
Subject: Re: Giant crosshairs

Posted by [thompson](#) on Fri, 28 Aug 1992 14:30:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

In article <laa-270892173708@berserk.c3.lanl.gov>, laa@lanl.gov (Lee Ankeny) writes...

> Hello Wave gurus,

>

> I'd like to implement a "giant crosshair", like the old tek terminals used
> to have. It would extend from top edge to bottom edge, and left edge to
> right edge of the window, intersecting at the current mouse pointer
> location.

>

> It's not obvious to me how to do this, so I'd appreciate any tips you might
> have. Thanks in advance.

The SunView driver in IDL has more-or-less this kind of crosshair (at least it's huge), but the X-windows driver does not. Maybe this is a limitation of X-windows servers? You can define your own cursor, but only within a 16x16 pixel box.

A while back there was a discussion about how to overplot lines that could be erased without disturbing the underlying plot. The following program (from an earlier post by Dr. Joseph M Zawodny) utilized this feature to draw a cursor box on the screen. It probably could be modified to produce the kind of crosshairs that you're talking about. (Please note though that this particular routine is now superceded by the built-in BOX_CURSOR command in IDL).

Another advantage to the Tektronix crosshair was that, with the arrow keys, one could control the left-right and up-down motion independantly. I once had an application which required measuring something at one position, and then measuring something else at a different Y position, but at the exact same X position. I wrote a routine which allowed me to move the cursor with the keyboard, which I could post if anyone's interested.

Bill Thompson

Path: nsisrv!ames!olivea!uunet!news.larc.nasa.gov!arbd0.larc.nasa.gov!zawodny
From: zawodny@arbd0.larc.nasa.gov (Dr. Joseph M Zawodny)
Newsgroups: comp.lang.idl-pvwave
Subject: Re: rubber band lines
Message-ID: <1992Jul13.112354.4945@news.larc.nasa.gov>
Date: 13 Jul 92 11:23:54 GMT
References: <1992Jul10.174058.29762@ll.mit.edu>
Sender: news@news.larc.nasa.gov (USENET Network News)
Organization: NASA Langley Research Center, Hampton, VA
Lines: 109

Here is a routine that will draw a "rubber band" box and will leave the underlying plot undisturbed.

```
pro BOX,xv,yv,ratio=ratio,device=devi,data=data

; check keyword compatability
if(keyword_set(device) and keyword_set(data)) then begin
  print,' BOX cannot be called with both /DEVICE and /DATA'
  return
endif

; keep it quiet
quiet=!quiet
!quiet=1
; save ther current graphics mode
device,get_graph=oldg,set_graph=6

; wait for the initial click
cursor,x0,y0,/down,data=data,device=devi
; Initialize
xo=x0
yo=y0
; Have to plot the first point as a dot
plots,[x0,x0,xo,xo,x0],[y0,yo,yo,y0,y0],data=data,device=devi

; sit here and watch for the cursor button to be released
again: cursor,xdummy,ydummy,/nowait,data=data,device=devi
if(!err ne 0) then goto,again

loop:
; make sure the window gets updated (wait forces a flush)
wait,.001

; monitor the cursor for movements or the second click
cursor,cx,cy,/change,data=data,device=devi
; if it was the second click we're done
if(!err eq 1) then goto,done
; otherwise update the coordinates
dx = cx-x0
dy = cy-y0

; check aspect ratio ?
if keyword_set(ratio) then begin
; be careful of division by zero
if(dx*dy eq 0) then begin
  dx = 0
```

```

dy = 0
endif else begin
; two possible sides
ay = abs(1.*dx/ratio)
ax = abs(1.*dy*ratio)
; select largest rectangle
if(ax gt abs(dx)) then dx=ax*dx/abs(dx) $
  else dy=ay*dy/abs(dy)
endelse

endif

; values of the opposite vertex
x1 = x0+dx
y1 = y0+dy

; erase the old box
plots,[x0,x0,x0,x0,x0],[y0,y0,y0,y0,y0],data=data,device=device
; draw the new box
plots,[x0,x0,x1,x1,x0],[y0,y1,y1,y0,y0],data=data,device=device
; save the new coordinates
x0=x1
y0=y1
; keep going back until we get the second click
goto,loop

done:
; restore graphics mode
device,set_graph=oldg
; create output arrays
xv=[x0,x1]
yv=[y0,y1]
; special processing?
if (not keyword_set(device)) then begin
; for output in data coordinates order according to !n.CRANGE
if(((!x.crangle(1)-!x.crangle(0)) * dx) lt 0) then xv=reverse(xv)
if(((!y.crangle(1)-!y.crangle(0)) * dy) lt 0) then yv=reverse(yv)
endif else begin
; device coordinates are always non-real
xv=long(xv)
yv=long(yv)
; for device coordinates use ascending order
if(x0 gt x1) then xv=reverse(xv)
if(y1 gt y0) then yv=reverse(yv)
endelse
;clean up and return
!quiet=quiet
return

```

end

You should be able to adapt this to your needs. As for the second question, you can preposition the cursor with the TVCRS command. Get a manual or use the ? function in IDL to get more info on this. Both of these should work under widgets.

Best of Luck

Joseph M. Zawodny (KO4LW) ^
NASA Langley Research Center V This space for rent.
Hampton VA, 23665-5225 ^
zawodny@arbd0.larc.nasa.gov V
