Subject: Re: Rapid "moving windows" access in IDL?
Posted by Craig Markwardt on Fri, 30 Jan 2004 22:55:26 GMT
View Forum Message <> Reply to Message

"Christopher Lee" <cl@127.0.0.1> writes:
> In article <BC3F7CE8.18DCE%greenberg@ucdavis.edu>, "Jonathan Greenberg"
> <greenberg@ucdavis.edu> wrote:
>
>
>> Ok, now to make this a bit more confusing -- in practice, what I will
>> actually be doing is selecting all pixels at about a given distance from
>> the center point, creating a ring of pixels that will be used for the
>> semivariogram. Getting back to an earlier question, is the array
>> subscripting an inherently slow process, and are there "better" and
>> "worse" ways of accessing array elements given an x,y coordinates? More
>> thoughts?
>> --j
>
> Why not make a ring of 1s and 0s..e.g.
>
> nx=100
> ny=100
>
> dist=shift(dist(nx,ny),nx/2, ny/2)
> mask=fltarr(nx,ny)
> mask[*]=0.0
> outer=25
> inner=23
>
> mask[where(dist lt outer)]=1.0
> mask[where(dist lt inner)]=0.0

This is a good idea. However, for large images you will run into a
problem where you are wasting a lot of time multiplying by zero.

One way around this is to not go as far as Chris said, and preserve
the "where" output, then use that as an offset.

Example:
```
 ;; Find center-point of the image
 wh0 = where(dist EQ min(dist))
 ;; Find the points around the center which are between inner and outer radius
 wh = where((dist LT outer) AND (dist GE inner))

 ;; Compute locations of the ring as a *offset* from the center pixel
 wh = wh - wh0
```

Now you can use this list of offsets, WH, to index into the original

array wherever you want.

Example: You are interested in the ring of points around the position
A[20,30]. Then you can do this:

```
ind_1d = 20 + 30*nx   ;; Compute the 1D index into the array

A_ring = A[ind_1d + wh]
```

Now A_RING contains all the elements within the ring surrounding
20,30, and you can compute the variance of those elements or whatever.

If you want to compute the whole image, well, then you will have to
loop, but at least the innermost loop(s) are vectorized.

Craig

--

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