
Subject: Re: hist_nd for creative grid resampling on big arrays

Posted by [JD Smith](#) on Tue, 09 May 2006 22:47:08 GMT

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

On Tue, 09 May 2006 13:19:40 -0700, Ed Hyer wrote:

```
> 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*120):((iy*120)
>> +119))),min=0,max=100,bin=1)
```

Since you have a fixed original grid, you don't need HISTOGRAM at all. You already know where the data that need to be combined are: they are adjacent in 120x120 block sizes. I presume MYBIGGRID is just a set of values you want to average over, and by "each point is the histogram of values within that area", you just mean average of values in each new large bin? If so, the easiest (and possibly quickest) approach is just to use REBIN directly:

```
small=rebin(mybiggrid,size(mybiggrid)/120L)
```

If you really mean to capture the full histogram of values in each bin, to create a 360x180x100 element array, you could use the fact that a long could hold everything in one place, and simply "tag" individual grid elements by adding 100L (assuming values really ranged from 0-99, which you should enforce a-prior), e.g.:

```
s=size(grid,/DIMENSIONS)
h=histogram(grid+100L*lindgen(s),MIN=0,MAX=100*n_elements(grid)-1L)
h=reform(h,[100,s],/OVERWRITE)
```

Now, obviously you can't use this trick on your full grid (unless you happen to have 8 or 16BG of RAM), but you can do 1/8x1/4 of the full grid at a time (depending on memory), accumulating the results (personally I'd write them to file for later recovery, in case of error), and stitch them back together later.

JD
