Subject: Re: Array manipulation
Posted by John-David T. Smith on Wed, 01 Nov 2000 17:00:24 GMT
View Forum Message <> Reply to Message

Leon Majewski wrote:
>
> Hello
> I was wondering whether any array minded person could suggest a way of using array
> indicies to chop up a large array into ordered windows.
> I can't think of a way to do it with reform, translate (though i'm sure this is my
> limitation not a reform translate limitation)
>
> ---------
> ie given an array of 30*30 elements
> return 100 3*3 elements
> or 36 5*5
> or...
>
> in=
>
> 00 01 02 03 04 05..
> 30 31 32 33 34 35..
> 60 61 62 63 64 65..
>
> out = blocks such as
> 00 01 02
> 30 31 32
> 60 61 62
>
> each block is then processed to one representative number (ie mean or median....) and
> returned
>

If you just want mean or median you should use:

mean=smooth(arr,n)
med=median(arr,n)

where n=3 or 5 or whatever odd box size you want. If you really were only interested in windows centered at [2,2],[5,2], etc., you can trivially extract those elements. Usually the full boxcar is more interesting though.

There are other things you can do without resorting to individually extracting "windows" as you call them. For instance:

; the nxn window total

```
total=smooth(arr,n)*n^2
; the nxn window total not including central pixel
neighbors=smooth(arr,n)*n^2-arr
; the mean of the neighboring pixels (excluding central)
neighmean=(smooth(arr,n)*n^2-arr)/(n^2-1)
; the square deviation from that mean
sqdev=(arr-neighmean)^2
; the variance of an nxn window of data, excluding central pixel
imvar=(smooth(sqdev,n)*n^2-sqdev)/(n^2-2)
```

With repeated applications of smooth you can generate a boxcar of any statistical moment you desire, and I guarantee it will be much faster than your code. See the EDGE_TRUNCATE keyword for dealing with edge issues. Cutoffs can be enforced by zeroing outliers, which will not affect the mean or average, but will affect higher moments.

As a bonus exercise, it's quite possible to do full exclude-certain-value (above max, below min, NaN, whatever) boxcar statistics using the same method. Precompute a mask of included elements, and use it judiciously throughout the computations. How can you count good elements in a given window? Simply total the mask as above. Be careful of integer truncation effects: (1+1)/3=0. Overcome this with a floating mask. You'll also have to add an fallback step if you expect boxcar windows with fewer valid elements than the highest moment number you wish to compute (i.e. 0 elements for a mean is pretty obvious).

JD

--
J.D. Smith | WORK: (607) 255-6263
Cornell Dept. of Astronomy | (607) 255-5842
304 Space Sciences Bldg. | FAX: (607) 255-5875
Ithaca, NY 14853 |