
Subject: Moving an ROI

Posted by [Andre Kyme](#) on Tue, 04 Dec 2001 21:35:50 GMT

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

Hi everybody,

Anyone know of a routine to move an ROI around in a draw widget? I've collected the ROI coordinates and now want to adjust its position. Wrote a bit of code that's kind of working but is very dodgy when the ROI gets to edges. Thanks for the help,

Andre Kyme

Dept Medical Physics
Westmead Hospital

Subject: Re: Moving an ROI

Posted by [Pavel A. Romashkin](#) on Wed, 05 Dec 2001 16:55:42 GMT

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

Nobody replied. I guess a routine is not there. But it seems simple to write one. Once you have a ROI, all it is is just a polygon. With motion events on, just add delta X and Y to its Data property and redraw the view. Edges need a special thought, seems to me. First, have you decided how do *you* want them treated? If you trim the ROI, you need to redefine Data because IDLgrPolygon has no *Range properties. I did this by keeping the original ROI data in Uvalue, then trimming Data as Data = [[X_min < Data[0,*] < X_max], [Y_min < Data[1,*] < Y_max], [Data[2,*]]], then redrawing.

If you want to expand the plot range as ROI gets dragged past the edge, you need to rescale the plot and change axes limits. Although I anticipate somewhat jerky screen updates with large plots, it also seems simple. For data processing, just reuse the data from Uvalue (the original ROI).

Pavel

Andre Kyme wrote:

>

> Hi everybody,

>

> Anyone know of a routine to move an ROI around in a draw widget? I've
> collected the ROI coordinates and now want to adjust its position. Wrote
> a bit of code that's kind of working but is very dodgy when the ROI gets
> to edges. Thanks for the help,

>

> Andre Kyme

>

> Dept Medical Physics
> Westmead Hospital

Subject: Re: Moving an ROI
Posted by [Andre Kyme](#) on Wed, 05 Dec 2001 21:55:48 GMT
[View Forum Message](#) <> [Reply to Message](#)

Thanks Pavel,

I had rewritten the move ROI program yesterday and it seems to be more robust. I use the `set_graphics_function` argument (to the device), and set this equal to 6 (XOR mode) so that the ROI "floats" over the image. When a new position is detected, the old position is replotted to make it disappear and the new position plotted. This is all OK.

Now about the edges:

As you say, by defining a new ROI equal to $(Xverts > 0) \text{AND} (Xverts < xMax)$, and similarly for Y, we create a new ROI that fits in the draw widget and has its edge along the draw widget boundary if there is any part of the ROI outside the allowed region.

Another way would be to form a bit mask for the Xverts, setting equal to 1 wherever the vertices are in the allowed region, and doing the same for the Yverts. If we then AND these two masks and finally do a WHERE on the result, we can plot `Xverts[result]`, `Yverts[result]`. The problem I see with this is that if we are over the boundary, the ROI stops at the point(s) of contact. I eventually want the total counts inside the ROI, which means that I need to CLOSE it off first, before doing a polyfillv. If I used this last method as is, the ROI would be truncated by simply drawing a line from one point of contact to the other - what I really want is the boundary to follow the edge of the draw widget where it has crossed the draw boundary. Closing off the region isn't so simple as far as I can see. I think you probably know what I'm talking about even if this explanation is incomprehensible!

Maybe I'll go for the first method.

Thanks for your reply Pavel,

Andre

Subject: Re: Moving an ROI
Posted by [Pavel A. Romashkin](#) on Thu, 06 Dec 2001 16:32:16 GMT
[View Forum Message](#) <> [Reply to Message](#)

Andre Kyme wrote:

>

> The problem I see with this is that if we are over
> the boundary, the ROI stops at the point(s) of contact. I eventually want the total

- > counts inside the ROI, which means that I need to CLOSE it off first, before doing
- > a polyfillv. If I used this last method as is, the ROI would be truncated by simply
- > drawing a line from one point of contact to the other - what I really want is the
- > boundary to follow the edge of the draw widget where it has crossed the draw
- > boundary.

I guess I am not quite sure we are talking the same ROI. Is it the IDLgrROI, or IDLanROI for that matter? I think that their ContainsPoits method will return the same results whether the trimmed edge of the ROI consists of two or thousand points. So it looks to me like drawing a line from one point of contact to the other is effectively the same as following the edge of the draw widget where it has crossed the draw boundary.

Cheers,
Pavel

Subject: Re: Moving an ROI
Posted by [Andre Kyme](#) on Fri, 07 Dec 2001 01:00:53 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Pavel,

Not using objects I'm afraid, just run-of-the-mill stuff.

I've gone with the mask method and got it up and running - the truncation of the ROI at the edges occurs by drawing a line from the points of contact rather than following the boundary, but this is OK because the ROI is actually considered unacceptable unless the entire array of X and Y vertices are within the draw widget area.

Think it works OK - but I've said those famous last words before.

Maybe I'll look into objects next year,
Thanks,
Andre
