Subject: Re: Incremental variable names?
Posted by Chris Lee on Wed, 04 Feb 2004 09:34:15 GMT

View Forum Message <> Reply to Message

In article <BC4475B7.18FD5%greenberg@ucdavis.edu>, "Jonathan Greenberg"
<greenberg@ucdavis.edu> wrote:

```
>  I'm trying to optimize this algorithm for use with large images.
>  ENVI/IDL work well if you extract a single line of an image at a time
>  (line = y axis, by the way).  For a circular mask, however, I'd need to
>  extract at least as many lines as the maximum circular diameter (e.g. If
>  my circle is 3 pixels in radius, I need to pull out 7 lines of data to
>  figure out the values along the ring).
>  It occurred to me (and tell me if I'm wrong), that while I CAN extract
>  the line I'm working on +/- 3 lines from the image each time, this seems
>  like an awful waste of I/O, when in reality the next line will use all
>  but one of the previous extracted rows (e.g. I just need to read the
>  next line in, and delete the current top line to free up memory).
>  Creating arrays of the lines would require rewriting the arrays each
>  time (e.g. Arraynew = (arrayold(2:7),newline) -- I know this isn't idl
>  code, but you get the idea) which seems slower than just having 7 named
>  variables (line1,line2,...line7), doing my calculation, then deleting
>  line1, and adding line8 and repeating.
>  No?  Is there a better way of "preserving" the data I've already read
>  from the HD than using incremental variable names?  --j
>  On 2/2/04 1:45 PM, in article
```

If I understand your method correctly.....Create a 2D array with the
dimension of (radius, line length). Each time you read in a new line..

```
array=fltarr(radius, line_length)
input=fltarr(line_length)
openr, file, "filename", /get_lun
counter=0L
cmax=2*radius+1
while(not eof(file)) do begin
 readu,file, input
 array[counter mod cmax,0]=input
 array=shift(array, 1,0) ;;;;This could be wrong, try -1,0...
 result=do_stuff(array)
 counter=counter+1
endwhile

free_lun, file

; I think that would work..
```

Of course, if I understand this whole ring thing, your doing this with every point on your image? If you are, the function DO_STUFF above (your semi-variance code) had better be vectorised along the line_length.

Chris.