Subject: Re: Averaging quaternions
Posted by GrahamWilsonCA on Mon, 29 Mar 2004 20:45:02 GMT
View Forum Message <> Reply to Message

I have now made a few attempts at averaging my quaternions. For what it is worth, the quaternion data I have is from a star-tracker on board a satellite and each quaternion represents the inferred spacecraft attitude from an algebraic computation using the positions of 3-6 stars on a CCD.

The paper by Curtis et al. in the 1993 IEEE Annual virtual reality international symposium was probably the easiest for me to actually understand so I started with it. However, their approach only works for averaging 2 quaternions and I have 15-1000 to average in time. Moreover, I need to weight each quaternion by its relative precision (i.e. the number of stars on the CCD and the amount of stray light) at any given time. Both of these points are eluted to in the future work section of this paper but I cannot find any further papers by these people.

From the suggested approach:

> 1. apply to all quaternions a rotation that moves one of them to 1
>   (for example one that is closest to the trivial average),
> 2. orient all results to positive real part,
> 3. average the results
> 4. rotate back the result,
> 5. normalize.

It isn't clear why steps 1 and 4 are required as they can be combined with 2 using a dot product? I'm not entirely sure of that but the following seems to work ok:

```
Quaternion_t AverageQuaternion(int numRotations,
    const Quaternion_t *sourceQ)
{
  Quaternion_t averageQ;
  int i;

  /* initialise result with first rotation */
  averageQ = (sourceQ[0]);

  /* loop over remaining rotations */
  for(i = 1; i<numRotations; ++i)
  {
    /* Check for identical rotations and add to the average quaternion
*/
    if(DotQuaternion(averageQ, sourceQ[i])>0.0f)
```

```
      averageQ = AddQuaternion (averageQ, sourceQ[i]);
    else
      averageQ = AddQuaternion (averageQ,
ReverseQuaternion(sourceQ[i]) );
  }

  /* Normalize */
  return NormalizeQuaternion (averageQ);
}
```

Does anyone have a suggestions on how do I can weight the different quaternions to get a weighted average rotation?

A previous post on averaging rotation matrices suggested:

> I'd suggest transformation of the rotation matrices into
> quaternions. The quaternion coefficients can be regarded as forming a
> unit vector in 4-space. Your observations should give a cluster of
> such vectors. The centroid of this cluster should give the mean
> rotation.

I quite like this description, but I have no idea how to find the centroid of a cluster of vectors in 4-space. Can anyone point me to a book/paper/letter that a mere physicist might be able to understand.

Cheers,
Graham