Subject: Re: Testers needed for TV benchmark
Posted by Liam E. Gumley on Wed, 08 Aug 2001 18:24:06 GMT
View Forum Message <> Reply to Message

"Bill B." wrote:
> I've been working at trying to get the fastest TV update rate possible
> and am a little discouraged by my results so far. I developed a short
> benchmark program and am curious what results might be obtained by
> other machines. I have a P3-550 and obtained the following:
>
> TV-Device Copy Test: Array Size: 512 13.520000 seconds
> => Frames/sec: 3.6982249 Frame time: 0.27040000
> Video Xfer rate: 2908402.4 bytes/sec
>
> TV Direct Test: Array Size: 512 13.290000 seconds =>
> Frames/sec: 3.7622272 Frame time: 0.26580000
> Video Xfer rate: 2958735.9 bytes/sec
>
> If your time permits, I would appreciate knowing anyone else's results
> and a brief mention of the computer used. My benchmarks were
> identical for IDL 5.2 and 5.3.1. Thanks in advance.

Bill,

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.

Cheers,
Liam.
Practical IDL Programming
http://www.gumley.com/

Subject: Re: Testers needed for TV benchmark
Posted by billb on Wed, 08 Aug 2001 20:55:29 GMT
View Forum Message <> Reply to Message

"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?

Thanks a bunch,
Bill

Subject: Re: Testers needed for TV benchmark
Posted by Pavel A. Romashkin on Wed, 08 Aug 2001 21:09:49 GMT
View Forum Message <> Reply to Message

Velocity Engine Support Enabled
IDL Version 5.4 (MacOS PowerMac). (c) 2000, Research Systems, Inc.
Welcome to IDL for Macintosh.

```
TV-Device Copy Test: Array Size:      512         3.5257369 seconds
  =>  Frames/sec:        14.181432   Frame time:     0.070514739
   Video Xfer rate:        11152732. bytes/sec
TV Direct Test: Array Size:      512         3.0961030 seconds
 =>  Frames/sec:        16.149334   Frame time:     0.061922059
   Video Xfer rate:        12700353. bytes/sec
```

"Bill B." wrote:
>
> Hi all,
>
> I've been working at trying to get the fastest TV update rate possible
> and am a little discouraged by my results so far. I developed a short

> benchmark program and am curious what results might be obtained by
> other machines. I have a P3-550 and obtained the following:
>
> TV-Device Copy Test: Array Size: 512 13.520000 seconds
> => Frames/sec: 3.6982249 Frame time: 0.27040000
> Video Xfer rate: 2908402.4 bytes/sec
>
> TV Direct Test: Array Size: 512 13.290000 seconds =>
> Frames/sec: 3.7622272 Frame time: 0.26580000
> Video Xfer rate: 2958735.9 bytes/sec
>
> If your time permits, I would appreciate knowing anyone else's results
> and a brief mention of the computer used. My benchmarks were
> identical for IDL 5.2 and 5.3.1. Thanks in advance.
>
> Bill B.

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

Subject: Re: Testers needed for TV benchmark
Posted by Pavel A. Romashkin on Wed, 08 Aug 2001 21:23:38 GMT
View Forum Message <> Reply to Message

JD Smith wrote:
>
> "Bill B." wrote:
>>
> 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.

I was trying this as JD sent this post. I must admit that there was no
speed increase, to my dismay.
Pavel

Subject: Re: Testers needed for TV benchmark
Posted by Liam E. Gumley on Wed, 08 Aug 2001 21:33:10 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?

Bill,

The rate at which you can *display* images is different than the rate at which you can *animate* images. To demonstrate, the following procedure uses two loops. The first loop displays a sequence of images in pixmap windows. The second loop copies images from the pixmap windows to a visible graphics window:

```
PRO TEST, DELAY=DELAY

;- Set initial parameters
xsize = 512
ysize = 512
nframes = 20
image = dist(512)
if (n_elements(delay) eq 0) then delay = 0.05

;- Display image in pixmap windows
t0 = systime(1)
for frame = 0, nframes - 1 do begin
  window, frame, /pixmap, xsize=xsize, ysize=ysize
  tvscl, shift(image, 10 * frame)
endfor
t1 = systime(1)
print, 'Display rate (frames/sec) = ', $
  float(nframes) / (t1 - t0)
```

```
;- Animate the images
window, /free, xsize=xsize, ysize=ysize
t0 = systime(1)
for frame = 0, nframes - 1 do begin
  device, copy=[0, 0, xsize, ysize, 0, 0, frame]
  wait, delay
endfor
t1 = systime(1)
print, 'Animation rate (frames/sec) = ', $
  float(nframes) / (t1 - t0)

END
```

If the DELAY keyword is not set, the default interval between each frame is 0.05 seconds. Set DELAY=0.0 to see how fast your system can copy images from pixmap windows to a visible window. On my PC running IDL 5.3:

```
IDL> test
Display rate (frames/sec) =       4.0265754
Animation rate (frames/sec) =       15.974441
IDL> test, delay=0.0
Display rate (frames/sec) =       4.0104271
Animation rate (frames/sec) =       56.980046
```

To summarize, the rate at which you can display the images with TV or TVSCL is much slower than the rate at which you can animate the images using DEVICE, COPY=[...].

Cheers,
Liam.
Practical IDL Programming
http://www.gumley.com/

Subject: Re: Testers needed for TV benchmark
Posted by Liam E. Gumley on Wed, 08 Aug 2001 21:47:26 GMT

View Forum Message <> Reply to Message

The true color version of my test procedure gives slightly lower display and animation rates, at least on my system:

```
PRO TEST_TRUE, DELAY=DELAY

;- Load the image
filename = filepath('rose.jpg', subdir='examples/data')
read_jpeg, filename, image
```

```
xsize = 3 * 227
ysize = 3 * 149
image = rebin(image, 3, xsize, ysize)

;- Set initial parameters
nframes = 20
if (n_elements(delay) eq 0) then delay = 0.05

;- Display image in pixmap windows
t0 = systime(1)
for frame = 0, nframes - 1 do begin
  window, frame, /pixmap, xsize=xsize, ysize=ysize
  tv, shift(image, 0, 10 * frame, 0), true=1
endfor
t1 = systime(1)
print, 'Display rate (frames/sec) = ', $
  float(nframes) / (t1 - t0)

;- Animate the images
window, /free, xsize=xsize, ysize=ysize
t0 = systime(1)
for frame = 0, nframes - 1 do begin
  device, copy=[0, 0, xsize, ysize, 0, 0, frame]
  wait, delay
endfor
t1 = systime(1)
print, 'Animation rate (frames/sec) = ', $
  float(nframes) / (t1 - t0)

END

IDL> test_true
Display rate (frames/sec) =       3.3394557
Animation rate (frames/sec) =       15.491868
IDL> test_true, delay=0.0
Display rate (frames/sec) =       3.3344448
Animation rate (frames/sec) =       45.351475
```

Cheers,
Liam.
Practical IDL Programming
http://www.gumley.com/

Subject: Re: Testers needed for TV benchmark
Posted by Pavel A. Romashkin on Wed, 08 Aug 2001 21:58:39 GMT
View Forum Message <> Reply to Message

"Pavel A. Romashkin" wrote:
>
> I was trying this as JD sent this post. I must admit that there was no
> speed increase, to my dismay.

Duh! It helps when you place the timer in the right place. I was timing everything, including preloading. Thank you, Liam. Of course, the correct output for preloaded pixmap windows is

```
TV-Device Copy Test: Array Size:      512        0.71153700 seconds
  =>  Frames/sec:        70.270414   Frame time:     0.014230740
   Video Xfer rate:        55262902. bytes/sec
TV Direct Test: Array Size:      512         3.1362751 seconds
 =>  Frames/sec:        15.942479   Frame time:     0.062725501
   Video Xfer rate:        12537676. bytes/sec
```

Sorry.
Pavel

Subject: Re: Testers needed for TV benchmark
Posted by billb on Thu, 09 Aug 2001 14:46:58 GMT
View Forum Message <> Reply to Message

Liam, Pavel and JD + others,

Thanks for your generous input. I now understand the multiple TV-then multiple update approach, but it does not fit what I need to do. I used the Profile function which fingers TV as the timehog. However, given the strength of the IDL array-based engine, I'm doubting that I can improve upon the image update speed with an external DLL - especially since this goes beyond my experience. Beyond increasing the CPU speed and/or transferring less data, I guess it is what it is. Thanks to all again.

Bill B.

Subject: Re: Testers needed for TV benchmark
Posted by karl_schultz on Thu, 09 Aug 2001 15:42:37 GMT
View Forum Message <> Reply to Message

I just want to point out something to look out for, since it is rather counter-intuitive and a bit surprising. I discovered this on a Linux system while trouble-shooting a performance problem.

X servers often implement pixmaps by storing them in unused

(off-screen) frame buffer memory under the assumption that the performance of these pixmaps is better than the ones in system memory. The idea is that blitting the pixmap to a window would be faster since it is just a copy within a memory subsystem, rather than a transfer from system memory to frame buffer memory.

Of course, once the unused frame buffer memory is exhausted, additional pixmaps are stored in system memory.

The surprise is that, on some X servers, the off-screen pixmaps are actually slower than the system memory pixmaps. I can't explain this, other than to suspect an implementation problem or oversight in the X server code.

You can detect this by caching a series of, say, 50 images in pixmaps, as described in other posts in this thread. Then, measure the time it takes to copy each one to a window. Or, just watch the animation. If the first few seem slower, then this might be the problem.

One way to "fix" this is to create a set of dummy pixmaps not used in the actual animation to "use up" the slow pixmaps. Then, the rest of the pixmaps will all be the "fast" ones. Of course, this is all too dependent on the X server, frame buffer memory size, desktop size, and size of the pixmaps.

Even when the X server implementation is better, that is, the off-screen pixmaps are faster, this can cause confusing benchmark results since the copy times can vary in hard-to-describe ways.

Karl

"Pavel A. Romashkin" <pavel.romashkin@noaa.gov> wrote in message news:<3B71B610.DDB1A33A@noaa.gov>...
> "Pavel A. Romashkin" wrote:
>>
>> I was trying this as JD sent this post. I must admit that there was no
>> speed increase, to my dismay.
>
> Duh! It helps when you place the timer in the right place. I was timing
> everything, including preloading. Thank you, Liam. Of course, the
> correct output for preloaded pixmap windows is
>
> TV-Device Copy Test: Array Size: 512 0.71153700 seconds
> => Frames/sec: 70.270414 Frame time: 0.014230740
> Video Xfer rate: 55262902. bytes/sec
> TV Direct Test: Array Size: 512 3.1362751 seconds
> => Frames/sec: 15.942479 Frame time: 0.062725501

> Video Xfer rate: 12537676. bytes/sec
>
> Sorry.
> Pavel

Subject: Re: Testers needed for TV benchmark
Posted by R.Bauer on Sun, 19 Aug 2001 18:20:01 GMT
View Forum Message <> Reply to Message

"Bill B." wrote:
>
> Liam, Pavel and JD + others,
>
> Thanks for your generous input. I now understand the multiple TV-then
> multiple update approach, but it does not fit what I need to do. I
> used the Profile function which fingers TV as the timehog. However,
> given the strength of the IDL array-based engine, I'm doubting that I
> can improve upon the image update speed with an external DLL -
> especially since this goes beyond my experience. Beyond increasing
> the CPU speed and/or transferring less data, I guess it is what it is.
> Thanks to all again.
>
> Bill B.

Dear Bill

I have an animation and displaying routine and I have rates about 10fps.
This is tested by your data. At the moment I am not using pixmap and
device copy.

The images are created first and than stored into a structure which is
saved
as idl sav file but the data could be hold in memory too if you like.

We are using the make_anim routine which writes the image data to a sav
file.

http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_source/idl_ html/dbase/download/make_anim.tar.gz

The animation data could be read by read_anim
http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_source/idl_ html/dbase/download/read_anim.tar.gz

The widget routine x_animate displays these kind of files
multiple gifs are supported too.
This widget has the nice feature thats during loop
all buttons are active.

http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_source/idl_ html/dbase/download/x_animate.tar.gz

regards
Reimar

--
Reimar Bauer

Institut fuer Stratosphaerische Chemie (ICG-1)
Forschungszentrum Juelich
email: R.Bauer@fz-juelich.de
http://www.fz-juelich.de/icg/icg1/
========== ======
a IDL library at ForschungsZentrum Juelich
http://www.fz-juelich.de/icg/icg1/idl_icglib/idl_lib_intro.h tml

http://www.fz-juelich.de/zb/text/publikation/juel3786.html
========== ======

read something about linux / windows
http://www.suse.de/de/news/hotnews/MS.html