lloyd@evilprofessor.co.uk writes:

> Hi
>
> Can anyone help me speed up this piece of code. I've hear that using
> histograms might be good, any ideas?
>
> This piece of code takes a point centered at [pixel_z,pixel_y] and
> looks for all detections within a 'radius'. The position of these
> detections are defined by two arrays, z and y, giving their
> coordinates. After this other things are done but this isn't a problem.
> The really slow part is the pt1 where section. This identifies from the
> arrays of 200,000 points, the points which lie in a box +-radius in y
> and z. Then the coners are removed using then next where condition.

If your points are regularly spaced, then you must be able to use the
IDL array indexing syntax instead of WHERE.

If the Y samples are irregularly spaced, but constant as a function of
Z, you should still be able to do a one dimensional selection on Y
instead of a 2-d selection, and then rely on the array indexing syntax.

If your points are completely irregularly spaced in both dimensions,
then I don't think there are any symmetries to exploit, and you'll
have to use WHERE.

Good luck,
Craig

>
> for pixel_y=uint(min(y)),roundup(max(y))-1 do begin
>   for pixel_z=uint(min(z)),roundup(max(z))-1 do begin
>   pt1 = where(y ge pixel_y-radius and y le pixel_y+radius and z ge
> pixel_z-radius and z le pixel_z+radius)
>   if pt1[0] ne -1 then begin
>   b = where((y[pt1]-pixel_y)^2+(z[pt1]-pixel_z)^2 le radius^2)
>   if b[0] ne -1 then begin
>   a = pt1[b]
>   w = weight(SQRT((y[a]-pixel_y)^2+(z[a]-pixel_z)^2),FWHM/3.0)
>   m[(pixel_y-uint(min(y))),(pixel_z-uint(min(z)))] = total(w*s[a])
>   endif
>   endif
> endfor

> endfor
>
>
> Any help would be greatly appriciated
>
> Lloyd
>

--
 --- -------------
Craig B. Markwardt, Ph.D.   EMAIL: craigmnet@REMOVEcow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
 --- -------------