
Subject: Re: arbitrary rotation of 3-d arrays

Posted by [David Foster](#) on Fri, 11 Jun 1999 07:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

morisset@my-deja.com wrote:

>
> The use of t3d will perform transformation on a coordinate
> cube, but will not 'rotate' the datas. It's like when you want to
> compare 2 images with one turned in respect to the other one. Or if you
> want to make a projection using total, but on an axis other than x or y.
> Then you have to use the ROT idl function to perform interpolation.
>
> I wrote a turn3d code (see below) that perform rotation of the datas
> in a 3D cube. It uses ROT slide by slide.
> It (seems to ;-)) works also with vector fields.
> Hope it helps, send (bug) reports to (witout blanck):
> morisset @ astrsp-mrs.fr

Christophe -

The previous posts discussing the use of T3D were all assuming that once the transformation of coordinates was performed, one would have to then use interpolation to actually transform the data. At least in my post I simply forgot to mention this important step. (oops!)

My main concern with your method is that it is using 2D techniques to perform a 3D task, and I believe will invariably suffer from poorer performance. Both methods will require large amounts of memory, but in your method you make entire copies of the original data-set, and three times! Often the data for such an operation will be huge, as in the case of medical images. The performance penalties this copying will incur, as well as the use of for loops to process each set of 2D images through the data, will make this method much slower than the use of T3D and interpolation.

I would also argue that it would be less accurate, since you are performing interpolation three times in succession basically, once within each 2D plane, whereas the T3D method will transform the coordinates mathematically and then interpolate the original data once from those computed coordinates. Quite a different animal.

Dave Foster

>
> -----Cut here -----
> function turn_3d,a_in,x_angle,y_angle,z_angle,RESIZE = resize,\$
> CONSERV = conserv,VERBOSE = verbose,HELP=help,_extra= _extra,vect=vect
> ;+

```

> ; NAME:
> ;     turn_3d
> ;
> ; CALLING SEQUENCE:
> ;     result = turn_3d(a,x_angle,y_angle,z_angle)
> ;
> ; PURPOSE:
> ;     Rotate a 3D array. It applies the ROT IDL function to each
> ;     2D sub_array of A. The computation is done in a 50% bigger
> ;     cube to assure that nothing will be lost.
> ;     If A is a structure, apply recursively turn_3d on all the
> ;     tags of the structure.
> ;
> ; INPUT PARAMETERS:
> ;     A =     The 3D array to be rotated. This array may be of any type
> ;             expected string. Can be a structure.
> ;     X, Y, Z_ANGLE =
> ;             3 angles of rotation in degrees CLOCKWISE.
> ;     WARNING: in case of multiple rotation, the order is Z, X and Y.
> ; KEYWORDS:
> ;     VECT : Setting this keyword if A is a 3D vector field
> ;     _extra will be passed to ROT:
> ;
> ;     RESIZE: Setting this keyword to resize the result to the maximum
> ;             size (x,y or z-one) of A. The resizing is NOT a rebining,
> ;             it extracts a 3D sub-array of the big 3D array in which
> ;             the computation is done.
> ;             If A is a structure, RESIZE is set.
> ;     CONSERVE: Setting this keyword to assure that
> ;             total(result) = total(A).
> ;
> ;     VERBOSE: Setting this keyword will print the ratio of the
> ;             sizes of the input array and the result. Works only if
> ;             RESIZE not set. If A is a structure, will say what is
> ;             being rotated.
> ;     HELP: print the calling sequence
> ;
> ; LIMITATIONS: They are those of ROT... For small dimensions arrays,
> ;     a rotation of +10deg followed by a rotation of -10deg will NOT
> ;     give you back the input data.
> ;
> ; BUGS: If A is a structure of arrays NON cubics (s(1) = s(2) = s(3),
> ;     then it crash!
> ;
> ; AUTHOR
> ;     Christophe MORISSET, 1997. morisset @ iagusp.usp.br
> ;
> ; HISTORY:

```

> ; 15-9-97 Post for me by D. Fanning on comp.lang.idl-pvwave
> ; 19-9-97 Add the HELP keyword
> ; 26-9-97 Add the possibiliy for A to be a structure
> ; Suppretion of CUBIC keyword
> ; 28-4-98 pass all the _extra to rot
> ; 19-1-99 add the vector facility (and keyword)
> ;-

<code deleted>

--

~~~~~  
David S. Foster      Univ. of California, San Diego  
Programmer/Analyst   Brain Image Analysis Laboratory  
foster@bial1.ucsd.edu   Department of Psychiatry  
(619) 622-5892      8950 Via La Jolla Drive, Suite 2240  
                    La Jolla, CA 92037  
~~~~~
