

SmartWrist

2015 / 10 / 13
Yoo, Kyunghwan

Pivot

- EV : A의 초기 위치
- EA : 현재 측정된 값
- VA : Pivot 후 결과

■ Rotation Matrix

■ Transpose

$${}^E_R^T = {}^V_R$$

■ Inner Product (Dot)

$${}^V_R \cdot {}^E_A = {}^V_A$$

■ Quaternion

■ Conjugate

$${}^E_q^* = {}^V_q$$

■ Product

$${}^E_q \otimes {}^V_q = {}^V_q$$

각도 계산

■ Input Value

■ DEG2RAD 처리 후 적용 $\left(\frac{\pi}{180}\right)$

- VA : 손목
- VB : 손등

Pseudocode Algorithm

```
erVA = cRPY2RM[VA];  
rqVA = cRM2Q[erVA];  
  
erVB = cRPY2RM[VB];  
rqVB = cRM2Q[erVB];  
  
rqAV = cQ_Conjugate[rqVA];  
rqAVB = cQ_Product[rqAV, rqVB];  
  
qeAVB = cQ2RPY[rqAVB];
```

■ Return Value

■ RAD2DEG 처리 후 적용 $\left(\frac{180}{\pi}\right)$

- qeAVB : 각도 결과

■ Rotation Matrix

■ Transpose

$${}^V_R{}^T = {}^A_R$$

■ Inner Product (Dot)

$${}^A_R \cdot {}^V_B R = {}^A_B R$$

■ Quaternion

■ Conjugate

$${}^V_q{}^* = {}^A_q$$

■ Product

$${}^V_B q \otimes {}^A_q = {}^A_B q$$

Pseudocode Function

```

function cRPY2RM[N]
{
    // Roll Pitch Yaw ⇒ Rotation Matrix
    O.R11 = Cos[N.Y] * Cos[N.P];
    O.R12 = Cos[N.Y] * Sin[N.P] * Sin[N.R] - Sin[N.Y] * Cos[N.R];
    O.R13 = Cos[N.Y] * Sin[N.P] * Cos[N.R] + Sin[N.Y] * Sin[N.R];
    O.R21 = Sin[N.Y] * Cos[N.P];
    O.R22 = Sin[N.Y] * Sin[N.P] * Sin[N.R] + Cos[N.Y] * Cos[N.R];
    O.R23 = Sin[N.Y] * Sin[N.P] * Cos[N.R] - Cos[N.Y] * Sin[N.R];
    O.R31 = -Sin[N.P];
    O.R32 = Cos[N.P] * Sin[N.R];
    O.R33 = Cos[N.P] * Cos[N.R];
    return O;
}

function cRM2Q[N]
{
    // Rotation Matrix ⇒ Quaternion
    O.q1 = 1 / 2 * Sqrt[N.R11 + N.R22 + N.R33 + 1];
    O.q2 = -Sign[N.R32 - N.R23] / 2 * Sqrt[N.R11 - N.R22 - N.R33 + 1];
    O.q3 = -Sign[N.R13 - N.R31] / 2 * Sqrt[N.R22 - N.R11 - N.R33 + 1];
    O.q4 = -Sign[N.R21 - N.R12] / 2 * Sqrt[N.R33 - N.R11 - N.R22 + 1];
    return O;
}

function cQ2RPY[N]
{
    // Quaternion ⇒ Roll Pitch Yaw
    O.R = ArcTan[2 * N.q1^2 - 1 + 2 * N.q4^2, 2 * (N.q3 * N.q4 - N.q1 * N.q2)];
    O.P = -ArcSin[2 * (N.q2 * N.q4 + N.q1 * N.q3)];
    O.Y = ArcTan[2 * N.q1^2 - 1 + 2 * N.q2^2, 2 * (N.q2 * N.q3 - N.q1 * N.q4)];
    return O;
}

function cQ_Conjugate[AB]
{
    // ABq* = > BAq
    BA.q1 = AB.q1;
    BA.q2 = -AB.q2;
    BA.q3 = -AB.q3;
    BA.q4 = -AB.q4;
    return BA;
}

function cQ_Product[AB, BC]
{
    // BCq⊗ABq = > ACq
    AC.q1 = BC.q1 * AB.q1 - BC.q2 * AB.q2 - BC.q3 * AB.q3 - BC.q4 * AB.q4;
    AC.q2 = BC.q1 * AB.q2 + BC.q2 * AB.q1 + BC.q3 * AB.q4 - BC.q4 * AB.q3;
    AC.q3 = BC.q1 * AB.q3 - BC.q2 * AB.q4 + BC.q3 * AB.q1 + BC.q4 * AB.q2;
    AC.q4 = BC.q1 * AB.q4 + BC.q2 * AB.q3 - BC.q3 * AB.q2 + BC.q4 * AB.q1;
    return AC;
}

```