Subject: Re: multiple images with postscript in wave ?
Posted by chase on Wed, 07 Apr 1993 17:10:13 GMT
View Forum Message <> Reply to Message

In article <1psjfg$2qb@mailgzrz.TU-Berlin.DE> dieh2133@w251zrz.zrz.tu-berlin.de (Rolf Diehl) writes:

> Can somebody tell me how to get multiple images with postscript on
> one page with wave?
> According to T12 of the idl-faq I tried the following:
>
> ; WAVE Version 3.10 (hp-ux hp9000s300)
> ximsize=0.333*!D.x_size
> yimsize=0.5*!D.y_size
> set_plot,'ps'
> device,/color,bits=8
> tvscl,rebin(float(rry),160,208),0, xsize=ximsize, ysize=yimsize
> tvscl, ... for the positions 1 to
>
> on screen I get 6 images, but previewing the postscriptfile with
> ghostscript gives 6 images one over the other in the same position.
> Is there something wrong with the code or is it just not possible to do it?
> Thanx for any help
> Rolf.

Hmmm, I believe your mistake was to get the plot window sizes for the device BEFORE setting the device. Try getting it after set_plot,'ps'.
I tried essentially the same thing as the above and it worked for me. Here is what I tried:

```
im = SIN(DIST(50,50)*(4*!pi/25.))
set_plot,'ps'
x=!D.x_size*.333
y=!d.y_size*.5
for i=0,5 do tvscl,rebin(im,150,150),i,xsize=x,ysize=y
device,/close

set_plot,'x'
for i=0,5 do tvscl,rebin(im,150,150),i,xsize=x,ysize=y
```

When the device is 'X' the size keywords are ignored because device 'X' does not have scalable pixels. One problem with this is that you may get a different number of images across the page with the postscript device than with 'X'. (I got 4 images across for 'X' and 3 across for 'PS')

I found that when displaying images I never got the same postscript output that I was getting on the screen. This is because the postscript device does not define a fixed image pixel size. In order to make my routines that display images independent of the current device, I wrote two routines which I am including below. I use MTV.pro in place of TV and TVSCL for displaying images. It uses a default dots (image pixels) per inch if the current device has scalable pixels and sizes the image appropriately. The other routine is SCL.pro which scales position and sizes given in image pixel units into device pixel units. This is useful when using routines like XYOUTS with MTV (see the examples.)

To get help on these routines enter:

mtv,/help
a=scl(/help)

Copy and paste the examples found in the help documentation into IDL to try them out.

The advantage of using these routines is that I get the same output whether the current device is 'X' or 'PS'.

---------- Begin Included Files: mtv.pro and scl.pro ----------

```
PRO Mtv, image, x, y, device=device, low=low, high=high, $
        Inches=inches, Order=order, $
        Res=res, Help=help
;+
; NAME:
;
;   MTV
;
; PURPOSE:
;
;   TV and TVSCL wrapper that is independent of the screen and
;   postscript devices. Use just as you would TV or TVSCL.  It
;   automatically scales if the data is not byte data. Also allows the
;   specification of an absolute scale.
;
; CATEGORY:
;
;   Image, Printer
;
; CALLING SEQUENCE:
;
;   MTV,Image,X,Y
;   MTV,Image,Position
```

```
;
; INPUTS:
;
;   Image - 2D array to be displayed. MTV only works with 2D arrays
;           (not with vectors).
;
;
; OPTIONAL INPUTS:
;
;   X,Y - Position of image lower left corner in image pixels, unless
;         the inches keyword is set, in which case they are in inches.
;         Defaults to (0,0).
;
;   Position - Same as for the TV routine. (The position of the image
;              on the screen as an integer. The screen is divided into
;              an integral number of segments of size equal to Image.)
;
; KEYWORD PARAMETERS:
;
;       Keywords equivalent to the TV keywords (see the description
;       for the TV routine):
;
;   DEVICE - all positions and sizes given in pixel coordinates. (default)
;
;   INCHES - all positions and sizes given in inches.
;
;   ORDER
;
;      New keywords:
;
;   RES - The resolution for a device with scalable pixels (e.g.
;         postscript printer). Default = 75dpi (dots per inch).  When
;         the device does not have scalable pixels, the resolution is
;         fixed and this parameter is ignored.
;
;   LOW - The minimum value for the image scale. Values less than LOW
;         will be set to LOW. Defaults to Image minimum. The LOW value
;         is mapped to the 0 value of the color table.
;
;   HIGH - The maximum value for the image scale. Values greater than
;          HIGH will be set to HIGH. Defaults to Image maximum. The
;          HIGH value is mapped to the maximum index of the color
;          table, which is !D.N_COLORS - 1 (!D.N_COLORS is the number
;          of colors availabe).
;
;   HELP - Provide help (this information). No other action is performed.
;
; OUTPUTS:
```

```
;
;   None
;
; RESTRICTIONS:
;
;   Only works with 2D arrays.
;
;   Only tested with 8bit devices. Does not support TRUE color or
;   CHANNEL options of TV routine. Does NOT support DATA, T3D, WORD,
;   NORMAL or Z keywords of TV routine.
;
;   It is recommended that you use the TV or TVSCL routines directly
;   if you want to scale an image to fit a display (e.g. plot or
;   contour). On the other hand this routine would be appropriate for
;   scaling a plot or contour display to overlay an image.
;
; PROCEDURE:
;
;   Scales the data according to the number of colors and the LOW,
;   HIGH keywords using the byscl procedure.
;
; EXAMPLE:
;
;   image=SIN(DIST(200,200)*(4*!Pi/200.))
;   MTV,image,50,50,RES=100,HIGH=2.,LOW=-2.
;
;   ; If the device does not have scalable pixels (e.g. 'X') then the
;   ; RES keyword has no effect. If the current device has scalable
;   ; pixels (e.g. 'PS') then image pixels will map to 100 dots per
;   ; inch. If my screen resolution is 100dpi my images have the
;   ; same displayed size independent of the current device.
;   ; Typically, one would just use the default value for RES.
;
;
; MODIFICATION HISTORY:
;
;       Mon Mar 22 13:36:05 1993, Chris Chase <chase@johnston>
;             Added low,high keywords. Scaled the data according to
;             the number of colors and the low, high keywords. No
;             longer uses tvscl. Instead uses bytscl and tv.
;
;             Added documentation header.
;
;       Wed Mar  3 17:34:16 1993, Chris Chase <chase@johnston>
;             Initial Version. Added proper header.
;
;-
IF KEYWORD_SET(help) THEN BEGIN
```

```
   doc_library, 'mtv'
   RETURN
ENDIF

IF NOT KEYWORD_SET(low) THEN low = min(image)
IF NOT KEYWORD_SET(high) THEN high = max(image)

; Get rid of trivial leading dimensions (i.e. dimension size = 1)
nimage = reform(image)

s = size(nimage)
IF s(0) NE 2 THEN BEGIN
   print, 'MTV ERROR: The image is not a 2D array'
   RETURN
ENDIF

bimage = bytscl(nimage, min = low, max = high, top = !D.n_colors)

cmd_str = 'tv,bimage'

IF N_ELEMENTS(x) EQ 1 THEN BEGIN
   cmd_str = cmd_str + ',xd'
   xd = x
ENDIF ELSE xd = 0
IF N_ELEMENTS(y) EQ 1 THEN BEGIN
   cmd_str = cmd_str + ',yd'
   yd = y
ENDIF ELSE yd = 0

;
; Use a default pixel size of 75 dots per inch for scalable device pixels.
; Otherwise, for fixed size device pixels use the actual size
;
IF NOT KEYWORD_SET(res) THEN res = 75.0 $
  ELSE res = float(res)

;; check for scalable pixels.
;; scl has units of device pixels per image pixel (dots).
IF (!D.flags AND 1) THEN BEGIN
   scl = 2.54*!D.x_px_cm/res    ; 2.54 cm/inch
ENDIF ELSE BEGIN
   ;; Fixed size device pixels
   scl = 1.0
   res = 2.54*!D.x_px_cm
ENDELSE

;; Convert to device pixel units when x,y offset given
IF (n_params() EQ 3) THEN BEGIN
```

```
   if KEYWORD_SET(inches) THEN BEGIN
      ;; convert from inches into image pixels
      xd = xd*res
      yd = yd*res
   ENDIF
   xd = xd*scl
   yd = yd*scl
ENDIF

IF KEYWORD_SET(order) THEN cmd_str = cmd_str + ',/order'

;; Using the size keywords only affects the devices with scalable
;; device pixels
xsize = s(1)*scl
ysize = s(2)*scl
cmd_str = cmd_str + ',xsize=xsize,ysize=ysize'
ans = execute(cmd_str+',/device')
RETURN
END

FUNCTION Scl, x, res=res, help=help
;+
; NAME:
;
;   SCL
;
; PURPOSE:
;
;   Convert image pixel units into device pixel coordinates so that
;   screen and printer pixels have the save size. This makes it easier
;   for image display that is independent of whether device is the
;   screen or a device with scalable pixels (e.g., a postscript
;   printer). Use this routine with XYOUTS or when scaling a plot and
;   you want to use pixel units for positions and sizes.
;
; CATEGORY:
;
;   Image Display
;
; CALLING SEQUENCE:
;
;   Result = SCL(X)
;
; INPUTS:
;
;   X - Array of values in pixel units.
;
; KEYWORD PARAMETERS:
```

```
;
;   RES - resolution in dots per inch - this only affects printing to
;         postscript. A dot is the same thing as an image pixel.
;         Defaults to 75.0 for a postscript device; otherwise, it is
;         fixed for the screen (around 75-100 dpi).
;
;   HELP - Provide help (this information). No other action is performed.
;
; OUTPUTS:
;
;   Result - The X data scaled from pixel units into device units
;            where the size of a pixel is given by RES. For fixed size
;            pixel devices (typically screen devices) Result = X.
;
; RESTRICTIONS:
;
;  Assumes the device pixel size is square. Thus, displays using SCL
;  may give unexpected results if the device y pixel size (!D.y_px_cm)
;  is not the same as the device x pixel (!D.x_px_cm).
;
; PROCEDURE:
;
;   Determines if the current device has scalable pixels. If so, gets
;   the size of the pixels (assumed square) and scales the data
;   according to the resolution (RES).
;
; EXAMPLE:
;
;    ; In this example I use SCL everywhere that I need a size or
;    ; position in terms pixels.
;   image = SIN(DIST(200,200)*(4*!pi/200.))
;
;    ; Display the image offset by 50 pixels
;     TVSCL,image,SCL(50),SCL(50),XSIZE=SCL(200),YSIZE=SCL(200),/D EVICE
;
;    ; Position a centered label, 5 pixels above the image
;   XYOUTS,SCL(150),SCL(255),'This is a label',ALIGN=.5,/DEVICE
;
;    ; Or using the MTV routine to do the same thing alongside the first.
;    ; MTV does not require scaling the image position.
;   MTV,image,250,50
;
;    ; Position a centered label, 5 pixels above the image
;   XYOUTS,SCL(350),SCL(255),'Another label',align=.5,/DEVICE
;
;    ; Overlay first image with a contour.
;     CONTOUR,image,POSITION=SCL(50+[0,0,200,200]),XSTYLE=1,YSTYLE =1, $
;           /NOERASE,/DEVICE
```

```
;
;   ; This will produce the same results whether the device is for the
;   ; screen or a postscript printer.
;
; MODIFICATION HISTORY:
;
;       Mon Mar 22 15:50:47 1993, Chris Chase <chase@johnston>
;             Created.
;
;-

IF KEYWORD_SET(help) THEN BEGIN
   doc_library, 'scl
   RETURN,0
ENDIF

;
; Use a default pixel size of 75 dots per inch for scalable device pixels.
; Otherwise, for fixed size device pixels use the actual size
;
IF NOT KEYWORD_SET(res) THEN res = 75.0 $
  ELSE res = float(res)

;; check for scalable pixels.
;; scl has units of device pixels per image pixel (dots).
IF (!D.flags AND 1) THEN BEGIN
   scl = 2.54*!D.x_px_cm/res    ; 2.54 cm/inch
   RETURN,(x*scl)
ENDIF
RETURN, x
END
```

--

==========

Bldg 24-E188
The Applied Physics Laboratory
The Johns Hopkins University
(301)953-6000 x8529