

Representations of Rotations

Several methods may be used to represent rotations: rotation matrices, axis/angle (which may be represented as unit quaternions), and Euler angles. The table below highlights some pros and cons for these representations.

	<i>Rotation Matrices</i>	<i>Axis/Angle</i>	<i>Euler Angles</i>
<i>Size</i>	9 numbers	4 numbers	3 numbers
<i>Composition</i>	Easy (multiplication)	Easy in quaternion representation (multiplication)	?
<i>Interpolation</i>	?	Visually well functioning methods exist in quaternion representation (slerp, squad)	Methods not visually pleasing
<i>Intuitive?</i>	No	Yes	To some extent
<i>Caveats</i>		Negation of axis and angle gives same rotation	Non-uniqueness of representation, gimbal lock

Note that the above table discusses representations of rotations at the application programming level. For use on the GPU, all rotations must be expressed as a matrix (in homogeneous coordinates) in the end.

The book contains formulas for converting between the various representations (axis/angle \Leftrightarrow rotation matrix \Leftrightarrow Euler angles).