
Subject: Re: EXPAND.PRO - needed by CONTTW.PRO

Posted by [landsman](#) on Fri, 11 Sep 1992 17:56:00 GMT

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

The EXPAND function given by Joe Zawadony will run faster and with less virtual memory problems if it uses the intrinsic INTERPOLATE function (introduced in V2.2.1). I include below a modified version (but without the keywords) that seems to give the same results. The keywords to account for missing data can be implemented in a straightforward way but I have not done so in this demo version.

-Wayne Landsman landsman@stars.gsfc.nasa.gov

pro EXPAND,a,nx,ny,result

```
;+
; NAME:
; EXPAND
; PURPOSE:
; Array magnification (CONGRID like except that this really works!)
; CATEGORY:
; Z4 - IMAGE PROCESSING
; CALLING SEQUENCE:
; EXPAND,A,NX,NY,RESULT
; INPUTS:
; A Array to be magnified
; NX Desired size of X Dimension
; NY Desired size of Y Dimension
; OUTPUTS:
; RESULT Magnified Floating point image of A array (NX by NY)
; 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.
;     Demo version to show use of intrinsic INTERPOLATE function
;             W. Landsman   Hughes/STX
; Please send suggestions and bugreports to zawodny@arbd0.larc.nasa.gov
;-
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!!!
```



```

;+
; 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 10, 1992 JMZ, converted to use INTERPOLATE function (tnx Wayne!)
; 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 = (ix-1.) * findgen(nx) / (nx-1.)
uy = (iy-1.) * findgen(ny) / (ny-1.)

; Are we to look for and ignore bad data? (can't use KEYWORD_SET here)
if(n_elements(maxval) eq 0) then begin
; NO
result = interpolate(a,ux,uy,/grid)
endif else begin
; YES then calculate the indicies and u-arrays
mx = long(ux)<(ix-2)
my = long(uy)<(iy-2)
uxa = ux # replicate(1,ny)
uya = replicate(1,nx) # uy

;Index vectors to A and RESULT arrays
mxy = (mx # replicate(1L,ny)) + (replicate(long(ix),nx) # my)
ind = lindgen(nx,ny)

; Fill RESULT with fill value, defaulting to -1 if none specified
if(n_elements(fillval) le 0) then fillval = -1.
result = replicate(fillval,nx,ny)

; Remove those elements which would be utilizing "bad" values from A
; Check lower left
m = where(a(mxy) le maxval,num)
if(num eq 0) then goto,out
mxy = mxy(m)
ind = ind(m)
; Check lower right
m = where(a(mxy+1) le maxval,num)
mxy = mxy(m)
ind = ind(m)
; Check upper left
m = where(a(mxy+ix) le maxval,num)
mxy = mxy(m)
ind = ind(m)
; Check upper right
m = where(a(mxy+(ix+1)) le maxval,num)
mxy = mxy(m)
ind = ind(m)

; Interpolate only the points which will not be the fill value
result(ind) = interpolate(a,uxa(ind),uya(ind))
endelse

; Done
return

```

```
OUT: ; If we had a problem
print,'Entire input array is greater than MAXVAL, ('+strtrim(maxval,2)+')'
return
end
```
