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) NASA Langley Research Center
zawodny@arbd0.larc.nasa.gov 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
```

Subject: Re: Expand
Posted by davidf on Mon, 05 Jan 1998 08:00:00 GMT
View Forum Message <> Reply to Message

Kelly Dean (krdean@lamar.colostate.edu) writes:

> I am reading some AVHRR LAC data in the Level 1B format. The image
> provides 51 latitude/longitude pairs for each 2048 pixel scan line. They
> are sampled every 40 points starting at point 25 (25, 65, 105,..., 1945,

> 1985, 2025).
>
> I created a simple IDL routine to expand the 51 values to 2048 values
> using FOR loops. I am looking for suggestions to make this more
> efficient.

Uh, why? It's pretty darn fast on my middling machine. :-)

[Code snipped.]

I don't think you can get rid of the outside loop (stepping
between the known points), but you can speed up the inside
loop I guess (interpolating values between the known points)
by doing something like this:

```
FOR j=25, 1985, 40 DO BEGIN
   array2048(j:j+39) = Findgen(40) * $
      ((value[j+39]-value[j])/39) + value[j]
ENDFOR
```

Cheers,

David

David Fanning, Ph.D.
Fanning Software Consulting
E-Mail: davidf@dfanning.com
Phone: 970-221-0438
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: Expand
Posted by Kelly Dean on Tue, 06 Jan 1998 08:00:00 GMT
View Forum Message <> Reply to Message

Kevin Ivory wrote:

> Look into the 'interpolate' documentation and use 'findgen' instead of the
> extrapolation for loops.

Thanks for the tip and code Kevin,

I looked into IDL EXPAND which uses INTERPOLATE to expand a 2-dim array. The sample I provided earlier only did one interpolation per scan line. I am able to create 2-dim array of [51,#scans] for each image file and use EXPAND. However, it modified my other 51 values, but not by much.

I try your suggestions,

Kelly

Subject: Re: Expand
Posted by Kelly Dean on Tue, 06 Jan 1998 08:00:00 GMT
View Forum Message <> Reply to Message

David Fanning wrote:

>> Uh, why? It's pretty darn fast on my middling machine. :-)
>

Thanks for the tip Dave,

I have to do the expansion twice (or three times) for each scan line. There can be from 500 to 1500 scan lines. The sample I provided only did one expansion at a time.

Kelly

Subject: Re: Expand
Posted by Kevin Ivory on Tue, 06 Jan 1998 08:00:00 GMT
View Forum Message <> Reply to Message

Kelly Dean wrote:
>
> I am reading some AVHRR LAC data in the Level 1B format. The image
> provides 51 latitude/longitude pairs for each 2048 pixel scan line. They
> are sampled every 40 points starting at point 25 (25, 65, 105,..., 1945,
> 1985, 2025).
>
> I created a simple IDL routine to expand the 51 values to 2048 values
> using FOR loops. I am looking for suggestions to make this more
> efficient.

Look into the 'interpolate' documentation and use 'findgen' instead of the extrapolation for loops.

I did your job for you: (untested code)

```
FUNCTION INTER2048, array51
;+
; PURPOSE:
;   Interpolate AVHRR LAC data in the Level 1B format to 2048 values.
```

```
;   The image provides 51 latitude/longitude pairs for each 2048 pixel
;   scan line. They are sampled every 40 points starting at point 25
;   (25, 65, 105,..., 1945, 1985, 2025).
;-
  start     = 24
  interval  = 40
  n_pix     = 2048
  n_in      = n_elements(array51) ; should be 51
  last      = start + (n_in-1) * interval ; this is going to be 2024
  array2048 = FLTarr(n_pix)
;
  dvalue = (array51(1) - array51(0)) / interval
; extrapolation to the left
  array2048(0:start-1)    = array51(0) - dvalue * findgen(start)
; interpolation
  array2048(start:last-1) = interpolate(array51, findgen(last-start)/interval)
; extrapolation to the right
  dvalue = (array51(n_in-1) - array51(n_in-2)) / interval
  array2048(last:n_pix-1) = array51(n_in-1) + dvalue * findgen(n_pix-last)

  RETURN, array2048
END
```

--
Kevin Ivory   Tel: +49 5556 979 434
Max-Planck-Institut fuer Aeronomie   Fax: +49 5556 979 240
Max-Planck-Str. 2   mailto:Kevin.Ivory@linmpi.mpg.de
D-37191 Katlenburg-Lindau, GERMANY   http://www.gwdg.de/~kivory2/