Geometry in Computer Vision

Spring 2014

Lecture 7A

Representations of 3D rotations

P*n*P

- We can solve P*n*P by minimizing

$$\varepsilon_{PnP,GEO} = \sum_{k=1}^{n} d_{\text{PP}}(\mathbf{y}_k, \mathbf{y}'_k)^2, \quad \text{where} \quad \mathbf{y}'_k = \mathbf{R}\,\bar{\mathbf{x}}_k + \bar{\mathbf{t}},$$

over $\mathbf{R} \in$ SO(3) and $\mathbf{t} \in \mathbb{R}^3$

- Initial solution from P3P
- We need to parameterize $\mathbf{R} \in$ SO(3)

SfM

- A similar case appears in SfM, where we minimize

$$\varepsilon_{BA} = \sum_{k=1}^{m} \sum_{j=1}^{p} w_{kj}\, d_{\text{PP}}(\mathbf{y}_{kj}, \mathbf{C}_k \mathbf{x}_j)^2,$$

over the camera poses: $\mathbf{C}_k \sim (\mathbf{R}_k\ \mathbf{t}_k)$

- Each rotation $\mathbf{R}_k \in$ SO(3) needs to be parameterized

Parameterization of SO(3)

- Each R $\in$ SO(3) is a 3 $\times$ 3 matrix that satisfies

$\mathbf{R}^T\mathbf{R} = I$ and $\det(\mathbf{R}) = +1$

- How can we change **R** to **R'** such that these constraints are maintained?

Axis-angle representation

- Any rotation **R** is characterized by
  - a rotation axis **n** (normalized) (2 dof)
  - a rotation angle α (1 dof)

  such that **R** rotates the angle α about **n** according to the “right-hand rule”

(**n**, α)
same **R** as
(-**n**, -α)

Rodrigues’ rotation formula

- Given (**n**, α), how do we determine **R**?
- Use Rodrigues’ rotation formula:

$$\mathbf{R}(\hat{\mathbf{n}}, \alpha) = \hat{\mathbf{n}}\hat{\mathbf{n}}^\top + \cos\alpha\,(\mathbf{I} - \hat{\mathbf{n}}\hat{\mathbf{n}}^\top) + \sin\alpha\,[\hat{\mathbf{n}}]_\times .$$

$$\mathbf{R}(\hat{\mathbf{n}}, \alpha) = \mathbf{I} + (1 - \cos\alpha)\,(\hat{\mathbf{n}}\hat{\mathbf{n}}^\top - \mathbf{I}) + \sin\alpha\,[\hat{\mathbf{n}}]_\times .$$

$$\mathbf{R}(\hat{\mathbf{n}}, \alpha) = \mathbf{I} + (1 - \cos\alpha)\,[\hat{\mathbf{n}}]_\times^2 + \sin\alpha\,[\hat{\mathbf{n}}]_\times .$$

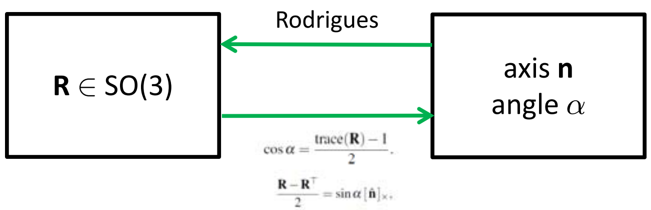
Rodrigues’ rotation formula

- Given **R**, how do we determine **n** and α?
- Based on Rodrigues’ formula:

$$\cos\alpha = \frac{\operatorname{trace}(\mathbf{R}) - 1}{2}. \qquad \frac{\mathbf{R} - \mathbf{R}^\top}{2} = \sin\alpha\,[\hat{\mathbf{n}}]_\times ,$$

- Notice: ambiguity at $\alpha = \pi$

Parameterization of SO(3)



The mapping (**n**, α) $\rightarrow$ SO(3) is easy to implement and can be differentiated w.r.t. (**n**, α)

Using so(3)

- so(3) is the set of 3 $\times$ 3 anti-symmetric matrices
- Can be parameterized by **m** $\in \mathbb{R}^3$: $[\mathbf{m}]_\times$
- 2 alternative mappings so(3) $\rightarrow$ SO(3)
  - Matrix exponential
  - Cayley transformation

so(3)

- $[\mathbf{m}]_\times \in$ so(3) has eigensystem:

  0, + i |**m**|, - i |**m**| eigenvalues

  m, **p** - i **q**, **p** + i **q** eigenvectors

  where (**m**, **p**, **q**) is a right-handed orthogonal basis in $\mathbb{R}^3$

Matrix exponential

- The matrix exponential function is defined for a square matrix **M** as

$$e^{\mathbf{M}} = \mathbf{I} + \mathbf{M} + \frac{1}{2}\mathbf{M}^2 + \frac{1}{6}\mathbf{M}^3 + \ldots = \sum_{k=0}^{\infty} \frac{1}{k!}\mathbf{M}^k,$$

Matrix exponential

- If **M** is diagonalized by unitary **E**:

  M = **E D E***, **D** diagonal, **E*****E** = **I**

  eigenvalues eigenvectors

  its exponential can be expressed as

  $e^{\mathbf{M}} = \mathbf{E}\, e^{\mathbf{D}}\, \mathbf{E}^*$

$$\exp\begin{pmatrix} d_1 & 0 & \ldots & 0 \\ 0 & d_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \ldots & d_n \end{pmatrix} = \begin{pmatrix} e^{d_1} & 0 & \ldots & 0 \\ 0 & e^{d_2} & \ldots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \ldots & e^{d_n} \end{pmatrix}$$

Exp of so(3)

- Based on the properties of the eigensystem of $[\mathbf{m}]_\times$ in combination with Rodrigues' formula:

$$e^{\alpha[\mathbf{n}]_x} = \mathbf{R}(\mathbf{n}, \alpha)$$

- exp : so(3) $\to$ SO(3)
- This mapping is *onto*
- It is not one-to-one

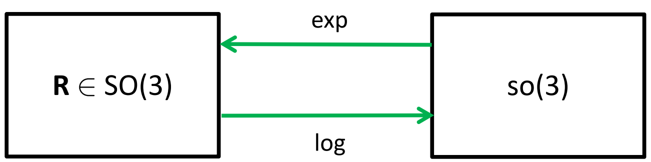
Exp of so(3)

- There is an inverse function: log **M**

$$\log[\mathbf{R}(\mathbf{n}, \alpha)] = \alpha[\mathbf{n}]_\times \qquad \text{(multiple-valued)}$$

- Matlab: expm and logm

Parameterization of SO(3)



The exp mapping can be implement and can be differentiated w.r.t. $\mathbf{m} = \alpha\, \mathbf{n}$
But less trivial than (n, α) $\to$ SO(3)

Cayley transformation

- If M $\in$ so(3), then

$$C(\mathbf{M}) = (\mathbf{I} - \mathbf{M})\,(\mathbf{I} + \mathbf{M})^{-1} \in SO(3)$$

- C : so(3) $\to$ SO(3)
- This mapping is almost onto
  - 180° rotations cannot be written as $C(\mathbf{M})$
- It is one-to-one on this domain

Cayley transformation

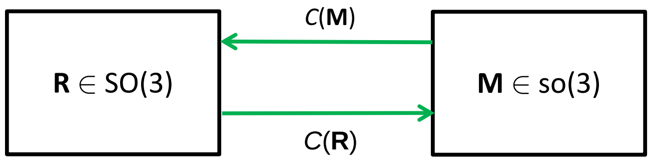
- If $\mathbf{M} = [a\ \mathbf{n}]_\times$ for normalized $\mathbf{n}$ and $a \in \mathbb{R}$:

$$C(\mathbf{M}) = \mathbf{R}(\mathbf{n}, \alpha), \quad a = \tan(\alpha/2)$$

- Inverse transformation:

$$\mathbf{M} = (\mathbf{I} - \mathbf{R})(\mathbf{I} + \mathbf{R})^{-1} = C(\mathbf{R})$$

Parameterization of SO(3)



The C mapping is easy to implement.
It can be differentiated w.r.t. $\mathbf{M} = [\mathbf{m}]_\times$
But less trivial than $(n, \alpha) \rightarrow SO(3)$

Quaternions

- Quaternions can be seen as a generalization of complex numbers to the case where we have three distinct imaginary units:

$$q = a + i\,b + j\,c + k\,d$$

$\mathbb{H}$

$\circ$	i	j	k
i	-1	k	$-j$
j	$-k$	-1	i
k	j	$-i$	-1

Quaternions

- Alternatively, we can see $\mathbb{H}$ as an algebra on $\mathbb{R}^4$, allowing us to multiply vectors in $\mathbb{R}^4$ to produce vectors in $\mathbb{R}^4$
- Alternatively, we can see $\mathbb{H}$ as an algebra on $\mathbb{R} \times \mathbb{R}^3$, consisting of ordered pairs of a real number and a vector in $\mathbb{R}^3$

- $q = (s, \mathbf{v}) \in \mathbb{H}$
- $|q|^2 = s^2 + |\mathbf{v}|^2$

Quaternions

- Given $q_1 = (s_1, \mathbf{v}_1)$ and $q_2 = (s_2, \mathbf{v}_2)$:

  $q_1 + q_2 = (s_1 + s_2, \mathbf{v}_1 + \mathbf{v}_2)$

  $q_1 \circ q_2 = (s_1 s_2 - \mathbf{v}_1 \cdot \mathbf{v}_2, s_1\mathbf{v}_2 + s_2\mathbf{v}_1 + \mathbf{v}_1 \times v_2)$

Quaternions

- Quaternion algebra satisfies
  - ❑ $\circ$ is distributive over +
  - ❑ $\circ$ is associative
  - ❑ (1, 0) is the unique identity element of $\circ$
  - ❑ Unique inverse of q = (s, v) is $(s, -v)/|q|^2$
- But:
  - ❑ $\circ$ is not commutative: $q_1 \circ q_2 \neq q_2 \circ q_1$ (in general)

Quaternions

- Unit quaternion q: $|q| = 1$
- A pure quaternion: $q = (0, \mathbf{v})$
  - – Pure quaternions can be seen as a representation of $\mathbb{R}^3$ in $\mathbb{H}$

- Any unit quaternion $q \in \mathbb{H}$ can be written as

  $$q = (\cos(\alpha/2), \sin(\alpha/2)\, \mathbf{n})$$

  for some α and $|\mathbf{n}|=1$.

- Any $\mathbf{u} \in \mathbb{R}^3$ can be represented as a pure quaternion $p = (0, \mathbf{u}) \in \mathbb{H}$

Quaternions and SO(3)

- Sandwich product:

$$q \circ p \circ q^{-1} = \ldots = (0, \mathbf{R}(\mathbf{n}, \alpha)\, \mathbf{u})$$

- Each rotation can be represented by a quaternion q = (cos(α/2), sin(α/2) **n**)
- Double embedding: both q and –q works

From $\mathbb{H}$ to SO(3)

- Given unit quaternion
  q = (q_1, q_2, q_3, q_4) = (cos(α/2), sin(α/2) **n**):

$$\mathbf{R} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} = \begin{pmatrix} q_1^2+q_2-q_3^2-q_4^2 & 2(q_2q_3-q_1q_4) & 2(q_1q_3+q_2q_4) \\ 2(q_2q_3+q_1q_4) & q_1^2-q_2+q_3^2-q_4^2 & 2(q_3q_4-q_1q_2) \\ 2(q_2q_4-q_1q_3) & 2(q_1q_2+q_3q_4) & q_1^2-q_2-q_3^2+q_4^2 \end{pmatrix}.$$

(A)

- Each element in **R** is a quadratic function in q

From SO(3) to $\mathbb{H}$

- From the previous mapping:

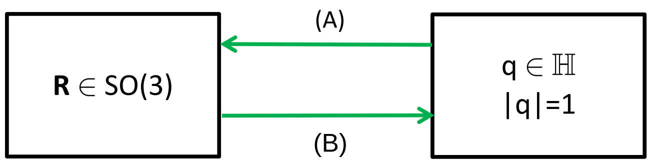
$$q_1^2 = \frac{1+r_{11}+r_{22}+r_{33}}{4}, \qquad q_2^2 = \frac{1+r_{11}-r_{22}-r_{33}}{4},$$
$$q_3^2 = \frac{1-r_{11}+r_{22}-r_{33}}{4}, \qquad q_4^2 = \frac{1-r_{11}-r_{22}+r_{33}}{4}.$$

and

$$r_{12}+r_{21} = 4\,q_2q_3, \qquad r_{13}+r_{31} = 4\,q_2q_4, \qquad r_{23}+r_{32} = 4\,q_3q_4,$$
$$r_{21}-r_{12} = 4\,q_1q_4, \qquad r_{13}-r_{31} = 4\,q_1q_3, \qquad r_{32}-r_{23} = 4\,q_1q_2.$$

(B)

Parameterization of SO(3)



Both (A) and (B) are easy to implement.
(A) can be differentiated w.r.t. unit quaternion q $\in \mathbb{R}^4$

Euler angles

- We can decompose any **R** $\in$ SO(3) into a product of 3 rotations around *fixed axes*
- For example:

R = $\mathrm{Rot}_z(\alpha_1)\,\mathrm{Rot}_x(\alpha_2)\,\mathrm{Rot}_z(\alpha_3)$

- $(\alpha_1, \alpha_2, \alpha_3)$ are the *Euler angles* of **R**

Euler angles

- There are straight-forward mappings

$$(\alpha_1, \alpha_2, \alpha_3) \leftrightarrow \mathbf{R} \in \mathrm{SO}(3)$$

- Notice: rotations about the z-axis always have an ambiguous representation:

$$\mathbf{R}(\alpha_1, 0, \alpha_3) = \mathbf{R}(\alpha_1+\Delta, 0, \alpha_3-\Delta)$$

Euler angles

- This ambiguity implies that **D**, the derivatives of **R** with respect to $(\alpha_1, \alpha_2, \alpha_3)$ is rank deficient when $\alpha_2 = 0$
- If Euler angles are use as a parameterization of **R** in a non-linear optimization, there will be a stationary point for all points $(\alpha_1, 0, \alpha_3)$ where the optimization can get stuck