
Subject: EXPAND.PRO - - - again!
Posted by [zawodny](#) on Thu, 29 Oct 1992 12:11:14 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello again,

Awhile back I posted a new and improved version of EXPAND.PRO only to retract it because it supposedly did not work. I challenged others to figure out why I should not work, but noone responded. I have finally gotten around to working on it only to find that it worked like it was supposed to all along. Since it is much faster than any previous version, I post it here once again. My apologies to those who are tired of this.

Joseph M. Zawodny (KO4LW)
zawodny@arbd0.larc.nasa.gov

NASA Langley Research Center
MS-475, Hampton VA, 23681

```
;+
; NAME:
; EXPAND
; PURPOSE:
; Array magnification
; CATEGORY:
; Z4 - IMAGE PROCESSING
; CALLING SEQUENCE:
; EXPAND,A,NX,NY,RESULT [,MAXVAL=MAXVAL,FILLVAL=FILLVAL]
; INPUTS:
; A Array to be magnified
; NX Desired size of X Dimension
; NY Desired size of Y Dimension
; Keywords:
; MAXVAL Largest good value. Elements greater than this are ignored
; FILLVAL Value to use when elements larger than MAXVAL are encountered.
; Defaults to -1.
; OUTPUTS:
; RESULT Magnified Floating point image of A array (NX by NY)
; COMMON BLOCKS:
; NONE
; SIDE EFFECTS:
; NONE
; RESTRICTIONS:
; A must be two Dimensional
; PROCEDURE:
; Bilinear interpolation.
; MODIFICATION HISTORY:
; Aug 15, 1989 J. M. Zawodny, NASA/LaRC, MS 475, Hampton VA, 23665.
; Aug 26, 1992 JMZ, Added maxval and fillval keywords.
; Sep 30, 1992 DMS of RSI, Improved the bad point location algorithm.
; Please send suggestions and bugreports to zawodny@arbd0.larc.nasa.gov
;-
```

```

pro EXPAND,a,nx,ny,result,maxval=maxval,fillval=fillval

s = size(a)
if(s(0) ne 2) then begin
  print,'EXPAND: *** array must be 2-Dimensional ***'
  retall ; This will completely terminate the MAIN program!!!
endif

; Get dimensions of the input array
ix = s(1)
iy = s(2)

; Calculate the new grid in terms of the old grid
ux = findgen(nx)*((ix-1.)/(nx-1.))
uy = findgen(ny)*((iy-1.)/(ny-1.))

; Interpolate the result
result = interpolate(a, ux, uy, /GRID)

; Are we to look for and ignore bad data?
if(n_elements(maxval) ne 0) then begin
; Find where missing points end up
  bad_pts = interpolate(float(a gt maxval), ux, uy, /GRID)
; The only Non-zero points are those resulting from
; bad points. Get their subscripts in the result
  bad_subs = where(bad_pts, count) ; Any bad points
  if count ge n_elements(result) then goto, out ; All bad
  if n_elements(fillval) le 0 then fillval = -1
; Substitute missing value
  if count gt 0 then result(bad_subs) = fillval
endif
; Done
return
OUT: ; If we had a problem
print,'Entire input array is greater than MAXVAL, ('+strtrim(maxval,2)+')'
return
end

```
