Computer Vision: Calibration and Reconstruction

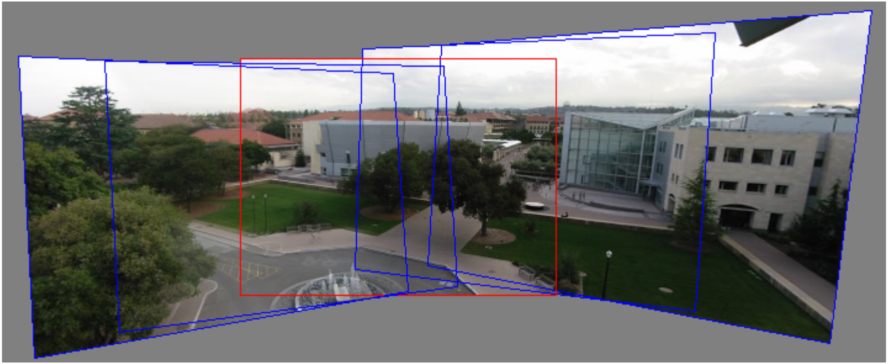
Raquel Urtasun

TTI Chicago

Feb 7, 2013

What did we see in class last week?

Panoramas



Today's Readings

- Chapter 6 and 11 of Szeliski's book

Let's look at camera calibration

Camera Calibration

Find the quantities internal to the camera that affect the imaging process as well as the position of the camera with respect to the world

- Rotation and translation

Camera Calibration

Find the quantities internal to the camera that affect the imaging process as well as the position of the camera with respect to the world

- Rotation and translation
- Position of image center in the image

Camera Calibration

Find the quantities internal to the camera that affect the imaging process as well as the position of the camera with respect to the world

- Rotation and translation
- Position of image center in the image
- Focal length

Camera Calibration

Find the quantities internal to the camera that affect the imaging process as well as the position of the camera with respect to the world

- Rotation and translation
- Position of image center in the image
- Focal length
- Different scaling factors for row pixels and column pixels

Camera Calibration

Find the quantities internal to the camera that affect the imaging process as well as the position of the camera with respect to the world

- Rotation and translation
- Position of image center in the image
- Focal length
- Different scaling factors for row pixels and column pixels
- Skew factor

Camera Calibration

Find the quantities internal to the camera that affect the imaging process as well as the position of the camera with respect to the world

- Rotation and translation
- Position of image center in the image
- Focal length
- Different scaling factors for row pixels and column pixels
- Skew factor
- Lens distortion (pin-cushion effect)

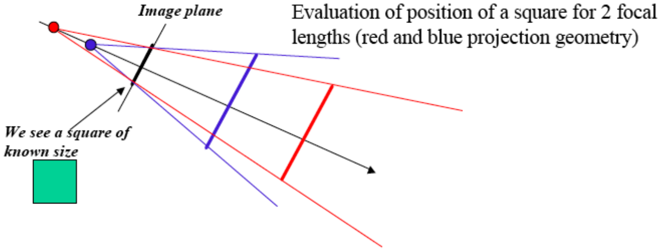
Camera Calibration

Find the quantities internal to the camera that affect the imaging process as well as the position of the camera with respect to the world

- Rotation and translation
- Position of image center in the image
- Focal length
- Different scaling factors for row pixels and column pixels
- Skew factor
- Lens distortion (pin-cushion effect)

Why do we need calibration?

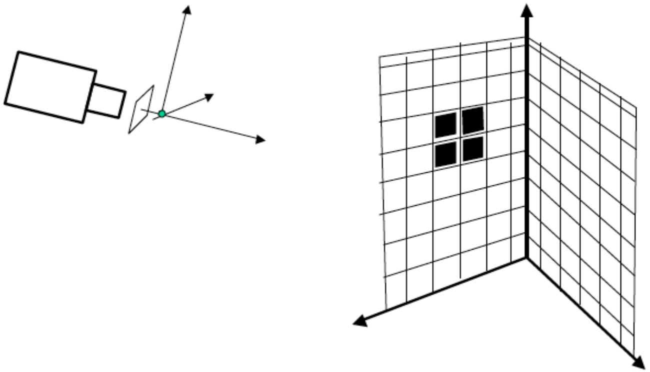
- Have good reconstruction
- Interact with the 3D world



[Source: Ramani]

Camera and Calibration Target

Most methods assume that we have a known 3D target in the scene



[Source: Ramani]

Most common used Procedure

Many algorithms!

- Calibration target: 2 planes at right angle with checkerboard patterns (Tsai grid)

Most common used Procedure

Many algorithms!

- Calibration target: 2 planes at right angle with checkerboard patterns (Tsai grid)
- We know positions of pattern corners only with respect to a coordinate system of the target

Most common used Procedure

Many algorithms!

- Calibration target: 2 planes at right angle with checkerboard patterns (Tsai grid)
- We know positions of pattern corners only with respect to a coordinate system of the target
- We position camera in front of target and find images of corners

Most common used Procedure

Many algorithms!

- Calibration target: 2 planes at right angle with checkerboard patterns (Tsai grid)
- We know positions of pattern corners only with respect to a coordinate system of the target
- We position camera in front of target and find images of corners
- We obtain equations that describe imaging and contain internal parameters of camera

Most common used Procedure

Many algorithms!

- Calibration target: 2 planes at right angle with checkerboard patterns (Tsai grid)
- We know positions of pattern corners only with respect to a coordinate system of the target
- We position camera in front of target and find images of corners
- We obtain equations that describe imaging and contain internal parameters of camera
- We also find position and orientation of camera with respect to target (camera pose)

[Source: Ramani]

Most common used Procedure

Many algorithms!

- Calibration target: 2 planes at right angle with checkerboard patterns (Tsai grid)
- We know positions of pattern corners only with respect to a coordinate system of the target
- We position camera in front of target and find images of corners
- We obtain equations that describe imaging and contain internal parameters of camera
- We also find position and orientation of camera with respect to target (camera pose)

[Source: Ramani]

Obtaining 2D-3D correspondences

- Canny edge detection
- Straight line fitting to detect linked edges. How?

Obtaining 2D-3D correspondences

- Canny edge detection
- Straight line fitting to detect linked edges. How?
- Intersect the lines to obtain the image corners

Obtaining 2D-3D correspondences

- Canny edge detection
- Straight line fitting to detect linked edges. How?
- Intersect the lines to obtain the image corners
- Matching image corners and 3D target checkerboard corners (sometimes using manual interaction)

Obtaining 2D-3D correspondences

- Canny edge detection
- Straight line fitting to detect linked edges. How?
- Intersect the lines to obtain the image corners
- Matching image corners and 3D target checkerboard corners (sometimes using manual interaction)
- We get pairs (image point)–(world point), $(x_i, y_i) \rightarrow (X_i, Y_i, Z_i)$.

[Source: Ramani]

Obtaining 2D-3D correspondences

- Canny edge detection
- Straight line fitting to detect linked edges. How?
- Intersect the lines to obtain the image corners
- Matching image corners and 3D target checkerboard corners (sometimes using manual interaction)
- We get pairs (image point)–(world point), $(x_i, y_i) \rightarrow (X_i, Y_i, Z_i)$.

[Source: Ramani]

Calibration

- Estimate matrix $\mathbf{P}$ using 2D-3D correspondences
- Estimate $\mathbf{K}$ and $(\mathbf{R}, \mathbf{t})$ from $\mathbf{P}$

$$\mathbf{P} = \mathbf{K} \cdot \mathbf{R} \cdot [\mathbf{I}_{3\times 3} \mid \mathbf{t}]$$

Calibration

- Estimate matrix $\mathbf{P}$ using 2D-3D correspondences
- Estimate $\mathbf{K}$ and $(\mathbf{R}, \mathbf{t})$ from $\mathbf{P}$

$$\mathbf{P} = \mathbf{K} \cdot \mathbf{R} \cdot [\mathbf{I}_{3\times 3} \mid \mathbf{t}]$$

- Left 3x3 submatrix of $\mathbf{P}$ is product of upper-triangular matrix and orthogonal matrix

Calibration

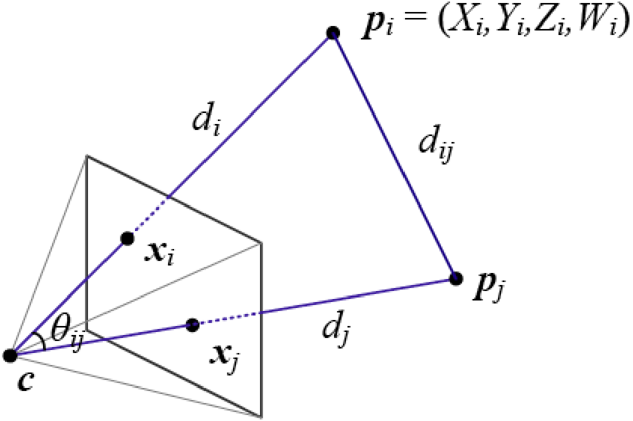
- Estimate matrix $\mathbf{P}$ using 2D-3D correspondences
- Estimate $\mathbf{K}$ and $(\mathbf{R}, \mathbf{t})$ from $\mathbf{P}$

$$\mathbf{P} = \mathbf{K} \cdot \mathbf{R} \cdot [\mathbf{I}_{3\times 3} \mid \mathbf{t}]$$

- Left 3x3 submatrix of $\mathbf{P}$ is product of upper-triangular matrix and orthogonal matrix

Inherent Constraints

- The visual angle between any pair of 2D points must be the same as the angle between their corresponding 3D points.



Direct Linear Transform

- Simplest to form a set of linear equations (analog to the 2D case)

$$
\begin{aligned}
x_i &= \frac{p_{00}X_i + p_{01}Y_i + p_{02}Z_i + p_{03}}{p_{20}X_i + p_{21}Y_i + p_{22}Y_i + p_{23}} \\
y_i &= \frac{p_{10}X_i + p_{11}Y_i + p_{12}Z_i + p_{13}}{p_{20}X_i + p_{21}Y_i + p_{22}Y_i + p_{23}}
\end{aligned}
$$

  with (x_i, y_i), the measured 2D points, and (X_i, Y_i, Z_i) are known 3D locations

- This can be solve in a linear fashion for **P**. How?

Direct Linear Transform

- Simplest to form a set of linear equations (analog to the 2D case)

$$
\begin{aligned}
x_i &= \frac{p_{00}X_i + p_{01}Y_i + p_{02}Z_i + p_{03}}{p_{20}X_i + p_{21}Y_i + p_{22}Y_i + p_{23}} \\
y_i &= \frac{p_{10}X_i + p_{11}Y_i + p_{12}Z_i + p_{13}}{p_{20}X_i + p_{21}Y_i + p_{22}Y_i + p_{23}}
\end{aligned}
$$

  with (x_i, y_i), the measured 2D points, and (X_i, Y_i, Z_i) are known 3D locations

- This can be solve in a linear fashion for **P**. How?
- Similar to the homography case.

Direct Linear Transform

- Simplest to form a set of linear equations (analog to the 2D case)

$$
\begin{aligned}
x_i &= \frac{p_{00}X_i + p_{01}Y_i + p_{02}Z_i + p_{03}}{p_{20}X_i + p_{21}Y_i + p_{22}Y_i + p_{23}} \\
y_i &= \frac{p_{10}X_i + p_{11}Y_i + p_{12}Z_i + p_{13}}{p_{20}X_i + p_{21}Y_i + p_{22}Y_i + p_{23}}
\end{aligned}
$$

  with (x_i, y_i), the measured 2D points, and (X_i, Y_i, Z_i) are known 3D locations

- This can be solve in a linear fashion for $\mathbf{P}$. How?
- Similar to the homography case.
- This is called the **Direct Linear Transform** (DLT)

Direct Linear Transform

- Simplest to form a set of linear equations (analog to the 2D case)

$$
\begin{aligned}
x_i &= \frac{p_{00}X_i + p_{01}Y_i + p_{02}Z_i + p_{03}}{p_{20}X_i + p_{21}Y_i + p_{22}Y_i + p_{23}} \\
y_i &= \frac{p_{10}X_i + p_{11}Y_i + p_{12}Z_i + p_{13}}{p_{20}X_i + p_{21}Y_i + p_{22}Y_i + p_{23}}
\end{aligned}
$$

  with (x_i, y_i), the measured 2D points, and (X_i, Y_i, Z_i) are known 3D locations

- This can be solve in a linear fashion for $\mathbf{P}$. How?
- Similar to the homography case.
- This is called the **Direct Linear Transform** (DLT)
- How many unknowns? how many correspondences?

Direct Linear Transform

- Simplest to form a set of linear equations (analog to the 2D case)

$$
\begin{aligned}
x_i &= \frac{p_{00}X_i + p_{01}Y_i + p_{02}Z_i + p_{03}}{p_{20}X_i + p_{21}Y_i + p_{22}Y_i + p_{23}} \\
y_i &= \frac{p_{10}X_i + p_{11}Y_i + p_{12}Z_i + p_{13}}{p_{20}X_i + p_{21}Y_i + p_{22}Y_i + p_{23}}
\end{aligned}
$$

  with (x_i, y_i), the measured 2D points, and (X_i, Y_i, Z_i) are known 3D locations

- This can be solve in a linear fashion for **P**. How?
- Similar to the homography case.
- This is called the **Direct Linear Transform** (DLT)
- How many unknowns? how many correspondences?
- 11 or 12 unknowns, 6 correspondences

Direct Linear Transform

- Simplest to form a set of linear equations (analog to the 2D case)

$$
\begin{aligned}
x_i &= \frac{p_{00}X_i + p_{01}Y_i + p_{02}Z_i + p_{03}}{p_{20}X_i + p_{21}Y_i + p_{22}Y_i + p_{23}} \\
y_i &= \frac{p_{10}X_i + p_{11}Y_i + p_{12}Z_i + p_{13}}{p_{20}X_i + p_{21}Y_i + p_{22}Y_i + p_{23}}
\end{aligned}
$$

  with (x_i, y_i), the measured 2D points, and (X_i, Y_i, Z_i) are known 3D locations
- This can be solve in a linear fashion for $\mathbf{P}$. How?
- Similar to the homography case.
- This is called the **Direct Linear Transform** (DLT)
- How many unknowns? how many correspondences?
- 11 or 12 unknowns, 6 correspondences
- Are we done?

Direct Linear Transform

- Simplest to form a set of linear equations (analog to the 2D case)

$$
\begin{aligned}
x_i &= \frac{p_{00}X_i + p_{01}Y_i + p_{02}Z_i + p_{03}}{p_{20}X_i + p_{21}Y_i + p_{22}Y_i + p_{23}} \\
y_i &= \frac{p_{10}X_i + p_{11}Y_i + p_{12}Z_i + p_{13}}{p_{20}X_i + p_{21}Y_i + p_{22}Y_i + p_{23}}
\end{aligned}
$$

  with (x_i, y_i), the measured 2D points, and (X_i, Y_i, Z_i) are known 3D locations

- This can be solve in a linear fashion for **P**. How?
- Similar to the homography case.
- This is called the **Direct Linear Transform** (DLT)
- How many unknowns? how many correspondences?
- 11 or 12 unknowns, 6 correspondences
- Are we done?

Finding Camera Translation

- Once $\mathbf{P}$ is recovered, how do I obtain the other matrices?
- $\mathbf{t}$ is the null vector of matrix $\mathbf{P}$ as

$$\mathbf{Pt} = 0$$

Finding Camera Translation

- Once **P** is recovered, how do I obtain the other matrices?
- **t** is the null vector of matrix **P** as

$$\mathbf{Pt} = 0$$

- **t** is the unit singular vector of **P** corresponding to the smallest singular value

Finding Camera Translation

- Once $\mathbf{P}$ is recovered, how do I obtain the other matrices?
- $\mathbf{t}$ is the null vector of matrix $\mathbf{P}$ as

$$\mathbf{Pt} = 0$$

- $\mathbf{t}$ is the unit singular vector of $\mathbf{P}$ corresponding to the smallest singular value
- The last column of $\mathbf{V}$, where

$$\mathbf{P} = \mathbf{UDV}^T$$

  is the SVD of $\mathbf{P}$

Finding Camera Translation

- Once $\mathbf{P}$ is recovered, how do I obtain the other matrices?
- $\mathbf{t}$ is the null vector of matrix $\mathbf{P}$ as

$$\mathbf{Pt} = 0$$

- $\mathbf{t}$ is the unit singular vector of $\mathbf{P}$ corresponding to the smallest singular value
- The last column of $\mathbf{V}$, where

$$\mathbf{P} = \mathbf{UDV}^T$$

  is the SVD of $\mathbf{P}$

Finding Camera Orientation and Internal Parameters

- Left 3×3 submatrix $\mathbf{M}$ of $\mathbf{P}$ is of form

$$\mathbf{M} = \mathbf{KR}$$

  where $\mathbf{K}$ is an upper triangular matrix, and $\mathbf{R}$ is an orthogonal matrix

- Any non-singular square matrix $\mathbf{M}$ can be decomposed into the product of an upper-triangular matrix $\mathbf{K}$ and an orthogonal matrix $\mathbf{R}$ using the RQ factorization

Finding Camera Orientation and Internal Parameters

- Left 3×3 submatrix $\mathbf{M}$ of $\mathbf{P}$ is of form

$$\mathbf{M} = \mathbf{KR}$$

  where $\mathbf{K}$ is an upper triangular matrix, and $\mathbf{R}$ is an orthogonal matrix

- Any non-singular square matrix $\mathbf{M}$ can be decomposed into the product of an upper-triangular matrix $\mathbf{K}$ and an orthogonal matrix $\mathbf{R}$ using the RQ factorization
- Similar to QR factorization but order of 2 matrices is reversed

Finding Camera Orientation and Internal Parameters

- Left 3×3 submatrix $\mathbf{M}$ of $\mathbf{P}$ is of form

$$\mathbf{M} = \mathbf{K}\mathbf{R}$$

  where $\mathbf{K}$ is an upper triangular matrix, and $\mathbf{R}$ is an orthogonal matrix

- Any non-singular square matrix $\mathbf{M}$ can be decomposed into the product of an upper-triangular matrix $\mathbf{K}$ and an orthogonal matrix $\mathbf{R}$ using the RQ factorization
- Similar to QR factorization but order of 2 matrices is reversed

RQ Factorization

- Define the matrices

$$\mathbf{R}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c & -s \\ 0 & s & c \end{bmatrix}, \mathbf{R}_y = \begin{bmatrix} c' & 0 & s' \\ 0 & 1 & 0 \\ -s' & 0 & c' \end{bmatrix}, \mathbf{R}_z = \begin{bmatrix} c'' & -s'' & 0 \\ s'' & c'' & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- We can compute

$$c = -\frac{M_{33}}{(M_{32}^2 + M_{33}^2)^{1/2}} \qquad s = \frac{M_{32}}{(M_{32}^2 + M_{33}^2)^{1/2}}$$

RQ Factorization

- Define the matrices

$$\mathbf{R}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c & -s \\ 0 & s & c \end{bmatrix}, \mathbf{R}_y = \begin{bmatrix} c' & 0 & s' \\ 0 & 1 & 0 \\ -s' & 0 & c' \end{bmatrix}, \mathbf{R}_z = \begin{bmatrix} c'' & -s'' & 0 \\ s'' & c'' & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- We can compute

$$c = -\frac{M_{33}}{(M_{32}^2 + M_{33}^2)^{1/2}} \qquad s = \frac{M_{32}}{(M_{32}^2 + M_{33}^2)^{1/2}}$$

- Multiply $\mathbf{M}$ by $\mathbf{R}_x$. The resulting term at $(3, 2)$ is zero because of the values selected for c and s

RQ Factorization

- Define the matrices

$$\mathbf{R}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c & -s \\ 0 & s & c \end{bmatrix}, \mathbf{R}_y = \begin{bmatrix} c' & 0 & s' \\ 0 & 1 & 0 \\ -s' & 0 & c' \end{bmatrix}, \mathbf{R}_z = \begin{bmatrix} c'' & -s'' & 0 \\ s'' & c'' & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- We can compute

$$c = -\frac{M_{33}}{(M_{32}^2 + M_{33}^2)^{1/2}} \qquad s = \frac{M_{32}}{(M_{32}^2 + M_{33}^2)^{1/2}}$$

- Multiply $\mathbf{M}$ by $\mathbf{R}_x$. The resulting term at $(3,2)$ is zero because of the values selected for c and s
- Multiply the resulting matrix by $\mathbf{R}_y$, after selecting c and s so that the resulting term at position $(3,1)$ is set to zero

RQ Factorization

- Define the matrices

$$\mathbf{R}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c & -s \\ 0 & s & c \end{bmatrix}, \mathbf{R}_y = \begin{bmatrix} c' & 0 & s' \\ 0 & 1 & 0 \\ -s' & 0 & c' \end{bmatrix}, \mathbf{R}_z = \begin{bmatrix} c'' & -s'' & 0 \\ s'' & c'' & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- We can compute

$$c = -\frac{M_{33}}{(M_{32}^2 + M_{33}^2)^{1/2}} \qquad s = \frac{M_{32}}{(M_{32}^2 + M_{33}^2)^{1/2}}$$

- Multiply $\mathbf{M}$ by $\mathbf{R}_x$. The resulting term at $(3, 2)$ is zero because of the values selected for c and s
- Multiply the resulting matrix by $\mathbf{R}_y$, after selecting c and s so that the resulting term at position $(3, 1)$ is set to zero
- Multiply the resulting matrix by $\mathbf{R}_z$, after selecting c and s so that the resulting term at position $(2, 1)$ is set to zero

RQ Factorization

- Define the matrices

$$\mathbf{R}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c & -s \\ 0 & s & c \end{bmatrix}, \mathbf{R}_y = \begin{bmatrix} c' & 0 & s' \\ 0 & 1 & 0 \\ -s' & 0 & c' \end{bmatrix}, \mathbf{R}_z = \begin{bmatrix} c'' & -s'' & 0 \\ s'' & c'' & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- We can compute

$$c = -\frac{M_{33}}{(M_{32}^2 + M_{33}^2)^{1/2}} \qquad s = \frac{M_{32}}{(M_{32}^2 + M_{33}^2)^{1/2}}$$

- Multiply $\mathbf{M}$ by $\mathbf{R}_x$. The resulting term at $(3, 2)$ is zero because of the values selected for c and s
- Multiply the resulting matrix by $\mathbf{R}_y$, after selecting c and s so that the resulting term at position $(3, 1)$ is set to zero
- Multiply the resulting matrix by $\mathbf{R}_z$, after selecting c and s so that the resulting term at position $(2, 1)$ is set to zero

RQ Factorization

- Why does this algorithm work?

$$\mathbf{M}\mathbf{R}_x\mathbf{R}_y\mathbf{R}_z = \mathbf{K}$$

- Thus we have

$$\mathbf{M} = \mathbf{K}\mathbf{R}_z^T\mathbf{R}_y^T\mathbf{R}_x^T = \mathbf{K}\mathbf{R}$$

RQ Factorization

- Why does this algorithm work?

$$\mathbf{M}\mathbf{R}_x\mathbf{R}_y\mathbf{R}_z = \mathbf{K}$$

- Thus we have

$$\mathbf{M} = \mathbf{K}\mathbf{R}_z^T\mathbf{R}_y^T\mathbf{R}_x^T = \mathbf{K}\mathbf{R}$$

Improved computation of $\mathbf{P}$

- We have equations involving homgeneous coordinates

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

- Thus $\mathbf{x}_i$ and $\mathbf{X}_i$ just have to be proportional

$$\mathbf{x}_i \times \mathbf{P}\mathbf{X}_i = 0$$

Improved computation of $\mathbf{P}$

- We have equations involving homgeneous coordinates

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

- Thus $\mathbf{x}_i$ and $\mathbf{X}_i$ just have to be proportional

$$\mathbf{x}_i \times \mathbf{P}\mathbf{X}_i = 0$$

- Let $\mathbf{p}_1^T, \mathbf{p}_2^T, \mathbf{p}_3^T$ be the row vectors of $\mathbf{P}$

$$\mathbf{P}\mathbf{X}_i = \begin{bmatrix} \mathbf{p}_1^T \mathbf{X}_i \\ \mathbf{p}_2^T \mathbf{X}_i \\ \mathbf{p}_3^T \mathbf{X}_i \end{bmatrix}$$

Improved computation of $\mathbf{P}$

- We have equations involving homgeneous coordinates

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

- Thus $\mathbf{x}_i$ and $\mathbf{X}_i$ just have to be proportional

$$\mathbf{x}_i \times \mathbf{P}\mathbf{X}_i = 0$$

- Let $\mathbf{p}_1^T, \mathbf{p}_2^T, \mathbf{p}_3^T$ be the row vectors of $\mathbf{P}$

$$\mathbf{P}\mathbf{X}_i = \begin{bmatrix} \mathbf{p}_1^T \mathbf{X}_i \\ \mathbf{p}_2^T \mathbf{X}_i \\ \mathbf{p}_3^T \mathbf{X}_i \end{bmatrix}$$

- Therefore

$$\mathbf{x}_i \times \mathbf{P}\mathbf{X}_i = \begin{bmatrix} v_i \mathbf{p}_3^T \mathbf{X}_i - w_i \mathbf{p}_2^T \mathbf{X}_i \\ w_i \mathbf{p}_3^T \mathbf{X}_i - u_i \mathbf{p}_2^T \mathbf{X}_i \\ u_i \mathbf{p}_3^T \mathbf{X}_i - v_i \mathbf{p}_2^T \mathbf{X}_i \end{bmatrix}$$

Improved computation of $\mathbf{P}$

- We have equations involving homgeneous coordinates

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

- Thus $\mathbf{x}_i$ and $\mathbf{X}_i$ just have to be proportional

$$\mathbf{x}_i \times \mathbf{P}\mathbf{X}_i = 0$$

- Let $\mathbf{p}_1^T, \mathbf{p}_2^T, \mathbf{p}_3^T$ be the row vectors of $\mathbf{P}$

$$\mathbf{P}\mathbf{X}_i = \begin{bmatrix} \mathbf{p}_1^T\mathbf{X}_i \\ \mathbf{p}_2^T\mathbf{X}_i \\ \mathbf{p}_3^T\mathbf{X}_i \end{bmatrix}$$

- Therefore

$$\mathbf{x}_i \times \mathbf{P}\mathbf{X}_i = \begin{bmatrix} v_i\mathbf{p}_3^T\mathbf{X}_i - w_i\mathbf{p}_2^T\mathbf{X}_i \\ w_i\mathbf{p}_3^T\mathbf{X}_i - u_i\mathbf{p}_2^T\mathbf{X}_i \\ u_i\mathbf{p}_3^T\mathbf{X}_i - v_i\mathbf{p}_2^T\mathbf{X}_i \end{bmatrix}$$

Improved computation of $\mathbf{P}$

- We have

$$\mathbf{x}_i \times \mathbf{P}\mathbf{X}_i = \begin{bmatrix} v_i \mathbf{p}_3^T \mathbf{X}_i - w_i \mathbf{p}_2^T \mathbf{X}_i \\ w_i \mathbf{p}_3^T \mathbf{X}_i - u_i \mathbf{p}_2^T \mathbf{X}_i \\ u_i \mathbf{p}_3^T \mathbf{X}_i - v_i \mathbf{p}_2^T \mathbf{X}_i \end{bmatrix} = 0$$

- We can thus write

$$\begin{bmatrix} 0_4 & -w_i \mathbf{X}_i^T & v_i \mathbf{X}_i^T \\ w_i \mathbf{X}_i^T & 0_4 & -u_i \mathbf{X}_i^T \\ -v_i \mathbf{X}_i^T & u_i \mathbf{X}_i^T & 0_4 \end{bmatrix} \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix} = 0$$

Improved computation of **P**

- We have

$$\mathbf{x}_i \times \mathbf{P}\mathbf{X}_i = \begin{bmatrix} v_i \mathbf{p}_3^T \mathbf{X}_i - w_i \mathbf{p}_2^T \mathbf{X}_i \\ w_i \mathbf{p}_3^T \mathbf{X}_i - u_i \mathbf{p}_2^T \mathbf{X}_i \\ u_i \mathbf{p}_3^T \mathbf{X}_i - v_i \mathbf{p}_2^T \mathbf{X}_i \end{bmatrix} = 0$$

- We can thus write

$$\begin{bmatrix} 0_4 & -w_i \mathbf{X}_i^T & v_i \mathbf{X}_i^T \\ w_i \mathbf{X}_i^T & 0_4 & -u_i \mathbf{X}_i^T \\ -v_i \mathbf{X}_i^T & u_i \mathbf{X}_i^T & 0_4 \end{bmatrix} \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix} = 0$$

- Third row can be obtained from sum of u_i times first row $-v_i$ times second row

Improved computation of $\mathbf{P}$

- We have

$$\mathbf{x}_i \times \mathbf{P}\mathbf{X}_i = \begin{bmatrix} v_i \mathbf{p}_3^T \mathbf{X}_i - w_i \mathbf{p}_2^T \mathbf{X}_i \\ w_i \mathbf{p}_3^T \mathbf{X}_i - u_i \mathbf{p}_2^T \mathbf{X}_i \\ u_i \mathbf{p}_3^T \mathbf{X}_i - v_i \mathbf{p}_2^T \mathbf{X}_i \end{bmatrix} = 0$$

- We can thus write

$$\begin{bmatrix} 0_4 & -w_i \mathbf{X}_i^T & v_i \mathbf{X}_i^T \\ w_i \mathbf{X}_i^T & 0_4 & -u_i \mathbf{X}_i^T \\ -v_i \mathbf{X}_i^T & u_i \mathbf{X}_i^T & 0_4 \end{bmatrix} \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix} = 0$$

- Third row can be obtained from sum of u_i times first row $-v_i$ times second row
- 2 independent equations in 11 unknowns (ignoring scale)

Improved computation of $\mathbf{P}$

- We have

$$\mathbf{x}_i \times \mathbf{P}\mathbf{X}_i = \begin{bmatrix} v_i \mathbf{p}_3^T \mathbf{X}_i - w_i \mathbf{p}_2^T \mathbf{X}_i \\ w_i \mathbf{p}_3^T \mathbf{X}_i - u_i \mathbf{p}_2^T \mathbf{X}_i \\ u_i \mathbf{p}_3^T \mathbf{X}_i - v_i \mathbf{p}_2^T \mathbf{X}_i \end{bmatrix} = 0$$

- We can thus write

$$\begin{bmatrix} 0_4 & -w_i \mathbf{X}_i^T & v_i \mathbf{X}_i^T \\ w_i \mathbf{X}_i^T & 0_4 & -u_i \mathbf{X}_i^T \\ -v_i \mathbf{X}_i^T & u_i \mathbf{X}_i^T & 0_4 \end{bmatrix} \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix} = 0$$

- Third row can be obtained from sum of u_i times first row $-v_i$ times second row
- 2 independent equations in 11 unknowns (ignoring scale)
- With 6 correspondences, we get enough equations to compute matrix $\mathbf{P}$

$$\mathbf{A}\mathbf{p} = 0$$

Improved computation of $\mathbf{P}$

- We have

$$\mathbf{x}_i \times \mathbf{P}\mathbf{X}_i = \begin{bmatrix} v_i \mathbf{p}_3^T \mathbf{X}_i - w_i \mathbf{p}_2^T \mathbf{X}_i \\ w_i \mathbf{p}_3^T \mathbf{X}_i - u_i \mathbf{p}_2^T \mathbf{X}_i \\ u_i \mathbf{p}_3^T \mathbf{X}_i - v_i \mathbf{p}_2^T \mathbf{X}_i \end{bmatrix} = 0$$

- We can thus write

$$\begin{bmatrix} 0_4 & -w_i \mathbf{X}_i^T & v_i \mathbf{X}_i^T \\ w_i \mathbf{X}_i^T & 0_4 & -u_i \mathbf{X}_i^T \\ -v_i \mathbf{X}_i^T & u_i \mathbf{X}_i^T & 0_4 \end{bmatrix} \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix} = 0$$

- Third row can be obtained from sum of u_i times first row $-v_i$ times second row
- 2 independent equations in 11 unknowns (ignoring scale)
- With 6 correspondences, we get enough equations to compute matrix $\mathbf{P}$

$$\mathbf{A}\mathbf{p} = 0$$

Solving the system

$$\mathbf{Ap} = 0$$

- Minimize $||\mathbf{Ap}||$ with the constraint $||\mathbf{p}|| = 1$

Solving the system

$$\mathbf{Ap} = 0$$

- Minimize $||\mathbf{Ap}||$ with the constraint $||\mathbf{p}|| = 1$
- $\mathbf{P}$ is the unit singular vector of $\mathbf{A}$ corresponding to the smallest singular value

Solving the system

$$\mathbf{Ap} = 0$$

- Minimize $||\mathbf{Ap}||$ with the constraint $||\mathbf{p}|| = 1$
- $\mathbf{P}$ is the unit singular vector of $\mathbf{A}$ corresponding to the smallest singular value
- The last column of $\mathbf{V}$, where

$$\mathbf{A} = \mathbf{UDV}^T$$

  is the SVD of $\mathbf{A}$

Solving the system

$$\mathbf{Ap} = 0$$

- Minimize $||\mathbf{Ap}||$ with the constraint $||\mathbf{p}|| = 1$
- $\mathbf{P}$ is the unit singular vector of $\mathbf{A}$ corresponding to the smallest singular value
- The last column of $\mathbf{V}$, where

$$\mathbf{A} = \mathbf{UDV}^T$$

  is the SVD of $\mathbf{A}$
- Called **Direct Linear Transformation (DLT)**

Solving the system

$$\mathbf{Ap} = 0$$

- Minimize $||\mathbf{Ap}||$ with the constraint $||\mathbf{p}|| = 1$
- $\mathbf{P}$ is the unit singular vector of $\mathbf{A}$ corresponding to the smallest singular value
- The last column of $\mathbf{V}$, where

  $$\mathbf{A} = \mathbf{UDV}^T$$

  is the SVD of $\mathbf{A}$
- Called **Direct Linear Transformation (DLT)**

Improving the $\mathbf{P}$ estimation

- In most applications we have prior knowledge about some of the parameters of $\mathbf{K}$, e.g., pixels are squared, skew is small, optical center near the center of the image
- Use this constraints and frame the problem as a minimization

Improving the $\mathbf{P}$ estimation

- In most applications we have prior knowledge about some of the parameters of $\mathbf{K}$, e.g., pixels are squared, skew is small, optical center near the center of the image
- Use this constraints and frame the problem as a minimization
  - Find $\mathbf{P}$ using DLT

Improving the $\mathbf{P}$ estimation

- In most applications we have prior knowledge about some of the parameters of $\mathbf{K}$, e.g., pixels are squared, skew is small, optical center near the center of the image
- Use this constraints and frame the problem as a minimization
  - Find $\mathbf{P}$ using DLT
  - Use as initialization for nonlinear minimization $\sum_i \rho(\mathbf{x}_i, \mathbf{P}\mathbf{X}_i)$, with ρ a robust estimator

Improving the $\mathbf{P}$ estimation

- In most applications we have prior knowledge about some of the parameters of $\mathbf{K}$, e.g., pixels are squared, skew is small, optical center near the center of the image
- Use this constraints and frame the problem as a minimization
  - Find $\mathbf{P}$ using DLT
  - Use as initialization for nonlinear minimization $\sum_i \rho(\mathbf{x}_i, \mathbf{P}\mathbf{X}_i)$, with ρ a robust estimator
  - Use Levenberg-Marquardt iterative minimization

Improving the $\mathbf{P}$ estimation

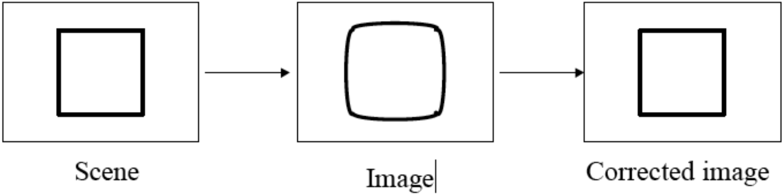
- In most applications we have prior knowledge about some of the parameters of $\mathbf{K}$, e.g., pixels are squared, skew is small, optical center near the center of the image
- Use this constraints and frame the problem as a minimization
  - Find $\mathbf{P}$ using DLT
  - Use as initialization for nonlinear minimization $\sum_i \rho(\mathbf{x}_i, \mathbf{P}\mathbf{X}_i)$, with ρ a robust estimator
  - Use Levenberg-Marquardt iterative minimization
- See 6.2.2. in Szeliski's book

Improving the $\mathbf{P}$ estimation

- In most applications we have prior knowledge about some of the parameters of $\mathbf{K}$, e.g., pixels are squared, skew is small, optical center near the center of the image
- Use this constraints and frame the problem as a minimization
  - Find $\mathbf{P}$ using DLT
  - Use as initialization for nonlinear minimization $\sum_i \rho(\mathbf{x}_i, \mathbf{P}\mathbf{X}_i)$, with ρ a robust estimator
  - Use Levenberg-Marquardt iterative minimization
- See 6.2.2. in Szeliski's book

Radial Distorsion

- We have assumed that lines are imaged as lines
- Significant error for cheap optics and for short focal lengths

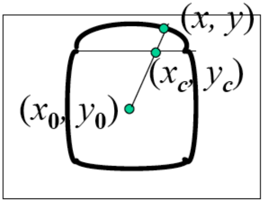


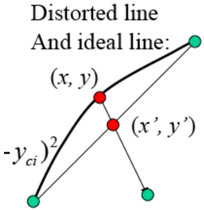
Radial Distorsion Correction

- We can write the correction as

$$\begin{aligned} x_c - x_0 &= L(r)(x - x_0) \\ y_c - y_0 &= L(r)(y - y_0) \end{aligned}$$

with $L(r) = 1 + \kappa_1 r^2 + \kappa_2 r^4$ and $r^2 = (x - x_0)^2 + (y - y_0)^2$





- We thus minimize the following function

$$f(\kappa_1, \kappa_2) = \sum_i (x_i' - x_{ci})^2 + (y_i' - y_{ci})^2$$

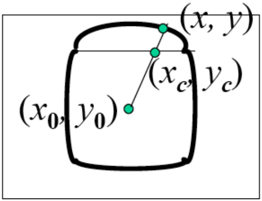
using lines known to be straight, with (x', y') the radial projection of (x, y) on straight line

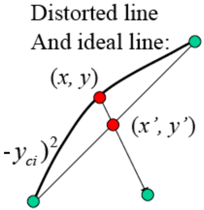
Radial Distorsion Correction

- We can write the correction as

$$\begin{aligned} x_c - x_0 &= L(r)(x - x_0) \\ y_c - y_0 &= L(r)(y - y_0) \end{aligned}$$

with $L(r) = 1 + \kappa_1 r^2 + \kappa_2 r^4$ and $r^2 = (x - x_0)^2 + (y - y_0)^2$





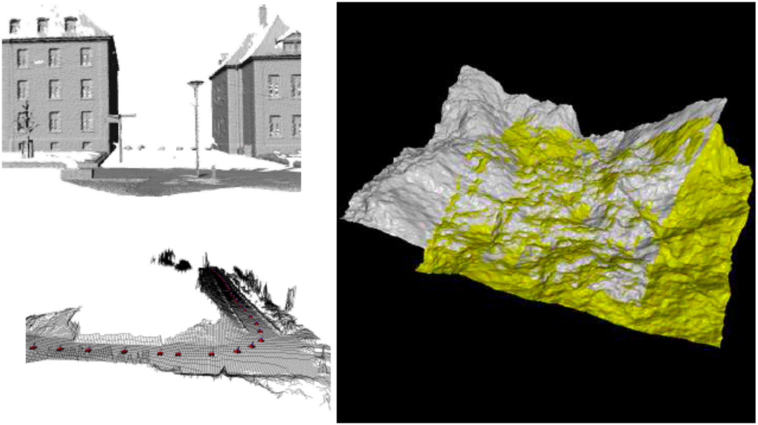
- We thus minimize the following function

$$f(\kappa_1, \kappa_2) = \sum_i (x_i' - x_{ci})^2 + (y_i' - y_{ci})^2$$

using lines known to be straight, with (x', y') the radial projection of (x, y) on straight line

Let's look at 3D alignment

Motivation



[Source: W. Burgard]

3D Alignment

- Before we were looking at aligning 2D points, or 2D to 3D points.
- Now let's do it in 3D

3D Alignment

- Before we were looking at aligning 2D points, or 2D to 3D points.
- Now let's do it in 3D
- Given two corresponding point sets $\{\mathbf{x}_i, \mathbf{x}'_i\}$, in the case of rigid (Euclidean) motion we want $\mathbf{R}$ and $\mathbf{t}$ that minimizes

$$E_{R3D} = \sum_i ||\mathbf{x}'_i - \mathbf{R}\mathbf{x}_i - \mathbf{t}||_2^2$$

3D Alignment

- Before we were looking at aligning 2D points, or 2D to 3D points.
- Now let's do it in 3D
- Given two corresponding point sets $\{\mathbf{x}_i, \mathbf{x}'_i\}$, in the case of rigid (Euclidean) motion we want $\mathbf{R}$ and $\mathbf{t}$ that minimizes

$$E_{R3D} = \sum_i ||\mathbf{x}'_i - \mathbf{R}\mathbf{x}_i - \mathbf{t}||_2^2$$

- This is call the **absolute orientation problem**

3D Alignment

- Before we were looking at aligning 2D points, or 2D to 3D points.
- Now let's do it in 3D
- Given two corresponding point sets $\{\mathbf{x}_i, \mathbf{x}'_i\}$, in the case of rigid (Euclidean) motion we want $\mathbf{R}$ and $\mathbf{t}$ that minimizes

$$E_{R3D} = \sum_i ||\mathbf{x}'_i - \mathbf{R}\mathbf{x}_i - \mathbf{t}||_2^2$$

- This is call the **absolute orientation problem**
- Is this easy to do?

3D Alignment

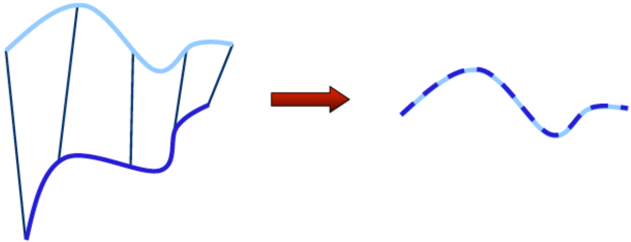
- Before we were looking at aligning 2D points, or 2D to 3D points.
- Now let's do it in 3D
- Given two corresponding point sets $\{\mathbf{x}_i, \mathbf{x}'_i\}$, in the case of rigid (Euclidean) motion we want $\mathbf{R}$ and $\mathbf{t}$ that minimizes

$$E_{R3D} = \sum_i ||\mathbf{x}'_i - \mathbf{R}\mathbf{x}_i - \mathbf{t}||_2^2$$

- This is call the **absolute orientation problem**
- Is this easy to do?

Key Idea

- If the correct correspondences are known, the correct relative rotation/translation can be calculated in closed form



[Source: W. Burgard]

Center of Mass

- The centroids of the two point clouds c and c' can be used to estimate the translation

$$\mu = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i \qquad \mu' = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}'_i$$

- Subtract the corresponding center of mass from every point in the two sets,

$$\begin{aligned}\bar{\mathbf{x}}_i &= \mathbf{x}_i - \mu \\ \bar{\mathbf{x}}'_i &= \mathbf{x}'_i - \mu'\end{aligned}$$

Center of Mass

- The centroids of the two point clouds c and c' can be used to estimate the translation

$$\mu = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i \qquad \mu' = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}'_i$$

- Subtract the corresponding center of mass from every point in the two sets,

$$\begin{aligned}\bar{\mathbf{x}}_i &= \mathbf{x}_i - \mu \\ \bar{\mathbf{x}}'_i &= \mathbf{x}'_i - \mu'\end{aligned}$$

Solving for the Rotation

- Let $\mathbf{W} = \sum_{i=1}^{N} \bar{\mathbf{x}}_i' \bar{\mathbf{x}}_i^T$
- We can compute the SVD of $\mathbf{W}$

$$\mathbf{W} = \mathbf{U}\mathbf{S}\mathbf{V}^T$$

Solving for the Rotation

- Let $\mathbf{W} = \sum_{i=1}^{N} \bar{\mathbf{x}}_i' \bar{\mathbf{x}}_i^T$
- We can compute the SVD of $\mathbf{W}$

$$\mathbf{W} = \mathbf{U}\mathbf{S}\mathbf{V}^T$$

- If the rank is 3 we have a unique solution

$$\begin{aligned} \mathbf{R} &= \mathbf{U}\mathbf{V}^T \\ \mathbf{t} &= \mu - \mathbf{R}\mu' \end{aligned}$$

Solving for the Rotation

- Let $\mathbf{W} = \sum_{i=1}^{N} \bar{\mathbf{x}}_i' \bar{\mathbf{x}}_i^T$
- We can compute the SVD of $\mathbf{W}$

$$\mathbf{W} = \mathbf{U}\mathbf{S}\mathbf{V}^T$$

- If the rank is 3 we have a unique solution

$$\begin{aligned} \mathbf{R} &= \mathbf{U}\mathbf{V}^T \\ \mathbf{t} &= \mu - \mathbf{R}\mu' \end{aligned}$$

- What if we don't have correspondences?

Solving for the Rotation

- Let $\mathbf{W} = \sum_{i=1}^{N} \bar{\mathbf{x}}'_i \bar{\mathbf{x}}_i^T$
- We can compute the SVD of $\mathbf{W}$

$$\mathbf{W} = \mathbf{U}\mathbf{S}\mathbf{V}^T$$

- If the rank is 3 we have a unique solution

$$\begin{aligned} \mathbf{R} &= \mathbf{U}\mathbf{V}^T \\ \mathbf{t} &= \mu - \mathbf{R}\mu' \end{aligned}$$

- What if we don't have correspondences?

ICP

- If the correspondences are not know, it is not possible to obtain the global solution
- **Iterated Closest Points** (ICP) [Besl & McKay 92]

ICP

- If the correspondences are not know, it is not possible to obtain the global solution
- **Iterated Closest Points** (ICP) [Besl & McKay 92]
- Iterative algorithm that alternates between solving for correspondences and solving for $(\mathbf{R}, \mathbf{t})$

ICP

- If the correspondences are not know, it is not possible to obtain the global solution
- **Iterated Closest Points** (ICP) [Besl & McKay 92]
- Iterative algorithm that alternates between solving for correspondences and solving for $(\mathbf{R}, \mathbf{t})$
- Does it remained you of any algorithm you have seen before?

ICP

- If the correspondences are not know, it is not possible to obtain the global solution
- **Iterated Closest Points** (ICP) [Besl & McKay 92]
- Iterative algorithm that alternates between solving for correspondences and solving for $(\mathbf{R}, \mathbf{t})$
- Does it remained you of any algorithm you have seen before?
- Converges to right solution if starting positions are "close enough"

ICP

- If the correspondences are not know, it is not possible to obtain the global solution
- **Iterated Closest Points** (ICP) [Besl & McKay 92]
- Iterative algorithm that alternates between solving for correspondences and solving for $(\mathbf{R}, \mathbf{t})$
- Does it remained you of any algorithm you have seen before?
- Converges to right solution if starting positions are "close enough"

ICP Variants

- Point subsets (from one or both point sets)
- Weighting the correspondences

ICP Variants

- Point subsets (from one or both point sets)
- Weighting the correspondences
- Data association

ICP Variants

- Point subsets (from one or both point sets)
- Weighting the correspondences
- Data association
- Rejecting certain (outlier) point pairs

[Source: W. Burgard]

ICP Variants

- Point subsets (from one or both point sets)
- Weighting the correspondences
- Data association
- Rejecting certain (outlier) point pairs

[Source: W. Burgard]

Performance of Variants

- Speed
- Stability (local minima)

Performance of Variants

- Speed
- Stability (local minima)
- Tolerance wrt. noise and/or outliers

Performance of Variants

- Speed
- Stability (local minima)
- Tolerance wrt. noise and/or outliers
- Basin of convergence (maximum initial misalignment)

[Source: W. Burgard]

Performance of Variants

- Speed
- Stability (local minima)
- Tolerance wrt. noise and/or outliers
- Basin of convergence (maximum initial misalignment)

[Source: W. Burgard]

ICP Variants

- Point subsets (from one or both point sets)
- Weighting the correspondences
- Data association
- Rejecting certain (outlier) point pairs

ICP Variants

- Use all points
- Uniform sub-sampling

ICP Variants

- Use all points
- Uniform sub-sampling
- Random sampling

ICP Variants

- Use all points
- Uniform sub-sampling
- Random sampling
- Normal-space sampling distributed as uniformly as possible

ICP Variants

- Use all points
- Uniform sub-sampling
- Random sampling
- Normal-space sampling distributed as uniformly as possible
- Feature based Sampling

[Source: W. Burgard]

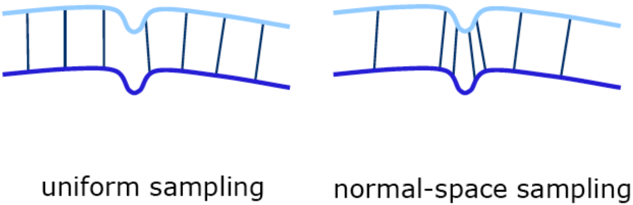
ICP Variants

- Use all points
- Uniform sub-sampling
- Random sampling
- Normal-space sampling distributed as uniformly as possible
- Feature based Sampling

[Source: W. Burgard]

Normal-based sampling

- Ensure that samples have normals distributed as uniformly as possible

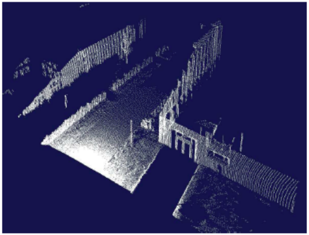


- better for mostly-smooth areas with sparse features

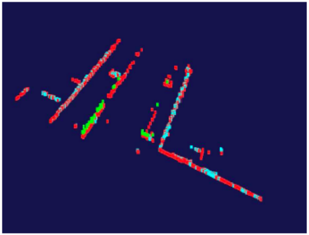
[Source: W. Burgard]

Feature-based sampling

- try to find important points
- decrease the number of correspondences
- higher efficiency and higher accuracy
- requires preprocessing



3D Scan (~200.000 Points)



Extracted Features (~5.000 Points)

[Source: W. Burgard]

ICP Variants

- Point subsets (from one or both point sets)
- Weighting the correspondences
- Data association
- Rejecting certain (outlier) point pairs

Selection vs. Weighting

- Could achieve same effect with weighting
- Hard to guarantee that enough samples of important features except at high sampling rates

Selection vs. Weighting

- Could achieve same effect with weighting
- Hard to guarantee that enough samples of important features except at high sampling rates
- Weighting strategies turned out to be dependent on the data.

Selection vs. Weighting

- Could achieve same effect with weighting
- Hard to guarantee that enough samples of important features except at high sampling rates
- Weighting strategies turned out to be dependent on the data.
- Preprocessing / run-time cost tradeoff (how to find the correct weights?)

[Source: W. Burgard]

Selection vs. Weighting

- Could achieve same effect with weighting
- Hard to guarantee that enough samples of important features except at high sampling rates
- Weighting strategies turned out to be dependent on the data.
- Preprocessing / run-time cost tradeoff (how to find the correct weights?)

[Source: W. Burgard]

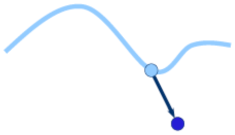
ICP Variants

- Point subsets (from one or both point sets)
- Weighting the correspondences
- Data association
- Rejecting certain (outlier) point pairs

Data association

Has greatest effect on convergence and speed

- **Closest point**: stable, but slow and requires preprocessing



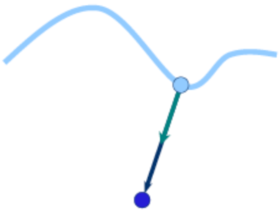
- **Normal shooting**: slightly better than closest point for smooth structures, worse for noisy or complex structures

[Source: W. Burgard]

Data association

Has greatest effect on convergence and speed

- **Closest point**: stable, but slow and requires preprocessing
- **Normal shooting**: slightly better than closest point for smooth structures, worse for noisy or complex structures



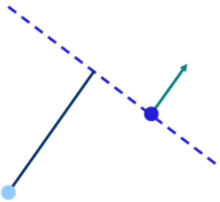
- **Point to plane**: lets flat regions slide along each other

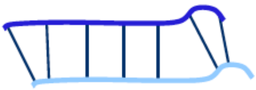
[Source: W. Burgard]

Data association

Has greatest effect on convergence and speed

- **Closest point**: stable, but slow and requires preprocessing
- **Normal shooting**: slightly better than closest point for smooth structures, worse for noisy or complex structures
- **Point to plane**: lets flat regions slide along each other





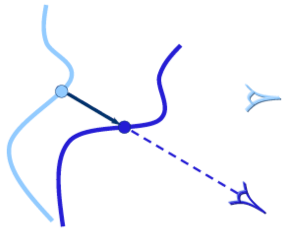
- **Projection**: much faster but slightly worst alignments

[Source: W. Burgard]

Data association

Has greatest effect on convergence and speed

- **Closest point**: stable, but slow and requires preprocessing
- **Normal shooting**: slightly better than closest point for smooth structures, worse for noisy or complex structures
- **Point to plane**: lets flat regions slide along each other
- **Projection**: much faster but slightly worst alignments



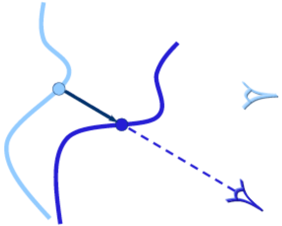
- **Closest compatible point**: normals, colors

[Source: W. Burgard]

Data association

Has greatest effect on convergence and speed

- **Closest point**: stable, but slow and requires preprocessing
- **Normal shooting**: slightly better than closest point for smooth structures, worse for noisy or complex structures
- **Point to plane**: lets flat regions slide along each other
- **Projection**: much faster but slightly worst alignments



- **Closest compatible point**: normals, colors

[Source: W. Burgard]

ICP Variants

- Point subsets (from one or both point sets)
- Weighting the correspondences
- Data association
- Rejecting certain (outlier) point pairs, e.g., Trimmed ICP rejects a %

Where do these 3D points come from?

Let's look into stereo reconstruction

Stereo

Public Library, Stereoscopic Looking Room, Chicago, by Phillips, 1923

[Source: N. Snavely]

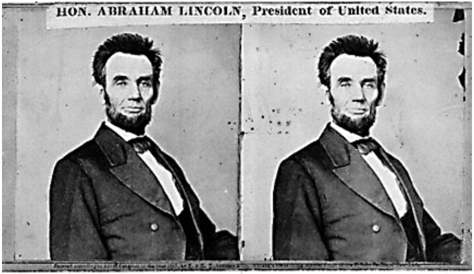
Stereo

- Stereo matching is the process of taking two or more images and **estimating a 3D model** of the scene by **finding matching pixels** in the images and converting their 2D positions into 3D depths
- We perceived depth based on the difference in appearance of the right and left eye.

Stereo

- Stereo matching is the process of taking two or more images and **estimating a 3D model** of the scene by **finding matching pixels** in the images and converting their 2D positions into 3D depths
- We perceived depth based on the difference in appearance of the right and left eye.

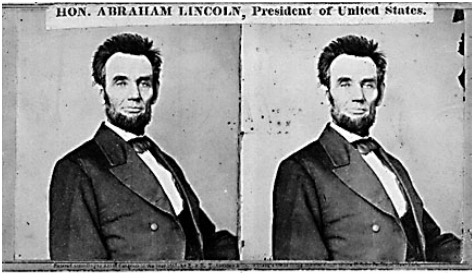
Stereo



Given two images from different viewpoints

- The depth is proportional to the inverse of the **disparity**
- How can we compute the depth of each point in the image?

Stereo

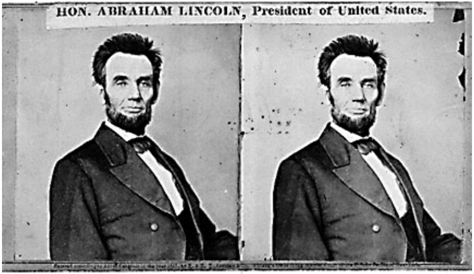


Given two images from different viewpoints

- The depth is proportional to the inverse of the **disparity**
- How can we compute the depth of each point in the image?
- Based on how much each pixel moves between the two images

[Source: N. Snavely]

Stereo

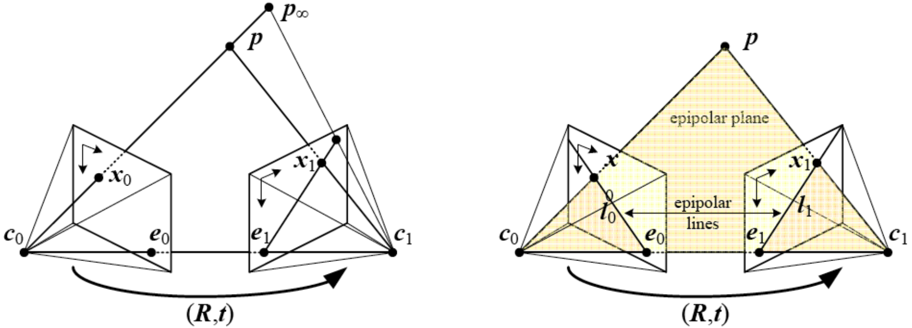


Given two images from different viewpoints

- The depth is proportional to the inverse of the **disparity**
- How can we compute the depth of each point in the image?
- Based on how much each pixel moves between the two images

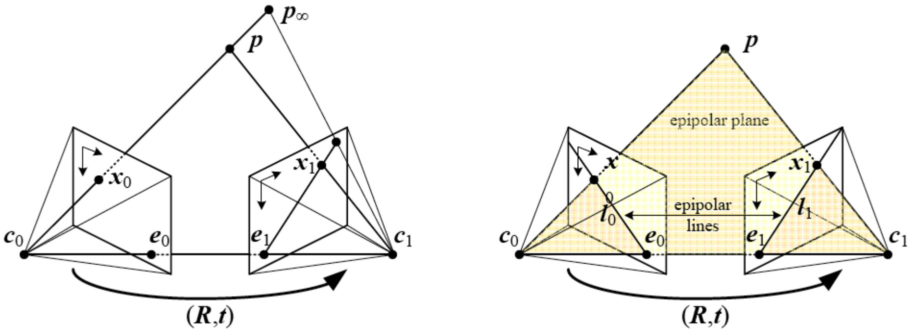
[Source: N. Snavely]

Epipolar Geometry



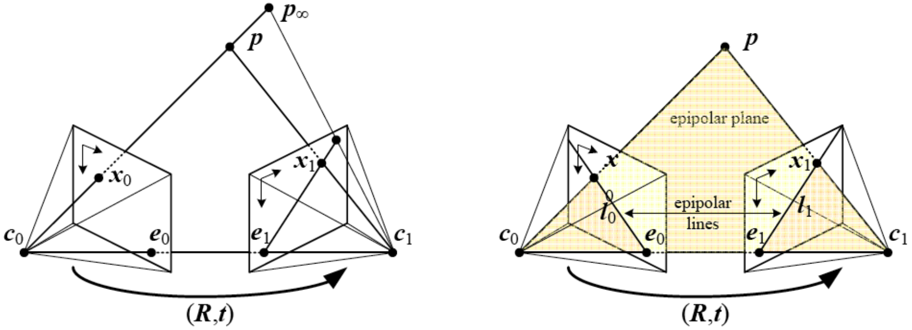
- Pixel in one image $\mathbf{x}_0$ projects to an **epipolar line** segment in the other image
- The segment is bounded at one end by the projection of the original viewing ray at infinity $\mathbf{p}_\infty$ and at the other end by the projection of the original camera center $\mathbf{c}_0$ into the second camera, which is known as the **epipole** $\mathbf{e}_1$.

Epipolar Geometry



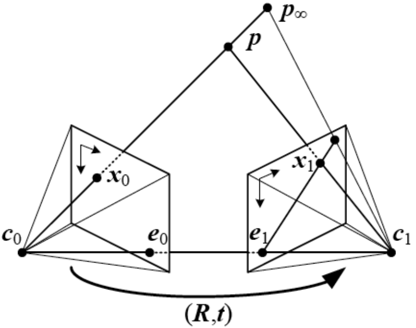
- Pixel in one image $\mathbf{x}_0$ projects to an **epipolar line** segment in the other image
- The segment is bounded at one end by the projection of the original viewing ray at infinity $\mathbf{p}_\infty$ and at the other end by the projection of the original camera center $\mathbf{c}_0$ into the second camera, which is known as the **epipole** $\mathbf{e}_1$.

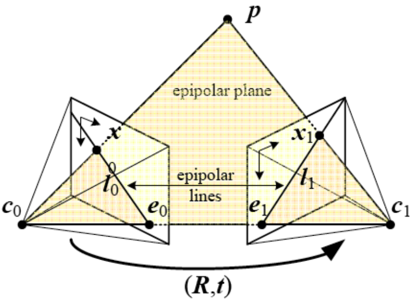
Epipolar Geometry



- If we project the epipolar line in the second image back into the first, we get another line (segment), this time bounded by the other corresponding **epipole** e_0
- Extending both line segments to infinity, we get a pair of corresponding epipolar lines, which are the intersection of the two image planes with the **epipolar plane** that passes through both camera centers $\mathbf{c}_0$ and $\mathbf{c}_1$ as well as the point of interest $\mathbf{p}$

Epipolar Geometry





- If we project the epipolar line in the second image back into the first, we get another line (segment), this time bounded by the other corresponding **epipole** e_0
- Extending both line segments to infinity, we get a pair of corresponding epipolar lines, which are the intersection of the two image planes with the **epipolar plane** that passes through both camera centers $\mathbf{c}_0$ and $\mathbf{c}_1$ as well as the point of interest $\mathbf{p}$

Rectification

- The epipolar geometry depends on the relative **pose** and **calibration** of the cameras
- This can be computed using the **fundamental matrix**

Rectification

- The epipolar geometry depends on the relative **pose** and **calibration** of the cameras
- This can be computed using the **fundamental matrix**
- Once this is computed, we can use the epipolar lines to restrict the search space to a 1D search

Rectification

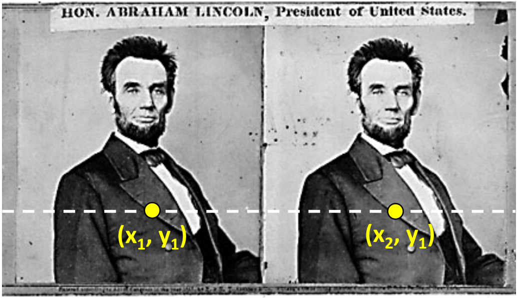
- The epipolar geometry depends on the relative **pose** and **calibration** of the cameras
- This can be computed using the **fundamental matrix**
- Once this is computed, we can use the epipolar lines to restrict the search space to a 1D search
- **Rectification**, the process of transforming the image so that the search is along horizontal line

Rectification

- The epipolar geometry depends on the relative **pose** and **calibration** of the cameras
- This can be computed using the **fundamental matrix**
- Once this is computed, we can use the epipolar lines to restrict the search space to a 1D search
- **Rectification**, the process of transforming the image so that the search is along horizontal line

Epipolar Geometry

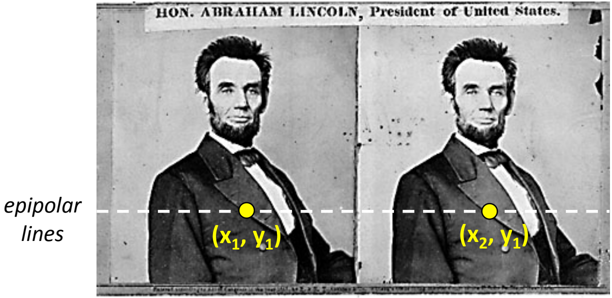
epipolar
lines



Two images captured by a purely horizontal translating camera
(*rectified* stereo pair)

- The disparity for pixel (x_1, y_1) is $(x_2 - x_1)$ if the images are rectified
- This is a one dimensional search for each pixel

Epipolar Geometry

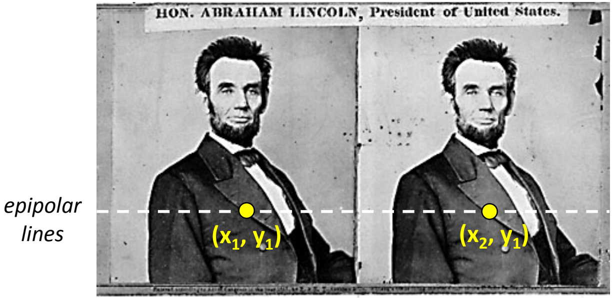


Two images captured by a purely horizontal translating camera
(*rectified* stereo pair)

- The disparity for pixel (x_1, y_1) is $(x_2 - x_1)$ if the images are rectified
- This is a one dimensional search for each pixel
- Very challenging to estimate the correspondences

[Source: N. Snavely]

Epipolar Geometry

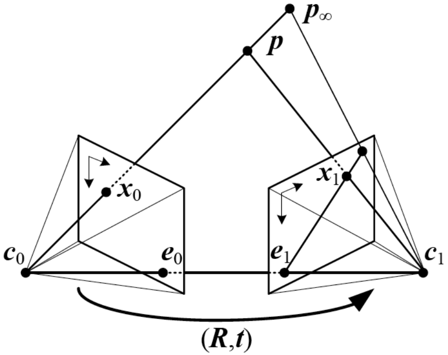


Two images captured by a purely horizontal translating camera
(*rectified* stereo pair)

- The disparity for pixel (x_1, y_1) is $(x_2 - x_1)$ if the images are rectified
- This is a one dimensional search for each pixel
- Very challenging to estimate the correspondences

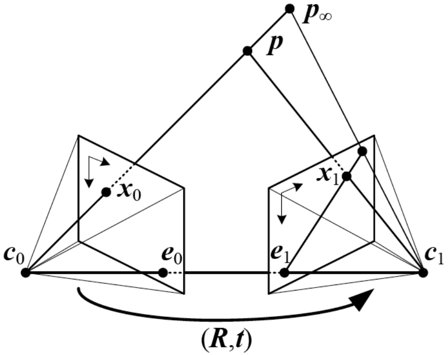
[Source: N. Snavely]

Fundamental Matrix



- Projective geometry depends only on the cameras internal parameters and relative pose of cameras
- Fundamental matrix **F** encapsulates this geometry

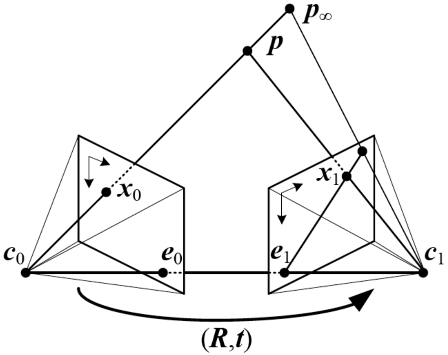
Fundamental Matrix



- Projective geometry depends only on the cameras internal parameters and relative pose of cameras
- Fundamental matrix **F** encapsulates this geometry
- For any pair of points correspoding in both images

$$\mathbf{x}_0^T \mathbf{F} \mathbf{x}_1 = 0$$

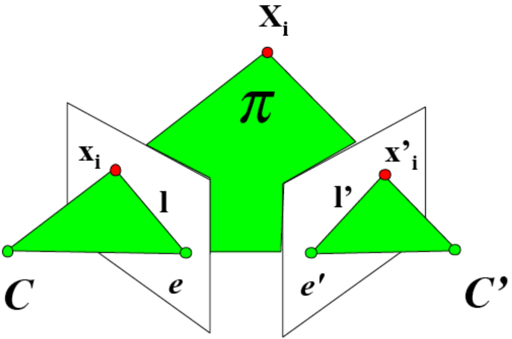
Fundamental Matrix



- Projective geometry depends only on the cameras internal parameters and relative pose of cameras
- Fundamental matrix $\mathbf{F}$ encapsulates this geometry
- For any pair of points correspoding in both images

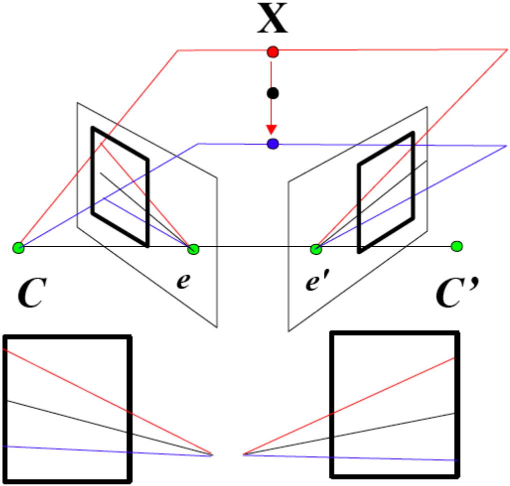
$$\mathbf{x}_0^T \mathbf{F} \mathbf{x}_1 = 0$$

Epipolar Plane



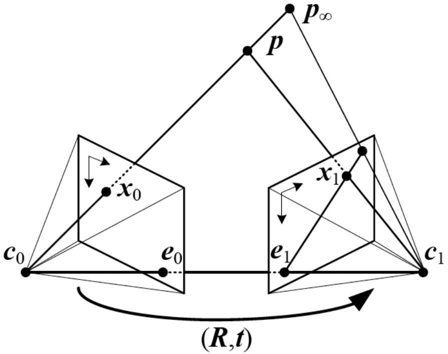
[Source: Ramani]

Pensils of Epipolar Lines



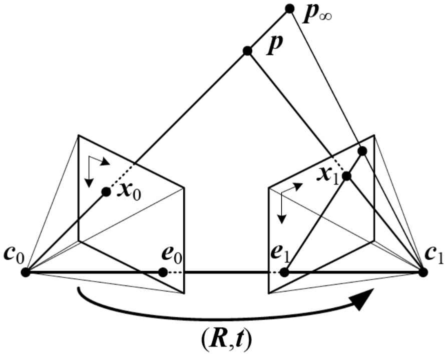
[Source: Ramani]

Computation of Fundamental Matrix



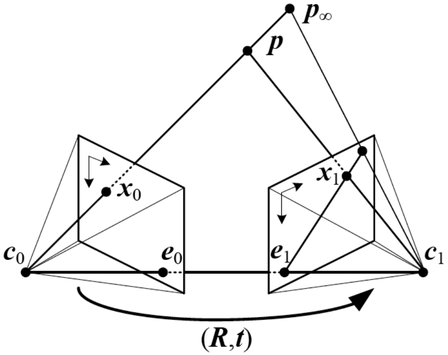
- **F** can be computed from correspondences between image points alone
- No knowledge of camera internal parameters required

Computation of Fundamental Matrix



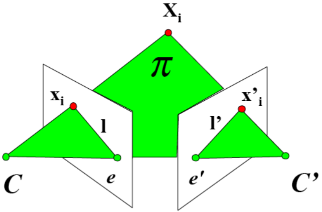
- **F** can be computed from correspondences between image points alone
- No knowledge of camera internal parameters required
- No knowledge of relative pose required

Computation of Fundamental Matrix



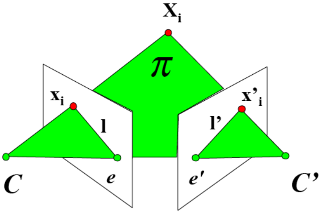
- **F** can be computed from correspondences between image points alone
- No knowledge of camera internal parameters required
- No knowledge of relative pose required

Fundamental Matrix and Projective Geometry



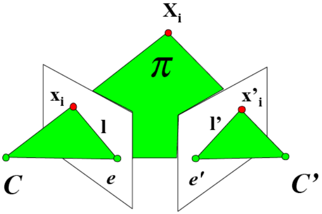
- Take $\mathbf{x}$ in camera $\mathbf{P}$ and find scene point $\mathbf{X}$ on ray of $\mathbf{x}$ in camera $\mathbf{P}$
- Find the image $\mathbf{x}'$ of $\mathbf{X}$ in camera $\mathbf{P}'$

Fundamental Matrix and Projective Geometry



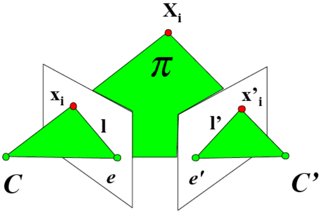
- Take $\mathbf{x}$ in camera $\mathbf{P}$ and find scene point $\mathbf{X}$ on ray of $\mathbf{x}$ in camera $\mathbf{P}$
- Find the image $\mathbf{x}'$ of $\mathbf{X}$ in camera $\mathbf{P}'$
- Find epipole $\mathbf{e}'$ as image of $\mathbf{C}$ in camera $\mathbf{P}'$, $\mathbf{e}' = \mathbf{P}'\mathbf{C}$

Fundamental Matrix and Projective Geometry



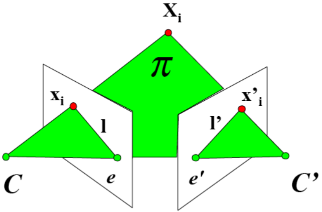
- Take $\mathbf{x}$ in camera $\mathbf{P}$ and find scene point $\mathbf{X}$ on ray of $\mathbf{x}$ in camera $\mathbf{P}$
- Find the image $\mathbf{x}'$ of $\mathbf{X}$ in camera $\mathbf{P}'$
- Find epipole $\mathbf{e}'$ as image of $\mathbf{C}$ in camera $\mathbf{P}'$, $\mathbf{e}' = \mathbf{P}'\mathbf{C}$
- Find epipolar line $\mathbf{l}'$ from $\mathbf{e}'$ to $\mathbf{x}'$ in $\mathbf{P}'$ as function of $\mathbf{x}$

Fundamental Matrix and Projective Geometry



- Take $\mathbf{x}$ in camera $\mathbf{P}$ and find scene point $\mathbf{X}$ on ray of $\mathbf{x}$ in camera $\mathbf{P}$
- Find the image $\mathbf{x}'$ of $\mathbf{X}$ in camera $\mathbf{P}'$
- Find epipole $\mathbf{e}'$ as image of $\mathbf{C}$ in camera $\mathbf{P}'$, $\mathbf{e}' = \mathbf{P}'\mathbf{C}$
- Find epipolar line $\mathbf{l}'$ from $\mathbf{e}'$ to $\mathbf{x}'$ in $\mathbf{P}'$ as function of $\mathbf{x}$
- The fundamental matrix $\mathbf{F}$ is defined $\mathbf{l}' = \mathbf{F}\mathbf{x}$

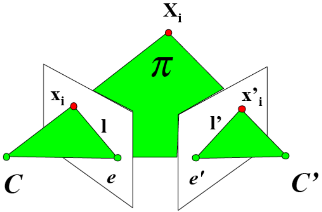
Fundamental Matrix and Projective Geometry



- Take $\mathbf{x}$ in camera $\mathbf{P}$ and find scene point $\mathbf{X}$ on ray of $\mathbf{x}$ in camera $\mathbf{P}$
- Find the image $\mathbf{x}'$ of $\mathbf{X}$ in camera $\mathbf{P}'$
- Find epipole $\mathbf{e}'$ as image of $\mathbf{C}$ in camera $\mathbf{P}'$, $\mathbf{e}' = \mathbf{P}'\mathbf{C}$
- Find epipolar line $\mathbf{l}'$ from $\mathbf{e}'$ to $\mathbf{x}'$ in $\mathbf{P}'$ as function of $\mathbf{x}$
- The fundamental matrix $\mathbf{F}$ is defined $\mathbf{l}' = \mathbf{F}\mathbf{x}$
- $\mathbf{x}'$ belongs to $\mathbf{l}'$, so $\mathbf{x}'^T\mathbf{l}' = 0$, so

$$\mathbf{x}'^T\mathbf{F}\mathbf{x} = 0$$

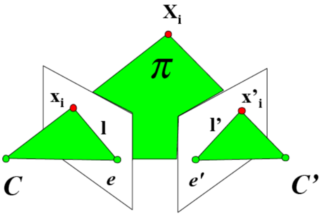
Fundamental Matrix and Projective Geometry



- Take $\mathbf{x}$ in camera $\mathbf{P}$ and find scene point $\mathbf{X}$ on ray of $\mathbf{x}$ in camera $\mathbf{P}$
- Find the image $\mathbf{x}'$ of $\mathbf{X}$ in camera $\mathbf{P}'$
- Find epipole $\mathbf{e}'$ as image of $\mathbf{C}$ in camera $\mathbf{P}'$, $\mathbf{e}' = \mathbf{P}'\mathbf{C}$
- Find epipolar line $\mathbf{l}'$ from $\mathbf{e}'$ to $\mathbf{x}'$ in $\mathbf{P}'$ as function of $\mathbf{x}$
- The fundamental matrix $\mathbf{F}$ is defined $\mathbf{l}' = \mathbf{F}\mathbf{x}$
- $\mathbf{x}'$ belongs to $\mathbf{l}'$, so $\mathbf{x}'^T\mathbf{l}' = 0$, so

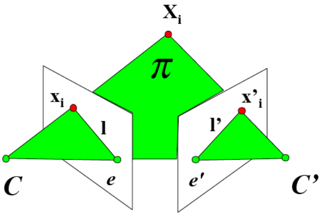
$$\mathbf{x}'^T\mathbf{F}\mathbf{x} = 0$$

Finding the Fundamental Matrix from known Projections



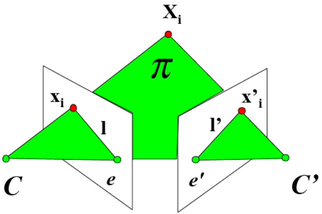
- Take $\mathbf{x}$ in camera $\mathbf{P}$ and find one scene point on ray from $\mathbf{C}$ to $\mathbf{x}$
- Point $\mathbf{X} = \mathbf{P}^{+}\mathbf{x}$ satisfies $\mathbf{x} = \mathbf{PX}$ with $\mathbf{P}^{+} = \mathbf{P}^{T}(\mathbf{PP}^{T})^{-1}$ so $\mathbf{PX} = \mathbf{PP}^{T}(\mathbf{PP}^{T})^{-1}\mathbf{x} = \mathbf{x}$

Finding the Fundamental Matrix from known Projections



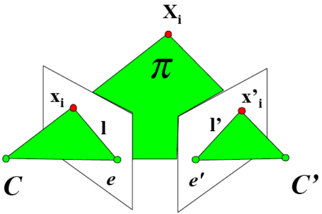
- Take $\mathbf{x}$ in camera $\mathbf{P}$ and find one scene point on ray from $\mathbf{C}$ to $\mathbf{x}$
- Point $\mathbf{X} = \mathbf{P}^{+}\mathbf{x}$ satisfies $\mathbf{x} = \mathbf{P}\mathbf{X}$ with $\mathbf{P}^{+} = \mathbf{P}^{T}(\mathbf{P}\mathbf{P}^{T})^{-1}$ so $\mathbf{P}\mathbf{X} = \mathbf{P}\mathbf{P}^{T}(\mathbf{P}\mathbf{P}^{T})^{-1}\mathbf{x} = \mathbf{x}$
- Image of this point in camera $\mathbf{P}'$ is $\mathbf{x}' = \mathbf{P}'\mathbf{X} = \mathbf{P}'\mathbf{P}^{+}\mathbf{x}$

Finding the Fundamental Matrix from known Projections



- Take $\mathbf{x}$ in camera $\mathbf{P}$ and find one scene point on ray from $\mathbf{C}$ to $\mathbf{x}$
- Point $\mathbf{X} = \mathbf{P}^{+}\mathbf{x}$ satisfies $\mathbf{x} = \mathbf{PX}$ with $\mathbf{P}^{+} = \mathbf{P}^{T}(\mathbf{PP}^{T})^{-1}$ so $\mathbf{PX} = \mathbf{PP}^{T}(\mathbf{PP}^{T})^{-1}\mathbf{x} = \mathbf{x}$
- Image of this point in camera $\mathbf{P}'$ is $\mathbf{x}' = \mathbf{P}'\mathbf{X} = \mathbf{P}'\mathbf{P}^{+}\mathbf{x}$
- Image of $\mathbf{C}$ in camera $\mathbf{P}'$ is epipole $\mathbf{e}' = \mathbf{P}'\mathbf{C}$

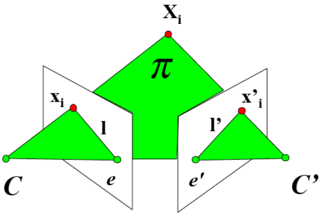
Finding the Fundamental Matrix from known Projections



- Take $\mathbf{x}$ in camera $\mathbf{P}$ and find one scene point on ray from $\mathbf{C}$ to $\mathbf{x}$
- Point $\mathbf{X} = \mathbf{P}^+\mathbf{x}$ satisfies $\mathbf{x} = \mathbf{P}\mathbf{X}$ with $\mathbf{P}^+ = \mathbf{P}^T(\mathbf{P}\mathbf{P}^T)^{-1}$ so $\mathbf{P}\mathbf{X} = \mathbf{P}\mathbf{P}^T(\mathbf{P}\mathbf{P}^T)^{-1}\mathbf{x} = \mathbf{x}$
- Image of this point in camera $\mathbf{P}'$ is $\mathbf{x}' = \mathbf{P}'\mathbf{X} = \mathbf{P}'\mathbf{P}^+\mathbf{x}$
- Image of $\mathbf{C}$ in camera $\mathbf{P}'$ is epipole $\mathbf{e}' = \mathbf{P}'\mathbf{C}$
- Epipolar line of $\mathbf{x}$ in $\mathbf{P}'$ is

$$\mathbf{l}' = (\mathbf{e}') \times (\mathbf{P}'\mathbf{P}^+\mathbf{x})$$

Finding the Fundamental Matrix from known Projections



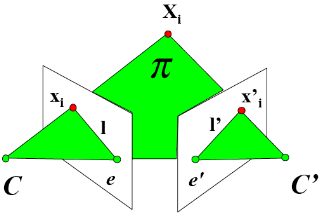
- Take $\mathbf{x}$ in camera $\mathbf{P}$ and find one scene point on ray from $\mathbf{C}$ to $\mathbf{x}$
- Point $\mathbf{X} = \mathbf{P}^+\mathbf{x}$ satisfies $\mathbf{x} = \mathbf{P}\mathbf{X}$ with $\mathbf{P}^+ = \mathbf{P}^T(\mathbf{P}\mathbf{P}^T)^{-1}$ so $\mathbf{P}\mathbf{X} = \mathbf{P}\mathbf{P}^T(\mathbf{P}\mathbf{P}^T)^{-1}\mathbf{x} = \mathbf{x}$
- Image of this point in camera $\mathbf{P}'$ is $\mathbf{x}' = \mathbf{P}'\mathbf{X} = \mathbf{P}'\mathbf{P}^+\mathbf{x}$
- Image of $\mathbf{C}$ in camera $\mathbf{P}'$ is epipole $\mathbf{e}' = \mathbf{P}'\mathbf{C}$
- Epipolar line of $\mathbf{x}$ in $\mathbf{P}'$ is

$$\mathbf{l}' = (\mathbf{e}') \times (\mathbf{P}'\mathbf{P}^+\mathbf{x})$$

- $\mathbf{l}' = \mathbf{F}\mathbf{x}$ defines the fundamental matrix

$$\mathbf{F} = (\mathbf{P}'\mathbf{C}) \times (\mathbf{P}'\mathbf{P}^+)$$

Finding the Fundamental Matrix from known Projections



- Take $\mathbf{x}$ in camera $\mathbf{P}$ and find one scene point on ray from $\mathbf{C}$ to $\mathbf{x}$
- Point $\mathbf{X} = \mathbf{P}^{+}\mathbf{x}$ satisfies $\mathbf{x} = \mathbf{P}\mathbf{X}$ with $\mathbf{P}^{+} = \mathbf{P}^{T}(\mathbf{P}\mathbf{P}^{T})^{-1}$ so $\mathbf{P}\mathbf{X} = \mathbf{P}\mathbf{P}^{T}(\mathbf{P}\mathbf{P}^{T})^{-1}\mathbf{x} = \mathbf{x}$
- Image of this point in camera $\mathbf{P}'$ is $\mathbf{x}' = \mathbf{P}'\mathbf{X} = \mathbf{P}'\mathbf{P}^{+}\mathbf{x}$
- Image of $\mathbf{C}$ in camera $\mathbf{P}'$ is epipole $\mathbf{e}' = \mathbf{P}'\mathbf{C}$
- Epipolar line of $\mathbf{x}$ in $\mathbf{P}'$ is

$$\mathbf{l}' = (\mathbf{e}') \times (\mathbf{P}'\mathbf{P}^{+}\mathbf{x})$$

- $\mathbf{l}' = \mathbf{F}\mathbf{x}$ defines the fundamental matrix

$$\mathbf{F} = (\mathbf{P}'\mathbf{C}) \times (\mathbf{P}'\mathbf{P}^{+})$$

Properties of the fundamental matrix

- Matrix 3×3 since $\mathbf{x'}^T \mathbf{F} \mathbf{x} = 0$
- Let $\mathbf{F}$ be the fundamental matrix of camera pair $(\mathbf{P}, \mathbf{P}')$, the fundamental matrix of camera pair $(\mathbf{P}', \mathbf{P})$ is $\mathbf{F}' = F^T$

Properties of the fundamental matrix

- Matrix 3×3 since $\mathbf{x}'^T \mathbf{F}\mathbf{x} = 0$
- Let $\mathbf{F}$ be the fundamental matrix of camera pair $(\mathbf{P}, \mathbf{P}')$, the fundamental matrix of camera pair $(\mathbf{P}', \mathbf{P})$ is $\mathbf{F}' = F^T$
- This is true since $\mathbf{x}^T \mathbf{F}'\mathbf{x}' = 0$ implies $\mathbf{x}'^T \mathbf{F}'^T \mathbf{x} = 0$, so $\mathbf{F}' = \mathbf{F}^T$

Properties of the fundamental matrix

- Matrix 3×3 since $\mathbf{x'}^T \mathbf{F} \mathbf{x} = 0$
- Let $\mathbf{F}$ be the fundamental matrix of camera pair $(\mathbf{P}, \mathbf{P'})$, the fundamental matrix of camera pair $(\mathbf{P'}, \mathbf{P})$ is $\mathbf{F'} = F^T$
- This is true since $\mathbf{x}^T \mathbf{F'} \mathbf{x'} = 0$ implies $\mathbf{x'}^T \mathbf{F'}^T \mathbf{x} = 0$, so $\mathbf{F'} = \mathbf{F}^T$
- Epipolar line of $\mathbf{x}$ is $\mathbf{l'} = \mathbf{F} \mathbf{x}$

Properties of the fundamental matrix

- Matrix 3×3 since $\mathbf{x'}^T \mathbf{F} \mathbf{x} = 0$
- Let $\mathbf{F}$ be the fundamental matrix of camera pair $(\mathbf{P}, \mathbf{P'})$, the fundamental matrix of camera pair $(\mathbf{P'}, \mathbf{P})$ is $\mathbf{F'} = F^T$
- This is true since $\mathbf{x}^T \mathbf{F'} \mathbf{x'} = 0$ implies $\mathbf{x'}^T \mathbf{F'}^T \mathbf{x} = 0$, so $\mathbf{F'} = \mathbf{F}^T$
- Epipolar line of $\mathbf{x}$ is $\mathbf{l'} = \mathbf{F}\mathbf{x}$
- Epipolar line of $\mathbf{x'}$ is $\mathbf{l} = \mathbf{F}^T \mathbf{x'}$

Properties of the fundamental matrix

- Matrix 3×3 since $\mathbf{x}'^T \mathbf{F} \mathbf{x} = 0$
- Let $\mathbf{F}$ be the fundamental matrix of camera pair $(\mathbf{P}, \mathbf{P}')$, the fundamental matrix of camera pair $(\mathbf{P}', \mathbf{P})$ is $\mathbf{F}' = F^T$
- This is true since $\mathbf{x}^T \mathbf{F}' \mathbf{x}' = 0$ implies $\mathbf{x}'^T \mathbf{F}'^T \mathbf{x} = 0$, so $\mathbf{F}' = \mathbf{F}^T$
- Epipolar line of $\mathbf{x}$ is $\mathbf{l}' = \mathbf{F}\mathbf{x}$
- Epipolar line of $\mathbf{x}'$ is $\mathbf{l} = \mathbf{F}^T \mathbf{x}'$
- Epipole $\mathbf{e}'$ is left null space of $\mathbf{F}$, and $\mathbf{e}$ is right null space.

Properties of the fundamental matrix

- Matrix 3×3 since $\mathbf{x'}^T \mathbf{F} \mathbf{x} = 0$
- Let $\mathbf{F}$ be the fundamental matrix of camera pair $(\mathbf{P}, \mathbf{P'})$, the fundamental matrix of camera pair $(\mathbf{P'}, \mathbf{P})$ is $\mathbf{F'} = F^T$
- This is true since $\mathbf{x}^T \mathbf{F'} \mathbf{x'} = 0$ implies $\mathbf{x'}^T \mathbf{F'}^T \mathbf{x} = 0$, so $\mathbf{F'} = \mathbf{F}^T$
- Epipolar line of $\mathbf{x}$ is $\mathbf{l'} = \mathbf{F}\mathbf{x}$
- Epipolar line of $\mathbf{x'}$ is $\mathbf{l} = \mathbf{F}^T \mathbf{x'}$
- Epipole $\mathbf{e'}$ is left null space of $\mathbf{F}$, and $\mathbf{e}$ is right null space.
- All epipolar lines $\mathbf{l'}$ contains epipole e', so $\mathbf{e'}^T \mathbf{l'} = 0$, i.e. $\mathbf{e'}^T \mathbf{F}\mathbf{x} = 0$ for all $\mathbf{x}$, therefore $\mathbf{e'}^T \mathbf{F} = 0$

Properties of the fundamental matrix

- Matrix 3×3 since $\mathbf{x}'^T \mathbf{F} \mathbf{x} = 0$
- Let $\mathbf{F}$ be the fundamental matrix of camera pair $(\mathbf{P}, \mathbf{P}')$, the fundamental matrix of camera pair $(\mathbf{P}', \mathbf{P})$ is $\mathbf{F}' = F^T$
- This is true since $\mathbf{x}^T \mathbf{F}' \mathbf{x}' = 0$ implies $\mathbf{x}'^T \mathbf{F}'^T \mathbf{x} = 0$, so $\mathbf{F}' = \mathbf{F}^T$
- Epipolar line of $\mathbf{x}$ is $\mathbf{l}' = \mathbf{F}\mathbf{x}$
- Epipolar line of $\mathbf{x}'$ is $\mathbf{l} = \mathbf{F}^T \mathbf{x}'$
- Epipole $\mathbf{e}'$ is left null space of $\mathbf{F}$, and $\mathbf{e}$ is right null space.
- All epipolar lines $\mathbf{l}'$ contains epipole e', so $\mathbf{e}'^T \mathbf{l}' = 0$, i.e. $\mathbf{e}'^T \mathbf{F} \mathbf{x} = 0$ for all $\mathbf{x}$, therefore $\mathbf{e}'^T \mathbf{F} = 0$
- $\mathbf{F}$ is of rank 2 because $\mathbf{F} = \mathbf{e}' \times (\mathbf{P}' \mathbf{P}^+)$ and $\mathbf{e}' \times$ is of rank 2

Properties of the fundamental matrix

- Matrix 3×3 since $\mathbf{x}'^T \mathbf{F} \mathbf{x} = 0$
- Let $\mathbf{F}$ be the fundamental matrix of camera pair $(\mathbf{P}, \mathbf{P}')$, the fundamental matrix of camera pair $(\mathbf{P}', \mathbf{P})$ is $\mathbf{F}' = F^T$
- This is true since $\mathbf{x}^T \mathbf{F}' \mathbf{x}' = 0$ implies $\mathbf{x}'^T \mathbf{F}'^T \mathbf{x} = 0$, so $\mathbf{F}' = \mathbf{F}^T$
- Epipolar line of $\mathbf{x}$ is $\mathbf{l}' = \mathbf{F}\mathbf{x}$
- Epipolar line of $\mathbf{x}'$ is $\mathbf{l} = \mathbf{F}^T \mathbf{x}'$
- Epipole $\mathbf{e}'$ is left null space of $\mathbf{F}$, and $\mathbf{e}$ is right null space.
- All epipolar lines $\mathbf{l}'$ contains epipole e', so $\mathbf{e}'^T \mathbf{l}' = 0$, i.e. $\mathbf{e}'^T \mathbf{F} \mathbf{x} = 0$ for all $\mathbf{x}$, therefore $\mathbf{e}'^T \mathbf{F} = 0$
- $\mathbf{F}$ is of rank 2 because $\mathbf{F} = \mathbf{e}' \times (\mathbf{P}' \mathbf{P}^+)$ and $\mathbf{e}' \times$ is of rank 2
- $\mathbf{F}$ has 7 degrees of freedom, there are 9 elements, but scaling is not important and $det(\mathbf{F}) = 0$ removes one degree of freedom

Properties of the fundamental matrix

- Matrix 3×3 since $\mathbf{x'}^T \mathbf{F} \mathbf{x} = 0$
- Let $\mathbf{F}$ be the fundamental matrix of camera pair $(\mathbf{P}, \mathbf{P'})$, the fundamental matrix of camera pair $(\mathbf{P'}, \mathbf{P})$ is $\mathbf{F'} = F^T$
- This is true since $\mathbf{x}^T \mathbf{F'} \mathbf{x'} = 0$ implies $\mathbf{x'}^T \mathbf{F'}^T \mathbf{x} = 0$, so $\mathbf{F'} = \mathbf{F}^T$
- Epipolar line of $\mathbf{x}$ is $\mathbf{l'} = \mathbf{F}\mathbf{x}$
- Epipolar line of $\mathbf{x'}$ is $\mathbf{l} = \mathbf{F}^T \mathbf{x'}$
- Epipole $\mathbf{e'}$ is left null space of $\mathbf{F}$, and $\mathbf{e}$ is right null space.
- All epipolar lines $\mathbf{l'}$ contains epipole e', so $\mathbf{e'}^T \mathbf{l'} = 0$, i.e. $\mathbf{e'}^T \mathbf{F}\mathbf{x} = 0$ for all $\mathbf{x}$, therefore $\mathbf{e'}^T \mathbf{F} = 0$
- $\mathbf{F}$ is of rank 2 because $\mathbf{F} = \mathbf{e'} \times (\mathbf{P'}\mathbf{P}^+)$ and $\mathbf{e'}\times$ is of rank 2
- $\mathbf{F}$ has 7 degrees of freedom, there are 9 elements, but scaling is not important and $det(\mathbf{F}) = 0$ removes one degree of freedom