Subject: Re: efficient kernel or masking algorithm ? UPDATE
Posted by John-David T. Smith on Mon, 26 Feb 2001 03:00:04 GMT
View Forum Message <> Reply to Message

Martin Downing wrote:
>
> From thread: http://cow.physics.wisc.edu/~craigm/idl/archive/msg03957.htm l
> a.. Date: Wed, 29 Nov 2000 16:30:54 -0500
> 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
>
> I was wandering through new Craig's IDL archive site (which is brilliant by
> the way) and came across this question asking for an efficient way of
> calculating the loacal standard deviation in an array. It seemed to me that
> the thread had not reached a full solution so perhaps some of you might be
> interested in this method which is very fast. It is based on the crafty
> formula for variance:
> variance = (sum of the squares)/n + (square of the sums)/n*n

Righto. I knew I was fishing for something like this. Except I think you mean:

(population) variance = (sum of the squares)/n - (square of the sums)/n*n

Luckily, that's how you've coded it too. Sample variance (=population variance*n/(n-1)) is of course the more common case in science (as opposed to gambling).

JD

P.S. I think I originally got the idea from sigma_filter.pro, a NASA library routine, dating back to 1991. It's chock-full of other good tidbits too. Thanks Frank and Wayne!