Subject: Re: Rapid "moving windows" access in IDL?
Posted by Jaco van Gorkom on Sun, 01 Feb 2004 22:20:51 GMT
View Forum Message <> Reply to Message

"Christopher Lee" <cl@127.0.0.1> wrote in message
news:20040130.140913.771151432.24672@buckley.atm.ox.ac.uk...

```
> 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
>
> ;mask contains a ring of 1s
>
> ring = mask * data .
> OR
> ring = data[where(mask eq 1)]
```

There is no really efficient way to acces the elements in a moving window like you need. There is however a fast built-in routine for accessing the (weighted) sum of these elements: CONVOL() . And that might just be enough for you, since your semivariance can probably be rewritten to depend only on the square of the sum over the elements, and on the sum of their squares.

To get an array of the sum over the ring elements for each centre point:

```
  sums = convol(data, mask)
```

The sums of their squares:

```
  sumsq = convol(data^2, mask)
```

There is also a very useful program and discussion on David's website:

http://www.dfanning.com/math_tips/variance.html . The program by Martin Downing on that page uses SMOOTH() for square "moving windows". By plugging in CONVOL() and its keywords you should be able to use rings of elements, or whatever else you might want.

Hope this helps,
Jaco