Subject: Re: rubber band lines
Posted by zawodny on Mon, 13 Jul 1992 11:23:54 GMT
View Forum Message <> Reply to Message

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=dev i

   ; 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,xo,xo,x0],[y0,yo,yo,y0,y0],data=data,device=dev i
  ; draw the new box
 plots,[x0,x0,x1,x1,x0],[y0,y1,y1,y0,y0],data=data,device=dev i
  ; save the new coordinates
xo=x1
yo=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.crange(1)-!x.crange(0)) * dx) lt 0) then xv=reverse(xv)
 if(((!y.crange(1)-!y.crange(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 \/ This space for rent.
Hampton VA, 23665-5225 /\
zawodny@arbd0.larc.nasa.gov \/
