OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

• Laplacian eigenfunctions and neural networks:... Quaternion algebra might come across as being too complicated for performing well-understood geometrical transformations in 3D space. While quaternion algebra is not routinely used in 3D geometric computations, this algebra has several characteristics that can make it a very desirable "alternative algebra tool" for performing various tasks encountered in 3D geometric data processing, transformation, interpolation, registration, analysis, characterization and classification. We summarize some of the desired benefits of quaternion algebra in the following list.

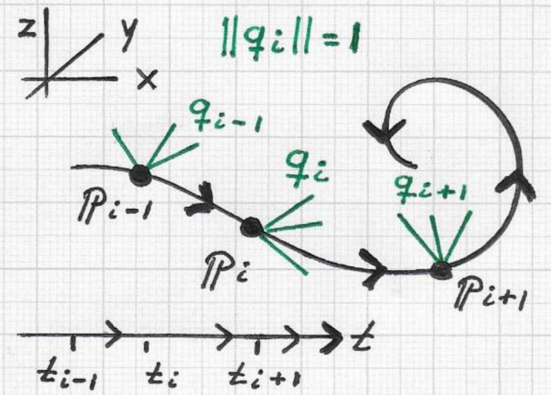
• **Characteristics and advantages of quaternions for 3D geometric data processing:**

1) **Orthogonal rotation matrices "store" redundant information and, when multiplying them, can introduce numerical errors that are not acceptable. Quaternions "encode" rotations in a "compact format" (axis and angle) and make possible storage/memory savings, efficient multiplication and reduction of numerical error values.**

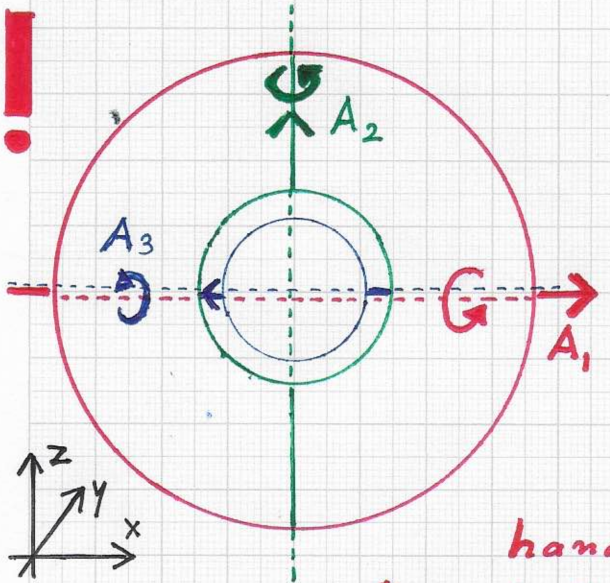
...

OBJECT AND MATERIAL EIGENFUNCTIONS - Cont'd.

- Laplacian eigenfunctions and neural networks: ...



2) Quaternions support "smooth and natural" interpolation of key frame orientations. See left figure for an example. Applications in robotics or computer animation often have to interpolate positions/points (p_i) with associated orientation information, to be blended continuously and "naturally"; quaternions (q_i) are ideal for representing the given orientations for time points t_i.



3) Engineered control systems used in many applications (robotics, aerospace, drones, ...) and sensor data processing (gyroscope data, ...) must be able to handle or avoid the so-called "gimbal lock" problem. The left figure shows a 3-axis inter-connected "rotational system" that can get locked in a singular axis configuration. Use quaternions!

...

OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

- **Laplacian eigenfunctions and neural networks:...** Discussing the "gimbal lock" problem in mathematical detail is beyond the scope of this high-level summary. Many papers discuss this problem with mathematical rigor. The important point is this:

! By using quaternions for the representation and geometrical processing of orientations, rotations, axes, local coordinate systems etc., the "gimbal lock" problem cannot arise.

! 4) Quaternions make it possible to represent orientation information with minimal storage requirements. Using a 3×3 matrix to represent an object's orientation, 9 values must be stored. The quaternion $q_{rot} = \cos\frac{\alpha}{2} + (i d_x + j d_y + k d_z)\sin\frac{\alpha}{2}$ that requires one to store only an angle value (α) and unit axis direction vector ($\mathbb{d} = (d_x, d_y, d_z)^T$) suffices to capture orientation of an object. The representation of a unit vector $\mathbb{d} = (d_x, d_y)^T$ in the complex plane is $d = \cos\alpha + i\sin\alpha = d_x + i d_y$

One can view the quaternion representation $q_{rot} = \cos\frac{\alpha}{2} + \langle \mathbb{i}, \mathbb{d} \rangle \sin\frac{\alpha}{2}$, $\langle \mathbb{i}, \mathbb{d} \rangle = i d_x + j d_y + k d_z$, as a "generalization": from the 2D complex plane to 4D complex quaternion space.

...

■ OBJECT AND MATERIAL EIGENFUNCTIONS - Cont'd.

• Laplacian eigenfunctions and neural networks:... **5)** Quaternions of the form $x_0 + i0 + j0 + k0$ represent real numbers ("scalar quaternions"), and quaternions of the form $0 + ix_1 + jx_2 + kx_3$ represent vectors, also points interpreted as positional vectors ("vector quaternions"). Relevant formulas are:

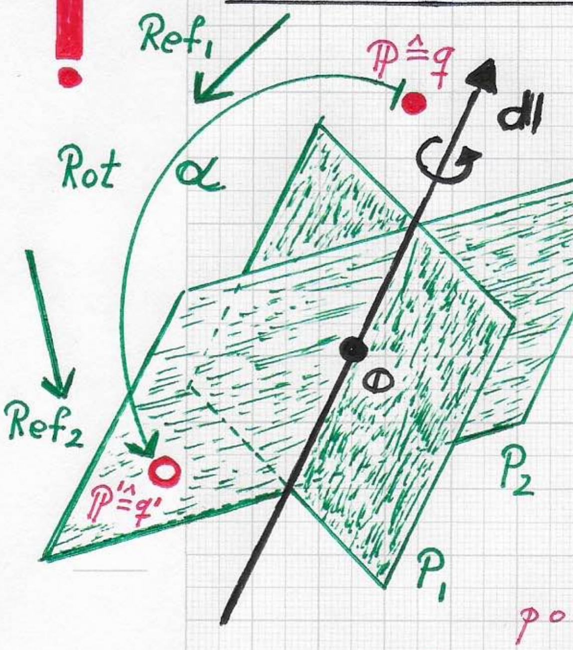
!

- **Norm.** The norm of quaternion $q = x_0 + ix_1 + jx_2 + kx_3$ is $\|q\| = (q \cdot \bar{q})^{1/2} = (x_0^2 + x_1^2 + x_2^2 + x_3^2)^{1/2}$.
- **Distance.** The distance between $q_1 = x_0 + ix_1 + jx_2 + kx_3$ and $q_2 = y_0 + iy_1 + jy_2 + ky_3$ is the norm of $\|q_1 - q_2\|$, i.e.,
$$\|(x_0 - y_0) + i(x_1 - y_1) + j(x_2 - y_2) + k(x_3 - y_3)\| = \\ = \left((x_0 - y_0)^2 + (x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2\right)^{1/2}.$$
- **Dot product.** The dot product of vector quaternions $q_1 = ix_1 + jx_2 + kx_3$ and $q_2 = iy_1 + jy_2 + ky_3$ is the negative value of the real part of $q_1 q_2$:
$$\begin{aligned} q_1 q_2 &= (ix_1 + jx_2 + kx_3)(iy_1 + jy_2 + ky_3) \\ &= -x_1 y_1 + ij\, x_1 y_2 + ik\, x_1 y_3 + ji\, x_2 y_1 - x_2 y_2 + jk\, x_2 y_3 \\ &\quad + ki\, x_3 y_1 + kj\, x_3 y_2 - x_3 y_3 \\ &= -(x_1 y_1 + x_2 y_2 + x_3 y_3) + k x_1 y_2 - j x_1 y_3 - k x_2 y_1 + i x_2 y_3 + j x_3 y_1 - i x_3 y_2 = \\ &= -(x_1 y_1 + x_2 y_2 + x_3 y_3) + i(x_2 y_3 - x_3 y_2) + j(x_3 y_1 - x_1 y_3) + k(x_1 y_2 - x_2 y_1). \end{aligned}$$
$$\Rightarrow q_1 \bullet q_2 = x_1 y_1 + x_2 y_2 + x_3 y_3.$$
- **Cross product.** The cross product of vector quaternions $q_1 = ix_1 + jx_2 + kx_3$ and $q_2 = iy_1 + jy_2 + ky_3$ is the vector defined by the vector part of $q_1 \cdot q_2$, i.e., the expression
$$i(x_2 y_3 - x_3 y_2) + j(x_3 y_1 - x_1 y_3) + k(x_1 y_2 - x_2 y_1) = \begin{vmatrix} x_1 & y_1 & i \\ x_2 & y_2 & j \\ x_3 & y_3 & k \end{vmatrix}.$$

...

OBJECT AND MATERIAL EIGENFUNCTIONS – Cont'd.

- Laplacian eigenfunctions and neural networks:...



6) Generalizations. First, we re-consider the possibility to decompose a rotation into two reflections, see left figure. Rotation by an angle α around an axis passing through the origin of the underlying 3D coordinate system and defined by a unit direction vector $\mathbf{d}$ can be performed via concatenation: Mapping a point $\mathbb{p}$ to $\mathbb{p}'$ is achieved by reflecting $\mathbb{p}$ with respect to plane P_1 and subsequently plane P_2. A single reflection alters the orientation of the reflected object, which is an interesting transformation in its own right. It is not obvious whether this concatenation of reflectione — using quaternion representations of objects — can potentially be advantageous for recognition/classification. (See also page 5, 12-20-2023.)

Second, octonions (Cayley numbers) of the form $\mathbf{o} = x_0 + i_1 x_1 + i_2 x_2 + \ldots + i_7 x_7 = x_0 + \langle \mathbb{i}, \mathbb{x} \rangle$ — and more general hyper-complex numbers — offer even more involved representations and algebraic operations for objects. Again, it is not known whether such generalizations are useful for classification.

...