3D Rotation

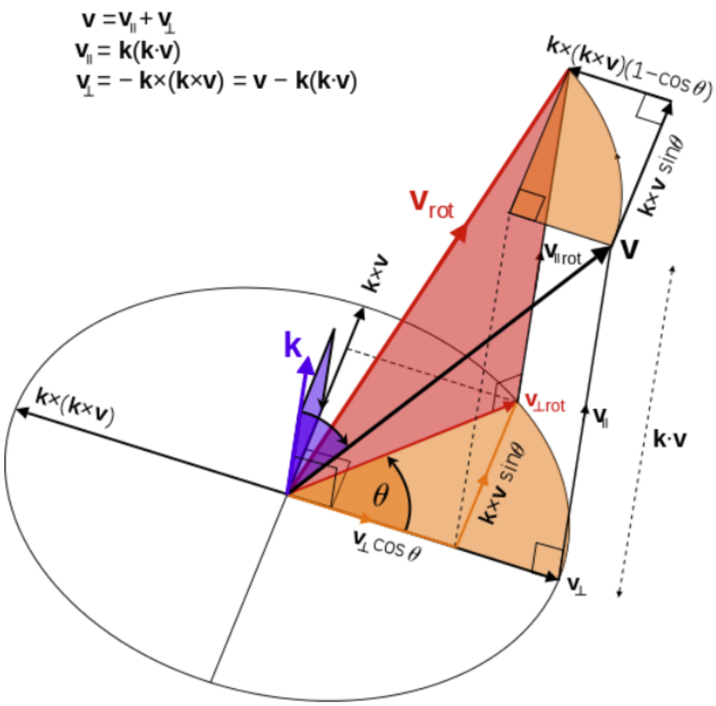
Rodirigous Formula

Basic Idea

A 3d-vector $\vec{v}$ rotate around the normalized axis $\vec{k}$, and result in $\vec{v}_{rot}$

$$\vec{v}_{rot} = \vec{v}\cos\theta + \vec{k}(\vec{k}\cdot\vec{v})(1-\cos\theta) + \vec{k}\times\vec{v}\sin\theta$$

Note that in a particular coordinate system, the formula can only represent the rotation of a point, with coordinates $\vec{v}$, around axis $\vec{k}$, when the axis $\vec{k}$ pass through the origin.



Rotation Matrix

We can write the Rodirigous formula to matrix form $\vec{v}_{rot} = R\vec{v}$

$$R = \begin{bmatrix} \cos\theta + k_x^2(1-\cos\theta) & k_xk_y(1-\cos\theta) - k_z\sin\theta & k_xk_z(1-\cos\theta) + k_y\sin\theta \\ k_xk_y(1-\cos\theta) + k_z\sin\theta & \cos\theta + k_y^2(1-\cos\theta) & k_yk_z(1-\cos\theta) - k_x\sin\theta \\ k_xk_z(1-\cos\theta) - k_y\sin\theta & k_yk_z(1-\cos\theta) + k_x\sin\theta & \cos\theta + k_z^2(1-\cos\theta) \end{bmatrix}$$

It is worth noting that R is an orthogonal matrix with three degree of freedom, i.e $R^{-1} = R^T$

Quaternion

Basics

- Representation
  A quaternion is a 4d-vector

$$q = w + x\vec{i} + y\vec{j} + z\vec{k}$$

We can rewrite it to the vector form as

$$q = (w, x, y, z) = (w, \vec{v})$$

- Operation
  - norm

$$||q|| = \sqrt{w^2 + x^2 + y^2 + z^2}$$

  - conjugate

$$q^* = (w, -x, -y, -z)$$

  - inverse

$$q^{-1} = \frac{q^*}{||q||^2}$$

  - add

$$q1 + q2 = (w_1 + w_2, x_1 + x_2, y_1 + y_2, z_1 + z_2)$$

  - scaler multiply

$$c \cdot q = (cw, cx, cy, cz)$$

  - multiply

$$q_1 \cdot q_2 = \begin{bmatrix} w_1 & -x_1 & -y_1 & -z_1 \\ x_1 & w_1 & -z_1 & y_1 \\ y_1 & z_1 & w_1 & -x_1 \\ z_1 & -y_1 & x_1 & w_1 \end{bmatrix} \begin{bmatrix} w_2 \\ x_2 \\ y_2 \\ z_2 \end{bmatrix}$$

Representing Rotation with Quaternion

Suppose we have a normalized rotation axis $\vec{k}$ and a rotation angle θ, than we can represent the unit rotate quaternion as

$$q = (\cos\frac{\theta}{2}, \vec{k}\sin\frac{\theta}{2})$$

To rotate a 3d-vector $\vec{v}$ with the above axis and angle, we can first represent $\vec{v}$ in quaternion form $[0, \vec{v}]$, and the result $\vec{v}_{rot}$ can be write in quaternion form

$$[0, \vec{v}_{rot}] = q \cdot [0, \vec{v}] \cdot q^*$$

Interpolation

To interpolate from two quaternion q_0 and q_1, there are many algorithms including Nlerp, Slerp. We'll introduce Slerp here, which ensure that the linear interpolation of the rotation angle.

$$q(t) = \frac{sin((1-t)\theta)}{\sin\theta} q_0 + \frac{\sin(t\theta)}{\sin\theta} q_1, \theta = arccos(w_0 w_1 + \vec{v}_0 \cdot \vec{v}_1)$$

Euler Angles

Basics

Rotation matrix of x-axis, y-axis and z-axis are as follows

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}, R_y(\beta) = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix}, R_z(\gamma) = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

We can express the 3d rotation with three euler angles, with specified the rotate order and rotate method.

- Rotate Order
  A rotate order a-b-c is understood as first rotate around a-axis, then b-axis and finally c-axis.
  - Tait-Bryan angles
    - x-y-z, y-z-x, z-x-y, x-z-y, z-y-x, y-x-z
  - proper Euler angles
    - z-x-z, x-y-x, y-z-y, z-y-z, x-z-x, y-x-y
- Rotate Method
  Different rotate method leading to different result. What is superising is that intrinsic rotation a-b-c gives the same result as extrinsic rotation c-b-a and vice versa.
  - intrinsic rotation
    - rotate around the rotated axis
  - extrinsic rotation
    - rotate around the fixed axis

Tait-Bryan angle examples

Tait-Bryan angle are widely used in engineering, so we will mainly focus on the Tait-Bryan angles/

- x-y-z intrinsic/z-y-x extrinsic rotate

$$R = R_x(\alpha)R_y(\beta)R_z(\gamma) = \begin{bmatrix} \cos\beta\cos\gamma & -\sin\gamma\cos\beta & \sin\beta \\ \sin\alpha\sin\beta\cos\gamma + \sin\gamma\cos\alpha & -\sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma & -\sin\alpha\cos\beta \\ \sin\alpha\sin\gamma - \sin\beta\cos\alpha\cos\gamma & \sin\alpha\cos\gamma + \sin\beta\sin\gamma\cos\alpha & \cos\alpha\cos\beta \end{bmatrix}$$

- x-z-y intrinsic/y-z-x extrinsic rotate

$$R = R_x(\alpha)R_z(\gamma)R_y(\beta) = \begin{bmatrix} \cos\beta\cos\gamma & -\sin\gamma & \sin\beta\cos\gamma \\ \sin\alpha\sin\beta + \sin\gamma\cos\alpha\cos\beta & \cos\alpha\cos\gamma & -\sin\alpha\cos\beta + \sin\beta\sin\gamma\cos\alpha \\ \sin\alpha\sin\gamma\cos\beta - \sin\beta\cos\alpha & \sin\alpha\cos\gamma & \sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\beta \end{bmatrix}$$

- y-x-z intrinsic/z-x-y extrinsic rotate

$$R = R_y(\beta)R_x(\alpha)R_z(\gamma) = \begin{bmatrix} \sin\alpha\sin\beta\sin\gamma + \cos\beta\cos\gamma & \sin\alpha\sin\beta\cos\gamma - \sin\gamma\cos\beta & \sin\beta\cos\alpha \\ \sin\gamma\cos\alpha & \cos\alpha\cos\gamma & -\sin\alpha \\ \sin\alpha\sin\gamma\cos\beta - \sin\beta\cos\gamma & \sin\alpha\cos\beta\cos\gamma + \sin\beta\sin\gamma & \cos\alpha\cos\beta \end{bmatrix}$$

- y-z-x intrinsic/y-z-x extrinsic rotate

$$R = R_y(\beta)R_z(\gamma)R_x(\alpha) = \begin{bmatrix} \cos\beta\cos\gamma & \sin\alpha\sin\beta - \sin\gamma\cos\alpha\cos\beta & \sin\alpha\sin\gamma\cos\beta + \sin\beta\cos\alpha \\ \sin\gamma & \cos\alpha\cos\gamma & -\sin\alpha\cos\gamma \\ -\sin\beta\cos\gamma & \sin\alpha\cos\beta + \sin\beta\sin\gamma\cos\alpha & -\sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\beta \end{bmatrix}$$

- z-x-y intrinsic/z-y-x extrinsic rotate

$$R = R_z(\gamma)R_x(\alpha)R_y(\beta) = \begin{bmatrix} -\sin\alpha\sin\beta\sin\gamma + \cos\beta\cos\gamma & -\sin\gamma\cos\alpha & \sin\alpha\sin\gamma\cos\beta + \sin\beta\cos\gamma \\ \sin\alpha\sin\beta\cos\gamma + \sin\gamma\cos\beta & \cos\alpha\cos\gamma & -\sin\alpha\cos\beta\cos\gamma + \sin\beta\sin\gamma \\ -\sin\beta\cos\alpha & \sin\alpha & \cos\alpha\cos\beta \end{bmatrix}$$

- z-y-x intrinsic/x-y-z extrinsic rotate

$$R = R_z(\gamma)R_y(\beta)R_x(\alpha) = \begin{bmatrix} \cos\beta\cos\gamma & \sin\alpha\sin\beta\cos\gamma - \sin\gamma\cos\alpha & \sin\alpha\sin\gamma + \sin\beta\cos\alpha\cos\gamma \\ \sin\gamma\cos\beta & \sin\alpha\sin\beta\sin\gamma + \cos\alpha\cos\gamma & -\sin\alpha\cos\gamma + \sin\beta\sin\gamma\cos\alpha \\ -\sin\beta & \sin\alpha\cos\beta & \cos\alpha\cos\beta \end{bmatrix}$$

Transformation

Rotation Matrix <=> Quaternion

- Rotation Matrix => Quaternion

$$w = \pm\frac{1}{2}\sqrt{1 + R_{11} + R_{22} + R_{33}}, x = \frac{1}{4w}(R_{32} - R_{23}), y = \frac{1}{4w}(R_{13} - R_{31}), z = \frac{1}{4w}(R_{21} - R_{12})$$

Note that the one rotation matrix is corresponding to two quaternion exactly.
Basically, the above formula is good, while w is zero leading to singularity. An alternative way to avoid this problem is to choose the biggest variable as the divider.

$$x = \pm\frac{1}{2}\sqrt{1 + R_{11} - R_{22} - R_{33}}, w = \frac{1}{4x}(R_{32} - R_{23}), y = \frac{1}{4x}(R_{12} + R_{21}), z = \frac{1}{4x}(R_{13} + R_{31})$$

$$y = \pm\frac{1}{2}\sqrt{1 - R_{11} + R_{22} - R_{33}}, w = \frac{1}{4y}(R_{13} - R_{31}), x = \frac{1}{4y}(R_{12} + R_{21}), z = \frac{1}{4y}(R_{23} + R_{32})$$

$$z = \pm\frac{1}{2}\sqrt{1 - R_{11} - R_{22} + R_{33}}, w = \frac{1}{4z}(R_{21} - R_{12}), x = \frac{1}{4z}(R_{13} + R_{31}), y = \frac{1}{4z}(R_{23} + R_{32})$$

- Quaternion => Rotation Matrix

$$R = \begin{bmatrix} 1 - 2y^2 - 2z^2 & 2xy - 2wz & 2xz + 2wy \\ 2xy + 2wz & 1 - 2x^2 - 2z^2 & 2yz - 2wx \\ 2xz - 2wy & 2yz + 2wx & 1 - 2x^2 - 2y^2 \end{bmatrix}$$

Note that the one quaternion correspond to one rotation matrix.
I'll provide the code from the glm::mat_cast. glm use column-first matrix storage.

Euler Angles to Rotation Matrix

It is easy to get rotation matrix from euler angles with specified rotate order i-j-k and rotate method

- extrinsic rotate

$$R = R_k(\theta_k)R_j(\theta_j)R_i(\theta_i)$$

- intrinsic rotate

$$R = R_i(\theta_i)R_j(\theta_j)R_k(\theta_k)$$

Rotation Matrix to Euler Angles

If gimbal lock occurs, then one rotation matrix will corresponding to infinity euler angles, otherwise, one rotation matrix is correspond to two euler angles.

Here I'll give the rotation matrix to Tait-Bryan angles

- x-y-z intrinsic / z-y-x extrinsic rotate

$$\beta = arcsin R_{13}, \alpha = atan2(-\frac{R_{23}}{\cos\beta}, \frac{R_{33}}{\cos\beta}), \gamma = atan2(-\frac{R_{12}}{\cos\beta}, \frac{R_{11}}{\cos\beta})$$

$$\beta = \pi - arcsin R_{13}, \alpha = atan2(-\frac{R_{23}}{\cos\beta}, \frac{R_{33}}{\cos\beta}), \gamma = atan2(-\frac{R_{12}}{\cos\beta}, \frac{R_{11}}{\cos\beta})$$

  if $\beta = \pm\frac{\pi}{2}$, gimbal lock occurs and the above solutions are invalid

$$\beta = \frac{\pi}{2}, \alpha + \gamma = atan2(R_{21}, R_{22})$$

$$\beta = -\frac{\pi}{2}, \alpha - \gamma = atan2(R_{32}, R_{22})$$

- x-z-y intrinsic / y-z-x extrinsic rotate

$$\gamma = -arcsin R_{12}, \alpha = atan2(\frac{R_{32}}{\cos\gamma}, \frac{R_{22}}{\cos\gamma}), \beta = atan2(\frac{R_{13}}{\cos\gamma}, \frac{R_{11}}{\cos\gamma})$$

$$\gamma = \pi + arcsin R_{12}, \alpha = atan2(\frac{R_{32}}{\cos\gamma}, \frac{R_{22}}{\cos\gamma}), \beta = atan2(\frac{R_{13}}{\cos\gamma}, \frac{R_{11}}{\cos\gamma})$$

  if $\gamma = \pm\frac{\pi}{2}$, gimbal lock occurs and the above solutions are invalid

$$\gamma = \frac{\pi}{2}, \alpha - \beta = atan2(R_{31}, R_{21})$$

$$\gamma = -\frac{\pi}{2}, \alpha + \beta = atan2(-R_{31}, R_{33})$$

- y-x-z intrinsic / z-x-y extrinsic rotate

$$\alpha = -arcsin R_{23}, \beta = atan2(\frac{R_{13}}{\cos\alpha}, \frac{R_{33}}{\cos\alpha}), \gamma = atan2(\frac{R_{21}}{\cos\alpha}, \frac{R_{22}}{\cos\alpha})$$

$$\alpha = \pi + arcsin R_{23}, \beta = atan2(\frac{R_{13}}{\cos\alpha}, \frac{R_{33}}{\cos\alpha}), \gamma = atan2(\frac{R_{21}}{\cos\alpha}, \frac{R_{22}}{\cos\alpha})$$

  if $\alpha = \pm\frac{\pi}{2}$, gimbal lock occurs and the above solutions are invalid

$$\alpha = \frac{\pi}{2}, \beta - \gamma = atan2(R_{12}, R_{11})$$

$$\alpha = -\frac{\pi}{2}, \beta + \gamma = atan2(-R_{12}, R_{11})$$

- y-z-x intrinsic / x-z-y extrinsic rotate

$$\gamma = arcsinR_{21}, \alpha = atan2(-\frac{R_{23}}{\cos\gamma}, \frac{R_{22}}{\cos\gamma}), \beta = atan2(-\frac{R_{31}}{\cos\gamma}, \frac{R_{11}}{\cos\gamma})$$

$$\gamma = \pi - arcsinR_{21}, \alpha = atan2(-\frac{R_{23}}{\cos\gamma}, \frac{R_{22}}{\cos\gamma}), \beta = atan2(-\frac{R_{31}}{\cos\gamma}, \frac{R_{11}}{\cos\gamma})$$

if $\gamma = \pm\frac{\pi}{2}$, gimbal lock occurs and the above solutions are invalid

$$\gamma = \frac{\pi}{2}, \alpha + \beta = atan2(R_{13}, R_{33})$$

$$\gamma = -\frac{\pi}{2}, \alpha - \beta = atan2(R_{32}, R_{12})$$

- z-x-y intrinsic / y-x-z extrinsic rotate

$$\alpha = arcsinR_{32}, \beta = atan2(-\frac{R_{31}}{\cos\alpha}, \frac{R_{33}}{\cos\alpha}), \gamma = atan2(-\frac{R_{12}}{\cos\alpha}, \frac{R_{22}}{\cos\alpha})$$

$$\alpha = \pi - arcsinR_{32}, \beta = atan2(-\frac{R_{31}}{\cos\alpha}, \frac{R_{33}}{\cos\alpha}), \gamma = atan2(-\frac{R_{12}}{\cos\alpha}, \frac{R_{22}}{\cos\alpha})$$

if $\alpha = \pm\frac{\pi}{2}$, gimbal lock occurs and the above solutions are invalid

$$\alpha = \frac{\pi}{2}, \beta + \gamma = atan2(R_{13}, R_{11})$$

$$\alpha = -\frac{\pi}{2}, \beta - \gamma = atan2(R_{13}, R_{11})$$

- z-y-x intrinsic / x-y-z extrinsic rotate

$$\beta = -arcsinR_{31}, \alpha = atan2(\frac{R_{32}}{\cos\beta}, \frac{R_{33}}{\cos\beta}), \gamma = atan2(\frac{R_{21}}{\cos\beta}, \frac{R_{11}}{\cos\beta})$$

$$\beta = \pi + arcsinR_{31}, \alpha = atan2(\frac{R_{32}}{\cos\beta}, \frac{R_{33}}{\cos\beta}), \gamma = atan2(\frac{R_{21}}{\cos\beta}, \frac{R_{11}}{\cos\beta})$$

if $\beta = \pm\frac{\pi}{2}$, gimbal lock occurs and the above solutions are invalid

$$\beta = \frac{\pi}{2}, \alpha - \gamma = atan2(R_{12}, R_{22})$$

$$\beta = -\frac{\pi}{2}, \alpha + \gamma = atan2(-R_{12}, R_{22})$$

Euler angles <=> Quaternion

Euler angles => Quaternion

It quite easy to get the euler angles to quaternion according to the definition.
The elementary rotation matrix can be interpreted as

$$Rx(\alpha) => q_x = (\cos\frac{\alpha}{2}, \sin\frac{\alpha}{2}, 0, 0)$$

$$Ry(\beta) => q_y = (\cos\frac{\beta}{2}, 0, \sin\frac{\beta}{2}, 0)$$

$$Rz(\gamma) => q_z = (\cos\frac{\gamma}{2}, 0, 0, \sin\frac{\gamma}{2})$$

so, we can get quaternion by multiply the above elementary quaternion.

- x-y-z intrinsic / z-y-x extrinsic rotate

$$q = q_x q_y q_z$$

$$w = -\sin\frac{\alpha}{2}\sin\frac{\beta}{2}\sin\frac{\gamma}{2} + \cos\frac{\alpha}{2}\cos\frac{\beta}{2}\cos\frac{\gamma}{2}$$

$$x = \sin\frac{\alpha}{2}\cos\frac{\beta}{2}\cos\frac{\gamma}{2} + \sin\frac{\beta}{2}\sin\frac{\gamma}{2}\cos\frac{\alpha}{2}$$

$$y = -\sin\frac{\alpha}{2}\sin\frac{\gamma}{2}\cos\frac{\beta}{2} + \sin\frac{\beta}{2}\cos\frac{\alpha}{2}\cos\frac{\gamma}{2}$$

$$z = \sin\frac{\alpha}{2}\sin\frac{\beta}{2}\cos\frac{\gamma}{2} + \sin\frac{\gamma}{2}\cos\frac{\alpha}{2}\cos\frac{\beta}{2}$$

- x-z-y intrinsic / y-z-x extrinsic rotate

$$q = q_x q_z q_y$$

$$w = \sin\frac{\alpha}{2}\sin\frac{\beta}{2}\sin\frac{\gamma}{2} + \cos\frac{\alpha}{2}\cos\frac{\beta}{2}\cos\frac{\gamma}{2}$$

$$x = \sin\frac{\alpha}{2}\cos\frac{\beta}{2}\cos\frac{\gamma}{2} - \sin\frac{\beta}{2}\sin\frac{\gamma}{2}\cos\frac{\alpha}{2}$$

$$y = -\sin\frac{\alpha}{2}\sin\frac{\gamma}{2}\cos\frac{\beta}{2} + \sin\frac{\beta}{2}\cos\frac{\alpha}{2}\cos\frac{\gamma}{2}$$

$$z = \sin\frac{\alpha}{2}\sin\frac{\beta}{2}\cos\frac{\gamma}{2} + \sin\frac{\gamma}{2}\cos\frac{\alpha}{2}\cos\frac{\beta}{2}$$

- y-x-z intrinsic / z-x-y extrinsic rotate

$$q = q_y q_x q_z$$

$$w = \sin\frac{\alpha}{2}\sin\frac{\beta}{2}\sin\frac{\gamma}{2} + \cos\frac{\alpha}{2}\cos\frac{\beta}{2}\cos\frac{\gamma}{2}$$

$$x = \sin\frac{\alpha}{2}\cos\frac{\beta}{2}\cos\frac{\gamma}{2} + \sin\frac{\beta}{2}\sin\frac{\gamma}{2}\cos\frac{\alpha}{2}$$

$$y = -\sin\frac{\alpha}{2}\sin\frac{\gamma}{2}\cos\frac{\beta}{2} + \sin\frac{\beta}{2}\cos\frac{\alpha}{2}\cos\frac{\gamma}{2}$$

$$z = -\sin\frac{\alpha}{2}\sin\frac{\beta}{2}\cos\frac{\gamma}{2} + \sin\frac{\gamma}{2}\cos\frac{\alpha}{2}\cos\frac{\beta}{2}$$

- y-z-x intrinsic / x-z-y extrinsic rotate

$$q = q_y q_z q_x$$

$$w = -\sin\frac{\alpha}{2}\sin\frac{\beta}{2}\sin\frac{\gamma}{2} + \cos\frac{\alpha}{2}\cos\frac{\beta}{2}\cos\frac{\gamma}{2}$$

$$x = \sin\frac{\alpha}{2}\cos\frac{\beta}{2}\cos\frac{\gamma}{2} + \sin\frac{\beta}{2}\sin\frac{\gamma}{2}\cos\frac{\alpha}{2}$$

$$y = \sin\frac{\alpha}{2}\sin\frac{\gamma}{2}\cos\frac{\beta}{2} + \sin\frac{\beta}{2}\cos\frac{\alpha}{2}\cos\frac{\gamma}{2}$$

$$z = -\sin\frac{\alpha}{2}\sin\frac{\beta}{2}\cos\frac{\gamma}{2} + \sin\frac{\gamma}{2}\cos\frac{\alpha}{2}\cos\frac{\beta}{2}$$

- z-x-y intrinsic / y-x-z extrinsic rotate

$$q = q_z q_x q_y$$

$$w = -\sin\frac{\alpha}{2}\sin\frac{\beta}{2}\sin\frac{\gamma}{2} + \cos\frac{\alpha}{2}\cos\frac{\beta}{2}\cos\frac{\gamma}{2}$$

$$x = \sin\frac{\alpha}{2}\cos\frac{\beta}{2}\cos\frac{\gamma}{2} - \sin\frac{\beta}{2}\sin\frac{\gamma}{2}\cos\frac{\alpha}{2}$$

$$y = \sin\frac{\alpha}{2}\sin\frac{\gamma}{2}\cos\frac{\beta}{2} + \sin\frac{\beta}{2}\cos\frac{\alpha}{2}\cos\frac{\gamma}{2}$$

$$z = \sin\frac{\alpha}{2}\sin\frac{\beta}{2}\cos\frac{\gamma}{2} + \sin\frac{\gamma}{2}\cos\frac{\alpha}{2}\cos\frac{\beta}{2}$$

- z-y-x intrinsic / x-y-z extrinsic rotate

$$q = q_z q_y q_x$$

$$w = \sin\frac{\alpha}{2}\sin\frac{\beta}{2}\sin\frac{\gamma}{2} + \cos\frac{\alpha}{2}\cos\frac{\beta}{2}\cos\frac{\gamma}{2}$$

$$x = \sin\frac{\alpha}{2}\cos\frac{\beta}{2}\cos\frac{\gamma}{2} - \sin\frac{\beta}{2}\sin\frac{\gamma}{2}\cos\frac{\alpha}{2}$$

$$y = \sin\frac{\alpha}{2}\sin\frac{\gamma}{2}\cos\frac{\beta}{2} + \sin\frac{\beta}{2}\cos\frac{\alpha}{2}\cos\frac{\gamma}{2}$$

$$z = -\sin\frac{\alpha}{2}\sin\frac{\beta}{2}\cos\frac{\gamma}{2} + \sin\frac{\gamma}{2}\cos\frac{\alpha}{2}\cos\frac{\beta}{2}$$

Quaternion => Euler angles

It's quite easy to transform quaternion to Euler angles, since each element in rotation matrix can be represent by quaternion.

Application

Quaternion

- inner rotation representation in 3D software
- interpolation
- optimization

Euler Angles

- user insterface in 3D editor
- FPS camera

Reference

- Quaternion and 3d rotation
- Definition of Euler angles

- Transform rotation matrix to quaternion
- Transform quaternion to rotation matrix
- Transformation rotation matrix to Euler angles