Subject: Re: Testers needed for TV benchmark
Posted by John-David T. Smith on Wed, 08 Aug 2001 21:12:55 GMT
View Forum Message <> Reply to Message

"Bill B." wrote:
>
> "Liam E. Gumley" <Liam.Gumley@ssec.wisc.edu> wrote in message
news:<3B7183C6.4D1A3622@ssec.wisc.edu>...
>
>> To obtain the best frame rate for animations, first you should display
>> all the images in a pixmap window, and *then* use DEVICE, COPY to copy
>> each image to a visible graphics window. I'm willing to bet you'll get
>> frame rates better than 10 frames/sec using this method.
>>
>
> Hi Liam,
>
> If you look at my benchmark, that is the 1st of the two tests being
> executed:
>
> <snip>
>

```
> FOR I = 0, 49 DO BEGIN
>   WSET, pixmap0_id
>   TV, data, true = 3
>   WSET, pixmap1_id
>   DEVICE, COPY = [0, 0, sz-1, sz-1, 0, 0, pixmap0_id]
> ENDFOR
```

>
> <snip>
>
> Also, the results I posted show no difference between the technique
> that you describe and just TVing directly to the visible window. Ten
> fps at 512*512 would be great but I see no indication that I can
> achieve that on a PC. BTW, this benchmarking is in preparation for
> what will be the SW end of a generic (any format) video frame grabber.
> Could you verify if the snippet above is what you had in mind?

I suspect Liam actually meant something more like:

```
WSET, pixmap0_id
TV, data, true = 3
T0 = SYSTIME(1)
FOR I = 0, 49 DO BEGIN
  WSET, pixmap1_id
  DEVICE, COPY = [0, 0, sz-1, sz-1, 0, 0, pixmap0_id]
ENDFOR
```

That is, preload your pixmap with the image. This will most certainly result in much higher frame rates, but incurring the expense of preloading. This technique can be useful for animation if you have the sequence of images to animate. Preload them all into their own pixmaps, and then blitz through them.

For real, dynamic frame rate, this obviously won't do. An example might be a filter being applied interactively, data streaming from a camera, etc. Here I could only say that IDL isn't really designed to support high frame rate streaming video. You might be able to accomplish something via external DLM code which fills a pixmap buffer (presumably much more rapidly than IDL can), and then let IDL retrieve data from that pixmap, but otherwise I think you're stuck.

TV will work much faster (especially with some video hardware), if you relax the need for TRUECOLOR. A single plane of 8-bit resolution data run through a hardware colortable will draw much more rapidly. I got frame rates for 512x512 images on the order of 13fps at 8 bit, on pretty pokey hardware.

This brings up another interesting technique which has to do with multiple draws to a given window -- a commonly enough performed operation in IDL. If you, for instance, need to plot several different data sets, overlay lines and annotation, while allowing the plot to be interactively modified (e.g. a changing parameter), a vastly superior solution to direct drawing is available as "double buffering". It's quite simple. As in this example, you simply perform all your draw operations to a pixmap, and then device,COPY= the resultant image over. If you have multiple draw operations (unlike in Bill's case), this will be noticeably faster. It will also be smoother, eliminating the jerkiness you may have experienced with interactive plotting.

JD