Subject: Speeding up a call to the where function
Posted by nathan12343 on Tue, 04 Mar 2008 21:08:31 GMT
View Forum Message <> Reply to Message

Hi all-

Let's first try and explain what this bit of code does. I have an array, centerdist, a 512 X 512 array which represents the distance form any particular pixel to some point of interest in the 512 X 512 window. I also have an array of radii, rads, and I want to find the indices which have a centerdist value between two particular radii values in rads. I also have a 512 X 512 mask array which tells me which pixels to ignore. I find the pixels that aren't masked out and are the correct distance from the point of interest, and I save their indices in heap variables referenced by an array of pointers, annpix. The code looks like this:

```
centerdist=sqrt((xdist-xc)^2 + (ydist-yc)^2)
rads=(dindgen(nrads+1))*info[j].radius^2/10
rads=sqrt(rads)
FOR anNum=0,nrads-2 DO BEGIN
        startwh=systime(1)
        whpix=where(centerdist LE rads[annum+1] AND centerdist GT
rads[annum] AND $
                mask eq 1)
        IF (whpix NE [-1]) THEN BEGIN
                *annpix[annum]=[*annpix[annum],whpix]
        ENDIF
ENDFOR
```

The problem is, that where function call takes a prohibitive amount of time and since I need to do this thousands of times, my code is really slow.

Since the only pixels I'm interested in are in a little window (~ 30 X 30 pixels instead of 512 X 512 pixels), I don't really need to search through all of centerdist. I would have to convert the indices the where function returns using a smaller window to the indices I get now and I have no idea how to do this.

Does anyone know how one might go about speeding my code up?

Thanks for your help, I'll try and answer any questions you might have.

-Nathan