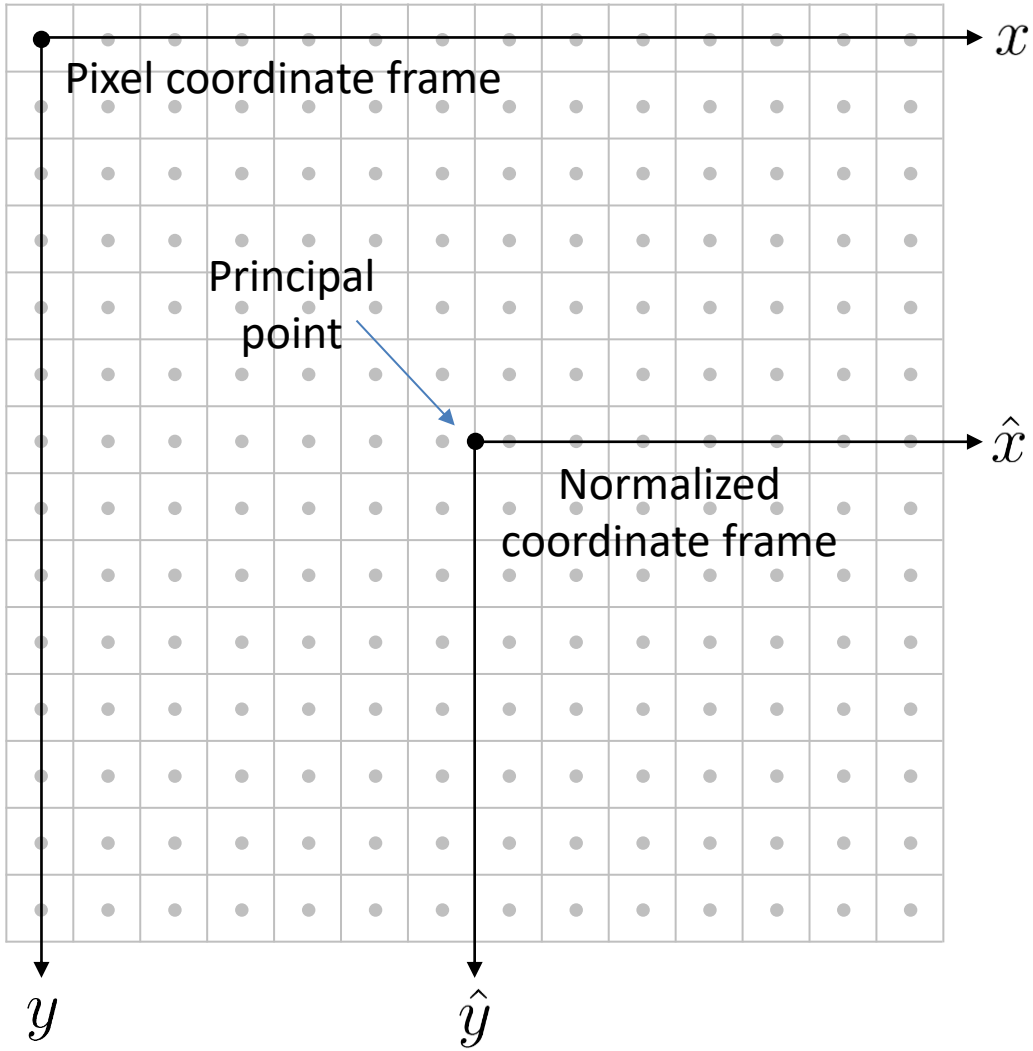


Image coordinate frames



Camera projection

Project 3D point in world coordinate frame $\mathbf{X}$ under camera projection matrix $\mathtt{P} = \mathtt{K}[\mathtt{R} \,|\, \mathbf{t}]$

$$\begin{aligned} \mathbf{x} &= \mathtt{P}\mathbf{X} \\ \mathbf{x} &= \mathtt{K}[\mathtt{R} \,|\, \mathbf{t}]\mathbf{X} \\ \mathtt{K}^{-1}\mathbf{x} &= [\mathtt{R} \,|\, \mathbf{t}]\mathbf{X} \\ \hat{\mathbf{x}} &= \hat{\mathtt{P}}\mathbf{X} \end{aligned}$$

Camera calibration matrix

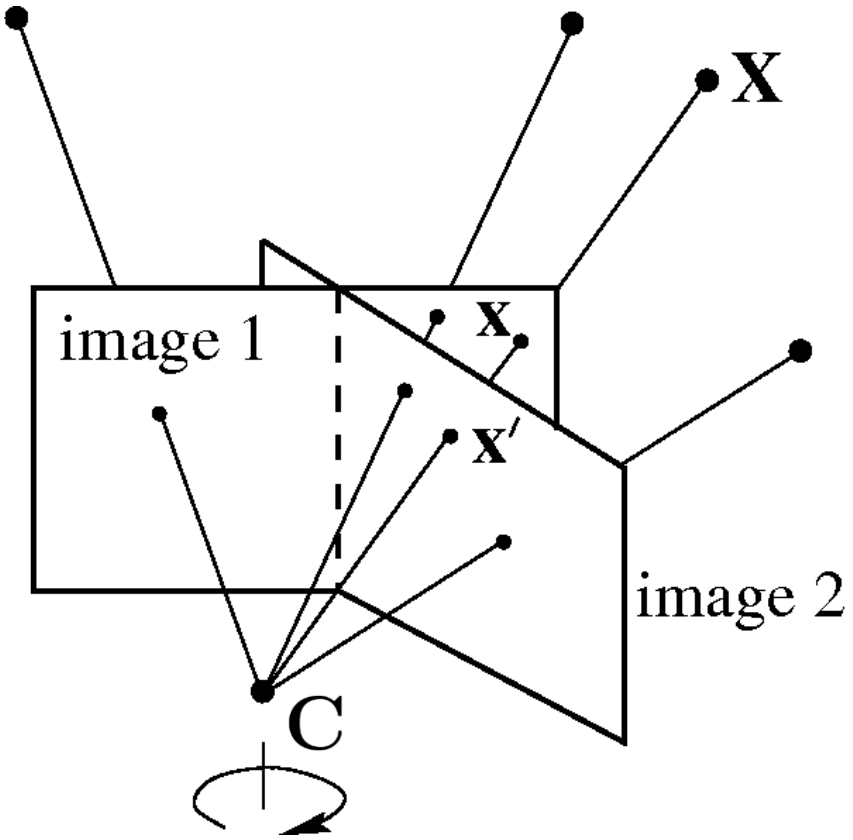
where

Camera projection matrix	$\mathtt{P} = \mathtt{K}[\mathtt{R} \,|\, \mathbf{t}]$	$= \mathtt{K}\hat{\mathtt{P}}$
Normalized camera projection matrix	$\hat{\mathtt{P}} = [\mathtt{R} \,|\, \mathbf{t}]$	$= \mathtt{K}^{-1}\mathtt{P}$
Image point in pixel coordinates	$\mathbf{x} = \mathtt{K}[\mathtt{R} \,|\, \mathbf{t}]\mathbf{X}$	$= \mathtt{K}\hat{\mathbf{x}}$
Image point in normalized coordinates	$\hat{\mathbf{x}} = [\mathtt{R} \,|\, \mathbf{t}]\mathbf{X}$	$= \mathtt{K}^{-1}\mathbf{x}$

Two view geometry

Two view geometry

- Rotation about the same camera center



Two view geometry

- Rotation about the same camera center
  - Second camera translation vector

The translation $\mathbf{t}'$ of the second camera (or view) rotating about the same camera center $\tilde{\mathbf{C}}$ of the first camera (or view) is given by

$$\tilde{\mathbf{C}} = -\mathtt{R}^\top \mathbf{t} = -\mathtt{R}'^\top \mathbf{t}'$$
$$\mathtt{R}'\mathtt{R}^\top \mathbf{t} = \mathbf{t}'$$

where

$\mathtt{R}$	First camera rotation matrix
$\mathbf{t}$	First camera translation vector
$\mathtt{R}'$	Second camera rotation matrix
$\mathbf{t}'$	Second camera translation vector

Calibrated cameras

Camera calibration matrix

$$\mathtt{K} = \begin{bmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

where

$\alpha_x = f m_x$	Focal length in x direction in terms of pixel dimensions
$\alpha_y = f m_y$	Focal length in y direction in terms of pixel dimensions
s	Skew
$(x_0, y_0)^\top = (p_x m_x, p_y m_y)^\top$	Coordinates of principal point in terms of pixel dimensions

where m_x and m_y are number of pixels per unit disance in x and y directions, respectively

(and lens distortion parameters)

Rotation about the same camera center

Given two cameras, the first camera and the second camera, rotating about the same camera center $\tilde{\mathbf{C}}$, a point $\hat{\mathbf{x}}$ in normalized coordinates in the first image maps to a point $\hat{\mathbf{x}}'$ in normalized coordinates in the second image as follows.

$$
\begin{aligned}
\mathbf{x} &= \mathtt{K}\mathtt{R}[\mathtt{I} \mid -\tilde{\mathbf{C}}]\mathbf{X} \\
\mathtt{K}^{-1}\mathbf{x} &= \mathtt{R}[\mathtt{I} \mid -\tilde{\mathbf{C}}]\mathbf{X} \\
\hat{\mathbf{x}} &= \mathtt{R}[\mathtt{I} \mid -\tilde{\mathbf{C}}]\mathbf{X}, \text{ where } \hat{\mathbf{x}} = \mathtt{K}^{-1}\mathbf{x} \\
\mathtt{R}^{\top}\hat{\mathbf{x}} &= [\mathtt{I} \mid -\tilde{\mathbf{C}}]\mathbf{X}
\end{aligned}
\qquad
\begin{aligned}
\mathbf{x}' &= \mathtt{K}'\mathtt{R}'[\mathtt{I} \mid -\tilde{\mathbf{C}}]\mathbf{X} \\
\mathtt{K}'^{-1}\mathbf{x}' &= \mathtt{R}'[\mathtt{I} \mid -\tilde{\mathbf{C}}]\mathbf{X} \\
\hat{\mathbf{x}}' &= \mathtt{R}'[\mathtt{I} \mid -\tilde{\mathbf{C}}]\mathbf{X}, \text{ where } \hat{\mathbf{x}}' = \mathtt{K}'^{-1}\mathbf{x}' \\
\hat{\mathbf{x}}' &= \mathtt{R}'\mathtt{R}^{\top}\hat{\mathbf{x}}
\end{aligned}
$$

where

$\mathtt{K}$	First camera calibration matrix
$\mathtt{R}$	First camera rotation matrix
$\mathbf{t}$	First camera translation vector
$\mathtt{K}'$	Second camera calibration matrix
$\mathtt{R}'$	Second camera rotation matrix
$\mathbf{t}'$	Second camera translation vector

Rotation about the same camera center

- Case of unrotated first camera

If $\mathtt{R} = \mathtt{I}$ (i.e., the first camera is unrotated), then

$$\hat{\mathbf{x}} = [\mathtt{I} \mid -\tilde{\mathbf{C}}]\mathbf{X} \qquad \hat{\mathbf{x}}' = \mathtt{R}'[\mathtt{I} \mid -\tilde{\mathbf{C}}]\mathbf{X}$$
$$\hat{\mathbf{x}}' = \mathtt{R}'\hat{\mathbf{x}}$$

which is often written as $\hat{\mathbf{x}}' = \mathtt{R}\hat{\mathbf{x}}$, where $\mathtt{R}$ is the rotation of the second camera
Mapping in pixel coordinates

$$\hat{\mathbf{x}}' = \mathtt{R}\hat{\mathbf{x}}, \text{ where } \hat{\mathbf{x}} = \mathtt{K}^{-1}\mathbf{x} \text{ and } \hat{\mathbf{x}}' = \mathtt{K}'^{-1}\mathbf{x}'$$
$$\mathtt{K}'^{-1}\mathbf{x}' = \mathtt{R}\mathtt{K}^{-1}\mathbf{x}$$
$$\mathbf{x}' = \mathtt{K}'\mathtt{R}\mathtt{K}^{-1}\mathbf{x}$$
$$\mathbf{x}' = \mathtt{H}\mathbf{x}, \text{ where } \mathtt{H} = \mathtt{K}'\mathtt{R}\mathtt{K}^{-1}$$

Backprojection of image points

Backprojection of image point in normalized coordinates to ray direction in camera coordinate frame

$$\hat{\mathbf{x}}_i = [\mathtt{R} \,|\, \mathbf{t}] \begin{bmatrix} \tilde{\mathbf{X}}_i \\ 1 \end{bmatrix}$$

$$\hat{\mathbf{x}}_i = \mathtt{R}\tilde{\mathbf{X}}_i + \mathbf{t}$$

$$\hat{\mathbf{x}}_i = \tilde{\mathbf{X}}_{\mathrm{cam},i}, \text{ where } \tilde{\mathbf{X}}_{\mathrm{cam},i} = \mathtt{R}\tilde{\mathbf{X}}_i + \mathbf{t}$$

$$\mathbf{d}_i = \tilde{\mathbf{X}}_{\mathrm{cam},i} \textbf{ up to nonzero scale}, \text{ where } \mathbf{d}_i = \hat{\mathbf{x}}_i$$

Then, unitize $\mathbf{d}_i = (d_{i,1}, d_{i,2}, d_{i,3})^\top$ and ensure $d_{i,3}$ is positive (i.e., in front of camera)

$$\mathbf{d}_i = \frac{1}{\mathrm{sign}(d_{i,3})\|\mathbf{d}_i\|}\mathbf{d}_i$$

Estimation of rotation matrix

Given the image point correspondences $\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}'_i$ in normalized coordinates, estimate the rotation matrix $\mathtt{R}$, where $\hat{\mathbf{x}}'_i = \mathtt{R}\hat{\mathbf{x}}_i \; \forall \, i$

1. Backprojection the points $\hat{\mathbf{x}}_i$ in image 1 to unit direction vectors $\mathbf{d}_i$ and backprojection the points $\hat{\mathbf{x}}'_i$ in image 2 to unit direction vectors $\mathbf{d}'_i$ (i.e., $\mathbf{d}_i \leftrightarrow \mathbf{d}'_i$)

2. Compute the matrix

$$\mathtt{S} = \sum_i \mathbf{d}'_i \mathbf{d}_i^\top$$

   and its singular value decomposition $\mathtt{S} = \mathtt{U}\Sigma\mathtt{V}^\top$

3. The rotation matrix

$$\mathtt{R} = \begin{cases} \mathtt{U}\,\mathrm{diag}(1, 1, -1)\mathtt{V}^\top & \text{if } \det(\mathtt{U})\det(\mathtt{V}) < 0 \\ \mathtt{U}\mathtt{V}^\top & \text{otherwise} \end{cases}$$

   such that $\det(\mathtt{R}) = +1$

Uncalibrated cameras

Rotation about the same camera center

Given two cameras, the first camera and the second camera, rotating about the same camera center $\tilde{\mathbf{C}}$, points $\mathbf{x}_i$ in pixel coordinates in the first image map to points $\mathbf{x}_i$ in pixel coordinates in the second image as follows.

$$\mathbf{x}_i = \mathtt{KR}[\mathtt{I} \mid -\tilde{\mathbf{C}}]\mathbf{X}_i \,\forall\, i \qquad \mathbf{x}'_i = \mathtt{K}'\mathtt{R}'[\mathtt{I} \mid -\tilde{\mathbf{C}}]\mathbf{X}_i \,\forall\, i$$

$$(\mathtt{KR})^{-1}\mathbf{x}_i = [\mathtt{I} \mid -\tilde{\mathbf{C}}]\mathbf{X}_i \,\forall\, i \qquad \mathbf{x}'_i = \mathtt{K}'\mathtt{R}'(\mathtt{KR})^{-1}\mathbf{x}_i \,\forall\, i$$

$$\mathbf{x}'_i = \mathtt{H}\mathbf{x}_i \,\forall\, i, \text{ where } \mathtt{H} = \mathtt{K}'\mathtt{R}'(\mathtt{KR})^{-1}$$

Rotation about the same camera center

- Case of unrotated first camera

If $\mathtt{R} = \mathtt{I}$ (i.e., the first camera is unrotated), then

$$\begin{aligned} \mathbf{x}_i &= \mathtt{K}[\mathtt{I} \mid -\tilde{\mathbf{C}}]\mathbf{X}_i \,\forall\, i & \mathbf{x}'_i &= \mathtt{K}'\mathtt{R}'[\mathtt{I} \mid -\tilde{\mathbf{C}}]\mathbf{X}_i \,\forall\, i \\ \mathtt{K}^{-1}\mathbf{x}_i &= [\mathtt{I} \mid -\tilde{\mathbf{C}}]\mathbf{X}_i \,\forall\, i & \mathbf{x}'_i &= \mathtt{K}'\mathtt{R}'\mathtt{K}^{-1}\mathbf{x}_i \,\forall\, i \\ & & \mathbf{x}'_i &= \mathtt{H}\mathbf{x}_i \,\forall\, i, \text{ where } \mathtt{H} = \mathtt{K}'\mathtt{R}'\mathtt{K}^{-1} \end{aligned}$$

which is often written as $\mathtt{H} = \mathtt{K}'\mathtt{R}\mathtt{K}^{-1}$, where $\mathtt{R}$ is the rotation of the second camera

Note if $\mathtt{H}$ is determined from point correspondences $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$ in pixel coordinates, then

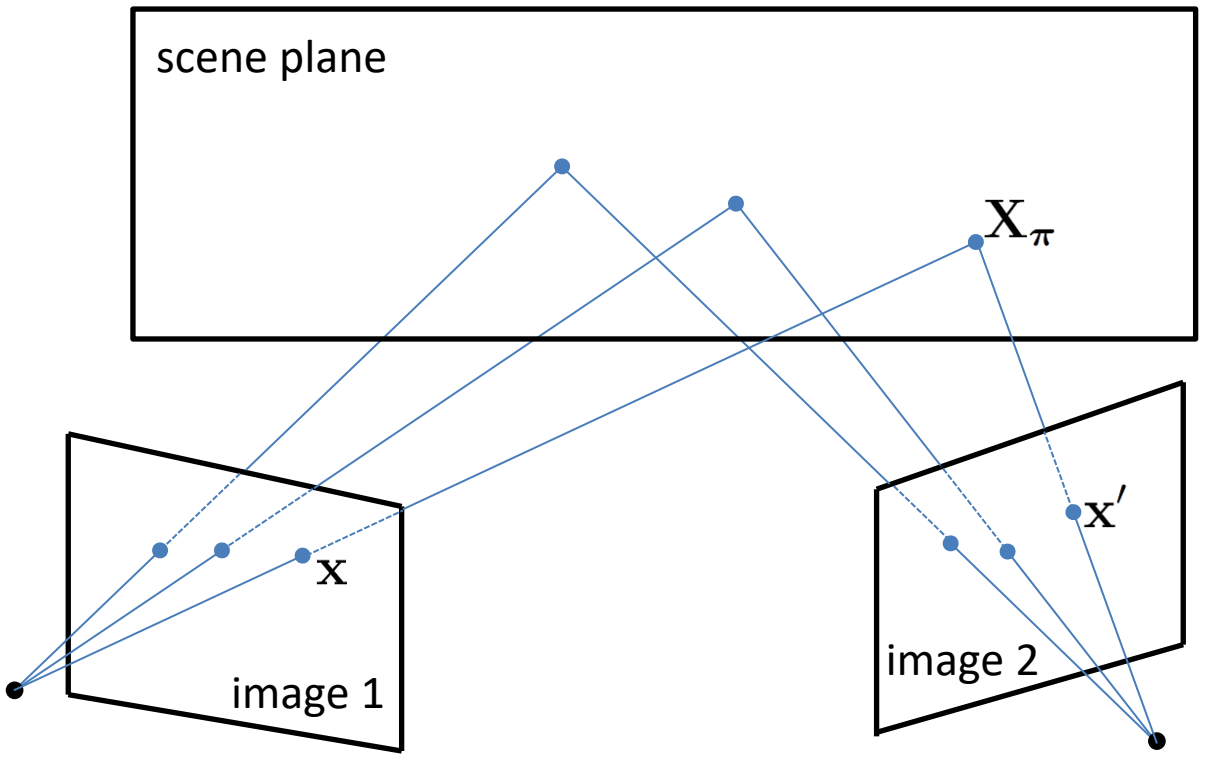
$$\begin{aligned} \mathtt{H} &= \mathtt{K}'\mathtt{R}\mathtt{K}^{-1} \\ \mathtt{R} &= \mathtt{K}'^{-1}\mathtt{H}\mathtt{K} \end{aligned}$$

In general, the resulting $\mathtt{R}$ will not be in $\mathrm{SO}(3)$. If the camera calibration matrices are known, then determine $\mathtt{R}$ directly from $\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}'_i$ in normalized coordinates.

Imaging a plane

Two view geometry

- Imaging a plane



Imaging a plane

- Calibrated cameras

The cameras $\mathtt{P} = \mathtt{K}[\mathtt{R} \,|\, \mathbf{t}]$ and $\mathtt{P}' = \mathtt{K}'[\mathtt{R}' \,|\, \mathbf{t}']$ image the points $\mathbf{X}_{\boldsymbol{\pi},i}$ on the plane $\boldsymbol{\pi} = (\mathbf{n}^\top, d)^\top$ as

$$\mathbf{x}_i = \mathtt{K}[\mathtt{R} \,|\, \mathbf{t}]\mathbf{X}_{\boldsymbol{\pi},i} \,\forall\, i \text{ and } \mathbf{x}'_i = \mathtt{K}'[\mathtt{R}' \,|\, \mathbf{t}']\mathbf{X}_{\boldsymbol{\pi},i} \,\forall\, i$$

Transform cameras and plane such that they are relative to the first camera coordinate frame

$$\mathtt{H}_\mathrm{E} = \begin{bmatrix} \mathtt{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix}$$

$$\hat{\mathtt{P}}\mathtt{H}_\mathrm{E}^{-1} = [\mathtt{R} \,|\, \mathbf{t}] \begin{bmatrix} \mathtt{R}^\top & -\mathtt{R}^\top\mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} = [\mathtt{R}\mathtt{R}^\top \,|\, -\mathtt{R}\mathtt{R}^\top\mathbf{t} + \mathbf{t}] = [\mathtt{I} \,|\, \mathbf{0}]$$

$$\hat{\mathtt{P}}'\mathtt{H}_\mathrm{E}^{-1} = [\mathtt{R}' \,|\, \mathbf{t}'] \begin{bmatrix} \mathtt{R}^\top & -\mathtt{R}^\top\mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} = [\mathtt{R}'\mathtt{R}^\top \,|\, -\mathtt{R}'\mathtt{R}^\top\mathbf{t} + \mathbf{t}'] = [\mathtt{R}'\mathtt{R}^\top \,|\, \mathbf{t}' - \mathtt{R}'\mathtt{R}^\top\mathbf{t}]$$

$$\mathtt{H}_\mathrm{E}^{-\top}\boldsymbol{\pi} = \begin{bmatrix} \mathtt{R} & \mathbf{0} \\ -\mathbf{t}^\top\mathtt{R} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{n} \\ d \end{bmatrix} = \begin{bmatrix} \mathtt{R}\mathbf{n} \\ -\mathbf{t}^\top\mathtt{R}\mathbf{n} + d \end{bmatrix} = \begin{bmatrix} \mathtt{R}\mathbf{n} \\ d - \mathbf{t}^\top\mathtt{R}\mathbf{n} \end{bmatrix}$$

Use the transformed cameras and plane

$$\hat{\mathtt{P}} = [\mathtt{I} \,|\, \mathbf{0}], \hat{\mathtt{P}}' = [\mathtt{R}'\mathtt{R}^\top \,|\, \mathbf{t}' - \mathtt{R}'\mathtt{R}^\top\mathbf{t}], \text{ and } \boldsymbol{\pi} = \begin{bmatrix} \mathtt{R}\mathbf{n} \\ d - \mathbf{t}^\top\mathtt{R}\mathbf{n} \end{bmatrix}$$

Imaging a plane

- Calibrated cameras

Backproject 2D image point $\hat{\mathbf{x}}$ to 3D ray

$$\hat{\mathbf{x}} = [\mathtt{I} \,|\, \mathbf{0}]\mathbf{X}$$
$$\hat{\mathbf{x}} = [\mathtt{I} \,|\, \mathbf{0}] \begin{bmatrix} \tilde{\mathbf{X}} \\ 1 \end{bmatrix}$$
$$\hat{\mathbf{x}} = \tilde{\mathbf{X}} \text{ up to nonzero scale}$$

Intersect ray with plane. Point $\mathbf{X}_{\boldsymbol{\pi}}$ is on ray and plane.

$$\boldsymbol{\pi}^\top \mathbf{X}_{\boldsymbol{\pi}} = 0$$
$$[\mathbf{n}^\top \mathtt{R}^\top \,|\, d - \mathbf{t}^\top \mathtt{R}\mathbf{n}] \begin{bmatrix} \hat{\mathbf{x}} \\ \rho \end{bmatrix} = 0, \text{ where } \quad \mathbf{X}_{\boldsymbol{\pi}} = (\hat{\mathbf{x}}^\top, \rho)^\top, \text{ solve for } \rho$$
$$\mathbf{n}^\top \mathtt{R}^\top \hat{\mathbf{x}} + \rho(d - \mathbf{t}^\top \mathtt{R}\mathbf{n}) = 0$$
$$\rho(d - \mathbf{t}^\top \mathtt{R}\mathbf{n}) = -\mathbf{n}^\top \mathtt{R}^\top \hat{\mathbf{x}}$$
$$\rho = -\frac{\mathbf{n}^\top \mathtt{R}^\top \hat{\mathbf{x}}}{d - \mathbf{t}^\top \mathtt{R}\mathbf{n}}$$
$$\mathbf{X}_{\boldsymbol{\pi}} = \begin{bmatrix} \hat{\mathbf{x}} \\ -\frac{\mathbf{n}^\top \mathtt{R}^\top \hat{\mathbf{x}}}{d - \mathbf{t}^\top \mathtt{R}\mathbf{n}} \end{bmatrix}$$

Imaging a plane

- Calibrated cameras

Project 3D point on plane to image point $\hat{\mathbf{x}}'$

$$\hat{\mathbf{x}}' = \hat{\mathtt{P}}'\mathbf{X}_{\pi}$$

$$\hat{\mathbf{x}}' = [\mathtt{R}'\mathtt{R}^{\top} \,|\, \mathbf{t}' - \mathtt{R}'\mathtt{R}^{\top}\mathbf{t}] \begin{bmatrix} \hat{\mathbf{x}} \\ -\frac{\mathbf{n}^{\top}\mathtt{R}^{\top}\hat{\mathbf{x}}}{d - \mathbf{t}^{\top}\mathtt{R}\mathbf{n}} \end{bmatrix}$$

$$\hat{\mathbf{x}}' = \mathtt{R}'\mathtt{R}^{\top}\hat{\mathbf{x}} - (\mathbf{t}' - \mathtt{R}'\mathtt{R}^{\top}\mathbf{t})\frac{\mathbf{n}^{\top}\mathtt{R}^{\top}\hat{\mathbf{x}}}{d - \mathbf{t}^{\top}\mathtt{R}\mathbf{n}}$$

$$\hat{\mathbf{x}}' = \left(\mathtt{R}' + (\mathtt{R}'\mathtt{R}^{\top}\mathbf{t} - \mathbf{t}')\frac{\mathbf{n}^{\top}}{d - \mathbf{t}^{\top}\mathtt{R}\mathbf{n}}\right)\mathtt{R}^{\top}\hat{\mathbf{x}}$$

$$\hat{\mathbf{x}}' = \hat{\mathtt{H}}\hat{\mathbf{x}}, \text{ where } \hat{\mathtt{H}} = \left(\mathtt{R}' + (\mathtt{R}'\mathtt{R}^{\top}\mathbf{t} - \mathbf{t}')\frac{\mathbf{n}^{\top}}{d - \mathbf{t}^{\top}\mathtt{R}\mathbf{n}}\right)\mathtt{R}^{\top}$$

2D projective transformation

$$\hat{\mathbf{x}}'_i = \hat{\mathtt{H}}\hat{\mathbf{x}}_i \,\forall\, i$$

$$\mathtt{K}'^{-1}\mathbf{x}'_i = \hat{\mathtt{H}}\mathtt{K}^{-1}\mathbf{x}_i \,\forall\, i$$

$$\mathbf{x}'_i = \mathtt{K}'\hat{\mathtt{H}}\mathtt{K}^{-1}\mathbf{x}_i \,\forall\, i$$

$$\mathbf{x}'_i = \mathtt{H}\mathbf{x}_i \,\forall\, i, \text{ where } \mathtt{H} = \mathtt{K}'\hat{\mathtt{H}}\mathtt{K}^{-1}$$

Imaging a plane

- Uncalibrated cameras

The points $\mathbf{X}_{\pi,i}$ on the plane $\boldsymbol{\pi}$ may be written as

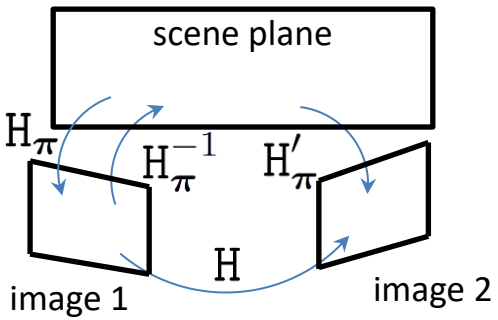
$$\mathbf{X}_{\pi,i} = \mathtt{M}\mathbf{x}_{\pi,i} \,\forall\, i, \text{ where } \mathtt{M} \text{ is the null space of } \boldsymbol{\pi}^\top$$

Projection

$$\begin{aligned} \mathbf{x}_i &= \mathtt{P}\mathbf{X}_{\pi,i} \,\forall\, i & \mathbf{x}'_i &= \mathtt{P}'\mathbf{X}_{\pi,i} \,\forall\, i \\ \mathbf{x}_i &= \mathtt{PM}\mathbf{x}_{\pi,i} \,\forall\, i & \mathbf{x}'_i &= \mathtt{P}'\mathtt{M}\mathbf{x}_{\pi,i} \,\forall\, i \\ \mathbf{x}_i &= \mathtt{H}_{\boldsymbol{\pi}}\mathbf{x}_{\pi,i} \,\forall\, i, \text{ where } \mathtt{H}_{\boldsymbol{\pi}} = \mathtt{PM} & \mathbf{x}'_i &= \mathtt{H}'_{\boldsymbol{\pi}}\mathbf{x}_{\pi,i} \,\forall\, i, \text{ where } \mathtt{H}'_{\boldsymbol{\pi}} = \mathtt{P}'\mathtt{M} \end{aligned}$$

2D projective transformation

$$\begin{aligned} \mathbf{x}'_i = \mathtt{H}\mathbf{x}_i \,\forall\, i, \text{ where } \mathtt{H} &= \mathtt{H}'_{\boldsymbol{\pi}}\mathtt{H}_{\boldsymbol{\pi}}^{-1} \\ \mathtt{H} &= \mathtt{P}'\mathtt{M}(\mathtt{PM})^{-1} \end{aligned}$$



Two view geometry

- Same model for rotation about the same camera center and imaging a plane
  - 2D projective transformation

2D projective transformation

2D geometric transformations

- Projective

3×3 homogeneous (i.e., defined up to nonzero scale) matrix (8 degrees of freedom)

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_1 \\ a_{21} & a_{22} & t_2 \\ v_1 & v_2 & u \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

$$\mathbf{x}' = \begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{v}^\top & u \end{bmatrix} \mathbf{x}, \text{ where } \mathtt{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \mathbf{t} = (t_1, t_2)^\top, \text{ and } \mathbf{v} = (v_1, v_2)^\top$$

$$\mathbf{x}' = \mathtt{H}_\mathrm{P}\mathbf{x}, \text{ where } \mathtt{H}_\mathrm{P} = \begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{v}^\top & u \end{bmatrix}$$

2D geometric transformations

Linear transformation of points in homogeneous coordinates

$$\mathbf{x}' = \mathtt{H}\mathbf{x}$$

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} xh_{11} + yh_{12} + wh_{13} \\ xh_{21} + yh_{22} + wh_{23} \\ xh_{31} + yh_{32} + wh_{33} \end{bmatrix}$$

Nonlinear transformation of points in inhomogeneous coordinates

$$\tilde{x} = \frac{x'}{w'} = \frac{xh_{11} + yh_{12} + wh_{13}}{xh_{31} + yh_{32} + wh_{33}} \text{ and}$$

$$\tilde{y} = \frac{y'}{w'} = \frac{xh_{21} + yh_{22} + wh_{23}}{xh_{31} + yh_{32} + wh_{33}}$$

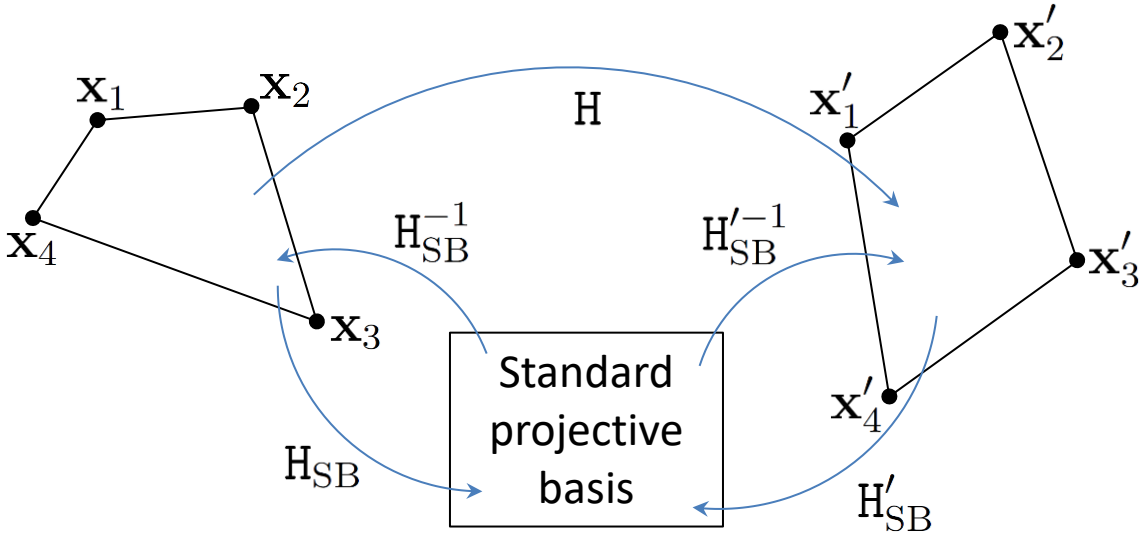
Do not forget this is a nonlinear transformation

2D projective transformation, minimum solution

- Transformation from 4 points in image 1 to 4 points in image 2

The minimum number of point correspondences $\hat{\mathbf{x}}_i \leftrightarrow \tilde{\mathbf{x}}'_i$ needed to exactly solve for the 2D projective transformation (8 parameters) is 4 (i.e., 8 measurements).

$$\mathbf{x}'_i = \mathtt{H}'^{-1}_{\mathrm{SB}} \mathtt{H}_{\mathrm{SB}} \mathbf{x}_i \text{ for } i = 1, 2, 3, 4$$
$$\mathbf{x}'_i = \mathtt{H} \mathbf{x}_i \text{ for } i = 1, 2, 3, 4 \text{ where } \mathtt{H} = \mathtt{H}'^{-1}_{\mathrm{SB}} \mathtt{H}_{\mathrm{SB}}$$



2D projective transformation, 4 points to standard basis

Standard basis of $\mathbb{P}^2$

Points at infinity along each axis — Origin — "Unit point"

$$\mathbf{e}_1 = \begin{bmatrix}1\\0\\0\end{bmatrix}, \mathbf{e}_2 = \begin{bmatrix}0\\1\\0\end{bmatrix}, \mathbf{e}_3 = \begin{bmatrix}0\\0\\1\end{bmatrix}, \text{ and } \mathbf{e}_4 = \begin{bmatrix}1\\1\\1\end{bmatrix}$$

4 points can be mapped to standard basis

$$\begin{bmatrix}\mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 & \mathbf{e}_4\end{bmatrix} = \mathtt{H}_{\mathrm{SB}} \begin{bmatrix}s_1\mathbf{x}_1 & s_2\mathbf{x}_2 & s_3\mathbf{x}_3 & s_4\mathbf{x}_4\end{bmatrix}$$

$$\begin{bmatrix}\mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 & \mathbf{e}_4\end{bmatrix} = \mathtt{H}_{\mathrm{SB}} \begin{bmatrix}\lambda_1\mathbf{x}_1 & \lambda_2\mathbf{x}_2 & \lambda_3\mathbf{x}_3 & \mathbf{x}_4\end{bmatrix}, \text{ where } \lambda_1 = \frac{s_1}{s_4}, \lambda_2 = \frac{s_2}{s_4}, \text{ and } \lambda_3 = \frac{s_3}{s_4}$$

$$\begin{bmatrix}\mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3\end{bmatrix} = \mathtt{H}_{\mathrm{SB}} \begin{bmatrix}\lambda_1\mathbf{x}_1 & \lambda_2\mathbf{x}_2 & \lambda_3\mathbf{x}_3\end{bmatrix}$$

$$\mathtt{I} = \mathtt{H}_{\mathrm{SB}} \begin{bmatrix}\lambda_1\mathbf{x}_1 & \lambda_2\mathbf{x}_2 & \lambda_3\mathbf{x}_3\end{bmatrix}$$

$$\mathtt{H}_{\mathrm{SB}}^{-1} = \begin{bmatrix}\lambda_1\mathbf{x}_1 & \lambda_2\mathbf{x}_2 & \lambda_3\mathbf{x}_3\end{bmatrix}$$

$$\mathbf{e}_4 = \mathtt{H}_{\mathrm{SB}}\mathbf{x}_4$$

$$\mathtt{H}_{\mathrm{SB}}^{-1}\mathbf{e}_4 = \mathbf{x}_4$$

$$\begin{bmatrix}\lambda_1\mathbf{x}_1 & \lambda_2\mathbf{x}_2 & \lambda_3\mathbf{x}_3\end{bmatrix}\mathbf{e}_4 = \mathbf{x}_4$$

$$\begin{bmatrix}\mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3\end{bmatrix}\boldsymbol{\lambda} = \mathbf{x}_4, \text{ solve for } \boldsymbol{\lambda} = (\lambda_1, \lambda_2, \lambda_3)^\top$$

then

$$\mathtt{H}_{\mathrm{SB}}^{-1} = \begin{bmatrix}\lambda_1\mathbf{x}_1 & \lambda_2\mathbf{x}_2 & \lambda_3\mathbf{x}_3\end{bmatrix}$$

Outlier rejection

Outlier rejection

- Even the presence of a single outlier may result in an inaccurate estimate
- Linear estimate
  - Minimizes sum of squared error and, in general, outliers have large error relative to inliers
  - Resulting estimate may not be near global minimum
- Nonlinear estimate
  - Cost function minimizes sum of squared error, so may not converge to accurate solution
    - Alternatively, use a robust cost function

Random sample consensus (RANSAC) and M-estimator sample consensus (MSAC)

Objective is to determine the consensus set with the minimum cost

tol is the tolerance for establishing datum/model compatibility
τ_{cost} is the upper bound on the cost of an acceptable consensus set
maxTrials is the maximum number of attempts to find a consensus set

$\text{consensus}_{\text{cost}} = \infty$
for (trials $= 0$; trials $<$ maxTrials && $\text{consensus}_{\text{cost}} > \tau_{\text{cost}}$; ++trials)
    Select a random sample of unique data points
    Calculate the model using the random sample
    Calculate the error for each data point using model
    Calculate the cost
    if cost $< \text{consensus}_{\text{cost}}$
        $\text{consensus}_{\text{cost}} = \text{cost}$
        $\text{consensus}_{\text{model}} = \text{model}$
Calculate the error for each data point using $\text{consensus}_{\text{model}}$
Calculate the set of inliers (i.e., data points with error $\leq$ tol)

Outlier rejection

- RANSAC cost

count is the total number of data points
tol is the tolerance for establishing datum/model compatibility

```
cost = 0
for (n = 0; n < count; ++n)
    cost += error[n] ≤ tol ? 0 : 1
```

- MSAC cost

count is the total number of data points
tol is the tolerance for establishing datum/model compatibility

```
cost = 0
for (n = 0; n < count; ++n)
    cost += error[n] ≤ tol ? error[n] : tol
```

RANSAC and MSAC

- Maximum number of trials
  - Adaptive maximum number of trials
- The tolerance for establishing datum/model compatibility

RANSAC and MSAC

- Maximum number of trials

The smaller the sample size, the smaller maximum number of trials. Use minimum solution!

Sample size	Proportion of outliers ϵ						
s	5%	10%	20%	25%	30%	40%	50%
2	2	3	5	6	7	11	17
3	3	4	7	9	11	19	35
4	3	5	9	13	17	34	72
5	4	6	12	17	26	57	146
6	4	7	16	24	37	97	293
7	4	8	20	33	54	163	588
8	5	9	26	44	78	272	1177

Table 4.3. *The number N of samples required to ensure, with a probability $p = 0.99$, that at least one sample has no outliers for a given size of sample, s, and proportion of outliers, ϵ.*

Adaptive maximum number of trials

s is the sample size

p is the assumed probability that at least one of the random samples does not contain any outliers

tol is the tolerance for establishing datum/model compatibility

τ_{cost} is the upper bound on the cost of an acceptable consensus set

$\text{maxTrials} = \infty$

$\text{consensus}_{\text{cost}} = \infty$

for (trials = 0; trials < maxTrials && $\text{consensus}_{\text{cost}} > \tau_{\text{cost}}$; ++trials)

- Select a random sample of unique data points
- Calculate the model using the random sample
- Calculate the error for each data point using model
- Calculate the cost
- if cost < $\text{consensus}_{\text{cost}}$
  - $\text{consensus}_{\text{cost}} = \text{cost}$
  - $\text{consensus}_{\text{model}} = \text{model}$
  - Calculate the number of inliers
  - $$w = \frac{\text{number of inliers}}{\text{total number of data points}}$$
  - $$\text{maxTrials} = \frac{\log(1-p)}{\log(1-w^s)}$$

Calculate the error for each data point using $\text{consensus}_{\text{model}}$

Calculate the set of inliers (i.e., data points with error $\leq$ tol)

RANSAC and MSAC

- The tolerance for establishing datum/model compatibility

Probability α that a data point is an inlier (usually α is chosen to be 0.95)
Variance σ^2 of the measurement error (if unkonwn, assumed to be 1)
Codimension m
Tolerance is square distance $t^2 = F_m^{-1}(\alpha)\sigma^2$, where $F_m^{-1}(\alpha)$ is the inverse chi-squared cumulative distribution function with m degress of freedom at the probability α

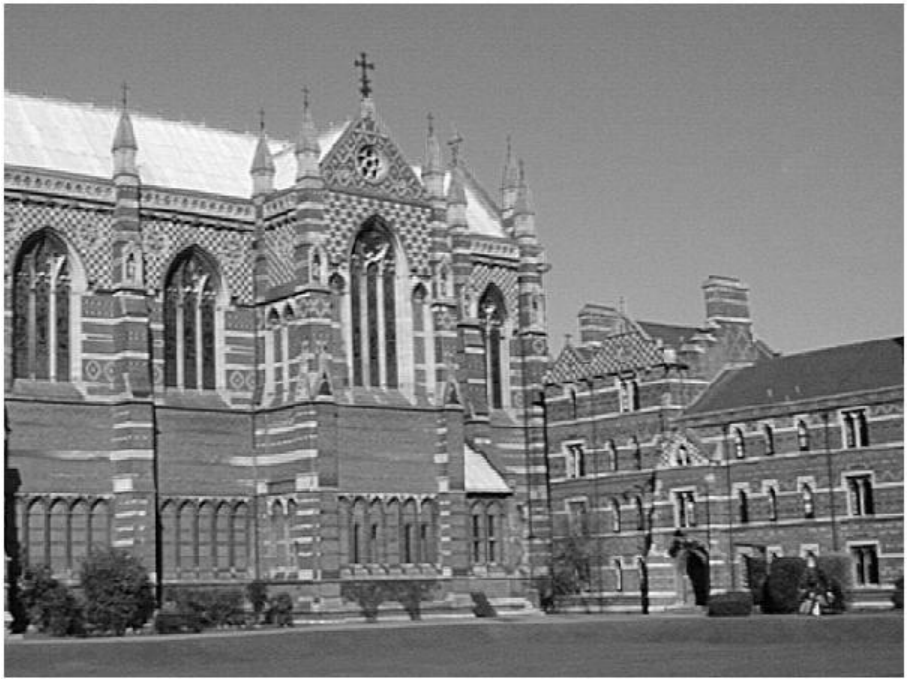
- Codimension

If w is a subspace of a finite-dimensional vector space v, then $\text{codim}(w) = \dim(v) - \dim(w)$
The 2D projective transformation is a variety of dimension 2 in $(\tilde{x}, \tilde{y}, \tilde{x}', \tilde{y}') \in \mathbb{R}^4$, so has codimension 2

Use the square Sampson error (covered next lecture)

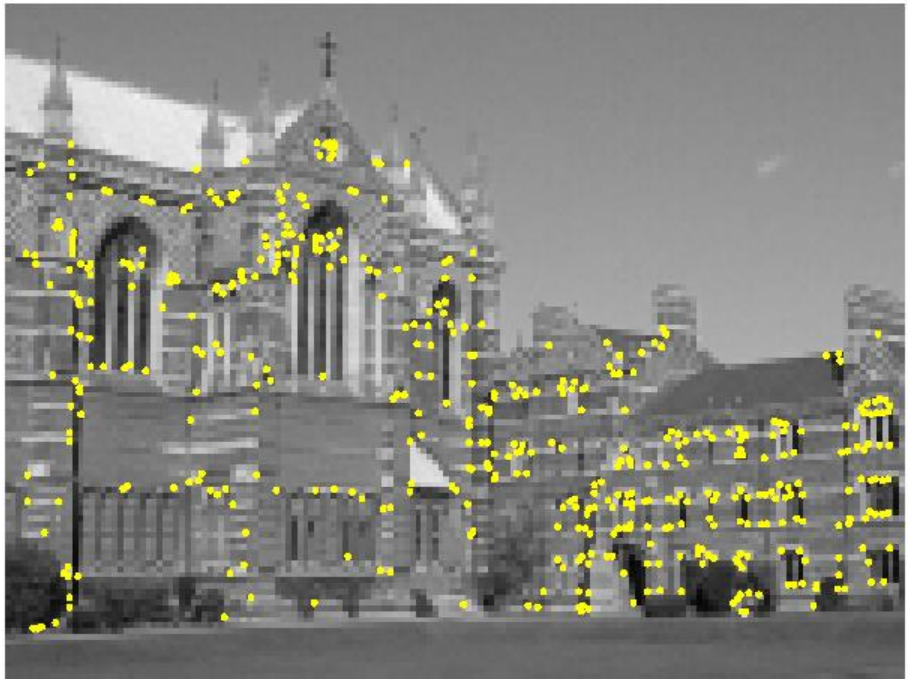
Feature detection and matching

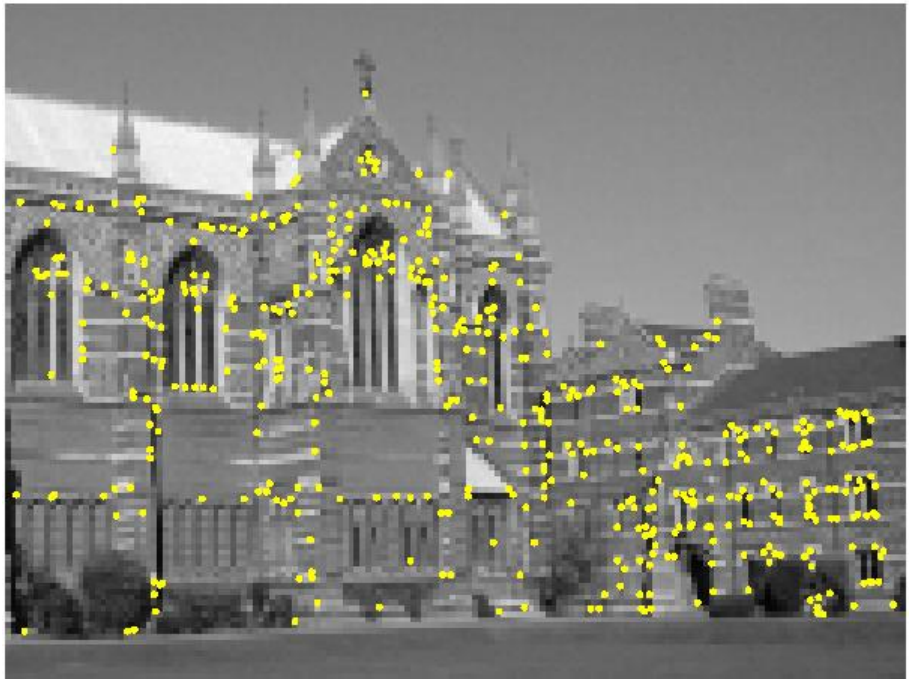




Input Images

Feature detection and matching

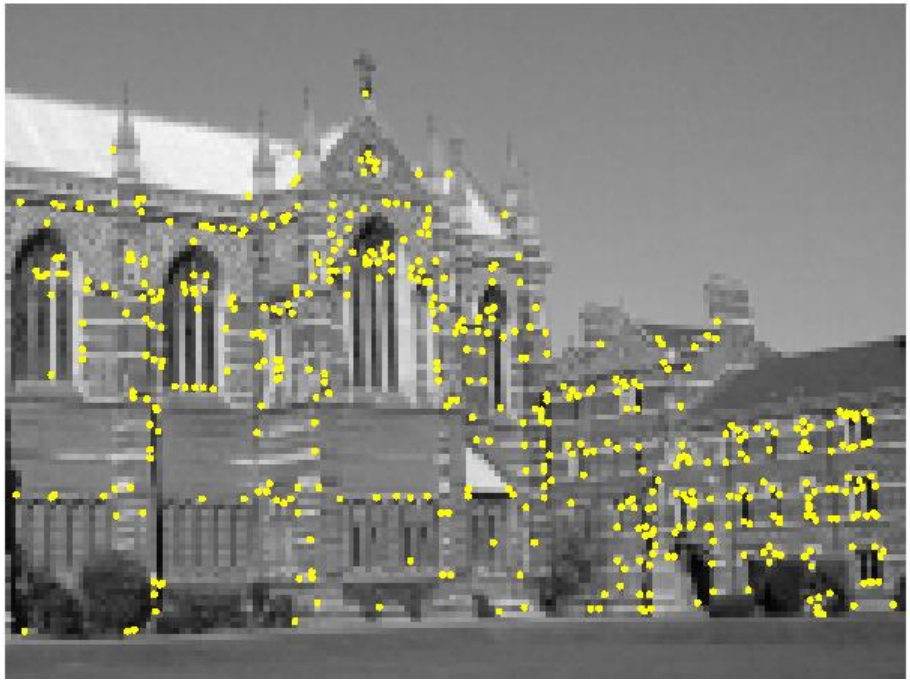




Detected Corners

Feature detection and matching

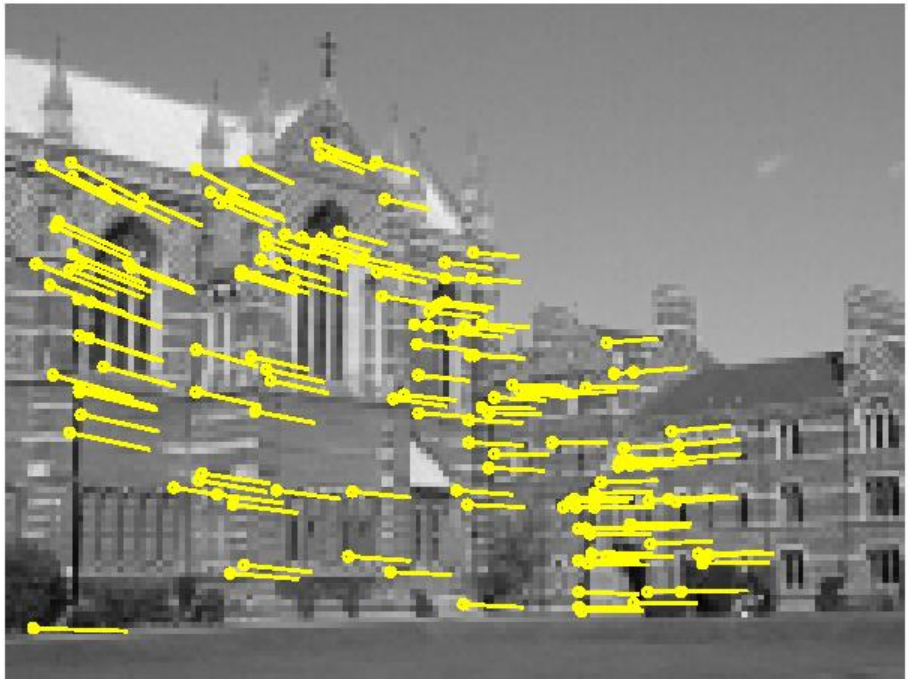


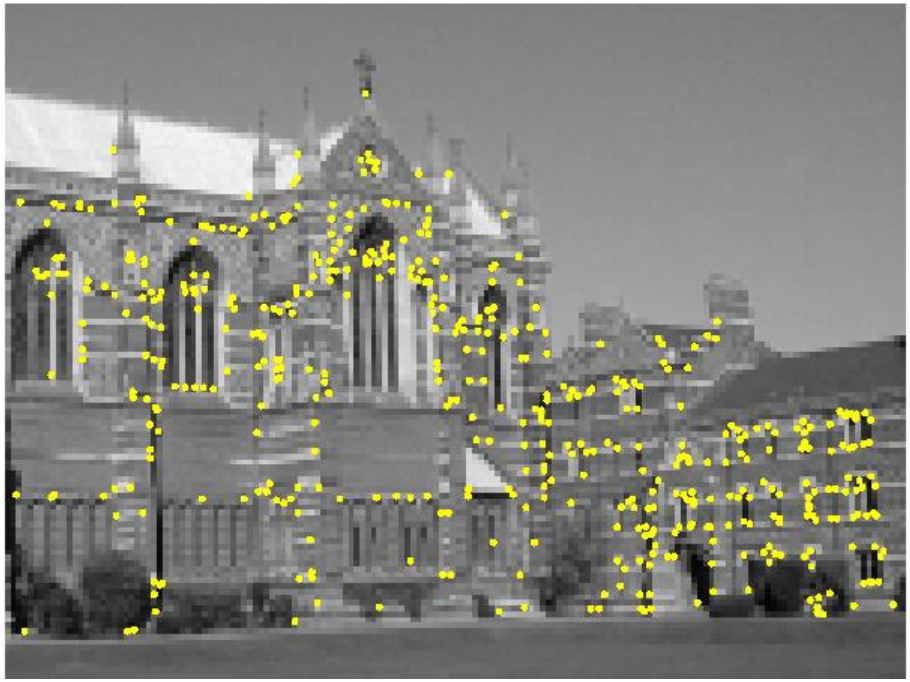


Simple Matching

Expect some false matches

Feature detection and matching





Simple Matching
Including Outlier Rejection

No false matches

Next lecture

- Estimation of 2D projective transformation
- Reading
  - Sections 4.1, 4.4, and 4.5