CH 3

3차원 강체 변화

1) Transformation Matrix

2) Eigen 라이브러리

3) 좌표계변환 표현

4) Eigen Geometry

3차원 강체 변환 (좌표계 변환)

- 왜? : 카메라위치 P 에 대해
  CAMERA 좌표계 : p_c
  세계(관성) 좌표계 : p_w
  p_c 를 p_w에 대응시키기 위해 (변환시키기 위해) 필요

즉, Pose란 좌표계 관계를 나타내는 T행렬로 표현 가능하다

Transformation Matrix

- R = 회전 행렬 , t = 병진 행렬
- $b = R_m a + t_m$

  $c = R_n b + t_n = R_n(R_m a + t_m) + t_n$ …복잡함

  $c' = T_n T_m a' = T_{nm} a'$ 으로 재정의

$$a' = \begin{matrix} a_{11} & a_{12} & a_{13} & 0 \\ a_{21} & a_{22} & a_{23} & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ 0 & 0 & 0 & 1 \end{matrix} \qquad \mathrm{T} = \begin{matrix} R_{11} & R_{12} & R_{13} & t_1 \\ R_{21} & R_{22} & R_{23} & t_2 \\ R_{31} & R_{32} & R_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{matrix} \qquad \mathrm{T}^{-1} = \begin{matrix} R_{11}^T & R_{12}^T & R_{13}^T & (-R^T t)_1 \\ R_{21}^T & R_{22}^T & R_{23}^T & (-R^T t)_2 \\ R_{31}^T & R_{32}^T & R_{33}^T & (-R^T t)_3 \\ 0 & 0 & 0 & 1 \end{matrix}$$

Eigen

- C++ Eigen라이브러리 활용
- Eigenmatrix.cpp

  예제 1. 기본 행렬 선언, 출력

  예제 2. 행렬 변수 타입 활용

  예제 3. 행렬 계산

  예제 4. Eigen 계산

  예제 5. 1차 연립방정식 $(AX = B)$ >> 역행렬 vs Qr분해법

좌표계변환 표현 (회전행렬, 오일러 각)

- 회전 행렬 SO3 (Special Orthogonal Matrix 3x3)
  $Det(R) = 1$, $RR^T = I$, 9개의 수 필요 … 솔루션이 제약적

- 오일러 각
  장점 : 인간에게 가장 직관적, 알고리즘 검사에 이용
  단점 : 보간/반복에 적합하지 않음, 짐벌락 문제

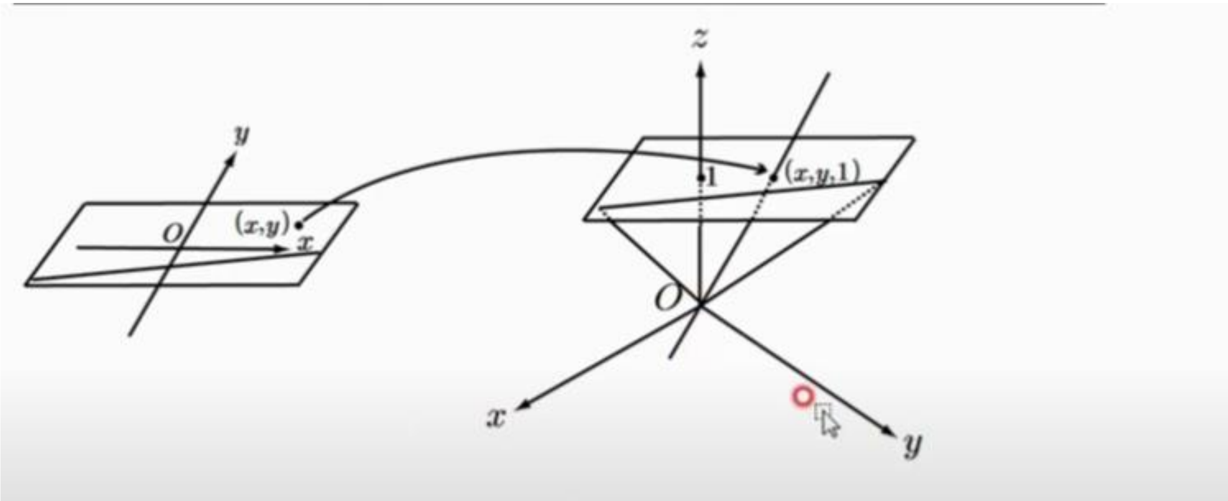
좌표계변환 표현 (쿼터니언)

- $q = \left[\cos\frac{\theta}{2}, \sin\frac{\theta}{2} v\right]$ … $|v| = 1$

  더하기, 빼기 가능
  곱셈 (복소수 곱셈, $ij = k$)
  내적 ($\mathrm{q_a}.\mathrm{q_b} = [s_a s_b,\ x_a x_b,\ y_a y_b,\ z_a z_b]$)
  켤레 ($q^* = [s, -x, -y, -z]$)
  모듈 ($\|q\| = \sqrt{s^2 + x^2 + y^2 + z^2}$)
  역 ($q^{-1} = \frac{q^*}{\|q\|}$, $qq^{-1} = 1$)
  회전 ($p' = qpq^{-1}$)

좌표계변환 표현 (유사, 아핀, 투영)

아핀평면 투영평면



$R^2_{z=1} = \{(x,y,1) \mid x,y \in R\}$		R^3
$(x,y,1)$	점은 원점을 지나는 직선으로 대응	$t(x,y,1)$
$ax+by+c=0$	직선은 원점과 직선을 지나는 평면으로 대응	$ax+by+cz=0$

좌표계변환 표현 (유사, 아핀, 투영)

- 점점 일반적인 경우로 확대
- 유사 (Similarity Transform)

  $\mathrm{T} = \begin{pmatrix} sR & t \\ 0^T & 1 \end{pmatrix}$ ... $\mathrm{Det(sR)} = s^2,\ Orthogonal$

- 아핀 (Affine Transform)

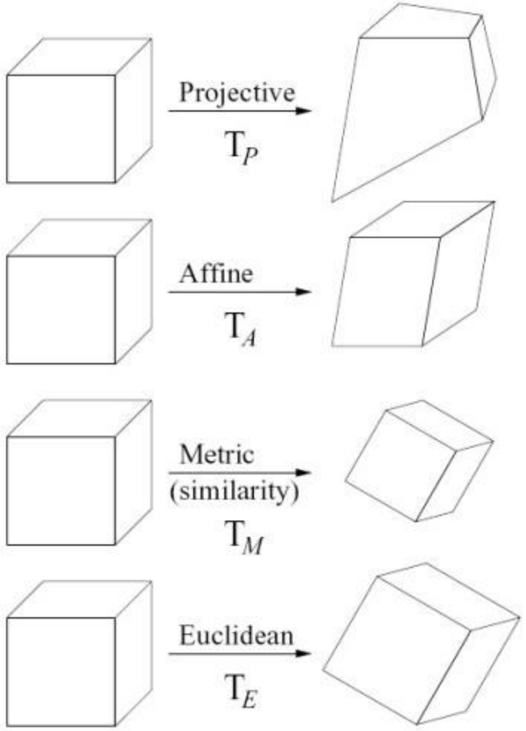
  $\mathrm{T} = \begin{pmatrix} A & t \\ 0^T & 1 \end{pmatrix}$ ... $\mathrm{Det}(A) \neq 0$

- 투영 (Projective Transform)

  (homography)

  3x3 지만 실제 $\mathrm{P}^2 \rightarrow \mathrm{P}^2 = \begin{pmatrix} A & t \\ s^T & v \end{pmatrix}$ ... $\mathrm{Det}(T) \neq 0$

  사영카메라의 경우, 3x4(사영행렬)으로 표현



관련 유튜브 강좌

https://www.youtube.com/playlist?list=PLyj-PTT6qfWbflLyKHp47vztZejEIfj3p

Eigen Geometry

- C++ Eigen라이브러리 활용
- EigenGeometry.cpp

  예제 1. 회전 벡터 Eigen::AngleAxisd

  예제 2. 회전 행렬 rotation_vector. toRotationMatrix( )

  예제 3. 오일러 각 rotation_matrix. eulerAngles( )

  예제 4. SO(3) T.rotate(rotation_vector)

  T.pretranslate(t)

  예제 5. 쿼터니언 Eigen::Quaterniond