Subject: Re: !p.multi and tv
Posted by davidf on Thu, 14 May 1998 07:00:00 GMT
View Forum Message <> Reply to Message

Cathy (csaute3@alumni.umbc.edu) writes:

> I would like to use !P.MULTI and TV to position images like you
> would plots. However I'm having difficulty and all of images
> are in the corner of the window. Help!

Whoops. Yes, TV doesn't work like that. You might try
TVImage on my web page. You will then be able to position
an image in the window with the same Position keyword
syntax that you use for a line or contour plot. For example,

```
TVImage, image, Position=[0.1, 0.1, 0.45, 0.45]
TVImage, image, Position=[0.55, 0.55, 0.9, 0.9]
```

You can find TVImage at this URL:

ftp://ftp.dfanning.com/pub/dfanning/outgoing/idl_examples/tv image.pro

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: !p.multi and tv
Posted by csaute3 on Fri, 15 May 1998 07:00:00 GMT
View Forum Message <> Reply to Message

I would like to use !P.MULTI and TV to position images like you
would plots. However I'm having difficulty and all of images
are in the corner of the window. Help!

Cathy

Subject: Re: !p.multi and tv
Posted by csaute3 on Wed, 27 May 1998 07:00:00 GMT
View Forum Message <> Reply to Message

davidf@dfanning.com (David Fanning) wrote:
>
> Cathy (csaute3@alumni.umbc.edu) writes:
>
>> I would like to use !P.MULTI and TV to position images like you
>> would plots.
>
>> IDL> window, 0
>> IDL> !p.multi = [0,1,4,0,0]
>> IDL> plot, indgen(10)
>> IDL> plot, indgen(20)
>> IDL> plot, indgen(30)
>> IDL> plot, indgen(40)
>>
>> Is there something else that has this capability?
>>
>
> I can see that "IDL doesn't work like this" won't satisfy
> you. How about something like this:
>
> ** ****
> PRO MultiImages, multi
> IF N_Params() NE 1 THEN multi = [0, 2, 2]
> imageFile = Filepath(SubDir=['examples','data'], 'worldelv.dat')
> image = BytArr(360, 360)
> OpenR, lun, imageFile, /Get_LUN
> ReadU, lun, image
> Free_Lun, lun
> Window, XSize=500, YSize=400
> !P.Multi = multi
> FOR j=0, multi[1]*multi[2]-1 DO BEGIN
>   Plot, Findgen(11), Color=!P.Background ; A hack, unfortunately. :-(
>   x1 = !X.Region[0] + 0.05
>   x2 = !X.Region[1] - 0.05
>   y1 = !Y.Region[0] + 0.05
>   y2 = !Y.Region[1] - 0.05
>   TVImage, image, Position=[x1, y1, x2, y2]
> ENDFOR
> END
> ** ******
>
> If you want the images in, for example, a 3-by-2 arrangement,
> you can type this:
>
>   !P.Multi = [0, 3, 2]

>   IDL> MultiImages, !P.Multi

This works well. In addition to the image, I would like to add axes, etc.
so I did this loop:

```
FOR j=0, multi[1]*multi[2]-1 DO BEGIN
   ; establish the size of the plot window
   Plot, Findgen(11), Color=!P.Background
   x1 = !X.Region[0] + 0.05
   x2 = !X.Region[1] - 0.05
   y1 = !Y.Region[0] + 0.05
   y2 = !Y.Region[1] - 0.05
   TVImage, image, Position=[x1, y1, x2, y2]
   Plot, Findgen(11), position=[x1, y1, x2, y2], xticklen=-0.02, $
      yticklen=-0.02, xtitle='latitude', ytitle='longitude', /nodata, $
      /noerase
ENDFOR
```

However the xtitle is "cut off" for image(s) along the bottom of the window,
and ytitle is "cut off" for image(s) along the left of the window.
Is there a way to "make them visible"?

Also, if I left the position keyword out of the second plot command,
the plot appeared in the quadrant next to the tvimage (not overlay).
Why?

Thanks for your help,
Cathy

Subject: Re: !p.multi and tv
Posted by davidf on Wed, 27 May 1998 07:00:00 GMT
View Forum Message <> Reply to Message

Cathy (csaute3@alumni.umbc.edu) writes in response to some
code I hacked up for her:

> This works well. In addition to the image, I would like to add axes, etc.
> so I did this loop:
>
> FOR j=0, multi[1]*multi[2]-1 DO BEGIN
>   ; establish the size of the plot window

```
>     Plot, Findgen(11), Color=!P.Background
>     x1 = !X.Region[0] + 0.05
>     x2 = !X.Region[1] - 0.05
>     y1 = !Y.Region[0] + 0.05
>     y2 = !Y.Region[1] - 0.05
>     TVImage, image, Position=[x1, y1, x2, y2]
>     Plot, Findgen(11), position=[x1, y1, x2, y2], xticklen=-0.02, $
>       yticklen=-0.02, xtitle='latitude', ytitle='longitude', /nodata, $
>       /noerase
> ENDFOR
>
> However the xtitle is "cut off" for image(s) along the bottom of the window,
> and ytitle is "cut off" for image(s) along the left of the window.
> Is there a way to "make them visible"?
```

The problem is that by positioning the axes in the plot "window"
I didn't leave enough room for the titles. Hence, they got
clipped. I prefer to work with the plot POSITION, rather than the
plot REGION (which includes the space allocated for titles and other
plot annotation), because it makes it much easier to do things
like puts axes around images, etc.

This code could probably be fixed just by making the "margin"
around each plot larger. Say 0.1 rather than 0.05. Or, alternatively,
I could set the "outside" margins of the plot larger with the
!X.OMargin and !Y.OMargin keywords, which is what I choose to
do in the modified code below.

```
> Also, if I left the position keyword out of the second plot command,
> the plot appeared in the quadrant next to the tvimage (not overlay).
> Why?
```

The PLOT command is what "advances" the !P.MULTI system
variable to the next plot. My original code had a single
PLOT command that drew each plot in the background color
before each TVImage call. The purpose of this is to set
the system variables manipulated by !P.MULTI to the next
plot. I had to do this because TV commands don't "advance"
anything, just like they don't erase the display before
they draw into it.

Normally, the second PLOT command in your code would
advance the plot. Using the POSITION keyword obviously
causes !P.MULTI to think seriously about what it is doing.
I'm frankly surprised it made such a smart decision in
this case. I would have expected all kinds of bad things
to happen with the code above. But then again, that's why
I love IDL! :-)

Here is the modified code, which shows plot titles clearly
on my display. (Remember that outside margins are specified
in character units, so your mileage may vary. Adjust as
necessary.)

```
 ********************************************************** ****
PRO MultiImages, multi
IF N_Params() NE 1 THEN multi = [0, 2, 2]
imageFile = Filepath(SubDir=['examples','data'], 'worldelv.dat')
print, imageFile
image = BytArr(360, 360)
OpenR, lun, imageFile, /Get_LUN
ReadU, lun, image
Free_Lun, lun
Window, XSize=500, YSize=400
!P.Multi = multi
!X.OMargin = [2, 2]
!Y.OMargin = [2, 2]
FOR j=0, multi[1]*multi[2]-1 DO BEGIN
   Plot, Findgen(11), Color=!P.Background
   x1 = !X.Region[0] + 0.05
   x2 = !X.Region[1] - 0.05
   y1 = !Y.Region[0] + 0.05
   y2 = !Y.Region[1] - 0.05
   TVImage, image, Position=[x1, y1, x2, y2]
   Plot, Findgen(11), position=[x1, y1, x2, y2], xticklen=-0.02, $
     yticklen=-0.02, xtitle='latitude', ytitle='longitude', $
     /nodata, /noerase
ENDFOR
END
 ********************************************************** ***********
```

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: !p.multi and tv
Posted by Craig Markwardt on Thu, 28 May 1998 07:00:00 GMT

davidf@dfanning.com (David Fanning) writes:
> ... [discussion of how the PLOT command advances the !P.MULTI plot
> position] ...
> Here is the modified code, which shows plot titles clearly
> on my display. (Remember that outside margins are specified
> in character units, so your mileage may vary. Adjust as
> necessary.)
>
> ** ****
> PRO MultiImages, multi
> IF N_Params() NE 1 THEN multi = [0, 2, 2]
> imageFile = Filepath(SubDir=['examples','data'], 'worldelv.dat')
> print, imageFile
> image = BytArr(360, 360)
> OpenR, lun, imageFile, /Get_LUN
> ReadU, lun, image
> Free_Lun, lun
> Window, XSize=500, YSize=400
> !P.Multi = multi
> !X.OMargin = [2, 2]
> !Y.OMargin = [2, 2]
> FOR j=0, multi[1]*multi[2]-1 DO BEGIN
> Plot, Findgen(11), Color=!P.Background
> x1 = !X.Region[0] + 0.05
> x2 = !X.Region[1] - 0.05
> y1 = !Y.Region[0] + 0.05
> y2 = !Y.Region[1] - 0.05
> TVImage, image, Position=[x1, y1, x2, y2]
> Plot, Findgen(11), position=[x1, y1, x2, y2], xticklen=-0.02, $
> yticklen=-0.02, xtitle='latitude', ytitle='longitude', $
> /nodata, /noerase
> ENDFOR
> END

You can also use the AXIS command to paste axes on top of the image.

```
axis, xaxis=0, xtickformat='(A1)', xstyle=1
axis, xaxis=1, xtickformat='(A1)', xstyle=1
axis, yaxis=0, ytickformat='(A1)', ystyle=1
axis, yaxis=1, ytickformat='(A1)', ystyle=1
```

This particular set of commands does not print the tick labels because
that part has been handled already by a preceding PLOT command. Since
the plot has already been set up, the AXIS will automatically position
the axes correctly.

This is what I do in PLOTIMAGE, available from:
http://astrog.physics.wisc.edu/~craigm/idl/idl.html

Craig

--
---------- ----------
Craig B. Markwardt, Ph.D. EMAIL: craigmnet@astrog.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
---------- ----------

Subject: Re: !p.multi and tv
Posted by davidf on Thu, 28 May 1998 07:00:00 GMT
View Forum Message <> Reply to Message

Martin Schultz (mgs@io.harvard.edu) writes:

> David Fanning wrote:
>
> And this works ?? You amaze me, David! Have they changed this? Here is
> what the online help of IDL 5.02 says:
>
> OMARGIN
> A 2-element array specifying the "outer" margin on the left (bottom) and
> right (top) sides of a multi-plot window, in units of character size. A
> multi-plot window is created by setting the !P.MULTI system variable
> field. OMARGIN controls the amount of space around the entire plot area,
> including individual plot margins set with !X.MARGIN and !Y.MARGIN.
> [...]
> Setting !P.POSITION, or specification of the POSITION parameter
> overrides the effect of this field.

Ah, they don't really mean this. Well, maybe they mean it,
but I think they are talking about something else. :-)

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: !p.multi and tv
Posted by Martin Schultz on Thu, 28 May 1998 07:00:00 GMT
View Forum Message <> Reply to Message

David Fanning wrote:

[...]

example code:

```
< [...]
>  !P.Multi = multi
>  !X.OMargin = [2, 2]
>  !Y.OMargin = [2, 2]
>  FOR j=0, multi[1]*multi[2]-1 DO BEGIN
>     Plot, Findgen(11), Color=!P.Background
>     x1 = !X.Region[0] + 0.05
>     x2 = !X.Region[1] - 0.05
>     y1 = !Y.Region[0] + 0.05
>     y2 = !Y.Region[1] - 0.05
>     TVImage, image, Position=[x1, y1, x2, y2]
>     Plot, Findgen(11), position=[x1, y1, x2, y2], xticklen=-0.02, $
>       yticklen=-0.02, xtitle='latitude', ytitle='longitude', $
>       /nodata, /noerase
>  ENDFOR
>  END
>
```

And this works ?? You amaze me, David! Have they changed this? Here is what the online help of IDL 5.02 says:

OMARGIN
A 2-element array specifying the "outer" margin on the left (bottom) and right (top) sides of a multi-plot window, in units of character size. A multi-plot window is created by setting the !P.MULTI system variable field. OMARGIN controls the amount of space around the entire plot area, including individual plot margins set with !X.MARGIN and !Y.MARGIN.
[...]
Setting !P.POSITION, or specification of the POSITION parameter overrides the effect of this field.

^^^^^^^^^

Regards,
Martin.

---------- -------
Dr. Martin Schultz
Department for Earth&Planetary Sciences, Harvard University
109 Pierce Hall, 29 Oxford St., Cambridge, MA-02138, USA

phone: (617)-496-8318
fax : (617)-495-4551

e-mail: mgs@io.harvard.edu
Internet-homepage: http://www-as.harvard.edu/people/staff/mgs/
---------- -------

Subject: Re: !p.multi and tv
Posted by pit on Fri, 29 May 1998 07:00:00 GMT
View Forum Message <> Reply to Message

In article <6kkn02$h39$1@nnrp1.dejanews.com>,
csaute3@alumni.umbc.edu writes:
> In article <6jp3cljus3@gwdu19.gwdg.de>,
> pit@uni-sw.gwdg.de wrote:

>> PRO Tvimg, image, xax, yax, position=pos, box=box, noerase=noerase, $
>> nolabels=nolabels, noscale=noscale, ASPECT=aspect, _EXTRA=extra
> [...]
>
> Tvimg.pro uses a function called "rescale". This is not an IDL function.
> Is this a routine that you wrote?

Yes - I overlooked it while scanning for unresolved routines - sorry:

Peter

```
------------------------
FUNCTION Rescale, image, xsize, ysize, scale=scale
;+
; NAME:
;       RESCALE
; PURPOSE:
;       Expand or shrink a given (1- or 2-dimensional) Array to new size
; CATEGORY:
;
```

```
; CALLING SEQUENCE:
;       RESULT = RESCALE( IMAGE, [[ XSIZE [, YSIZE]] || SCALE=Scale] )
; INPUTS:
;       IMAGE : Original Image
;       XSIZE : New dimension for the X-axis
;       YSIZE : 1-d case: If ysize is given, a pseudo-2d Array of
;               identical rows is returned, else a 1-dim Array of size
;               xsize .
;               2-d case: If ysize is given it is taken as the new
;               dimension for the Y-axis. If omitted, the Y-axis is
;               scaled with the same factor as the X-axis to keep the
;               aspect ratio
; KEYWORDS:
;       SCALE : (input) 1- or 2-element vector with the scales for x
;               and y axis. Overrides YSIZE and YSIZE.
; OUTPUTS:
;       RESULT: Rescaled array of same type as IMAGE, size is
;               (XSIZE,YSIZE)               (1/2-d case, YSIZE given)
;               (XSIZE,YOrig*XOrig/XSIZE)     (2-d case, YSIZE omitted)
;               (XSIZE)                     (1-d case, YSIZE omitted)
; PROCEDURE:
;       Scale-Factors are computed from the Size of the original data
;       and the parameters XSIZE and YSIZE. For 1-dim case SPLINE is
;       used for interpolation, in 2-d case POLY_2D is used.
; MODIFICATION HISTORY:
;       19-Feb-1993  P.Suetterlin, KIS
;-

on_error, 2                             ; Return on error

IF n_params() LT 2 AND NOT keyword_set(scale) THEN BEGIN
   message, 'Usage:  result = rescale( Image, XSize [, YSize] )'
ENDIF

s=size(image)
IF s(s(0)+1) EQ 8 THEN BEGIN    ; Image is ST4-Structure?
   pic = image.pic              ; Only for my personal
                        ; use
   s = size(pic)
ENDIF ELSE pic = image

IF keyword_set(scale) THEN BEGIN
   IF n_elements(scale) EQ 1 THEN scale = [scale, scale]
     ;;; scale overrides xsize/ysize
   xsize = s(1)*scale(0)
   IF s(0) NE 1 THEN ysize = s(2)*scale(1)
ENDIF
```

```
IF s(0) EQ 1 THEN BEGIN                    ; Image is 1-d Array
   line = spline(indgen(s(1)), pic, $
              findgen(xsize)/xsize*s(1))
   IF n_params() EQ 3 THEN BEGIN           ; Expand to 2-dim
      scalepic = fltarr(xsize, ysize)
      FOR i = 0, ysize-1 DO scalepic(*, i) = line
      return, scalepic                     ; Return 2-dim
   ENDIF ELSE return, line                 ; Return 1-dim
ENDIF ELSE BEGIN
   IF n_params() LT 3 AND NOT keyword_set(scale) THEN  $  ; Rescale Y by same
     ysize = fix(s(2)*(float(xsize)/s(0))+.5)           ; amount
   P = [[0, 0], [float(s(1))/xsize, 0]]     ; Matrices for
   Q = [[0, float(s(2))/ysize], [0, 0]]     ; Poly_2d
   return, poly_2d(pic, P, Q, 1, xsize, ysize)
ENDELSE
END
-------------------------
```

--
~~~~~~

Peter "Pit" Suetterlin http://www.uni-sw.gwdg.de/~pit
Universitaets-Sternwarte Goettingen
Tel.: +49 551 39-5048 pit@uni-sw.gwdg.de

-- * -- * ...-- * -- * ...-- * -- * ...-- * -- * ...-- * -- * ...-- * --

Come and see the stars! http://www.kis.uni-freiburg.de/~ps/SFB
Sternfreunde Breisgau e.V. Tel.: +49 7641 3492
~~~~~~