Subject: Smooth and /EDGE_TRUNCATE
Posted by Fabzi on Tue, 29 Apr 2014 14:35:39 GMT
View Forum Message <> Reply to Message

IDLers,

I've lost 3 hours of my life on this. Can someone explain me why the
/EDGE_TRUNCATE keyword has an incidence on the results WITHIN the image,
where the kernel meets no edge?

```
pro test_smooth

  ; make an array with a nan in the center
  array = FINDGEN(5,5)
  array[2,2] = !VALUES.F_NAN

  print, ' Exp 1'
  print, 'Expected', MEAN(array[1:3,1:3], /NAN)
  print, 'No truncate', (smooth(array, 3, /NAN))[2,2]
  print, 'Truncate', (smooth(array, 3, /NAN, /EDGE_TRUNCATE))[2,2]

  ; so far so good. Add a NaN somewhere else
  array[1,1] = !VALUES.F_NAN
  print, ' Exp 2'
  print, 'Expected', MEAN(array[1:3,1:3], /NAN)
  print, 'No truncate', (smooth(array, 3, /NAN))[2,2]
  print, 'Truncate', (smooth(array, 3, /NAN, /EDGE_TRUNCATE))[2,2]

end
```

Thanks a lot!

Fabien

Subject: Re: Smooth and /EDGE_TRUNCATE
Posted by David Fanning on Tue, 29 Apr 2014 16:04:09 GMT
View Forum Message <> Reply to Message

Fabien writes:

> I've lost 3 hours of my life on this. Can someone explain me why the
> /EDGE_TRUNCATE keyword has an incidence on the results WITHIN the image,
> where the kernel meets no edge?
>
> pro test_smooth
>
> ; make an array with a nan in the center

```
>     array = FINDGEN(5,5)
>     array[2,2] = !VALUES.F_NAN
>
>     print, ' Exp 1'
>     print, 'Expected', MEAN(array[1:3,1:3], /NAN)
>     print, 'No truncate', (smooth(array, 3, /NAN))[2,2]
>     print, 'Truncate', (smooth(array, 3, /NAN, /EDGE_TRUNCATE))[2,2]
>
>     ; so far so good. Add a NaN somewhere else
>     array[1,1] = !VALUES.F_NAN
>     print, ' Exp 2'
>     print, 'Expected', MEAN(array[1:3,1:3], /NAN)
>     print, 'No truncate', (smooth(array, 3, /NAN))[2,2]
>     print, 'Truncate', (smooth(array, 3, /NAN, /EDGE_TRUNCATE))[2,2]
>
>  end
```

Not sure this "explains" it, but this section of the documentation might have at least saved you a couple of hours. :^)

Note: Normally, two-dimensional floating-point arrays are smoothed in one pass. If any of the EDGE_* keywords are specified for a two-dimensional floating-point array, the result is obtained in two passes, first for all of the rows, and second for all of the columns. Therefore, the results for points in the interior of the array may differ slightly when any of the EDGE_* keywords are set. This difference will be most pronounced if the array contains NaN values.

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: Smooth and /EDGE_TRUNCATE
Posted by Fabzi on Tue, 29 Apr 2014 16:23:24 GMT
View Forum Message <> Reply to Message

On 29.04.2014 18:04, David Fanning wrote:
> Not sure this "explains" it, but this section of the documentation might
> have at least saved you a couple of hours. :^)

OMG, "obvious" as allways. My opinion surely has absolutely no value, but this means to me that the EDGE_* keywords are more than useless:

they are also dangerous.

Note, also, that CONVOL doesn't have this issue:

```
   print, 'Convol', (convol(array, fltarr(3,3)+1./9, /NAN, $
/EDGE_TRUNCATE, /NORMALIZE))[2,2]
```

Produces the correct results.

Not for the future: "smooth" is ok for using on your last holiday's picures, not for real data filtering ;)

Thanks dave!

Fabien
