*I know maths can be a bit boring for some, so I’ve added in some abusive humour in there to keep you entertained while reading it.*

Any sequence of rotation of a sphere, no matter how complicated, can be summarized into rotation about a single axis.  
Not-very-rigorous proof:

Pick any two arbitrary points on the initial unit sphere, r1’ and r2’, and track their landing points after the sequence of arbitrary rotations. Say, they landed on the points r1’’ and r2’’.

There is a ring on which every point could’ve acted as the pivot that rotates r1’ to r1’’. This ring is the equatorial plane to r1’ and r1’’ (bisects the sphere at the mid-point of r1’ and r1’’).   
Drawing 1

(Go on, try as hard as you want, but you’ll not be able to make this plane nonequatorial.)

The same goes for r2’ and r2’’. It should be intuitive that these two points are sufficient to represent the pose of the entire sphere before and after the sequence of rotations.

These two rings must intersect at two points (or more. If they intersect at more than 2 points (i.e. overlaps entirely, you’ve chosen the wrong points 🙃 you little shit. Go back and reconsider your choices of points. And reconsider your life choices while you’re at it.)

You’d imagine that each of these two points is a possible pivot to rotate the sphere from the original orientation to the final orientation. And you’d be right. But why two pivot points instead of one you’d ask? If you’re a smartass you would’ve probably figured out that these two pivot points must be polar opposites of each other.

But if you’re like me, who’s totally not a dumbass and has voluntarily did four pages of maths just to realize ¾ way through that this is true, you can notice this by the fact that these two rings are essentially, each an equatorial plane intersecting the unit sphere. They must therefore each pass through the origin. Now consider only these two equatorial planes. They intersect each other at one line only, and this line must pass through the origin also. Therefore, the intersection between these two planes AND the unit sphere are the two points that are polar opposite to each other, because the origin and these two pivot points are colinear.

Eql plane 1 and eql plane 2

And the reason why rotation around these polar opposite pivot points are complementary is trivial: rotation clockwise around one pivot point is equivalent to anticlockwise rotation in the opposite pivot point.

Any combination of r1’ r2’ tracked is already representative of the entire sequence of rotation.

If you’re a sceptic, convince yourself by drawing a pattern between r1’ and r2’. The same pattern should be found between r1’’ and r2’’.

And so, you can therefore, convince yourself, even if you’ve picked an r3’ on this pattern, and let it fall onto r3’’ after the rotation, you could’ve easily deduced where r3’’ will have landed from r1’’ and r2’’ . In other words, tracking a third point r3’ → r3’’ won’t give you any new information (it’ll only give you redundant information.)

Conversely, the above statement is saying given a particular rotation around a particular axis, only tracking two points’ starting and landing position is sufficient to give enough information to uniquely describe this rotation. Choosing a different starting point shouldn’t matter.

(Doesn’t matter whether your r2’ is over here or over there. Any two points will do.)

I realized something:

You know how in continuum mechanics, you can choose a an orientation that has it’s diagonal elements’ value goes to zero,  
(pure shear stress frame)  
or an orientation that has all non-diagonal elements’ value goes to zero  
(Principal stress frame)  
And any other frames are combinations of these two?

I think the same may apply to Quaternions:  
you can choose a point (think of a small, but still 2-dimensional cross) on the sphere which will only be rotated by the sphere (thus lands back upon itself);  
or choose a frame which is purely translated by the rotation, and is not rotated by the transformation: two tips of the cross remains pointing in the same direction after the transformation.  
?