Subject: Re: hist_nd to place just one point on a grid
Posted by JD Smith on Tue, 24 Apr 2007 23:58:32 GMT
View Forum Message <> Reply to Message

On Tue, 24 Apr 2007 15:10:15 -0700, Ed Hyer wrote:

> IDL Wizards,
>
> I use hist_nd a great deal. One of the important things I do with it
> is match gridded maps to point data sets and vice versa. Like this:
>
> IDL> histogram = hist_nd(transpose([[point_x],[point_y]]),
> [dx_grid,dy_grid],min=[minx_grid,miny_grid],max=[maxx_grid,m axy_grid],reverse_indices=ri)
>
> followed by:
>
> IDL> matched_pts = point_x * 0.0; initialize array to hold sampled
> output
> IDL> for i=0l,n_elements(histogram)-1 do if(histogram[i] gt 0) then
> matched_points[ri[ri[i]:(ri[i+1]-1)]] = gridded_data[i]
>
> or:
>
> IDL> grid_totals = gridded_data * 0.0; initialize array to hold
> gridded output
> IDL> for i=0l,n_elements(histogram)-1 do if(histogram[i] gt 0) then
> grid_totals[i] = total(point_data[ri[ri[i]:(ri[i+1]-1)]])
>
> My problem is that hist_nd is unhappy when the inputs [point_x] and
> [point_y] contain only one point. I could code an exception for this
> case, but surely there is a more elegant solution?

Glad you're getting some good use. I presume you're up to date with the version from a month back? If not,

http://turtle.as.arizona.edu/idl/hist_nd.pro

HIST_ND wants an NxP input array. This is to warn against accidentally sending it: [point_x,point_y] or some such. The problem with your method of forming an NxP array with a single point is easily illustrated:

```
IDL> x=[1] & y=[2]
IDL> print,transpose([[x],[y]])
IDL> help,transpose([[x],[y]])
<Expression>    INT       = Array[2]
```

IDL has happily trimmed your final shallow dimension for you, and

HIST_ND can no longer tell that you've intended a 2x1 array. To rememdy this, you'll need to add that trailing dimension back on with REFORM. I suppose I could do this for you inside of HIST_ND, but then it would fail to trap accidental usage of a pure vector input.

JD
