Subject: Re: XSTRETCH in P.Multi - is it efficient?
Posted by David Fanning on Sat, 06 Apr 2002 15:21:32 GMT
View Forum Message <> Reply to Message

Reno Choi (K.Y.Choi@soton.ac.uk) writes:

> Now, I am trying to make widget program that contains two plot with a
> image. I also would like to include mouse-oriented indication bar in
> both plot area, which synchronise with the position of the image. I
> guess it would be based on something like XSTRETCH.pro from D.Fanning.
>
> Question are, if I put those plots and image in a simgle Widget_Draw,
> would it become much slower while interactive moving lines on plots
> are moving? I wonder this is one of reason why XSTRETCH.pro has two
> windows, i.e. two TLB.

No, it would not become much slower. The reason
for two windows had to do with ease of resizing.
At the time the program was written the reliability
of the widget geometry and cross-platform sizing
issues were worse than they are now. I wished to avoid
such problems in an example program.

> In fact, since I tend to include loop that scans through the data
> range, approximately 5000 steps within 1 or 2 minutes, speed really
> matters to me. XSTRETCH continuously uses PIXMAP while moving lines
> with mouse, and I guess such processing is possibly slower in, for
> instance, 800-by-800 Widget_Draw.

No, an 800-by-800 window is piddly for modern computers.
(I'm not sure about OpenVMS computers.) And, anyway, you
don't have to copy the entire window with the Device Copy
technique, just the part of the window that needs repair.

> Alternatively, I thought three different Widget_Draws in a single TLB.
> Problem again comes when resize windows. Also, I am not sure how much
> difference in processing speed with comparison to the pervious method,
> and which method would be less complicate and more economical in
> programming-wise.

Separating three functions into three draw widgets
rather than one greatly simplifies the event handling
code, because you don't have to figure out whether the
user started in this part of the draw widget and dragged
to the other, etc. (You can think "no user would be stupid
enough to do something like that", but such thinking is
sure to doom your code.)

Resizing a window with more than one thing in it is
more complicated, but it is not onerous. You just
divide the total size up into proportional "thing" sizes
and size each thing in turn. Problems sometimes arise
when some "things" are not suppose to resize, etc. But
these things can always be worked out with intelligent
design.

Cheers,

David

--
David W. Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155