
Subject: EXPAND.PRO - needed by CONTTW.PRO
Posted by [zawodny](#) on Fri, 11 Sep 1992 14:48:10 GMT
[View Forum Message](#) <> [Reply to Message](#)

```
;+
; 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.
; 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 ***'
  retail ; 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.)
```

```
; Calculate the indices
```

```
mx = long(ux)
```

```
my = long(uy)
```

```
; Calculate the offsets
```

```
ux = ux - mx
```

```
uy = uy - my
```

```
; Correct the points at edges
```

```
mx(nx-1) = ix-2
```

```
my(ny-1) = iy-2
```

```
ux(nx-1) = 1.
```

```
uy(ny-1) = 1.
```

```
; Create u-arrays
```

```
uxa = ux # replicate(1,ny)
```

```
uya = replicate(1,nx) # uy
```

```
uxy = ux # uy
```

```
; Index to A array
```

```
mxy = (mx # replicate(1L,ny)) + (replicate(long(ix),nx) # my)
```

```
; Index to RESULT array
```

```
ind = lindgen(nx,ny)
```

```
; Are we to look for and ignore bad data?
```

```
if(n_elements(maxval) ne 0) then begin
```

```
; Do we have a value to use for fill?
```

```
; Cannot use KEYWORD_SET because zero is a valid value for FILLVAL
```

```
if(n_elements(fillval) le 0) then fillval = -1.
```

```
; Initialize the RESULT array with fill value
```

```
result = replicate(fillval,nx,ny)
```

```
; Remove those elements which would be utilizing "bad" values in 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)
```

```

if(num eq 0) then goto,out
mxy  = mxy(m)
ind  = ind(m)
; Check upper left
m    = where(a(mxy+ix) le maxval,num)
if(num eq 0) then goto,out
mxy  = mxy(m)
ind  = ind(m)
; Check upper right
m    = where(a(mxy+(ix+1)) le maxval,num)
if(num eq 0) then goto,out
mxy  = mxy(m)
ind  = ind(m)
endif else result = replicate(0.,nx,ny) ; Otherwise make RESULT empty

; Interpolate only the points which will not be the fill value
result(ind) = a(mxy) * (1.-uxa(ind)-uya(ind)+uxy(ind))
result(ind) = result(ind)+a(1+mxy) * (uxa(ind)-uxy(ind))
result(ind) = result(ind)+a(ix+mxy) * (uya(ind)-uxy(ind))
result(ind) = result(ind)+a(ix+1+mxy) * uxy(ind)

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

```
