Subject: coherence.pro re-re-post (ignore previous re-post)
Posted by Dick Jackson on Thu, 13 May 1999 07:00:00 GMT

View Forum Message <> Reply to Message

(now attaching it *after* changing the Windows settings... sorry, folks!)

Hi again,

eddie haskell wrote:

```
>
>  Dick,
>  [...]
>  As an aside, I had a difficult time extracting the code you attached to
>  your post.  It could easily be my news reader but if others are using
>  similar readers and you have other options for attachments, ...
```

Thanks for letting me know (*blush!*). If this attachment works better, then
my lesson for the day is that in Windows File Types Options panel
(accessible in Windows Explorer/View/Options...) the IDL source file should
have Content Type (MIME): text/plain. Thanks to Dave Foster (who knows how
to attach a file!) for the good example.

Cheers,
--
-Dick

Dick Jackson   Fanning Software Consulting, Canadian Office
djackson@dfanning.com   Calgary, Alberta   Voice/Fax: (403) 242-7398
   Coyote's Guide to IDL Programming: http://www.dfanning.com/

```
FUNCTION coherence, array, width

;;    Returns 'coherence' matrix from 2D input array.
;;    Each element in the result is the Std. Dev. of the
;;    'width' x 'width'-pixel neighborhood from the original image.

;;    Dick Jackson, Fanning Software Consulting
;;    djackson@dfanning.com

size = Size(array, /Dimensions)
IF N_Elements(size) NE 2 THEN $
  Message, 'Input array must be 2-dimensional.'
nx = size[0]
ny = size[1]
IF nx LT 3 OR ny LT 3 THEN $
  Message, 'Input array must have at least 3 rows and 3 columns.'

IF N_Elements(width) EQ 0 THEN width = 3
```

```
;    Make Sum-of-X array (sx) where each element contains the sum of
;    values from the neighborhood of the original array.

kernel = Replicate(1, width, width)     ; kernel for convolution to simply
                                        ; sum all values with equal weight
sx = Convol(Float(array), kernel)

;    Make Sum-of-X-squared array (sx2) where each element contains the sum of
;    *squared* values from the neighborhood of the original array.

sx2 = Convol(array ^ 2.0, kernel)

;    Calculate, for each pixel, the standard deviation of the set of
;    the neighborhood values:
;       SD = Sqrt((Sum(X^2) - (Sum(X)^2 / n)) / (n-1))

n = width ^ 2.0

Return, Sqrt((sx2 - (sx^2 / n)) / (n-1))

END ;; coherence
```

File Attachments

1) coherence.pro, downloaded 86 times