Subject: Re: efficient kernel or masking algorithm ?
Posted by John-David T. Smith on Wed, 29 Nov 2000 08:00:00 GMT
View Forum Message <> Reply to Message

Richard Tyc wrote:

>
> I need to apply a smoothing type kernel across an image, and calculate the
> standard deviation of the pixels masked by this kernel.
>
> ie. lets say I have a 128x128 image. I apply a 3x3 kernel (or simply a
> mask) starting at [0:2,0:2] and use these pixels to find the standard
> deviation for the center pixel [1,1] based on its surrounding pixels, then
> advance the kernel etc deriving a std deviation image essentially.
> I can see myself doing this 'C' like with for loops but does something exist
> for IDL to do it better or more efficiently ?
>
> Rich

Oh my this is a common topic lately. See my recent posts in a thread
with title "Array Manipulations". Here's the good stuff:

```
; 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)
```

Take a look at the "EDGE*" keywords too, if you care about what happens
near the borders.

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 |