
Subject: Re: CONTTW.PRO and EXPAND.PRO - some bug fixes, enhancements
Posted by [zawodny](#) on Fri, 09 Oct 1992 11:59:37 GMT

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

Oh, one other thing I forgot to mention and should be of interest. I got a modified version of EXPAND from Dave Stern of RSI which I would like to pass on to the net. It employs an interesting trick to avoid all the array indexing I was doing.

Here it is (and thanks again Dave!).

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

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

```
;+
; NAME:
; EXPAND
; PURPOSE:
; Array magnification (CONGRID like except that this really works!)
; 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.
; Not really fast if you have to swap memory (eg. NX*NY is a big number).
; OK Postscript users don't forget that postscript pixels are scaleable!
; 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(a) 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

```
