
Subject: Array manipulation

Posted by [majewski](#) on Wed, 01 Nov 2000 05:21:21 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hello

I was wondering whether any array minded person could suggest a way of using array indices to chop up a large array into ordered windows.

I can't think of a way to do it with reform, translate (though i'm sure this is my limitation not a reform translate limitation)

ie given an array of 30*30 elements

return 100 3*3 elements

or 36 5*5

or...

in=

00 01 02 03 04 05..

30 31 32 33 34 35..

60 61 62 63 64 65..

out = blocks such as

00 01 02

30 31 32

60 61 62

each block is then processed to one representative number (ie mean or median....) and returned

What i've used so far is attached below (it does what i want, just slowly)

leon

```
FUNCTION QMean2, DATA, $
```

```
  WIDTH=WIDTH, $
```

```
  MINVAL=MINVAL, $
```

```
  MAXVAL=MAXVAL, $
```

```
  SILENT=SILENT
```

```
  ;+
```

```
  ;NAME:
```

```
  ; QMean2
```

```
  ;
```

```

;PURPOSE:
; Calculate the mean of an array, excluding values 0 and NaN.
;
;CATEGORY:
; Array Manipulation
; Statistics
;
;CALLING SEQUENCE:
; Result = QMean(DATA)
;
;ARGUMENTS:
; DATA
; A 2-dimensional array to be averaged
;
;KEYWORDS:
; MAXVAL
; If the data is above this value it is omitted.
; MINVAL
; Only data greater than this value is included in the mean.
; If this keyword is not set, it defaults to 0.
; WIDTH
; The width of the averaging box. If this keyword is not set then
; it defaults to a value of 4.
;
;OUTPUTS:
; An array holding the average of the input data.
; This array is of size (Xdim/Width, Ydim/Width).
;
;SIDE EFFECTS;
; If the size of the array holding the data is not a multiple of the
; Width of the filter, then the Array is truncated to a size that is a
; Multiple of the Width - The program Outputs a warning if this is the
; case.
;
;EXAMPLES:
; Result = QMean(findgen(20,20), Width = 5)
; Result = QMean(findgen(20,20), Width = 5)
; Result = QMean(findgen(20,20), Width = 15, MAXVAL = 210.3)
;
;MODIFICATION HISTORY:
; Created, Leon Majewski, 3rd August 2000
;
;-

IF Size(MINVAL, /n_elements) EQ 0 THEN MINVAL = 0
IF Size(WIDTH, /n_elements) EQ 0 THEN WIDTH = [4,4]
IF Size(WIDTH, /n_elements) EQ 1 THEN WIDTH = [WIDTH,WIDTH]
IF NOT KEYWORD_SET(SILENT) THEN SILENT = 1 ELSE SILENT = 0

```

```

;start_time = SYSTIME(1)

DATA_IN = DATA
Size_Data = SIZE(DATA_IN)

xdim = Size_Data[1]/Width[0]
xdim_less1 = xdim-1
ydim = Size_Data[2]/Width[1]
ydim_less1 = ydim-1

IF Size_Data[1] NE xdim*width[0] THEN BEGIN
  if silent then begin
    print, 'The x dimension is not a multiple of the specified WIDTH.'
    print, 'Reducing the size of the array from'
    help, DATA_IN
  endif

  DATA_IN = DATA_IN[0:(xdim)*width[0]-1, *]

  if silent then help, DATA_IN
ENDIF

IF Size_Data[2] NE ydim*width[1] THEN BEGIN
  if silent then begin
    print, 'The y dimension is not a multiple of the specified WIDTH.'
    print, 'Reducing the size of the array from'
    help, DATA_IN
  endif

  DATA_IN = DATA_IN[*,0:(ydim)*width[1]-1]

  if silent then help, DATA_IN
ENDIF

IF KEYWORD_SET(MAXVAL) THEN $
  IF MAXVAL GT MINVAL THEN $
    DATA_IN = DATA_IN*(DATA_IN lt MAXVAL)

Size_Data = SIZE(DATA_IN)
n_els = Size_Data[4]

DATA_IN = REFORM(DATA_IN, Width[0], n_els/Width[0], /overwrite)
DATA_IN = REFORM(DATA_IN, Width[0], xdim, Width[1], ydim, /overwrite)

Average = FLTARR(xdim,ydim)

FOR j = 0, ydim_less1 DO BEGIN
  FOR i = 0, xdim_less1 DO BEGIN

```

```
data_sub = REFORM(DATA_IN[* ,i,* ,j])
goodpos = WHERE(data_sub GT MINVAL AND $
  FINITE(data_sub) EQ 1, c_goodpos)

IF c_goodpos NE 0 THEN BEGIN
  data_sub = data_sub[goodpos]
  Average[i,j] = TOTAL(data_sub)/N_ELEMENTS(data_sub)
ENDIF ELSE Average[i,j]=0

ENDFOR
ENDFOR
```

```
; print, SYSTIME(1) - start_time, 's'
```

```
RETURN, Average
END
```

Leon Majewski

Remote Sensing & Satellite Research Group
Curtin University of Technology, Perth, Australia

email: majewski@ses.curtin.edu.au

Subject: Re: Array Manipulation
Posted by [Heinz Stege](#) on Wed, 12 Nov 2008 03:11:33 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Tue, 11 Nov 2008 18:45:07 -0800 (PST), duxiyu@gmail.com wrote:

```
> Hi, all.
> There are two array A[3, 3, m] and B[3, 3, n].
> I want to combine them into a new array C[3, 3, (m+n)].
> The first part of C (C[* , * , 0:(m-1)]) is equal to A and the second
> part of C (C[* , * , m:(m+n-1)]) is equal to B.
> Is there a simple method to get the new array C?
```

Yes, there is:
c=[[[a]],[[b]]]

Heinz
