Robotics - Single view, Epipolar geometry, Image Features

Matteo Pirotta

pirotta@elet.polimi.it

Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano

2 April 2013

Inspired from Simone Ceriani’s slides (Robotics @ Como 2012)
Outline

1. Pin Hole Model
2. Distortion
3. Camera Calibration
4. Two views geometry
5. Image features
6. Edge, corners
7. Exercise
Outline

1. Pin Hole Model
2. Distortion
3. Camera Calibration
4. Two views geometry
5. Image features
6. Edge, corners
7. Exercise
Pin hole model - Recall

The intrinsic camera matrix

or calibration matrix

\[ K = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \]

- \( f_x, f_y \): focal length (in pixels)
  - \( f_x / f_y = s_x / s_y = a \): aspect ratio
- \( s \): skew factor
  - pixel not orthogonal
    - usually 0 in modern cameras
- \( c_x, c_y \): principal point (in pixel)
  - usually \( \neq \) half image size due to misalignment of CCD

Projection

\[
\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{bmatrix} f_x & 0 \\ 0 & f_y \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + \begin{pmatrix} c_x \\ c_y \end{pmatrix} \]
Points in the world

**Consider**

- \( p^{(I')} = \begin{bmatrix} f_x & s & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} P^{(O)} \)

- \( P^{(O)} = T_{OW}^{(O)} P^{(W)} \)

**One step**

- \( p^{(I')} = \begin{bmatrix} K & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} P^{(W)} \)

- \( \pi = \begin{bmatrix} KR & Kt \end{bmatrix} \)

**Note**

- \( R \) is \( R_{OW}^{(O)} \)
- \( t \) is \( t_{OW}^{(O)} \)

i.e., the position and orientation of \( W \) in \( O \)
**Note on camera reference system**

**Camera reference system**
- $z$: front
- $y$: down

**World reference system**
- $x$: front
- $z$: up

**Rotation of $O$ w.r.t. $W$**
- Rotate around $y$ of $90^\circ$
  - $z'$ front
- Rotate around $z'$ of $-90^\circ$
  - $y''$ point down

\[
R_{WO}^{(W)} = R_y(90^\circ)\ R_z(-90^\circ)
\]

\[
R_{WO}^{(W)} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \\
\begin{bmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}
\]

\[
R_{OW}^{(O)} = R_{WO}^{(W)^T} = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{bmatrix}
\]
**Interpretation line**

**Given**
- \( \mathbf{p}^{(I)} = \begin{bmatrix} u, v \end{bmatrix}^T \) : point in image (pixel)

**Calculate \( \mathbf{P}^{(O)} \)?**
- No, only \( \mathbf{l}_{Po} \): interpretation line
- \( \forall \mathbf{P}_i^{(O)} \in \mathbf{l}_{Po} \) image is \( \mathbf{p}^{(I)} \)

**Calculate \( \mathbf{l}_{Po} \)**
- 3D lines not coded in 3D
  - remember duality points \( \leftrightarrow \) planes
- \( \mathbf{p}^{(I')} = [K \ 0] \mathbf{P}^{(O)} \)
- \( \mathbf{p}^{(I')} = [K \ 0] [X, Y, Z, W]^T \)
- \( \mathbf{p}^{(I')} = K [X, Y, Z]^T \)
  - \( W \) "cancelled" by zeros fourth column
- \( \mathbf{d}^{(O)} = K^{-1} [u, v, 1]^T \)
- \( \bar{\mathbf{d}}^{(O)} = \mathbf{d}^{(O)}/\| \mathbf{d}^{(O)} \| \) : unit vector
- \( \mathbf{P}_i^{(O)} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} + \lambda \begin{bmatrix} \bar{\mathbf{d}}^{(O)} \\ 0 \end{bmatrix}, \lambda > 0 \)
- \( \mathbf{l}_{Po} \) in parametric form
Interpretation line & Normalized image plane

**INTERPRETATION LINE DIRECTION**

- \( \mathbf{d}^{(O)} = \mathbf{K}^{-1} \begin{bmatrix} u, v, 1 \end{bmatrix}^T \)
- \( \mathbf{K}^{-1} = \begin{bmatrix} 1/f_x & 0 & -c_x/fx & 0 \\ 0 & 1/f_y & -c_y/fy & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \)

Assume skew = 0

- \( \mathbf{d}^{(O)} = \begin{bmatrix} u-c_x/fx, \, v-c_y/fy, \, 1 \end{bmatrix}^T \)

- \( \mathbf{P}^{(O)}_{\lambda=\|\mathbf{d}^{(O)}\|} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} \mathbf{d}^{(O)} \end{bmatrix} \)

lies on \( \pi_N \)

- If \( u = c_x, \, v = c_y, \, \mathbf{d} \) is
  - the *principal direction*

**NORMALIZED IMAGE PLANE**

- Distance 1 from the optical center
- Independent of camera intrinsic

Given a cartesian point \( \mathbf{P}^{(O)} = \begin{bmatrix} X, \, Y, \, Z \end{bmatrix}^T \)

\[
\mathbf{P}_{\pi_N}^{(O)} = \begin{bmatrix} X/Z, \, Y/Z, \, 1 \end{bmatrix}^T
\]
**General Projective Camera**

A general projective camera $\pi$ maps world points $p^{(W)}$ to image points $p^{(I')}$ according to

$$p^{(I')} = \pi p^{(W)}$$

A general projective camera may be decomposed into

$$\pi = \begin{bmatrix} M & m \end{bmatrix}$$

If $M$, a $(3 \times 3)$ matrix, is non–singular, $\pi$ represents a *finite camera model*

$$M = KR, \quad m = Kt$$

$$\pi = \begin{bmatrix} KR & Kt \end{bmatrix} = KR \begin{bmatrix} I & -C \end{bmatrix}$$

where $C$ is the camera centre in world coordinates ($O^{(W)}$)

$$C = -M^{-1}m$$
Consider

- \( \mathbf{X}^{(W)} = \text{RNS} (\pi) \rightarrow \pi \mathbf{X}^{(W)} = 0 \)
- a line through \( \mathbf{A}^{(W)} \) and \( \mathbf{X}^{(W)} \)

\[
\begin{align*}
\mathbf{B}_{\lambda}^{(W)} &= \lambda \mathbf{A}^{(W)} + (1 - \lambda) \mathbf{X}^{(W)} \\
\mathbf{B}_{\lambda}^{(l')} &= \pi \mathbf{B}_{\lambda}^{(W)} \\
&= \lambda \pi \mathbf{A}^{(W)} + (1 - \lambda) \pi \mathbf{X}^{(W)} = \lambda \pi \mathbf{A}^{(W)}
\end{align*}
\]

- All the points on the line are mapped on the same image point \( \lambda \pi \mathbf{A}^{(W)} \)
- Then, \( \mathbf{X}^{(W)} \) is the camera centre (\( \mathbf{X}^{(W)} = \mathbf{O}^{(W)} \))
**Forward Projection**

- Consider $D^{(W)} = (d^T, 0)^T$ (direction, point at infinity)
  
  $$p^{(I')} = \pi D^{(W)} = \begin{bmatrix} M & m \end{bmatrix} D^{(W)} = Md$$
  
  $$d = M^{-1} p^{(I')}$$

- In the case of finite camera
  
  $$d = R^{-1} K^{-1} p^{(I')} = R^T K^{-1} p^{(I')}$$
**Interpretation line in the world - 4**

**Back-Projection – from point to ray**

- Given a point \( p^{(I')} \) in an image, the ray back-projection is \( \tilde{\pi} p^{(I')} = p^{(W)} \)
- The pseudo inverse of \( \pi \) is \( \pi^+ = \pi^T (\pi \pi^T)^{-1} \), for which \( \pi \pi^+ = I \)
- We know the point \( p^{(I')} \) and the camera centre \( O^{(W)} \)
- The ray is the line \( X_\lambda = \pi^+ p^{(I')} + \lambda O_h^{(W)} \)
- In the case of finite camera, an image point back-projcts to a ray intersecting the plane at infinity at the point \( D = (d^T, 0)^T \), and \( D \) provides a second point on the ray. The interpretation line in world coordinates is

\[
p^{(W)}_\mu = O_h^{(W)} + \mu \begin{bmatrix} d \\ 0 \end{bmatrix} = \begin{bmatrix} -M^{-1}m \\ 1 \end{bmatrix} + \mu \begin{bmatrix} M^{-1}p^{(I')} \\ 0 \end{bmatrix}
\]
Interpretation line in the world - an other way

Consider

- Interpretation line in camera coordinate

  \[ P_i^{(O)} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \lambda \begin{bmatrix} \bar{d}^{(O)} \\ 0 \end{bmatrix} \]

- Interpretation line in world coordinate

  \[
  P_i^{(W)} = \begin{bmatrix} R_{OW}^T \\ 0 \end{bmatrix} - R_{OW}^T t_{OW} \left( \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \lambda \begin{bmatrix} \bar{d}^{(O)} \\ 0 \end{bmatrix} \right) \\
  = \begin{bmatrix} -R_{OW}^T t_{OW} \\ 1 \end{bmatrix} + \lambda R_{OW}^T \bar{d}^{(O)} \\
  = O^{(W)} + \lambda \bar{d}^{(W)}
  \]

- Camera center in world coordinate + direction rotated as world reference
Consider

- Interpretation line in world coordinate

\[
P_i^{(W)} = \lambda R^{(O)}_{OW} T d^{(O)} - R^{(O)}_{OW} T t^{(O)}_{OW} = \lambda R^{(O)}_{OW} T K^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} - R^{(O)}_{OW} T t^{(O)}_{OW}
\]

- Complete projection matrix

\[
\pi = \begin{bmatrix} KR^{(O)}_{OW} \\ Kt^{(O)}_{OW} \end{bmatrix} = [M \ m]
\]

- \( M^{-1} = R^{(O)}_{OW} T K^{-1} \): direction in world coordinate given pixel coordinate

- \(-M^{-1} m = -R^{(O)}_{OW} T K^{-1} K t^{(O)}_{OW} = -R^{(O)}_{OW} T t^{(O)}_{OW} = t^{(W)}_{WO} \):

  camera center in world coordinate \( O^{(W)} \)
Principal ray

**INTERPRETATION LINE OF PRINCIPAL POINT**

\[ d^{(O)} = K^{-1} \begin{bmatrix} c_x \\ c_y \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \]

z-axis of the camera reference system

\[ d^{(W)} = R^{(O)}_{OW} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = M^{-1} \begin{bmatrix} c_x \\ c_y \\ 1 \end{bmatrix} \]

z-axis of the camera in world reference system

\[ P_i^{(W)} = \lambda M^{-1} \begin{bmatrix} c_x \\ c_y \\ 1 \end{bmatrix} - M^{-1} m \]

parametric line of z-axis of the camera in world reference system
Vanishing points & Origin

**Vanishing points**

- \( \mathbf{V}_x^{(W)} = [1, 0, 0, 0]^T \)
- \( \mathbf{V}_y^{(W)} = [0, 1, 0, 0]^T \)
- \( \mathbf{V}_z^{(W)} = [0, 0, 1, 0]^T \)

**Projection on the image**

- \( \mathbf{p}^{(I')} = [M \ m]\begin{bmatrix} d \\ 0 \end{bmatrix} = Md \)
- \( \mathbf{p}_x^{(I')} = [M \ m] \mathbf{V}_x^{(W)} = M_1 \)
- \( \mathbf{p}_y^{(I')} = [M \ m] \mathbf{V}_y^{(W)} = M_2 \)
- \( \mathbf{p}_z^{(I')} = [M \ m] \mathbf{V}_z^{(W)} = M_3 \)

**Origin**

- \( \mathbf{O}^{(W)} = [0, 0, 0, 1]^T \)

**Projection on the image**

- \( \mathbf{p}^{(I')} = [M \ m] \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \mathbf{m} \)

**Note**

- \( \pi = \begin{bmatrix} M_1 & M_2 & M_3 & \mathbf{m} \end{bmatrix} \)
  
  - Col 1 of \( \pi (M_1) \) is image of \( x \) vanishing point
  - Col 2 of \( \pi (M_2) \) is image of \( y \) vanishing point
  - Col 3 of \( \pi (M_3) \) is image of \( w \) vanishing point
  - Col 4 of \( \pi (\mathbf{m}) \) is image of \( \mathbf{O}^{(W)} \)
Angle of View

**Given**

- Image size: \([w, h]\)
- Focal length: \(f_x\) (assume \(f_x = f_y\))

**Angle of view**

- \(\theta = 2\arctan2(w/2, f_x)\)
- \(\theta < 180^\circ\)

**Examples**

- 14mm
- 20mm
- 28mm
- 35mm
- 50mm
**Problem**

- Given \( \pi = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 3 & 1 & 0 & 4 \\ 1 & 2 & 3 & 1 \end{bmatrix} \)
- Where is the camera center in world reference frame?

**Solution**

\[ \pi = \begin{bmatrix} M & m \end{bmatrix} \]

\[ O^{(W)} = -M^{-1}m \]

\[ M^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ -3 & 1 & 0 \\ 5/3 & -2/3 & 1/3 \end{bmatrix} \]

\[ O^{(W)} = \begin{bmatrix} -1 \\ -1 \\ 2/3 \end{bmatrix} \]
Outline

1. Pin Hole Model
2. Distortion
3. Camera Calibration
4. Two views geometry
5. Image features
6. Edge, corners
7. Exercise
DISTORTION

- Deviation from rectilinear projection
- Lines in scene don’t remains lines in image

ORIGINAL IMAGE

CORRECTED
**Distortion**

- Can be irregular
- Most common is *radial* (radially symmetric)

**Radial Distortion**

**Barrel distortion**
- Magnification decrease with distance from optical axis

**Pincushion distortion**
- Magnification increase with distance from optical axis
Consider

- $\mathbf{P}^{(O)} = [X, Y, Z]^T$ in camera reference system
- Calculate $\mathbf{p}^{(I)} = [x, y, 1]^T = [X/Z, Y/Z, 1]^T$ on the normalized image plane

Distortion model

- $\tilde{\mathbf{p}}^{(I)} = (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) \mathbf{p}^{(I)} + d_x$
- $r^2 = x^2 + y^2$: distance wrt optical axis $(0, 0)$
- $d_x = \begin{bmatrix} 2p_1 xy + p_2(r^2 + 2x^2) \\ p_1(r^2 + 2y^2) + 2p_2 xy \end{bmatrix}$: tangential distortion compensation

Image coordinate

- $\tilde{\mathbf{p}}^{(I')} = K\tilde{\mathbf{p}}^{(I)}$: pixel coordinate of $\mathbf{P}^{(O)}$ considering distortion
Brown distortion model

**From image points**

- $\tilde{p}^{(I')}$ in image (pixel)
- Calculate $\tilde{p}^{(I)} = K^{-1}\tilde{p}^{(I')} = [x, y, 1]^T$ on the (distorted) normalized image plane
- Undistort: $p^{(I)} = dist^{-1}(\tilde{p}^{(I)})$
- Image projection: $p^{(I')} = Kp^{(I)}$

**Evaluation of $dist^{-1}(\cdot)$**

- No analytic solution
- Iterative solution ($N = 20$ is enough):
  1. $p^{(I)} = \tilde{p}^{(I)}$: initial guess
  2. for $i = 1$ to $N$ do
  3. $r^2 = x^2 + y^2$, $k_r = (1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$, $d_x = \begin{bmatrix} 2p_1 xy + p_2 (r^2 + 2x^2) \\ p_1 (r^2 + 2y^2) + 2p_2 xy \end{bmatrix}$
  4. $p^{(I)} = (\tilde{p}^{(I)} - d_x) / k_r$
  5. end for
Outline

1. Pin Hole Model
2. Distortion
3. Camera Calibration
4. Two views geometry
5. Image features
6. Edge, corners
7. Exercise
Camera calibration

**INTRINSIC CALIBRATION**

- Find parameters of $K$
- Nominal values of optics are not suitable
- Differences between different exemplar of same camera/optic system
- Include distortion coefficient estimation

**EXTRINSIC CALIBRATION**

- Find parameters of $\pi = \begin{bmatrix} M & m \end{bmatrix}$
- i.e., find $K$ and $R, t$
Camera calibration - Approaches

**CALIBRATION**

- Very large literature!
- Different approaches

**KNOWN 3D PATTERN**

**METHODS**

- Based on correspondances
- Need for a pattern

**PLANAR PATTERN**
Camera calibration - Formulation

FORMULATION

- \( \mathbf{M}_i \): model points on the pattern
- \( \mathbf{p}_{ij} \): observation of model point \( i \) in image \( j \)
- \( \mathbf{p} = \begin{bmatrix} f_x, f_y, s, \cdots k_1, k_2, \cdots \end{bmatrix}^T \): intrinsic parameters
- \( \mathbf{R}_j, \mathbf{t}_j \): pose of the pattern wrt camera reference frame \( j \)
  - i.e., \( \mathbf{R}_{CP}^{(C)}, \mathbf{t}_{CP}^{(C)} \)
- \( \hat{\mathbf{m}}(\mathbf{p}, \mathbf{R}_j, \mathbf{t}_j, \mathbf{M}_i) \): estimated projection of \( \mathbf{M}_i \) in image \( j \).

ESTIMATION

- \( \arg \min_{\mathbf{p}, \mathbf{R}_j, \mathbf{t}_j} \sum_j \sum_i \mathbf{p}_{ij} - \hat{\mathbf{m}}_{ij}(\mathbf{p}, \mathbf{R}_j, \mathbf{t}_j, \mathbf{M}_i) \): observation of model point \( i \) in image \( j \)
- Gives both intrinsic (unique) and extrinsic (one for each image) calibration

Z. Zhang, “A flexible new technique for camera calibration”, 2000
Camera Calibration Toolbox for Matlab

COLLECT IMAGES

AUTOMATIC CORNERS IDENTIFICATION

FIND CHESSBOARD EXTERNAL CORNERS

CALIBRATION

Camera Calibration Toolbox for Matlab - http://www.vision.caltech.edu/bouguetj/calib_doc/
Outline

1. Pin Hole Model
2. Distortion
3. Camera Calibration
4. Two views geometry
5. Image features
6. Edge, corners
7. Exercise
Epipolar geometry introduction

**Epipolar geometry**

- Projective geometry between two views
- **Independent** of scene structure
- Depends **only** on
  - Cameras parameters
  - Cameras relative position
- Two views
  - Simultaneously (stereo)
  - Sequentially (moving camera)
- The geometry of the intersection of image planes with the pencil of planes having the base line as axis (base line: the line joining the camera centres, see next)

Bumblebee camera

Robot head with two cameras
Correspondence

Consider

- **P** a 3D point in the scene
- Two cameras with **π** and **π′**
- **p** = **π**P image on first camera
- **p′** = **π′**P image on second camera
- **p** and **p′**: images of the same point → correspondence

Questions

1. Correspondence geometry
   - **p** on first image
   - How **p′** is constrained by **p**?
2. Camera geometry (motion)
   - Given a set of correspondances \( \{p_i \leftrightarrow p'_i\}_{i=1}^n \)
   - what are the cameras **π′**, **π**?
3. Scene geometry (structure)
   - Given corresponding image points **p** ↔ **p′**
   - what is the position of **P** in 3D–space?
**Correspondences and epipolar geometry**

**Suppose (1)**

- \( P \), a 3D point imaged in two views
- \( p_L \) and \( p_R \) image of \( P \)
- \( P, p_L, p_R, O_L, O_R \) are coplanar on \( \pi \)
- \( \pi \) is the *epipolar plane*

**Suppose (2)**

- \( P \) is unknown
- \( p_L \) is known
- Where is \( p_R \)?
  - or how is constrained \( p_R \)?
- \( P_i = O_L + \lambda d_{p_L O_L} \) is the interpretation line of \( p_L \)
- \( p_R \) lies on a line:
  - intersection of \( \pi \) with the 2\textsuperscript{nd} image
  \( \rightarrow \) *epipolar line*
Epipolar geometry - Definitions

**Base line**
- Line joining $O_L$ and $O_R$

**Epipoles ($e_L, e_R$)**
- Intersection of base line with image planes
- Projection of camera centres on images
- Intersection of all epipolar lines

**Epipolar line**
- Intersection of epipolar plane with image plane

**Epipolar plane**
- A plane containing the baseline
- It’s a pencil of planes
- Given an *epipolar line* is possible to identify a unique epipolar plane
Epipolar constraints

Correspondences problem

- Given $p_L$ in one image

- Search on second image along the *epipolar line*: *1D search!*

- A point in one image “generates” a line in the second image, that is, for each point $p_L$ in one image exists a corresponding epipolar line $l_R$ in the other image

- Any point $p_R$ in the second image matching the point $p_L$ must lie on the epipolar line $l_R$: $p_L \leftrightarrow l_R$

Correspondences example
The fundamental matrix $F$ – Geometric derivation

- Two steps process:
  - The point $p_L$ is mapped to some point $p_R$ in the other image lying on the epipolar line $l_R$
  - The epipolar line $l_R$ is obtained as the line joining $p_R$ to the epipole $e_R$

**Epipolar geometry**

- Given $p_L$ on one image
- $p_R$ lies on $l_R$, i.e. the epipolar line
- $p_R \in l_R \iff p_R^T l_R = 0$
- Thus, there is a map $p_L \rightarrow l_R$
EPIPOLAR GEOMETRY

- \( I_R = e_R \times p_R = [e_R]_\times p_R \)
- Since \( p_R = H_\pi p_L \)
- \( I_R = [e_R] \times H_\pi p_L = Fp_L \)
- \( F = [e_R] \times H_\pi \) fundamental matrix
  - \( H_\pi \) a rototranslation
  - \( a \times b = [a]_\times b \) where
    \[
    [a]_\times = \begin{bmatrix}
    0 & -a_3 & a_2 \\
    a_3 & 0 & -a_1 \\
    -a_2 & a_1 & 0
    \end{bmatrix}
    \]
The fundamental matrix $F$ – Algebraic derivation

- Given a point $p_L$ in an image, the ray back-projection is $\pi_L^+ p_L = P$
- The pseudo inverse of $\pi_L$ is $\pi_L^+ = \pi_L^T (\pi_L \pi_L^T)^{-1}$, for which $\pi_L \pi_L^+ = I$
- We know the point $p_L$ and the camera centre $O_L$ ($\pi_L O_L = 0$)
- The ray is $P_\lambda = \pi_L^+ p_L + \lambda O_L$
We have

- $\pi^+ p_L$ and $O_L$ for the first camera $\pi_L$
- The image of these two points for the second camera $\pi_R (\pi_R\pi^+_L p_L$ and $\pi_R O_L)$

The epipolar line joining the two projected points is

$$l_R = (\pi_R O_L) \times (\pi_R\pi^+_L p_L)$$

$$= [e_R] \times \pi_R\pi^+_L p_L$$

Note: does not work with $O_L = O_R$, then $F = 0$
From calibrated cameras

- $\pi_L$ and $\pi_R$ are known
- $F = [e_R] \times \pi_R \pi_L^+$
  - where
    - $e_R = \pi_R O_L = \pi_R \left( -M_L^{-1} m_L \right)$
    - $\pi_L^+ = \pi_L^T (\pi_L \pi_L^T)^{-1}$: pseudo-inverse
    - $a \times b = [a] \times b$ where $[a] \times = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$

Calibrated cameras with $K$ and $R, t$

- $\pi_L = K_L \begin{bmatrix} I & 0 \end{bmatrix}$: origin in the left camera
- $\pi_R = K_R \begin{bmatrix} R & t \end{bmatrix}$
- $F = K_R^{-T} [t] \times R K_L^{-1}$
- Special forms with pure translations, pure rotations, ...
The fundamental matrix $F$ properties

**Properties**

- If $p_L$ correspond to $p_R$ $\rightarrow p_L F p_R = 0$, necessary condition for correspondence
- If $p_L F p_R = 0$ interpretation lines (a.k.a. viewing ray) are coplanar
- $F$ is a 3x3 matrix
- $\det(F) = 0$
- $\text{rank}(F) = 2$
- $F$ has 7 dof (1 homogeneous, 1 rank deficient)
- $l_R = F p_L$, $l_L = F^T p_R$
- $Fe_L = 0$, $F^T e_R = 0$, i.e.: epipoles are the right null vector of $F$ and $F^T$

**Proof:** $\forall p_L \neq e_L$, $l_R = F p_L$ and $e_R \in l_R$ $\rightarrow \forall p_L$ $e_R^T F p_L = 0$ $\rightarrow$ $F^T e_R = 0$
**F estimation - procedure sketch**

FROM UNCALIBRATED IMAGES

- Get point correspondances (“by hand” or automatically)
- Compute $F$ by consider that
  - $p_R^T F p_L = 0$
  - At least 7 correspondances are needed but the 8-point algorithm is the simplest
- Impose $\text{rank}(F) = 2$

Details

"Multiple View Geometry in computer vision"

Hartley Zisserman. Chapters 9,10,11,12.
**F is not unique**

- If estimated by correspondences

- Without any additional constraints allow at least a projective reconstruction
Metric reconstruction and reconstruction

**ADDITIONAL CONSTRAINTS**

- Parallelism, measures of some points, ...

- → allow affine/similar/metric reconstruction
Suppose

- $p_R$ and $p_L$ are correspondent points
- $\pi_L$ and $\pi_R$ are known
- Due to noise is possible that interpretation lines don’t intersect
- $p_L F p_R \neq 0$

3D Point Computation

- $\arg\min_{\hat{p}_L, \hat{p}_R} d(\hat{p}_L, p_L)^2 + d(\hat{p}_R, p_R)^2$
- subject to $\hat{p}_L F \hat{p}_R = 0$
Features in image

What is a feature?

- No common definition
- Depends on problem or application
- Is an *interesting part* of the image

Types of features

- Edges
  - Boundary between regions
- Corners / interest points
  - Edge intersection
  - Corners
  - Point-like features
- Blobs
  - Smooth areas that define regions
**Thresholding**

- On a gray scale image $I(u, v)$
- If $I(u, v) > T$  $I'(u, v) = \text{white}$
- else $I'(u, v) = \text{black}$

**Properties**

- Simplest method of image segmentation
- Critical point: threshold $T$ value
  - Mean value of $I(u, v)$
  - Median value of $I(u, v)$
  - (Local) adaptive thresholding
**KERNEL MATRIX FILTERING**

- **Given an image** \( I(i, j), i = 1 \cdots h, j = 1 \cdots w \)
- **A kernel** \( H(k, z), k = 1 \cdots r, z = 1 \cdots c \)
- \( I'(i, j) = \sum_k \sum_z I(i - \lfloor r/2 \rfloor + k - 1, j - \lfloor c/2 \rfloor + z - 1) \ast H(k, z) \)
- special cases on borders

\[
\begin{array}{cccc}
I(i, j) & H(k, z) & I'(2, 2) \\
2 & 2 & 2 & 3 \\
2 & 1 & 3 & 3 \\
2 & 2 & 1 & 2 \\
1 & 3 & 2 & 2 \\
\end{array}
\begin{array}{cccc}
1 & 1 & 1 \\
-1 & 2 & 1 \\
-1 & -1 & 1 \\
\end{array}
\begin{array}{cccc}
5 & 4 & 4 & -2 \\
9 & 6 & 14 & 5 \\
11 & 7 & 6 & 5 \\
9 & 12 & 8 & 5 \\
\end{array}
\]

**Final result**
Filter Examples - 1

**Identity**

\[
\begin{bmatrix}
0 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 0
\end{bmatrix}
\]

**Translation**

\[
\begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & 1 \\
0 & 0 & 0
\end{bmatrix}
\]
Filter Examples - 2

**Average (3 × 3)**

\[
\begin{array}{ccc}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1 \\
\end{array}
\]

\[
\frac{1}{9}
\]

\[
= \\
\begin{array}{ccc}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1 \\
\end{array}
\]

**Average (5 × 5)**

\[
\begin{array}{ccccc}
1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 \\
\end{array}
\]

\[
\frac{1}{25}
\]

\[
= \\
\begin{array}{ccccc}
1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 \\
\end{array}
\]
**Filter Examples - 3**

**Gaussian** - $\sim N(0, \sigma)$

**Gaussian vs Average**
**GENERALLY EXPECT**

- Pixels to “be like” neighbours
- Noise independent from pixel to pixel

**IMPLIED**

- Smoothing suppress noises
- Appropriate noise model (?)
**Image Gradient - 1**

**HORIZONTAL DERIVATIVES ($\nabla I_x$)**

![Image Gradient Example](image_url)

The image gradient is calculated using the horizontal derivatives ($\nabla I_x$). The horizontal derivatives are computed using a kernel:

$$
\begin{bmatrix}
1 & 0 & -1
\end{bmatrix}
\begin{bmatrix}
1 & 0 & 1 \\
0 & 0 & 0 \\
1 & 0 & 1
\end{bmatrix} =
$$

The result is a filtered image that highlights horizontal edges.
**Image Gradient - 2**

**Vertical derivatives** ($\nabla l_y$)

![Image Gradient Example](image.png)

Vertical derivatives are calculated using a small matrix kernel. The matrix

$$
\begin{pmatrix}
1 & 1 & 1 \\
0 & 0 & 0 \\
-1 & -1 & -1
\end{pmatrix}
$$

is applied to the image to compute the gradient in the vertical direction.
Rough edge detector

\[ \nabla I_x^2 + \nabla I_y^2 \]

then apply threshold...
Outline

1. Pin Hole Model
2. Distortion
3. Camera Calibration
4. Two views geometry
5. Image features
6. Edge, corners
7. Exercise
Canny Edge Detector

**Criterion**

- *Good detection*: minimize probability of false positive and false negatives
- *Good localization*: Edges as closest as possible to the true edge
- *Single response*: Only one point for each edge point (thick = 1)

**Procedure**

- Smooth by Gaussian \( S = I \ast G(\sigma) \)
- Compute derivatives \( \nabla S_x, \nabla S_y \)
  
  Alternative in one step: Filter with derivative of Gaussian
- Compute magnitude and orientation of gradient
  \[
  \| \nabla S_x \| = \sqrt{\nabla S_x^2 + \nabla S_y^2}, \quad \theta_{\nabla S} = \text{atan2} \nabla S_y, \nabla S_x
  \]
- Non maxima suppression
  Search for local maximum in the gradient direction \( \theta_{\nabla S} \)
- Hysteresis threshold
  Weak edges (between the two thresholds) are edges
  if connected to strong edges (greater than high threshold)
Canny Edge Detector - Non Maxima Suppression

**NON MAXIMA SUPPRESSION**

**EXAMPLE**

Original image

Gradient magnitude

Non-maxima suppressed
Hysteresis threshold example

gap is gone

Original image

Strong edges only

Strong + connected weak edges

Weak edges

courtesy of G. Loy
Lines from edges

**How to find lines?**

- *Hough Transformation* after Canny edges extraction
- Use a *voting procedure*
- Generally find imperfect instances of a shape class
- Classical Hough Transform for lines detection
- Later extended to other shapes (most common: circles, ellipses)

**Example**
Corners - Harris and Shi Tomasi

**Edge Intersection**
- At intersection point gradient is ill defined
- Around intersection point gradient changes in “all” directions
- It is a “good feature to track”

**Corner Detector**
- Examine gradient over window
- \[ C = \sum_w \sum \begin{bmatrix} \nabla I_x^2 & \nabla I_x \nabla I_y \\ \nabla I_x \nabla I_y & \nabla I_y^2 \end{bmatrix} \]
- Shi-Tomasi: corner if \( \min \text{eigenvalue}(C) > T \)
- Harris: approximation of eigenvalues
Template matching - Patch

Filtering with a template

- Correlation between template (patch) and image
- Maximum where template matches
- Alternatives with normalizations for illumination compensation, etc.

Good features

- On corners: higher repeatability (homogeneous zone and edges are not distinctive)
Template matching - SIFT

**Template matching issues**
- Rotations
- Scale change

**SIFT**
- Scale Invariant Feature Transform
- Alternatives descriptor to patch
- Performs orientation and scale normalization
- *See also SURF (Speeded Up Robust Feature)*

**SIFT example**

D. Lowe - “Object recognition from local scale-invariant features” - 1999
Other references

- The reader may refer to the slides of the course of Image Analysis for additional details
  
  http://home.deib.polimi.it/boracchi/teaching/IAS.htm

- A Matlab and Octave implementation of a lot of algorithms can be found at
  
  http://www.csse.uwa.edu.au/~pk/research/matlabfns/
Camera matrix - 1

**Given**

\[
P = \begin{bmatrix}
122.5671 & -320.0000 & -102.8460 & 587.3835 \\
-113.7667 & 0.0000 & -322.2687 & 350.6050 \\
0.7660 & 0 & -0.6428 & 4.6711
\end{bmatrix}
\]

- \( f_x = f_y = 320 \)
- \( c_x = 160 \)
- \( c_y = 120 \)

**Questions**

- Where is the camera in the world?
- Compute the coordinate of the vanishing point of \( x, y \) plane in the image
- Where is the origin of the world in the image?
- Write the parametric 3D line of the principal axis in world coordinates
- ...
Where is the camera in the world?

- \( P = \begin{bmatrix} KR & Kt \end{bmatrix} \)
- \( K = \begin{bmatrix} 320 & 0 & 160 \\ 0 & 320 & 120 \\ 0 & 0 & 1 \end{bmatrix} \)
- \( R = K^{-1} P(1:3,1:3) \)
- \( t = K^{-1} P(1:3,4) \)
- \( T_{WC}^{(W)} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}^{-1} \)

Camera is at \([-4, -0.5, 2.5]\)

Rotation around axis \(x, y, z\) is \([-130^\circ, 0.0^\circ, -90^\circ]\)

To be more clear, remove the rotation of camera reference frame

- \( T_{WC}^{(W)} R_z(90^\circ), R_y(-90^\circ) \)
  rotation around axis \(x, y, z\) is \([0^\circ, 40.0^\circ, 0^\circ]\)
Vanishing point of \( x, y \) plane in the image

- \( \mathbf{v}_x = \mathbf{P} \begin{bmatrix} 1, 0, 0, 0 \end{bmatrix}^T \equiv \begin{bmatrix} 160.0, -148.5, 1 \end{bmatrix}^T \)
- \( \mathbf{v}_y = \mathbf{P} \begin{bmatrix} 0, 1, 0, 0 \end{bmatrix}^T \equiv \begin{bmatrix} 1, 0, 0 \end{bmatrix}^T \) (improper point)
- Remember: they are the 1\(^{st}\) and 2\(^{nd}\) column of \( \mathbf{P} \)

Where is the origin of the world in the image?

- \( \mathbf{o} = \mathbf{P} \begin{bmatrix} 0, 0, 0, 1 \end{bmatrix}^T \equiv \begin{bmatrix} 125.75, 75.05, 1 \end{bmatrix}^T \)
- Remember: it is the 4\(^{st}\) column of \( \mathbf{P} \)

Principal axis in world coordinates

- \( \mathbf{P} = \begin{bmatrix} \mathbf{M} & \mathbf{m} \end{bmatrix} \)
- \( \mathbf{O}^{(W)} = -\mathbf{M}^{-1} \mathbf{m} = \begin{bmatrix} -4, -0.5, 2.5 \end{bmatrix}^T \)
- \( \mathbf{d}^{(W)} = \mathbf{M}^{-1} \begin{bmatrix} c_x, c_y, 1 \end{bmatrix}^T = \begin{bmatrix} 0.766, 0, -0.6428 \end{bmatrix}^T \)
- \( \mathbf{a}^{(W)} = \mathbf{O}^{(W)} + \lambda \mathbf{d}^{(W)} \)
Questions

- Where is the intersection between principal axis and the floor?
- Calculate the field of view of the camera (image size is 320 × 240)
  
  i.e., the portion of the plane imaged by the camera
INTERSECTION BETWEEN PRINCIPAL AXIS AND THE FLOOR

- \( a^{(W)} = O^{(W)} + \lambda d^{(W)} \)
- \( a_{z=0}^{(W)} = [X, Y, 0]^T \)
- \( \lambda_{z=0} = -O_z^{(W)}/d_z^{(W)} = 3.89 \)
- \( a_{z=0}^{(W)} = [-1.02, -0.5, 0]^T \)

FIELD OF VIEW (1)

- \( a_1^{(W)} = O^{(W)} + \lambda_1 M^{-1} [0, 0, 1]^T \)
- \( a_2^{(W)} = O^{(W)} + \lambda_2 M^{-1} [320, 0, 1]^T \)
- \( a_3^{(W)} = O^{(W)} + \lambda_3 M^{-1} [0, 240, 1]^T \)
- \( a_4^{(W)} = O^{(W)} + \lambda_4 M^{-1} [320, 240, 1]^T \)
- calculate \( \lambda_i \) such that \( a_i^{(W)} \) has \( z = 0 \)
- ...
Field of view (2)

- Transformation between plane $z = 0$ and image plane is a 2D homography
- Consider $p_{z=0} = \begin{bmatrix} x, y, 0, 1 \end{bmatrix}^T$
- Projection $Pp_{z=0}$
- Notice that $p_{z=0} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} W \begin{bmatrix} x, y, 1 \end{bmatrix}^T$

$H = PW$ is the homography

maps points $(x, y)$ on the $z = 0$ plane to the image

$H' = H^{-1}$ is the inverse homography

maps image points $(u, v)$ to the $z = 0$ plane
**Field of view (2) (Continue)**

\[
\mathbf{H} = \begin{bmatrix}
122.56 & -320.00 & 587.38 \\
-113.76 & 0.00 & 350.60 \\
0.76 & 0 & 4.67
\end{bmatrix}
\]

- \( \mathbf{p}_{1z=0} = \mathbf{H}' \begin{bmatrix} 0, 0, 1 \end{bmatrix}^T \)
- \( \mathbf{p}_{2z=0} = \mathbf{H}' \begin{bmatrix} 320, 0, 1 \end{bmatrix}^T \)
- \( \mathbf{p}_{3z=0} = \mathbf{H}' \begin{bmatrix} 0, 240, 1 \end{bmatrix}^T \)
- \( \mathbf{p}_{4z=0} = \mathbf{H}' \begin{bmatrix} 320, 240, 1 \end{bmatrix}^T \)

The 3D world with the camera reference system (green), the world reference system (blue), the principal axis (dashed blue) and the Field of view (FoV) (grey)
**Questions**

- There is a “flat robot” moving on the floor, imaged by the camera

- Two distinct and coloured point are drawn on the robot.
  
  \[ \mathbf{p}_1^{(R)} = \begin{bmatrix} -0.3, 0 \end{bmatrix}^T, \mathbf{p}_2^{(R)} = \begin{bmatrix} 0.3, 0 \end{bmatrix}^T \]

- Could you calculate the robot position and orientation?
Robot pose

- Call $p'_1, p'_2$ the points in the image
- $p_{1z=0} = H'p'_1$
- $p_{2z=0} = H'p'_2$
- Position: $\frac{1}{2}(p_{1z=0} + p_{2z=0})$
- $d = p_{2z=0} - p_{1z=0}$
- Orientation: $\text{atan2}(d_y, d_x)$

A complete execution with a robot trajectory depicted in the world (top) and in the image (bottom)
A complete execution with a robot trajectory depicted in the world (top) and in the image (bottom). A Square is drawn on the floor to check correctness of the calculated vanishing point x (see previous questions)