
Subject: Re: Rubberband box for object graphics
Posted by [Chris Lee](#) on Sat, 12 Feb 2005 12:43:02 GMT
[View Forum Message](#) <> [Reply to Message](#)

In article <MPG.1c766aa9e08148df989908@news.frii.com>, "David Fanning"
<davidf@dfanning.com> wrote:

```
> "Rainnie, JA \(\John\)" <J.A.Rainnie@rl.ac.uk> writes:
>> I'm delving into the IDL graphics objects, and specifically I wish to
>> develop a plot window using the IDLgrPlot class. Thanks to several
>> examples (mostly Dave Fanning's xPlot) I've managed to do this.
>> However, I really need to implement a rubberband box so the user can
>> interactively zoom into this plot (like DF's zPlot). The conversion
>> from device to data coordinates is of course done in direct graphics
>> with the Convert_Coord function. However, I can't find any examples of
>> how this is done for object graphics.
>> Has anyone done this - and are there any examples you can direct me to.
```

Hi John,

The simplest example I can make, shoehorned into David's xPlot is to add
the /button_events and /motion_events to the draw widget vis.

```
drawID = Widget_Draw(tlb, XSize=400, YSize=400, Color_Model=0, $
  Graphics_Level=2, Expose_Events=1, Retain=0, /button_events,/motion_events,$
  Event_Pro='XPlot_Draw_Widget_Events')
```

then add a few variables to the INFO struct to handle the rubberband box..

```
....  i_rubber_band:[0,0,0L,0L], $
      tr_rubber_band:[0,0,0.0,0.0], $
      rb_down:0L }
```

Then modify the xplot_draw_widget_events to

```
PRO XPlot_Draw_Widget_Events, event
  ; This event handler handles draw widget expose events.
Widget_Control, event.top, Get_UValue=info, /No_Copy
  ; Draw the graphic.
```

```
if(event.type eq 4) then $
  info.thisWindow->Draw, info.plotView
```

```
if(event.type eq 0) then begin
  info.i_rubber_band[0]=event.x
```

```

    info.i_rubber_band[1]=event.y
info.rb_down=1
endif else if (event.type eq 1) then begin
    info.i_rubber_band[2]=event.x
    info.i_rubber_band[3]=event.y
    info.rb_down=0

info.thisPlot->getproperty, xcoord_conv=xs,ycoord_conv=ys
info.thisWindow->getproperty,dimensions=sd

info.tr_rubber_band[[0,2]]=info.i_rubber_band[[0,2]]*1.0/sd[ 0]
info.tr_rubber_band[[1,3]]=info.i_rubber_band[[1,3]]*1.0/sd[ 1]

info.tr_rubber_band[[0,2]]=(info.tr_rubber_band[[0,2]]-xs[0])/xs[1]
info.tr_rubber_band[[1,3]]=(info.tr_rubber_band[[1,3]]-ys[0])/ys[1]

;plot the data
info.thisPlot->getproperty, data=d
xr=info.tr_rubber_band[[0,2]] & xr=xr[sort(xr)]
yr=info.tr_rubber_band[[1,3]] & yr=yr[sort(yr)]
plot, d[0,*], d[1,*], xrange=xr, yrange=yr

endif

if(event.type eq 2 and rb_down eq 1) then begin
;draw a rubberband box with polylines here.
endif

;Put the info structure back.
Widget_Control, event.top, Set_UValue=info, /No_Copy
END

```

```

.....
,,,,,,,,,

```

This will draw a direct graphics plot for you of the zoomed region. The basic method is to use the button and motion events to get the DEVICE coordinates. Convert these to normalised coordinates using the window dimensions, then convert the NORMAL coordinates to DATA coordinates using the plot object ycoord_conv and xcoord_conv (backwards).

There are obviously many things you need to be wary of, this example only works on the trivial case of one plot-one line. If you have a number of lines in your plot, then you need to set those up as well. This is probably easier in object graphics because you just put the IDLgrPlot objects into a new View/Window and your done. The same would be true of you have multiple views in a window, and models..

Chris.
