Subject: hist_nd for creative grid resampling on big arrays
Posted by MarioIncandenza on Tue, 09 May 2006 20:19:40 GMT
View Forum Message <> Reply to Message

Hey IDL Geniuses!

It starts with one of these:

```
> help, MyBigGrid
> MYBIGGRID    BYT             [43200,21600]
```

That's 30-second data over the whole globe. For my application, I'd like to have a 1-degree grid where each point is the histogram of values within that area. With unlimited memory, I could do something like this:

```
>  Xs1 = (lindgen(43200)/120.)-180+(1/120.); longitudes of columns
>  Ys1 = (lindgen(21600)/120.)-90+(1/120.); latitude of rows
>  Xs2 = rebin(Xs1,43200,21600); longitude of cells
>  Ys2 = rebin(transpose(Ys1),43200,21600); latitude of cells
>  EndProduct =  hist_nd([[[Xs2]],[[Ys2]],[[MyBigGrid]]],[1.,1.,1],min=[-180.
,-90.,0],max=[179.999,89.999,255])
```

Due to the size of the arrays, I don't have enough memory to quite pull this off. But maybe there is some way to trick the histogram routine into parcelling MyBigGrid according to Xs1|Ys1 without having to put Xs2|Ys2 into memory. What do you think?

The workaround looks like this:

```
>  EndProduct = lonarr(360,180,256)
>  for ix=0,359 do for iy=0,179 do EndProduct[ix,iy,*] = $
>          histogram((MyBigGrid[((ix*120):((ix*120)+119)),*])[*,((iy*12 0):((iy*120)
>                                  +119))],min=0,max=100,bin=1)
```

My workaround has been running for a few minutes now. It probably worked, but I'd like something more elegant (and faster) as this problem comes up all the time.

Any histogram masters out there, a faster solution to this general class of problem would be a big help.