Subject: Re: STANDARD DEVIATON
Posted by Struan Gray on Fri, 04 Aug 2000 07:00:00 GMT
View Forum Message <> Reply to Message

I think this works:

```
; ************************************

function smg_imageSD, image

  fimage = float(image)
  localmean = smooth(fimage, 3)
  sum = (fimage - localmean)^2
  sum = temporary(sum) + $
    shift((fimage - shift(localmean, 1, 1))^2,-1,-1)
  sum = temporary(sum) + $
    shift((fimage - shift(localmean, 0, 1))^2, 0,-1)
  sum = temporary(sum) + $
    shift((fimage - shift(localmean,-1, 1))^2, 1,-1)
  sum = temporary(sum) + $
    shift((fimage - shift(localmean, 1, 0))^2,-1, 0)
  sum = temporary(sum) + $
    shift((fimage - shift(localmean,-1, 0))^2, 1, 0)
  sum = temporary(sum) + $
    shift((fimage - shift(localmean, 1,-1))^2,-1, 1)
  sum = temporary(sum) + $
    shift((fimage - shift(localmean, 0,-1))^2, 0, 1)
  sum = temporary(sum) + $
    shift((fimage - shift(localmean,-1,-1))^2, 1, 1)

  sum = sqrt(temporary(sum)/8)

  dims = size(sum, /dim)
  sum[0,*] = 0.0
  sum[*,0] = 0.0
  sum[dims(0)-1,*] = 0.0
  sum[*,dims(1)-1] = 0.0

  return, sum
end

; ************************************
```

You can generalise the shifting and put it into a double loop over the kernal indices. You can also deal with edge-effects (and avoid the zeroing of edge elements) if you creat an oversize image and pad it appropriately. On my machine this is approx ten times faster than Ben_imageSD.