Subject: Re: CONTTW.PRO and EXPAND.PRO - some bug fixes, enhancements
Posted by landsman on Fri, 09 Oct 1992 04:57:00 GMT
View Forum Message <> Reply to Message

I recently installed V2.4.0 and noticed that CONGRID now has a /MINUS_ONE keyword. Using both /MINUS_ONE and the /INTERP keyword makes CONGRID work identical to EXPAND.PRO. However, CONGRID still lacks the FILL_VALUE and /MAX_VALUE keywords, so EXPAND.PRO will still be needed.

Incidentally, I was told indirectly that V3.0 will appear soon (in a couple of weeks or so), so that users may not want to bother upgrading to V2.4.0.

Among the interesting upgrades in 2.4.0 are

(1) Compiling a procedure that is active no longer gives an error. Instead there is an automatic RETALL and the program compiles.
(2) The widget help has been rewritten in C so that one can return to the IDL prompt without losing the help display. One can have an IDL help icon working throughout the session.
(3) Under Motif, clicking on an IDL graphics window will no longer take the focus away from the terminal window. (I have yet to verify that this works under VMS Motif.)

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

Subject: Re: CONTTW.PRO and EXPAND.PRO - some bug fixes, enhancements
Posted by zawodny on Fri, 09 Oct 1992 11:53:00 GMT
View Forum Message <> Reply to Message

I guess to each his own. I do admit that the autoscaling was crude at best. I included it, for the same reason PV-WAVE has a point and click interface, there are managers out there who still like to play with data that they know little about ;-) . I will take exception to the alleged "misuse" of keyword_set. The only time n_elements is preferable to keyword_set is when the keyword needs to allow zero to be a "good" value or when you need to know whether the keyword is a scalar or an array. IMHO the first exception is more of a bug. Zero should not be considered a sign of a non-set keyword, an unset keyword has no value (be nice if the good folks at RSI could follow up on this). Just to reiterate the tone of the original posting, I posted it so that people could hack it to suit thier needs. I'm happy to see it was of some use to someone.

Joseph M. Zawodny (KO4LW) NASA Langley Research Center
zawodny@arbd0.larc.nasa.gov MS-475, Hampton VA, 23681

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) NASA Langley Research Center
zawodny@arbd0.larc.nasa.gov MS-475, Hampton VA, 23681

```
;+
; NAME:
; EXPAND
; PURPOSE:
; Array magnification  (CONGRIDI 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
```

Subject: Re: CONTTW.PRO and EXPAND.PRO - some bug fixes, enhancements
Posted by thompson on Fri, 09 Oct 1992 12:37:00 GMT
View Forum Message <> Reply to Message

In article <BvuroD.60q@news.larc.nasa.gov>, zawodny@arbd0.larc.nasa.gov (Dr. Joseph M Zawodny) writes...
> I guess to each his own. I do admit that the autoscaling was crude at best.

> I included it, for the same reason PV-WAVE has a point and click interface,
> there are managers out there who still like to play with data that they know
> little about ;-) . I will take exception to the alleged "misuse" of
> keyword_set. The only time n_elements is preferable to keyword_set is when the
> keyword needs to allow zero to be a "good" value or when you need to know
> whether the keyword is a scalar or an array. IMHO the first exception is more
> of a bug. Zero should not be considered a sign of a non-set keyword, an unset
> keyword has no value (be nice if the good folks at RSI could follow up on
> this). Just to reiterate the tone of the original posting, I posted it so that
> people could hack it to suit thier needs. I'm happy to see it was of some use
> to someone.

It's not a bug. "Set" keywords take true or false values. In IDL (as in most languages), false values are encoded as zero. Joseph Zawodny is suggesting instituting a whole new data type to do the same thing for keywords.

When passing keywords from higher level routines to lower level routines, it is imperative that there be a value to pass from one routine to the next. Otherwise, how could one do the following

```
 PRO DUMMY1,KEY=KEY,...
  ...
 DUMMY2,KEY=KEY,...
  ...
 END

 PRO DUMMY2,KEY=KEY,...
  ...
 END
```

In other words, how could you take the value of KEY that was (or was not) passed to DUMMY1 and pass it on to DUMMY2?

Also, you can pass the true/false value of the keyword directly. You don't have to use the /KEY form. You can call

```
 DUMMY1,KEY=0,...
```

or

```
 DUMMY1,KEY=myvalue,...
```

which is a lot better than

```
 IF myvalue THEN DUMMY1,/KEY,... ELSE DUMMY1,...
```

The upshot is, use KEYWORD_SET only for keywords that take true/false values, and use N_ELEMENTS to test whether a non-T/F keyword was passed or not.

Bill Thompson

Subject: Re: CONTTW.PRO and EXPAND.PRO - some bug fixes, enhancements
Posted by zawodny on Fri, 09 Oct 1992 13:35:57 GMT
View Forum Message <> Reply to Message

Bill, good argument! I retract my request for a bug fix. I've been at this IDL thing awhile, am set in my ways, and can get defensive.

At one point, I had a problem with passing keys to other routines by

DUMMY1,KEY=KEY,...

when KEY was undefined in the calling routine. That must have been a short lived "feature" (I'm now leary of calling anything a bug). After that I just got into the habit of setting ALL keywords in a proceedure to their default value whenever they were not passed.

Joseph M. Zawodny (KO4LW) NASA Langley Research Center
zawodny@arbd0.larc.nasa.gov MS-475, Hampton VA, 23681