Subject: Re: Positions in 3-d
Posted by Xavier Llobet on Fri, 29 Apr 2005 18:35:59 GMT
View Forum Message <> Reply to Message

In article <1114797354.327497.219120@f14g2000cwb.googlegroups.com>,
panblosky@gmail.com wrote:

>   Hi, I have the following problem. I have a 3xn array, where n can go
> from 32000 to 16.000.000. This array represents positions in space, or
> just lets say x,y,z. The numbers go from 0 to 1. I have a cube of sides
> 1. I divide that cube into a 3-D grid, where my gridsize can go from 32
> to 512 in every direction (depends on how big I want the grid). So, in
> 1-D, the box is going to be divided in:
>
> lon=findgen(n0)/float(n0-1)*float(boxsize)/boxsize
>
> where n0 is the size of the grid (for example, 128) and boxsize is 1.
> The same thing goes for the other two dimensions.
>   Now, I want to find what points (x,y,z) lies in which gridcell
> (between lon[i+1] and lon[i] in every direction).
>   If I do it with a for loop (together with a where), it will take for
> ever. I have tried sorting, but I just can't get it right. Does
> somebody knows a fast way?
>   Thanks,
>
> Pablo

Look at the HISTOGRAM function, REVERSE_INDICES keyword.

--
_xavier
--
Only one "o" in my e-mail address
--
A: Because it messes up the order in which people normally read text.
Q: Why is top-posting such a bad thing?
A: Top-posting.
Q: What is the most annoying thing on usenet and in e-mail?

Subject: Re: Positions in 3-d
Posted by K. Bowman on Fri, 29 Apr 2005 21:45:36 GMT
View Forum Message <> Reply to Message

In article <1114797354.327497.219120@f14g2000cwb.googlegroups.com>,
panblosky@gmail.com wrote:

> Hi, I have the following problem. I have a 3xn array, where n can go
> from 32000 to 16.000.000. This array represents positions in space, or
> just lets say x,y,z. The numbers go from 0 to 1. I have a cube of sides
> 1. I divide that cube into a 3-D grid, where my gridsize can go from 32
> to 512 in every direction (depends on how big I want the grid). So, in
> 1-D, the box is going to be divided in:
>
> lon=findgen(n0)/float(n0-1)*float(boxsize)/boxsize
>
> where n0 is the size of the grid (for example, 128) and boxsize is 1.
> The same thing goes for the other two dimensions.
> Now, I want to find what points (x,y,z) lies in which gridcell
> (between lon[i+1] and lon[i] in every direction).
> If I do it with a for loop (together with a where), it will take for
> ever. I have tried sorting, but I just can't get it right. Does
> somebody knows a fast way?
> Thanks,
>
> Pablo

This sounds like a job for ... HISTOGRAM!

It's a bird, it's a plane, no, it's HISTOGRAM!

Compute the indices of the boxes containing each point. Then use HISTOGRAM on the indices (with REVERSE_INDICES).

David has JD's HISTOGRAM tutorial at
http://www.dfanning.com/tips/histogram_tutorial.html.

Ken Bowman

Subject: Re: Positions in 3-d
Posted by panblosky on Mon, 02 May 2005 16:05:11 GMT

View Forum Message <> Reply to Message

Thanks for your help Ken. It didn't occur to me to use histogram...
But now I have another question: using histogram (and reverse_indices) tells me in which bin the x (or y or z) coordinate would be (and how many x-points are in the bin), but how do I know where the point (x,y,z) lies? I mean, if my cube goes from 0 to 1, and I have 4 bins (it could be more) in each dimension (so I would have 64 sub-cubes in 3-D), how can I tell, in a fast way, in which sub-cube does the point (x,y,z) lies and how many points are in that sub-cube?
Maybe there is an easy answer, but I haven't been able to do it...
Thanks,

Pablo

Subject: Re: Positions in 3-d
Posted by K. Bowman on Mon, 02 May 2005 17:38:58 GMT
View Forum Message <> Reply to Message

In article <1115049911.382768.8960@o13g2000cwo.googlegroups.com>,
panblosky@gmail.com wrote:

> Thanks for your help Ken. It didn't occur to me to use histogram...
> But now I have another question: using histogram (and
> reverse_indices) tells me in which bin the x (or y or z) coordinate
> would be (and how many x-points are in the bin), but how do I know
> where the point (x,y,z) lies? I mean, if my cube goes from 0 to 1, and
> I have 4 bins (it could be more) in each dimension (so I would have 64
> sub-cubes in 3-D), how can I tell, in a fast way, in which sub-cube
> does the point (x,y,z) lies and how many points are in that sub-cube?
> Maybe there is an easy answer, but I haven't been able to do it...
> Thanks,
>
> Pablo

Assume you have a 3-D space that you divide into a regular grid of nx x ny x nz boxes. The coordinates of the space range from [xmin, xmax], [ymin, ymax], [zmin, zmax]. The box sizes for each dimension are dx = (xmax - xmin)/(nx -1), ...

You have N points with coordinates (x, y, z), and you want to know within which box each point lies.

For the x-dimension, for example, the index of the grid box containing a point is

```
i = LONG(dx*(x - xmin))
j = LONG(dy*(y - ymin))
k = LONG(dz*(z - zmin))
```

The trick is to index the 3-D grid of boxes with a 1-D index:

```
m = i + (j*nx) + (k*nx*ny)
```

The index m ranges from 0 to (nx*ny*nz)-1. Use HISTOGRAM and REVERSE_INDICES on the array of m's (BINSIZE = 1, MIN = 0, NBINS = nx*ny*nz). There will be one m for each point. Histogram will tell you how many points in each box, and reverse indices tells you which points.

You can use the ARRAY_INDICES function to convert from m back to (i, j, k).

Ken Bowman