Subject: collapsing 3-d arrays
Posted by deb on Wed, 10 Jul 1996 07:00:00 GMT
View Forum Message <> Reply to Message

I have a pretty large 3-D data array (~10 million pts). I'd like to collapse
the data and look at a 2-d image that contains the max value encountered
along the third coordinate. I can write a nested loop that
looks thru the array and pulls out the max data value along the z-axis for each
x,y coordinate pair, but that's pretty cumbersome. It seems like IDL should have
a built-in function for doing this kind of thing, but i can't seem to find it.
The project_vol command looks like it might do what i want but i can't seem to
convince it to give projections along the (x,y), (x,z) and/or (y,z) planes..it
does some weird pseudo-isometric thing. Does anyone have any hints on how to
accomplish this efficiently within IDL?

Subject: Re: collapsing 3-d arrays
Posted by hahn on Thu, 11 Jul 1996 07:00:00 GMT
View Forum Message <> Reply to Message

deb <summa@lanl.gov> wrote:

> I have a pretty large 3-D data array (~10 million pts). I'd like to
> collapse
> the data and look at a 2-d image that contains the max value encountered
> along the third coordinate. I can write a nested loop that
> looks thru the array and pulls out the max data value along the z-axis
> for each
> x,y coordinate pair, but that's pretty cumbersome. It seems like IDL
> should have
> a built-in function for doing this kind of thing, but i can't seem to
> find it.

The IDL function is MAX. It accepts an array as argument and returns
a scalar. Thus you need two nested loops to make a 2-d matrix.

Let's assume your 3-D data array is named D3, you may write:

si = size (D3)

D2 = fltarr(si(1),si(2), /nozero)

```
for i=0,si(1)-1 do begin
  for j=0,si(2)-1 do          D2(i,j) = max ( D3(i,j,*) )
endfor
```

Hope this helps
Norbert

Subject: Re: collapsing 3-d arrays
Posted by paritosh on Sun, 14 Jul 1996 07:00:00 GMT
View Forum Message <> Reply to Message

Perhaps VOXEL_PROJ with maximum_intensity keyword
will be appropriate built-in function.

Paritosh
