Subject: [Q]:Rubberbanding in IDL

Posted by on Mon, 20 Mar 1995 09:43:31 GMT

View Forum Message <> Reply to Message

Hi there!

Has anyone got a good rubberbanding routine for IDL available?

You know what I mean, don't you?
You press and hold the left mouse button in a graphics window and you get a scalable rectangle where one corner is fixed at the location you pressed, and the other (diagonal) corner follows the mouse until you release the button.
(And then the rectangle is either fixed or removed - without a trace in the image.)

I want it for selecting "area of interest" in an image.

TIA,

Subject: Re: [Q]:Rubberbanding in IDL Posted by mbastian on Fri, 31 Mar 1995 08:00:00 GMT View Forum Message <> Reply to Message

Here is my rubber banding program.

Hope this helps.

Matthew Bastian
National Research Council of Canada

pro RubberBand, LeftHandSide, RightHandSide, Bottom, Top;
; this program will draw a rubber band box that follows the cursor; the first corner starts when the left mouse button is pressed; and the other corner ends when the mouse button is released;
!err = 0
LeftButtonPressed = 1
LoopCount = 0L
;
; set the clip rectangle to it's maximum limits in order for "zoom" to work.

```
!p.clip = [0,0,!d.x\_size,!d.y\_size,0,0]
 keep looping until the user presses the left mouse button
while !err ne LeftButtonPressed do begin
   every 100 times through this loop print the usage message
  if LoopCount mod 100L eq 0 then begin
     print, 'Press left mouse button to activate rubber band box'
     LoopCount = 0
  endif
  LoopCount = LoopCount + 1
   get the current mouse location in pixels relative to lower lhs corner
  ; note that this function return immediatly without waiting for a mouse
  ; button to be pressed
  cursor, x1, y1, /nowait, /data
endwhile
print, 'Release left mouse button to de-activate rubber band box'
 keep looping and drawing the rubber band box until the button is released
!linetype = 0
x^2 = x^1
y2 = y1
x = -1
y = -1
device, set_graphics=6, get_graphics=oldg
oplot,[x1, x1, x2, x2, x1], [y1, y2, y2, y1, y1]; /device
while (!err eq LeftButtonPressed) or ((x eq -1) and (y eq -1)) do begin
  cursor, x, y, /nowait, /data
  if (x ne x2) or (y ne y2) then begin
     oplot,[x1, x1, x2, x2, x1], [y1, y2, y2, y1, y1]
     x2 = x
     y2 = y
     oplot,[x1, x1, x2, x2, x1], [y1, y2, y2, y1, y1]
  endif
endwhile
oplot,[x1, x1, x2, x2, x1], [y1, y2, y2, y1, y1]
device, set_graphics=oldg
; set up the return values
```

```
; converting from screen coordinates to world coordinates
if x1 lt x2 then begin
  LeftHandSide = float (x1)
  RightHandSide = float (x2)
endif else begin
  RightHandSide = float (x1)
  LeftHandSide = float (x2)
endelse
if y1 It y2 then begin
  Bottom = float (y1)
  Top = float (y2)
endif else begin
  Top = float (y1)
  Bottom = float (v2)
endelse
return
end
```

Subject: Re: [Q]:Rubberbanding in IDL

View Forum Message <> Reply to Message

Posted by rouse on Sat, 01 Apr 1995 08:00:00 GMT

In article <D6B64n.CJ0@hpl.hp.com>, peter@hpl.hp.com (Peter Webb) writes: |>Newsgroups: comp.lang.idl-pvwave |>Path: lanews.la.asu.edu!news.asu.edu!asuvax!ncar!newshost.lanl.gov !ferrari.mst6.lanl.gov!nntp-server.caltech.edu!netline-fddi. jpl.nasa.gov!news.byu.edu!news.mtholyoke.edu!uhog.mit.edu!ne ws.mathworks.com!news.duke.edu!agate!howland.reston.ans.net! swrinde!sdd.hp.com!hplabs!hplntx!peter |>From: peter@hpl.hp.com (Peter Webb) |>Subject: Re: [Q]:Rubberbanding in IDL |>Sender: news@hpl.hp.com (HPLabs Usenet Login) |>Message-ID: <D6B64n.CJ0@hpl.hp.com> |>Date: Fri, 31 Mar 95 07:12:23 MST |>References: <3kiio3\$fin@nms.telepost.no> <3kikos\$gmk@nms.telepost.no> <3ldd65INNksq@lanews.la.asu.edu> |>Nntp-Posting-Host: oz.hpl.hp.com |>Organization: Hewlett-Packard Laboratories, Palo Alto, CA |>X-Newsreader: TIN [version 1.2 PL2] I>Lines: 22 |>

>I'd post my favorite routine, but it's copywrite of my previous employer...

|>

```
|>A couple of tricks I found useful:
|>
> Use device, /copy to write the image to a hidden pixmap, then you can
> clean up the box during resizes by copying it back. This is faster
> than rewriting the section you just drew a box on, since the x server
> does it directly. You can drag a box around a 1024^2 image as fast as
> you please. You also don't have to worry about off-by-one errors when
> cleaning up, which tend to leave little goobers all over the image
> (yes, I know, just don't make mistakes in the first place...).
Isn't the same thing accomplished by setting graphics to 6:
 device, get_graphics=old, set_graphics=6
|>
> Using normalized co-ordinates, rather than device co-ordinates, cleans
> up the code a bunch, especially if you need to calculate back to some
|> physical co-ordinates anyway. The advantage is that the code is
> independent of image screen size.
|>
>Email me if you want more details.
|>
l>Peter
|>
|>
- Roger
Roger B. Rouse
                     #
Arizona State University
|Dept. Physics & Astronomy
Tempe, Az, 85287-1504
                         . @ .
lrouse@sevens.la.asu.edu
The AscII Galaxy found
in the Local Network."
                               + .
                                    Rouse|
                         .#+#.
```