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

Struan Gray wrote:
 root.
>
>
>   function imageSD, image
>
>   localmean = smooth(float(image), 3, /edge_truncate)
>   localsd = (float(image)-temporary(localmean))^^2
>   localsd = smooth(temporary(localsd), 3, /edge_truncate)
>   localsd = sqrt(temporary(localsd))
>
>   return, localsd
>
>   end
>
>
> Struan

I believe that this gives different results to the inbuilt IDL routines.
The problem is a wee bit more complicated than it looks.

My routine:

```
 ;============================================================ ==
function ben_imageSD, image

siz = size(image,/dim)

standdev = fltarr(siz[0],siz[1])

for i=1,siz[0]-2 do begin
   for j=1,siz[1]-2 do begin
      standdev[i,j] = sqrt(total((image[i-1:i+1,j-1:j+1]-$

total(image[i-1:i+1,j-1:j+1])/9.)^2)/8.)
   endfor          ;tricky bit ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
endfor             ;to do without loops
return,standdev
end
 ;============================================================ ==
```

IDL's version:

```
 ;============================================================ ==
```

```
function IDL_imageSD, image

siz = size(image,/dim)

standdev = fltarr(siz[0],siz[1])

for i=1,siz[0]-2 do begin
   for j=1,siz[1]-2 do begin
      standdev[i,j]=stddev(image[i-1:i+1,j-1:j+1])
   endfor
endfor
return,standdev
end
 ;============================================================ ==
```

They do similar things but with loops. Can't figure out how to do this
without loops!

Thanks,
Ben
